ORBS

Blockchain Architecture Considerations to Compete with PaaS/Cloud Services

Oded Noam, Chief Architect, Orbs

oded@orbs.com

Executive Summary

In order for decentralized architectures to make it into mass-market applications, platforms need to be competitive with their centralized alternatives on various key aspects. In working with established product companies that are adding blockchain products to their offering, Orbs and its affiliates have identified a few of these key aspects. In this document we present our findings, along with some architecture designs that may be used to create a blockchain platform that is competitive with centralized platforms. These aspects include cost structures, system performance when scaling up, avoidance of platform lock-ins to reduce technical risks, and completeness of the smart contracts environment.

Overview

In order to create a blockchain platform that can accomodate mass-market applications, one must consider the challenges that these applications need to meet. While decentralized applications may enjoy inherent advantages, such as better alignment with the interests of users, they will never be adopted if the solution itself doesn't stack up to its centralized contemporaries. Simply put, the vast majority of consumers do not care about decentralization. We believe that rather than aiming to be "the best blockchain platform", a platform that wishes to be used by mass-market apps must consider cloud services as their real competition.

Orbs works as a design partner with businesses that provide blockchain related services to mass market users, and collaborates with system integrators and consulting firms that provide mass-market product companies with with blockchain strategy and development services. Our knowledge base is founded on the accumulated experience from working with a variety of companies in these partnerships. By analyzing the daily challenges and issues as well as the raw data, we understand what concerns a business migrating from a centralized cloud based environment to a decentralized blockchain platform.

We identified six constraints that are the core cause for blockchain's failure to win over cloud based clients. These six constraints hamper the success of DApps in key performance areas including operational costs, cost structures, performance, future-proofing applications, and as a comprehensive business solution.

Closing the Gap in Costs and Cost Structures

Decentralized platforms generate high operation costs based on their inherent architecture and mode of integration. A standard decentralized platform creates redundant computing, storage and network traffic leading to higher costs.

Beyond operation costs themselves, we see a threat in the models used to determine blockchain fees. Whereas the prices of cloud services are priced very close to the infrastructure costs (and are thus very stable and tend to only go down over time), blockchain fees fluctuate significantly with demand. Normally, in a standard cloud environment, the marginal revenue per user is decreasing. However, when fees are determined by demand, an app's growth increases demand for platform services, causing the app's per-user costs to rise until the point where they exceed the value created per new user. This means that even an otherwise-successful app would hit a barrier to growth and would not be able to beat out its centralized competitor.

In order for a blockchain platform to be a relevant choice for a DApp that needs to compete with centralized apps, it must provide predictably low operation costs. These costs don't necessarily need to be lower than or equal to those of a centralized infrastructure, but if they are higher they must be proportional to the centralized alternative costs. Otherwise, it will hit a growth barrier at a point where a centralized contemporary will not, and the competitor will likely enjoy the larger network effects necessary to win over a market.

For a decentralized platform to be sustainable, the costs to validators must be economical. If, as shown above, the platform fees need to be proportional to the cost of an alternative centralized infrastructure, the infrastructure costs of validators must also be proportional to that infrastructure cost in that centralized alternative. System architecture, and in particular the consensus model it employs, must ensure an upper bound on the level of redundancy in the parts that are duplicated: processing, storage and network traffic. This requires mostly a constant level of redundancy when the amounts of traffic, data, and users, grow.

Market Design for Sustainable Low Pricing

Curbing the costs that validators pay is not enough: validator profits also need to remain proportional to the costs. To ensure this, the market needs to be designed in such way that increased demand for capacity will be answered with increased supply.

Typically for a normal market, an increase in price of goods results in an increased supply and reduced demand, i.e. at low prices we expect fewer sellers (or lower amounts of goods for sale) and more buyers (or willingness to buy higher amounts), whereas at high prices we expect more sellers (or amounts of goods for sale) and fewers buyers (or lower willingness to buy). This is commonly described as a demand/supply chart as seen in *illustration 1*. The conjunction of the supply and demand curves indicates the prices at which the quantities in demand match those supplied, and the market is cleared (quantity of goods Q^0 is exchanged at price P^0). In such a market, an extrinsic shift in the demand curve, which could occur as a result of more DApps entering the market or a growth in the usage of a single DApp, would create a new market equilibrium in which the market clears with larger quantities being sold

at a higher price (see *illustration 2*, new equilibrium indicated as Q^1 and P^1 respectively). As discussed earlier, such a price increase is undesirable, but actual results in today's blockchain platforms are even worse.

