Request-Reply Message Exchange Pattern

CAAS
Civil Aviation Authority of Singapore

What is Request-Reply

- **A method for systems to communicate with one another over the network.**
- **At its simplest, a requestor will ask for something via a *request*. It could be a data request, or a request to start something.**
- **The replier will process the *request* and return a message in *reply*.**

What is Request-Reply

CAAS
Civil Aviation Authority of Singapore

- Two Types of Request-Reply
  - Synchronous
    - SOAP HTTP
    - RESTful HTTP
  - Asynchronous
    - AMQP
    - gRPC
    - WebSocket
    - MQTT

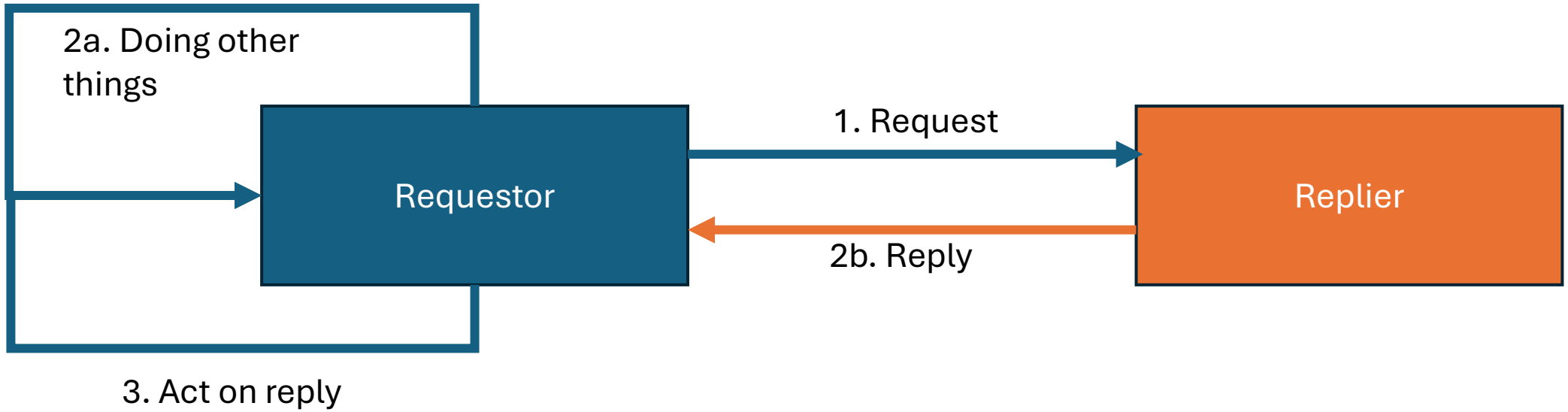
What is Synchronous Request-Reply

1. Ask (Request)
2. Processing is blocked
   a. Wait for reply
3. Act on reply (response) data

Requestor stalls until a reply is received

What is Asynchronous Request-Reply

1. Ask (Request)
2. Processing is unblocked
   a. Do other things whilst waiting
   b. Reply comes in
3. Act when reply (response) data is in



Requestor does not stall until a reply is received

What is Request-Reply

- From ICAO Doc 10203 – SWIM Implementation Guidance

5.3.3.4.2.1 Several types of MEPs are expected to be supported within a SWIM environment, including synchronous request/reply, asynchronous request/reply, one-way ("fire-and-forget") and publish/subscribe. The MEP used in any given exchange is directed by the information service provider to meet information service objectives. These MEPs include:

a) **synchronous request/reply:** The consumer initiates a request to an information service; the service processes the request and generate a reply to the consumer. The consumer waits for the information service to provide a response. During this waiting period, the consumer cannot send or receive any other requests or responses. This pattern is specifically applicable to information services that can quickly execute and respond to consumer request;

b) **asynchronous request/reply:** The consumer initiates a request to an information service; the service processes the request and generates a reply to the consumer. However, the consumer is not restricted from performing other operations while waiting for the information service's response. This MEP requires that the consumer be able to receive messages at any time and correlate them with prior requests;

