Fabric Lifecycle Playback

2019/01/13

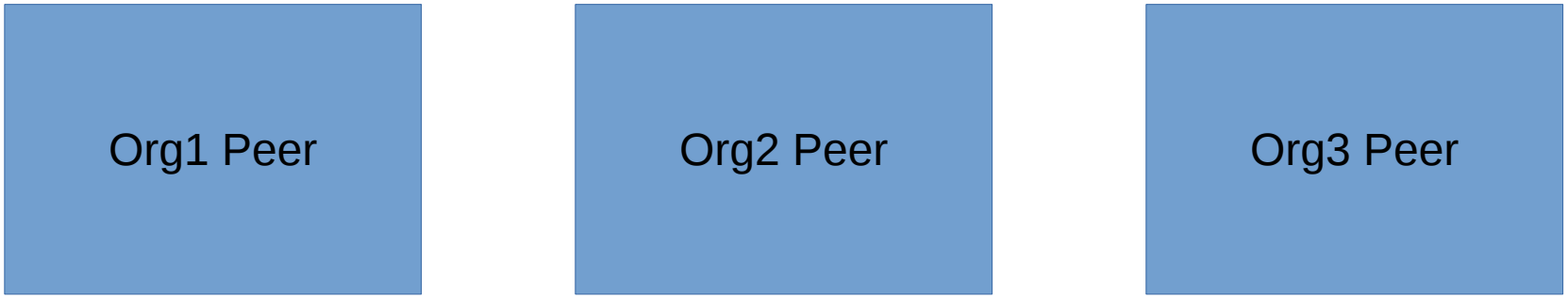
User Perspective

- Scenario:
  - Organizations: Org1, Org2, Org3
  - Everyone wants to run some code as version ‘v1.0’ of ‘mycc’ with endorsement policy of “2 out of 3 orgs must agree”.

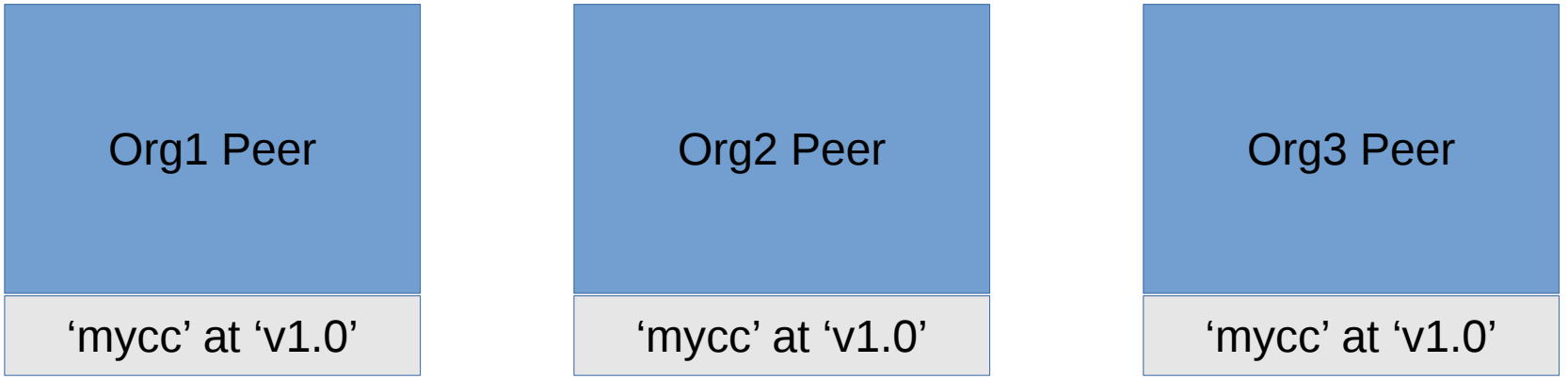
Workflow

- Administrator of Org1:
  - Packages chaincode into ‘mycc-v1.0-package.tar’
  - Disseminates the package (to Org2, Org3).
- Administrators of Org2, Org3 inspect the package using standard tools.
- Administrators of Org1, Org2, Org3:
  - Install the chaincode package.
  - Define the chaincode for their org.
- Adminstrator of Org1 defines the chaincode for the channel.

