1. **What is an Application Network?**

An application network is a way to connect applications, data and devices through APIs that exposes some or all of their assets and data on the network. The network allows other consumers from other parts of the business to come in and discover and use those assets. Building an application network involves developing reusable assets and then encouraging those in the business to reuse and self-serve those assets.

An example of why an application network is necessary

This can be demonstrated with an example that is common in many enterprises — integrating Salesforce with an SAP database. It might be quite tempting to build a custom integration to tie the two applications together.

In theory, that approach is fine. It's on time, it's on budget and meets requirements. The problem with this approach is that there will not be any leverage or reuse out of the project. The applications are tightly coupled together and there is no additional value extracted from connecting Salesforce to SAP. After this has been done for the first time, if another team needs to have access to these applications, the integrations must be created again. Once a large number of these integrations are built, it becomes very hard to govern anything or get visibility into what is happening between systems.

1. **What is API Led Connectivity?**

It is a multi-layered approach that scales IT capacity through its emphasis on modular components, decentralized authority over apps development and reusable assets.

Instead of building one application with all the functionality, it uses a group of services each with own API that can be govern and monitored.

1. **Features of Any Point Platform.**

Any point platform is a unified platform that has all the components needed for entire API lifecycle to design, build, deploy and manage.

* It includes tools that help you create modern API that are productized and designed for ease of consumption using design center
* Discoverable and accessible through self-service using Exchange
* Easily managed for security, scalability and performance using management center

1. **When would you use Any Point Studio?**

After API specification in designed we need to implement the design. During implementation phase use anypoint studio. It is used to import the API specification, build the application and perform automated testing using embedded MUnit framework.

Output: You have an API that can be deployed as web service

1. **How can you build Application Networks with AnyPoint Platform?**

Once the discipline of consumption, self-service and reuse is established in the organization, a different type of landscape emerges — an application network. In this landscape, not only are there reusable assets available to be deployed throughout the organization in order to create and deliver new products more efficiently, there are enablement teams to help those development groups understand best practices on how to build new products and services as well as point the way to a repository of services available for reuse. Everything that is posted in the application network is discoverable, managed, governed and secured; the central IT organization has management and governance over all the services while allowing development teams in the lines of business to use them for whatever project is deemed necessary. People building new services and products throughout the organization can take them as is or they can take a building block, add to it and publish a new building block

1. **What are the benefits of Application Network and API-Led Connectivity?**

The benefit of API led connectivity are:

- To solve the IT capacity gap

- ensures consumption, self-service and reuse

- complete project on time and everything is reusable

- designed such a way that it can easily be changed

-Connection are secure and monitored so it is agile and productive

-Meets business requirement and also meets governance, compliance security and scalability

- The benefits are:

* Speed of delivery: through pre build APIs, connectors and templates and automated unit integration testing CICD
* actionable visibility: into nodes and backend system that each application uses to trace problem quickly and into the application n/w to new dependencies and full down stream and up stream impact changes
* secure by design: carefully curated assets for specific users/ groups
* Separate API tier provided: multiple level, isolation, governance and control for each layer
* Future proof architecture: that extensible using custom connectors which can be build with in the product and decomposable using loosely coupled architecture to isolate impact of changes
* Intentional self service: with embedded capacity to catalogue, discover and reuse api and leverage digital assets across entire organisation

1. **Why would you use AnyPoint Platform?**

Anypoint Platform is a complete solution for API-led connectivity that helps companies build application networks of apps, data, and devices, both on-premises and in the cloud.

**AnyPoint Platform** is the most advance enterprise platform for designing, developing and managing APIs and integration. The benefit of using anypoint platform are

* Uniquely built as a single product
* Deploy any where
* Wide range of use case

8. How would you build an Application Network using API led connectivity?

**9. What are APIs and Web services? What are the major differences?**

The term “**API**” stands for [Application Programming Interface](https://nordicapis.com/focus-topics/what-is-an-api/).  An API is an interface that can be used to program software that interacts with an existing application.

A **Web service** is a software system designed to support interoperable machine-to-machine interaction over a network.

