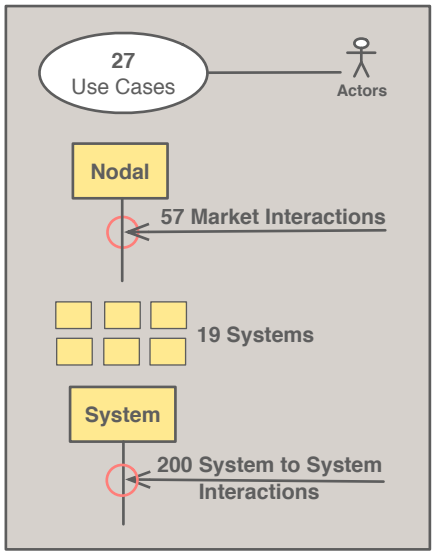
Understanding:
System of Systems Architecture

The System of Systems Architecture (SoSA) is a technique for modeling a complex system that is itself comprised of complex systems. The value of this approach comes from its ability to look at customers of Nodal and how they expect to operate in a Nodal Market. We believe this approach will provide TPTF and ERCOT a holistic view of Nodal and the means to ensure that the requirements of the individual projects effectively integrate to meet the needs of the Nodal Market.



System of Systems Statistics

What are the benefits?

The key benefits of the System of Systems Architecture approach are:

1. The use cases, requirements and interfaces defined by the projects can be validated against the Enterprise Level Use Cases.
2. Application interfaces can be clearly distinguished and described in terms of attributes and operations.
3. The System of Systems model can be used to derive end-to-end test cases for the Nodal program.
4. The Enterprise Level architecture provides ERCOT and TPTF with a context that can ensure complete coverage of the Nodal protocols by each individual system component.

Use Case Flow Down

The Nodal System of Systems model started with the ERCOT business architects deriving the top level services (**Use Cases**) that Nodal provides to the Market and other external entities (**Actors**) from the Protocols. This list of services were analyzed and refined by the business architects and the SoSA team. These service or use case names make up the Enterprise Level Use Cases included in the handout in the middle of this booklet.

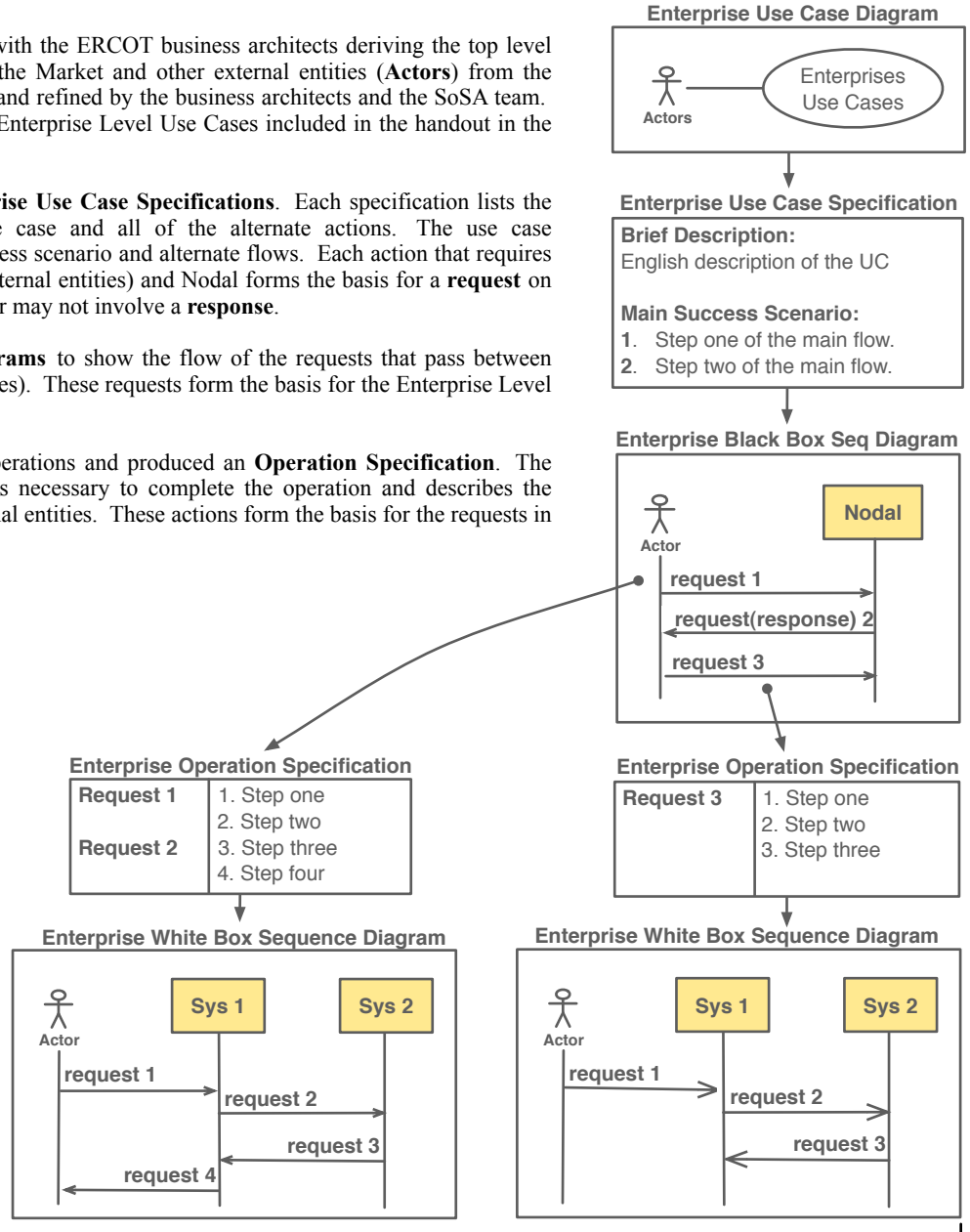
The project team then worked on the **Enterprise Use Case Specifications**. Each specification lists the major actions necessary to perform the use case and all of the alternate actions. The use case specifications are structured into the main success scenario and alternate flows. Each action that requires a interaction between the Market (and other external entities) and Nodal forms the basis for a **request** on the Black Box Sequence Diagram which may or may not involve a **response**.

