IEEE HotICN 2018
August 2018

Towards Application Portability on Blockchains

Kazuyuki Shudo, Reiki Kanda, Kenji Saito

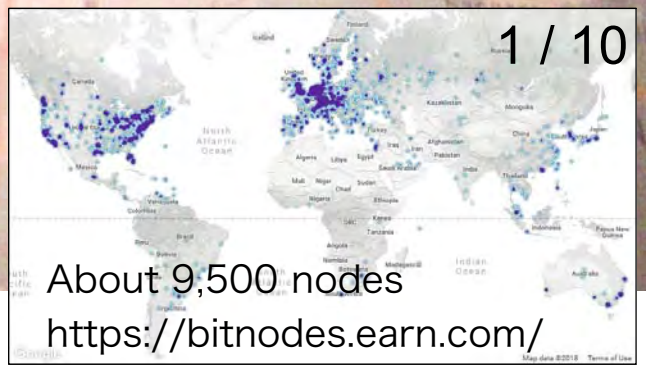
Tokyo Institute of Technology　　Keio University

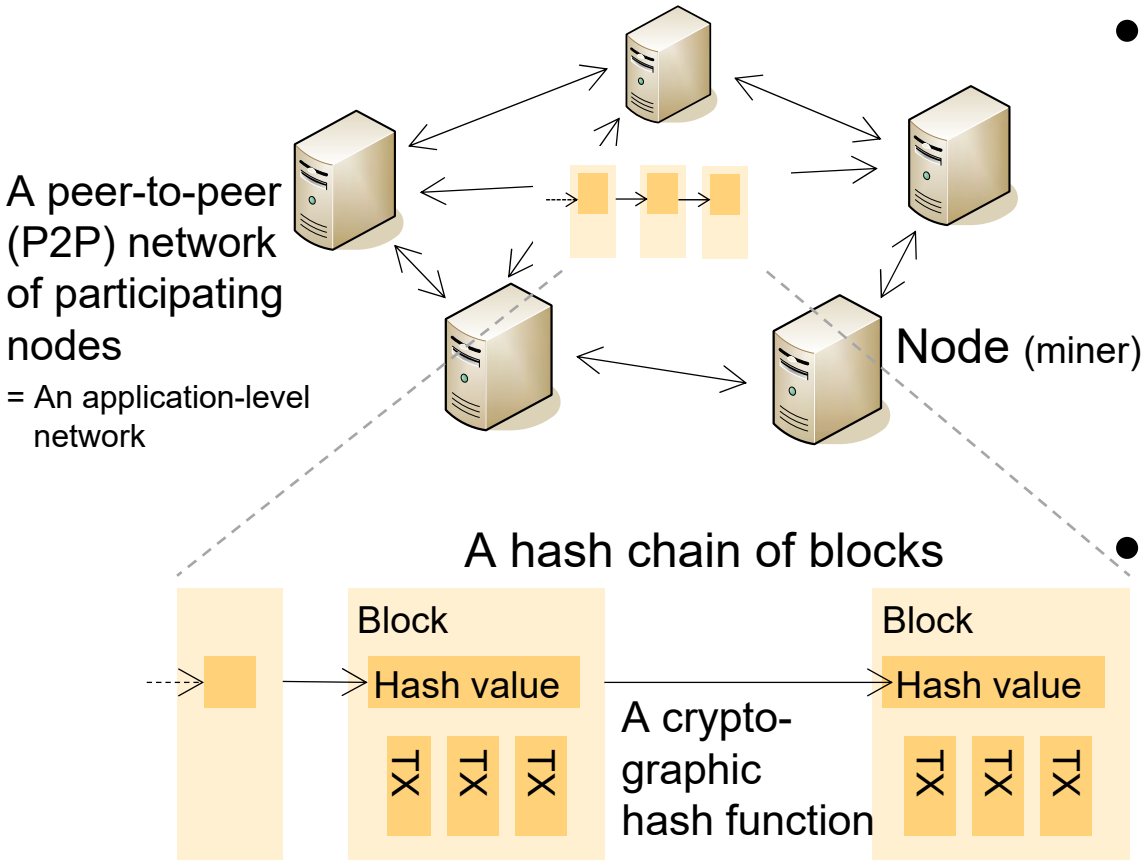
首藤 一幸, 神田 伶樹, 斉藤 賢爾

東京工業大学　　慶應義塾大学

Tokyo Tech

Background: (Public) Blockchain

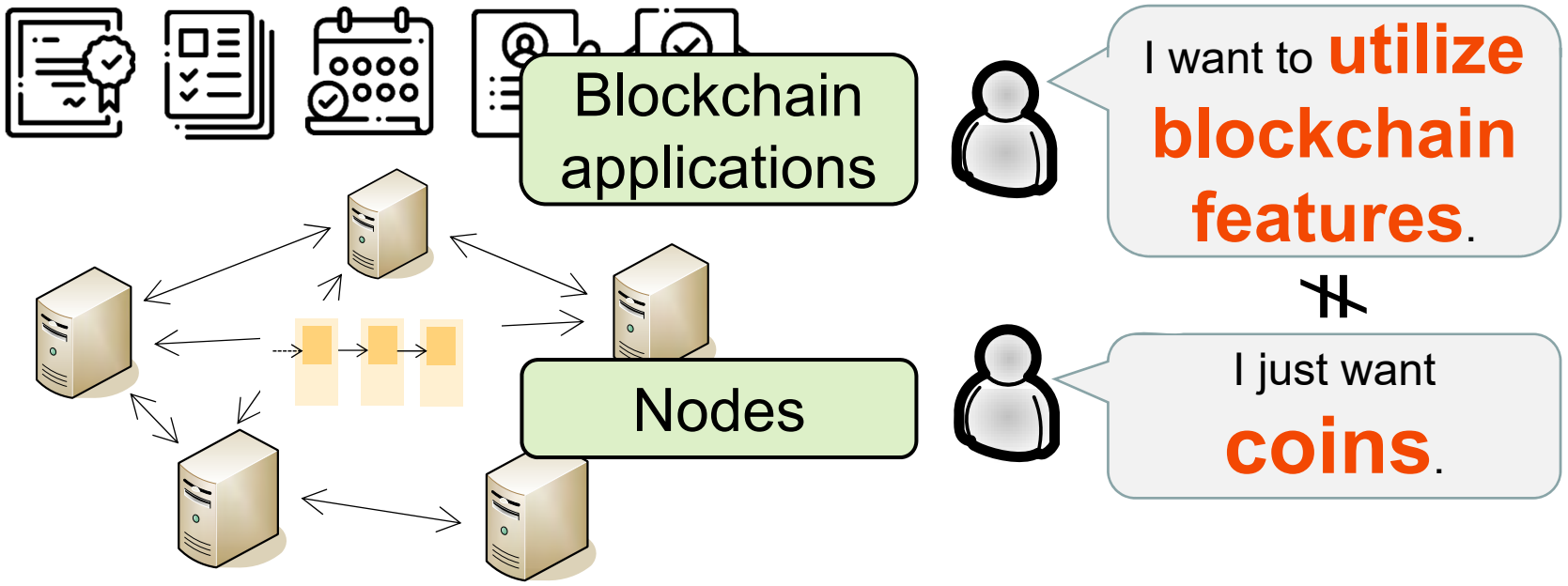




- A distributed system supporting Today's cryptocurrencies
  - Bitcoin's market capitalization is US$ 110B.
  - 9,500 Bitcoin nodes
- A block and transactions in the block are confirmed by Proof of Work / Stakes / Importance / Elapsed Time / … algorithms.
  - A winning node (miner) takes some new **coins**.

Problem: "Incentive mismatch" / インセンティブ (動機) 不整合

- Blockchain nodes and applications do not share common incentive.



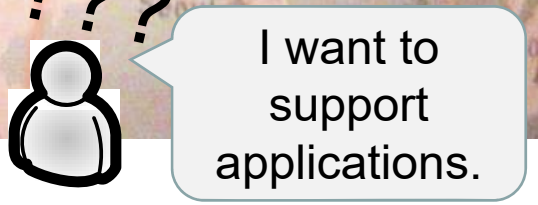
- If a public blockchain cannot provide sufficient economic incentives …

Problem: "Incentive mismatch" / インセンティブ (動機) 不整合

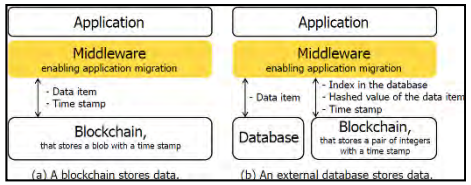
- If a public blockchain cannot provide sufficient economic incentives …
  - E.g. A fall in coin prices

