www.csiro.au

A Collaborative Monitoring Mechanism for Making a Multitenant Platform Accountable

Chen Wang
CSIRO ICT Centre
Australia
chen.wang@csiro.au

Ying Zhou
The University of Sydney
Australia
ying.zhou@sydney.edu.au

CSIRO

Service Level Agreement (SLA)

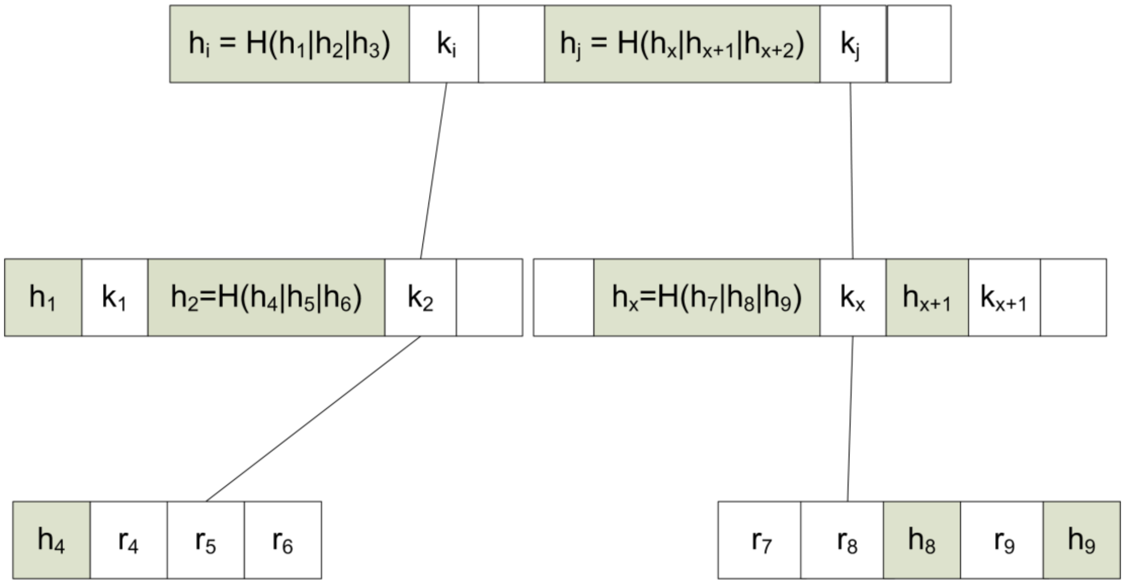
- **Service providers often offer Service Level Agreements as a means to address uncertainty**
  - Attempt to meet certain QoS metrics
  - Current status:
    - Only support very limited metrics
    - Not machine processable

- **Few means are provided to clients to make a SLA accountable when problems occur**
  - Monitoring is provided by the service provider itself

- **Clients are often required to furnish evidence all by themselves to be eligible to claim credit for a SLA violation**
  - Existing application design practice does not take into account of evidence collection functionalities for credit claiming purpose

Maintain the Data State in an External Accountability Service

- **The accountability service maintains a view of the data state**
  - Reflects what data should be from users’ perspective
  - Aggregates data updating requests of users to calculate the data state
  - Authenticates query results based on the calculated data state
- **Using Merkle B-tree[SIGMOD’06] for Data State Calculation**
  - A



Conclusions

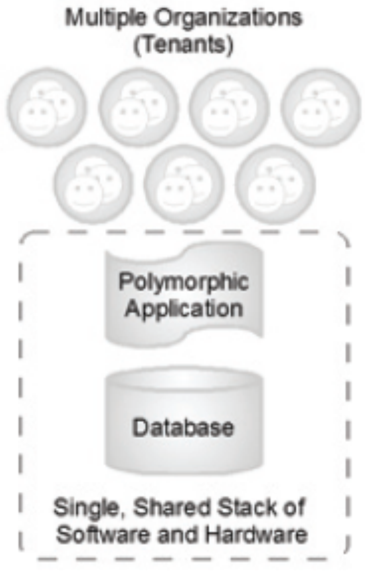
- **Accountability is one of the foundations that form realworld trust relationships.**
  - The capability of identifying a party that is responsible when things go wrong with evidences can potentially enhance the trustworthiness of a system.