Next we built the **Black Box Sequence Diagrams** to show the flow of the requests that pass between Nodal and the Market (and other external entities). These requests form the basis for the Enterprise Level operations.

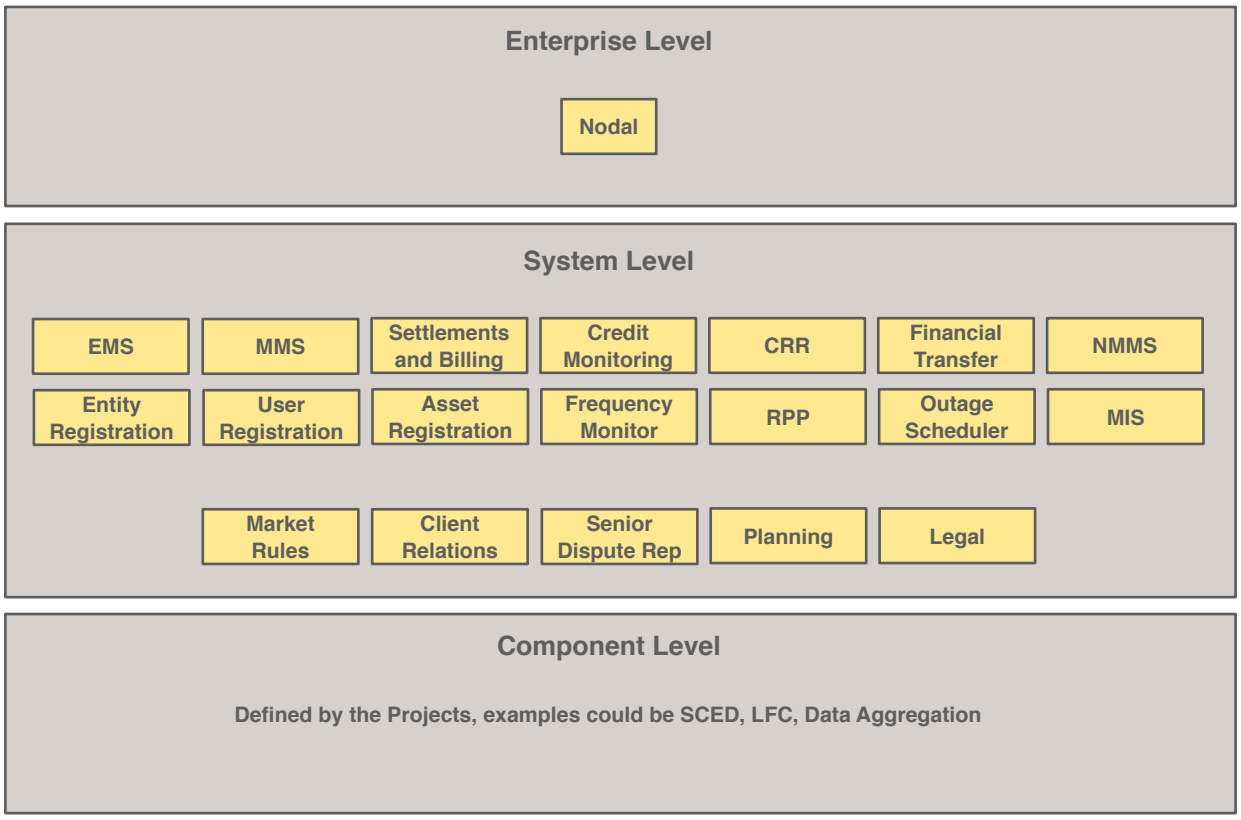
The team then took each Enterprise Level Operations and produced an **Operation Specification**. The Operation Specification documents the actions necessary to complete the operation and describes the interactions between the systems and the external entities. These actions form the basis for the requests in the White Box Sequence diagrams.

