PROYECTO

Sistema de Banca por Internet

Byron Vladimir Tirado Ortiz

Enero 2026

1. PROBLEMA

Usted ha sido contratado por una entidad llamada BP como arquitecto de soluciones para diseñar un sistema de banca por internet, en este sistema los usuarios podrán acceder al histórico de sus movimientos, realizar transferencias y pagos entre cuentas propias e interbancarias.

Toda la información referente al cliente se tomará de 2 sistemas, una plataforma Core que contiene información básica de cliente, movimientos, productos y un sistema independiente que complementa la información del cliente cuando los datos se requieren en detalle.

Debido a que la norma exige que los usuarios sean notificados sobre los movimientos realizados, el sistema utilizará sistemas externos o propios de envío de notificaciones, mínimo 2.

Este sistema contará con 2 aplicaciones en el Front, una SPA y una Aplicación móvil desarrollada en un Framework multiplataforma. (Mencione 2 opciones y justifique el porqué de su elección).

Ambas aplicaciones autenticarán a los usuarios mediante un servicio que usa el estándar Oauth2.0, para el cual no requiere implementar toda la lógica, ya que la compañía cuenta con un producto que puede ser configurado para este fin; sin embargo, debe dar recomendaciones sobre cuál es el mejor flujo de autenticación que se debería usar según el estándar.

Tenga en cuenta que el sistema de Onboarding para nuevos clientes en la aplicación móvil usa reconocimiento facial, por tanto, su arquitectura deberá considerarlo como parte del flujo de autorización y autenticación, a partir del Onboarding el nuevo usuario podrá ingresar al sistema mediante usuario y clave, huella o algún otro método especifique alguno de los anteriores dentro de su arquitectura, también puede recomendar herramientas de industria que realicen estas tareas y robustezca su aplicación.

El sistema utiliza una base de datos de auditoría que registra todas las acciones del cliente y cuenta con un mecanismo de persistencia de información para clientes frecuentes, para este caso proponga una alternativa basada en patrones de diseño que relacione los componentes que deberían interactuar para conseguir el objetivo.

Para obtener los datos del cliente el sistema pasa por una capa de integración compuesta por un api Gateway y consume los servicios necesarios de acuerdo con el tipo de transacción, inicialmente usted cuenta con 3 servicios principales, consulta de datos básicos, consulta de movimientos y transferencias que realiza llamados a servicios externos dependiendo del tipo, si considera que debería agregar más servicios para mejorar el rendimiento de su arquitectura o agregar más servicios para mejorar la repuesta de información a sus clientes, es libre de hacerlo.

2. MODELAMIENTO

El modelamiento se lo realizó con el modelo C4, que es un marco para diagramar arquitecturas de software a diferentes niveles de abstracción (Contexto, Contenedores, Componentes, Código), que facilita la comunicación clara entre equipos técnicos y no técnicos, documentando la estructura de un sistema de forma jerárquica y simplificada, desde su visión general hasta los detalles del código.

