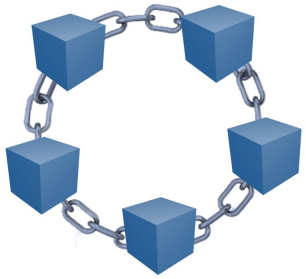
Not ACID, not BASE, but SALT

A Transaction Processing Perspective on Blockchains



Authors: *Stefan Tai, Jacob Eberhardt* and *Markus Klems*
Presentation by: *Georgiou Zacharias* and *Paschalides Demetris*

Overview

1. Blockchain Definition
2. Transaction and System Perspectives
   - 2.1. ACID - Transaction Perspective
   - 2.2. BASE - System Perspective
3. Introduction to SALT
   - 3.1. SALT - Transaction Perspective
   - 3.2. SALT - System Perspective
4. SALT, ACID and BASE Comparison
5. SALTy Use Cases
   - 5.1. Monegraph
   - 5.2. Provenance
6. Challenges

Blockchain Definition

The blockchain developers and technology view

*"A blockchain is a peer-to-peer protocol for **trustless** execution and recording of transactions secured by asymmetric cryptography in a consistent and immutable chain of blocks."*

The IT architect and data management view

"*A blockchain is a shared append-only distributed database with full replication and a cryptographic transaction permissioning model.*"

The business executive and applications view

"*A blockchain is a shared decentralized ledger, enabling business disintermediation and trustless interactions, thereby lowering transaction costs.*"

Transaction and System Perspectives

Comparing Blockchain-based transactions with ACID transactions and BASE model

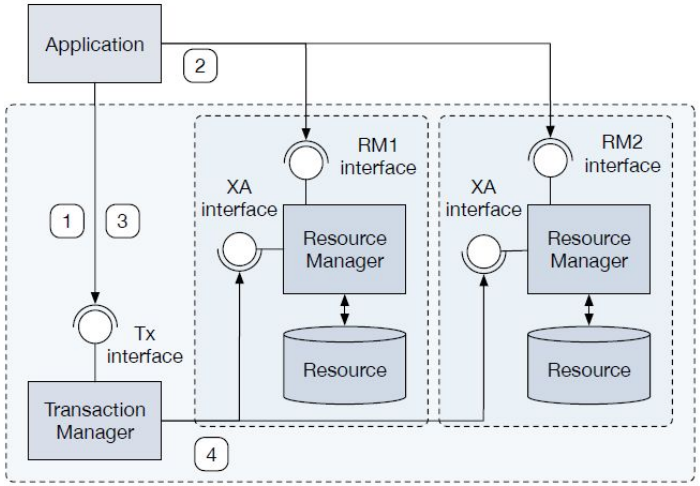
- **Transaction Perspective**
  - **ACID** transactions supported by Relational Database Management Systems
  - **SALT: S**equential, **A**greed, **L**edgered and **T**amper-resistant

- **System Perspective**
  - **BASE** model, favored by cloud systems and NoSQL data stores
  - **SALT: S**ymmetric, **A**dmin-free, **L**edgered and **T**ime-consensual

Transaction Perspective - ACID [1]

***A**tomicity*, ***C**onsistency*, ***I**solation* and ***D**urability*

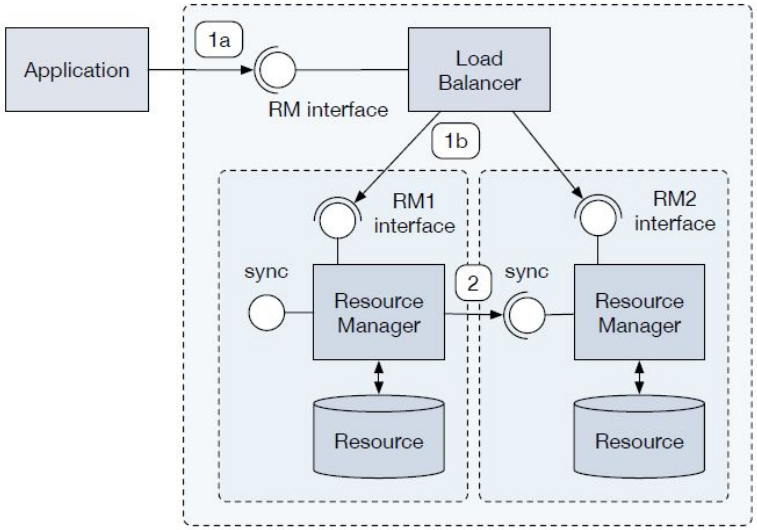
- Transactions executed as a whole or not at all
- Each transaction transforms the database from one consistent, valid state to another
- Concurrent transactions executed by maintaining isolation
- Once a transaction has been committed, the results become permanent



