



Salesforce Architecture Sample Pack

Prepared by: Niyati Vyas

Role: Senior Salesforce Consultant | Cloudsage Consultancy Inc.

1. Overview

This pack includes sanitized, reusable architecture samples and process documentation related to Salesforce package development, data synchronization, and deployment planning—prepared for demonstration purposes.

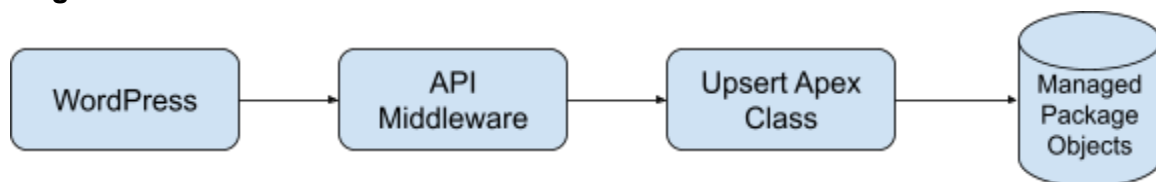
2. Sample Architecture: WordPress → Salesforce Data Sync

Use Case: Syncing external funding opportunities from a WordPress site to a managed package in Salesforce.

Flow Summary:

- WordPress endpoint publishes JSON feed daily.
- Middleware service (e.g., Azure Function or AWS Lambda) parses and transforms data.
- Data is upserted into Salesforce custom objects inside the managed package using REST API.
- Records are matched via external ID and timestamp.
- Apex triggers ensure data integrity and bulk-safe processing.

Diagram:



Key Considerations:

- Bulk-safe Apex with asynchronous handling.
- External ID usage for idempotency.
- Retry and logging logic.

- Secure endpoint and IP restrictions.
-

3. Sample Apex Class Snippet (Sanitized)

```
public with sharing class FundingOpportunitySync {
    public static void upsertOpportunities(List<FundingWrapper> records) {
        List<Opportunity> oppsToUpsert = new List<Opportunity>();
        for (FundingWrapper fw : records) {
            Opportunity o = new Opportunity(
                External_Id__c = fw.externalId,
                Name = fw.name,
                Amount = fw.amount,
                CloseDate = fw.deadline,
                StageName = 'Prospecting'
            );
            oppsToUpsert.add(o);
        }
        upsert oppsToUpsert External_Id__c;
    }
}
```

4. CI/CD and Packaging Strategy Outline

- **Tooling:** Salesforce CLI, GitHub Actions, OrgShape
 - **Branching Strategy:** Dev → QA → UAT → Production
 - **Release Packaging:**
 - Convert from 1GP to 2GP unlocked/managed package
 - Use `sfdx force:package` to create versioned releases
 - Automate deployments with changelog annotation
-

5. Versioning & Security Review Checklist (Excerpt)

Security Controls:

- CRUD/FLS checks in Apex
- Enforced sharing rules
- Use of Platform Events for async operations
- No hard-coded credentials

Versioning Playbook:

- Increment minor/patch versions per feature or fix
 - Maintain metadata compatibility
 - Automate regression testing with Apex and Postman
-

6. Documentation Template Sample

Release Notes Template:

- Version:
- Release Date:
- New Features:
- Bug Fixes:
- Dependencies:
- Manual Steps (if any):

Admin Guide Outline:

- Setup Steps
- External Feed Configuration
- Sync Schedule
- Logging & Monitoring