



Solution Guide

---

# LSO Sonata On-boarding and Inter-op Verification

May 2021

## Disclaimer

© 2021 MEF Forum. All Rights Reserved.

The information in this publication is freely available for reproduction and use by any recipient and is believed to be accurate as of its publication date. Such information is subject to change without notice and MEF Forum (MEF) is not responsible for any errors. MEF does not assume responsibility to update or correct any information in this publication. No representation or warranty, expressed or implied, is made by MEF concerning the completeness, accuracy, or applicability of any information contained herein and no liability of any kind shall be assumed by MEF as a result of reliance upon such information.

The information contained herein is intended to be used without modification by the recipient or user of this document. MEF is not responsible or liable for any modifications to this document made by any other party.

The receipt or any use of this document or its contents does not in any way create, by implication or otherwise:

- a) any express or implied license or right to or under any patent, copyright, trademark or trade secret rights held or claimed by any MEF member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- b) any warranty or representation that any MEF members will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- c) any form of relationship between any MEF member and the recipient or user of this document.

Implementation or use of specific MEF standards, specifications, or recommendations will be voluntary, and no Member shall be obliged to implement them by virtue of participation in MEF Forum. MEF is a non-profit international organization to enable the development and worldwide adoption of agile, assured and orchestrated network services. MEF does not, expressly or otherwise, endorse or promote any specific products or services.

## Contents

1	Introduction.....	5
	Highlights .....	5
2	Terminology and Abbreviations.....	6
3	What? Solution Requirements.....	8
	3.1 Goals.....	8
	3.2 Partner Engagement Lifecycle.....	9
	3.3 Solution Requirements.....	11
	3.3.1 Seller On-boarding & Verification Solution Requirements.....	11
	3.3.2 Buyer On-boarding & Verification Solution Requirements .....	13
	3.3.3 Common Solution Requirements.....	15
4	How? Solution .....	16
	4.1 Testing a Seller Implementation Using a Buyer Emulator.....	16
	4.1.1 Main Buyer Emulator Operational Flow .....	18
	4.2 Testing a Buyer Implementation Using a Seller Emulator.....	22
	4.2.1 Main Seller Emulator Operational Flow .....	24
5	Provider Benefits.....	25
6	About MEF .....	25
7	Acknowledgements .....	26

## List of Figures

Figure 1 – LSO Sonata Partner Engagement Lifecycle .....	9
Figure 2 – Expected Use of LSO Sonata Buyer & Seller Emulators .....	10
Figure 3 – Buyer Emulator High-Level Reference Architecture .....	16
Figure 4 – Buyer Emulator Test Case Template Design Process .....	18
Figure 5 – Seller Emulator High-Level Reference Architecture .....	22

## List of Tables

Table 1 – Terminology & Abbreviations .....	7
Table 2 – Expected use of Buyer & Seller Reference Implementation Emulators .....	11
Table 3 – Buyer Emulator Function/Component Descriptions .....	17
Table 4 – Seller Emulator Function/Component Descriptions .....	23

## 1 Introduction

This Solution Guide document has been developed by the MEF Commerce & Business Committee (CBC) Incubation Group (IG) for LSO (Lifecycle Service Orchestration) Sonata Partner On-boarding and Inter-op Verification. The document comprises an overview of the goals and objectives for the solution, the recommended solution requirements and a solution description, including recommended abstract reference architecture and descriptions of the key operational flows / use-cases. The intended users of this document are:

- Service providers who have implemented LSO Sonata or are preparing a first implementation and want to know what is required to on-board and maintain inter-op with partners. It is specifically targeted to those service provider personnel who are responsible for LSO Sonata API partner on-boarding.
- As a guide for those tasked with designing and developing the On-boarding and Inter-op solution.

### Highlights

- Context and overarching goals for the LSO Sonata On-boarding and Inter-op verification solution.
- Description of the Partner Engagement lifecycle (timeline) and the application of the solution at each stage of the lifecycle.
- What? – Description of the key solution requirements for the solution.
- How? – High-level reference architecture and description of key operational flows for the solution covering both the LSO Sonata Buyer and Seller perspectives.
- Benefits of the solution.

While the solution scope for the purpose of this document is LSO Sonata APIs, the recommendations provided cater for possible future expansion to support other LSO APIs.

## 2 Terminology and Abbreviations

This section defines the terms used in this document.

Term	Definition
Buyer Emulator	A programmable, standardized reference implementation MEF LSO Sonata Buyer emulation that can be configured to support Buyer-specific settings, including product offerings, LSO Sonata handling options (sync /async) and data-sets to emulate real Buyer system requests.
Seller Emulator	A programmable, standardized reference implementation MEF LSO Sonata Seller emulation (or "twin") that can be configured to support Seller-specific settings, including product offerings, LSO Sonata handling options (sync /async) and data-sets to emulate real Seller system behavior.
Test Suite	Grouping of Test Cases and Test Sequences, including but not limited to tests for a specific partner or product offering.
Test Sequence	Sequence of related Test Cases, including the transition rules/logic needed to execute the sequence, including but not limited to sequences for real trades POQ -> Quote -> Order.
Test Case Template	A pre-defined, re-usable parameterized template with rules and logic (if applicable) that is used to simplify the design of a Test Case for a specific Sonata function.
Test Case	An executable test for a specific LSO Sonata function that is formed by combining a Test Case Template with specific input-data/parameter values.
LSO Sonata Envelope	The independent part of the LSO Sonata function request that is common to all Test Cases for that LSO Sonata function (release specific).
Partner Configs	Settings required to communicate with a specific partner, including base URL and security credentials.
Product/Offering	The formal specification of a type of product traded between partners and the specific flavor of that product as it is offered to partners.

