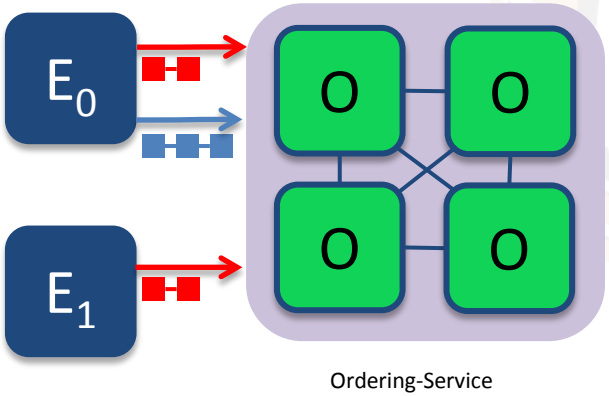
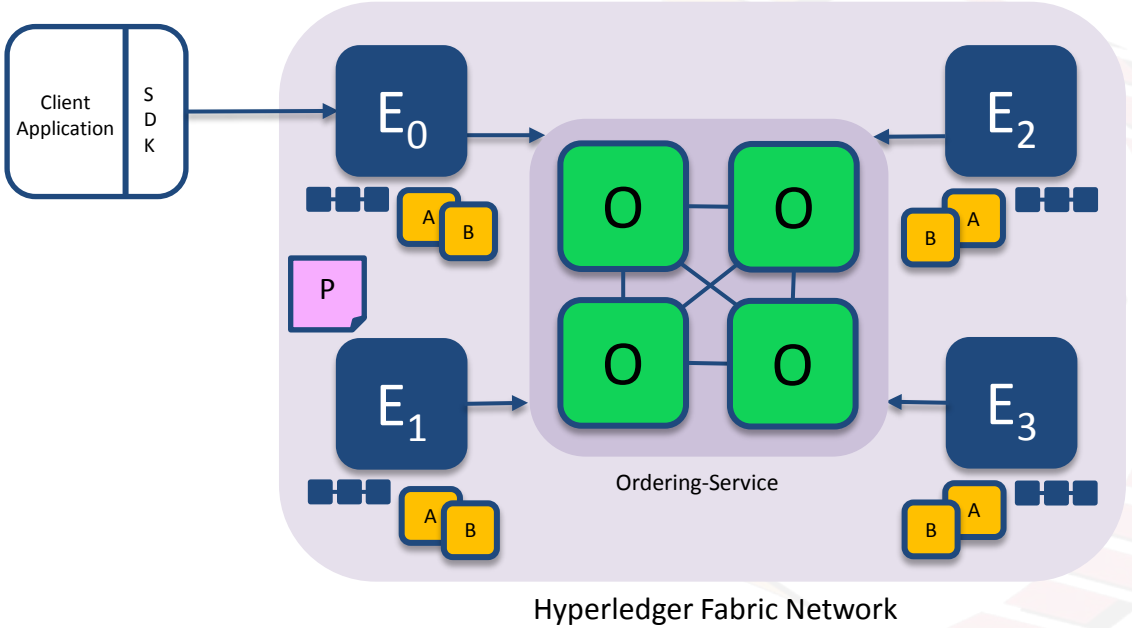
Channels

Channels provide privacy between different ledgers

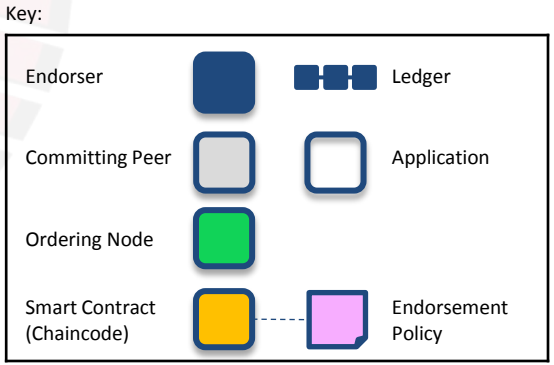


- Ledgers exist in the scope of a channel
  - Channels can be shared across an entire network of peers
  - Channels can be permissioned for a specific set of participants
- Chaincode is installed on peers to access the worldstate
- Chaincode is instantiated on specific channel
- Peers can participate in multiple channels
- Concurrent execution for performance and scalability

