Boosting digital adoption of MSME ecosystem

TSI DPDP Consent Management System

System Architecture Document

23rd May 2025
V1.0

TSI Tech Solutions Cooperative Foundation
A section 8 company

https://tsicoop.org

Table of Contents

Version History

Author(s)	Date	Version	Description
Satish Ayyaswami TSI Tech Solutions Cooperative Foundation	23/05/2025	0.1	Draft

Abbreviations, Terms and Definitions

DPDP Act	Digital Personal Data Protection Act 2023
CMS	Consent Management System
DF	Data Fiduciary
DP	Data Processor
CM & CA	Consent Manager & Consent Aggregator
DPB	Data Protection Board
SDK, API, WCAG	Software Development Kit, Application Programmable Interface, Web Content Accessibility Guidelines
PII	Personally Identifiable Information
MFA, RBAC	Multi factor authentication, Role based Access Control

Introduction

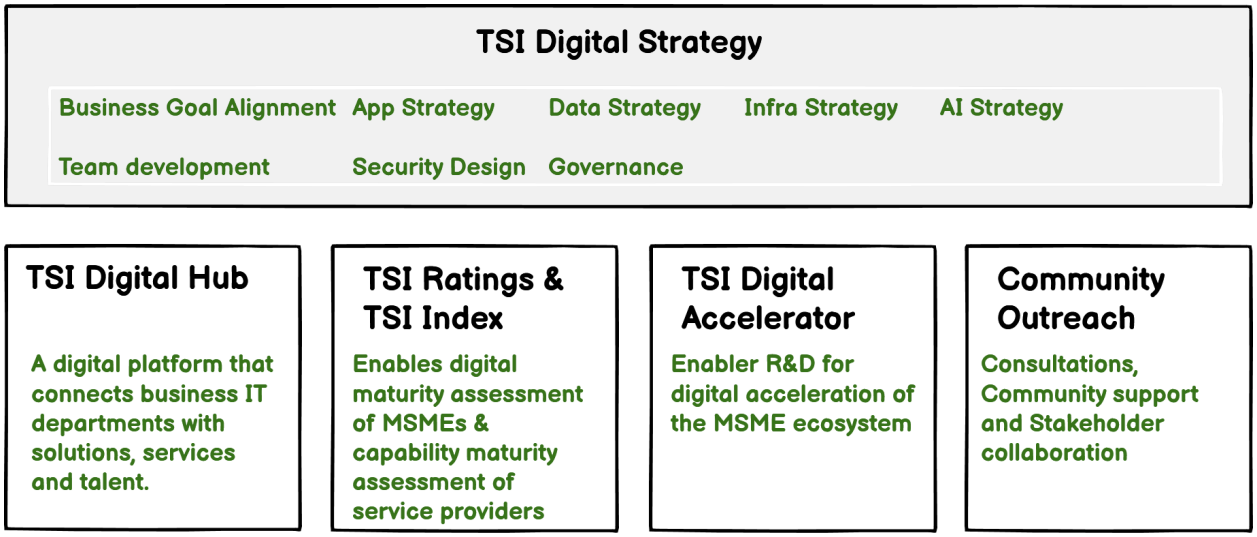
TSI Tech Solutions Cooperative Foundation (TSI Coop) initiative aims to address critical gaps in the MSME digital ecosystem by enabling businesses to implement successful technology strategies with the help of local service providers. Our mission also focuses on creating sustainable pathways for graduates from tier-3 and tier-4 institutions while fostering a more equitable and efficient domestic IT supply chain.

TSI stands for Technology & Social Impact. 'Coop' embodies the spirit of a cooperative economic model, where smaller providers and MSMEs collectively drive digital adoption within the ecosystem, fostering shared growth and opportunity.

While our IT machinery prioritizes foreign markets, GCCs, high profile startups and large enterprises, MSMEs are underserved. We aim to unlock their potential by facilitating the following:

- Discovery of niche providers, products, services and talent. Direct Interactions.
- Continuous digital maturity assessment of MSMEs ecosystem participants, helping stakeholders' identify areas for improvement
- Enabler R&D for MSME ecosystem
- Community support and stakeholder collaboration

Our offerings for the MSME ecosystem below:



TSI DPDP Consent Management System is a key component of our TSI Digital Accelerator program.

Scope

This document outlines the System Architecture of the TSI DPDP Consent Management System. It is assumed that the readers are familiar with the requirements of Digital Personal Data Protection Act 2023.

Our solution is developed as a simpler open source reference implementation under MIT License that can be deployed out-of-the-box by a smaller Consent Manager or Consent Aggregator serving the Indian MSME ecosystem. The solution can be enhanced by system integrators to meet the needs of larger organizations / large Consent Aggregators..

This document covers:

- Functional and Technical design of key modules
- Integration approaches with Data Fiduciary/ Processor systems and the Data Protection Board
- Security, deployment, and operational considerations.

Intended Audience

Architects, Developers, DevOps, QA, Project Managers, Business Stakeholders

High level System Goals

Enable DPDP Act Compliance: Provide tools for DFs and DPs to meet consent, data principal rights, retention, and reporting obligations.
Foster Trust & Transparency: Empower Data Principals with control over their data and clear understanding of processing.
Reference Implementation: We want to create a blueprint solution that the industry and local service providers can refer to and learn. The idea is to create something similar to Java PetStore Application & Glassfish Server that Sun Microsystems created to promote J2EE development and adoption during the dotcom days.
Robustness & Ease of use: We want to prioritize installation simplicity & system stability. The out-of-the-box solution should be very easy for a small-time IT Solution Providers in a tier-2/tier-3 town to implement the DPDP solution for the local hospitals, manufacturing units etc.

Facilitate Customization: Allow System Integration (SI) partners to tailor the solution for specific enterprise needs or a large Consent Aggregator that processes millions of consents per day..
Reduce User Fatigue: Enable a unified consent management experience, down the road, via Digilocker Wallet across multiple DFs.

Key Business Requirements Summary

- Collect explicit, informed, granular, and unambiguous consent.
- Manage consent lifecycle (collection, validation, update, renewal, withdrawal).
- Enable Data Principals to exercise their rights (Access, Correction, Erasure, Grievance Redressal).
- Provide dashboards for DPOs, Processors, Auditors, and Users.
- Support multilingual consent policies and user interfaces.
- Securely store all consent records and audit trails.
- Facilitate secure communication and reporting with the DPB.
- Allow for flexible deployment (on-prem/cloud).
- Support linking anonymous consent to authenticated user identities.
- Provide an SDK for easy integration by Fiduciary/Processor clients.
- Future: Support for unified consent management experience via DigiLocker and Data Wallets.

Architectural Goals

Installation Simplicity

- The Consent Manager installation should be as simple as installing a proper open source database in a system
- For organisations that have only websites and simpler data collection processes, the integration steps should be a simple configuration of Consent Policy and dropping a Javascript
- For slightly more complex integrations, an implementation guide with examples should be available for local IT solution partners to help their clients easily comply with the DPDP Act.

Security & Data Privacy:

- DPDP Act Compliance: Foremost priority.
- Data Confidentiality: Encryption at rest and in transit.
- Data Integrity: Prevention of unauthorized modification, robust audit trails.

- Authentication: Strong Password with Email OTP (2nd factor) for CMS users, API Keys for programmatic access.
- Authorization: Granular RBAC.
- Vulnerability Management: Secure coding, regular scanning/testing.

Performance:

- Fast enough consent banner/form loading times.
- Real-time consent validation API responses.
- Efficient data processing for audit logs and reports.

Reliability & Availability:

- Stable single node installations.
- Support for backup and disaster recovery.

Usability & Accessibility:

- Intuitive and user-friendly interfaces for all user types.
- WCAG 2.1 Level AA compliance for all user-facing components.
- Multilingual support.

Easy Maintenance & Support:

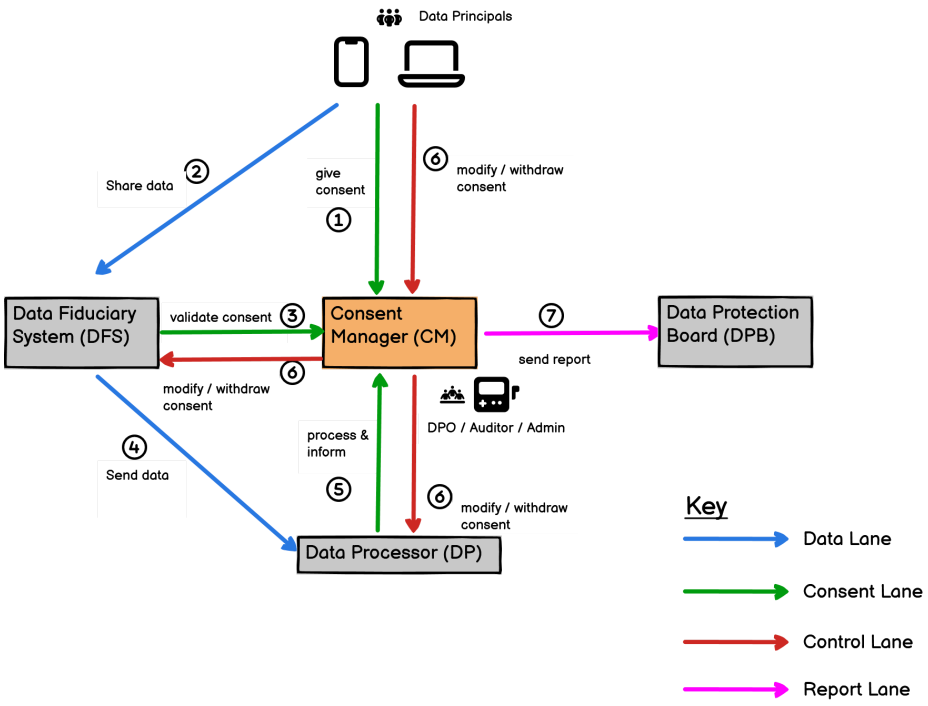
- During the initial stage, an automated purge from the DF system may be infeasible especially for smaller organisations. The system should allow manual rectification by DPOs along with evidence submission.
- Timely notifications for the principal & data protection stakeholders for prompt compliance
- One click report submission to the DPB

Interoperability & Extensibility:

- Open standards for APIs (REST, JSON).
- Support for integrations (TOTP,OAuth etc).
- Future integration with national platforms (DigiLocker, API Setu).

Functional Design

1. High Level Design



This Level 0 Data Flow Diagram (DFD) illustrates the high-level architecture of a Consent Management System.

It shows how the **Data Fiduciary System (DFS)**, interacts with a **Consent Manager (CM)** to collect consent (1) before receiving data (2). The DFS validates consent with the CM (3) before sending data (4) to a **Data Processor (DP)**. The DP also processes and notifies the CM (5). When the user modifies/withdraws consent, the CM notifies DFS and DP to limit/delete the data (6).

Finally, the system allows the **Data Protection Officer (DPO)** to generate and submit a single-click report to the **Data Protection Board (DPB)** (7), ensuring regulatory compliance.

2. CMS Server

2.1. Installation & Setup

Initial Server Setup & Software Installation: The administrator first deploys the open-source solution on the server, sets up docker .env file with postgresql & email configurations and ensures network accessibility. A batch script creates the database schema and loads the master data like roles and lookups.
First-Time Admin Account Creation: When the CMS instance is booted, the system prompts for the creation of a primary administrator account, requiring a strong, unique password.
OTP Verification: The system verifies Administrator email, completes the setup process and directs the Administrator to login.
Administrator Login: The administrator logs in using email, password and Email OTP (2nd factor authentication)
Quick Tour: After the first-login, the administrator is given a quick tour of the CMS.

2.2. Configure Consent Manager

The CM is designed to facilitate transparent and compliant personal data consent management for Data Fiduciaries, as mandated by the DPDP Act, 2023. It will support two primary operational modes:

- **Single Mode (Data Fiduciary-Specific):** The CM instance is deployed or dedicated to serving the consent management needs of a *single* Data Fiduciary. This mode suits when the Data Fiduciary decides to deploy the Consent Manager by itself.
- **Aggregator Mode (Centralized SAAS):** The CM acts as a central platform (e.g., hosted as a service) that manages consent for *multiple, distinct Data Fiduciaries*. This mode suits when the Data Fiduciary decides to work with an external partner for Consent Management. The external partner (Consent Aggregator) will manage consents on behalf of the data fiduciaries.

The purpose of this module is to

- To register and maintain essential identification details of a Consent Manager.
- To establish the Consent Manager's digital presence (domain, CNAME) where the CMS will operate.

- To indicate if the Consent Manager works exclusively for a single Data Fiduciary or Consent Aggregator mode (i.e., multiple fiduciaries)

Key Information Captured (Consent Manager):

- **Consent Manager Id:** The ID which is registered with the Data Protection Board (DPB) of India.
- **Consent Manager Name:** Full legal name of the Consent Manager (e.g., "SecureFin Consent Aggregator Pvt. Ltd.").
- **Contact Information:**
  - Primary Contact Person Name
  - Email Address (for official communication, e.g., privacy@securefinca.com)
  - Phone Number
  - Registered Address
- **Primary Domain(s):** The main website domain(s) where the Consent Manager collects personal data (e.g., `securefinca.com`).
- **CMS Deployment CNAME:** The specific subdomain or CNAME where the CMS will be hosted (e.g., `consent.securefinca.com`).
- **DNS TXT Record for Validation:** A unique token generated by the CMS for the Consent Manager to add to their DNS TXT records, used for domain ownership verification.
- **Mode of Operation (Single/Multiple):**
  - `is_aggregator`: Boolean flag.
- **Status:** Active/Inactive.
- **Creation/Last Updated Timestamps & User:** For auditing.

Features/Capabilities:

- **Consent Manager Creation:** Ability to register Consent Manager
- **Profile Editing:** Modify existing Contact Person Name, Address etc
- **Domain Validation Trigger:** Initiate the DNS TXT record validation process.
- **Validation Status Display:** Show the current validation status of the domain.

Functional Requirements:

- **Uniqueness:** Consent Manager ID registered with the Data Protection Board (DPO)
- **Mandatory Fields:** Ensure core fields (Name, Contact Details, Domain, CNAME, Primary Contact Email) are mandatory.
- **DNS TXT Record Generation:** The system must generate a unique, cryptographically secure TXT record string for each domain validation request.
- **Automated DNS Lookup:** The CMS must periodically (or on-demand) perform DNS lookups to verify the presence and correctness of the required TXT record.
- **Status Management:** Update the validation status (e.g., "Pending Validation", "Validated", "Validation Failed").
- **Alerting:** Notify the CMS admin if validation fails or expires.
- **Audit Trail:** Log all creation, modification, and deletion events related to Consent Manager profiles.

Workflow:

Admin Initiates: CMS Admin navigates to "Consent Manager" and clicks "Add Consent Manager".

Input Details: Admin enters Consent Manager Id, Name, Contact Info, Primary Domain, and desired CMS Deployment CNAME.

TXT Record Generation: Upon submission, the CMS generates and displays the unique DNS TXT record string.

DNS Action: The Consent Manager DNS Admin adds this TXT record to their domain's DNS settings.

CMS Validation: The CMS backend automatically performs a DNS lookup to confirm the TXT record.

Status Update: The CMS updates the Consent Manager’s profile status to "Validated" if successful, or "Validation Failed" otherwise.

Configuration Linkage: Once validated, the Consent Manager profile can be linked to onboard Data Fiduciaries within the CMS.

2.3. Configure Data Fiduciary

This module allows the CMS administrator to define and manage the profile of a Data Fiduciary, establishing their identity and digital footprint within the consent ecosystem for compliance and operational purposes.

The purpose of this module is to

- To register and maintain essential identification details of a Data Fiduciary.
- To establish the Data Fiduciary's digital presence (domain, CNAME) that the CMS can validate.
- To enable verification of the Data Fiduciary's ownership over the specified domain.
- Enable Cross Origin Resource Sharing (CORS) for the DF domain
- To link the Data Fiduciary to their specific privacy policies and consent configurations.
- To establish connectivity with Data Fiduciary’s Service Adapter for facilitating automated data purge.

Key Information Captured (Data Fiduciary Profile):

- **Fiduciary ID:** Unique identifier for the Data Fiduciary within the CMS (system-generated).
- **Fiduciary Name:** Full legal name of the Data Fiduciary (e.g., "MSMELoan Solutions Pvt. Ltd.").
- **Contact Information:**
    - Primary Contact Person Name
    - Email Address (for official communication, e.g., privacy@msmeloan.com)
    - Phone Number
    - Registered Address
- **Primary Domain(s):** The main website domain(s) where the Data Fiduciary collects personal data (e.g., `msmeloan.com`).
- **DNS TXT Record for Validation:** A unique token generated by the CMS for the Data Fiduciary to add to their DNS TXT records, used for domain ownership verification.
- **Service Adapter Webhook Registration**: Service Adapter enables automated purge at the Data Processor via Webhook
- **DPDP Act Specifics (Optional/Future):**
    - `is_significant_data_fiduciary`: Boolean flag.

    - DPO Contact Details (if SDF).
    - Registration ID with DPB (if applicable).
- **Status:** Active/Inactive.
- **Creation/Last Updated Timestamps & User:** For auditing.

Features/Capabilities:

- **Fiduciary Profile Creation:** Ability to add new Data Fiduciary entries.
- **Profile Editing:** Modify existing Data Fiduciary details.
- **Domain Validation Trigger:** Initiate the DNS TXT record validation process.
- **Validation Status Display:** Show the current validation status of the domain.
- **Profile Deactivation:** Temporarily disable a Data Fiduciary's configuration in the CMS.
- **Dashboard View:** Provide a summary list of all configured Data Fiduciaries with their key details and validation status.

Functional Requirements:

- **Uniqueness:** Fiduciary ID and Primary Domain must be unique within the CMS.
- **Mandatory Fields:** Ensure core fields (Name, Domain, CNAME, Primary Contact Email) are mandatory.
- **DNS TXT Record Generation:** The system must generate a unique, cryptographically secure TXT record string for each domain validation request.
- **Automated DNS Lookup:** The CMS must periodically (or on-demand) perform DNS lookups to verify the presence and correctness of the required TXT record.
- **CORS enablement:** Set `Access-Control-Allow-Origin` for the DF domain in the HTTP header
- **Service Adapter Configuration**:Webhook Url and X-Api-Key configuration for purge notifications
- **Status Management:** Update the validation status (e.g., "Pending Validation", "Validated", "Validation Failed").
- **Alerting:** Notify the CMS admin if validation fails or expires.
- **Audit Trail:** Log all creation, modification, and deletion events related to Data Fiduciary profiles.

Workflow:

Admin Initiates: CMS Admin navigates to "Manage Data Fiduciaries" and clicks "Add New Fiduciary".

Input Details: Admin enters Fiduciary Name, Contact Info, Primary Domain, and desired CMS Deployment CNAME.

TXT Record Generation: Upon submission, the CMS generates and displays the unique DNS TXT record string.

Fiduciary Action: The Data Fiduciary's IT team adds this TXT record to their domain's DNS settings.

CMS Validation: The CMS backend automatically performs a DNS lookup to confirm the TXT record.

Status Update: The CMS updates the Fiduciary's profile status to "Validated" if successful, or "Validation Failed" otherwise.

Configuration Linkage: Once validated, the Data Fiduciary's profile can be linked to specific consent policies, cookie configurations, and user accounts within the CMS.

2.4. Policy Definition

This module provides the administrative interface and backend logic for Data Fiduciaries to define, manage, and publish their comprehensive personal data consent policies. It ensures that all data processing activities are transparently articulated, categorized by purpose and legal basis, and presented in multiple languages to Data Principals.

The purpose of this module is

- To enable Data Fiduciaries to create, manage, and version clear, comprehensive policies detailing all personal data processing activities (beyond just cookies).
- To ensure policies explicitly state the purpose, legal basis, data categories, and recipients for each processing activity, as mandated by the DPDP Act.
- To support multilingual presentation of all policy content to Data Principals.
- To serve as the authoritative source for consent request content and the basis for consent record validation.

Key Information Captured

An Example Policy JSON in Annexure A**.**

- **Policy Metadata:**
  - policy_id: Unique identifier (auto-generated or user-defined).
  - version: Policy version number (e.g., "1.0", "1.1").
  - effective_date: Date/time when this version becomes active.
  - jurisdiction: Primary legal jurisdiction (e.g., "IN" for India).
  - status: Draft, Active, Archived, Expired.
- **Data Fiduciary Information:** Link to the associated Data Fiduciary profile from the "Configure Data Fiduciary" module.
- **Multilingual Content (languages object):** For each supported language (e.g., 'en', 'ta', 'hi'):
  - title: Main title of the privacy policy/consent notice.
  - introduction: General overview and commitment to privacy.
  - general_purpose_description: Summary of overall data processing reasons.
  - **data_processing_purposes (Array of objects):** For each distinct processing purpose:
    - id: Unique, language-agnostic identifier (e.g., "purpose_kyc_validation").
    - name: Localized display name (e.g., "KYC Validation").
    - description: Localized detailed explanation of the purpose.
    - legal_basis: Localized statement of the legal ground (e.g., "Consent (DPDP Act, Section 6(1)(a))", "Legal Obligation (DPDP Act, Section 7(h))").
    - data_categories_involved: List of associated ids from data_categories_details.
    - recipients_or_third_parties: List of specific Data Processors/Fiduciaries (from "Configure Data Processors" module) or types of recipients.
    - retention_period: Specific duration (e.g., "5 years post-account closure").
    - is_mandatory_for_service: Boolean if core service depends on consent for this purpose.
    - is_sensitive: Boolean flag indicating if sensitive personal data (e.g., Aadhaar, biometrics) is primarily processed for this purpose.

    - **data_categories_details (Array of objects):** Definitions for each type of personal data collected.
        - id: Unique identifier (e.g., "aadhaar_number").
        - name: Localized display name.
        - description: Localized detailed description.
        - is_sensitive: Boolean flag indicating if this is sensitive PII.
    - **data_principal_rights_summary**: Localized summary of DPDP Act rights.
    - **grievance_redressal_info**: Localized contact details for DPO/grievances.
    - **buttons & links**: Localized text for UI elements and full policy URLs.
    - **important_note**: Any crucial disclaimers (e.g., about mandatory data).

Features/Capabilities:

- **Policy Creation/Editing Wizard:** A structured interface to define all aspects of a new or existing policy.
- **Version Control & Publishing:**
    - Create new policy versions (e.g., by duplicating an active one).
    - Publish a version to make it Active.
    - Maintain a complete history of all policy versions (Draft, Active, Archived).
- **Multilingual Content Editor:** Dedicated sections or tabs for entering/managing content for all supported languages, ensuring consistency across translations.
- **Purpose & Data Category Management:**
    - Define custom processing purposes and data_categories_details.
    - Allow linking data_categories_involved to predefined categories.
    - Enable linking recipients_or_third_parties to registered Data Processors.
- **Legal Basis Mapping:** Provide options to select and define appropriate legal_basis for each purpose, aligning with DPDP Act sections.

