Institute for Data Engineering
Programming Exercise

Project Topic – Blockchain-based Sensor Data Integration

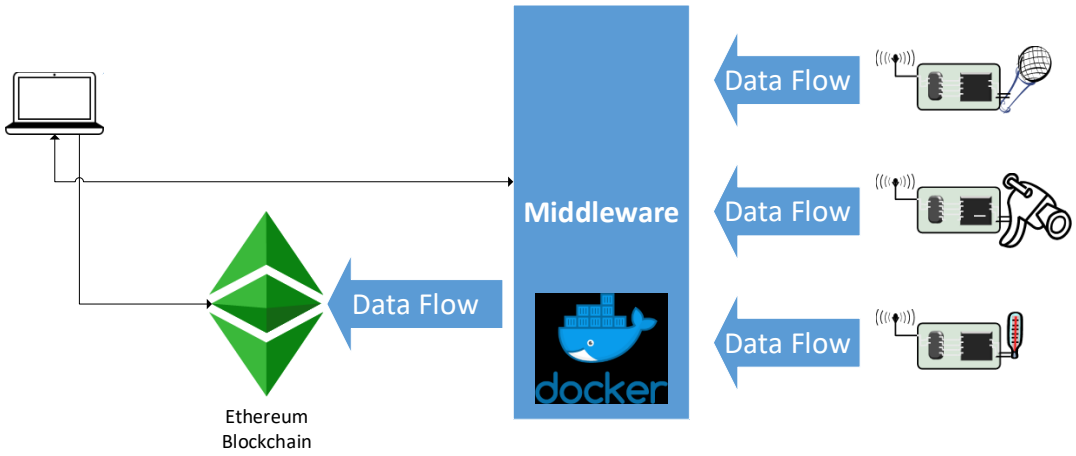

Figure 1: Example Scenario

Introduction

In Internet of Things (IoT) scenarios, e.g., smart factories or smart grids, huge amounts of data may be generated [1]. While not all of this data is interesting, it is important to identify relevant events and make them available for users. Once events have been identified, it is necessary to distribute them to end users using a trusted channel. Especially in distributed scenarios, where different organizations might be involved, it is necessary to send and store this data in an immutable way.

In the world of cryptocurrencies like Bitcoin, a similar problem arises, since transactions need to be stored in a permanent and unchangeable way [5]. For this, blockchains are applied. Within this project, it is the goal to make use of this basic idea and to apply blockchain technology in order to store IoT events [3, 4] and distribute these events to users.

Assume a basic setup as exemplified in Figure 1. As it can be seen, the data flow is from the devices to a Docker-based middleware, which filters the data, and only adds filtered data to the blockchain. The user (indicated by the laptop in the figure) does not access the single sensors. Instead, the user interface is only able to visualize data from the blockchain. In addition, the user can interact with the middleware in order to set parameters for filtering the data before it is forwarded to the blockchain. For instance, if the value from a temperature sensor falls below or exceeds a predefined parameter, this data item will be added to the blockchain, while all other data items are ignored.

As blockchain technology, Ethereum[1] is chosen. Like other second generation blockchains, Ethereum allows to implement *smart contracts*. In the project, a simple smart contract has to be implemented which pushes a notification to the user in case a critical event has been added to the blockchain. However, the notification does not contain the actual data: The user has to get this data from the blockchain.

[1] `https://www.ethereum.org/`

As dataset, you have to make use of the 2017 release of the *UMass Smart* Dataset*[2] [2]. You may choose any dataset from the *Apartment dataset* or *Home dataset* for your project. The dataset can be used to emulate the generation of data items over time.

General Remarks

1. Make sure to use a repository for your code (e.g., GitHub, GitLab, ...)
2. Please do not produce boilerplate code.
3. Please do not reinvent the wheel. Familiarize yourself with popular frameworks and libraries to solve the given tasks.
   a) Smart Contracts (Truffle, HardHat, Brownie, ...)
   b) Communication (REST, Message Brokers, gRPC, ...)
   c) Middleware (Web Frameworks depending on which programming language you want to use)
   d) Frontend (React, Angular, Vue, ...)
4. Familiarize yourself with best-practices for full-stack development

Stage 1

In the first stage, the focus should be on setting up the infrastructure.

Tasks:

1. Start by designing the architecture of the middleware.
2. Familiarize yourself with Ethereum and how to use it. It is sufficient to set up a local Ethereum blockchain (e.g., Ganache[3]).
3. Build a *sensor data provider*, which emits data from the UMass Smart* Dataset over time. The sensor data provider emulates the generation of data items. It is advisable to make the submission speed configurable to apply different submission speeds during testing and actual performance tests of the system.
4. Integrate first data items into the blockchain. Prefiltering is not required at this stage.

Stage 2

In Stage 2, the solution from Stage 1 needs to be extended by more sophisticated functionalities. Most importantly, the user interface and data filtering functionalities need to be integrated.

Tasks:

1. Design a Web-based user interface for the end user. Through this user interface, data items from the blockchain will be provided to the user. Also, the user gets notifications (in case new data items have been added to the blockchain) through this user interface. Last but not the least, the user should be able to set the parameters for data filtering in the user interface.

[2] `http://traces.cs.umass.edu/index.php/Smart/Smart`
[3] `https://www.trufflesuite.com/ganache`

2. Realize filtering functionalities, i.e., parameters which are used as thresholds for adding data to the blockchain.

3. Implement the smart contract which notifies the user about new data items in the blockchain.

References

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. *Computer Networks*, 54:2787–2805, 2010.

[2] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, Prashant Shenoy, and Jeannie Albrecht. Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes. In *2012 Workshop on Data Mining Applications in Sustainability (ACM SustKDD 2012)*. ACM, 2012.

[3] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4:2292–2303, 2016.

[4] Ana Reyna, Cristian Martin, Jaime Chen, Enrique Soler, and Manuel Diaz. On blockchain and its integration with IoT. Challenges and opportunities. *Future Generation Computer Systems*, 88:173–190, 2018.

[5] Florian Tschorsch and Björn Scheuermann. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys and Tutorials*, 18(3):2084–2123, 2016.