2.1 Diagrama de Contexto:

Actores Clave

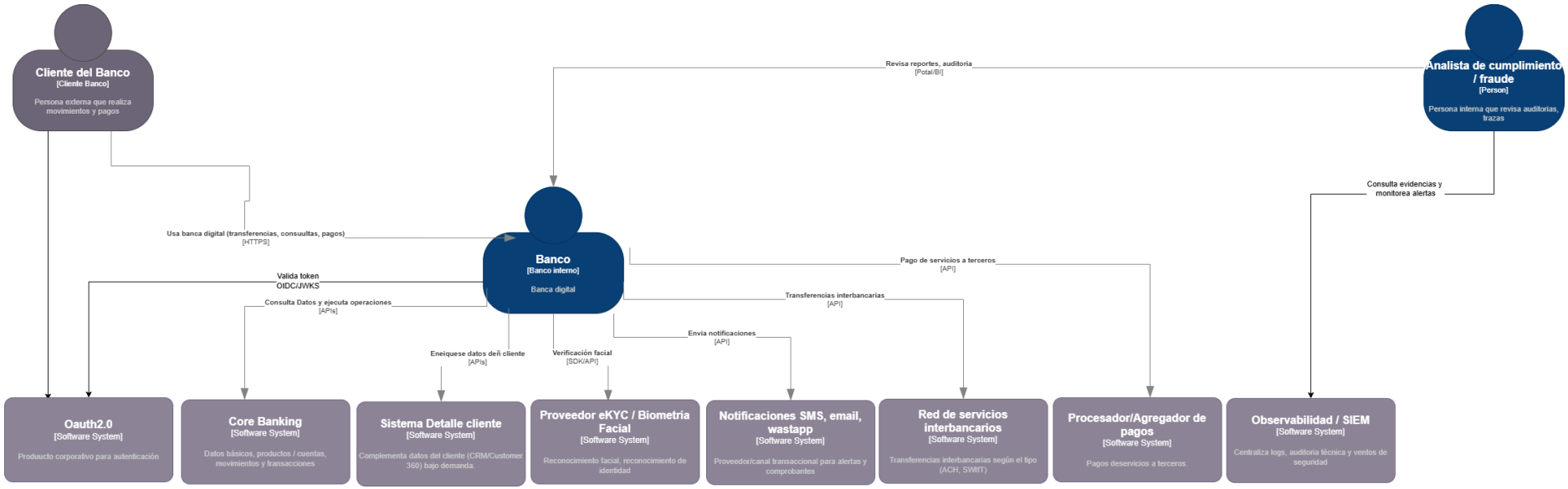
Personas

- Cliente BancoP (Usuario final)
  - Consume Web SPA y App móvil para operar su banca.
  - Requiere notificación de movimientos por norma.

- Analista de Cumplimiento / Fraude (Usuario interno)
  - Consulta trazas/auditoría para investigación y cumplimiento.

Sistemas externos

- Core Banking (Plataforma Core)
  - Fuente de: datos básicos del cliente, productos/cuentas, movimientos.
  - Ejecución/registro de operaciones (transferencias/pagos, según integración).
- Sistema de Detalle de Cliente (CRM/KYC/Customer 360)
  - Complementa información cuando se requiere mayor detalle (contactabilidad, KYC, segmentación, preferencias, etc.).
- IdP corporativo (OAuth2/OIDC)
  - Producto configurable existente para autenticación/autorización.
- Proveedor de biometría facial / eKYC (Onboarding móvil)
  - Reconocimiento facial + liveness detection y validación de identidad (según política).
- Proveedores de Notificaciones
  - Proveedor A: Email / SMS (ej. canal transaccional).
  - Proveedor B: Push (FCM/APNs), WhatsApp u otro canal alterno.
  - Requisito: redundancia/failover.
- Red/Servicios de Transferencias Interbancarias
  - Rails interbancarios (ACH/SPI/SWIFT u operador local).
  - Puede incluir verificación de beneficiario y estado.
- Agregador/Procesador de Pagos
  - Pago de servicios, terceros, recargas, etc.
- Plataforma de Observabilidad / SIEM (recomendado)
  - Recepción de logs, eventos de seguridad, trazas.


Cliente del Banco
[Cliente Banco]
Persona externa que realiza movimientos y pagos
Analista de cumplimiento / fraude
[Person]
Persona interna que revisa auditorias, trazas
Revisa reportes, auditoria
[Potal/BI]
Consulta evidencias y monitorea alertas
Usa banca digital (transferencias, consuultas, pagos)
[HTTPS]
Banco
[Banco interno]
Banca digital
Pago de servicios a terceros
[API]
Valida token
OIDC/JWKS
Transferencias interbancarias
[API]
Consulta Datos y ejecuta operaciones
[APIs]
Envia notificaciones
[API]
Eneiquese datos deñ cliente
[APIs]
Verificación facial
[SDK/API]
Oauth2.0
[Software System]
Produucto corporativo para autenticación
Core Banking
[Software System]
Datos básicos, productos / cuentas, movimientos y transacciones
Sistema Detalle cliente
[Software System]
Complementa datos del cliente (CRM/Customer 360) bajo demanda.
Proveedor eKYC / Biometría Facial
[Software System]
Reconocimiento facial, reconocimiento de identidad
Notificaciones SMS, email, wastapp
[Software System]
Proveedor/canal transaccional para alertas y comprobantes
Red de servicios interbancarios
[Software System]
Transferencias interbancarias según el tipo (ACH, SWIFT)
Procesador/Agregador de pagos
[Software System]
Pagos deservicios a terceros.
Observabilidad / SIEM
[Software System]
Centraliza logs, auditoria técnica y ventos de seguridad

2.2 Diagrama de Contenedores:

Infraestructura Propuesta:

A) Front: SPA + App móvil multiplataforma

App Web:

- React: ecosistema masivo, flexibilidad, buen soporte para micro-frontends si crece por dominios.
- Angular: opinionado, fuerte en enterprise, estructura y convenciones útiles en banca.