Term	Definition
LSO Sonata Config	LSO Sonata release settings and API endpoint configurations.
Address	Geographic address in seller database, representing a particular location (place).
Service Site	Fixed physical location at which a Product can be installed.
Product Inventory	Inventory of Seller product instances sold to customers and/or introduced manually by the seller.
Pricing	Pricing data supplied by a seller in response to a quote / order request.

Table 1 - Terminology & Abbreviations

### 3 What? Solution Requirements

Success of the LSO Sonata ecosystem is dependent on attracting a rich and diverse set of Providers who can on-board and inter-operate quickly. Each Provider can have 10s or even 100s of partners (customers and suppliers), making it prohibitively expensive and time-consuming in terms of adding and maintaining inter-operability with partners and growing the overall LSO Sonata ecosystem.

#### 3.1 Goals

The overarching goals for LSO Sonata On-boarding and Inter-op are to:

- Introduce standards, processes, and tools each Provider can use to develop their individual LSO Sonata implementations, including a means to verify that it reaches an agreed benchmark before the Provider engages partners in on-boarding and inter-op testing.
- Ensure a fast, cost efficient, automated, and measurable means for a provider to on-board and maintain (including after updates and upgrades) inter-op with its network of partners.
- Provide programmable, standardized reference implementation emulators for MEF LSO Sonata Buyer and Seller that all providers use during development of their LSO Sonata implementation as well as partner On-boarding and Inter-op verification to ensure a high degree of compatibility and compliance across implementation.
- Ensure an overall solution that is accessible and economically viable to support the range of large, medium, and small providers who form the LSO Sonata eco-system (must have).
- Be able to support current and future standardized inter-provider services as well as non-standard ones.
- Provide a natural path to certification.



### 3.2 Partner Engagement Lifecycle

To meet the goals and address the problems identified we must understand the partner engagement lifecycle (timeline) as described below:

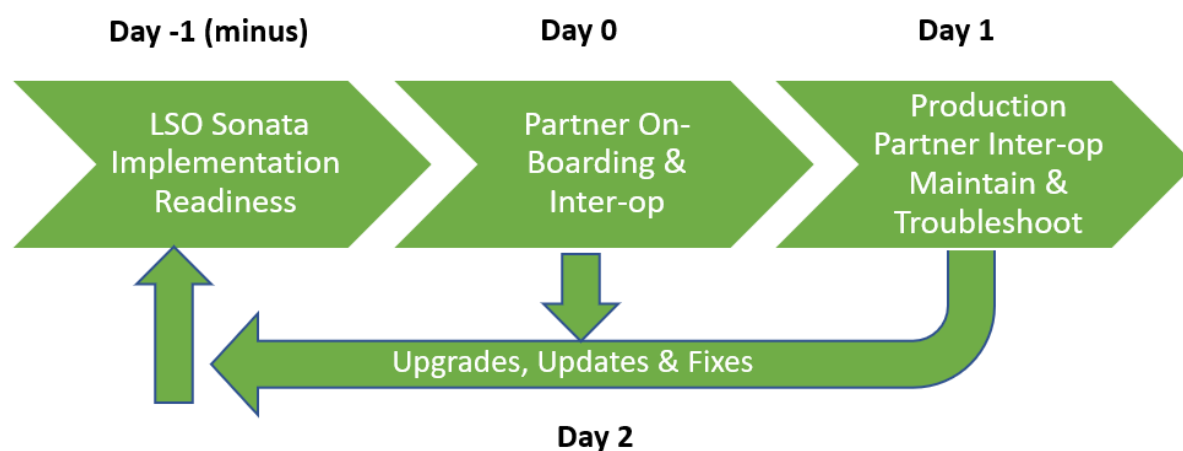


Figure 1 – LSO Sonata Partner Engagement Lifecycle

**Day -1 (minus)** – LSO Sonata implementation readiness (prior to engaging any partners).

**Day 0** – Initial partner On-Boarding and Inter-op testing (specific partner engagement).

**Day 1** – Troubleshooting and maintaining inter-op in production.

**Day 2** – Re-establishing inter-op after modifications (upgrades, updates, additional features, new product offerings, fixes) though (automated) regression testing.

The MEF LSO Sonata Buyer and Seller emulators will support providers at each stage in the lifecycle. Initially, for familiarization and testing when preparing their LSO Sonata implementation; then, by allowing them to pre-configure emulators to support their specific configurations (LSO Sonata release/function support, handling, product offerings, and default data sets) to reduce the time and cost of dealing with multiple partners, then to undertake on-boarding and inter-operability with each partner, and finally for maintenance and trouble-shooting issues with partners when in production.

The following diagram and table describe the expected use of the solution at each stage of the partner engagement lifecycle:

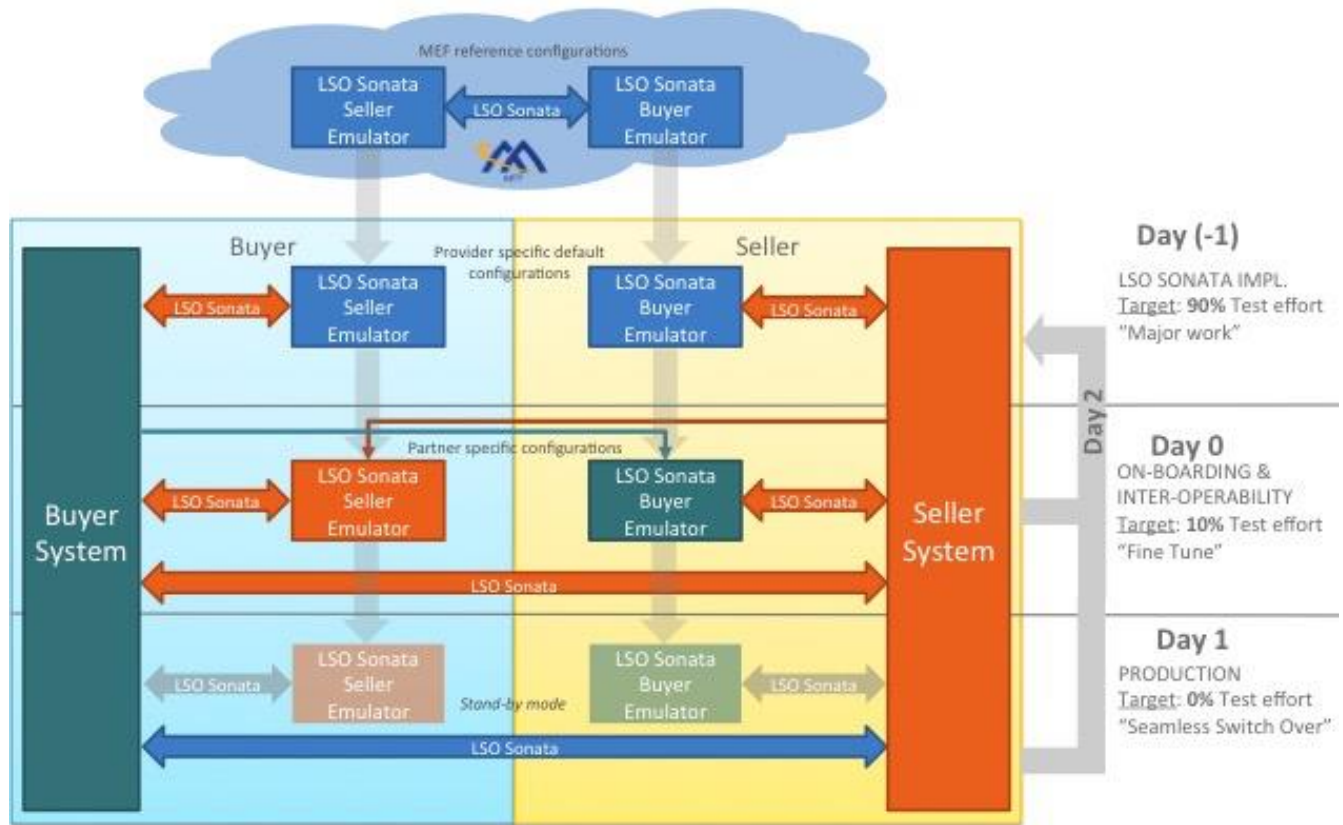


Figure 2 – Expected Use of LSO Sonata Buyer & Seller Emulators

Stage	Expected use of Buyer and Seller Reference Implementation Emulators
Day -1 (minus) LSO Sonata Implementation readiness (prior to engaging partners)	<ul style="list-style-type: none"> <li>Buyer and Seller Emulators to be used in combination for familiarization with LSO Sonata.</li> <li>Separately to test during LSO Sonata implementation development.</li> <li>To create pre-configured version(s) of the emulator(s) loaded with provider-specific configurations (LSO Sonata release/function support, handling, product offerings and default data sets) applicable to all partners that are ready to be fine-tuned with minimum effort for each partner to support Day 0.</li> </ul>
Day 0 – Initial partner On-Boarding and Inter-op testing (specific partner)	<ul style="list-style-type: none"> <li>To emulate real per-partner behavior prior to engaging the partner in inter-op testing with their real</li> </ul>

	<p>LSO Sonata implementation – starts with the provider-specific pre-configured emulators, which are then fine-tuned for the specific partner.</p> <ul style="list-style-type: none"> <li>• Be used during partner inter-op testing to troubleshoot issues that may arise when testing with the partner’s real LSO Sonata implementation.</li> </ul>
Day 1 - Troubleshooting and maintaining inter-op in production	<ul style="list-style-type: none"> <li>• To emulate real per-partner behavior as a means to quickly isolate the source of problems that may arise with production system i.e. test baseline.</li> </ul>
Day 2 - Re-establishing inter-op after modifications	<ul style="list-style-type: none"> <li>• The same as for Day -1 (minus) for supporting new LSO Sonata releases, features, new product offerings, and fixes.</li> </ul>

Table 2 – Expected use of Buyer & Seller Reference Implementation Emulators

Support for Day -1 (minus) and Day 0 stages are most important for scaling adoption among providers. By introducing the LSO Sonata reference implementation emulators, each provider can verify their own implementation is compliant and compatible with their partner configurations (using partner-specific emulators) before engaging partners in inter-op testing. This will eliminate the current time-consuming exercise that requires partners to work together to debug their implementations to resolve differences in LSO Sonata interpretations as well as common coding issues *e.g. wrong field names*.

### 3.3 Solution Requirements

The following solution requirements are recommended (non-normative) to be addressed by the solution. They are categorized into Seller On-boarding, Buyer On-boarding, and common requirements.

#### 3.3.1 Seller On-boarding & Verification Solution Requirements

The primary requirement here is for a programmable, standardized, reference implementation of the LSO Sonata Buyer emulator that can be used by providers during development of LSO Sonata implementation, as well as partner On-boarding and Inter-op verification. The Buyer emulator is analogous to a client in a client-server relationship; therefore, its role is to generate requests and handle responses or notifications from the Seller. It is responsible for initiating tests towards the Seller.

