Boson Protocol White Paper
Decentralized Autonomous Commerce

Authors: Justin Banon, Gregor Boroša
Whitepaper Version 1.1: draft, 01 November 2020

Abstract

We present Boson Protocol, decentralized infrastructure for enabling autonomous commercial exchanges on Ethereum. Boson is a peer-to-peer system which replicates the benefits of a market intermediary, without the disbenefits of centralized systems- which abuse trust by extracting excess profits and hoarding data, and arbitrated decentralized systems- which add cost and friction. The protocol is a permissionless, generic mechanism for enabling the decentralized exchange of digital value for a non-fungible token voucher representing any product, service or *thing*. This, without centralized intermediaries and with minimized arbitration, trust and cost. Boson implements a 2-sided deposit structure within a dynamic game, which automates the mediation of disputes and mitigates reversal losses, by ensuring that both agents have skin in the game. Commerce data is pooled and equitably monetized within a secure, privacy-preserving, shared data layer with ownership retained by the individual. The system is community-owned, public infrastructure, which is resistant to capture. As such Boson represents a breakthrough in the scalable, automated coordination of commercial exchange with a vision:-
"To be the world's open, public infrastructure layer for commercial transactions and their data".

Document structure

This white paper is structured as follows. The first section is an overview of the protocol, and can be read as a standalone 'Light Paper'. The Overview has three sub-sections. A Background which describes the current situation and the problem which Boson Protocol addresses. Next, Vision & Objectives which specifies the solution requirements. Then, Boson Protocol Overview, which provides a high-level description of the protocol. Following the Overview section are separate sections detailing the Core Exchange Mechanism, Token Model, Governance and Technology. These subsequent sections are not required reading for a general understanding of the protocol.

Index

Overview

Background

Online commerce coordinates the exchange of *monetary value* as digital payments, for *non-monetary value* as goods or services (henceforth *things*). Today, online commerce typically remains intermediated by two types of trusted third parties:

- **Financial intermediaries** for processing payments
- **Market intermediaries** for facilitating exchange (e.g. Amazon, eBay, Priority Pass)

Financial intermediaries

Bitcoin was first to offer *"an electronic payment system based on cryptographic proof instead of trust"* (Nakamoto, 2008) that enabled disintermediated payments. Satoshi's white paper addresses two problems caused by the cost of reversible transactions:

1. **Dispute mediation.** Reversible transactions require dispute mediation- which adds transaction costs.
2. **Reversal costs.** Sellers incur costs whenever payment transactions are reversed for non-reversible products and services.

Satoshi presents Bitcoin as a non-reversible payment solution which protects sellers from transaction reversal, and suggests that *"routine escrow mechanisms could easily be implemented to protect buyers"* (Nakamoto, 2008). So, whilst Bitcoin renders the monetary side of the transaction irreversible, the non-monetary, market side retains the cost of dispute mediation and reversal.

Market intermediaries

The management of dispute mediation and reversal is a primary function of market intermediaries. For centralized market intermediaries, dispute mediation and reversal costs are lost within the typically excess profits extracted by such platforms. For decentralized market intermediaries (such as Openbazaar and Origin), dispute mediation is typically performed by arbitrators and represents a visible and material additional cost. The impact of dispute mediation costs on otherwise free services, limits the *"minimum practical transaction size and cuts off the possibility for small casual transactions"* (Nakamoto, 2008). Further, the additional cost and friction, renders arbitrated protocols impractical for many use cases, particularly for machine-to-machine and decentralized applications.

Problem 1

The coordination of commercial transactions requires either centralized intermediaries or decentralized arbitrators to manage dispute mediation and transaction reversal. This adds cost and trust, which limits the scope and reduces the efficiency of commerce.

Data silos

Data is a highly valuable component of online commerce. Market intermediaries capture and take ownership of a wide range of data including buyers' personal and product preference data, together with seller pricing, ratings and reviews. The value of this data derives from its ability to predict consumer buying behaviour, inform product development and provide market insight. Despite its utility, the vast majority of data is locked-away in the proprietary data silos of tech titans such as Amazon and eBay, or sold privately on the *shadow data economy*[1].

Extraction and capture

As they scale, centralized market intermediaries amass competitive advantage via their data troves, network effects and economies of scale. Such networks invariably move from cooperating with their participants to competing, and from attracting customers to extracting[2]. This is neither a coincidence nor a choice, since profit-making entities have a fiduciary responsibility to maximise shareholder value. An *extraction imperative*[3], if you will. Amazon is replete with examples. First, the EU alleged that Amazon was using competitively sensitive data gathered in its role as a marketplace -regarding marketplace sellers, their products and transactions- to unfairly advantage itself as a seller[4]. Second, Amazon has used this data to launch competing, Amazon-branded, products. Third, Amazon has vertically integrated into areas such as freight[5] so aggressively that it could soon become the world's largest freight company[6]. As a result, centralized market intermediaries have the means and motive to capture and dominate multiple markets.

Problem 2

Data collected from commercial transactions by centralized intermediaries is locked-away and used to strengthen anti-competitive market dominance which imperils the interests of the consumer, other firms and even governments.

---

[1] "The Web3 Data Economy - Ocean Protocol." https://blog.oceanprotocol.com/the-web3-data-economy-b6fd8ecac4c4. Accessed 24 May. 2020.

[2] "Why Decentralization Matters - OneZero." 18 Feb. 2018, https://onezero.medium.com/why-decentralization-matters-5e3f79f7638e. Accessed 24 May. 2020.

[3] "The Future Of Network Effects: Tokenization and the End of ...." 17 Jul. 2018, https://medium.com/public-market/the-future-of-network-effects-tokenization-and-the-end-of-extraction-a0f895639ffb. Accessed 24 May. 2020.

[4] "Antitrust: EC opens formal investigation against ... - europa.eu." 17 Jul. 2019, https://europa.eu/rapid/press-release_IP-19-4291_en.htm. Accessed 24 May. 2020.

[5] "Amazon loses contract with FedEx Express as ... - The Verge." 7 Jun. 2019, https://www.theverge.com/2019/6/7/18656813/amazon-prime-fedex-express-delivery-logistics-network-contract-termination-usps-ups. Accessed 24 May. 2020.

[6] "Amazon Could Soon Be The World's Biggest Shipping Company." 24 Sep. 2019, https://www.forbes.com/sites/stephenmcbride1/2019/09/24/amazon-could-soon-be-the-worlds-biggest-shipping-company/. Accessed 24 May. 2020.

Challenge, Vision & Objectives

The Challenge

> *Is it possible to design a peer-to-peer system which replicates the benefits of a market intermediary, without the disbenefits of current centralized and decentralized systems? Such a system would coordinate economic exchange whilst minimizing trust and cost. Commerce data would be pooled within a secure, privacy-preserving, shared data layer and would be owned by the individual, and monetized in an equitable way. The system would be community-owned, public infrastructure, which would be resistant to capture and would be capable of out-innovating and out-competing entrenched incumbents.*

This is the challenge which we take-up and which we believe Boson Protocol has the potential to meet.

Vision

Boson Protocol's vision is:
"To be the world's open, public infrastructure layer for commercial transactions and their data "

Objectives

To achieve our vision we define a number of objectives and challenges as follows:

Decentralized coordination of economic exchange:

To design a decentralized protocol which coordinates the exchange of monetary for non-monetary value whilst minimizing arbitration friction and costs, so as to be widely applicable and practical.

This requires the following properties:

- **Arbitration minimized** - the core system is automated and requires human intervention from arbitrators as an exception only.
- **Trust minimized** - all parties can reach consensus on the truth without requiring a trusted third party, with arbitration as an exception only.
- **Practical atomicit**y - payment and receipt of goods happens together or not at all, so Buyers can trust that either they receive the goods or their money back, and Sellers can trust that they will be paid for goods supplied.
- **Incentive compatible** - the mechanism enforces system rules and ensures that there is no advantage to be gained by breaking the rules.
- **Practical and commercially acceptable** - the rules are simple enough to understand and use, and commercially acceptable to all parties.

Incentivization

To optimize the Boson Protocol Objective function:

maximize the supply of high quality voucher redemptions.

This requires that the system incentivizes:

- **Supply-side acquisition** and **demand-side distribution** of inventory.
- **Early adopters**, in order to overcome the chicken and egg effect.
- **Curation** and **redemption** of **quality** inventory.
- **Data sharing** and **monetization**.

Data

To develop a planetary-level Web3 data marketplace for commerce.

This requires the following properties:

- **Share, pool and monetize data** - in a secure, privacy-preserving and self-sovereign way.
- Incentivize voluntary **data sharing** via an **equitable distribution** of the value it creates.

Governance

To implement a governance model across the three phases of: start-up, scale-up and decentralize which will progressively enable the following properties:

- **fair and equitable distribution of ownership, value and control.**
- **capture resistance** - from centralized, extractive entities.
- **regulatory compliance**- with legitimate authorities.
- **community ownership and operation**.

Overall Protocol

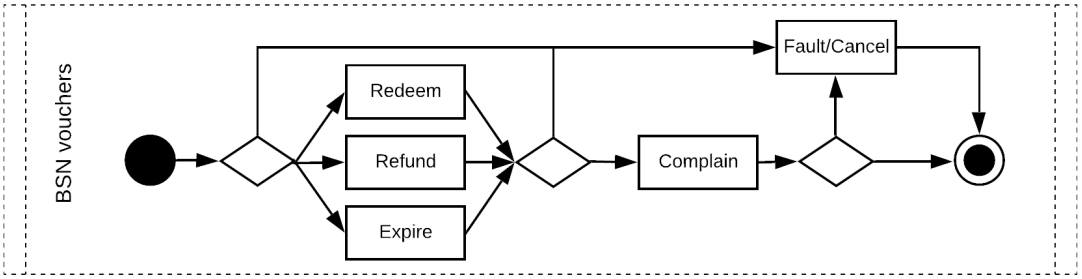
We aim for protocol-market fit, sustainable value capture and equitable value distribution.

Boson Protocol Overview

The vision for Boson: "To be the world's open, public infrastructure layer for commercial transactions and their data", is enabled by a design with five modular and substitutable components. First, a commitment to perform a future commercial exchange represented as a tokenized voucher. Second, a core mechanism for autonomous coordination of commercial exchange. Third, a token model for incentivising actors, and capturing and distributing value. Fourth, a Web3 data marketplace for monetizing data. And finally, an evolving governance system for directing and controlling the protocol throughout its lifecycle.

NFT Vouchers

Rather than tokenizing *things* directly, Boson instead represents *a promise to exchange digital value for a thing at a future date*, as a non-fungible token voucher (NFTV). Thus Boson NFTVs can be conceptualized as a type of futures contract for a thing. Boson contracts are implemented as stateful non-fungible token vouchers (NFTVs), whose states change as they flow through Boson's core exchange mechanism.



Events in a voucher's life cycle.

Core exchange mechanism

Boson's core exchange mechanism is sufficiently complex to manage the exchange, dispute mediations and reversals; but simple enough to be governed by a game. Boson NFVTs escrow three monetary amounts upon both parties' commitment to transact. First a payment amount is taken from the Buyer and is released to the Seller if, and only if, the Buyer cryptographically signs a redemption transaction. This ensures practical atomicity, by which we mean the transaction is as atomic as a cash transaction. Second, the Buyer transfers to escrow a deposit, which is held as a commitment for the Buyer to proceed with the transaction, and is forfeited should the Buyer reverse the transaction through no fault of the Seller. Third, the Seller transfers to escrow- a deposit, which is held as commitment for the Seller to redeem the Boson NFVT for the thing, to an acceptable quality. These 2-sided deposits represent commitments within a dynamic game which minimizes arbitration by automating dispute mediation and reversal via an incentive compatible and commercially acceptable set of rules. The game's algorithm evolves over time towards increasing automation and decreasing arbitration.

It should be noted that the Boson core exchange mechanism constitutes a new primitive which can be used as a generic building block for enabling decentralized exchange with minimized trust and cost, across a wide range of contexts.

Token Model

Boson's token model is another modular building block, whose efficacy is judged by its ability to incentive actors towards optimisation of the system objectives. Boson's current architecture leverages Ocean Protocol's design for a Curated Proofs Market (CPM). Here, actors stake behind vouchers which they *predict* will optimize for the objective function: *maximize the supply of high quality voucher redemptions.* Actors are then rewarded with Boson tokens as a function of the amount staked, the timing of the stake (with increased rewards for early stakes) and the *actual* transaction value and quality.

