klaytn

Building and Operating a Public Blockchain: Engineering Perspective

March 22, 2023

Sangmin Seo (Sam), Ph.D.

Director / Klaytn Foundation

About Sam

Sangmin Seo (Sam)

Director / Klaytn Foundation
Seoul National University, Ph.D.

- **(22~) Klaytn Foundation, Director**
- (22~23) Krust Universe, CKO (Chief Klaytn Officer)
- (18~21) GroundX, CTO
- (17~18) Samsung Research
- (14~17) Argonne National Laboratory
- (12~14) ManyCoreSoft, CEO

Outline

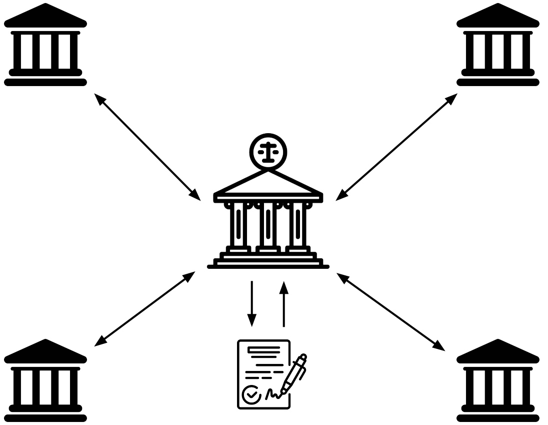
- Problems that Blockchain is Solving
- Mainnet Development & Operation
- What’s Next?

Problems that Blockchain is Solving

klaytn

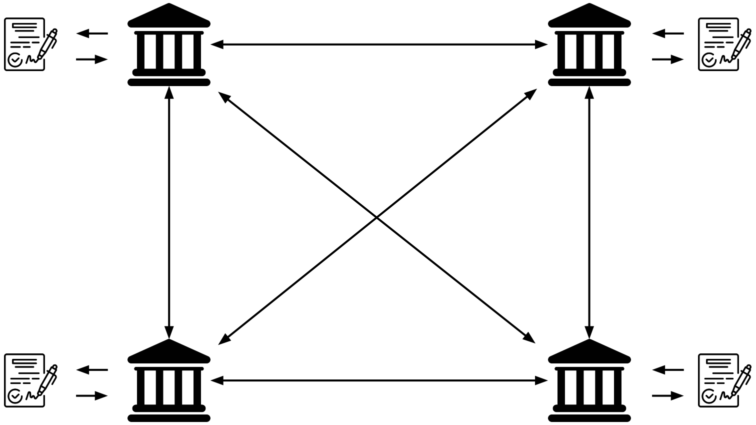
Trust Problem

Can we create trust among untrusted entities in an algorithmic way?



Traditional Approach

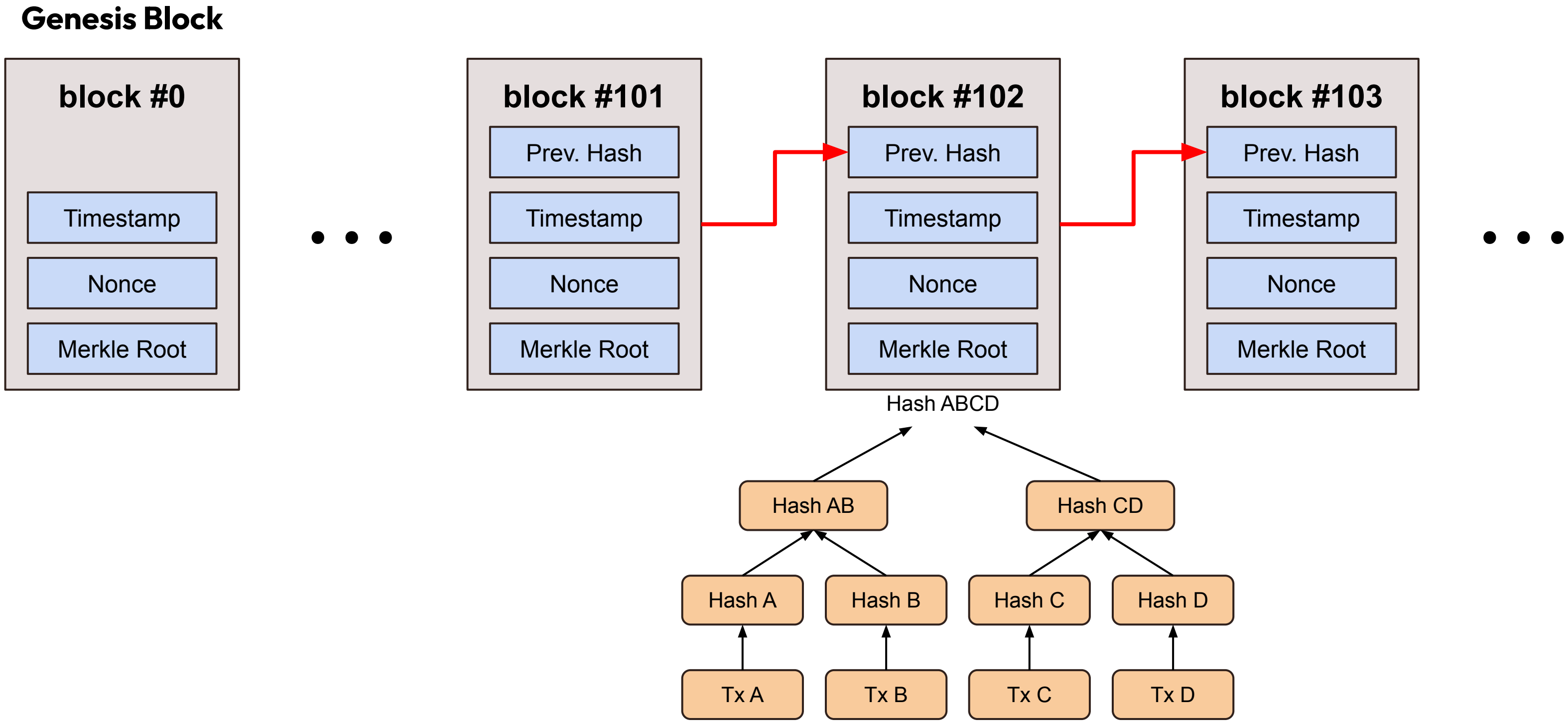
Database is controlled
by a central and trusted third-party.



Blockchain Approach

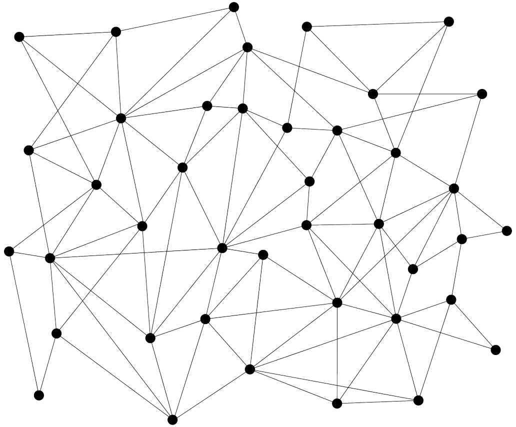
Each participant has a copy of the database,
ensuring immutability.

Blockchain’s Data Structure - Bitcoin

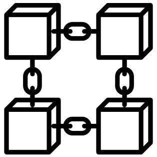


Blockchain from the Perspective of Computer Science

Problem: How can untrusted nodes in a network create an immutable chain of data blocks?



P2P Network



Blockchain

Data Structure

- Block: a container of transactions
- Blockchain: a singly-linked list of blocks

Smart Contract

- Programming language
- Virtual machine

Architecture

- Distributed system with P2P network

Consensus Algorithm

- Distributed consensus algorithm that can solve the Byzantine generals problem*

* The Byzantine Generals Problem, Lamport *et al.*, TOPLAS ’82

Blockchain Trilemma

SECURITY

DECENTRALIZATION

SCALABILITY

No blockchain can have all 3 attributes;
They must choose **2 out of 3** of the attributes.

Blockchain Trilemma

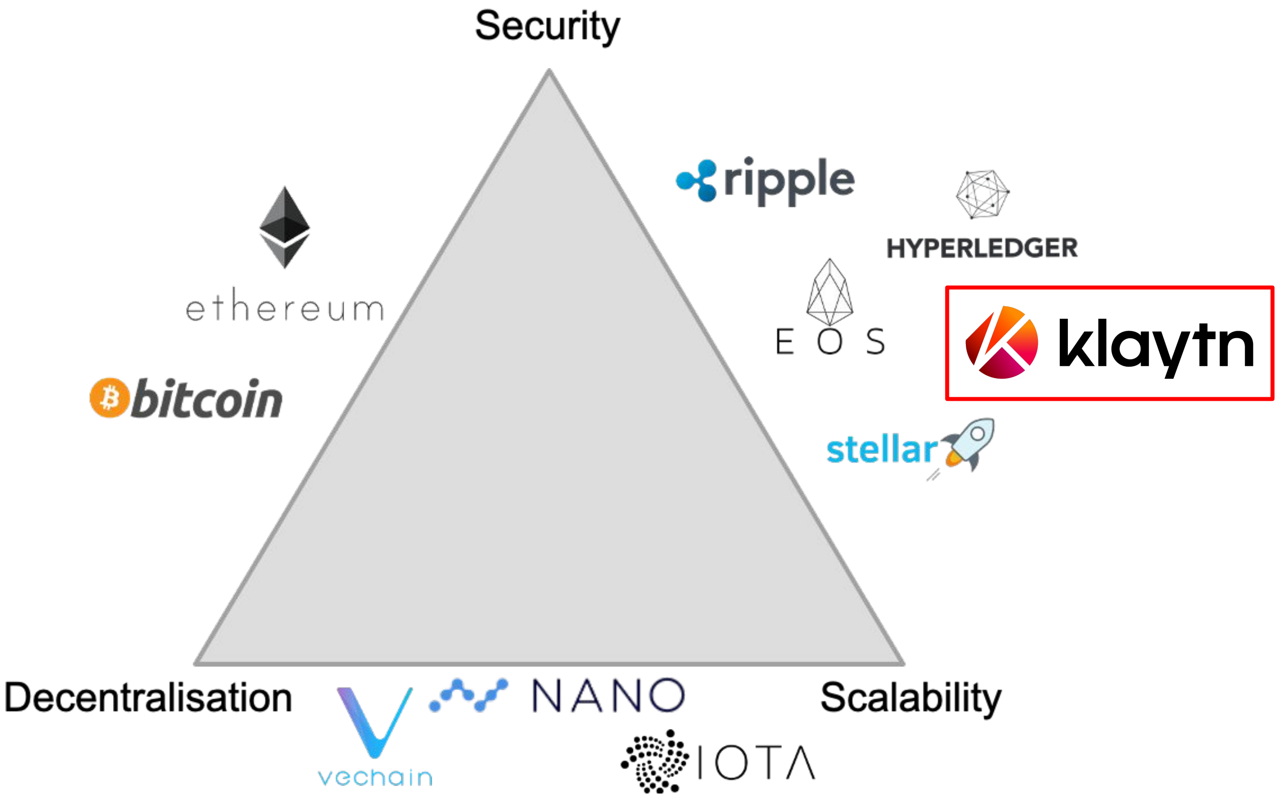
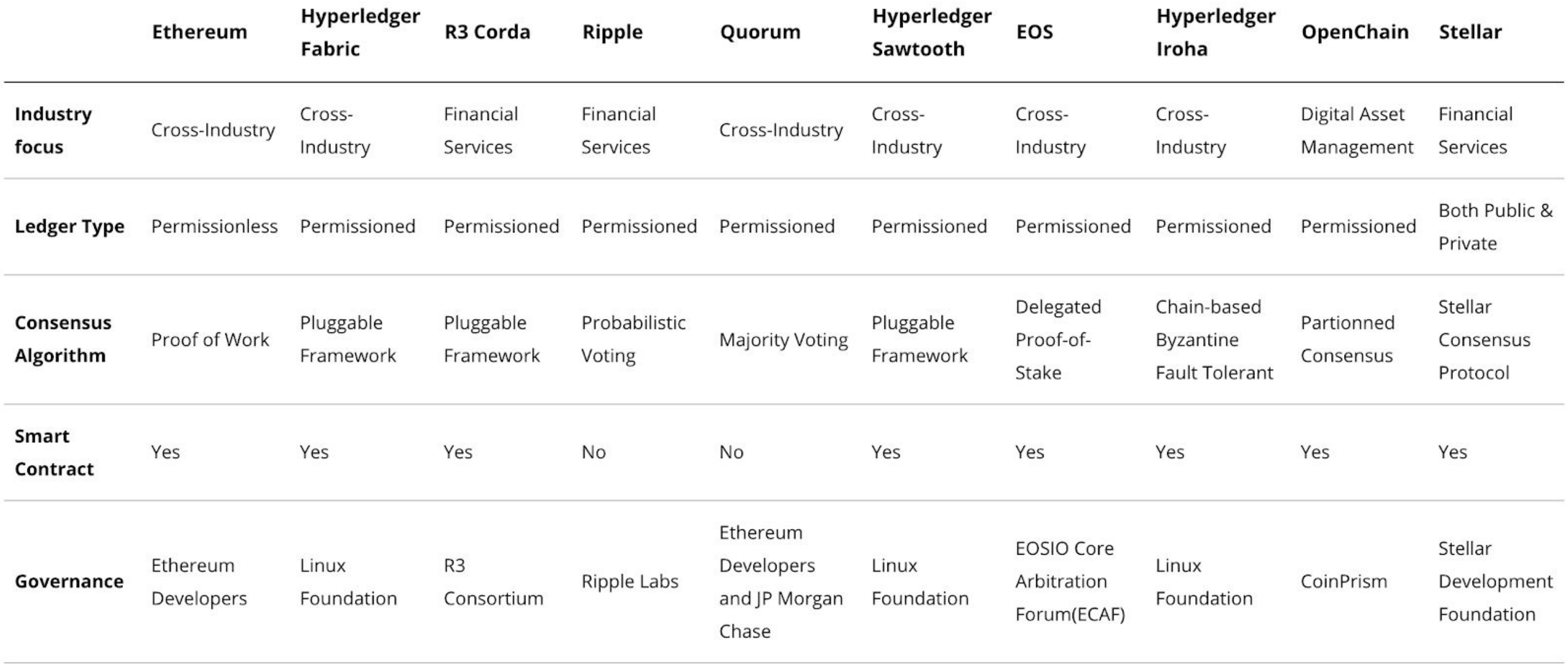


figure: https://blog.fingo.pl/blockchain-trilemma/

Blockchain Platform

A platform that provides blockchain while enabling blockchain-based application development



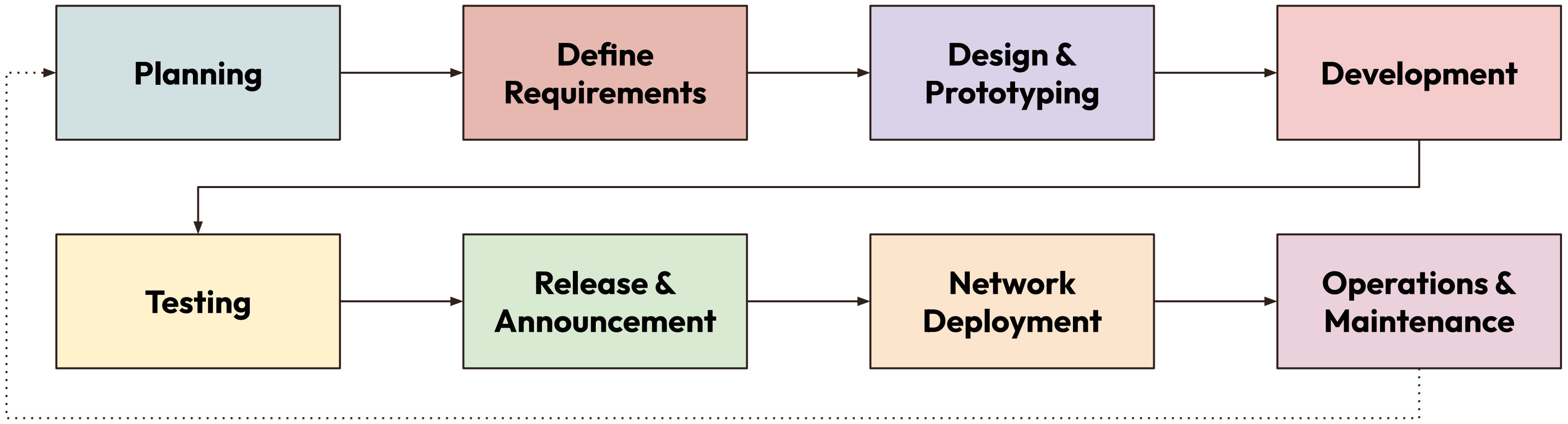
	Ethereum	Hyperledger Fabric	R3 Corda	Ripple	Quorum	Hyperledger Sawtooth	EOS	Hyperledger Iroha	OpenChain	Stellar
Industry focus	Cross-Industry	Cross-Industry	Financial Services	Financial Services	Cross-Industry	Cross-Industry	Cross-Industry	Cross-Industry	Digital Asset Management	Financial Services
Ledger Type	Permissionless	Permissioned	Permissioned	Permissioned	Permissioned	Permissioned	Permissioned	Permissioned	Permissioned	Both Public & Private
Consensus Algorithm	Proof of Work	Pluggable Framework	Pluggable Framework	Probabilistic Voting	Majority Voting	Pluggable Framework	Delegated Proof-of-Stake	Chain-based Byzantine Fault Tolerant	Partionned Consensus	Stellar Consensus Protocol
Smart Contract	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Governance	Ethereum Developers	Linux Foundation	R3 Consortium	Ripple Labs	Ethereum Developers and JP Morgan Chase	Linux Foundation	EOSIO Core Arbitration Forum(ECAF)	Linux Foundation	CoinPrism	Stellar Development Foundation