The last part of the Enterprise Level work is producing the **White Box Sequence Diagrams**. These diagrams depict the flow of requests between the systems, and between the systems and the external entities.

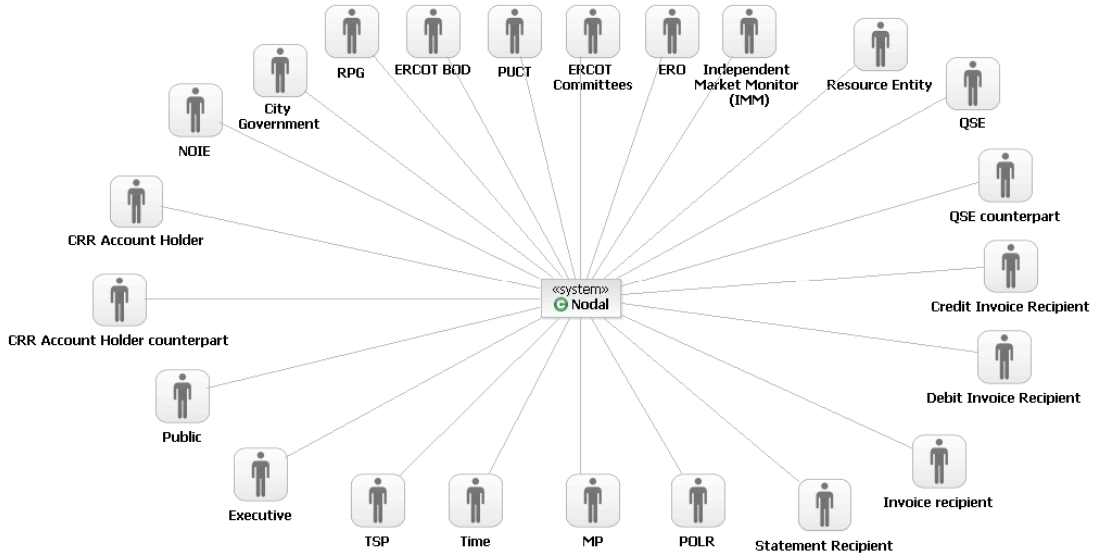
This process of decomposition using the Operations Specification and White Box Sequence diagram continues until the level of detail required to generate the solution is reached. For Nodal, we will complete the Enterprise Level White Box Sequence Diagrams and reconcile these with the use cases developed by the projects.



Levels in the Nodal System of Systems Model



Enterprise Level Actors



Enterprise Level Use Case Specification:
<Use Case Name>

Actor 1 Actor 2

Brief Description:
English description of the use case.

Pre-Condition:
The state the system should be in for this use case to start.

Main Success Scenario:
1. Step one of the main flow.
2. Step two of the main flow.

Anchored Alternative Flows
2.a. Alternative to step two in the main flow.
1. Step one of the alternative flow
2.a.1a. Alternative to step one in the alternative flow 2a.
1. Step one in the alternative flow

Unanchored Alternative Flows
a1. Alternative that applies at any step

Syntax of the Use Case Specification

What is the Use Case Specification?

The Use Case Specification describes the set of steps required to meet the objectives of the use case.

The Use Case Specification begins with the descriptive name of the use case starting with an action verb and then the object the action is applied to.

The **Main Success Scenario** describes the primary path through the use case, sometimes called the "happy path". The **Alternate Flows** describe the flow when a step in the Main Success Scenario either fails or meets a condition when an alternative is required.

The labeling convention for Alternative Flows uses the number of the step from the Main Success Scenario and appends a letter to indicate which alternate it is for that step. So Alternate Flow **2.a.** is the first alternate "**a**" for step 2 in the Main Success Scenario. So after completing step 1 in the Main Success Scenario, there are 2 possible paths to continue. The text part of the name reflects the condition when the flow would follow the alternate. You can also have Alternative Flows for steps in the Alternative Flows. **2.a.3.a** is the Alternative Flow for step 3 of the Alternative Flow **2.a**.

Enterprise Level Use Case Specification: Perform Day Ahead Financial Transfer

Perform Day Ahead Financial Transfer Use Case

Brief Description:
Perform Day Ahead Financial Transfer Use Case begins when the time arrives to generate the statements and invoices and concludes when the System has distributed credits.

Preconditions:
None

Post Conditions:
Settlement Statements and Invoices are posted.

Main Success Scenario
1. Time requests System generate Settlement Statements and Invoices
2. System begins DA Settlement batch processing
3. System prepares DA Settlement Statement, DA invoice, and calculates late fees.
4. System presents approved Settlement Statement and invoice to QSE.
5. Invoice Recipient pays DA invoice by due date.
6. System applies payments received to Invoice Recipients by 1700 of day received.
7. System sends out credits to invoice recipients by 1700 next business day.
8. Use case ends.

