Technische Universität Braunschweig

Institut für Betriebssysteme und Rechnerverbund

SEP Distributed Systems

Data Logging on Trains Using Blockchains

Ines Messadi

April 24, 2020

Table of Contents

Organisation

- Our weekly meetings start from next week
  - You present the current status
  - Progress and any issues or questions? Get feedback, Next steps..
- SELPstorganisation
  - We are your customers, you are the developers
  - You organize yourselves in the group
  - Each member must contribute content to all project phases, **equally!**
  - One programs the other writes text unfortunately **does not work!**
- You all can meet on **Discord** channel, SEP has its own channel
- More tools on:
  `https://sep.isf.cs.tu-bs.de/redmine/projects/betreuer-informationen/wiki`

Organization (2)

- Mailing List: `sepds@ibr.cs.tu-bs.de`
- Support:
  - **Ines Messadi**
    messadi@ibr.cs.tu-bs.de
  - Email or we agree on a time to meet on Discord
- Documents:
  - **Kai Bleeke**
    bleeke@ibr.cs.tu-bs.de
- More information in isf redmine:

  `https://sep.isf.cs.tu-bs.de/redmine/projects/20-ibr_ds_1`

Organization

- Document submission
  - in LaTeX, template from ISF
  - Kai and me will review it:
    1. Upload it to svn
    2. Send document to the mailing list when ready for review
  - ISF
    - **Only** in ReadMe
    - **Everyone** is reponsible
  - Upload **only** in SVN
    No Email, Dropbox, GoogleDrive, USB Sticks, Github, ...

First document:The offer (Das Angebot)

- Summary of the topic as you understood it
- Including project flow, general conditions, guidelines, project organization.
- Delivery on 30.04.2020, at the latest 23:59:59 CEST with us by mail.
- Questions about documents can be asked at any time

Dokumente - zeitlicher Ablauf

ISF	intern	Dokument
06.05	**30.04.**	Angebot
27.05	**19.05.**	Pflichtenheft & Abnahmetestspezifikation (Specification and test specification)
10.06	**01.06.**	Fachentwurf (Design)
01.07	**22.06.**	Technischer Entwurf (Technical Design)
15.07	**06.07.**	Testdokumenation
23.07		TDSE ?

- ISF deadlines are always on **Wednesdays**
- Internal deadline one week before on Monday
- Our feedback till **Friday**
- No intermediate presentation with ISF!
- No respect of any deadline would automatically lead to failure

Questions about organization?

