**Phase 7: Integration & External Access**

**Named Credentials**

* Store authentication details for external APIs (URL, token, key).
* Removes need for hardcoding credentials in Apex.
* **Example:** Store credentials for Razorpay/Stripe API → fetch donation payment status securely.

**External Services**

* Connect declaratively with APIs using OpenAPI/Swagger schema.
* Generates invocable Apex actions usable in Flows.
* **Example:** Connect to Payment Gateway API → donation verification action inside Flow.

**Web Services (REST/SOAP)**

* **REST API:** Lightweight, JSON-based, most common.
* **SOAP API:** XML-based, less common for NGOs.
* **Example:** Create REST endpoint in Salesforce → external NGO partner can fetch campaign donation summary.
* **Example:** Donor portal calls Salesforce REST API to retrieve donor profile + loyalty points.

**Callouts**

* Outbound calls from Salesforce to external APIs.
* Requires **Remote Site Settings** to whitelist endpoints.
* **Example:** Call currency conversion API → convert foreign donations (USD/EUR) to INR before saving record.

**Platform Events**

* Event-driven communication inside Salesforce.
* **Example:** Large donation received (> ₹1,00,000) → publish event → NGO Manager notified instantly.

**Change Data Capture (CDC)**

* Real-time tracking of changes to Salesforce records.
* **Example:** Whenever a Donation record is updated → event fired → CSR Partner Dashboard auto-refreshes.

**Salesforce Connect**

* Access external system data without importing it.
* Creates External Objects linked to outside databases/APIs.
* **Example:** CSR corporate donation pledges stored in external DB → visible live in Salesforce campaigns.

**API Limits**

* Salesforce restricts daily API calls (e.g., 15,000/day in Developer Org).
* Important for donation-heavy campaigns.
* **Example:** Use batch callouts to payment gateway instead of one call per donation → saves API calls.

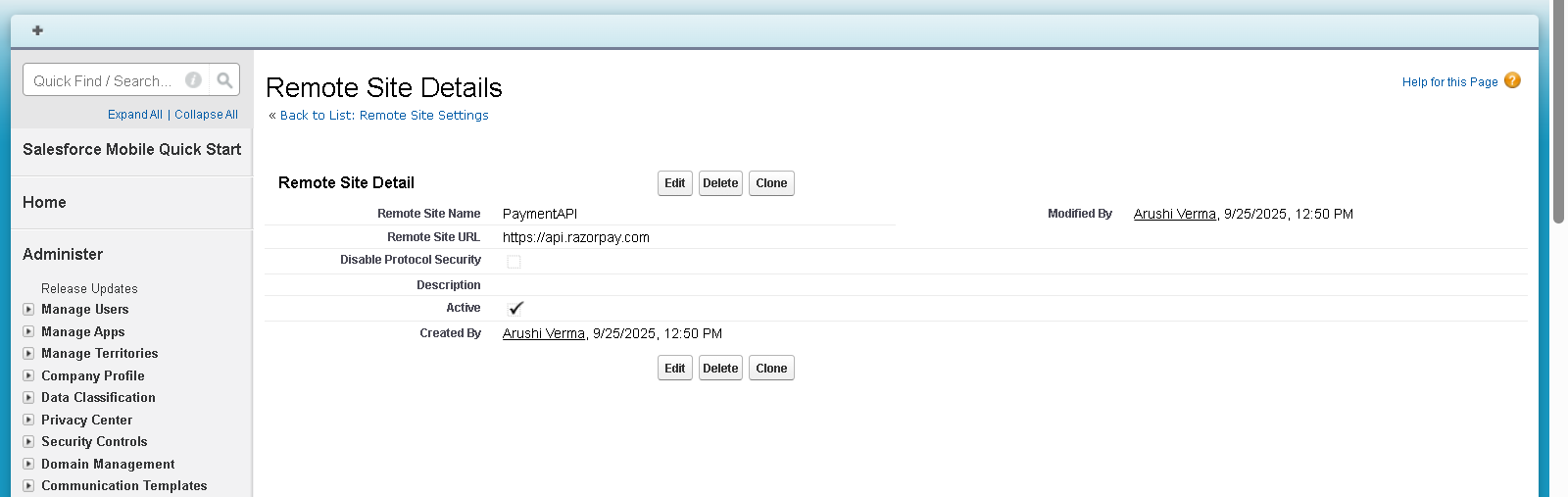
**OAuth & Authentication**

* Secure way for external apps to access Salesforce.
* Uses **OAuth 2.0 token exchange**.
* **Example:** Donor Portal authenticates via OAuth → ensures only registered donors see their dashboards.

**Remote Site Settings**

* Required before making Apex callouts.
* Whitelists domains for security.
* **Example:** Add https://api.razorpay.com before calling payment API.

**Screenshots:**

**Remote Site Details**

**Named credential: crs** 