Boson tokens are the native utility token for the protocol and are used to incentivize actions across the system. Firstly on the supply side, Bosons reward supply acquisition via *Aggregators and Sellers*, and supply quality via *Curators*. Secondly on the demand side, *Relayer* marketplaces earn fees In Bosons to incentivize distribution of inventory. Thirdly for data sharing, Buyers are incentivized to permit their data to be monetized in return for an equitable share of the value it creates. In addition, a token allocation is reserved to enable the system to incentivize early adopters and contributors. This enables the system to overcome the bootstrapping challenge and also funds development.

[N.B. At time of writing an alternative candidate for the Boson token model has emerged and is being assessed.]

Web 3 data marketplace

Web3 Data model

As a protocol for coordinating commercial exchange, Boson Protocol will be in a position to pool a valuable graph of consumer preference data. Whereas, Web 2 tech titans capture users' data and separate users from the value their data creates; Boson Protocol incentivizes voluntary data sharing by providing users with an equitable distribution of the proceeds from data monetization. Boson Protocol's strategic vision for data is to develop a planetary-level Web3 data marketplace for commerce.

Boson's design leverages an Ocean Protocol data marketplace to enable the pooling of data in a secure, privacy-preserving and self-sovereign way. Instead of locking-away valuable data within proprietary data silos, Boson's data marketplace makes data openly available for purchase. Data buyers may purchase personal, product preference, pricing and ratings data to predict consumer buying behaviour, inform product development or develop market insight. Further details of Boson's data marketplace will be released in a subsequent version of this white paper.

Governance

Boson will follow the pragmatic path of progressive decentralization[7] across three phases. First a centrally controlled *lean start-up* in order to achieve protocol-market fit. Then, once the protocol has developed defensibility via network effects, a minimally extractive fee will be levied on protocol services. This will be used to incentive a community of early adopters and contributors to scale-up the project. Finally Boson will fully decentralize to a DAO or similar, which will be structured to ensure regulatory compliance, whilst ensuring fair and equitable distribution of ownership, value and control.

Overall protocol

Boson is a protocol which enables decentralized autonomous commercial exchange, in a highly generic and unopinionated manner. The protocol functions as a commercial primitive or 'lego brick', thereby enabling digital and decentralized apps to be easily developed and integrated. This composability supports Boson's universal application and broad adoption in pursuit of protocol-market fit.

Whilst the protocol itself is minimally extractive, it possesses significant value capture potential. This, both as a standard for exchanging non-monetary value across the internet, and also as a planetary-level web 3 data marketplace for commerce. Boson redistributes value equitably to founders, investors, contributors, its community and users. With particular emphasis on early contributors.

Boson can be conceptualized in a variety of ways:

1. As a set of smart contracts, components and standardized interfaces- think Amazon's APIs as a decentralized protocol.
2. As SMTP for transferring non-monetary value- think Bitcoin for transferring non-monetary value.
3. As a universal means to commit, store and transfer promises.
4. As a decentralized commercial oracle.
5. As "thing tokens flowing around a thing economy" (Trent McConaghy).

[7] "Progressive Decentralization: A Playbook for Building Crypto ...." 9 Jan. 2020, https://a16z.com/2020/01/09/progressive-decentralization-crypto-product-management/. Accessed 2 Jun. 2020.

Use cases

Boson Protocol's universality spawns a wide range of use cases, a subset is described below:

Online commerce

- An open digital marketplace where any *thing* can be offered, searched and exchanged with minimized trust and cost.

Voucher distribution of non-monetary value

- The COVID 19 pandemic has increased the requirement for the distribution of food or essentials to those in need. Boson NFT vouchers enable this distribution in a highly automated, auditable and low-cost way

Machine-to-machine commerce

- Enabling autonomous cars to purchase tyres or servicing with autonomous management of disputes and redemption.

Loyalty and rewards

- Enabling loyalty programs and credit card rewards to offer any digital or physical thing in a standardized, composable and interoperable digitized format, without the cost and friction of intermediaries.

Games

- Enabling video games to gift or grant permission to buy rare or special items. For example, on reaching Grand Wizard status a player has the right to buy a special t-shirt.

Gaming

- Enabling blockchain gaming applications such as on-chain bingo to pay-out prizes in Boson NFVTs, redeemable for off-chain products.

Crypto exchanges

- Enabling exchange tokens to be redeemed for real world rewards in order to differentiate on rewards rather than compete on fees, whilst increasing token value.
- Enabling exchange users to purchase real world items directly from an exchange marketplace, without touching fiat.

Service bookings

- Enable bookings for any service from restaurants to collection of groceries to be secured via two-sided deposits, to ensure that Buyer and Seller meet their commitments to redemption and terms.

Tokenized networks

- Enabling users to exchange their network tokens for digital and physical goods and services, in order to increase user perceived value and token value.

NFT Vouchers

A digital voucher is a familiar mechanism used to redeem stored value for products and services. Stored, often monetary, value is exchanged for a redeemable promise in the form of a digital voucher, which can be redeemed at a later date for physical or digital goods and services. We borrow the following generic definition of a voucher:

> *A 'voucher' represents a redeemable promise or "a digital representation of the right to claim services or goods" - Fujimura*

Boson vouchers tokenize this promise of a trade as a non-fungible token voucher (NFTV). Where a *promise* represents one agent's commitment to *exchange digital value for a thing at a future date*. Boson tokenizes a commitment to execute a future exchange, rather than the thing itself or an on-chain ownership transfer. Thus, Boson NFTVs can be viewed as a type of futures contract for the exchange of any digital or physical thing using a universal, interoperable and programmable format.

Essentially, an promise is a tuple consisting of the issuer, the underlying asset that is the subject of the trade (i.e. a good or service), the value of that asset (typically monetary), and the conditions on the implementation of the promise (whatever they may be, as long as they are computable). A NFTV is then a tokenized promise which is redeemable by the bearer, subject to any conditions programmed within the voucher smart contract.

Boson NFT vouchers are:

- **universal** - can represent any physical or digital product, service, experience or promise
- **interoperable** - provide a common standard for representing any *thing,* be it digital or physical, and a common interface for conducting commerce.
- **composable** - can be assembled into composite products or used to modularize products.
- **programmable** - can be programmed to enforce any computable rules.
- **transferable and storable** - can be easily stored and transferred between actors using standard wallet infrastructure.
- **stateful** - Boson NFTVs change state as they flow through Boson's core exchange mechanism.

Core exchange mechanism

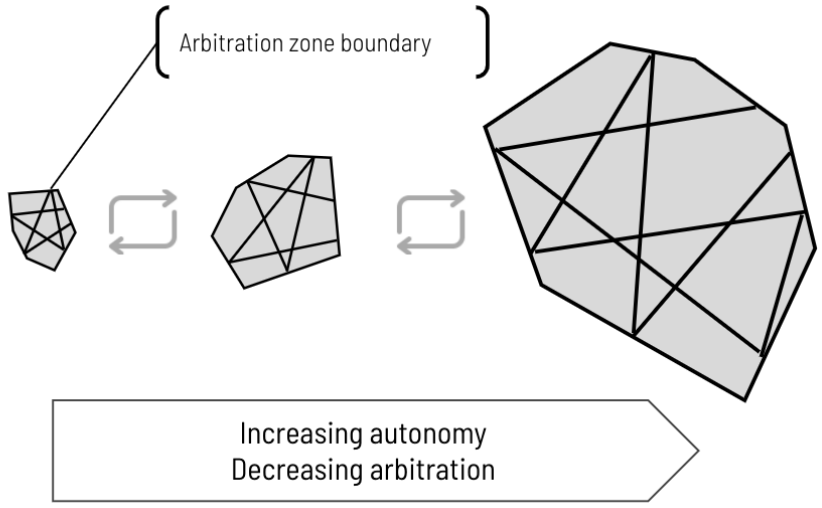
Boson's core exchange mechanism can be viewed as a decision support system which coordinates exchange, governs reversals and handles the main load of disputes in order to significantly reduce arbitration cost and friction versus arbitrated protocols. The mechanism is analysed as a dynamic game which relies on two-sided deposits and a lightweight reputation system.

System evolution

The system and the role of arbitration evolves over time as follows:

1. Scaffolding

Following the principle of evolving complex systems with guard rails, the system starts with arbitration to ensure the system behaves as intended; however, arbitration doesn't scale. The system will bounce around within a specified zone, when the system hits a boundary, arbitration will be used to push it back. Algorithmic triage will then be added recursively to automate this intervention in future. In principle this could be a temporary, or at least decaying function, resulting in a progressive increase in autonomy and decrease in arbitration.



System evolution

2. Watcher

The presence of an observer with the capability to change outcomes, changes the game theory- whether or not they act. So, in theory, the system could evolve towards having a watcher. That is, an arbitrator who, in the limit, does not act.

3. Escalated arbitration

Given 1 (scaffolding) and 2 (a watcher), there may be a subset of attack vectors which cannot be triaged away or resolved without action by an arbitrator. The relative size of this subset, whether 10x or 100x , is an important determinant of the utility of the mechanism. However, utility has diminishing returns above a threshold level, where the mechanism is practically 'good enough' and further improvement does not merit further investment.

System design

The ideal arbitrated system

Boson Protocol implements a 2-sided deposit structure within a dynamic game, which automates the mediation of disputes and mitigates reversal losses, by ensuring that both agents have skin in the game. However, first we conceive of an ideal system which is highly automated, has minimum viable functionality for coordinating commercial exchange, and in which disputes and reversals are mediated by an ideal arbitrator.

The objective function of the system is:

to maximize the supply of high quality voucher redemptions.

The core exchange process proceeds as follows:

- **Offer** - Seller makes an offer of a voucher for a thing, and sets required Buyer Payment (P_{Bu}), Seller deposit (D_{Se}) and Buyer deposit (D_{Bu}) amounts.
- **Commit** - Buyer accepts offer by signing a commit transaction. Buyer payment, Seller deposit and Buyer deposit are escrowed.

Deposit amounts are variable and form part of the commercial terms of the exchange, Deposit levels are proposed by the Seller and then accepted by the Buyer. It is possible that the same underlying *thing* could be offered via multiple vouchers, each with differing Buyer and Seller deposit levels, indicating different levels of commitment to redemption and quality. We elaborate on this below.

Practical atomicity

For the happy path, at point of exchange the Buyer unilaterally signs a redemption transaction in return for the Seller transferring the thing. If, and only, if the redemption transaction is signed, the system will transfer the Buyer Payment amount to the Seller. We refer to this as *practical atomicity*, where *practical* refers to the assumption that a Buyer who signs the redemption receives the thing. The transaction is *practical* rather than absolute, because it is as atomic as handing over cash for goods. With a cash transaction the Seller could take the cash and not deliver the service, but this is not a practical concern. Therefore, we contend that this is atomic enough for most commercial purposes.

Transaction reversibility

It follows- firstly, from the fact that not all promises will be implemented; and secondly, from the requirement for atomicity- that transactions must be reversible. For example, if a Buyer escrows payment for a product, and the product subsequently sells-out, then unless there is a means to reverse the escrow, the atomicity property will break. As a consequence, the system enables a Buyer to trigger a Refund transaction, and automatically reverses transactions at Expiry of their validity period. This confers the following three options on the Commit state:

- **Redeem** - the Buyer cryptographically signs a redemption transaction as proof that they have received the thing. Payment is transferred to the Seller.
- **Refund** - the Buyer signs a refund transaction to trigger a refund. Payment is returned to the Buyer.
- **Wait** - No action taken, the validity period of the NFT voucher expires. Payment is returned to the Buyer.

However, transaction reversibility introduces the challenges of reversal costs and dispute mediation identified by Satoshi. The example of buying a used car online illustrates these challenges.

Reversal costs

Alice views a used car on a website, offered for sale at a cost of $10,000 and with 100,000 miles on the clock. Alice makes an offer to Bob of $10,000, which Bob accepts. Alice, who lives in London, agrees to travel the following weekend to complete the exchange with Bob, who lives 200 miles away in Leeds. Now, if Alice fails to arrive to complete the exchange, Bob has missed out on other opportunities for exchange and incurred a loss. Conversely, if Alice arrives in Leeds and Bob has already sold the car, then Alice incurs a loss.

