On Asynchronous TEE-Based State Machine Replication

Marc Leinweber, Hannes Hartenstein
marc.leinweber@kit.edu, hannes.hartenstein@kit.edu
Karlsruhe Institute of Technology (KIT)

For decades, *asynchronous* state machine replication (SMR) was flying more or less under the radar. Instead, previous research focused more on partially synchronous SMR and showed that Trusted Execution Environments (TEEs) can effectively be used to prevent equivocation and, thereby, to increase performance and fault tolerance [Ver+13]. Recently, asynchronous SMR has come to the focus of system researchers (e.g., [Mil+16]) promising increased performance, robustness, and simplicity in implementation. In contrast to partial synchrony, TEEs cannot be combined in a rather straightforward fashion with the asynchronous model as non-equivocation does not help in reaching stricter variants of the validity property as needed, e.g., for the Binary Agreement primitive used in HoneyBadgerBFT [Mil+16]. Based on the seminal DAG-Rider result [Kei+21], we designed and proved *TEE-Rider*, the first USIG-based [Ver+13] asynchronous Atomic Broadcast algorithm [LH23]. In this talk we explore some challenges and sketch solutions to bring TEE-Rider from research to practice: a meaningful operating model and client behavior, the setup phase, a TEE-based common coin, and crash recovery. We argue that TEEs give us the possibility to explore an operating model being stronger than crash and omission while not actually being maliciously Byzantine, as significant trust in the correctness of the TEE at use is needed. In this operating model, it seems to be superfluous for clients to send a transaction request to $\lfloor \frac{n}{2} \rfloor + 1$ replicas. In fact, when focusing on preventing accidental and unintended faults in addition to crashes, e.g., human errors, instead of malicious behavior, we can exploit asynchronous, especially DAG-based, protocols to scale the transaction throughput in the number of replicas. Most common coin primitives are based on rather expensive threshold cryptography. We investigate ways to provide a dealerless, TEE-based and asynchronous common coin primitive. Additionally, a truly asynchronous setup phase seems to be more academic than useful: if all parties participating in the distributed system genuinely agreed on operating a replica, a quick and efficient setup should be the goal. Thus, we will present our idea for a synchronous setup phase including the setup of the common coin primitive. Finally, long-running SMR-based applications will eventually reach the limit of $\lfloor \frac{n-1}{2} \rfloor$ faulty replicas which is somewhat ignored for TEE-based approaches in literature so far. In contrast to the recovery of, e.g. PBFT-based applications, the TEE and asynchrony makes recovery a harder issue. A wrong approach could lead to equivocation, spoiling the whole endeavor. Our thoughts and findings will be supported by empirical evidence and live explorations using our *ABCperf* framework [Spa+23].

[Kei+21] I. Keidar et al. "All You Need is DAG". In: *PODC '21: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, July 26-30, 2021*. ACM, 2021, pp. 165–175. DOI: 10.1145/3465084.3467905.

[LH23] M. Leinweber and H. Hartenstein. "Brief Announcement: Let It TEE: Asynchronous Byzantine Atomic Broadcast with n $\geq$ 2f+1". In: *37th International Symposium on Distributed Computing, DISC 2023, October 10-12, 2023, L'Aquila, Italy*. Vol. 281. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 43:1–43:7. DOI: 10.4230/LIPIcs.DISC.2023.43.

[Mil+16] A. Miller et al. "The Honey Badger of BFT Protocols". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. ACM, 2016, pp. 31–42. DOI: 10.1145/2976749.2978399.

[Spa+23] T. Spannagel et al. "ABCperf: Performance Evaluation of Fault Tolerant State Machine Replication Made Simple: Demo Abstract". In: *Proceedings of the 24th International Middleware Conference Demos, Posters and Doctoral Symposium, Bologna, Italy, December 11-15, 2023*. ACM, 2023, pp. 35–36. DOI: 10.1145/3626564.3629101.

[Ver+13] G. S. Veronese et al. "Efficient Byzantine Fault-Tolerance". In: *IEEE Trans. Computers* 62.1 (2013), pp. 16–30. DOI: 10.1109/TC.2011.221.