WSO2CON 2024

WSO2 MI for Enterprise Integration in a Decentralized, Microservice Oriented Landscape



Isuru Udana
Director - Engineering
WSO2



Shazni Nazeer
Director - Solutions Architecture
WSO2

Agenda

- Introduction to Microservices and decentralized architectures
- WSO2 MI overview and features
- Designing microservice-friendly integrations
- Implementation with WSO2 Micro Integrator

Microservices & Decentralized Architecture

Microservices Architecture

- *“Microservices architecture is about building a software application as a collection of independent, autonomous (developed, deployed and scaled independently), business capability oriented and loosely coupled services.”*

  Source: Microservices for Enterprise

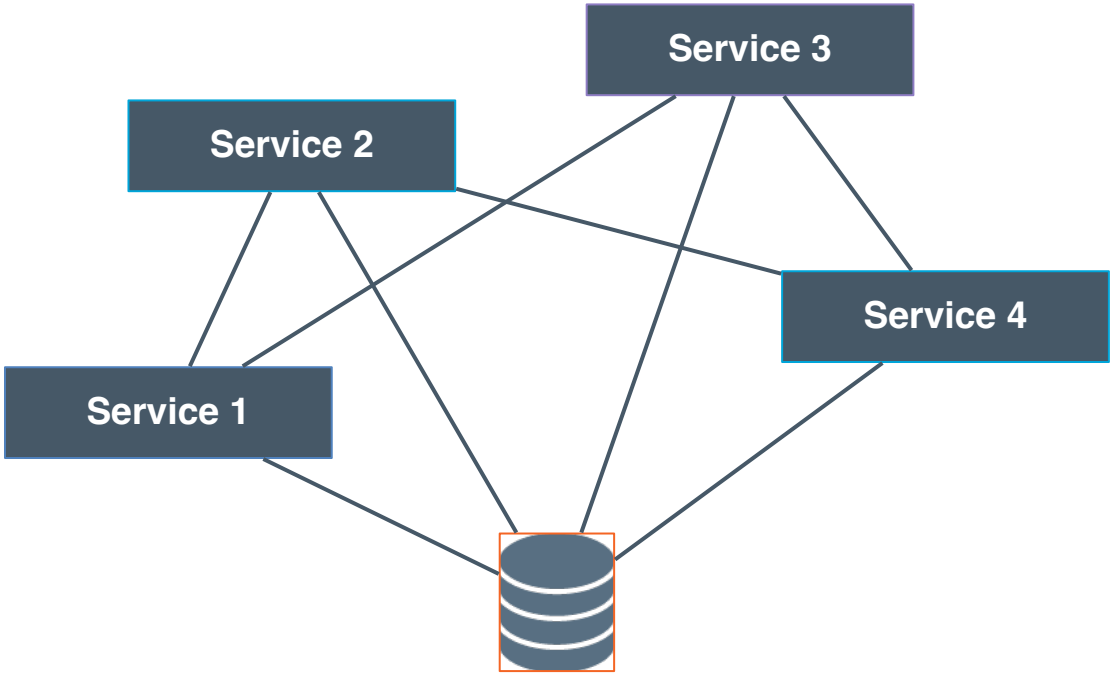


Source : https://www.jitterbit.com/blog/integrating-the-modern-hybrid-cloud-architecture/