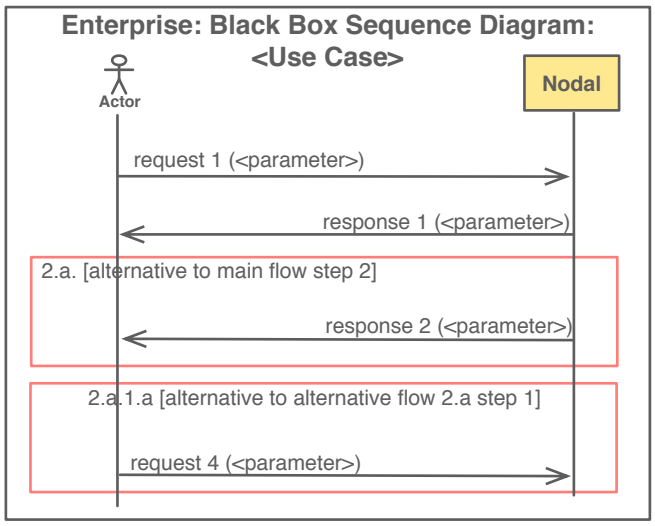
Alternate Flows

2.a. Create DA Late Fee Invoices on the 10th of each month (9.4.5)
1. Time requests System generate Late Fee Invoices
2. System aggregates late fees from previous month.
3. System presents validated, approved invoices to invoice recipients by 23:59:59
4. Invoice Recipient pays DA Late Fee invoice by due date.
5. System applies payments received to invoice recipients by 1700 of day received.
6. System sends out DA Late Fee credits to invoice recipients by 1700 next business day.
7. Return to Main Flow, Step 2.

3.a. System doesn't approve statements or invoices
1. System resolves issues
2. System prepares corrected Settlement Statement and invoice
3. Return to Main Flow, Step 3.

4.a. QSE doesn't pay by due date
1. System calculates short pay amount.
2. System notifies QSE of enforcement action.
3. Return to Main Flow, Step 6.

4.a.3.a. Enforcement action initiated (16.11.6.1)
1. System notifies QSE of increased financial security requirements.
2. Return to Main Flow, Step 6.



What is the Black Box Sequence Diagram?

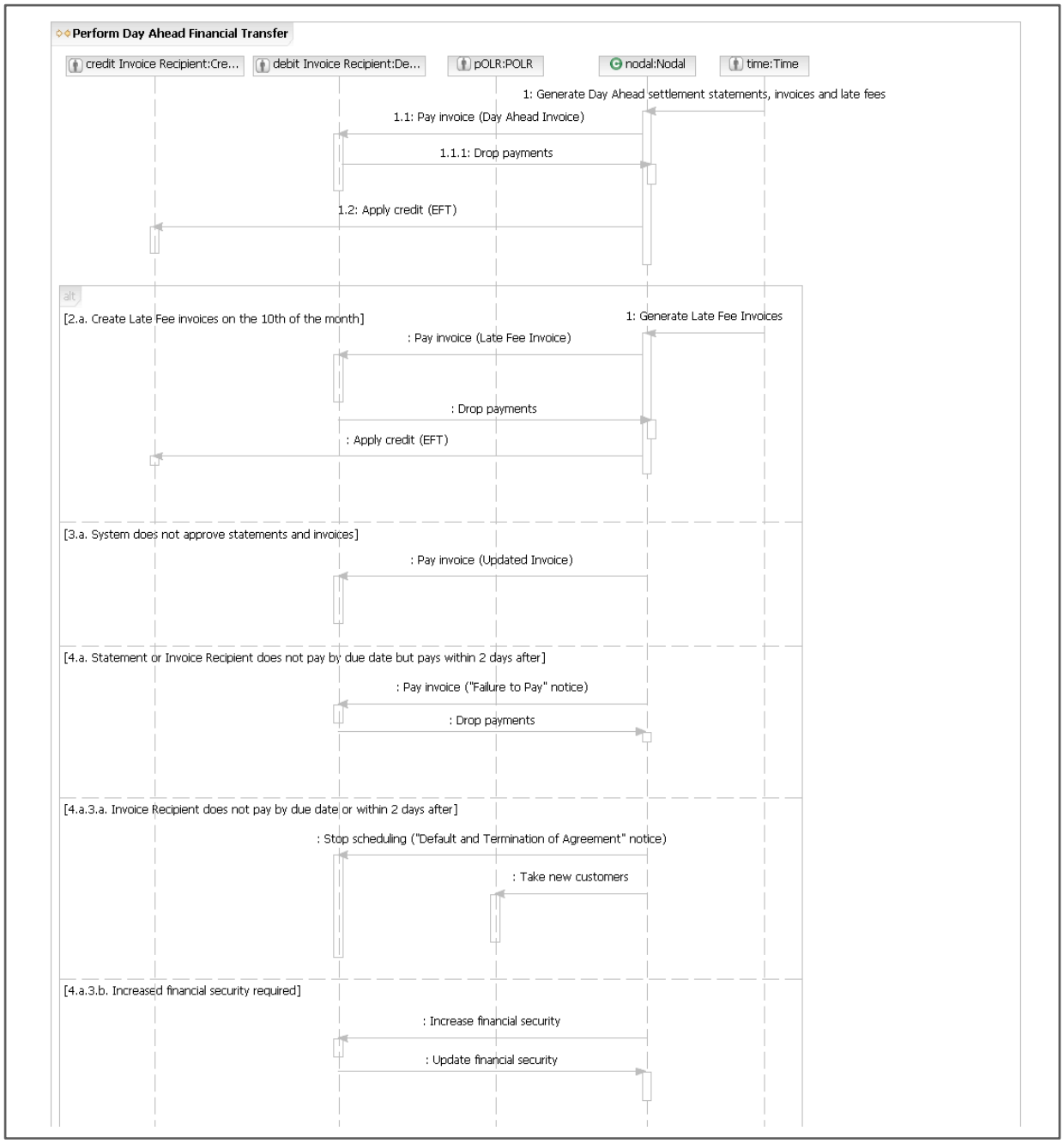
The Black Box Sequence Diagrams take the Use Case Specification and converts it to a series of **requests** between the entities mentioned in the specification. It covers the **Main Success Scenario** and all of the **Alternate Flows**.

