Network Orchestration Using Blockchain

Vinay Chaudhary

@_VinayChaudhary
Agenda

• About Blockchain
  – Permissionless Blockchain
  – Permissioned Blockchain
• About Hyperledger Fabric
• Multi Domain And Single Domain
• About Network Orchestrator (ONAP)
• Onap Using Hyperledger Fabric Blockchain (Single Domain Orchestration using Blockchain)
Why Blockchain?

- Blockchain adds distributed trust.
- Blockchain is a distributed network with consensus.
- Blockchain’s were built for transaction consistency in a datastore, allowing for auditability, atomicity, linearizability, durability, and most importantly, data integrity.
Which Blockchain for Orchestration?

- Public
- Private
Hyperledger At Glance

The Hyperledger Greenhouse
Business Blockchain Frameworks & Tools Hosted by Hyperledger

Frameworks
- Hyperledger Burrow: Permissionable smart contract machine (EVM)
- Hyperledger Fabric: Permmisioned with channel support
- Hyperledger Grid: WebAssembly-based project for building supply chain solutions
- Hyperledger Indy: Decentralized identity
- Hyperledger Iroha: Mobile application focus
- Hyperledger Sawtooth: Permissioned & permissionless support; EVM transaction family

Tools
- Hyperledger Aries: Infrastructure for peer-to-peer interactions
- Hyperledger Caliper: Blockchain framework benchmark platform
- Hyperledger Cello: As-a-service deployment
- Hyperledger Composer: Model and build blockchain networks
- Hyperledger Explorer: View and explore data on the blockchain
- Hyperledger Quilt: Ledger interoperability
- Hyperledger Transact: Advanced transaction execution and state management
- Hyperledger Ursa: Shared Cryptographic Library

Community Stewardship and Technical, Legal, Marketing, Organizational infrastructure
Private Blockchain: Hyperledger Fabric

source: https://medium.com/@nemesis1346/the-solo-default-configuration-for-hyperledger-composer-8189057aeb87
What is Single Domain.

- A single domain refers to a One telecom Operator or an Organization.
- A single domain can be a Multiple Subdomains in terms of Distributed Environment.

What is Single Domain Orchestrator.

A Single Domain orchestration represents a new approach to step-by-step evolution, allowing service providers to orchestrate and automate various operations at a time via “domains,” which can be defined by network layers, specific services or other parameters.
ONAP SO WorkFlow

High Level Architecture and Workflow

Run-time Environment

Service Orchestration

Common Service

Controllers

SDN-C

APP-C

VF-C

VNF (TOSCA)

Homing

Users On-board VIM/Cloud services

Registration

OOF

A&AI

ESR

ONAP Platform Deployment

Close loop remediation

DCAE

VNF (HEAT)

Network (YANG)

ONAP Operations Manager (OOM)

ONAP SO WorkFlow

Design-time Environment (SDC)

Distribute Service Model and VNF artifacts (TOSCA/HEAT/YANG)

VID/UI

Deploy Service

Multi-VIM/Cloud

Source: https://onap.readthedocs.io/en/dublin/_images/mc-arch-workflow.png

Notes for Casablanca Release:
1. SDN Overlay is not available yet.
2. Azure and Kubernetes plugins are not yet integrated into end to end workflow.
ONAP Design Time and Run Time

ONAP has two main parts.

**Design Time**
- Pre-Onboarding
- Resource Onboarding
- VF Creation and Testing
- Service Design
- Governance Approval and Service Distribution
- Importing a VFC into SDC
- Control Loop Design
- Policy Design
- SDC Artifacts List
- Onboarding Services into SDC and Instantiating through VID

**Run Time**
- Instantiate a network
- Gather and validate data for an infrastructure service/network
- Instantiate an Infrastructure service
  - VID instantiation flow
  - Verify that prerequisite configuration is in the cloud and inventory
  - Convert spreadsheet data
  - Upload conversion output to SDNC
  - Trigger instantiation of an infrastructure service using VID
  - VID instantiation errors
ONAP SO API

Source: https://wiki.onap.org/display/DW/Release+Planning
AnAI is a ONAP Sub module which provides real-time views of Resources and Services and Relationship Between them. It’s an Central Repository of active, available and assigned assets in Onap. It also maintain the multidimensional relationship among these assets.
Information supplied by AAI needs to be accurate to minimize fallout in operational processes.

- Maintaining an accurate view of the service/resource is made more challenging by Increased autonomy within the network
- AAI needs mechanisms to ensure tight alignment with network, without significant additional load to the controllers. etc.
- Need to refresh from VIM
  - data integrity checks and reconciliation
  - event based updates from VIM/SO/Controller
- Different VIMs have different levels of detail.
- Variety of different options here (VIM/SO/Controller) - all valid.
Onap On Blockchain

Audit & SLA Management
Service Specification Verification
Service Change Event Based on Smart Contract to BSS
Inventory Assets and Relationship Among Assets

Source: BSS/OSS notification Reg and SDC Distribution Using Blockchain.pptx
Onap AnAI and DCAE notification Using Blockchain

Solution Using Blockchain (As an Smart Inventory)
• Smart Contract for available Resources and Services.
• All Resources and Services are Assets in Blockchain.
• Smart Contract provide the automation SLA settlement for BSS.
• Any Automation for future need can be done using Smart Contract.

