FH Salzburg

Peer-to-peer Energy Trading on a Blockchain

Lessons Learned from an Austrian Proof of Concept

Andreas Unterweger,

Center for Secure Energy Informatics

SNIPPET School 2022, Crete

Technology
Health
Media

Context

- Photovolatic (PV) power plants
  - can significantly decrease power drawn from the electricity grid
  - are becoming more and more wide-spread
  - are also installed onto multi-party homes

- How to best share energy from PV power plants?
  - Use as much of the generated power as possible
  - Avoid feeding power into the electricity grid (low reimbursements as of 2017)
  - Avoid having „unused“ energy
    → transfer to another party instead of feeding it into the electricity grid

Legal framework in Austria (as of 2017)

- Energy “transfers“ between parties of shared PV power plants
  - Parties agree on a (default) distribution key for generated energy
  - Parties can change the distribution
  - 15-minute granularity
  - Energy provider needs to consider transferred poritions for each customer

- Example
  - Customers A, B and C share a PV power plant
  - Default shares: A: 20%, B: 20%, C: 60%
  - 21.10. 11:00-12:00: A transfers its 20% to B
    → Distribution A: 0%, B: 40%, C: 60% for one hour
  - After 12:00: Default shares restored

Illustrated example

Date/Time	PV name	Customer A	Customer B	Customer C
21.10. 10:45	PV 1	20%	20%	60%
21.10. 11:00	PV 1	0%	40%	60%
21.10. 11:15	PV 1	0%	40%	60%
21.10. 11:30	PV 1	0%	40%	60%
21.10. 11:45	PV 1	0%	40%	60%
21.10. 12:00	PV 1	20%	20%	60%

Motivation

- Legal requirement for energy providers to consider tranferred PV portions of each customer with 15-minute accuracy

- Industry partners (energy providers) would like to
  - have a prototype for P2P energy trading
  - see whether blockchain technology is adequate for P2P energy trading
  - learn about blockchain technology

Design decisions I

- Private permissioned blockchain
  - All participants are known
  - Outside access (also read-only) is undesired
  - There is no central trusted party (customers can choose their energy provider)
  - Not all parties mutually trust each other
  - Different permissions for participants, e.g., read only for energy provider

- Legal caveats
  - Trading energy formally requires a license (as of 2017)
  - PV „trading“ only allows for exchange of portions, not kWh or €

Portion examples for consensus

Date/Time	PV name	User 1	User 2	Consensus?
…				
1. Okt. 15:00	PV 1	50%	50%	OK
1. Okt. 15:15	PV 1	70%	30%	OK
1. Okt. 15:30	PV 1	70%	40%	Not OK
1. Okt. 15:45	PV 2	80%	20%	OK
…				

Design decisions II

- Conditions for consensus
  - The sum of portions must be exactly equal to 100%
  - Individual portions must not be negative
    → No indivual can transfer more portions than they currently have

- Transfers do not require consent by the recipient
  - Tranfers are performed by the sender only
  - Deliberate design decision together with industry partners
  - Recipient can pass on the portions if they do not need them
  - No real negative effect if B does not use the portion vs. A not using it