The vertical lines ("Lifelines") each represent the different entities involved in the use case. The horizontal lines represent the requests from the initiator to the performer. For example, in the Black Box Sequence Diagram below, to read the first request you would say "*Time requests that Nodal generate settlement statements, invoices, and late fees*".

The Alternate Flows are contained in the box below the Main Success Scenario and marked with the condition when the alternative would be applied.

Syntax of the Black Box Sequence Diagram

Enterprise Level Black Box Sequence Diagram: Perform Day Ahead Financial Transfer

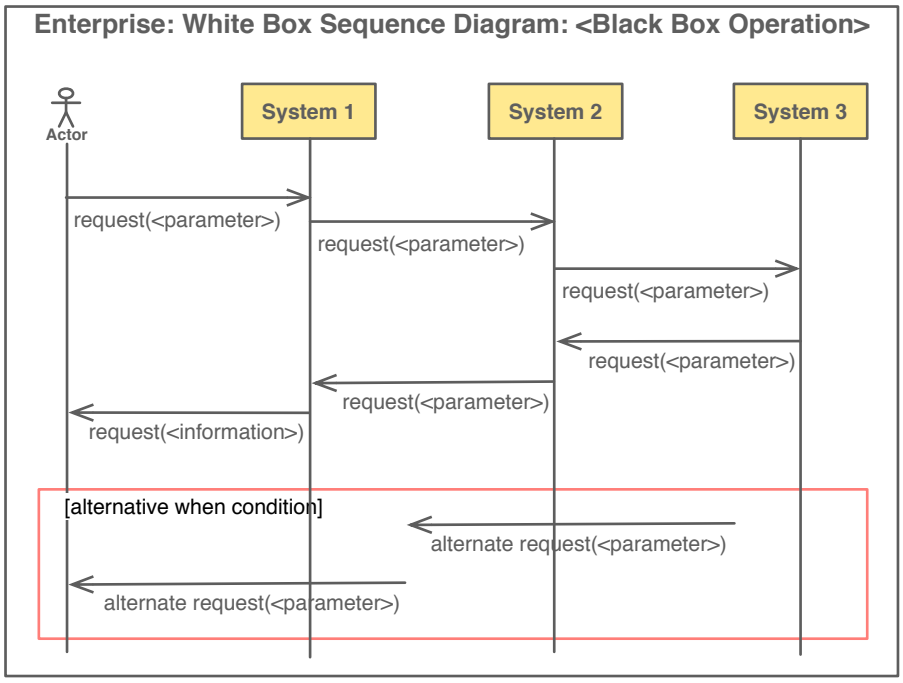


What is the Operation Specification?

The Operation Specification takes a request made of Nodal and breaks it down into the interactions between the system components at that level. There may be more than one Operation Specification for a use case. For the Nodal SoSA effort 27 use cases resulted in 57 Enterprise Operations and therefore 57 Operation Specifications. An operation may also be part of more than one use case. For example, the "Drop Payment" operation is part of all of the "Financial Transfer" use cases.

Enterprise Level Operation Specification: Drop Payment

Actor Action	Black Box Step	Step	Subsystem Action	White Box Budgeted Requirements	Locality	Process
Actor requests System Drop payment	System applies payments received to Invoice Recipients by 1700 of day received (verify with CM).	1.	Actor requests FT drop payments			
		2.	FT updates S&B	By 1700 of day received		
		3.	S&B calculates credits			
	System sends out credits to invoice recipients by 1700 next business day.	4.	FT requests recipients to "Apply Credit".	By 1700 of day after receipt		
[Invoice Recipient doesn't pay invoice by due date]						
	System sends "Failure to Pay" notice to Invoice Recipient.	1.	FT sends Late Pay info to S&B			
		2.	FT sends "Failure to Pay" notice to Invoice Recipient contact.			
[QSE doesn't pay by within 2 days after Due Date]						
	System sends "Default and Termination of Agreement" notice to QSE, QSE no longer in market.	1.	Legal sends "Default and Termination of Agreement" notice to QSE			
		2.	Legal notifies Registration that QSE is not in Market			
	System initiates drop to POLR process – notifies POLR QSE of customer transfer.	3.	Registration initiates drop to POLR			
	System transfers customers to POLR QSE	4.	Registration notifies POLR of switches			
[Increased Financial Security required]						
	System notifies QSE of increased financial security requirements..	1.	FT notifies QSE of increased financial security requirements			