App móvil multiplataforma (2 opciones):

Flutter

- Rendimiento alto y UI consistente (motor propio).
- Excelente para experiencias bancarias con UI/animaciones homogéneas.
- Buen control de diseño y componentes reutilizables.

React Native

- Time-to-market rápido si el equipo domina JS/TS.
- Enorme ecosistema de librerías.
- Acceso a capacidades nativas mediante módulos (biometría, almacenamiento seguro, etc.).

Recomendación práctica típica en banca: Flutter cuando priorizas consistencia visual/rendimiento; React Native cuando priorizas velocidad y talento disponible en JS/TS.

B) OAuth2/OIDC: flujo recomendado

Para SPA y móvil (clientes "públicos"), el estándar recomendado hoy es: OAuth 2.0 Authorization Code Flow con PKCE + OpenID Connect

Recomendación adicional (seguridad banca): patrón BFF para SPA
SPA no gestiona tokens directamente; usa un Backend-for-Frontend que mantiene sesión (cookie HttpOnly) y custodia tokens.
Reduce superficie de ataque (XSS/token theft) en canal web.

C) Onboarding móvil con reconocimiento facial y métodos posteriores

Contextualmente, el onboarding introduce un sistema externo (eKYC/biometría) y condiciona el IAM:

Herramientas de industria (referenciales)

eKYC/biometría: Onfido, Jumio, FaceTec, AWS Rekognition, Azure Face (siempre con liveness y controles anti-spoofing).
Autenticación fuerte: FIDO2/WebAuthn (passkeys), motores MFA del IdP corporativo.

D) Auditoría + "persistencia para clientes frecuentes" basada en patrones

A nivel de contexto, esto se materializa como capacidades internas, pero la recomendación de patrones es:

- Audit Trail: arquitectura dirigida por eventos
- Clientes frecuentes / datos de acceso rápido (p.ej. beneficiarios frecuentes, últimos movimientos, perfil resumido):

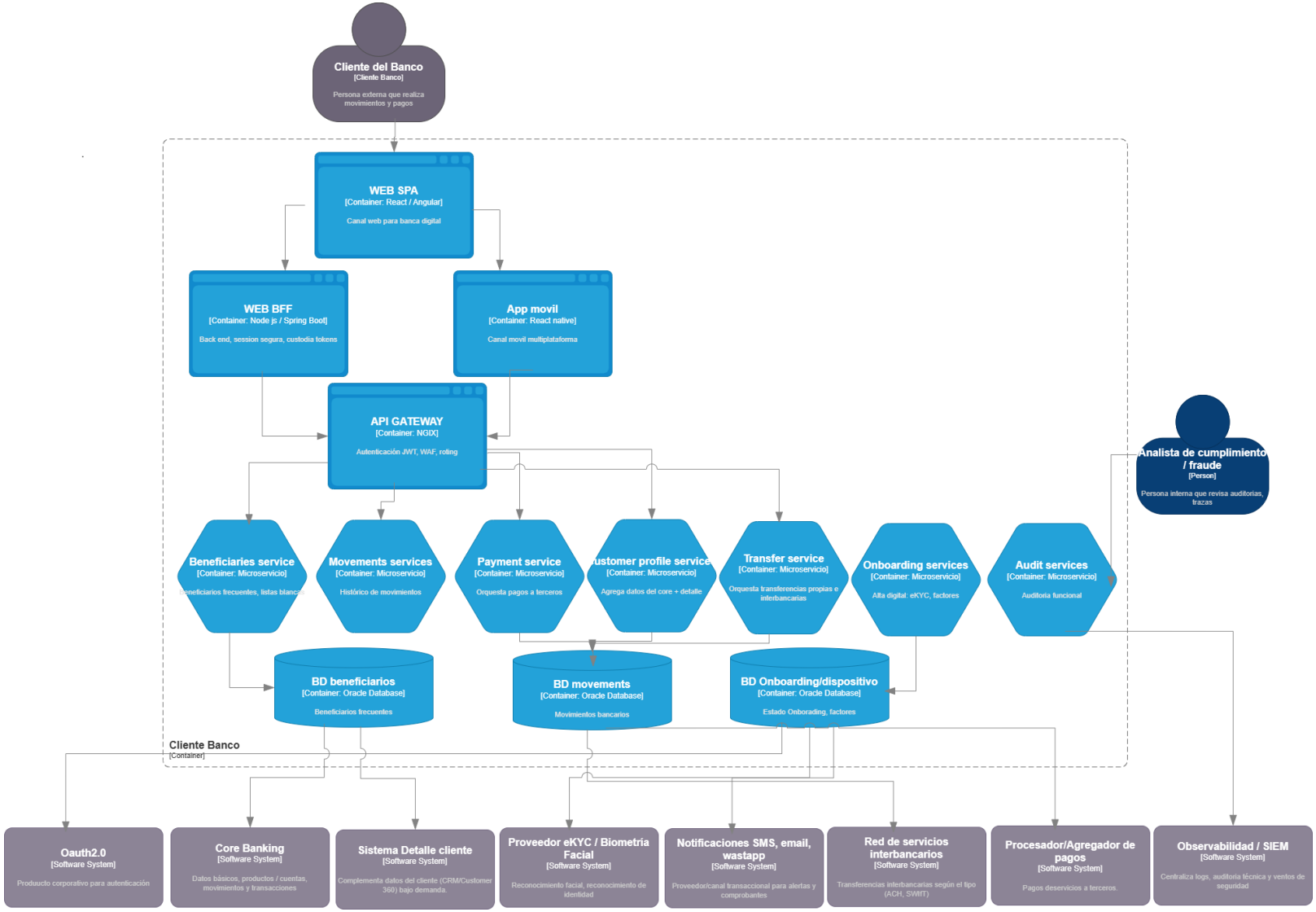
E) Capa de integración: API Gateway + servicios

