ODL Restconf Bierman Analysis: odl-restconf-nb-bierman02

Summary

1. Odl-restconf-bierman is having around 20Kloc of code.
2. Odl-restconf-bierman is installed as a karaf feature.
   a. All required dependent feature/bundle (provided by ODL) are also used by other ODL features (like odl-restconf-8040, netconf)
      Note: Couldn't find usage of some of dependent bundle listed as part of restonf-bierman feature.xml (would need to check with ODL community)
3. Some amount of code clean up would be better while porting this odl-restconf-bierman code in ONAP (like remove some unused code etc.). Any other ONAP guideline to be followed like test coverage?

Open Points

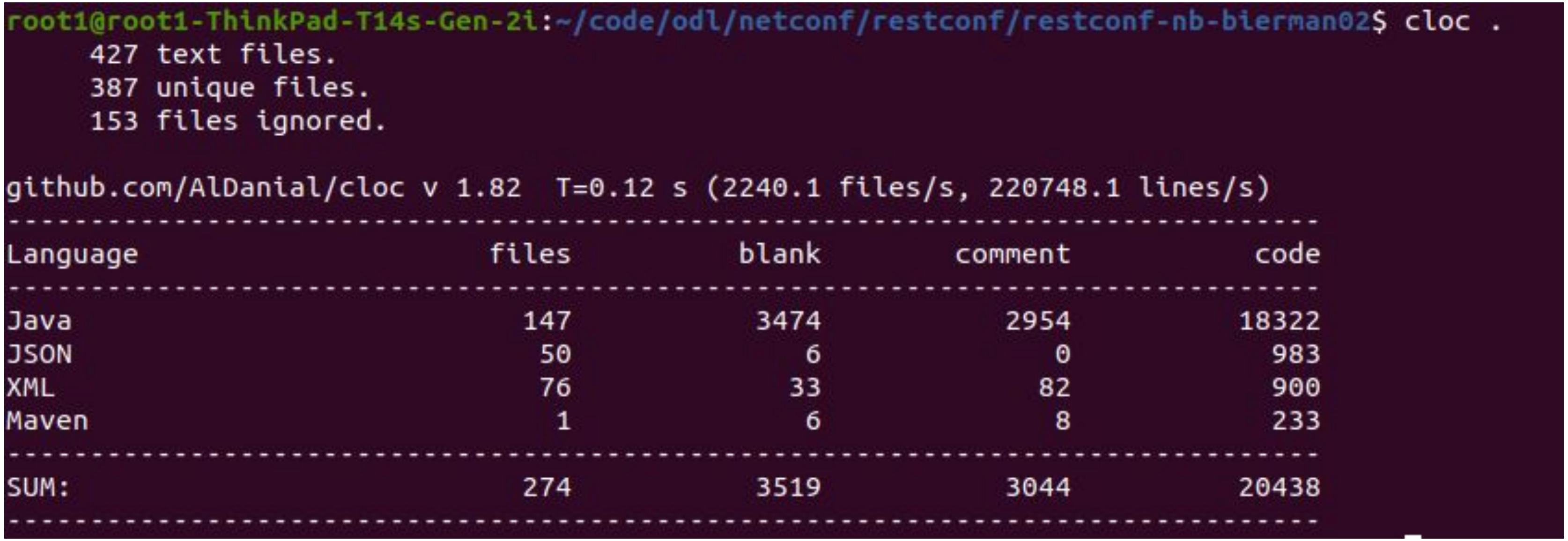
1. ONAP security specification check on ODL code.
2. Test coverage as per ONAP standards (Dan already confirmed with ODL community)
3. License compatibility and acceptance to port the code.

Challenges

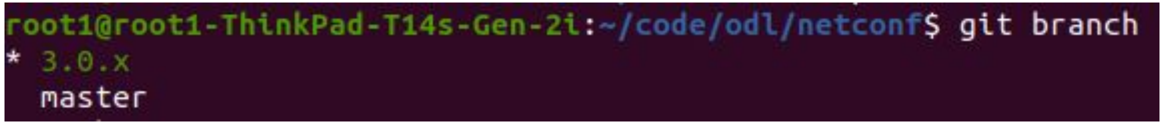
1. It's a good amount of code to be maintained in ONAP.
2. Any changes to dependent features/bundles would need to be incorporated in this code for every ODL version upgrade.

Maintaining this code for long term in ONAP might not be a good option. But maintaining for a short term, while deprecating restconf-beirman in ONAP can still be considered as one of the option.

