Sistema de Banca por Internet (BP)

1. **Resumen ejecutivo y alcance**

El sistema de **Banca por Internet (BP) en Ecuador** se concibe como una plataforma digital integral que permita a los clientes acceder de manera **segura, rápida y sencilla** a los servicios financieros más relevantes: consulta de saldos y movimientos, transferencias entre cuentas propias e interbancarias, pagos de servicios, notificaciones transaccionales y un proceso de **onboarding digital con biometría facial**.

Objetivo del sistema

- **Facilitar la inclusión financiera** al ofrecer un canal digital accesible 24/7 desde web y móvil.
- **Mejorar la experiencia del cliente** mediante tiempos de respuesta bajos, alta disponibilidad y procesos intuitivos.
- **Reducir costos operativos** asociados a agencias físicas, migrando transacciones al canal digital.
- **Cumplir con la normativa ecuatoriana** (Superintendencia de Bancos, LOPDP, BCE) y estándares internacionales de seguridad.
- **Garantizar transparencia y trazabilidad** de las operaciones mediante una base de auditoría.

Stakeholders clave

- **Negocio**: responsables de la estrategia digital y la propuesta de valor.
- **Riesgos/Fraude**: aseguramiento contra suplantación de identidad, fraude transaccional y cumplimiento normativo.
- **Canales**: equipos de experiencia digital para web y móvil.
- **Seguridad**: encargados de ciberseguridad, monitoreo y respuesta a incidentes.
- **Datos**: administradores de bases de datos, gobierno de datos y analítica.
- **Operaciones**: soporte técnico, continuidad del negocio y monitoreo de disponibilidad.

Alcance vs. fuera de alcance

- **Alcance inicial (MVP):**
  - Aplicación **SPA Web** y aplicación **móvil multiplataforma** (framework multiplataforma).
  - Autenticación y autorización con OAuth2.0/OIDC.
  - Consultas de saldos y movimientos.
  - Transferencias propias e interbancarias.
  - Pagos básicos (servicios y préstamos).
  - Notificaciones en al menos 2 canales (email + SMS/Push).
  - Onboarding digital con biometría facial y validación documental, integrado al flujo de autenticación.
  - **Base de auditoría** para registro de operaciones y persistencia de información frecuente.

- **API Gateway** como capa de integración para exponer servicios internos y consumir sistemas externos.

- **Futuras fases (fuera del MVP, pero planificadas):**
  - Pagos avanzados (tarjetas de crédito, impuestos, seguros).
  - Integración con open finance y APIs externas.
  - Servicios de inversión y gestión patrimonial.
  - Analítica avanzada en tiempo real para detección de fraude.

Justificación de decisiones iniciales

1. Uso de OAuth2.0/OIDC para autenticación:
   a. Alternativa evaluada: autenticación propietaria (más compleja de mantener, menos segura).
   b. Decisión: estándar abierto para interoperabilidad y cumplimiento con NIST 800-63.
2. Arquitectura en AWS:
   a. Alternativa: infraestructura on-premise (menos escalable, mayor costo inicial, más lenta en time-to-market).
   b. Decisión: nube pública (AWS) por resiliencia, escalabilidad, servicios financieros certificados y cumplimiento (PCI DSS, ISO 27001).
3. SPA + App móvil multiplataforma:
   a. Alternativa: desarrollo móvil nativo (más costoso, mayor tiempo de mantenimiento).
   b. Decisión: SPA + framework multiplataforma (Flutter/React Native) para reducir costos y acelerar el time-to-market.

2. Requisitos Funcionales y no funcionales

El sistema de Banca por Internet cubrirá los siguientes **casos de uso principales**:

- **Acceso seguro**: autenticación con estándares internacionales (OAuth2/OIDC), con múltiples factores de seguridad (ej. OTP, biometría facial).
- **Consultas rápidas**: saldos y movimientos disponibles en segundos gracias a optimización de consultas y almacenamiento temporal en caché.
- **Transferencias seguras**: entre cuentas propias e interbancarias, con confirmación inmediata y controles que evitan errores o duplicados.
- **Pagos digitales**: servicios, préstamos y convenios, con conciliación automática en el Core Bancario.
- **Notificaciones multicanal**: alertas por email y mensajería móvil (ej. SMS o WhatsApp), con posibilidad de configurar preferencias.
- **Onboarding digital**: registro de nuevos clientes con biometría facial y validación documental (KYC), reduciendo fraude de identidad.

- **Gestión de dispositivos**: administración de accesos, cierre remoto de sesiones y alertas cuando se detectan accesos sospechosos.

Integraciones clave

- **Core Bancario**: fuente principal de información sobre productos, saldos y movimientos.
- **Sistemas complementarios**: validaciones de identidad, listas de fraude y scoring crediticio.

Beneficios esperados

- **Clientes**: acceso rápido, seguro y disponible 24/7.
- **Banco**: reducción de fraude, menores costos operativos y mayor adopción digital.
- **Reguladores**: cumplimiento normativo (Superintendencia de Bancos, BCE, LOPDP).

Requisitos de calidad (no funcionales)

- **Disponibilidad**: plataforma siempre disponible, con una meta de 99.9% en producción.
- **Rendimiento**: operaciones críticas con tiempos de respuesta menores a 1 segundo en la mayoría de los casos.
- **Escalabilidad**: capacidad de crecer dinámicamente en campañas o picos de uso.
- **Seguridad**: cifrado de datos, protección de accesos y cumplimiento con estándares internacionales.
- **Continuidad del negocio**: recuperación rápida ante fallos o desastres, con respaldo de datos permanente.
- **Monitoreo y soporte**: alertas proactivas, despliegues seguros y capacidad de recuperación ágil en caso de incidentes.

3. Supuestos, restricciones y riesgos

El éxito de la arquitectura depende de la validez de ciertos **supuestos de base**, del cumplimiento de **restricciones legales y técnicas** y de la **gestión activa de riesgos**.

a. Supuestos técnicos y de negocio

i. **Conectividad segura y estable** entre los sistemas del banco (Core, sistemas complementarios) y AWS, mediante VPN o Direct Connect.
ii. **Adopción del canal digital** por al menos el 40% de los clientes activos en los primeros 12 meses.
iii. **Capacidad de integración** del Core Bancario con APIs expuestas (REST/ SOAP) y posibilidad de evolucionar hacia un ESB o event bus.
iv. **Disponibilidad de personal especializado** en AWS, DevSecOps y ciberseguridad bancaria.
v. **Aprobación regulatoria**: la SB y el BCE aceptan el uso de nube pública siempre que se cumplan los controles de seguridad y localización de datos.

b. Restricciones
i. Legales:
- Cumplimiento obligatorio de la **Ley Orgánica de Protección de Datos Personales (LOPDP)** en Ecuador.
- Normativa de la **Superintendencia de Bancos (SB)** sobre seguridad de la información y continuidad.

ii. Normativa del **Banco Central del Ecuador** sobre pagos y transferencias interbancarias.
iii. Tecnológicas:
- Algunas dependencias del Core Bancario son sistemas legados con tiempos de respuesta variables (>2s).
- Integraciones con BCE sujetas a disponibilidad y protocolos actuales (ej. SOAP/XML).

iv. Presupuesto:
- Para el MVP/POV se limitará el gasto mensual en AWS usando arquitecturas bajo demanda.
- En producción, el presupuesto deberá escalar, pero con métricas FinOps y optimización continua.

v. Plazos: (Tiempo propuesto dispuesto a variación)
- POV debe estar disponible en **6 meses**.
- Producción full-feature en un plazo de **18–24 meses**.

c. Riesgos y mitigaciones

Riesgo	**Severidad**	**Probabilidad**	**Mitigación**
Fraude de identidad (onboarding, accesos)	Alta	Media	MFA obligatorio, biometría *liveness*, monitoreo antifraude en tiempo real.

