InfoGlobalTech

Innovative Solutions for your Enterprise

JMMB Apache Service Mix ESB Upgrade –

Technical Reference Document

Author: IGT InfoGlobalTech • Version 1.0

Revision History

NOTE: The revision history cycle begins once changes or enhancements are requested after the initial version of the document has been completed.

Date	Version	Description	Author	Comments
02-02-2024	1.0	Template	Sahil D	Draft version

Contents

Revision History 2

1. Introduction 4

2. Compatible bundles 5

2.1 Challenges identified by upgrading incrementally, (4-5; 5-6; 6-7) 5

2.2 Supported database or business systems for 'Application' components 5

2.3 Messaging brokers support by 'Application' messaging components 5

2.4 Supported database for profiling data 5

3. Architectural Design 6

3.1 Architectural Design Overview 7

3.2 Functional architecture 7

3.2.1 ESB Cloud Functional Architecture 8

3.2.2 ESB Data Integration Functional Architecture 8

3.2.3 ESB Functional Architecture 8

3.2.4 ESB MDM functional Architecture 8

4. Architectural Decisions 9

5. Interfaces and deployment integration 9

6. Data Integration and Data Services 9

6.1 Designing Jobs and Routes 9

6.2 Designing Services 9

6.3 API Integration 9

6.4 Mapping Data Flow 9

6.5 Change Data Capture 9

6.6 Database support for CDC 9

7. Quality Attributes 10

8. Testing 11

9.5.1 Functional testing and report 13

9.5.2 Performance testing and report 13

9.5.3 Integration testing and report 13

10. Configurations and Custom components 13

11. Reference 13

12. Appendix 13

1. Introduction

(Points to cover)
-Purpose of document;
- Software System's scope and context
- Market research (vendor evaluation and cost-benefit analysis) justify why version update is required.
- Introduction to Bundles (Enterprise Service Bus)
- JMMB Enterprise Service Bus concept overview
- JMMB Enterprise Service Bus Integration features
- Connectivity & Document Transformation
- Content-Based and Header Based Routing
- Clarity on business objective of project

(Keywords/sentences)
- Customer market needs demand software interrelations between the functions
- Architecture that realizes these functions
- Sentence to keep - The requirements for 'Such' systems are defined at different level of granularity and details from high level user requirements to detailed technical requirements

(Possible sub-headings)

2. Compatible bundles

(Possible sub-headings)

2.1 Challenges identified by upgrading incrementally, (4-5; 5-6; 6-7)

(points to cover)
version release notes
feature set of application consistency for all versions of Apache ServiceMix

2.2 Supported database or business systems for 'Application' components

(points to cover)
Maybe we can show input-output comparison of dataset for both versions

2.3 Messaging brokers support by 'Application' messaging components

2.4 Supported database for profiling data

2.5 Bundle ESB Setup (basic steps) documentation in UAT and QA environment.

3. Architectural Design

(Points to cover)
- A high/low-level overview of the system's architecture
- System objectives and Requirements
    > Assessment of current and future needs
    > Hardware requirements (CPU, memory storage, network bandwidth)
    > Scalibility and Flexibility
    > Redundancy and reliability
    > Virtualization and Containerization
    > Security
    > Energy Efficiency and Coolig
    > Cost
    > Vendor support and warrant
    > Compatibility with existing infrastructure
- System stakeholders and their roles

(keywords/ sentences)
Architectural Design aspects considered while documenting Architectural design,
1) Hierarchical decomposition (Abstraction layers)
2) Separation of concern - The problem of decomposing the individual sub-systems into hardware and software
3) Categorization of requirements
4) Requirement Stability
5) Traceability & Rationales - The strong mutual influence, between requirements and architectural artefacts demands that a development method supports highly co-development of requirement and architectural artefacts.
6) The choice of specific design of architectural solution is typically influenced by quality properties such as performance, accuracy, safety, reliability, physical properties such as safety, reliability, physical properties such size weight and energy consumption, or economic properties such as production cost.

Layer	Goals	Scenarios	Solution oriented requirements	Structural architecture
System Layer (L1)	System Goals	Interaction between the system and external actors	Required function, data behaviour, and quality of the system	System Interfaces
Functional Decomposition (L2)	Goals for each individual functional component	Interaction between the functional component of system and the environment	Required function, data behaviour, and quality for each functional component	Functional decomposition of the system including logical building blocks, interfaces and logical connection
Hardware/Software portioning (L3)	Goals for each individual SW component	Interaction between software and hardware components and the environment (independent of the deployment)	Required functions, data, behavior, and quality for each SW component	Decomposition of the system into hardware and software components
Deployment (L4)	Deployment goals	Platform specific interactions	Platform specific functions, data behavior and quality for each software component	Mapping of the hardware and the software components onto a specific hardware platform

Table 1: COSMOD-Re artefact template - Layered approach to architecture design documentation

(Possible sub-headings)

3.1 Architectural Design Overview

(Points to cover)
- OSGi Framework Updates:
- Java version compatibility
- Components upgrades: (like ActiveMQ, Camel, CXF, etc.,)
- API Changes
- New features and tools
- Removed Deprecated Features
- Integration and Compatibility with other systems

3.2 Functional architecture

(Points to cover)
- Diagram of logical system architecture
- Diagrams if sub-systems and components illustrating system structure and integration
- Diagram of hardware and software configuration for deployment
- Description for diagrams

(Possible sub-headings)

3.2.1 ESB Cloud Functional Architecture

3.2.2 ESB Data Integration Functional Architecture

3.2.3 ESB Functional Architecture

3.2.4 ESB MDM functional Architecture

(Points for us to consider)
- Detailed requirements should not hide the implicit and explicit design decisions and assumptions made by stakeholders during the definition of detailed requirement.
- Each abstraction layer includes hierarchy of abstraction layer. The architecture must satisfy the requirements defined at this layer.
- In addition, the requirements defined at lower abstraction must satisfy the requirements defined at higher abstraction layer. The architecture at lower abstraction layer satisfies the architecture defined at higher abstraction layer.

