The Blockchain as a Software Connector

Sherry Xu, Cesare Pautasso
Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran, Shiping Chen

Data61@CSIRO | University of Lugano (USI)
www.csiro.au design.inf.usi.ch

Università della Svizzera italiana
Faculty of Informatics

CSIRO

Contribution

- Characterizing blockchain from software architecture perspective
- Rationales to support the architectural decision on whether to employ a blockchain as opposed to other software connectors
- Implications of using the blockchain as a software connector
  - Design trade-offs regarding quality attributes
  - Experience harvested from real-world projects

Blockchain

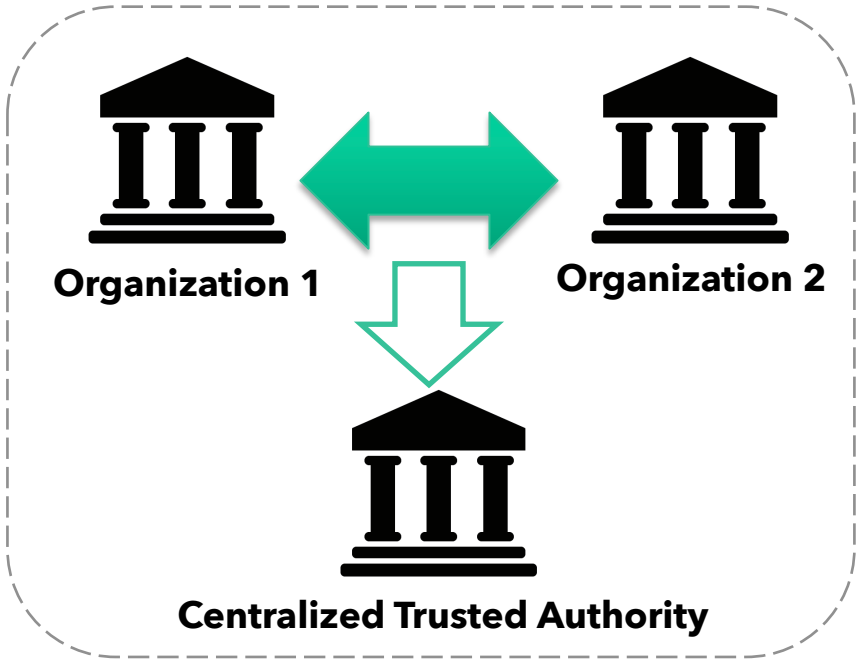
Background

Blockchain 1.0 – Cryptocurrency

Bitcoin Network

A. Antonopoulos, Mastering Bitcoin-unlocking Digital crytocurrencies, 2014

Benefits of Using Blockchain



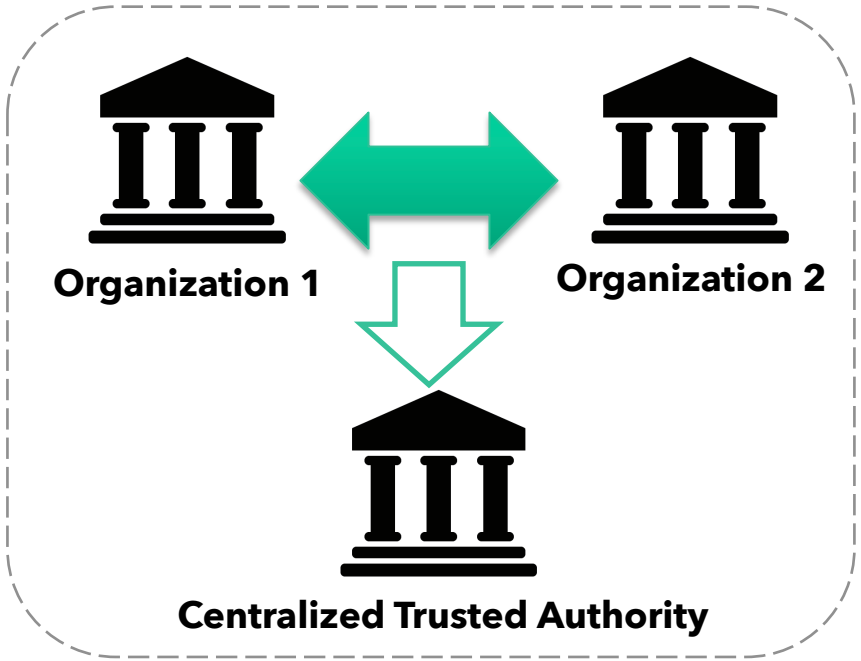
Traditional trusted environment

