Essays in Decentralized Finance

Dissertation
submitted to the
Faculty of Business, Economics and Informatics
of the University of Zurich

to obtain the degree of
Doktor der Wirtschaftswissenschaften, Dr. oec.
(corresponds to Doctor of Philosophy, PhD)

presented by
Luzius Meisser
from Klosters GR

approved in February 2024 at the request of

Prof. Dr. Thorsten Hens
Prof. Dr. Claudio Tessone
Prof. Dr. Marco Dell'Erba

The Faculty of Business, Economics and Informatics of the University of Zurich hereby authorizes the printing of this dissertation, without indicating an opinion of the views expressed in the work.

Zurich, 14th February 2024

The Chairman of the Doctoral Board: Prof. Dr. Steven Ongena

Acknowledgements

First, I would like to wholeheartedly thank my wife Verena for her support and patience with my pursuit of a doctorate in finance over the course of the past seven years. Then, I would like to express my highest gratitude to Thorsten Hens for making this thesis possible, his invaluable inputs, the inspiring discussions and his kind openness to unconventional topics. Likewise, I would like to thank Claudio Tessone and Marco Dell'Erba for their bold participation in this interdisciplinary endeavor and their highly appreciated thoughts. Lastly, I would like to thank the University of Zurich for its institutional support on my academic journey over the past ten years.

Abstract

This is a cumulative, interdisciplinary dissertation that revolves around legal, economic and technical questions in the field of decentralized finance. It presents the Frankencoin, an oracle-free stablecoin with veto-based governance, and employs tools from game-theory and finance to substantiate its soundness. It further introduces the continuous capital corporation, a novel concept for how companies should price their own shares when continuously offering them in an automated way, relying on blockchain technology and stablecoins like the Frankencoin for settlement. Finally, two related legal articles show how financial intermediaries can offer crypto custody and staking under Swiss law, thereby helping to clear the path towards a more open and free financial system from the regulatory side.

Contents

Acronyms and Abbreviations

AG	Aktiengesellschaft / corporation
AML	Anti money-laundering
Art.	Article
BBI	Bundesblatt
BIS	Bank for International Settlement
BTC	Bitcoin
BTCCHF	Bitcoin to Swiss franc exchange rate
CHF	Swiss franc
CBDC	Central bank digital currency
DAO	Decentralized Autonomous Organization
Dr	Doktor / doctor of philosophy
eds	Editors
e.g.	Exempli gratia / for example
EVM	Ethereum Virtual Machine
ERC-20	Ethereum Request for Comment No. 20
ESTV	Eidgenössische Steuerverwaltung / Swiss Federal Tax Administration
ETH	Ether cryptocurrency
ETH	Eidgenössische Technische Hochschule
FATF	Financial Action Task Force
ff	Folio / and following
fn	Footnote
GR	Grisons
ICO	Initial Coin Offering
i.e.	Id est / that is
KOV	Verordnung über die Geschäftsführung der Konkursämter
max	Maximum
min	Minimum
MKR	Maker governance token

N	Note
No	Number
noop	No operation
Oec	Oeconomiae / economics
OECD	Organisation for Economic Co-operation and Development
p.	Page
para.	Paragraph
pp.	Pages
PhD	Doctor of philosophy
Prof	Professor
QR	Quick Response
RWA	Risk weighted average
SchKG	Bundesgesetz über Schuldbetreibung und Konkurs
SR	Systematische Rechtssammlung
UK	United Kingdom
UNI	Uniswap governance token
US	United States
USD	United States Dollar
USDC	Circle USD Stablecoin
USDT	Tether USD Stablecoin
UST	Terra USD Stablecoin
VaR	Value at risk
vs	Versus
XCHF	Crypto Franc
ZCHF	Frankencoin
ZGB	Zivilgesetzbuch

Chapter 1

Introduction

This introduction sets the stage for the main contributions. It summarizes my journey, establishes the deeper context of decentralized systems, and assesses the legal developments with a view on the impact of the relevant chapters.

1.1 Interdisciplinary Journey

This thesis is an embodiment of the thoughts that have been breeding in my mind for the past seven years. It is an interdisciplinary endeavor in which I try to symbiotically combine insights from different fields with the aspiration to create something novel and maybe even useful. The thesis at hand leverages my background in computer science and my experience as a software engineer, makes use of the knowledge gained through my studies in economics and finance, my entrepreneurial experience in the blockchain sector, and lastly, the insights I have gained as a co-author of legal publications and through regulatory consultations. Adams (2013) notes that commanding a unique combination of skills – or *skill stack* – can yield a noteworthy output even if none of the combined skills are extraordinary in isolation.

My PhD started with the plan to leverage my software engineering experience to build agent-based economic simulations. I did so in the publication *An Agent-based Simulation of the Stolper-Samuelson Effect* (Meisser and Kreuser, 2017) and later also in a conference contribution titled *Seemingly Equivalent Firm Decision Heuristics* (Meisser, 2016c). For a few semesters, I was teaching a course *Agent-Based Financial Economics*, during which I tried to convey a unifying view at the intersection of financial economics and computer science, with many assignments involving the programming of agents in a simulated economy.[1]

Whereas economists and mathematicians are often interested in a problem's solution and whether it exists, computer scientists pay more attention to the path on how a solution can be found. Computer science has even developed its notation to specify how hard it is to solve a problem and to quantify its computational complexity (Widmayer and Ottmann, 1993). The focus on the path toward the solution is a necessity when creating an agent-based computer simulation to replicate an economic model. The main difficulty is often not finding the mathematical equilibrium of the setting of interest but structuring the simulation such that the agents collectively find the optimal outcome without centralized coordination. This requires the development of a deep understanding of the problem. Feynman (1988) expressed this insight with the words: *"What I cannot create, I do not understand"*. While economists typically see software as a tool for performing calculations, the view from a software engineering perspective is fundamentally different: here, the software is the model and building the software is understanding the model.

I have tried to advocate the software engineering perspective on economic models in my article *The Code is the Model* (Meisser, 2017). However, the impact seems to have been very limited. While it is commonly accepted to consider the mathematical formulation of a problem as its specification, the source code of a model is still rarely considered its specification. This might partially be owed to programming languages often demanding a level of attention to detail that is not required for understanding an overall concept, whereas mathematical models are more suitable to concisely express high-level abstractions. While I had many valuable insights on how an economy functions by trying to simulate it, it also became apparent that this research direction offered less reward for the effort than I originally hoped for.

Meanwhile, developments in the field of blockchain technology enabled the creation of so-called smart contracts that can be used to create small pieces of software that run on a blockchain. These smart contracts are the basis of decentralized finance and allow the creation of decentralized exchanges, decentralized stablecoins, decentralized lending protocols, and more generally decentralized autonomous organizations. In decentralized finance, computer code governs the world and the ability to program reliable smart contracts that render an economically meaningful service is invaluable. Creating decentralized applications requires a combination of skills

[1] The source code of the simulation used in the course can be found on github.com/meisser/course2019.

from computer science, financial economics, and law. This development increasingly pulled my away from the topic of agent-based simulations and towards the field of crypto currencies and blockchain technology.

Having experienced the hype around initial coin offerings and having seen their short-comings, I tried to push forward a more sustainable model on three fronts.

First, I started advocating the creation of tokens that represent real value. The crypto currencies typically issued in initial coin offerings during the hype years 2017 and 2018 often left the investors without anything tangible in their hands. Unlike securities, tokens do not come with any shareholder rights or other forms of investor protection. Therefore, I argued as early as 2016 that Switzerland should adjust its laws to legally allow the issuance of traditional securities in the form of tokens (Meisser, 2016a; Meisser, 2016b). To my great astonishment, such a change of law was actually implemented only a few years later, allowing Swiss companies to issue their shares in the form of blockchain-based security tokens.

Second, it is essential to be able to transact in the right reference currency when trading security tokens. This requires a representation of the desired reference currency on the blockchain. To that end, I helped initiating *Swiss Crypto Tokens AG*, the company that launched the Swiss franc stablecoin named CryptoFranc (Finews, 2018). In the meantime, this company has been fully absorbed by Bitcoin Suisse, but the CryptoFranc is still in circulation using the initially deployed smart contracts. More true to the spirit of blockchain technology are decentralized stablecoins that do not depend on an issuer. With the Frankencoin, an attempt at creating such a stablecoin is presented in this thesis.

Third, while the tokenization of assets helps in making them digitally transferable, it does not magically create a liquid market. To that end, it is necessary to establish trading venues and to have market makers that provide liquidity. The traditional way to do so is through a centralized exchange. But again, the more interesting way of organizing a market for blockchain-based assets is to rely on decentralized mechanisms. Here, the decentralized exchange Uniswap pioneered a novel concept to create liquidity. Instead of having a traditional order book, it relies on liquidity pools to which anyone can contribute. Inspired by this, I supported the unfortunately unsuccessful startup Alethena and later co-founded the company Aktionariat that provides technical services to tokenize the shares of Swiss companies and to make them tradable through a tool named *Brokerbot*. Unlike Uniswap, the liquidity in the Brokerbot comes from the issuer alone, giving raise to the questions tackled in chapter 3 *The Continuous Capital Corporation.*

However, there also is a forth dimension besides the three mentioned above that is not explored herein and that could play a crucial role in bootstrapping a thriving decentralized financial market. The forth dimension concerns transparency about the state of the company. Financial markets work most smoothly if all participants are fully informed about the state of the world. In practice, this is rarely the case and there is a risk of insiders exploiting asymmetries in information. Traditional markets have institutionalized procedures to update shareholders with audited reports at established intervals and various mechanisms to guard against insider trading. Unfortunately, these are neither very effective nor efficient and cannot be directly applied to smaller companies without burdening them with disproportionate costs. Instead, new ways to keep investors informed need to be explored, which might be an interesting task for future work, especially in the context of blockchain technology. Having all relevant data represented on a public blockchain could potentially lead to real-time transparency with regard to the accounting of a company, as noted by Wagner (2017). For decentralized autonomous organizations like the Frankencoin protocol, real-time transparency over every single transaction already is a reality by design.

Most blockchain based protocols rely on a combination of technical solutions and economic incentives to design a self-reliant system that functions without explicit coordination between its participants and that is ideally also carefully embedded in the applicable legal context. A surprisingly large part of the spendings of a typical crypto company goes into legal and compliance. Ripple's legal defense in the dispute with the SEC is said to cost 200 million dollars alone (Browne, 2023). The three touched fields seem to not only share their high relevance in the context of crypto currencies and smart contracts, but also a common theme that is explored in the subsequent section.

1.2 Decentralized Systems

This section qualitatively explores a recurring theme of the three aforementioned disciplines, computer science, financial economics, and law, namely *decentralization.* They are all in one way or another concerned with the organization of complex systems through decentralized decision taking.

In computer science, this observation applies two-fold. First, the field of distributed system is devoted to creating systems that are robust and scalable without depending on a single point of failure. Second, the field of software engineering developed principles for handling complexity in the specification of large systems, resulting in ideas like object-oriented programming to maintain a clean separation of concerns. In both, it is key to take decisions as closely as possible to the problem that is to be solved by relying on locally available information, and at the same time shielding other system components from unnecessary complexity.

In economics, one pivotal concern is whether the economy, a highly complex system, is better organized through central planning or by letting consumers and firms take decisions on their own. One of the most significant works in economics is the proof of the existence of an equilibrium in a competitive market without central planning by Arrow and Debreu (1954). While the proof of the existence of an equilibrium does not provide a recipe for the market participants to find that equilibrium, this result is generally seen as an indication that competitive markets can function equally well as a market that is under the full control of a benevolent and omniscient authority. Taking into account the earlier qualitative arguments from Hayek (1945), one can arrive at the conclusion that a competitive market often is the superior way to organize an economy. While Arrow and Debreu (1954) focus on mathematical considerations, Hayek (1945) exhibits an intuitive understanding for the benefits of decentralized decision taking and for the important role of local information.

From a computer science perspective, free markets can be seen as distributed systems with prices carrying the information necessary to coordinate the behavior of the participants. With this view, the search for an equilibrium in a competitive economy can be analyzed for its computational complexity as for example Chen et al. (2009) have done. I for one found the insight of Feldman (1973) particularly useful. He shows that in the absence of money, it might be necessary to find long chains of multi-party trades to move closer to the equilibrium, whereas in an economy with money, the problem of the market participants is reduced to finding Pareto-improving trades between two parties. Knowing this tremendously simplified the complexity of the agents my students were tasked with implementing in the *Agent-Based Financial Economics* course. It also exemplifies how a complexity view can provide a more intuitive explanation for the usefulness of money than economic equilibrium models.

While computer science typically sees the benefits of decentralization in scalability and reliability, there is another invaluable benefit in the context of economic systems: competition. By definition, competition depends on having multiple independent entities that fulfill the same or a similar function. Competition is the key ingredient to allow free markets to achieve the optimal outcome. Like with decentralized systems in computer science, the system as a whole gains a new quality by having multiple entities fulfilling the same function in parallel.

Similarly, in law and politics, there are various principles that aim at decentralized decision taking. A healthy legal system is founded in a well-defined separation of powers and follows a principles-based approach that allows for agile decision taking by those that are closest to the matter, i.e. those in possession of the local information. The legislature formulates general norms, with the executive refining them and the judicature applying them, filling gaps, and maybe even adjusting them in case of conflicts with established principles (Meier-Hayoz, 1951). Just like a consumer is in a better position to choose a pair of shoes for himself than a central planner in a distant city, a judge that sees a concrete case in all its details is in a better position to decide what the best course of action is than a member of parliament that has to come up with abstract rules in advance. Like modules in a large piece of software, the local processing of information shields other system components from unnecessary complexity. From an economic perspective, the separation of powers in politics serves the prevention of conflicts of interest and concentration of power. From a software engineering point of view, it also is a way to separate distinct concerns in order to manage complexity.

Having a healthy separation of concerns increases the agility of the system and makes it easier to introduce changes. For example, when Bitcoin Suisse approached the Swiss Federal Tax Administration (ESTV) in 2014 with questions on whether VAT should be charged on the sale of crypto currencies, ESTV had the freedom to take that decision on its own and to answer the inquiry within reasonable time. In contrast, when the German tax administration was approached, they concluded that the German law did not leave them with the freedom of taking a decision in the matter and it took much longer to reach a satisfying answer, involving the parliament and the judiciary. This is a typical symptom of too much centralized decision taking. While from the perspective of the centralized decision taker, having more power and more to say might be desirable, taking away autonomy from the peripheral system participants generally slows down the adaption to new circumstances, just like a large software system with a non-modular design is much harder to change than a similar software system with a clean division of responsibilities.

In the end, I believe that being aware of the design principles for complex systems in combination with an economic perspective can be of enormous value in the design of legal systems. As an example, consider the insight by Tainter (1988), who observed that given a state with the capacity to enforce 1000 laws, the introduction of a well-intended 1001st law will have a net negative effect if it is less useful than the average existing law. Even if the law is beneficial on its own, this benefit is cancelled out by the dilutive effect on the executive's and judiciary's capacity to enforce the other 1000 laws. His bleak thesis is that every empire will eventually crumble under the weight of its own bureaucracy. I do not share that pessimism and remain optimistic about society's capacity to reinvent itself, potentially with the help of blockchain technology.

1.3 Legal Developments

Having a relatively decentralized political system itself, Switzerland early on became a crystallization point for notable crypto ventures. The most relevant is Ethereum, which pioneered the idea of an initial coin offering through a Zug based foundation (Buterin et al., 2014). Today, the Ethereum system has a higher market capitalization than all Swiss banks combined. In this section, I point out what made Switzerland a fertile ground for crypto startups and how the two legal articles presented in chapter four and five helped shaping Switzerland's regulatory environment. Furthermore, I shortly touch the legal questions that are raised by the Frankencoin system.

1.3.1 Crypto Custody

One of the elemental services of crypto asset service providers is the custody of crypto assets for their clients. Economically, the desired relation is clear: the client wants to delegate the safe-keeping of their assets to a specialized service provider while retaining legal ownership. They want strong property rights over their crypto assets even when they delegate the handling to a third party. Conceptually, this usually is addressed by distinguishing ownership and possession. Unfortunately, Swiss law reserves this distinction to physical items. With Meisser and Hauser (2018), we started to explore the possibility of distinguishing possession and ownership despite the literal interpretation of the law suggesting otherwise. This view was later substantiated in the article that is included in this thesis as chapter 4, originally published in German under the title *Verfügungsmacht und Verfügungsrecht an Bitcoins im Konkurs*. Our key argument to reach separability in bankruptcy under Swiss law was to distinguish the power of disposal from the right of disposal in analogy to possession and ownership. Liechtenstein's blockchain law, that was adopted in the following year, makes explicit use of that distinction when regulating the custody of crypto assets, using the slightly deviating terms *Verfügungsgewalt und Verfügungsberechtigung*, which in English also translate to power of disposal and right of disposal (Landesverwaltung, 2019).

A few months after our publication, on September 3rd 2018, Finma published a fact sheet titled *Faktenblatt Virtuelle Währungen*, which agreed that crypto currencies stored on a dedicated address for a client are indeed to be treated like a physical item in custody. Thereby, Finma recognized that such segregated crypto assets are not part of the bankruptcy estate and do not appear on the balance sheet of the custodian. [2] Formally, this distinction is first and foremost a matter of civil and bankruptcy law, so one might assume that it would be up to the cantonal bankruptcy offices to establish a legal practice. However, the one cantonal office that we approached did not want to take a position, indicating that they would leave the resolution of this question to Finma. Unfortunately, Finma only partially followed our assessment and did not recognize the possibility of collective custody, where the assets of multiple clients are stored on the same blockchain-based address. The consequence of that is that crypto assets in collective custody are considered deposits, which require a banking license from the custodian unless one of the exceptions of the banking act applies.

Soon after Finma, the Federal Council published a report on crypto assets (Swiss Federal Council, 2018). The report concludes, among other things, that it would be desirable to adjust the bankruptcy law to create legal certainty with regards to the questions we raised, quoting our and other related articles. Based on this report, multiple working groups were formed to come up with a package of legal changes to support distributed ledger technology in Switzerland, with me being fortunate enough to be one of the consulted experts in the creation of the law (Swiss Federal Council, 2019b).

[2] Today, this fact sheet is named *Faktenblatt Kryptobasierte Vermögenswerte* and the original version from 2018 is not publicly available any more.

The proposed law touched three legal areas: the creation of crypto securities, the trading of crypto securities, and the custody of crypto assets. I was involved in two academic publications that contributed to this development, one with main author Martin Monsch on how to issue tokenized shares before the law was adapted (Crone, Monsch, and Meisser, 2019) and one with the responsible working group of the Swiss Blockchain Federation publicly commenting on the first published draft of the law (Kuhn et al., 2019). Over the course of 2020, the law unanimously passed both chambers of the national parliament (Swiss Federal Council, 2019a). It was set into force partially by February 1st 2021 (Swiss Federal Council, 2020) and fully by August 1st 2021 (Swiss Federal Council, 2021).

1.3.2 Staking Controversy

The more recent paper that is presented in chapter five concerns the *staking* of crypto currencies. Staking is the process of providing crypto currencies as a collateral in order to be allowed to take part in the processing of the transactions of a blockchain that relies on 'proof of stake' as opposed to 'proof of work'. The purpose of the collateral is to provide the system with a possibility to punish ('slash') the owner in case of malicious or otherwise incorrect behavior. At the same time, there is a reward for correct behavior.

Economically, the staking reward is an income earned through the provision of computing power to the system. But since it is proportional to the provided stake, it might look like an interest or other financial income, which would potentially subject it to financial market laws. Furthermore, the provided collateral ('stake') is locked for a certain period even after the owner stops to do staking in order to give the system enough time to determine whether the owner needs to be slashed. The publication presented in chapter five already showed in 2021 why this should not prevent the staked crypto assets to be considered 'available at all times', which is a requirement for the legally separable custody of crypto assets under the newly created law.

Unfortunately, Finma does not seem to share this interpretation of the law. According to a presentation recently held at University of Zurich by a Finma representative, staked crypto assets generally are not to be considered to be *available* within the meaning of the newly created article 242a SchKG concerning the custody of crypto assets (Obrecht, 2023). As a consequence, it would no longer be possible to hold staked crypto assets on behalf of a client. Instead, it would have to be treated as owned by the custodian, with the client only being left with a monetary claim instead of an ownership-like legal position. This would turn the staked assets into bank deposits and consequently make it necessary to obtain a banking license before offering staking services. However, even as a bank, it would not be economically feasible to offer staking services as crypto assets are subject to extreme capital requirements designed to deter banks from holding crypto assets on their balance sheet.

Overall, one can observe a shift in the regulatory regime from a ‘regulate’ approach to a ‘contain’ strategy, under which crypto assets are not treated under the maxim of ‘same risks, same rules’, but instead are faced with stricter regulatory requirements than traditional assets (Aquilina, Frost, and Schrimpf, 2023). Of particular concern in this regard are the upcoming capital requirements imposed by the Basel Committee (Basel Committee on Banking Supervision, 2022). Among other things, the Basel Committee requires banks to hold at most 1% of their tier one capital in crypto assets. This means that a bank holding 10 million in Bitcoin would need at least 1 billion in tier one equity capital, thereby overprotecting against the maximum conceivable risk by a factor of 100. Less extreme, but still asymmetric is the Crypto Asset Reporting Framework, which demands from crypto asset providers to collect and report more data on its clients for tax purposes than the equivalent standard demands for traditional assets (OECD, 2022). A third example are exchange transactions in Switzerland, where crypto currencies are subject to tighter limits than traditional currencies (Roussel, 2023).

Despite these developments, I remain hopeful that the staking controversy can be resolved for the better of all involved parties, preserving the possibility to legally hold staked assets on someones behalf.[3]

1.3.3 Decentralized Stablecoins

Decentralized stablecoins like the Frankencoin raise a number of interesting legal questions, as smart contracts do in general. See for example Raskin (2016) and

[3] On December 12th 2023, FINMA held a staking roundtable at which the author was present on behalf of Swiss Blockchain Federation and on December 20th 2023, FINMA published FINMA Guidance 08/2023 on Staking, specifying conditions under which FINMA considers staked assets of clients to be separable in bankruptcy and therefore possible without banking license. This is a step forward, but leaves the legal classification of staked assets open in cases that remain unspecified in the guidance.

Dell'Erba (2018), or also Weber (2017) and Essebier and Wyss (2017) for a Swiss perspectives. In here, I will only superficially raise the most important aspects and provide some hints at potential problems under Swiss law. These aspects are:

1. **The legal classification of the Frankencoin (ZCHF) stablecoin**: stablecoins that are pegged to fiat currencies are generally considered payment tokens, even if they represent a negotiable instrument backed by an issuer, which could make them formally look like securities. The Frankencoin does not have this issue as there is no identifiable issuer towards which a Frankencoin represents a claim. Given its purpose, it would most likely be classified as payment token under the classification system of Finma (2018). Finma was the first regulator to adopt the distinction between payment token, utility token, and asset token. This was later also adopted by other regulators in Switzerland and internationally, although the definition might vary. For example, in Switzerland, a token that might be considered a utility token from the perspective of financial market law can still be considered a payment token from a taxation perspective, as the definitions of Finma and ESTV exhibit subtle differences (ESTV, 2019). Generally, financial authorities tend to classify tokens as payment tokens in order to be able to apply the anti-money laundering regulations designed for payments, one example of such a regulation being the travel rule. At the same time, there is a substantial volume of regulation designed to protect investors that only applies to securities but not to payment instruments. That is probably why Finma (2018) explicitly expresses the possibility of hybrid tokens, that are payment tokens and security tokens at the same time. An example of such a hybrid token is the initial version of the CryptoFranc, which was legally issued as a bond while clearly also serving as a payment token. In its current version, it has been recognized as a pure payment token by the Swiss authorities, resolving some of the legal uncertainty described by Dell'Erba (2019) for stablecoins in general.

2. **Presence of lending**: Lending is a highly regulated activity that falls under anti-money laundering and consumer protection laws. However, when a minter obtains Frankencoins against a collateral, there is no identifiable lender as the Frankencoins are freshly minted. Therefore, the central element of lending, namely the presence of a lender and the transfer of a monetary value from the lender to the borrower, is missing. The preliminary conclusion therefore is that it is likely that no lending within the meaning of Swiss AML laws is

taking place in the Frankencoin system.

3. **Issuance of a means of payment**: The issuance of a means of payment is a regulated activity that requires issuers to identify their counterparty when the means of payment is issued or redeemed. Based on this rule, issuers of payment tokens must identify all investors with the help of a regulated financial intermediary or be a registered as a financial intermediary themselves. In the case of the Frankencoin, it is unclear who the issuer is. Is it the whole system that issues Frankencoins to the individual minters? Or are the minters the issuers as they are the ones that technically create new Frankencoins? If the minters are the issuers, who are they issuing the Frankencoins to? Are they are issuing the Frankencoins to the first buyer or to themselves with the Frankencoin system resembling a self-service money printing facility? Depending on how these questions are answered, the regulatory consequences might differ.

4. **The legal classification of the Frankencoin Pool Share (FPS) governance token**: lacking an identifiable issuer, governance tokens of decentralized protocols typically do not have the formal characteristics of securities and are often carefully designed not to legally appear like securities. But in practice, they often serve an investment purpose and could therefore be classified as securities under the Howey test in the US (Henderson and Raskin, 2019). Even though Frankencoin Pool Shares (FPS) do not fulfill the formal requirements of a security, they functionally serve the same purpose, putting them at risk of being classified as a security token under a *function over form* approach. The main legal challenge with tokens is that they often serve an investment and fundraising purpose, but lack the formal features of traditional security as they do not represent a claim or membership right. While the US congress has not enacted specific regulation yet and the SEC tends to take a binary view on the classification of tokens, a recent New York ruling took a more differentiated view, saying that the same token can be a security or not, depending on the contractual circumstances of a transaction (Michaels, 2023). In contrast to the US, it is the parliament has drives the regulatory process in the European Union. Here, the European Parliament has agreed on a regulation titled *Markets in Crypto-Assets* (MICA) that specifically regulates tokens as a new form of financial instrument distinct from traditional securities (Zetzsche et al., 2021). It is likely that FPS pool shares would fall

under MICA in the European Union, whereas its classification under US and Swiss law is uncertain. Given its functional similarities to the MKR and the UNI token that are recognized as not being securities, it is possible than that the FPS is neither.

5. **Liability of governance token holders**: Often, decentralized protocols are de facto under the control of a small group of persons that effectively govern the protocol, which could potentially make them liable. FATF (2023) describes a concrete such case and lists some criteria for determining whether a decentralized protocol should be considered to be under the control of a person or group of persons. The Frankencoin has no administrative keys or other mechanism that would give a specific person special powers, but if there is a de facto concentration of control among a small group of persons that actively participate in the system, a regulator might consider them responsible. Here, it helps to have a system that does not require any active participation as long as everything goes well. In direct comparison, Frankencoin's governance is less centralized than that of many other systems, but offers more possibilities for influence than governance-free systems like that presented by Lauko and Pardoe (2021) .

The last question is probably the one that leaves the largest room for interpretation. To a large degree, decentralized protocols rely on the normative power of the factual. They create new factual circumstances that are non-trivial to reconcile with established legal principles. This also leaves a lot of freedom to the judiciary, allowing the judges responsible for concrete cases to effectively shape the law by filling gaps or applying their own moral judgement as outlined by Meier-Hayoz (1981). As judges often consult not only the law and its history, but also academic sources when confronted with new circumstances, legal scholars can de facto help in shaping the rules applicable to decentralized finance. When doing so, they will likely have their ethical judgement influence their conclusions. And this judgement is more likely to be positive if the crypto sector as a whole delivers value to society through meaningful innovation and by showing that there are better ways to organize the financial system.

Unfortunately, there is an additional set of institutions that exerts increasing influence in the shaping of new regulations through *soft law*, namely international organizations such as the Financial Action Task Force (FATF), the Bank of International Settlement (BIS), or the Organisation for Economic Co-operation and

Development (OECD). Hayes (2012) identifies the following risk with these international organizations: “The workings of the intergovernmental bodies that developed and implemented these rules are largely shielded from public scrutiny; the ‘international community’ has accepted the rules uncritically while failing to subject the bodies that created them to meaningful scrutiny or democratic control.” Effectively, these processes can subvert one of the cornerstones of modern democracies, namely a clean separation of powers. Most of the time, the member countries of international regulatory bodies are represented by experts working within the authorities that later have to apply the created regulation. This makes sense insofar as these experts are the the ones that are close to the regulated subject matter within the executive branch of the government. But at the same time, this setting denies the electorate and the public the traditional means of democratically contributing to the creation of new laws. I personally believe that this is an unhealthy development. Legislators all over the world should try to take back control by insisting on elected members of parliament being sent to represent their country in international organizations that are tasked with the design of new laws. Only the specification of low-level technicalities should be left to the unelected experts of the executive branch.

In the nomenclature of economists, the problem with having regulators essentially designing their own laws through international organizations is not only an inherent conflict of interest and an erosion of the separation of powers, but also the risk that regulators might have different preferences than the rest of society. Overall, regulators tend to be more risk averse and tend to have little sense for utilitarian trade-offs. Take the hypothetical example of Alice and Bob, with Alice doing five good deeds and a bad one, whereas Bob sits on his couch all day watching TV. From an economic and utilitarian perspective, Alice is the better person as she created a net benefit for society. But from a regulatory perspective, Bob is the better person as he did nothing wrong. Given this thought experiment, it is doubtful whether it is a good idea to put the regulators in a position where they can shape the legal foundations that in theory ought to represent an aggregation of everyone’s preferences and not only those of a risk-averse minority.

Overall, I see a great opportunity in decentralized finance and decentralized stablecoins. They have the potential to become an essential building block of a more free and open financial system. In an ideal world, regulators should adopt a similar strategy as the Clinton administration did at the advent of the Internet in the 1990ies. Instead of subjecting market participants to new restrictions out of fear of risks, the US followed a hands-off strategy to facilitate growth that included measures like a moratorium on taxing Internet commerce (Magaziner, Cutter, and Costa, 1998).

References

Adams, Scott (2013). *How to fail at almost everything and still win big: Kind of the story of my life.* Portfolio.

Aquilina, Matteo, Jon Frost, and Andreas Schrimpf (2023). "Addressing the risks in crypto: laying out the options". In: *BIS Bulletin* 66. Bank for International Settlements.

Arrow, Kenneth J and Gerard Debreu (1954). "Existence of an equilibrium for a competitive economy". In: *Econometrica: Journal of the Econometric Society*, pp. 265–290.

Basel Committee on Banking Supervision (2022). *Prudential treatment of cryptoasset exposures.* URL: bis.org/bcbs/publ/d545.htm.

Browne, Ryan (2023). *Ripple will have spent $200 million fighting SEC lawsuit, CEO says.* URL: cnbc.com/2023/05/08/ripple-will-have-spent-200-million-fighting-sec-lawsuit-ceo-says.html.

Buterin, Vitalik et al. (2014). *Ethereum White Paper: A next-generation smart contract and decentralized application platform.* URL: finpedia.vn/wp-content/uploads/2022/02/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf.

Chen, Xi et al. (2009). "Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities". In: *2009 50th Annual IEEE Symposium on Foundations of Computer Science.* IEEE, pp. 273–282.

Crone, Hans Caspar von der, Martin Monsch, and Luzius Meisser (2019). "Aktien-Token: Eine privatrechtliche Analyse der Möglichkeit des Gebrauchs von DLT-Systemen zur Abbildung und Übertragung von Aktien". In: *GesKR* 1.

Dell'Erba, Marco (2018). "Demystifying Technology. Do smart contracts require a new legal framework? Regulatory fragmentation, self-regulation, public regulation". In: URL: ssrn.com/abstract=3228445.

– (2019). "Stablecoins in cryptoeconomics from initial coin offerings to central bank digital currencies". In: *NYUJ Legis. & Pub. Pol'y* 22, p. 1.

Essebier, Jana and Dominic A Wyss (2017). "Von der blockchain zu smart contracts". In: *Jusletter* 24.

ESTV (2019). *2.7.3 Leistungen im Zusammenhang mit Blockchain- und Distributed Ledger-Technologie.* URL: gate.estv.admin.ch/mwst-webpublikationen/public/pages/taxInfos/cipherDisplay.xhtml?publicationId=1003047&componentId=1479003.

FATF (2023). *Targeted Update on Implementation of the FATF Standards on Virtual Assets and Virtual Asset Service Providers.* URL: fatf-gafi.org/content/dam/fatf-gafi/guidance/June2023-Targeted-Update-VA-VASP.pdf.coredownload.inline.pdf.

Feldman, Allan M (1973). "Bilateral trading processes, pairwise optimality, and Pareto optimality". In: *The Review of Economic Studies*, pp. 463–473.

Feynman, Richard (1988). *Richard Feynman's blackboard at time of his death.* California Institute of Technology. URL: digital.archives.caltech.edu/collections/Photographs/1.10-29.

Finews (Oct. 2018). "Und schon geht der zweite Krypto-Franken an den Start". In: URL: finews.ch/news/finanzplatz/33868-crypto-franc-xchf-bitcoin-suisse-armin-schmid.

Finma (2018). *Guidelines for enquiries regarding the regulatory framework for initial coin offerings (ICOs).* URL: finma.ch/en/~/media/finma/dokumente/dokumentencenter/myfinma/1bewilligung/fintech/wegleitung-ico.pdf.

Hayek, Friedrich August (1945). "The Use of Knowledge in Society". In: *The American Economic Review* 35.4.

Hayes, Ben (2012). *Counter-terrorism, policy laundering and the FATF: Legalising surveillance, regulating civil society.* URL: https://www.statewatch.org/media/documents/analyses/no-171-fafp-report.pdf.

Henderson, M Todd and Max Raskin (2019). "A regulatory classification of digital assets: toward an operational Howey test for cryptocurrencies, ICOs, and other digital assets". In: *Columbia Business Law Review*, p. 443.

Kuhn, Hans et al. (2019). "Wertrechte als Rechtsrahmen für die Token-Wirtschaft". In: *Justletter IT.* URL: jusletter-it.weblaw.ch/issues/2019/23-Mai-2019/wertrechte-als-recht_703eae33f1.html.

Landesverwaltung (2019). *Gesetz über Token und VT-Dienstleister (Token- und VT-Dienstleister-Gesetz; TVTG).* URL: gesetze.li/konso/2019301000.

Lauko, Robert and Richard Pardoe (2021). *Liquity: Decentralized Borrowing Protocol.* URL: docsend.com/view/bwiczmy.

Magaziner, Ira C., Ann Grier Cutter, and Len A. Costa (1998). “The Framework for Global Electronic Commerce: A Policy Perspective”. In: *Journal of International Affairs* 51.2, pp. 527–538. ISSN: 0022197X.

Meier-Hayoz, Arthur (1951). *Der Richter als Gesetzgeber: eine Besinnung auf die von den Gerichten befolgten Verfahrensgrundsätze im Bereiche der freien richterlichen Rechtsfindung gemäss Art. 1 Abs. 2 des schweizerischen Zivilgesetzbuches.* Juris-Verlag.

– (1981). “Strategische und taktische Aspekte der Fortbildung des Rechts: Zur Frage nach den Grenzen richterlicher Rechtssetzung”. In: *Juristenzeitung* 36.13, pp. 417–423.

Meisser, Luzius (Aug. 2016a). “Die Blockchain als Standortvorteil”. In: URL: fuw.ch/article/die-blockchain-als-standortvorteil.

– (Aug. 2016b). “Eine Chance für den Finanzplatz”. In: URL: nzz.ch/meinung/kommentare/ld.118851.

– (2016c). “Seemingly Equivalent Firm Decision Heuristics”. In: CEF 2016 - Computing in Economics and Finance.

– (2017). “The Code is the Model”. In: *International Journal of Microsimulation* 10.3, pp. 184–201.

Meisser, Luzius and Gabriela Hauser (2018). “Eigenschaften der Kryptowährung Bitcoin”. In: *Digma* 1, pp. 6–12.

Meisser, Luzius and Carl Friedrich Kreuser (2017). “An Agent-Based Simulation of the Stolper–Samuelson Effect”. In: *Computational Economics* 50, pp. 533–547.

Michaels, David (2023). “Ripple Ruling Deals a Blow to SEC’s Effort to Regulate Crypto”. In: *Wall Street Journal.* URL: wsj.com/articles/ripple-wins-early-dismissal-of-some-claims-in-sec-lawsuit-over-xrp-sales-f88f968f.

Obrecht, Mathias (2023). *Einblicke in die DLT-Praxis der FINMA*. Europa Institut an der UniversitÃ¤t ZÃ¼rich (EIZ).

OECD (2022). *Crypto-Asset Reporting Framework and Amendments to the Common Reporting Standard.* oecd.org/tax/exchange-of-tax-information/crypto-asset-reporting-framework-and-amendments-to-the-common-reporting-standard.htm.

Raskin, Max (2016). “The law and legality of smart contracts”. In: *Georgetown Law Technology Review* 1, p. 305.

Roussel, Alexis (2023). *Finma must be subjected to public oversight.* URL: schweizermonat.ch/finma-must-be-subjected-to-public-oversight.

Swiss Federal Council (2018). "Rechtliche Grundlagen für Distributed Ledger-Technologie und Blockchain in der Schweiz". In: URL: newsd.admin.ch/newsd/message/attachments/55150.pdf.

– (2019a). *Anpassung des Bundesrechts an Entwicklungen der Technik verteilter elektronischer Register.* parlament.ch/de/ratsbetrieb/suche-curia-vista/geschaeft?AffairId=20190074.

– (2019b). "Botschaft zum Bundesgesetz zur Anpassung des Bundesrechts an Entwicklungen der Technik verteilter elektronischer Register". In: URL: newsd.admin.ch/newsd/message/attachments/59301.pdf.

– (Nov. 2020). *Bundesrat setzt DLT-Vorlage teilweise in Kraft.*

– (June 2021). *Der Bundesrat setzt DLT-Gesetz vollständig in Kraft und erlässt Verordnung.*

Tainter, Joseph (1988). *The collapse of complex societies.* Cambridge university press.

Wagner Alexander und Weber, Rolf (2017). "Corporate Governance auf der Blockchain". In: *SZW* 1, pp. 59–70.

Weber, Rolf H. (2017). "Leistungsstörungen und Rechtsdurchsetzung bei Smart Contracts". In: *Jusletter.*

Widmayer, Peter and Thomas Ottmann (1993). *Algorithmen und Datenstrukturen.* Springer.

Zetzsche, Dirk A et al. (2021). "The Markets in Crypto-Assets regulation (MiCA) and the EU digital finance strategy". In: *Capital Markets Law Journal* 16.2, pp. 203–225.

Chapter 2

Frankencoin

JEL Classification Codes: D40, D44, E42, G23, G32

Keywords: digital currencies, stablecoins, auctions, risk management, decentralized systems

Co-authors: Basile Maire (primarily section 2.7), Luzius Meisser

Publication (2022 version): Maire, B., Meisser, L., Frankencoin. *SNB-CIF Conference on Cryptoassets and Financial Innovation*, June 3rd 2022

Abstract

Frankencoin is a decentralized, collateralized stablecoin with a modular architecture. Core contributions are a novel auction-based liquidation mechanism and a mostly interaction-free governance protocol. These two in combination make the Frankencoin independent of external oracles and allow for a high degree of flexibility with regard to the assets used as collateral. Additional equity capital to secure the system is supplied by offering a governance token through the pricing mechanism presented in chapter 3, effectively making the Frankencoin system a fully decentralized *Continuous Capital Corporation.* The long-term fundamental value that mirrors that of the reference currency is obtained via collateralization and trust in the stakeholders to align the return of holding Frankencoins with that of holding Swiss francs. Market forces, rather than an oracle-based conversion mechanism, are hoped to maintain the short-term peg. Finally, we assess the system's risks through the lens of a bank regulator, using historical data. The richly commented and audited Frankencoin

source code is provided as an appendix to complete the specification. The name “Frankencoin" hints at the contemplated reference currency, the Swiss franc, as well as the system’s self-governing nature.

2.1 Introduction

We start from the use-case of a Swiss franc Lombard loan, collateralized with tokens such as crypto-currencies or tokenized securities. The borrower deposits collateral into the system and thereby gains the possibility to mint a token, the “Frankencoin” ZCHF, that is pegged to the Swiss franc. If the value of the borrower’s collateral falls below a certain threshold, the loan can be liquidated and the borrower incurs a haircut. If the value of the collateral falls below the loan value before the liquidation ends, the loss eats into the system’s capital reserve. Implementing this system in a fully decentralized way on the blockchain leads us to our main contributions:

- A modular approach that allows anyone to propose new minting mechanisms and new collateral assets under a novel governance mechanism
- An innovative auction mechanism to enable oracle-free collateralized loans
- Capital requirements inspired by traditional banking rules

In comparison to other decentralized stablecoins, the Frankencoin stands out with its simplicity and versatility, which is achieved by having a basic but extensible setup that relies on the overarching economic incentives of the system participants instead of strictly enforcing a narrow peg with technical means.

2.1.1 Existing Systems

This section describes existing stablecoins and how they solve the challenge of having a stable value compared to a reference currency. In comparison to the Frankencoin, most stablecoins are more directly concerned about steering the short-term peg to the reference currency at small deviations, whereas the Frankencoin aims at creating a credible fundamental value in the long term and relies more on market forces to keep the exchange rate close to the fundamental value in the short term.

Stabilization Methods

The largest two stablecoins, the Tether (USDT) and the USD Coin (USDC) are based on the promise of an issuer to always sell and repurchase them at the price of the reference currency, thereby enforcing a 1:1 peg for as long as the issuer is solvent.[1] A similar stablecoin tracking the Swiss franc is the CryptoFranc (XCHF).[2] In the background, the issuer can apply any stabilization mechanism the law allows. While capital requirements for banks are prohibitively restrictive when it comes to the recognition of cryptocurrencies as collateral (Basel Committee on Banking Supervision, 2022), creators of decentralized stablecoins can be much more creative as their coins usually fall outside the scope of banking regulation, at least for now. This allows them to explore new, innovative ways to stabilize the system. Generally, a fully transparent and well-designed blockchain-based stablecoin is inherently more stable than an opaque issuer-backed stablecoin that relies on traditional banking infrastructure.

An example of a not so well-designed stablecoin is the TerraUSD (UST), belonging to the class of *algorithmic stablecoins*. Before its spectacular collapse in May 2022, it was the third largest stablecoin in circulation (Liu, Makarov, and Schoar, 2023) (Briola et al., 2023). The TerraUSD was built on the Terra blockchain with a built-in oracle at the protocol level. The blockchain validators had to provide quotes for the supported currencies and were punished if they failed to do so or if their quotes were too far from the quotes of the other validators. The system's own token was the Luna, and the stabilization mechanism was based on a mechanism to print Luna tokens to buy TerraUSD in case it fell below the price of the reference currency, and to do the opposite if the price of the TerraUSD was too high. The stability of the TerraUSD rested on the assumption that the fundamental value of the Luna token does not fall too fast when more have to be printed to stabilize the system. Similarly to the currency of a country, this works well for as long as the economy around the Luna token, namely everything that happens on the Luna blockchain, is strong enough to support its value. However, it collapsed quite fast once the market lost trust in its stability. Another similarly designed token, the Iron Coin, faced the same fate. The Iron coin, which was supposed to track the USD dollar, was intended to be stabilized by printing Titanium tokens as needed. Generally, algorithmic stablecoins are considered risky as their value does not rest

[1] tether.to and circle.com/en/usdc

[2] bitcoinsuisse.com/cryptofranc

on tangible collateral (Clements, 2021). Nonetheless, some authors argue that algorithmic stablecoin can be *rational Ponzi schemes* if they can provide sound ways to credibly postpone their collapse indefinitely (Fu et al., 2023).

Besides issuer-backed and algorithmic stablecoins, there is a third class of stablecoins that is based on a blockchain-based collateral. Two of the most popular instances are the DAI and the Liquity USD (LUSD) (Chen, Fogel, and John, 2022; Lauko and Pardoe, 2021). Typically they let anyone deposit a collateral and then mint a certain quantity of the stablecoin. If the value of the collateral falls below a critical threshold, the deposit is liquidated and the proceeds are used to repay the minted coins. This is the principle that we apply to the Frankencoin, with the major difference being that the Frankencoin is much more adaptable to different collateral types, as it introduces a liquidation mechanism that does not rely on the presence of an oracle.

Problems with Oracles

An oracle is a service that records external observations on a blockchain, creating a data feed that is accessible to the smart contracts that reside on this blockchain. Typically, this data consists of prices observed in other places. When a decentralized stablecoins relies on a price oracle, this introduces an external dependency and potentially leads to centralization. For example, when analyzing the currently most popular oracle in the Ethereum system, Chainlink, one finds that its administrators could collude to manipulate prices and potentially exploit this capability to steal billions from decentralized protocols that rely on Chainlink. While the price feeds offered by Chainlink are the median of often more than a dozen independent sources, configuring the feed only requires the signatures of four of its administrators, enabling them to arbitrarily manipulate prices if they wish to do so.[3] But even if the administration was more decentralized, one would still have to trust the independent price sources to not collude. History shows that sometimes even the most reputable institutions cannot resist doing some price manipulation when the incentive to do so is big enough (Wheatley, 2012).

[3] For example, the Chainlink price feed for the ETH / USD price pair found at data.chain.link/ethereum/mainnet/crypto-usd/eth-usd is based on smart contract 0x5f4ec3df9cbd43714fe2740f5e3616155c5b8419. When we first looked into this smart contract in late 2021, only 3 out of 19 administrator signatures were required to configure the feed. After our observation made its way to the prolific crypto influencer Chris Blec (twitter.com/ChrisBlec), who prominently voiced his concerns on Twitter and Coindesk TV, Chainlink adjusted it to 4 out of 9 signatures. This is slightly better, but still a risk.

A more direct type of price feed is based on trades from decentralized exchanges that reside on the same blockchain. Here, an attacker would need to manipulate the prices through actual trades, which can be costly for popular assets traded on liquid markets, but less for illiquid assets. A further challenge is that a price feed chosen today might not be reliable any more in the future. Building a robust system based on the price-feed of a decentralized exchange would require a fallback mechanism in case liquidity on that exchange falls below an acceptable level, with that fallback in practice often boiling down to having a group of administrators that can configure the price source. That is why we came up with a new approach of price discovery that is tailored towards the situation at hand.

Governance

The launch of the Ethereum system made it possible to run programs on a blockchain. These programs are usually referred to as *smart contracts* and one of its first use cases was the creation of *decentralized autonomous organizations* (DAOs), that are governed by unstoppable code instead of human-driven processes (Jentzsch, 2016; Beck, Müller-Bloch, and King, 2018). However, the hacking of the first and at the time largest DAO (simply named *The DAO*) led to a shift away from complex governance mechanisms towards protocols that are as simple as possible and ideally even governance-free (Morrison, Mazey, and Wingreen, 2020; Ellinger et al., 2020).

In practice, this ideal is hard to attain and it is often desirable to have some minimally invasive form of governance to enable the protocol to evolve and adapt in a guided manner. Typically, this is done by the issuance of a transferable governance token that conveys voting rights that can be exercised in a democratic process. This section summarizes the governance process of two successful protocols, Uniswap and Maker, setting a benchmark for Frankencoin's governance.

Uniswap has a governance token called UNI that allows one to take part in a governance process based on majority votes. There are 1 billion UNI tokens in circulation, out of which about 200 million have registered themselves for voting by specifying a delegate. Delegates who command at least 2.5 million votes can create proposals. At the time of the proposal, a snapshot of the voting registry is taken and a voting period of 7 days starts. The proposal passes if it has at least 40 million supporting votes and if enjoys majority approval (Adams et al., 2021).