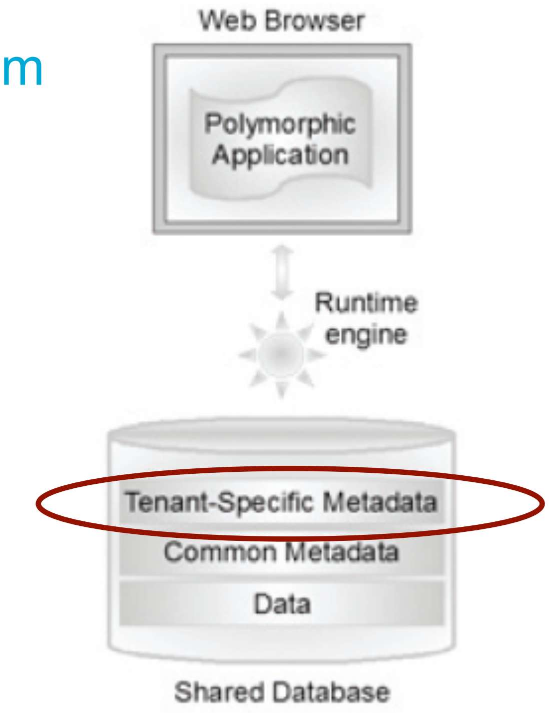
- **Accountability support is important to the cloud computing ecosystem**
  - Proper disclosure of important state transition logic of cloud services
  - External parties are capable of verifying the evidence that support these state transitions

Their slides

Cloud Services

- User data and applications are in a trend of moving to the cloud
  - Running on rented infrastructures (IaaS)
  - Using third party provisioned platform (PaaS) and software (SaaS)
  - Business logic is executed in different administrative domains on a pay-as-you-go basis
- Cloud service architecture sample:force.com


Multiple Organizations (Tenants)
Polymorphic Application
Database
Single, Shared Stack of Software and Hardware



