# Dynamic Parallelism in Consortium Blockchains

**Goals:** Define and design a blockchain protocol enabling parallel processing of conflict-free transactions

**Tools:** Logic, algorithmic reasoning, programming

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

## Summary

The prominent *blockchain* technology implements a public "ledger": a decentralized consistent history of transactions proposed by an open set of participating processes. This problem can be seen as an instance of fault-tolerant *state-machine replication* [7], prominent examples of which are the *crash-tolerant* Paxos protocol by Lamport [4] and the PBFT system by Castro and Liskov [3] tolerating arbitrary (*Byzantine*) faults. These systems use instances of *consensus* protocols in order to ensure that users get consistent views of the system evolution.

Downslides of classical consensus protocols are lack of scalability and the need for a fixed or properly reconfigurable set of participants out of which The original blockchain protocol [6] achieving nondeterministic consistency in an *open* (so called permissionless) system via the difficulty of the imposed *proof of work*. Proof-of-work is notoriously slow and energy-demanding, so more and more attention is paid to *consortium* blockchains [2] which maintain an open set of participants that can propose transactions, but only a controlled set of *consensus nodes* are to agree on the order in which the transactions (also known as *smart contracts*) are applied.

Maintaining a total order on transactions may not be necessary: intuitively, nonconflicting transactions may be accepted in parallel without consensus. The goal of this project is to explore to which extent this can be done, for example, employing ideas developed in the distributed computing community (see, e.g., [1]) for the crash-tolerant settings [5]. An immediate difficulty here is to bound the effect malicious Byzantine nodes may have on the system state. The expected outcome of the project is a prototype of a consortium blockchain providing dynamic parallelism to nonconflicting transactions.

## Contact

Prof. Petr Kuznetsov
`http://www.infres.enst.fr/~kuznetso/`
`petr.kuznetsov@telecom-paristech.fr`
INFRES, Télécom ParisTech
Office C213-2, 46 Rue Barrault

## References


[1] PODC 2016 workshop on Distributed Cryptocurrencies and Consensus Ledgers, July 2016. https://www.zurich.ibm.com/dccl/.

[2] V. Buterin. On public and private blockchains, August 2015. https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/.

[3] 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.

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

[5] L. Lamport. Generalized consensus and paxos. Technical report, March 2005. https://www.microsoft.com/en-us/research/publication/generalized-consensus-and-paxos/.

[6] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, May 2009. https://bitcoin.org/bitcoin.pdf.

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