Blockchain or not Blockchain?
Costs and benefits of large-scale synchronization

Goals: Design protocols for scalable and consistent data sharing in models with mutual distrust

Tools: Logic, algorithmic reasoning, programming, system design

Prerequisites: Basic knowledge of distributed algorithms, concurrent programming skills, curiosity and persistence

Technical Skills: Fluent with Linux and OpenSource stacks, DIY Spirit

State of the art: consensus & blockchain

Blockchain can be viewed as a mechanism to implement *trustworthy* access to shared data in systems with mutual distrust. System participants may have conflicting interests and might even be willing to cheat, but the implementation is expected to ensure that they share data and exchange assets in a consistent, available and fair way. The TrustShare project is devoted to the algorithmic basics of blockchains, in order to discover novel efficient data-sharing and asset-transfer implementations.

Any software system, be it a multiprocessor application or a large-scale distributed service, involves manipulation of shared data by a collection of *users*. Depending on the application requirements, the data can be accessed for simple *reading* and *writing*, or with more sophisticated *conditional* operations that may combine reading data units and, in case certain conditions on the read values are met, updating them. In a distributed system prone faults and security attacks, we cannot rely on a trusted "central party" that can store and process data. Instead, users of the systems maintain *replicas* of the shared data and involve a *synchronization* protocol to keep the replicas up to date.

The prominent *blockchain* technology aims at implementing a public "ledger": a decentralized consistent history of transactions proposed by an *open* set of participating processes, with no static membership. This problem can be seen as an instance of fault-tolerant *state-machine replication* [26], prominent examples of which are the *crash-tolerant* Paxos protocol by Lamport [21] and the BFT (*Byzantine* fault-tolerant) system by Castro and Liskov [4]. These systems use instances of *consensus* protocols in order to ensure that users get consistent views of the system evolution.

Principal downside of classical consensus protocols are lack of scalability and the need for a fixed or properly reconfigurable set of participants out of which only a bounded fraction (up to one third) can be faulty. This can be hard to ensure in an *open* (*permissionless*) system, where an arbitrary fraction of participants can be controlled by the adversary [9]. Prominent blockchain protocols [25,29] achieve (nondeterministic) consistency by assuming that (1) the system is synchronous, (2) participants can use asymmetric cryptography, and (3) the adversary can control

at most a minority (in practice, a minor fraction) of computing power. These protocols are, however, notoriously expensive and slow. Even protocols that obviate the energy demands via using *proof-of-stake* [1,6,18], *proof-of-space* [10], or *proof of space-time* [24]. However, the proposals still resort to synchronous networks and/or impose restrictions on the fraction of honest players to ensure proper security levels. An immediate question is whether these costs and assumptions are unavoidable.

One may argue that in many practical settings, the set of participants is well under control, and *private* (or *permissioned*) case, when the set of participants is well under control, we can resort to private solutions [5]. These solutions can tolerate periods of network asynchrony and exhibit way better performance than permissionless ones.

Even more, one might also want to reconsider the very problem. "Heavy-weight", strongly consistent solution are required to implement a consensus-based total order on events in the distributed system. But is total order always necessary? In many practical applications, a weaker form of consistency, based the notion of *causality*, can be enough. This is the case, e.g., for *asset transfer* systems [8, 14]. In particular efficient and *responsive* implementations of an asset transfer system can be achieved. (Recall that consensus and, thus, has no responsive fault-tolerant implementations. [12, 17].)

Objectives

There are multiple ways in which the existing solutions for asset transfer and more general problems can be improved.

One can extend classical solutions designed for static systems with globally known trust assumptions to the more general context using *active reconfiguration*. To anticipate security attacks and get rid of compromised system components, the system may explicitly reconfigure itself. As has been recently shown [19,20], reconfiguration can be implemented in an asynchronous and transparent way, so that the users of the system get consistent service even though the components are periodically reconfigured.

Furthermore, instead of anticipating failures and proactively investing into fault-tolerance, accountability [15] can be seen as a way to *react* to failures by detecting them and reconfiguring the system by replacing faulty components. Generic accountable services can be seen as *Byzantine fault detectors* [16] and, due to their generality, incur considerable costs. For specific system specifications, accountability may be easier to achieve, as suggested by some recent work on Byzantine consensus [7]. A promising direction is to apply the approach to systems providing weaker semantics where consensus is replaced with *lattice agreement* [11].