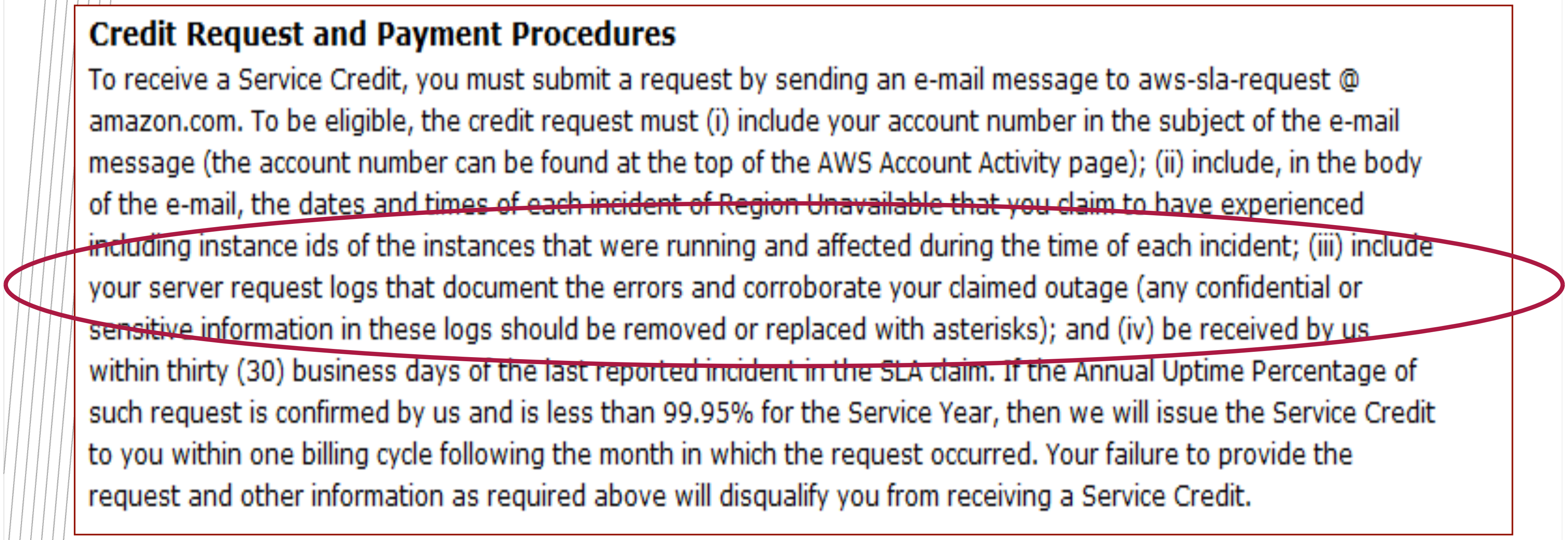
Cloud Service Uncertainty

- Performance variation is high
- Correctness of business logic
  - Data service providers in clouds often trade consistency for scalability.
    - Yahoo! PNUTS, Amazon SimpleDB consistency option
  - How can an application be sure that its query results satisfy its consistency constraint?
  - How to ensure the business logic is handled correctly?

Service Level Agreement (SLA)

- Service providers often offer Service Level Agreements as a means to address uncertainty
  - Attempt to meet certain QoS metrics
  - Current status:
    - Only support very limited metrics
    - Not machine processable
- Few means are provided to clients to make a SLA accountable when problems occur
  - Monitoring is provided by the service provider itself
- Clients are often required to furnish evidence all by themselves to be eligible to claim credit for a SLA violation
  - Existing application design practice does not take into account of evidence collection functionalities for credit claiming purpose

A Sample Service Level Agreement (Amazon EC2)



Credit Request and Payment Procedures

To receive a Service Credit, you must submit a request by sending an e-mail message to aws-sla-request @ amazon.com. To be eligible, the credit request must (i) include your account number in the subject of the e-mail message (the account number can be found at the top of the AWS Account Activity page); (ii) include, in the body of the e-mail, the dates and times of each incident of Region Unavailable that you claim to have experienced including instance ids of the instances that were running and affected during the time of each incident; (iii) include your server request logs that document the errors and corroborate your claimed outage (any confidential or sensitive information in these logs should be removed or replaced with asterisks); and (iv) be received by us within thirty (30) business days of the last reported incident in the SLA claim. If the Annual Uptime Percentage of such request is confirmed by us and is less than 99.95% for the Service Year, then we will issue the Service Credit to you within one billing cycle following the month in which the request occurred. Your failure to provide the request and other information as required above will disqualify you from receiving a Service Credit.

How is the problem tackled in realworld?

- The use of a trusted third party to make a deal
- The use of legal/social systems
  - Contract law provides incentives that promote good behaviour between parties
- Using these principles to make services accountable
  - **Disclosing** important state transition logic inside a service
  - **Collecting and managing evidence** based on a given SLA
  - Runtime **compliance check** and **problem detection**

