Edge Blockchain Provisioning for Mobile Edge Computing Applications

Diploma Thesis Presentation

BSc Filip Rydzi
Advisor: Privatdoz. Dr.techn. Hong-Linh Truong

Faculty of Informatics, TU Wien

June 3, 2019

Overview

Introduction & Motivation

Introduction

Mobile Edge Computing (MEC)

- An architecture, which brings cloud capabilities (processing, storage, etc.) closer to the users.
- Additional layer between cloud and Internet of Things (IoT).

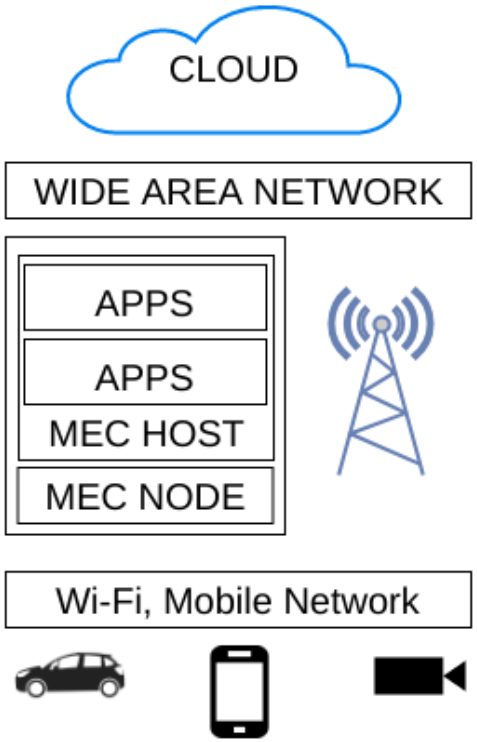


Figure: MEC architectural overview, taken from [1]

Introduction

MEC

- An architecture, which brings cloud capabilities (processing, storage, etc.) closer to the users.
- Additional layer between cloud and IoT.

Blockchain for MEC

- Decentralized trust-less peer-to-peer messaging solution.
- Autonomous management of IoT devices.

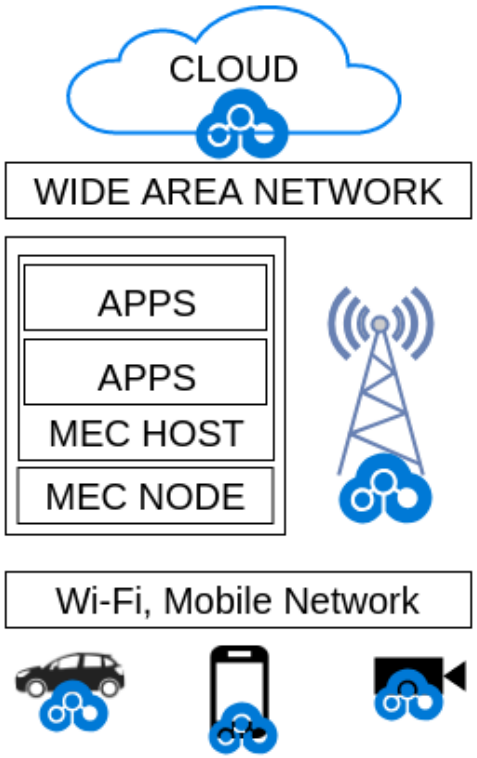


Figure: MEC with blockchain architectural overview, taken from [1]

