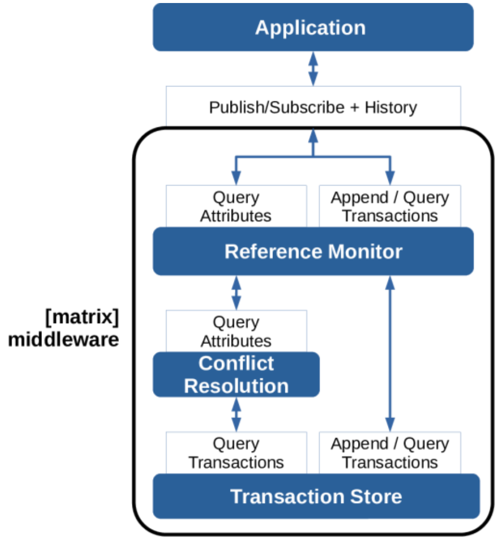
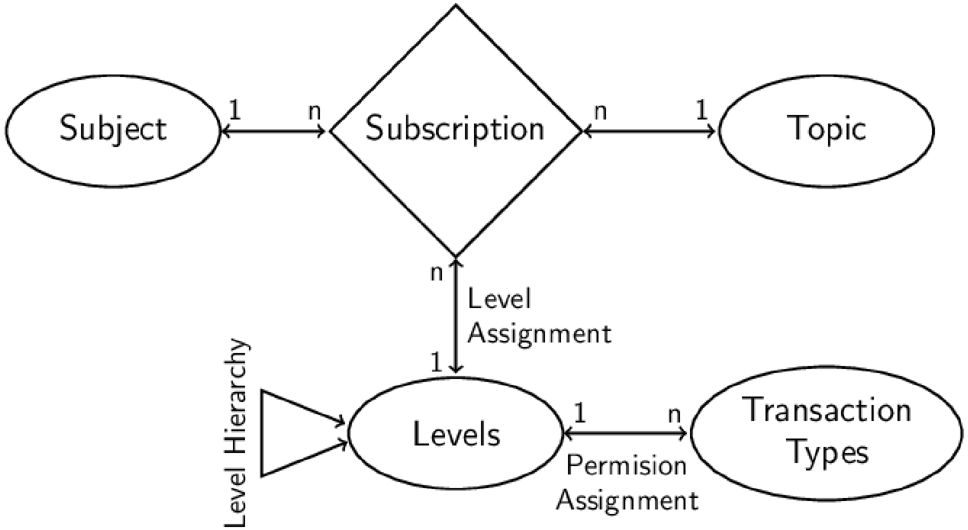
# Matrix Decomposition: Analysis of an Access Control Approach on Transaction-based DAGs without Finality

