**Aug 10, 2017**

**Abstract**

This document contains the design and code change for Establish-subscription RPC in stable boron release

**OpenDaylight stable Boron Documentation**

**RPC Establish Subscription Design Document**

**Contents**

Contents

[**Introduction** 2](#_Toc492639172)

[**Scope** 2](#_Toc492639173)

[**Assumptions** 2](#_Toc492639174)

[**Software component overview** 4](#_Toc492639175)

[**Flow diagrams** 5](#_Toc492639176)

[YangpushProvider module 5](#_Toc492639177)

[YangPushSubscriptionEngine module 7](#_Toc492639178)

[YangpushRPCImpl module 8](#_Toc492639179)

[YangpushListener module 11](#_Toc492639180)

[**Sequence diagram** 13](#_Toc492639181)

[**RPC Design** 14](#_Toc492639182)

[**Steps to develop an OpenDaylight YANG: PUBSUB application** 14](#_Toc492639183)

[Setup the working directory with yang-push code 14](#_Toc492639184)

[Building, starting and verifying the yang-push code 14](#_Toc492639185)

[YANG definition for RPC Establish-subscription in ietf-event-notifications.yang 15](#_Toc492639186)

## **Introduction**

The OpenDaylight project is an open source platform for Software Defined Networking (SDN) that uses open protocols to provide centralized, programmatic control and network device monitoring.

OpenDaylight provides an interface that allows you to connect network devices quickly and intelligently for optimal network performance.

The YANG PUBSUB Stable Boron project allows subscriptions to be placed on targeted subtrees of YANG datastores residing on remote devices. Changes in YANG objects within the remote subtree can be pushed to an OpenDaylight controller as specified without a requiring the controller to make a continuous set of fetch requests

OpenDaylight installation guide ODL\_Boron\_Build\_Installation\_Guide takes you through the installation process for Stable Boron release.



## **Scope**

The intent is to have all code being able to integrate with the IETF draft in the IETF NETCONF WG.

* Client implementation, which can operate with ietf-yang-push@2016-10-28.yang and ietf-event-notifications@2016-10-27.yang compliant datastores.
* Enhance Netconf protocol client API to enable data subscription

## **Assumptions**

The following assumptions noted with regard to the content and the purpose of this document.

* The implementation of new RPC establish-subscription testing will be performed only with the devices, which has the support of yang push with the establish-subscription.
* Create-subscription RPC implementation available but not tested
* The devices which supports establish subscription also supports create-subscription
* There is no implimention provided for modify subscription.

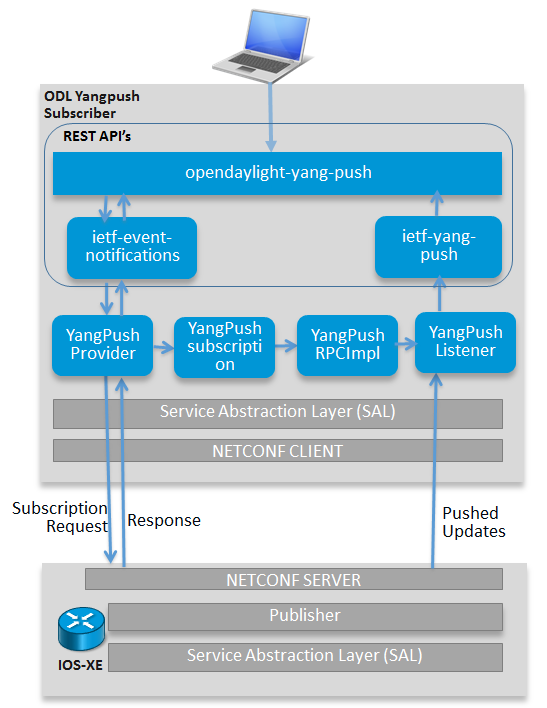
## **Software component overview**

Yang Push consists of two components YANGPUSH Provider and YANGPUSH Listener

YANGPUSH Provider receives create-subscription requests from applications and then establishes/registers the corresponding listener, which will receive information pushed by a publisher. In addition, YANGPUSH Provider also invokes an augmented OpenDaylight create-subscription RPC which enables applications to register for notification as per rfc5277.

