[Date]

Pankaj Sitpure

Intel Security

ISC Open Stack Event Notification Infrastructure

Team ISC

Table of Contents

[ISC infrastructure: 2](#_Toc415148760)

[1. Provider: 3](#_Toc415148761)

[Consist of: 3](#_Toc415148762)

[Responsible for: 3](#_Toc415148763)

[2. Consumer: 4](#_Toc415148764)

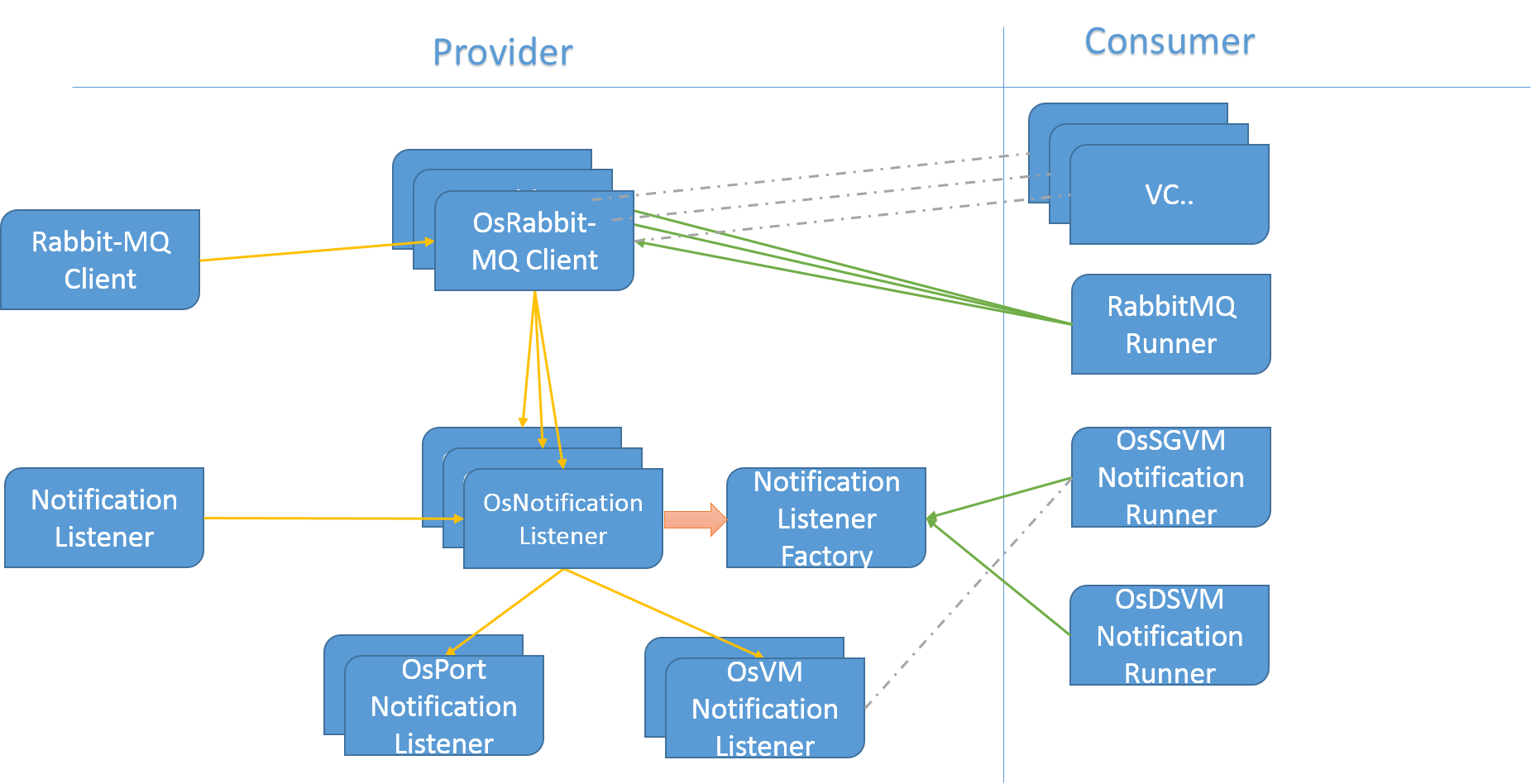
[Consist of: 4](#_Toc415148765)

[Responsible for: 4](#_Toc415148766)

[UML diagram explaining relationships between implemented classes: 5](#_Toc415148767)

# ISC infrastructure:

Diagram below explains overview of ISC Infrastructure built to receive and handle event notifications from RabbitMQ Server (Open Stack).



The architecture is divided in two categories:

## Provider:

### Consist of:

* **RabbitMQClient**:
  + Generic client capable of connecting with given RabbitMQ server.
* **OsRabbitMQClient**:
  + Extends RabbitMQClient and instantiates with Open Stack specific default attributes like: Port, User, Exchange, Routing Key, Topic etc. for every Virtualization Connector ISC have.
  + Lifecycle of this object is managed by OsRabbitMQRunner
* **Notification Listener**:
  + An Interface dictating the body of a Notification listener object
* **OsNotificationListener**:
  + An Abstract class implementing Notification listener and containing common code reused by different open stack type notifications listeners.
* **NotificationListenerFactory**:
  + Factory which will provide an object of Notifications listener based on the Object Type provided by the user.

### Responsible for:

* Connect to the RabbitMQ server for each Virtualization Connector (Type: Open Stack) user creates in ISC.
* Subscribe to queue “Notifications.info”
* Receive event Notifications From Subscribed Queue
* Filter these notifications based on the Object type (e.g. VM, Network, Host etc.)
* Delegate notifications to the registered Listeners

## Consumer:

### Consist of:

* **RabbitMQRunner**:
  + Runner class which is responsible for managing (Creating, Updating, Deleting) instances of OsRabbitMQClient based on the lifecycle of Virtualization Connector in ISC.
* **OsSGNotificationRunner**
  + Runner class which is responsible for managing (Creating, Updating, Deleting) instances of NotificationListeners based on the lifecycle of Security Group etc. in context

### Responsible for:

* Receive Notifications based on registered object type.
* Filter them based on the ObjectIdList
* If the notifications represents the object of interest
  + Parse Notification Jason and get Value for the interested key
  + Perform Post Notification operation aka Business Logic. For ex.
    - SG sync if any member is updated, added or deleted
    - DS Sync if Tenant, Host Aggregates, Host changes

# C:\Users\Pankaj\Desktop\UML.pngUML diagram explaining relationships between implemented classes: