Blockchain-based Open Data: An Approach for Resolving Data Integrity and Transparency

Dinh-Duc Truong

Ho Chi Minh University of Technology – HCMUT

November 21, 2019

Contents

Contents

Introduction

Motivation

- Building a smart city or electronic government trend.
- The security aspects of *open data* – CIA triangle.
- Centralize system disadvantage
  - => Invalid data sets for people to make bad decisions

Introduction

Motivation

- Building a smart city or electronic government trend.
- The security aspects of *open data* – CIA triangle.

- Centralize system disadvantage
  - => Invalid data sets let people to make bad decisions

Introduction

Motivation

- Building a smart city or electronic government trend.
- The security aspects of *open data* – CIA triangle.

- Centralize system disadvantage
  $\Longrightarrow$ Invalid data sets led people to make bad decisions.

Introduction

Motivation

- Building a smart city or electronic government trend.
- The security aspects of *open data* – CIA triangle.

- Centralize system disadvantage

  => Invalid data sets led people to make bad decisions.

- Building a smart city or electronic government trend.
- The security aspects of *open data* – CIA triangle.

- Centralize system disadvantage
  $=>$ Invalid data sets led people to make bad decisions.

Here are a few famous attacks since 2008[1]:

- 2008 - Hackers infiltrate the Brazilian governments systems and inflate the logging quotas to disrupt logging industry.
- 2010 - Hackers use the Stuxnet Worm to make minor changes in Iran's nuclear power program in an attempt to destroy it.
- 2015 - Anonymous begin releasing financial reports exposing firms in the US and China trying to cheat the stock market.
- 2015 - JP Morgan Chase was breached with subsequent attempts at market manipulation.
- 2016 - Both the World Anti-Doping Agency and Democratic National Committee are breached with hackers manipulating their data to embarrass the organisations.

[1] https://www.itsecurityguru.org/2016/11/29/2017-year-data-integrity-breach/

Contents

Open data

Definition

Open data

"Open data is data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and share alike".

Open data

Definition

Open data

"Open data is data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and share alike".

Open data

Properties

+ Technically open:
  - Accessing to data sets: unrestricted, easy.
  - Data formats: computer-readable.
+ Legally open:
  - No restrictions on use and/or redistribution

Open data

Properties

+ Technically open:
  - Accessing to data sets: unrestricted, easy.
  - Data formats: computer-readable.
+ Legally open:
  - No restrictions on use and/or redistribution

Open data

Properties

+ Technically open:
  - Accessing to data sets: unrestricted, easy.
  - Data formats: computer-readable.
+ Legally open:
  - No restrictions on use and/or redistribution

Open data

Properties

+ Technically open:
  - Accessing to data sets: unrestricted, easy.
  - Data formats: computer-readable.
+ Legally open:
  - No restrictions on use and/or redistribution.

Open data

Properties

+ Technically open:
  - Accessing to data sets: unrestricted, easy.
  - Data formats: computer-readable.
+ Legally open:
  - No restrictions on use and/or redistribution.

Open data

Properties

+ Technically open:
  - Accessing to data sets: unrestricted, easy.
  - Data formats: computer-readable.
+ Legally open:
  - No restrictions on use and/or redistribution.

Open data

Benefits

+The government:

- Improving transparency and publicity.
- Reducing the government operation cost.

+The citizens:

- Monitoring the government operation.
- Accessing to larger data resources.

Open data

Benefits

+The government:

- Improving transparency and publicity.
- Reducing the government operation cost.

+The citizens:

- Monitoring the government operation.
- Accessing to larger data resources.

Open data

Benefits

+The government:

- Improving transparency and publicity.
- Reducing the government operation cost.

+The citizens:

- Monitoring the government operation.
- Accessing to larger data resources.

Open data

Benefits

+The government:

- Improving transparency and publicity.
- Reducing the government operation cost.

+The citizens:

- Monitoring the government operation.
- Accessing to larger data resources.

Open data

Benefits

+The government:

- Improving transparency and publicity.
- Reducing the government operation cost.

+The citizens:

- Monitoring the government operation.
- Accessing to larger data resources.

Open data

Benefits

+The government:

- Improving transparency and publicity.
- Reducing the government operation cost.

+The citizens:

- Monitoring the government operation.
- Accessing to larger data resources.

Open data

Benefits

+The government:

- Improving transparency and publicity.
- Reducing the government operation cost.

+The citizens:

- Monitoring the government operation.
- Accessing to larger data resources.

Open data

Security issues

- In terms of the law:
  - no legal violation;
  - no disclosure of economic secrets;
  - no leak of personal information;
  - no publication of information on infrastructure.
- In terms of the technical:
  - ensuring the availability and integrity of data.

Open data

Security issues

- In terms of the law:
  - no legal violation;
  - no disclosure of economic secrets;
  - no leak of personal information;
  - no publication of information on infrastructure.
- In terms of the technical:
  - ensuring the availability and integrity of data.

Open data

Security issues

- In terms of the law:
  - no legal violation;
  - no disclosure of economic secrets;
  - no leak of personal information;
  - no publication of information on infrastructure.
- In terms of the technical:
  - ensuring the availability and integrity of data.

Open data

Security issues

- In terms of the law:
  - no legal violation;
  - no disclosure of economic secrets;
  - no leak of personal information;
  - no publication of information on infrastructure.
- In terms of the technical:
  - ensuring the availability and integrity of data.

Open data

Security issues

- In terms of the law:
  - no legal violation;
  - no disclosure of economic secrets;
  - no leak of personal information;
  - no publication of information on infrastructure.
- In terms of the technical:
  - ensuring the availability and integrity of data.

Open data

Security issues

- In terms of the law:
  - no legal violation;
  - no disclosure of economic secrets;
  - no leak of personal information;
  - no publication of information on infrastructure.
- In terms of the technical:
  - ensuring the availability and integrity of data.

