Functional Requirements: MDM Request Hub (Detailed)

1. 🚀 Project Objective

This project aims to design and deploy a robust, centralized **Master Data Management (MDM) Request Hub** application utilizing the SAP Cloud Application Programming (CAP) model. The primary purpose of this hub is to serve as the authoritative, single source of truth and the unified user interface for managing the complete lifecycle of **Business Partner (BP)** master data. This encompasses both the **creation of new BPs** and the **modification of existing BPs** originating from diverse upstream systems.

The MDM Request Hub acts as a critical governance layer, ensuring data consistency, quality, and compliance before synchronization with the core ERP system (S/4HANA). It bridges the gap between operational satellite systems—specifically **Purchasing Interface (PI)**, **Coupa**, and **Salesforce**—and the central SAP S/4HANA backend. By consolidating requests into a single view, the application streamlines the workflow for the MDM team, reduces data redundancy, and enforces standardized data entry rules across the enterprise.

1.1 Scope and Integrated Systems

The application will integrate requests from three distinct source systems, each with unique data requirements and user workflows:

1. **Purchasing Interface (PI):** primarily used for supplier onboarding and management.
2. **Coupa:** An automated procurement platform that will programmatically submit supplier requests.
3. **Salesforce:** A CRM platform managing customer data, which utilizes a complex **Main Account (Business Partner)** and **Sub-Account (Contract Account)** data model.

The solution must provide a seamless user experience for different user roles while strictly adhering to the data governance policies defined in the "B2B Customer Master Data Approach" and "MDM Solution Blueprint".

2. 👥 Personas and User Roles

The application is designed to serve three distinct human personas, each with specific responsibilities and access rights. Understanding these personas is crucial for implementing the correct security roles and UI views.

2.1 Requesting Satellite Owner (Persona 1)

- **Description:** This user represents the business operational side. They initiate the need for a new or updated Business Partner record. They operate primarily within their domain (e.g., a Sales Manager in Salesforce or a Procurement Officer using PI).
- **Responsibilities:**
  - Initiating requests for new Business Partners.
  - Searching for existing Business Partners to verify before creating duplicates.
  - Submitting change requests to update outdated information (e.g., address changes, new bank details).
  - Tracking the status of their submitted requests.
- **Key Interaction:** They require a simplified, system-specific form that only exposes the fields relevant to their context (e.g., a Salesforce user should not see Coupa-specific procurement fields).

2.2 MDM Team (Persona 2)

- **Description:** The central data governance team responsible for maintaining master data quality across the enterprise. They are the "Super Users" of the application.
- **Responsibilities:**
  - Reviewing all incoming requests (`NEW` status) from all satellite systems.
  - Validating data against corporate standards (e.g., correct VAT formats, valid addresses).
  - Approving valid requests to trigger integration with S/4HANA.
  - Rejecting incomplete or incorrect requests with mandatory feedback.
  - Handling integration errors (e.g., if S/4HANA creation fails).

- **Key Interaction:** They need a comprehensive "Inbox" view with advanced filtering, a detailed view of all data fields (regardless of source), and the ability to see "Change Logs" (diff views) for update requests.

2.3 Acknowledging Satellite Owner (Persona 3)

- **Description:** Business owners of *other* satellite systems who utilize shared master data. For example, if a BP is used by both Salesforce and Coupa, the Coupa admin is the "Acknowledging Owner" when the Salesforce team changes that BP's data.
- **Responsibilities:**
  - Receiving notifications when shared data is changed by another department.
  - Reviewing the changes to ensure they do not negatively impact their own system's operations.
  - Acknowledging the changes to confirm awareness.
- **Key Interaction:** A read-only view of the Business Partner that highlights exactly what changed (e.g., "Old Phone: 123" vs. "New Phone: 456"), with a simple "Acknowledge" action.

3. 🖥️ Core Application Features (Fiori UI)

The application will be built as a single SAP CAP project serving a Fiori Elements-based UI. To accommodate the different personas, the UI will be divided into distinct "views" or "apps" that can be assigned via SAP BTP Role Collections.

A. Requestor View (Target: Persona 1)

This view is optimized for data entry and self-service tracking.

- **Dynamic UI Adaptation:** The application logic must identify the logged-in user's assigned system (e.g., via user attributes or role scope 'Salesforce').

