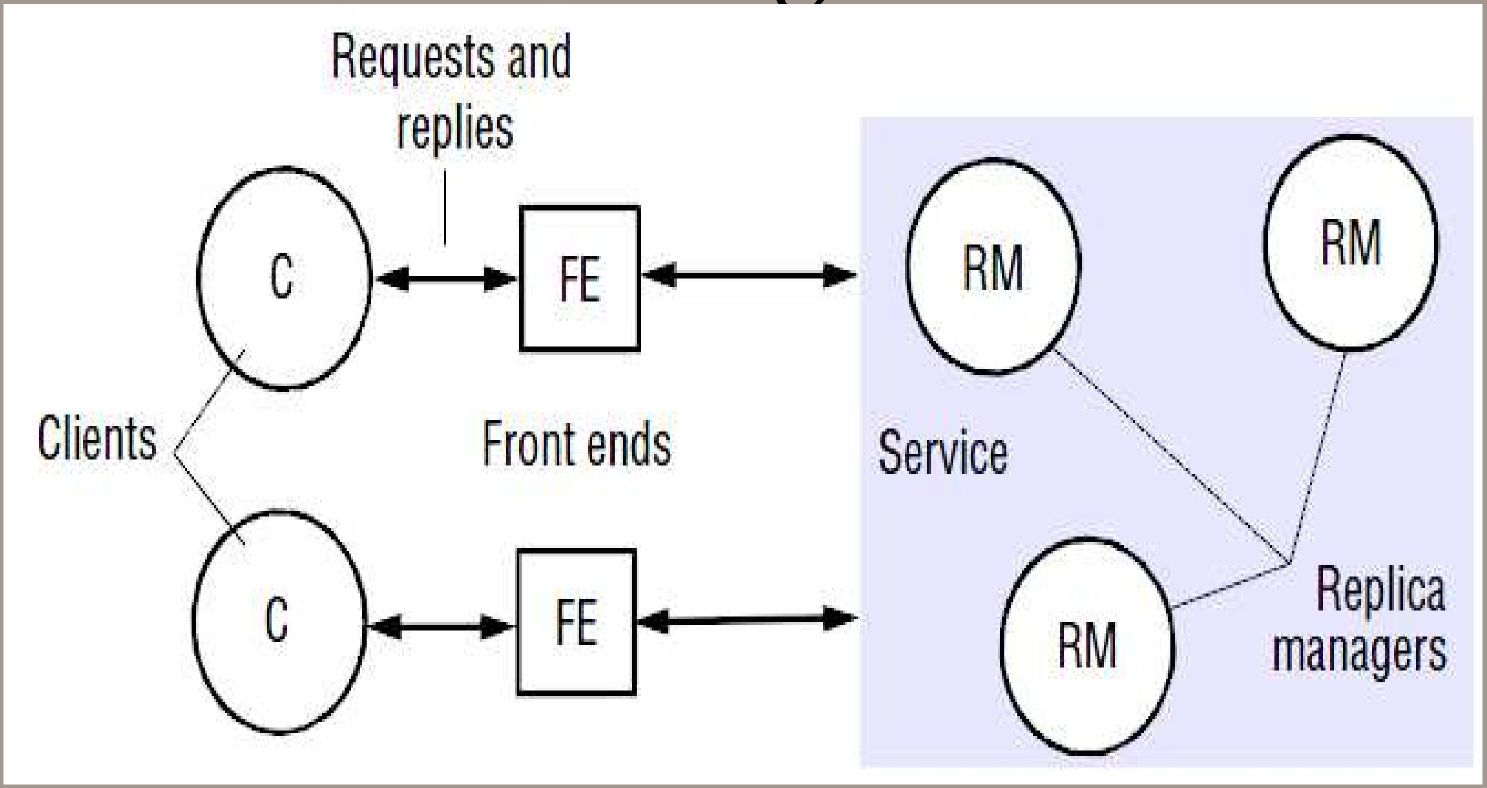
Replication in Distributed System

George Coulouris, Jean Dollimore and Tim Kindberg, "Distributed Systems Concepts and Design", Fifth Edition, Pearson Education, 2012

Replication

- Replication is the maintenance of copies of data at multiple computers.
- Replication is a key to the effectiveness of distributed systems.
- It can provide enhanced performance, high availability and fault tolerance.
- Fault Tolerance: The system should work correctly inspite of failures, system crash or network partition.

