**Flow Design Document for Consent UI Application**

**Overview**

## **1. Overview**

The Consent UI application facilitates user consent collection during authentication with PingFederate. It integrates with PingFederate's Reference ID Adapter, Consent API, and CIAM to manage user attributes, fetch system-related data, display consent details, and handle user interactions. This document outlines the flow design for the Consent UI application. It describes the key components, data flow, and conditional logic to handle user consent during the login process. The system integrates with PingFederate, Consent API, and CIAM API to manage user authentication and consent management.

**Components**

1. **User**
   * The end-user interacting with the application through a web browser.
2. **PingFederate**
   * The identity provider handling user authentication and reference ID generation.
3. **Consent UI**
   * A central application (frontend and backend combined) that manages user consent.
4. **Consent API**
   * A service that stores and retrieves user consent records.
5. **CIAM API**
   * A service that provides system IDs associated with the user.

## 2. Process Flow

### 2.1 Login and Redirection

1. A customer initiates login through PingFederate.
2. PingFederate authenticates the user.
3. The PingFederate Reference ID Adapter redirects the user to the Consent UI.

### 2.2 Attribute Pickup Process

1. The Consent UI backend invokes the Attribute Pickup Service via the /ext/ref/pickup endpoint using the provided Reference ID.
2. The service retrieves the user's ID and other necessary attributes from PingFederate.
3. The backend processes and stores these attributes for further steps.

### 2.3 Consent Verification

1. The backend calls the Consent API's Read Consents endpoint to verify if the user has already provided consent.
2. If consent exists, the backend:
   * Redirects the user back to PingFederate.
   * Performs a drop-off process using the /ext/ref/dropoff endpoint to generate a new Reference ID.
   * Passes the Reference ID back to PingFederate for redirection.
3. If consent does not exist, the backend proceeds to the consent acquisition flow.

### 2.4 Fetching System Data and Consent Text

1. The backend calls the CIAM API to fetch the system IDs associated with the user.
2. The backend calls the Consent API to fetch the consent text/verbose that will be displayed to the user.

### 2.5 User Interaction with Consent UI

1. The UI displays the fetched consent text and a dropdown allowing the user to select multiple system IDs.
2. The user can either:
   * Agree and submit the form.
   * Deny, in which case access to the application is denied.

### 2.6 Handling User Submission

1. If the user agrees:
   * The backend calls the Consent API's Create Consent endpoint to save the consent data.
   * Redirects back to PingFederate via the drop-off process, generating a new Reference ID.
2. If the user denies:
   * Redirects back to PingFederate via the drop-off process, and access to the application is denied. The user is redirected to an error page.

**Data Flow**

1. **PingFederate → Consent UI:** REF and user attributes.
2. **Consent UI → Consent API:** Consent Read and Create operations.
3. **Consent UI → CIAM API:** Fetch system IDs.
4. **Consent UI → PingFederate:** Attribute Pickup and Dropoff operations.