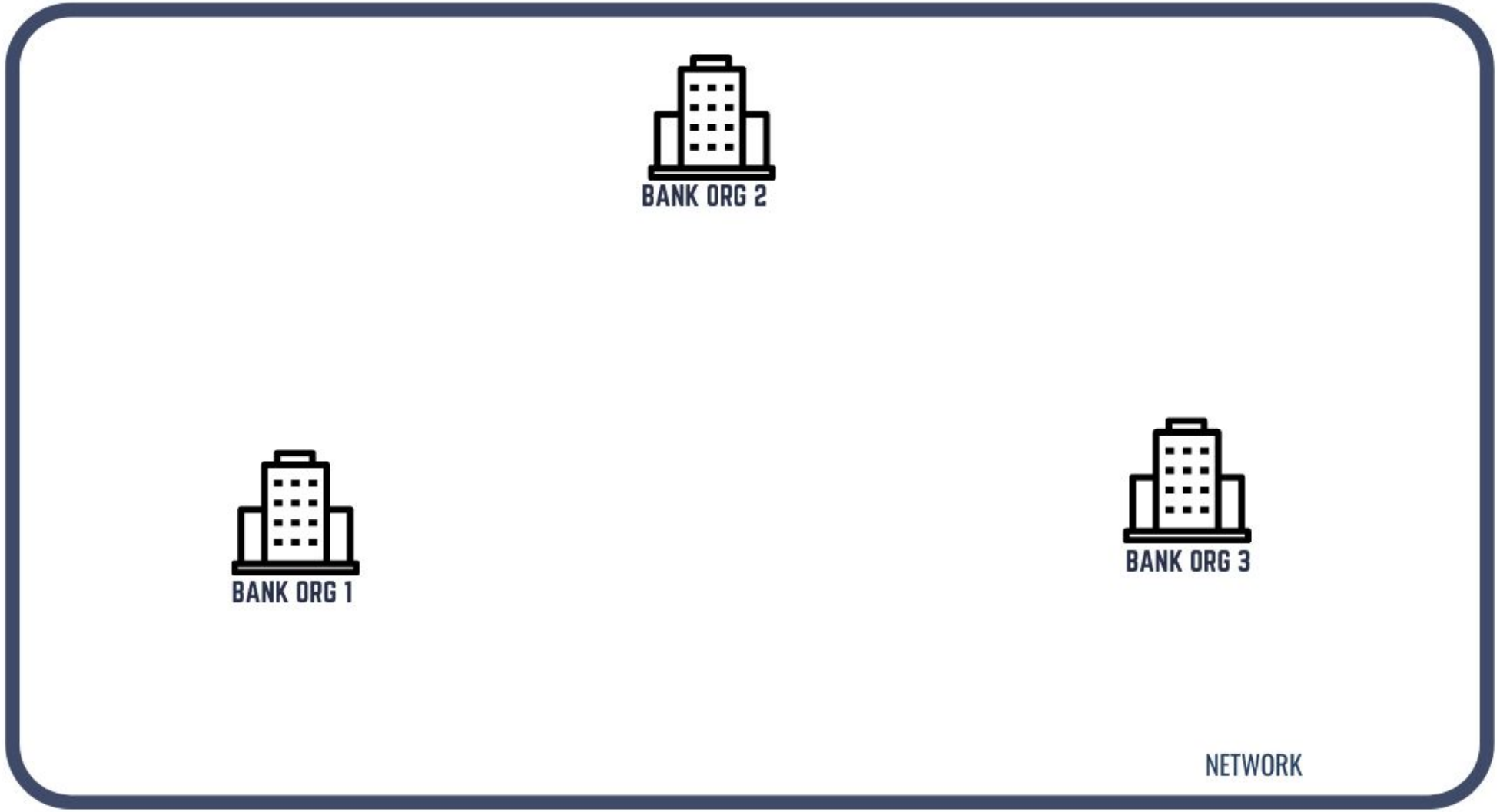
