

Trading Derivatives on Hyperledger

Siriwat Kasamwattananarote, Thierry Gibraltar, Vsevolod Yugov, Shibo Lin
Hideaki Takei, Fernando Vazquez

LinuxCon + ContainerCon Japan 2016
15 July 2016

We are



We use



HYPERLEDGER PROJECT



People

- SBI BITS (Better IT Solution)
- We're building a next generation trading platforms



Fernando Vazquez



Siriwat Kasamwattananote



Thierry Gibraltar



Vsevolod Yugov



Shibo Lin



Hideaki Takei

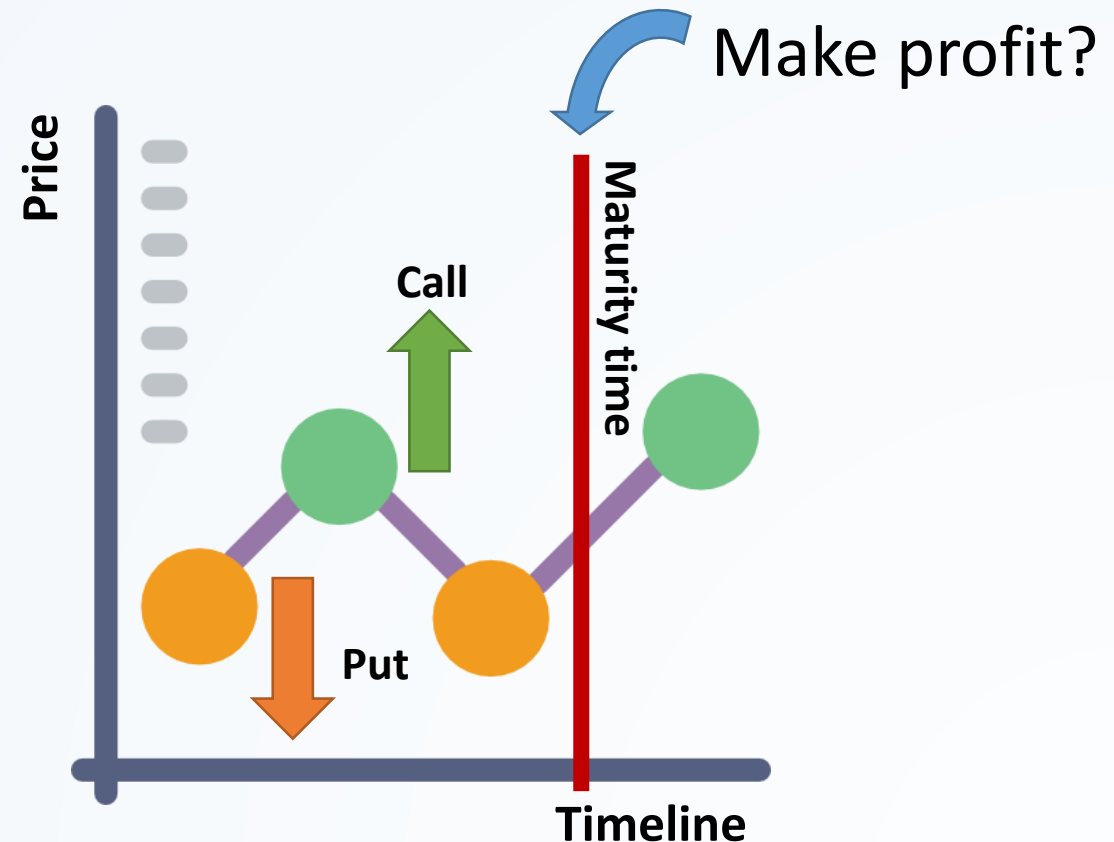
Agenda

- What are we doing?
 - Trading derivatives
 - Hyperledger with container technology.
- Our design
- Performance evaluation
- Next challenges
- Key takeaways

A little more about us

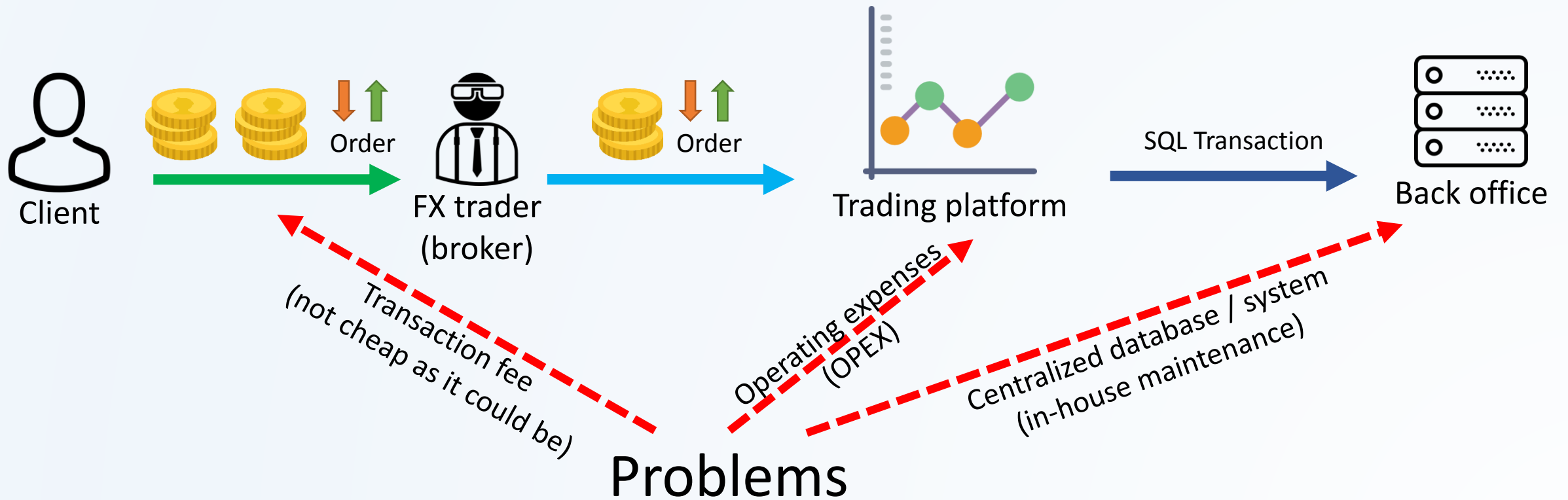
Trading Derivatives

- What are derivatives?
 - Derivatives are securities linked to the other securities.
 - E.g. options, futures, and swaps.
- In case of an option
 - **Option** created when
 - Client **sells** their **Put/Call option** and other client **buys** the available **Put/Call option**.
 - Fee is called **premium**.
 - **Transaction** happen when
 - The option is **created**.
 - Existing option is **bought/sold/transferred**.
 - Exercising
 - At the **maturity time**
 - **Premium < Payout = Profit**



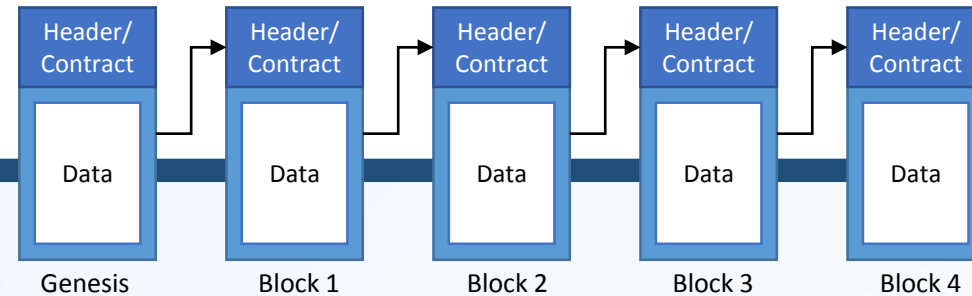
Trading Derivatives (2)