Out of the 41 proposals made so far, 28 have passed, 2 have failed, and 11 have been withdrawn again by the proposer. The ones that failed did so because they did not reach the 40 million quorum. No one seems to have ever attempted to make a malicious proposal. A potential attacker would have to invest more than 100 million to acquire 40 million UNI tokens, hope that the other tokens holders do not notice the proposal or do not take it seriously enough to bother with voting, and then make it pass in the last minute.

Maker employs a custom voting process they call 'top hat voting'. In this process, new proposals are automatically approved as soon as they received more votes than the previously most popular proposal. Furthermore, the votes of a participant move as they vote for a different proposal than they previously voted for, making it possible for the approval threshold to decline over time (Christensen et al., 2021).

Typically, it takes about 80000 to 90000 votes for a proposal to pass, which is about 9% of the MKR in circulation. That would also mark the amount of tokens needed to launch a successful attack. Potentially, a successful attacker might be able to get control over the system with an investment of roughly 100 million dollars at current market prices, allowing the attacker to mint arbitrary amounts of the DAI stablecoin if the attack is not detected early enough averted by a concerted effort of the major token holders.

Both the Uniswap and the Maker governance seem fit for their purpose. However, both require active participation for passing proposals and de facto control is exerted by only a handful of large token holders (Fritsch, Müller, and Wattenhofer, 2022). Also, the capacity of the system seems limited, as every proposal needs the attention of the major token holders and the potential for parallelization is limited.

2.1.2 Structure of the Chapter

This chapter is structured as follows. In section 2.2, the reader is introduced to the first cornerstone of the contribution, namely a method for oracle-free collateralized minting based on auctions. In Section 2.3, we introduce a reserve pool to address the residual risk of the system not being able to liquidate collateral at a price high enough to repay an open position. This pool is capitalized by the minters as well as voluntary contributors seeking a return on their Frankencoin holdings. At the same time, the reserve pool shares serve as governance tokens for the veto-based governance process described in section 2.4. Next, we turn our attention to how the peg to the Swiss franc is maintained in section 2.5. This completes the overall design of the Frankencoin, allowing us to outline the audited implementation in section 2.6

and concluding with an extensive risk analysis in section 2.7, in which we apply the typical capital requirements found in banking regulation to the system.

2.2 Oracle-free Collateralized Minting

This section specifies and analyzes the proposed oracle-free collateralized minting process from a game-theoretic perspective. There are two relevant games to be analyzed. The first is the initialization game between the minter and the system when proposing to mint new Frankencoins against a collateral. The second is the liquidation game to ensure the liquidation of positions that cease to be well-collateralized. The latter game has four participants: the minter, a challenger, bidders, and the system, whereas the challenger and the bidders are the actors that take the relevant decisions.

For simplicity, market prices are assumed to be constant for the duration of the games. Without loss of generality, the reference currency of the system is assumed to be the Swiss franc and the systems stablecoin is referred to as the Frankencoin or ZCHF. The calibration of the loan-to-value ratio l is subject to a separate analysis. It is not necessary to have a liquid market for the collateral asset. It suffices that some independent potential challengers owning sufficient quantities exist, such that the minter cannot corner the market. Finally, it is assumed that the market is efficient and that all actors are rational.

2.2.1 Definitions

Table 2.1 defines the four actors that play a role in the two games, the minter, the challenger, the bidder and the system. These actors have to choose strategies given the decision variables listed in table 2.2 and the exogenous parameters defined in table 2.3.

Letter	Name	Description
M	minter	Deposits a collateral of C_M and proposes a trigger price p_T.
C	challenger	Challenges the minter offering C_C units of the collateral.
B	bidder	Bids Z_B Frankencoins for C_C units of the collateral asset.
S	system	Vetoes new positions. Absorbs liquidation profits and losses.

Table 2.2: Actors. The four relevant actors in the two discussed games.

Letter	Name	Description
$C_M \in \mathbb{R}_{>0}$	minter collateral	Amount of the collateral provided by the minter.
$p_T \in \mathbb{R}_{>0}$	trigger price	Price level below which the position is liquidated.
$C_C \in (0, C_M]$	challenger collateral	Collateral asset quantity offered by the challenger.
$p_B \in \mathbb{R}_{>0}$	bid	The price the bidder is offering.
$D_S \in noop, veto$	governance	The system's approval decision.

Table 2.4: Strategies. The decision variables in the two discussed games.

Letter	Name	Description
$l \in [0, 1]$	loan-to-value	The usable fraction of the collateral value.
$p \geq 0$	price	The market price of the collateral in Frankencoin.
$k \in [0, 1 - l)$	reward	Fraction of the bid to reward successful challenges.
$v > 0$	value	The utility value the minter derives from a successful minting.

Table 2.6: Parameters. The relevant parameters and their meaning.

2.2.2 Initialization Game

The initialization game is about initiating a new collateralized Frankencoin position.

Specification

Two actors, a minter M and the system S take part in this sequential game with two rounds as depicted in Figure 2.1. In the first round, the minter chooses a

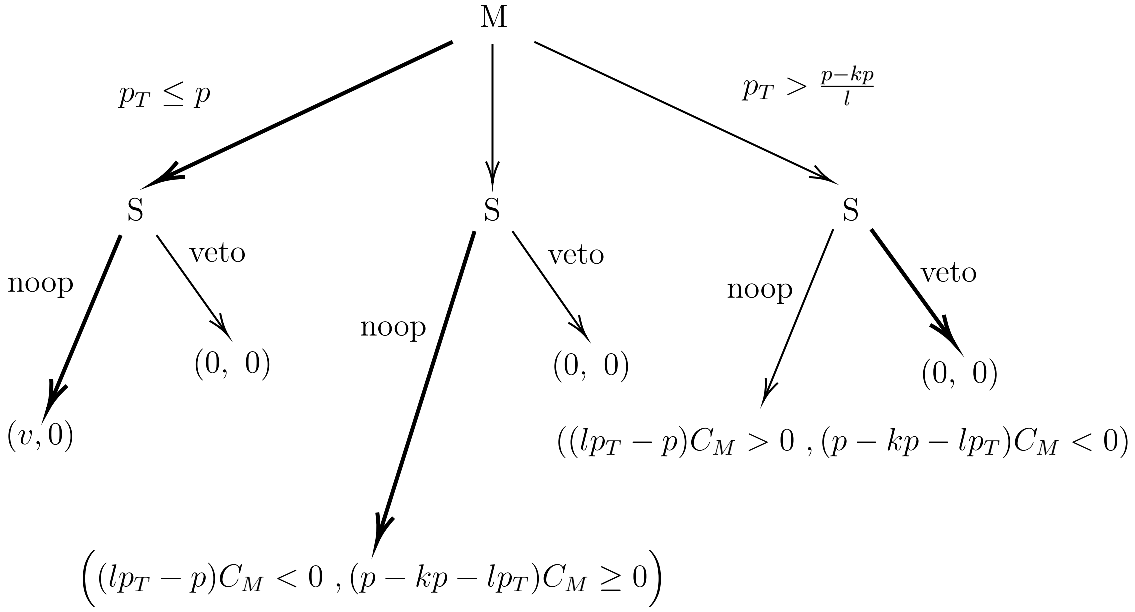

M
$p_T \leq p$
$p_T > \frac{p-kp}{l}$
S
S
S
noop
veto
noop
veto
noop
veto
(0, 0)
(0, 0)
(0, 0)
$(v, 0)$
$((lp_T - p)C_M > 0 \ , (p - kp - lp_T)C_M < 0)$
$\Big((lp_T - p)C_M < 0 \ , (p - kp - lp_T)C_M \geq 0\Big)$

Figure 2.1: Initialization game. Extensive form graph of the initialization game between the minter (M) and the system (S), with the preferred choices in bold. The first element in the round brackets is the minter's payoff and the second element that of the system. A rational minter M goes for the leftmost path, choosing a liquidation trigger p_T below the market price, allowing them to successfully mint some Frankencoins and derive utility v from the obtained liquidity. In the other two cases, the minter is either liquidated in the subsequent liquidation game and suffers a loss, or the position is so undercollateralized that it is immediately vetoed by the system to avert a loss.

strategy $(C_M, p_T) \in \mathbb{R}^2_{>0}$. The variable C_M is the amount of the collateral that the minter deposits, and the variable p_T is the price point at which the liquidation is triggered. The trigger price also determines how many Frankencoins the minter can mint and withdraw, namely up to $Z_M = lp_TC_M$. In the second round, the system chooses a strategy $D_S \in \{noop, veto\}$, whereas *noop* is a common abbreviation for 'no operation' and represents the choice of doing nothing. The other possibility is to *veto* the proposal.

It is assumed that the minter has a desire to borrow some Frankencoins while retaining ownership of the provided collateral. This is represented by a utility of value v in case of a successful initialization of a well-collateralized position. Since the minted Frankencoins and the resulting debt are of equal magnitude, there is no benefit for the minter besides v. A position is considered *well-collateralized* as long as $p_T \leq p$. On the two depicted paths where the position is not well-collateralized, the payoffs are determined by the subsequent liquidation game. The two relevant ranges to consider are the one where the collateral suffices to cover the loss, such that the system benefits from the liquidation, i.e. $p_T \leq \frac{p-kp}{l}$, and the one where the position is so under-collateralized that the liquidation leads to a loss for the system.

Analysis

Theorem 1 (Valid Initialization). *Assuming the liquidation game is successful at liquidating undercollateralized positions and yields the payoffs from table 2.5 (Theorem 2), the initialization game will never end with a position being opened that is not well-collateralized, whereas well-collateralized is defined as $p_T \leq p$.*

Proof. Theorem 1 can be shown by analyzing the initialization game as depicted in Figure 2.1.

Case 1 (left path, $p_T \leq p$). *The collateralization is sufficient to prevent a liquidation. The system S is indifferent between veto or noop, as there is no loss or gain for either decision. The resulting position, if any, is well-collateralized.*

Case 2 (middle path, $p < p_T \leq \frac{p-kp}{l}$). *In this case, the position is not well-collateralized. The system has to choose between vetoing it and doing nothing. Doing nothing leads to the successful initialization of the position and, assuming that Theorem 2 holds, a liquidation with a positive payoff for the system and a negative payoff for the minter. The minter would get away with up to $Z_M = lp_TC_M$ Frankencoins, but would also suffer a loss of $pC_M > Z_M$. This leads to a negative payoff and the conclusion that no rational minter will ever choose p_T in the range $p < p_T \leq \frac{p-kp}{l}$.*

Case 3 (right path, $p_T > \frac{p-kp}{l}$). *In this case, the position is so under-collateralized that a liquidation would yield a negative outcome for the system after auctioning off the collateral and paying the challenger reward. With a veto, the system S can avert that loss, resulting in no position being opened and a payoff of 0 for the minter.*

To conclude, no rational minter will ever propose an position that is not well-collateralized. □

Extensions

In practice, casting a veto comes at a cost for the system as someone needs to invest time into reviewing the proposed position, estimating the market price of the collateral, and potentially casting a veto. To counter this, the actual system imposes an application fee f on the minter when applying for a new position. Furthermore, it is possible for the system to impose an interest $i < v - f$ without deterring the minter from making a proposal. For simplicity, we do not formalize these additional factors.

2.2.3 Liquidation Game

The purpose of the liquidation game is to liquidate undercollateralized positions after the market price has fallen below the trigger price, i.e. $p < p_T$. The payoffs are designed such that the challengers have an incentive to start the challenge only when the position actually is undercollateralized.

The liquidation game consists of two stages: a decision to challenge the collateralization of a position and the subsequent competitive decisions to bid on the challenge. Anyone can start a challenge and thereby become the challenger. Also, anyone can place a bid. In the base scenario, it is assumed that the minter, challenger, bidder and system are independent. In subsequent sections, we show that the game still works as intended if identities overlap, for example when the bidder and the minter are the same person.

Specification

Two actors, the challenger and a group of competitive bidders take part in a sequential game as depicted in Figure 2.2. The minter is not allowed to repay the outstanding balance and reclaim the collateral for as long as a challenge is pending and is therefore not an actor in this game. Neither is the system.

The liquidation game starts with the challenger's decision to start a challenge with quantity $C_C \leq C_M$ of the collateral. Then, it is the bidder's turn to bid for that quantity of the collateral and to choose a bid $Z_B = p_B C_C$, implying the bidding price p_B. In practice, the bidding process is conducted as a Dutch auction. The price starts at p_T for a while and then starts to linearly decline towards 0 until a bidder finds the offer attractive enough. The price point at which this happens is denoted p_B. Given a competitive process and an efficient market, the sale will either happen at p_T or the market price p, i.e. $p_B = min(p, p_T)$.

Challenges that end with $p_B < p_T$ are considered successful. They imply that the position is not well-collateralized. Challenges that end with $p_B = p_T$ are considered averted. In the successful case, it is C_C of the minter's collateral that the bidder is bidding for. In the averted case, the bidder will get the challenger's collateral. The switch in the bidding target is the key element of the Frankencoin auction.

In the averted case, the bidder ends up buying C_C units of the collateral asset from the challenger for $p_B C_C = p_T C_C$, with the other actors being unaffected. In the successful case, the bidder ends up buying the same quantity of the collateral from the minter, with the proceeds going to the system, the minter's associated debt $l p_T C_C$ being erased, and the challenger being rewarded with $k p_B C_C$. These payoffs are shown in Table 2.4. and illustrated in Figure 2.3 for the case of the successful challenge.

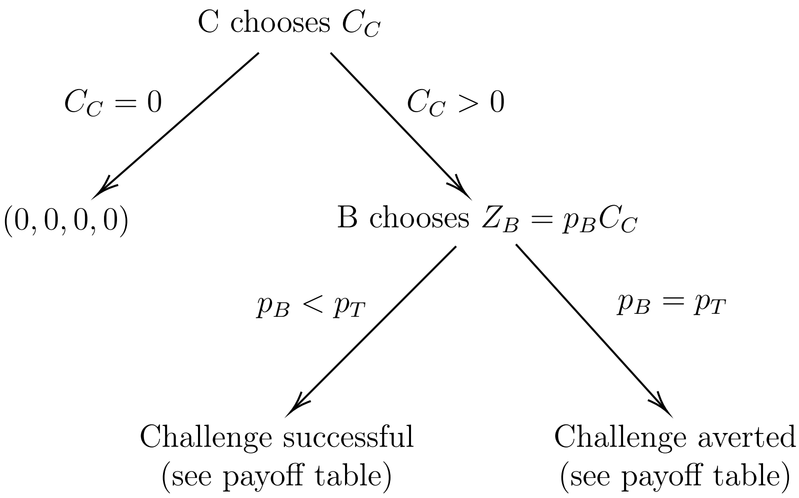


Figure 2.2: Liquidation game. Extensive form graph of the liquidation game. It only makes sense to start a challenge if the highest bid can be expected to imply a violation of the loan-to-value threshold, i.e. if the market price of the collateral is below the trigger price.

Base Scenario

In the base scenario, it is assumed that the minter, challenger, bidder and system are distinct persons.

Theorem 2 (Successful Liquidation). *Given rational actors, a challenge is started if and only if the market price has fallen below the trigger price, i.e. $p < p_T$, and the challenge will end successfully.*

Actor		Challenge successful	Challenge averted
B	bidder	$p - p_B$	$p - p_B$
C	challenger	kp_B	$-p + p_B$
M	minter	$lp_T - p$	0
S	system	$p_B - kp_B - lp_T$	0

Table 2.8: Payoffs. The payoff for each actor in the liquidation game in relative terms. To get the absolute payoff, all values need to be multiplied by the size of the challenge C_C.

Proof. The payoff of the bidder does not depend on whether the challenge is successful or averted, making it perfectly predictable. Being rational, the bidders will never bid higher than the market value, i.e., $p_B \leq p$. Considering that the Dutch auction starts at price p_T, the highest possible bid is $p_B = p_T$. Under perfect competition in an efficient market, the highest bid therefore is:

$$Z_B = p_B C_C = \min(p, p_T) C_C \tag{2.1}$$

Predicting the behavior of the bidder, the challenger immediately start a challenge once the market price has fallen below the trigger price, i.e. $p < p_T$, allowing them to earn the challenger reward $kp_B C_C$. In contrast, they will not start a challenge when $p \geq p_T$ as the payoff would be $p_B - p = p_T - p \leq 0$.

Therefore, given the specified payoffs, rational challengers will start a challenge as soon as $p < p_T$ (but not earlier) and that challenge will be successful. □

Overlapping Identities Scenario

A key assumption of the base scenario was that all actors are independent. However, in practice, the bidder and the minter might be the same person, or there might be side-payments between them to tilt the incentives. This section shows how theorem 2 is affected and how it can be preserved when the bidder does not act on its own. The other three conceivable combinations (i.e. challenger and system as the same person) are less problematic and not further discussed.

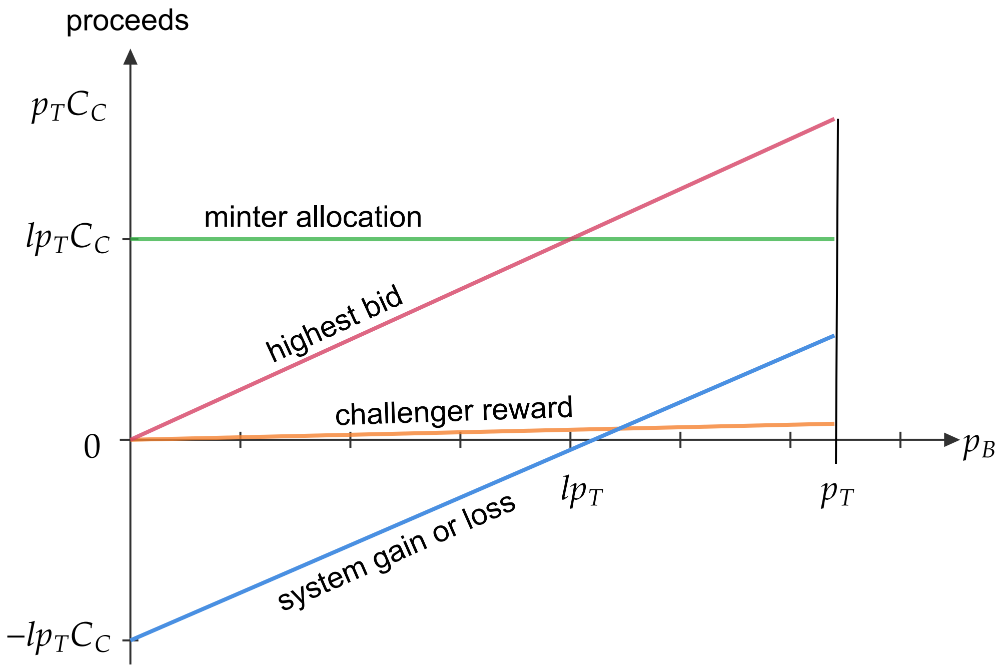


Figure 2.3: Bid allocation after a successful challenge. How the proceeds from the winning bid (red) are allocated to the minter (green), the challenger (orange), and the system (blue), as a function of the bidding price p_B. The four lines sum up to zero. It is pivotal for the system to ensure that challenges happen before the price has fallen too low in order to avoid losses. In the extreme case of the collateral having become worthless, the system has no choice but to forgive the outstanding amount $lp_T C_C$.

Proposition 1 (The challenger as bidder)**.** *Theorem 2 still holds when allowing the challenger to be the bidder.*

This proposition is shown by distinguishing the case of the successful from the case of the averted challenge.

In case of a successful challenge, $p < p_T$ holds by definition. If that was not the case, other bidders would bid $p_B = p_T$ and avert the challenge. When the highest bidder and the challenger are the same person, the cumulative payoff for the successful case is:

$$p - p_B + kp_B > 0$$

In case of an averted challenge, the cumulative payoff of the bidder and the challenger is:

$$p - p_B - p + p_B = 0$$

This implies that a challenger can cancel a challenge at any time without incurring any costs, simply by bidding $p_B = p_T$. It also implies that the challenger has no incentive to artificially avert an otherwise successful challenge.

Furthermore, within the successful challenge case, the challenger has no incentive to make a bid above the market price ($p_C > p_B = p$) as the payoff without interference is larger than the payoff with interference ($kp_B > p - p_C + kp_B$).

Therefore, a rational challenger will not bid any different than any other bidder in case of $p < p_T$, and would not start a challenge otherwise, preserving theorem 2.

Proposition 2 (The minter as bidder)**.** *Theorem 2 is violated when allowing the minter to be the bidder in an isolated game, but still holds when allowing the challenger to repeat the game under the same market price p.*

The minter might want to avert a looming liquidation by placing bids above the market price, i.e. $p_B = p_T > p$. Bidding above the market price to avert a challenge yields a negative payoff of $p - p_B < 0$. However, taking this loss might still be preferable over taking the liquidation loss of $lp_T - p$. So in such a case, a rational minter would bid above the market price, leading to the challenge being averted despite $p < p_T$, violating theorem 2!

To address this problem, one needs to expand the scope from a one-shot game to a repeated game. When the minter averts the challenge by bidding above the market price with $p_B = p_T > p$, the challenger makes a profit of $p_B - p > 0$. If the challenger is given the opportunity to restock the auctioned asset at market price p and then repeat the game before the minter gets a chance to close the open position, theorem 2 still holds. In that case, the challenger can threaten to infinitely repeat the game, causing an infinite loss to the minter. Under this threat, a rational minter reverts to the original bidding strategy under which the ideal bid is the same as for the other bidders.

Proposition 3 (The system as bidder)**.** *Theorem 2 is violated when allowing the system to be the bidder in an isolated game, but still holds when allowing the challenger to repeat the game under the same market price p.*

This is a somewhat theoretic scenario as the system is governed by commonly agreed rules and cannot act arbitrarily. Nonetheless, it is worth analyzing for completeness. To analyze this scenario, we distinguish three cases, one in which the challenge is already averted without the system's intervention, one in which the system overbids to avert a challenge, and one in which the system places a bid such that $p < p_B < p_T$.

In case of the market price being high enough that other bidders already can be predicted to avert the challenge, the incentive of the system acting as bidder are identical to that of the other bidders, as the payoff for the system in that case is zero.

For the other two cases, the cumulative payoff of the system for a successful challenge needs to be considered. It is:

$$p - p_B + p_B - kp_B - lp_T = p - kp_B - lp_T$$

This payoff is identical to the payoff without intervention with $p = p_B$. In the case of a positive payoff, the system has no incentive to deviate from the standard behavior. In the case of a negative payoff, the system could be tempted to bid artificially high like the minter in proposition 2 to avert the challenge and to postpone the looming loss. However, this is not sustainable in a repeated game and even in a one-shot game one could argue that the system did not improve its situation as it still sits on the same undercollateralized loan.

Therefore, the outcome remains the same and the validity of Theorem 2 is preserved.

2.2.4 Example

As an example, let us say the minter opened a position with 10 ABC tokens at a trigger price of $p_T = 100$ as collateral and that $l = 0.8$ and $k = 0.02$. The minter withdraws 800 ZCHF. Then, a challenge is started with 4 ABC tokens, and the highest bid ends up being 360 ZCHF, so the challenge is successful since $360 < 400$. The bidder gets 4 ABC tokens from the minter. The minter's debt is reduced by 320 ZCHF to 480 ZCHF backed by 6 ABC tokens. From the 360 ZCHF bid, 320 ZCHF are burned, $k * 360 = 7.2$ ZCHF are given to the challenger as a reward, and the remaining 32.8 ZCHF are assigned to the reserve as profits for the system.

2.2.5 Minter Reserve

The Frankencoin system has three lines of defense that guard against an undercollateralization of the system. The first line of defense is the overcollateralization of the individual positions as defined by the parameter l. The second line of defense is the equity capital of the system as described in the subsequent chapter. And the third line of defense is the use of other positions in case one position fails through a mechanism denoted as the *minter reserve*. This allows the system to burden the minters with additional debt to bring the books back into balance and should be seen as a measure of last resort once everything else has failed.

To feed the minter reserve, the minting mechanism is adjusted and two additional global variables introduced, the actual reserve R and the target reserve $\tilde{R}$. Given a position with C_M and p_T, the minter will still be allowed to mint and withdraw up to $Z_M = lp_TC_M$ Frankencoins, but at the same time an additional $Z_{R,t} = (1-l)p_TC_M$ Frankencoins are minted and added to the reserve on behalf of the minter. When the minter repays the position, the reserve contribution is fully returned again under normal circumstances, but might be lower in stress scenarios.

When adding $Z_{R,t}$ to the reserve at time t, the actual reserve and the target reserve are updated as follows:

$$R_t = R_{t-1} + Z_{R,t}$$
$$\tilde{R}_t = \tilde{R}_{t-1} + Z_{R,t}$$

Including the reserve, the de facto total debt of the minter at time t is $Z_T = Z_M + Z_{R,t}$. In order to close position at time s, the minter must repay:

$$Z'_M = Z_T - \frac{R_s}{\tilde{R}_s} Z_{R,t}$$

As long as $R_s = \tilde{R}_s$, this is identical to the scenario without the minter reserve. However, they might have to repay more in case the system suffered from a loss that could not be covered by equity capital. In case of a loss L_t at time t that cannot be covered by the first two lines of defense, the missing Frankencoins are taken out of the minter reserve, implicitly increasing the amount of Frankencoins every minter needs to return in order to get their collateral back:

$$R_t = R_{t-1} - L_t$$

After a loss has happened, the reserve will be below its target, i.e. $R < \tilde{R}$ and minters will have to return $Z'_M > Z_M$ to close their positions. As long as the system reserve is below its target, all capital flows that normally go into the equity capital of the system are redirected to the minter reserve to replenish it.

Effectively, this also leads to a temporarily higher loan-to-value ratio $\tilde{l}$ when reasoning about the stability of a position:

$$\tilde{l} = (l-1)\frac{R}{\tilde{R}} + 1 \geq l$$

This new loan-to-value ratio $\tilde{l}$ only applies when repaying an outstanding amount. When minting new Frankencoins, the old l is still applicable.

So in a distress scenario, a minter might for example mint 80 ZCHF but would need to repay 81 ZCHF to close the position again with $l = 0.8$ and $\frac{R}{\tilde{R}} = 0.95$. So as long as the minter reserve is not replenished, closing a position will come with an additional implied fee.

This change has no qualitative impact on the initialization and the liquidation game, although it temporarily makes the initialization more costly until the minter reserve is replenished.

2.3 Equity and Pool Shares

The Frankencoin offers different functions that are fulfilled by commercial banks in the off-chain economy. It offers a fungible means of payment and it allows its users to borrow money against a collateral. Furthermore, it can suffer from losses and generate profits. These losses and profits accumulate in a pool that belongs to the holders of Frankencoin Pool Shares (FPS).

Frankencoin Pool Shares are freely transferable tokens that resemble the shares in a company. Formally, they do not fulfill the requirements to be a security. But functionally, they let the holders participate in the system's economic success (or failure). Like traditional shares, they are also the basis for participation in the governance process.

2.3.1 Issuance and Redemption

Issuance and redemption of pool shares are governed by the principles of the *Continuous Capital Corporation* presented in chapter 3. At any time, new investors can contribute additional capital and get Frankencoin Pool Shares in return. And existing shareholders can redeem their shares at the same price as determined by the pricing rule of the continuous capital corporation. To derive this pricing rule, one needs to specify its production function first.

There is extensive literature on the production function of the banking sector (Benston, Hanweck, and Humphrey, 1982; Gilbert, 1984; Clark, 1984). However, it is unclear how to apply the existing insights to the Frankencoin system. While the early research often assumes Cobb-Douglas production, which would make it easy to derive the pricing function of the continuous capital corporation, their insights are often not directly applicable. For example, Bell and Murphy (1968) use a Cobb-Douglas production function, but their measure for the bank's output is the number of accounts, whereas we are looking at how monetary income depends on the employed capital. Therefore, we resort to just assume Cobb-Douglas production for simplicity and then show that the chosen parameters lead to a sensible result given the circumstances.

For the purpose of deriving the pricing rule, we focus on capital as the only input, arriving at a production function of the form:

$$f(K) = AK^{\alpha}$$

The exponent α represents the *capital share* (as opposed to the labor share that is skipped here) in macroeconomic models. It denotes how much of the system's value stems from its capital and typically floats between 20% and 40% for the economy as a whole (Ilo, 2015). The factor A represents the level of productivity or technology, but cancels itself out in the subsequent calculations.

With the economic consequences presented in section 2.3.3 in mind, we have chosen a value of $\alpha = 1/3$ for the Frankencoin system, leading us to a market cap or valuation of:

$$V(K) = \frac{f(K)}{f'(K)} = 3K$$

Given the number of shares in circulation s, the price of one FPS is given by $p(K, s) = \frac{3K}{s}$.

To calculate the number of shares an investor gets by contributing ΔK to the reserve, we calculate the number of shares s_{t+1} after the investment given the number of shares s_t before the investment using the capital-dependent function 3.9 from Chapter 3 for the number of shares $\theta(K)$ as follows:

$$s_{t+1} = \frac{\theta(K + \Delta K)}{\theta(K)} s_t = \frac{f(K + \Delta K)}{f(K_0)} \frac{f(K_0)}{f(K)} s_t = (\frac{K + \Delta K}{K})^{\frac{1}{3}} s_t$$

The advantage of looking at the relative increase of the number of shares s is that this equation is still valid in the presence of other factors that influence K between transactions, namely incoming profits and outgoing losses. In contrast, the static equation for $\theta(K)$ derived in chapter 3 is only generally valid when profits and losses are immediately attributed to the shareholders, like it is done in general equilibrium models, but not in reality.

2.3.2 Frontrunning and Remedy

One of the risks of the continuous capital corporation is that it opens the door for frontrunning at the cost of the shareholders. This risk is particularly accentuated for systems running on a public ledger where imminent flows of capital can be anticipated and acted on before they are processed by the system. For example, if a profitable liquidation of a large position is imminent, it would be profitable to buy some pool shares immediately before the liquidation and redeem them again immediately afterwards in a so-called sandwich attack, allowing the attacker to make a nice return in a short time at the expense of the other shareholders. The Frankencoin system guards against this risk by enforcing a minimal holding period and a transaction fee.

To quantify the profits that can be extracted through such a sandwich trade, consider how the number of shares changes during the attack, with the attacker first investing i, then the system receiving a profit π, and finally the attacker divesting d again:

$$s_{t+2} = (\frac{K + \pi + i - d}{K + \pi + i})^{\frac{1}{3}} s_{t+1} = (\frac{K + \pi + i - d}{K + \pi + i})^{\frac{1}{3}} (\frac{K + i}{K})^{\frac{1}{3}} s_t$$

Here, s_t denotes the number of shares in circulation before the trade, s_{t+1} after investment i, and s_{t+2} after the trade. Assuming that the attacker wishes to do a neutral attack and end up with the same number of shares as they started, $s_{t+2} = s_t$ holds. With this assumption, the above equation can be solved for d in order to quantify the proceeds from the divestment $d(i)$ as a function of the investment i.

The attacker wishes to maximize their profits by choosing i:

$$\max_i d(i) - i = K + \pi - \frac{(K + i + \pi)K}{K + i}$$

The larger the employed capital i, the larger the amount the attacker can extract, with:

$$\lim_{i \to \infty} d(i) - i = \pi$$

With infinite Frankencoins, a frontrunner could potentially reap all the imminent system profits for themselves in a sandwich attack. Such a sandwich attack would consist of a transaction consisting of a large flash loan, buying as many shares as possible, triggering the profitable event (i.e. the end of an auction), selling the shares again, and finally repaying the flash loan. In a scenario in which the Frankencoin system itself supports native flash loans, such loans could potentially be much larger than the amount of ZCHF in circulation.

The Frankencoin system averts such attacks and discourages short-term speculation in general by enforcing a minimal holding period of 90 days. However, this measure on its own still leads to undesired arbitrage opportunities where a minority of active traders can extract a profit at the expense of the passive investors. The associated price dynamics are illustrated in figure 2.4.

Having small-scale arbitrageurs extract small profits from the system whenever there is a small inflow or outflow of equity capital does not seem to add value to the system. Therefore, we seek to prevent it by introducing a price spread of 0.6%, which is equivalent to a 0.3% fee and in line with the standard fees of the Uniswap system. A price spread is a common method to protect against losses when trading with informed traders (Glosten and Milgrom, 1985).

Another way to guard against such attacks would be to smoothen incoming profits and losses. This would make the attack more costly as the attacker would not only need to deploy capital for an atomic instant, but would need to employ it over a longer period of time. However, such a smoothing would cause the pricing function

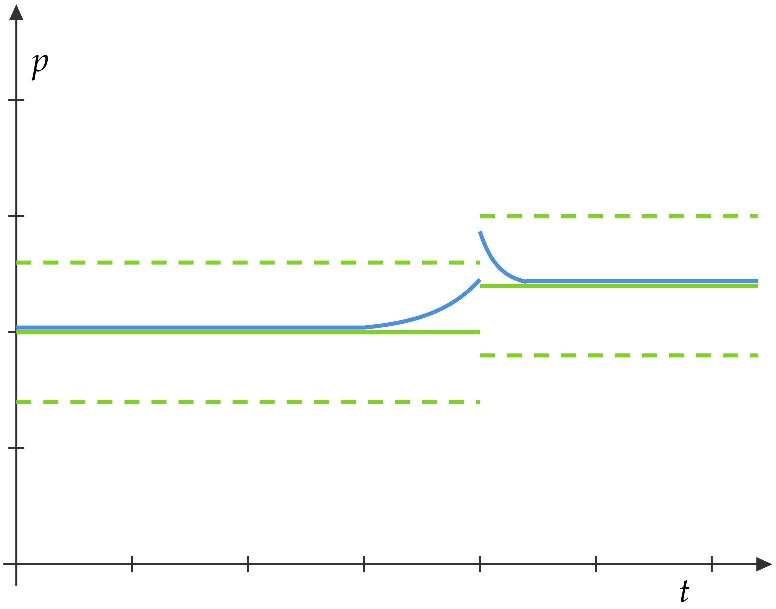


Figure 2.4: Stylized anticipated price change. In the absence of any trading, the inflow of a profit leads to an immediate, proportionate adjustment of the price at which pool shares can be obtained or redeemed (green line). In an active market, traders anticipate this change and start buying before the change, and sell again after the change (blue line), making a profit at the expense of the passive shareholders. Introducing a price spread (dashed green line) prevents intertemporal arbitrage for small price changes that happen at discrete steps.

to exhibit a lag. While the trading fee also leads to a lagged price discovery, the mispricing caused by the fee is limited relative to the price (the 0.3%), whereas the mispricing introduced by profit smoothing is only bound in time and can potentially be much larger as changes accumulate.

2.3.3 Equilibrium Reserves

Assuming similar risk premia for debt in the Frankencoin system and pool shares, the economic equilibrium is reached when about 1/3 of the effectively borrowed Frankencoins are in the reserve pool.

To arrive at this result, we disregard the minters reserve and assume that the interest paid on the effectively usable part D of all open Frankencoin debt positions is r_d, whereas $D = O - R$ is defined as the total amount of Frankencoins minted against a collateral O minus the minters reserve R. The letter O stands for *minter repayment obligation* as depicted in appendix 2.A. Further, let us denote the expected return of holding pool shares as r_e. Assuming an equivalent risk premium, this leads to an arbitrage opportunity whenever $r_d \neq r_e$. If $r_d < r_e$, minters can borrow more and invest it into pool shares, thereby driving up the price. If $r_d > r_e$, investors have an incentive to sell pool shares instead of increasing or renewing their debt positions.

We define r_d to contain not only the nominal interests, but also fees, liquidation proceeds, liquidation losses, and all other income streams directly or indirectly attributable to the minters. Then, one can expect in equilibrium $r_d D = r_e V(K)$ given the aforementioned arbitrage opportunities. The interest paid on the sum of debt positions D corresponds to the return one can expect on investments in pool shares with the market capitalization of the pool shares being $V(K) = 3K$ as defined in section 2.3.1. This implies $K = \frac{1}{3}D$ in the static equilibrium. The effective ratio can vary in times of changing interest rates as there is no incentive to cancel already established positions early and no mechanism to adjust the interest rates of established positions.

This, in turn, also implies that in equilibrium, $\frac{2}{3}D$ ZCHF are used for other purposes, for example as a transactional currency, and that the Frankencoin system does not pay any interest to their holders. The underlying assumption is that there is sufficient demand for a stablecoin as a transactional currency such that the system as a whole can reap some seignorage gains and the early investors in Frankencoin Pool Shares get a return on their capital that significantly exceeds r_e. Considerations for what happens if there is a mismatch between the demand for Frankencoins as a transactional currency and the demand for loans are made in section 2.5.

2.4 Governance

The Frankencoin system is designed for passive, decentralized governance. Under the assumption of perfect information and rational participants, no governance action will ever be necessary. This is achieved by having a passive system based on vetoes that only require an intervention if things go wrong. Having a credible threat of intervention can elicit correct behavior even if the intervention actually never takes place.

Most decentralized autonomous organizations build on a vote-based governance system with some form of majority votes as discussed in 2.1.1. In contrast, the Frankencoin system is based on vetoes. Its speed and the reduced amount of required attention make veto-based governance more agile, faster, and more light-weight in comparison to a full-scale majority vote. However, a veto-based system comes with the disadvantage of a small minority being able to block the whole system. This is averted by time-weighting the votes and the introduction of a 'kamikaze' function that allows an altruistic token holder to reduce the voting power of an attacker.

In the following, we specify the governance process, derive how a smart contract can keep track of the voting weights of each shareholder, and show that the governance of the Frankencoin system yields the same outcome as a majority vote while at the same time requiring fewer interactions than traditional voting schemes.

2.4.1 Specification

The governance system is subject to the following rules:

1. Anyone can make proposals. Making a proposal costs a fee of c_p.
2. Proposals pass after t_p days unless someone vetoes them.
3. Anyone with more than $q = 2\%$ of the total votes V has veto power, i.e. a user with v votes can veto if $v > qV$ holds.
4. The number of votes of a user is calculated by multiplying their Frankencoin Pool Shares with the time they have held them.
5. Users can delegate their votes to other users, who in turn can delegate them further. This allows minority shareholders to team up for a veto.
6. Users can persistently cancel each others votes. For example, Alice can sacrifice 100 votes in order to also reduce Bob's number of votes by 100.

Having time-weighted votes shifts power towards users with a long-term interest in the system. In particular, it guards against malicious actors that take flash loans or use other short-term instruments to temporarily get a sufficiently large number of shares in order to sabotage the system by vetoing good proposals.

Figure 2.6 shows the decision tree with the decision points for the supporters and the opposition of a proposal. It is more deeply analyzed in section 2.4.3. There, we prove that the outcome of this veto-based process is equivalent to that of a majority-based voting process, show how both approaches differ when taking costs into account, and finally argue that the veto based approach is more efficient.

2.4.2 Vote Accumulation

To be able to determine whether a given token holder has more than 2% of all votes, the system does not only need to keep track of each individual user's vote, but also of the total number of votes. Since computations and storage slots on the blockchain are expensive, it is important to keep track of the votes as well as the total in an efficient way. This section specifies how this is done in the Frankencoin system, namely by storing anchor time stamps that are adjusted when tokens are moved. This method requires one additional stored variable per user as well as two global variables to keep track of the total. All operations are performed in constant time, i.e. in O(1) according to the nomenclature of computer science.

The calculation of the votes of a given user is based on that user's balance and a time anchor. Denoting a_s the anchor of a user at discrete time step s and b_s the token balance at the same time, the number of votes at continuous time t can be defined as:

$$v(t) = b_{s(t)}(t - a_{s(t)})$$

Here, $s(t)$ is a function that maps continuous time onto the number of balance changes that have happened at this point in time. For every point in time t at which a balance change happens, s increases by one. Further, let us denote t_s as the time at which balance change s happened and t_{s-} the point in time immediately before that, i.e. $t_s = t_{s-} + \epsilon$ for an arbitrary small $\epsilon > 0$. Consequently, $s(t_s) = s = s(t_{s-}) + 1$.

$$a_s = \begin{cases} a_{s-1}, & \text{for } b_s \leq b_{s-1} \\ t_s - \frac{v(t_{s-})}{b_s} & \text{otherwise} \end{cases}$$

Proposition 4 (Anchor updates). *The anchor update rule exhibits the required properties, namely that votes are adjusted downwards proportionally when the user's balance declines and stay constant when the balance increases.*

Proof. When the balance of a user declines in step s, the number of votes declines proportionally:

$$\frac{v(t_s)}{v(t_{s-})} = \frac{b_s(t_s - a_s)}{b_{s-1}(t_{s-} - a_{s-1})} = \frac{b_s}{b_{s-1}}$$

When a user receives tokens, the votes stay the same:

$$v(t_s) = b_s(t_s - a_s) = b_s(t_s - t_s + \frac{v(t_{s-})}{b_s}) = v(t_{s-})$$

.

□

To keep track of the total votes $V(t)$ in the system, two further variables are needed. These are the total vote anchor A_s and the total vote anchor timestamp T_s. Whenever a number of votes l_s have been discarded in step s, these variables need to be updated as follows:

$$A_s = V(t_{s-}) - l_s$$

$$T_s = t(s)$$

This ensures that the total number of votes can always be calculated as

$$V(t) = A_{s(t)} + B_{s(t)}(t - T_{s(t)})$$

with B_s denoting the total token supply at step s.

Continuously updating the vote counts on each transaction comes at a cost and makes the transfer of pool share tokens about twice as expensive as a plain token. Other governance tokens, for example those of Uniswap or the Maker protocol, address this by only tracking the votes of those tokens that registered themselves for voting. At the same time, they come with more heavy-weight voting mechanisms that rely on snapshots of the token register.

2.4.3 Outcome Equivalence

This section shows that the Frankencoin governance system yields the same outcome as a majority vote with rational participants. Let us denote Y the number of votes in favor of a proposal, and N the votes against (regardless of whether they have been cast or not in an actual voting process).

Our benchmark is a standard majority vote that accepts the proposal when $Y > N$ as depicted in figure 2.5 with payoff $p_Y > 0$ for the supporters and payoff $p_N < 0$ for the opposition, and voting costs $c_v > 0$ per vote.

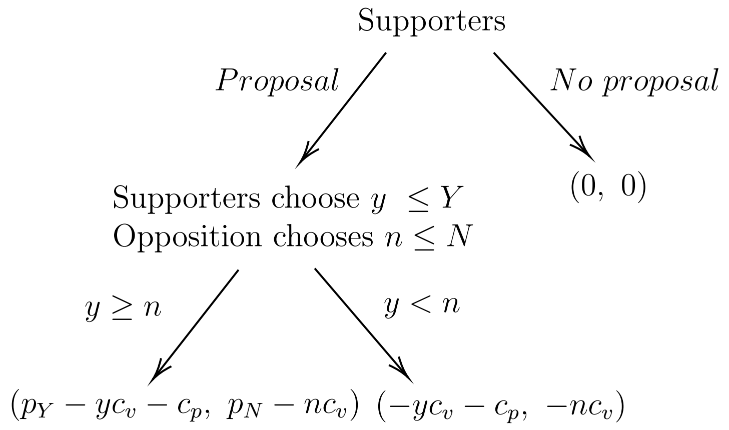

Supporters
Proposal
No proposal
(0, 0)
Supporters choose $y \leq Y$
Opposition chooses $n \leq N$
$y \geq n$
$y < n$
$(p_Y - yc_v - c_p,\ p_N - nc_v)$
$(-yc_v - c_p,\ -nc_v)$

Figure 2.5: Vote game. With profits $p_Y > 0$ for those in favor, loss $p_N < 0$ for those against, voting costs $c_v \geq 0$ per vote, and proposal costs $c_p \geq 0$. Once the supporters decided to make a proposal, the Y supporters and the opposition of size N simultaneously choose how many votes y and n to cast.

An extensive form graph of the Frankencoin's governance process is shown in figure 2.6. In addition to the variables from the benchmark game, there is a value k to denote the number of mutually cancelled votes in case the supporters choose to do so in order to prevent the opposition from vetoing the proposal in future attempts. Also, in this case c_v stands for the costs of casting a veto and not the costs for casting a vote.

Theorem 3 (Outcome Equivalence). *Given rational actors and disregarding participation costs, the supporters of a proposal can get it approved if and only if it would also pass in a simple majority vote, i.e. in both the vote game and the veto game, proposals will get through if and only if* $Y > N$ *when* $c_v = c_p = 0$ *with veto quorum* $q \in (0, 1]$.

Proof. For the base case of the majority vote without costs, supporters will choose $y = Y$, and the opposition will choose $n = N$ and the proposal will pass whenever $Y > N$. What is left to prove is that the same applies to Frankencoin's veto based governance system. This is done by looking at the two cases with and without majority support.

In the case with majority support, i.e. $Y > N$, and a proposal that was vetoed by the opposition, the supporters can mutually cancel N votes and repeat the proposal.

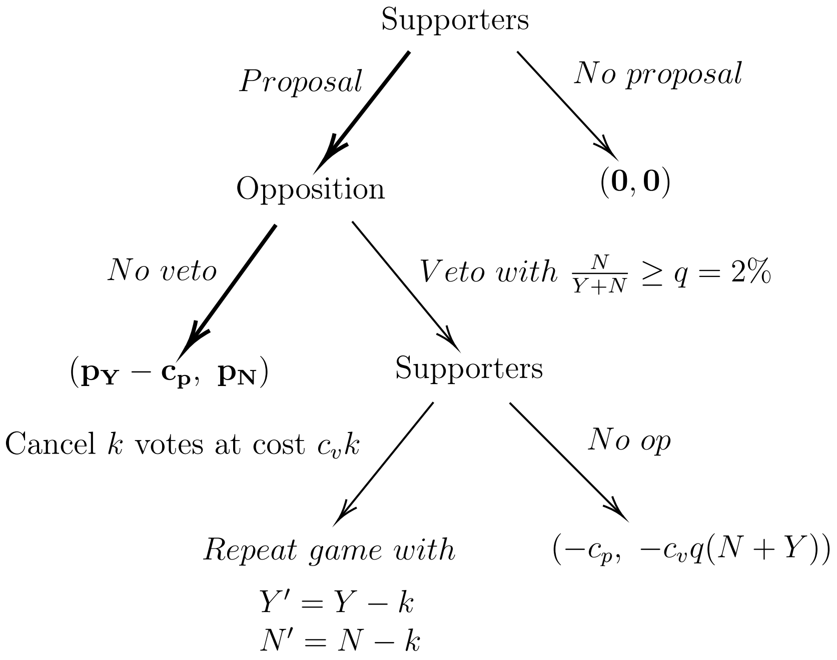

Supporters
Proposal
No proposal
Opposition
$(\mathbf{0}, \mathbf{0})$
No veto
Veto with $\frac{N}{Y+N} \geq q = 2\%$
$(\mathbf{p_Y} - \mathbf{c_p},\ \mathbf{p_N})$
Supporters
Cancel k votes at cost $c_v k$
No op
Repeat game with
$Y' = Y - k$
$N' = N - k$
$(-c_p,\ -c_v q(N+Y))$

Figure 2.6: Veto game. Offers the possibility to mutually reduce voting power and then repeat it.

This time, the opposition is left with $N' = N - N = 0$ votes and the supporters with $Y' = Y - N > 0$ votes, such that $N < q(N + Y)$, which means that the opposition has lost its veto power. The proposal passes.

