HBASECON ASIA2019

THE COMMUNITY EVENT FOR APACHE HBASE™

The Procedure v2 Implementation of WAL Splitting and ACL

meiyi@xiaomi.com
HBase Committer

Abstract

❑ Introduction of Procedure v2

Overview
Execution and Rollback
Models

❑ ACL

ACL based on ZK Notification
ACL based on Procedure v2

❑ WAL Splitting

WAL Splitting based on ZK Coordination
WAL Splitting based on Procedure v2

Abstract

❑ Introduction of Procedure v2

Overview
Execution and Rollback
Models

❑ ACL

ACL based on ZK Notification
ACL based on Procedure v2

❑ WAL Splitting

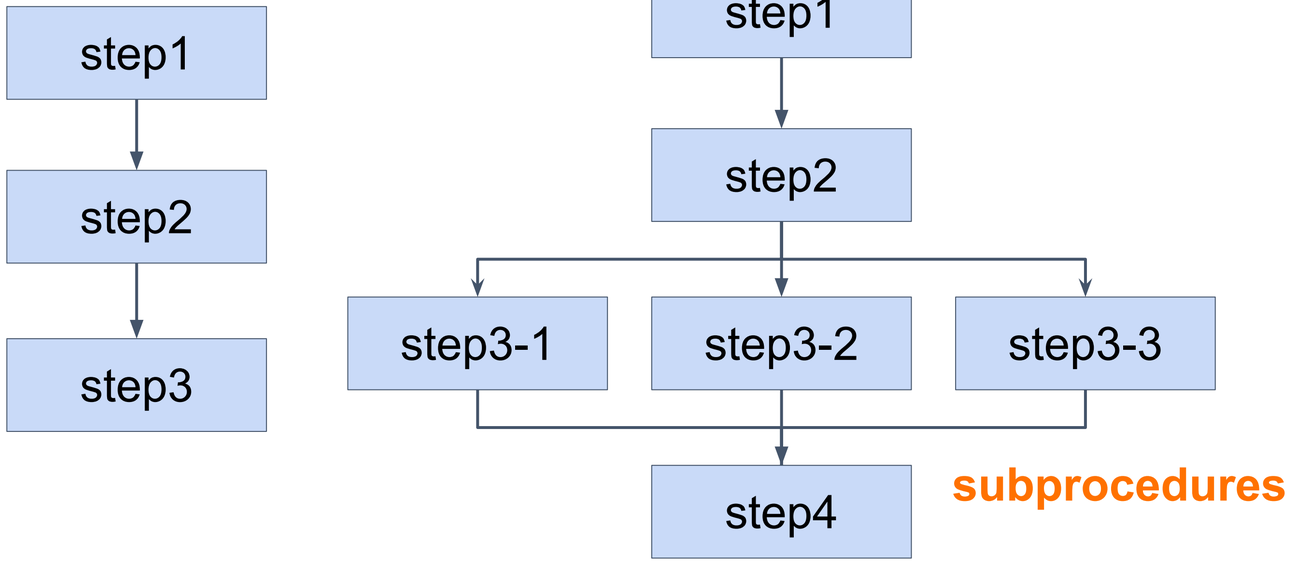
WAL Splitting based on ZK Coordination
WAL Splitting based on Procedure v2

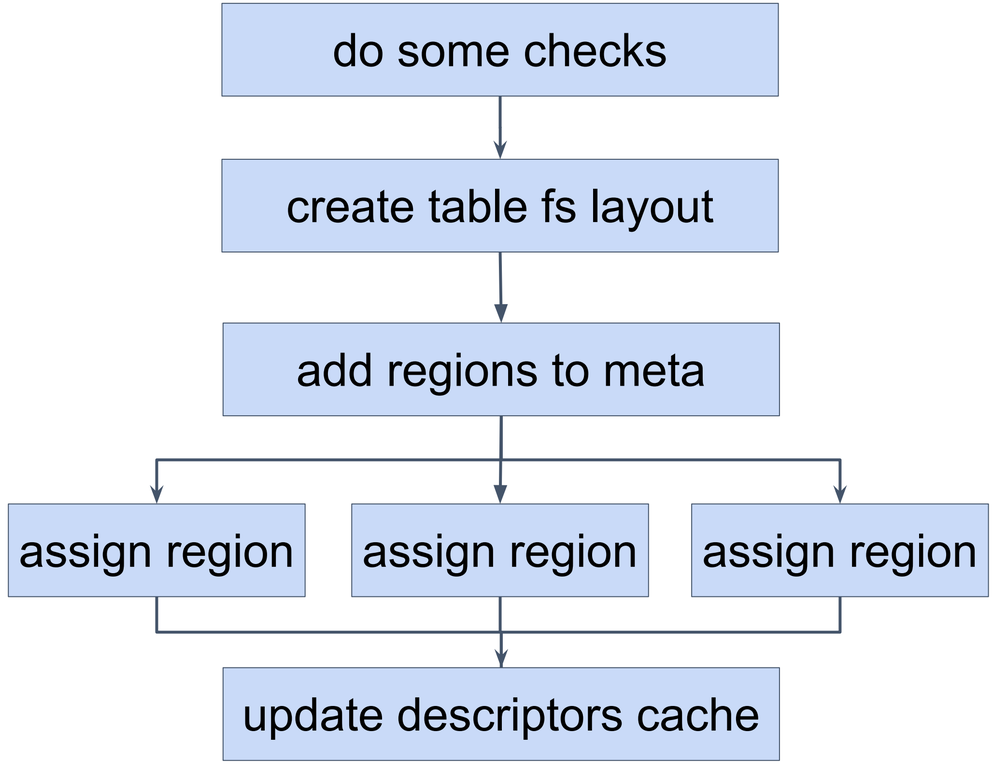
Goal of Procedure v2

Aims to provide a unified way to build:

- **multi-steps** procedure in case of failure (e.g. **Create table**)
- **notifications** across multiple machines (e.g. **ACLs/Quota cache updates**)
- **coordination** of long-running/heavy procedures (e.g. splits)
- procedures across **multiple machines** (e.g. **Assignment**)

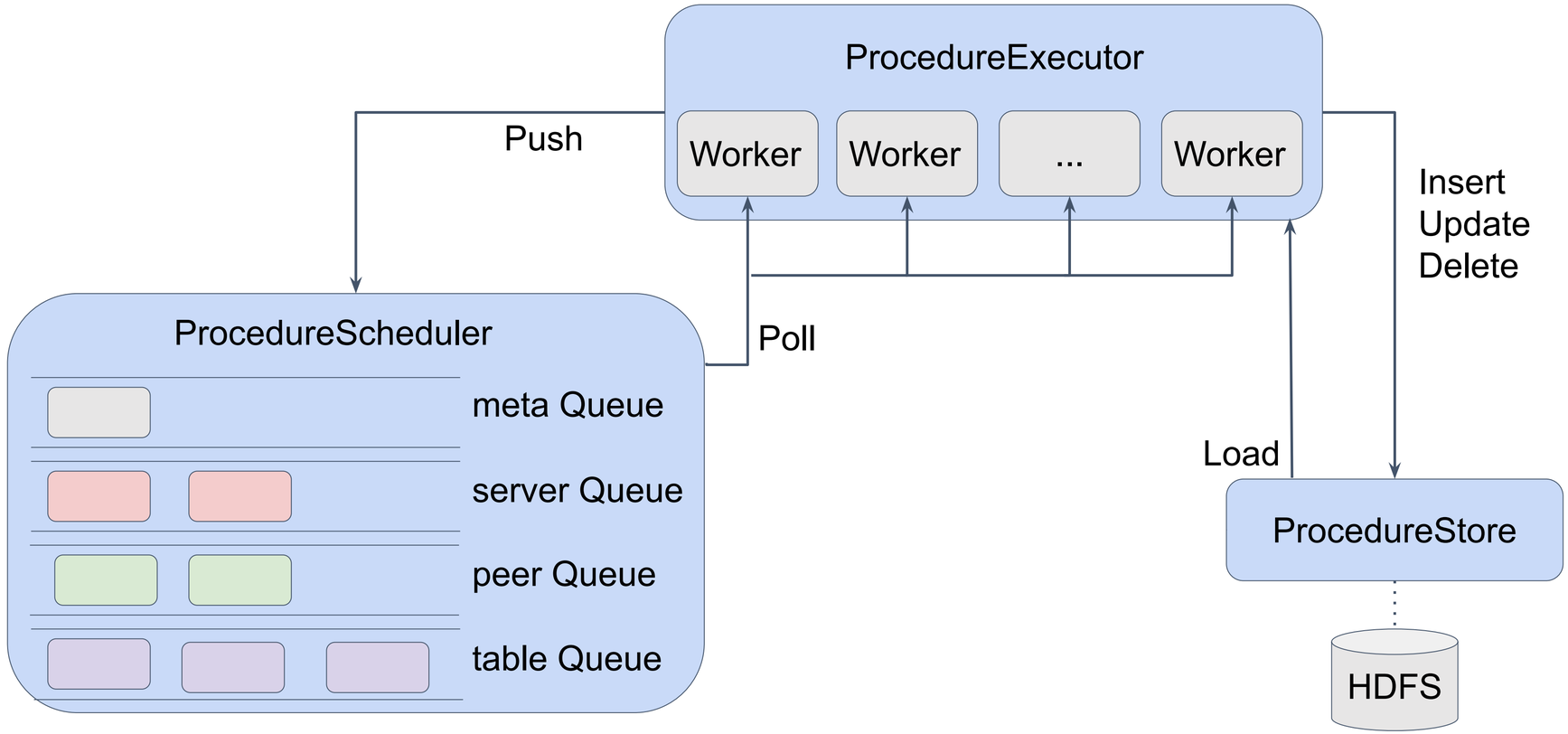
Build and run state machines



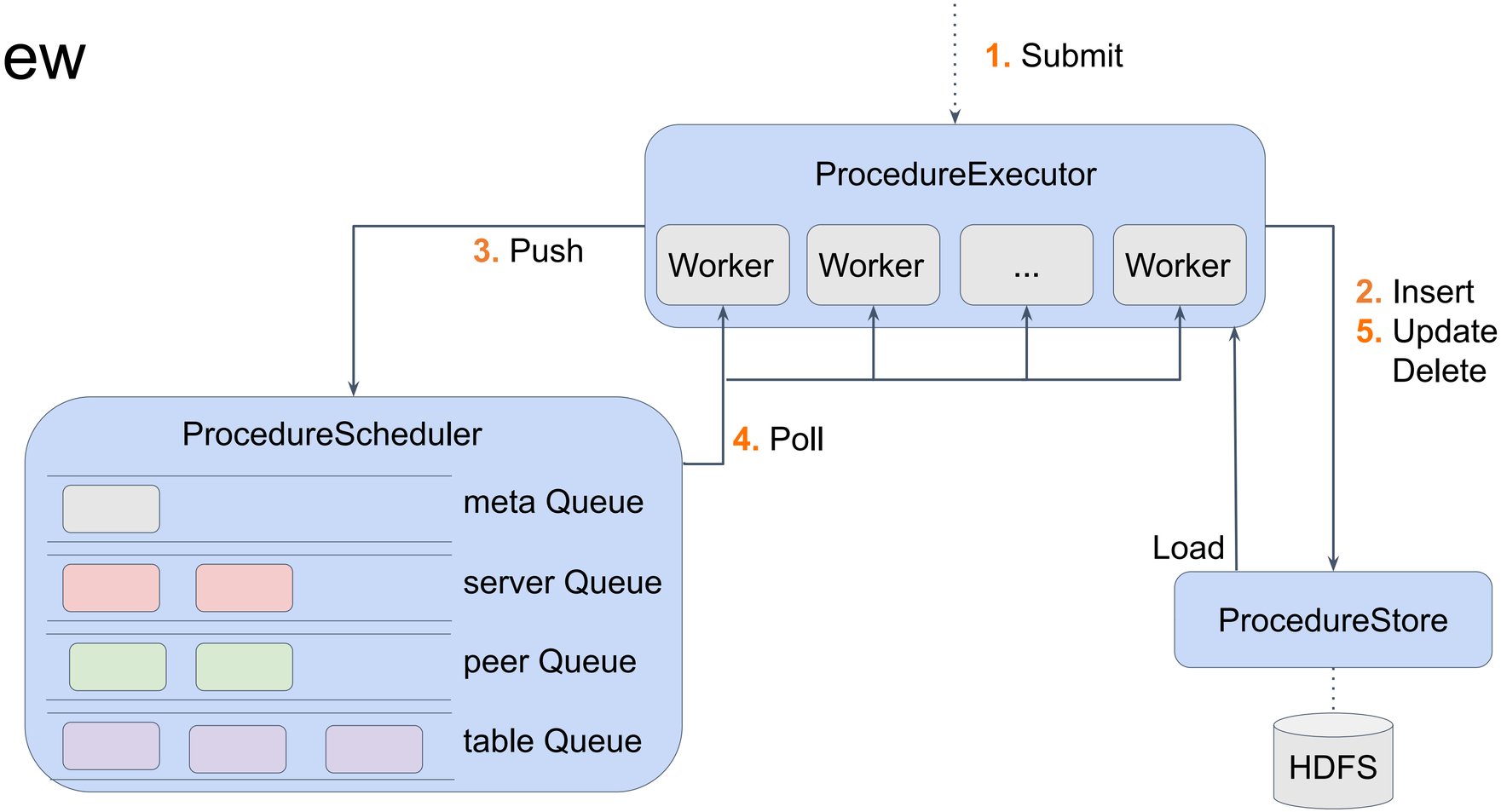


(steps of create table)

