Operating Permissioned Blockchain in Clouds: A Performance Study of Hyperledger Sawtooth

Zeshun Shi, Huan Zhou, Yang Hu, Jayachander Surbiryala, Cees de Laat, Zhiming Zhao. “Operating Permissioned Blockchain in Clouds: A Performance Study of Hyperledger Sawtooth” In *18th IEEE International Symposium on Parallel and Distributed Computing,* ***ISPDC 2019***

UNIVERSITY OF AMSTERDAM

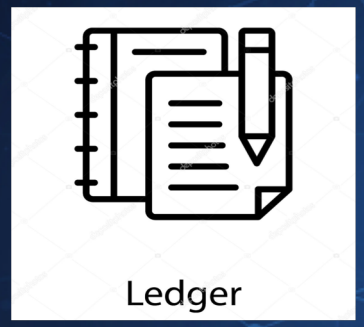
System and Network Engineering

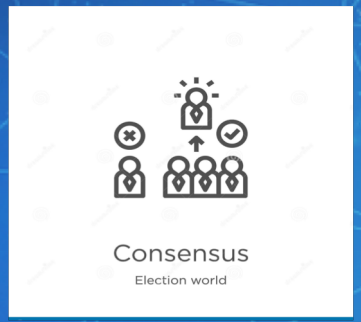
Outline

ARTICONF

- Blockchain: background and challenges
- Hyperledger & Sawtooth and Key Features
  - System architecture
  - Key features
- Experimental Design & Result
- Conclusions

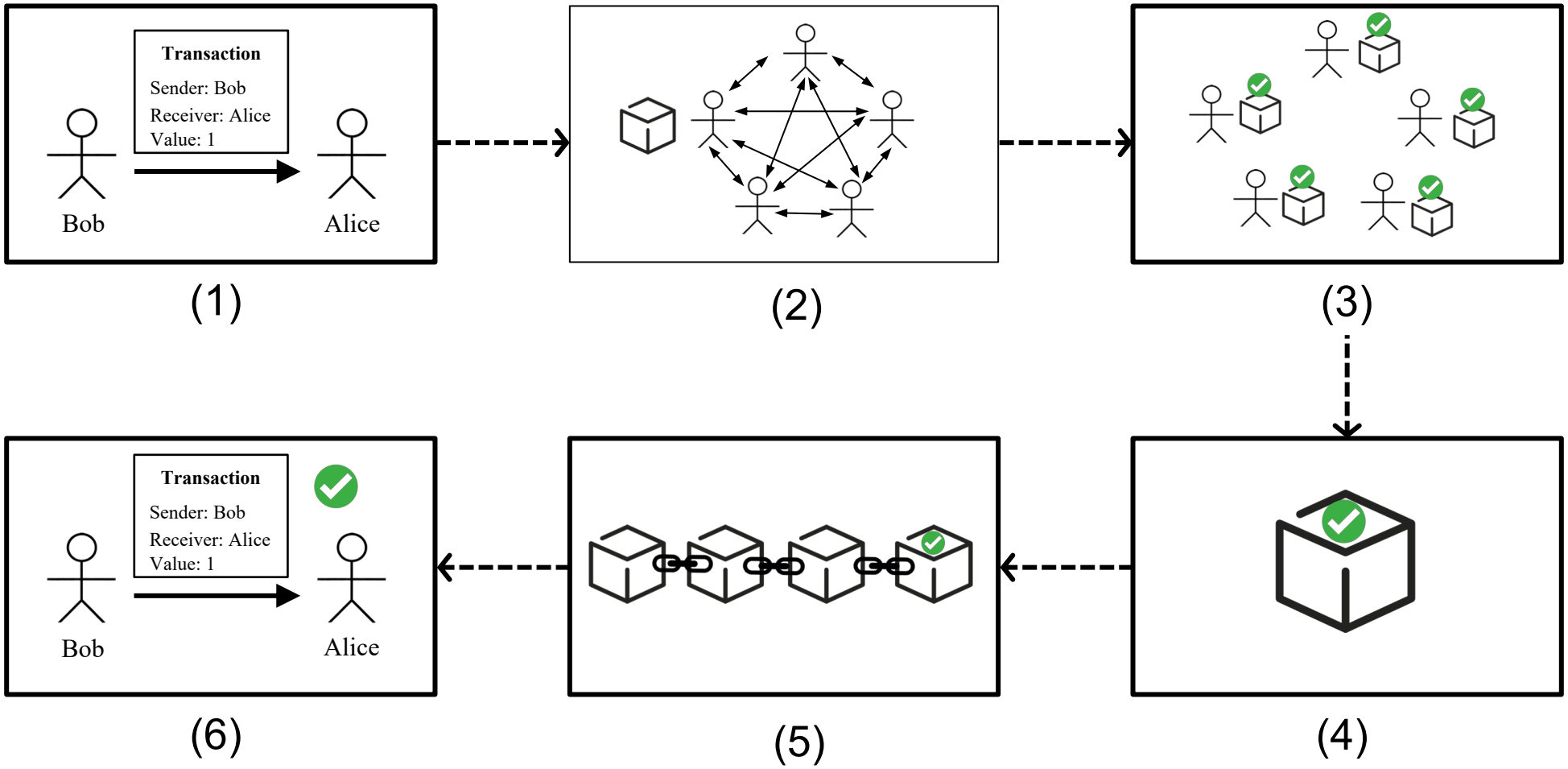
What is Blockchain?





Blockchain is technique, which makes every participant having consensus on a decentralized ledger.

How blockchain works?