In the case without majority support, i.e. $Y \leq N$, and a proposal that was vetoed by the opposition, the supporters can choose to mutually cancel up to Y votes. When they repeat the proposal, the opposition can still veto it as $N' = N - Y \geq 0$ and $Y' = Y - Y = 0$, such that $N' \geq q(N' + Y')$ still holds. The proposal does not pass.

Therefore, for both the veto game and the vote game with rational players, the proposal passes whenever $Y > N$. □

2.4.4 Efficiency

This section takes the costs of the individual actions into account and argues without strict proof that passing a good proposal in the vote game is more expensive than in the veto game, making the veto game more efficient.

For the cost analysis, casting a vote or a veto is assumed to be associated with costs $c_v > 0$ per vote cast, so that casting y votes costs $c_v y$ and casting a veto costs

$q(Y+N)c_v$. In line with established research, voter turnout y and n in simultaneous voting games with fully informed participants are assumed to be significant, i.e. in the same order of magnitude as Y or N, even if voting costs are high (Palfrey and Rosenthal, 1983).

Similarly, the costs for the mutual cancellation of k votes is assumed to be $c_v k$. The costs for making a proposal is assumed to be $c_p > 0$. Further, it is assumed that proposals can be repeated arbitrarily often.

Comparing the resulting payoffs for a passing proposal in the vote game $(p_Y - c_p - yc_v, p_N - nc_v)$ with that of the veto game $(p_Y - c_p, p_N)$, it is apparent that the veto game is potentially much more efficient as one can save the efforts of voting.

For these efficiency gains to materialize, it has to be assumed that proposals are mostly uncontroversial. This is the case if participants behave rationally and if there is an effective system in place to ensure that they are well-informed about Y and N. The two benchmark systems, Uniswap and Maker, run discussion forums to create a consensus and perform off-chain consultative votes in order to vet proposals before they are formally voted on in the expensive blockchain-based voting process. Another method to achieve a high degree of information about Y and N is to establish commonly accepted criteria for what makes a good proposal. Given such processes, we expect the Frankencoin's governance system to allow for a much larger number of good proposals passing that come from a broader spectrum of participants than the governance systems of the benchmark protocols described in section 2.1.1.

2.4.5 Attacks

In the context of decentralized autonomous organizations, it has to be assumed that passing a proposal is irreversible and therefore could potentially cause irreparable harm. This section formally shows that in the vote game, it is expensive to defend against such malicious proposals, while in the veto game, it is expensive to approve a good proposal against a malicious opposition. We argue that the latter is to be preferred.

Proposition 5 (Griefing with Votes)**.** [4]. *In a system based on majority votes, it is expensive for the majority to avert harmful proposals that come from a griefer, i.e. a player that is willing to pay c_p in order to cause disproportionate grief to others.*

[4] Griefing is a term borrowed from online gaming and denotes the intentional causing of harm for entertainment (Foo and Koivisto, 2004)

Formally, given the payoffs $(-c_p - c_v n, -c_v y)$ from figure 2.5 for the averted proposal, there exists a strategy for the malicious supporters that forces the opposition to cast $n \geq Y$ votes even though the malicious supporters asymptotically do not need to vote at all, resulting in an asymmetric payoff close to $(-c_p, -c_v Y)$, allowing the supporters to cause disproportionate harm to the opposing majority.

Proof. Given that there is a malicious proposal that would cause a lot of harm, i.e. $-p_N \gg c_v Y$, and that the supporters irrationally made despite knowing that $Y \leq N$, the opposing majority has no choice but to cast at least Y votes to rule out the possibility that the proposal passes.

The decision variable of the supporters is $y \leq Y$, denoting their actual number of votes cast. Likewise, $n \leq N$ is the decision variable of the opposition and denotes the number of votes cast against the proposal. Given that voting is costly, the players prefer not to vote if it does not make a difference. Both decision variables are chosen in a simultaneous game.

Under these circumstances, the supporters can cause disproportionate costs to the majority by adopting the following mixed strategy with an arbitrarily small $\epsilon > 0$:

$$y = \begin{cases} Y & \text{with probability } p = \frac{c_v Y}{-p_N} + \epsilon, \\ 0 & \text{otherwise} \end{cases}$$

The only rational answer for the opposition is to choose $n = Y$, the smallest number of votes that guarantees that the proposal is declined. This leads to the payoff of $-c_v Y$ for the opposing majority, whereas the supporting minority only has costs $c_p + pYc_v$, with

$$\lim_{p_N \to -\infty, \epsilon \to 0} c_p + pYc_v = c_p$$

for proposals that cause a lot of harm p_N to the opposing majority. This results in the asymptotic payoff of $(-c_p, -c_v Y)$ for the supporters (the griefer) and the opposing majority. □

A majority-based system allows a malicious minority to launch griefing attacks in which both parties take a loss, but the loss of the good majority is much bigger than that of the attackers. Decentralized protocols usually discourage this type of griefing by requiring a minimum number of votes to make a proposal, for example 2.5% in the case of Uniswap. With this requirement, the attack requires substantial capital that is at risk of suffering from a capital loss as the value of the governance token can be expected to decline if the protocol is attacked. However, such a requirement also introduces an element of centralization and restricts the ability of making a proposal to only a handful of participants.

Note that voting on a public blockchain is not simultaneous as the votes become publicly visible as they happen. However, the participants could try to cast their votes immediately before the vote closes, making the game effectively simultaneous. Both benchmark systems, Uniswap and Maker, are in principle susceptible to such 'last minute attacks'. As a remedy, they both chose to impose high barriers for proposals to be made and to pass, making the governance process heavy.

Proposition 6 (Griefing with Vetos)**.** *In the veto-based system, it is expensive to push through a majority-supported proposal against a malicious opposition with power $N < 0.5(N + Y)$. More formally, an unknown minority only needs to spend Nc_v on casting vetoes whereas the cost for the supporting majority can grow infinitely high, depending on N.*

Proof. This proposition departs from the fully informed assumption. Otherwise, if the supporters knew the identity of the attackers, they could simply cancel their votes in advance and then pass the proposal at a total cost of $c_p + Nc_v$. However, if the griefers falsely identify themselves as supporters, their identities must be discovered incrementally during the veto game at high costs.

In each repetition of the game, the attackers cast a veto at cost $c_v q(N_i + Y_i)$, with Y_i and N_i denoting the voting power of each group at round i. In each round, the supporters cancel the votes of the newly revealed opposition members at cost $c_v q(N_i + Y_i)$ and repeat the proposal with costs c_p. These costs are determined by how often this attack can be repeated until the attackers do not have enough votes to cast a veto any more.

To formalize how many rounds that takes, consider the total number of votes $T_i = Y_i + N_i$ shrinks by the same factor in each round i, such that $T_i = (1 - 2q)^i T_0$. We are looking for the point in time k at which the opposition can cast the veto for the last time, which is at $N_k = qT_k$, implying $Y_k = (1-q)T_k$. Further using that both parties must have lost the same number of votes in the end, i.e. $N_k - N_0 = Y_k - Y_0$, one can solve for k:

$$k(Y_0, N_0) = \log_{1-q} \frac{Y_0 - N_0}{(Y_0 + N_0)(1 - 2q)}$$

Notably:

$$\lim_{N_0 \to Y_0} k(Y_0, N_0) = \infty$$

So in case of a large unknown opposition that almost has a majority, the griefing potential is unbounded as each of the potentially infinite repetitions comes with the costs c_p for renewing the proposal. □

However, for small N_0, the number of rounds needed is approximately $\frac{N_0}{q}$, which implies that increasing the quorum q needed to cast a veto can reduce the vulnerability of the system. At the same time, having a low q makes it more risky to launch controversial proposals with an unknown opposition, making the system more conservative overall.

2.5 Fundamental Value

We now address the conditions under which the *peg to the Swiss franc* or any other reference currency of choice should hold. The fundamental value of the Frankencoin comes from a combination of three elements: first, it is necessary to have enough collateral such that eventually, each ZCHF can be sold for at least one Swiss franc to a minter who wants to get their collateral back. Second, the interest rate of the system (r_d as defined in section 2.3.3) should be managed such that given a non-zero value of the stablecoin at some point in the future translates into a current value that matches the reference currency. Third, one needs some assurance for the minters to ensure that they cannot be made subject to a short-squeeze. These three factors are addressed in the following three sections.

2.5.1 Sufficient Backing

On the supply side of the Frankencoin, pool share holders should not allow any mintings that create a risk of the value of the supplied collateral falling below the nominal value of the Frankencoins in circulation. Assuming that each Frankencoin is always backed with at least one Swiss franc worth of collateral, the Frankencoin is fundamentally sound and one can assume that the minters are willing to pay at least one Swiss franc per Frankencoin at the point in time when they want to get their collateral back again. This is taken care of by the collateralized minting mechanism discussed in section 2.2.

2.5.2 Comparable Yield

On the demand side, we approach the valuation of the Frankencoin from the perspective of a *perpetual bond*. A perpetual bond, or consol, is a bond with coupon payments but no redemption date (Jorion et al., 2010). We price this perpetual along the lines of (Jarrow and Turnbull, 2000), by discounting the interest payments on a risky term structure. Let's assume that interest payments happen at discrete time-steps $0, ..., \infty$ and we have corresponding risky rates of the term-structure so that the date-0 value of a promised Swiss franc at time t of a risky Franc promise is equal to $\exp(-r_t t)$. Let the constant coupon rate per Frankencoin be c. Now, the value of the perpetual can be written as

$$\begin{aligned} v(0) &= \sum_{t=0}^{\infty} ce^{-r_t t} && (2.2) \\ &= \sum_{t=0}^{\infty} ce^{-yt} && (2.3) \\ &= \frac{c}{1-e^{-y}}, && (2.4) \end{aligned}$$

where the second line replaces the time-specific discount rates by a yield, and the last line is an application of geometric series. For the value to be at par, $v(0) = 1$, we have to choose the coupon rate accordingly: $c = 1 - e^{-y}$. Hence, if the interest earned from contributing ZCHF are in line with discounting, the present value of one ZCHF is equal to one Swiss franc.

The risky term-structure corresponds to the Swiss franc risk-free term-structure plus a spread that compensates the investor for the risks. Whenever the risk-free term-structure, or the Swiss franc risk changes, c has to be adapted for the value $v(0)$ to be equal to one. In the Frankencoin system, the yield is implicitly set by the pool share holders as they can veto minters that do not yield the right risk-adjusted return for the system. In effect, the pool share holders act as an oracle, but for long term interest rates instead of short term prices.

For this to work, minters must be required to pay an interest on their open positions and positions should be limited in time. The shorter the average term of the open positions is, the faster it is possible to push the yield that can be earned from buying pool shares to the correct c. Generally, the system should seek to increase the yield if there is too little demand for holding ZCHF and vice versa.

The comparison with the consol shows nicely that it is not necessary for a stablecoin to be directly convertible to the reference currency in order to have the same fundamental value. It suffices if the coin offers the same expected return. Thereby, the problem is reduced to the ability to pay out an equivalent interest under the assumption that there are market makers that recognize the long-term value and engage in arbitrage trading in order to avert short-term deviations from the peg.

2.5.3 Price Ceiling

While it is generally not possible to exchange Frankencoins directly into collateral provided by the minters, the minters will have to buy back their minted Frankencoins before they can get their collateral back. Here, the minters face the risk of a short squeeze. By minting and selling Frankencoins, they are short ZCHF and might be forced to pay more than one Swiss franc per Frankencoin to unlock their collateral. So while those holding the stablecoin face the risk of it falling below the peg, minters face the risk of the Frankencoin departing upwards from the peg.

In the proposed setup, we start with a very simple mechanism to avert the risk of an overvaluation: we provide a bridge contract that allows holders of other Swiss franc based stablecoins to convert them 1:1 into Frankencoins. As long as such bridges exist, minters can be confident that they do not need to overpay for the unlocking of their collateral. However, while relying on other stablecoins can help in practice, it is not desirable to depend on external systems.

In the absence of stablecoin bridges, minters have to trust the system to always allow the minting of new Frankencoins at competitive terms, such that a short-squeeze can be averted by simply minting additional Frankencoins and repaying the open position with those. In effect, the system relies on good governance on both the supply and demand side. On the supply side, the system must allow economically sensible mint contracts and disallow irresponsible ones. On the demand side, the system must ensure that the risk-adjusted interest rate tracks that of the Swiss franc.

2.5.4 Long-term Incentives

Like many blockchain-based systems, the Frankencoin relies on the rationality of its participants. There is no technical barrier for the participants to collectively mismanage the Frankencoin, for example by letting a buggy minter contract pass. It is in the economic interest of all involved parties to look after the system and to make it work as expected. In particular, pool share holders have to be aware that they would destroy the long-term value of the system by letting the stablecoin deport too far from the peg. Since their reserve is at stake and they are the last who could cash out in case of a crash due to a loss of trust, pool share holders have a strong incentive to not let the price fall. At the same time, they also have a strong incentive to ensure is a wide range of available mint contracts such that the risk of a short-squeeze for minters is under control.

The reason why the system has such a strong interest in making sure the Frankencoin tracks the value of the Swiss franc is that it enables them to earn net interest income to the extent that the Frankencoin is used as a transactional currency. For example, if the open market interest rate is 3% without any margin between borrowers and lenders, the equity holders collectively earn around 9% on the deployed capital in equilibrium as outlined in section 2.3.3. This is only possible if the Frankencoin is valuable enough as a transactional currency such that two thirds of the holders do not care about missing out on potential interest. And the transactional value is highest when the Frankencoin reliably tracks the value of the reference currency.

2.6 Implementation

This section briefly touches on the software architecture of the Frankencoin as shown in figure 2.7. In line with the insights presented in *The Code is the Model*, the source code should be considered the ultimate specification of the system, with the text and mathematical parts of this chapter serving to document, analyze, and to generally present the system in a commonly accessible way (Meisser, 2017).

The richly commented source code can be found in the end of the chapter starting with appendix 2.D and ending with appendix 2.J. The Frankencoin smart contracts are written in Solidity, a programming language developed for the Ethereum system

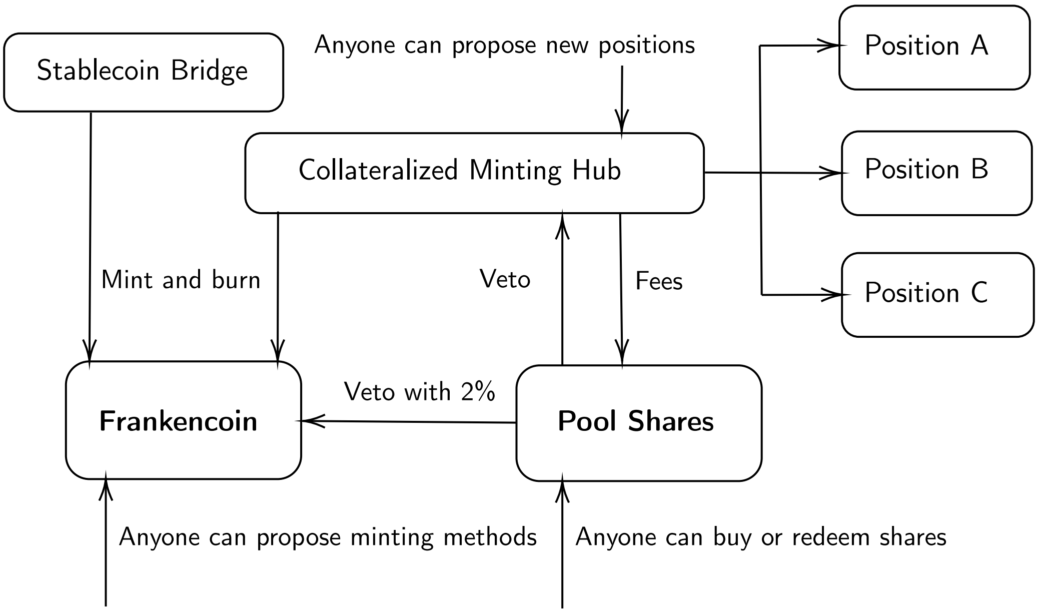

Stablecoin Bridge
Anyone can propose new positions
Position A
Collateralized Minting Hub
Position B
Mint and burn
Veto
Fees
Position C
Frankencoin
Veto with 2%
Pool Shares
Anyone can propose minting methods
Anyone can buy or redeem shares

Figure 2.7: Architecture. There are two ERC-20 tokens, the Frankencoin (ZCHF) and the Frankencoin Pool Shares (FPS), both with extra functionality to allow new minters to be proposed subject to the veto-based governance process. The diagram shows two types of minters, the stablecoin bridge and the minting hub. The latter contains all functionality needed to initiate new collateralized minting positions.

and also supported by other blockchains that are compatible with the Ethereum Virtual Machine (EVM). The code listed in the appendices does not include standard contracts such as ERC-20 and Ownable (Vogelsteller and Buterin, 2015). Furthermore, interface declarations are excluded to avoid redundancy. The full source code can be found on github.com/Frankencoin-ZCHF/FrankenCoin.

The source code underwent three security audits over the course of the year, by Scherer (2023), code4rena (2023), and Mackinga, Ulqinaku, et al. (2023). Besides uncovering a number of technical vulnerabilities, the audits led us to change the auction process from a traditional auction with competing bids to a Dutch auction. In a Dutch auction, the price starts high and is continuously reduced until a buyer appears. This type of auction is less susceptible to last-minute manipulations and requires less information to be stored on chain.

2.7 Risk Analysis

We start by analyzing the risk of a capital loss section 2.7.1, leading us to a discussion of meaningful capital requirements in section 2.7.2, and concluding with a proposal for risk-based capital requirements inspired by Basel iii rules in section 2.7.3.

We find that with the chosen parameter $h = 0.1$, implying a loan-to-value ratio of $l = \frac{1}{1+h} \approx 0.91$, using Bitcoin price data and an auction duration of 24 hours, the system earns a significant net profit from the simulated liquidations and therefore should not need additional risk compensation in the form of an increased interest or otherwise higher fees. These results serve as a foundation to calibrate risk-based capital requirements, where we find when applying Basel iii rules in a technology-neutral way, equity ratios in the low single digit percentage point range would suffice and note that the Frankencoin system has a generous risk buffer in its current design.

2.7.1 Risk from Collateralized Minting

The Frankencoin system is at risk of losing equity capital if a liquidation ends at a price that does not suffice to cover the outstanding stablecoin balance and the challenger reward. To evaluate that risk, we abandon the earlier assumption that the price is constant during the auction. It is also noted that a rational challenger will not immediately start a challenge when the market price p hits the liquidation price p_T, but at a price level $p_0 < p_T$ slightly below the trigger price that depends on the opportunity costs of having the collateral locked up and the probability of the challenge succeeding. Based on these considerations, simulations are performed to evaluate the risk of a loss depending on the price level at which the challenge was started using historical Bitcoin price data.

In deviation from the previous section 2.2.3, the bidding process is assumed to be a normal auction that runs for a fixed amount of time τ or ends immediately when the highest bid reaches the liquidation price, i.e. $p_B = p_T$. Also, the challenger reward is not based on the bid as before (i.e. kp_BC_C), but based on the minted amount of Frankencoins (i.e. klp_TC_C). These deviations should not make a notable difference in the simulation results, so it was decided against redoing them after the changes were introduced. As before, we note that the rational bids imply prices matching (exogenous) market prices. We do not account for slippage incurred when selling the collateral in exogenous markets so that we can use Bitcoin market prices for the exercise at hand.

We define p_0 as the exogenous market price at the time the challenge is started and further define $h = \frac{1}{l} - 1$ and $h' \leq h$ the level of overcollateralization (or undercollateralization if $h' < 0$) at the point in time when the challenge is initiated. Consequently:

$$p_0 = (1 + h')lp_T \tag{2.5}$$

From Section 2.2.3 we have the *condition for liquidation*:

$$p_B < (1 + h)lp_T, \tag{2.6}$$

to which we add the time dimension now. We consider the asset return $\tilde{r}_t$ over the liquidation horizon τ. The challenge is averted earlier after t' if during the liquidation period a bid at the liquidation threshold is made. We can now express the condition for liquidation as a function of the asset return, and the values h and h':

$$e^{\tilde{r}_t} < \frac{1 + h}{1 + h'} \quad \forall 0 \leq t \leq \tau. \tag{2.7}$$

This is another way of expressing that the challenge is successful if at no point in time t the price reaches p_T.

Proof. By definition, given the return r_τ we can express the price at the end of the auction as $p_\tau = exp(r_\tau)p_0$. Using $p_B = p_\tau$ in the case of the successful auction, one can reformulate inequality 2.6 as follows:

$$p_B = p_\tau = exp(r_\tau)p_0 = exp(r_\tau)(1 + h')lp_T < (1 + h)lp_T$$

which trivially leads to inequality 2.7 by dividing by $(1 + h')lp_T$. □

To avoid the risk of the market price declining below an already made bid, a rational bidder will wait with the placement of the bid until near the end of the auction. We can therefore assume that successful challenges lead to a liquidation at price p_τ even if the market price was temporarily higher during the auction.

The profit $\tilde{P}$ for the system is zero in case the challenge is averted, positive in case of a liquidation at a price $p_\tau \geq (1 + k)lp_T$, negative if the price is below, and zero if the challenge is averted and no liquidation takes place. We define a dummy variable D that is exactly 1 in case a challenge is successful and 0 otherwise:

$$D = \mathbf{1}_{e^{\tilde{r}_t} < \frac{1+h}{1+h'}\ \forall\, 0 \leq t \leq \tau} \tag{2.8}$$

We can now express the profit $\tilde{P}$ per unit of the challenged amount as the following random variable:

$$\tilde{P} = \left[(1+h')e^{\tilde{r}_\tau} - (1+k)\right] D \tag{2.9}$$

Note that if the challenge starts at $h' \geq h$, the indicator function is always $D = 0$ as the challenge is immediately ended again. We now determine the value $\tilde{P}$ to decide whether the system needs to be compensated by a minting fee.

To tackle the valuation of $\tilde{P}$, we resort to the arbitrage free pricing principle, which states that the value of a contingent claim is given by its discounted expected value under the risk-neutral probability measure (Björk, 2009). We assume that the risk-free rate used for discounting is equal to zero, which we consider adequate especially since the period τ is very short. Let $f_\tau(x)$ be the density function for the return distribution over the period τ. Now, we can value $\tilde{P}$ conditional on the starting level of liquidation h' as follows

$$\mathbb{E}^{\mathbb{Q}}_{\tau}\left[\tilde{P}|h'\right] = D \int_{-\infty}^{\log \frac{1+h}{1+h'}} \left[(1+h')e^x - (1+k)\right] f_\tau(x)dx \tag{2.10}$$

where the subscript τ emphasizes that the distribution depends on the liquidation period.

We now discuss h'. If we knew the distribution of starting levels h', we could integrate out h' to arrive at $\mathbb{E}^{\mathbb{Q}}_{\tau}\left[\tilde{P}\right]$. As established, challengers trade off the probability of collecting the reward given they issue a challenge versus the probability of not being able to be front-run by other challengers. To investigate this trade off, we assume that challengers issue a challenge when the collateral value reaches a level for which there is a given probability α that they receive the challenger reward. We then investigate what minting fees each h' (or equivalent, each α) implies.

Equation (2.10) uses the risk-neutral probability measure, often referred to as $\mathbb{Q}$, rather than the objective measure $\mathbb{P}$. In practice, parameters for the measure

$\mathbb{P}$ are extracted directly from market data (e.g., sample volatilities and expected returns), whereas parameters for the measure $\mathbb{Q}$ have to be extracted from option data under the same model assumptions (e.g., option implied volatilities). With risk averse investors, the $\mathbb{Q}$-measure puts more weight on adverse market events, see for example Breeden and Litzenberger (1978), leading to a higher risk-neutral price of $\tilde{P}$, compared to the value obtained when integrating Equation (2.10) under the objective measure. We therefore proceed by calibrating a probability distribution to observed market data and use this as a lower bound for the price of $\tilde{P}$, or, equivalently the minting fee should be at least equal to the negative value of $\tilde{P}$ under the measure $\mathbb{P}$:

$$\Theta_F^{(i)} \geq -\mathbb{E}_\tau^{\mathbb{P}}\left[\tilde{P}\right], \tag{2.11}$$

where we add the negative sign because $\tilde{P}$ is a profit. For brevity, we omit the superscript $\mathbb{P}$ in the sequel.

Challenge success probability

We are now ready to calibrate the parameters. To do so, we evaluate the probability of liquidation for different challenger levels h'. We then locate reasonable challenger levels and investigate the lower bound of the minting fee, $\Theta_F^{(i)}$, for the given level of h'. In Appendix 2.C we describe the BTCCHF candle-data that we use to calibrate the fees. We use 24h candle data (Open, Low, High, Close) and use the bootstrap method introduced by Efron (1992) for estimation.

With candle data, we can readily determine whether for a given period, the challenge started at the beginning of the candle would have been averted, $D = 0$, or succeeded $D = 1$, by

$$\hat{D}_\tau = \mathbf{1}_{\left\{\frac{P_H}{P_O} < \frac{1+h}{1+h'}\right\}}, \tag{2.12}$$

where P_O is the open price, P_H the high price over period τ (without indexing the candle for brevity), which follows directly from definition of D in Equation (2.8). For the bootstrap, we define the following variables. Let N be the number of candle observations, B the number of bootstrap replications, and $\hat{\mathbf{D}}_b = \{\hat{D}_\tau^{(b,1)}, ..., \hat{D}_\tau^{(b,N)}\}$ the b^{th} bootstrap replication for a period-τ dummy that equals one if the challenge was not averted. The bootstrap estimate for the probability of liquidation follows directly from Equation (2.12):

$$\hat{\mathbb{E}}\left[D|h'\right] = \frac{1}{B}\sum_{j=0}^{B-1}\frac{1}{N}\sum_{n=1}^{N}\hat{D}_{\tau}^{(j,n)}. \tag{2.13}$$

For comparison, we also evaluate the probability of liquidation conditional on h', assuming the challenge cannot be averted prior to time τ. For this assumption we use both, a bootstrap evaluation (replacing P_H/P_O by P_C/P_O, with P_C equal to the close price, in Equation (2.12)) and a closed-form evaluation assuming the returns are normally distributed:

$$\hat{\mathbb{E}}\left[\mathbf{1}_{\{\tilde{r}_\tau<\log\frac{1+h}{1+h'}\}}\middle|h'\right] = \Phi\left(\frac{\log(1+h)-\log(1+h')-\mu_\tau}{\sigma_\tau}\right), \tag{2.14}$$

where μ_τ and σ_τ are location and scale parameters of the normal distribution, and $\Phi\left(\cdot\right)$ represents the standard normal cumulative distribution function. We use the sample mean and standard deviation of the observed returns to estimate μ_τ and σ_τ respectively. Figure 2.8 shows the results. We see that allowing the challenge to be averted early significantly reduces the probability that a challenge is successful when it is issued at a price close to the liquidation threshold. At $(1+h')$ about 2% below the liquidation threshold $(1+h)$, there is a fifty percent chance of the challenge being successful.

The normal distribution overestimates the probability of liquidation for values h' above the liquidation threshold h, and underestimates the liquidation probability for value of $h' < h$ compared to the bootstrap result. We use the variance resulting from the bootstrap replications to estimate confidence intervals.

Minting fee valuation

Finally, we use use Equation (2.9) to arrive at our bootstrap point estimate for the value of the liquidation profit P.

Let $\hat{\mathbf{r}}_b = \{\hat{r}_\tau^{(b,1)}, ..., \hat{r}_\tau^{(b,N)}\}$ the b^{th} bootstrap replication for a period-τ return, now

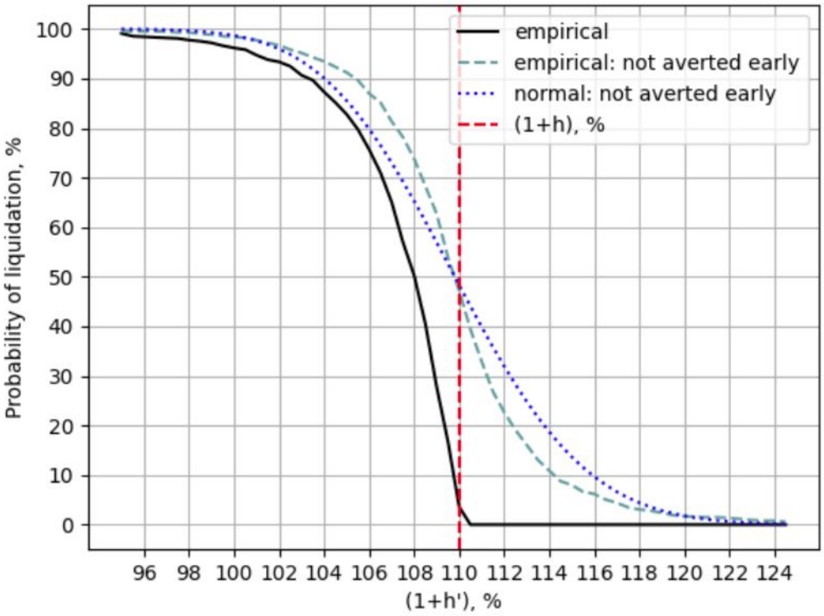


Figure 2.8: Probability of liquidation given h'. This figure plots the probability of the challenge ending in a liquidation, when the challenge is started at a level h', with the liquidation level at $h = 0.10$ and a challenge duration of 24 hours (solid black line). The probabilities are estimated via bootstrap ($B = 5,000$ samples). Confidence intervals at the 1%-level remain below 0.05% around the point estimates for each h'. For comparison, we hypothetically assume the challenges cannot be averted before the end of the liquidation horizon (dashed and dotted curves). We compare the bootstrap estimate (dashed line termed “empirical: not averted early"), with the theoretical probability using normally distributed returns (with sample mean and standard-deviation).

$$\hat{\mathbb{E}}_\tau\left[\tilde{P}|h'\right] = \frac{1}{BN}\sum_{j=0}^{B-1}\sum_{n=1}^{N}\hat{D}_\tau^{(j,n)}\left[(1+h')e^{\hat{r}_\tau^{(j,n)}} - (1+k)\right]. \tag{2.15}$$

We use $B = 10,000$ bootstrap replications and calculate confidence intervals at the 1%-level by applying the central limit theorem.[5] The confidence interval result in ranges in the 0.01%-area. Figure 2.9 shows the results. At low challenge levels, the contributors have to burn ZCHF and there is a loss for them. At very high challenge levels, often the challenge does not result in a liquidation and hence there is no gain or loss for contributors. The system makes the most profit for challenges that start at a level of about $1 + h' = 107\%$.

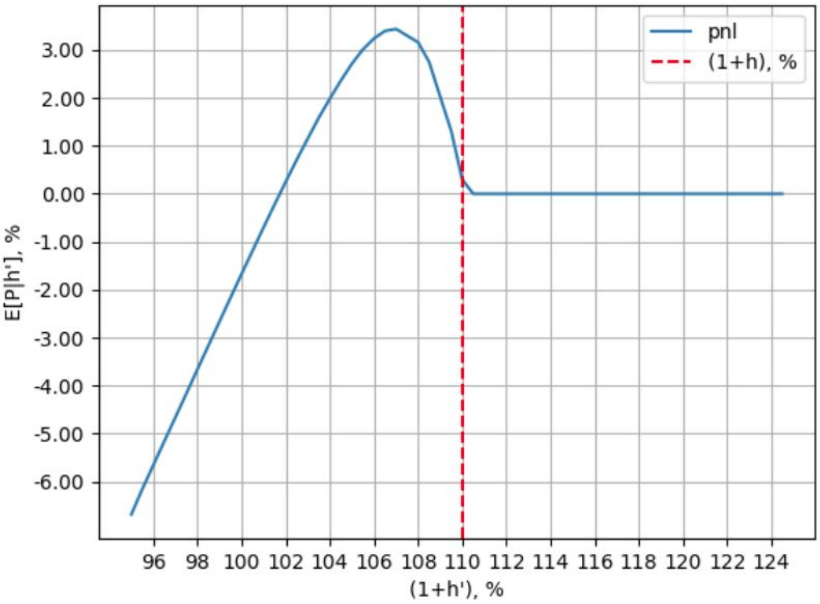


Figure 2.9: Expected system profit: We plot the expected profit (or loss) to the system depending on the level h' at which the challenge was started. Negative values observed for approximately $1+h' < 102\%$ correspond to a loss for the system. Challengers are likely to start the challenge at a level above $h' = 0.02$ which corresponds to a liquidation probability of over 90% as we see in Figure 2.8. Therefore we conclude that with these parameters, we do not need to compensate the system with a minting fee, unless there is a high risk-aversion.

In Figure 2.9 we see that the system does not lose on average if the challenge starts above $1 + h' = 102\%$. From Figure 2.8 we learn that starting the challenge

[5] Having a vector of B bootstrap estimates $\mathbf{x}$, we report the point estimate as the mean of $\mathbf{x}$, and the error at the a-level as $z_{1-a}\sqrt{V[\mathbf{x}]/B}$ with z_{1-a} the standard normal quantile.

process at $1 + h' = 102\%$, there is about a 90% probability that the challenge ends in a liquidation, hence the challenge is expected to start above the 102% level. We conclude that the minters do not need to be charged a risk premium to compensate the system, unless risk-aversion would be very high.

2.7.2 Capital Requirements

Banking regulation prescribes three types of capital requirements for which we find a meaningful analogue in the Frankencoin system, see for example the Basel III banking regulation (Basel Committee on Banking Supervision, 2010) or the Dodd-Frank Act (Acharya et al., 2010).

1. *Risk-based capital requirements.* These reserve requirements are based on the sum of the reserve requirements of each individual position, whereas each position can have its own risk weight. The equivalent to that in the Frankencoin system is the requirement to contribute to the minters reserve, which might differ for each position based on its riskiness.

2. *Leverage limits* restrict the overall amount of leverage in the balance sheet. The balance sheet of Frankencoin is presented in Appendix 2.A. Leverage limits are largely model-free and are therefore a safeguard against model risk (e.g., model risk arising from the risk-models used to calibrate plugin-specific capital requirements). Further, a leverage limit provides us with a global (i.e. Frankencoin-system-wide) capital limit. In the Frankencoin system, the global leverage limit is not directly enforced but economic incentives are designed such that in equilibrium, the system should exhibit a robust leverage ratio. On top of that, the collective minters reserve provides an additional layer of protection that is comparable to the leverage limit.

3. *Concentration limits.* If the collateral of Frankencoin was mainly exposed to the price of one asset, Frankencoin could more easily collapse following a large price drop of that asset, compared to a more diversified collateral in the system. The concentration limit aims at limiting the exposure to a single asset or a group of related assets.

The main difference between traditional Lombard loans and the Frankencoin's collateralized minting positions is the treatment of the proceeds in case of a liquidation. When a traditional Lombard loan is under-collateralized and the margin call not answered, the lender only sells as much collateral as needed to restore the loan-to-value ratio. Also, excess proceeds from the sale are returned to the owner of the collateral. In contrast, the Frankencoin system liquidates the whole positions as soon as the price reaches the trigger level and collects excess proceeds as profits. The range between the loan-to-value level and the liquidation level can be considered as conditional capital contribution by the borrower and for capital requirements considerations, this conditional capital has equity-like risk-absorbing properties. So in effect, both the equity and the minters reserve add to the risk-absorbing capacity of the system and should be taken into account when calculating the equivalent to the leverage ratio for banks. With regards to the minters reserve, the system resembles clearing houses, where each clearing member is required to contribute to the guaranty fund based on the risk of their position.

2.7.3 Risk-based Capital Requirements

Lombard loans are loans extended by banks to their customers, secured by the customers' securities that are held in bank custody. In the Lombard loan agreement, bank and customer agree on the terms of the loan, including that additional assets have to be provided in case the value of the collateral falls below a margin call level. The bank has the right to sell the pledged securities, if the value falls below an agreed level. In the Frankencoin system, there is no agreement beyond what is defined in the smart contract, and hence there will not be any bankruptcy litigation. As a consequence, there is a loss to the system that has to be covered with Frankencoin reserves, as soon as the collateral drops below the loan amount. Not surprisingly, this difference renders the Basel iii capital requirements for collateralized loans inadequate for the Frankencoin system. We therefore propose a more conservative approach to capital reserves.

Each mint plugin defines the required reserves of ZCHF to be held against the issued volume of ZCHF. We now assess how to specifically determine the risk-based capital requirements for our illustrative setup. We do not require risk-based capital reserves for the bridge plugins, so we focus entirely on the collateralized mint plugin, collateralized in Bitcoin.

The assessment we perform now differs from the one we made to determine the minimal fee in two main aspects, (1) we want to estimate a loss for the whole collateralized mint plugin not only for an individual position, and (2) we are not looking for an insurance valuation but for the capital required. The latter point implies that we do not need any assumption on risk-neutral measures but we can directly work with the observed data and use real world probabilities.

We start by applying the Basel iii rules for risk based capital.

From Basel Rules to Frankencoin Capital Requirements

The Basel Committee of Banking Supervision defines capital rules for collateralized loans in the Basel iii regulatory framework. In view of the authors, the collateralized mint plugin best meets the conditions to be considered a "repo-style transaction", see 22.66 in Basel Committee on Banking Supervision (2019a), henceforth [CRE22]. In this framework, the bank first weights assets according to the risk to obtain the risk-weighted assets (RWA). The bank is then required to hold a certain amount of capital, for example Common Equity Tier 1 must be at least 4.5% of RWA and total capital must be at least 8% of RWA, for details see Basel Committee on Banking Supervision (2019b). For collateralized loans, the rules are detailed in [CRE22]. Banks have two options: *"Banks may opt for either the simple approach, which substitutes the risk weighting of the collateral for the risk weighting of the counterparty for the collateralized portion of the exposure (generally subject to a 20% floor), or for the comprehensive approach, which allows a more precise offset of collateral against exposures, by effectively reducing the exposure amount by the value ascribed to the collateral"*, see 22.12 in [CRE22].

We focus on the comprehensive approach. First, the bank determines the exposure amount after risk mitigation, E^*, which depends on the loan and the collateral. Both are subject to a haircut, see 22.40 in [CRE22]:

$$E^* = \max[0, E(1 + H_e) - C(1 - H_c - H_{fx}), \tag{2.16}$$

where E^* is the exposure amount after risk mitigation, E the current value of the exposure (the loan), H_e haircut appropriate to the exposure, C the current value of the collateral received, H_c the haircut appropriate to the collateral, H_{fx} the haircut appropriate for currency mismatch between the collateral and exposure. The exposure amount E^* is to be multiplied by the risk weight of the counterparty to obtain the risk weighted asset amount for the collateralized loan. Collateral haircuts are either determined based on the type of collateral (e.g., 25% for 'other equity' and a holding period of 10 days). The exposure is then multiplied by the risk weight of the counterparty to obtain RWA. If the counterparty is not rated, a weight of 1.5 is applied. Figure 2.10 summarizes this approach. In Appendix 2.B we calculate the Basel iii capital requirements using the comprehensive approach and using the approach with own haircut estimates based on a 99%-VaR as per Basel iii. Table 2.6 shows the results: per Basel, the required capital reserve results to only about 1% of outstanding loans, even when assuming a loan-to-value ratio of $1/(1+h) = 1/1.1$ (which is always below current exposure assuming liquidations are effective).

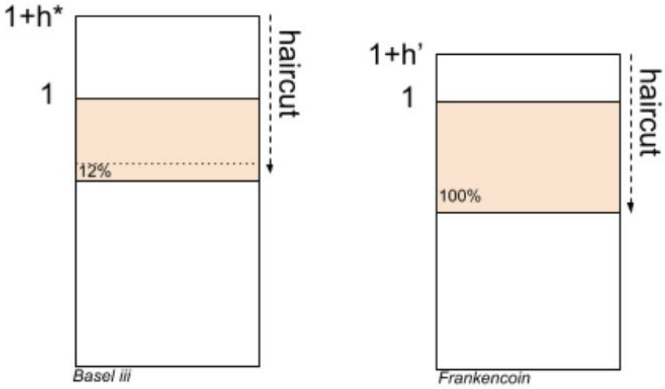


Figure 2.10: Basel iii vs own approach. Basel iii subtracts a haircut from the current collateralization value $(1 + h^*)$ to obtain the exposure after risk mitigation (shaded area). This quantity is multiplied by 1.5 for unrated counterparties. Total capital is to be 8% of this. However, in the Frankencoin system there is no loan agreement beyond the smart contract, and hence no litigation. We therefore propose a more strict approach, in which (1) the calculation of the haircut starts at a conservative liquidation level $(1 + h')$, (2) the entire exposure after risk mitigation is to be covered, as opposed to $1.5 \cdot 8\% = 12\%$.

To render the Frankencoin system resilient, the capital requirements must not assume that litigation is possible, hence the whole exposure after risk mitigation should be covered by capital reserves, as opposed to 12% per Basel iii. Second, instead of applying the haircut on the current exposure value, $(1 + h^*)$ as per Basel iii, we apply the haircut on a conservative liquidation level $(1+h')$. This is illustrated in Figure 2.10. To calculate the haircut, we estimate an empirical VaR at the 99%-level (a VaR level in line with Basel iii). We call this the Frankencoin Proposal in Table 2.9. Second, we look at the Basel iii requirements for collateralized lending

Method	Capital
Frankencoin Proposal (empirical VaR)	10%
RWA based, Basel iii Comprehensive	1.1%
RWA based, Basel iii Comprehensive (own haircuts)	1.3%

Table 2.9: Risk-based capital requirements. Results of capital reserves required using different approaches. For the setup at hand, we recommend that the system should hold 10% of the outstanding loans as Frankencoin reserves.

Capital Estimation per Frankencoin Proposal

We construct empirical samples of loss data by applying the 24h log-returns to the loss function given by Equation (2.9), which we repeat here:

$$\tilde{P} = \left[(1+h')e^{\tilde{r}_\tau} - (1+k)\right] D,$$

and we multiply the samples by -1 to have a positive number for a loss. Figure 2.11 gives an overview of the loss data for different challenger levels h' via boxplots. We see that the median level of losses are benign, even for very low challenger levels of $h' = -1\%$ (meaning the liquidation is only started when the collateral has a value of 99% of the loan notional, although the position would liquidate if the collateral is at 110% of the loan notional). However, we also see that there are high losses for all levels of h' depicted, driven by two extreme returns in the data, where the loss exceeds 20% of the loan notional. These losses are at a loan level and apply if the loan challenge starts at a collateral coverage ratio of $(1 + h')$. On a plugin-wide level, we would only lose the same percentage of the outstanding loan notional, if all loans started at $(1 + h')$ simultaneously. Realistically, collateral coverage ratios are more spread out and hence, the loan-level losses observed in Figure 2.11 are upper bounds to the plugin-wide loss.

Table 2.10 presents the 99-percentiles for daily losses (using the loss empirical loss distribution seen in Figure 2.11, given the challenge start level h'. A conservative starting level is $h' = 2\%$ as we have seen in the previous chapter. The associated 10.73% loss at the 99%-level would assume that all loans have an equally low collateralization of h'. To conclude, using our proposed approach that deviates from the Basel iii rules towards the conservative side, we propose to set the capital requirement to 10% of the outstanding loans.

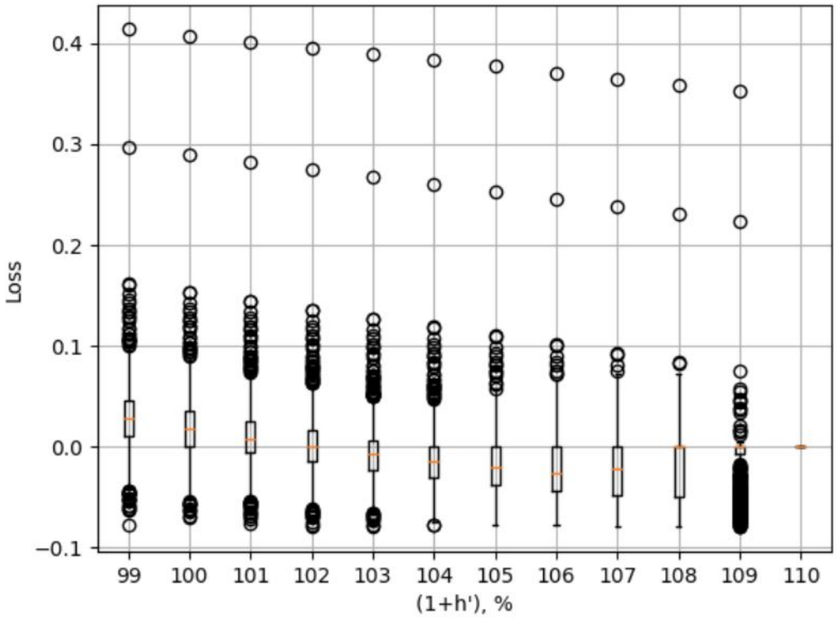


Figure 2.11: Empirical loss boxplot. This figure shows standard boxplots for different challenger levels h', given that a liquidation is triggered when the challenge results in a collateral value of below 110% of the loan, and a challenger fee of $k = 2\%$. The vertical bar depicts the median, the boxes reach from the first quartile to the third quartile, the whiskers extend to the max/min or at most 1.5 times the range of the box, circles are plotted outside the span of the whiskers. For values h' closer to $h = 0.10$, the loss more often ends in a gain (negative number). At $h = 0.10$ the challenge is averted. Losses stay on average benign but we also observe a few very high losses that exceed 20% of the loan notional due to two very negative returns (-49%, -31%).

h'	VaR99, %
-0.01	13.42
0.00	12.52
0.01	11.63
0.02	10.73
0.03	9.84
0.04	8.94
0.05	8.05
0.06	7.15
0.07	6.26
0.08	5.36
0.09	3.77
0.10	0.00

Table 2.10: Empirical VaR. This table shows the empirical VaR in percent of the outstanding loan amount given the challenge starting level h'.

2.7.4 Resulting Leverage Ratio

We have observed in section 2.3.3 that in equilibrium, we can expect equity capital to be about a third of the effectively borrowed capital. Furthermore, there is a minters reserve that can be used to cover losses once the equity holders have been wiped out. Assuming that the average minters reserve requirement is about 20% of the borrowed capital, we arrive at an exceptionally conservative leverage ratio of above 50%! However, taking into account that it is further possible to mint stablecoins through stablecoin bridges as depicted in the balance sheet of appendix 2.A, this leverage ratio can be lower to the extent that the bridges are used. To guard against the imported risks from bridged stablecoins, the bridges have built-in limits for the amount of Frankencoins that can be minted by them. Given this conservative approach, the main risk for the Frankencoin system does not seem to stem from the under-collateralization of individual positions, but from a potential lack of competitiveness with other protocols that are willing to take more risks.

2.8 Conclusion

Blockchain technology opens the possibility to create a new kind of monetary institutions that are governed in a decentralized way. Unlike bank loans that are issued by a centralized entity, the presented Frankencoin can be minted by the borrowers themselves, given that they provide suitable collateral, and the system has sufficient capital in its reserve. In the presence of a central bank digital currency (CBDC), see for example Chaum, Grothoff, and Moser (2021) or Brunnermeier and Niepelt (2019), the presented Frankencoin could come with a bridge to that CBDC, in which case the Frankencoin system could be seen as a fractional reserve multiplier just like the established banks are for traditional central bank money. With that perspective, it is no surprise that risk mitigation strategies from the current banking regulation can serve as a starting point to risk mitigation in the Frankencoin system.