The YANGPUSH Listener accepts update notifications from a device after they have been de-encapsulated from the NETCONF transport. The YANGPUSH Listener then passes these updates to MD-SAL. Applications should monitor MD-SAL for the availability of newly pushed subscription updates.

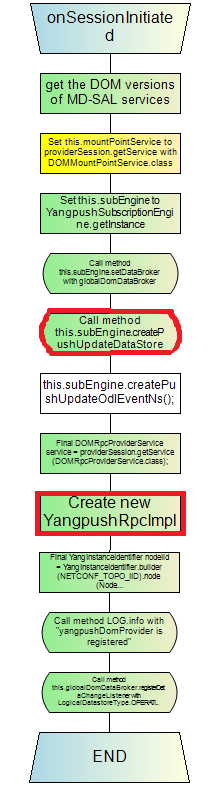
More data can found in [draft-ietf-netconf-yang-push-00](https://tools.ietf.org/html/draft-ietf-netconf-yang-push-00)

****

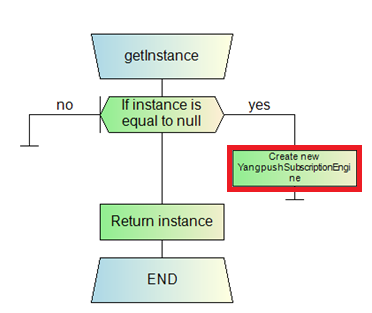
## **Flow diagrams**

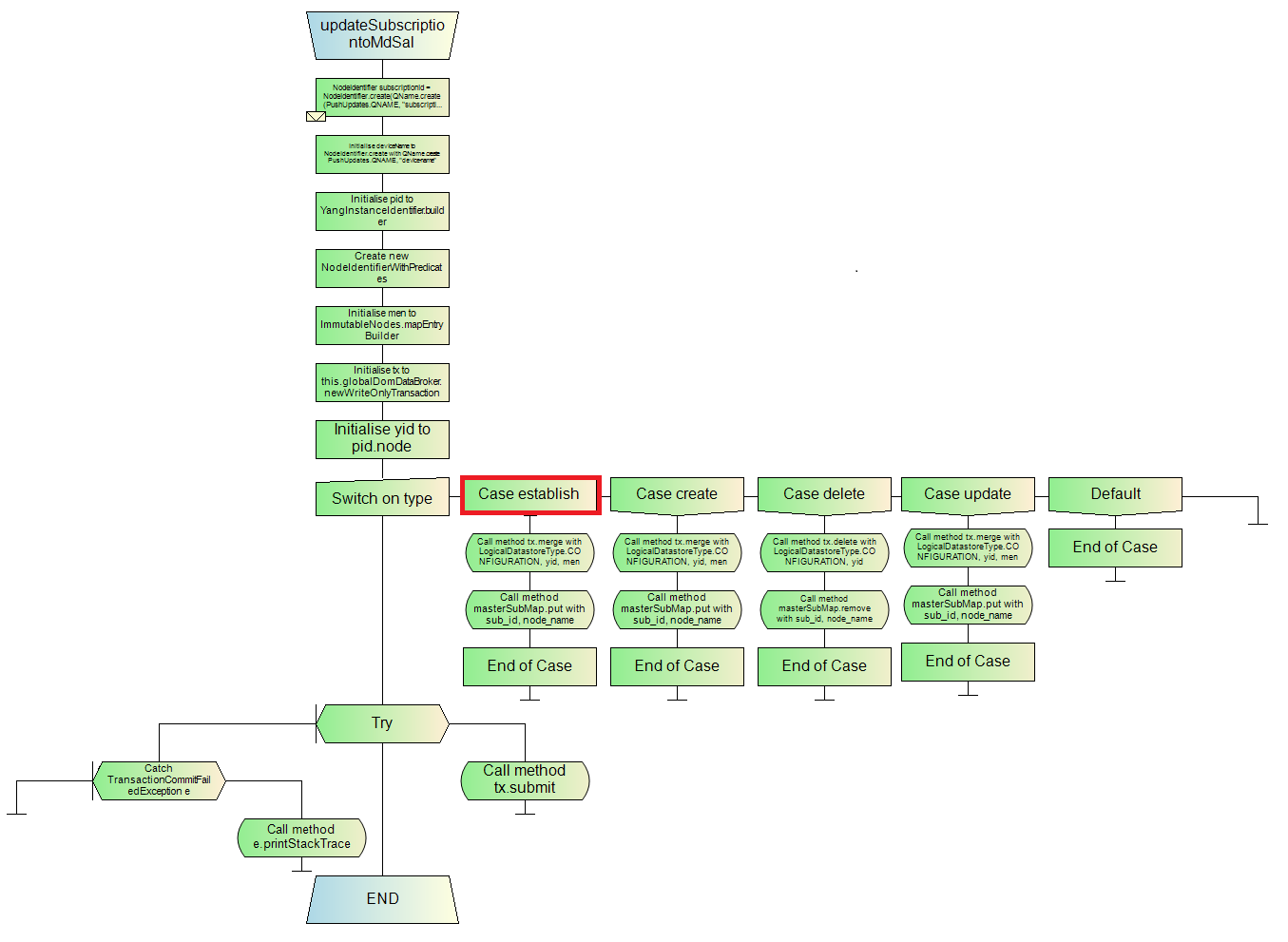
### YangpushProvider module

The method OnSessionInitiated initializes DomDataBroker and Mountpoint service. This service needed throughout the lifetime of the yangpush application and registers its RPC implementation and Data change Listener with the MD-SAL.



