Strategic Architecture for India's First Intelligent Loan Document Analyzer: A Cloud-Native Approach on Azure

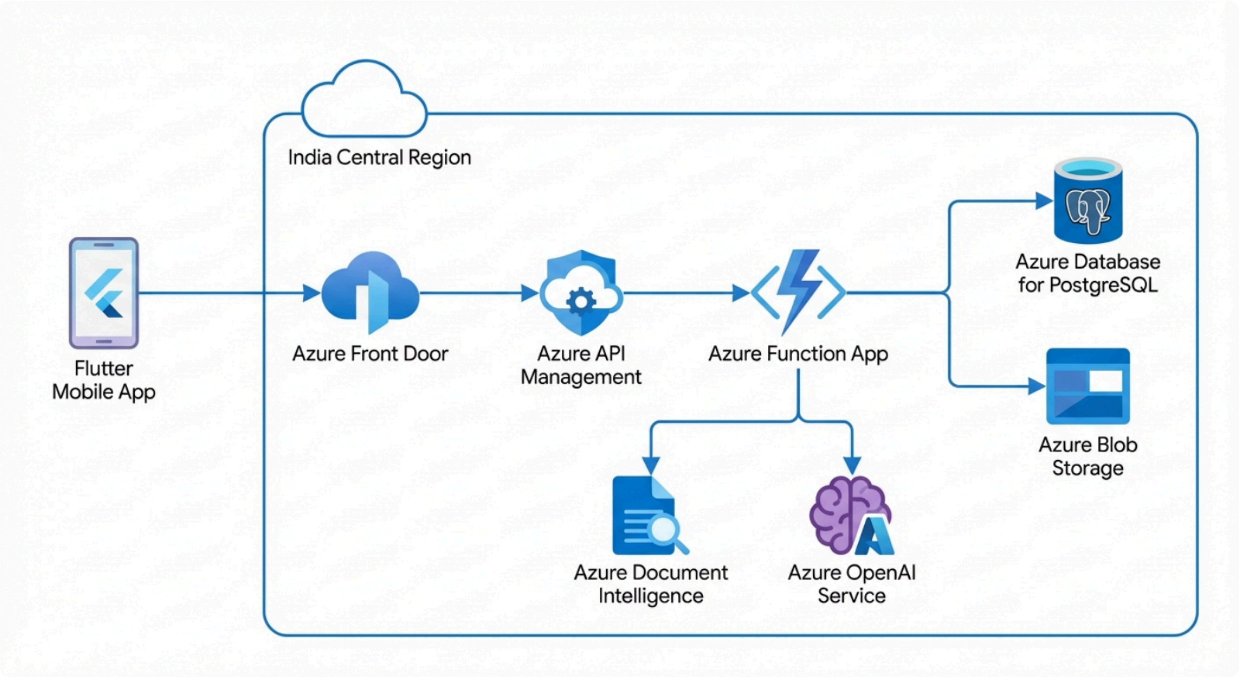
1. Executive Summary

The Indian financial services sector stands at a pivotal juncture. While the digitization of payments via the Unified Payments Interface (UPI) has been revolutionary, the digitization of credit—specifically the underwriting process—remains tethered to legacy workflows involving unstructured data. As loan volumes surge, driven by increasing internet penetration and the aggressive expansion of fintech into tier-2 and tier-3 cities, the traditional reliance on manual verification of documents such as bank statements, Income Tax Returns (ITR), and Know Your Customer (KYC) records has become an unsustainable bottleneck. This report presents a comprehensive, production-grade architectural blueprint for building India's first fully automated, AI-driven Loan Document Scanner and Analyzer on Microsoft Azure.

The proposed solution represents a convergence of cutting-edge cloud-native patterns, specific regional optimizations, and advanced Artificial Intelligence. It leverages **Azure AI Document Intelligence** for high-fidelity structural extraction and **Azure OpenAI Service (GPT-4o)** for semantic reasoning, orchestrated within a secure, scalable ecosystem designed to meet the rigorous standards of the **Digital Personal Data Protection (DPDP) Act, 2023**. Unlike generic global architectures, this design explicitly addresses Indian market constraints, such as the necessity for offline-first mobile capabilities to handle intermittent connectivity and strict data residency mandates that confine all financial data processing within Indian borders.

Key strategic decisions detailed in this report include the adoption of **Azure Database for PostgreSQL Flexible Server with pgvector** as a unified transactional and vector store to optimize cost and performance, and a robust **hybrid connectivity architecture** utilizing Flutter and Drift. Furthermore, the report provides a granular analysis of regional infrastructure availability, specifically navigating the capacity constraints of the South India region to ensure high availability for mission-critical AI workloads.

End-to-End Azure Architecture for Loan Document Analysis



The architecture flows from the Flutter mobile client through Azure Front Door and APIM to a microservices layer hosted on Azure Functions. Data persists in PostgreSQL and Blob Storage, while intelligence is derived via Document Intelligence and Azure OpenAI.