Appendix

2.A Frankencoin Balance Sheet

Figure 2.A.1 presents a stylized balance sheet view of the Frankencoin system. When central banks print and issue currency, the outstanding amounts appear on the liability side of the balance sheet. Similarly, the minted Frankencoins also appear on the same side of the balance sheet. To see how they end up there, one first needs to consider the balance sheet of an individual minter. When a minter mints new Frankencoins, the freshly minted coins appear on the asset side of the minter's balance and at the same time, a repayment obligation is created on the liabilities side. For the Frankencoin system, the opposite happens: the minter's repayment obligation appears on the asset side and the minted ZCHF among the liabilities.

One should note that the total balance sheet of the Frankencoin system is larger than the total number of Frankencoins in circulation. That is because the same Frankencoin can appear multiple times. If a minter sells his ZCHF and the buyer uses them to buy Frankencoin Pool Shares, these Frankencoins appear a second time on the balance sheet, this time as reserves with a corresponding increase in equity on the passive side. Similarly, when for example minting 100 ZCHF against a collateral with a loan-to-value ratio $l = 0.8$, it is not only the total supply and the minter repayment obligations that increase by 100 ZCHF each, but also the reserve and the minter reserve go up by 20 ZCHF each as 20% of the Frankencoins are immediately redirected to the reserve after minting.

In case a minter cannot meet the repayment obligation and the subsequent liquidation results in a shortfall, it is in first priority covered by the minter reserve associated to the under-collateralized position, in second priority covered by equity, and in third priority covered by the collective minters reserve. While the balance sheet shows the Frankencoins minted against collateral assets, the provided collateral itself does not appear on the balance sheet as it belongs to the minter.

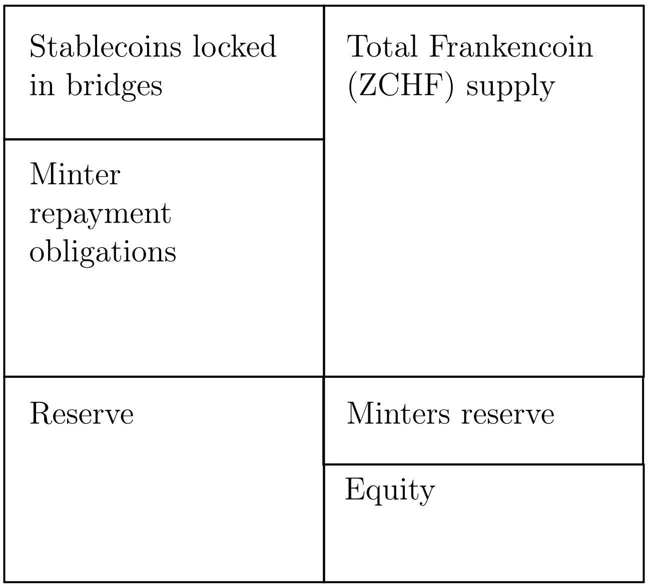

Stablecoins locked in bridges
Minter repayment obligations
Reserve
Total Frankencoin (ZCHF) supply
Minters reserve
Equity

Figure 2.A.1: Balance sheet. The balance sheet of the Frankencoin system.

2.B Basel Rules for Collateralized Lending

Haircuts are based on the type of exposure, see 22.4 in [CRE22]. For main equity indices and gold, the haircut is 15%, for other equities 25% (10 day holding period). Supervisors may also permit the banks to calculate their own haircuts.

Application of the comprehensive Approach

For repo-style transactions, supervisors may choose not to apply the haircuts specified in the comprehensive approach and may instead apply a haircut of zero, if the counterparty is a core market participant' as determined by the regulator. This would result in zero RWA because the collateralized mint plugin requires over-collateralization of all loans.

If the Frankencoin system is not exempt from the requirement that the counterparty is to be a core market participant, we may choose the parameters as follows.

$H_e = 0$	The exposure is in CHF (cash), hence no valuation risk
$H_c = 25\%/\sqrt{2}$	The 'other equity' haircut seems to be the most appropriate to reflect collateral in BTC. The 25% are for a 10-day holding period, hence we scale by $1/\sqrt{2}$ to have a 5-day haircut (which is the minimum allowed – our actual holding period is 1 day)

$H_{fx} = 0\%$	The asset is denominated in CHF, hence no additional currency risk

The above numbers are based on a 10-day holding period. If holding periods differ, the percentages are to be scaled by the square-root of time formula. However, for repo-style transactions, a minimum holding period of five days applies. Now, assuming a current loan to value ratio of $1/(1+h^*)$, and an (artificial) holding period of 5 days, we use Equation (2.16) and set $E = 1$, $C = (1+h^*)$:

$$E(h^*) = \max[0, 1 - (1+h^*)(1-H_c)], \tag{2.17}$$

$$= \max[0, H_c(1+h^*) - h^*], \tag{2.18}$$

where the second line follows from algebra. For example, if $h^* > 0.21$, the resulting exposure is zero (using $Hc = 25\%/\sqrt{2}$, we set the second term in the max-term to zero and solve for h^*). Finally, the exposure E^* needs to be weighted by the counterparty risk weight. The counterparty in the Frankencoin system has generally no rating and cannot be forced to make up for losses in the system, hence the worst weight has to apply, which is w=150%, see Basel Committee on Banking Supervision (2017). Assuming that the loans are just at the (previously defined) liquidation level of $h = 10\%$, we arrive at RWA of

$$E(h)w = 0.09445 \cdot 1.5 = 14\%$$

of the loan amount, of which 8% are required as total capital. Hence, the capital requirement amounts to 1.1% of outstanding loans.

Comprehensive Approach: own Estimates for Haircuts

To calculate the haircuts, a 99$^{\text{th}}$ percentile, one-tailed confidence interval is to be used (see 22.50 of [CRE22]), and the choice of historical observation period (sample period) for calculating haircuts shall be a minimum of one year (see 22.53). The minimum holding period is to be set to 5 days. The VaR results in

$$VaR_q = (1 - \exp{(r_q)}) + k, \tag{2.19}$$

where q is the quantile (99% per Basel iii), r_q is the quantile return, and k the challenger fee. Using the historical data detailed in Appendix 2.C, the 5-day loss at the 99% quantile (using 1-day overlapping returns to calculate historical VaR at 1%-level) amounts to $19.44\% + k = 21.44\%$ (this compares to a haircut of $25\%/\sqrt{2} \approx 17.68\%$ used above). The resulting RWA, using Equation (2.18) and multiplying by the risk weight, at $h^* = 10\%$ is 20.38%, of which 8% are required as capital.[6] Hence, the capital requirement amounts to 1.6% of outstanding loans.

2.C Data

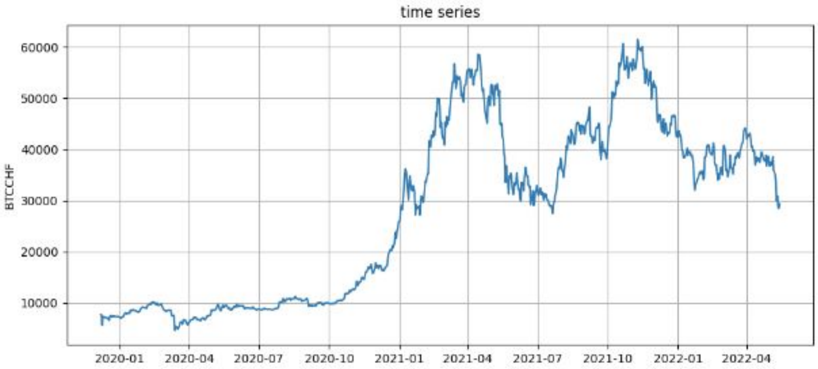


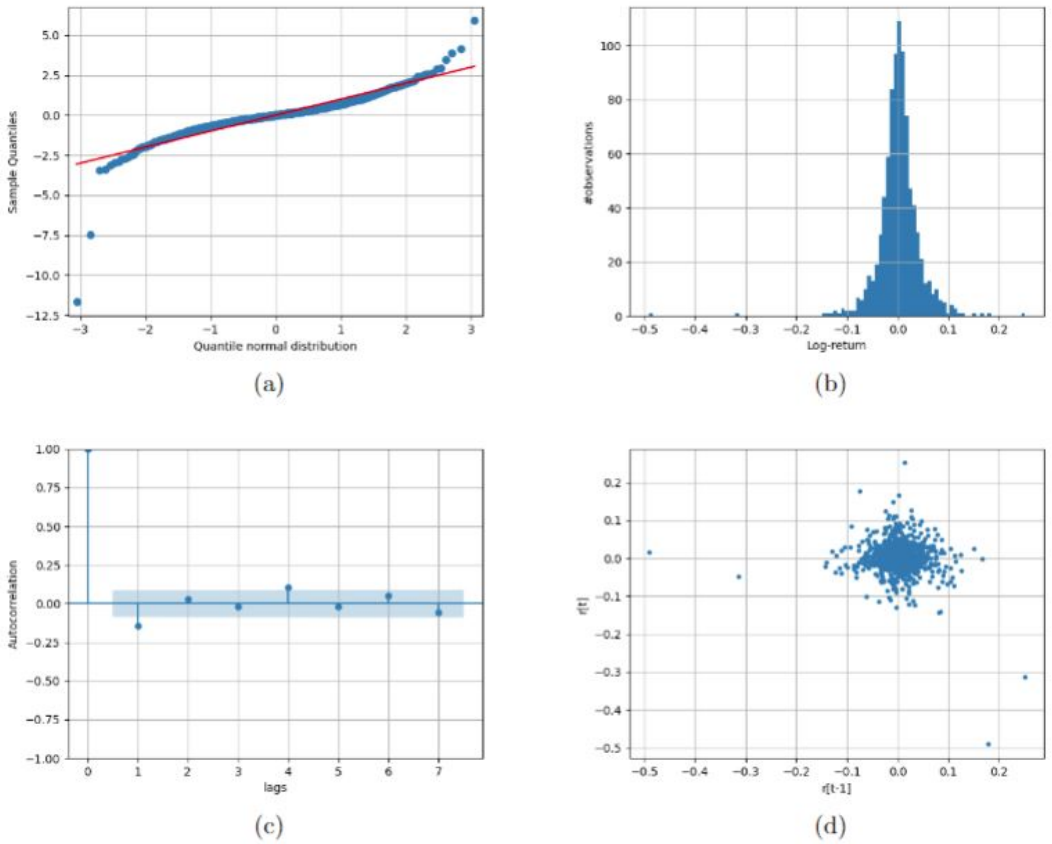
Figure 2.C.1: BTCCHF trade data. This figure plots the level data of the BTCCHF time-series. We have 890 observations of daily candle data without gaps from 2019-12-07 to 2022-05-14.

We gather 1-hour candle data from Kraken, consisting of a timestamp, open, low, high, close, number of trades, and volume.[7] From each open and close price we calculate 24h log-returns. Our return data has the following summary statistics. Table 2.9 tests the time-series of 24h log-returns for stationarity via Augmented Dickey-Fuller test and rejects the null-hypothesis of non-stationarity.

Figure 2.C.2 present a quantile-quantile plot against the normal distribution and a histogram. Figure 2.C.2 also analyses serial correlation of the 24h log-returns. The Autocorrelation Function shows a significant negative correlation at lag one, which however turns out to be driven from extreme returns, as we can see on plot (d).

[6] RWA $= E(h^*)w = (\text{VaR}(1 + h^*) - h^*)1.5$

[7] See support.kraken.com/hc/en-us/articles/360047124832-Downloadable-historical-OHLCVT-Open-High-Low-Close-Volume-Trades-data.



(a) (b)

(c) (d)

Figure 2.C.2: Return data from Kraken. Plot (a) shows a quantile-quantile plot of the 24h log-return data against a normal distribution, (b) plots a histogram of the 24h log-returns. From (a) we see that we have more extreme returns than what a normal distribution would suggest. In the center of the distribution, the returns move less than the normal distribution would imply. Plot (c) shows the autocorrelation function and a confidence band at the 0.95-level. Chart (d) plots the returns against the previous day return. The figure indicates that the significant autocorrelation at level 1 must be driven by rare extreme returns.

num. observations	890
min, max	[-49.09%, 25.13%]
mean	0.18%
variance	0.0018
skewness	-1.90
kurtosis	25.93

Table 2.C.1: Summary statistics for 24h log-return data.

ADF Statistic:	-13.8	
p-value:	0.000000	vspace15pt
Critical Value for 1%:	-3.4	

Table 2.C.2: Stationarity test. The ADF test suggests that the log-return data is stationary.

2.D Frankencoin (ZCHF) Token

The Frankencoin contract is an ERC-20 token with ticker ZCHF. It allows anyone to suggest new minter contracts that can mint Frankencoins under arbitrary rules. New minting contracts can be vetoed by qualified pool share holders. It also keeps the accounting for the minter's reserves when Frankencoins are minted or burned.

```
pragma solidity ^0.8.0;

import "./utils/ERC20PermitLight.sol";
import "./Equity.sol";
import "./interface/IReserve.sol";
import "./interface/IFrankencoin.sol";

/**
 * @title FrankenCoin
 * The Frankencoin (ZCHF) is an ERC-20 token that is designed to track the value of the Swiss franc.
 * It is not upgradable, but open to arbitrary minting plugins. These are automatically accepted if none of the
 * qualified pool share holders casts a veto, leading to a flexible but conservative governance.
 */
contract Frankencoin is ERC20PermitLight, IFrankencoin {
    /**
     * Minimal fee and application period when suggesting a new minter.
     */
    uint256 public constant MIN_FEE = 1000 * (10 ** 18);
    uint256 public immutable MIN_APPLICATION_PERIOD; // for example 10 days

    /**
     * The contract that holds the reserve.
     */
    IReserve public immutable override reserve;

    /**
     * How much of the reserve belongs to the minters. Everything else belongs to the pool share holders.
     * Stored with 6 additional digits of accuracy so no rounding is necessary when dealing with parts per
     * million (ppm) in reserve calculations.
     */
    uint256 private minterReserveE6;

    /**
```

```
     * Map of minters to approval time stamps. If the time stamp is in the past, the minter contract is allowed
     * to mint Frankencoins.
     */
    mapping(address minter => uint256 validityStart) public minters;

    /**
     * List of positions that are allowed to mint and the minter that registered them.
     */
    mapping(address position => address registeringMinter) public positions;

    event MinterApplied(address indexed minter, uint256 applicationPeriod, uint256 applicationFee, string message);
    event MinterDenied(address indexed minter, string message);
    event Loss(address indexed reportingMinter, uint256 amount, uint256 reserve);
    event Profit(address indexed minter, uint256 amount, uint256 reserve);

    error PeriodTooShort();
    error FeeTooLow();
    error AlreadyRegistered();
    error NotMinter();
    error TooLate();

    modifier minterOnly() {
        if (!isMinter(msg.sender) && !isMinter(positions[msg.sender])) revert NotMinter();
        _;
    }

    /**
     * Initiates the Frankencoin with the provided minimum application period for new plugins
     * in seconds, for example 10 days, i.e. 3600*24*10 = 864000
     */
    constructor(uint256 _minApplicationPeriod) ERC20(18) {
        MIN_APPLICATION_PERIOD = _minApplicationPeriod;
        reserve = new Equity(this);
    }

    function name() external pure override returns (string memory) {
        return "Frankencoin";
    }

    function symbol() external pure override returns (string memory) {
        return "ZCHF";
    }

    function initialize(address _minter, string calldata _message) external {
        require(totalSupply() == 0 && reserve.totalSupply() == 0);
        minters[_minter] = block.timestamp;
        emit MinterApplied(_minter, 0, 0, _message);
    }

    /**
     * Publicly accessible method to suggest a new way of minting Frankencoin.
     * @dev The caller has to pay an application fee that is irrevocably lost even if the new minter is vetoed.
     * The caller must assume that someone will veto the new minter unless there is broad consensus that the new minter
     * adds value to the Frankencoin system. Complex proposals should have application periods and applications fees
     * above the minimum. It is assumed that over time, informal ways to coordinate on new minters emerge. The message
     * parameter might be useful for initiating further communication. Maybe it contains a link to a website describing
     * the proposed minter.
     *
     * @param _minter              An address that is given the permission to mint Frankencoins
     * @param _applicationPeriod   The time others have to veto the suggestion, at least MIN_APPLICATION_PERIOD
     * @param _applicationFee      The fee paid by the caller, at least MIN_FEE
     * @param _message             An optional human readable message to everyone watching this contract
     */
    function suggestMinter(
        address _minter,
        uint256 _applicationPeriod,
        uint256 _applicationFee,
        string calldata _message
    ) external override {
        if (_applicationPeriod < MIN_APPLICATION_PERIOD) revert PeriodTooShort();
        if (_applicationFee < MIN_FEE) revert FeeTooLow();
        if (minters[_minter] != 0) revert AlreadyRegistered();
        _transfer(msg.sender, address(reserve), _applicationFee);
        minters[_minter] = block.timestamp + _applicationPeriod;
        emit MinterApplied(_minter, _applicationPeriod, _applicationFee, _message);
    }

    /**
     * Make the system more user friendly by skipping the allowance in many cases.
     * @dev We trust minters and the positions they have created to mint and burn as they please, so
     * giving them arbitrary allowances does not pose an additional risk.
     */
    function _allowance(address owner, address spender) internal view override returns (uint256) {
        uint256 explicit = super._allowance(owner, spender);
        if (explicit > 0) {
            return explicit; // don't waste gas checking minter
        } else if (isMinter(spender) || isMinter(getPositionParent(spender)) || spender == address(reserve)) {
            return INFINITY;
        } else {
            return 0;
        }
    }

    /**
     * The reserve provided by the owners of collateralized positions.
     * @dev The minter reserve can be used to cover losses after the equity holders have been wiped out.
     */
    function minterReserve() public view returns (uint256) {
        return minterReserveE6 / 1000000;
    }

    /**
     * Allows minters to register collateralized debt positions, thereby giving them the ability to mint Frankencoins.
     * @dev It is assumed that the responsible minter that registers the position ensures that the position can be trusted.
     */
    function registerPosition(address _position) external override {
        if (!isMinter(msg.sender)) revert NotMinter();
        positions[_position] = msg.sender;
    }

    /**
     * The amount of equity of the Frankencoin system in ZCHF, owned by the holders of Frankencoin Pool Shares.
     * @dev Note that the equity contract technically holds both the minter reserve as well as the equity, so the minter
     * reserve must be subtracted. All fees and other kind of income is added to the Equity contract and essentially
     * constitutes profits attributable to the pool share holders.
     */
    function equity() public view returns (uint256) {
        uint256 balance = balanceOf(address(reserve));
        uint256 minReserve = minterReserve();
        if (balance <= minReserve) {
            return 0;
        } else {
            return balance - minReserve;
        }
    }

    /**
     * Qualified pool share holders can deny minters during the application period.
     * @dev Calling this function is relatively cheap thanks to the deletion of a storage slot.
     */
    function denyMinter(address _minter, address[] calldata _helpers, string calldata _message) external override {
        if (block.timestamp > minters[_minter]) revert TooLate();
        reserve.checkQualified(msg.sender, _helpers);
        delete minters[_minter];
        emit MinterDenied(_minter, _message);
    }

    /**
     * Mints the provided amount of ZCHF to the target address, automatically forwarding
     * the minting fee and the reserve to the right place.
     */
    function mintWithReserve(
        address _target,
        uint256 _amount,
        uint32 _reservePPM,
        uint32 _feesPPM
    ) external override minterOnly {
        uint256 usableMint = (_amount * (1000_000 - _feesPPM - _reservePPM)) / 1000_000; // rounding down is fine
        _mint(_target, usableMint);
        _mint(address(reserve), _amount - usableMint); // rest goes to equity as reserves or as fees
        minterReserveE6 += _amount * _reservePPM;
    }

    function mint(address _target, uint256 _amount) external override minterOnly {
        _mint(_target, _amount);
    }

    /**
     * Anyone is allowed to burn their ZCHF.
     */
    function burn(uint256 _amount) external {
        _burn(msg.sender, _amount);
    }

    /**
     * Burn someone elses ZCHF.
     */
    function burnFrom(address _owner, uint256 _amount) external override minterOnly {
        _burn(_owner, _amount);
    }

    /**
     * Burn that amount without reclaiming the reserve, but freeing it up and thereby essentially donating it to the
     * pool share holders. This can make sense in combination with 'notifyLoss', i.e. when it is the pool share
     * holders that bear the risk and depending on the outcome they make a profit or a loss.
     *
     * Design rule: Minters calling this method are only allowed to so for tokens amounts they previously minted with
     * the same _reservePPM amount.
     *
     * For example, if someone minted 50 ZCHF earlier with a 20% reserve requirement (200000 ppm), they got 40 ZCHF
     * and paid 10 ZCHF into the reserve. Now they want to repay the debt by burning 50 ZCHF. When doing so using this
     * method, 50 ZCHF get burned and on top of that, 10 ZCHF previously assigned to the minter's reserved are
     * reassigned to the pool share holders.
     */
    function burnWithoutReserve(uint256 amount, uint32 reservePPM) public override minterOnly {
        _burn(msg.sender, amount);
        uint256 reserveReduction = amount * reservePPM;
        if (reserveReduction > minterReserveE6) {
            emit Profit(msg.sender, minterReserveE6, 0);
            minterReserveE6 = 0; // should never happen, but we want robust behavior in case it does
        } else {
            minterReserveE6 -= reserveReduction;
            emit Profit(msg.sender, reserveReduction, minterReserveE6);
        }
    }

    /**
     * Burns the provided number of tokens plus whatever reserves are associated with that amount given the reserve
     * requirement. The caller is only allowed to use this method for tokens also minted through the caller with the
     * same _reservePPM amount.
     *
     * Example: the calling contract has previously minted 100 ZCHF with a reserve ratio of 20% (i.e. 200000 ppm).
     * Now they have 41 ZCHF that they do not need so they decide to repay that amount. Assuming the reserves are
     * only 90% covered, the call to burnWithReserve will burn the 41 plus 9 from the reserve, reducing the outstanding
     * 'debt' of the caller by 50 ZCHF in total. This total is returned by the method so the caller knows how much less
     * they owe.
     */
    function burnWithReserve(
        uint256 _amountExcludingReserve,
        uint32 _reservePPM
    ) external override minterOnly returns (uint256) {
        uint256 freedAmount = calculateFreedAmount(_amountExcludingReserve, _reservePPM); // 50 in the example
        minterReserveE6 -= freedAmount * _reservePPM; // reduce reserve requirements by original ratio
        _transfer(address(reserve), msg.sender, freedAmount - _amountExcludingReserve); // collect assigned reserve
        _burn(msg.sender, freedAmount); // burn the rest of the freed amount
        return freedAmount;
    }

    /**
     * Burns the target amount taking the tokens to be burned from the payer and the payer's reserve.
     * Only use this method for tokens also minted by the caller with the same _reservePPM.
     *
     * Example: the calling contract has previously minted 100 ZCHF with a reserve ratio of 20% (i.e. 200000 ppm).
     * To burn half of that again, the minter calls burnFrom with a target amount of 50 ZCHF. Assuming that reserves
     * are only 90% covered, this call will deduct 41 ZCHF from the payer's balance and 9 from the reserve, while
     * reducing the minter reserve by 10.
     */
    function burnFromWithReserve(
        address payer,
        uint256 targetTotalBurnAmount,
        uint32 reservePPM
    ) external override minterOnly returns (uint256) {
        uint256 assigned = calculateAssignedReserve(targetTotalBurnAmount, reservePPM);
        _transfer(address(reserve), payer, assigned); // send reserve to owner
        _burn(payer, targetTotalBurnAmount); // and burn the full amount from the owner's address
        minterReserveE6 -= targetTotalBurnAmount * reservePPM; // reduce reserve requirements by original ratio
        return assigned;
    }

    /**
     * Calculates the reserve attributable to someone who minted the given amount with the given reserve requirement.
     * Under normal circumstances, this is just the reserver requirement multiplied by the amount. However, after a
     * severe loss of capital that burned into the minter's reserve, this can also be less than that.
     */
    function calculateAssignedReserve(uint256 mintedAmount, uint32 _reservePPM) public view returns (uint256) {
        uint256 theoreticalReserve = (_reservePPM * mintedAmount) / 1000000;
        uint256 currentReserve = balanceOf(address(reserve));
        uint256 minterReserve_ = minterReserve();
        if (currentReserve < minterReserve_) {
            // not enough reserves, owner has to take a loss
            return (theoreticalReserve * currentReserve) / minterReserve_;
        } else {
            return theoreticalReserve;
        }
    }

    /**
     * Calculate the amount that is freed when returning amountExcludingReserve given a reserve ratio of reservePPM,
     * taking into account potential losses. Example values in the comments.
     */
    function calculateFreedAmount(
        uint256 amountExcludingReserve /* 41 */,
        uint32 reservePPM /* 20% */
    ) public view returns (uint256) {
        uint256 currentReserve = balanceOf(address(reserve)); // 18, 10% below what we should have
        uint256 minterReserve_ = minterReserve(); // 20
        uint256 adjustedReservePPM = currentReserve < minterReserve_
            ? (reservePPM * currentReserve) / minterReserve_
            : reservePPM; // 18%
        return (1000000 * amountExcludingReserve) / (1000000 - adjustedReservePPM); // 41 / (1-18%) = 50
    }

    /**
     * Notify the Frankencoin that a minter lost economic access to some coins. This does not mean that the coins are
     * literally lost. It just means that some ZCHF will likely never be repaid and that in order to bring the system
     * back into balance, the lost amount of ZCHF must be removed from the reserve instead.
     *
     * For example, if a minter printed 1 million ZCHF for a mortgage and the mortgage turned out to be unsound with
     * the house only yielding 800'000 in the subsequent auction, there is a loss of 200'000 that needs to be covered
     * by the reserve.
     */
    function notifyLoss(uint256 _amount) external override minterOnly {
        uint256 reserveLeft = balanceOf(address(reserve));
        if (reserveLeft >= _amount) {
            _transfer(address(reserve), msg.sender, _amount);
        } else {
            _transfer(address(reserve), msg.sender, reserveLeft);
            _mint(msg.sender, _amount - reserveLeft);
        }
        emit Loss(msg.sender, _amount, reserveLeft);
    }

    /**
     * Returns true if the address is an approved minter.
     */
    function isMinter(address _minter) public view override returns (bool) {
        return minters[_minter] != 0 && block.timestamp >= minters[_minter];
    }

    /**
```

```
     * Returns the address of the minter that created this position or null if the provided address is unknown.
     */
    function getPositionParent(address _position) public view override returns (address) {
        return positions[_position];
    }
}
```

2.E Frankencoin Pool Share (FPS) Token

The Equity contract is an ERC-20 token with name *Frankencoin Pool Shares* ticker FPS. It contains an automated market maker that governs the issuance and redemption of pool shares, keeps track of how long pool shares have been held on the same address, and contains the governance logic.

```
pragma solidity ^0.8.0;

import "./Frankencoin.sol";
import "./utils/MathUtil.sol";
import "./interface/IReserve.sol";
import "./interface/IERC677Receiver.sol";

/**
 * @title Equity
 * If the Frankencoin system was a bank, this contract would represent the equity on its balance sheet.
 * Like with a corporation, the owners of the equity capital are the shareholders, or in this case the holders
 * of Frankencoin Pool Shares (FPS) tokens. Anyone can mint additional FPS tokens by adding Frankencoins to the
 * reserve pool. Also, FPS tokens can be redeemed for Frankencoins again after a minimum holding period.
 * Furthermore, the FPS shares come with some voting power. Anyone that held at least 3% of the holding-period-
 * weighted reserve pool shares gains veto power and can veto new proposals.
 */
contract Equity is ERC20PermitLight, MathUtil, IReserve {
    /**
     * The VALUATION_FACTOR determines the market cap of the reserve pool shares relative to the equity reserves.
     * The following always holds: Market Cap = Valuation Factor * Equity Reserve = Price * Supply
     *
     * In the absence of profits and losses, the variables grow as follows when FPS tokens are minted:
     *
     * |   Reserve     |   Market Cap  |     Price     |     Supply   |
     * |          1000 |          3000 |             3 |         1000 |
     * |       1000000 |       3000000 |           300 |        10000 |
     * |    1000000000 |    3000000000 |         30000 |       100000 |
     * | 1000000000000 | 3000000000000 |       3000000 |      1000000 |
     *
     * I.e., the supply is proporational to the cubic root of the reserve and the price is proportional to the
     * squared cubic root. When profits accumulate or losses materialize, the reserve, the market cap,
     * and the price are adjusted proportionally, with the supply staying constant. In the absence of an extreme
     * inflation of the Swiss franc, it is unlikely that there will ever be more than ten million FPS.
     */
    uint32 public constant VALUATION_FACTOR = 3;

    uint256 private constant MINIMUM_EQUITY = 1000 * ONE_DEC18;

    /**
     * The quorum in basis points. 100 is 1%.
     */
    uint32 private constant QUORUM = 200;

    /**
     * The number of digits to store the average holding time of share tokens.
     */
    uint8 private constant TIME_RESOLUTION_BITS = 20;

    /**
     * The minimum holding duration. You are not allowed to redeem your pool shares if you held them
     * for less than the minimum holding duration at average. For example, if you have two pool shares on your
     * address, one acquired 5 days ago and one acquired 105 days ago, you cannot redeem them as the average
     * holding duration of your shares is only 55 days < 90 days.
     */
    uint256 public constant MIN_HOLDING_DURATION = 90 days << TIME_RESOLUTION_BITS; // Set to 5 for local testing

    Frankencoin public immutable zchf;

    /**
     * @dev To track the total number of votes we need to know the number of votes at the anchor time and when the
     * anchor time was. This is (hopefully) stored in one 256 bit slot, with the anchor time taking 64 Bits and
     * the total vote count 192 Bits. Given the sub-second resolution of 20 Bits, the implicit assumption is
     * that the timestamp can always be stored in 44 Bits (i.e. it does not exceed half a million years). Further,
     * given 18 decimals (about 60 Bits), this implies that the total supply cannot exceed
     *   192 - 60 - 44 - 20 = 68 Bits
     * Here, we are also save, as 68 Bits would imply more than a trillion outstanding shares. In fact,
     * a limit of about 2**36 shares (that's about 2**96 Bits when taking into account the decimals) is imposed
     * when minting. This means that the maximum supply is billions shares, which is could only be reached in
     * a scenario with hyper inflation, in which case the stablecoin is worthless anyway.
     */
    uint192 private totalVotesAtAnchor; // Total number of votes at the anchor time, see comment on the um
    uint64 private totalVotesAnchorTime; // 44 Bit for the time stamp, 20 Bit sub-second time resolution

    /**
     * Keeping track on who delegated votes to whom.
     * Note that delegation does not mean you cannot vote / veto any more, it just means that the delegate can
     * benefit from your votes when invoking a veto. Circular delegations are valid, do not help when voting.
     */
    mapping(address owner => address delegate) public delegates;

    /**
     * A time stamp in the past such that: votes = balance * (time passed since anchor was set)
     */
    mapping(address owner => uint64 timestamp) private voteAnchor; // 44 bits for time stamp, 20 subsecond resolution

    event Delegation(address indexed from, address indexed to); // indicates a delegation
    event Trade(address who, int amount, uint totPrice, uint newprice); // amount pos or neg for mint or redemption

    constructor(Frankencoin zchf_) ERC20(18) {
        zchf = zchf_;
    }

    function name() external pure override returns (string memory) {
        return "Frankencoin Pool Share";
    }

    function symbol() external pure override returns (string memory) {
        return "FPS";
    }

    /**
     * Returns the price of one FPS in ZCHF with 18 decimals precision.
     */
    function price() public view returns (uint256) {
        uint256 equity = zchf.equity();
        if (equity == 0 || totalSupply() == 0) {
            return ONE_DEC18; // initial price is 1000 ZCHF for the first 1000 FPS
        } else {
            return (VALUATION_FACTOR * zchf.equity() * ONE_DEC18) / totalSupply();
        }
    }

    function _beforeTokenTransfer(address from, address to, uint256 amount) internal override {
        super._beforeTokenTransfer(from, to, amount);
        if (amount > 0) {
            // No need to adjust the sender votes. When they send out 10% of their shares, they also lose 10% of
            // their votes so everything falls nicely into place. Recipient votes should stay the same, but grow
            // faster in the future, requiring an adjustment of the anchor.
            uint256 roundingLoss = _adjustRecipientVoteAnchor(to, amount);
            // The total also must be adjusted and kept accurate by taking into account the rounding error.
            _adjustTotalVotes(from, amount, roundingLoss);
        }
    }

    /**
```

```
     * Returns whether the given address is allowed to redeem FPS, which is the
     * case after their average holding duration is larger than the required minimum.
     */
    function canRedeem(address owner) public view returns (bool) {
        return _anchorTime() - voteAnchor[owner] >= MIN_HOLDING_DURATION;
    }

    /**
     * Decrease the total votes anchor when tokens lose their voting power due to being moved
     * @param from      sender
     * @param amount    amount to be sent
     */
    function _adjustTotalVotes(address from, uint256 amount, uint256 roundingLoss) internal {
        uint64 time = _anchorTime();
        uint256 lostVotes = from == address(0x0) ? 0 : (time - voteAnchor[from]) * amount;
        totalVotesAtAnchor = uint192(totalVotes() - roundingLoss - lostVotes);
        totalVotesAnchorTime = time;
    }

    /**
     * the vote anchor of the recipient is moved forward such that the number of calculated
     * votes does not change despite the higher balance.
     * @param to        receiver address
     * @param amount    amount to be received
     * @return the number of votes lost due to rounding errors
     */
    function _adjustRecipientVoteAnchor(address to, uint256 amount) internal returns (uint256) {
        if (to != address(0x0)) {
            uint256 recipientVotes = votes(to); // for example 21 if 7 shares were held for 3 seconds
            uint256 newbalance = balanceOf(to) + amount; // for example 11 if 4 shares are added
            // new example anchor is only 21 / 11 = 1 second in the past
            voteAnchor[to] = uint64(_anchorTime() - recipientVotes / newbalance);
            return recipientVotes % newbalance; // we have lost 21 % 11 = 10 votes
        } else {
            // optimization for burn, vote anchor of null address does not matter
            return 0;
        }
    }

    /**
     * Time stamp with some additional bits for higher resolution.
     */
    function _anchorTime() internal view returns (uint64) {
        return uint64(block.timestamp << TIME_RESOLUTION_BITS);
    }

    /**
     * The relative voting power of the address.
     * @return A percentage with 1e18 being 100%
     */
    function relativeVotes(address holder) external view returns (uint256) {
        return (ONE_DEC18 * votes(holder)) / totalVotes();
    }

    /**
     * The votes of the holder, excluding votes from delegates.
     */
    function votes(address holder) public view returns (uint256) {
        return balanceOf(holder) * (_anchorTime() - voteAnchor[holder]);
    }

    /**
     * Total number of votes in the system.
     */
    function totalVotes() public view returns (uint256) {
        return totalVotesAtAnchor + totalSupply() * (_anchorTime() - totalVotesAnchorTime);
    }

    /**
     * The number of votes the sender commands when taking the support of the helpers into account.
     * @param sender    The address whose total voting power is of interest
     * @param helpers   An incrementally sorted list of helpers without duplicates and without the sender.
     *                  The call fails if the list contains an address that does not delegate to sender.
     *                  For indirect delegates, i.e. a -> b -> c, both a and b must be included for both to count.
     * @return          The total number of votes of sender at the current point in time.
     */
    function votesDelegated(address sender, address[] calldata helpers) public view returns (uint256) {
        uint256 _votes = votes(sender);
        require(_checkDuplicatesAndSorted(helpers));
        for (uint i = 0; i < helpers.length; i++) {
            address current = helpers[i];
            require(current != sender);
            require(_canVoteFor(sender, current));
            _votes += votes(current);
        }
        return _votes;
    }

    function _checkDuplicatesAndSorted(address[] calldata helpers) internal pure returns (bool ok) {
        if (helpers.length <= 1) {
            return true;
        } else {
            address prevAddress = helpers[0];
            for (uint i = 1; i < helpers.length; i++) {
                if (helpers[i] <= prevAddress) {
                    return false;
                }
                prevAddress = helpers[i];
            }
            return true;
        }
    }

    /**
     * Checks whether the sender address is qualified given a list of helpers that delegated their votes
     * directly or indirectly to the sender. It is the responsiblity of the caller to figure out whether
     * helpes are necessary and to identify them by scanning the blockchain for Delegation events.
     */
    function checkQualified(address sender, address[] calldata helpers) public view override {
        uint256 _votes = votesDelegated(sender, helpers);
        if (_votes * 10000 < QUORUM * totalVotes()) revert NotQualified();
    }

    error NotQualified();

    /**
     * Increases the voting power of the delegate by your number of votes without taking away any voting power
     * from the sender.
     */
    function delegateVoteTo(address delegate) external {
        delegates[msg.sender] = delegate;
        emit Delegation(msg.sender, delegate);
    }

    function _canVoteFor(address delegate, address owner) internal view returns (bool) {
        if (owner == delegate) {
            return true;
        } else if (owner == address(0x0)) {
            return false;
        } else {
            return _canVoteFor(delegate, delegates[owner]);
        }
    }

    /**
     * Since quorum is rather low, it is important to have a way to prevent malicious minority holders
     * from blocking the whole system. This method provides a way for the good guys to team up and destroy
     * the bad guy's votes (at the cost of also reducing their own votes). This mechanism potentially
     * gives full control over the system to whoever has 51% of the votes.
     *
     * Since this is a rather aggressive measure, delegation is not supported. Every holder must call this
     * method on their own.
     * @param targets   The target addresses to remove votes from
     * @param votesToDestroy    The maximum number of votes the caller is willing to sacrifice
     */
    function kamikaze(address[] calldata targets, uint256 votesToDestroy) external {
        uint256 budget = _reduceVotes(msg.sender, votesToDestroy);
        uint256 destroyedVotes = 0;
        for (uint256 i = 0; i < targets.length && destroyedVotes < budget; i++) {
            destroyedVotes += _reduceVotes(targets[i], budget - destroyedVotes);
        }
        require(destroyedVotes > 0); // sanity check
        totalVotesAtAnchor = uint192(totalVotes() - destroyedVotes - budget);
        totalVotesAnchorTime = _anchorTime();
    }

    function _reduceVotes(address target, uint256 amount) internal returns (uint256) {
        uint256 votesBefore = votes(target);
        if (amount >= votesBefore) {
            amount = votesBefore;
            voteAnchor[target] = _anchorTime();
            return votesBefore;
        } else {
            voteAnchor[target] = uint64(_anchorTime() - (votesBefore - amount) / balanceOf(target));
            return votesBefore - votes(target);
        }
    }

    /**
     * Call this method to obtain newly minted pool shares in exchange for Frankencoins.
     * No allowance required (i.e. it is hardcoded in the Frankencoin token contract).
     * Make sure to invest at least 10e-12 * market cap to avoid rounding losses.
     *
     * @dev If equity is close to zero or negative, you need to send enough ZCHF to bring equity back to 1000 ZCHF.
     *
     * @param amount            Frankencoins to invest
     * @param expectedShares    Minimum amount of expected shares for frontrunning protection
     */
    function invest(uint256 amount, uint256 expectedShares) external returns (uint256) {
        zchf.transferFrom(msg.sender, address(this), amount);
        uint256 equity = zchf.equity();
        require(equity >= MINIMUM_EQUITY, "insuf equity"); // ensures that the initial deposit is at least 1000 ZCHF

        uint256 shares = _calculateShares(equity <= amount ? 0 : equity - amount, amount);
        require(shares >= expectedShares);
        _mint(msg.sender, shares);
        emit Trade(msg.sender, int(shares), amount, price());

        // limit the total supply to a reasonable amount to guard against overflows with price and vote calculations
        // the 36 bits are 68 bits for magnitude and 60 bits for precision, as calculated in an above comment
        require(totalSupply() <= type(uint96).max, "total supply exceeded");
        return shares;
    }

    /**
     * Calculate shares received when investing Frankencoins
     * @param investment    ZCHF to be invested
     * @return shares to be received in return
     */
    function calculateShares(uint256 investment) external view returns (uint256) {
        return _calculateShares(zchf.equity(), investment);
    }

    function _calculateShares(uint256 capitalBefore, uint256 investment) internal view returns (uint256) {
        uint256 totalShares = totalSupply();
        uint256 investmentExFees = (investment * 997) / 1000; // remove 0.3% fee
        // Assign 1000 FPS for the initial deposit, calculate the amount otherwise
        uint256 newTotalShares = capitalBefore < MINIMUM_EQUITY || totalShares == 0
            ? totalShares + 1000 * ONE_DEC18
            : _mulD18(totalShares, _cubicRoot(_divD18(capitalBefore + investmentExFees, capitalBefore)));
        return newTotalShares - totalShares;
    }

    /**
     * Redeem the given amount of shares owned by the sender and transfer the proceeds to the target.
     * @return The amount of ZCHF transferred to the target
     */
    function redeem(address target, uint256 shares) external returns (uint256) {
        return _redeemFrom(msg.sender, target, shares);
    }

    /**
     * Like redeem(...), but with an extra parameter to protect against frontrunning.
     * @param expectedProceeds  The minimum acceptable redemption proceeds.
     */
    function redeemExpected(address target, uint256 shares, uint256 expectedProceeds) external returns (uint256) {
        uint256 proceeds = _redeemFrom(msg.sender, target, shares);
        require(proceeds >= expectedProceeds);
        return proceeds;
    }

    /**
     * Redeem FPS based on an allowance from the owner to the caller.
     * See also redeemExpected(...).
     */
    function redeemFrom(
        address owner,
        address target,
        uint256 shares,
        uint256 expectedProceeds
    ) external returns (uint256) {
        _useAllowance(owner, msg.sender, shares);
        uint256 proceeds = _redeemFrom(owner, target, shares);
        require(proceeds >= expectedProceeds);
        return proceeds;
    }

    function _redeemFrom(address owner, address target, uint256 shares) internal returns (uint256) {
        require(canRedeem(owner));
        uint256 proceeds = calculateProceeds(shares);
        _burn(owner, shares);
        zchf.transfer(target, proceeds);
        emit Trade(owner, -int(shares), proceeds, price());
        return proceeds;
    }

    /**
     * Calculate ZCHF received when depositing shares
     * @param shares number of shares we want to exchange for ZCHF,
     *               in dec18 format
     * @return amount of ZCHF received for the shares
     */
    function calculateProceeds(uint256 shares) public view returns (uint256) {
        uint256 totalShares = totalSupply();
        require(shares + ONE_DEC18 < totalShares, "too many shares"); // make sure there is always at least one share
        uint256 capital = zchf.equity();
        uint256 reductionAfterFees = (shares * 997) / 1000;
        uint256 newCapital = _mulD18(capital, _power3(_divD18(totalShares - reductionAfterFees, totalShares)));
        return capital - newCapital;
    }

    /**
     * If there is less than 1000 ZCHF in equity left (maybe even negative), the system is at risk
     * and we should allow qualified FPS holders to restructure the system.
     *
     * Example: there was a devastating loss and equity stands at -1'000'000. Most shareholders have lost hope in the
     * Frankencoin system except for a group of small FPS holders who still believes in it and is willing to provide
     * 2'000'000 ZCHF to save it. These brave souls are essentially donating 1'000'000 to the minter reserve and it
     * would be wrong to force them to share the other million with the passive FPS holders. Instead, they will get
     * the possibility to bootstrap the system again owning 100% of all FPS shares.
     *
     * @param helpers          A list of addresses that delegate to the caller in incremental order
     * @param addressesToWipe  A list of addresses whose FPS will be burned to zero
     */
    function restructureCapTable(address[] calldata helpers, address[] calldata addressesToWipe) external {
        require(zchf.equity() < MINIMUM_EQUITY);
        checkQualified(msg.sender, helpers);
        for (uint256 i = 0; i < addressesToWipe.length; i++) {
            address current = addressesToWipe[i];
            _burn(current, balanceOf(current));
        }
    }
}
```