Trusted market

Trusted payment

Trusted authentication

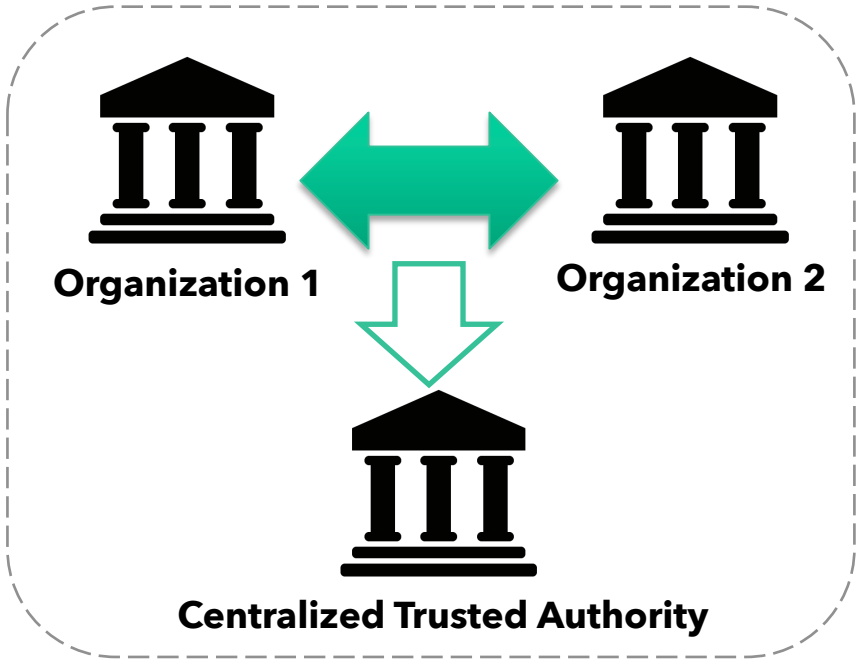
Benefits of Using Blockchain



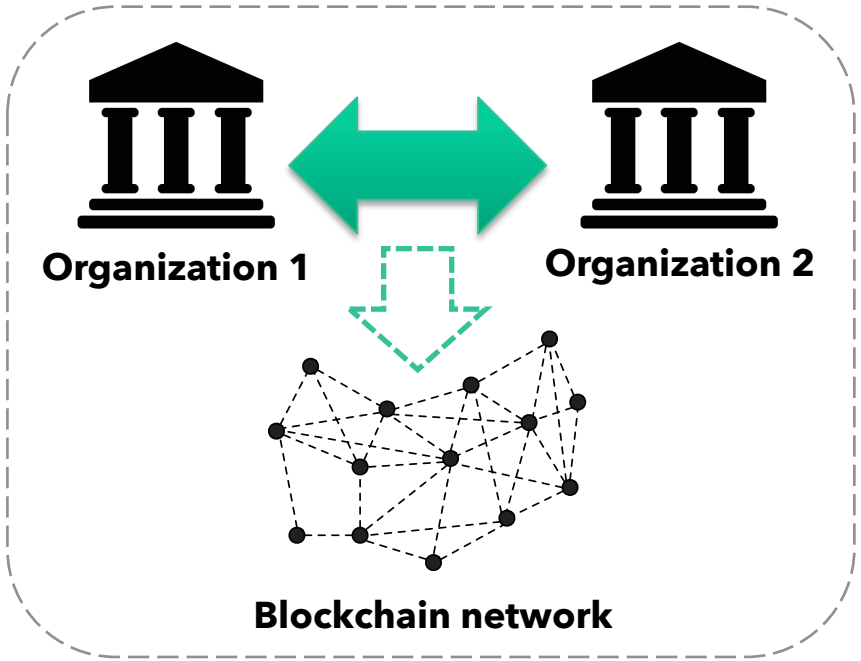
Traditional trusted environment

- Centralization
  - Single point failure
- Access control across systems
  - System internal status is opaque
- Collaboration/interoperability
  - Fragmented internal systems centralized in their own way
  - Costly to interoperate and collaborate

Benefits of Using Blockchain



Traditional trusted environment

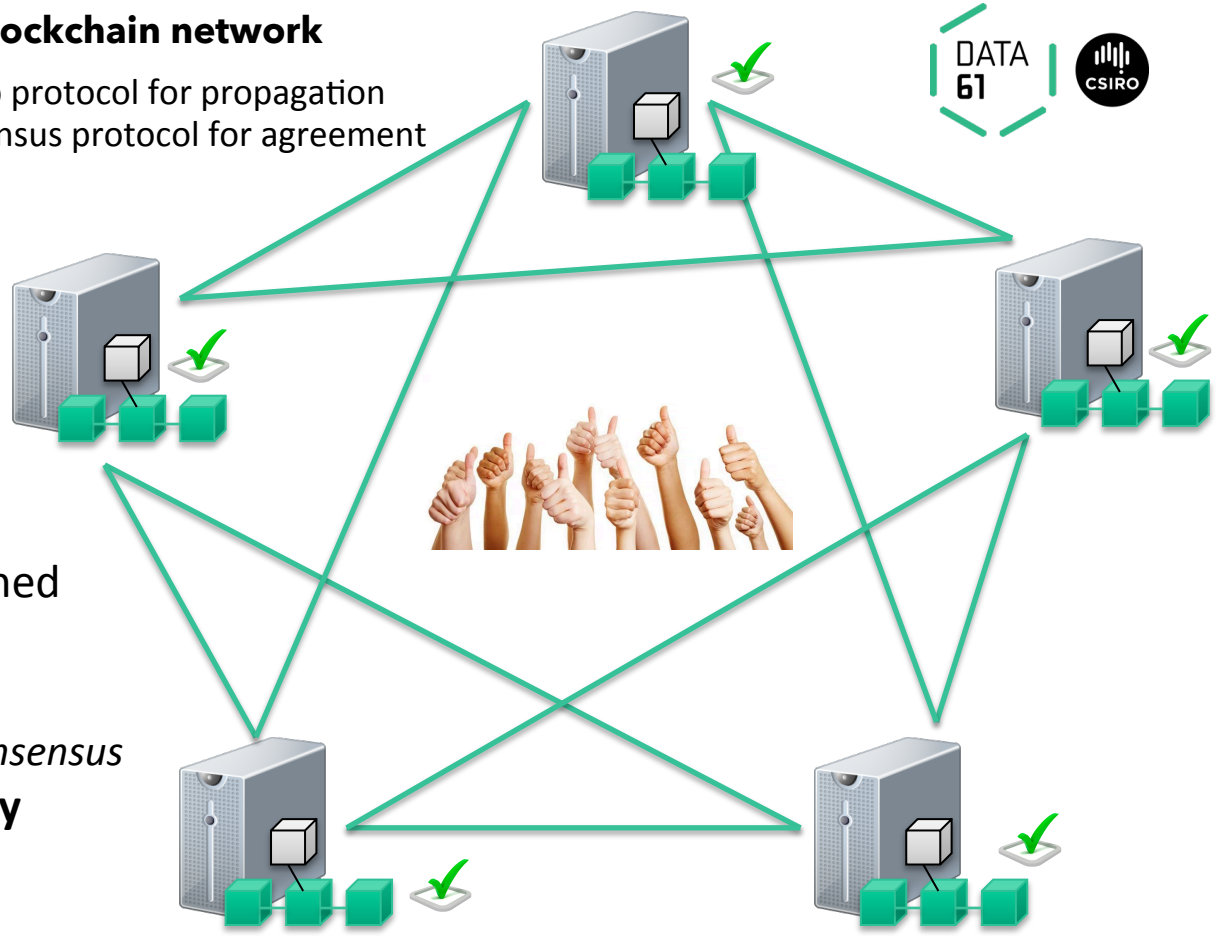


Blockchain trustless environment

Why?

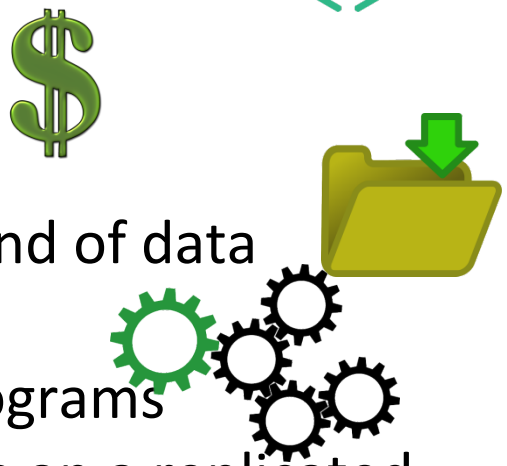
Blockchain network

- Gossip protocol for propagation
- Consensus protocol for agreement



- **Immutable public ledger**
  - Audit trail of what happened
- **Every node hosts a replica**
  - Distributed consensus
    - *No central owner of consensus*
- **Transaction is verified by every node**

