Original Structure

Chain ModPolicy = OrdererAdministration	Orderer ModPolicy = OrdererAdministration	Peer ModPolicy = PeerAdministration	MSP ModPolicy = implicit on self	Policy
OrdererAddrs = []	IngressPolicyNames = [OrdererWriters, PeerWriters]	AnchorPeers = []	oOrg1	OrdererReaders ModPolicy=OrdererAdministration
HashingAlgorithm	EgressPolicyNames = [OrdererReaders, OrdererWriters]		oOrg2	OrdererWriters ModPolicy=OrdererAdministration
BlockDataStructure	ChainCreationPolicyNames = []		pOrg1	PeerReaders ModPolicy=PeerAdministration
	BatchSize		pOrg2	PeerWriters ModPolicy=PeerAdministration
	BatchTimeout		pOrg3	OrdererAdministration ModPolicy=OrdererAdministration
	...			PeerAdministration ModPolicy=PeerAdministration
Value is marshaled common.<Key>	Value is marshaled orderer.<Key>	Value is marshaled peer.<Key>	Value is marshaled msp.MSPConfig and key is MSP ID	Value is marshaled common.Policy, and key is arbitrary

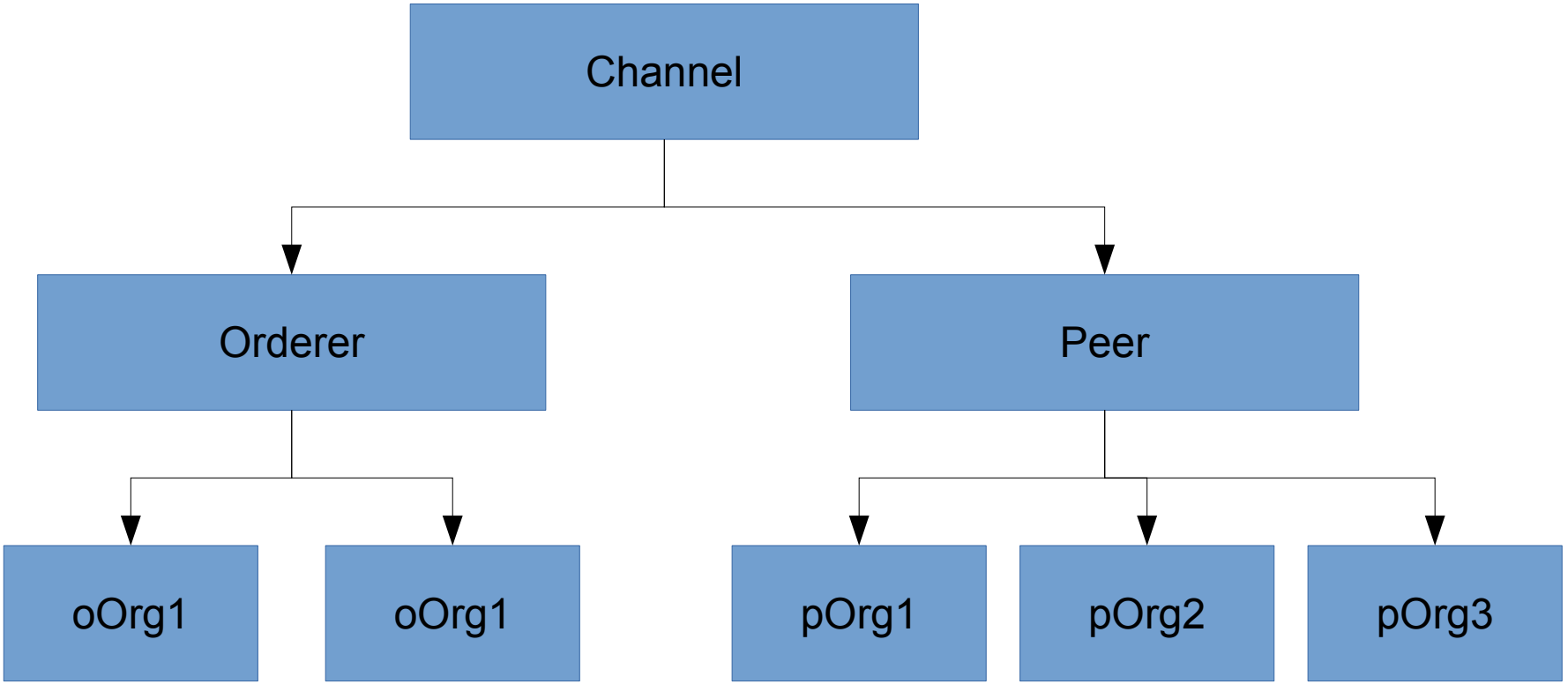
Benefits

- Conceptually very simple.
  - Set of items, each with a modification policy
  - Policies are global
- Updates are discouraged
- Works best in a network where the orderers and peers are the same organizations.
- Already implemented
  - Not validated/tested end to end

Problems

- The <type>.<Key> marshaling scheme only works well for shared resources that are controlled by the whole network.
  - Problematic for AnchorPeers
  - Problematic for MSP definitions
  - Problematic for individual node/replica certs
- The only way for orderer to reference peer policies, is by name, requiring that the peer creates them.
  - Problematic as exhibited by IngressPolicyNames
  - Problematic as exhibited by EgressPolicyNames
  - Problematic for allowing peer orgs to add MSPs

Proposed New Structure



Leverage the hierarchical nature of the relationships, channel policies may refer to peer and orderer policies. The orderer policies may refer to the indvidual orderer org policies, etc. In this way, Peers may add peer orgs, which automatically roll up into the channel policy. Invidual orgs may edit their own properties.