System Perspective - BASE [2]

Basically Available, Soft state, Eventually consistent

- A system is basically available when supporting partial failures without total system failure
- The state of the system is soft in that it can change over time even if no further updates are made
- The system will eventually become consistent, if no new updates are made to the system



Introduction to SALT

- **ACID** transactions provide convenient consistency
- **BASE** systems scale to meet the larger demands

➡ **Both require users' trust**

Blockchains address the trust concern with a system design than enables transactions which do not involve trust in a central party - becoming ***TRUSTLESS***

Blockchains transactions cannot be described by neither ***ACID*** nor ***BASE***.

⇨ **SALT** is introduced as a new acronym to describe the unique properties of blockchain-based transactions and systems

SALT - Transaction Perspective

- **S**equential: All transactions are processed sequentially - ACID's **I**solation.
- **A**greed: A transaction is accepted when the majority of the network agrees on its validity - Unlike ACID and BASE, there is no central authority but a community consensus that determines the system's state.
- **L**edgered: All agreed-on transactions are added to an append-only transaction ledger and cannot be revoked - ACID's **D**urability.
- **T**amper-resistant: A transaction cannot be manipulated or censored - Unlike with ACID and BASE, there is no central access management. The access control model is completely decentralized and tied to asymmetric cryptography.

SALT - System Perspective

- **S**ymmetric: All nodes in the peer-to-peer network symmetrically share their responsibilities
- **A**dmin-free: There is no concept of system administrator. Updates happen on an individual basis and are subject of a community consensus
- **L**edgered: All peers maintain an append-only data structure of transactions which refer to as the ledger. For each transaction to be appended all nodes need to agree in order to maintain ledger's consistency
- **T**ime-consensual: To ensure timely processing of transactions, the consensus algorithm targets a defined average time between the creation of two blocks

★ ***Block****: a data structure that groups a set of transaction in the blockchain model.*

SALT, ACID and BASE Comparison

- ACID's **A**tomicity: A SALT transaction is simpler and there is no concept of grouping into an atomic unit-of-work.
- ACID's **I**solation: Closely related with SALT's **S**equential property.
- BASE's **S**oft State: ACID and SALT define a global order of transaction execution
- ACID's **C**onsistency: Closely related with SALT's **A**greed-on property that refers to the validity of the transactions
- ACID's **D**urability: Related with SALT's **L**edgered property
- ACID loses in availability for consistency, BASE loses in consistency for availability and SALT loses in scalability for trustlessness [3]