Design decisions III

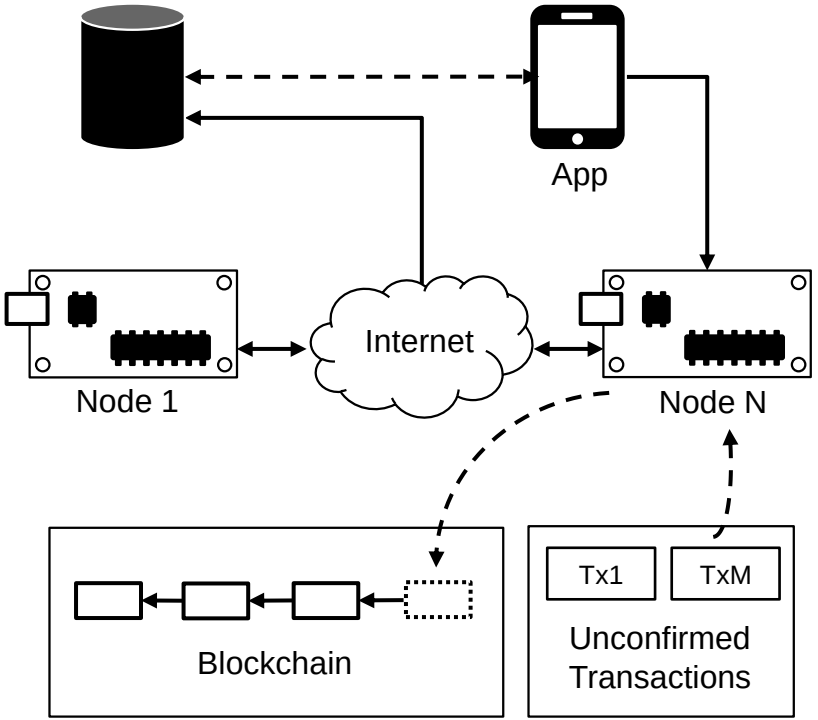
- Blockchain consensus mechanism: Proof of work
  - Proof of stake would not have made sense in this use case
  - BFT would have made sense, but has not had mature implementations (2017)
  - How about energy consumption?
    - Use low-power hardware (details later)
    - Limit mining rates (requires sealed hardware or trusted platform)

- Energy provider hardware is typically sealed and tamper-proof

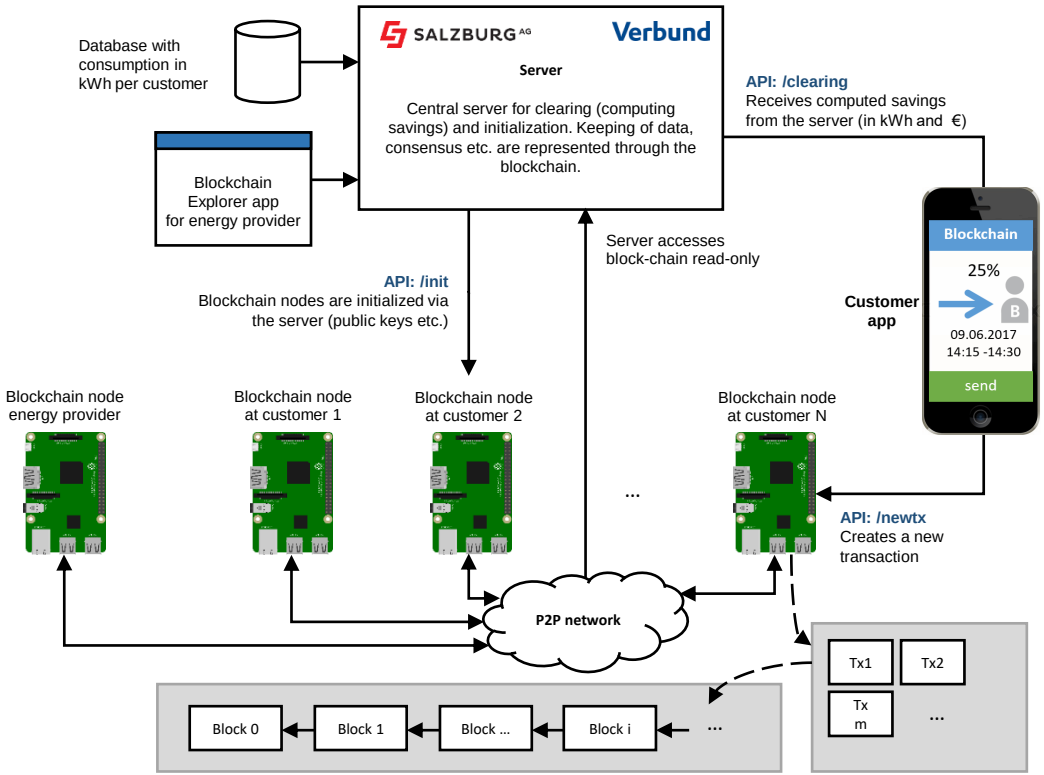
Which blockchain to use (short version, 2017)?

