AT&T and OpenDaylight: A Platform for Programmable Service Logic

Brian Freeman
Distinguished Member of Technical Staff
Introduction

• AT&T has developed the Service Logic Interpreter as an application on top of Opendaylight to create a Platform for Programmable Service Logic
• Using a domain specific language based on a-cyclic directed graphs, AT&T has implemented a model driven application which enables service designers to create software defined network solutions within the ECOMP framework.
• The talk will cover a brief overview of
  • the Service Logic Interpreter language,
  • the graphical user interface,
  • the integration points with OpenDaylight and
  • enhancements to OpenDaylight that would be useful for the future.
What is “A Platform for Programmable Service Logic”?

• An application that interacts with the network through APIs
• An application that does resource assignment through micro-service APIs.
• An application that implements Service Provider engineering rules and maintains customer data.
• An application that has a language for dealing with
  • the service, network and device models
  • The service model to network model mapping
  • The network model to device model mapping.
• An application that can deploy new service or update existing services without recompiling the controller code.
Goals of “A Platform for Programmable Service Logic”? 

• Hide complexity from service designers
• Create reusable functions for service designers
• Create a mechanism to change network and customer logic without a code release
• Create a mechanism to specify how to fill out or map the data in the various yang models
• Create a mechanism to define the Northbound “Service” API via code generation
• Provide the ability for service designers to add “user defined” functions to the language.
AT&T SDN Controller Framework

- AT&T SDN Controller (SDN-C)
- API Handlers
- Service Logic Interpreter
- SDN-C Database
- Network Adapters
- Service-Related Artifacts for SLI, API Handlers, Network Adapters
- Directed Graph
- REST API (REST)
- Control Loop Applications
- Service Orchestrators
- Security Applications
- External API Calls
- A&AI
- Inventory

OpenDaylight at AT&T with AT&T customizations
How does a Service Designer view the internals

- **Northbound “API Handler”** (parses API calls)
  - API Handler is auto-generated from NBI Yang using ODL “yangtools” with AT&T glue for RPCs
  - Simple helper class to read md-sal, populate context, find DG and start execution.

- **The “Directed Graphs”**
  - set of “nodes” that control the flow of data through SDN-C
  - Example: Service-configuration-operation (black)
  - Example: Service-configuration-notification (red)
  - Maps data from input (service model) to network model
  - Maps data from network model to adaptor (device model)

- **The “Service Logic Interface”**
  - Java-based engine that interprets/processes the DGs

- **A “Config Node”** - action node in the DG
  - Java plug-in that generates the API calls to an adaptor
Hooking the SLI into OpenDaylight

Two main mechanism for hooking the SLI into ODL

- **REST RPC**
  - Majority of the NBI interfaces use this mechanism since they are synchronous.
  - Permits the Directed Graph to control the HTTP response

- **OnDataChange/OnDataTreeChange Notification**
  - Used for Neutron ML2 Agent processing and a few other services.
  - Originally would have liked to use this for more but no On-Commit hook so we couldn’t control the HTTP response.
  - Neutron and a few other NBI were asynchronous so we could use OnDataTreeChange
  - Prefer this method since we use less “helper” code around the builder classes.
What does a Directed Graph Look Like?
Yang model
Module L3SDN-API {
    ... 
    rpc svc-configuration-operation {
        uses, container, list, leaf ... 
    }

    ...
Example of Directed Graph Builder - [Demo of creating a Directed Graph and Important Nodes in Language]
Service Logic Interpreter basic nodes

- Flow Control
  - block
  - call
  - for
  - return
  - set
  - switch

- Device Management
  - configure

- Java Plugin Support
  - execute

- Recording
  - record

- Resource Management
  - delete
  - exists
  - get-resource
  - is-available
  - notify
  - release
  - reserve
  - save
  - update
Examples of User Defined Nodes

- **Execute Nodes:**
  - IpAddressTools
  - Generic RestApiCallNode
  - IP Address Management Node

- **Configure Nodes:**
  - BGP Adaptor
  - PCEP Adaptor
  - Legacy CLI Adaptor
  - Smart CLI Adaptor
  - Netconf Adaptor

```java
public void someFunction(Map<String, String> parameters, SvcLogicContext ctx) {
```

```java
public ConfigStatus configure(String key, Map<String, String> parameters, SvcLogicContext ctx) {
```
Applications of the Programmable Platform

• **SDN-G**: Resource Assignment and Inventory Update & L3 VNF Configuration
• **APPC**: L4-L7 VNF Configuration & VNF Management
• **SDN-FC**: IP Flow Redirection
  BGP ipv4 unicast, ipv4 flowspec, etc
• **SDN-MLC**: MPLS Traffic Engineering and L0 Federated Controller Interface
  BGP-LS, PCEP
• **SDN-W**: NetBond
• **SDN-O**: Operations
Deployment Options

Central

Regional/Nodal

MoW Regions

© 2016 AT&T Intellectual Property. All rights reserved. AT&T, Globe logo, Mobilizing Your World and DIRECTV are registered trademarks and service marks of AT&T Intellectual Property and/or AT&T affiliated companies. All other marks are the property of their respective owners.
Enhancements to OpenDaylight

- **Multi-threaded onDataTreeChange Notification**
  For some application like BGP, a form of multi-threading would be a nice performance enhancement. Understand that it would not make sense in all environments. This could be tied to general adaptor elastic scaling.

- **MD-SAL Shard Management utilities**
  When we store data in MD-SAL it would be useful to be able to do micro-sharding and split the processing across multiple nodes. General desire to have more control of where and how many nodes are running RPCs.

- **On-commit handler**
  So we can call the SLI before committing data to the MD-SAL on a PUT/POST/DELETE via RESTCONF.

- **Better ENUM handling**
  So we can convert Java ENUM back to original Yang value.

- **Better Disaster Recovery site synchronization options**
  Today we do snapshot copies of MD-SAL to the DR site. We plan to use non-voting cluster member in DR site. Would like to consider eventually consistent replication or update the DR site with some low overhead standard mechanism.
Q&A
Things that we use the most (today)

- Yangtools
- MD-SAL
- Clustering
- BGP-LS
- PCEP
- BGP
- Netconf
- Northbound Services (MSO and other AT&T OSSs)
- Custom Adaptors
  - East/West Adaptors to AT&T OSS’s
  - Southbound REST Adaptors
  - Southbound CLI VNF Adaptors

Over 30 AT&T Specific Karaf Feature Bundles