CSIRO

Accountability

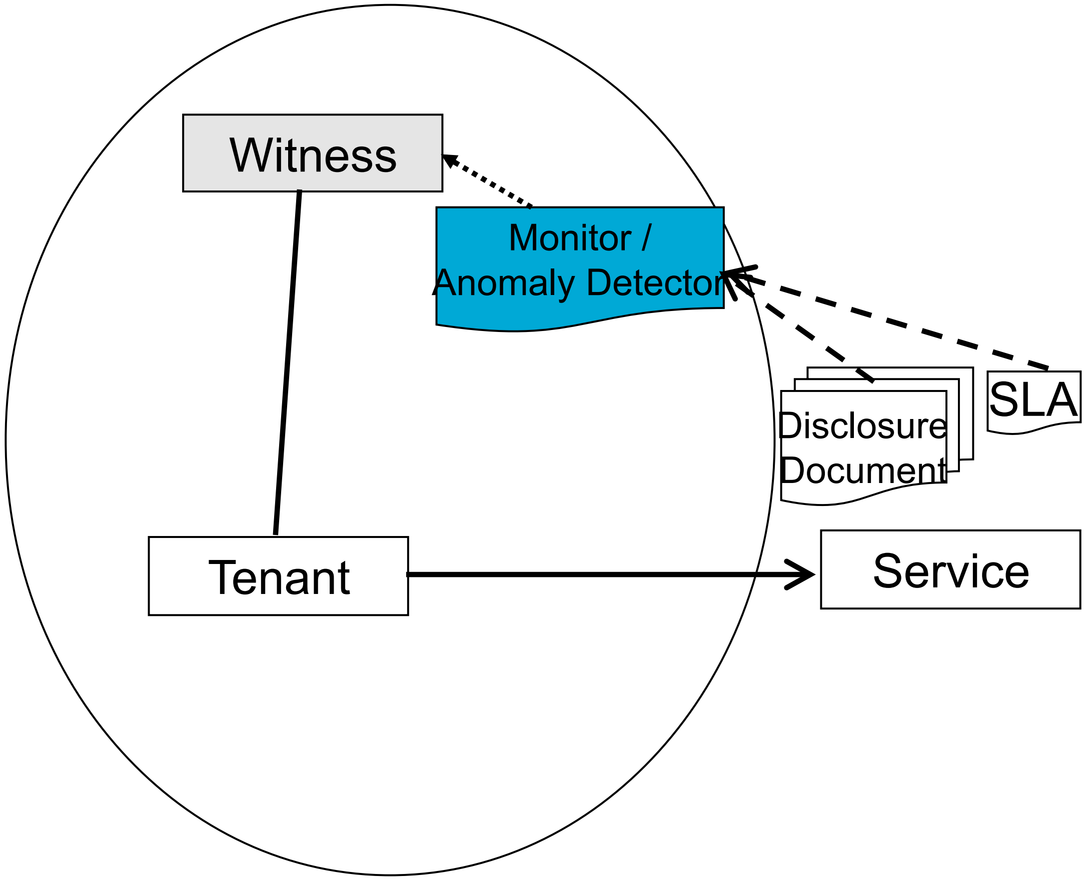
“Accountability is the ability to hold an entity, such as a person or organization, responsible for its actions.”

Butler Lampson:“Accountability and Freedom”, 2005

“People think that security in the real world is based on locks. In fact, realworld security depends mainly on deterrence, and hence on the possibility of punishment.”

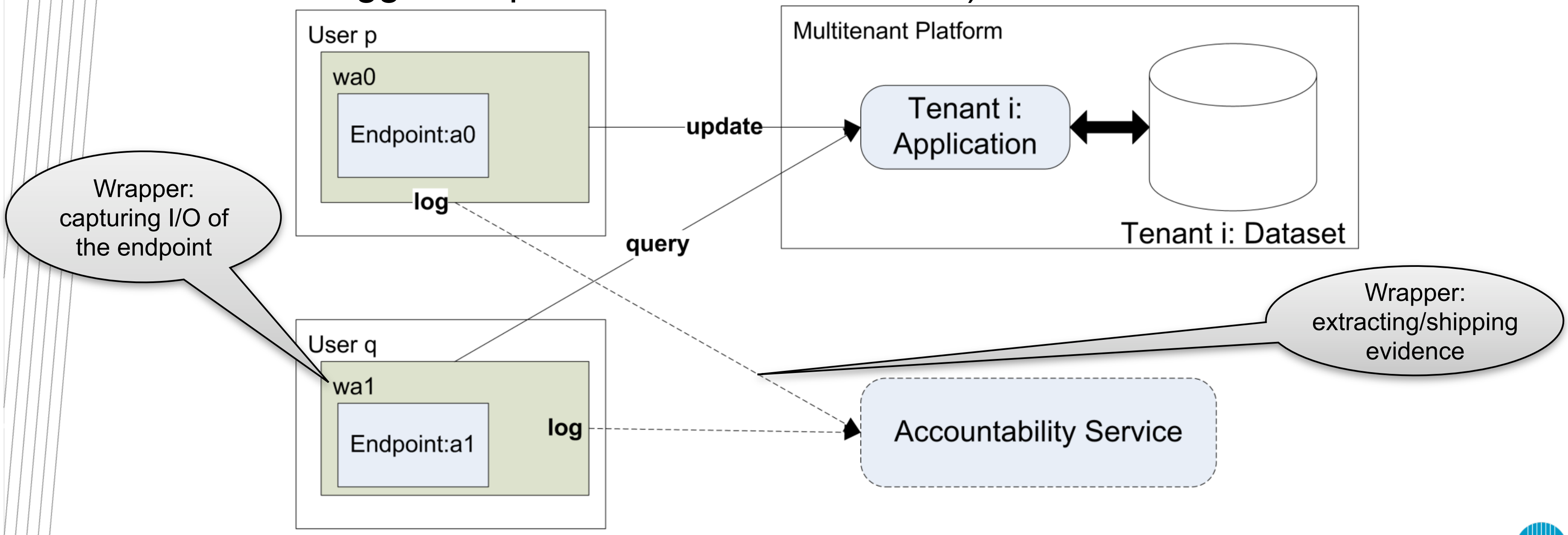
Butler Lampson: “Privacy and security - Usable security: how to get it.” CACM 52(11), 2009