Riesgo	Severidad	Probabilidad	Mitigación
Caída de Core Bancario	Alta	Media	Circuit breaker, caché de última consulta, colas con reintentos, DRP de Core.
Latencia alta en BCE para transferencias interbancarias	Media	Alta	Manejo asíncrono de confirmaciones, notificación posterior, SLA comunicados al cliente.
Fuga de datos sensibles (PII, biometría)	Alta	Baja	Cifrado E2E, KMS, controles de IAM, auditoría de accesos, retención mínima de biometría.
Sobrecostos en AWS	Media	Media	Uso de Savings Plans, límites presupuestales, monitoreo con AWS Budgets y alarmas.
Resistencia cultural interna (usuarios internos del banco)	Media	Media	Plan de capacitación, gestión del cambio, pilotos internos antes de liberar a clientes.
Regulador cambia requisitos (ej. localización de datos)	Alta	Baja	Diseñar multi-región con opción de almacenar en S3 en región compatible, revisión legal continua.
Ataques DDoS o bots	Alta	Alta	AWS WAF, Shield Advanced, rate limiting, scrubbing centers de ISP.

d. Conclusiones del análisis de riesgos

- **Mayor impacto**: seguridad y cumplimiento (fraude, fuga de datos, ataques DDoS).
- **Mayor probabilidad**: dependencias externas (Core y BCE).
- **Mitigaciones**: incluyen controles técnicos (circuit breakers, cifrado, WAF) y organizacionales (capacitación, gestión del cambio, comunicación con regulador).

4. Diagrama de alto nivel

4.0. Diagrama de Contexto (C1).

Propósito: Mostrar el ecosistema que rodea a **Banca por Internet (BP)**: quiénes usan el sistema, con qué se conecta y qué intercambia.

Actores principales

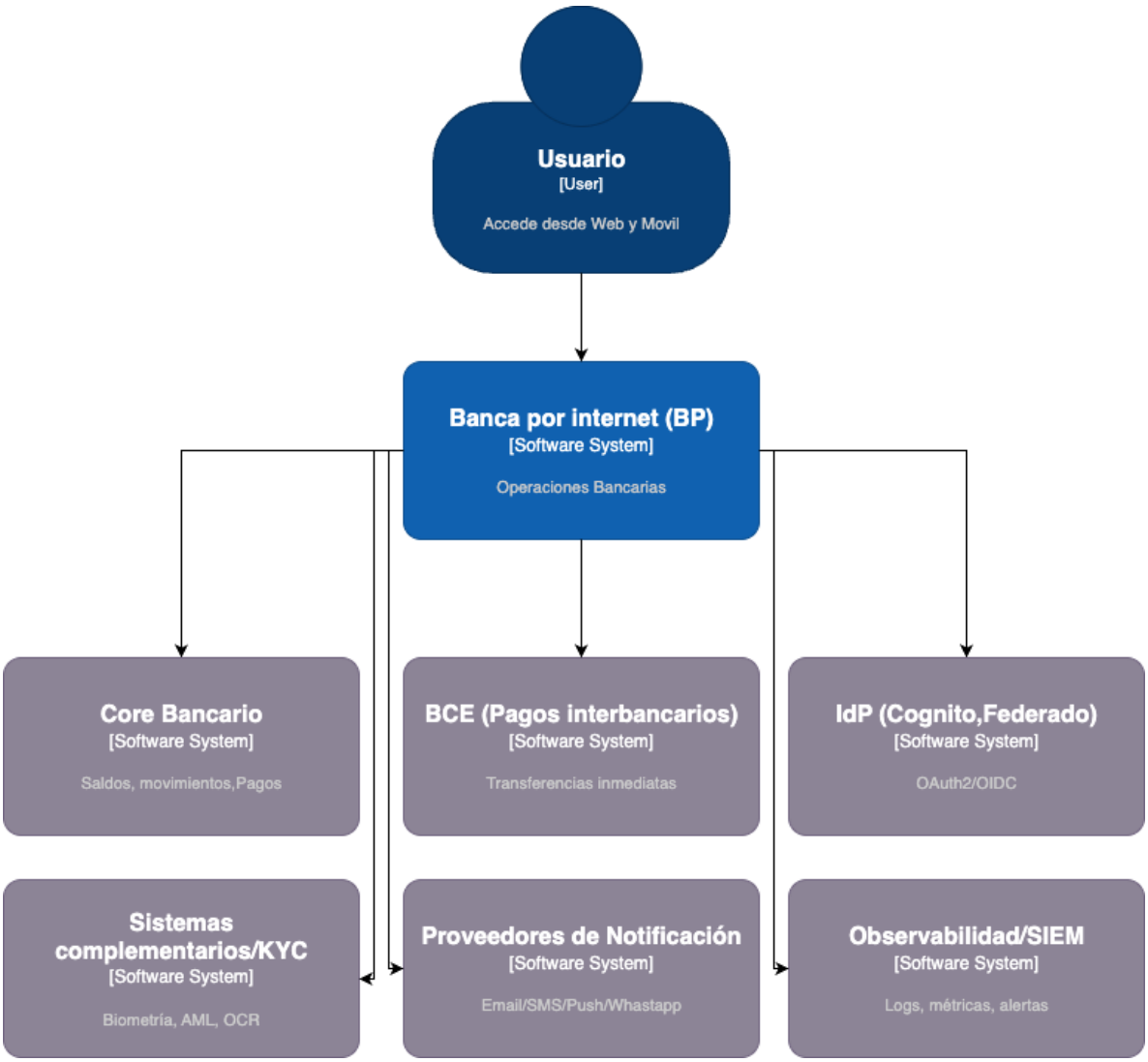
- Cliente Web (navegador moderno).

- Cliente Móvil (app Android/iOS).
- Backoffice (operadores del banco para soporte y monitoreo).
- Administrador de Seguridad (SOC/Equipo de ciberseguridad).

Sistemas externos

- **Core Bancario** (saldos, movimientos, pagos, transferencias).
- **Sistema Complementario/KYC** (biometría, liveness, OCR, AML/CFT, buró).
- **IdP** (Cognito o federado) para **OAuth2/OIDC**.
- **BCE** (pagos/transferencias interbancarias).
- **Proveedores de notificación** (Email/SMS/Push/WhatsApp).
- **Observabilidad/SIEM** (logs, métricas, alertas).

Diagrama:



Detalles:

Flujos de alto nivel (qué se intercambia)

1. Autenticación y autorización: Cliente ↔ IdP (OIDC/OAuth2 con PKCE). BP verifica tokens en cada solicitud.
2. Consultas de información: BP ↔ Core Bancario (saldos/movimientos). Caché de lectura para mejorar latencia.

3. Órdenes transaccionales: BP → Core (transferencias propias/pagos) y BP ↔ BCE (interbancarias). Confirmación síncrona o asíncrona según el caso.
4. Onboarding biométrico/KYC: Cliente ↔ BP ↔ Sistema Complementario (liveness, OCR, listas AML/CFT). Solo persistir evidencias mínimas requeridas por regulación.
5. Notificaciones: BP → Proveedores (email/SMS/push/WhatsApp) según preferencias del cliente, con registro de auditoría.
6. Observabilidad y seguridad: BP → SIEM/Logging/SOC (eventos, alertas, auditoría, cumplimiento).

Límites de confianza y datos personales

- PII y biometría se tratan con minimización y propósito específico (LOPDP). Se documentan bases legales (consentimiento explícito para biometría) y tiempos de retención.
- Tokens y secretos nunca viajan por canales inseguros; TLS 1.2+ extremo a extremo.
- Separación de dominios: autenticación (IdP) aislada de la lógica transaccional.

Amenazas y controles de nivel contexto

- Suplantación de identidad → MFA/step-up, liveness, device fingerprinting.
- Exposición de datos → cifrado en tránsito/represo, política de acceso mínimo, registro y auditoría.
- Indisponibilidad de dependencias (Core/BCE) → circuit breaker, colas y compensaciones.
- Abuso automatizado (bots) → WAF/Bot Control, rate limiting, detección de anomalías.

Métricas clave a nivel contexto

- Disponibilidad canal (web/móvil), tiempo de login, latencia consultas, éxito de transferencias, tasa de fraude y entregabilidad de notificaciones.

4.1. Diagrama de Contenedores (C2)

Propósito
Mostrar cómo se compone el sistema en **contenedores desplegables** y cómo se comunican entre sí y con sistemas externos (Core, BCE, IdP, Notificaciones).