* https://www.leewayhertz.com/blockchain-platforms-for-top-blockchain-companies/

Mainnet Development & Operation

klaytn

Mainnet (Blockchain Platform) Development Cycle

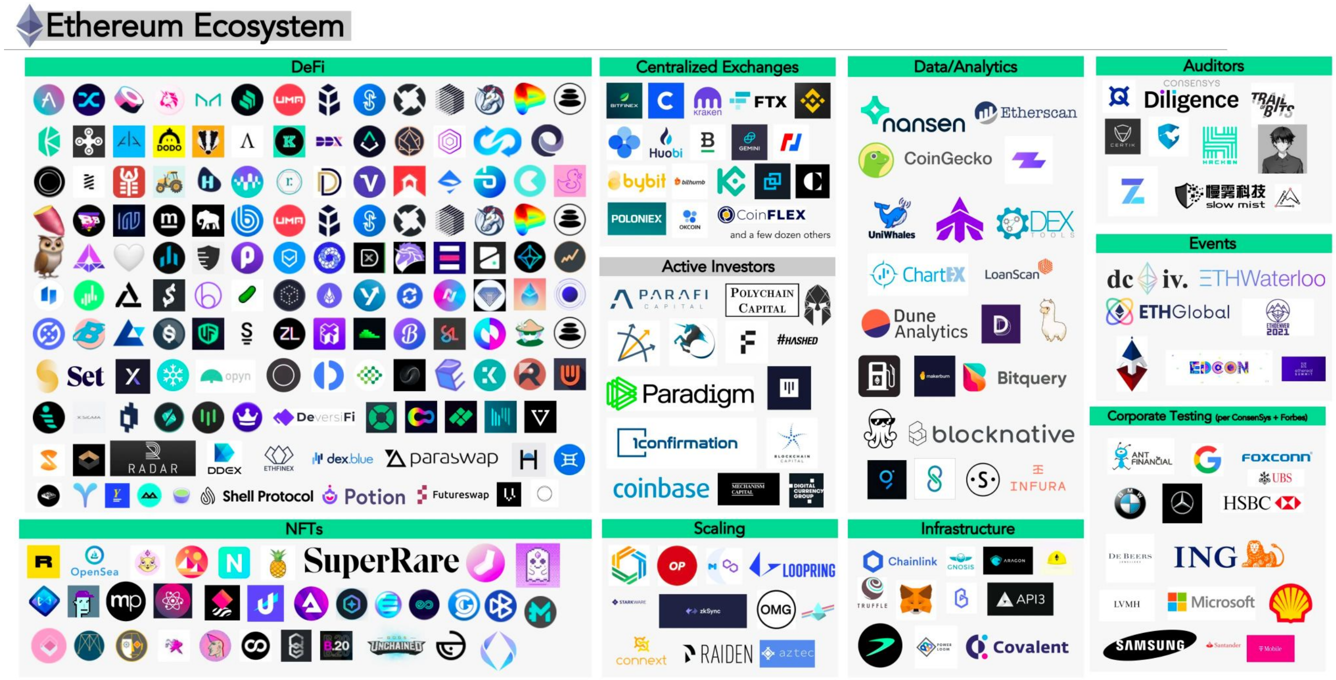


Similar to the typical software development cycle, but there are some unique challenges

- **Backward compatibility**: new versions have to handle old chain data.
- **Error-free code**: the mainnet must not be halted due to any error → this requires tremendous testing effort.
- **Asynchronous network deployment/update**: nodes are operated by different entities and the mainnet should not stop → deploying a new version needs a special process (e.g., rolling update) and communication between node operators.

Mainnet Scope & Ecosystem

Today, we will focus on **the protocol/mainnet development and operation**



https://www.reddit.com/r/ethereum/comments/m9sgqj/ethereum_ecosystem/

Klaytn in a Nutshell

- **EVM-compatible public layer 1 blockchain**
- **Launched the mainnet Cypress in June 2019**
- **Istanbul BFT-based consensus algorithm**
  - 30 global companies are participating in the governance council and operating consensus nodes (as of Feb 2023)
- **Unique account & transaction model**
- In line with the **Klaytn 2.0 initiative**, focusing on **Metaverse** and **games** while expanding the DeFi and NFT ecosystem

1 second

Time to generate and finalize a block

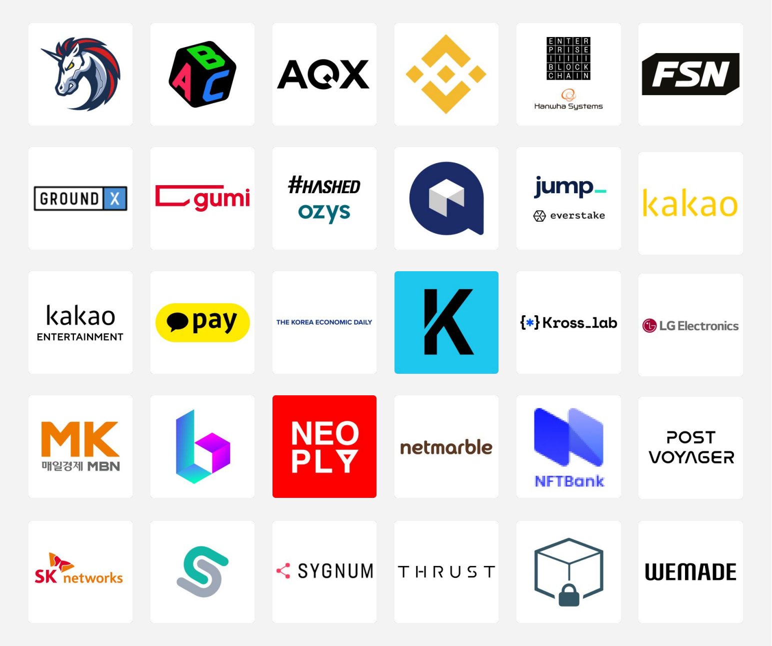
4,000+ TPS*

*Transaction Per Second for KLAY transfers

Low tx fee

< 1/1000 gas cost compared to Ethereum

Klaytn Governance Council (as of Feb 2023)



Challenges on Scalability / Performance

Klaytn Cypress Performance

Latency

1 sec

block interval

Enables mobile
app-like performance

Throughput

4,000

TPS

Supports production-grade
enterprise usage

Performance Comparison (2019)

	Klaytn	Bitcoin	Ethereum	Ripple	EOS	Stellar
Time to finality	1 sec	15 min	6 min	4 sec	180 sec	2-5 sec
Transactions per second (TPS)	4,000	7	15	1,500	3,000	1,000

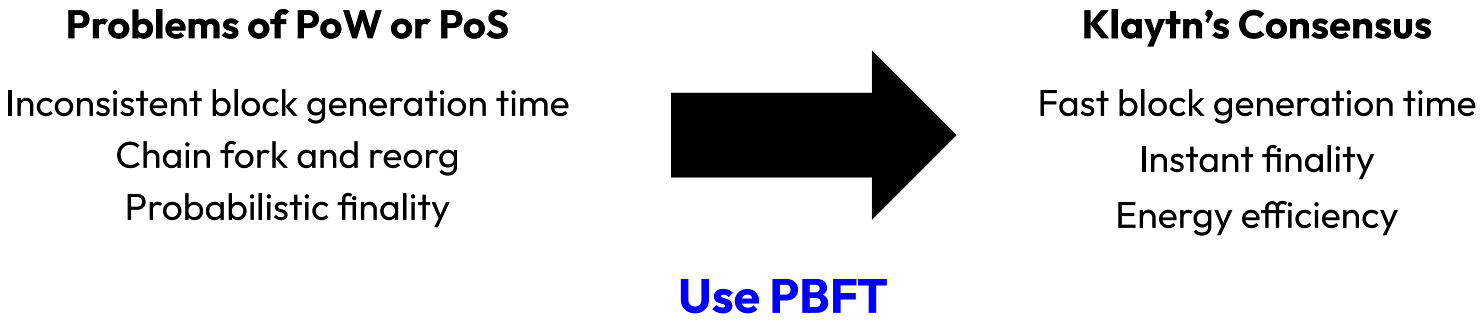
Blockchains for Supply Chain Management: Architectural Elements and Challenges Towards a Global Scale Deployment. Logistics, Litke et. al. (2019).
https://medium.com/perlin-network/bite-sized-2-why-is-tps-time-to-finality-important-bd01baffdf05
https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times

Performance Comparison (2022)

	Klaytn	Solana	BSC (Binance Smart Chain)	Polygon	Polkadot	Avalanche	Fantom
Transactions per second (TPS)	4,000	65,000	100	10,000	1,500	4,500	10,000
Block time	1 sec	0.4 sec	3 sec	2 sec	2-3 sec	1-5 sec	1-2 sec
Time to finality (Confirmation time)	1 sec	7-10 sec	3 sec	2 sec	1 min	2 sec	< 1sec

https://www.reddit.com/r/solana/comments/pdxw84/solana_vs_other_blockchain_platforms/
https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times

Klaytn's Consensus Algorithm



Problems of PoW or PoS

Inconsistent block generation time

Chain fork and reorg

Probabilistic finality

Use PBFT

Klaytn's Consensus

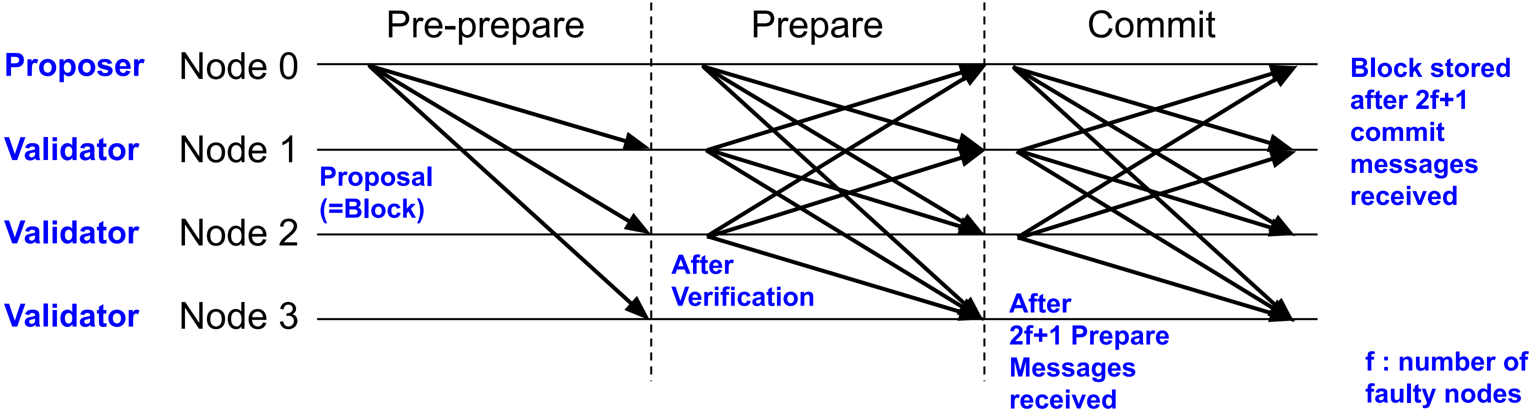
Fast block generation time

Instant finality

Energy efficiency

PBFT (Practical Byzantine Fault Tolerant) Consensus

- **# of faulty nodes should be smaller than ⅓ of all nodes.**
- **# of message for N nodes => O(N^2)**
- **On consensus, immediate finality is guaranteed**



How Klaytn's Consensus Overcomes PBFT's Weakness

Finality

Strong consistency guarantee → **1-second Finality**

Scalability

O(n^2) communication complexity → **Committee Selection**

O(n) verification complexity → **Committee Selection**

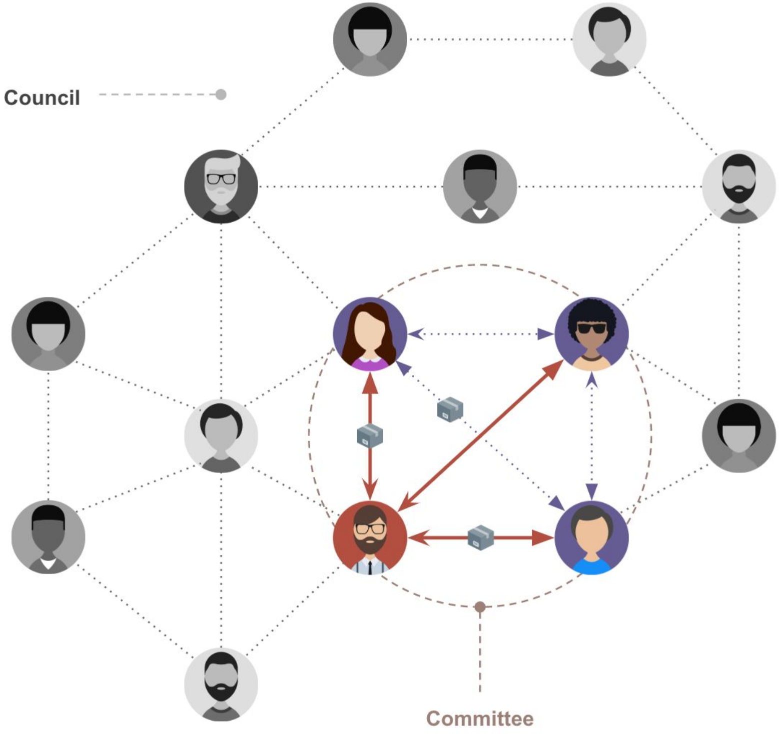
Sybil attack

Create multiple pseudonymous identities to subvert the 3f + 1 of PBFT

→ **Permissioned Council**

Klaytn's Securely Scalable BFT

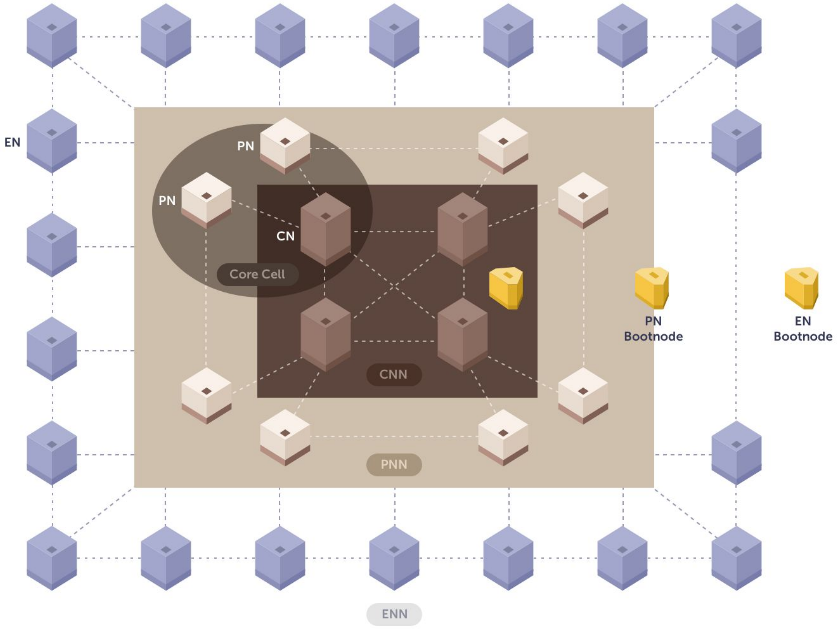
- Trustful node operators form a network called **Governance Council (GC)**
- For each block, Klaytn randomly selects a subset of the council;
  we call this subset a **Committee**
- Klaytn runs IBFT on a chosen Committee to achieve fast, efficient consensus



How to Secure Consensus Nodes in Open Network?

