Project Presentation Group 7

Master Lab: Cloud Databases

Prof. Dr. Ruben Mayer | Chair for Application and Middleware Systems (I13) | Supervisor: Jawad Tahir | Students: Mohamed Elgamal, Nico Denner, Ahmed Mahmoud
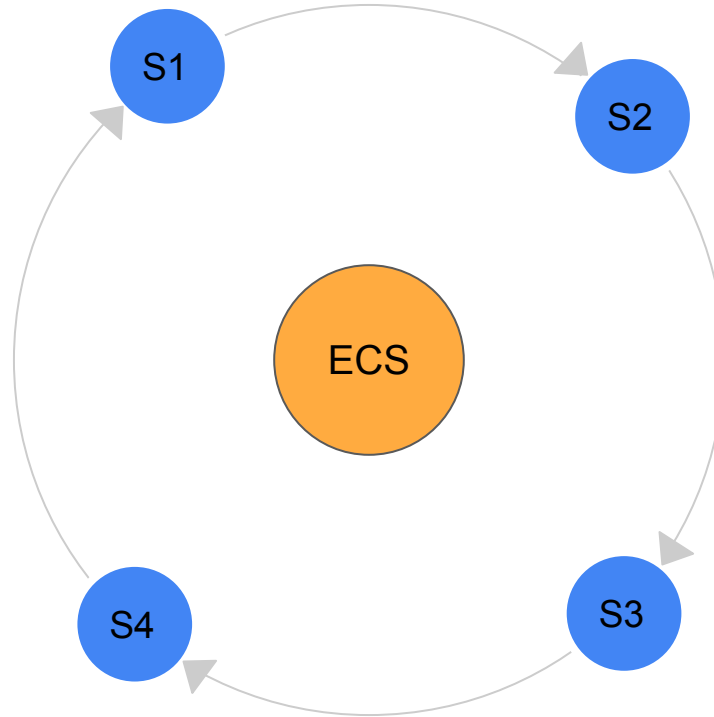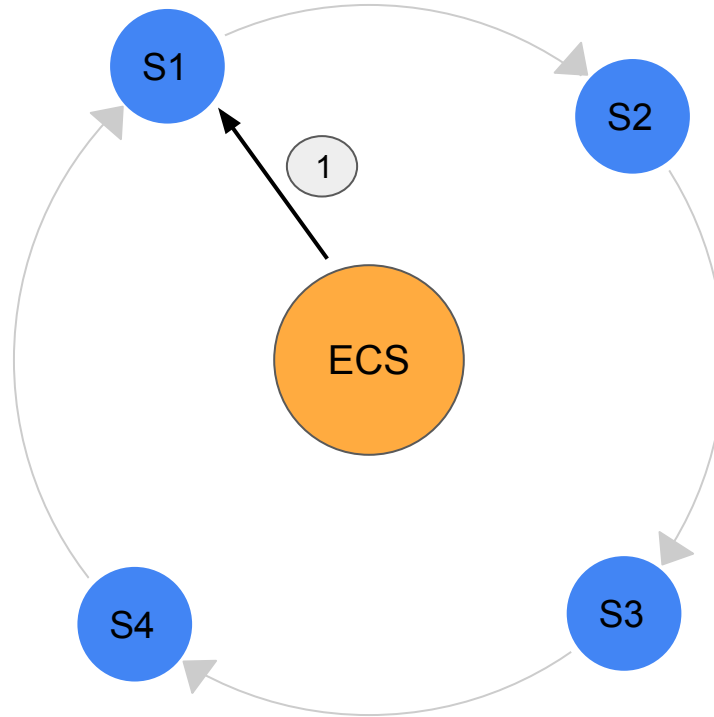
System Overview

Novel Extensions:
1. Gossip Protocol
2. Strong Consistency
3. Client Authentication
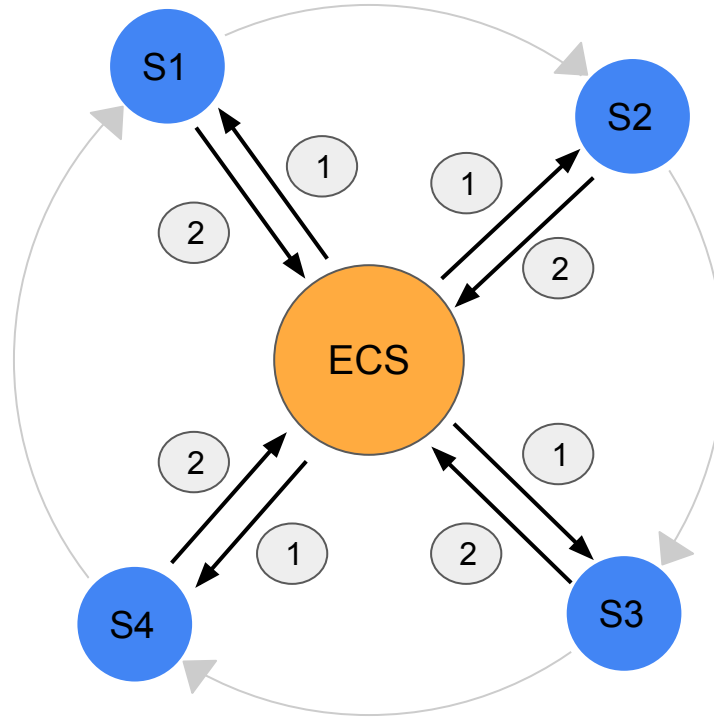
Heartbeat

Heartbeat
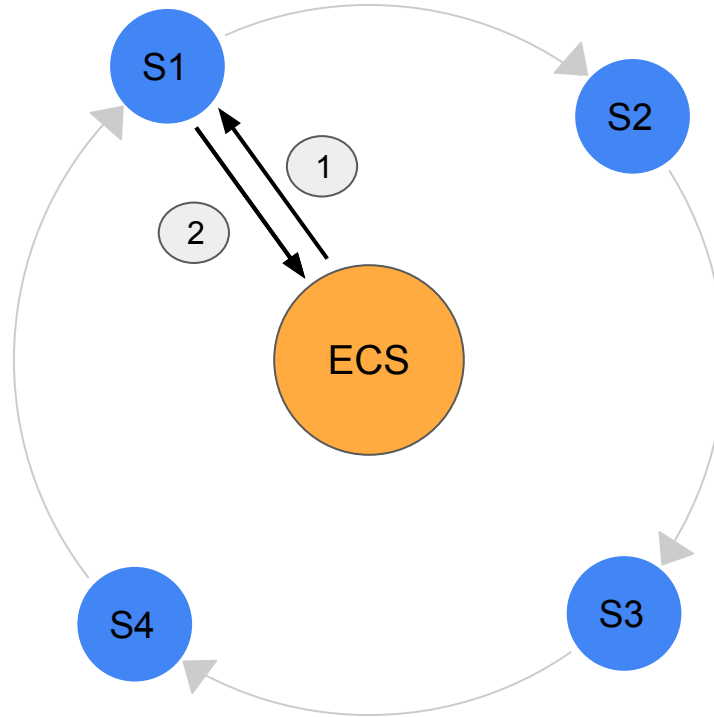
Heartbeat
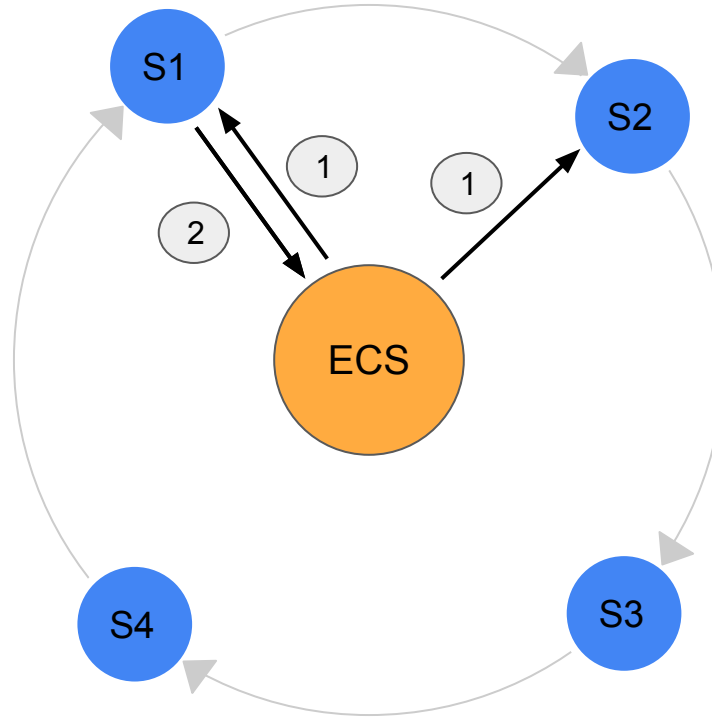
Heartbeat
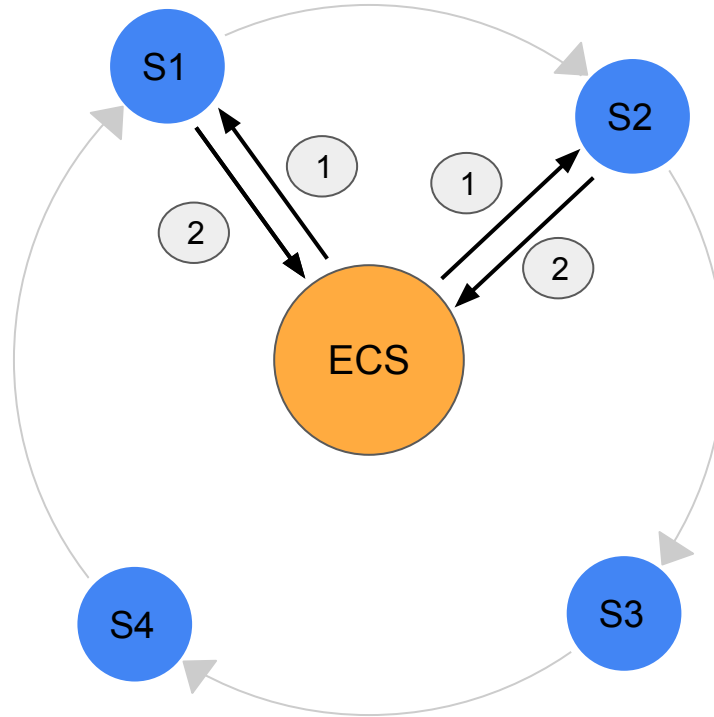
Heartbeat