Accountability as a Service: the Conceptual Model



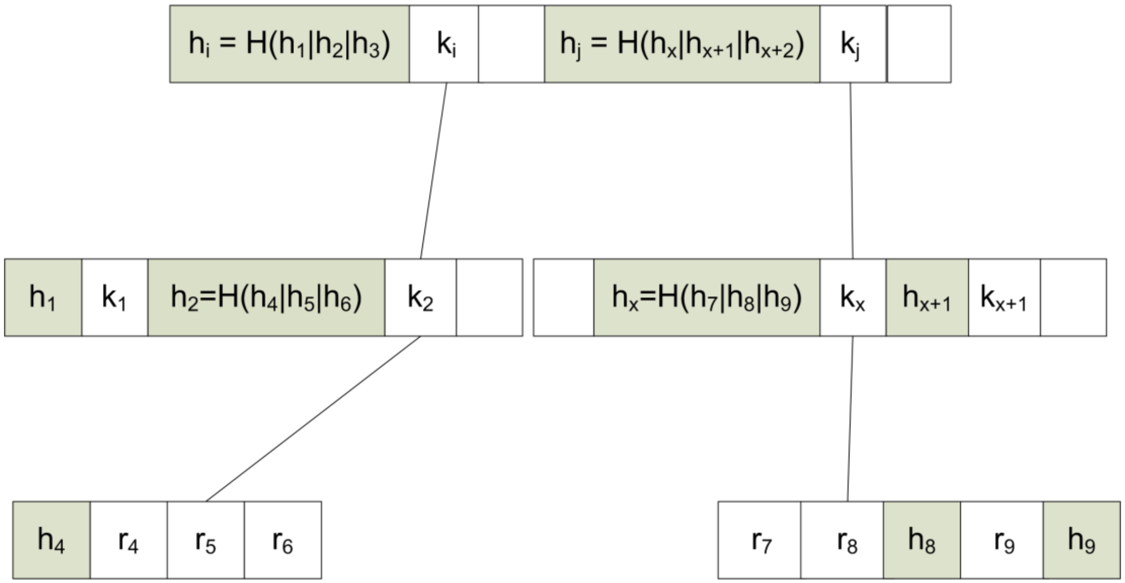
System Architecture

- A client has a set of applications running on shared data.
- These applications provide a set of endpoints.
- The SLA defines the following:
  - The data can only be accessed through these endpoints.
  - An endpoint is well-defined (the data state transition it may trigger is specified and deterministic).