- **Preview Functionality:** Allow administrators to preview how the consent notice/preference center will appear for different languages and device types before publishing.
- **Policy Audit Trail:** Log all policy creation, versioning, publication, and deletion events.

Functional Requirements:

- **Mandatory Fields Validation:** Enforce all critical fields for a valid policy (e.g., ID, Version, Titles, Purpose/Category IDs, Descriptions, Legal Basis).
- **Uniqueness Constraints:** Ensure unique policy_id, version, and ids within purposes and data_categories_details.
- **Immutability of Active Policies:** Once a policy version is Active, it becomes read-only; any changes necessitate creating a new version.
- **Comprehensive Localization:** System must ensure that all required text fields have corresponding localized content for all declared languages before activation.
- **Data Type Flagging:** Clearly flag is_sensitive: true for sensitive data categories.
- **Audit Logging:** Detailed logging of all administrative actions within this module.
- **Relationship Enforcement:** Ensure data_categories_involved and recipients_or_third_parties accurately link to existing definitions in other modules.

Workflow:

1. **Admin Access:** CMS Admin/DPO navigates to the "Personal Data Policies" module.
2. **Initiate Policy/Version:** Admin clicks "Create New Policy" or "Create New Version" for an existing policy.
3. **Define Core Metadata:** Admin inputs policy_id, version, effective_date, jurisdiction, and selects the Data Fiduciary.
4. **Input Multilingual Content:** For each language, admin populates title, introduction, general_purpose_description, and defines/selects data_processing_purposes and data_categories_details.
5. **Configure Purposes:** For each purpose, admin specifies its id, name, description, legal_basis, data_categories_involved,

recipients_or_third_parties, retention_period, is_mandatory_for_service, and is_sensitive flags.

6. **Configure Data Categories:** For each data category, admin specifies its id, name, description, and is_sensitive flag.
7. **Preview & Validate:** Admin uses the preview feature to review content across languages. System runs validations.
8. **Publish/Save Draft:** Admin saves as a draft or publishes the policy to make it Active.

2.5. Consent Lifecycle

This module governs the entire journey of a Data Principal's personal data consent, from its initial capture to its eventual termination, ensuring continuous compliance with the DPDP Act's principles of control, transparency, and accountability.

The purpose of this module is

- To manage the complete consent lifecycle, from initial collection to final withdrawal or expiry.
- To link seamlessly associate anonymous consent records with a definitive, authenticated Data Principal ID once a user logs in or registers.
- To ensure consent remains valid, up-to-date, and reflects the Data Principal's true preferences at all times data is processed.
- To provide a robust, auditable record of consent status throughout the Data Principal's relationship with the Data Fiduciary.
- To dynamically adjust data processing activities based on the current consent status.

Core Principles & Requirements:

- **DPDP Act Compliance:** Consent must always be Free, Specific, Informed, Unconditional, and Unambiguous. Withdrawal must be as easy as granting.
- **Auditability:** Every stage and change in consent status must be securely logged.
- **Granularity:** Consent must be managed at the level of specific purposes and data categories.

- **Real-time Enforcement:** Changes in consent status must trigger immediate adjustments to data processing activities.
- **Version Control:** All consent records are tied to a specific version of the consent policy.

Key Stages & Functional Requirements:

Consent Collection (Initial Grant)

- **Objective:** Obtain first-time, valid consent from an Anonymous User / Data Principal.
- **Functional Requirements:**
    - **Dynamic Policy Retrieval:** Fetch the active Personal Data Consent Policy (from "Policy Retrieval") based on Fiduciary, Jurisdiction, and Language.
    - **Form Rendering:** Dynamically display the interactive consent form/preference center ("Consent Form Rendering") according to the retrieved policy, defaulting non-mandatory purposes to opt-out.
    - **Affirmative Action Capture:** Record explicit user interaction (e.g., button clicks, toggle selections).
    - **Initial Record Creation:** Compile a complete consent payload and submit it to "Consent Storage" for a new, active consent record.
    - **Client-Side Persistence:** Store a temporary client-side cookie/local storage entry indicating active consent to suppress repeated prompts.

Link Principal (Once the user registers / logs in)

- **Objective:** To associate anonymous consent records with a definitive, authenticated Data Principal ID
- **Functional Requirements:**

  Anonymous ID Detection: The client-side system (e.g., website frontend JavaScript) MUST detect the presence of a temporary, anonymous user identifier (e.g., from localStorage or a transient cookie).

  Authenticated ID Acquisition: Upon successful user login or registration with the Data Fiduciary's core system, the client-side

system MUST securely obtain the definitive, server-generated authenticated_data_principal_id.

Automatic Linking Trigger: The client-side system MUST automatically initiate the linking process if an anonymous_user_id is present and differs from the authenticated_data_principal_id acquired after login/registration.

Secure Backend API: The CMS backend MUST expose a secure API endpoint to receive both the anonymous and authenticated IDs.

Atomic Database Reconciliation: The backend service processing the linking request MUST perform an atomic database transaction. This transaction MUST update all consent_records (and other relevant tables) where the user_id matches the anonymous_user_id, replacing it with the authenticated_data_principal_id.The system MUST ensure that no consent history is lost during this process.

Audit Logging:The backend service MUST record every linking event in the audit_logs table, documenting both the anonymous_user_id and the authenticated_data_principal_id, along with the timestamp and the actor initiating the link.

Consent Validation (Ongoing Check)

- **Objective:** Verify that current data processing activities are covered by active, valid consent.
- **Functional Requirements:**
    - **Active Consent Lookup:** Provide an API endpoint (from "Consent Storage") to quickly retrieve the is_active_consent record for a given user_id.
    - **Purpose-Based Validation:** For any proposed data processing, check if the specific purpose and associated data categories are marked as consent_granted: true in the active record.
    - **Policy Version Check:** If the policy_id_version_reference in the stored consent record differs from the current active policy, flag for renewal/re-prompt.
    - **Automated Enforcement:** Block or permit data processing activities (e.g., script execution, data sharing) based on validation outcome.

Consent Update (Modification of Preferences)

- **Objective:** Allow Data Principals to modify their existing consent preferences.
- **Functional Requirements:**
    - **Preference Center Access:** Provide an easily accessible link (e.g., "Privacy Choices") to re-open the interactive consent form ("Consent Form Rendering").
    - **Pre-population:** Populate the form with the user's *current* active consent preferences.
    - **Granular Modification:** Allow changes to individual purpose toggles.
    - **New Record Creation:** On "Save Choices," compile a new consent payload and submit it to "Consent Storage." This new record replaces the previous active one (marking the old one as inactive).
    - **Immediate Application:** Apply the updated consent (e.g., activate/deactivate scripts) upon successful storage.

Consent Renewal (Proactive Re-prompt)

- **Objective:** Periodically re-obtain consent, especially when consent expires or policies change, to maintain validity and freshness.
- **Functional Requirements:**
    - **Expiry Monitoring:** System monitors timestamp and CONSENT_EXPIRY_DAYS (from "Frontend Loading") in client-side storage.
    - **Policy Change Detection:** System compares the policy_id_version_reference in stored consent with the current active policy on the backend.
    - **Re-prompt Trigger:** If consent has expired or the policy has changed, trigger the re-display of the consent banner/form on the next user visit.
    - **Graceful Handling:** Allow users to continue with essential services if they choose not to renew immediately, but persist re-prompting until a new decision is made.

Consent Withdrawal (Opt-out/Erasure Request)

- **Objective:** Enable Data Principals to easily withdraw all or specific parts of their consent, leading to cessation of processing.
- **Functional Requirements:**

- **Easy Withdrawal Interface:** "Reject All Non-Essential" button on banner/preference center, or a specific "Withdraw Consent" option for individual purposes.
- **Withdrawal Record:** Create a new consent record in "Consent Storage" marking the preferences as false for specified purposes, or globally denied.
- **Timestamping:** Record the exact timestamp of withdrawal.
- **Cessation of Processing:** Immediately trigger the cessation of data processing for the withdrawn purposes (e.g., deactivate scripts, stop sharing data with Data Processors).
- **Data Erasure/Anonymization:** Initiate processes (via "Data Retention Policy Configuration") to delete or anonymize data processed solely based on the withdrawn consent, after accounting for legal retention obligations.
- **Confirmation:** Provide a confirmation message to the Data Principal.
- **Audit Trail:** Log all withdrawal actions and subsequent data processing changes.

2.6. Consent Provenance

Consent Provenance ensures every consent decision's complete, tamper-evident history is recorded. It captures details like policy version, timestamp, IP, and specific choices. Changes create new, active records while deactivating old ones, ensuring an immutable audit trail of who, what, when, where, why, and how consent was given or changed, vital for DPDP Act compliance and dispute resolution.

Key design principles below

- The Consent Record should be immutable
- Every time the User Saves Consent Form, a new Consent Record should be created and should be marked ACTIVE. All the earlier Consent Records associated with the user should be deactivated

2.7. Configure Data Processors

This module enables a Data Fiduciary to define, manage, and audit the details of third-party entities (Data Processors) that process personal data on their behalf. This ensures transparency, accountability, and compliance with contractual obligations.

The purpose of this module is to

- To register and maintain identifiable details of Data Processors used by a Data Fiduciary.
- To establish the Data Processor's digital presence (domain, CNAME) that the CMS can validate.
- To enable verification of the Data Processor's ownership over the specified domain.
- To explicitly define the specific purposes for which each Data Processor is authorized to process data.
- To link Data Processors to relevant data categories they handle.
- To maintain records of Data Processing Agreements (DPAs) or contracts.
- To facilitate the enforcement of consent preferences for data sharing with these processors.
- To establish connectivity with Data Processor's Service Adapter for facilitating automated data purge.

Key Information Captured (Data Processor Profile):

- **Processor ID:** Unique identifier for the Data Processor within the CMS (system-generated).
- **Processor Name:** Legal name of the Data Processor (e.g., "AnalyticsPro Inc.", "CloudStorage Solutions Ltd.").
- **Primary Domain(s):** The main website domain(s) where the Data Processor processes personal data (e.g., msmeanalyticspro.com).
- **DNS TXT Record for Validation:** A unique token generated by the CMS for the Data Processors to add to their DNS TXT records, used for domain ownership verification.
- **Contact Information:**
    - Primary Contact Person Name
    - Email Address (for official communication regarding data processing)

    - Phone Number
    - Registered Address
- **Associated Data Fiduciary:** Link to the Data Fiduciary entity that utilizes this Processor
- **Processing Purposes:**
    - A list of predefined purposes (e.g., "Marketing Analytics", "Credit Scoring", "Cloud Storage", "Customer Support"). Each purpose should be linked to the consent purposes defined in the "Configure Data Fiduciary" module's consent policies.
- **Data Categories Processed:** A list of specific data categories (e.g., "Email Address", "Browse History", "Financial Data", "Contact Details") that this Processor handles for the defined purposes.
- **Data Processing Agreement (DPA) Reference:**
    - DPA ID/Reference Number
    - DPA Effective Date
    - DPA Expiry Date (if applicable)
    - Link to stored DPA document.
- **Processor Jurisdiction:** The primary location/jurisdiction of the Data Processor.
- **Security Measures:** A high-level description or attestation of the security measures implemented by the Processor.
- **Service Adapter Webhook Registration**: Service Adapter enables automated purge at the Data Processor via Webhook configuration
- **Status:** Active/Inactive (e.g., if a processor relationship is terminated).
- **Creation/Last Updated Timestamps & User:** For auditing.

Features/Capabilities:

- **Processor Profile Creation:** Ability to add new Data Processor entries.
- **Profile Editing:** Modify details, update processing purposes, or change associated data categories for an existing Processor.
- **Domain Validation Trigger:** Initiate the DNS TXT record validation process.
- **Validation Status Display:** Show the current validation status of the domain.
- **DPA Management:** Record and update DPA details.
- **Status Management:** Activate or deactivate Processor profiles.

- **Link to Consent:** Ensure that purposes defined here can be linked to the consent options presented to Data Principals.
- **Search & Filter:** Easily find processors by name, purpose, or associated Data Fiduciary.
- **Dashboard View:** Provide a summary list of all configured Data Processors.

Functional Requirements:

- **Uniqueness:** Processor ID must be unique within the CMS.
- **Mandatory Fields:** Ensure core fields (Name, Domain, Primary Contact Email) are mandatory.
- **DNS TXT Record Generation:** The system must generate a unique, cryptographically secure TXT record string for each domain validation request.
- **Automated DNS Lookup:** The CMS must periodically (or on-demand) perform DNS lookups to verify the presence and correctness of the required TXT record.
- **Status Management:** Update the validation status (e.g., "Pending Validation", "Validated", "Validation Failed").
- **Purpose Enforcement:** The system must use the configured purposes to validate data sharing requests (e.g., ensuring data is only shared with a Processor for a purpose consented to by the Data Principal and defined here).
- **Relational Linkage:** Must have a clear, enforceable link to the Data Fiduciary entity.
- **Service Adapter Configuration**: Webhook URL & X-Api-Key header for automatic purge integration
- **Audit Trail:** Log all creation, modification, and deactivation events related to Data Processor profiles, including who made the changes and when.
- **Data Categories Validation:** When associating purposes, enforce selection from predefined data categories (from the Data Fiduciary's policy).

Workflow:

Admin Initiates: CMS Admin (or Data Fiduciary's DPO/Privacy Officer) navigates to "Manage Data Processors" and clicks "Add New Processor".

Input Details: Admin enters Processor Name, Contact Info, Primary Domain and selects the associated Data Fiduciary.

TXT Record Generation: Upon submission, the CMS generates and displays the unique DNS TXT record string.

Processor Action: The Processor's IT team adds this TXT record to their domain's DNS settings.

CMS Validation: The CMS backend automatically performs a DNS lookup to confirm the TXT record.

Status Update: The CMS updates the Processor's profile status to "Validated" if successful, or "Validation Failed" otherwise.

Define Processing Scope: Admin selects the specific `Processing Purposes` (e.g., "Marketing Analytics") and lists the `Data Categories Processed` (e.g., "Email Address", "Browse History") for each purpose.

DPA Information: Admin inputs DPA ID, effective date, and any other relevant contractual details.

Submission & Review: Admin submits the profile. The system stores it and makes it available for linking to consent requests and data sharing rules.

Periodic Review: Regularly review Processor profiles to ensure purposes, data categories, and DPA details remain accurate and current.

2.8. Register Data Fiduciary with Data Protection Board

This module enables a Data Fiduciary to formally register its presence and its associated CMS with the Data Protection Board of India, facilitating secure, direct communication and automated compliance reporting. The purpose of this module is

- To formally register the Data Fiduciary with the Data Protection Board as mandated by the DPDP Act (if such a registration process is implemented by DPB).
- To establish a secure, authenticated communication channel between the CMS (acting on behalf of the Data Fiduciary) and the DPB's systems.
- To enable automated or streamlined submission of mandatory reports (e.g., breach notifications, compliance audits) to the DPB.

- To facilitate direct, secure digital interactions between the Data Fiduciary's Data Protection Officer (DPO) and the DPB.

Key Information Captured/Managed:

- **Fiduciary ID (CMS Internal):** Link to the Data Fiduciary's profile configured in the "Configure Data Fiduciary" module.
- **DPB Registration ID:** The official registration number issued by the DPB (once obtained).
- **DPB Authentication Credentials:**
    - **Client Certificate & Private Key:** Generated/uploaded by the CMS for two-way SSL client authentication with DPB.
    - (Optional) API Key or other authentication tokens provided by DPB.
- **DPB Endpoint URL:** The specific URL for the DPB's API or submission portal.
- **DPO Contact Details (for DPB interaction):** Relevant contact information of the Data Fiduciary's DPO for direct communication with the DPB (e.g., email, dedicated portal ID).
- **Registration Status:** Current status of registration (e.g., "Not Registered", "Pending Approval", "Registered", "Revoked").
- **Last Successful Communication/Report Submission Timestamp:** For auditing and monitoring connectivity.
- **Creation/Last Updated Timestamps & User:** For auditing.

Features/Capabilities:

- **Registration Initiation:** Allows the CMS admin/DPO to initiate the registration process with the DPB.
- **Credential Management:**
    - Facilitate generation or upload of client certificates and private keys for two-way SSL.
    - Secure storage and management of these credentials.
    - Renewal/revocation of credentials.
- **Connection Test:** Provide a utility to test the secure connection to the DPB's endpoint.
- **Status Tracking:** Display the real-time registration status and any communication issues.
- **Report Submission Interface:** A mechanism (API/UI) to prepare and send structured reports (e.g., breach notifications, periodic compliance reports) to the DPB via the established secure channel.

- **Secure Messaging Gateway:** Provide an interface for secure, authenticated, peer-to-peer messaging between the DPO and the DPB.
- **Automated Retries/Notifications:** For failed report submissions or connection issues.

Functional Requirements:

- **Two-Way SSL Implementation:** The CMS must technically support and enforce two-way SSL (mutual TLS) for all communications with the DPB's designated endpoints. This means both the client (CMS) and server (DPB) authenticate each other using digital certificates.
- **Secure Credential Storage:** All cryptographic keys and authentication tokens must be stored securely (e.g., encrypted at rest, hardware security modules where appropriate).
- **Standardized API Integration:** The module must integrate with the DPB's official API specifications for report submission and communication.
- **Report Formatting:** Ensure reports conform to DPB-mandated data formats and structures.
- **Audit Trail:** Comprehensive logging of all registration attempts, credential management, report submissions (including status and timestamps), and communication failures.
- **Error Handling:** Robust error handling for network issues, authentication failures, and DPB service unavailability.
- **Role-Based Access:** Only authorized CMS admins/DPOs (as per User Role Management) can access and manage this module.

Workflow:

Admin Prepares: CMS Admin/DPO configures the Data Fiduciary profile, including DPO contact details.

Initiate Registration: Admin navigates to the "DPB Registration" module and initiates the process.

Credential Setup: If required, the CMS guides the Admin to generate/upload necessary client certificates.

Endpoint Configuration: Admin inputs the DPB's API endpoint URL.

Connection & Registration Request: The CMS attempts to establish a two-way SSL connection and sends the registration request payload (containing Fiduciary details) to the DPB.

DPB Response & Status Update: The CMS receives a response from the DPB (e.g., "Registration successful," "Pending review," "Error") and updates the registration status in the CMS. If successful, the DPB Registration ID is stored.

Ongoing Communication: The CMS periodically uses this secure channel for automated report submissions (e.g., breach notifications) or allows the DPO to send direct queries via the secure messaging gateway.

2.9. Data Fiduciary Dashboard

This module provides the Data Protection Officer (DPO) and other authorized Data Fiduciary personnel with a comprehensive, centralized interface for overseeing, managing, and reporting on all personal data protection and consent management activities, ensuring compliance with the DPDP Act. The purpose of this module is

- To provide a holistic view of the Data Fiduciary's data protection posture and compliance status.
- To enable efficient management of Data Principal requests and grievances.
- To facilitate proactive monitoring of data processing activities and policy adherence.
- To streamline the generation and submission of regulatory reports to the Data Protection Board (DPB).
- To provide immediate visibility into exceptions, breaches, and critical system notifications.

Key Information Displayed/Managed:

- **Overall Compliance Status Dashboard:**
    - Summary of active consent policies and their versions.
    - Consent rates (overall, by purpose).
    - Number of active/inactive Data Processors.
    - Overview of pending Data Principal rights requests.
    - Key performance indicators (KPIs) for privacy compliance.
- **Grievance Redressal & Resolution Tracking:**
    - List of all open and closed Data Principal grievances.

    - Status of each grievance (e.g., "New," "In Progress," "Pending DPO Review," "Resolved," "Escalated").
    - Assigned personnel, submission date, last updated date, and due date (SLA tracking).
    - Ability to view full grievance details, communication logs, and resolution steps.
    - Filtering/sorting by status, severity, and assigned personnel
- **Notifications & Alerts:**
    - Centralized feed of system-generated notifications:
        - **Critical Alerts:** Data breach warnings, system security alerts, failed data purge operations.
        - **Compliance Alerts:** Upcoming consent expiry for large user groups, policy version change notifications, upcoming regulatory reporting deadlines.
        - **Operational Alerts:** New Data Principal rights requests, new grievances.
- **Data Purge Handling:**
    - Summary of data scheduled for purge/anonymization based on retention policies.
    - List of successfully executed data purge operations including the Data Processors (what data, when, by whom, policy applied).
    - Reports on failed or incomplete purge operations (exceptions).
    - In the initial days of CMS implementation, an automated purge from the Data Fiduciary system may be infeasible especially for smaller organisations. The system should allow manual rectification along with evidence submission.
    - Ability to generate audit logs for specific purge events.
- **Exception Handling:**
    - Dashboard view of all identified system exceptions related to data processing, consent management, or policy enforcement.
    - Details of the exception (e.g., failed consent recording, invalid data format, policy conflict).
    - Status of exception resolution (e.g., "New," "Under Investigation," "Resolved").
    - Assigned personnel and resolution steps.
- **Report Submission to Data Protection Board (DPB):**

    - List of mandatory regulatory reports (e.g., Breach Notifications, Compliance Reports).
    - Status of each report (e.g., "Draft," "Ready for Submission," "Submitted," "Failed").
    - Submission history with timestamps and confirmation receipts (if provided by DPB).
    - Templates for various report types.

Features/Capabilities:

- **Secure DPO Login:** Robust authentication (SSO, MFA) and role-based access control (RBAC) to ensure only authorized personnel access the dashboard.
- **Centralized Oversight:** Aggregated view of all privacy-related metrics and operational statuses.
- **Grievance Workflow Management:** Assign grievances, update status, add internal notes, trigger communications to Data Principals.
- **Notification Management:** View, dismiss, filter notifications.
- **Data Purge Monitoring:** Track status and results of data retention policies.
- **Exception Triage:** Investigate, assign, and resolve exceptions.
- **Report Generation & Submission:**
    - Ability to populate required data into predefined report templates (e.g., DPB breach notification form).
    - Securely submit reports to the DPB via the "Register Data Fiduciary with DPB" secure channel.
    - View submission confirmations and any errors.
- **Audit Trail Access:** Direct access to granular audit logs of all actions performed within the CMS, especially those related to consent, policies, and data processing.
- **User/Processor/Policy Management Shortcuts:** Quick links to configure/manage Data Fiduciaries, Data Processors, Users, and Consent Policies.

