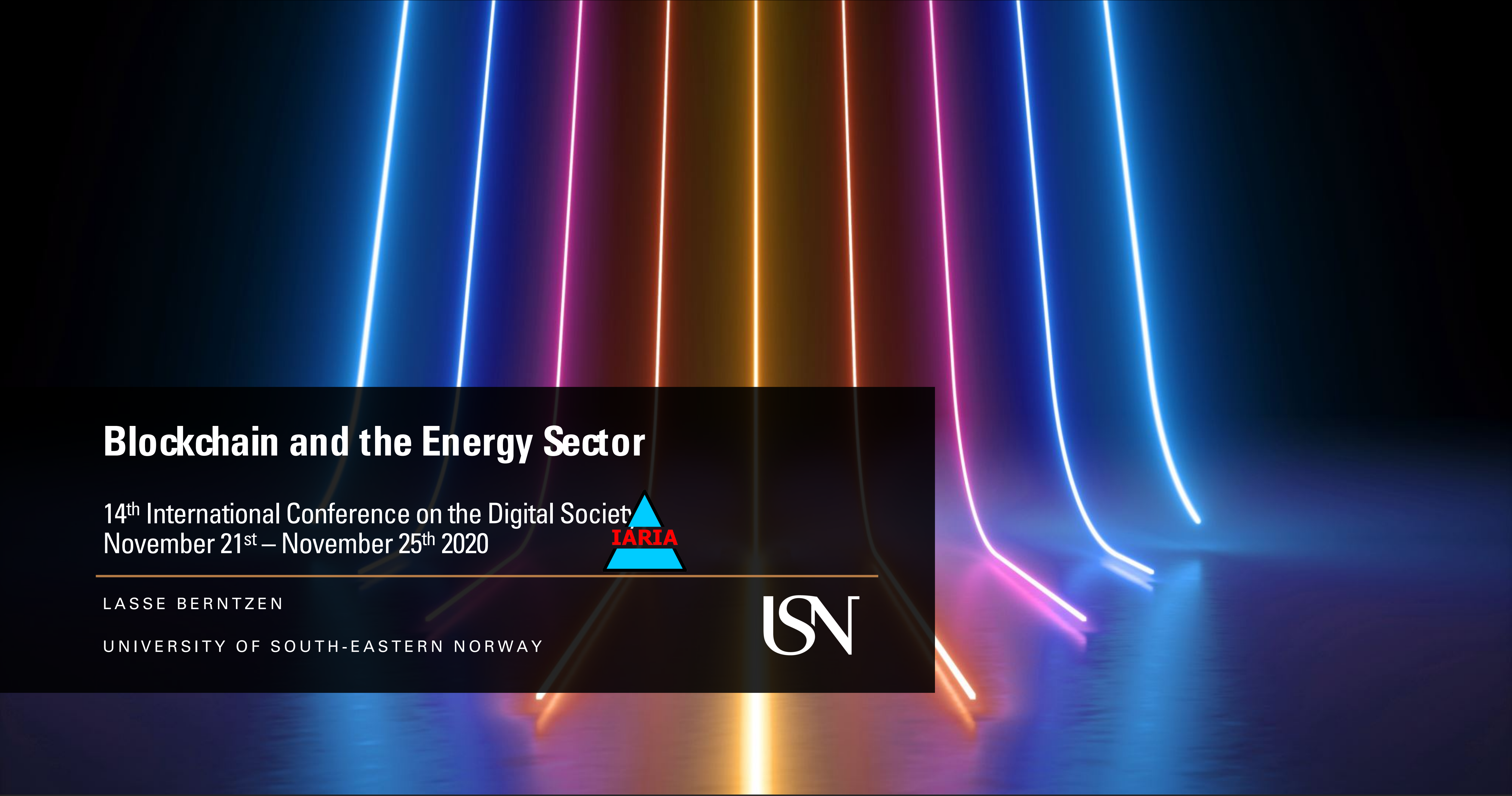


Blockchain and the Energy Sector

14th International Conference on the Digital Society
November 21st – November 25th 2020

IARIA

LASSE BERNTZEN

UNIVERSITY OF SOUTH-EASTERN NORWAY

USN

About myself

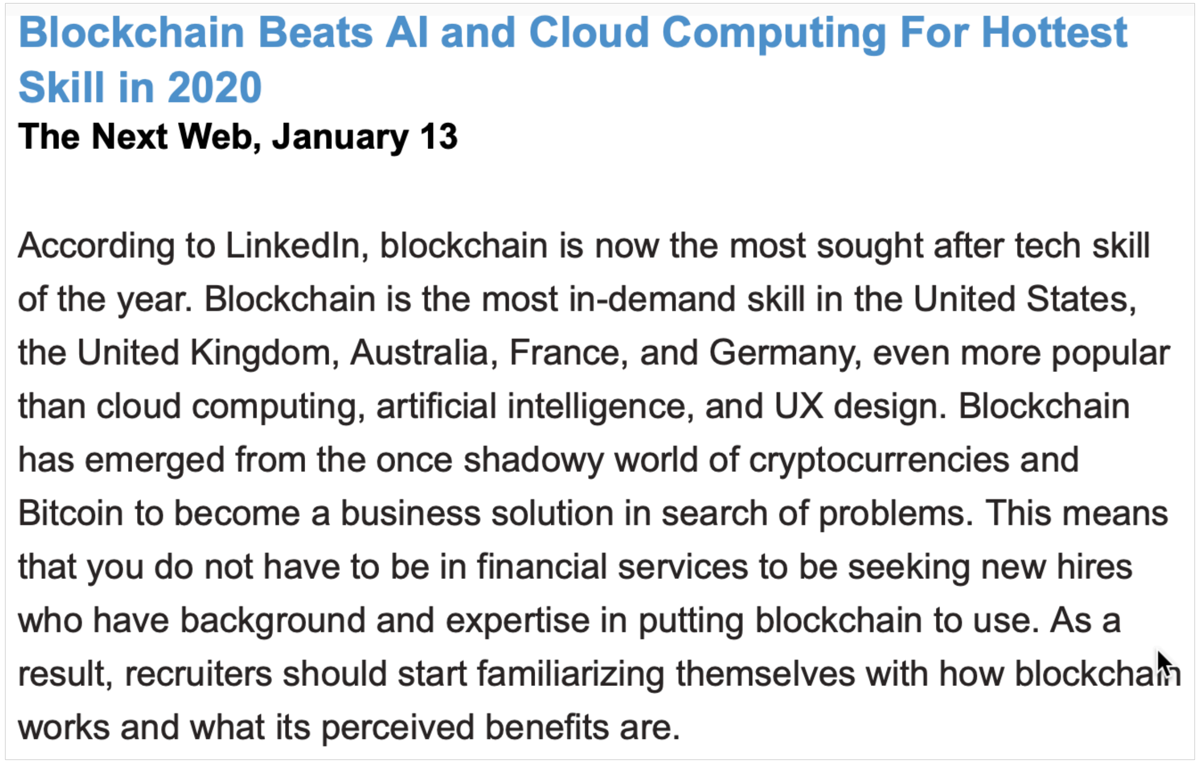
Professor in Information Systems

Research on electronic participation since 2003

Smart cities research group

Teaching IoT, business analytics

IARIA Fellow



Blockchain Beats AI and Cloud Computing For Hottest Skill in 2020

The Next Web, January 13

According to LinkedIn, blockchain is now the most sought after tech skill of the year. Blockchain is the most in-demand skill in the United States, the United Kingdom, Australia, France, and Germany, even more popular than cloud computing, artificial intelligence, and UX design. Blockchain has emerged from the once shadowy world of cryptocurrencies and Bitcoin to become a business solution in search of problems. This means that you do not have to be in financial services to be seeking new hires who have background and expertise in putting blockchain to use. As a result, recruiters should start familiarizing themselves with how blockchain works and what its perceived benefits are.

Riding the Hype ☺

Motivation: The Smart-MLA Project

University of South-Eastern Norway (USN) is partner in the project “Smart Multi-layer Aggregator” (Smart-MLA) funded by the ERANET SmartGrid program.

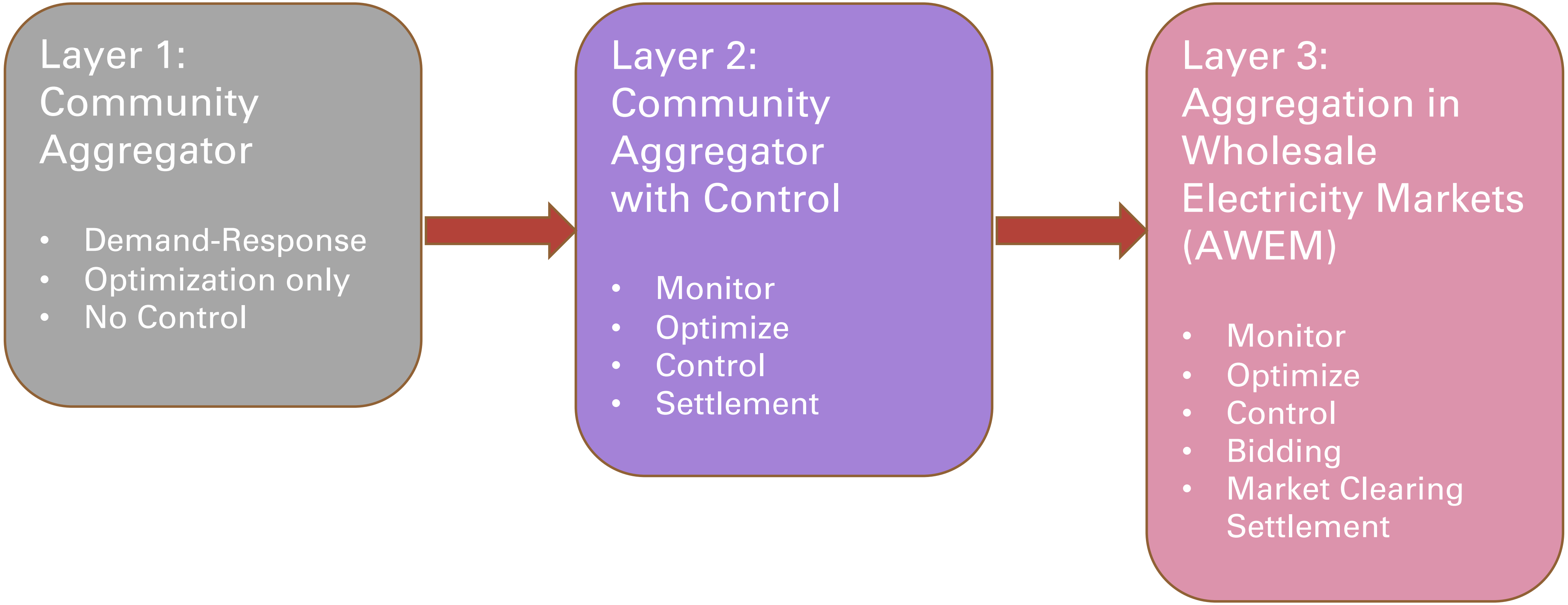
USN is working on two specific tasks:

1. Investigate the use of Blockchain for settlements between energy producers and energy consumers and smart contracts for energy optimization.
2. Study emerging business models and platform adoption in the energy sector, and discuss possible implementation of business models in the project.

Content

- Motivation: Smart-MLA Project
- Blockchains and the energy sector
- A short introduction to blockchains
- Ethereum and smart contracts
- A short introduction to smart grids
- Smart grids and blockchain technology
- Implications for the Smart-MLA project

Smart Multi-layer Aggregator Concept



Layer 1: Community Aggregator

- Demand-Response
- Optimization only
- No Control

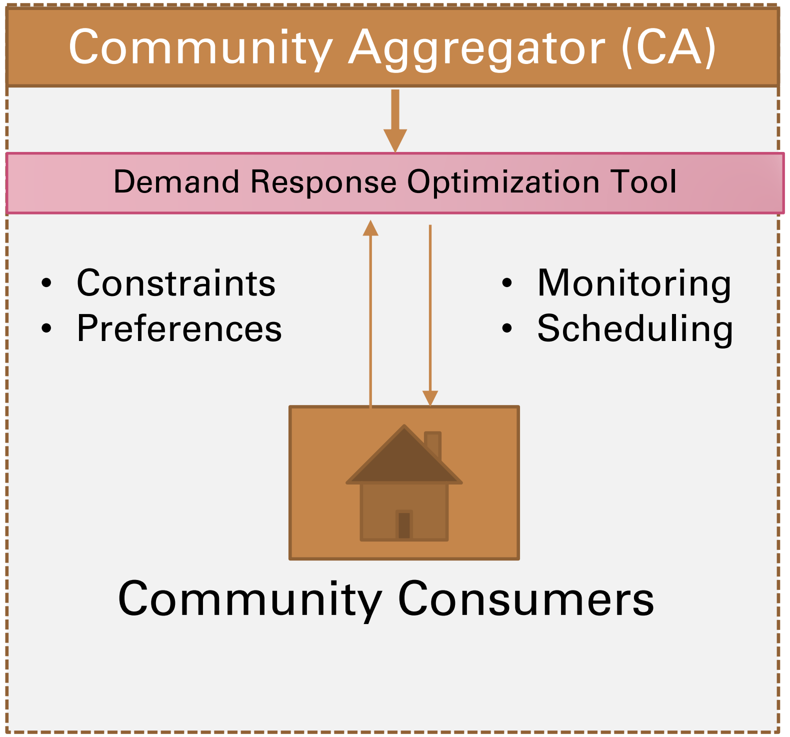
Layer 2: Community Aggregator with Control

- Monitor
- Optimize
- Control
- Settlement

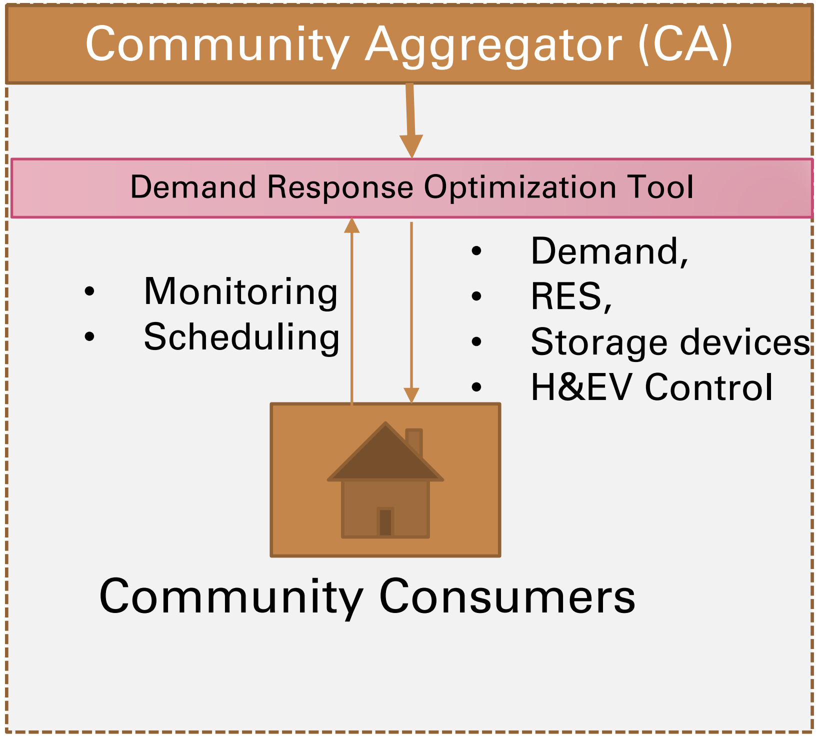
Layer 3: Aggregation in Wholesale Electricity Markets (AWEM)

- Monitor
- Optimize
- Control
- Bidding
- Market Clearing Settlement

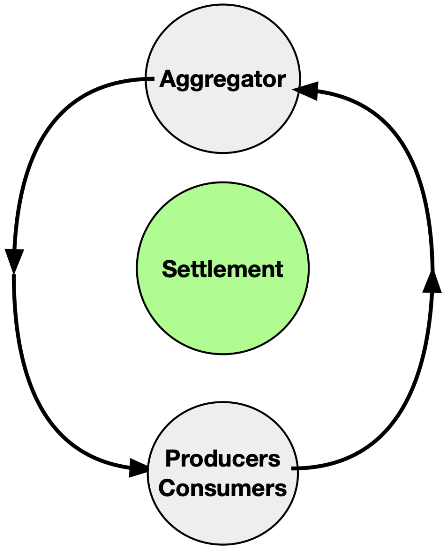
Smart Multi-layer Aggregator Concept



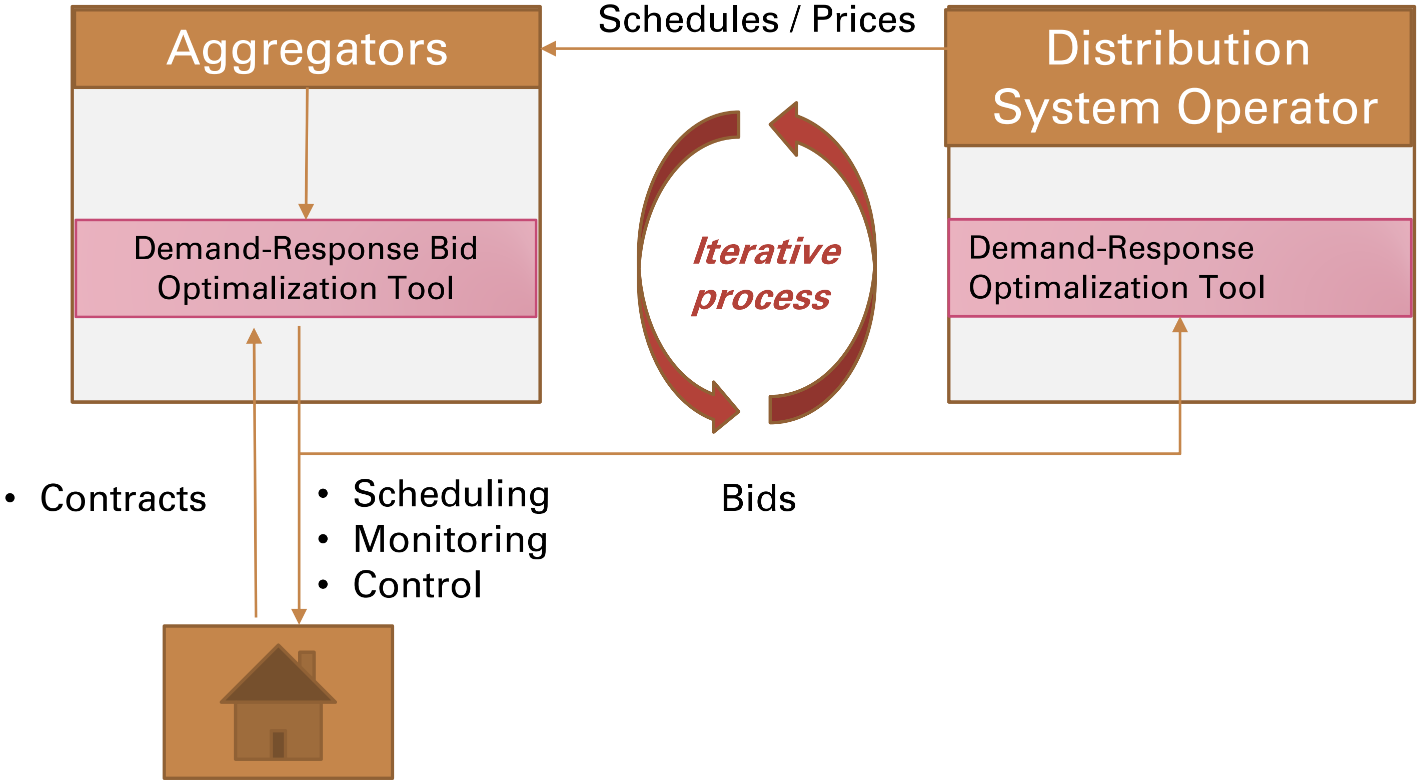
Layer 1: Community Aggregator
(DR Optimization only / No control