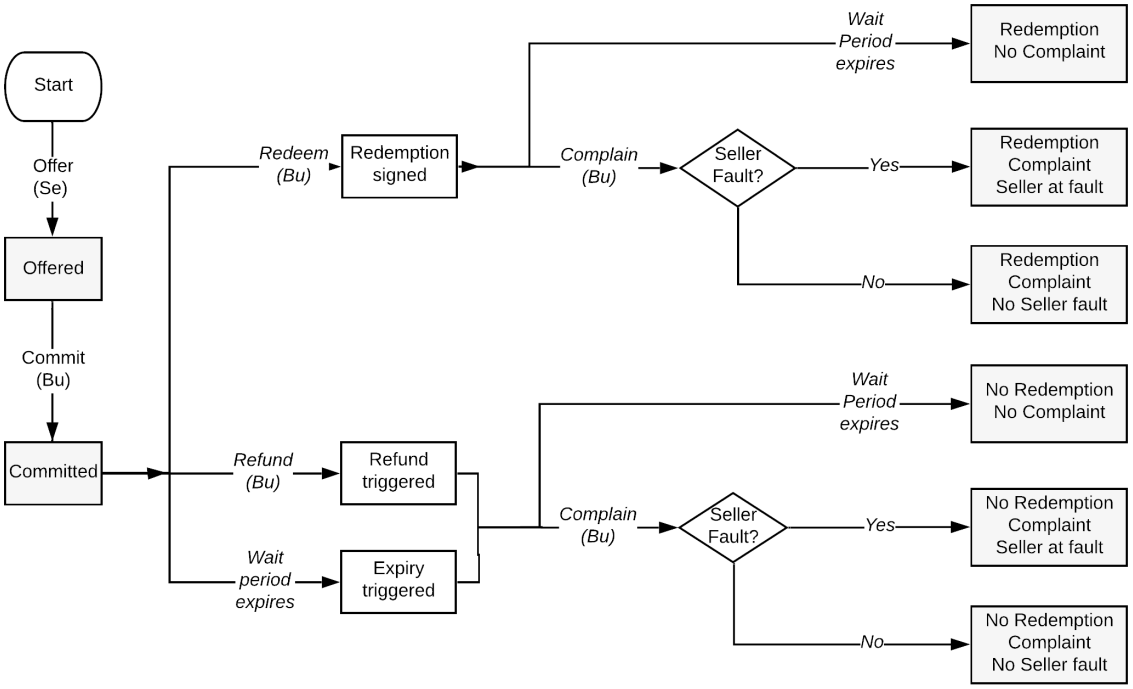
Dispute mediation

If Buyer and Seller beneficiaries make conflicting claims on real world events, then smart contracts cannot determine which version of reality to trust. An example being, if Alice arrives in Leeds to complete the exchange and detects that the car has actually driven 200,000 miles, Alice has a quality dispute. With our ideal system, this is not a problem.

The Buyer has the option to unilaterally report quality issues as follows:

- **Wait**. No action taken, complaint period expires.
- **Complain**. Buyer signs complaint transaction.

In the event of a complaint, an ideal arbitrator observes real-world events and adjudicates perfectly. This results in an ideal logic tree, in which the system can make optimal decisions via arbitrators (see diagram below). To continue the previous example, the arbitrator would be able to observe the car's odometer and either uphold or dismiss the complaint.



Ideal arbitrated system - logic tree

The ideal system with an ideal arbitrator, can determine whether the Seller is at fault. The Buyer and Seller payoff tables below describe how the system applies penalties and rewards by slashing or transferring deposits. This is shown for each combination of complaint, redemption and Seller fault.

Ideal Payoff Table — **Target Seller**

		Redemption			Non-redemption		
No complaint		Seller	S1		Seller	S5	
Complaint	Seller at fault	Seller	S3	- High	Seller	S7	- High
Complaint	No Seller Fault	Seller	S4		Seller	S8	

Ideal Payoff Table — **Target Buyer**

		Redemption			Non-redemption		
No complaint		Buyer	B1		Buyer	B5	- High
Complaint	Seller at fault	Buyer	B3	+High	Buyer	B7	+High
Complaint	No Seller Fault	Buyer	B4		Buyer	B8	- High

Legend

Low penalty	- Low
Medium penalty	- Med
High penalty	- High
Medium reward	+Med
High reward	+High

Ideal arbitrated system - payoff tables

Incentive compatibility analysis - ideal arbitrated system

No Seller Fault

Incentive Compatible	Index	Analysis
Yes	B5, B8	Buyer penalized for non-redemption when no Seller fault
Yes	S4, S8	Seller not penalized when not at fault

Seller at Fault

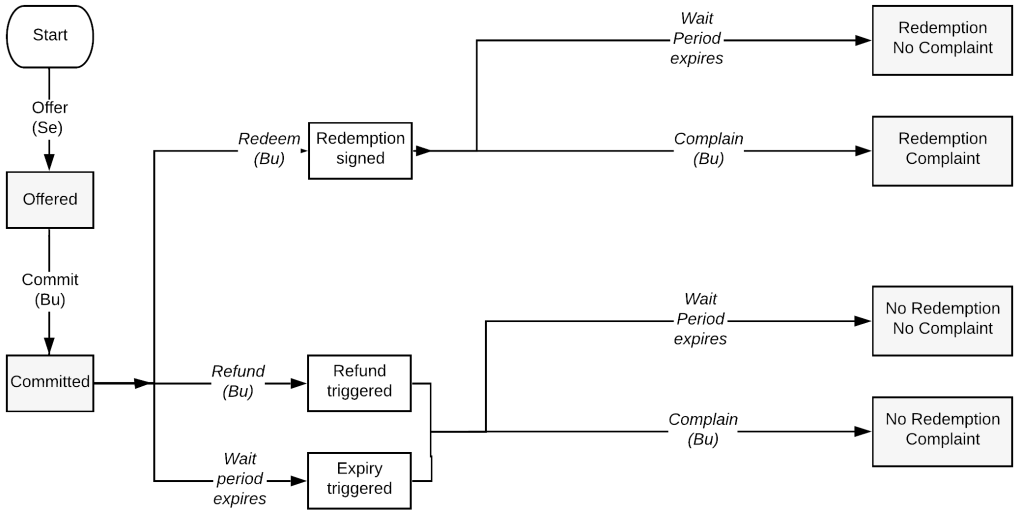
Incentive Compatible	Index	Analysis
Yes	S3, S7	Seller penalized when at fault and Buyer complains
Yes	B7	Buyer not penalized for non-redemption when Seller at fault
Yes	B3, B7	Buyer receives compensation for fault

The analysis above shows that for all outcomes, the ideal system achieves incentive compatibility, given a sufficient level of Buyer and/or Seller deposit (elaborated below).

The ideal system represents a target for incentive compatibility, against which we measure successive iterations of Boson Protocol with the aim to achieve an acceptable level of incentive compatibility, without human arbitration.

Boson core exchange mechanism iteration 1

The first iteration of Boson's core mechanism implements the ideal system schema, except with the naive assumption that Buyer complaints are valid. This leads to the following representation of Boson's basic exchange mechanism.



Boson exchange mechanism (iteration 1) - logic tree

The outcomes of this basic mechanism are described by two orthogonal dimensions:

- Redemption (by Buyer)
- Complaint (by Buyer)

and are governed by the following rules:

1. **Payment made if, and only if, a Redemption transaction is signed.**
    - Implements practical atomicity between payment and promise implementation.
2. **Seller deposit (DSe) is slashed* when Buyer complains.**
    - Incentivises Seller quality.
    - Indirectly incentivises Sellers to redeem, in order to avoid Buyer complaints for refused redemptions.
3. **Buyer deposit (DBu) is slashed for non-redemption.**
    - This incentivises Buyers to redeem, by penalising Buyers who trigger refunds or allow vouchers to expire.

*By slashed we mean transferred to an address controlled by Boson admins or eventually DAO.

The logic tree for Iteration 1 results in 4 end-states whose payoffs are as follows:

Boson Iteration 1 (naive) Payoff Table — **Target Seller**

	Redemption	**Non-redemption**
No complaint	Seller S1	Seller S5
Complaint	Seller S4 - High	Seller S8 - High

Boson Iteration 1 (naive) Payoff Table — **Target Buyer**

	Redemption	**Non-redemption**
No complaint	Buyer B1	Buyer B5 - High
Complaint	Buyer B4	Buyer B8 - High

Boson exchange mechanism (iteration 1) - payoff table

The analysis below shows that iteration 1 achieves incentive compatibility for a subset of outcomes only.

Incentive compatibility analysis - iteration 1

No Seller Fault

Incentive Compatible	Index	Analysis
Yes	B5	Buyer penalized for non-redemption when no Seller fault
No	S4, S8	Seller penalised when not at fault (Malicious Buyer)

Seller at Fault

Incentive Compatible	Index	Analysis
Yes	S4, S8	Seller penalized when at fault (quality or cancel)
No	B8	Buyer penalized for non-redemption when Seller at fault, and cannot redeem (double whammy)
No	B4, B8	Buyer receives no compensation for when Seller at fault (uncompensated Buyer)

Analysis of the above Boson core mechanism iteration 1 payoffs reveals three primary challenges / attack vectors. We label these as *Buyer Double-whammy, Malicious Buyer* and *Uncompensated Buyer*, and describe their mitigations below.

Buyer double-whammy

In this case, a quality redemption is not available to the Buyer, due to the fault of the Seller, and yet the Buyer is also penalised. For example, when a Buyer redemption is refused by a Seller, the Buyer will also lose their deposit.

In this case our aim is to incentivise the Seller to admit a fault if, and only if, they believe they are at fault. To incentive this we leverage the inequity aversion and preference for fairness observed with the *Ultimatum Game*[8], and described as a *resentment mechanic* by Vitalik in his *Scorched earth 2-of-2 escrow*[9]. Here it is demonstrated that actors will punish others for perceived unfairness even if this results in economic loss for themselves. Therefore, we expect the Seller to be incentivized to honestly admit a fault if this would cause a portion of their deposit to be transferred to the Buyer, rather than be confiscated by Boson. Conversely, we expect that a Seller would not want to incentivize nor enable a Buyer to benefit from malicious complaints, even if that resulted in an economic penalty for the Seller.

[8] "Ultimatum game - McGill CS." Ultimatum game.
[9] "List of primitives useful for using cryptoeconomics-driven ...." List of primitives useful for using cryptoeconomics-driven internet / social media applications.

Boson core exchange mechanism iteration 3

Within Boson core mechanism iteration 3, we operationalize the resentment mechanic by introducing an additional transaction type:

- **Cancel or Fault (CoF)** - a Seller signs a CoF transaction to cancel or admit a fault.

CoF simplifications

The CoF transaction conflates two subtly different operations as a system simplification, designed to reduce the number of end-states by collapsing separate Seller Cancel and Fault states into a single CoF state. Another simplification is that the system does not distinguish between the order of a CoF and a Complaint, in order to eliminate multiple permutations from end-states. This is despite the fact that it might be preferable for a Seller to proactively admit a fault, rather than wait for a customer to complain. Future iterations of the mechanism might update these simplifications with more precise, and yet more complex, algorithmic elements.

Iteration 3 transaction rules

CoF transactions are governed by the following rules:

Triggering CoFs

1. The CoF transaction can be unilaterally triggered by the Seller at any point after Commit and before final expiry.

Cancel versus fault

2. A CoF signifies a Seller's admission of fault. If a CoF happens before Redeem, Refund or Expiry, then a voucher is also cancelled.
3. Cancellation means a voucher cannot be subsequently redeemed, refunded or expired.

Payment (Bu)

4. Payment / Redemption atomicity is maintained.

Deposit **(Se)**

As a Seller admission of fault, the CoF transaction primarily operates on the Seller deposit (D_{Se}).

5. **If Complain = N AND CoF = N, Then 100% D_{Se} to Se**
   If Buyer does not complain and Seller does not cancel or admit fault via a CoF; then 100% of the Seller deposit is returned to the Seller, since there is no question of a Seller fault.