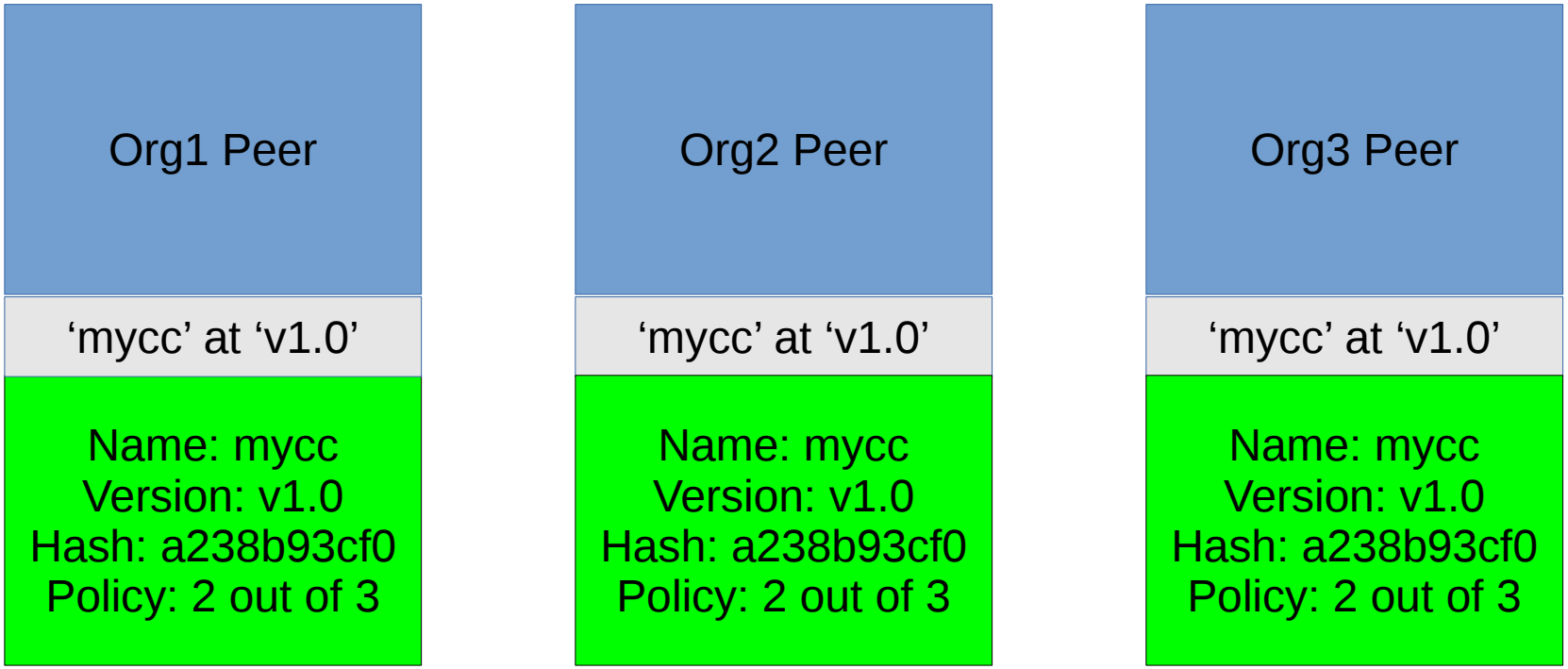
Network of 3 Peers



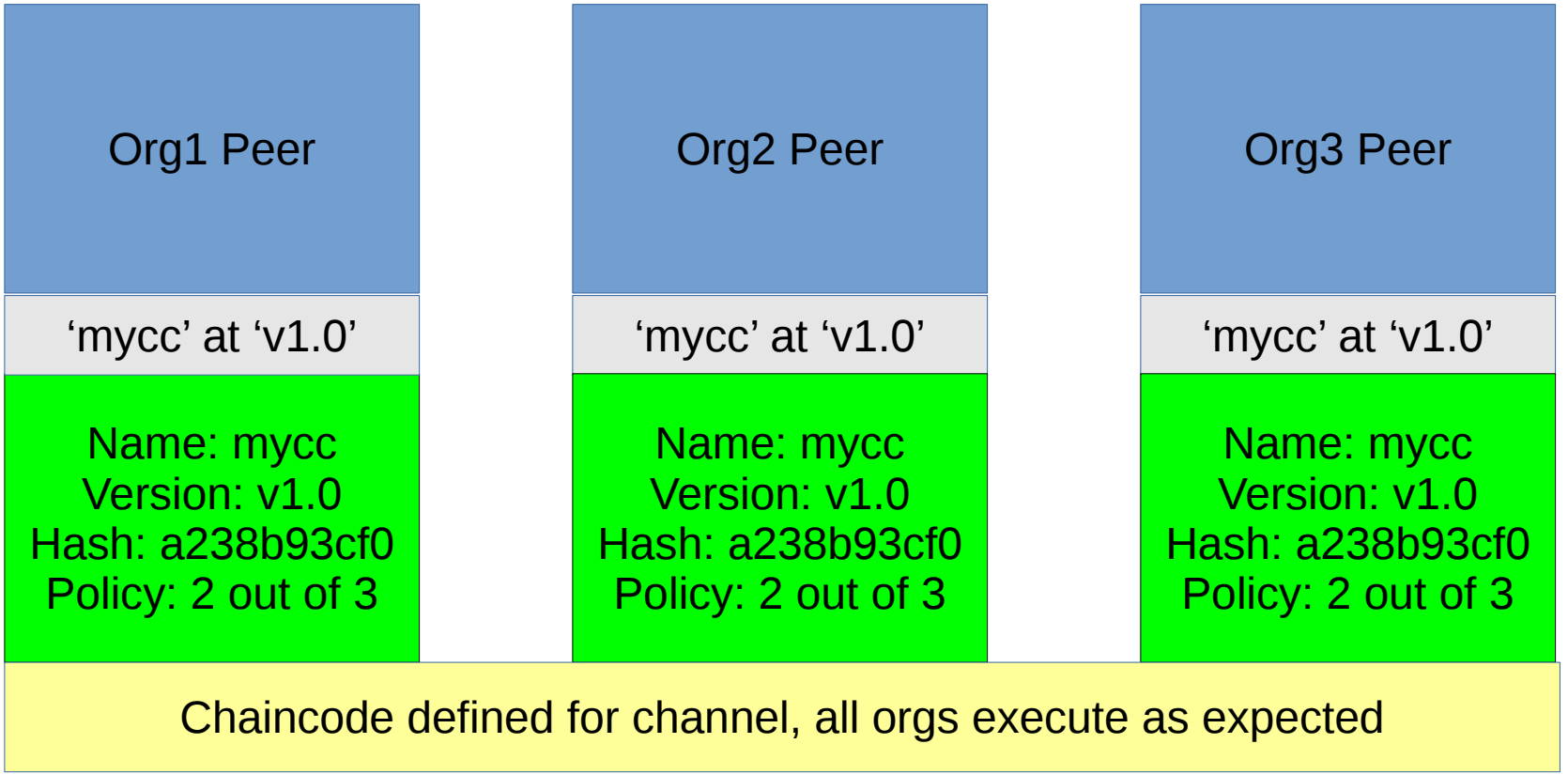
Administrators Install Chaincode



Administrators DefineForMyOrg

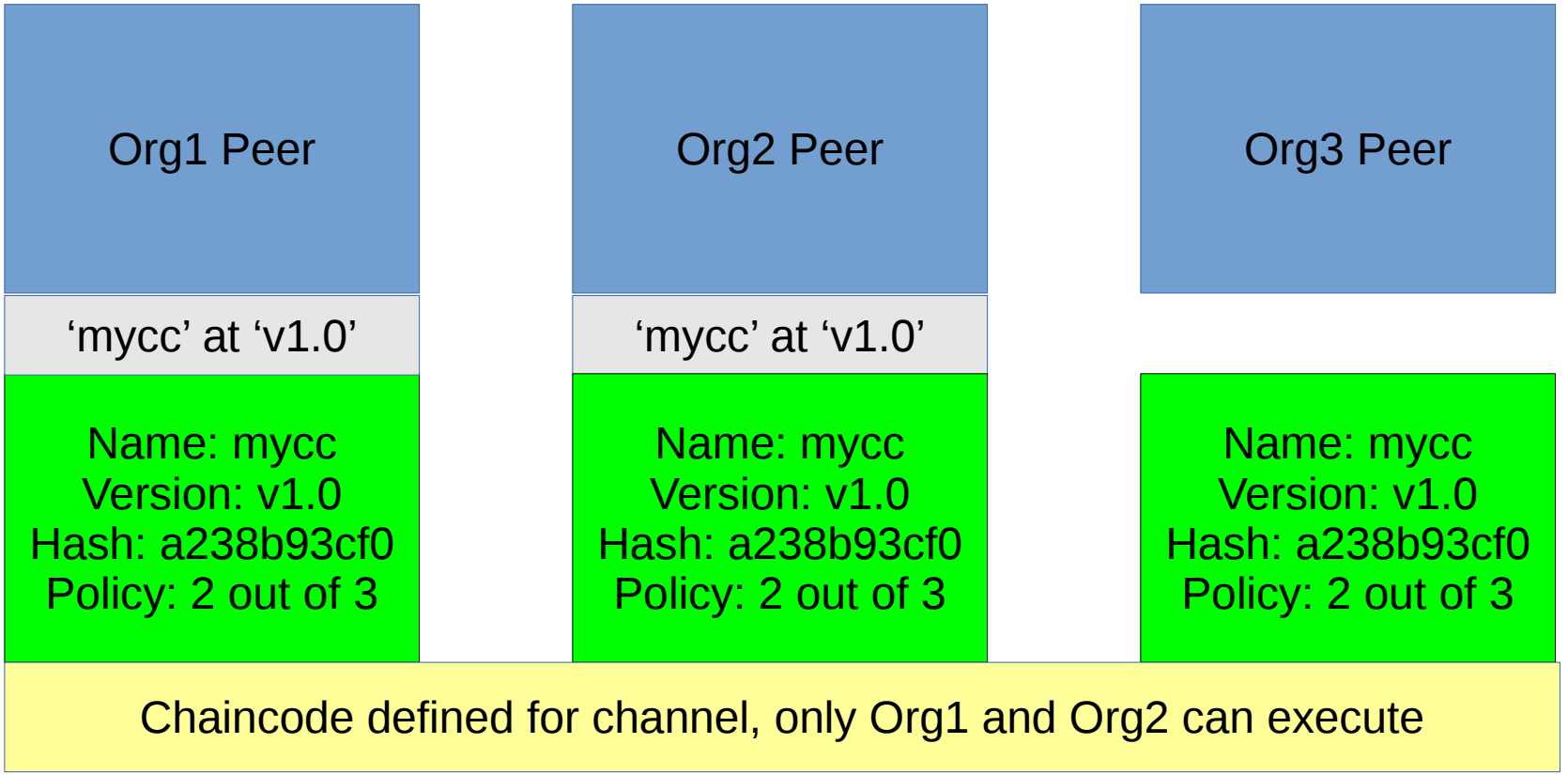


Administrator Invokes Define

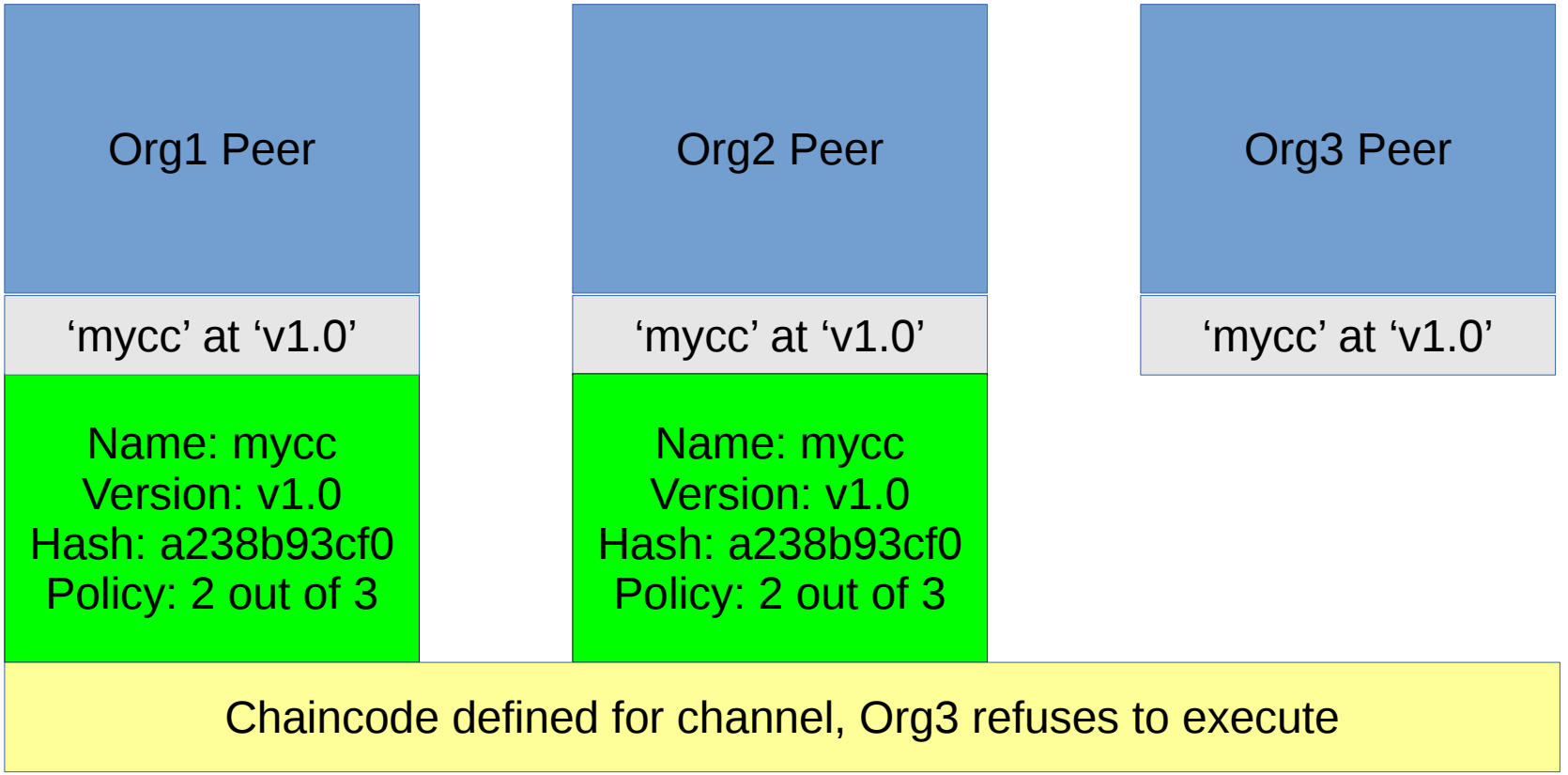


Lifecycle Permutations

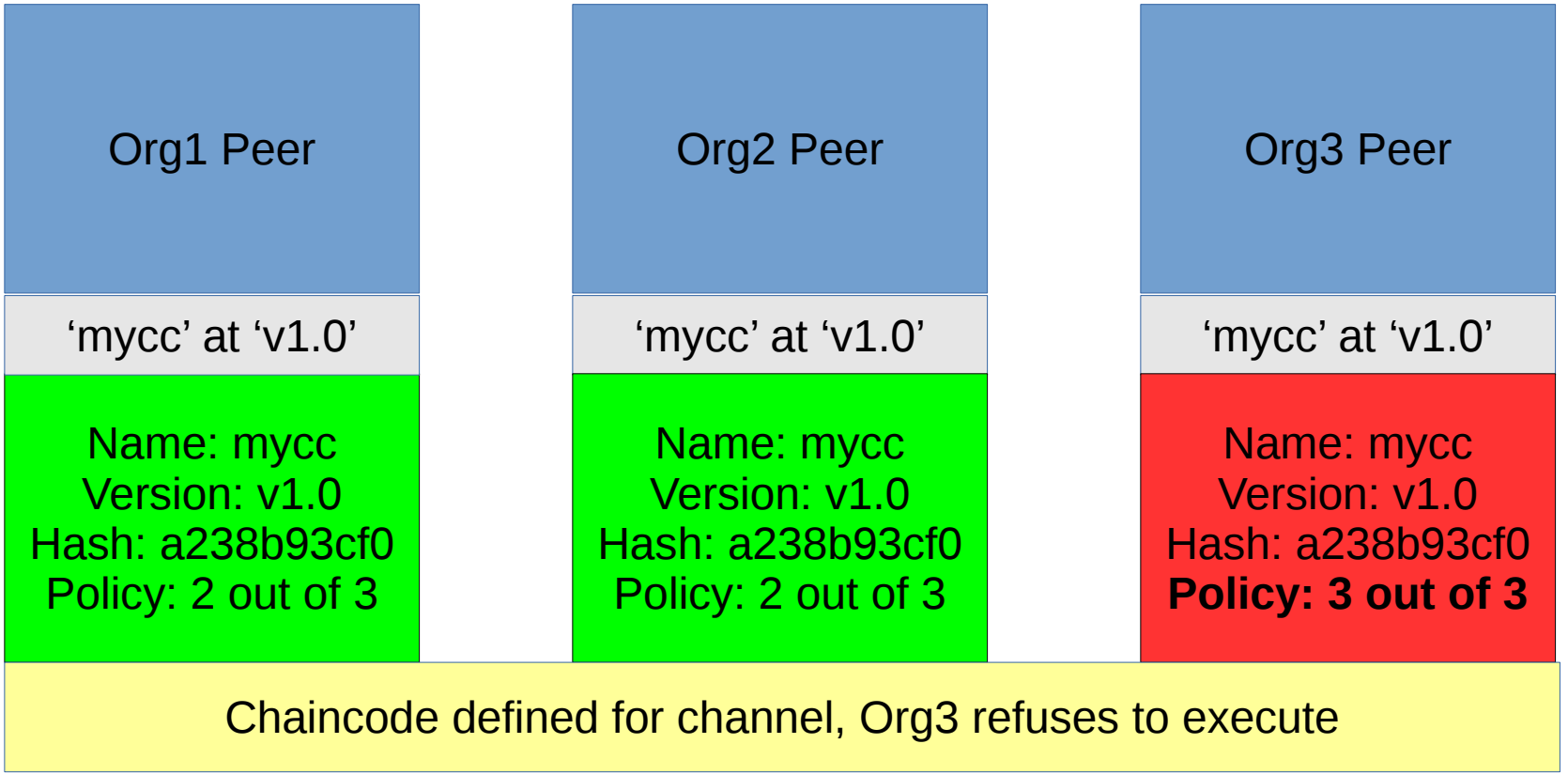
One admin does not install



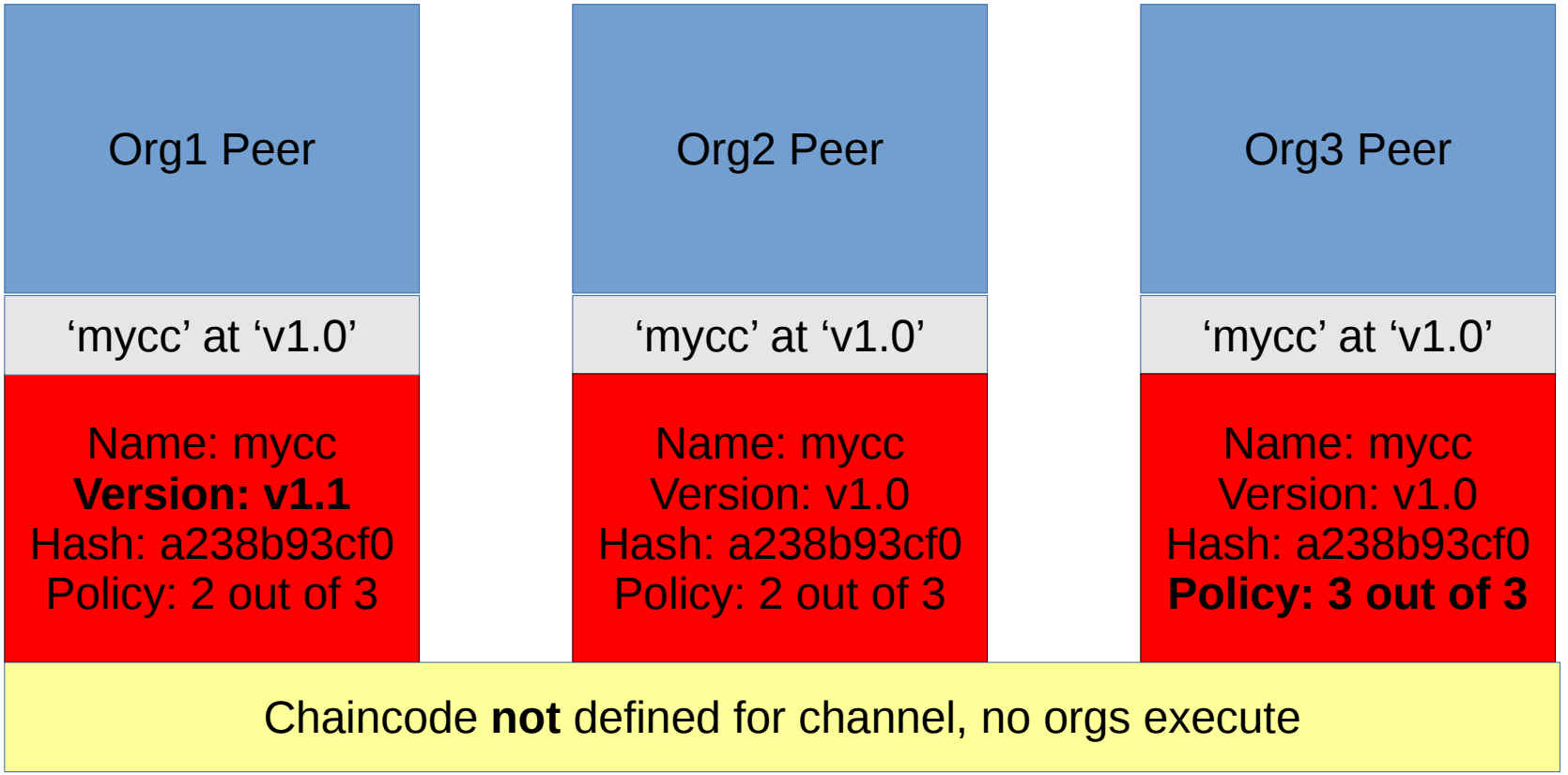
One admin does not define



Admin does not agree to policy



