Rethinking Service Systems: A Path Towards Secure and Equitable Resource Markets

Ghada Almashaqbeh

NuCypher
ghada@nucypher.com

Abstract. Escalating demand for digital services has motivated building non-traditional solutions to reduce costs and achieve a greater level of flexibility. Among these solutions, peer-assisted models are gaining more popularity in replacing infrastructure-based (possibly) centralized approaches. Cryptocurrencies have strengthened this trend by providing a fully distributed mechanism to compensate service provision, and their blockchains and consensus protocols offer a trustless and publicly verifiable way to govern the system. We present a generic framework for building distributed service markets by utilizing these new technologies. We also discuss the security and efficiency challenges, with a focus on how the economic considerations impact system design and threat mitigation, along with some potential solutions.

1 Introduction

When obtaining digital services, usually we deal with traditional systems that are centrally managed. Most of the time, we resort to third party providers, e.g., commercial companies, to obtain services like file storage, content distribution, computation outsourcing, and many others. Despite being effective and widely deployed, this centrally-managed paradigm introduces several trust, cost, and transparency issues. It requires establishing complex business relationships with these companies, where customers usually overprovision their needs in order to handle future peak demands [25,28]. Also, it constrains the customers with the service specifications these companies can offer, such as geographic coverage and service speed, not to mention the limited visibility into the real status of the system regarding its performance and the available amount of resources.

These issues motivated the community to revisit the old ideas of peer-to-peer (P2P) based models in which anyone is allowed to join the system and serve others. In order to encourage collaborative work and compliance with the protocol, payments are provided in return; this creates a market for trading resources. This paradigm builds flexible systems, scales more easily with demand, and extends the network coverage since peers from anywhere can join. Furthermore, this paradigm builds transparent and equitable ecosystems in which participants can negotiate service terms and price directly instead of having a few entities monopolizing the market.

Table 1: Examples of centrally-managed digital services and their counterparts of P2P-based ones.

Service Type	**Traditional Solution**	**P2P-based Solution**
Payments	Banks	Bitcoin
File storage	Dropbox [9]	Filecoin [11]
Content distribution	Akamai [1]	CacheCash [18]
Key management system	Azure Key Vault [14]	NuCypher [15]

Although monetary-incentivized P2P-based systems are an old idea, especially for content sharing, most of existing solutions introduce some form of centralization or trust. They either rely on centralized payment services, place trust in specific parties to handle these payments and resolve disputes, or even rely on some centralized entities to manage participants and defend against some security threats [25,26,34]. Such design choices bring us back to the central management model and the trust issues of traditional solutions.

The evolution of cryptocurrencies and blockchain technology [27,33] has provided templates for reshaping large-scale distributed systems and services. Cryptocurrencies implement a decentralized virtual currency exchange medium that permits participants to be rewarded without any pre-authentication or identification requirements. And their underlying blockchains and consensus protocols support public verifiability, auditing, and decentralized governance without needing to place trust in any entity. These features can be exploited in P2P-based schemes to manage and pay for the correct service without driving the system toward centralization (see Table 1 for examples of traditional service solutions and their P2P counterparts).

However, the open access environment of P2P networks (i.e., allowing anyone to join and dealing with untrusted participants) introduces several security and performance challenges that need to be addressed before having any practical deployment. In addition, having monetary incentives motivates attackers to attack the system in novel ways to maximize their financial profits. Thus, traditional practices of secure systems design need to be modified and expanded to account for such factors.

To address these issues, we present a generic framework for designing secure, scalable, and equitable resource markets to provide services in a fully distributed way. It consist of systematized design steps distilled from experiences in building blockchain-based services and large-scale distributed systems. The proposed framework accounts for the security, performance, and economic aspects of monetary-incentivized decentralized systems. It also highlights how such an emerging work model requires more sophisticated techniques (for risk management, threat mitigation, service-payment exchange, service pricing, etc.) than

these employed by traditional, infrastructure-based services. We discuss these challenges along with some potential solutions.

2 Distributed Resource Markets Design

One effective idea for building fully decentralized and equitable services is to build distributed markets to trade resources. That is, implement a protocol that allows anyone to join to serve others and collect payments in return. Such an approach needs to solve several challenges introduced by the open access and decentralized work environment. In other words, dealing with untrusted, possibly financially motivated participants requires deploying additional measures that may impact efficiency, usability, and compatibility with existing infrastructures. Consequently, there is a need for carefully-tailored threat mitigation techniques and efficiency optimization mechanisms in order to promote the adoption of these systems.

