Multi-Tenant Architecture for Banking Systems

This document outlines a robust, scalable, and secure multi-tenant architecture designed for banking systems. It details the interplay between centralised authentication, bank-specific portals, and shared microservices, ensuring data isolation and operational efficiency across diverse financial institutions.

Centralised Authentication with Corexfin Auth Service

At the heart of the Corexfin multi-tenant banking system lies the centralised Authentication Service. This critical component is responsible for issuing JSON Web Tokens (JWTs) to all authenticated users, regardless of their role (administrators, managers, or customers) or the specific bank they are associated with. This centralisation ensures a single, authoritative source for identity management across the entire ecosystem.

The Corexfin Auth Service acts as the gatekeeper, verifying user credentials against a unified identity store. Upon successful authentication, it generates a cryptographically signed JWT. This token encapsulates essential user information and, crucially for a multi-tenant system, includes a **bankId** claim. This claim uniquely identifies the financial institution to which the user belongs (e.g., Hex Bank, UBI, SBI).

Unified User Management

All user types (admins, managers, customers) are managed centrally, streamlining access control.

Single Sign-On (SSO)

Users authenticate once with Corexfin, gaining access to their respective bank portals and services.

Enhanced Security

Consistent security policies and token issuance procedures are applied across all tenants.

JWT Tokens and the bankId Claim

The JWT tokens issued by the Corexfin Auth Service are fundamental to maintaining tenant isolation and secure data access within the multi-tenant architecture. Each JWT is a compact, URL-safe means of representing claims to be transferred between two parties. For this banking system, the most crucial claim is the **bankId**.

This bankId claim serves as a definitive identifier, specifying the specific bank (e.g., Hex Bank, UBI, SBI) to which the authenticated user belongs. This mechanism allows downstream services to immediately identify the user's tenant context without needing to query a separate database, significantly reducing latency and improving performance.

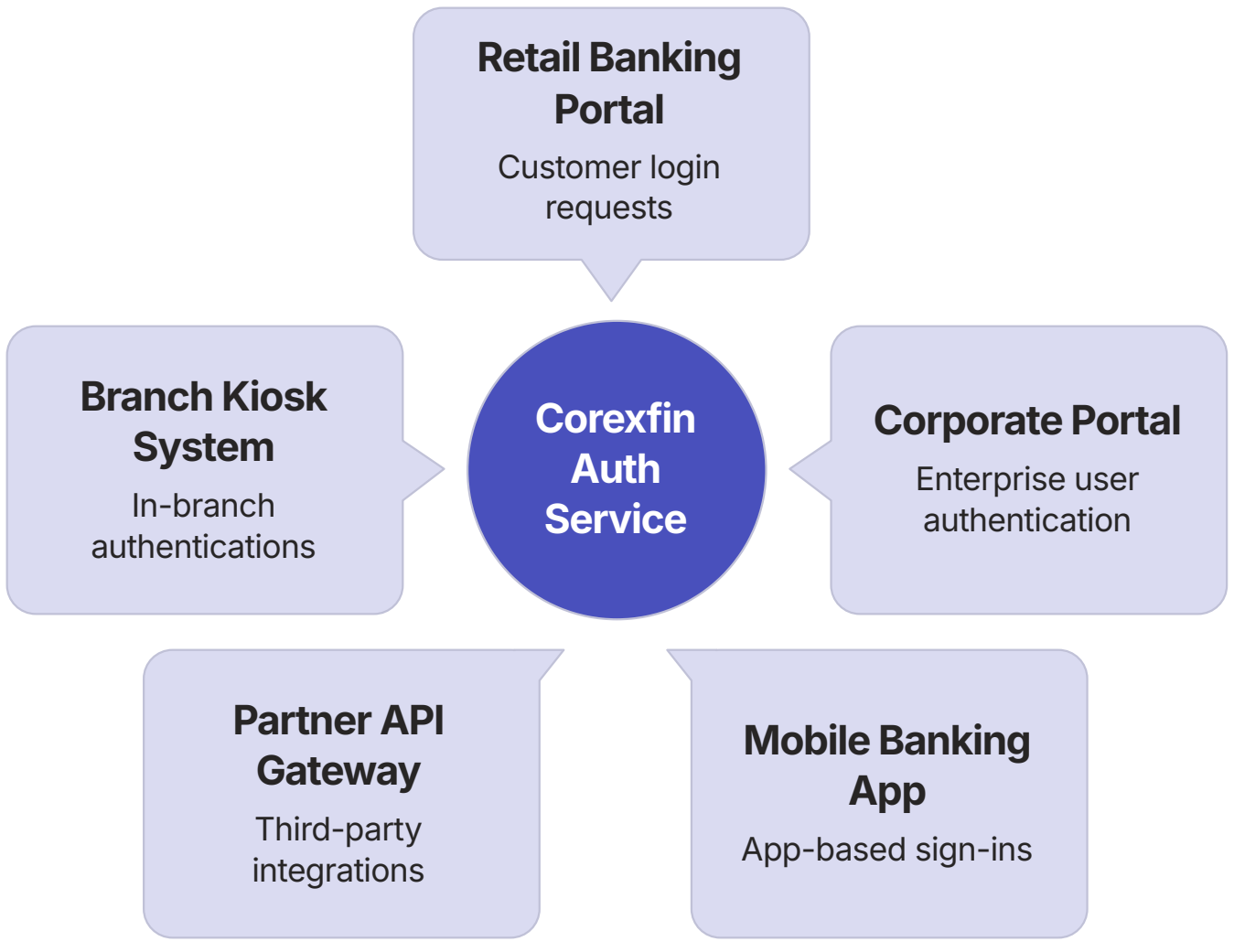
> "The inclusion of a **bankId** claim directly within the JWT token is a powerful architectural decision. It decouples the authentication process from the authorisation enforcement, allowing microservices to be truly stateless and scalable while ensuring strict data segmentation."