Current blockchain protocols impose limits on block sizes and frequency. This limits the network capacity, so that when the capacity is met further increases in demand cannot be met by an increase in supply ("inelastic supply"). This is indicated in *illustration 3* as a vertical tail of the supply curve. As can be seen in the illustration, the typical behavior of such a market in case of an extrinsic shift of the demand is a significant spike in prices and no increase in quantity. It is easy to understand that if validators are unable to increase quantities, the only way to reach equilibrium (pairing of the supply and demand) is by increasing prices until fewer buyers wish to transact.

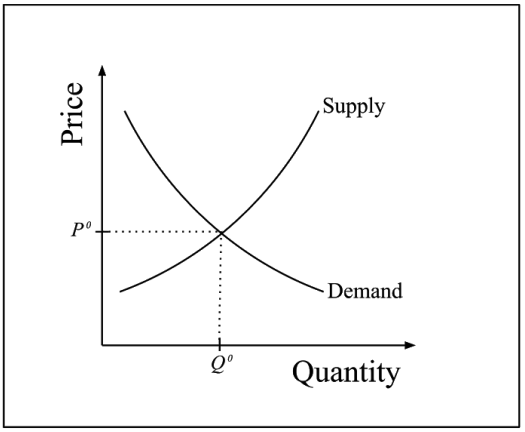

Illustration 1: Normal market behavior

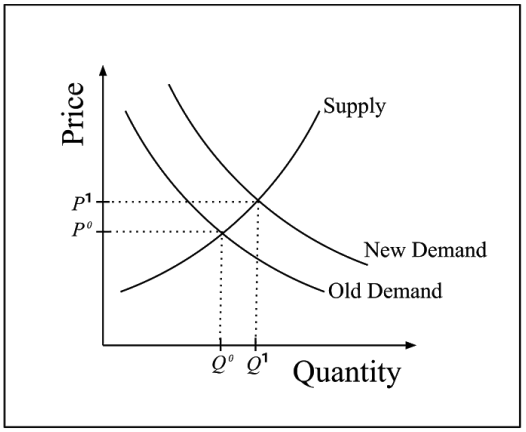

Illustration 2: Normal market, extrinsic increase in demand

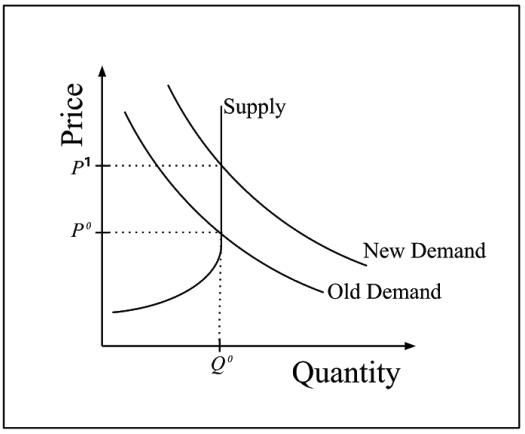

Illustration 3: Inelastic supply, extrinsic increase in demand

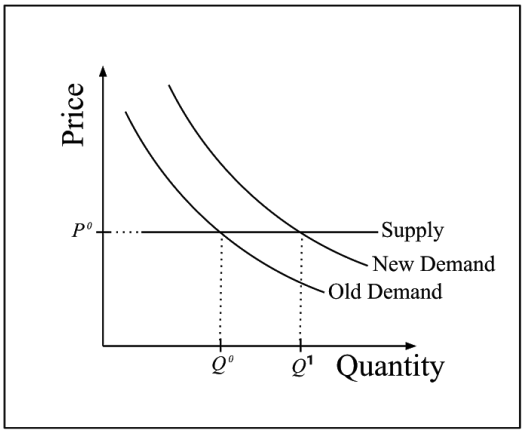

Illustration 4: Perfectly elastic supply, extrinsic increase in demand

The marketplace for cloud services, with providers like AWS, GCS and Azure, demonstrates a contrary effect: the supply side appears to be detached from the prices ("perfect elasticity"). This is caused in part by the overall scale of the systems being many orders of magnitude larger than any single app using them, which makes the effect of every user's choice

negligible compared to the overall quantities, and in part by the high level of portability users enjoy, enabling them to easily transition between competing cloud services creating a state of near-perfect competition. As seen in illustration 4, in such a market an extrinsic shift in demand would result in an increase in quantities to match the new demand, without change in prices.

A decentralized platform that aims to compete with cloud services must ensure that similarly to cloud services, increased demand would yield increased quantities at similar prices. For validators, that means any request for additional capacity in a virtual chain should be met with an equal increase in the system capacity. As previously demonstrated on the normal model, if a node operator believes her choice of quantities to supply affect market prices, she is likely to meet increases in demand with a mix of increase in supply and a price raise. However, if the operator expects that her choices do not affect market prices, she will choose to increase prices to match demand. Practically, when faced with a request for additional capacity, she believes that asking for higher prices will not win her the deal and the buyer will end up taking the service from other suppliers. To increase her total revenues, her only option is to offer the requested capacity without changing prices.