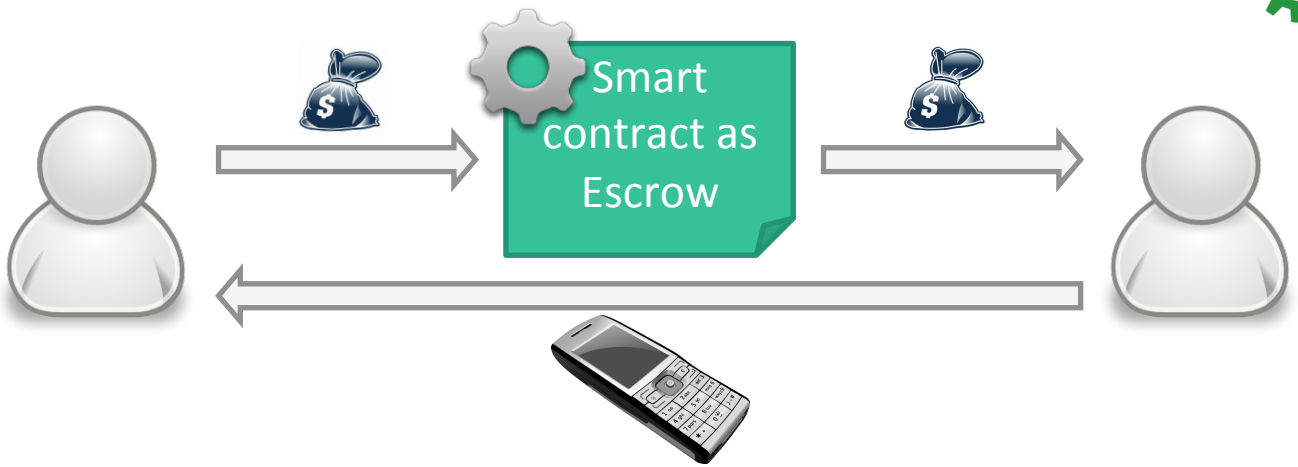
Blockchain Evolution

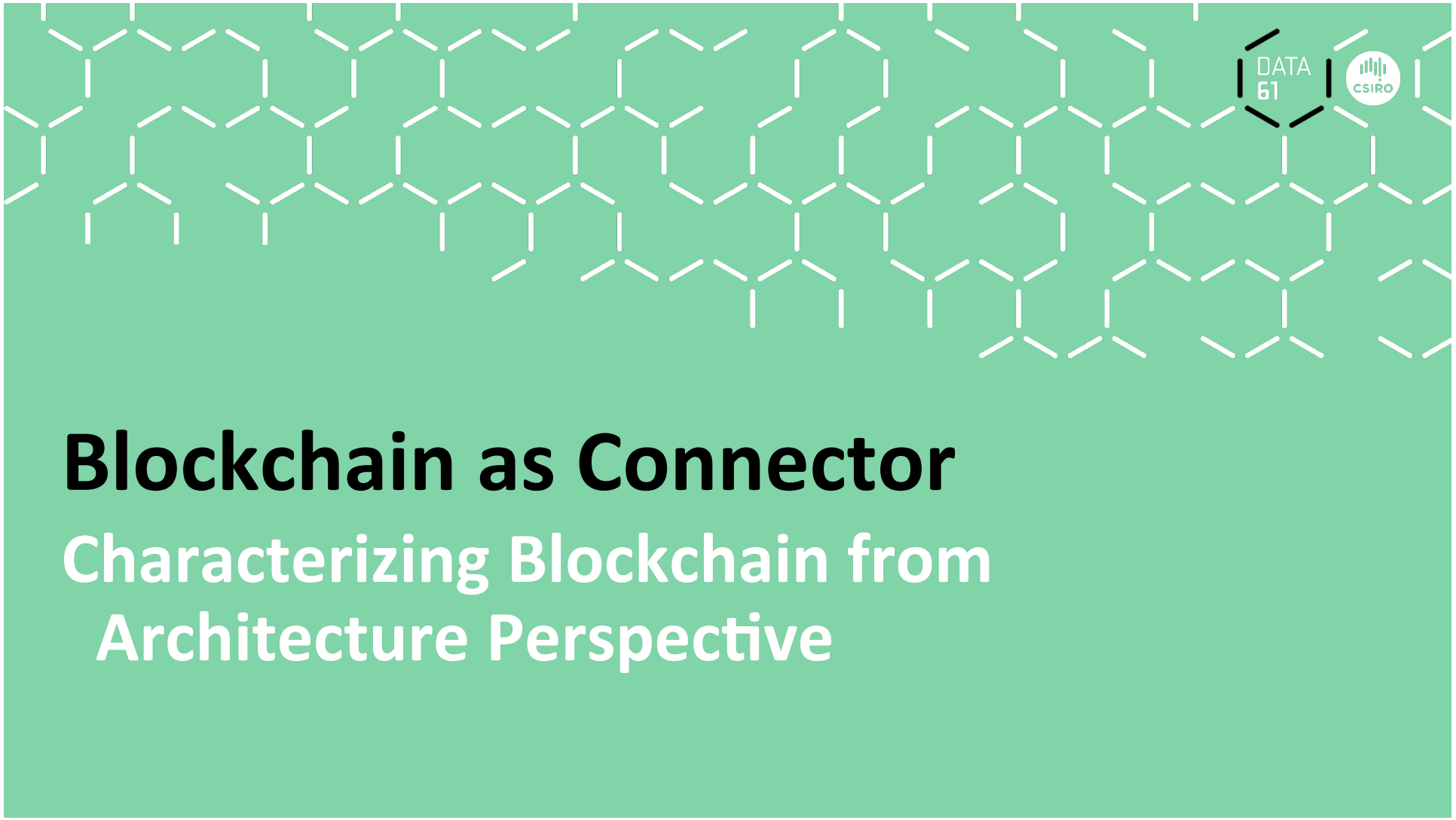


- Blockchain 1.0
  - Bitcoin transactions are financial transfers
  - Blockchain ledger can store/transact any kind of data
- Blockchain 2.0 – "Smart contract"
  - Global computational infrastructure for programs
  - Event-driven program (with state) that runs on a replicated, shared ledger
  - Can enact decisions on complex business conditions
    - Coordination with business processes through APIs
  - Can hold and transfer assets held by the contract itself

Blockchain Evolution

- Blockchain 1.0
  - Bitcoin transactions are financial transfers
  - Blockchain ledger can store/transact any kind of data
- Blockchain 2.0 – “Smart contract”





Blockchain as Connector

Characterizing Blockchain from Architecture Perspective

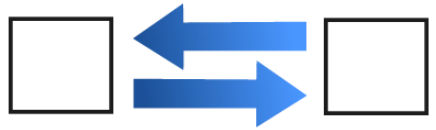
Software Connectors

- Building blocks of software component interaction
  - Performance
  - Reliability
  - Security
- Services
  - **Communication**: transfer data
  - **Coordination**: transfer control
  - **Facilitation**: enable and optimise component's interactions
  - Conversion: adjust the interactions between incompatible interfaces

