Transactional Properties of Permissioned Blockchains

Ghareeb Falazi[1], Vikas Khinchi[2], Uwe Breitenbücher[1], Frank Leymann[1]

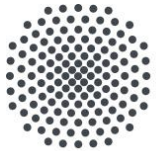
Michael Hahn (Presenter)

[1]{lastname}@iaas.uni-stuttgart.de

Institute of Architecture of Application Systems, University of Stuttgart

[2]vikas.khinchi@diconium.com

Diconium Digital Solutions GmbH*, Stuttgart*

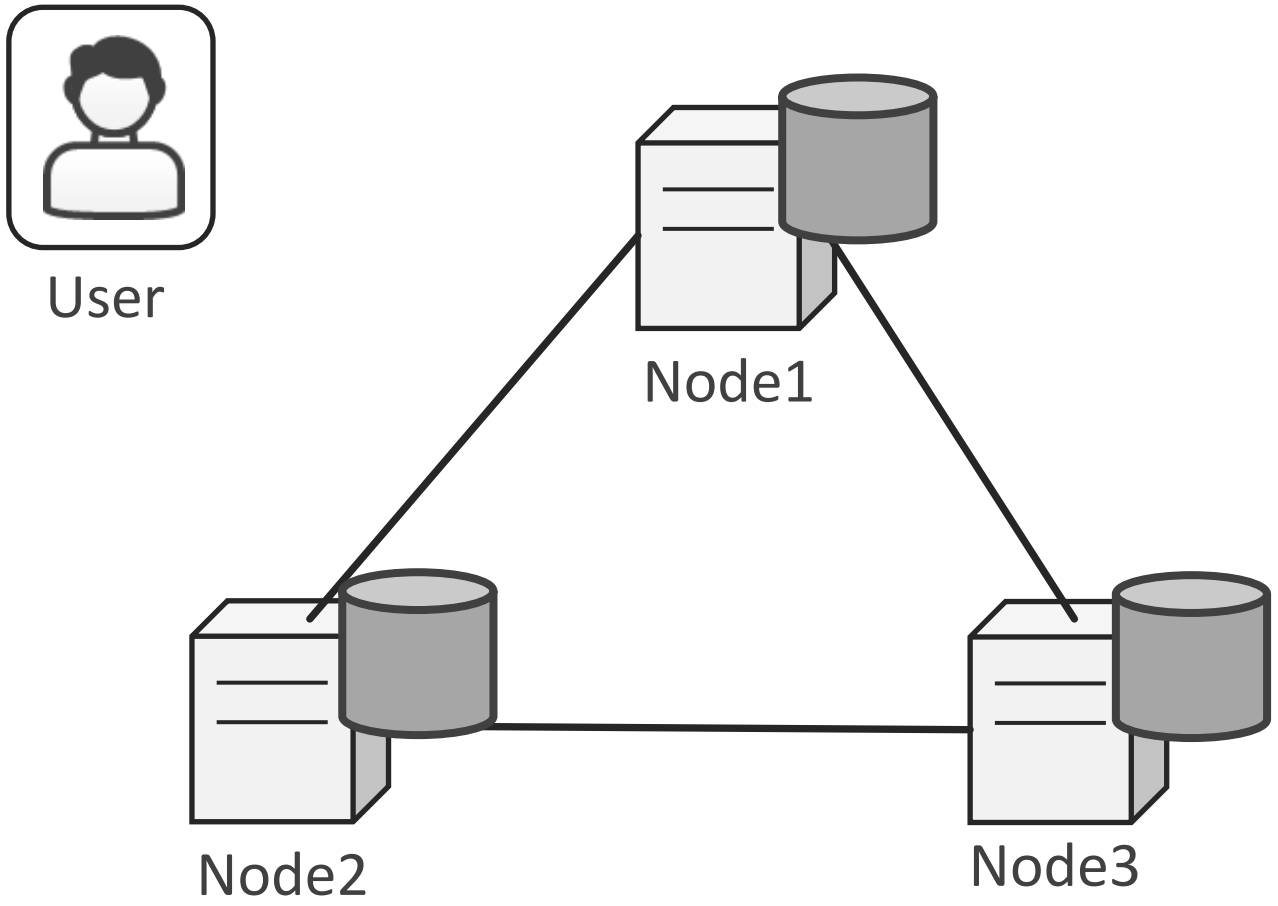


University of Stuttgart

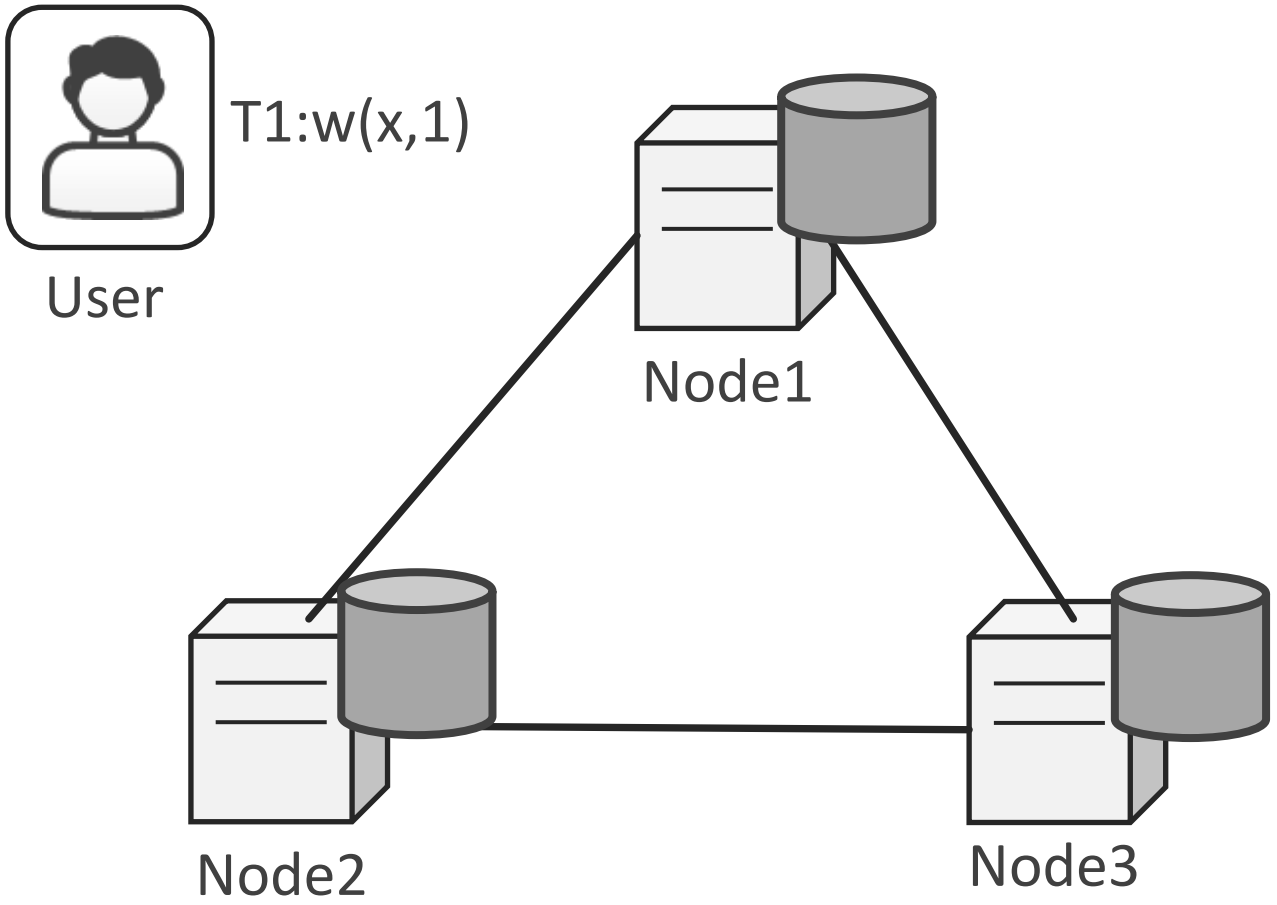
Agenda

- Background and Motivation
- Method
  - Identifying Suitable Criteria
  - Analyzing Existing Systems
  - Determining Properties and Categorizing Systems
- Conclusion