1. Day -1 LSO Sonata implementation readiness: The solution should define standardized testing processes, Test Suites and quality benchmark(s) for the independent testing that a Provider should undertake with their LSO Sonata Seller implementation before they engage partners in on-boarding and inter-op testing.
2. Day 0 to 2 Partner On-boarding and Inter-op verification: The solution should define standardized processes, Test Suites and quality benchmark(s) for initial and subsequent testing a Provider should undertake for partner on-boarding and inter-op verification.
3. The solution should ensure alignment/compatibility with MEF W92.1 Test Requirements Specification (common to/used by LSO Sonata On-boarding and Inter-op verification, as well as LSO Sonata Test and Certification).

4. Buyer Emulator: The solution should provide a Buyer emulator tool that incorporates:
- a) Partner management:
    - Ability to manage partners and partner-specific configurations, including partner-specific test-suites and results for Day 0-2 testing
    - Support a default partner for Day -1 (minus) testing
  - b) Per-partner configurations, including:
    - Security settings (*per MEF LSO Sonata security requirements, including oauth2 and openidconnect*).
    - LSO Sonata release support
    - LSO Sonata feature/capability support *e.g., Address Validation, Site Query, PoQ, Quote, Order and optional subfunctions, etc.*
    - LSO Sonata handling support *e.g. Async/Sync response, response time limits, etc.*
    - Pair-wise Product offering(s)
    - Test data set(s)
  - c) A common industry standard open-test framework *e.g. Postman*
  - d) Test design
    - Supports processes and mechanisms to create and manage Test Case Templates
    - Supports processes and mechanisms to create and manage Test Cases, including support for both positive and negative Test Cases (for example for those defined in MEF W92.1)
    - Supports processes and mechanisms to create and manage Test Sequences
  - e) Test execution
    - Support a mechanism to load and execute Test Cases and Test Sequences
    - Support for LSO Sonata-compliant request generation
    - Support for LSO Sonata-compliant response handling logic
    - Support for LSO Sonata-compliant notification registration and handling
    - Support for closed-loop, policy driven request/response handling to automate response handling and sequencing of tests *e.g. end-to-end trades*
  - f) Result management
    - Support ability to view and analyze test results
    - Support for management of historical test results
  - g) Diagnostic support

- Support logging and traceability of test requests and responses to isolate the source of unexpected test results
  - Support configuration of underlying operating system to use an NTP Server to ensure accurate temporal references.
- h) Automated testing
- Support a programmable means to integrate testing into an automated test or CI/CD environment
- i) Load testing
- Support API load testing
5. The solution should provide a set/sub-set of pre-built Test Case Templates for individual LSO Sonata function API (per release) tests that match the test requirements specified in MEF W92.1 LSO Sonata Test Requirements Specifications.
6. The solution should define and publish a standardized list of Test Cases and Test Sequences sellers must pass for:
- a) Day -1 Implementation Readiness
  - b) Day 0-2 Partner On-boarding and Inter-op Verification
7. The solution should provide the following Products support:
- a) Ability to manage definitions of Products / Product offerings that are traded between partners
  - b) Ability to support any type of standard or non-standard inter-provider product offering (e.g. connectivity, cloud, VAS and combinations)
  - c) Ability to support current and future standardized inter-provider product offerings: Initially “component” services like Carrier Ethernet extending to Internet Access IP and other services in the future per a roadmap to be agreed upon.
8. The solution should provide management of inventory and relationships for Product, Sites, Address (of each type), POQs, Quotes, and Orders.
9. The solution should support LSO Sonata compliant notification mechanism.

### 3.3.2 Buyer On-boarding & Verification Solution Requirements

The primary requirement here is for a programmable, standardized reference implementation MEF LSO Sonata Seller emulator that can be used by providers during development of LSO Sonata implementation, as well as partner On-boarding and Inter-op verification. The Seller emulator is analogous to a server in a client-server relationship; therefore, its role is to process requests and send response or notifications to the Buyer. It is not responsible for initiating any tests.

10. Seller Emulator: The solution should provide a Seller emulator tool that incorporates:
- a) Partner management:

- Ability to manage partners and partner-specific configurations, including partner-specific test logs, etc. for Day 0-2 testing.
  - Support a default partner for Day -1 testing.
- b) Per-partner configurations, including:
- Security settings (*per MEF LSO Sonata security requirements, including oauth2 and openIdconnect*) as well as the ability to integrate with API Gateways and IAM (Integrated Access Management)
  - LSO Sonata release support
  - LSO Sonata feature/capability support, *e.g, Address Validation, Site Query, PoQ, Quote, Order and optional subfunctions etc.*
  - Sonata handling support *e.g. Async/Sync response, response time limits etc.*
  - Pair-wise Product offering(s)
  - Test data set(s)
- c) Support for LSO Sonata-compliant request handling
- d) Support for LSO Sonata-compliant response generation logic using the per-partner specific configurations
- e) Support for LSO Sonata-compliant notification server, including support for notification registration and handling and ability to notify buyers of events they have registered for
- f) Support for closed-loop, policy-driven request handling to automate responses
- g) Support logging and traceability of test requests, processing and results to quickly isolate the source of unexpected behavior
- h) Support API Load testing
- i) Support configuration of underlying operating system to use an NTP Server to ensure accurate temporal references.