Motivation Scenario

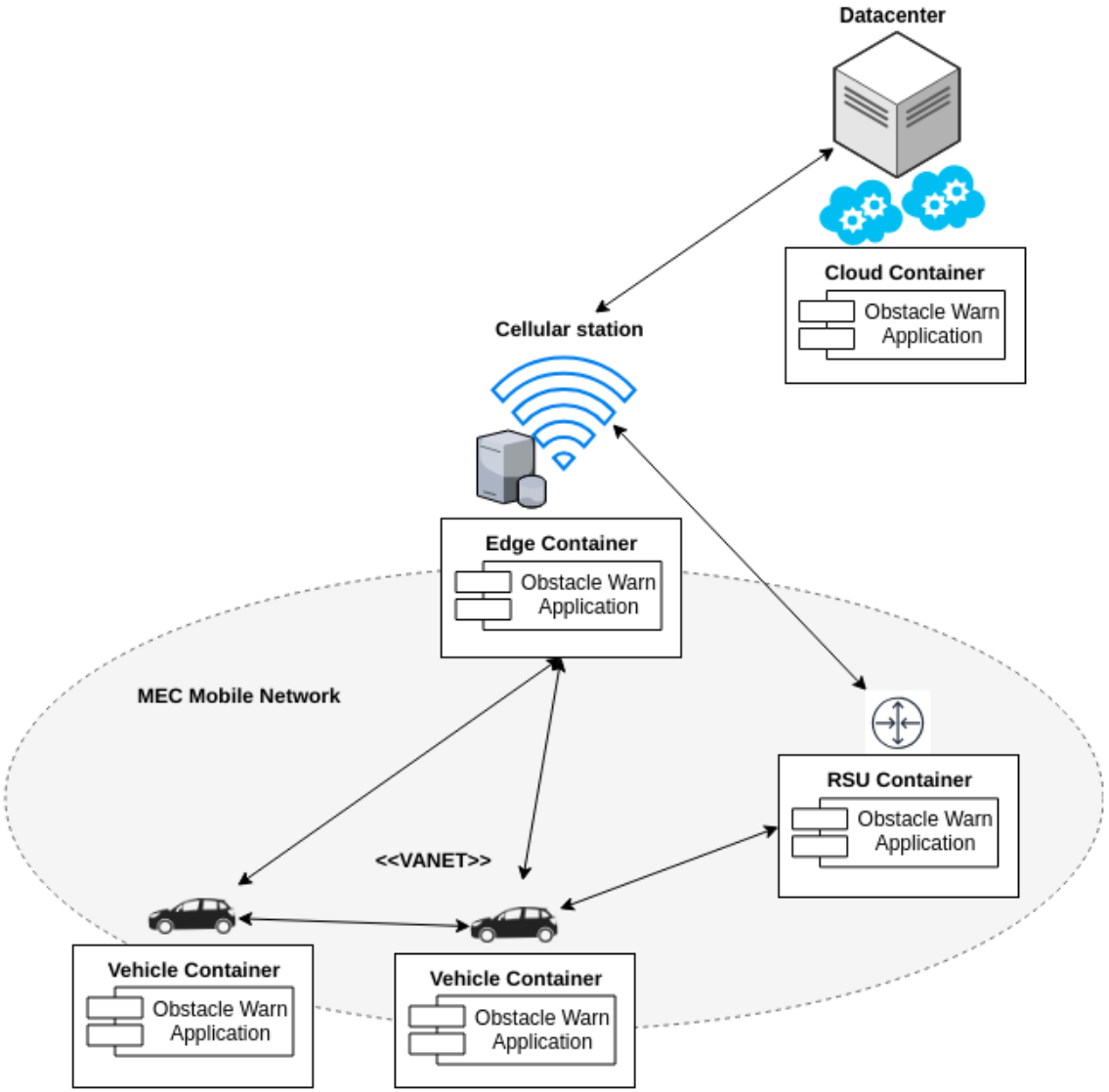


Figure: Motivation Scenario

Motivation Scenario

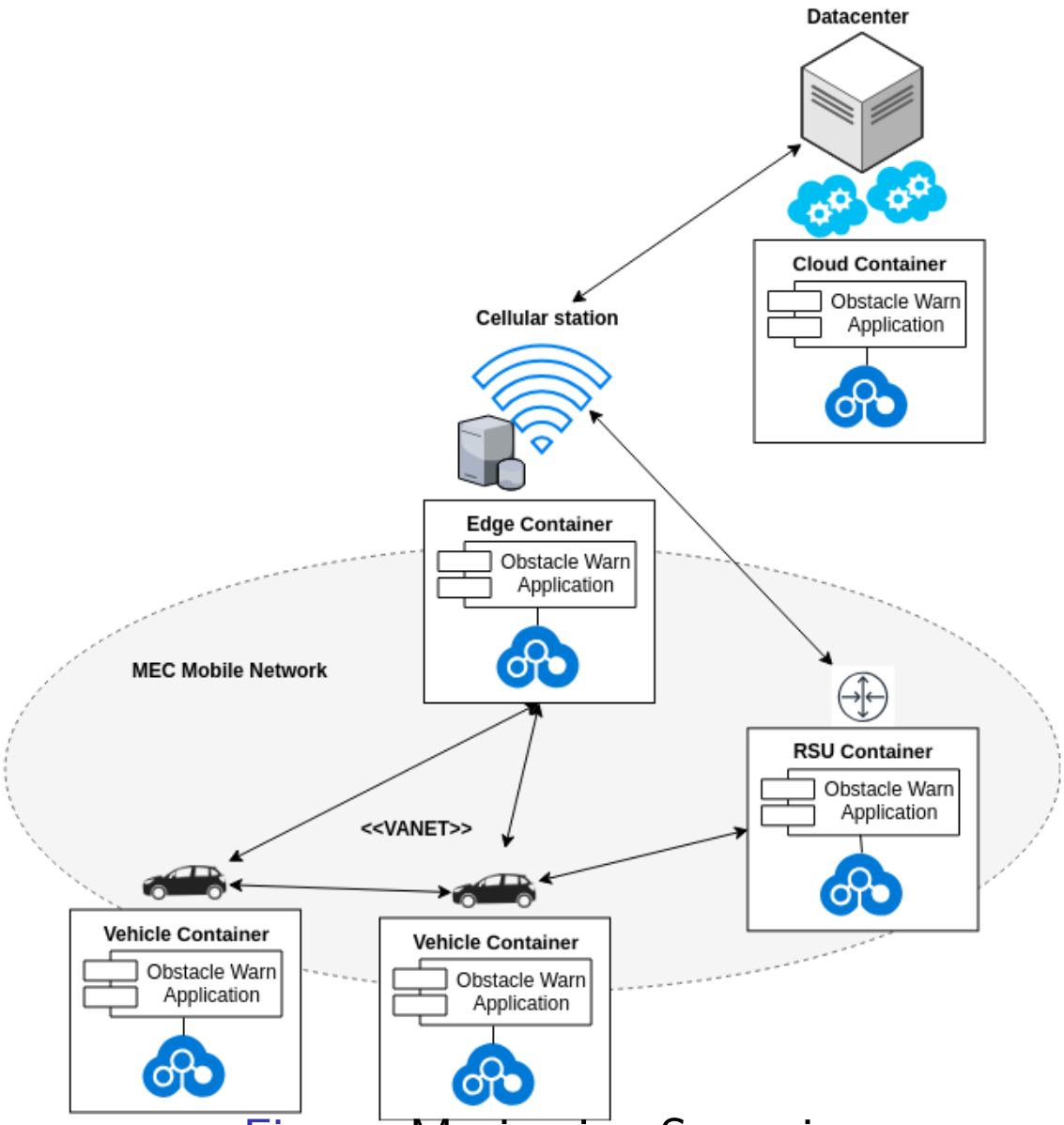


Figure: Motivation Scenario

Motivation Scenario

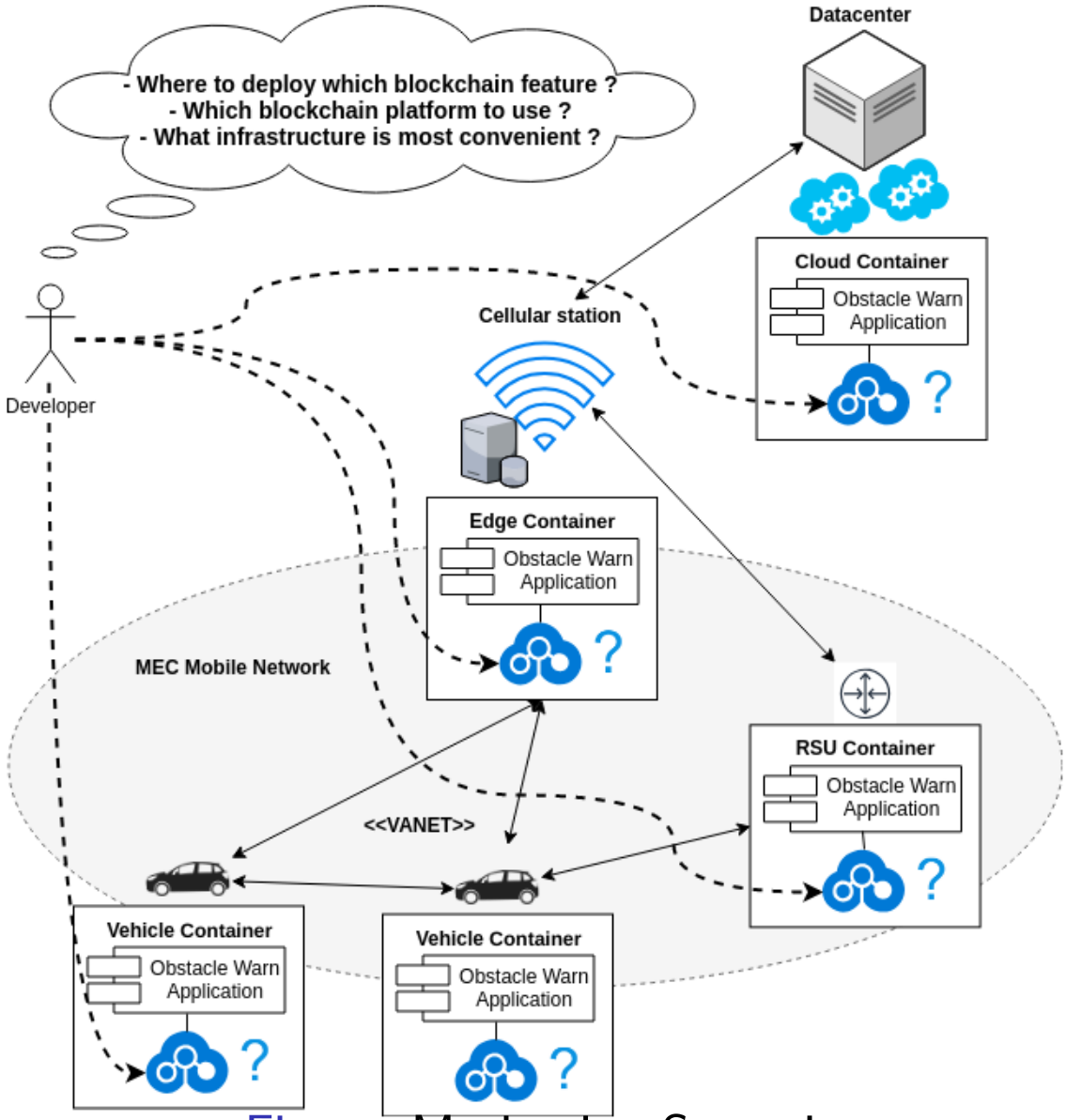


Figure: Motivation Scenario

Research Objectives

1. Benchmark different patterns of blockchain interactions among MEC components. Evaluate various deployments of blockchain artefacts to the components and various configurations of infrastructure, consisting of compute resources and networks.
2. Provide knowledge gathered by the benchmarks to the developers.

Contribution

- Benchmark Framework
  - Benchmark blockchain interactions among MEC components in an application's topology.
- Experiment Knowledge Service
  - Manage knowledge gathered by benchmarks to help developers during design phase of an application.

Benchmarks

Quality Metrics

- Transaction Acceptance Rate
  - The ratio of accepted transactions to the ones which have been submitted to blockchain.
- Transaction Acceptance Time
  - The time it takes to accept a transaction by blockchain.
- Synchronization State
- Scalability
- Infrastructure Resources Utilization

Scenarios and Interactions

Examples of interaction patterns:

1. Vehicle to Vehicle Interaction
2. Vehicle to Road-side unit (RSU) Interaction
3. Vehicle to Edge Interaction
4. Vehicle to RSU and Edge Interaction
5. Vehicle to Edge and Cloud Interaction
6. Vehicle to RSU, Edge and Cloud Interaction

Scenarios and Interactions

Examples of interaction patterns:

1. Vehicle to Vehicle Interaction
2. **Vehicle to RSU Interaction**
   - *Obstacle on the road warning scenario*
3. Vehicle to Edge Interaction
4. Vehicle to RSU and Edge Interaction
5. Vehicle to Edge and Cloud Interaction
6. Vehicle to RSU, Edge and Cloud Interaction

Benchmarks Design

Table: A deployment of blockchain features for Interaction 2 (vehicle-RSU-vehicle)

Interaction id	**Blockchain features deployment**				
	ID	**vehicle**	**RSU**	**Edge**	**Cloud**
2	0	*creator*	*all*	-	-
	2	*all*	*creator*	-	-
	4	*all*	*all*	-	-

Blockchain features

- *Creator feature* (creating, signing, submitting and verifying a transaction, accepting a block)
- *Consensus feature* (achieve consensus)

Benchmark Framework

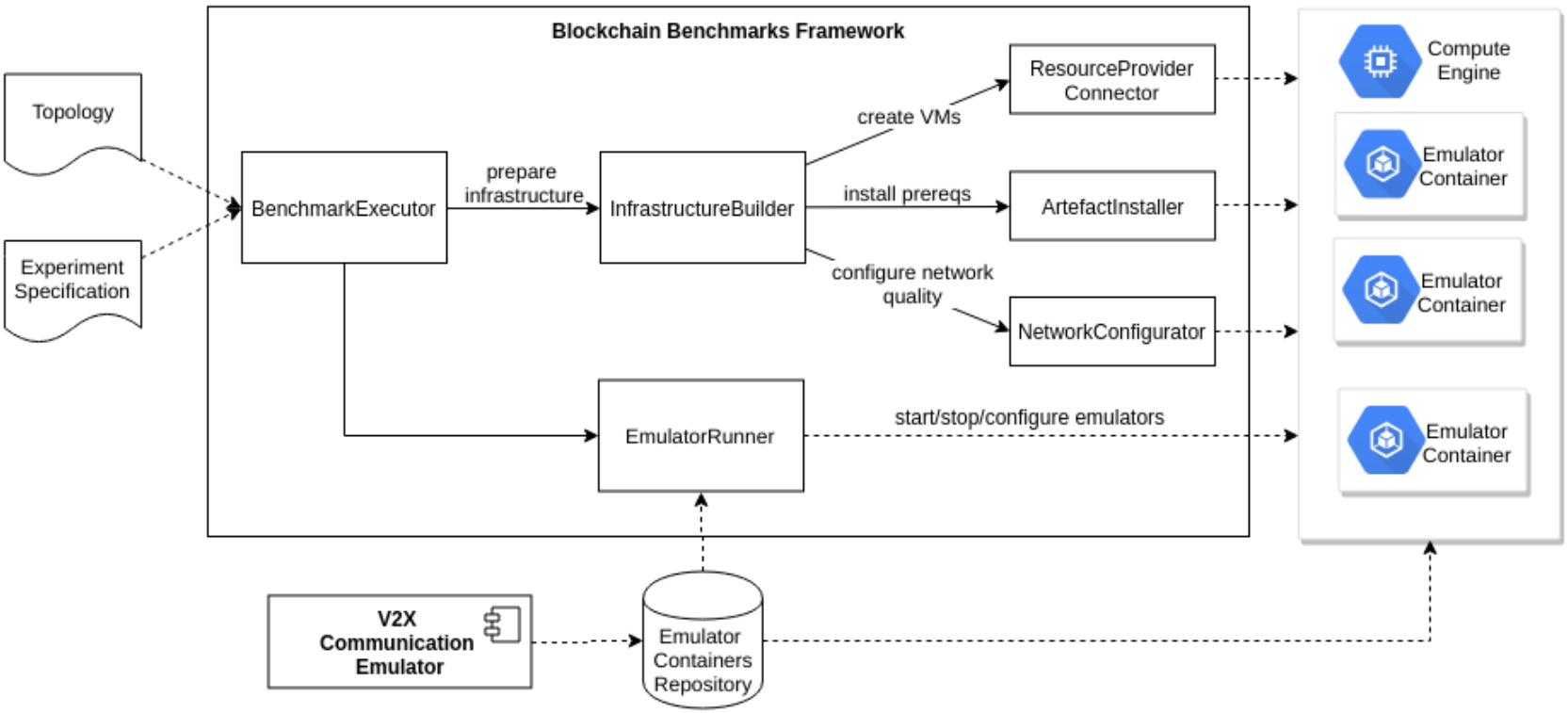
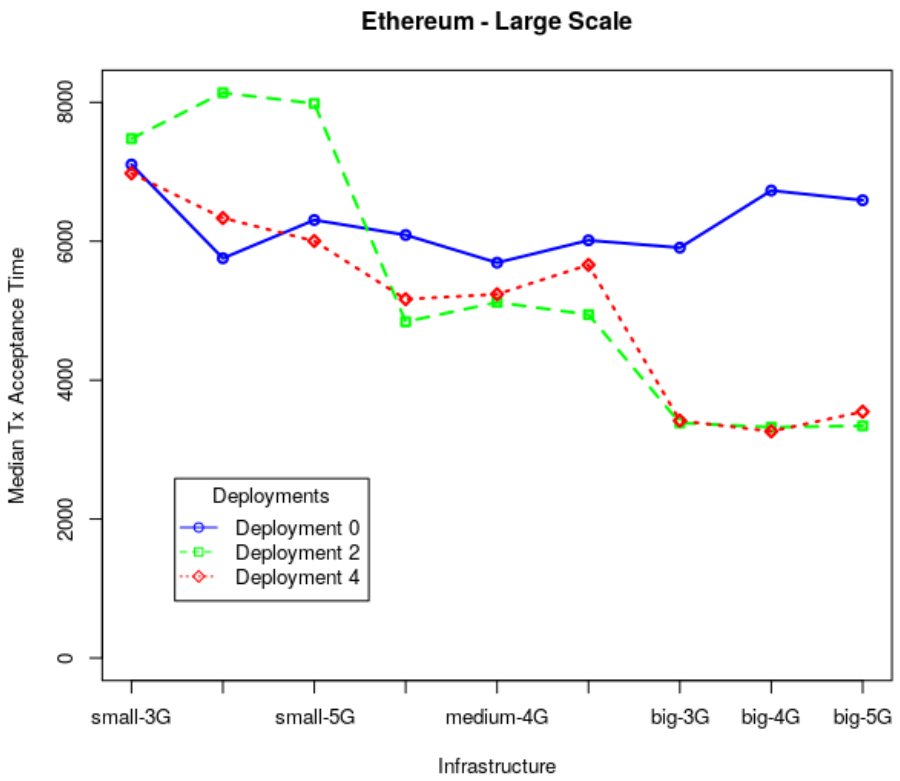


Figure: Component diagram of the framework

- Generate and benchmark experiments based on a specification.
- Build emulated MEC infrastructure.
- Deploy blockchain artefacts into a specified topology.
- Emulate and benchmark blockchain interactions among MEC components in the topology.

Experiments

- 324 experiments have been generated and benchmarked by the benchmark framework.