The token also includes standard JWT claims such as issuer, audience, subject, expiration, and issued-at times, providing a comprehensive and tamper-proof package of user and session information.

Bank-Specific Portal Services

While authentication and core backend services are centralised, each individual bank (e.g., Hex Bank, UBI, SBI) maintains its own dedicated Portal Service. These Portal Services are designed to provide a highly customised user interface (UI) and implement bank-specific business logic, catering to the unique branding, features, and regulatory requirements of each financial institution.

When a user attempts to log in via their bank's portal, the Portal Service redirects the authentication request to the central Corexfin Auth Service. Upon successful authentication, Corexfin returns a JWT token to the Portal Service. This token is then used by the Portal Service to make requests to the various Corexfin backend microservices on behalf of the user, ensuring that all data interactions are properly authenticated and authorised.



This federated approach allows banks to differentiate their customer experience while leveraging a shared, secure, and robust backend infrastructure. It strikes a balance between customisation and standardisation, offering the best of both worlds for financial service providers.

Shared Corexfin Backend Microservices

Underpinning the entire Corexfin multi-tenant architecture is a suite of backend microservices. These services, including **Account**, **Transaction**, **Loan**, and **Notification** services, are part of the Corexfin platform and are designed to serve all participating banks. This shared model maximises resource utilisation and streamlines development and maintenance efforts.

Each microservice is built to be tenant-agnostic at its core. This means that the fundamental business logic for managing accounts, processing transactions, handling loans, or sending notifications is generic and applicable across all banks. The distinction and data isolation for each tenant are enforced through the JWT's **bankId** claim, which is validated at the service layer.

This architecture promotes efficiency, as enhancements or bug fixes to these core services benefit all tenants simultaneously, without requiring individual deployments or customisations for each bank.

1

Account Service

Manages customer accounts, balances, and statements across all banks.

2

Transaction Service

Processes all financial transactions, ensuring atomic operations and integrity.

3

Loan Service

Handles loan origination, management, and repayment schedules.

4

Notification Service

Delivers real-time alerts and communications to users via various channels.