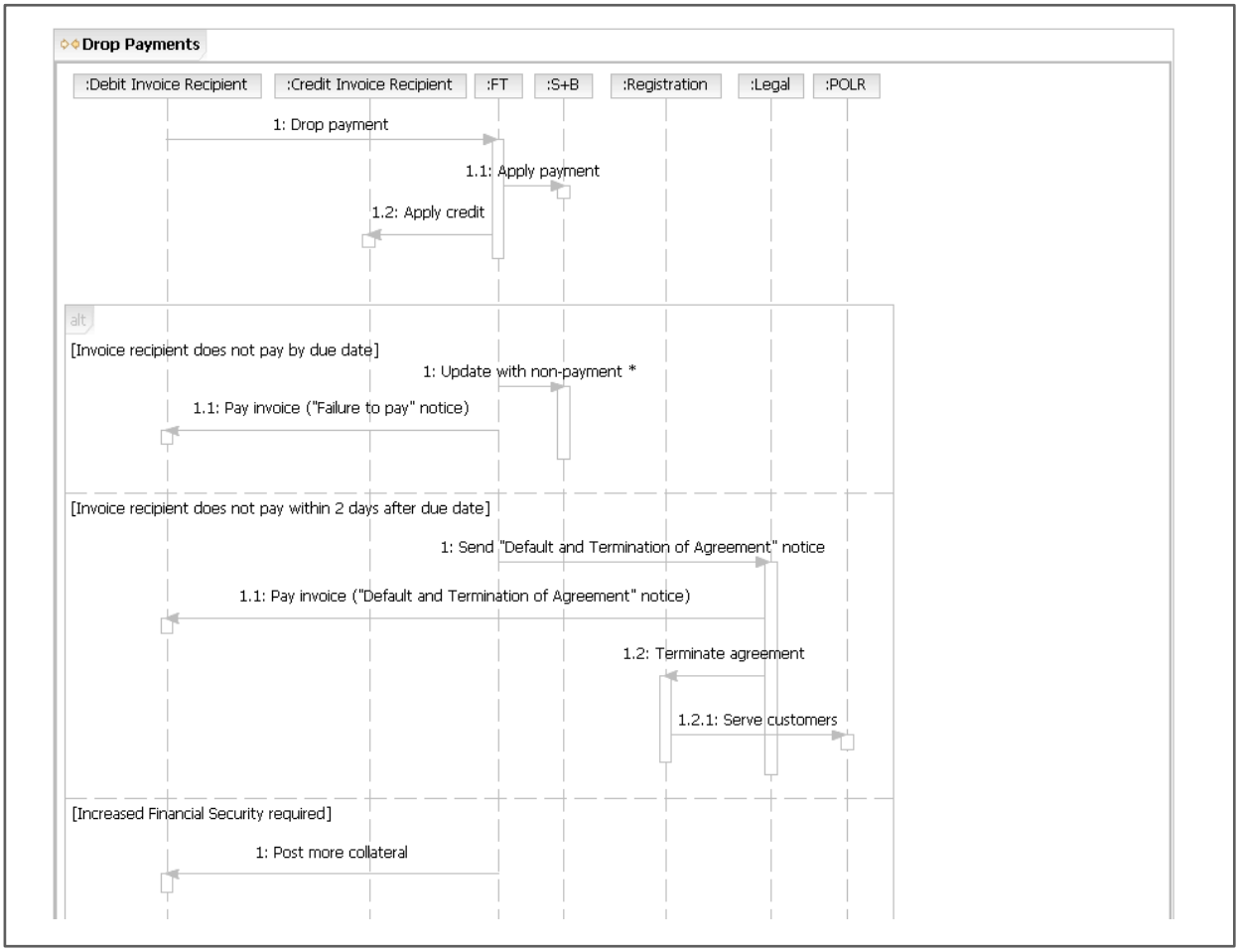


What is the White Box Sequence Diagram?

The White Box Sequence Diagram documents the flow of **requests** between **System** components and the external **Actor**. This flow shows the collaboration of Systems required to meet the objectives of the initiating request.The White Box Sequence Diagram is similar to the Black Box Sequence Diagram but instead of being based on the Use Case Specification, it is based on the Operation Specification.

