

Fed4FIRE+ Experiment Report

Author
Fred Stefan

Date
04.02.2022

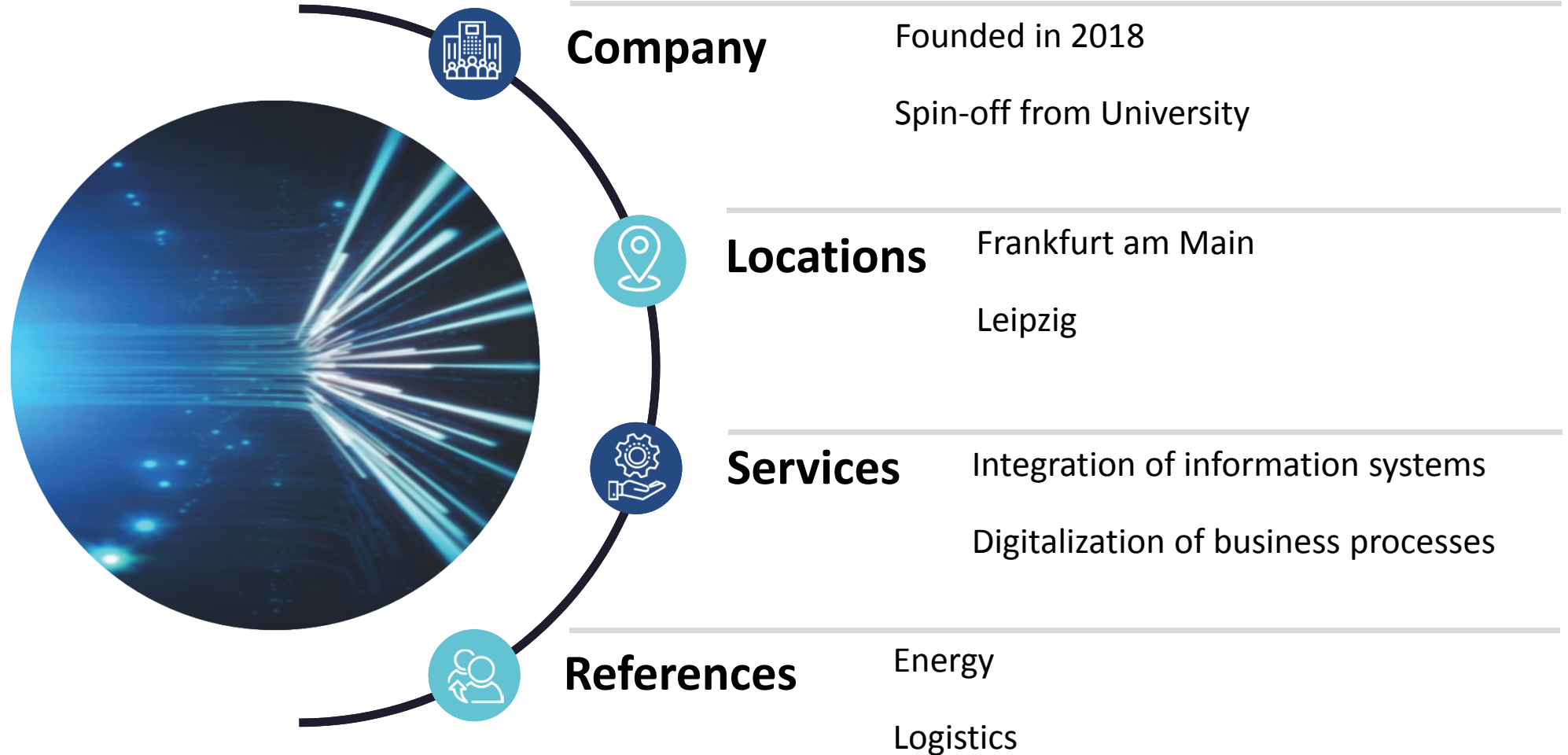


MANET4E