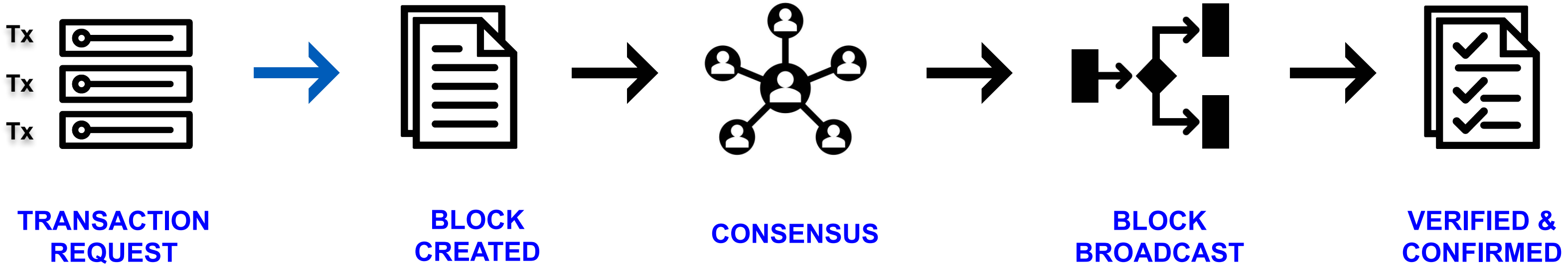
Cypress Network Architecture

- **Tiered hybrid networks** with **role-based node types** for **fast and secure block generation**
  - ***Permissioned*** Consensus Node Network (CNN)
    - Block generation and validation
    - Implemented securely scalable BFT
  - ***Permissioned*** Proxy Node Network (PNN)
    - Propagate txs from EN to CN
    - Propagate blocks from CN to EN
  - ***Permissionless*** Endpoint Node Network (ENN)
    - Provide API for users/services
    - Propagate txs to PN and receive results

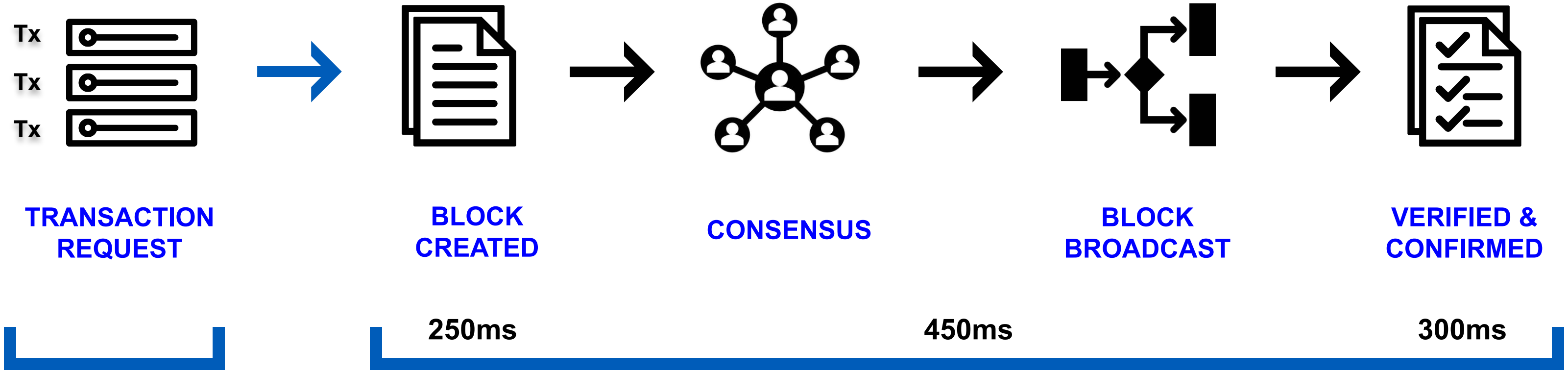


Cypress Network Architecture

From Transaction Request to Block Confirmation



1-Second Block Interval



TRANSACTION REQUEST

BLOCK CREATED

CONSENSUS

BLOCK BROADCAST

VERIFIED & CONFIRMED

250ms

450ms

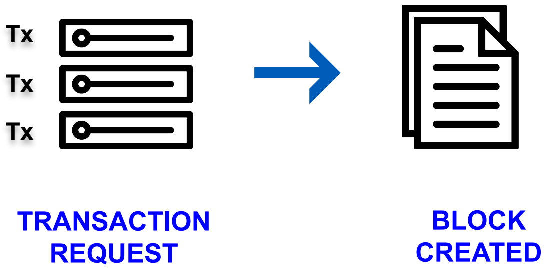
300ms

External Process

1 second (On-Chain)

Block Creation and Verification

Create a block within 250 ms



- When creating a block, the time taken for validation and execution of all transactions in the block should be smaller than 250 ms.
- Block validation is very similar to block creation.

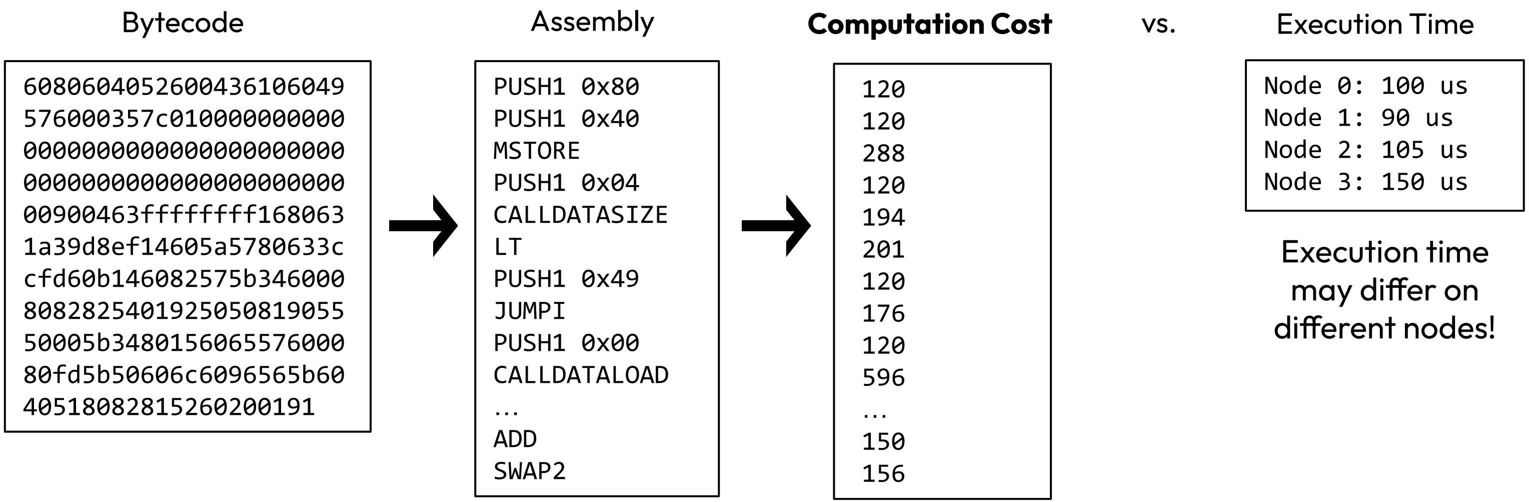
Challenges

- **How can we limit the execution time of a single transaction?**
- **How many transactions can be included in a block to meet the time restriction of 250 ms?**

Limit the Transaction Execution Time (I)

Limit the execution time of a single transaction by its computation cost

Nodes cannot make a consensus on time. → **Need a value that is deterministic and verifiable**



Bytecode

```
6080604052600436106049
576000357c010000000000
0000000000000000000000
0000000000000000000000
00900463ffffffff168063
1a39d8ef14605a5780633c
cfd60b146082575b346000
8082825401925050819055
50005b3480156065576000
80fd5b50606c6096565b60
40518082815260200191
```

Assembly

```
PUSH1 0x80
PUSH1 0x40
MSTORE
PUSH1 0x04
CALLDATASIZE
LT
PUSH1 0x49
JUMPI
PUSH1 0x00
CALLDATALOAD
...
ADD
SWAP2
```

Computation Cost

```
120
120
288
120
194
201
120
176
120
596
...
150
156
```

vs.

Execution Time

```
Node 0: 100 us
Node 1: 90 us
Node 2: 105 us
Node 3: 150 us
```

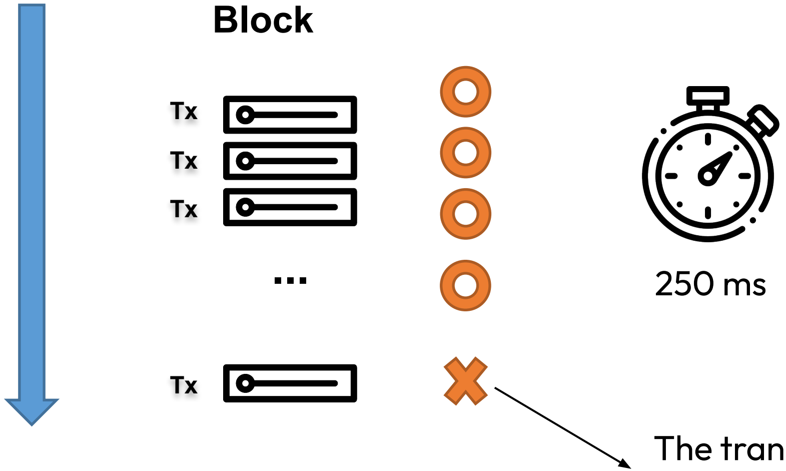
Execution time may differ on different nodes!

Define computation cost for each opcode and set the limit for the sum of computation costs for all opcodes.

* https://docs.klaytn.foundation/content/klaytn/design/computation/computation-cost

Limit the Transaction Execution Time (II)

Limit the execution time of all transactions in a block by timer

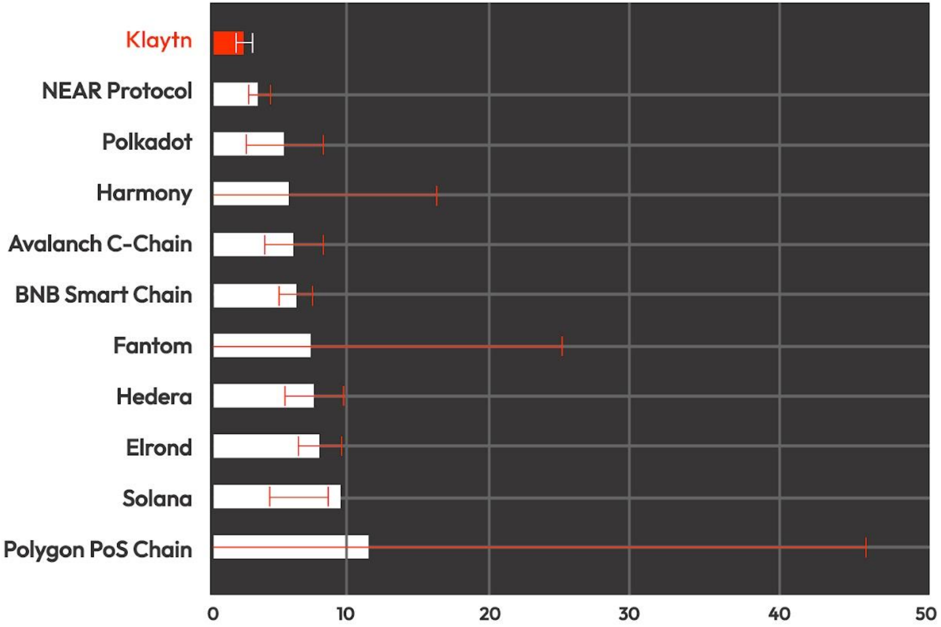


The max. number of transactions included in a block is limited by the total execution time.

The transaction cut by the total execution time limit will be retried in the next blocks.

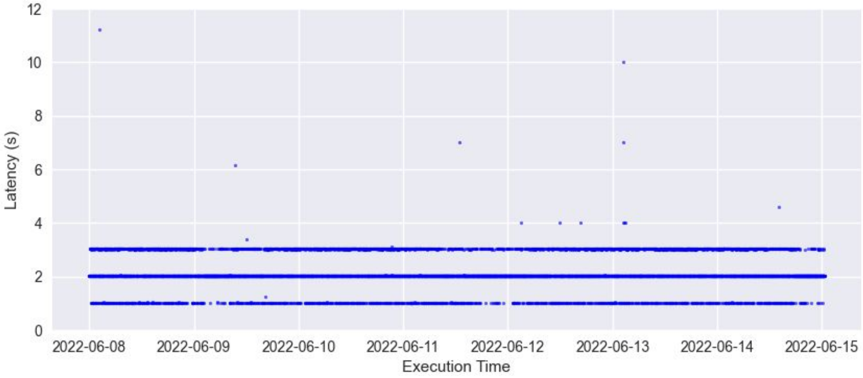
Only Block Time is Important?

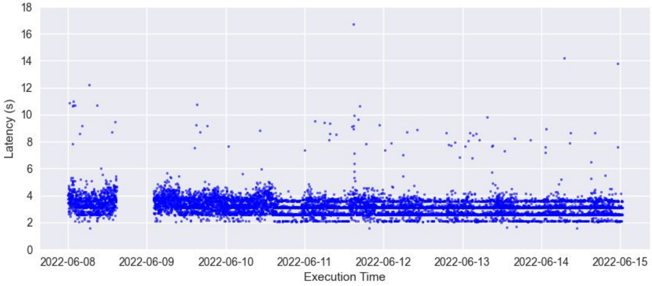
For better UX, the transaction latency is more important!

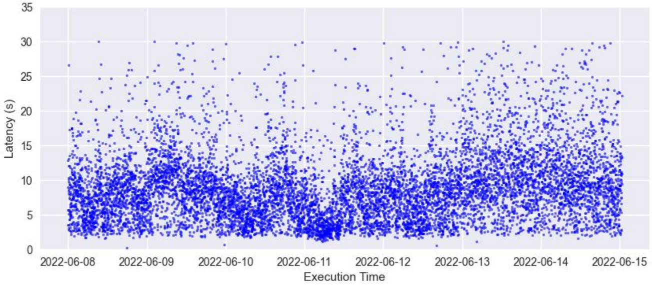


Source: *Klaytn's Comparison of Blockchain Network Latencies*

The Stability of Transaction Latency is also Important!