- It loses the ability to confirm transactions securely.
  - Fewer supporting nodes -> Vulnerable to attacks
    - Majority (51%) attack, eclipse attack, …
- In May 2018, such attacks succeeded in a row.
  - Bitcoin Gold: 388,200 BTG = US$ 18,600,000 = 20 億 JPY/円
  - Monacoin: 23,832 MONA = US$ 93,500 = 1,000 万 JPY/円

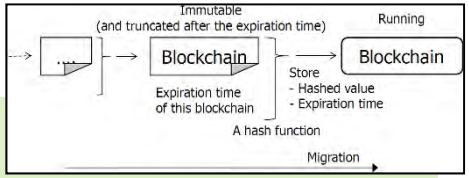
Possible solutions



1. Align their incentives ???
   - … Non-trivial and an open problem
2. Make applications **portable**
   - A potential (and next-best) solution



Middleware design



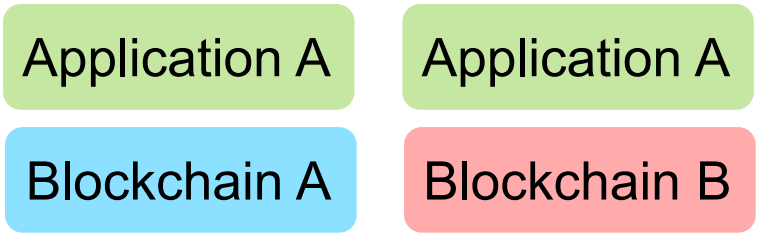
Migration process

- Our contributions
  - (Pointing out "incentive mismatch" problem)
  - A middleware design that enables application migration between blockchains
  - Migration process
  - Note: Implementation is part of future work.

Preparation:
Portability levels

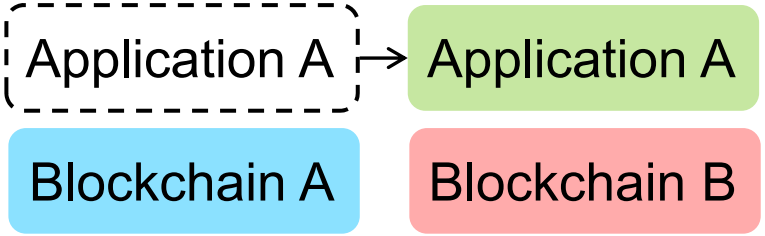
(1) *Runnable*

An application can run on both.



(2) *Migratable* Our target

An application can migrate.



- *Runnable*
  - A common API and smart contract language enable *runnable* portability.
  - Different blockchains running the same middleware also enable it.
  - It is unlikely to be useful because it requires application restart and loses fundamental blockchain features:
    - Proof of data existence
    - Verification of state changes
- *Migratable* – our target

Preparation:
Data to be migrated

In case of a cryptocurrency:

(1) Current state of the application ↔ ・ Account balance

(2) Logs ⟷ ・ Transactions

– Metadata of the states that describe state changes and their time stamps.

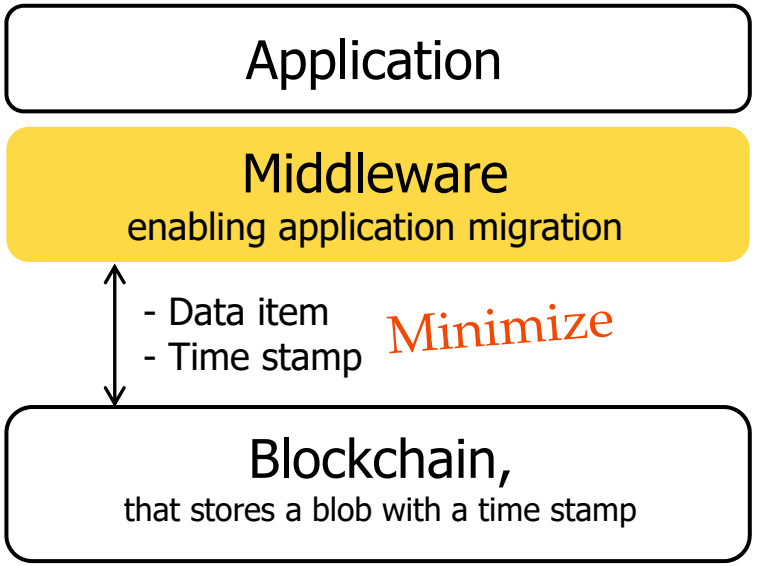
Fundamental blockchain features:

・ Proof of data existence	does not require logs.
・ Verification of state changes	requires logs.