SALTy Use Cases

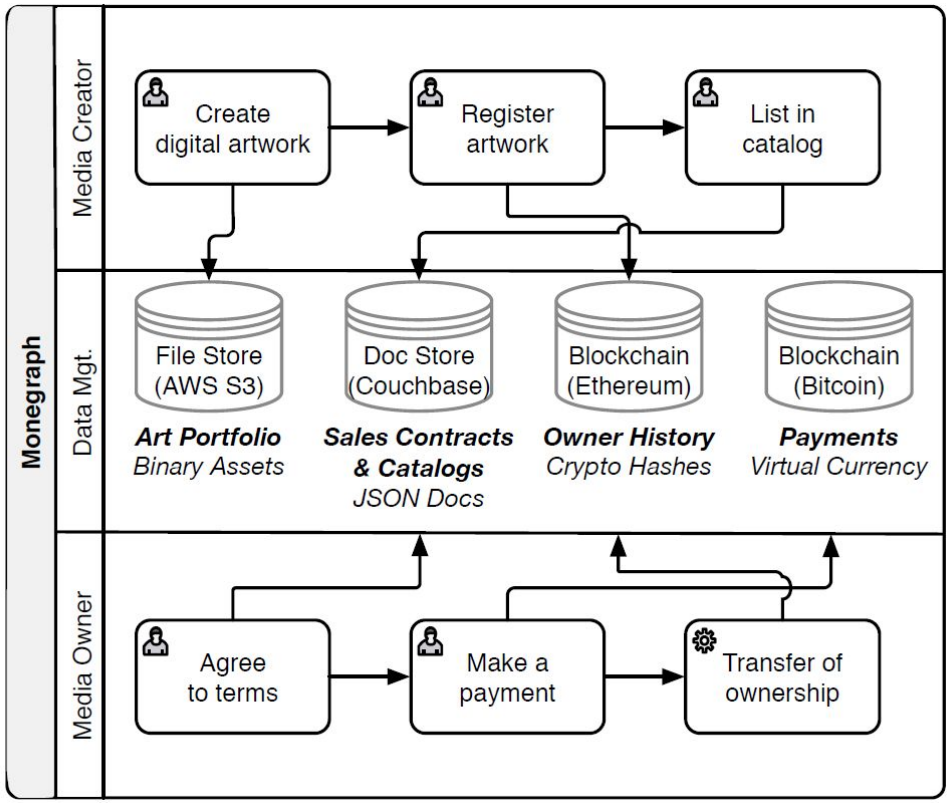
- How to design decentralized transactional applications that use blockchains?
- Innovative companies already adopted blockchain technology in several domains:
  - Financial applications
  - Notary services
  - Digital content monetization
  - Decentralized storage
  - Decentralized IoT
- Two non-financial use cases will be discussed

Monegraph Use Case

Content Distribution and Monetization platform for transferring rights of digital artwork to media owners [4]

Transactions Perspective

- Media creators create and register their digital artwork as a public record
- Media owners use the platform to agree terms, make payments using bitcoin and transfer ownership of the public record



SALT Properties

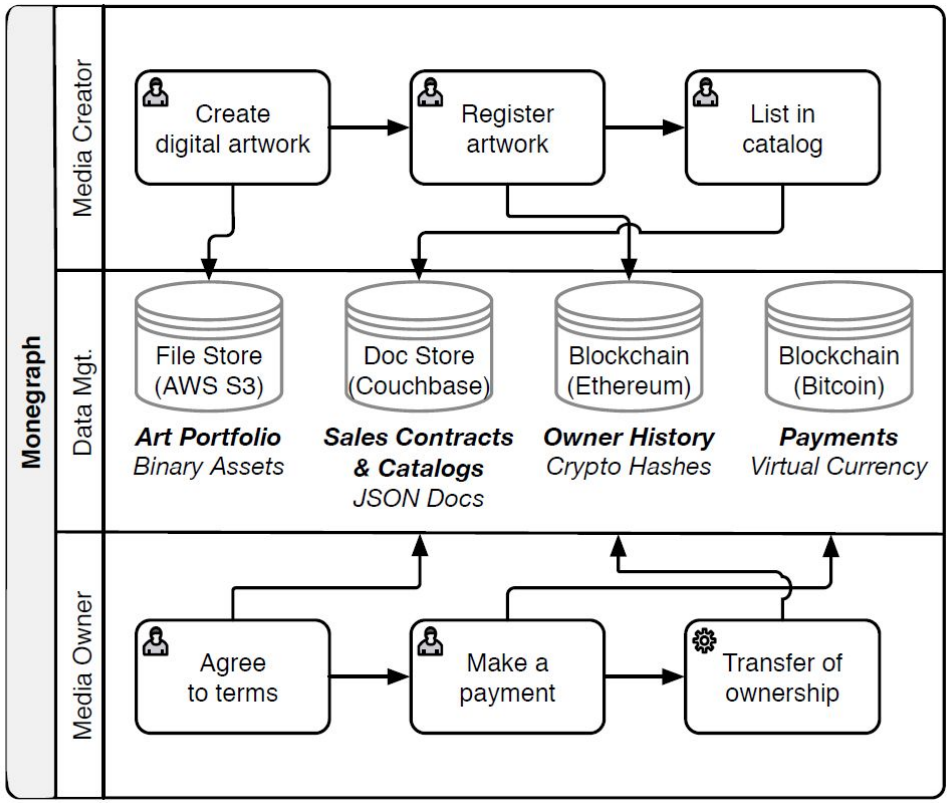
- **Sequential:** Prevent double selling
- **Agreed:** Trust the Majority
- **Ledgered**: Records cannot be revoked
- **Tamper-resistant:** Difficult to forge transactions