Functional Requirements:

- **Authentication & Authorization:** Strictly enforce DPO/Admin roles (via "User Role Management").
- **Real-time Updates:** Dashboard data should be near real-time, leveraging efficient database queries and potentially caching.

- **Search & Filter:** Comprehensive search and filtering capabilities for all lists (grievances, purge reports, exceptions).
- **DPDP Act Compliance:**
    - **Accountability:** All actions taken on the dashboard are logged for audit.
    - **Transparency:** Provides necessary data for internal and external audits.
    - **Timeliness:** Facilitate adherence to regulatory deadlines for grievance resolution and report submission.
- **Data Visualization:** Use charts and graphs for quick understanding of trends (e.g., consent rates over time, grievance resolution SLAs).
- **Security:** Ensure data displayed on the dashboard is protected from unauthorized access.
- **Usability:** Intuitive UI/UX for efficient management by DPOs.

Workflow:

1. **DPO Login:** DPO authenticates securely via SSO/MFA.
2. **Dashboard Overview:** DPO sees a summary of critical alerts, new grievances, pending requests, and compliance KPIs.
3. **Action - Grievance Resolution:** DPO clicks on a "New Grievance" notification, reviews details, assigns it, updates status to "In Progress," communicates with Data Principal, and eventually marks as "Resolved."
4. **Action - Data Purge Review:** DPO reviews "Data Purge Reports" to confirm successful operations and investigate any exceptions.
5. **Action - Report Submission:** DPO navigates to "Report Submission," selects a report type (e.g., Breach Notification), reviews pre-populated data, and initiates secure submission to the DPB.
6. **Action - Exception Handling:** DPO reviews "Exception Handling" dashboard, identifies failed consent recordings, investigates root cause, and initiates corrective action.
7. **Ongoing Monitoring:** DPO regularly monitors dashboard for new alerts, trends, and ensures continuous compliance.

2.10. Data Processor Dashboard

This module provides Data Processors with a specialized interface to receive, manage, and report on personal data processing tasks assigned by Data Fiduciaries, ensuring adherence to instructions, data security, and efficient operational oversight. The purpose of this module is

- To centralize the management of data processing instructions received from various Data Fiduciaries.
- To provide visibility into the status of data processing tasks, including data receipt, processing, and purging.
- To alert the Data Processor to critical operational and data privacy events.
- To facilitate communication with Data Fiduciaries regarding processing tasks and exceptions.
- To support the Data Processor's obligation to assist the Data Fiduciary in meeting their compliance duties.

Key Information Displayed/Managed:

- **Dashboard Overview:**
    - List of associated Data Fiduciaries.
    - Summary of active data processing agreements/tasks.
    - Key metrics (e.g., data volume processed, task completion rates).
- **Data Process Setup (Instructions from Fiduciary):**
    - List of data processing agreements (DPAs) or service contracts with Data Fiduciaries.
    - Details of specific processing instructions from each Data Fiduciary:
        - Fiduciary ID and Name.
        - Processing Purpose (as defined by Fiduciary, e.g., "Marketing Automation," "Customer Support Analytics").
        - Data Categories to be Processed (e.g., "Email Address," "Browse History").
        - Data Retention Period (as per Fiduciary's instruction).
        - Data Transfer/Receipt Method.
        - Security Requirements.
        - Start/End Dates of Processing.

    - Status of setup (e.g., "Pending Approval," "Active," "Inactive").
- **Notifications:**
    - Centralized feed of alerts and communications specific to Data Processor tasks:
        - **New Processing Instructions:** Notification of new tasks from Data Fiduciaries.
        - **Data Fiduciary Requests:** Requests for data access, correction, or erasure from Data Principals (forwarded by Fiduciary).
        - **Instruction Updates:** Changes to processing scope or retention periods.
        - **Security Alerts:** Warnings about suspicious activities or potential vulnerabilities related to processed data.
        - **System Alerts:** Internal alerts regarding processing job failures or data integrity issues.
- **Data Purge Handling:**
    - Summary of data scheduled for deletion or anonymization as per Data Fiduciary's retention instructions.
    - List of successfully executed purge operations (what data, when, policy source).
    - Reports on failed or incomplete purge operations (exceptions).
    - Ability to manually correct failures along with evidence
    - Audit trail of purge confirmations sent to Data Fiduciary.
- **Exception Handling:**
    - Dashboard view of operational exceptions related to data processing tasks:
        - Failed data transfers, corrupted data, processing job errors.
        - Non-compliance with Fiduciary instructions.
        - Security incidents affecting data in their custody.
    - Details of the exception (e.g., error code, timestamp, affected data volume).
    - Status of exception resolution (e.g., "New," "Investigating," "Resolved").
    - Assigned personnel for resolution.
    - Communication log with Data Fiduciary regarding the exception.

Features/Capabilities:

- **Secure Authentication:** Access controlled via robust authentication (e.g., username/password, MFA) for authorized Data Processor personnel.
- **Instruction Acknowledgment:** Ability to confirm receipt and understanding of new or updated processing instructions from Data Fiduciaries.
- **Task Status Tracking:** Monitor progress and completion of assigned data processing tasks.
- **Notification Management:** View, filter, and mark notifications as read.
- **Purge Operation Oversight:** Review and confirm execution of data purge commands received from Fiduciaries.
- **Exception Logging & Triage:** Log new exceptions, assign them for investigation, and track resolution.
- **Secure Communication Channel:** Integrated messaging or reporting tools to communicate with Data Fiduciaries regarding tasks, issues, or requests.
- **Audit Trail Access:** Provide limited access to audit logs relevant to their specific processing activities (e.g., data transfers, purges).

Functional Requirements:

- **Fiduciary-Specific Views:** The dashboard must clearly segregate data and tasks by the originating Data Fiduciary.
- **Instruction Adherence:** The system should prevent the Data Processor from configuring or performing processing activities outside the scope of instructions received from the Data Fiduciary.
- **Automated Notifications:** Automatically generate notifications to the Data Fiduciary for critical events (e.g., data breach, failed purge, major processing error).
- **Auditability:** Every action taken on the dashboard, and every processing task executed, must be logged for audit.
- **Data Integrity & Security:** Ensure that data handled by the Processor system (and reflected on the dashboard) maintains integrity and is secured as per Fiduciary's instructions and DPDP Act.
- **Compliance with Retention:** The system must strictly adhere to the retention periods and purge instructions provided by the Data Fiduciary.

- **API Integration:** Robust APIs for receiving instructions, sending notifications, and confirming purge operations with the Data Fiduciary's CMS.

Workflow:

1. **Processor Login:** Authorized Data Processor personnel authenticate.
2. **Dashboard Overview:** View summaries of active Fiduciaries, pending instructions, and critical alerts.
3. **Action - New Instructions:**
    - Notification of "New Processing Instructions" from a Data Fiduciary appears.
    - Processor personnel review the "Data Process Setup" details (purpose, data categories, retention).
    - Acknowledge receipt of instructions.
4. **Action - Process Data:** Data Processor's internal systems perform the processing as per acknowledged instructions. Dashboard may show progress updates if integrated.
5. **Action - Data Purge:**
    - The system automatically flags data for purge based on Fiduciary's retention instructions.
    - Processor personnel review and confirm purge operations.
    - "Data Purge Reports" are updated, and confirmation sent to Fiduciary.
6. **Action - Handle Exception:**
    - An "Exception" notification appears (e.g., data transfer error).
    - Processor personnel investigate, update status, and communicate with the Data Fiduciary if needed.
7. **Ongoing Monitoring:** Processor continuously monitors the dashboard for operational health and compliance.

2.11. Auditor Dashboard

This module provides authorized auditors and administrators with a centralized, immutable, and searchable repository of all system activities and exceptions, enabling comprehensive compliance verification, incident investigation, and operational oversight. The purpose of this module is to

- To provide a transparent and auditable record of all critical events and actions within the Personal Data Consent Management System.

- To enable efficient review and investigation of privacy-related incidents, policy deviations, and system anomalies.
- To demonstrate accountability to Data Protection Authorities (e.g., DPB) by providing verifiable audit trails.
- To support internal compliance checks and security posture assessments.

Key Information Displayed/Managed:

- **Audit Log**:
    - **Timestamp:** Date and time of the event (UTC).
    - **User/System ID:** The user (or internal system process) that performed the action.
    - **Action Type:** Specific operation performed (e.g., "ConsentGranted", "PolicyPublished", "UserRoleAssigned", "DataPurgeExecuted", "DataAccessed").
    - **Entity Type:** The type of object affected (e.g., "ConsentRecord", "ConsentPolicy", "User", "DataFiduciary", "DataProcessor").
    - **Entity ID:** The unique ID of the specific object affected (e.g., consent record ID, policy ID and version, user ID).
    - **Context/Details:** JSON payload containing relevant data about the action (e.g., old vs. new values for updates, specific preferences granted, parameters of a report).
    - **Source Module:** The module that generated the log (e.g., "ConsentStorage", "PolicyDefinition", "UserRoleManagement").
    - **IP Address:** IP address from which the action was initiated (if user-driven).
    - **Status:** Success/Failure of the action.
- Exceptions Review:
    - **Exception ID:** Unique identifier for each exception.
    - **Timestamp:** When the exception occurred.
    - **Severity:** Critical, High, Medium, Low.
    - **Type:** Category of exception (e.g., "DataBreachWarning", "PolicyConflict", "DataValidationFailure", "PurgeExecutionError", "APIAuthFailure").
    - **Source Module:** Where the exception was detected.
    - **Details/Message:** Specific error message or contextual information.

    - **Affected Entities:** References to user_id, policy_id, data_processor_id if relevant.
    - **Resolution Status:** (e.g., "New," "Acknowledged," "Investigating," "Resolved," "Closed").
    - **Assigned To:** User assigned to resolve the exception.
    - **Resolution Notes:** Text field for documenting investigation and resolution steps.

Features/Capabilities:

- **Centralized Audit Log View:** Display all audit records in a chronological, paginated, and sortable table.
- Comprehensive Filtering & Search:
    - Filter by date range, user/system ID, action type, entity type, source module.
    - Free-text search across Context/Details.
- **Exception Dashboard:** Dedicated view for active exceptions, sortable by severity, status, and assignment.
- Exception Management Workflow:
    - Ability to change Resolution Status (e.g., New -> Acknowledged -> Resolved).
    - Assign exceptions to specific users for investigation.
    - Add Resolution Notes.
- **Export Functionality:** Export filtered audit logs and exception reports (e.g., to CSV, PDF) for external audits.
- **Alerting Integration:** Link exceptions to notifications for DPO/Admin via email/in-app alerts (from Notification Service).
- **Immutability Enforcement:** Prevent modification or deletion of historical audit log entries.
- **Retention Policies:** (Optional) Link audit logs to their own data retention policies for archival/deletion.

Functional Requirements:

- **Immutable Storage:** Audit logs must be stored in a write-once, read-many fashion (or functionally equivalent) to ensure their integrity. Use a secure database schema that prevents tampering.
- **Standardized Logging Format:** Ensure all modules log events in a consistent, structured format.
- **High Performance Querying:** Optimize database indexing for rapid filtering and searching of large volumes of audit data.

- **Access Control:** Strict RBAC (via "User Role Management") to ensure only authorized personnel (e.g., Auditor, Admin) can view and manage audit logs/exceptions.
- **Data Integrity:** Prevent accidental or malicious modification of log entries.
- **Real-time Logging:** Events must be logged as they occur.
- **Automated Exception Capture:** System must automatically detect and log exceptions from all integrated modules.
- **Correlation:** Ability to link related audit log entries (e.g., showing a user's login, then policy change, then logout).

Workflow:

1. **Auditor Login:** Auditor/Admin authenticates securely and accesses the Audit Dashboard.
2. **Overview & Alerts:** Review overall audit activity and high-priority exceptions.
3. Investigate Specific Event:
    - Auditor filters the Audit Log by Action Type ("DataPurgeExecuted") and a specific date range.
    - They click on a specific log entry to view its Context/Details (e.g., which data was purged, the retention policy applied).
4. Manage Exception:
    - Auditor navigates to "Exceptions Review".
    - They filter for "New" or "Critical" exceptions.
    - Select an exception, change its status to "Investigating", assign it to a team member, and add initial notes.
    - Upon resolution, update status to "Resolved" and add final notes.
5. **Generate Report:** Auditor selects filters for a compliance report (e.g., all "ConsentGranted" actions for a specific Fiduciary in the last quarter) and exports the data.

2.12. Notification System

This module is responsible for generating, delivering, and managing personalized, timely, and secure notifications to various stakeholders based on system events, compliance triggers, and user-driven actions.

Notifications are used for the following purposes

- To inform Data Principals about critical privacy events related to their data (e.g., policy updates, breach notifications, request status changes).
- To alert Data Fiduciary personnel (DPOs, Admins) about compliance obligations, privacy requests, grievances, and system health.
- To communicate operational instructions and alerts to Data Processors.
- To provide an auditable record of notifications sent.

Key Information Managed (Notification Template & Instance):

- **Notification Template:**
  - template_id: Unique ID.
  - name: (e.g., "Privacy Policy Update", "Data Breach Alert").
  - category: (e.g., "Compliance", "Security", "Grievance", "Operational").
  - severity: (e.g., "CRITICAL", "HIGH", "MEDIUM", "INFO").
  - channels_enabled: (e.g., ["EMAIL", "SMS", "IN_APP"]).
  - content_template: Multilingual text with placeholders (e.g., "en": "Policy {policy_version} effective from {effective_date}").
  - action_link_template: URL template for clickable actions (e.g., "https://{fiduciary_domain}/privacy-policy/{policy_version}").
- **Notification Instance (Record of Sent Notification):**
  - notification_id: Unique ID for each sent notification.
  - template_id: FK to the template used.
  - recipient_type: (DATA_PRINCIPAL, DPO_ADMIN, DATA_PROCESSOR).
  - recipient_id: User ID, Fiduciary ID, Processor ID, or specific email/phone.
  - fiduciary_id: FK to associated Fiduciary (if relevant to a specific Fiduciary).
  - status: (SENT, FAILED, DELIVERED, READ).
  - channel_used: (EMAIL, SMS, IN_APP).
  - sent_at: Timestamp.
  - payload_data: JSON of actual data used to populate template (e.g., { "policy_version": "1.1", "effective_date": "2025-06-01" }).
  - error_details: (if FAILED).

Features/Capabilities:

- **Template Management:**
  - Create, edit, and version multilingual notification templates with placeholders.
  - Define default severity and channels_enabled for each template type.
- **Event-Driven Triggering:**
  - Receive triggers from other CMS modules (e.g., "Consent Policy Published," "Data Breach Detected," "Grievance Status Changed").
  - Map triggers to specific notification templates.
- **Recipient Resolution:** Identify the correct recipients based on the event context (e.g., all Data Principals, specific DPO, specific Data Processor).
- **Content Generation:** Dynamically populate template placeholders with actual data from the event payload.
- **Multi-Channel Delivery:**
  - **In-App Notifications:** Display alerts/messages directly within the respective dashboards (User, DPO, Processor).
  - **Email Gateway Integration:** Send emails via a configured email service.
  - **SMS Gateway Integration:** Send SMS via a configured SMS service.
- **Delivery Status Tracking:** Monitor whether notifications were successfully sent/delivered.
- **Notification History:** Store records of all sent notifications for auditing and recipient review.
- **Preference Management (Recipient Side):** (Optional, for Data Principals) Allow Data Principals to manage their notification preferences (e.g., opt-out of marketing emails, choose SMS vs. email).
- **Retry Mechanism:** For failed delivery attempts.

Functional Requirements:

- **Timeliness:** Ensure notifications are sent promptly for critical events.
- **Accuracy:** Notifications must correctly reflect the event and populate data accurately.
- **Security:**

    - Sensitive data within notifications (if any) must be handled securely (e.g., minimal exposure, no PII in subject lines if possible).
    - Integration with email/SMS gateways must be secure.
- **Scalability:** Handle high volumes of notifications, especially for breach notifications to many Data Principals.
- **Auditability:** Every sent notification instance is logged with details of recipient, content, and status.
- **Multilingual Support:** Render notifications in the recipient's preferred language.
- **Graceful Degradation:** If one channel fails, attempt delivery via others (if configured).

Workflow:

1. **Event Trigger:** A CMS module (e.g., Policy Service on policy:publish, Audit Log Service on breach:detected, Grievance Service on grievance:status_update) publishes an event to the Notification System (e.g., via internal message queue or API call).
2. **Event Processing:** The Notification System's core logic receives the event.
3. **Template Selection:** It identifies the appropriate notification template based on the event type and severity.
4. **Recipient Resolution:** It determines the recipient(s) (e.g., all Data Principals associated with fiduciary_id, the assigned_dpo_user_id, the data_processor_id).
5. **Content Personalization:** It populates the template's placeholders with data from the event payload and recipient's preferences (e.g., recipient's name, localized text).
6. **Channel Delivery:** It sends the personalized message through configured channels (in-app, email, SMS) using integrated gateways.
7. **Status Tracking & Logging:** It updates the notification_instance record with status (SENT, DELIVERED), and logs the event to the audit_logs table.
8. **Recipient Interaction:**
    - In-app: Displayed on the respective dashboard.
    - Email/SMS: Received by recipient.

2.13. System Administration

2.13.1. User Role Management

This service manages Role-Based Access Control (RBAC), enabling secure and auditable access control within the CMS.

createRole(roleName: string, permissions: string[]): Role

- **Purpose:** Defines a new custom role with a set of specific permissions.
- **Input:** roleName (e.g., "CustomEditor"), permissions (array of action strings like "cms:content:edit", "user:read").
- **Output:** The created Role object (including its generated id).
- **Behavior:** Persists role definition. Logs creation.

updateRole(roleId: string, newPermissions: string[]): Role

- **Purpose:** Modifies the permissions associated with an existing role.
- **Input:** roleId, newPermissions.
- **Output:** The updated Role object.
- **Behavior:** Updates role permissions. Logs modification.

deleteRole(roleId: string): boolean

- **Purpose:** Removes a role. (Requires careful handling of assigned users).
- **Input:** roleId.
- **Output:** true if successful, false otherwise.
- **Behavior:** Disassociates role from users before deletion. Logs deletion.

assignUserRole(userId: string, roleId: string): boolean

- **Purpose:** Assigns an existing role to a specific user.
- **Input:** userId, roleId.
- **Output:** true if successful.
- **Behavior:** Persists user-role mapping. Logs assignment.

revokeUserRole(userId: string, roleId: string): boolean

- **Purpose:** Removes a specific role from a user.

- **Input:** userId, roleId.
- **Output:** true if successful.
- **Behavior:** Removes user-role mapping. Logs revocation.

checkUserPermission(userId: string, requiredPermission: string): boolean

- **Purpose:** Validates if a user has the necessary permission to perform an action.
- **Input:** userId, requiredPermission (e.g., "cms:settings:update").
- **Output:** true if the user's assigned roles grant the permission, false otherwise.
- **Behavior:** Efficiently queries user's aggregated permissions.

getAuditLog(filter: {entityType?: 'role'|'userRole', entityId?: string, startDate?: Date, endDate?: Date}): AuditRecord[]

- **Purpose:** Retrieves a historical record of role definitions, assignments, and modifications.
- **Input:** Optional filters for specific entities or time ranges.
- **Output:** An array of AuditRecord objects detailing changes.

2.13.2. Data Retention Policy Configuration

This module enables Data Fiduciaries to define, manage, and apply rules for the storage limitation and eventual deletion of personal data, aligning with legal requirements and data minimization principles.

The module enables the following:

- To centrally define data retention periods based on purpose, data category, and legal/regulatory obligations.
- To ensure automatic or semi-automatic deletion/anonymization of personal data once its purpose is fulfilled or retention period expires.
- To demonstrate compliance with storage limitation principles (e.g., DPDP Act).

Key Information Captured (Retention Policy Definition):

- **Policy ID:** Unique identifier (system-generated/user-defined).
- **Policy Name:** Human-readable name (e.g., "KYC Data Retention", "Analytics Data Retention").
- **Applicability Scope:**
    - **Data Fiduciary:** Which Fiduciary this policy applies to (if CMS manages multiple).
    - **Purpose(s):** Linked to specific data processing purposes (e.g., "Identity Verification", "Marketing").
    - **Data Categories:** Linked to specific personal data categories (e.g., "Browse History", "Email").
- **Retention Period:**
    - **Duration:** Numeric value (e.g., 5).
    - **Unit:** Time unit (e.g., "Days", "Months", "Years").
    - **Start Event:** Event that triggers the retention clock (e.g., "Consent Given", "Service Terminated", "Last Activity Date").
- **Action at Expiry:**
    - "Delete" (Permanently delete data).
    - "Anonymize" (Transform data to render it non-identifiable).
    - "Archive" (Move to long-term, restricted access storage, while still adhering to retention rules for the *live* copy).
- **Legal/Regulatory Reference:** (Optional) Link to specific legal sections justifying the period (e.g., "DPDP Act, Section X", "RBI KYC Norms").
- **Status:** Active/Inactive.
- **Creation/Last Updated Timestamps & User.**

Features/Capabilities:

- **Policy Creation/Editing:** Define and modify retention policies.
- **Policy Assignment:** Link policies to specific data processing purposes and categories.
- **Override Management:** (Optional) Allow more specific policies to override general ones for specific data types or contexts.
- **Policy Deactivation:** Suspend a policy without deleting its definition.
- **Compliance Review:** View a summary of active policies and their coverage.

Functional Requirements:

- **Rule Engine Integration:** The CMS must have an internal rule engine that applies configured policies to actual data records.
- **Automated Triggering:** The system must automatically identify and flag data records that have met their retention period based on defined policies.
- **Execution Mechanism:**
    - For "Delete"/"Anonymize": Initiate secure deletion/anonymization processes.
    - For "Archive": Initiate data transfer to designated archives.
- **Confirmation/Approval Workflow:** For sensitive deletions, require DPO/Admin confirmation before execution.
- **Audit Trail:** Log all policy definitions, changes, and every instance of data deletion/anonymization/archiving executed under a policy.
- **Conflict Resolution:** Define how the system resolves conflicts if multiple policies apply to the same data (e.g., longest retention period takes precedence).
- **Reporting:** Generate reports on data scheduled for deletion/anonymization and records of past operations.

Workflow:

Admin Defines Policy: CMS Admin/DPO navigates to "Data Retention Policies" and creates a new policy, specifying its name, applicable purposes/categories, retention duration, start event, and expiry action.

Policy Activation: Admin activates the policy.

System Monitoring: The CMS backend continuously monitors data records.