Difference between Sync and Async

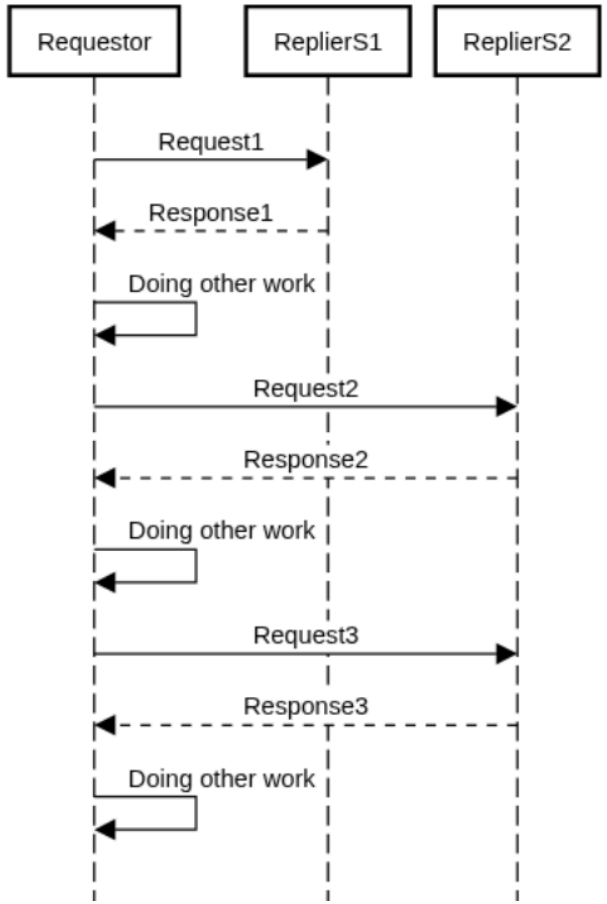
	Synchronous	Asynchronous
Time Coupling	Both requestor and replier available at the same time.	Usually a buffer to “store” messages that are slow or unable to process.
Space Coupling	Requestor needs to know where its sending. Its protocol, address, and API e.g https://test.com/api1	Usually sends to a topic or a queue. Multiple repliers possible Only need to know broker protocol, address. No need to know replier information.
Reliability Handling	Retries happen at requestor side.	Redelivery and retries usually happen at broker

Difference between Sync and Async

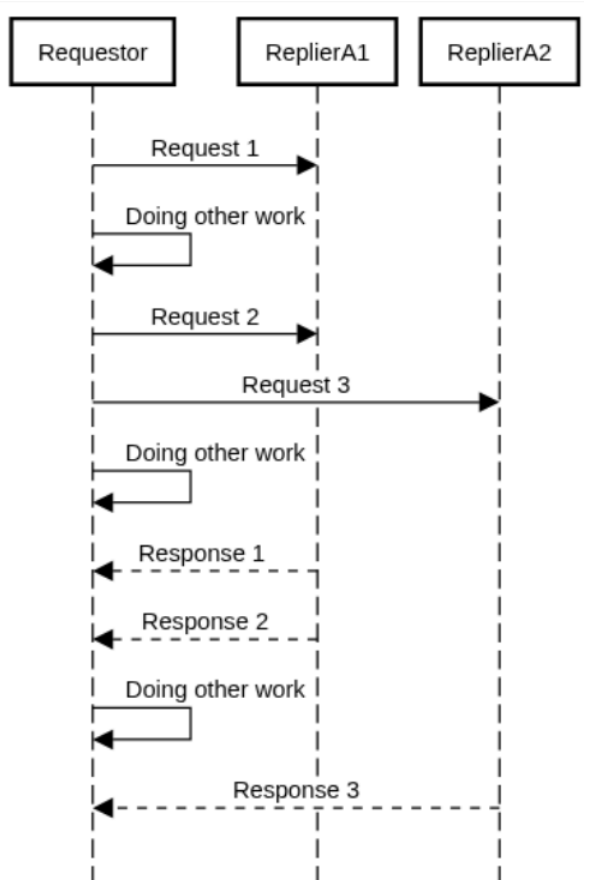
	Synchronous	Asynchronous
Use Case	• Low latency expected • Both parties are available • Interaction needs an immediate response	• Processing of request takes time. • No immediate processing required
Typical Scenarios	• User Authentication • User Interface Interactions • Database Read and Immediate Write	• Order Fulfillment • Email and notification system • Image or video processing • Database Synchronization