**Florian Jacob**, Luca Becker, Jan Grashöfer, Hannes Hartenstein

DECENTRALIZED SYSTEMS AND NETWORK SERVICES RESEARCH GROUP (DSN)
INSTITUTE OF TELEMATICS, FACULTY OF INFORMATICS

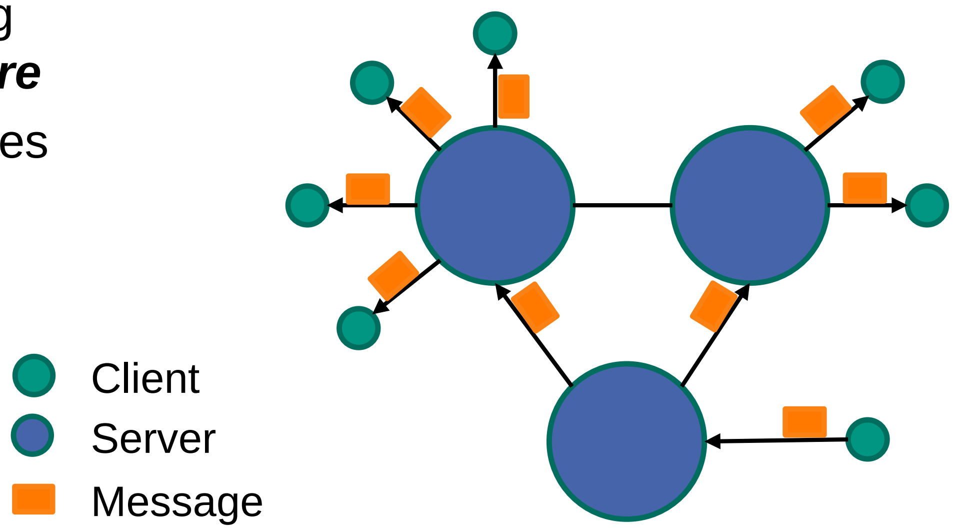


# Introduction to Matrix

- Middleware for decentralized applications
  - Topic-based publish-subscribe
  - Eventually-consistent attribute storage
- Most prominent use case: decentralized instant messaging
  - French government, Mozilla, German Federal Defense Forces, …

- Servers form a network, cooperate with limited mutual trust
  - Replace pure message passing with a ***replicated data structure***
  - Broadcast data structure updates
  - One federation per topic

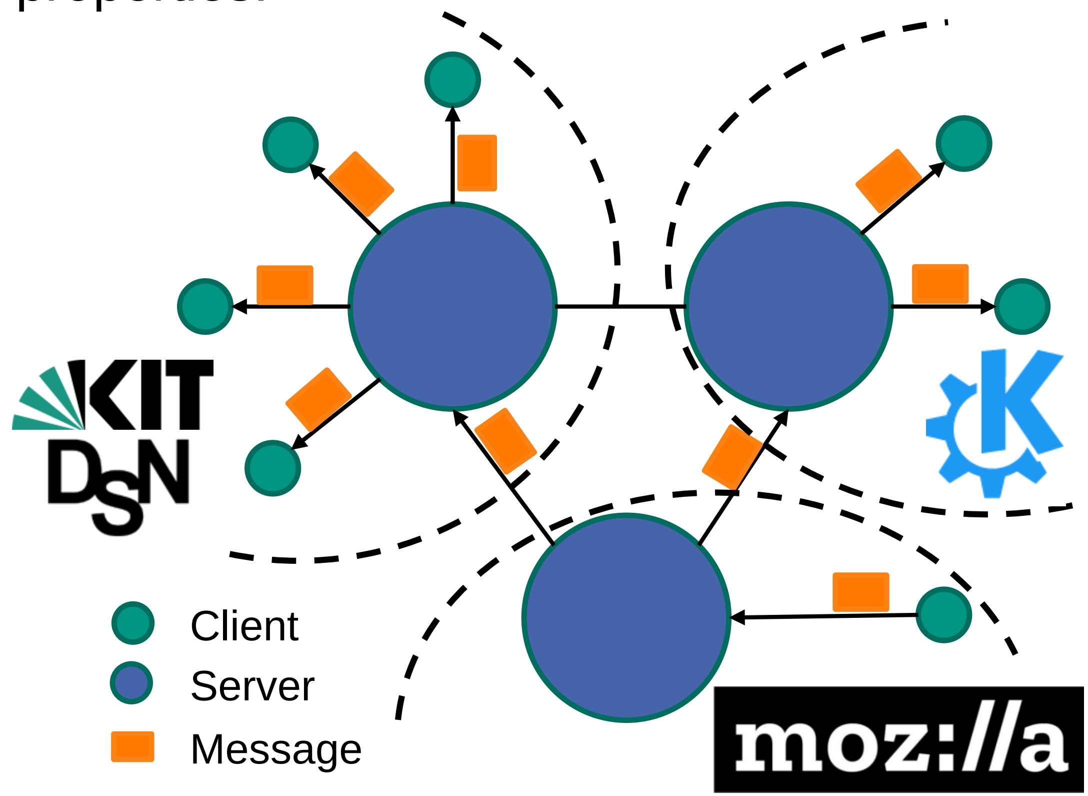


# Federated Publish-Subscribe Access Control

- Is the user allowed to publish the message / update?
- Do other servers accept it, and forward it to their subscribed users?