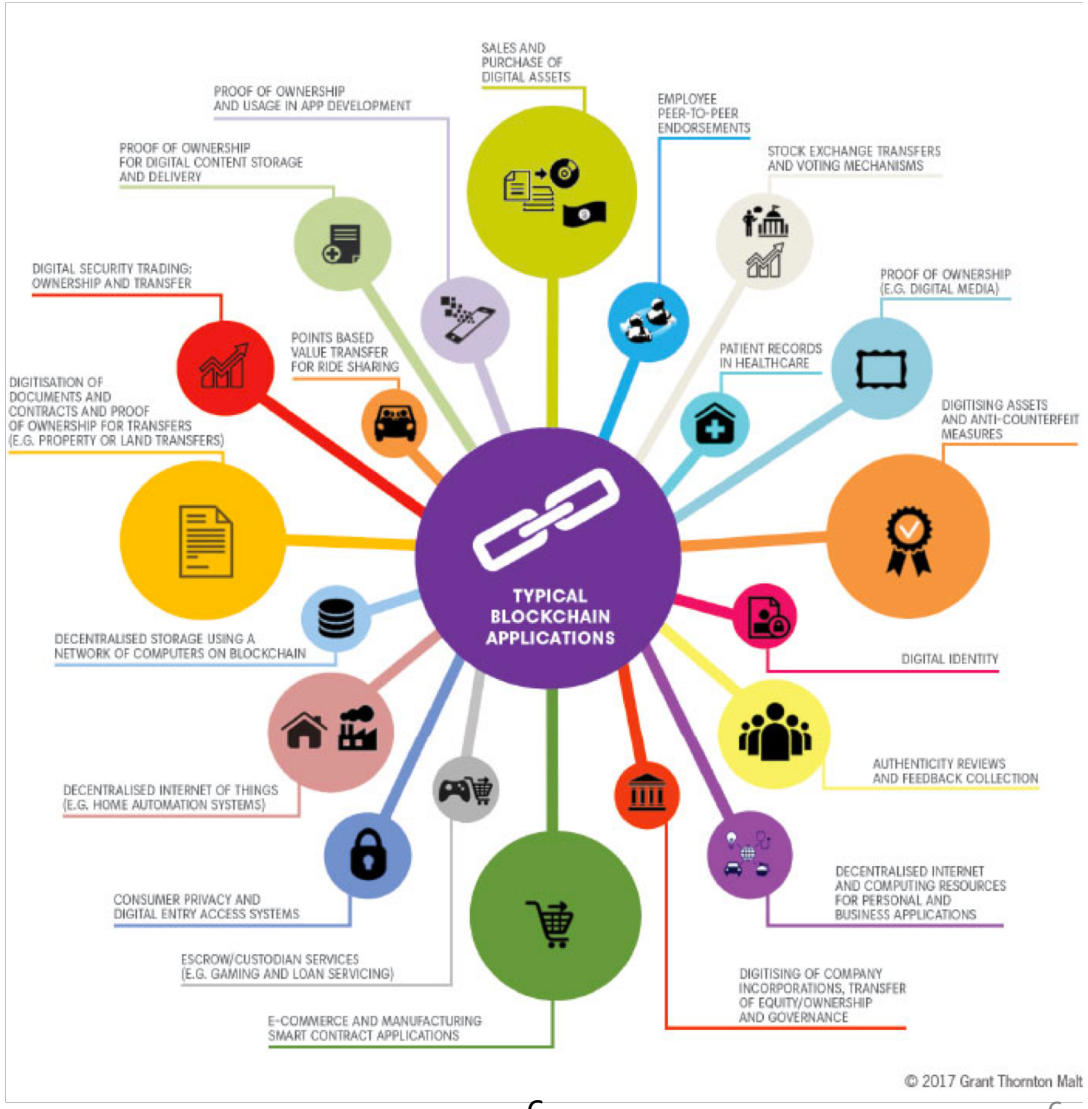
Public vs. private blockchain

- Permissionless Blockchain
  - anyone can join the network and validate the blocks
  - energy-consuming (mining)
- Permissioned Blockchain
  - more suitable for enterprise usage
  - more efficient performance

ethereum

Why Performance so important?

- Blockchain has been used to solve some problems in IoT, cloud computing, distributed computing (systems)
- In industrial applications and cloud computing: blockchain must be integrated into cloud infrastructure
- How to use it?
- Current problem: user/developers lack knowledge of technical performance
- Research Gap: Performance characteristics (permissioned blockchain)



6

Hyperledger

- A Linux Foundation project – www.hyperledger.org
- Open-source collaboration, developing blockchain technologies for business
- Started in 2016
- 135 members in 1st, June.
- Today 6 frameworks and 7 tools, hundreds of contributors

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Hyperledger is an open source collaborative effort created to developing blockchain technologies for business.

Hyperledger

Hyperledger Members

ARTICONF

Premier

General

Associate

Source: https://www.hyperledger.org/about/members

Hyperledger Sawtooth

Sawtooth was started as a research project of Inter lab's in 2014. (Sawtooth Lake).

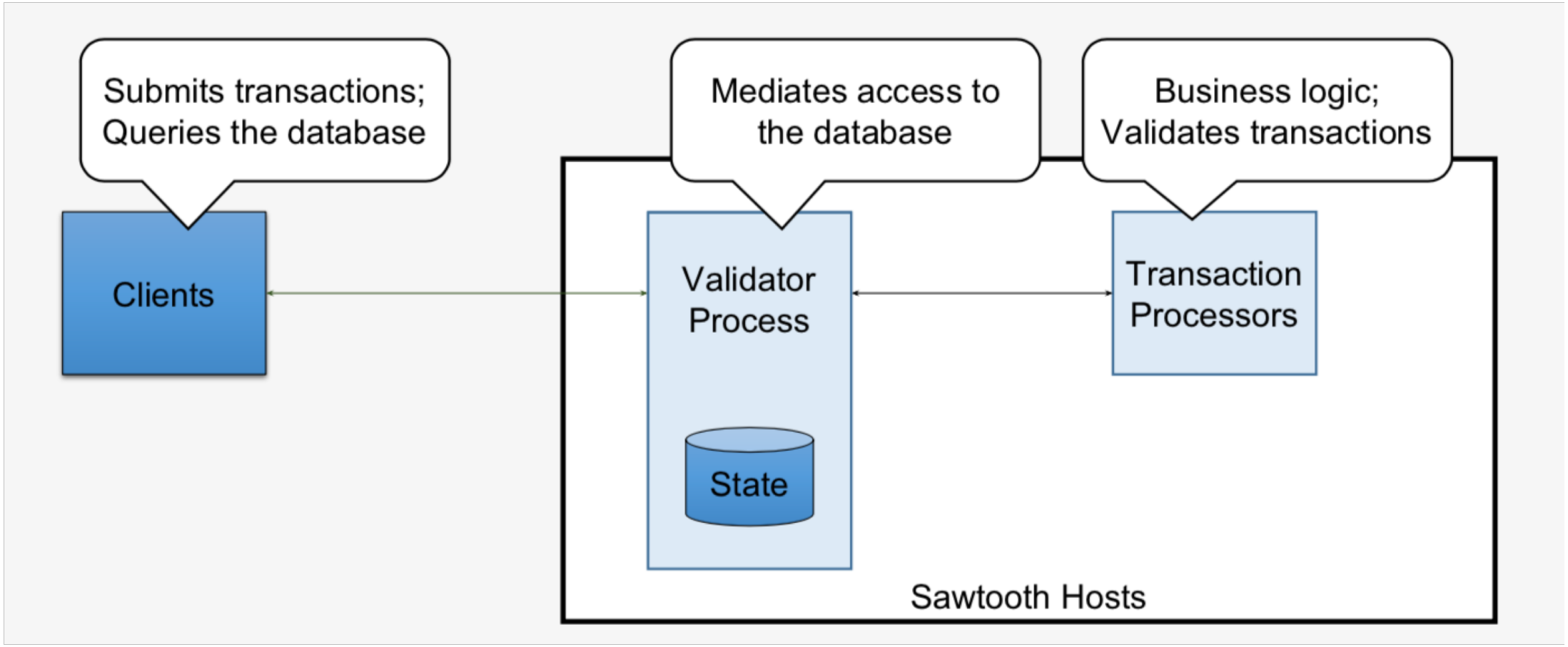
It was donated to Linux foundation in 2016.

It's an enterprise blockchain which aims to provide privacy, security, and scalability service.