Code count



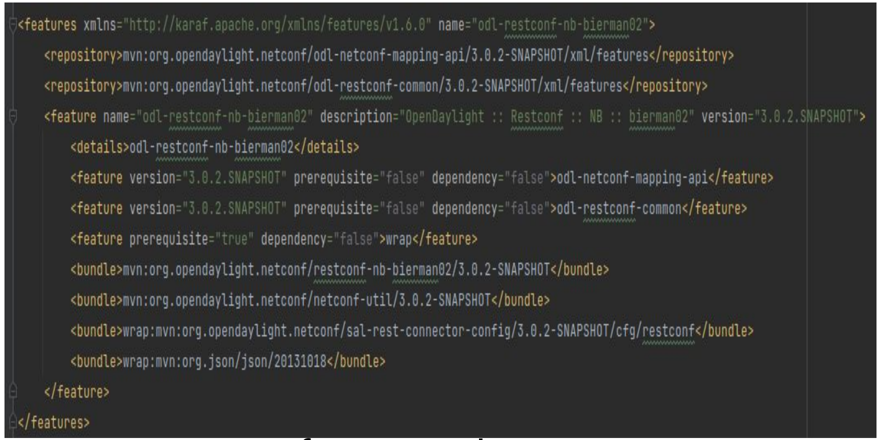
```
root1@root1-ThinkPad-T14s-Gen-2i:~/code/odl/netconf/restconf/restconf-nb-bierman02$ cloc .
     427 text files.
     387 unique files.
     153 files ignored.

github.com/AlDanial/cloc v 1.82  T=0.12 s (2240.1 files/s, 220748.1 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Java                           147           3474           2954          18322
JSON                            50              6              0            983
XML                             76             33             82            900
Maven                            1              6              8            233
-------------------------------------------------------------------------------
SUM:                           274           3519           3044          20438
-------------------------------------------------------------------------------
```



```
root1@root1-ThinkPad-T14s-Gen-2i:~/code/odl/netconf$ git branch
* 3.0.x
  master
```

restconf-beirman feature dependency analysis

Feature/bundle	**Remarks/Usage**
MD SAL Restconf Connector (odl-restconf-bn-bierman02)	odl-restconf-bn-bierman02 bundle itself
netconf-util	Used to translate xml/json to ODL Normalized Node **Note**: It's also used by NETCONF
odl-restconf-common	Common classes which are also used by restconf-8040
json-20131018	Some utilities from this bundle used to convert XML to JSON Note: This is not provided by ODL
netconf-mapping-api	**Unused in code**
netconf_sal-rest-connector-config _3.0.2_sal-rest	**Unused in code**



```xml
<features xmlns="http://karaf.apache.org/xmlns/features/v1.6.0" name="odl-restconf-nb-bierman02">
    <repository>mvn:org.opendaylight.netconf/odl-netconf-mapping-api/3.0.2-SNAPSHOT/xml/features</repository>
    <repository>mvn:org.opendaylight.netconf/odl-restconf-common/3.0.2-SNAPSHOT/xml/features</repository>
    <feature name="odl-restconf-nb-bierman02" description="OpenDaylight :: Restconf :: NB :: bierman02" version="3.0.2.SNAPSHOT">
        <details>odl-restconf-nb-bierman02</details>
        <feature version="3.0.2.SNAPSHOT" prerequisite="false" dependency="false">odl-netconf-mapping-api</feature>
        <feature version="3.0.2.SNAPSHOT" prerequisite="false" dependency="false">odl-restconf-common</feature>
        <feature prerequisite="true" dependency="false">wrap</feature>
        <bundle>mvn:org.opendaylight.netconf/restconf-nb-bierman02/3.0.2-SNAPSHOT</bundle>
        <bundle>mvn:org.opendaylight.netconf/netconf-util/3.0.2-SNAPSHOT</bundle>
        <bundle>wrap:mvn:org.opendaylight.netconf/sal-rest-connector-config/3.0.2-SNAPSHOT/cfg/restconf</bundle>
        <bundle>wrap:mvn:org.json/json/20131018</bundle>
    </feature>
</features>
```

feature.xml

TODO: Check with ODL community about the two unused dependencies to cross verify if we are missing something.

pom dependency

ODL YANGTools dependency

1. org.opendaylight.yangtools.yang-data-api
2. org.opendaylight.yangtools.yang-data-impl
3. org.opendaylight.yangtools.yang-model-util
4. org.opendaylight.yangtools.yang-data-codec-gson
5. org.opendaylight.yangtools.yang-data-codec-xml
6. org.opendaylight.yangtools.yang-model-export