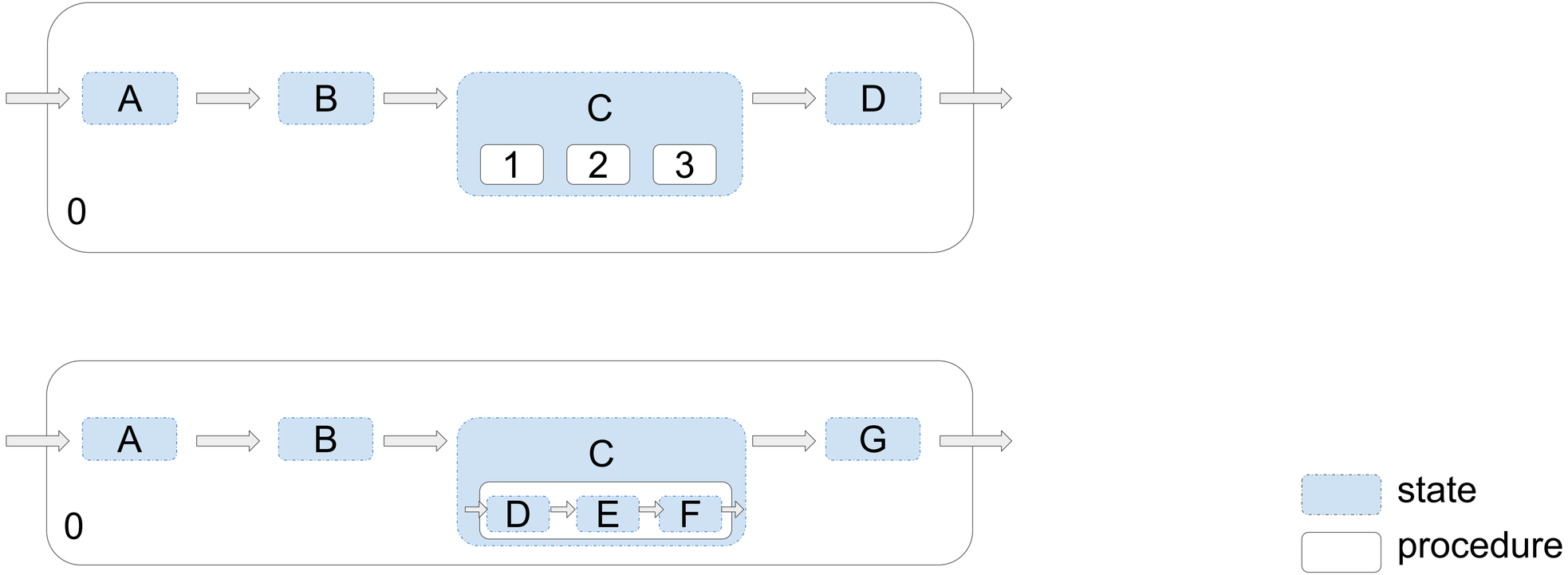
Overview


ProcedureExecutor
Worker
Worker
...
Worker
Push
Poll
Insert
Update
Delete
Load
ProcedureScheduler
meta Queue
server Queue
peer Queue
table Queue
ProcedureStore
HDFS

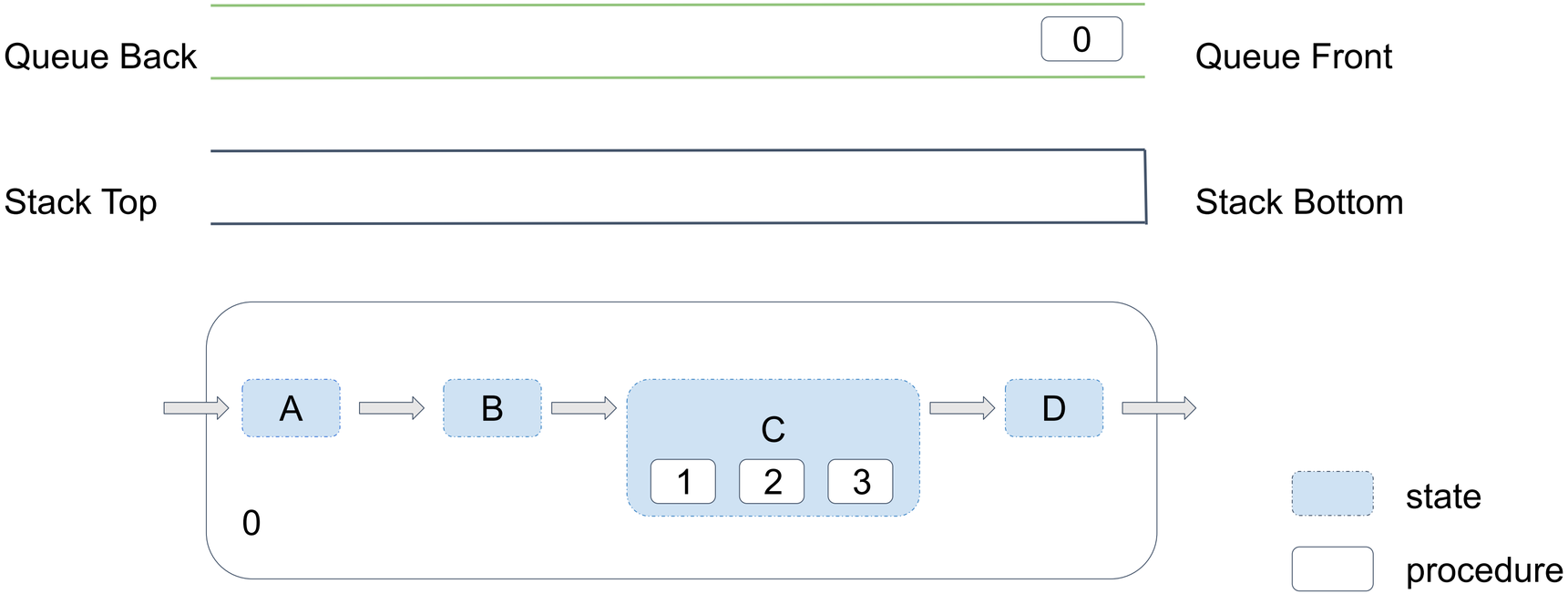
Overview


1. Submit
ProcedureExecutor
Worker
Worker
...
Worker
3. Push
2. Insert
5. Update
Delete
4. Poll
ProcedureScheduler
meta Queue
server Queue
peer Queue
table Queue
Load
ProcedureStore
HDFS

