Teller Protocol:

An algorithmic credit risk protocol for decentralized lending

Version 1.0

July 15, 2020

Authors

Ryan Berkun and Ivan Perez

Abstract

Teller is an algorithmic credit risk protocol, built to enable the creation of decentralized lending markets that can offer unsecured loans. The protocol's unique cloud-based infrastructure can connect to, and privately compute credit and banking data to generate individual loan terms based on a users' creditworthiness. Teller Protocol was designed to develop decentralized loan products, without collateralized debt, reducing consumer risk and costs. Teller can interoperate with centralized finance data, offering everyone the freedom to develop a new wealth of trustless financial instruments.

This white paper is a reader-friendly description of the protocol, which is built on the Ethereum blockchain.

Glossary of terms

Teller Protocol

An algorithmic credit risk protocol, built to enable the creation of decentralized lending markets that interoperate with centralized financial data on the Ethereum blockchain.

Teller Labs

A distributed team of stakeholders focused on protocol development, with plans to support the network towards progressive decentralization.

Distributed Cloud

A network of distributed cloud nodes that interoperate with the Ethereum blockchain and act as protocol validators.

Autonomous Teller Markets (ATMs)

Ethereum smart contracts[1] that govern the protocol's liquidity pools. Users interact with ATMs for the deposit and/or borrowing of assets within the protocol's liquidity pools.

[1] Buterin, Vitalik. "Ethereum Whitepaper: A Next Generation Smart Contract & Decentralized Application Platform" (2013).

Credit Risk Algorithms (CRAs)

Open-source algorithms that interact with user credit and banking to assess default risk and generate loan terms based on creditworthiness.

Index

1 Introduction

Since its inception, blockchain technology, the technology underpinning the Bitcoin network[2], has given rise to opportunities to create unbiased, transparent, and efficient financial products. By using infrastructure that distributes data and value across a network of computers, individuals can subvert flawed, central-entity-controlled business models and create self-custodial solutions that operate without the shortcomings of centralized financial systems.

The Decentralized Finance (DeFi) industry expanded this premise with the promise of an open, trustless alternative to existing financial instruments like peer to peer loans, high yield savings accounts, and derivatives contracts, among other alternative investment products.

Projects like Compound[3] were able to automate cryptocurrency loans negotiated trustlessly on the internet without a bank. Users can now earn interest for their crypto assets through a lending platform built on Compound's protocol. Similarly, The Maker Protocol[4] created a U.S. Dollar-pegged stablecoin[5] (DAI) backed by collateral. Users can interact with the protocol to mint and borrow DAI against their crypto via Maker's Oasis application (app).

However, early success does not come without limitations. Current DeFi protocols rely on high collateralization ratios to mitigate risk for the lending and borrowing of crypto assets. The average DeFi user is subject to collateralization ratios starting at 150% to upwards of 300%. At present, a borrower might need to deposit anywhere from 1.5 to over three (3) units of collateral to mint one (1) stablecoin.

Over-collateralized[6] loan products increase the borrower's risk exposure to the underlying collateralized asset's volatility. Additionally, overcollateralized loans present a barrier to entry for individuals who do not currently own crypto, or have limited financial means, further narrowing the gap between DeFi and the addressable global loans market.

---

[2] Nakamoto, Satoshi. "Bitcoin: A Peer-to-Peer Electronic Cash System" (October 2008). www.bitcoin.org

[3] Leshner, Robert; Hayes, Geoffrey. "Compound: The Money Market Protocol" (Feb 2019). https://compound.finance/documents/Compound.Whitepaper.pdf

[4] The Maker Protocol: Makerdao's Multi-collateral Dai (mcd) System https://makerdao.com/en/whitepaper/#the-maker-protocol

[5] Stablecoins are cryptocurrencies designed to minimize the volatility of the price of the stablecoin, relative to some "stable" asset or basket of assets.

[6] Loans requiring collateral that exceeds a 1-to-1 ratio against the collateralized asset

Crossing The Chasm: From DeFi to CeFi

DeFi protocols have been designed to be inherently composable and serve as the building blocks for a permissionless global finance system that caters to the public good. To achieve this, DeFi developers require a financial primitive that can interact with centralized financial data to evaluate credit risk, and lower or eliminate the need for collateralized debt.