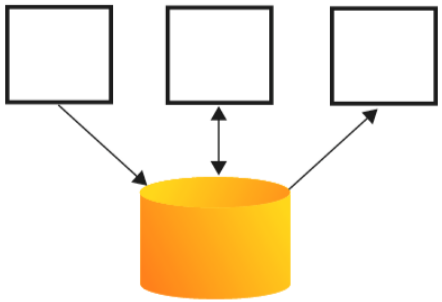
Example Software Connectors

File Transfer

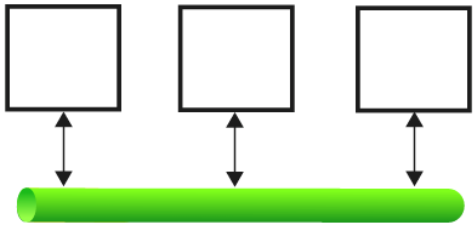
Stream



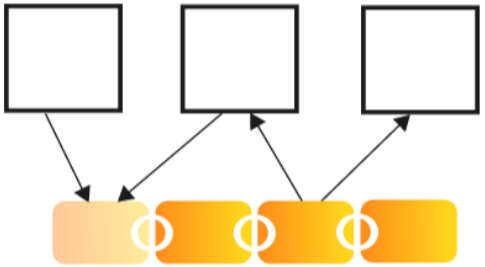
Remote Procedure Call



Shared Database

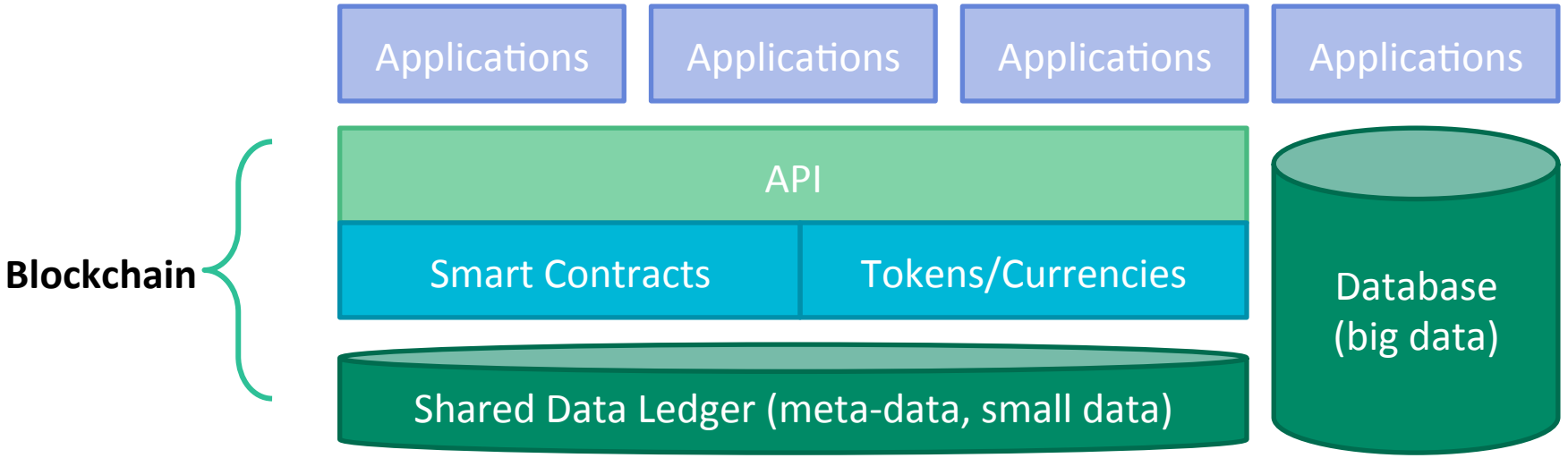


Message Bus

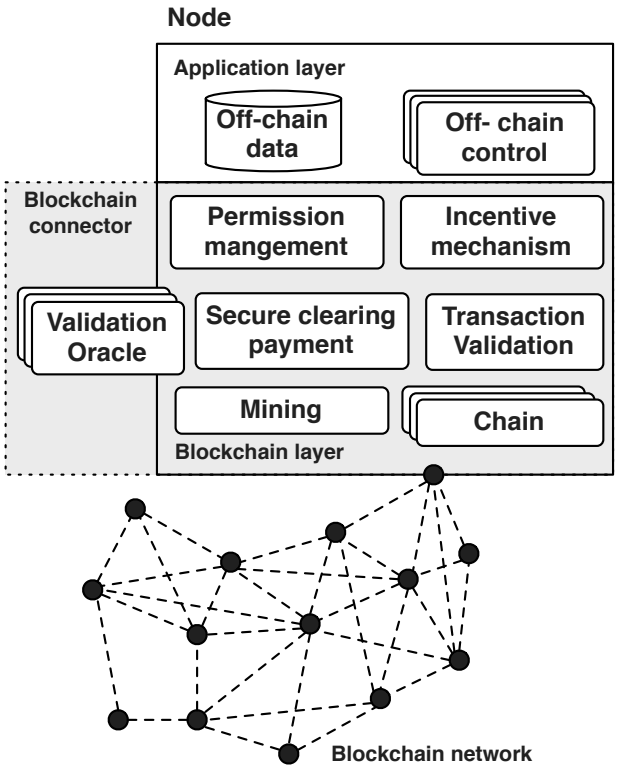


Blockchain

Blockchain used in a Web application

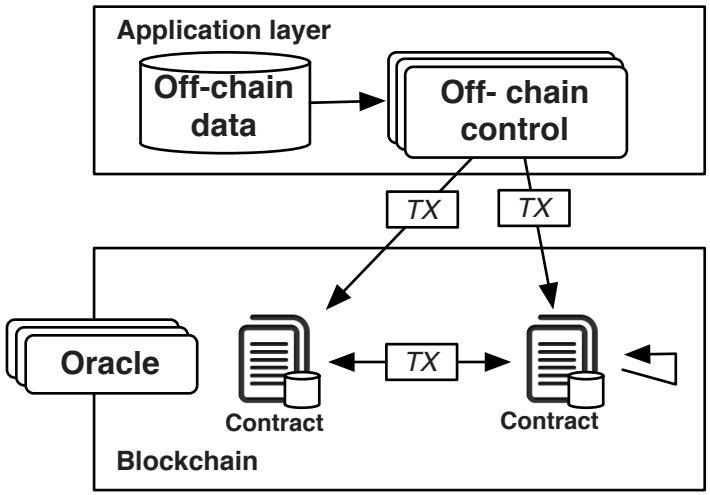


Blockchain as a Software Connector



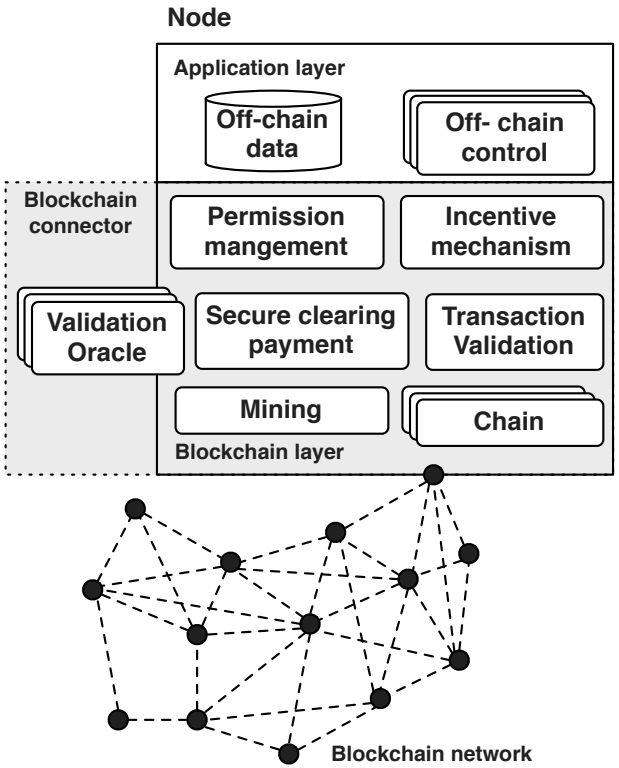
- Communication
- Coordination
- Facilitation

Blockchain as a Software Connector



- Communication
  - Arbitrary data within transaction
  - Contract storage
- Coordination
  - Transactions
    - From external owners
    - From contract accounts
    - Call functions defined in contracts
    - Create new contracts
  - Oracle
- Facilitation

Blockchain as a Software Connector



- Communication
- Coordination
- Facilitation
  - Transaction validation
  - Mining mechanism
  - Secure clearing payment
  - Incentive mechanism
  - Permission management

Blockchain Limitations

- Limited scalability of public blockchain
  - The public blockchain processes 3-20 transactions per second
    - VISA handles around 2000 transactions per second
  - Improving transaction processing rate
    - Larger blockchain size
    - Off-chain transactions
    - Smaller transaction
      - Remove signature
- Privacy of public blockchain
  - Encryption