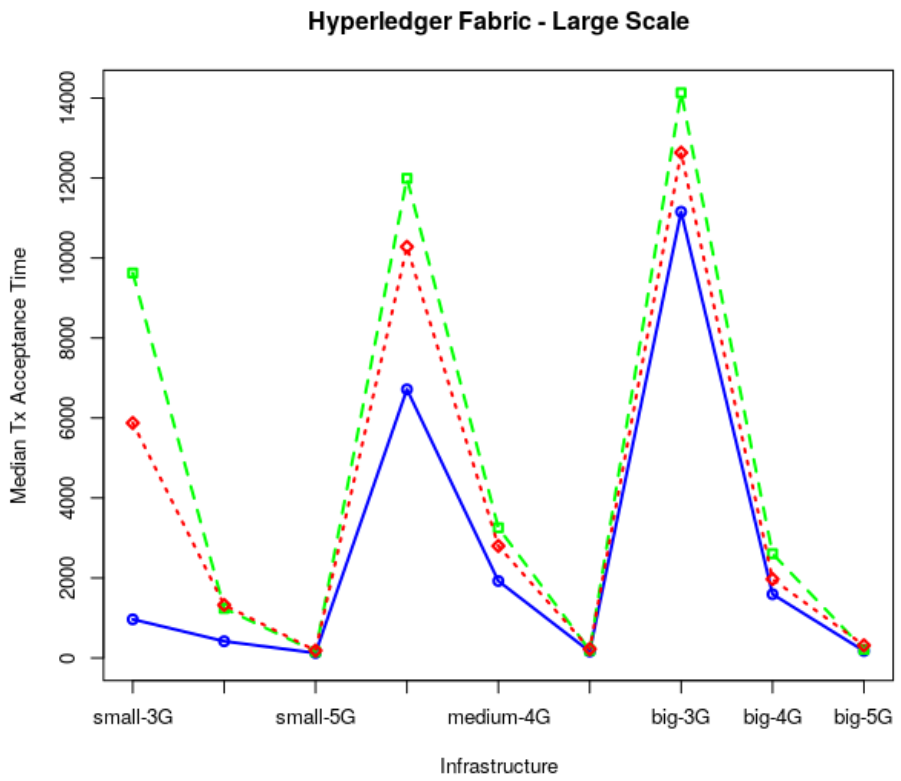


Figure: Median of transaction acceptance times for interaction 2 (vehicle-RSU-vehicle)

Experiments - Discussion

- All identified interactions for the *obstacle on the road warning scenario* have been benchmarked.
  2. Vehicle to RSU Interaction
  3. Vehicle to Edge Interaction
  4. Vehicle to RSU and Edge Interaction
- Best results concerning reliability and performance have been measured for: *interaction 2, Hyperledger-Fabric blockchain, deployment 0, small machine type for vehicles and 5G network* .

Experiments Knowledge Service

Overview

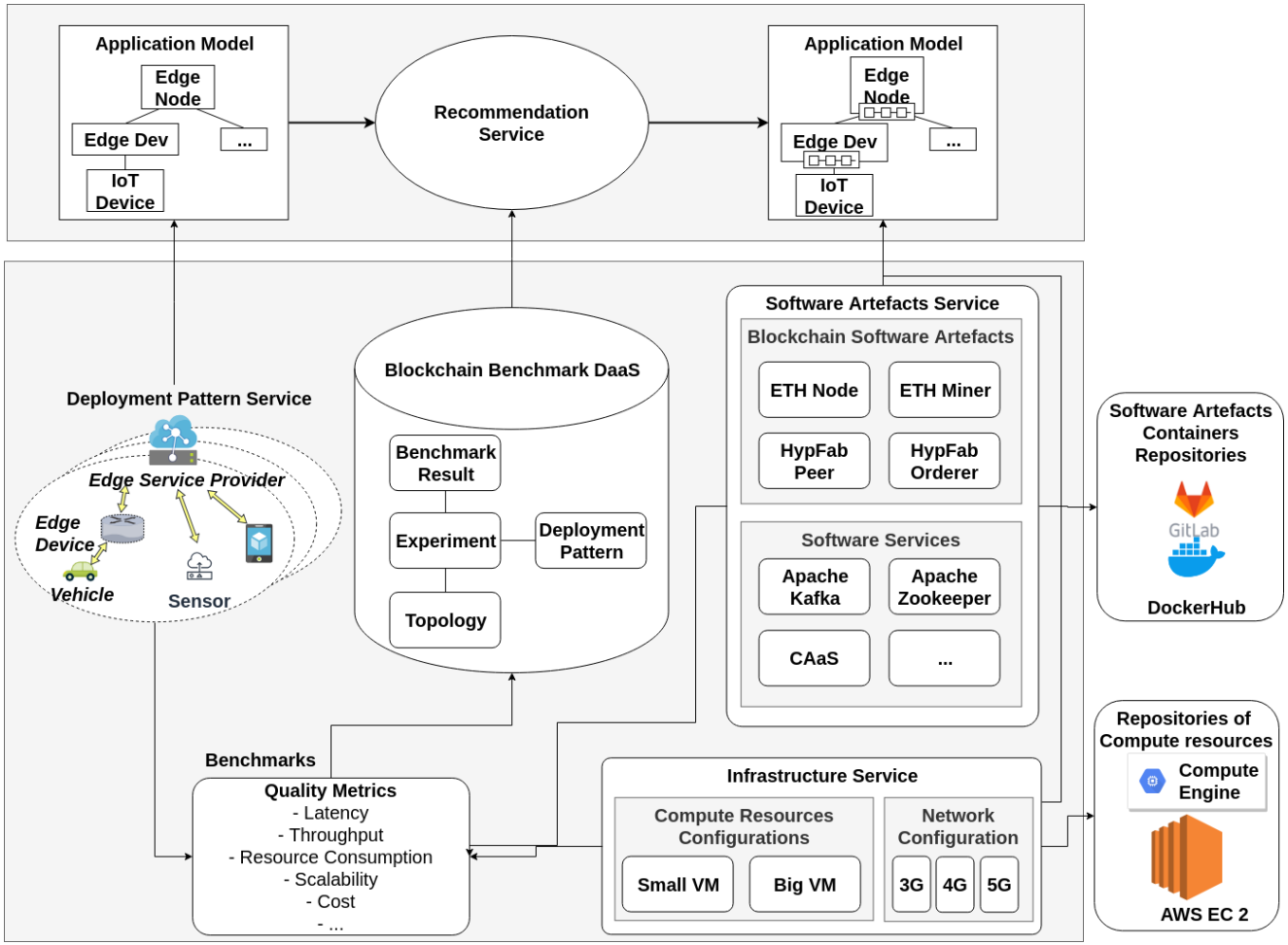


Figure: High-level architectural overview of Experiments Knowledge Service

Data model

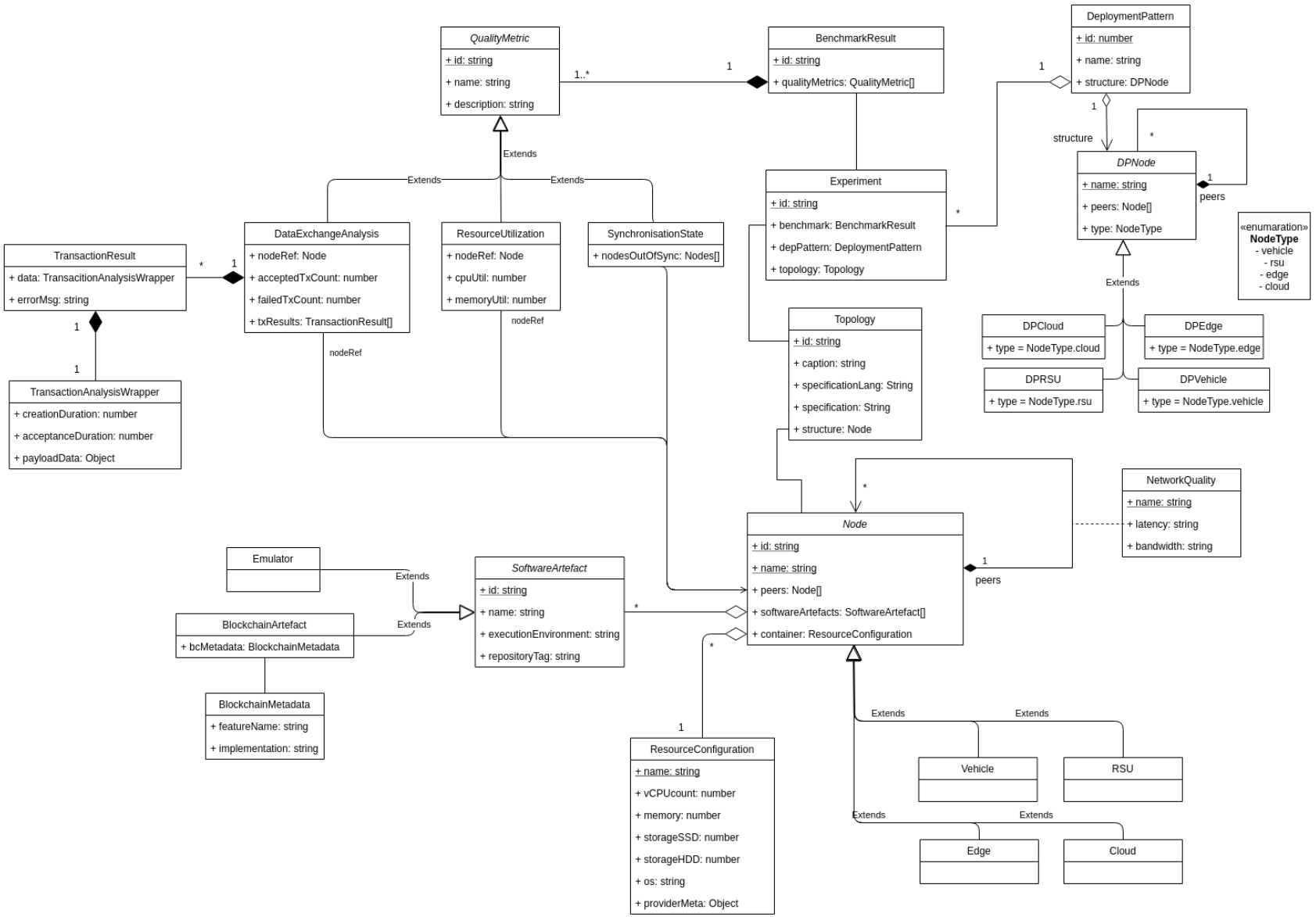


Figure: Model of data stored in Experiments Knowledge Service

Features

- Sharing benchmarks with the Experiment Knowledge Service.
- Search benchmarking interactions, topologies or infrastructures.
- Recommend a deployment of blockchain artefacts into a model of application in MEC.

Examples

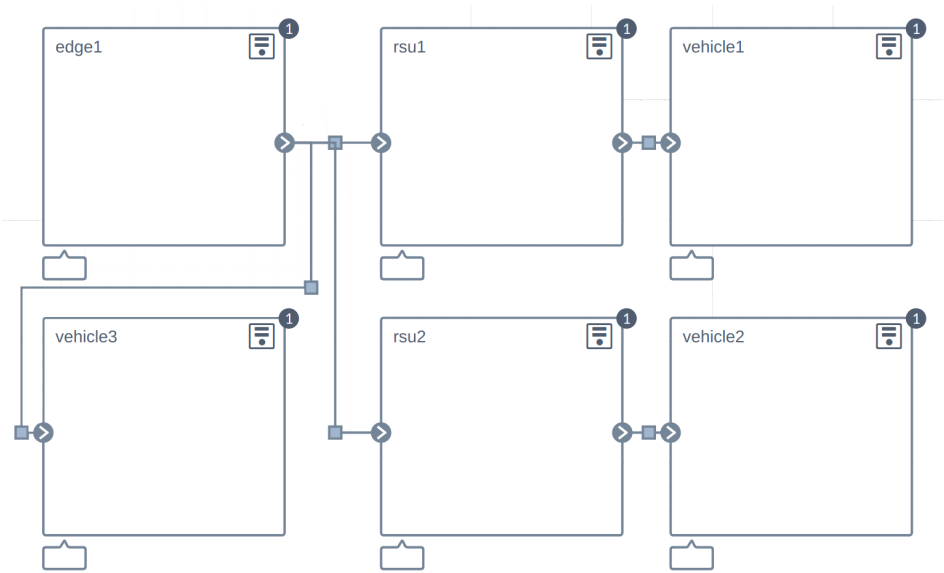


Figure: Input for Recommendation Service, depicted via Cloudify Composer [2]