Details

Channel

Policies:
- Readers = peer.Readers or orderer.Readers
- Writers = peer.Writers or orderer.Writers
- Admin = peer.Admin and orderer.Admin

Values:
- HashingAlgorithm = SHAKE256
- BlockDataHashingStructure = ...

Application for this channel

Policies:
- Readers = 1 of pOrg{1,2,3}.Readers
- Writers = 1 of pOrg{1,2,3}.Writers
- Admin = majority(pOrg{1,2,3}.Admin)

Values:
- SysCCHashes = … ?
- LSCCPolicy = bytes (marshaled peer.LSCCPolicy)

Orderer

Policies:
- Readers = oOrg1.Readers or oOrg2.Readers
- Writers = oOrg1.Writers or oOrg2.Writers
- Admin = oOrg1.Admin and oOrg2.Admin

Values:
- BatchSize = …

pOrg1

Policies:
- Readers, Writers, Admin

Values:
- AnchorPeers = [pOrg1Host, …]
- MSP = MSPConfig

pOrg2

Policies:
- Readers, Writers, Admin

Values:
- AnchorPeers = [pOrg1Host, …]
- MSP = MSPConfig

oOrg1

Policies:
- Readers, Writers, Admin

Values:
- MSP = MSPConfig

oOrg2

Policies:
- Readers, Writers, Admin

Values:
- MSP = MSPConfig

Drawbacks

- Data structure itself is more complicated
  - No longer just a single set of config, now, nested maps
- Encourages abuse of the configuration system for storing non-channel level configuration data
- Not yet implemented

Cross Channel Common Configuration

- Some channel config is common
  - MSPs
  - Consensus type
  - Block validation policy
  - Hashing algorithm
- Some channel config is not
  - Anchor peers
  - Batch size / timeout
  - ACL

Source Channel Modification Policy

- Could have a special policy type, which is “source channel” or something similar.
  - Specifies source channel ID where the config item update comes from e.g. Identity Channel
  - These items would only ever be updated via automated orderer driven updates
- Candidates include all shared config on previous slide.
- Allowable source channels could be included in the ordering system channel.
  - Identity channel / channels