2.F Minting Hub

The minting hub is a minter contract that serves as a basis to open collateralized positions with the Frankencoin's unique collateralized, oracle-free minting mechanism. The minting hub allows users to open new positions, to clone existing positions, to launch challenges, and to bid on challenges.

```
pragma solidity ^0.8.0;

import "./interface/IERC20.sol";
import "./interface/IReserve.sol";
import "./interface/IFrankencoin.sol";
import "./interface/IPosition.sol";
import "./interface/IPositionFactory.sol";

/**
 * @title Minting Hub
 * The central hub for creating, cloning and challenging collateralized Frankencoin positions.
 * @dev Only one instance of this contract is required, whereas every new position comes with a new position
 * contract. Pending challenges are stored as structs in an array.
 */
contract MintingHub {
    /**
     * Irrevocable fee in ZCHF when proposing a new position (but not when cloning an existing one).
     */
    uint256 public constant OPENING_FEE = 1000 * 10 ** 18;

    /**
     * The challenger reward in parts per million (ppm) relative to the challenged amount, whereas
     * challenged amount if defined as the challenged collateral amount times the liquidation price.
     */
    uint32 public constant CHALLENGER_REWARD = 20000; // 2%

    IPositionFactory private immutable POSITION_FACTORY; // position contract to clone

    IFrankencoin public immutable zchf; // currency
    Challenge[] public challenges; // list of open challenges

    /**
     * Map to remember pending postponed collateral returns.
     * @dev It maps collateral => beneficiary => amount.
     */
    mapping(address collateral => mapping(address owner => uint256 amount)) public pendingReturns;

    struct Challenge {
        address challenger; // the address from which the challenge was initiated
        uint64 start; // the start of the challenge
        IPosition position; // the position that was challenged
        uint256 size; // how much collateral the challenger provided
    }

    event PositionOpened(
        address indexed owner,
        address indexed position,
        address zchf,
        address collateral,
        uint256 price
    );
    event ChallengeStarted(address indexed challenger, address indexed position, uint256 size, uint256 number);
    event ChallengeAverted(address indexed position, uint256 number, uint256 size);
    event ChallengeSucceeded(
        address indexed position,
        uint256 number,
        uint256 bid,
        uint256 acquiredCollateral,
        uint256 challengeSize
    );
    event PostPonedReturn(address collateral, address indexed beneficiary, uint256 amount);

    error UnexpectedPrice();
    error InvalidPos();

    modifier validPos(address position) {
        if (zchf.getPositionParent(position) != address(this)) revert InvalidPos();
        _;
    }

    constructor(address _zchf, address _factory) {
        zchf = IFrankencoin(_zchf);
        POSITION_FACTORY = IPositionFactory(_factory);
    }

    function openPositionOneWeek(
        address _collateralAddress,
        uint256 _minCollateral,
        uint256 _initialCollateral,
        uint256 _mintingMaximum,
        uint256 _expirationSeconds,
        uint64 _challengeSeconds,
        uint32 _annualInterestPPM,
        uint256 _liqPrice,
        uint32 _reservePPM
    ) public returns (address) {
        return
            openPosition(
                _collateralAddress,
                _minCollateral,
                _initialCollateral,
                _mintingMaximum,
                7 days,
                _expirationSeconds,
                _challengeSeconds,
                _annualInterestPPM,
                _liqPrice,
                _reservePPM
            );
    }

    /**
     * Open a collateralized loan position. See also https://docs.frankencoin.com/positions/open .
     * @dev For a successful call, you must set an allowance for the collateral token, allowing
     * the minting hub to transfer the initial collateral amount to the newly created position and to
     * withdraw the fees.
     *
     * @param _collateralAddress        address of collateral token
     * @param _minCollateral     minimum collateral required to prevent dust amounts
     * @param _initialCollateral amount of initial collateral to be deposited
     * @param _mintingMaximum    maximal amount of ZCHF that can be minted by the position owner
     * @param _expirationSeconds position tenor in unit of timestamp (seconds) from ’now’
     * @param _challengeSeconds  challenge period. Longer for less liquid collateral.
     * @param _annualInterestPPM ppm of minted amount that is paid as fee for each year of duration
     * @param _liqPrice          Liquidation price with (36 - token decimals) decimals,
     *                           e.g. 18 decimals for an 18 dec collateral, 36 decs for a 0 dec collateral.
     * @param _reservePPM        ppm of minted amount that is locked as borrower’s reserve, e.g. 20%
     * @return address           address of created position
     */
    function openPosition(
        address _collateralAddress,
        uint256 _minCollateral,
        uint256 _initialCollateral,
        uint256 _mintingMaximum,
        uint256 _initPeriodSeconds,
        uint256 _expirationSeconds,
        uint64 _challengeSeconds,
        uint32 _annualInterestPPM,
        uint256 _liqPrice,
        uint32 _reservePPM
    ) public returns (address) {
        require(_annualInterestPPM <= 1000000);
        require(_reservePPM <= 1000000);
        require(IERC20(_collateralAddress).decimals() <= 24); // leaves 12 digits for price
        require(_initialCollateral >= _minCollateral, "must start with min col");
        require(_minCollateral * _liqPrice >= 5000 ether * 10 ** 18); // must start with at least 5000 ZCHF worth of collateral
        IPosition pos = IPosition(
            POSITION_FACTORY.createNewPosition(
                msg.sender,
                address(zchf),
                _collateralAddress,
                _minCollateral,
                _mintingMaximum,
                _initPeriodSeconds,
                _expirationSeconds,
                _challengeSeconds,
                _annualInterestPPM,
                _liqPrice,
                _reservePPM
            )
        );
        zchf.registerPosition(address(pos));
        zchf.transferFrom(msg.sender, address(zchf.reserve()), OPENING_FEE);
        IERC20(_collateralAddress).transferFrom(msg.sender, address(pos), _initialCollateral);

        emit PositionOpened(msg.sender, address(pos), address(zchf), _collateralAddress, _liqPrice);
        return address(pos);
    }

    /**
     * Clones an existing position and immediately tries to mint the specified amount using the given collateral.
     * @dev This needs an allowance to be set on the collateral contract such that the minting hub can get the collateral.
     */
    function clonePosition(
        address position,
        uint256 _initialCollateral,
        uint256 _initialMint,
        uint256 expiration
    ) public validPos(position) returns (address) {
        IPosition existing = IPosition(position);
        require(expiration <= IPosition(existing.original()).expiration());
        existing.reduceLimitForClone(_initialMint);
        address pos = POSITION_FACTORY.clonePosition(position);
        zchf.registerPosition(pos);
        IPosition(pos).initializeClone(msg.sender, existing.price(), _initialCollateral, _initialMint, expiration);
        existing.collateral().transferFrom(msg.sender, pos, _initialCollateral);

        emit PositionOpened(
            msg.sender,
            address(pos),
            address(zchf),
            address(IPosition(pos).collateral()),
            IPosition(pos).price()
        );
        return address(pos);
    }

    /**
     * Launch a challenge (Dutch auction) on a position
     * @param _positionAddr      address of the position we want to challenge
     * @param _collateralAmount  size of the collateral we want to challenge (dec 18)
     * @param expectedPrice      position.price() to guard against the minter fruntrunning with a price change
     * @return index of the challenge in challenge-array
     */
    function launchChallenge(
        address _positionAddr,
        uint256 _collateralAmount,
        uint256 expectedPrice
    ) external validPos(_positionAddr) returns (uint256) {
        IPosition position = IPosition(_positionAddr);
        if (position.price() != expectedPrice) revert UnexpectedPrice();
        IERC20(position.collateral()).transferFrom(msg.sender, address(this), _collateralAmount);
        uint256 pos = challenges.length;
        challenges.push(Challenge(msg.sender, uint64(block.timestamp), position, _collateralAmount));
        position.notifyChallengeStarted(_collateralAmount);
        emit ChallengeStarted(msg.sender, address(position), _collateralAmount, pos);
        return pos;
    }

    /**
     * Post a bid in ZCHF given an open challenge.
     *
     * @dev In case that the collateral cannot be transfered back to the challenger (i.e. because the collateral token
     * has a blacklist and the challenger is on it), it is possible to postpone the return of the collateral.
     *
     * @param _challengeNumber  index of the challenge as broadcast in the event
     * @param size              how much of the collateral the caller wants to bid for at most
     *                          (automatically reduced to the available amount)
     * @param postponeCollateralReturn To postpone the return of the collateral to the challenger. Usually false.
     */
    function bid(uint32 _challengeNumber, uint256 size, bool postponeCollateralReturn) external {
        Challenge memory challenge = challenges[_challengeNumber];
        (uint256 liqPrice, uint64 phase1, uint64 phase2) = challenge.position.challengeData();
        size = challenge.size < size ? challenge.size : size; // cannot bid for more than the size of the challenge

        if (block.timestamp <= challenge.start + phase1) {
            _avertChallenge(challenge, _challengeNumber, liqPrice, size);
            emit ChallengeAverted(address(challenge.position), _challengeNumber, size);
        } else {
            _returnChallengerCollateral(challenge, _challengeNumber, size, postponeCollateralReturn);
            (uint256 transferredCollateral, uint256 offer) = _finishChallenge(
                challenge,
                liqPrice,
                phase1,
                phase2,
                size
            );
            emit ChallengeSucceeded(address(challenge.position), _challengeNumber, offer, transferredCollateral, size);
        }
    }

    function _finishChallenge(
        Challenge memory challenge,
        uint256 liqPrice,
        uint64 phase1,
        uint64 phase2,
        uint256 size
    ) internal returns (uint256, uint256) {
        // Repayments depend on what was actually minted, whereas bids depend on the available collateral
        (address owner, uint256 collateral, uint256 repayment, uint32 reservePPM) = challenge
            .position
            .notifyChallengeSucceeded(msg.sender, size);

        // No overflow possible thanks to invariant (col * price <= limit * 10**18)
        // enforced in Position.setPrice and knowing that collateral <= col.
        uint256 offer = (_calculatePrice(challenge.start + phase1, phase2, liqPrice) * collateral) / 10 ** 18;
        zchf.transferFrom(msg.sender, address(this), offer); // get money from bidder
        uint256 reward = (offer * CHALLENGER_REWARD) / 1000_000;
        uint256 fundsNeeded = reward + repayment;

        if (offer > fundsNeeded) {
            zchf.transfer(owner, offer - fundsNeeded);
        } else if (offer < fundsNeeded) {
            zchf.notifyLoss(fundsNeeded - offer); // ensure we have enough to pay everything
        }
        zchf.transfer(challenge.challenger, reward); // pay out the challenger reward
        zchf.burnWithoutReserve(repayment, reservePPM); // Repay the challenged part
        return (collateral, offer);
    }

    function _avertChallenge(Challenge memory challenge, uint32 number, uint256 liqPrice, uint256 size) internal {
        if (msg.sender == challenge.challenger) {
            // allow challenger to cancel challenge without paying themselves
        } else {
            zchf.transferFrom(msg.sender, challenge.challenger, (size * liqPrice) / (10 ** 18));
        }

        challenge.position.notifyChallengeAverted(size);
        challenge.position.collateral().transfer(msg.sender, size);
        if (size < challenge.size) {
            challenges[number].size = challenge.size - size;
        } else {
            require(size == challenge.size);
            delete challenges[number];
        }
    }

    /**
     * Returns 'amount' of the collateral to the challenger and reduces or deletes the relevant challenge.
     */
    function _returnChallengerCollateral(
        Challenge memory challenge,
        uint32 number,
        uint256 amount,
        bool postpone
    ) internal {
        _returnCollateral(challenge.position.collateral(), challenge.challenger, amount, postpone);
        if (challenge.size == amount) {
            // bid on full amount
            delete challenges[number];
        } else {
            // bid on partial amount
            challenges[number].size -= amount;
        }
    }

    /**
     * Calculates the current Dutch auction price.
     * @dev Starts at the full price at time 'start' and linearly goes to 0 as 'phase2' passes.
     */
    function _calculatePrice(uint64 start, uint64 phase2, uint256 liqPrice) internal view returns (uint256) {
        uint64 timeNow = uint64(block.timestamp);
        if (timeNow <= start) {
            return liqPrice;
        } else if (timeNow >= start + phase2) {
            return 0;
        } else {
            uint256 timeLeft = phase2 - (timeNow - start);
            return (liqPrice / phase2) * timeLeft;
        }
    }

    /**
     * Get the price per unit of the collateral for the given challenge.
     * @dev The price comes with (36-collateral.decimals()) digits, such that multiplying it with the
     * raw collateral amount always yields a price with 36 digits, or 18 digits after dividing by 10**18 again.
     */
    function price(uint32 challengeNumber) public view returns (uint256) {
        Challenge memory challenge = challenges[challengeNumber];
        if (challenge.challenger == address(0x0)) {
            return 0;
        } else {
            (uint256 liqPrice, uint64 phase1, uint64 phase2) = challenge.position.challengeData();
            return _calculatePrice(challenge.start + phase1, phase2, liqPrice);
        }
    }

    /**
     * Challengers can call this method to withdraw collateral whose return was postponed.
     */
    function returnPostponedCollateral(address collateral, address target) external {
        uint256 amount = pendingReturns[collateral][msg.sender];
        delete pendingReturns[collateral][msg.sender];
        IERC20(collateral).transfer(target, amount);
    }

    function _returnCollateral(IERC20 collateral, address recipient, uint256 amount, bool postpone) internal {
        if (postpone) {
            // Postponing helps in case the challenger was blacklisted or otherwise cannot receive at the moment.
            pendingReturns[address(collateral)][recipient] += amount;
            emit PostPonedReturn(address(collateral), recipient, amount);
        } else {
            collateral.transfer(recipient, amount); // return the challenger's collateral
        }
    }
}
```

2.G Position Factory

The position factory is a helper contract to create and clone positions.

```
pragma solidity ^0.8.0;

import "./Position.sol";
import "./interface/IFrankencoin.sol";

contract PositionFactory {
    /**
     * Create a completely new position in a newly deployed contract.
     * Must be called through minting hub to be recognized as valid position.
     */
    function createNewPosition(
        address _owner,
        address _zchf,
        address _collateral,
        uint256 _minCollateral,
        uint256 _initialLimit,
        uint256 _initPeriod,
        uint256 _duration,
        uint64 _challengePeriod,
        uint32 _annualInterestPPM,
        uint256 _liqPrice,
        uint32 _reserve
    ) external returns (address) {
        return
            address(
                new Position(
                    _owner,
                    msg.sender,
                    _zchf,
                    _collateral,
                    _minCollateral,
                    _initialLimit,
                    _initPeriod,
                    _duration,
                    _challengePeriod,
                    _annualInterestPPM,
                    _liqPrice,
                    _reserve
                )
            );
    }

    /**
     * clone an existing position. This can be a clone of another clone,
     * or an original position.
     * @param _existing address of the position we want to clone
     * @return address of the newly created clone position
     */
    function clonePosition(address _existing) external returns (address) {
        Position existing = Position(_existing);
        Position clone = Position(_createClone(existing.original()));
        return address(clone);
    }

    // github.com/optionality/clone-factory/blob/32782f82dfc5a00d103a7e61a17a5dedbd1e8e9d/contracts/CloneFactory.sol
    function _createClone(address target) internal returns (address result) {
        bytes20 targetBytes = bytes20(target);
        assembly {
            let clone := mload(0x40)
            mstore(clone, 0x3d602d80600a3d3981f3363d3d373d3d3d363d73000000000000000000000000)
            mstore(add(clone, 0x14), targetBytes)
            mstore(add(clone, 0x28), 0x5af43d82803e903d91602b57fd5bf30000000000000000000000000000000000)
            result := create(0, clone, 0x37)
        }
        require(result != address(0), "ERC1167: create failed");
    }
}
```

2.H Collateralized Position

This smart contract represents a collateralized position. There is a separate instance of this contract for each position and each instance belongs to an owner. The owner is the person that created (or cloned) the position and also considered the owner of the provided collateral.

```
pragma solidity ^0.8.0;

import "./utils/Ownable.sol";
import "./utils/MathUtil.sol";

import "./interface/IERC20.sol";
import "./interface/IPosition.sol";
import "./interface/IReserve.sol";
import "./interface/IFrankencoin.sol";

/**
 * @title Position
 * A collateralized minting position.
 */
contract Position is Ownable, IPosition, MathUtil {
    /**
     * Note that this contract is intended to be cloned. All clones will share the same values for
     * the constant and immutable fields, but have their own values for the other fields.
     */

    /**
     * The zchf price per unit of the collateral below which challenges succeed, (36 - collateral.decimals) decimals
     */
    uint256 public price;

    /**
     * Net minted amount, including reserve.
     */
    uint256 public minted;

    /**
     * Amount of the collateral that is currently under a challenge.
     * Used to figure out whether there are pending challenges.
     */
    uint256 public challengedAmount;

    /**
     * Challenge period in seconds.
     */
    uint64 public immutable challengePeriod;

    /**
     * End of the latest cooldown. If this is in the future, minting is suspended.
     */
    uint256 public cooldown;

    /**
     * How much can be minted at most.
     */
    uint256 public limit;

    /**
     * Timestamp when minting can start and the position no longer denied.
     */
    uint256 public immutable start;

    /**
     * Timestamp of the expiration of the position. After expiration, challenges cannot be averted
     * any more. This is also the basis for fee calculations.
     */
    uint256 public expiration;

    /**
     * The original position to help identifying clones.
     */
    address public immutable original;

    /**
     * Pointer to the minting hub.
     */
    address public immutable hub;

    /**
     * The Frankencoin contract.
     */
    IFrankencoin public immutable zchf;

    /**
     * The collateral token.
     */
    IERC20 public immutable override collateral;

    /**
     * Minimum acceptable collateral amount to prevent dust.
     */
    uint256 public immutable override minimumCollateral;

    /**
     * Always pay interest for at least four weeks.
     */
    uint256 private constant MIN_INTEREST_DURATION = 4 weeks;

    /**
     * The interest in parts per million per year that is deducted when minting Frankencoins.
     * To be paid upfront.
     */
    uint32 public immutable annualInterestPPM;

    /**
     * The reserve contribution in parts per million of the minted amount.
     */
    uint32 public immutable reserveContribution;

    event MintingUpdate(uint256 collateral, uint256 price, uint256 minted, uint256 limit);
    event PositionDenied(address indexed sender, string message); // emitted if closed by governance

    error InsufficientCollateral();
    error TooLate();
    error RepaidTooMuch(uint256 excess);
    error LimitExceeded();
    error ChallengeTooSmall();
    error Expired();
    error Hot();
    error Challenged();
    error NotHub();

    modifier alive() {
        if (block.timestamp >= expiration) revert Expired();
        _;
    }

    modifier noCooldown() {
        if (block.timestamp <= cooldown) revert Hot();
        _;
    }

    modifier noChallenge() {
        if (challengedAmount > 0) revert Challenged();
        _;
    }

    modifier onlyHub() {
        if (msg.sender != address(hub)) revert NotHub();
        _;
    }

    /**
     * @dev See MintingHub.openPosition
     */
    constructor(
        address _owner,
        address _hub,
        address _zchf,
        address _collateral,
        uint256 _minCollateral,
        uint256 _initialLimit,
        uint256 _initPeriod,
        uint256 _duration,
        uint64 _challengePeriod,
        uint32 _annualInterestPPM,
        uint256 _liqPrice,
        uint32 _reservePPM
    ) {
        require(_initPeriod >= 3 days); // must be at least three days, recommended to use higher values
        _setOwner(_owner);
        original = address(this);
        hub = _hub;
        zchf = IFrankencoin(_zchf);
        collateral = IERC20(_collateral);
        annualInterestPPM = _annualInterestPPM;
        reserveContribution = _reservePPM;
        minimumCollateral = _minCollateral;
        challengePeriod = _challengePeriod;
        start = block.timestamp + _initPeriod; // at least three days time to deny the position
        cooldown = start;
        expiration = start + _duration;
        limit = _initialLimit;
        _setPrice(_liqPrice);
    }

    /**
     * Method to initialize a freshly created clone. It is the responsibility of the creator to make sure this is only
     * called once and to call reduceLimitForClone on the original position before initializing the clone.
     */
    function initializeClone(
        address owner,
        uint256 _price,
        uint256 _coll,
        uint256 _initialMint,
        uint256 expirationTime
    ) external onlyHub {
        if (_coll < minimumCollateral) revert InsufficientCollateral();
        uint256 impliedPrice = (_initialMint * ONE_DEC18) / _coll;
        _initialMint = (impliedPrice * _coll) / ONE_DEC18; // to cancel potential rounding errors
        if (impliedPrice > _price) revert InsufficientCollateral();
        _setOwner(owner);
        limit = _initialMint;
        expiration = expirationTime;
        _setPrice(impliedPrice);
        _mint(owner, _initialMint, _coll);
    }

    function limitForClones() public view returns (uint256) {
        uint256 backedLimit = (_collateralBalance() * price) / ONE_DEC18;
        if (backedLimit >= limit) {
            return 0;
        } else {
            // due to invariants, this is always below (limit - minted)
            return limit - backedLimit;
        }
    }

    /**
     * Adjust this position's limit to allow a clone to mint its own Frankencoins.
     * Invariant: global limit stays the same.
     *
     * Cloning a position is only allowed if the position is not challenged, not expired and not in cooldown.
     */
    function reduceLimitForClone(uint256 mint_) external noChallenge noCooldown alive onlyHub {
        if (mint_ > limitForClones()) revert LimitExceeded();
        limit -= mint_;
    }

    /**
     * Qualified pool share holders can call this method to immediately expire a freshly proposed position.
     */
    function deny(address[] calldata helpers, string calldata message) external {
        if (block.timestamp >= start) revert TooLate();
        IReserve(zchf.reserve()).checkQualified(msg.sender, helpers);
        _close(); // since expiration is immutable, we put it under eternal cooldown
        emit PositionDenied(msg.sender, message);
    }

    function _close() internal {
        cooldown = type(uint256).max;
    }

    function isClosed() public view returns (bool) {
        return cooldown == type(uint256).max;
    }

    /**
     * This is how much the minter can actually use when minting ZCHF, with the rest being used
     * assigned to the minter reserve or (if applicable) fees.
     */
    function getUsableMint(uint256 totalMint, bool afterFees) external view returns (uint256) {
        if (afterFees) {
            return (totalMint * (1000_000 - reserveContribution - calculateCurrentFee())) / 1000_000;
        } else {
            return (totalMint * (1000_000 - reserveContribution)) / 1000_000;
        }
    }

    /**
     * "All in one" function to adjust the outstanding amount of ZCHF, the collateral amount,
     * and the price in one transaction.
     */
    function adjust(uint256 newMinted, uint256 newCollateral, uint256 newPrice) external onlyOwner {
        uint256 colbal = _collateralBalance();
        if (newCollateral > colbal) {
            collateral.transferFrom(msg.sender, address(this), newCollateral - colbal);
        }
        // Must be called after collateral deposit, but before withdrawal
        if (newMinted < minted) {
            zchf.burnFromWithReserve(msg.sender, minted - newMinted, reserveContribution);
            minted = newMinted;
        }
        if (newCollateral < colbal) {
            withdrawCollateral(msg.sender, colbal - newCollateral);
        }
        // Must be called after collateral withdrawal
        if (newMinted > minted) {
            mint(msg.sender, newMinted - minted);
        }
        if (newPrice != price) {
            adjustPrice(newPrice);
        }
    }

    /**
     * Allows the position owner to adjust the liquidation price as long as there is no pending challenge.
     * Lowering the liquidation price can be done with immediate effect, given that there is enough collateral.
     * Increasing the liquidation price triggers a cooldown period of 3 days, during which minting is suspended.
     */
    function adjustPrice(uint256 newPrice) public onlyOwner noChallenge {
        if (newPrice > price) {
            _restrictMinting(3 days);
        } else {
            _checkCollateral(_collateralBalance(), newPrice);
        }
        _setPrice(newPrice);
        emit MintingUpdate(_collateralBalance(), price, minted, limit);
    }

    function _setPrice(uint256 newPrice) internal {
        require(newPrice * minimumCollateral <= limit * ONE_DEC18); // sanity check
        price = newPrice;
    }

    function _collateralBalance() internal view returns (uint256) {
        return IERC20(collateral).balanceOf(address(this));
    }

    /**
     * Mint ZCHF as long as there is no open challenge, the position is not subject to a cooldown,
     * and there is sufficient collateral.
     */
    function mint(address target, uint256 amount) public onlyOwner noChallenge noCooldown alive {
        _mint(target, amount, _collateralBalance());
    }

    function calculateCurrentFee() public view returns (uint32) {
        uint256 exp = expiration;
        uint256 time = block.timestamp < start ? start : block.timestamp;
        uint256 timePassed = time >= exp - MIN_INTEREST_DURATION ? MIN_INTEREST_DURATION : exp - time;
        // Time resolution is in the range of minutes for typical interest rates.
        return uint32((timePassed * annualInterestPPM) / 365 days);
    }

    function _mint(address target, uint256 amount, uint256 collateral_) internal {
        if (minted + amount > limit) revert LimitExceeded();
        zchf.mintWithReserve(target, amount, reserveContribution, calculateCurrentFee());
        minted += amount;

        _checkCollateral(collateral_, price);
        emit MintingUpdate(_collateralBalance(), price, minted, limit);
    }

    function _restrictMinting(uint256 period) internal {
        uint256 horizon = block.timestamp + period;
        if (horizon > cooldown) {
            cooldown = horizon;
        }
    }

    /**
     * Repay some ZCHF. If too much is repaid, the call fails.
     * It is possible to repay while there are challenges, but the collateral is locked until all is clear again.
     *
     * The repaid amount should fulfill the following equation in order to close the position,
     * i.e. bring the minted amount to 0:
     * minted = amount + zchf.calculateAssignedReserve(amount, reservePPM)
     *
     * Under normal circumstances, this implies:
     * amount = minted * (1000000 - reservePPM)
     *
     * E.g. if minted is 50 and reservePPM is 200000, it is necessary to repay 40 to be able to close the position.
     */
    function repay(uint256 amount) public {
        IERC20(zchf).transferFrom(msg.sender, address(this), amount);
        uint256 actuallyRepaid = IFrankencoin(zchf).burnWithReserve(amount, reserveContribution);
        _notifyRepaid(actuallyRepaid);
        emit MintingUpdate(_collateralBalance(), price, minted, limit);
    }

    function _notifyRepaid(uint256 amount) internal {
        if (amount > minted) revert RepaidTooMuch(amount - minted);
        minted -= amount;
    }

    /**
     * Withdraw any ERC20 token that might have ended up on this address.
     * Withdrawing collateral is subject to the same restrictions as withdrawCollateral(...).
     */
    function withdraw(address token, address target, uint256 amount) external onlyOwner {
        if (token == address(collateral)) {
            withdrawCollateral(target, amount);
        } else {
            uint256 balance = _collateralBalance();
            IERC20(token).transfer(target, amount);
            require(balance == _collateralBalance()); // guard against double-entry-point tokens
        }
    }

    /**
     * Withdraw collateral from the position up to the extent that it is still well collateralized afterwards.
     * Not possible as long as there is an open challenge or the contract is subject to a cooldown.
     *
     * Withdrawing collateral below the minimum collateral amount formally closes the position.
     */
    function withdrawCollateral(address target, uint256 amount) public onlyOwner noChallenge {
        if (block.timestamp <= cooldown && !isClosed()) revert Hot();
        uint256 balance = _withdrawCollateral(target, amount);
        _checkCollateral(balance, price);
        if (balance < minimumCollateral && balance > 0) revert InsufficientCollateral(); // Prevent dust amounts
    }

    function _withdrawCollateral(address target, uint256 amount) internal returns (uint256) {
        if (amount > 0) {
            // Some weird tokens fail when trying to transfer 0 amounts
            IERC20(collateral).transfer(target, amount);
        }
        uint256 balance = _collateralBalance();
        if (balance < minimumCollateral && challengedAmount == 0) {
            // This leaves a slightly unsatisfying possibility open: if the withdrawal happens due to a successful
            // challenge, there might be a small amount of collateral left that is not withheld in case there are no
            // other pending challenges. The only way to cleanly solve this would be to have two distinct cooldowns,
            // one for minting and one for withdrawals.
            _close();
        }

        emit MintingUpdate(balance, price, minted, limit);
        return balance;
    }

    /**
     * This invariant must always hold and must always be checked when any of the three
     * variables change in an adverse way.
     */
    function _checkCollateral(uint256 collateralReserve, uint256 atPrice) internal view {
        if (collateralReserve * atPrice < minted * ONE_DEC18) revert InsufficientCollateral();
    }

    /**
     * Returns the liquidation price and the durations for phase1 and phase2 of the challenge.
     * In this implementation, both phases are always of equal length.
     */
    function challengeData() external view returns (uint256 liqPrice, uint64 phase1, uint64 phase2) {
        return (price, challengePeriod, challengePeriod);
    }

    function notifyChallengeStarted(uint256 size) external onlyHub {
        // Require minimum size. Collateral balance can be below minimum if it was partially challenged before.
        if (size < minimumCollateral && size < _collateralBalance()) revert ChallengeTooSmall();
        if (size == 0) revert ChallengeTooSmall();
        challengedAmount += size;
    }

    /**
     * @param size   amount of collateral challenged (dec18)
     */
    function notifyChallengeAverted(uint256 size) external onlyHub {
        challengedAmount -= size;
        // Don't allow minter to close the position immediately so challenge can be repeated before
        // the owner has a chance to mint more on an undercollateralized position
        _restrictMinting(1 days);
    }

    /**
     * Notifies the position that a challenge was successful.
     * Triggers the payout of the challenged part of the collateral.
     * Everything else is assumed to be handled by the hub.
     *
     * @param _bidder   address of the bidder that receives the collateral
     * @param _size     size of the collateral bid for (dec 18)
     * @return (position owner, effective challenge size in ZCHF, repaid amount, reserve ppm)
     */
    function notifyChallengeSucceeded(
        address _bidder,
        uint256 _size
    ) external onlyHub returns (address, uint256, uint256, uint32) {
        challengedAmount -= _size;
        uint256 colBal = _collateralBalance();
        if (colBal < _size) {
            _size = colBal;
        }
        uint256 repayment = _mulDiv(minted, _size, colBal);
        _notifyRepaid(repayment); // we assume the caller takes care of the actual repayment
        _withdrawCollateral(_bidder, _size); // transfer collateral to the bidder and emit update

        // Give time for additional challenges before the owner can mint again. In particular,
        // the owner might have added collateral only seconds before the challenge ended, preventing a close.
        _restrictMinting(3 days);

        return (owner, _size, repayment, reserveContribution);
    }
}
```

2.I Math Utilities

A contract with some basic mathematical utilities. Most notable is a function to calculate the cubic root using Halley's approximation method as Solidity only supports integer exponents natively.

```
pragma solidity ^0.8.0;

/**
 * @title Functions for share valuation
 */
contract MathUtil {
    uint256 internal constant ONE_DEC18 = 10 ** 18;

    // Let's go for 12 digits of precision (18-6)
    uint256 internal constant THRESH_DEC18 = 10 ** 6;

    /**
     * Cubic root with Halley approximation
     *         Number 1e18 decimal
     * @param _v     number for which we calculate x**(1/3)
     * @return returns _v**(1/3)
     */
    function _cubicRoot(uint256 _v) internal pure returns (uint256) {
        // Good first guess for _v slightly above 1.0, which is often the case in the Frankencoin system
        uint256 x = _v > ONE_DEC18 && _v < 10 ** 19 ? (_v - ONE_DEC18) / 3 + ONE_DEC18 : ONE_DEC18;
        uint256 diff;
        do {
            uint256 powX3 = _mulD18(_mulD18(x, x), x);
            uint256 xnew = _mulDiv(x, (powX3 + 2 * _v), (2 * powX3 + _v));
            diff = xnew > x ? xnew - x : x - xnew;
            x = xnew;
        } while (diff > THRESH_DEC18);
        return x;
    }

    /**
     * Divides and multiplies, with divisor > 0.
     */
    function _mulDiv(uint256 x, uint256 factor, uint256 divisor) internal pure returns (uint256) {
        if (factor == 0) {
            return 0;
        } else if (type(uint256).max / factor > x) {
            return (x * factor) / divisor;
        } else {
            // divide first to avoid overflow
            return x > factor ? (x / divisor) * factor : (factor / divisor) * x;
        }
    }

    function _mulD18(uint256 _a, uint256 _b) internal pure returns (uint256) {
        return (_a * _b) / ONE_DEC18;
    }

    function _divD18(uint256 _a, uint256 _b) internal pure returns (uint256) {
        return (_a * ONE_DEC18) / _b;
    }

    function _power3(uint256 _x) internal pure returns (uint256) {
        return _mulD18(_mulD18(_x, _x), _x);
    }

    function _min(uint256 a, uint256 b) internal pure returns (uint256) {
        return a < b ? a : b;
    }
}
```

2.J Stablecoin Bridge

A minter contract that allows the minting of Frankencoins using a trusted stablecoin with the same reference currency. Separate instances of bridges are required to support multiple bridged coins or to mint beyond the specified limit.

```
\begin{spacing}{0.5}
pragma solidity ^0.8.0;

import "./interface/IERC20.sol";
import "./interface/IERC677Receiver.sol";
import "./interface/IFrankencoin.sol";

/**
 * @title Stable Coin Bridge
 * A minting contract for another Swiss franc stablecoin ('source stablecoin') that we trust.
 * @author Frankencoin
 */
contract StablecoinBridge {
    IERC20 public immutable chf; // the source stablecoin
    IFrankencoin public immutable zchf; // the Frankencoin

    /**
     * The time horizon after which this bridge expires and needs to be replaced by a new contract.
     */
    uint256 public immutable horizon;

    /**
     * The maximum amount of outstanding converted source stablecoins.
     */
    uint256 public immutable limit;
    uint256 public minted;

    error Limit(uint256 amount, uint256 limit);
    error Expired(uint256 time, uint256 expiration);
    error UnsupportedToken(address token);

    constructor(address other, address zchfAddress, uint256 limit_) {
        chf = IERC20(other);
        zchf = IFrankencoin(zchfAddress);
        horizon = block.timestamp + 52 weeks;
        limit = limit_;
        minted = 0;
    }

    /**
     * Convenience method for mint(msg.sender, amount)
     */
    function mint(uint256 amount) external {
        mintTo(msg.sender, amount);
    }

    /**
     * Mint the target amount of Frankencoins, taking the equal amount of source coins from the sender.
     * @dev This only works if an allowance for the source coins has been set and the caller has enough of them.
     */
    function mintTo(address target, uint256 amount) public {
        chf.transferFrom(msg.sender, address(this), amount);
        _mint(target, amount);
    }

    function _mint(address target, uint256 amount) internal {
        if (block.timestamp > horizon) revert Expired(block.timestamp, horizon);
        zchf.mint(target, amount);
        minted += amount;
        if (minted > limit) revert Limit(amount, limit);
    }

    /**
     * Convenience method for burnAndSend(msg.sender, amount)
     */
    function burn(uint256 amount) external {
        _burn(msg.sender, msg.sender, amount);
    }

    /**
     * Burn the indicated amount of Frankencoin and send the same number of source coin to the caller.
     */
    function burnAndSend(address target, uint256 amount) external {
        _burn(msg.sender, target, amount);
    }

    function _burn(address zchfHolder, address target, uint256 amount) internal {
        zchf.burnFrom(zchfHolder, amount);
        chf.transfer(target, amount);
        minted -= amount;
    }
}
\end{spacing}
```

References

Acharya, Viral V et al. (2010). *Regulating Wall Street: The Dodd-Frank Act and the new architecture of global finance.* Vol. 608. Wiley Hoboken, NJ.

Adams, Hayden et al. (2021). *Uniswap v3 Core.* URL: uniswap.org/whitepaper-v3.pdf.

Basel Committee on Banking Supervision (2010). "Basel III: A global regulatory framework for more resilient banks and banking systems". In: *Bank For International Settlements.*

– (Dec. 2017). *High-level summary of Basel III reforms.* bis.org/bcbs/publ/d424_hlsummary.pdf.

– (Dec. 2019a). *CRE Calculation of RWA for Credit Risk.* bis.org/basel_framework/chapter/CRE/22.htm.

– (Dec. 2019b). *RBC Risk-based capital requirements.* bis.org/basel_framework/chapter/RBC/20.htm.

– (2022). *Prudential treatment of cryptoasset exposures.* URL: bis.org/bcbs/publ/d545.htm.

Beck, Roman, Christoph Müller-Bloch, and John Leslie King (2018). "Governance in the blockchain economy: A framework and research agenda". In: *Journal of the association for information systems* 19.10, p. 1.

Bell, Frederick W and Neil B Murphy (1968). "Economies of scale and division of labor in commercial banking". In: *Southern Economic Journal*, pp. 131–139.

Benston, George J, Gerald A Hanweck, and David B Humphrey (1982). "Scale economies in banking: A restructuring and reassessment". In: *Journal of money, credit and banking* 14.4, pp. 435–456. DOI: 10.2307/1991654.

Björk, Tomas (2009). *Arbitrage theory in continuous time.* Oxford university press.

Breeden, Douglas T and Robert H Litzenberger (1978). "Prices of state-contingent claims implicit in option prices". In: *Journal of business*, pp. 621–651.

Briola, Antonio et al. (2023). "Anatomy of a Stablecoin's failure: The Terra-Luna case". In: *Finance Research Letters* 51, p. 103358.

Brunnermeier, Markus K and Dirk Niepelt (2019). “On the equivalence of private and public money”. In: *Journal of Monetary Economics* 106, pp. 27–41.

Chaum, David, Christian Grothoff, and Thomas Moser (2021). “How to issue a central bank digital currency”. In: *SNB Working Papers*. URL: snb.ch/en/mmr/papers/id/working_paper_2021_03.

Chen, Jason, Kathy Fogel, and Kose John (2022). “Understanding the Maker Protocol”. In: *arXiv preprint arXiv:2210.16899*.

Christensen, Rune et al. (2021). *MKR Governance 101*. URL: makerdao.com/governance.

Clark, Jeffrey A (1984). “Estimation of economies of scale in banking using a generalized functional form”. In: *Journal of Money, Credit and Banking* 16.1, pp. 53–68. DOI: 0.2307/1992648.

Clements, Ryan (2021). “Built to Fail: The Inherent Fragility of Algorithmic Stablecoins”. In: *Wake Forest L. Rev. Online* 11, p. 131.

code4rena (2023). *Frankencoin Findings & Analysis Report*. Tech. rep. code4rena. URL: code4rena.com/reports/2023-04-frankencoin.

Efron, Bradley (1992). “Bootstrap methods: another look at the jackknife”. In: *Breakthroughs in statistics*. Springer, pp. 569–593.

Ellinger, Eleunthia Wong et al. (2020). “Skin in the Game: The Transformational Potential of Decentralized Autonomous Organizations”. In: 1. URL: researchgate.net/publication/371831972.

Foo, Chek Yang and Elina MI Koivisto (2004). “Defining grief play in MMORPGs: player and developer perceptions”. In: *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pp. 245–250.

Fritsch, Robin, Marino Müller, and Roger Wattenhofer (2022). “Analyzing voting power in decentralized governance: Who controls daos?” In: *arXiv preprint arXiv:2204.01176*.

Fu, Shange et al. (2023). “Rational Ponzi Game in Algorithmic Stablecoin”. In: *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, pp. 1–6.

Gilbert, R Alton (1984). “Bank market structure and competition: a survey”. In: *Journal of Money, Credit and Banking* 16.4, pp. 617–645. DOI: 10.2307/1992096.

Glosten, Lawrence R and Paul R Milgrom (1985). “Bid, ask and transaction prices in a specialist market with heterogeneously informed traders”. In: *Journal of financial economics* 14.1, pp. 71–100.

Ilo, Oecd (2015). “The labour share in G20 economies”. In: *Report prepared for the G20 Employment Working Group, Antalya, February*, p. 101.

Jarrow, Robert A and Stuart McLean Turnbull (2000). *Derivative securities.* South-Western Pub.

Jentzsch, Christoph (2016). *Decentralized autonomous organization to automate governance.* URL: lawofthelevel.lexblogplatformthree.com/wp-content/uploads/sites/187/2017/07/WhitePaper-1.pdf.

Jorion, Philippe et al. (2010). *Financial Risk Manager Handbook: FRM Part I/Part II.* Vol. 625. John Wiley & Sons.

Lauko, Robert and Richard Pardoe (2021). *Liquity: Decentralized Borrowing Protocol.* URL: docsend.com/view/bwiczmy.

Liu, Jiageng, Igor Makarov, and Antoinette Schoar (2023). *Anatomy of a Run: The Terra Luna Crash.* Tech. rep. National Bureau of Economic Research.

Mackinga, Torgin, Enis Ulqinaku, et al. (2023). *Code Assessment of the Frankencoin Smart Contracts.* Tech. rep. Chainsecurity.

Meisser, Luzius (2017). “The Code is the Model”. In: *International Journal of Microsimulation* 10.3, pp. 184–201.

Morrison, Robbie, Natasha CHL Mazey, and Stephen C Wingreen (2020). “The DAO controversy: the case for a new species of corporate governance?” In: *Frontiers in Blockchain* 3, p. 25.

Palfrey, Thomas R and Howard Rosenthal (1983). “A strategic calculus of voting”. In: *Public choice* 41.1, pp. 7–53.

Scherer, Mathias (2023). *Audit Report Frankencoin.* Tech. rep. Blockbite. URL: github.com/Frankencoin-ZCHF/FrankenCoin/blob/main/audits/blockbite-audit.pdf.

Vogelsteller, Fabian and Vitalik Buterin (2015). *ERC-20: Token Standard.* URL: eips.ethereum.org/EIPS/eip-20.

Wheatley, Martin (2012). “The Wheatley review of LIBOR”. In: *Final report.*

Chapter 3

The Continuous Capital Corporation

Abstract

Traditionally, a capital increase is a one-time event driven by the management of a firm. I explore the dynamics of the opposite, namely letting a company engage in a continuous offering and buyback of its own shares at a zero spread. When doing so according to deterministic and publicly known pricing mechanics, control over the capital level is shifted from the company to the invisible hand of the open market. I derive pricing mechanics that allow a market consisting of rational investors to push an economy consisting of continuous capital corporations to the optimal capital allocation under a wide range of circumstances. These circumstances include unanticipated interest rate changes, technology shocks, as well as input and output price changes.

Keywords: decentralized finance, corporate finance, equilibrium theory

3.1 Introduction

In theory, financial markets play a crucial role in the capital allocation of the real economy. While this might indeed be the case for credit markets and private equity, there is only a weak such connection for public stock markets. The vast majority of capital flows in the stock market are between investors, and only very few between the issuer and the investors. Dow and Gorton (1997) show that as long as there is no direct link between stock markets and the economy, an efficient stock market can guide management decisions to a certain degree through its informational value, but it is not sufficient for economic efficiency.

The emergent field of decentralized finance (DeFi) might change this by enabling more direct channels between issuers and investors. The defining feature of DeFi is the use of blockchain technology and smart contracts to transparently and deterministically enforce market rules, enabling novel market setups without traditional intermediaries. A successful example of such a new type of market enabled by smart contracts is the decentralized exchange Uniswap (Adams et al., 2021). Uniswap works without any centralized authority and is based on completely deterministic pricing rules. Anyone can take part in the market making by passively contributing to so-called liquidity pools. Also, anyone can participate in its economic success by buying UNI tokens, a tailor-made cryptocurrency for the governance of the system and the allocation of profits. The daily trading volume on Uniswap is about half a billion dollars and the UNI tokens in circulation enjoy a market capitalization of over two billion dollars at the time of writing (DefiLlama, 2023; Coinmarketcap, 2023). Maybe smart contracts could not only be used to enable new types of markets for speculative cryptocurrencies, but also improve equity financing for real-world companies?

When a company uses a smart contract to kickstart a market for its own shares, providing the necessary liquidity itself, it becomes what I call a *continuous capital corporation*. Financially, it engages in the continuous issuance and buyback of its own shares at a minimal spread. This gives the markets much more direct control over the capital allocation in the real economy. Buying shares of a firm injects capital, and selling the same shares extracts capital from the firm. The main challenge when creating a continuous capital corporation is the design of the firm's pricing function so that a free and efficient market consisting of rational investors automatically drives the capital allocation between firms toward the efficient level, without explicit financing operations initiated by the firm's management.

The pricing mechanism proposed herein fulfills three important properties. First, it is shown to lead to the efficient capital allocation in a market with rational investors, at least in the stationary case. Second, it provides a price point at which, under a wide range of circumstances, both the sale of new shares as well as the repurchase of old shares is Pareto-improving when transacting with an informed investor. These circumstances include interest rate changes, technology shocks, and under the assumption of Cobb-Douglas production also input price changes. Third, the pricing mechanism is simple enough to support automated market-making with a blockchain-based smart contract, obsoleting financial intermediaries that are traditionally needed to run financial markets.

Like Tinn (2017), I do not focus on efficiency gains enabled by blockchain technology but on the new possibilities it enables. Also, similar to Tinn and unlike authors in the corporate finance literature such as Biais, Mariotti, and Rochet (2013), I disregard agency issues and assume symmetric information.

This paper is structured as follows: after introducing the reader to the technical and legal background in section 3.2, I state the problem to be tackled in section 3.3. Section 3.4 presents an economic model of equity financing through the continuous issuance and repurchase of shares and derives a pricing function for the continuous capital corporation. The model is then extended in section 3.5 and the market interactions of the continuous capital corporation compared to traditional market making in section 3.6. Finally, I conclude with section 3.7.

3.2 Background

This section provides some background about the technical and legal developments that enable the continuous capital corporation.

3.2.1 Technical Developments

Since Nakamoto (2008) invented Bitcoin, the underlying blockchain technology has been improved by various others such that it can today not only serve as a payment system but as a general-purpose, unstoppable "Internet-computer". The most popular general-purpose blockchain is Ethereum as formally specified in Wood et al. (2014). It comes with its own programming language that allows anyone to formulate so-called smart contracts and then deploy them to the system for a fee, paid to an anonymous collective of system operators. Smart contracts are small computer programs anyone can interact with in the ways their author defined. Since the underlying system is completely decentralized, there is no authority that could ever stop a smart contract unless the author explicitly added a function to do so. Thanks to these properties, systems like Ethereum provide reliable foundations for digital ownership registries that keep track of financial assets. The process of registering securities in a smart contract is called *tokenization*, and the individual securities are then often referred to as *security tokens*. Unlike other digital assets, such tokens cannot be copied and are thus a suitable replacement for physical paper certificates. No centralized securities depository or other intermediary is necessary to issue security tokens, enabling disintermediation of financial markets. The resulting ecosystem can be seen as an "Internet of Finance" and is often referred to as *Decentralized Finance* or short *DeFi*.

Bypassing the middle-men and red tape, Decentralized Finance lowers the barriers to enter financial markets. Whereas a traditional IPO can cost legal and other fees in the seven-digit range, it is possible for a firm to create a small market for its own shares for a few hundred Francs using blockchain technology. Besides enabling the trading of shares, a blockchain could also enforce the drag-along clause of the shareholder agreement or automate other corporate actions that previously required considerable paperwork. Thanks to these efficiency gains, it becomes feasible for small companies to have hundreds of shareholders without losing control over shareholder relations.

Of particular interest in our context are smart contracts that perform fully automated market making for a pair of tokens. Existing examples include Bancor (Hertzog, Guy Benartzi, and Galia Benartzi, 2023), Curve (Egorov, 2021) and Uniswap (Adams et al., 2021). With Uniswap, anyone is free to contribute to the capital reserves of the market maker and will, in turn, proportionally participate in its gains and losses. A detailed discussion of the properties of Uniswap has been published by Angeris et al. (2019). An important source of inspiration for the continuous capital corporation was Rosenfeld (2017), where I first encountered the idea of continuously creating and redeeming tokens for the purpose of market-making.

3.2.2 Legal Developments

On the legal side, I want to point to three notable local developments that help to make the continuous capital corporation possible under Swiss law.

First, new accounting rules have come into force in 2013 that reclassified a firm's transactions with its own shares. Previously, buying its own shares low and selling them high would have caused a taxable profit. Recognizing that this is economically equivalent to a buyback and reissuance and following international accounting standards, it is now possible to book transactions with own shares in a profit-neutral way (Schnell Luchsinger and Montavon, 2018; Schweizerisches Bundesgericht, 2019).

Second, Switzerland has introduced a so-called *capital band*, allowing the general assembly of a corporation to authorize the board of directors to issue new or destroy old shares much more easily, creating up to 50% additional shares or taking up to 50% of the old shares out of circulation. This is still not as flexible as the rules of the United States and many other European countries, but a significant step towards enabling a higher frequency of capital adjustments (Forstmoser and Küchler, 2020).

Third, the Swiss parliament has passed a law to provide explicit legal foundations for tokenized securities (Swiss Federal Council, 2020). Before, this was only possible on a contractual basis (Crone, Monsch, and Meisser, 2019). But having a law that essentially says "the token *is* the share" provides a much higher level of legal certainty. While, for example, Germany is also warming up to the topic, Switzerland seems to be ahead of most other European countries (Deutsche Bundesregierung, 2020).

Considerations on how blockchain technology could impact corporate governance can be found in Wagner (2017).

3.3 Problem Statement

Traditionally, both the issuance of new shares as well as the repurchase of old shares are one-time decisions driven by the management of a firm. But when a firm does its own market making using a smart contract, thereby engaging in the deterministic and continuous issuance and repurchase of its own shares at a narrow spread, this is turned around, and the capital allocation decision is put into the invisible hand of the free market. The question that this paper aims to answer is: at which price should a continuous capital corporation offer and repurchase its shares in order to support the efficient capital allocation?

The pricing function shall fulfill the following three criteria:

1. **Efficiency**: In a competitive market with rational investors, the efficient outcome should be reached. It must not be possible for market participants to exploit the firm's market making. The value created for the shareholders when investors buy newly issued shares should outweigh the dilution effect. The capital outflow when repurchasing shares should be compensated for by

an according increase in the value per outstanding share. Under these conditions, all transactions will be Pareto-improving as they will increase both the wealth of the existing shareholders as well as that of the rationally acting counterparty.

2. **Attractiveness**: The efficient outcome should be an attractive equilibrium reachable in incremental steps with each marginal purchase or sale of a share making sense in itself. In more colloquial terms, I want to rule out situations in which buying two shares would make the buyer better off than before, but buying only one share does not. This ensures that each self-confirming equilibrium is also an efficient equilibrium and allows agents to follow what computer scientists call a *greedy algorithm* to find the efficient outcome (Cho and Sargent, 2016).

3. **Simplicity**: It should be feasible to fully automate the market making, for example, in a blockchain-based smart contract that mechanically applies the pricing function. This implies that the price must only depend on easily observable variables and that the function should be of low computational complexity.

3.4 Basic Model

To build an economic model of the continuous capital corporation, we first need a model of equity financing. Most classic models avoid the explicit modeling of share issuance and resort to a workaround instead. For example, Diamond (1967) proposes to model equity financing by separating it into a first step of transactions between shareholders and a second step of all shareholders adding capital to the company in proportion to their holdings, which essentially constitutes a negative dividend. Hens and Elmiger (2019) is a notable exception. They show how equity issuance can be explicitly modelled in a general equilibrium setting. In their model, households treat shares like bonds, namely seeing them as a security with a fixed coupon, disregarding dilution and other secondary effects their buying of newly printed shares might have.