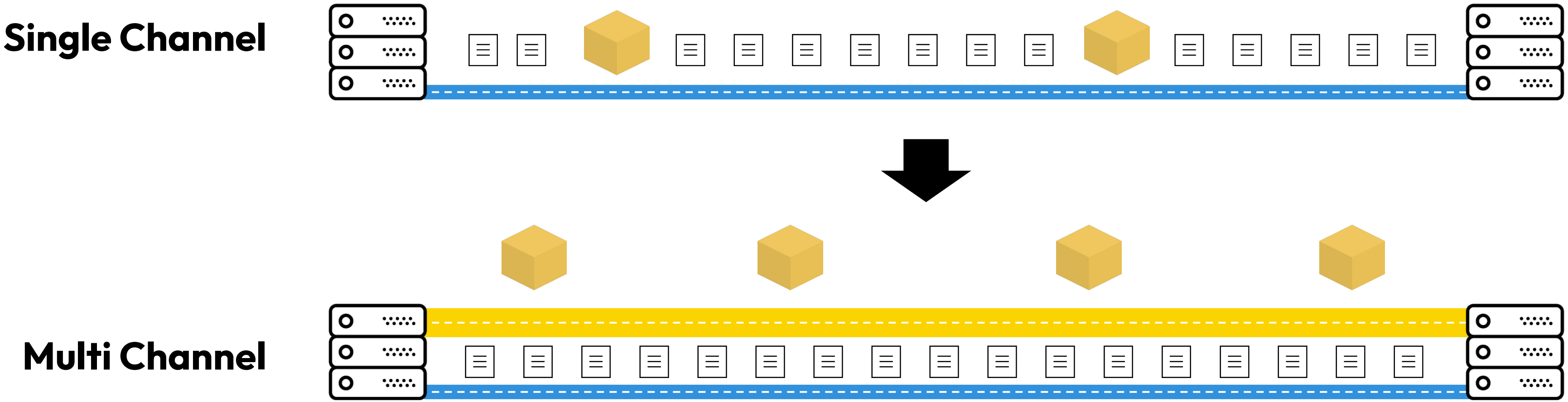


klaytn

Source: *Klaytn's Comparison of Blockchain Network Latencies*

Block Propagation - Single Channel vs. Multi Channel

To **ensure on-time latency** even though transaction congestions,
Klaytn network provides a **separate channel** for block dissemination



Block Propagation - Evaluation

Evaluate performance of using multi-channel on block propagation

Experimental setup

- Network
  - CNs: 7 x AWS EC2 c5.4xlarge
  - PNs: 4 x AWS EC2 c4.2xlarge
  - ENs: 10 x AWS EC2 c4.2xlarge
- Test scenario
  - 10 users send KLAY transfer transactions to EN as many as possible

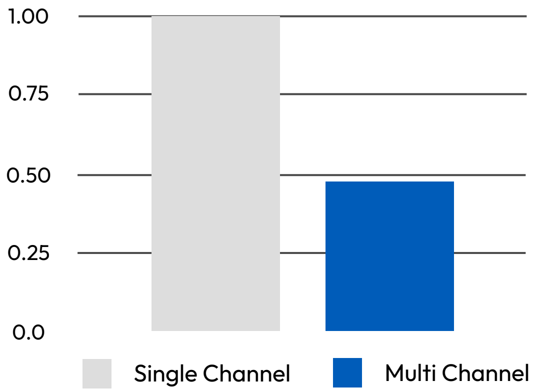
Performance metrics

- **Std. dev. of block propagation latency**
  - Measure the time when EN receives a new block
- **Transaction-to-receipt time**
  - Measure the time from a transaction is broadcasted to the receipt is available

Block Propagation - Evaluation

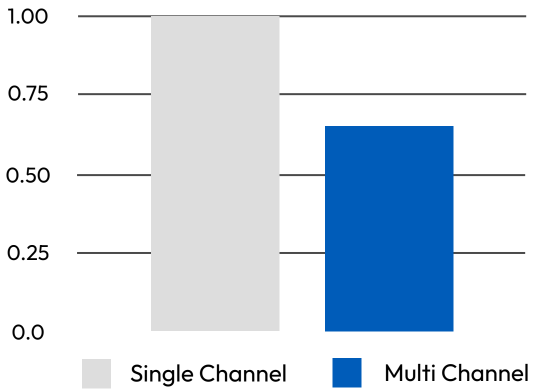
Evaluate performance of using multi-channel on block propagation

Normalized Std.Dev. of Block Transfer Latency



52% improvement

Normalized Transaction-to-Receipt Time



32% improvement

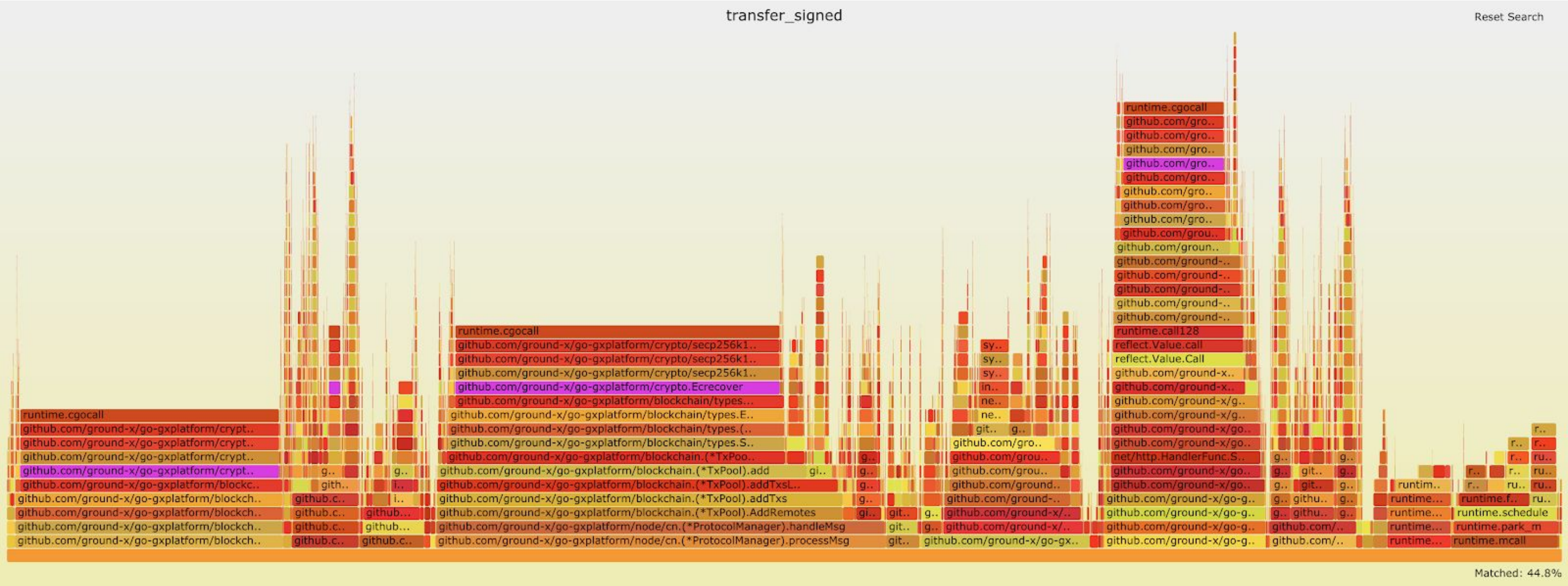
How to Increase TPS (Transactions per Second)?

Key is **PARALLELIZATION**!

Parallelizing Compute-Intensive Tasks	**Isolating Network Resources**
Limiting Concurrency with a Worker Pool	**Utilizing Fine-Grained Locking**

Analyze Performance Bottlenecks in Block Creation/Verification

Find most time-consuming parts via execution time profiling

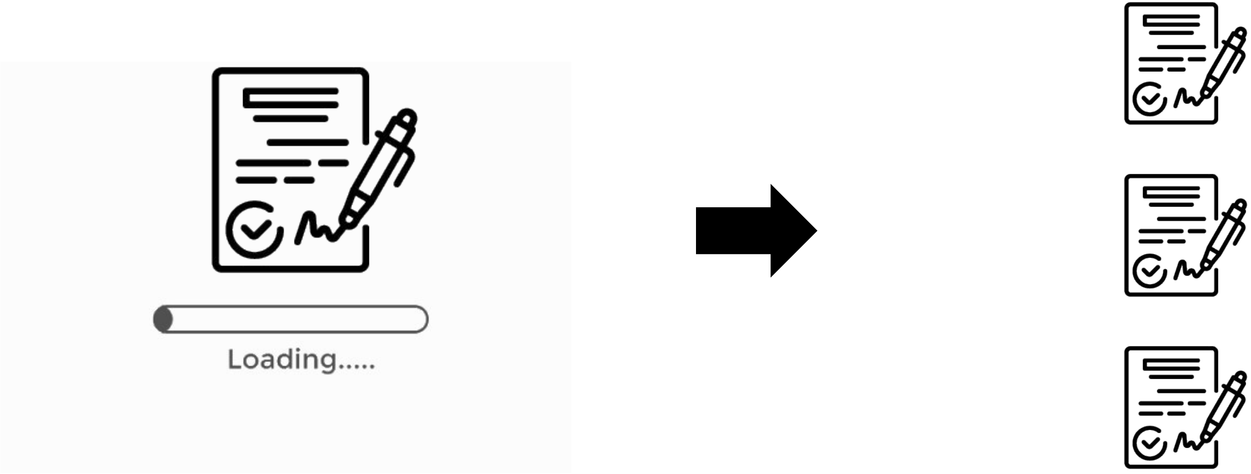


crypto.Ecrecover takes the largest portion in the execution time on KLAY transfer transaction.

Parallel Signature Verifications

Sequential Signature Verifications

Parallel Signature Verifications



Parallel Execution of Transactions

Parallelize independent ECDSA recovery computations

Situation

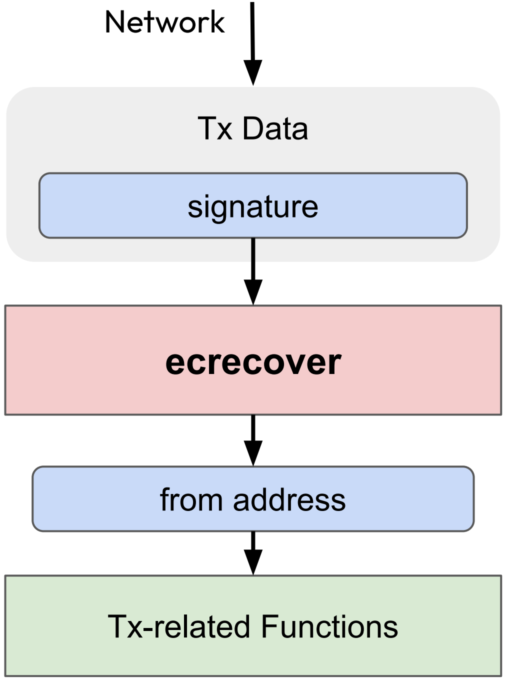
- Transaction data contains sender's signature
- Sender's address has to be derived from the signature using ECDSA recovery function
  - To invoke transaction-related functions

Problems

- Calculating the address from a signature is compute-intensive
- Need to calculate all addresses from all transactions

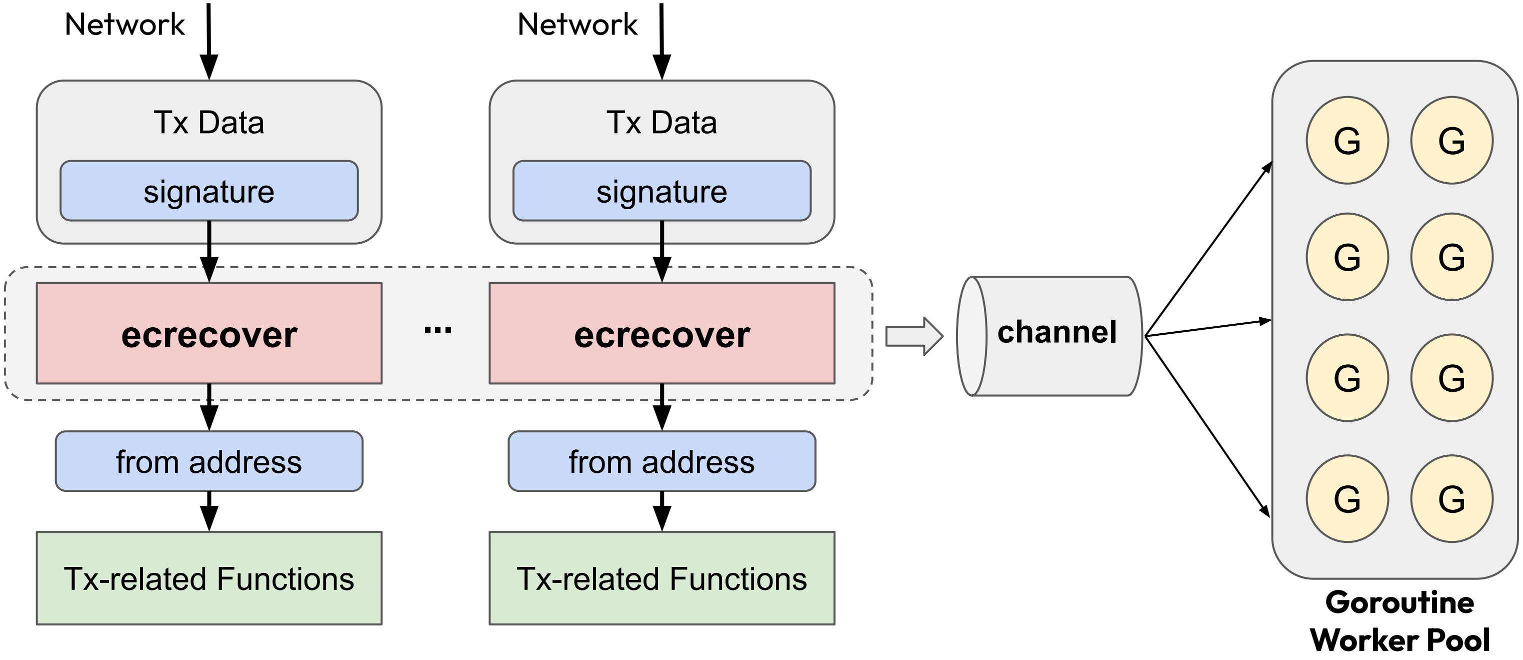
Parallelization

- Create a goroutine worker pool that executes ECDSA recovery functions
- Request executing the ECDSA recovery function for all transactions to the worker pool



Parallel Execution of Transactions

Parallelize independent ECDSA recovery computations



Parallel Execution of Transactions - Evaluation

Experiment on multicore machines by varying # of cores

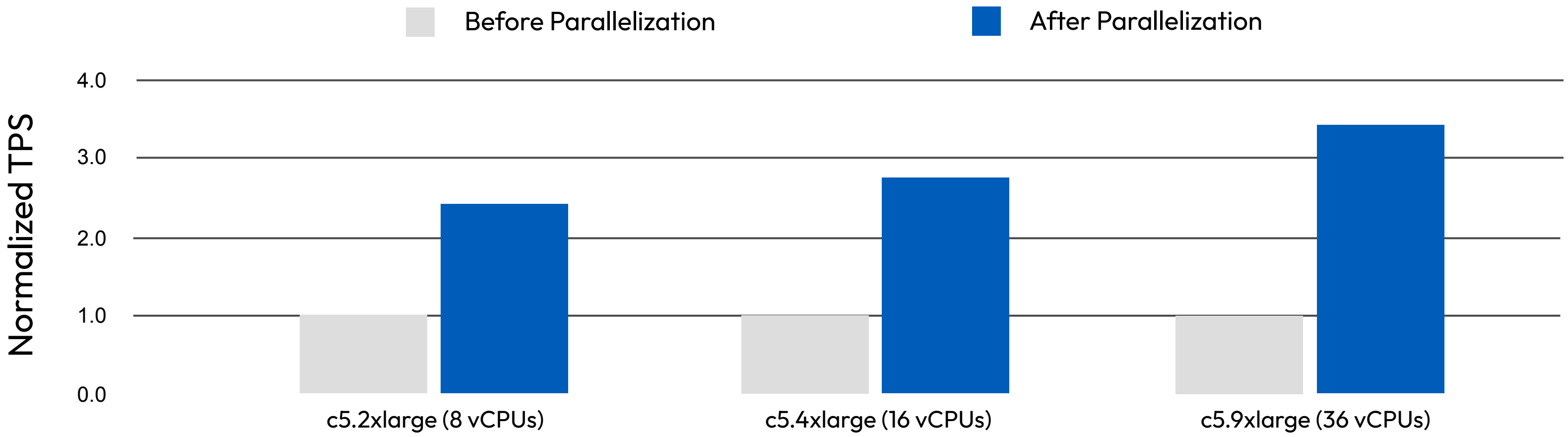
Experimental setup

- Hardware
  - AWS EC2 c5.2xlarge (Intel Xeon Platinum 8124M CPU @ 3.00GHz, 8 vCPUs)
  - AWS EC2 c5.4xlarge (Intel Xeon Platinum 8124M CPU @ 3.00GHz, 16 vCPUs)
  - AWS EC2 c5.9xlarge (Intel Xeon Platinum 8124M CPU @ 3.00GHz, 36 vCPUs)
- Test scenario
  - 20 users send KLAY transfer transactions to Klaytn nodes as many as possible

Parallel Execution of Transactions - Evaluation

Experiment on multicore machines by varying # of cores

Normalized TPS of Value Transfer Transactions



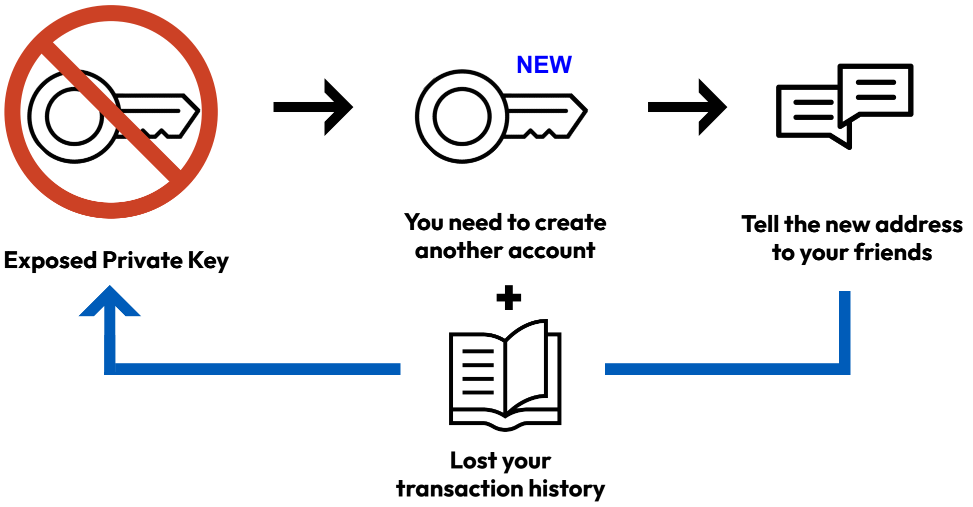
Improvement	**2.5x**	**2.7x**	**3.3x**

Challenges on UX & DX

Problem: Exposed Private Key

What if your private key is exposed?

