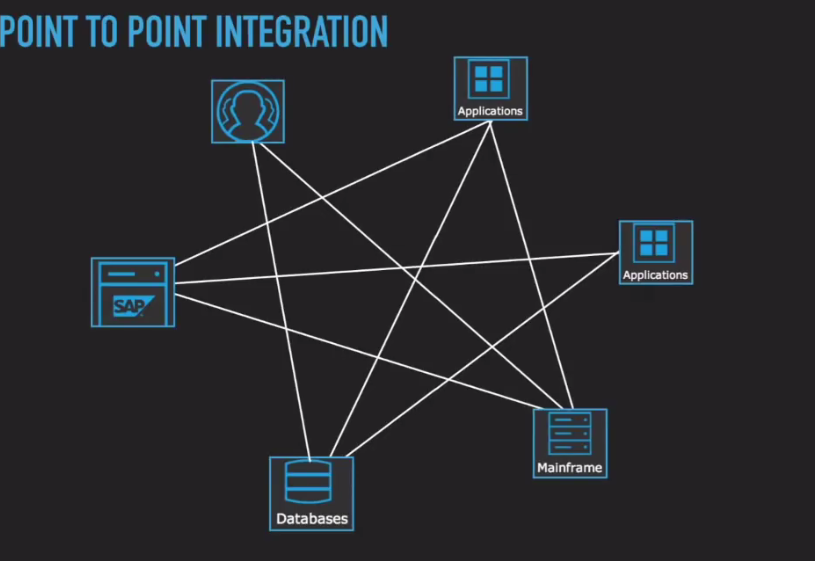
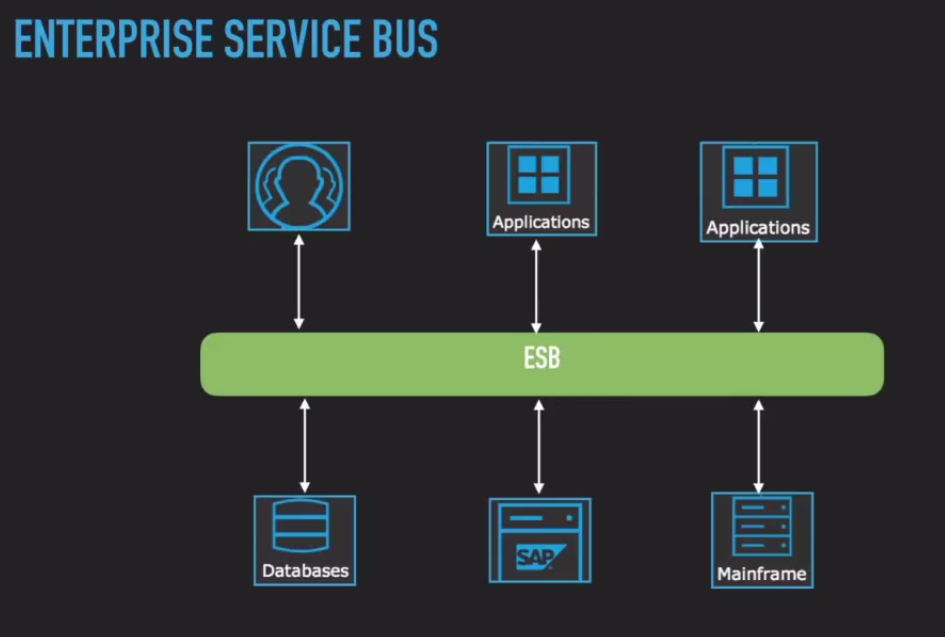
**Mule Soft**

ESB: Point to point integration - > Application connects b/w each other (Connection. Eg- Connection web application to Database)