Heartbeat
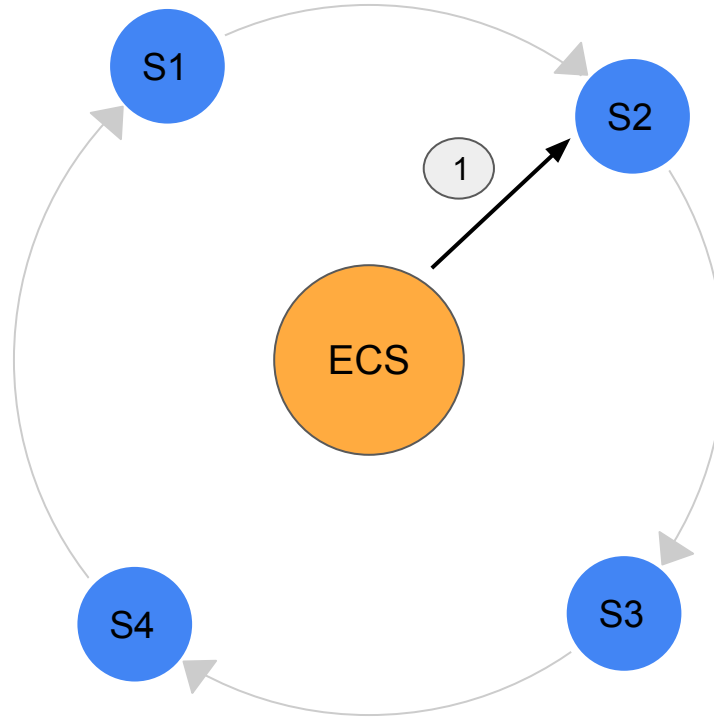
Heartbeat
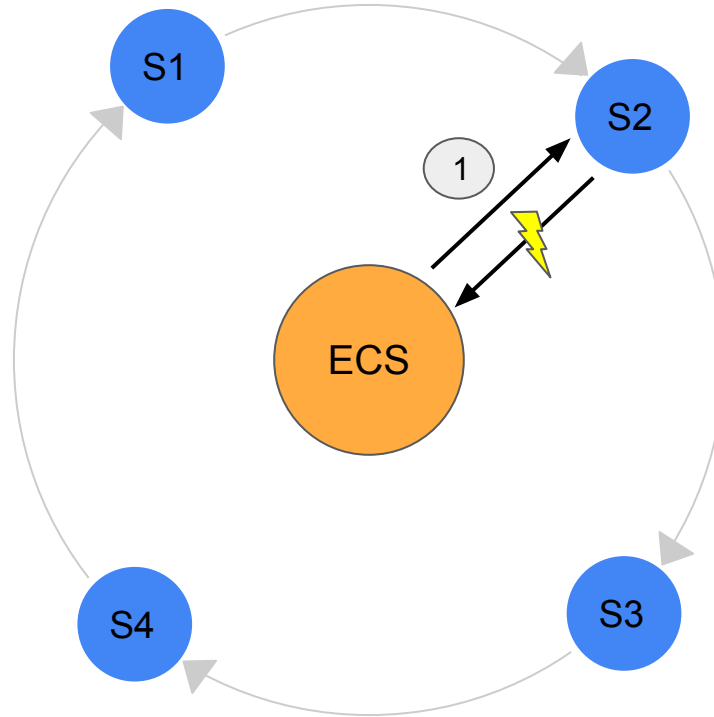
Heartbeat
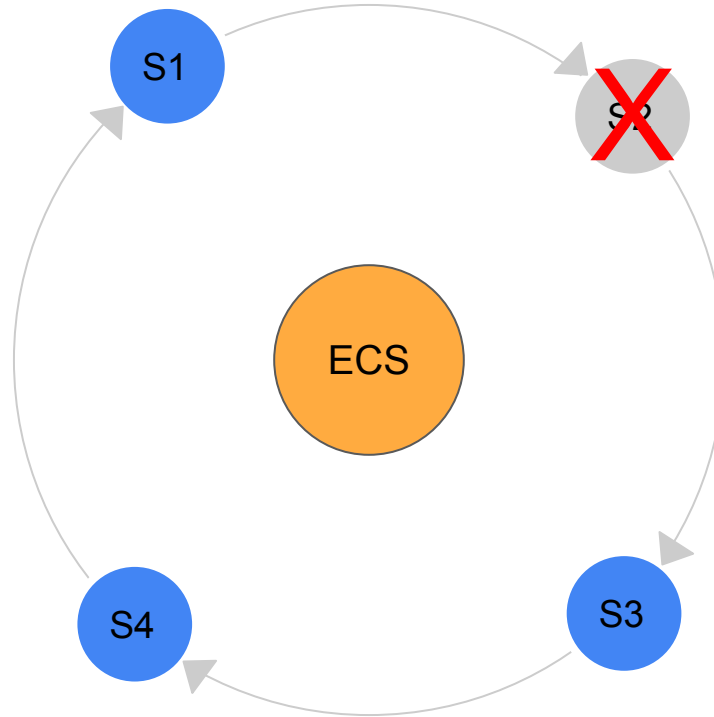
Heartbeat