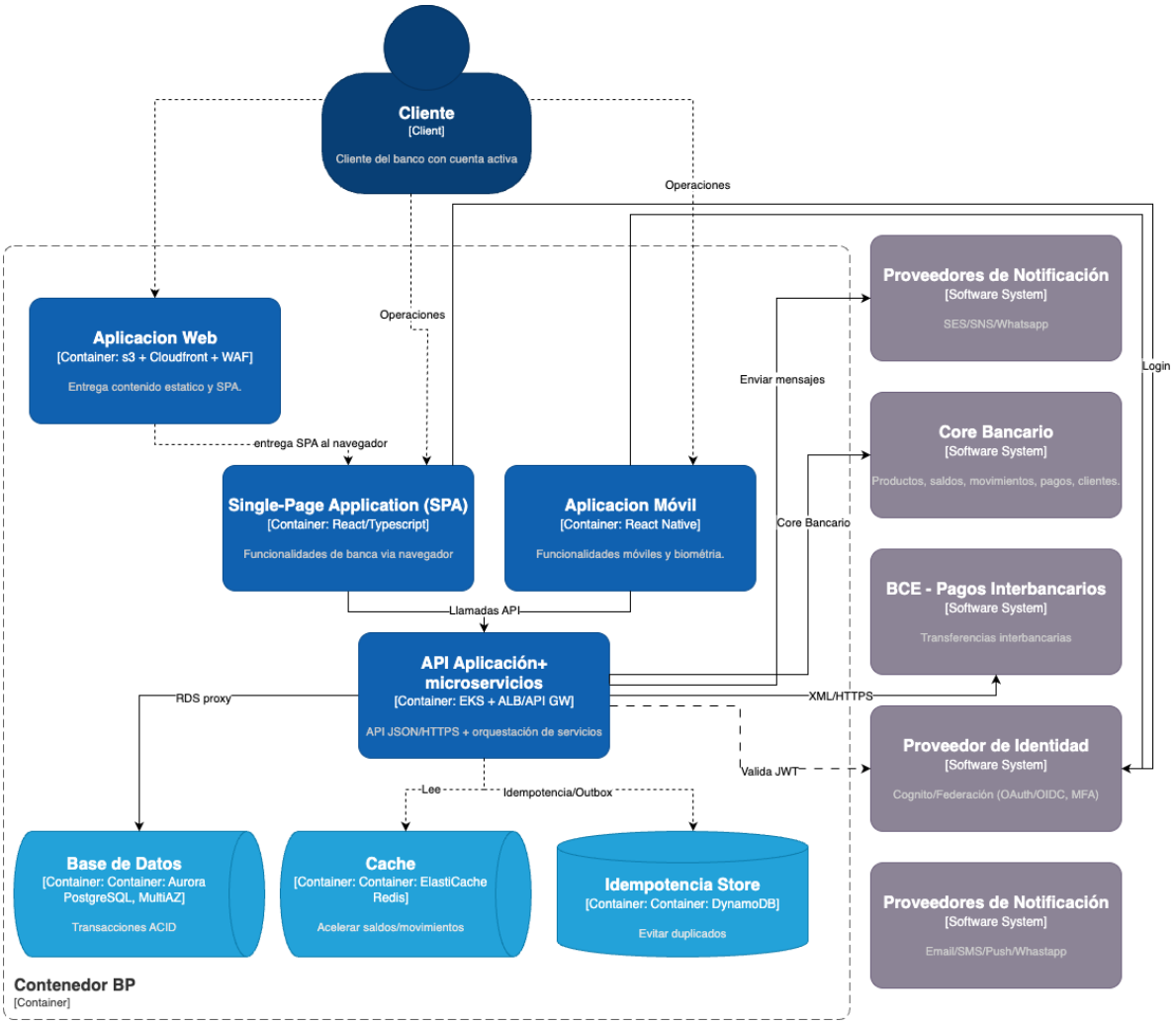
Contenedores (dentro del límite BP)

- Web Application (S3 + CloudFront + WAF/Shield): sirve contenido estático y la SPA con CDN y protección perimetral.
- Single-Page Application (React/TS): interfaz web que consume la API.
- Mobile App (React Native/Flutter): interfaz móvil y onboarding biométrico.
- API Application / BFF + Microservicios (EKS + ALB/API GW): expone JSON/HTTPS, valida JWT y orquesta dominios (Cuentas, Movimientos, Transferencias, Pagos, Notificaciones, Auditoría).
- Database (Aurora PostgreSQL Multi-AZ): transacciones ACID (órdenes, saldos, movimientos) con alta disponibilidad.
- Cache (ElastiCache Redis): acelera lecturas (saldos/movimientos) y aplica rate-limit.
- Idempotency Store (DynamoDB): previene duplicados y soporta patrón outbox.

Sistemas externos (fuera del límite)

- Identity Provider (Cognito/federado): OAuth2/OIDC + MFA (PKCE) y validación de tokens.
- Core Bancario: productos, saldos, movimientos, pagos.
- BCE (Interbancario): transferencias SPI.
- Proveedores de Notificación (SES/SNS/WhatsApp/Push): envíos transaccionales.

Gráfico:



Flujo:

Acceso y autenticación

- El **Cliente** abre **Web** (CloudFront → S3) o la **App Móvil**.
- La SPA/App redirige al **IdP (Cognito)** con **PKCE**; al volver, consume la **API** portando el **JWT**.

Consultas (saldos/movimientos)

- SPA/App llama a **API/BFF** → primero consulta **Redis** (si hay caché) → si no, lee de **Aurora** o integra con **Core**.
- La API devuelve respuesta rápida (sub-segundo con caché).

Transferencias y pagos

- SPA/App envía la orden a **API/BFF** con un **idempotency-key**.
- La API valida el **JWT**, reglas de límites y riesgo; persiste en **Aurora**; guarda la clave en **DynamoDB** para evitar duplicados.
- Para interbancarias, la API integra con **BCE**; para propias/pagos, con **Core**.
- Se publican **eventos outbox** (conciliación, auditoría, notificaciones).

Notificaciones

- La API invoca **SES/SNS/WhatsApp/Push** con las preferencias del cliente y registra el envío.

Operación continua

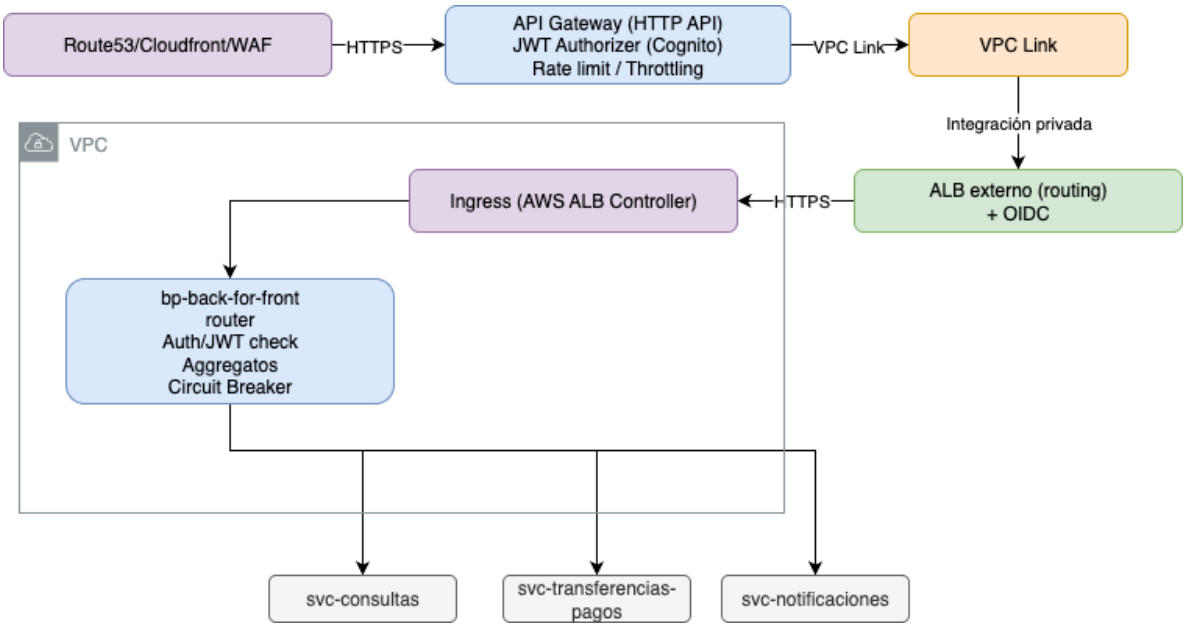
- **WAF/Shield** protegen el perímetro; **ALB/API GW** enrutan; **EKS** escalan microservicios; **Aurora Multi-AZ** mantiene HA.
- Todas las llamadas se trazan y monitorean (CloudWatch/X-Ray/Grafana, no mostrado para simplificar el C2).