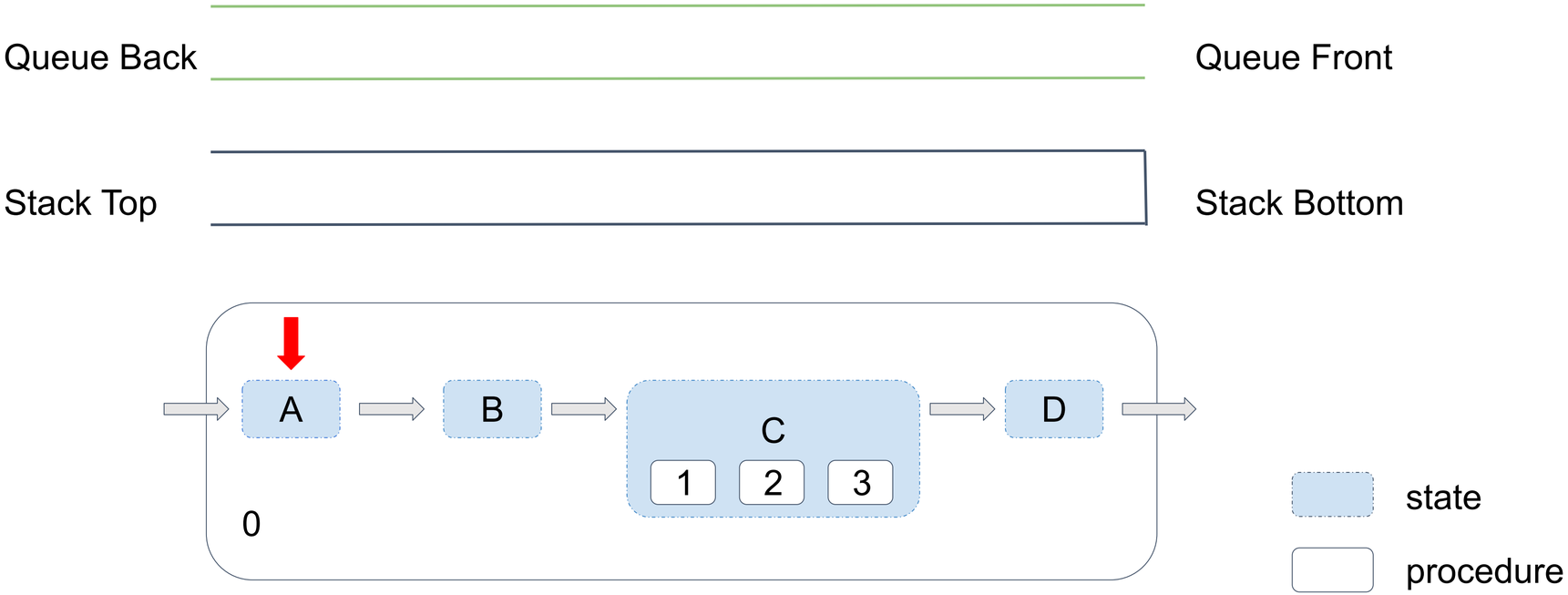
Procedure execution



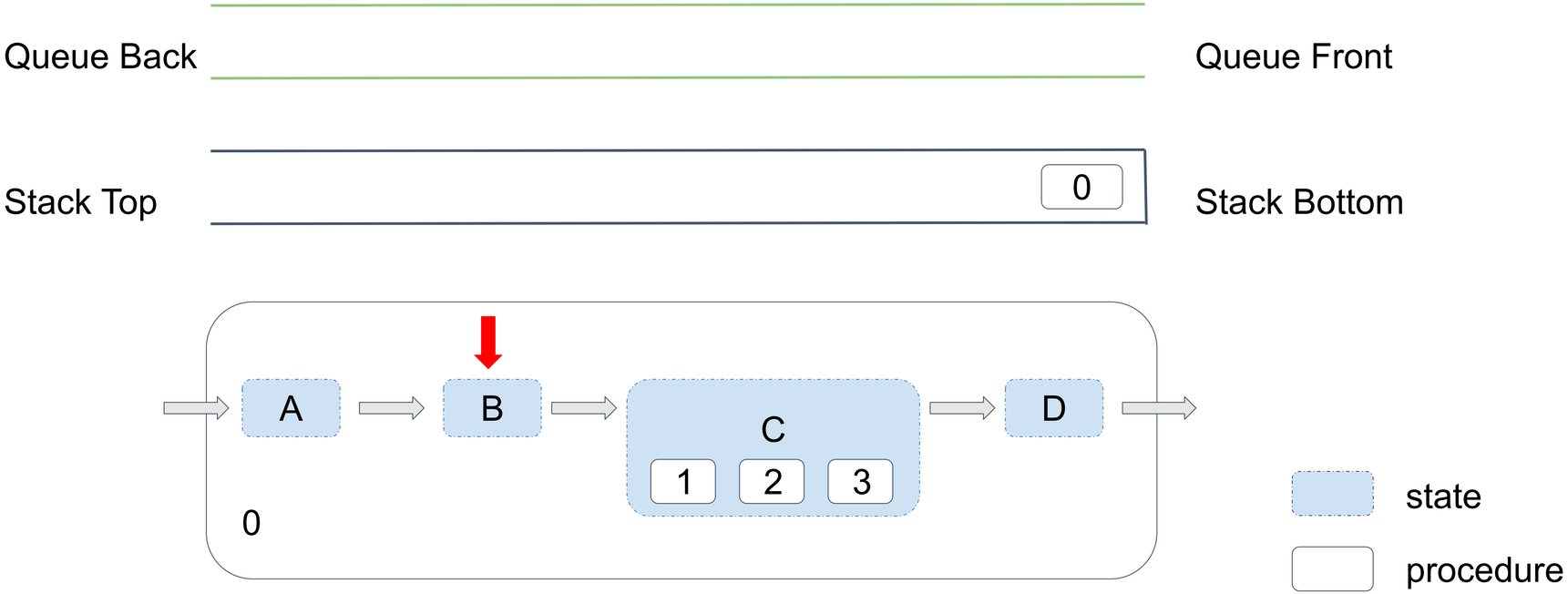
Procedure execution



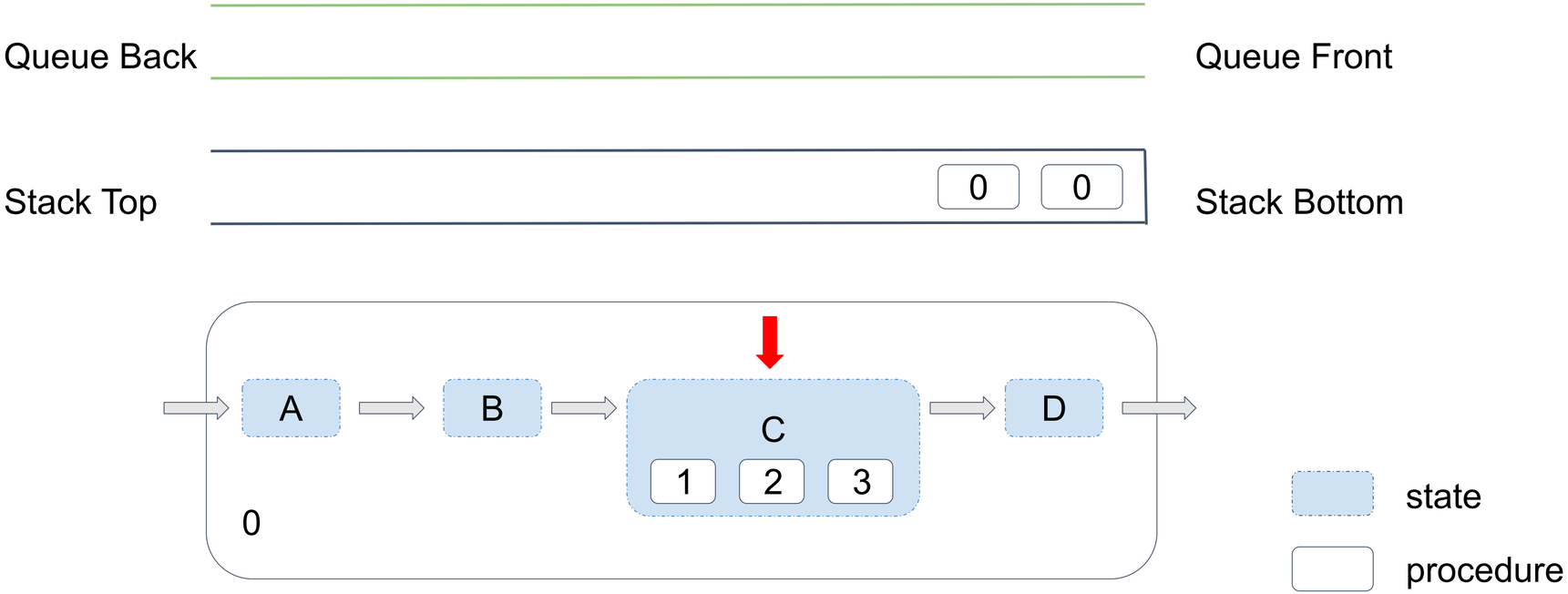
Procedure execution



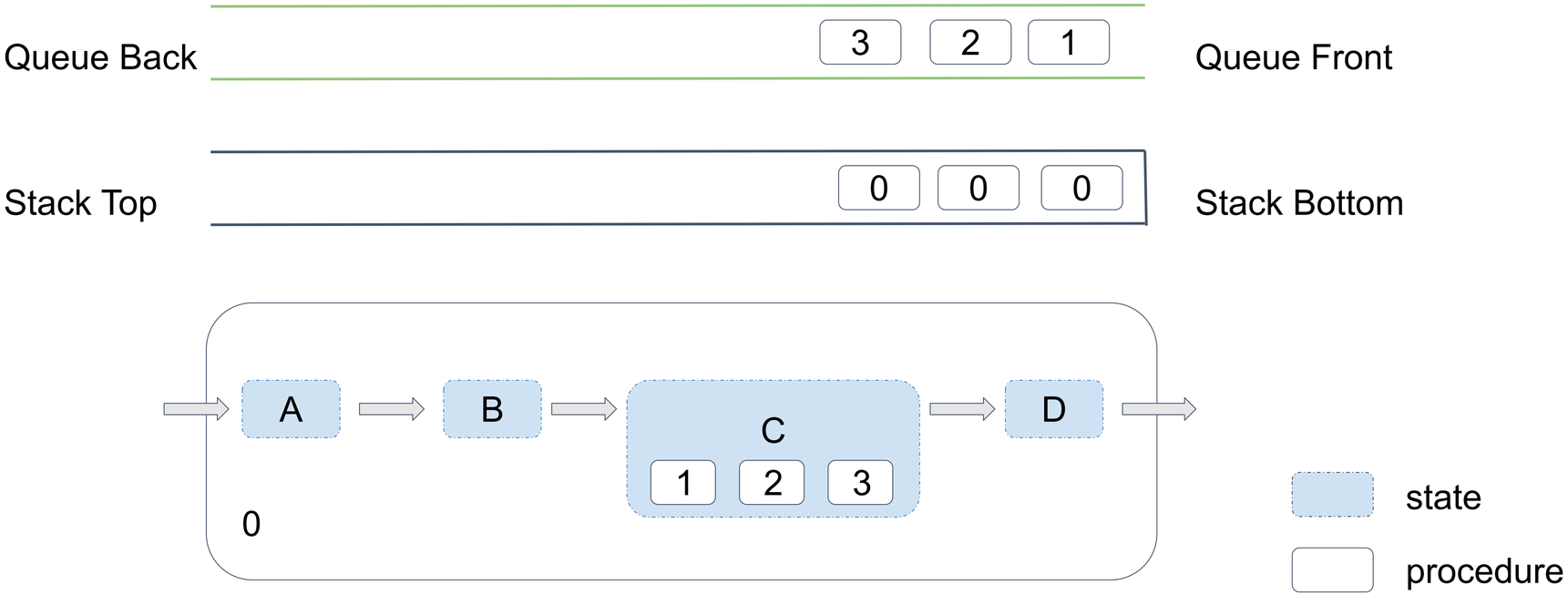
Procedure execution



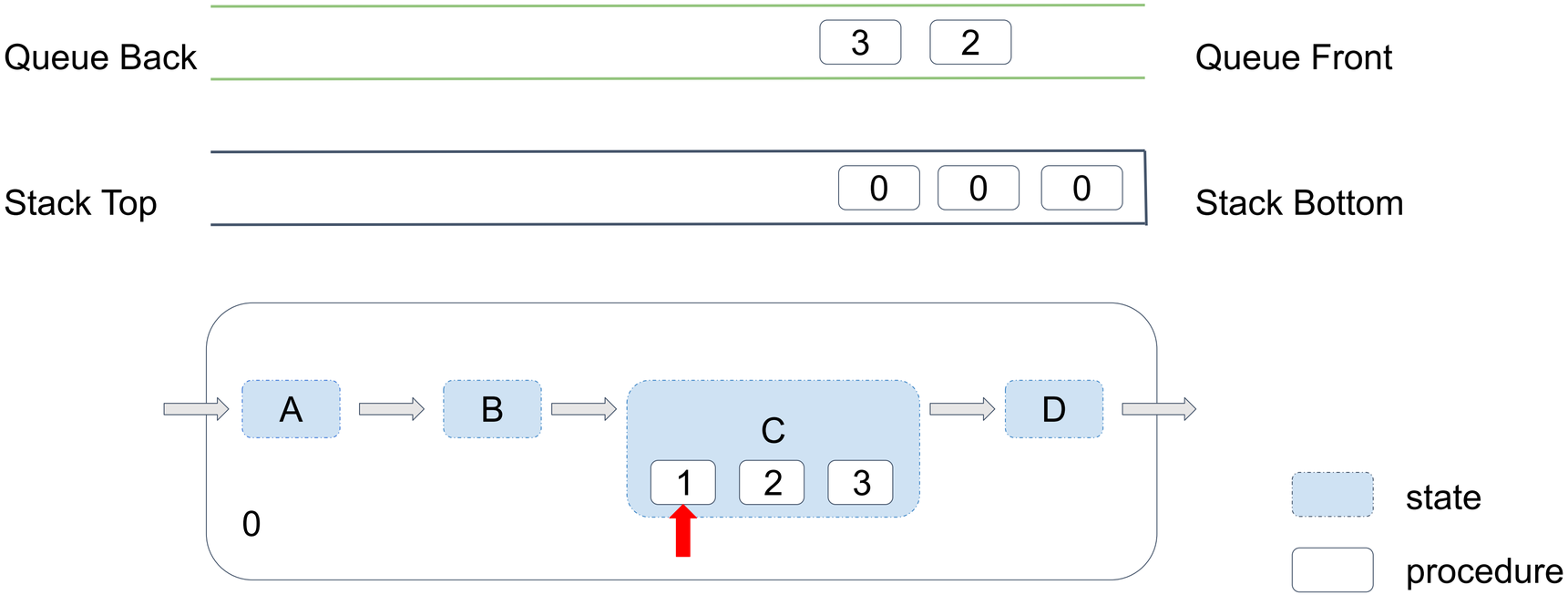
Procedure execution



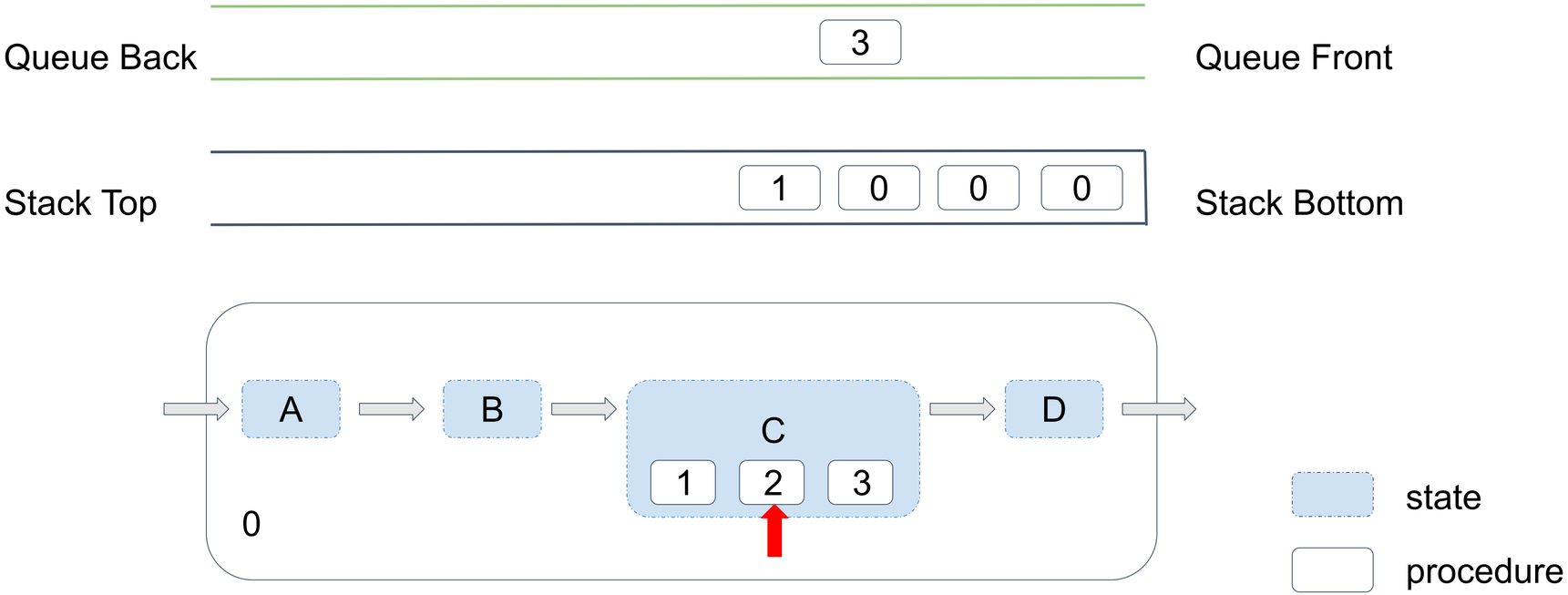
Procedure execution



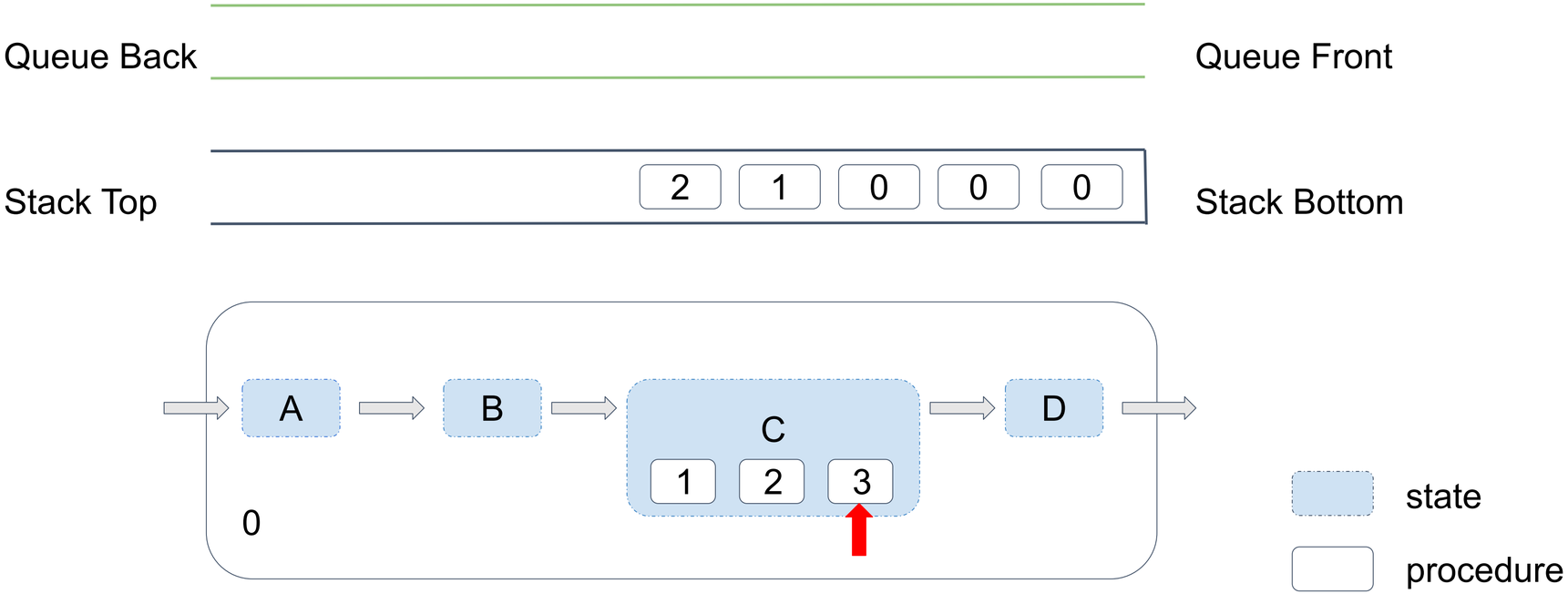
Procedure execution



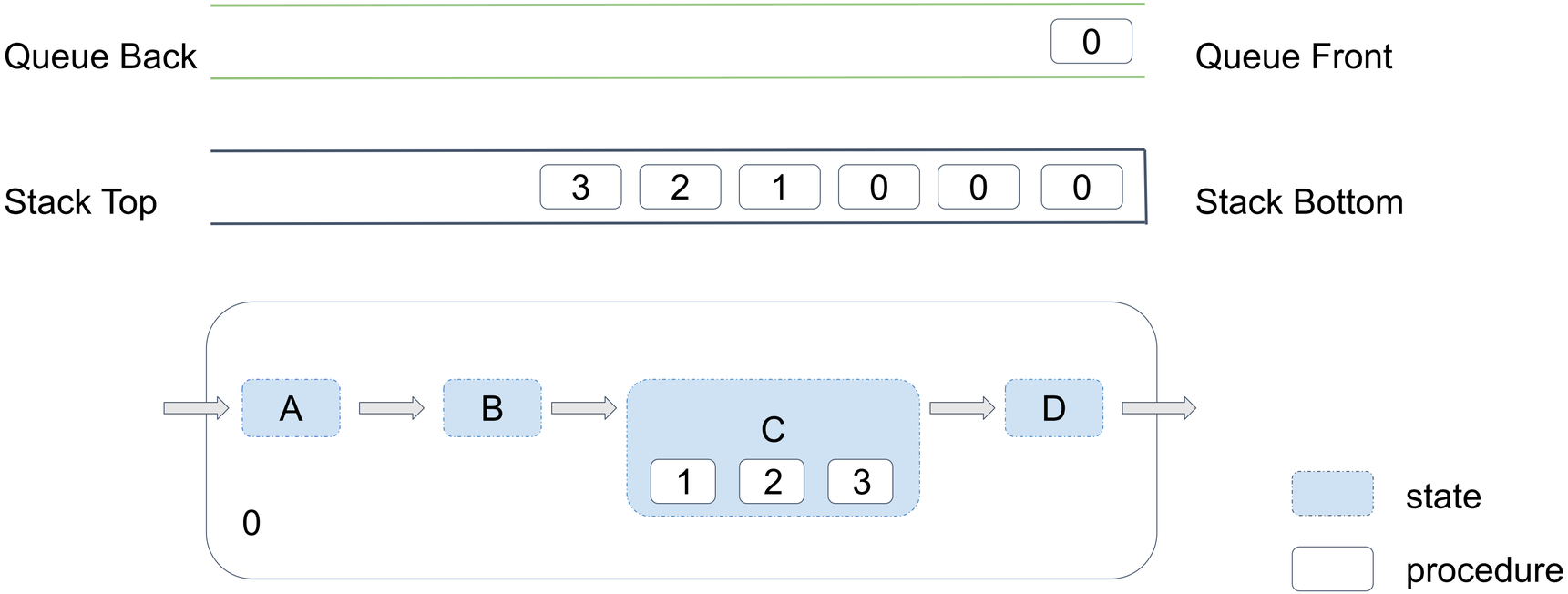
Procedure execution



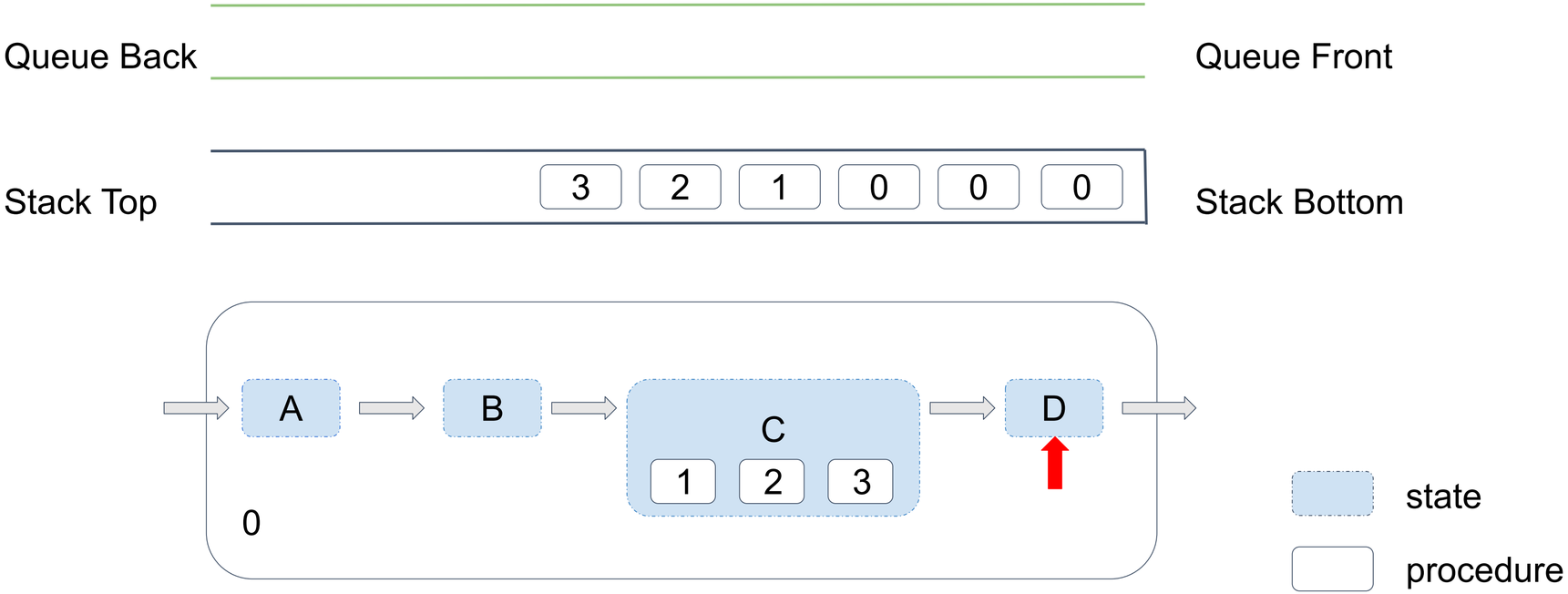
Procedure execution



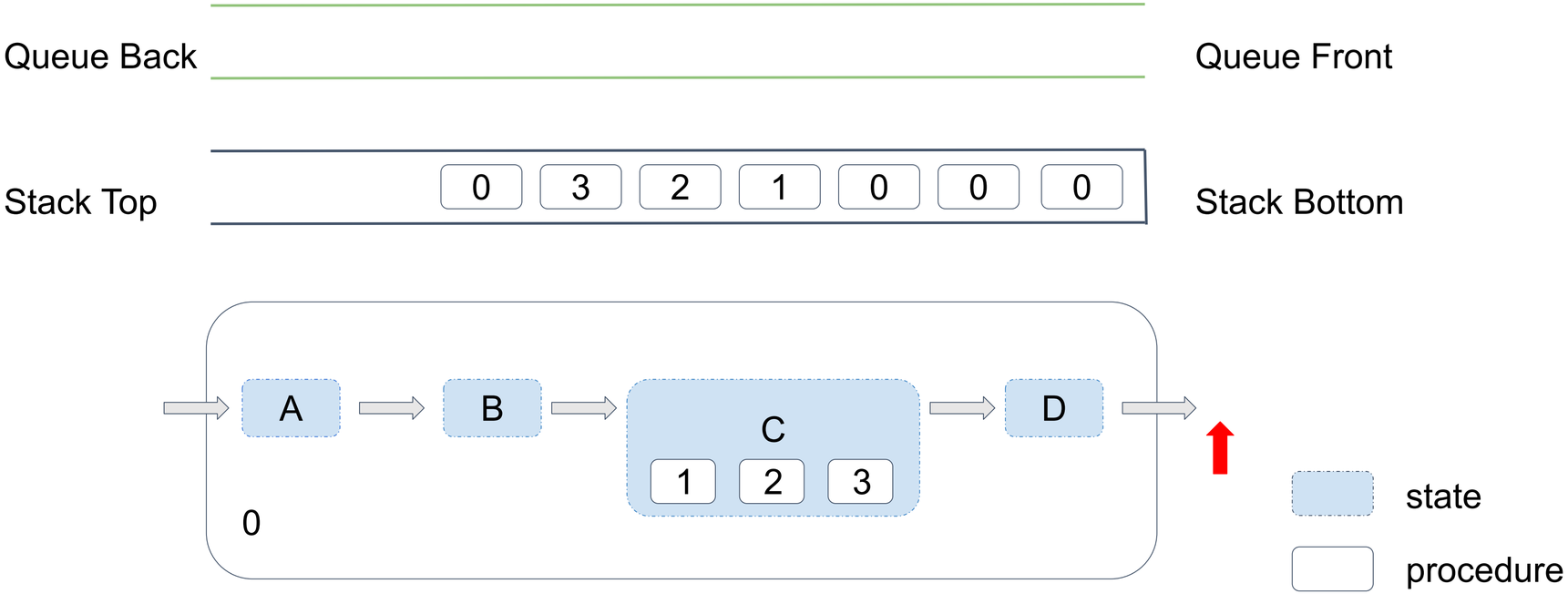