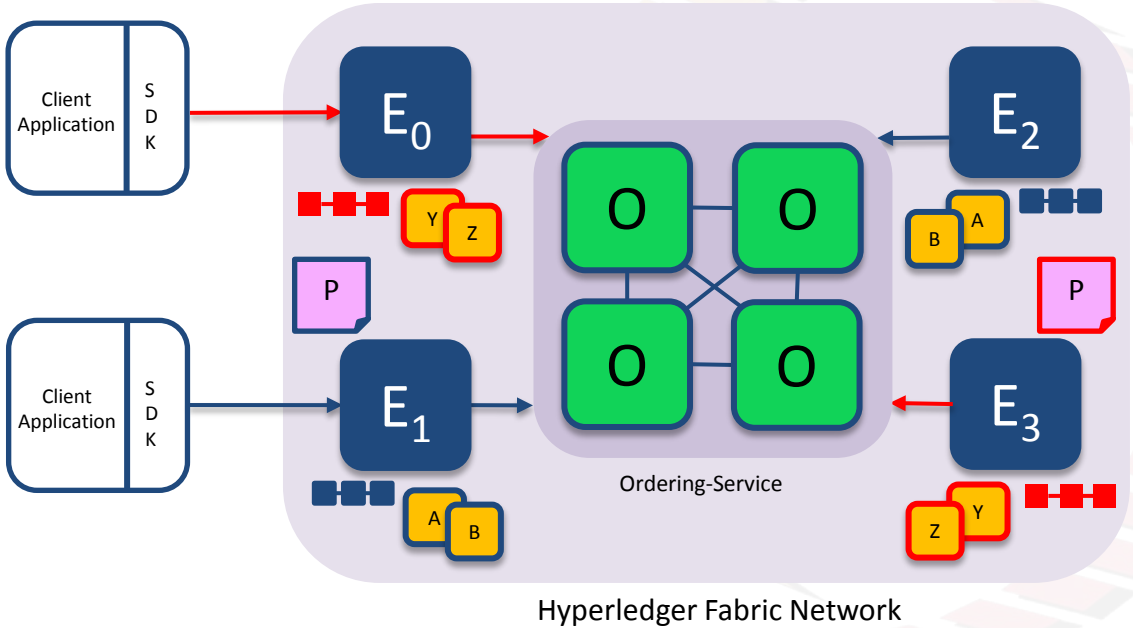
Single Channel Network



- All peers connect to the same system channel (blue).
- All peers have the same chaincode and maintain the same ledger
- Endorsement by peers E_0, E_1, E_2 and E_3

