



swether

Electric Switch with Ethereum Blockchain Support

Why blockchains?



- Trusted transactions
- Removes (trusted) intermediaries
 - replaced by cryptographic mechanisms in a decentralized system
- Cryptocurrencies: bitcoin, ether, ...
 - > micropayments
- Autonomous M-2-M transactions
- Monetization of IoT
- Application development
 - **off-chain:** BC client exposes API for Web/mobile/etc application development
 - **on-chain:** Smart contract logic (not in Bitcoin)
- Smart contract
 - Programming code
 - Deployed to the blockchain
 - Executed in the blockchain (in trusted manner)
- Smart contract use cases:
 - Cryptocurrencies
 - Auctions, bidding
 - DAO
 - ...



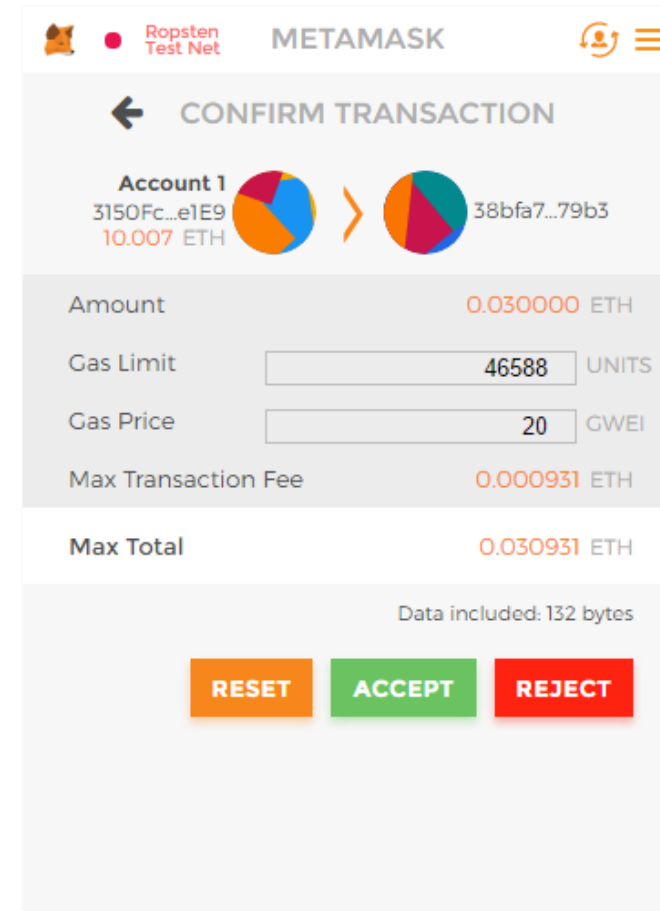
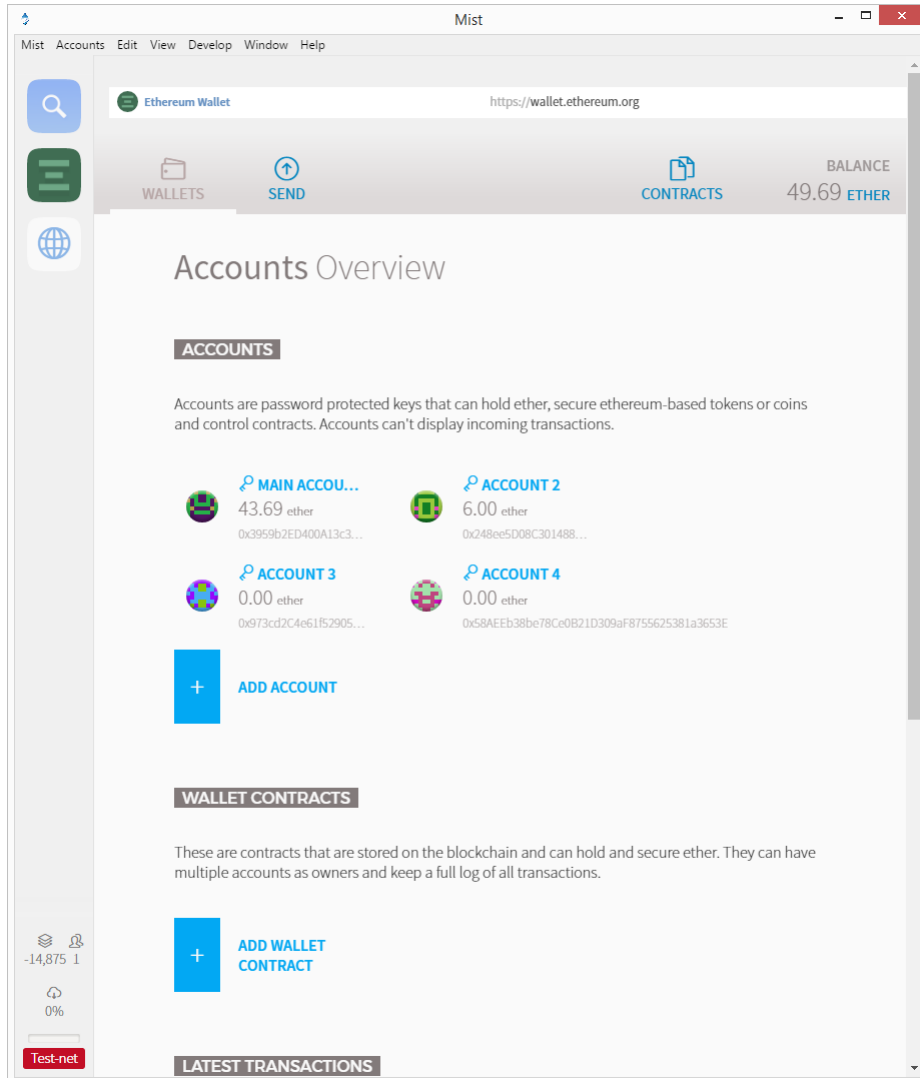
Ethereum

