Implement

service implementation is the set of activities where the information service is implemented in a target environment and technology context.

Technical Interoperability Design

- Selection/definition of the data format
- Definition of the message(s) used to interact with the service interface
- Selection of the service interface protocols
- Implementation of service(s) using technology and based on implementation choices made
- Integration of the service(s) into the target environment
- Verification and testing of the service(s)
- Validation of the service(s)

- Chosen XM or other data format.
- Chosen service interface protocol.
- Message definition
- Implemented service(s) (interfaces and operations)
- Machine readable service definition Verification report, validation report
- Service Overview (update)
- Verification report
- Validation report
- Service implementation information (e.g. service interface protocols and QoS characteristics)

Technical Interoperability Design
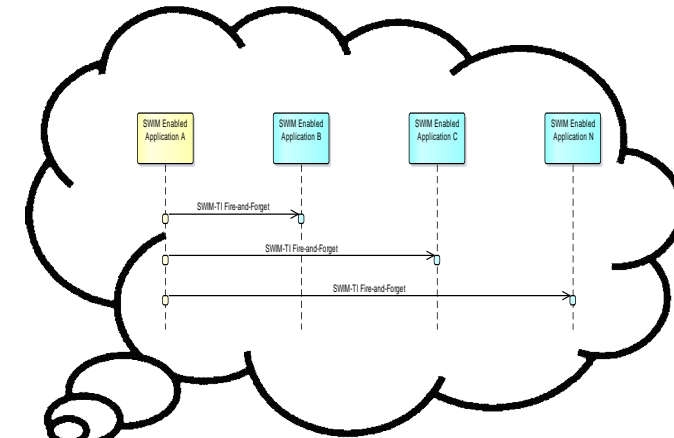
Process: Design

- Selection of the message exchange pattern (MEP)
- Definition of the service (interface, service operations and information service payload)