For the purposes of this paper, I will depart from the classic approach by letting the investors know about the impact of buying newly issued shares or returning existing shares to the company. These effects are two-fold: first, there is a dilution effect. The more shares there are, the smaller the dividend per share. Second, there is a productivity impact. The more shares the firm sells, the higher its capital, and therefore also, its profits. Unlike in a traditional Walrasian market or the setup of Hens and Elmiger (2019), where the market participants take their decisions solely based on market prices, the investors in the present model can see into the firms and gauge the profitability impact of buying shares.

Based on these premises and the assumptions outlined below, this section derives the participation constraint of the firm acting in the interest of its existing shareholders as well as that of new investors. I show that the efficient capital allocation is reached in equilibrium when following these constraints. The participation constraint of the firm is later used to provide a pricing function for the market making by the firm. By definition, it provides the point at which the shareholders are indifferent about a marginal issuance or repurchase of shares.

3.4.1 Assumptions

To start, I assume a production function $f(K)$ that depends on capital alone, and that satisfies the Inada conditions ($f(0) = 0$, $f'(K) \geq 0$, $f''(K) \leq 0$). Time is split into discrete steps, with the fruits of production from capital K_t at time t becoming available in the subsequent period $t + 1$. Later on, the production function will be extended to include other inputs and technology. In the absence of other inputs and debt, there are no costs and the firm's profits only depend on capital: $\pi(K) = f(K)$. With equity financing, the capital costs do not appear in the accounting and do not reduce profits. The 'costs' of raising capital are hidden in the dilution that all existing shareholders suffer from when new shares are issued. To account for that, firms should not maximize profits $\pi(K)$, but the wealth of their current shareholders as later expressed in equation 3.2.

Unlike in reality, the number of outstanding shares θ is a continuous variable. The price of share number θ is given by the pricing function $p(\theta)$. It provides the price at which the continuous capital corporation issues and repurchases marginal units of a share. Its integral $P(\theta) = K(\theta)$ returns the total amount of capital raised through the net issuance of θ shares. In the case of a constant price $p(\theta) = p$, capital raised is trivially $K(\theta) = p\theta$. The pricing function is assumed to be positive: $p(\theta) \geq 0$, and increasing: $p'(\theta) \geq 0$. It is implied that the only way of adding or reducing capital is through the issuance or repurchase of shares. Profits and losses are directly attributed to the shareholders and do not affect capital K.

The pricing function $p(\theta)$ is path independent. The price is fully determined by the number of outstanding shares, regardless of the path taken to reach this number. Path independence guarantees that there is no sequence of trades that would allow an attacker to exploit the market maker as long as the sequence starts and ends at the same price.

The internal valuation of the firm is defined as:

$$V_i(K) = V_i(P(\theta)) = \theta p(\theta) \tag{3.1}$$

It denotes the valuation at which the firm is currently willing to issue new or repurchase existing shares. It is fully determined by the pricing function and the outstanding number of shares.

As long as the market is not in equilibrium, the internal valuation can differ from the external market valuation, which is denoted $V_e(K, r)$ and typically depends on the interest rate and other external variables or anticipated events. To anchor the optimization problem, I assume that there always is an alternative investment opportunity that pays interest rate r and that all market participants except the firm can borrow as much capital as they want, also at interest rate r. The only way for the firm to increase or decrease its capital under this model is through the issuance or repurchase of its own shares. It can neither borrow nor retain earnings.

A notable consequence of having a pricing function instead of a constant price is that it allows the firm to issue different shares at different prices within the same period. This generalization allows the firm to preserve path independence and helps to fulfill the attractiveness requirement. At the same time, this also means that we are not in a traditional market setting in which the law of one price can be assumed to hold. Such an assumption would be questionable anyway since the traded good is not uniform. The n-th issued share represents something else (namely $\frac{1}{n}$ of the company at the time of issuance) than the next issued share (namely $\frac{1}{n+1}$ of the company at the time of issuance). Consequently, models that deal with the explicit issuance of new shares should not be built on the assumption that the law of one price holds.

If the law of one price does not hold, the order of purchases within a period starts to matter for the individual market participants. However, it does not matter for the overall economic outcome, and the latter is what we are interested in.

3.4.2 The Firm's Optimization Problem

At each point in time t, the firm maximizes the profits attributable to its existing shareholders, who hold θ_{t-1} shares. Given pricing function $p(\theta)$, it chooses the right amount of outstanding shares θ_t at each point in time t.

$$\max_{\theta_t} \frac{\theta_{t-1}}{\theta_t}\,\pi(K(\theta_t)) = \max_{\theta_t} \frac{\theta_{t-1}}{\theta_t}\,f(P(\theta_t)) \tag{3.2}$$

The more shares are issued, the more capital is available and the higher are the profits of the firm. At the same time, issuing more shares also dilutes existing shareholders, decreasing their profit share. The optimal θ, where these two effects are in balance, is reached when:

$$\frac{d}{d\theta_t}\frac{\theta_{t-1}}{\theta_t}\,f(P(\theta_t)) = 0 \tag{3.3}$$

This results in:

$$\theta\,p(\theta) = \frac{f(P(\theta))}{f'(P(\theta))} \tag{3.4}$$

This is a maximum if the second derivative at this point is negative. This is the case as long as:

$$\frac{p'(\theta)}{p(\theta)^2} < -\frac{f''(P(\theta))}{f'(P(\theta))} \tag{3.5}$$

The right side of this inequality is positive, thanks to the Inada conditions. The inequality holds for constant prices (implying $p'(\theta) = 0$) and more generally for prices $p(\theta)$ that do not grow too quickly, whereas the meaning of 'quickly' depends on the shape of the production function.

Noting that $\theta p(\theta)$ corresponds to the internal valuation of the firm as defined in equation 3.1, one can conclude that choosing the optimal θ implies:

$$V_i(\theta) = \theta\, p(\theta) = \frac{f(P(\theta))}{f'(P(\theta))} \tag{3.6}$$

3.4.3 The Investor's Optimization Problem

The aim of this section is to derive the investor's valuation function. It provides the current valuation of the firm from the point of view of a marginal investor and is based on potential capital gains, dividends, and the return r of the outside option. Given that capital gains and dividends are fully determined by the number of outstanding shares θ_t at each point in time t, this section shows that the external valuation function looks as follows:

$$V_e(\theta_t, \theta_{t+1}, r) = \frac{p(\theta_{t+1}) - p(\theta_t)}{r}\theta_t + \frac{f(P(\theta_t))}{r} \tag{3.7}$$

This valuation function is shown to hold for any increasing utility function and any discount rate. In equilibrium, it coincides with the valuation function of a strategic investor. In contrast to the strategic investor, the marginal investor does not consider the impact of buying or selling a marginal share on the value of other shares the investor already holds or might hold in the future.

Investors maximize lifetime utility with discount rate β and a utility function $U(c)$ with $U'(c) > 0$:

$$U = \sum_t \beta^t U(c_t)$$

At each point in time, the investor holds ϵ_t shares, whereas θ_t denotes the shares held by everyone else. As an alternative to investing his wealth W_t in shares, the investor can also deposit it with interest rate r. Labeling capital $K_t = P(\theta_t + \epsilon_t)$ and investment $I_t(x) = P(\theta_t + x) - P(\theta_t)$, the amount available for consumption at time $t + 1$ is determined by the decisions taken in the previous period and captured by the following term:

$$c_{t+1} = I_{t+1}(\epsilon_t) + \frac{\epsilon_t}{\theta_t + \epsilon_t} f(K_t) + (1 + r)(W_t - I_t(\epsilon_t)) - W_{t+1}$$

It consists of the current value of the shares bought in the previous period $I_{t+1}(\epsilon_t)$, the dividend income $\frac{\epsilon_t}{\theta_t+\epsilon_t}f(K_t)$, the part of the previous periods savings that was not used to buy shares and the interest on these savings $(1+r)(W_t-I_t(\epsilon_t))$, minus the wealth W_{t+1} put aside for the next period. Taking the derivative of lifetime utility with respect to one particular ϵ_t, all but one summand drop out and one obtains the following first-order condition:

$$\frac{dU}{d\epsilon_t} = \beta^{t+1}U'(c_{t+1})\frac{d}{d\epsilon_t}c_{t+1} = 0$$

Since $U'(x) > 0$ and $\beta > 0$ by assumption, this implies that $\frac{d}{d\epsilon_t}c_{t+1}$ must be zero, leading to:

$$p(\theta_{t+1}+\epsilon_t) + \frac{\theta_t}{(\theta_t+\epsilon_t)^2}f(K_t) + \frac{\epsilon_t}{\theta_t+\epsilon_t}f'(K_t)p(\theta_t+\epsilon_t) = (1+r)p(\theta_t+\epsilon_t)$$

In the case of the single strategic investor with all other shareholders staying passive, $\theta_t = \theta_o$ can be assumed constant. Under these conditions, the above condition simplifies to:

$$\theta_o f(K_t) + (\theta_o+\epsilon_t)p(K_t)\epsilon_t f'(K_t) = (\theta_o+\epsilon_t)^2 rp(K_t)$$

This implies that future prices do not matter for the strategic investor. Why is that? Wouldn't anticipated price changes pose an opportunity to make capital gains? No. Given a path independent pricing function and our assumption of the strategic investor being the only active investor, any such gains would evaporate when trying to realize them. The strategic investor would be playing a zero-sum game with herself.

The first point of interest is where the strategic investor and the firm both have no incentive to buy or sell, thereby denoting a market equilibrium. This is found by plugging equation 3.6 into the strategic investor's first order condition, replacing $(\theta_o+\epsilon_t)p(K_t)$ with $\frac{f(K_t)}{f'(K_t)}$. After simplifying $\theta_o+\epsilon_t = \theta$, this yields the strategic investor's valuation function:

$$V_s(\theta, r) = \frac{f(P(\theta))}{r} \tag{3.8}$$

In order to verify that this point represents a maximum (and not a minimum) for the strategic investor, we take the second derivative of the maximization problem with regards to ϵ_t, evaluate it at the same point, and require it to be negative, obtaining:

$$\frac{\theta_t}{\theta_t + \epsilon_t}(f''(K_t)p(K_t)^2 + f'(K_t)p'(K_t)) < rp'(K_t)$$

Considering the expression in the brackets, one finds that this inequality holds when equation 3.5 is fulfilled, making equation 3.5 a sufficient condition for making the point at which the two valuation functions V_s and V_f meet a maximum for both the strategic investor and the firm.

Let us now turn to the case of a marginal investor and the question of when it makes sense to buy (or sell) a marginally small amount of shares. This time, I allow the amount of shares held by the other shareholders to vary over time. The intratemporal optimization of consumption c_t tells us that buying ϵ_t shares with an investment $I(\theta_t, \epsilon_t) = P(\theta_t + \epsilon_t) - P(\theta_t)$ is worthwhile for a new shareholder as long as the capital gains from selling the ϵ_t shares at $t + 1$ and the earned dividends are at least as high as the interest income that could be earned otherwise:

$$I(\theta_{t+1}, \epsilon_t) - I(\theta_t, \epsilon_t) + \frac{\epsilon_t}{\theta_t + \epsilon_t}f(K_t) \geq rI(\theta_t, \epsilon_t)$$

Since this is satisfied with equality at $\epsilon_t = 0$, we know that it is satisfied for all values of ϵ_t if its derivative is also satisfied for all values of ϵ_t. The derivative with respect to ϵ_t is:

$$p(\theta_{t+1}) - p(\theta_t + \epsilon_t) + \frac{\theta_t}{(\theta_t + \epsilon_t)^2}f(P(\theta_t + \epsilon_t)) + \frac{\epsilon_t}{\theta_t + \epsilon_t}(...) \geq rp(\theta_t + \epsilon_t)$$

Letting ϵ_t approach zero, we recognize the following condition in the limit:

$$p(\theta_{t+1}) - p(\theta_t) + \frac{f(P(\theta_t))}{\theta_t} \geq p(\theta_t)r$$

This condition says that the capital gains per share and the dividends per share must outweigh the opportunity cost of not being able to earn interest on the price of a share. It can also be expressed in terms of valuation, saying that the valuation of the company must not exceed the value of the discounted dividends and price appreciation, leading us to equation 3.7, which is a more general version of the strategic investor's valuation function 3.8.

3.4.4 Efficiency

The equilibrium is reached when the firm and the investors are indifferent between buying and selling additional shares. At this point, their valuations match:

$$V_e(K, r) = \frac{p(\theta_{t+1}) - p(\theta_t)}{r}\theta_t + \frac{f(K)}{r} = \frac{f(K)}{f'(K)} = V_i(K)$$

Stationary Case

For the stationary case with $\theta_{t+1} = \theta_t$, the market equilibrium is reached when the marginal return on capital matches the interest rate: $f'(K) = r$. This is the efficient capital allocation. Apparently, our setup leads to a Pareto-efficient outcome even if everyone only has their self-interest in mind, at least in the stationary case. Furthermore, it is not necessary to have a strategic investor to reach the equilibrium. The equilibrium can be reached incrementally with a series of marginally small investments that pay off on their own, as required in the problem statement.

Figure 3.4.1 shows how the two valuations depend on the capitalization of an example firm with production function $f(K) = K^{0.66}$. As long as the firm is selling shares below the valuation of the investor, the investor keeps buying. An additional example with Cobb-Douglas production $f(K) = K^{\alpha}$ can be found in appendix 3.A.

An attentive reader might have noticed that I only derived the incentive constraints for two extreme cases of investors: a sole strategic investor and a marginally small investor. Without providing a rigorous proof, I argue that all other cases fall in between the two extremes.

As an example, consider the situation illustrated in figure 3.4.1 and assume a current level of capital $K = 500$. At this level, the modeled marginal investor is willing to buy shares at a valuation of roughly 1200. In contrast, the strategic investor would be willing to invest 1500 units of capital at a valuation of up to 3000, anticipating the value of the company once the optimal level of capital $K^* = 2000$. Everyone else would be willing to make a bid in between. For example, an investor

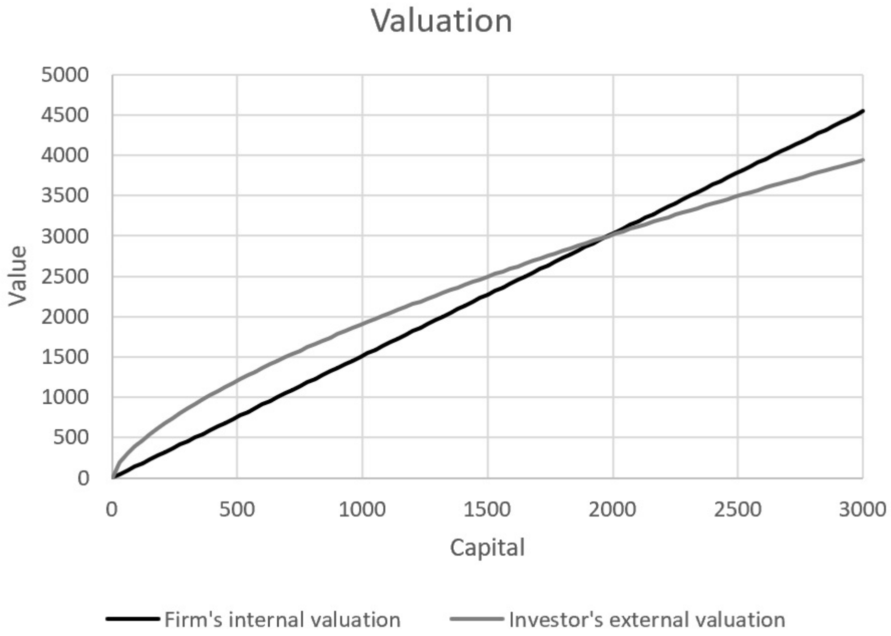


Figure 3.4.1: Equilibrium. Illustration of how the internal valuation $V_i(K)$ and the external valuation $V_e(K)$ depend on the capital K of the firm, with $f(K) = K^{0.66}$, $r = 0.05$ and equilibrium $K^* \approx 2000$. For $K < K^*$, $V_i(K) < V_e(K)$ holds, leading the marginal investor to buy, and vice versa for selling.

investing 1000 units of capital would be willing to invest that amount at a valuation of almost 2000, in between the maximum acceptable valuation of the marginal and the strategic investor.

3.4.5 Outstanding Shares and Their Price

One might wonder how many shares $\theta(K)$ the market participants are getting for the invested capital K. Anchoring the number of issued shares at $\theta(K_0) = 1$, function $\theta(K)$ is:

$$\theta(K) = \int_{K_0}^{K} \frac{1}{p(k)} dk + \theta(K_0) = \int_{K_0}^{K} \frac{f'(k)}{f(k)} \theta(k) dk + 1 = \frac{f(K)}{f(K_0)} \tag{3.9}$$

This result can be verified by substituting $\theta(k)$ with the result. In combination with the internal valuation function 3.6, this implies that the price per share depends on capital as follows:

$$p(K) = \frac{f(K_0)}{f'(K)}$$

Anticipated Price Changes

So far, we have seen that the efficient outcome is reached in the stationary case, with interest rates and the parameters of the production function being constant. In the next section, I will show that the outcome stays efficient under a wide range of unanticipated shocks. However, one major weakness of the model is that it can yield inefficient capital allocations in the face of anticipated price changes from one period to the next.

In the case of an anticipated price increase with $\theta_{t+1} > \theta_t$ and therefore $p(\theta_{t+1}) > p(\theta)$, the investors will over-invest, injecting more capital than necessary into the firm. Similarly, an anticipated price decrease leads to under-investment. Figure 3.4.2 illustrates an example with a series of anticipated technology shocks.

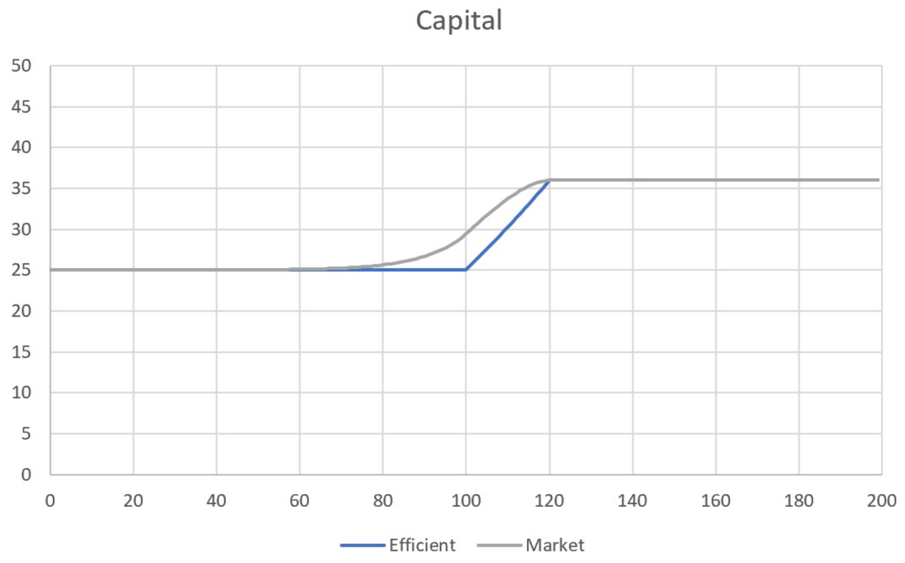


Figure 3.4.2: Temporary over-investment. Plotting capital K on the y-axis against time on the x-axis in a scenario with an anticipated and gradual increase of the technology factor A from 1.0 to 1.2 over the course of 20 periods starting at $t = 100$, one can observe that the continuous capital corporation and marginal investors following their self-interest will coordinate on a temporary over-allocation of capital (grey line) that exceed the efficient allocation (blue line). This example sets interests at $r = 0.1$ and assumes a production function $f(K) = A\sqrt{K}$.

A simple way to restore efficiency would be to allow the firm to invest excess capital at the same interest rate r as everyone else (or to borrow at that rate in the case of under-investment). This, however, would go against the article's premise that debt financing is not available and the idea of putting capital allocation into the invisible hands of the market.

Another argument one could make is that long-term anticipated price changes are rather rare and if there are any, the company might be the first to know and therefore be able to adjust its price before the market reacts. However, this would erode the deterministic nature of the pricing function.

I have a number of further ideas for how this problem could be alleviated, but for now, I leave it unresolved. After all, it is just of temporary nature. As soon as the anticipated change has materialized, the efficient capital allocation is reached again. Before spending too much time on elaborate schemes to counter this temporary problem, I'd like to turn our attention to how robust the pricing function is under various model extensions.

3.5 Extensions and Robustness

The outstanding property of valuation function 3.6 is its robustness against a wide range of shocks. The robustness is owed to its simplicity and the fact that it only depends on internal observables, namely capital K and the shape of the production function $f(K)$. The valuation function does not budge in the presence of technology shocks or interest rate changes. When extending the function to include additional input goods in the style of Cobb and Douglas, it is even indifferent to input price changes! This makes our valuation function an ideal candidate for fully automated market making. A numerical example of what happens during an interest rate change can be found in appendix 3.B. Finally, some considerations on how the model can be adjusted to more realistic scenarios in which earnings (and losses) are retained are made in section 3.5.3, introducing *capital drift*.

3.5.1 Technology Shocks

Valuation function 3.6 is clearly robust against interest rate shocks, as r does not even enter the equation. But what about technology shocks?

To investigate this, I extend the production function to probabilistic technology shocks of the form proposed by Diamond (1967):

$$y(X, K) = g(X)f(K)$$

Here, y is the stochastic output of a firm that depends on a random variable X and a capital level K. As the stochastic effect is multiplicative, it drops out of the equation:

$$V_i(X, K) = \frac{y(X, K)}{y'(X, K)} = \frac{g(X)f(K)}{g(X)f'(K)} = \frac{f(K)}{f'(K)}$$

Therefore, the market maker does not need to adjust its price when confronted with technology news. An exception is technological changes that impact the shape of $f(K)$, for example, a change in α with Cobb-Douglas production $f(K) = K^\alpha$. Nonetheless, it is great to see that the valuation function V_i is robust against all kinds of multiplicative changes.

3.5.2 Input Price Changes

So far, we have assumed a simplistic production function $f(K)$ that only takes capital as input, resulting in profits $\pi(K) = f(K)$. Now, I extend the model with an additional input good x that costs price p_x and impose Cobb-Douglas production. Besides having nice mathematical properties, Cobb-Douglas production functions have an economically plausible shape (Jones, 2005). In that case, profits are:

$$\pi(K, x) = f(K, x) - p_x x = K^\alpha x^\beta - p_x x$$

The price of the output good has been normalized to 1 without loss of generality. It is assumed that $\alpha > 0$, $\beta > 0$ and $\alpha + \beta < 1$.

Given capital K and price p_x, the firm chooses x^* to maximize profits, which happens at $\frac{d}{dx} f(K, x) = p_x$, leading to:

$$x^* = \beta^{\frac{1}{1-\beta}} K^{\frac{\alpha}{1-\beta}} p_x^{\frac{1}{\beta-1}}$$

Plugging the optimal x^* back into the profit function and defining $b = \beta^{\frac{1}{1-\beta}}$, $\gamma = \frac{\alpha}{1-\beta}$, and $\delta = \frac{\beta}{\beta-1}$ reveals:

$$\pi(K, x^*) = K^{\frac{\alpha}{1-\beta}} p_x^{\frac{\beta}{\beta-1}} \beta^{\frac{1}{1-\beta}} (\frac{1}{\beta} - 1) = K^\gamma p_x^\delta b$$

with $0 < \gamma < 1$, $\delta < 0$ and $b > 0$. Both p and b drop out when applying the internal valuation function:

$$V_i(K) = \frac{\pi(K, x^*)}{\pi'(K, x^*)} = \frac{K^\gamma}{\gamma K^{\gamma-1}} = \frac{K}{\gamma}$$

So at least for Cobb-Douglas production, our valuation function is indifferent to input or output price shocks. Adding further input factors would not change that either. Generally, when having a production function of the form $f(K, X) = K^{\alpha_0} \prod x_i^{\alpha_i}$ with many input factors, the valuation function is:

$$V_i(K) = \frac{1}{\alpha_0} K - \frac{\sum \alpha_i}{\alpha_0} K + K \tag{3.10}$$

For the special case of constant returns to scale ($\sum \alpha_i = 1$), this simplifies to $V_i(K) = K$. For the special case where K is the only input factor, it simplifies to $V_i(K) = \frac{1}{\alpha} K$.

Interestingly, equation 3.10 could be used to argue that there is a direct connection between the returns to scale of the production function of a company and its price-to-book ratio. It implies that a firm with returns to scale α should in the absence of debt have a price-to-book ratio of $\frac{1}{\alpha}$. It might be worthwhile to explore this relation more deeply in a separate publication.

3.5.3 Capital Drift

The basic model implies that all profits magically move from the firm to the shareholders as they happen. In practice, profits accrue within the company before they are paid out in a periodic dividend to the shareholders. Economically equivalent, but more flexible, are share buyback programs. In the case of a continuous capital corporation, a buyback program can be implemented by letting the firm continuously adjust the price at which it sells and repurchases its own shares, thereby setting the incentives for capital inflows or outflows in accordance with the accrued profit or loss. This again moves control from the firm to the market.

When a firm with capital K_t makes profits π_t without the hitherto assumed immediate payout to the shareholders, its capital increases to K_{t+1} at the beginning of the next period. This, in turn, increases the valuation according to the internal valuation function $V_i(K)$, but not the number of shares, leading to inconsistency $V_i(K_t) = p(\theta_t)\theta_t = p(\theta_{t+1})\theta_{t+1} \neq V_i(K_{t+1})$ as $\theta_{t+1} = \theta_t$. This can be corrected by letting the pricing function depend on time t, extending it to $p(\theta, t)$ such that:

$$\frac{p(\theta, t+1)}{p(\theta, t)} = \frac{V_i(K_{t+1})}{V_i(K_t)}$$

In the case of Cobb-Douglas production, this simply implies that the share price should be increased in proportion to the accrued capital:

$$\frac{p(\theta, t+1)}{p(\theta, t)} = \frac{K_t + \pi_t}{K_t}$$

For example, a continuous capital corporation with Cobb-Douglas production and a return on equity of $\pi/K = 10\%$ should let the price $p(\theta, t)$ drift upwards by also 10% over the course of a profit period. Doing so will continuously attract sellers, pulling profits out of the company as they happen and making the firm stay at the efficient level of capital. While the valuation $V_i(K)$ also stays the same, the shareholders gain π in cash, the number of outstanding shares declines, and the price per share increases.

Essentially, capital drift is a share buyback program. But unlike traditional buyback programs, the exact extent and timing of capital outflows are not driven by the managers and the traders they hire, but by the open market and its participants. The continuous capital corporation provides incentives for the sellers, but the sales themselves are coming from the market.

Companies could also use capital drift as a tool to adjust the capital level of a company and to bring it back onto the optimal capitalization path. Capital drift could even be negative, creating an incentive to inject more capital into the firm. We will make use of this tool in the next section, when applying our insights to real-world scenarios.

Lastly, I would like to refer to chapter 2 section 2.3. There, it is shown how retained profits and losses can be handled without resorting to *capital drift* and the resulting rules for calculating the number of shares an investor receives when investing.

3.6 Relation to Traditional Market Making

This section discusses the most important differences between the market making a continuous capital corporation engages in and that of traditional market makers, looking at the inventory risk and the pricing risk.

3.6.1 Inventory Risk

Garman (1976) formally describes the problem of the market maker in a very similar setting as ours: the market maker is assumed to be a price-setter, defining the relation between demand and prices. However, what complicates Garman's model significantly is the requirement that the market maker has a limited inventory and must never run out of shares to sell. The same applies to the model of Amihud, Mendelson, et al. (1986). Among other things, they show that the market maker's behavior is to a significant extent driven by the risk of running out of stock and that the replenishment costs play a significant role.

In contrast to a traditional market marker, the continuous capital corporation does not fear running out of shares, as it can always print new ones. Furthermore, unlike owning another company's shares, owning one's own shares does not come with a market risk or a liquidity risk. A share owned by the firm itself is economically equivalent to a share that does not exist at all. It cannot vote and it does not lead to an outflow of dividends. According to the latest accounting standard, it is not even a position on the active side of the balance sheet, but a negative position on the passive side, leading to an overall situation that is equivalent to these shares not existing at all.

When a firm does its own market making, the inventory risk vanishes, leading to a significant simplification of the process. What is left, however, is the risk of paying too much for a share or selling a share for too little. This is discussed in the next section.

3.6.2 Pricing Risk

From an accounting point of view, transactions with own shares are profit-neutral. They are economically equivalent to a capital increase or decrease. However, just with the capital increase or decrease, it is still possible to hurt the shareholders by either unduly diluting them (when selling too cheap) or by unduly reducing the value of the company (when buying back for too much). For the traditional market maker, this means that she has to live in constant fear to trade against a better informed trader who knows more about the true value of the firm than the market maker does. If there are too many informed traders, market making can even become unprofitable regardless of the spread, making the market illiquid.

While the pricing risk cannot be eliminated with the continuous capital corporation, it can be significantly reduced, and the risk of trading against better informed traders even turns into an opportunity. When a traditional market maker sells for price p even though the fundamental value is $p_f > p$, she has made a loss of $p_f - p$. However, in the case of a continuous capital corporation, the proceeds of the sale are used to capitalize the company, thereby creating additional value for everyone and making the transaction Pareto-improving when selling for a price at least as high as valuation function 3.6 dictates. Of course, there is still the opportunity cost of not having sold the share at the highest possible price, but the fact that value is being created through the addition of capital turns the zero-sum game of the stock market into a mutually beneficial scenario where everyone can win.

The same holds when repurchasing shares at a price that does not exceed that given by valuation function 3.6. Here, the added value stems from the informed seller being able to reinvest the capital at a higher return elsewhere, thereby also making the transaction Pareto-improving.

3.7 Conclusion

3.7.1 Learnings

What have we learned from all of this? The conclusions are three-fold.

First, we have found a valuation function that fulfills the criteria from the problem statement in section 3.3, supporting the efficient outcome under a wide range of circumstances. It provides guidance for companies that want to make use of the new legal and technical possibilities to create a small market for their own shares. A company that does so is considered a continuous capital corporation. Its capital is continuously adjusted as the market price of its shares moves up and down.

Second, having a rigid mechanism that links capital allocation to market prices shifts the capital allocation decision into the invisible hands of the free market, potentially obsoleting dividend payments and other management-driven financing activities. Whether this is a good idea depends on how much trust you have in the management of a firm to allocate the optimal amount of capital versus the trust you have in the market to allocate the capital where it is most productive.

As a small-scale investor, I like the thought of actually investing in a company when buying its shares. Today, buying a Tesla share on the stock market just sends money to its previous holder. There is no direct connection to the company itself. However, if Tesla was a continuous capital corporation, buying its shares would actually inject cash into the company, maybe allowing it to speed up the roll-out of its next great car. The continuous capital corporation would allow impact investing to actually have an impact.

Third, there remains a lot of unresolved work to make the continuous capital company function in practice. In order to apply even the simple valuation function 3.6, one would need to know the shape of the production function of a firm and its current capital level. But in practice, even the capital of a company might be non-trivial to determine as there are various forms of capital that might show up or not show up on the balance sheet depending on the applied accounting methods. Furthermore, a lot of the capital of a company consists of permanent and firm-specific investments that cannot be as easily undone as the mathematical model suggests. As clean and simple as equation 3.6 looks, it is not straightforward to apply.

Even if all problems of practical applicability were solved, there would still be the fundamental issue that the suggested pricing function represents the participation constraint of the current shareholder such that, by definition, all created surplus goes to the new investors. In reality, firms will try to sell at a higher price than what the valuation function suggests and repurchase at a lower price in order to capture their fair share of the created surplus when the capital allocation of the economy is improved.

3.7.2 Outlook

Whereas some practical questions remain unresolved, I expect an increasing number of companies to make use of the new legal and technical possibilities and to experiment with approaches between the one extreme of having a market that is detached from the firm's capital and the other extreme of having a market that fully determines a firm's capital.

A firm could achieve this relatively easily by choosing a pricing function that is steeper than what the presented model suggests. That way, the market would have some impact on the capital allocation, but the firm would still depend on some traditional management-driven financing to fully reach the efficient level. This also alleviates the surplus allocation problem, as a larger fraction of the created surplus would end up with the initial shareholders. In general, having an incremental path towards an innovative new form of financial markets is invaluable to its adoption.

I hope that many firms will recognize these new opportunities and start creating markets for their own shares, thereby eliminating the illiquidity discount quantified by Damodaran (2005), enabling founders or other significant shareholders to diversify their assets, and providing seed stage investors with an opportunity to exit a firm that has grown so they can use the proceeds for new seed stage investments. Further, if this could enable more firms to tap into a broader financial market, it would broaden the investment universe for the common investor again, countering the decline in the number of listed companies, an unfortunate trend described by Doidge et al. (2018) and first postulated by Jensen (1989).

As the proof of the pudding is in the eating, I am currently testing the presented ideas with my own company. Its shares can be bought and sold on its own website by anyone. The price is deterministically adjusted in accordance with a linear valuation function (Aktionariat, 2021). Also, I am applying the insights from this article in the design of the Frankencoin system discussed in chapter 2, allowing a decentralized protocol to become a continuous capital corporation.

Appendix

3.A Basic Example

This example shows what happens when a continuous capital corporation raises capital in the basic model.

The firm is assumed to have production function $f(K) = \sqrt{K}$. Interest rates are $r = 0.1$, and the firm is initially equipped with capital $K_0 = 1$ by its founders who hold $\theta_0 = 1$ shares. The optimal capital allocation is $K^* = 25$, at which point $f'(K) = r$.

When the firm starts offering shares at the valuation $V_i(K_0) = 2$ according to the internal valuation function 3.6, market participants immediately recognize that this is an attractive buying opportunity as the fundamental value of the firm is $V_e(K_0) = 10$, calculated using the external valuation function 3.8. The market participants buy shares until $V_i(K) = V_e(K)$, which happens at the efficient capital level $K_1 = 25 = K^*$.

But how many shares did the market participants get for their investment of $K_1 - K_0 = 24$ units of capital? This can be calculated with the function derived in equation 3.9. The market participants receive $\theta(K_1) - \theta(K_0) = 4$ shares, bringing the total number of outstanding shares to $\theta(K_1) = \sqrt{K} = 5$.

The existing shareholders started with fully owning a firm worth $V_e(K_0) = 10$ and ended with owning $\frac{1}{5}$ of a firm worth $V_e(K_1) = 50$, neither losing nor gaining anything despite the firm selling shares below the fundamental value. The new investors provided $K_1 - K_0 = 24$ in capital and got $\frac{4}{5}$ of the firm, increasing their wealth by 16. All of the surplus went to the new investors.

One might wonder what would happen if the firm used the external valuation $V_e(K)$ instead of $V_i(K)$ to sell its shares. To find out, we need to derive the number of shares $\theta_e(K)$ issued when K is invested at the external valuation:

$$\theta_e(K) = \int_{K_0}^{K} \frac{1}{p_e(k)} dk + \theta(K_0) = \int_{K_0}^{K} \frac{r}{f(k)} \theta_e(k) dk + 1$$

This time, the investors only get $\theta(K_1) - \theta(K_0) \approx 1.22$ shares (solved numerically), and the initial shareholders can significantly increase their wealth. They start with fully owning a firm worth 10 and end up with owning $\frac{1}{2.22}$ of a firm worth 50, gaining about 12.5. Surprisingly, some of the total surplus of 16 still goes to the new investors. This is owed to the observation that the new investors become existing shareholders themselves as they invest and increasingly participate in the surplus assigned to the existing shareholders with each marginal trade.

3.B Example with an Interest Rate Shock

Given a continuous capital corporation with production function $f(K) = \sqrt{K}$, this example starts with capital $K_0 = 25$, interest rate $r_0 = 0.1$, and a total number of outstanding shares $\theta_0 = 5$ and shows what happens when the interest rate drops to $r_1 = 0.05$.

When the interest rate drops to r_1, the internal valuation does not change, but the external valuation jumps from $V_e(K_0, r_0) = f(K_0)/r_0 = 50$ to $V_e(K_0, r_1) = f(K_0)/r_1 = 100$. This attracts new investors who buy freshly printed shares until the internal valuation and the external valuation are in balance again at the new efficient capital level $K_1 = 100$. The new investors get $\theta(K_1) - \theta(K_0) = 5$ shares for their investment, bringing the total to 10.

Before the transaction, the old shareholders fully owned a firm worth 100. After the transaction, they own 50% of a firm worth $V_e(K_1, r_1) = \frac{10}{0.05} = 200$, neither losing nor gaining anything. The new investors see their wealth increase from 75 in cash to owning shares worth 100. The transaction is Pareto-improving.

In the case of a drop of the interest back to $r_2 = 0.1$, everything would symmetrically unwind, and shareholders would return shares to the market maker until the initial capital level $K_2 = K_0$ is reached again. The shareholders who sell will move from owning 50% of a firm worth $V_e(K_1, r_2) = \frac{10}{0.1} = 100$ to owning 75 in cash, gaining 25 units of capital. The remaining shareholders move from owning 50% of a firm worth 100 to completely owning a firm worth 50, neither gaining nor losing anything. Again, the transaction is a Pareto-improvement.

References

Adams, Hayden et al. (2021). *Uniswap v3 Core*. URL: uniswap.org/whitepaper-v3.pdf.

Aktionariat (2021). *Investor Relations*. URL: aktionariat.com/investors.html.

Amihud, Yakov, Haim Mendelson, et al. (1986). "Asset pricing and the bid-ask spread". In: *Journal of financial Economics* 17.2, pp. 223–249.

Angeris, Guillermo et al. (2019). "An analysis of Uniswap markets". In: *arXiv preprint arXiv:1911.03380*.

Biais, Bruno, Thomas Mariotti, and Jean-Charles Rochet (2013). "Dynamic financial contracting". In: *Advances in economics and econometrics* 1, pp. 125–71.

Cho, In-Koo and Thomas J. Sargent (2016). "Self-confirming Equilibria". In: *The New Palgrave Dictionary of Economics*. London: Palgrave Macmillan UK, pp. 1–5.

Coinmarketcap (2023). *Uniswap*. URL: coinmarketcap.com/currencies/uniswap.

Crone, Hans Caspar von der, Martin Monsch, and Luzius Meisser (2019). "Aktien-Token". In: *GesKR* 1, pp. 1–17.

Damodaran, Aswath (2005). "Marketability and value: Measuring the illiquidity discount". In: *SSRN*. URL: ssrn.com/abstract=841484.

DefiLlama (2023). *Uniswap Volume*. URL: defillama.com/dexs/uniswap.

Deutsche Bundesregierung (2020). *Gesetz zur Einführung von elektronischen Wertpapieren*. URL: www.bmjv.de/SharedDocs/Gesetzgebungsverfahren/DE/Einfuehrung_elektr_Wertpapiere.html.

Diamond, Peter A. (1967). "The Role of a Stock Market in a General Equilibrium Model with Technological Uncertainty". In: *The American Economic Review* 57.4, pp. 759–776. ISSN: 00028282.

Doidge, Craig et al. (2018). "Eclipse of the public corporation or eclipse of the public markets?" In: *Journal of Applied Corporate Finance* 30.1, pp. 8–16.

Dow, James and Gary Gorton (1997). "Stock market efficiency and economic efficiency: is there a connection?" In: *The Journal of Finance* 52.3, pp. 1087–1129.

Egorov, Michael (2021). *Automatic market-making with dynamic peg.* URL: classic.curve.fi/files/crypto-pools-paper.pdf.

Forstmoser, Peter and Marcel Küchler (2020). "Schweizerische Aktienrechtsreform: Die Schlussrunde ist eingeläutet!" In: *Jusletter.* URL: jusletter . weblaw . ch / juslissues/2020/1010/schweizerische-aktie_0417b4599b.html.

Garman, Mark B (1976). "Market microstructure". In: *Journal of financial Economics* 3.3, pp. 257–275.

Hens, Thorsten and Sabine Elmiger (2019). "Economic Foundations for Finance". In: *Springer Texts in Business and Economics.*

Hertzog, Eyal, Guy Benartzi, and Galia Benartzi (2023). *Bancor Protocol - Continuous Liquidity and Asynchronous Price Discovery for Tokens through their Smart Contracts.* URL: whitepaper.io/document/52/bancor-whitepaper.

Jensen, Michael C (1989). "Eclipse of the Public Corporation". In: *Harvard Business Review* 5, p. 61. URL: hbr.org/1989/09/eclipse-of-the-public-corporation.

Jones, Charles I (2005). "The shape of production functions and the direction of technical change". In: *The Quarterly Journal of Economics* 120.2, pp. 517–549.

Nakamoto, Satoshi (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* URL: bitcoin.org/bitcoin.pdf.

Rosenfeld, Meni (2017). *Formulas for Bancor system.* URL: drive.google.com/file/d/0B3HPNP-GDn7aRkVaV3dkVl9NS2M/view.

Schnell Luchsinger, Monique and Pascal Montavon (2018). "Der Erwerb eigener Anteile durch die AG und die GmbH – 2. Teil: Steuerliche Aspekte". In: *Der Treuhandexperte* 5, pp. 284–297. URL: carlicahn.com/our_letter_to_tim_cook.

Schweizerisches Bundesgericht (2019). *2C_119/2018 - Urteil vom 14. November 2019.*

Swiss Federal Council (2020). *Anpassung des Bundesrechts an Entwicklungen der Technik verteilter elektronischer Register.* URL: parlament . ch / fr / ratsbetrieb / suche-curia-vista/geschaeft?AffairId=20190074.

Tinn, Katrin (2017). "Blockchain and the future of optimal financing contracts". In: *Available at SSRN 3061532.*

Wagner Alexander und Weber, Rolf (2017). "Corporate Governance auf der Blockchain". In: *SZW* 1, pp. 59–70.

Wood, Gavin et al. (2014). "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014, pp. 1–32.

Chapter 4

Power and Right of Disposal of Bitcoins in Bankruptcy

A legal assessment based on the characteristics of the new technology under the applicable Swiss laws

Co-authors: Christian Meisser, Luzius Meisser, Ronald Kogens

Publication: Meisser, C., Meisser, L., & Kogens, R. Verfügungsmacht und Verfügungsrecht an Bitcoins im Konkurs. In: *Jusletter IT* 2018, Mai 24. doi: 10.38023/ee56b560-ad84-4065-901c-bb6369c1a719

4.1 Abstract

The distinction between the power of disposal and the right of disposal is key when determining how to handle bitcoins in bankruptcy and in particular when deciding who they belong to under art. 197 SchKG. Further, we classify bitcoins as rival, fictitious, intangible assets sui generis. We argue that the law contains a gap with regards to the segregation (*Aussonderung*) and inclusion (*Admassierung*) of this asset type and show how to fill the gap in accordance with the intent behind the law.

4.2 Introduction

[1] Enormous potential is ascribed to cryptocurrencies and the underlying blockchain technology.[1] In this context, Johann Schneider-Ammann coined the term *Crypto Nation Switzerland*.[2] This technology is an opportunity to renew the strength of Switzerland as a global financial powerhouse, but only can only be fully leveraged with the right legal framework. One of the open legal questions is the treatment of cryptocurrencies under insolvency law. The answer to this question has far-reaching consequences for Swiss crypto asset service providers and their clients. It ultimately determines whether a bitcoin held in custody for a client appears on the balance sheet as a liability and thus whether it represents a deposit. If so, companies storing bitcoins for clients would, in most cases, need a banking license, and banks doing so would, among other things, have to observe the relevant capital requirements. As long as the bitcoins are only held in custody and not borrowed or used in any other way, treating them as an on-balance liability neither makes sense economically, nor helps protecting the client. For the clients, this would even have the undesirable consequence that in the event of the custodian's insolvency, they would be worse off than if they had direct title on the bitcoins in custody. Based on the rival nature of bitcoins, this article proposes to distinguish between the power of disposal and right of disposal, analogously to possession and ownership of things, in order to reach a more satisfactory legal outcome. These two terms are suitable for clarifying who owns a bitcoin and how it is to be treated in the event of a bankruptcy.

4.3 Real-world Example

[2] The Tezos Foundation received USD 232 million worth of Bitcoins and Ether via an Initial Coin Offering (ICO). Thanks to the steep rise in prices for these cryptocurrencies, the foundation's net worth temporarily even exceeded one billion dollar in December 2017. The power of disposal over the foundation's assets resides with a set of two cryptographic keys, both of which are required to initiate a transaction. The foundation has the first key. The second key is held by Bitcoin Suisse AG.[3]

[1] For example: Konrad Hummler, Blockchain - der nächste Wohlstandsschock, NZZ, May 3, 2016, p. 27; Luzius Meisser, Eine Chance für den Finanzplatz, NZZ, September 27, 2016, p. 10; Luzius Meisser, Die Blockchain als Standortvorteil, Finanz & Wirtschaft, August 13, 2016, p. 2

[2] Fadrina Hofmann, Schneider-Ammann will eine Crypto Nation Switzerland, Südwestschweiz, January 19, 2018.

[3] Bitcoin Suisse AG, Statement concerning the Tezos Crowd Contribution and the Tezos Foundation, November 13, 2017, bitcoinsuisse.ch/tezos-statement (all websites last visited on May 13,

There is a "Signature Service Agreement" between the two parties, which obliges Bitcoin Suisse AG to verify and sign transaction on request with its key. This form of storage significantly increases security. For example, a hacker who gains access to only one of the two keys cannot do much with it. The question now is whether and how one party should proceed in the event of a bankruptcy of the other party, and into which bankruptcy estate the foundation's assets would fall.

4.4 Legal Foundations

4.4.1 What are Bitcoins and How are They Held?

[3] Bitcoin is a decentralized Internet currency. Using the underlying blockchain technology, each bitcoin or fraction thereof is assigned to an address. The power of disposal over the bitcoins assigned to an address is exercised using one or more cryptographic keys (private keys). Bitcoin is referred to by the inventor as "digital cash" because bitcoins, like cash, can be transferred directly from person to person without an intermediary.[4] The price is determined on the free market by supply and demand. While addresses and private keys are data, even from the point of view of information theory, bitcoins are not data.[5] A blockchain consists of a chain of data blocks that serve as a transaction archive, but does not actually *contain* bitcoins in any meaningful sense. The bitcoins it refers to are purely fictitious.[6] The blockchain of the bitcoin system aims at creating a global consensus among all system participants as to which bitcoins are currently assigned to which address.[7] The address itself is a cryptographic fingerprint (hash) of the rules for disposing of the assigned bitcoins. In the simplest case, a single private key is sufficient and the address is derived from the associated public key, with which the validity of the transaction can be verified.[8]

2018); Finews, Tezos ICO: Bitcoin Suisse weiss, wo das Geld ist, finews.ch/news/finanzplatz/29590.

[4] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, bitcoin.org/bitcoin.pdf

[5] See Gabriela Hauser-Spühler/Luzius Meisser, Eigenschaften der Kryptowährung Bitcoin, in: Digma, 2018, Issue 1, p. 6.

[6] Shawn Bayern, Dynamic Common Law and Technological Change: The Classification of Bitcoin, 71 WASH. & LEE L.REV. ONLINE 22, 31 (2014).