---

2. The Context: Digital Public Infrastructure and the Unstructured Data Challenge

2.1 The Indian Fintech Landscape

India's financial ecosystem is unique in its duality. On one hand, it possesses the "India Stack"—a world-class set of open APIs for identity (Aadhaar), payments (UPI), and data sharing (Account Aggregator). On the other, the "last mile" of credit underwriting involves a chaotic mix of paper-based and digital documents. A typical home loan application might involve a digitally signed e-statement, a scanned JPEG of a property deed, and a grainy photo of a handwritten rent agreement.

Processing this heterogeneous mix requires more than simple digitization; it demands intelligent analysis. The system must not only read the text but understand the context—distinguishing between a "salary credit" and a "friend's transfer," or verifying that the "Pan Number" on a tax return matches the applicant's KYC record. This capability—moving from OCR to "Document Understanding"—is the core value proposition of the proposed architecture.

2.2 The "Iron Triangle" of Requirements

Any solution built for this market must satisfy three competing demands:

1. **Scale and Burstability:** Loan applications are not linear. They spike during festival seasons (Diwali, Dhanteras) and month-ends. The infrastructure must auto-scale to handle terabytes of uploads without human intervention.
2. **Resilience in Connectivity:** A significant portion of loan sourcing happens in semi-urban and rural India, where 4G connectivity can be spotty. The application cannot fail simply because the internet dropped during a document upload.
3. **Regulatory Compliance:** With the DPDP Act 2023, the handling of personal financial data carries strict liability. Data residency is not just a preference; it is a legal requirement.

---

3. Regional Strategy and Infrastructure Planning

Deploying cloud resources for a regulated Indian financial entity requires a meticulous approach to region selection. Azure offers three regions in India: Central (Pune), South (Chennai), and West (Mumbai). However, "availability" is a nuanced concept involving not just the data center existence but the specific services and model quotas present in each.

3.1 Primary Region Selection: Central India (Pune)

For this workload, **Central India** is the designated Primary Region.

- **Availability Zones (AZs):** Central India supports Availability Zones, which are physically separate datacenters within the same region. Deploying critical services (Database, Compute) across zones ensures that the system can survive a localized failure (e.g., a power outage or fire in one building) with zero data loss.
- **Service Maturity:** It is the most mature region for AI services, offering the widest range of SKUs for Azure OpenAI and Document Intelligence.

3.2 The South India Constraint: A Critical Advisory

While **South India (Chennai)** is the natural choice for a secondary/DR region due to its geographic distance from Pune, architects must be aware of specific constraints regarding Generative AI models.

- **GPT-4o Availability:** As of late 2024 and early 2025, the South India region has faced intermittent quota restrictions for high-end models like GPT-4o.[1] Attempts to deploy new GPT-4o resources in South India often result in "Quota Exceeded" errors or forced redirection to other regions, which would violate data residency if not managed carefully.

- **Architectural Implication:** Consequently, the architecture strictly limits South India's role to **Storage Replication (GRS)** and **Database Read Replicas**. It does *not* host active inference endpoints for GPT-4o. If the primary region fails, the Disaster Recovery (DR) plan involves failover to pre-provisioned compute resources in South India, but with the understanding that AI inference might need to be routed to Central India (if accessible) or degrade gracefully to older models (like GPT-3.5-Turbo or GPT-4o-mini) that have better availability in the secondary region.[1]

3.3 Data Residency and Sovereignty Implementation

The architecture adheres to a strict "India-Only" data boundary.

- **Inference guarantees:** When deploying Azure OpenAI resources, selecting the **"Regional"** deployment option (specifically referencing Central India) is mandatory. This contractually ensures that the model inference—the processing of prompts and generation of completions—occurs physically within the selected region.[2] This is distinct from "Global" deployments, which might route traffic to the US or Europe for processing efficiency, a scenario that is legally impermissible for Indian banking data.
- **Ephemeral Processing:** For services like Azure AI Document Intelligence, the data (document images) sent for analysis is processed in memory or ephemeral storage within the region and is not persisted by Microsoft after the API response is returned, aligning with the "Data Minimization" principle of the DPDP Act.[3]

---

4. The Ingestion Layer: Mobile and API Gateway

The entry point for the loan document analyzer is a dual-interface system: a mobile application for field agents/end-users and a REST API for integration with existing Loan Origination Systems (LOS).

4.1 Offline-First Mobile Architecture with Flutter

The mobile application is the primary ingestion channel. Given the target demographic—field agents in tier-2/3 cities—connectivity cannot be guaranteed. The architecture employs an **Offline-First** pattern using **Flutter**.

4.1.1 Local Persistence Layer: Drift

To manage offline state, the application utilizes **Drift**, a reactive persistence library for Flutter and Dart, built on top of SQLite.[4] Unlike simple key-value stores (like SharedPreferences), Drift provides a robust, type-safe relational database on the device.

