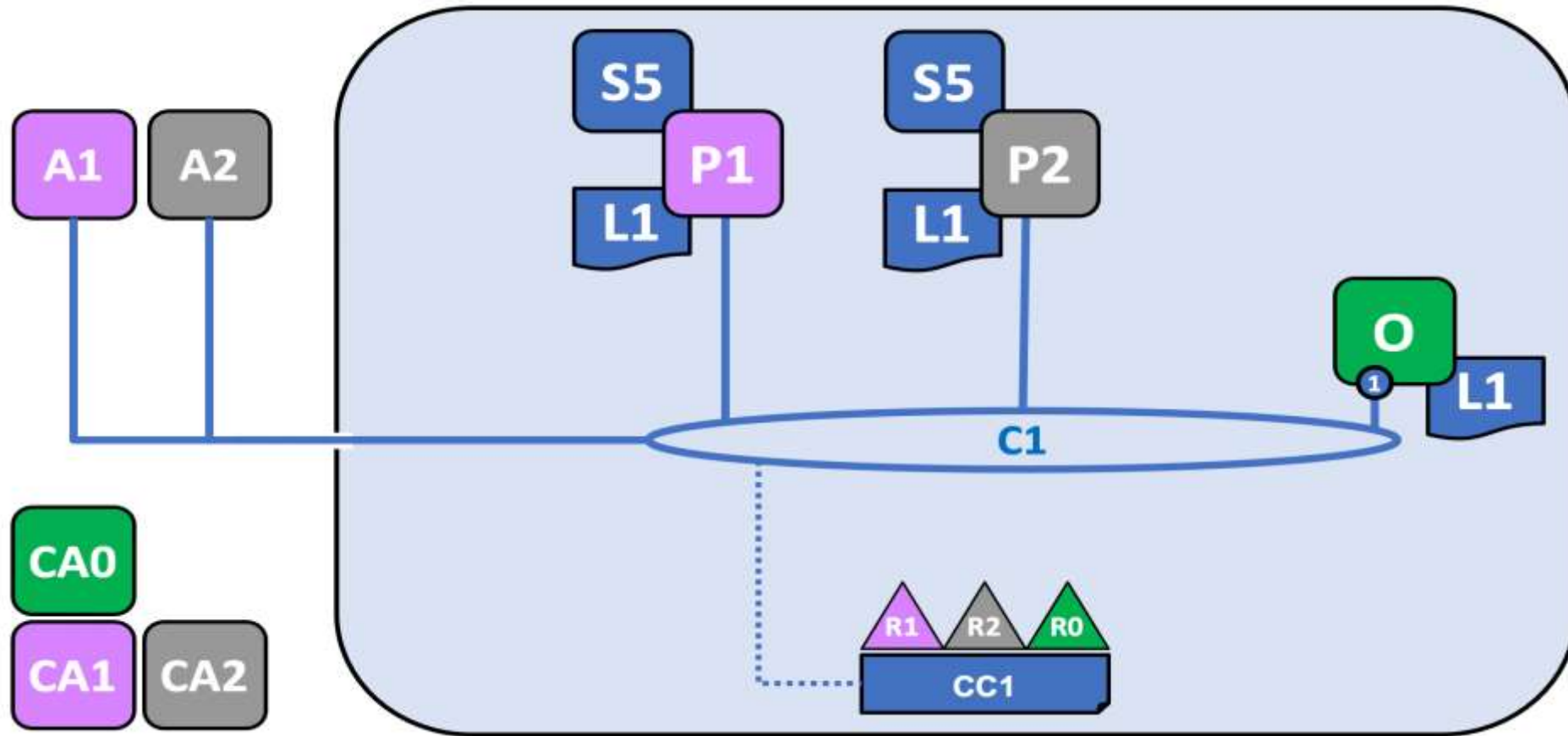
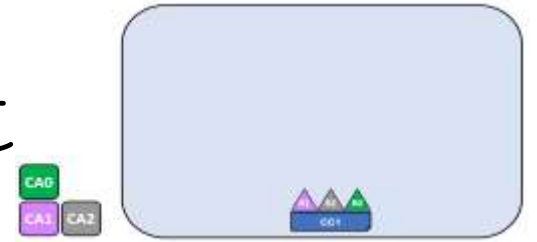