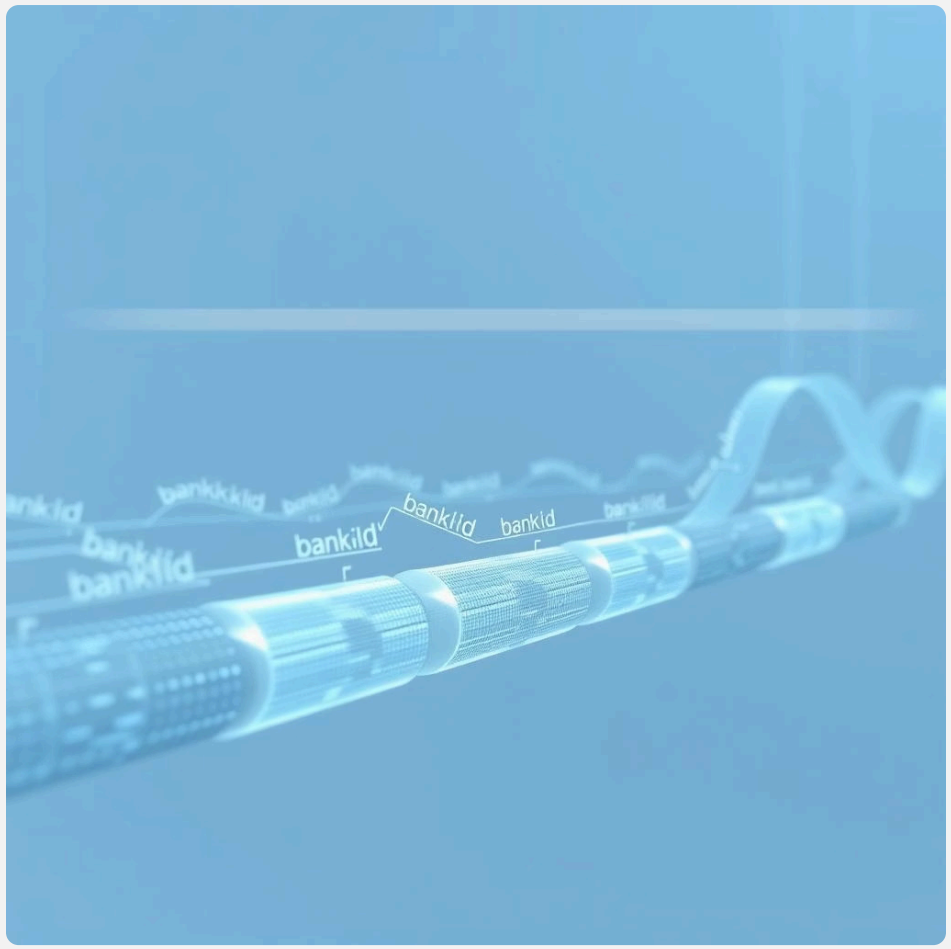
Microservice Data Scoping with bankId

A cornerstone of this multi-tenant design is the mechanism by which microservices ensure data isolation. All backend microservices are engineered to validate incoming JWT tokens and, crucially, to use the **bankId** claim embedded within them to scope data access. This means that a user from Hex Bank can only access Hex Bank's data, even when interacting with a shared microservice instance.

Upon receiving a request, each microservice first verifies the JWT's signature and expiration. Once validated, it extracts the **bankId** claim. All subsequent database queries or data operations performed by that microservice for that particular request are then automatically filtered by this **bankId**. This could involve adding a `WHERE bank_id = '...'` clause to SQL queries, or leveraging tenant-aware features of NoSQL databases.