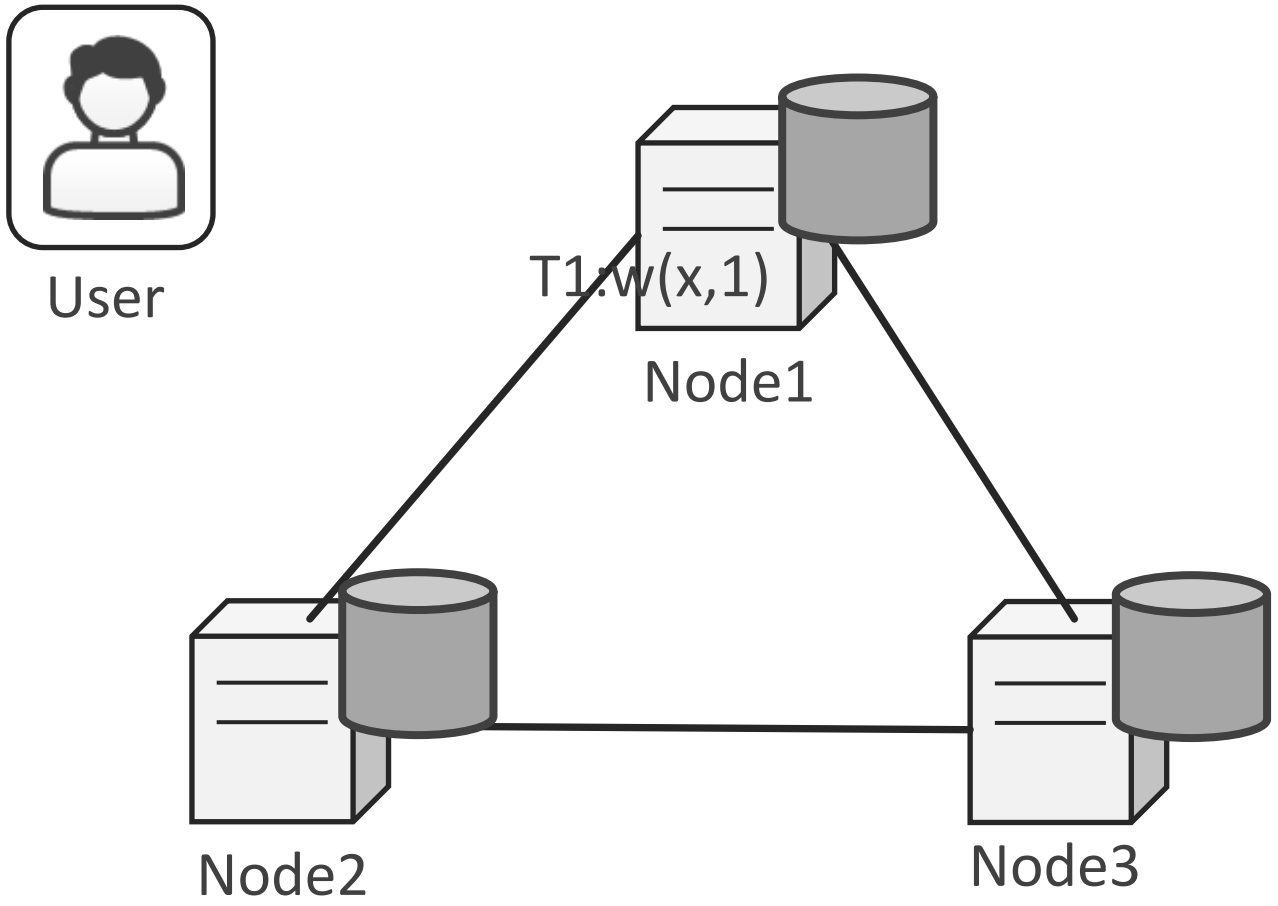
Background: Replicated Database Systems



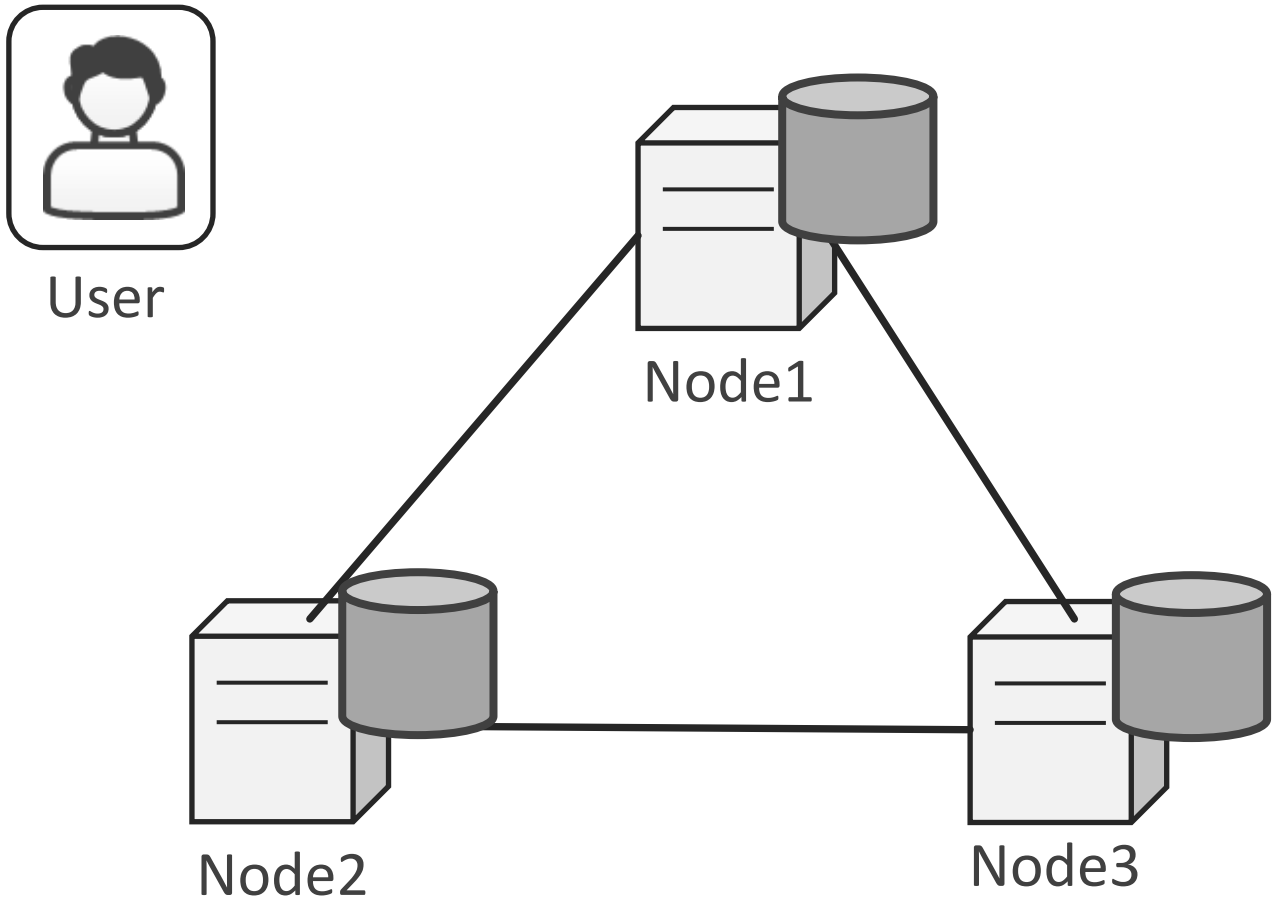
Background: Replicated Database Systems



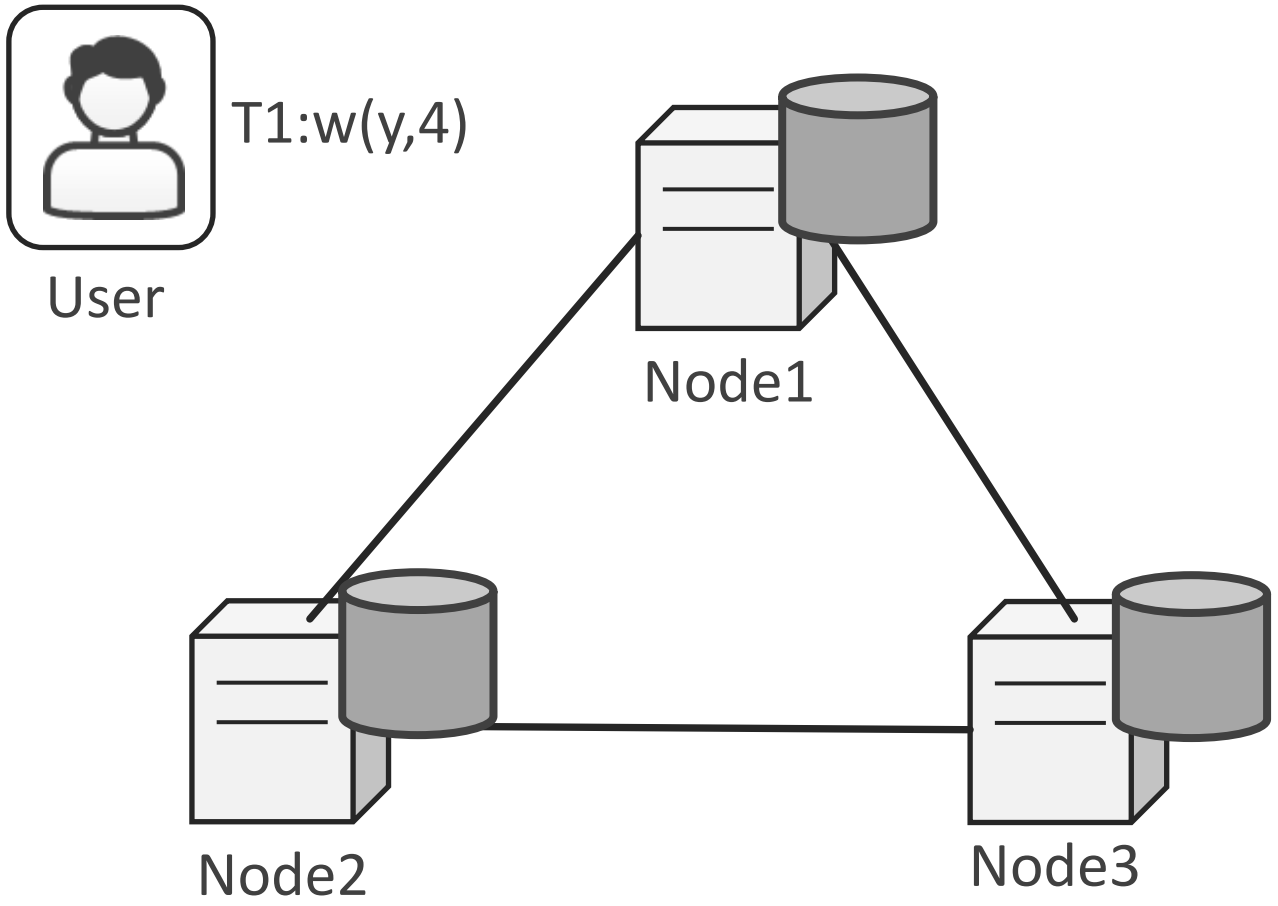
Background: Replicated Database Systems



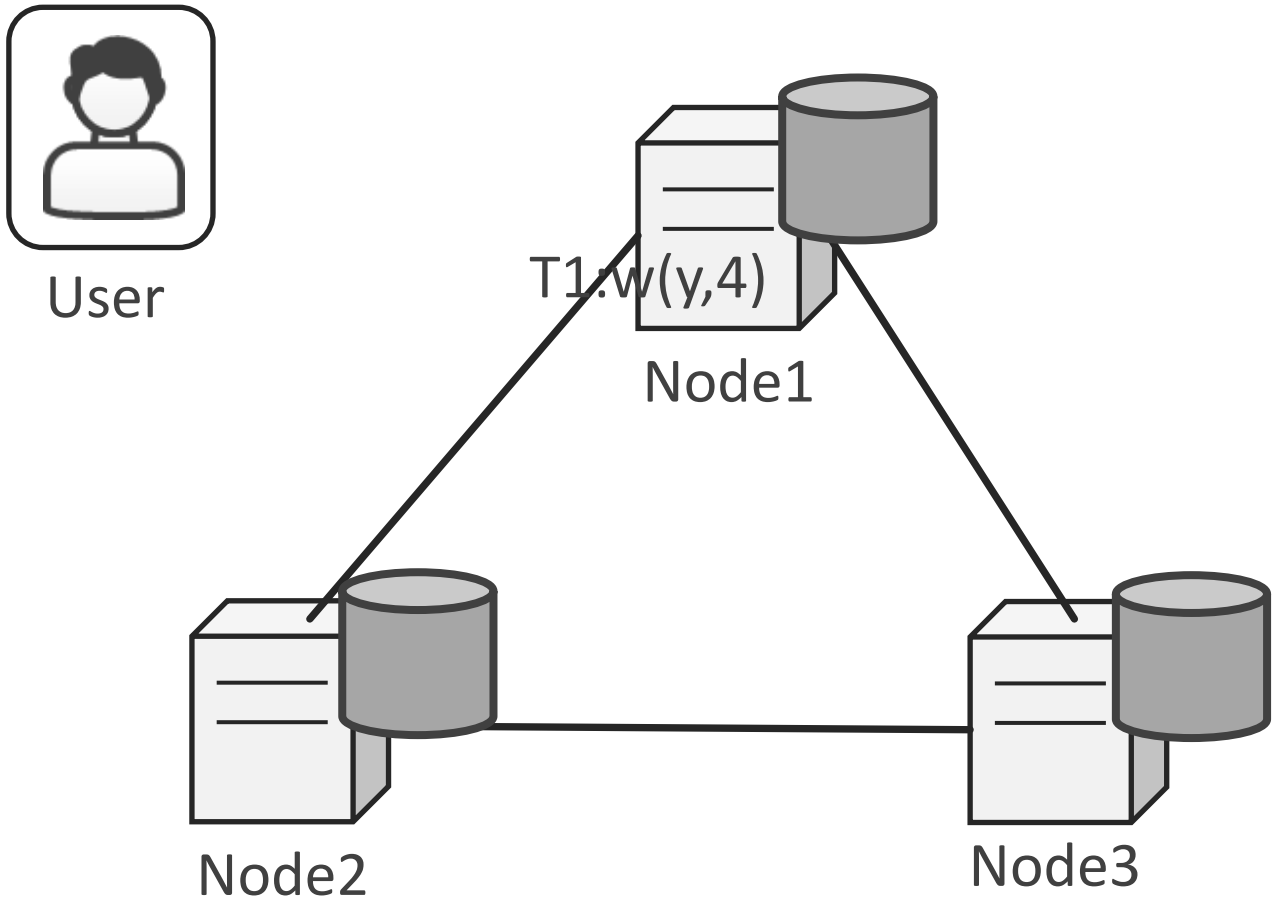
Background: Replicated Database Systems