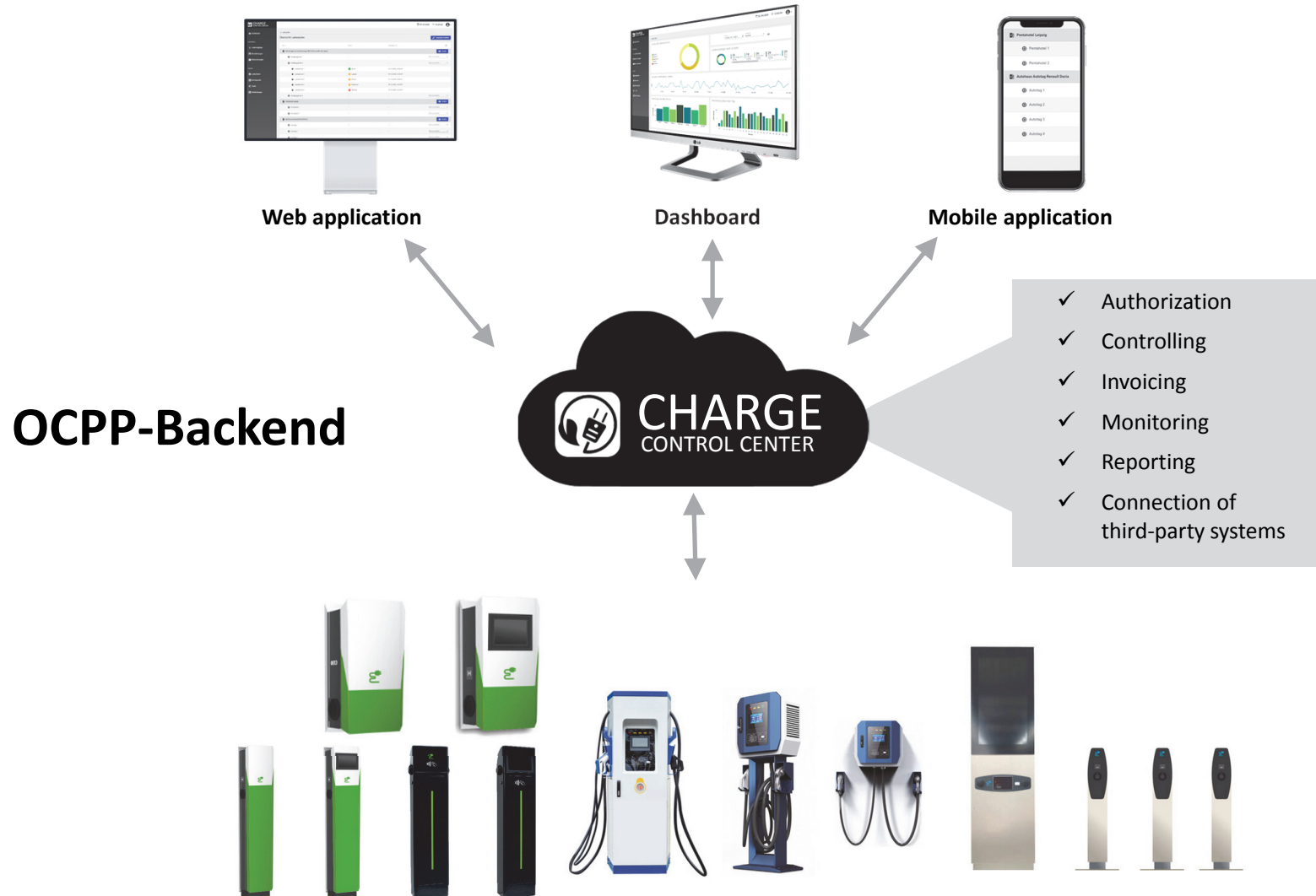
Mobile ad-hoc peer-to-peer network for self-organizing energy management systems