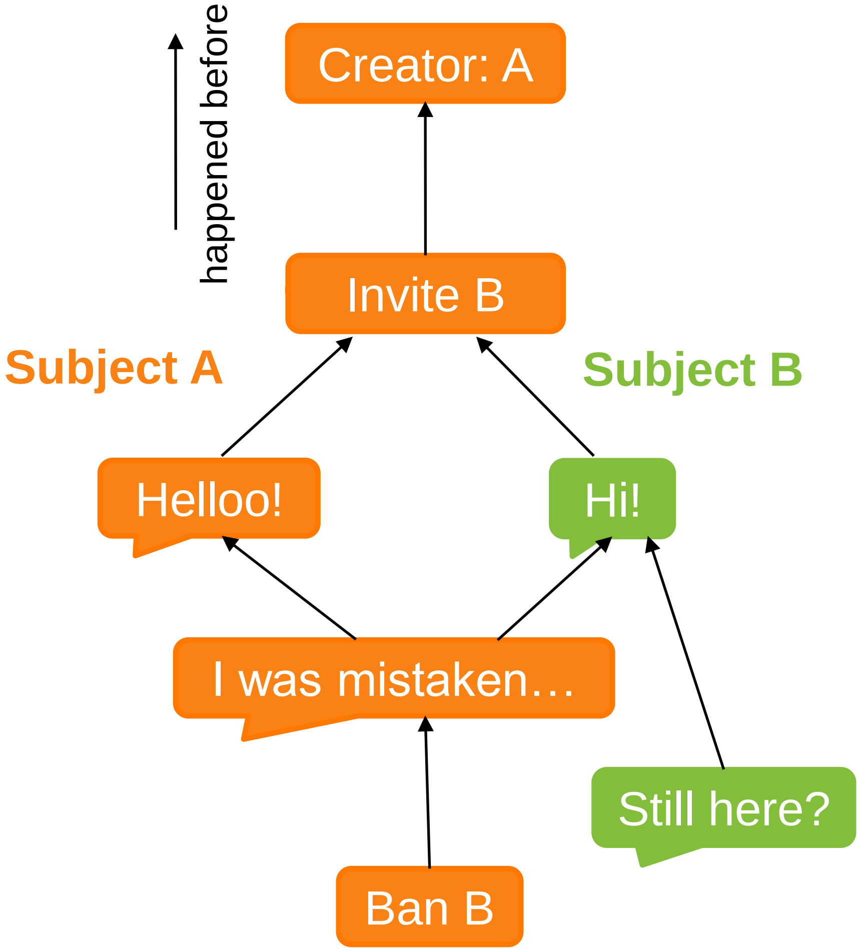
- Message and attribute store properties:
  - Partial (causal) order
  - Eventual Consistency
  - No Finality, no Consensus
  - Byzantine Participants

► Basis for Authorization Database in Matrix!



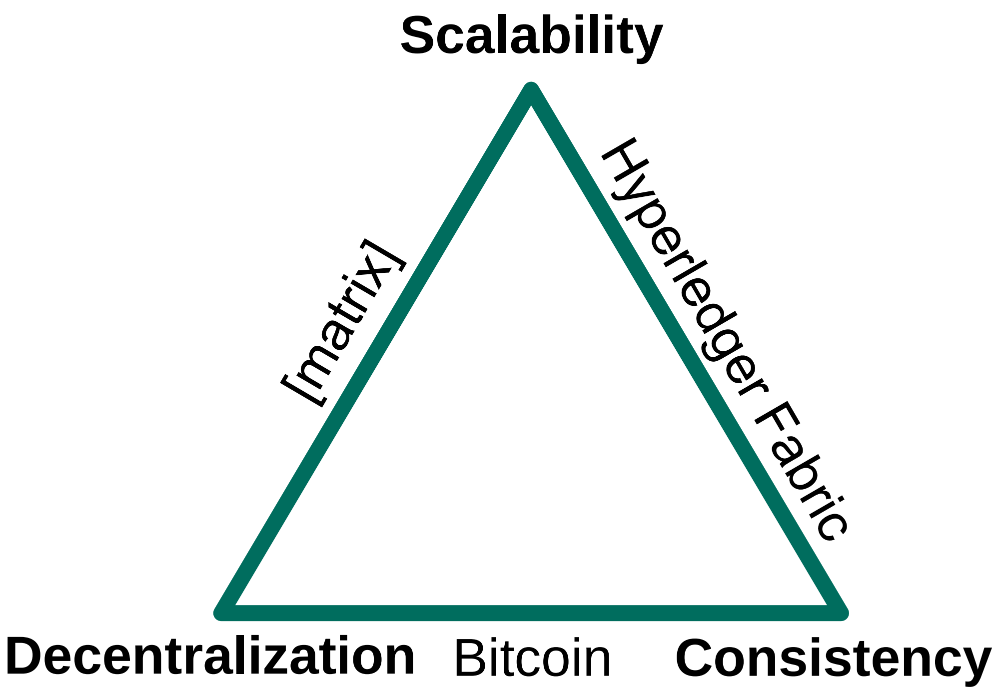
# Message & Attribute Store: Matrix Event Graph