Expiry Detection: When a record's retention period (calculated based on the policy and its "Start Event") expires, the system flags it.

Action Execution: The system automatically (or after DPO approval) executes the defined action (delete) for the flagged data.

Audit Logging: All actions are logged, including which policy triggered them.

2.13.3. API Keys Management (for DF & DP)

This module provides a secure and auditable mechanism for CMS administrators to generate, manage, and revoke API keys, granting Data

Fiduciaries and Data Processors controlled access to the DPDP CMS APIs. The module's purpose is to

- To enable secure, authenticated, and controlled programmatic access for Data Fiduciaries' and Data Processors' systems to the DPDP Solution's APIs.
- To allow granular control over API key permissions, linking specific keys to specific Fiduciaries/Processors and their allowed API operations.
- To provide a mechanism for key rotation, revocation, and auditing for enhanced security.
- To ensure that API access aligns with defined DPA (Data Processing Agreement) terms.

Key Information Stored (API Key Record):

- **id**: UUID, Primary Key. Unique identifier for the API key.
- **key_value**: VARCHAR(255), Not Null, UNIQUE (hashed or encrypted for security). The actual API key string.
- **fiduciary_id**: UUID, Not Null. FK to fiduciaries.id. The Data Fiduciary this key belongs to.
- **processor_id**: UUID, Nullable. FK to processors.id. (If key is for a specific Processor).
- **owner_type**: VARCHAR(50), Not Null (e.g., FIDUCIARY_APP, PROCESSOR_INTEGRATION, CMS_ADMIN_TOOL).
- **description**: TEXT. Human-readable description of the key's purpose (e.g., "TSI Coop Website Frontend Consent Submission Key", "AnalyticsPro Purge Confirmation Key").
- **status**: VARCHAR(50), Not Null (ACTIVE, INACTIVE, REVOKED, EXPIRED).
- **permissions**: JSONB, Not Null. Array of API permissions granted (e.g., ["consent:write", "policy:read", "purge:confirm"]).
- **created_at**: TIMESTAMP, Not Null.
- **created_by_user_id**: UUID, FK to users.id. CMS admin who created the key.
- **expires_at**: TIMESTAMP, Nullable. Optional expiry date for the key.
- **last_used_at**: TIMESTAMP, Nullable. Timestamp of the last successful API call using this key.
- **revoked_at**: TIMESTAMP, Nullable. Timestamp if key was revoked.
- **revoked_by_user_id**: UUID, FK to users.id. User who revoked the key.

Features/Capabilities:

- **API Key Generation:**
  - **Secure Generation:** The system must generate cryptographically strong, random API keys (e.g., UUID-based or high-entropy random strings).
  - **Association:** Keys are always linked to a specific fiduciary_id and optionally a processor_id.
  - **Permission Scoping:** CMS administrators can define the exact API permissions (e.g., consent:write, policy:read, grievance:submit) granted to each key.
- **API Key Listing & Search:**
  - Provide a dashboard view for CMS administrators to list all generated keys.
  - Filter by fiduciary_id, processor_id, status, owner_type.
- **API Key Status Management:**
  - **Activation/Deactivation:** Change status between ACTIVE and INACTIVE.
  - **Revocation:** Instantly invalidate an API key (set status to REVOKED, record revoked_at and revoked_by_user_id).
  - **Expiry:** Set expires_at for automatic key invalidation.
- **Key Rotation:** Provide a feature to "rotate" a key by generating a new one and deactivating the old one, simplifying periodic key updates.
- **Usage Monitoring:** Track last_used_at for each key. (Advanced: Monitor call volume, error rates).
- **Audit Trail:** All key management actions (generation, update, status change, revocation) are logged in the Audit Log Service.

Functional Requirements:

- **Security of Key Value:** key_value must be stored securely (hashed or encrypted) and never displayed in plain text in the UI after generation. It should only be displayed *once* upon creation.
- **Unique Keys:** Enforce uniqueness for key_value.
- **Referential Integrity:** Ensure fiduciary_id and processor_id link to valid existing records.
- **RBAC Enforcement:** Only users with admin:keys:manage or fiduciary:keys:manage (if Fiduciaries get limited self-service) can perform key management operations.

- **API Gateway Integration:** The API Gateway must be able to validate incoming X-API-KEY headers against the key_value, status, and permissions stored in this module.
- **Performance:** API key validation must be fast, as it's performed on every incoming API call.
- **Audit Logging:** Every action on an API key (creation, status change, revocation, last usage) must be logged in audit_logs.

Workflow:

Admin Navigates: CMS Admin logs in and goes to "API Keys Management."

Initiate Generation: Admin clicks "Generate New API Key."

Configure Key: Admin selects:

fiduciary_id (e.g., "TSI Coop").

owner_type ("FIDUCIARY_APP").

description ("TSI Coop Website Frontend Consent Submission Key").

permissions (e.g., consent:write, policy:read).

Expires_at.

Key Generation & Display: System generates a unique id and key_value. It displays the key_value *once* to the Admin.

Store Record: The id, hashed_key_value, and all other details are stored in the database.

Audit Log: audit_logs record API_KEY_CREATED.

Key Provisioning: Admin provides the key_value to the Data Fiduciary's development team, who configure it in their frontend application.

API Usage: When the Fiduciary's frontend calls /api/v1/consent, it includes the X-API-KEY header. The API Gateway validates this key (status, permissions) against the stored record before routing the request.

2.13.4. Deactivation & Reactivation

This module provides authorized administrators with the capability to immediately suspend (deactivate) or restore (reactivate) the operational status and system access for Data Fiduciaries, Data Processors, and CMS User accounts. This is critical for security, compliance, and managing relationships. The module's purpose is to

- To provide granular control over the operational status and access privileges of key entities within the DPDP Solution.
- To enable rapid response to security incidents (e.g., suspected compromise, unauthorized activities).
- To manage the lifecycle of contractual relationships (e.g., Fiduciary/Processor service termination).
- To maintain a clear, auditable record of all status changes for compliance and operational transparency.

Key Information Involved (Common Across Entities):

- **Entity Type:** (e.g., FIDUCIARY, PROCESSOR, CMS_USER).
- **Entity ID:** The unique identifier of the specific entity (e.g., fiduciaries.id, processors.id, users.id).
- **status field:** A common status field in the respective database tables (e.g., fiduciaries.status, processors.status, users.status).
    - **Values:** ACTIVE (fully operational), INACTIVE (suspended/deactivated), REVOKED (permanently invalidated/terminated), PENDING (initial state).
- **deactivated_at**: TIMESTAMP, Nullable. Timestamp when deactivated.
- **deactivated_by_user_id**: UUID, FK to users.id. CMS user who deactivated.
- **reactivated_at**: TIMESTAMP, Nullable. Timestamp when reactivated.
- **reactivated_by_user_id**: UUID, FK to users.id. CMS user who reactivated.
- **reason_for_status_change**: TEXT. Mandatory explanation for deactivation/reactivation.

Features/Capabilities:

- **Entity Selection:** Administrators can select a specific Data Fiduciary, Data Processor, or CMS User account for status modification.
- **Deactivate Function:**
    - Changes the status of the selected entity to INACTIVE.
    - Requires a mandatory reason_for_status_change.
    - Records deactivated_at and deactivated_by_user_id.
    - **Immediate Enforcement:**
        - **For Fiduciaries:** Immediately suspends all active policies, prevents consent collection/validation, and blocks API access tied to that Fiduciary.
        - **For Processors:** Immediately suspends all processing instructions, prevents purge confirmations, and blocks API access tied to that Processor.
        - **For CMS Users:** Immediately revokes login access to all CMS dashboards and APIs.
- **Reactivate Function:**
    - Changes the status of the selected entity back to ACTIVE.
    - Requires a mandatory reason_for_status_change.
    - Records reactivated_at and reactivated_by_user_id.
    - **Immediate Enforcement:** Restores operational status and API/login access.
- **Revoke/Terminate Function:**
    - Changes the status to REVOKED (a permanent state, typically not reactivatable).
    - Records revoked_at and revoked_by_user_id.
    - **Impact:** Triggers data archival/deletion procedures (via Data Retention/Purge Service) for associated data where permissible (e.g., if a Fiduciary's contract is fully terminated).
- **Status Display:** Clearly indicates the current status of each entity in their respective dashboards (e.g., Data Fiduciary Dashboard, Data Processor Dashboard).
- **Audit Trail:** All status changes are meticulously logged in the Audit Log Service.

Functional Requirements:

- **Granular Control:** Provide separate actions/buttons for Deactivate, Reactivate, and Revoke/Terminate.
- **Confirmation Prompts:** Require confirmation for Deactivate and Revoke/Terminate actions due to their immediate impact.

- **Mandatory Reasons:** Enforce capturing a reason for every status change.
- **RBAC Enforcement:** Only CMS users with appropriate roles (e.g., admin:entity:manage_status, dpo:fiduciary:manage_status) can perform these actions.
- **Atomic Operation:** Status changes in the database must be atomic transactions to ensure consistency.
- **Real-time Enforcement:** The system's authorization and routing layers (e.g., API Gateway, core microservices) must immediately check the status field for every request or operation involving the entity.
- **Notification:** Notify affected parties (e.g., the Fiduciary if their status is changed, the CMS user if their account is deactivated) via the Notification System.

Workflow (Illustrative: Deactivating a Fiduciary):

Admin Access: CMS Admin navigates to "Manage Data Fiduciaries."

Select Fiduciary: Admin selects "Fiduciary X" and clicks "Deactivate."

Confirmation & Reason: A modal pops up asking for confirmation and a mandatory reason_for_status_change (e.g., "Contract Dispute," "Security Alert").

Backend Update: System updates fiduciaries.status to INACTIVE, records deactivated_at and deactivated_by_user_id.

Immediate Enforcement: The API Gateway immediately blocks all incoming API requests from "Fiduciary X" (by checking fiduciaries.status). The Fiduciary's website/app will lose CMS functionality.

Notification: Notification System sends an alert to the DPO of "Fiduciary X" and relevant CMS Admins about the deactivation.

Audit Log: audit_logs records FIDUCIARY_DEACTIVATED with the reason_for_status_change.

3. CMS Client (Integrated by DF & DP)

3.1. SDK Integration

3.1.1.1. Policy Retrieval

This module is responsible for providing authorized consumers (primarily Data Fiduciaries' websites or applications) with the most current, active, and appropriately localized Personal Data Consent Policy. The module's purpose is to

- To reliably serve the legally effective and up-to-date Personal Data Consent Policy content from the CMS backend.
- To ensure that Data Principals are presented with the precise policy they are consenting to, matching the version logged with their consent.
- To support dynamic rendering of consent notices and preference centers by consuming frontend applications.
- To provide policy content in the Data Principal's preferred language and relevant to their jurisdiction.

Key Information Retrieved (Policy Data):

- The full JSON structure of the policy_content as defined in the "Personal Data Consent Management - Policy Definition" module (including policy_id, version, effective_date, data_fiduciary_info, languages object with all localized purposes, categories, buttons, and links).

Features/Capabilities:

- **Active Policy Lookup:** Ability to query the CMS backend for the currently Active policy version for a given Data Fiduciary and jurisdiction.
- **Language-Specific Delivery:** Can deliver the entire multilingual policy or filter to deliver only the content for a specific requested language.
- **Jurisdiction-Aware Selection:** Selects the appropriate policy based on the Data Fiduciary and potentially the user's detected jurisdiction (if multiple distinct policies are maintained per Fiduciary for different regions).
- **Version History Retrieval:** (For auditing) Ability to retrieve any past policy version using policy_id and version.

- **HTTP Caching Headers:** Support for ETag, Last-Modified, Cache-Control headers to optimize client-side caching.

Functional Requirements:

- **Dedicated API Endpoint:** The CMS backend must expose a robust, read-only API endpoint for policy retrieval (e.g., GET /api/personal-data-policies/active).
- **Active Status Enforcement:** The API *must* only return policies with a status of ACTIVE and an effective_date in the past or current.
- **Fiduciary & Jurisdiction Filtering:** The API call must include parameters for fiduciary_id and jurisdiction to ensure the correct policy is served (e.g., .../active?fiduciaryId=xyz&jurisdiction=IN).
- **Language Parameter:** Support for an optional lang query parameter (e.g., ?lang=ta). If provided, the API can return only the content for that language; otherwise, it returns the full languages object.
- **Performance Optimization:** The API must be highly performant, as it's typically called on every page load (or upon cache miss). Database indexing on fiduciary_id, status, effective_date, and jurisdiction is crucial.
- **Error Handling:** Graceful handling and informative error responses for scenarios like "Policy Not Found," "Invalid Fiduciary ID," or "No Active Policy."
- **Access Control:** This endpoint may be publicly accessible (e.g., via API key) as it serves the public-facing policy text, but sensitive configuration details should remain internal.

Workflow:

Frontend/Client Request: A Data Fiduciary's website or application (frontend) initiates an HTTP GET request to the CMS backend's policy retrieval API.

    - The request includes parameters: fiduciaryId=Fid_XYZ, lang=en).

Backend Policy Selection:

    - The Backend Service queries the consent_policies database table.

    - It searches for an entry matching fiduciary_id, jurisdiction, with status='ACTIVE' and effective_date <= current_timestamp.
    - If multiple active policies exist (e.g., for different regions), the system selects the most appropriate one based on predefined rules or the jurisdiction parameter.

Content Filtering & Response:

    - The Backend Service retrieves the full policy_content JSON.
    - If a lang parameter was provided, it extracts only the relevant language object (e.g., policy.languages.en) from the policy_content.
    - It constructs an HTTP response containing the retrieved (and potentially filtered) policy JSON.
    - Appropriate HTTP caching headers are included in the response.

Frontend/Client Consumption:

    - The frontend receives the policy JSON.
    - It caches the policy client-side (e.g., in localStorage) along with its version.
    - It uses the policy content for dynamic rendering of the consent banner and preference center.

3.1.1.2. Consent Form Rendering

This module is responsible for dynamically generating and presenting interactive consent forms to Data Principals, allowing them to explicitly grant or deny consent for various personal data processing purposes as defined in the active policy. The purpose is to

- To translate the structured Personal Data Consent Policy (JSON) into a user-friendly and interactive form or preference center.
- To enable Data Principals to review data processing purposes and granularly provide or withdraw consent.
- To ensure the consent interface is dynamic, localized, and compliant with DPDP Act requirements (e.g., clear, specific, informed, unambiguous consent).

Key Information Used (from Policy JSON - data_processing_purposes array):

- policy_id & version: For context and linkage.
- languages.<lang>.title, introduction, general_purpose_description: For overall form context.
- **languages.<lang>.data_processing_purposes (array of objects):**
    - id: Unique identifier for the purpose.
    - name: Localized display name.
    - description: Localized detailed explanation.
    - legal_basis: Legal basis for transparency.
    - is_mandatory_for_service: Boolean to indicate if opting out breaks core service.
    - is_sensitive: Boolean to flag if purpose involves sensitive data.
- languages.<lang>.data_categories_details: For providing descriptive tooltips or inline explanations of data categories.
- languages.<lang>.buttons, links: For localized calls-to-action and policy references.

Features/Capabilities:

- **Dynamic Form Generation:** Automatically renders consent forms or preference centers based on the JSON policy data.
- **Multilingual Display:** Presents all text (purpose names, descriptions, button labels) in the Data Principal's selected language.
- **Granular Consent Toggles:** Provides interactive elements (e.g., checkboxes, toggle switches) for each non-mandatory data processing purpose.
    - **Disabled/Pre-ticked Options:** For purposes marked is_mandatory_for_service: true, the corresponding consent option is pre-selected and disabled, with a clear explanation.
- **Contextual Information:** Display detailed descriptions, data categories involved, third-party recipients, and legal basis for each purpose (e.g., via expandable sections, tooltips).
- **Initial State Pre-population:** Load and display the user's previously saved consent preferences (if any) to pre-select toggles.
- **User Action Buttons:** "Accept All", "Reject All Non-Essential", "Save Choices".

- **Responsive UI:** Adapts seamlessly to different screen sizes (desktop, tablet, mobile).
- **Accessibility:** Adheres to WCAG guidelines for usability by diverse users.

Functional Requirements:

- **Policy-Driven Rendering:** The rendering logic must strictly follow the structure and content of the retrieved policy JSON. Any change in the JSON policy should dynamically reflect in the UI without code changes.
- **State Management:** Maintain the current selection state of all consent toggles in real-time as the user interacts.
- **Input Validation (Client-side):** (Minimal, as consent is about choice, not data input validity). Ensure all mandatory purposes are acknowledged if the user attempts to "Save."
- **Event Handling:** Capture user interactions (clicks, toggle changes) and prepare a structured consent payload for submission.
- **Performance:** Render the form efficiently to ensure a smooth user experience.
- **Fallback Rendering:** Provide a basic, compliant fallback if the policy JSON is malformed or retrieval fails.

Workflow:

Policy Retrieval: The "Policy Retrieval" module successfully fetches the active and localized Personal Data Consent Policy JSON.

Current Consent Retrieval: The user's previously saved consent preferences are retrieved from client-side storage.

Dynamic Rendering: The Consent Form Rendering module processes the policy JSON and user's current preferences.

- It iterates through data_processing_purposes.
- For each purpose, it creates a UI element (e.g., a card with a toggle).
- It populates the element with name, description, legal_basis, and lists data_categories_involved, recipients_or_third_parties.

    - If is_mandatory_for_service is true, the toggle is enabled and disabled. Otherwise, its initial state is set based on previous user choice or a default (opt-out by default for non-essential).

  Event Listeners: Attach event listeners to the generated toggles and buttons.

  User Interaction: Data Principal interacts with the form, adjusting preferences.

  Payload Preparation: Upon "Save Choices," the module compiles a JSON payload reflecting the user's final consent for each data_processing_purpose.

  Submission: The payload is sent to the "Consent Storage" module via API.

Tooling & SDK Suggestions:

To implement this functional design efficiently, leveraging existing declarative UI frameworks and data-driven form generators is highly recommended:

- **For Declarative UI/Web Frameworks:**
    - **ReactJS / Angular / Vue.js:** These modern JavaScript frameworks are excellent for building complex, dynamic, and responsive user interfaces. You would use their component-based architecture to create reusable UI components for purpose toggles, category descriptions, etc., and manage the form state.
- **For JSON-Schema-driven Forms (reduces manual UI coding):**
    - **react-jsonschema-form (for React):** Allows you to define your policy's input structure using JSON Schema and it generates the form UI automatically. You would then customize the widgets (e.g., using custom uiSchema or widgets) to get the specific toggle-based consent experience. This significantly reduces boilerplate UI code.
    - **jsonform.io (Vanilla JS / jQuery based):** A flexible JavaScript library that generates HTML forms from JSON Schema. Good for simpler integrations or if you prefer less framework dependency.

    - **@rjsf/core (React JSON Schema Form):** The latest version of react-jsonschema-form, offering greater flexibility.
- **For Cross-Platform Mobile/Web (if applicable):**
    - **Flutter:** A UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase. Ideal if your consent forms need to appear consistently across web and mobile apps. You'd likely parse the JSON policy and build custom Flutter widgets.
    - **React Native:** For mobile apps, leveraging React's principles.
- **Custom JavaScript SDK:** If a light-weight, framework-agnostic solution is preferred, a custom JavaScript SDK can be developed. This SDK would:
    - Fetch the policy JSON.
    - Dynamically create DOM elements using document.createElement().
    - Attach event listeners.
    - Manage the consent state in memory.
    - Provide methods for init(), showConsentForm(), hideConsentForm(), and getConsentPayload().

3.1.1.3. Link Data Principal

The Link Principal feature enables the CMS to **seamlessly associate anonymous consent records with a definitive, authenticated Data Principal ID** once a user logs in or registers. This function ensures that any privacy choices made during an unauthenticated session (identified by a client-side temporary ID) are correctly linked to the user's permanent, server-generated ID. Upon successful linking, the system updates all relevant consent records in the backend, linking them to the authenticated Data Principal, thereby maintaining a complete and accurate consent history for auditability and compliance, regardless of the user's initial interaction state.

Workflow:

Anonymous Interaction: A Data Principal visits the website, makes consent choices (e.g., accepts cookies). A `temporary_anonymous_id`

(e.g., `anon_XYZ`) is generated client-side and stored in `localStorage` along with consent preferences.

User Authentication: The Data Principal proceeds to register or log in to the Data Fiduciary's system.

Authenticated ID Available: Upon successful authentication, the Data Fiduciary's core system (backend) provides a definitive `authenticated_data_principal_id` (e.g., `user_ABC`). This ID is made available to the client-side JavaScript.

Client-Side Trigger: The client-side JavaScript detects both the presence of `temporary_anonymous_id` in `localStorage` AND the newly available `authenticated_data_principal_id`. It compares them to ensure they are different.

API Call: The client-side JavaScript makes a `POST` request to the DPDP Solution's `Link Principal` API endpoint, sending both the `temporary_anonymous_id` and the `authenticated_data_principal_id`.

Backend Processing (CMS):

- The API receives the request and verifies its authorization.
- It initiates a database transaction.
- It updates all records in the `consent_records` table (and potentially other related tables like `grievances`) where `user_id` matches the `temporary_anonymous_id`, setting `user_id = authenticated_data_principal_id`.
- The transaction is committed.
- An entry is made in the `audit_logs` table, documenting the linking event (who, what, when, anonymous ID, authenticated ID).

Unified Experience: From this point forward, all consent management and history for this user are accessed and managed using the `authenticated_data_principal_id`.

3.2. Consent Gated Features - Making Privacy Policy Actionable

This functional design describes how a Privacy Policy, typically a static, informative document, is transformed into a dynamic and interactive mechanism for **explicit consent capture** at critical points of user interaction. The core idea is to ensure that consent is not merely acknowledged, but actively provided *in context*, directly linked to specific data processing activities, and enforced by system functionality.

1. Contextual Enforcement Points:

- **Identification of Gateways:** The system identifies specific points within the user journey where access to certain functionalities, features, or content (e.g., "Provider Zone," "Solution Submission," "Service Listing") requires processing personal data beyond what the user has *currently* consented to. These are the "actionable moments."
- **Purpose-Based Activation:** Each restricted functionality is pre-mapped to the precise data processing purpose(s) (e.g., "Solutions & Services Showcase," "Community Engagement") that require consent as defined in the active Privacy Policy.