Layer 2: Community Aggregator with Control
(Monitor / Optimize / Control / Settlement)



Smart Multi-layer Aggregator Concept



Schedules / Prices

Aggregators

Demand-Response Bid Optimalization Tool

Iterative process

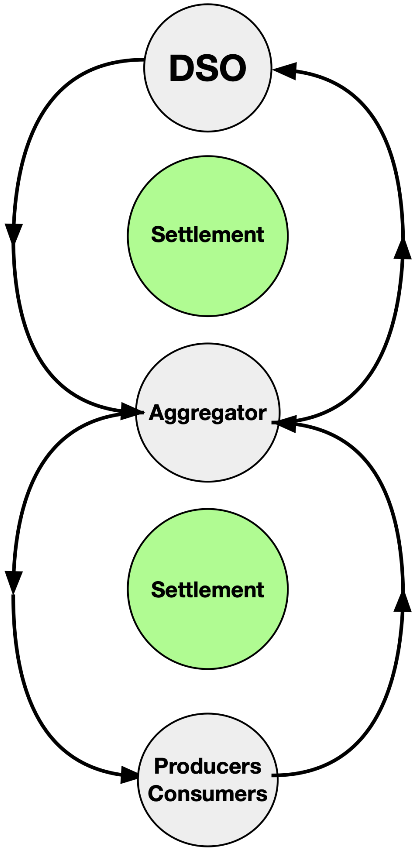
Distribution System Operator

Demand-Response Optimalization Tool

- Contracts

- Scheduling
- Monitoring
- Control

Bids



DSO

Settlement

Aggregator

Settlement

Producers Consumers

Layer 3: Aggregation in Wholesale Electricity Markets – AWEM
(Monitor / Optimize / Control / Bidding / Market Clearing / Settlement)

Blockchain and the Energy Sector

Blockchain and the Energy Sector

Blockchains require huge amounts of power, since validation of transactions is based on complex algorithms to solve mathematical problems.

The work involved in validating transactions is what makes blockchains almost unhackable.

Therefore, blockchain is an important consumer of energy.

Blockchain and the Energy Sector

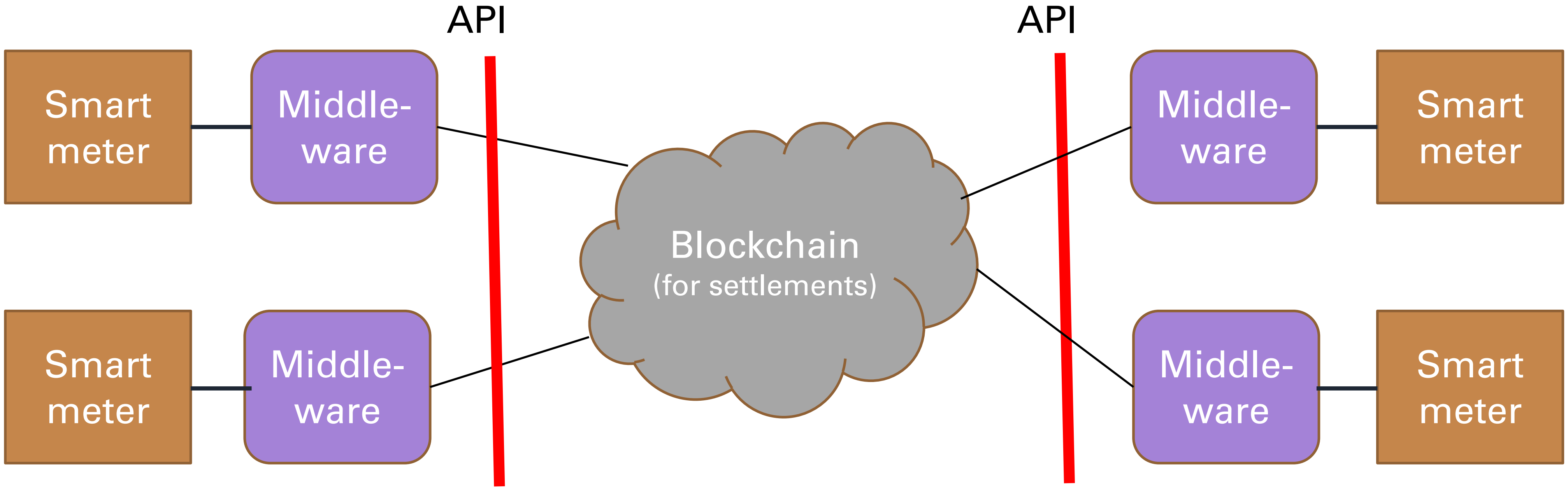
Blockchain can also be used to handle settlements between producers and consumers.

In a local smart grid, the consumers can also be producers, called prosumers.

The prosumers need to settle transactions between themselves.

Blockchain has no central authority, it is decentralized

Blockchain for Settlements



The API is described in another slide

Blockchain and the Energy Sector

Blockchain can also be used for smart contracts where tasks and transactions are executed when certain requirements are met.

One example is Electric Vehicle charging.

Charging when the price of energy is lowest within a time period (e.g., before 7 a.m. next morning).

A Short Introduction to Blockchain

The Blockchain

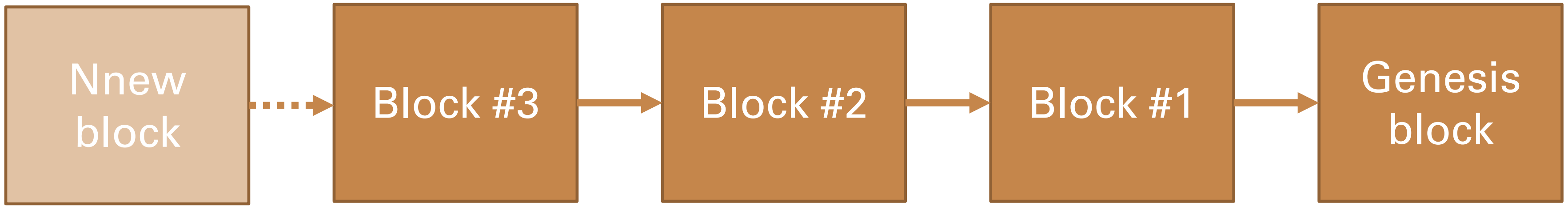
Blockchain is a *decentralized*, *immutable distributed ledger*. Cryptographic techniques make sure that it is not possible to change the content of an entry when it is put on the blockchain.

All entries are recorded in blocks that are distributed to many computers. If one or more computers fails, the data is still obtainable from the network.

If there should be an inconsistency, the majority will win.