**11. The solution should provide the following Product support:**

- a) Ability to manage definitions of Products / Product offerings that are traded between partners
- b) Ability to support any type of standard or non-standard inter-provider product offering (e.g. connectivity, cloud, VAS and combinations)
- c) Ability to support current and future standardized inter-provider product offerings: Initially “component” services like Carrier Ethernet, extending to Internet Access, IP and other services in the future, roadmap to be agreed upon

**12. The solution should provide storage of Address (of each type) and Site Information**

### 3.3.3 Common Solution Requirements

The following requirements are common to both the Buyer and Seller Emulators:

13. The solution should provide the following LSO Sonata release and feature support:
  - a) Support LSO Sonata SDK R4 as the base release for the initial release of the solution
  - b) Support planned LSO Sonata SDKs, including new feature per a roadmap to be agreed upon
  - c) Configurable to support pre-release APIs, including the ability to develop, load, configure and execute tests for those APIs
14. The solution should support the following flexible deployment options:
  - a) Portable, containerized solution
  - b) Default public cloud hosted deployment
  - c) Optional on-premise deployment (to overcome lab firewall issues)
15. The solution should include training, support and documentation

The solution could be delivered in a few ways, including by MEF as a service, by an independent vendor, or as open source. The commercial model, pros and cons of each option need to be evaluated.

## 4 How? Solution

This section defines the recommended abstract reference architecture and key operational flows / use-cases of the solution to address the requirements specified in the previous section. It is provided as guidance for design of the solution. It does not include detailed design descriptions or mandate the technologies or software components to be used to realize the solution. That is left to those tasked with developing actual solutions.

The solution description looks at the solution from two directions:

1. Testing a seller implementation using a Buyer Emulator
2. Testing a buyer implementation using a Seller Emulator

Buyer and Seller emulators are central to the solution. The solutions will differ depending on the direction because of the nature of the client/server relationship of the Buyer and Seller.

### 4.1 Testing a Seller Implementation Using a Buyer Emulator

The following high-level reference architecture diagram depicts the key components of the Buyer Emulator:

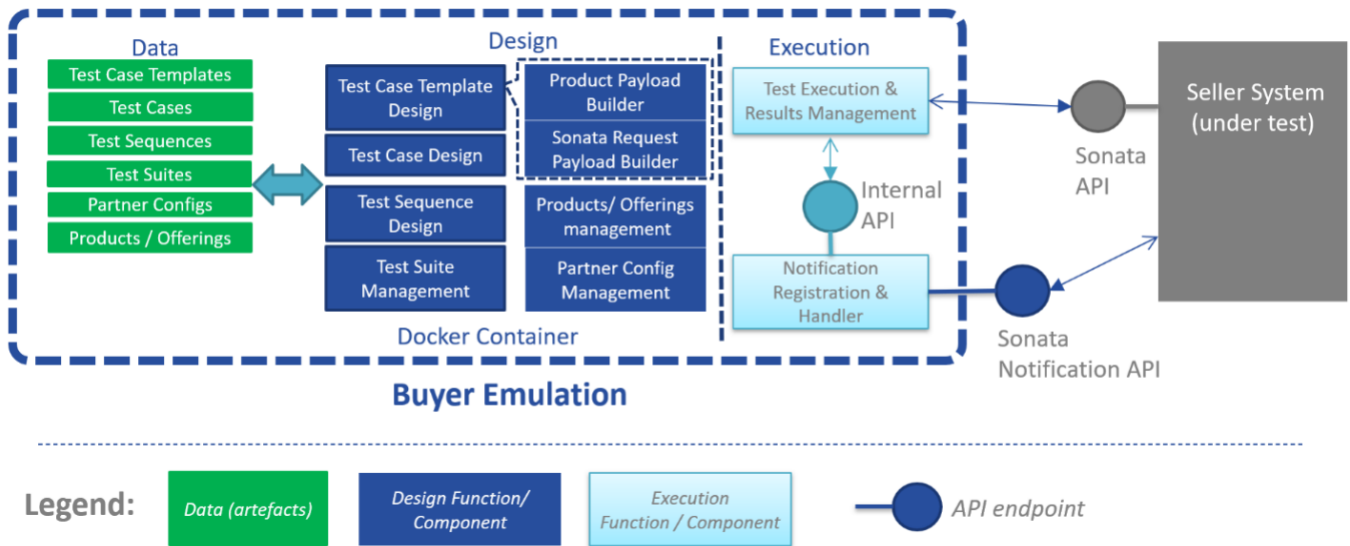


Figure 3 – Buyer Emulator High-Level Reference Architecture

The reference architecture is divided into design and execution components. Design consists of a set of software functions/components that are used to configure, manage, and generate executable tests. Execution consists of a set of components required to load and execute tests against Seller systems under test and manage results. Data represent the key data objects required by the solution.

The table below provides descriptions of the individual components and their roles.



<b>Test Case Template Design</b>	for creating and managing re-usable Test Case Templates
<b>Test Case Design</b>	for creating and managing executable Test Cases
<b>Test Sequence Design</b>	for creating and managing executable Test Sequences
<b>Test Suite Management</b>	for managing (create, read, update and delete) Test Suites
<b>Product Payload Builder</b>	for building product payload(s) (via a UI with input validation) required for use in Test Case Template involving product(s) e.g. PoQ, Quote, Order
<b>LSO Sonata Request Payload Builder</b>	for building LSO Sonata request payload for use in Test Case Templates
<b>Products / Offerings Management</b>	for managing (create, read, update and delete) specifications of Products and specific offerings to be tested with Sellers
<b>Partner Config Management</b>	for managing (create, read, update and delete) per partner configuration details required by the emulator to communicate with each partner
<b>Test Exec. &amp; Mgmt.</b>	for loading and executing Test Cases and Test Sequences against the Seller system under test, processing responses and storing results (based on an industry standard testing framework, e.g. Postman™)
<b>Notification Registration &amp; Handler</b>	for registering and handling events with/from the Seller system and for invoking test execution and management components to handle those events

Table 3 – Buyer Emulator Function/Component Descriptions

### 4.1.1 Main Buyer Emulator Operational Flow

The following are the high-level descriptions of the main operational flows for the Buyer Emulator. They are provided as guidance and a means to convey the intended operation to those tasked with detailed design of the solution.

