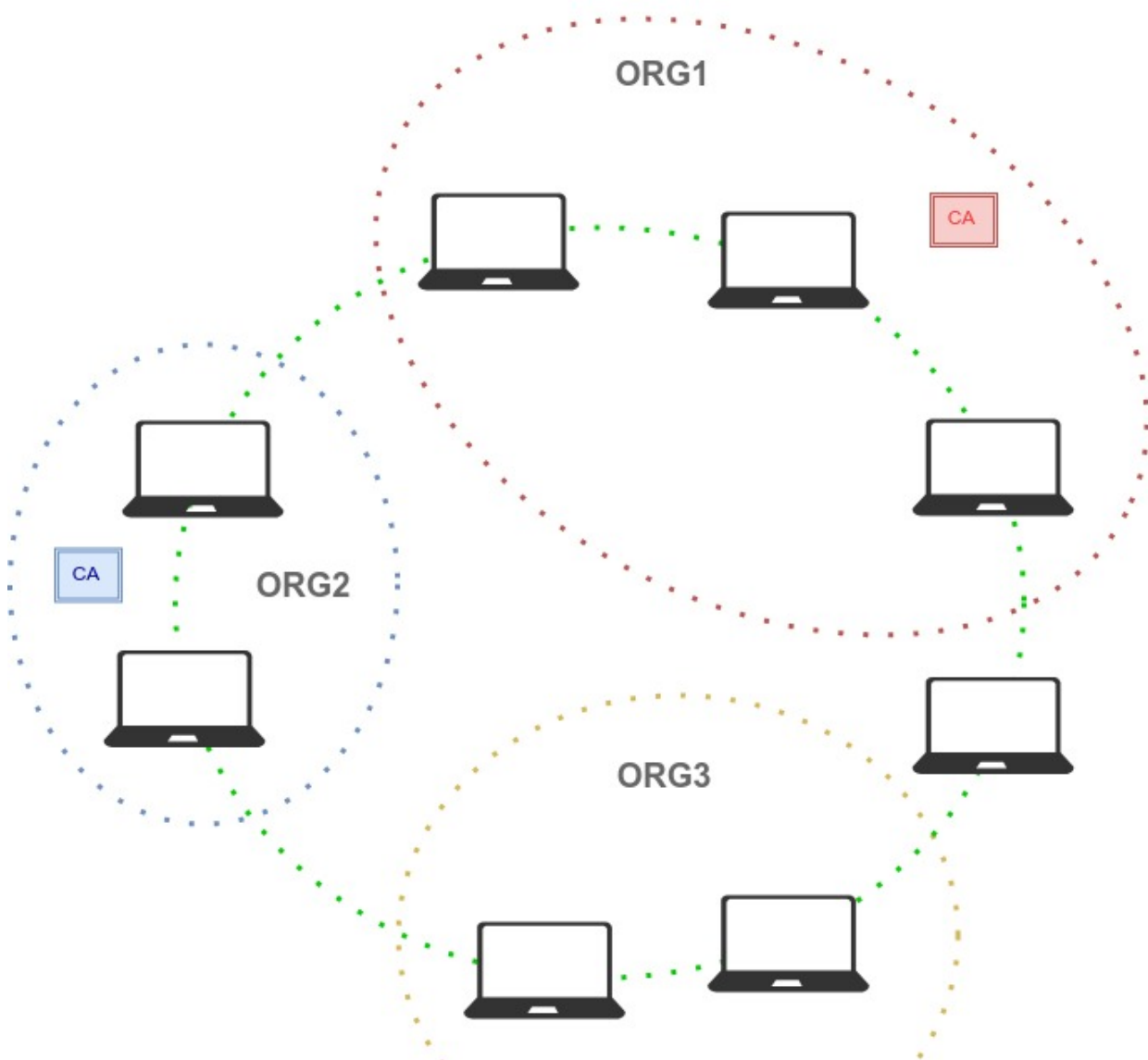


# Network Concepts

In this lesson, we will give a quick introduction to Hyperledger Fabric.