Finally, the very idea of trust assumptions was recently explored in completely new way. All this started as cryptocurrency systems [23,27] proposed to encompass users who do not necessarily hold the same assumptions of who to trust. Indeed, conventional data-replication services are based on *quorums* [22, 28], subsets of system participants matching two important conditions: in every run of the system, *every* two quorums should have at least one benign participant in common and *some* quorum should only contain benign participant. It is assumed that the participants share the global knowledge about the quorum system. A few recent articles [3, 13, 30] proposed alternative (and sometimes contradicting each other) formalizations of *federated* or *decentralized* quorum systems and explored their power in implementing Byzantine reliable broadcast [2] and solving consensus. Under the *decentralized trust* assumptions, a rich variety of important distributed abstractions can be implemented and we expect these solutions to be, though less consistent, but more efficient than their classical counterparts.

The goals of the project are:

- Devise reconfigurable and accountable storage systems.
- Extend the solutions to the decentralized trust model.
- Test the resulting implementations against existing permissioned and permissionless solutions.

This project is intended to be a first step towards a CIFRE (Industrial Agreement of Training through Research) doctoral thesis. A CIFRE fellow undertakes her/his research within a partnership between Mazars, R&D and Télécom Paris. The result of the work is a PhD thesis. The fellow is jointly supervised by both the monitor in the company and the academic thesis advisor.

Why Mazars R&D? Mission Briefing

Mazars is a global audit, accounting and consulting group employing more than 40,000 professionals in 89 countries through member firms.[1] Our core business, financial audit, has known little evolution over the last 30 years. It relies on a selective orientation of tasks taking into account the specificity of each client, associated with sampling methods. These methods rely only marginally on the power of IT/Computing Technology.

Mazars' Ambition

Mazars' Ambition for Audit is to be the avant-garde of the transformation of financial audit by creating "Augmented Audit". For consulting Mazars is empowering consulting with BigData architectures, IA and DevOps.

Mazars will rely extensively on internally developed IT solutions to monitor all transactions, analyze and/or rejouer all our client data. High end techniques will be used to identify anomalies and atypical behaviour. Audit represents a multi tens-of-billions of dollars market worldwide and our belief is that this strategic project will generate a key competitive advantage. Thus it has received consequential investments and is at the heart of our organization's focus.

Mazars' R&D

Despite our exponential growth, our values are still Autonomy, Initiative and Team Challenges at small and large scale. We are adamant that each individual can make a difference. The Mazars R&D reflects our Start-Up spirit, built on the knowledge gathered from Start-Ups acquired by Mazars in areas from API oriented software architecture to AI and Big Data.

You will join the R&D team, one of the Lab tenants, to participate in one of our most strategic projects. The Mazars R&D team is in charge of developing and producing the complete set of tools required to revolutionize audit. These tools cover a vast field of applications: GED, Electronic Signatures, Robotisation, OCR, SmartContract, Datalakes, Systems Experts, NLP.

You will work directly with the head of Mazars R&D with the Research Chair dedicated to Blockchains for the Finance industry as a playground.

[1] `https://www.mazars.com/Home/About-us/Mazars-at-a-glance`

Contact

Luis Belmar Letelier, PhD, Partner at Mazars
@: luis.belmar-letelier@mazars.fr
Phone: +33149976330 +33666802291
https://www.mazars.fr/
61 rue Henri Regnault, tour EXALTIS
92075 PARIS LA DEFENSE

Petr Kuznetsov
http://www.infres.enst.fr/~kuznetso/
petr.kuznetsov@telecom-paristech.fr
INFRES, Télécom Paris, Institut Polytechnique Paris
19 place Marguerite Perey, F-91120 Palaiseau, FRANCE
Office: 4D55

References

[1] I. Bentov, A. Gabizon, and A. Mizrahi. Cryptocurrencies without proof of work. In *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, pages 142–157, 2016.

[2] G. Bracha and S. Toueg. Asynchronous Consensus and Broadcast Protocols. *JACM*, 32(4), 1985.

[3] C. Cachin and B. Tackmann. Asymmetric distributed trust. In *OPODIS*, 2019.

[4] M. Castro and B. Liskov. Practical byzantine fault tolerance. In *OSDI: Symposium on Operating Systems Design and Implementation*. USENIX Association, Co-sponsored by IEEE TCOS and ACM SIGOPS, Feb. 1999.

[5] M. Castro and B. Liskov. Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, Nov. 2002.