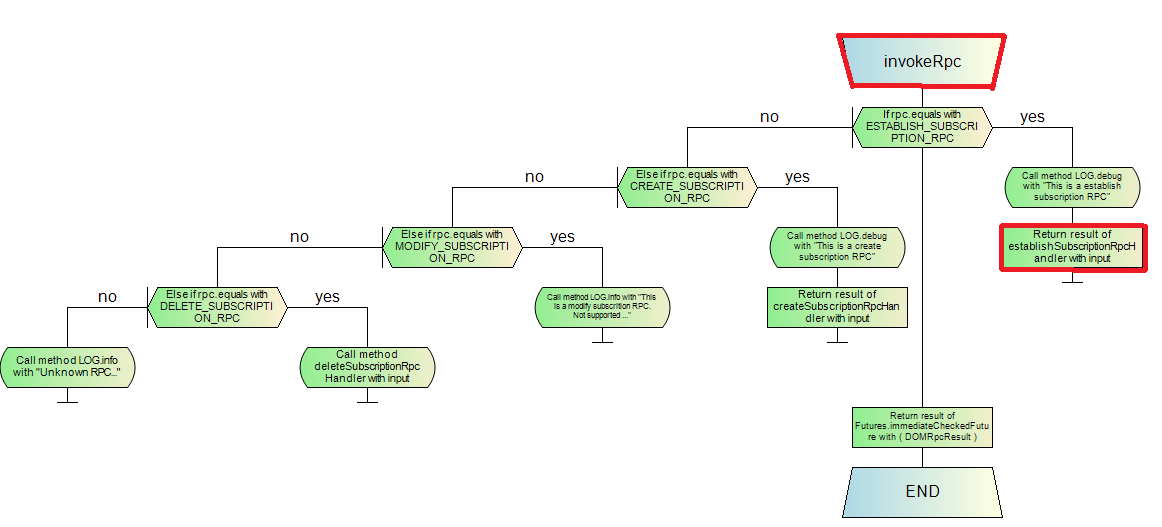
### YangPushSubscriptionEngine module

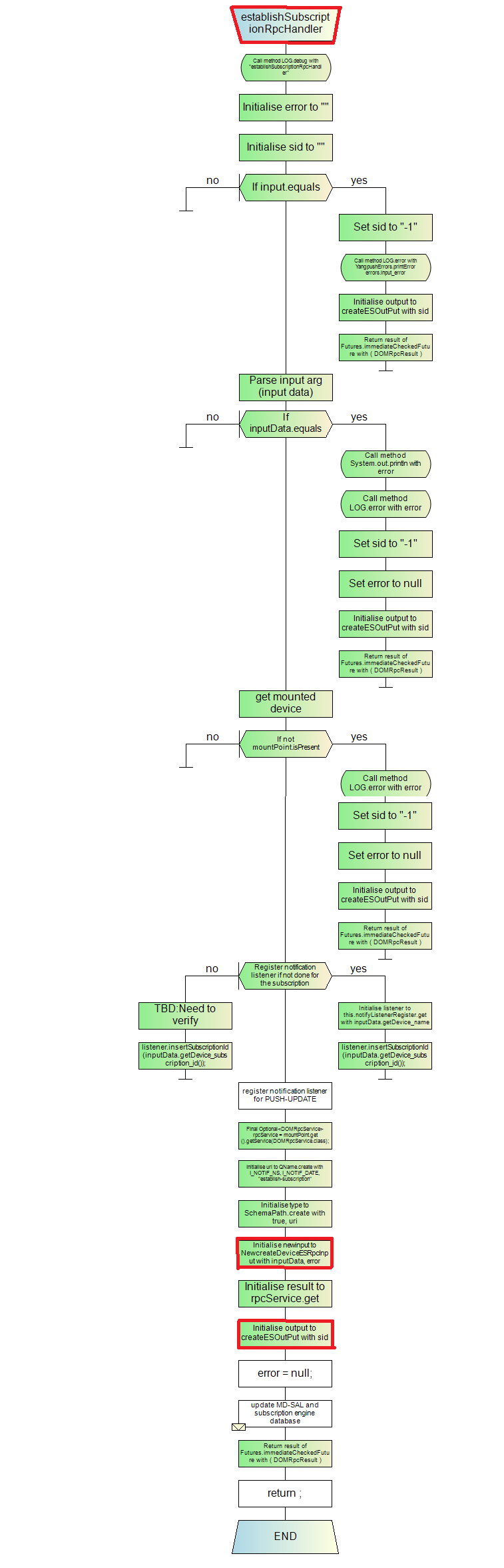
This is a singleton class for handing all subscription related data for YANG-PUSH at MD-SAL.