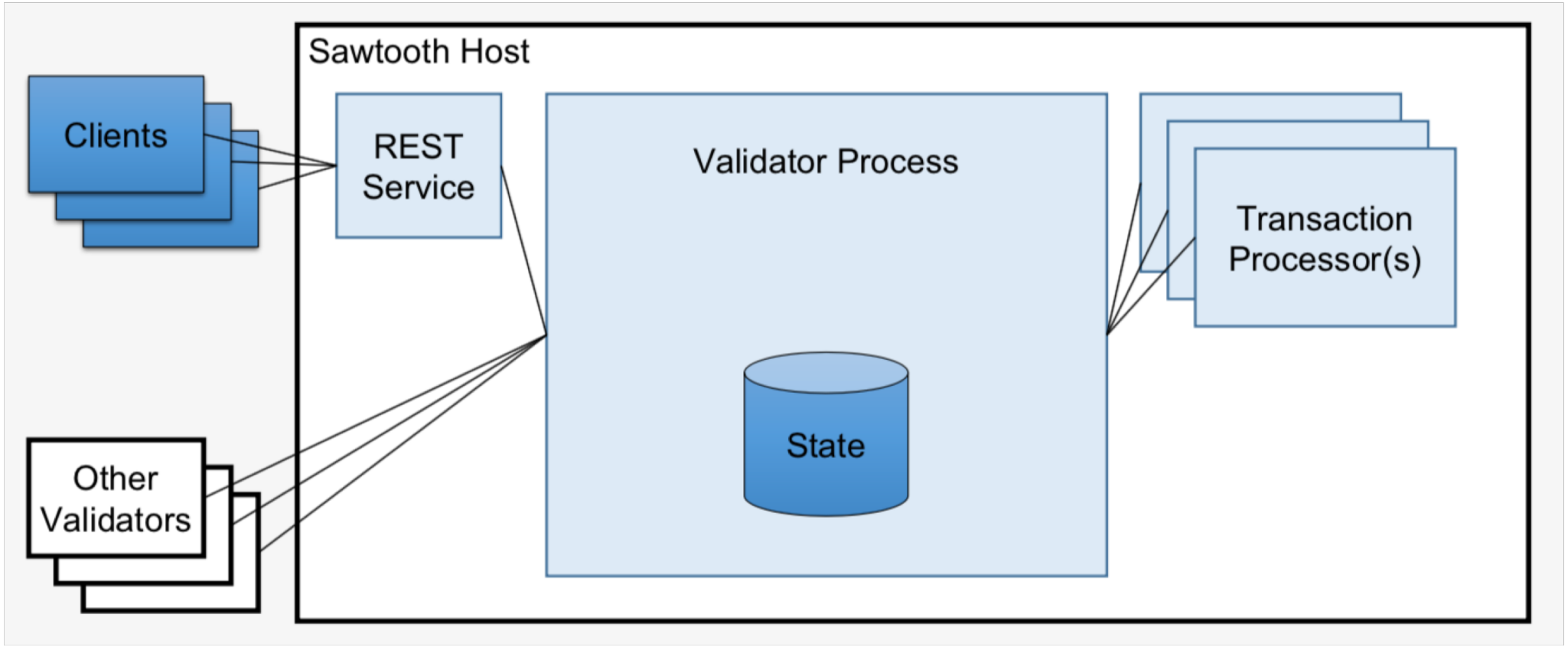
- **Scalable, highly modular architecture**
- **Clear separation between network/application layers**
- **Pluggable consensus algorithms**
- **Multiple programming language support**

