Phase 7: Integration & External Access

Farmers Market Salesforce Project (Deep-Dive Guide)

Objective

Enable the Farmers Market CRM to connect securely with external apps/services, move data reliably, and expose selected capabilities to partners. This guide gives **click-by-click steps, reference architectures, Apex/Flow patterns, tests, deployment notes, and troubleshooting**.

1) Named Credentials (NC) & External Credentials

What: Securely store endpoints + auth and call them as `callout:My_NC`.

A. Simple (no OAuth) – Step-by-Step

1. **Setup → Named Credentials → New** (or **External Credentials** first in newer UI).
2. **URL**: `https://api.example.com`
3. **Identity Type**: Named Principal (shared) or Per-User.
4. **Authentication**:
    - **Password Authentication** → username/password
    - **Custom Header** → API key
5. **Generate Authorization Header**: Off (if you pass headers yourself).
6. **Save**.

Use in Apex:

```
HttpRequest req = new HttpRequest();
req.setEndpoint('callout:My_NC/v1/shipping/quote');
req.setMethod('POST');
req.setHeader('Content-Type', 'application/json');
req.setBody(JSON.serialize(new
Map<String,Object>{'weight'=>12,'pin'=>'500001'}));
HttpResponse res = new Http().send(req);
```

B. OAuth 2.0 – Step-by-Step

1. **Setup → Auth. Providers → New** → choose e.g., **OpenID Connect** or **Generic OAuth 2.0**.
    - Fill **Authorize URL**, **Token URL**, **Client Id/Secret**, **Scopes**.

2. **External Credential** → Add **Principal** using the Auth Provider (incl. scope mapping).

3. **Named Credential** → **Authentication Protocol: OAuth 2.0** → link the **External Credential** + **Scope**.

4. **Initiate**: Click **Authenticate** (admin) or let per-user auth run.

Notes - Prefer NC over legacy **Remote Site Settings** (RSS). RSS still required when **not** using NC. - Secrets belong in NC/External Credential, not in Apex.

2) External Services (Schema-Driven APIs in Flow)

What: Import an **OpenAPI (Swagger)** spec to auto-create **invocable Flow actions**.

Steps

1. Create **Named Credential** for the API host.

2. **Setup → External Services → Register**.

3. Provide **Schema** (JSON or URL) + choose the **Named Credential**.

4. Salesforce generates **Actions** usable in **Flow**.

Minimal OpenAPI Fragment (example)

```
{
  "openapi": "3.0.1",
  "info": {"title": "Logistics API", "version": "1.0"},
  "servers": [{"url": "https://api.logistics.example"}],
  "paths": {
    "/v1/shipments": {
      "post": {
        "operationId": "createShipment",
        "requestBody": {"content": {"application/json": {"schema": {"$ref":
"#/components/schemas/Shipment"}}}},
        "responses": {"201": {"description": "Created"}}
      }
    }
  },
```

```
  "components": {
    "schemas": {"Shipment": {"type": "object", "properties": {"orderId":
{"type": "string"}, "pin": {"type": "string"}}}}
  }
}
```

In Flow: Add Action **Logistics.createShipment**, map **Order__c.Id** → `orderId`, Buyer PIN → `pin`.

3) Inbound Web Services (Expose Salesforce)

Use when partners need to push data into Salesforce.

A. Apex REST Resource (example: create an Order)

```
@RestResource(urlMapping='/v1/orders')
global with sharing class OrdersApi {
  @HttpPost
  global static Id createOrder() {
    RestRequest req = RestContext.request;
    Map<String, Object> body = (Map<String, Object>)
JSON.deserializeUntyped(req.getBody());
    Order__c o = new Order__c(
      Name = (String)body.get('orderName'),
      Buyer__c = (Id)body.get('buyerId'),
      Produce__c = (Id)body.get('produceId'),
      Order_Quantity__c = (Integer)body.get('qty'),
      Status__c = 'Pending'
    );
    insert o;
    return o.Id;
  }
}
```

Security - Create a **Connected App** (OAuth) → issue tokens to partners. - Scope to least privilege (refresh_token, api).
- Use **profiles/perm sets** for field/object access; avoid `without sharing` unless justified. - Add **CORS** entry if using browser apps.

4) Outbound Callouts (Apex & Flow Patterns)

A. Synchronous callout (simple)

```
public with sharing class FreightService {
  public static Decimal quote(Integer weight, String pin) {
    HttpRequest r = new HttpRequest();
    r.setEndpoint('callout:Freight_NC/v1/quote');
```

```
    r.setMethod('POST');
    r.setHeader('Content-Type','application/json');
    r.setBody(JSON.serialize(new
Map<String,Object>{'w'=>weight,'pin'=>pin}));
    HttpResponse h = new Http().send(r);
    if (h.getStatusCode() >= 200 && h.getStatusCode() < 300) {
      Map<String,Object> resp = (Map<String,Object>)
JSON.deserializeUntyped(h.getBody());
      return (Decimal) resp.get('amount');
    }
    throw new CalloutException('Freight quote failed: ' + h.getStatus());
  }
}
```