4.2. Diagramas de Componentes (C3)

En este capítulo bajamos al nivel de componentes de los dominios críticos.

4.2.1. API/BFF (EKS, ingreso con **API Gateway → ALB → EKS**)

Propósito. Ser la “puerta” inteligente para Web/Móvil: validar JWT, aplicar rate-limit y resiliencia, y entregar a la UI respuestas ya agregadas, sin acoplarla a microservicios.



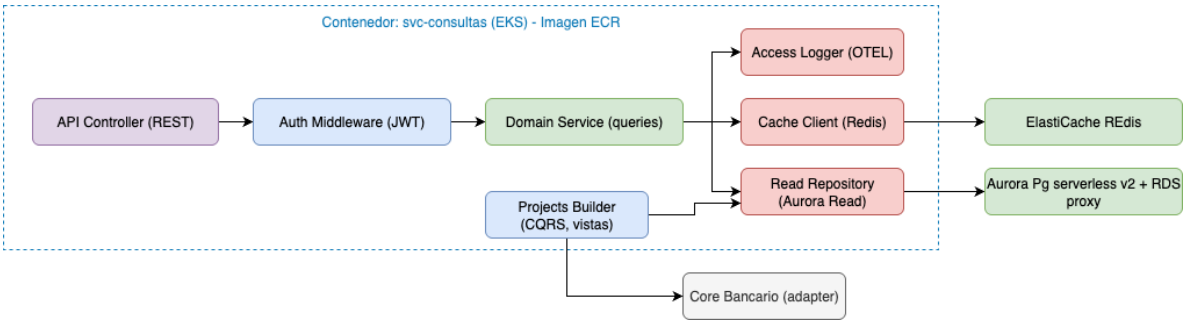
Flujo:

Route53/CloudFront/WAF → **API Gateway (HTTP API)** con **JWT Authorizer (Cognito/IdP)** y *throttling*. API Gateway integra por **VPC Link** a un **ALB externo** (enrutamiento; **OIDC opcional** para defensa en profundidad). El **Ingress** de EKS (AWS Load Balancer Controller) publica el **BFF**. El BFF usa *aggregators/facades* y *circuit breaker/retry* para hablar con **svc-consultas, svc-transferencias-pagos y svc-notificaciones.**

4.2.2. Consultas (Cuentas/Movimientos)

Propósito. Responder saldos/movimientos con baja latencia y costo.

Concepto. CQRS liviano con vistas de lectura en **Aurora PostgreSQL (Multi-AZ/Serverless v2) + RDS Proxy** y **ElastiCache Redis** como caché; adaptadores al **Core**; telemetría a **CloudWatch.**

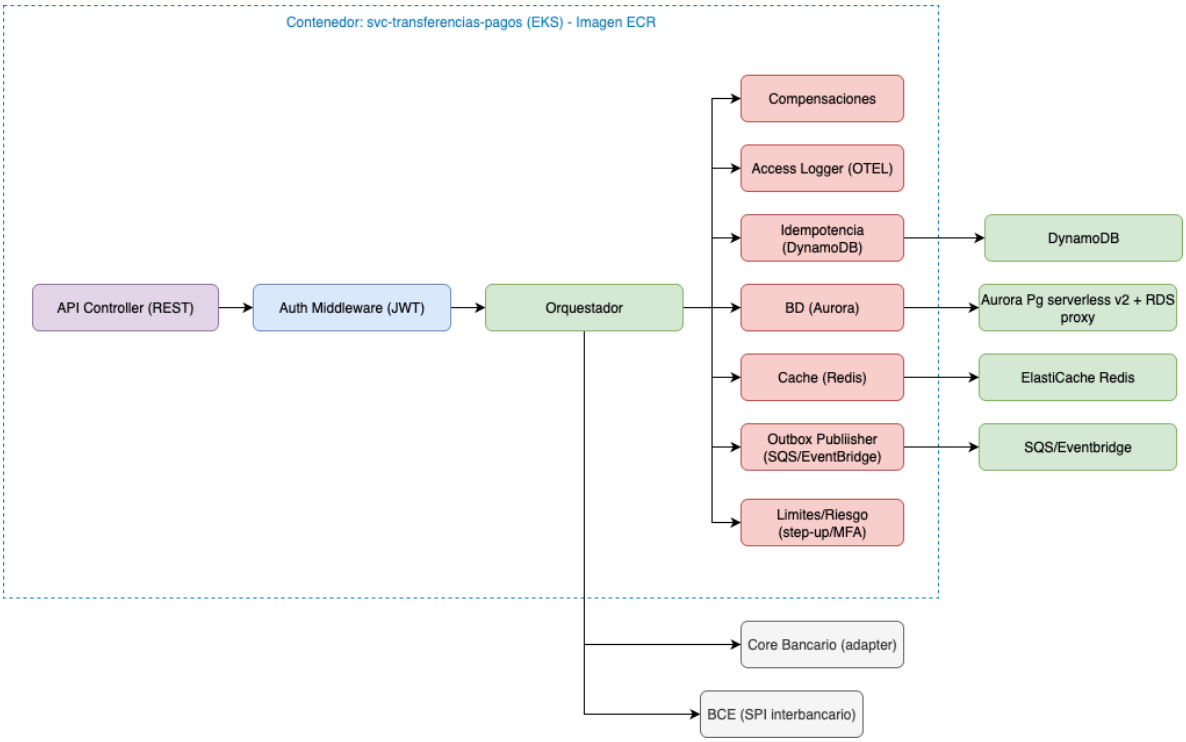


Flujo. BFF → API (JWT) → Domain Service → (hit en **Redis** o lectura en **Aurora**) → OTEL → respuesta.

4.2.3. Transferencias / Pagos

Propósito. Ejecutar órdenes críticas sin duplicados, con límites/riesgo, compensaciones y auditoría.