Monolithic Architecture



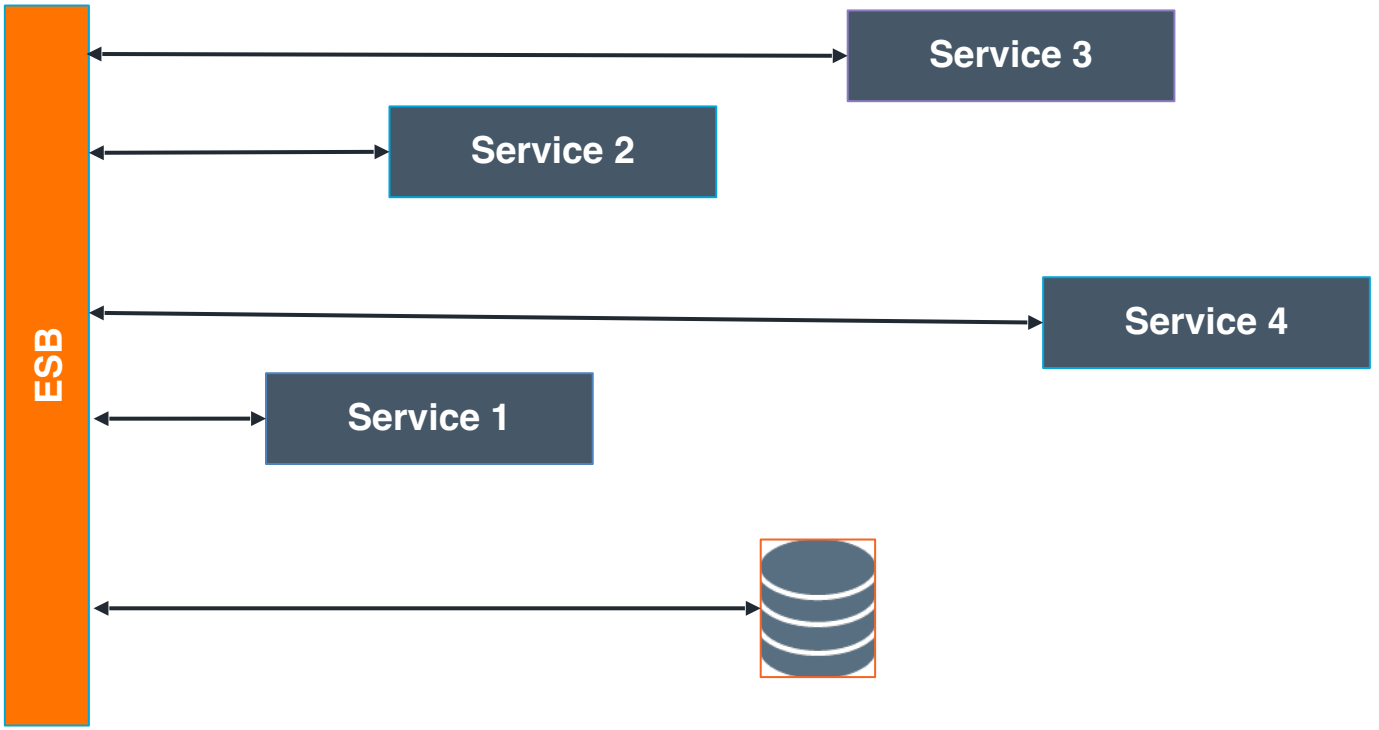
Consumers



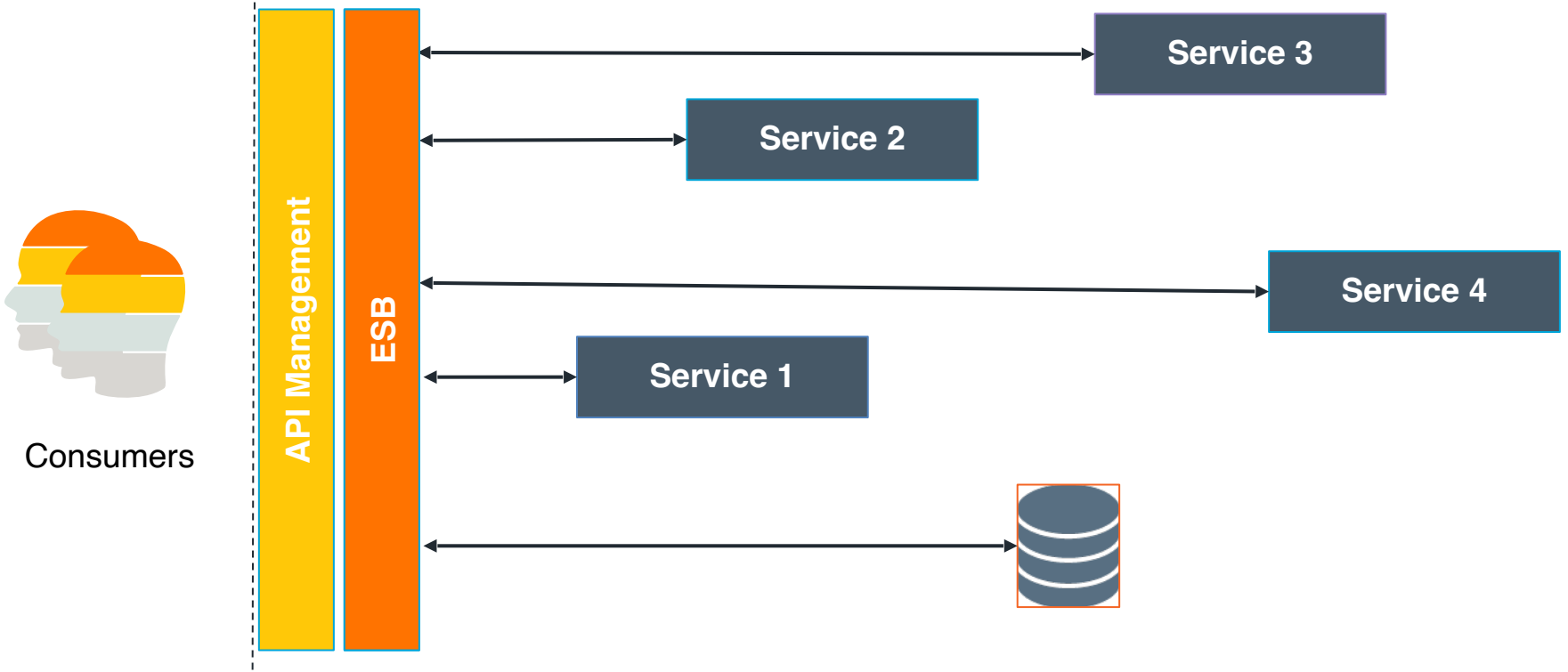
SOA and ESB



Consumers



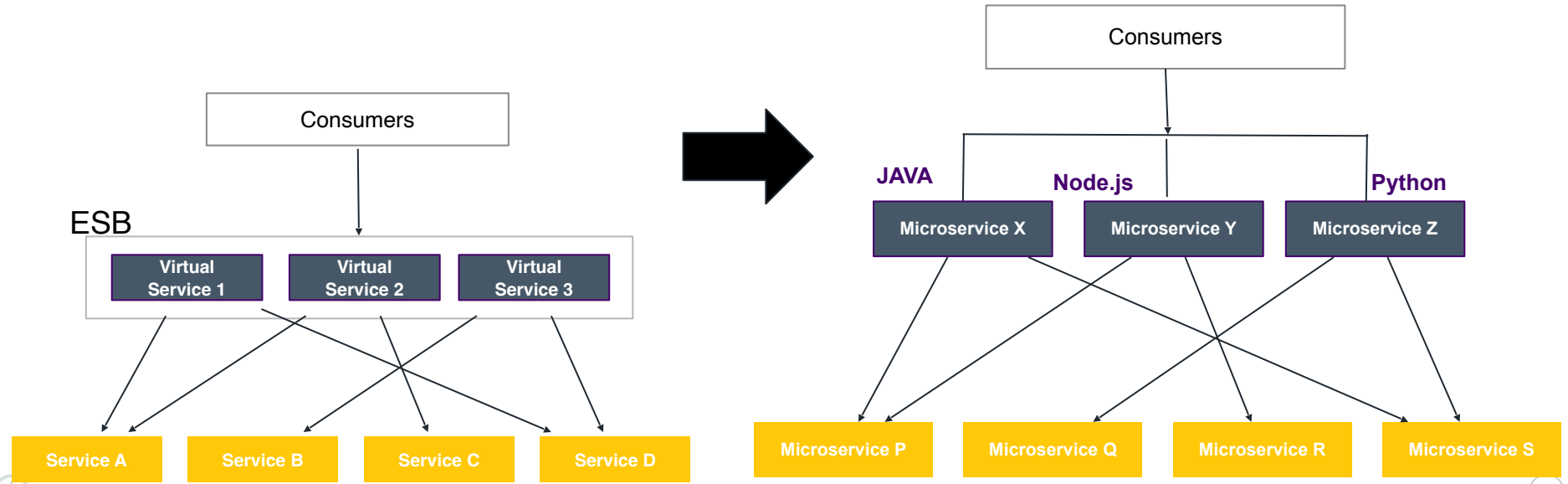
SOA with API Management



Smart Endpoints and Dumb Pipes

Eliminating the Central ESB
ESB contains business logic + network communication logic

With MSA, business logic + network communication logic are dispersed across independent services

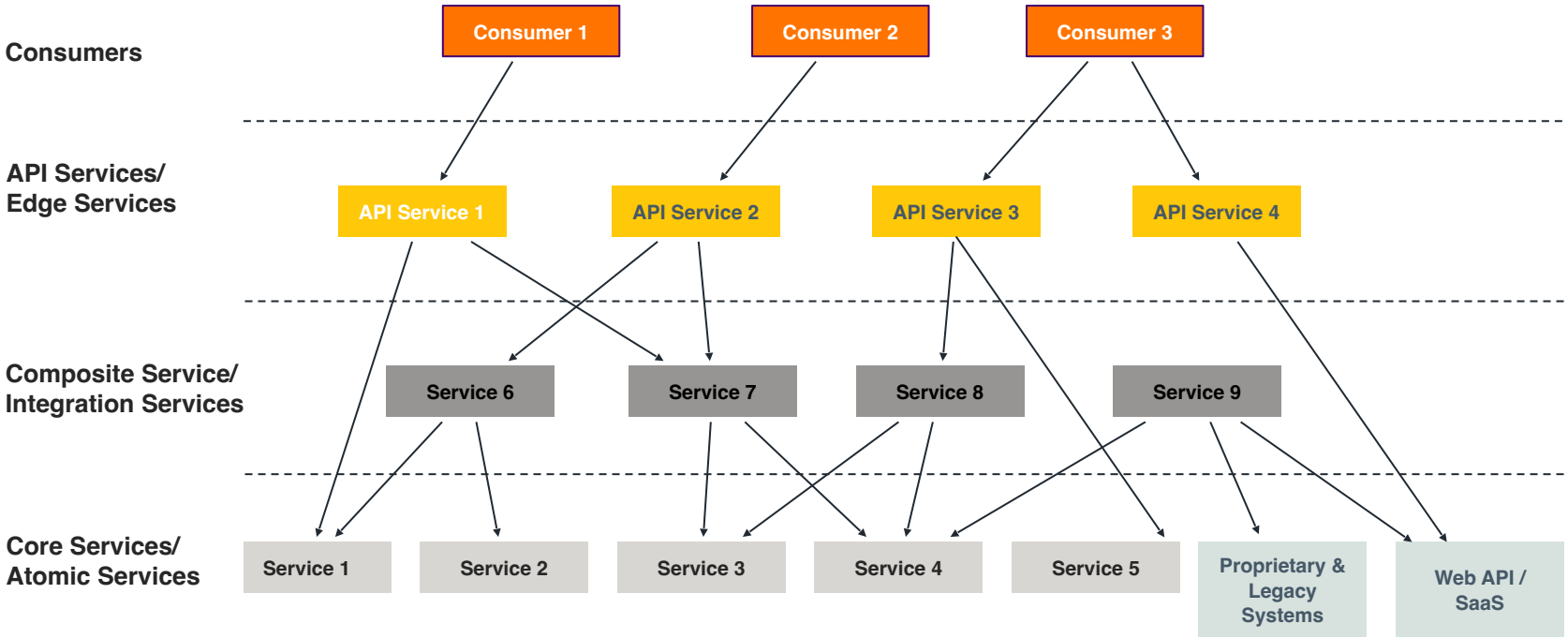


Organizing Microservices

Microservices Types with Different Granularities

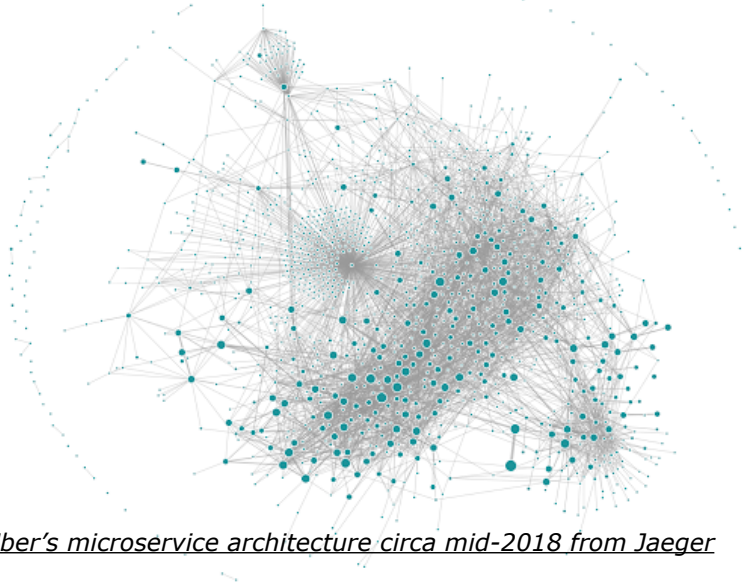
- Not all microservices are similar
- Some services are business logic heavy, while some contain a lot of interservice calls
- Some services are too fine-grained to be exposed as a business functionality
- Some services are exposed as APIs

Microservices Types with Different Granularities



Integrating Microservices is hard!

