

Transforming Banking Series

A BIAN Reference Implementation for Cross-Selling Product Offerings

Tech Stack: Open Source Technologies

Author

Dilish Kuruppath | Enterprise Architect, BridgeApps

Executive Summary

The banking landscape is undergoing a profound transformation driven by technological advancements and the ever-evolving expectations of customers. In this era of digitalization, banks face the imperative to adapt, innovate, and provide personalized services to meet the diverse needs of their clientele. The Banking Industry Architecture Network (BIAN) emerges as a pivotal framework, offering a unique and standardized approach to address the complexities of modern banking processes.

This whitepaper presents a comprehensive reference implementation using open-source technologies that helps bankers to understand the principles of BIAN from an application perspective. The focal point of our implementation revolves around offering alternative products to existing banking customers and ensuring a seamless application process. By leveraging a microservices-based architecture, our solution not only adheres to BIAN standards but also demonstrates the agility and scalability required by the banking industry.

Audience

All BIAN Enthusiasts, Enterprise Architects, Technology Leaders (CIO, CTO, CXO), Enterprise Solution (Banking) Architects, Integration Architects, Designers and Developers.

We have attempted to cover details at different levels. *Connect with us to see a Live Demo*

○ What is BIAN ?

BIAN or Banking Industry Architecture Network, provides a standardized framework that, if implemented across the banking ecosystem, enables interoperability, flexibility, and innovation within the bank and across the industry. Our Reference implementation aligns with BIAN APIs and principles, and attempts to showcase how adherence to industry standards enhances efficiency, reduces complexity, and ultimately delivers a superior banking experience.

○ Why the Use Case ?

The chosen use case clearly reflects one of the several real-world challenge faced by banks, i.e engaging existing customers with alternative product offerings. As customer preferences shift and financial landscapes evolve, to stay competitive its imperative for banks to proactively present tailored solutions that meet cutomers' personal needs. Our reference implementation addresses this imperative by facilitating the identification and promotion of alternative products, coupled with a frictionless customer application process.

Reference Implementation Overview

The reference implementation presents an opportunity to get a real life view of a typical BIAN implementation. We have picked 'Alternative Product Offering' use case for a pre-registered or Customer-To- Bank (CTB) scenario. At a high level the use case for a pre-registered user could be depicted as given in fig (a) below:

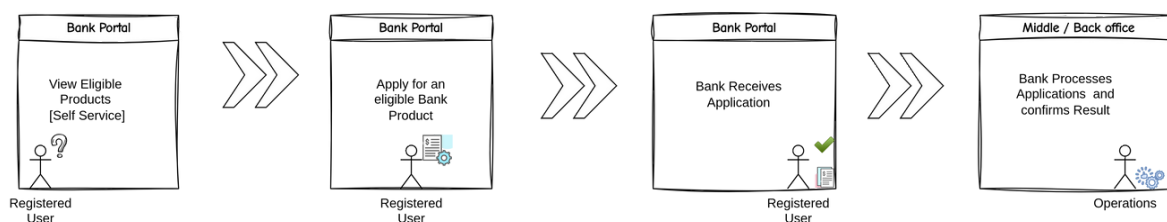


fig (a) - High Level Use case

In this usecase implementation, we brought BIAN APIs to life. The implementation involved identifying functional use case with the Business Capability, narrowing down to the Service Domains and finally identifying the APIs before implementing them using open source tech stack and simulated end systems. Tooling to capture NFRs were also used to showcase an end to end life cycle for such an implementation.

Architecture Principles

1. Microservices driven / BIAN APIs

In the context of a cross-sell use case, microservices were employed to implement various BIAN APIs, ensuring modularity and interoperability. These services collectively enable the end-to-end flow of the customer journey.

3. Open Source Technologies

Only Open source technologies and products were used leading to an ISV agnostic approach. This PoV could be implemented within every bank in their own product and platform stack.