Difference between API and web services are:

* web service is an API — since it exposes an application’s data and/or functionality — but not every API is a web service
* **Web services require a network.** While APIs can be on- or offline, web services must use a network.
* **APIs are protocol agnostic**. While APIs can use any protocols or design styles, web services usually use SOAP (but sometimes REST, UDDI, and XML-RPC).
* API is an interface that allows you to build on the data and functionality of another application, while a web service is a network-based resource that fulfills a specific task

**10. How can you secure your APIs?**

We can secure API in three ways. The pros and cons of using each method is given below

## Custom code using existing frameworks

## This means that developers can use existing Java frameworks and functions (such as Spring) to secure their apps by importing respective libraries.

## Adv:

## The upside of this approach is that developers could write custom functions that fit the security requirements 100%

## Disadv:

## the major challenge with the custom code approach is that these libraries will have to be embedded into all deployed Mule application that need them (probably all of them). So once security requirements or endpoints change over time, the security libraries will have to be maintained and updated — and all of the Mule applications will likely have to be redeployed as well which could result in a disruption to the business over time.

### **External third party or cloud services**

**Pros:**

* this approach gives companies flexibility to pick the best tools for the job
* potentially be cost effective

**Cons**:

creates technical debt that will be hard to untangle later, and that will further complicate the task of quickly creating business functions and exposing them as APIs. Also, companies create a dependency on third party solutions that may change over time — or even worse, may not be available suddenly which would unnecessarily expose the APIs

### **Anypoint API Manager**

* It basically creates an [API Proxy](https://blogs.mulesoft.com/dev/api-dev/proxying-with-api-manager/) for each backend API running on Anypoint Platform and thereby secures requests coming into the platform again the API.
* Developers can quickly attach API policies to the endpoints and secure them efficiently without changing the underlying code being dependent on external solutions
* its components are tightly integrated with Anypoint Platform, so will not require additional thinking about firewalls or tunnels. The policies can also be easily applied or removed from APIs with no custom coding and no redeployments involved.

# [**5 ways to ensure data and API security**](https://blogs.mulesoft.com/biz/api/ensure-data-and-api-security/)**:**

<https://blogs.mulesoft.com/biz/api/ensure-data-and-api-security/>

1. Multi factor Authetication
2. Token based
3. Digital Signature
4. Public-key cryptography
5. Digital certificates

https://www.slideshare.net/mulesoft/best-practices-for-api-security

## https://apisero.com/cache-scope-and-object-store-in-mule-4/

# PGP Encryption

## PGP: Pretty Good Privacy

## It provides data encryption as well as data compression. It first encrypt the data then it compress the data and send the data over the network to reduce the n/w latency. Thus reduces the size of the payload by compressing data and then send over the n/w to reduce the transmission time

## It reduces the size of payload while transferring data over network

## Use case:

## It is used when transferring file from one location to another location.

## It is used when transferring the email securely and faster

## JCE Cryptography Vs PGP

## JCE does not compress the data. It only encrypts/ decrypts the data and send/receive over the n/w

## PGP not only compress the data. It also encrypts/ decrypts the data and send/receive over the n/w

## PGP is slow compared to JCE because of compression and encryption. But it is faster to transfer the data over the n/w

# Generate gpg keys

## We want to generate keys in gpg format

## gpg –gen-key

## 

## 

## Pass phase: mulesoft123

## List all the keys available

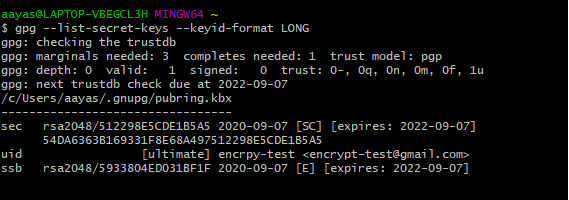
## gpg --list-secret-keys --keyid-format LONG

sec rsa2048/512298E5CDE1B5A5 2020-09-07 [SC] [expires: 2022-09-07]