Monegraph Use Case

Content Distribution and Monetization platform for transferring rights of digital artwork to media owners [4]

Systems Perspective

- Use of blob storage for digital artworks (Petabytes of unstructured data)
- Use of NoSQL for sales contracts and catalog data (Terabytes of semi-structured data)
- Use of Ethereum system for ownership history of digital artworks as crypto hashes
- Use of Bitcoin system for facilitating payments



SALT Properties

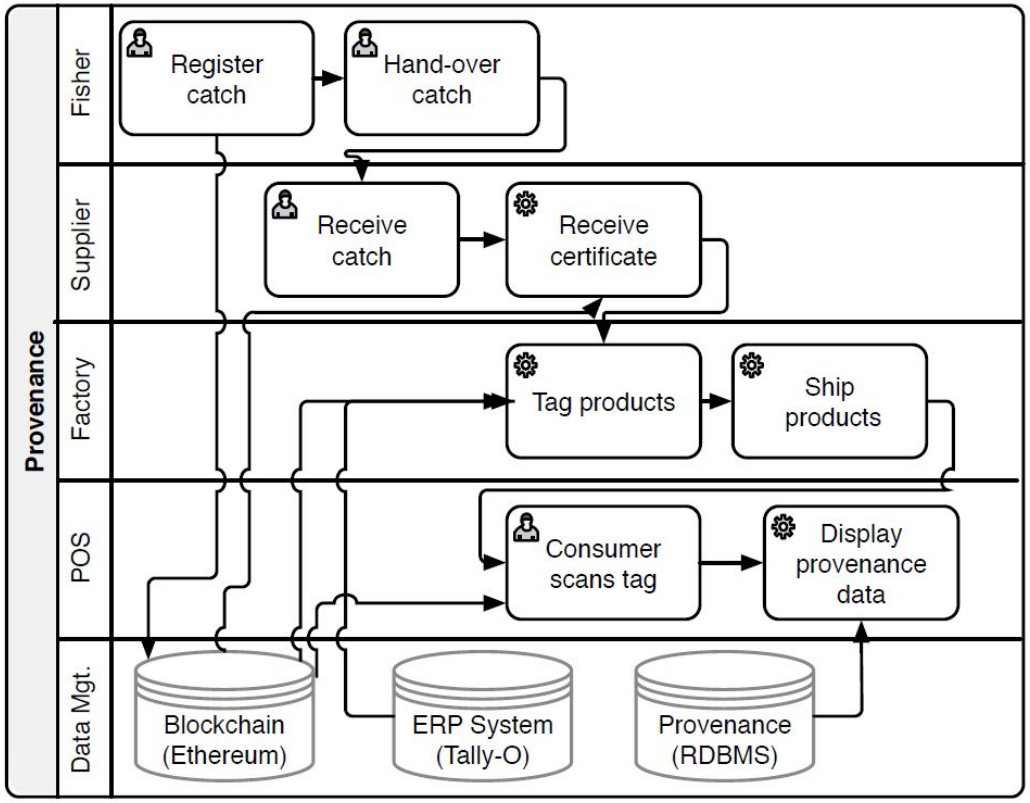
- **Ledgered:** Simplify revenue sharing between multiple parties.
- **Time-consensual:** Append transactions after a defined average time

Provenance Use Case

A supply chain traceability application that provides insight into the provenance of items across different supply chains [5]

Transactions Perspective

- Fisher triggers the creation of a public digital certificate that represents the physical good (e.g., tuna)
- Suppliers receive the product along with its digital certificate
- The factory tags the product (e.g., using QR codes, NFC tags, etc.), and linking to the certificate is created
- Consumers scan the tag and retrieve information regarding the product's provenance