- **Schema Mirroring:** The local SQLite database schema mirrors the backend PostgreSQL schema for core entities: LoanApplication, DocumentImage, and SyncQueue.
- **Transaction Safety:** Drift supports transactions, ensuring that if a user saves a loan application with five documents, either all records are saved locally, or none are,

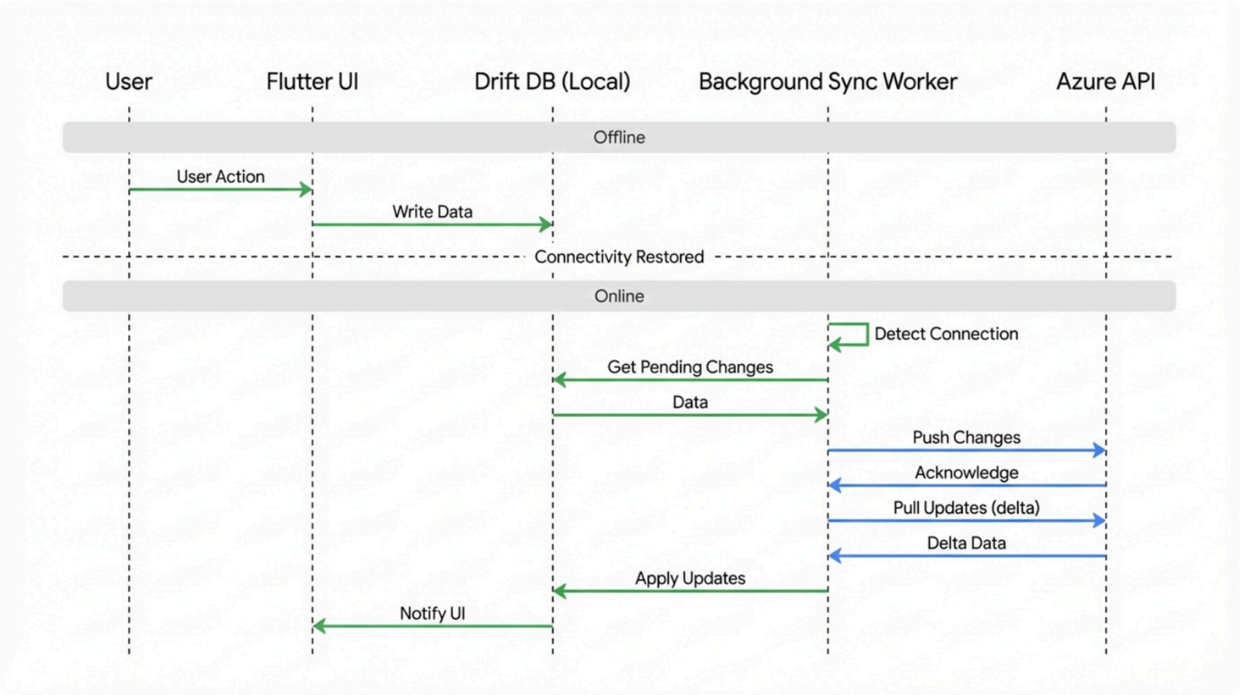
preventing data corruption.[5]

4.1.2 The "Capture Now, Sync Later" Pattern

The user experience is decoupled from network status.

1. **Capture:** When the user scans a document, the image is compressed (JPEG, ~1MB) and stored in the device's secure file system.
2. **Queueing:** A record is inserted into the SyncQueue table with a status of PENDING and a local timestamp.
3. **Connectivity Monitoring:** The app utilizes the connectivity_plus package to listen for network state changes (e.g., transition from none to mobile or wifi).[4]
4. **Background Synchronization:** A background worker (using workmanager) is triggered upon reconnection. It reads the SyncQueue, uploads the images to Azure Blob Storage via SAS tokens, and posts the metadata to the API.

Resilient Mobile Synchronization Pattern



The sequence details the 'Capture Now, Sync Later' pattern. User actions write immediately to the local Drift database. A background worker monitors connectivity, pushing pending changes to Azure Functions and pulling updates via a delta-sync using 'last_modified' timestamps.

4.1.3 Delta Sync and Conflict Resolution

To minimize bandwidth usage—a critical cost factor for users on metered mobile data—the app implements a **Delta Sync** strategy.

- **Logical Clocks:** The server maintains a last_modified timestamp for every record. The client stores the timestamp of its last successful sync.[6]
- **Pull Strategy:** On sync, the client requests GET /sync?since={last_sync_timestamp}. The server returns only records modified after that time.
- **Conflict Handling:** For document uploads, conflicts are rare (append-only). However, for edits to loan details, a "Server Wins" policy is enforced to ensure the central ledger remains the source of truth.

4.2 The Secure Gateway: Azure API Management (APIM)

All backend APIs are shielded by **Azure API Management**. This is not just a proxy; it is a security enforcement point.

- **IP Filtering:** Custom policies restrict access to known IP ranges if the app is used within corporate branches.
- **Rate Limiting:** To prevent cost overruns from OCR usage, throttling policies limit the number of calls per user/key (e.g., 100 docs/minute).
- **Token Validation:** APIM validates the JSON Web Tokens (JWT) issued by Entra ID before the request ever reaches the backend compute, offloading CPU cycles from the microservices.

---

5. The Core Processing Engine: Azure Functions & Orchestration

The backend logic is hosted on **Azure Functions**, utilizing the **Premium Plan** to handle the "bursty" nature of document processing workloads (e.g., huge spikes during business hours, near-zero traffic at night) while avoiding the latency of "cold starts" associated with the Consumption plan.[7]

5.1 Asynchronous Orchestration with Durable Functions

Analyzing a loan document is a multi-step, long-running process. It involves uploading a file, scanning it for viruses, running OCR (which can take 5-10 seconds for large PDFs), classifying the document, extracting data, and validating it. A standard synchronous HTTP request would likely time out or block the client.