54DA6363B169331F8E68A497512298E5CDE1B5A5

uid [ultimate] encrpy-test [encrypt-test@gmail.com](mailto:encrypt-test@gmail.com)

ssb rsa2048/5933804ED031BF1F 2020-09-07 [E] [expires: 2022-09-07]



## Generate a public key

gpg --output <where you want to generate the public key> --export <where you want to export public key >

Example: gpg --output C:\\Certificates\\mule\_pub.gpg --export [encrypt-test@gmail.com](mailto:encrypt-test@gmail.com)

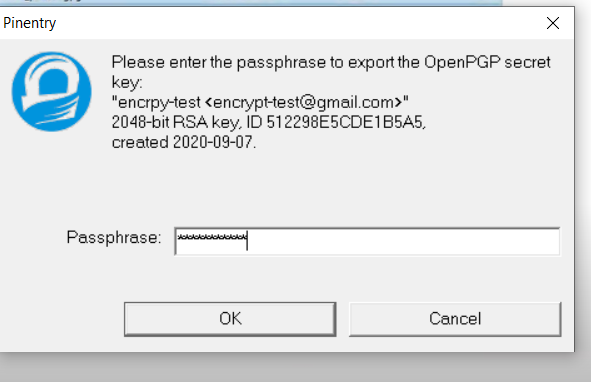
Check the C:\\Certificates\\mule\_pub.gpg you will find the public key

## Generate a private key

gpg --export-secret-keys <secret key id> > <where you want to export private key >

Example: gpg --export-secret-keys 54DA6363B169331F8E68A497512298E5CDE1B5A5 > C:\\Certificates\\mule\_private.gpg

Enter the same passphrase



## PGP encrypt configuration

## 

## PGP decrypt configuration

## 

Things to do:

<https://www.youtube.com/watch?v=fVASrNQFIaQ>

1. Object store connector

<https://docs.mulesoft.com/object-store-connector/1.1/>

<https://docs.mulesoft.com/mule-runtime/4.3/mule-object-stores>

https://docs.mulesoft.com/object-store-connector/1.1/object-store-to-define-a-new-os  
----------------------

An object store is a facility for storing objects in or across Mule applications. Mule runtime engine (Mule) uses object stores to persist data for eventual retrieval. Internally, Mule uses object stores in various filters, routers, and other message processors that need to store states between messages

* 1. How to use it?

To use it the mule pallet has the object store in its core component. We need to drag the module that we want to perform in the canvas. Then configure the object store.

If configuration is not done then it will use the default object store configuration.

By default, each Mule application has an object store that is persistent and is always available to the app. You can also create custom object stores, which can be persistent or in-memory.

You can configure an object store as a global element, to be accessible for any component in your app, or configure a custom object store for a specific component that supports its usage.

* 1. When should we use OS?

It is used for following cases:

* To store and synchronize information
* To store temporal information’
* To store user information
  1. Use cases of using OS?

You can configure a custom object store to use in the following scenarios:

* When configuring an Idempotent Message Validator, to store the unique IDs in the specified object store.
* When configuring a Redelivery policy in a listener (HTTP, File, JMS, VM, Socket), to store the redelivery count in the specified object store.
* When configuring a Cache scope that uses a custom caching strategy, to store the cached data in the specified object store.
* When a custom component uses an object store to persist information.
* When storing or retrieving information from a Mule flow, by using Anypoint Connector for Object Store (Object Store Connector).
  1. Other uses of OS?

Other use cases for object store are:

* OAuth returns the token during an authorization request and this token needs to send with every request and it has expiration period (e.g. 2 hours). You can store the token in an object store so another flow or Mule application can retrieve that token from the object store to send requests to the client. Every two hours, a new token is generated and can be stored in the object store.
* It can be used when configuring an idempotent filter or until successful scope.
* It can be used when configuring a custom component that must use an object store to persist information.
* It can be used when storing or retrieving information from a Mule flow through the Object Store module.
  1. Limitations of OS?

Object Stores are not a universal solution for data storage. They do not replace a database, and they are not suitable for every use case. Most importantly, they do not support transactional access or modification. For use cases in which [ACID](https://en.wikipedia.org/wiki/ACID) semantics are needed, or for cases where you expect the same key to be updated in parallel, consider another solution

* 1. Does OS is persistent or transient?

