ORBS

Use of Ethereum as a Base Layer for PoS and PoA Platforms

Oded Wertheim, CTO Oded Noam, Chief Architect

Overview

The concept of a two-layer blockchain came out of the need to extend established blockchain protocols whilst enjoying it as 'trust layer'. While the idea originated from the need to delegate the security of a less-secure network, it is not limited to this use. The concept of creating hybrid systems to enjoy the advantages of other blockchains has many uses, as in many aspects, a hybrid has an inherent advantage. We designed Orbs for applications that make use of multiple blockchain platforms because we believe such design makes more sense for complex, real-world use cases. Orbs itself delegates some of its core functions to Ethereum, not because it lacks in inherent security, but rather because it enables a hybrid system that is more robust than an independent network. In this paper, we would like to highlight some of the advantages and challenges hybrid designs can bring to infrastructure projects.

How PoS and PoA differ and complement PoW

The fundamental properties of current byzantine agreement, proof-of-stake/proof-of-authority DLT networks differ significantly from those of nakamoto-consensus, proof-of-work DLT networks. These differences are varied and many of them cannot be considered purely negative or positive, but should be analyzed as to how they contribute to the properties of the networks that use them. In some cases, combining both types of networks into a hybrid enables us to enjoy the best of both worlds, as presented below.

Finality

Byzantine agreement networks provide near-immediate finality, as opposed to Nakamoto's eventual consensus. Beyond the time to reach consensus, this also entails a dichotomy in how truth is assumed: as a binary state, either true or false in finality-offering systems or as a probabilistic measure—probability of being accepted—in eventual consensus systems.

Fast finality is a huge advantage for applications: it allows low latency in approving transactions, enabling better user experience, low friction in processing complex sequences of transactions, and greatly simplifies client code that can synchronously handle execution results. In general, lack of finality in proof-of-work is a significant obstacle to client code simplicity, which, in turn, reduces the average code quality of applications.

The burden on client code goes beyond the complexity associated with the need to be asynchronous: software logic assumes truth to be binary, and to effectively use logic in client software it needs to convert the probabilistic measure of logic states into a boolean. This puts the burden of assuming finality of the state (does *p>0.9999* mean TRUE?) on the client code, and is vastly important in how responsibility and liability is spread across a decentralized network. When client code converts probabilities to boolean states, it is responsible for its decisions should the unlikely event occur (e.g. when evaluated, we had *p>0.9999* but eventually the state ended up being FALSE). If final states were agreed upon by the network consensus, it was the network nodes (albeit decentralized) that made the call that turned out to be wrong. Obviously, this creates a completely different structure and is expected to have legal, political and financial implications.

The merits of eventual consensus are found in the inherent robustness of the system security: the entire system is not at risk of falling into an incoherent state due to a momentary attack. By definition, systems that offer finality cannot resolve a situation in which there was consensus (even momentarily) over bad data.

Integrity of the Ledger

In Nakamoto consensus, every node validates the integrity of the ledger at any point in time. This means that the protocol rules determine the ledger integrity, and the logic is guaranteed to be consistent with the protocol. The opposite is for the logic to be consistent with the blockchain history: once a block is validated and finalized, it is—by definition—logically true.

Committing to consistency with the protocol comes at a great cost. In principle, it requires all network participants to validate the entire blockchain as they join the network, and re-validate it every time their code changes (else risking an accidental fork: a network partition who's protocol implementation disagrees on the validity of at least one block, and therefore sees an alternative state). The protocol logic has to include deprecated parts that still apply for past blocks, forcing it to grow indefinitely and become more and more complex.

On the other hand, committing to consistency with the ledger contents means accepting finalized content regardless of whether it complies with the protocol. Finality is absolute, regardless of how secure the network that reached it is. This creates an opportunity for attackers to exploit temporary security lapses to create permanent damage. It also makes it more difficult to bootstrap the network's security, as it may be hard to assure the network is secure enough when it is just starting out. In proof-of-work chains, any confidence level can be reached even when the network has low participation. The cost invested in the proof-of-work determines the confidence level in each block, and increases monotonically as further blocks are appended.