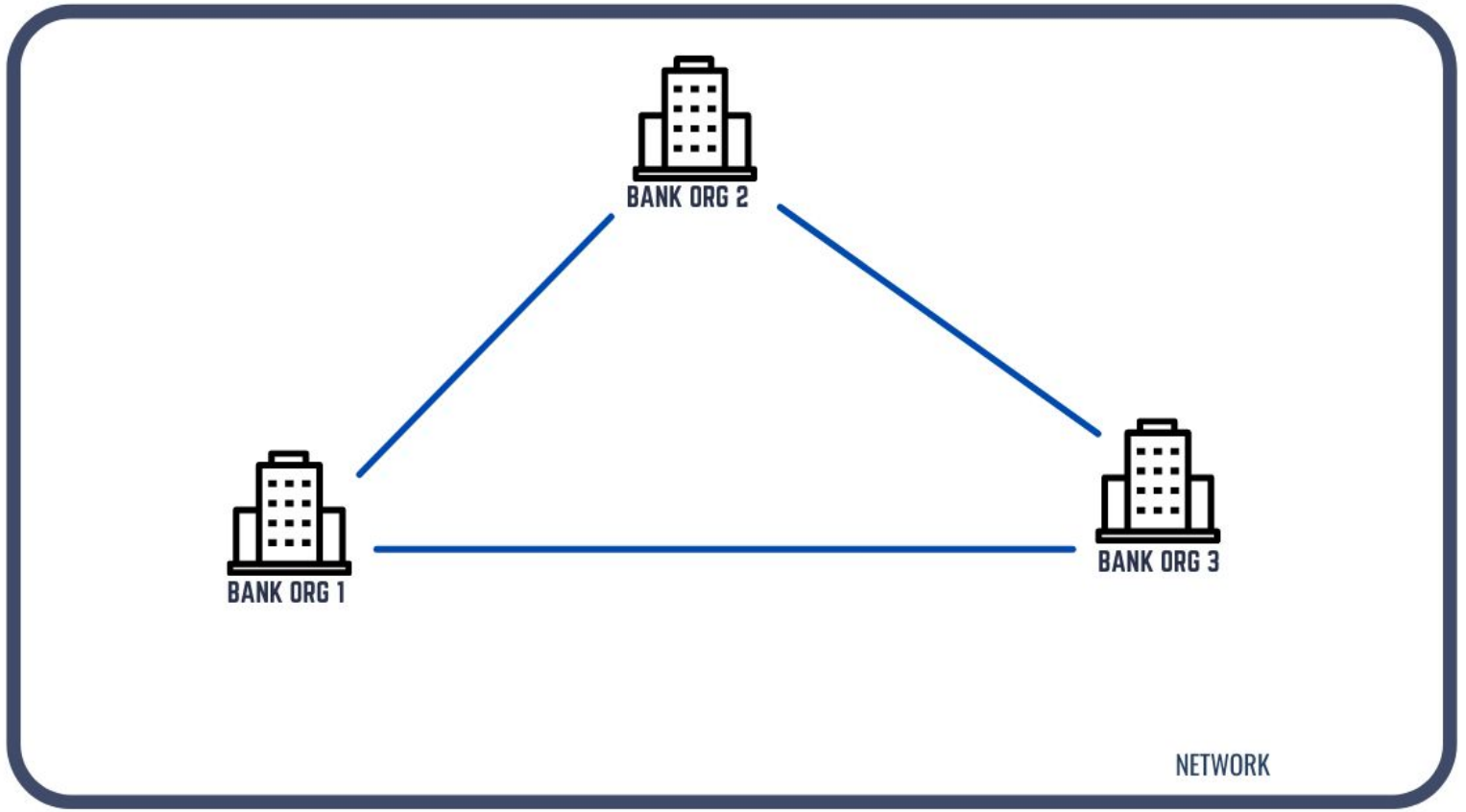
Private Data Collections