OS allows to store data in persistent as well as transient storage. The use of storage type depends on the user requirement

* 1. What is persistent or transient?  
     **Persistent OS**: Persistent means storing data in a reliable place such as hard disk. So when the system get shutdown you don’t loose the data. It will be slow compared to transient storage

**Transient OS**: Transient means storing data in memory. The scope of the values stored in the object store is till the program executes.

* 1. Where does the object store values stored in Mule?

Depending upon the settings of the connector and where you deploy and how you deploy determines where the data is stored in object store.

* If you are using mule connector and you deploying the application on premise runtime it will leverage the mule object store
* If you are deploying application in the cloudhub and you are using Mule 3, if you uncheck anypoint object store v2 it will use the version v1 of object store
* If you are using Mule 4 or check the object store v2 option it store to default v2  
  1. Why cant we use database to store values instead of OS?  
     Database is used to store huge amount of data for future use while we use object store to store small amount of data for the better performance of the application.
  2. What is caching?  
     Caching is the process of storing data in a cache.

A cache is a temporary storage area

* 1. Difference between object-store and cache scope?  
     **Object Store** is a mule component that stores data in key value pair. It lets applications store data and states across batch processes, Mule components, and applications, from within an application. If used on cloud hub, the object store is shared between applications deployed on Cluster

The Cache scope is used for storing and reusing frequently called data. You can use a Cache scope to reduce the processing load on the Mule instance and to increase the speed of message processing within a flow.

* 1. When you will make Object-store transient?

Reasons for creating customized object store:

* You want to partition your information by storing them in different stores
* You want to use advanced ObjectStore features such as:
  + Transient/Persistent storage
  + Specify a time to live (TTL)
  + Specify a max capacity
* You want to keep different components from sharing state by feeding them with different stores
* You want different components to share information by feeding them with the same store

You can define an object store globally in the application so it can be referenced by name and shared between multiple components.

https://blogs.mulesoft.com/dev/anypoint-platform-dev/new-mule-4-objectstore-connector/

Aggregator pattern.  
---------------------  
**What is a design pattern?**

Design Patterns are reusable solution to a commonly occurring problem within a given context in software design. These are tested, proven and well developed paradigms that can help to speed up the development process

**Why do we need to follow design patterns?**

Design patterns are solutions generated by the experts for the recurring problems. When we follow design patterns, it reduces the effort required to build a solution

**What is enterprise integration patterns?**

Enterprise Integration Patterns are accepted solutions to recurring problems within a given context. The patterns provide a framework for designing, building messaging and integration systems, as well as a common language for teams to use when architecting solutions.

**What is aggregator pattern?**

The *Aggregator* is a special *Filter* that receives a stream of messages and identifies messages that are correlated. Once a complete set of messages has been received, *Aggregator* collects information from each correlated message and publishes a single, aggregated message to the output channel for further processing.

https://www.enterpriseintegrationpatterns.com/patterns/messaging/Aggregator.html

**How it is used in Mule 4.**

<https://docs.mulesoft.com/aggregators-module/1.0/aggregators-module-reference>

An Aggregator is a component that receives some data, processes it to extract some value and then adds that value to a list of aggregated elements. After that, and depending on the configuration, the list of elements is sent to a set of components.

When an aggregator releases the stored values, the list of aggregated elements is processed either by a route within the aggregator itself or in another Mule Flow, through an aggregator listener. All of that depends on the type and configuration of the aggregator being used.

couple of common parameters used in the aggregator are:

* Content: The expression that defines what to aggregate
* Object store: All information related to an aggregator is stored in an Object Store

**Aggregator types**

They are size-based, time-based, and group-based aggregators.

1. Size based: aggregates elements until a pre-defined size is reached, executing the routes and listeners.
2. Time based: Aggregates elements until a time period is completed, executing the routes and listeners
3. Group based: Aggregates elements in different groups according to a group ID.

Does scatter-gather uses Aggregator Module internally?