Benefits of JWT-based Scoping:

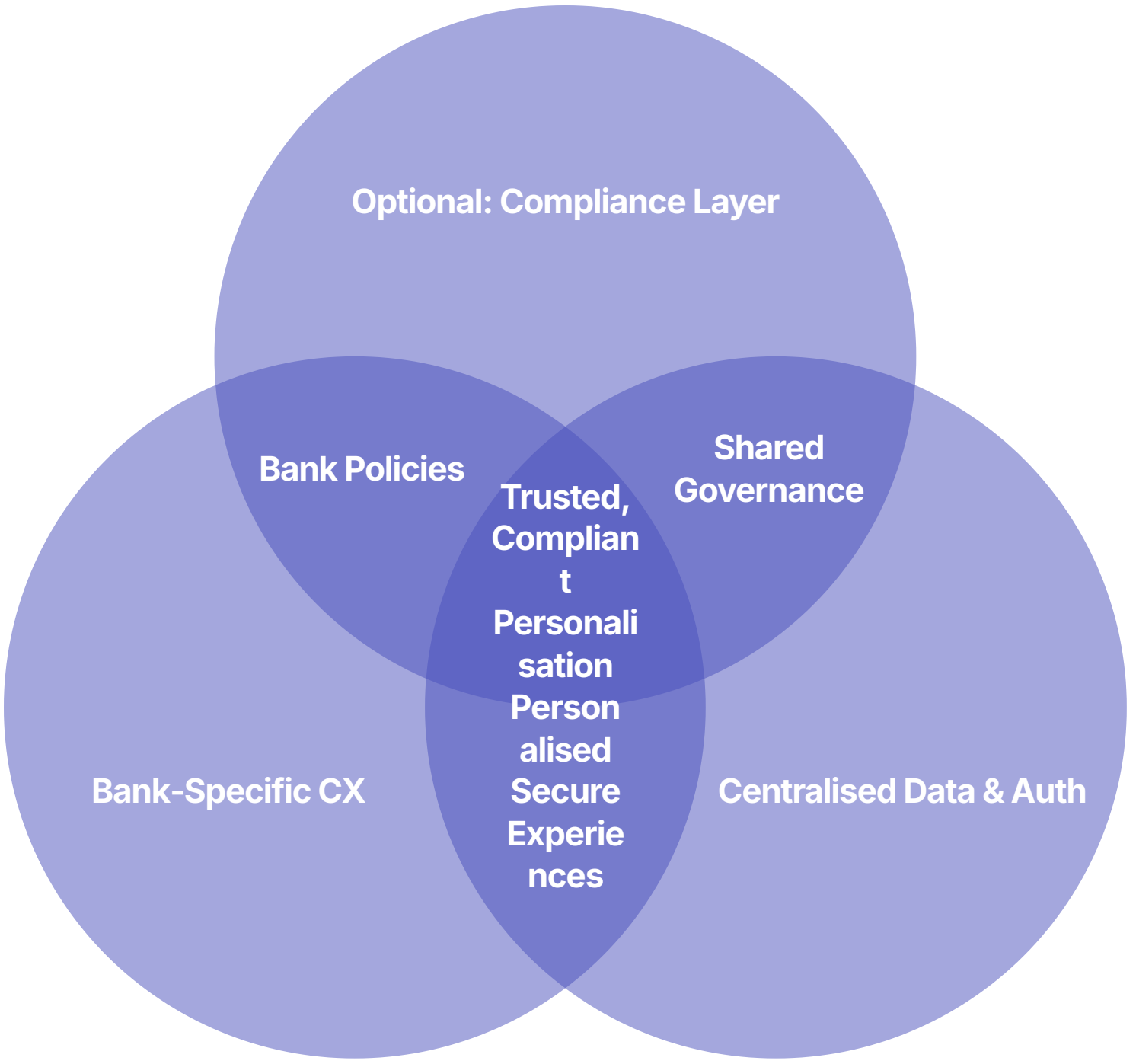
- **Strict Data Isolation:** Ensures that data from one bank is never accidentally exposed to another.
- **Stateless Microservices:** Microservices do not need to maintain session state or tenant context, simplifying their design and scaling.
- **Reduced Latency:** The **bankId** is readily available in the token, avoiding additional lookup calls.
- **Simplified Authorization:** Fine-grained authorisation rules can be applied at the service layer, incorporating the **bankId** as a key parameter.