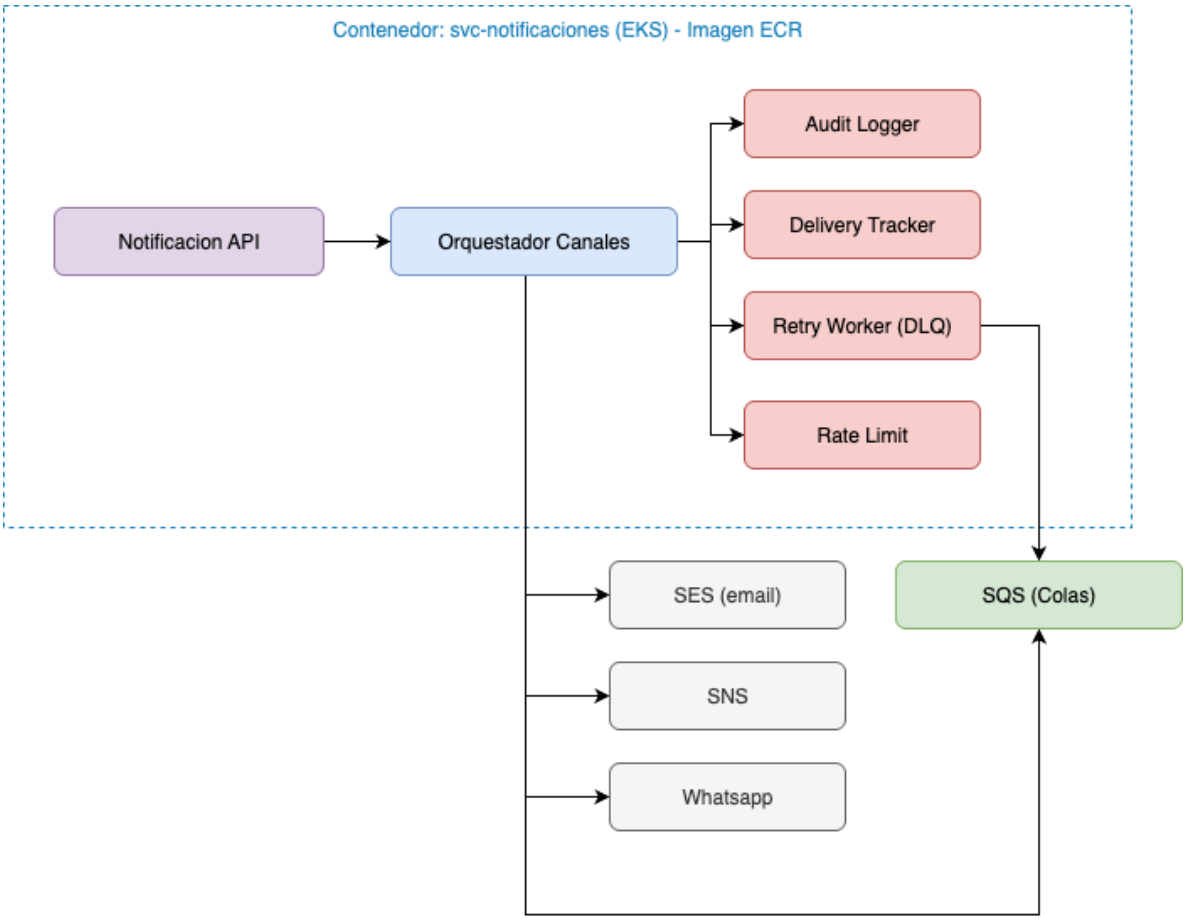
Concepto. Orquestador transaccional + **idempotencia en DynamoDB**, **Outbox** (**SQS/EventBridge**) y **Saga** para fallos parciales; adapters a **Core** (propias/pagos) y **BCE** (interbancarias); persistencia en **Aurora + RDS Proxy**, caché con **Redis**.



Flujo. BFF → API (JWT) → Orquestador → (límites/step-up) → **DynamoDB** (idempotencia) → **Aurora** → Core/BCE → **SQS/EventBridge** (outbox) → (si falla) **Saga** → auditoría OTEL.

4.2.4. Notificaciones (SES/SNS/SQS/WhatsApp/Push)

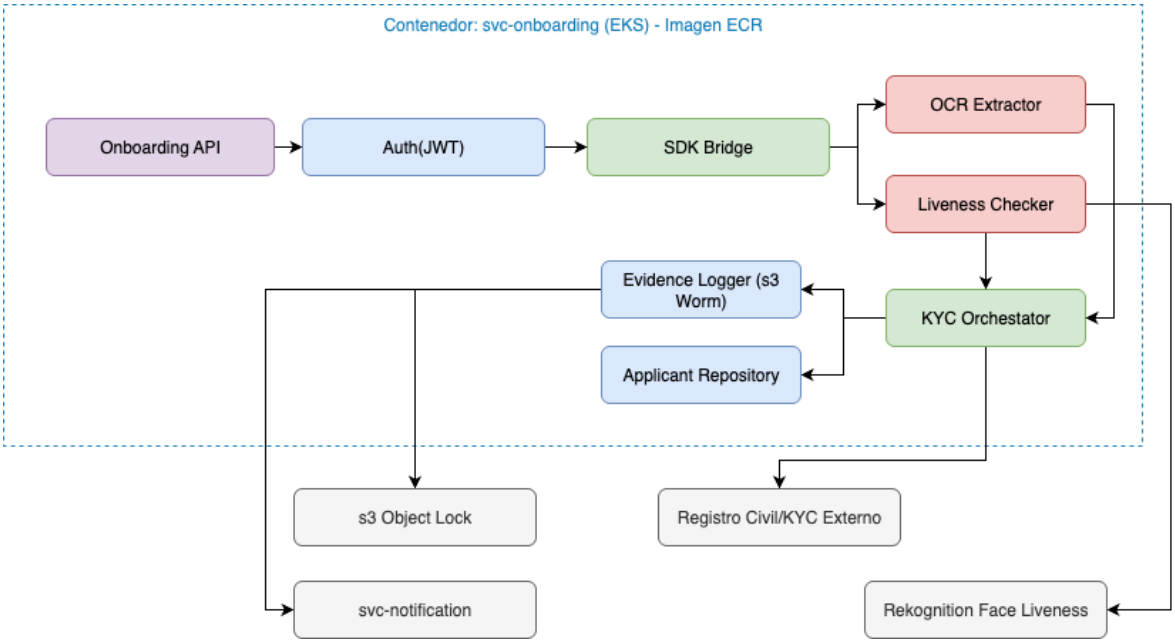
Propósito. Entrega multicanal confiable con preferencias, control de velocidad, reintentos y trazabilidad.
Concepto. API interna → Orquestador → **SQS** (buffer/DLQ) → canales **SES/SNS/WhatsApp**; estados y metadatos en **DynamoDB/Aurora**; auditoría.



Flujo. Servicio emisor → Notification API → (preferencias + template) → Orquestador → rate-limit → **SQS** → **SES/SNS/WhatsApp** → tracking → auditoría.

4.2.5. Onboarding Biométrico / KYC (Rekognition)

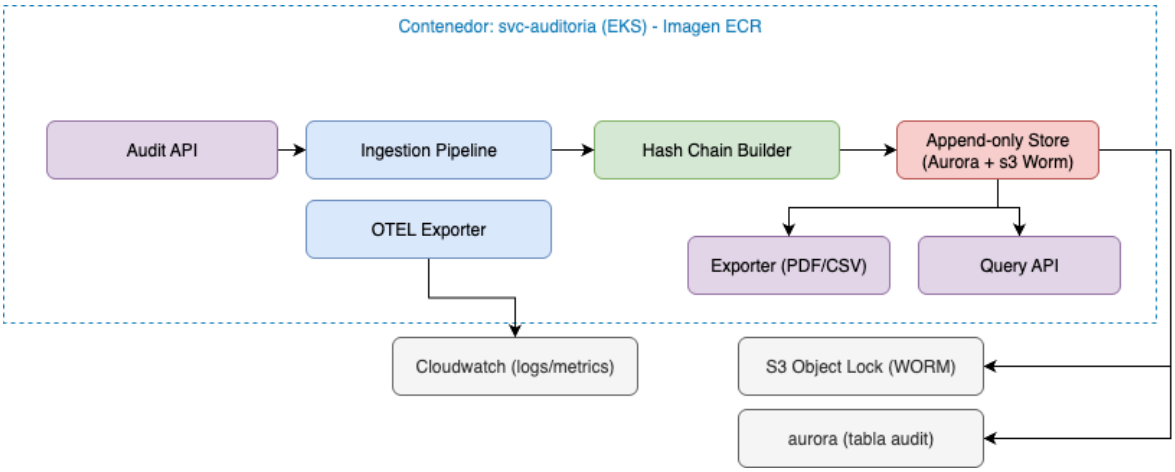
Propósito. Alta digital segura con prueba de vida, OCR y validación AML/CFT.
Concepto. Captura selfie/DNI → **Rekognition Face Liveness** + OCR → **KYC Orchestrator** (listas/Registro Civil) → expediente en **Aurora/DynamoDB**, evidencias en **S3 Object Lock (WORM)** → notificación.



Flujo. App → Onboarding API (JWT) → Capture SDK → Liveness (**Rekognition**) + OCR → KYC → guarda solicitud + evidencias (**S3 WORM**) → notifica resultado.

4.2.6. Auditoría / No Repudio & Observabilidad

Propósito. Evidencias inmutables para cumplimiento y telemetría centralizada.
Concepto. API *write-only* → ingest → **hash-chain** → **Aurora + S3 Object Lock (WORM)**; **Query/Export** para auditores; **OTEL** a **CloudWatch.**



Flujo. Servicios → Audit API → Ingest → Hash-chain → Append-only store (**Aurora/S3 WORM**) → consultas/export → métricas/logs/trazas (**CloudWatch**).