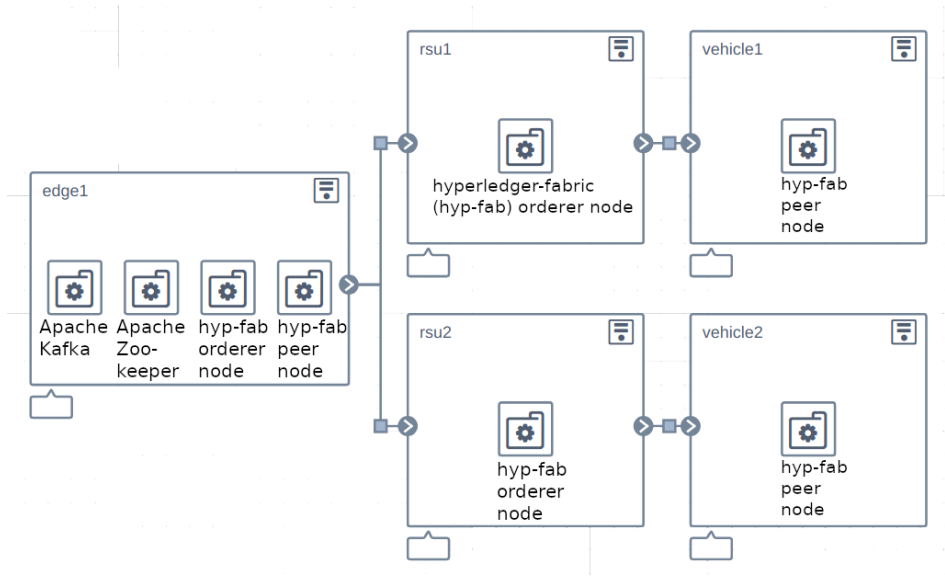


Figure: Output from Recommendation Service, depicted via Cloudify Composer

[2] https://docs.cloudify.co/4.5.0/developer/composer/

Prototype & Demo

Prototypes & Demo

- Benchmark Framework
  - Dockerized NodeJS application, developed in Typescript.
- Experiment Knowledge Service
  - Dockerized NodeJS application, developed in Typescript, stores data in MongoDB and Neo4J.

The prototypes are available in the GitHub repository [3].

[3] `https://github.com/rdsea/blockchainbenmarkservice`

Conclusions & Future Works

Conclusions

- Blockchain in MEC brings new challenges for developers.
- To help the developers we introduced a framework able to benchmark blockchain interactions among MEC components.
- 324 experiments have been performed to demonstrate flexibility of the framework.
- To enable reuse of the knowledge gathered by benchmarks we developed an Experiments Knowledge Service.

Current & Future Works

Contribution Papers:

- Benchmarking Blockchain Interactions in Mobile Edge Cloud Software Systems
  - submitted to IEEE MASCOTS 2019 [4]
- Sharing Blockchain Performance Knowledge for Edge Service Development
  - submitted to ICSOC 2019 [5]

[4] https://sites.google.com/view/mascots-2019
[5] https://icsoc-laas.fr/

References


K. Dolui and S. K. Datta.
Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing.
In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6, June 2017.

Backup slides

Quality Metrics

- Transaction Acceptance Rate
  - The ratio of accepted transactions to the ones which have been submitted to blockchain.
- Synchronization State
  - The number of blockchain nodes, which have been removed from a blockchains topology during a period of time.
- Transaction Acceptance Time
  - The time it takes to accept a transaction by blockchain.
- Scalability
  - Changes in transaction acceptance rate and time and in synchronization state when scale of a topology is increased.
- Infrastructure Resources Utilization
  - % utilization of a CPU core, RAM memory, etc.

Vehicle to Vehicle Interaction

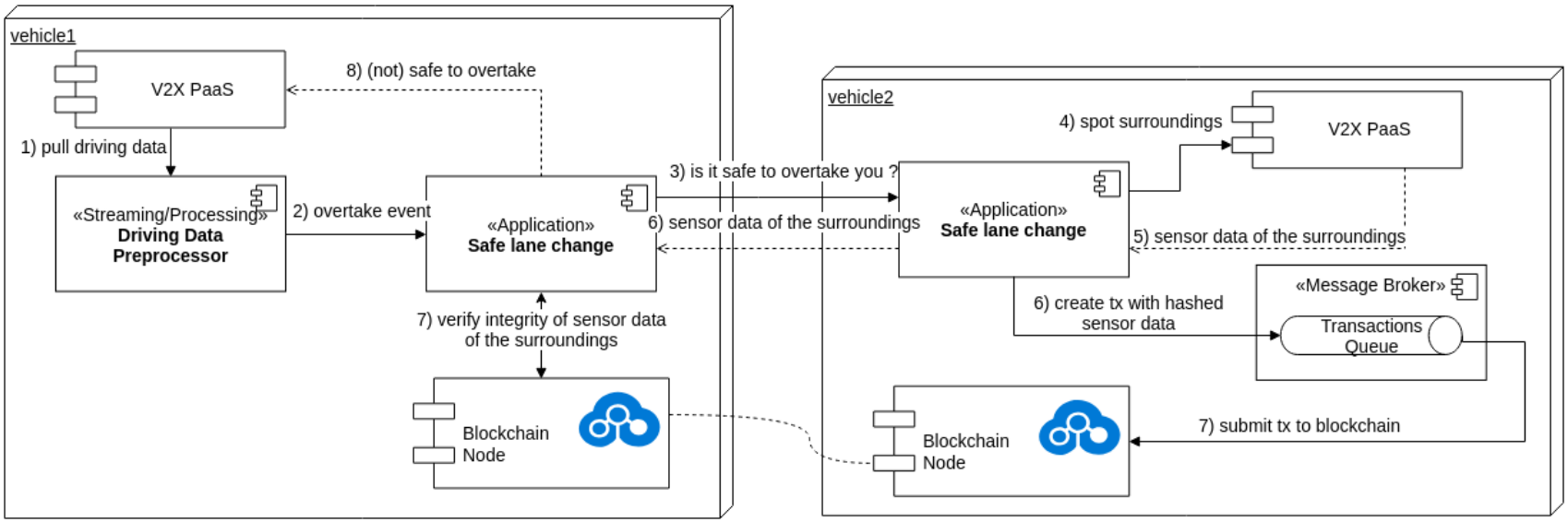


Figure: Lane Change Scenario in a blockchain-enabled V2X communication

Vehicle to RSU Interaction

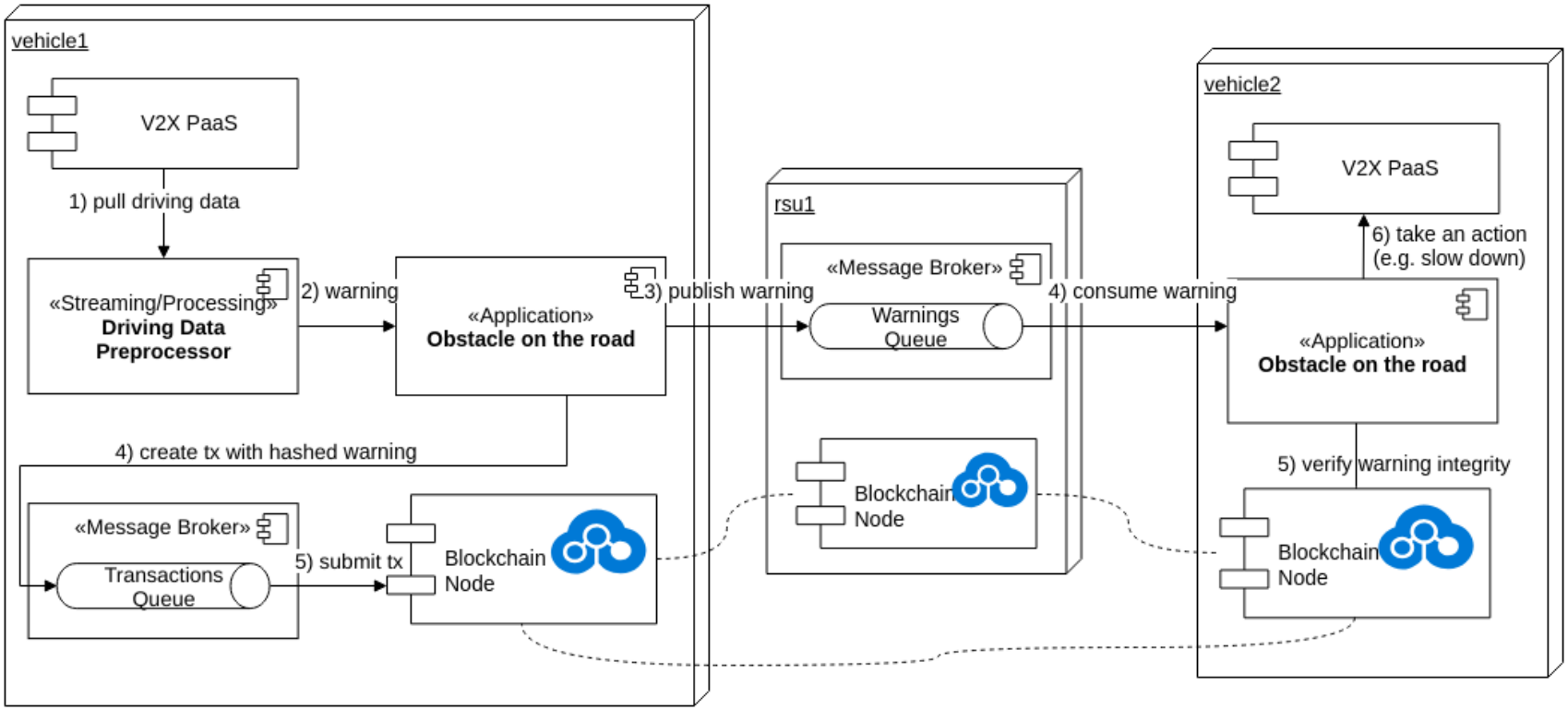


Figure: The *obstacle on the road warning scenario* in blockchain-enabled V2X over RSU

Vehicle to Edge Interaction

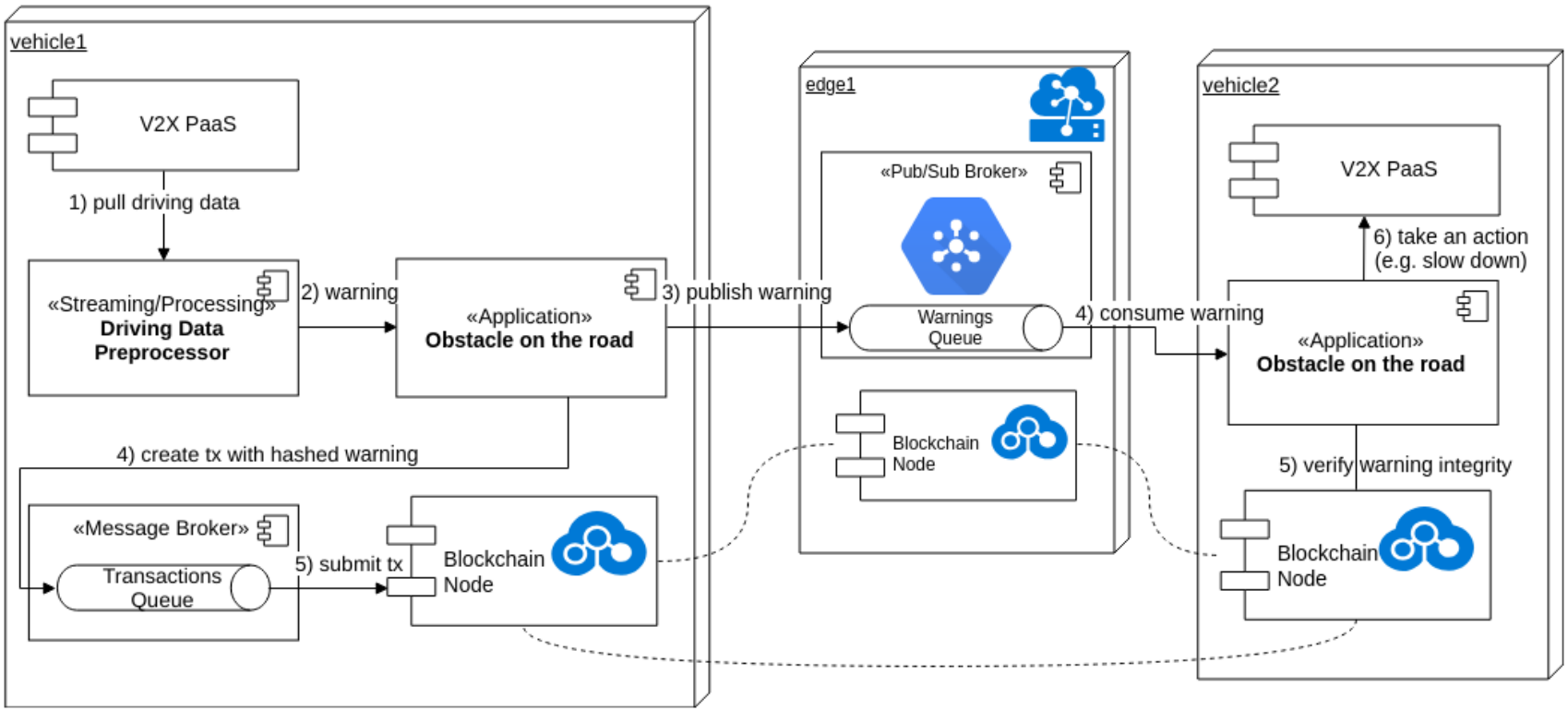


Figure: The *obstacle on the road warning scenario* in blockchain-enabled V2X over edge node