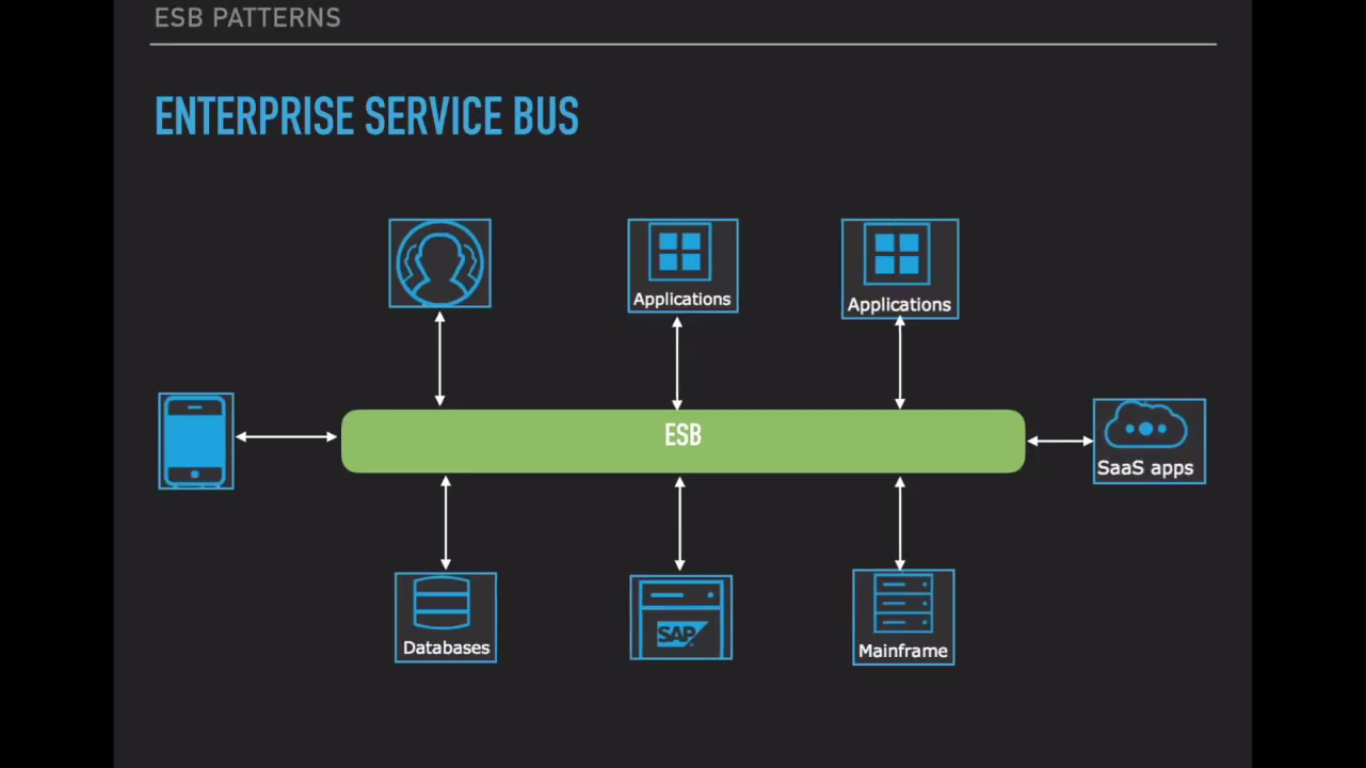
If number of application grows day by day and if they are interrelated to eachother, then there will be more complexity.



* If there is change in one of application, it will affect all other application which are inter-related.
* Delay in development
* Communication b/w many providers
* Solution is ESB



* Here we have a middle layer which will act between different application. It will take all the information of application, which wants to connect which application, thus making application loosely coupled.
* In future if we need to add new application, we can configure it through the ESB.



* ESB is software architecture for middleware.
* Supports SOA. {Service oriented architecture. Build application using design which supports robust, agile etc}
* Hide complexity {Abstraction: Java application can interact with .net without any knowledge. ESB gets msg and send to respective application}
* Simplify access
* Canonical pattern support{ General design pattern for data}
* Integration driven by business requirements not by available technology.{ Any kind of technology. So mainly it depends on requirements}