- Per-topic replicated data structure
  - Directed, acyclic graph
  - Causal relation of *transactions*
- Instant messaging use case:
  - topic ≅ chat group / 1:1 chat
  - message ≅ text message, file, …
  - attribute ≅ group description, permission assignments, …
- Properties:
  - Partial (causal) order
  - Eventual Consistency
  - No Finality, no Consensus



# Why are we putting up with these properties?

- Matrix is a Distributed Ledger Technology that scales!
- "DCS Trilemma" of Distributed Ledger Technologies [Zhang2018]:
  - Desired Properties:
    - Decentralization
    - Consistency
    - Scalability
  - Conjecture:
    - Cannot achieve all three
    - Gradual tradeoff




[Zhang2018]: Zhang et al., Towards Dependable, Scalable, and Pervasive Distributed Ledgers with Blockchains.

# Problem Statement

- Unlike conventional DLTs, Matrix does not aim for strict consensus
  - Trades consensus on final total ordering for decentralization and scalability
  - Advantage: no need for consensus mechanisms (e.g. Proof of Work)
  - Consequence: No system-wide consensus on authorization database and access control decisions
- Empirically, access control still works in Matrix
  - ► How is access control possible in such a setting?
  - ► Is it sound & secure?

**Approach:**

A. Analyze Access Control Model

B. Analyze Enforcement Mechanism

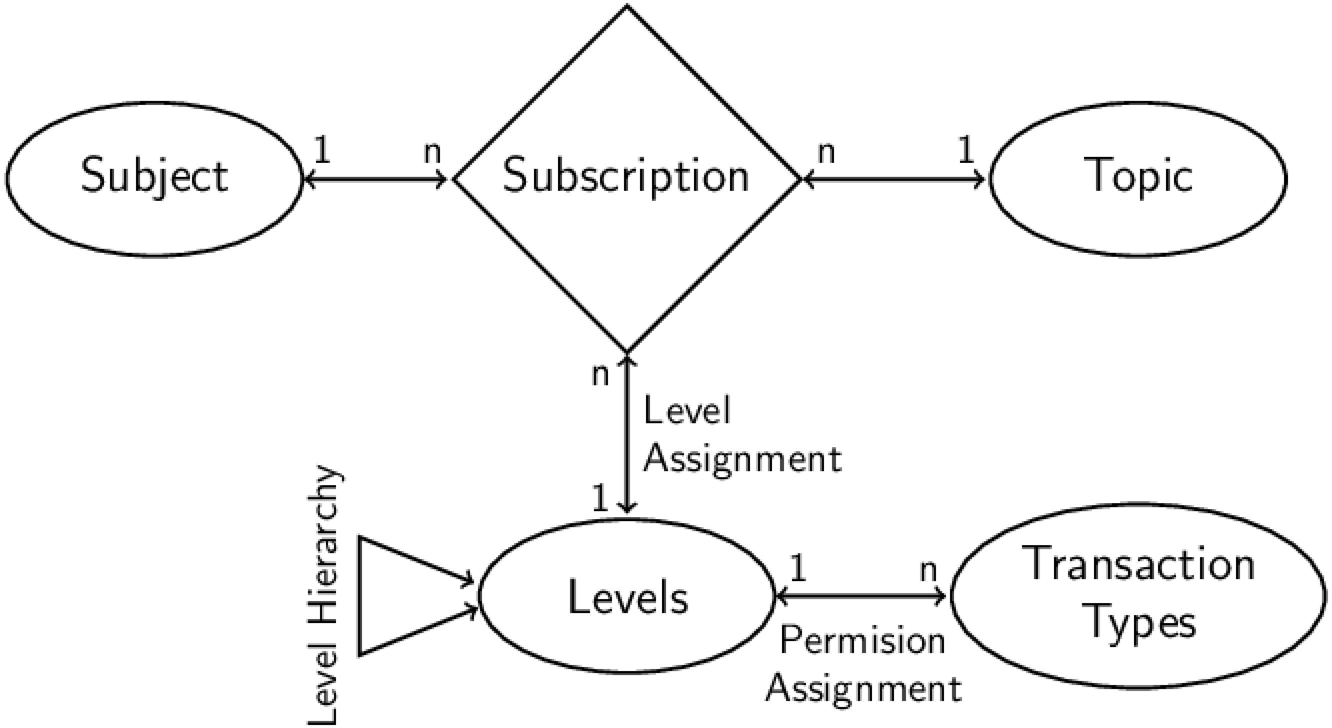
Assertion: Working communication and data structure