#### 1. Buyer Emulator Design Operation Flows

The main functions of Design are to support the creation and management of Test Cases and Test Sequences, so they are ready for Execution. The diagram below depicts the high-level flow for designing a Test Case Template for an individual LSO Sonata function that involves a Product Payload.

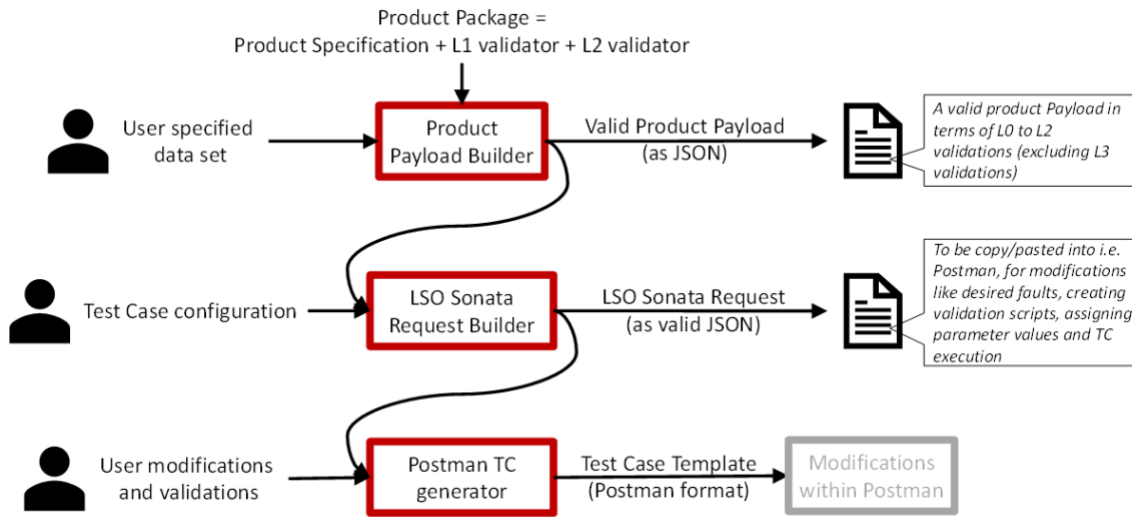


Figure 4 – Buyer Emulator Test Case Template Design Process

**Test Case Template Design** is a function of the solution that supports generation of a comprehensive set of Test Case Templates—one for each type of Test Case required to verify an LSO Sonata implementation (such as those specified in the MEF W 92.1 LSO Sonata Test Specification). It consists of a Product Payload Builder and the LSO Sonata Request Payload Builder as described below:

- **Product Payload Builder** is a comprehensive UI-driven component that supports loading of definitions of Products / Offering, application of validation rules (or ignore validation feedback when creating negative tests) and generation of Product Payload(s) for use in LSO Sonata Test Case Templates.
- **LSO Sonata Request Payload Builder** is a comprehensive UI-driven component that supports pre-built logic for generation of well-formed, compliant LSO Sonata Request Payloads comprised of the LSO Sonata Envelope, user-input and product payload (if applicable) for each LSO Sonata function. These payloads are then wrapped as Test Case Templates for use in the test tool.

The tables below describe the key Design flows for the Buyer Emulator in more detail, including Test Case Template Design, Test Case Design, and Test Sequence Design.

<b>Scope:</b>	<b>Design Test Case Template (re-usable)</b>
---------------	--

Pre-Conditions	<p>The LSO Sonata function and specialization of that function (<i>e.g., specific Product types/items for POQ or alternatives, such as formatted or geographic address for Address Validation</i>) you wish to create a Test Case Template for are known</p> <p>Product Payload Builder with pre-built support for building a parameterized test request product payload (logic and validation rules)*</p> <p>Test Case Template Designer with pre-built support for building a compliant parameterized test request payload for the specified LSO Sonata function (logic and validation rules)</p>
<b>Steps:</b>	
Pre-config steps	1) Product specification and validation rules for checking input data already configured in the Product Payload Builder*
Design steps	<p>1) Launch Buyer Emulator</p> <p>Build Product Payload (only applies for LSO Sonata functions where products need to be specified)*:</p> <ul style="list-style-type: none"> <li>a) Run Product Payload builder to capture product item details (multiple)</li> <li>b) Enter requested inputs (validated and iterative)</li> </ul> <p>Test Case Template Design</p> <ul style="list-style-type: none"> <li>c) Run LSO Sonata Payload Builder and select the LSO Sonata function that you wish to create the Test Case Template for</li> <li>d) Populate the requested inputs (for example, compliant with W 92.1 test-case definition), including pre-built Product Payloads where applicable</li> <li>e) Generate and save the Test Case Template (JSON formatted output)</li> </ul> <p>Switch to test tool (i.e. Postman), copy and wrap the JSON formatted Test Case Template payload name and save it so that it's ready to be executed in the test tool.</p>
Notes	<p><i>*Only applies to LSO Sonata APIs that involve a product payload, so does not apply to Address Validation or Site Query.</i></p> <p><i>The above describes the sequence for creating Test Case Templates that comply with the LSO Sonata, it includes support for generating both positive and negative Test Cases as specified in W 92.1</i></p>