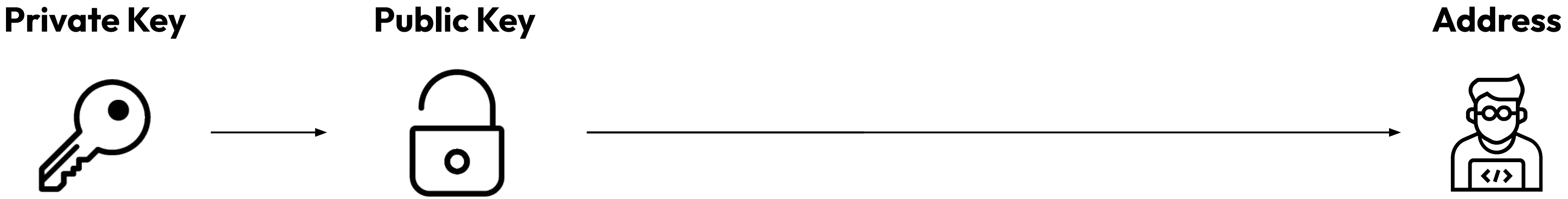
Address = Bank account number | Private key = Password


NEW
Exposed Private Key
You need to create another account
Tell the new address to your friends
+
Lost your transaction history

Solution: make private key changeable

Klaytn Account Model: Decouple Private Key and Address

Klaytn **satisfies user's real-life digital ID needs**,
natively supporting multiple keys and flexible key management



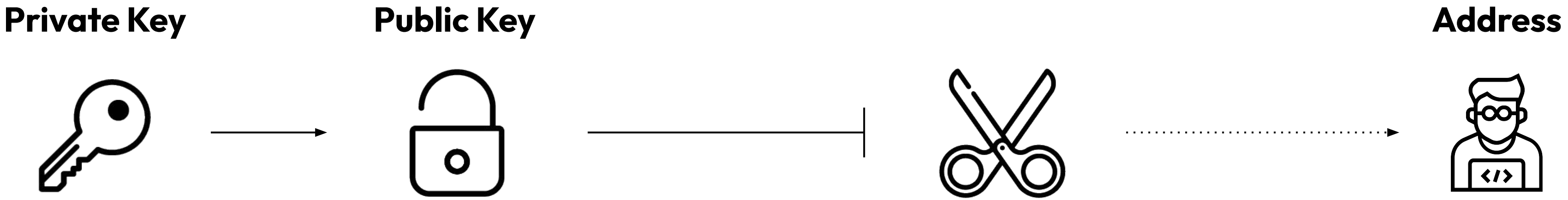
Bank Account = **Address**

Password = **Private Key**

0xA29a0AEBb4cC53794569
9A4Ef712b83981141a79

Klaytn Account Model: Decouple Private Key and Address

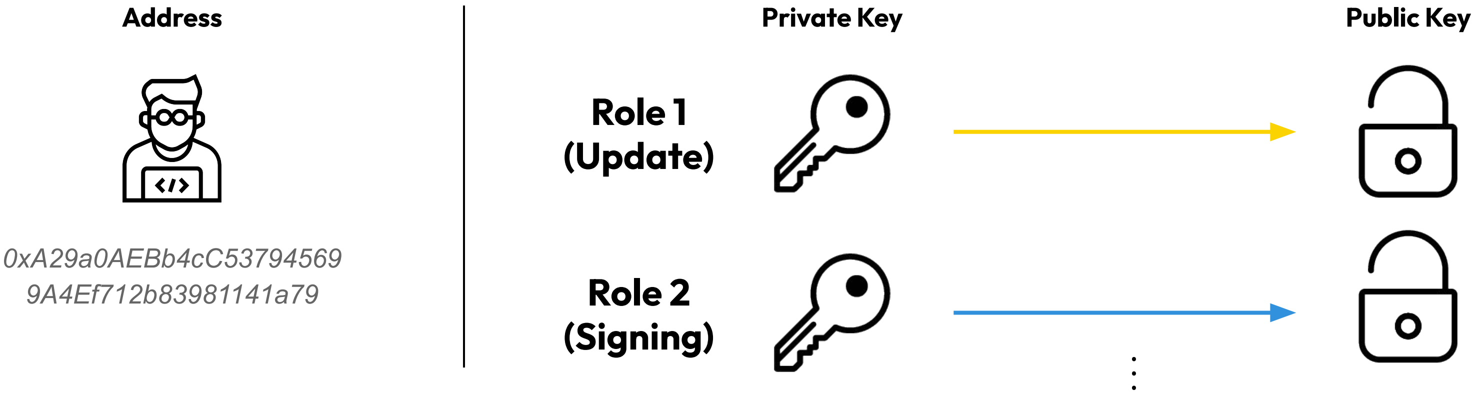
Klaytn **satisfies user's real-life digital ID needs**,
natively supporting multiple keys and flexible key management



*0xA29a0AEBb4cC53794569
9A4Ef712b83981141a79*

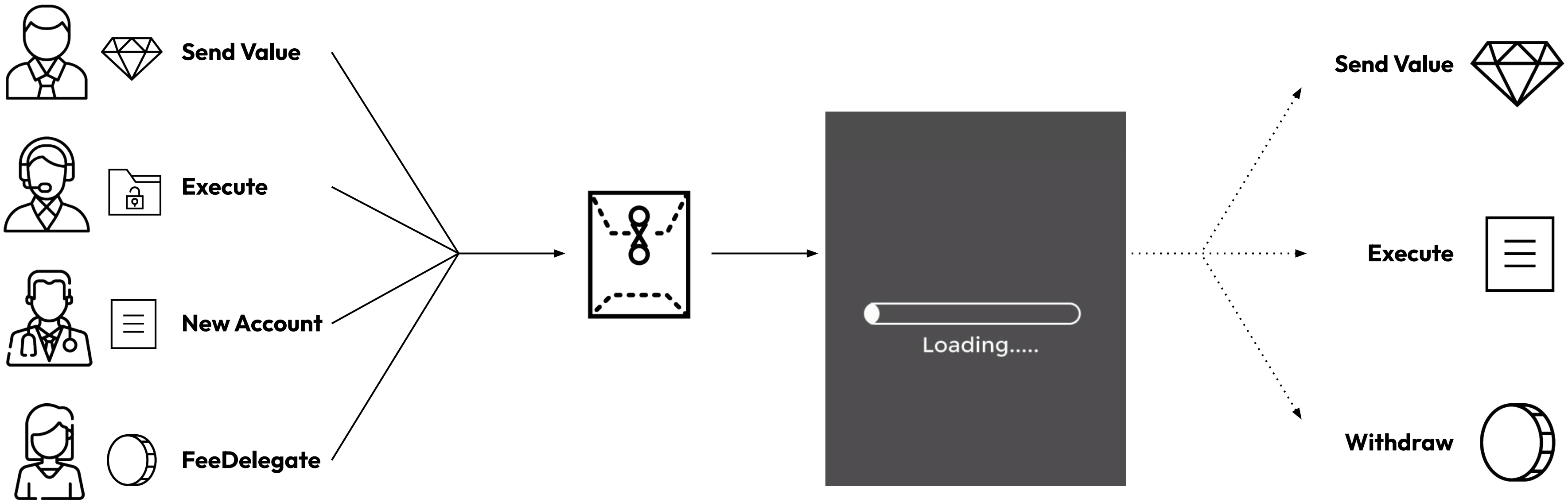
Klaytn Account Model for Better Security

Klaytn's account with role-based keys improves security.



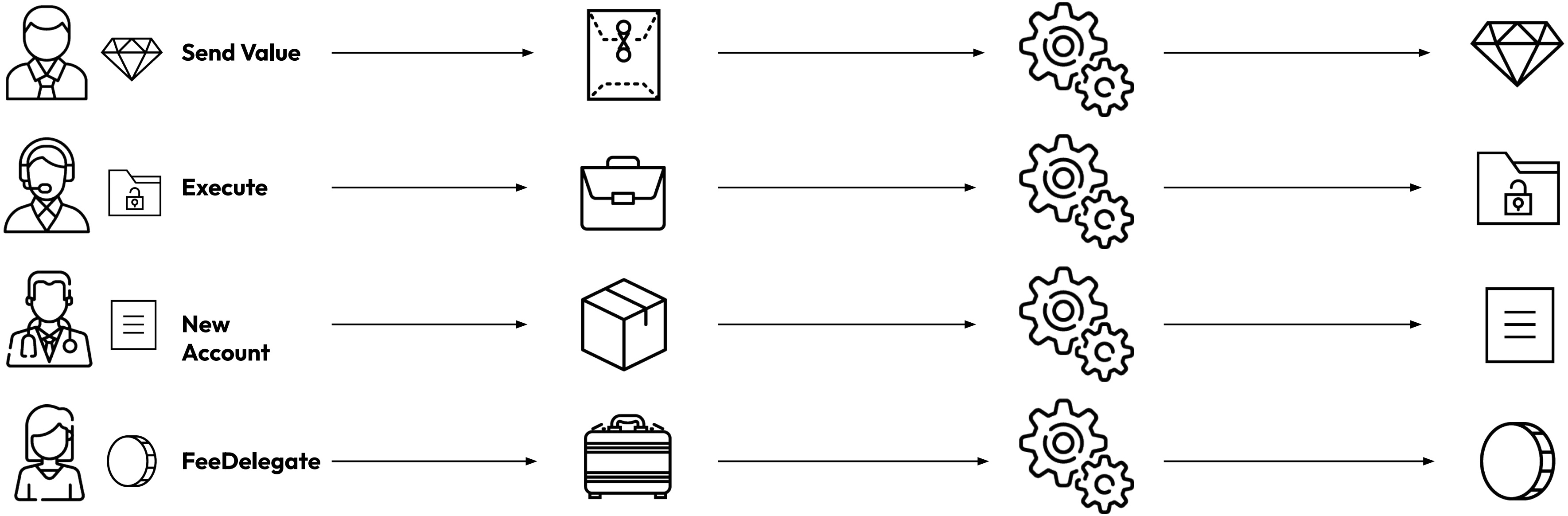
Traditional Transaction

When handling traditional transactions,
one size does NOT fit all - uniform handling leads to big waste of time

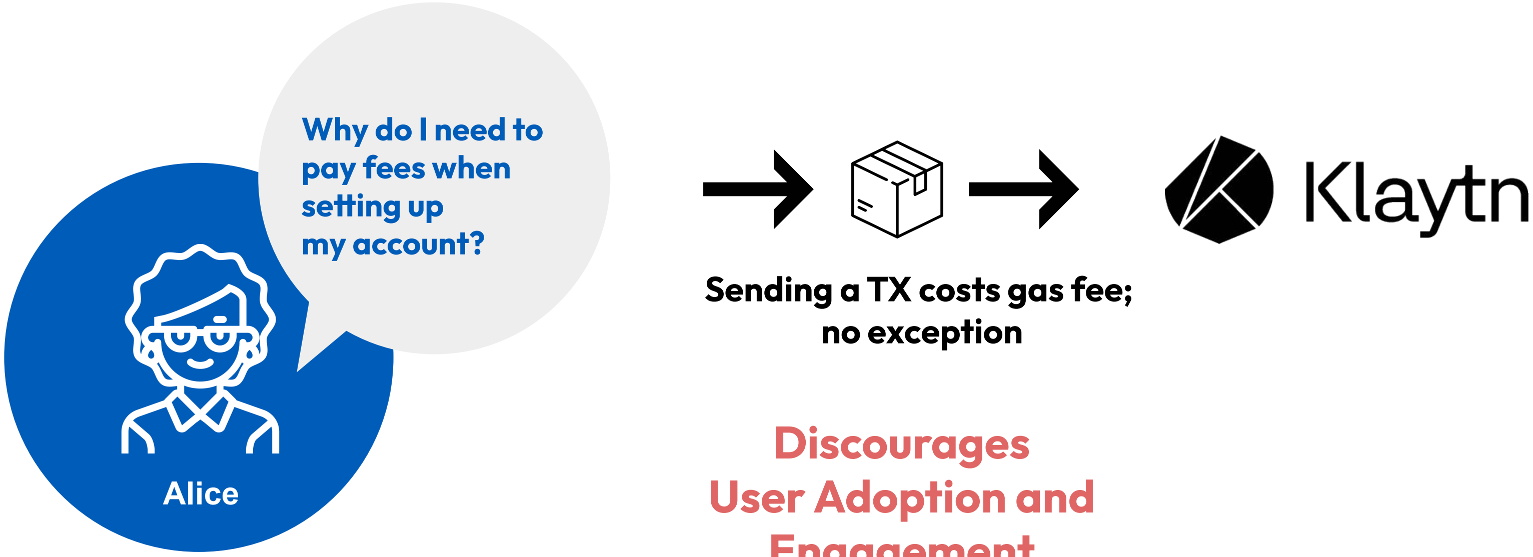


Klaytn Transaction Model

Klaytn has efficiently yet securely **expanded** its **native transaction types**, providing better developer experience and business logic opportunities



User Adoption Obstacle - Gas Fee



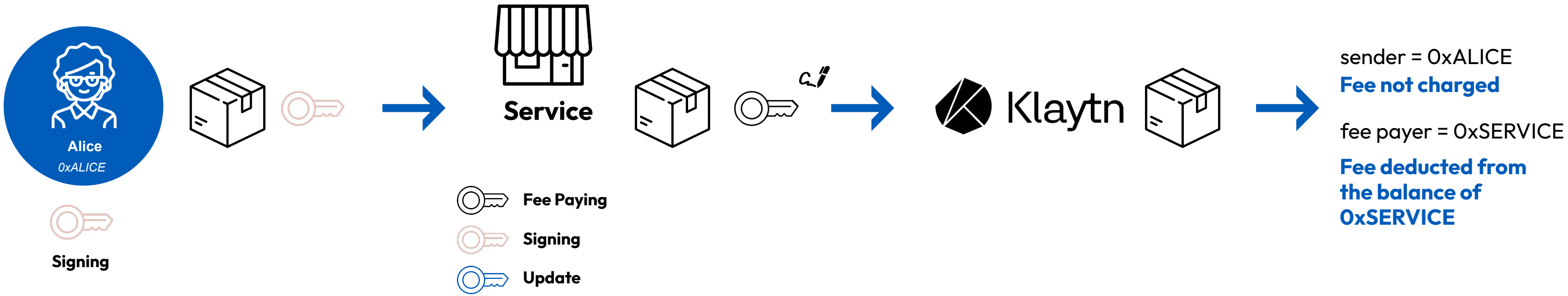
Why do I need to pay fees when setting up my account?