[6] J. Chen and S. Micali. Algorand: A secure and efficient distributed ledger. *Theor. Comput. Sci.*, 777:155–183, 2019.

[7] P. Civit, S. Gilbert, and V. Gramoli. Polygraph: Accountable byzantine agreement. *IACR Cryptol. ePrint Arch.*, 2019:587, 2019.

[8] D. Collins, R. Guerraoui, J. Komatovic, P. Kuznetsov, M. Monti, M. Pavlovic, Y. A. Pignolet, D. Seredinschi, A. Tonkikh, and A. Xygkis. Online payments by merely broadcasting messages. In *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2020, Valencia, Spain, June 29 - July 2, 2020*, pages 26–38. IEEE, 2020.

[9] J. R. Douceur. The sybil attack. In *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers*, pages 251–260, 2002.

[10] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak. Proofs of space. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 585–605, 2015.

[11] J. M. Falerio, S. K. Rajamani, K. Rajan, G. Ramalingam, and K. Vaswani. Generalized lattice agreement. In D. Kowalski and A. Panconesi, editors, *ACM Symposium on Principles of Distributed Computing, PODC '12, Funchal, Madeira, Portugal, July 16-18, 2012*, pages 125–134. ACM, 2012.

[12] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, Apr. 1985.

[13] Á. García-Pérez and A. Gotsman. Federated byzantine quorum systems. In *22nd International Conference on Principles of Distributed Systems, OPODIS 2018, December 17-19, 2018, Hong Kong, China*, pages 17:1–17:16, 2018.

[14] R. Guerraoui, P. Kuznetsov, M. Monti, M. Pavlovic, and D. Seredinschi. The consensus number of a cryptocurrency. In P. Robinson and F. Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 307–316, 2019.

[15] A. Haeberlen, P. Kouznetsov, and P. Druschel. Peerreview: practical accountability for distributed systems. In T. C. Bressoud and M. F. Kaashoek, editors, *Proceedings of the 21st ACM Symposium on Operating Systems Principles 2007, SOSP 2007, Stevenson, Washington, USA, October 14-17, 2007*, pages 175–188. ACM, 2007.

[16] A. Haeberlen and P. Kuznetsov. The fault detection problem. In T. F. Abdelzaher, M. Raynal, and N. Santoro, editors, *Principles of Distributed Systems, 13th International Conference, OPODIS 2009, Nîmes, France, December 15-18, 2009. Proceedings*, volume 5923 of *Lecture Notes in Computer Science*, pages 99–114. Springer, 2009.

[17] M. Herlihy. Wait-free synchronization. *ACM Trans. Program. Lang. Syst.*, 13(1):123–149, 1991.

[18] A. Kiayias, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 357–388, 2017.

[19] P. Kuznetsov, T. Rieutord, and S. Tucci Piergiovanni. Reconfigurable lattice agreement and applications. In P. Felber, R. Friedman, S. Gilbert, and A. Miller, editors, *23rd International Conference on Principles of Distributed Systems, OPODIS 2019, December 17-19, 2019, Neuchâtel, Switzerland*, volume 153 of *LIPIcs*, pages 31:1–31:17, 2019.

[20] P. Kuznetsov and A. Tonkikh. Asynchronous reconfiguration with byzantine failures. In H. Attiya, editor, *34th International Symposium on Distributed Computing, DISC 2020, October 12-16, 2020, Virtual Conference*, volume 179 of *LIPIcs*, pages 27:1–27:17, 2020.

[21] L. Lamport. The Part-Time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998.

[22] D. Malkhi and M. Reiter. Byzantine quorum systems. *Distributed Computing*, 11?(?):203–213, 1998.

[23] D. Mazieres. The stellar consensus protocol: A federated model for internet-level consensus, 2016.

[24] T. Moran and I. Orlov. Proofs of space-time and rational proofs of storage. *IACR Cryptology ePrint Archive*, 2016:35, 2016.

[25] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[26] F. B. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. 22(4):299–319, Dec. 1990.

[27] D. Schwartz, N. Youngs, and A. Britto. The ripple protocol consensus algorithm, 2018.

[28] M. Vukolic. The origin of quorum systems. *Bulletin of the EATCS*, 101:125–147, 2010.

[29] G. Wood. Ethereum: A secure decentralized generalized transaction ledger. White paper, 2015.

[30] Álvaro García-Pérez and M. A. Schett. Deconstructing stellar consensus. In *OPODIS*, 2019.