Table of Contents

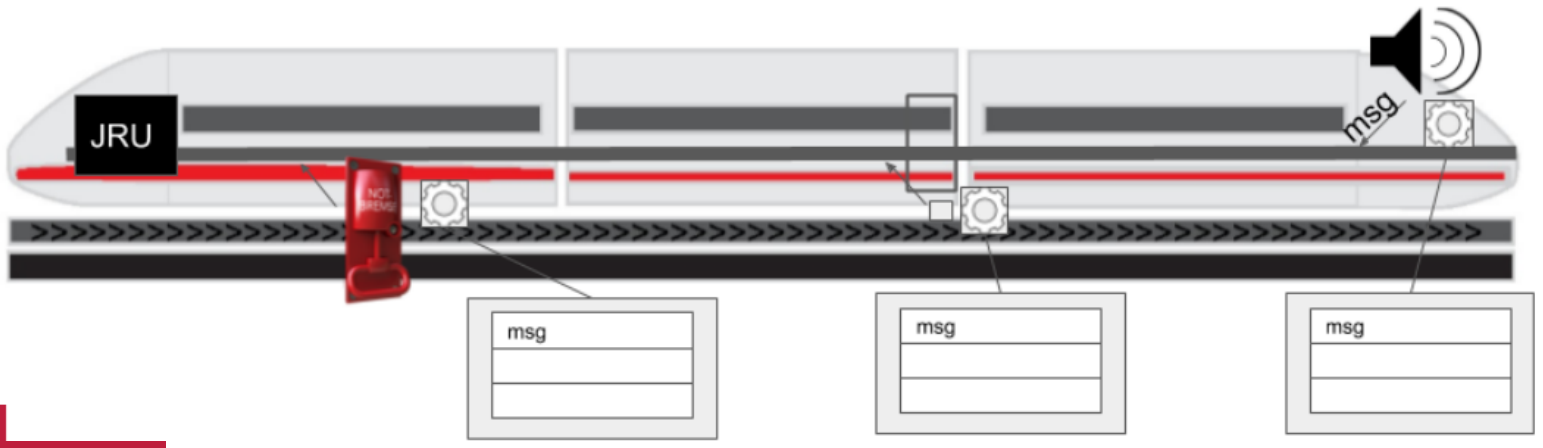
Data Logging on Trains Using Blockchain

- Train needs to be monitored for repairs, accidents
- **Event Recorder**: Central device resilient to crash
- Record the specified data (Door signals, Brake controls..)
  → **Functional**
- Resist tampering, prevent misuse
  → **Secure**
- Reported accident cases [3]
  - Data manipulation
  - Damage of the recorder

Why Blockchain?

- We need a decentralized solution
- Data integrity and availability
  → Decentralized and secure train components logging
- **Log and monitor status changes**
  - Speed
  - Diagnostic messages (Train stops, Time)

  → Recorded and stored reliably in the Blockchain



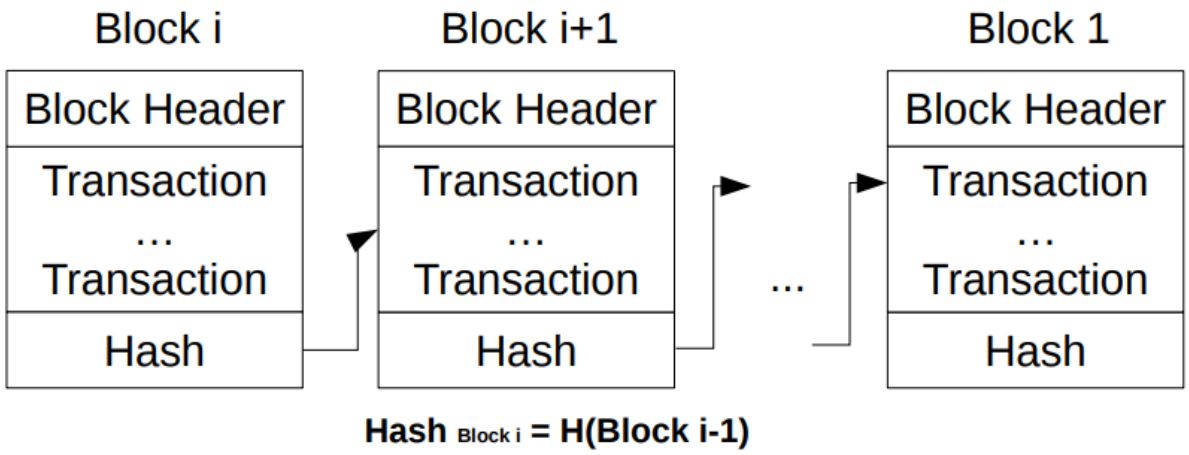
Why Blockchain?

- We need a decentralized solution
- Data integrity and availability
  → Decentralized and secure train components logging
- **Log and monitor status changes**
  - Speed
  - Diagnostic messages (Train stops, Time)

  → Recorded and stored reliably in the Blockchain

Blockchain Basics

- Transactions are stored decentralized and tamper-proof
- Each block contains hash from the previous
- **Participants** owns a copy of the blockchain and verifies transactions across the network.
- **Consensus**: Agreeing to the transactions on the blockchain.



