ORACLE

Extending the Operations Smart Contract for Hyperledger Fabric to Support Consortia Governance

Todd Little

Oracle Blockchain Platform Chief Architect

Agenda

- Introduction to Consortia
- Introduction to Operations Smart Contract
- Consortia Governance
- Voting Enhancements
- Summary

Consortia – Blockchain is a Team Sport

- Consortium – An association of businesses
- Forming, managing, and maintaining is as much a business issue as technical issue
- Need multiple organizations to work together
- Decisions and actions must be auditable
- Ideally automate technical issues

Introduction to the Operations Smart Contract - OpsSC

- Two smart contracts
  - Chaincode Lifecycle
  - Channel Management
  - These manage the state transitions for chaincode and channel operations
- OpsSC Agent
  - Automates operation actions in response to chaincode events
  - Runs as a separate server
- OpsSC API Server
  - Provides REST APIs to access OpsSC chaincodes
  - Accepts proposals and votes
  - Runs as a separate server

- Focused on automation, not enforcement.

Chaincode Operations

- Org1 proposes new or upgraded chaincode by calling its API server
  - API server invokes OpsSC chaincode to record proposal
  - OpsSC chaincode emits event – received by all orgs agents
- Org2 and Org3 vote for proposal by calling its API server
  - API server invokes OpsSC chaincode to record vote
  - OpsSC chaincode emits event – received by all orgs agents
- When vote passes:
  - Org1's agent downloads, installs, approves, and commits the chaincode
  - Org2 and Org3's agent downloads, installs, and approves the chaincode

Channel Operations

- Org1 proposes human-readable channel update proposal by calling its API server
  - API server invokes OpsSC channel chaincode to record proposal
  - OpsSC channel chaincode converts proposal to ConfigTX
  - OpsSC chaincode emits event – received by all orgs agents
- Org2 and Org3 vote for proposal by calling their API servers
  - API server invokes OpsSC channel chaincode to record vote
  - OpsSC chaincode emits event – received by all orgs agents
- When vote passes:
  - One agent will take the converted ConfigTX and update the channel

What’s Missing?

- Consortia Governance
  - Who are the members of the consortia?
  - What can members do?
  - How can members be added or removed?

- Voting Enhancements
  - Support more than simple majority
  - Support non-uniform voting rights
  - Support abstaining and opposing

Consortia Need Additional Governance Extensions

- Focused on automation
- Moving from founder led consortia to decentralized governance
- Not adding new enforcement policies in Fabric (at least initially)
  - Nothing preventing members using standard Fabric capabilities
- All members are not necessarily equal
  - Initial proposal determines which members can do what
  - Support non-voting members
  - Support for asymmetric voting rights – Some members may have more votes than others
- All decisions and actions must be recorded in a ledger

Consortia Formation Issues

- How to exchange information?
  - No in-band mechanism available as there is no network
  - Must be partially out of band – bootstrapping issue
- What information needs to be exchanged?
  - Minimally MSP for each member
  - OSN information
- What actions must each participant perform?

Proposed Consortia Formation Flow – How to Bootstrap a Consortia

- Initially single member – “proposer”
  - Creates ordering service
  - Creates initial OpsSC channel (ops-channel)
  - Deploys OpsSC chaincodes (chaincode_ops, channel_ops, consortia_ops)
  - Starts local OpsSC agent and API server
    - Creates temporary limited API server credentials for other proposed members
- Defines and commits initial consortia formation proposal
- Emails or otherwise notifies other organizations:
  - URL for its API server
  - their temporary limited credentials
- As the other organizations provide their MSP info, agent will update the ops-channel with their info

Consortia Formation Flow – Cont'd

- Other organizations:
  - using initial members API server:
    - download initial consortia formation proposal to configure their instance
    - provide their MSP info if they choose to join
- Once initial member's agent adds invited org's MSP info the invited org will:
    - Join peer(s) to ops-channel
    - Install OpsSC chaincodes
    - Start local OpsSC agent and API server

Initial Consortium Formation Proposal

- For the consortia:
  - Name and description
  - List of initial orgs
  - Consortium voting policy (what’s required to change the consortium configuration)
  - Membership voting policy (what’s required to change membership)
- For each org:
  - Name
  - Contact information
  - Number of votes
  - Permissions
- For proposer org:
  - MSP info

Voting Policies – How is Agreement Achieved?

- Currently Fabric uses:
  - signature policies for chaincode transactions and channel config (configtx) transactions
- Signature Policy
  - Expression with one or more principals (MSP.ROLE e.g. Org1.admin)
  - Expressions consist of AND, OR, or OutOf options
    - AND(‘Org1.member’, ‘Org2.member’) – Both Org1.member and Org2.member must sign
    - AND(OR(‘Org1.member’, ‘Org2.member’),’Org3.admin’) – Org3.admin must sign and one of Org1.member or Org2.member must sign
    - AND(‘Org1.admin’, OutOf(2, ‘Org2.admin’, ‘Org3.admin’, ‘Org4.admin’)) – Org1.admin must sign and any 2 of Org2.admin, Org3.admin, Org4.admin must sign

Voting – What’s Missing?

- In OpsSC:
  - No ability to abstain
    - Can be useful to know whether an org has voted or not
  - Only supports single vote per org
  - No veto capability
- In Fabric:
  - Only “for votes” (signatures) supported
  - Above features would require Fabric enhancements

Proposal

- Add consortium formation and administration support to OpsSC
- Add enhanced voting features to OpsSC
- Add additional endorsement policies in Fabric to support enhanced voting
- Integrate OpsSC agent with a deployment tool such as Hyperledger Cello to handle initial deployment

Summary

- Consortia formation and management is hard and currently largely manual
- Leverage the existing OpsSC project to automate as much as possible
- Simplify and automate consortia operations

- To join in, join the OpsSC rocket chat channel: https://chat.hyperledger.org/channel/fabric-opssc
- Source code at: https://github.com/hyperledger-labs/fabric-opssc

Q&A