Name	Consensus	Permissioned	Limitation
Bitcoin	PoW	–	Extent of modifications infeasible
Ethereum	PoW	–	Complexity exceeds requirements
MultiChain	PoW	√	Limited to high power consumption platforms
OpenChain	PoA	√	PoA algorithm not suitable for use case
Hyperledger Sawtooth	Dynamic	√	Not mature at time of evaluation
Hyperledger Fabric	Dynamic	√	Known security flaws

Simplified architecture



Architecture

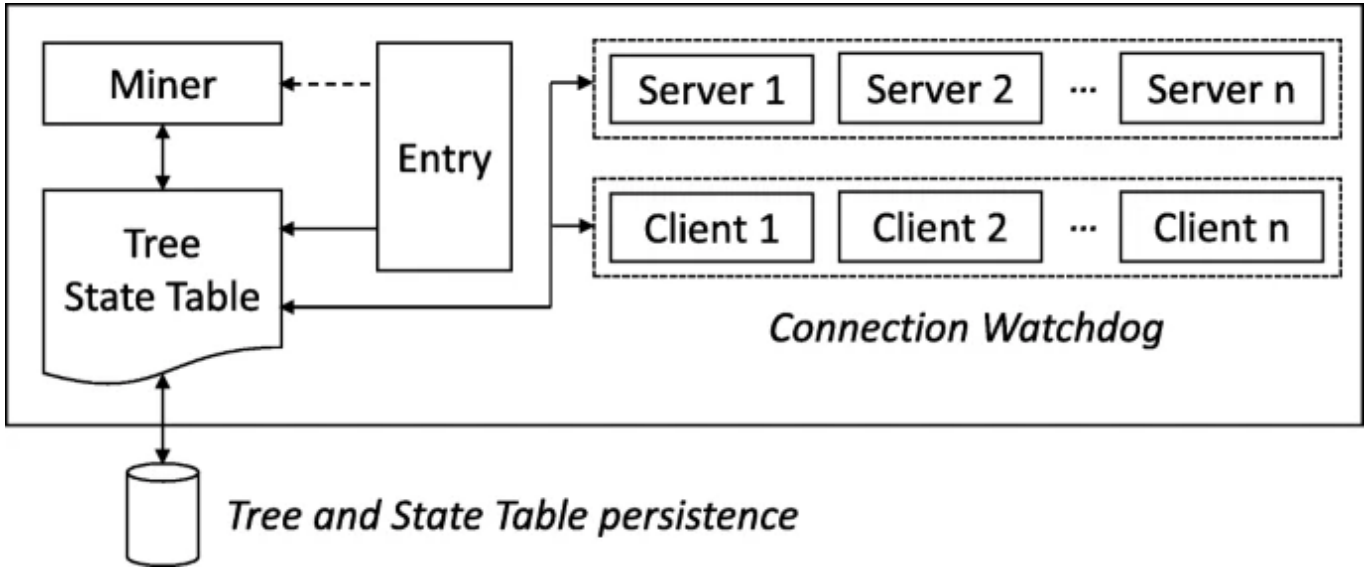


Implementation details I

- Blockchain nodes
  - Hardware: Raspberry Pi 2 Model B
    - is small
    - runs Linux
    - has a low energy consumption
  - Software: Custom implementation in Java 8
    - comes with networking and cryptographic functionality included
    - comes with concurrent algorithms and datastructures included
    - Open source: https://github.com/CenterForSecureEnergyInformatics/ResselChain

- Customer app
  - Android-based
  - Pre-installed on tablets given to the customers