Difference between Sync and Async

- Synchronous Message Flow

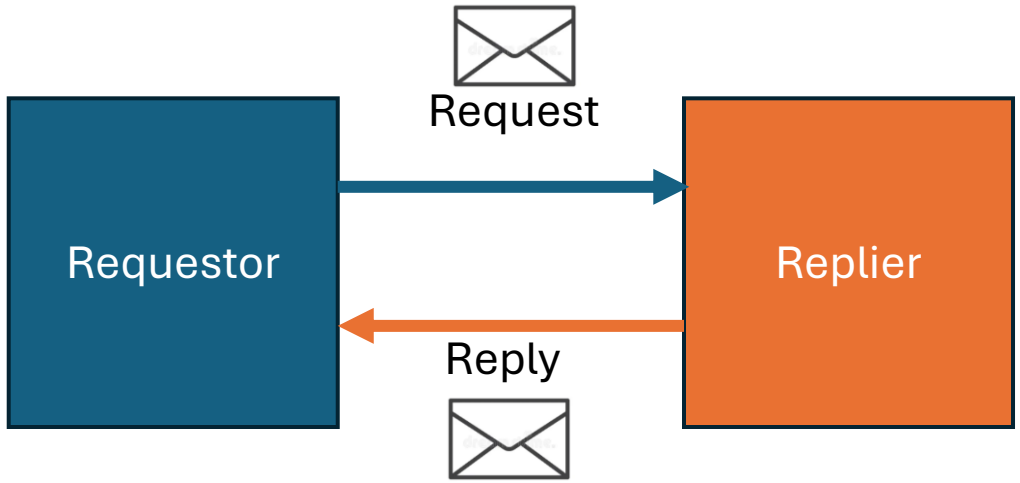


- Asynchronous Message Flow

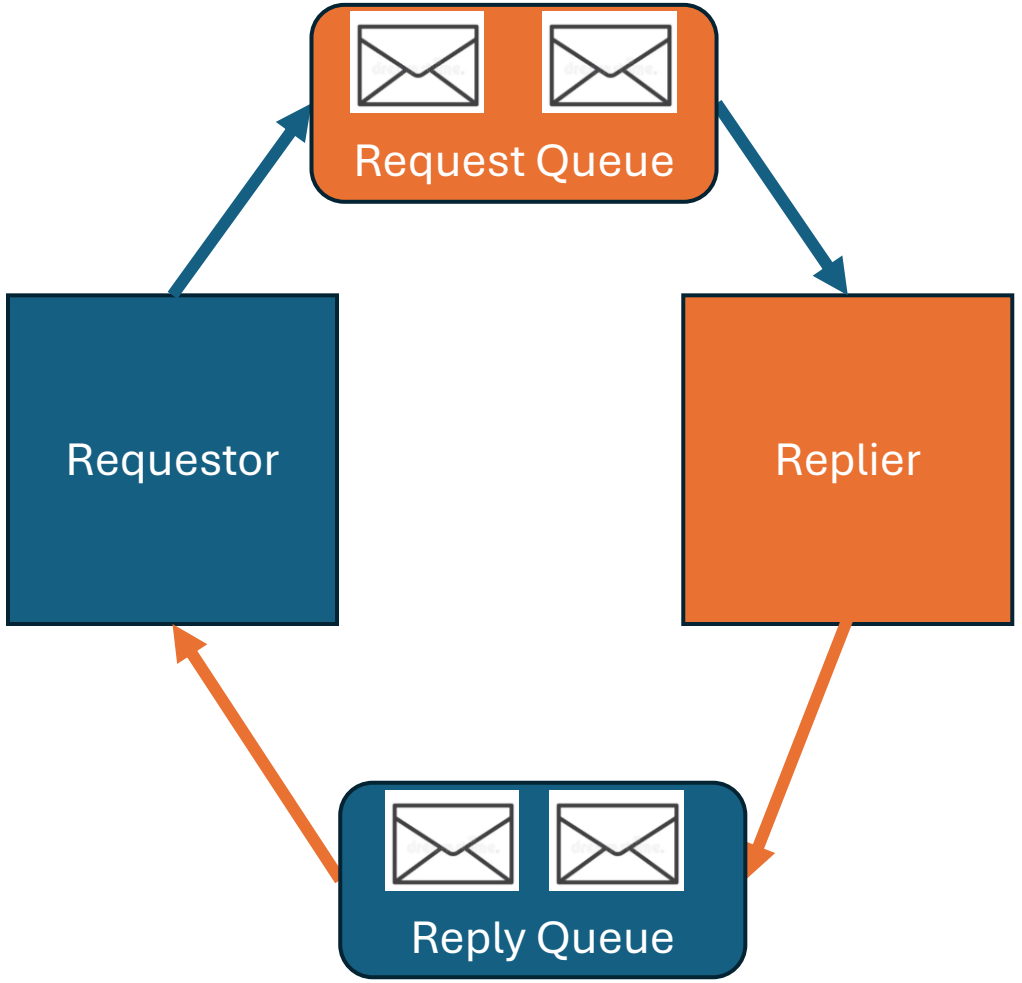


Difference between Sync and Async

- Synchronous Message Flow



- Asynchronous Message Flow