# Hyperledger Fabric Structure

# Design a Sample Network



# First Step- Create a Net



- The first step in creating a network or a channel is to agree to and then define its configuration:  
The channel configuration, CC1, has been agreed to by organizations R1, R2, and R0 and is contained in a block known as a "configuration block" that is, typically, created the **configtxgen** tool from a **configtx.yaml**
- This configuration block contains a record of the organizations that can join components and interact on the channel, as well as the **policies** that define the structure for how decisions are made and specific outcomes are reached.

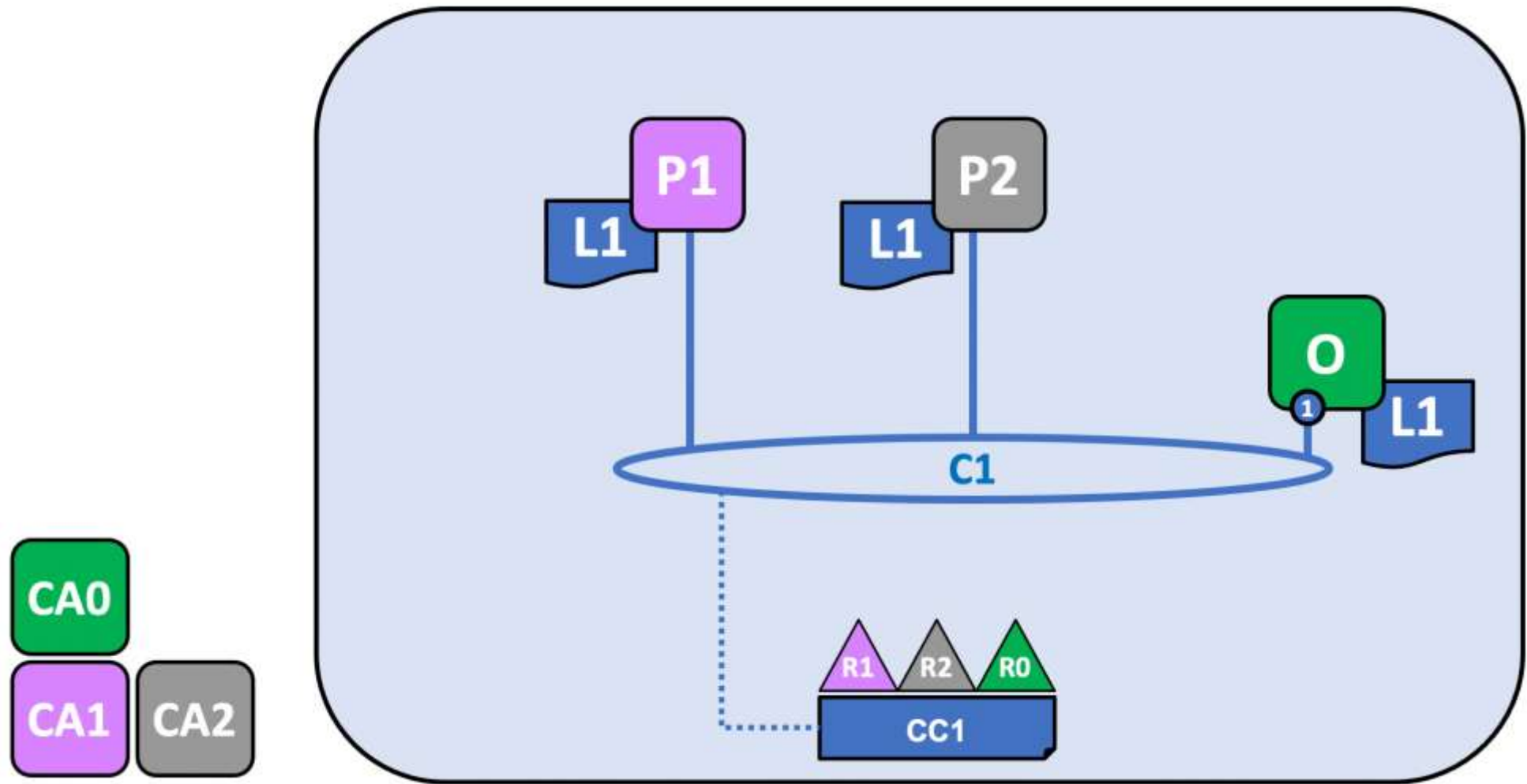
# Certificate Authorities

- Certificate Authorities play a key role in the network because they dispense X.509 certificates that can be used to identify components as belonging to an organization.
- Certificates issued by CAs can also be used to sign transactions to indicate that an organization endorses the transaction result – a precondition of it being accepted onto the ledger.
- Different organizations often use different CAs.

