

SWIM-SERV-017 Message exchange pattern

- [Requirement](#)
- [Guidance](#)
 - [Understanding message exchange patterns](#)
 - [Examples](#)
- [Guidance for JSON service description](#)
 - [Schema](#)
 - [Guidance](#)
 - [Enumerated values](#)
 - [Example](#)

Requirement

Title	Message exchange pattern
Identifier	SWIM-SERV-017
Requirement	A service description shall include the message exchange pattern used by the service.
Rationale	The message exchange pattern helps understanding how the information interaction with the service works.
Verification	Completeness: Verify that the information is included. Consistency: Verify that the information is consistent with the selected service interface binding. Correctness: Not Applicable.
Examples/Notes	Note: Typical message exchange patterns (as from the SWIM Technical Infrastructure Yellow Profile [RD 3]): <ul style="list-style-type: none">• Request/Reply (synchronous or asynchronous);• Publish/Subscribe (Push or Pull);• One Way (also known as Fire and Forget).
Level of Implementation	Mandatory

Guidance

Understanding message exchange patterns

The document [MEP identification](#), developed in the SWIM-TEC community, provides information on the various notions of message exchange pattern (MEP).

- it introduces the concept of MEP
- it explains the difference between primitive MEPs and application MEPs.
- it provides the semantic of each MEP in text and diagram

Info

This requirement refers to the **application** message exchange pattern used by the service.

Examples

See the [Message exchange pattern section](#) within the [Donlon TOBT Setting Service Description](#).

Guidance for JSON service description

tentative JSON Guidance

[Guidance for JSON service descriptions](#) integrated within the [SWIM Service Description Handbook](#).

i info

Guidance on the full set of categories as available within the European SWIM Registry is provided at requirement [SWIM-REG-0004 Service categorization](#).

Schema

The guidance concerns JSON Schema v0.0.3 (see [Schema releases](#)).

```
    "serviceCategorisation":
      {
        "description" : "A taxonomy used to classify a service by the
type of service provided or by some other technological or architectural solution. [SWIM-SERV-009]",
        "$ref" : "#/definitions/ServiceCategorisation"
      }

    "ServiceCategorisation" :
    {
      "description" : "A taxonomy used to classify a service by the type of service
provided or by some other technological or architectural solution. [SWIM-SERV-009]",
      "type": "object",
      "additionalProperties": false,
      "required": ["serviceType", "lifeCycleStage", "businessActivityType",
"intendedConsumer", "informationCategory", "applicationMessageExchangePattern"],
      "properties":
      {
        "applicationMessageExchangePattern":
        {
          "description" : "The type of Application MEP. [SWIM-SERV-017]",
          "type" : "array",
          "items" : { "$ref": "#/definitions
/CodeApplicationMessageExchangePatternType" },
          "minItems": 1
        },
      },

      "CodeApplicationMessageExchangePatternType" :
      {
        "description" : "A code listing types of application message exchange patterns.
\n",
        "type": "string",
        "enum":
        [
          "ASYNCHRONOUS_REQUEST_REPLY",
          "BROKERED_PUBLISH_SUBSCRIBE_WITH_PUSH_MECHANISM",
          "FAN_OUT",
          "ONE_WAY",
          "PUBLISH_SUBSCRIBE",
          "PUBLISH_SUBSCRIBE_WITH_PULL_MECHANISM",
          "PUBLISH_SUBSCRIBE_WITH_PUSH_MECHANISM",
          "REQUEST_REPLY",
          "SYNCHRONOUS_REQUEST_REPLY"
        ]
      },
    },
```

Rules expressed for the cases as defined in Registry URD.

case	rules
COMPLIANT	mandatory, 1 or more values
CANDIDATE	same
DEFINITION	same

Guidance

Select **one or more values** in field `applicationMessageExchangePattern`, within field `serviceCategorisation`.

Enumerated values

`CodeApplicationMessageExchangePatternType`

REQUEST_REPLY	An application MEP consisting of a consumer sending a request to an information service, the service then executing the request and providing a reply to the consumer.
ONE_WAY	An application MEP consisting of a consumer sending a message to an information service without any requisite response from the information service.
PUBLISH_SUBSCRIBE	An application MEP consisting of a consumer sending a subscription request to an information service, the service then providing the consumer with means to access the subscribed information.
FAN_OUT	An application MEP consisting of a consumer sending sequentially a message to multiple information service according to the One-Way application MEP.
SYNCHRONOUS_REQUEST_REPLY	A Request/Reply application MEP requiring that the consumer wait for the information service to provide a response, and is unable to send/receive any other requests/responses until the initial response is received.
ASYNCHRONOUS_REQUEST_REPLY	A Request/Reply application MEP that does not restrict the consumer from completing other operations while waiting for the information service to respond.
PUBLISH_SUBSCRIBE_WITH_PUSH_MECHANISM	A Publish/Subscribe pattern whereby the information service sends necessary updates (publish) to the consumer, in accordance with the subscription.
PUBLISH_SUBSCRIBE_WITH_PULL_MECHANISM	A Publish/Subscribe pattern whereby the information service keeps necessary updates available to the consumer, in accordance with the subscription.
BROKERED_PUBLISH_SUBSCRIBE_WITH_PUSH_MECHANISM	A Publish/Subscribe pattern with push mechanism introducing a layer of decoupling between the publisher and subscribers by means of a broker.

Example

```
"serviceCategorisation": {  
  "applicationMessageExchangePattern": [  
    "SYNCHRONOUS_REQUEST_REPLY"  
  ],  
}
```

A complete JSON example is available in page [JSON example - Donlon TOBT Setting service description](#).