Vehicle to RSU and Edge Interaction

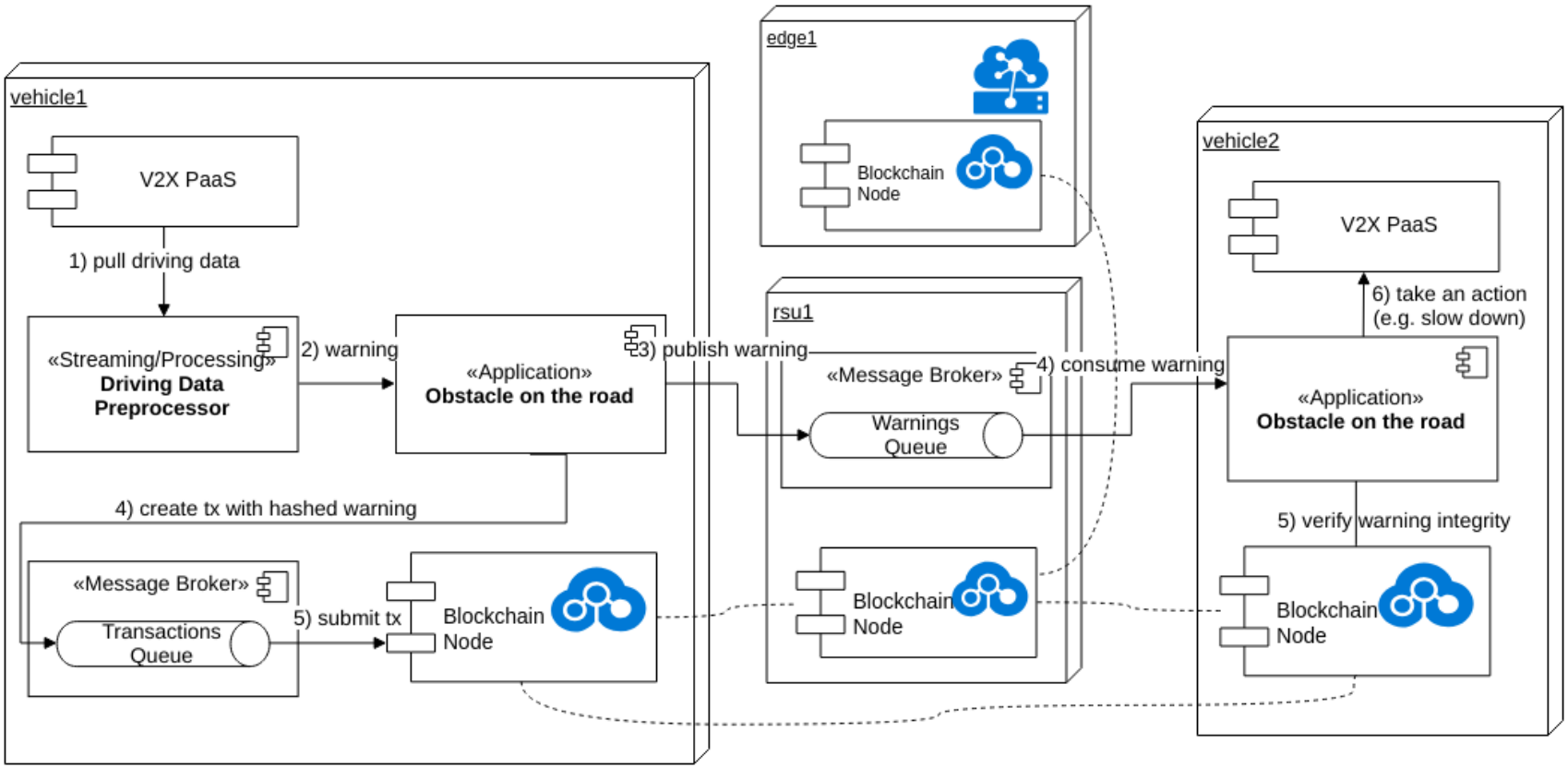


Figure: The *obstacle on the road warning scenario* in blockchain-enabled V2X over RSU and edge node

Blockchain Artefact

Blockchain features

- *Creator feature*
  - Create a transaction
  - Sign a transaction
  - Submit a transaction
  - Verify a transaction
  - Accept a block
- *Consensus feature*
  - Achieve consensus

Blockchain nodes

- Standard node
  - Capable of executing the *Creator feature*.
  - Real examples: `Geth node`, `Hyperledger-Fabric peer node`
- Miner node
  - Capable of executing the *Consensus feature*.
  - Real examples: `Geth miner node`, `Hyperledger-Fabric orderer node`