Bring the benefits of blockchain to business

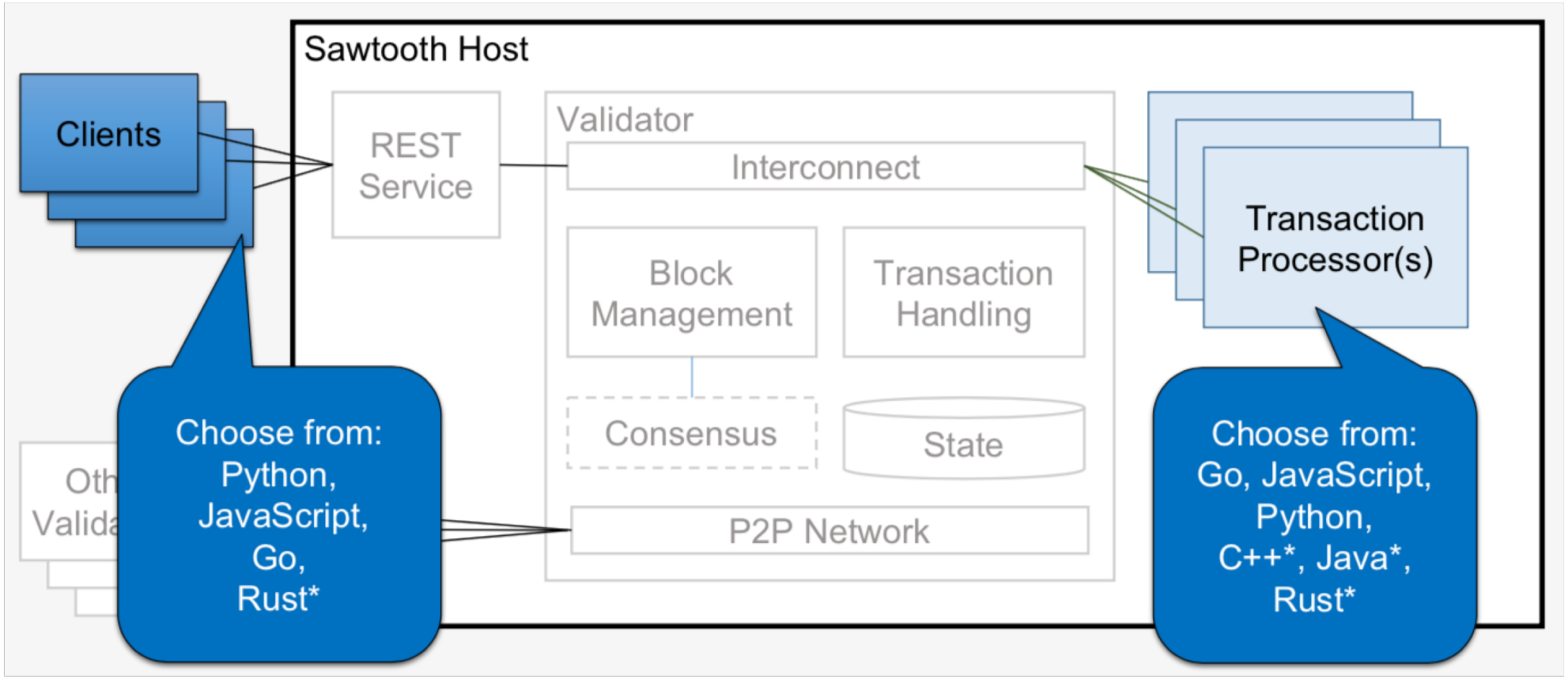
Sawtooth Architecture



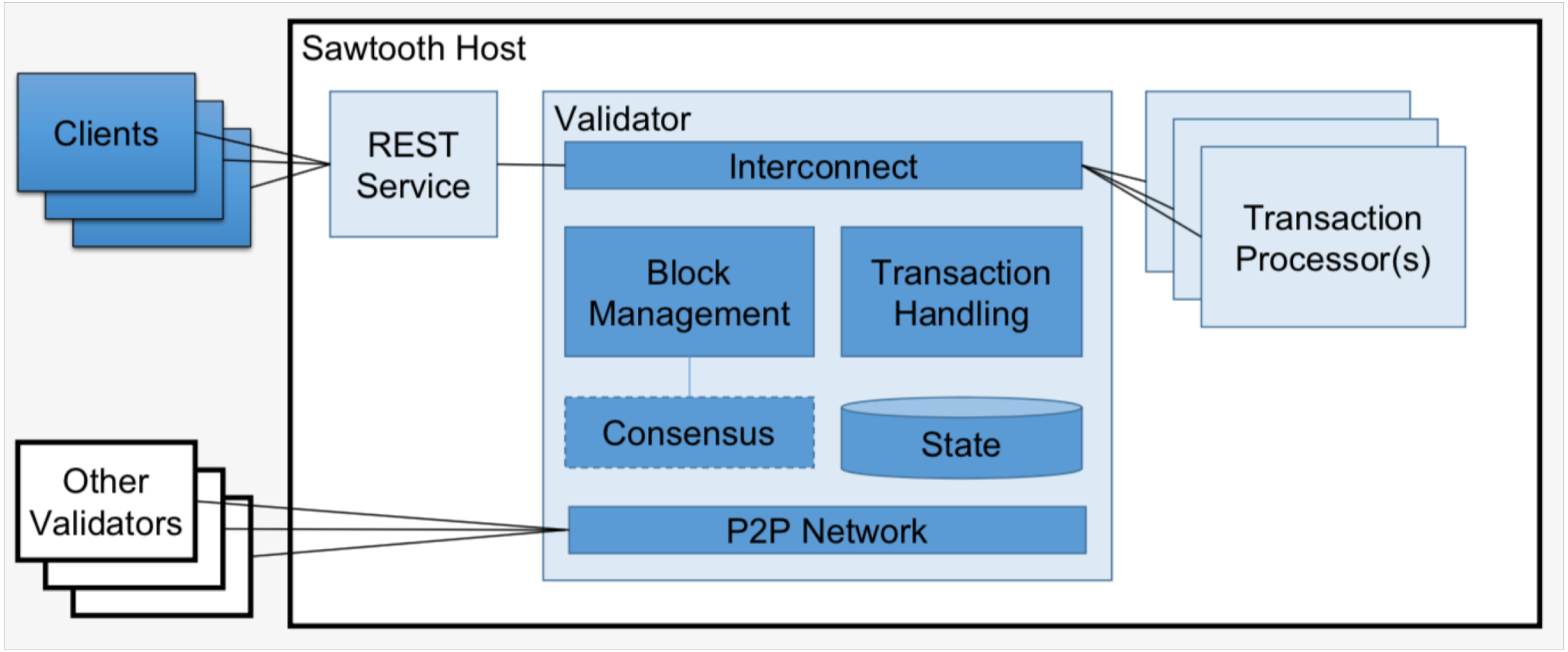
Sawtooth Architecture



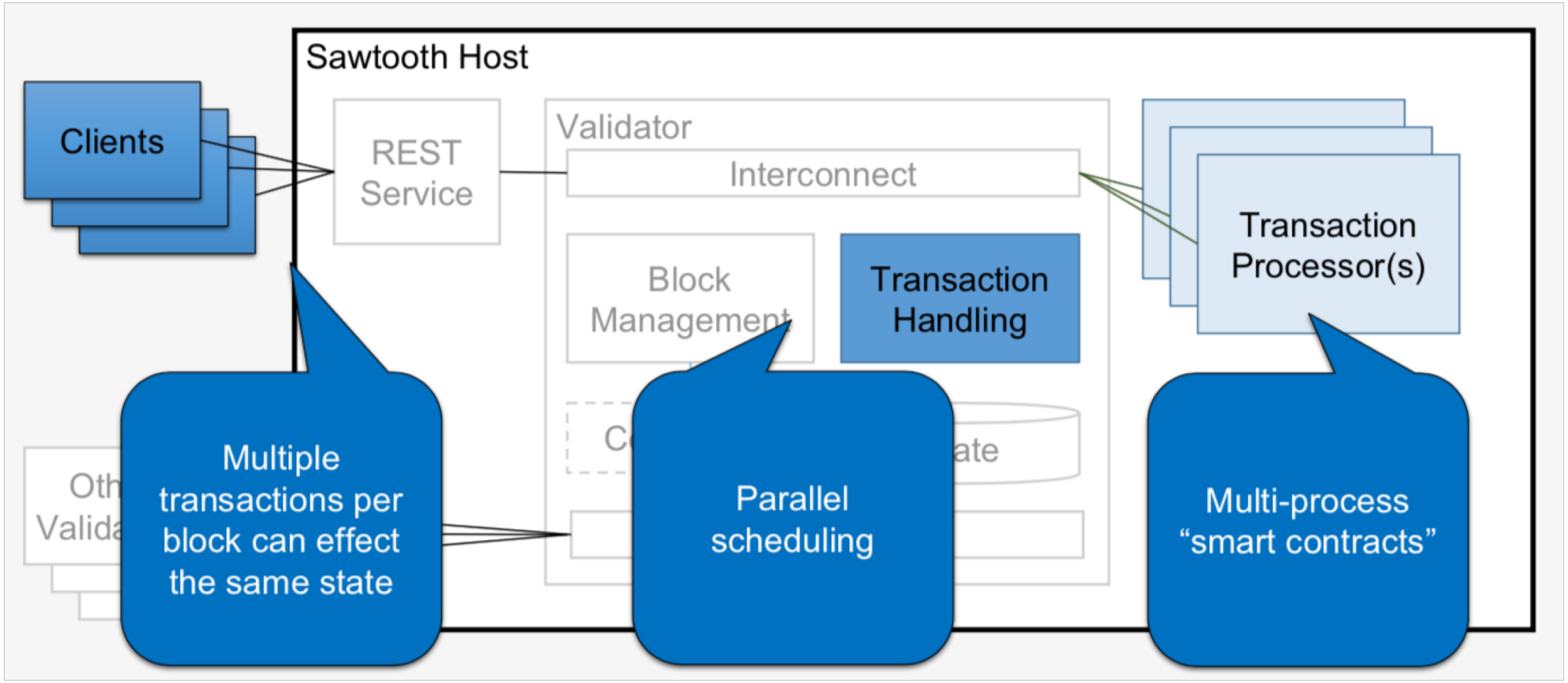
Features: Multi-Language Support



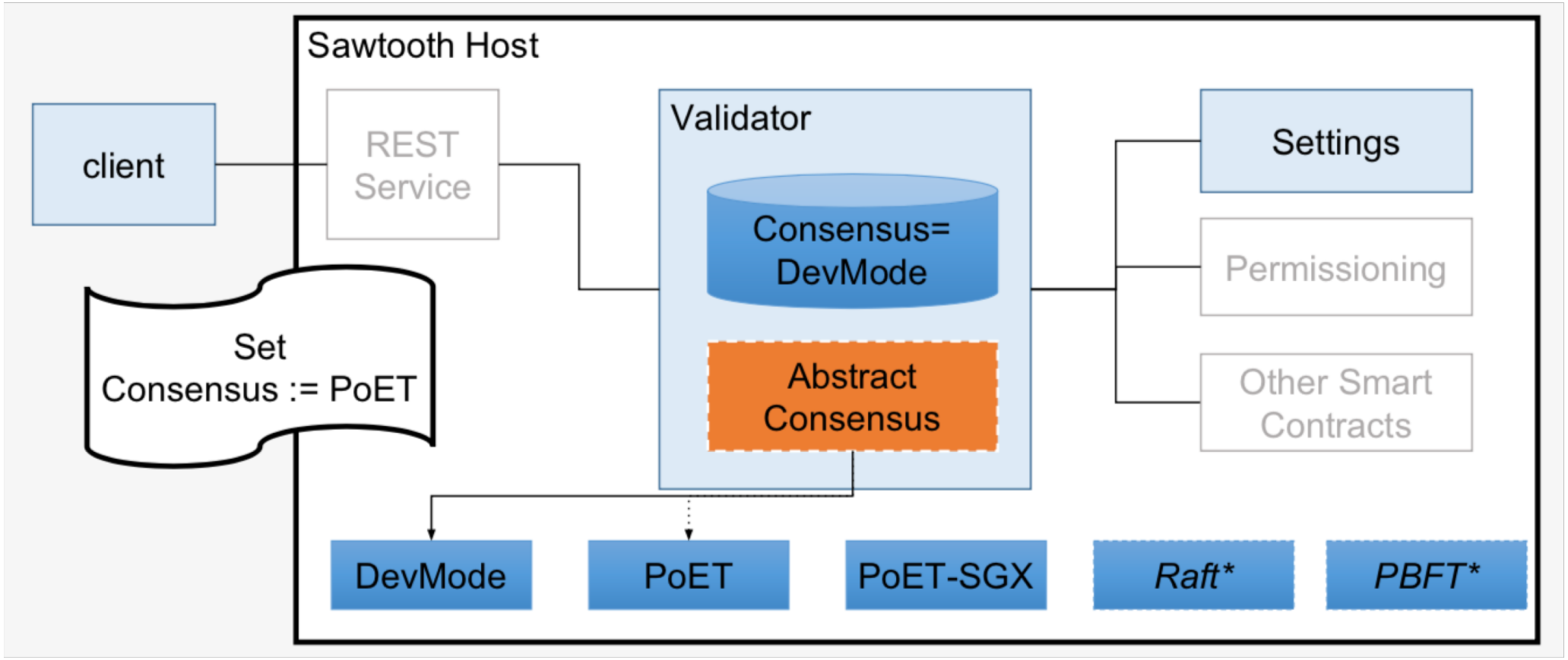
Sawtooth Architecture



Features: Parallel Execution



Features: Dynamic Consensus



Proof of Elapsed Time (PoET)

Scalable	
Latency	X
Scalability	XXX
Trust	X

- Lottery based consensus
- Hardware based
- Uses a TEE (Trusted Execution Environment), i.e. Intel SGX
- How does it work?
  - Every validator requests a wait time from an enclave a (trusted function)
  - Enclave provides them with a random timer
  - Validator with the shortest time wins, they're chosen as the leader
  - Enclave checks the timer, to see if they waited their specified time, and if they did they can claim leadership

Experiment Implementation

Source: https://github.com/zh9314/CloudsStorm