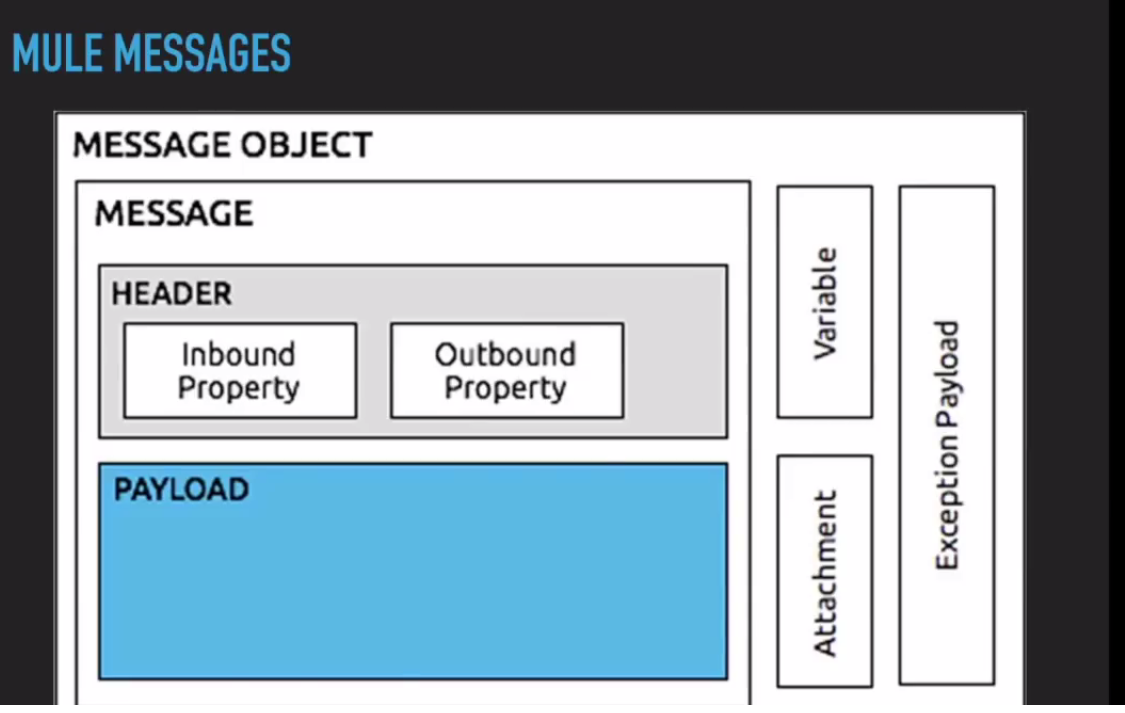
Feature of ESB:

* Routing: Should able to route message from one source to other.
* Message transformation: Conversion of messages(eg: xml to json)
* Message enhancement: Enhance the messages( Eg: Destination might be excpecting additional things).
* Protocol transformation: Transformation of protocols (Source might be HTTP but destination might be excepting FTP. So transforming to respective protocols)
* Service mapping : Transformation of different services
* Message processing : Transform the message using different message processor(Application running in different processor management)
* Process choreography: Choreograph of our messages (process the messages like to do in sequence. Eg:- Run operation 1 in begining, operation 2 in second etc )
* Service orchestration
* Transaction Management
* Security: Security management . Stateful transaction requires security. Message security (Message spoofing is one of message hacking). HTTP should support HTTPS

FTP should support SFTP

* All ESB should support Enterprise design pattern

Mule message:



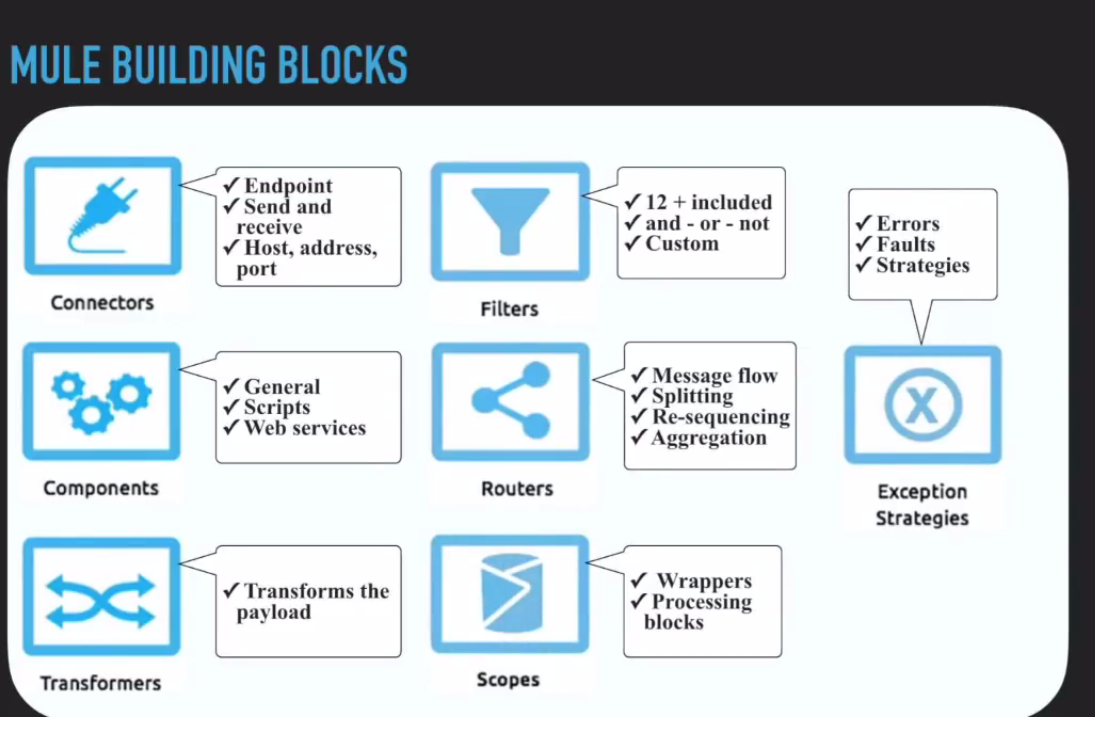
* In any ESB message is core or base of anything.
* Mule message is message object. It contains below components.
* ESB receives the message, process the message and sends the message
* Header : Meta-data of any message

