Application Delivery Fundamentals

Introduction to Mulesoft


Our Mission
To help organizations
change and innovate faster
by making it easy to connect
the world's applications, data, and devices
MuleSoft
Connect anything.
Change everything.

Goals

- Understanding Mule concepts and terminologies 8
- Setting up the Mule IDE
- Installing Mule Studio
- Configuring Mule components
- Working with Components and Patterns
- Using Message Property, Processors, and Sources
- Endpoints
- Transformers
- Configuring Filters

Goals

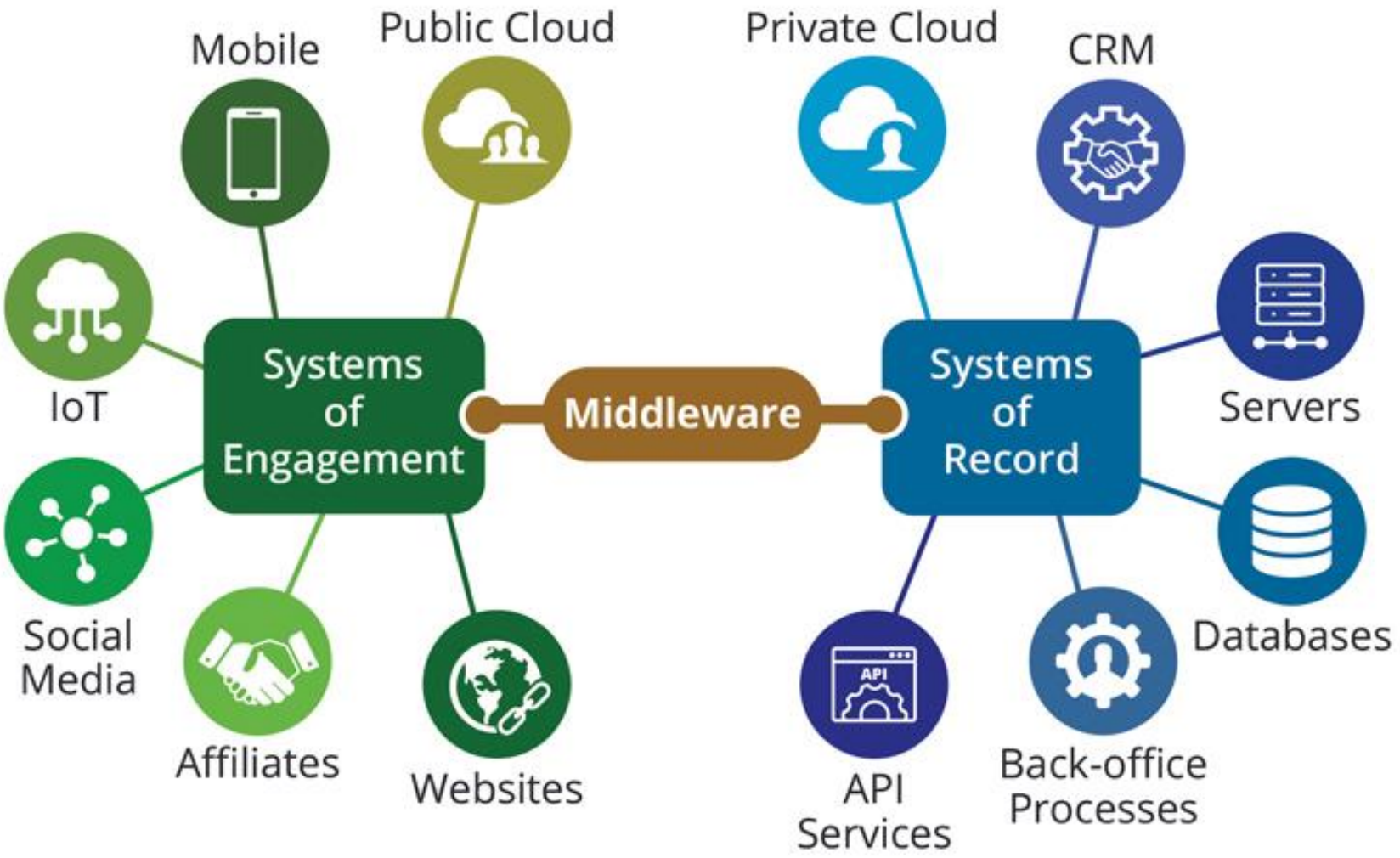
- Handling Exceptions and Testing
- Introducing Web Services
- Understanding Flows, Routers, and Services
- Configuring Cloud Connectors

Middleware

Middleware - The enterprise-wide bridge

- Middleware is the software that connects software components or enterprise applications.
- Middleware is the software layer that lies between the operating system and the applications on each side of a distributed computer network.
- Typically, it supports complex, distributed business software applications.
- Middleware is the infrastructure which facilitates creation of business applications.
- It provides core services like concurrency, transactions, threading, messaging, and the SCA (Service Component Architecture) framework for service-oriented architecture (SOA) applications.
- It also provides security and enables high availability functionality to enterprise.

Middleware - The enterprise-wide bridge



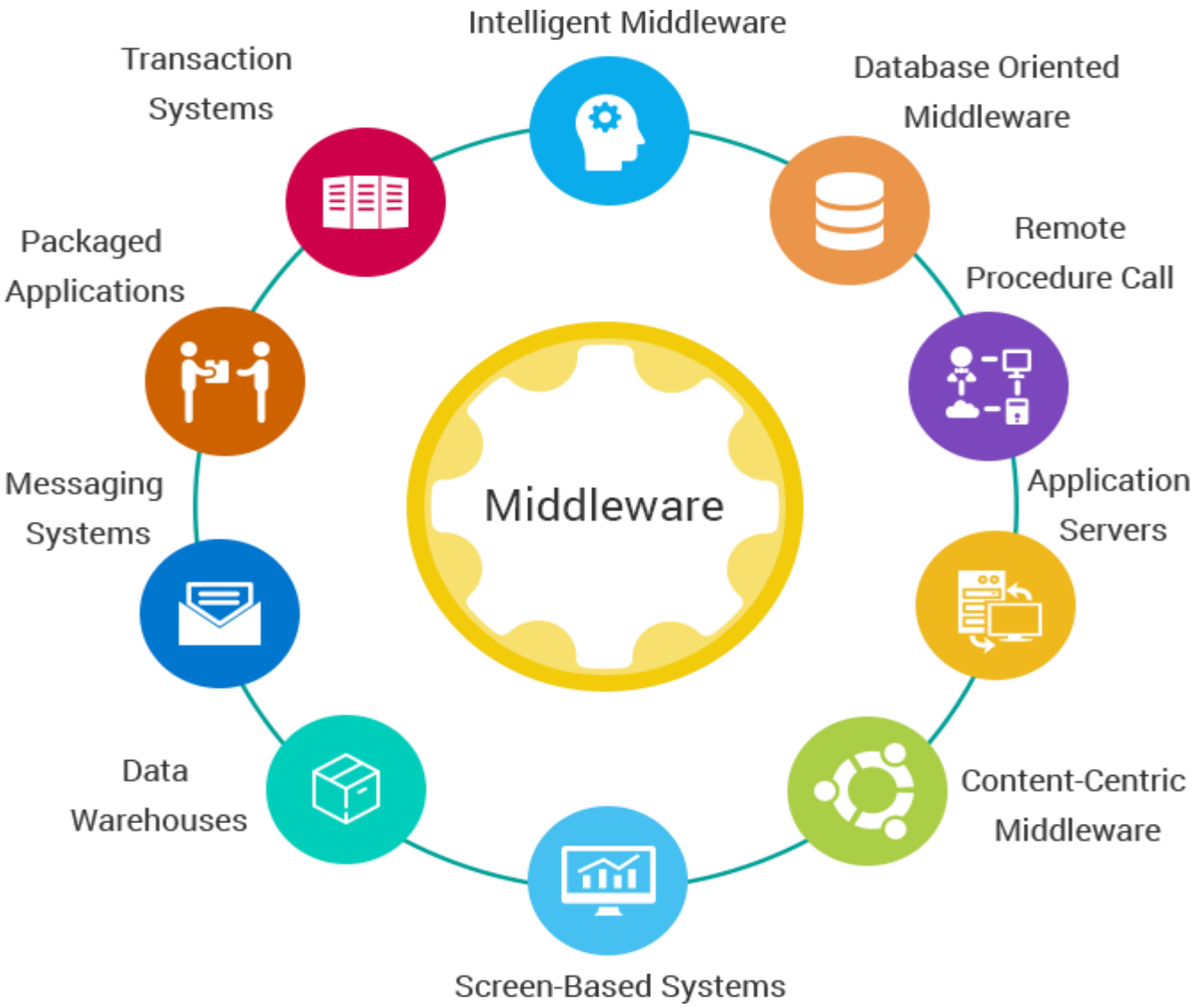
Middleware - The enterprise-wide bridge

- **Middleware is software that bridges gaps between other applications, tools, and databases in order to provide unified services to users.**
- It is commonly characterized as the glue that connects different software platforms and devices together.

Where is middleware used?

- Transaction management – ensures that corruption within the system or database doesn't happen should a problem occur.
- Application server.
- Security – provides authentication for a particular client program.
- Message queues – allows coupled systems to transfer messages back and forth.
- Directory – gives the client the ability to find other services in the business.
- Web server – a program responsible for accepting requests from web browsers.

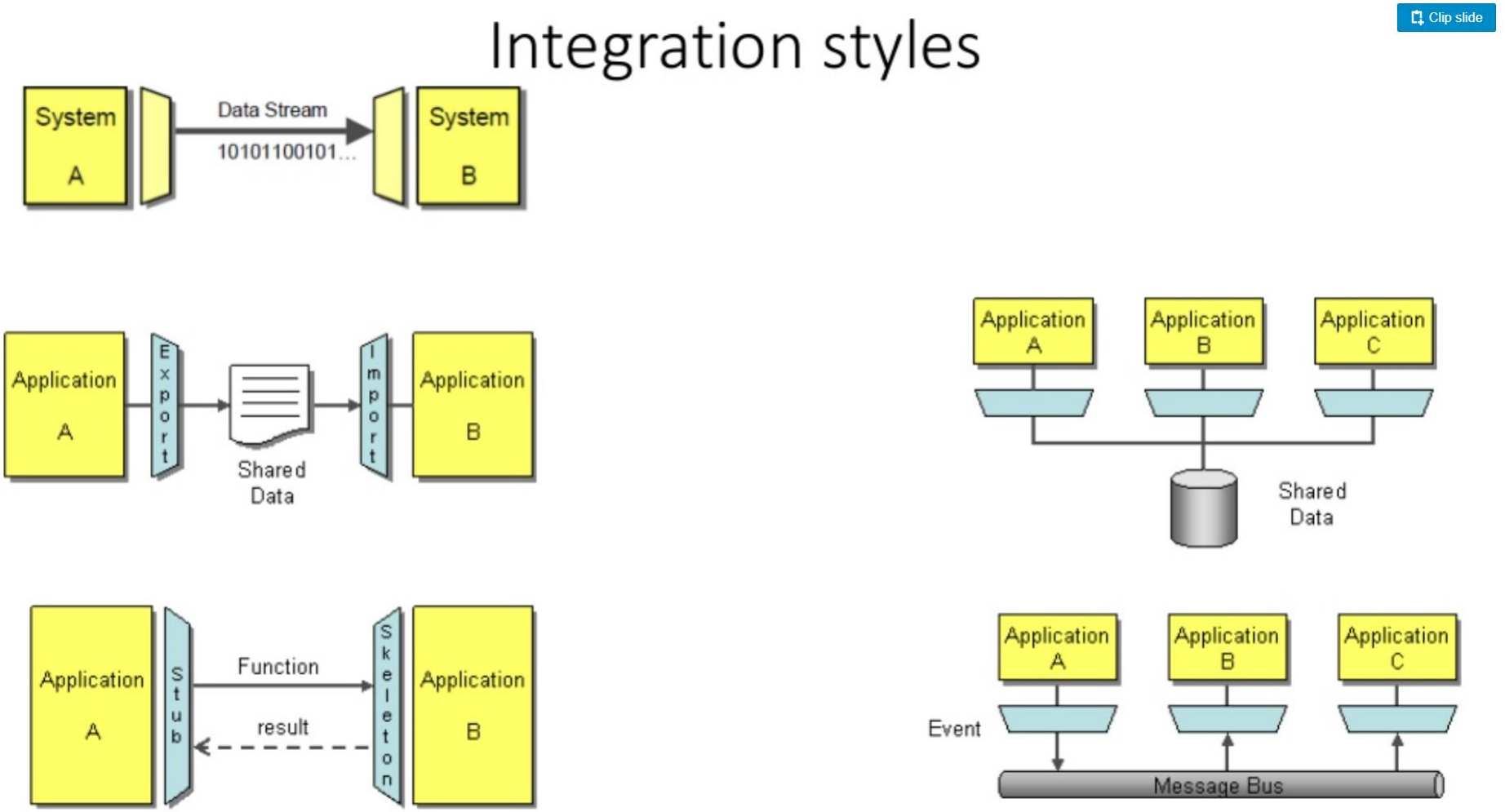
Where is middleware used?



From enterprise to platform middleware

- Middleware includes Web servers, application servers, content management systems, and similar tools that support application development and delivery.
- It is especially integral to information technology based on Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web services, SOA, Web 2.0 infrastructure, and Lightweight Directory Access Protocol (LDAP)m etc.

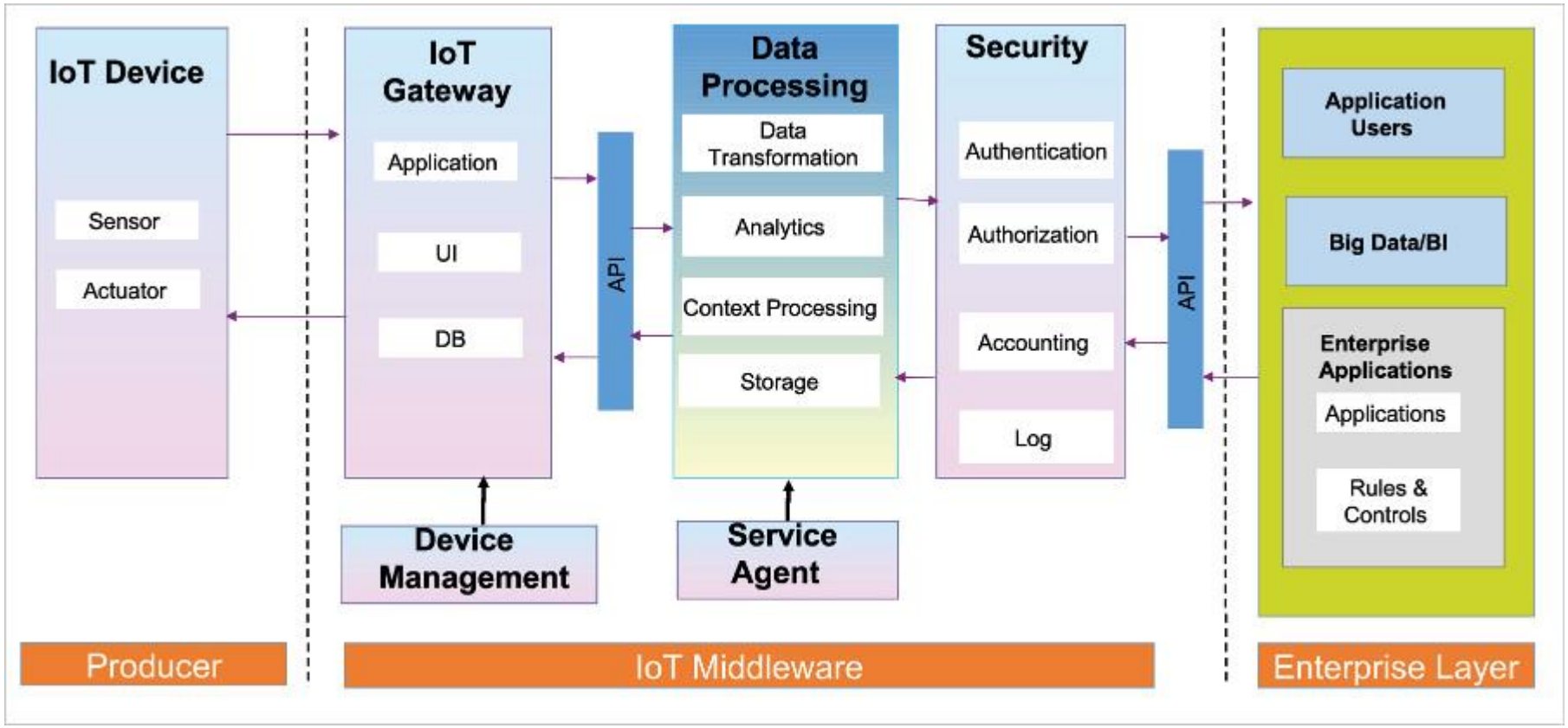
Integration Styles



From enterprise to platform middleware

- Application-centric (application and data management)
- Platform-centric (application enablement, device management and connectivity management)
- Industry-specific (manufacturing, healthcare, energy and utilities, transportation and logistics, agriculture, etc)

