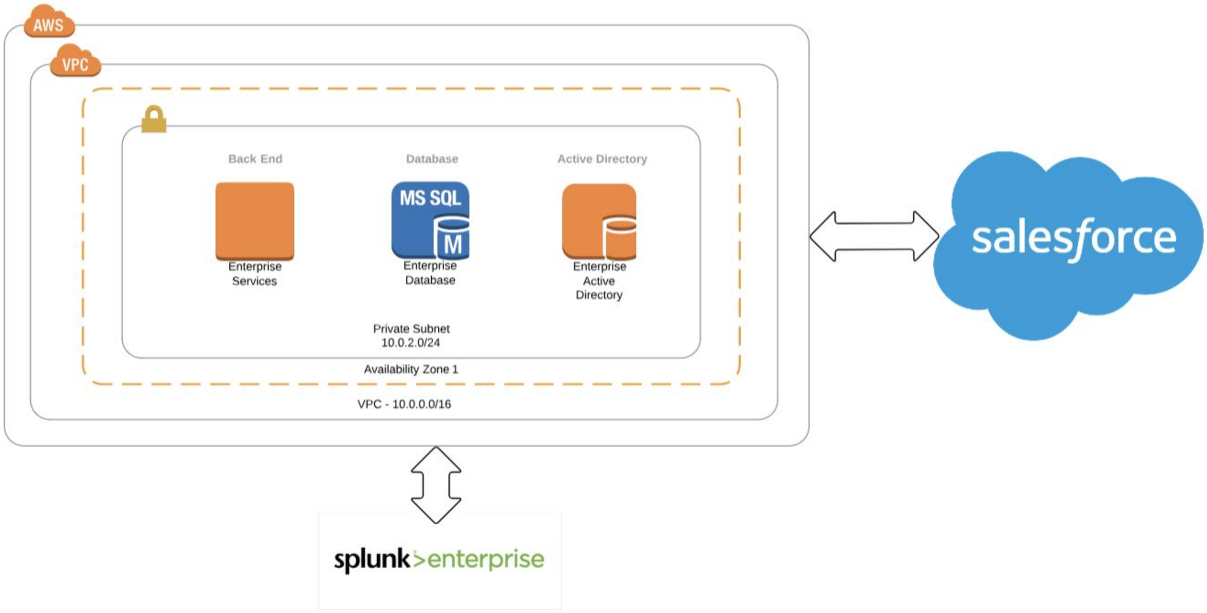
Acme Insurance Inc Organization System

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

Acme's plan is to migrate all existing policies from legacy system to CRM as one-time load.



Note: Designing of System API is key objective of exercise.

A glimpse into Acme Insurance's baseline Technology Architecture and Functional Requirements

Acme's legacy enterprise system is Java based monolith applications. It is using Salesforce as CRM system for policy management. It wants to implement MuleSoft's API Led Connectivity for applications which allow its customer support team to create new policy and modify existing policies.

Acme Inc is planning to develop few System API for customer information on top of Salesforce and reuse existing enterprise legacy system, and layered into System, Process and Experience layers. The enterprise wants to secure system APIs and want to maintain the low latency between the rest of API layers. Existing capability of monitoring and analytics is provided by Splunk in enterprise. Provide traceability of transaction across various integration layer.

Acme uses legacy enterprise database system in its enterprise system (monolith system) and it is not scalable so want to throttle request to enterprise system. The enterprise system uses SOAP for communication.

ES is secured using AWS security, propose right integration solution with ES. Salesforce integration has to be secured and used right integration solution.

The customer policy information contains PII data such as SSN, address. The PII data should be encrypted in transit and audit. If data persisted in CH, it should be secure and highly available. Logging and log retention policy for application should be clearly defined with alerting in case of failures in application. As application is critical to success of Acme, it is advisable to have monitoring is setup for application.

Acme has multiple environments such as development, stage, QA and production. The security of application/APIs is key concern for stakeholder and would like to have clear demarcation between production and non-production environments. It also wants to encrypt all passwords to external systems. The Acme wants to use two-way SSL certificate with Salesforce. Acme wants to host all its APIs under acme.com domain.

Acme has its identity management system and wants to authenticate its Anypoint platform users from its own identity system.

It has multiple teams working on Anypoint platform, it is using GIT as source code management system. The application modularization is highly encouraged in Acme, reusability of libraries and API and automation of repetitive tasks are rewarded equally. The performance team from Acme is planning to perform profiling of application in local and deployment sandbox. It is looking to implement Continuous Integration and Continuous Delivery with application configuration management. Configuration management should kept all credentials secure.

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

Non-Functional Requirements

The customer data should be reliably transferred between enterprise system and Salesforce, Acme is looking for high availability of 99.99%. The load balancing or clustering solution can be considered for high availability and it has to be cost effective.

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

The communication between integration systems should use TLS. It important to select right persistence for cache and batch jobs. Data Sync between ES and SF should be maintained integrity of systems.

Document views of architecture to model your architecture

Document all decisions made in process of architecting

Document trade-off with non-functional requirement can be expected when meeting goals of performance.

Define combination of deployment options of the Anypoint Platform control plane and runtime plane(s) best serves this organization at the start of this strategic journey

Define network architecture supports this requirement

Define a CIDR block for this VPC that meet organization requirement for private IP address range

Define role-based access control (RBAC) to features of the platform

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 can be automated using MuleSoft-provided Maven plugins

Plan an effective way to approach to conduct performance tests of the API implementation in the performance environment

Define logging strategy for Mule applications

Propose architecture, design, runtime and OS/JVM/Network/Protocol tuning for 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 Mule application so the credentials required to access the backend systems are managed centrally

Plan type of Anypoint Exchange artifact(s) should be added to Anypoint Exchange to publish the project skeleton

Plan type of artifact(s) should be added to source to publish the project source code

Plan to manage your libraries, shared resources in mule application

Define your alert and monitoring strategy to avoid infrastructure and application bottleneck in production environment