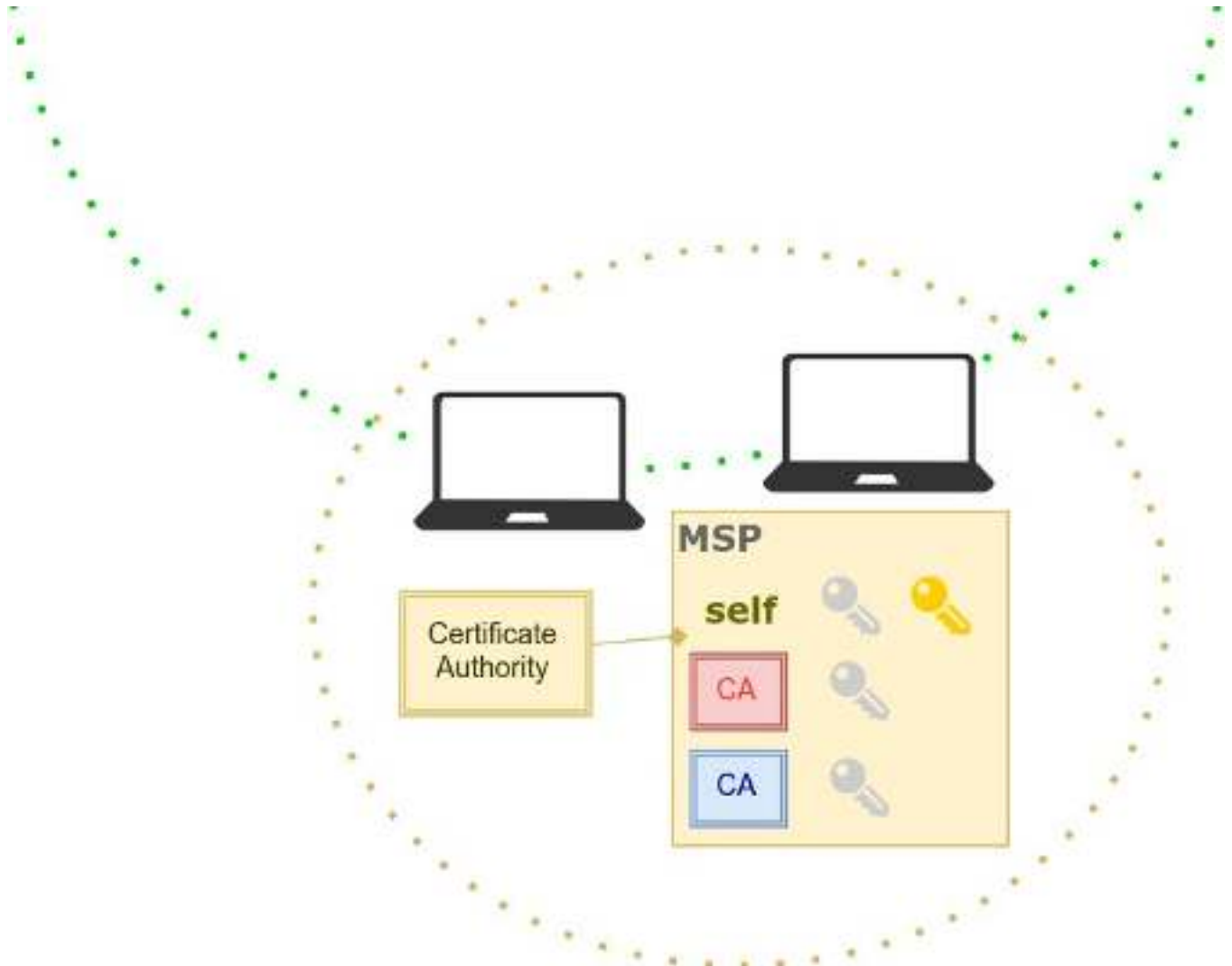
## WE'LL COVER THE FOLLOWING ^

- Channels
  - Ledger
  - Chaincode
  - Ordering Service



CA

Multiple organizations can form a network. Each organization can have its own membership services provider which will issue and revoke identities for users and peers of that organization. This membership service provider can be a certificate authority hierarchy or a single root certificate authority. In order for other organizations to validate transactions by each other, they need to be set up by having trusted root certificate authorities pre-configured on them in something called an MSP (Membership Service Provider).