Sawtooth Blockchain

CloudsStorm

Cloud Service Provider

InfluxDB

Grafana

Experiment Implementation

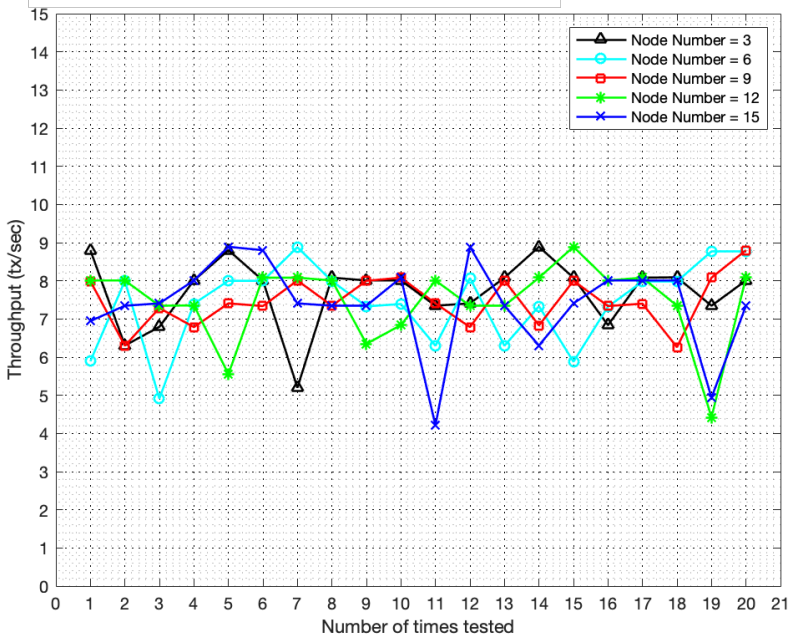
1. Performance Scalability: how is the scalability of the sawtooth blockchain (within PoET consensus)?
2. Performance Stability: what are the factors that influence the stability of the sawtooth blockchain?
3. Performance Configuration: can the performance of the sawtooth blockchain be influences/optimized with some parameters?

Performance analysis

VM Node Numbers

- Different **VM Nodes (3-15)** with the **same workload (1000).**
- Fixed **Input Transaction Rate (9 tps)**
- No significant change in the average performance
- This indicates a trend that Sawtooth (PoET) is able to keep scalable to fit a large-scale network.

$$Throughput = \frac{total\ committed\ transactions}{total\ time\ taken\ in\ seconds}$$



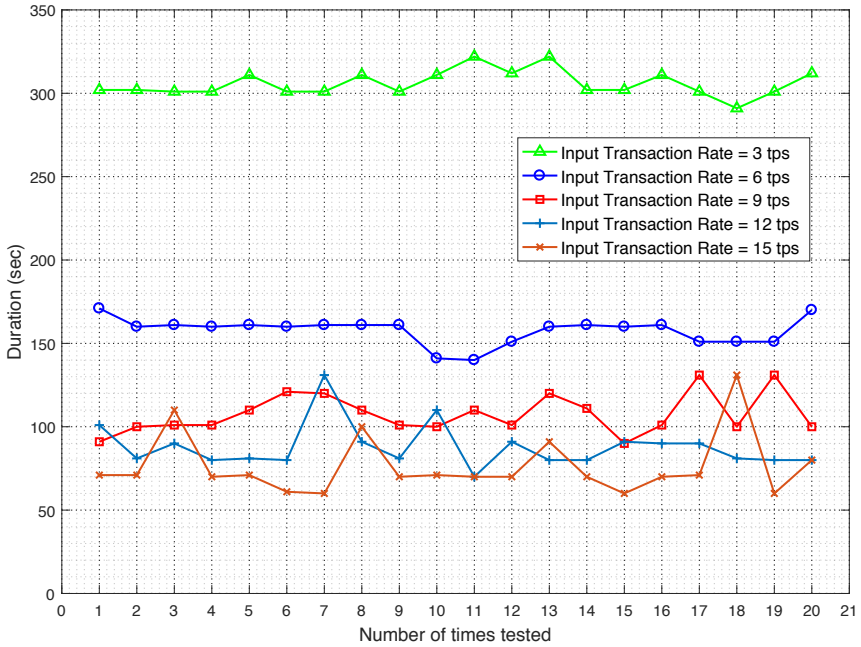
PERFORMANCE VARIATION OF DIFFERENT NUMBER OF VMS.