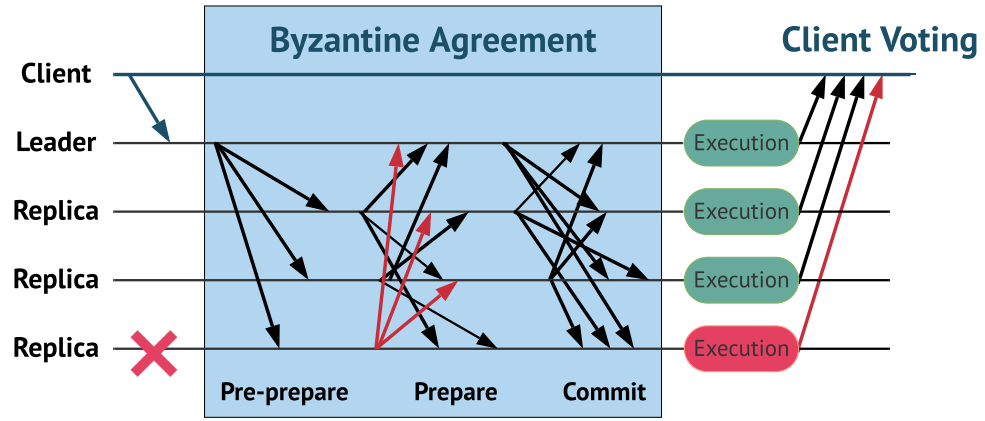
Blockchain and Cryptocurrenties

- **Proof-of-Work**
  A security feature in blockchain to prevent attackers from easily taking over the blockchain.
  → Slow and energy consumption issues

- **Byzantine fault tolerant (BFT) protocols**
  Consensus even with participants showing arbitrarily wrong behaviour Suitable for block creation

Practical Byzantine Fault Tolerance

- Replication algorithm that tolerates **Byzantine faults** (More details in the papers: [1]and [2])
- Malicious attacks and software errors can cause nodes to lie
- $3f + 1$ nodes reach consensus on order of requests
- Consensus even with malicious participants
- If the leader fails $\rightarrow$ **View change algorithm**
- In our case, we will use a **Rust framework** implementation
  - Safety & performance
  - Stable, documented, Continuous integration and Integration tests