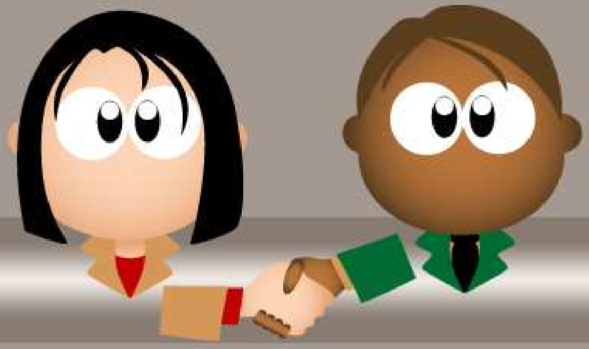
Architecture of Replication Management



Replication

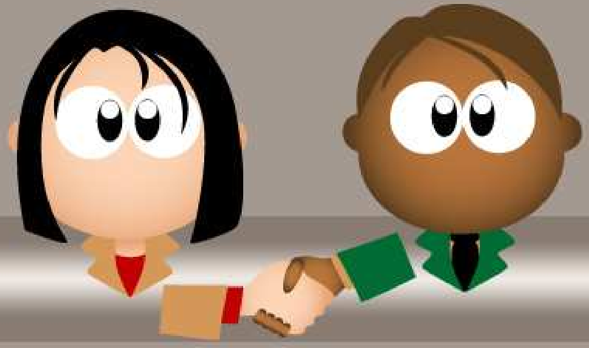
- 5 Phases in Replica management.
- **Request**: The front end issues the request to one or more replica managers
- **Coordination**: The replica managers coordinate for executing the request consistently. They also decide on the ordering of this request relative to others
- FIFO ordering, Causal ordering, Total ordering
- **Execution**: The replica managers execute the request
- **Agreement**: The replica managers reach consensus on the effect of the request
- **Response**: One or more replica managers responds to the front end

Types of Replication



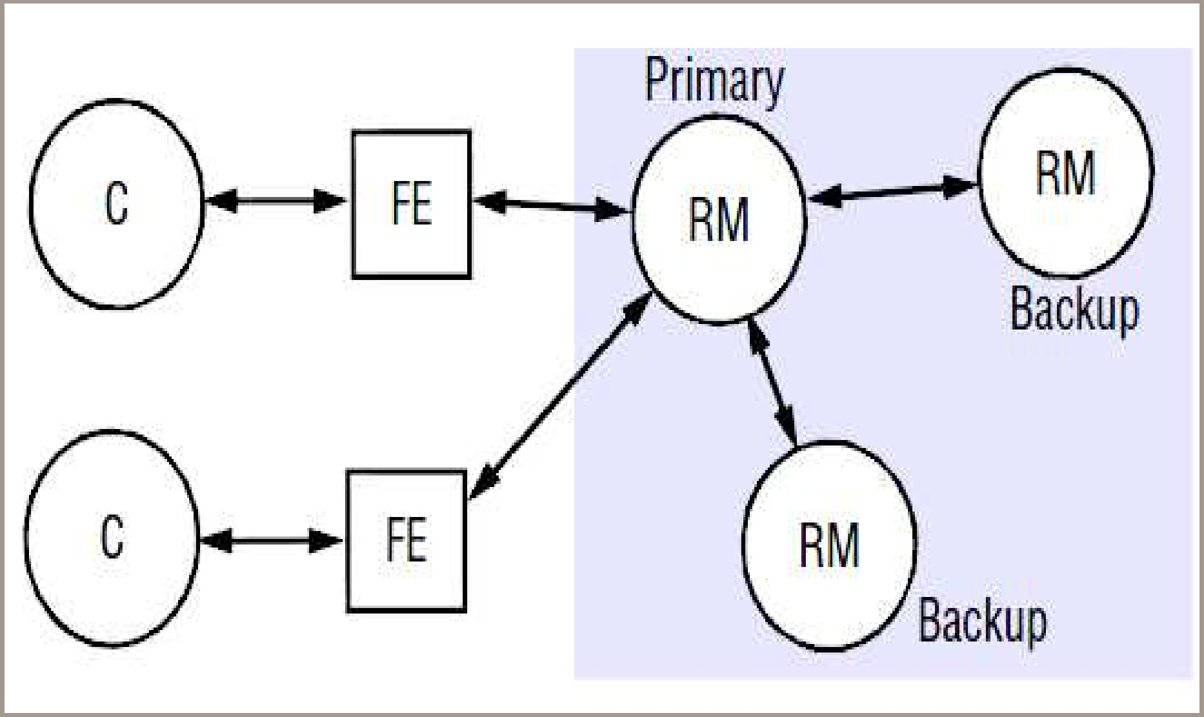
Types of Replication

- Passive (Primary-Backup) Replication
- Active Replication



Passive (Primary-Backup) Replication

- Passive or primary-backup Replication model
- A single primary replica manager and one or more secondary replica managers – 'backups' or 'slaves'.



Passive (Primary-Backup) Replication

- Request: The front end issues the request, containing a unique identifier, to the primary replica manager.
- Coordination: The primary takes each request atomically, in the order in which it receives it. It checks the unique identifier, in case it has already executed the request, and if so it simply resends the response.
- Execution: The primary executes the request and stores the response.
- Agreement: If the request is an update, then the primary sends the updated state, the response and the unique identifier to all the backups. The backups send an acknowledgement.
- Response: The primary responds to the front end, which hands the response back to the client.
- Front-End can send read only request to slaves and write request to Primary.