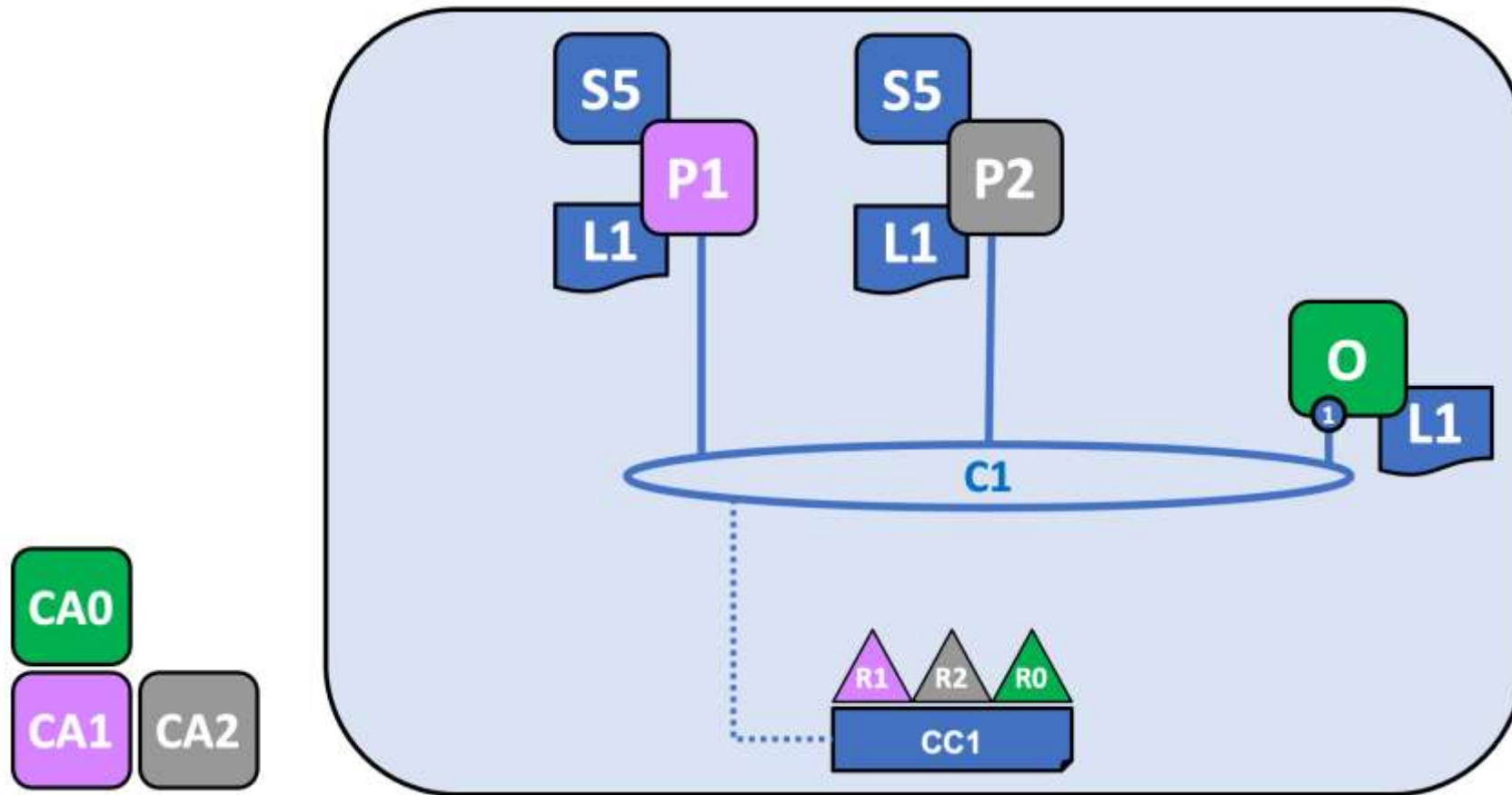
# Certificate Authorities

- The mapping of certificates to member organizations is achieved via a structure called a [Membership Services Provider \(MSP\)](#), which defines an organization by creating an MSP which is tied to a root CA certificate to identify that components and identities were created by the root CA.

# Join nodes to the channel



# Install, approve, and commit a chaincode



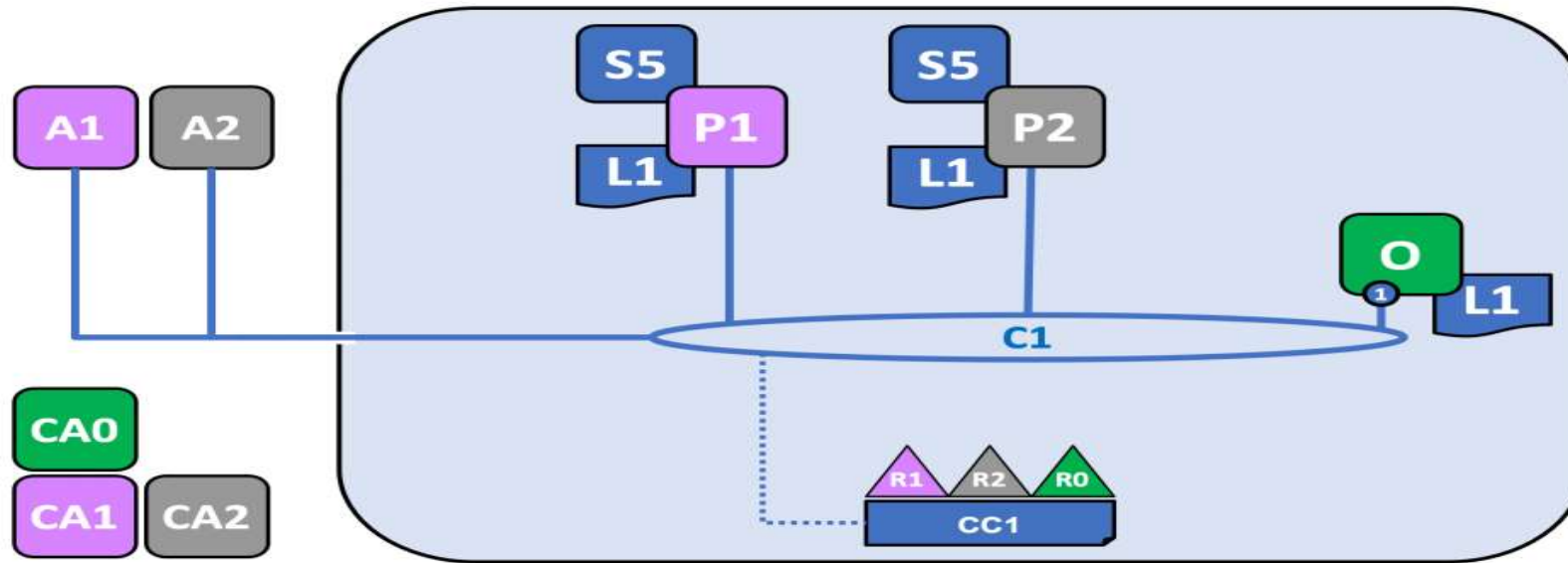
# Chaincode

- The most important piece of information supplied within the chaincode definition is the endorsement policy.
- It describes which organizations must endorse transactions before they will be accepted by other organizations onto their copy of the ledger.
- An endorsement policy can be set to any combination of members in a channel, depending on the use case. If an endorsement policy is not set, it is inherited from the default endorsement policy specified in the channel



# Using an application on the channel

- After a smart contract has been committed, client applications can be used to invoke transactions on a chaincode, via the Fabric Gateway service (the gateway).