Heartbeat
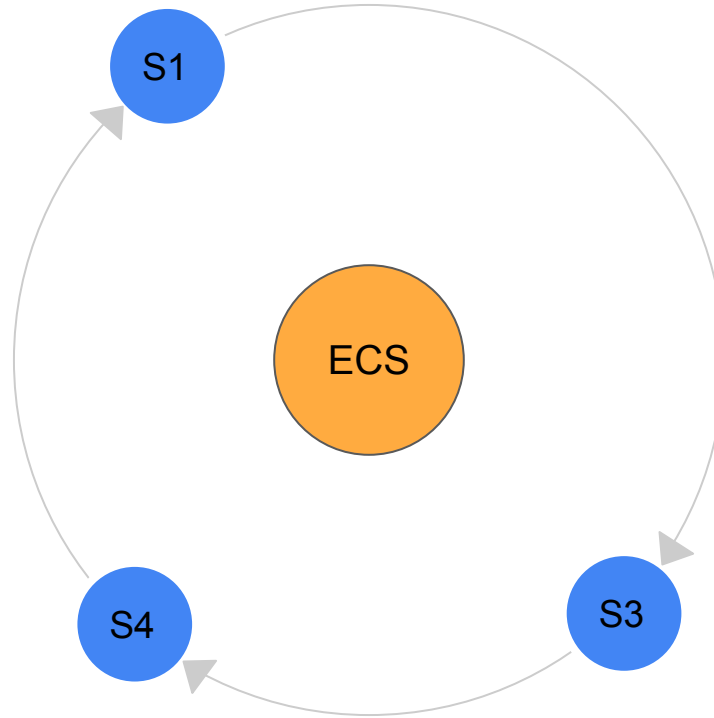
Gossip Protocol
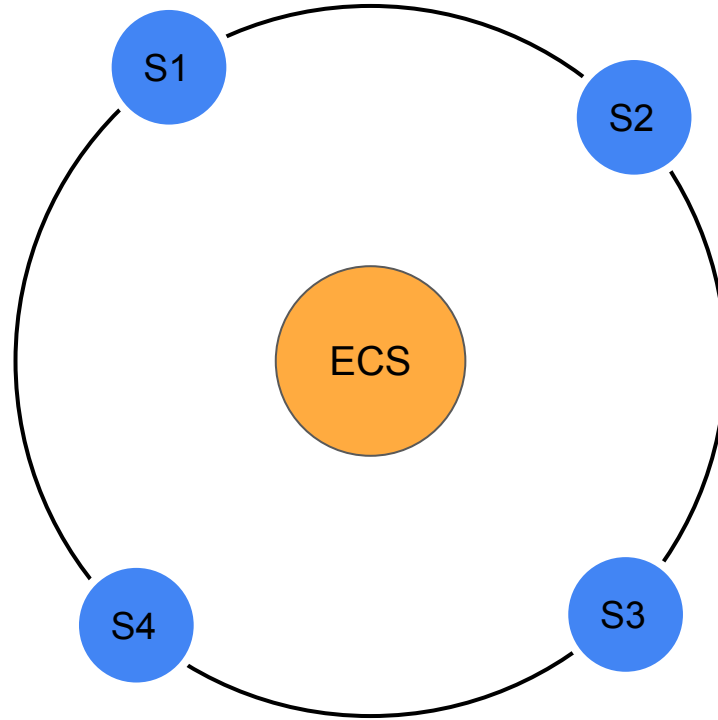
Gossip Protocol
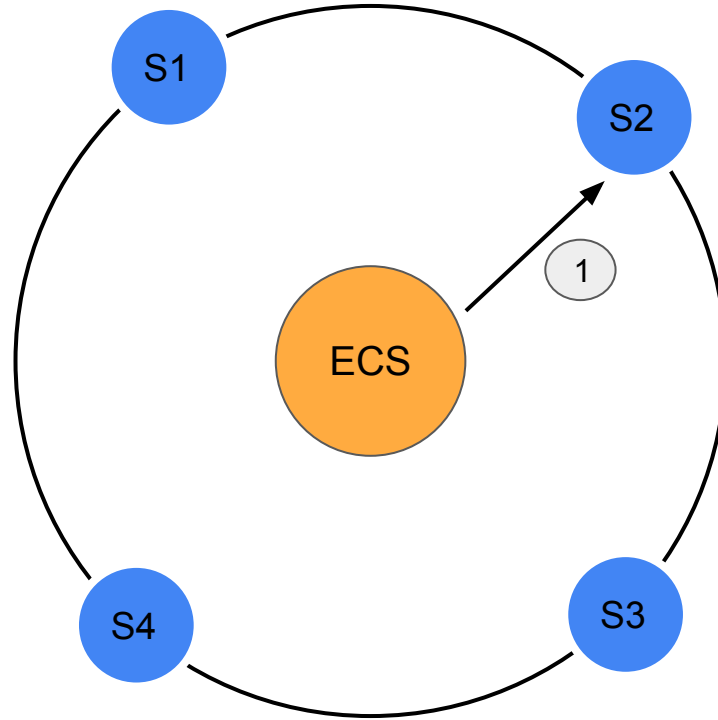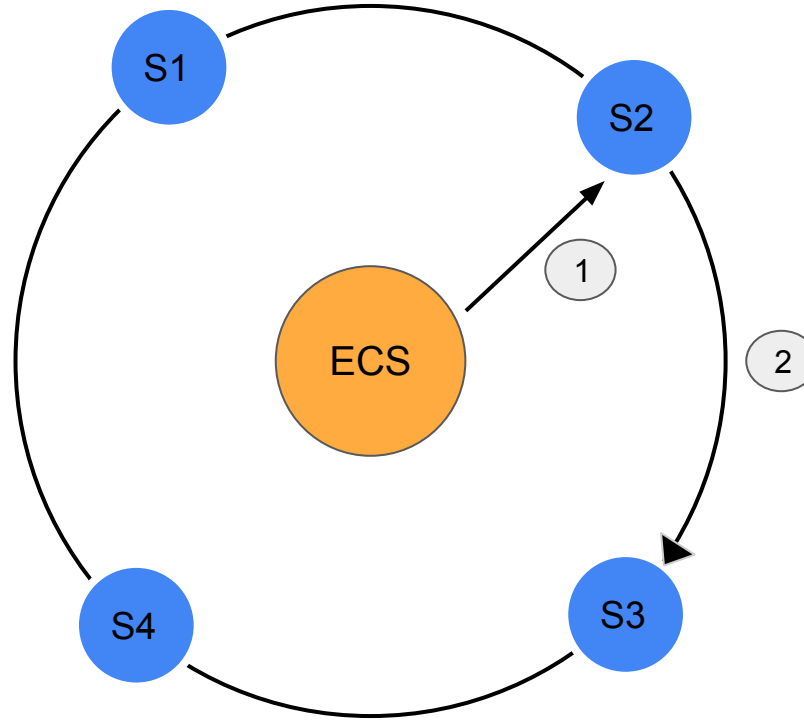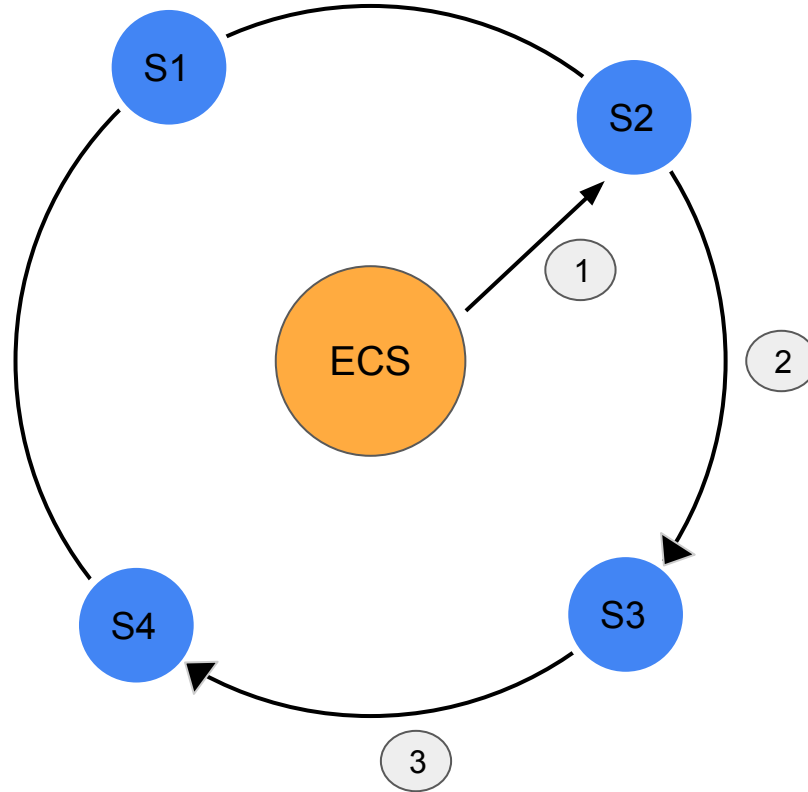
Gossip Protocol