Alice

Sending a TX costs gas fee; no exception

Discourages User Adoption and Engagement

Fee Delegation via Fee Delegated Transaction and Fee Paying Role Key



sender = 0xALICE
Fee not charged

fee payer = 0xSERVICE
Fee deducted from the balance of 0xSERVICE

- Klaytn allows anyone to pay gas fees on other users’ behalf
- Fee delegation is done by adding additional signature to sender’s TX with fee delegated transaction types

DX: Can Developers Use Familiar Tools on Klaytn?

klaytn

klaytn scope

klaytn finder

×

METAMASK

TRUFFLE

REMIX

Hardhat

OpenZeppelin

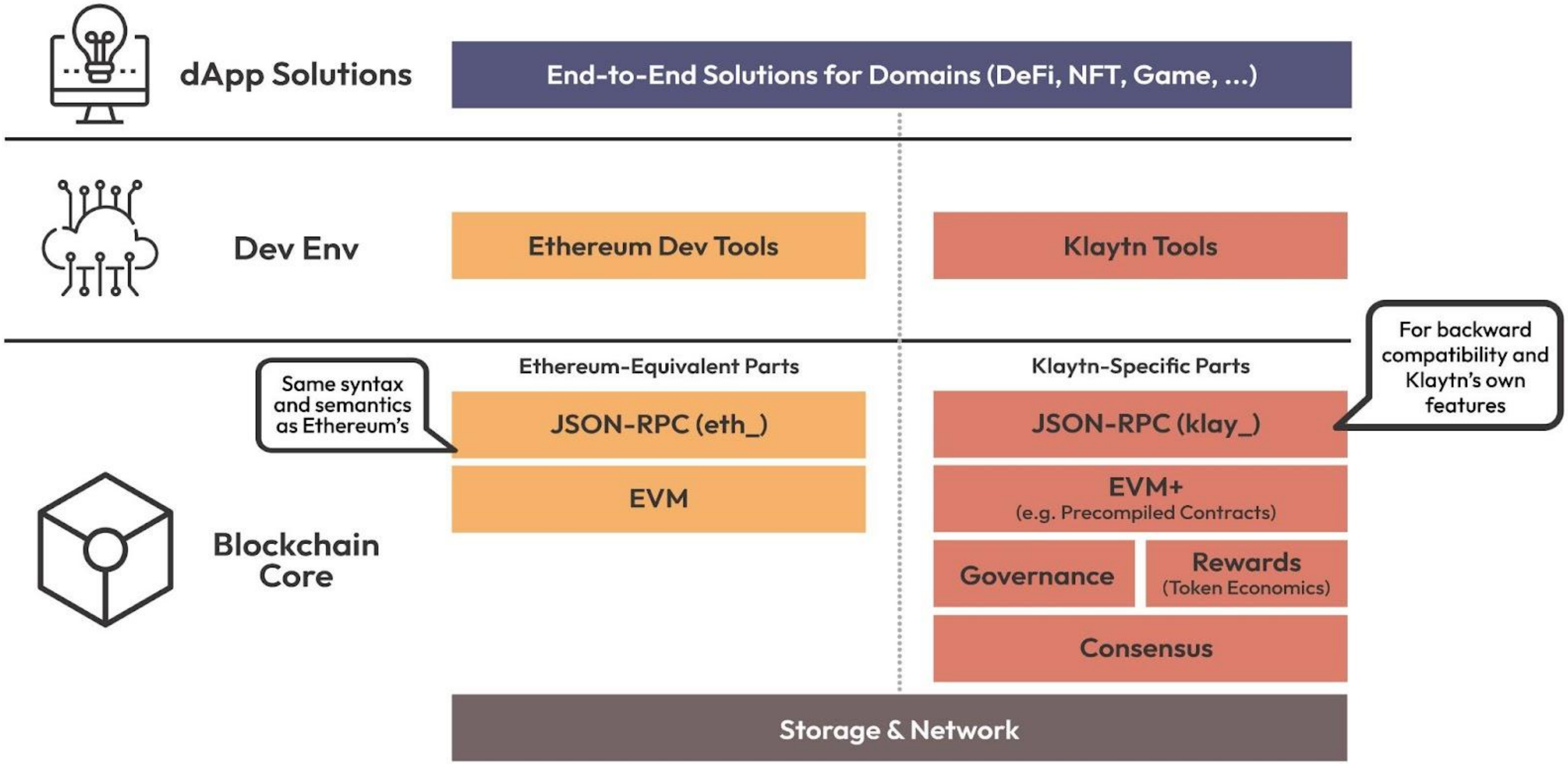
ethers

Covalent

TATUM

And more ...

Supporting Ethereum Equivalence on Klaytn


dApp Solutions
End-to-End Solutions for Domains (DeFi, NFT, Game, ...)
Dev Env
Ethereum Dev Tools
Klaytn Tools
For backward compatibility and Klaytn's own features
Ethereum-Equivalent Parts
Klaytn-Specific Parts
Same syntax and semantics as Ethereum's
JSON-RPC (eth_)
JSON-RPC (klay_)
EVM
EVM+
(e.g. Precompiled Contracts)
Blockchain Core
Governance
Rewards
(Token Economics)
Consensus
Storage & Network

Providing an Open-source Development Suite

Enjoy seamless and simplified building with Klaytn's Metaverse Package

The right tools to get started

- Customized L2 solution
- SDKs and smart contract libraries
- Wallets and chain explorers
- Distributed storage solutions
- Oracle support
- Bridges
- Dedicated package manager

Extensive ecosystem to enhance your solution

- Integration/abstraction services
- Stablecoin integrations
- DAOs
- NFT marketplaces
- DEXes and DeFi
- Traditional finance interfaces

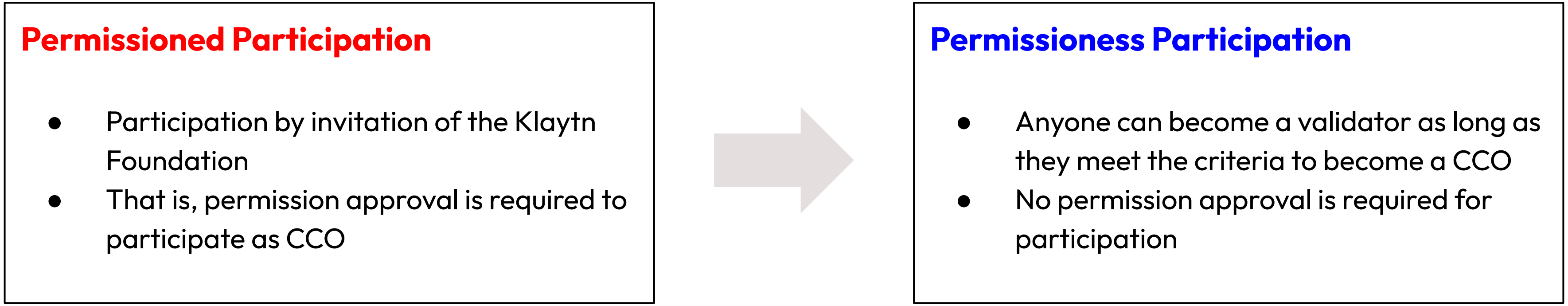
Find out more about the Metaverse Package at *klaytn.foundation*

Challenges on Decentralization

How to Make Validators Permissionless in BFT?

Change direction in how to participate as a validator (a.k.a, Core Cell Operator or CCO) in Klaytn

→ **For more nodes to freely participate as validators**



Permissioned Participation

- Participation by invitation of the Klaytn Foundation
- That is, permission approval is required to participate as CCO

Permissioness Participation

- Anyone can become a validator as long as they meet the criteria to become a CCO
- No permission approval is required for participation

Technical Challenges for Permissionless Validators

Validator Candidate Selection Mechanism

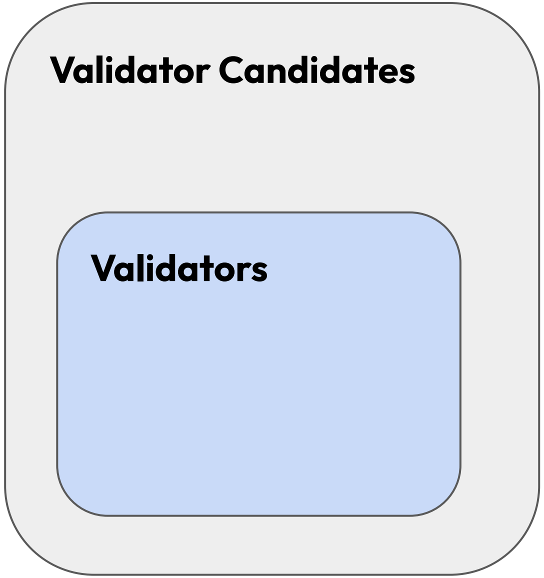
- The number of validator candidates
- Criteria for validator candidates
- Selection cycle of validator candidates

Validator Selection Mechanism

- The number of validators
  - Depending on the scalability of consensus algorithm
- Criteria for validators
- Selection cycles of validators

Network Security & Stability

- A systematic way of entering/leaving validators while preserving the network security
  - Need to prevent DoS attack, IP spoofing, hacking, etc.
- A mechanism to replace malfunctioning or underperforming validators while keeping the network stability



Architectural Decentralization

How many physical computers is a system made up of?

How many of those computers can it tolerate breaking down at any single time?

The number of nodes participating in consensus should increase

https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274

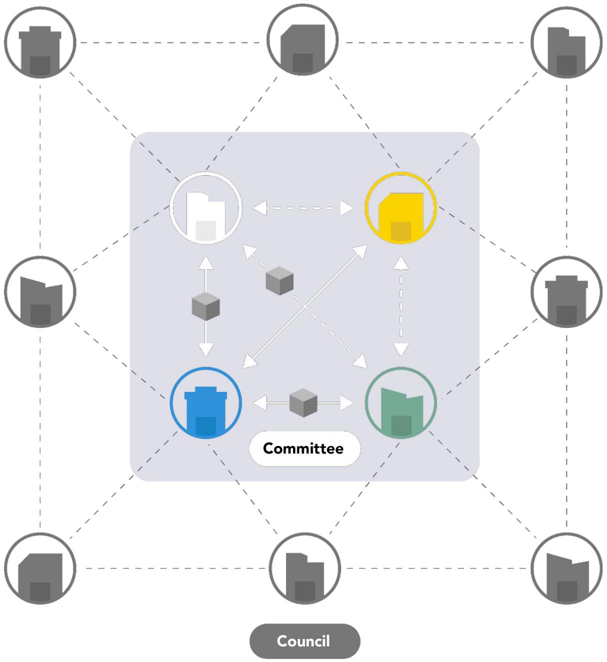
Technical Challenges for Architectural Decentralization

Message Complexity

- If there are more consensus nodes (validators), the message for consensus increases to O(N^2).

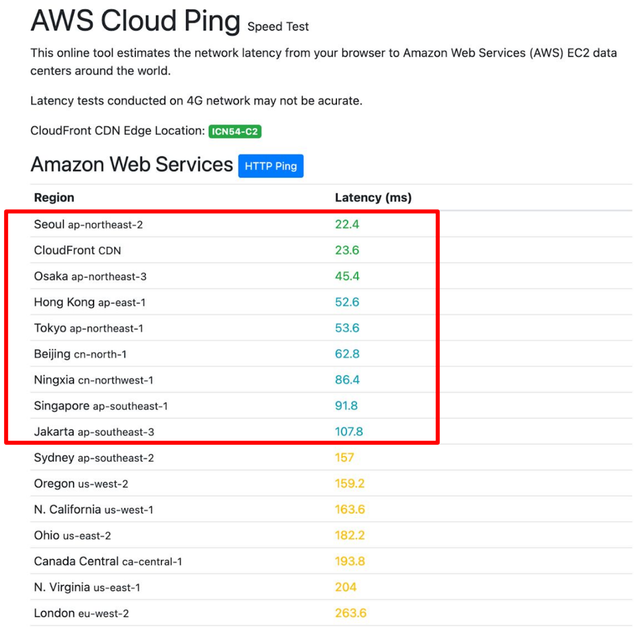
Needs a way to limit the number of messages even if the number of nodes is increased

- Optimize the consensus algorithm
  - Devise ways to reduce messages in the consensus process
- Limit the number of nodes participating in consensus
  - Reduce the consensus message by choosing only a fraction of all nodes to participate in the consensus process
  - Council vs. Committee
  - Validator Candidate Set vs. Validator Set



Technical Challenges for Architectural Decentralization

Physical network latency affects the distribution of validators



AWS Cloud Ping Speed Test

This online tool estimates the network latency from your browser to Amazon Web Services (AWS) EC2 data centers around the world.

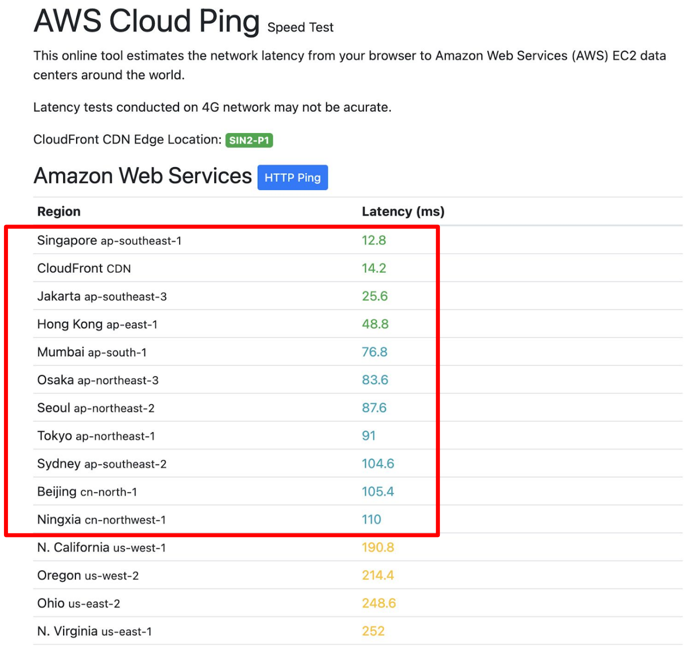
Latency tests conducted on 4G network may not be acurate.

CloudFront CDN Edge Location: ICN54-C2

Amazon Web Services HTTP Ping

Region	Latency (ms)
Seoul ap-northeast-2	22.4
CloudFront CDN	23.6
Osaka ap-northeast-3	45.4
Hong Kong ap-east-1	52.6
Tokyo ap-northeast-1	53.6
Beijing cn-north-1	62.8
Ningxia cn-northwest-1	86.4
Singapore ap-southeast-1	91.8
Jakarta ap-southeast-3	107.8
Sydney ap-southeast-2	157
Oregon us-west-2	159.2
N. California us-west-1	163.6
Ohio us-east-2	182.2
Canada Central ca-central-1	193.8
N. Virginia us-east-1	204
London eu-west-2	263.6

From Seoul



AWS Cloud Ping Speed Test

This online tool estimates the network latency from your browser to Amazon Web Services (AWS) EC2 data centers around the world.

Latency tests conducted on 4G network may not be acurate.