Open data

Security issues

- In terms of the law:
  - no legal violation;
  - no disclosure of economic secrets;
  - no leak of personal information;
  - no publication of information on infrastructure.
- In terms of the technical:
  - ensuring the availability and integrity of data.

Blockchain

A **blockchain** is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

Blockchain

A **blockchain** is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

Blockchain

A **blockchain** is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

Types of blockchain:

Blockchain

A **blockchain** is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

Types of blockchain:

- Public blockchain.

Blockchain

A **blockchain** is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

Types of blockchain:

- Public blockchain.
- Private blockchain.

Blockchain

A **blockchain** is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

Types of blockchain:

- Public blockchain.
- Private blockchain.
- Consortium blockchain.

Hyperledger Fabric

Definition

Hyperledger Fabric:

- Private blockchain.
- Flexible consensus protocol.

Hyperledger Fabric

Definition

Hyperledger Fabric:

- Private blockchain.
- Flexible consensus protocol.

Hyperledger Fabric

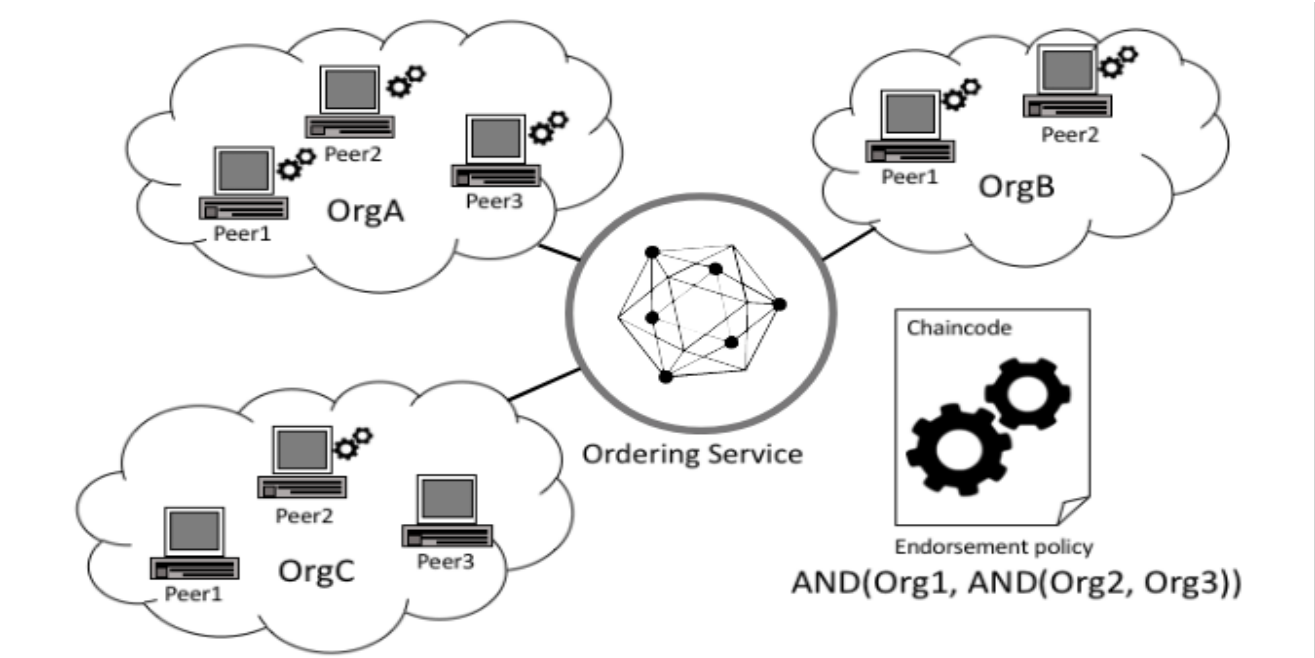
Definition

Hyperledger Fabric:

- Private blockchain.
- Flexible consensus protocol.

Hyperledger Fabric:

- Private blockchain.
- Flexible consensus protocol.