El API Gateway es el punto único de entrada a las APIs del dominio (microservicios) y actúa como Policy Enforcement Point (PEP):

- Seguridad: autenticación, autorización, WAF/rate limiting, validación de tokens.
- Gobernanza de APIs: versionado, contratos, trazabilidad, control de cambios.
- Observabilidad: métricas, logs, trazas, correlación end-to-end.
- Resiliencia: timeouts, retries seguros, circuit-breaking a nivel edge (limitado).
- Recomendación: mantener el Gateway "delgado": políticas y enrutamiento; la lógica de negocio se queda en servicios.

Para esta solución serían:

- Customer Profile / Customer Aggregator (compone Core + Detalle y cachea)
- Accounts & Products
- Movements
- Transfers (con sub-integraciones interbancarias)
- Payments
- Beneficiaries/Frequent Contacts
- Notifications
- Audit
- Onboarding & Device Security


Cliente del Banco
[Cliente Banco]
Persona externa que realiza movimientos y pagos
WEB SPA
[Container: React / Angular]
Canal web para banca digital
WEB BFF
[Container: Node js / Spring Boot]
Back end, session segura, custodia tokens
App movil
[Container: React native]
Canal movil multiplataforma
API GATEWAY
[Container: NGIX]
Autenticación JWT, WAF, roting
Analista de cumplimiento / fraude
[Person]
Persona interna que revisa auditorias, trazas
Beneficiaries service
[Container: Microservicio]
Movements services
[Container: Microservicio]
Histórico de movimientos
Payment service
[Container: Microservicio]
Orquesta pagos a terceros
ustomer profile service
[Container: Microservicio]
Agrega datos del core + detalle
Transfer service
[Container: Microservicio]
Orquesta transferencias propias e interbancarias
Onboarding services
[Container: Microservicio]
Alta digital: eKYC, factores
Audit services
[Container: Microservicio]
Auditoria funcional
BD beneficiarios
[Container: Oracle Database]
Beneficiarios frecuentes
BD movements
[Container: Oracle Database]
Movimientos bancarios
BD Onboarding/dispositivo
[Container: Oracle Database]
Estado Onborading, factores
Cliente Banco
[Container]
Oauth2.0
[Software System]
Produucto corporativo para autenticación
Core Banking
[Software System]
Datos básicos, productos / cuentas, movimientos y transacciones
Sistema Detalle cliente
[Software System]
Complementa datos del cliente (CRM/Customer 360) bajo demanda.
Proveedor eKYC / Biometría Facial
[Software System]
Reconocimiento facial, reconocimiento de identidad
Notificaciones SMS, email, wastapp
[Software System]
Proveedor/canal transaccional para alertas y comprobantes
Red de servicios interbancarios
[Software System]
Transferencias interbancarias según el tipo (ACH, SWIT)
Procesador/Agregador de pagos
[Software System]
Pagos deservicios a terceros.
Observabilidad / SIEM
[Software System]
Centraliza logs, auditoria técnica y ventos de seguridad