- Bitcoin, Ethereum
- Ethereum applications
 - Financial
 - Semi-financial
 - Non-financial
- Ethereum network
 - Mainnet
 - Testnet (Ropsten)
- Accounts
 - Externally owned
 - Contract
- Ethereum smart contracts
 - „Autonomous agents“
 - Smart contract code
 - Messages, events
- Ethereum virtual machine
- Solidity smart contract
 - Programming
 - Compiling
 - Storing in a chain
- geth – client for Ethereum network

Bitcoin vs. Ethereum vs. Hyperledger Fabric

	Bitcoin	Ethereum	Hyperledger Fabric
Native cryptocurrency	Yes	 Yes	No
Distributed applications	No (very limited)	 Yes – smart contracts	Yes - chaincode
Smart contracts (business logic)	-	Solidity	Go, Java (executed in containers)
Consensus algorithm	PoW – Proof of Work	PoW – Proof of Work (PoS foreseen) – Proof of stake	PBFT - Practical Byzantine fault tolerance
Anonymous accounts	Yes	Yes	No (permissioned network)
Network	Public	Public or permissioned	Permissioned
Suitable for IoT	No	Yes	Yes
State channels	Yes (Lightning)	Yes (Raiden)	Not known

Wallets and browsers: Mist, Metamask, etc.