InterPlanetary File System

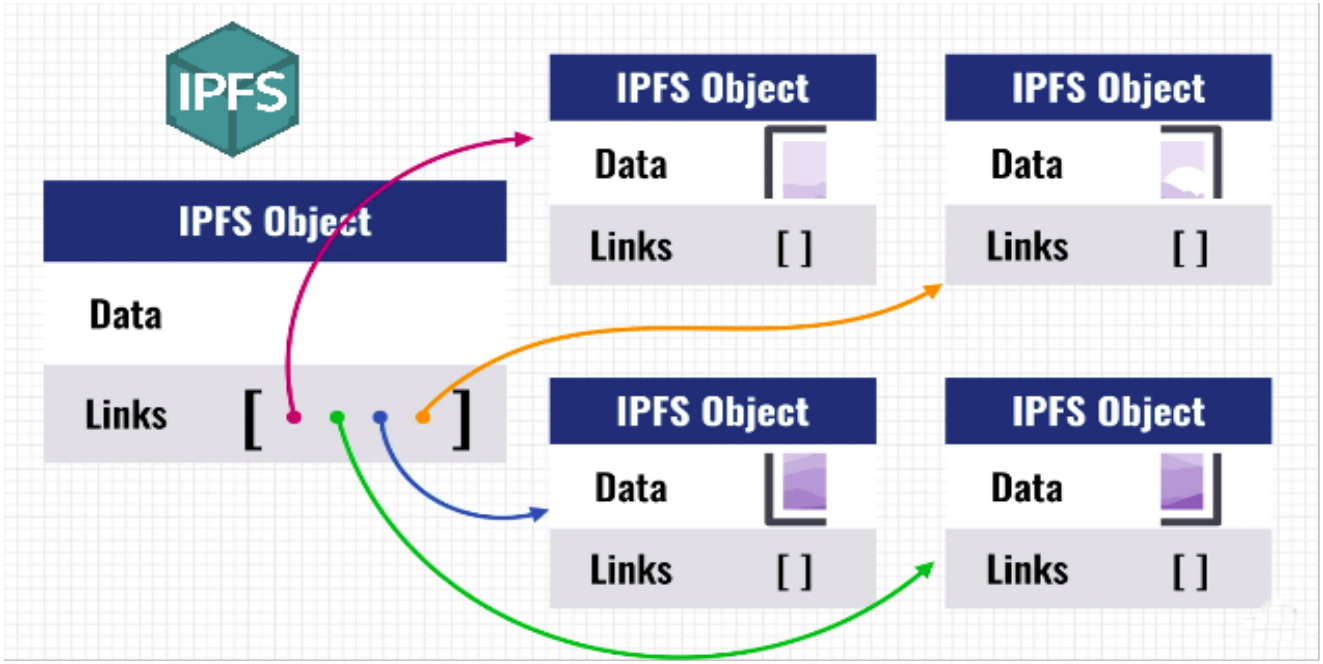
Definition

IPFS

InterPlanetary File System, IPFS for short, is a peer-to-peer distributed file system for storing and sharing hypermedia files over a network.

IPFS

InterPlanetary File System, IPFS for short, is a peer-to-peer distributed file system for storing and sharing hypermedia files over a network.



Contents

Related works

Non blockchain-based

1. US open data portal
   - since 2009;
   - provides a number of 229,371 data sets in various formats and related to many fields.
2. Ho Chi Minh open data portal
   - since 2017;
   - still thin in the content due to lacking data sets.

Related works

Non blockchain-based

1. US open data portal
   - since 2009;
   - provides a number of 229,371 data sets in various formats and related to many fields.
2. Ho Chi Minh open data portal
   - since 2017;
   - still thin in the content due to lacking data sets.

Related works

Non blockchain-based

1. US open data portal
   - since 2009;
   - provides a number of 229,371 data sets in various formats and related to many fields.
2. Ho Chi Minh open data portal
   - since 2017;
   - still thin in the content due to lacking data sets.

Related works

Non blockchain-based

1. US open data portal
   - since 2009;
   - provides a number of 229,371 data sets in various formats and related to many fields.
2. Ho Chi Minh open data portal
   - since 2017;
   - still thin in the content due to lacking data sets.

Related works

Non blockchain-based

1. US open data portal
   - since 2009;
   - provides a number of 229,371 data sets in various formats and related to many fields.
2. Ho Chi Minh open data portal
   - since 2017;
   - still thin in the content due to lacking data sets.

Related works

Non blockchain-based

1. US open data portal
   - since 2009;
   - provides a number of 229,371 data sets in various formats and related to many fields.
2. Ho Chi Minh open data portal
   - since 2017;
   - still thin in the content due to lacking data sets.

3. Vienna open data portal
   - called "Data Notarization Blockchain" in 2018;
   - uses public blockchain.

Related works

Blockchain-based

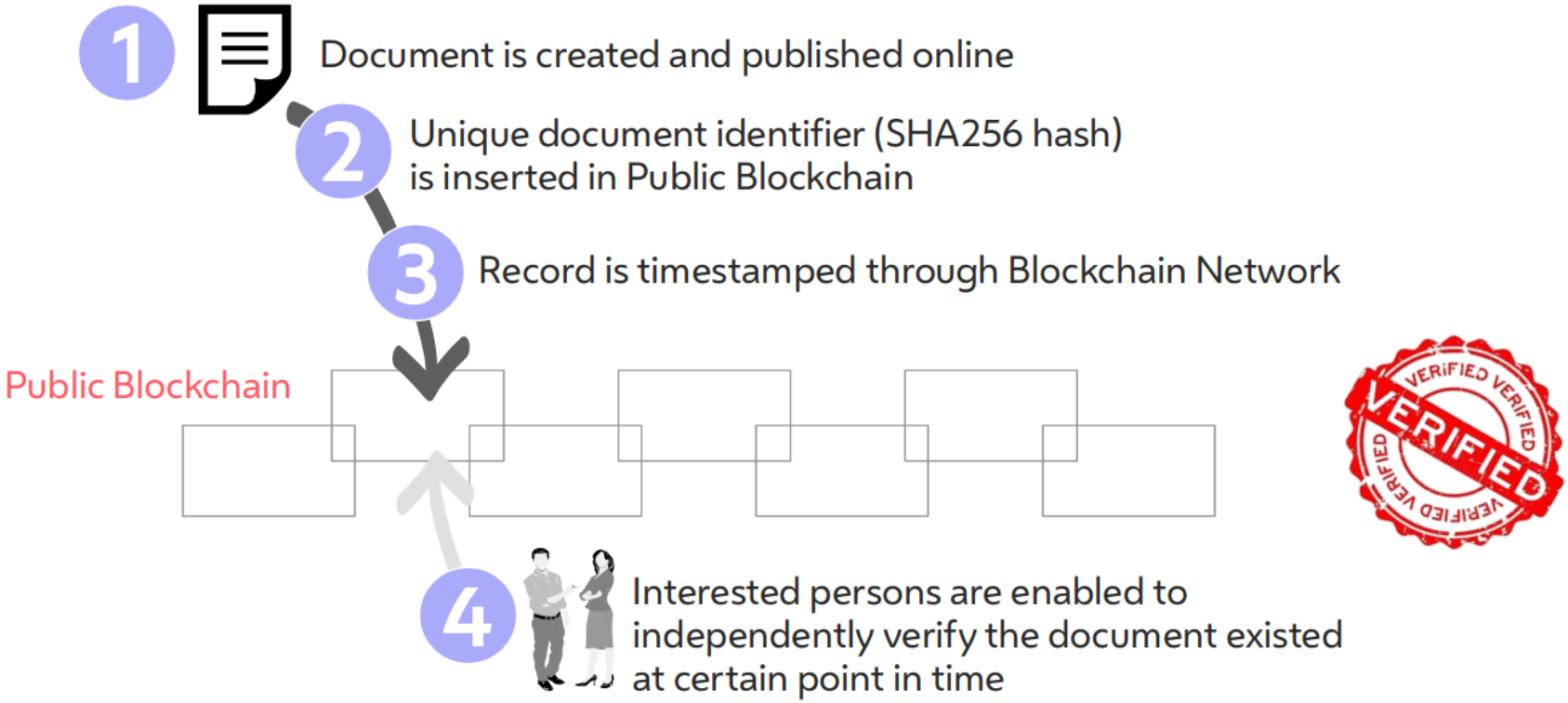
3. Vienna open data portal
   - called "Data Notarization Blockchain" in 2018;
   - uses public blockchain.