4. Architectural Decisions

(Points to cover)
- Critical Architectural decisions and their rationale
- Design patterns, architectural styles, and frameworks
- Justification for significant trade-offs and constraints
- System components (Hardware requirements)
- Changes in OSGI container, integration patterns, and components

(Keywords/sentences)
There are two most prominent challenges faced when taking Architectural Decisions,
Challenge 1: To clearly demonstrate hierarchy of clearly defined abstraction layer
Challenge 2: Dealing mutual influences between requirements and architectural artefacts

(Possible sub-headings)

5. Interfaces and deployment integration

(Points to cover)
- APIs and external system Interfaces
- Communication protocols and message formats
- Software architecture patterns and criteria for integration
- Deployment and Monitoring tool (was there need to update when updating application for version 4-7)
- Deployment environment description (e.g., on-premises or cloud-based)
- Hardware and software specifications
- Network configurations and deployment procedures

6. Data Integration and Data Services

(Points to cover)
Database Schema architecture
Data access and storage mechanisms
Handling of data security, integrity, and privacy

(Possible sub-headings)

6.1 Designing Jobs and Routes

6.2 Designing Services

6.3 API Integration

Points to cover
- API Integration Overview
- Using meta data for API Specifications
- Using REST/SOAP meta data with jobs & routes

6.4 Mapping Data Flow

6.5 Change Data Capture

(points to cover)
- CDC Overview
- CDC Subscribe/Publish principle
- Setting up CDC environment

6.6 Database support for CDC

7. Quality Attributes

(Points to cover)
- Performance and optimization strategies
- Capacity Planning and scalability
- Access controls and security measures
- Extensibility and maintainability guidelines

(Possible sub-headings)

7.1 Managing Metadata

(Points to cover)
- Evaluation Dependency Changes – Which integration are not compatible with newer version of Apache ServiceMix 7

8. Testing

(Points to cover)
- Testing Strategy
- Performance and stress testing – Current load, network traffic, data processing volume, user concurrency, time taken
- Documentation of Critical test scenarios for architecture evaluation

Use Case [ID]	[Short name of the Use Case]	
Abstraction Layer	[Name the abstraction layer at which use case is defined]	
Scope	[Name the system or component whose interactions are documented in this use case]	
Primary Actor	[Name of the actor whose goal(s) shall be satisfied by executing this use case]	
Secondary Actor	[Other actors i.e persons, systems, components, devices etc. involved in this use case]	
Input and output variables	✓ Input: [Name of external variables whose values are measured or read during the execution of this use case] ✓ Output: [Names of external variables whose values are changed/ influeced during execution of this use case]	
Related Goals	[References to the goals that shall be satisfied by executing this use case.]	
Pre conditions	[Conditions on the system/ component or its environment that are guaranteed to hold before the use case has been completed successfully]	
Success Guarantee	[Conditions on the system/ component or its environment that are guaranteed to hold after the use case has been completed successfully]	
Minimal Guarantee	Conditions on the system / component or its environment that are guaranteed to hold after any execution of this use case, both, in case of success and in case of failure]	
Extension points	[Location, ie steps of the use case scenarios, at which this use case is extended by other use cases or includes other use cases.]	
Trigger	[Event that initiates the use case, e.g., a system event time event, or external event.]	
Main Scenario	Step	Action
	---	---
Alternate Scenario	Step	Action
	---	---
Failure Scenario	Step	Action
	---	---
Technology and data variation	[Description of realisation alternatives for performing individual steps of the use case scenario	
Special requirements	[Description of quality requirements and constraints for individual steps of the use case scenarios or the entire use case]	

Table 2: Use case template

(Possible sub-headings)

8.1 Disaster recovery plan
8.2 Post implementation review
8.3 Complaince and Security Review
8.4 Risk Assessment in UAT and QA environment
8.5 Concurrent Testing
8.6 Testing Overview

(Possible sub-headings)

Goal [ID]	[Short name of the goal]
Abstraction layer	[Name of the abstraction layer at which goal is defined[
Scope	Name of the system or component for which goal has been defined
Goal Category	[Soft Goals or hard goals]
Goal Description	[Precise description of the goal in natural language]
Goal responsibility	Name of the agent for satisfying this goal
Super-goals	References of goals that are refined by this goal and the type of refinement
Conflicting goals	References of goals that are in conflict with this goal
Related documents	References of scenarios documenting the satisfaction of this goal

Table 3: Goal Template

8.6.1 Functional testing and report

8.6.2 Performance testing and report

8.6.3 Integration testing and report

9. Configurations Parameters and Custom components

(Points to cover)
- Stats & Logs
- The custom components and configuration that were needed to re- build for Service Mix 4 upgrade
- System updates and software maintenance services
- Issue identified
- Support and troubleshooting instructions
- Configuration Details for sub-ordinate (internal/external system)

(Possible sub-headings)

10. Reference

11. Appendix

(Possible Sub-headings)