Stage 2 experiment

Runtime: 01.06.2021 - 30.09.2021



Background



Motivation

Today

Conventional power plants



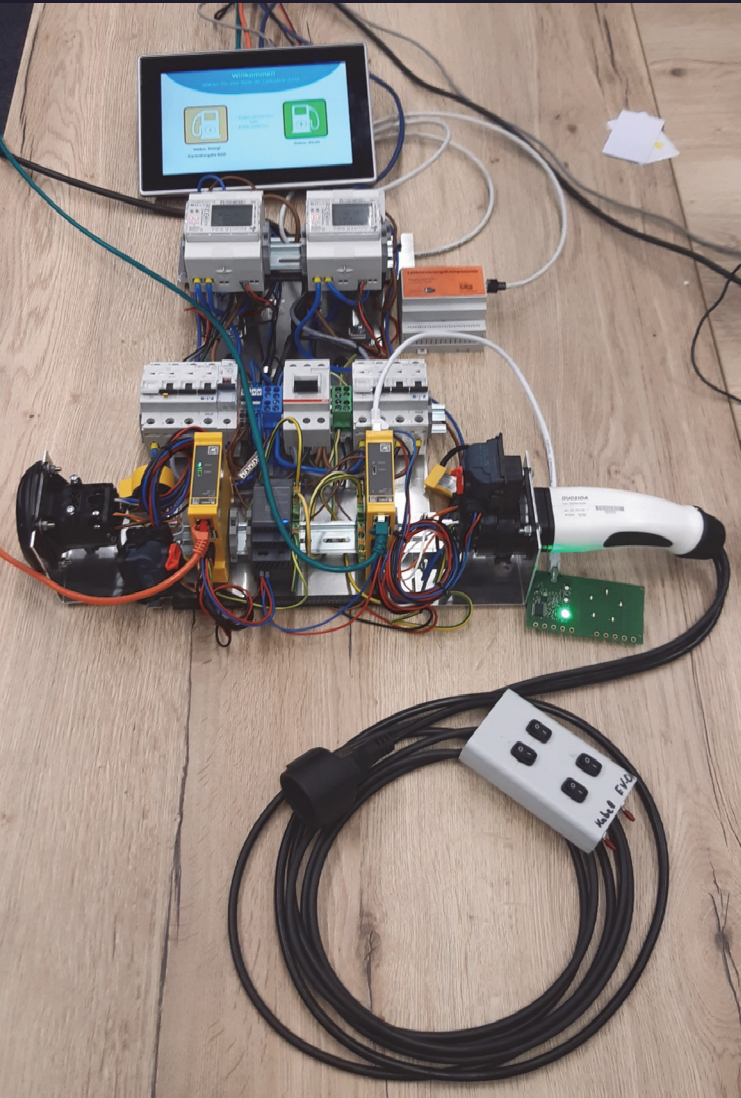
Tomorrow

New prosumer units

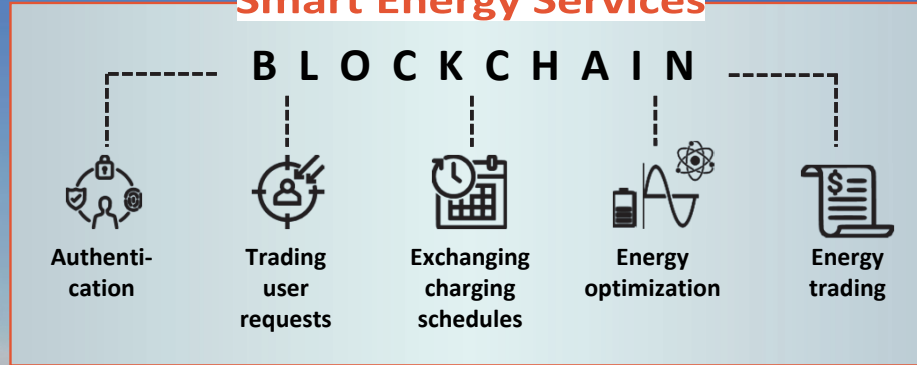


New consumer units





Smart Energy Services



Our objectives



New Product

The load management component should be offered as an independent product



Decentralized energy management

The component should enable an intelligent holistic decentralized energy management




New services for the grid of the future


It enables completely new services for the generation, trading and consumption of energy

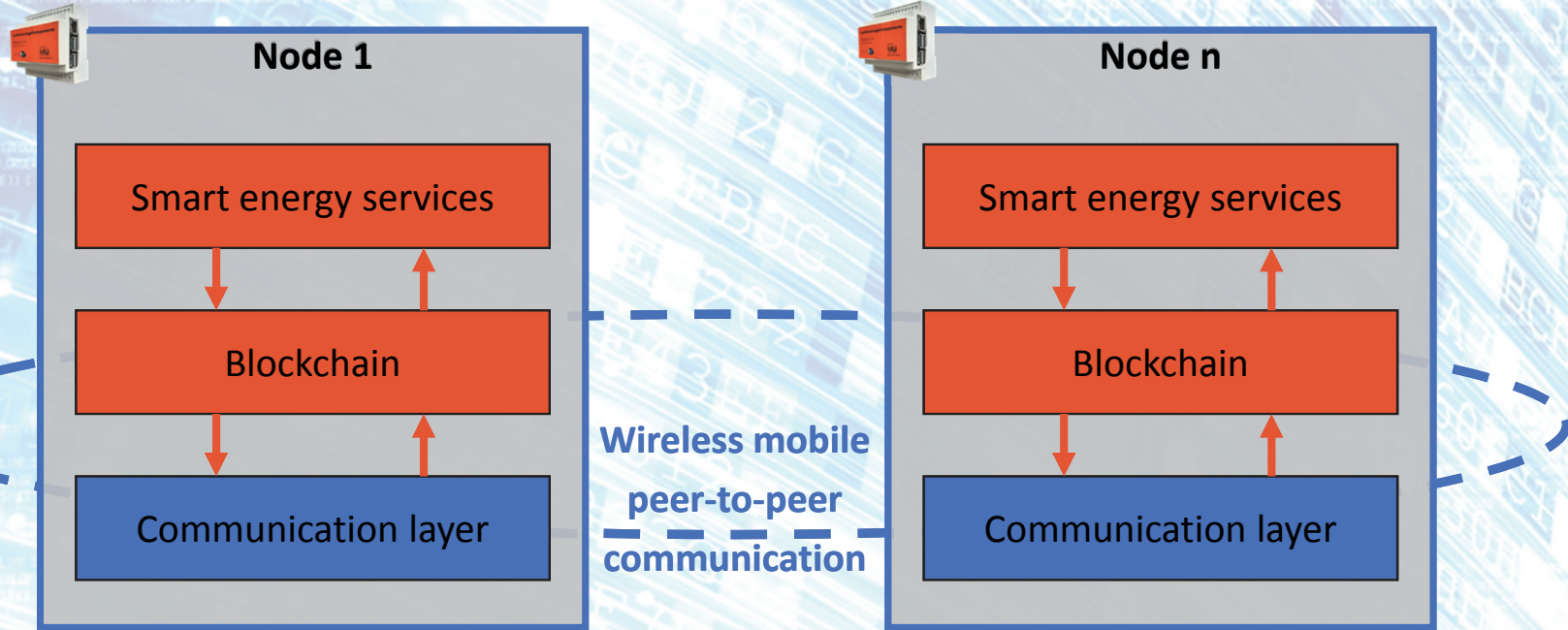


Easy to retrofit

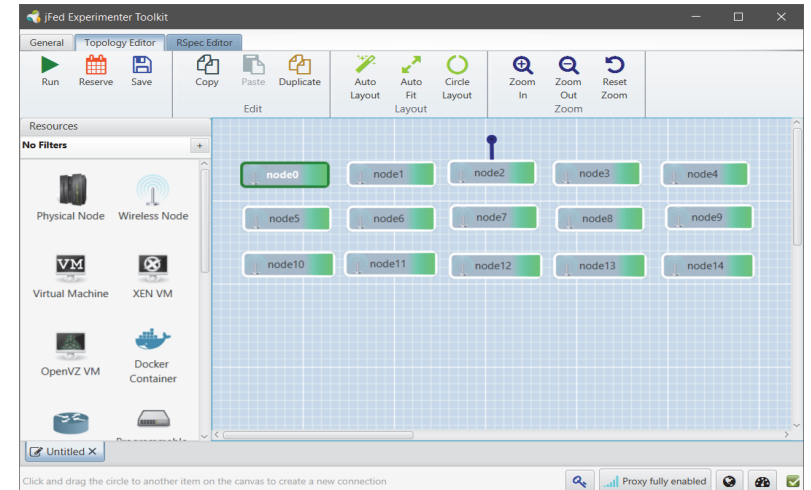
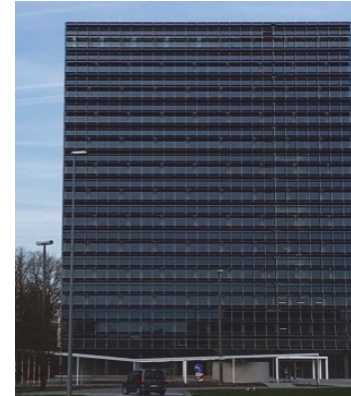
Former “stupid” charging stations, can be easily upgraded to “intelligent” charging stations

 Fed4FIRE+ Stage 2 experiment

 Fed4FIRE+ Stage 1 experiment



- **Testbed: w-iLab.1**
- Experiment resource specification with JFed
- **Node setup**
 - Intel NUC devices
 - Ubuntu 18.04.2 LTS
 - WIFI interface 802.11abgn
 - BATMAN batctl debian-2018.0-1 [batman-adv: 2017.4]