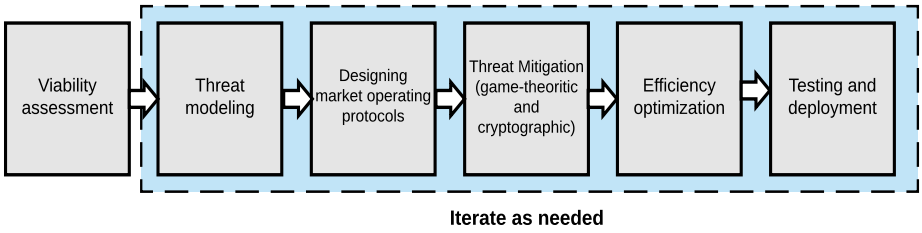

Fig. 1: Design process of distributed resource markets.

In this section, we discuss the main steps and challenging issues that need to be considered when designing a distributed service system. These steps are captured at a high level in Figure 1, which we discuss in greater detail in the following paragraphs.

Viability Assessment. Before looking into building a distributed resource market, one has to assess its viability. This includes studying the demand side (who is interested in the service) and the supply side (who can provide it) to answer several questions; are there tangible advantages to encourage replacing traditional solutions with fully distributed ones? Can the system match the reliability and performance offered by these traditional solutions? Does providing the service require large amounts of resources that exceed the capabilities of average end-users? Such a viability study is an important step to assess the potential for practical adoption before investing time and effort into building the system.

Threat Modeling. Despite the many advantages they offer — decentralization, transparency, and lowered service costs — there is still a big gap between the promise of P2P-based systems and their performance in practice. Adding

monetary incentives, by using another P2P-based payment service, i.e., cryptocurrencies, widens this gap. This is due to the perception that these systems are not secure, where the recent large number of security breaches give credence to these doubts [2,3,4,5,6,7,17,13,12,16,10,8].

The best practice for designing a secure system requires a threat modeling step to investigate potential security risks. Such a model can guide designers in deploying the proper countermeasures, and evaluating the security level of a system. For resource markets, building a threat model requires a framework that can handle large-scale distributed systems, explicitly account for the financial motivations of the attackers, and help in spotting any potential collusion or Sybil attacks.

This observation encouraged the community to visit old threat modeling frameworks and adapt them to address these issues. For example, the ABC framework [19] was designed to achieve these goals by accounting for both the underlying cryptocurrency medium and the service provided on top of it. ABC enables building comprehensive threat models by holistically analyzing the threat space while managing its complexity, and distilling the impactful cases that need to be neutralized to secure the system. It also allows for classifying threats based on their mitigation techniques, i.e., threats that can be addressed cryptographically or algorithmically, and these that require game theoretic means, thus providing insights about the proper measures to deploy.

It should be noted that the threat modeling step need to be revisited each time the system design is altered. Furthermore, it should be performed as the last step before shipping the system to argue formally about its security.

Unique Aspects of Operating a Distributed Market. The open access work environment helps to create flexible services and transparent ecosystems. However, this comes at a cost. Dealing with untrusted parties means that fair exchange is impossible [23,29], which raises the question of when to pay service-providers - before or after providing the service? If paid first, a malicious service-provider may not serve the customer, and if served first, a malicious customer may not pay afterwards.

Furthermore, accounting attacks, in which participants collude with each other, pretending that the service has been delivered, could be a hammer that destroys the market. This is a particular problem in systems that require sponsoring service requests. For example, in content distribution, a publisher (e.g., Netflix) can hire caches to distribute content to its clients, and hence, it pays for the service. In this case, caches (or servers in general) and clients may collude so that clients pretend to be served, allowing servers to collect payments from the sponsor for free.

The above security issues (an many others depending on the service type) require a careful design of a decentralized service-payment exchange protocol that can reduce the risks of dealing with untrusted, possibly colluding parties. Such a protocol represents the backbone of the resource market; if it fails the whole market fails. Servers will not be willing to participate if they are not being paid.

The same is true for customers; they will not be willing to use the system if they pay for a service that they do not receive. Operating the market also requires devising mechanisms for service pricing, term negotiation for server recruiting, and matching protocols to match these servers with interested customers.

