Cross VPC RPC Framework based on NuMessage Queue

Motivation

Most of the applications in platform and infrastructure could be divided into two parts:

- public service: exposed to the end user and serve the user requests
- backend service: receives the requests from public service and does the underlying processing, returning responses.

For example, in Rheos, the public service is Rheos portal, exposed to the end user and Rheos core service is the backend service.

VPC(virtual private cloud) is widely used in eBay, for example, staging, CORP, production and sandbox are several well known VPC. Most of the time, different VPC could not talk with each other. TCOP(trusted control plane) is a special VPC, and the services located in this VPC could be called by any other VPC. But, the services in TCOP need to follow all the security requirements, like authentication and authorization, scoping etc.
NuMessage queue service is one of the messaging services located in TCOP and meets all the rules of TCOP.

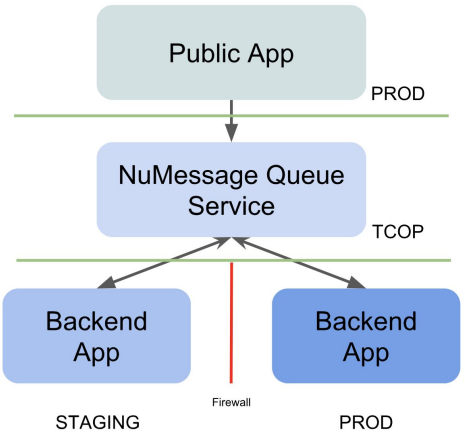
So currently, most of the applications need to deploy both public services and backend services in each VPC. Comparing to an unified public service, this kind of isolation deployment cost a lot:

- Redundancy deployment and maintenance efforts.
- Overhead on the user side. They need to remember and switch between public services.
- Manually efforts to sync between different VPC. Sometimes, different VPC need to talk with each other, then service owners have to be manually involved.

In a word, a standard pattern is needed to support unified public service. This pattern should satisfy the following two requirements:

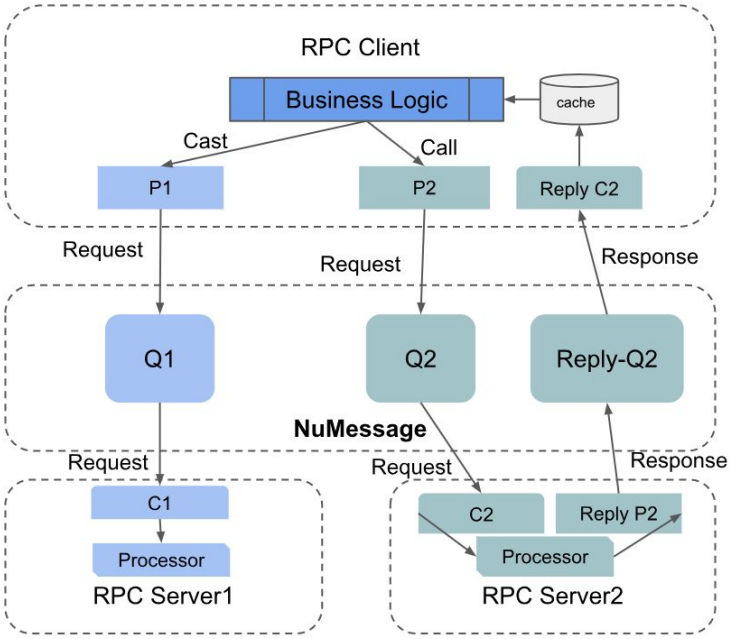
- Force the application to meet all the security requirements. This pattern should follow the process of GIS and should not bypass any security rules.
- Easy to integrate. The efforts to take this pattern should never exceed the cost to migrate service to TCOP.

Basic Idea



- Public service deploys in production. Instead of calling the backend service directly, the public service produces messages to the NuMessage queue.
- Backend services still deploy in each VPC separately. Backend service listens on the predefined queue and waits for the requests, processing, then returning the response.
- NuMessage queue service is used to decouple the public service and the backend services.
- Regard the requests from public service to backend service as RPC(remote procedure call). The public service is the RPC client and backend services are RPC server.

Design



- Two RPC semantics are considered:
    - Cast: client sends out request, but doesn't wait for response
    - Call: client waits the response from server side
- Server and client appoint to listen on the same queue when starts up.
- Different servers appoint different queues with their client. Queues could belong to different scope, i.e., different namespace mapping to NuMessage world.
- All the producing and consuming requests need to provide the namespace and corresponding token.

The procedure of "cast" semantic is as follows:

- RPC server 1 starts up and its consumer(C1) listens on Q1
- Client invokes "cast" in its business logic
- P1 sends a request to Q1
- C1 receives the request and passes to the processor to handle it.