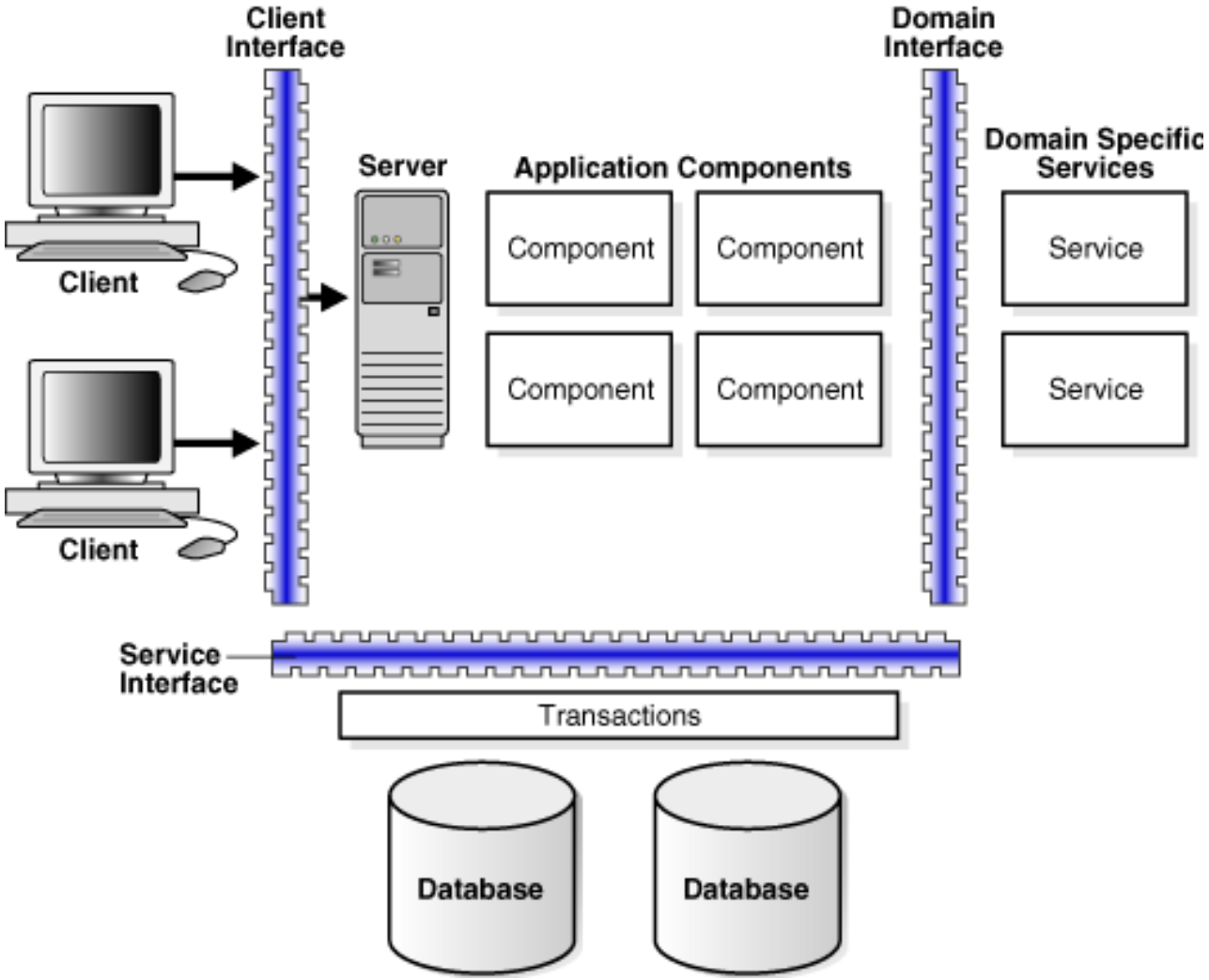
IoT middleware architecture

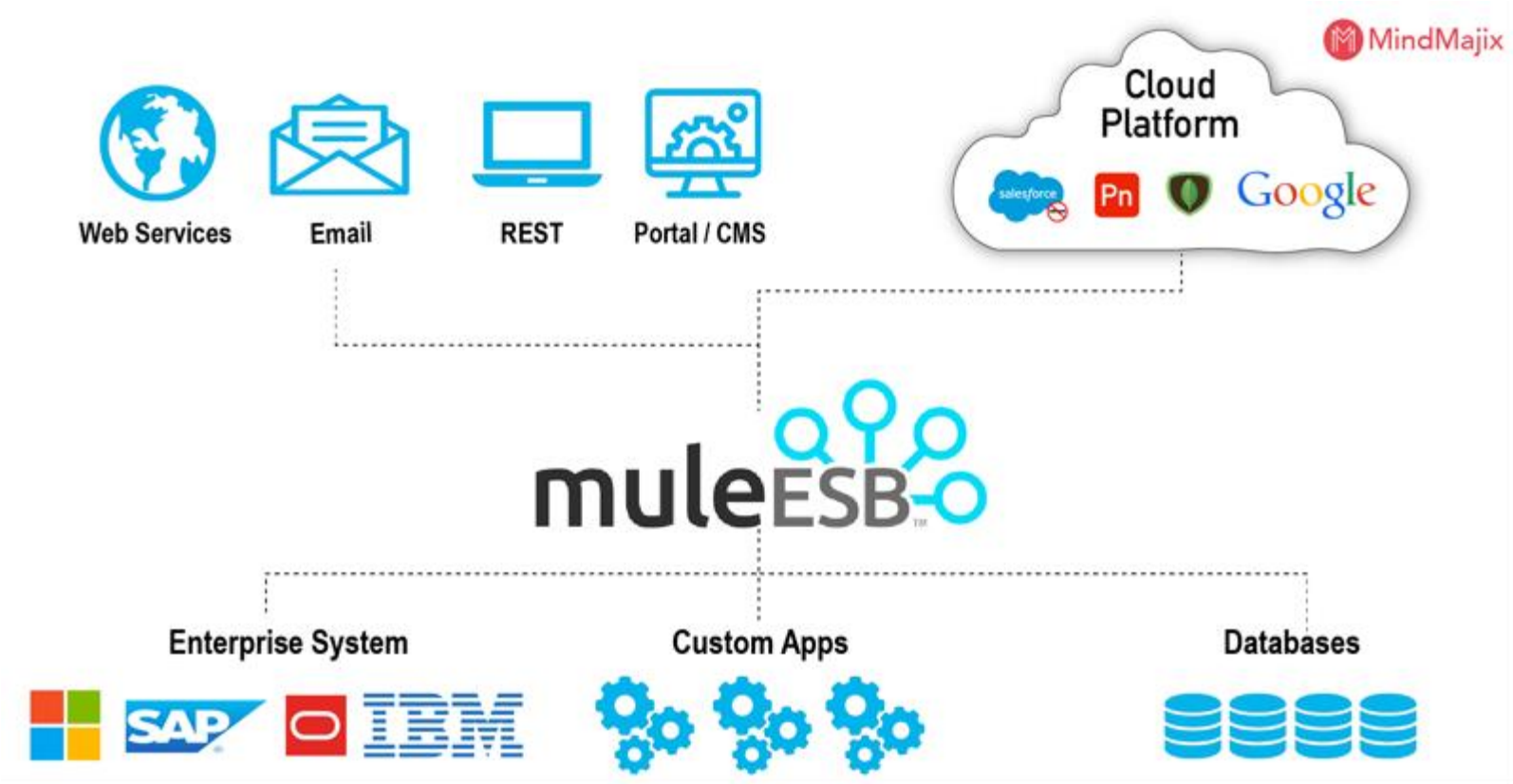


IoT Middleware

- Kaa is platform-centric middleware.
- It manages an unlimited number of connected devices with cross-device interoperability.
- Kaa is built on open components like Kafka, Cassandra, MongoDB, Redis, NATS, Spring, React, etc.
- **SiteWhere** is platform-centric middleware. It provides ingestion, storage, processing and integration of device data.

From enterprise to platform middleware




MindMajix
Cloud Platform
Web Services
Email
REST
Portal / CMS
Google
mule ESB
Enterprise System
Custom Apps
Databases
SAP
IBM

Few List of Middleware

- Message Oriented Middleware (MOM)
- Object Middleware
- Remote Procedure Call (RPC) Middleware
- Database Middleware
- Transactional Middleware
- Robotic Middleware
- Integration Middleware
- Application Framework
- Device Middleware
- Game Engines Disconnected
- Embedded Middleware
- Content-centric Middleware

Understanding middleware is quite difficult but examining each specific types makes them easier to comprehend.

Types of Middleware

- MESSAGE ORIENTED MIDDLEWARE
- This type of middleware is an infrastructure that supports the receiving and sending of messages over distributed applications.
- It enables applications to be disbursed over various platforms.
- It makes the process of creating software applications spanning many operating systems and network protocols much less complicated.
- It holds many advantages over middleware alternatives (e.g. hard coding logic).
- It is one of the most widely used types of middleware.

Types of Middleware

- **OBJECT MIDDLEWARE**
  - Object middleware, also called an object request broker, gives applications the ability to send objects and request services via an object oriented system.
  - In short, it manages the communication between objects.
- **REMOTE PROCEDURE CALL (RPC) MIDDLEWARE**
  - It calls procedures on remote systems and is used to perform synchronous or asynchronous interactions between applications or systems.
  - It is usually utilized within a software application.

Types of Middleware

- **DATABASE MIDDLEWARE**
  - This type of middleware allows for direct access to databases, providing direct interaction with them.
  - There are many database gateways and connectivity options.
  - This is the most general and commonly known type of middleware. This includes SQL database software.
- **TRANSACTION MIDDLEWARE**
  - This type of middleware includes applications like transaction processing monitors.
  - It also encompasses web-application servers.

Types of Middleware

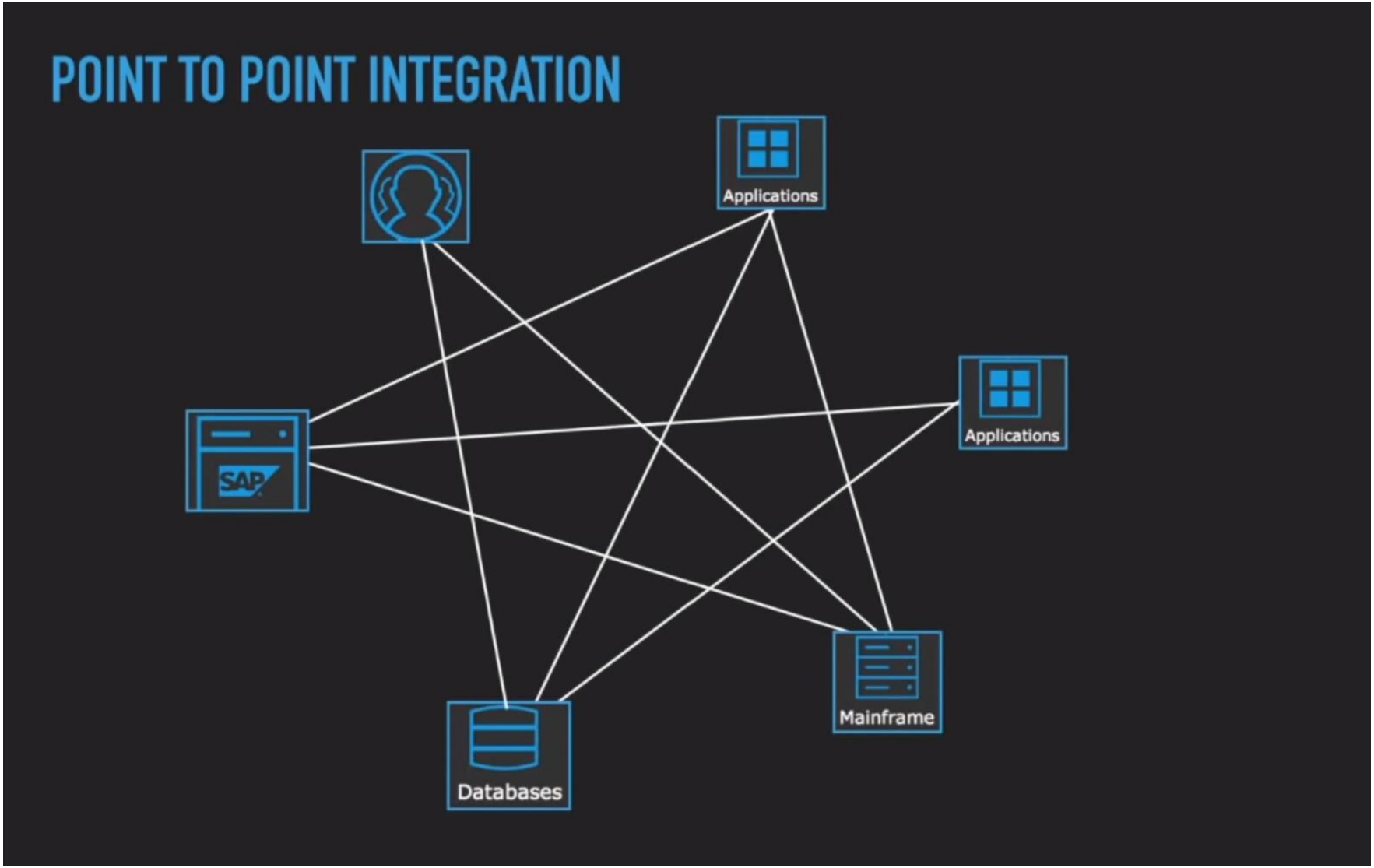
- **PORTALS**
  - This refers to enterprise portal servers.
  - It is considered middleware because portals facilitate front-end integration.
  - They are used to create interactions between a user's computer or device and back-end systems and services.
- **EMBEDDED MIDDLEWARE**
  - This type of middleware allows for communication and integration services with an interface of software or firmware.
  - It acts as a liaison between embedded applications and the real-time operating system.

Types of Middleware

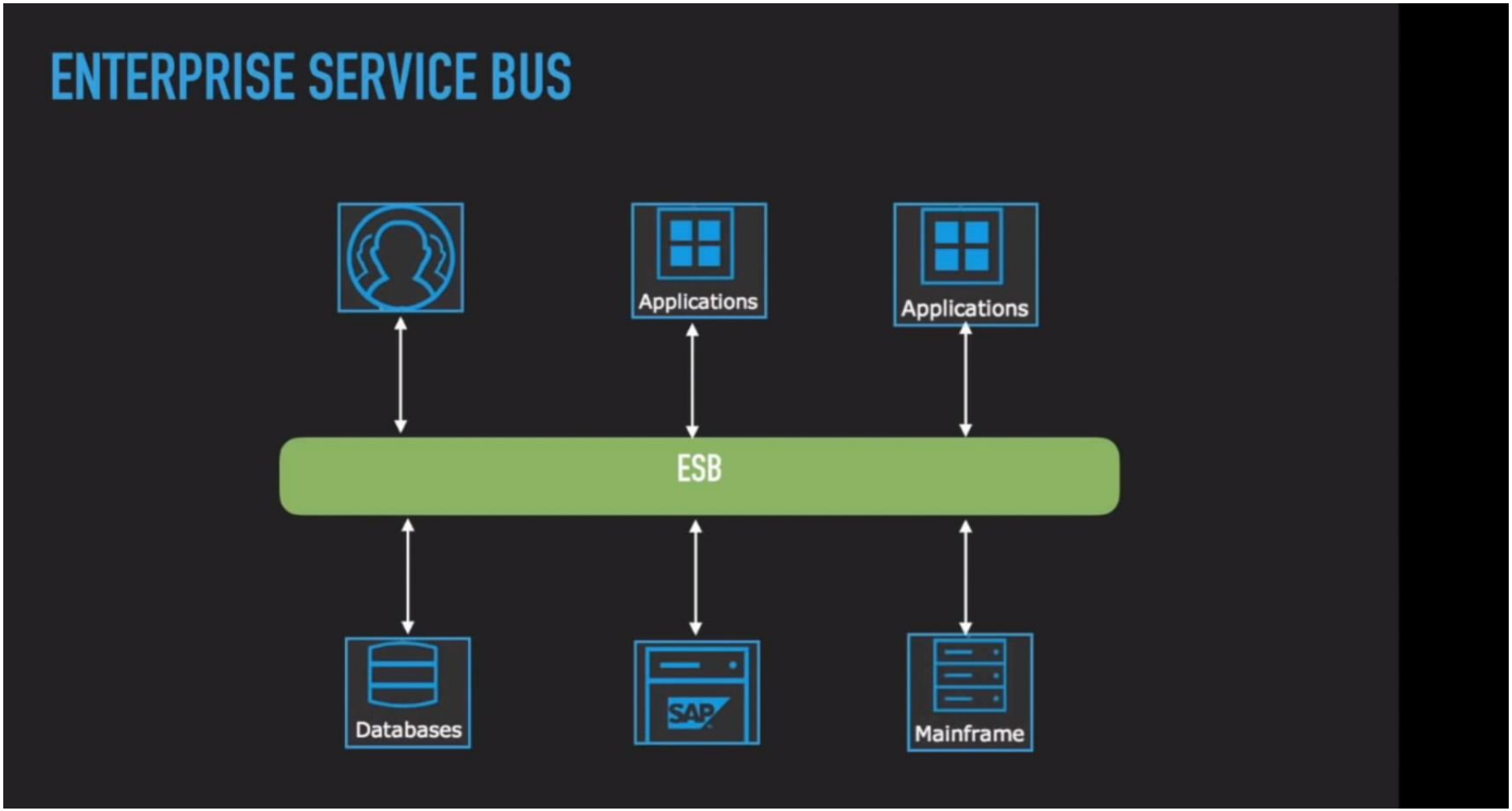
- **CONTENT-CENTRIC MIDDLEWARE**
  - This type of middleware allows you to abstract specific content without worrying how it is obtained.
  - This is done through a simple provide / consume abstraction.
  - It is similar to publish / subscribe middleware, which is another type of this software that is often used as a part of web-based applications.
-

ESB

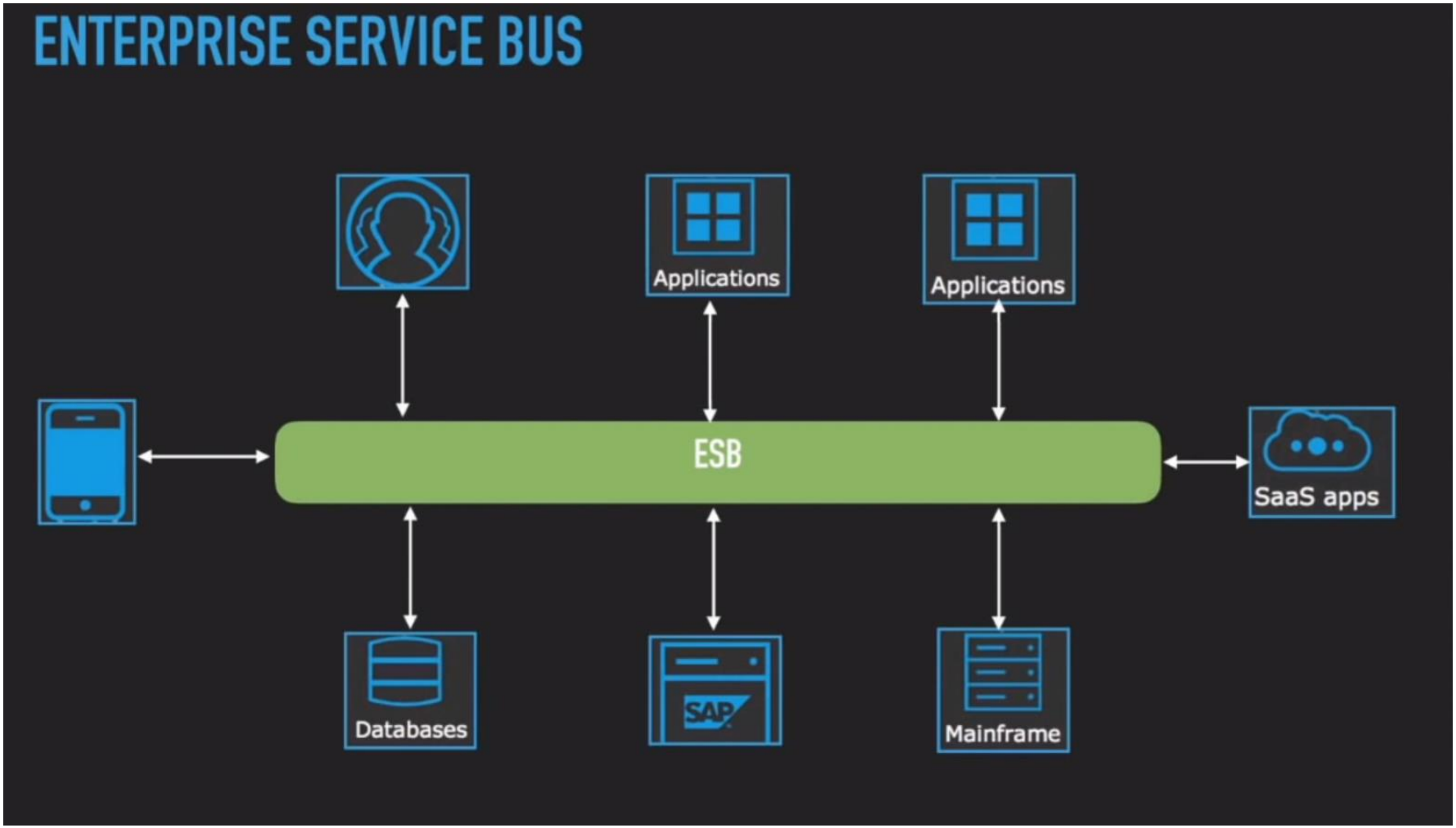
Legacy Integration Patterns



ESB


ENTERPRISE SERVICE BUS
Applications
Applications
ESB
Databases
SAP
Mainframe

ESB Adding Components without Pain



What is ESB

- ESB is a software architecture for middleware
- Supports SOA
- Hide complexity
- Simplify access
- Canonical pattern support
- Integration driven by business requirements not by available technology