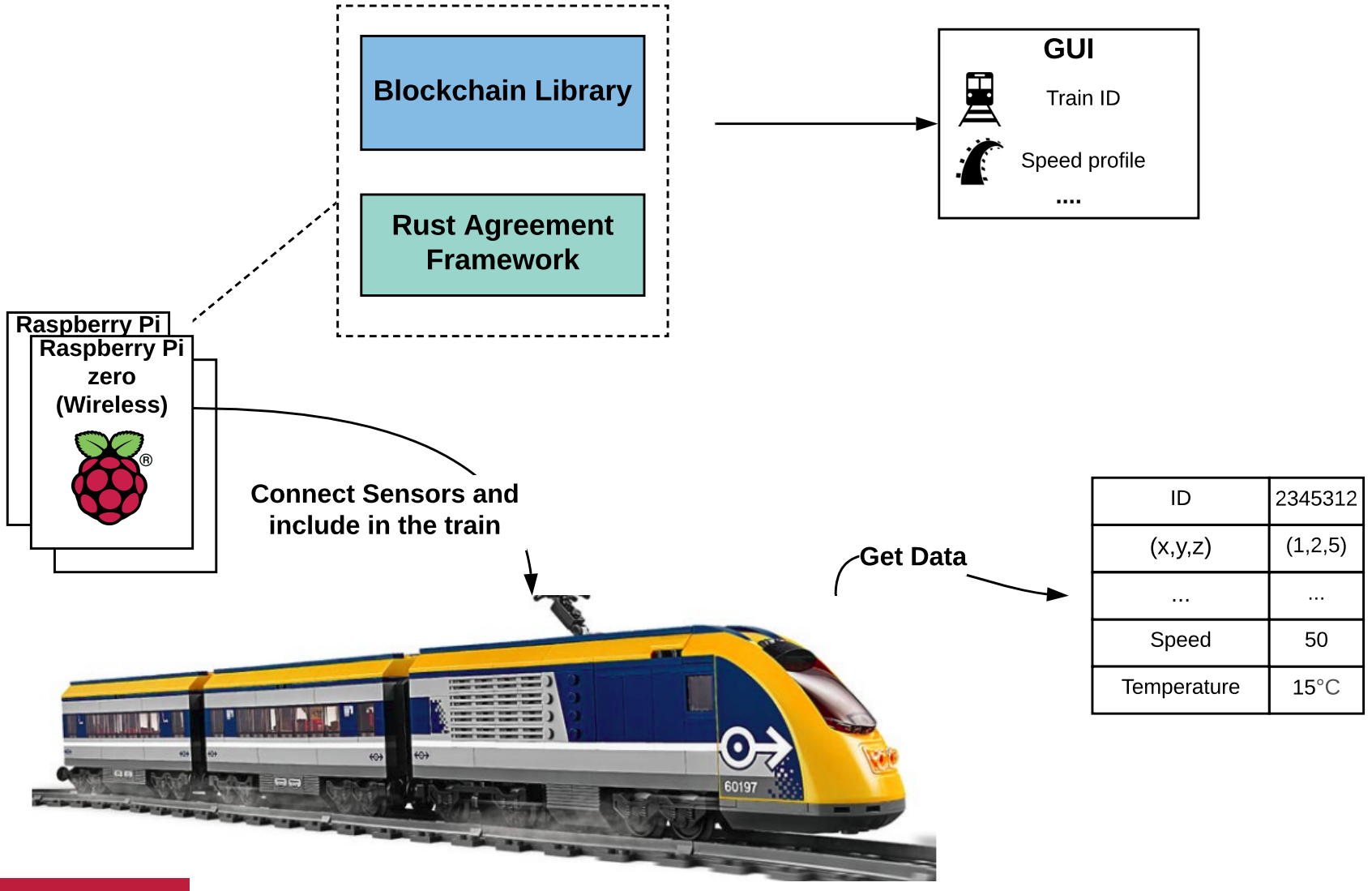


Project Idea

- **General idea**
  - Implement a Blockchain on a model moving train
  - Integration with our Rust-based consensus
  - Offer data as a cloud-based web interface for maintenance(GUI for the blockchain)

- **Minimum data to be collected**
  - Train Speed
  - Time
  - Distance
  - Positions
  - Temperature

Proof of Concept



ID	2345312
(x,y,z)	(1,2,5)
...	...
Speed	50
Temperature	15°C

Hardware

- Two Model Trains & its materials
- Sensors hats
- Distance senors in front of the train to detect collision/couple carriage
- 8 Raspberry Pi Zero
- Related materials: power banks, cables

Next Steps

- Preparation of the offer till next Thursday
  - What is the problem?
  - How do we solve it?
  - How is the plan to organize this? (e.g Gantt chart)
- Subdivide to three groups
- Two groups get the hardware
  - Configuration, installation of Raspberry Pi
  - Record data, implement a program that displays the data
  - Connect it as a BFT request
- Third group
  - Getting familiar with the rust framework
  - Blockchain implementation
  - Web interface implementation

Questions about anything?

References


Miguel Castro, Barbara Liskov u. a. „Practical Byzantine fault tolerance“. In: OSDI. Bd. 99. 1999. 1999, S. 173–186.


Leslie Lamport, Robert Shostak und Marshall Pease. „The Byzantine generals problem“. In: Concurrency: the Works of Leslie Lamport. 2019, S. 203–226.


FREIGHT TRAINS. „RAILROAD ACCIDENT REPORT“. In: (1972).