- Inter-service communication is essential.
- The capabilities that ESB provided have to be implemented at each service level.
- Integration complexity is not reduced but dispersed across services.



Uber's microservice architecture circa mid-2018 from Jaeger

Abstractions for Inter-Service communication

- Supporting high-level abstractions for inter-service communication.
- Support for multiple protocols and standards.
  - HTTP, gRPC, GraphQL, Web Sockets, Web Hooks, SOAP
  - Kafka, AMQP, NATS
  - FTP, SFTP
  - TCP, Rsocket
- Integrating with SaaS, proprietary and legacy systems
  - e.g. Salesforce, SAP, Amazon S3, SQS

Technologies for Building Microservices

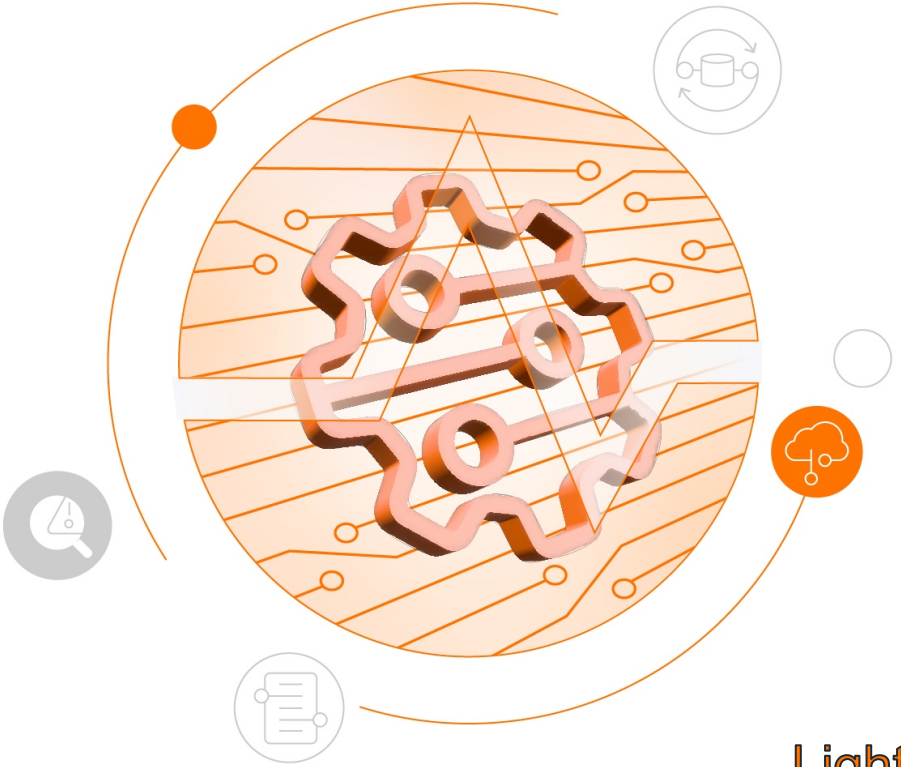
Technologies

- Frameworks Based on General Purpose Lang
- Languages designed for Integration
- Low Code Integration