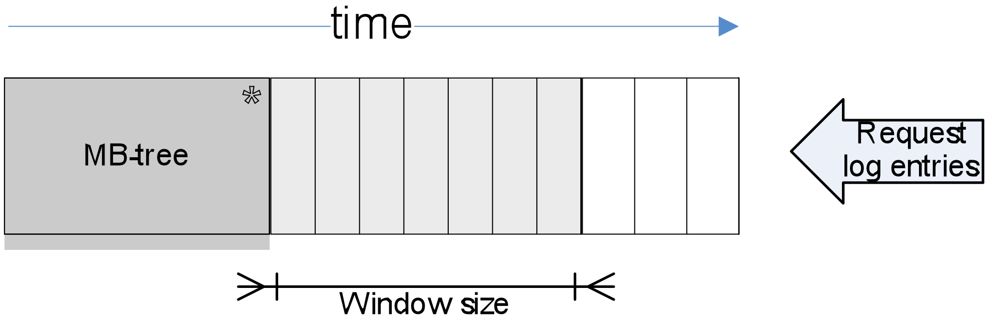
Maintain the Data State in an External Accountability Service

- The accountability service maintains a view of the data state
  - Reflects what data should be from users' perspective
  - Aggregates data updating requests of users to calculate the data state
  - Authenticates query results based on the calculated data state
- Using Merkle B-tree[SIGMOD'06] for Data State Calculation
  - A combination of Merkle (hash) tree and B+-tree



Consistency Issue

- The arrivals of request log entries to the accountability service (W) may be out of order
  - Solution: bind the update to the actual data and the view of state in W in a transaction → performance issue; negative impact of tightly coupling the actual service to W.
  - Trade accuracy of problem detection with performance and decoupling.
- Using a sliding window to sort out of order log entries
  - Eventual consistency between W and the cloud service it monitors

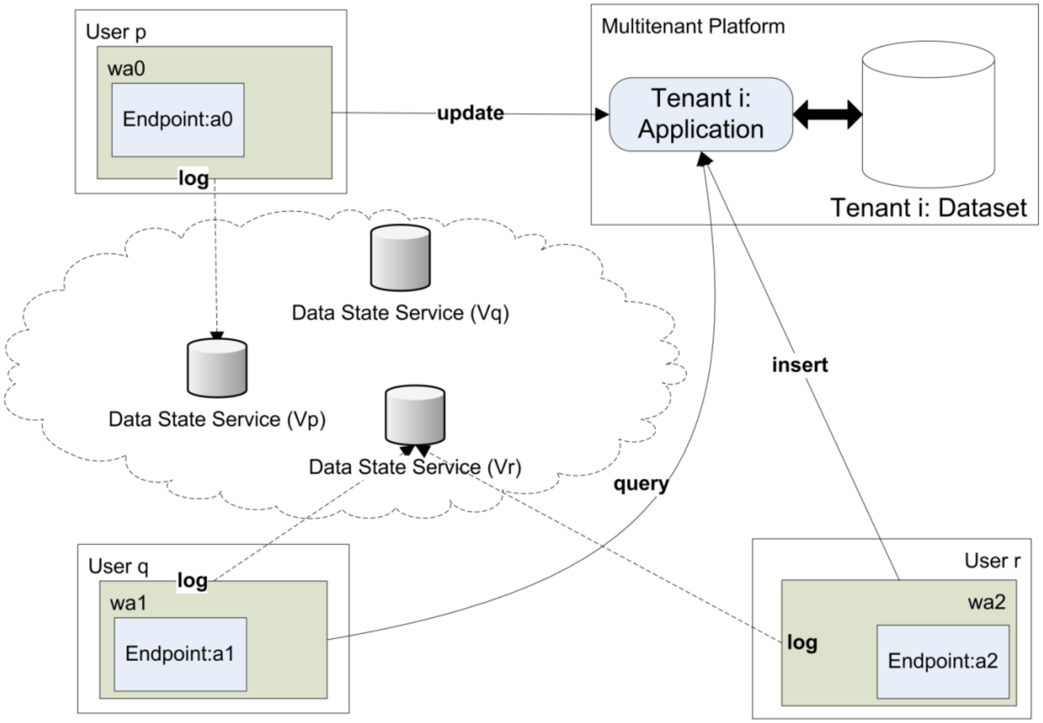


Replicate Data State among Multiple Accountability Services

Trustworthiness can be better achieved through the separation of responsibility.

The data state of a multitenant database is maintained by a set of data state services.