MDSAL dependency

1. org.opendaylight.mdsal.mdsal-dom-api
2. org.opendaylight.mdsal.mdsal-dom-spi
3. org.opendaylight.mdsal.binding.model.ietf.rfc6991-ietf-inet-types
4. org.opendaylight.mdsal.binding.model.ietf.rfc6991-ietf-yang-types
5. org.opendaylight.controller.sal-common-util

Netconf dependency

1. org.opendaylight.netconf.restconf-common-models
2. org.opendaylight.netconf.restconf-common
3. org.opendaylight.netconf.netconf-util

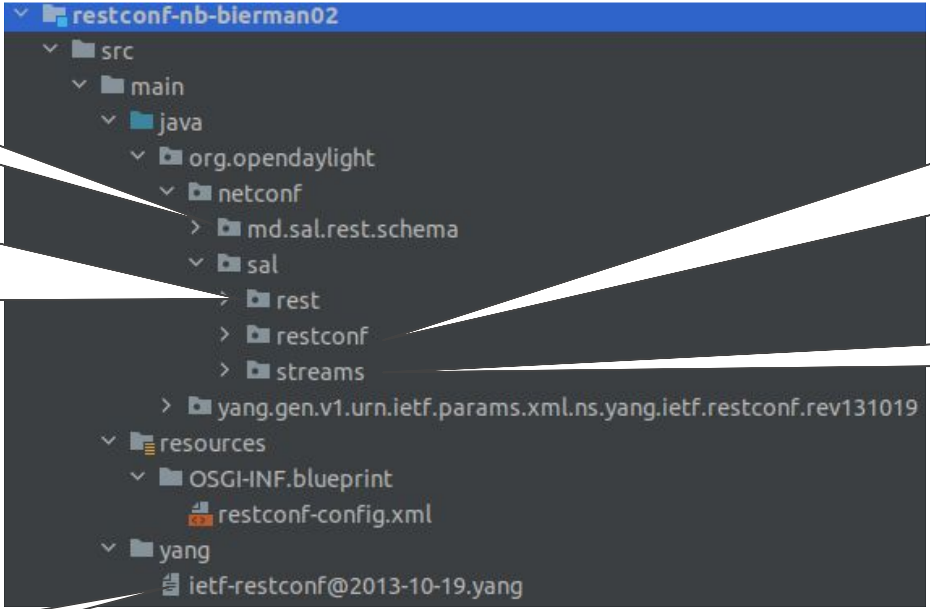
AAA dependency

1. org.opendaylight.aaa.web.web-api
2. org.opendaylight.aaa.web.servlet-api
3. org.opendaylight.aaa.aaa-filterchain

Other dependency

1. javax.annotation.javax.annotation-api
2. javax.inject
3. com.google.code.gson.gson
4. io.netty.netty-codec-http
5. net.java.dev.stax-utils.stax-utils
6. org.json.json:20131018

Code Analysis (high level)



```
restconf-nb-bierman02
  src
    main
      java
        org.opendaylight
          netconf
            md.sal.rest.schema
            sal
              rest
              restconf
              streams
          yang.gen.v1.urn.ietf.params.xml.ns.yang.ietf.restconf.rev131019
      resources
        OSGI-INF.blueprint
          restconf-config.xml
      yang
        ietf-restconf@2013-10-19.yang
```

Contains Schema Retrieval Service to fetch YANG module schema via YANG Tools

Contains RestconfService and Implementation of Restconf operations with input/output in ODL NormalizedNode format.

Also contains logic to interfaces with MDSAL Broker.

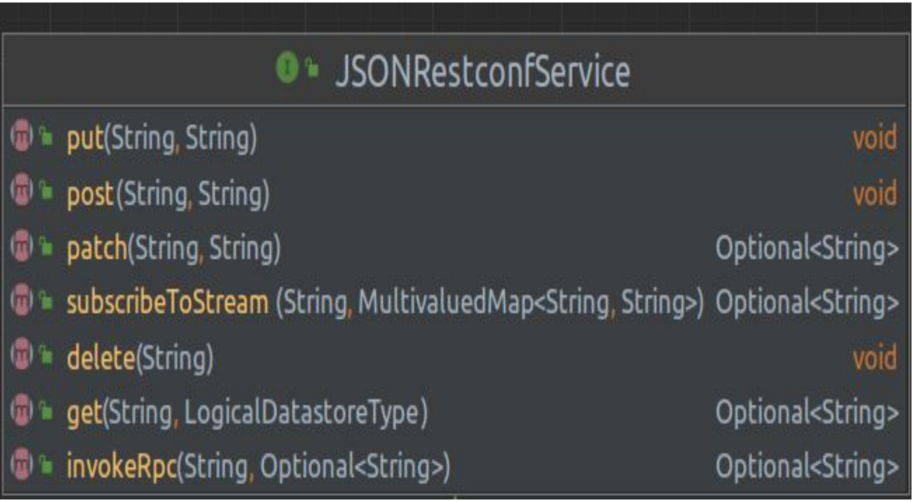
Restconf-beirman YANG

Contains JsonRestconfService and Implementation of Restconf operations with input/output in JSON format.