### YangpushRPCImpl module

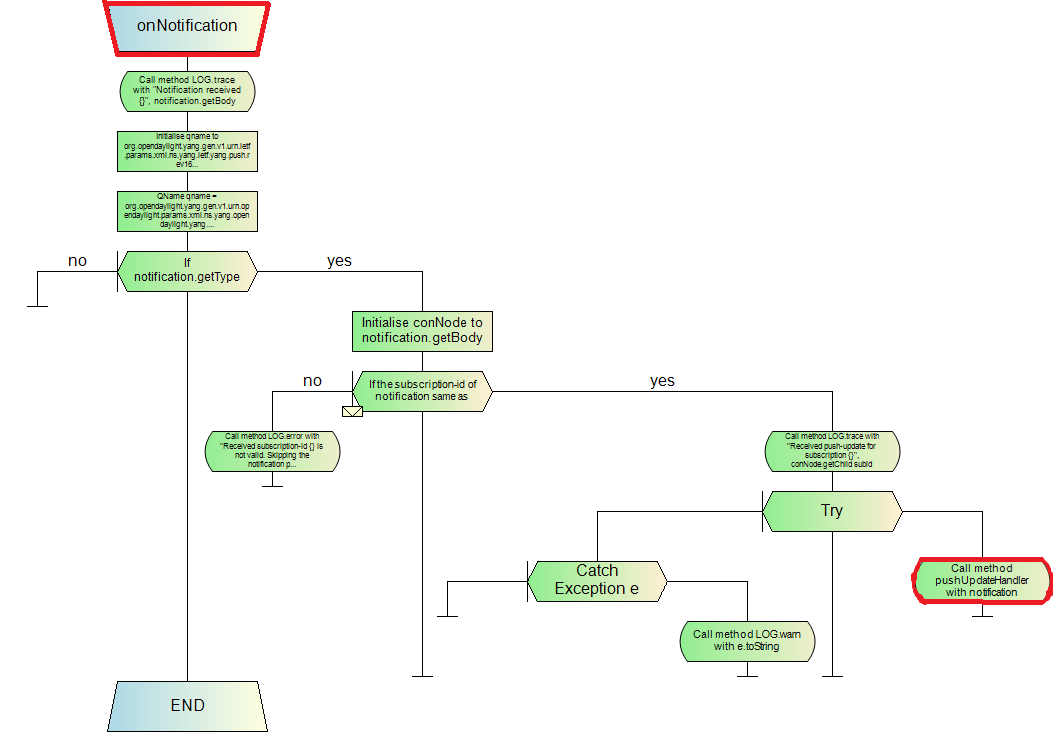
This module implements RPC defined in ietf-yang-push@2016-10-28.yang yang model and registers the Notification listener for the notification defined in the model. This is the BI implementation of handling all RPC.