[7] In computer science, finding a consensus in a network of participants with limited reliability and limited trust is generally known as the "distributed consensus" problem. The blockchain is not the first, but a new approach to solving this problem. See Roger Wattenhofer, The Science of the Blockchain, 2016.

[8] It is often claimed that the public key *is* the address. However, this is a simplification that does not allow for a precise discussion. Actually, the address is only a fingerprint of the rules

4.4.2 Legal Nature of Bitcoins de Lege Lata: Asset Sui Generis

[4]Bitcoin is an alternative currency. However, it does not come in any of the two known forms of currency. Bitcoins are not book money as they do not constitute a claim.[9] However, lacking a physical manifestation, bitcoins are not cash either. They also do not represent a membership right. In contrast to bitcoin addresses and private keys, bitcoins are also not data.[10] Only the transaction data (comparable to a chain of cession chain for claims, but without personal data) are recorded in the Bitcoin blockchain, but not the bitcoins themselves. Through use of the private key, bitcoins can be controlled factually[11] and legally [12]. Unlike data, bitcoins are rival goods because they cannot be used concurrently for competing purposes.[13] Due to these characteristics, bitcoins are, in our opinion, "rival, fictitious, intangible assets sui generis".[14]

[5] Recently, there has been an increasing discussion as to whether bitcoins can

to sign transactions concerning the associated bitcoins. An address can be derived from a single public key or more complex access rules involving multiple keys, but not vice versa. In order to be able to deduce the owner of the power of disposal over the assigned bitcoins from an address, additional information is required that cannot be found on the blockchain. The frequently used designation of the blockchain as an "ownership register" is misleading insofar as it is not possible to draw conclusions about the owner based solely on the data available in the blockchain. Conversely, however, the "owners" of a bitcoin can identify themselves as such without a doubt by disclosing the missing information.

[9] Mirjam Eggen, Chain of Contracts, AJP 2017, p. 14; François Piller, Virtual Currencies - Real Legal Problems?, AJP 2017, p. 1428.

[10] The power of disposal over bitcoins is recorded using a data structure, namely the blockchain, but that does not make the bitcoins themselves data. Book money, book-entry securities and real estate, whose ownership is also recorded in suitable data structures, are not data either. In contrast to bitcoins, data can be copied without diminishing the use of the original. The contrary and, in our opinion, incorrect view that bitcoins are data is often repeated in the legal literature without substantiation or source.

[11] Factual control exists when legal subjects can control a good according to its possibilities and subject it to their will. For more information see: Oliver Kälin, Der Sachbegriff im schweizerischen ZGB, dissertation, Zurich 2002, p. 56; Wolfgang Wiegand, in: Heinrich Honsell/Nedim Peter Vogt/Thomas Geiser (eds.), Basel Commentary, Civil Code Vol. II, 5th edition 2014, N 12 on Art. 641 ff. ZGB; Heinz Rey, Die Grundlagen des Sachenrechts und des Eigentums, Grundriss des schweizerischen Sachenrechts, Volume I, 3rd ed., Bern 2007, Rz 77

[12] A good is considered to be legally controllable if positive law permits control over it or rights can be established on it. For more details see: Kälin (fn. 11), p. 58; Wiegand (fn. 11), N 13 on Art. 641 ff. ZGB.

[13] Katie Szilagyi, A Bundle of Blockchains? Digitally Disrupting Property Law, 1 Cumberland Law Review 48 2018 (forthcoming), p. 15.

[14] Fictitious implies intangible, but not vice versa. For example, data is intangible but not fictitious. Of all the mentioned asset types, bitcoins are probably closest to cash. The main difference is that cash is bound to metal or paper to locate it in physical space, while bitcoins are bound to abstract locationss in the address space of the bitcoin system.

be regarded as *things*. The doctrine defines things as impersonal, physical, and definable objects that can be subjected to legal control.[15] The term *thing* is of functional nature, which is why its qualification as a thing does not only depend on its physical properties, but above all on its economic function, on its common perception, and on ethical considerations.[16] Furthermore, teleological arguments have an influence on whether the legal system recognizes something as a thing or not.[17] Without disecting each of the defining legal characteristics of a *thing*, we take the view that bitcoins fail to qualify due to their lack of a physical body. Its physical manifestation is the central quality of a thing according to the currently prevailing doctrine. [18] The requirement of a physical manifestation leads to a limitation of the term *thing* to material objects with a spatial presence and a mass. It distinguishes *things* from claims and other rights or intangible objects.[19]

[6] Art. 713 Swiss Civil Code (ZGB; SR 210) puts natural forces on an equal footing with moveable physical property ("Fahrniseigentum"), despite their (partial) intangibleity, and applies the rules of movable property analogously,[20] provided that the respective natural forces can be subjected to legal control. These legally controllable natural forces include energy of a hydraulic, electrical, chemical or nuclear nature.[21] According to the Federal Supreme Court, energy of an electrical nature means the

[15] Wiegand (fn. 11), N 6 on Art. 641 ff.; Ruth Arnet, in: Peter Breitschmid/Alexandra Rumo-Jungo (editors), Handkommentar zum Schweizer Privatrecht, Sachenrecht, vol. 3, 3rd edition 2016, N 6 on Art. 641 ZGB; Arthur Meier-Hayoz, Berner Kommentar, Systematischer Teil und Allgemeine Bestimmungen, Art. 641-654 ZGB, 5th edition, Bern 1981, N 115 zu Sachen und anderen Rechtsobjekten; Rey (fn. 11), para. 66 ff.

[16] Meier-Hayoz (footnote 15), N 115 zu Sachen und anderen Rechtsobjekten; Rey (fn. 11), para. 68

[17] Meier-Hayoz (footnote 15), N 115 zu Sachen und anderen Rechtsobjekten; Rey (fn. 11), para. 115 f.; Wiegand (fn. 11), N 6 f. to Art. 641 ff.

[18] Gl.M. Harald Bärtschi/Christian Meisser, Virtuelle Währungen aus finanzmarkt- und zivilrechtlicher Sicht, in: Rolf H. Weber/Florent Thouvenin (eds.), Rechtliche Herausforderungen durch webbasierte und mobile Zahlungssysteme, Zurich 2015, p. 141; Benedikt Maurenbrecher/Urs Meier, Insolvenzrechtlicher Schutz der Nutzer virtueller Währungen, in: Jusletter December 4, 2017, para. 20; Jean-Marc Schaller, Blockchain-Serie | # 2 use cases: Rechtsprobleme und Lösungen, blog post 33 from March 15, 2018, www.finblog.ch, para. 10; Of course, an intangible good such as software or a collection of data can be stored on a physical data carrier, such that it can qualify as thing in that concrete manifestation, subject to also fulfilling the other necessary criteria.

[19] Wiegand (fn. 11), N 10 on Art. 641 ff.; Arnet (fn. 15), N 10 on Art. 641 ZGB; Meier-Hayoz (footnote 15), N 117 zu Sachen und anderen Rechtsobjekten; Rey (fn. 11), para. 81

[20] Jörg Schmid/Bettina Hürlimann-Kaup, Sachenrecht, 5th edition 2017, Rz 1080; Rey (fn. 11), para. 86

[21] Schmid/Hürlimann-Kaup (fn. 20), para. 1079; Ivo Schwander, in: Heinrich Honsell/Nedim Peter Vogt/Thomas Geiser (eds.), Basler Kommentar, Zivilgesetzbuch Bd. II, 5. Aufl. 2014, N 8 zu Art. 713

supply, allocation and provision of electricity.[22] An expansion of this concept onto intangible assets predominantly produced with electrical energy is in our opinion not possible. This is why bitcoins also cannot be subsumed under the interpretation of the term "natural forces" under the prevailing doctrine.

4.4.3 Legal Nature of Private Keys and Addresses

[7] In contrast to the bitcoins themselves, the private keys and the addresses used in the Bitcoin system consist of a sequence of characters that can be copied at will and therefore qualify as data.

[8] In the legal classification of data, a distinction is made between personal data and object data.[23] Both types are inherently public goods and non-rival: the use of data by one person does not restrict its concurrent use by another person. Of course, the economic value of data can be negatively impacted through the use by other people. But this is only a secondary economic consequence and not a direct usage restriction. For this reason, data is generally not protectable and, in our opinion, cannot be considered a thing under the applicable laws due to the absence of a physical manifestation and its lack of factual and legal controllability.[24] Any opposing view would contradict positive law, which only assigns property-like rights to very few well-defined types of data (e.g. personal data, object data that represent intellectual creations with an individual character, trademarks, etc.), and would lead to considerable legal uncertainty due to overlaps with established intellectual property laws.

[9] The handling of personal data is regulated in the Federal Data Protection Act (FADP; SR 235.1). It grants certain rights to the persons affected by the processing of their personal data. However, despite their property-like character, these rights are not comparable with the property rights from property law. They are mainly limited to information rights and rights regarding the correction, blocking or deletion of personal data.

[22] BGE 48 II 366 E.2.

[23] Florent Thouvenin, Wem gehören meine Daten? Zu Sinn und Nutzen einer Ausdehnung des Eigentumsbegriffs, SJZ 113/2017, p. 21 f.

[24] Eq. M. Florent Thouvenin/Alfred Früh/Alexandre Lombard, Eigentum an Sachdaten: Eine Standortbestimmung, SZW 2017, p. 26; Bärtschi/Meisser (fn. 18), p. 141; Maurenbrecher/Meier (footnote 18), Rz 20; Schaller (fn. 18), para. 10; Gianni Fröhlich-Bleuler, Eigentum an Daten?, in: Jusletter March 6, 2017, para. 15

[10] Also object data is often unprotected. In order for object data to enjoy copyright protection, it must be an intellectual creation of literature or art with an individual character (Federal Act on Copyright and Related Rights, Copyright Act [URG; SR 231.1]). The rights of the performing artists are set out and protected under Art. 33 URG, which refers to audio-visual recordings. Protection of data under trademark law (e.g. a character string or sequence of letters) is only possible if there is no absolute or relative reason for exclusion (Article 2 and Article 3 of the Federal Act on the Protection of Trademarks and Indications of Source, Trademark Protection Act [MSchG; SR 232.11]).

[11] Other normative foundations such as competition, tort, criminal and contract law do not confer exclusive rights "*erga omnes*" to the person a piece of data belongs to, nor do they grant any positive rights. They only prohibit certain actions related to data by making them a punishable offence.[25] For example, it is forbidden to misapropriate market-ready intellectual fruits of labor (Art. 5 lit. c Federal Act on Unfair Competition [UWG; SR 241]), to make use of business secrets without authorization (Art. 6 UWG) or to violate the general clause regarding behavior in competition (Art. 2 UWG). Tort law is only relevant if, in addition to damages and causality, there is unlawful behavior and fault. In connection with data, the unlawfulness can result from, among other things, Art. 162 of the Swiss Criminal Code (StGB; SR 311.0), Art. 143 StGB, Art. 143bis StGB and Art. 144bis StGB. In contrast to the rights associated with an item, any claims for in rem restitution can only be asserted against the infringer. This also applies to contractual agreements that are only binding "*inter partes*".

[12] The private key and the address are object data, but they do not represent an intellectual creation within the meaning of copyright law, which is why there is no copyright protection. Registration as a trademark would at best be legally permissible, but would contradict the purpose of the address and the private key.[26] Consequently, in connection with the private key and the address, at most claims for in rem restitution can be asserted against the infringer.

[25] Thouvenin/Früh/Lombard (fn. 24), p. 30.

[26] The address should not be protected against use by third parties because it is required to receive bitcoins. Strict secrecy must be exercised over the private key. A trademark registration would counteract this and quickly lead to the loss of the associated bitcoins.

4.4.4 Analogous Application of Principles of Property Law

[13] As already laid out, the prevailing doctrine defines things as impersonal, corporeal, and distinguishable objects which are subject to actual and legal control. Natural forces are treated like movable physical property and the according rules applied analogously,[27] given that the respective natural forces are actually and legally controllable. Actual controllability does not depend on the legal object, but rather on the legal subject and his or her ability to control a good.[28] A good is considered legally controllable (transactionable) if positive law permits control over it or rights can be based on it.[29]29 [14] Besides the factual and legal controllability, legal certainty and the associated publicity principle are other important elements of property law in order to establish the associated rights that can be precisely distinguished and reliably recognized by anyone.[30] Publicity for rights to movable objects is conveyed through physical possession, namely having actual control over an object (but also certain registers such as the public *Eigentumvorbehaltsregister*). Publicity for immovable real-estate is established through a public land registry.[31]

[15] The power of disposal (similar to possession, see below) of bitcoins is exerted through a private key, with the publicly accessible blockchain enabling third parties to verify ownership claims. Even if the bitcoin system could be copied as a whole, there is only one "*true*" Bitcoin blockchain and the number of bitcoins is limited to 21 million.[32] A non-rival use of bitcoins is therefore impossible. Even if several people have access to the same private key, use by one person to move the associated bitcoins restricts the other people's use of the same bitcoins. In the case of bitcoins, due to their rival nature, the power of disposal conveys publicity analogously to possession of physical objects. Bitcoins as "*rival, fictitious, intangible assets sui generis*" can actually and legally be controlled (acquired, appropriated, used) through cryptographic methods, usually the signing of transactions with the associated private key. It therefore seems permissible to apply legal principles related

[27] Schmid/Hürlimann-Kaup (fn. 20), para. 1080; Meier-Hayoz (footnote 15), N 118 zu Sachen und anderen Rechtsobjekten.

[28] Kälin (fn. 11), p. 56; Wiegand (fn. 11), N 12 on Art. 641 ff. ZGB: "*Tatsächliche Beherrschbarkeit:* [. . .] *the possibility of subjecting objects to the human will*".

[29] Külin (fn. 11), p. 58; Wiegand (fn. 11), N 13 on Art. 641 ff. ZGB.

[30] Fröhlich-Bleuler (footnote 24), Rz 15; Meier-Hayoz (fn. 15), N 57 ff. zu Quellen und Hilfsmittel. Rey (fn. 11), para. 272 ff.; Arnet (fn. 15), N 20 on Art. 641 ZGB.

[31] Rey (footnote 11), Rz 278 et seq.

[32] Everyone is free to copy the Bitcoin system as a whole and to separate their copy from the original system by taking appropriate measures. However, this does not create additional bitcoins, but a new currency. Well-known copies are "Litecoin" and "Bitcoin Cash". These often arise due to ideological disputes among the developers.

to rights in rem analogously to bitcoins. In particular, this concerns the distinction between power of disposal in analogy to possession and right of disposal as an "*absolute right of a special kind*" in analogy to ownership. In that regard, it is always necessary to critically examine whether and, if so, to what extent a norm of property law can and should be applied by analogy.

[16] In contrast to Bitcoins, the private key and the address are factual data and there for not rival goods. The use of the private key or the address by one person does not restrict their use by other persons. The secondary effect of their loss in economic value when used by other persons is not relevant for the assessment of their rival nature.

4.4.5 Power of Disposal and Right of Disposal

[17] The power of disposal over bitcoins is held by those who can directly determine to which address they will be assigned next. The power of disposal is normally exercised by means of a private key. Which private keys are required in which combination depends on the address to which the bitcoins are assigned. In most cases, there is exactly one private key per address, and anyone who knows this key can transfer the bitcoins associated with it to a new address using a signed transaction. The signatures used are impersonal and differ from the qualified electronic signatures from the Federal Act on Certification Services in the Field of Electronic Signatures and Other Applications of Digital Certificates (ZertES; SR 943.03) as they are not associated with a name and also do not contain any other personal data. It is also possible to generate addresses that are linked to several private keys, so that a collective signature setup or similar schemes can be represented directly in the Bitcoin system. In such cases, the addresses are commonly referred to as *multi-signature addresses*. For example, the assets of the Bitcoin Association Switzerland are assigned to a *two out of six multi-signature* address. Each of the six board members has a private key, and two of these are required to dispose of the bitcoins assigned to the address.[33]

[18] The power of disposal over certain bitcoins does not necessarily have to be represented directly in the Bitcoin system to be effective. It can also be restricted through technical measures outside of the Bitcoin system. A well-known example

[33] Since the address is publicly known and all transactions in the Bitcoin system are publicly visible, the Bitcoin Association's account balance can be viewed at any time, for example at blockchain.info/address/35TTXLEtU8ZKAeTEBkx6qG7Cox8RyDw3uW.

are the metal coins with the Bitcoin sign that are often featured in the media. These coins contain a printed private key behind a physical seal. This makes it possible to transfer control of a bitcoin by handing over a physical object. In general, solutions for managing private keys are called wallets and these physical coins fall into the category of *paper wallets* despite being made out of metal. A wallet helps to keep private keys safe and simplifies the handling of cryptocurrencies. Depending on the nature of the wallet, the user transfers the power of disposal over the bitcoins to the wallet provider to a different degree.[34] The decisive question is whose cooperation is necessary in order to be able to dispose of the bitcoins. This power of disposal is to be understood in the same way as possession of things, whereby setups with distributed custody can be encountered. [35] In addition to the power of disposal, the right of disposal is analogous to ownership as a right in rem, for the clarification of which the contractual relationships must be taken into account.[36]

4.5 Bankruptcy of the Custodian and Consequences for the Beneficiary

4.5.1 Introduction

[19] Lacking a physical body, bitcoins are not recognized as *things* by the law. Whether a qualification of bitcoins as a thing, with the consequence of the application of all legal provisions, would be desirable at all, is not examined in this article. Of practical importance at the moment is the question under which circumstances bitcoins fall into the bankruptcy estate of a custodian and whether the clients have a right of separation.

[20] The following shows that there is a legal gap in bankruptcy law when it comes to bitcoins, as they are a novel phenomenon that was not anticipated by the legislator. Due to the nature of bitcoins, the analogous application of principles from property law is a suitable way to fill this gap. This leads to results that correspond to the presumed will of the legislator.

[34] See Maurenbrecher/Meier (fn. 18) for a more precise distinction between different wallet types.

[35] An example for distributed custody is the service blockchain.info, in which the user's secret keys are stored in encrypted form without the provider knowing the user's password, which is necessary for decryption and usage of the stored keys. Therefore, neither the user nor the provider can access the stored bitcoins on their own.

[36] We would like to thank Benedikt Maurenbrecher for the idea of distinguishing between power of disposal and authority/right of disposal.

4.5.2 Existence of a Legal Gap

Principle

[21] In current teaching and case law, a legal gap is defined as an incompleteness within the law that is contrary to the law's purpose. An incompleteness is contrary to a law's purpose if the law does not provide an answer to a legal question even though its purpose would demand an answer, or if the law lacks an exception although its purpose would demand for an exception. At the same time, there is no legal gap if the law provides an answer that is unsatisfactory. The law requires courts to identify legal gaps and to rectify them when applying the law, but it does not allow them in principle to change factually unsatisfactory answers in the legislation.[37]

[22] A gap can in particular result "*from a change in the technical, economic or social context of the application of the law*".[38] The completeness or incompleteness of the law is to be judged in the light of its immanent purpose.[39]

4.5.3 Immanent Purpose of Bankruptcy Laws

[23] The Federal Law on Debt Enforcement and Bankruptcy (SchKG; SR 281.1) is intended to ensure an orderly enforcement of open claims against the debtor by the state.[40] In the case of a bankruptcy, in particular, the entirety of the debtor's assets are accumulated in the so-called general execution to satisfy the creditors.[41] The immanent purpose of bankruptcy law is therefore to satisfy the creditors using the debtor's assets.[42] When bankruptcy proceedings are opened against a debtor, a special estate is created for this purpose, the so-called bankruptcy mass, which consists of all seizable and salvagable assets [43] that belong to the debtor at the time

[37] Susan Emmenegger/Axel Tschentscher, Berner Kommentar zum Schweizerischen Zivilgesetzbuch, Volume I No. 1, 2012, Art. 1-9 ZGB, N 344 f.

[38] Emmenegger/Tschentscher (fn. 37), N 354.

[39] Emmenegger/Tschentscher (fn. 37), N 356.

[40] Jolanta Kren Kostkiewicz, Debt Enforcement and Bankruptcy Law, 2nd edition, 2014, N 1 ff.

[41] Kren Kostkiewicz (fn. 40), N 52.

[42] An even further reaching liability beyond the debtor's assets, such as the ancient Roman method to lay hands on the debtor, is not the purpose of the law; Hansjörg Peter, 125 years SchKG - 125 Jahre Rechtssprechung zum SchKG, in: Blätter für Schuldbetreibung und Konkurs 2017, pp. 45-62, p. 48.

[43] On the requirements for the salvage of bitcoins: Olivier Hari, La revendication et la distraction d'office d'actifs dans une procédure d'insolvabilité: application des principes aux monnaies cryptographiques, in: GesKR 2017, p. 462.

the proceedings were opened,[44] and includes the entirety of the debtor's assets[45] (cf. Article 197 (1) SchKG).

[24] However, it is precisely not the purpose of bankruptcy law to use assets that does not belong to the debtor or that are not part of the debtor's wealth to satisfy the creditors, even if the debtor has sole control over the assets.

[25] This is also reflected in the rule that the bankruptcy assets should be "*what appears to third parties as the property of the bankrupt*".[46]

[26] These principles should be our guidelines when interpreting the bankruptcy law in relation to the question under what circumstances bitcoins fall into the bankruptcy estate and whether segregation or a forced inclusion is possible. First, the existing regulations in bankruptcy law are discussed and it is shown that in practice, gaps have already been filled for (non-rival) factual data, despite a greater distance to the wording and intent of the law. We then discuss the appropriate interpretation of bankruptcy law in the case of bitcoins as "rival, fictitious, intangible assets sui generis".

4.5.4 Regulations in Bankruptcy Law

[27] The basic provisions of bankruptcy law with regard to the bankruptcy estate are based on two concepts of Roman law that seem self-evident today: the legal construct of property as a bankruptcy-proof, absolute right that legally assigns a thing to a person, and the *obligation*: a non-bankruptcy-proof, relative right consisting of a claim or a debt.[47] In order to account for economic and technological changes, various special legal regulations have been added over the years. Certain immaterial goods, especially intellectual property, now also enjoy absolute protection. In addition, there are various other new regulations, including the Banking Act (BankG; SR 952.0), in the Insurance Contract Act (VVG; SR 221.229.1), in the Federal Law on Pledging and Compulsory Liquidation of Railway and Shipping Companies (VZEG;

[44] Kren Kostkiewicz also refers to assets to which the debtor is *legally entitled*: Kren Kostkiewicz (fn. 40), N1194.

[45] Jolanta Kren Kostkiewicz, in: Daniel Hunkeler (ed.), Kurzkommentar Schuldbetreibungs- und Konkursrecht, 2nd edition, 2014, Art. 197, N 2; Zur Voraussetzung der Verwertbarkeit: Hari (fn. 43), p. 462.

[46] Lukas Handschin/Daniel Hunkeler, in: Daniel Staehelin/Thomas Bauer (eds.), Basler Kommentar zum Bundesgesetz über Schuldbetreibung und Konkurs II, 2nd edition, 2017, Art. 197 N 71.

[47] Peter (fn. 42), p. 45.

SR 742.211) and much more. Furthermore, data protection laws introduced certain absolute rights which, in the event of bankruptcy, may prevent the use of personal data or even justify a claim for return.

[28] Even with personal data, but especially with factual data, the bankruptcy trustee is faced with legal questions for which the law currently has no or at least no satisfactory solution: factual data does not enjoy absolute legal protection, and the central contractual (relative) claims of the "owner" the factual data, that (i) restrict the use or exploitation of the data by the debtor and (ii) justify a claim for return, are non-binding for the bankruptcy estate (Art. 211 SchKG).[48] However, factual data is stored on a physical carrier, so that at least the substantive provisions can be applied indirectly.

[29] The bankruptcy law does not contain any specific regulation for cryptocurrencies such as bitcoins, which are neither factual data nor things and are not stored on a data medium. One approach discussed in teaching would be to focus on the private key as a non-rival piece of factual data - but the resulting consequences contradict the purpose of the law. Against the background of the novelty of Bitcoins, it is obvious that this is an incompleteness that is contrary to the law's purpose and therefore a true legal gap.

4.5.5 Filling the Legal Gap

Unsatisfactory Solution for Factual Data

[30] The difficulty of constructing an absolutely protected, bankruptcy-proof property right for factual data results from the fact that they lack the property of rivalry. A clear allocation of factual data to an asset - i.e. a "belonging" - is difficult or impossible to construct. Since factual data is usually stored on physical data carriers, and data carriers as a thing can be clearly assigned to an owner, factual data follows the physical data carrier in the absence of other regulations. This leads to problems if someone has saved "their" factual data on the competitor's data carrier. According to the clear wording of the law, the data carriers are included in the bankruptcy estate of the party entitled to the physical data carrier (hosting provider). The data carriers are then to be treated like objects without consideration for the value of the data they carry. In the absence of any other regulation, the factual data follows the

[48] Kren Kostkiewicz (fn. 40), N 1249 ff.

fate of the data carrier. A client of the hosting provider has no statutory claim to return "his" data. The data could either simply be destroyed or, if it has a market value and can be decrypted without the client's involvement, used together with the data carrier.

[31] In the case of a teleological interpretation of the term "belonging", the salvaging of data for the benefit of the bankruptcy estate [49] also violates in our opinion the principle that the bankruptcy estate is formed from the debtor's assets, which should consist of what is externally, for third parties, recognizable as the property of the debtor: a creditor of a bankrupt hosting provider will hardly expect that the data stored on behalf of third parties will be part of the hosting provider's bankruptcy estate. The problem with this interpretation, however, is that due to the non-rival nature of data, it is not always possible to clearly assign it to a specific owner and to identify who it "belong" to.

[32] Competition and criminal law may offer a certain degree of protection against exploitation when it comes to special categories of factual data, such as business secrets. However, a claim for return - which is probably the greatest need in practice - cannot be derived from these provisions.

[33] In Luxembourg, this deficiency in the law has already been remedied by the legislature.[50] In Switzerland, however, the Federal Council has not (yet) recognized a need for action,[51] but various bankruptcy offices are filling this gap in application of the legal principles and using "common sense". According to their own statements, they in fact do release business data to clients of bankrupt hosting providers.[52]

[49] Zur Verwertung von Daten: David Schwaninger/Stephanie S. Lattmann, Cloud Computing: Ausgewählte rechtliche Probleme in der Wolke, in: Jusletter March 11, 2013, para. 58 ff.

[50] "*Art. 567 para. 2 of the Luxembourg Commercial Code has stipulated since 2013 that there is a right to the return or transfer of intangible movable assets against a bankrupt company. The prerequisite is that the bankrupt only hosts the data (and is not the legal owner of it), that the data has been entrusted to the debtor and that the data can be separated. The client must bear the costs of the release.*" Mark A. Reutter, When Your Cloud Provider Goes Bankrupt, in: Computer Week, April 25, 2016; see also: www.computerworld.ch/technik/digitalisierung/clou-provider-bankrott-geht-1341186.html.

[51] Federal Council response to Question 14.1064 by Jean Christophe Schwaab, Muss das Konkursrecht in Bezug auf Computerdaten ergänzt werden?, www.parlament.ch/de/ratsbetrieb/suche-curia-vista/geschaeft?AffairId=20141064. The need for action is denied with the argument (amont others) that a change of ownership is not only possible in the event of bankruptcy, but also in the event of a company takeover or in the event of inheritance. This fails to recognize that in case of a bankruptcy, the contractual claims for return of data and agreed usage restrictions cease to apply. This is what creates the need for protection.

[52] According to Reutter (fn. 50): "*Sometimes, out of good will or pragmatism, bankruptcy offices still offer to return client data, regardless of the lack of a sound legal foundation.*"; and Peter K. Neuenschwan der, Daten im Konkurs, lecture at the conference of the Schweizer Forum für Kom-

[34] So it can be observed the unsatisfactory answer the bankruptcy law provides regarding factual data is in practice corrected in accordance with the rules of filling legal gaps. Whether there is actually is a gap in the regulation regarding factual data that has to be filled by the judiciary, or whether this is merely an unsatisfactory result that needs to be addressed by the legislators, can be left open for the assessment of the handling of bitcoins. Bitcoins are not data and they are not "trapped" in a data carrier belonging to the debtor.

Custody versus Claim

[35] The following explanations only apply to the storage of bitcoins for a client. As soon as there is only a claim for payment of an amount denominated in bitcoins, the established legal provisions about claims are applicable.[53] Similar to the deposit agreement, the contractual agreement between the parties and the chosen method of storage are decisive for distinguishing between claims and custody. In practice, the following three types of custody may be the most relevant:

- *Individual custody:* The stored bitcoins are kept for the client on a dedicated address or technically separated from one another through other suitable measures. The custodian is obliged to return the same bitcoins to the client. It can be further distinguished between the case where the custodian has sole power of disposal and the case of shared power of disposal (e.g. in a multi-signature setup like in the Tezos described earlier).
- *Collective Custody:* The bitcoins of the client are mixed with the bitcoins of other clients with the explicit approval of the clients. They are kept on addresses dedicated for client assets and separated from the custodian's own bitcoins. Recognizing their fungibility, the custodian does not have to return the same bitcoins to the clients, but only the same number of bitcoins.[54]
- *Depositum irregulare:* The bitcoins are mixed with the custodian's bitcoins. The client is only entitled to the same number of bitcoins.

munikationsrecht zu Dateneigentum und Datenzugang on November 29, 2017 (see slides at: www.sf-fs.ch/files/sffs__itsl__tagungsschriften_dateneigentum_und_datenzugang_website.pdf).

[53] On the question of whether a claim denominated in bitcoins should be enforced through a traditional payment or actual delivery: Francois Piller, Virtuelle Währungen - Reale Rechtsprobleme?, in: AJP 2017, pp. 1426-1438, pp. 1433 ff.

[54] On this topic in general and in particular on the distinction between collective custody and depositum irregulare: Eugen Bucher, Schweizerisches Obligationenrecht, Besonderer Teil, 3rd edition, Zurich 1988, §16 III.

[36] As with the deposit of fungible items, custody can only be recognized as such in the first two cases. In the case of the so-called "depositum irregulare", according to the view represented herein, both the power of disposal and the right of disposal are transferred to the debtor and in the event of bankruptcy, the client can only assert a claim.[55]

Filling the Legal Gap for Bitcoins

[37] While the law offers certain starting points for factual data, since these are stored on a data carrier whose treatment under bankruptcy law is regulated as a thing, bankruptcy law is largely silent on how bitcoins as "rival, fictitious, intangible assets sui generis" are to be treated. Here, the law fails to provide an answer even though it should according to its purpose and we are faced with a true legal gap with regards of a new technical form of assets. This gap can be closed with a teleological interpretation of bankruptcy law.[56]

[38] We show below how a teleological interpretation of bankruptcy law[57] with the treatment of bitcoins as "*rival, fictitious, intangible assets sui generis*" leads to appropriate solutions in practice that do justice to the will of the legislature and the technological and economic circumstances.

[39] For this purpose, the term "*belong*", which is central in bankruptcy law, is to be interpreted in accordance with the purpose of the law and, in the event of a dispute about who a bitcoin in the context of a bankruptcy actually belongs to, the segregation and inclusion pursuant to Art. 242 SchKG is to be applied analogously.

[55] Luxembourg law goes even further with regard to the segregation of data in the bankruptcy of the Hosting providers: here, it is sufficient that the data can be separated in the bankruptcy proceedings. Separate storage beforehand is not necessary (www.internationallawoffice.com/Newsletters/Insolvency-Restructuring/Luxembourg/NautaDutilh-Avocats-Luxembourg/New-right-to-reclaim-data-from-bankrupt-cloudcomputing-providers).

[56] A parliamentary initiative to eliminate the legal gap at least for data is pending. See Marcel Dobler's parliamentary initiative, Parlamentarische Initiative von Marcel Dobler, Daten sind das höchste Gut privater Unternehmen. Datenherausgabe beim Konkurs von Providern regeln, www.parlament.ch/de/ratsbetrieb/suche-curiavista/geschaeft?AffairId=20170410.

[57] One could also argue that the legal gap is already at the level of the Civil Code and that the failure of the law to recognize bitcoins as *things* consistutes the legal gap. In our opinion, however, it seems more appropriate to apply the analogy to tangible property only in specific areas. A general treatment of bitcoins as a things may have unforeseen and impractical consequences. Furthermore, according to the current state of scholarly discussion, the legal gap in bankruptcy law is the most relevant.

What does "Belong" Mean?

[40] The dictionary defines "belong" as "*to be part of someone's property*", or as "to be rightly placed".[58] The literal sense of "belonging" according to Art. 197 SchKG goes further than the strict definition of property and also encompasses other items of value that can be attributed to someone. This is also apparent from the French version of the law, in which the affiliation is expressed with the preposition "du" ("*Tous les biens saisissables du failli*"). With this in mind, we base our assessment on the following definition of the term "*belong*":

> [41] "*Belong*" means that a specific asset can be attributed to a specific estate.

Things

[42] With regards to things, "belong" refers to the property right, and not to possession.[59] The determining factor is therefore who may rightfully dispose of an item, not who can actually dispose of an item.

Data

[43] Non-rival data can often not be clearly assigned to a single estate: factual data can belong to two or more estates at the same time, without this necessarily limiting the possibility of liquidating it in bankruptcy. An analog application of property law is therefore highly questionable.[60] For the treatment of data under bankruptcy law, ownership of the data carrier is therefore linked: if the data carrier is part of the bankruptcy estate, the law states that the data stored on it is also part of the bankruptcy estate.[61]

[58] The term used in the German version of the law is "gehören", which differs in subtle ways from its English twin "belong".

[59] Handschin/Hunkeler (fn. 46), Art. 197 N 8.

[60] Thouvenin/Früh/Lombard come to the conclusion: "Our review of the state of the discourse has shown that it is currently not possible to say whether and, if so, in what form legal ownership of data should be introduced." In addition: "*An orientation towards copyright laws seems to be the most obvious at the moment. These tools can adequately deal with the non-rival nature of data* [. . .]", Thouve nin/Früh/Lombard (fn. 24), p. 33 f. In their differentiated contribution, Benhamou/Tran refer to foreign decisions in which domain names are treated analogously to property - in our opinion a good example of rival factual data . Yaniv Benhamou/Laurent Tran, Circulation des biens numériques: de la com mercialisation á la portabilité, in: sic! 2016, pp. 571-591, p. 575.

[61] Reutter (fn. 50); Peter K. Neuenschwander/Simon Oeschger, data in bankruptcy, in: Jusletter IT Flash 11 December 2017.

Bitcoins

[44] In contrast to data, bitcoins are always rival, i.e. unlike data and like things, bitcoins can be assigned to a specific estate. At the same time, the nature of Bitcoin as a rival asset means that it cannot reside with multiple estates at the same time. The rival nature of bitcoins not only enables to clearly attribute bitcoins to the estate of the debtor or to the estate of a third party in the event of bankruptcy, but also makes such an assignment necessary. In order to determine who a bitcoin belongs to, it is necessary to distinguish between "power of disposal" (analogous to possession) and "right of disposal" (analogous to property). The power of disposal usually comes with knowing the private key.[62] As with the possession of things, the power of disposal can be shared or be non-exclusive. However, in order to do justice to the purpose of the law, we believe that the "right of disposal" must be used as the basis for the allocation of the assets in question. The actual, economic and contractual circumstances must be taken into account to determine the right of disposal. In accordance with the purpose of the law and the interest of the creditors, the expectation of the creditors must also be observed when attributing the right of disposal, asking the question: "Which bitcoins can the creditors reasonable expect to fall into the bankruptcy estate?"

4.5.6 What Falls into the Bankruptcy Estate?

[45] In principle, only what "belongs" to the bankrupt debtor is part of the bankruptcy estate. Assets belonging to a third party do not fall into the bankruptcy estate.[63] In the case of rival assets that are co-owned or jointly owned and in disputed cases, further differentiation must be made for the assignment of an asset to the bankruptcy estate.

[46] Assets that are only "in the possession" of the debtor but that belong to a third party are also to be included in the inventory of the debtor, but with a note regarding the true ownership, with the exception of obvious cases.[64] These assets do not fall into the bankruptcy estate.[65] If the ownership is disputed, the location

[62] See Maurenbrecher/Meier (fn. 18), who only consider setups with one private key. In our opinion, this approach does not fully capture the problem at hand, which becomes particularly apparent when considering multisignature addresses with multiple distinct key holders.

[63] Kren Kostkiewicz (fn. 45), Art. 197 N 3.

[64] Kren Kostkiewicz (fn. 40), N 1345.

[65] Handschin/Hunkeler (fn. 46), Art. 197 N 66.

of the assets is crucial to determine the allocation to the bankruptcy estate (this is particularly important for the allocation of the plaintiff's role in the dispute, i.e. whether the creditor must sue for separation from the bankruptcy estate or the bankruptcy estate must sue for inclusion). The following rule applies: the assets only fall into the bankruptcy estate by default if the debtor has exclusive custody. In all other cases, the assets do not fall into the bankruptcy estate and the estate must sue the third party for inclusion.[66] If the ownership of the third party claimant is to be regarded as proven from the outset, the bankruptcy trustee can hand over the item before the bankruptcy proceedings are completed (Art.51 Ordinance on the Management of Bankruptcy Offices [KOV; SR 281.32]).

[47] In the case of things, two prerequisites are necessary in order for them to be assigned to the bankruptcy estate from the outset:

- The thing belongs to the debtor alone, which means it is not subject to third party ownership, co-ownership, or collective ownership.
- If ownership of the item is disputed, the item must also be under *exclusive* control of the debtor.

4.5.7 Analogous Application to Bitcoin

[48] As explained above, bitcoins are a "*rival, fictitious, immaterial asset sui generis*". Bitcoins have all the typically properties of things, with the exception of a material body. In the absence of specific legal regulations, an analogous application of the law suggests itself and leads to the legally and economically desirable outcome. Observing the concepts of *power of disposal* and *right of disposal* is essential when doing so.[67]

[49] In order to correct deficiencies in the assignment to the bankruptcy estate, the law specifies inclusion and separation procedures. Without such procedures, there would be a risk that assets that do not belong to the debtor are assigned to the bankruptcy estate and vise versa, which would be contrary to the purpose of the law.

[66] Kren Kostkiewicz (fn. 40), N 1304.

[67] According to Art. 197 SchKG, everything that belongs to the debtor forms a single estate, "regardless of where it is located". This is an additional indication that the location (which, in the case of Bitcoin, corresponds most closely to the address and the associated power of disposal) is not the decisive feature of the assignment, but that the actual ownership (in the case of Bitcoin, the right of disposal) is decisive.

[50] This leads to a further argument in favor of allowing the separation of bitcoins in analogy to things. The publicity of owning things is more apparent than that of bitcoins: without any indication to the contrary, things in the possession of the debtor regularly appear to third parties as the debtor's property. A creditor thus runs the risk of overestimating the estate of the debtor and therefore also their creditworthiness. The separation of items from the bankruptcy estate breaks the rule that what appears to third parties as the property of the debtor also belongs to the bankruptcy estate. Nevertheless, the legislature has (rightly) allowed a segregation of things.

[51] The pseudonymous nature of the Bitcoin blockchain normally requires active involvement of the custodian for the publicity effect to unfold, for example by demonstrating the power of disposal through a transaction. The risk that a creditor would incorrectly attribute bitcoins held with a custodian to the custodian is therefore significantly lower than in the case of other forms of property. Allowing for the segregation in the case of bitcoins constitutes therefore a smaller interference with the protection of creditors than in the case of other forms of property. Given that the law already allows things to be separated, this must be possible all the more for bitcoins according to the presumed will of the lawmaker.

[52] These finding imply the following consequences for the treatment of bitcoins in various cases:

- *Exclusive power of disposal with right of disposal:* the debtor can dispose of the bitcoins without the help of third parties and the bitcoins economically and contractually belong to the debtor. The bitcoins fall into the bankruptcy estate.

- *Exclusive power of disposal without right of disposal:* The debtor has sole control over some Bitcoins that belong to a third party. In case the right of disposal of the third party is evident, the Bitcoins can be handed out in bankruptcy. Otherwise, they fall into the bankruptcy mass and the third party must try to get them back by means of a segregation action.

- *Non-exclusive power of disposal with right of disposal:* The debtor does not have sole control over the bitcoins, but the bitcoins are economically attributable to the debtor. If the right of disposal is disputed, the bitcoins are not part of the bankruptcy estate due to the lack of exclusive custody, but can be included by means of an inclusion action.

- *Non-exclusive power of disposal without right of disposal:* the debtor cannot alone or not be the only one to dispose of the bitcoins, and the bitcoins are economically attributable to a third party. Due to the lack of exclusive custody and the lack of right of disposal, the bitcoins do not fall into the bankruptcy estate. If the person entitled to dispose of the bitcoins cannot dispose of them without the help of the debtor, the bitcoins are to be transferred to the entitled person.

[53] The clarification of the term "belong" to Bitcoin in the context of Art. 197 SchKG is a mandatory prerequisite for the formation of the bankruptcy estate. The question here is simply what is the correct interpretation of the term, not whether an interpretation needs to be made. This stands in contrast to the questions of segregation and inclusion, where one first needs to be establish whether there is a gap in the law that ought to be filled. If this is affirmed and the thing-like nature of bitcoins is recognized, then separation and inclusion should also be possible for bitcoins in accordance with the purpose of the law.

4.6 Release of Bitcoins through the Bankruptcy Trustee

[54] The release of bitcoins or other crypto assets to the entitled person might pose a practical problems to the bankruptcy trustee as the correct handling of crypto assets requires a basic level of technical expertise. In the case of tangible items, it is usually up to the owner to travel to their location to pick them up. Analogously, it would seem reasonable to impose an obligation on the recipients to cover the costs of handling and transferring their bitcoins.

4.7 Evaluation of Two Example Case

4.7.1 Tezos Example

[55] The example described in section 2 formidably illustrates the tension between the law and the purpose of the law. If, purely hypothetically, Bitcoin Suisse AG were to go bankrupt, the bankruptcy office would be faced with a delicate task: Should it

(i) rely on a grammatical interpretation of the legal text and accept results that go against the immanent purpose of the law as well as the economic intent and the good faith of the parties involved; or (ii) follow the presumed intent of the legislator by filling the identified legal gap and taking the economic circumstances into account?

[56] When establishing the inventory of the bankruptcy estate, the bankruptcy administrator would find a storage device on which one of the two private keys necessary to access the relevant bitcoins is located. The key found on the premises of Bitcoin Suisse AG alone does not grant any power of disposal - the bitcoins can only be disposed of in combination with the private key of the Tezos Foundation. The initial step of the bankruptcy office is undisputed: the storage device as a tangible object belonging to the bankrupt entity falls into the bankruptcy estate. Then, the bankruptcy office has several options:

- *Treating the private key as general data I:* In order to avoid violating any business secrecy or personal data protection rights, all storage devices are erased and recycled as a measure of precaution. The assets of the Tezos Foundation would thus be lost forever. This would neither benefit the creditors of the bankruptcy estate nor the Tezos Foundation.

- *Treating the private key as general data II:* The storage device and with it the stored data is auctioned off to the highest bidder.[68] The economic value of the storage device is primarily determined by the value of the private key it contains. How much does the highest bidder expect to receive from the Tezos Foundation for issuing the private key? This is an interesting question from a game-theoretic perspective and would maximize the benefit of the creditors of the bankruptcy estate. But besides ethical (if not even criminal) concerns, such an auction would lead to an absurd result from an economic point of view: if the proceeds from the sale were roughly half of the value of the assets secured with the two keys, the balance sheet of the bankruptcy estate would increase by about half a billion Swiss francs and suddenly be in balance again, with assets significantly exceeding liabilities. In other words: A valuation of Bitcoin Suisse AG based on the liquidation value would be significantly higher than a valuation based on "going-concern" principles. Alternatively, Bitcoin Suisse AG would have to include the expected proceeds from the sale in its balance sheet today - which would also hardly make sense to implement due to

[68] For the sake of simplicity, it is assumed that the storage device is not encrypted or auctioned together with the necessary access codes.

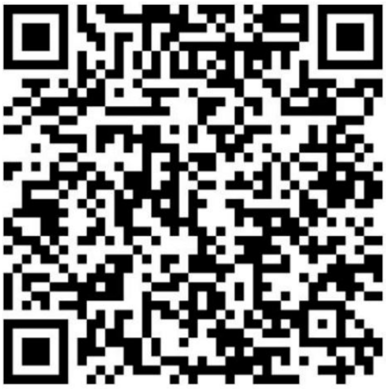


Figure 4.7.1: Private Key. A QR code representing the private key for address 17sXedYBwciV9phuhX1HtHEXSjnZxHvHME.

the lack of accessibility. The expectations of the creditors must also be taken into account: When concluding transactions with Bitcoin Suisse AG, nobody should have expected that part of the assets of the Tezos Foundation would belong to Bitcoin Suisse AG.

- *Treating Bitcoin as "rival, fictitious, intangible asset sui generis" with analogous application of property laws:* The bankruptcy trustee recognizes the economic intent of the two parties and includes the bitcoins of the Tezos Foundation in the inventory with a reference to the third-party claim. As the circumstances are undisputed, the bankruptcy trustee, in collaboration with the Tezos Foundation and with the help of specialist staff, transfers the bitcoins to an address that is under the sole power of disposal of the Tezos Foundation. If there are doubts about Tezos Foundation's right of disposal over the bitcoins, the bankruptcy estate would have to sue for inclusion. The Tezos Foundation reimburses the bankruptcy estate for the expenses incurred.

4.7.2 Additional Example

[57] At the time of writing the initial version of this article, the private key shown in figure 4.7.1 provided power of disposal over 0.01 bitcoins belonging to the authors of this article. It was stolen immediately publication. Today, the address is empty.

[58] Publishing a private key is about as sensible as leaving a bicycle unlocked at the train station, and the question is how long it will take for someone to appropriate these 0.01 bitcoins. But this example helps to illustrate the question at hand: What goes into the bankruptcy estate of Example AG if, before going bankrupt, it

I prints out this article?

II saves is to Google Drive?

III transfers the bitcoins to an address under its sole control?

[59] According to the views presented in this article, the bitcoins do not fall into the bankruptcy estate in cases 1 and 2 due to a lack of exclusivity of the power of disposal. In the absence of a right of disposal, there is also no possibility of adjustment. In the third case, the bitcoins are inventoried with the note that a third party has the right of disposal, but do not fall into the bankruptcy estate (if documented accordingly). If the persons entitled to dispose are not apparent, the right of disposition is presumed to lie with Example AG due to their exclusive power of disposition and the bitcoins fall into the bankruptcy estate. However, the authors could demand that the data be handed over or separated out if they can rebut the presumption and prove that they are still the persons entitled to dispose of it.

4.8 Retention Best Practices

[60] Based on these considerations, best practices for the storage of cryptocurrencies can be formulated. In our opinion, the following is important:

- The contractual relationship with the client must make clear that the bitcoins are kept by the custodian on the clients behalf and that they are not converted into a deposit or other monetary claim against the custodian upon delivery.
- For the effective protection of creditors, one must not create the impression that the stored bitcoins would belong to the custodian.
- For bitcoins in the custodian's power of disposal, the custodian must at all time be able to determine who these bitcoins belong to.
- Client assets can, but need not, be segregated directly on the blockchain by using separate addresses per client. If the segregation takes place at a higher level, it is advantageous not only to do this in terms of accounting, but also to support it with suitable technical means. In the case of collective custody, the client's consent is required.