WSO2 Micro Integrator

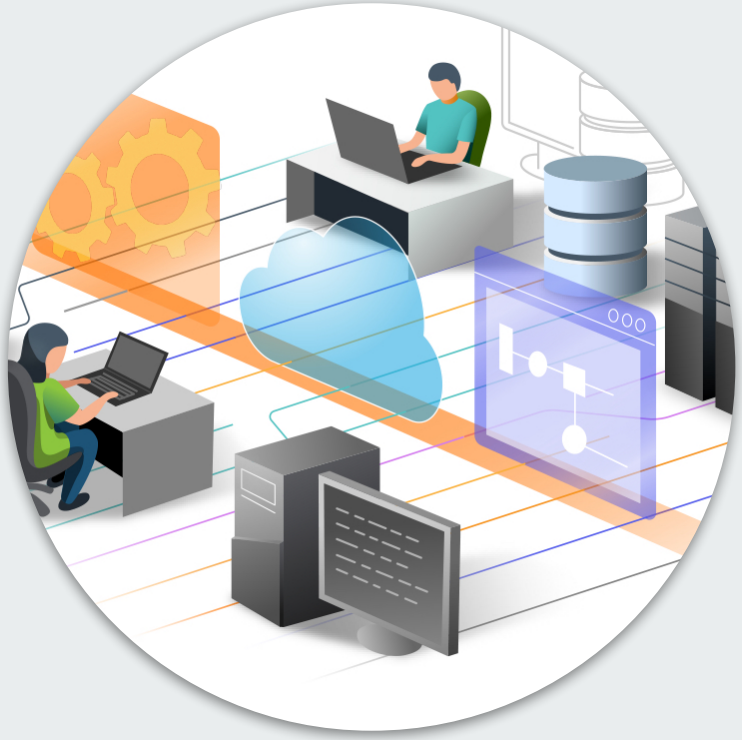
Micro Integrator

Robust



Powerful

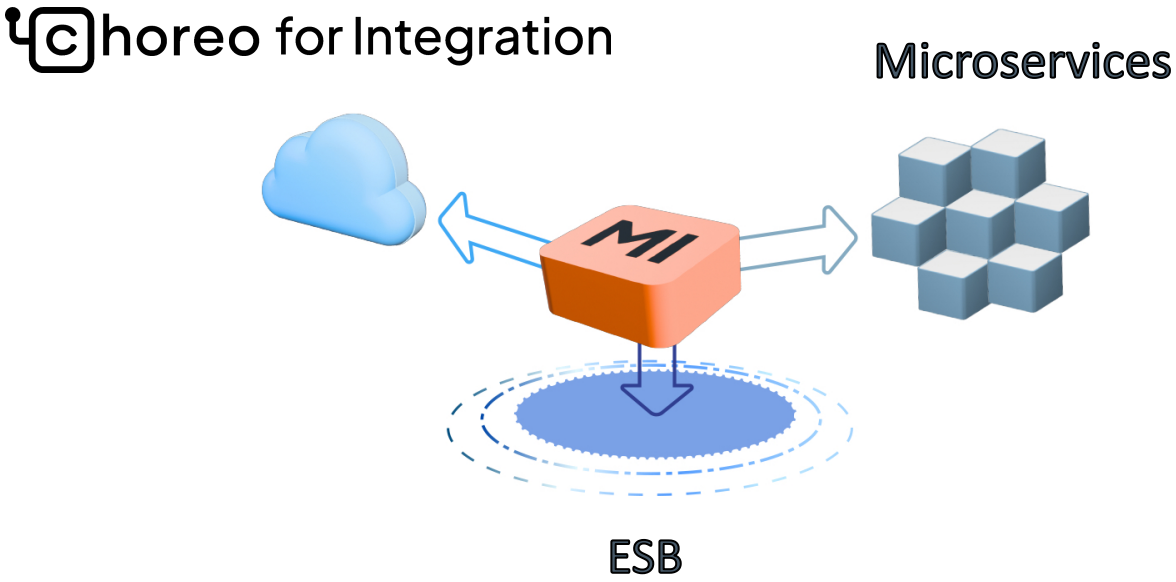
Lightweight

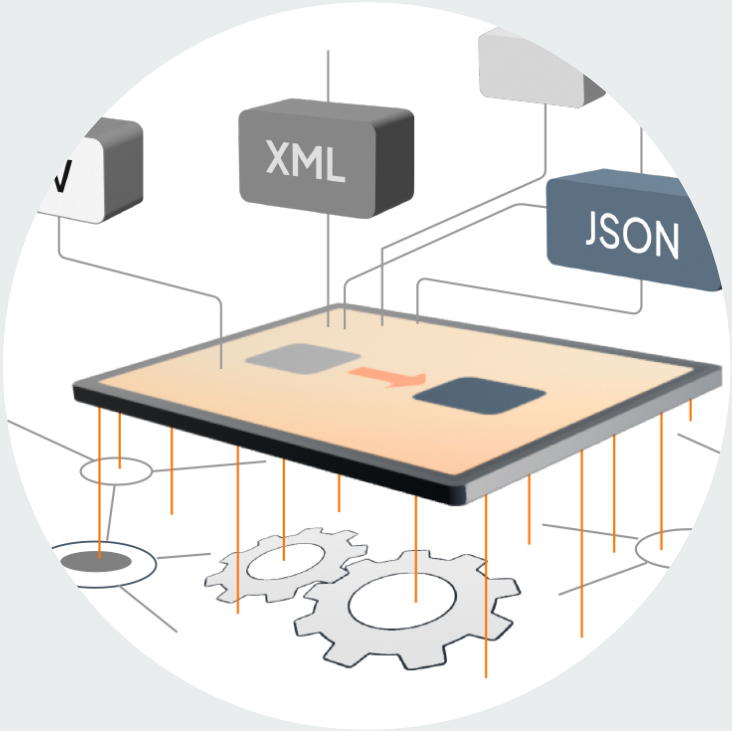


Solve Any

Integration Challenge

Flexible Deployment





Powerful
Transformations

Simplified Debugging and Testing

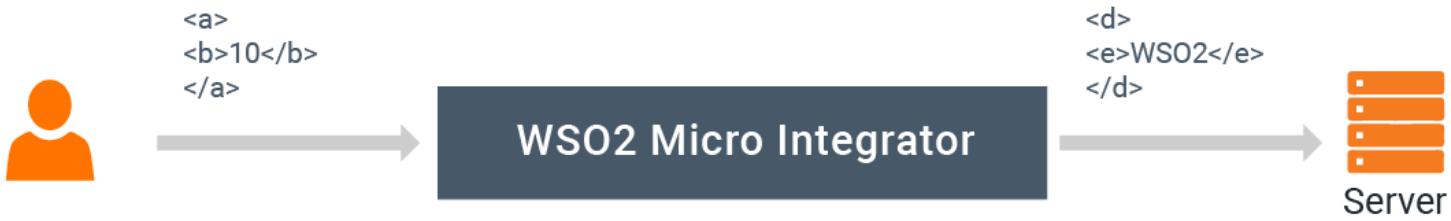


Key Features

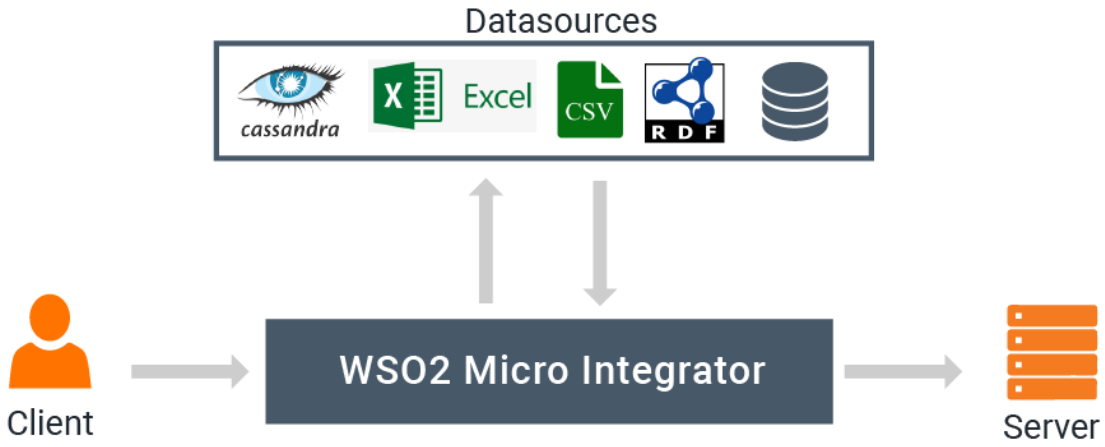
Message Routing