It contains logic to create NormalizedNode and invoke RestconfService

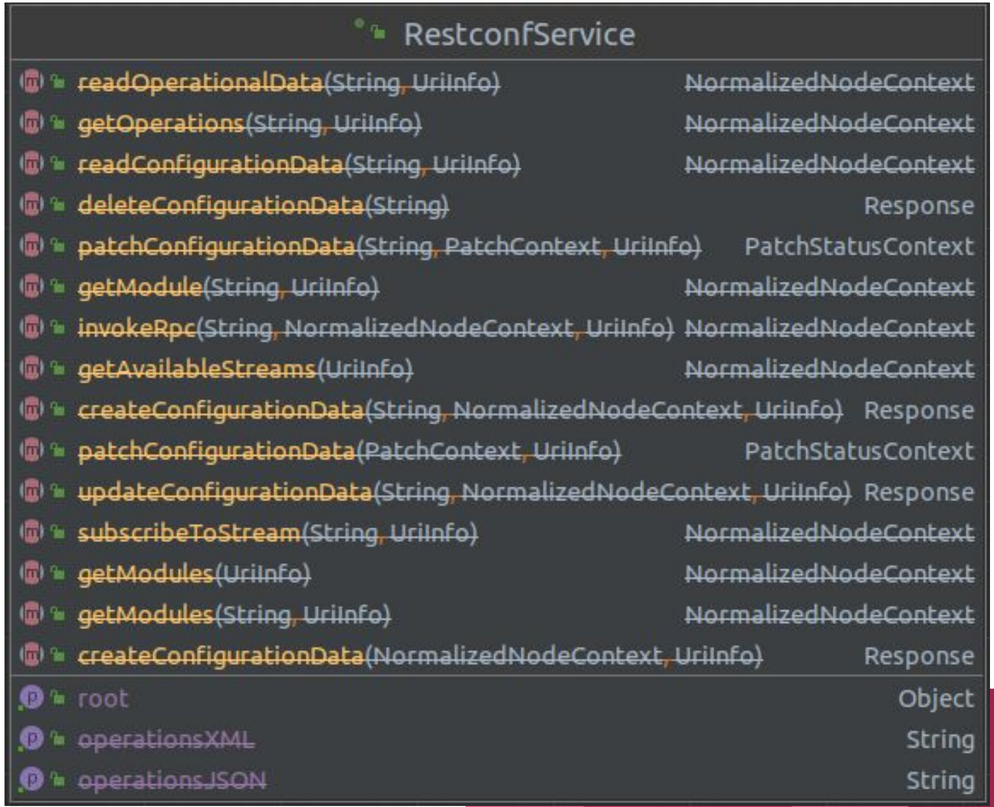
Contains Code related to YANG notifications

RestconfService Api’s



JSONRestconfService

put(String, String)	void
post(String, String)	void
patch(String, String)	Optional<String>
subscribeToStream (String, MultivaluedMap<String, String>)	Optional<String>
delete(String)	void
get(String, LogicalDatastoreType)	Optional<String>
invokeRpc(String, Optional<String>)	Optional<String>