Number of VMs	Average Throughput	Average Duration	Variance
3	7.75 tps	116.60 sec	16.34
6	7.43 tps	122.20 sec	20.10
9	7.47 tps	119.80 sec	10.58
12	7.46 tps	122.05 sec	22.23
15	7.40 tps	124.00 sec	26.13

Performance analysis

Transactions Input Rate

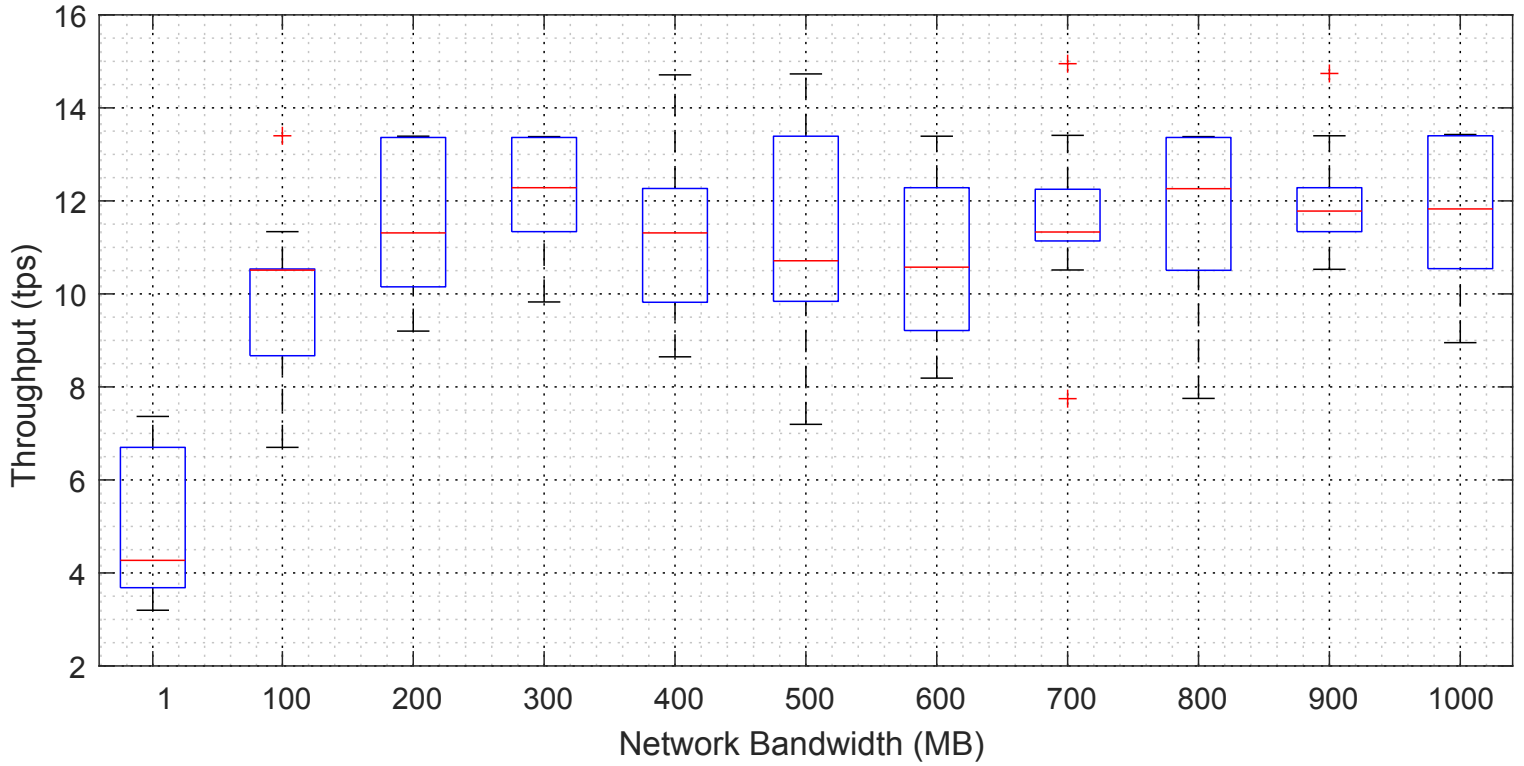
- Different **Input Transaction Rate (3 tps to 15 tps)** with **the same workload (1000)**.
- When the **Input Transaction Rate** is low, the Sawtooth platform performs more stable, but duration time is long.
- Trade-off: the **bigger** of the **Input Transaction Rates** are, the less the stability is.



PERFORMANCE VARIATION OF DIFFERENT INPUT TRANSACTION RATES.

Input Tx Rate	Average Throughput	Average Duration	Variance
3 tps	2.93 tps	305.90 sec	57.39
6 tps	5.67 tps	157.65 sec	60.13
9 tps	8.36 tps	107.50 sec	132.25
12 tps	10.24 tps	87.95 sec	172.25
15 tps	12.03 tps	76.40 sec	316.44

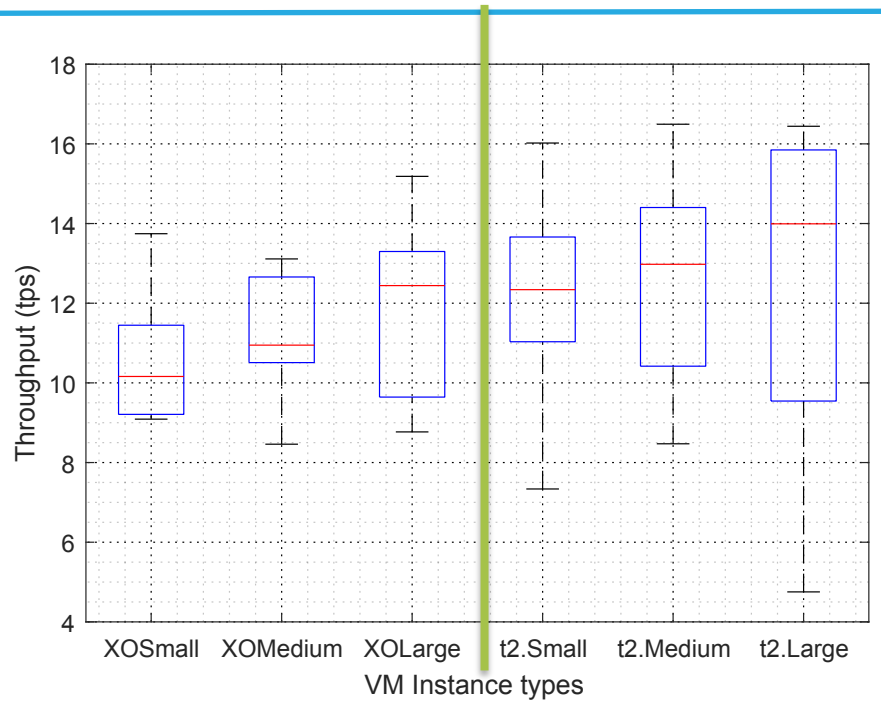
Performance analysis



Network Bandwidth

- Different network **bandwidths** with the **same workload**.
- When bandwidth > 100 MB, the throughput is stable.
- When bandwidth = 100 MB, platform performance begins to show a dropping trend.
- As the bandwidth drops extremely to 1 MB, the platform performance drops to around one-third (4 tps).

