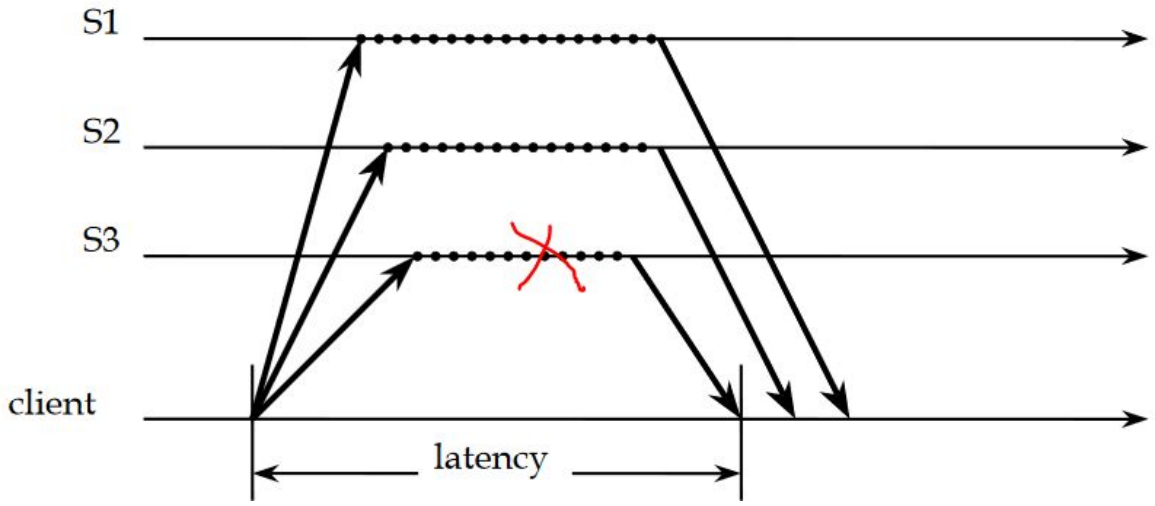
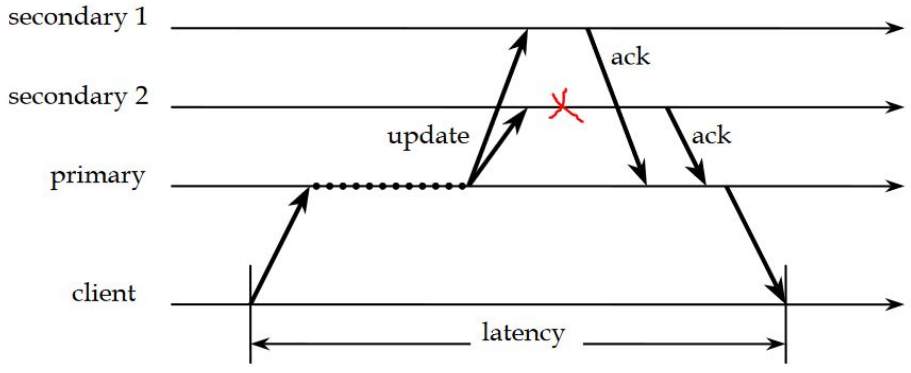
1.- Discuss in the group what are the consequences of a failure in one of the replicas in each of the techniques. Consider first the active replication technique and then the passive replication technique. For the moment, for the latter consider only the possibility of failure on a secondary server. For each of the techniques, write down the answers to the following questions: (A) How do they behave according to the properties of order and atomicity? (B) What specific actions must be taken to guarantee the properties in case of failure?

Failure in active replication: Active replication is performed by processing the same request at every replica. Client requests are made through total order broadcast. The client waits, then, for the first reply. The failure of a replica is completely transparent to the client. It has a lower latency than passive replication, because of the simplicity. Anyway, the total order broadcast is more expensive. A failure in a secondary server, as seen on the image in red, is not that bad, because the other servers can end the process.



As seen, if there is an error on S3, the other servers can end the process. The latency depends on the first response of a server.