Stability of Power Structures

Proof-of-work networks are designed to be permissionless — allowing anyone to participate in the network validation. False validations are absent since the attackers can expect the blockchain consensus to reject invalid blocks, yet the cost of validation will have to be paid whether or not the block is valid. What makes this possible is that these costs are paid as energy and hardware

costs outside of the network's ledger. In other words, the right to participate is granted by spending resources that are extrinsic to the system. Ideally, participation costs should be equal for every participant ("one cpu, one vote"), although variance in energy costs around the globe gets to an order of magnitude, and in many proof-of-work puzzle types specialized hardware can give its exclusive holders an advantage.

In proof-of-stake, the right to participate is determined by ownership of the network's native token. This has a potential to create an unstable positive feedback loop: decline in token value causes a reduction in network security, and vice versa. Such positive feedback loops may amplify temporary variations in network security and, as mentioned above, be used by attackers to exploit temporary security lapses to make permanent damage.

Inherent Advantages of Hybrid Architectures

Separation of Powers

Any implementation of on-chain governance adds an additional form of circular power structure: in processing of on-chain governance procedures, the processors (which may be validators, miners, block producers or other network functions) are in conflict as their result of the process may affect their own status. As a result, processors may try to manipulate these procedures to their benefit. Furthermore, it gives network validators excessive power over the network governance. By taking over the network's governance institutions (such as those that control network connectivity, stake holding registries, protocol upgrades, conflict resolution) and the assets at their disposal (control of legal entities, development funds, reward pools etc), validators may have the power to take over the network in a crypto version of a military coup d'etat. As an example, consider the trivial case of changes to the delegated voting power of a validator in DPoS: validators may delay or avoid processing transactions that record changes to power delegation, as these could delegate power out of their own hands. Such implicit veto can be countered by a hybrid-model creating checks-and-balances, as well as separation of powers.

Such participation of validators in network governance is assumed in many blockchain networks, either by default or due to the power they inherently have over it. Most famously, Bitcoin votes on controversial protocol changes by miner voting. But just like a case of participation of the military in state governance, participation of the validators in setting policy doesn't always reflect the ideals of the network. For example, Orbs positioned itself as a network for large-scale applications, and as such, should be governed mainly by representatives of this group. While the participation of network operators is important to realizing this ideal, it is not their interests that should govern the network.

Hybrid architectures in which governance is delegated to another blockchain offer the ability to have on-chain governance while avoiding distorted power structures.

Aggregate Security and Stability of the Network Security

Independent of the security properties of each of the networks involved, in designing cross-validation methods between networks it is possible to strengthen the security of the hybrid network so that it enjoys the aggregate protection of both networks. For example, the cost of a simple double-spend attack in a proof-of-work network can be the sum costs of double-spend attacks on two networks, if blocks in one network are notarized on the other, and requiring both ledgers to be consistent to accept transfers of assets. Of course, this applies to any network type, and not just proof-of-work.

Aggregate security has additional benefit when there is low correlation between the cost of attack on the various networks, contributing to better stability of level of security the hybrid network enjoys. When fluctuations in cost factors determining to the security of one network have little or no effect on the security of the other network, the probability of opportunistic attacks is significantly reduced. For example: hash power available to two separate proof-of-work networks may be non-correlated or even anticorrelated (in case both employ similar mining rigs); exchange rates of coins used in different proof-of-stake networks may be non-correlated; etc.

Technical Risks of Hybrid Implementation

Liveness and Longevity of the solution

Interdependence between systems creates a “weakest link” chain where if either system falls, service cannot be provided. This, of course, applies also to hybrid blockchain implementations, where one blockchain network (“service network”) depends on another blockchain network for security and/or governance services (“adjunct network”). In the short term, the result of this dependency is that the uptime guarantees of the hybrid system cannot exceed that of any system separately. In the long term, it means that premature decline and eventual termination of the adjunct network will require modification of the service network ahead of time, probably towards replacement of the adjunct network with another system.