- Traditional platform



The new approach

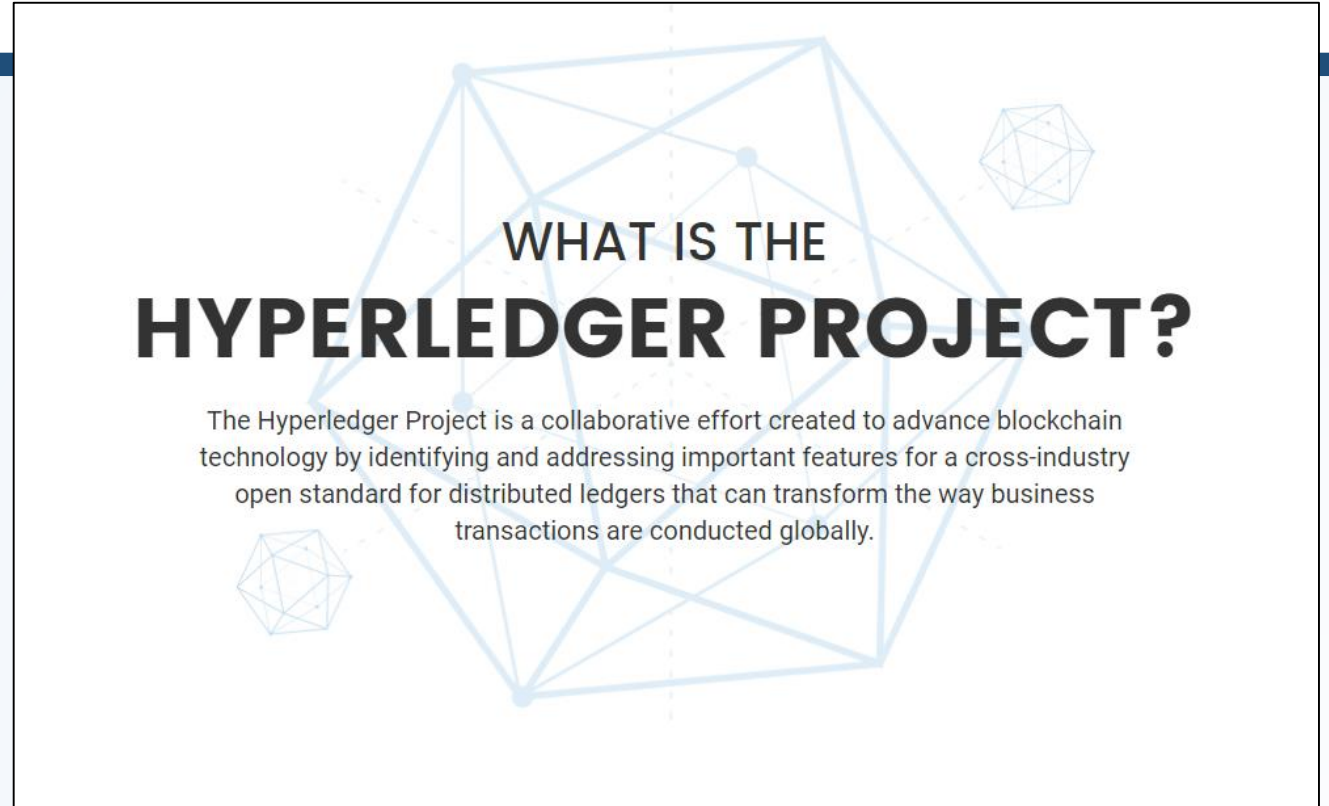
Blockchain



- A chain of blocks.
- Blockchain is a *framework*.
- A block may contain *code* as a smart contract.
- Ideal smart contract examples
 - Car *insurance activates* only when driving.
 - Money *is transferred* from A to B at 11:00 AM 15 July 2016 JST.
- The contract triggers **automatically**.
 - When a contract meets its true conditions.

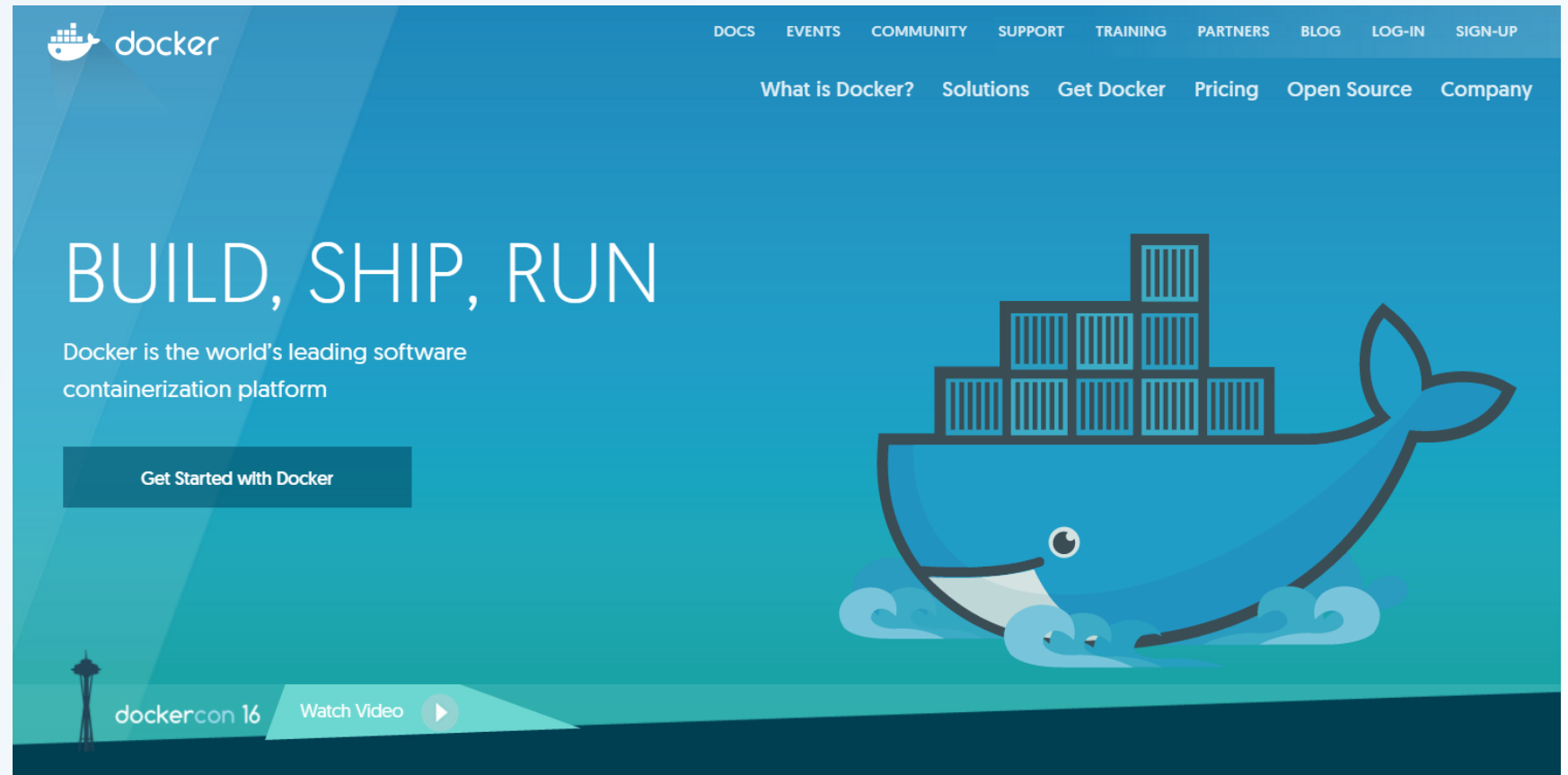
Hyperledger

- A blockchain framework.
- Open standard/source.
- Customizable
 - Smart contract
 - Namely, *chaincode*.
 - Consensus plugins
 - Data payload
- Community driven.
- Written in Go.
- Heavily depends on the container technology.



Container

- Docker is a kind of containerization platform.
- No guest OS.
- Only bin/lib/app.
- We all know :)