Inbound property: Info about receiving of message

Outbound property: Info about sending the message

* Payload: The actual message/data.
* These three are important component. Other will use when ever required.
* Variables: Send variables along with message during message processing.
* Attachment: Form based messaging will contain all data as attachment.
* Exception payload: If there is any exception in messaging we get exception payload.

**Mule building blocks:**



Connectors: Endpoint. It will send and receive the messages. It host, address and port details. Eg: Http, FTP, file connector etc

Components: General components. Script, WebService. We can use any of technology as component

Transformers: Transform the payload from one form to other.

Filters: Filter the messages. We can filter the message using some filters.

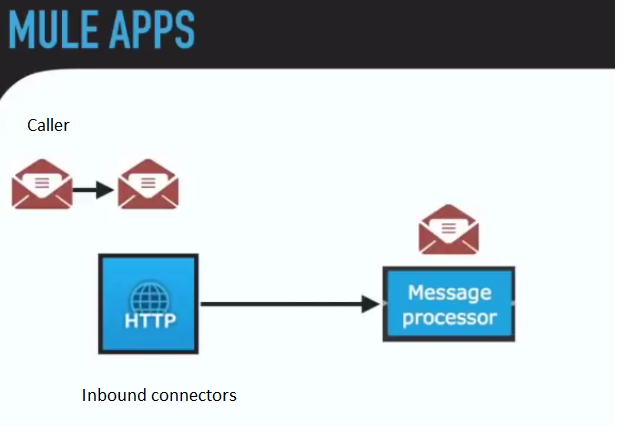
Routers: Controls the flow. Splitting, resequencing, aggregation etc can be done using routers.

Scopes: Wrappers for any processing. We can scope for transformers, components etc.

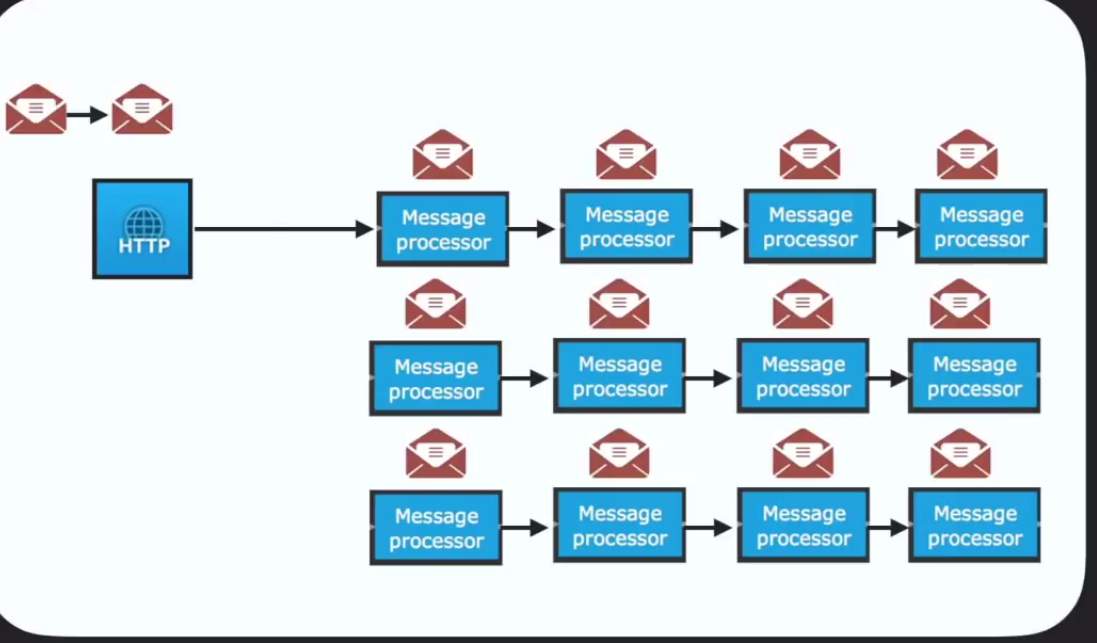
Exception Strategies: Error handling, Exception handling.