6. **If Complain = N AND CoF = Y, Then 50% to Se, 50% to Bu**
   If Buyer does not complain, but Seller triggers a CoF; then 50% of the Seller deposit is returned to the Seller as incentive for admitting a fault, the remaining 50% of Seller deposit is transferred to the Buyer as compensation for the fault. Whilst this does financially incentivise Seller CoFs, the Seller also has an incentive to transact and so is merely reducing the loss of a complaint and not benefiting per se. Consequently, if Seller can redeem to quality, then it is economically rational for them to do so. The portion of the Seller Deposit which is transferred to Buyer creates a Seller altruistic /

customer service incentive. In the case of the Buyer, although they are unable to redeem the voucher to quality, they avoid having their deposit slashed for non-redemption, and receive a proportion of the Seller deposit as compensation for the reversal.

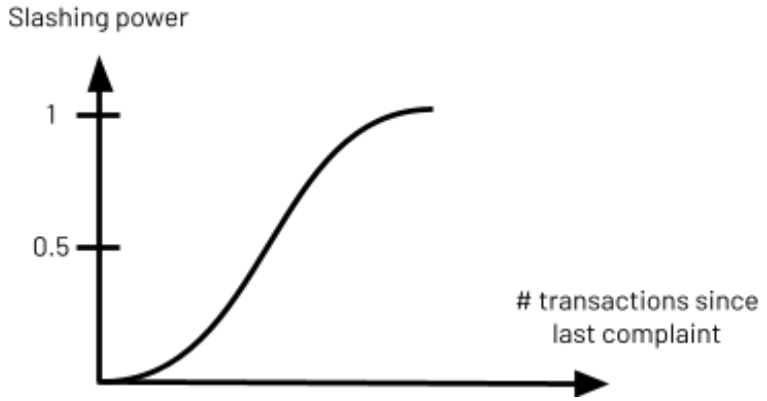
7. **If Complain = Y AND CoF = Y, Then 25% to Se, 50% to Bu, 25% to BOSON**
   If Buyer Complains and Seller triggers a CoF, then 25% of Seller deposit is returned to Seller, 50% to Buyer and 25% is slashed. This has three effects. First, Seller is incentivized for admitting a fault, but less so than if Buyer had not complained. This relates to either the Seller triggering a CoF after a complaint- so not being proactive, or a Buyer still complains after a CoF- an escalated complaint. Second, Buyer always gets the same compensation if a Seller triggers a CoF, so there is no additional incentive for Buyer to complain. Third, in order to keep this balance between Buyer and Seller incentives, the remainder of the Seller deposit must be slashed (transferred to Boson).

8. **If Complain = Y AND CoF = N, Then 100% D_{Se} to BOSON**
   If Buyer complains and Seller does not CoF, then Seller loses deposit for an un-admitted complaint. The resentment / Ultimatum mechanic incentivizes Sellers to CoF when at fault, however this case is still subject to the Malicious Buyer attack vector, which we address below.

Malicious Buyer

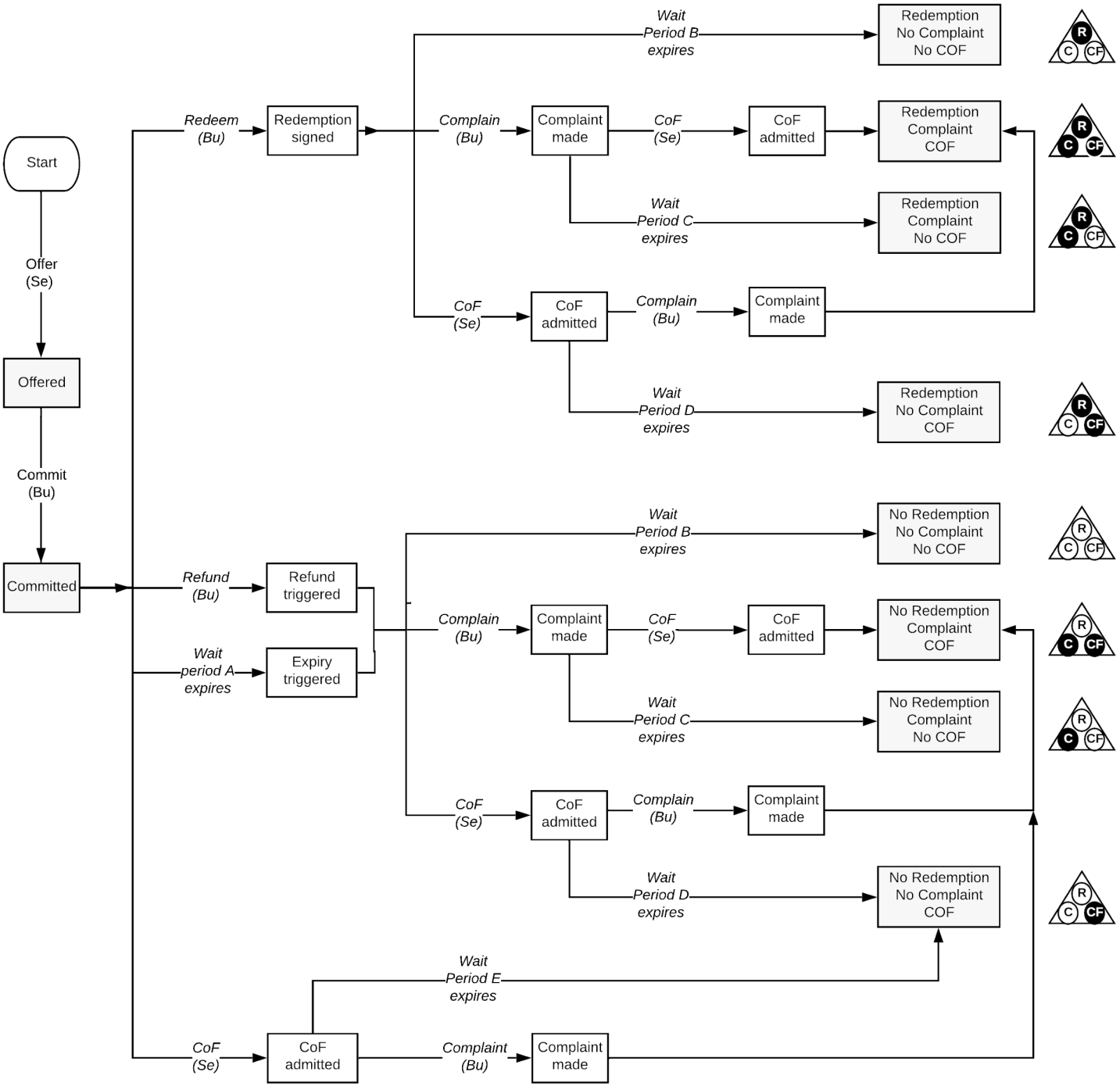
The 'Malicious Buyer' attack vector occurs when a Buyer makes an unjustified complaint. With Iteration 1 a Malicious Buyer could cause repeated economic loss to Sellers. In order to mitigate this attack we implement additional algorithmic triage in the form of a lightweight reputation system. The slashing power of a Buyer is made dependent on the history of a Buyer address' previous transactions, shaped as a basic sigmoid function:

- If a Buyer complaint is not admitted (no Seller CoF), then slashing power is reduced.
- If a transaction occurs where a Buyer either does not complain, or complains and Seller CoFs, then slashing power increases.



Variable Buyer slashing power

The resultant logic tree for Boson's core exchange mechanism iteration 3 is shown below.



Boson exchange mechanism (iteration 3) - logic tree

Iteration 3 results in 8 end-states. These triangular figures alongside the logic tree diagram depict these end-states and the symmetries between them. Iteration 3 payoffs are as follows:

Boson Iteration 3 (deposits only) **Target**
Payoff Table **Seller**

		Redemption			Non-redemption		
No complaint	**No CoF**	Seller	S1		Seller	S5	
No complaint	**CoF**	Seller	S2	- Low	Seller	S6	- Low
Complaint	**CoF**	Seller	S3	- Med	Seller	S7	- Med
Complaint	**No CoF**	Seller	S4	- High	Seller	S8	- High

Boson Iteration 3 (deposits only) **Target**
Payoff Table **Buyer**

		Redemption			Non-redemption		
No complaint	**No CoF**	Buyer	B1		Buyer	B5	- High
No complaint	**CoF**	Buyer	B2	+Med	Buyer	B6	+Med
Complaint	**CoF**	Buyer	B3	+Med	Buyer	B7	+Med
Complaint	**No CoF**	Buyer	B4		Buyer	B8	- High

Boson exchange mechanism (iteration 3) - payoff table

Under the following assumptions, the analysis below shows that iteration 3 achieves overall incentive compatibility by mitigating the main attack vectors.

Assumptions

Seller admits only if at fault

- economic incentive to admit if at fault, but resentment not to mis-admit

Malicious Buyer mitigated

- via lightweight reputation system on slashing power

Buyer not directly incentivized to complain

- as only receive compensation if Seller admits
- and lose slashing power for when they need it

Incentive compatibility analysis - iteration 3

No Seller Fault

Incentive Compatible	Index	Analysis
Yes	B5, B8	Buyer penalized for non-redemption when no Seller fault admitted
Mitigated	S4, S8	Seller penalised when not at fault (Malicious Buyer) ● mitigated via lightweight reputation system on slashing power
Yes	S2,3,6,7	Seller will not mis-admit fault, due to resentment mechanic

Seller at Fault

Incentive Compatible	Index	Analysis
Yes	S1, S5	Seller not penalized, as no fault reported via Complaint nor CoF
Yes	S4, S8	Seller penalized (high) as unadmitted complaint (incentive to admit)
Yes	S3, S7	Seller penalized (medium) as admits fault, with Buyer complaint
Yes	S2, S6	Seller penalized (low) as admits fault, with no Buyer complaint
Mitigated	B8	Buyer penalized for non-redemption when Seller at fault, and cannot redeem (double whammy) ● mitigated via Seller incentive to admit when at fault
Mitigated	B4, B8	Buyer receives no compensation for fault ● mitigated via Seller incentive to admit when at fault
Yes	B2,3,6,7	Buyer deposit returned and compensated for fault

Whilst Boson core mechanism iteration 3 is broadly incentive compatible when the above assumptions and mitigations are taken into account, there remain some challenges.

Firstly, whilst the above payoff table shows the deposit incentive gradient *pushing* the Seller upwards towards No Complaint, No CoF; it does not show a similar gradient pushing the Seller leftwards towards redemption.

Secondly, Buyer payoffs do not show a gradient pushing Buyer towards No Complaint (if the Seller is not at fault).

These two challenges are due to our analysis looking at deposits only. We have also modelled non-deposit incentives including:

- **Buyer fault loss** - Buyer incurs a loss when Complaint or CoF occurs with non-redemption.

- **Seller profit -** Seller makes a profit on exchange.
- **Seller fault loss -** Seller loss due to non-redemption.
- **Seller resentment -** Seller disincentive to mis-admit.
- **Buyer dishonest complaint disincentive -** altruism, and retain slashing power for when needed.

Whilst details of the full analysis is outside of the scope of this white paper, we include below payoff tables which model these additional incentives. The results show that when the additional incentives are considered, the system pushes both Buyers and Sellers towards Redemption, No-complaint. This is the case whether or not the Seller is actually at fault.

An academic paper with a more formal analysis of the incentives and game theory, within Boson's exchange mechanism, is being written by one of our advisors.

Incentive compatibility analysis - iteration 3 with full payoffs

Boson Iteration 3 (full payoffs)
Payoff Table

Actual
Seller
No Seller Fault

		Redemption			**Non-redemption**		
No complaint	**No CoF**	Seller	S1	15.0	Seller	S5	-25.0
No complaint	**CoF**	Seller	S2	10.0	Seller	S6	-30.0
Complaint	**CoF**	Seller	S3	-22.5	Seller	S7	-62.5
Complaint	**No CoF**	Seller	S4	-25.0	Seller	S8	-65.0

Boson Iteration 3 (full payoffs)
Payoff Table

Actual
Buyer
No Seller Fault

		Redemption			**Non-redemption**		
No complaint	**No CoF**	Buyer	B1	0.0	Buyer	B5	-10.0
No complaint	**CoF**	Buyer	B2	5.0	Buyer	B6	5.0
Complaint	**CoF**	Buyer	B3	-15.0	Buyer	B7	-15.0
Complaint	**No CoF**	Buyer	B4	-20.0	Buyer	B8	-30.0

Deposit levels

As previously stated, deposits amounts are variable and are set as part of the commercial terms of the exchange. Here we briefly analyse the implications of various deposit levels.