2.3 Diagrama de Componentes:

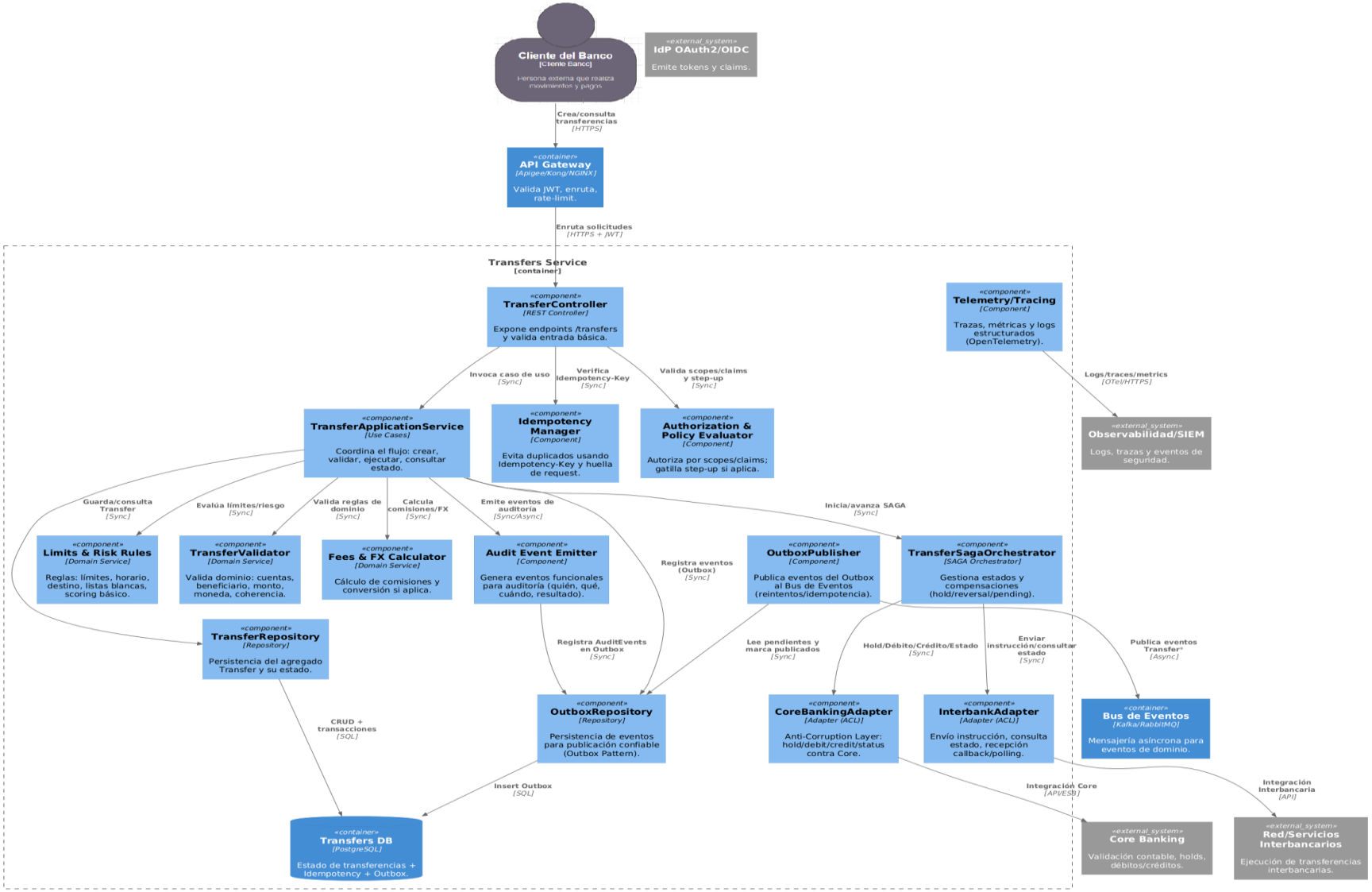
Patrones arquitectónicos

Para este ejercicio (banca digital con Core + sistema de detalle, OAuth2/OIDC, onboarding biométrico, notificaciones obligatorias, auditoría, caché para clientes frecuentes, API Gateway y varios servicios).