2. Dynamic Policy Presentation & Consent Capture:

- **Pre-Access Prompt:** When a user attempts to access a restricted functionality for which their current consent is insufficient, the system *intercepts* the access request.
- **In-Context Policy Display:** Instead of just blocking access, the system dynamically presents the relevant section of the Privacy Policy (or the full policy, with the relevant section highlighted) directly within the application's flow. This ensures the user is *informed* at the exact moment their data is needed for a specific action.
- **Interactive Consent Form:** Immediately following or integrated with the policy display, the system renders an interactive consent form (e.g., a modal preference center). This form presents the specific data processing purpose(s) required for the desired functionality, along with their associated data categories, legal bases, and recipients.
- **Granular Choice:** The user is given clear, granular options to "Accept" or "Decline" consent for the specific purpose(s) required to proceed.

3. Real-time Consent Enforcement & Access Control:

- **Conditional Access:** User access to the restricted functionality is granted *only if* explicit consent for the required data processing purpose(s) is affirmatively provided.
- **Consent Record Update:** Upon the user's choice, the system captures and stores the new consent decision in the consent_records database, linked to the active Privacy Policy version and the specific purposes chosen.
- **Immediate Activation/Deactivation:** Based on the new consent record, the system immediately activates the relevant data processing pipelines or third-party integrations needed for the functionality. Conversely, if consent is declined, access remains blocked, or the functionality is disabled.
- **Audit Trail:** Every instance of this contextual consent capture, the user's decision, and the resulting access grant/denial is logged in the audit_logs for compliance verification.

3.3. Service Adapter for automated data purge handling

This module enables the DPDP CMS to trigger and confirm the compliant deletion or anonymization of personal data within both Data Fiduciary and Data Processor systems, driven by consent withdrawal or retention policies. The module’s purpose is

- To initiate and manage the secure deletion or anonymization of personal data across distributed systems (Data Fiduciary's own databases and Data Processors' systems).
- To ensure data is purged in accordance with Data Principal rights (e.g., Right to Erasure) and defined Data Retention Policies (DPDP Act's storage limitation).
- To provide an auditable confirmation mechanism for successful data purges.
- To offer a standardized, easily pluggable API for Data Fiduciaries and Data Processors to integrate their purge capabilities.

Key Information Involved:

- **Purge Request:** user_id, fiduciary_id, data_categories_to_purge, processing_purposes_affected, trigger_event (e.g.,

"ConsentWithdrawal", "RetentionPolicyExpiry"), policy_reference (if applicable).
- **Purge Confirmation:** purge_request_id, status ("SUCCESS", "FAILED"), details (e.g., count of records purged, error messages), timestamp.

Functional Modules & Adaptations:

- **Data Retention/Purge Service (CMS Backend):** Orchestrates the overall purge process, generating purge requests and tracking their status.
- **Consent Record Service:** Notifies Data Retention/Purge Service upon consent withdrawal.
- **Data Fiduciary Dashboard (DPO):** Displays purge reports and exceptions.
- **Audit Log Service:** Logs all purge requests, confirmations, and failures.

Purge Handler API Interface (Provided by CMS to DF/DP):

The CMS will expose a dedicated API for Data Fiduciaries and Data Processors to implement and report on purge operations.

- **Endpoint:** POST /api/v1/purge-status
  - **Purpose:** The Data Fiduciary or Data Processor calls this endpoint to **acknowledge receipt of a purge instruction** and/or to **report the status/completion** of a purge operation.
  - **Request Body (from DF/DP confirming purge):** JSON

JSON

```
{
  "purge_request_id": "PRG-UUID-FROM-CMS",      #
The ID of the original request from CMS
  "fiduciary_id": "DF-UUID",                    #
ID of the Data Fiduciary
```

```
  "processor_id": "DP-UUID-OR-NULL",            #
ID of the Data Processor (if DP is calling)
  "status": "COMPLETED",                        #
COMPLETED, FAILED, IN_PROGRESS, NOT_FOUND
  "timestamp": "2025-05-30T10:30:00Z",          #
Time of status update
  "records_affected_count": 12345,              #
Number of records deleted/anonymized
  "details": "Data for user_id XXXXX purged as
per erasure request.", # Human-readable details
  "error_message": null                         #
Populated if status is FAILED
}
```

- **Authentication:** X-API-KEY or OAuth 2.0 token (issued by CMS to DF/DP).
- **Response:** 200 OK or 400 Bad Request if invalid payload.

Functional Workflow: Data Purge Initiation & Confirmation

Purge Trigger (CMS Internal):

Right to Erasure: Data Principal requests erasure via User Dashboard. "Consent Record Service" receives withdrawal, sets is_active_consent=FALSE for all purposes.

Retention Policy Expiry: "Data Retention/Purge Service" identifies data records whose retention period has expired based on configured policies.

Audit Log: The trigger event is logged in audit_logs (e.g., ERASURE_REQUEST_RECEIVED, RETENTION_PERIOD_EXPIRED).

Generate Purge Request (CMS - Data Retention/Purge Service):

The CMS creates an internal Purge Request record (e.g., purge_request_id, user_id, fiduciary_id, data_categories_to_purge, trigger_event).

Notify Data Fiduciary/Processor: The CMS (via Webhook call) sends an **instruction** to the relevant Data Fiduciary's *core systems* or the specific Data Processor's systems, outlining the data to be purged (e.g., user_id, fiduciary_id, affected data_categories, processing_purposes). This notification includes the purge_request_id.

Method: A Webhook from CMS to DF/DP.

Execute Purge (at Data Fiduciary/Processor):

Data Fiduciary (DF): Their internal systems (e.g., CRM, transactional DBs) receive the instruction via their implemented purge handler. They perform the deletion/anonymization for data they *directly* hold.

Data Processor (DP): Their systems receive the instruction from the Data Fiduciary (or directly from CMS if registered to do so). They execute the purge on data they process on behalf of the Fiduciary.

Security: Purge operations must be secure, irreversible, and logged internally by DF/DP.

Confirm Purge (DF/DP calls CMS API):

After executing the purge (or encountering a failure), the Data Fiduciary's or Data Processor's system calls the CMS's POST /api/v1/purge-status endpoint.

They provide the purge_request_id, status (COMPLETED/FAILED), records_affected_count, and details/error_message.

Update Purge Status & Audit (CMS - Data Retention/Purge Service):

The CMS receives the confirmation via the API.

It updates the internal Purge Request record with the status and details.

It logs the confirmation event in audit_logs (e.g., DATA_PURGE_CONFIRMED_SUCCESS, DATA_PURGE_FAILED).

If failed, it triggers an Exception notification to the DPO.

Report & Visibility (CMS - DPO Dashboard):

DPO Dashboard displays updated Data Purge Reports reflecting the execution status.

3.4. User Dashboard

This module provides Data Principals with a personalized, secure portal to transparently view and manage their personal data consent, track privacy requests, and receive important notifications from the Data Fiduciary.The purpose is

- To empower Data Principals to exercise their rights under the DPDP Act (e.g., Right to Access, Right to Correction, Right to Erasure, Right to Grievance Redressal, Right to Withdraw Consent).
- To provide a transparent overview of how their personal data is being processed.
- To facilitate direct communication and resolution tracking for privacy-related concerns.
- To deliver relevant privacy-related notifications to the user.

Key Information Displayed/Managed:

- **User Profile Information:** Basic authenticated user details (e.g., Name, Email, User ID).
- **Current Consent Status Summary:** High-level overview of active consent preferences (e.g., "Marketing: On", "Analytics: Off").
- Consent History Details:
    - List of all recorded consent transactions (Policy ID, Version, Timestamp, Mechanism, IP Address).
    - Ability to view granular details of each historical consent record (which purposes were consented/denied).

- **Pending Privacy Requests:** Status of open Data Principal Rights requests (e.g., Access Request, Correction Request, Erasure Request).
- Grievance Redressal & Resolution Tracking:
    - List of submitted grievances.
    - Current status of each grievance (e.g., "Submitted," "Under Review," "Resolution Pending," "Resolved," "Closed").
    - Timestamp of submission and last update.
    - (Optional) Communication log or secure messaging for the grievance.
- **User Notifications:** A feed or list of privacy-related alerts (e.g., "Privacy Policy Updated," "Data Breach Notification," "Consent Expiring").

Features/Capabilities:

- **Secure Authentication:** Requires Data Principal to authenticate (e.g., username/password, MFA) to access their dashboard.
- **View Consent History:** Display a chronological list of all consent records, allowing drill-down to view details of each record (purpose-by-purpose consent).
- Modify or Revoke Consent:
    - Provides a direct link/button to access the "Personal Data Consent Management - Consent Form Rendering" (Preference Center) module.
    - Enables Data Principals to update their current preferences or withdraw consent for specific purposes/all non-essential processing.
- Initiate Data Principal Rights Requests:
    - **Right to Access:** Button to request a summary of personal data held by the Data Fiduciary.
    - **Right to Correction/Completion:** Form to submit requests for data correction/completion.
    - **Right to Erasure:** Button to request erasure of personal data.
    - (Optional) Other rights as specified by DPDP Act (e.g., Right to Grievance Redressal - separate section).
- **Submit Grievance:** A dedicated form to submit a new privacy-related grievance or complaint.
- **Track Grievance Status:** A table or list showing all submitted grievances with their current status and progress updates.

- **Receive & View Notifications:** Displays a list of system-generated notifications relevant to the Data Principal's privacy.
- **(Optional) Secure Messaging:** A secure, authenticated channel for direct communication with the Data Fiduciary's DPO regarding privacy requests/grievances.

Functional Requirements:

- **Authentication & Authorization:** Strictly enforce user authentication. Ensure Data Principal can *only* view and modify *their own* data/preferences.
- **Real-time Updates:** Consent status and notification feed should be as close to real-time as possible.
- **Audit Trail Linkage:** Every action taken on the dashboard (consent modification, request submission) must be linked to the backend's audit logs.
- DPDP Act Compliance:
    - **Clarity:** Information presented must be clear, concise, and in plain language (multilingual support).
    - **Accessibility:** Adhere to WCAG guidelines for the dashboard interface.
    - **Easy Access:** Dashboard should be easily accessible from the main website/application.
    - **Timely Responses:** System should facilitate timely responses to Data Principal requests, with notifications if SLAs are approaching.
- **Security:** Protect the dashboard from unauthorized access and data leakage. Implement robust data encryption for data displayed or exchanged.

Workflow:

User Login: Data Principal authenticates and accesses their personalized dashboard.

Dashboard Display: The dashboard loads, displaying:

- Summary of current consent.
- Notifications (e.g., "Policy Updated").
- List of pending privacy requests/grievances.

User Action - Modify Consent:

- Data Principal clicks "Modify Consent."
- The "Consent Form Rendering" module loads, pre-filled with current preferences.
- User makes changes and clicks "Save Choices."
- (Consent Storage) Updates the consent record.
- Dashboard updates immediately.

User Action - Submit Grievance:

- Data Principal clicks "Submit Grievance."
- A form appears for entering grievance details.
- User submits the form.
- (Grievance Management System) Creates a new grievance record.
- Dashboard displays the new grievance with "Submitted" status.

User Action - View Consent History:

- Data Principal clicks "View History."
- A paginated list of past consent records loads from "Consent Storage."
- User can click on individual records to see granular details.

3.5. WCAG Guidelines for Consent Forms

This module ensures that the dynamically generated personal data consent forms are fully accessible to all Data Principals, including those with disabilities, by adhering to Web Content Accessibility Guidelines as specified by MeitY's Digital Brand Identity Manual (DBIM). This guarantees that consent is genuinely "informed" and "free" for everyone. The purpose is

- To make the consent collection process usable and understandable for individuals with diverse abilities (e.g., visual, auditory, motor, cognitive impairments).
- To ensure legal compliance with accessibility standards (which may be implicitly required by broader privacy laws and specific regional accessibility acts in India).
- To enhance the overall user experience and trust for all Data Principals.

Key WCAG Principles Applied to Form Elements:

- **Perceivable:** Information and UI components are presented in ways users can perceive.
- **Operable:** UI components and navigation are operable.
- **Understandable:** Information and the operation of UI are understandable.
- **Robust:** Content is robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

Specific WCAG Guidelines & Implementation Requirements:

The dynamic rendering process must ensure the following for every generated element:

- **Keyboard Navigability (Operable 2.1.1):**
  - All interactive elements (buttons, toggles, links) must be navigable and operable using only the keyboard (Tab, Shift+Tab, Enter, Spacebar).
  - Logical tab order must be maintained, flowing naturally through the form.
  - Focus indicator (Operable 2.4.7): A clear visual indicator (e.g., outline) must be present for the currently focused element.
- **Screen Reader Compatibility (Perceivable 1.1.1, Robust 4.1.2):**
  - **Semantic HTML:** Use appropriate HTML5 semantic elements (e.g., <button>, <label>, <input type="checkbox">, <fieldset>, <legend>) for their intended purpose.
  - **ARIA Attributes:**
    - aria-label / aria-labelledby for clear labels on interactive elements, especially custom toggles.
    - aria-describedby to link toggles/checkboxes to their detailed descriptions.
    - aria-checked for custom checkboxes/toggles (if not using native inputs).
    - role="checkbox" / role="switch" for custom toggles.
    - aria-live="polite" for dynamic messages (e.g., confirmation after saving preferences).
  - **Alt Text for Images:** Any informational images (e.g., icons) must have descriptive alt text.
- **Color Contrast & Readability (Perceivable 1.4.3, 1.4.6):**

    - Minimum contrast ratio of **4.5:1** for text and background colors (WCAG AA).
    - Minimum contrast ratio of **3:1** for UI components and graphical objects.
    - Avoid relying solely on color to convey information (e.g., use text labels in addition to color for consent status).
    - Ensure text is resizable without loss of content or functionality (Perceivable 1.4.4).
- **Form Labels & Instructions (Perceivable 1.3.1, Understandable 3.3.2):**
    - All form controls (checkboxes/toggles for purposes) must have clearly associated and visible <label> elements using the for attribute.
    - Clear instructions for completing the form.
    - Error messages (if any validation) must be clear, descriptive, and programmatically associated with the fields.
- **Language Declaration (Understandable 3.1.1):**
    - The primary language of the page (e.g., <html lang="en">) and any language changes within content (e.g.,) must be correctly declared.
- **Consistent Navigation & Identification (Understandable 3.2.3):**
    - Ensure consistent naming, labeling, and presentation across the banner, preference center, and related links.

Functional Requirements:

- **Semantic Element Generation:** The rendering logic must prioritize the use of native HTML form elements (<input type="checkbox">, <label>, <button>) for accessibility benefits. If custom elements are used for styling, they must correctly implement ARIA roles and attributes.
- **Dynamic Attribute Population:** The rendering engine must dynamically populate id, for, aria-label, aria-describedby, aria-checked, role, and tabindex attributes based on policy data and component state.
- **Focus Management:** When the preference center modal opens, focus should be trapped within the modal and the first interactive element should receive focus. On modal close, focus should return to the element that triggered it.
- **Error Handling (for policy loading):** If the policy JSON fails to load or is malformed, display an accessible error message and

provide a compliant fallback mechanism (e.g., "Accept Only Essential Cookies" message).

Workflow Integration:

Policy Retrieval: The "Policy Retrieval" module provides the policy JSON.

Dynamic Rendering Execution: The rendering engine (e.g., React component rendering, jsonform.io instance) processes the JSON.

WCAG-Compliant Element Creation:

- For each purpose in data_processing_purposes, it creates an <input type="checkbox"> or a custom component with role="switch", aria-checked, and visually styled as a toggle.
- It creates a <label> element for each purpose, linking it via for="id_of_checkbox".
- Descriptive text from description is wrapped in an element and linked via aria-describedby.
- Buttons are rendered as <button> elements with clear text.

Accessibility Testing during Development: Developers must use accessibility linters (e.g., Axe-core) and screen readers (e.g., NVDA, VoiceOver) during development to verify compliance.

Testing Considerations:

Automated Testing: Integrate accessibility testing tools (e.g., Axe-core, Lighthouse CI) into the CI/CD pipeline to catch common issues.

Manual Testing: Conduct thorough manual testing with:

- Keyboard-only navigation.
- Screen readers (e.g., NVDA on Windows, VoiceOver on macOS/iOS, TalkBack on Android).
- Users with disabilities, if possible.

Color Contrast Checkers: Use online tools to verify contrast ratios.

3.6. WCAG Guidelines for User Dashboard

This section outlines how the User Dashboard (for Data Principals) will adhere to WCAG 2.1 Level AA, ensuring it is Perceivable, Operable, Understandable, and Robust for all users.

Perceivable (Information and UI are Presentable to Users):

- **Non-text Content (AA):**
  - **Guideline:** All images and non-text content (e.g., icons, charts, graphs for consent trends) must have equivalent **alt text** describing their purpose or information conveyed.
  - **Application:**
    - Profile avatars, notification icons, status indicators (e.g., green checkmark for consent) will have descriptive alt text.
    - Charts showing consent rates or grievance resolution progress will have programmatic labels and descriptions (e.g., <canvas role="img" aria-label="Pie chart showing consent breakdown: 80% Analytics, 20% Marketing">).
- **Info and Relationships (A):**
  - **Guideline:** Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.
  - **Application:**
    - Use semantic HTML: <h1>, <h2>, <h3> for headings, <nav> for navigation, <section>, <article>, <aside>, <footer>.
    - Tables (Consent History, Privacy Requests, Notifications) must use <table>, <th>, <thead>, <tbody>, <tr>, <td> correctly. <th> elements must use the scope attribute (scope="col" or scope="row") to define header relationships.
    - Lists (<ul>, <ol>) for navigation items or notification lists.
    - Form fields within request submission modals are correctly associated with their labels using <label for="id">.
- **Meaningful Sequence (A):**

    - **Guideline:** When the sequence in which content is presented affects its meaning, that sequence can be programmatically determined.
    - **Application:** Ensure content order in HTML matches the visual reading order (e.g., important information comes first in the DOM).
- **Use of Color (A):**
    - **Guideline:** Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.
    - **Application:** Consent status indicators (e.g., green/red dots) will also have text labels (e.g., "Active," "Pending," "Denied") or icons, not just color.
- **Contrast (Minimum) (AA):**
    - **Guideline:** The visual presentation of text and images of text has a contrast ratio of at least 4.5:1. Larger text has a 3:1 ratio.
    - **Application:** All text (headings, labels, body text, link text) and essential icons on the dashboard will meet these contrast ratios against their background colors.
- **Resize text (AA):**
    - **Guideline:** Text can be resized without assistive technology up to 200% without loss of content or functionality.
    - **Application:** Use relative units (em, rem) for font sizes and responsive design techniques so that layout adapts without horizontal scrolling or content overlap when text is zoomed.

Operable (UI Components and Navigation are Operable):

- **Keyboard (A):**
    - **Guideline:** All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes.
    - **Application:**
        - All interactive elements (buttons: "View Details," "Modify Consent," "Submit Grievance"; links: "View All Notifications"; form fields, toggles) are fully operable using Tab, Shift+Tab, Enter, and Spacebar.
        - No mouse-only interactions.
- **No Keyboard Trap (A):**

    - **Guideline:** If keyboard focus can be moved to a component of the content, then focus can be moved away from that component using only a keyboard interface.
    - **Application:** If a modal (e.g., for submitting a request) appears, keyboard focus is trapped within the modal, but the user can easily exit the modal and return to the main page using keyboard (e.g., Esc key, or tabbing to a close button).
- **Focus Order (A):**
    - **Guideline:** If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability.
    - **Application:** The tab order on the dashboard will be logical and intuitive, following the visual flow of information (e.g., top-to-bottom, left-to-right).
- **Focus Visible (AA):**
    - **Guideline:** Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.
    - **Application:** Ensure that default browser focus outlines are not suppressed, or custom focus indicators are clearly visible (e.g., a distinct border or highlight around the focused element).
- **Label in Name (A):**
    - **Guideline:** For user interface components with labels that include text or images of text, the name contains the text that is presented visually.
    - **Application:** The accessible name (what a screen reader reads) of a button or input field must include the visible text label.

Understandable (Information and the Operation of UI are Understandable):

- **Language of Page (A):**
    - **Guideline:** The default human language of each Web page can be programmatically determined.
    - **Application:** Use <html lang="en"> (or ta, hi for respective language versions).
- **Language of Parts (AA):**

    - **Guideline:** The human language of each passage or phrase in the content can be programmatically determined.
    - **Application:** If a section of text is in a different language than the main page (e.g., a Tamil phrase on an English page), use to indicate this.
- **Labels or Instructions (A):**
    - **Guideline:** Labels or instructions are provided when content requires user input.
    - **Application:** All fields in privacy request forms (e.g., for Access, Correction, Grievance) will have clear, persistent, and associated labels.
- **Error Suggestion (AA):**
    - **Guideline:** If an input error is automatically detected, and suggestions for correction are known, then the suggestions are provided to the user.
    - **Application:** For privacy request forms, if an input error occurs (e.g., invalid email format), clearly state the error and suggest how to correct it.
- **Error Prevention (Legal, Financial, Data) (AA):**
    - **Guideline:** For Web pages that cause legal commitments or financial transactions for the user to a secure environment for any kind of data (legal, financial, data), user-controllable, reversible, check, and confirmed methods are adopted.
    - **Application:** For submitting privacy requests or major consent changes, provide opportunities for the user to review and confirm their action before final submission.