ESB Features

- Routing
- Message transformation
- Message enhancement
- Protocol transformation
- Service mapping
- Message processing
- Process choreography
- Service orchestration
- Transaction Management
- Security

ESB Features

- Connecting Anything to Anything
- Minimal Custom Development, Developer Friendly and Easy to Deploy
- Routing, Mediation & Transformation
- Message, Service, API & Security Gateway
- High Performance, High Availability, Scalability & Stability
- Manage & Monitor

Routing, Mediation & Transformation

- Routing:
  - header based
  - content based
  - rule-based
  - priority-based
- Enterprise Integration Patterns (EIP):
  - message filter, aggregator, splitter, ..
  - recipient list
  - dead-letter channels
  - guaranteed delivery

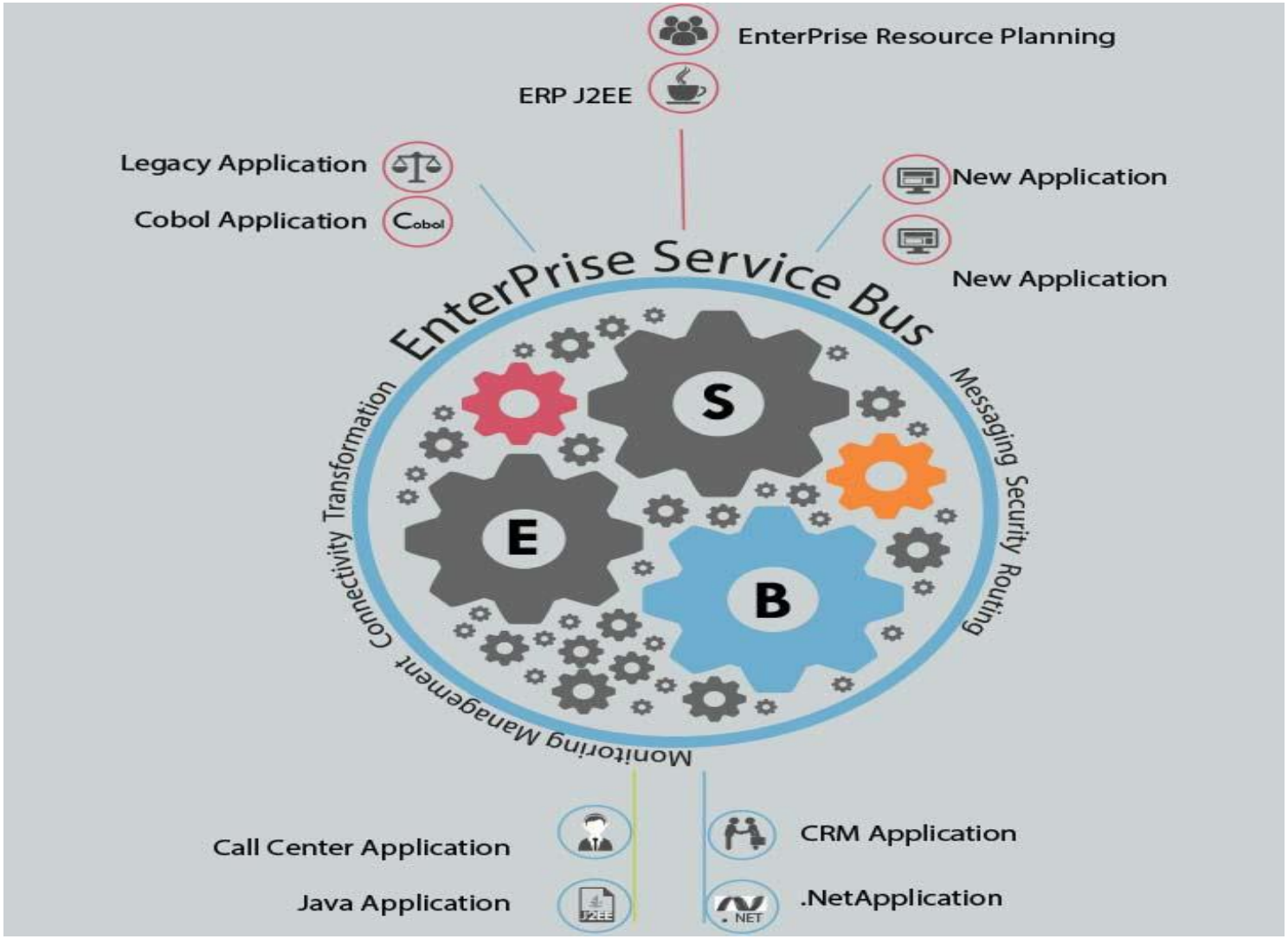
Manage & Monitor

- Comprehensive management & monitoring
- Built-in collection and monitoring of standard access and performance statistics
- JMX MBeans for key metrics monitoring and management
- Flexible logging support with integration to enterprise logging systems
- Centralized configuration management

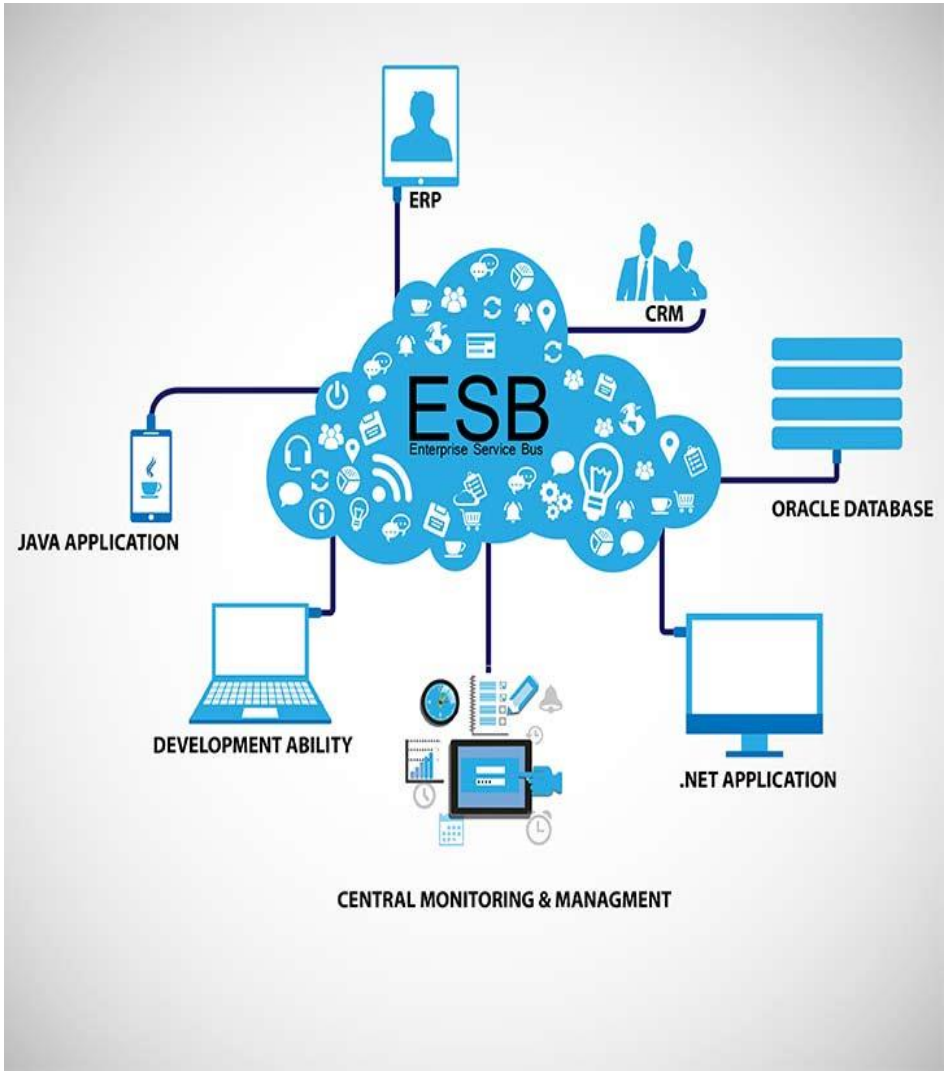
Auditing & Logging

- Centralized Auditing & Monitoring
- Capture any user action capture any user action
- Logging can be a very crucial aspect
- ESB is a distributed clustered set up where we have several applications
- We need to keep centralized application logs
- Asynchronous and None-Blocking Data publishing
- Receives and Stores Log Events in highly scalable and Big Data Repository
- Rich tool set for analytics
- Dashboards and reports

ESB


EnterPrise Resource Planning
ERP J2EE
Legacy Application
Cobol Application
New Application
New Application
EnterPrise Service Bus
S
E
B
Connectivity Transformation
Messaging Security Routing
Monitoring Management
Call Center Application
Java Application
CRM Application
.NetApplication

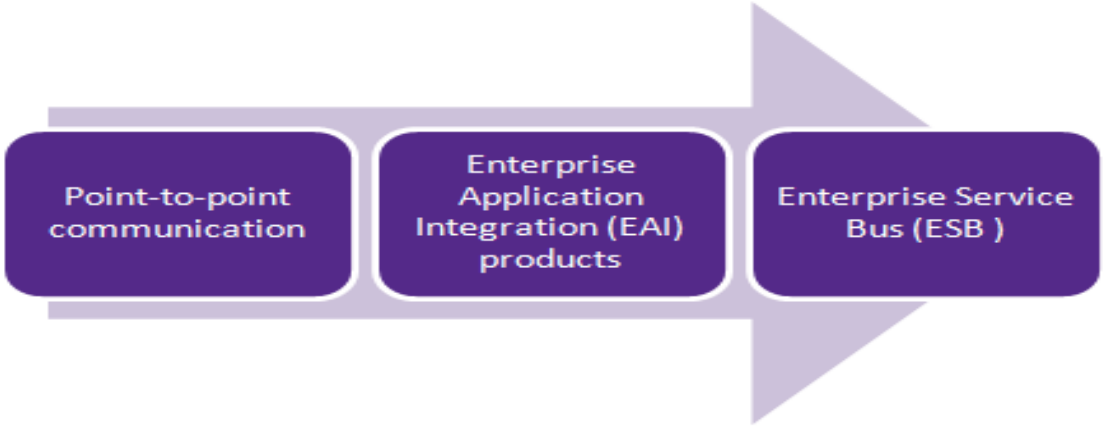
ESB



- software architecture construct, which lives between the (business) applications and enables communication among them
- replace all direct contact with the applications on the bus
- reduces the number of point-to-point connections between communicating applications

ESB

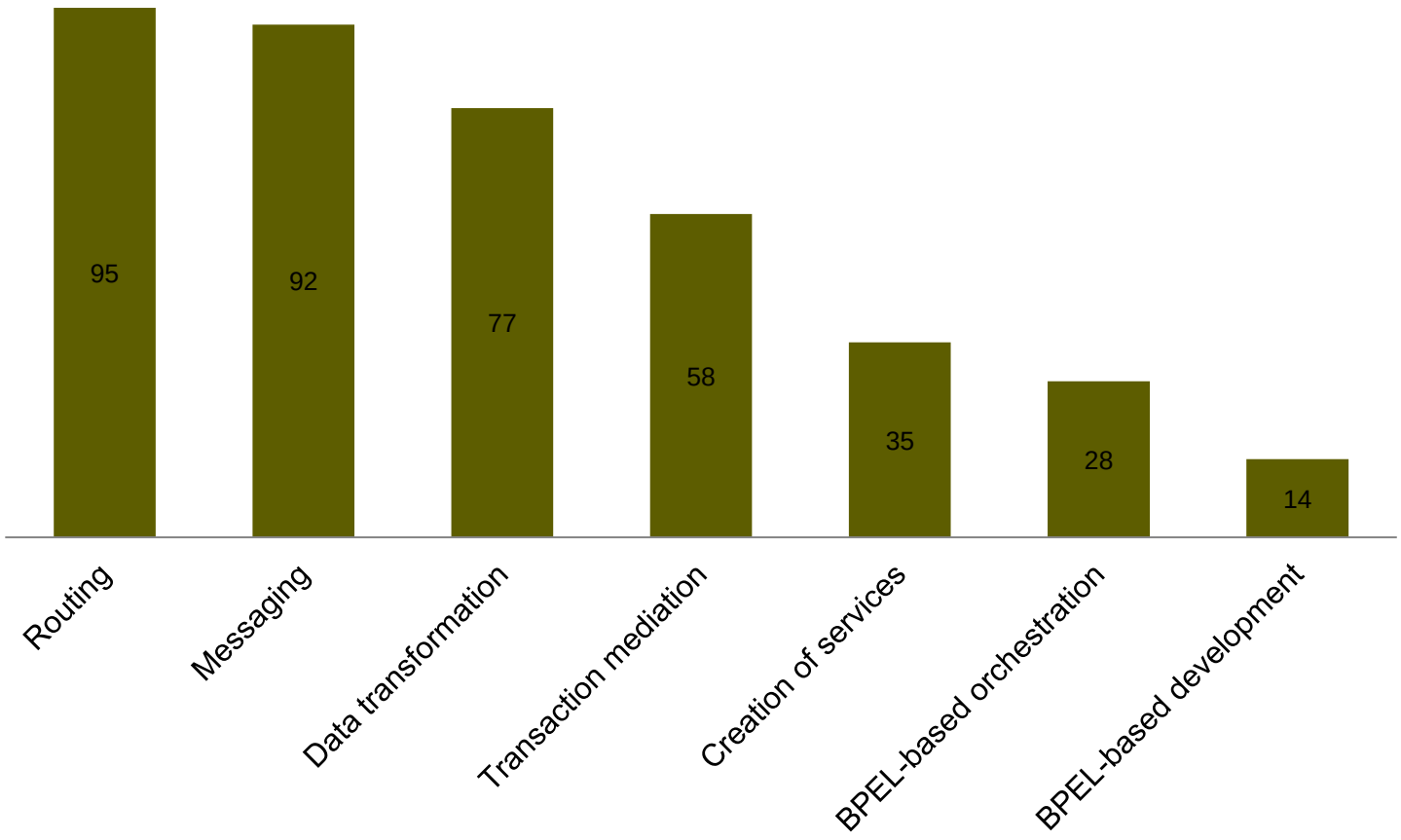
- Event handling - Guarantee event processing
- Protocol conversion - Transparently translate between communication protocols (e.g., HTTP, FTP, REST, SOAP, JSON, DCOM, CORBA, SAP RFC etc.)
- Mapping - Transfer between tabular data formats



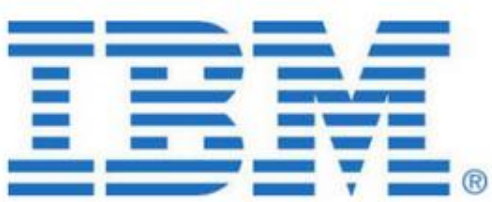
ESB

- Translation and transformation - Change data content based on rules
- Queuing and buffering - Handle differing data processing speeds between sender and receiver

ESB Usage


95
92
77
58
35
28
14
Routing
Messaging
Data transformation
Transaction mediation
Creation of services
BPEL-based orchestration
BPEL-based development

Enterprise Service Bus (ESB) Tools: Technical Comparison and Review



ESB

	Forrester's Weighting	FuseSource	IBM (WESB)	MuleSoft	Oracle	Progress Software	Red Hat	Software AG	Tibco Software	WSO2
CURRENT OFFERING	**50%**	**4.6**	**3.68**	**4.35**	**4.74**	**4.78**	**3.21**	**4.89**	**4.71**	**4.47**
Architecture	40%	4.88	3.92	4.7	4.58	5	2.9	4.88	5	4.48
Orchestration	10%	5	3	2.5	5	5	2.5	5	4	4
Mediation	20%	3.89	3.77	4.14	4.76	4.27	3.98	4.69	4.63	4.75
Connection	10%	4.7	4.6	5	4.6	4.6	3.33	5	4.6	4.26
Change and control	20%	4.52	2.99	4.47	5	4.84	3.37	5	4.6	4.49
STRATEGY	**50%**	**3.48**	**3.1**	**2.78**	**4.2**	**4.05**	**3.05**	**4.6**	**4.45**	**2.95**
Product strategy	50%	4.2	3.6	3.2	5	4.7	3.7	5	4.7	4.5
Strategic alliances	10%	1.8	5	3.8	5	5	4	5	5	3
Customer reference checks	40%	3	2	2	3	3	2	4	4	1
MARKET PRESENCE	**0%**	**2.2**	**3.45**	**1.73**	**4.55**	**3.23**	**1.45**	**4.78**	**4.63**	**1.03**
Installed base	40%	2.5	3	1.5	5	3	1	5	5	0.5
New customers	30%	2	2.5	2	4	3	1	5	5	1
Delivery footprint	30%	2	5	1.75	4.5	3.75	2.5	4.25	3.75	1.75

Red Hat JBoss Fuse

- JBoss Fuse is more than an enterprise service bus (ESB).
- It is a lightweight open source integration platform – based on Apache ServiceMix – that is available on premise or in the cloud.
- Built on open standards, JBoss Fuse is bolstered by a large community of developers, rather than the small teams who typically maintain proprietary source code.

Red Hat JBoss Fuse

- Apache ActiveMQ: A fast, open source message broker that supports JMS as well as clients written in other languages like C and Python
- Apache Camel: An open source framework that provides implementations of tried and true EIPS (Enterprise Integration Patterns). This allows developers to leverage pre-existing solutions to frequently encountered coding challenges related to enterprise integration
- Apache CFX: An open source web services framework, which provides for communication using various standards such as JAX-WS and JAX-RS, HTTP and FTP, as well as different formats like JSON, XML, CSV, etc.
- Apache Karaf: An OSGI runtime container for deploying applications
- Fabric8: An orchestration tool for large middleware deployments

Mule ESB

- Mule ESB is low footprint, Java-based enterprise service bus.
- It is open source and like most ESBs, allows for the integration of systems via JMS, Web Services, HTTP, JDBC, and more.

Mule ESB

- AMQP (Advanced Message Queuing Protocol): Support is based on the RabbitMQ Java Client
- Routers: MuleSoft uses routers to split, combine, reorder, evaluate, and broadcast messages
- Anypoint Connectors: Pre-built protocol, database, transport, and database connectors. You can also build your own if needed
- Mule Runtime Engine: The heart of the MuleSoft Anypoint platform. Deployable in the cloud or on premise
- Mule Runtime Manager: Allows for the deployment, monitoring, and troubleshooting of Mule instances

Oracle ESB

- Oracle ESB is based on a prior Oracle product called Retail Integration Bus Essentials.
- It is intended to aid in communication between existing Oracle products and third-party applications.