- Ideally, the client should be able to find out which bitcoins are assigned to him at a given point in time. This assignment of bitcoins to a client should be stable as far as operational requirements allow.
- The most important segregation is that of client assets from custodian assets. This makes it clear which bitcoins belong to the custodian and therefore would also fall into the bankruptcy estate before determining in detail which clients they belong to.
- If bitcoins are lost and the custodian is liable for the loss, the custodian should consequently only be liable towards the client to whom the lost bitcoins were assigned at the time of the loss. This stands in contrast to the usually taken even distribution of the loss among the clients, like Bitfinex did in the summer of 2016, for example. In such cases, it would be desirable to have insurance coverage.

[61] A technically and legally interesting but, depending on the operational requirements, also very complex type of storage is the distribution of the power of disposal over several parties, for example by means of 2-out-of-3 multi-signature addresses, the keys of which are held by independent parties. This means that the client can continue to exercise his right of disposal even in the event of bankruptcy of one of the parties without the involvement of the bankruptcy administrator. Since the bitcoins are under custody, they do not fall into the bankruptcy estate anyway. A less complex variant of co-custody is depositing backups of the keys with appropriately instructed third parties or with the respective client. In both variants, however, the involvement of additional parties also results in additional risks, so that this option must be carefully evaluated before it is implemented.

[62] Ultimately, every measure must make sense in its respective context. In innovative business areas in particular, one should not blindly rely on best practices or other recommendations. Accordingly, these best practices can only serve as a guide. They are not a substitute for your own expertise.

Chapter 5

Classification of Cryptocurrency Staking under Financial Market Law

Co-authors: Kilian Schärli, Luzius Meisser, Reto Luthiger

Publication: Schärli, K., Meisser, L., & Luthiger, R. Finanzmarktrechtliche Einordnung des Stakings von Kryptowährungen. In: *Jusletter IT* 2021, September 30. doi: 10.38023/ee56b560-ad84-4065-901c-bb6369c1a719

5.1 Abstract

“Staking" is the use of cryptocurrencies or other crypto assets as collateral for the purpose of actively participating in a blockchain-based system with a decentralised organisation. Due to the technical complexity, holders often do not “stake" their crypto assets themselves but transfer them to an intermediary with the necessary operational expertise. The article discusses the consequences of staking on the separability of the deposited crypto assets and derives the requirements under financial market law that the service provider must fulfil. The authors conclude that crypto assets can be used as collateral and therefore also for staking without compromising their separability. If crypto assets of clients are held collectively during staking, professional service providers usually require a banking or fintech licence.

5.2 Introduction

[1] In September 2020, Parliament adopted a set of legislative amendments to provide a better legal foundation for various applications of blockchain technology.[1] Together with the associated covering ordinance, these amendments bring about various improvements to the Swiss legal framework in connection with the use of decentralised technologies and crypto-based assets or crypto assets.[2] A total of ten federal laws were amended, including the Federal Debt Collection and Bankruptcy Act (SchKG) and the Banking Act (BankG).

[2] The SchKG now expressly regulates the segregation of crypto-based assets and data from the bankruptcy estate. This recognises the property-like characteristics of crypto assets under bankruptcy law. Some voices would have welcomed an even more fundamental classification as chattel according to art. 713 of the Swiss Civil Code (ZGB); however, this was rejected due to the legal consequences that were difficult to foresee.[3] At the same time, the banking insolvency provisions of the BankG were harmonised with the amendments to the SchKG and the “fintech licence" under art. 1b BankG, which had been little used until then, was upgraded. In the SchKG, the legislator has explicitly granted the holders of crypto assets held in collective custody the right to separation. The depositories of cryptocurrencies in collective custody require a licence under the BankG for their activities. This fits into the concept of the fintech licence, which was already geared to the safekeeping of client assets without the associated differential interest business. At the same time, this addressed the concern that, due to the changes in the SchKG, “bank-like" business models based on collective safe custody[4] would not have been subject to any supervision by FINMA, even if they were large in scale.[5]

[3] This article focuses on the very specific issue of the right to separate staked crypto assets and the associated consequences under financial market law. Staking received little attention at the time of the consultation on the draft law and can be

[1] The authors would like to thank MLaw Julia Pugliese for her valuable assistance with this article.

[2] In this article, the terms “crypto-based assets" and “crypto assets" are used synonymously.

[3] See for example Ronald Kogens/Catrina Luchsinger Gähwiler, Blockchain: neue Rechtsgrundlagen müssen wasserfest sein, in: NZZ dated 20 June 2019, www.nzz.ch/meinung/herausforderungen-von-blockchainld.1488163?reduced=true.

[4] “Cryptocurrencies" are crypto-based assets or crypto assets of a payment nature.

[5] Federal Department of Finance FDF, Consultation procedure on the Ordinance on the Adaptation of Federal Law to Developments in the Technology of Distributed Electronic Registers, Results Report dated 18 June 2021, P. 3 (cited EFD Results Report 2021).

seen as the first major test of the future viability of the legislative package.[6] The core question is how the process of staking with the help of a third party can be classified economically and legally.

5.3 Staking of Crypto Assets

5.3.1 What is Staking?

[4] A blockchain is a decentrally organised data structure that allows globally consistent execution of transactions. Its most important application is the protection of cryptocurrencies and other crypto assets. Their decentralised nature necessitates a consensus-building mechanism within the system.[7] There are two main methods that are important for public blockchains: “proof-of-work" and “proof-of-stake". Unlike the proof-of-work method, validation in the proof-of-stake method is not based on computing power (“mining"), but on the use of crypto assets by the network participants (“staking").[8] Proof-of-stake is significantly more energy-efficient than proof-of-work and is generally considered the more sustainable option.[9]

[5] Only those who are willing to provide a minimum amount of crypto assets as collateral and immobilise them for the duration of their participation in the system may actively participate in systems based on proof-of-stake. This “blocking" serves to punish the participant by destroying a part of his crypto assets if he or she violates the rules of the system (“slashing"). At the same time, the participant receives a reward (“staking reward") for system-compliant behaviour in line with the majority of participants.[10] While rewards are continuously credited, slashing, for example, due to a deliberate rule violation or operational error, is very rare in practice.

[6] In the context of the consultation, only the Swiss Blockchain Federation SBF referred to the issue of staking. The SBF already assumed that staking should not affect the separability of crypto assets. This assumption was not contradicted in the Federal Council’s consultation report - but it was not confirmed either, fedlex.data.admin.ch/filestore/fedlex.data.admin.ch/eli/dl/proj/6019/15/cons_1/doc_5/de/pdf-a/fedlex-data-admin-ch-elidl-proj-6019-15-cons_1-doc_5-de-pdf-a.pdf

[7] DANIEL RUTISHAUSER/RALF KUBLI/Rolf H. WEBER, Grundlagen, in: Rolf H. Weber/Hans Kuhn (Hrsg.), Entwicklungen im Schweizer Blockchain-Recht, Basel 2021, P. 9 et seq., N 22.

[8] RUTHISHAUSER/KUBLI/WEBER (Fn. 7), N 27.

[9] In this context, see already Postulate 21.3199 Molina “Climate protection and cryptocurrencies. Promoting energy-efficient blockchain technologies", which points out the enormous energy consumption of proof-of-work blockchains and the related climate problem.

[10] Bitcoin Suisse, What is staking?, 27 October 2020, www.bitcoinsuisse.com/de/fundamentals/was-ist-staking.

[6] Those who wish can also participate several times and therefore increase the reward proportionally to the capital invested. This suggests the fallacy that the staking reward is an interest rate for lending capital to a third party. However, this analogy must be rejected on closer examination, because the capital invested is not transferred to a third party and remains allocated to the holder in the system, albeit in a blocked state. In economic terms, this transaction is therefore not a loan, but the use of an asset as collateral.

[7] For the sake of completeness, it should be noted that, unlike traditional collateral, for example to secure a loan, the stake in slashing is not realised but destroyed. So even in the event of slashing, no third-party gains power of disposal over the blocked crypto assets. In our opinion, however, it should make no difference in the further analysis whether the crypto assets are utilised or destroyed in the event of slashing. The decisive factor is that they are not used by any third party as long as they serve as collateral.

5.3.2 Different Forms of Staking

[8] Staking takes on different forms. In the simplest case, the holder of the crypto assets and the operator of the technical infrastructure are one and the same person. Legally more interesting, on the other hand, are cases in which the holder of the crypto assets and the technical operator of the network node are not identical.

[9] Contractually, a service provider can structure staking to apply as a deposit under banking law. This is the case if the holder transfers the cryptocurrencies in their entirety to the operator, thereby only creating a claim in the amount of the transferred cryptocurrencies, but not a right of separability. In general, however, it is more advisable to deposit the crypto assets with the operator in a separable manner.

[10] Some systems have explicit functions for the appointment of a third party as operator by the holder (“nominated" or “delegated" staking). In this case, the holder does not have to give the operator power of disposal over the crypto assets but remains exposed to the risk of slashing if the operator makes a mistake. In delegated staking, it is already apparent from the data available on the blockchain that the delegated service provider is acting on behalf of his client. In contrast, the question of whether the operator is staking in his own name or in the name of the client can only be answered on the basis of the present contractual relationship.

[11] The minimum stake in staking is often significant. In the Ethereum system, it is 32 Ether and, therefore, currently around CHF 100,000. For this reason, joint or non-segregated staking is also offered in practice (“pooled" staking). Although this variant is legally the most demanding, it is often the most sensible economically.

5.4 Bankruptcy Appraisal

5.4.1 Legal Development

[12] Crypto assets are often not held in custody by the beneficial owner himself, but by a corresponding service provider. Depending on how it is structured, safekeeping by a depository can pose various advantages for the holder of crypto assets, such as the secure storage and management of private keys or the simple exchange of crypto assets for state currencies.[11]

[13] Whether or under what conditions the crypto assets in question could have been separated in the event of the bankruptcy of the depository on the basis of art. 242 SchKG and to what extent there was a gap to be filled is disputed in the literature and has not been clarified by case law.[12]

[14] The legislator has eliminated this legal uncertainty with the introduction of art. 242a and 242b SchKG by legally anchoring the material prerequisites for the separation of crypto-based assets (hereinafter often referred to simply as “crypto assets") in the bankruptcy of the depository. In this respect, it must be assumed that art. 242a and 242b SchKG create a final regulation for the separability of crypto assets, which is why art. 242 SchKG can no longer be invoked for the separation of crypto assets.

[15] Under certain conditions, the clients of such service providers now enjoy an explicit right to separation and therefore a similar legal position under bankruptcy law as the holders of objects held in custody.[13]

[11] STEFAN KRAMER/DOMINIC WYSS, Verwahrung von digitalen Aktiven, in: Rolf H. Weber/Hans Kuhn (Hrsg.), Entwicklungen im Schweizer Blockchain-Recht, Basel 2021, P. 145 et seq., N 5.

[12] Message on the Federal Act on the Adaptation of Federal Law to Developments in the Technology of Distributed Electronic Registers of 27 November 2019, BBl 2020 233, P. 265 with references to relevant doctrinal opinions.

[13] BBl 2020 (Fn. 12), P. 291 f.

5.4.2 Separability According to Art. 242a SchKG

[16] Provided that the client does not have his own access to the crypto assets and the depository has all the necessary keys to access the crypto assets, the crypto assets would fall into the bankruptcy estate in the event of the bankruptcy of the depository, unless otherwise specified.[14] For clients who cannot directly access their crypto-based asset rights due to a lack of (exclusive) actual power of disposal,[15] The right of separation is legally enshrined in art. 242a SchKG is, therefore, of central importance.

Art. 242a

1. *The bankruptcy trustee shall make an order for the surrender of crypto assets over which the bankrupt party has the power of disposition at the time of bankruptcy, and which are claimed by a third party.*

2. *The claim is well-founded if the bankrupt party has undertaken to keep the crypto assets for the third party at all times and these:*

   a *are individually assigned to the third party; or*

   b *are assigned to a community and it is evident which share of the community assets the third party is entitled to.*

3. *If the bankruptcy trustee considers the claim to be unfounded, a time limit of 20 days shall be set for the third party to file an action with the court at the place of bankruptcy. If this time limit is not met, the claim is forfeited.*

4. *The costs of surrender shall be borne by the person who requests it. The bankruptcy trustee may demand a corresponding advance payment.*

[17] Pursuant to art. 242a para. 1 SchKG, crypto assets are the subject of the separation proceedings. For the purposes of this provision, crypto assets are all assets for which the power of disposition is exclusively conveyed by means of a

[14] BBl 2020 (Fn. 12), 291 f.; KRAMER/WYSS (Fn. 11), N 38; DOMINIK Vock/DAVID MEIRICH, Aussonderung kryptobasierter Vermögenswerte und Zugang zu Daten im Konkurs, DLT-Mantelverordnung schafft Rechtssicherheit im SchKG, September 2021, www.mme.ch/de/magazin/aussonderung_krypto-basierter_vermoegenswerte_und_zugang_zu_daten_im_konkurs/.

[15] Actual power of disposal is to be assumed, for example, if (1) the access key is known only to the client and only the client can directly dispose of it, if (2) the client and the depository have the identical access key and therefore both have direct access or (3) if there is a so-called multi-signature address. In these cases, the crypto assets do not form part of the bankruptcy estate.

cryptographic process.[16] All blockchain-based tokens are therefore covered by the separation rule - irrespective of their design as a register value right.[17]

[18] According to art. 242a SchKG, the claim against the bankruptcy estate for the separation or transfer of the crypto assets concerned is subject to two (cumulative) conditions. Firstly, the bankrupt depository must have given an undertaking to the client to keep the crypto assets available for him at all times (para. 2). Secondly, it must be possible to allocate the value units held either to the client individually (para. 2 lit. a) or to a community (para. 2 lit. b).

5.4.3 Obligation of Availability (Art. 242a para. 2 SchKG)

[19] With the first condition for separability under art. 242a SchKG, the obligation of the depository to keep the crypto assets available for the client at all times, it becomes clear that separability depends on the contractual relationship between the parties. These can choose via the structure of their contractual relationship whether the transfer of crypto assets to the service provider constitutes a non-separable deposit under banking law or whether a separable custody exists. The decisive factor here is the obligation of the depository to keep the crypto assets available at all times. Whether this obligation is actually complied with seems to be irrelevant for the assessment of separability under SchKG.

[20] The contractual relationship can be structured in such a way that it is possible at any time to convert crypto assets held in custody into deposits with the client's consent and vice-versa, provided that the respective requirements are met. Likewise, the separability of the crypto assets in question is not affected if crypto assets are exchanged for other crypto assets, reallocated to different clients or otherwise replaced with the client's consent, provided that the respective requirements for the crypto assets resulting from these processes are met.[18]

[21] However, the question arises as to when the obligation to hold crypto assets available is breached by immobilisation or other restriction of the power of disposal. Such a breach of duty is likely to occur at the latest when the power of disposal

[16] BBl 2020 (Fn. 12), P. 292.

[17] DOMINIC WYSS, Gegenstand und Übertragung von DLT Wertrechten, Gemäss den vorgesehenen Gesetzesanpassungen im Zusammenhang mit verteilten elektronischen Registern, in: Jusletter 1 July 2019; KRAMER/WYSS (Fn. 11), N 37; STEFAN KRAMER/Urs MEIER, Tokenisierung von Finanzinstrumenten, in: GesKR 1/2020, P. 60 et seq., P. 72.

[18] BBl 202 (Fn. 12), P. 292; KRAMER/MEIER (Fn. 17), 73; KRAMER/WYSS (Fn. 11), N 40.

over the crypto assets is transferred to a third party. This means that crypto assets are treated differently from physical objects in this context, but analogously to securities. For example, if a garage owner lends a car entrusted to him to a third party, the client's ownership rights to the car remain unaffected. On the other hand, a client of a depository of crypto assets would lose his separation rights against the depository if the depository transfers the crypto assets to a third party. The same applies to securities lending.[19]

[22] In contrast, when securities are used as collateral, for example for a Lombard loan, the client retains his ownership rights.[20] Likewise, it is permissible to use crypto assets as collateral in favour of the client. When the realisation event occurs, the crypto assets are also transferred to a third party, but at this point the client is no longer the holder. Therefore, the depository has complied with the requirement to keep the crypto assets available at all times during the entire period of the client's entitlement to the crypto assets and has fulfilled his obligation. Any immobilisation during this period should not constitute a breach of duty, provided that this is done with the consent of the holder.

5.4.4 Assignability (Art. 242a para. 2 SchKG)

[23] The second prerequisite is the assignability of the crypto assets to the client. Such assignability requires that the crypto assets held are actually present. The criterion is therefore also decisive for the fulfilment of the obligation to be available at all times. Assignability ensures that in the event of bankruptcy it is evident who owns which crypto assets and therefore represents a natural prerequisite for separation. It can be fulfilled in two ways:

[24] In analogy to property law, art. 242a para. 2 lit. a SchKG provides that the crypto assets must be individually assignable to the client at the time of bankruptcy. Such individualised allocation can be achieved by holding the crypto assets on a system address assigned to the client, i.e. technically directly in a wallet segregated on the blockchain.[21] According to almost unanimous criticism[22] in the consultation

[19] ALEXANDER VOGEL/CHRISTOPH HEIZ/RETO LUTHIGER (Hrsg.), in: FIDLEG/FINIG Kommentar, Bundesgesetz über die Finanzdienstleistungen und Bundesgesetz über die Finanzinstitute und weiteren Erlassen, Zürich 2020, art. 19 N 1 et seq.

[20] ELISABETH MOSKRIC, Der Lombardkredit, in: SSBR, Vol. 74, Zurich 2003, P. 84.

[21] BBl 2020 (Fn. 12), P. 293.

[22] BBl 2020 (Fn. 12), P. 245 f. and 251; a.M. FINMA, cf. BBl 2020 (Fn. 12), P. 246 f.

on individual assignability in the register at any time as a prerequisite for the assignment, however, it is now already sufficient if the assignment of specific crypto assets results from an internal register (e.g. an accounting system) of the depository.[23] Therefore, the custody of the crypto assets in a separate wallet can even be dispensed with altogether if it is technically possible for the depository to individualise the crypto assets in question in another way, for example by means of a separate serial number. In such cases, it is sufficient that the crypto assets specified with numbers can be assigned to the client concerned by means of an assignment table available from the bankrupt party.[24] Such individual assignment is likely to be possible especially in the case of NFTs held in collective custody.

[25] Art. 242a para. 2 lit. b SchKG also provides for possible separation in the case of crypto assets held in collective custody. The regulation applies if the crypto assets cannot be individually assigned to the authorised client but belong to a community. It must be clear which share of the joint assets each client is entitled to. As in the case of individual assignability, it is sufficient in the case of collective custody that the assignment results from an internal register of the depository. Analogous to deposit relationships, the share of the crypto assets still held in a collective account is separable in each case.[25] This regulation applies, for example, to collectively held bitcoins.

[26] It is therefore crucial that the value units owed are kept at the client's disposal at all times. This regulation is based on the idea that the clear separation of crypto assets in the event of the insolvency of the depository should primarily serve to protect the client. It does not matter whether the crypto assets are held in an individual account or a collective account,[26] as long as they are assignable to the specific client and are actually present.

5.4.5 Effect of Staking on Separability

[27] The law is designed to protect clients in the best possible way. This is done by granting them far-reaching rights of separation and preventing proprietary trading

[23] The message repeatedly emphasises that it was expressly refrained from requiring individual assignment in the register at any time for the surrender of the assets. The objectives of the revision could also be achieved if assignment takes place outside the actual registry, cf. BBl 2020 (Fn. 12), P. 245 f., 265 and 293.

[24] BBl 2020 (Fn. 12), P. 293; KRAMER/MEIER (Fn. 17), P. 73.

[25] BBl 2020 (Fn. 12), 293; Kramer/WYSS (Fn. 11), N 41; Vock/MEIRICH (Fn. 14).

[26] BBl 2020 (Fn. 12), P. 247.

by the depository with the crypto assets held.[27] In this context, it is to be assumed that lending or proprietary business, by analogy with art. 1b para. 1 lit. b BankG, means lending of value, transfer to ownership or to a limited right in rem in the sense of banking asset business.[28] If the terms and conditions of business allow the depository to conduct transactions on his own account, it is no longer possible to speak of an obligation to keep the assets available at all times, which means that the prerequisites for separability are no longer fulfilled and a claim or deposit now exists. However, in the case of staking for the account and at the risk of the client, one cannot speak of a proprietary transaction of the depository, and it may therefore be assumed that staking for the account of the client does not stand in the way of separability from this point of view. The same should generally apply to the provision of collateral for the account of the client.

[28] Furthermore, immobilisation during staking should not stand in the way of separability. This already follows from the wording of art. 242a SchKG, which stipulates that crypto assets must be assignable. Accordingly, the only decisive factor is that the assets are available at all times and not that they can be moved at any time. This deliberately chosen formulation was based on the legislator's intention, as just described, to prevent asset or proprietary transactions and at the same time enable the greatest possible separability at the level of debt and bankruptcy law in order to strengthen the rights of the clients. The same result is reached when looking at similar constellations; constellations in which crypto assets are subject to a legally prescribed, contractually agreed or other lock.[29] These cases in particular make it clear that it could not have been the intention of the legislator to prevent separation in the event of bankruptcy.

[29] Separation of crypto assets held for the account of a client in the event of bankruptcy is therefore possible, irrespective of whether the assets are held individually (at separate addresses per client) or collectively (at a common address for several clients), provided that they can be assigned to the client. This is also not changed by any lockup period. As long as the bankrupt depository has the (de facto) power of disposal, e.g., through the private keys located at the depository,

[27] BBl 2020 (Fn. 12), P. 265 f. and 292 f.

[28] Cf. KRAMER/MEIER (Fn. 17), P. 73 Fn 123; EFD Results report 2021 (Fn. 5), P. 34.

[29] Examples: (i) tokens frozen by the depository for compliance reasons to fulfil AML obligations, as the depository can continue to dispose of them at any time and the AML serves as a justification for it, (ii) crypto assets acquired by the client that are subject to a lockup period determined by the protocol (e.g. LQTY tokens), (iii) crypto assets acquired by the client that are subject to a lockup period contractually determined with a third party (e.g. BZZ tokens).

and therefore, the crypto assets can be issued by the bankruptcy trustee, e.g., by handing over the private keys, separation must be possible.

5.4.6 Separability Acc. to Art. 242b SchKG

[30] The power of disposal over the crypto assets assigned to an address is exercised by means of one or more cryptographic keys (private key). As mentioned above, only those crypto assets to which the beneficiary does not have his own access and for which the depository has all the necessary keys to be able to dispose of them directly fall into the bankruptcy estate. If, however, the bankrupt party does not have the necessary keys to be able to dispose of the assets directly himself, surrender on the basis of art. 242a SchKG is out of the question.

[31] In addition to the right to separation for crypto assets under art. 242a SchKG, the legislator has therefore created art. 242b SchKG, which grants the custody client a legal right of access to data and therefore to private keys and addresses over which the bankruptcy estate has power of disposal. The power of disposal over the crypto assets can therefore alternatively also be secured by issuing the private key.

Art. 242b

1. *If data are in the bankruptcy estate's power of disposal, any third party who proves a legal or contractual entitlement to the data may, depending on the nature of the entitlement, demand access to the data or their release from the bankruptcy estate's power of disposal.*
2. *If the bankruptcy trustee considers the claim to be unfounded, the third party shall be set a time limit of 20 days within which it may file an action with the court at the place of bankruptcy. The data may not be destroyed or used until the court's decision has become final.*
3. *The costs for access to the data or for their release shall be borne by the party who requests access to the data. The bankruptcy trustee may demand a corresponding advance payment.*
4. *The right to information under the data protection provisions of the Federal Government or the cantons is reserved.*

[32] Based on art. 242b SchKG, data must be released to the applicant if there is a legal or contractual entitlement. In the case of the safekeeping of crypto-based assets by a depository, there is a contractual claim against the wallet provider in

each case, whereby the private key can be separated. The only prerequisite is that this contractual claim was already established before the opening of bankruptcy proceedings - i.e., the client already had a claim to access to the data or the private key prior to the opening of the bankruptcy proceedings, the access to the data does not lead to an unjustified devaluation of the bankruptcy estate and the claim is due,[30] whereby it is irrelevant whether the deposited crypto assets are staked or not.

5.5 Financial Market Law Appraisal

[33] The following section describes the extent to which staking can be carried out without a licence (4.1) and the situations in which a fintech or even bank licence is required (4.2).

5.5.1 Unauthorised Staking

[34] In order to assess whether staking without license is permissible, it is first necessary to briefly discuss some principles of the regulation of banks and fintech companies. Banking regulation is concerned, among other things, with minimising the risk of default by the depository for depositors, which is why it is particularly relevant whether (i) the depositor incurs a risk of default in the event of the bankruptcy of the depository, whether (ii) the depositor has a right of separation in respect of his assets, or whether (iii) his need for protection is reduced[31] for other reasons.[32] In the event of separability, the investor's need for deposit protection is usually denied and neither the bank licence nor the fintech licence for public deposits applies. The fintech license for cryptocurrencies held in collective custody already includes per se the separability of the cryptocurrencies held in collective custody in the event of bankruptcy.

[35] In analogy to the practice described above for the non-applicability of the deposit regulation in the event of separability in bankruptcy, before the DLT legislation came into force on 1 August 2021, FINMA's practice was that a banking licence was not

[30] BBl 2020 (Fn. 12), P. 295 f.

[31] Typical other grounds can be found in the exception catalogues of art. 5 para. 2 and 3 BankV.

[32] Cf. NINA REISER, Ist der Bankbegriff im Lichte aktueller technologischer Entwicklungen noch zeitgemäss?, in: AJP 7/2018, P. 811 et seq., P. 815 with additional references.

required for virtual currencies under the strict conditions that the assets in virtual currency (e.g., Bitcoin) could only be transferred for safekeeping and that these virtual currency units could be held separately per client on the blockchain and assigned to individual clients at any time.[33]

[36] This FINMA practice was developed at the time in an act of official gap-filling. On the one hand, art. 197 SchKG states that only what falls into the bankruptcy estate also “belongs" to the debtor. This suggests that crypto assets held on behalf of clients do not fall within the bankruptcy estate. On the other hand, however, the owner lacked the instrument of an action for separation under art. 242 SchKG, since this presupposes corpo-reality.[34] The legislator has filled this legal gap with the two new art. 242a and 242b SchKG as lex specialis with regard to crypto assets and data.

[37] Particularly in the following cases, staking can in principle be operated as a bank or fintech company without a licence:

a The requirements of art. 242a para. 2 lit. a SchKG or art. 16 para. 1^{bis} lit. a BankG regarding individualisability and availability at any time are fulfilled; or

b the requirements for the separation of the private keys pursuant to art. 242b SchKG are met; or

c the crypto-based assets do not meet the requirements of art. 16 para. 1^{bis} lit. b BankG, but qualify as a public deposit and an exemption from the public deposit or from the deposit pursuant to art. 5 para. 2 or 3 BankV applies; or

d the concept of commercial activity (incl. sandbox regime up to CHF 1 million) is not fulfilled.

[38] Without going into more detail here, it should be noted that in most of these cases there is at least an obligation to submit to the Money Laundering Act (GwG).

[33] Swiss Financial Market Supervisory Authority FINMA, Fact Sheet Virtual Currencies, as of 1 January 2020, P. 2.

[34] CHRISTIAN MEISSER/LUZIUS MEISSER/RONALD KOGENS, Verfügungsmacht und Verfügungsrecht an Bitcoins im Konkurs, in: Jusletter IT 24 May 2018.

5.5.2 Authorised Staking

[39] A bank engages in interest rate derivatives business with maturity transformation, i.e., by accepting deposits from the public on the liabilities side of the balance sheet, it enters into short-term obligations against payment of an interest rate, with which it grants medium to long-term loans on the assets side of the balance sheet against payment of an interest rate. Fintech companies often only engage in deposit-taking business, which is why they lack the maturity transformation typical of banks and, in particular, the associated liquidity and interest rate risks, and why the licensing requirements of the Banking Act are also very excessive.[35] For this reason, the legislator created the fintech licence in art. 1b BankG for the acceptance of public deposits of up to CHF 100 million, which came into force on 1 January 2019 and, in contrast to the banking licence, imposes lower requirements in terms of or organisation, minimum capital, capital and liquidity requirements, accounting, auditing and deposit insurance (fintech licence).

[40] As explained above, separation based on art. 242a SchKG is also possible in the case of crypto assets held in collective custody, provided that it is evident to which share of the joint assets the respective custody client is entitled. This separability means that the crypto assets are off-balance sheet and therefore do not qualify as a public deposit, which is why no banking licence is required. Since the legislator considered this circumstance to be questionable[36] from the perspective of investor protection as well as from the perspective of a level playing field for institutions that accept customer deposits and thus require a licence, it was decided to extend the fintech licence to collectively held cryptocurrencies or crypto assets that actually or according to the intention of the organiser or issuer serve to a significant extent as a means of payment for the acquisition of goods or services or the transfer of money or value (art. 5a para. 1 BankV in connection with art. 1b para. 1 BankG).

[41] The fintech license to accept deposits from the public differs qualitatively from the authorisation to accept crypto assets held in collective custody, which is why we can speak of two subcategories or two types of fintech licenses. This article focuses primarily on the new sub-category of fintech license for crypto-based assets held in collective custody.

[35] Federal Department of Finance FDF, Consultation draft concerning the amendment of the Banking Act and the Banking Ordinance (FinTech), Explanatory Report of 1 February 2017, P. 17 (cit. EFD-Fintech Explanatory report 2017).

[36] BBl 2020 (Fn. 12), P. 301.

[42] In the following, the common features of both types of fintech licences are presented first, which lie in the investment prohibition of the depository (4.2.1), in the allocation of risk bearing (4.2.2) and in interest prohibition (4.2.3). In a next step, the characteristics of the two types of fintech licences are examined (4.2.4), before concluding with a brief comment on the applicability of maximum amounts pursuant to art. 4^{sexies} BankG (4.2.5).

5.5.3 No Investment by Depository (Art. 1b para. 1 lit. b BankG)

[43] For the fintech licence to be applicable, the licensee must not "invest" the deposits or public assets (art. 1b para. 1 lit. b BankG). This requirement is intended to enforce the banks' renunciation of lending business. Public deposits or assets of the clients must therefore be available on a permanent basis[37] and may not be invested for proprietary transactions in the name and for the account of the fintech institution.[38] Risks for the client must be largely excluded and public deposits must be available in liquid form as well as crypto assets in the form in which they were accepted, so that they can be forwarded or refunded within a reasonable period of time in accordance with their intended purpose.[39] In addition, the clients' public deposits or crypto assets held in collective custody must be kept separate from the fintech institution's funds. Alternatively, these must at least be recorded in the fintech institution's books in such a way that they can be shown separately from its own funds at any time, but in this case an ordinary audit must be carried out in accordance with art. 727 CO due to the mere accounting separation[40] (art. 14f para. 1 BankV).

[44] The deposit of client funds as a demand deposit with a bank or other person shall not be deemed to be an inadmissible investment[41] pursuant to art. 1b BankG, provided they are held as high-quality liquid assets (HQLA) of category 1 in accordance with art. 15a of the Liquidity Ordinance (LiqV) (art. 14f para. 2 BankV e contrario). In addition, such an investment must be held in the currency in which

[37] EFD-Fintech Explanatory report 2017 (Fn. 35), P. 34.

[38] Federal Department of Finance FDF, Revision of the Banking Ordinance (BankV) "Fintech License", Explanatory Notes of 30 November 2018, P. 6 (cit. EFD-BankV Explanations 2018).

[39] EFD-BankV Explanations 2018 (Fn. 38), P. 17.

[40] EFD-BankV Explanations 2018 (Fn. 38), P. 17.

[41] EFD-BankV Explanations 2018 (Fn. 38), P. 17, explicitly mentions that the deposit must be made with "another" person in accordance with art. 1b BankG.

the client's claim for repayment is denominated (art. 14f para. 3 BankV).

[45] Cryptocurrency held in collective custody must be held (i) in Switzerland and (ii) in the form in which it was received (art. 14f para. 4 BankV). This does not affect the conversion of the form in consultation with the client.

[46] If a service provider makes a mistake when staking for his clients, some of the crypto assets deposited can be lost. To the extent that the service provider is liable for such errors vis-à-vis the client, the question arises whether the former, due to the risk incurred, must be considered an "investor". This is not the case, as slashing is an operational risk in connection with safekeeping and not an investment risk. Insofar as staking is carried out for the account of the client, this does not constitute a violation of the investment prohibition.

5.5.4 Who bears the risk of staking?

[47] The FDF's explanatory notes to the Banking Ordinance explicitly state that the investment prohibition (4.2.1) does not apply if staking is initiated in the name of the fintech institution but on the account of the depositor.[42] Although this is neither directly stated in the law nor in the ordinance, this position is supported insofar as the banks' lending business, which is prohibited under the fintech license, also constitutes a classic proprietary business of the bank, i.e. a business of the bank in its own name and for its own account.[43] Conversely, in the case of fiduciary transactions, i.e. transactions in one's own name but for the account of the client, the long-standing practice of the SFBC (FINMA's predecessor authority) must be taken into account in order not to fall under the banking licence requirement - probably also within the scope of the investment prohibition of the fintech license:[44] It must therefore be stated in legally binding form vis-à-vis the client that the investment is made for the client's account, i.e. that all risks and therefore also the del credere and transfer risk are borne by the client.[45]

However, the relevant lending business for banking purposes is affirmed if client funds are pooled, i.e. held in collective custody, in order to make investments that do

[42] EFD-BankV Explanations 2018 (Fn. 38), P. 6.

[43] BEAT KLEINER/RENATE SCHWOB/STEFAN KRAMER, in: Dieter Zobl et al. (Hrsg.), Kommentar zum Bundesgesetz über die Banken und Sparkassen dated 8 November 1934, Zurich 2011, art. 1 N 49.

[44] KLEINER/SCHWOB/KRAMER (Fn. 43), art. 1 N 63.

[45] EBK-Bulletin 17, P. 12 f.

not correspond in currency and maturity to the obligations entered into vis-à-vis the clients, the investors are promised a minimum return or currency losses are assumed, and neither the details nor the type of investments made are evident from the client statements, which means that there can no longer be any fiduciary investments.[46]

[48] Taking into account the materials on the DLT legislation and the old SFBC practice on fiduciary investments, investments or staking in one's own name but for the account of a third party would therefore probably have to be permissible under the fintech license and would not violate the investment prohibition, provided in particular that the type of crypto asset is not changed and the client bears all risks. In this respect, the question arises as to whether slashing also represents a risk that must be borne by the client. In our opinion, slashing is an error in the validation process that is either hardware-related, software-related or due to human error, i.e. an operational risk. If the corresponding nodes as well as the validation services are operated by the fintech institution itself, slashing would be an error within the fintech institution's sphere of influence, which would also constitute a breach of the fintech institution's duty to act diligently. This is not a classic risk due to third-party involvement, which the SFBC's practice requires to be transferred to the client, but an (operational) risk within the fintech company's sphere of influence, which should not even occur if the fintech company acts with caution. For these reasons, we believe it must be permissible for the fintech company to assume the slashing risk itself.

[49] In the case of a systematic promise of reimbursement to all staking clients in the event of slashing, the possible existence of an insurance transaction subject to authorisation under the Insurance Supervision Act (ISA) must also be examined. Of the five criteria, the existence of insurance, i.e. (i) the existence of a risk or hazard, (ii) the payment of the insured (premium), (iii) the nsurer's performance in the event of insurance/damage, (iv) the autonomy of the operation as well as (v) the compensation of the risks according to the laws of statistics (scheduled business operation),[47] at least criterion (iv) is unlikely to be met: The required independence of the organisation serves to distinguish insurance from other legal transactions in which the obligation to provide a service in the event of a claim is merely an ancillary agreement or modality of the other party to the contract, whereby in this respect it is not the formal arrangement but the inner connection between the promised services

[46] EBK-Bulletin 20, P. 16 et seq.; Judgement of the BG 2A.399/2004 and 2A.466/2004 dated 24 March 2005 E. 3.2.2.

[47] Cf. instead numerous BGE 114 Ib 224 E. 4.a

that is decisive.[48] In addition, as already mentioned, it could be argued that in those cases in which the wallet provider operates the node for staking himself, no slashing should occur due to the obligation to act diligently or that this would have to be taken over by the operator of the node if it should occur nonetheless, which ultimately means that there should never be a sector-typical transfer of risk to the client.

5.5.5 No Interest Paid by Depository (Art. 1b para. 1 lit. b BankG)

[50] For the fintech licence to apply, the custodian may not pay interest on the public deposits or assets (art. 1b para. 1 lit. b BankG), otherwise the banking licence applies. As in the case of the ban on investing, the interest prohibition is mainly intended to prevent the lending business of the bank and therefore the interest difference business.[49] Therefore, only interest payments directed by the fintech institution itself can be prohibited.

[51] The staking rewards accrued during staking are paid by the system to the address of the staker. As the staking rewards do not originate from the fintech company itself but from the respective distributed ledger system, there is no prohibited interest in the sense of the fintech license requirements.

[52] If the fintech institution wishes to secure a portion of the staking reward as its own compensation, it should do so not by directly establishing ownership of a specific portion of the staking reward itself, but by establishing a separate, merely contractual claim against the client, and should account for it separately so that the obligation to separate client and own assets is not breached.

5.5.6 Types of Fintech License?

[53] As explained above, the already existing Fintech license has been expanded so that there are now two subcategories of Fintech licenses: one for public deposits (4.2.4.2) and one for crypto assets held in collective custody (4.2.4.1).

[48] BGE 114 Ib 224 E. 4. c; 76 I 372.

[49] EFD-Fintech Explanatory report 2017 (Fn. 35), P. 34.

[54] Both licenses were created for different reasons and with different directions of impact: The fintech licence for public deposits in order to grant such fintech companies relief from the otherwise applicable banking licence, and the fintech licence for crypto assets held in collective custody in order to not allow such fintech companies to operate without a licence due to the level playing field principle for all institutions holding collective custody and the corresponding need for investor protection.

[55] In relation to the banking licence, it should be noted that, according to FINMA's established practice, no licences are granted on a voluntary basis if they are neither necessary nor used, which is why an affected institution probably cannot voluntarily upgrade to a banking licence if one of the two fintech licences is also sufficient or applicable. Conversely, it can be said that, for example, if an investment and/or interest-bearing activity is prohibited under fintech legislation, a banking licence is required unless an exception[50] is applicable.

Fintech License for Cryptocurrencies Held in Collective Custody

[56] The fintech license for cryptocurrencies held in collective custody applies to persons who are primarily active in the financial sector and who accept cryptocurrencies held in collective custody on a professional basis or who publicly recommend themselves as such (art. 1b para. 1 lit. a BankG in connection with art. 5a para. 1 BankV). Cryptocurrencies are crypto assets that actually or according to the intention of the organiser or issuer serve to a significant extent as a means of payment for the purchase of goods or services or the transfer of money or value and thus usually qualify as payment tokens according to FINMA ICO Guidelines.[51]

[57] Crypto assets held in collective custody are crypto-based assets that are allocated to a community, whereby it is evident which share of the community assets the custody client is entitled to, provided that the bank or fintech institution has undertaken to keep them available for the custody client at all times (art. 16 para. 1^{bis} lit. b BankG). It is precisely these collectively held crypto assets that are to be surrendered by the bankruptcy administration in the bankruptcy of the depository pursuant to art. 242a para. 2 lit. b SchKG and are therefore kept as depository

[50] In particular, exceptions from the public deposit pursuant to art. 5 (2) BankV, exceptions from the deposit pursuant to art. 5 (3) BankV, non-existence of professional activity (incl. sandbox exception) pursuant to art. 6 BankV.

[51] Swiss Financial Market Supervisory Authority FINMA, Guidelines for Submission Requests Regarding Initial Coin Offerings (ICOs) of 16 February 2018 (cit. FINMA-ICO-Guidelines).

assets outside the balance sheet of the fintech company. The separability of cryptocurrencies or payment tokens in bankruptcy would have led to the non-application of banking legislation in accordance with standard FINMA practice on banking legislation before the DLT legislation came into force, but the legislator wanted to expressly subject collectively held payment tokens to the fintech license requirement under the DLT legislation for investor protection reasons and for reasons of a level playing field in the context of the collective custody of various means of payment.

[58] On the other hand, according to art. 5a para. 1 BankV e contrario, the safekeeping of crypto assets, which are separable according to art. 242a para. 2 lit. a SchKG or art. 16 para. 1^{bis} lit. a BankG, is not subject to fintech licensing. This applies to crypto assets that are held directly on the blockchain in a wallet that is individual to the client, provided that the wallet provider has undertaken to keep them available at all times.

[59]In contrast to the fintech license for public deposits, the fintech license for cryptocurrencies held in collective custody does not have an upper limit of CHF 100 million for the permissible assets held in custody.

[60] However, the following assets are not considered crypto-based assets and are exempt from both the requirement for a Fintech license for crypto-based assets held in collective custody and the requirement for a Fintech license for retail deposits and the requirement for a banking licence:

a Assets held in client accounts as non-interest-bearing[52] credit balances solely for the settlement of client transactions (i) by precious metals dealers, asset managers or similar entities, provided settlement takes place within 60 days, or (ii) by investment firms or DLT trading systems (so-called settlement account exception);

b of domestic and foreign banks or other state-supervised companies,

c of institutional investors with professional vaulting.

[61]With art. 5a para. 2 BankV, the Federal Council has explicitly defined the existing exceptions to crypto assets. This in turn means that the exemptions from public deposits and deposits pursuant to art. 5 (2) and (3) BankV do not apply

[52] The no-interest requirement for the settlement account exception is actually unnecessary under the fintech license, as the fintech license already contains a general ban on interest.

to crypto-based assets. In particular, the following exceptions, which are frequently encountered in practice, can therefore not be claimed:

a Art. 5 para. 3 lit. b BankV concerning bonds with information required by banking law;

b Art. 5 para. 3 lit. e BankV concerning funds that are supplied in small amounts to a means of payment or payment system and are used solely for the future purchase of goods or services; and

c Art. 5 para. 3 lit. f BankV regarding default guarantees.

Fintech License for Public Deposits

[62] The Fintech licence for public deposits applies to persons who are primarily active in the financial sector and who accept public deposits of up to CHF 100 million on a professional basis or who publicly recommend themselves as such (art. 1b para. 1 lit. a BankG).

[63] The Federal Supreme Court defines the central element of a deposit as entering into obligations towards third parties. The obligor therefore becomes the repayment debtor of the corresponding performance.[53] According to art. 5 para. 1 BankV, liabilities to clients shall be deemed to be public deposits subject to the exceptions to the definition of public deposits pursuant to para. 2 and to the definition of deposits pursuant to para. 3. Public deposits are therefore always carried on the balance sheet of the fintech institution.

[64] Public deposits are based on the concept of money or means of payment,[54] whereby the FINMA ICO Guidelines also qualify liabilities in the form of tokens with the character of debt capital, e.g. repurchase promises with guaranteed returns, also qualify as deposits under the Banking Act;[55] this practice applies in particular to the custody of payment tokens.

[65] In contrast to the Fintech licence for cryptocurrencies held in collective custody, the general exceptions for public deposits and deposits pursuant to art. 5 para. 2

[53] BGE 132 II 382 E. 6.3.1; 136 II 43 E. 4.2; cf. Reiser on the term deposit (Fn. 32), P. 814; FLORIAN SCHÖNKNECHT, Der Einlagebegriff nach Bankengesetz, in: GesKR 3/2016, P. 300 et seq.

[54] Cf. REISER (Fn. 32), P. 811 et seq.

[55] FINMA-ICO-Guidelines (Fn. 51), P. 5 f.

and 3 BankV apply in full to the Fintech licence for public deposits. In return, the fintech licence for public deposits is limited to accepting public deposits up to a maximum of CHF 100 million.

[66] The fintech license for public deposits can only be used for the depository of crypto assets or for the staking of crypto assets in the institution's own name, provided that

a there is collective custody, but this does not meet the requirements of art. 16 para. 1^{bis} lit. b BankG with regard to separability (otherwise the fintech license for cryptocurrencies held in collective custody would be applicable); or

b there is individual custody, which does not meet the requirements of art. 16 para. 1^{bis} lit. a BankG with regard to separability.

5.5.7 Applicability of art. 4^{sexies} BankG

[67] For cryptocurrencies held by the bank as assets in custody for custody clients, FINMA may set a maximum amount on a case-by-case basis in accordance with art. 4^{sexies} BankG if this appears advisable in light of the risks associated with the transaction. In particular, it considers the function of crypto assets, their underlying technologies and risk mitigating factors.

[68] It is important to note here that due to its clear wording, this provision only applies to banks, but not to fintech companies.

5.6 Result

[69] Staking is equivalent to the provision of collateral, whereby this is not transferred to a third party but merely blocked for the duration of the staking period. Accordingly, staking services can be offered without affecting the separation rights of the holders of the staked crypto assets.

[70] In the case of individual custody of client assets, the staking service provider, similar to the custody of effects and other valuables, does not require any further license for this activity in addition to an affiliation with a self-regulatory organisation under the Anti-Money Laundering Act. Staking of crypto assets held in

collective custody that do not qualify as cryptocurrencies under art. 5a BankV, such as investment tokens, is also possible.[56]

[71] The provision of staking services for cryptocurrencies held in collective custody, on the other hand, requires a fintech or banking licence. Instead of allocation on the block chain directly, it follows via a database or other method of accounting at the service provider. This not only saves operational costs, but also opens up the possibility of staking for those clients who wish to provide less than the respective minimum amount of the system.

[72] If the service provider accepts crypto assets in non-separable manner, this constitutes a deposit and staking of the same is an inadmissible investment under the Fintech license for public deposits. Accordingly, a banking licence is required for the staking of crypto assets originating from client deposits for the account of the service provider or in the name of the service provider and for the account of the client without compliance with the fiduciary investment requirements, unless the service provider has a bank guarantee (art. 5 para. 3 lit. f BankV) or the deposit falls under another exemption.

[56] In particular, the financial market regulatory requirements in relation to securities must be examined, which is not the subject of this article.

Curriculum Vitae

Person

Name	Luzius David Meisser
Date of Birth	1979-12-27
Email	luzius.meisser@gmail.com
Civil Status	Married, four children
Citizenship	Klosters GR and Davos GR, Switzerland

Education

2016 - 2024	University of Zurich, Track A Doctoral Program in Finance
2013 - 2016	University of Zurich, Master of Arts in Economics
2000 - 2006	ETH Zurich, Master of Science in Computer Science
1993 - 2000	EMS Schiers, Type C Matura (science)

Work

2020 - now	Aktionariat AG, tokenization services, Chairman
2022 - now	Bitcoin Suisse Group, crypto broker, Chairman
2018 - 2021	Bitcoin Suisse Group, crypto broker, Board Member
2018 - 2021	ServiceHunter AG, administrative services, Chairman
2018 - 2021	Wyden AG (formerly AlgoTrader AG), Board Member
2016 - now	Meisser Economics AG, research and consulting, Chairman
2013 - 2014	FHNW Brugg, university of applied science, part-time lecturer
2012 - 2014	ServiceHunter AG, administrative services, Board Member
2009 - 2013	LaCie SA, Wuala, secure cloud storage, CTO
2007 - 2009	Caleido AG, Wuala, secure cloud storage, founder and CTO
2004	IBM UK, Hursley Development Lab, intern

Other

2021 - now	Autonomiesuisse, Board Member
2018 - now	Swiss Blockchain Federation, member of the Expert Council
2013 - now	Bitcoin Association Switzerland, founder and Board Member
2012 - 2013	Zeeder, venture funding, Partner
2005 - 2006	Unitech Alumni Association, Local Chapter Coordinator