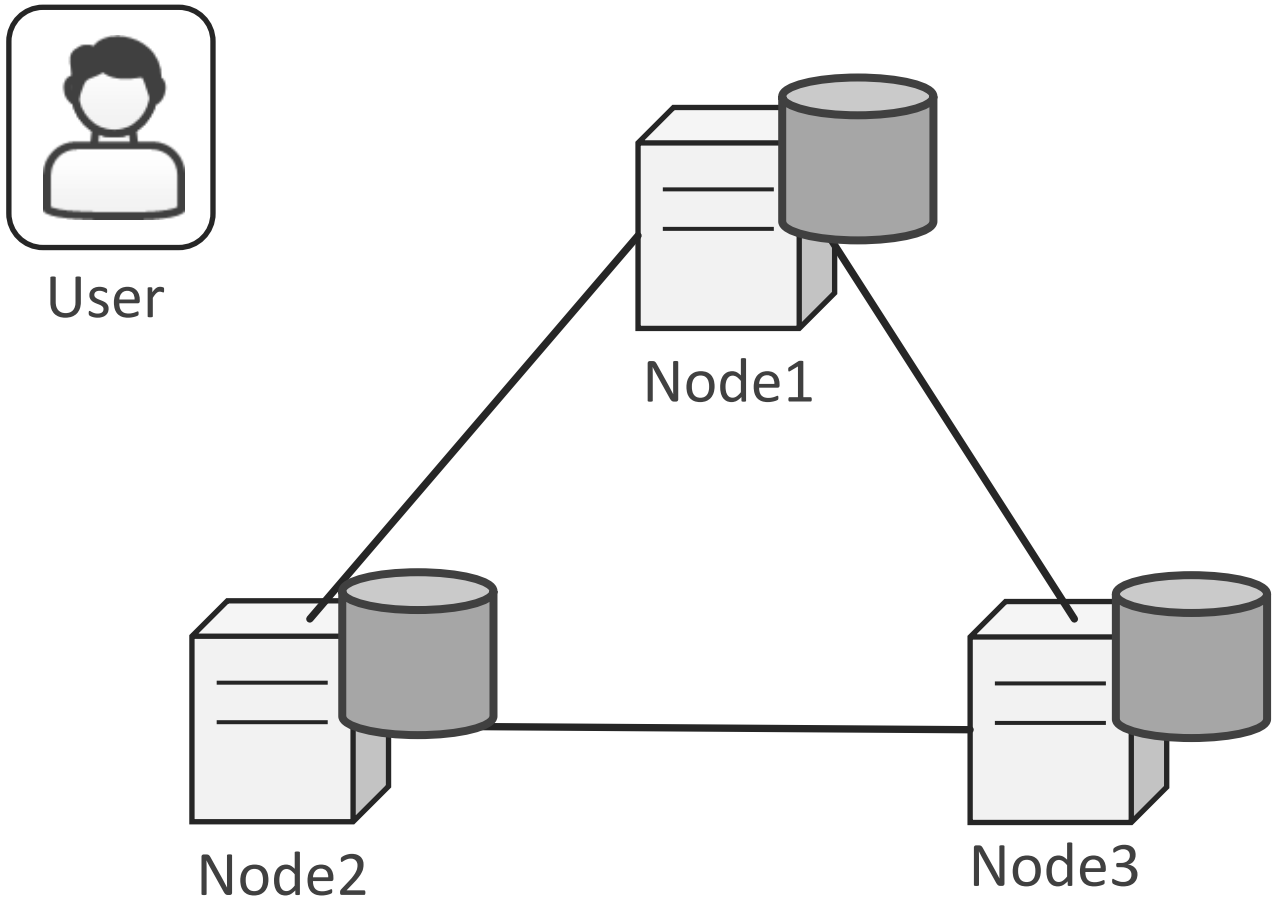
Background: Replicated Database Systems



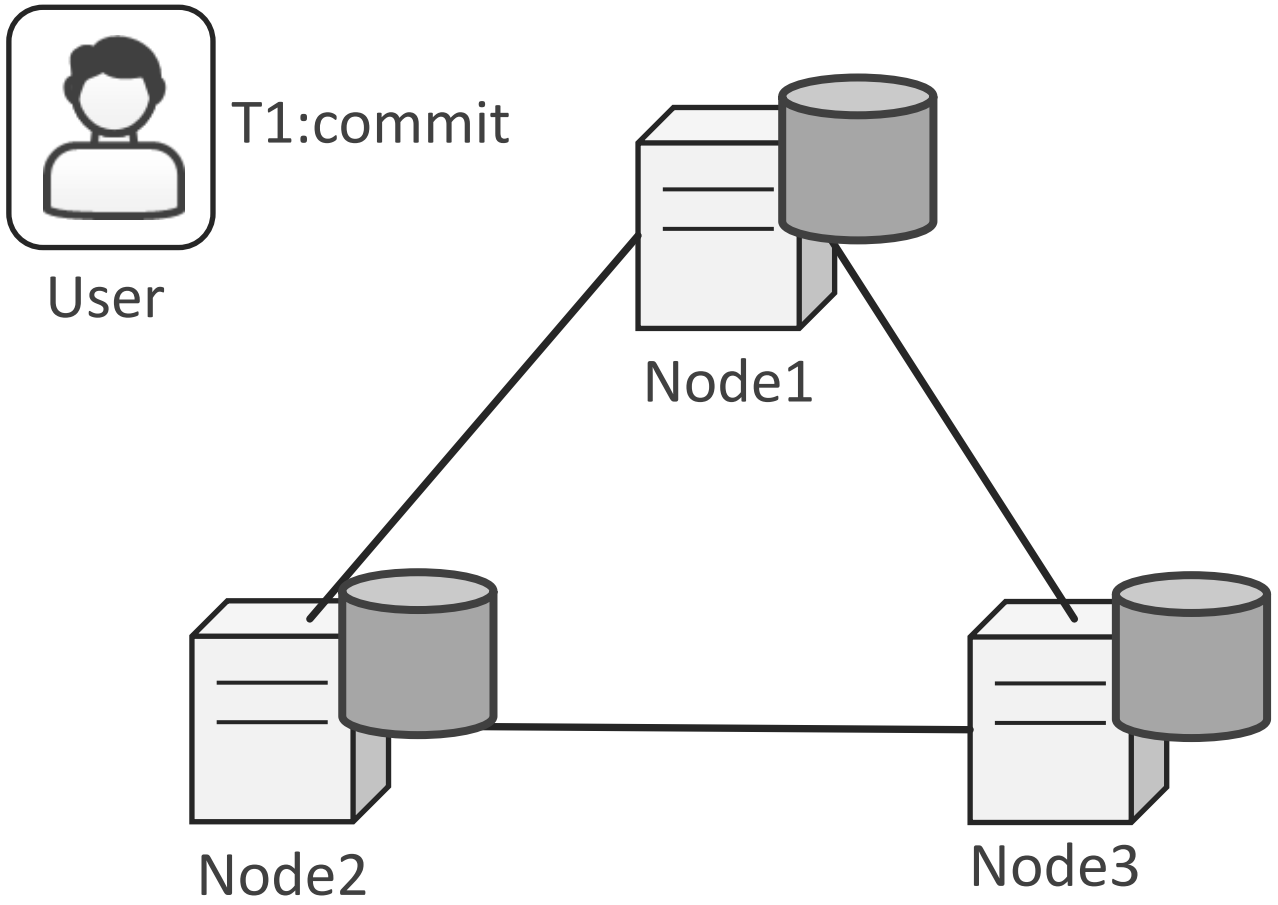
Background: Replicated Database Systems



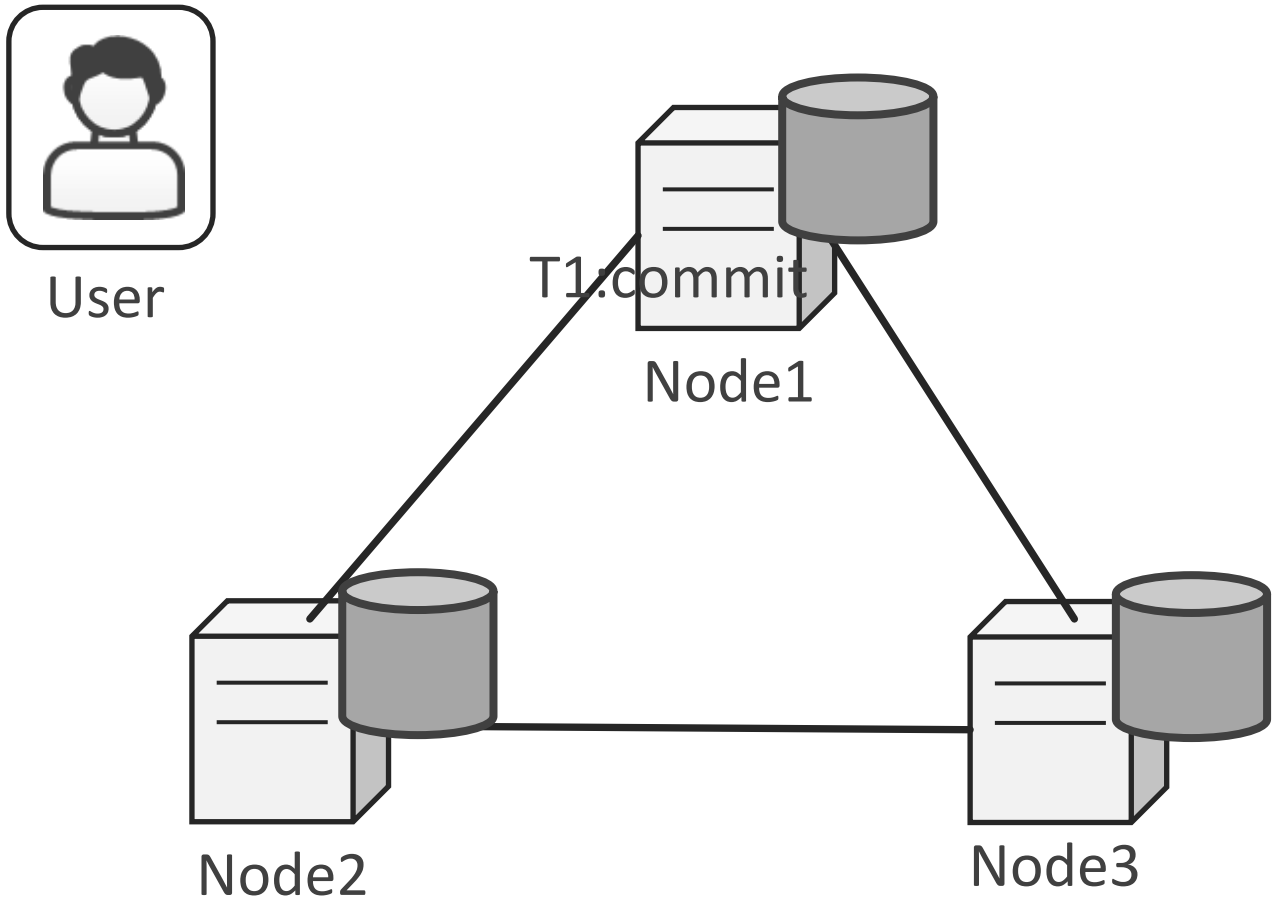
Background: Replicated Database Systems



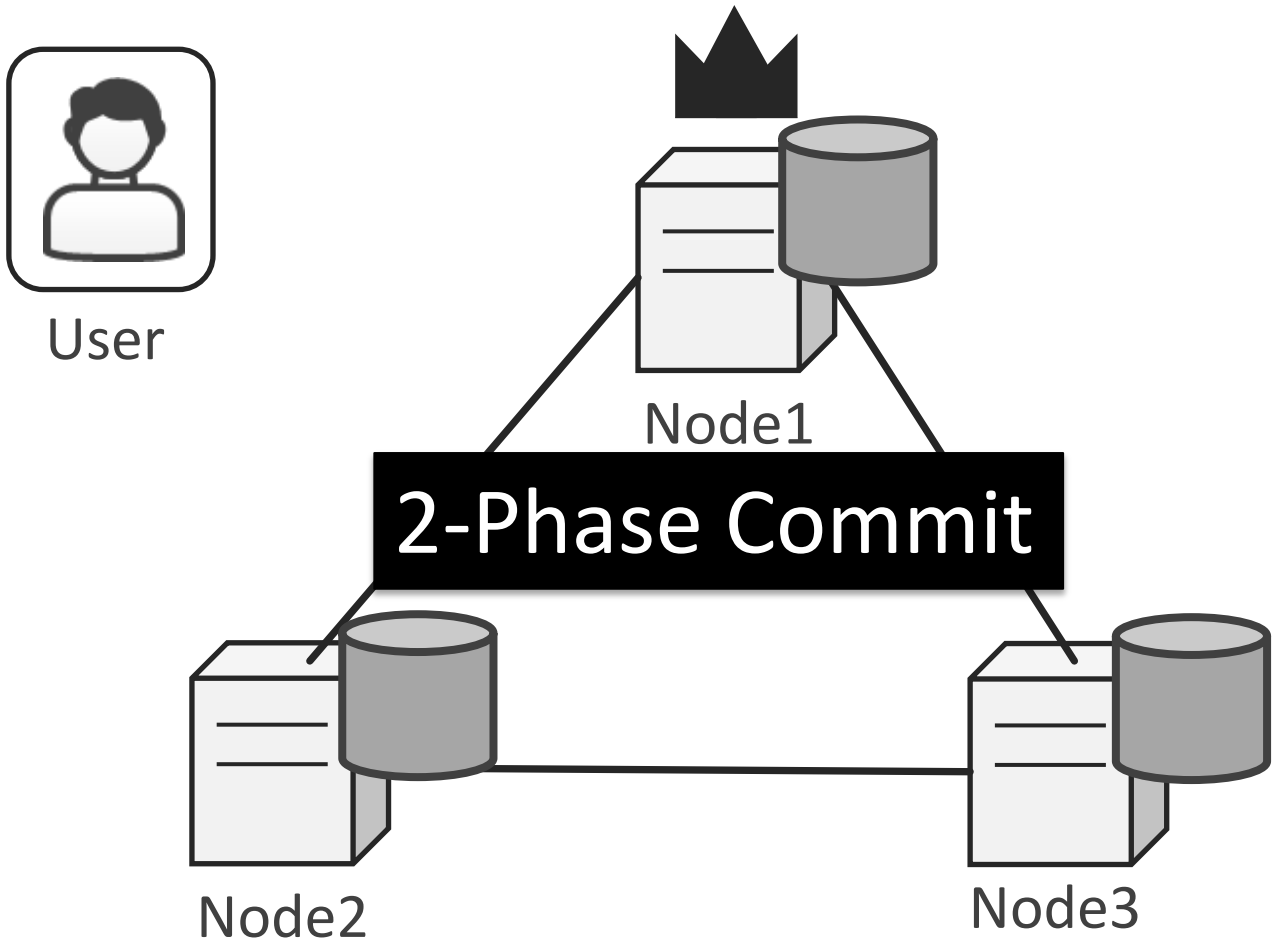