When a Seller offers a voucher for sale at a price (Buyer payment amount), they also set the Buyer deposit and Seller deposit amounts. The Buyer deposit sets the Buyer commitment to redeem, whereas the Seller deposit sets the Seller commitment to a quality redemption. A Seller may choose to offer the same *thing* via multiple vouchers with different commercial terms, by varying the deposit (and payment) amounts. The taxonomy below describes the various commercial scenarios and deposit levels for Buyer and Seller.

		Seller Deposit	
		Low	**High**
Buyer Deposit	**High**	**1** High Seller loss for Buyer reversal Low Buyer loss for Seller CoF or high Buyer trust in Seller	**2** High Seller loss for Buyer reversal High Buyer loss for Seller CoF or low Buyer trust in Seller
	Low	**3** Low Seller loss for Buyer reversal Low Buyer loss for Seller CoF or high Buyer trust in Seller	**4** Low Seller loss for Buyer reversal High Buyer loss for Seller CoF or low Buyer trust in Seller

Taxonomy of Buyer and Seller deposit levels

Below we provide some example use cases per deposit level category.

1. **Low Seller deposit, High Buyer Deposit**
   Example
   Expensive restaurant with limited seating and low passing traffic.
   Seller will incur loss in event of a no-show. Buyer trusts Seller to redeem to quality.

2. **High Seller deposit, High Buyer deposit.**
   Example
   Travelling to another city to buy a second hand car
   Low Buyer and Seller trust, and high loss of reversal.

3. **Low Seller deposit, Low Buyer Deposit**
   Reversible, standardized quality (commoditized) transactions
   Example

Airport lounge visit. Buyer has no commitment to show.
Seller has no commitment to have room available

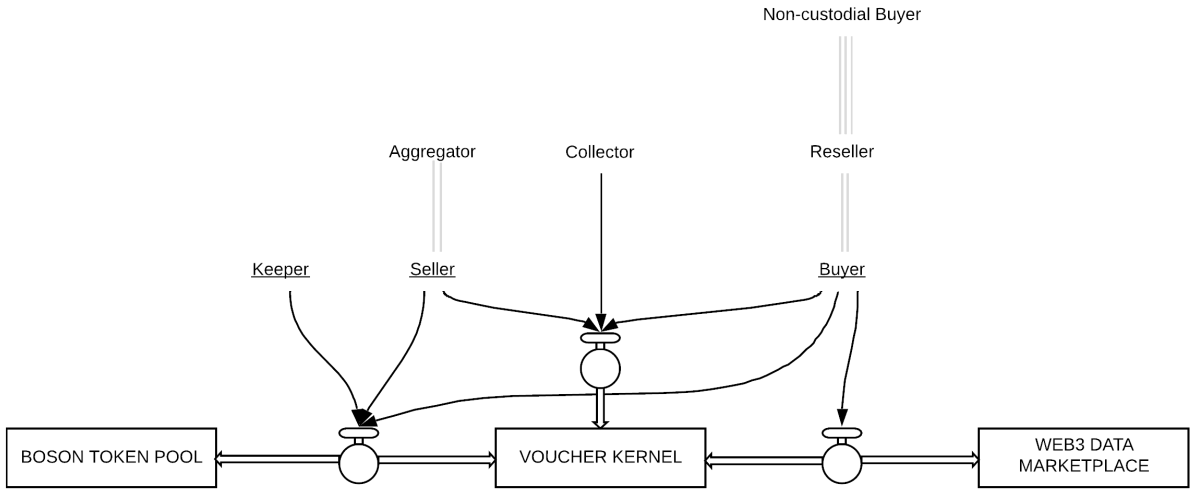
4. **High Seller deposit, Low Buyer Deposit**
   Example
   Busy tourist attraction.
   Buyer pays a premium (payment price) for guaranteed entry.
   Seller commits to access, but can replace Buyer at short notice due to high footfall.

Token Model and Incentives

We adopt the token design methodology, proposed by Trent McConaghy[10], to: 1) formulate the problem, 2) try an existing pattern and 3) use a new pattern only if needed. We view the token model as a modular building block, with a number of potential options for optimizing our objective function. At time of writing, we have identified that Ocean Protocol's Curated Proofs Market[11] (CPM) pattern meets our requirements. However, we are also investigating an emerging token model which also meets our requirements, in a simpler way. What follows therefore, is a high-level description of the current best candidate token model. Full implementation details are outside of the scope of this document and will be specified elsewhere.

Boson Protocol's objective function is to maximize the supply of quality redemptions. Boson incentivizes network actors to optimize this objective function by rewarding them with Boson's network token: Bosons. (Ticker name BOSON.)

The elementary agents in Boson Protocol are Sellers and Buyers, with their actions coordinated via smart contracts. Keepers[12] are providing support to the operation of the ecosystem. Aggregators can be considered as non-custodial hubs for multiple Sellers. Resellers act as custodial power Buyers that have their own, out-of-bond non-custodial buyers. Voucher Kernel set of smart contracts covers the business logic, the token pool is where accumulated tokens reside, while the web3-compliant data marketplace acts as a connector to an external environment to where the commercially interesting data of Boson Protocol agents and their transactions is routed.



Stakeholder mapping at a high level

[10] "Towards a Practice of Token Engineering - Ocean Protocol." 1 Mar. 2018, https://blog.oceanprotocol.com/towards-a-practice-of-token-engineering-b02feeeff7ca. Accessed 3 May. 2020.

[11] "Whitepaper - Ocean Protocol." https://oceanprotocol.com/tech-whitepaper.pdf. Accessed 3 May. 2020.

[12] For keepers we use Ryan Zurrer's definition:"a catchall term for the different utility players in distributed networks that maintain stability and perform crucial jobs in the crypto-economic model": https://medium.com/@rzurrer/keepers-workers-that-maintain-blockchain-networks-a40182615b66

These roles are elaborated within the stakeholder table below.

Stakeholder Role	Value contributed	Reward
Voucher Seller	Inventory	Boson tokens for supplying inventory
Voucher Aggregator	Onboard new sources of supply	Boson tokens for supply acquisition = early curation
Voucher Curator	Signals value, popularity and quality of vouchers	Boson tokens for curating
Voucher Buyer	Boson Tokens	Voucher inventory item
Voucher Relayers, marketplaces	Connect demand	Transaction fees for voucher orders
Data Buyers	Boson Tokens	Consumer product preference data

4.1 Curated proofs market

Bosons Protocol emits Boson tokens as rewards in order to incentivize actors to optimize the system objective function: to maximize the supply of quality redemptions.

Boson Protocol combines cryptographic proof of redemption and game theoretic quality signals with Curation Markets[13], to implement a Curated Proofs Market for physical things as non-fungible token voucher tokens. Each voucher type has a curation market, with a unique derivative token called Gluons, which measure the stake in the voucher type. Actors stake behind vouchers types which they predict will generate transaction value at an acceptable level of quality. Actors are then rewarded as a function of: the amount staked, the timing of the stake (with increased rewards for early stakes) and the actual transaction value and quality.

Assuming no computational constraints, the ideal allocation of rewards is calculated as follows. R_{ij} is the reward given to actor *i* for voucher *j*, prior to normalization

$$R_{ij} = log_{10}(S_{ij}) * log_{10}(V_j) * Q_j$$

And $R_{ij,\ norm}$ is the network rewards after normalization across all actors and voucher types.

$$R_{ij} = \frac{R_{ij}}{\sum_i \sum_j R_{ij}} * T$$

where:

- S_{ij} = actor *i*'s stake in voucher *j*, measured in Gluons, the derivative token for a particular voucher type.
- V_j = value of no-complaint redemptions per block interval for voucher *j, see details below.*
- Q_j = ratio of no-complaint redemptions to total redemptions for voucher *j*, see details below.
- *T* = total Boson Tokens awarded for the block interval according to the rewards schedule.

Description of terms

- $log_{10}(S_{ij})$represents an actor's prediction of the level of quality redemptions for a voucher *j*, using the metric of the value of transactions with no complaints, measured in Gluons. If a Seller posts a voucher believing that it will deliver a high value of no-complaint transactions, then they may stake more than the minimum amount in order to receive more Gluons and the chance to receive more rewards. Any actor may stake behind a voucher, but importantly, Aggregators are well placed to benefit from the early adopter increased rewards by staking early behind high value-quality vouchers

[13] "Introducing Curation Markets: Trade Popularity of Memes ...." https://medium.com/@simondlr/introducing-curation-markets-trade-popularity-of-memes-information-with-code-70bf6fed9881. Accessed 3 May. 2020.

which they have sourced and onboarded. ($\log_{10}$ is applied in order to mitigate token whales and encourage the onboarding of a higher number of vouchers.)

- $log_{10}(V_j)$ represents the actual level of quality redemptions for a voucher j, using the metric of the value of transactions with no complaints.

These first two terms are core to the Curated Proofs Market mechanism and reflect the relationship between the predicted and actual (proofed) level of quality transactions.

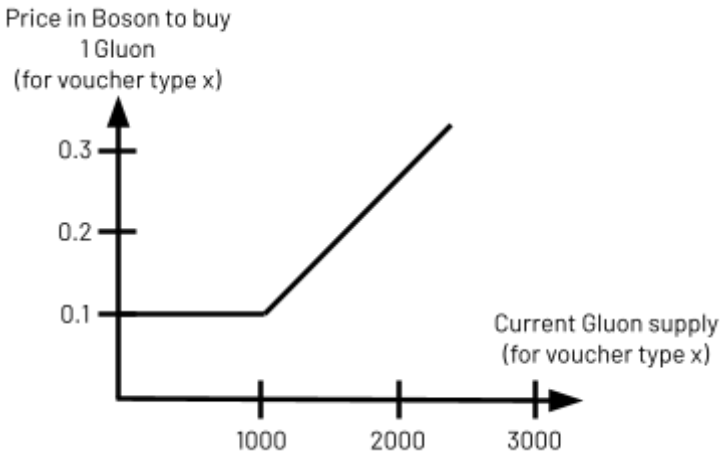
- Q_j is introduced to incentivize a high percentage of non-complaint transactions, which factors-in the failure rate as per standard service level schemes. This ensures that vouchers with a high complaint rate are penalized, irrespective of whether they have a high total value of complaint-free transactions.

Curation markets

Curation markets are used to signal how much supply of quality redemptions a voucher type might deliver. Each voucher type has a curation market, where actors can signal by staking and unstaking, a voucher-specific Gluon token, whose price is related to $BOSON tokens via a voucher-specific bonding curve.

Bonding curves

The bonding curve below describes the relationship between a particular voucher type's derivative Gluon token and Boson tokens. The curve provides an initial low price for voucher-specific Gluon tokens, the price then increases as more tokens are purchased.



Bonding curve for tokens for voucher type x

This enables early stakers to buy tokens in a particular voucher type at a discount. This rewards Sellers who stake behind their own vouchers and Aggregators who onboard new vouchers, as well as Curators, who provide a valuable service.

Actors can thus capture value from their activities from:

- Staking rewards - by staking on a successful voucher type
- Un-staking - by buying early at low cost and then unstaking at a profit

Governance

Whilst the end-game for Boson Protocol is a fully decentralized protocol, the governance requirements for developing a new protocol will evolve through time. In recognition of this, Boson adopts a progressive decentralization[14] strategy across three phases.

First, as a start-up, the protocol will need to achieve protocol-market fit by rapidly iterating across multiple build-measure-learn cycles. This requires strong opinionated leadership with the ability to respond rapidly to market feedback. Therefore smart contracts will have admin rights and the protocol will be centrally controlled and developed.

Next, as the protocol scales-up, network effects will provide sufficient defensibility to enable a minimally extractive[15][16] fee to be charged for vouchers. This fee has the potential to capture significant value, which will enable the community to be incentivized to contribute valuable work via token incentive schemes, grants and bounties. The founding team will step back and pass more control and responsibility to the community via increased community engagement and input through rough consensus and permissioned voting.

> *"minimal extraction doesn't mean cryptoassets that capitalize protocols will capture minimal value; if something is minimally extractive, but globally produced and consumed, the coordinating asset can capture a significant amount of value"*. Chris Burniske[17]

Finally, having built a strong community, and with a clear model for a sustainable decentralized network, the core team will step back and allow the community to govern the protocol. Ownership and control of the protocol will be distributed to the community. At time of writing, it is anticipated that the governance structure will be a decentralized autonomous organization (DAO), structured to ensure regulatory compliance as well as resist capture from centralized entities or groups. However, this is an area of rapid development , and so the final governance model is subject to change. Community operation will be sustained via value captured from minimally extractive fees.