Gossip Protocol
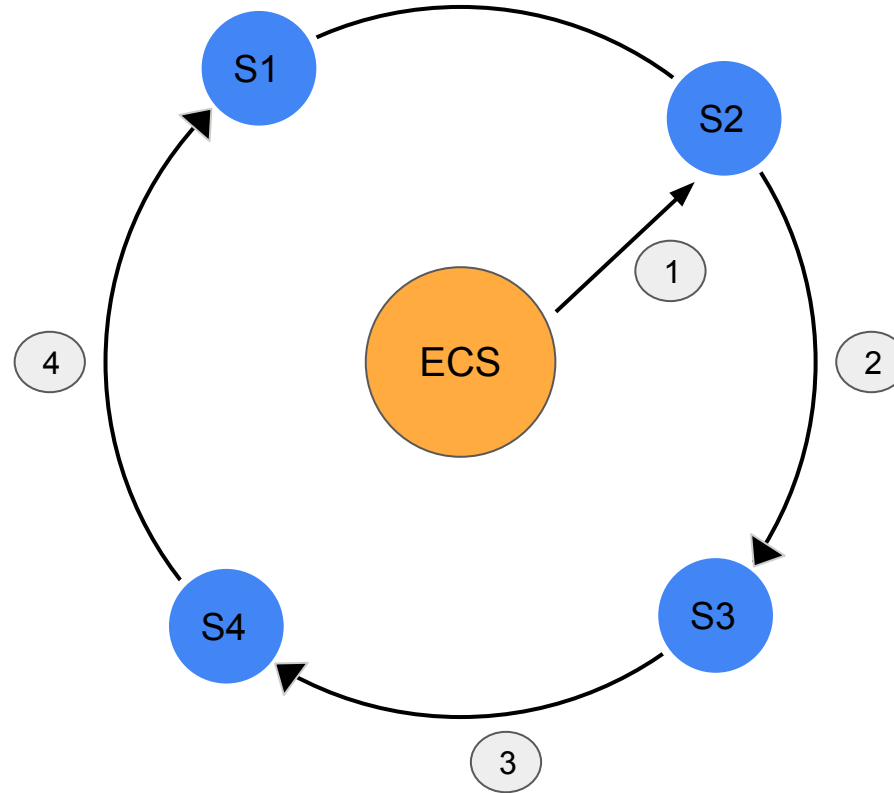
Gossip Protocol
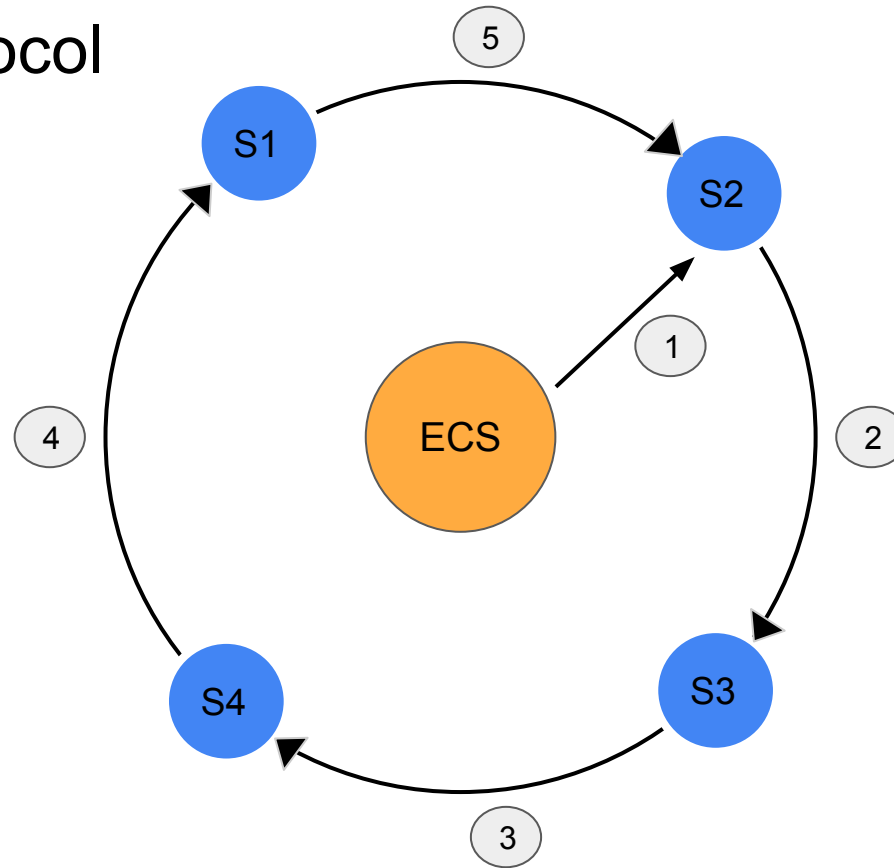
Gossip Protocol
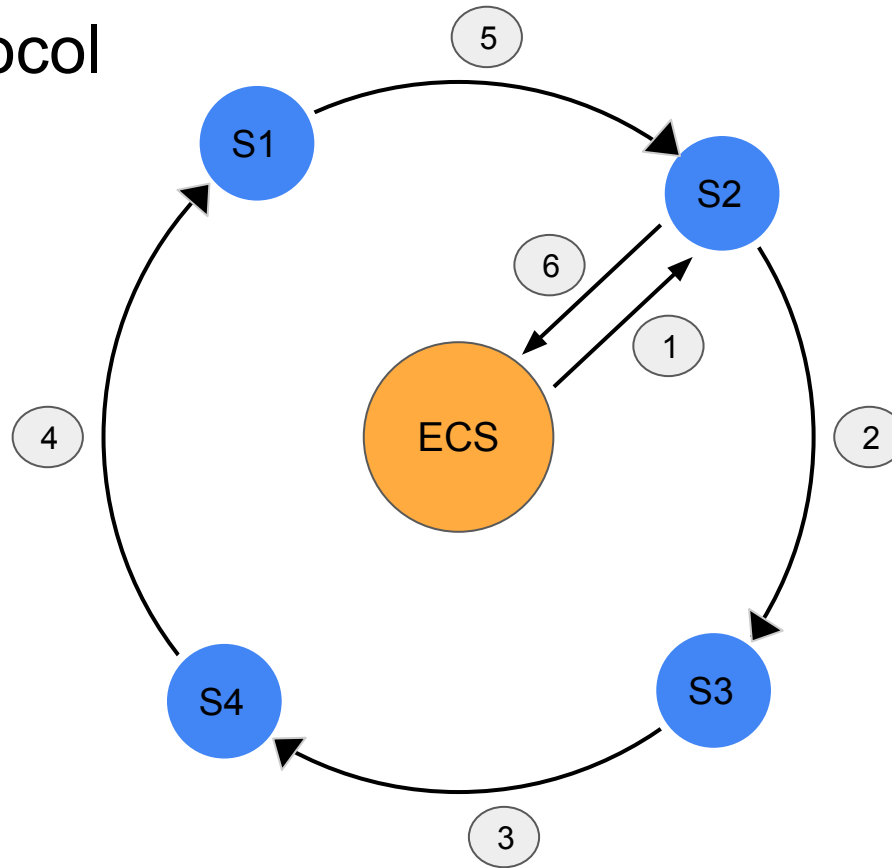
Gossip Protocol
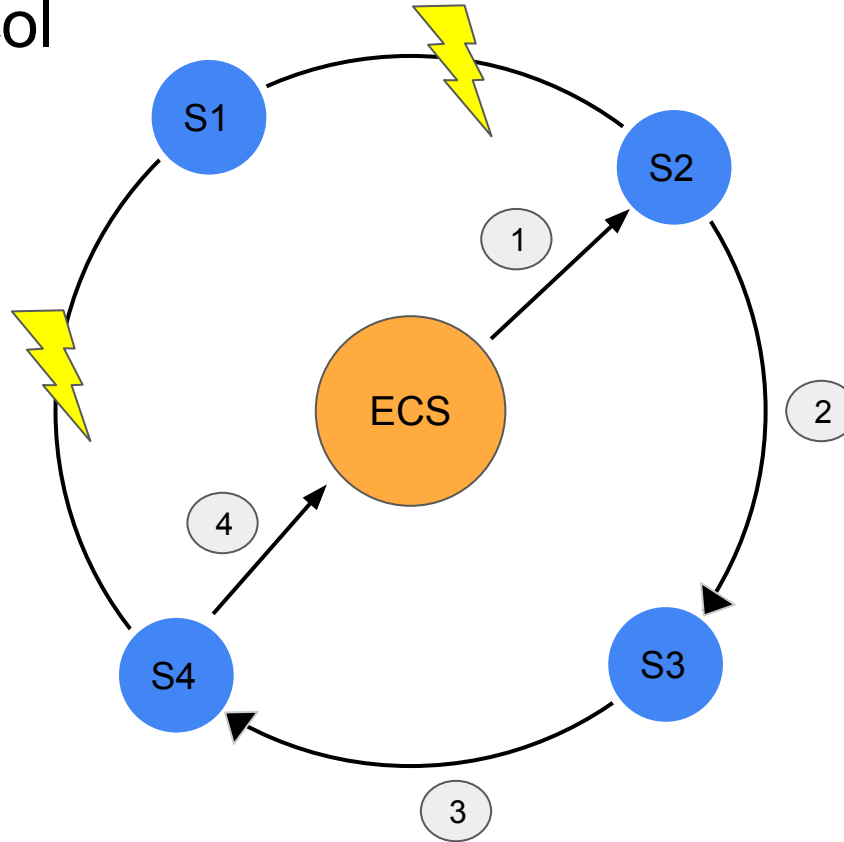
Gossip Protocol
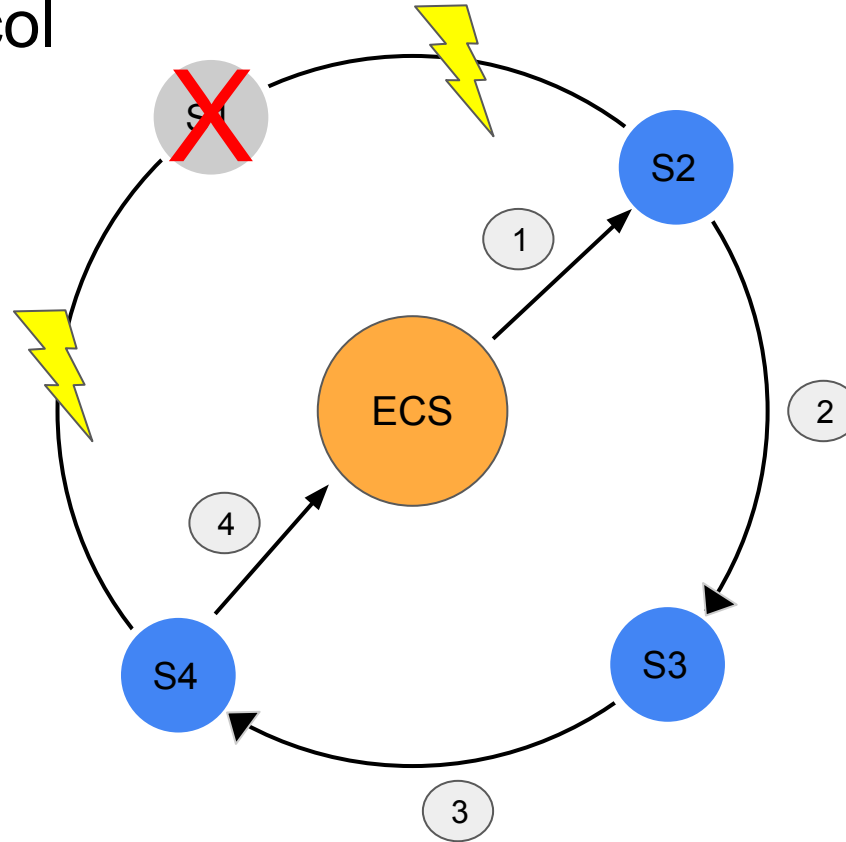
Gossip Protocol