Procedure execution



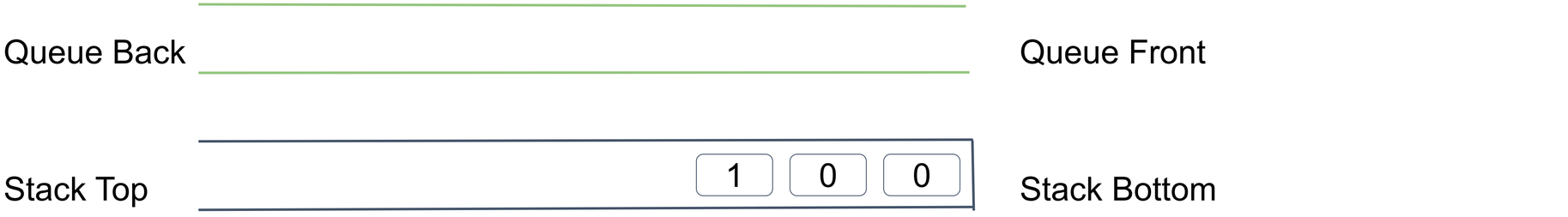
Procedure execution

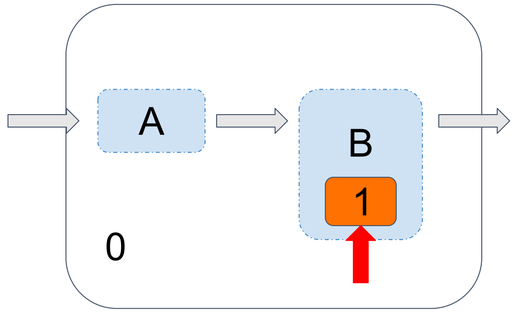


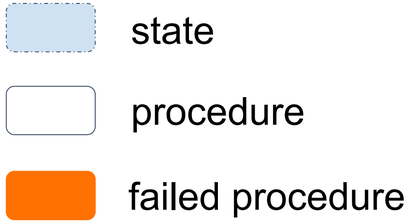
Procedure execution



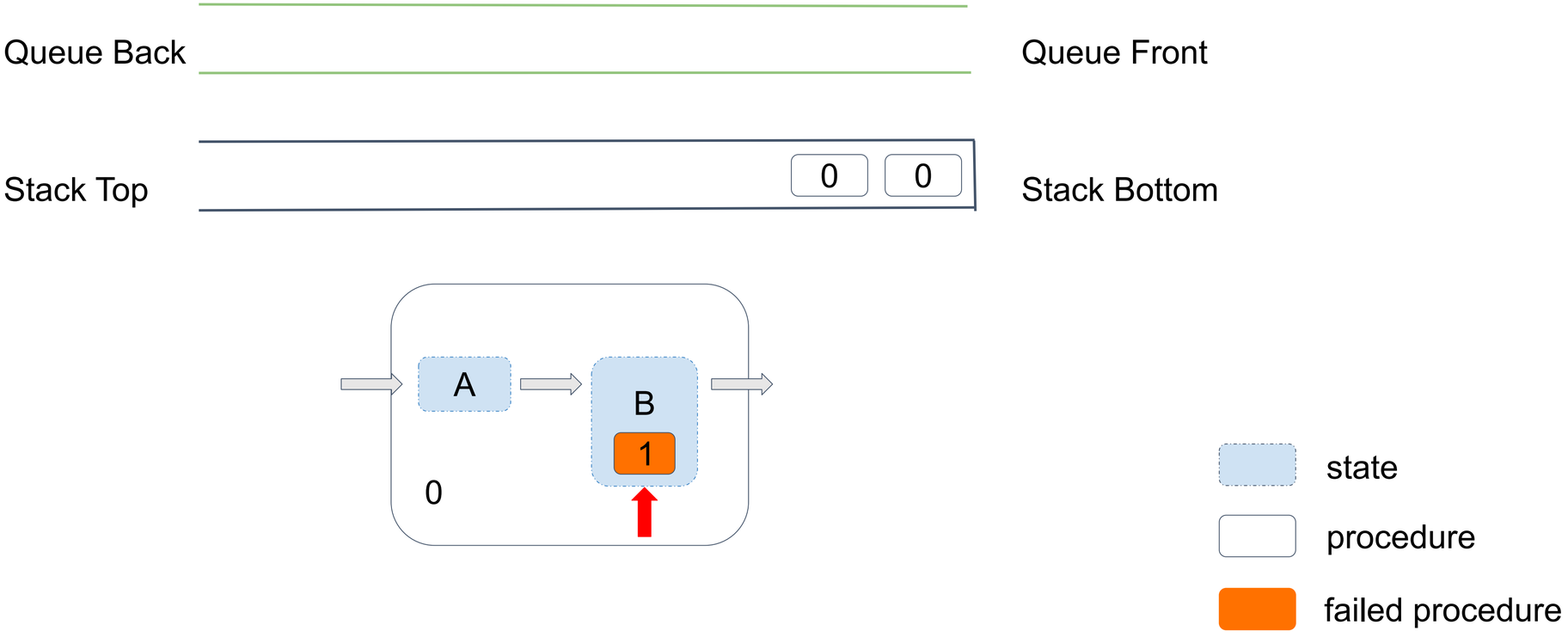
Procedure Rollback



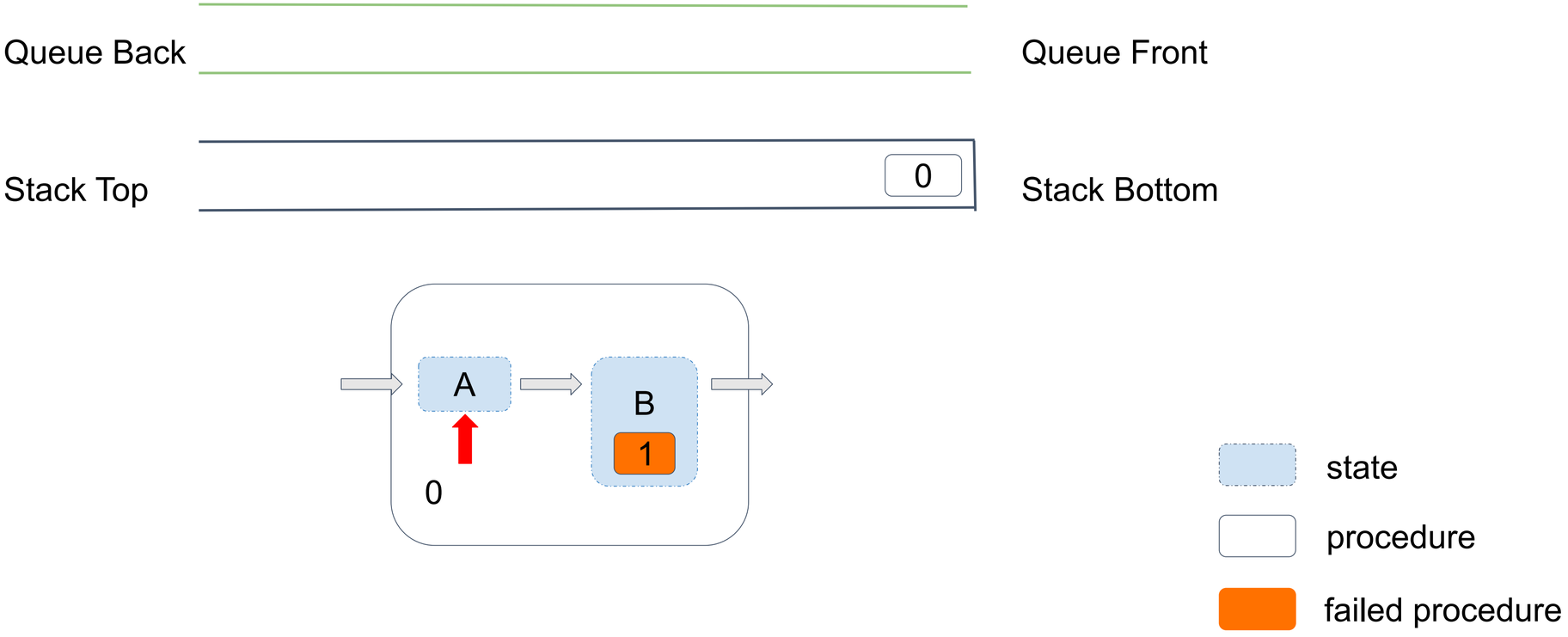




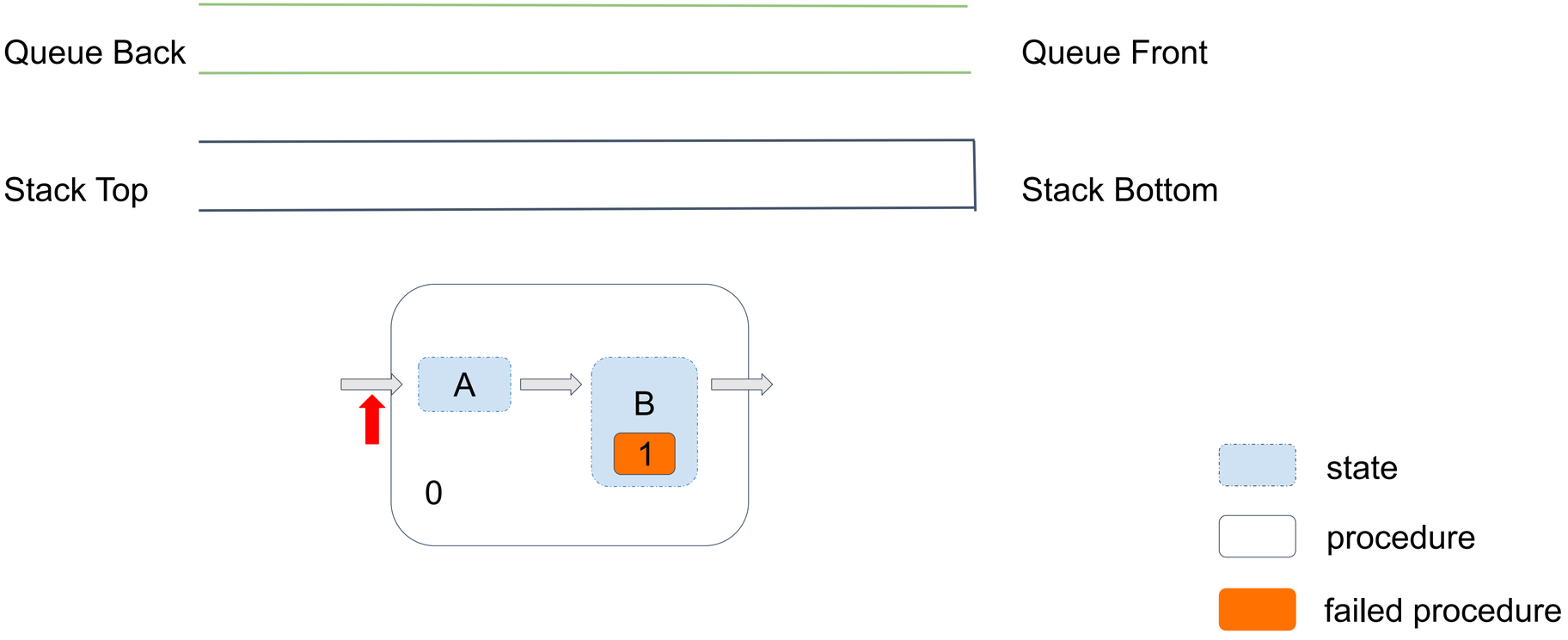
Procedure Rollback



Procedure Rollback

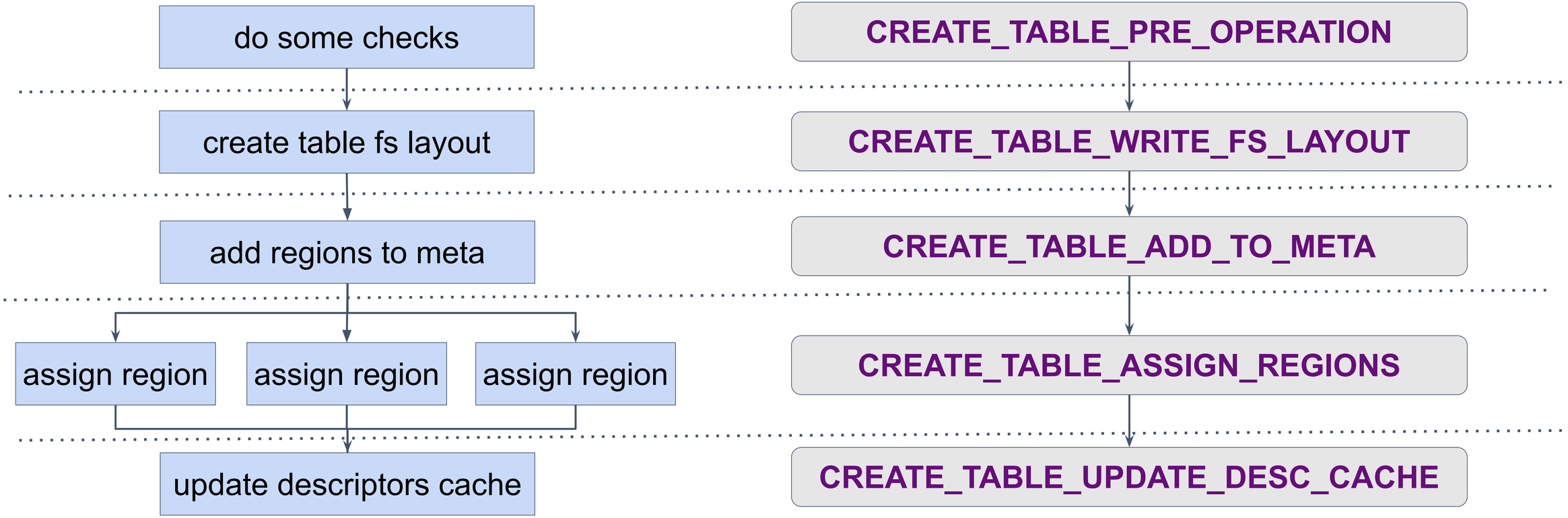


Procedure Rollback



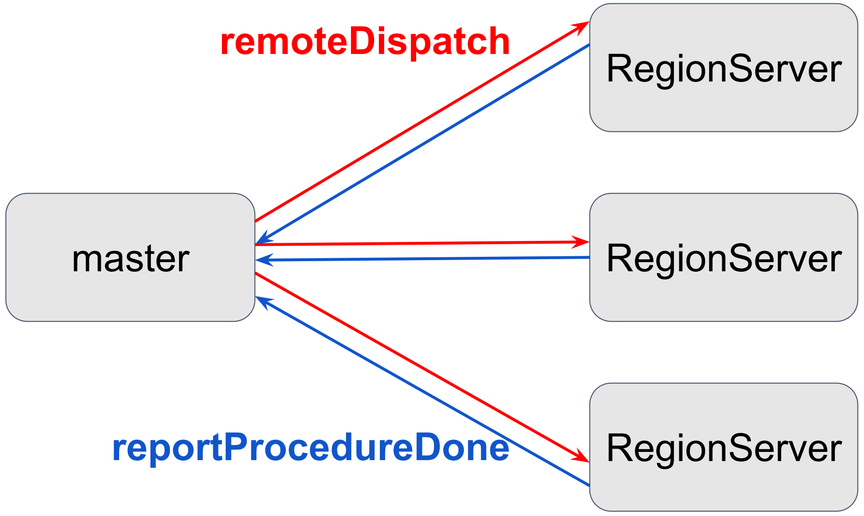
StateMachineProcedure

- enum of **states**, describing the various steps of the procedure