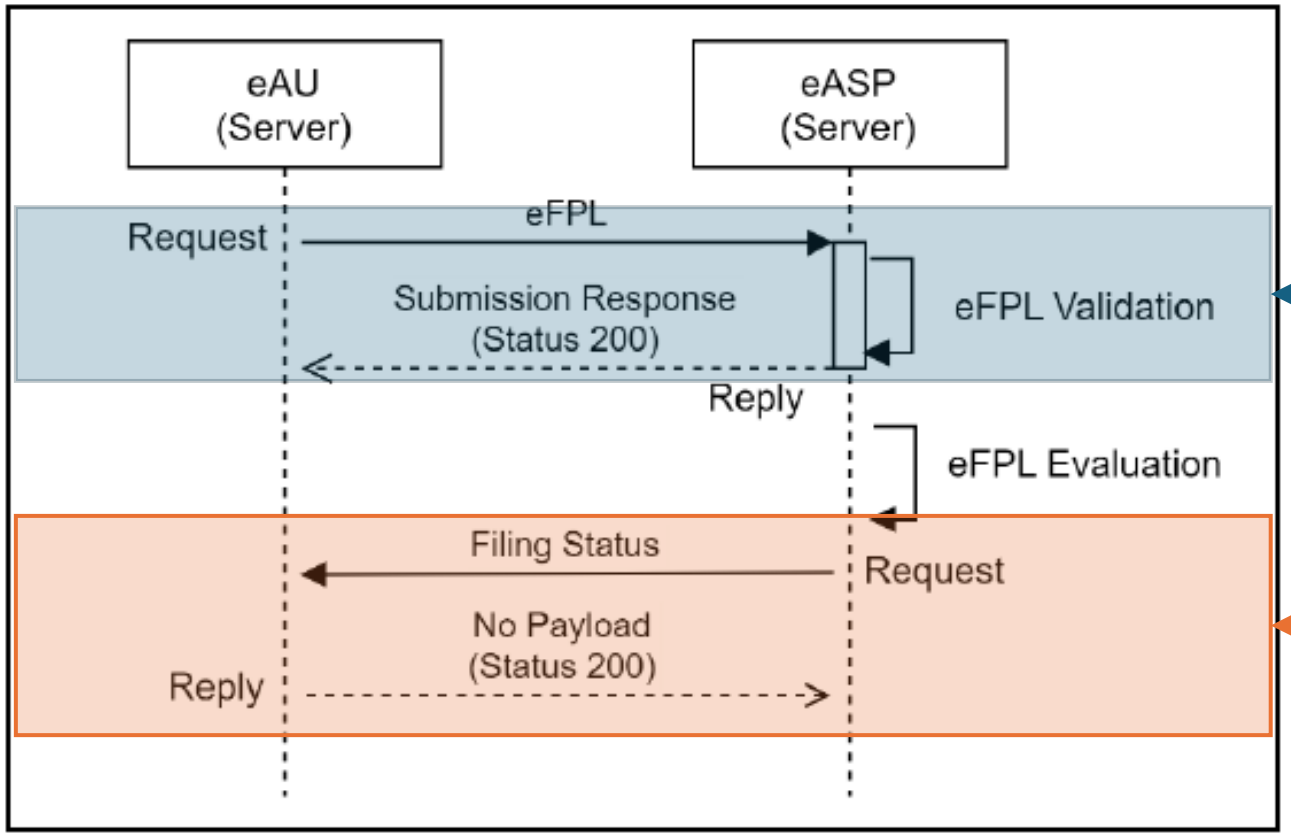
When to use Sync or Async

Civil Aviation Authority of Singapore

- Synchronous Request-Reply
  - Strong consistency
  - Deterministic Outcomes
  - Fast feedback
- Asynchronous Request-Reply
  - Throughput over latency
  - Processing tasks long-running or resource-intensive
  - Systems need to continue operation under partial failure
  - Retry logic and durability are essential

Where's the possible confusion?

FF-ICE Filing Service



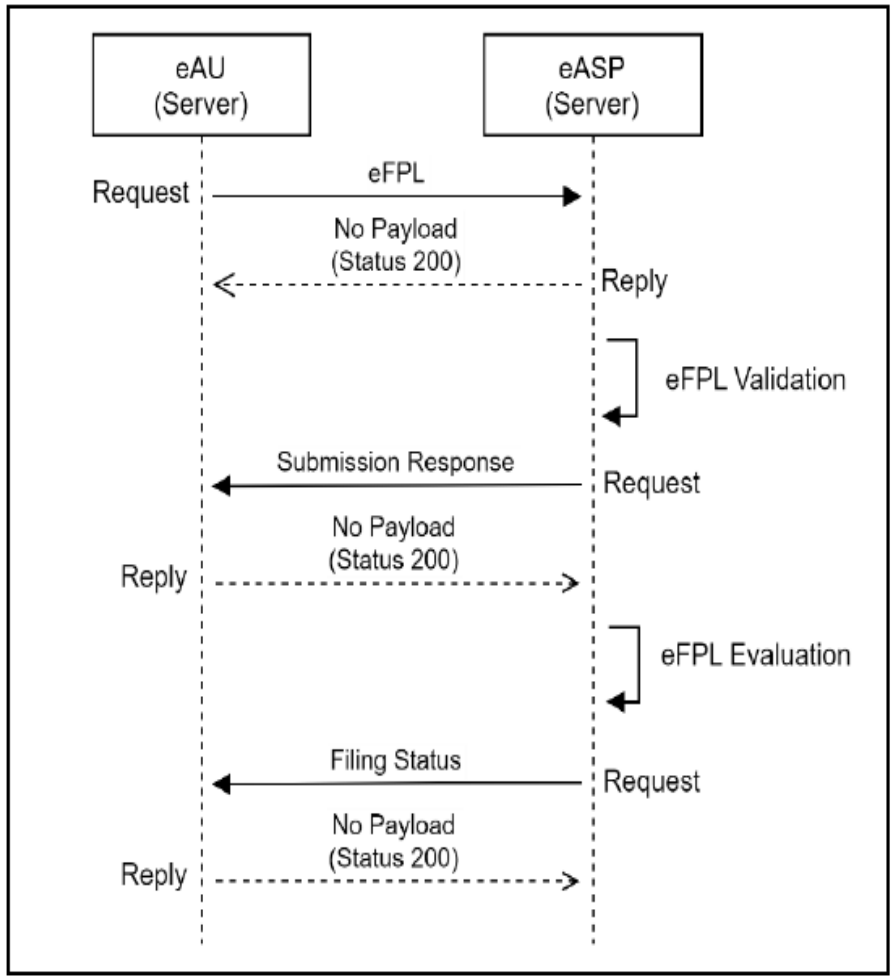
Synchronous Request-Reply

Asynchronous Request-Reply?