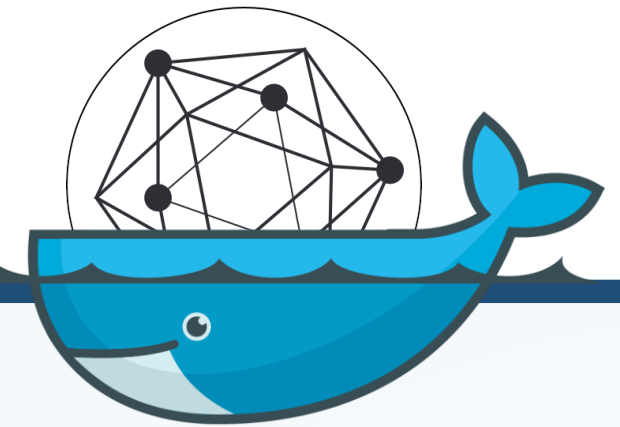


HYPERLEDGER PROJECT

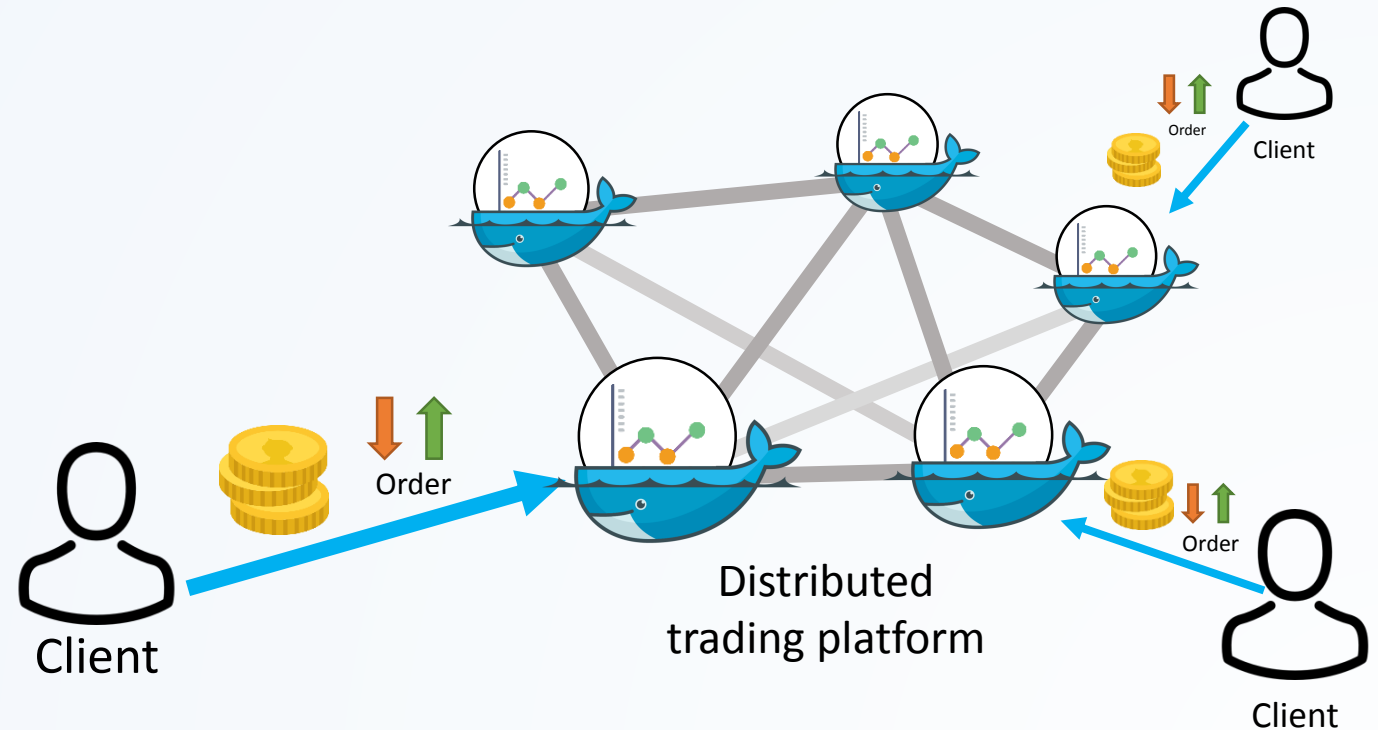
+



docker =



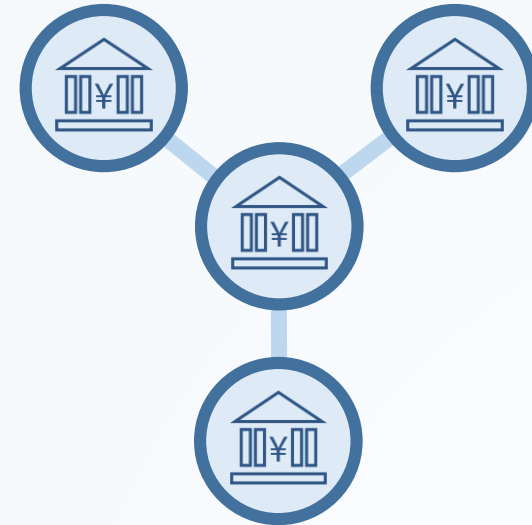
- Becomes a **fully distributed** blockchain framework.
- Motivated by
 - Our benefits
 - Simplified back office (BO).
 - Shareable BO between holder.
 - Less operating expenses (OPEX).
 - De-centralized database.
 - Contributing a common ledger.
 - Increasing transaction volume.
 - Customer benefits
 - Lower fee
- Expected benefits
 - High availability
 - Disaster recovery



Our design

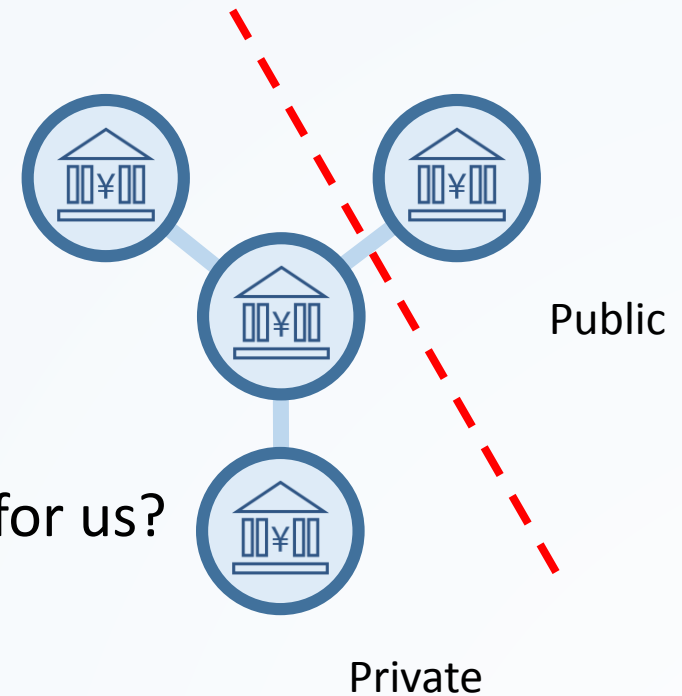
Use case

- Aims for derivative trading platform
- Trust model
 - Trusted nodes
 - Know your customer (KYC) procedure (by law)
 - Private network
 - Permissioned blockchain
- Regulatory oversight



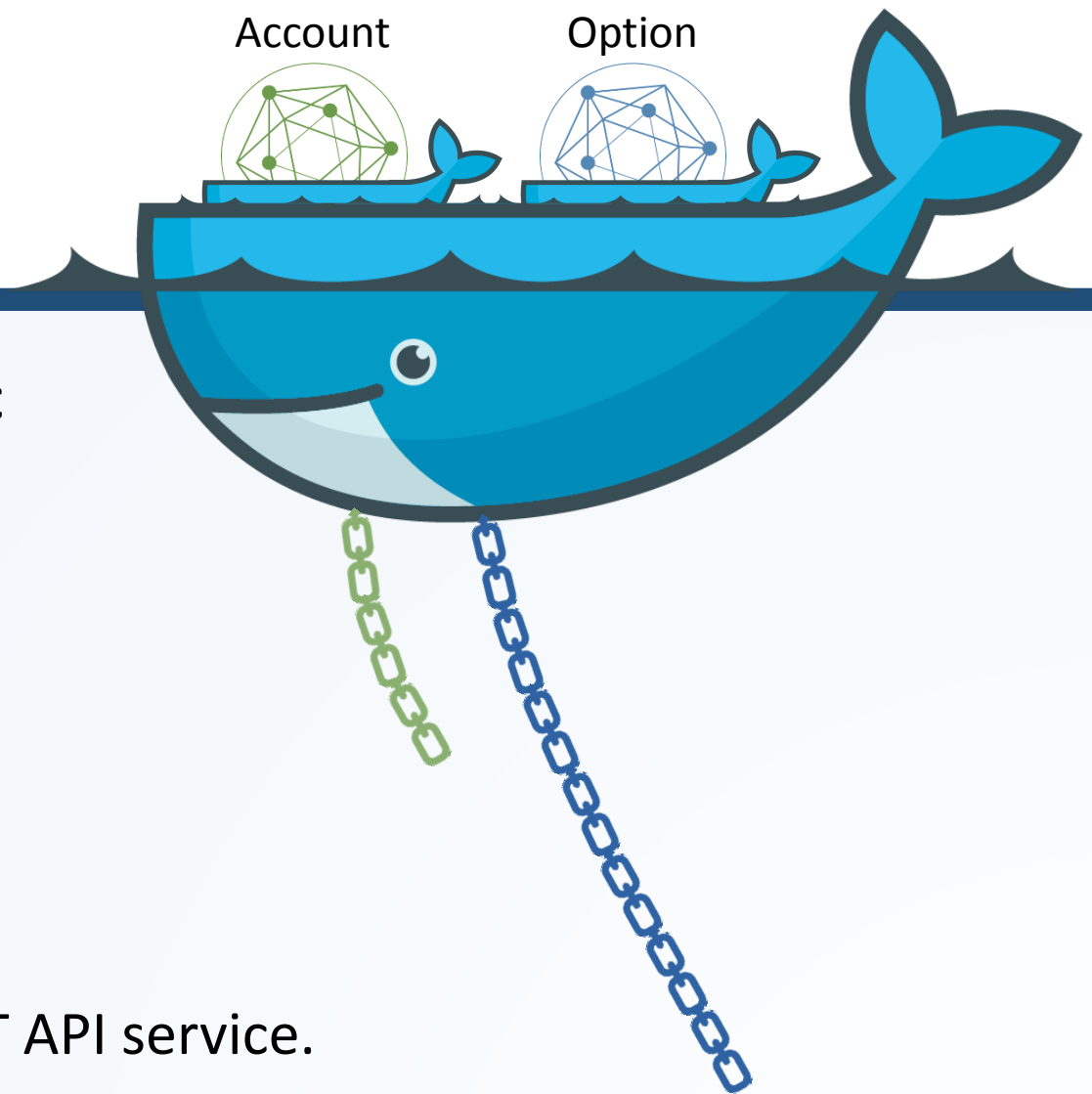
Consensus

- Hyperledger's default consensus plugins:
 - Noops
 - Trusts everyone.
 - No integrity.
 - PBFT
 - Trusts majorities
 - Partially recoverable.
 - What could be the ***optimal*** consensus algorithm for us?