Gossip Protocol

Strong consistency

Eventual Consistency

Eventual Consistency is a guarantee that when an **update** is made in a distributed database, that update will **eventually** be reflected in all nodes that store the data, resulting in the same response every time the data is queried [1].

[1] https://www.scylladb.com/glossary/eventual-consistency/

Strong consistency

Eventual Consistency

Eventual Consistency is a guarantee that when an **update** is made in a distributed database, that update will **eventually** be reflected in all nodes that store the data, resulting in the same response every time the data is queried [1].
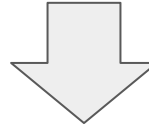
Strong Consistency

Strong Consistency simply means the data must be **strongly consistent** at all times. All the server nodes across the world should contain the **same value** as an entity at any point in time [2].

[1] https://www.scylladb.com/glossary/eventual-consistency/
[2] https://www.geeksforgeeks.org/eventual-vs-strong-consistency-in-distributed-databases/
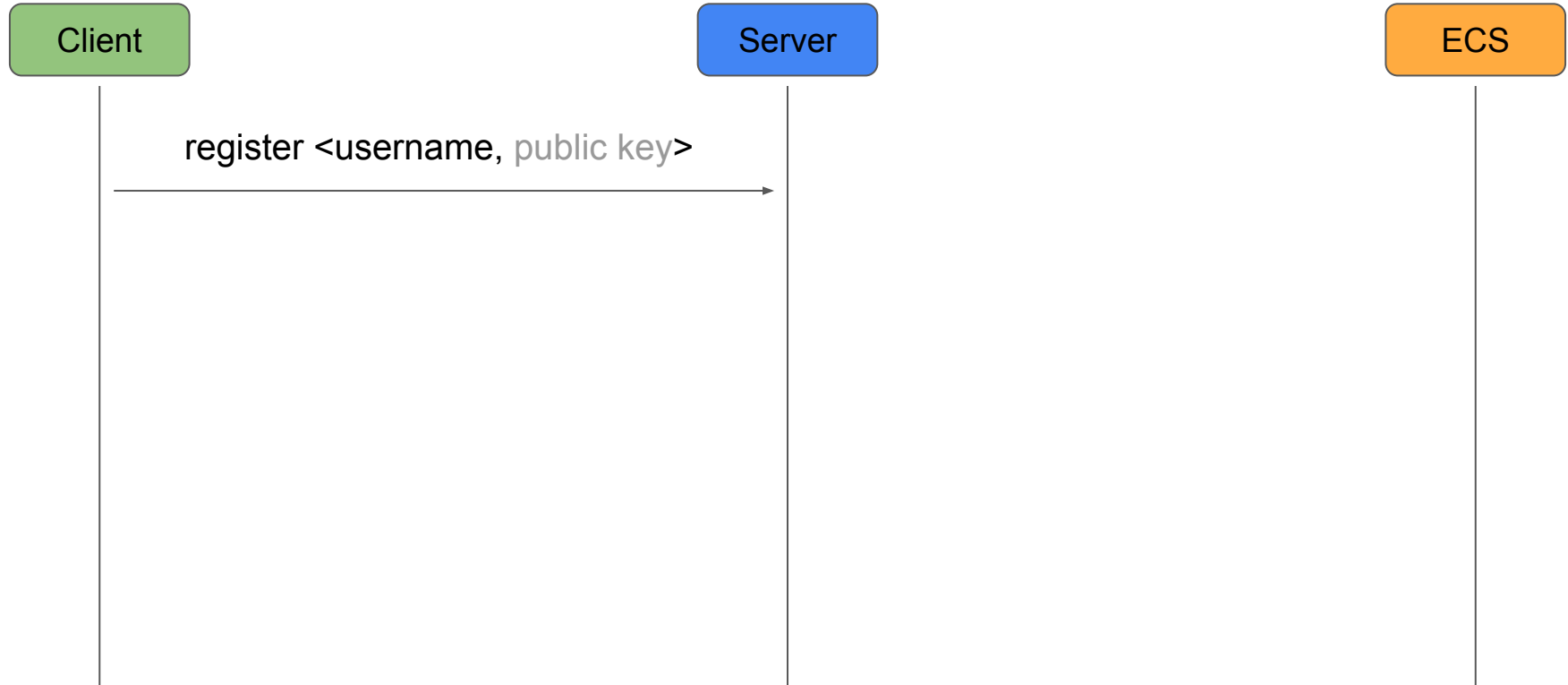
Client Authentication - Register

Client

Server

ECS

Client Authentication - Register

Client   Server   ECS

register <username, public key> →

Client Authentication - Register

Client Authentication - Register

Client   Server   ECS

register <username, public key>

success

add <username, public key>

Client Authentication - Register

```
Client                          Server                                    ECS

  |                                |                                        |
  |   register <username, public key>                                       |
  |------------------------------->|                                        |
  |                                |                                        |
  |           success              |                                        |
  |<-------------------------------|                                        |
  |                                |                                        |
  |                                |   add <username, public key>           |
  |                                |--------------------------------------->|
  |                                |                                        |
```