Is FF-ICE Filing Service a synchronous or asynchronous service?

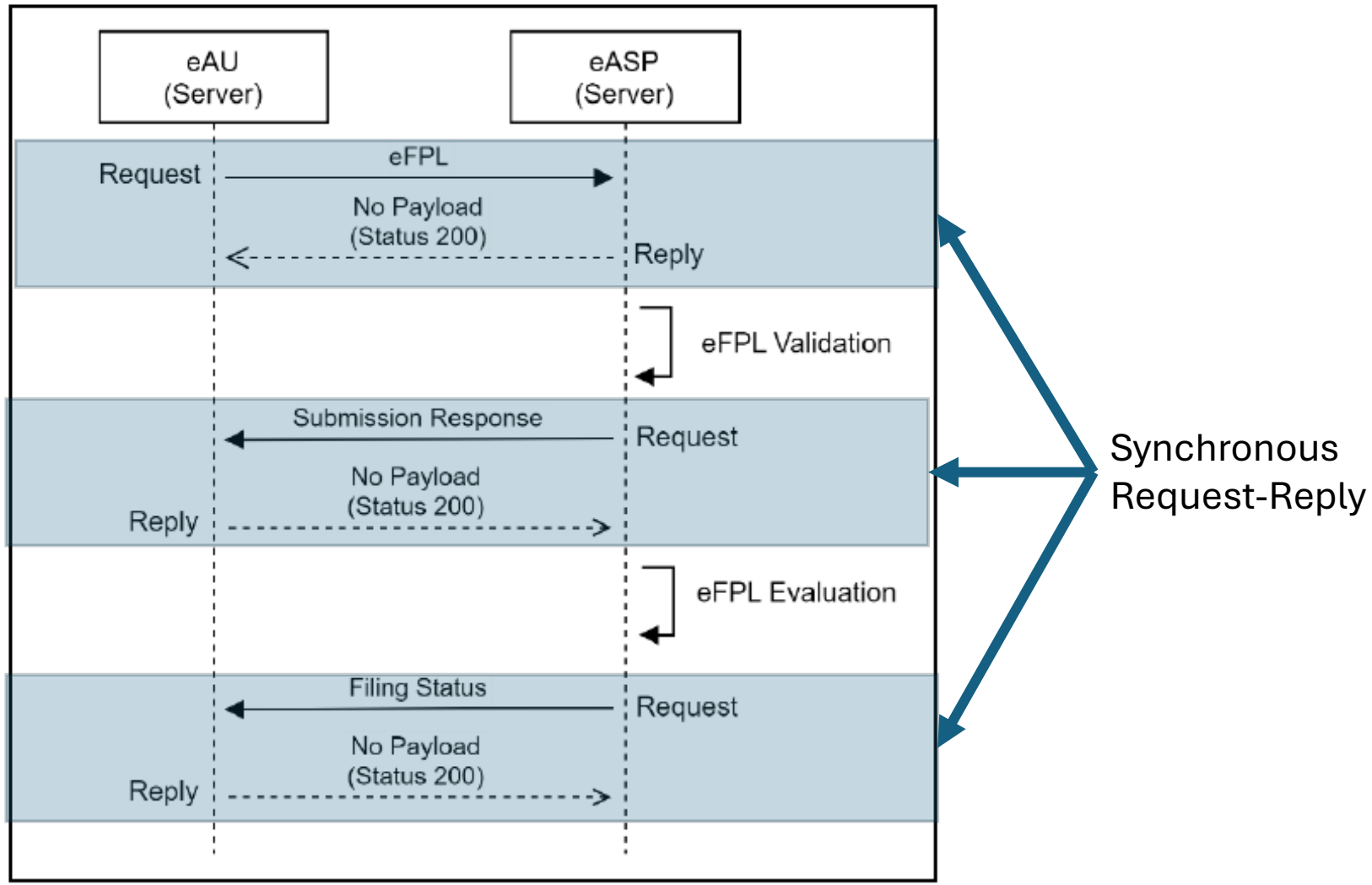
Where's the possible confusion?

