Anypoint Platform Architecture: Integration Solutions

Outline

PART 1: Architecting and Designing Integration Solutions

Module 1: Introducing Integration Solution Architectures

- Describe the objectives of enterprise integration solution
- Summarize how to architect for success with Anypoint Platform
- Describe how integration solutions using Anypoint Platform are documented
- Start using an architecture template for the course case study

Module 2: Identifying Anypoint Platform Components and Capabilities

- Identify and document the overall design intentions of Anypoint Platform
- Summarize Anypoint Platform capabilities and high-level components
- Identify and distinguish between Anypoint Platform infrastructure and deployment options
- Choose Anypoint Platform components to be used to meet the functional and non-functional requirements of an integration use case

Module 3: Designing Integration Solutions using Mule Applications

- Explain the typical uses of Mule components (including connectors, transformers, routers, and error handlers) in Mule applications
- Describe the differences between Mule 4 and Mule 3 applications
- Choose appropriate Mule application components to design an integration use case
- Summarize how class loading isolation is implemented in Mule runtimes

Module 4: Choosing Appropriate Mule 4 Event Processing Models

- Distinguish between Mule 4 blocking, non-blocking, parallel, and reactive event processing options
- Identify the event processing models used in various Mule 4 scopes and components
- Describe Mule 4 streaming options and behaviors
- Describe the event processing options for JMS and VM connectors
- Select appropriate event processing for an integration use case
- Design batch, near real-time, and real-time data synchronization integration use cases

Module 5: Choosing Appropriate Message Transformation and Routing Patterns

- Describe reusable ways to transform and process events
- Explain how to simplify and decouple complex data mappings using common data models
- Design transformations between data models
- Choose the best event transformation, data validation, and event routing patterns to an integration use case

Module 6: Designing Mule Application Testing Strategies

- Describe possible testing strategies for Mule applications
- Document a testing strategy for an integration use case
- Define the types of MUnit tests required for an integration use case and document the code coverage
- Design integration and performance tests for Mule applications

PART 2: Operationalizing Integration Solutions
Module 7: Choosing and Developing a Deployment Strategy

- Distinguish between various Anypoint Platform runtime service models
- Distinguish between various Anypoint Platform deployment models
- Select the best deployment and runtime service options for an integration use case

Module 8: Designing with Appropriate State Preservation and Management Options

- Decide the best way to store Mule application state in persistent or non-persistent storage
- Explain how to store Mule application state using Object Store v2
- Decide when to manage storage of Mule application state using persistent VM queues
- Decide when to use Mule application provided caches to store Mule application state
- Design an integration solution to avoid duplicate processing of previous records using Mule connector watermarks
- Design the best storage and state management options for an integration use case

Module 9: Designing Effective Logging and Monitoring

- Describe auditing and logging options for Mule applications
- Design logging strategies for Mule applications
- Choose logging policies for Mule application log files
- Describe integration options with third-party log management systems
- Specify Anypoint Platform monitoring, alerting, notification, visualization, and reporting options for APIs and integration solutions
- Choose the best monitoring, alerting, and notification options for an integration use case

Module 10: Designing an Efficient and Automated Software Development Lifecycle

- Design to manage property files for Mule applications across different environments
- Design to manage Anypoint Platform environments for Mule application deployments
- Describe how to implement continuous integration and continuous delivery (CI/CD) for an organization
- Describe how to automate deployment and management with Anypoint Platform

PART 3: Designing Strategies to Meet Non-Functional Requirements

Module 11: Designing Transaction Management in Mule Applications

- Identify why and when transactions are supported in Mule applications
- Identify resources that participate in transactions in Mule applications
- Describe how to manage a transaction using a transaction manager or the Saga pattern
- Describe how to demarcate transaction boundaries in Mule applications
- Choose the correct transaction type based on the participating resources

Module 12: Designing for Reliability Goals

- Identify and distinguish between reliability patterns for Mule application and their deployments
- Design to effectively balance competing non-functional requirements
- Clarify and validate reliability goals for an integration use case
- Design Mule applications and their deployments to meet reliability goals
- Design to effectively balance reliability goals with other project goals and requirements

Module 13: Designing for High Availability Goals

- Identify various types of high availability (HA) goals for Mule applications
- Identify ways to achieve HA in CloudHub and on-premises deployments
- Identify HA aware connectors and their design tradeoffs
- Describe how clustering and load balancing work in CloudHub and on-premises deployments
- Design to effectively balance HA goals with other project goals and requirements