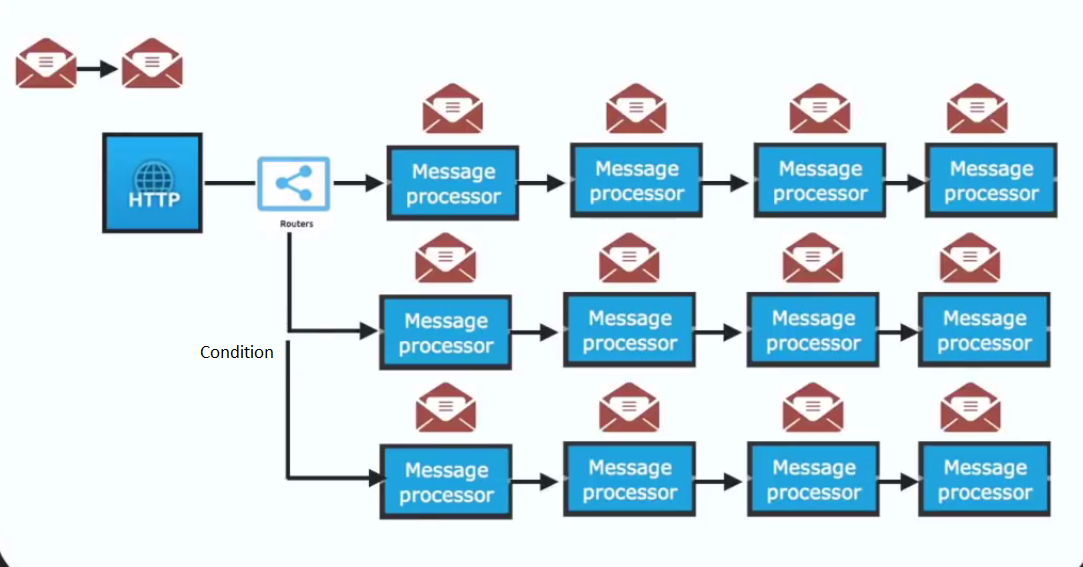
What is inside a Mule App?

* Inbound connector (http) where caller will connect. (Send the message {request} to inbound {like WebService} and get backs the message. )