- When defining asynchronous or synchronous, do we look at the high-level functionality?
- Using FF-ICE Filing Service as an example, taken in totality, it can be considered asynchronous.
- In actual fact, it's made up of 3 different service calls here.
  - eFPL Submission (eASP Service)
  - Submission Response (eAU Service)
  - Filing Status Update (eAU Service)



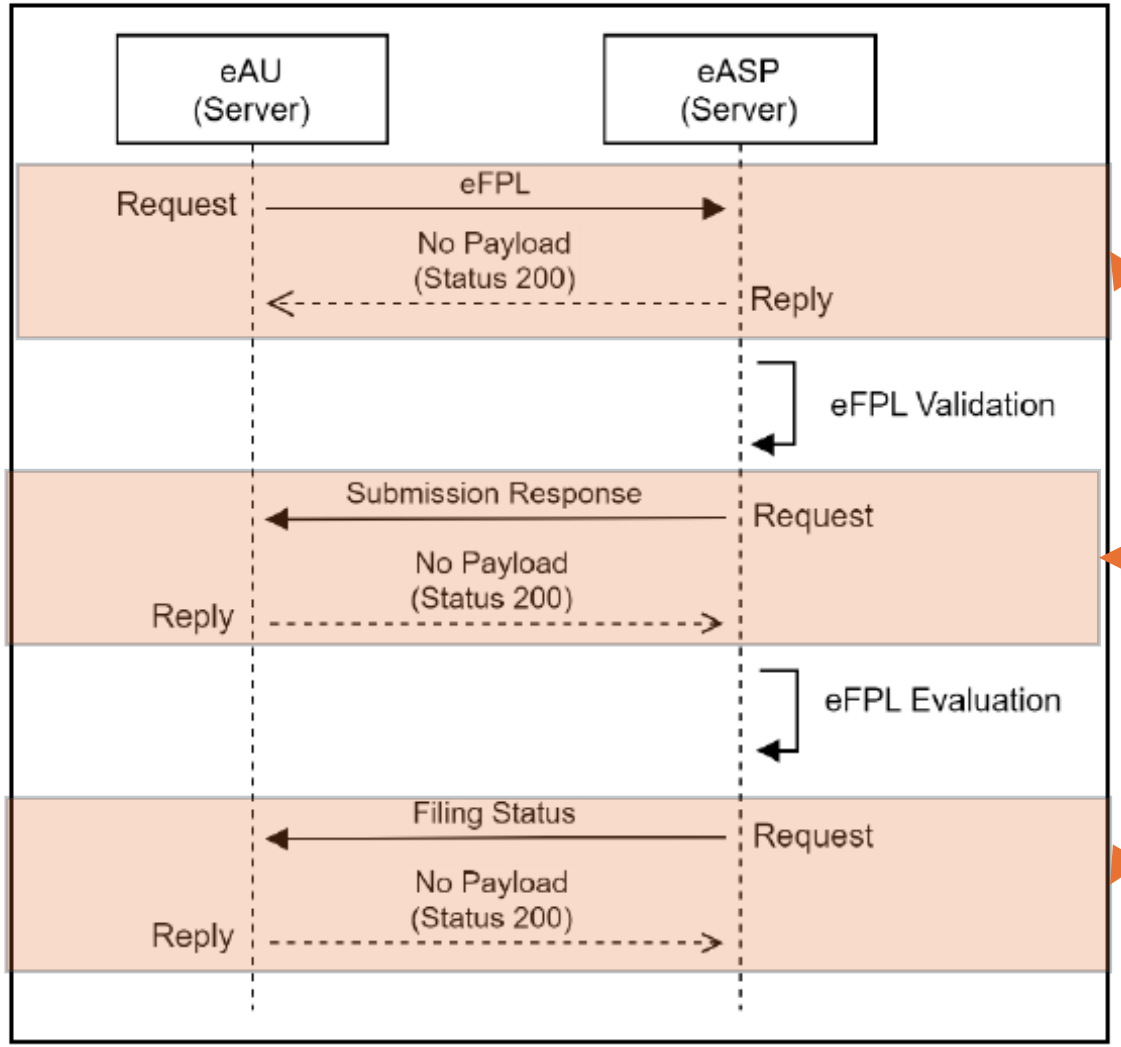
Where's the possible confusion?