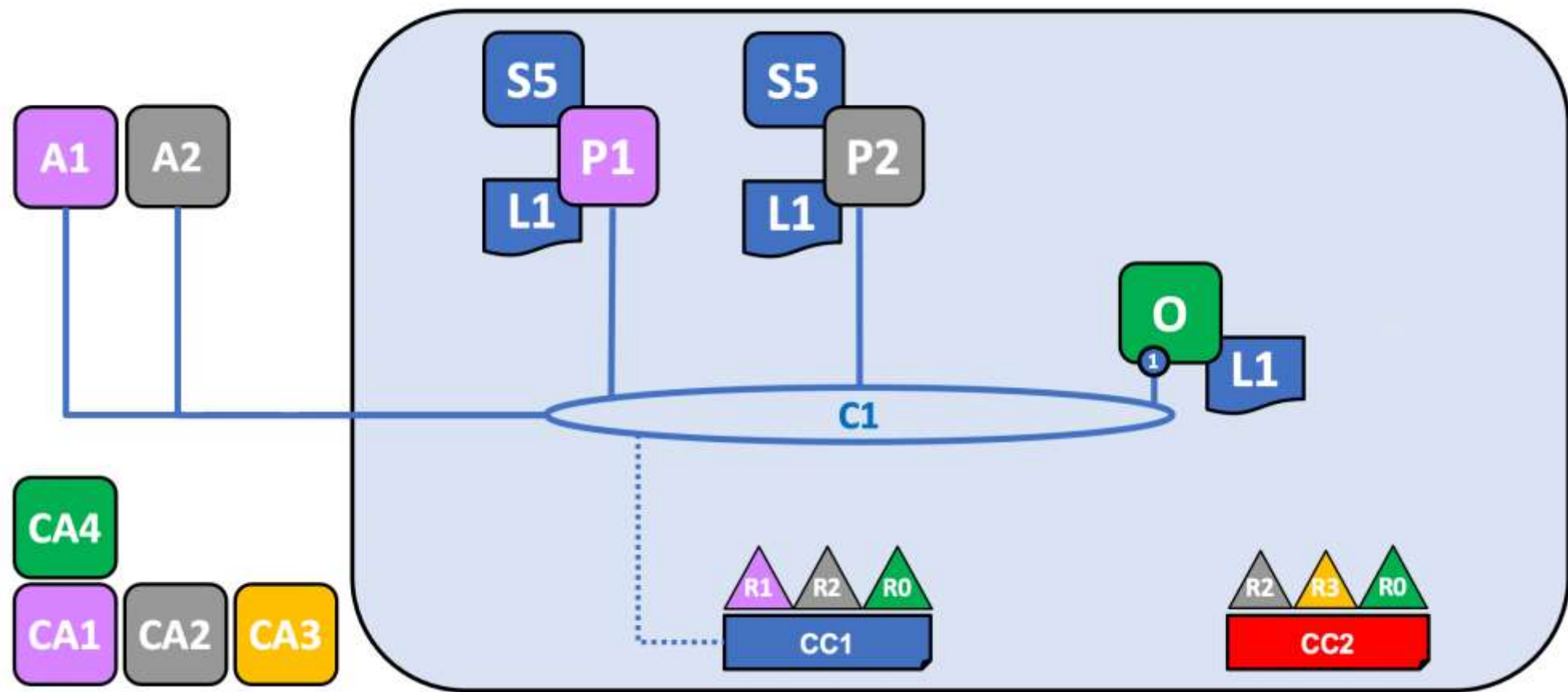
- Just like peers and orderers, a client application has an identity that associates it with an organization. In our example, client application A1 is associated with organization R1 and is connected to C1.
- Starting in Fabric v2.4, the client application (developed using a Gateway SDK v1.x) makes a gRPC connection to the gateway service, which then handles the transaction proposal and endorsement process on behalf of the application. The transaction proposal serves as input to the chaincode, which uses it to generate a transaction response.