Bank Consortium



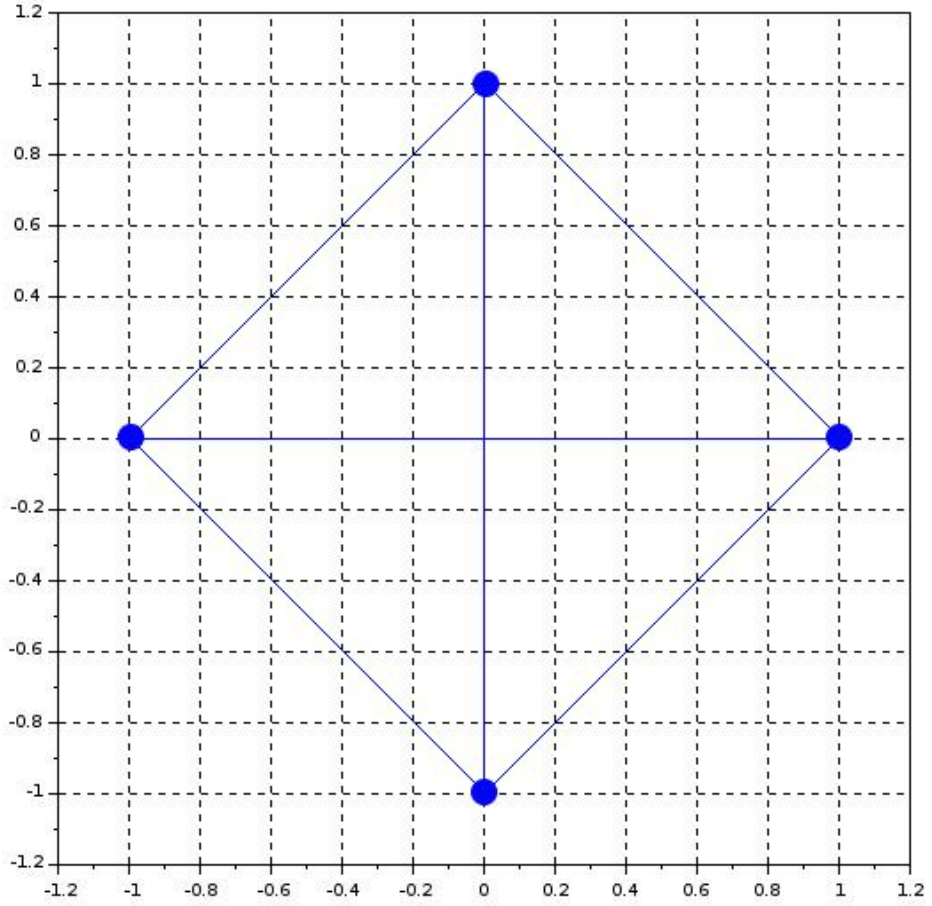
3 Organisations

- 3 Organisations
- n(n-1)/2
- 3(3-1)/2 = 3
- 3 Channels



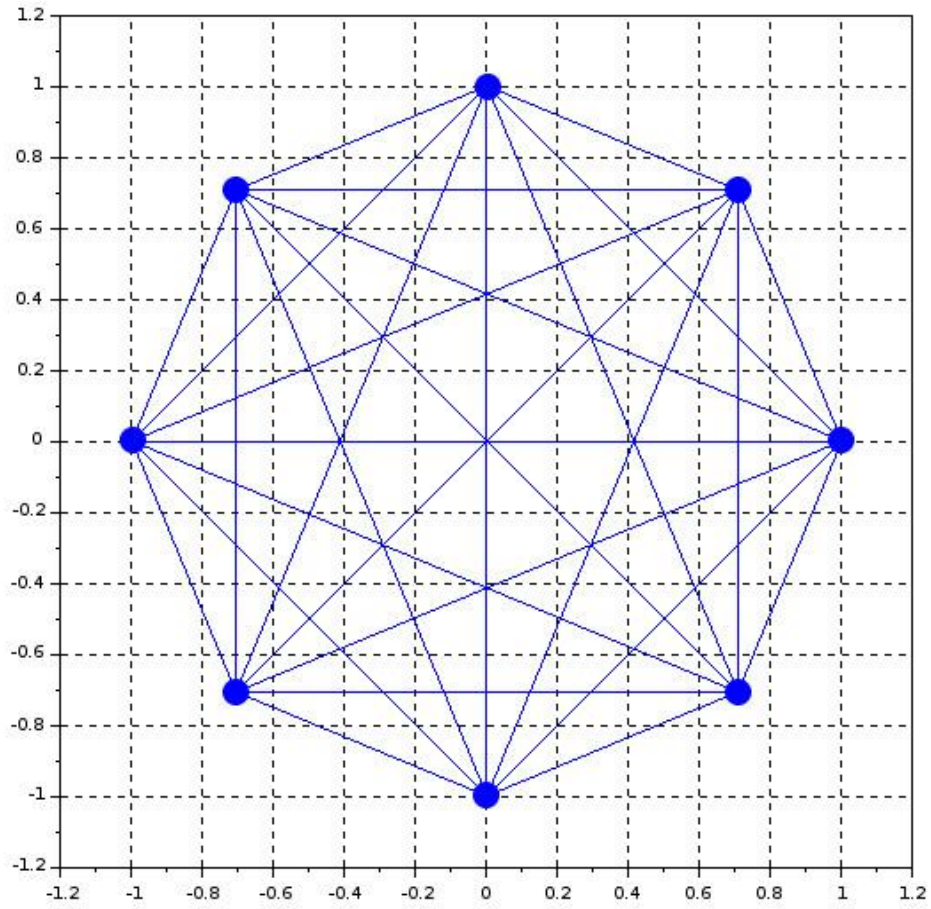
4 Organisations

- 4 Organisations
- n(n-1)/2
- 4(4-1)/2 = 6
- 6 Channels



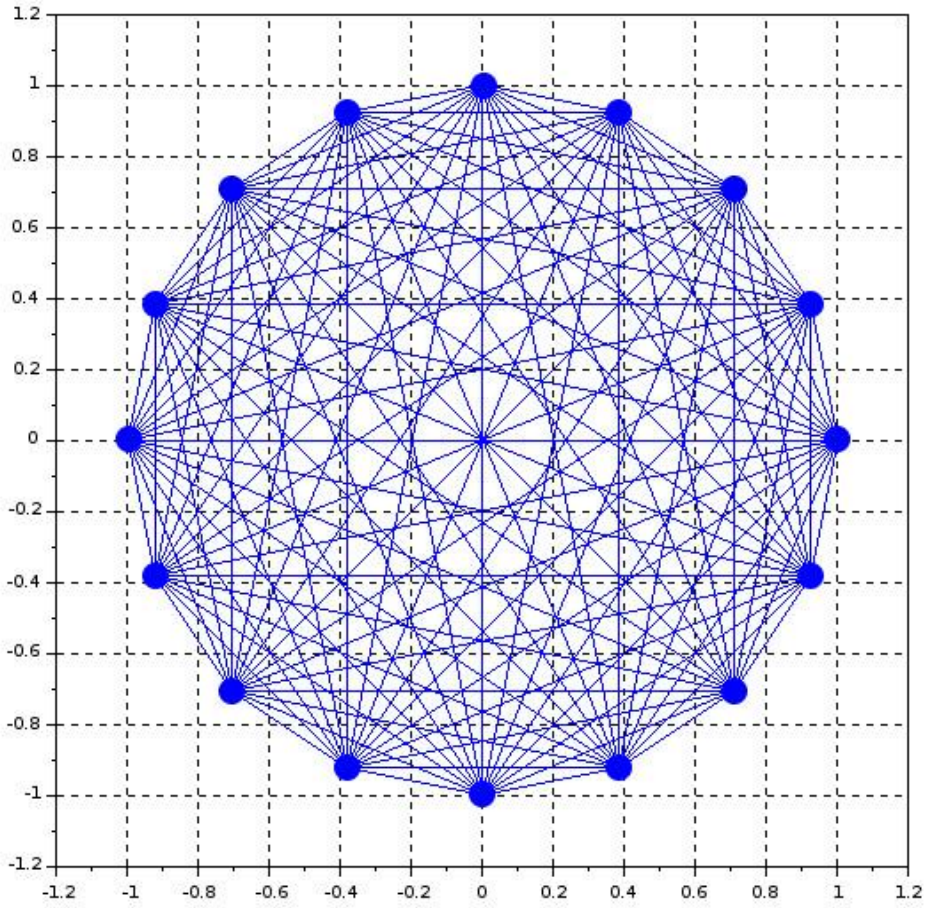
8 Organisations

- 8 Organisations
- n(n-1)/2
- 8(8-1)/2 = 28