To solve this, we implement the **Async Request-Reply** pattern using **Azure Durable Functions**.[8]

5.1.1 The Orchestrator Workflow

The workflow is defined in code (C# or Python), providing a clear, versionable definition of the business process.

1. **Ingestion Trigger:** The client uploads the document to a "Raw" Blob Storage container. This event triggers the Durable Function Orchestrator.
2. **Activity 1 - Validation:** The file is validated (virus scan, file type check).
3. **Activity 2 - Intelligence:** The orchestrator calls the **Document Intelligence Activity**. This activity submits the document to the Azure AI service and polls for completion.
4. **Activity 3 - Semantic Analysis:** The raw JSON from OCR is passed to the **OpenAI Activity**, which uses GPT-4o to extract specific entities (PAN, Salary, Obligations).
5. **Activity 4 - Vectorization:** The extracted text is sent to the **Embedding Activity**, which generates vector embeddings.
6. **Activity 5 - Persistence:** All data—structured JSON and vectors—is written to PostgreSQL.

5.1.2 Status Polling

Upon triggering the process, the API returns a 202 Accepted response with a status_query_GetUri location header. The mobile client polls this endpoint to update the UI progress bar (e.g., "Scanning...", "Analyzing...", "Complete"), ensuring a responsive user experience without holding an open connection.[8]

---

6. Document Intelligence: The Vision Layer

The foundational capability of this system is the accurate extraction of text and structure from images and PDFs. **Azure AI Document Intelligence (v4.0)** is the service of choice, specifically utilizing the **Layout model**.[9]

6.1 Layout Analysis and Table Extraction

Loan documents are replete with tables—bank statements, salary slips, and tax returns. The Layout model excels at identifying table structures, including merged cells and headers, which are common in Indian financial documents.[10]

- **Handling Indian Languages:** The system must robustly support documents in English and Hindi, as well as mixed-script documents (English headers with Hindi content). Azure Document Intelligence's universal models are deep-learning-based and auto-detect languages, including Devanagari scripts, without requiring explicit language codes.[9] This is vital for processing documents from public sector banks in Hindi-speaking belts.
- **Table Limitations:** While the Layout model is powerful, it has known limitations with complex, nested tables or uneven rows often found in scanned copies of older documents.[11]
- **Mitigation Strategy:** For recurring document types with fixed formats (e.g., standard Form 16s), a **Custom Neural Model** should be trained. This requires a dataset of just

5-50 documents but significantly improves accuracy over the generic Layout model for specific templates.[12]

6.2 Handling Handwriting and Signatures

Loan documents often contain handwritten notes or signatures. The read model, which underpins the Layout model, effectively extracts handwritten text in English and supported Indian languages.[9] However, verifying the *presence* of a signature versus verifying the *identity* of the signer are distinct. This system focuses on presence detection (using the selection marks and signature detection capabilities of the newer API versions).[13]

---

7. Intelligent Analysis: The Generative AI Layer

Once text and structure are extracted, the raw JSON output from Document Intelligence is often too verbose for direct business logic. **Azure OpenAI Service** is employed to synthesize this data.

7.1 Model Selection: GPT-4o

GPT-4o is selected for its multimodal capabilities and superior reasoning speed.[14]

- **Role:** It acts as the "Analyzer." It takes the raw OCR output (and potentially the image itself for multimodal verification) and performs:
  - **Named Entity Recognition (NER):** Extracting Applicant Name, PAN Number, Aadhaar Number.
  - **Financial Computation:** Calculating average monthly balance from a list of transactions extracted from a bank statement.
  - **Cross-Verification:** Comparing the name on the IT Return with the name on the Bank Statement to flag discrepancies.

7.2 Retrieval-Augmented Generation (RAG) Architecture

To analyze a loan application against the bank's massive corpus of policy documents (e.g., "RBI Guidelines 2025", "Internal Risk Policy v4"), a RAG pattern is essential.

- **Chunking:** Policy documents are chunked into manageable segments.
- **Embedding: Azure OpenAI text-embedding-3-small** model is used to create vector representations of these chunks.[15]
- **Retrieval:** When the system analyzes a document (e.g., "Is this agricultural income taxable?"), it queries the vector store for relevant policy clauses and feeds them to GPT-4o for a grounded answer.

---

8. Data Persistence: The "PGVector" Advantage

A critical architectural decision is the choice of the vector store. While Azure AI Search is a robust, feature-rich option [16], this architecture recommends **Azure Database for PostgreSQL Flexible Server** with the pgvector extension.