Related works

Blockchain-based

3. Vienna open data portal
   - called "Data Notarization Blockchain" in 2018;
   - uses public blockchain.

3. Vienna open data portal
   - called "Data Notarization Blockchain" in 2018;
   - uses public blockchain.



Contents

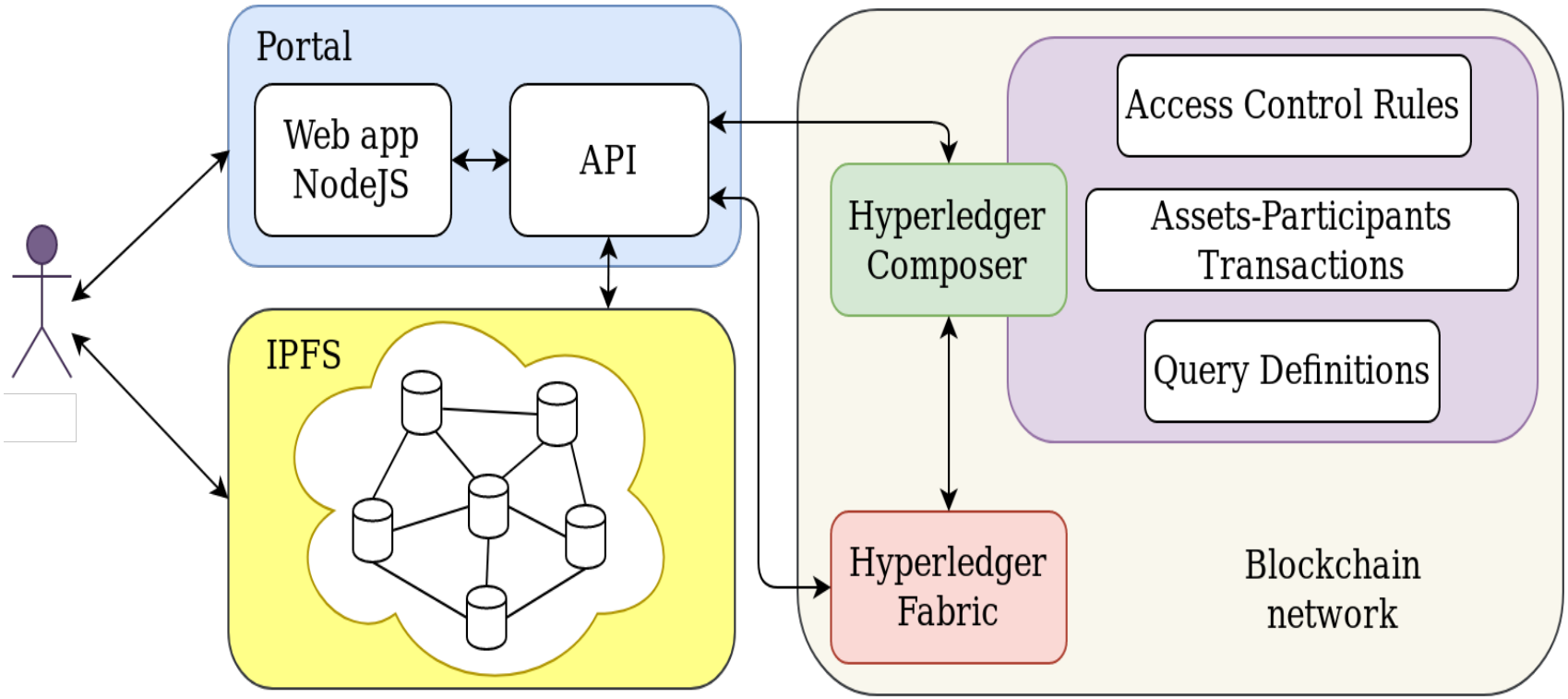
Our system includes:

Our system includes: a **Portal**,

Our system includes: a **Portal**, a distributed file storage system **IPFS**

Our system includes: a **Portal**, a distributed file storage system **IPFS** and a private blockchain network **Hyperledger Fabric**.

Our system includes: a **Portal**, a distributed file storage system **IPFS** and a private blockchain network **Hyperledger Fabric**.



Why Hyperledger Fabric and IPFS?

- We want to decentralize our system as much as we can.
- Hyperledger Fabric: authenticates data contributor, traces data log, checks data integrity, enhances the system transparency.
- IPFS: ensures the data integrity and availability.

Why Hyperledger Fabric and IPFS?

- We want to decentralize our system as much as we can.
- Hyperledger Fabric: authenticates data contributor, traces data log, checks data integrity, enhances the system transparency.
- IPFS: ensures the data integrity and availability.

Why Hyperledger Fabric and IPFS?