System Architecture used in Orbs to Provide Sustainable Low Pricing

Blockchain platforms are at in infancy, so we cannot expect excess capacity to keep prices at a constant level. We can, however, design the protocol in such a way that nodes compete on providing additional resources, and design the system in a way that maximizes the flexibility of the system when increasing or decreasing its scale.

Orbs contains multiple “virtual chains”, with each serving as an independent blockchain processed in isolation from the others, all using a randomized-proof-of-stake (rPOS) consensus model. In rPOS, every node may be selected to participate in the processing of every block at random. Nodes remain busy most of the time because all virtual chains are processed in parallel, so when a node is not selected to participate in generating a block of one virtual chain, it may be selected to participate in other virtual chains’ blocks. The random selection is adjusted (weighted) to the amount of resources contributed by each node, so even if different nodes contribute different amount of resources, all resources remain utilized equally. This mechanism assures asymmetric distribution of traffic is feasible and efficient.

Additionally, the system was designed for horizontal scaling within each node, making it easy for nodes to change their capacity almost immediately. It is reasonable to assume most nodes run on top of a cloud platform. For those, the costs of additional capacity remain predictably low and operators can easily configure automatic scaling making them extremely dynamic with the capacity they are providing.

The combination of these architectural properties of Orbs enables the platform to open capacity changes to bidding. When a virtual chain is set up, the required capacity will be added to the network by the lowest bidder(s) in an auction on providing that additional capacity. Note that the virtual blockchain remains decentralized, as the winners of that

auction are not directly executing that virtual chain but rather contribute the required amount of resources to a pool shared with all other virtual chains.

Closing the Gap in Performance

Blockchain performance is lower than that of its centralized alternative, and unfortunately degrades as apps scale up. The problem with current-generation blockchain platforms is that they rely on consensus models that have an inherent conflict between security and performance: their security relies on a large number of consensus participants, and their performance decreases with any increase in the number of participants.

In consensus models that create a performance-security dilemma, the only choice mass-market apps have is to choose platforms that limit their level of decentralization in order to achieve growth. Methods such as delegated voting and proof-of-authority are capable of providing performance proportional to that can be achieved when using centralized services. For example, EOS (a blockchain platform employing a consensus concept called Delegated Proof of Stake) limits its set of active verifiers to just 21, as opposed to several tens of thousands active verifiers in Ethereum. Despite that, limited decentralization does create a security risk, which is exacerbated when the value of assets processed on the network increases. Compromises that are reasonable for small apps may not be valid for processing contracts handling millions of dollars every second.

There are two prominent approaches to replacing this paradigm, and enabling networks to scale without compromising neither security or performance: off-chain processing and randomized consensus.

Under the umbrella of off-chain processing solutions we see a range varying from completely centralized state channels that only occasionally settle accounts on the ledger (lightning network being the most prominent example of such), to types of sharding or sharding-like scaled blockchain solutions that compensate for potential compromises on security by allowing arbitration on a secure, proof-of-work base-layer platform in case suspicious activity (such as Plasma[1], Lightning[2], Truebit[3], and others).

Randomized Consensus Protocols take a different approach. These protocols eliminate the performance-security dilemma by maintaining a fixed-size consensus committee. This fixed size remains constant irrespective of the size of the platform, thereby reversing the diametric model into a model that sustains the performance of the decentralized platform even as it grows. However, this model requires the development of newer cryptographic techniques, and this obviously leads to new engineering challenges.

Deeper analysis of randomized consensus protocols is available in our paper [link].

[1] http://plasma.io/plasma.pdf
[2] https://lightning.network/lightning-network-paper.pdf
[3] https://people.cs.uchicago.edu/~teutsch/papers/truebit.pdf

Future-Proofing and Avoiding Lock-Ins

As a general rule in system architecture, dependence on a single platform provider is viewed as a business risk. It also poses a technical risk in fields that are undergoing rapid evolution and may change significantly and quickly. Blockchain platforms inflate this risk even further as their evolution could end in hard forks that may risk the underlying security platforms gain as they accumulate users.

Looking at centralized cloud platforms, the fundamental services they offer are generally considered stable and dependable, a low-risk choice even when there is a dependency on a single provider. Longevity and availability of the larger services is widely regarded as an axiom. Basic services like virtual machine hosting and storage are considered commodities and it is easy to exchange between the different providers. Legacy PaaS (such as Google App Engine, Azure Stack or Heroku) have open-source alternatives that users can deploy on their own, making portability possible, albeit at a premium. Interconnectivity enables an app migrating between platforms to have a transition phase in which services operate on two platforms concurrently.