Teller Protocol will provide an interoperability solution that can leverage existing legacy credit scoring systems. Moreover, the protocol's ability to algorithmically calculate a user's creditworthiness will enable the creation of decentralized lending markets that can offer unsecured loans. Eliminating the need for collateral while simultaneously lowering users' financial exposure will accelerate the adoption of DeFi apps and facilitate the development of a new class of cryptocurrency loan products.

2 Protocol Overview & Features

Teller Protocol

Teller is an algorithmic credit risk protocol that enables the creation of decentralized lending markets for unsecured loans on the Ethereum[7] blockchain. The protocol operates via a unique, cloud-based infrastructure that can connect to and privately compute credit and banking data to assess a user's creditworthiness.

Network Architecture

Teller Protocol operates via a network of distributed cloud nodes that interoperate with the Ethereum blockchain. The protocol's nodes act as validators, each of which run on cloud-based infrastructures, such as Amazon Web Services (AWS) and Google Cloud Platform (GCP). Teller's cloud-based infrastructure allows it to connect with other, external, or centralized financial data providers. In other words, Teller's network or 'Distributed Cloud' acts as a data router for the protocol's smart contracts.

---

[7] Buterin, Vitalik. "Ethereum Whitepaper: A Next Generation Smart Contract & Decentralized Application Platform". www.blockchainlab.com, November 2013.

All data accessed through the cloud is digested by the protocol's validators and analyzed by open-sourced credit-risk algorithms. Computation of the protocol's Credit Risk Algorithms (CRAs) is the primary purpose of each validator.

For more information on the conceptualization of a distributed cloud network and its architecture, please review the Stratosphere Whitepaper[8].

How does Teller manage data privacy?

The protocol retrieves user data through Teller's whitelisted, privacy-preserving data providers[9]. User data sources include, but are not limited to, banking transactions, credit reports, and income verification. Users who interact with Teller's data providers are in primary control of their data and can opt-in to share specific data through API-based, third party data transfer networks that interoperate with the protocol.

Once a user's data is retrieved, the data provider cryptographically signs and returns it to the user. The data provider then stores each users' account access tokens alongside its respective Ethereum wallet. This process maintains a one-to-one mapping of the Ethereum wallet to the account and enables future retrieval of current user information.

Teller's data providers follow GDPR compliance on data privacy by only maintaining the index of the user's data concerning the specific data provider.

In future iterations of the protocol, Teller validators will utilize both secure enclaves and zk-Snark[10] technology to enhance data privacy and protection further.

Credit-Risk Algorithms

Teller's Credit Risk Algorithms (CRAs) serve as the foundation for managing the protocol's liquidity pools, also known as Autonomous Teller Markets (ATMs). CRAs interact with users' credit and banking data via Teller's distributed cloud network to assess default risk and generate loan terms based on creditworthiness.

The resulting loan terms are submitted to a consensus contract on the Ethereum blockchain. If consensus is reached, i.e., higher than ⅔ of validators agree on the proposed loan terms, the protocol automatically initiates the loan.

---

[8] The Stratosphere Whitepaper describes a distributed cloud network where cloud operations are performed by selected nodes, in which each node runs a replication of a given computer function.
[9] Centralized data providers contracted by Teller DAO to supply credit information to the protocol.
[10] What Are Zk-snarks?
https://z.cash/technology/zksnarks/

Note, at its inception, the Teller Protocol will feature a single global CRA called Arrowhead. Future iterations of the protocol will introduce new CRAs, and offer open-source developers the ability to propose CRAs that can be particular to an ATM, or a specific type of loan product.