Oracle ESB

- Oracle Message Broker: A JMS-compliant API that supports AQ, IBM MQSeries, TIBCo Rendezvous, and more
- Routing Service: SOA style routing services that allows for routing rules to be defined and published with a WSDL
- Integration Adapters: A set of JCA adapters available for download. These allow for communication with databases, message queues, various enterprise applications, and over various protocols
- ESB Server: The runtime server that listens to topics for updates
- ESB Control: Allows configuration changes to be made in real time

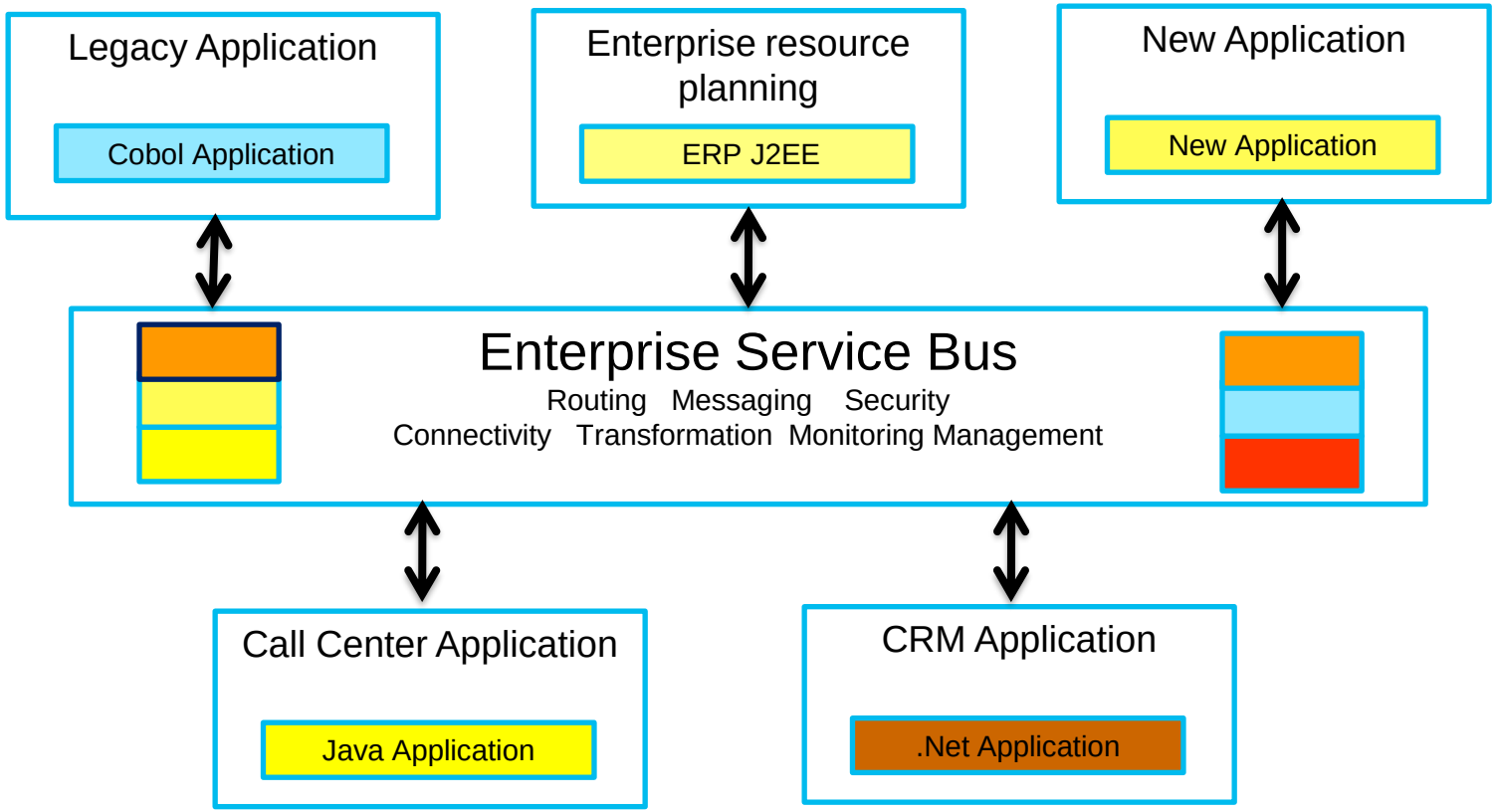
Microsoft BizTalk

- Biztalk is Microsoft's Inter-Organizational Middleware System.
- In other words, it's basically an ESB.
- It leverages .NET and allows developers to write their integration pieces in Visual Studio.

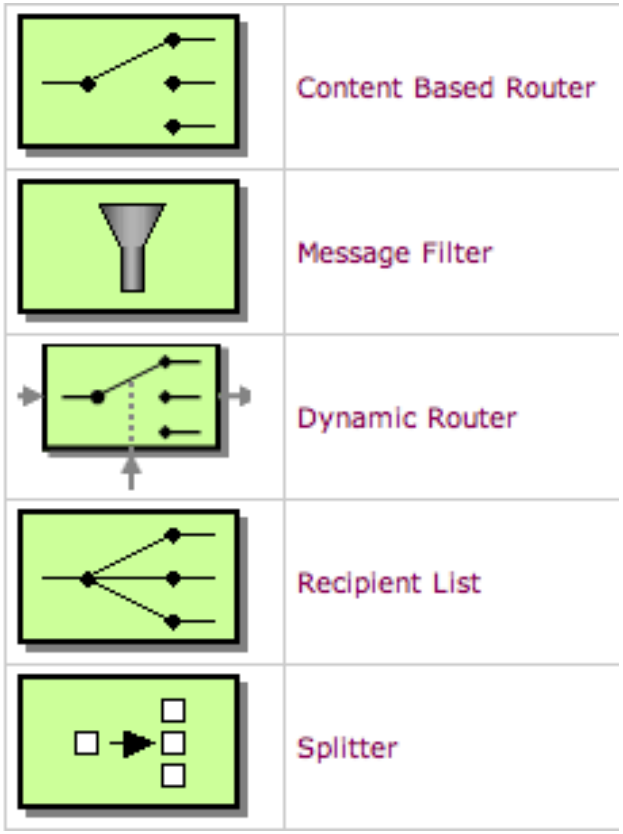
Microsoft BizTalk

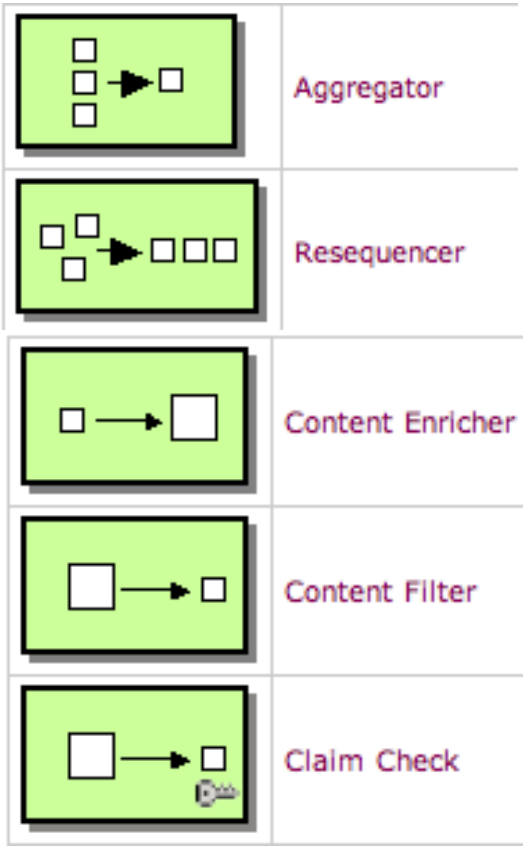
- MSMQ (Microsoft Message Queuing): First released in 1997, this message queue implementation is still available for installation on current versions of Windows Server
- Routing: Message/routing specifications are implemented through XML, but generally this XML is generated using graphical tools
- Adapters: BizTalk has a variety of built-in adapters. As expected, has great adapter support for Microsoft technologies such as the various WCF protocols
- BizTalk Server: BizTalk requires IIS (Internet Information Services) for various functionality such as HTTP, SOAP, SSL and more. Typically, this is deployed on Windows Server
- BizTalk Server Administration Console: This is a MMC (Microsoft Management Console) that allows for extensive configuration and management of the server.

After ESB


Legacy Application
Cobol Application
Enterprise resource planning
ERP J2EE
New Application
New Application
Enterprise Service Bus
Routing Messaging Security
Connectivity Transformation Monitoring Management
Call Center Application
Java Application
CRM Application
.Net Application

Enterprise Integration Patterns





ESB Functionalities

- VETRO, to summarize the ESB functionality:
  - V – validate the schema
  - E – enrich
  - T – transform
  - R – route (either itinerary or content based)
  - O – operate (perform operations; they run at the backend)

Mule 4

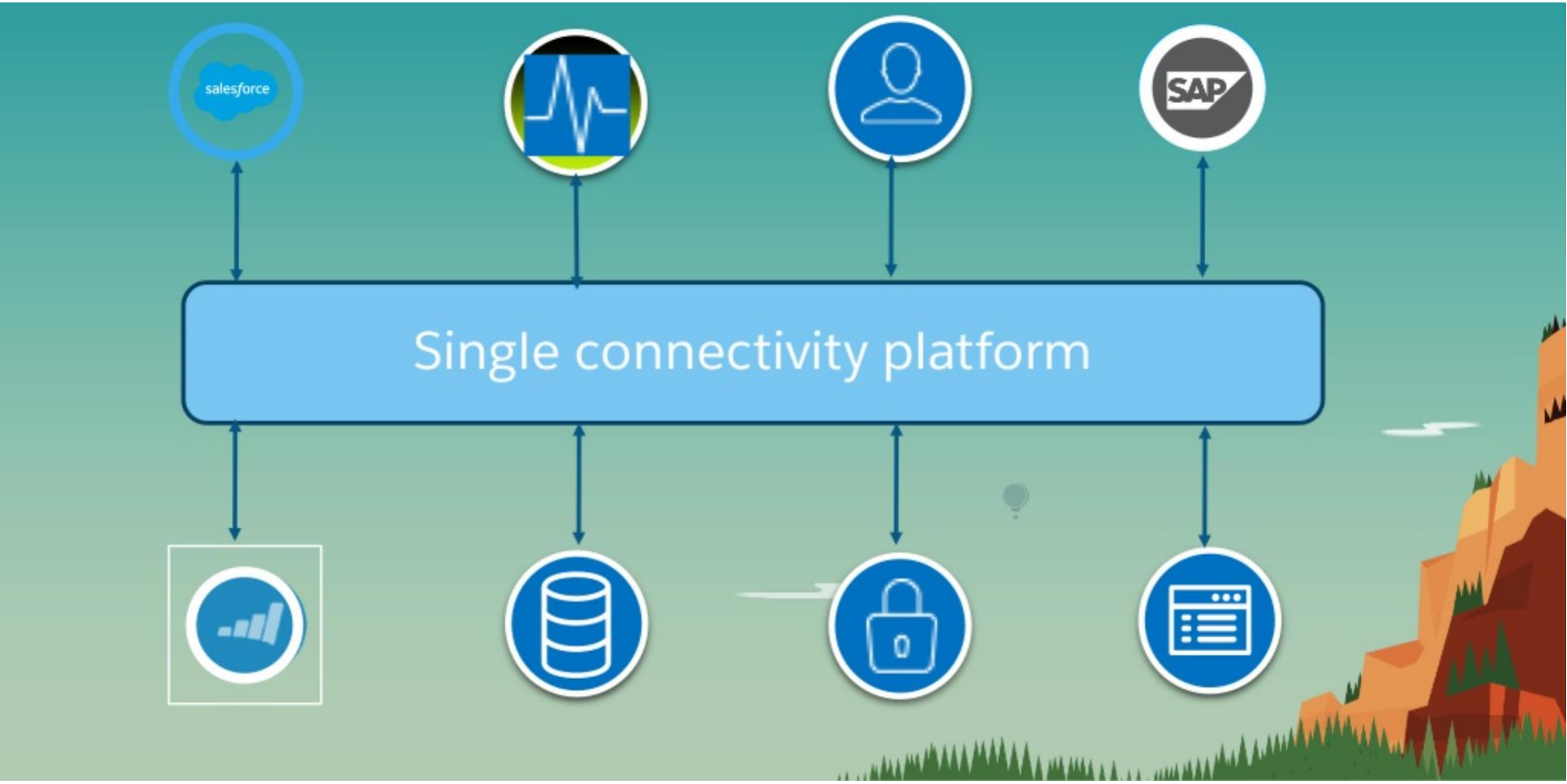
Use-Case Scenario:

- Multiple Vendors Sell their Products
- Vendors send the Product Catalog in CSV, JSON or XML
- Vendors send the Rate Card in CSV, JSON or XML
- Each Vendor has own Business Rules and Promotions
- Users Purchases Products Online.
- Orders to Comply 99% Delivery Assurance.
- All Orders must be tracked for Reference Purposes.

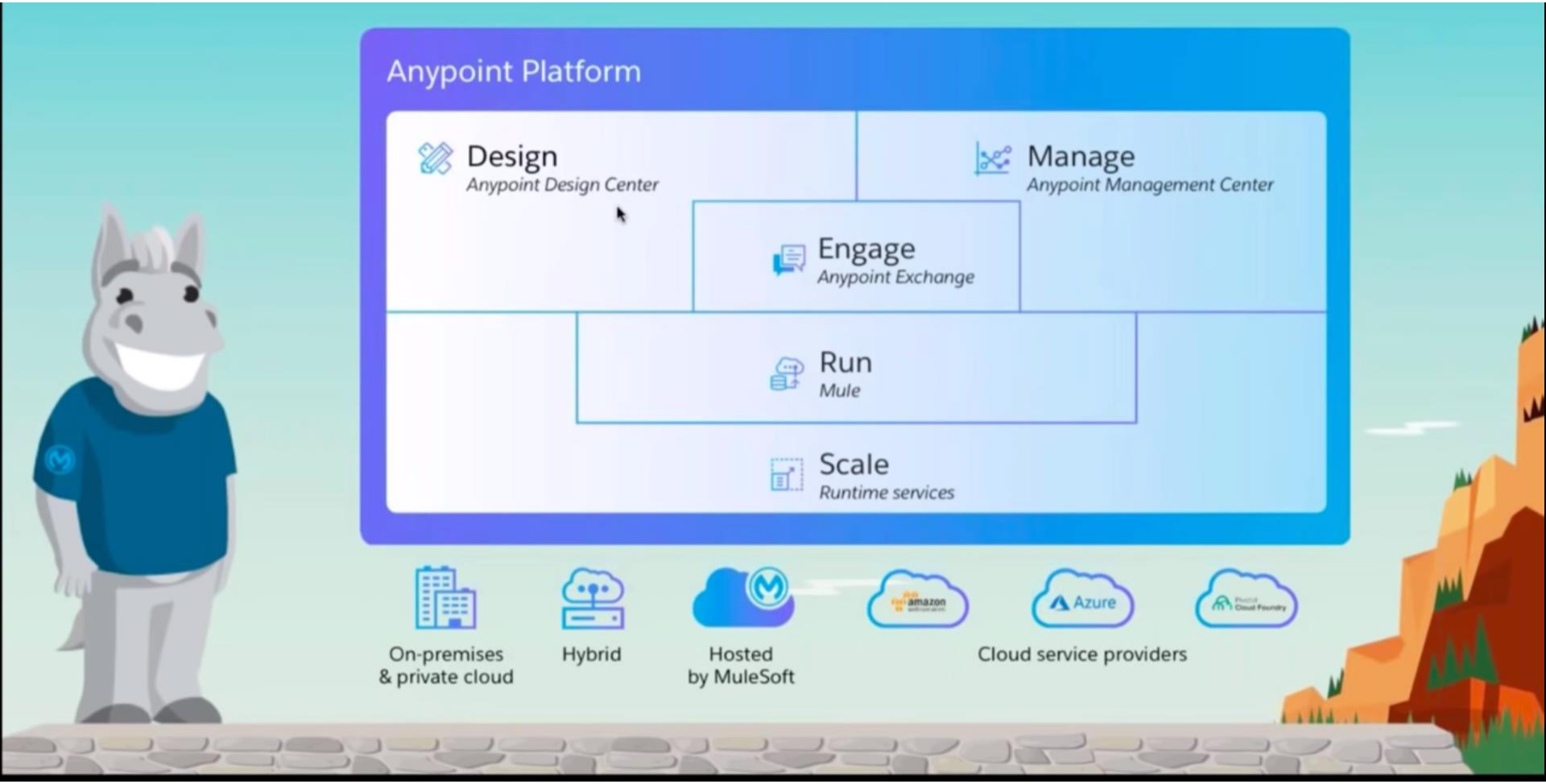
Mule Architectural Decisions

1.	Security	-	OAuth Multifactor Auth, HTTPs, API Policy
2.	API Layers	-	Proxy, Experience, Process & System
3.	Scalability	-	Micro-Services
4.	Reliability	-	Message Queue, Topic-Subscribers
5.	Maintenance	-	API Monitoring & Alerts
6.	Design Patterns	-	REST, Asynchronous, Broadcast, Proxy
7.	SLA Compliance	-	Rate Limiting, Reconnection, Conn Pooling