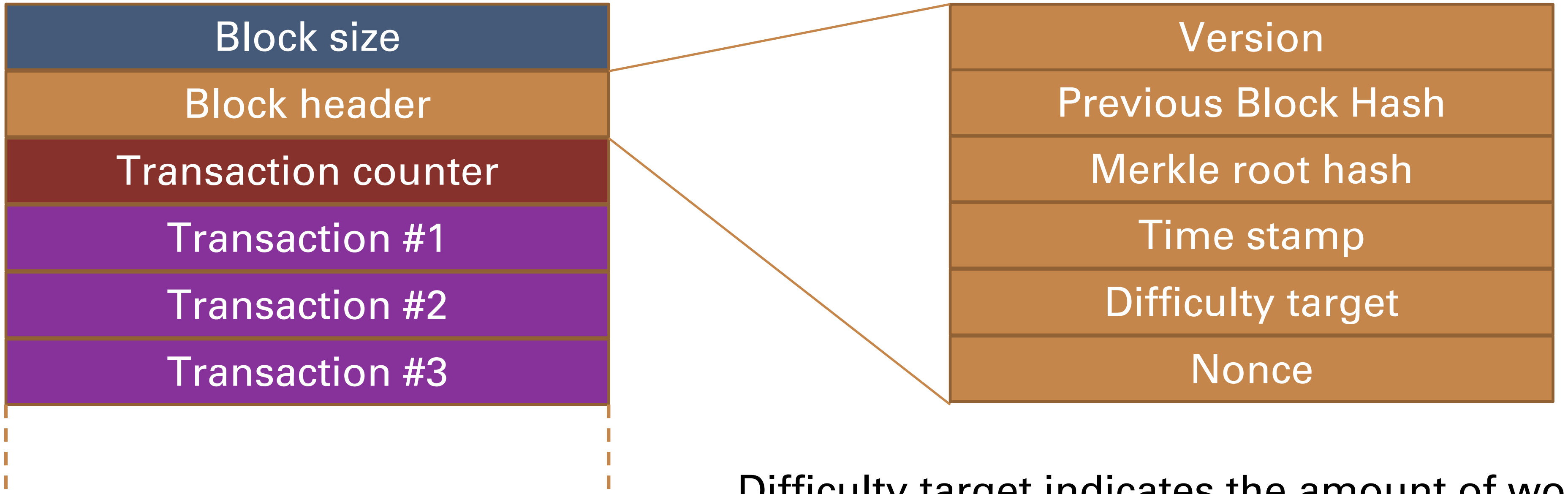
Essentially, blockchain makes it possible to do transactions without an intermediary.

The Blockchain



- Each block contains a link to the previous block
- Each block has a hash value of the previous block
- Each block is encrypted (including the hash value of the previous block)
- Changes to one block will make the whole blockchain invalid

Blockchain Blocks and Block Header



Block size

Block header

Transaction counter

Transaction #1

Transaction #2

Transaction #3

Version

Previous Block Hash

Merkle root hash

Time stamp

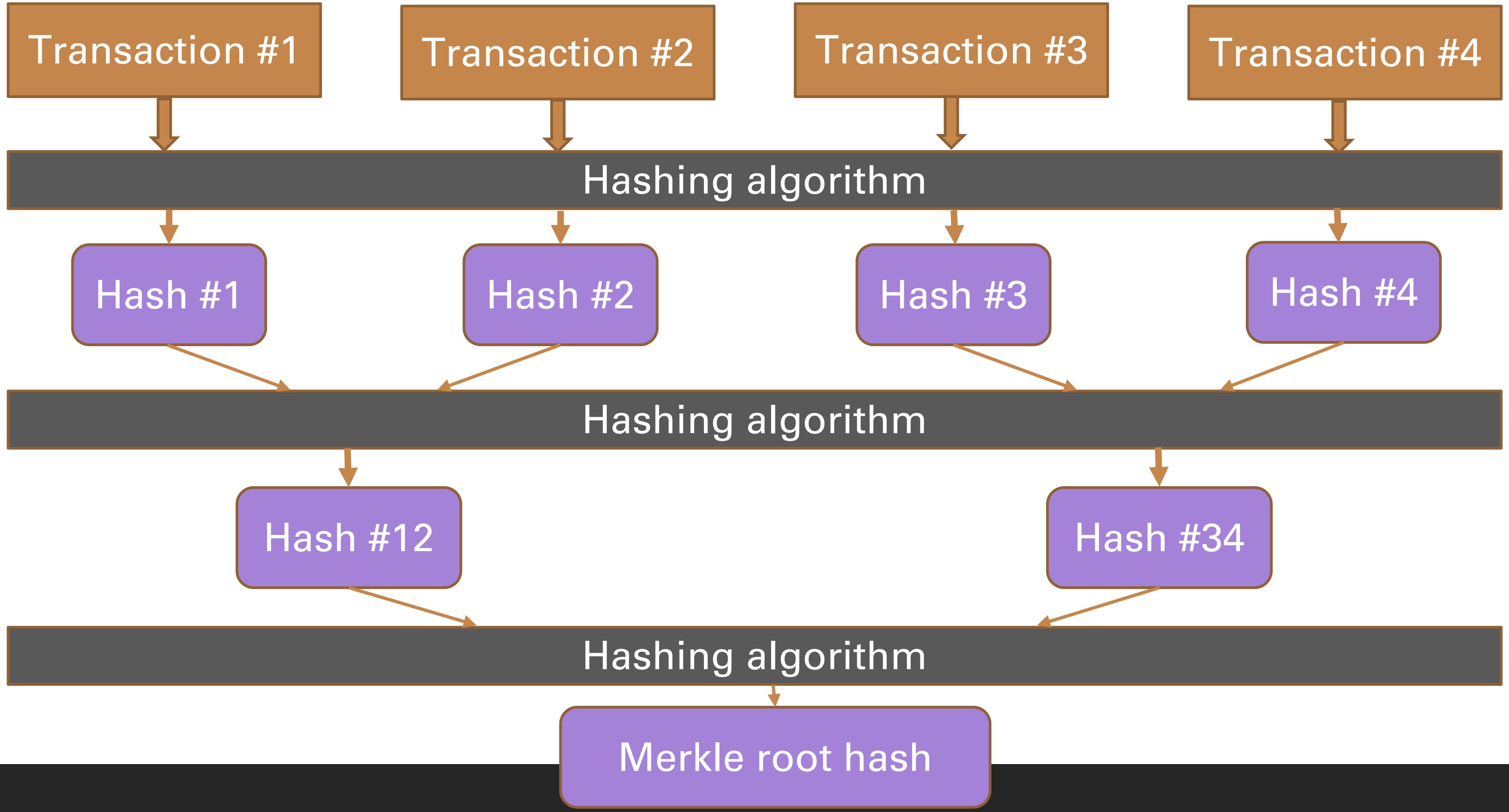
Difficulty target

Nonce

Difficulty target indicates the amount of work for mining

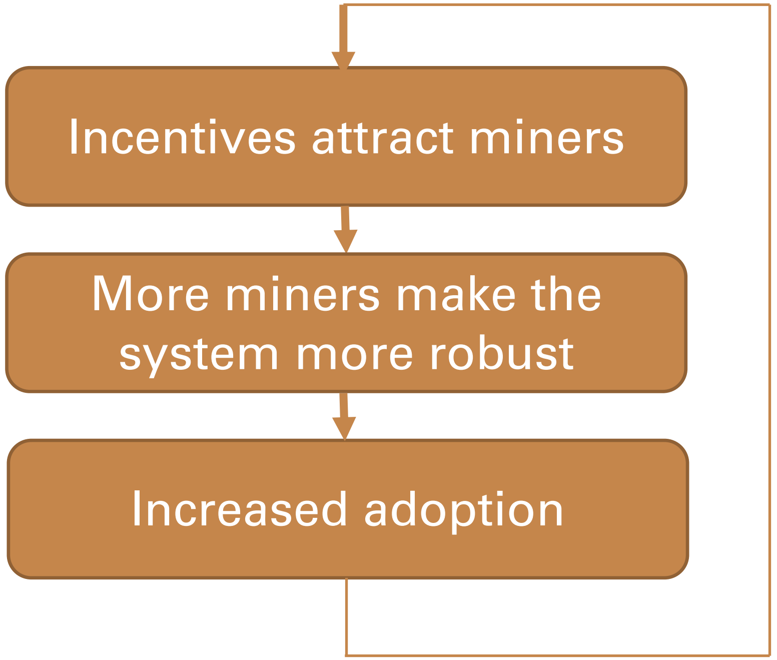
Nonce

Computing the Merkle Root Hash



Validation

- Miners validate the blockchain
- The miners that first succeeds to solve the mathematicl problem is awarded (with cryptocurrency)
- Faster CPUs / GPUs / ASICs reduces the time to solve problems. Bitcoin network responds by increasing the difficulty of the mathematical problem.
- Mining pools is a way to consolidate individual resources to solve more problems and get awarded.



Incentives attract miners

More miners make the system more robust

Increased adoption

Distributed Ledger

Each node processes and stores an identical copy of the latest version of the ledger (blockchain).

Updates are performed idependently and recorded by every node in the network.

The nodes then execute a voting process on the validity of the new update.

When consensus is reached (a majority of nodes votes yes) the block is added to the ledger.

Therefore, a possible attack on a minority of the nodes will be rejected.

The Wallet

Wallet

The wallet is used for sending, receiving, and storing cryptocurrency.

Similar to a bank account.

Different types:

- Online (software) wallets
- Physical (hardware) wallets (e.g. USB-stick or even paper)

Exchanges

An exchange converts between currencies and cryptocurrencies.

Example: CoinBase

Necessary to do transactions on the public (no test) blockchains

Public or Private Blockchain

Public

- Everyone can access (transparent)
- Global infrastructure
- Users pays for transactions
- Price is dynamic (supply-demand)
- Payment may be exchanged into real currency

Private

- Only members can access
- Local infrastructure
- Operator has responsibility for infrastructure (including scaling)
- Users pay for transactions, but payment has no real value

Ethereum and Smart Contracts

Ethereum

Ethereum is an open source blockchain platform with excellent support for writing smart contracts.

The developer support offered by Ethereum makes the Ethereum platform a viable candidate for the Smart-MLA application.

The main reason for using blockchain technology is that multi-layer aggregation involves multiple sellers and buyers without any centralized controlling entity.