Each service maintains a view of the data state



The Organization of Multiple Accountability Services: Design Choices

- An update log entry is sent to any known state keeper and propagate to other state keepers in a synchronous manner
  - Strong consistency among data keepers
  - Poor logging performance
- An update log entry is sent to any known state keeper and propagate to other state keepers in an asynchronous manner
  - Weak consistency among data keepers
    - No guarantee that the answer to an authentication request will reflect the recent data state change
  - Good logging performance

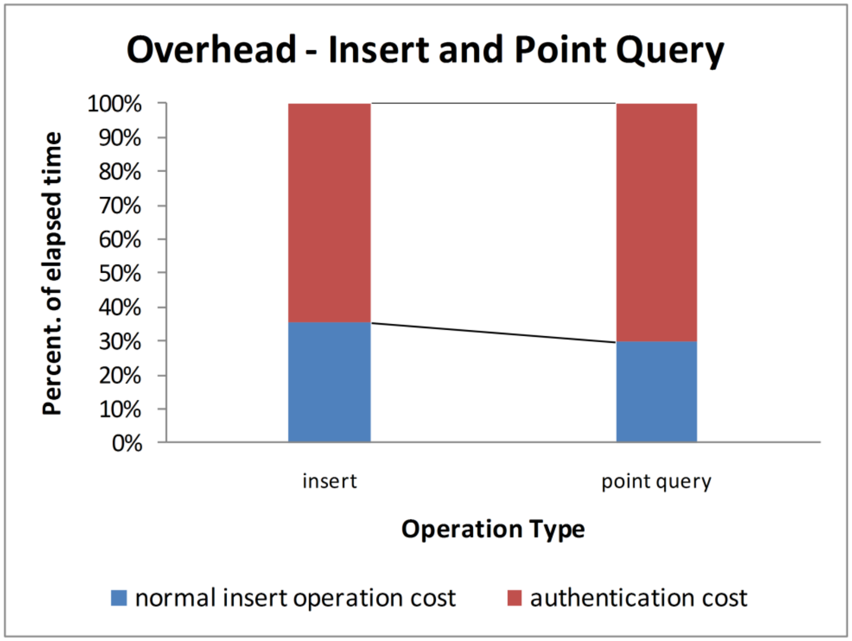
The Organization of Multiple Accountability Services: A Hybrid Approach

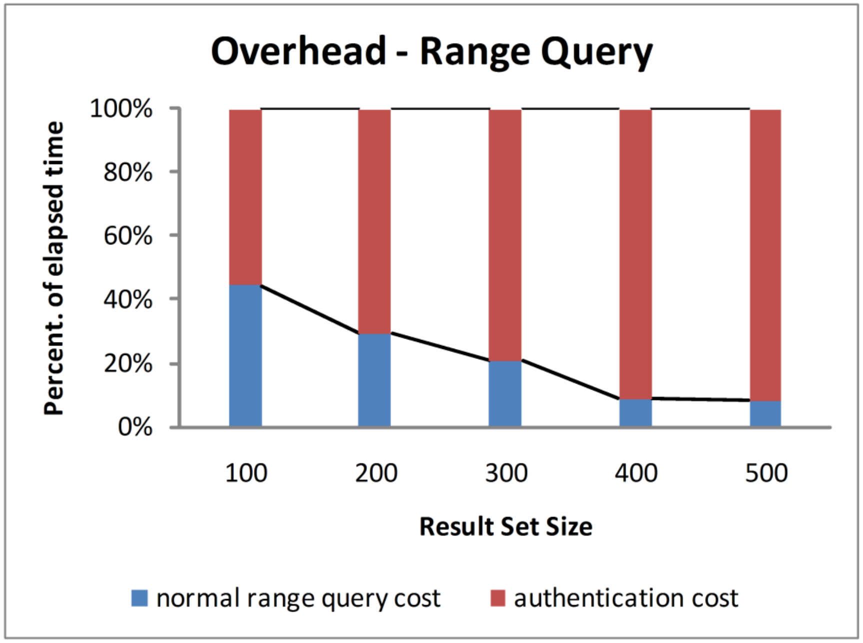
- Partition the whole range of the indexed attribute into a few non-overlapped regions
  - Each region is mapped to one or more state keepers.
  - An update to a key falling into certain region will be logged in the state keepers that are responsible for the region synchronously
  - The update is propagated to state keepers that are not responsible for the region asynchronously
- An authentication request is directed to a data keeper whose region overlaps most with the request data range
  - If the region covers the data range, authenticate the request immediately
  - Otherwise, wait for an allowable delay window for the update logs from other involved region to arrive