Background: Replicated Database Systems



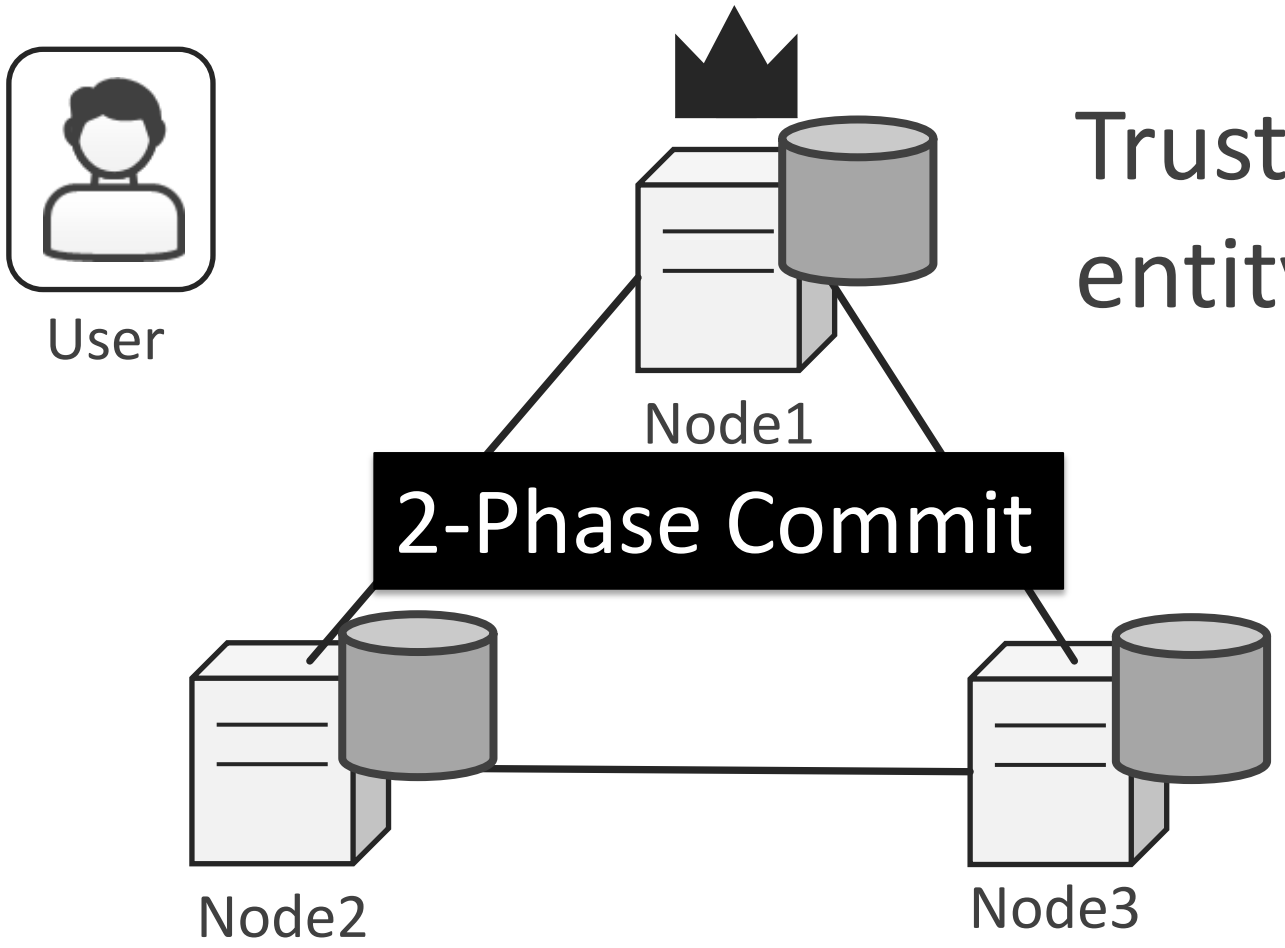
Background: Replicated Database Systems



Background: Replicated Database Systems

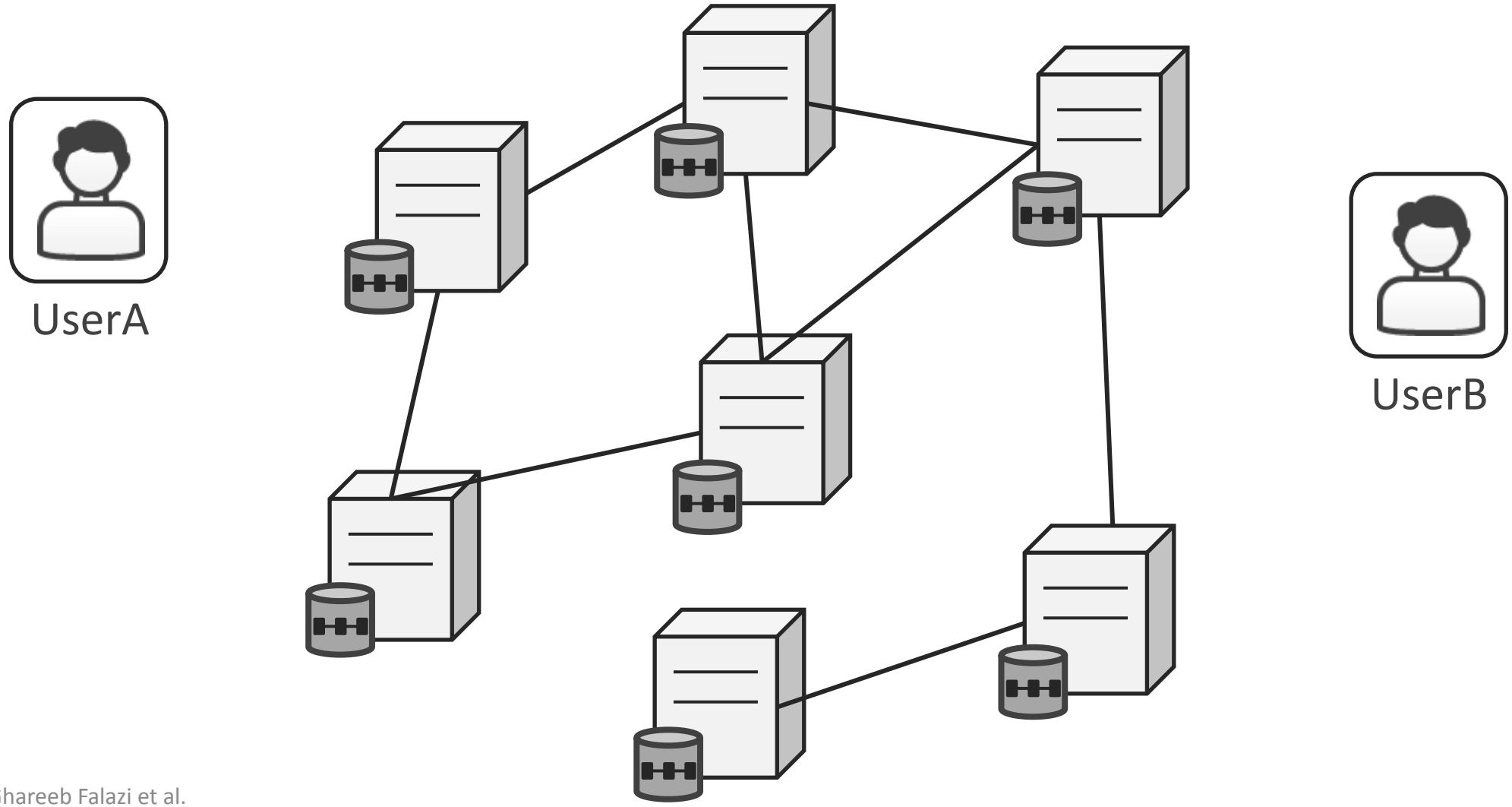


Background: Replicated Database Systems

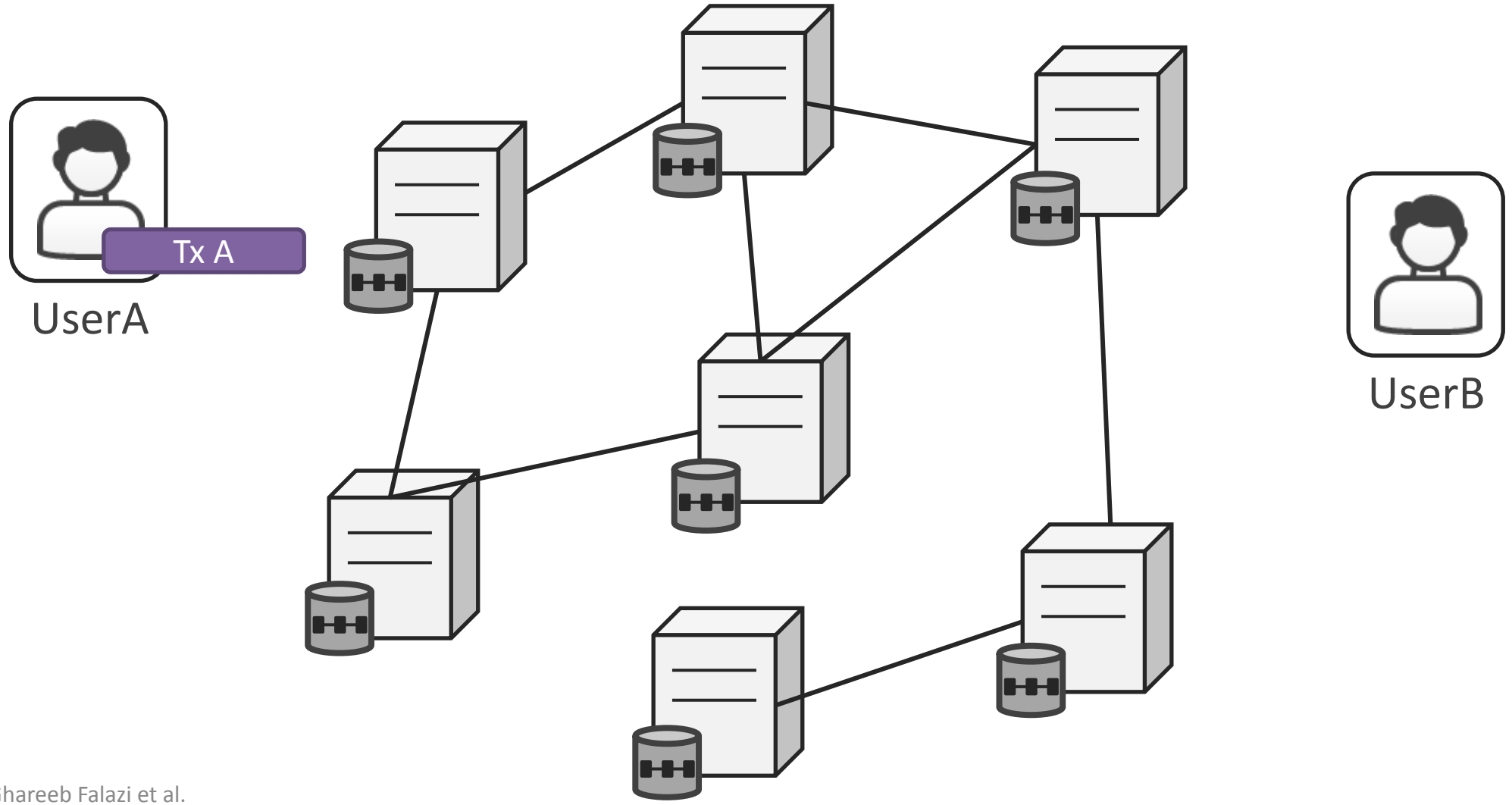


Trust in single entity!