- We want to decentralize our system as much as we can.
- Hyperledger Fabric: authenticates data contributor, traces data log, checks data integrity, enhances the system transparency.
- IPFS: ensures the data integrity and availability.

Why Hyperledger Fabric and IPFS?

- We want to decentralize our system as much as we can.
- Hyperledger Fabric: authenticates data contributor, traces data log, checks data integrity, enhances the system transparency.
- IPFS: ensures the data integrity and availability.

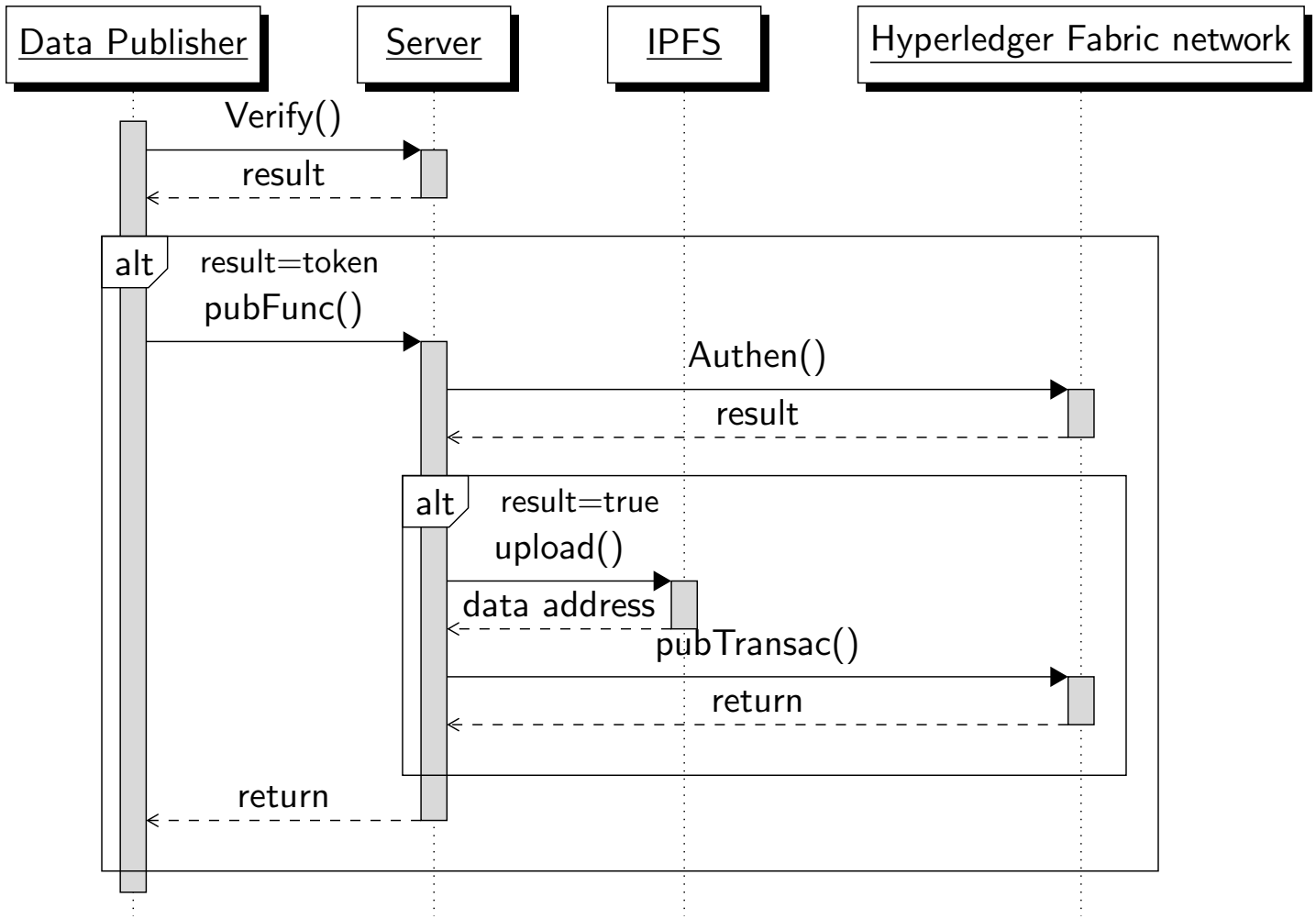
System features

Our system has two main features:

- Publishing the data sets.
- Downloading-Verifying the data sets.