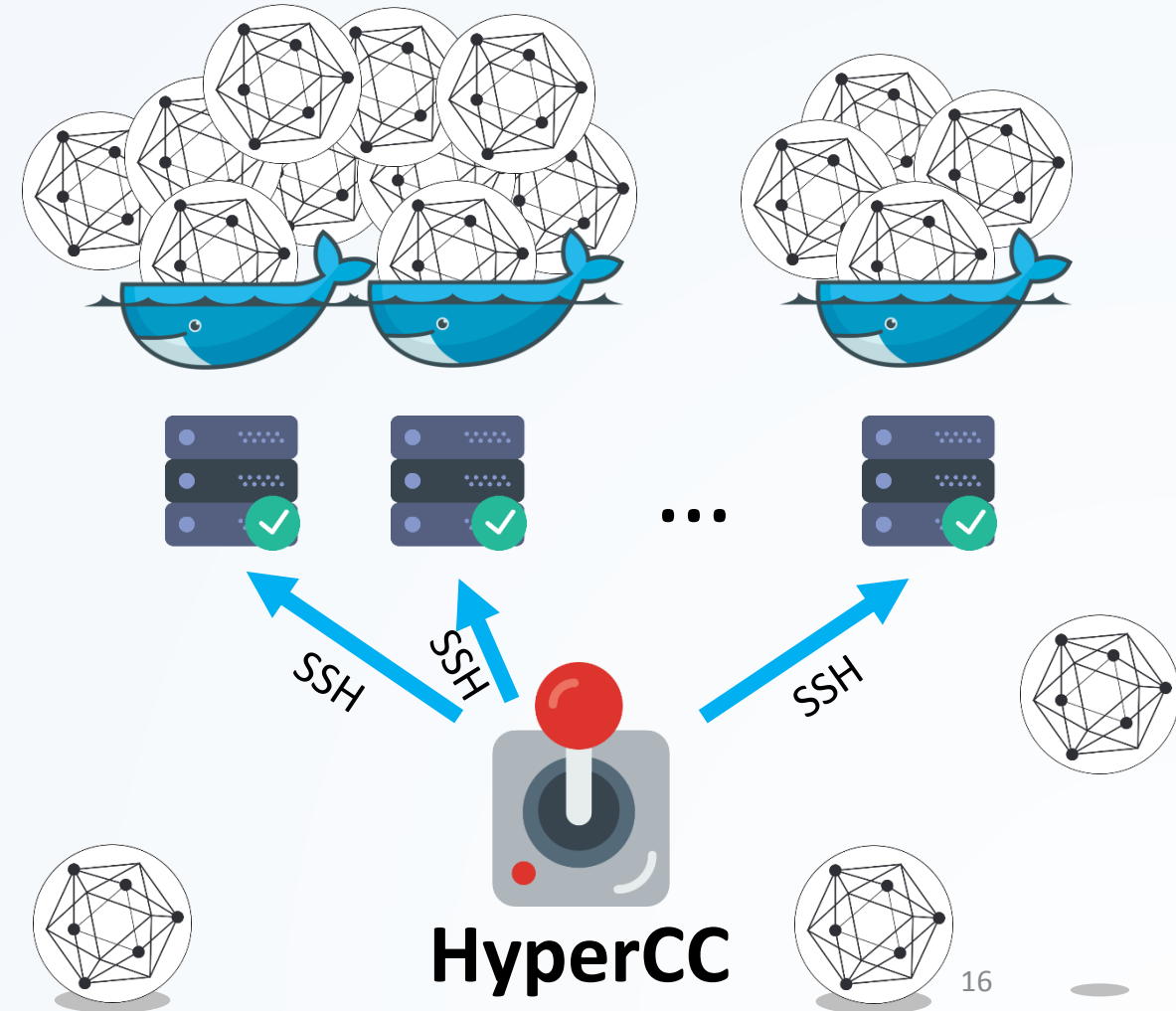
Prototype

- Vanilla github.com/hyperledger/fabric
 - Two consensus plugins
 - No additional features
- Two simple chaincodes
 - Account management
 - Option trading
- Two type of nodes
 - **Validator** node – Validating the blocks.
 - **Non-validator** node – Providing the REST API service.
- Tested on the variety of architectures.



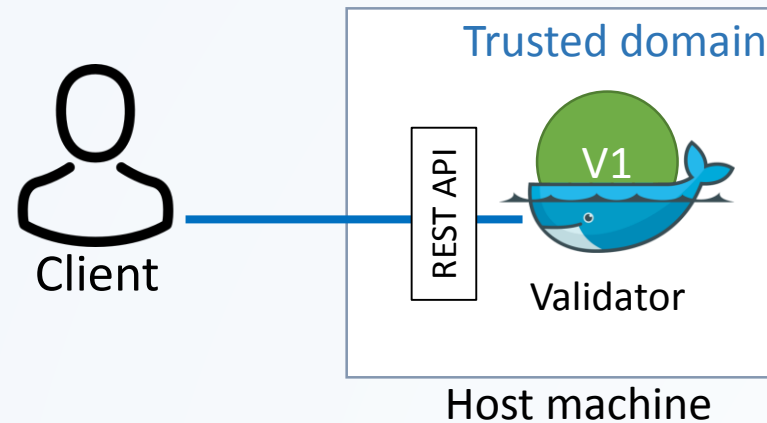
Hyperledger control center - HyperCC

- Currently
 - We use our specific node controller script.
 - Namely, **HyperCC**
- Built with
 - Docker CLI
 - SSH
 - ENV variables
- Parameters (port, volume, etc.)
 - Programmable
 - On-the-fly
- Available functions
 - **Start** – To create the nodes.
 - **Stop** – To terminate the nodes.
 - **Upgrade** – To rolling upgrade the nodes.
 - **SyncDB** – To recovery broken chains (experiment).



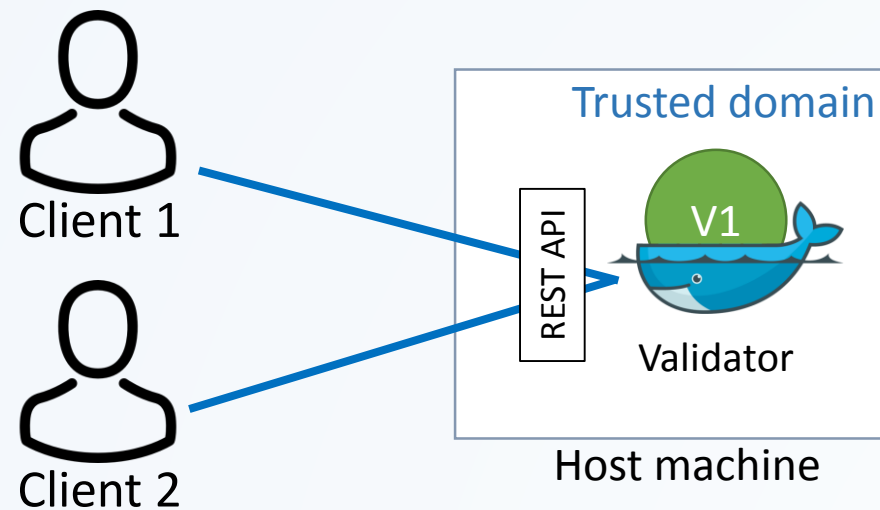
Deployment example

- 1 host machine
 - 1 Client
 - 1 Validator node



Deployment example (2)