# Joining components to multiple channels

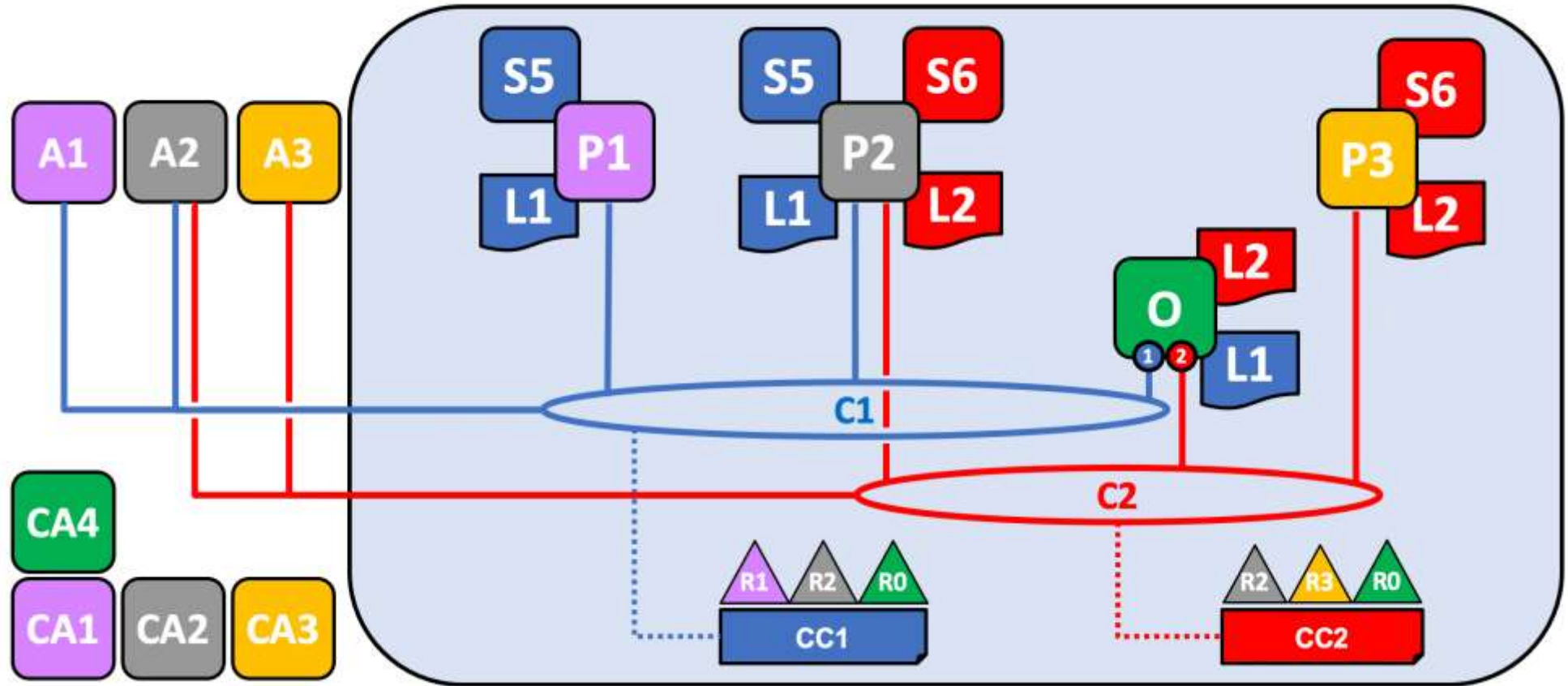
- **Creating the new channel configuration**

the first step in creating a channel is to create its configuration. This channel will include not just R2 and R0, but a new organization, R3, which has had its identities and certificates created for it by CA3.