Arrowhead CRA (V1)

The Arrowhead CRA will prioritize a particular set of risk parameters that will serve as the protocol's global risk parameters. The diagram attached in the following page (Figure A) illustrates Arrowhead's logic flow:

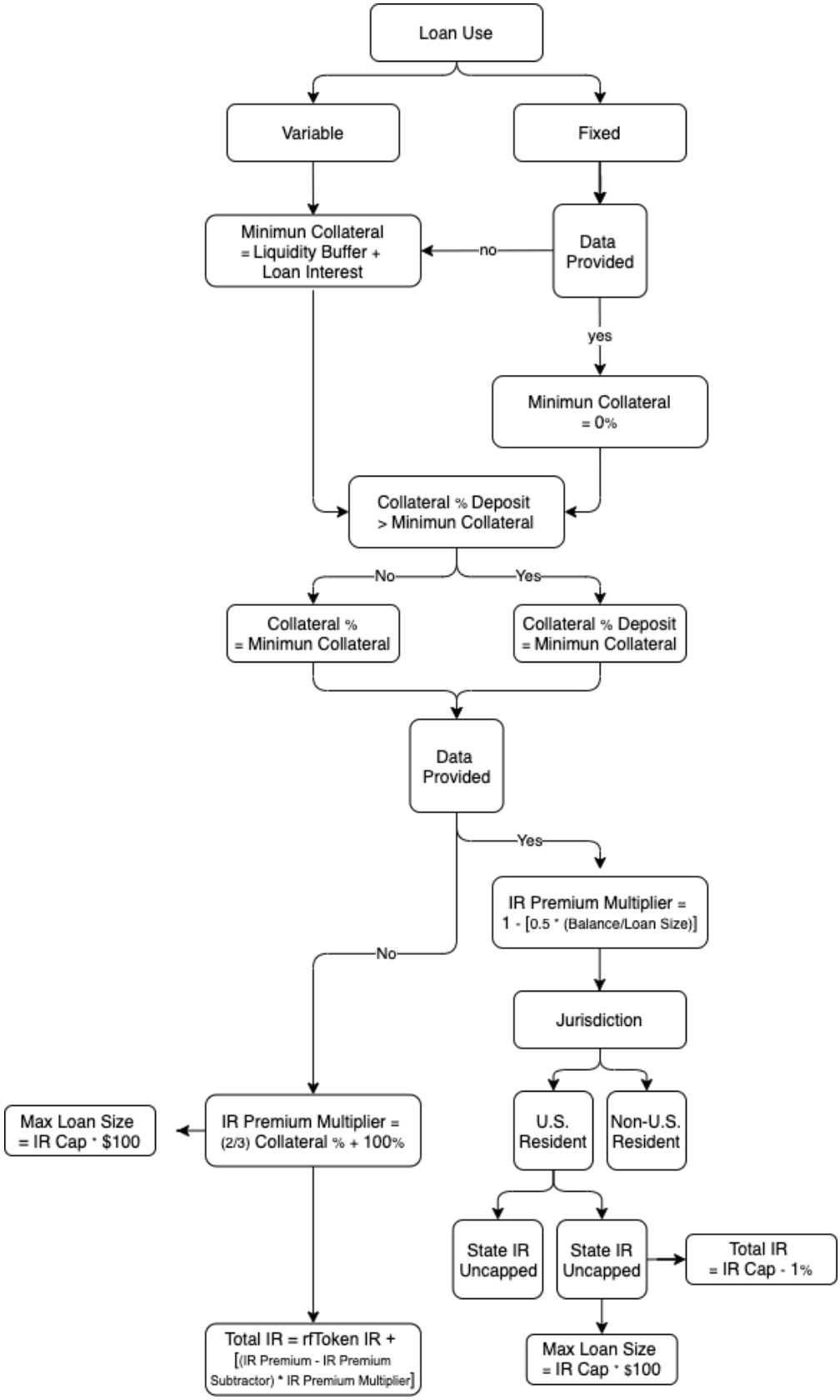
Equation

$$Interest\ Rate\ per\ Asset = rfToken\ IR\ + Risk\ Premium\ IR$$

Key Variables

- Risk Premium IR = Global, asset-specific additional interest rate to account for risk associated with fully unsecured loans
- rfToken IR = Asset specific risk-free interest rate accounting for said assets
- Collateral = Logarithmic correlation to default rates
- State Jurisdiction = Capped IR for states with a limit
- Bank Balance = Proves ability to repay a loan
- Income = Proves cash flow
- Net Income = Proves cash flow available to repay a loan
- Recent Cash Flow = Proves recency of cash flow (absence is a NEG)
- Loan Use = Explains different risk assessments per the type of loan
- Variable = Loans that will be converted into an asset different from the asset they were issued in
- Fixed = Loans that will not be converted into an asset different from the asset they were issued in