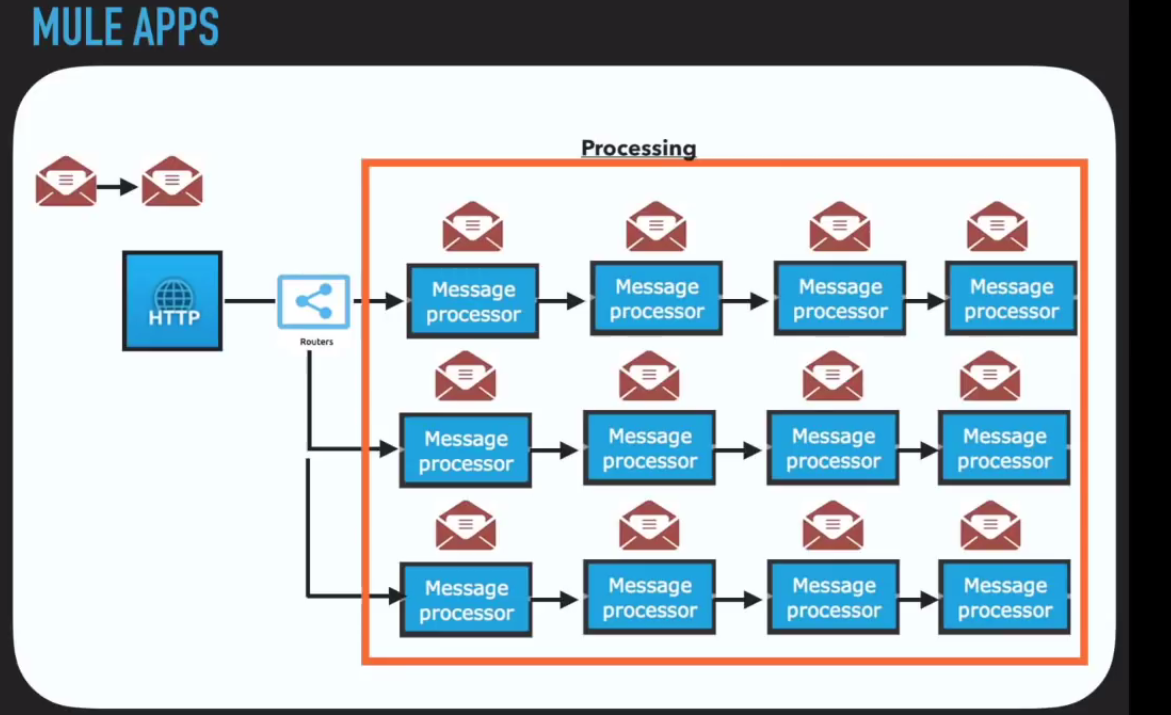


* If many message processor (Message flows), then we can use routers to identify the required message processor.