ECS broadcasts new user to all servers
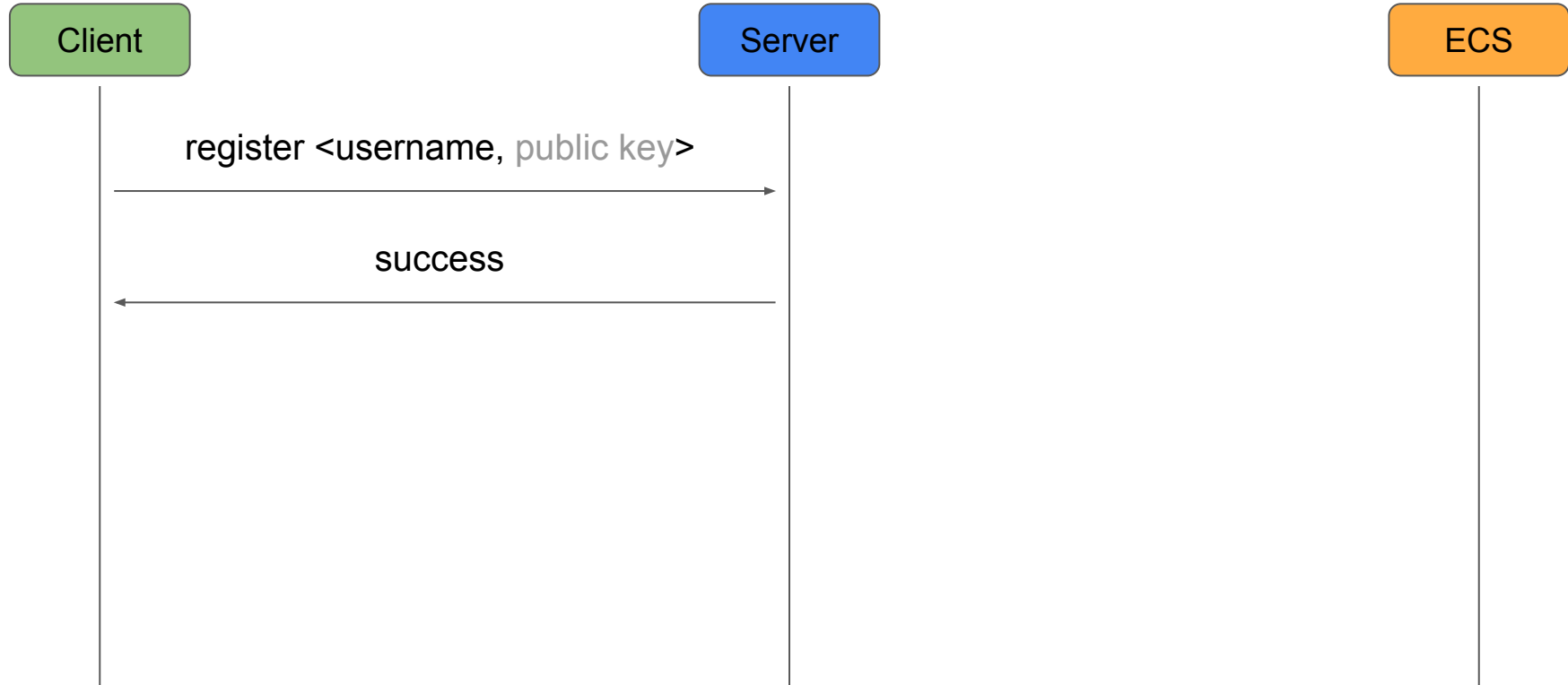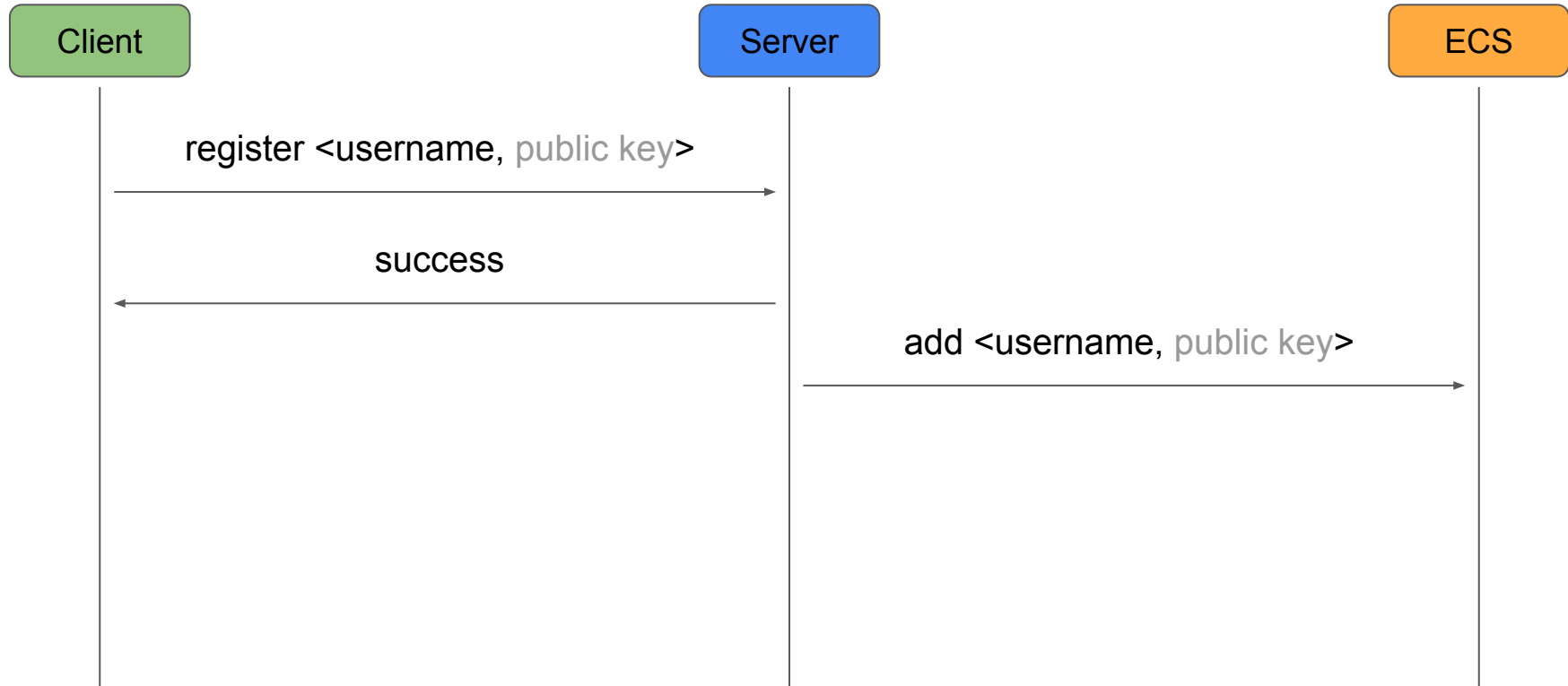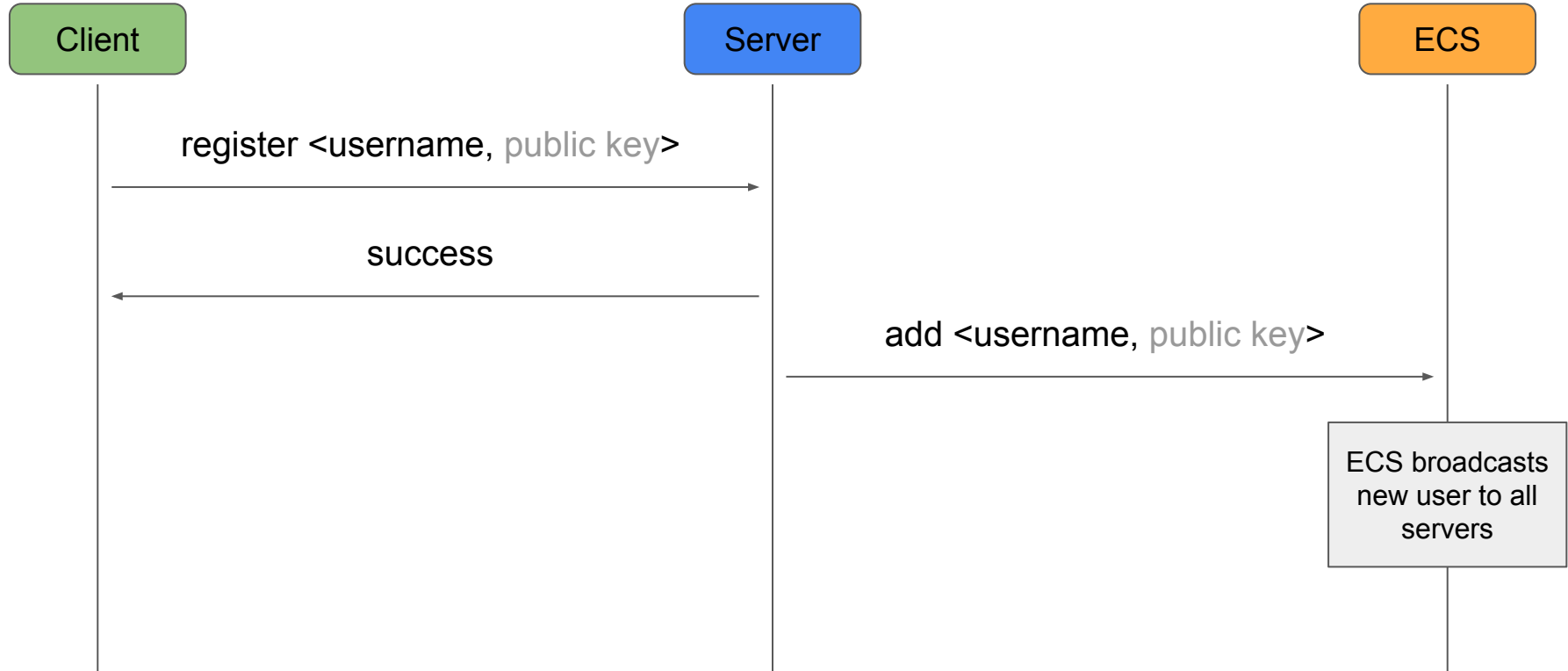
Client Authentication - Login