2. Simulated Systems for Capabilities

While building demonstration in this reference implementation, simulated applications for Capabilities (for e.g. core banking systems) were used to showcase BIAN API interactions, for e.g. Product Portal, a Sales Portal, and a Document Management System.

4. End to End Tooling

The use case reference implementation was envisaged to showcase as many of the enterprise capabilities that might be involved in a typical API implementation ecosystem, including sso, monitoring, tracing and development capabilities (e.g. Swagger).

Use Case Overview

Let's delve into the details of the usecase - a typical Cross-Sell opportunity to offer an existing bank customer some alternative products, which match their profile. For the purpose of this reference architecture, we have used the *customer's Credit Rating*, as the sole criteria to identify products suitable for them. In real banking scenarios this may be a combination of multiple business functionalities.

A Registered user (CTB) in the Bank's Portal will be taken through the following screens in post log-in, leading to a manual workflow in the bank after receiving user's application.

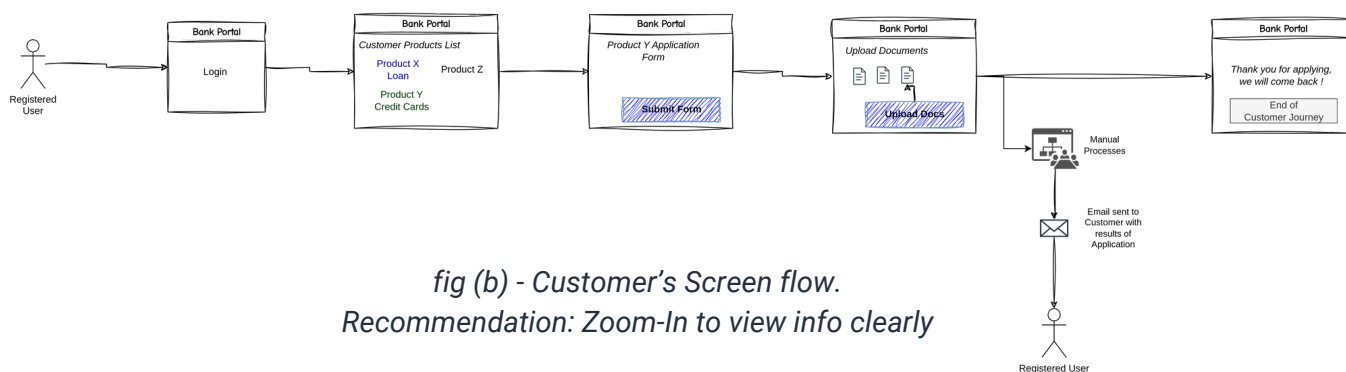


fig (b) - Customer's Screen flow.
Recommendation: Zoom-In to view info clearly

Following are the brief steps in this flow :

1. CTB User logs in into the Bank's Customer Portal.

Authentication / Authorization / SSO has been considered for the purposes of this reference implementation

2. User is offered a series of "Alternative Products" suitable to the User's Profile

Credit Check Rating of the User has been assumed to be the only criteria for identifying these products

3. User chooses a product

4. User is shown an Application form to fill in to submit basic details

5. Further screens are generated to capture documents specific to each user profile.

In a potential Open Banking implementation, documents may be avoided and user qualified via OpenBanking capabilities.

6. Post document uploads, user sees a "thank you for application" page

7. Manual workflow kicks off to process the application

Considered out of scope of this reference implementation

As an initial step, based on the BIAN Business Capabilities list, the ideal business capability for the usecase in context was identified as - Offer Management.