geth – Ethereum client

- Ethereum client
 - geth is responsible for running Ethereum protocols and thus the entire communication with the blockchain
 - management of blocks and transactions
 - management of peers and network
 - monitoring of chain status
 - managing Ethereum accounts
 - mining
 - ...
 - exposes web3.js API to applications
- web3.js API
 - Through the web3.js API application can monitor status of the client and monitor or create transactions

```
Command Prompt - geth attach
Z:\>geth attach
Welcome to the Geth JavaScript console!

instance: Geth/v1.6.5-stable-cf87713d/windows-amd64/go1.8.3
coinbase: 0x3959b2ed400a13c3920d7e457120563c1cc74e91
at block: 1105346 (Mon, 12 Jun 2017 18:43:33 CEST)
  datadir: C:\Users\matevzp\AppData\Roaming\Ethereum
  modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> net.peerCount
1
> eth.syncing
false
> eth.blockNumber
1105349
> admin.nodeInfo
{
  enode: "enode://f70a7bfcdf21c56766bbf25abdabce11b1ad27ff51eb3c0029e130aeffa10b9d38c561dd5cb7fe5f8c15568220",
  id: "f70a7bfcdf21c56766bbf25abdabce11b1ad27ff51eb3c0029e130aeffa10b9d38c561dd5cb7fe5f8c15568220",
  ip: ":",
  listenAddr: "[*]:30303",
  name: "Geth/v1.6.5-stable-cf87713d/windows-amd64/go1.8.3",
  ports: {
    discovery: 30303,
    listener: 30303
  },
  protocols: {
    eth: {
      difficulty: 740110738139515,
      genesis: "0x41941023680923e0fe4d74a34bdac8141f2540e3ae90623718e47d66d1ca4a2d",
      head: "0x30a6893906f111f9ed8f55923510440c873de5864742482217e132a09eee976a",
      network: 3
    }
  }
}
> admin.peers
[
  {
    caps: ["eth/62", "eth/63"],
    id: "20c9ad97c081d63397d7b685a412227a40e23c8bdc6688c6f37e97cfbc22d2b4d1db1510d8f61e6a8866ad7f0",
    name: "Geth/v1.6.0-unstable-6d038e76/linux/go1.7.3",
    network: {
      localAddress: "10.0.3.105:59332",
      remoteAddress: "52.169.14.227:30303"
    },
    protocols: {
      eth: {
        difficulty: 740100207977488,
        head: "0xbfc7b08d3794c22588297dbbb6122373cac5f7e92253e9b5e533282feb2dc8c0",
        version: 63
      }
    }
  }
]
```