Figure A:


Loan Use
Variable
Fixed
Minimun Collateral = Liquidity Buffer + Loan Interest
no
Data Provided
yes
Minimun Collateral = 0%
Collateral % Deposit > Minimun Collateral
No
Yes
Collateral % = Minimun Collateral
Collateral % Deposit = Minimun Collateral
Data Provided
Yes
IR Premium Multiplier = 1 - [0.5 * (Balance/Loan Size)]
No
Jurisdiction
Max Loan Size = IR Cap * $100
IR Premium Multiplier = (2/3) Collateral % + 100%
U.S. Resident
Non-U.S. Resident
State IR Uncapped
State IR Uncapped
Total IR = IR Cap - 1%
Total IR = rfToken IR + [(IR Premium - IR Premium Subtractor) * IR Premium Multiplier]
Max Loan Size = IR Cap * $100

3 Autonomous Teller Markets (ATMs)

All lending capital deposited to the protocol is held in the protocol's liquidity pools through Ethereum smart contracts[11] called Autonomous Teller Markets (ATMs). Users can access ATMs within the protocol through a variety of network access points. These access points include the UserTeller Application[12] and any other interfaces built by network contributors.

Importantly, ***ATMs are non-custodial.*** Users interact with Teller Protocol through ATMs; however, each user has direct control of their funds. Borrowers can withdraw their collateral at any time, provided their loan is repaid. Likewise, lenders can withdraw their earned interest at any time, and their total deposited funds, provided there are enough unlent funds in the pool, or all loans have been repaid.

How are ATMs Created?

Anyone can create an ATM by interacting with the protocol. An ATM requires four components: supplied assets, collateral assets, data adaptors, and CRAs. In essence, all ATMs will require the deployment of a new smart contract that will house the liquidity pools for the particular assets that will be lent or borrowed through said ATM. For a more detailed explanation of the technical implications of creating an ATM, please refer to Introduction to the Teller Protocol in the Teller Documentation Portal.

Depositing into an ATM

Individuals that deposit lending capital into an ATM are known as Liquidity Providers (LPs). LPs are responsible for sourcing the protocol's liquidity and are rewarded overtime via a percentage of loan interest payments made by borrowers. The amount of loan interest, or Annual Percentage Yield (APY) an LP earns varies per loan and asset type.

Additionally, LPs can insure their ATM deposits through a network of smart contract coverage[13] vendors. In other words, the smart contract itself is the lender, while LPs are simply providing liquidity to said lender or smart contract.

---

[11] Buterin, Vitalik. "Ethereum Whitepaper: A Next Generation Smart Contract & Decentralized Application Platform" (2013).
www.blockchainlab.com

[12] A decentralized application built by the Teller Lab's team to function as a front end marketplace of ATMs that have been whitelisted by the protocol's stakeholders.

[13] A novel insurance product that covers "unintended code usage" where someone has suffered a financial loss on a smart contract.

Annual Percentage Yields

LPs receive APY for every loan that originates from an ATM. Teller's ATMs can host multiple liquidity pools, each specific to deposited assets into a particular ATM. LPs will only earn interest from loans issued by a liquidity pool into which they have deposited funds. Interest yields are automatically deposited to an LP's wallet every time a borrower repays a loan.

Calculation and fluctuation. APYs are calculated per asset liquidity pools, in an ATM, as the net return of all loan repayments and defaults during the specified time horizon, divided by the global asset liquidity pool supply. The equation below demonstrates logic for APY calculations:

$$A = (1) - [r - (D + L)] \div [(lLiq) \times (lLiq \div sLiq)] + [(rfToken\ Value \div uLiq) \times (uLiq \div sLiq)] \quad \textbf{(1)}$$

Equation **(1)** represents the relationship between the latent variables in calculating APY where: A equals APY and r represents *loans repaid* minus *loans defaulted* (D) or *liquidated* (L). In a similar fashion, Liq represents *total liquidity* in an asset pool and is categorized as *lent* (l), *supplied* (s) and *unlent* (u). The $rfToken\ Value$ is equal to an asset specific risk-free interest rate accounting for said asset.