Financial and Cryptographic Security Measures. Usually, security threats are mitigated by using cryptographic means (e.g., encryption and digital signatures), or algorithmic approaches (e.g., ordering the actions in a way that enforces secure behavior). Monetary-incentivized systems introduce new types of attacks that cannot be addressed using conventional approaches. In particular, having financially-motivated attackers introduces new threat vectors that need to be mitigated by using financial means. These fall into three categories. First, detect-and-punish mechanisms, where any participant is required to lock a collateral that is forfeited if they are caught cheating. Second, designing algorithms that, if performed maliciously, require larger amounts of resources than when performed honestly. And third, designing service pricing and payments mechanisms that make it more profitable on the long run to act honestly in every service request than cheating or ignoring the request (even if cheating or ignoring are not detectable). Such techniques make cheating unprofitable so that rational parties will choose to adhere to the protocol.

For example, to reduce the risks of the impossibility of fair exchange, micropayments can be employed. That is, instead of paying a large chunk of money for the full service, the payment is divided into small values, each of which is exchanged for a small service amount. For instance, one can pay for retrieving a file in small data chunks instead of paying for the full retrieval all at once. Hence, a server loses a small payment if a client does not pay after receiving a chunk. Similarly, a client loses a small payment if it pays in advance and the service-provider does not send a data chunk in return.

On the other hand, to thwart accounting attacks, system designers need to incorporate suitable techniques to prove or confirm resource expenditure, and consequently, confirm that payments are well deserved. In online content delivery, for example, the CAPnet puzzle [21] can be used to ensure that caches have delivered the requested content. The design of this puzzle follows the second category mentioned above, where solving the puzzle without doing the work is more expensive (resources-wise) than solving it after retrieving the content. In file storage, proof-of-replication [24] can be used to prove that a server is still storing the clients' files with the agreed-upon number of replicas. Here, a detect-and-punish mechanism is used, where failure to provide a correct proof leads to slashing part of the deposit a server pledged when joining the system [11]. Thus, the type of the provided service directly influences the delivery confirmation mechanism that need to be deployed.

Optimize for Efficiency. Although designing a secure system is the ultimate goal, efficiency is an important driving factor of practical adoption and deployment. System designers need to exploit any opportunity that allows for optimizing performance. This also involves choosing the right trade-off between

security and efficiency in the sense of risk management. That is, threats that have high impact need to be prioritized over low impact ones. Moreover, looking into alternative cryptographic primitives that are lightweight, or optimize their implementation, while maintaining the required security guarantees is another effective avenue to utilize.

Furthermore, reducing interaction between participants is beneficial. It speeds up the service and promotes the system's scalability. This can take the form of batching requests/replies between customers and service-providers to reduce costs and optimize resource allocation, in addition to batching or aggregating record verification on the blockchain.

Another important aspect is related to handling micropayments. Micropayments create a scalability problem as they produce a huge number of transactions that overwhelm the system and require large processing fees. Here, probabilistic schemes are useful in aggregating the small transactions into few larger ones before processing [32,31]. In particular, payments take the form of lottery tickets, and only winning tickets are processed in the system with values that compensate properly for the tickets exchanged so far. Several fully-distributed micropayment schemes exist in the literature [30,22,20] that provide trade-offs between efficiency, anonymity, and security guarantees.

Testing and Deployment. To examine the viability of the system, conventional practices of prototyping, benchmarking, and controlled deployment can be used to evaluate both efficiency and resistance to attacks. These provide a starting point to attract early adopters and test the system at a large scale. This stage may inspire designers to revisit specific parts of the system for further optimization based on the results of the conducted experiments, or feedback from the community based on a testnet deployment for example. Testing and improving also continue beyond the testing stage, i.e., after public launch, but deploying protocol modifications could be harder especially if they result in hard forks or community division.

3 Conclusion

While centrally-managed systems provide reliable services, they introduce trust, cost, and transparency issues. This has motivated developing non-traditional peer-assisted models to create distributed resource markets that are open to anyone to join and serve others while collecting payments in return. However, such a paradigm comes with security and performance challenges that need to be addressed before having any practical deployment. Our work puts forward a generic framework for designing efficient and secure resource markets. It also examines in greater depth how financial incentives affect security measures and system design choices.