# Access Control Model

- LeBAC: Level-based Access Control
  - Assigns levels to subjects and transaction types
  - Variant of Lattice-Based Access Control
    - Specialization: Requires total order on levels
    - **Consequence:** Total order on subjects by permissions

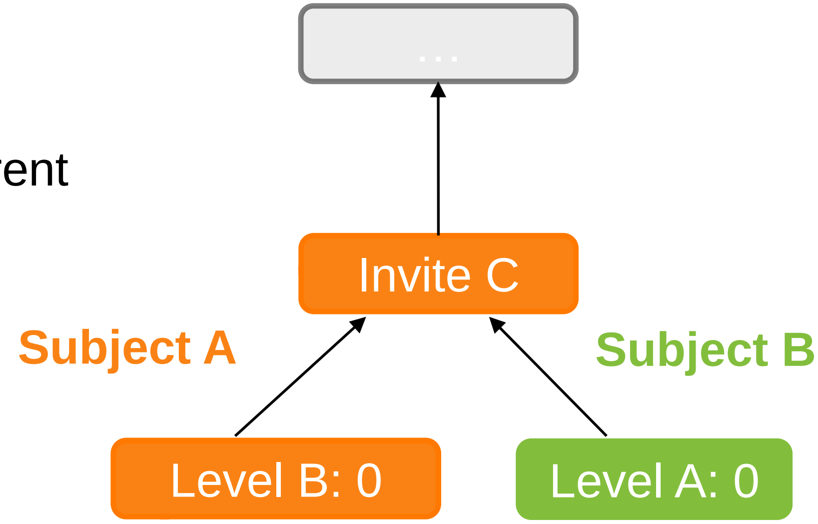
| Subscription | Level |
|---|---|
| Subject A | 100 |
| Subject B | 0 |
| Subject C | 50 |

| Transaction | Level |
|---|---|
| Chat | 0 |
| Invite | 50 |
| Ban | 100 |



# Challenge: Secure Conflict Resolution

- Event Graph: Partial Order
  - Two level assignments can be concurrent
- Problem: Which to prefer of two concurrent attribute changes?




**Conflict Resolution**

- Provides authorization database for Reference Monitor
- General concept: Linearization / Topological Ordering of Causal Order
  - Easy: *some* topological ordering
  - Hard: What is a *secure, consistent* topological ordering?