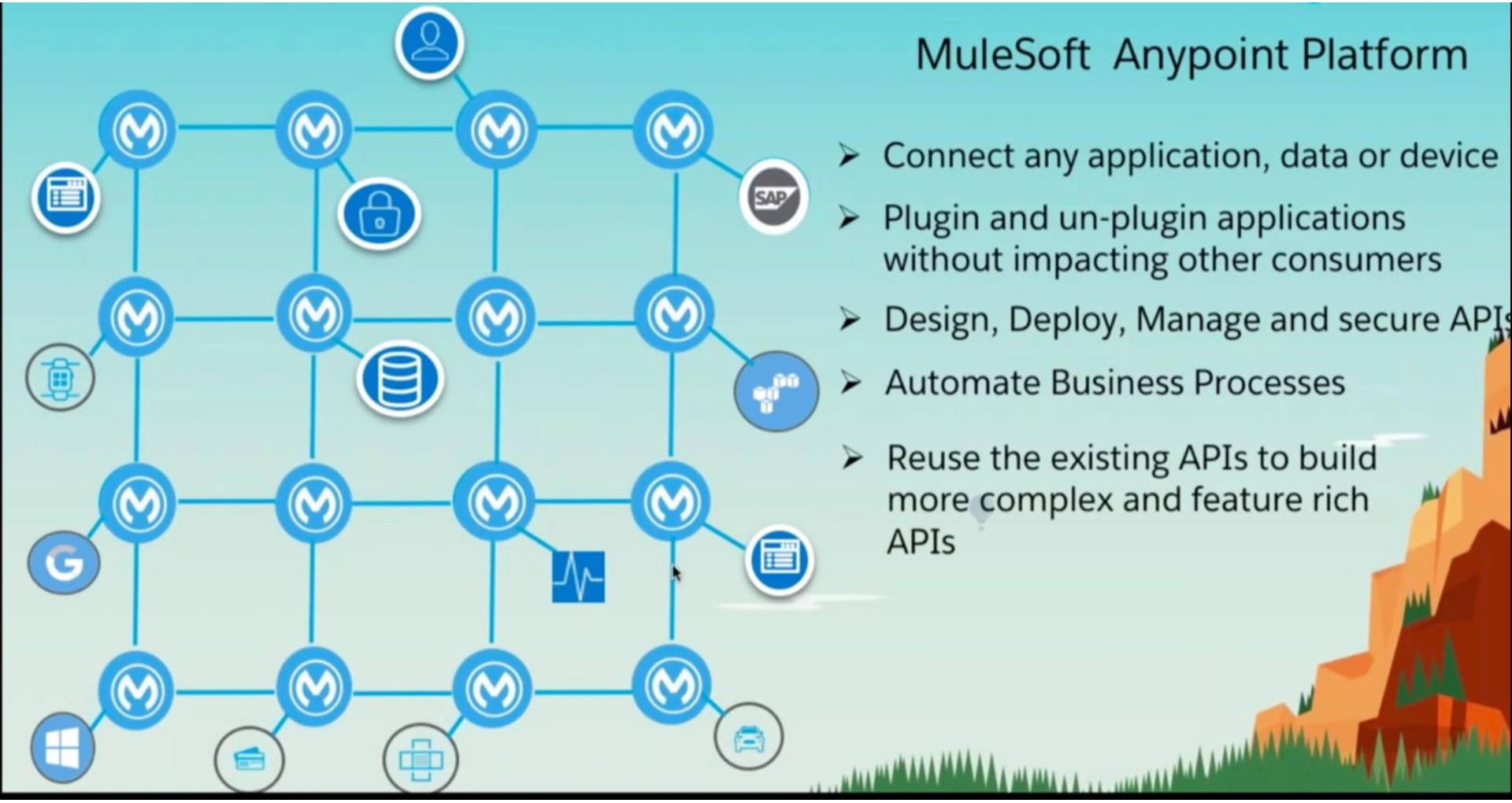
Integrations Made Simple



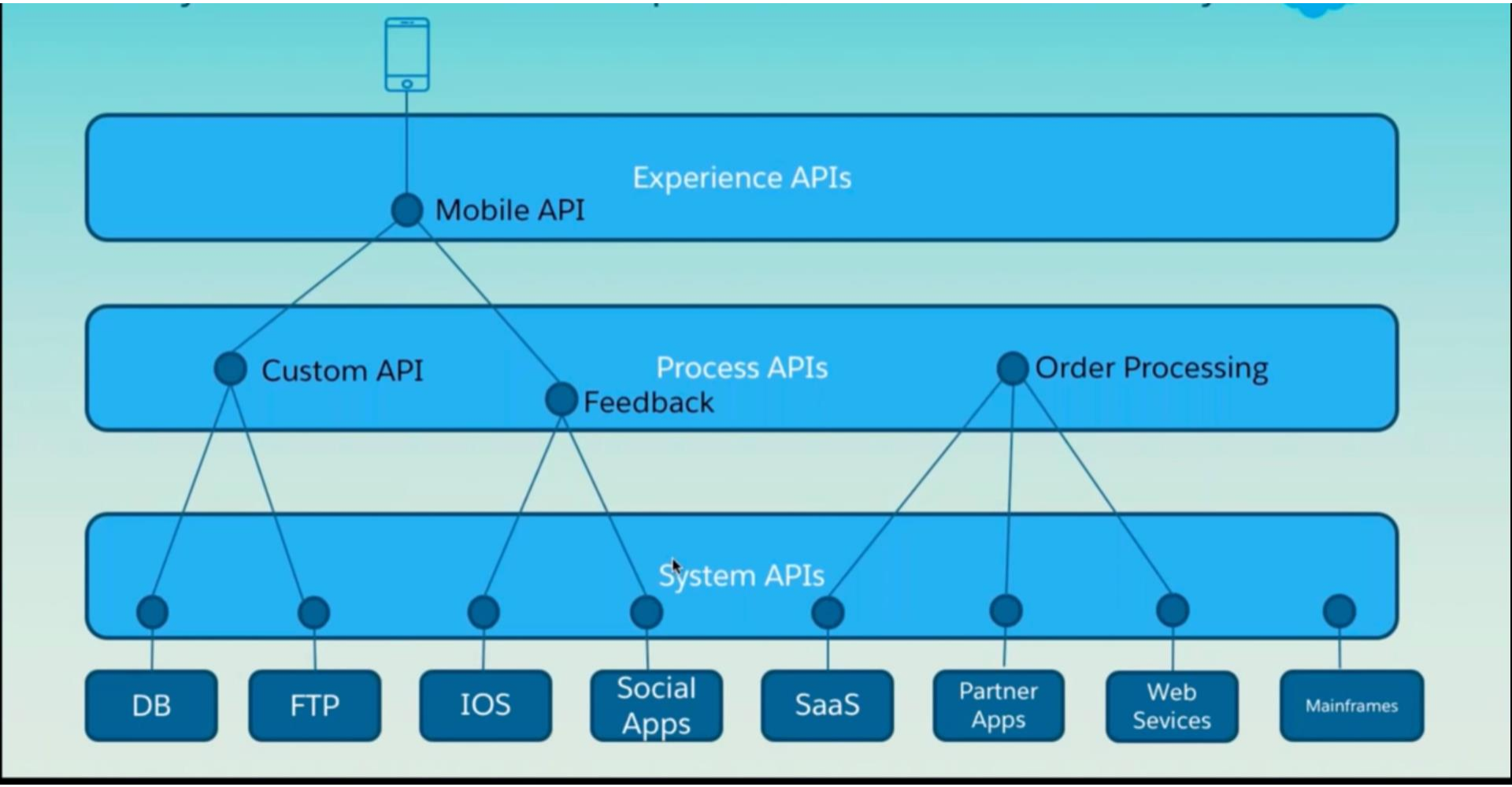
Mule soft Platform



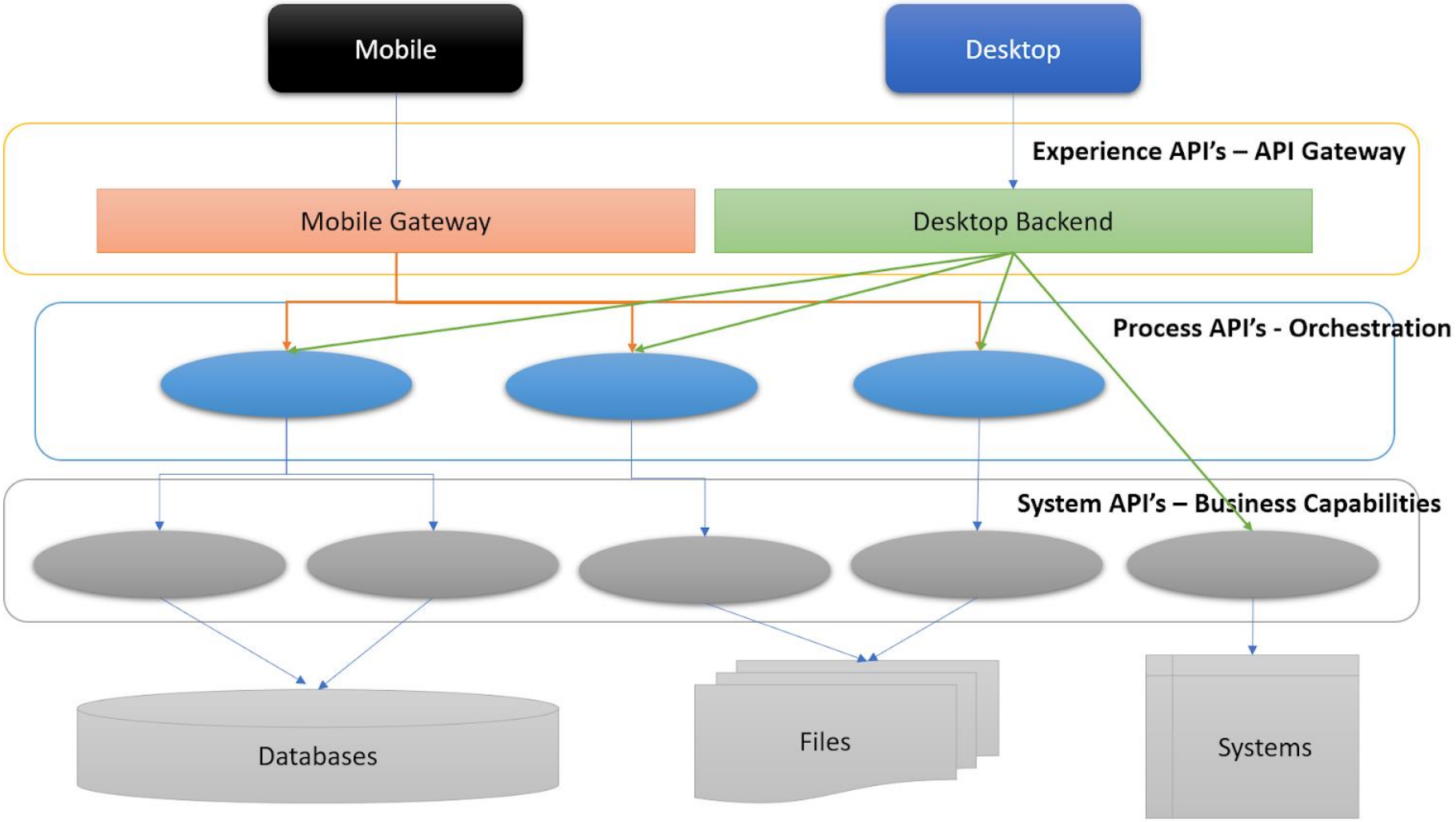
Application Network



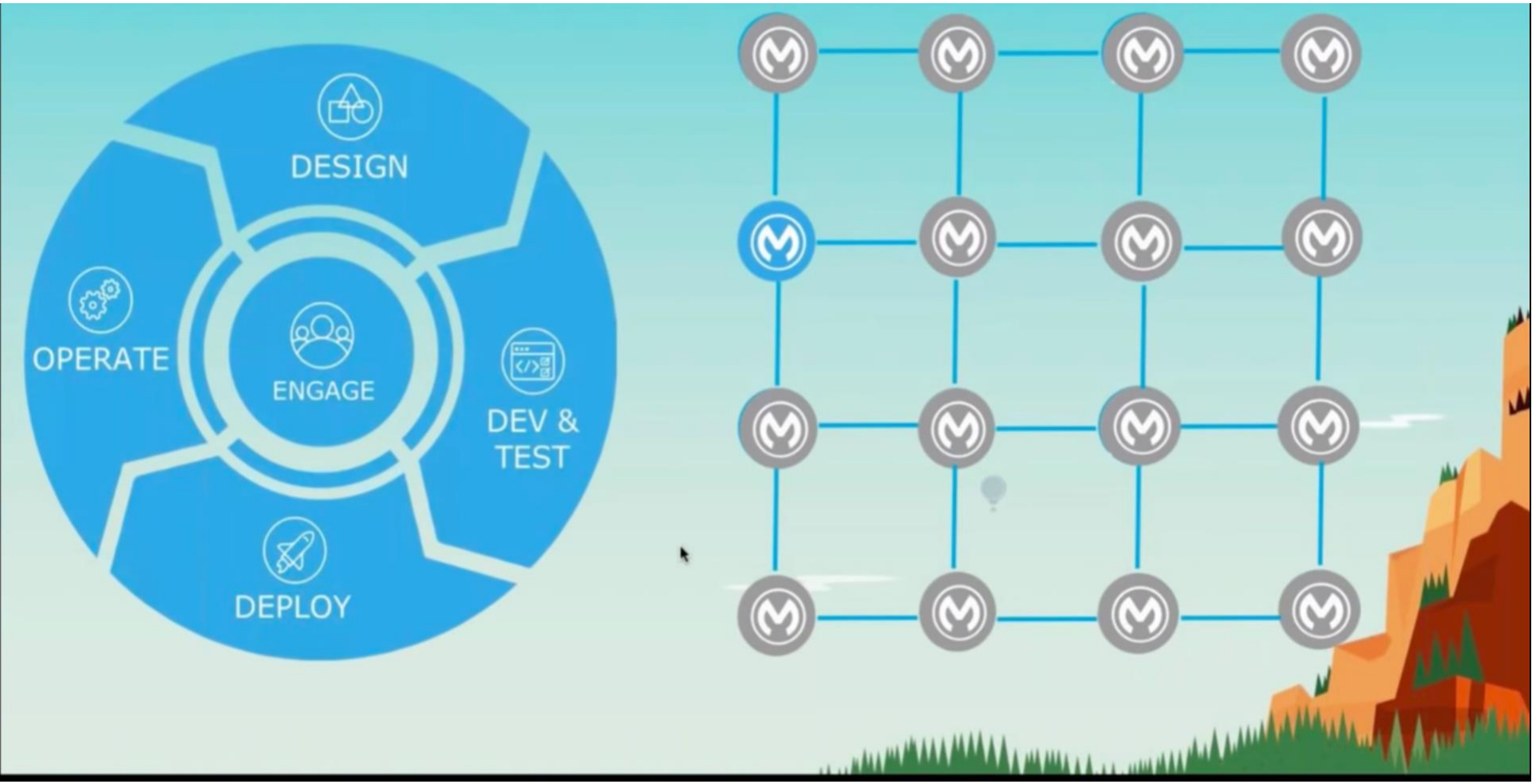
Multi-layered Architecture to implement API-led Connectivity



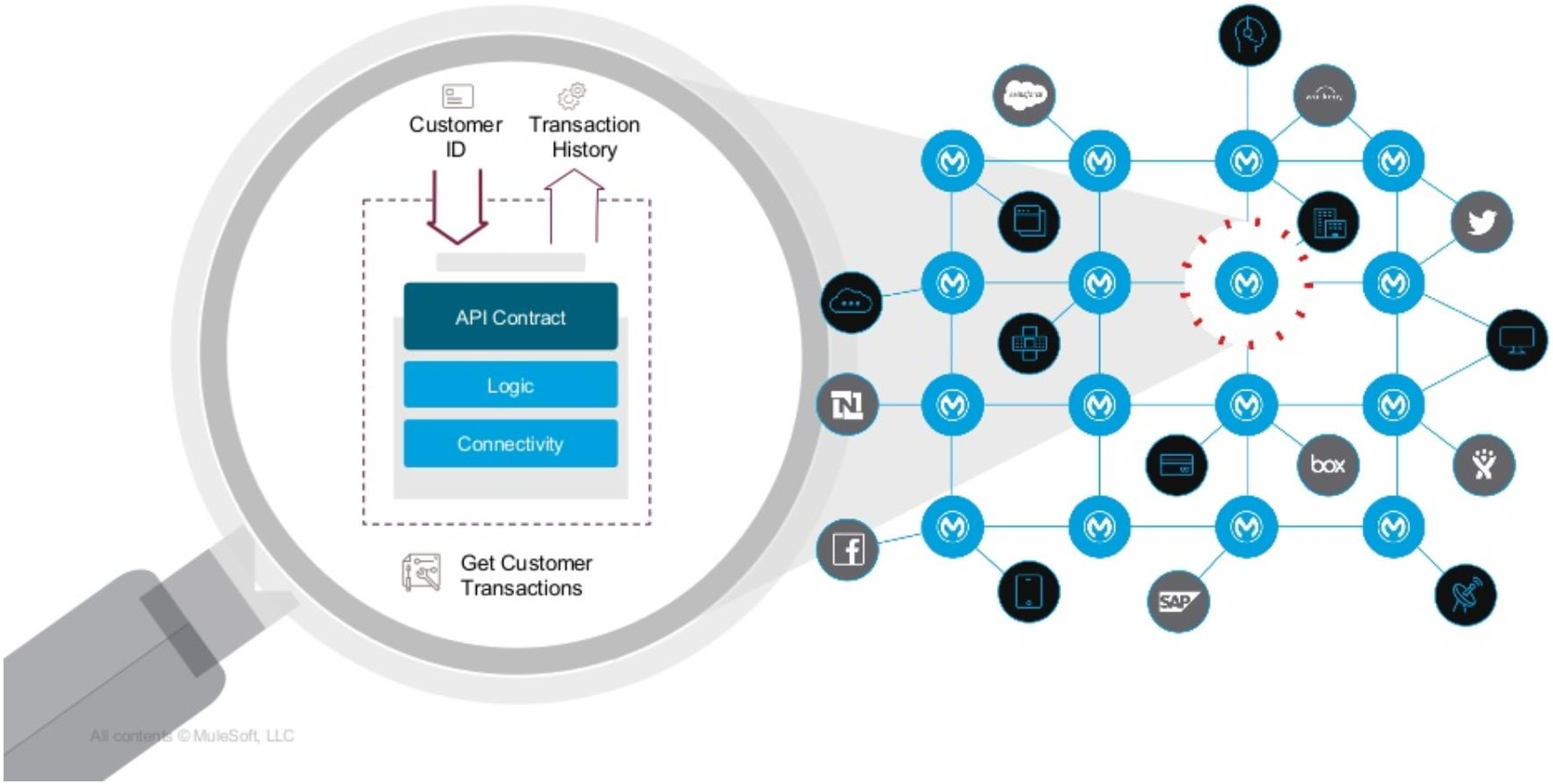
Multi-layered Architecture to implement API-led Connectivity



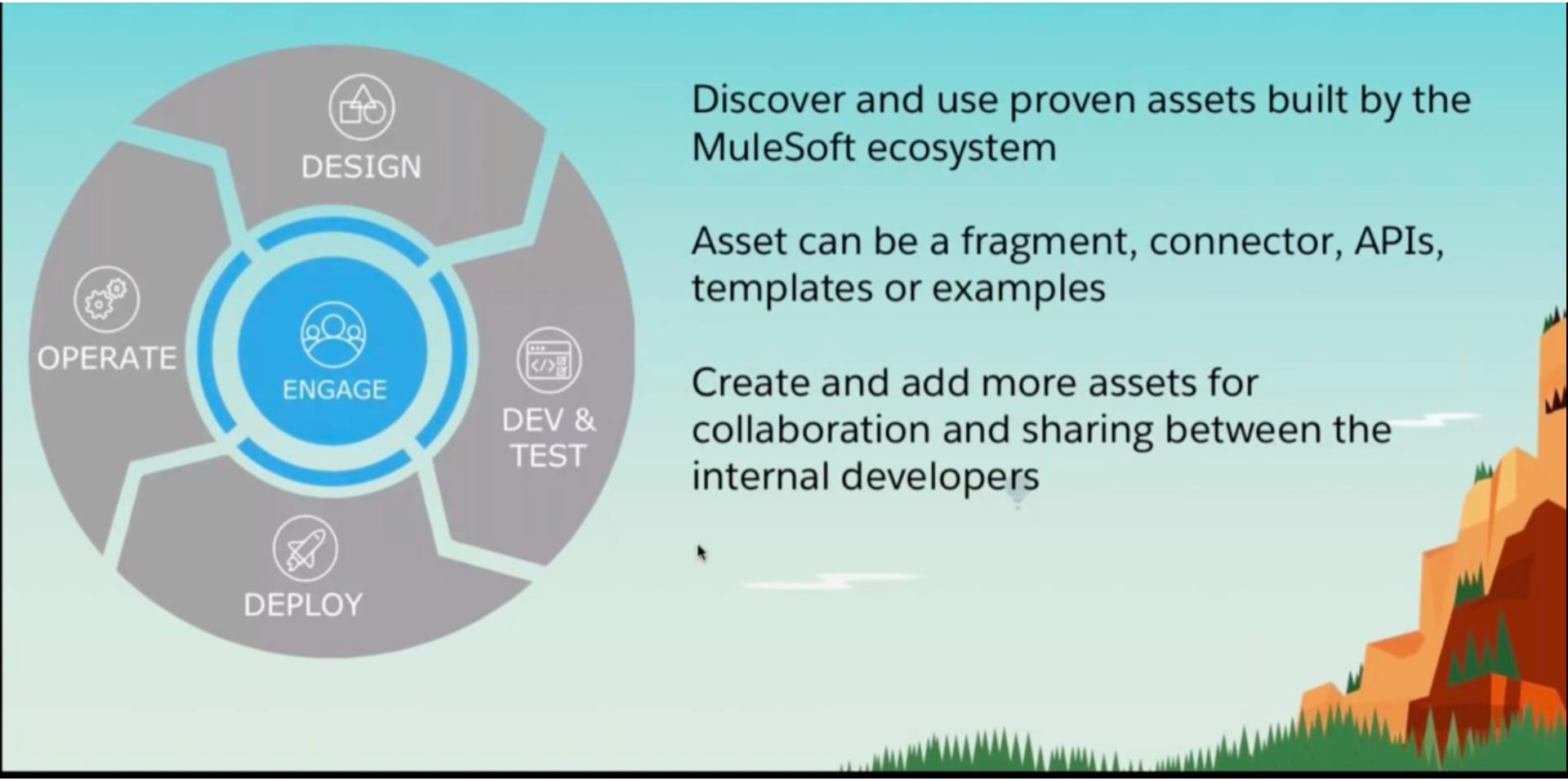
Lifecycle of an Application network node