Varying APYs. APYs vary depending on the interest rates set for each loan. Higher risk loans, i.e. loans that vary in size, or borrower risk parameters, will have higher APYs.

Earning Compound Interest

Assets deposited into Teller ATMs as lending capital will be automatically converted into interest-accumulating tokens called tTokens via an algorithmic interest rate protocol.

LPs' t-tokens essentially act as a 'proof of liquidity.' tTokens are a deposit statement or receipt confirming that the LP has provided liquidity for a Teller ATM. All assets within a Teller ATM have an underlying supply interest rate (APR). This accrual of interest is directly attributed to the LP wallet holding the proportional tTokens. When an LP chooses to liquidate his position or withdraw interest yields, tTokens are converted into the ATM's underlying asset.

Borrowing from an ATM

Users can interact with Teller Protocol to request a loan from an existing ATM. All users have the option to share their credit and banking data. As with traditional loan products, user data is used to determine a borrower's creditworthiness, subsequently affecting their loan interest rate and amount of collateral needed to request a loan.

How does Teller use credit history?

The data shared with the protocol is stored in an encrypted format, with nodes processing the data able to decrypt it for analysis. Once a borrower has received loan terms from the Teller's distributed cloud, they can submit the loan terms to an ATM for approval and loan issuance. Borrowers that do not provide sufficient data, or do not meet the protocol's global risk parameters will have to provide collateral.

How and when is collateral needed?

Collateral needed is determined by the protocol's CRAs. This amount varies per loan based on specific data parameters. If a loan is not approved to be wholly unsecured, the borrower must provide collateral before redeeming a given loan.

Borrowers will need to repay their loans before withdrawing their collateral. If the collateral's value falls below the minimum total value of collateral needed, the protocol will automatically trigger a collateral liquidation to repay the loan.

Generating Loan Terms (Risk Analysis)

An individual's creditworthiness can be a combination of their existing credit score, banking data, and global consensus among the protocol's CRAs. Creditworthiness will be determined by the amount and type of data a user provides.

Loan Interest Rates

Loan interest rates (APRs) are generated through CRA computations. Teller Protocol sets a global interest rate, or risk premium, for users who do not supply personal data. For consumers that do supply personal data, eg., bank transactions, the data is computed through the protocol's CRAs. Results from such CRA computations include both the loan terms (such as interest rate, maximum loan size, and collateral needed) specific to the individual consumer, and a cryptographic signature from each Teller validator for proof of computation.

Defaulting on a Loan

Defaulting on a loan can negatively impact a borrower's real-world credit history, and trigger a default auction that results in collateral provided by the borrower to be liquidated and used to repay the outstanding debt.

Default Auction

When a default occurs, the borrower's collateral is auctioned for the underlying ATM's lent asset. The collateral asset's price and the resulting amount of the lent asset needed to repay the loan are determined by the live oracle price for the specific asset pair.

Debt Collection

The Teller Protocol works with existing debt collectors such that when a borrower defaults on a loan,they are subject to a potential collections mark on their credit report. Additionally, borrowers that default on a loan will be subject to being contacted by a third party collections agency that will attempt to recover the outstanding debt.

Users must only consent to be contacted by third party debt collection agencies regarding defaults for loans with collateral liquidations falling below 100%. If the asset amount recovered is below the loan principal plus interest.

4 Key Protocol Actors

Liquidity providers. Liquidity providers (LPs) are individuals that provide capital for Teller Protocol's liquidity pools. In other words, LPs are responsible for supplying and depositing the lending capital available via Teller's ATMs.

Auctioneers. Auctioneers are network participants that help liquidate collateral for defaulted loans to ensure LPs' capital. They sell or buy collateral when an asset's market price falls below the protocol's liquidation ratio.

Price oracles. The Teller Protocol requires real-time information about the market price of the collateral assets in its ATMs to know when to trigger liquidations. The protocol derives its internal collateral prices from a decentralized oracle network consisting of a broad set of individual nodes called oracle feeds.

Data providers. Data providers are vendors contracted by Teller Labs to supply credit and banking information to the protocol. This information is computed by the protocol's credit-risk algorithms that are computed and maintained by the protocol's validators.