Admins do not agree to parameters

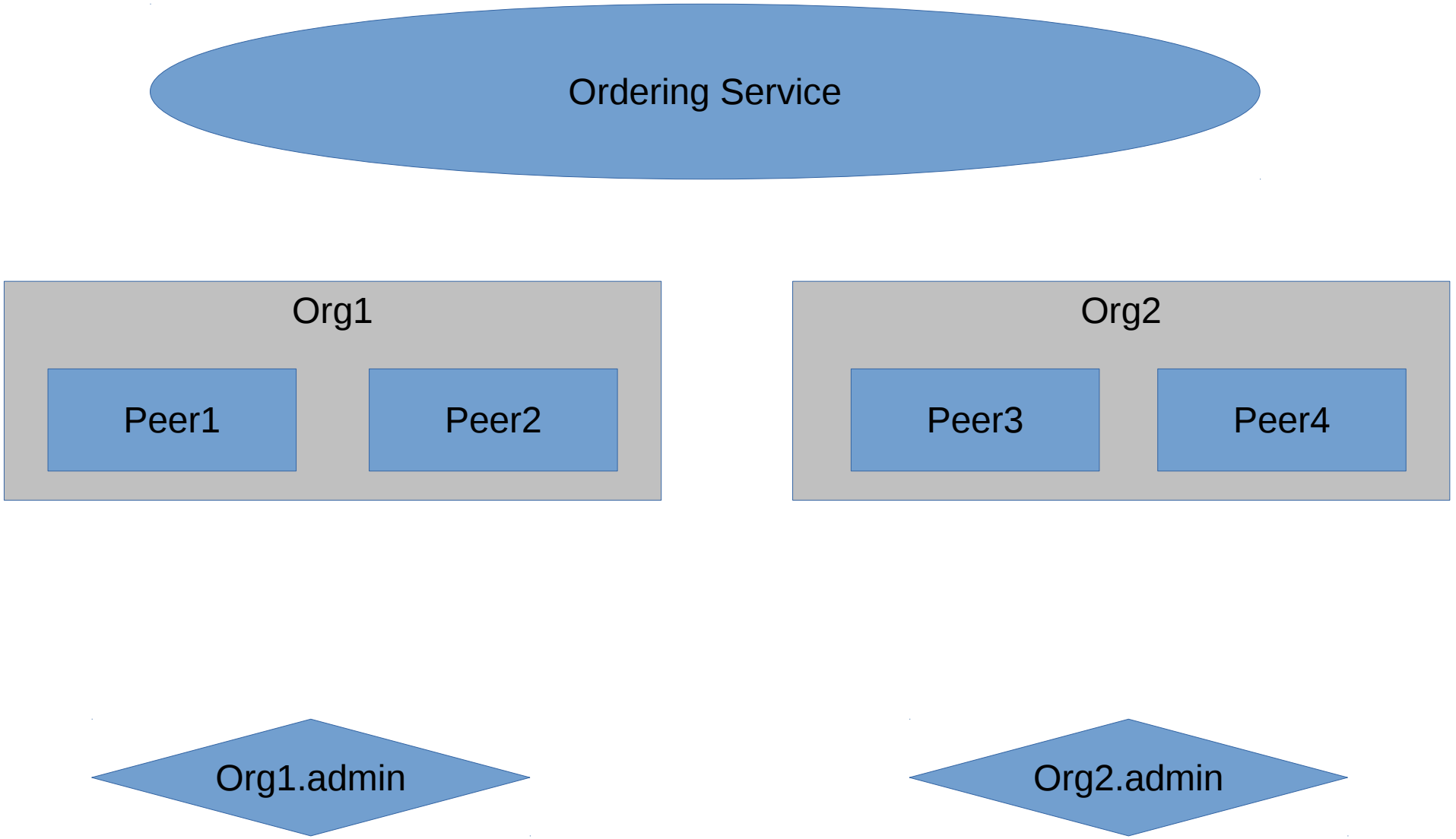


Peer CLI Commands

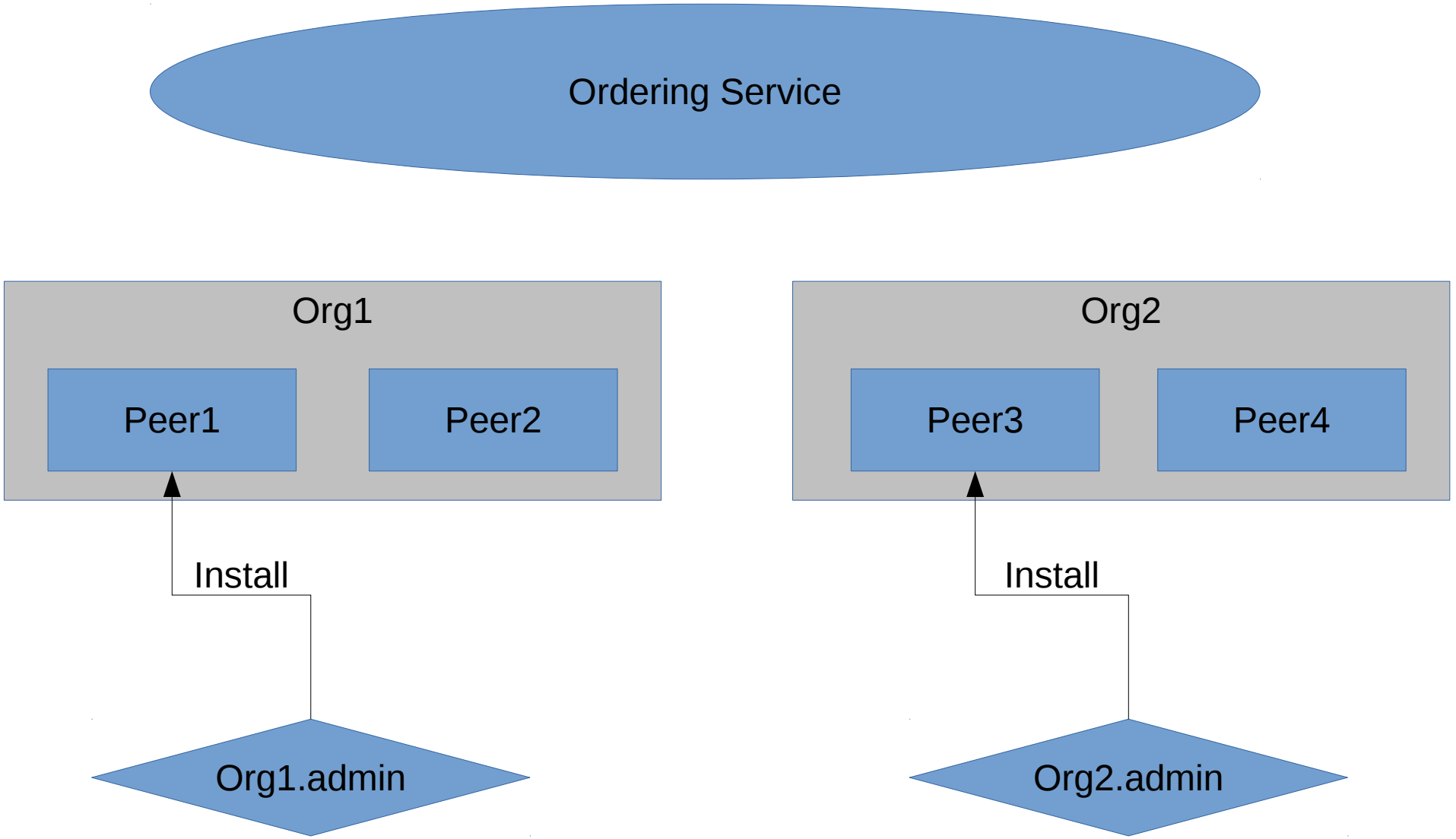
- Package
  - `peer chaincode package fabcar-v1.0-package.tar.gz --newLifecycle --path github.com/chaincode/fabcar/go/ --lang golang`
- Install
  - `peer chaincode install fabcar-v1.0-package.tar.gz --newLifecycle --name fabcar --version 1.0`
- DefineForMyOrg