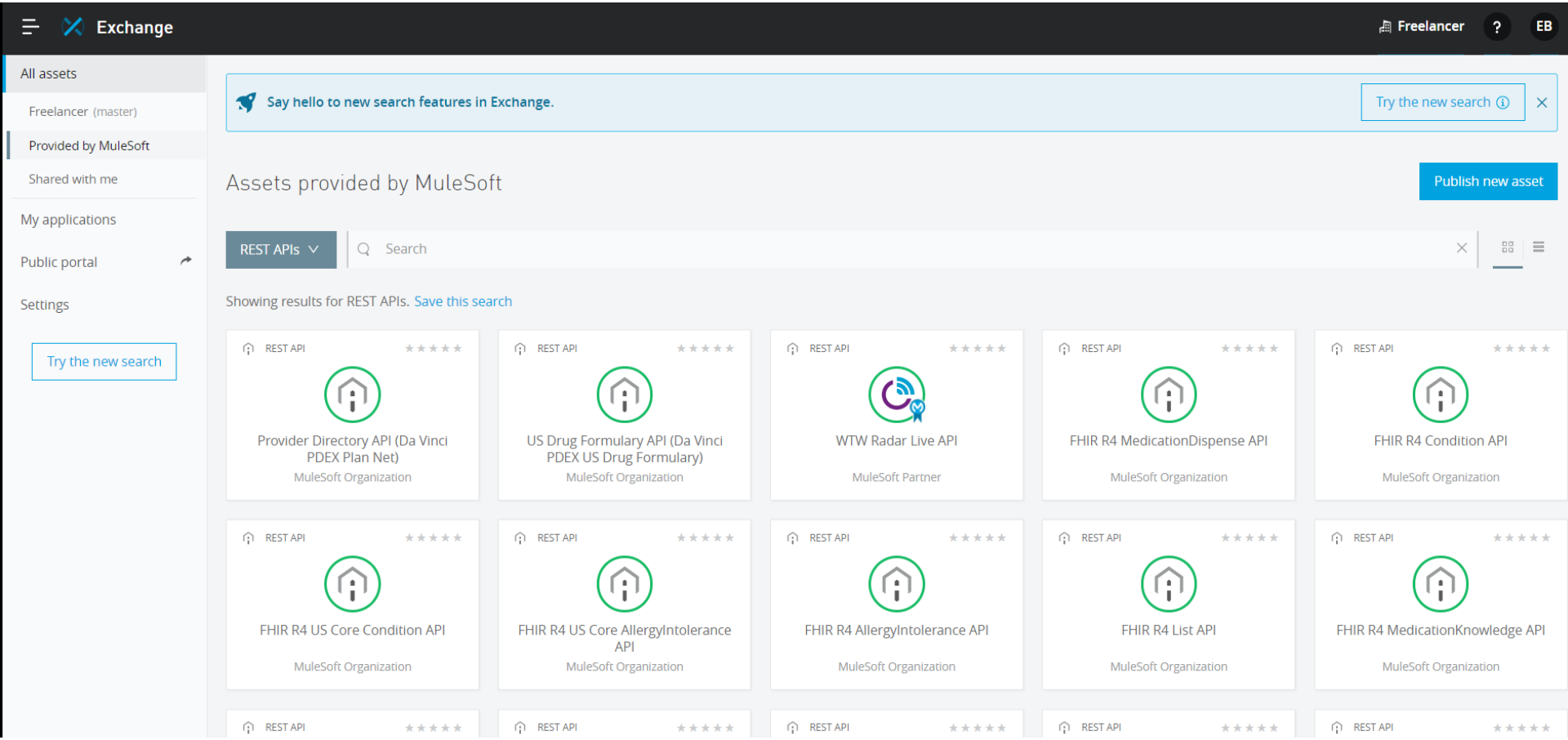
An application network emerges, which is built by modern API

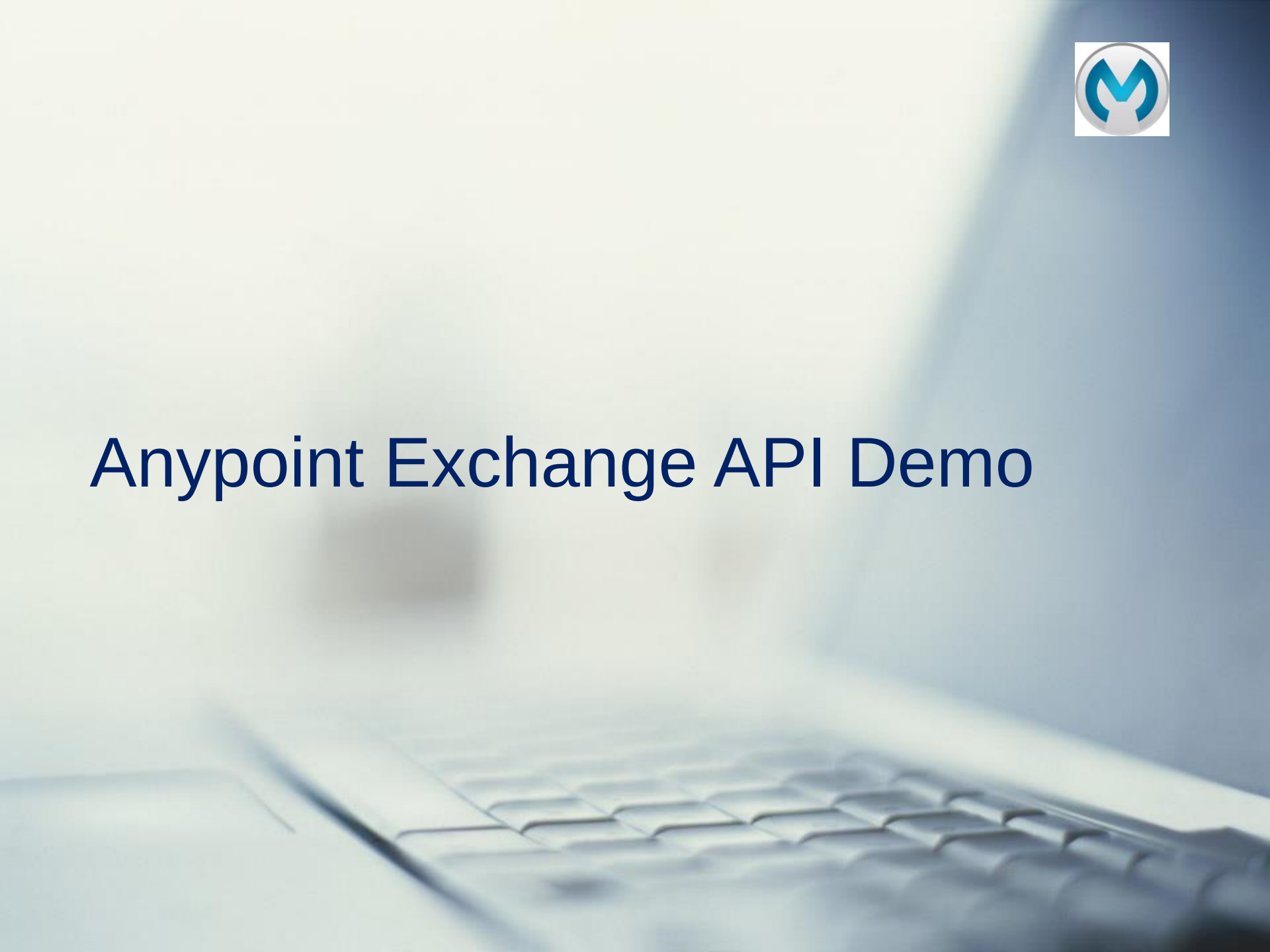


Engage with Any point Exchange



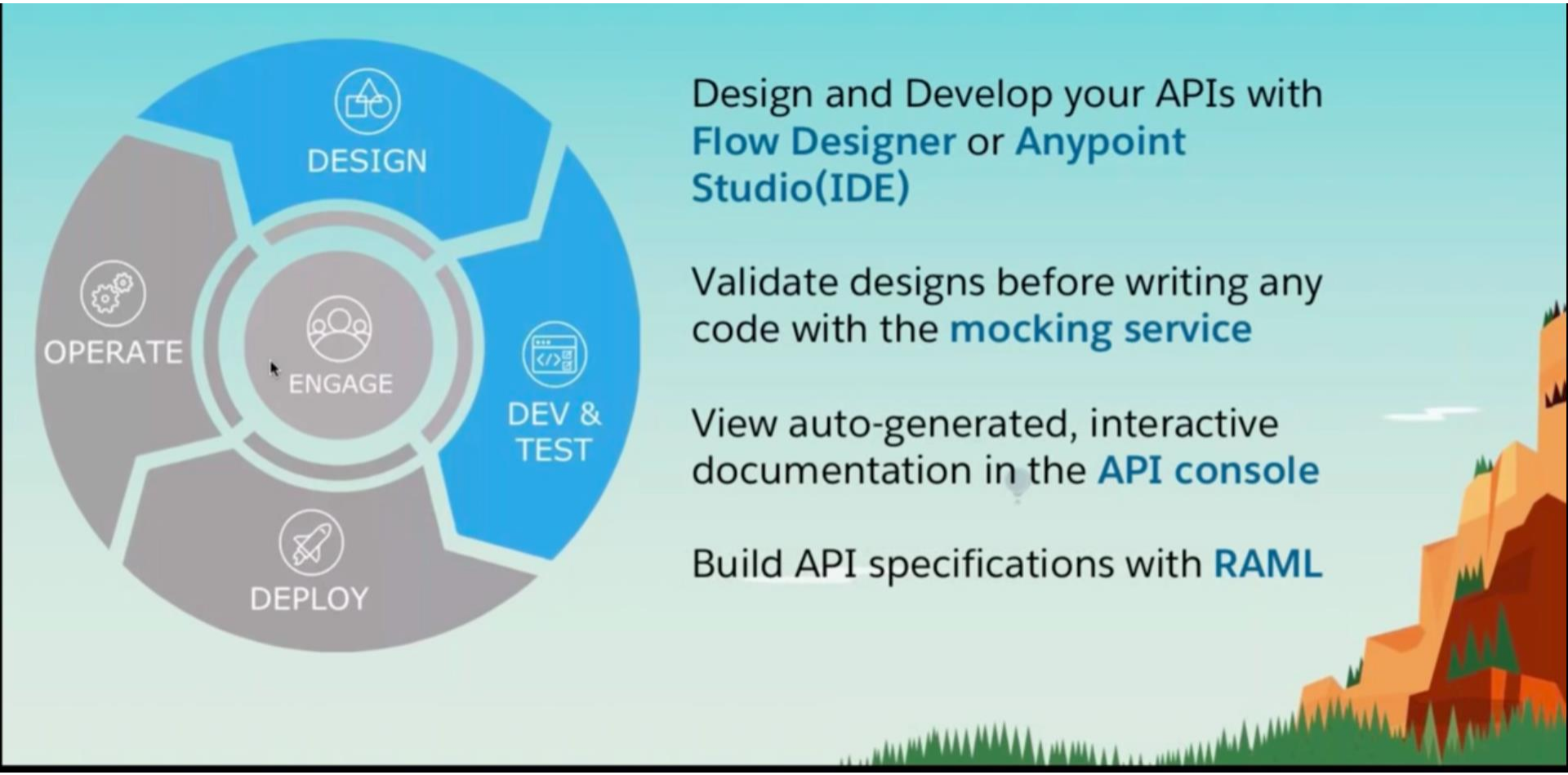
Assets Provided by Mulesoft

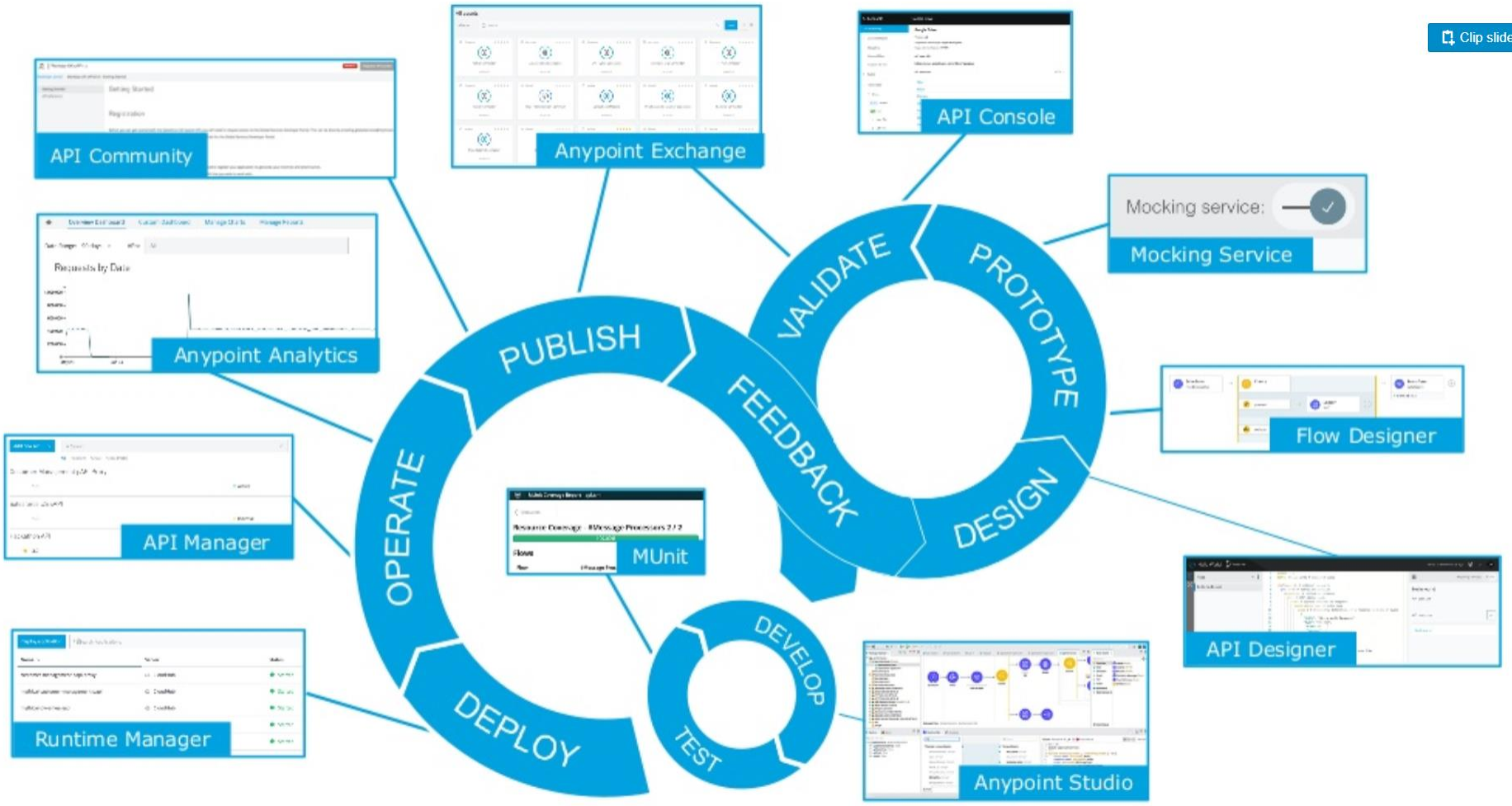




Anypoint Exchange API Demo

Design and Develop with Design Center




Clip slide
API Community
Anypoint Exchange
API Console
Mocking service:
Mocking Service
Anypoint Analytics
PUBLISH
VALIDATE
PROTOTYPE
FEEDBACK
DESIGN
OPERATE
DEPLOY
DEVELOP
TEST
Flow Designer
API Manager
MUnit
API Designer
Runtime Manager
Anypoint Studio

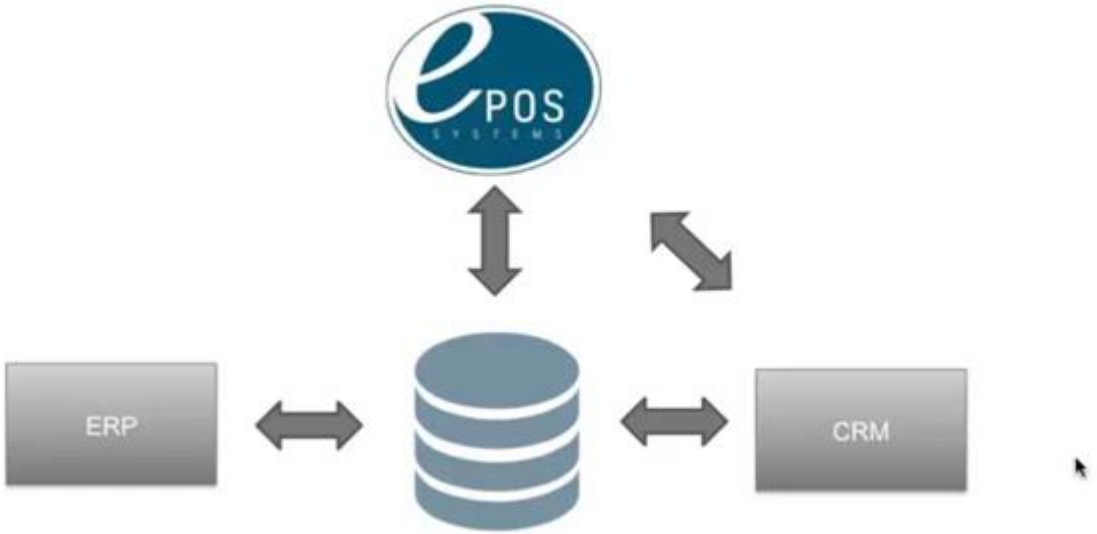
Problem Statement

ABC is a UK based luxury mobile manufacturing company that manufactures high quality luxury smart phones and the mobile accessories. ABC also provides a 24/7, dedicated concierge service that offers worldwide assistance, recommendations and priority bookings, personalized to the owner.