8.1 Why PostgreSQL + pgvector?

1. **Unified Architecture:** It allows storing relational business data (loan IDs, user profiles, transaction logs) alongside high-dimensional vector data (policy embeddings, document embeddings) in a single database instance.[15] This reduces architectural complexity and latency associated with managing two separate data stores.
2. **Cost Efficiency:** For a startup or new product line, the cost of a dedicated Azure AI Search instance (especially at tiers supporting semantic ranking) can be high. PostgreSQL Flexible Server offers a more granular cost curve.[15]
3. **Relational Filtering:** pgvector enables hybrid queries—performing vector similarity searches that are pre-filtered by standard relational criteria (e.g., "Find similar loan fraud cases, but only from the 'Maharashtra' region"). This is natively efficient in SQL.[17]

8.2 Implementation Details

The vector extension must be explicitly enabled (CREATE EXTENSION vector;).[17] Vectors generated by the OpenAI embedding model (1,536 dimensions for text-embedding-3-small) are stored in a column of type vector(1536). Indexing is handled via HNSW (Hierarchical Navigable Small World) indexes for performance optimization.[18]

Strategic Comparison: Azure AI Search vs. PostgreSQL + pgvector

FEATURE / DIMENSION	AZURE AI SEARCH	POSTGRESQL + PGVECTOR RECOMMENDED
Primary Use Case	**Dedicated RAG & Search** Designed as the "de facto RAG buddy" with native search capabilities.	**Unified Data Platform** Ideal for hybrid apps where embeddings are part of a broader data model.
Vector Capability	**Native + Semantic Ranking** Built-in vector search with proprietary semantic ranking features.	**Open-Source Extension** Uses `pgvector` for similarity search (Euclidean, Cosine) via standard SQL.
Integration Complexity	⚠ **External Embeddings** Requires external generation (e.g., Azure OpenAI) before indexing.	✓ **In-Database Generation** Can generate embeddings directly via SQL using `azure_ai` extension.
Hybrid Search	✓ **First-Class Feature** Natively combines full-text, vector search, and reciprocal rank fusion.	● **Via Relational Joins** Combines structured filters and semantic search in standard SQL queries.
Infrastructure & Cost	**Separate Service** Adds architectural overhead and separate cost model.	✓ **Unified Infrastructure** No new APIs or servers needed; reduces architectural overhead.

While Azure AI Search offers advanced features like semantic ranking out-of-the-box, PostgreSQL with pgvector provides a unified, cost-effective solution ideal for reducing architectural overhead in the initial production phase.

Data sources: Visual Studio Magazine, Microsoft Learn (pgvector), Microsoft Learn (Vector Search), Medium

9. Security and Identity Management

Security is paramount, particularly given the sensitivity of financial data.

9.1 Identity: Entra External ID

For customer authentication (loan applicants), the legacy **Azure AD B2C** is being superseded by **Microsoft Entra External ID**.[19]

- **Pricing:** The first 50,000 Monthly Active Users (MAUs) are free, which is highly advantageous for a new platform scaling in India.[20] Beyond this, the cost is approximately ₹2.50 ($0.03) per MAU, making it cost-effective.[21]
- **Features:** It supports modern authentication flows, social logins, and branding customization, essential for a consumer-facing app.

9.2 Network Security

- **Private Endpoints:** All PaaS services (PostgreSQL, Storage, OpenAI, Document Intelligence) are accessed via **Private Links**. This ensures that traffic between the Azure Functions and these services travels exclusively over the Microsoft backbone network, never traversing the public internet.[22]
- **App Service Environment:** The Function App is deployed in a VNET-integrated subnet, restricted by Network Security Groups (NSGs).

---

10. Compliance: Navigating the DPDP Act 2023

The **Digital Personal Data Protection (DPDP) Act, 2023** fundamentally shapes the architecture. The system acts as a **Data Fiduciary**, collecting data from **Data Principals** (loan applicants).[23]

10.1 Consent Architecture

A dedicated "Consent Manager" module is required. Before any document is scanned, the mobile app must capture explicit, verifiable consent. This consent record—timestamped and versioned—is stored in PostgreSQL.