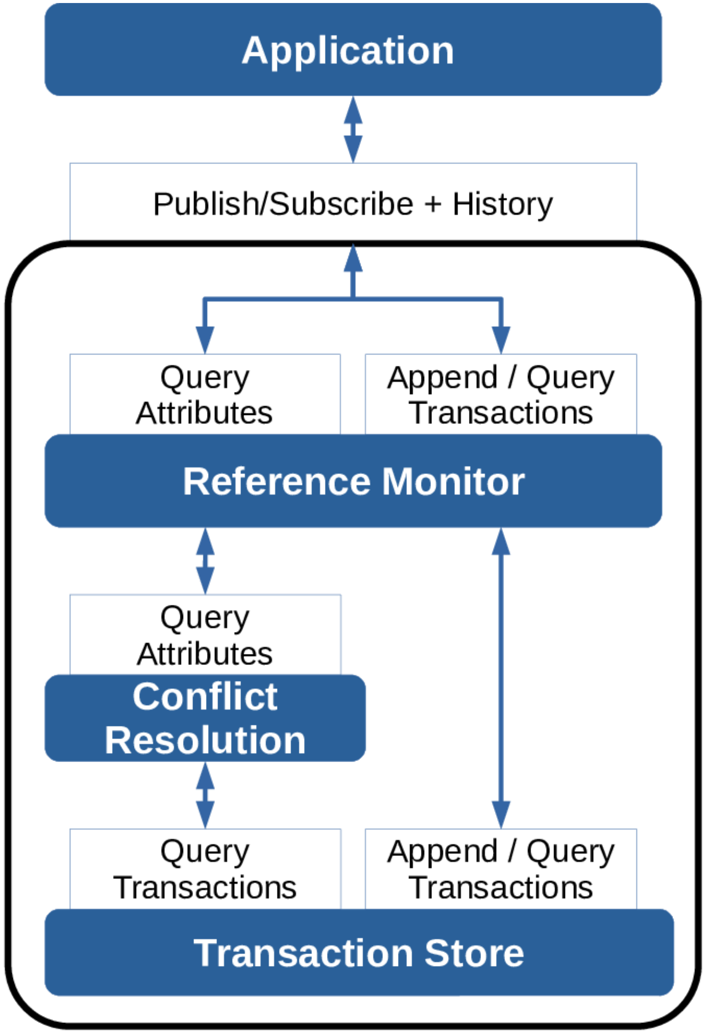
- Matrix idea: tied to the access control model
  - For concurrent transactions: Prefer subjects with higher level.

# How do we make it secure?

Define ideal functionality:

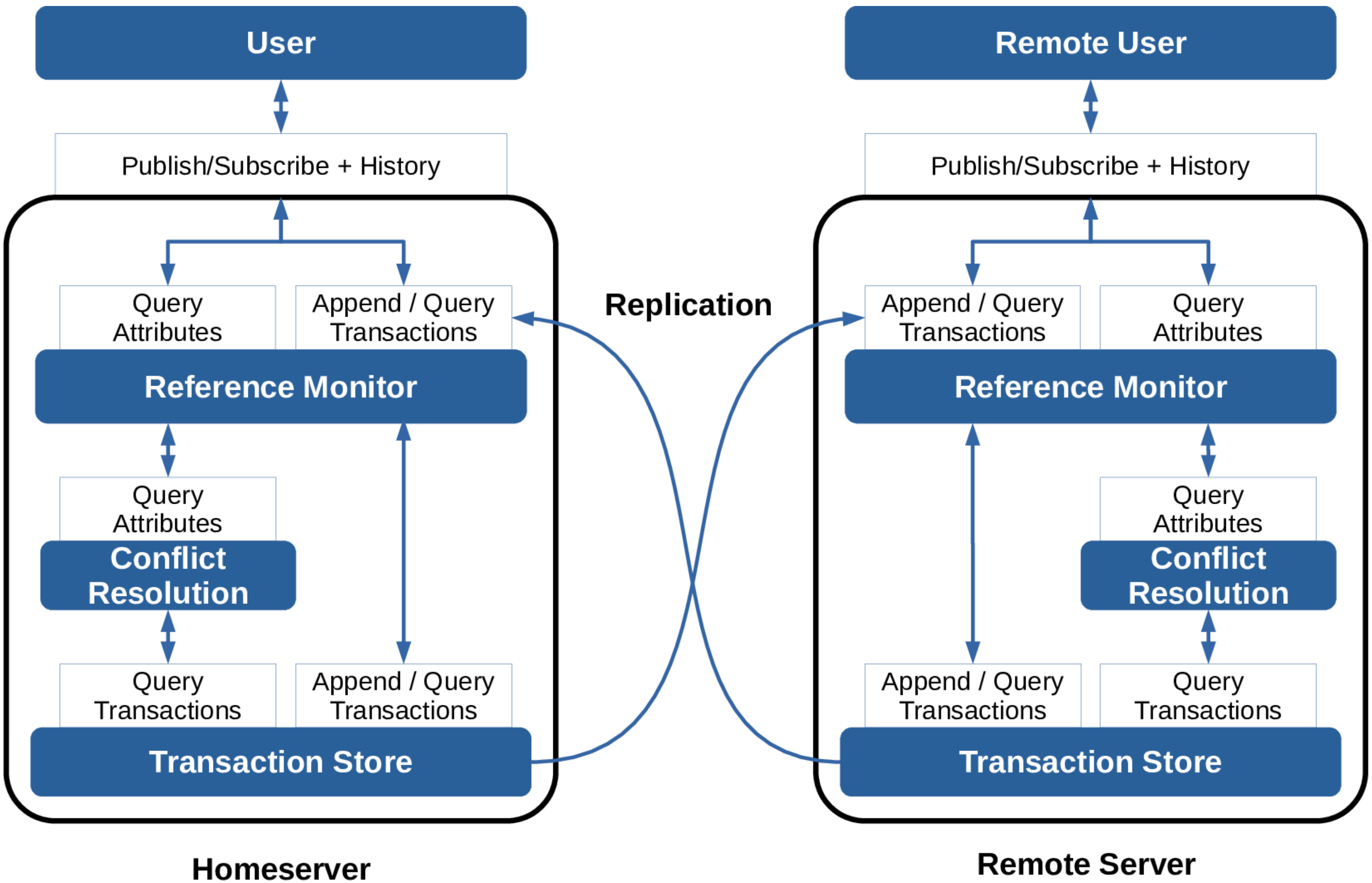
- Trusted Third Party provides
  - Reference Monitor
  - Conflict Resolution
  - Transaction Store
- …to potentially malicious users