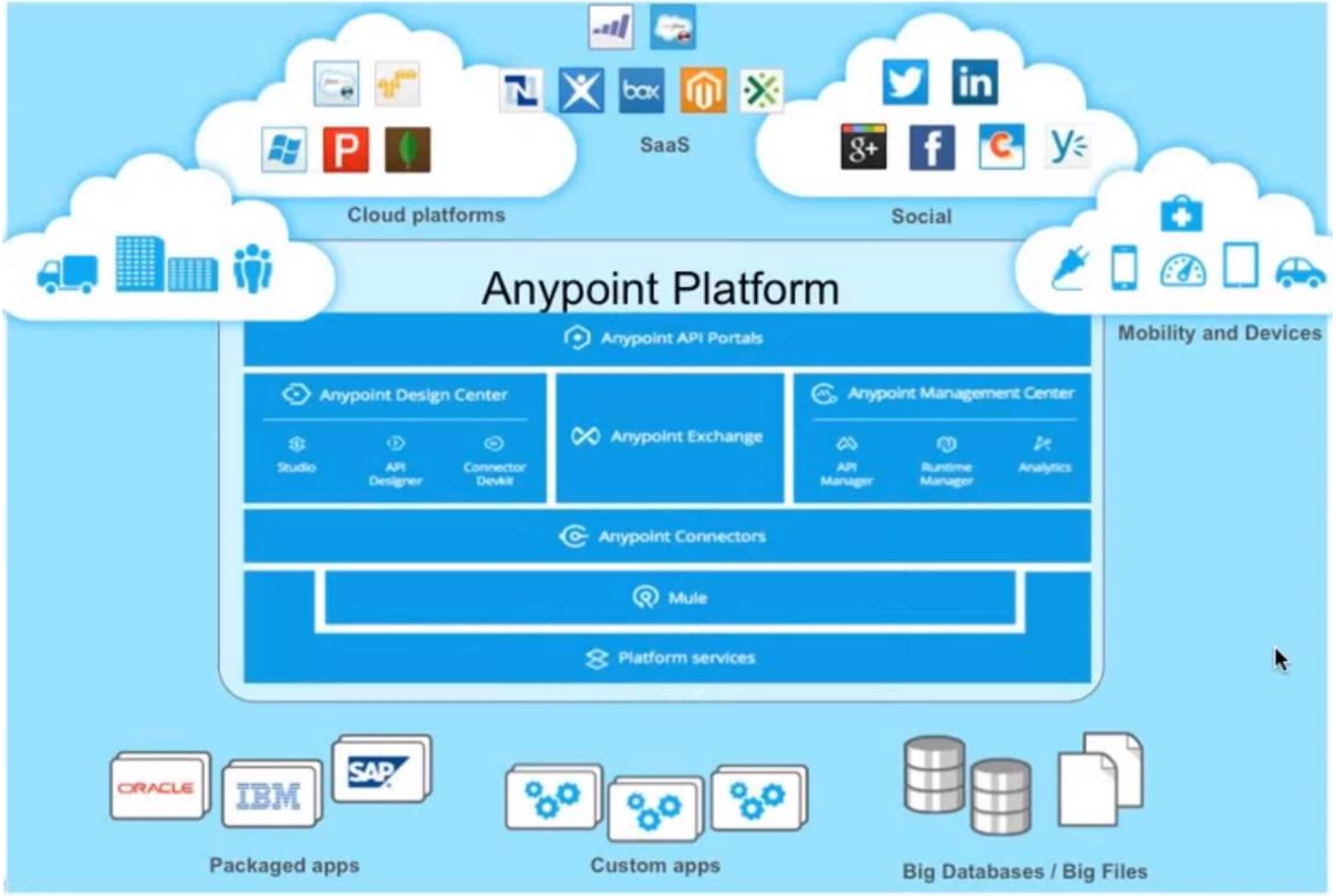
Their phones are currently sold in various stores across many countries. They are planning to launch an e-commerce site to promote and sell their phones and accessories online and make it available worldwide. They are also planning to provide a subset of their website's functionalities using a mobile app.

Key to the ongoing evolution ABC company wants to build an integration solution to address the need of unlocking some of their internal data for the use of their website and mobile devices via APIs and also accepting orders coming from various sources and store and process them through their internal systems. The company is also transitioning to follow agile practices where there will be multiple scrum teams focusing on their individual projects and delivery.

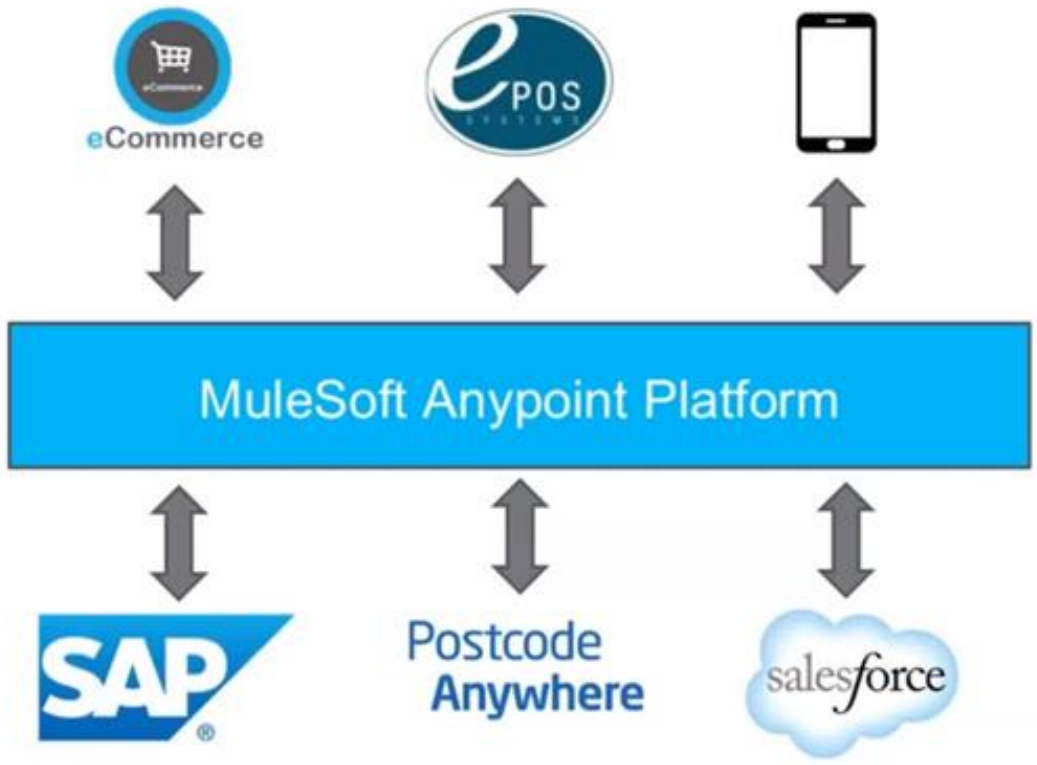
Current Landscape



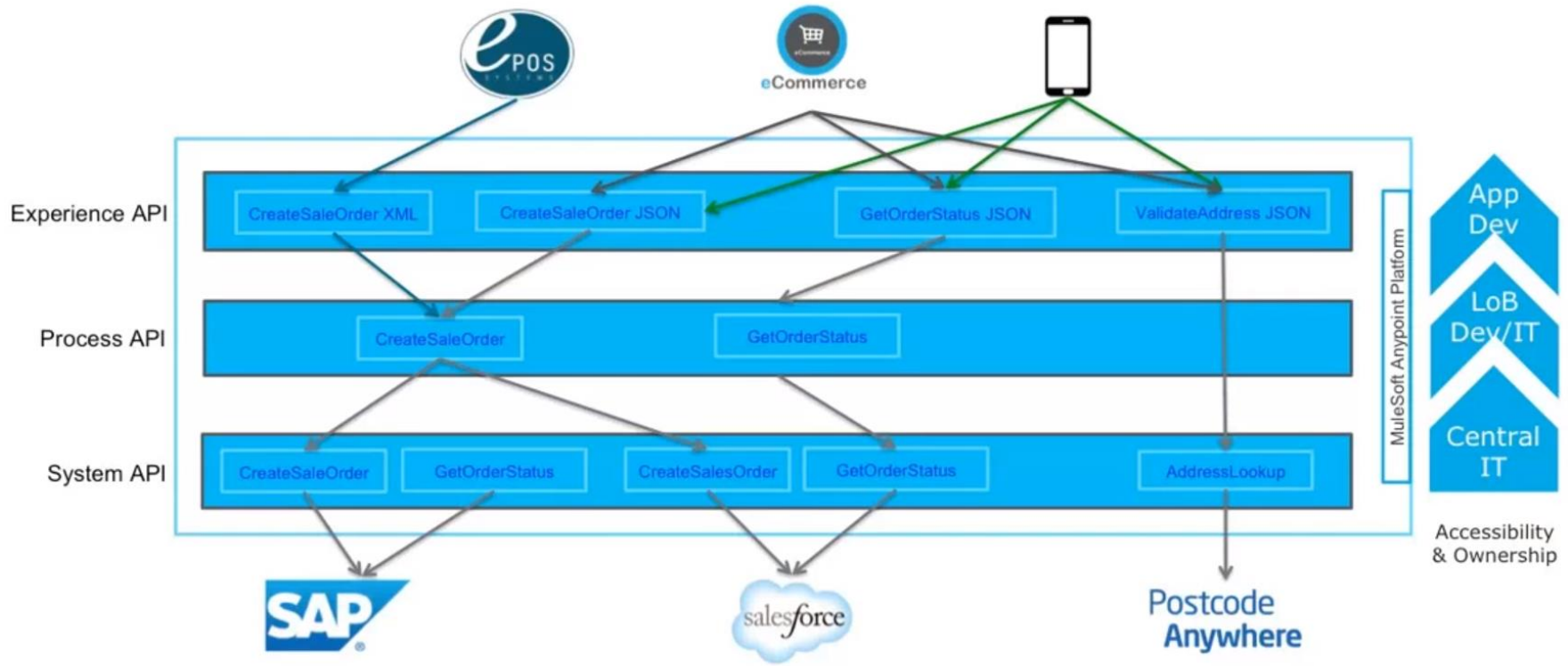
Proposed System



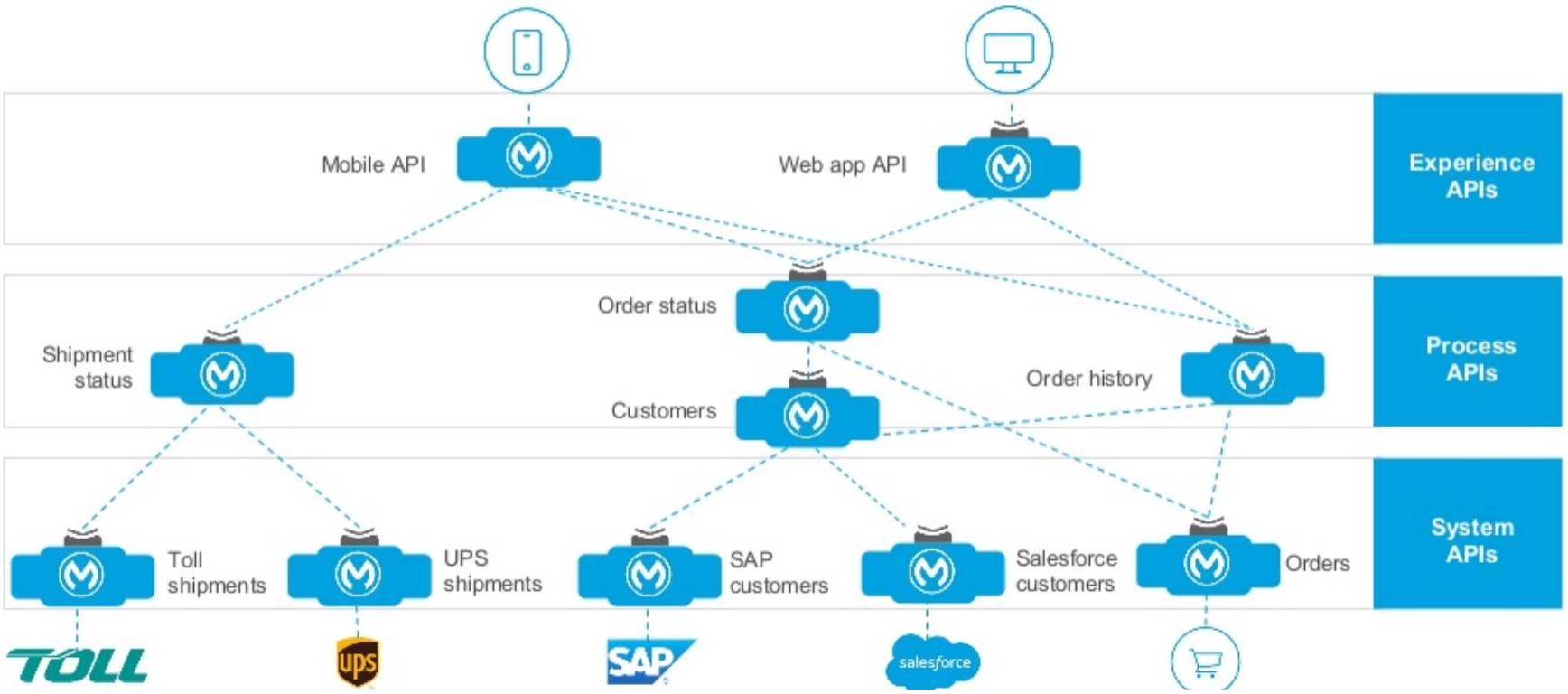
Solution Architecture



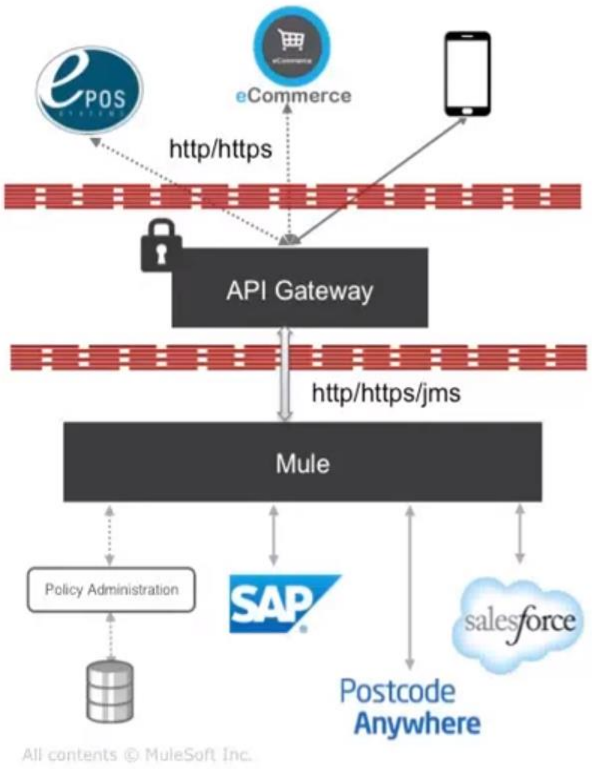
Integration Architecture



API led Connectivity – driving reuse and agility



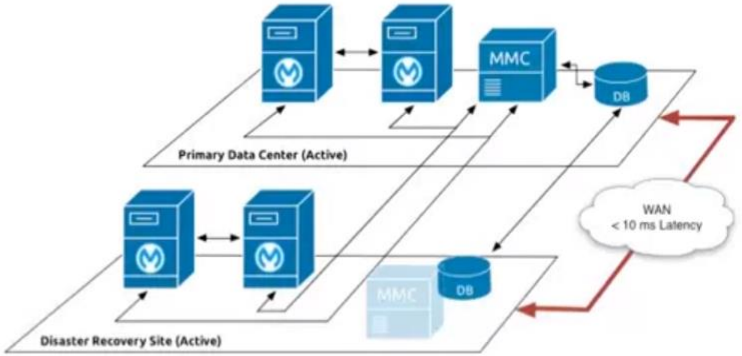
Deployment Architecture



Full On Premise deployment

Restrict the intranet mule to accept request from the gateway in DMZ

All interaction from Internet only goes to the gateway which applies the necessarily validation and policies.



HA - DR
Active/Active–Traffic intended for a failed node is either passed onto an existing node or load balanced across the remaining nodes. This group-oriented approach provides enhanced reliability, better load balancing, and scalability.

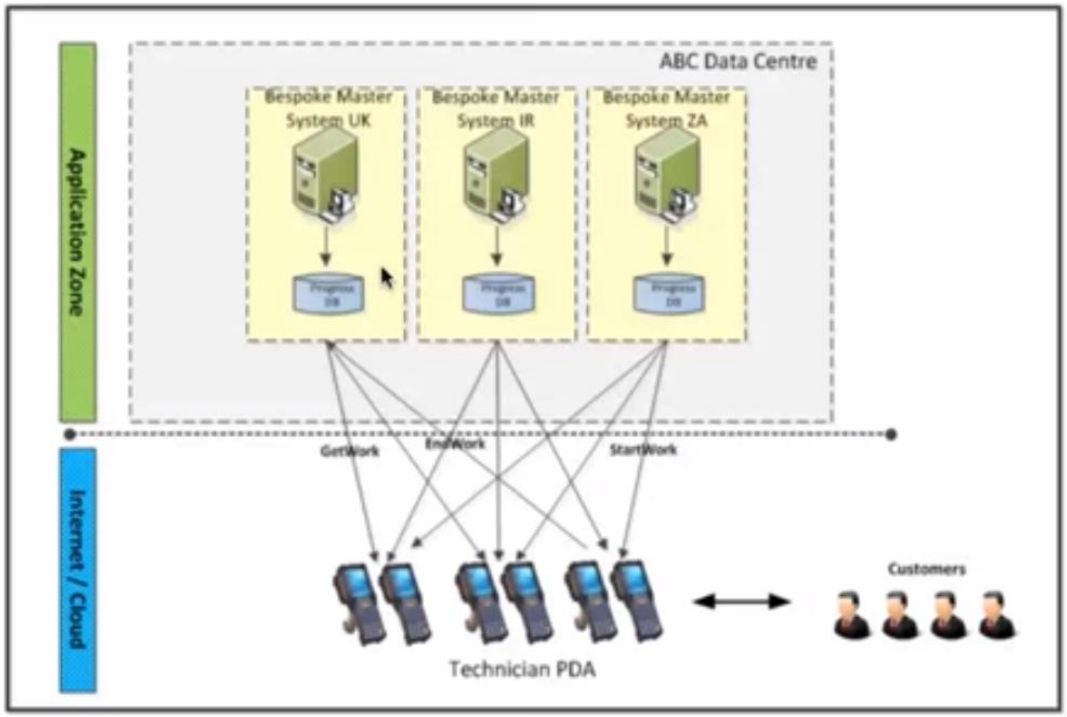
Business Problem

ABC is the manufacturer and expert in pest control. They are committed to provide practical, valuable advice and solutions for an effective pest management to their customer.
They have the existing PDA solution which is used by the field Technicians to provide effective pest management solution to their customer base however this PDA solution is having a point to point integration with different backend system including high level of manual processes.

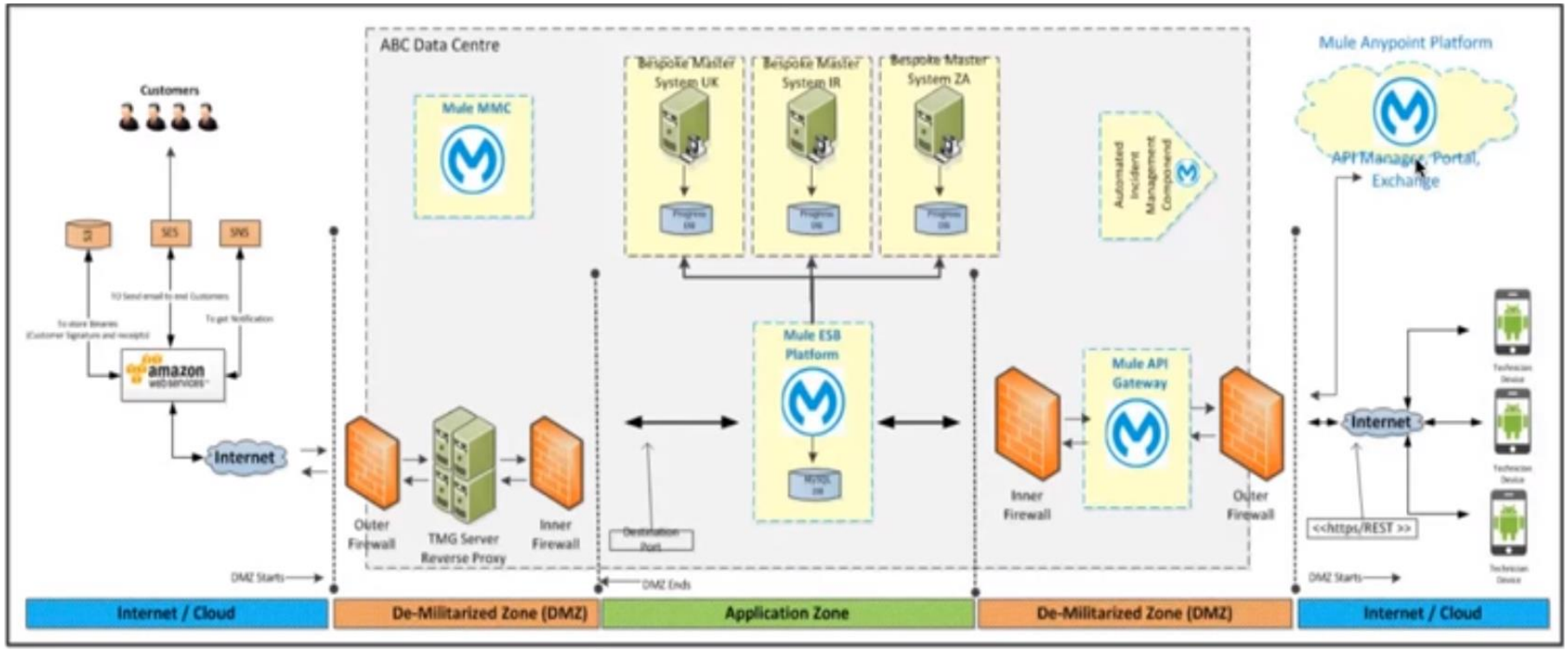
Functional areas they are looking with the new solutions are
- Start/End Day – To allow the start/end time of the Technician.
- Work View - Will allow the Technician to view the work that has been assigned to them.
- Start/End Visit - Will allow the technician to record the start/end time of a visit for productivity purposes, and to view the mandatory customer information prior to starting work.
- Proof of Service - Will allow the technician to trigger a Service Receipt to be emailed to the customer following a completed visit.