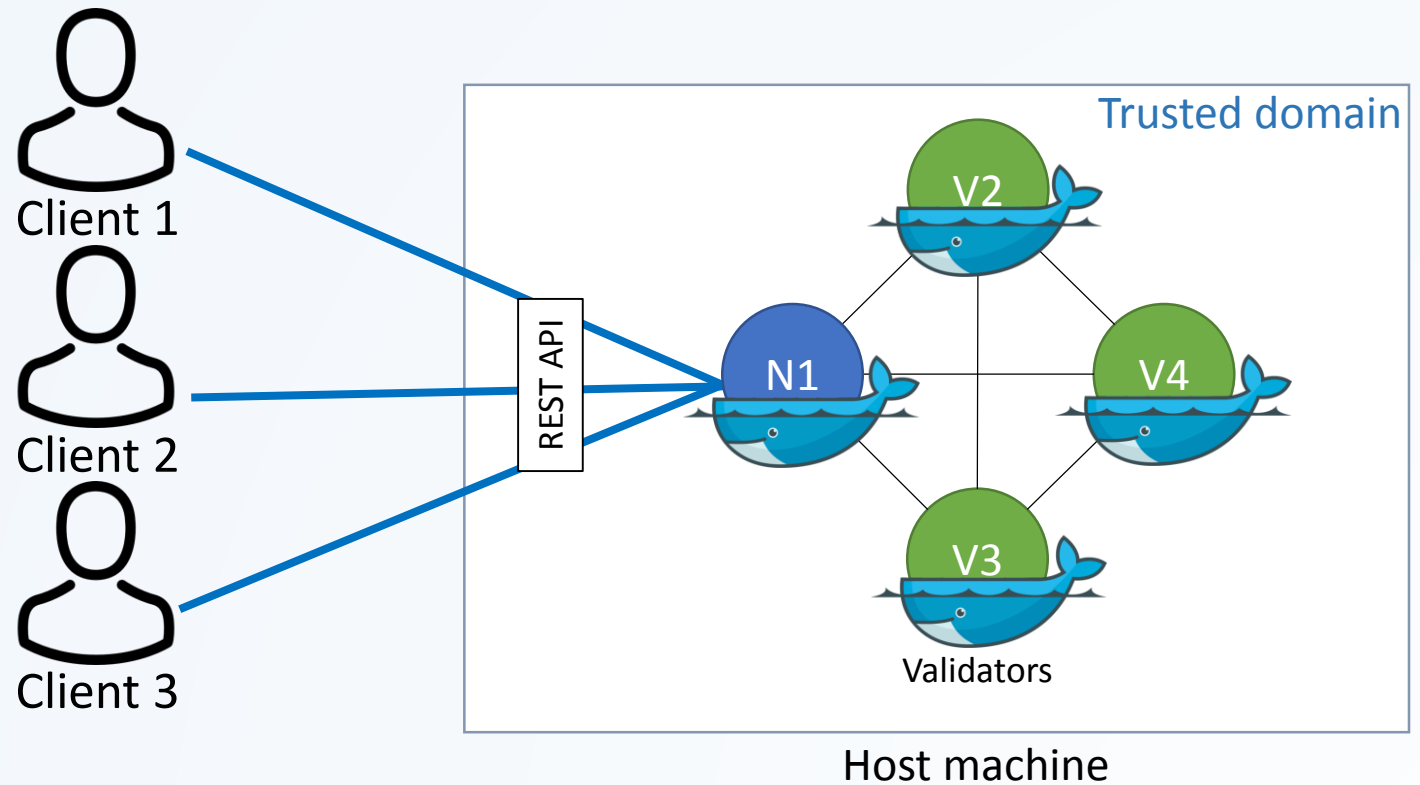
- 1 host machine
 - 2 Clients
 - 1 Validator node



Deployment example (3)

- 1 host machine

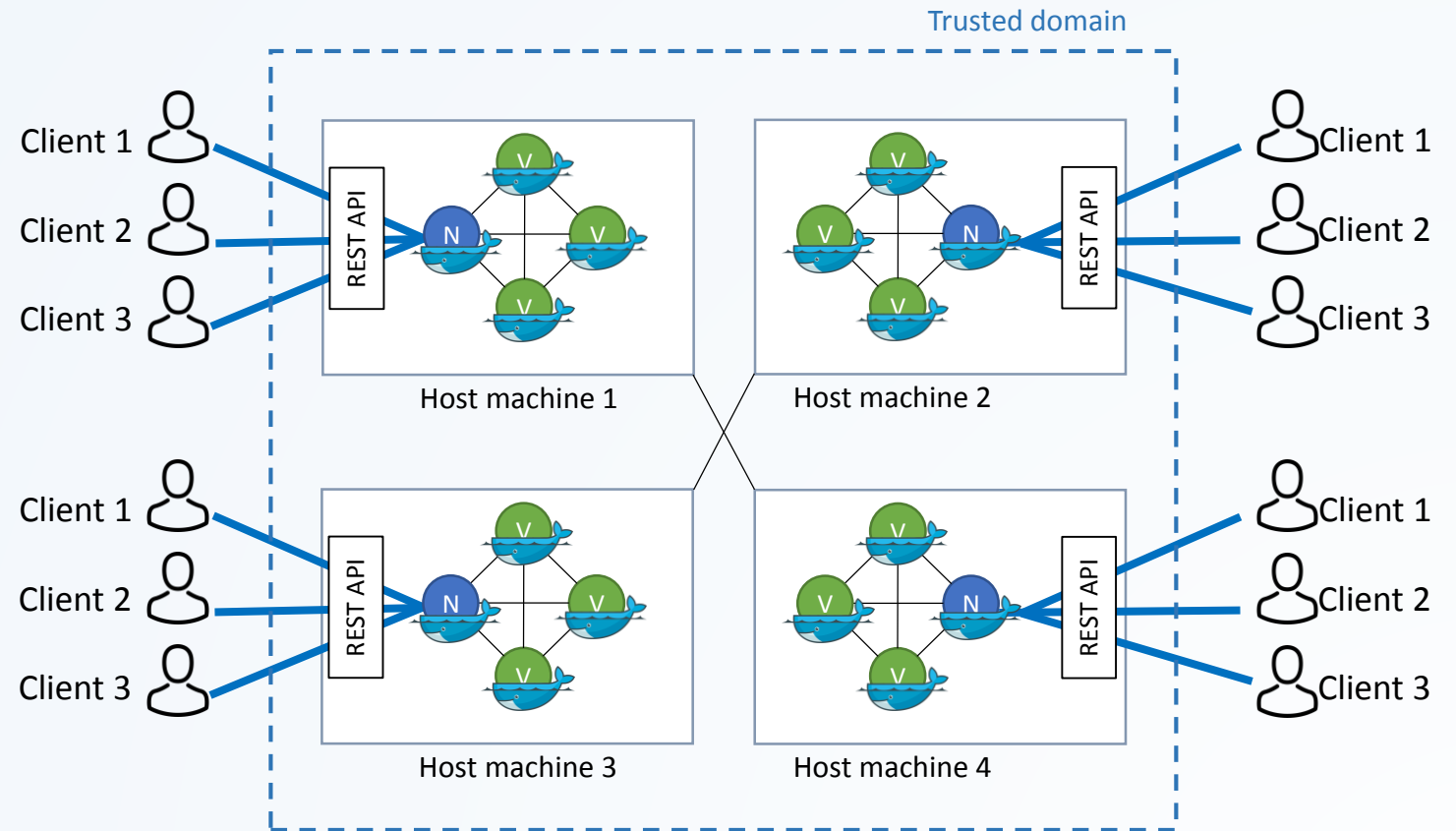
- 3 Clients
- 3 Validator nodes
- 1 Non-validator node



Deployment example (4)

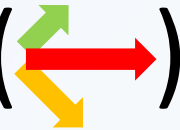
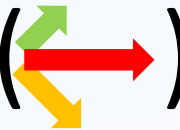
- 4 machines

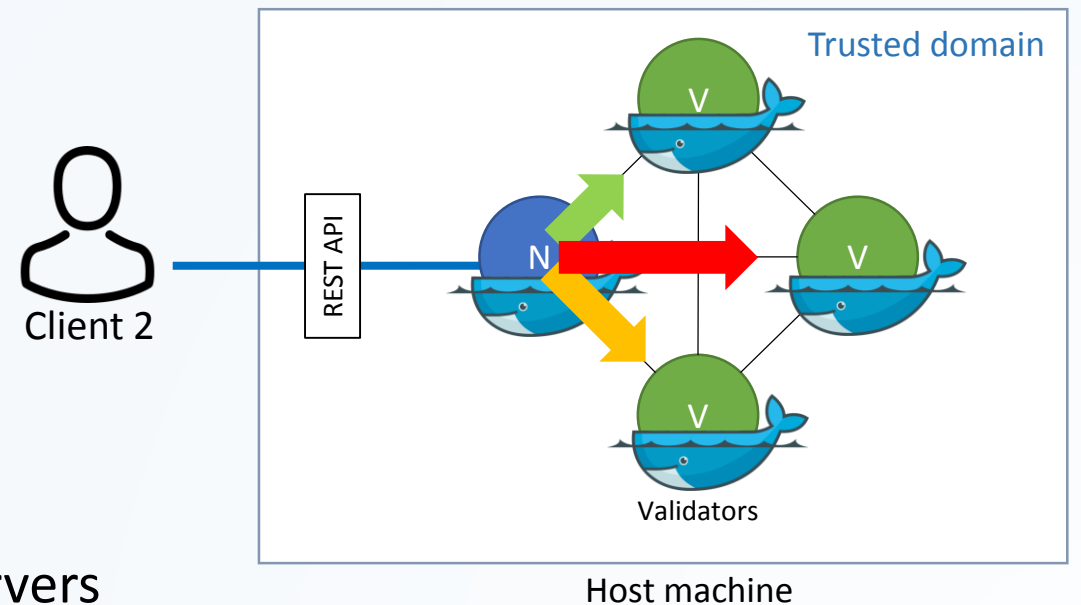
- 3 Clients
- 3 Validator nodes
- 1 Non-validator nodes
- each!