Performance analysis



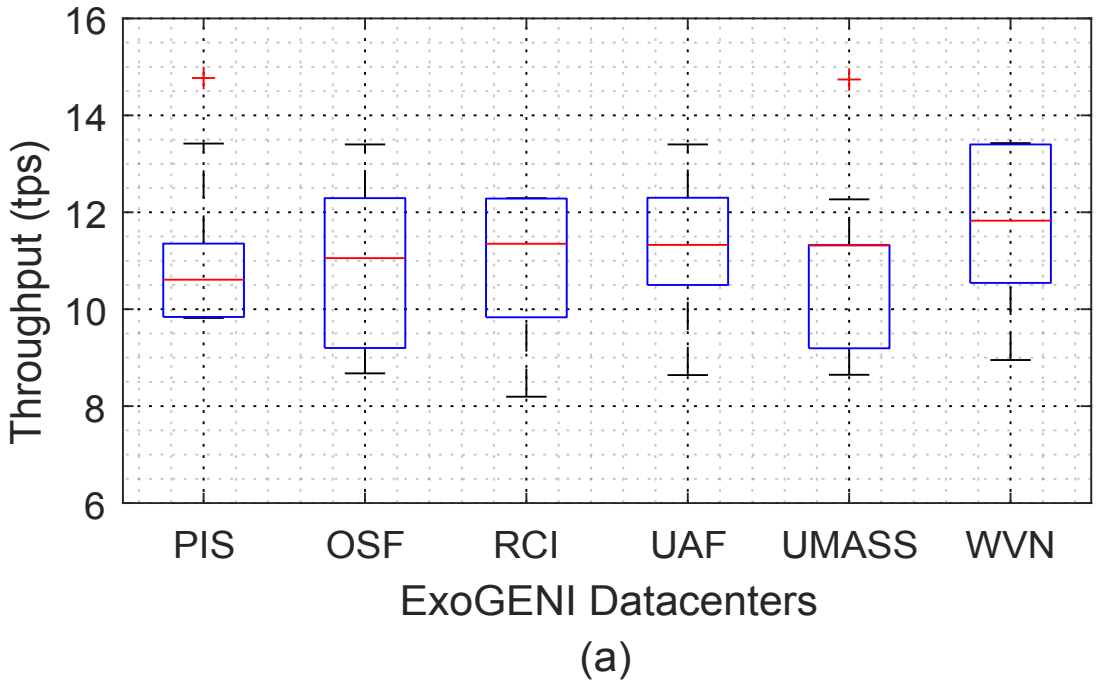
RESOURCE TYPE OFFERED BY EXOGENI AND AWS.

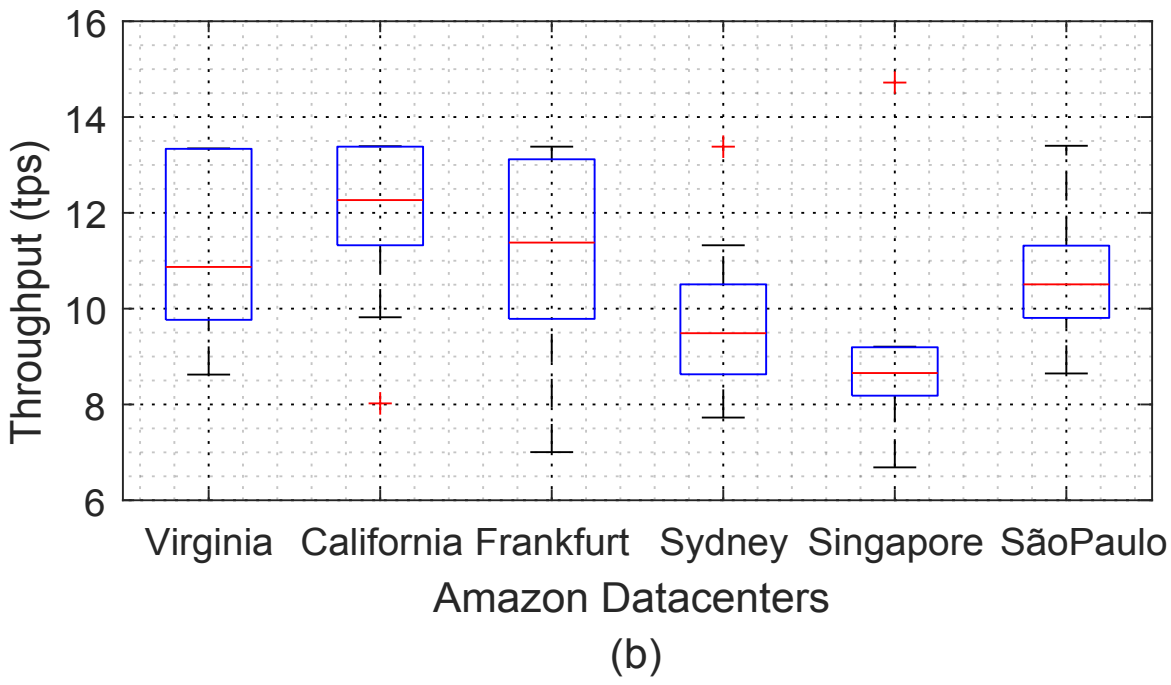
Cloud Provider	Resource Name	CPU Cores	Memory	DISK Size
ExoGENI	XOSmall	1	1G	10G
ExoGENI	XOMedium	1	3G	25G
ExoGENI	XOLarge	2	6G	50G
Amazon	t2.Small	1	2G	8G
Amazon	t2.Medium	2	4G	8G
Amazon	t2.Large	2	8G	8G

VM Instance Types

- When the workload is small, the performance variance is not obvious.
- As the VM instance type changes from small to medium and from medium to large, the average performance has a significant improvement, (both ExoGENI and EC2)
- EC2 outperforms ExoGENI.
- Better instace types has the better resilience and self-recovery ability.