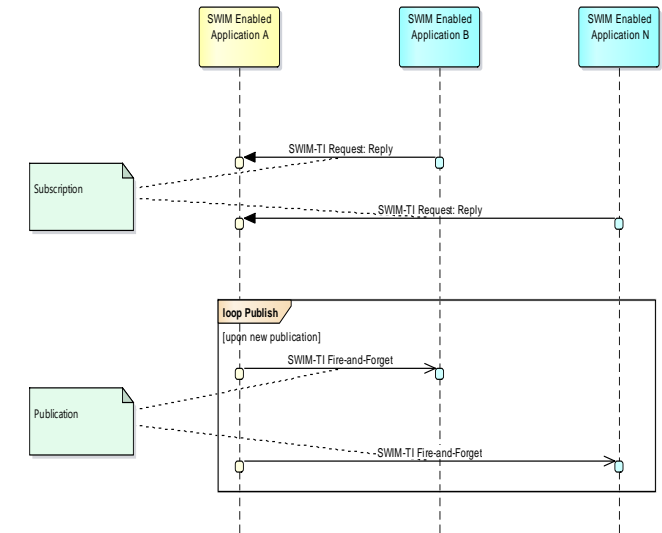## Process: Implementation

- Information
  - Exchange Formats
  - Message Definitions
- Interface
  - Interface Bindings
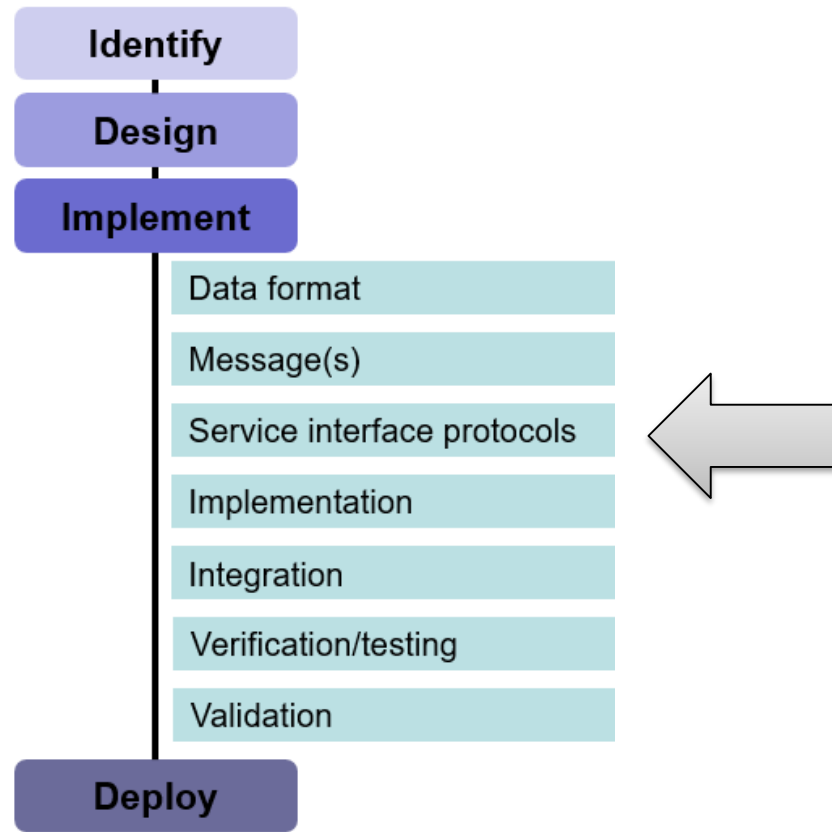   - Protocol Methods
    - Primitive Patterns

Logical/conceptual design

Implementation Design

Interface Binding Determination

References

Resources

SWIM TI Binding Selection Guidelines

https://ost.eurocontrol.int/sites/AISWIM/SWIMspecs/TEC/Shared%20Documents/Supporting%20Material/03%20-%20SWIM%20TI%20Binding%20Selection%20Guidelines/SWIM%20TI%20Binding%20Selection%20Guidelines.docx

Introduction
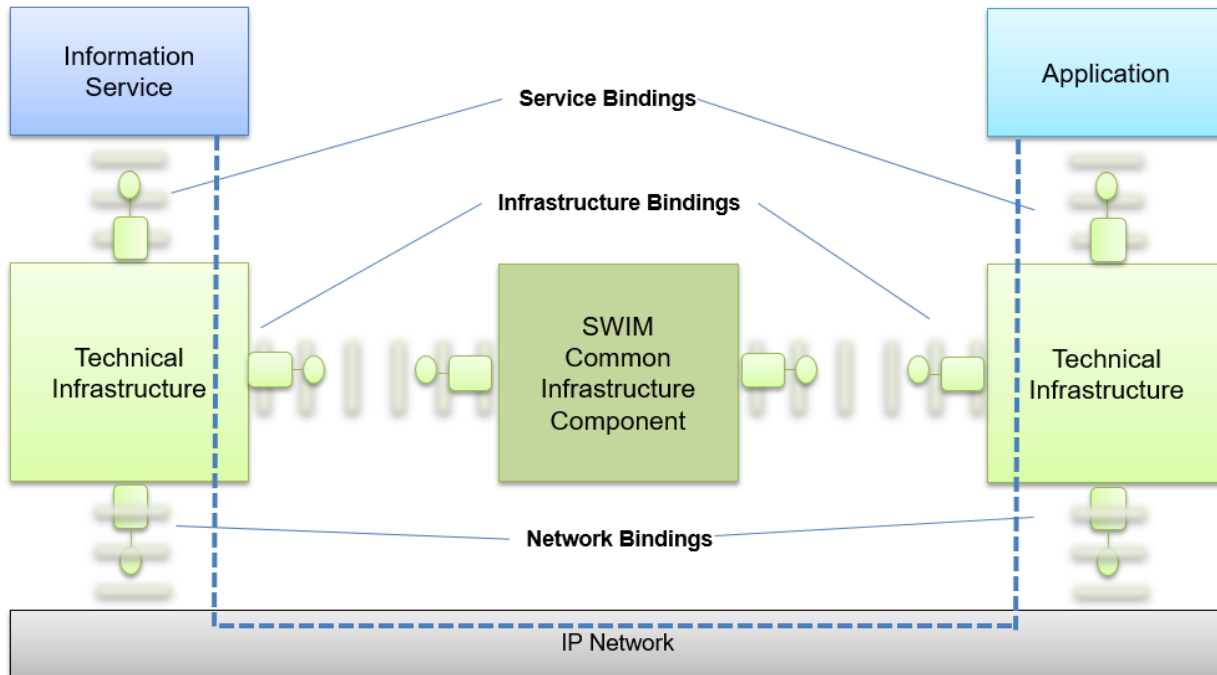
Technical interoperability enabled by the SWIM TI Interface Bindings.