<b>Scope:</b>	<b>Design Test Case</b>
---------------	-------------------------

Pre-Conditions	1) Test Case Template already exists and is configured in the testing tool (see Design Test Case Template above)
<b>Steps:</b>	
Pre-configuration steps	
Design steps	<p>1) Launch test tool (i.e. Postman)</p> <p>Select the appropriate Test Case Template in the test tool copy it and save it under a new name as a test case.</p> <p>Edit the copied Test Case and provide the necessary input values for the Test Case.</p> <p>Name and save the populated Test Case for later execution</p>

<b>Scope:</b>	<b>Design Test Sequence</b>
Pre-Conditions	<p>1) Test Cases to be used in the sequence already exist and are configured in the testing tool (see Design Test Case above)</p> <p>Sequence we wish to create is already known, including logic / rules for validating and handling outcome of each step, and transitioning to the next step</p>
<b>Steps:</b>	
Pre-configuration steps	
Design steps	<p>1) Launch test tool (i.e. Postman)</p> <p>Select the first Test Cases in the sequence in the test tool</p> <p>Add the validation and logic for the next step</p> <p>Go to #2 until all Test Sequence steps are defined.</p> <p>Name and Save the Test Sequence for later execution.</p>

**The following list of related operational flows are to be elaborated on in the detailed design of the solution:**

1. Design product specification
2. Manage partner-specific configuration
3. Configure notification registration and handling

## 2. Buyer Emulator Execution Flows

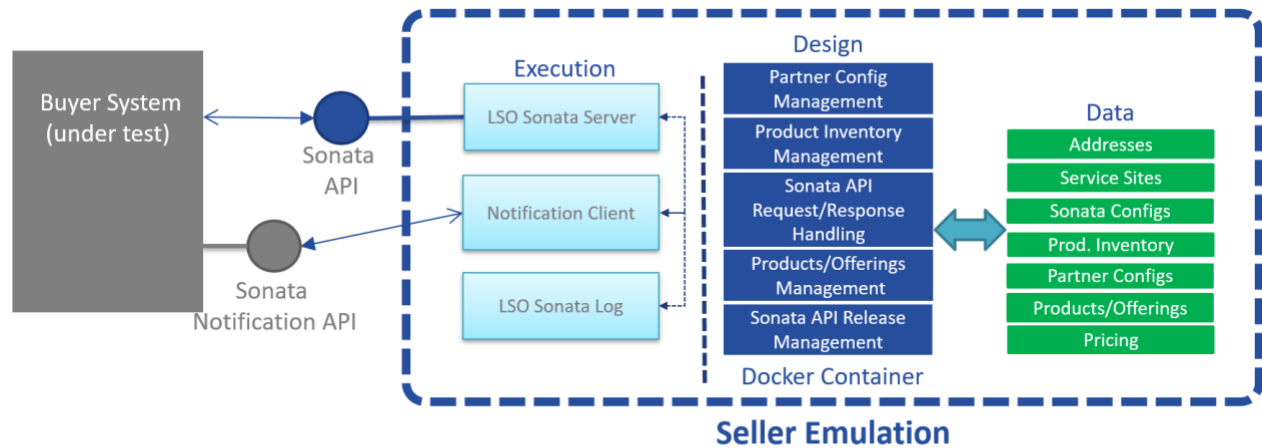
The main functions of Execution are to run the Test Cases and Test Sequences against the Seller systems and process / manage results. The tables below describe the steps involved for both.

<b>Scope:</b>	<b>Execute Test Case</b>
Pre-Conditions	<ol style="list-style-type: none"> <li>1) Test Case already designed (see Design Test Case above)</li> <li>2) Partner security credentials already configured in test tool and connectivity to Seller system is working</li> <li>3) Seller System is running and able to process the Test Case request</li> </ol>
<b>Steps:</b>	
Pre-Configuration Steps	
Execution Steps	<ol style="list-style-type: none"> <li>1) Launch Buyer Emulator</li> </ol> <p>Select Test Case and execute against the Seller system</p> <p>Receive responses and analyze them to see if they are as expected</p>

<b>Scope:</b>	<b>Execute Test Sequence</b>
Pre-Conditions	<ol style="list-style-type: none"> <li>1) Test Sequence already designed (see Design Test Sequence above)</li> <li>2) Partner security credentials already configured in test tool and connectivity to Seller system is working</li> <li>3) Seller System is running and able to process the Test Case request</li> </ol>
<b>Steps:</b>	
Pre-Configuration Steps	
Execution Steps	<ol style="list-style-type: none"> <li>1) Launch Buyer Emulator</li> <li>2) Select Test Case Sequence and execute against the Seller system</li> <li>3) Receive results and analyze them to see if they are as expected</li> </ol>

## 4.2 Testing a Buyer Implementation Using a Seller Emulator

The following high-level reference architecture diagram depicts the key components of the Seller Emulator. The Seller Emulator is expected to operate like a portable “Twin” of a provider’s LSO Sonata Seller system (much like a flight simulator in aviation). It must be possible to configure and calibrate the Seller Emulator to behave as an exact match of a provider’s Seller system. It will need to be configured with real Addresses and Product Inventory. Site Inventory could be real or fictitious examples and Pricing will almost certainly be fictitious examples only.



### Legend:



Figure 5 – Seller Emulator High-Level Reference Architecture

The reference architecture is divided into Design and Execution components. Design consists of a set of software functions/components used to configure how the Seller emulator handles requests from Buyer systems. Execution consists of a set of components required to load and execute the configuration for handling requests from Buyer Systems. Data represents the key data objects required by the solution.

The table below provides descriptions of the individual components and their roles.