Methodical procedure

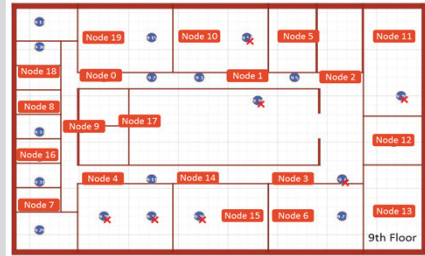


Implementation of different scenarios

Typical single-family home settlements



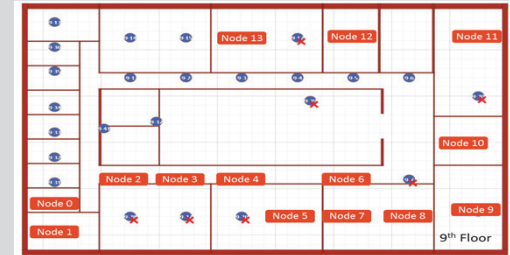
Mesh topology



Along the road



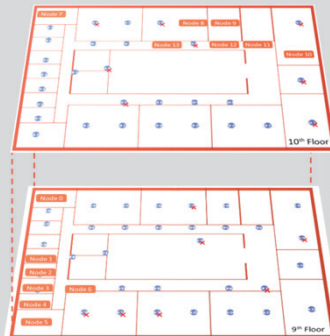
Bus topology



Companies and housing associations



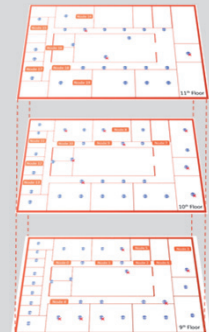
Hub-and-spoke topology



Multi storage parking



Multi-layer mesh topology



Our experiments

Experiment	Goal	Description
1	Verification of network topology, hops and transmission speeds	Ping and traceroute tests
2	Determination of the optimal block size for the blockchain	Investigation of the transmission times with incremental increase of the block size to be transferred
3	Determination of the optimal transmission interval of the blockchain blocks	Investigation of the transmission times for the identified block size for different intervals; Investigation of error rates
4	Determination of the optimal transmission interval for charging schedules	Investigation of the transmission times for 600-byte data blocks at different intervals; Investigation of error rates
5	Determination of the maximum blockchain size and optimal distribution of the blockchain	Investigation of the transmission speed and quality for larger files and different hops (when a node gets the complete blockchain); Investigation of error rates



Results

Experiment 1

Verification of network topology, hops and transmission speeds



Scenario 1



Scenario 2



Scenario 3



Scenario 4

Reachability



Ø Ping time

17 ms

26.6 ms

4 ms

16 ms

Min/Max Ping time

1.4 / 65 ms

1.4 / 80 ms

1.5 / 72 ms

1.5 / 80 ms

Ø Hop rate

1.8

2.3

1.4

1.8

Comments

communication between the two floors is only possible via nodes 0 and 7

Routing not over every node along the bus, but rather via the fastest connection

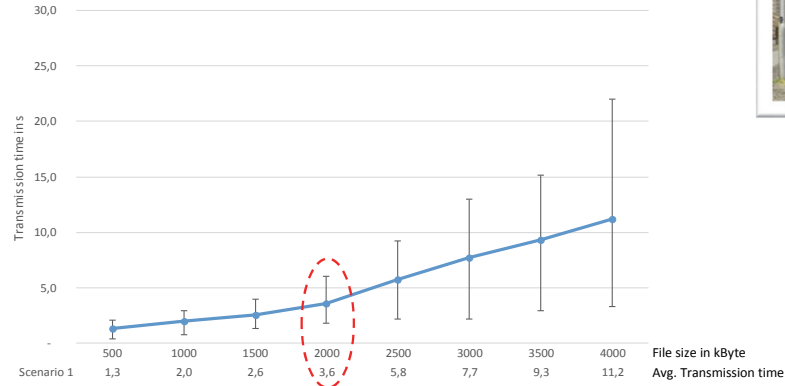
Interconnection between the floors over multiple nodes

Experiment 2

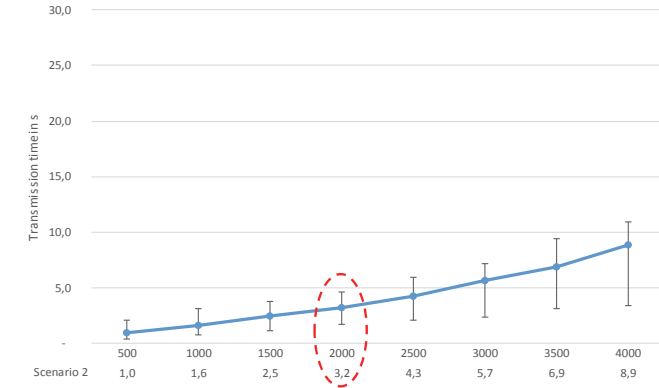
Determination of the optimal block size for the blockchain