A binding is a consistent, self-contained grouping of interface requirements that specify the protocols and configuration options to be used for the exchange of information between systems.

Three types of interface bindings: Service, Network, Infrastructure.

Service Binding Introduction

Specifying the technical infrastructure protocols that are used by the application to interoperate with information services.

- **WS Light**

- **WS SOAP**

- **WS SOAP with Basic Message Security**

- **WS SOAP with Message Security**

- **WS SOAP with Federated Security**

- **WS-N SOAP**

- **WS-N SOAP with Basic Message Security**

- **WS-N SOAP with Message Security**

- **WS-N SOAP with Federated Security**

- **AMQP Messaging**

Service Binding Selection Criteria

- MEP
- Protocols and Formats
- Security Protocols
- Reliability Capabilities
- Bandwidth Efficiency

Service Binding Selection based on MEP

- Primitive MEPs are those that are directly related to the capabilities of the lower level protocols of the SWIM TI:
  - Fire and Forget
  - Synchronous Request/Reply

	SWIM-TI Fire-and-Forget	SWIM-TI Synchronous R/R
WS Light	NO	YES
WS SOAP	NO	YES
WS SOAP with Basic Message Security	NO	YES
WS SOAP with Message Security	NO	YES
WS SOAP with Federated Security	NO	YES
AMQP Messaging	YES	NO