Enterprise Management and Controlling		Product and Service Offering		Enterprise Enabling		Marketing and Sales	
Business Division Management	Fixed Incident Management	Agreement Management	Financial Plan Management	Facility and Equipment Management	Incident Management	Brand Management	
Policy Management	Initiative Management	Financial Instrument Management	Investment Portfolio Management	Human Capital Management	Inquiry Management	Campaign Management	
Business Entity Management	Inventory Management	Issued Device Management	Money Movement Management	Tax Management	Content Management	Event Management	
Risk Management		Order Management	Product Management	Vendor and Supplier Management	Location Management	Market Management	
Finance Management		Trade Finance Management	Trust Management	Competency Management	Time Management	Message Management	
Financial Account Management		Payment Management	Cultural Management	Information Management	Language Management	Loyalty Management	
Tax Management			Intellectual Property Management		Legal Support Management	Lead Management	
Financial Position Management						Offer Management	
						Sales Plan Management	

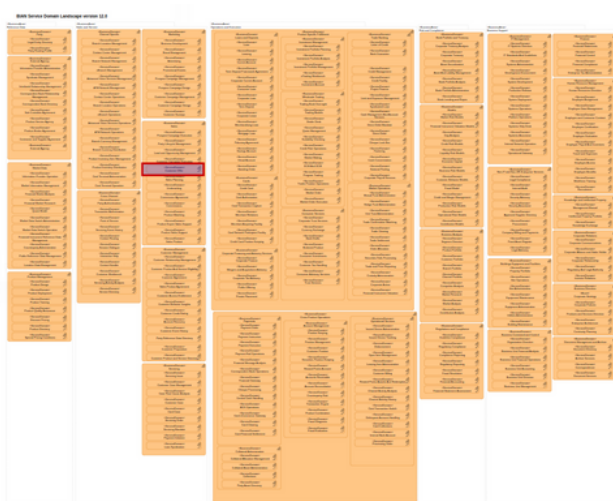


fig(d) Business Capability: Offer Management

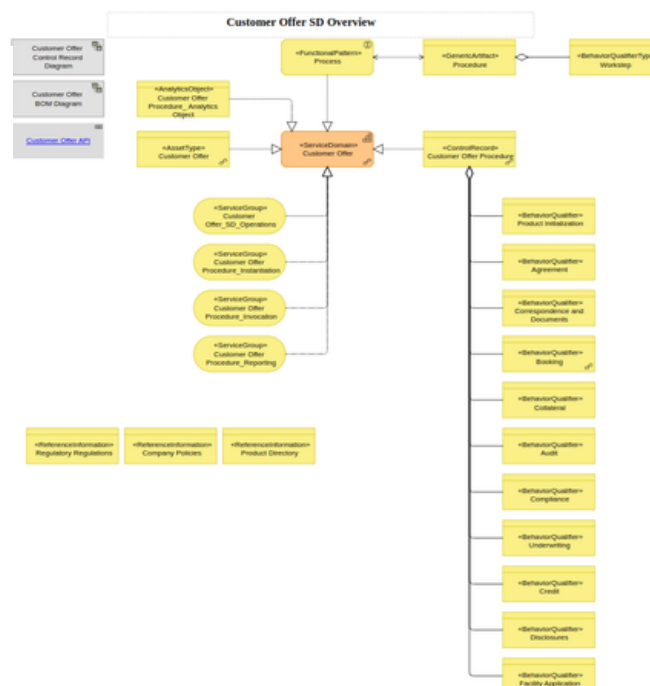
fig(c) Business Capability - Top Level View

“Ability to create, present, and deliver a targeted personalized opportunity intended to create a purposeful engagement, that are delivered to a particular group of customers or prospects, using one or more channels.”

The following Service Domains were identified as part of this reference implementation:



fig(e) All service domains

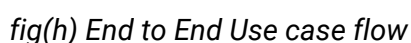


fig(f) Service Domain : Customer Offer

This section delves into the intricacies of our reference implementation's technical design, providing a detailed exploration of the architectural decisions, integration patterns, and system interactions.