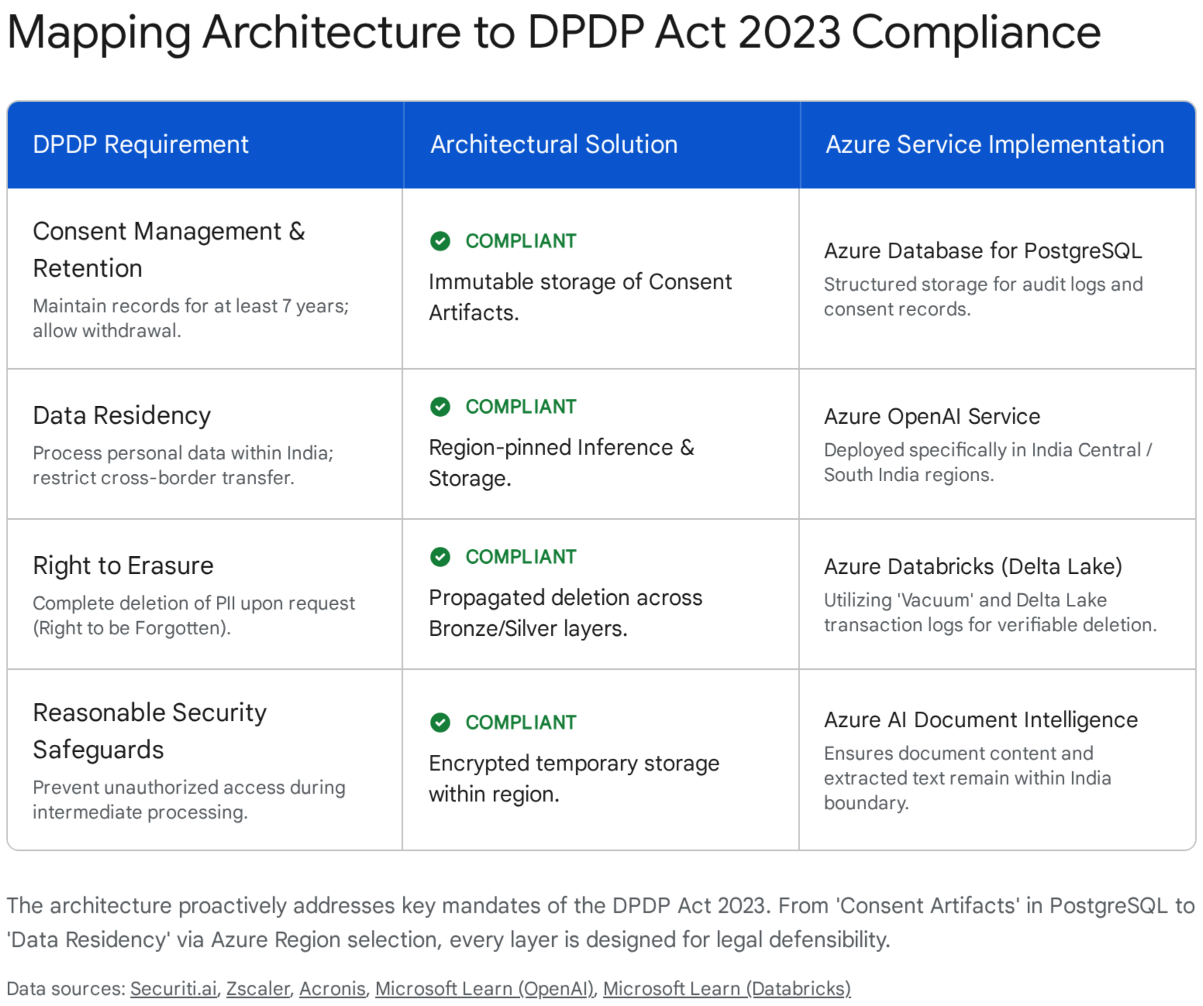
- **Purpose Limitation:** The consent must explicitly state *why* the data is being collected (e.g., "Creditworthiness Assessment").
- **Data Minimization:** The AI analysis layer (GPT-4o) should be instructed (via system prompts) to extract *only* necessary fields and ignore extraneous personal details found in documents.

10.2 Right to Erasure

The architecture must support the "Right to Erasure" (Right to be Forgotten).

- **Implementation:** A "Purge" workflow is implemented in Durable Functions. When a user requests deletion, this function orchestrates the deletion of:
  1. Raw documents in Blob Storage (Bronze/Silver layers).[24]

   2. Metadata and vectors in PostgreSQL.
   3. User identity records in Entra External ID.
- **Audit Trail:** While the personal data is deleted, a log of the *deletion event* is retained for compliance auditing.[25]



Mapping Architecture to DPDP Act 2023 Compliance

DPDP Requirement	Architectural Solution	Azure Service Implementation
Consent Management & Retention Maintain records for at least 7 years; allow withdrawal.	✓ COMPLIANT Immutable storage of Consent Artifacts.	Azure Database for PostgreSQL Structured storage for audit logs and consent records.
Data Residency Process personal data within India; restrict cross-border transfer.	✓ COMPLIANT Region-pinned Inference & Storage.	Azure OpenAI Service Deployed specifically in India Central / South India regions.
Right to Erasure Complete deletion of PII upon request (Right to be Forgotten).	✓ COMPLIANT Propagated deletion across Bronze/Silver layers.	Azure Databricks (Delta Lake) Utilizing 'Vacuum' and Delta Lake transaction logs for verifiable deletion.
Reasonable Security Safeguards Prevent unauthorized access during intermediate processing.	✓ COMPLIANT Encrypted temporary storage within region.	Azure AI Document Intelligence Ensures document content and extracted text remain within India boundary.

The architecture proactively addresses key mandates of the DPDP Act 2023. From 'Consent Artifacts' in PostgreSQL to 'Data Residency' via Azure Region selection, every layer is designed for legal defensibility.

Data sources: Securiti.ai, Zscaler, Acronis, Microsoft Learn (OpenAI), Microsoft Learn (Databricks)

11. Infrastructure as Code (IaC)

To ensure reproducibility and eliminate configuration drift, the entire infrastructure is defined as code. **Bicep** is chosen over Terraform for this specific Azure-native architecture.

- **Reasoning:** Bicep offers day-zero support for new Azure features (like specific OpenAI model versions or new Document Intelligence tiers) and deeper integration with Azure identity management for assigning RBAC roles.[26] While Terraform is excellent for multi-cloud, a strictly Azure-based solution benefits from Bicep's streamlined syntax and state management (which doesn't require a separate state file).