CloudFront CDN Edge Location: SIN2-P1

Amazon Web Services HTTP Ping

Region	Latency (ms)
Singapore ap-southeast-1	12.8
CloudFront CDN	14.2
Jakarta ap-southeast-3	25.6
Hong Kong ap-east-1	48.8
Mumbai ap-south-1	76.8
Osaka ap-northeast-3	83.6
Seoul ap-northeast-2	87.6
Tokyo ap-northeast-1	91
Sydney ap-southeast-2	104.6
Beijing cn-north-1	105.4
Ningxia cn-northwest-1	110
N. California us-west-1	190.8
Oregon us-west-2	214.4
Ohio us-east-2	248.6
N. Virginia us-east-1	252

From Singapore

https://www.cloudping.cloud/aws

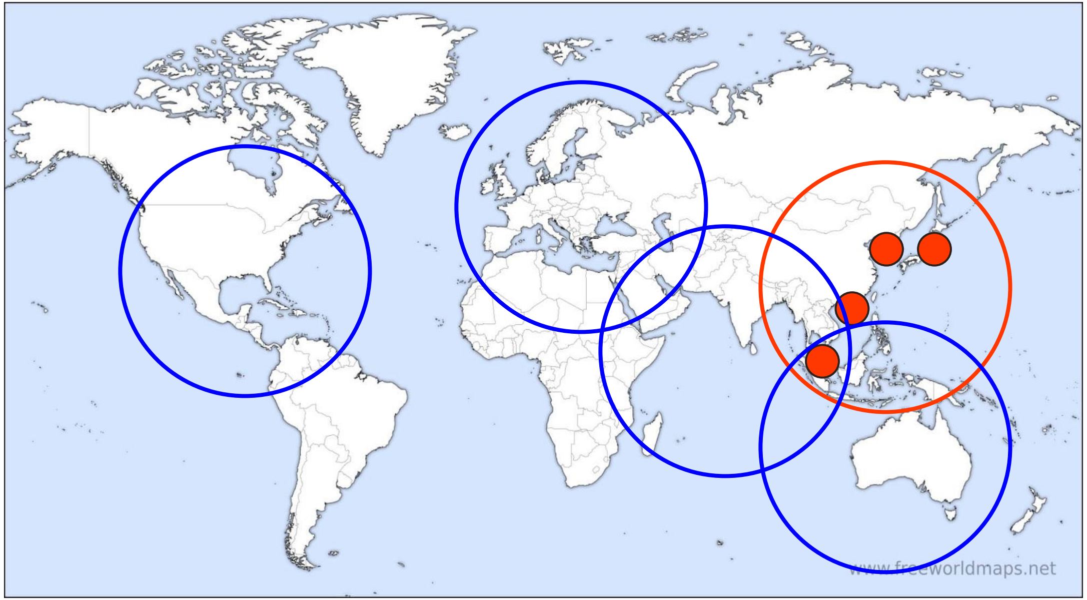
Technical Challenges for Architectural Decentralization

Resolve regional restrictions due to 1-second block time

[Present]
Validators are distributed in Asia

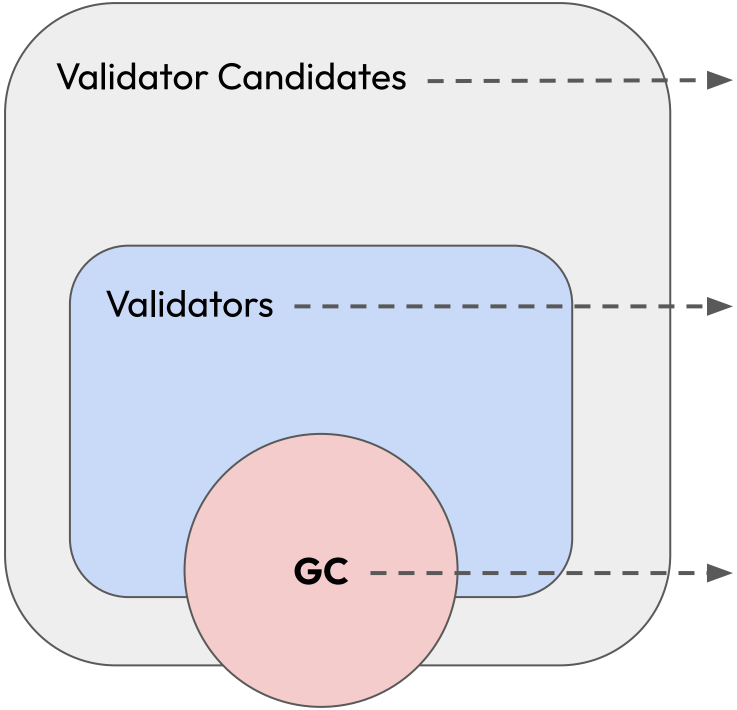
[Future]
Globally distribute validators by solving regional constraints

Need validator selection technology in consideration of regional latency



Distribution of Klaytn Validators

Governance Council with Permissionless Validators



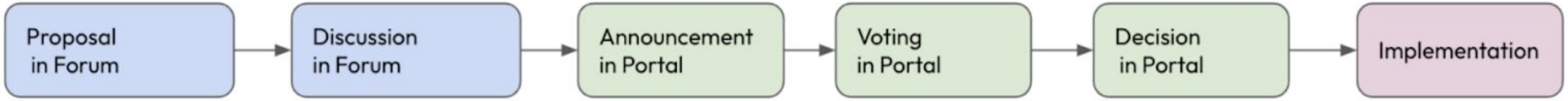
Anyone satisfying *conditions* can be a candidate by deploying a transaction

Validators are selected by certain *qualifications* every *epoch*

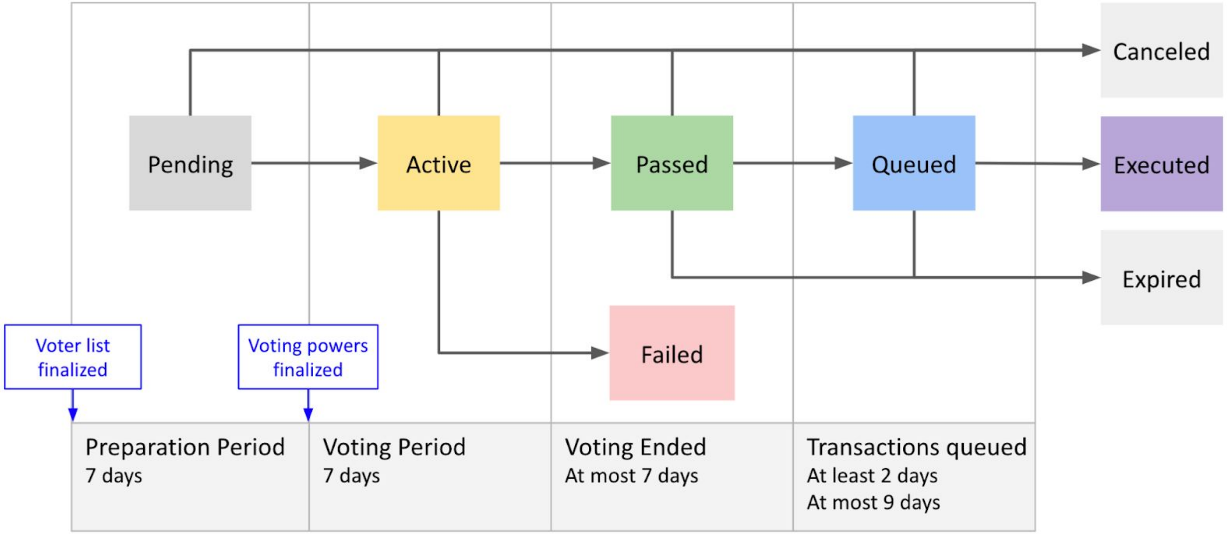
GC membership can be acquired by another *ways*

As of October 2022, GC is the same as validators

Governance On-Chain Voting Process



Proposal in Forum → Discussion in Forum → Announcement in Portal → Voting in Portal → Decision in Portal → Implementation



Pending → Active → Passed → Queued → Executed

Pending, Active, Passed, Queued → Canceled

Passed, Queued → Expired

Active → Failed

Voter list finalized

Voting powers finalized

Preparation Period
7 days

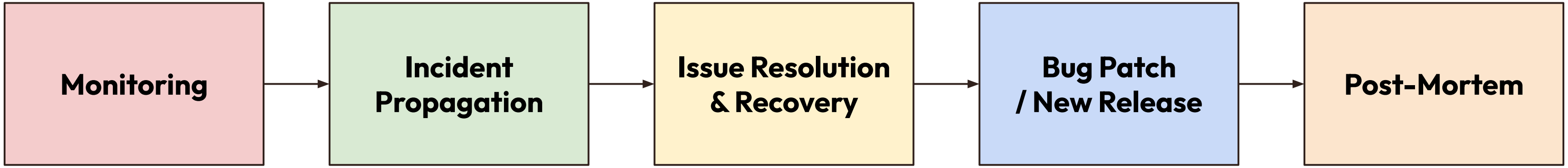
Voting Period
7 days

Voting Ended
At most 7 days

Transactions queued
At least 2 days
At most 9 days

Challenges on Network Operation

Network Monitoring & Incident Response

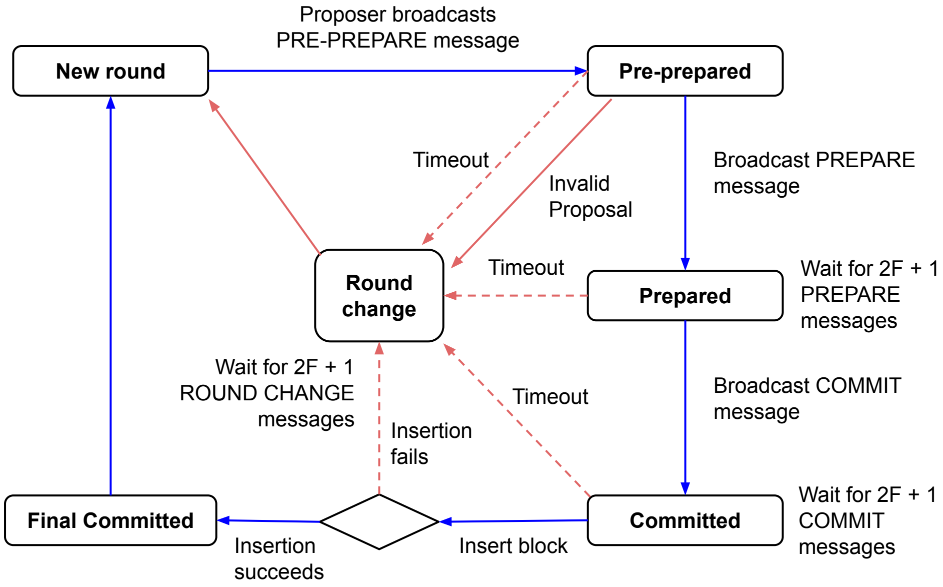


Scope of Network Operation

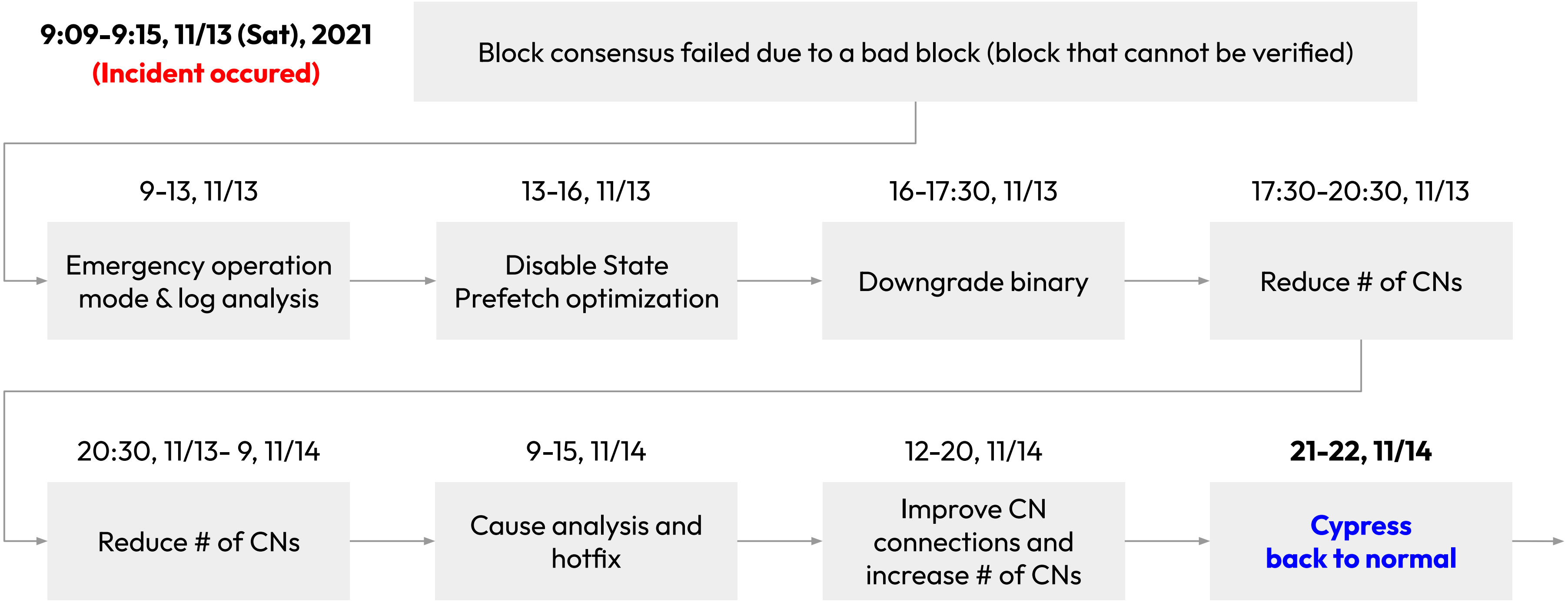
- Infrastructure (bare metal, cloud) setup & maintenance
- Automation, monitoring, dashboard, incident response, communication system, etc.

Round Change (View Change)

- Round Change occurs when a waiting operation is not completed due to some problem
  - The proposer does not propose a block OR
  - If more than ⅔ of PREPARE or COMMIT messages are not received
- In this case, the next proposer proposes a block
- It is a factor that delays the speed of the network and must be quickly restored when it occurs



Incident Example: Block Generation Failure on Klaytn Mainnet (Cypress) of Nov. 13th, 2021



9:09-9:15, 11/13 (Sat), 2021
(Incident occured)

Block consensus failed due to a bad block (block that cannot be verified)

9-13, 11/13
Emergency operation mode & log analysis

13-16, 11/13
Disable State Prefetch optimization

16-17:30, 11/13
Downgrade binary

17:30-20:30, 11/13
Reduce # of CNs

20:30, 11/13- 9, 11/14
Reduce # of CNs

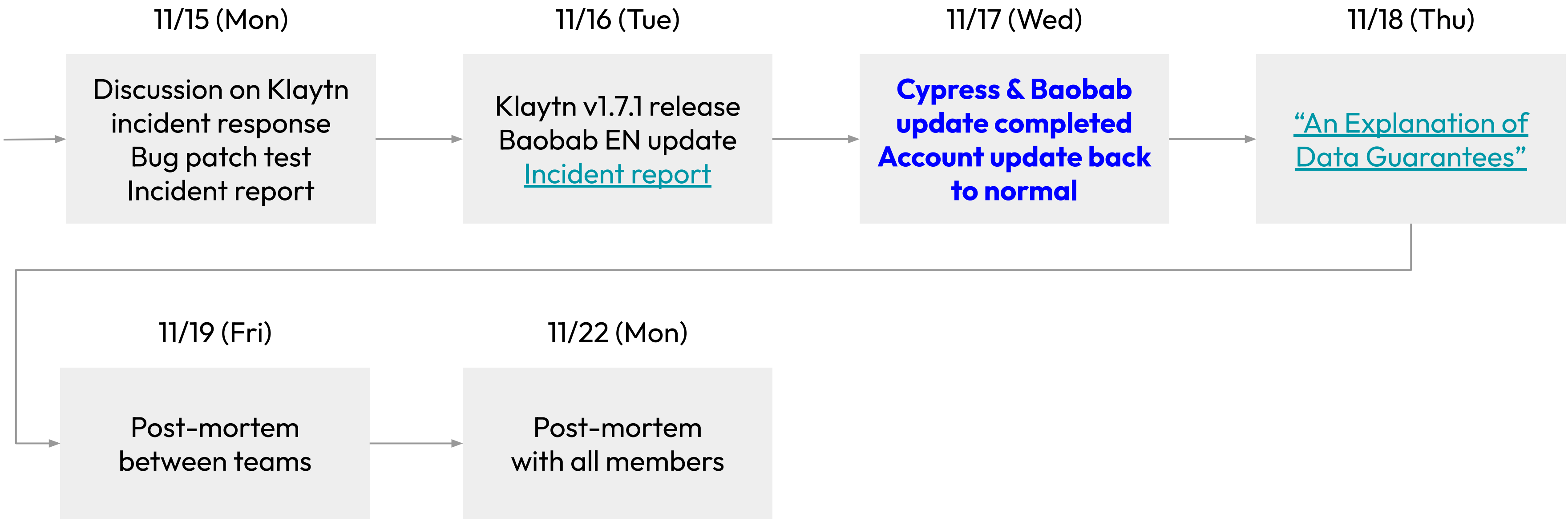
9-15, 11/14
Cause analysis and hotfix

12-20, 11/14
Improve CN connections and increase # of CNs

21-22, 11/14
Cypress back to normal

https://medium.com/p/8efb849bb8c9

Incident Example: Block Generation Failure on Klaytn Mainnet (Cypress) of Nov. 13th, 2021



11/15 (Mon)

Discussion on Klaytn incident response
Bug patch test
Incident report

11/16 (Tue)

Klaytn v1.7.1 release
Baobab EN update
Incident report

11/17 (Wed)

Cypress & Baobab update completed
Account update back to normal

11/18 (Thu)

“An Explanation of Data Guarantees”

11/19 (Fri)

Post-mortem between teams

11/22 (Mon)

Post-mortem with all members

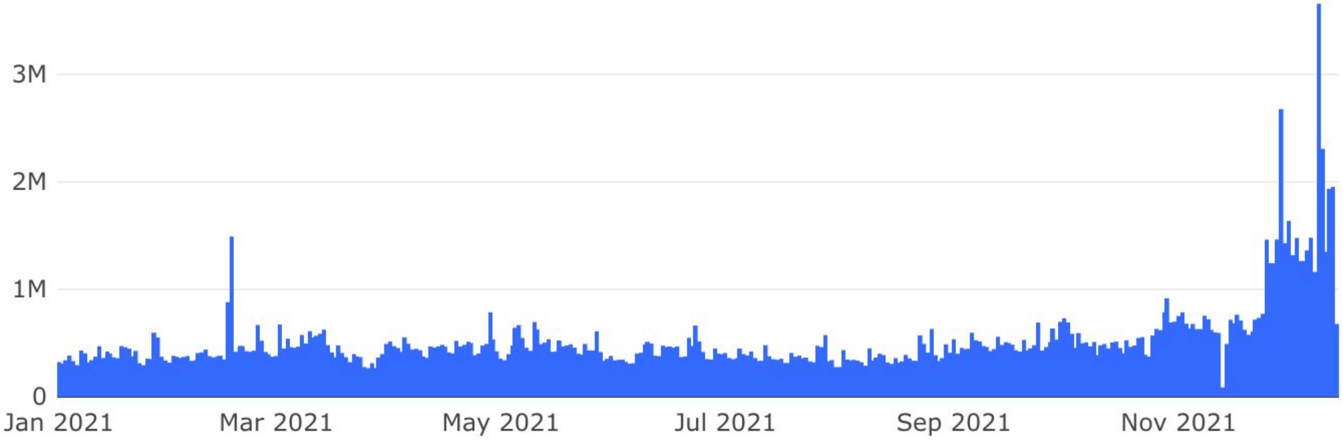
https://medium.com/p/8efb849bb8c9

Lessons Learned from Incident on Nov. 13th, 2021

1. Incidents can occur at any time, so it is necessary to quickly recognize and respond to them.
2. Incident response process should be in place and all involved should be aware of it.
3. Incident response training must be conducted regularly.
4. Klaytn's development manpower needs to be rapidly increased.
5. There must be a team dedicated to network (mainnet) operation.
6. There should be external people who understand Klaytn's code and network operations.
7. There should be a way to go into emergency recovery mode.