- transition from one state to another after calling **executeFromState** method



RemoteProcedureDispatcher

- Dispatch aggregated RPCs to remote server
- Remote server report procedures execute states in a heartbeat



Abstract

❑ Introduction of Procedure v2

Overview
Execution and Rollback
Models

❑ ACL

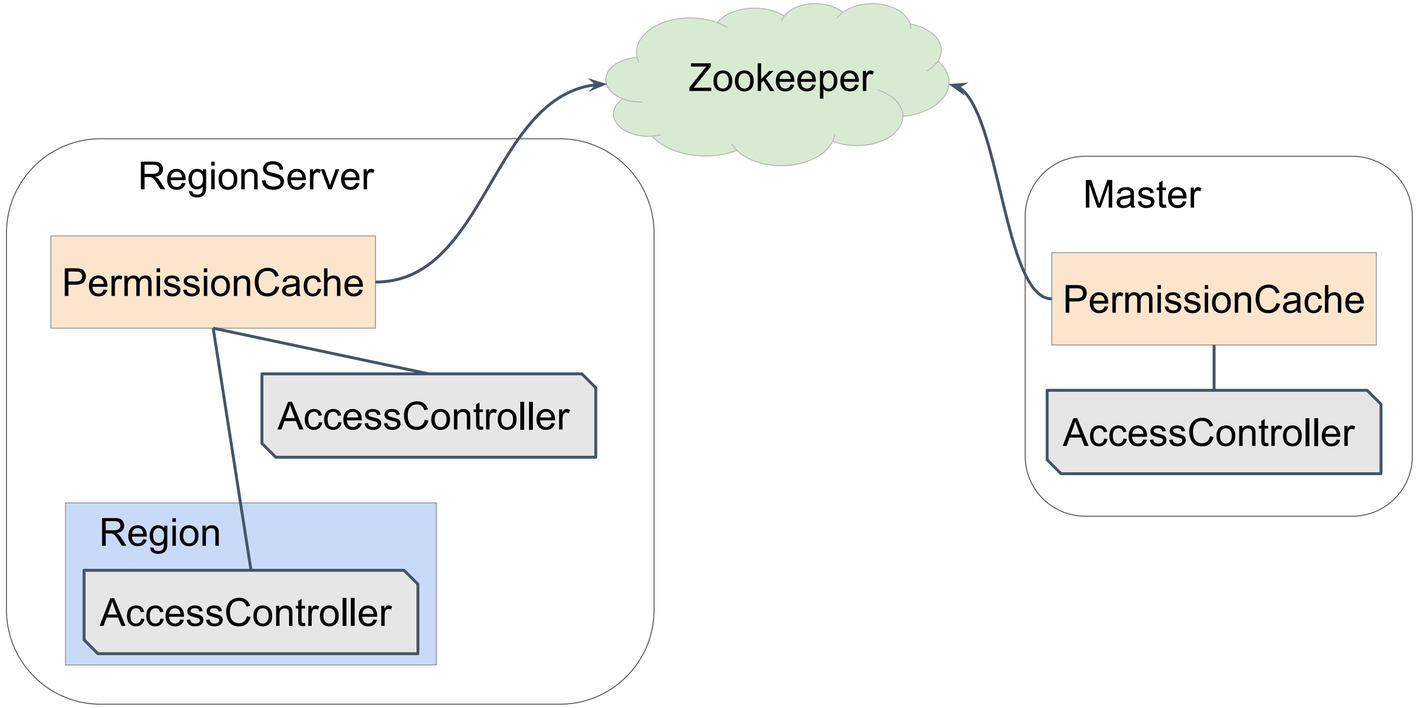
ACL based on ZK Notification
ACL based on Procedure v2

❑ WAL Splitting

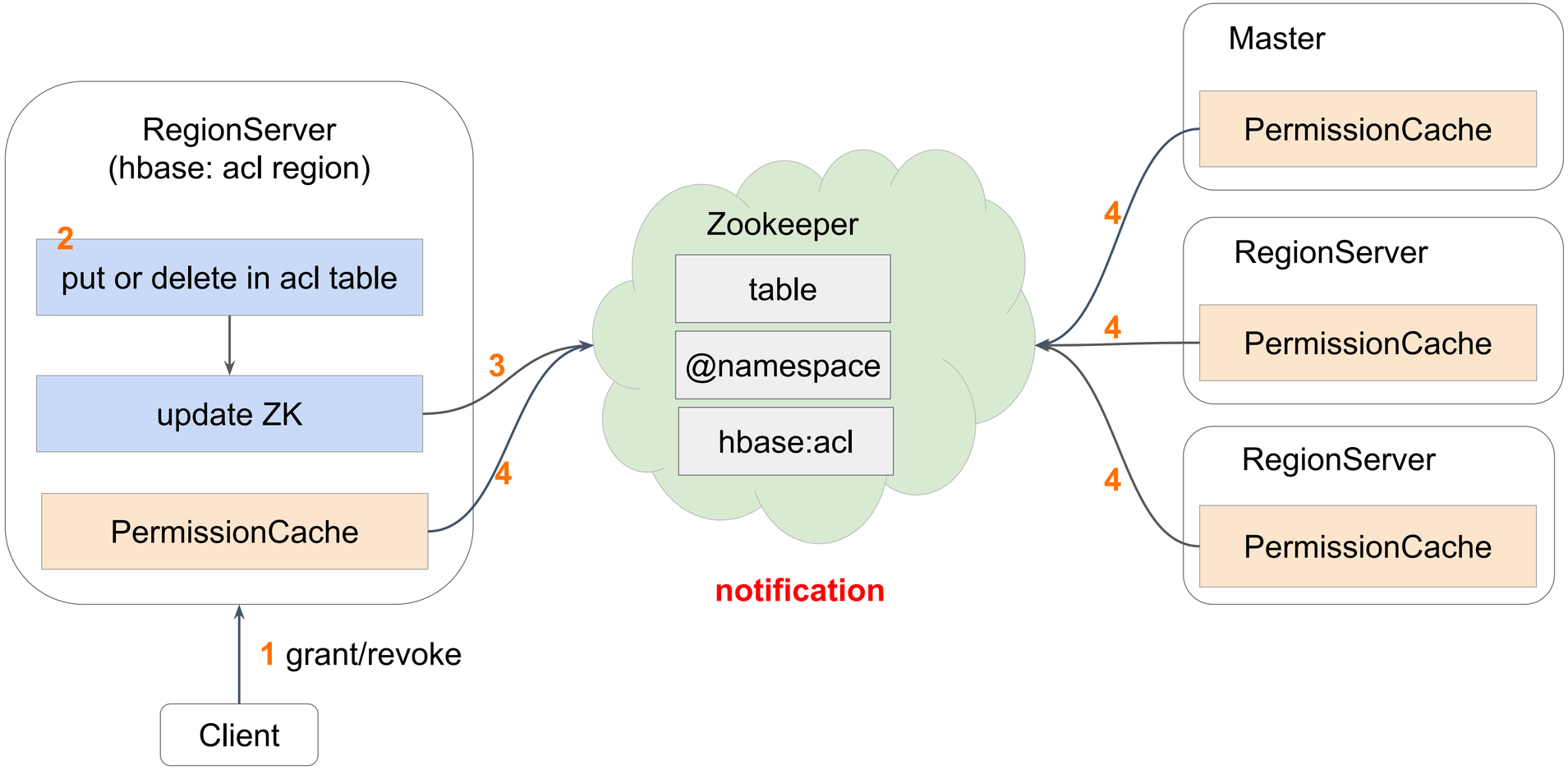
WAL Splitting based on ZK Coordination
WAL Splitting based on Procedure v2

ACL Overview

- Every server keeps permission cache.
- Check if operation is allowed by hooks of AccessController.