FF-ICE Filing Service



Where's the possible confusion?

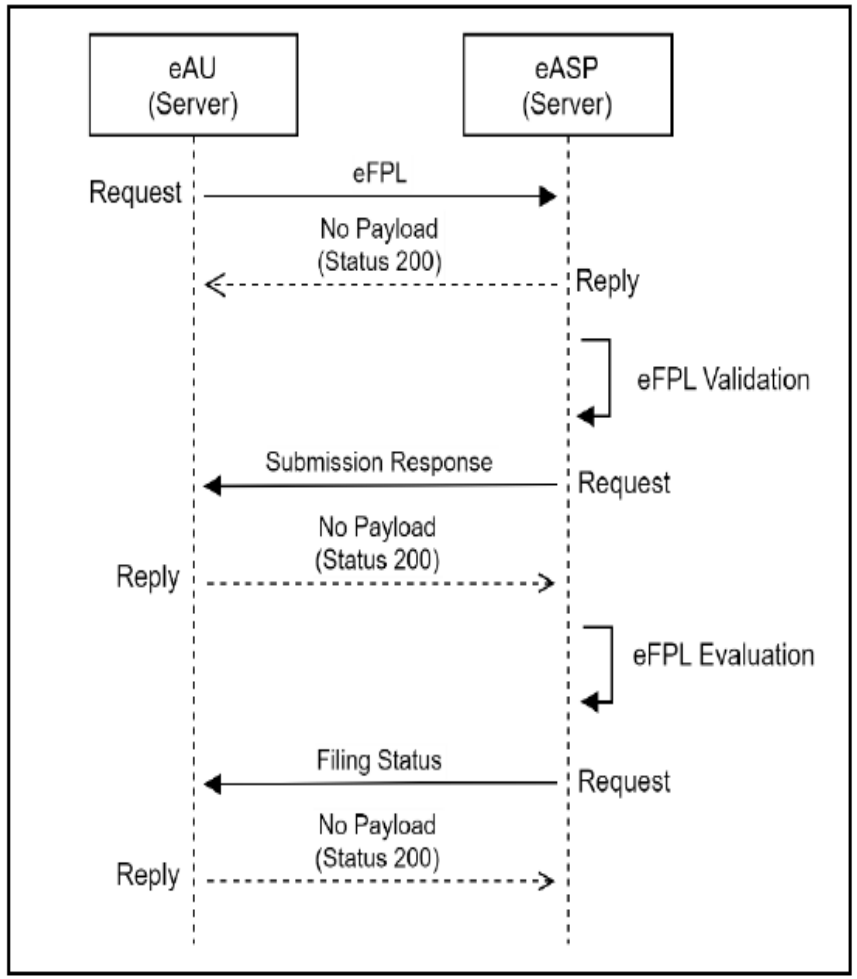
FF-ICE Filing Service



Asynchronous Request-Reply?

Where's the possible confusion?

- Complicating the situation a little more, the 3 individual services can be synchronous or asynchronous here.
- Sequence diagram is unable to tell the full picture.
- In a synchronous Request-Reply, the diagram is as such.
- In an asynchronous Request-Reply, the Request could be going into a queue at the Requestor side, and the response could be a queue at the Replier side.



Way Forward?

- Granularity of Information Service, of Business Use case will need to be discussed and agreed upon.
- Does information services need to be more granular to more accurately reflect the message exchange pattern?
- Common understanding of what synchronous and asynchronous will need to be achieved.
- Ideally, protocols used for asynchronous Request-Reply should be synchronized, i.e. use of AMQP instead of HTTP / Websocket hybrid.

Thank you