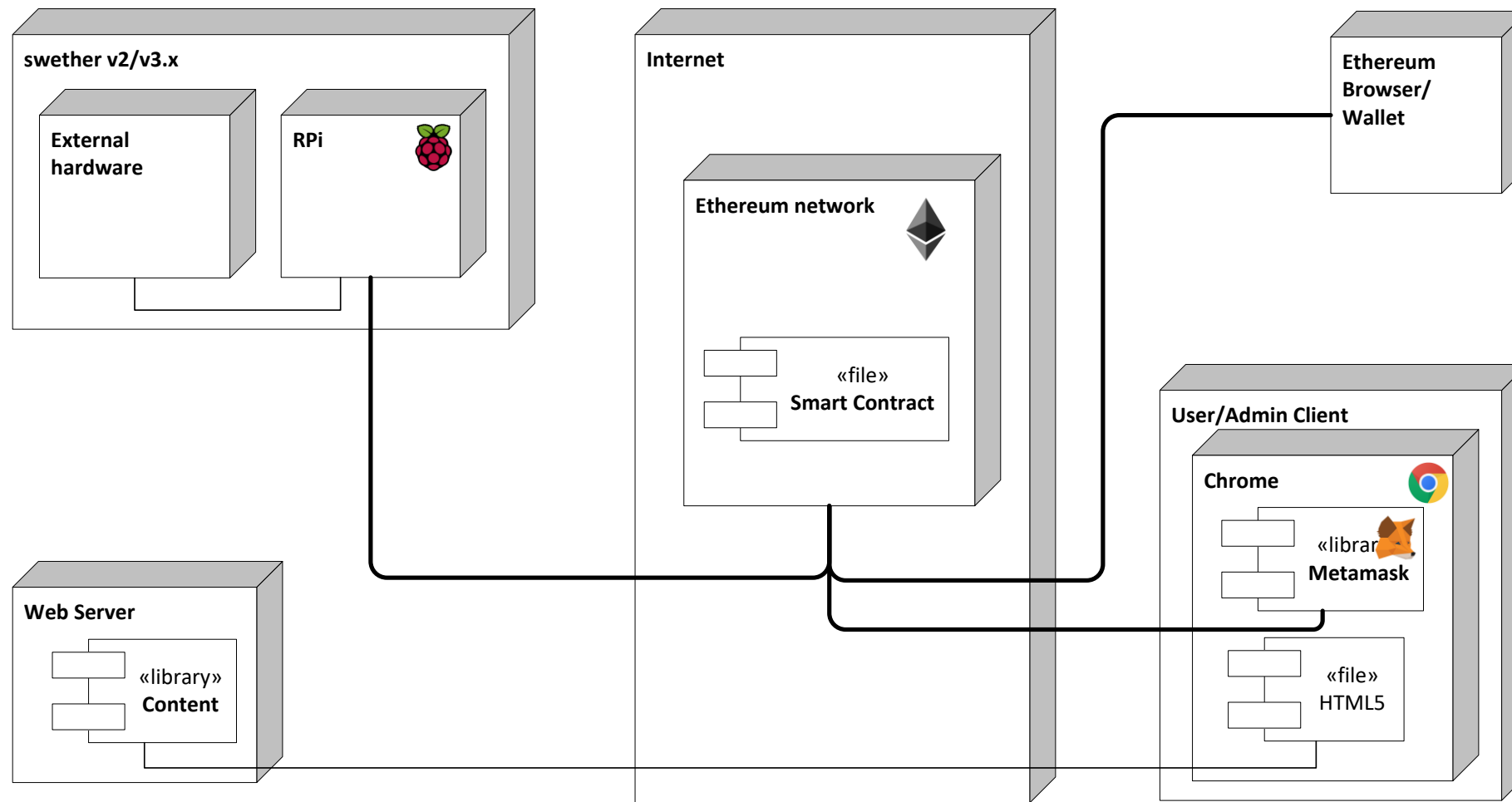

Swether - what is it?

- Blockchain based end-to-end prototype system comprised of IoT devices, decentralized applications (DAPPs) and user interfaces
- Enables control of electric switch via Ethereum blockchain transactions. The user can reserve electric charging slot and pay for consumed time/energy via Ethereum network. It is a practical application of blockchain technologies in IoT devices.
- Prototype system consists of:
 - **Smart contract**: deployed in Ethereum network to execute DAPP logic
 - **User and Admin Web applications**: to be run in Ethereum compliant browsers like Mist or Chrome+Metamask plugin
 - Swether v2/v3.1: **IoT device** that intercepts events from Ethereum network and toggles its state accordingly
- The concept can be easily extended to e.g. charging of electric vehicles, arbitrary control of IoT devices, device-to-device transactions and interactions.

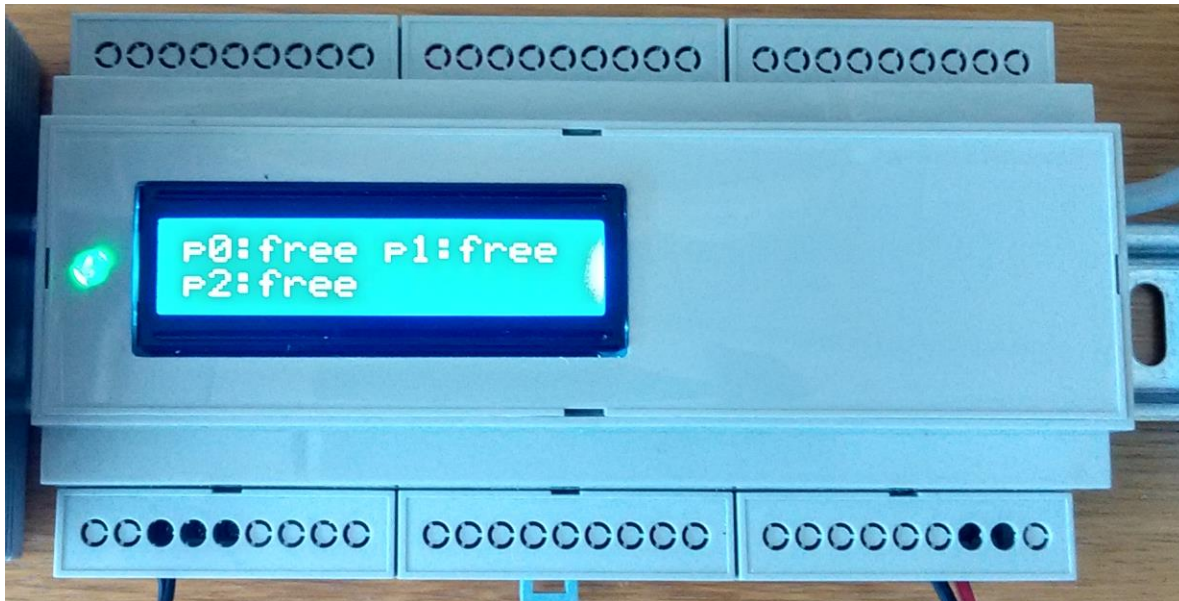
Why swether?