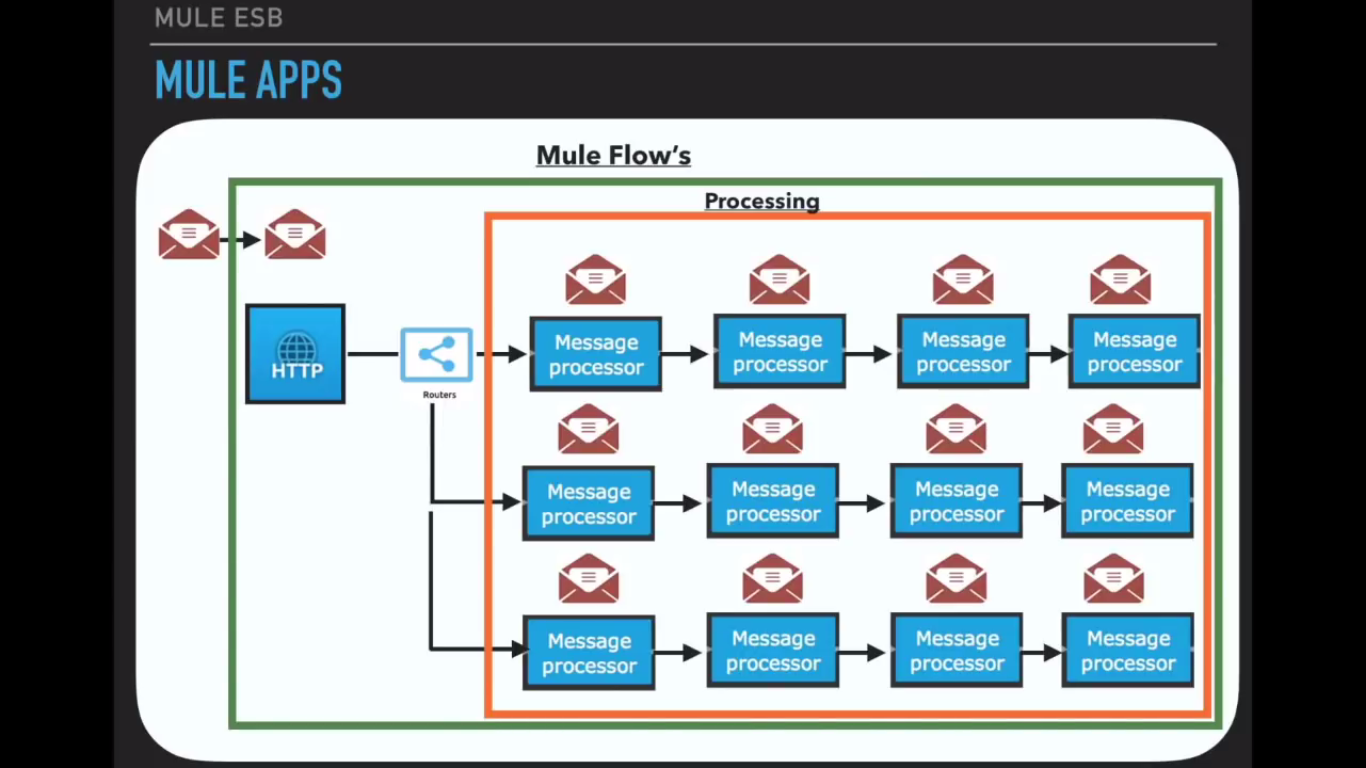




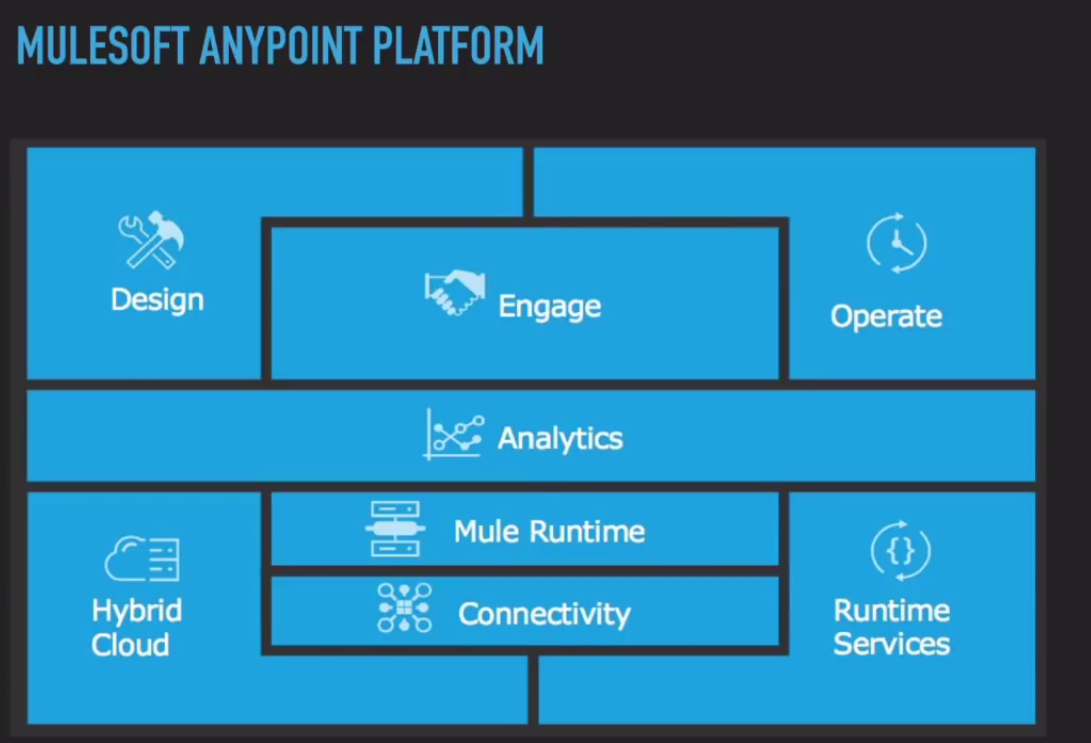
* Below whole is processing of message.

