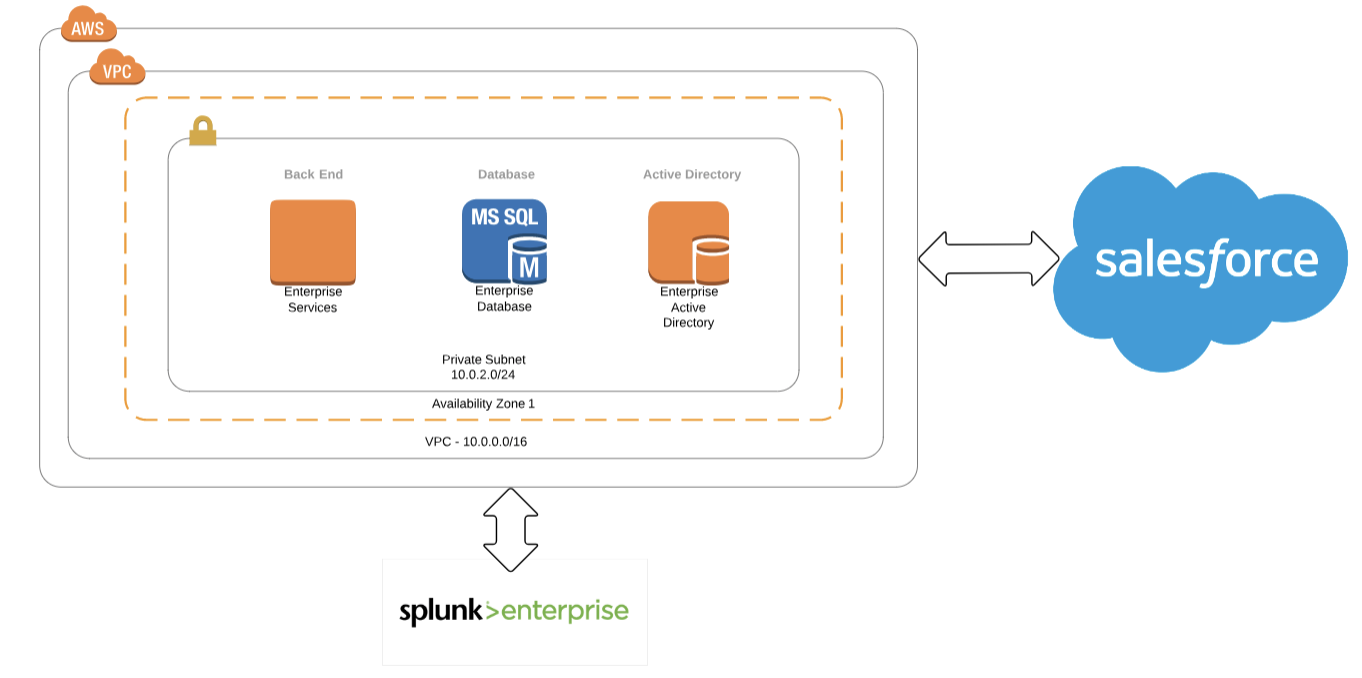
# Acme Insurance Inc Organization System case

Acme Insurance Inc is a well-established, medium-sized, nationwide auto insurance provider. Currently, all customer representatives create **policies** for customers using an **enterprise legacy system**. The legacy system allows the customer support team to create new policies and modify existing policies. However, Acme has recently started using a **CRM** **system** as its **policy management tool**. In the first phase of the project, the CRM system is going to manage only customer data for policies. Currently, Acme wants to maintain **customer information** in **both** the **legacy enterprise system** and the **CRM system**. Therefore, it is important for Acme **to synchronize customer information from the legacy enterprise system to the CRM system, and vice-versa**.

**Note: A key objective of the integration is to avoid point-to-point integration, but instead to design System APIs to these existing systems**

# Physical View of Acme Insurance Inc Organization Systems

This diagram summarizes the current physical view of the enterprise systems involved in the integration project.



# Kicking off the Integration Effort

Acme’s plan is to initialize the integration effort by migrating all existing policies from the legacy system to the CRM system in a one-time only load.

# A Glimpse into Acme Insurance's Baseline Technology Architecture and Functional Requirements

### The Existing Legacy Java-Based Enterprise System

Acme’s legacy enterprise system is a Java-based monolith application. Salesforce is the **CRM system** used for **policy management**. Acme wants to implement this integration solution using MuleSoft’s API-led Connectivity to design new Mule applications to allow the customer support team to create new policies and modify existing policies.

### Planning for New System APIs

Acme is planning to develop a **few System APIs** to access **customer information from Salesforce** and **reuse the existing enterprise legacy system**. Acme will also layer the processing logic into between System API, Process API, and Experience API layers. The enterprise wants to secure system APIs and wants to achieve low latency between the other API layers.