Scenario 1



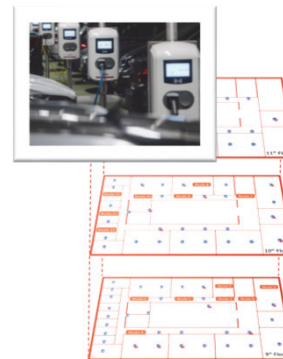
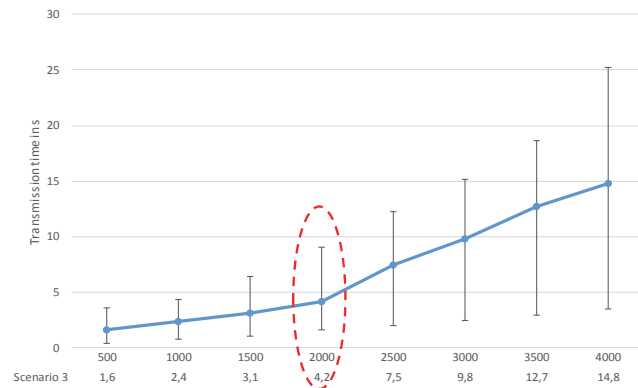
Scenario 3



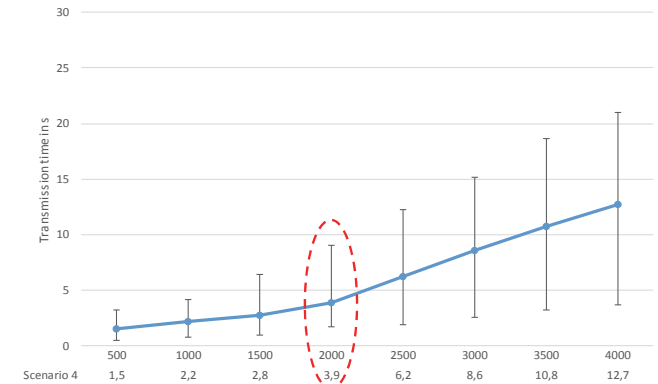
The optimal size of a block is 2 MByte

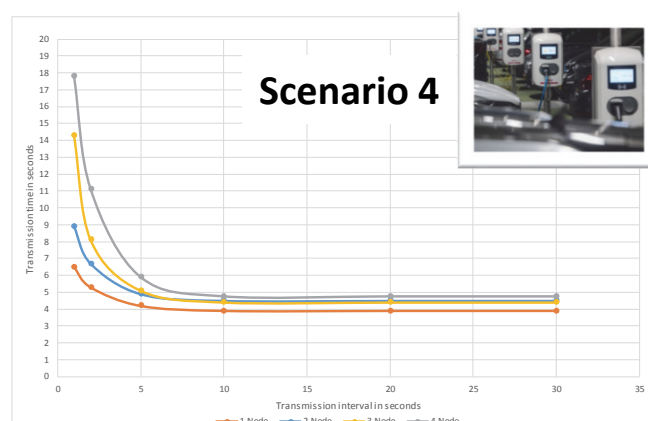
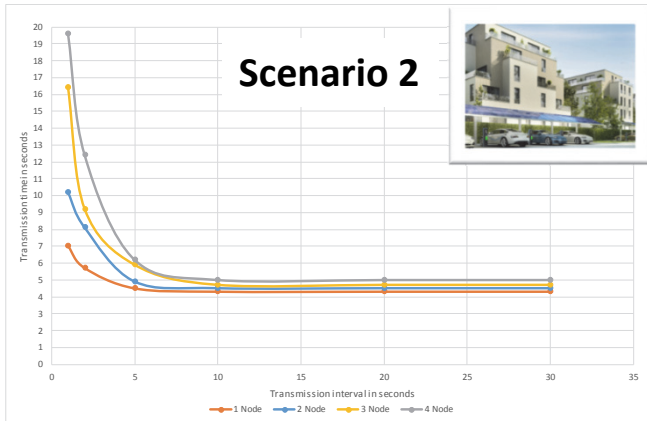
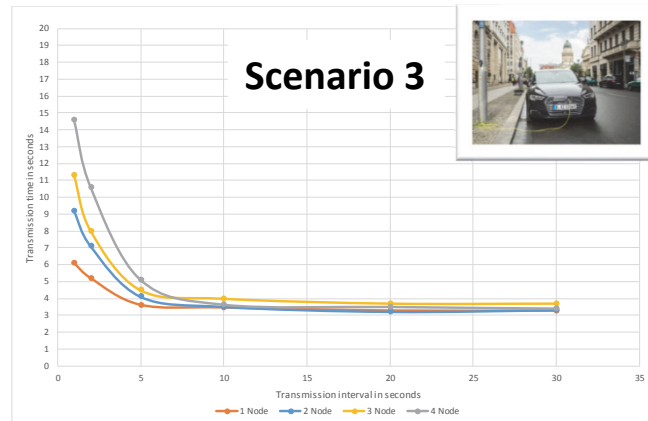
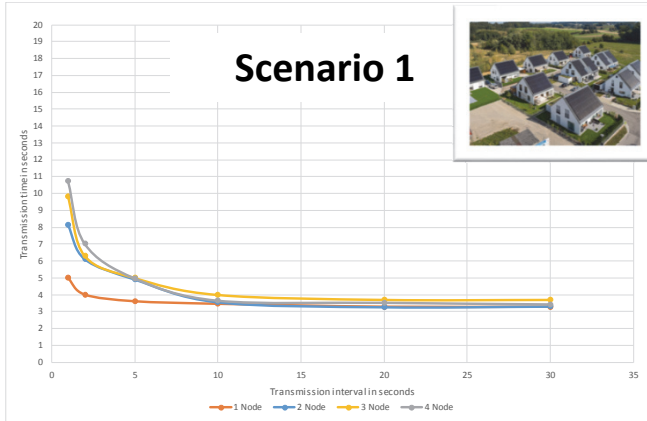