---

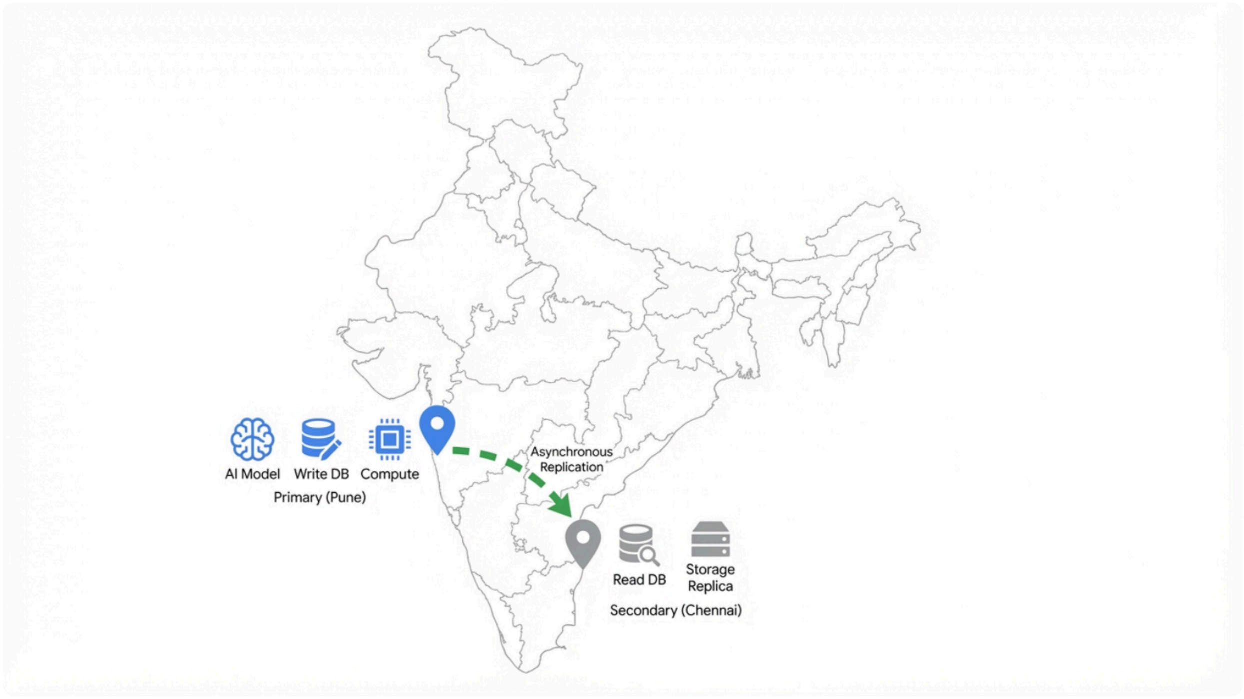
12. Cost Analysis and Optimization

Understanding the running costs is vital for business viability.

12.1 Key Cost Drivers

- **Azure OpenAI:** Pricing is token-based. A standard loan application (approx. 2,000 words analysis) might consume ~3k input tokens and ~1k output tokens. With GPT-4o, this can add up.
  - *Optimization:* Use **GPT-4o-mini** for routine extraction tasks (NER) and reserve **GPT-4o** only for complex reasoning (risk analysis).[2]
- **Document Intelligence:** The Layout model costs $10 per 1,000 pages.[28]
  - *Optimization:* Pre-processing images on the mobile device (cropping, grayscale) to reduce file size and potential re-scans.
- **Storage:** Tiered storage is essential. Documents should move from "Hot" to "Cool" immediately after processing, and to "Archive" after 90 days, significantly reducing costs.[24]

Data Residency & Disaster Recovery Strategy (India)



Primary operations occur in Central India (Pune) to ensure low latency and model availability. Data is asynchronously replicated to South India (Chennai) for disaster recovery, strictly adhering to geo-fencing requirements.

---

13. Conclusion

This architecture represents a sophisticated synthesis of cloud-native patterns and AI capabilities, rigorously tailored for the Indian financial ecosystem. By strategically combining **Azure AI Document Intelligence** for structure, **GPT-4o** for reasoning, and **PostgreSQL with pgvector** for unified data management, we create a system that is greater than the sum of its parts. The inclusion of offline-first mobile capabilities and strict adherence to India's data residency laws ensures that the solution is not just technically sound, but market-ready.

The path forward involves an iterative deployment: starting with the "Layout" model for broad document coverage, then fine-tuning "Custom Neural Models" for high-volume forms, and finally evolving the RAG layer into a comprehensive "Credit Policy Assistant" for underwriters. This is not just a scanner; it is the first step towards autonomous credit decisioning.