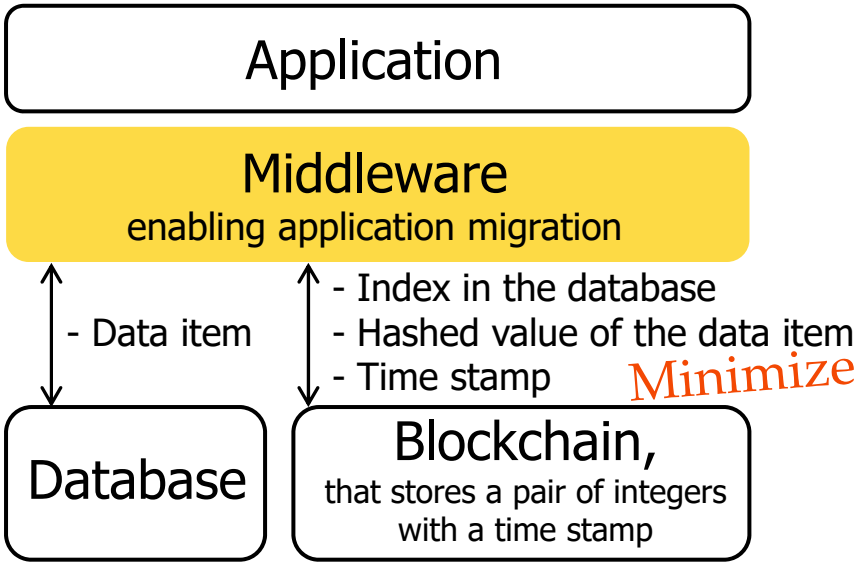
Middleware design

- Design principles
  - Minimize dependence on specific blockchain middleware
    - It stores simple data items, such as numbers or byte sequences, together with time stamps.
  - Do not expect to be able to retain trust in the original (source) blockchain
    - We cannot continue to rely on it in case of imminent collapse of it.

(a) A blockchain stores data.

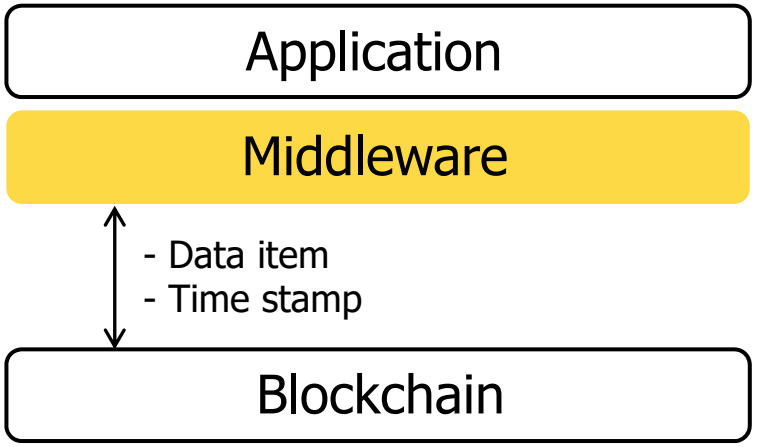


(b) An external database stores data.

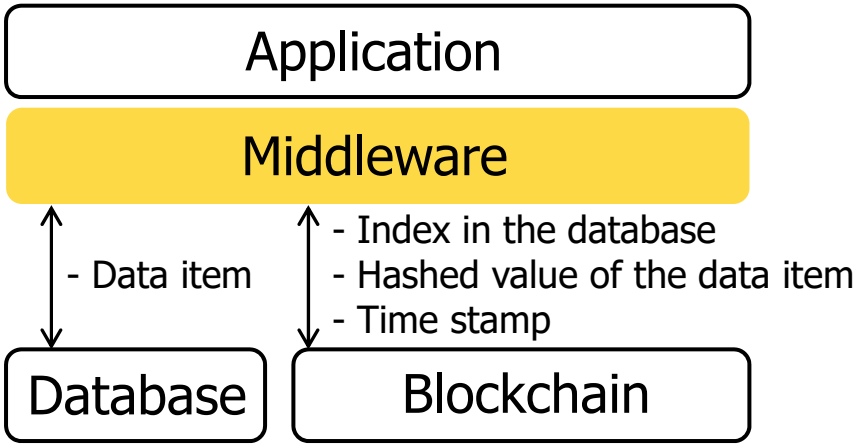


Middleware design (cont'd)

(a) A blockchain stores data.