Syntax of the White Box Sequence Diagram

Enterprise Level White Box Sequence Diagram: Drop Payment



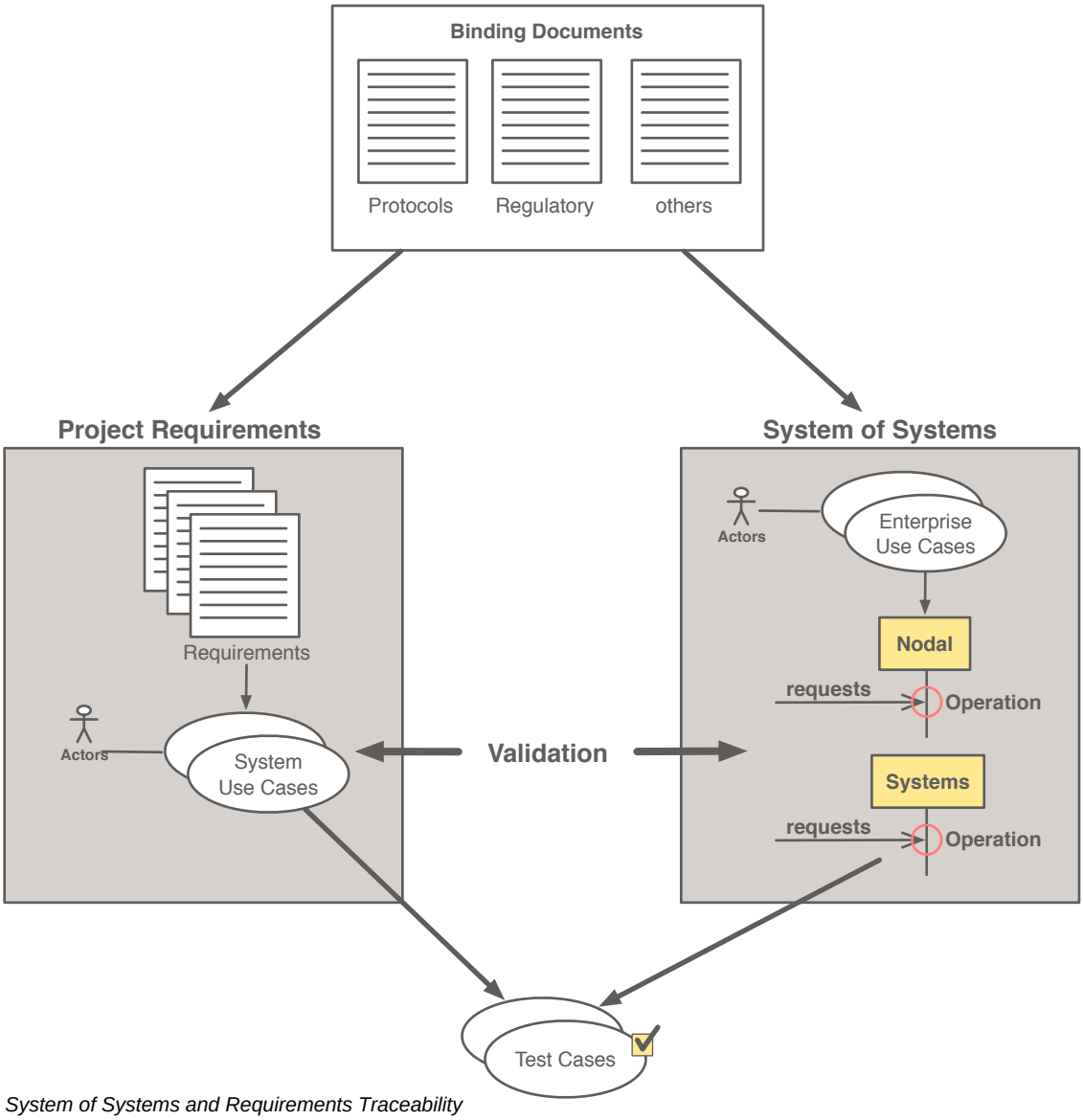
Validating System Requirements Using SoSA

Nodal requirements are produced at the component and system levels by analyzing the protocols and other binding documents. These requirements are then expressed as use cases, and these uses cases are then tested using test cases derived from the requirements and corresponding use cases. These requirements, uses cases and test cases express requirements at the System Level, for example EMS, MMS etc.

The System of Systems approach begins by created use cases at the Nodal Enterprise Level. These complete uses are then broken down into requests between the market and Nodal (Black Box Operations) and requests between systems inside the Nodal (white box operations). These black box and white box operations describe the interactions required to provide a complete solution for an end-to-end usage of Nodal by the market.

These operations are then verified with the projects' Uses Cases to ensure that all request have been covered and all uses of the Nodal system have been captured in the SoSA Enterprise Level Use Cases.

This verification between the individual projects requirements and use cases, and the System of Systems model is one of the key benefits of the System of Systems approach.