Works cited

1. How to deploy open ai azure got-4o model in south India region? - Microsoft

Learn, accessed on February 6, 2026, https://learn.microsoft.com/en-us/answers/questions/2284203/how-to-deploy-open-ai-azure-got-4o-model-in-south

2. Azure OpenAI Service - Pricing, accessed on February 6, 2026, https://azure.microsoft.com/en-us/pricing/details/azure-openai/
3. Clarification on India-Only Data Residency for Azure OpenAI, Embeddings, and Azure AI Document Intelligence - Microsoft Learn, accessed on February 6, 2026, https://learn.microsoft.com/en-us/answers/questions/5664854/clarification-on-india-only-data-residency-for-azu
4. Bringing Flutter Apps Offline: How Drift and Connectivity Tools Made It Simple - Medium, accessed on February 6, 2026, https://medium.com/@oyempemia/bringing-flutter-apps-offline-how-drift-and-connectivity-tools-made-it-simple-2c6ebd1f3875
5. drift_crdt | Dart package - Pub.dev, accessed on February 6, 2026, https://pub.dev/packages/drift_crdt
6. Backend synchronization - Drift! - Simon Binder, accessed on February 6, 2026, https://drift.simonbinder.eu/examples/server_sync/
7. Architecture Best Practices for Azure Functions - Microsoft Azure Well-Architected Framework, accessed on February 6, 2026, https://learn.microsoft.com/en-us/azure/well-architected/service-guides/azure-functions
8. Asynchronous Request-Reply pattern - Azure Architecture Center | Microsoft Learn, accessed on February 6, 2026, https://learn.microsoft.com/en-us/azure/architecture/patterns/async-request-reply
9. Language and locale support for Read and Layout document analysis - Document Intelligence - Foundry Tools | Microsoft Learn, accessed on February 6, 2026, https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/language-support/ocr?view=doc-intel-4.0.0
10. Document layout analysis - Document Intelligence - Foundry Tools | Microsoft Learn, accessed on February 6, 2026, https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/prebuilt/layout?view=doc-intel-4.0.0
11. Custom extraction model can't improve table detection when Layout model fails, accessed on February 6, 2026, https://learn.microsoft.com/en-us/answers/questions/5532824/custom-extraction-model-can-t-improve-table-detect
12. Azure Document Intelligence in Foundry Tools, accessed on February 6, 2026, https://azure.microsoft.com/en-us/products/ai-foundry/tools/document-intelligence
13. Azure Document Intelligence release history - Azure AI services | Azure Docs, accessed on February 6, 2026, https://docs.azure.cn/en-us/ai-services/document-intelligence/reference/release-history?view=doc-intel-4.0.0
14. Foundry Models sold directly by Azure - Microsoft Learn, accessed on February 6,

2026, https://learn.microsoft.com/en-us/azure/ai-foundry/foundry-models/concepts/models-sold-directly-by-azure?view=foundry-classic

15. Building AI-Powered Applications with Azure Database for PostgreSQL, accessed on February 6, 2026, https://visualstudiomagazine.com/articles/2025/08/07/building-ai-powered-applications-with-azure-database-for-postgresql.aspx
16. Choosing the right Azure Vector Database | by Michael John Peña - Medium, accessed on February 6, 2026, https://medium.com/@mjtpena/choosing-the-right-azure-vector-database-d62a64f30e68
17. Enable and use pgvector in Azure Database for PostgreSQL - Microsoft Learn, accessed on February 6, 2026, https://learn.microsoft.com/en-us/azure/postgresql/extensions/how-to-use-pgvector
18. Choose an Azure Service for Vector Search - Azure Architecture Center | Microsoft Learn, accessed on February 6, 2026, https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/vector-search
19. Confusion around Azure AD B2C vs Microsoft Entra External ID, accessed on February 6, 2026, https://learn.microsoft.com/en-us/answers/questions/1556632/confusion-around-azure-ad-b2c-vs-microsoft-entra-e
20. Microsoft Entra External ID—Pricing | Microsoft Security, accessed on February 6, 2026, https://www.microsoft.com/en-us/security/pricing/microsoft-entra-external-id
21. Microsoft Entra External ID—Pricing | Microsoft Security, accessed on February 6, 2026, https://www.microsoft.com/en-in/security/pricing/microsoft-entra-external-id
22. Microsoft Purview classic data governance best practices for security, accessed on February 6, 2026, https://learn.microsoft.com/en-us/purview/data-gov-classic-security-best-practices
23. Understanding the Digital Personal Data Protection (DPDP) Act: A Comprehensive Guide for Businesses | Zscaler, accessed on February 6, 2026, https://www.zscaler.com/blogs/product-insights/understanding-digital-personal-data-protection-dpdp-act-comprehensive-guide
24. Prepare your data for GDPR compliance - Azure Databricks - Microsoft Learn, accessed on February 6, 2026, https://learn.microsoft.com/en-us/azure/databricks/security/privacy/gdpr-delta
25. Navigating India's Digital Personal Data Protection Act (DPDPA) Rules: A Compliance Guide - Securiti, accessed on February 6, 2026, https://securiti.ai/india-digital-personal-data-protection-act-dpdpa-rules/
26. Bicep vs. Terraform in 2026: Is Multi-Cloud Flexibility Still Worth the Pain? : r/AZURE - Reddit, accessed on February 6, 2026,

https://www.reddit.com/r/AZURE/comments/1po3sgh/bicep_vs_terraform_in_2026_is_multicloud/

27. Bicep or terraform for Azure - Reddit, accessed on February 6, 2026, https://www.reddit.com/r/AZURE/comments/1o5o8ld/bicep_or_terraform_for_azure/
28. Azure Document Intelligence in Foundry Tools pricing, accessed on February 6, 2026, https://azure.microsoft.com/en-us/pricing/details/document-intelligence/