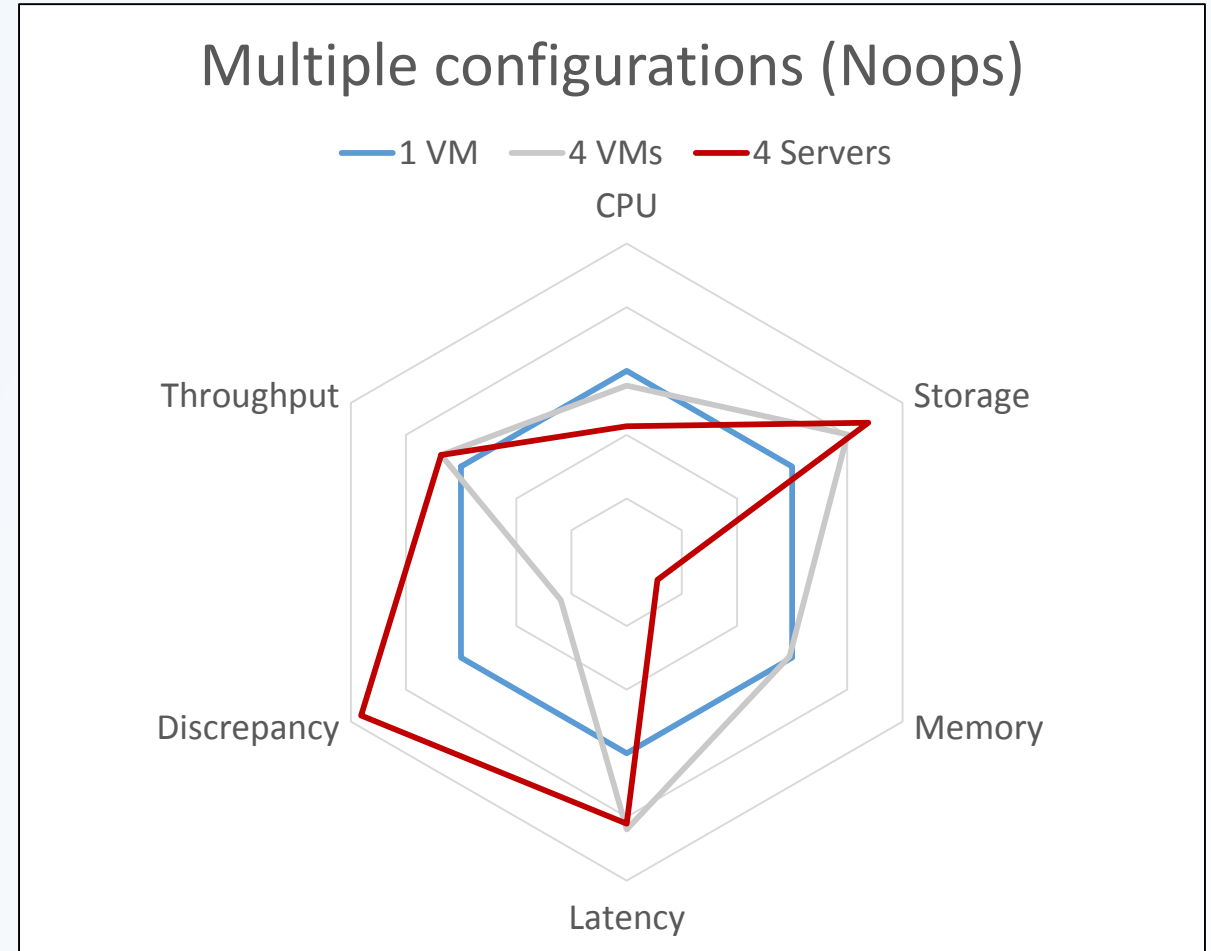
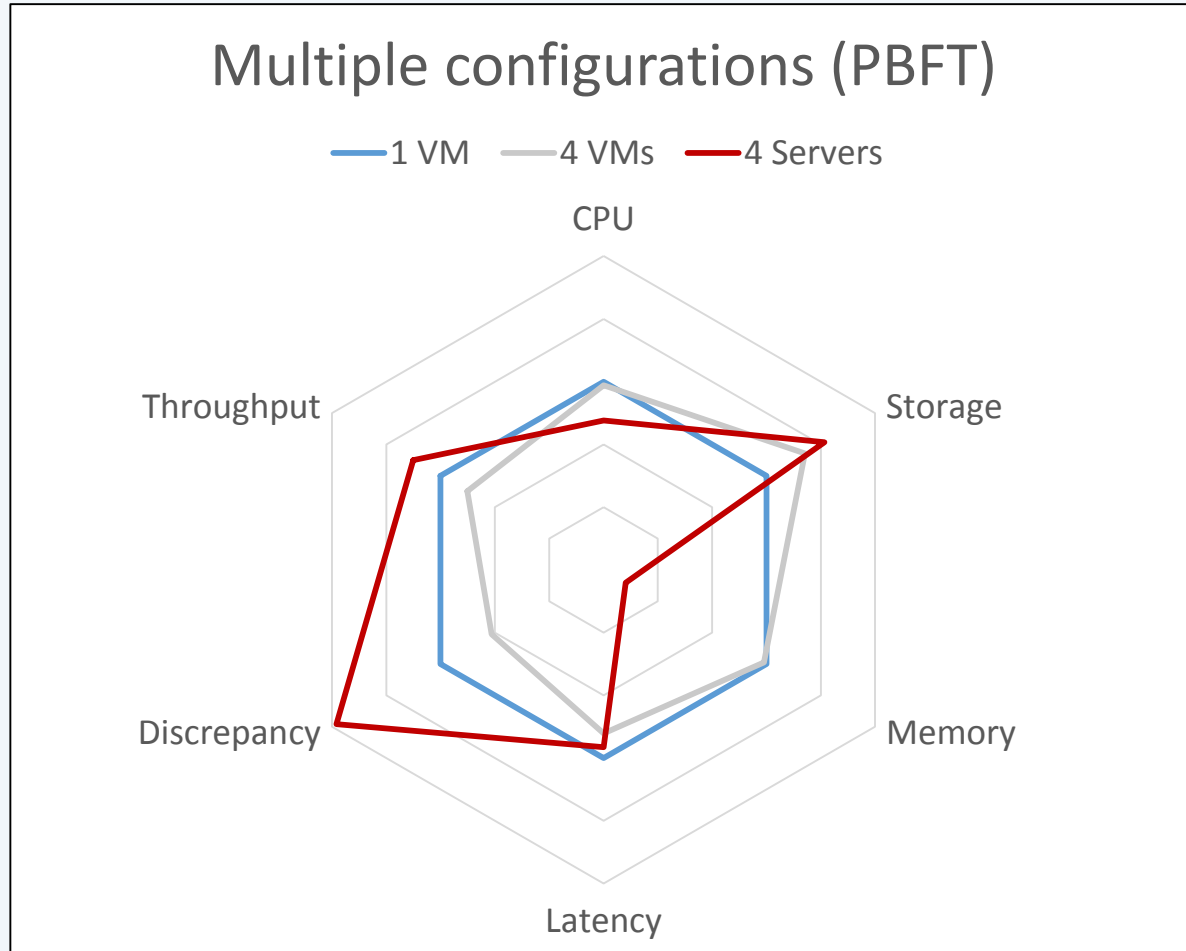
Performance evaluation

Test protocol and configurations

- What we measure?
 - **Throughput**, CPU, memory, storage, **latency** = Avg (), and **discrepancy** = Avg (Diff ())
- How we send the transactions?
 - **REST API**
- What kind of transaction?
 - Randomly **transfer money** between users.
- Docker configuration
 - v. 1.11
 - --storage-driver=overlay
- Node configurations
 - Single VM, multiple VMs, multiple physical servers
 - Node size: 1/2/4/5/6/10/15/20