Client

Server

Client Authentication - Login

Client

Server

login <username>

Client Authentication - Login

Client

Server

login <username>

decrypt_this <random_encrypted_message>

Client Authentication - Login

```
Client                                                              Server

          login <username>
   ──────────────────────────────────────────────────────────────────▶

          decrypt_this <random_encrypted_message>
   ◀──────────────────────────────────────────────────────────────────

          check <decrypted_message>
   ──────────────────────────────────────────────────────────────────▶
```

Client Authentication - Login

Client   Server

login <username>

decrypt_this <random_encrypted_message>

check <decrypted_message>

login_success

Client Authentication - Login

Client → Server: login <username>

Server → Client: decrypt_this <random_encrypted_message>

Client → Server: check <decrypted_message>

Server → Client: login_success
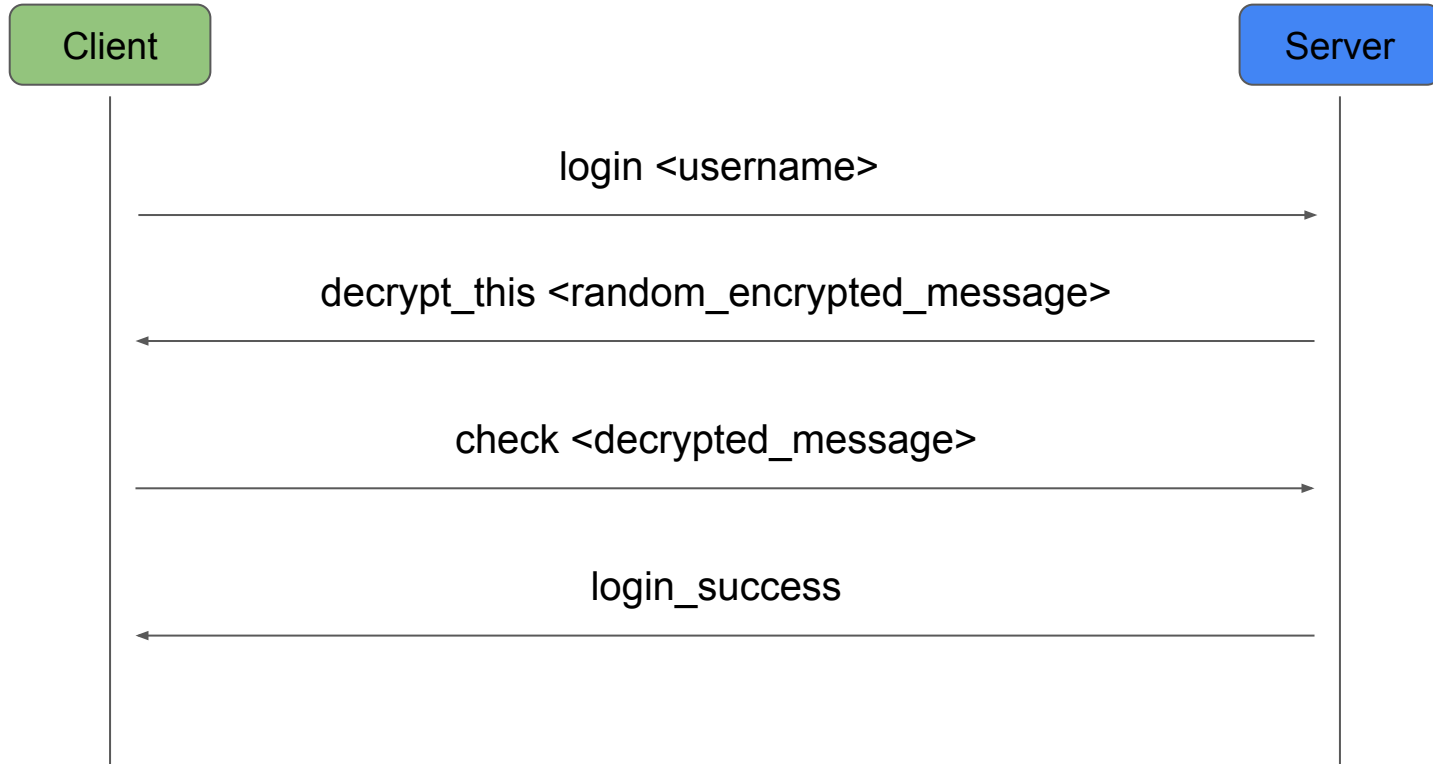
Video Demo

Performance Evaluation - Insertion

Insertion Completion
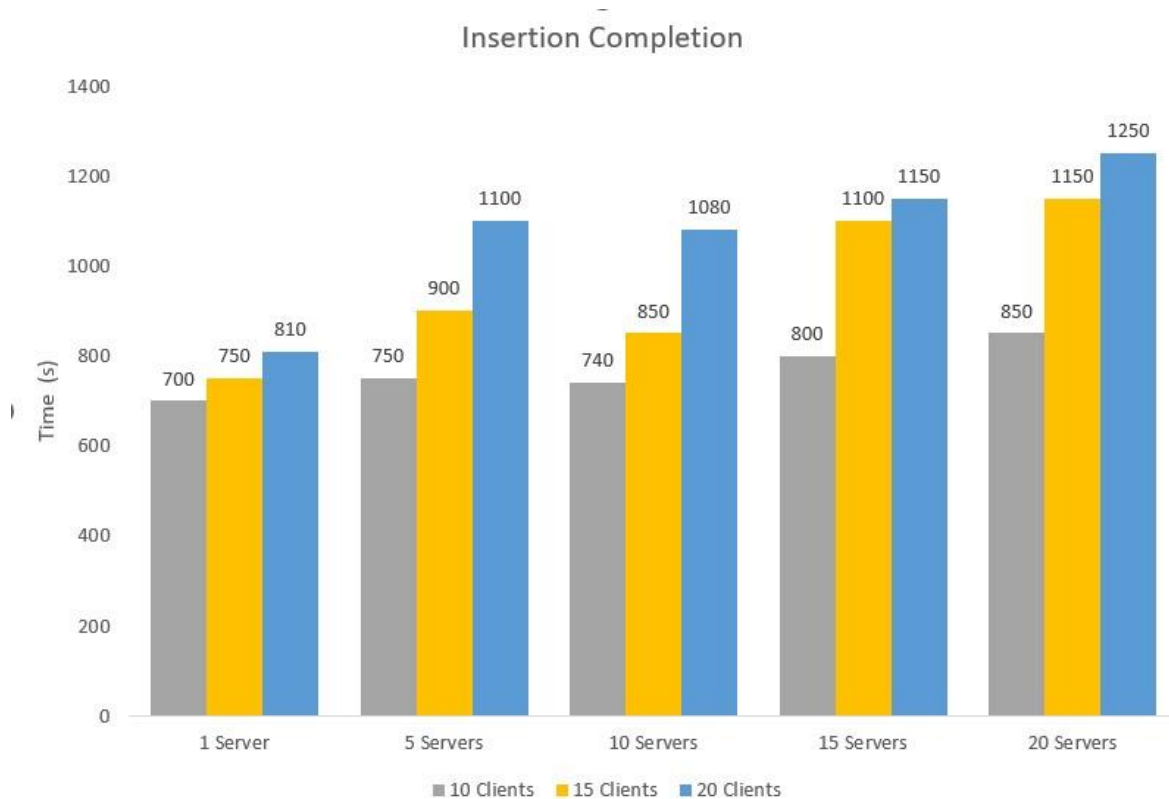
Performance Evaluation - Retrieval