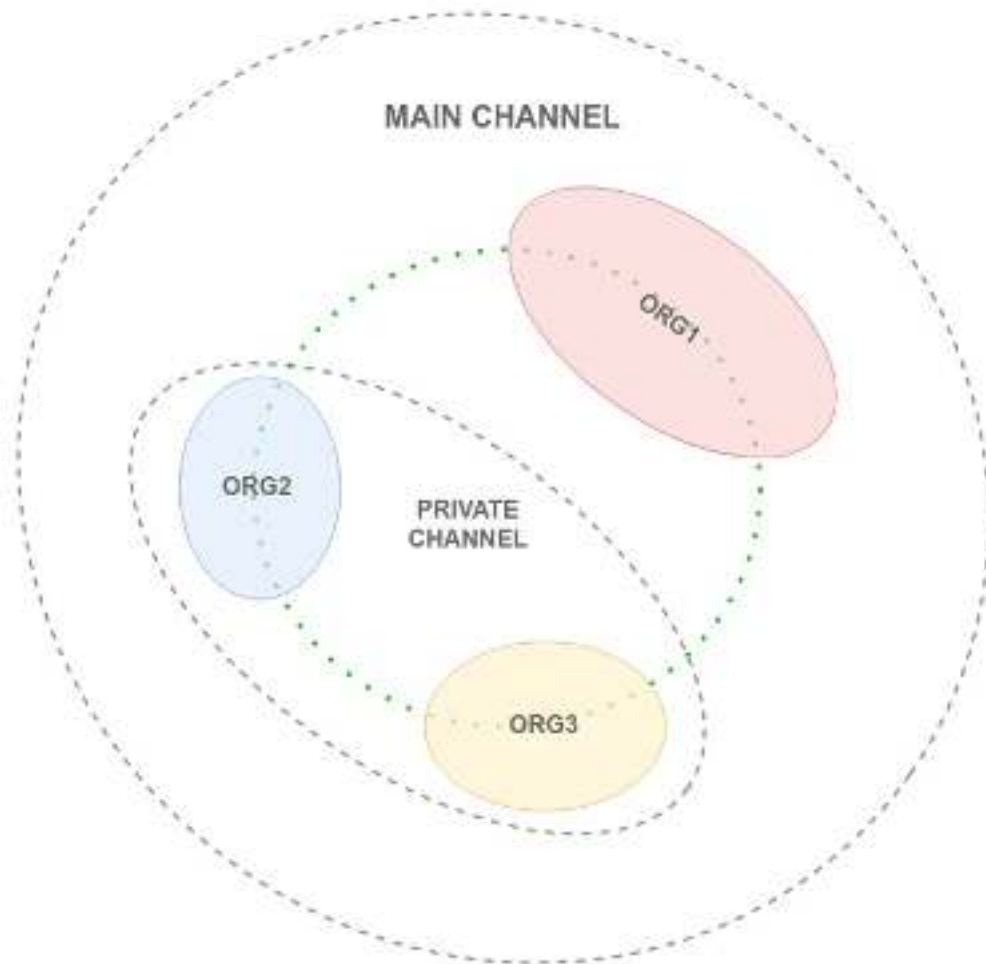


## Channels #

In fabric there is a concept of channels. Each peer can join one or more channels. If there are some special transactions that need to be shared only by org1 and org2 and they don't want any peer from org3 to read it, both the organizations can join a separate channel and be present on the main channel as well.

This provides privacy that does not exist in public blockchains.

Internally a separate ledger is maintained on each peer for each channel that it is on.



## Ledger #

Each peer maintains a ledger. A ledger is essentially two things:

- Timestamped transactions organized in blocks and stored.
- A State Database which is the computed outcome of all transactions executed in order.

By having a state database it is easier to query the ledger as the current state is pre-calculated and stored on each peer.

The **state database** is essentially a key value based datastore(an instance of Apache CouchDB). The state database contains the final state of the ledger after applying all transactions recorded in blockchain. It is therefore like a cache. It helps make querying the blockchain faster, as the state is pre-calculated. It also gives developers an easier programming model to work with. Developers writing chaincode do not need to write transactions directly on blockchain, but rather, write to state in state database (the required transactions are generated under the hood).

The peers communicate with each other to ensure their final state is the same at all times using the *gossip protocol*.

## Chaincode #

This is a piece of code that is part of the ledger. The chaincode provides logic on what, how, when, and by whom things can be written on ledger. It is essentially business rules coded to store data onto the ledger. For example the chaincode can make sure an account holder has enough balance before he transfers an amount to another account.

## Ordering Service #

Ordering Service is a distributed(solo in dev mode) service that is responsible for organizing endorsed transactions into sequenced blocks and distributing to all peers. This service is run on multiple nodes. Typically each org will have at least one node of ordering service to ensure they are a part of the end to end transaction processing.

In the next lesson, we will take a look at Transaction processing.