(b) An external database stores data.



- Pros and cons
  - **(b)** reduces the blockchain size, Instead, the database should be fault-tolerant. Replication or erasure coding work.

To be fault-tolerant

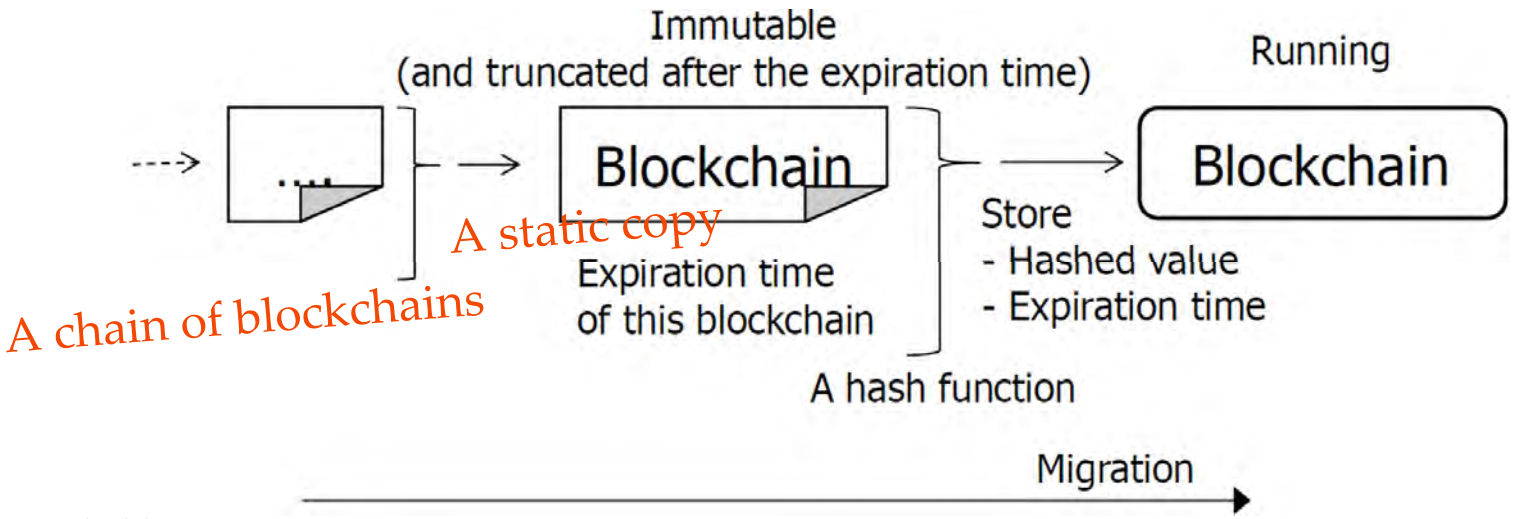
Reduced-size

- If verification of state (data) changes is
  - required: **(a)** must keep the original blockchains.
    (b) must keep all the versions of data.

    } In other words, **Logs** are required.
  - not required: **(a)** can abandon the original blockchains after copying metadata to prove the data exist.
    (b) can overwrite data.

Migration process

In case verification of state (data) changes is required



The middleware

- stores a static copy of the original blockchain, truncating it at the expiration time specified in block height.
  - Not rely on the original blockchain's middleware to still be running and accessible online.
  - Truncating allows us to stop trusting the original blockchain after the expiration time.
- makes a chain of the stored blockchains. It prevents tampering.
- accesses the stored blockchain. It requires a parsing library.

Summary

- "Incentive mismatch" problem pointed out
- Middleware design enabling application migration presented
- Migration process provided

- Future work
  - Proof of the design by implementation
  - Incentive-matched blockchain
  - Interoperability between heterogeneous blockchains

Tokyo Tech