Robust (Content is Robust Enough that it can be Interpreted Reliably by a Wide Variety of User Agents):

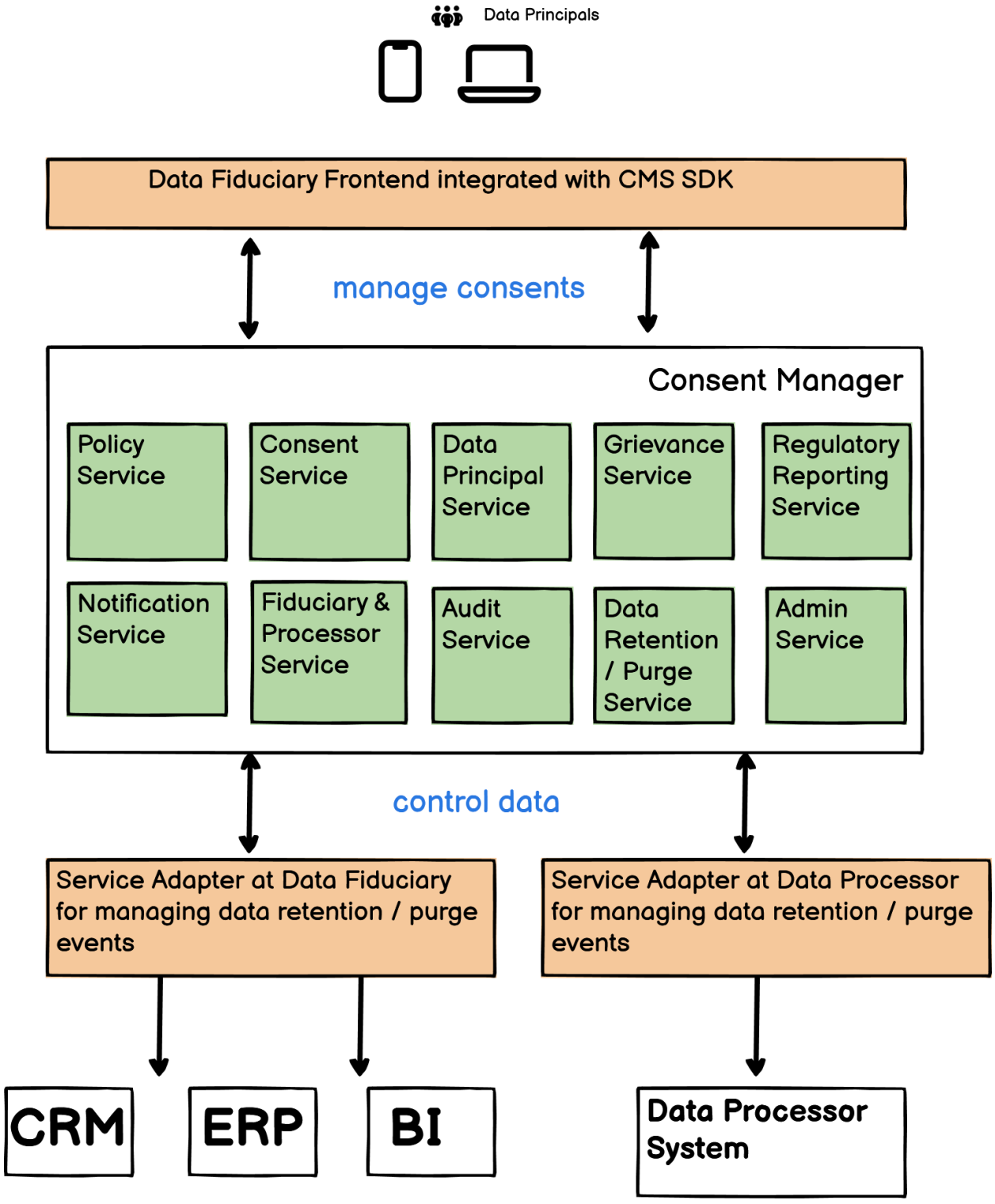
- **Parsing (A):**
    - **Guideline:** In content implemented using markup languages, elements have complete start/end tags, elements are nested according to their specifications, no duplicate attributes, and any IDs are unique.
    - **Application:** Use valid, well-formed HTML.
- **Name, Role, Value (A):**
    - **Guideline:** For all user interface components (including form elements, links, and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be

programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

- **Application:** Ensure custom components (if any, e.g., custom toggle switches) correctly use ARIA attributes (aria-label, role, aria-checked, aria-expanded) to expose their name, role, and state to screen readers.
- **ARIA Live Regions:** Use aria-live regions for dynamic content updates that happen outside the user's immediate focus (e.g., "Request Submitted Successfully" messages appearing without a page refresh)

Technical Design

1. High Level Design



This image depicts a high-level technical architecture for a Consent Manager system. It shows how Consent Manager services interact with Data Fiduciary & Data Processor systems.

2. Core Microservices

This table outlines the key microservices that constitute the DPDP Solution, detailing their core responsibilities, primary data elements, key API interactions, data stores, and inter-service dependencies.

Microservice	Responsibility	Key Data Elements Managed	Key APIs/Interactions
Policy Service	Manages the creation, versioning, and serving of consent policies (JSON structure) for all types of personal data and cookies.	Policy ID, Version, Effective Date, Jurisdiction, Status (Draft, Active, Archived), Full multilingual JSON content (title, purposes, data categories, buttons, links), Associated Fiduciary ID.	POST /policies (Create), PUT /policies/{id}/versions/{v} (Update Draft), POST/policies/{id}/versions/{v}/publish GET /policies/active?fiduciaryId={fiduciaryId}}&lang={l} (Frontend Retrieval)
Consent Record Service	Records, stores, validates, and retrieves Data Principals' granular consent decisions and their lifecycle events.	Consent Record ID, User ID, Policy ID & Version Reference, Timestamp, Jurisdiction, Language, General Consent Status, Mechanism (e.g., "accept_all"), IP Address, User Agent, Granular data_point_consents (JSONB for purposes/categories consented).	POST /consent (Record new/update), GET /consent/{userId} (Retrieve current), GET /consent/{userId}/history, GET /consent/validate?userId={u}&purposeId={p}&dataCategory={d} (Real-time check)

Fiduciary & Processor Service	Registers, configures, and manages profiles for Data Fiduciaries and Data Processors.	Fiduciary: ID, Name, Contact, Primary Domain, CMS CNAME, DNS TXT record for validation, Status, DPB Registration ID, DPO Contact. Processor: ID, Name, Contact, Associated Fiduciary ID, Defined Processing Purposes, Data Categories Processed, DPA Reference.	POST /fiduciaries, PUT /fiduciaries/{id}/validate, GET /fiduciaries/{id}, POST /processors, PUT /processors/{id}, GET /processors/byFiduciary/{id}
User & Role Service	Manages user authentication (for DPO/Admin), authorization (RBAC), and user profile management within the CMS.	User ID, Username, Hashed Password, MFA Status, Assigned Roles, Permissions granted by roles, Session tokens.	POST /users (Register), POST /auth/login, GET /users/{id}/roles, PUT /users/{id}/roles, GET /roles/{id}/permissions, POST /roles/{id}/assign
Grievance Service	Facilitates Data Principal grievance submission, manages their workflow, and tracks resolution status.	Grievance ID, User ID, Fiduciary ID, Description, Severity, Status, Assigned DPO/Agent, Communication Log, Resolution Steps, Timestamps.	POST /grievances (Submit), GET /grievances/{id}, PUT /grievances/{id}/status, GET /grievances/user/{userId} (DP dashboard), GET /grievances/dpo/{dpoId} (DPO dashboard)
Notification Service	Delivers system-generated alerts and messages to relevant users (Data Principals, DPOs, Processors).	Notification ID, Recipient User/Role/Fiduciary ID, Type (e.g., breach, policy update, grievance status), Message Content, Timestamp, Read	POST /notify (Internal trigger), GET /notifications/user/{userId}, GET /notifications/dpo/{dpoId}

		Status.	
Audit Log Service	Centralizes and securely stores an immutable, chronological record of all critical system events and actions.	Log ID, Timestamp, Actor (User ID/System Process), Action Type, Entity Type, Entity ID, Context Details (JSON payload of changes), Source Module, Status (Success/Failure), IP Address.	POST /logs (Internal logging hook), GET /logs (Query/Filter), GET /logs/entity/{entityId}
Data Retention/Purge Service	Manages data retention policies and executes automated/semi-automated data deletion/anonymization operations.	Retention Policy ID, Policy Rules (Fiduciary ID, Purpose, Data Categories, Retention Period, Start Event, Action at Expiry), Purge Job ID, Status, Affected Records Count, Execution Logs.	POST /retention-policies (Define/Update), GET /retention-policies/upcoming-purges, POST /purge-jobs/{jobId}/execute (Internal trigger), GET /purge-reports
Regulatory (DPB) Service	Manages secure communication and report submission with the Data Protection Board (DPB).	DPB Registration ID, Data Fiduciary Credentials (e.g., client cert/key), DPB API Endpoints, Report Templates, Report Submission Status, Submission History, Confirmation Receipts.	POST /dpb/register, POST /dpb/report/{type}, GET /dpb/status, GET /dpb/reports/{id}/status

3. Database Design

This design focuses on the core tables and their relationships, keeping DPDP Act compliance and auditability in mind.

1. Entity-Relationship Diagram (Conceptual):

The core entities revolve around **Data Fiduciaries** who define **Consent Policies**. Data Principals (users) provide their consent, which is recorded as **Consent Records**. Data Fiduciaries engage **Data Processors** for specific purposes. All actions are tracked in **Audit Logs**, and Data Principals can raise **Grievances**. Internal users manage the system with defined **Roles**. Data is managed according to **Retention Policies**.

- `Fiduciaries` --1:N--> `Consent_Policies`
- `Fiduciaries` --1:N--> `Processors`
- `Fiduciaries` --1:N--> `Grievances`
- `Consent_Policies` --1:N--> `Consent_Records`
- `Users` (CMS Admins/DPOs) --N:M--> `Roles`
- `Users` (CMS Admins/DPOs) --generates--> `Audit_Logs`
- `Grievances` --N:1--> `Users` (Assigned DPO/Admin)
- `Retention_Policies` --N:1--> `Fiduciaries` (applies to)
- `Audit_Logs` --N:1--> `Users` (actor)
- `Consent_Records` --N:1--> `Users` (Data Principal, if managed within CMS)

2. Table Schemas:

a) `fiduciaries` (Managed by Fiduciary & Processor Service)

Stores registered Data Fiduciary profiles.

- `id` UUID PRIMARY KEY
- `name` VARCHAR(255) NOT NULL UNIQUE
- `contact_person` VARCHAR(255)
- `email` VARCHAR(255) NOT NULL UNIQUE
- `phone` VARCHAR(50)
- `address` TEXT
- `primary_domain` VARCHAR(255) NOT NULL UNIQUE
- `cms_cname` VARCHAR(255) NOT NULL UNIQUE
- `dns_txt_record_token` VARCHAR(255) UNIQUE
- `domain_validation_status` VARCHAR(50) NOT NULL DEFAULT 'PENDING'
- `is_significant_data_fiduciary` BOOLEAN NOT NULL DEFAULT FALSE
- `dpo_user_id` UUID REFERENCES `users(id)` (FK to CMS user)
- `dpb_registration_id` VARCHAR(100) UNIQUE
- `status` VARCHAR(50) NOT NULL DEFAULT 'ACTIVE' (`ACTIVE`, `INACTIVE`)

- `created_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `created_by_user_id` UUID REFERENCES `users(id)`
- `last_updated_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `last_updated_by_user_id` UUID REFERENCES `users(id)`

b) `processors` (Managed by Fiduciary & Processor Service)

Stores registered Data Processor profiles.

- `id` UUID PRIMARY KEY
- `fiduciary_id` UUID NOT NULL REFERENCES `fiduciaries(id)` (FK to owning Fiduciary)
- `name` VARCHAR(255) NOT NULL
- `contact_person` VARCHAR(255)
- `email` VARCHAR(255)
- `phone` VARCHAR(50)
- `address` TEXT
- `jurisdiction` VARCHAR(50)
- `dpa_reference` VARCHAR(255) UNIQUE
- `dpa_effective_date` DATE
- `dpa_expiry_date` DATE
- `processing_purposes` JSONB NOT NULL DEFAULT '[]' (Array of purpose IDs, e.g., `["purpose_analytics", "purpose_marketing"]`)
- `data_categories_processed` JSONB NOT NULL DEFAULT '[]' (Array of data category IDs, e.g., `["email_address", "Browse_history"]`)
- `security_measures_description` TEXT
- `status` VARCHAR(50) NOT NULL DEFAULT 'ACTIVE' (`ACTIVE`, `INACTIVE`)
- `created_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `created_by_user_id` UUID REFERENCES `users(id)`
- `last_updated_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `last_updated_by_user_id` UUID REFERENCES `users(id)`
- `UNIQUE (fiduciary_id, name)`

c) `consent_policies` (Managed by Policy Service)

Stores definitions of personal data and cookie consent policies.

- `id` VARCHAR(255) NOT NULL

- `version` VARCHAR(10) NOT NULL
- `fiduciary_id` UUID NOT NULL REFERENCES `fiduciaries(id)`
- `effective_date` TIMESTAMP NOT NULL
- `status` VARCHAR(20) NOT NULL DEFAULT 'DRAFT' (`DRAFT`, `ACTIVE`, `ARCHIVED`, `EXPIRED`)
- `jurisdiction` VARCHAR(5) NOT NULL
- `policy_content` JSONB NOT NULL (Full multilingual JSON policy)
- `created_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `created_by_user_id` UUID REFERENCES `users(id)`
- `last_updated_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `last_updated_by_user_id` UUID REFERENCES `users(id)`
- `PRIMARY KEY (id, version)`
- `UNIQUE (fiduciary_id, jurisdiction, status)` (ensures only one active policy per fiduciary/jurisdiction at a time, may need logic for overlapping effective dates)

d) `consent_records` (Managed by Consent Record Service)

Stores every instance of Data Principal consent.

- `id` UUID PRIMARY KEY
- `anon_user_id` VARCHAR(255) NOT NULL (ID from Data Fiduciary's system)
- `principal_id` VARCHAR(255) NOT NULL (Authenticated ID from Data Fiduciary's system)
- `fiduciary_id` UUID NOT NULL REFERENCES `fiduciaries(id)` (FK to the Fiduciary the consent is for)
- `policy_id` VARCHAR(255) NOT NULL
- `policy_version` VARCHAR(10) NOT NULL
- `timestamp` TIMESTAMP NOT NULL (When consent was given)
- `jurisdiction` VARCHAR(5) NOT NULL
- `language_selected` VARCHAR(5) NOT NULL
- `consent_status_general` VARCHAR(50) NOT NULL (`granted`, `denied`, `custom`)
- `consent_mechanism` VARCHAR(100) NOT NULL (e.g., `accept_all_banner`, `preference_center_update`)
- `ip_address` INET (PostgreSQL type for IP address)
- `user_agent` TEXT

- `data_point_consents` JSONB NOT NULL (e.g., `{"purpose_analytics": true, "aadhaar_number": true}`)
- `is_active_consent` BOOLEAN NOT NULL DEFAULT TRUE (Only one `TRUE` per user_id, fiduciary_id)
- `created_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `FOREIGN KEY (policy_id, policy_version) REFERENCES consent_policies(id, version)`
- `UNIQUE (user_id, fiduciary_id, is_active_consent)` (where `is_active_consent` is true)

e) `data_principal` (Managed by Data Principal Service)

Stores Authenticated Data Principals.

- `id` UUID PRIMARY KEY
- `Name` VARCHAR(100) NOT NULL UNIQUE
- `email` VARCHAR(30) NOT NULL UNIQUE
- `mobile` VARCHAR(10) NOT NULL
- `status` VARCHAR(50) NOT NULL DEFAULT 'ACTIVE' (`ACTIVE`, `INACTIVE`)
- `created_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `last_login_at` TIMESTAMP
- `last_updated_at` TIMESTAMP NOT NULL DEFAULT NOW()

f) `backend_users` (Managed by User & Role Service)

Stores CMS user accounts (DPOs, Admins, Auditors, Operators).

- `id` UUID PRIMARY KEY
- `username` VARCHAR(100) NOT NULL UNIQUE
- `email` VARCHAR(255) NOT NULL UNIQUE
- `password_hash` VARCHAR(255) NOT NULL
- `mfa_secret` VARCHAR(255) (for TOTP)
- `mfa_enabled` BOOLEAN NOT NULL DEFAULT FALSE
- `status` VARCHAR(50) NOT NULL DEFAULT 'ACTIVE' (`ACTIVE`, `INACTIVE`, `PENDING_MFA_SETUP`)
- `created_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `last_login_at` TIMESTAMP
- `last_updated_at` TIMESTAMP NOT NULL DEFAULT NOW()

g) `backend_roles` (Managed by User & Role Service)

Defines roles and their permissions.

- `id` UUID PRIMARY KEY
- `name` VARCHAR(100) NOT NULL UNIQUE (e.g., 'Admin', 'DPO', 'Auditor')
- `description` TEXT
- `permissions` JSONB NOT NULL DEFAULT '[]' (Array of strings, e.g., `["fiduciary:create", "policy:publish", "consent:read"]`)
- `created_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `last_updated_at` TIMESTAMP NOT NULL DEFAULT NOW()

h) `backend_user_roles` (Managed by User & Role Service)

Junction table for N:M relationship between users and roles.

- `backend_user_id` UUID NOT NULL REFERENCES `users(id)`
- `backend_role_id` UUID NOT NULL REFERENCES `roles(id)`
- `assigned_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `assigned_by_user_id` UUID REFERENCES `users(id)`
- `PRIMARY KEY (user_id, role_id)`

i) `audit_logs` (Managed by Audit Log Service - Append Only)

Stores immutable logs of all significant system events and user actions.

- `id` UUID PRIMARY KEY
- `timestamp` TIMESTAMP NOT NULL DEFAULT NOW()
- `actor_user_id` UUID REFERENCES `users(id)` (User who performed action, NULL for system actions)
- `actor_system_id` VARCHAR(100) (ID of the system process/service)
- `action_type` VARCHAR(100) NOT NULL (e.g., `POLICY_PUBLISHED`, `CONSENT_UPDATED`, `USER_DEACTIVATED`, `GRIEVANCE_STATUS_CHANGE`)
- `entity_type` VARCHAR(100) NOT NULL (e.g., `ConsentPolicy`, `ConsentRecord`, `User`, `Fiduciary`, `Grievance`)
- `entity_id` UUID/VARCHAR(255) (ID of the affected entity)
- `context_details` JSONB (JSON payload of relevant data/changes, e.g., `{"old_status": "DRAFT", "new_status": "ACTIVE"}`)

- `ip_address` INET
- `status` VARCHAR(50) NOT NULL (`SUCCESS`, `FAILURE`)
- `source_module` VARCHAR(100) NOT NULL (e.g., `PolicyService`, `ConsentRecordService`)

j) `grievances` (Managed by Grievance Service)

Stores Data Principal grievances.

- `id` UUID PRIMARY KEY
- `principal_id` VARCHAR(255) NOT NULL (Data Principal's ID)
- `fiduciary_id` UUID NOT NULL REFERENCES `fiduciaries(id)`
- `subject` VARCHAR(255) NOT NULL
- `description` TEXT NOT NULL
- `submission_timestamp` TIMESTAMP NOT NULL DEFAULT NOW()
- `status` VARCHAR(50) NOT NULL DEFAULT 'NEW' (`NEW`, `IN_PROGRESS`, `PENDING_DPO_REVIEW`, `RESOLVED`, `CLOSED`, `ESCALATED`)
- `assigned_dpo_user_id` UUID REFERENCES `users(id)`
- `resolution_details` TEXT
- `resolution_timestamp` TIMESTAMP
- `last_updated_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `last_updated_by_user_id` UUID REFERENCES `users(id)`
- `due_date` TIMESTAMP (for SLA tracking)

k) `retention_policies` (Managed by Data Retention/Purge Service)

Defines data retention rules.

- `id` UUID PRIMARY KEY
- `fiduciary_id` UUID NOT NULL REFERENCES `fiduciaries(id)`
- `name` VARCHAR(255) NOT NULL
- `description` TEXT
- `applicable_purposes` JSONB NOT NULL DEFAULT '[]' (Array of purposeIDs)
- `applicable_data_categories` JSONB NOT NULL DEFAULT '[]' (Array of data category IDs)
- `retention_duration_value` INTEGER NOT NULL
- `retention_duration_unit` VARCHAR(50) NOT NULL (`DAYS`, `MONTHS`, `YEARS`)

- `retention_start_event` VARCHAR(100) NOT NULL (`CONSENT_GIVEN`, `SERVICE_TERMINATED`, `LAST_ACTIVITY_DATE`, `TRANSACTION_COMPLETED`)
- `action_at_expiry` VARCHAR(50) NOT NULL (`DELETE`, `ANONYMIZE`, `ARCHIVE`)
- `legal_reference` TEXT
- `status` VARCHAR(50) NOT NULL DEFAULT 'ACTIVE' (`ACTIVE`, `INACTIVE`)
- `created_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `created_by_user_id` UUID REFERENCES `users(id)`
- `last_updated_at` TIMESTAMP NOT NULL DEFAULT NOW()
- `last_updated_by_user_id` UUID REFERENCES `users(id)`

4. Security Design

Security by Design and **Privacy by Design** are foundational, ensuring data protection is inherent, not an afterthought.

Layered Defense (Defense-in-Depth):

- **Network Security:** Implement firewalls, intrusion detection/prevention systems (IDS/IPS), and network segmentation to isolate sensitive components. Use Virtual Private Clouds (VPCs) and private subnets.
- **Application Security:** Secure coding practices (OWASP Top 10 mitigation), API security (rate limiting, input validation, secure headers), and regular vulnerability scanning.
- **Data Security:** Encryption of all personal data both **at rest** (database, storage volumes) and **in transit** (HTTPS/TLS 1.2+ for all communication, including internal microservices).
- **Endpoint Security:** Secure configurations for all servers and client devices accessing the system.

Robust Authentication & Authorization:

- **Multi-Factor Authentication (MFA):** Mandate MFA for all CMS administrators, DPOs, and privileged users (e.g., using Email OTP).
- **Role-Based Access Control (RBAC):** Implement granular permissions, ensuring users (including Data Fiduciaries and

Processors) can only access and perform actions strictly defined by their roles (principle of least privilege).
- **Secure Credential Management:** Store hashed passwords (with strong salts) and manage API keys/certificates securely (e.g., via secrets management services).

Data Privacy Controls:

- **Purpose Limitation:** Enforce data processing strictly according to consented purposes.
- **Data Minimization:** Design systems to collect and process only the minimum necessary personal data.
- **Consent Enforcement:** The system will dynamically block or activate data processing based on the user's explicit consent state.
- **Data Retention Policy Enforcement:** Automated deletion or anonymization of data based on configured retention rules.
- **Secure Data Sharing:** All data sharing with Data Processors or third parties is governed by clear contractual agreements (DPAs) and enforced via API-level access controls, ensuring data is transferred only for permitted purposes.

Comprehensive Monitoring & Auditing:

- **Immutable Audit Logs:** Capture all critical system events, access attempts, configuration changes, consent actions, and data processing activities in tamper-proof audit logs.
- **Real-time Monitoring:** Implement logging, monitoring, and alerting tools to detect suspicious activities, security incidents, and processing exceptions.
- **Exception Handling:** Robust error reporting and centralized exception management to ensure timely investigation and resolution of anomalies.

Incident Response & Vulnerability Management:

- **Breach Notification:** Establish clear protocols and tools for rapid detection, assessment, and notification of personal data breaches to the Data Protection Board and affected Data Principals, as mandated by the DPDP Act.
- **Regular Security Assessments:** Conduct frequent vulnerability scans, penetration testing, and security audits (internal and external) to identify and mitigate weaknesses.

- **Secure Development Lifecycle (SDL):** Integrate security practices into every phase of software development, from design to deployment.

5. Deployment Design

This design streamlines the deployment of the DPDP Solution by consolidating all components onto a single server, leveraging Docker Compose for container orchestration. It provides a robust, self-contained environment with simplified management suitable for MSMEs.