Smart Contracts

A smart contract is a computer program that is automatically executed when certain predetermined conditions are met

Code is normally written using the Solidity programming language on top of the Ethereum platform. Solidity has a syntax close to JavaScript.

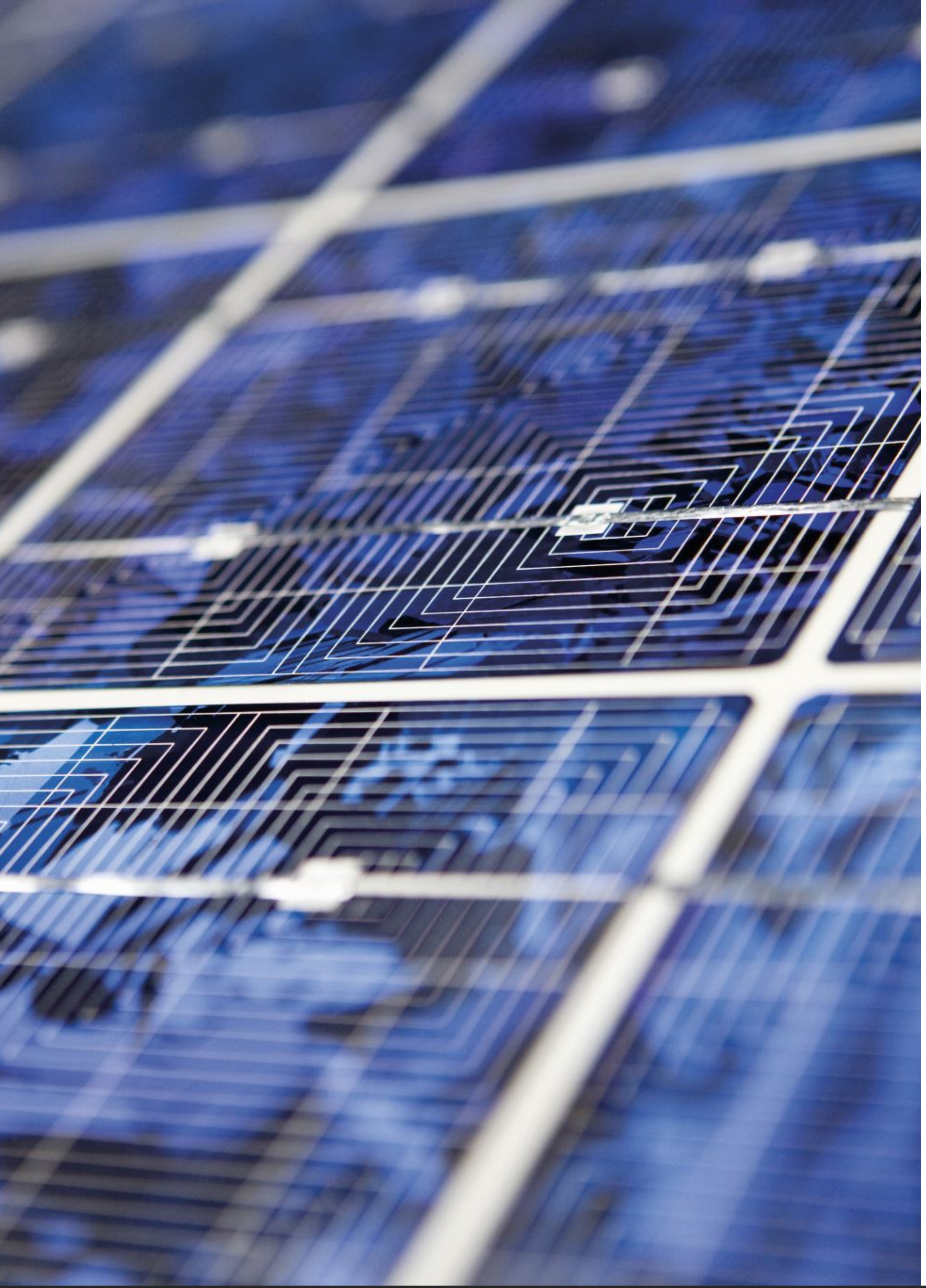
Smart contracts facilitates storing and retrieving transaction information in the Ethereum blockchain.

Smart contracts are autonomous and distributed.

A Short Introduction to Smart Grids

Renewable Energy Adoption

This picture is from Hurdal Eco-Village north of Oslo, Norway. Solar panels on all roofs, but no smart grid. Each house produces energy for own use.



Smart Grids

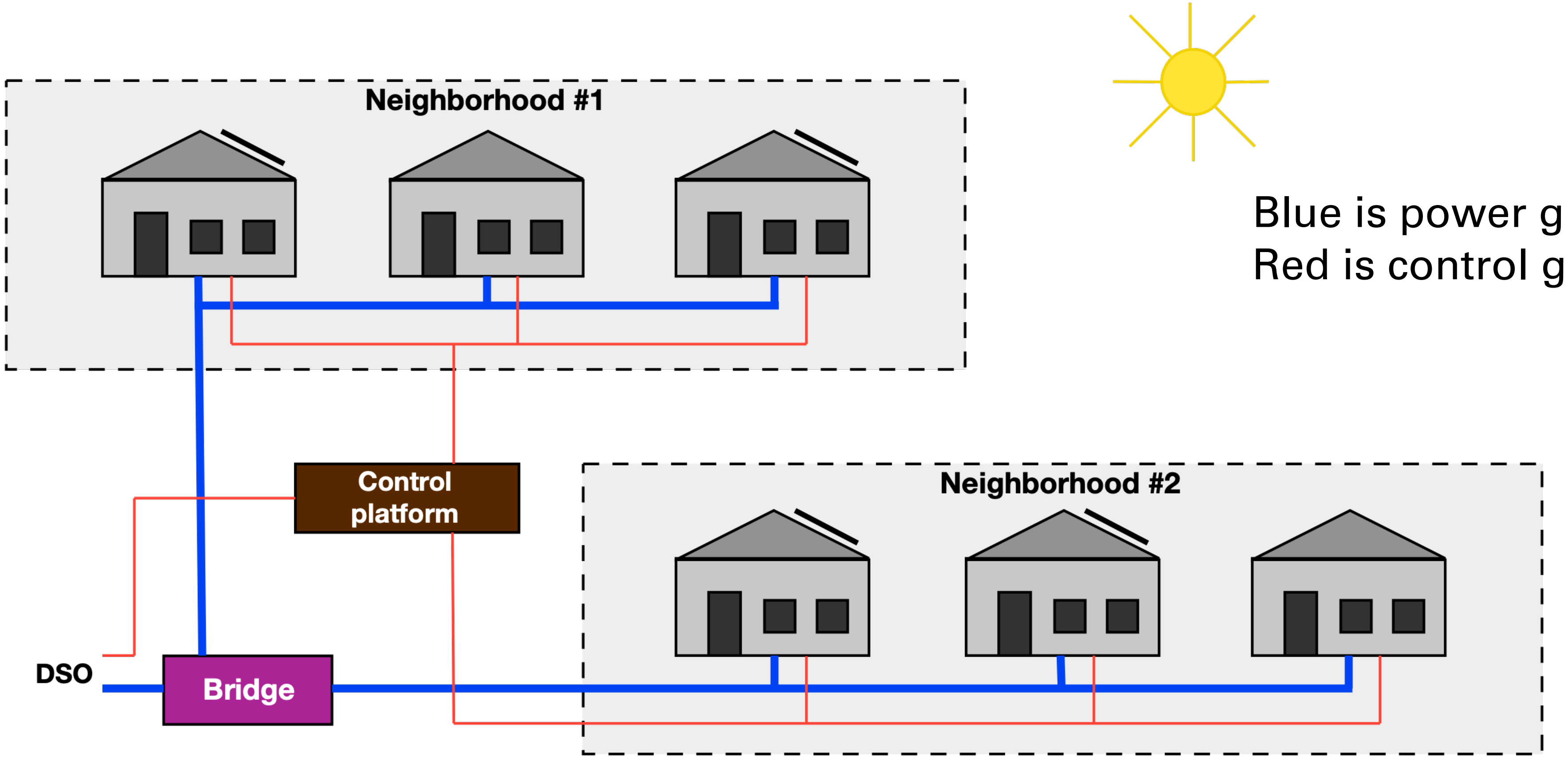
Connects producers and consumers together.

Smart meters keeps track of energy production and consumption.

A smart grid has a control platform to collect information from the smart meters.

Possible to streamline power consumption according to grid demand.

Smart Grids



Blue is power grid
Red is control grid

Smart Grids and Blockchain Technology

Blockchain in the Smart Grid

The smart grid consists of energy producers and consumers – prosumers

Blockchain can be used to store transactions between producers and consumers.