SALT Properties

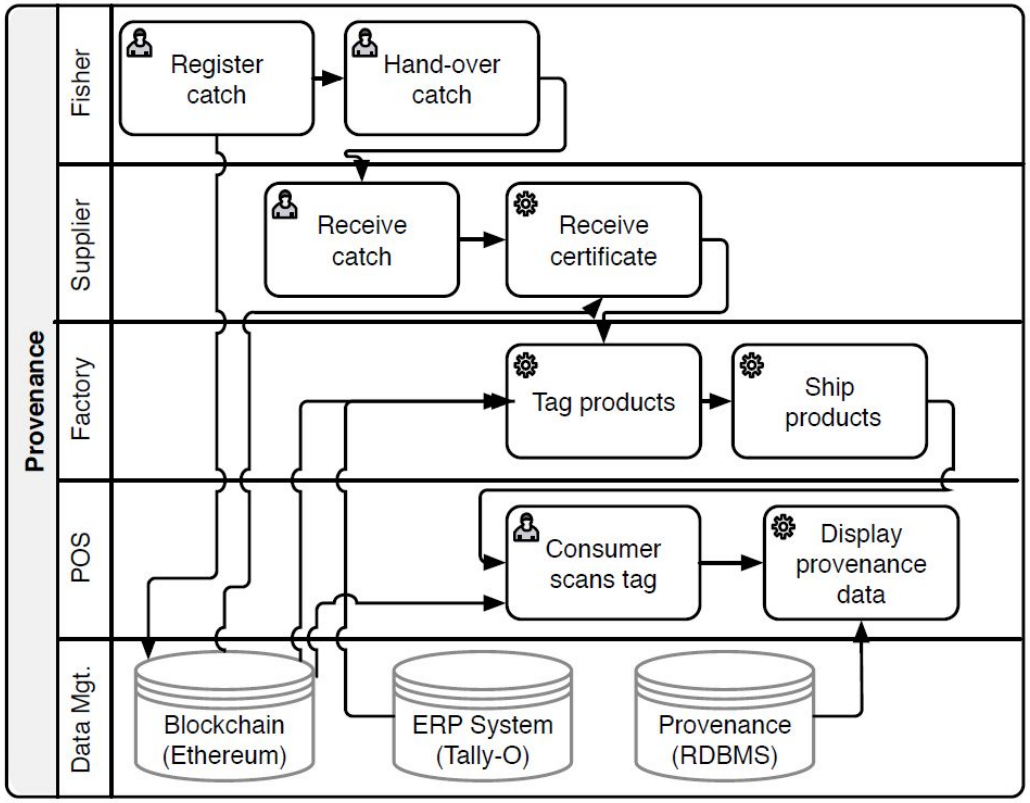
- **Agreed:** All parties agree automatically through a consensus protocol
- **Ledgered:** Practically impossible to revoke transactions once they have been appended

Provenance Use Case

A supply chain traceability application that provides insight into the provenance of items across different supply chains [5]

Systems Perspective

- Use of Ethereum blockchain as a secure, auditable and shared data layer
- Blockchain serves as an integration layer between different stakeholders with heterogeneous IT systems
- Peer-to-peer setup and cryptographic properties make it impossible for single governing party to manipulate data at any point of the supply chain



SALT Properties

- **Symmetric:** Symmetrically shared responsibilities to all stakeholders
- **Admin-free:** No IT system is required for integration and auditing. Also simpler authentication system.

Challenges

- Given a combination of different transaction models and systems, including SALT, **are there system-wide application properties that can be guaranteed?**
- With blockchain technology advancing at a rapid pace, **will ACID, BASE and SALT continue to co-exist as alternatives, or can frameworks and solutions stacks be designed that impose an integrated data management using all three models?**
- **Other perspectives?**
  - e.g., a consensus-perspective, or a blockchain technology (platforms and tools) perspective.
- **Is less SALT actually healthy, whenever closed networks with different consensus algorithms and asymmetric peers are suggested ?**
  - Private and permissioned blockchains, as opposed to the public blockchains discussed in this paper, maintain some of the SALT properties, while compromising others.
- **How, and at what costs, can scalability be improved?**
  - Scalability (of blockchains and blockchain-based applications) is a major limitation of current blockchain technology.

References

1. Gray, J. and Reuter, A. (1992). Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition

2. Pritchett, D. (2008). BASE: An acid alternative. ACM Queue, 6(3):48–55

3. Tai, S. (2017). Continuous, trustless, and fair: Changing priorities in services computing. In Advances in Service-Oriented and Cloud Computing (ASOCC). Springer

4. Monegraph (2017). https://monegraph.com/ Accessed: 2017-02-07

5. Provenance (2017). https://www.provenance.org/ - Accessed: 2017-02-07