5. Arquitectura de despliegue en AWS

La plataforma de Banca por Internet se construirá sobre AWS siguiendo buenas prácticas de arquitectura multi-cuenta, seguridad desde el diseño e infraestructura como código (IaC). Esto garantiza escalabilidad, cumplimiento regulatorio en Ecuador y resiliencia ante fallas.

5.0. Multi-cuenta (Landing Zone)

- **Estructura de cuentas**:
  - **Prod:** cargas en producción con controles más estrictos.
  - **QA/Staging:** pruebas de calidad y pre-producción.
  - **Dev:** entornos de desarrollo aislados.
  - **Seguridad:** cuentas centralizadas para GuardDuty, Security Hub, auditoría.
  - **Datos/Analítica:** procesamiento de logs y analítica avanzada.
- **Control centralizado:** AWS Organizations con Service Control Policies (SCPs).
- **Justificación:** mejora de seguridad y gobernanza; aislamiento de riesgos entre ambientes.

5.1. Red y conectividad

- **VPCs dedicadas** por ambiente, con subredes públicas, privadas y de datos.
- **Subredes privadas** para microservicios y bases de datos.
- **Endpoints VPC** (S3, DynamoDB, Secrets Manager, KMS) para evitar tráfico a internet.
- **Conexión on-premise:** AWS Site-to-Site VPN en POV; migración a **Direct Connect** para producción con latencia estable y cumplimiento BCE.
- **Seguridad perimetral:** WAF + Shield Advanced en CloudFront/ALB.

- **Justificación:** cumplimiento de requerimientos de la SB sobre redes segregadas y resilientes.

5.2. Estrategia IaC (Infrastructure as Code)

- **Herramientas:** Terraform para infraestructura y AWS CDK para componentes específicos.
- Prácticas:
  - Repositorios separados por ambiente.
  - Validación de cambios con pipelines (CI/CD).
  - Escaneo de seguridad IaC (tfsec, cdk-nag).
- **Naming y etiquetado:** convención uniforme con campos, ayuda de Helpers DRY
- **Justificación:** reproducibilidad, auditoría y reducción de errores manuales.

5.3. Gestión de identidades y accesos

- **Clientes (CIAM):** Amazon Cognito User Pools con federación a Google/Microsoft/Apple (OIDC/SAML).
- **Colaboradores (Workforce):** AWS IAM Identity Center (ex SSO) federado con Microsoft Entra ID.
- **Principio de menor privilegio:** IAM Roles con políticas gestionadas; SCPs en Organization.
- **Gestión de secretos y claves:** AWS Secrets Manager para credenciales y certificados; AWS KMS/HSM para cifrado en reposo y llaves maestras.
- **Rotación automática:** contraseñas, llaves y certificados rotados periódicamente.
- **Justificación:** cumplimiento con LOPDP y regulaciones bancarias de Ecuador, evitando exposición indebida de credenciales.

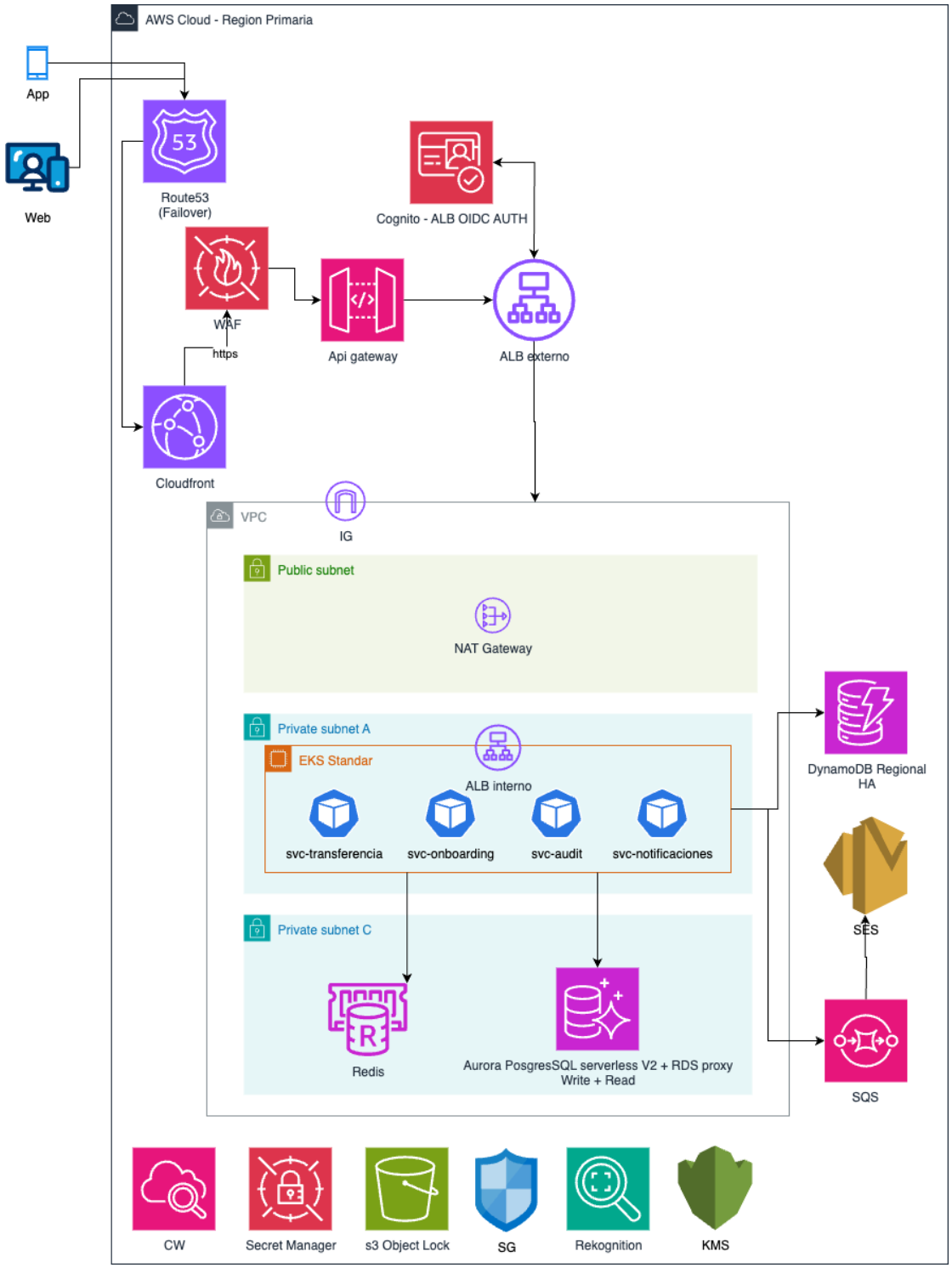
5.4. Observabilidad y gobierno

- **Logs centralizados:** CloudWatch Logs enviados a cuenta de seguridad.
- **Monitoreo:** CloudWatch metrics, X-Ray, OpenTelemetry → Grafana.
- **Alertas:** integradas con SNS/Slack/Teams.
- **Gobierno:** tagging obligatorio, CMDB sincronizado con AWS Config.

5.5. Beneficios de la estrategia AWS

- **Seguridad:** cuentas aisladas + SCPs + monitoreo central.
- **Escalabilidad:** redes y servicios listos para crecer de POV a producción masiva.
- **Cumplimiento:** facilita auditorías de SB, LOPDP y estándares internacionales.
- **FinOps:** costos controlados con presupuestos, Savings Plans y visibilidad por etiquetas.

Arquitectura:



6. Frontend (Web & Mobile)

6.0. Propósito y alcance

- **Propósito:** ofrecer una experiencia segura, rápida e intuitiva para banca digital 24/7 en **web** (SPA) y **móvil** (app iOS/Android).

- **Alcance:** autenticación OIDC/PKCE, dashboard, saldos/movimientos, transferencias, pagos, notificaciones, onboarding digital, gestión de perfiles y dispositivos.
- **Fuera de alcance (frontend):** reglas de negocio núcleo, idempotencia y compensaciones (se resuelven en BFF/servicios).