  - *Behavior:* If a user is identified as 'Salesforce', the UI must *hide* the 'Supplier' specific sections (like Purchasing Org data) and *show* the 'Customer' specific sections (like Sales Area data).
- **Request Dashboard:** A table listing only the requests submitted by the current user.
  - *Columns:* Request ID, BP Name, Request Type (Create/Change), Overall Status, Submission Date.
- **"Create New BP Request" Action:** Launches a creation wizard or form.
  - *Action Details:*
    1. User clicks "Create".
    2. System determines `Source System` from user context.
    3. Form opens with fields filtered for that source system.
    4. **Real-time Validation:** On field exit, system checks format (e.g., Email regex, Phone number format).
    5. **Submit:** On click, system validates all mandatory fields. If valid, creates `MDMRequest` with `OverallStatus='NEW'`. If invalid, highlights errors.
- **"Change Existing BP Request" Action:**
  - *Search:* A search bar allowing the user to find a BP by Name, legacy ID, or SAP BP ID.
  - *Pre-fill:* Selecting a result loads the current master data into the form.
  - *Edit:* User modifies specific fields. System tracks "Old Value" vs "New Value".
  - *Submit:* System generates the `changePayload` JSON diff and saves the request.

B. MDM Approval View (Target: Persona 2)

This view is the central command center for data governance.

- **Central Dashboard (Inbox):** A table displaying *all* active requests from *all* sources.

    - *Default Filter:* Shows requests where `OverallStatus` is 'NEW' or 'ERROR'.
    - *Columns:* Source System, Request Type, Overall Status, Requestor Name, Business Partner Name, Creation Date.
    - *Filters:* Faceted filters for Source System, Request Type, and Status.
- **Detailed Request Page:**
    - **Header:** High-level info (Request ID, Source, Status).
    - **Data Sections:** Tabbed or sectioned layout displaying the payload.
    - **"Change Log" / Diff View (CRITICAL):**
        - *Logic:* If `Request Type = 'Change'`, this section renders a comparison view.
        - *Display:* A table showing only the fields that were modified. Columns: "Field Name", "Old Value", "New Value".
        - *Visuals:* Use color coding (e.g., red for old, green for new) to highlight changes instantly.
    - **Integration Status Monitor:** A read-only section displaying the technical status of the backend steps:
        - `sapStatus`: Status of the S/4HANA write.
        - `satelliteStatus`: Status of the write back to the originating system.
- **Decision Actions:**
    - **"Submit for Integration" (Approve):**
        1. User clicks button.
        2. System performs final validation (e.g., duplicate check against S/4).
        3. System updates `OverallStatus` to `PENDING_INTEGRATION`.
        4. System emits `mdm/request/submit` event to Event Mesh.
        5. UI displays success toast message.

  - **"Reject":**
    1. User clicks button.
    2. System opens a dialog requiring a "Rejection Reason" (Mandatory).
    3. User enters reason and confirms.
    4. System updates `OverallStatus` to `REJECTED`.
    5. System triggers notification to Requestor.
  - **"Mark as Completed":**
    1. Button is enabled *only* if `sapStatus`='Success' AND `satelliteStatus`='Success'.
    2. User clicks button.
    3. System updates `OverallStatus` to `COMPLETED`.

C. Change Acknowledgment View (Target: Persona 3)

A streamlined, read-only interface for cross-system communication.

- **Pending Tasks List:** A simple list of "Change Notifications" targeting the user's system.
- **Review Page:**
  - *Context:* Displays the Business Partner's name and ID.
  - *The Change:* Embeds the exact same **"Change Log" / Diff View** component used in the MDM view, ensuring the acknowledging owner sees exactly what was altered.
- **Action:**
  - **"Acknowledge":**

1. User clicks button.
2. System records user ID and timestamp in `satelliteAcknowledgments`.
3. System updates row status to 'Acknowledged'.
4. Item is removed from the "Pending" list.

4. 📋 Data Field Requirements and Validation (SAP Standard Alignment)

The data model uses SAP S/4HANA standard field names (BusinessPartner A2X) where applicable to ensure easy integration.

A. Common Request Fields (System-Agnostic)

These fields manage the request lifecycle and metadata.