Dissonance between opposing sources of truth

Another problem with interdependence between networks is when there are conflicts in data. Such situation will translate to outages in service until the conflicts are resolved. In the case of a fast-finality network that is dependent on an eventual-finality network, such as a PoS/PoA network dependant on a PoW/Nakamoto consensus network, it is possible (though unlikely) to have conflicts in which the divergence occurs several blocks before the point in time when it is discovered. Resolving such conflicts may turn out to be complicated, depending on the protocol proposed to resolve the conflicts.

Ethereum as a Base Layer

Beyond the general and theoretical advantages to hybrid networks, Ethereum is particularly well suited to act as a base layer for such in practice. It enjoys significant, varied use (relative to any other blockchain), and as such, has many independent parties interested in its liveness and security. Having a variety of independent parties that are mostly indifferent to whatever happens on another network, makes it extremely hard for an attacker to orchestrate an attack against the other network. This property is the main benefit of using Ethereum as a base layer in second-layer protocols, for example:

- *Raiden Network* creates a fast payment network that implements the LN protocol on Ethereum, essentially using off-chain hubs to process payments instantly, and falling back to Ethereum for collections.
- *Loom* is a protocol for private or centralized blockchain networks, that notarize their state on Orbs regularly, thus enabling users to verify the network integrity was not violated.

In addition, Ethereum is the center of a high-quality ecosystem that revolves around crypto-assets. This ecosystem includes a multitude of tools and services for securing and trading assets, many of which are essential to high-value economies, and which may require years to develop and gain trust. Without the existence of such ecosystem, the tokenization of voting power in proof-of-stake networks would be cumbersome and risky.

Orbs as a Hybrid Blockchain

Orbs is designed as a hybrid blockchain that uses Ethereum for its token economics and governance functions, with an additional notarization function that can optionally be used as a countermeasure to potential “double-spend” attacks. Unlike “second layer” networks, only mandatory functions that are not mission-critical may depend on processing in Ethereum, reducing the risks of service interruptions due to the inter-dependency.

The Orbs token is a standard ERC-20 token on Ethereum, whose utility includes payments for network fees and participation in the election process of validators. Both functions are implemented as Ethereum smart contracts. By placing the entire token economy on the Ethereum network, Orbs users are enjoying several advantages:

- They enjoy the benefits of the Ethereum token ecosystem, which includes support by the most common and well-established solutions for token security, storage and trade. These include wallet software, custodian services, hardware wallets, exchanges etc.
- Attacks on the Orbs network do not comprise a persistent threat because the attacker cannot take over the token distribution, interfere with validator selection, or force protocol amendments.

Additionally, the fact that the smart contracts that manage network fees (which, in Orbs, are paid for setting up and maintaining virtual chains rather than per-transaction) and validator selection make most protocol upgrades simpler, because the code processing the upgrade is not directly affected by the upgrade.

As a side-benefit of this design choice, apps using Orbs can enjoy the infrastructure built to enable Orbs' own hybrid model, including APIs for smart contracts to read data from Ethereum (in the process, reaching network consensus on the integrity of the data) and to commit transactions to it. Normally, the choice to place the entire token economy on Ethereum would not fit applications whose asset transactions happen at high frequency and may involve millions of users. Still, many will see value in enabling their token economy on both platforms: using the tokens in large scale on Orbs, and tapping in to the tools and services that tokens enjoy in the Ethereum ecosystem. Orbs Atomic Swap Bridge enables any Orbs token to move between the two networks freely, without relying on centralized parties to complete the operation.

The notarization function helps mitigate a security threat common to all consensus protocols that offer immediate finality: a byzantine group can double-sign a block, creating two seemingly valid states of the blockchain and potentially double spend assets. At the cost of a few minutes' delay to finality, applications can require validation of the network state to match the merkle root notarized on Ethereum, eliminating this attack vector.

---

info@orbs.com