Interface Binding Determination
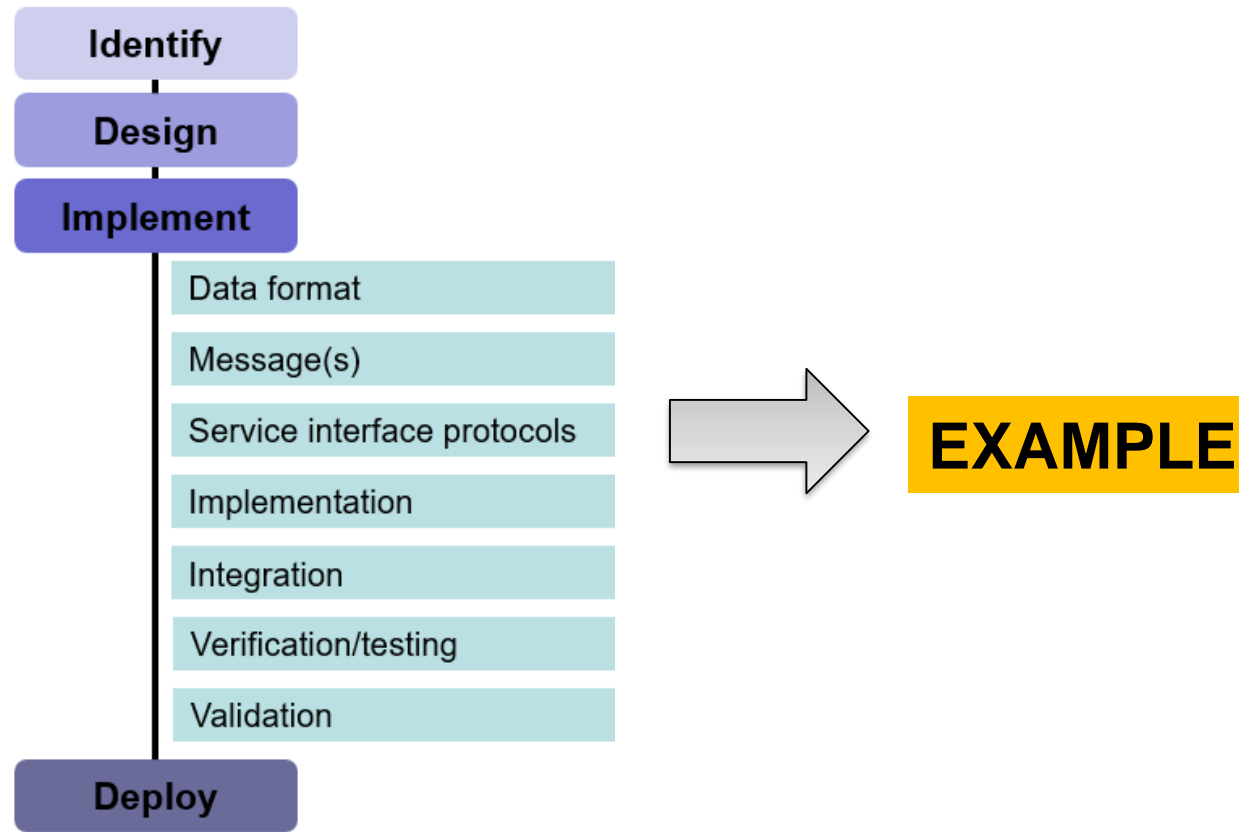
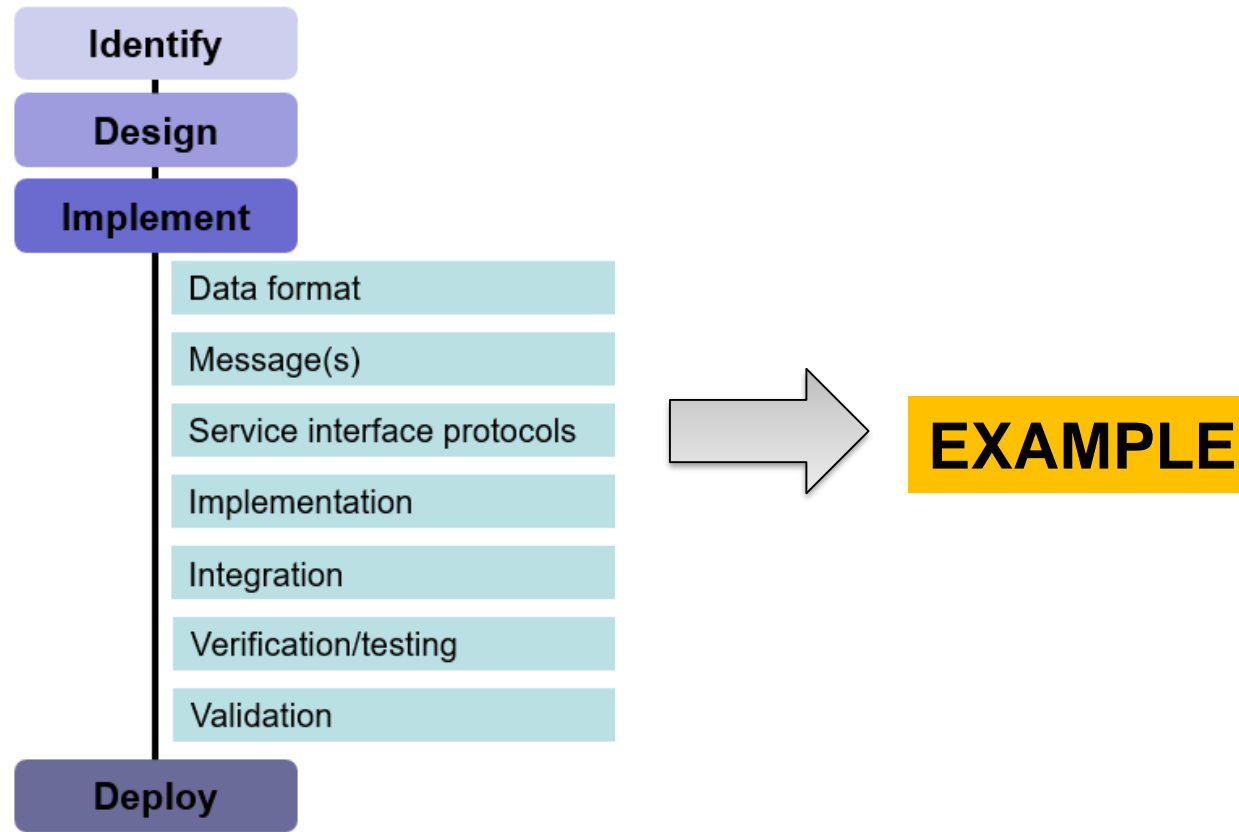# Binding Selection / Requirements and Preferences (example)