Evaluation

- Settings
  - A data management service
    - Contains Web service interfaces that map business logic into DB operations: inserts, point queries, and range queries.
    - gSOAP + MySQL
  - An accountability service
  - A few database clients
  - Each Party runs on an Amazon EC2 small instance (Linux version 2.6.21.7-2.fc8xen)
- Dataset
  - "Census Income" dataset from the UCI Machine Learning Repository
  - Indexed column is "fnlwgt"

Evaluation (cont)





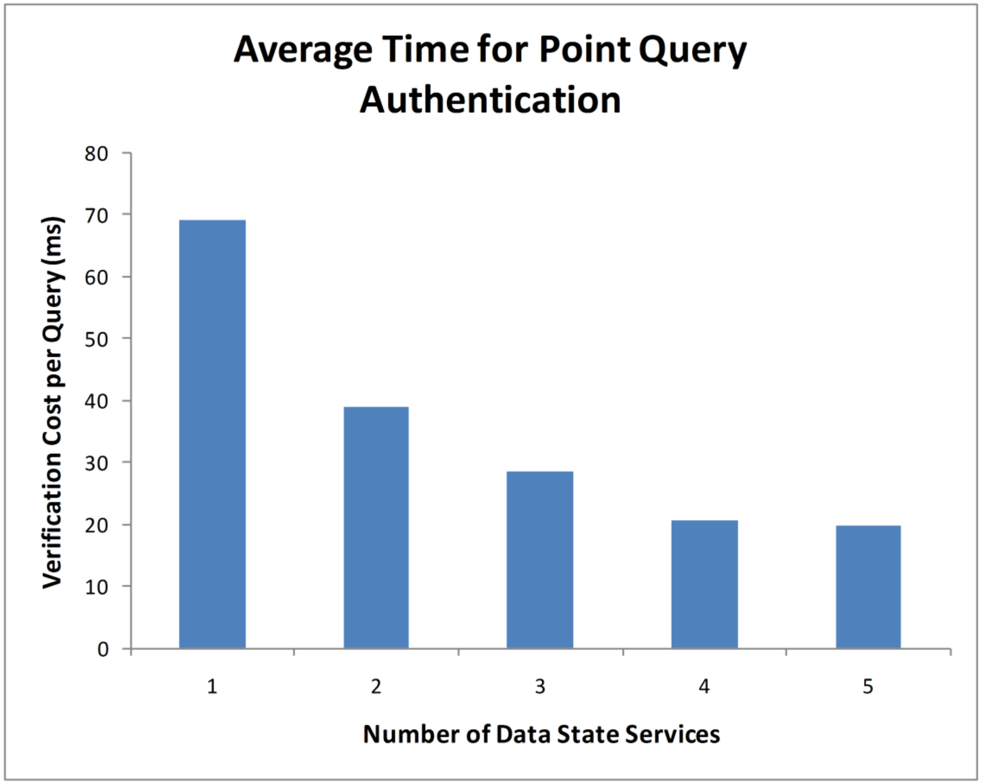
65 - 70% overhead of the total transaction time of a point query/insert (in synchronous mode)

The overhead is related to the result set size for range queries

The overhead can be reduced through asynchronous logging

Impact of Multiple Accountability Services on Authentication Time

- Performance gap between accountability and normal service
  - An accountability service faces scalability issue
- Maintaining a certain number of accountability services can reduce the performance loss



Conclusions

- Accountability is one of the foundations that form realworld trust relationships.
  - The capability of identifying a party that is responsible when things go wrong with evidences can potentially enhance the trustworthiness of a system.
- Accountability support is important to the cloud computing ecosystem
  - Proper disclosure of important state transition logic of cloud services
  - External parties are capable of verifying the evidence that support these state transitions