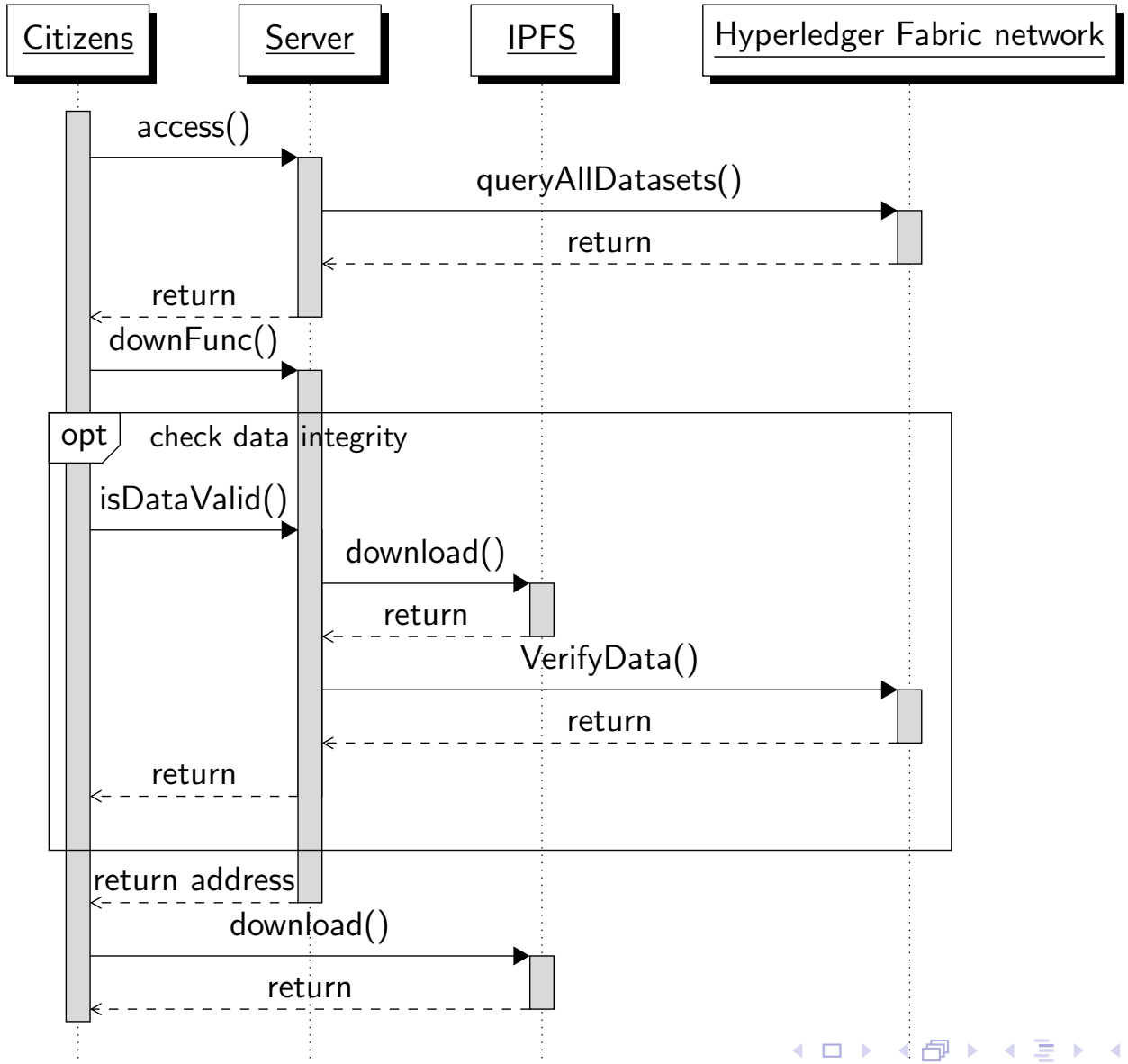
System features

Publishing the data sets



System features

Downloading-verifying the data sets



Contents

Implementation

Our implementation included:

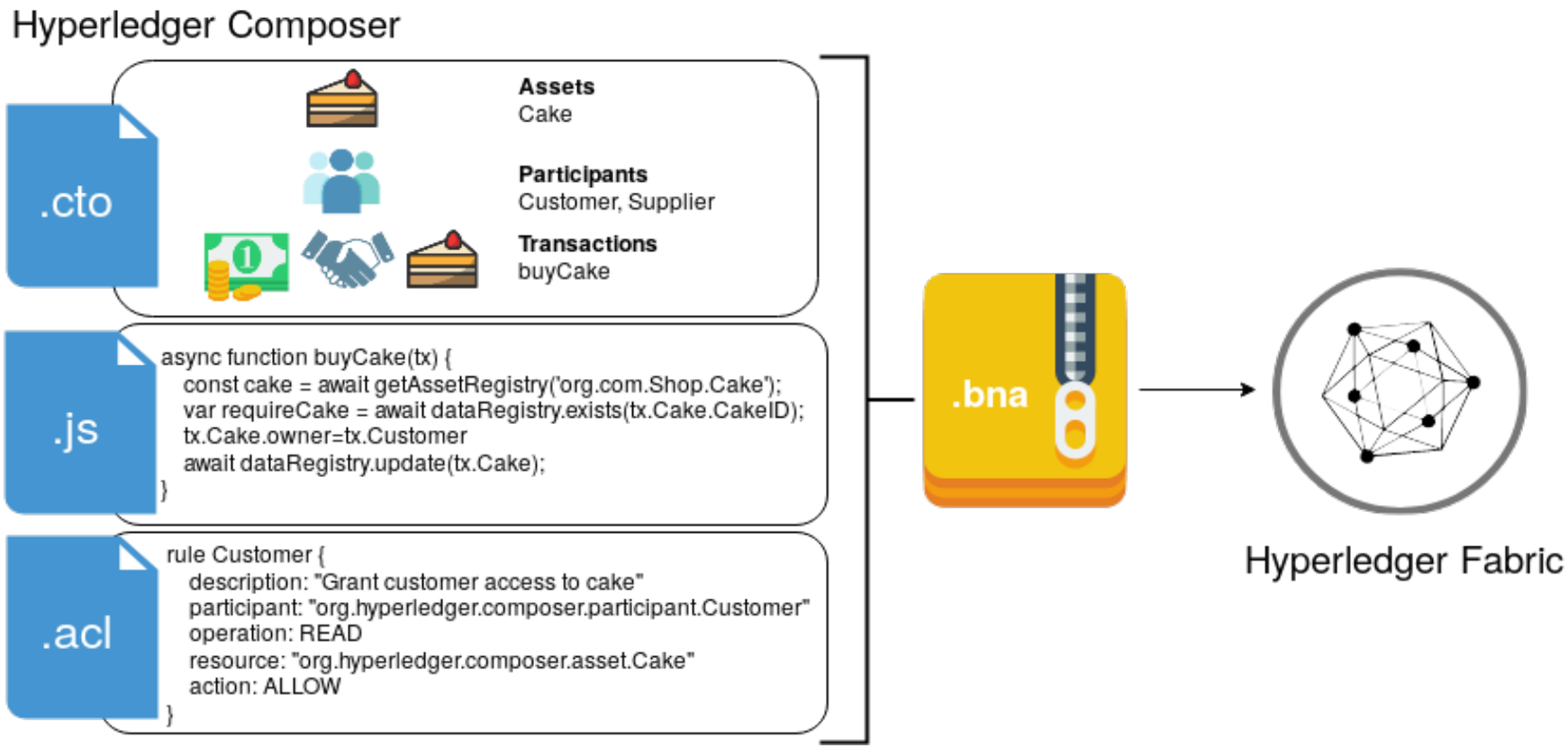
- Hyperledger Fabric network
- The open data portal

Hyperledger Fabric network

Hyperledger Composer

Hyperledger Composer is an extensive, open development toolset and framework to make developing blockchain applications easier.