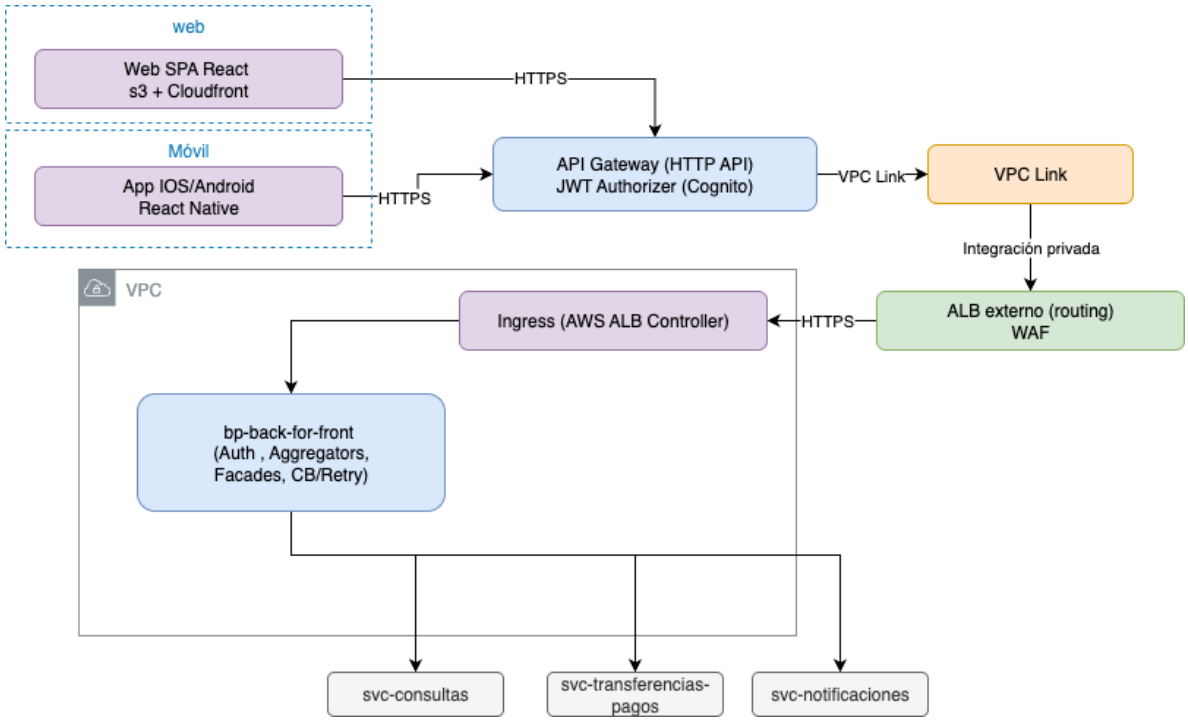
6.1. Arquitectura de Frontend

6.1.1. Web (SPA)

- **Stack:** React (SPA) servido desde **S3 + CloudFront**, detrás de **WAF**.
- **Autenticación: OIDC/PKCE** con **Cognito Hosted UI**; tokens en **memory** (o sessionStorage con rotación silenciosa); **nunca** localStorage para el ID/Access token.
- **Backends:** llamadas **REST** al **BFF** vía **API Gateway** → **ALB** → **EKS** (JWT en Authorization: Bearer).
- **Estado y datos:** React Query para cache/invalidación; normalización DTO desde BFF; loading incremental (skeletons).
- **Rendimiento:** code-splitting, lazy routes, HTTP/2 + gzip/brotli, imágenes responsivas; caché de estáticos en CloudFront.
- **Accesibilidad:** WCAG 2.1 AA (semántica, contraste, foco); i18n/es-EC.

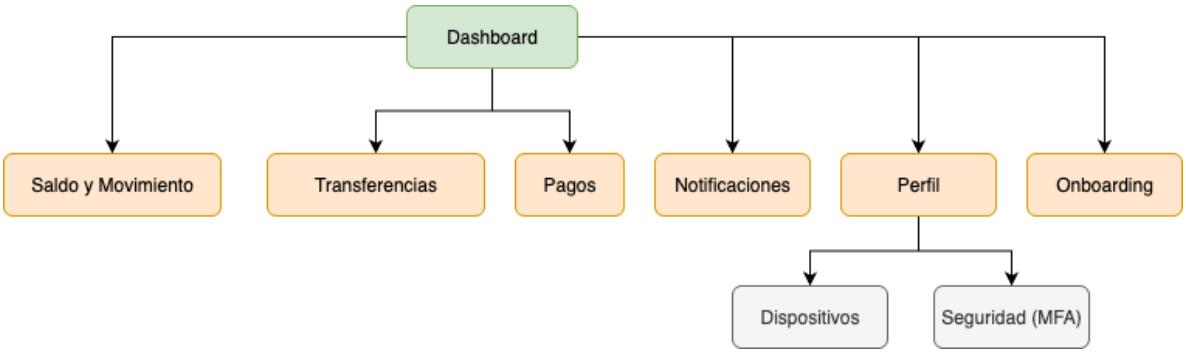
6.1.2. Móvil (IOS/Android)

- **Stack:** React Native con arquitectura MVVM.
- **Autenticación: OIDC/PKCE** con **Cognito**; almacenamiento de **refresh token** en **Keychain/Keystore**; **device binding** (claim “device_id”).
- **Seguridad del dispositivo:** detección root/jailbreak (lib nativa), ofuscación código, screenshot-blocking en pantallas sensibles.
- **Conectividad:** modo **offline-aware** (cola local de acciones no críticas, lectura desde cache), **retry** exponencial y estados “pendiente/en curso”.



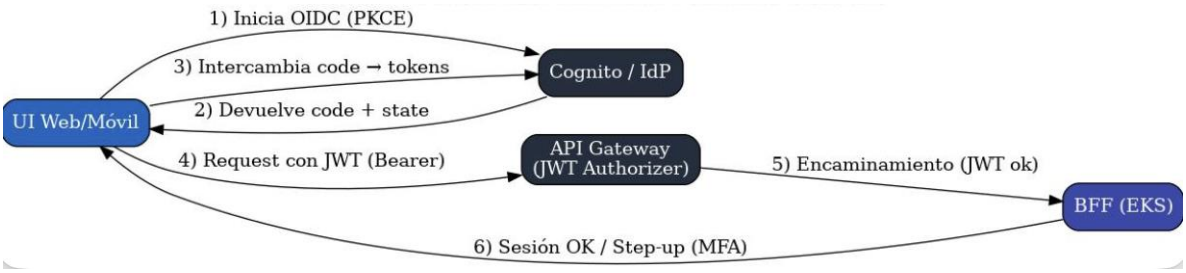
6.2. Módulos funcionales (UI)

- **Inicio de sesión & MFA:** flujo OIDC/PKCE, biometría del dispositivo (FaceID/AndroidBiometric) como **step-up** opcional.
- **Dashboard:** resumen de productos, accesos rápidos y alertas.
- **Saldos y movimientos:** filtro por rango/etiquetas, exportación (web), paginación infinita.
- **Transferencias/pagos (wizard guiado):** validación en cliente, **confirmación** clara, estado **asíncrono** para interbancarias y recibos descargables.
- **Notificaciones (centro):** preferencias opt-in/opt-out, horarios de silencio y email/SMS solo como reflectores.
- **Onboarding digital:** captura guiada, verificación de calidad (blur, brillo), feedback en tiempo real; consentimiento LOPDP.
- **Perfil y seguridad:** gestión de dispositivos/sesiones activas, cambio de PIN/clave, autenticadores (FIDO2/WebAuthn en web).



6.3. Seguridad y cumplimiento

- **Tokens:** rotación/refresh, expiraciones cortas, **no** persistir Access Token en disco móvil; CSRF mitigado (SPA + token en header).
- **PII mínima en memoria**, masking (número de cuenta parcial), sanitización de inputs y headers seguros.
- **LOPDP:** consentimiento explícito (biometría/onboarding), granularidad por canal y auditoría de preferencias.