Field Name	Data Type	System	Validation / Logic	Context
Source System	`String(20)`	All	**Required**. Value must be one of: `PI`, `COUPA`, `SALESFORCE`.	Determines which UI sections are visible.
Request Type	`String(10)`	All	**Required**. Values: `Create`, `Change`.	Determined by user action (New vs. Search & Edit).
Overall Status	`String(30)`	All	Enum: `NEW`, `PENDING_INTEGRATION`, `IN_PROCESS`, `PENDING_ACKNOWLEDGMENT`, `COMPLETED`, `REJECTED`, `ERROR`.	The primary state machine variable.
Requestor Name	`String(100)`	All	Read-only.	Captures the logged-in user's name.

MDM Notes	`String(1024)`	All	Optional.	Internal notes for the MDM team.
changePayload	`LargeString`	All	JSON Object.	Stores the "Diff" (Old/New values) for Change requests.
sapBusinessPartnerID	`String(10)`	All	Read-only. Maps to `BusinessPartner`.	populated by the Integration Suite callback after S/4 creation.
satelliteSystemID	`String(50)`	All	Read-only.	Populated by Integration Suite; ID from the source system.
sapStatus	`String(10)`	All	Enum: `Pending`, `Success`, `Error`.	Tracks S/4HANA sync status.
satelliteStatus	`String(10)`	All	Enum: `Pending`, `Success`, `Error`.	Tracks source system sync status.
associatedSystems	`LargeString`	All	JSON Array.	List of all systems linked to this BP (e.g., `['Salesforce', 'Coupa']`).

B. PI & Coupa Specific Fields

These fields map to the `A_Supplier` and `A_BusinessPartnerAddress` OData entities.

Field Name (UI Label)	SAP Tech Name (OData)	Validation	Context

Supplier Name	`OrganizationBPName1`	**Required**. Max 80 chars.	Legal entity name.
VAT ID	`TaxNumber1`	**Required**. Format check (VIES).	Tax identification.
PO Currency	`PurchaseOrderCurrency`	**Required**. ISO 4217 (e.g., 'EUR').	Default currency.
Payment Terms	`PaymentTerms`	**Required**. Valid S/4 Key (e.g., 'Z030').	Supplier payment terms.
Payment Method	`PaymentMethodsList`	**Required**. Valid S/4 Key (e.g., 'T').	Payment method list.
Street	`StreetName`	**Required**. Max 60 chars.	Address: Street.
House Number	`HouseNumber`	**Required**. Max 10 chars.	Address: House Number.
City	`CityName`	**Required**. Max 40 chars.	Address: City.
Postal Code	`PostalCode`	**Required**. Max 10 chars.	Address: ZIP/Postal Code.
Country	`Country`	**Required**. ISO 2-char (e.g., 'DE').	Address: Country Code.
Bank Details	`A_BusinessPartnerBank`	**Required** (1..n).	Composition.
- Bank Country	`BankCountryKey`	**Required**. ISO 2-char.	Bank Country.
- Bank Account	`BankAccount`	**Required**.	Account Number.

- IBAN	`IBAN`	**Required**. Checksum validation.	Int. Bank Account Number.

C. Salesforce Specific Fields

These fields support the Customer (`A_Customer`) model and the custom Contract Account logic.

C.1 Main Account (Business Partner)

Field Name (UI Label)	SAP Tech Name (OData)	Validation	Context
Customer Name	`OrganizationBPName1`	**Required**. Max 80 chars.	Legal entity name.
Established VAT ID	`TaxNumber1`	**Required**.	Primary VAT.
BP Type	`BusinessPartnerType`	Optional. Valid Key (e.g., '0001').	Grouping (e.g., Influencer).
Addresses	`A_BusinessPartnerAddress`	**Required** (1..n).	List of addresses.
Bank Info	`A_BusinessPartnerBank`	**Required** (1..n).	Pool of bank accounts.