  - `peer chaincode defineformyorg --channelID mychannel --name fabcar --version 1.0 --hash 3a69..35e1 --sequence 1 --policy "AND ('Org1MSP.member','Org2MSP.member')"`
- Define for channel
  - `peer chaincode define --channelID mychannel --name fabcar --version 1.0 --hash 3a69..a35e1 --sequence 1 --policy "AND ('Org1MSP.member','Org2MSP.member')"`

Begin Backup

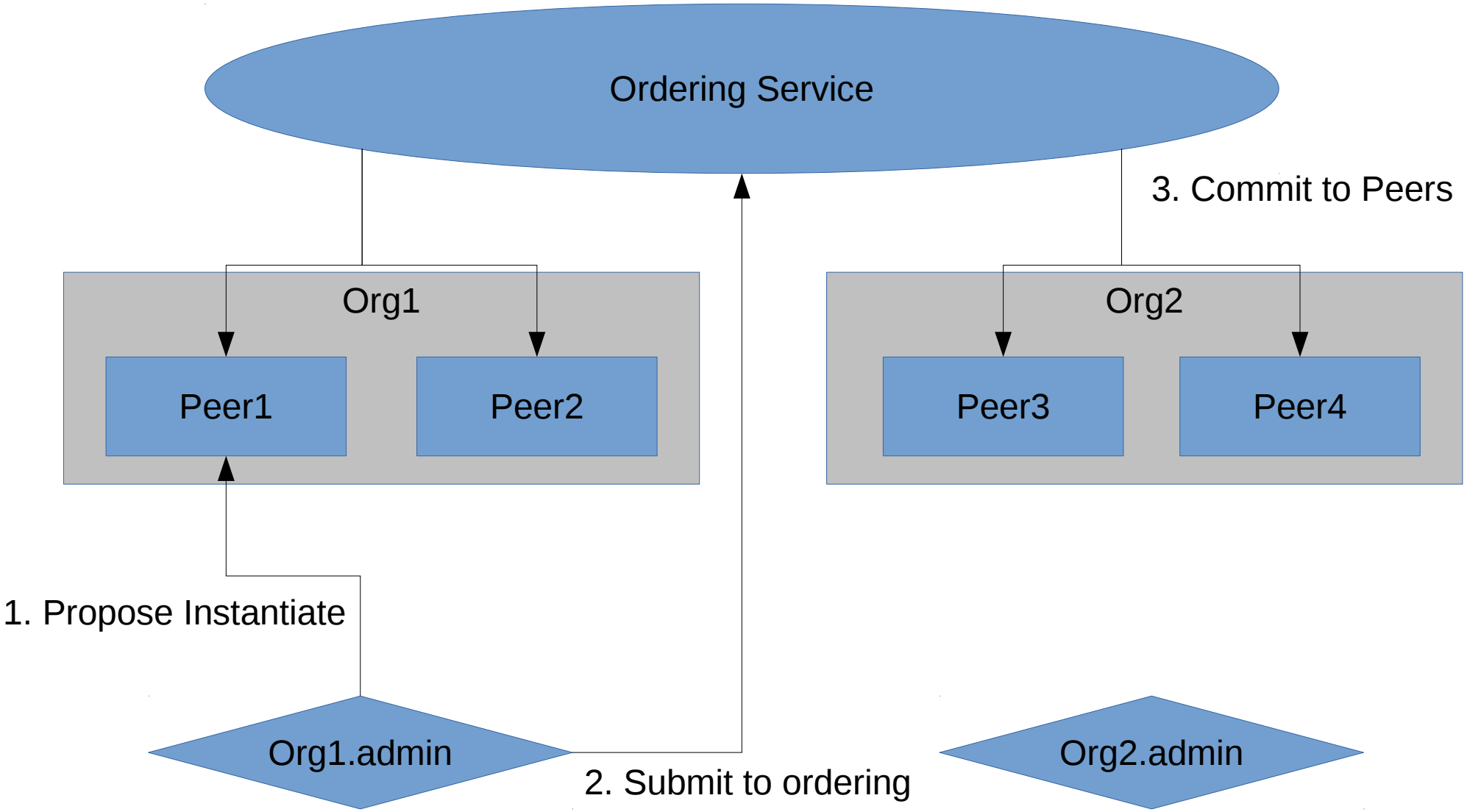
Fabric v1.0/v1.1/v1.2 Lifecycle



Fabric v1.0/v1.1/v1.2 Lifecycle



Fabric v1.0/v1.1/v1.2 Lifecycle



Notable Problems

- Installing chaincode agrees to instantiate unconditionally
  - Org2.admin has no say in endorsement policy
  - Org2.admin has no say in upgrade/downgrade
  - Org2.admin has no say in his own peer’s execution
- Org2 does not endorse instantiation
  - Only Org1’s peers validate chaincode parameters
  - Only Org1’s peers execute Init, contrary to endorsement policy
- Changing any parameter is a ‘chaincode upgrade’
  - Modifying collections
  - Modifying endorsement policy