<b>Partner Config Management:</b>	for managing (create, read, update and delete) per partner configuration details required by emulator to enable LSO Sonata communication and define Buyer partner handling
<b>Product Inventory Management:</b>	for managing (create, read, update and delete) product instances manually configured by the Seller (e.g., ENNIs)
<b>LSO Sonata API Request/Response Handling</b>	for generic (specific to a Seller but common to all Buyer partners) LSO Sonata request handling i.e., building a list of alternatives for APIs that declare such a capability
<b>Products / Offerings Management:</b>	for managing (create, read, update and delete) specifications of Products and specific offerings to be tested with Buyers
<b>LSO Sonata API Release Management:</b>	for managing (create, read, update and delete) LSO Sonata release settings and API endpoint configurations
<b>LSO Sonata Server:</b>	an LSO Sonata server reference implementation and associated API endpoints, configured with both: common and partner-specific data and handling (i.e. per partner-configured workflows), includes a database of registered notification listeners, and logic for building LSO Sonata notification messages
<b>Notification Client:</b>	API client capable of sending LSO Sonata notifications to the registered listeners (Hubs)
<b>LSO Sonata Log:</b>	for storing historical Buyer partner API interactions with tools for searching, filtering and viewing stored information

Table 4 – Seller Emulator Function/Component Descriptions

### 4.2.1 Main Seller Emulator Operational Flow

The tables below describe the high-level conceptual description of the main operational flows for the Seller Emulator. They are provided as guidance and a means to convey the intended operation to those tasked with detailed design of the solution.

#### 1. Seller Emulator Design Flows

<b>Scope:</b>	<b>Configure Emulator for Buyer Partner</b>
Pre-Conditions	<p>1) The target LSO Sonata Seller configuration (Addresses, Sites, Product Types, etc.) that is needed to support the Buyer test plans is known.</p> <p>The Seller emulator already supports the ability to be configured to support Seller-specific settings (e.g. products/offerings, LSO Sonata handling sync /async) and data-sets for agreed Test Cases and can emulate real Seller system behavior</p>
<b>Steps:</b>	
Pre-Configuration Steps	
Design Steps	<ol style="list-style-type: none"> <li>1) Launch Seller Emulator</li> <li>2) Import desired Product Definitions (specs), offering, and validation rules using Product Management component if/as needed to support target Test Cases</li> <li>3) Import and configure Addresses, Sites, Pricing (i.e. alternatives handling) if/as needed to support target Test Cases</li> <li>4) Configure expected Buyer-Seller pair-wise behavior for LSO Sonata Request/Response handling (like i.e. rules for building alternatives list, etc.)</li> <li>5) Create/configure Buyer partner and associated partner configurations (API details, LSO Sonata Release, available Product/Offerings)</li> <li>6) Manually Populate Product Inventory if/as needed to support target Test Cases (e.g. ENNIs)</li> </ol>

#### 2. Seller Emulator Execution Flows

<b>Scope:</b>	<b>Handle (Receive, Process and Respond) Buyer Requests</b>
Pre-	1) Seller emulator already configured for Buyer partner (see Configure



Conditions	emulator for Buyer partner above) and ready to receive requests
<b>Steps:</b>	
Pre-Configuration Steps	
Design Steps	<ol style="list-style-type: none"> <li>1) Initiate agreed requests from Buyer system to Seller emulator</li> <li>2) Seller system automatically processes Buyer request per pre-configured handling</li> <li>3) Verify that request is handled as expected</li> </ol>

## 5 Provider Benefits

- Enables faster and broader adoption of LSO Sonata among providers by streamlining the processes and tools for implementation, on-boarding and interop verification.
- Significant time and costs improvements for implementation, on-boarding and inter-op verification testing of LSO Sonata implementations resulting from a well-defined, standardized, industry-wide approach.
- Greatly reduce risk of failed or multi-cycle inter-provider testing, greater control when planning the number of, cost, and effort associated with on-boarding and verifying partner inter-op.
- High level of standard compliance and compatibility of provider LSO Sonata implementations by having all providers test against the same standardized, LSO Sonata reference implementation emulators.
- Immediate test solution availability, no or minimum lead time, reduced development and maintenance effort and cost versus each provider building their own test solution.
- Pseudo-qualification solution for partners requesting interoperability testing.
- Increased probability of successful certification.

## 6 About MEF

An industry association of 200+ member companies, MEF has introduced the MEF 3.0 transformational global services framework for defining, delivering, and certifying assured services orchestrated across a global ecosystem of automated networks. MEF 3.0 services are designed to provide an on-demand, cloud-centric experience with user- and application-directed control over network resources and service capabilities. MEF 3.0 services are delivered over automated, virtualized, and interconnected networks powered by LSO, SDN, and NFV. MEF produces service specifications, LSO frameworks, open LSO APIs, software-driven reference implementations, and certification programs. MEF 3.0 work will enable automated delivery of standardized Carrier Ethernet, Optical Transport, IP, SD-WAN, Security-as-a-Service, and other Layer 4-7 services across multiple provider networks. For more information, visit <https://www.mef.net/> and follow us on [LinkedIn](#) and Twitter [@MEF\\_Forum](#).

## 7 Acknowledgements

- Michael Kearns (Amartus) – Editor
- Konrad Herba (Amartus)
- Aidan Anderson (AT&T)
- Nico Walters (CMC Networks)
- Jacqui Teo (HGC)
- Laurent Sevette (Lumen)
- Jack Pugaczewski (Lumen)
- Daniel Bar-Lev (MEF)
- Divesh Gupta (PCCW Global)
- Daniele Mancuso (TI Sparkle)
- Mike Bencheck (Spirent)



MEF-2021038

© MEF Forum Solution Guide. Any reproduction of this document, or any portion thereof, shall contain the following statement: "Reproduced with permission of MEF Forum." No user of this document is authorized to modify any of the information contained herein.