Arbitrage Bot Transaction Issue

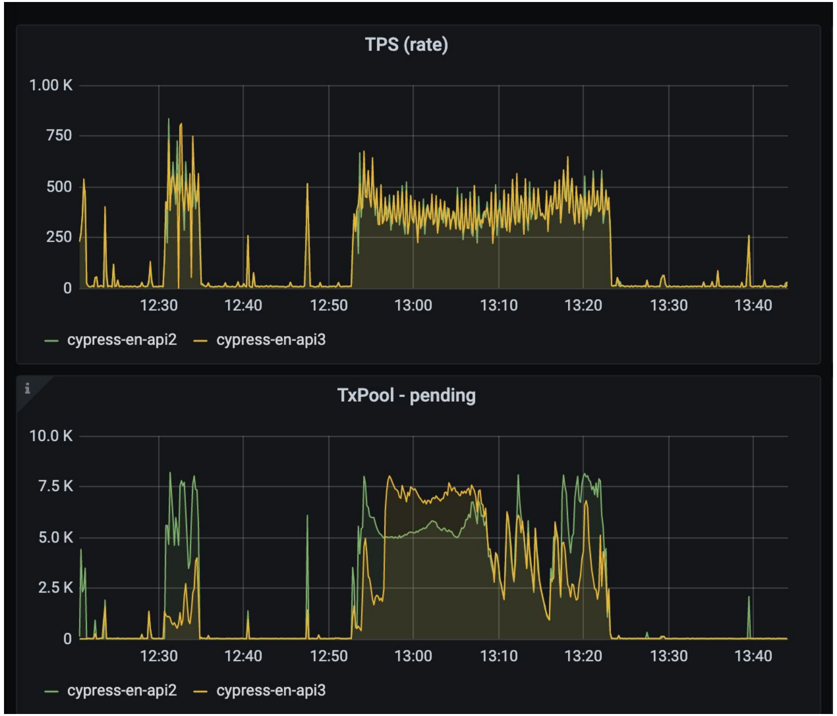
Cypress Daily Transaction: Clear trend of Klaytn network usage increase



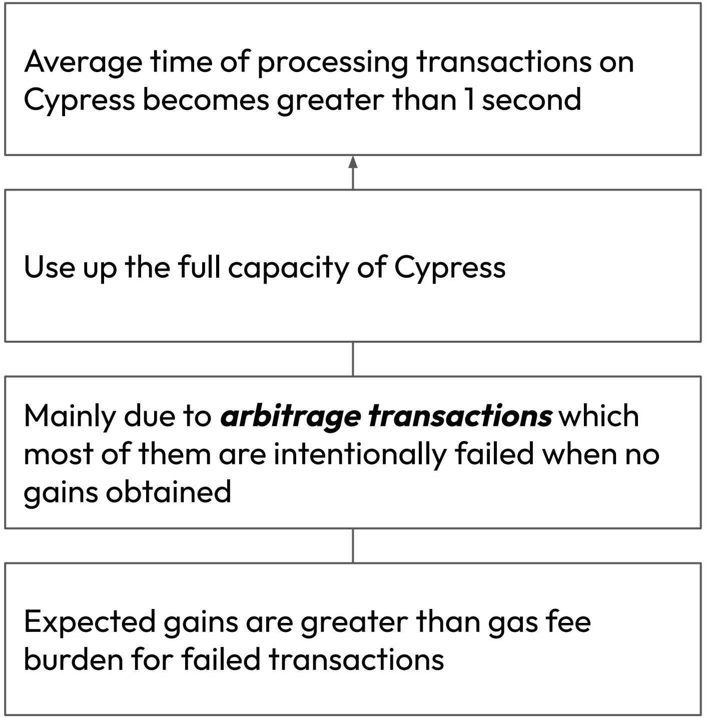
Now to assess Klaytn's gas fee policy to support fair and efficient network usage

Arbitrage Bot Transaction Issue (cont'd)

Recent Trend of Bursty Transactions (Getting worse!)



Issue and Root Cause



Average time of processing transactions on Cypress becomes greater than 1 second

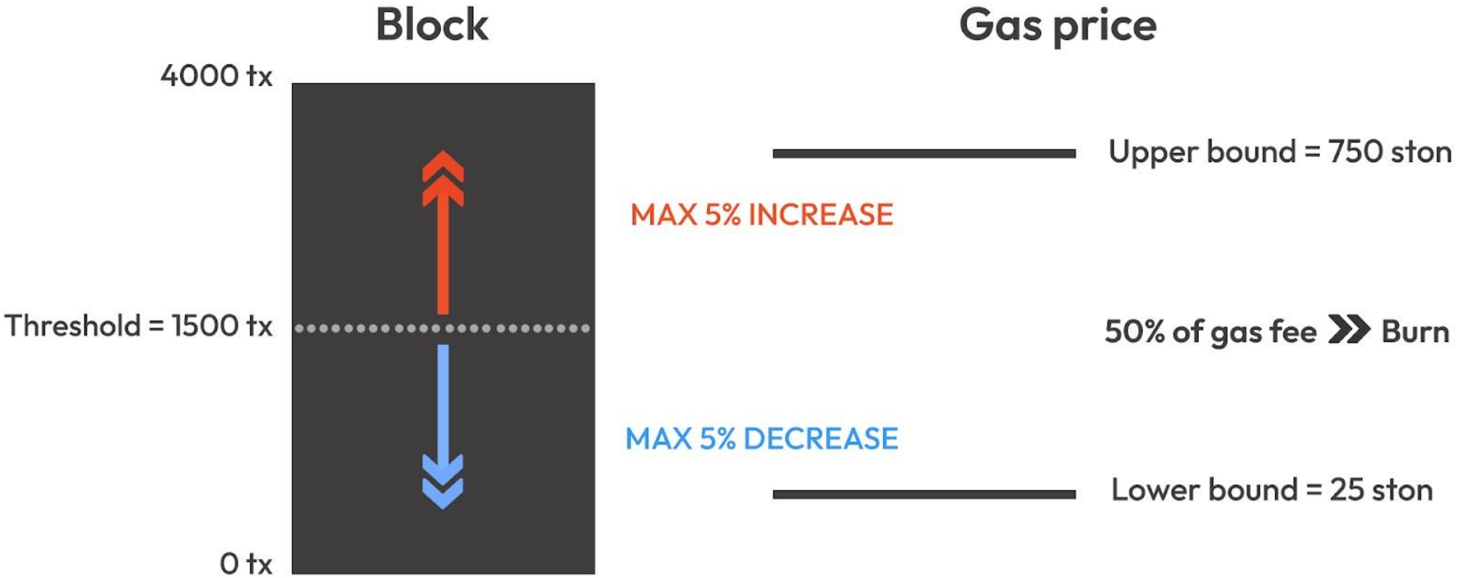
Use up the full capacity of Cypress

Mainly due to ***arbitrage transactions*** which most of them are intentionally failed when no gains obtained

Expected gains are greater than gas fee burden for failed transactions

Dynamic Gas Fee Pricing Model

Enjoy cheap gas price usually. Pay more only when the network is congested.



Sources:
Klaytn Improvement Proposal
Klaytn's Dynamic Gas Free Pricing Mechanism

klaytn

What's Next?

Lessons Learned from Mainnet Development & Operation

Good Points

- It means a lot to pioneer a new development or career path that no one else has gone before.
- It is rewarding to solve difficult problems and develop good technology that many people can use.
- It is fun to solve various factors such as values, performance, and constraints in reality all together.

Challenges

- It is technically very difficult to keep evolving a nondisruptive distributed system.
- It is difficult to find people who are interested in or suitable for mainnet development.
- Mainnet development is not so visible to the public.
- It's not very noticeable if we're good at it, but if we're not good at it, there's a lot of critics.
    - In particular, if a failure occurs in the mainnet, the aftermath could be large and a big problem.

Try the Mainnet Development

Mainnet development is a challenge for people
who are serious about technology development.

If you are a person who digs into problems, likes systems,
and enjoys helping others develop better,
you may take on the challenge of the mainnet development.

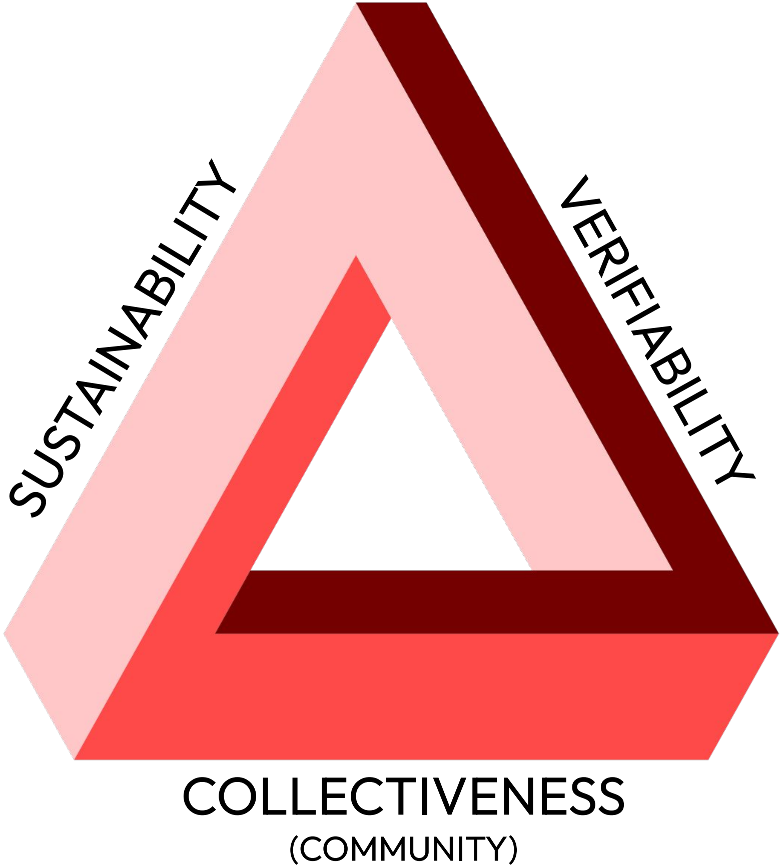
Problems that Klaytn Encountered and Have to Solve

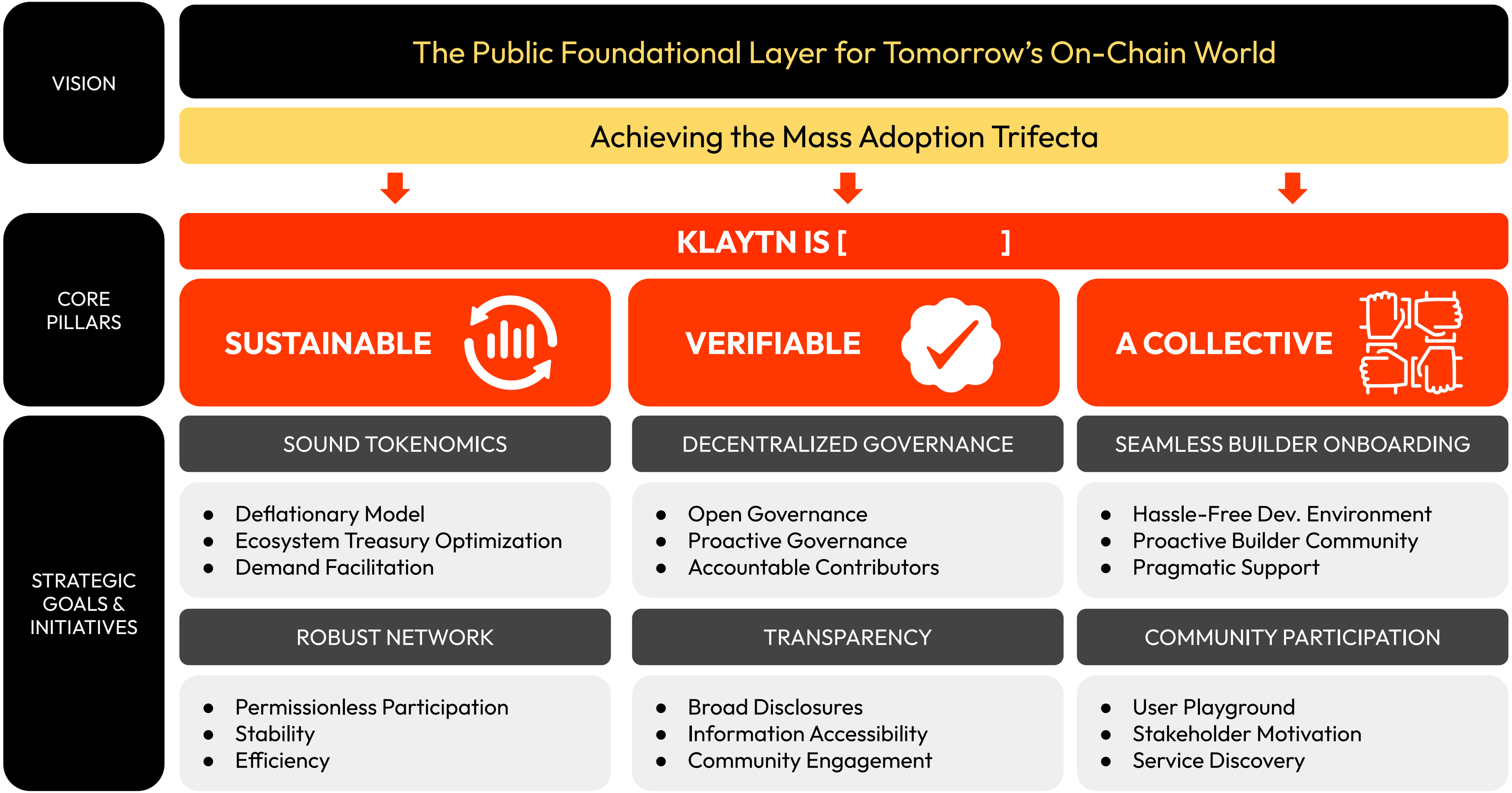
Low Market Penetration

Sustainability Issues

Trust Issues

Mass Adoption Trifecta





VISION

The Public Foundational Layer for Tomorrow's On-Chain World

Achieving the Mass Adoption Trifecta

CORE PILLARS

KLAYTN IS [ ]

SUSTAINABLE	VERIFIABLE	A COLLECTIVE

STRATEGIC GOALS & INITIATIVES

SOUND TOKENOMICS	DECENTRALIZED GOVERNANCE	SEAMLESS BUILDER ONBOARDING
● Deflationary Model ● Ecosystem Treasury Optimization ● Demand Facilitation	● Open Governance ● Proactive Governance ● Accountable Contributors	● Hassle-Free Dev. Environment ● Proactive Builder Community ● Pragmatic Support
ROBUST NETWORK	**TRANSPARENCY**	**COMMUNITY PARTICIPATION**
● Permissionless Participation ● Stability ● Efficiency	● Broad Disclosures ● Information Accessibility ● Community Engagement	● User Playground ● Stakeholder Motivation ● Service Discovery

Q&A

Thank you for your attention!