---

[14] "Progressive Decentralization: A Playbook for Building Crypto ...." 9 Jan. 2020, https://a16z.com/2020/01/09/progressive-decentralization-crypto-product-management/. Accessed 2 Jun. 2020.

[15] "Progressive Decentralization: A Playbook for Building Crypto ...." 9 Jan. 2020, https://a16z.com/2020/01/09/progressive-decentralization-crypto-product-management/. Accessed 2 Jun. 2020.

[16] "Protocols as Minimally Extractive Coordinators — Placeholder." 6 Oct. 2019, https://www.placeholder.vc/blog/2019/10/6/protocols-as-minimally-extractive-coordinators. Accessed 2 Jun. 2020.

[17] "Protocols as Minimally Extractive Coordinators - Placeholder VC." 6 Oct. 2019, https://www.placeholder.vc/blog/2019/10/6/protocols-as-minimally-extractive-coordinators. Accessed 24 May. 2020.

Technology overview

Introduction to the design

Boson Protocol enables smart contracts to perform on-chain financial transfers in line with the off-chain, often physical, delivery of goods and services. The ultimate goal is to do it in an atomic and autonomous manner - any other way raises security/trust issues and casts doubts on using blockchain in the first place.

Atomicity means that the financial exchange happens if and only if the goods or services are exchanged simultaneously. The autonomy in transacting implies that a smart contract contains enough logic and data to execute the transaction on its own, without depending on another trusted entity.

Representing goods and services as on-chain tokens doesn't provide the means to control physical transfers from within the isolated and discrete blockchain world. Instead, Boson Protocol builds on the notion of *promises*. A good or a service is promised to be implemented, usually with a time delay, by the issuer of the promise, under the promised conditions, to the holder of that promise. The vehicle to represent a promise is, quite intuitively, a voucher. Thus, vouchers are exactly the objects that Boson Protocol tokenizes and uses heavily.

Redemption of a voucher releases the escrowed payment and here lies the delicate challenge of how to guarantee atomicity - the dilemma that is usually solved by introducing intermediaries to the detriment of decentralization, Boson Protocol favors decentralization with the trade-off of requiring proximity in time-space at redemption.

Boson Protocol, at its core, is highly *practical* and commercial in its nature. The user experience of using vouchers is not changed beyond added self-sovereignty over data and value, which is significant on its own, but the user's journey largely follows the already perfected traditional, centralized voucher apps. The crux of the differentiation lies in the open, non-proprietary use and programmable components.

There is one more caveat: the throughput of token transactions on common blockchain networks and the associated costs. Our research shows that existing global voucher transactions are in hundreds of millions per year[18]. Therefore, our solution will on average have to handle about 10 transactions per second, with peaks at least 10x that.

[18] E.g. Ethereum has about 300,000 token transactions per day, from thousands of different token contracts. Note that token transfers within a single contract can be significantly optimized by batching. See: https://decrypt.co/11655/report-what-the-ethereum-transaction-flippening-means and https://www.readycloud.com/info/groupon-statistics-that-will-make-you-rethink-digital-coupons

Design requirements

In order to provide uniform access, the protocol must be open and decentralized, making it a good fit for using blockchain technology. The actual degree of required decentralization is hard to pinpoint and while there are known benefits and tradeoffs with maximizing it, such as resilience versus throughput, we acknowledge the nascency of this space and plan for continuously adapting to the state-of-the-art infrastructure.

Blockchain-based systems also provide radically improved security over traditional vouchers (though admittedly adding non-negligible friction to the user experience), e.g. preventing forgery and alteration. It is impossible to forge a Voucher Token as long as standard cryptographic primitives remain secure. While traditional, centralized voucher systems are typically limited in ways they can validate the redemption of a voucher, Boson Protocol offers significantly more powerful verification capabilities. Vouchers also cannot be altered after issuance. Voucher orders can only be cancelled by the issuer.

There are several additional benefits of vouchers that are blockchain-based, such as: preventing duplicate redemption: a voucher can only be redeemed by the voucher holder. Once consumed, it cannot be redeemed again; if the voucher can be used repeatedly, for example a membership card, it is bound to an expiration period. Non-repudiation: it is not possible to repudiate issuance, trade or redemption by actors, as their digital signatures bind them to the commitments.

The increased friction for users is mainly a result of manipulating cryptographic transactions, that is, the management of private keys, signing transactions, and running a blockchain node. We address these issues by adopting best practices and by maintaining partnerships with complementary solution providers. To that end, we are investigating Universal Login[19] for managing decentralized identities; Trustology[20] for custodial wallet; Torus[21] as a key management solution integrated with OAuth providers or Argent[22] that uses so-called guardians to delegate key management via social or hardware delegation; using Clef[23] to decouple secure transaction signing away from interacting with untrusted, often remote blockchain nodes; then Moonlet Portal[24] as a fully-featured SDK to interface with the web3 esp. for end-user facing applications; also Biconomy[25] for relayer infrastructure.

Since Boson Protocol is middleware, it enables front-end applications to be built on top of it in arbitrary ways. However, effort is made to make the blockchain experience comparatively safe and straightforward to its primary users, that is, businesses. With that in mind, we aim to offer both thick reference clients to be run on-premise or connect to remote nodes, operated by external providers. There is a tangible lack of support in this area, but not due to

[19] https://universallogin.io/
[20] https://trustology.io/
[21] https://tor.us/
[22] https://www.argent.xyz/
[23] https://github.com/ethereum/go-ethereum/tree/master/cmd/clef
[24] https://moonlet.xyz/
[25] https://www.biconomy.io/

node-as-a-service vendor lock-ins[26]. If merchants are to become first-class users in the blockchain space, projects such as light clients (and for end-users, ultra-light clients[27], capable of running from mobile devices) need much more attention. As is the case for protecting those nodes from the public, adversarial network by shielding them with sentinel nodes. Going to the extreme, nodes would have a radically small attack vector and be at performance optimum, if built as unikernels[28], specialized machine images that run directly on bare metal. This single gem has been neglected for years[29], but if any software is suitable for being deployed as unikernel, those are high-value nodes.

Last but not least, privacy by design is a required feature to address the toxicity of data. In practical terms it means that sharing personally identifiable information (PII) follows strict consent-based policies and is done against equitable compensation. Out of the four privacy aspects (i.e. PII, transaction amounts, identities of senders and recipients, asset information), the first two are most relevant. Since Boson Protocol merely connects the owners of PII to the data demand side, PII is not stored or processed on Boson. Transaction amounts are subject to optional concealing, with several zero-knowledge approaches being researched.

Main concepts

Promises and vouchers

Let $Asset$be claimed at the redemption of a voucher, i.e. the goods or services delivered. It is offered by the Seller Se to the Buyer Bu. Versioning enables continuous operation of older assets to coexist with new ones. Categorization is used for narrowing the discovery of vouchers and can include prefixed namespaces for more detailed categorization. An asset can be further described with a pointer to additional information, URI (Uniform Resource Identifier).

$$Asset \equiv \{ID, Se, version, title, description, category, URI, merchandise, conditions_{TXT}\}$$

where ID_{asset} is asset's identification, calculated as:

$$ID_{asset} \equiv hash(Se,\ version,\ title)$$

$merchandise$is an optional, collector-specific meaning of the voucher, important for the Collector's interpretation, such as a voucher identification by the Collector or a pointer to an external restrictions object. Note that if $merchandise$is specified, then $conditions_{TXT}$must also be specified in order to enable understanding of restrictions in natural language.

$$merchandise \equiv ID^{Cl}_{voucher} \vee ext.restrictions^{Cl}_{voucher}$$

[26] Infura processes billions of interactions per day, which is beneficial to them, however they are actively trying to push the pendulum in the other direction as well. For example, see: https://blog.infura.io/investing-in-the-decentralization-of-ethereum-da59a734f61e

[27] Ultra Light Client in detail: https://hackmd.io/@GMFZzCl1SH6s2lX25nC15A/HJy7jjZpm?type=view. Also Incubed client from Slock.it: https://in3.readthedocs.io/en/latest/intro.html

[28] http://unikernel.org/

[29] https://github.com/ethereum/webthree-umbrella/issues/98

Let $Promise$be the promise of the Seller to deliver the Asset to the Buyer under programmable restrictions on redemption, collection, monetary allocations and human-readable conditions. Promises are a core construct that enable reusability across multitude of participants, while still being anchored to the same Seller.

$Promise \equiv \{ID_{asset}, value, challenge_{period}, cancelFault_{period}, quantity, validity_{period}\}$

$value$is an encoded value of the voucher, defined with the type of the voucher, the type of the representation of value which can be either amount or percentage, currency denomination, and the number of vouchers needed for redemption (zero, if the voucher can be used repeatedly and is therefore not consumed).

$value \equiv \{type_{voucher}, type_{value}, quantity, currency, spend\}$

where:

$type_{voucher} \in \{exchange\ if\ 0, discount\ if\ 1, monetary\ if\ 2\}$

$type_{value} \in \{fixed\ if\ 0, ratio\ if\ 1 \wedge type_{voucher} = discount\}$

$quantity \in \{amount\ if\ type_{value} = 0, percentage\ if\ type_{value} = 1\}$

$spend \in \{n_{consume}\ if > 0, 0_{present}\ if\ 0\}$

$challenge_{period}$ is optional. The redemption process can support a challenge by the Buyer when $challenge_{period} > 0$.

$cancelFault_{period}$ is optional. The promise can support a cancelation of the already sold promise or admitting a fault at redemption by the Seller when $cancelFault_{period} > 0$.

$quantity$ denotes the number of same vouchers in the group. Default is 1, meaning that a default group of equal ("fungible") vouchers represents only one "non-fungible" voucher.

Finally, a $Voucher$ is instantiated from the $Promise$ when it is tokenized and assigned to a holder (when minted, the holder is the Seller, and is later sold/transferred to the Buyer). The identification of the voucher is specified later.

$Voucher \equiv \{ID_{voucher}, holder, ID_{promise}\}$

Overview of actors

The details of the actors in Boson Protocol are specified in a separate document, here is but an overview to understand the general activities.

At the conceptual level, there are: Sellers, Buyers and Keepers. Sellers can have logically separated accounts for making offers and deposits management, and can delegate some of the functionalities to other logical accounts (e.g. a rudimentary hierarchy of capabilities across multiple key pairs). Keepers consist of: Relayers, Curators, Schedulers, Coordinators.

User	Goals	Context	Defined in

Seller	Offers goods and services and commits to their quality	Elementary actor, representing the supply side.	BOSON
Buyer	Buys vouchers to redeem them later for goods or services	Elementary actor, representing the demand side.	BOSON
Aggregator	Sources supply into BOSON as B2B	Custodial hub for multiple sellers.	externally
Reseller	Leverages BOSON's voucher infrastructure for B2C	Custodial "power buyer". Has their own end-user app.	externally
Non-crypto Customer	Uses vouchers in a traditional, off-chain way	Non-custodial customer of Reseller. No blockchain interactions required.	externally
Relayer	Matches Sellers supply with Buyers demand	Non-custodial intermediary. Receives fees for matching.	BOSON
Collector	Collects the voucher and implements the promise	No blockchain interactions required.	externally
Curator	Curates offered vouchers	Semi-automated or even AI-driven. Rewarded for work.	BOSON
Scheduler	Triggers scheduled tasks	Automated service. Rewarded for work.	BOSON
Coordinator	Votes for protocol updates	The token holder with aspirations to benefit to and from the protocol. Has a deep understanding of BOSON.	BOSON
API Operator	Provides API towards BOSON, RDM	Automated service.	externally
Data Buyer	Purchases voucher and demographic data from Buyers	External entity	BOSON

Note, that *BOSON* stands for Boson Protocol and *externally* means a construct built on top of Boson, e.g. a persona like an Aggregator could be defined in a rewards entity built on top of the decentralized Boson infrastructure.

Technology

To the maximum extent possible, components in the stack are developed in a modular way, enabling updating or replacing building blocks as better alternatives are discovered.

Boson Protocol will start by building on Ethereum as it is the most battle tested and widely accepted ecosystem. Later on, we will follow the state of the art and potentially support multiple blockchain types.