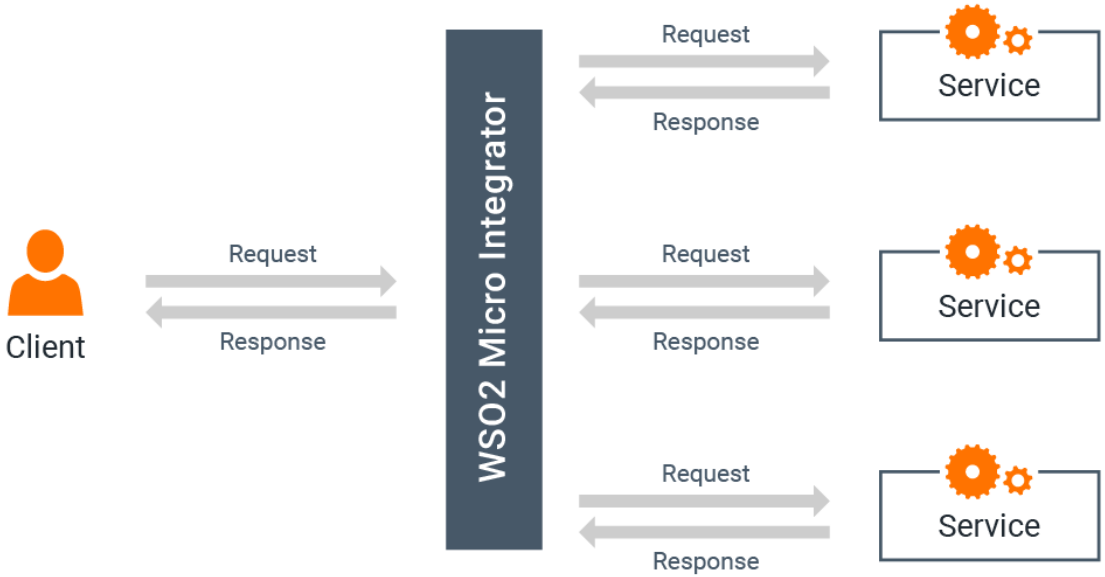
Message Transformation



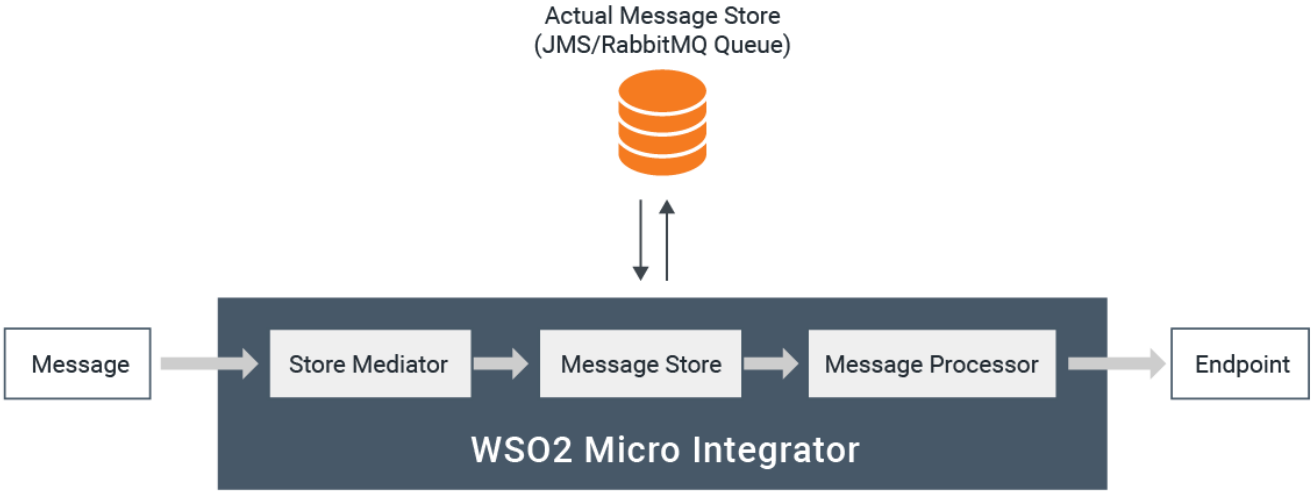
Data Integration



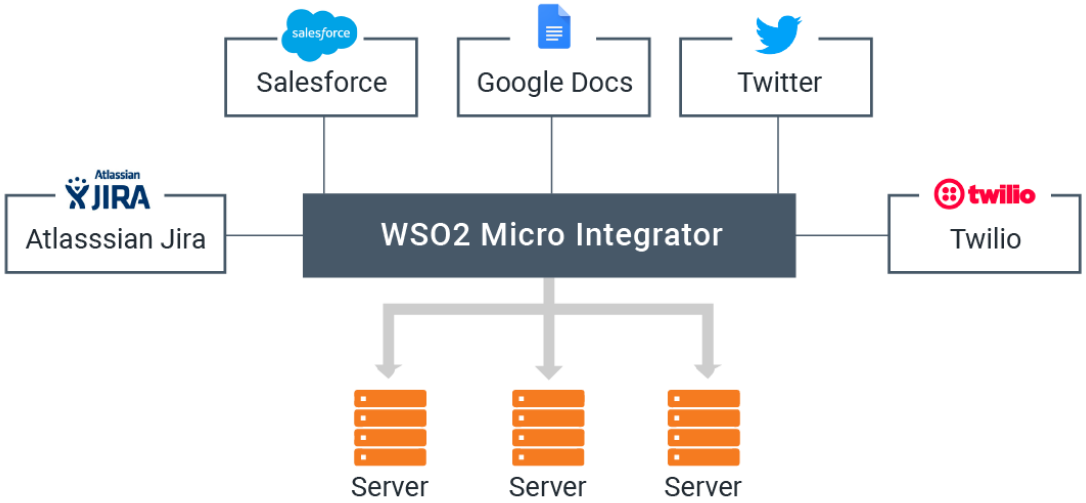
Service Orchestration



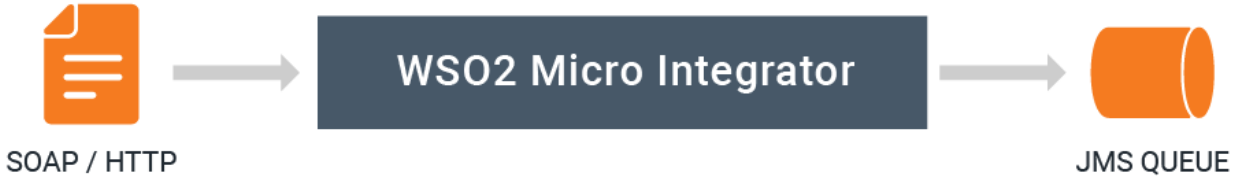
Asynchronous Messaging



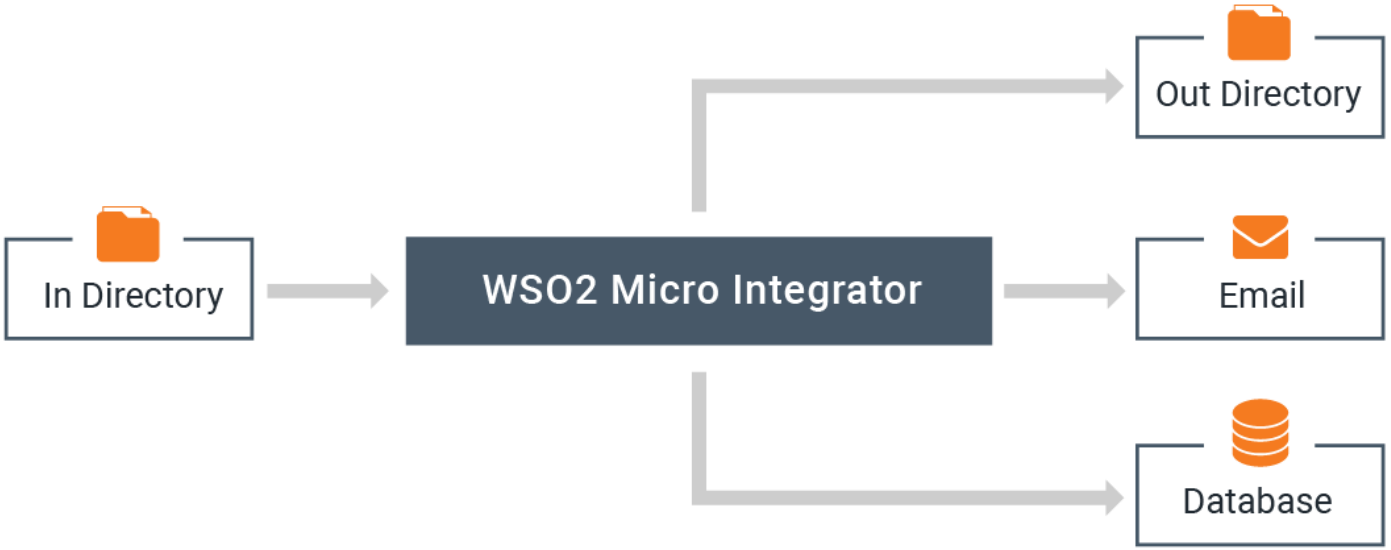
Connecting Web APIs/Cloud Services



Protocol Switching

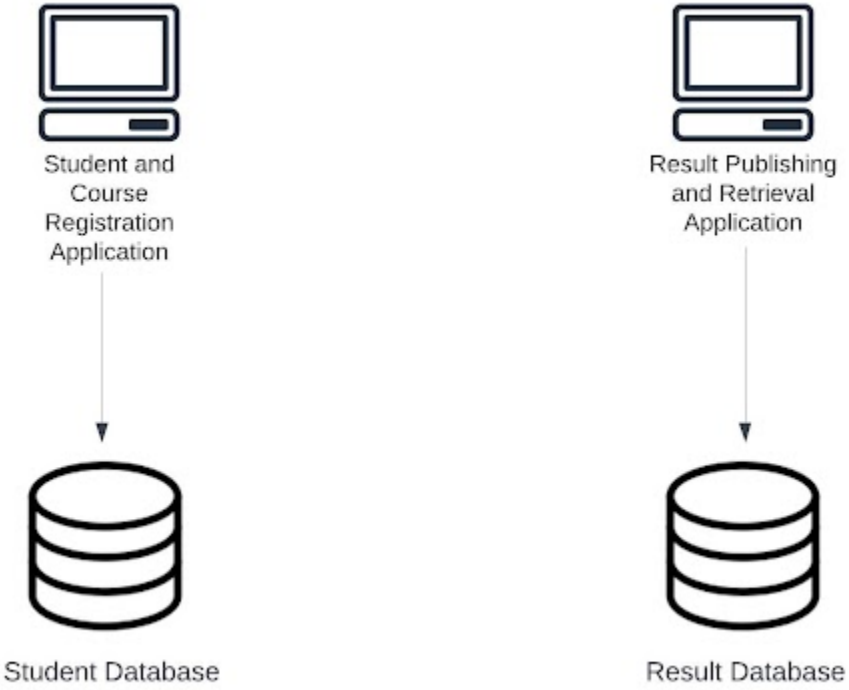


File Processing

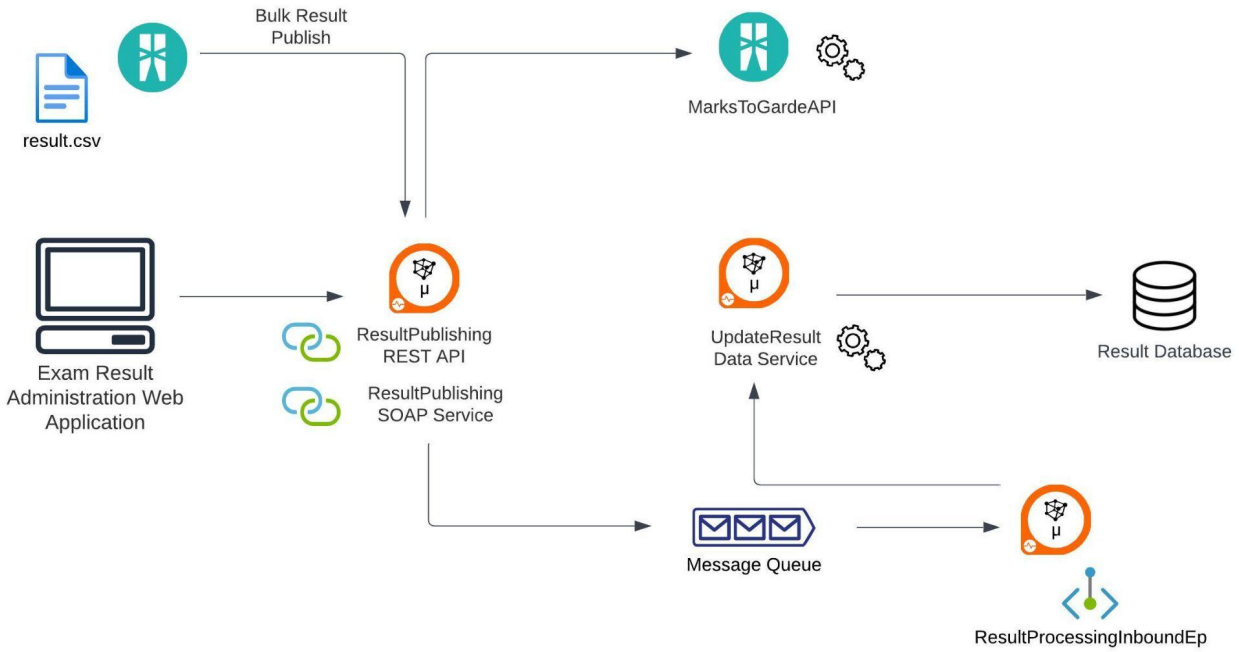


Scenario