The diagram illustrates the transition from monolithic systems to microservices. The top row, labeled 'Systems', shows four large, integrated components: 'Cx Banking Portal', 'Credit Service System', 'Product Portal', and 'Customer Acq System'. The bottom row, labeled 'MicroServices', shows five smaller, decoupled components: 'Credit Rating', 'Product', 'Customer Management', 'Doc Mgr', and 'Security'. Arrows indicate the decomposition of the top systems into the bottom microservices.

The microservices, represented by rounded boxes in the flow diagram below, implements BIAN APIs to facilitate the seamless execution of the customer journey in the selected use case.



Recorded Demo

View the recorded demo of the reference Implementation here, which showcases the end to end Implementation, including data and calls flow, and the reference implementation calls & observability capability, with specific focus on implementing a product Cross sell use case.



Vid : <https://youtu.be/-Vg1bZsIsNk>

Technology Components

API Gateway

All the identified BIAN APIs were implemented on the API Gateway and the events for invocation sent over to the Monitoring Dashboard (Grafana)



With features like load balancing, policy based security enforcement, and protocol translation, the API Gateway streamlines the complexity of microservices architecture, enhancing scalability and maintaining a unified interface for clients. We used KrakenD as the API gateway for this reference Implementation. The API Gateway serves as the central entry point for managing and orchestrating the flow of data between microservices, applications, and external systems. Acting as a traffic cop, it handles requests, performs authentication and authorization, and ensures seamless communication between diverse components.

SSO - Across all Systems

Keycloak, as the central Single Sign-On (SSO), acts as a centralized authentication and authorization solution, seamlessly integrating with all Portals involved (simulated end systems). By providing a secure identity management platform, Keycloak enables users to authenticate once and access multiple services without the need for repeated logins.



Leveraging Keycloak via OpenID Connect ensured a streamlined and secure SSO experience for users across our ecosystem, emphasising on convenience and security.

Monitoring Dashboard

All invocation events on BIAN APIs were captured on a Dashboard for use by Support team. Monitoring dashboard, powered by Grafana, offers insights into API performance:



- Tracks API invocation and response times.
- Analyzes request and response sizes.
- Allows observation of throughput metrics, including average, minimum, and maximum values per API.

This tool provides valuable data to enhance the overall customer journey and can be customized based on the preferences and tooling used by different banks.

API Tracing with Jaeger

Utilizing Jaeger, we trace the execution flow of BIAN APIs hosted in the Kraken-D API Gateway:

- Observes call stacks for each BIAN API.
- Provides a visual representation of the API execution flow.



Jaeger serves as a crucial tool for understanding and optimizing the end-to-end BIAN implementation journey.

Swagger Documentation

Swagger documentation for BIAN APIs, allow for discoverability of all the APIs, by allowing to distribute the API implementations to partners. This allows to understand data schemas and details of API functionalities.



In this reference implementation, we have taken the respective service domain, the components involved and then implemented these APIs to showcase the capability.

Summary

In conclusion, our whitepaper offers a comprehensive exploration of a reference implementation tailored to address the evolving needs of the banking industry. By leveraging the Banking Industry Architecture Network (BIAN) framework and embracing open-source technologies, we've attempted to showcase the agility, scalability, and interoperability achievable within the banking ecosystem, by adopting BIAN.

Through a detailed examination of the use case scenario, technical design, and implementation specifics, we've demonstrated how a typical implementation solution should offer a seamless experience for all parties involved including banks and their customers. From the adoption of microservices architecture to the integration of Single Sign-On (SSO) capabilities with Keycloak, this reference architecture attempts to offer a holistic representation of a typical Production ecosystem. By embracing industry standards, leveraging cutting-edge technologies, and prioritizing customer-centricity, banks can navigate the complexities of modern banking with confidence and drive sustainable growth.

In essence, we believe, this whitepaper would be a good starting point for banks seeking to embrace digital transformation via BIAN adoption.

For more details and Live Demo, contact:

info@bridgeapps.co.uk

+44 (0) 7553 226 553

<https://bridgeapps.co.uk/bian>