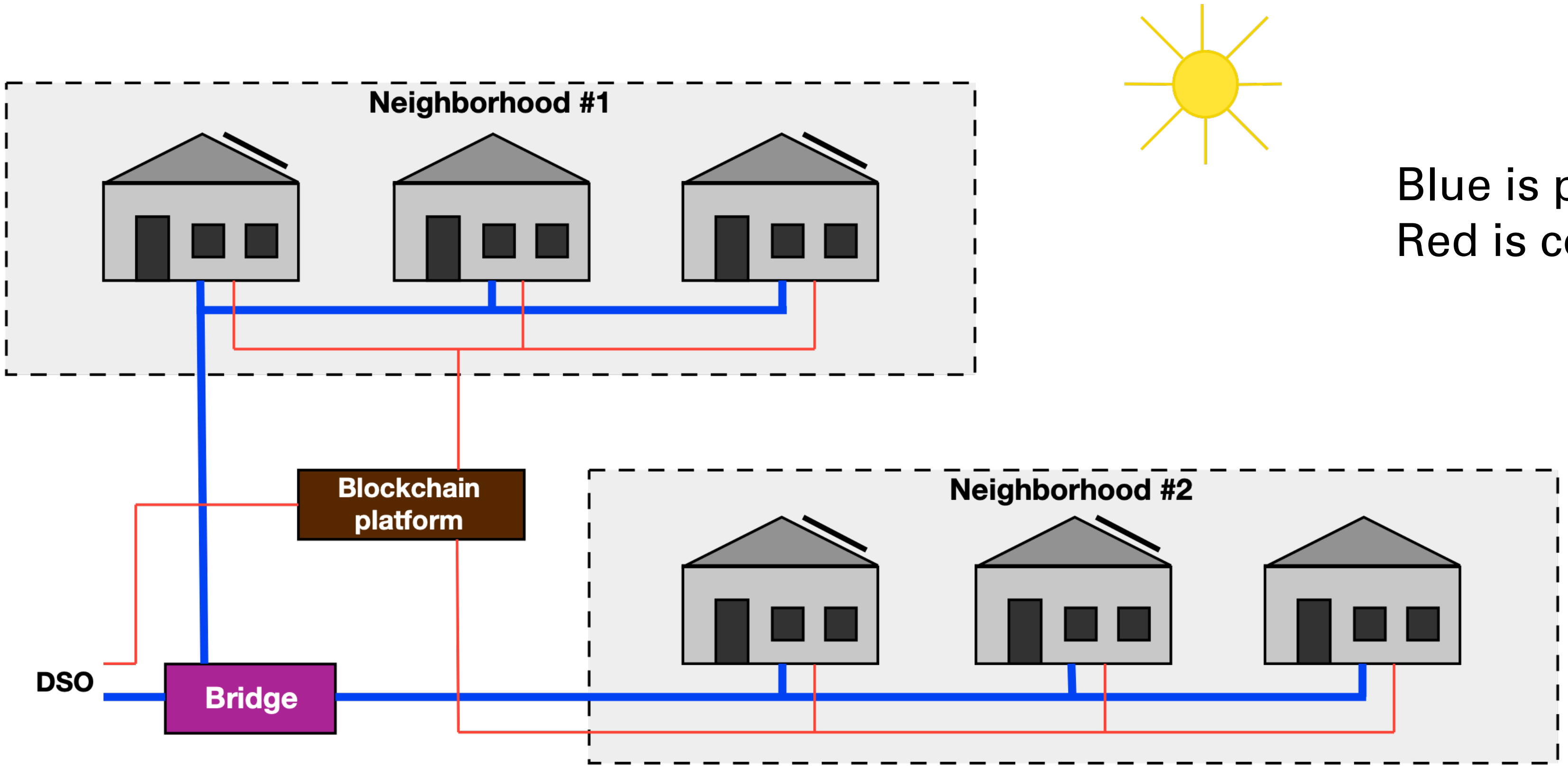
When transaction is added to the blockchain, it cannot be altered.

Can be used for settlements between producers and consumers.

Smart contracts can be used to initiate transactions when certain conditions are met, e.g., to optimize power consumption.

Example: Electric vehicle will be charged when the price is lowest within a certain time period.

Smart Grid with Blockchain Technology



Blue is power grid
Red is control grid

Smart Grid with Blockchain Technology

The first widely known smart grid project using blockchain was the **Brooklyn Microgrid**, launched in a neighbourhood in Brooklyn, New York. The project successfully implemented a P2P electricity trading platform based on the blockchain in a microgrid setting.

Around the world, several blockchain have been introduced by start-ups and utility companies such as **Vattenfall** in the Netherlands, **Innogy** in Germany, **Wien Energie** in Austria and **Power Ledger** in Australia.

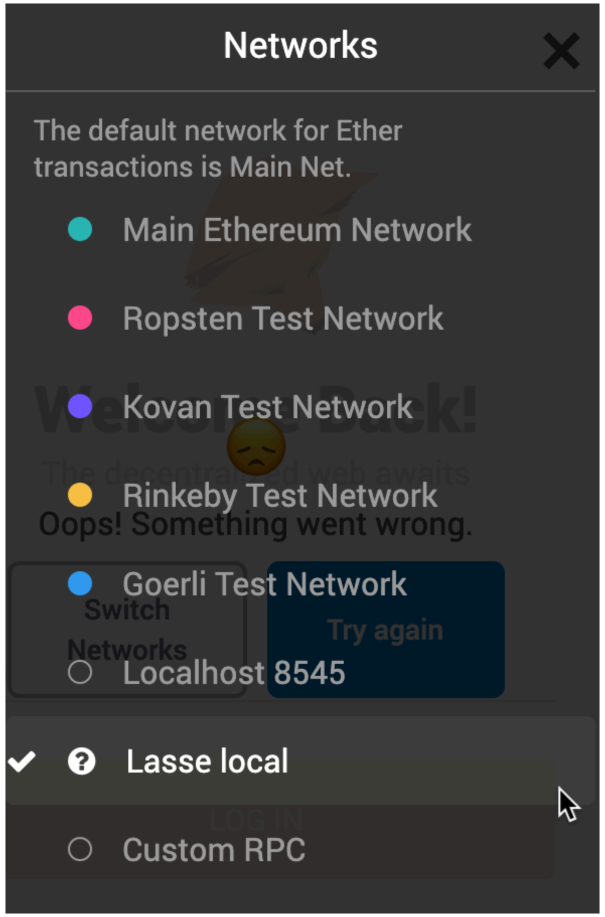
Making a Demonstrator for Settlements using Blockchain

Tools for Blockchain Development

Implementing blockchain technology is one of the aims of the Smart-MLA project. As already explained, many other projects have experimented with blockchain for smart grids, but we are adressing the specific needs of the Smart-MLA project.

This section introduces tools we have used to create a demonstrator for settlements among prosumers in a local smart grid. We also made experiments with smart contracts to optimize energy consumption.

Public Blockchains



The default network for Ether transactions is Main Net.

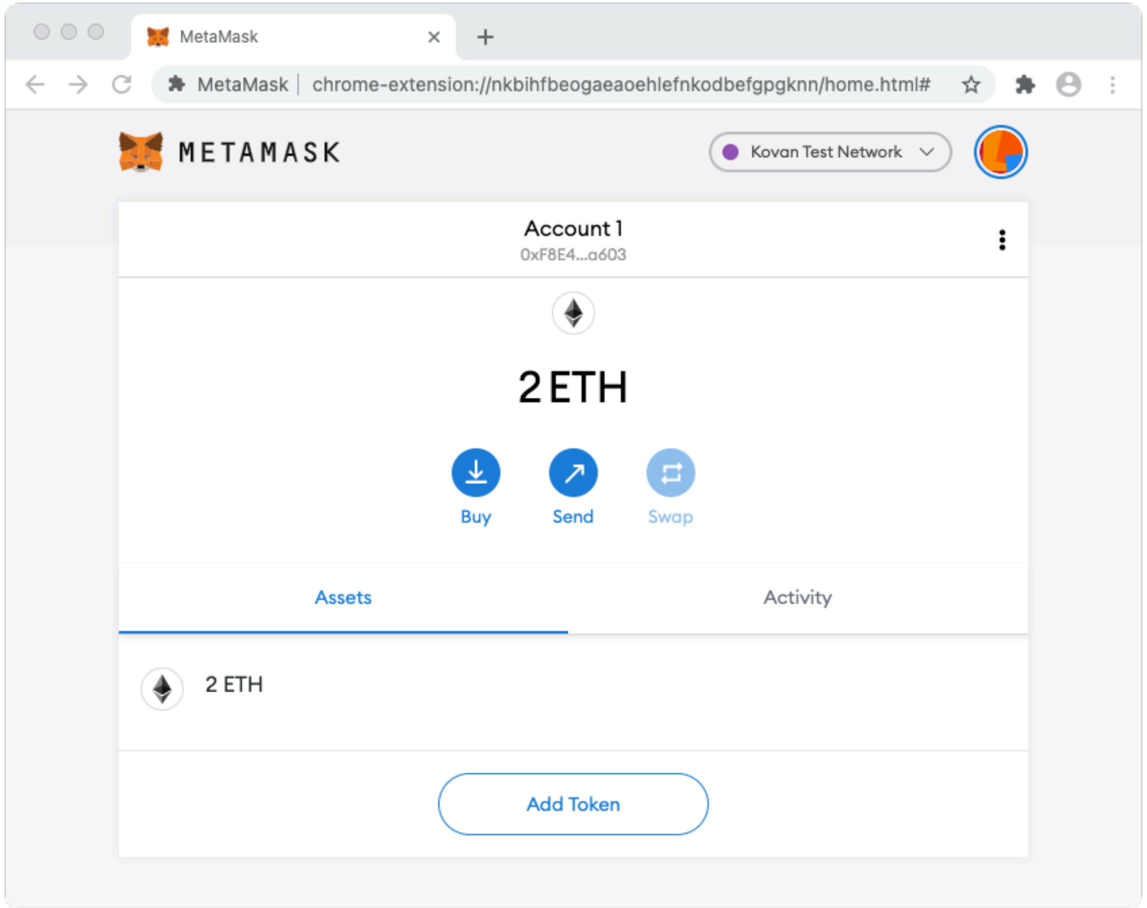
- Main Ethereum Network
- Test networks:
  - Ropsten
  - Kovan
  - Rinkeby
  - Goerli
- Test networks are for demonstration purposes. Transactions are paid with cryptocurrency (Ether). The currency in the test networks has no real value, and can be obtained for free.