- 28 Channels



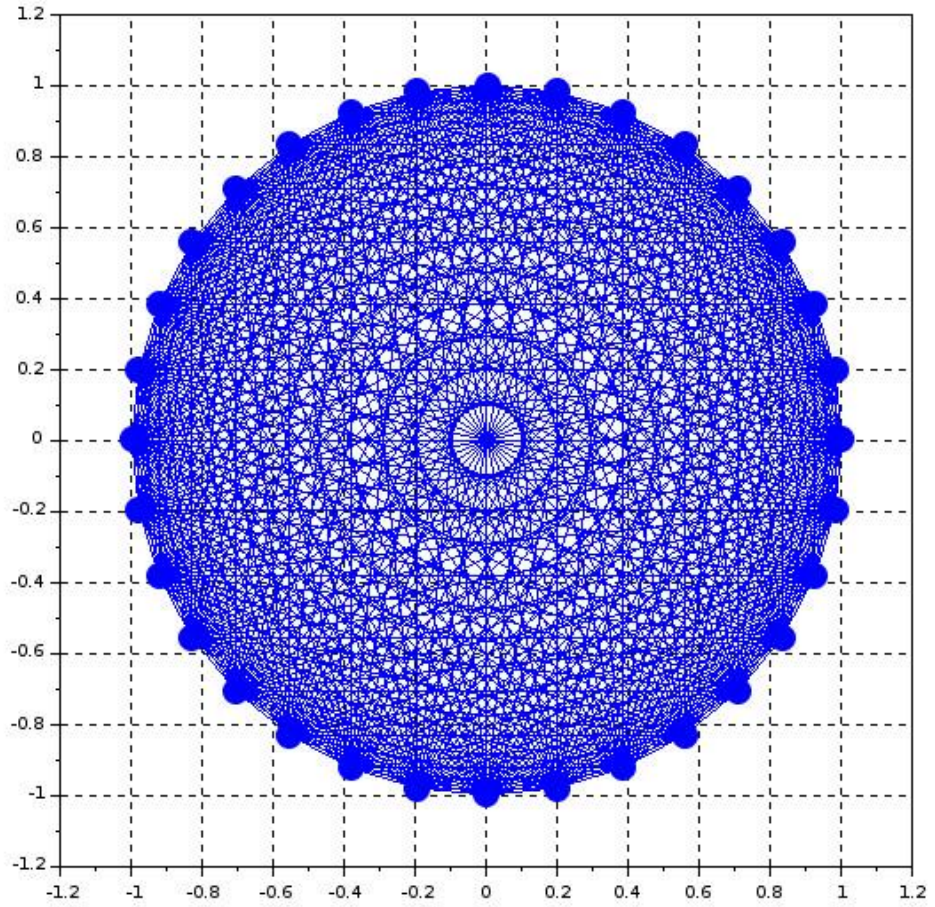
16 Organisations

- 16 Organisations
- n(n-1)/2
- 16(16-1)/2 = 120
- 120 Channels



32 Organisations

- 32 Organisations
- n(n-1)/2
- 32(32-1)/2 = 496
- 496 Channels



Is adding channels a good idea ?

- Usage of more resources
- Difficulty to manage channels (Administrative Overhead)
- Maintaining chaincode versions, policies, ledgers, etc.

Private Data Collections

- A group of organizations on **a channel can keep data private** from other organizations on that channel

- **Private Data Collections** allow a defined subset of organizations on a channel to **endorse, commit, or query private data** without having to create a separate channel