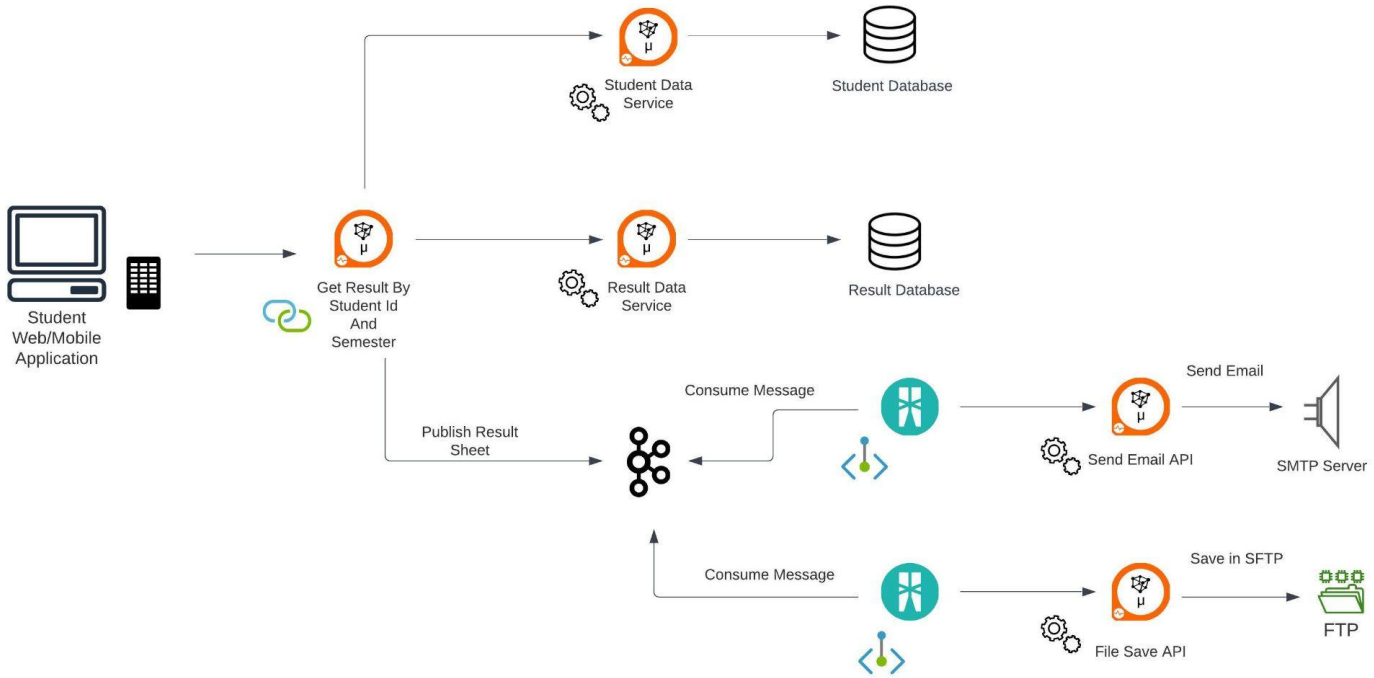
Scenario


Student and Course Registration Application
Result Publishing and Retrieval Application
Student Database
Result Database

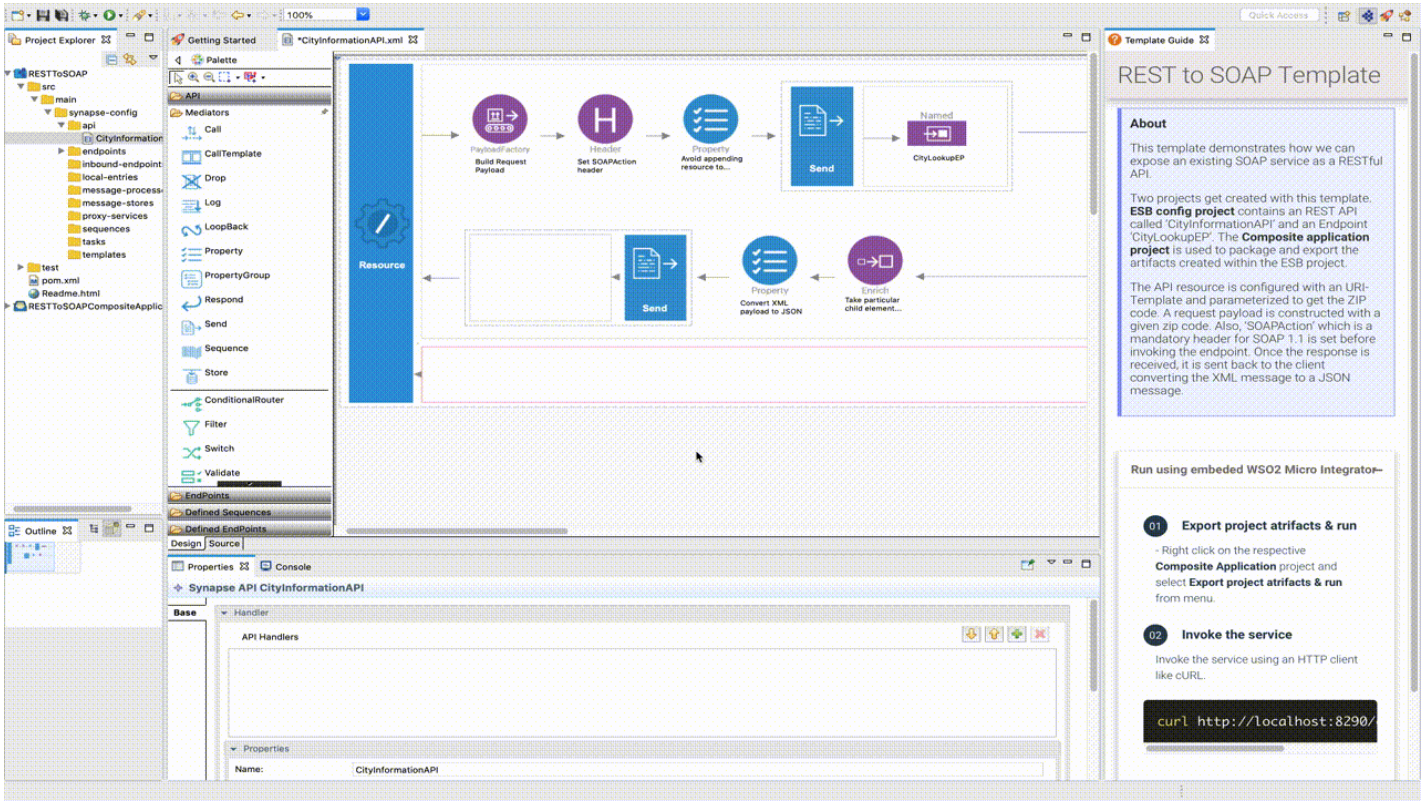
Scenario


Bulk Result
Publish
result.csv
MarksToGardeAPI
Exam Result
Administration Web
Application
ResultPublishing
REST API
ResultPublishing
SOAP Service
UpdateResult
Data Service
Result Database
Message Queue
ResultProcessingInboundEp