The Metamask Ethereum Wallet

Chrome Extention

Here Metamask is connected to the Konan test network. The current balance is 2 ETH. This currency has no real value.

Metmask is also used with local blockchains (Geth or Ganache described in the follwing slides)



Node.js and NPM

Node.js is a JavaScript runtime platform

NPM (*Node Package Manager*) uses Node.js and is a global repository of JavaScript software packages. NPM contains a website, a command-line interface, and a registry of software packages.

The Geth Client

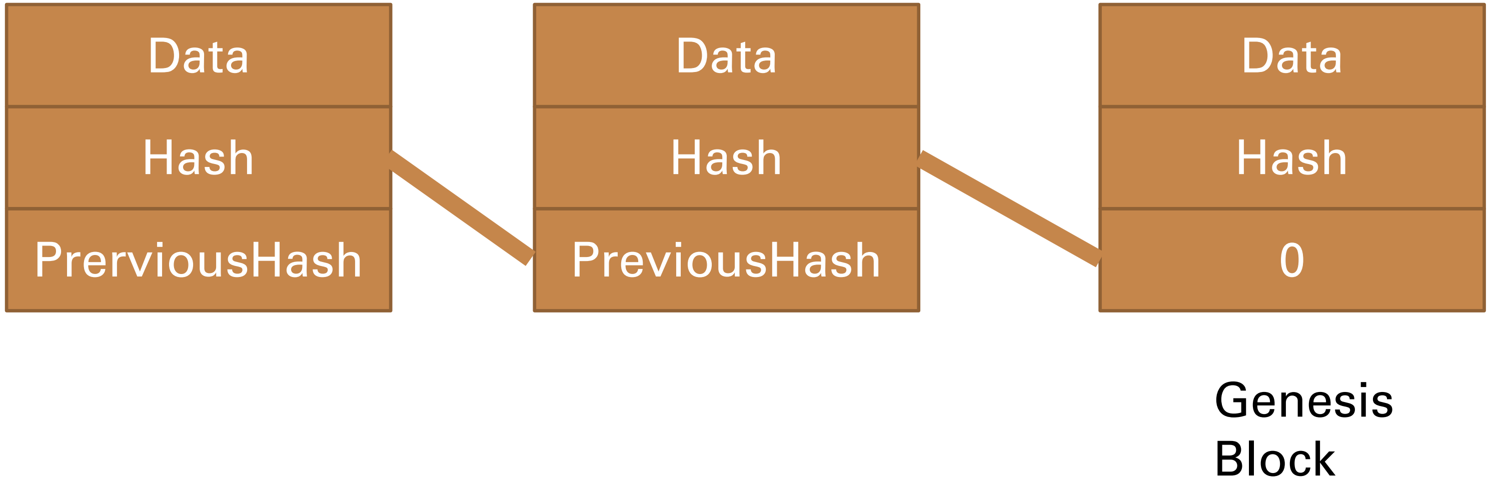
To establish a private Ethereum test network, the **Geth** client was used. **Geth** is the official Ethereum client, written in the Go programming language. The **Geth** client can be downloaded from: http://geth.ethereum.org/downloads/

The current version is 1.9.24, and downloads exist for Linux, macOS and Windows.

We used the macOS version for our test network. After downloading and uncompressing, the **Geth** binary is moved into the home directory.

The Genesis Block

The first thing that needs to be done is to create the genesis block. We need to create a file called **genesisblock.json** with all the fields of a Ethereum block.



Creating Two Nodes of the Network

```
geth –datadir ~/MyTestNet/data/node1 init ~/MyTestNet/genesisblock.json

geth –datadir ~/MyTestNet/data/node2 init ~/MyTestNet/genesisblock.json
```

Ganache

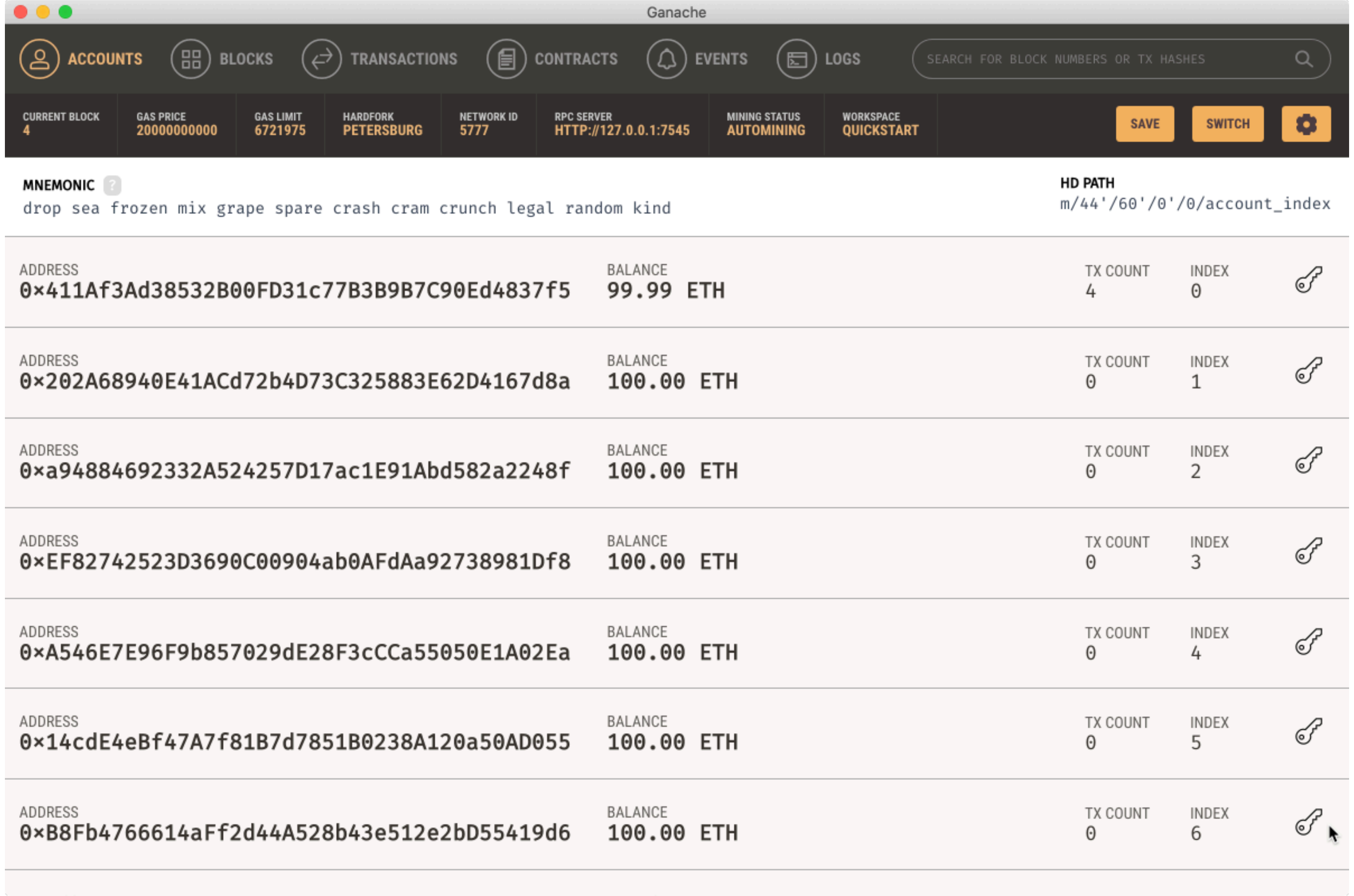
Ganache is a host for a local blockchain. It can be used for testing Blockchain applications.

Ganache can be downloaded from https://www.trufflesuite.com/ganache

The figure on the next slide shows the graphical user interface, but it is also a command line interface called ganache-cli.

Ganache

It sets up 10 accounts, each holding 100.00 ETH. These accounts can be used to test smart contracts in the local blockchain.



CURRENT BLOCK 4 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK PETERSBURG | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE QUICKSTART

MNEMONIC
drop sea frozen mix grape spare crash cram crunch legal random kind

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×411Af3Ad38532B00FD31c77B3B9B7C90Ed4837f5	99.99 ETH	4	0
0×202A68940E41ACd72b4D73C325883E62D4167d8a	100.00 ETH	0	1
0×a94884692332A524257D17ac1E91Abd582a2248f	100.00 ETH	0	2
0×EF82742523D3690C00904ab0AFdAa92738981Df8	100.00 ETH	0	3
0×A546E7E96F9b857029dE28F3cCCa55050E1A02Ea	100.00 ETH	0	4
0×14cdE4eBf47A7f81B7d7851B0238A120a50AD055	100.00 ETH	0	5
0×B8Fb4766614aFf2d44A528b43e512e2bD55419d6	100.00 ETH	0	6

The Solidity Programing Language

Solidity is an object-oriented language for implementing smart contracts. Smart contracts are programs which govern the behavior of accounts within the Ethereum state.