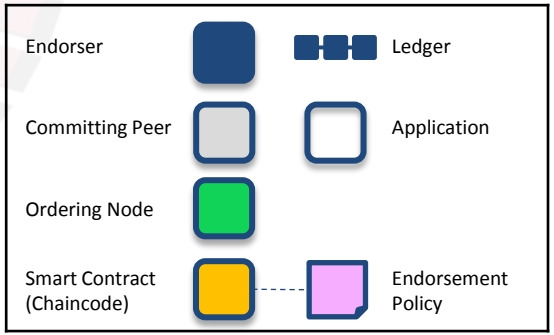


Multi-Channel Network



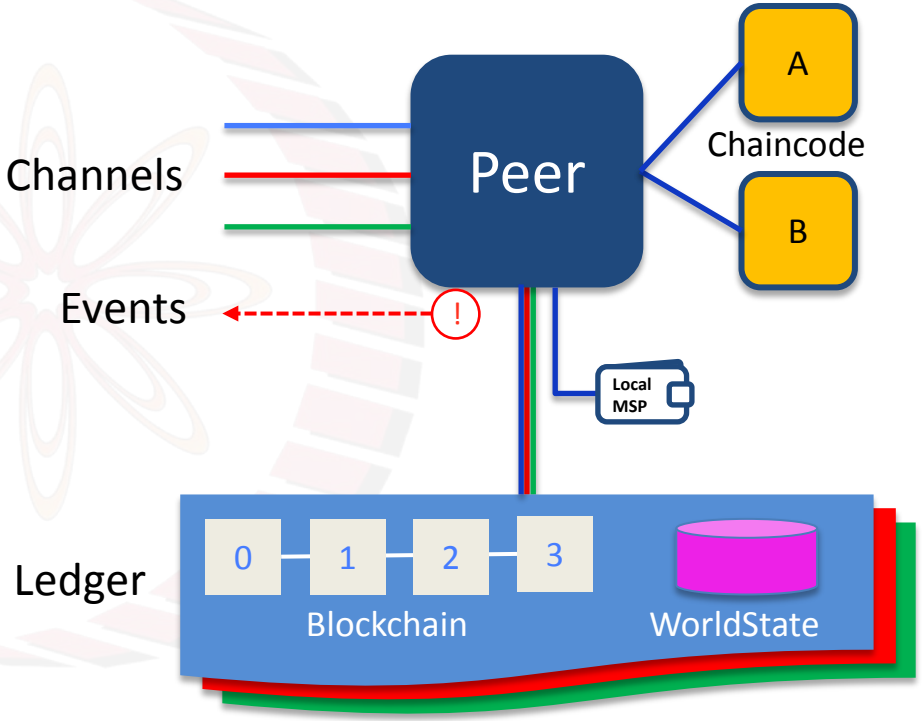
- Peers E_0 and E_3 connect to the red channel for chaincodes Y and Z
- Peers E_1 and E_2 connect to the blue channel for chaincodes A and B

Key:



Fabric Peer

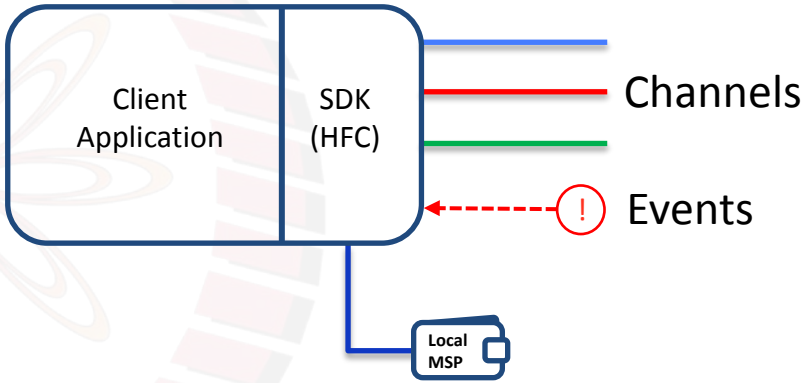
- Each peer:
  - Connects to one or more channels
  - Maintains one or more ledgers for each channel
  - Chaincodes are instantiated in separate docker containers
  - Chaincodes are shared across channels (no state is stored in chaincode container)
  - Local MSP (Membership Services Provider) provides crypto material
  - Emits events to the client application



Client Application

- Each client application uses Fabric SDK to:
  - Connects over channels to one or more peers
  - Connects over channels to one or more orderer nodes
  - Receives events from peers
  - Local MSP provides client crypto material

  - Client can be written in different languages (Node.js, Go, Java, Python?)



Fabric Certificate Authority

- Default (optional) Certificate Authority within Fabric network for issuing Ecerts (long-term identity)
- Supports clustering for HA characteristics
- Supports LDAP for user authentication
- Supports HSM for security
- Can be configured as an intermediate CA