Passive (Primary-Backup) Replication

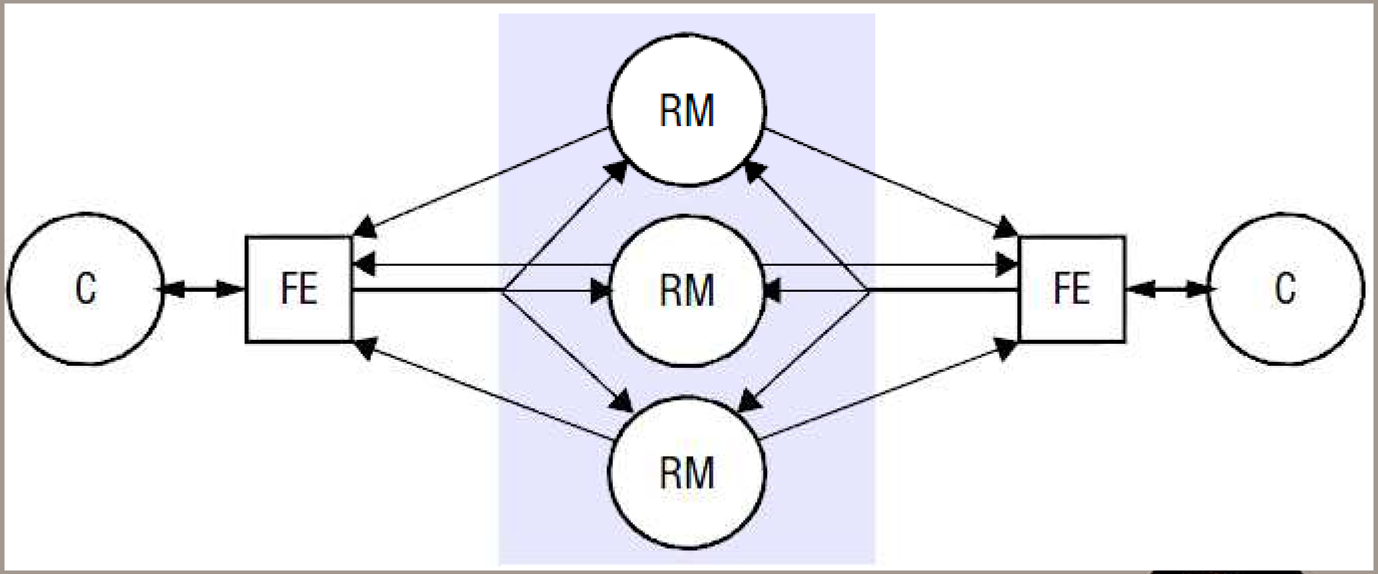
- Problems with Passive Replication
- The primary may have crashed at any point during the operation. If it gets crashed before the agreement stage (4), then the surviving replica managers have not processed the request.
- If it gets crashed during the agreement stage, then they may have processed the request.
- If it gets crashed after aggreement stage, then they have definitely processed it.
- But the new primary does not have to know what stage the old primary was in when it got crashed.
- When it receives a request, it proceeds from stage 2 and above.
- To survive up to f process crashes, a passive replication system requires f +1 replica managers.
- A system cannot tolerate Byzantine failures

Passive (Primary-Backup) Replication

- Problems with Passive Replication
- Large overheads
- If the primary fails then yet more latency is incurred
- Clients can submit read requests to the backups, thus offloading work from the primary.
- Passive Replication is used in
- Harp replicated file system
- The Sun Network Information Service (NIS, formerly Yellow Pages)

Active Replication

- Active Replication model
- Replica managers are state machines that play equivalent roles and are organized as a group.



fppt.com

Active Replication

- **Request**: The front end attaches a unique identifier to the request and multicasts it to the group of replica managers, using a totally ordered, reliable multicast primitive.
- **Coordination**: The group communication system delivers the request to every correct replica manager in the same (total) order.
- **Execution**: Every replica manager executes the request. Since they are state machines and requests are delivered in the same total order, correct replica managers all process the request identically.
- **Agreement**: **Agreement phase is not needed**, because of the multicast delivery semantics.
- **Response**: Each replica manager sends its response to the front end.
- Front end passes the first response to arrive back to the client and discards the rest
- Each front end's requests are served in FIFO order

Active Replication

- Active replication system can mask up to *f Byzantine failures, as long as the service incorporates at least 2f + 1 replica* managers.
- Each front end waits until it has collected *f + 1 identical responses and passes* that response back to the client
- Front ends may send read-only requests only to individual replica Managers
- Front end can easily mask the failure of a replica manager in this case, simply by submitting the read-only request to another replica manager.

Thank You