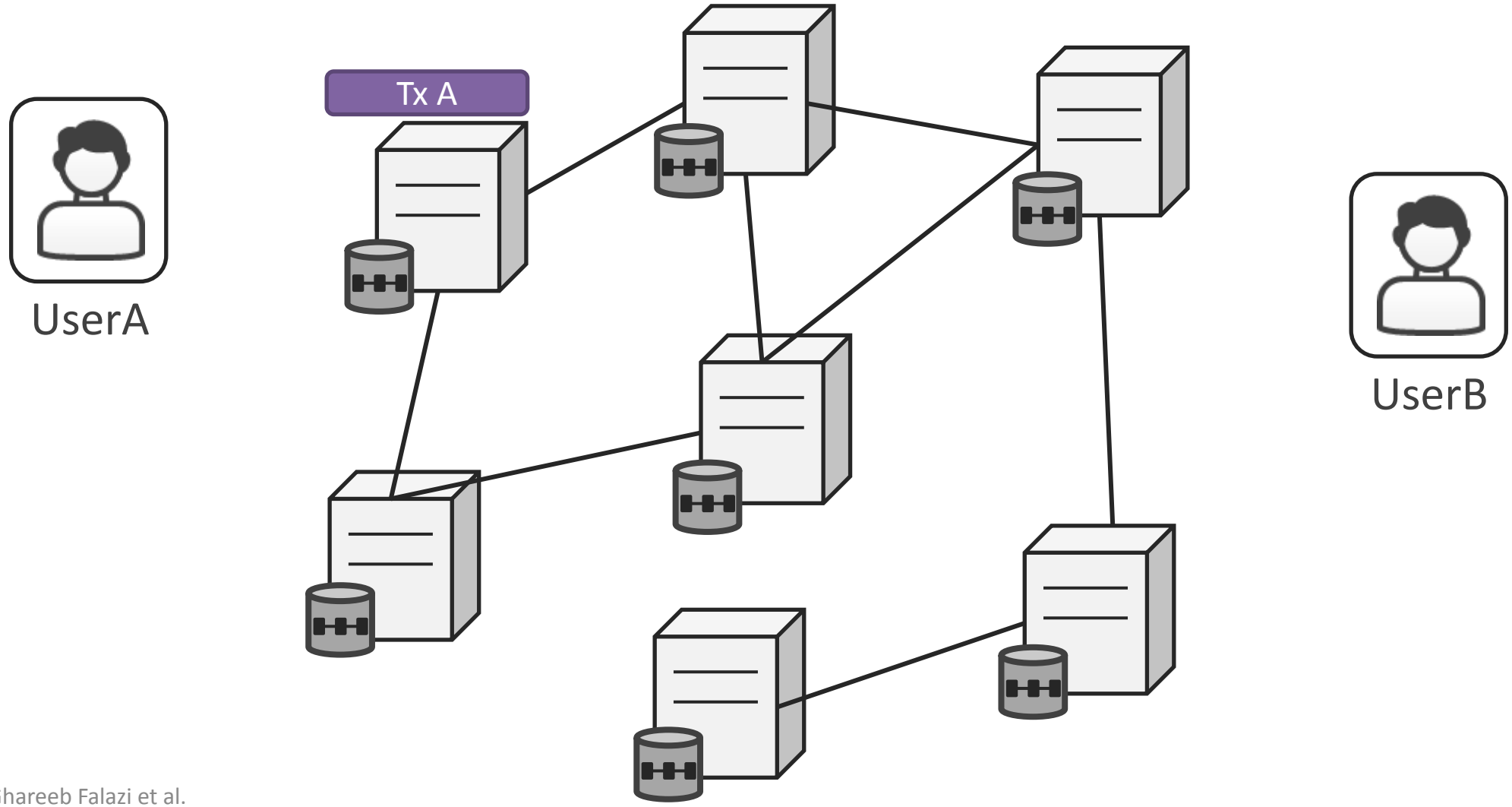
Background: Blockchains



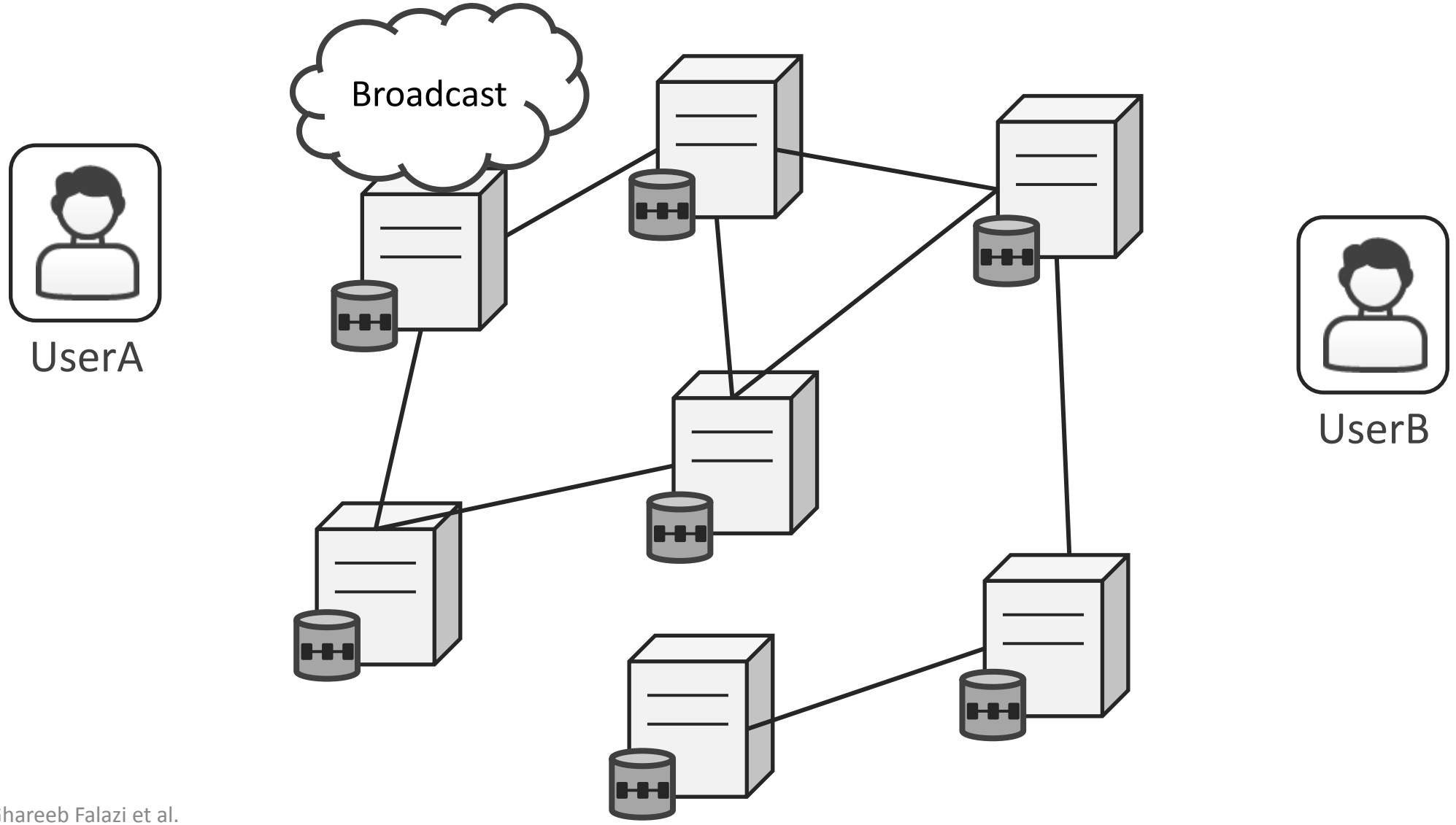
Background: Blockchains



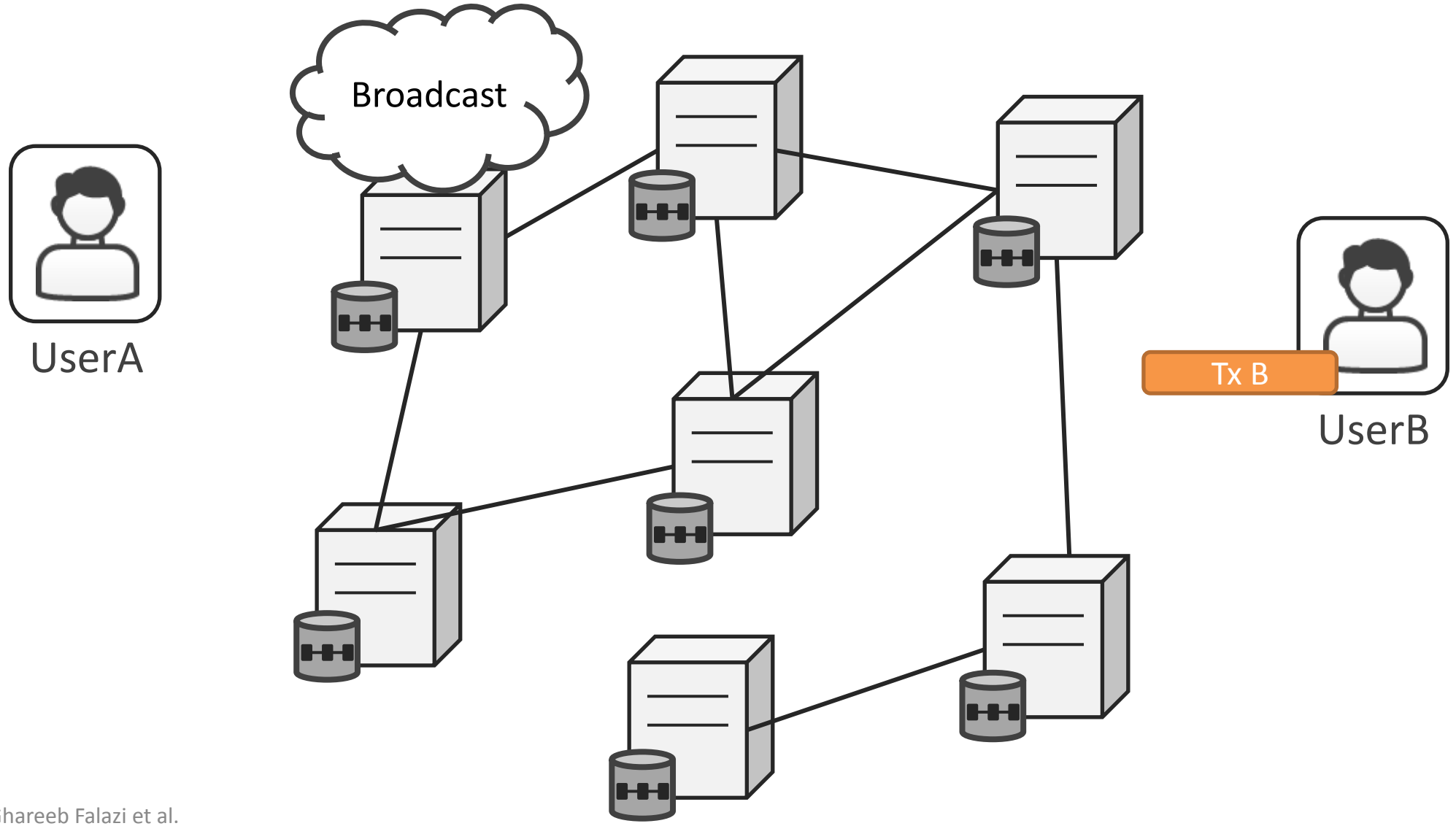
Background: Blockchains



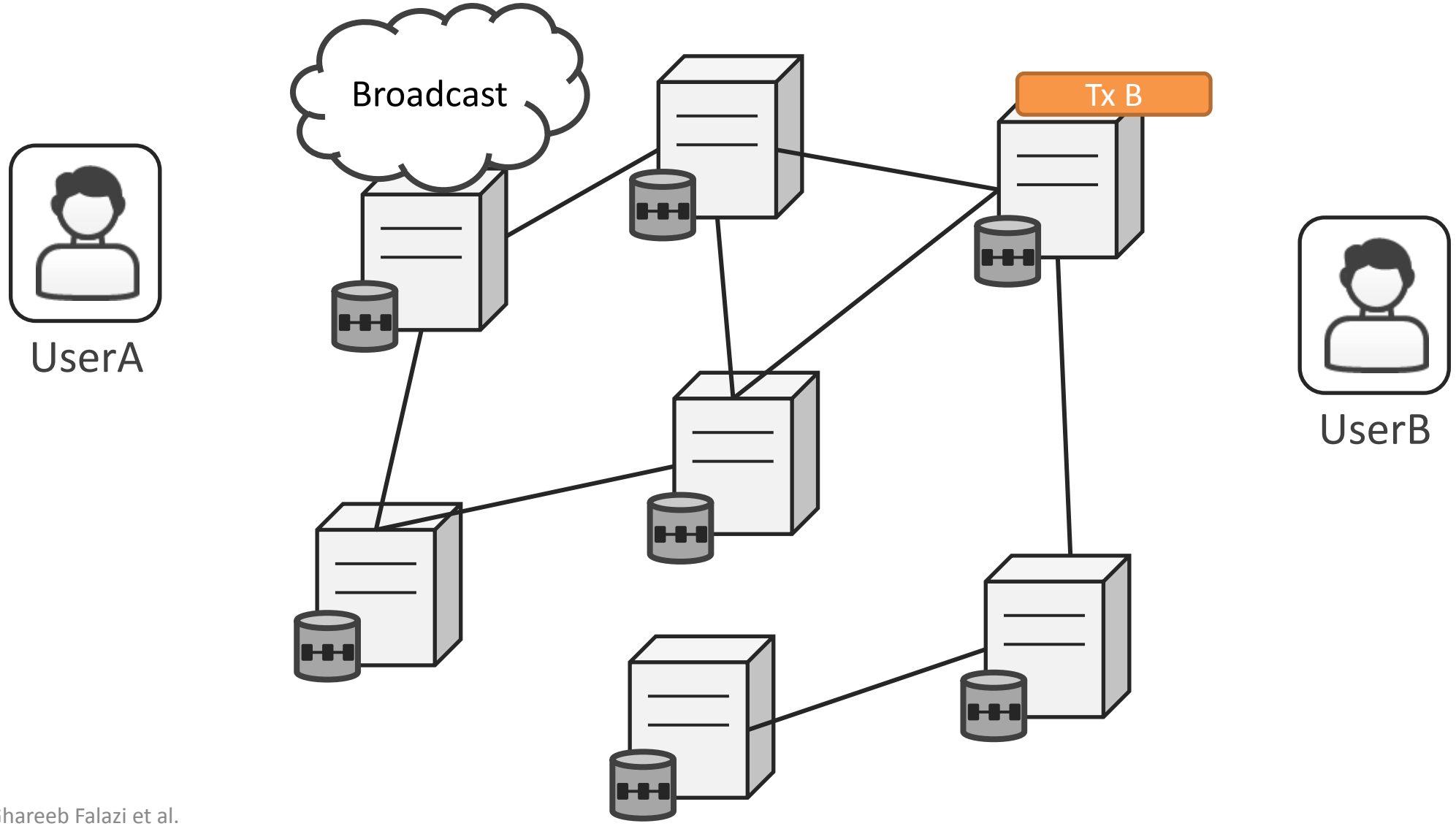
Background: Blockchains



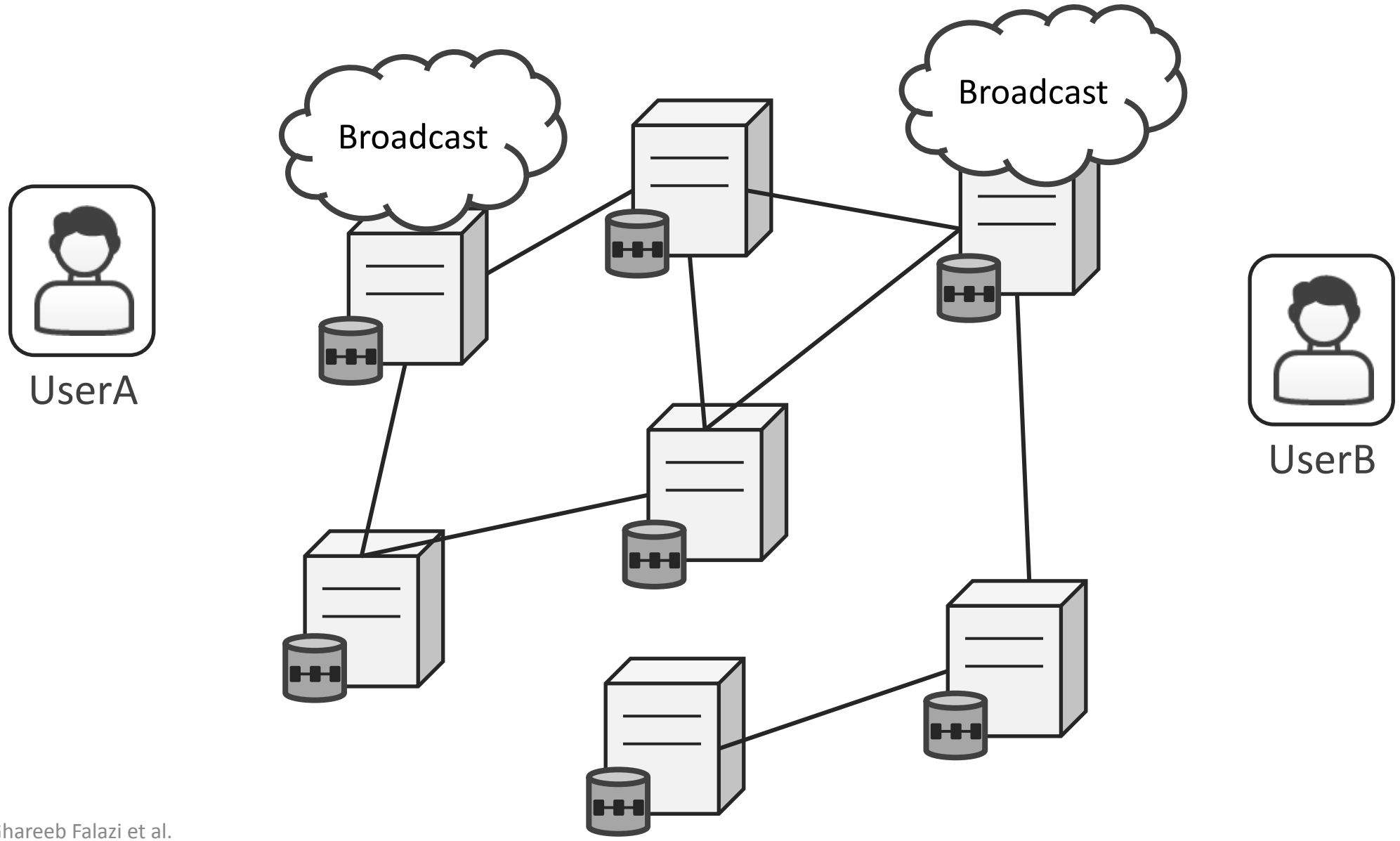
Background: Blockchains



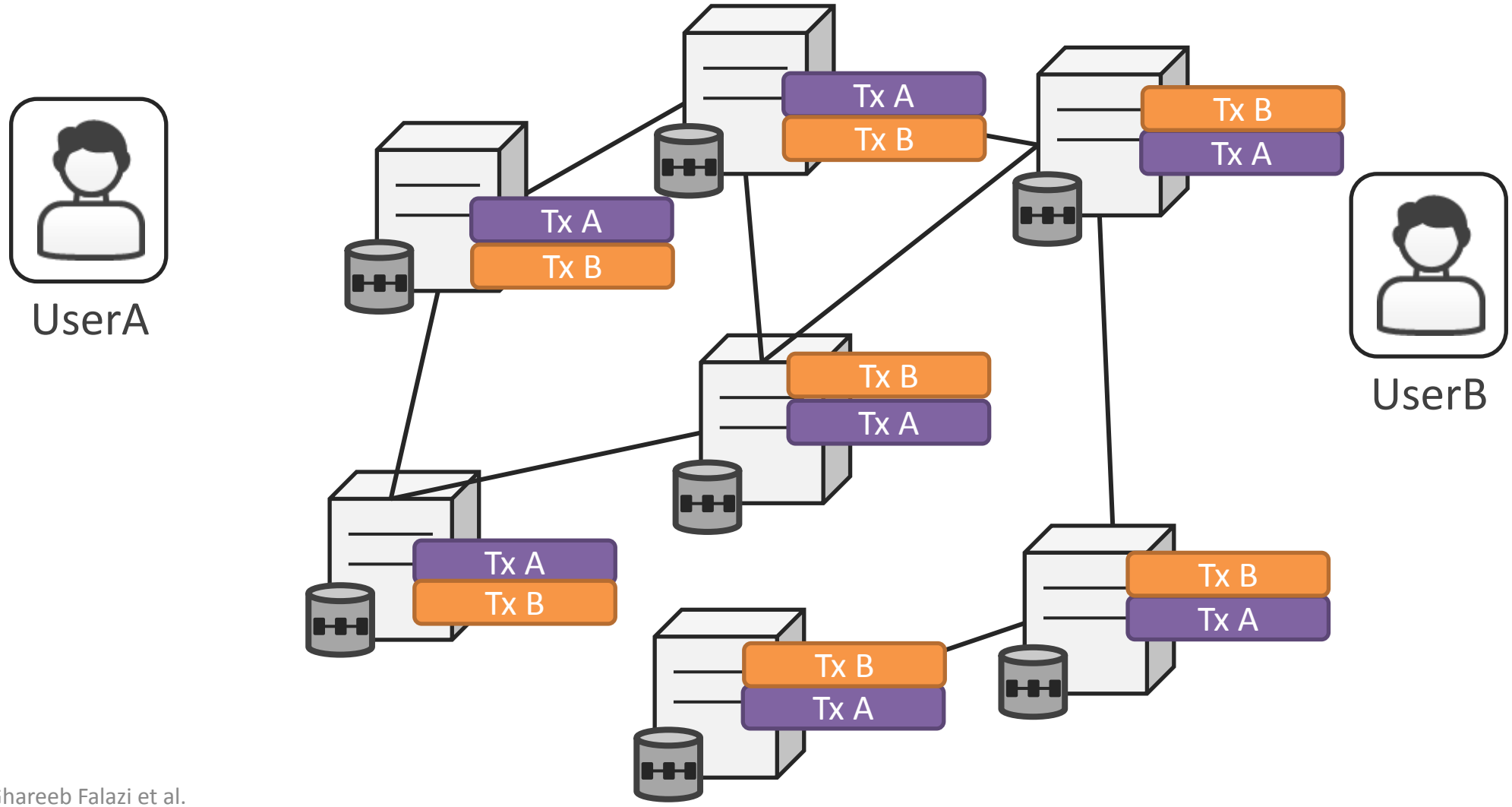