Hyperledger Composer is an extensive, open development toolset and framework to make developing blockchain applications easier.



We use Hyperledger Composer to define our customize chaincode.

1. Participants, Assets
   - The data contributors represented as **DataPublisher** object.

```
participant DataPublisher identified by DataPublisherID {
  o String DataPublisherID
  o info PublisherInfo
}
```

   - The metadata of the data sets represented as **Data** object.

```
asset Data identified by DataID {
  o String DataID
  o metaData meta
  o String checksum
  o String cid
  --> DataPublisher publisher
}
```

We use Hyperledger Composer to define our customize chaincode.

1. Participants, Assets
   - The data contributors represented as **DataPublisher** object.

```
participant DataPublisher identified by DataPublisherID {
  o String DataPublisherID
  o info PublisherInfo
}
```

   - The metadata of the data sets represented as **Data** object.

```
asset Data identified by DataID {
  o String DataID
  o metaData meta
  o String checksum
  o String cid
  --> DataPublisher publisher
}
```

We use Hyperledger Composer to define our customize chaincode.

1. Participants, Assets
   - The data contributors represented as **DataPublisher** object.

```
participant DataPublisher identified by DataPublisherID {
  o String DataPublisherID
  o info PublisherInfo
}
```

   - The metadata of the data sets represented as **Data** object.

```
asset Data identified by DataID {
  o String DataID
  o metaData meta
  o String checksum
  o String cid
  --> DataPublisher publisher
}
```

We use Hyperledger Composer to define our customize chaincode.

1. Participants, Assets
   - The data contributors represented as **DataPublisher** object.

```
participant DataPublisher identified by DataPublisherID {
  o String DataPublisherID
  o info PublisherInfo
}
```

   - The metadata of the data sets represented as **Data** object.

```
asset Data identified by DataID {
  o String DataID
  o metaData meta
  o String checksum
  o String cid
  --> DataPublisher publisher
}
```

We use Hyperledger Composer to define our customize chaincode.

1. Participants, Assets
   - The data contributors represented as **DataPublisher** object.

```
participant DataPublisher identified by DataPublisherID {
  o String DataPublisherID
  o info PublisherInfo
}
```

   - The metadata of the data sets represented as **Data** object.

```
asset Data identified by DataID {
  o String DataID
  o metaData meta
  o String checksum
  o String cid
  --> DataPublisher publisher
}
```

2. Transactions
   - Publish the data sets process: **AddAsset()**, **PublishData()**

```
transaction PublishData {
  --> DataPublisher publisher
  --> Data data
}
```

   - Modify and Verify the data sets process: **ModifyData()** **VerifyData()**

```
transaction  ModifyData {
  --> Data data
  --> DataPublisher modifier
  o metaData modifiedMeta
  o String newCid
}
transaction VerifyData {
  --> Data data
  o String checksum
}
```

2. Transactions
   - Publish the data sets process: **AddAsset()**, **PublishData()**

```
transaction PublishData {
  --> DataPublisher publisher
  --> Data data
}
```

   - Modify and Verify the data sets process: **ModifyData() VerifyData()**

```
transaction  ModifyData {
  --> Data data
  --> DataPublisher modifier
  o metaData modifiedMeta
  o String newCid
}
transaction VerifyData {
 --> Data data
 o String checksum
}
```

2. Transactions
   - Publish the data sets process: **AddAsset()**, **PublishData()**

```
transaction PublishData {
  --> DataPublisher publisher
  --> Data data
}
```

   - Modify and Verify the data sets process: **ModifyData() VerifyData()**

```
transaction ModifyData {
  --> Data data
  --> DataPublisher modifier
  o metaData modifiedMeta
  o String newCid
}
transaction VerifyData {
  --> Data data
  o String checksum
}
```

2. Transactions
   - Publish the data sets process: **AddAsset()**, **PublishData()**

```
transaction PublishData {
  --> DataPublisher publisher
  --> Data data
}
```

   - Modify and Verify the data sets process: **ModifyData() VerifyData()**

```
transaction ModifyData {
  --> Data data
  --> DataPublisher modifier
  o metaData modifiedMeta
  o String newCid
}
transaction VerifyData {
 --> Data data
 o String checksum
}
```

2. Transactions
   - Publish the data sets process: **AddAsset()**, **PublishData()**

```
transaction PublishData {
  --> DataPublisher publisher
  --> Data data
}
```

   - Modify and Verify the data sets process: **ModifyData() VerifyData()**

```
transaction  ModifyData {
  --> Data data
  --> DataPublisher modifier
  o metaData modifiedMeta
  o String newCid
}
transaction VerifyData {
 --> Data data
 o String checksum
}
```

3. Access control
   - The data contributor permission.
   - The citizens permission.

```
rule PublishTransaction{
  description: "Only publisher have all permission on his/
      her data"
  participant(p): "org.com.opendata.DataPublisher"
  operation: ALL
  resource(r): "org.com.opendata.PublishData"
  condition: (p.getIdentifier()===r.publisher.
      getIdentifier()&&r.publisher.getIdentifier()===r.
      data.publisher.getIdentifier())
  action: ALLOW
}
```