Blockchain configuration decision

Placement: on-chain vs. off-chain

Enable verification of computational result, limited computation power and data storage, publicly available

More computation power and data storage, less cost, additional trust required, integrate with existing systems

Oracle placement: Internal vs. External

Inject external state into the blockchain, increase latency

Introduce a trusted third party

Pe

W

at

Be

Si

Heterogeneous dataset

Information isolation, harder chain and permission management

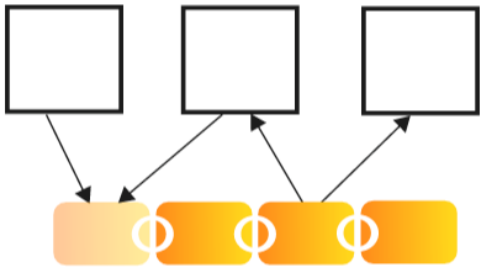
Public chain vs. Private chain

Information transparency, scalability, trustworthy

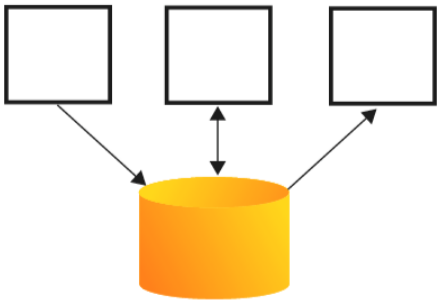
Information isolation, easier asset-specific auditablility

Calculated from Ethereum
Store 1kb data costs around $0.32
read 1 kb data costs $0.015

Blockchain vs. Shared DB: **Operations**

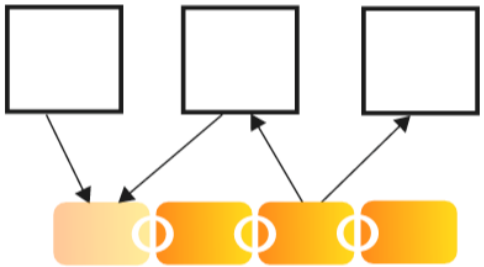


- Insert Transaction (Append Only)

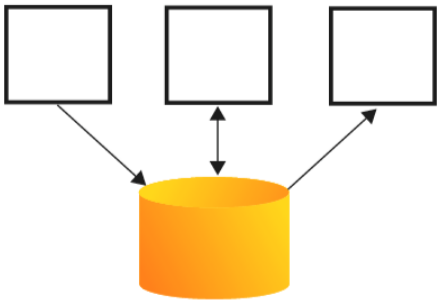


- Create
- Read
- Update
- Delete

Blockchain vs. Shared DB: **Replication**

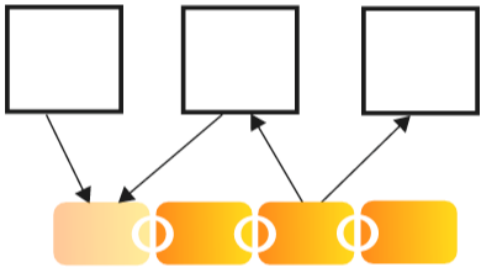


- Full Replication on every peer

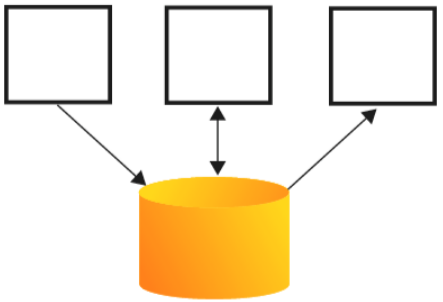


- Master-Slave
- Multi-master

Blockchain vs. Shared DB: **Consensus**

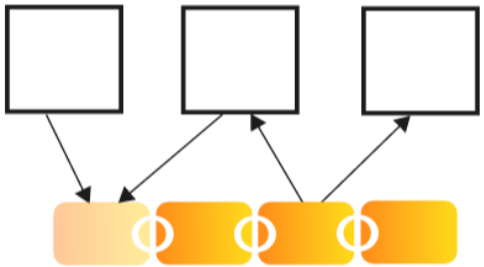


- Majority of peers agree on the outcome of transactions
- Tolerant of Byzantine Generals'problem

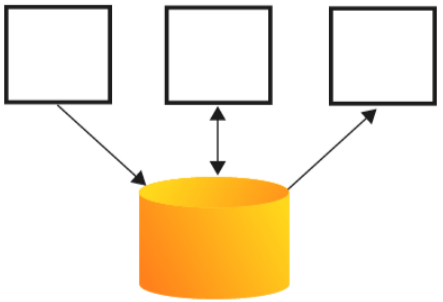


- Distributed Transactions (2 Phase Commit, Paxos)
- Synchronization

Blockchain vs. Shared DB: **Invariants**



- Transactions validated everywhere
- Global rules enforced on the whole blockchain
  - No extra money created during a spending transaction