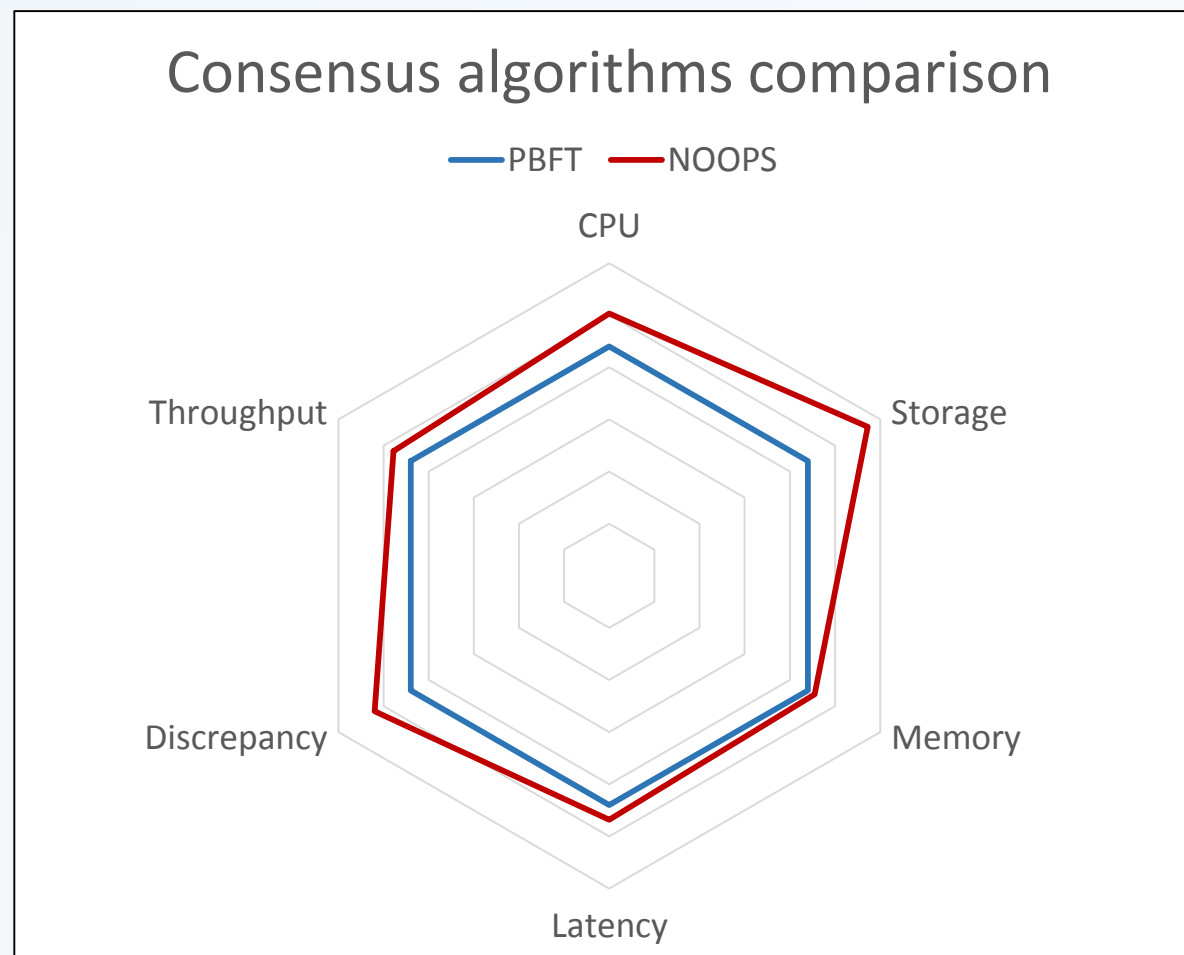
Overall result*



*Higher is better, comparing relatively to 1 VM.

Overall result* (2)

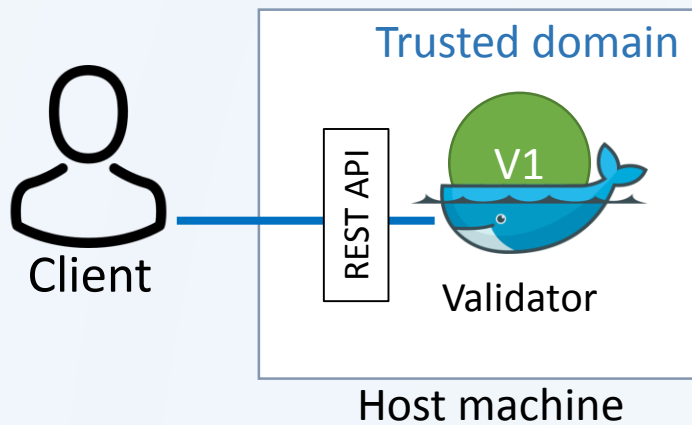
- Noops vs. PBFT



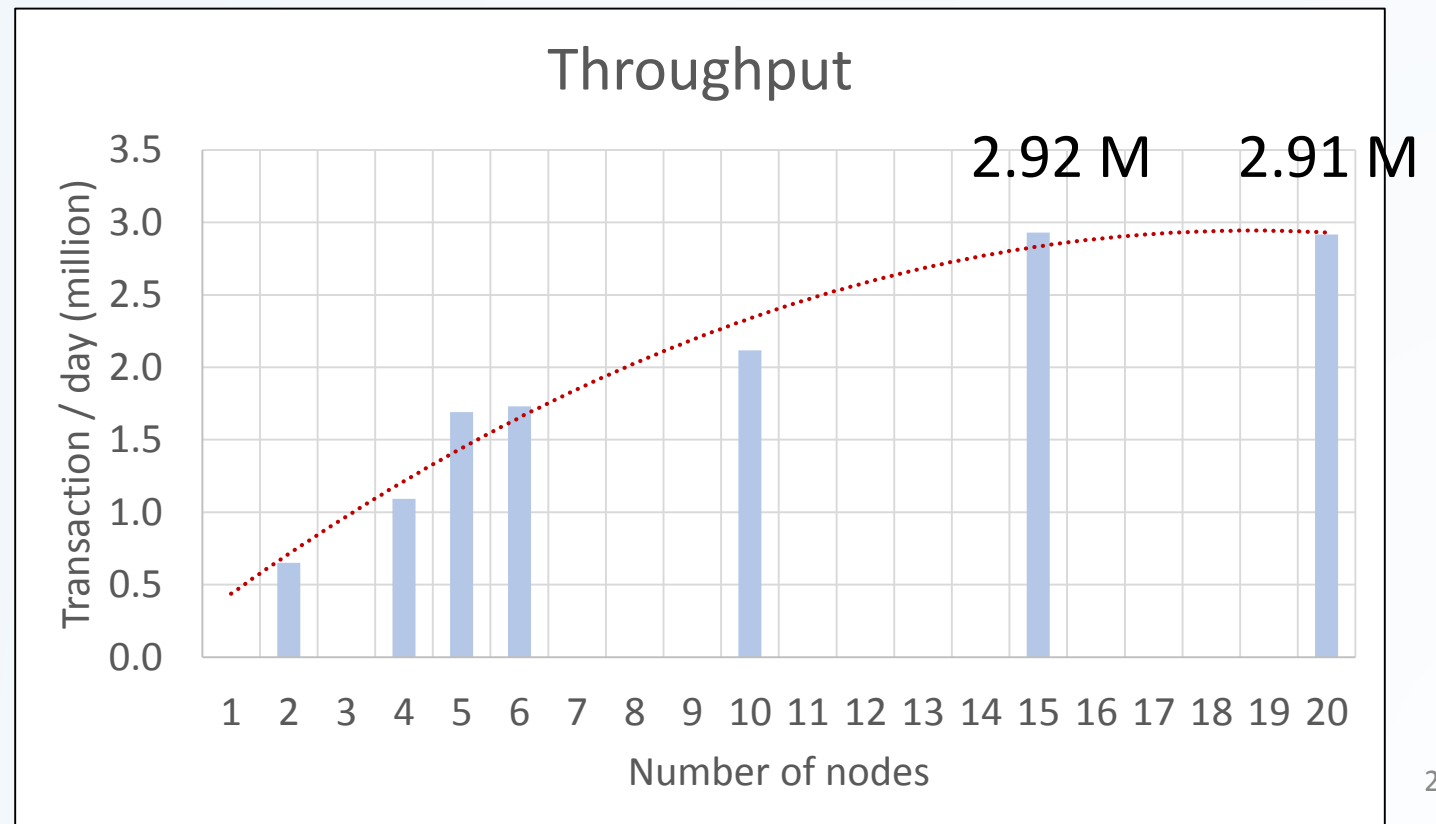
*Higher is better, comparing relatively to PBFT.

Throughput result in detail

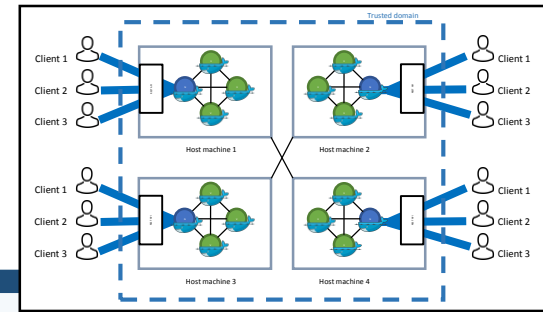
- Estimated for *market hours (7 hours)*
 - Clients connect directly to **validator** node.



Number of node
defines number of machine.



Throughput result in detail (2)