1. Core Principles:

- **Consolidation:** All microservices, database, and frontend components run on a single host.
- **Docker Compose:** It manages networking, volumes, and service dependencies on the local host.
- **Primary Database:** A single, persistent PostgreSQL instance serves as the primary data store.
- **Persistence:** All critical data (database, logs, configurations) is persisted using Docker volumes.
- **Simplified Networking:** Docker Compose's default bridge network manages inter-container communication.

2. Deployment Components:

- **Single Server:**
  - **Operating System:** Linux distribution (e.g., Ubuntu LTS, CentOS Stream) is highly recommended.
  - **Docker Engine:** Installed and running, providing the containerization runtime.
  - **Docker Compose:** Installed to manage the multi-container application stack.
- **Docker Containers (defined in docker-compose.yml):**
  - **db (PostgreSQL Container):**
    - **Purpose:** Primary database for all application data.
    - **Image:** postgres:14-alpine (or similar stable version).

        - **Volumes:** Persistent volume mounted to /var/lib/postgresql/data to ensure data survives container restarts.
        - **Environment Variables:** Database name, user, password.
        - **Health Checks:** Configured to ensure the DB is ready before dependent services start.
        - **For Production**: Consider using managed database services from cloud vendors instead of dockerized postgresql installation
    - **backend-services (Consolidated Backend Container(s)):**
        - **Purpose:** Runs all backend microservices (e.g., Policy Service, Consent Record Service, Fiduciary Service, User Service, Audit Log Service, Grievance Service, Data Retention Service, Regulatory Service). For simplicity, these might be packaged into a single large application image or a few logical images if resource constraints allow.
        - **Image:** Custom Docker image built from your backend codebase.
        - **Dependencies:** Configured to depend on the db service.
        - **Ports:** Internal ports exposed to the Docker network.
        - **Environment Variables:** Database connection strings, API keys, etc.
    - **frontend (Web Server Container):**
        - **Purpose:** Serves the static HTML, CSS, JavaScript files for the admin UI and user-facing consent elements. Acts as a reverse proxy for backend APIs.
        - **Image:** nginx:alpine (for static serving and proxying).
        - **Volumes:** Mounts the built frontend static files.
        - **Ports:** Maps standard HTTP/HTTPS ports (e.g., 80, 443) to the host.
        - **Configuration:** nginx.conf configured to serve static files and proxy API requests to the backend-services container.
    - **admin-ui (Optional, if separate):**
        - **Purpose:** Hosts the administrative dashboard frontend.

        - **Image:** Custom Docker image for the Admin UI (e.g., Node.js app serving React/Angular/Vue build).
        - **Dependencies:** Configured to depend on backend-services.
        - **Ports:** Internal port mapped to the host if accessed directly, or proxied via frontend (nginx).
- **Docker Volumes:**
    - db-data-volume: For PostgreSQL data persistence.
    - app-logs-volume: (Optional) For centralized application logs.
    - config-volume: (Optional) For externalized application configuration files.
- **Docker Network:** Docker Compose automatically sets up a default bridge network for containers to communicate by service name (e.g., backend-services can reach db using db:5432).

3. Deployment Workflow:

1. **Server Preparation:**
    - Install Docker Engine and Docker Compose on the single server.
    - Ensure adequate RAM, CPU, and disk space are available.
2. **Code & Configuration:**
    - Clone your DPDP Solution's Git repository to the server.
    - Place your docker-compose.yml file and necessary configuration files (e.g., .env for secrets, nginx.conf) in a designated directory.
3. **Build Docker Images:**
    - If not using pre-built images from a registry, navigate to each service's directory and run docker build -t <image_name> .
4. **Launch Application:**
    - From the directory containing docker-compose.yml, run: docker compose up -d
    - This command builds (if needed), pulls, creates, starts, and links all defined containers in detached mode.
5. **Access:**
    - Access the application via the server's IP address or domain name on the exposed ports (e.g., http://your-server-ip/).
6. **Management:**
    - docker compose down: Stop and remove containers.
    - docker compose ps: View running services.

    - docker compose logs -f <service_name>: View container logs.

4. Security Considerations:

- **Host Security:** Secure the underlying server OS (patching, firewall rules, user access control).
- **Container Security:** Use minimal base images, regularly scan container images for vulnerabilities.
- **Secrets Management:** Do not hardcode secrets in docker-compose.yml. Use .env files or Docker Compose secrets for sensitive information (e.g., DB passwords).
- **Network Access:** Only expose necessary ports to the public internet (typically 80/443 for frontend).
- **SSL/TLS:** Terminate SSL at the Nginx container (frontend) for encrypted communication.

This Docker Compose-based single-node deployment offers simplicity and efficiency for development or small-scale operations, making it easy to set up and manage the entire DPDP solution on one machine.

Technology Choice

- ReactJS for Dashboards (Users, DPO, Data Processor & Admin)
- Java REST API framework
- PostgreSQL
- SDK Integration with Javascript, Flutter/React Native support

Solution Roadmap

Our vision is to continually enhance this DPDP solution, establishing it as one of the reference implementations across the country. While the robust open-source core is designed to fully empower MSMEs, we actively partner with system integrators to customize and extend its capabilities, precisely meeting the complex requirements of larger Data Fiduciaries / larger Consent Aggregators / organisations in other jurisdictions.

1. Consent for Minors

Prompt for Parent/Guardian: The system displays a clear message (multilingual) explaining that parental/guardian consent is required for individuals under 18.
Guardian Details Capture: Prompts for Parent/Guardian's Name, Relationship to Minor, and Mobile Number (linked to Aadhaar for DigiLocker).
Explanation of DigiLocker: Explains that DigiLocker will be used to verify identity and guardianship.
Trigger DigiLocker Verification: The system redirects the user to DigiLocker (or initiates DigiLocker SDK/API interaction).

2. API Setu & Digilocker Integration

We see a future problem: people getting tired of managing their privacy. The user has to log in to *each organisation's* privacy portal to control his/her choices. If the user engages with 10 or 15 different organisations, this quickly becomes too hard to handle.

Our idea is simpler: the user should be able to save a special QR code containing the consent artifact into his/her Digilocker wallet. Then, from that moment on, the user can easily control all privacy choices for *every company* right from the Digilocker app. In order for this to happen, all the DPDP solutions across the country should be interoperable and converge to the same data exchange standard. TSI Coop Foundation will make the software ready for the Consent Manager / Aggregator to publish the APIs to the API Setu platform.

3. TOTP based 2nd Factor Authentication

Alternate MFA Enrollment: As an alternative to Email OTP, the System Administrator should be able to configure TOTP based 2nd factor authentication. This typically involves scanning a QR code with an authenticator app (like Google Authenticator or Authy) to link a time-based one-time password (TOTP) generator, or registering a security key.
Emergency Recovery Setup: The administrator is often prompted to configure backup MFA methods (e.g., recovery codes, backup phone number) to ensure access in case of device loss.

4. Handling Data Purge Requests with Legal Exceptions

The **legal obligation to retain data always takes precedence** over an individual's right to erasure, provided that legal obligation is valid and clearly defined. The DPDP Act includes provisions for such exceptions.

Define and maintain a centralized repository of legal obligations that necessitate data retention, overriding purge requests. The approach ensures strict compliance with legal obligations while transparently communicating the reasons for partial fulfillment to the Data Principal.

5. SSO Integration

For our core MSME users in India, direct Single Sign-On (SSO) integration isn't a primary requirement. However, we fully recognize its critical importance for enterprise-level deployments. Therefore, our system integration partners are empowered to readily customize and extend the solution to include robust SSO capabilities, precisely meeting the complex authentication needs of larger Data Fiduciaries.

Annexure

Annexure A: Example multilingual policy definition for customer onboarding

JSON

```
{
  "policy_id": "personal_data_policy_kyc_credit_v2025_05",
  "version": "1.1",
  "effective_date": "2025-05-26T00:00:00Z",
  "last_updated": "2025-05-26T21:00:00Z",
  "jurisdiction": "IN", // India (DPDP Act)
  "data_fiduciary_info": {
    "name": "SecureCredit Solutions Pvt. Ltd.",
    "address": "45, Financial Hub, R.S. Puram, Coimbatore -
641002, Tamil Nadu, India",
    "email": "privacy@securecredit.com",
    "phone": "+91 80000 54321"
  },
```

```
  "languages": {
    "en": {
      "title": "Your Personal Data, KYC & Credit Privacy
Choices",
      "introduction": "At SecureCredit Solutions, we are
dedicated to safeguarding your personal data, especially
sensitive identity and financial information. This policy details
how we collect, use, and share your personal data, including
Aadhaar, Voter ID, PAN, Credit Bureau Reports, and Bank
Statements, strictly for identity verification (KYC) and credit
decisioning, in full compliance with the Digital Personal Data
Protection Act, 2023, and financial regulations. You have
comprehensive control over your data preferences.",
      "general_purpose_description": "We process personal data to
provide lending and financial services, verify identity (KYC),
assess creditworthiness, prevent fraud, comply with
legal/regulatory obligations (e.g., RBI, SEBI), and offer
customer support.",
      "data_processing_purposes": [
        {
          "id": "purpose_kyc_validation",
          "name": "KYC Validation & Identity Verification",
          "description": "To verify your identity using official
documents like Aadhaar, Voter ID, and PAN for regulatory
compliance and account opening, as mandated by the Reserve Bank
of India (RBI) and Securities and Exchange Board of India (SEBI).
This is essential for providing any regulated financial
service.",
          "legal_basis": "Legal Obligation (DPDP Act, Section
7(h)) and Explicit Consent for specific e-KYC methods (DPDP Act,
Section 6(1)(a)).",
          "data_categories_involved": ["aadhaar_number",
"voter_id_number", "pan_number", "full_name", "date_of_birth",
"address", "biometric_data"],
          "recipients_or_third_parties": ["UIDAI (for Aadhaar
e-KYC)", "National Securities Depository Limited (NSDL) / UTIITSL
```

```
(for PAN verification)", "Regulatory Bodies (RBI, SEBI) as
mandated", "Government Agencies for voter ID verification"],
          "retention_period": "As per Prevention of Money
Laundering Act (PMLA) and RBI/SEBI guidelines (typically 5-8
years post-account closure).",
          "is_mandatory_for_service": true,
          "is_sensitive": true
        },
        {
          "id": "purpose_credit_decisioning",
          "name": "Credit Decisioning & Risk Assessment",
          "description": "To assess your creditworthiness and
eligibility for financial products (e.g., loans, credit cards) by
analyzing your credit bureau reports and bank statements, as per
our lending policies.",
          "legal_basis": "Explicit Consent (DPDP Act, Section
6(1)(a)) and Contractual Necessity (DPDP Act, Section 7(b)) for
loan/credit applications.",
          "data_categories_involved": ["credit_bureau_report",
"bank_statements", "financial_transaction_data",
"income_details", "loan_account_details"],
          "recipients_or_third_parties": ["Credit Information
Companies (e.g., CIBIL, Experian, Equifax)", "Bank APIs (for
statement fetching, with your authentication)", "Internal Risk
Assessment Teams"],
          "retention_period": "As per RBI guidelines for lending
institutions (e.g., 7-10 years post-loan closure/settlement).",
          "is_mandatory_for_service": true,
          "is_sensitive": true
        },
        {
          "id": "purpose_service_delivery",
          "name": "Core Service Delivery & Account Management",
          "description": "To process your financial product
applications, manage your account, execute transactions, and
```

```
provide essential functionalities. This is fundamental to our
services.",
          "legal_basis": "Contractual Necessity (DPDP Act,
Section 7(b))",
          "data_categories_involved": ["contact_information",
"payment_details", "transaction_history",
"loan_account_details"],
          "recipients_or_third_parties": ["Payment Gateways",
"Banking Partners"],
          "retention_period": "As per contractual obligations and
regulatory financial record-keeping requirements.",
          "is_mandatory_for_service": true,
          "is_sensitive": false
        },
        {
          "id": "purpose_customer_support",
          "name": "Customer Support & Communication",
          "description": "To respond to your inquiries, provide
technical support, manage feedback, and communicate about your
service requests.",
          "legal_basis": "Consent (DPDP Act, Section 6(1)(a)) or
Legitimate Use (DPDP Act, Section 7(g) - reasonable expectation
of data principal).",
          "data_categories_involved": ["contact_information",
"support_query_details", "communication_history", "device_info"],
          "recipients_or_third_parties": ["Helpdesk Software
Providers"],
          "retention_period": "Up to 3 years after resolution of
the last support query, or longer if required by law.",
          "is_mandatory_for_service": false,
          "is_sensitive": false
        },
        {
          "id": "purpose_marketing_communication",
          "name": "Personalized Marketing & Offers",
```

```
          "description": "To send you personalized offers,
updates on financial products, and promotional content relevant
to your interests, based on your preferences and eligibility
(excluding data used for KYC/credit decisioning unless separate
explicit consent is given).",
          "legal_basis": "Consent (DPDP Act, Section 6(1)(a))",
          "data_categories_involved": ["email_address",
"phone_number", "basic_profile_info",
"product_interest_summary"],
          "recipients_or_third_parties": ["Email Marketing
Platforms", "SMS Gateway Providers"],
          "retention_period": "Until consent is withdrawn or 2
years of inactivity.",
          "is_mandatory_for_service": false,
          "is_sensitive": false
        }
      ],
      "data_categories_details": [
        {
          "id": "aadhaar_number",
          "name": "Aadhaar Number",
          "description": "Your 12-digit unique identification
number issued by UIDAI. Collected for mandatory identity
verification and e-KYC. This is highly sensitive personal data.",
          "is_sensitive": true
        },
        {
          "id": "voter_id_number",
          "name": "Voter ID Number (EPIC)",
          "description": "Your unique identification number from
your Indian Voter ID card (Electors Photo Identity Card). Used as
an alternative identity proof for KYC. This is highly sensitive
personal data.",
          "is_sensitive": true
        },
        {
```

```
          "id": "pan_number",
          "name": "PAN (Permanent Account Number)",
          "description": "Your 10-character alphanumeric
identifier issued by the Income Tax Department. Mandatory for
financial transactions and KYC. This is sensitive personal
data.",
          "is_sensitive": true
        },
        {
          "id": "biometric_data",
          "name": "Biometric Data (e.g., Fingerprint/Iris Scan)",
          "description": "Unique physical characteristics used
for Aadhaar e-KYC or other biometric authentication. This is
highly sensitive personal data.",
          "is_sensitive": true
        },
        {
          "id": "credit_bureau_report",
          "name": "Credit Bureau Report",
          "description": "Comprehensive report from credit
information companies detailing your credit history, scores, and
past loan/credit card behaviour. This is highly sensitive
financial data.",
          "is_sensitive": true
        },
        {
          "id": "bank_statements",
          "name": "Bank Statements",
          "description": "Detailed records of your financial
transactions, account balances, and income/expenditure patterns
from your bank accounts. This is highly sensitive financial
data.",
          "is_sensitive": true
        },
        {
          "id": "contact_information",
```

```
          "name": "Contact Information",
          "description": "Includes your full name, email address,
phone number, and mailing address."
        },
        {
          "id": "full_name",
          "name": "Full Name",
          "description": "Your legal first name, middle name (if
any), and last name."
        },
        {
          "id": "date_of_birth",
          "name": "Date of Birth",
          "description": "Your full date of birth (DD/MM/YYYY)."
        },
        {
          "id": "address",
          "name": "Address",
          "description": "Your residential or mailing address."
        },
        {
          "id": "payment_details",
          "name": "Payment Details",
          "description": "Includes encrypted payment instrument
numbers, billing address, and transaction amounts. (Actual
sensitive financial details are handled by secure payment
gateways and are not stored by us directly)."
        },
        {
          "id": "transaction_history",
          "name": "Transaction History",
          "description": "Details of your financial transactions
and services availed."
        },
        {
          "id": "loan_account_details",
```

```
          "name": "Loan Account Details",
          "description": "Information related to your loan
accounts with us, including loan amount, EMI, tenure, and
repayment history."
        },
        {
          "id": "income_details",
          "name": "Income Details",
          "description": "Information related to your stated or
verified income, including salary, business income, etc."
        },
        {
          "id": "basic_profile_info",
          "name": "Basic Profile Information",
          "description": "General demographic and profile data
not covered by other categories."
        },
        {
          "id": "product_interest_summary",
          "name": "Product Interest Summary",
          "description": "Aggregated or inferred data about your
interest in specific financial products."
        },
        {
          "id": "support_query_details",
          "name": "Support Query Details",
          "description": "Content of your support tickets and
interactions."
        },
        {
          "id": "communication_history",
          "name": "Communication History",
          "description": "Records of emails, SMS, and calls with
our customer service."
        },
        {
```

```
          "id": "usage_data",
          "name": "Usage Data",
          "description": "Information about how you use our
services, features accessed, and time spent."
        },
        {
          "id": "preferences",
          "name": "User Preferences",
          "description": "Your saved settings and preferences
within our application."
        },
        {
          "id": "device_info",
          "name": "Device Information",
          "description": "Details about the device you use,
including IP address, operating system, browser type, and unique
device identifiers."
        }
      ],
      "data_principal_rights_summary": "As per the Digital
Personal Data Protection Act, 2023, you have robust rights
including the right to access information, correction,
completion, updating, erasure of your personal data, and the
right to grievance redressal. You also have the right to withdraw
consent for specific purposes, provided it does not violate
contractual obligations or legal mandates. For complete details,
please refer to our full Privacy Policy.",
      "grievance_redressal_info": "For any data protection
concerns or to exercise your rights under the DPDP Act, please
contact our Data Protection Officer at: Email:
dpo@securecredit.com | Phone: +91 80000 54321 | Address: 45,
Financial Hub, R.S. Puram, Coimbatore - 641002, Tamil Nadu,
India. Your concern will be addressed within the stipulated
timeframes.",
      "buttons": {
        "accept_all": "Accept All & Continue",
```

```
        "reject_all_non_essential": "Reject Non-Essential",
        "manage_preferences": "Manage My Preferences",
        "save_preferences": "Save Choices"
      },
      "links": {
        "full_privacy_policy_text": "Full Privacy Policy",
        "full_privacy_policy_url":
"https://www.securecredit.com/privacy-policy-en",
        "terms_of_service_text": "Terms of Service",
        "terms_of_service_url":
"https://www.securecredit.com/terms-of-service-en"
      },
      "important_note": "For purposes explicitly marked
'Mandatory for Service', your data is essential for contract
fulfillment or legal/regulatory compliance. Denying consent for
these may prevent us from providing the core service. For KYC and
Credit Decisioning, specific explicit consent (where applicable)
and compliance with legal mandates are required."
    },
    "ta": {
      "title": "உங்கள் தனிப்பட்ட தரவு, KYC மற்றும் கடன்
தனியுரிமைத் தேர்வுகள்",
      "introduction": "செக்யூர் கிரெடிட் சொல்யூஷன்ஸில், உங்கள்
தனிப்பட்ட தரவைப் பாதுகாப்பதற்கும், குறிப்பாக உணர்வுப்பூர்வமான
அடையாளம் மற்றும் நிதித் தகவல்களைப் பாதுகாப்பதற்கும் நாங்கள்
அர்ப்பணிப்புடன் உள்ளோம். 2023 டிஜிட்டல் தனிப்பட்ட தரவுப்
பாதுகாப்புச் சட்டம் மற்றும் நிதி ஒழுங்குமுறைகளுக்கு முழு இணங்க,
ஆதார், வாக்காளர் அடையாள அட்டை, பான், கடன் பணியக
அறிக்கைகள் மற்றும் வங்கிக் கணக்கு அறிக்கைகள் உட்பட உங்கள்
தனிப்பட்ட தரவை நாங்கள் எவ்வாறு சேகரிக்கிறோம்,
பயன்படுத்துகிறோம் மற்றும் பகிர்கிறோம் என்பதை இந்தக் கொள்கை
விவரிக்கிறது, இது அடையாள சரிபார்ப்பு (KYC) மற்றும் கடன்
முடிவெடுக்கும் நோக்கங்களுக்காக மட்டுமே. உங்கள் தரவு
விருப்பங்களின் மீது உங்களுக்கு விரிவான கட்டுப்பாடு உள்ளது.",
      "general_purpose_description": "கடன் மற்றும் நிதி
சேவைகளை வழங்க, அடையாளத்தைச் சரிபார்க்க (KYC), கடன்
```