Protocolos de comunicación con seguridad

Para este ejercicio (banca digital con SPA + móvil, microservicios, bus de eventos, Core/externos y auditoría), la selección de protocolos de comunicación seguros debe cubrir: canal cliente, tráfico este-oeste interno, mensajería asíncrona, integraciones con terceros, y operación/observabilidad.

Usar cloud en esta solución se justifica porque permite escalar los canales digitales, asegurar alta disponibilidad, mejorar seguridad y cumplimiento, reducir carga operativa con servicios administrados (API Gateway, mensajería, Redis, DB, observabilidad), acelerar el time-to-market con DevSecOps, y soportar mejor flujos críticos como notificaciones normativas, auditoría inmutable y onboarding biométrico, todo manteniendo un enfoque híbrido compatible con un Core bancario legado.


Cliente del Banco
[Cliente Banco]
Persona externa que realiza movimientos y pagos
«external_system»
IdP OAuth2/OIDC
Emite tokens y claims.
Crea/consulta transferencias
[HTTPS]
«container»
API Gateway
[Apigee/Kong/NGINX]
Valida JWT, enruta, rate-limit.
Enruta solicitudes
[HTTPS + JWT]
Transfers Service
[container]
«component»
TransferController
[REST Controller]
Expone endpoints /transfers y valida entrada básica.
«component»
Telemetry/Tracing
[Component]
Trazas, métricas y logs estructurados (OpenTelemetry).
Logs/traces/metrics
[OTel/HTTPS]
«external_system»
Observabilidad/SIEM
Logs, trazas y eventos de seguridad.
Invoca caso de uso
[Sync]
Verifica Idempotency-Key
[Sync]
Valida scopes/claims y step-up
[Sync]
«component»
TransferApplicationService
[Use Cases]
Coordina el flujo: crear, validar, ejecutar, consultar estado.
«component»
Idempotency Manager
[Component]
Evita duplicados usando Idempotency-Key y huella de request.
«component»
Authorization & Policy Evaluator
[Component]
Autoriza por scopes/claims; gatilla step-up si aplica.
Guarda/consulta Transfer
[Sync]
Evalúa límites/riesgo
[Sync]
Valida reglas de dominio
[Sync]
Calcula comisiones/FX
[Sync]
Emite eventos de auditoría
[Sync/Async]
Inicia/avanza SAGA
[Sync]
«component»
Limits & Risk Rules
[Domain Service]
Reglas: límites, horario, destino, listas blancas, scoring básico.
«component»
TransferValidator
[Domain Service]
Valida dominio: cuentas, beneficiario, monto, moneda, coherencia.
«component»
Fees & FX Calculator
[Domain Service]
Cálculo de comisiones y conversión si aplica.
«component»
Audit Event Emitter
[Component]
Genera eventos funcionales para auditoría (quién, qué, cuándo, resultado).
Registra eventos (Outbox)
[Sync]
«component»
OutboxPublisher
[Component]
Publica eventos del Outbox al Bus de Eventos (reintentos/idempotencia).
«component»
TransferSagaOrchestrator
[SAGA Orchestrator]
Gestiona estados y compensaciones (hold/reversal/pending).
«component»
TransferRepository
[Repository]
Persistencia del agregado Transfer y su estado.
Registra AuditEvents en Outbox
[Sync]
Lee pendientes y marca publicados
[Sync]
Hold/Débito/Crédito/Estado
[Sync]
Enviar instrucción/consultar estado
[Sync]
Publica eventos Transfer*
[Async]
«component»
OutboxRepository
[Repository]
Persistencia de eventos para publicación confiable (Outbox Pattern).
«component»
CoreBankingAdapter
[Adapter (ACL)]
Anti-Corruption Layer: hold/debit/credit/status contra Core.
«component»
InterbankAdapter
[Adapter (ACL)]
Envío instrucción, consulta estado, recepción callback/polling.
«container»
Bus de Eventos
[Kafka/RabbitMQ]
Mensajería asíncrona para eventos de dominio.
CRUD + transacciones
[SQL]
Insert Outbox
[SQL]
Integración Core
[API/ESB]
Integración Interbancaria
[API]
«container»
Transfers DB
[PostgreSQL]
Estado de transferencias + Idempotency + Outbox.
«external_system»
Core Banking
Validación contable, holds, débitos/créditos.
«external_system»
Red/Servicios Interbancarios
Ejecución de transferencias interbancarias.

3. NORMATIVAS A APLICAR

1) Protección de datos personales y privacidad

Ley Orgánica de Protección de Datos Personales (LOPDP) y su reglamento.