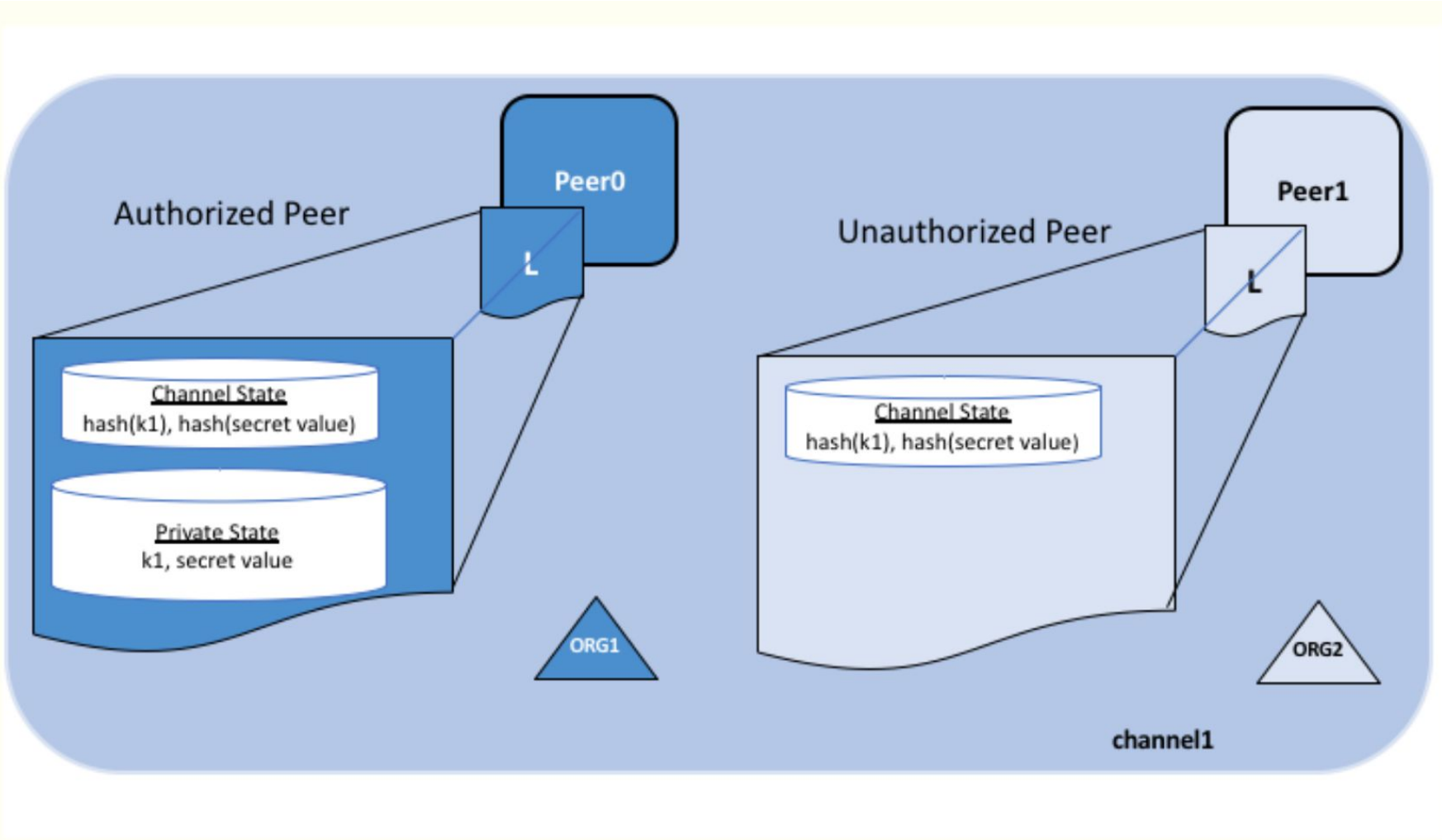
Private Data Collection

A collection is the combination of two elements:

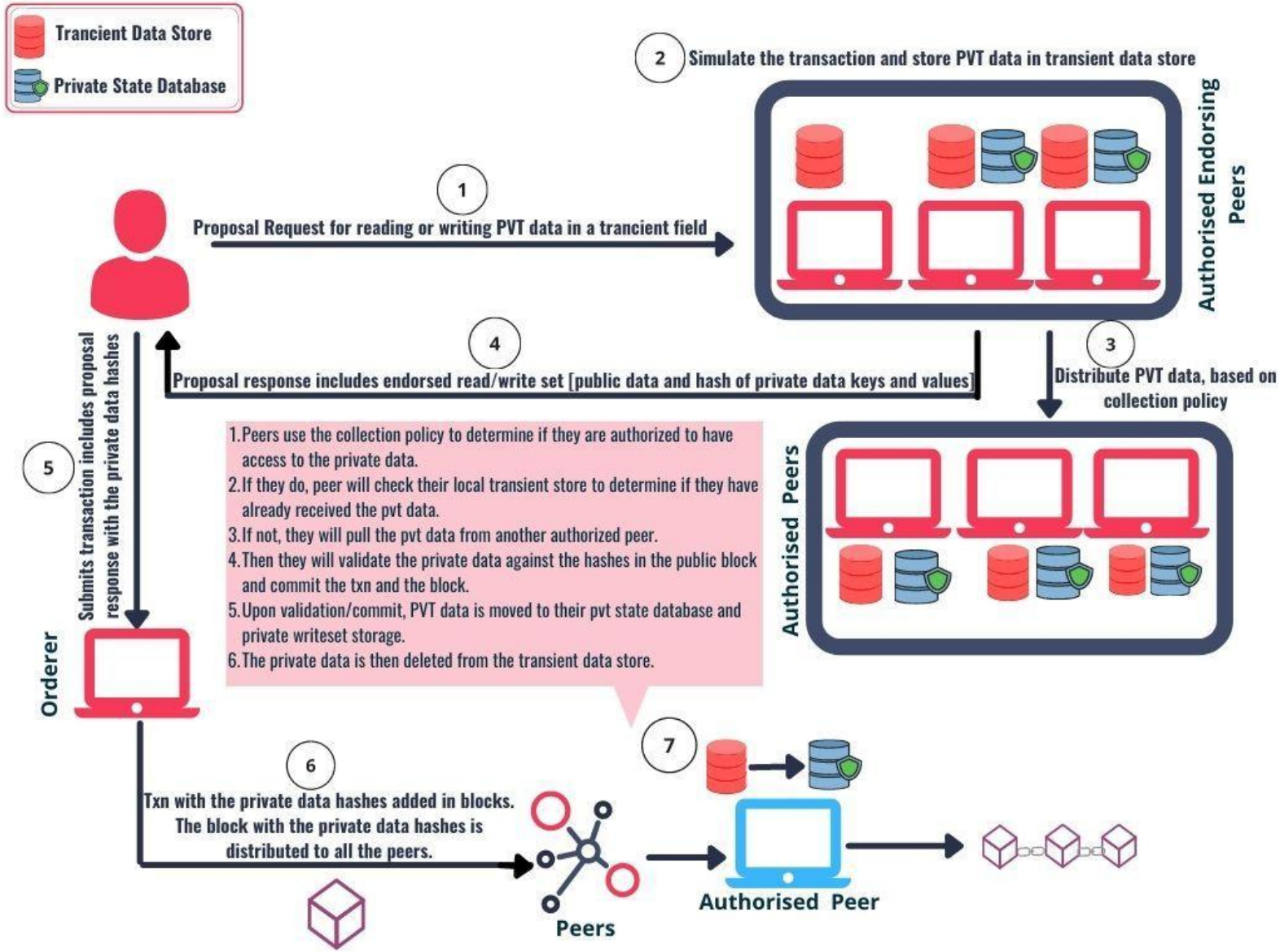
- **The actual private data**
  - stored in a private state database on the peers of authorized organizations
  - can be accessed from chaincode on these authorized peers
  - ordering service is not involved here and does not see the private data.

- **A hash of that data**,
  - which is endorsed, ordered
  - written to the ledgers of every peer on the channel
  - the hash serves as evidence of the transaction and is used for state validation and can be used for audit purposes.

Private State



Transaction Flow - PDC



At the time of Block Commit

- **Authorized peers** use the collection policy to determine if they are authorized to have access to the private data.
- If they do, they will first **check their local transient data store** to determine if they have already received the private data at chaincode endorsement time.
- If not, they will attempt to **pull the private data** from another **authorized peer**.
- Then they will **validate the private data** against the **hashes** in the public block and commit the transaction and the block.
- Upon validation/commit, the private data is moved to their copy of **the private state database**.
- The private data is then **deleted from the transient data store**.

Purging Private Data

For very sensitive data, even the parties sharing the private data might want — or might be required by government regulations — **to periodically "purge" the data on their peers**, leaving behind a hash of the data on the blockchain to serve as immutable evidence of the private data.

Purged private data cannot be queried from chaincode, and is not available to other requesting peers.