Scenario

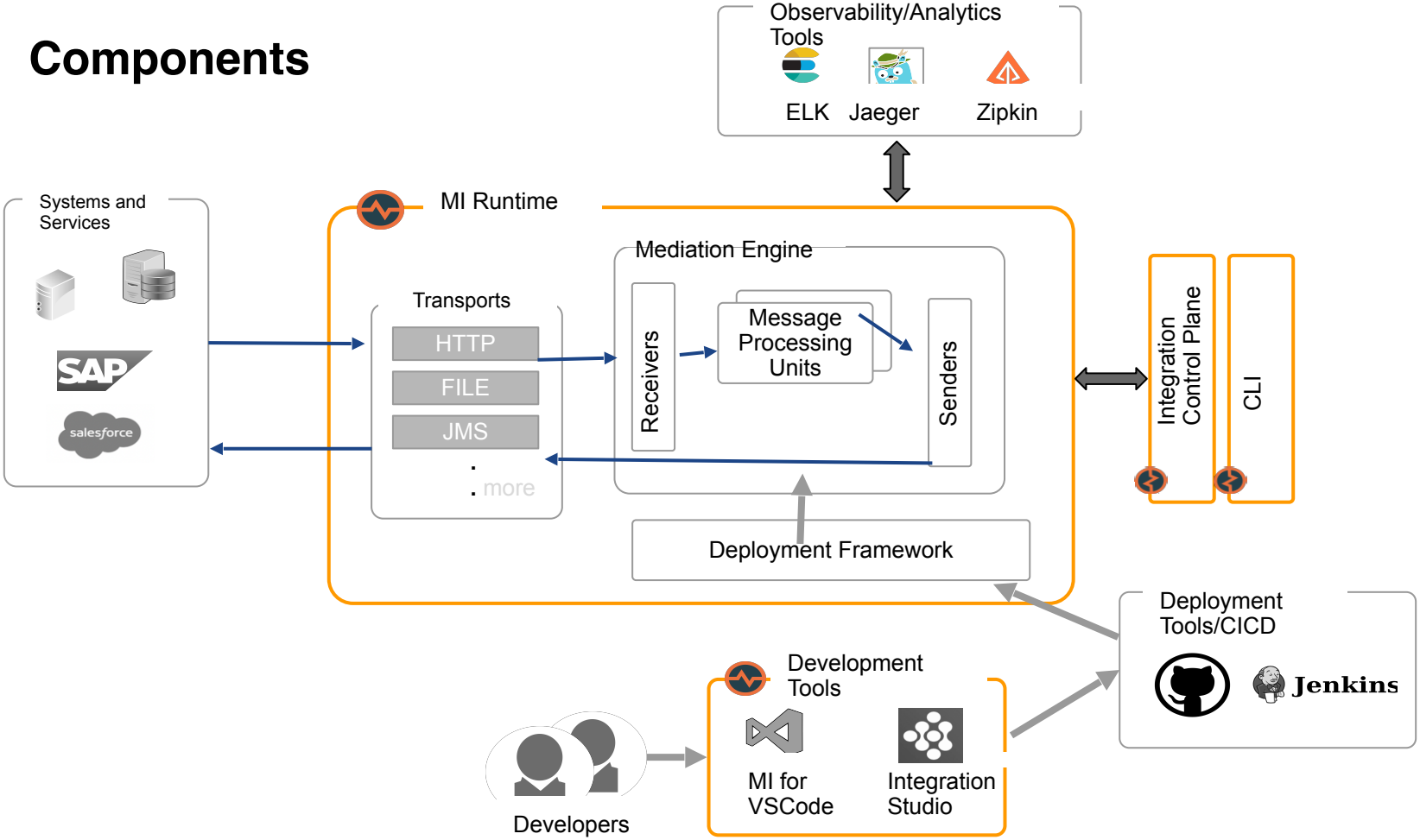

Student Data Service
Student Database
Student Web/Mobile Application
Get Result By Student Id And Semester
Result Data Service
Result Database
Consume Message
Send Email
Publish Result Sheet
Send Email API
SMTP Server
Consume Message
Save in SFTP
FTP
File Save API

Integration Studio



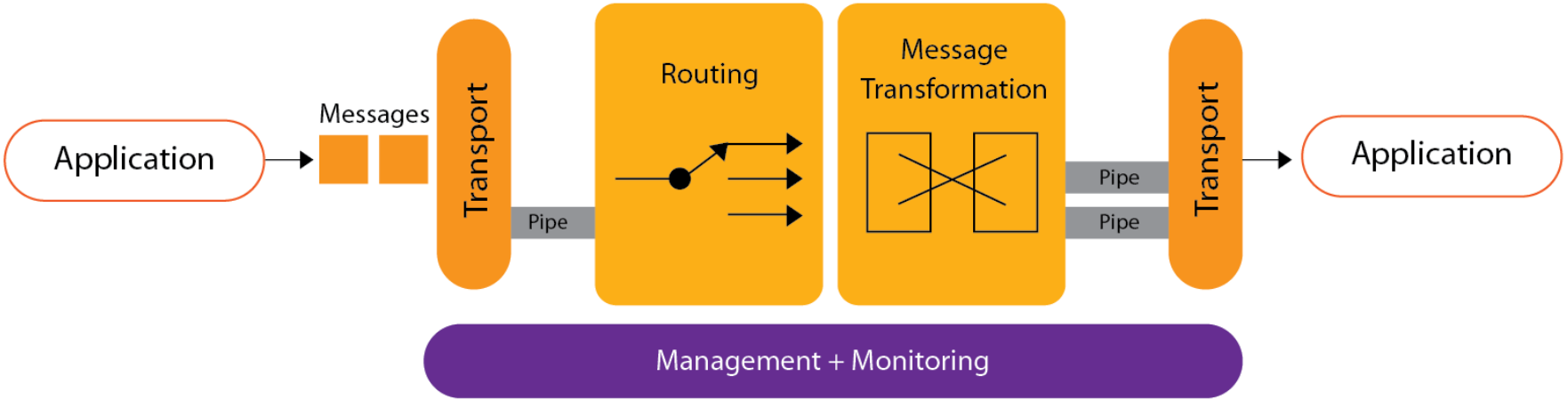
High-Level Architecture

Components


Observability/Analytics Tools
ELK
Jaeger
Zipkin
Systems and Services
SAP
salesforce
MI Runtime
Transports
HTTP
FILE
JMS
. . more
Mediation Engine
Receivers
Message Processing Units
Senders
Deployment Framework
Integration Control Plane
CLI
Deployment Tools/CICD
Jenkins
Development Tools
MI for VSCode
Integration Studio
Developers

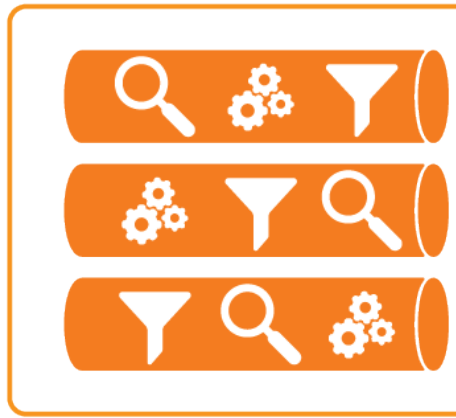
Concepts

Messaging Architecture



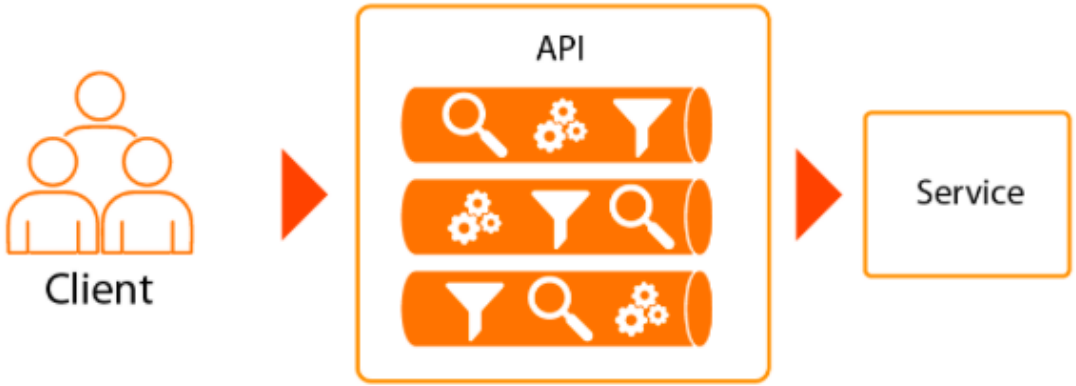
Triggering Messages

- Proxy Services
- Main Sequence
- APIs (http Services)
- Tasks
- Inbound Endpoints