Scenario 2



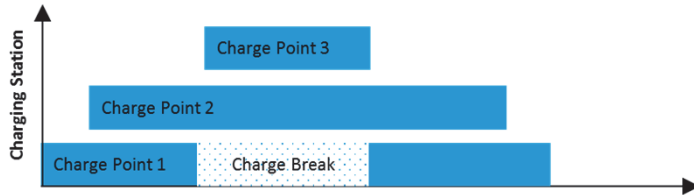
Scenario 4





- For transmission intervals **between 1 and 2 seconds**, there are very long transmission times
- Possible reason: high utilization of the mesh network
- Transmission **interval of 5 seconds** improved the transmission times
- The best transmission time is achieved from **10 seconds**
- Obviously it doesn't matter whether 1 node or 4 nodes transmit at the same time

Setup:



```
Ladeskule: 1
ScheduleEntry(time=0, currentOutput=20, remainingParkingDuration=10, currentLoadQuantity=20, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=1, currentOutput=10, remainingParkingDuration=9, currentLoadQuantity=40, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=2, currentOutput=3, remainingParkingDuration=8, currentLoadQuantity=50, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=3, currentOutput=3, remainingParkingDuration=7, currentLoadQuantity=53, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=4, currentOutput=3, remainingParkingDuration=6, currentLoadQuantity=56, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=5, currentOutput=3, remainingParkingDuration=5, currentLoadQuantity=59, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=6, currentOutput=10, remainingParkingDuration=4, currentLoadQuantity=62, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=7, currentOutput=10, remainingParkingDuration=3, currentLoadQuantity=72, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=8, currentOutput=10, remainingParkingDuration=2, currentLoadQuantity=82, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=9, currentOutput=10, remainingParkingDuration=1, currentLoadQuantity=92, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=10, currentOutput=0, remainingParkingDuration=0, currentLoadQuantity=100, totalLoadQuantity=100, scheduleState=SCHEDULED_FINISHED)
Ladeskule: 2
ScheduleEntry(time=1, currentOutput=10, remainingParkingDuration=3, currentLoadQuantity=70, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=2, currentOutput=10, remainingParkingDuration=2, currentLoadQuantity=80, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=3, currentOutput=10, remainingParkingDuration=1, currentLoadQuantity=90, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=4, currentOutput=0, remainingParkingDuration=0, currentLoadQuantity=100, totalLoadQuantity=100, scheduleState=SCHEDULED_FINISHED)
Ladeskule: 3
ScheduleEntry(time=2, currentOutput=7, remainingParkingDuration=3, currentLoadQuantity=80, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=3, currentOutput=7, remainingParkingDuration=2, currentLoadQuantity=87, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=4, currentOutput=7, remainingParkingDuration=1, currentLoadQuantity=94, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=5, currentOutput=0, remainingParkingDuration=0, currentLoadQuantity=100, totalLoadQuantity=100, scheduleState=SCHEDULED_FINISHED)
```