Solidity is executed on the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

The Solidity web site provides the following applications for smart contracts: Voting, crowdfunding, blind auctions, and multi-signature wallets.

The Solidity Programing Language

Solidity can be used for settlements between producers and consumers in a smart grid. The basic functionality will be to get the current market price, subtract from one account and add to another account.

The settlements can be done at predefined intervals, e.g., each hour. The market price may be a local market price within the grid.

Additional functionality may include web access to show transaction data, and also transfer money to and from the cybercurrency account.

https://solidity.readthedocs.io

Version used: 0.5.17

Remix

A browser-based IDE for developing and deploying smart contracts.

Runs on remote host.

We used this for smart contract development and testing.

Truffle Framework

The Truffle Framework is used to create and deploy smart contracts.

Runs on local computer.

Some application examples are available as “boxes” that can be ”unboxed”.

In practice, these examples have not been helpful, mostly due to dependency conflicts.

Starting with empty projects from scratch have worked well.

Web3

Ethereum JavaScript API. Other libraries exist for other programming languages, i.e., Python.

Web3 can be used from the Truffle development console.

```
truffle(development)> web3.eth.getAccounts()
```

Web3

```
truffle(ganache)> web3.eth.getAccounts();

[

  '0x411Af3Ad38532B00FD31c77B3B9B7C90Ed4837f5',

  '0x202A68940E41ACd72b4D73C325883E62D4167d8a',

  '0xa94884692332A524257D17ac1E91Abd582a2248f',

  '0xEF82742523D3690C00904ab0AFdAa92738981Df8',

  '0xA546E7E96F9b857029dE28F3cCCa55050E1A02Ea',

  '0x14cdE4eBf47A7f81B7d7851B0238A120a50AD055',

  '0xB8Fb4766614aFf2d44A528b43e512e2bD55419d6',

  '0x9317bAaF6d0BB248cf2d4e8CF60c72dfE592c9AC',

  '0xDa3c7d4bf750611702F7DD04fF9c0630Dda81728',

  '0xF32B9D81E03642801949486E5671003D59540Fda'

]
```

This is the output from running
the command on the previous slide

Compare the accounts to the accounts
listed by Ganache in an earlier slide.

Application Programming Interface

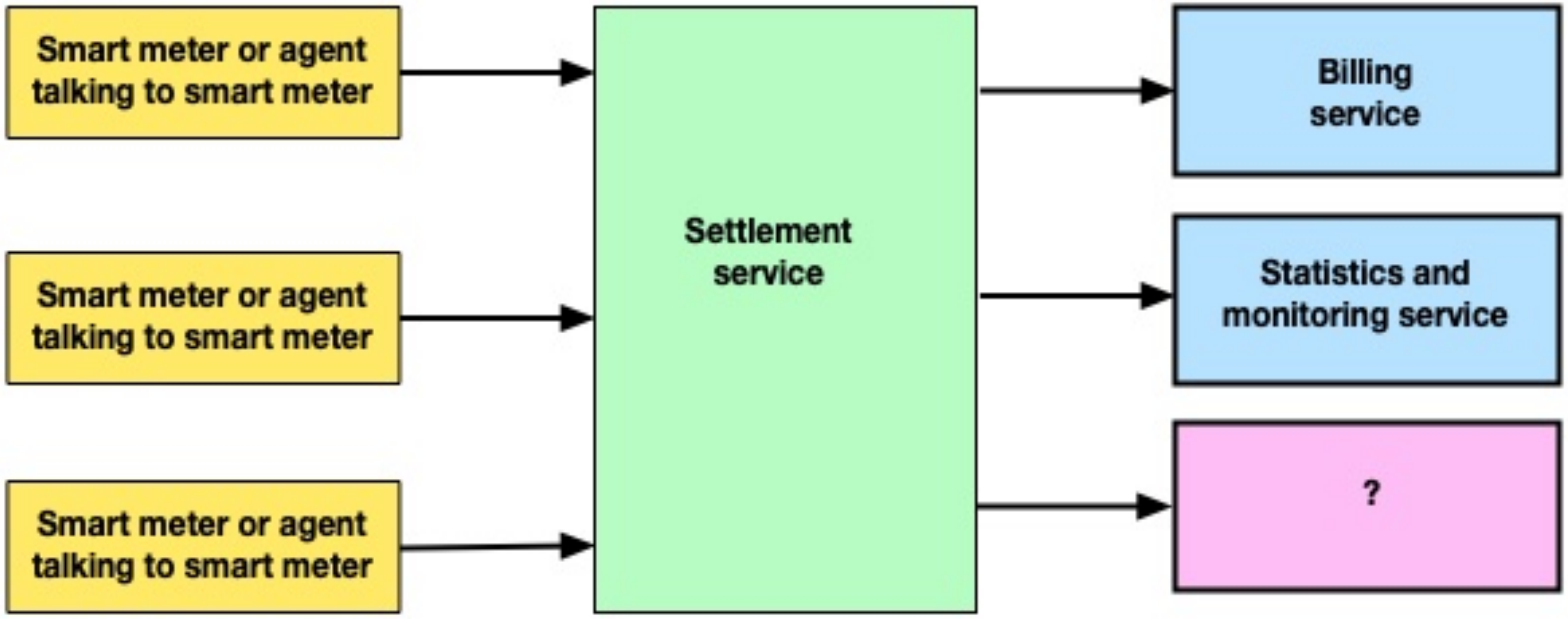
A web-based service running on TCP port 80 (or port 443).

The data can be sent by using the HTTP protocol using POST or GET messages.

The obvious advantage of using a web-based interface is to avoid the problems caused by firewalls, that often blocks access to other TCP ports.

The disadvantage is increased vulnerability. This is handled by careful crafting of the messages along with use of secure communication (HTTPS running on port 443).

Application Programming Interface


Smart meter or agent talking to smart meter
Smart meter or agent talking to smart meter
Smart meter or agent talking to smart meter
Settlement service
Billing service
Statistics and monitoring service
?

Application Programming Interface (API)

To communicate with the blockchain, we made an API for settlements: This is used by the settlement service to register transactions and retrieve information from the blockchain.

```
Register_transaction(date_time, energy, tariff, buyer_ID, seller_ID)
```

Note: The seller may be an aggregator.

```
Get_all_transactions(fromdate_time, todate_time)
```

```
Get_all_transactions(fromdate_time, todate_time, seller_ID)
```

```
Get_all_transactions(fromdate_time, todate_time, buyer_ID)
```

Implications for the Smart-MLA Project

Implications for the Smart-MLA Project

Lessons learned

Economy

Blockchain risks

Lessons Learned

Blockchain is still an emerging technology.

We experienced many setbacks due to new versions of tools, frameworks and the Solidity language. What you did yesterday was suddenly not working with a new version.

For Solidity, version number is set in the pragma directive.

We learned not do do uncessesary upgrades just because a new version appeared.

Economy

Settlements in a smart grid is about very small amounts.

Ethereum does not have a maximum block size, but the block size is regulated by how many units of gas can be spent on each block. This is supply-demand driven.

Miners are currently accepting blocks with an average gas limit of approximately 10.000.000 gas.

The average block size is now between around 40 kb.

https://etherscan.io/chart/blocksize

Economy

On the public Ethereum network the average transaction cost for November 19th was around USD 2,70. This would be OK for registering such things as car ownership, since the transaction cost would be neglible compared to the object being part of the transaction.

For smart grid settlements the situation is very different. The transaction cost are many times higher that the value being transferred, even if many transactions are squeezed into the same block.

Therefore, using the Ethereum public blockchain for smart grid settlements seems to be a bad idea. Still, because of the other characteristics of the Ethereum blockchain, a consortium or private blockchain seems to be a viable alternative.

The costs of the infrastructure could then be covered by a (very) small transaction fee, or by a subscription fee.

Blockchain Risks

Being aware of the risks associated with blockchains is essential.

Today, the public Ethereum blockchain is distributed and free (except for transactions modifying the state of the blockchain).

However, there have been some warnings about the scalability and energy consumption to uphold the blockchain.

Some future technology may make blockchain obsolete, and at some point, there may be an unwillingness to pay for the computer resources to run the blockchain. (It is based on supply-demand).

Blockchain Risks

In our case the risks are considered low for two reasons:

- We consider a private blockchain as the best alternative.
- The transaction records are only needed for a limited period. The records are needed for fulfilment of payments, and maybe to generate reports for tax purposes.
- Therefore, it should not be a problem to substitute the private blockchain with alternative technologies.

Discussion and Future Work

Multi-layer aggregation may create new smart grid business models that have the capacity to disrupt large energy distributors market monopoly of selling energy to households and propose new value proposition that is compelling to customers, achieves advantageous cost and reduce risk structure.

This keynote has introduced blockchain and smart grids with focus on demonstrating the use in the Smart-MLA project.

Discussion and Future Work

The Smart-MLA project will run for another year.

An experiment with smart contracts for energy optimization will be done next spring.

The aim is to demonstrate the whole multi-layer aggregation concept at the end of the project, including the use of blockchain to handle settlements and smart contracts.

Thank you for your interest

Please send me an e-mail if you are working on similar things.

E-mail: lasse.berntzen@usn.no