The procedure of "call" semantic is as follows:

- PRC server 2 starts up and its consumer(C2) listens on Q2
- Client invokes "call" in its business logic
- P2 sends a request to Q2, and specifies the corresponding reply queue(Reply-Q2) in this message and also an unique message id.
- Client starts reply C2 to wait for the response on Reply-Q2.
- C2 receives the request and passes it to the processor on Server 2.
- The producer sends out a response message(Reply P2) sends a message to the predefined Reply-Q2.
- Reply C2 receives the response and puts it into the cache. The key of this cache is the original unique message id.
- Framework gets the response with a message id and returns to business logic.

Security rules

Here is the main list of rules that GIS team is care for when cross VPC:

- HTTPS
- Authentication
- Authorization
- Scoping, which means the requests and data belong to different VPC should not be mixed together. This requirement usually depends on how the application uses the service.
- The data is approved to cross VPC

As a RPC framework which targets for serving cross VPC use cases, all of these rules need to be followed by nature. For all the requests in this framework is talking to NuMessage queue service, NuMessage queue service must support these requirements.

Authentication & Authorization in NuMessage

NuMessage defines a concept called “Namespace”. Namespace holds a group of resources(called quota) and all the resources in NuMessage(Queue, for example) will belong to a namespace. Namespace in NuMessage is very like the tenant in openstack and Namespace in TESS.

Onboard a namespace is the very first step before a user starts to use NuMessage. For RPC user, the following two fields are the most important:

- Name: the name of this namespace
- Identity Type: Keystone
- Identity: tenant id

When produce/consume messages to/from NuMessage queue over HTTPS, the two fields need to be added into header:

- Namespace: the namespace that this queue belongs to
- Token: the token that belongs to that namespace, i.e., belongs to the tenant id bind to the corresponding namespace

Then,

- Authentication: check whether the token is not valid
- Authorization: check whether the token belongs to the right namespace

Note that the identity type in NuMessage is not limited to Keystone, for example, IAF.

Scoping in RPC framework

Two concepts are introduced in this RPC framework:

- Application: the name of the current application
- Scope: the scope of the corresponding server. For example, the servers deployed in different VPC should belong to different scope.

Mapping to NuMessage, each <application, scope> pair is mapped to a unique queue. And, different scope should use different namespaces.

By this design, this RPC framework meets the scoping requirements. To be more specific:

- Different scope produce/consume messages to/from different queues
- The queue for each scope, belong to different namespaces
- The services belong to each scope, could not access the queue that they don’t own

Data to cross VPC

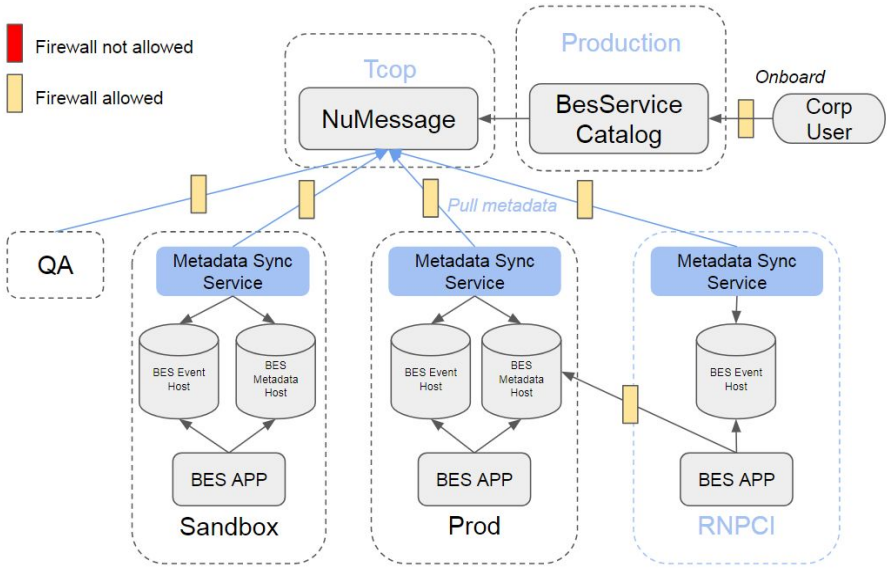
By design, this RPC framework could carry any type of data and cross between VPC. However, this is not allowed. For each use case that wants to leverage this framework, I need to do a review with GIS first. GIS will check whether the data is allowed to cross between VPC. Currently, two kind of data is quite common to cross VPC:

- The metadata need to synced between different VPC
- The commands send out from the control plane and need to be executed by the data plane agent

Use case

BES service catalog(BESSC)

BESSC will be the first use case within Rheos team. Previously, this service was deployed in the ES zone and pushed the user onboarded to each VPC. This path is no longer approved by the GIS team.
Here is the new design based on this solution:



- BESSC is the RPC server and metadata sync service is the RPC client
- Move the BESSC from ES to production
- Instead of push model, all the metadata sync service pull messages from NuMessage
- Re-implement the metadata sync api with RPC interface

Application specific details:

- RPC server:
- RPC client:
- Traffic: manually called, less than 1000 times per day