Blockchain Artefacts

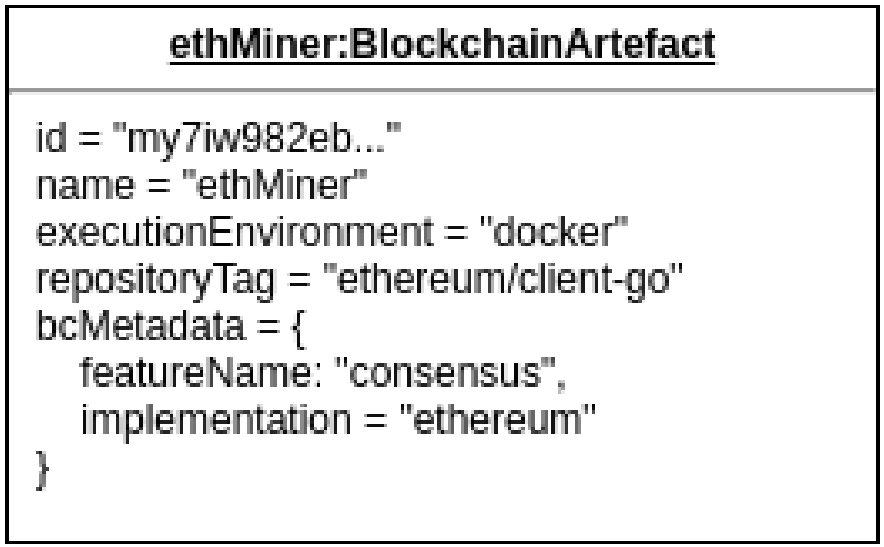


Figure: Ethereum miner node

Benchmark Infrastructure

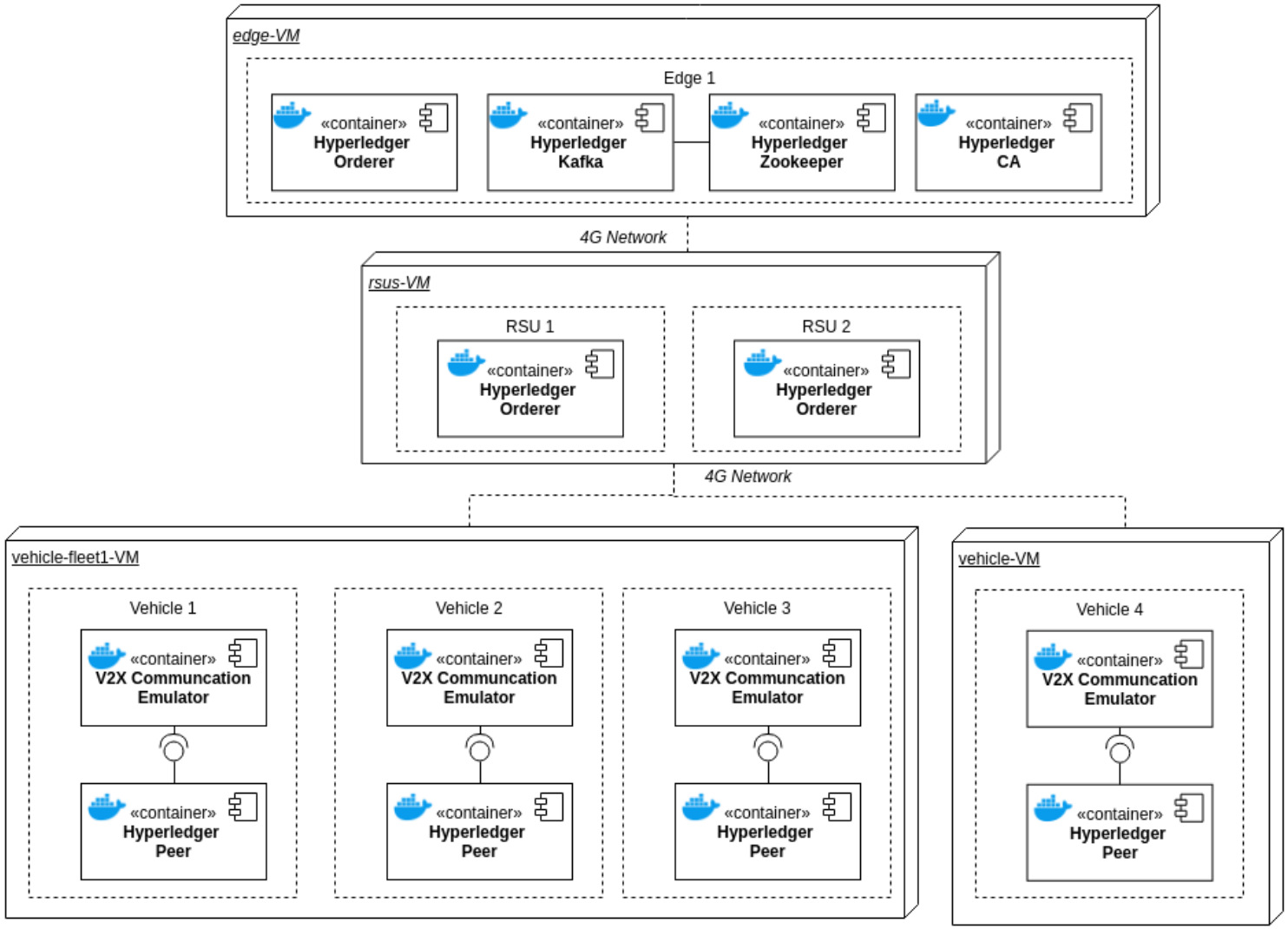


Figure: Example of emulated MEC infrastructure

Benchmark Framework

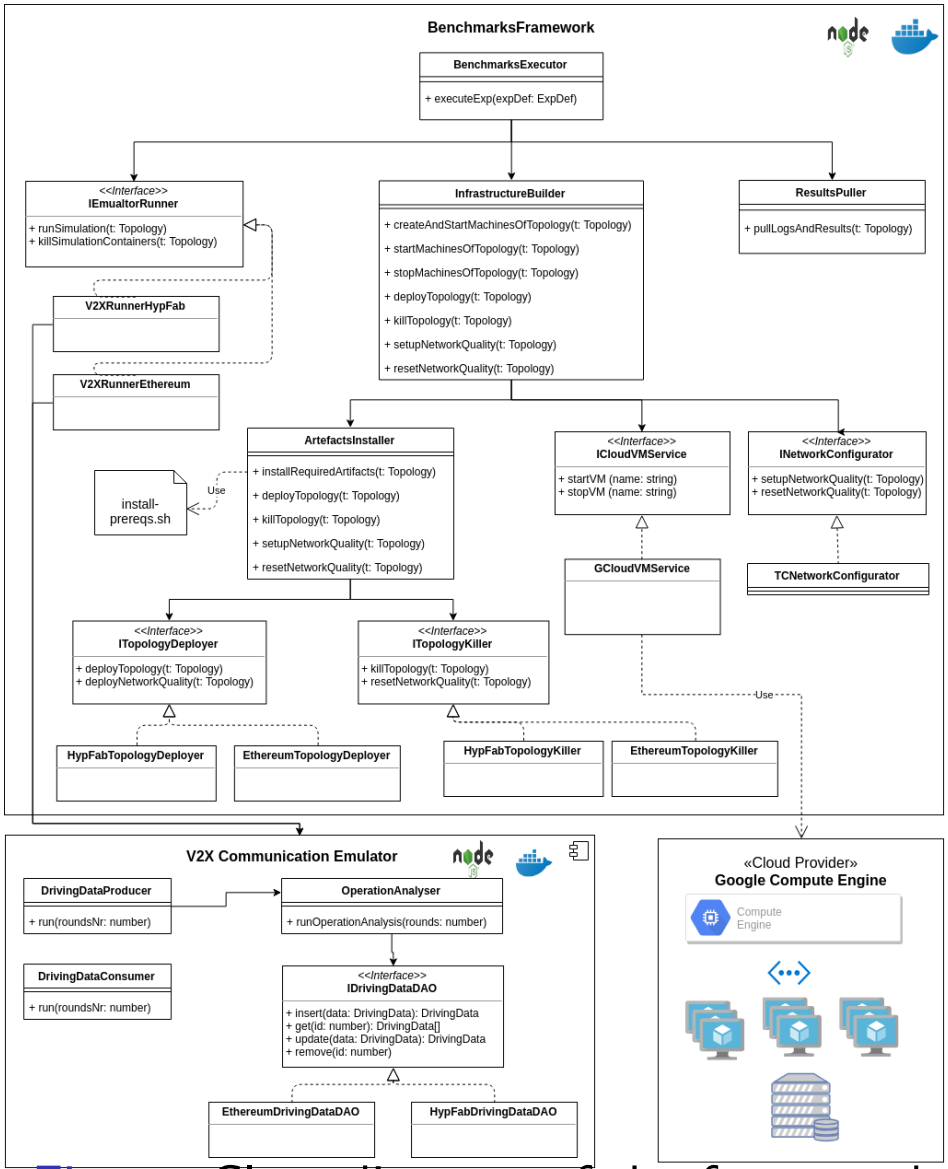


Figure: Class diagram of the framework

Benchmark Framework - Input Topology

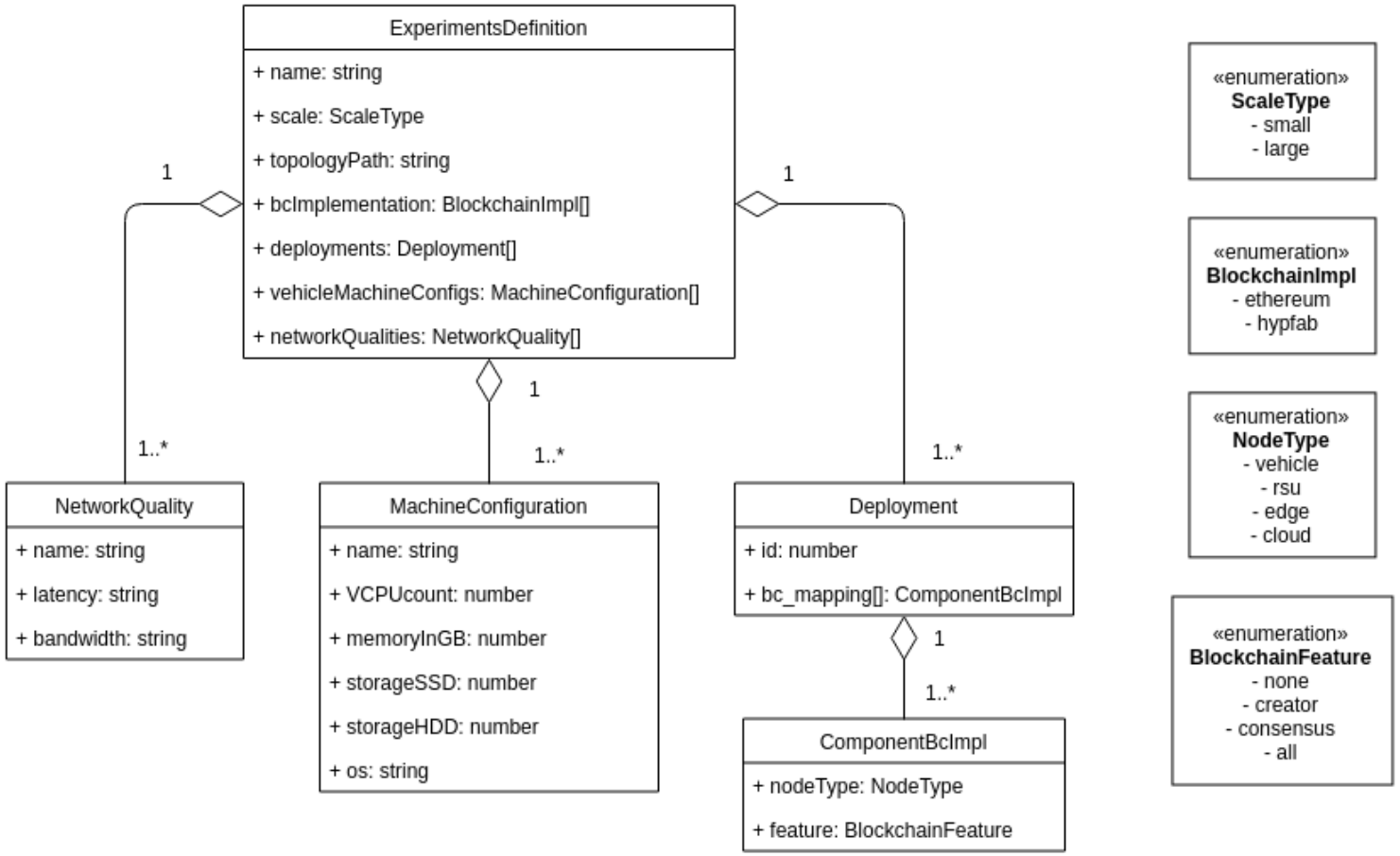


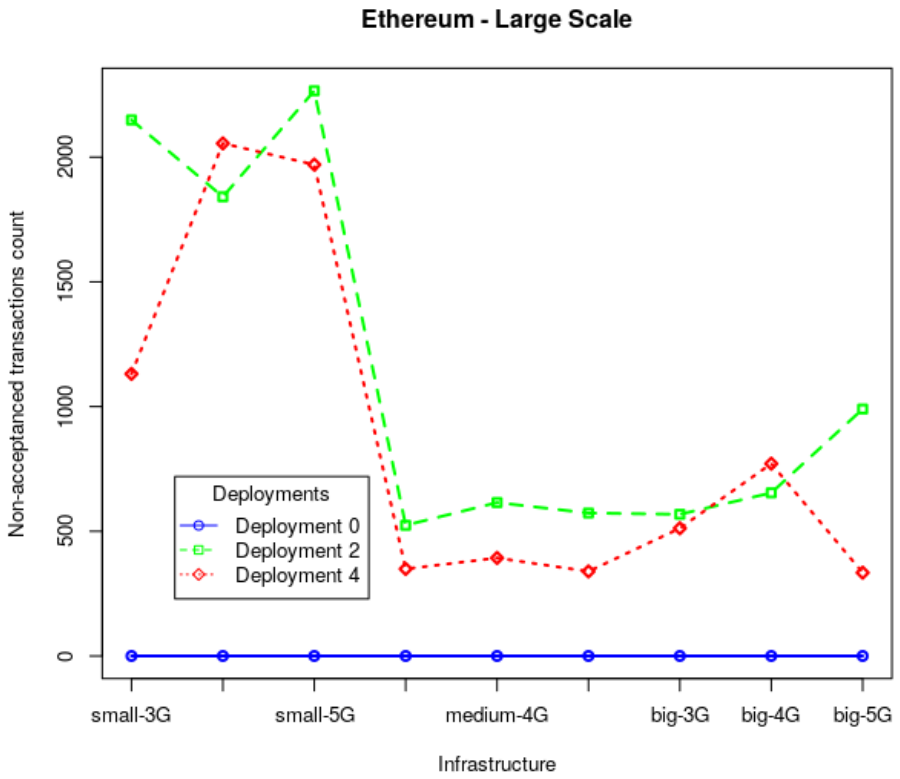
Figure: JSON representation of topology accepted by benchmark framework

Benchmark Framework - Experiment Specification

```
1  name: interaction2
2  description: Experiments
       for Interaction2
3  workloadEmulator:
4  - type: docker
5    imageTag: filiprydzi/
         v2x_communication
6    roundsNr: 100
7  bcImplementations:
8  - eth
9  - hypfab
10 bcDeployments:
11 - id: 4
12   featuresMapping:
13   - nodeType: rsu
14     feature: all
15   - nodeType: vehicle
16     feature: all
17 ...
```

```
18 vehicleContainerConfigurations
       :
19 - name: small
20   vCPUcount: 1
21   memory: 2
22   storageSSD: 10
23   storageHDD: 0
24   os: ubuntu18.04
25   ...
26 networkQualities:
27 - name: 3G
28   latency: 200ms
29   bandwidth: 1000kbps
30   ...
```

Experiments



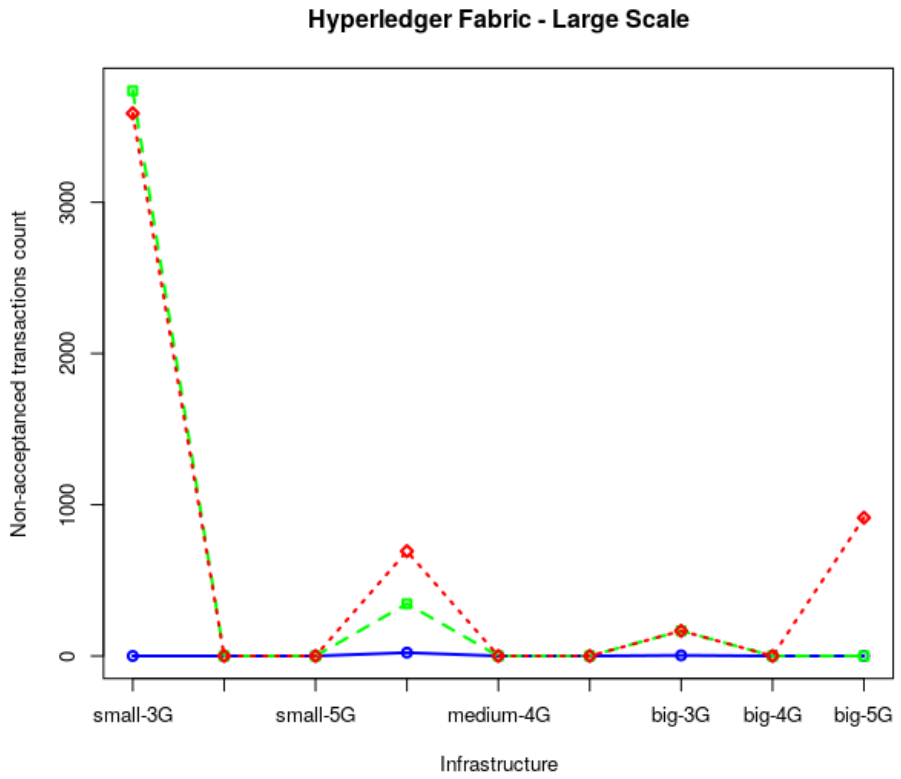


Figure: Number of rejected transactions in interaction 2 (vehicle-RSU-vehicle)

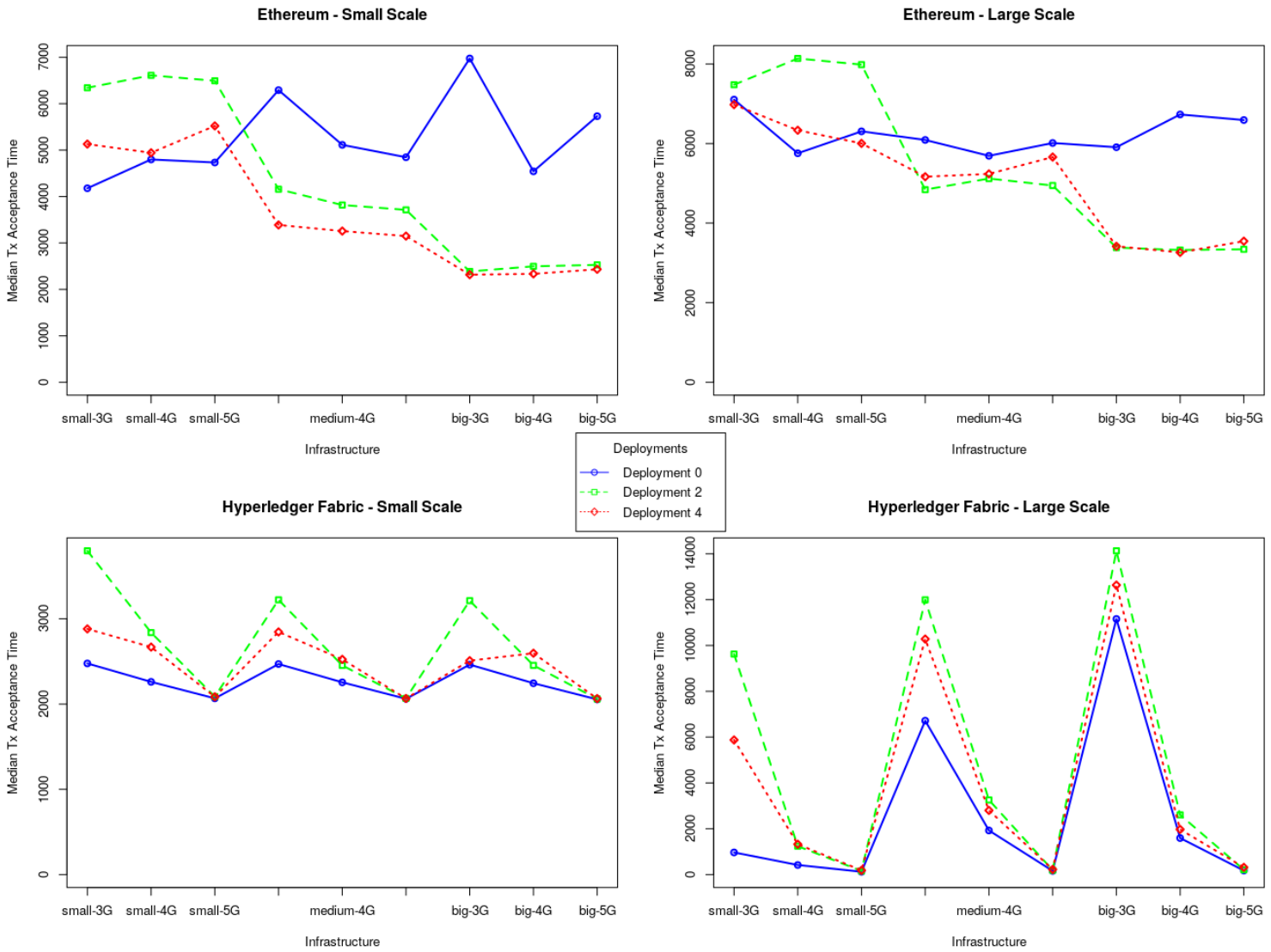


Figure: Median of transaction acceptance times for interaction 2 (vehicle-RSU-vehicle)

Experiments

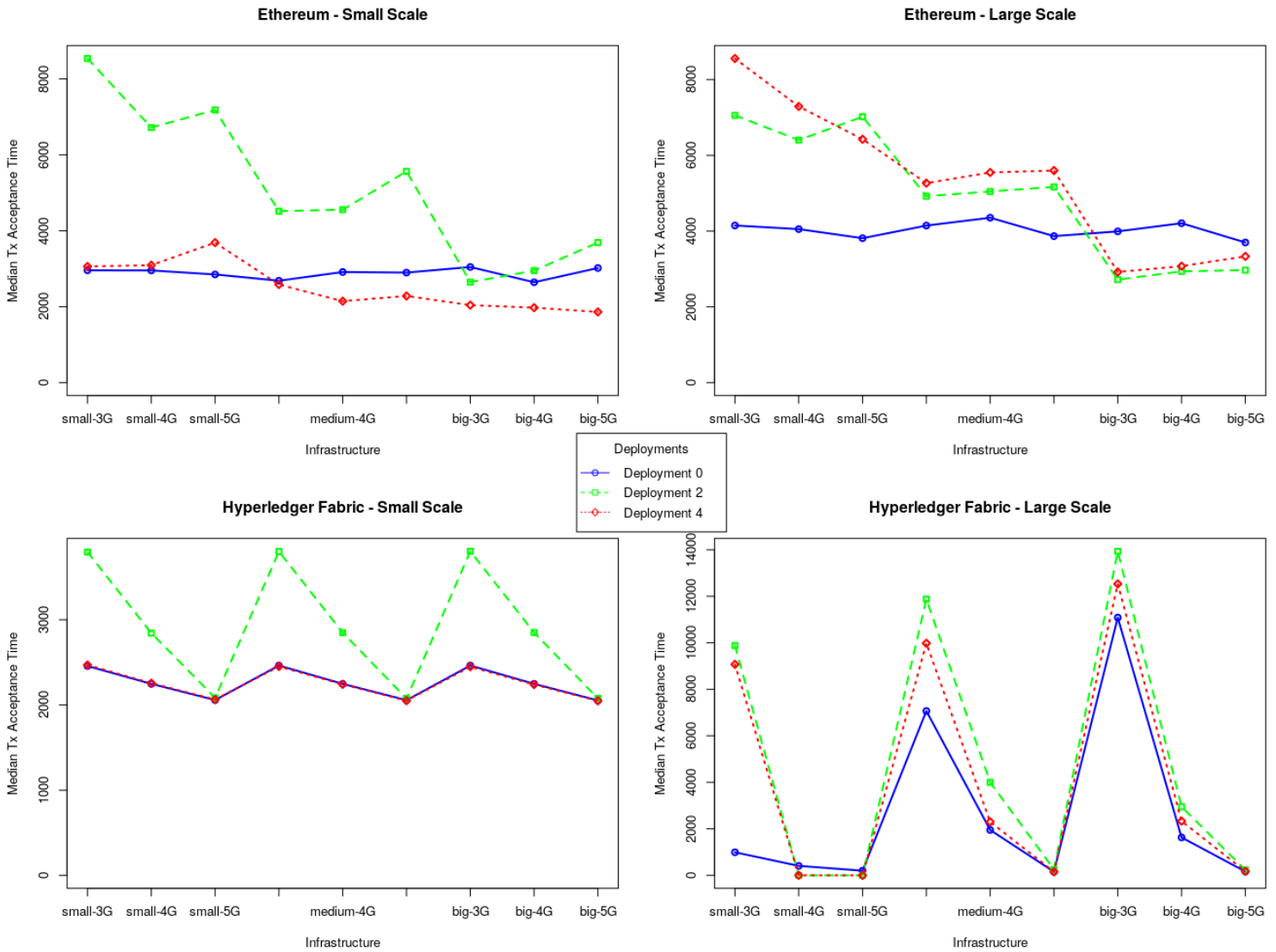


Figure: Median of transaction acceptance times for interaction 3 (vehicle-edge node-vehicle)

Experiments

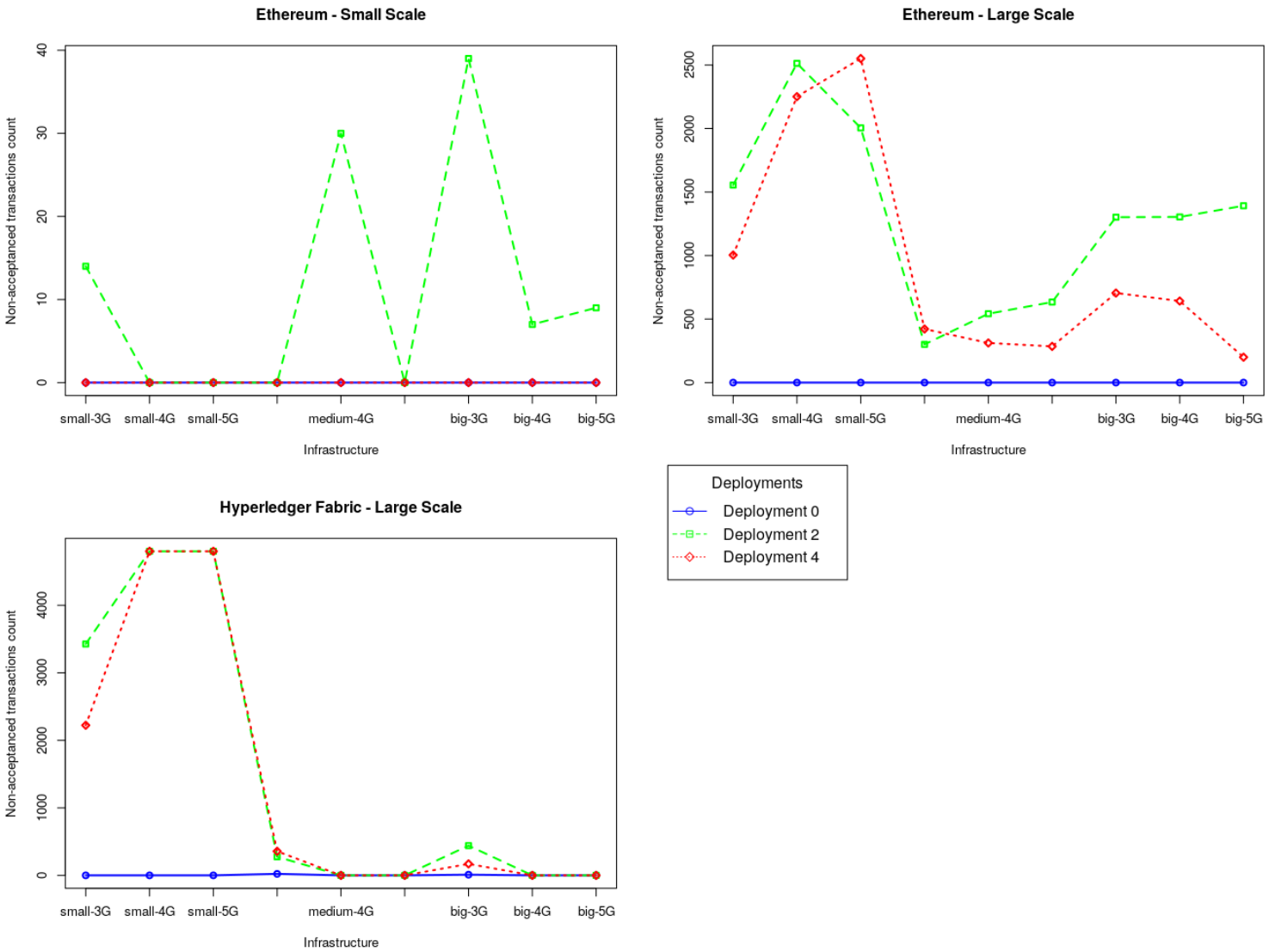


Figure: Number of rejected transactions in interaction 3 (vehicle-edge node-vehicle)