Average Size of a charging plan: 6 Kbytes

Results:

- For each szenario transmission times <1 second

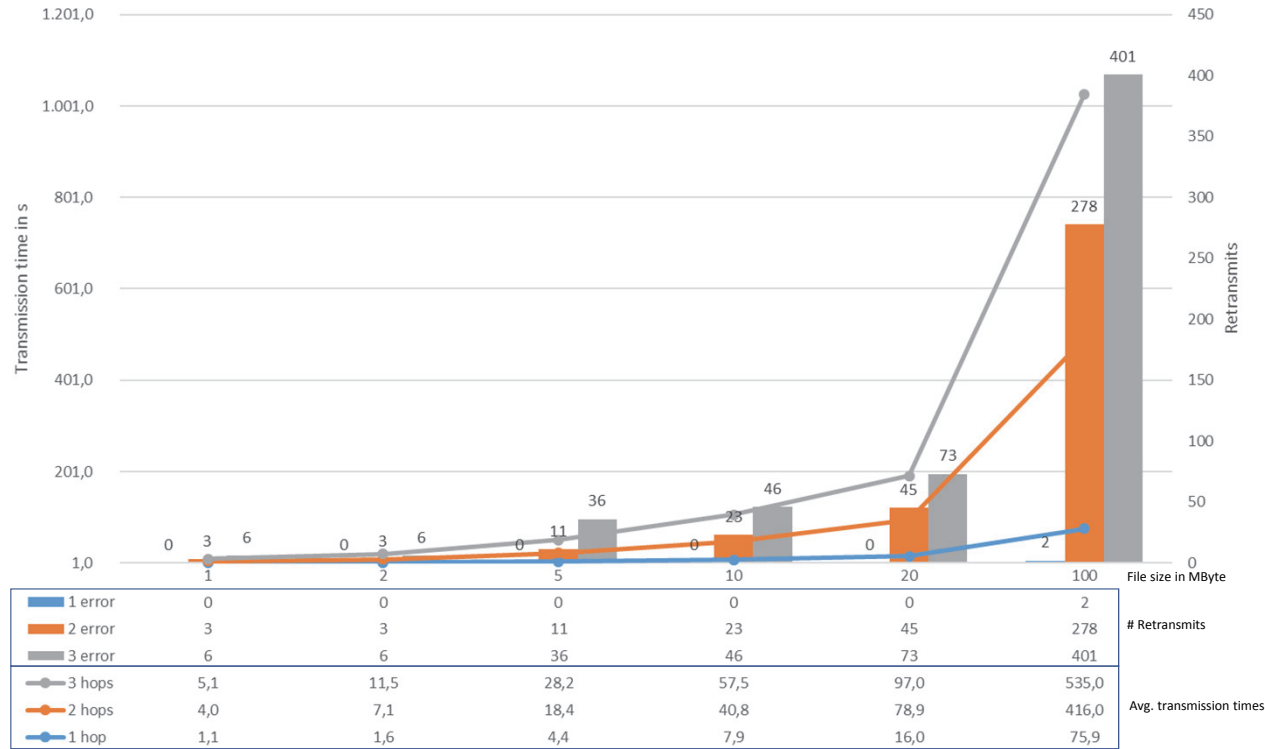
- Worst case szenario 3 and 4



- Worst transfer speed: 350 KBit/s

Experiment 5

Determination of the maximum blockchain size and optimal distribution of the blockchain



Results:

- Optimum 20 MByte
- 20 MByte $\hat{=}$ 10 blocks of 2 MB each with a total of approx. 3,000 charging schedules

Summary

- Different network topologies were set up with up to 20 nodes
- We examined our peer-to-peer communication layer
- The optimal size of a block is 2 MByte
- The optimal total size of the blockchain is 20 MByte
- Optimal transmission interval is 10 seconds

Summary

- **Fed4FIRE+** is a very useful environment for experiments
- **Hardware as a service** is very convenient (saves time and money)
- We could investigate the scalability of our approach
 - From 3 nodes in our office up to 20 nodes with Fed4FIRE
 - The w-iLab.1 office building was ideal to emulate a parking garage
- We derived some further important requirements for our solution

Fed4FIRE+ Experiment Report

Thank you.

Author
Fred Stefan

Date
04.02.2022