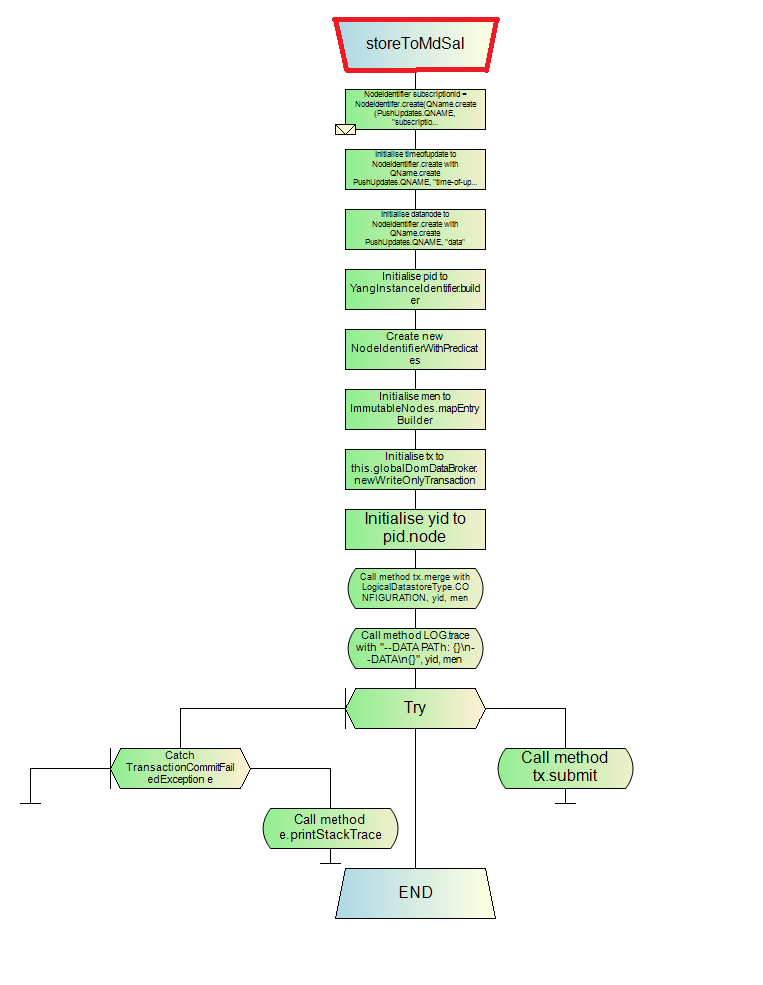
****

****

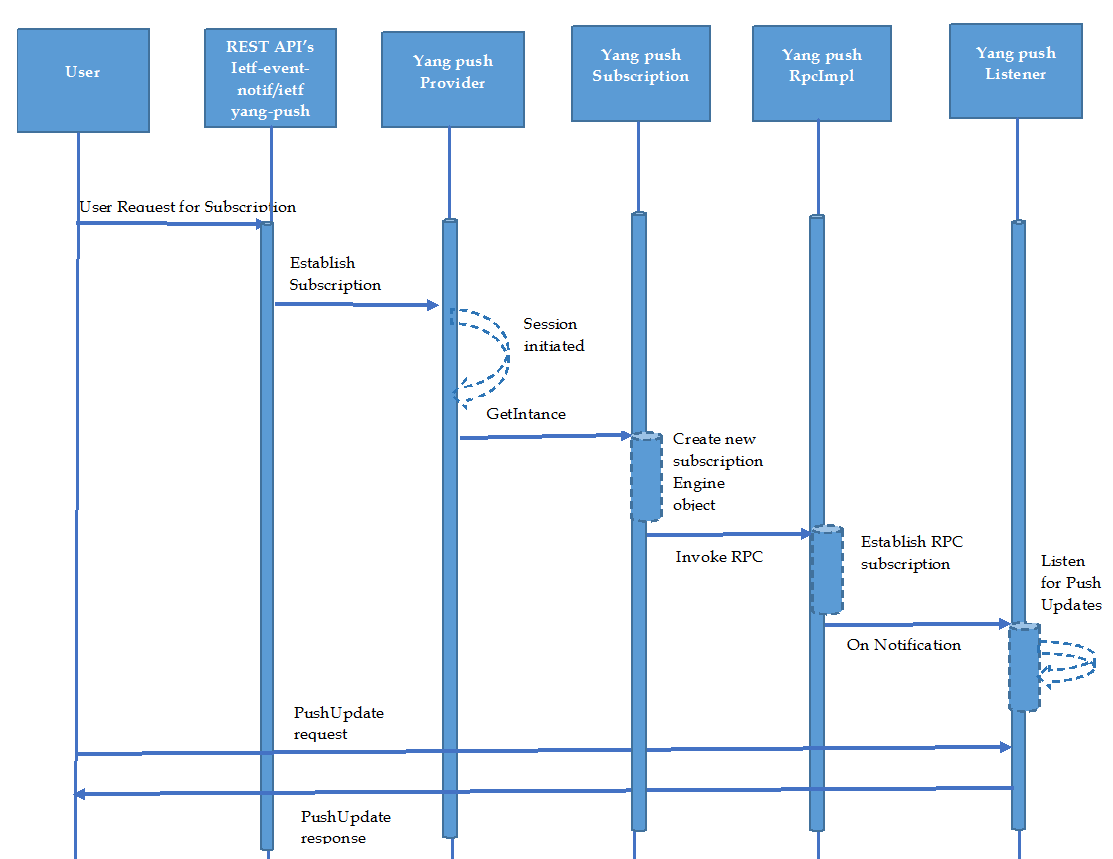
### YangpushListener module

Notification listener for the notification defined in ietf-event-notifications.yang model.





## **Sequence diagram**



## **RPC Design**

This section provides the detailed steps for establish-subscription rpc for yang pubsub project.

## **Steps to develop an OpenDaylight YANG: PUBSUB application**

### Setup the working directory with yang-push code

Pull the YANG PUB-SUB code for stable boron release code <https://git.opendaylight.org/gerrit/p/yang-push.git>

Set the maven repository dependencies

cp -n ~/.m2/settings.xml{,.orig} ; \

wget -q -O - https://raw.githubusercontent.com/opendaylight/odlparent/stable/boron/settings.xml > ~/.m2/settings.xml

It should have generated the **yang-push** directory with its sub-folders in the following hierarchy:

***---yang-push***

***+– api  
+– artifacts  
+– features  
+– impl  
+– it  
+– karaf  
+– pom.xml***

### Building, starting and verifying the yang-push code

Execute the following command to build the project:

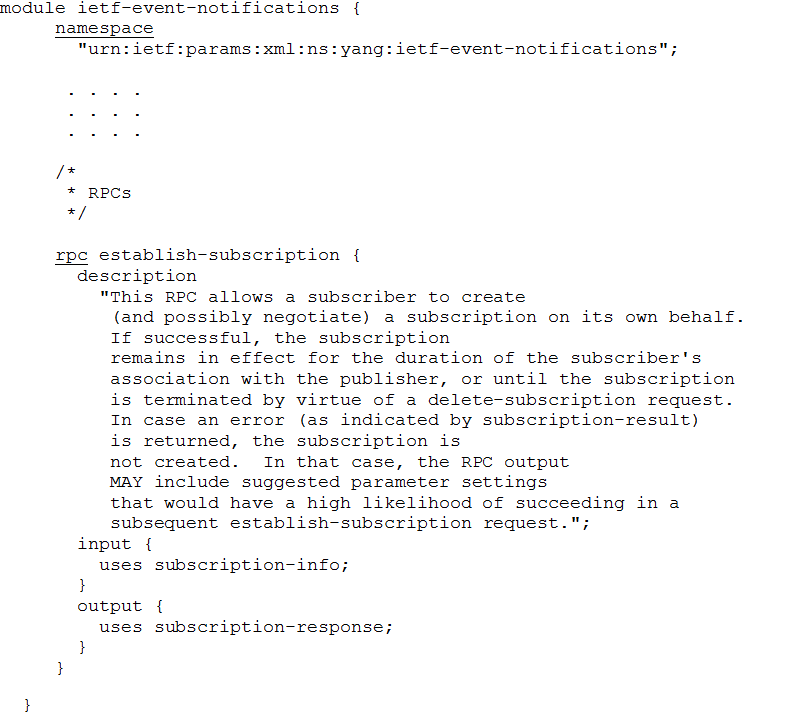
***mvn clean install -Pq -DskipTests -Denforcer.skip=true***

The run time build available at

***karaf/target/assembly/bin/karaf***

### YANG definition for RPC Establish-subscription in ietf-event-notifications.yang

As shown below, YANG defines RPC named an establish-subscription with inputs and outputs.



On re-building the yang-push, code with this YANG in ***+-api/src/main/yang*** it would generate Java Objects and interfaces corresponding to the attributes defined in YANG. Please refer to the following files, which are auto generated under the API sub-directories.

***+– IetfEventNotificationsService.java  
+– EstablishSubscriptionInput.java   
+– EstablishSubscriptionInputBuilder.java  
+– EstablishSubscriptionOutput.java  
+– EstablishSubscriptionOutputBuilder.java   
+– $YangModelBindingProvider.java  
+– $YangModuleInfoImpl.java***

Here provided the complete path for each file generated while we build the yang model.

1. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/ietf/params/xml/ns/yang/ietf/event/notifications/rev161027/IetfEventNotificationsService.java
2. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/ietf/params/xml/ns/yang/ietf/event/notifications/rev161027/EstablishSubscriptionInput.java
3. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/ietf/params/xml/ns/yang/ietf/event/notifications/rev161027/EstablishSubscriptionInputBuilder.java
4. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/ietf/params/xml/ns/yang/ietf/event/notifications/rev161027/EstablishSubscriptionOutput.java
5. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/ietf/params/xml/ns/yang/ietf/event/notifications/rev161027/EstablishSubscriptionOutputBuilder.java
6. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/ietf/params/xml/ns/yang/ietf/event/notifications/rev161027/$YangModelBindingProvider.java
7. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/ietf/params/xml/ns/yang/ietf/event/notifications/rev161027/$YangModuleInfoImpl.java

The below files generated for the opendaylight-yang-push.yang model, the path for generated files are:

1. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/Subscription2.java
2. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/PushUpdates.java
3. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/OpendaylightYangPushData.java
4. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/$YangModelBindingProvider.java
5. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/MountedDeviceInfo.java
6. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/push/updates/PushUpdate.java
7. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/ConfiguredDeviceInfo.java
8. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/push/updates/PushUpdateBuilder.java
9. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/$YangModuleInfoImpl.java
10. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/push/updates/PushUpdateKey.java
11. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/EstablishSubscriptionInput1.java
12. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/PushUpdatesBuilder.java
13. /home/odl-17/new-yangpush/yang-push/api/target/generated-sources/mdsal-binding/org/opendaylight/yang/gen/v1/urn/opendaylight/params/xml/ns/yang/opendaylight/yang/push/rev170721/Subscription1Builder.java

**Implementing RPC**

An implementation for the service interface generated by *yangtools* from the YANG definition of establish-subscription. Create the ***file YangpushRpcImpl.java***, which provides that implementation under */home/odl-17/new*-yangpush/yang-push/impl/src/main/java/org/opendaylight/yangpush/rpc/YangpushRpcImpl.java