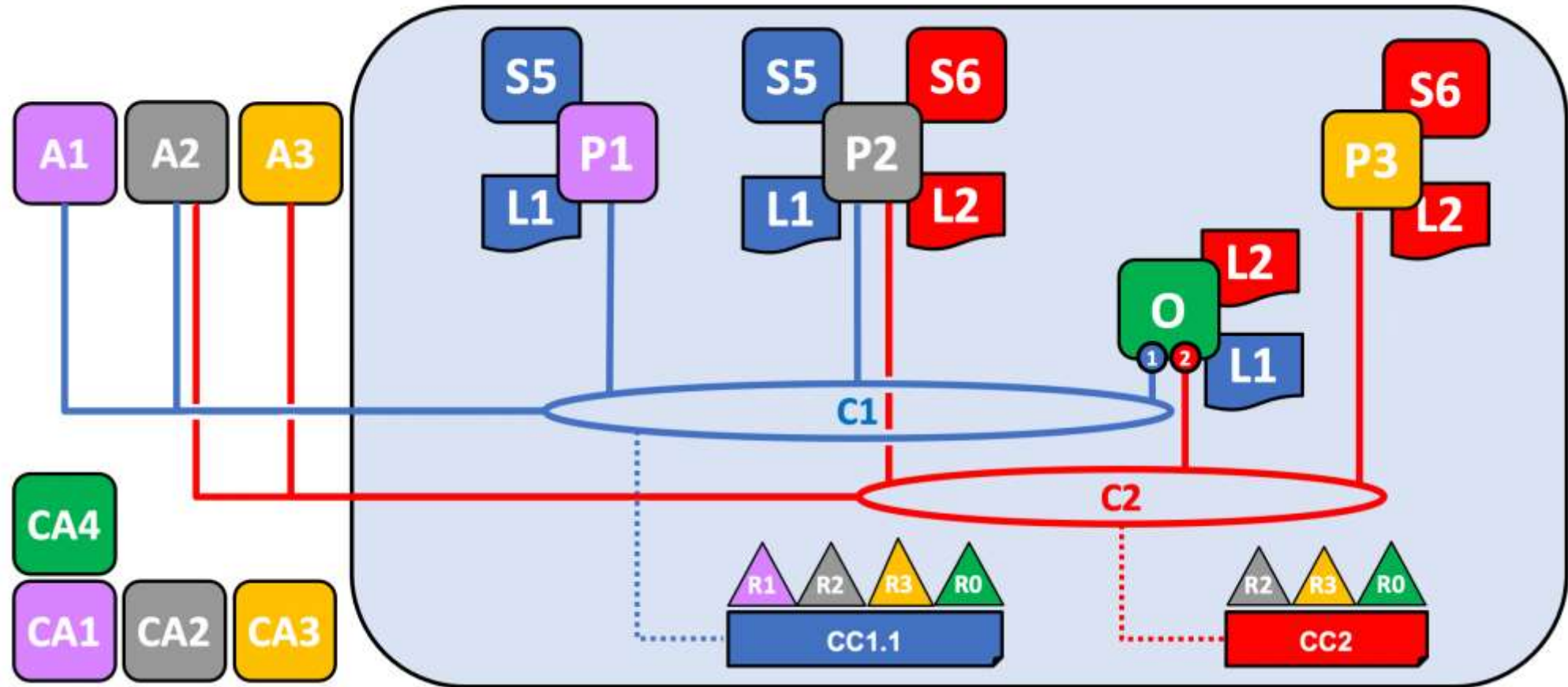
R1 will have no rights over this channel and will not be able to join components to it. In fact, it has no way to know it even exists!