Experiments

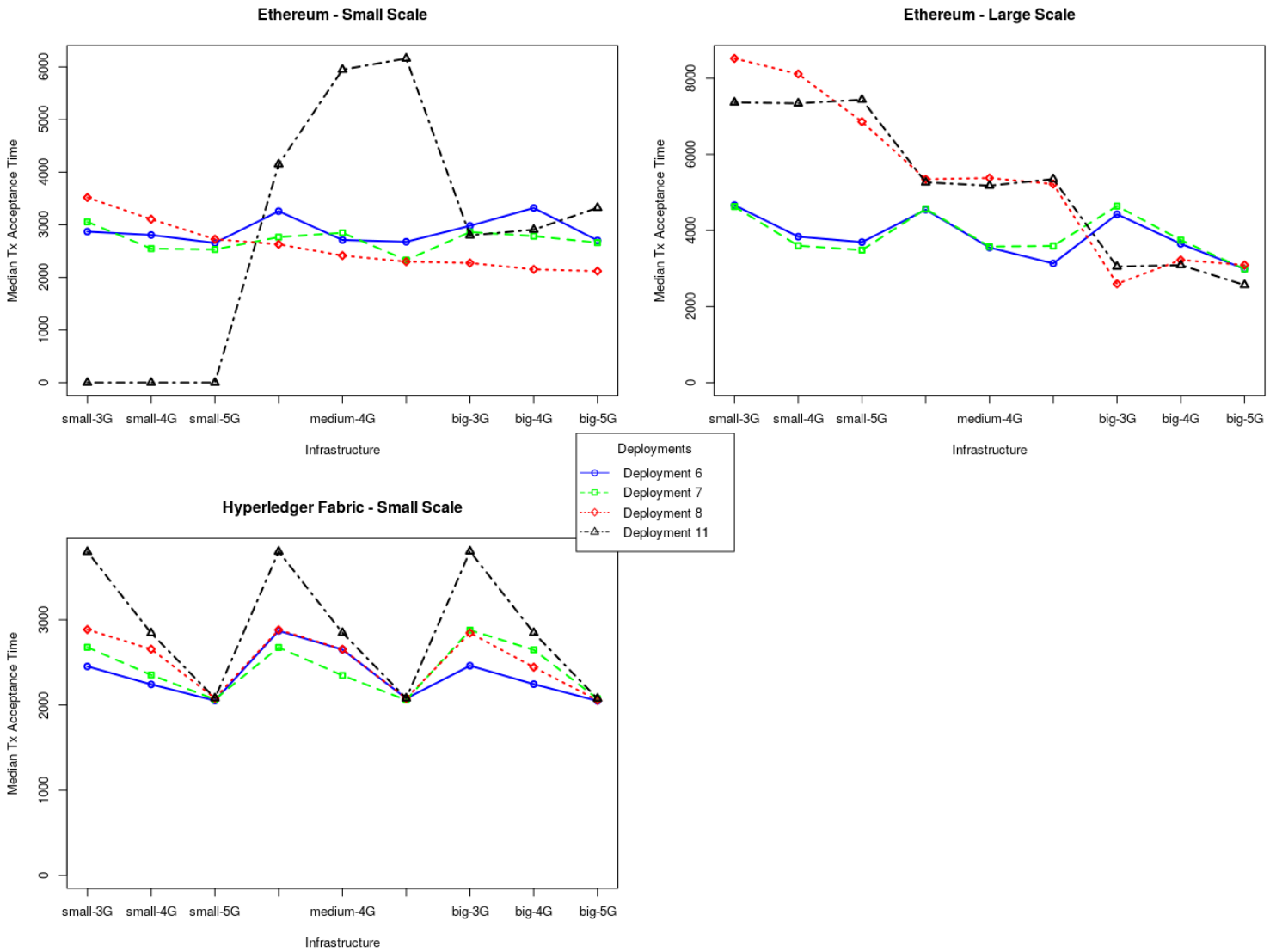


Figure: Median of transaction acceptance times for interaction 4 (vehicle-RSU-edge node-vehicle)

Experiments

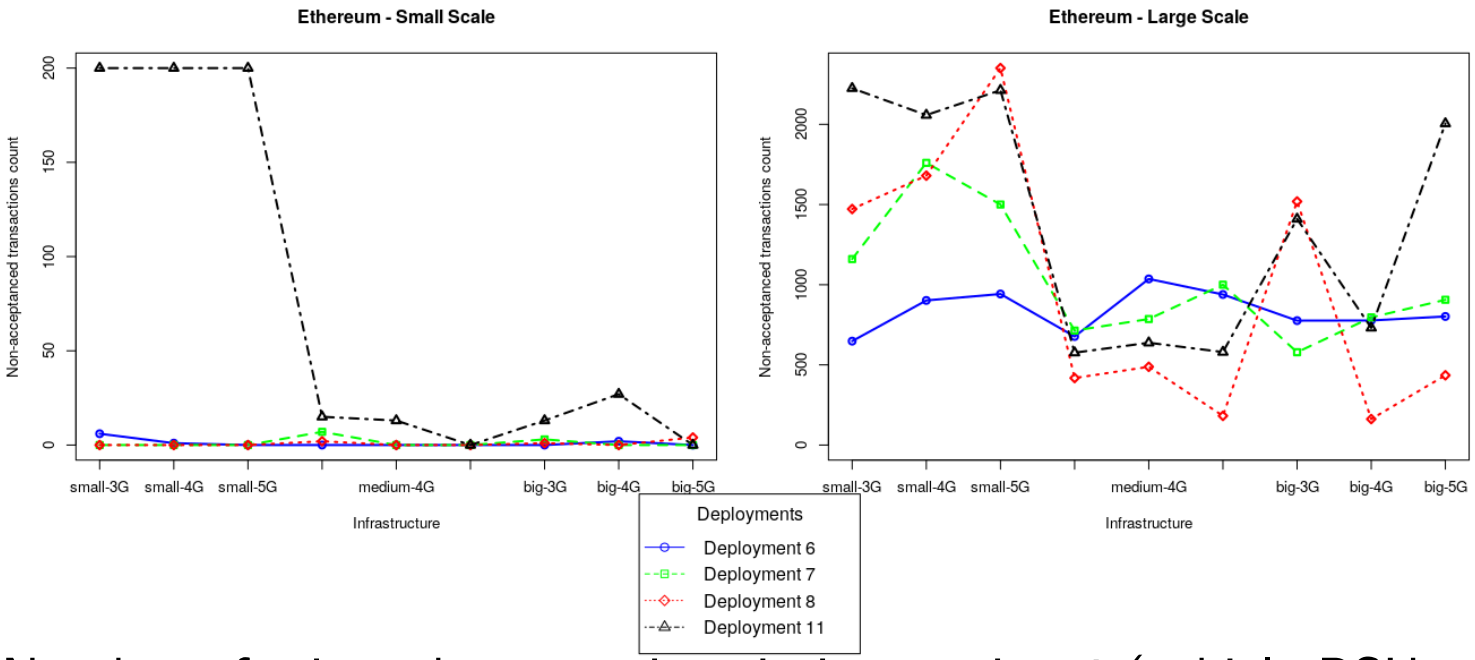


Figure: Number of rejected transactions in interaction 4 (vehicle-RSU-edge node-vehicle)

Features

- Sharing benchmarks with the Experiment Knowledge Service.
- Search benchmarking interactions, topologies or infrastructures.
- Recommend a deployment of blockchain artefacts into a model of application in MEC.
  - Load application model and preferences on metrics of quality.
  - Find most similar deployment pattern in the Experiment Knowledge Service.
  - Find a benchmark of the most similar deployment pattern, for which best results concerning the preferred quality metrics have been measured.
  - Return a topology of the benchmark.

Experiment Knowledge Service - Prototype

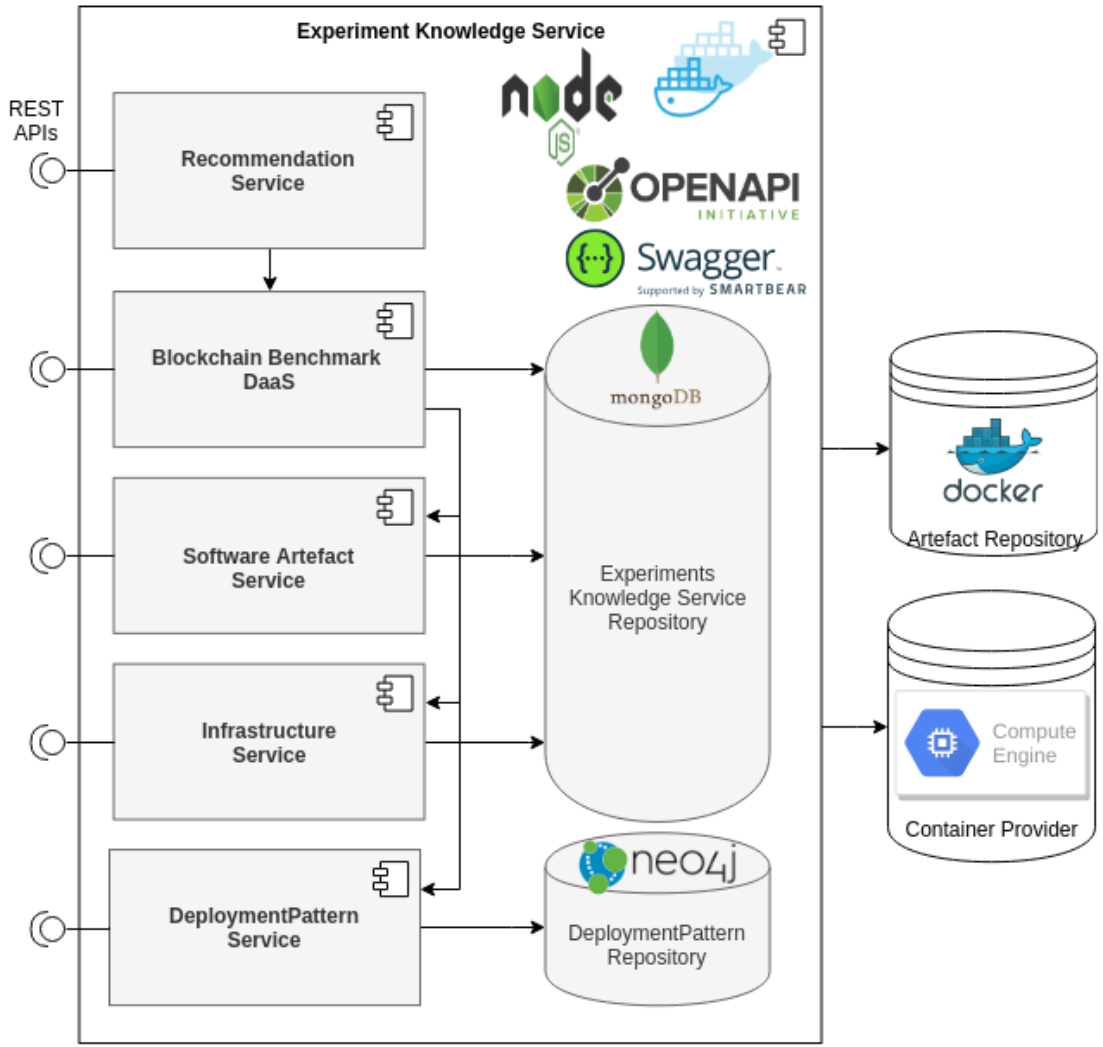


Figure: Prototype of Experiments Knowledge Service