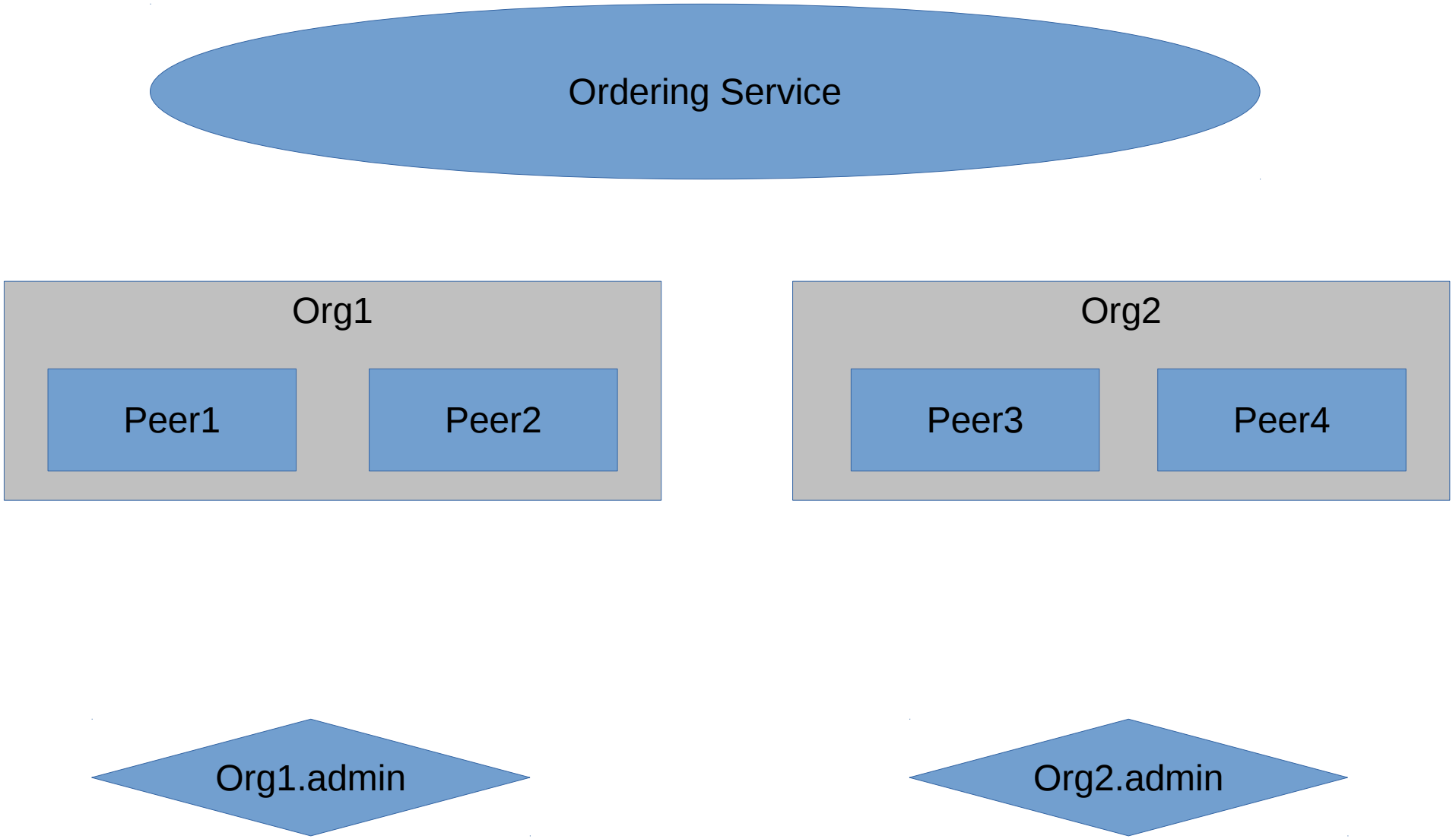
Fabric v1.3 Lifecyle Principles

- Peers only participate in chaincode if Org admin agrees
  - Will not endorse instantiation
  - Will not endorse any chaincode invoke, including Init
- No exceptions to endorsement policies
  - Chaincode definition agreed to by endorsement
  - Chaincode Init executes as normal invoke
- Redefining parameters does not require upgrade
  - Update collections and EP without new install

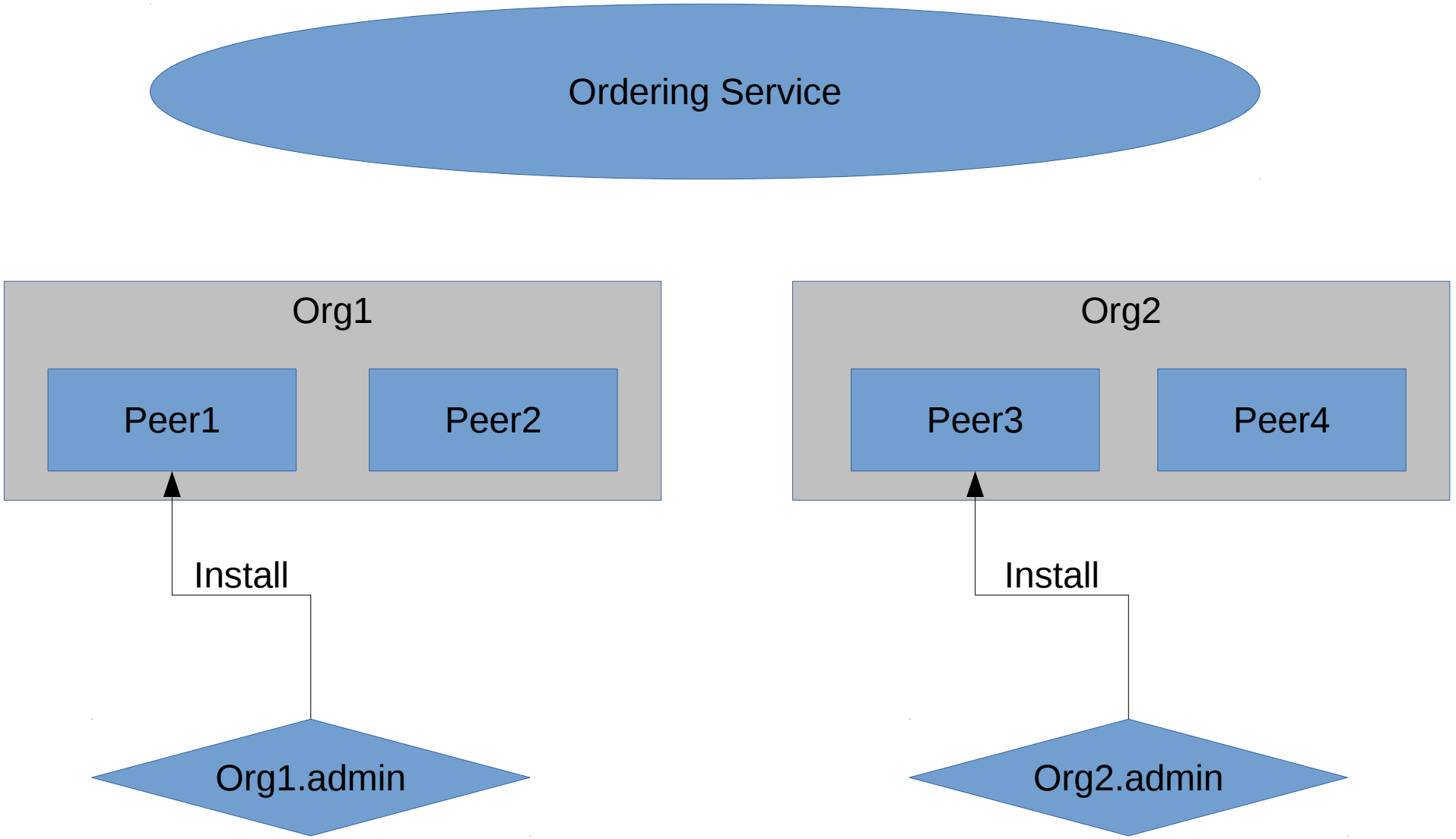
Fabric v1.3 Lifecycle Prereq

- Organizaiton Implicit Collections
  - Like a standard private data collection
  - Only peers from an org may write to it
  - Only peers from an org may read private data
  - Same chaincode may execute differently by organization based on collection data
- Usable for ‘voting’ outside of lifecycle
  - Collection name in reserved collection namespace: _implicit_org_<MSPID>