# Join components to the new channel



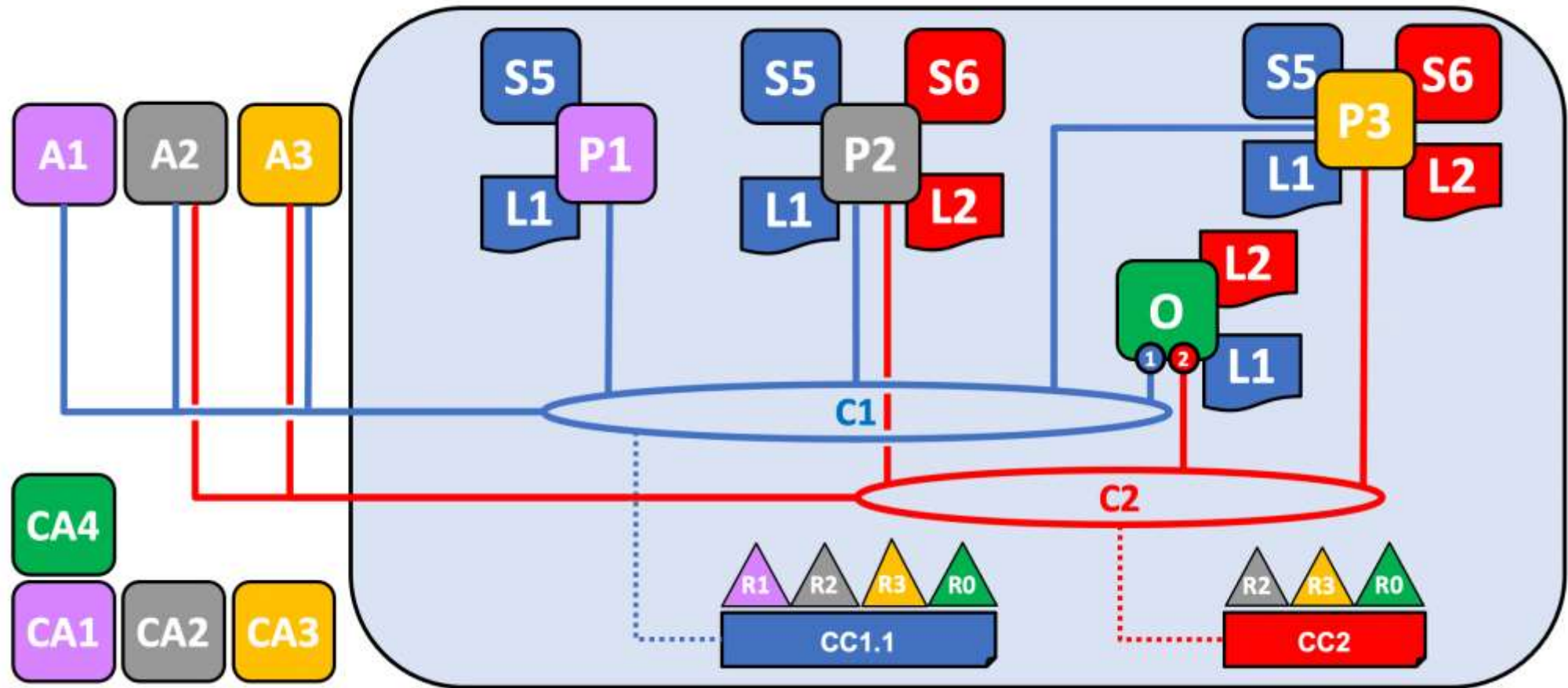
# Adding an organization to an existing channel



# Adding a new organization to a channel is, at a high level, a three step process:

- Decide on the new organization's permissions and role. The full scope of these rights must be agreed to before R3 is added to C1 and is beyond the scope of this topic, but comprise the same kinds of questions that must be answered when creating a channel in the first place. What kind of permissions and rights will R3 have on C1? Will it be an admin on the channel? Will its access to any channel resources be restricted (for example, R3 might only be able to write to C1, which means it can propose changes but not sign them)? What chaincodes will R3 install on its peers?
- Update the channel, including the relevant chaincodes, to reflect these decisions.
- The organization joins its peer nodes (and potentially ordering nodes) to the channel and begins participating.

# Adding existing components to the newly joined channel





- In this example, R3 adds P3, which was previously joined to C2, to C1. When it does this, P3 pulls C1's ledger, L1.
- As we mentioned in the previous section, R3 has been added to C1 with equivalent rights as R1 and R2.
- Similarly, because the chaincode S5 was redefined and reapproved on the channel to include R3, R3 can now install S5 and begin transacting.
- Just as R2 modified its application A2 to be able to be used with channel C2, A3 is also now able to invoke transactions on C1.