Feature Of Blockchain Smart Contract
• Smart Contract for Reputation Management of Active VNFs based on VES.
• Smart Contract provisioning reliability between SDC and SO.
• Smart Contract provisioning the Automation of validation at the time of Services instantiation by SO.
• Provide trust between modules in ONAP.
• Auditing by Smart Contract by Calling Ext API of Blockchain by BSS.

Future Expectation Using Blockchain
• Authenticity from AAF.
Assets on Blockchain at SDC time (Same as AAI)

**
* A tenant asset.
*/
asset tenant identified by tenant-id{
 o String tenant-id  //Unique id relative to the cloud-region.
 o String tenant-name
 o String tenant-context
 o String resource-version
 -->vservers vservers
 -->nos-server nos-servers
}

/**
* A cloud-region asset.
*/
asset cloud-region identified by cloud-region-id{
 o String cloud-region-id
 o String cloud-owner
}

/**
* A model asset.
*/
asset model identified by model-invariant-id{
 o String model-invariant-id
 o String model-type  //service, resource, widget, etc.)
 o String resource-version
 -->model-ver model-vers
}

/**
* A service asset.
*/
asset service identified by service-id{
 o String service-id
 o String service-description
 o String service-selflink
 o String service-version
 o String resource-version
 -->service-capability service-capability
}

/**
* A vnf-image asset.
*/
asset vnfc-image identified by service-id{
 o String vnfc-name  //Unique ID of vnfc
 o String vnfc-function
 o String prov-status
 o String orchestration-status //(Orchestration status of this VNF, mastered by APP-C)
 o String ipaddress-v4-oam-vip
 o Boolean in-maint  //(maintenance mode = true)
 o Boolean is-closed-loop-disabled
 o String group-notation
 o String model-customization-id
 o String resource-version
 -->model-ver model-version-id
 -->model model-invariant-id
 -->cp cps
 -->l3-interface-ipv4-address-list l3-interface-ipv4-address-list
 -->l3-interface-ipv6-address-list l3-interface-ipv6-address-list
}

Cloud Owner
Cloud Region Id
Cloud Type
Cloud Region Version
Owner Defined Type
Cloud Zone
Complex Name
username
password
Aff IR
ssl CA
ssl Insecure
Cloud Domain
Default Tenant
Cloud Extra Info
BSS registers listener to be notified of ONAP new Service Specification Creation. BSS is notified that a new ONAP Service has been created. BSS can view the ONAP service Catalog. BSS can then incorporate the new ONAP Service into Product offerings (add pricing etc.).

POST {User/Api call for SDC for VNFs}

Post Return the DistributionID with all required params

POST sdc/v1/registerForDistribution

GET nbi/api/v3/serviceSpecification/{uuid}/specificationInputsSchema

Generate and Persist specificationInputsSchema

POST sdc/v1/registerForDistribution

POST nbi/api/v3/hub

POST sdc/v3/catalog/bc/service/{serviceInstanceId}/metadata

Generate and Persist specificationInputsSchema

ONAP SDC distributes a new service to the ONAP Runtime. External API uses the sdc distribution client to register for notifications of SDC distribution events for new Service Specifications the become available for use in the run-time.
Service Creation At Run Time (ONAP) Using Blockchain

BSS/OSS

POST nbi/api/v3/serviceOrder

{"externalId": "BSSProvidedTrackingID", "priority": "1", "orderItem": [{ "id": "1", "action": "add", "service": { "serviceSpecification": { "id": "81b9430b-8abe", "name": ..., "serviceCharacteristic": [{ "name": "EPL_ServiceCharacteristics", "valueType": "Object", "value": { "@type": "EPL_ONAPserviceCharacteristic", "@schemaLocation": "/serviceSpecification/{uuid}/apiSchema", "evclid": "2CTYHRIUTR" } } ] } }, "orderItem": [{ "id": "1", "action": "add", "state": "ACKNOWLEDGED" } ]

Return ServiceOrder id and orderItem(s) serviceinstance Ids

Ext API/VID

GET sdc/v3/catalog/services/(id)/metadata

Return ServiceInstanceId and OperationId for Internal ONAP tracking

SDC

POST so/infra/e2eServiceInstances/v3

ONAP validates the Service Order Details against the Service Specification details. ONAP then orchestrates the fulfilment Of the Service by calling the relevant Controllers to deliver the service according to its specification.

SO

GET blockchain/v3/catalog/services/(id)/metadata

AII

PUT /business/customers/customer/(global-customer-id)/service-subscriptions/service-subscription/(service-type)/service-instance/(service-instance-id)

PUT /blockchain/customers/customer/(global-customer-id)/service-subscriptions/service-subscription/(service-type)/service-instances/service-instance/(service-instance-id)

BlockChain

Source: BSS/OSS notification Reg and SDC Distribution Using Blockchain.pptx
For further questions reach out to me at - vcteotia@gmail.com

Here is the link (TCSIG)
https://wiki.hyperledger.org/display/TCSIG/Telecom+SIG

Thank you