The Process of Grant/Revoke



Why use Procedure v2 instead of ZK?

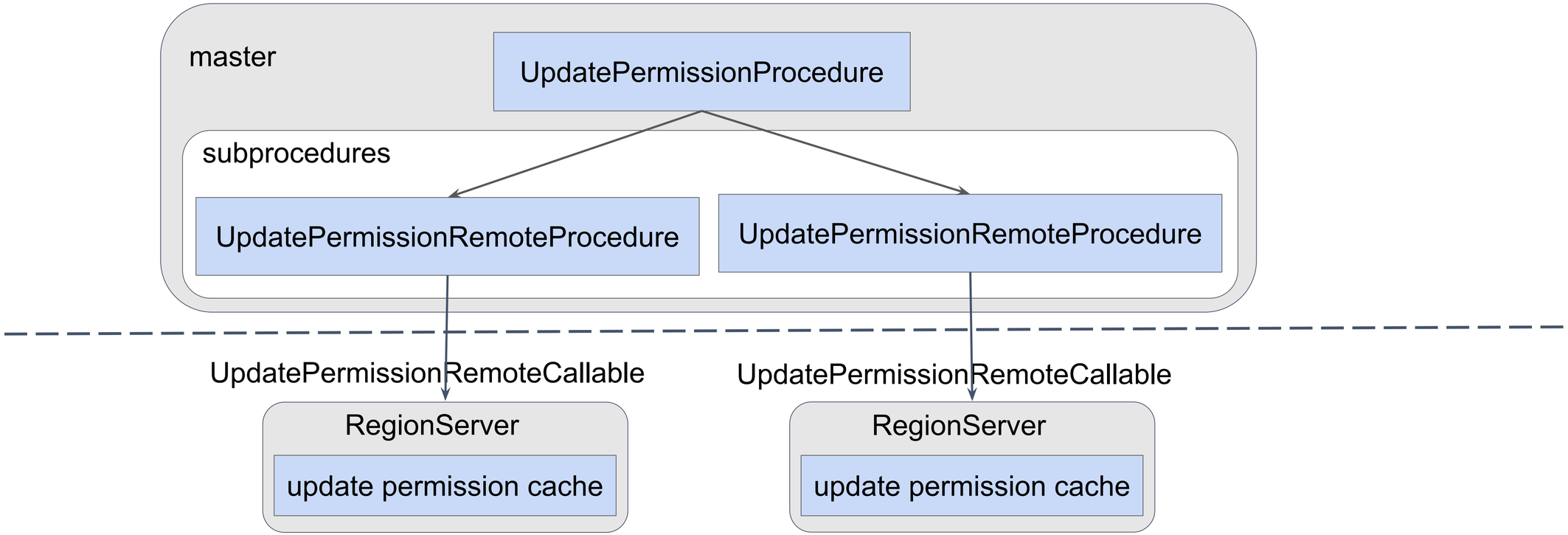
1. The permission cache may be inconsistent

2. Make grant/revoke a master method to implement HBASE-18659
   Use HDFS ACL to give user the ability to read snapshot directly on HDFS：
   Scan is expensive in HBase → SnapshotScanner→HBase users have no permission to access HDFS
   Basic idea is to add user ACLs to file → Make it a plugin →A master coprocessor → Send grant/revoke request to master

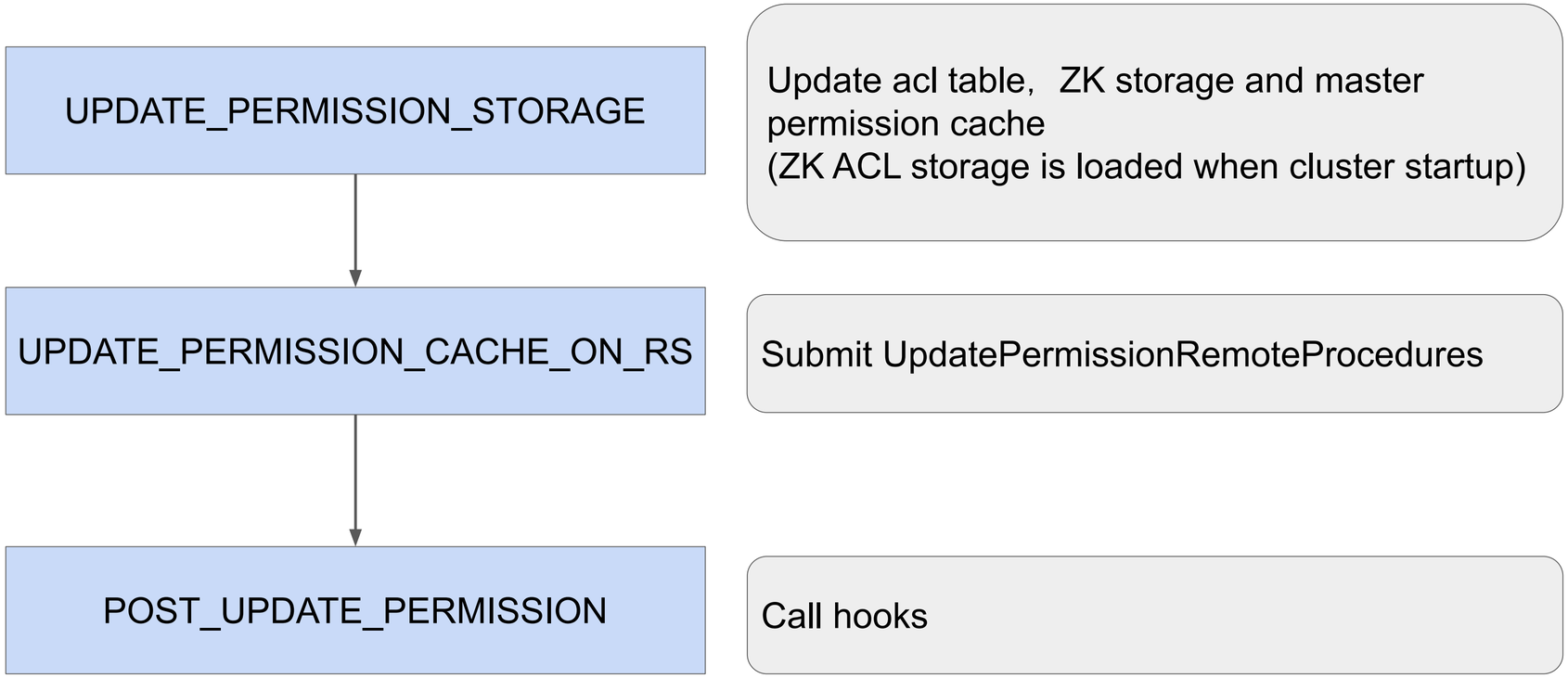
3. Procedure v2 provides the ability of notifications across of multi servers

ACL based on Procedure v2

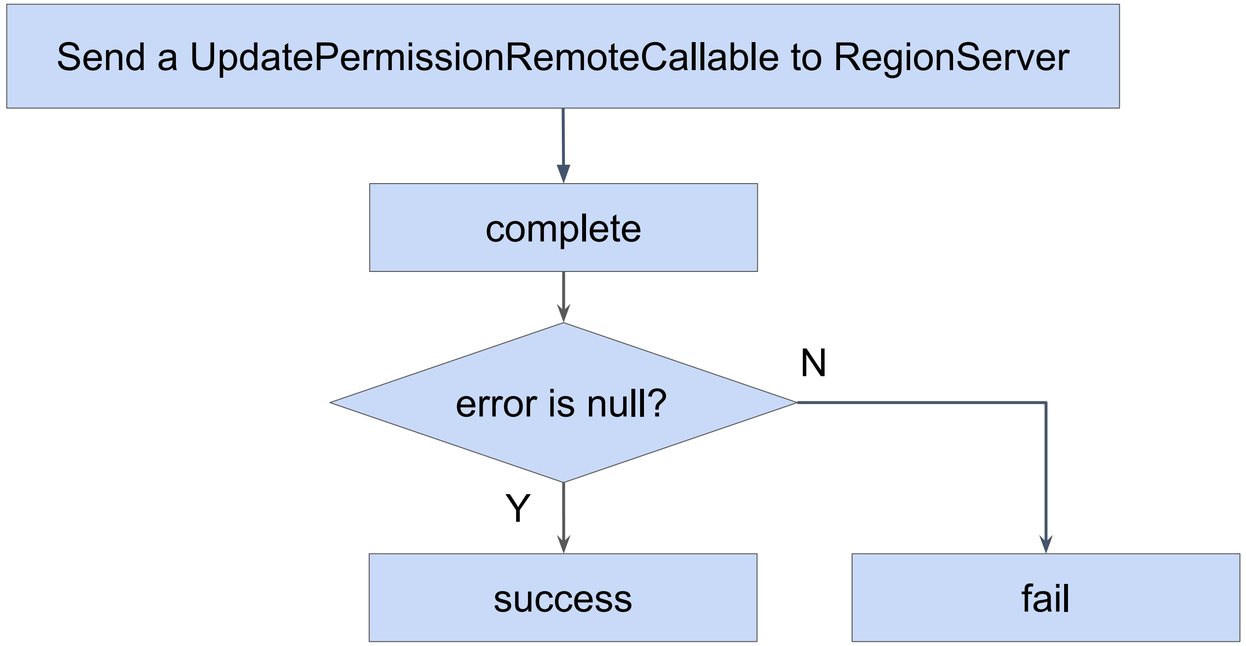
Client send grant/revoke request to master rather than RegionServer



UpdatePermissionProcedure



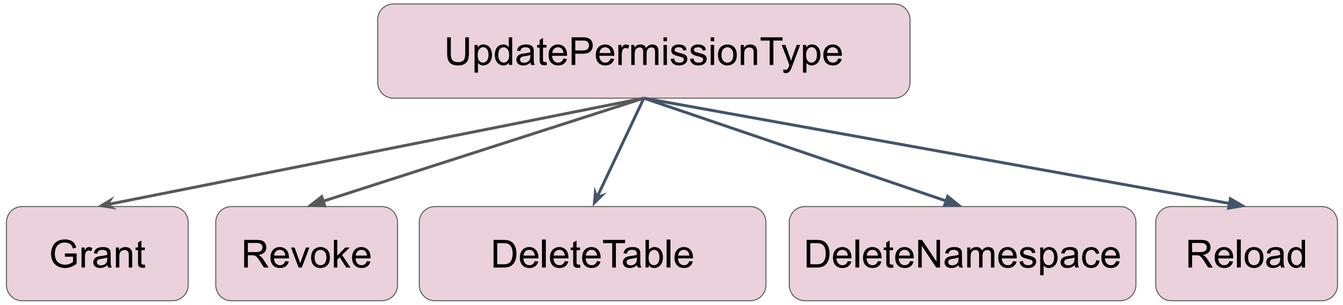
UpdatePermissionRemoteProcedure



Handle other cases of update ACL

- Delete namespace →Delete namespace permission cache
- Delete table →Delete table permission cache
- Master starts up, update ACL ZK storage to be consistent with acl table → Reload permission cache