தகுதியை மதிப்பிட, மோசடியைத் தடுக்க, சட்ட/ஒழுங்குமுறை கடமைகளுக்கு (எ.கா., RBI, SEBI) இணங்க, மற்றும் வாடிக்கையாளர் ஆதரவை வழங்க நாங்கள் தனிப்பட்ட தரவைச் செயலாக்குகிறோம்.",
      "data_processing_purposes": [
        {
          "id": "purpose_kyc_validation",
          "name": "KYC சரிபார்ப்பு மற்றும் அடையாள சரிபார்ப்பு",
          "description": "இந்திய ரிசர்வ் வங்கி (RBI) மற்றும் இந்தியப் பத்திரங்கள் மற்றும் பரிவர்த்தனை வாரியம் (SEBI) கட்டளையிட்டபடி, ஒழுங்குமுறை இணக்கம் மற்றும் கணக்கு திறப்புக்காக ஆதார், வாக்காளர் அடையாள அட்டை மற்றும் பான் போன்ற அதிகாரப்பூர்வ ஆவணங்களைப் பயன்படுத்தி உங்கள் அடையாளத்தைச் சரிபார்க்க. எந்தவொரு ஒழுங்குபடுத்தப்பட்ட நிதி சேவையை வழங்குவதற்கும் இது அத்தியாவசியமானது.",
          "legal_basis": "சட்டக் கடமை (டிபிடிபி சட்டம், பிரிவு 7(h)) மற்றும் குறிப்பிட்ட இ-KYC முறைகளுக்கான வெளிப்படையான ஒப்புதல் (டிபிடிபி சட்டம், பிரிவு 6(1)(a)).",
          "data_categories_involved": ["aadhaar_number", "voter_id_number", "pan_number", "full_name", "date_of_birth", "address", "biometric_data"],
          "recipients_or_third_parties": ["UIDAI (ஆதார் இ-KYC க்காக)", "தேசியப் பத்திரங்கள் வைப்பு நிறுவனம் (NSDL) / UTIITSL (பான் சரிபார்ப்புக்காக)", "அதிகாரமளிக்கப்பட்ட ஒழுங்குமுறை அமைப்புகள் (RBI, SEBI)", "வாக்காளர் அடையாள அட்டை சரிபார்ப்புக்காக அரசு முகமைகள்"],
          "retention_period": "பணமோசடி தடுப்புச் சட்டம் (PMLA) மற்றும் RBI/SEBI வழிகாட்டுதல்களின்படி (பொதுவாக கணக்கு மூடிய பிறகு 5-8 ஆண்டுகள்).",
          "is_mandatory_for_service": true,
          "is_sensitive": true
        },
        {
          "id": "purpose_credit_decisioning",
          "name": "கடன் முடிவெடுக்கும் மற்றும் இடர் மதிப்பீடு",

```
          "description": "எங்கள் கடன் கொள்கைகளின்படி, உங்கள் கடன் பணியக அறிக்கைகள் மற்றும் வங்கிக் கணக்கு அறிக்கைகளை பகுப்பாய்வு செய்வதன் மூலம் உங்கள் கடன் தகுதி மற்றும் நிதி தயாரிப்புகளுக்கான (எ.கா., கடன்கள், கிரெடிட் கார்டுகள்) தகுதியை மதிப்பிட.",
          "legal_basis": "வெளிப்படையான ஒப்புதல் (டிபிடிபி சட்டம், பிரிவு 6(1)(a)) மற்றும் கடன்/கடன் விண்ணப்பங்களுக்கான ஒப்பந்தத் தேவை (டிபிடிபி சட்டம், பிரிவு 7(b)).",
          "data_categories_involved": ["credit_bureau_report", "bank_statements", "financial_transaction_data", "income_details", "loan_account_details"],
          "recipients_or_third_parties": ["கடன் தகவல் நிறுவனங்கள் (எ.கா., CIBIL, Experian, Equifax)", "வங்கி APIகள் (அறிக்கை எடுப்பதற்குக், உங்கள் அங்கீகாரத்துடன்)", "உள் இடர் மதிப்பீட்டுக் குழுக்கள்"],
          "retention_period": "கடன் வழங்கும் நிறுவனங்களுக்கான RBI வழிகாட்டுதல்களின்படி (எ.கா., கடன் மூடிய பிறகு/தீர்வுக்குப் பிறகு 7-10 ஆண்டுகள்).",
          "is_mandatory_for_service": true,
          "is_sensitive": true
        },
        {
          "id": "purpose_service_delivery",
          "name": "முக்கிய சேவை வழங்கல் மற்றும் கணக்கு மேலாண்மை",
          "description": "உங்கள் நிதி தயாரிப்பு விண்ணப்பங்களைச் செயலாக்க, உங்கள் கணக்கை நிர்வகிக்க, பரிவர்த்தனைகளைச் செயல்படுத்த, மற்றும் அத்தியாவசிய செயல்பாடுகளை வழங்க. இது எங்கள் சேவைகளுக்கு அடிப்படையானது.",
          "legal_basis": "ஒப்பந்தத் தேவை (டிபிடிபி சட்டம், பிரிவு 7(b))",
          "data_categories_involved": ["contact_information", "payment_details", "transaction_history", "loan_account_details"],
          "recipients_or_third_parties": ["கட்டண நுழைவாயில்கள்", "வங்கி கூட்டாளர்கள்"],
```

```
          "retention_period": "ஒப்பந்தக் கடமைகள் மற்றும் ஒழுங்குமுறை நிதி பதிவு வைத்தல் தேவைகளின்படி.",
          "is_mandatory_for_service": true,
          "is_sensitive": false
        },
        {
          "id": "purpose_customer_support",
          "name": "வாடிக்கையாளர் ஆதரவு மற்றும் தொடர்பு",
          "description": "உங்கள் விசாரணைகளுக்கு பதிலளிக்க, தொழில்நுட்ப ஆதரவை வழங்க, கருத்துக்களை நிர்வகிக்க, மற்றும் உங்கள் சேவை கோரிக்கைகள் குறித்து தொடர்புகொள்ள.",
          "legal_basis": "ஒப்புதல் (டிபிடிபி சட்டம், பிரிவு 6(1)(a)) அல்லது சட்டப்பூர்வ பயன்பாடு (டிபிடிபி சட்டம், பிரிவு 7(g) - தரவு முதல்வரின் நியாயமான எதிர்பார்ப்பு).",
          "data_categories_involved": ["contact_information", "support_query_details", "communication_history", "device_info"],
          "recipients_or_third_parties": ["ஹெல்ப் டெஸ்க் மென்பொருள் வழங்குநர்கள்"],
          "retention_period": "கடைசி ஆதரவு கேள்வி தீர்க்கப்பட்ட பிறகு 3 ஆண்டுகள் வரை, அல்லது சட்டத்தால் தேவைப்பட்டால் நீண்ட காலம்.",
          "is_mandatory_for_service": false,
          "is_sensitive": false
        },
        {
          "id": "purpose_marketing_communication",
          "name": "தனிப்பயனாக்கப்பட்ட சந்தைப்படுத்தல் மற்றும் சலுகைகள்",
          "description": "உங்கள் விருப்பங்கள் மற்றும் தகுதியின் அடிப்படையில் உங்களுக்குத் தனிப்பயனாக்கப்பட்ட சலுகைகள், நிதி தயாரிப்புகள் குறித்த அறிவிப்புகள் மற்றும் விளம்பர உள்ளடக்கங்களை அனுப்ப (தனியாக வெளிப்படையான ஒப்புதல் வழங்கப்படாவிட்டால் KYC/கடன் முடிவெடுப்பதற்காகப் பயன்படுத்தப்படும் தரவு தவிர).",
          "legal_basis": "ஒப்புதல் (டிபிடிபி சட்டம், பிரிவு 6(1)(a))",
```

```
          "data_categories_involved": ["email_address",
"phone_number", "basic_profile_info",
"product_interest_summary"],
          "recipients_or_third_parties": ["மின்னஞ்சல்
சந்தைப்படுத்தல் தளங்கள்", "எஸ்எம்எஸ் கேட்வே வழங்குநர்கள்"],
          "retention_period": "ஒப்புதல் திரும்பப் பெறப்படும் வரை
அல்லது 2 ஆண்டுகள் செயலற்ற தன்மை வரை.",
          "is_mandatory_for_service": false,
          "is_sensitive": false
        }
      ],
      "data_categories_details": [
        {
          "id": "aadhaar_number",
          "name": "ஆதார் எண்",
          "description": "UIDAI ஆல் வழங்கப்பட்ட உங்கள் 12
இலக்க தனிப்பட்ட அடையாள எண். கட்டாய அடையாள சரிபார்ப்பு
மற்றும் இ-KYC க்காக சேகரிக்கப்பட்டது. இது மிகவும்
உணர்வுப்பூர்வமான தனிப்பட்ட தரவு.",
          "is_sensitive": true
        },
        {
          "id": "voter_id_number",
          "name": "வாக்காளர் அடையாள அட்டை எண் (EPIC)",
          "description": "உங்கள் இந்திய வாக்காளர் அடையாள
அட்டையிலிருந்து உங்கள் தனிப்பட்ட அடையாள எண். KYC க்கான
மாற்று அடையாள ஆதாரமாகப் பயன்படுத்தப்படுகிறது. இது மிகவும்
உணர்வுப்பூர்வமான தனிப்பட்ட தரவு.",
          "is_sensitive": true
        },
        {
          "id": "pan_number",
          "name": "PAN (நிரந்தர கணக்கு எண்)",
          "description": "வருமான வரித் துறையால் வழங்கப்பட்ட
உங்கள் 10 எழுத்துக்கள் கொண்ட எண்ணெழுத்து அடையாளங்காட்டி.
```

நிதி பரிவர்த்தனைகள் மற்றும் KYC க்காக கட்டாயமானது. இது உணர்வுப்பூர்வமான தனிப்பட்ட தரவு.",
          "is_sensitive": true
        },
        {
          "id": "biometric_data",
          "name": "பயோமெட்ரிக் தரவு (எ.கா., கைரேகை/கண் கருவிழி ஸ்கேன்)",
          "description": "ஆதார் இ-KYC அல்லது பிற பயோமெட்ரிக் அங்கீகாரத்திற்குப் பயன்படுத்தப்படும் தனிப்பட்ட உடல் பண்புகள். இது மிகவும் உணர்வுப்பூர்வமான தனிப்பட்ட தரவு.",
          "is_sensitive": true
        },
        {
          "id": "credit_bureau_report",
          "name": "கடன் பணியக அறிக்கை",
          "description": "உங்கள் கடன் வரலாறு, மதிப்பெண்கள் மற்றும் கடந்தகால கடன்/கிரெடிட் கார்டு நடத்தை ஆகியவற்றை விவரிக்கும் கடன் தகவல் நிறுவனங்களின் விரிவான அறிக்கை. இது மிகவும் உணர்வுப்பூர்வமான நிதித் தரவு.",
          "is_sensitive": true
        },
        {
          "id": "bank_statements",
          "name": "வங்கி அறிக்கைகள்",
          "description": "உங்கள் வங்கிக் கணக்குகளிலிருந்து உங்கள் நிதி பரிவர்த்தனைகள், கணக்கு நிலுவைகள் மற்றும் வருமானம்/செலவின முறைகளின் விரிவான பதிவுகள். இது மிகவும் உணர்வுப்பூர்வமான நிதித் தரவு.",
          "is_sensitive": true
        },
        {
          "id": "contact_information",
          "name": "தொடர்பு தகவல்",

```
          "description": "உங்கள் முழு பெயர், மின்னஞ்சல் முகவரி, தொலைபேசி எண் மற்றும் அஞ்சல் முகவரி ஆகியவை அடங்கும்."
        },
        {
          "id": "full_name",
          "name": "முழு பெயர்",
          "description": "உங்கள் சட்டப்பூர்வ முதல் பெயர், நடுப் பெயர் (ஏதேனும் இருந்தால்), மற்றும் கடைசிப் பெயர்."
        },
        {
          "id": "date_of_birth",
          "name": "பிறந்த தேதி",
          "description": "உங்கள் முழு பிறந்த தேதி (நாள்/மாதம்/ஆண்டு)."
        },
        {
          "id": "address",
          "name": "முகவரி",
          "description": "உங்கள் வசிப்பிட அல்லது அஞ்சல் முகவரி."
        },
        {
          "id": "payment_details",
          "name": "கட்டண விவரங்கள்",
          "description": "குறியாக்கப்பட்ட கட்டண கருவி எண்கள், பில்லிங் முகவரி மற்றும் பரிவர்த்தனை தொகைகள் அடங்கும். (உண்மையான உணர்திறன் நிதி விவரங்கள் பாதுகாப்பான கட்டண நுழைவாயில்களால் கையாளப்படுகின்றன மற்றும் நேரடியாக எங்களால் சேமிக்கப்படுவதில்லை)."
        },
        {
          "id": "transaction_history",
          "name": "பரிவர்த்தனை வரலாறு",
          "description": "உங்கள் நிதி பரிவர்த்தனைகள் மற்றும் சேவைகளின் விவரங்கள்."
```

```
        },
        {
          "id": "loan_account_details",
          "name": "கடன் கணக்கு விவரங்கள்",
          "description": "எங்களுடன் உங்கள் கடன் கணக்குகள்
தொடர்பான தகவல், கடன் தொகை, EMI, காலம் மற்றும் திருப்பிச்
செலுத்தும் வரலாறு உட்பட."
        },
        {
          "id": "income_details",
          "name": "வருமான விவரங்கள்",
          "description": "உங்கள் கூறப்பட்ட அல்லது
சரிபார்க்கப்பட்ட வருமானம் தொடர்பான தகவல், சம்பளம், வணிக
வருமானம் போன்றவை."
        },
        {
          "id": "basic_profile_info",
          "name": "அடிப்படை சுயவிவரத் தகவல்",
          "description": "பிற வகைகளால் உள்ளடக்கப்படாத
பொதுவான மக்கள்தொகை மற்றும் சுயவிவரத் தரவு."
        },
        {
          "id": "product_interest_summary",
          "name": "தயாரிப்பு ஆர்வ சுருக்கம்",
          "description": "குறிப்பிட்ட நிதி தயாரிப்புகளில் உங்கள்
ஆர்வம் பற்றிய தொகுக்கப்பட்ட அல்லது ஊகிக்கப்பட்ட தரவு."
        },
        {
          "id": "support_query_details",
          "name": "ஆதரவு கேள்வி விவரங்கள்",
          "description": "உங்கள் ஆதரவு டிக்கெட்டுகள் மற்றும்
தொடர்புகளின் உள்ளடக்கம்."
        },
        {
          "id": "communication_history",
          "name": "தொடர்பு வரலாறு",
```

```
          "description": "எங்கள் வாடிக்கையாளர் சேவைக்கான மின்னஞ்சல்கள், SMS மற்றும் அழைப்புகளின் பதிவுகள்."
        },
        {
          "id": "usage_data",
          "name": "பயன்பாட்டுத் தரவு",
          "description": "நீங்கள் எங்கள் சேவைகளை எவ்வாறு பயன்படுத்துகிறீர்கள், அணுகப்பட்ட அம்சங்கள் மற்றும் செலவழித்த நேரம் பற்றிய தகவல்."
        },
        {
          "id": "preferences",
          "name": "பயனர் விருப்பங்கள்",
          "description": "எங்கள் பயன்பாட்டில் உங்கள் சேமிக்கப்பட்ட அமைப்புகள் மற்றும் விருப்பங்கள்."
        },
        {
          "id": "device_info",
          "name": "சாதன தகவல்",
          "description": "IP முகவரி, இயக்க முறைமை, உலாவி வகை மற்றும் தனிப்பட்ட சாதன அடையாளங்காட்டிகள் உள்ளிட்ட நீங்கள் பயன்படுத்தும் சாதனத்தைப் பற்றிய விவரங்கள்."
        }
      ],
      "data_principal_rights_summary": "டிஜிட்டல் தனிப்பட்ட தரவுப் பாதுகாப்புச் சட்டம், 2023 இன் படி, தகவல் அணுகல், திருத்தம், பூர்த்தி செய்தல், புதுப்பித்தல் மற்றும் உங்கள் தனிப்பட்ட தரவை அழித்தல் உள்ளிட்ட வலுவான உரிமைகள் உங்களுக்கு உள்ளன, அத்துடன் குறை தீர்க்கும் உரிமையும் உள்ளது. ஒப்பந்தக் கடமைகள் அல்லது சட்டக் கட்டளைகளை மீறாத பட்சத்தில், குறிப்பிட்ட நோக்கங்களுக்காக ஒப்புதலை திரும்பப் பெறவும் உங்களுக்கு உரிமை உண்டு. முழு விவரங்களுக்கு, எங்கள் முழு தனியுரிமைக் கொள்கையைப் பார்க்கவும்.",
      "grievance_redressal_info": "எந்தவொரு தரவு பாதுகாப்பு கவலைகள் அல்லது டிபிடிபி சட்டத்தின் கீழ் உங்கள் உரிமைகளைப் பயன்படுத்த, எங்கள் தரவு பாதுகாப்பு அதிகாரி: மின்னஞ்சல்:
```

dpo@securecredit.com | தொலைபேசி: +91 80000 54321 | முகவரி: 45, Financial Hub, R.S. Puram, Coimbatore - 641002, Tamil Nadu, India ஐ தொடர்பு கொள்ளவும். உங்கள் கவலைகள் குறிப்பிட்ட காலக்கெடுவுக்குள் தீர்க்கப்படும்.",
      "buttons": {
        "accept_all": "அனைத்தையும் ஏற்றுக்கொண்டு தொடரவும்",
        "reject_all_non_essential": "அத்தியாவசியமற்றவற்றை நிராகரிக்கவும்",
        "manage_preferences": "எனது விருப்பங்களை நிர்வகிக்கவும்",
        "save_preferences": "தேர்வுகளைச் சேமிக்கவும்"
      },
      "links": {
        "full_privacy_policy_text": "முழு தனியுரிமைக் கொள்கை",
        "full_privacy_policy_url": "https://www.securecredit.com/privacy-policy-ta",
        "terms_of_service_text": "சேவை விதிமுறைகள்",
        "terms_of_service_url": "https://www.securecredit.com/terms-of-service-ta"
      },
      "important_note": "தெளிவாக 'சேவைக்கு கட்டாயமானது' எனக் குறிக்கப்பட்ட நோக்கங்களுக்காக, உங்கள் தரவு ஒப்பந்த பூர்த்தி அல்லது சட்ட/ஒழுங்குமுறை இணக்கத்திற்கு அவசியம். இவற்றுக்கு ஒப்புதலை மறுப்பது, முக்கிய சேவையை நாங்கள் வழங்குவதைத் தடுக்கலாம். KYC மற்றும் கடன் முடிவெடுக்கும் நோக்கங்களுக்காக, குறிப்பிட்ட வெளிப்படையான ஒப்புதல் (பொருந்தும் இடங்களில்) மற்றும் சட்டக் கட்டளைகளுக்கு இணங்குதல் தேவை."
    },
    "hi": {
      "title": "आपका व्यक्तिगत डेटा, केवाईसी और क्रेडिट गोपनीयता विकल्प",
      "introduction": "सिक्योरक्रेडिट सॉल्यूशंस में, हम आपके व्यक्तिगत डेटा की सुरक्षा के लिए समर्पित हैं, विशेष रूप से संवेदनशील पहचान और वित्तीय जानकारी। यह नीति बताती है कि हम डिजिटल व्यक्तिगत डेटा संरक्षण अधिनियम, 2023 और वित्तीय विनियमों के पूर्ण अनुपालन में, पहचान सत्यापन (केवाईसी) और क्रेडिट निर्णय लेने के लिए आपके व्यक्तिगत डेटा, जिसमें आधार, मतदाता पहचान पत्र, पैन, क्रेडिट ब्यूरो रिपोर्ट और

बैंक स्टेटमेंट शामिल हैं, को कैसे एकत्र, उपयोग और साझा करते हैं। आपके डेटा प्राथमिकताओं पर आपका व्यापक नियंत्रण है।",
"general_purpose_description": "हम उधार और वित्तीय सेवाएं प्रदान करने, पहचान सत्यापित करने (केवाईसी), क्रेडिट योग्यता का आकलन करने, धोखाधड़ी को रोकने, कानूनी/नियामक दायित्वों (जैसे आरबीआई, सेबी) का पालन करने और ग्राहक सहायता प्रदान करने के लिए व्यक्तिगत डेटा संसाधित करते हैं।",
"data_processing_purposes": [
{
"id": "purpose_kyc_validation",
"name": "केवाईसी सत्यापन और पहचान सत्यापन",
"description": "भारतीय रिजर्व बैंक (आरबीआई) और भारतीय प्रतिभूति और विनिमय बोर्ड (सेबी) द्वारा अनिवार्य रूप से नियामक अनुपालन और खाता खोलने के लिए आधार, मतदाता पहचान पत्र और पैन जैसे आधिकारिक दस्तावेजों का उपयोग करके आपकी पहचान सत्यापित करने के लिए। यह कोई भी विनियमित वित्तीय सेवा प्रदान करने के लिए आवश्यक है।",
"legal_basis": "कानूनी दायित्व (डीपीडीपी अधिनियम, धारा 7(h)) और विशिष्ट ई-केवाईसी विधियों के लिए स्पष्ट सहमति (डीपीडीपी अधिनियम, धारा 6(1)(a))।",
"data_categories_involved": ["aadhaar_number", "voter_id_number", "pan_number", "full_name", "date_of_birth", "address", "biometric_data"],
"recipients_or_third_parties": ["UIDAI (आधार ई-केवाईसी के लिए)", "नेशनल सिक्योरिटीज डिपॉजिटरी लिमिटेड (एनएसडीएल) / यूटीआईआईटीएसएल (पैन सत्यापन के लिए)", "नियामक निकाय (आरबीआई, सेबी) जैसा कि अनिवार्य है", "मतदाता पहचान पत्र सत्यापन के लिए सरकारी एजेंसियां"],
"retention_period": "धन शोधन निवारण अधिनियम (पीएमएलए) और आरबीआई/सेबी दिशानिर्देशों के अनुसार (आमतौर पर खाता बंद होने के 5-8 साल बाद)।",
"is_mandatory_for_service": true,
"is_sensitive": true
},
{
"id": "purpose_credit_decisioning",
"name": "क्रेडिट निर्णय लेने और जोखिम मूल्यांकन",

```
          "description": "हमारी उधार नीतियों के अनुसार, आपकी क्रेडिट योग्यता और वित्तीय उत्पादों (जैसे, ऋण, क्रेडिट कार्ड) के लिए पात्रता का आकलन करने के लिए आपके क्रेडिट ब्यूरो रिपोर्ट और बैंक स्टेटमेंट का विश्लेषण करके।",
          "legal_basis": "स्पष्ट सहमति (डीपीडीपी अधिनियम, धारा 6(1)(a)) और ऋण/क्रेडिट आवेदनों के लिए संविदात्मक आवश्यकता (डीपीडीपी अधिनियम, धारा 7(b))।",
          "data_categories_involved": ["credit_bureau_report", "bank_statements", "financial_transaction_data", "income_details", "loan_account_details"],
          "recipients_or_third_parties": ["क्रेडिट सूचना कंपनियां (जैसे, सिबिल, एक्सपेरियन, इक्विफैक्स)", "बैंक एपीआई (आपके प्रमाणीकरण के साथ स्टेटमेंट प्राप्त करने के लिए)", "आंतरिक जोखिम मूल्यांकन टीमें"],
          "retention_period": "उधार देने वाली संस्थाओं के लिए आरबीआई दिशानिर्देशों के अनुसार (जैसे, ऋण बंद/समाधान के बाद 7-10 साल)।",
          "is_mandatory_for_service": true,
          "is_sensitive": true
        },
        {
          "id": "purpose_service_delivery",
          "name": "मुख्य सेवा वितरण और खाता प्रबंधन",
          "description": "आपके वित्तीय उत्पाद आवेदनों को संसाधित करने, आपके खाते का प्रबंधन करने, लेनदेन निष्पादित करने और आवश्यक कार्यक्षमताएं प्रदान करने के लिए। यह हमारी सेवाओं के लिए मौलिक है।",
          "legal_basis": "संविदात्मक आवश्यकता (डीपीडीपी अधिनियम, धारा 7(b))",
          "data_categories_involved": ["contact_information", "payment_details", "transaction_history", "loan_account_details"],
          "recipients_or_third_parties": ["भुगतान गेटवे", "बैंकिंग भागीदार"],
          "retention_period": "संविदात्मक दायित्वों और नियामक वित्तीय रिकॉर्ड-कीपिंग आवश्यकताओं के अनुसार।",
          "is_mandatory_for_service": true,
          "is_sensitive": false
        },
        {
```

```
"id": "purpose_customer_support",
"name": "ग्राहक सहायता और संचार",
"description": "आपकी पूछताछ का
```