- MEP (as identified)
  - Synchronous Request/Reply
- Protocols and Formats (implementer preferences)
  - Java environment, XML
- Bandwidth Efficiency (environment constraints)
  - WAN not particularly constraint
- Security (technical decisions)
  - Transport security, no need for message level security

Binding Selection / Decision (example)

WS-SOAP

Identifier	Title	Level of Implementation
SWIM-TIYP-0041	Data Compression	O
SWIM-TIYP-0035	WS Reliable Messaging	O
SWIM-TIYP-0045	HTTP POST	M
SWIM-TIYP-0044	HTTP Reason Phrase Header	M
SWIM-TIYP-0043	HTTP Status Code Header	M
SWIM-TIYP-0042	TLS Authentication	M
SWIM-TIYP-0023	WS Policy	O
SWIM-TIYP-0039	HTTP Header Transfer Encoding	O
SWIM-TIYP-0038	HTTP Compression and Content Encoding Header	M Conditional
SWIM-TIYP-0033	HTTP Content Type Header	M
SWIM-TIYP-0032	XML Schema Validation	M
SWIM-TIYP-0029	XML	M
SWIM-TIYP-0016	WS-I	M
SWIM-TIYP-0015	WSDL 2.0	O
SWIM-TIYP-0014	WSDL 1.1	M
SWIM-TIYP-0013	MTOM	O
SWIM-TIYP-0012	SOAP 1.2	O
SWIM-TIYP-0011	SOAP 1.1	M
SWIM-TIYP-0010	HTTP over TLS	M
SWIM-TIYP-0009	HTTP	M
SWIM-TIYP-0008	TLS	M

Interface Binding Configuration

Binding Configuration

Title	Level of Implementation
Data Compression	O
WS Reliable Messaging	O
HTTP POST	M
HTTP Reason Phrase Header	M
HTTP Status Code Header	M
TLS Authentication	M
WS Policy	O
HTTP Header Transfer Encoding	O
HTTP Compression and Content Encoding Header	M Conditional
HTTP Content Type Header	M
XML Schema Validation	M
XML	M
WS-I	M
WSDL 2.0	O
WSDL 1.1	M
MTOM	O
SOAP 1.2	O
SOAP 1.1	M
HTTP over TLS	M
HTTP	M
TLS	M

Data transmission optimization

❖ None of the following mechanisms available in the binding will be implemented:

- Compression
- Chunked transfer encoding
- Message transmission optimisation for binary data transfers