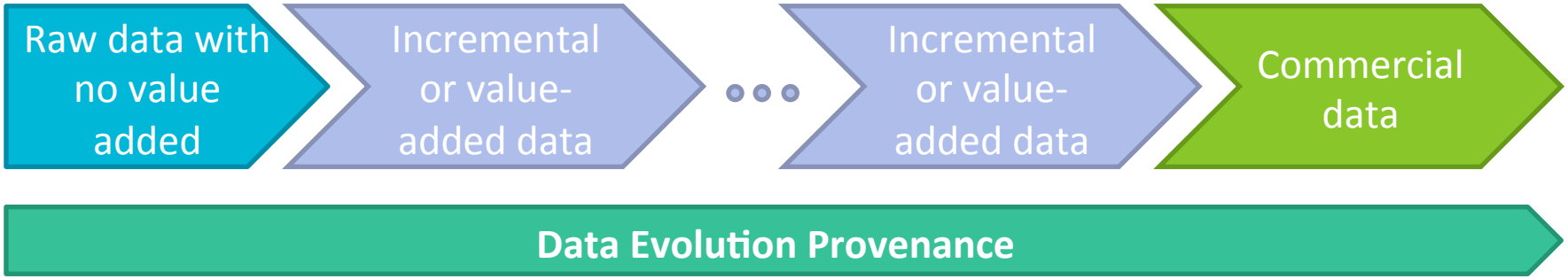


- Integrity Constraints

Project Retrospective

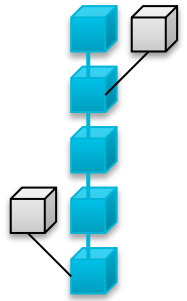
Open Data Registry

Data Prosumption Chain

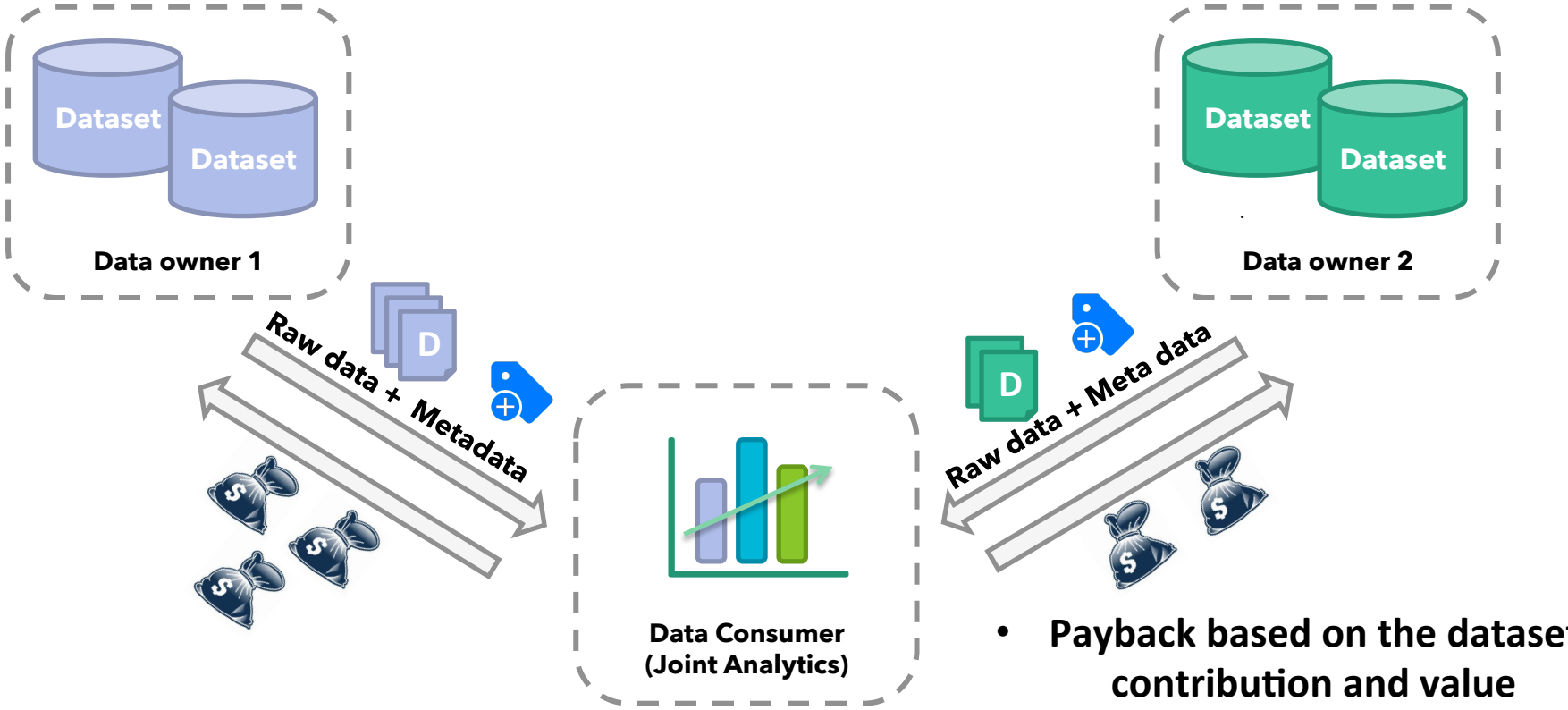


- Provenance
  - How dataset evolves from raw data to value-added data
    - Raw Data
      - Government open data, individual device data
      - Priced at zero, or at marginal cost
    - Value-added data
      - Private weather services
  - Who, when, what ,how (**Metadata**)