Validators. Validators are in charge of running computations for the Teller Protocol. Validators act as cloud nodes, receiving user data that is then computed by Teller's CRAs. They do not

control loan terms. Validators' sole purpose is to ensure the health of the protocol's CRAs, which generate borrower's loan terms.

5 Protocol Governance

At its inception, protocol governance will be delegated to internal team members and the protocol's stakeholders. Future iterations of the Teller Protocol will introduce a decentralized governance module through which community members will be able to submit proposals for review and approval by the protocol's stakeholders. Additionally, the protocol will utilize this governance module to transparently vote to whitelist new ATMs on the UseTeller App.

6 Protocol Risk Parameters

Max loan size. Controls the USD equivalent ceiling of a specific asset and loan type.

Max interest rate. Outlines the interest rate ceiling of a specific loan type. For example, personal loans may have different maximum interest rates than leveraged loans.

Collateral-to-debt ratio. Establishes the minimum cumulative ATM collateral needed relative to the outstanding debt. If a newly requested loan would lower the collateral-to-debt ratio, the requested loan will not be processed.

Supply-to-debt ratio. Governance ratio which limits the total amount of liquidity to be supplied, relative to the outstanding debt. If a newly requested loan would cause the supply-to-debt ratio to drop below the approved amount, the request will not be processed.

Liquidation spread. Measures the difference between the needed collateral percent and the liquidation percent.

Liquidation reward. Establishes the bounty percent earned by the liquidator for settling the collateral swap on a defaulted loan.

Liquidation penalty. Establishes the fee withdrawn from the loan collateral, to be returned to the ATM as additional recovered debt.

7 Future of Teller Protocol

Interoperable, Risk Assessed Lending Markets

Teller aims to enable the creation of highly liquid, decentralized, risk assessed lending markets. Ultimately, the protocol can serve as the foundation of a borderless, trustless hybridization of current banking systems into a wealth of new decentralized financial instruments.

An Autonomous Organization

Progressively decentralized. Decentralization is the cornerstone of a blockchain's immutability and permissionless attributes. Slow or measured permissionless development within the Teller network will translate into a combination of academic and scientific measures towards proper governance within a self-sustained ecosystem.

Open source; self-sustaining. Teller Protocol is an iterative amalgamation of many other open-source projects. We plan to continually engage with developers, academics, auditors, and financial experts. We will pursue ongoing testing, security audits, and cross-industry collaborations that help us develop a robust and secure financial primitive that can lead to the adoption of decentralized finance applications.

References

Nakamoto, Satoshi. “Bitcoin: A Peer-to-Peer Electronic Cash System” (2008).
www.bitcoin.org

Hsieh, Ying-Ying, "The Rise of Decentralized Autonomous Organizations: Coordination and Growth within Cryptocurrencies" (2018). *Electronic Thesis and Dissertation Repository*. 5393.
https://ir.lib.uwo.ca/etd/5393

The Year in Ethereum 2019 - Josh Stark - Medium
https://medium.com/@jjmstark/the-year-in-ethereum-2019-242012e4276d

The Maker Protocol: Makerdao's Multi-collateral Dai (mcd) System
https://makerdao.com/en/whitepaper/#the-maker-protocol

Buterin, Vitalik. “Ethereum Whitepaper: A Next Generation Smart Contract & Decentralized Application Platform” (2013).
www.blockchainlab.com

Shiftehfar, Reza. “Uber’s Big Data Platform: 100+ Petabytes with Minute Latency” (Oct 2018).
https://eng.uber.com/uber-big-data-platform/

Ellis, Steve; Juels, Ari; and Nazarov, Sergey. “Chainlink. A Decentralized Oracle Network” (Sep 2017).
https://link.smartcontract.com/whitepaper

Buchman, Ethan. “Tendermint: Byzantine Fault Tolerance in the Age of Blockchains” (2016).
https://allquantor.at/blockchainbib/pdf/buchman2016tendermint.pdf

Szabo, Nick. “Smart contracts: building block for digital markets” (1996). *Phonetic Sciences Amsterdam.*
http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

Leshner, Robert; Hayes, Geoffrey. “Compound: The Money Market Protocol” (Feb 2019).
https://compound.finance/documents/Compound.Whitepaper.pdf