References

1. Akamai. `https://www.akamai.com/`.

2. Benebit – the biggest ico exit scam in history nets up to $4 million. https://www.coinbureau.com/ico/benebit-biggest-ico-exit-scam-history-nets-4-million/.
3. Binance cryptocurrency sell-off disaster blamed on mass phishing campaign. https://www.zdnet.com/article/binance-cryptocurrency-sell-off-disaster-blamed-on-mass-phishing-campaign/.
4. Bitcoin cash exploit cripples network during scheduled hardfork upgrade. https://cryptoslate.com/bitcoin-cash-exploit-cripples-network-hardfork/.
5. Bitcoin gold suffers double spend attacks, $17.5 million lost. https://www.zdnet.com/article/bitcoin-gold-hit-with-double-spend-attacks-18-million-lost/.
6. The bitfinex bitcoin hack: What we know (and don't know). https://www.coindesk.com/bitfinex-bitcoin-hack-know-dont-know/.
7. Bitfloor hacked, $250,000 missing. https://bitcoinmagazine.com/articles/bitfloor-hacked-250000-missing-1346821046/.
8. Coffeeminer hijacks public wi-fi users' browsing sessions to mine cryptocurrency. https://www.zdnet.com/article/how-to-hack-public-wi-fi-to-mine-for-cryptocurrency/.
9. Dropbox. https://www.dropbox.com/.
10. Enigma's hack: $500,000 of ether stolen, accounts compromised. https://cointelegraph.com/news/enigmas-hack-500000-of-ether-stolen-accounts-compromised.
11. Filecoin. https://filecoin.io/.
12. Hackers hijack another ethereum ico, small number of users affected. https://www.bleepingcomputer.com/news/security/hackers-hijack-another-ethereum-ico-small-number-of-users-affected/.
13. Hackers stole $32 million in ethereum. https://thehackernews.com/2017/07/ethereum-cryptocurrency-hacking.html.
14. Microsoft azure key vault. https://azure.microsoft.com/en-us/services/key-vault/.
15. Nucypher. https://www.nucypher.com/.
16. One of the world's biggest bitcoin exchanges has been hacked. http://www.businessinsider.com/south-korean-bitcoin-exchange-bithumb-hacked-ethereum-2017-7.
17. South korean cryptocurrency exchange hack sees $40m in altcoin stolen. https://www.zdnet.com/article/south-korean-cryptocurrency-exchange-hack-sees-40m-in-altcoin-stolen/.
18. Almashaqbeh, G. *CacheCash: A Cryptocurrency-based Decentralized Content Delivery Network*. PhD thesis, Columbia University, 2019.
19. Almashaqbeh, G., Bishop, A., and Cappos, J. Abc: A threat modeling framework for cryptocurrencies. In *IEEE INFOCOM Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock)* (2019).
20. Almashaqbeh, G., Bishop, A., and Cappos, J. Microcash: Practical concurrent processing of micropayments. In *Financial Cryptography and Data Security* (2020).
21. Almashaqbeh, G., Kelley, K., Bishop, A., and Cappos, J. Capnet: A defense against cache accounting attacks on content distribution networks. In *2019 IEEE Conference on Communications and Network Security (CNS)* (2019), IEEE, pp. 250–258.
22. Chiesa, A., Green, M., Liu, J., Miao, P., Miers, I., and Mishra, P. Decentralized anonymous micropayments. In *Annual International Conference on the*

Theory and Applications of Cryptographic Techniques (2017), Springer, pp. 609–642.

23. Even, S., and Yacobi, Y. Relations among public key signature systems. Tech. rep., Computer Science Department, Technion, 1980.
24. Fisch, B. Tight proofs of space and replication. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2019), Springer, pp. 324–348.
25. Kassa, D. F., and Nahrstedt, K. Hincent: Quick content distribution with priorities and high incentives. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE* (2013), IEEE, pp. 22–30.
26. Nair, S. K., Zentveld, E., Crispo, B., and Tanenbaum, A. S. Floodgate: A micropayment incentivized p2p content delivery network. In *Computer Communications and Networks, 2008. ICCCN'08. Proceedings of 17th International Conference on* (2008), IEEE, pp. 1–7.
27. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system.
28. Nicolae, B., Riteau, P., and Keahey, K. Bursting the cloud data bubble: Towards transparent storage elasticity in iaas clouds. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium* (2014), IEEE, pp. 135–144.
29. Pagnia, H., and Gärtner, F. C. On the impossibility of fair exchange without a trusted third party. Tech. rep., Technical Report TUD-BS-1999-02, Darmstadt University of Technology ..., 1999.
30. Pass, R., and Shelat, A. Micropayments for decentralized currencies. In *CCS* (2015), ACM, pp. 207–218.
31. Rivest, R. L. Electronic lottery tickets as micropayments. In *International Conference on Financial Cryptography* (1997), Springer, pp. 307–314.
32. Wheeler, D. Transactions using bets. In *International Workshop on Security Protocols* (1996), Springer, pp. 89–92.
33. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper* (2014).
34. Zhang, K., Antonopoulos, N., and Mahmood, Z. A review of incentive mechanism in peer-to-peer systems. In *2009 First International Conference on Advances in P2P Systems* (2009), IEEE, pp. 45–50.