Binding Configuration

Title	Level of Implementation
Data Compression	O
WS Reliable Messaging	O
HTTP POST	M
HTTP Reason Phrase Header	M
HTTP Status Code Header	M
TLS Authentication	M
WS Policy	O
HTTP Header Transfer Encoding	O
HTTP Compression and Content Encoding Header	M Conditional
HTTP Content Type Header	M
XML Schema Validation	M
XML	M
WS-I	M
WSDL 2.0	O
WSDL 1.1	M
MTOM	O
SOAP 1.2	O
SOAP 1.1	M
HTTP over TLS	M
HTTP	M
TLS	M

Reliable messaging

❖ The WS Reliable Messaging protocol will not be implemented

❖ The consuming application will require to implement exception handling based on error messages, time-outs.

*The purpose of WS-RM is to ensure delivery of messages in situations such as the destination endpoint being temporarily unavailable (for example, in the case of a server restart) or the message path crossing multiple transport connections, any of which might fail (for example, across a firewall). WS-RM offers greater reliability when using HTTP transport, but has a performance impact.

Binding Configuration

Title	Level of Implementation
Data Compression	O
WS Reliable Messaging	O
HTTP POST	M
HTTP Reason Phrase Header	M
HTTP Status Code Header	M
TLS Authentication	M
WS Policy	O
HTTP Header Transfer Encoding	O
HTTP Compression and Content Encoding Header	M Conditional
HTTP Content Type Header	M
XML Schema Validation	M
XML	M
WS-I	M
WSDL 2.0	O
WSDL 1.1	M
MTOM	O
SOAP 1.2	O
SOAP 1.1	M
HTTP over TLS	M
HTTP	M
TLS	M

HTTP Implementation

✓ HTTP1.1

✓ HTTP POST

✓ HTTP responses including status code and phrase header.

The services make use of the standard HTTP 400 error [Bad Request] in any of the following cases:
•The request is for an unsupported release
•The request is not a well-formed XML
•The request is a well-formed XML but it is not valid with respect to the XSD (i.e. it does not conform to the type and attribute names defined in the XSD and documented in the reference manuals). Examples of causes for invalid XML documents are:
- Unexpected element or attribute
- Element order violation
- Incorrect primitive value
- Unexpected enum value

Binding Configuration

Title	Level of Implementation
Data Compression	O
WS Reliable Messaging	O
HTTP POST	M
HTTP Reason Phrase Header	M
HTTP Status Code Header	M
TLS Authentication	M
WS Policy	O
HTTP Header Transfer Encoding	O
HTTP Compression and Content Encoding Header	M Conditional
HTTP Content Type Header	M
XML Schema Validation	M
XML	M
WS-I	M
WSDL 2.0	O
WSDL 1.1	M
MTOM	O
SOAP 1.2	O
SOAP 1.1	M
HTTP over TLS	M
HTTP	M
TLS	M

Security protocols

✓ TLS1.2

✓ Server authentication with X.509 and Client authentication with HTTP Basic ~~or HTTP Digest~~.

✓ AES_128_GCM_SHA256, AES_256_CCM

Binding Configuration

Title	Level of Implementation
Data Compression	O
WS Reliable Messaging	O
HTTP POST	M
HTTP Reason Phrase Header	M
HTTP Status Code Header	M
TLS Authentication	M
WS Policy	O
HTTP Header Transfer Encoding	O
HTTP Compression and Content Encoding Header	M Conditional
HTTP Content Type Header	M
XML Schema Validation	M
XML	M
WS-I	M
WSDL 2.0	O
WSDL 1.1	M
MTOM	O
SOAP 1.2	O
SOAP 1.1	M
HTTP over TLS	M
HTTP	M
TLS	M

WS Interoperability

✓ WS-I Basic Profile Version 1.2

✓ SOAP 1.1

WS Description

✓ WSDL 1.1

✓ XML

✓ No use of WS Policy

Intended to describe the capabilities and constraints (e.g. security tokens, supported encryption algorithms)

Interface Binding Determination