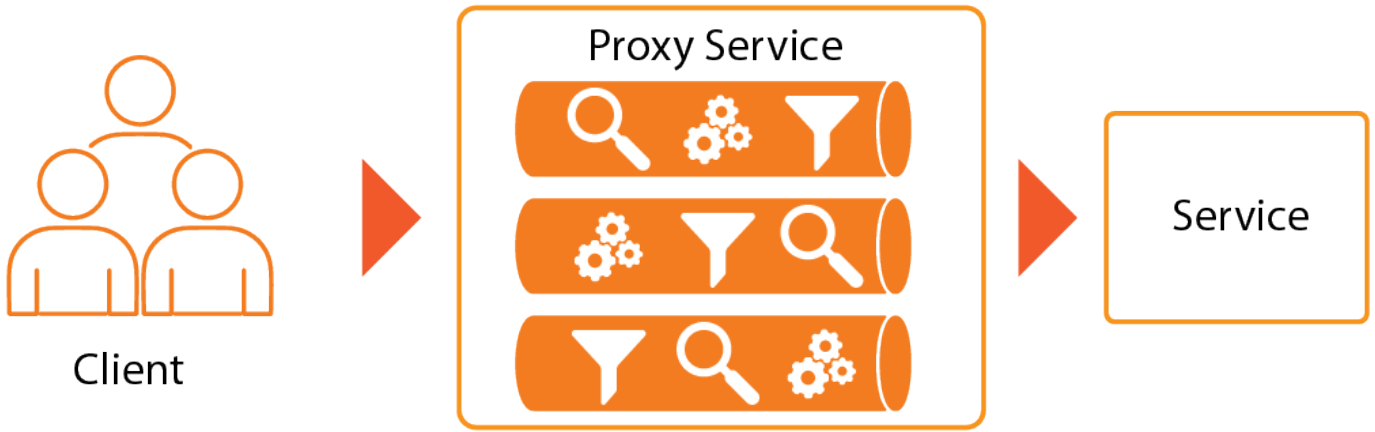
Sequence

APIs



Proxy Services

Acts as a virtual service. Receives messages and mediates them before sending them to the endpoint.



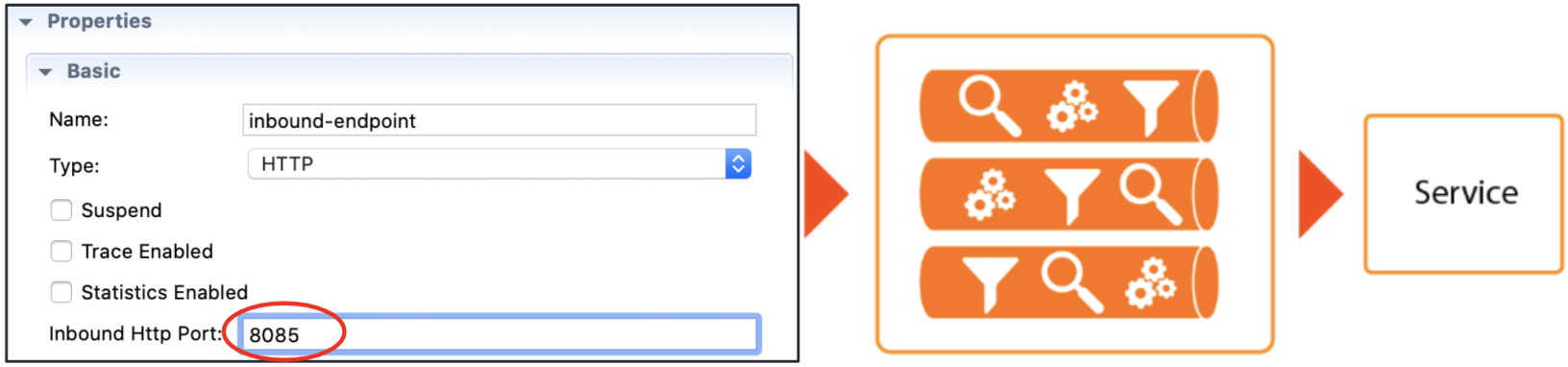
Tasks

Configure scheduled jobs that execute internal/external commands.



Inbound Endpoints

A message source that can be configured dynamically.

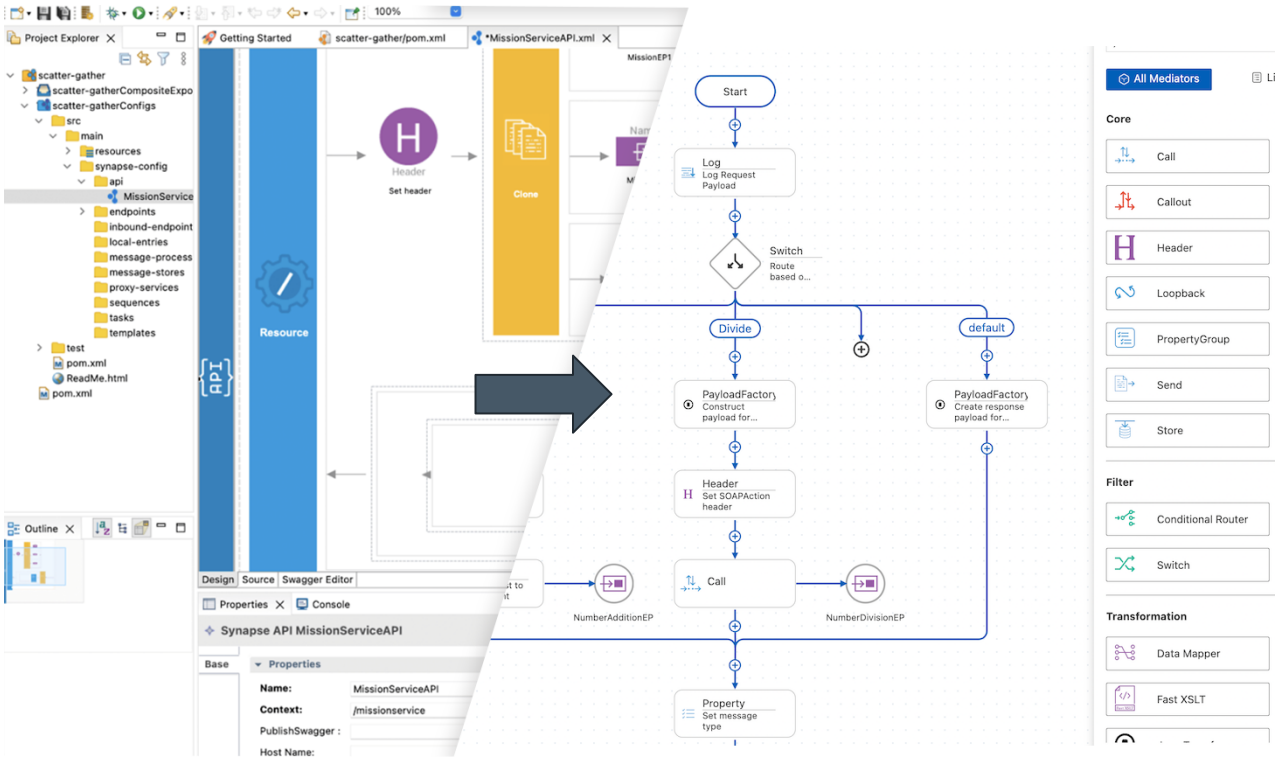


Mediators

Name	Description
Log Mediator	Logs full or part of the message at various severity levels (Trace, Debug, etc.).
Sequence Mediator	Invokes existing sequence. Sequence name can be static or dynamic.
Call Mediator	Sends a message out using static information or an endpoint definition.
Respond Mediator	Responds back to the client.
Switch Mediator	Evaluates message content against regular expressions and invokes the corresponding mediator (switch-case-default).
Property Mediator	Assigns a value to a property to be retrieved later
Drop Mediator	Stops processing the current message.
Fault Mediator	Transforms current message into a custom Fault message.

Scenario Contd..

Micro Integrator for VS Code



Question Time!



Thank You!

Provide Feedback