### Current Enterprise Analytics and Monitoring

Currently, **enterprise monitoring and analytics** for all enterprise systems is provided by **Splunk**.

### Transactional Monitoring is Required

It is also required to be able to **trace transactions** across all the integration solution's API layers

### The Existing Legacy Enterprise Database System

Acme has a legacy database system that is used as an **enterprise system** (monolith system). This enterprise system is not scalable, so Acme wants to throttle request to this enterprise system. The enterprise system provides a SOAP service for communication.

The enterprise system (**ES**) is currently secured using AWS security. The Salesforce integration also has to be secured and use integration solution best practices.

### The Existing Legacy Enterprise Database System

The customer policy information contains PII data such as SSN and address. The PII data should be encrypted in transit and also needs to be audited. If data persisted in CloudHub, it must be secure and highly available. The logging and log retention policy for Mule applications should be clearly defined with alerts must be generated when a Mule application fails. Because these Mule applications are integrating critical path use cases, monitoring must also be setup for the Mule applications.

### Current Environments in the Organization

Acme has **multiple environments** such as development, stage, QA, and production.

The **security** of all Mule applications and API implementations is a key concern for stakeholders, and they would like to have a clear demarcation between production and non-production environments. Acme also wants to encrypt all passwords to external systems, use two-way SSL certificates with Salesforce, and host all its APIs within the acme.com domain.

### Strong Security Requirements

Acme has an **identity management system** and wants to authenticate its Anypoint Platform users from this identity management system.

### Parallel Work Streams use Anypoint Platform

Acme has multiple teams working on **Anypoint Platform**, and uses **GIT** as its **source code management system**. **Modularization** of the **Mule application** is highly encouraged at Acme. **Reusability** of **libraries and APIs** and **automation** of **repetitive tasks** are equally important.

Acme encourages team to create **reusable** **artifacts** and **promote** using **MuleSoft Anypoint Exchange** as **central repository** for artifacts.

The Acme **performance team** is planning to conduct performance profiling of Mule application in local and deployment sandbox, and is looking to implement **Continuous Integration and Continuous Delivery** (**CI/CD**) with Mule application configuration management. Configuration management must store and transmit all credentials securely.

**Non-Functional Requirements**

The customer data should be reliably transferred between the enterprise system and the Salesforce CRM system. Acme requires a high availability **service level agreement** (**SLA**) of 99.99%. A load balancing or clustering solution can be considered to achieve high availability goals, but it must be cost effective.

The **response time** for the **API** should **not** be more than **300 milliseconds** and **expected** **throughput** from the API is **20 TPS**.

The communication between the **integration system**s should use **TLS**. It is important to select the right persistence for caches and batch jobs. **Data synchronization** between the **enterprise system** and the **Salesforce CRM** system is planned to maintain the integrity of both systems.

# Case Study Architecture

**Document views of the architecture for all stakeholders**

**Document all decisions made in the process of architecting**

**Document trade-off between meeting non-functional requirements and performance goals**

**Define the combination of deployment options for the Anypoint Platform control plane and runtime plane(s) which best serve this organization at the start of their strategic journey using MuleSoft**

**Define a network architecture that supports these requirement and deployment options**

**Define a CIDR block for an Anypoint Platform provided VPC that meets the organization requirement for a required range of private IP addresses**

**Define role-based access control (RBAC) to control and manage access to Anypoint Platform features**

**Define the most appropriate integration style for an integration solution with idiomatic usage of mule components for integration that meets the organization's current requirements**

**Plan the best way to implement the data transformation logic for this new Mule application while minimizing the overall testing effort**

**Plan the most appropriate way to implement persistence for the watermark in order to support the required data replication integration logic**

**Plan aspects of a CI/CD pipeline for Mule applications that can be automated using MuleSoft-provided Maven plugins**

**Plan an effective way to conduct performance tests of the API implementations within the performance environment**

**Define a logging strategy for Mule applications**

**Propose an architecture, design, runtimes, and OS/JVM/Network/Protocol tuning choices for integration solution**

**Plan an effective way for the project team responsible for the Mule applications and APIs being built to communicate with these stakeholders using Anypoint Platform and its supplied toolset**

**Plan and design a Mule application and related services so the credentials required to access all the related backend systems are managed centrally**

**Plan the type of Anypoint Exchange artifact(s) that should be added to Anypoint Exchange to publish a project skeleton to promote future reuse and parallel development**

**Plan the type of artifact(s) that should be added to the Mule application source files to properly publish the project source code to Anypoint Exchange**

**Plan to manage your libraries and shared resources in a Mule application**

**Define an effective alert and monitoring strategy that avoids infrastructure and application bottlenecks in the production environment**