Background: Blockchains



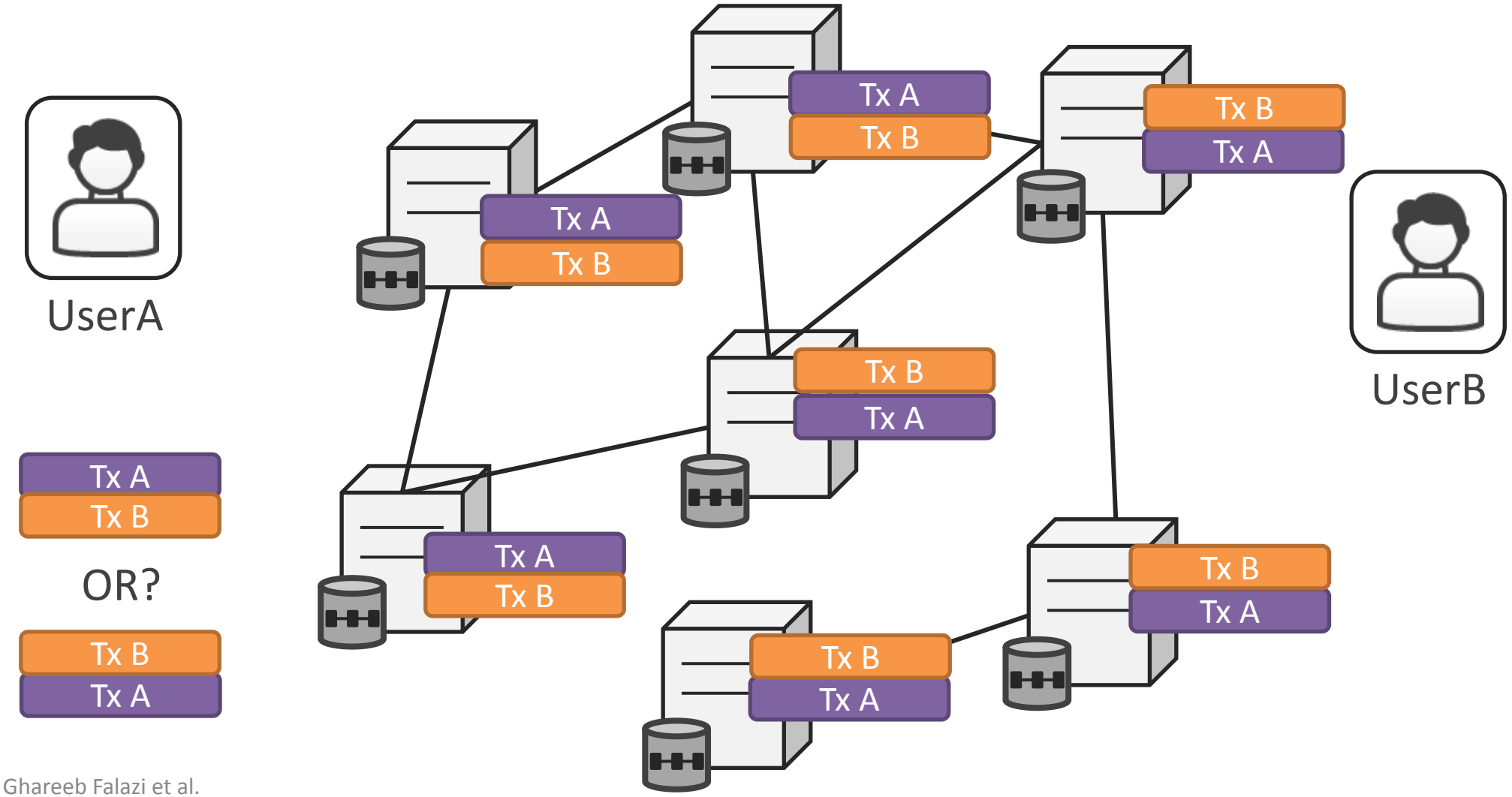
Background: Blockchains



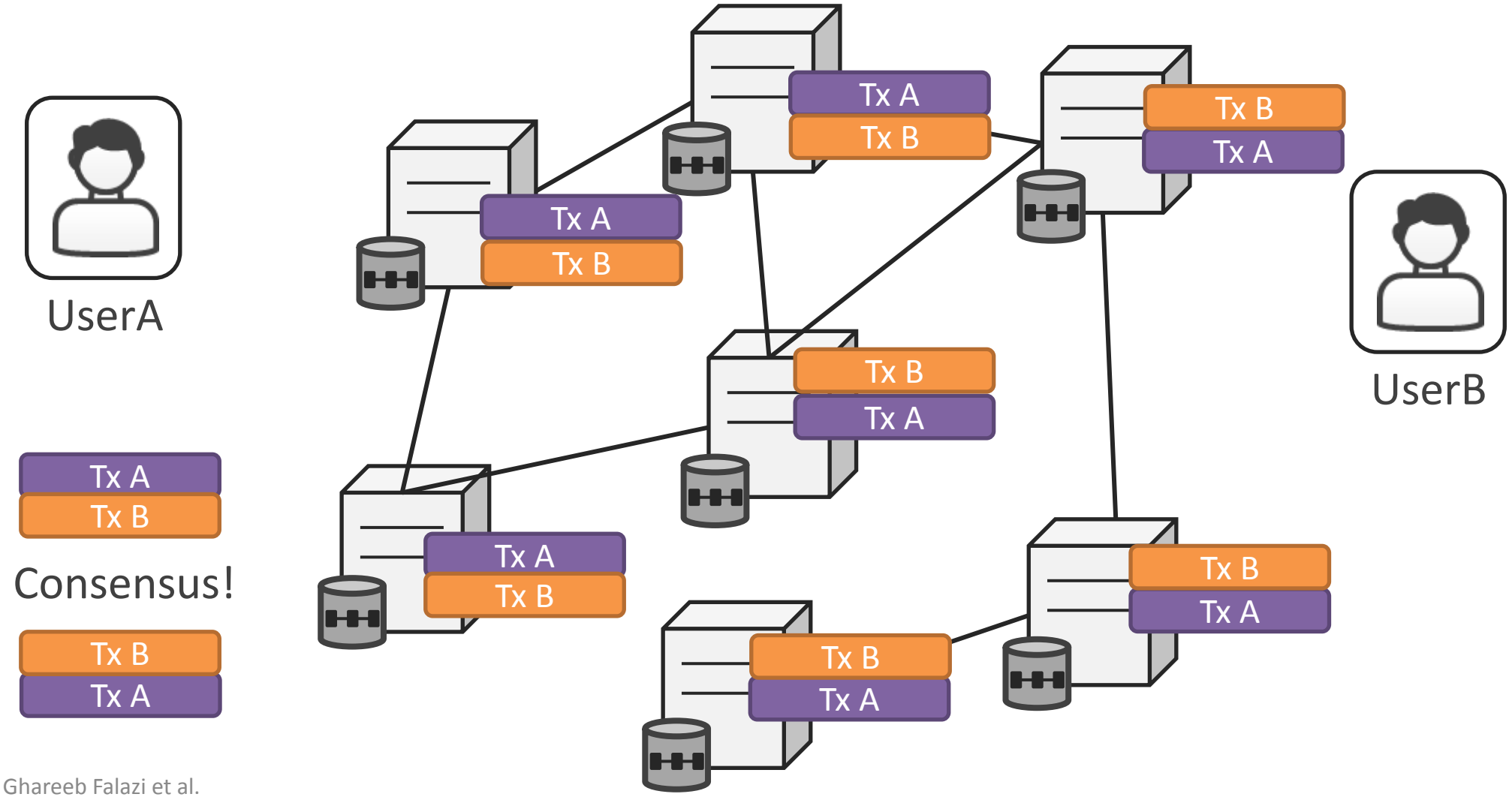
Background: Blockchains



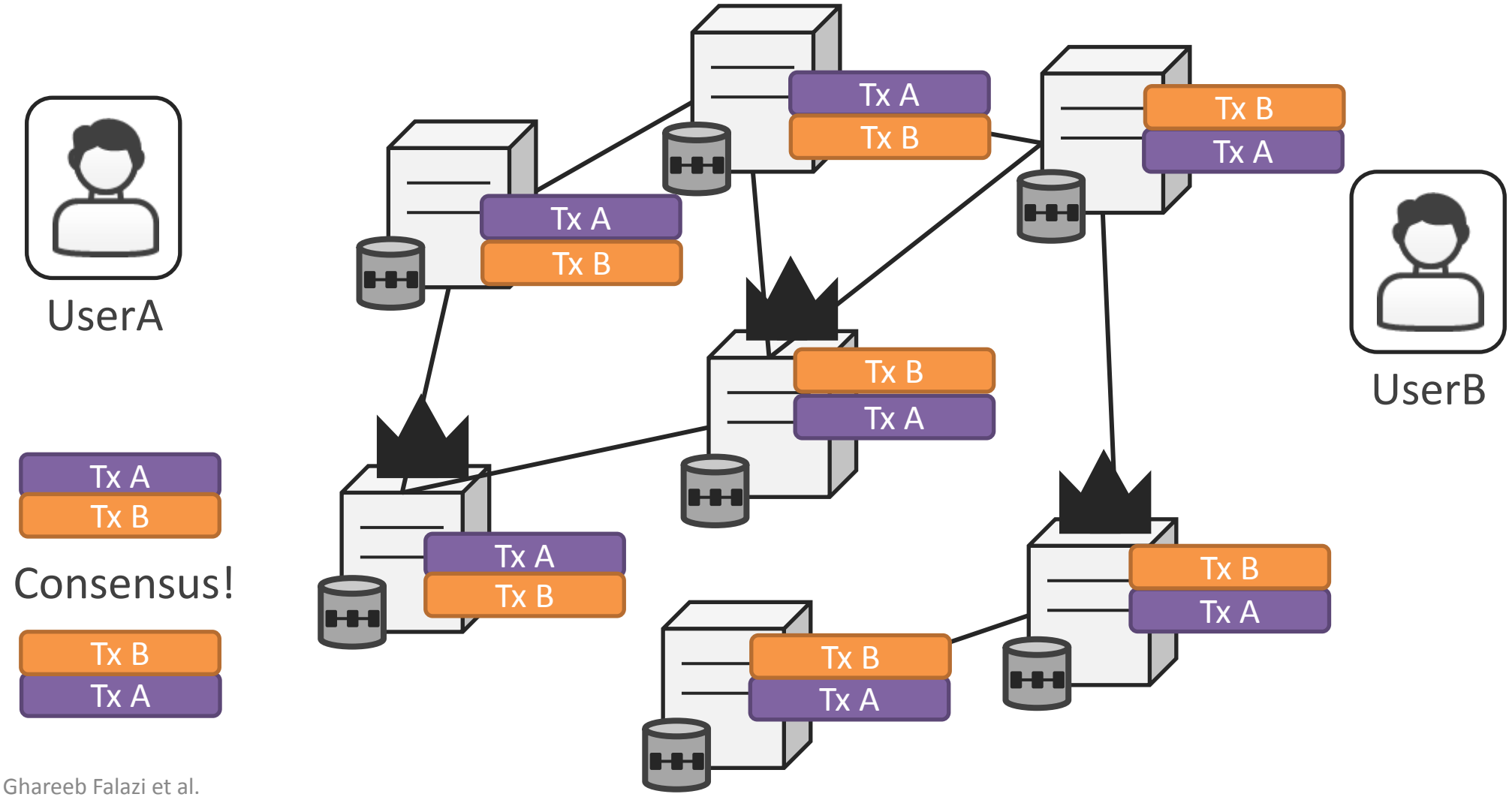
Background: Blockchains



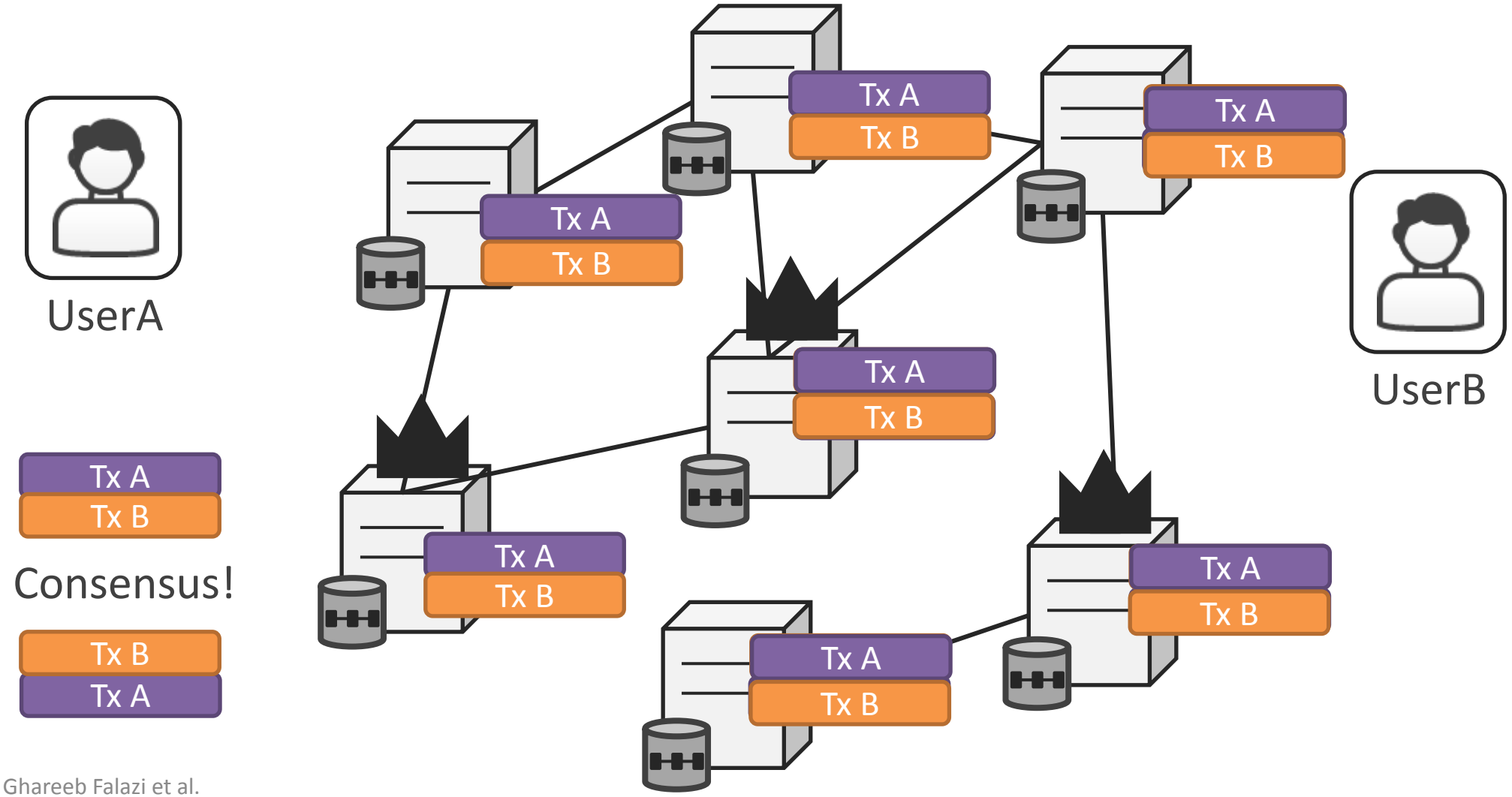
Background: Blockchains



Background: Blockchains



Background: Blockchains



Background: Permissionless Blockchains

- **Open** for anyone (users, regular nodes, and validating nodes -miners-).
- Proof Of Work (**PoW**) consensus protocol has drawbacks:
  - **Slow** (7-20 tx/s).
  - High power consumption.
  - No **data confidentiality**.
  - **Not ACID**[1]

[1]Tai, Stefan, Jacob Eberhardt, and Markus Klems. "Not ACID, not BASE, but SALT."