- End-2-End: IoT + Blockchain solution
 - no "dirty details" are skipped
- Blockchain in IoT
- Use cases:
 - Smart grid
 - EV charging stations
 - Utility metering
- Extensible for other use cases

Deployment



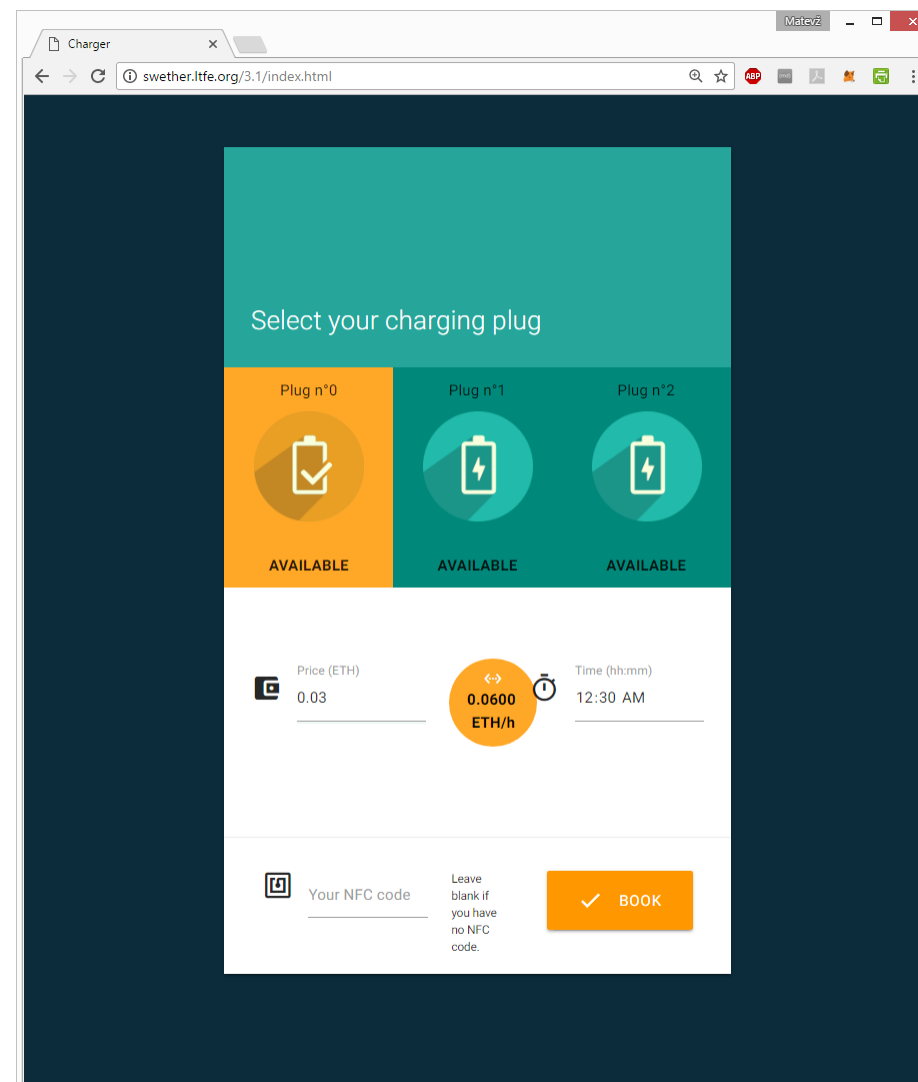
Swether v3.1



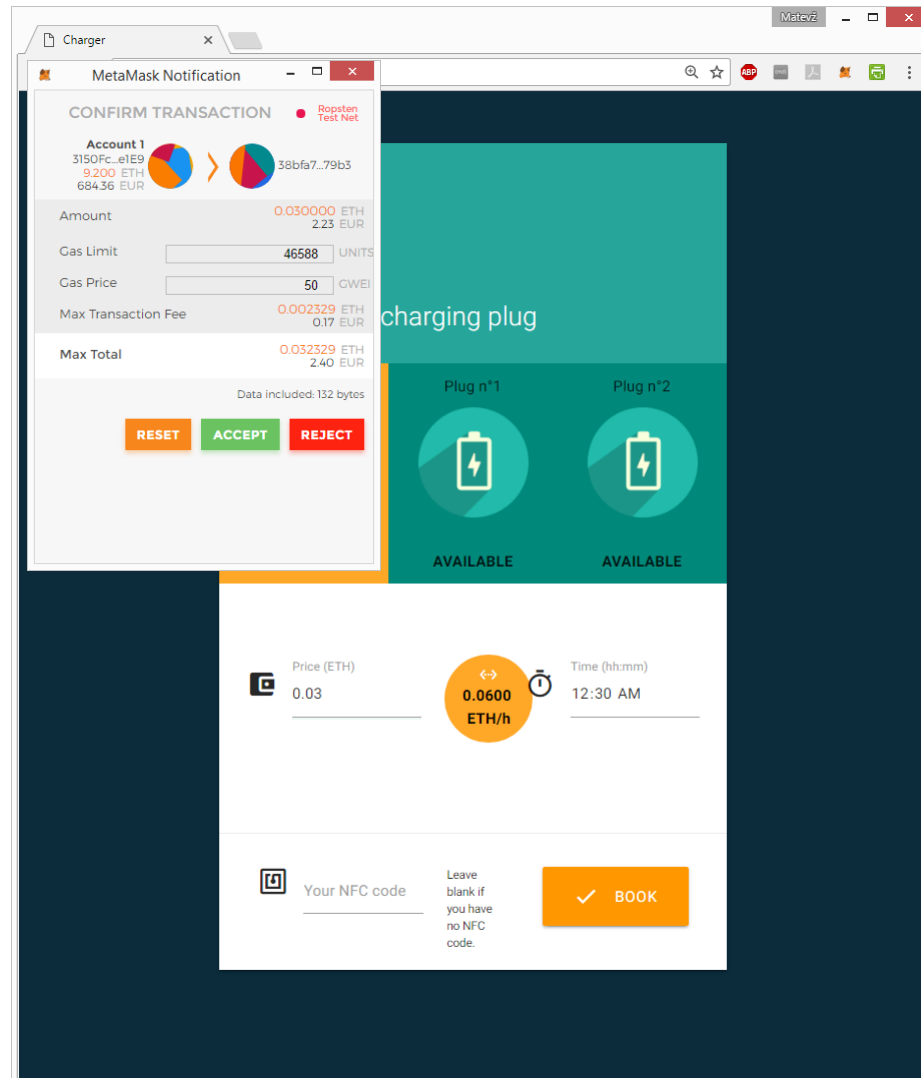
- Ethereum compliant IoT device
- 3x power relays – 230V, 10 A
- LCD
 - device status
 - blockchain status
 - current device occupation indication
- NFC
 - additional level of authentication

Review the status and make a reservation

1. Open the Web page in Ethereum compliant browser
2. Review the status of the charging device and reserve a charging slot



Confirm the payment and send transaction



3. Confirm the Ethereum transaction and payment; the transaction is sent to blockchain for validation
4. Once valid the transaction triggers an event, which is intercepted by Swether device
5. Swether device toggles charging slot according to the data in transaction

Swether roadmap

- v3.3
 - Powermeters
 - Device becomes active BC network node with its own Ethereum address and autonomously communicates with the contract.
- v4
 - Elaborated smart contract (use of tokens, reservation system, etc).
- v5
 - State channels: Raiden protocol/network

Thanks!

matevz.pustisek@fe.uni-lj.si