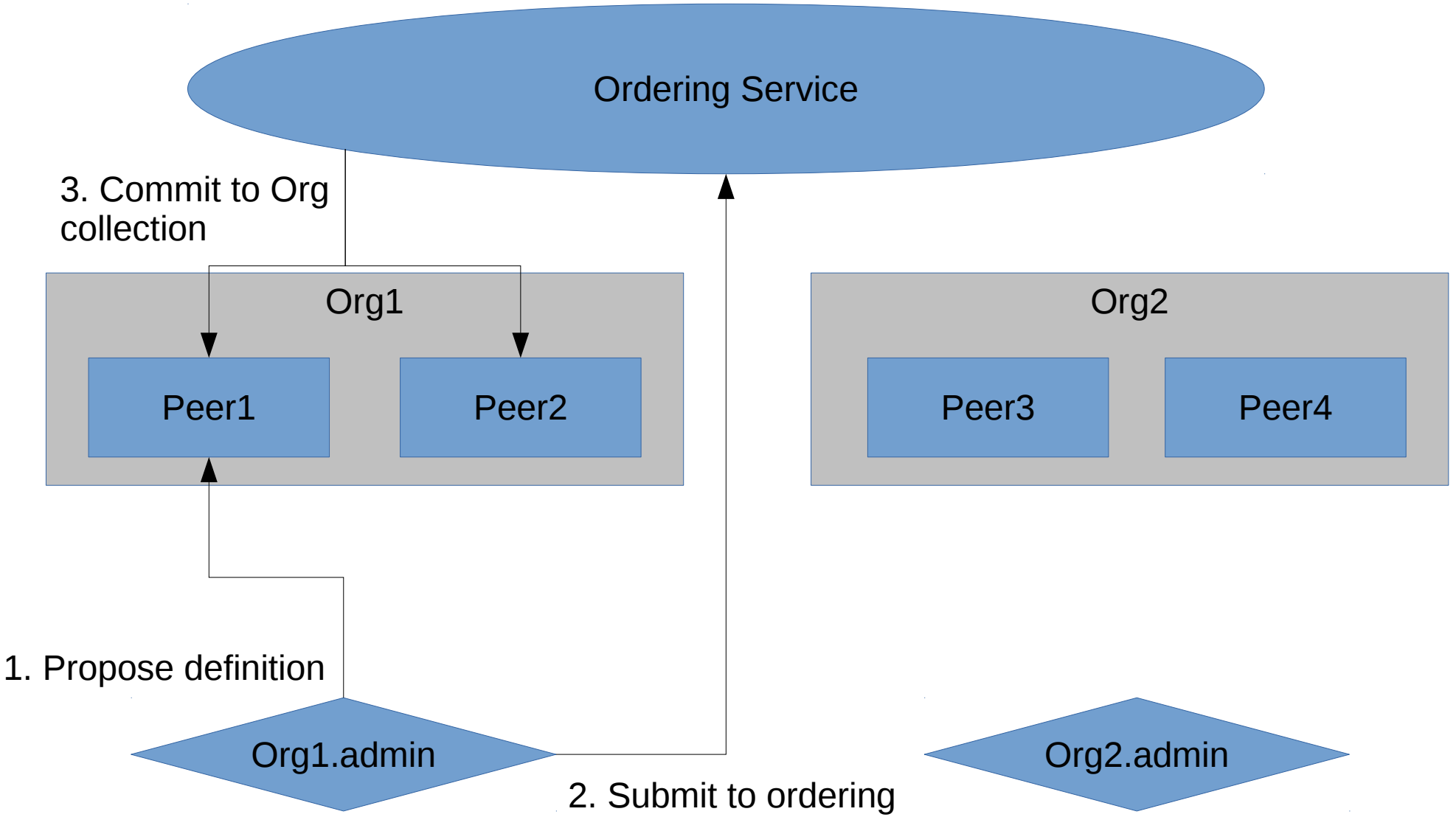
Fabric v1.3 Lifecycle



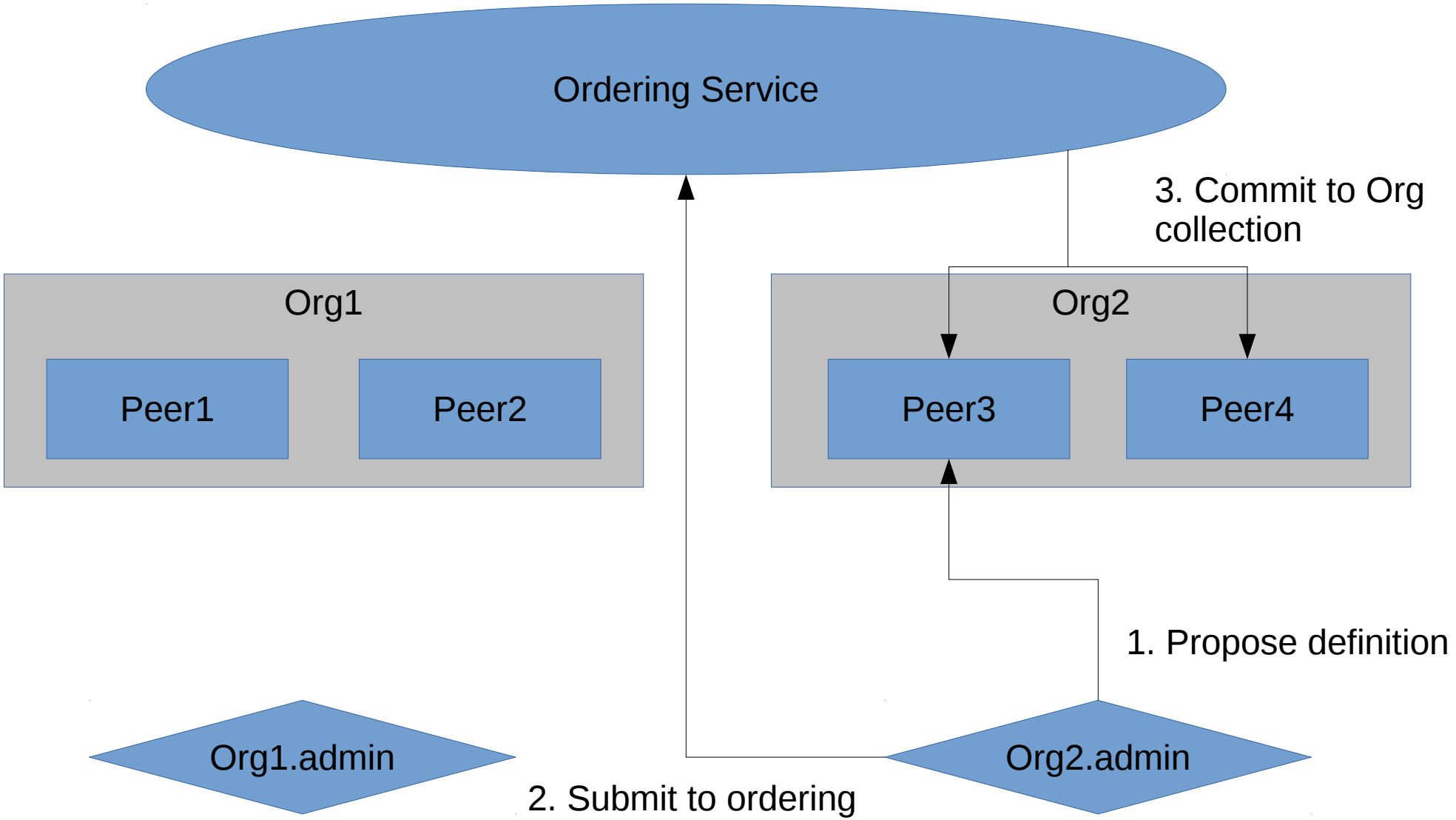
Fabric v1.3 Lifecycle



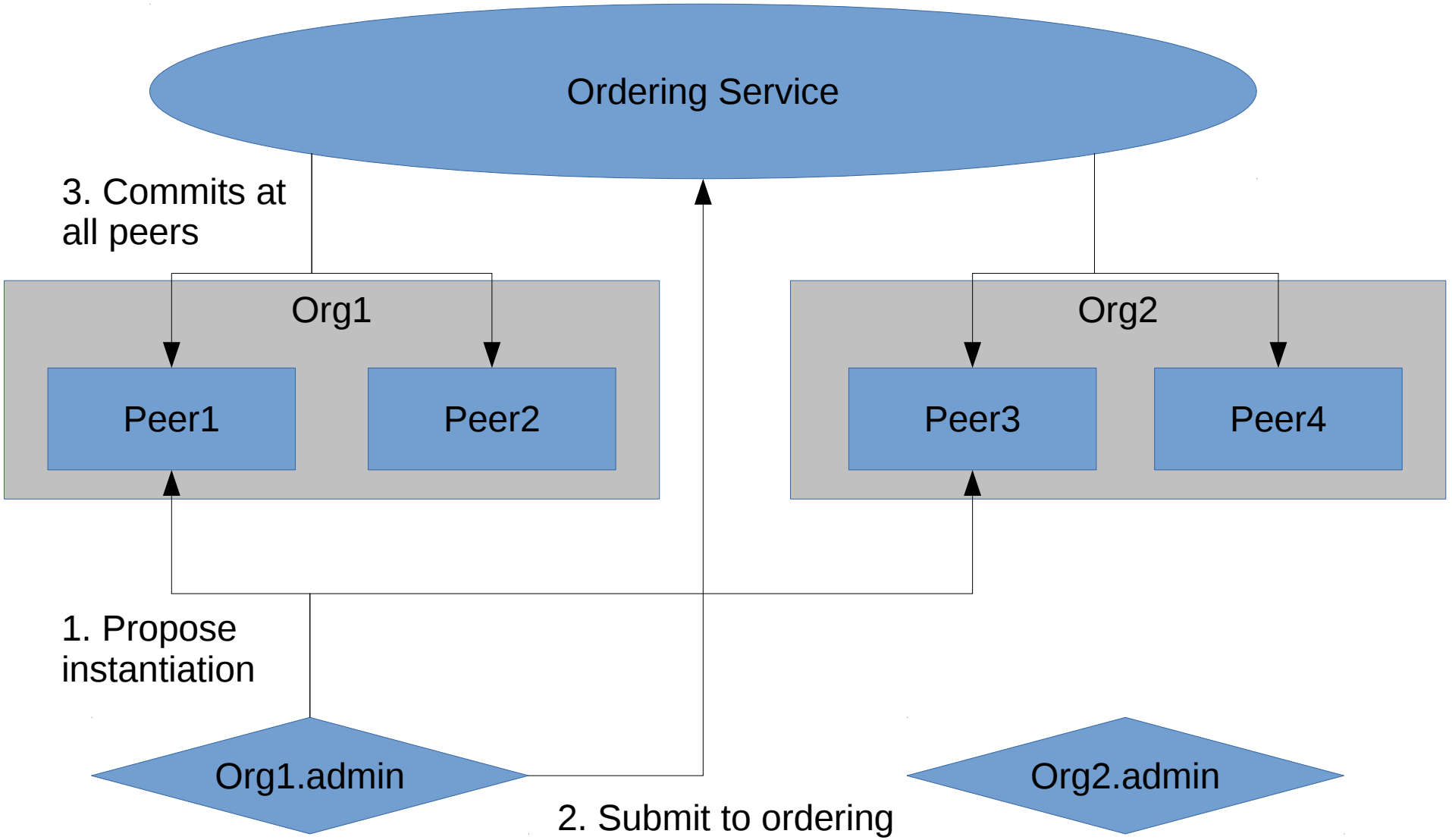
Fabric v1.3 Lifecycle



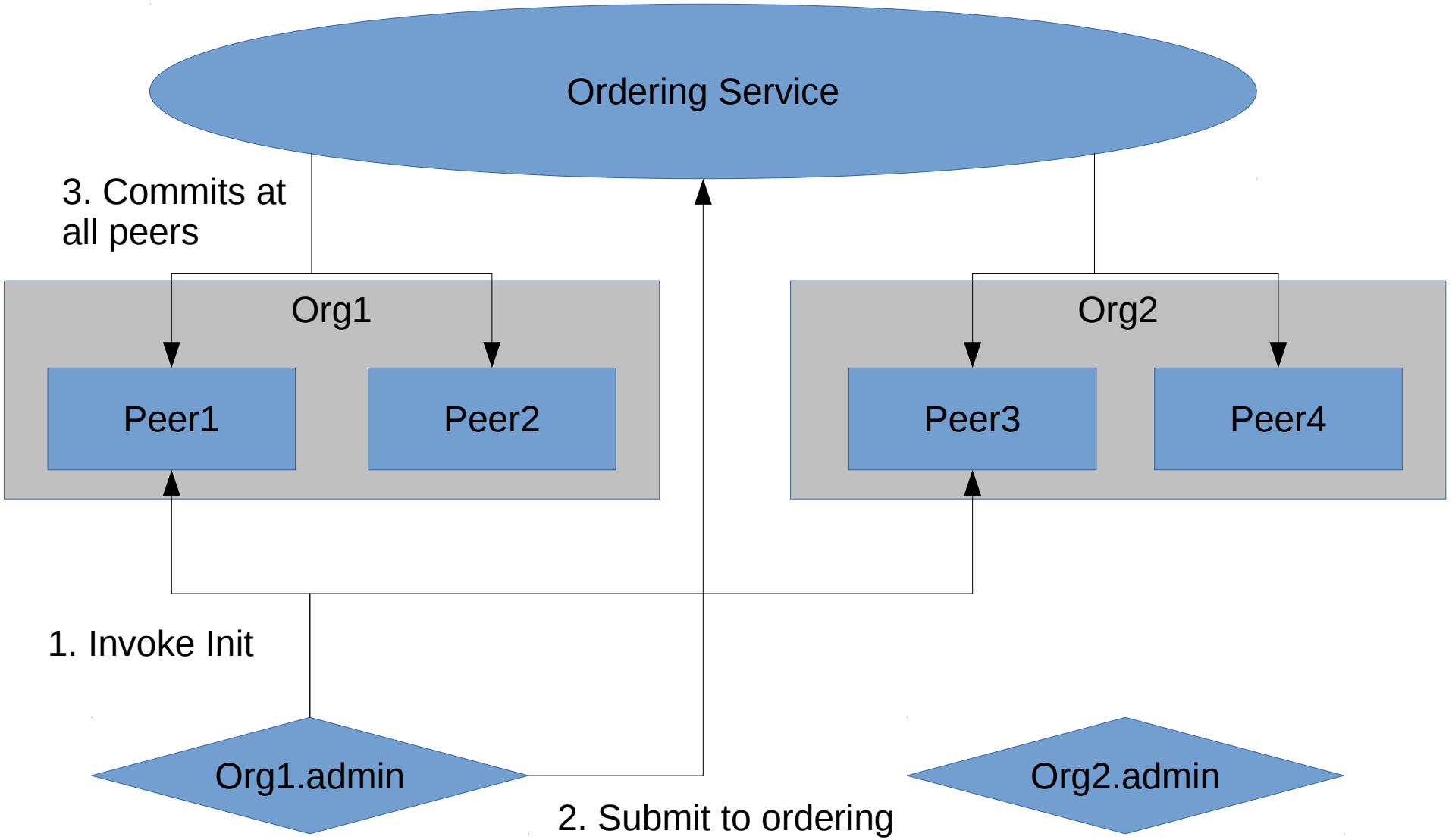
Fabric v1.3 Lifecycle



Fabric v1.3 Lifecycle



Fabric v1.3 Lifecycle



Lifecycle Rollout

- New lifecycle chaincode and namespace `_lifecycle`
- Fabric v2.0 will support both legacy and new lifecycle
  - Prefer chaincode definition from new lifecycle, fallback to chaincode definition in old lifecycle
- Fabric v2.1 will remove legacy support
  - Users should move to new chaincode lifecycle during v2.0 to avoid any interruption of service in v2.1