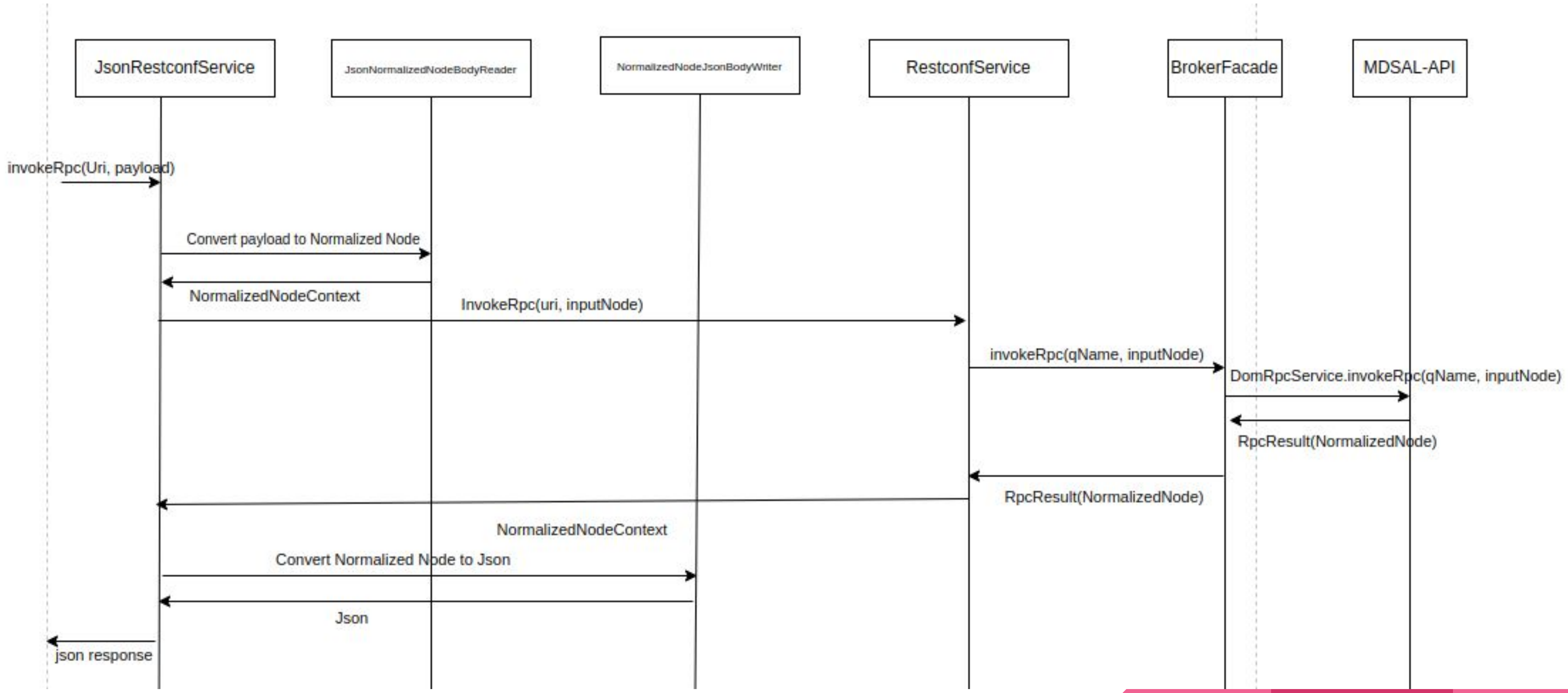


RestconfService

~~readOperationalData(String, UriInfo)~~	~~NormalizedNodeContext~~
~~getOperations(String, UriInfo)~~	~~NormalizedNodeContext~~
~~readConfigurationData(String, UriInfo)~~	~~NormalizedNodeContext~~
~~deleteConfigurationData(String)~~	Response
~~patchConfigurationData(String, PatchContext, UriInfo)~~	PatchStatusContext
~~getModule(String, UriInfo)~~	~~NormalizedNodeContext~~
~~invokeRpc(String, NormalizedNodeContext, UriInfo)~~	~~NormalizedNodeContext~~
~~getAvailableStreams(UriInfo)~~	~~NormalizedNodeContext~~
~~createConfigurationData(String, NormalizedNodeContext, UriInfo)~~	Response
~~patchConfigurationData(PatchContext, UriInfo)~~	PatchStatusContext
~~updateConfigurationData(String, NormalizedNodeContext, UriInfo)~~	Response
~~subscribeToStream(String, UriInfo)~~	~~NormalizedNodeContext~~
~~getModules(UriInfo)~~	~~NormalizedNodeContext~~
~~getModules(String, UriInfo)~~	~~NormalizedNodeContext~~
~~createConfigurationData(NormalizedNodeContext, UriInfo)~~	Response
root	Object
~~operationsXML~~	String
~~operationsJSON~~	String

Code Flow diagram (RPC)

TODO: Flow needs to be re-looked to see whether JsonRestconfService is still in use.


JsonRestconfService
JsonNormalizedNodeBodyReader
NormalizedNodeJsonBodyWriter
RestconfService
BrokerFacade
MDSAL-API
invokeRpc(Uri, payload)
Convert payload to Normalized Node
NormalizedNodeContext
InvokeRpc(uri, inputNode)
invokeRpc(qName, inputNode)
DomRpcService.invokeRpc(qName, inputNode)
RpcResult(NormalizedNode)
RpcResult(NormalizedNode)
NormalizedNodeContext
Convert Normalized Node to Json
Json
json response

Thank you