Architecture

NB, as the underlying technology is rapidly evolving, we are planning multiple iterations of the architecture, starting from a simpler design for a limited audience, to more complex designs of which we are keen on zk-rollups, discussed below.

Release 1: The MVP will be designed as an on-chain solution, leveraging proven ERC standards, enabling fast deliveries - with the largest tradeoff being scalability.

See the Tokens management section for details.

Additional performance could be achieved by temporarily relaxed ERC standard compliance, thus re-evaluating the requirement for voucher token design and/or just-in-time minting (a.k.a. segregated activation)[30].

Release 2: In order to support massive numbers of vouchers, the full product will have the majority of processing in Boson Protocol performed off the main Ethereum blockchain. Instead, it is done on a sidechain, that is linked to the main chain by way of rollups. This means that state updates of the sidechain are published on the main chain, as well as sufficient data and state transition proof, which in the zero-knowledge rollup[31] means that the off-chain state is correct and consistent.

Rollup techniques are currently designed for hundreds of non-trivial transactions per second for simple token transfers. Conservatively decreasing these numbers by an order of magnitude, on account of complex transactions in Boson Protocol, the throughput is still sufficient with regards to our initial goal.

Vouchers are created and live predominantly in the sidechain. On-demand, a voucher could be exported to Ethereum main chain, where it could be traded or used in arbitrary ways, but to redeem or refund it, it must be imported back to the sidechain.

The operator of the bridge is a non-custodial, untrusted entity, which can eventually be delegated to a pool of such keepers.

Boson Protocol maintains a set of smart contracts on Ethereum main net, that control the protocol's surface and act as gatekeepers to the sidechain:

- inflow of staked voucher offers and staked requests to purchase vouchers,
- extraction and return of standard-compliant NFTs between the sidechain and Ethereum, for purposes of secondary trading, leveraging DeFi products and similar,
- outflow of funds from finalized vouchers inside the sidechain back to Ethereum,
- decentralized identity management and light-weight reputation system.

Identities and authentication (including AML/KYC restrictions etc.) are controlled at the surface, which thus acts as a gatekeeper to the off-chain processing.

The circuits in the sidechain support the core of the business logic, which is predominantly covering the life cycle of vouchers, the token model and the disbursement of funds.

---

[30] See for example the transaction count of Gods Unchained at: https://public.tableau.com/profile/alethio#!/vizhome/shared/KJRNW2796

[31] More information about the promising zk-rollup approach is gathered at: https://github.com/thecryptofruit/education/blob/master/zk-proofs-rollups.md#zkr

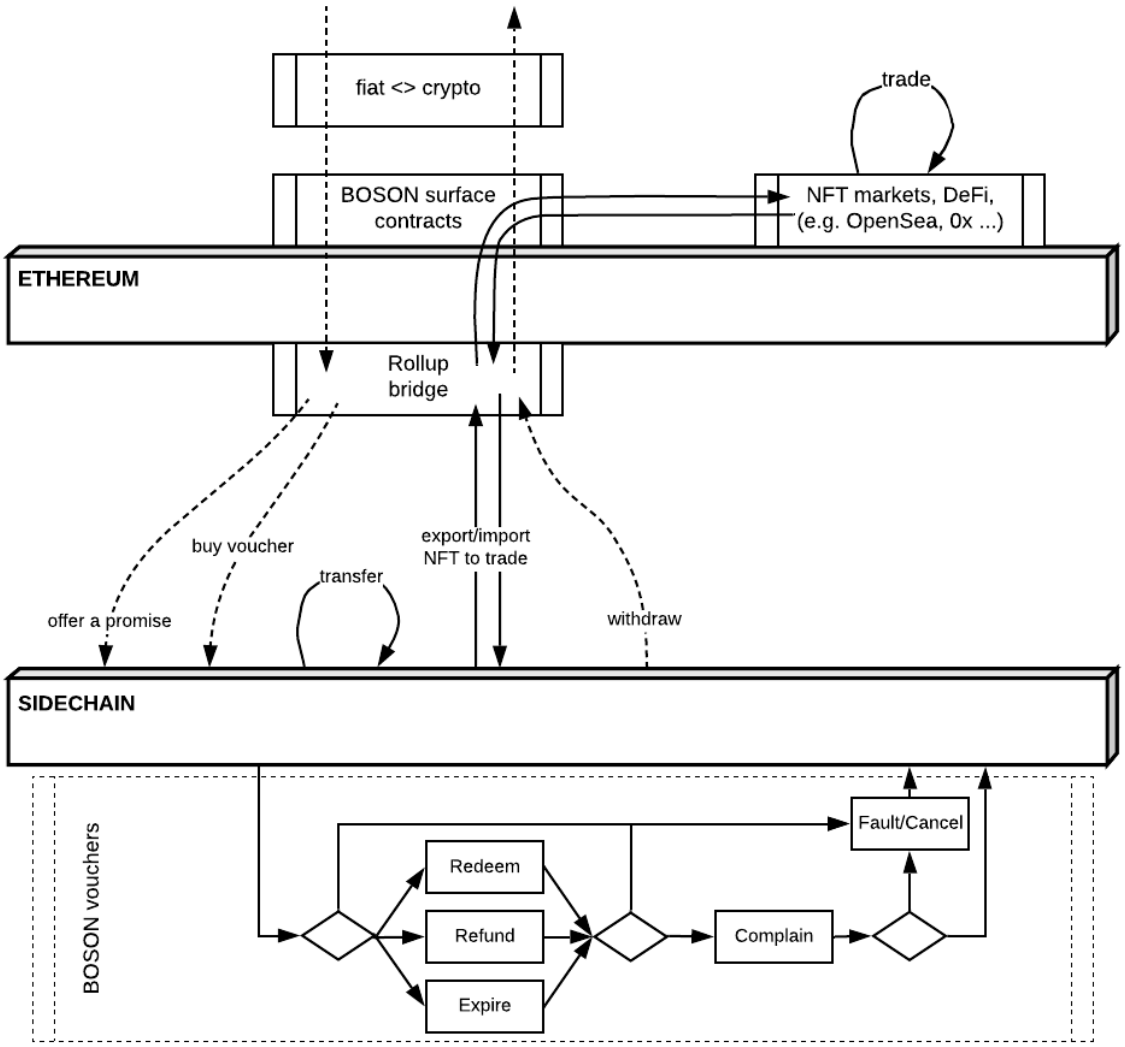


Figure 2: architecture in a rollup setup

From the interoperability point of view, there are two aspects to Boson Vouchers. Firstly, the core voucher life cycle happens in a relatively closed context. Secondly, the trading, transferring, lending, borrowing and so on, are orthogonal to the core, which implies that the exposure to the risks in the wider space are limited. Here we mean DeFi in particular, as the risks there do affect the liquidity side in Boson Protocol, but do not affect its core operation. Thus, Boson relies on the security of the base layer, e.g. Ethereum mainnet.

Seller's deposits could be managed and optimized with regards to the current state, such as maintaining only a minimal pool of available funds. The assets on offer do carry Seller's commitment, but it could be checked only at the time of purchase. Furthermore, note that vouchers have a start and an end date of when they can be redeemed or disputed, which provides valuable insight to the Seller of when a specific voucher might be used - thus forecasting when the security deposit must be available. This enables an additional possibility of having the locked funds temporarily forwarded into DeFi products.

Transaction types

Most important transactions in Boson Protocol are concerning the state of NFTs.

Seller:

- mint - creating a supply voucher token, acting as an order;
- cancelOrFault - canceling an offer or admitting fault for committed vouchers;
- close - closing an order.

Buyer:

- buy - buying a voucher token, therefore decreasing the supply in the seller's order;
- transfer - transfering a voucher token;
- export - exporting to the main blockchain network, e.g. Ethereum mainnet;
- import - importing from the main blockchain network back to the off-chain realm;
- redeem - redeeming a voucher token for the promised good or service;
- refund - refunding the payment and making the voucher token non-redeemable;
- complain - issuing a complaint before or after voucher redemption.

Keeper:

- expire - triggering the expiration flag on expired vouchers;
- withdraw - withdrawing available funds in push mode or only unlocking funds in pull mode;
- setExchangeRate - setting the exchange rate between currencies used;
- ...

Note that keepers have several roles:
$Keeper \in \{Relayer, Curator, Scheduler, Coordinator\}$

For details about transaction formats, see Appendix - Transaction formats.

Tokens management

Smart contracts for managing tokens, such as minting, transferring and burning, are interacting with the core logic in a maximally restricted way. There are potentially multiple token types used, some representing actual vouchers, others used for payments, interactions with the external ecosystem, financing etc.

Tokens used for release 1 (Ethereum layer 1):

- natively using ERC-1155 standard for non-fungible voucher tokens and fungible work tokens,
- payment tokens conforming to the ERC-20 standard[32],
- later, more native token types could be incorporated, depending on the state-of-the-art research on curation, reputation, coordination etc.

[32] Also considering a more advanced ERC-777: https://eips.ethereum.org/EIPS/eip-777

Using a single token factory for multiple native token types has a significant benefit of consolidating a diverse token logic and enables batch processing to reduce transaction costs.

Using ERC-1155 and ERC-721 standards together adds contract complexity and costs, which is alleviated to some degree by utilizing the overlap in several properties (e.g. transfer approvals, metadata etc.). Depending on the market trends, ERC-721 could eventually be omitted in favor of ERC-1155, which would decrease some operational costs.

Since vouchers are specified with rich descriptions, we are supporting the new standard ERC-2477 Token Metadata Integrity[33], to guarantee token metadata integrity for all tokens from the factory - fungible and all non-fungible ones.

Token IDs are calculated using a common algorithm, that enables easy off-chain derivation of some of the token characteristics. See Appendix - Token ID specification for details.

Bundling

Packaging multiple vouchers into a bundle enables interesting use cases, for example offering a package of services (e.g. purchasing a spare tire combined with service work). The key consideration is the level of coupling:

1. tightly coupled, where a set of NFTs is mapped to one new NFT - bundle-type NFT: supporting this scenario is ERC-998[34], with the cost tradeoff for on-chain bundling;
2. loosely coupled, leveraging batch transfers within a multi-token factory: this is achievable out-of-the-box via ERC-1155[35], which leaves the exact bundling logic to the off-chain logic;
3. and the degree of backward compatibility between NFT standards used: non-backward compatible ERC-1155 or backward-compatible emerging standard ERC-2547[36], which again incurs a costs tradeoff.

All bundling implementations must consider the challenge of buy-a-bundle, redeem-a-bundle, complain-about-one-part.

Coordination

With the ultimate goal being a public utility, its governance will evolve from a centralized control, used to ensure smooth initial operation, later gradually towards a decentralized model. The details of this transition will be specified when the wider ecosystem matures enough to ensure the necessary robustness of decentralized governance. We are actively participating in the community to contribute to this nascent space and are taking measures that decentralized governance does happen the soonest, e.g. releasing development funds as milestones reached.

---

[33] https://github.com/ethereum/EIPs/blob/3518f3086476a0bc03c0b754ed3b73bc521176c5/EIPS/eip-2477.md

[34] https://github.com/ethereum/EIPs/blob/master/EIPS/eip-998.md

[35] https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1155.md

[36] https://github.com/ethereum/EIPs/blob/2587c1461a2f7bc2aa917b893fa896956a6f97c1/EIPS/eip-2547.md

Curation of the content could be considered as a subset of coordination. Its design is pending further research as we are still looking into solutions that would fit decentrally managed inventory, without sacrificing the trust model. As a starting point, the content might undergo several verifications, and better approaches are to be applied when proven. We acknowledge that the state of regulation in any jurisdiction right now, as clear or fuzzy it may be, will not stay the same in the next couple of years, due to the sheer growth of a decentralized economy.

Apart from the overarching governance, coordination also requires an identity system that goes well beyond one key-pair : one user. In other words, cryptographic security is not a sufficient property of an identity, there must be other, more practical nuances than that, such as: key recovery, key rotation, multiple-key use, delegation of permissions, ease of use, etc.

The basis for actors in Boson Protocol will be a decentralized identity model (DID), initially loosely designed, to enable future adaptations[37].

- DID (erc725 good segregation of DII and multiple keys/accounts, but also need recoverability, but also issues: base protocols often require 1 key : 1 user (PSS crypto, 3box even ...)

SDK-driven DX

To complement Boson Protocol as middleware, an extensive Software-Development Kit is being developed, that will provide state-of-the-art developer experience.