C.2 Sub-Account (Contract Account)

Custom entity structure for Salesforce Sub-Accounts. | Field Name (UI Label) | SAP Tech Name (Custom) | Validation | Context | | :--- | :--- | :--- | :--- | | **Salesforce Sub-ID** | `YY1_SF_SubID` | Optional. | "NEW" for new records. | | **Revenue Stream** | `YY1_RevenueStream` | **Required**. Value List. | e.g., Marketplace. | | **Billing Cycle** | `YY1_BillingCycle` | **Required**. Value List. | e.g., Monthly. | | **Shipping Addr Ref** | `AddressID` | **Required**. | Selection from Main Addresses. | | **Bank Account**

Ref | `BankID` | Optional. | Selection from Main Banks. | | **Email Contacts** | `A_AddressEmailAddress` | **Required**. | List of email contacts. |

5. 🔌 API & Integration Specifications

A. Inbound API: `POST /service/mdm/CoupaRequest`

- **Payload:** JSON object using the **SAP Tech Names** defined in Section 4.B (e.g., `OrganizationBPName1`, `TaxNumber1`).
- **Logic:**
  1. Validates payload against schema.
  2. Sets `Source System` = 'COUPA'.
  3. Sets `Request Status` = 'NEW'.
  4. Persists data to `MDMRequest` entity.
  5. Returns HTTP 201 (Created) with the Request ID.

B. Callback API: `PATCH /service/mdm/Requests(ID)`

- **Payloads Accepted:**
  - `{ "sapStatus": "Success", "sapBusinessPartnerID": "100123" }`
  - `{ "satelliteStatus": "Success" }`
  - `{ "sapStatus": "Error", "MDM Notes": "Error description from S/4" }`

C. Outbound Event: `Topic: mdm/request/submit`

- **Payload:**

- `requestID`: UUID of the request.
- `data`: The full, validated Business Partner payload (using standard SAP field names).
- `changePayload`: The JSON diff (if Change request).
- `associatedSystems`: List of other systems (e.g., `['Coupa']`).

6. 🔄 Detailed End-to-End Workflows

A. Scenario: New Business Partner Creation

1. **Initiation:**
    - *User (PI/Salesforce):* Logs in, clicks "Create", fills form, submits.
    - *System (Coupa):* Calls `POST /CoupaRequest`.
    - *Result:* Request created with `OverallStatus = 'NEW'`.
2. **MDM Review:**
    - MDM User opens the request from the Inbox.
    - Reviews data validity.
    - Clicks **"Submit for Integration"**.
    - *System:* Updates status to `PENDING_INTEGRATION` and emits `mdm/request/submit` event.
3. **Integration Execution (External):**
    - Integration Suite (IS) subscribes to the event.
    - IS calls S/4HANA API to create the BP.
    - IS calls CAP `PATCH` API: `sapStatus = 'Success'`, `sapBusinessPartnerID = '12345'`.

- IS calls Satellite System (e.g., Salesforce) to update with the SAP ID.
- IS calls CAP `PATCH` API: `satelliteStatus = 'Success'`, `satelliteSystemID = 'SF-999'`.

4. **Completion:**
    - MDM User sees the status updates in the UI.
    - "Mark as Completed" button becomes active.
    - User clicks it; status updates to `COMPLETED`.

B. Scenario: Change Request (Shared Partner)

1. **Initiation:**
    - Salesforce User searches for "Acme Corp".
    - Updates "Phone Number" from "111" to "222".
    - Clicks "Submit".
    - *System:* Calculates `changePayload` (`{phone: {old: 111, new: 222}}`), sets `OverallStatus = 'NEW'`.
2. **MDM Review:**
    - MDM User sees the "Change Log" highlighting the phone number change.
    - Clicks **"Submit for Integration"**.
    - *System:* Updates status to `PENDING_INTEGRATION`, emits event with `changePayload` and `associatedSystems=['Coupa']`.
3. **Integration & Notification:**
    - IS updates S/4HANA with the new phone number.

   - IS updates Salesforce (originating system).
   - *Parallel Step:* IS sees 'Coupa' in `associatedSystems`. IS sends an email to the Coupa Owner: "Acme Corp data has changed. Click here to review."
   - IS calls CAP `PATCH` API: adds row to `satelliteAcknowledgments` table for Coupa with status 'Pending'.
4. **Acknowledgment:**
   - Coupa Owner clicks the email link.
   - Opens "Change Acknowledgment View".
   - Sees "Phone changed from 111 to 222".
   - Clicks **"Acknowledge"**.
   - *System:* Updates `satelliteAcknowledgments` to 'Acknowledged'.
5. **Completion:**
   - MDM User marks the request as `COMPLETED` once the main integration steps are done. (Acknowledgment is non-blocking).