Background: Permissioned Blockchains

- **Restricted** participation (users, regular nodes, validating nodes).
  - Inherent **data confidentiality**
  - Possible to use Byzantine Fault Tolerant **(BFT)** consensus protocols.
- BFT is:
  - Faster (1000 tx/s).
  - Low power consumption.
  - ACID

Motivation

- Permissioned blockchains are a promising mixture of performance and trust distribution.
- But they are not isolated: existing systems need to interact with them.
- *What are the transaction processing properties of permissioned blockchains?*

Method

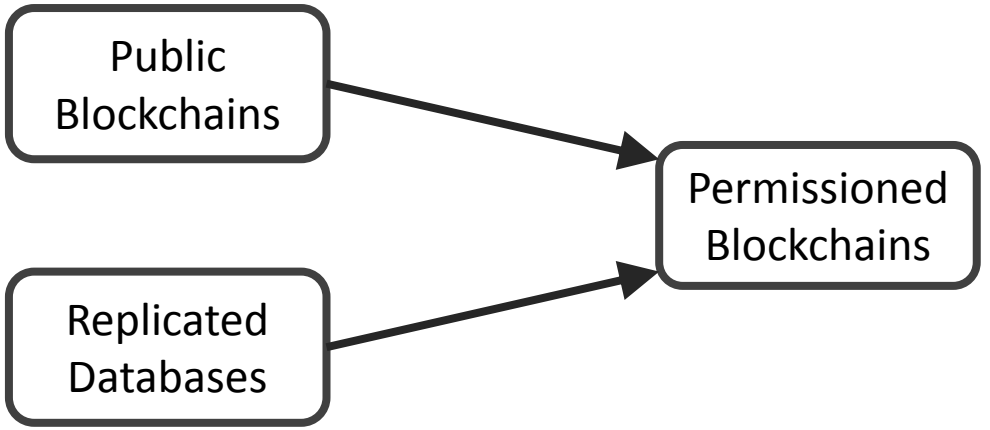
1. Identify the prominent transactional processing properties and guarantees of replicated database systems.
2. Analyze how the considered permissioned blockchain systems process transactions.
3. Determine the resulting properties and guarantees of each system.
4. Categorize systems based on the outcome.

Method

1. Identify the prominent transactional processing properties and guarantees of replicated database systems.
2. Analyze how the considered permissioned blockchain systems process transactions.
3. Determine the resulting properties and guarantees of each system.
4. Categorize systems based on the outcome.

1 – Identifying Suitable Properties and Correctness Criteria

- Permissioned blockchains:
  - An alternative to public blockchains that is scalable, provides finality (mostly), and facilitates confidentiality. OR:
  - A replicated database system that facilitates trustless b2b interactions via the blockchain technology.



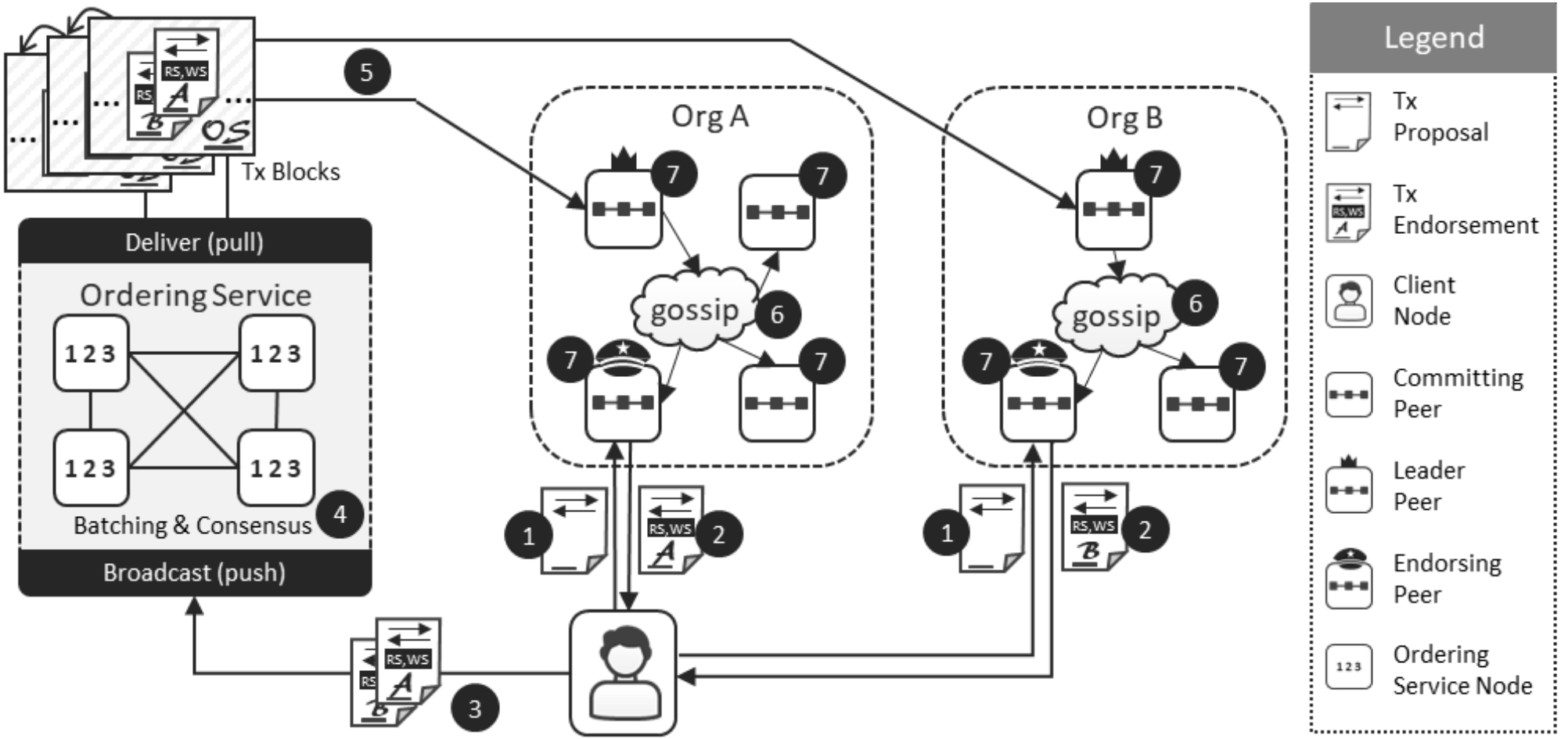
  - **Idea:** treat permissioned blockchains as replicated database systems and evaluate them based on their properties.

1 – Identifying Suitable Properties and Correctness Criteria

Property	Meaning	Category
Local ACID	Each replica should support ACID transactions locally	Correctness Criteria
Global Atomicity	A global transaction either entirely commits or entirely aborts at *all* replicas.	Correctness Criteria
Global Isolation	1-Copy-Serializability: the execution of a set of interleaved global transactions is equivalent to a serial execution of these transactions on a single logical copy of the database.	Correctness Criteria
Session Consistency	Transactions see the effects of previously committed ones from the same session	Correctness Criteria
Location of Transaction	Primary-copy, Update-anywhere	System Property
Synchronization Strategy	Eager, Lazy	System Property
Execution Strategy	What is executed at each replica: Statement replication, Object replication	System Property
Concurrency Control	The mechanism to ensure global isolation: 2PL, Multi-version Concurrency Control	System Property
Architecture	The place where the replica control protocol is: Kernel-based, Middleware-based	System Property

Based on: Kemme B, Jiménez-Peris R, Patiño-Martínez M, Alonso G (2010) Database replication: a tutorial, Springer, chap 12, pp 219–252

2 – Analysis of Existing Systems – Example: Hyperledger Fabric



Transaction flow in Hyperledger Fabric

3 & 4 – Determining Properties and Categorizing Systems

	Aura, Clique, Multichain	Quorum[a], Ripple	Hyperledger Sawtooth[b]	Chain[c]	Tendermint /BigchainDB	Hyperledger Fabric
Local ACID	No	Yes	Yes	Yes	Yes	Yes
Global Atomicity	No	Yes	Yes	Yes	Yes	Yes
Global Isolation	No	1CS	1CS	1CS	1CS	1CS
Session Consistency	No	Yes	Yes	No	No[d]	No
Tx Location	Anywhere	Anywhere	Anywhere	Anywhere	Anywhere	Policy-Driven
Execution Strategy	Symmetric	Symmetric	Symmetric	Symmetric	Symmetric	Asymmetric
Architecture	Kernel-Based	Kernel-Based	Kernel-Based	Kernel-Based	Middleware-based	Middleware-Based
Synchronization Strategy	Lazy	Eager	Eager	Eager	Eager	Eager
Concurrency Control	Serial Execution	Serial Execution	Deterministic Predecessor List	Serial Execution	Serial Execution	MVVC

[(a)] Providing that IBFT is used, and considering public transactions only. [(b)]Assuming a consensus choice guaranteeing finality. [(c)]Assuming correctly behaving block generator. [(d)]Other applications using Tendermint can enforce it.

3 & 4 – Determining Properties and Categorizing Systems

	Aura, Clique, Multichain	Quorum[a], Ripple	Hyperledger Sawtooth[b]	Chain[c]	Tendermint /BigchainDB	Hyperledger Fabric
Local ACID	No	Yes	Yes	Yes	Yes	Yes
Global Atomicity	No	Yes	Yes	Yes	Yes	Yes
Global Isolation	No	1CS	1CS	1CS	1CS	1CS
Session Consistency	No	Yes	Yes	No	No[d]	No
Tx Location	Anywhere	Anywhere	Anywhere	Anywhere	Anywhere	Policy-Driven
Execution Strategy	Symmetric	Symmetric	Symmetric	Symmetric	Symmetric	Asymmetric
Architecture	Kernel-Based	Kernel-Based	Kernel-Based	Kernel-Based	Middleware-based	Middleware-Based
Synchronization Strategy	Lazy	Eager	Eager	Eager	Eager	Eager
Concurrency Control	Serial Execution	Serial Execution	Deterministic Predecessor List	Serial Execution	Serial Execution	MVVC

[(a)] Providing that IBFT is used, and considering public transactions only. [(b)]Assuming a consensus choice guaranteeing finality. [(c)]Assuming correctly behaving block generator. [(d)]Other applications using Tendermint can enforce it.

3 & 4 – Determining Properties and Categorizing Systems

	Aura, Clique, Multichain	Quorum[a], Ripple	Hyperledger Sawtooth[b]	Chain[c]	Tendermint /BigchainDB	Hyperledger Fabric
Local ACID	No	Yes	Yes	Yes	Yes	Yes
Global Atomicity	No	Yes	Yes	Yes	Yes	Yes
Global Isolation	No	1CS	1CS	1CS	1CS	1CS
Session Consistency	No	Yes	Yes	No	No[d]	No
Tx Location	Anywhere	Anywhere	Anywhere	Anywhere	Anywhere	Policy-Driven
Execution Strategy	Symmetric	Symmetric	Symmetric	Symmetric	Symmetric	Asymmetric
Architecture	Kernel-Based	Kernel-Based	Kernel-Based	Kernel-Based	Middleware-based	Middleware-Based
Synchronization Strategy	Lazy	Eager	Eager	Eager	Eager	Eager
Concurrency Control	Serial Execution	Serial Execution	Deterministic Predecessor List	Serial Execution	Serial Execution	MVVC

[a] Providing that IBFT is used, and considering public transactions only. [b]Assuming a consensus choice guaranteeing finality. [c]Assuming correctly behaving block generator. [d]Other applications using Tendermint can enforce it.

Conclusions

- Permissioned blockchains **can be analyzed** by comparing them to replicated database systems.
- A class of systems provides transaction processing properties and guarantees **similar to regular replicated database systems.**
- A class of systems provide transaction processing properties and guarantees **similar to public blockchain systems.**
- **Transaction finality** is the most decisive factor in guaranteeing database-like transaction processing capabilities.
- There is room for improvement especially in guaranteeing **session consistency** and more sophisticated **concurrency control**.

Thank You!
And See You in the Poster Session.