Retrieval Completion
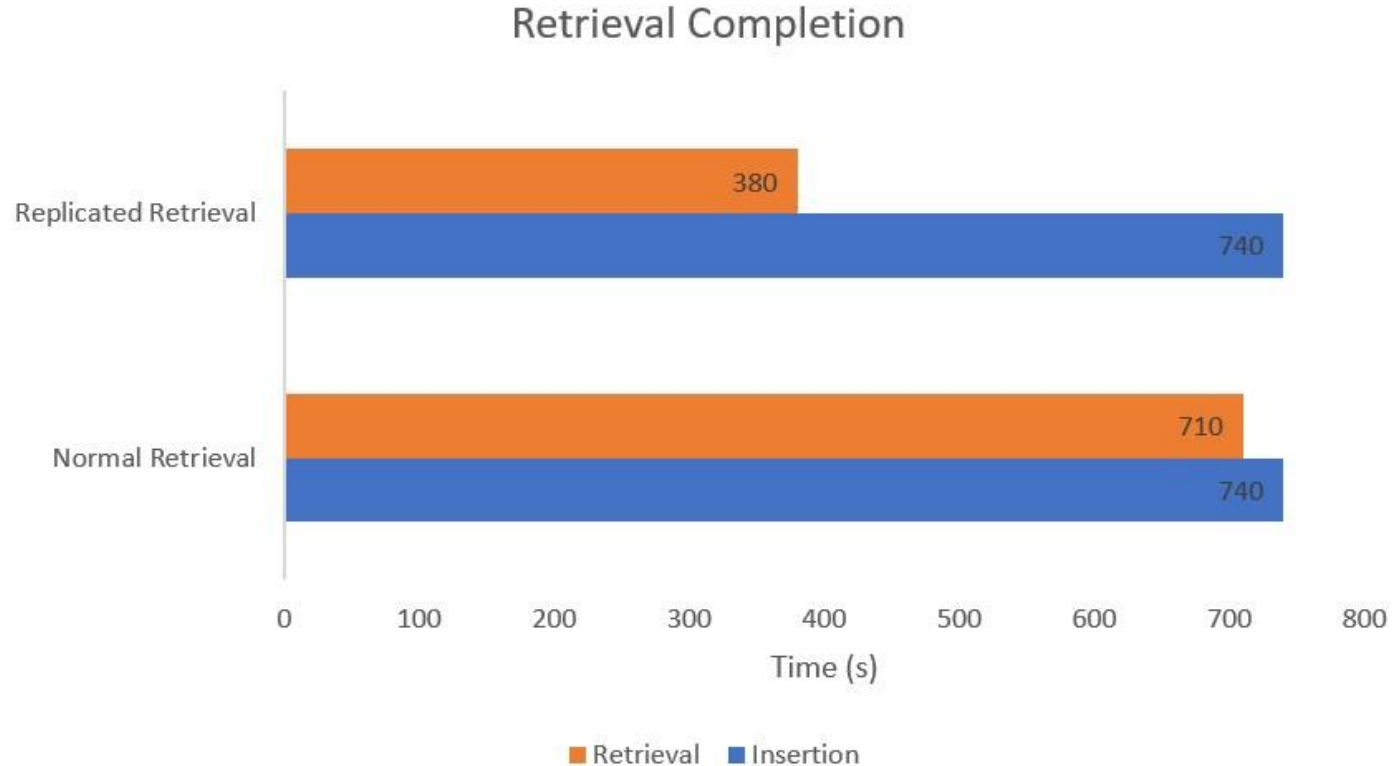
Performance Evaluation - Caching

Retrieval Completion & Cache size
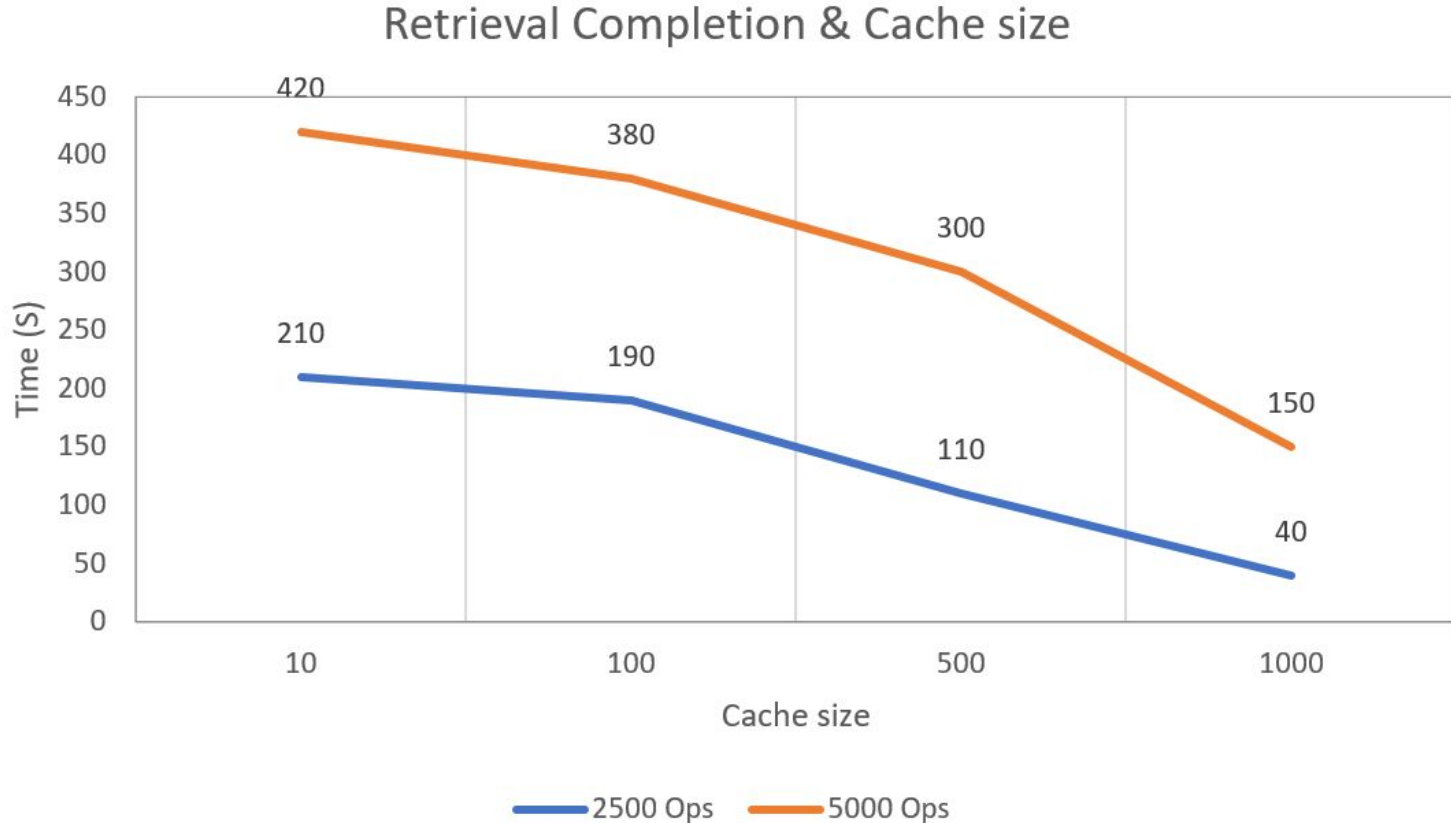
Conclusion

- Novel extensions with the goal to improve
  - stability
  - consistency
  - security

- Limitation & Future work
  - Hardware limitation - use cloud services to validate more tests
  - ECS Failure - ECS replication or peer to peer network of multiple ECS
  - Data replica to more than three servers

Project Presentation Group 7

Master Lab: Cloud Databases

Prof. Dr. Ruben Mayer | Chair for Application and Middleware Systems (I13) | Supervisor: Jawad Tahir | Students: Mohamed Elgamal, Nico Denner, Ahmed Mahmoud