- With non-validator node enabled

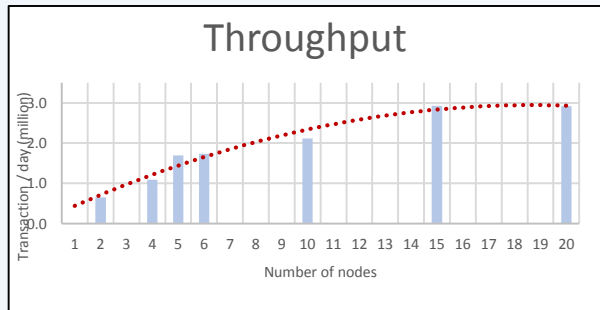
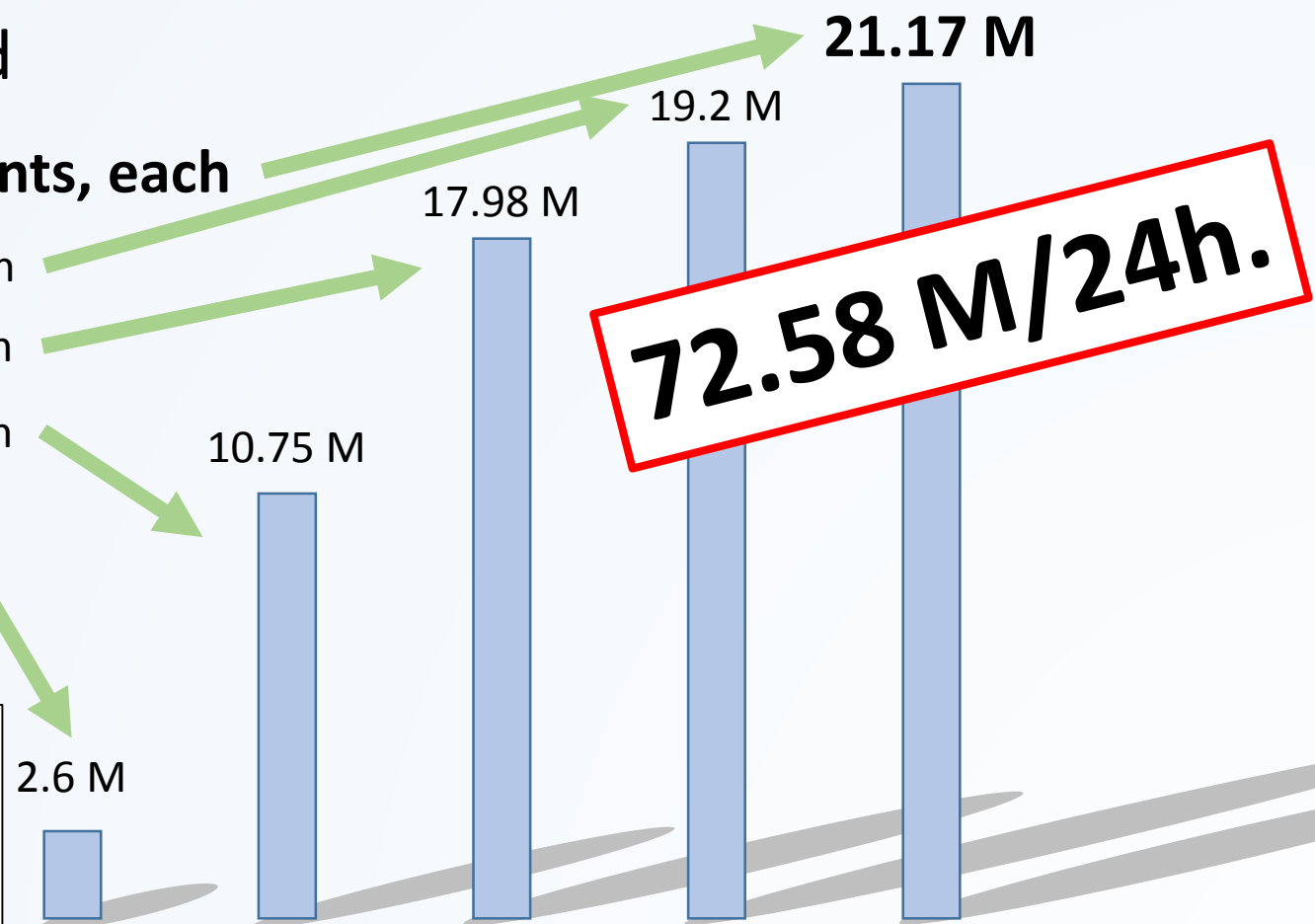
4 machines, 1 V node, 1 N nodes, 3 Clients, each

4 machines, 3 V nodes, 1 N nodes, 3 Clients, each

4 machines, 4 V nodes, 1 N nodes, 2 Clients, each

4 machines, 3 V nodes, 1 N nodes, 1 Clients, each

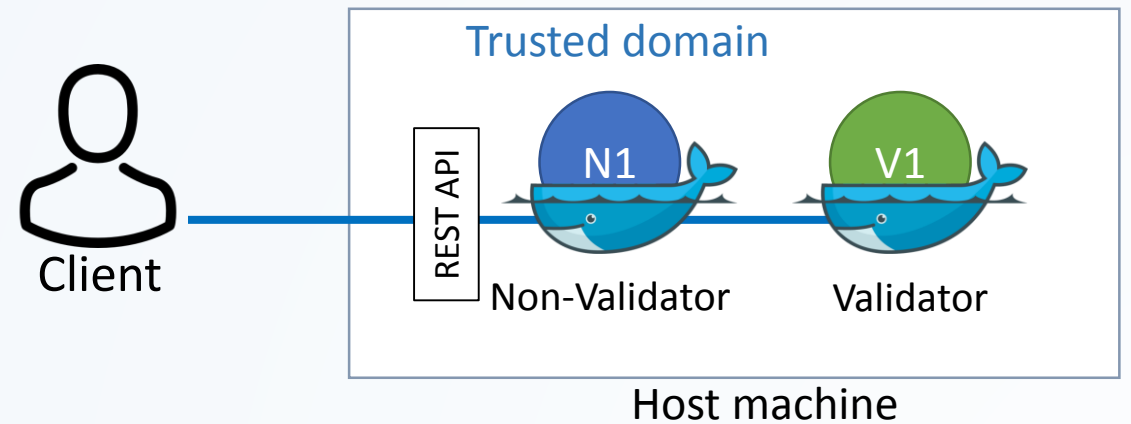
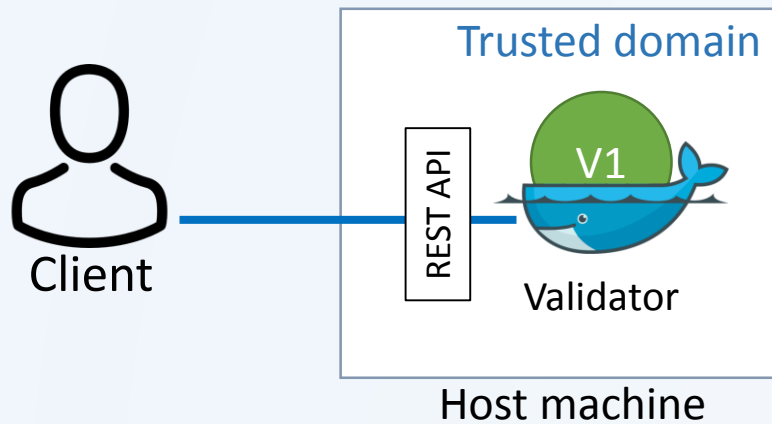
1 machine, 3 V nodes, 1 N node, 1 Client



V = validator
N = non-validator

Result summary

- Noops gives higher performance than PBFT.
- Higher throughput can be achieved with physical servers.
- Better to send transaction through a non-validator node.



 Recommended!

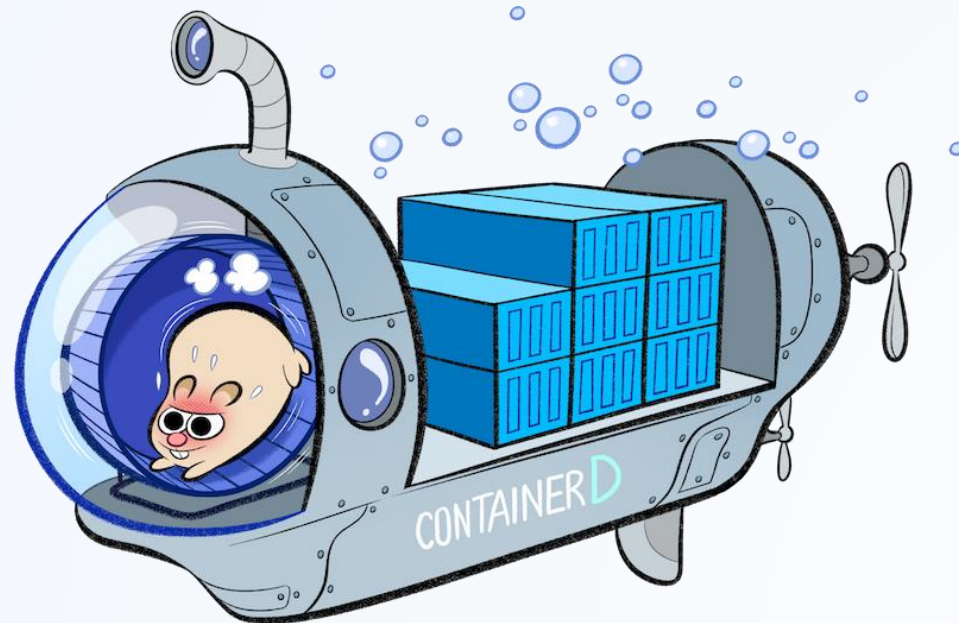
Next challenges

Hyperledger side

1. Backup / Recovery
2. Something in between Noops and PBFT consensus.
 - Hybrid consensus?
3. Throttling rather than silent rejection.
4. ROOT discovery node fail?
5. Way to get transaction result on commit (transaction-specific event?).
6. Deprecated HostConfig in Docker 1.12
 - go-dockerclient still not support Docker 1.12.
 - Chaincode cannot be created.
7. Slow chaincode deployment.
8. Container not recycles automatically.

Docker side

1. Image is quite big
2. Make Docker image smaller
3. Might need runC/containerd



3 things to takeaway

- Hyperledger + Docker provides a powerful blockchain framework.
- Way to deploy a new chaincode without creating a new image.
- Hybrid consensus, fast, adjustable level of trust.

Talk with us

- Our leader:

Fernando Luis Vázquez Cao

fernando.vazquez@sbibits.com

- Me:

Siriwat Kasamwattanarote

siriwat.k@sbibits.com