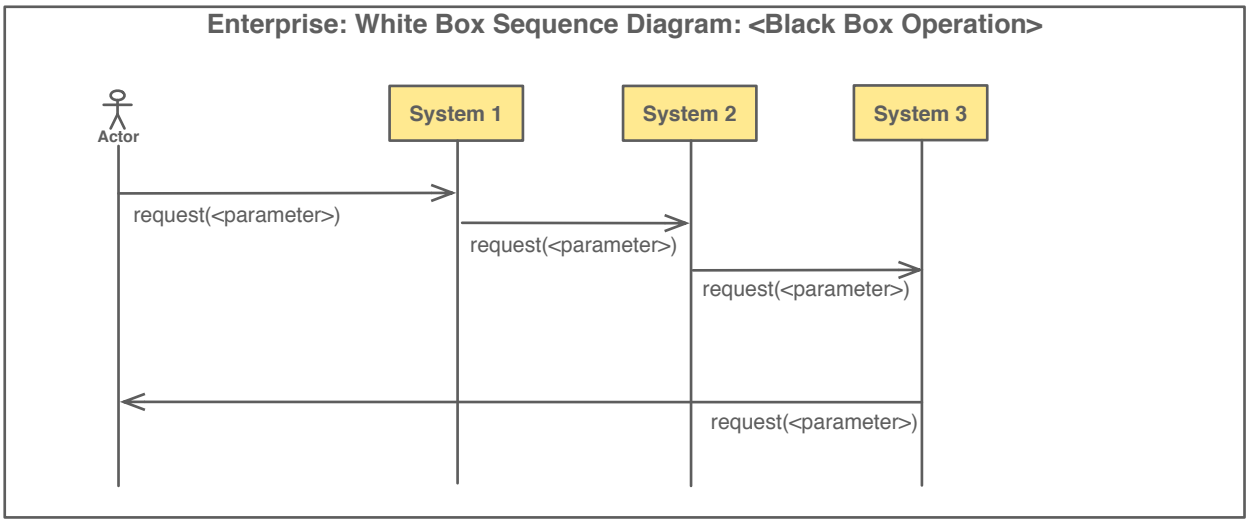


System of Systems and Requirements Traceability

Modeling integration in the System of Systems Architecture

An important benefit of the System of Systems model is its ability to identify and document interactions between the system components and the interactions with the Market. As part of the next iteration of the System of Systems model we will be modeling the integration approach for each System-to-System and Market-to-System interaction. This will be done by adding the Integration Layer as a System in the SoSA model.

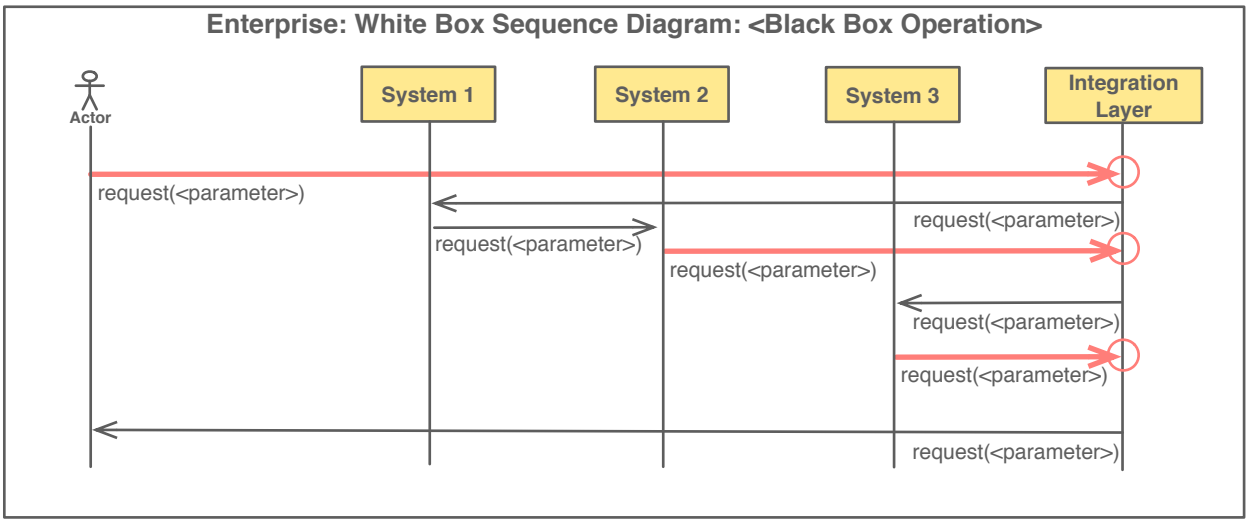
The example below shows how the Integration Layer can be added into the White Box Sequence diagram to show which requests flow through the integration layer. Each operation on the Integration Layer then becomes a requirement for the integration team. Each operation is then document by the Integration Team showing the transformation, auditing, security, reliability, monitoring and management decisions made for this particular integration.



White box sequence diagram without the integration layer

The type of requests between internal and external systems will determine the scope of the Integration Layer. The integration team will use System of Systems as a means to identify the requests that will communicate point to point (for example ICCP) and requests that communicate through the Integration Layer.

The diagram below shows the effect of adding the Integration Layer to the design of the Nodal systems. You can now see that System 1 and System 2 communicate point to point but all other systems, in this example, communicate through the Integration Layer.



White box sequence diagram with the Integration layer added