PDC Definition

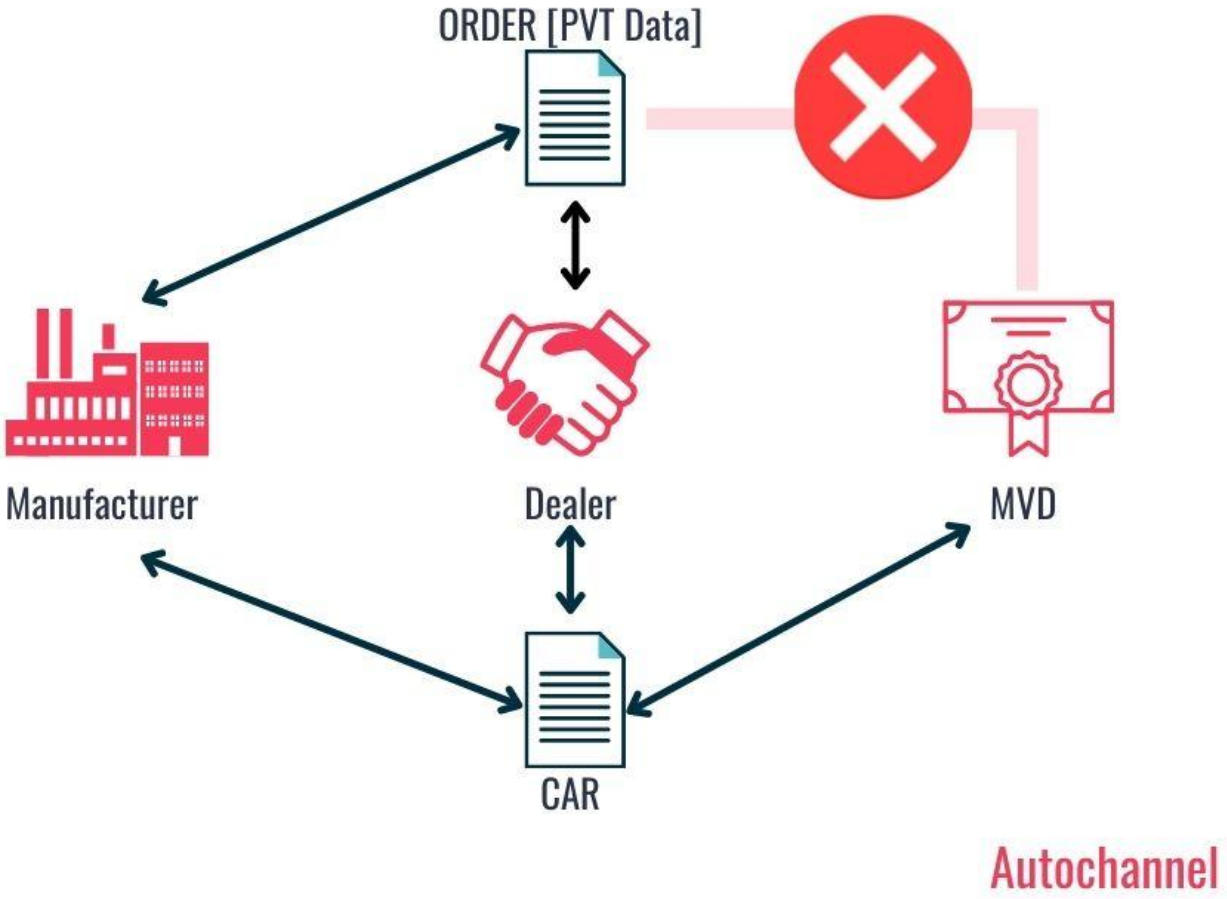
The collection definition gets deployed to the channel at the time of chaincode instantiation

- **name**: Name of the collection.
- **policy**: The PDC distribution policy defines which organizations' peers are allowed to persist the collection data expressed
- **requiredPeerCount**: Minimum number of peers that endorsing peer must successfully disseminate private data to before the peer signs the endorsement and returns the proposal response back to the client.
- **maxPeerCount**: Maximum number of other peers that each endorsing peer will attempt to distribute the private data to.
- **blockToLive**: Represents how long the data should live on the private database in terms of blocks.
- **memberOnlyRead**: enforce that only clients belonging to one of the collection member organizations are allowed read access to private data

Collection Definition

```
[
  {
    "name": "CollectionOrder",
    "policy": "OR('manufacturer-auto-com.member', 'dealer-auto-com.member')",
    "requiredPeerCount": 1,
    "maxPeerCount": 1,
    "blockToLive": 1000000,
    "memberOnlyRead": true
  }
]
```

PDC in Automobile App



Functions

Some of the commonly used functions are:

- **ctx.GetStub().GetTransient()**
- **ctx.GetStub().GetPrivateDataHash(collectionName, orderID)**
- **ctx.GetStub().GetPrivateData(collectionName, orderID)**
- **ctx.GetStub().PutPrivateData(collectionName, orderID, bytes)**
- **ctx.GetStub().DelPrivateData(collectionName, orderID)**

THANK YOU