Module 14: Optimizing the Performance of Deployed Mule Applications

- Clarify performance goals for Mule applications
- Identify the need for performance optimization and associated tradeoffs
- Describe ways to search for and locate performance bottlenecks
- Describe how to design, architect, design, and implement for performance
- Describe ways to measure performance
- Describe methods and best practices to performance tune Mule applications and Mule runtimes
- Design to effectively balance performance goals with reliability and HA goals

Module 15: Designing Secure Mule Applications and Deployments

- Describe Anypoint Platform security concepts and options
- Explain how to secure APIs on Anypoint Platform
- Differentiate between MuleSoft and customer responsibilities related to Anypoint Platform security
- Evaluate security risks for Mule applications
- Describe how to secure Mule application properties and data in transit

Module 16: Designing Secure Network Communications between Mule Applications

- Describe Anypoint Platform network security options and architectures
- Identify MuleSoft-owned and customer-owned roles and responsibilities related to Anypoint Platform network security
- Describe how to secure Mule applications using Java key stores
- Design TLS communication and other network security options for an integration use case
- Properly size an Anypoint VPC to support deployment of all expected Mule applications

Module 17: Putting It All Together

- Review the essential steps for designing an integration solution using Anypoint Platform and Mule applications
- Carry out all the steps to design an integration solution for an integration use case

Exam Details

This instructor-led course teaches experienced solution and technical architects or lead/senior developers how to design integration solutions using Anypoint Platform that meet functional and nonfunctional requirements. It builds on prior experience developing and deploying Mule applications, and gives concrete guidance on how to design integration solutions for the Mule runtime (both Mule 4 and Mule 3) in the various deployment options supported by the Anypoint Platform. It is case-study driven, with the solution architecture and effect on the organization's overall architecture being documented as the course progresses. This course prepares students to take the MuleSoft Certified Integration Architect - Level 1 certification exam and includes a voucher for two exam attempts. Note: This course is the replacement for the deprecated Anypoint Platform Architecture: Solution Design course.

Duration

5 days in-person or online

Objectives

At the end of this course, students should be able to:

- Translate functional and non-functional requirements into well documented integration interfaces and implementation designs.
- Select the best Mule components and patterns for integration solution designs.
- Select the deployment approach and configuration of Anypoint Platform with any of the available deployment options (MuleSoft-hosted or customer-hosted control plane and runtime plane).
- Design Mule applications for the various deployment options of Anypoint Platform runtime plane.
- Apply standard development methods covering the full development lifecycle to ensure solution quality.
- Design reusable assets, components, standards, frameworks, and processes to support and facilitate API and integration projects.
- Design and be responsible for the technical quality, governance (ensuring compliance), and operationalization of integration solutions.
- Advise technical teams on performance, scalability, reliability, monitoring, and other operational concerns of integration solutions on Anypoint Platform.

Audience

Solution and technical architects or lead/senior developers that are focused on designing enterprise integration solutions and have prior experience developing and deploying non-trivial Mule applications

Prerequisites

Experience developing and deploying Mule applications as demonstrated by one of the following:

- Passing of the MuleSoft Certified Developer – Level 1 (Mule 4) or MCD - Integration and API Associate (Mule 3) exam
- Completion of the instructor-led Anypoint Platform Development: Fundamentals (Mule 4 or Mule 3) course or the self-study MuleSoft.U Development Fundamentals (Mule 4 or Mule 3) course

Other development and architecture knowledge and experience including:

- Proficiency in any JVM-based programming language with ability to read procedural, object-oriented, and (ideally) functional code
- Familiarity with threads, thread pools, locks, server/client sockets, JDBC data sources, and connection pools on the JVM
- Proficiency with current software development tools like Git/GitHub, Maven, Jenkins, or similar
- Experience as an architect or lead/senior developer on at least one integration project using any technology stack
- A full understanding of the fundamental ingredients of enterprise integration including interface definitions and contracts; data encoding using XML or JSON; REST APIs or SOAP web services; SQL or NoSQL database access; message-passing using JMS, AMQP or similar; network protocols like TCP/IP, HTTP and HTTPS; single-resource transactions
- Familiarity with basic security concepts including certificates and encryption at rest and in transit