  All use a UpdatePermissionProcedure



Abstract

❑ Introduction of Procedure v2

Overview
Execution and Rollback
Models

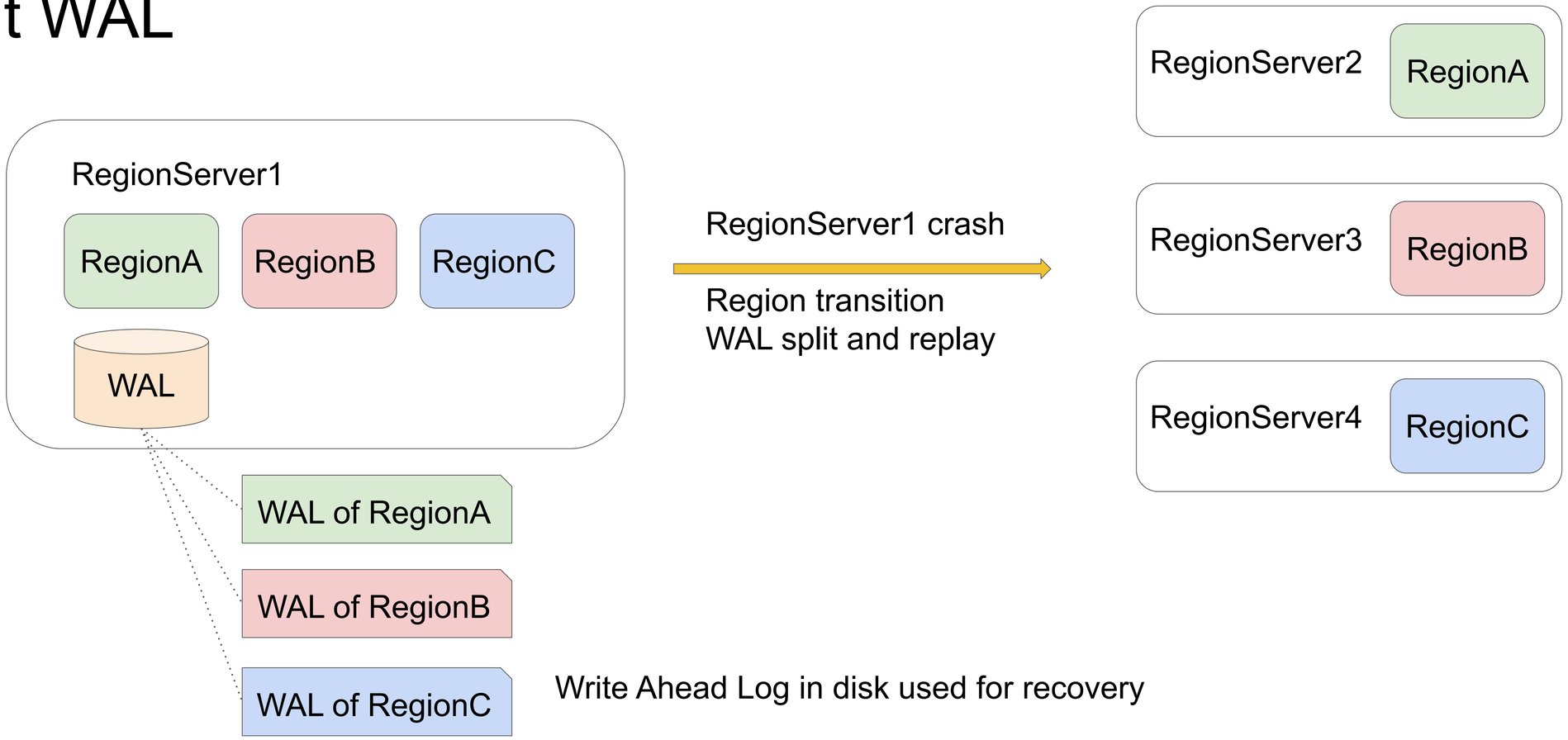
❑ ACL

ACL based on ZK Notification
ACL based on Procedure v2

❑ WAL Splitting

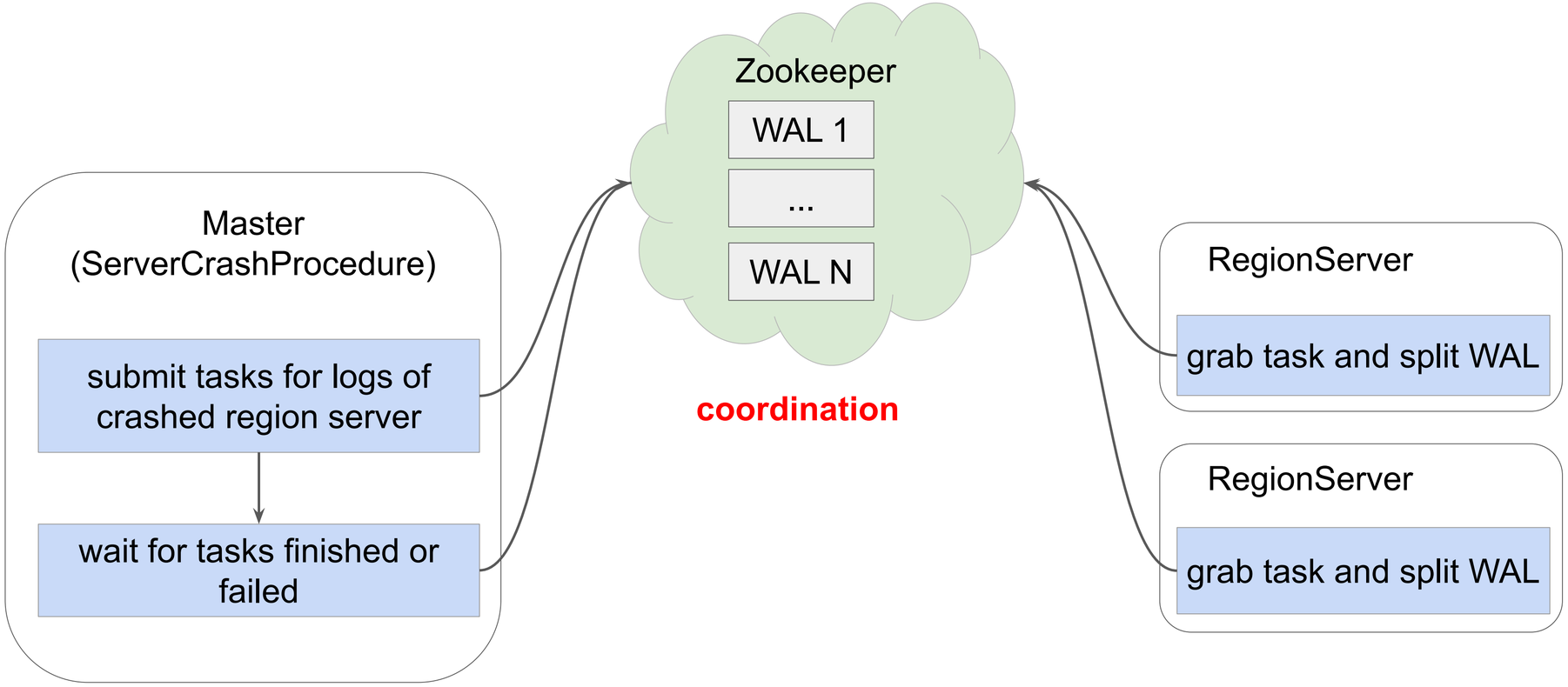
WAL Splitting based on ZK Coordination
WAL Splitting based on Procedure v2

Split WAL

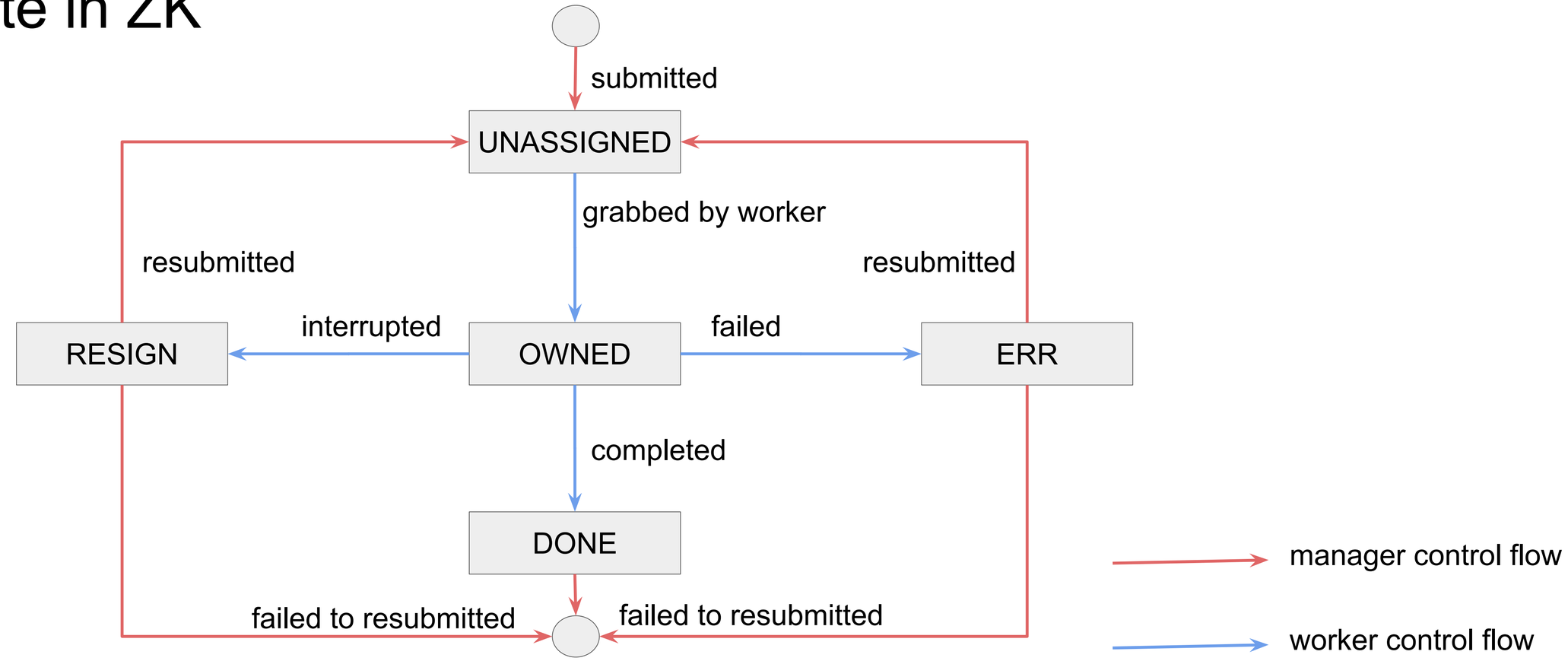


Write Ahead Log in disk used for recovery

Split WAL based on ZK Coordination



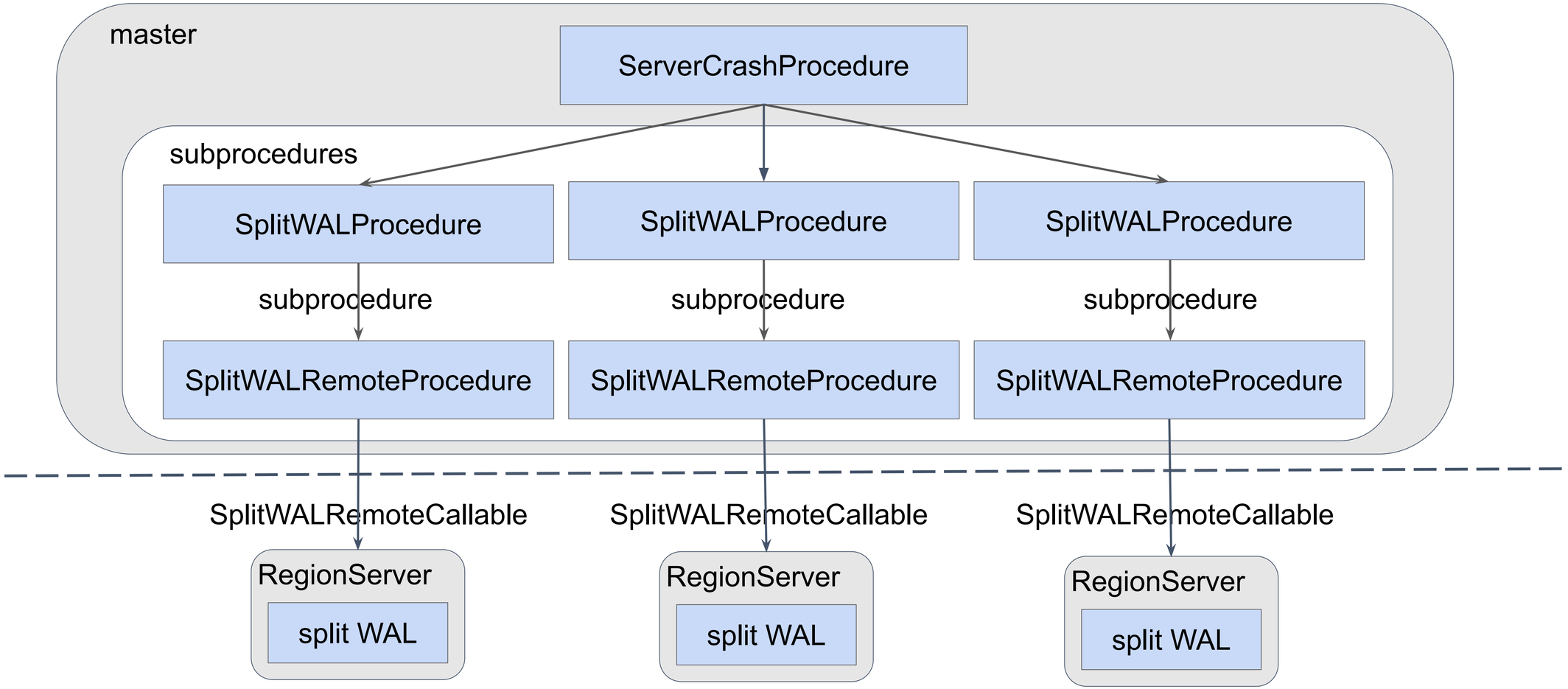
Task state in ZK



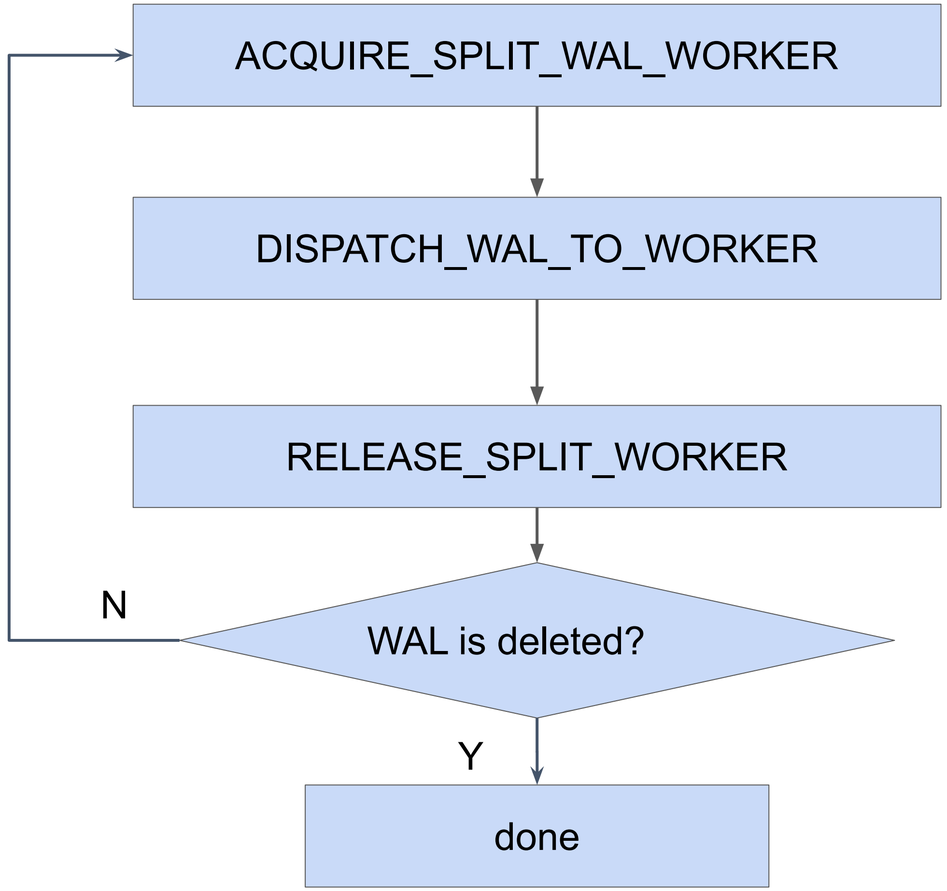
Why use Procedure v2 instead of ZK?