Current Solution

Below is the current solution used in ABC where systems are having point to point connectivity to exchange data -

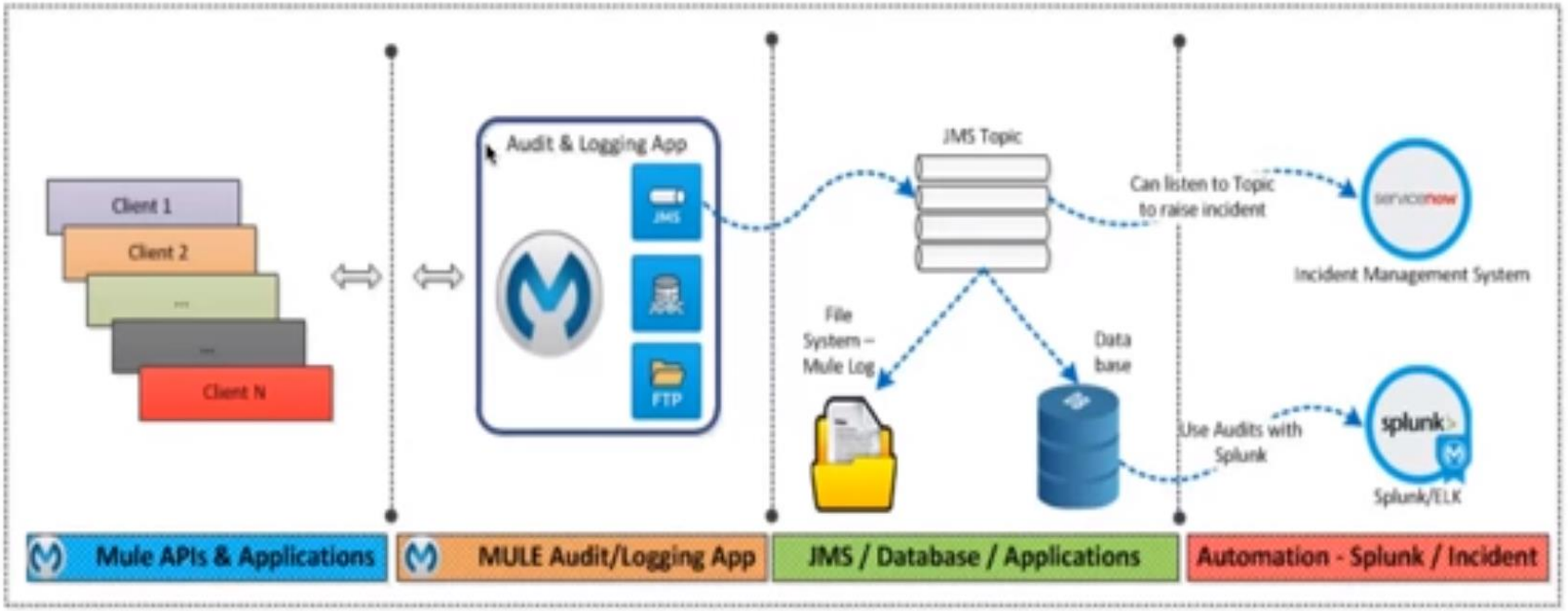


Deployment Architecture

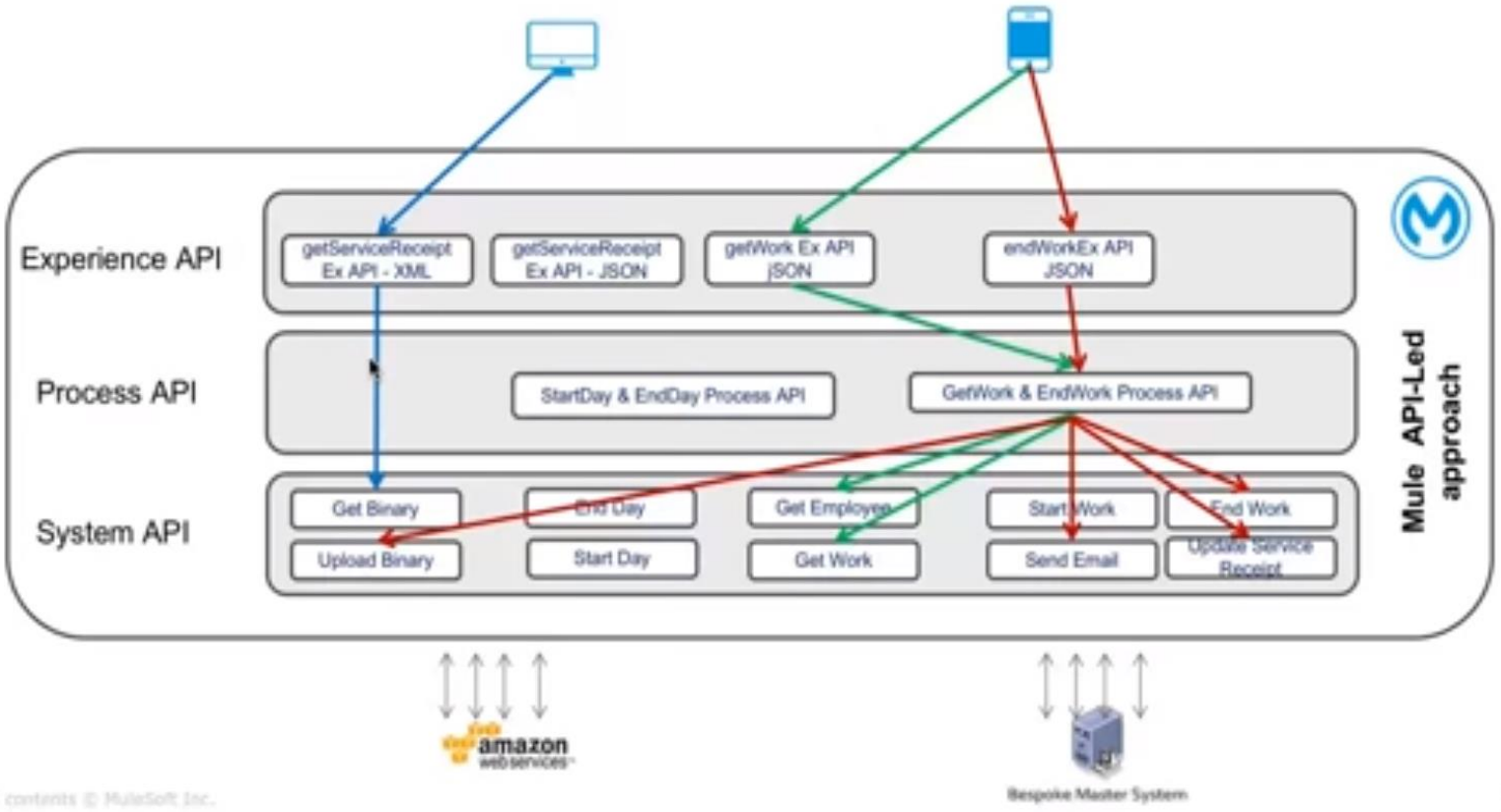


Current Solution

Below is the Solution Architect using MuleSoft On-Prim to Automate incident management for Operations to deal with the application error -

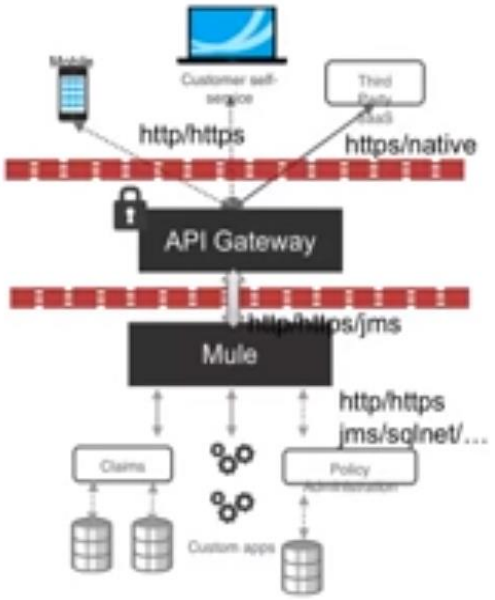


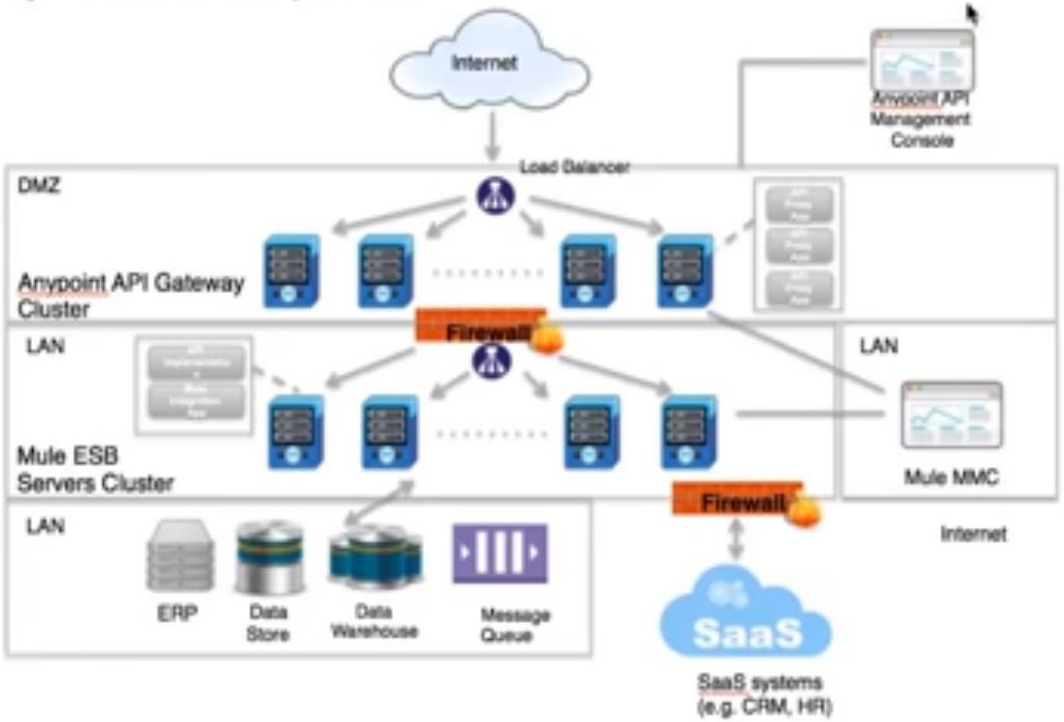
Integration Architecture



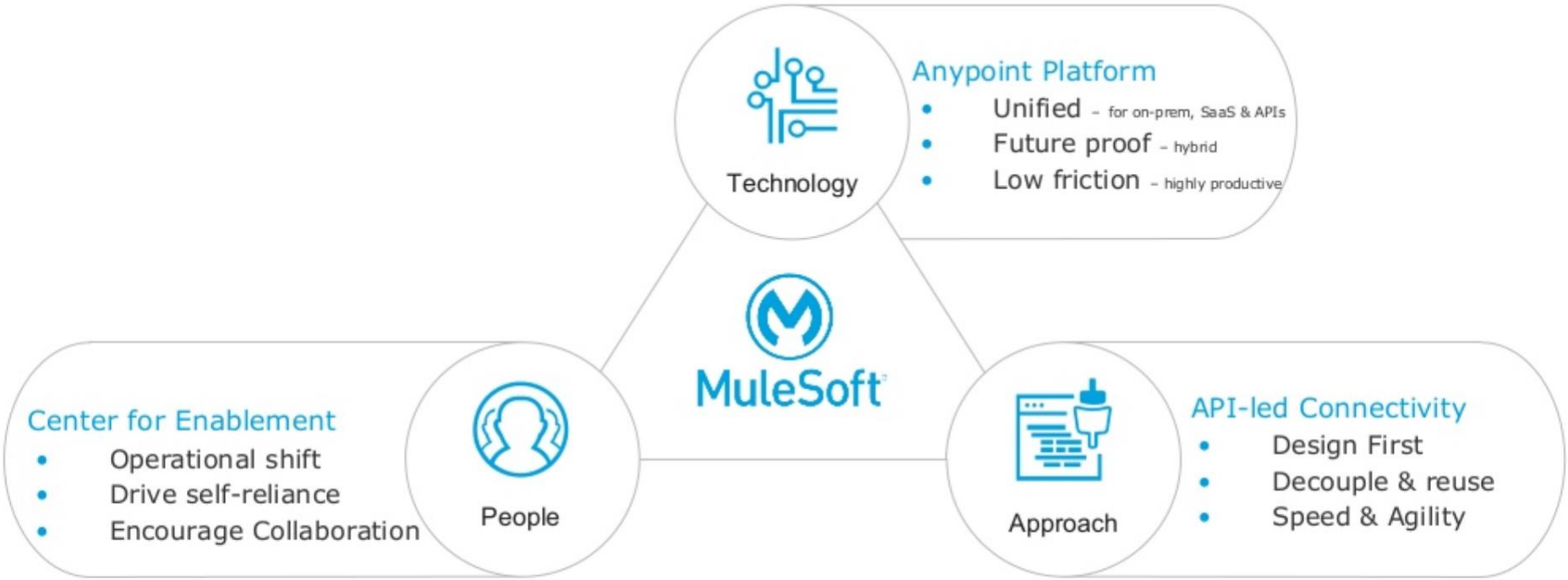
Deployment Architecture

Deployment with DMZ (Full on-prem + DMZ) - All interaction from Internet only goes to the gateway which applies the necessarily validation and policies



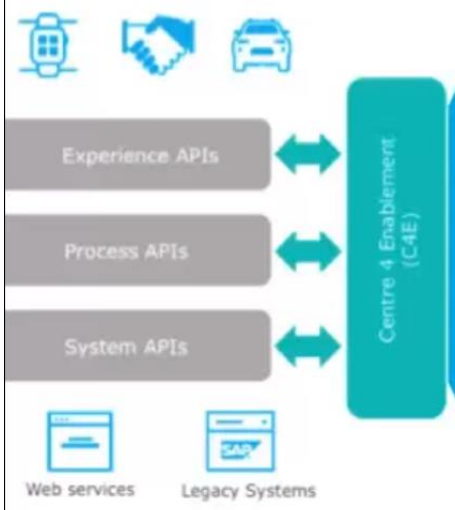


MuleSoft: Technology, Approach & People



The vision cannot be realised without organisational change
This is the Centre 4 Enablement (C4E)

MuleSoft



The C4E is a new operating model for IT

- It enables IT and business units to fulfill connectivity needs through an API led approach
- It is not a project delivery team, nor is it an ivory tower, in the way that many Centre of Excellence (CoE) teams have become
- Its primary purpose is the **enablement** of project delivery teams to leverage the platform and work in a faster, more agile and scalable manner - whilst also driving reuse and mass adoption
- It is an organisational way of working to drive multi-modal / multi-speed integration

Activities with the C4E

MuleSoft

Technology Platform

- Provide the platform to deliver an API led approach, application and license management and central product roadmap influence

Enablement & Evangelisation

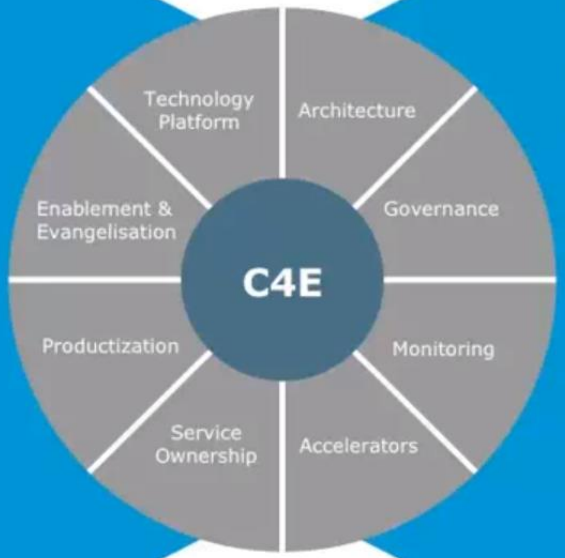
- Persuade and evangelise APIs in the community, coordinate training and certification, thought leadership and feature / function awareness.

Productization

- Best practice 'how to' guides on how to design, document and brand APIs. Manage the central marketplace / store to discover and consume APIs services, repurpose and resupply

Service Ownership

- Provision of the necessary tools, analysis and processes to allow effective ownership and continuous improvement



Architecture

- Thought leadership and standards to achieve API led connectivity within the organisation

Governance

- Automate governance as much possible. Coordinate collaboration with business and IT leadership, information security, risk and audit

Monitoring

- Provide a single view of all APIs, their endpoints, their utilization, their parameters and their adoption across the business

Accelerators

- Centralized provision and ownership of core APIs, templates, connectors and patterns with a high degree of reusability across the business to create synergistic value

What's the difference between a CoE and a C4E?

MuleSoft

	IT as owner (CoE)	IT as enabler (C4E)
Goal	Reliability	Agility
Approach	Waterfall, V-Model	Agile, Kanban, Minimum Viable Product
Governance	Continuous, process-based	Plan-driven, approval-based
Cycle Time	Long (months, years)	Short (days, weeks)
Sourcing	Enterprise suppliers, long term deals	Enterprise and niche, responsive short term deals
Culture	IT-centric, removed from customer	Business-centric, close to customer, fail fast
Talent	Good at traditional projects	Good at new and uncertain projects

Mule Runtime Manager

./amc_setup -H 43affe38-6d24-41da-9aa9-3112ab263336---396207 virtusa-server

Benefits of MuleSoft's Anypoint Platform

Accelerated delivery

Fast to start, faster over time

Automated security

Security and governance by default

Resilient operations

Built-in reliability, scalability, and visibility

Future-proof foundation

Flexible and built for change


MuleSoft
Thank You
All contents © MuleSoft Inc.