- Decentralized implementation is secure if:
  - equivalent to central trusted third party for all honest subjects
  - regardless of the presence of an arbitrary number of byzantine faults



# Decentralized Implementation

Karlsruhe Institute of Technology



# Decomposition Result: When is there Equivalence?

**Requirements of the Matrix Approach:**

1) Out of two subjects, the one with the higher level is the ‘honest’ one.
2) Authorization policies and conflict resolution are deterministic and equivalently implemented by all ‘honest’ servers.
3) Whatever an attacker is doing: authorization policies and conflict resolution:
   - do not allow unauthorized transactions
   - do not allow unauthorized privilege escalation
   - always prefer the ‘honest’ subject.

- 1) is an axiomatic assumption
- ***2) & 3) are the main breakpoints of security***

# Security Assessment

## Fundamental Threats:

- Non-equivalence – targets assumption 2)
  - Diverging implementations → divergence from trusted third party model
- Incorrect specification – targets assumption 3)
  - authorization errors also present in the trusted third party model

- Found four practical security issues
  - Both threat categories affected
  - Mitigations now in place

## General Solution:

- Specify Conflict Resolution + Authorization Policies in Formal Calculus
  - For first threat: Generate equivalent code for all implementations
  - For second threat: Prove security and correctness

# Security Assessment: Discussion

- Access Control based on:
  - Eventual Consistency, Partial Order, No Finality, No Consensus?

- Behaves differently than traditional, consensus-based access control:
  - Matrix allows for "pluralism of opinions" on current state
    - Every server does its own, independent access control decision
    - instead of following the majority or an assigned leader
  - Agreement only if all honest servers exactly adhere to the protocol
  - No decision is ever final

► Crucial: Good understanding of consequences,
with regard to deployments in sensitive environments

► Security without Consensus nor Finality requires Formal Verification

# Summary

## Matrix Decomposition:

- **Level-based Access Control**
- **Event Graph**
  - Partial order, eventual consistency, no finality, no consensus
- **Conflict Resolution**
  - Topological Ordering
- Despite those weak guarantees: sound access control

## Security Analysis:

- Highly dependent on:
  - secure topological ordering
  - equivalent implementations
  - ► possible points of attack on concrete implementations

## Outlook:

- Formal verification of conflict resolution & authorization policies