Blockchain migration is considerably more difficult. Transition of DApps between platforms will require, in addition to data migration, the ability to transfer crypto-assets between the platforms maintaining the guarantees of the smart contract that generated the assets. However, such transfers are not always supported by the platforms and the smart contracts. A blockchain platform (“source” platform) that wants to minimize the lock-in risks of its users can employ several means, each contributing to reducing the theoretical cost of migrating out of the platform (to a “destination” platform):

- Provide infrastructure for moving crypto-assets between platforms. It is preferable to require two-way transfers as they enable full flexibility for hybrid operation and rolling back during the transition phase.
- Provide infrastructure for interconnectivity with other blockchain platforms. Ideally, such infrastructure should enable both source network to read from the destination network, and vice versa. The former is best provided by enabling a smart contract in the source platform to directly read states from the destination platform, which is the most thorough solution for smart contracts on the source network to refer to the universal/multi-chain state. For the latter, it is best not to assume the destination platform will be able to read data from the source network, and generation of proofs of state on the source network, so providing a facility for issuing proofs for state of data on the source platform, that can be verified by a generic smart contract (low assumptions as to the capabilities of the destination network).
- The platform itself should be forkable (patent-free, and subject to a permissive open-source license).

Complete solution to smart contracts

Even when the end user thinks an app needs to do just one thing, in order to do it well the app may need to incorporate a wide variety of features that are quite more advanced. These range from data handling and analysis, AI, data collection from external sources and more. Cloud services make use of their Internet connectivity to integrate third-party services (SaaS) and employ 3rd-party tools inside their cloud deployments (using facilities such as AWS Marketplace[4]).

Current generation smart-contract platforms cannot integrate with such services. Mostly, the execution paradigm for smart contracts is that of closed-system execution, in which no external connectivity can be relied upon and execution results must be deterministic. Also, the use of external SaaS is problematic because due to their open nature, smart contracts are unable to make use of secret data (such as retaining cryptographic keys) and therefore cannot authenticate with third party services. This is a severe limitation on the ability of developers aiming to create “Pure-DApps”, which are applications that have no parts that are operating as centralized backends. Developers are thereby forced to adopt hybrid models in which DApps have centralized parts and decentralized ones. Hybrid models differ from the Pure-DApp model not only in system architecture but also in the form of business organizations it requires to operate: The centralized backend parts will have to be operated by centralized, usually for-profit entities.

This calls for reviewing the principles of the closed-system paradigm. Blockchain systems that are based on variants of the Nakamoto consensus only approximate final agreement over the inclusion of a certain block in the ledger. The expectation is for every past block to be re-verified by any future verifier joining the network. The ability to re-validate any block at any time in the future can only be assured in closed systems.

The “eventual consensus” model is being replaced by a “finality” model: an agreement, at a certain point in time, as to the state of the ledger disallowing any future changes to the blocks up to that point. This shift encompasses a compromise on the security of the ledger, but clearly has benefits in engineering and business aspects. Prominent blockchain platforms today, including future revisions of Ethereum, provide finality of the ledger. In platforms that provide finality, the closed-system principle is no longer mandatory. Measuring its benefits, we believe they are outweighed by the benefits of an open system, which can call to external systems and arrive at a final consensus as to the results of such call. This requires, beyond the ability to connect to external systems, the ability to perform non-deterministic processing and reach final agreement as to the majority results of such execution.

Having support for non-deterministic processing opens the door to another type of operation. If each node can access a private key share that it stores securely, the decentralized network can perform threshold-cryptographic operations, such as generating digital signatures securely by a smart contract. Among other things, this could enable a decentralized system to authenticate and interact with SaaS and 3rd-party systems.

[4] https://aws.amazon.com/marketplace

Being Practical

We believe ability to compete with cloud services is a strategic necessity for blockchain platforms to be used by mass-market apps. Yet, some of the barriers to effective competition are significant because the solutions may be partial, primitive, or hard to implement.

The entire field is evolving rapidly, and new technologies may emerge in the next few years that offer better ways to overcome these barriers. Adoption, on the other hand, is just as gradual -- if today most DApps are mere proof-of-concept, the entirety of mass-market applications are not yet behind the corner. We expect to see early attempts of mass-market DApps to be in non-critical functionality of established apps, and in parallel, entirely new DApps aimed at early adopters.

Working hand-in-hand with the developers of these DApps allows us to gradually improve the ability of the platform we are developing; Orbs, to compete with cloud services.

Copyright © Orbs Ltd., 2018-2019

info@orbs.com