Performance analysis



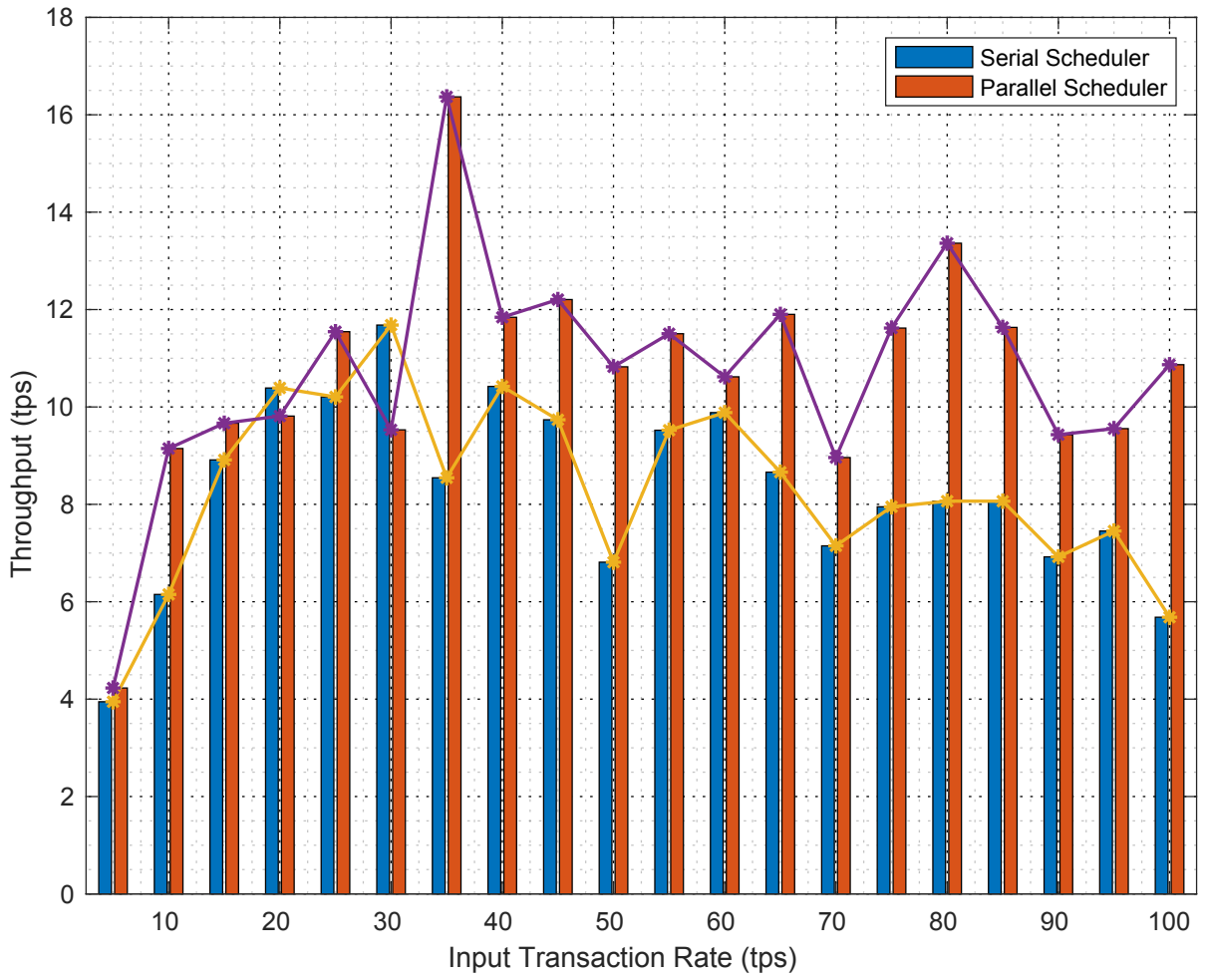


Different Datacenters

- ExoGENI: **UAF** and **PIS** are more stable than others. WVN has the highest average throughput.
- EC2: The California datacenter has the highest and most stable performance。
- An interesting observation -- Singapore

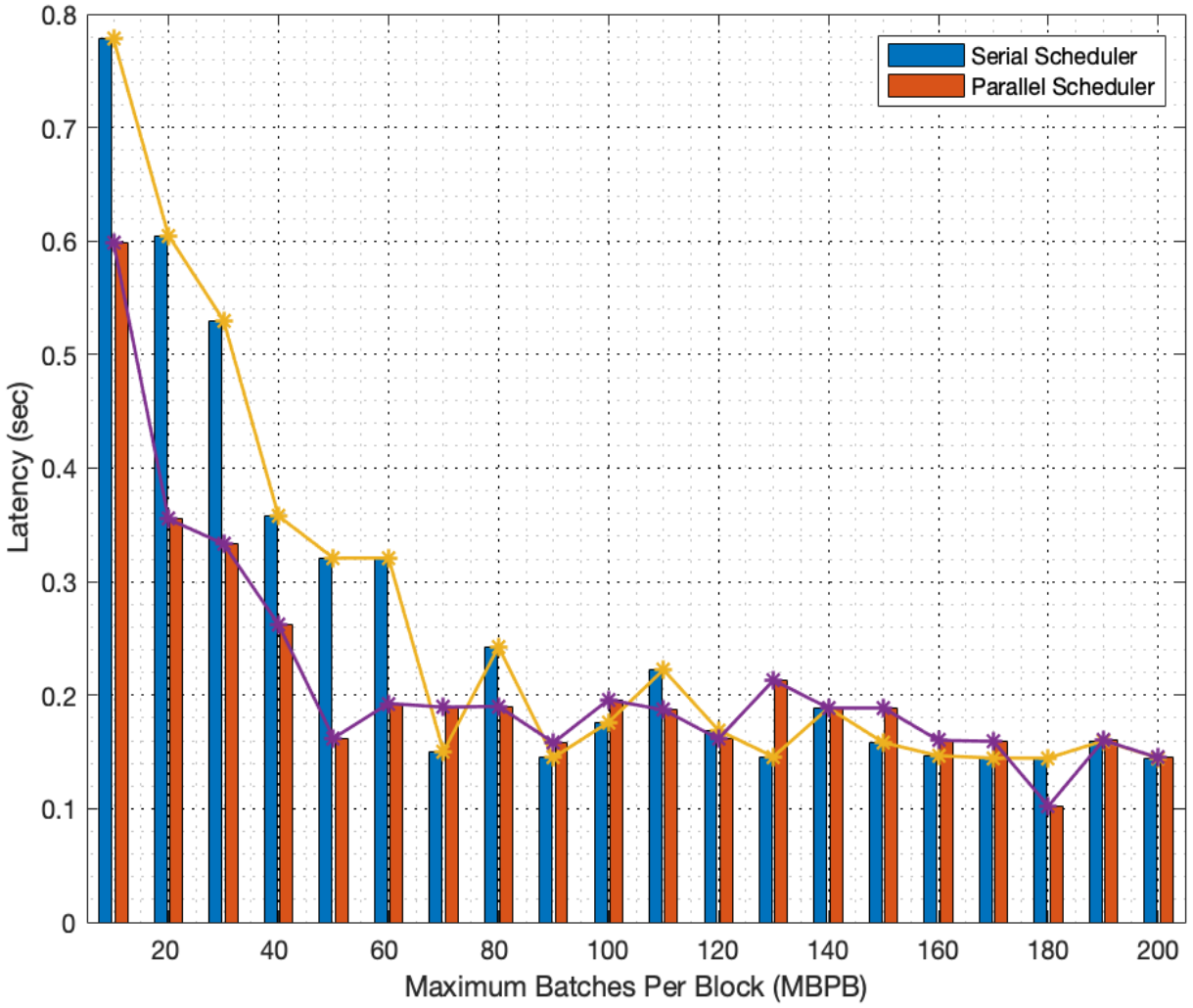
Parameter: Scheduler Type

- When the input rate increases, the performance first rises to the threshold value and then falls.
- Parallel scheduling model has overall better throughput than serial scheduling model.
- The parallel scheduler can increase performance by around 30% when running a series of incremental workloads in our experiments.



Parameter: Maximum Batches Per Block (MBPB)

- Batch is the **atomic unit** of the state change of Sawtooth.
- As the MBPB value increases from 10 to 200, the latency first decreases rapidly and then reaches its threshold and tends to be stable.
- When the MBPB value is less than a certain threshold, the parallel model is significantly better than the serial model.



Conclusions

1. The Hyperledger Sawtooth blockchain (**PoET consensus)** is scalable.
2. Performance stability can be influenced by the **cloud resource conditions** (e.g., Network Bandwidth, VM instance types, datacenters) and different **blockchain workloads** (e.g., Input Transaction Rate).
3. Performance can be improved by optimizing the configuration **parameters** (e.g., Scheduler, MBPB).

Future Work

- Select more **consensus algorithms** (e.g., PBFT, Raft) and more **blockchain platforms** (e.g. Fabric, Ethereum).
- Develop **automatic performance monitoring tools** and integrate them into the sawtooth blockchain platform.
- Design a **programmable Blockchain infrastructure** based on the sawtooth blockchain.

To the best of our knowledge, this is the first study which focuses on the comprehensive performance analysis of **Hyperledger Sawtooth** in a **dynamic cloud environment**.

ARTICONF

https://articonf.eu/

Thank you!
Questions?

ARTICONF

https://articonf.eu/

Features: Dynamic Consensus

ARTICONF

- Nakamoto (lottery) vs. Classical (voting) consensus
- BFT vs. CFT consensus
- PoET

Fault Tolerance	Type	Consensus Algorithm
BFT	Lottery	Proof of Work (PoW) -- classic Bitcoin/Ethereum mining (energy consuming)
BFT	Lottery	Proof of Elapsed Time (PoET) -- SGX
CFT	Lottery	Proof of Elapsed Time (PoET) -- without SGX (simulator)
BFT	Voting	Practical Byzantine Fault Tolerance (PBFT) -- does not scale, $O(N^2)$
CFT	Voting	Raft – uses an elected leader; fast