Customer Experience: Bank-Specific Portals with Centralised Core

From a customer's perspective, their interaction with the Corexfin platform is seamlessly integrated within their specific bank's portal. Users of Hex Bank, for instance, will only ever navigate and interact with the Hex Bank Portal. They are unaware that the underlying data and authentication mechanisms are handled centrally by Corexfin.

This design provides the best of both worlds: a tailored, branded experience for each bank's customer base, coupled with the security, efficiency, and robustness of a shared, enterprise-grade backend infrastructure. The centralisation of data and authentication within Corexfin ensures consistency and reliability across all tenants.



This clear separation of concerns—UI and bank-specific logic on the portal side, and core financial services with authentication on the Corexfin side—allows for independent evolution of both layers. Banks can rapidly update their customer-facing features without impacting the stable, secure backend, and Corexfin can enhance its core services without forcing changes on individual bank portals (beyond API compatibility).

Onboarding New Banks: Simplicity and Speed

One of the significant advantages of this multi-tenant architecture is the simplified and expedited process for onboarding new financial institutions. Adding a new bank to the Corexfin ecosystem primarily involves two key steps, which are significantly less complex than building an entire banking system from scratch:

1. New Portal Service Development:

- Each new bank requires the development of its own dedicated Portal Service.
- This service will handle the bank's unique branding, user interface, and any bank-specific business logic not covered by the generic Corexfin microservices.
- The Portal Service is responsible for integrating with the Corexfin Auth Service for user authentication and then making calls to the shared backend microservices.

2. Corexfin Configuration:

- The new bank's information, including its unique **bankId** and other essential metadata, must be securely configured within the Corexfin platform.
- This configuration ensures that the Corexfin Auth Service can correctly issue JWTs with the appropriate **bankId** claim and that the backend microservices can properly scope data access for the new tenant.

This streamlined onboarding process reduces time-to-market for new banks, making the Corexfin platform an attractive solution for financial institutions looking to quickly leverage modern banking infrastructure.

Benefits of the Multi-Tenant Architecture

This architectural design offers a multitude of benefits, solidifying its position as a robust and future-proof solution for multi-tenant banking systems:

Single Source of Truth

Centralised authentication and shared core microservices establish a single, authoritative source for user identities and financial data, eliminating inconsistencies and simplifying data governance.

Consistent Security

Unified security policies, JWT validation, and the bankId claim ensure consistent and robust data isolation and access control across all tenants.

Stateless & Scalable Microservices

By relying on JWTs for authentication and authorisation context, microservices remain stateless, highly scalable, and resilient, capable of handling growing loads from multiple banks.

Simplified Maintenance

Updates and bug fixes to core microservices benefit all tenants simultaneously, reducing operational overhead and accelerating release cycles.

Rapid Onboarding

The clear separation of concerns enables quick integration of new banks, primarily by developing their unique portal and configuring their tenant details.

Cost Efficiency

Shared infrastructure and development efforts lead to significant cost savings compared to maintaining separate systems for each bank.

Future Enhancements: API Gateway and Observability

While the current architecture is robust, further enhancements can significantly improve its security, performance, and manageability. One critical addition is the introduction of an API Gateway.

API Gateway Integration:

- An API Gateway can centralise JWT validation for all incoming requests before routing them to the respective microservices.
- This offloads token validation logic from individual microservices, reducing boilerplate code and ensuring consistent enforcement of security policies.
- The Gateway can also handle other cross-cutting concerns such as rate limiting, logging, caching, and request/response transformation, further streamlining the microservice layer.

Enhanced Observability:

- Implementing comprehensive logging, monitoring, and tracing solutions across all services and the API Gateway is crucial for understanding system behavior, diagnosing issues, and ensuring performance.
- Distributed tracing, using tools like OpenTelemetry, can provide end-to-end visibility of requests as they traverse multiple services, critical in a distributed microservices environment.

These additions will further solidify the multi-tenant banking architecture, making it even more resilient, secure, and easier to operate at scale.