The scope and technology used in the SDK will initially be adapted to the requirements of the first partners.

Integrations, reference clients

Lastly, to help with user onboarding, due support will be provided with integrations to other web3 and legacy systems, as well as maintaining ready-to-use reference clients.

Roadmap

Leveraging the most battle-tested and widely adopted blockchain ecosystem, we have started building on Ethereum. This enables us the fastest time-to-deploy and access to the largest user base at the time. Further in the future, going with Ethereum now gives us the option to scale horizontally, if need be, to potentially multiple Ethereum-based Cosmos zones, a.k.a. Ethermint[38], which are attracting an increasingly wider audience. Spreading to other EVM-based networks, such as RSK[39] also remains an option. In principle, we remain open to new arrivals, such as Solana[40], but have put Ethereum first and are researching some of its novel scaling approaches, predominantly rollups.

---

[37] ERC-725 has a good segregation of DII and multiple keys/accounts, but the requirement must also be recoverability and legal compliance.
[38] https://github.com/ChainSafe/ethermint
[39] https://www.rsk.co/
[40] https://solana.com/

We have a short-term plan to build and deploy an MVP in 2020, consisting of two major parts: the minimum viable core of Boson Protocol and the integration with the first customer(s). Technological stack will consist of three components: the main business logic on Ethereum, comprehensive SDK to interact with it, reference back-office client for merchants and reference mobile app for end-users.

Thus, the one-to-two year plan is to build on top of Ethereum main net, with maximally optimized components and implementing a logic that updates the blockchain state as minimally as possible, using tight data packing, batch processing, eventual rollups, etc. 2+ year plan is to potentially explore other blockchain networks, also taking into account that the goal of Boson Protocol is to be a pluggable web3 component, that is, to be generally interoperable. Compatibility with multiple blockchains presents a challenge on its own and is especially delicate in terms of coordination over heterogeneous partitions. We are researching several approaches - most notably, 0x is spearheading this space with the EXTCODEHASH "metamodel" (must watch: https://youtu.be/iU0FkWpFyUY)

Technology appendix

Token ID specification within Ethereum network

256-bit identifier starts with one bit specifying whether it is a fungible or non-fungible token, followed by 127 bits for base token type, followed by either 128-bit zeroes or, in the case of an ERC-721 token, its index in the base token superset.
<0: N/F><1-127: base token><128-255: index of extracted ERC721>

For fungible tokens, this would manifest in:
<0><uint127: base token id><uint128: 0>

For non-fungible supply tokens, following ERC-1155 standard, this would manifest in:
<1><uint127: base token id><uint128: 0>

For non-fungible voucher tokens, extracted from supply tokens, following ERC-721 standard, this would manifest in:
<1><uint127: base token id><uint128: index of non-fungible>

State representation in the rollup

Depending on the zk-rollup capabilities at the time of use, Boson Protocol state could be organized in sparse trees in a number of ways, e.g. one or multiple state trees. Here we propose account-balance-state design, recording voucher holders (account-ownership-escrow) and voucher data (note that most of asset's descriptive data, a.k.a. token metadata, is stored off-chain in a decentralized storage).

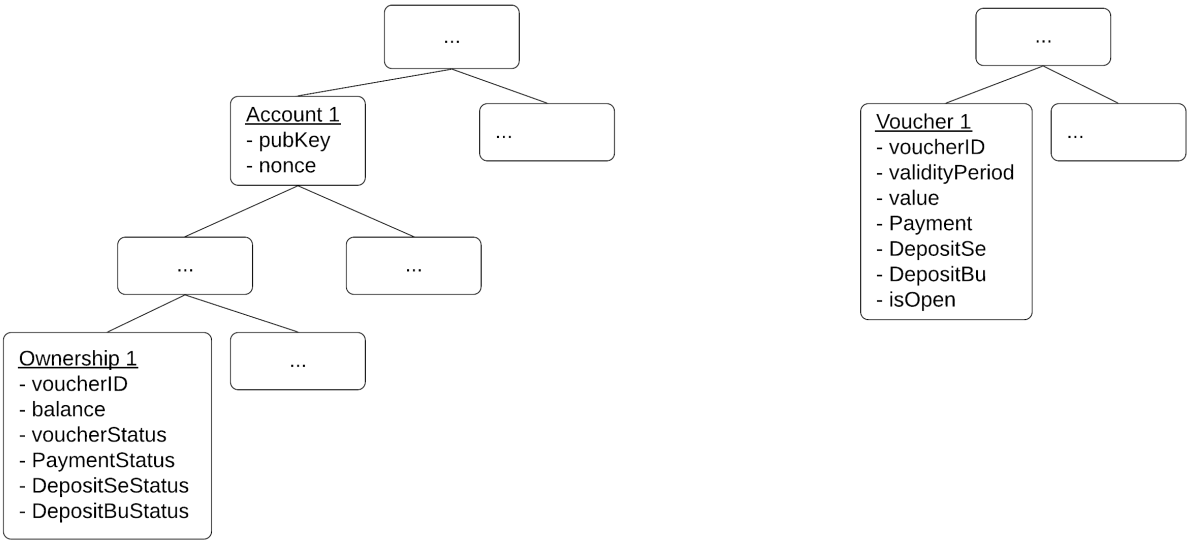


Figure 3: vouchers state in a rollup setup

Transaction data packing in the rollup

txType	8-bit
accountID	24-bit (65-byte pubkey representation, though some use 5-bytes truncated Pedersen hashes)
voucherID	32-bit
amount	24-bit
balance	12-bit
voucherStatus	8-bit
paymentStatus	1-bit
depositSeStatus	1-bit
depositBuStatus	1-bit
validityPeriod	24-bit
value	16-bit
payment	24-bit
depositSe	24-bit
depositBu	24-bit
isOpen	1-bit

Transaction formats in the rollup

Tx to create new vouchers: mint [8+24+32+24+24+16+24+24+24 = 200 bits]

TxType	acount_from	voucherID	amount	validityPeriod	value	payment	depositSe	depositBu

Txs that change ownership: buy, transfer, export, import [8+24+24+32+24 = 112 bits]

TxType	acount_from	account_to	voucherID	amount

Txs of a voucher life cycle: cancelOrFault, close, redeem, refund, complain [8+24+32 = 64 bits]

TxType	acount_from	voucherID

Txs for keepers at regular intervals: expire, withdraw [8+24 = 32 bits]

TxType	acount_from

Txs for keepers, fired on changes only: setExchangeRate [8+24+8+8+16 = 64 bits]

TxType	acount_from	currency1	currency2	rate

Appendices

Existing approaches

Several approaches exist to bridge the gap between the real, often physical world and the economies in the growing decentralized, blockchain-based realm. Choosing one depends on a particular use case and contextual constraints.

Tokenizing items in the most straightforward way works by creating an on-chain twin of a real-world thing, represented as a non-fungible token. We can embed a unique identifier onto the thing itself and operate with this identifier on the blockchain network. Blockchain technology is quite good for provenance tracking as it offers a shared ledger with irreversible transactions. The downsides are the costs of having chips on all items, it is not suitable for services and the exchange of the physical item is not enforced on-chain and vice-versa. Examples: Cryptokaiju[41], Colletrix[42].

Physical-to-blockchain transubstantiation can be performed by oracles, which act as (semi-) trusted bridges between the off-chain and on-chain worlds. Trades of digital content are quite suitable for this approach, though can be applied for practically anything as long as oracles are set up, secured, highly-available and paid for. Examples: Chainlink[43], Provable[44].

Tokenizing contract of ownership is suitable for cases where there already exist business contracts that specify the ownership of the item, as contracts can be structured, parsed and tokenized efficiently. However, requiring formal contracts for all trades could prove prohibitively costly as well as potentially inducing unwanted legal ramifications. Example: Centrifuge[45].

Tokenized ownership under an autonomous object is a sophisticated approach, consisting of on-chain records of item's ownership tracking as well as arbitrary set of claims about its state that are made and potentially challenged meritocratically. It is a good match for items with a high perceived value, less so for others. Example: Asset Passport, created by Mattereum[46].

Tokenizing convictions for the performance of the trade works by creating a prediction market for a specific event, which works well for more exposed use cases, but is infeasible to be applied to each and every commercial transaction. Examples: Augur[47], Gnosis[48].

[41] https://cryptokaiju.io/
[42] https://www.colletrix.com/
[43] https://chain.link/
[44] https://provable.xyz/
[45] https://centrifuge.io/
[46] https://mattereum.com
[47] https://www.augur.net/
[48] https://gnosis.io/

Previous work

There exists a variety of on-chain solutions to the problem. These include *caveat emptor*, reputation based systems and arbitrated multi-signature transactions. However, the holy grail would be a mechanism that is programmatic, avoids human arbitration and is incentive compatible.

1. Satoshi described a simple escrow in August 2010[49]. It involves the buyer's payment in escrow, that is either released by the buyer when he/she receives the goods - or the funds remain stuck. Despite its simplicity, it is quite elegant in the sense that the seller has no incentive to cheat, while the buyer doesn't profit for not releasing the payment as it is essentially burned. The downside is that if the buyer is lazy or uncooperative, the seller loses.
2. Gavin Andresen evolved it in March 2012[50]. He added an expiration period, after which the seller can unlock the payment, provided that the buyer didn't raise a dispute. This process already covered more edge cases, but the dispute is problematic. Gavin described three possibilities: either the funds go to the blockchain miners as fees, or go to a trusted arbitrator, or go to a shared beneficiary, such as a charity.
3. Alan Reiner evolved it further in April 2012[51] by constructing a 2-sided "Risk Deposit", which a third party can arbitrate in one's favor - or is forever lost in the case without an arbitrator. At the same time, Alan Reiner acknowledges that forever locked funds could potentially be transferred elsewhere, pending that Bitcoin tech eventually supports it[52].
4. NashX implemented the idea of an intermediated escrow in 2013[53].
5. Oleg Andreev published an article about "contracts without trust or third parties" (which seems very much like the Alan Reiner's) in August 2013[54], notably adding also David Friedman's suggestion of expiration in his pittoresque "bilateral hostage solution"[55] (which was discussed between Gavin and Alan in 2012).
6. Kleros[56], founded in 2017, employs crowdsourcing as a dispute resolution mechanism, where jurors put stake on their actions. It is a comprehensive set of machinery, referencing existing courts, thus seems a good potential candidate for legacy-like interventions.
7. Colony, founded in 2014, has developed a meritocratic dispute resolution[57] in Q1 2017, performed by staking the reputation for those that challenge and those that want to keep the current, unchallenged state. The resolution is engaging multiple, "contextually-relevant" participants. Essentially, as long as the action is not challenged by the members of the community, nobody bothers (that is, there is no on-chain action

---

[49] Satoshi Nakamoto: https://bitcointalk.org/index.php?topic=750

[50] Gavin Andresen: https://gist.github.com/gavinandresen/830ca16758fb9ad496d7

[51] Alan Reiner: https://gist.github.com/etotheipi/2305966

[52] Alan Reiner v2: https://bitcointalk.org/index.php?topic=75481.msg837002#msg837002

[53] NashX: http://nashx.com/About

[54] Oleg Andreev's blog post, as an updated version linked there is no longer accessible: https://blog.oleganza.com/post/58240549599/contracts-without-trust-or-third-parties

[55] David Friedman's contemplation: http://daviddfriedman.blogspot.com/2013/08/a-bilateral-hostage-via-bitcoin.html

[56] Kleros as an opt-in Ethereum-based court: https://kleros.io/en/

[57] Colony's meritocracy: https://colony.io/whitepaper.pdf#subsection.3.4

needed), so it can be applicable to more substantial volumes, however it is still micro-management of every dispute separately.

This is a clear indication that in order to support real-life trade, with all its nuances, the technological solutions are getting complex. Even though Satoshi's original escrow design is quite elegant, it has severe commercial drawbacks. There is an extensive body of work on escrow protocols from 2017[58] that formalizes many of the security-privacy issues.

The size of security deposits is then the next dilemma, with Alan Reiner anticipating ~20% stakes relative to the transacted value and Oleg Andreev justifying significantly larger deposits, even 200%. Boson Protocol doesn't prescribe deposit amounts and leaves it as customizable parameters, though some guidelines will emerge as the system undergoes adequate simulations.

[58] http://stevengoldfeder.com/papers/escrow.pdf