3. Access control
   - The data contributor permission.
   - The citizens permission.

```
rule PublishTransaction{
  description: "Only publisher have all permission on his/
    her data"
  participant(p): "org.com.opendata.DataPublisher"
  operation: ALL
  resource(r): "org.com.opendata.PublishData"
  condition: (p.getIdentifier()===r.publisher.
    getIdentifier()&&r.publisher.getIdentifier()===r.
    data.publisher.getIdentifier())
  action: ALLOW
}
```

3. Access control
   - The data contributor permission.
   - The citizens permission.

```
rule PublishTransaction{
  description: "Only publisher have all permission on his/
      her data"
  participant(p): "org.com.opendata.DataPublisher"
  operation: ALL
  resource(r): "org.com.opendata.PublishData"
  condition: (p.getIdentifier()===r.publisher.
      getIdentifier()&&r.publisher.getIdentifier()===r.
      data.publisher.getIdentifier())
  action: ALLOW
}
```

3. Access control
   - The data contributor permission.
   - The citizens permission.

```
rule PublishTransaction{
  description: "Only publisher have all permission on his/
      her data"
  participant(p): "org.com.opendata.DataPublisher"
  operation: ALL
  resource(r): "org.com.opendata.PublishData"
  condition: (p.getIdentifier()===r.publisher.
      getIdentifier()&&r.publisher.getIdentifier()===r.
      data.publisher.getIdentifier())
  action: ALLOW
}
```

4. Endorsement Policy

```
{
    "identities": [
        { "role": { "name": "member", "mspId": "Org1MSP" } },
        { "role": { "name": "member", "mspId": "Org2MSP" } },
        { "role": { "name": "member", "mspId": "Org3MSP" } }
    ],
    "policy": { "2-of": [
            { "signed-by": 0 },
            { "signed-by": 1 },
            { "signed-by": 2 }
        ]}
}
```

Open data portal

Server-side implementation:

- Written by NodeJS.
- Using Hyperledger Composer to interact with Hyperledger Fabric blockchain network.
- Connecting to IPFS network.

Client-side implementation:

- Written by VueJS.

Open data portal

Server-side implementation:

- Written by NodeJS.
- Using Hyperledger Composer to interact with Hyperledger Fabric blockchain network.
- Connecting to IPFS network.

Client-side implementation:

- Written by VueJS.

Open data portal

Server-side implementation:

- Written by NodeJS.
- Using Hyperledger Composer to interact with Hyperledger Fabric blockchain network.
- Connecting to IPFS network.

Client-side implementation:

- Written by VueJS.

Open data portal

Server-side implementation:

- Written by NodeJS.
- Using Hyperledger Composer to interact with Hyperledger Fabric blockchain network.
- Connecting to IPFS network.

Client-side implementation:

- Written by VueJS.

Open data portal

Server-side implementation:

- Written by NodeJS.
- Using Hyperledger Composer to interact with Hyperledger Fabric blockchain network.
- Connecting to IPFS network.

Client-side implementation:

- Written by VueJS.

Open data portal

Server-side implementation:

- Written by NodeJS.
- Using Hyperledger Composer to interact with Hyperledger Fabric blockchain network.
- Connecting to IPFS network.

Client-side implementation:

- Written by VueJS.

Contents

Testing

Hyperledger Caliper[2] is a blockchain performance benchmark framework.

[2] `https://github.com/hyperledger.caliper`

Testing

Hyperledger Caliper[2] is a blockchain performance benchmark framework.

Table: Evaluated result of blockchain network performance
Number of transactions: 100, transactions rate: 100 tps

Transaction type	**Send rate (tps)**	**Latency (s)**			**Throughput (tps)**
		Min	**Max**	**Avg**	
PublishData	96.4	6.45	16.08	11.73	6.1
ModifyData	101.0	6.60	15.67	11.27	6.3
DownloadData	100.6	6.66	16.09	11.06	6.1

[2] `https://github.com/hyperledger.caliper`

Contents

Conclusions

+ The combination of the Hyperledger Fabric and the IPFS technology enhances the transparency, availability and integrity of the open data.
+ The limitation:
  - storing the private key of the publisher.
+ Future works:
  - integrating hardware security modules,
  - evaluating the performance of the IPFS network.

Conclusions

+ The combination of the Hyperledger Fabric and the IPFS technology enhances the transparency, availability and integrity of the open data.
+ The limitation:
  - storing the private key of the publisher.
+ Future works:
  - integrating hardware security modules,
  - evaluating the performance of the IPFS network.

Conclusions

+ The combination of the Hyperledger Fabric and the IPFS technology enhances the transparency, availability and integrity of the open data.
+ The limitation:
  - storing the private key of the publisher.
+ Future works:
  - integrating hardware security modules,
  - evaluating the performance of the IPFS network.

Conclusions

+ The combination of the Hyperledger Fabric and the IPFS technology enhances the transparency, availability and integrity of the open data.
+ The limitation:
  - storing the private key of the publisher.
+ Future works:
  - integrating hardware security modules;
  - evaluatating the performance of the IPFS network.

Conclusions

+ The combination of the Hyperledger Fabric and the IPFS technology enhances the transparency, availability and integrity of the open data.
+ The limitation:
  - storing the private key of the publisher.
+ Future works:
  - integrating hardware security modules;
  - evaluatating the performance of the IPFS network.

Conclusions

+ The combination of the Hyperledger Fabric and the IPFS technology enhances the transparency, availability and integrity of the open data.
+ The limitation:
  - storing the private key of the publisher.
+ Future works:
  - integrating hardware security modules;
  - evaluatating the performance of the IPFS network.

Q&A