Blockchain

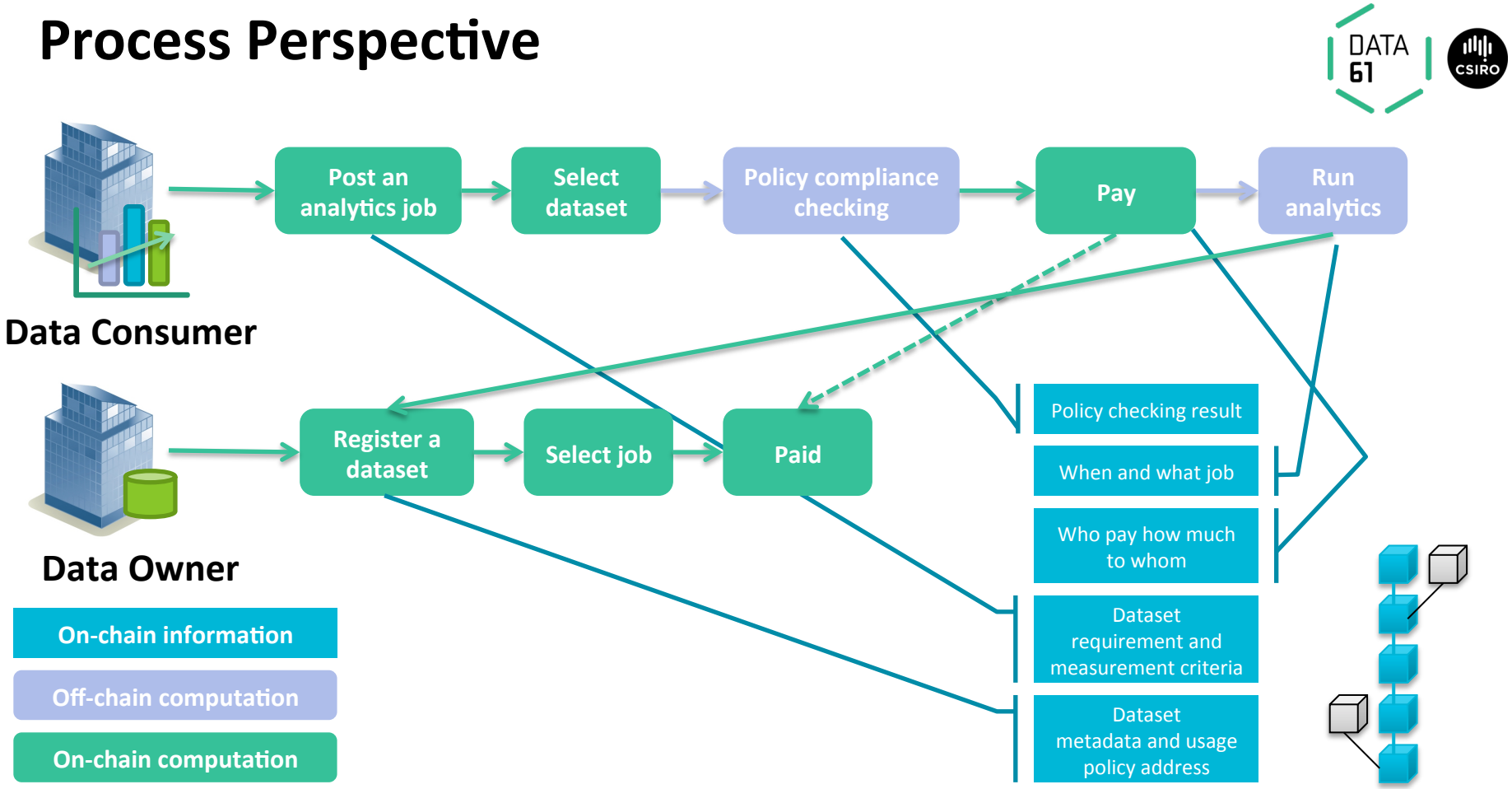


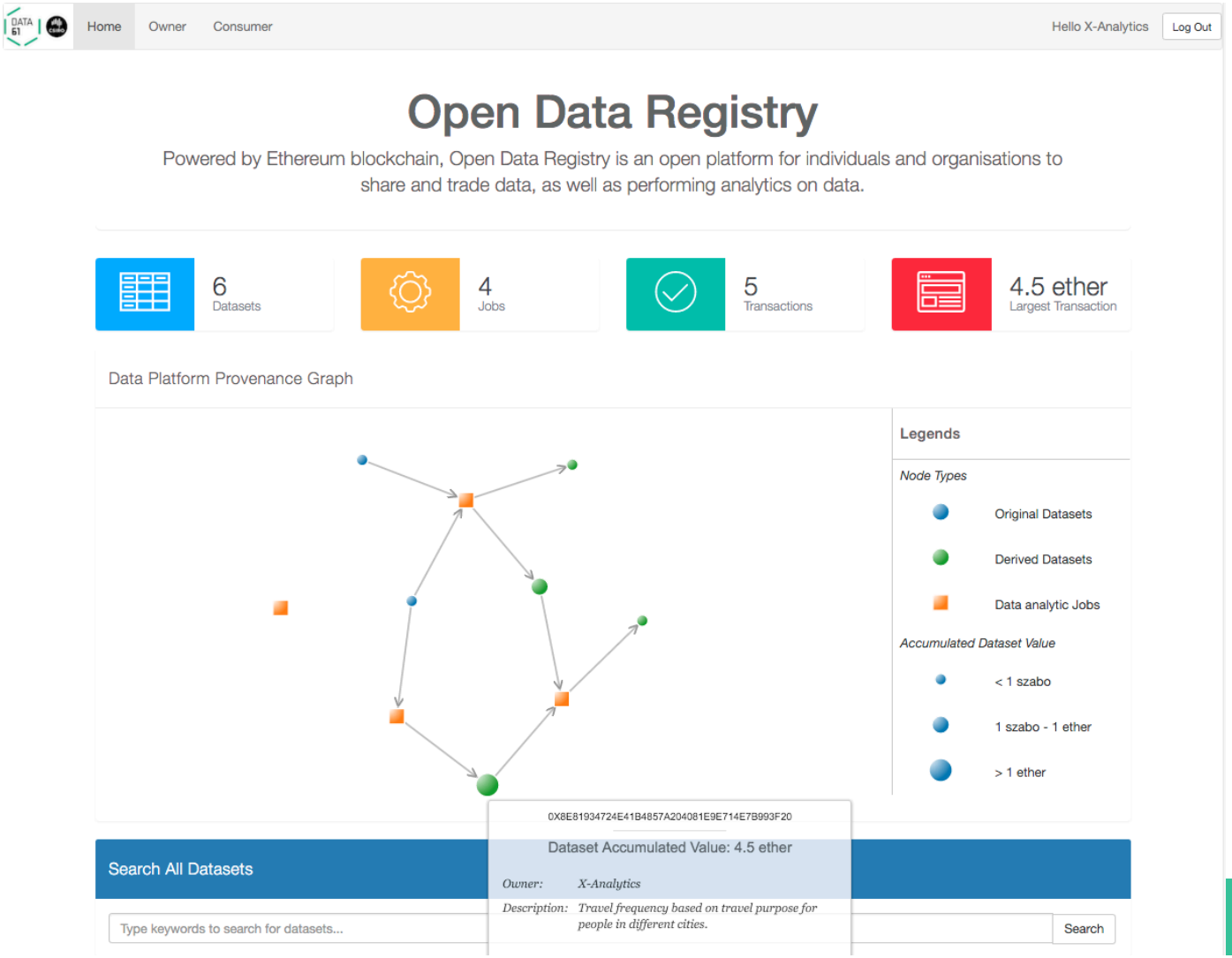
Data Monetization



- Payback based on the dataset contribution and value

Process Perspective





Home page of Open Data Registry

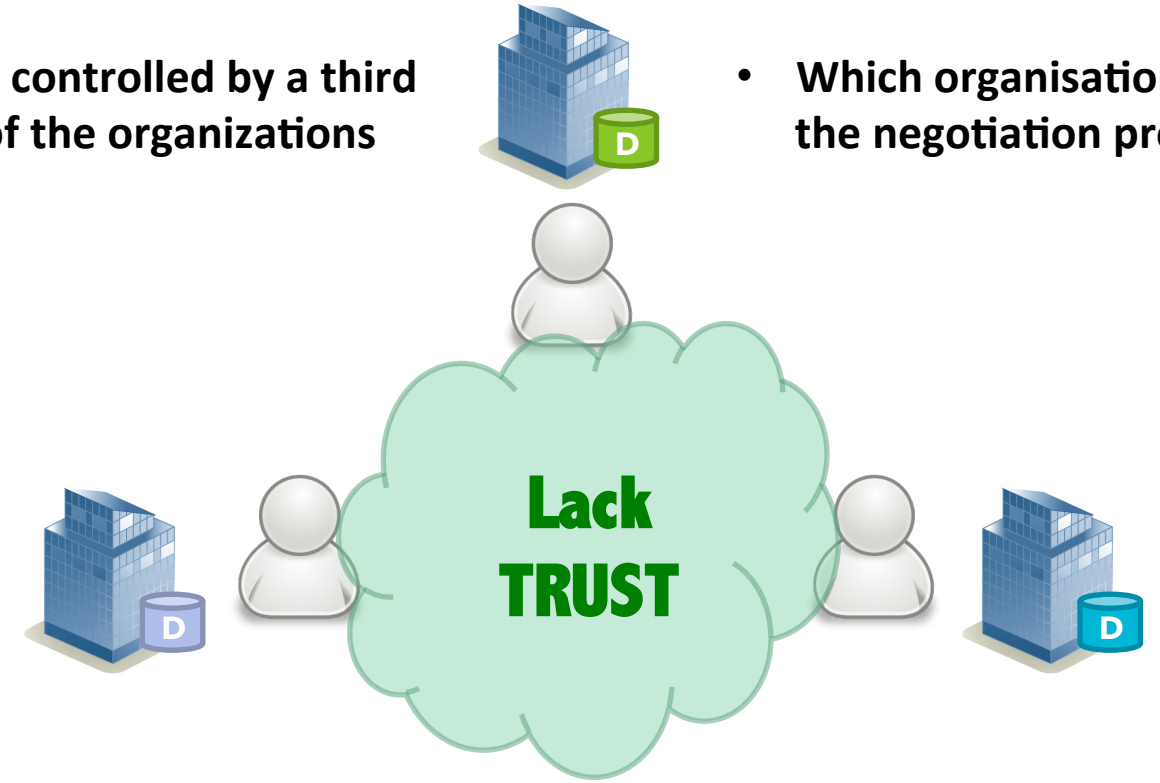
- Statistics
- Provenance

Project Retrospective

Secure Contract Negotiation

Scenario

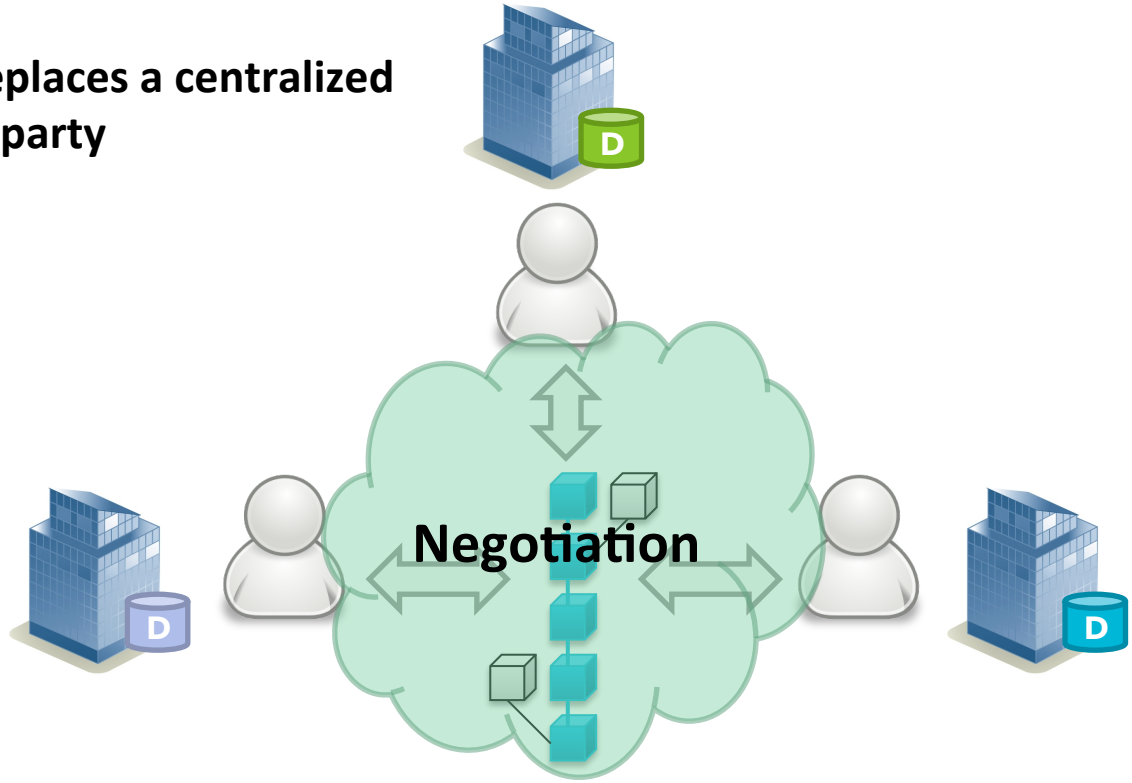
- Negotiation is controlled by a third party or one of the organizations get involved.



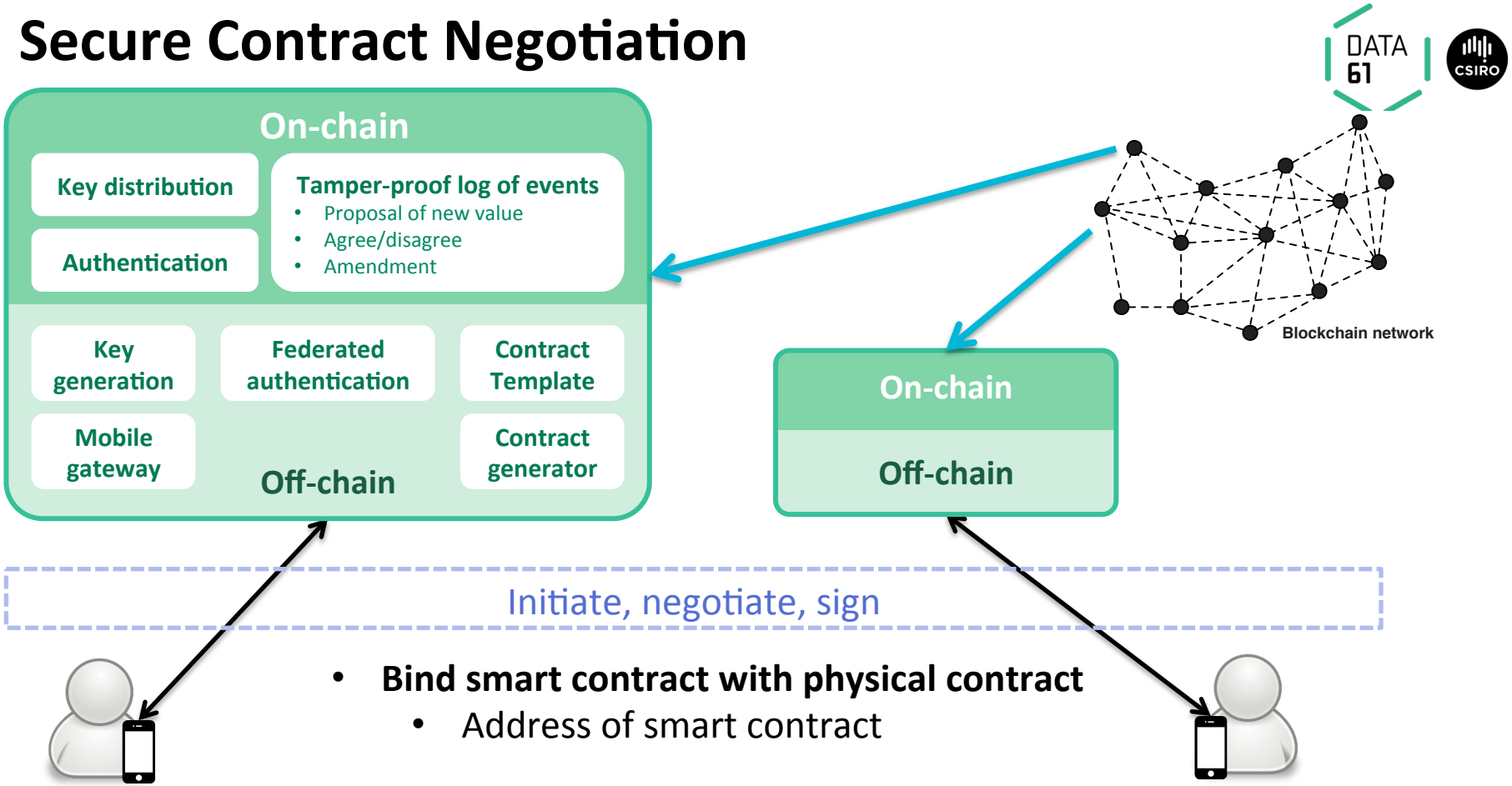
- Which organisation should control the negotiation process?

Secure Contract Negotiation

- **Blockchain replaces a centralized trusted third party**



Secure Contract Negotiation



- **Bind smart contract with physical contract**
  - Address of smart contract

Conclusion

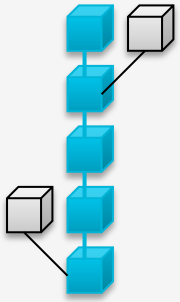
- Integration projects sometime struggle to find a central party trusted by all participants
- The blockchain offers a trusted shared transaction log built on top of an untrusted and decentralized network of peers.
- Software components may read the transaction history (immutable) and add transactions to extend the blockchain
- Given its fully replicated nature, the blockchain has some limitations (performance, data size)
- We have applied the blockchain as a software connector in several integration projects (open data registry, legal contract negotiation, smart meters)


DATA
61
Thank you
WICSA2016
www.csiro.au
CSIRO

Backup

Blockchain

- A public/community ledger
  - No central owner of consensus
  - Transaction verification
- Payment
  - Conditional payment
  - Micropayment channel



Metadata

- Quantify the value of the dataset
- Consumer defined criteria
  - For example, data size, data coverage

Usage policy compliance

- Focus on ETL (extract, transform, load) phase
- Assumption 1: ETL is before every data pipe
- Assumption 2: SQL/Hive/SparkSQL is used to do ETL
- Limitation: tabular data only