Towards a Blockchain-based Data Recorder for Small Drones

Harald Böhm[1], Tobias Distler[1], and Rüdiger Kapitza[1]

Abstract: Building a reliable and tamper-resistant flight data recorder for small drones is a task that comes with unprecedented requirements in terms of robustness, resource efficiency and weight. In this paper, we report on our ongoing efforts to solve the associated problems by developing a solution that records all relevant information (e.g., positional data) in an on-board permissioned blockchain. The proposed system, through physical replication, aims to achieve data persistence despite catastrophic events such as a drone crash. All recorded data is replicated across three nodes and kept consistent by a Byzantine fault-tolerant (BFT) protocol. Leveraging the specific properties of our hardware setting, this protocol does not implement full-fledged BFT consensus, but instead relies on a tailored reliable-broadcast mechanism optimized for memory efficiency.

1 Introduction

Storing the history of significant parameters such as the location of an aircraft, positions of actuators, and timestamps of events, flight data recorders often times enable a detailed post-incident analysis, and hence play an important role in improving air travel safety; both for passengers inside the airplane as well as for people on the ground. Unfortunately, the large size and high weight of existing airplane flight data recorders makes it impossible to use them for most unmanned aerial vehicles, especially the rapidly increasing number of small drones that are only able to carry a few hundred grams of payload. As a consequence, the development of flight data recorders for small drones makes it necessary to pursue new avenues that are taking the particular characteristics of these vehicles into account.

Following the general idea of ZugChain [Rü22], a recently proposed data-recording system for trains, our solution relies on a blockchain to store the collected data in a tamper-resistant manner. However, our approach differs from ZugChain in three main aspects: (1) While ZugChain employs a full-fledged BFT consensus algorithm to keep the blockchain replicas consistent, our flight data recorder ensures the same based on a less complex reliable-broadcast mechanism. (2) ZugChain comprises $3f + 1$ replicas to tolerate f faults, whereas our solution only requires $2f + 1$ replicas due to exploiting the specifics of our hardware platform. (3) Finally, unlike ZugChain, which uses heavy and powerful on-board units, the proposed solution is based on lightweight ESP32-C3 microcontrollers. In combination, these three aspects significantly increase resource efficiency.

2 System Architecture

As illustrated in Figure 1, the system architecture and basic workflow of our approach are as follows. The flight controller, a central component running on its own chip, is responsible for stabilizing the drone and executing flight manoeuvres. In addition, the flight controller collects real-time positional data and other relevant flight information, and hence represents

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany
This work was supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant number 20E2122B (BALu) and the German Research Foundation (DFG) – 554710377 (BFTeam).

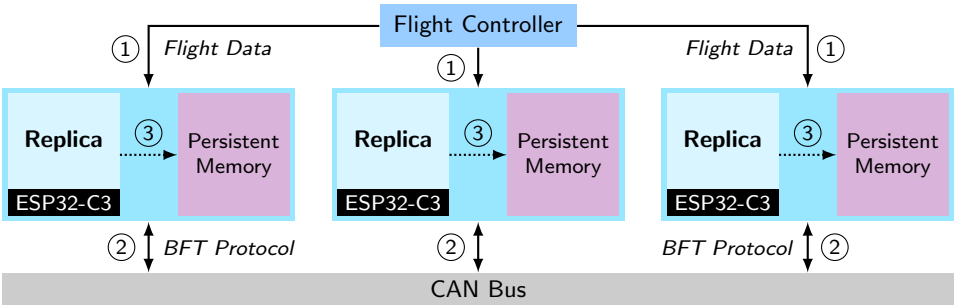

Fig. 1: System architecture

the source of all data to be recorded. To log a dataset, (1) the flight controller forwards the dataset to a group of three replicas via asynchronous serial communication; each of these replicas runs on a dedicated ESP32-C3 microcontroller and is equipped with a separate persistent-memory module. In a next step, the replicas combine one or more datasets into a new block and (2) execute a BFT protocol (over a shared CAN bus) to reliably and consistently append the block to the blockchain. In contrast to traditional BFT system architectures [Di21], the three nodes in this case act as both clients and replicas of the BFT protocol. Once a block is committed, (3) a replica writes the block to persistent memory.

3 Design Considerations

With the flight controller being the only data source, and the sequential transmission of datasets to the replicas being enforced by the hardware, the BFT protocol does not need to perform consensus to establish an order [RH21]. Instead, replicas can simply maintain the order in which the datasets arrive from the flight controller, and then use a modified reliable-broadcast mechanism to achieve resilience against data corruption (e.g., caused by bit flips) and arbitrary replica behavior (e.g., as the result of hardware failures). For this reason, the main design challenges pertain to (1) tailoring the BFT protocol to the guarantees provided by the CAN bus that connects the three replicas (e.g., reliable communication, synchrony), and (2) developing a memory-efficient implementation [BDW24, BD24] that accounts for the fact that each ESP32-C3 has only 400 KB of SRAM at its disposal.

Bibliography

[BD24] Böhm, Harald; Distler, Tobias: Memory-Efficient Byzantine Fault-Tolerant Replication for Highly Resource-Constrained Systems. In: ROBUST '24. 2024.

[BDW24] Böhm, Harald; Distler, Tobias; Wägemann, Peter: TinyBFT: Byzantine Fault-Tolerant Replication for Highly Resource-Constrained Embedded Systems. In: RTAS '24. 2024.

[Di21] Distler, Tobias: Byzantine Fault-Tolerant State-Machine Replication from a Systems Perspective. ACM Computing Surveys, 54(1), 2021.

[RH21] Roth, Edo; Haeberlen, Andreas: Do Not Overpay for Fault Tolerance! In: RTAS '21. 2021.

[Rü22] Rüsch, Signe; Bleeke, Kai; Messadi, Ines; Schmidt, Stefan; Krampf, Andreas; Olze, Katharina; Stahnke, Susanne; Schmid, Robert; Pirl, Lukas; Kittel, Roland; Polze, Andreas; Franz, Marquart; Müller, Matthias; Jehl, Leander; Kapitza, Rüdiger: ZugChain: Blockchain-based Juridical Data Recording in Railway Systems. In: DSN '22. 2022.