**White Paper: API and Integration Architecture**

Designing Scalable, Standards-Based Integration for Modern Systems (VA Context)

White Paper: API and Integration Architecture – Designing Scalable, Standards-Based Integration for Modern Systems (VA Context)

## Executive Summary

The Department of Veterans Affairs (VA) relies on API-first integration to enable real-time services, streamline claims processing, and maintain system interoperability across platforms like Salesforce, AWS, BGS, MPI, and VA Profile. Secure, scalable APIs allow for standardized data sharing while reducing system coupling. This paper explains how Salesforce APIs, API gateways, developer portals, middleware platforms, and compliance frameworks like MISMO enable VA to implement robust integration.

## 1. Salesforce APIs Overview

Salesforce provides multiple APIs to meet different integration needs.

## 1.1 REST API

- Use Case: Real-time data operations such as reading or updating Veteran records.

- Features:

- Lightweight HTTP-based protocol.

- Supports OAuth 2.0 for secure, token-based authentication.

- Uses JSON for simple request/response interactions.

- VA Usage:

- AWS Lambda functions call Salesforce REST APIs to sync claim data.

- VA.gov captures user-submitted data and updates Salesforce records via REST endpoints.

- Postman or Swagger can be used to test these endpoints during development.

## 1.2 Bulk API (V1 & V2)

- Use Case: Mass uploading or syncing large datasets.

- Features:

- Supports asynchronous data operations to handle tens of thousands of records.

- REST-based interface (especially in v2), improves session handling.

- VA Usage:

- Batch loading records from legacy systems (e.g., BGS) into Salesforce.

- ETL pipelines using AWS Glue or Informatica rely on Bulk API for high-volume imports.

## 1.3 Metadata API

- Use Case: Deployment and version control of Salesforce configurations.

- Features:

- Retrieve or deploy metadata such as custom fields, Apex code, and page layouts.

- Compatible with SFDX DevOps practices.

- VA Usage:

- Automating migration between development, staging, and production orgs.

- Enabling rollback via metadata backups during deployment failures.

2. API Management Strategy

## 2.1 API Gateway

- Purpose: Central control point for routing, security, and traffic management.

- Key Features:

- Security: OAuth2 authentication, JWT validation, WAF protections, IP whitelisting.

- Rate Limiting: Ensures systems don't get overwhelmed (e.g., throttling requests from external vendors).

- Monitoring: Integrated with tools like Splunk, AWS CloudWatch, Datadog to track API performance.

- Routing: Directs requests to services like Salesforce, MPI, or VA.gov APIs.

- Gateways in Use:

- AWS API Gateway: Natively integrates with Lambda and IAM for fine-grained control.

- Apigee: Provides advanced analytics and access controls.

- Kong/Tyk: Open-source gateways used for internal prototyping or lower-cost environments.

## 2.2 Developer Portals

- Purpose: Provide secure access for internal/external developers to discover and use APIs.

- Features:

- OpenAPI (Swagger) documentation for onboarding.

- API key provisioning and usage analytics.

- Self-service testing and mock API environments.

- VA Usage:

- VA Lighthouse portal for third-party developers.

- Internal portals for onboarding APIs via Mulesoft or Apigee.

3. Integration Patterns and Compliance

## 3.1 Event-Driven Architecture

- Concept: Systems interact via events, not direct calls, reducing coupling.

- Benefits:

- Improves scalability (systems only process relevant events).

- Increases fault tolerance (retry mechanisms).

- VA Usage:

- Salesforce emits Platform Events when data is changed.

- AWS EventBridge routes these to Lambda, which updates VA Profile or stores in S3/Kafka.

- Example: Status change on a Veteran claim triggers downstream workflows.

## 3.2 MISMO Compliance

- Purpose: Ensures mortgage data formats are compatible across banks, underwriters, and GSEs.

- VA Usage:

- Standardizes Loan Guaranty workflows to interoperate with external financial institutions.

- Approach:

- Middleware (Informatica, Mulesoft) transforms Salesforce objects to MISMO-compliant XML.

- Canonical data models reused across integration to align with NIEM or HL7 as needed.

- Tools:

- XMLSpy or Altova MapForce to validate schema and transform data.

## 4. Tools and Platforms

| Category | Tools |

|---------------|--------------------------------------------------|

| API Gateway | AWS API Gateway, Apigee, Kong |

| Eventing | Salesforce Platform Events, AWS EventBridge |

| Middleware | Mulesoft, Boomi, Informatica |

| API Design | Swagger, Postman, Redoc |

| Compliance | XMLSpy, Altova MapForce (MISMO transformation) |

5. Architecture Example: Mortgage Loan Integration

## Scenario: Loan data flows from Salesforce to a Loan Origination System (LOS).

1. A Salesforce Flow captures new loan data.

2. A Platform Event is triggered on update.

3. AWS EventBridge picks up the event and routes to a Lambda function.

4. Lambda transforms the data into MISMO XML format.

5. XML is sent to the external LOS over a secure REST endpoint.

6. The LOS returns a response (e.g., approved or rejected), which is written back to Salesforce.

## Security Controls:

- OAuth2 with JWT for secure token access.

- KMS used to encrypt sensitive data before transmission.

- API Gateway enforces throttling and WAF to protect entry points.

- Splunk and CloudWatch capture the audit trail.

## 6. Resources and References

| Resource | Description |

|----------------------------|------------------------------------------------|

| Salesforce Developer Docs | REST, Bulk, Metadata API references |

| MuleSoft API Design Guide | API-led connectivity patterns |

| MISMO Standards | Mortgage data schemas and documentation |

| OpenAPI | Swagger specifications for REST API docs |

| Salesforce Events | Guide to Platform Events and streaming APIs |

## Conclusion

The VA's integration success depends on robust, reusable, and secure APIs. By leveraging API gateways, event-driven flows, and standards like MISMO, the VA ensures data consistency, developer productivity, and Veteran-focused service delivery.

## Next Steps:

- Establish a Center of Excellence (CoE) for API onboarding and lifecycle governance.

- Create standardized API templates using OpenAPI.

- Expand reusable ETL/transform libraries in Mulesoft and Informatica.

- Extend event-based models to additional domains like notifications and analytics.