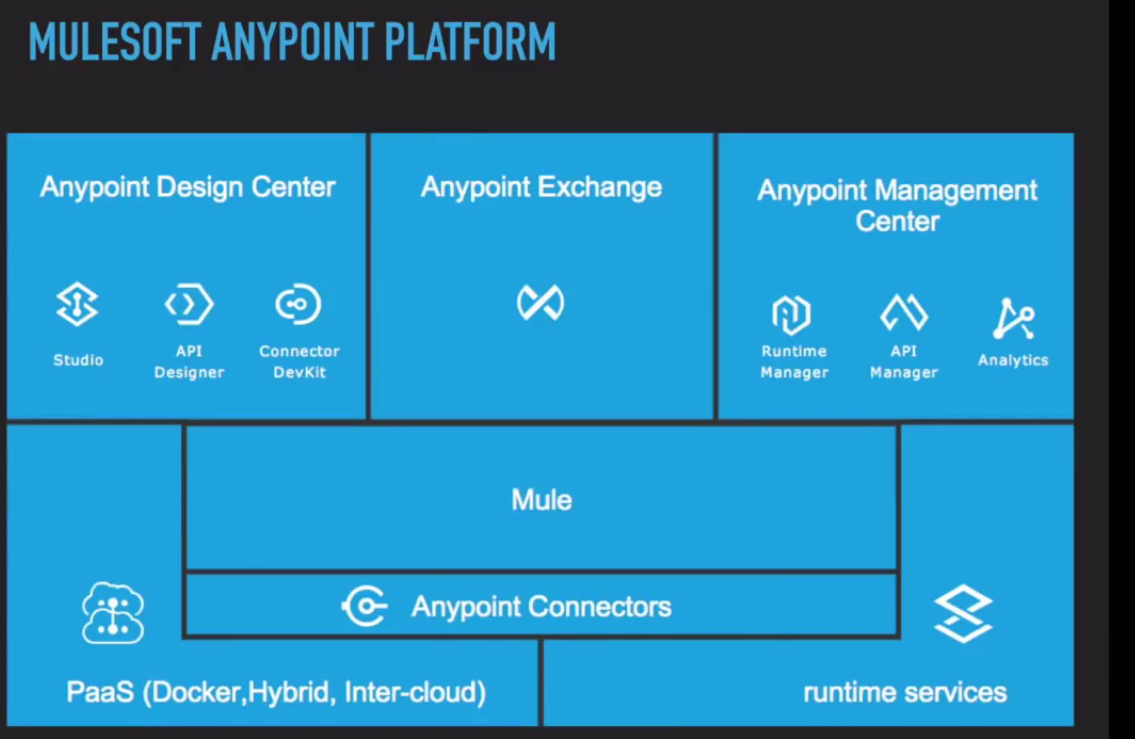


* Below complete is mule flow

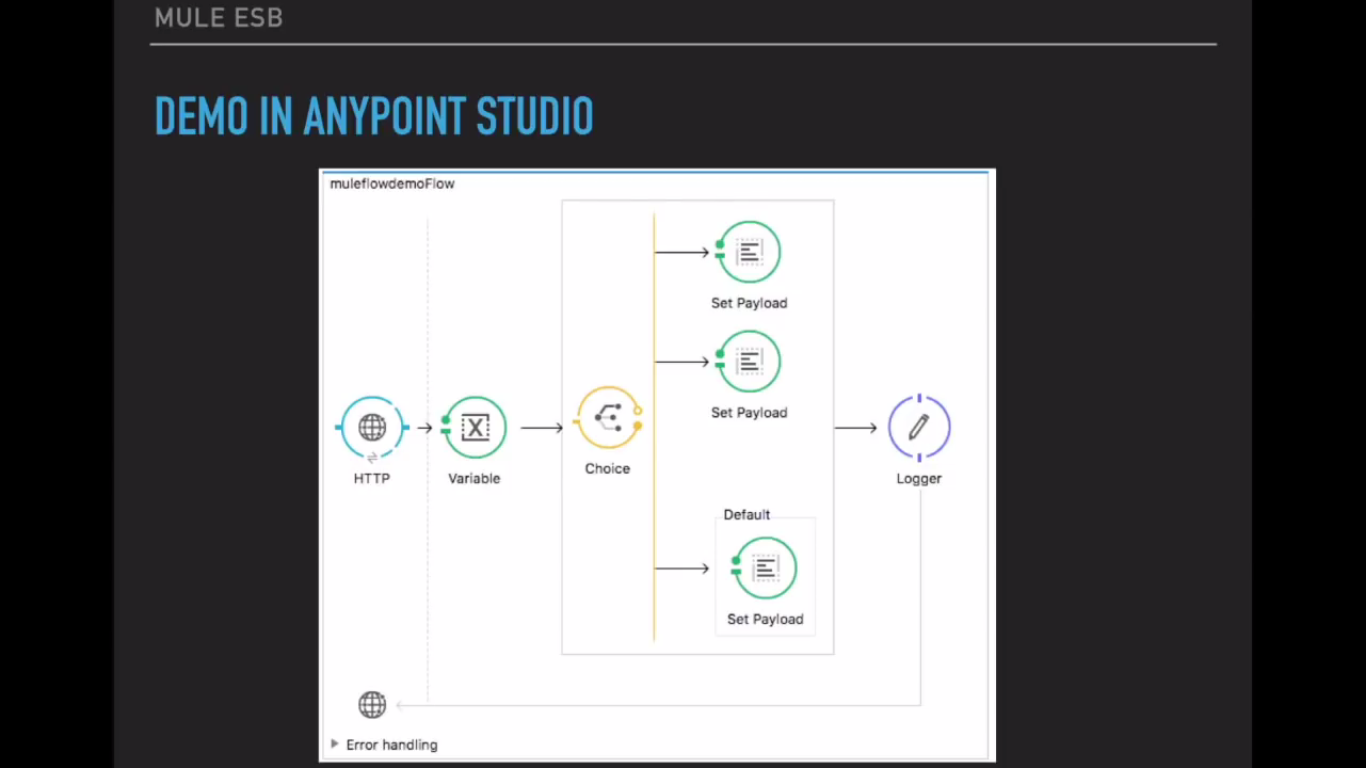


Mule has provided MuleSoft Anypoint platform. This will provide all features of mule soft. We can use as per our requirement.





Mule demo:



* We are creating mule demo using simple Http endpoint inbound connector.
* We will be passing a message from the caller(Browser)
* Based on above we will set the variable.
* Choice is nothing but the router.
* Based on choice. We will set the payload. Based on the payload we are giving back the message.
* Logger is used for logging
* Error handling is used for error and exception handling