1. Reduce pressure to ZK
2. Reduce ZK dependency
3. Procedure v2 provides the ability to handle this case

Split WAL based on Procedure v2



SplitWALProcedure

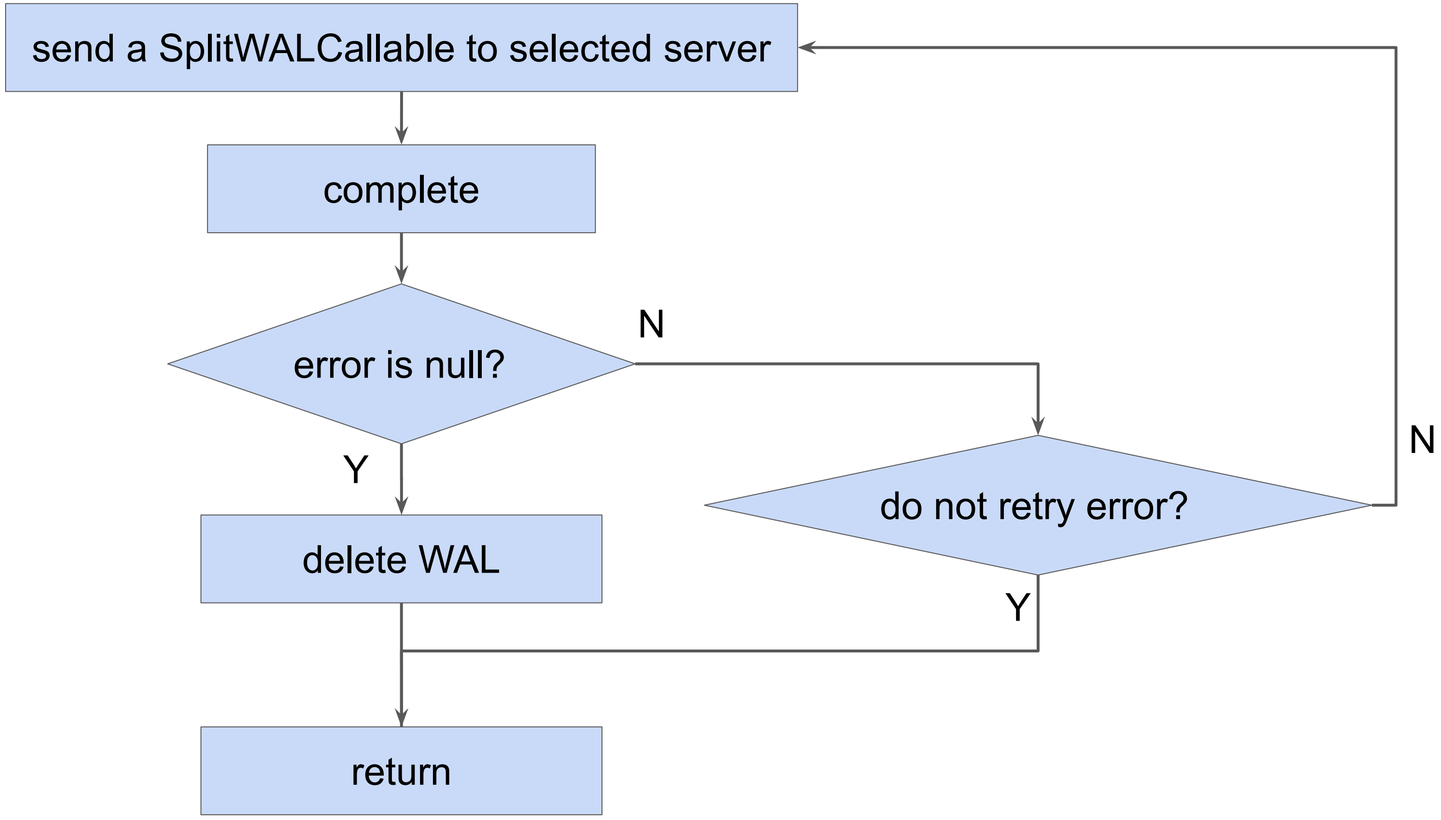


Choose a server to execute the split WAL task

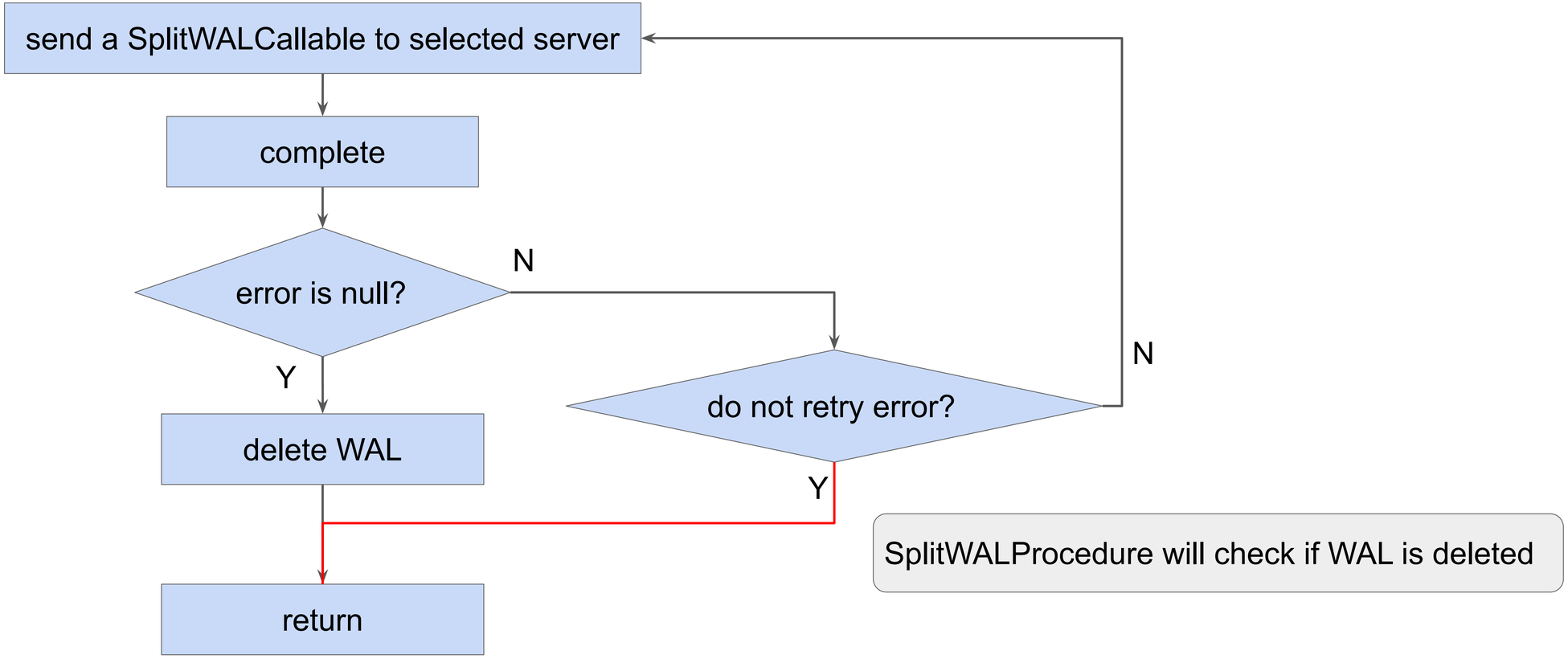
Submit a SplitWALRemoteProcedure

Release the server
Retry this procedure if split WAL is not finished

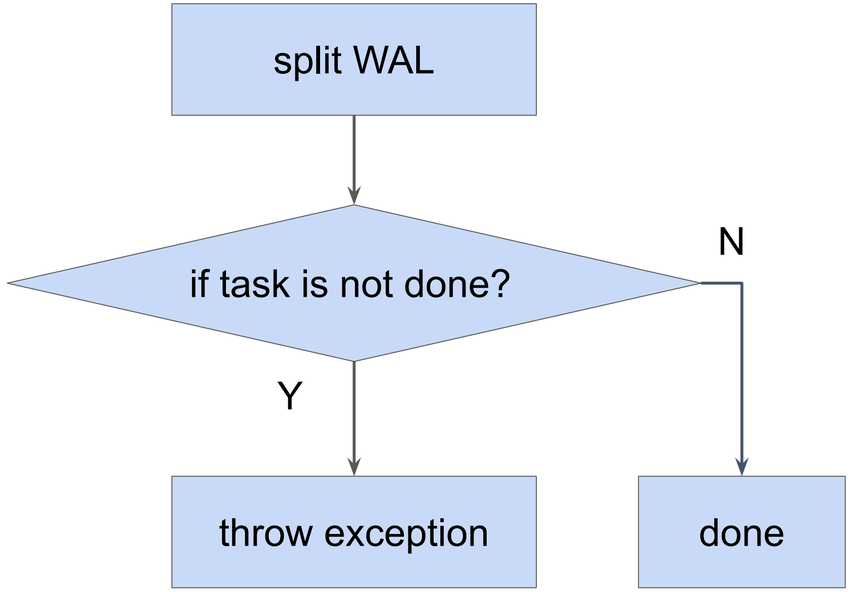
SplitWALRemoteProcedure



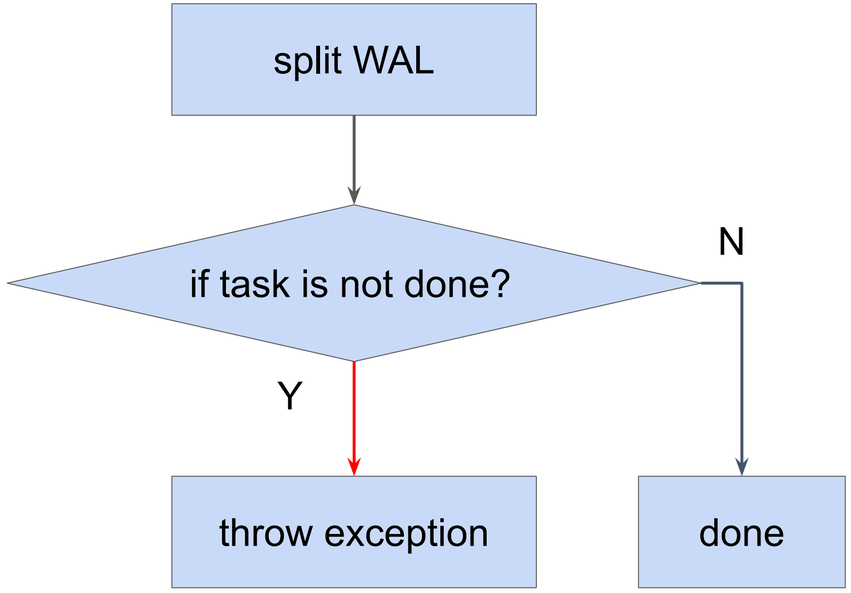
SplitWALRemoteProcedure



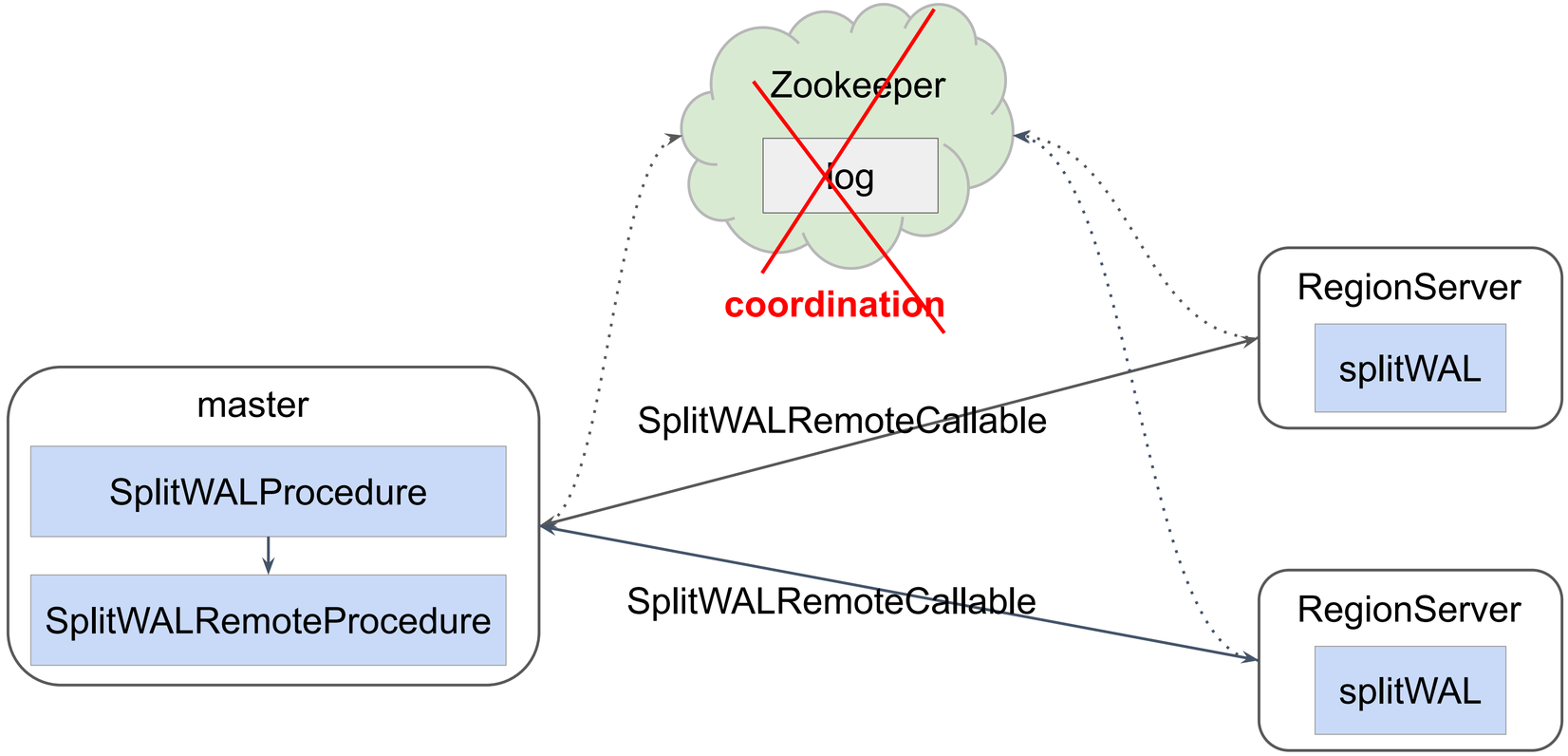
SplitWALCallable



SplitWALCallable



SplitWALRemoteProcedure will check this error


Zookeeper
log
coordination
RegionServer
splitWAL
master
SplitWALProcedure
SplitWALRemoteProcedure
SplitWALRemoteCallable
SplitWALRemoteCallable
RegionServer
splitWAL

Performance

1 master and 5 regionservers.

- Restart whole cluster

Num of WALs	before(cost time in ms)	after(cost time in ms)	Reduce time
181	57595	52258	9.27%
382	118193	96766	18.13%

- Restart 1 region server

Num of WALs	before(cost time in ms)	after(cost time in ms)	Reduce time
38	19606	16581	15.43%
78	35536	32710	7.95%

How to enable WAL splitting based on Procedure v2?

1. Upgrade the whole cluster to the package with the new implementation(2.2.0+)
2. Set **hbase.split.wal.zk.coordinated=false**, and upgrade master
3. Upgrade region servers

Thanks !