B. Queueable for resilience

```
public class ShipOrderQueueable implements Queueable, Database.AllowsCallouts
{
  Id orderId;
  public ShipOrderQueueable(Id orderId){ this.orderId = orderId; }
  public void execute(QueueableContext qc){
    // build payload from Order__c and call Logistics API via Named
Credential
  }
}
```

Trigger enqueue example:

```
trigger OrderAfterUpdate on Order__c (after update){
  for (Order__c o : Trigger.new){
    if (o.Status__c == 'Shipped' && Trigger.oldMap.get(o.Id).Status__c !=
'Shipped'){
      System.enqueueJob(new ShipOrderQueueable(o.Id));
    }
  }
}
```

C. Testing callouts (required)

```
@IsTest
private class FreightServiceTest implements HttpCalloutMock {
  public HTTPResponse respond(HTTPRequest req){
    HttpResponse res = new HttpResponse();
    res.setStatusCode(200);
    res.setBody('{"amount": 123.45}');
    return res;
  }
  @IsTest static void testQuote(){
    Test.setMock(HttpCalloutMock.class, new FreightServiceTest());
    Decimal amt = FreightService.quote(10, '500001');
    System.assertEquals(123.45, amt);
```

```
  }
}
```

Best Practices - Timeouts (e.g., 10s), exponential backoff on transient errors, idempotency keys for retries. - Log requestId/endpoint/status to a custom **Integration_Log__c**.

5) Platform Events (PE) – Event-Driven Updates

Use when external systems only need notifications, not full data syncs.

Steps

1. **Setup → Platform Events → New**: `Order_Confirmed__e` with fields:
    - `OrderId__c (Text 18)`, `BuyerName__c (Text)`, `Total__c (Currency)`, `When__c (DateTime)`.
2. **Publish** from Flow (Record-Triggered on Order__c when Status__c = "Confirmed"). Or publish via Apex:

```
Order_Confirmed__e evt = new Order_Confirmed__e(
  OrderId__c = o.Id,
  BuyerName__c = o.Buyer__r.Name,
  Total__c = o.Order_Total__c,
  When__c = System.now()
);
Database.SaveResult sr = EventBus.publish(evt);
```

3. **Subscribe**
    - Internal: **Trigger** on PE or **Flow** (Platform Event–Triggered Flow).
    - External: via **CometD/Bayeux** (EMP API) on `/event/Order_Confirmed__e`.

6) Change Data Capture (CDC)

Use when downstream systems need **all data changes** (create/update/delete/undelete) for objects like **Order__c**.

Steps

1. **Setup → Change Data Capture** → enable **Order__c, Produce__c, Buyer__c**.
2. External subscribers use channel: `/data/Order__ChangeEvent`.
3. Payload includes changed fields, replayId, commitTimestamp, Tx boundaries.

Tips - Use **replay** IDs to recover missed events.
- Avoid sensitive fields unless required; consider field-level encryption.

7) Salesforce Connect (Virtualized External Data)

Goal: View external inventory (e.g., warehouse DB) **without importing**.

Steps

1. **Setup → External Data Sources → New**: Type **OData 4.0** (or 2.0/Custom).
2. Enter Endpoint + **Named Credential** (OAuth supported).
3. **Validate & Sync** → choose external tables to create **External Objects** (e.g., `Inventory__x`).
4. Create **Indirect Lookup** from `Produce__c` to `Inventory__x` using a common SKU/Code.

Notes - Read-only by default; upsert only if provider supports it.
- External objects use _x suffix.

8) OAuth & Connected Apps (Inbound Auth)

Steps to create a Connected App

1. **Setup → App Manager → New Connected App**.
2. Enable **OAuth**; set **Callback URL** (partner's URL for web server flow, or `sfdc://success` for testing).
3. Select **Scopes**: `api`, `refresh_token`, `openid` as needed.
4. Share **Consumer Key/Secret** securely.
5. Assign **Policies**: IP relaxation (as per security), session timeout, JWT bearer if using certificates.

JWT Bearer Flow - Exchange signed JWT for access token: no user interaction; good for server-to-server.

9) API Limits & Governance

- **Daily API calls** per org (varies by edition/licenses).
- **Concurrent long-running callouts**; **heap size**; **max 100 callouts per tx**.
- Use **Bulk API** for large data moves; **Composite** for chattiness reduction.
- Cache reference data (Custom Metadata/Cache) to cut calls.

10) Remote Site Settings (RSS)

- Needed when doing callouts **without** Named Credentials.
- **Setup → Security → Remote Site Settings → New** → add `https://api.example.com`.

Prefer **Named Credentials**; they handle auth + eliminate many RSS needs.

11) Testing Strategy

Unit Tests - Mock every callout with **HttpCalloutMock**.
- 75%+ coverage including error paths.

Integration Tests - Use **Postman** to test Apex REST endpoints (OAuth token).
- Validate Platform Events & CDC with **EMP Connector** (replayId handling).

UAT Checklist - Positive/negative test cases for each integration.
- Retry/backoff verified.
- Permissions/FLS enforced.

12) Deployment & Release

- Use **Change Sets** or **SFDX** for: Apex classes, Platform Events, Flows, External Services registrations, Auth Providers metadata.
- **Secrets (client secrets, passwords)** are **not** moved by metadata—recreate in target org.
- Post-deploy: re-authenticate Named Credentials; validate endpoints; rotate keys if required.