**Methods in subcomponents**

1. **Yangpush Provider module**

* OnSessionInitiated()
* createPushUpdateDatastore()

1. **Yangpush Subscription module**

* updateSubscriptiotoMdsal()

1. **Yangpush RpcImpl module**

* invokeRpc()
* establishSubscriptionRPCHandler()
* NewcreateDeviceESRpcImpl()
* createESOutput()

1. **Yangpush Listener module**

* onNotification()
* pushUpdateHandler()
* storeToMdsal()

Below are the set of classes and methods/functions updated for the establish-subscription RPC.

1. Create the constant values for the yang models ietf-event-notifcations.yang, ietf-yang-push.yang and opendaylight-event-notifications.yang

The constant values defined in constant-values.html



1. Create Qnames for the new RPC in ietf-event-notifcations.yang, ietf-yang-push.yang and opendaylight-event-notifications.yang using the method **“QName.create()”**

public class YangpushRpcImpl implements DOMRpcImplementation { } in YangpushRpcImpl.java

QNames used to construct input arguments defined in opendaylight-event-notifications.yang

QNames = “device-subscription-id” and “device-name”

QNames used to construct input arguments defined in ietf-event-notifications.yang

QName = "stream”, “starttime”, “stoptime”,”filter” under parent yang ietf-event-notifications.yang

The list of QNames available in “filed summary” section ofYangpushRpcImpl.html



1. Declare the new RPC SchemaPath of type DOMRpcIdentifier using method **“DOMRpcIdentifier .create(schemaPath.create())”**

The declaration present in “filed detail” section of YangpushRpcImpl.html



1. Register RPC to DOMRpcProviderService using method **“registerRPCs()”** and update the method **service.registerRpcImplementation() for ESTABLISH\_SUBSCRIPTION\_RPC**

The method definition is in “method summary” sectionof YangpushRpcImpl.html



1. To invoke the RPC establish-subscription add the handler method **“establishSubscriptionRpcHandler()”**

The method documented in “method details” section of YangpushRpcImpl.html



1. In the handler function add the support to parse the input arguments and register for push-updates changes
2. Implement the method **“createESOutPut”** it creates and returns container node for Establish Subscription Output

The method documented in “method details” section of YangpushRpcImpl.html



1. Implement the method **“EsParseExternalRpcInput”** input to the method expects augment data defined opendaylight-event-notifications.yang

The method documented in “method details” section of YangpushRpcImpl.html



1. Implement the method **“NewcreateDeviceESRpcInput()”** to send the device parameters to the mounted device

The method documented in “method details” section of YangpushRpcImpl.html



1. Implement the method **“EstablishSubscriptionRpcInput()”** to store parsed input parameters for establish-subscription RPC

The method documented in EstablishSubscription.html



1. Add operation establish as enum constant to [YangpushSubscriptionEngine](file:///C:\Users\ckempapu\Desktop\IOS-XE-PubSub\Web-pages\org\opendaylight\yangpush\subscription\YangpushSubscriptionEngine.html) operations class

The declaration is in YangpushSubscriptionEngine.operations.html



1. Update the method **“updateSubscriptiontoMdSal()”** for devicesubscription-id and device name.

The method documented in “method details” section of YangpushSubscriptioEngine.html



1. Override the methods **“onNotificationComplete()”** and **“onReplayComplete()”**

The method documented in YangpushDOMNotificationListener.html



**Registering RPC**  
Add RPC registration in the file *YangPushDOMProvider.java* under under */home/odl-17/new-yangpush/yang-push/impl/src/main/java/org/opendaylight/yangpush/impl/YangPushDOMProvider.java.*

The class **YangpushDomProvider** is Binding Independent version of yangpushProvider and provide the implementation of yangpush application.

Dom provider services needed throughout the lifetime of the yangpush application and registers its RPC implementation and Data change Listener with the MD-SAL. We are reusing the instance of **YangpushRpcImpl** for new RPC establish-subscription.