Implicaciones arquitectónicas: clasificación de datos (PII/KYC/financieros/biométricos), minimización, consentimiento cuando aplique, derechos ARCO/ARCO+ (según marco), retención/expurgo, cifrado en tránsito y reposo, control de accesos (least privilege), registro de accesos y trazabilidad.

2) Prevención de Lavado de Activos y Financiamiento de Delitos (AML/CFT)

Ley orgánica de prevención/detección/combate del LA/FT (vigencia, reformas y obligaciones de reporte/controles, según sujeto obligado).

Implicaciones arquitectónicas: onboarding con KYC robusto, evidencia y trazabilidad, monitoreo transaccional (eventos), segregación de funciones, auditoría funcional inmutable, retención de información y reporting (cuando corresponda).

3) Marco financiero: regulación y supervisión del sistema financiero

Código Orgánico Monetario y Financiero (COMF) como marco base del sistema monetario/financiero y su supervisión.

- Normativa de la Superintendencia de Bancos (gestión de riesgos y gobierno; incluye lineamientos de seguridad/ciberseguridad y riesgo operativo).

- Implicaciones arquitectónicas: gobierno de riesgos, continuidad, gestión de incidentes, controles de seguridad y evidencia auditada de cumplimiento (políticas, monitoreo, segregación de ambientes, etc.).

4) Pagos y transferencias interbancarias (BCE / SPI)

SPI (Sistema de Pagos Interbancarios) y su reglamentación publicada por el Banco Central del Ecuador.

Implicaciones arquitectónicas: idempotencia (evitar dobles débitos), conciliación, trazabilidad end-to-end, manejo de estados (SAGA), ventanas/SLAs, evidencias para disputas.

5) Validez legal de transacciones electrónicas y firma

Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos y su reglamento. biblioteca.defensoria.gob.ec

Implicaciones arquitectónicas: no repudio (evidencias), integridad de comprobantes (hash), sellado temporal si aplica, custodia de mensajes/consentimientos y logs.

6) Estándares de seguridad exigibles por industria/auditoría

PCI DSS (si almacenas/procesas/transmites datos de tarjeta).
ISO/IEC 27001 (SGSI/ISMS) como referencia de controles y gobierno.

ISO 22301 (continuidad del negocio) para resiliencia operativa.
NIST CSF 2.0 como marco de gestión de riesgo ciber (muy defendible en un diseño).
OWASP ASVS (seguridad verificable para APIs/web) y OWASP MASVS (móvil).
OWASP Foundation

7) Autenticación fuerte para banca digital (muy recomendado)

WebAuthn / Passkeys (FIDO2) como estándar para autenticación fuerte (útil para "huella u otro método" y reducir phishing).

Implicaciones arquitectónicas: step-up authentication para transacciones sensibles, device binding, gestión de claves públicas por usuario/dispositivo, políticas de riesgo.

4. ALTA DISPONIBILIDAD

La alta disponibilidad (HA) para este ejercicio debe asegurar continuidad de servicio en los canales (Web SPA y App móvil), mantener operación transaccional (transferencias/pagos) sin duplicidades, y conservar trazabilidad (auditoría/notificaciones) aun cuando fallen componentes o una zona completa.

5. JUSTIFICACIÓN EN CLOUD

1) Escalabilidad por picos y lectura intensiva

El sistema tiene cargas típicamente variables:

- consultas de movimientos (picos por quincena, cierres, campañas),
- notificaciones masivas,
- onboarding.

Cloud permite autoescalado de:

- API Gateway / BFF / microservicios,
- consumidores del bus de eventos,
- capas de caché y lectura.

Esto evita sobredimensionar on-prem para "el peor día del año".

2) Alta disponibilidad y resiliencia más accesibles

En cloud es más directo implementar:
Multi-AZ (tolerancia a caída de zona),
DR multi-región (activo–pasivo) con RTO/RPO definidos,
despliegues blue/green/canary y rollback rápido.

En banca digital, la continuidad del canal es un requisito operativo central.

3) Patrones del ejercicio encajan naturalmente con servicios administrados

El diseño requiere componentes que en cloud suelen ser más confiables y rápidos de operar como “managed services”:

- API Gateway + WAF,
- mensajería/eventos (Kafka/PubSub),
- Redis,
- bases de datos con failover/backup/PITR,
- observabilidad (logs, métricas, trazas).

Esto reduce carga operativa (parches, HA manual, upgrades) y acelera entrega.