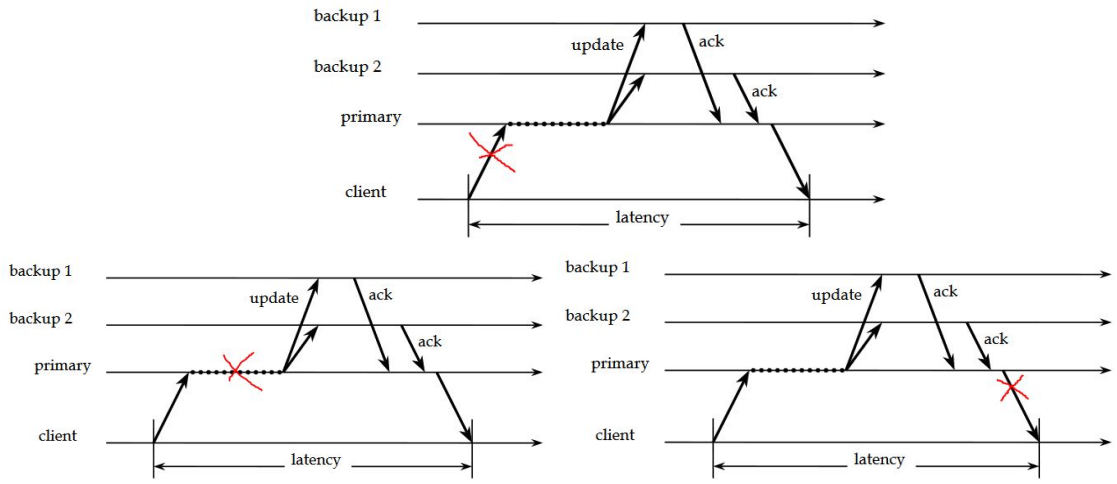
Failure in passive replication: One server is designated as the primary, while all the others are backups. The clients send their requests to the primary only. The primary executes the request, atomically updates the other copies, and sends the response to the client . The order is determined or defined by the primary. A failure in the active replication can lead to reprocessing the client's request, so we have to make sure that the client's request is only processed once.



If a problem occurs in backup 2, it can be solved with the help of backup 1. This also would reduce the latency, taking the latency it would get only with backup 1. This latency reduction is achieved by a constant communication between servers. These communications are not represented in that scheme.

2.- Now, consider the failure of the primary server in your passive replication technique. What consequences does it have? Consider the different situations in which the failure may occur. What measures must be taken to ensure transparency of failures?

There are several points in which the error may occur: during the client's request to the primary server, processing the client's request on the primary server and sending the request processed to the client, as seen on the images below.



Usually the most common failure is the server failure, not a communication failure.
If there is a failure on the primary server in the passive replication, the worst thing that could happen is that the client's request is not processed only once.
In order to ensure complete transparency of errors we can't rely on just passive replication, since a failure of the primary server will result in a timeout and having to restart the request which would not be transparent. If we implement active replication on the other hand a failure of any of the servers will not matter to the user as any can respond.

3.- Prepare a report that summarizes, for each of both techniques: (A) what is it based on to guarantee the property of atomicity; (B) on what is it based to guarantee the order property (this example can be useful to show the conditions imposed by the order property on the communication); (C) how it manages the transparency of failures, particularly in the case of passive replication.

1. Passive replication

a) Atomicity can be ensured because the primary waits for the “ack” messages of the replicas and if one fails it will be disconnected and not considered valid until it is updated and brought back online.
b) Order is defined by the primary, it will send the order in which the replicas will process the requests.
c) It ensures transparency by hiding the failure of a replica to process and send the ack message(timeout), the primary will remove the backup from the list of available backups but it will not inform the client of that, it will simply give the reply of the request and the client won’t be the wiser. The only time it won’t be transparent will be if the primary fails in which case it will simply timeout with the client having to re request breaking the transparency.

2. Active replication
   a) Total order broadcast guarantees the properties of order and atomicity required by linearizability. Total order broadcast is the base to guarantee consistency: All the correct processes are sent in the same message sequence. The reliable broadcast combines with different orders, thus leading different semantics, as seen in the table.

	FIFO Order	Causal Order	Total Order
Reliable broadcast			
FIFO broadcast	X		
Causal broadcast	X	X	
Total Order broadcast			X
FIFO Total Order broadcast	X		X
Causal & Total Order broadcast	X	X	X

   Unlike in passive replication, where the client must detect and manage the failure of the primary node, in active replication, thanks to the total order broadcast semantics, the failures of the nodes are transparent to the client. (provided that all the replicas do not fail)

   b) Total order broadcast guarantees the property of order.
   c) It manages transparency because if correctly implemented all the replies are the same and since the client discards all the answers but the first one, barring the scenario of total failure of all servers it will always be transparent to errors.