<https://help.mulesoft.com/s/question/0D52T00004mXT8MSAW/aggregation-in-scattergather>

Scatter Gather will aggregate using default aggregation strategy. For example, you have a message which is sent to multiple legs (flows) of Scatter Gather. When these flows complete, the output Mule Message of each flow is aggregated by default aggregator and you get a Collection of Mule Messages.

If you want to override default aggregator, you can use a Custom aggregator and aggregate based on your business condition.

You can set a timeout to a scatter gather leg (flow), and default aggregator ignores the message from that particular scatter gather leg.

Scatter gather

https://www.tutorialsatoz.com/mulesoft-tutorial-scatter-gather/

Rate Limiting policy.  
**What is Rate limiting policy?**

A Rate Limiting policy limits the number of requests an API accepts within a window of time. The API rejects requests that exceed the limit

**What is use-case of applying rate limiting policy?**

**Use case 1:** To market the application you might want to allow user to access a free version of API. In this you might want to limit the number of times user can request for the service.

**Use case 2:** For security such as DNS attack you can allow users to request for a service for limited number of time for a given time interval

**How to apply the rate limiting policy?**

Steps to implement rate limiting policy

* Select the policy in the left menu in API manager
* Select add policy
* From the list of out-of-box select rate limiting policy
* A dialogue bocx appears
* Add # request, time period, and time unit
* Select header if you want to show it to clients
* Then save

Difference between throttling and spike control

Munit  
**What is Munit?**

* MUnit is a framework provided by the mulesoft. This enables programmers to write automated test cases for APIs and integrations. It consist of three section:
  + Behavior: this create pre-requisites like data or required data. What you want to mock
  + Execution: where actual execution occur. Logic are added, what you want to execute, set payload .Used to manipulate
  + Validation: you can add sessions. Used to validate the output, error. Done using assert and verify call. Verify call ensures that you have requested connector eg db connector

**Why do any project needs to be unit tested?**

Unit Testing is used to design robust software components that help maintain code and eliminate the issues in code units. We all know the importance of finding and fixing defects in the early stage of the software development cycle. This testing serves the same purpose. If we set this as a standard process, many defects would be caught in the early development cycle, saving much testing time.

**What are the advantages of unit testing?**

### **Benefits Of Unit Testing**

1. **The process becomes agile:** For adding new functions or features to the existing software we need to make changes to the old code. But changing things to the already tested code can be risky as well as costly.
2. **Code quality improves:** The quality of code is automatically improved when unit testing is done. The bugs identified during this testing are fixed before it is sent for the integration testing phase. Result in robust design and development as developers write test cases by understanding the specifications first.
3. **Detects bugs early:** As developers run unit tests, they detect bugs early in the software development life cycle and resolves them. This includes flaws or missing parts in the specification as well as bugs in the programmer’s implementation.
4. **Easier changes and simplified integrations:** Doing unit testing makes it easy for the developer to restructure the code, make changes, and maintain the code. It also makes testing the code after integration much easier. Fixing an issue in Unit Testing can fix many other issues occurring in later development and testing stages
5. **Documentation availability:** Developers who are looking into the functionality at a later stage can refer to the unit testing documentation and can easily find the unit test interface and correct or work fast and easily.
6. **Easy debugging process:** It helps in simplifying the debugging process. If the test fails at any stage the code needs to be debugged or else the process can be continued without any obstacles.
7. **Lower cost:** When bugs are detected and resolved during unit testing, cost and development time is reduced. Without this testing, if the same bugs are detected at a later stage after the code integration, it becomes more difficult to trace and resolve, making it more costly and increasing development time.
8. **Code completeness can be demonstrated using unit tests:** This is more useful in the agile process. Testers don't get the functional builds to test until integration is completed. Code completion cannot be justified by showing that you have written and checked in the code. But running Unit tests can demonstrate code completeness.
9. **Saves development time:** Code completion may take more time but due to fewer bugs in the System and Acceptance testing, overall development time can be saved.
10. [Code coverage](https://www.softwaretestinghelp.com/test-coverage/) can be measured