4) Seguridad “a escala” con controles estándar

Cloud facilita controles que en on-prem cuestan más operar consistentemente:

- KMS/HSM para llaves,
- secret manager con rotación,
- WAF/DDoS,
- segmentación de red con políticas,
- mTLS/service mesh (si lo implementas),
- monitoreo y postura de seguridad.

Con buen diseño, mejora el “security posture” de forma medible.

5) Time-to-market y DevSecOps

Para una entidad financiera, la ventaja competitiva es iterar rápido sin sacrificar controles:

- CI/CD,
- IaC,
- entornos reproducibles,
- pruebas y controles automatizados.

En resumen, sí se justifica una solución en cloud (preferentemente híbrida), porque el cloud habilita de forma más eficiente los requisitos clave del ejercicio: alta disponibilidad, escalabilidad, desacoplamiento por eventos, auditoría y notificación normativa, y onboarding biométrico, manteniendo el Core como sistema de registro y reduciendo el riesgo de migración.

PRECIOS REFERENCIALES

A continuación se detalla los costos referenciales de una nube con google (GCP)

Servicios Profesionales

Costo por infraestructura Opción 1

Descuento por Consumo a 1 año

Google Cloud	Cantidad	Precio unitario	Recurrencia	Total Anual USD
Consumo GCP Estimado modalidad CUD (Descuento por compromiso de uso) a un año	1	$14,822.43	Mensual	$177,869.17
Total Consumo GCP				**$177,869.17**

Costo por implementación

Servicios profesionales Xertica	Cantidad	Precio unitario	Recurrencia	Total USD
Servicios profesionales: Workload Migration (Migración de 88 máquinas virtuales), implementación de Landing Zone y ambiente de red	1	$ 50,600.00	Único	$ 50,600.00
Total Servicios Implementación				**$ 50,600.00**

Costo por transferencia de conocimiento

Servicios profesionales Xertica	Cantidad	Precio unitario	Recurrencia	Total USD
Servicio de transferencia de conocimiento para 5 personas, Xertica entregará certificado de participación	1	$ 4,500.00	Único	$ 4,500.00
Total Servicios Implementación				**$ 4,500.00**

Costo por servicios de soporte

Servicios profesionales Xertica	Cantidad	Precio unitario	Recurrencia	Total USD
Servicios administrados Optimize 7*24 incluye soporte	1	$ 2,000.00	Mensual	$ 24,000.00
Total Servicios de Soporte				**$ 24,000.00**
Total General Propuesta 12 meses				**$ 256.969,17**

La solución de monitoreo para BancoP debe diseñarse como una plataforma de observabilidad (no solo “monitoring”) que cubra, de punta a punta, disponibilidad, rendimiento, seguridad, cumplimiento (auditoría) y experiencia de usuario en Web y Móvil. En banca, además, debe soportar investigación forense y evidencias.

Objetivos de monitoreo

- Disponibilidad del canal: SPA, App móvil, API Gateway/BFF.
- Salud transaccional: transferencias/pagos sin duplicidad, estados SAGA consistentes.
- Cumplimiento y trazabilidad: evidencias auditables por acción y por transacción.
- Rendimiento: latencia p95/p99 por operación crítica (movimientos, transferencias, login).

- Resiliencia: degradación controlada ante caída de Core/interbancario/proveedores.
- Detección de fraude/anomalías: patrones de abuso, brute force, device changes.
- Operación eficiente: alertas accionables, runbooks, tiempos de respuesta bajos.

6. MONITOREO

La solución de monitoreo para BancoP debe diseñarse como una plataforma de observabilidad (no solo “monitoring”) que cubra, de punta a punta, disponibilidad, rendimiento, seguridad, cumplimiento (auditoría) y experiencia de usuario en Web y Móvil. En banca, además, debe soportar investigación forense y evidencias.

1) Objetivos de monitoreo

- Disponibilidad del canal: SPA, App móvil, API Gateway/BFF.
- Salud transaccional: transferencias/pagos sin duplicidad, estados SAGA consistentes.
- Cumplimiento y trazabilidad: evidencias auditables por acción y por transacción.
- Rendimiento: latencia p95/p99 por operación crítica (movimientos, transferencias, login).
- Resiliencia: degradación controlada ante caída de Core/interbancario/proveedores.
- Detección de fraude/anomalías: patrones de abuso, brute force, device changes.
- Operación eficiente: alertas accionables, runbooks, tiempos de respuesta bajos.