Implementation details II

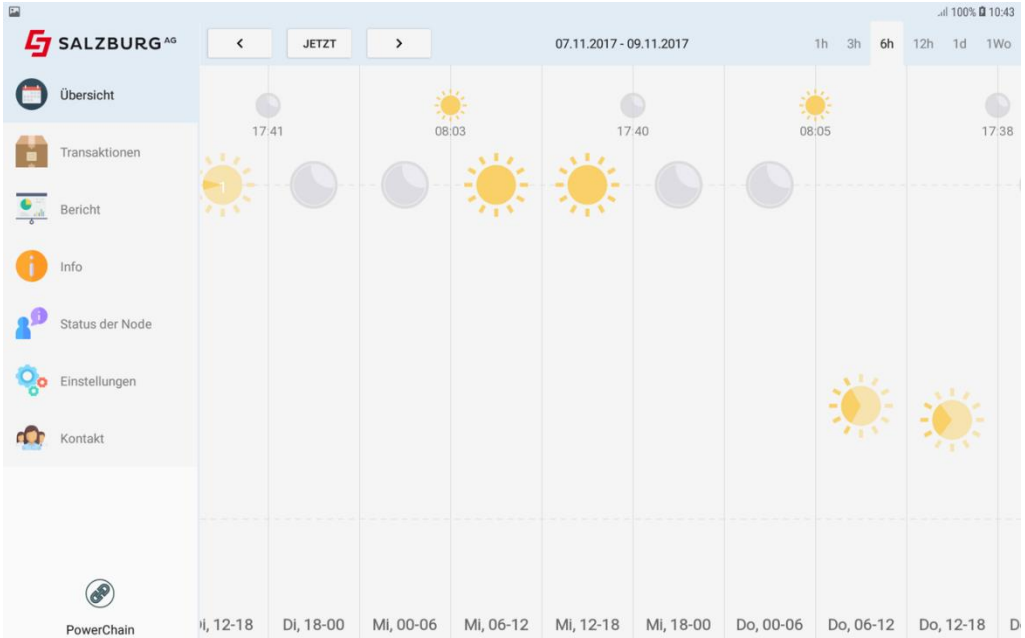


Implementation details III

- Network setup
  - VPN between nodes over the public Internet
  - Proof of concept with power users: Internet access is made available, but the public IP may change and there may be NAT
  - Each customer app is connected to the corresponding node
  - Protocols: XML over TCP/IP (nodes), HTTPS (clearing server)

- Security aspects
  - Public-private key pair for each node
  - All transactions are signed
  - TLS connections with ECDHE and AES-256 CBC

User interface for customers



Proof of concept in Austria

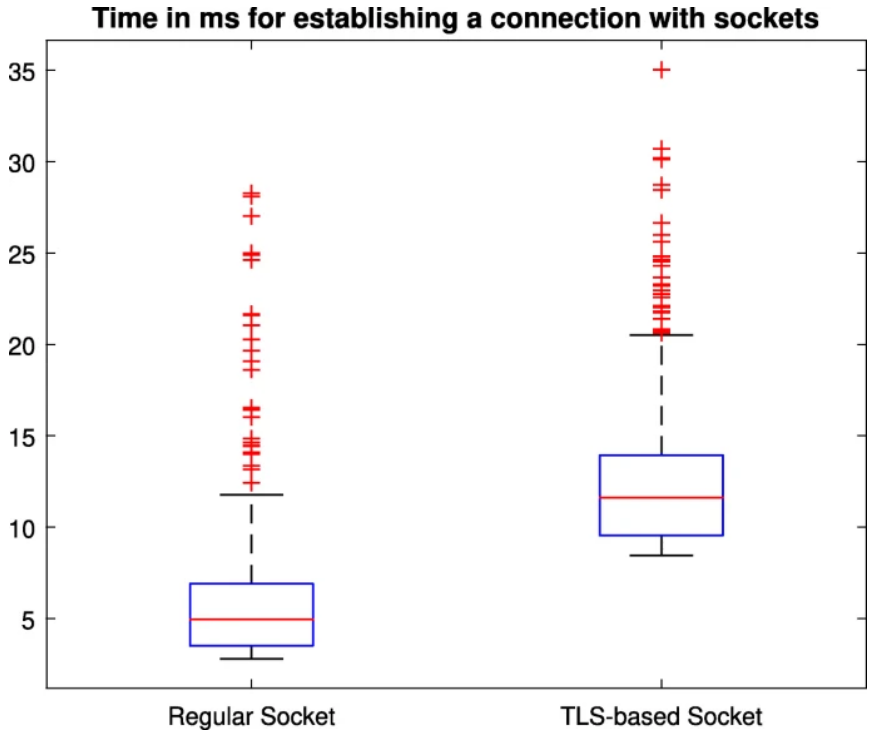
- 2 locations in Austria
- 2 energy providers
- Small number of participants (only at one location)
- Trial through around second half of 2018
- Nodes at
  - each customer/participant
  - both energy providers
  - our University (mining and reading only)

Lessons learned I

- BFT would have been more suitable as a consensus algorithm
  - Smaller energy consumption than PoW
  - Better scalability up to typical number of households in big apartment houses
  - Only (practically) available after the start of our trial

- TLS handshakes introduce significant delays
  - Encrypted vs. unencrypted connections behave very differently
  - Even small payloads induce relatively high delays
    → Connection establishment (key negotiation) takes more time than expected
    → Allow for high(er) timeouts

Lessons learned II



Lessons learned III

- AES-256 is disabled in Java 8 by default (unexpected)
  - Export restrictions limit AES key size to 128 bits

  → Explicitly enable strong cryptography
  - Two solutions
    - Official: Install policy files (requires root privileges; if missing, encryption fails)
    - Unofficial: Remove restrictions via reflection (relies on internals and may break)

- One communication thread is not enough
  - One single thread with a message queue works well in stable networks
  - In unstable networks, waiting for timeouts fills the message queue

    → No more processing or communication with other nodes until after timeout

    → Use two communication threads (send/receive) per neighboring node

Lessons learned IV

- Resynchronization is hard
  - Nodes get out of sync due to temporary network and/or node failures
    → Newly received blocks do not “fit” because intermediate blocks are missing
    → Backup algorithm for fetching blocks for resync from peer(s)
      - Needs to be synchronous to rebuild the chain(s)
        → Incoming blocks do not affect the building process
      - Having separate sender/receiver threads allows passing on transactions

  - Idle chains are a security liability
    - Bitcoin and other public PoW blockchains always have some transactions to process
    - If participants do not trade, e.g., at night, there are no transactions
      → An attacker could build a longer chain due to the inactivity
      → Allow “empty” blocks without any transactions if there are none to process

Lessons learned V

- User engagement is low even with power users
  - Engagement has been relatively high in the beginning
  - After some time, practically no more transactions – why?
    - kWh and monetary amounts are typically relatively small
    - Not kWh or € can be transferred, but only portions (how much are 10%, 20%, …?)
    - Incentive to send PV portions to others may be low(er) in general

- Laws change
  - Shortly after our trial, the law was changed
  - Energy providers may now charge fees for every change in proportions

  → Even less incentive for customers to change proportions due to cost trade-off

Publication

- F. Knirsch, A. Unterweger, and D. Engel, “Implementing a Blockchain from Scratch: Why, How, and What We Learned,” *EURASIP Journal on Information Security*, vol. 2019, iss. 2, p. 1–14, 2019. Available at https://jis-eurasipjournals.springeropen.com/articles/10.1186/s13635-019-0085-3.

FH Salzburg

Thank you for your attention!

Questions?

Acknowledgements: Salzburg AG (Georg Baumgartner), Salzburg Netz GmbH (Christoph Groiß) and Verbund AG (Werner Eder, Hans-Linus Pfau, Peter Poier, Michael Schramel and Clemens Theuermann-Bernhardt), FH Salzburg (Andreas Unterweger, Clemens Brunner, Mathias Lackner)

Technology
Health
Media