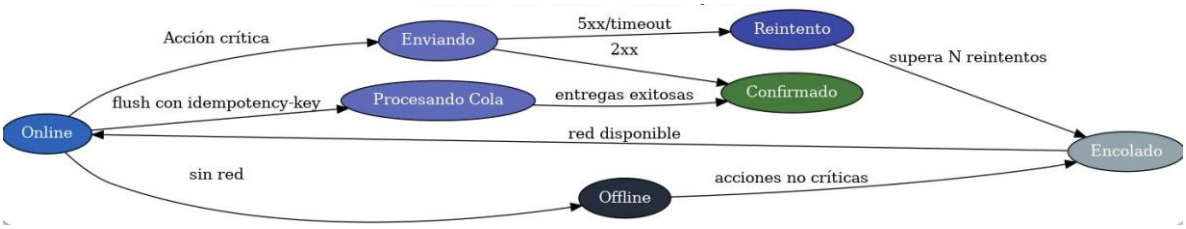


6.4. Observabilidad y calidad

- **Trazas/Logs de UX:** eventos de navegación, latencias percibidas, errores de red; envío a **CloudWatch** vía BFF (privacidad por defecto).
- **Métricas UX clave:** p95 login, p95 saldos, TTI (web), ANR/crash rate (móvil), entregabilidad push/email (como espejo).
- **Experimentos/Feature Flags:** toggles remotos (por ejemplo, AppConfig/LaunchDarkly) para despliegues graduales.

6.5. Rendimiento y resiliencia de la UI

- **Web:** prerender de shell, cache de assets (immutable), revalidación de datos (stale-while-revalidate), **optimización de imágenes** y fuentes.
- **Móvil:** cache normalizada, paginación, prefetch en Wi-Fi, reintentos con backoff y **idempotency-key** cliente→BFF para acciones críticas.



6.6. CI/CD y despliegue

- **Web:** build en CI → artefactos a **S3** → invalidación **CloudFront**; pruebas (unitarias, E2E con Playwright), Lighthouse budget.
- **Móvil:** pipelines por plataforma (Fastlane/GitHub Actions), firmas y **release tracks** (internal/beta/prod); **CodePush/OTA** sólo para UI (sin romper contratos).

6.7. Lineamientos UX

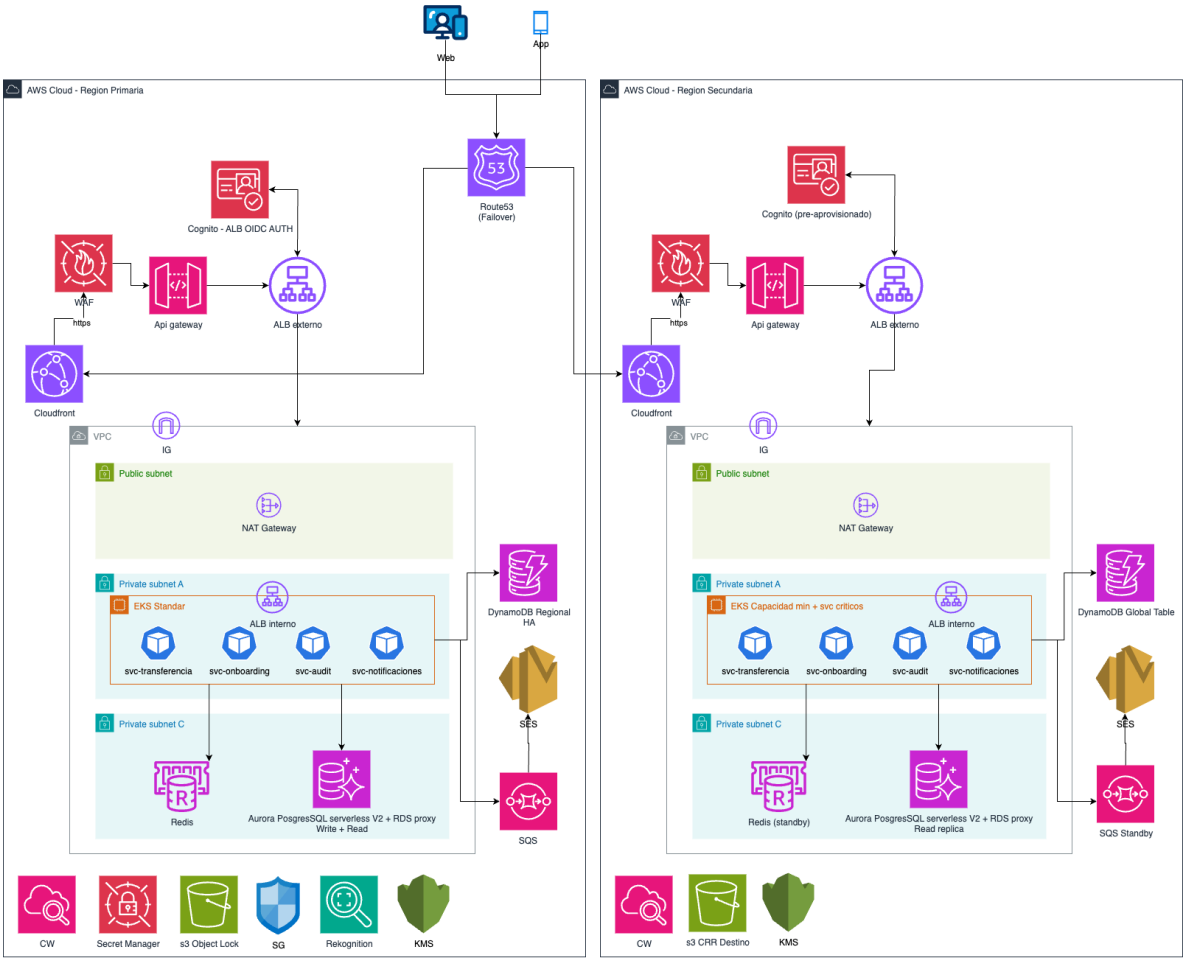
- Acciones críticas siempre con **doble confirmación** y resumen antes de enviar.
- Estados claros: *cargando* / *enviado* / *en curso* / *fallido* con **reintentar**.
- Inclusión: textos legibles, preferencia por contraste alto y soporte a lectores de pantalla.
- Soporte a **deep links** (p. ej., abrir una transferencia desde una notificación).

7. Caracteristicas transversales

7.0. Alta disponibilidad, resiliencia y DR

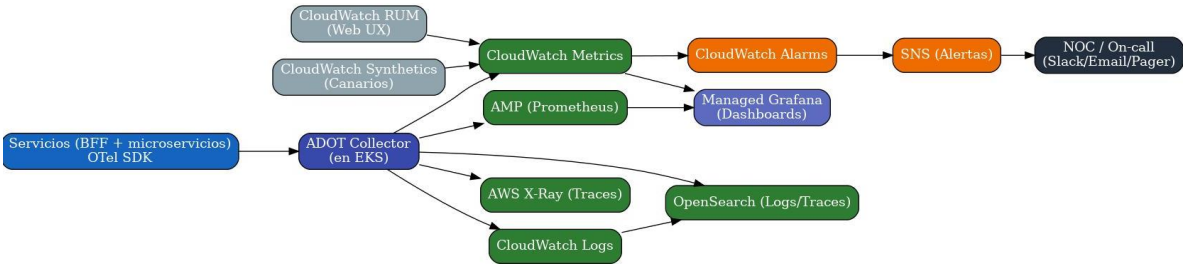
Alta disponibilidad & DR

- **Objetivo.** Seguir operativos ante fallas de AZ/región y recuperar con RTO ≤ 30 min, RPO ≤ 5 min.
- Alta disponibilidad intra-región. Multi-AZ en todo: EKS (3 AZ, PDB/anti-affinity/HPA), Aurora Multi-AZ + RDS Proxy, DynamoDB (regional), Redis con réplica/failover, ALB/API GW en subredes multi-AZ, CloudFront+WAF/Shield en perímetro.
  → Evita single points of failure y permite mantenimiento sin downtime.
- **DR multi-región (Warm Standby).** Secundaria con capacidad mínima y datos replicados:
  Aurora Global DB (read-only → promueve a writer), DynamoDB Global Tables, S3 CRR + Object Lock, EventBridge Global Endpoint, SQS en standby, KMS MRK, Cognito pre-provisionado.
  → Equilibrio costo/tiempo: replicas listas y arranque rápido.
- **Conmutación (runbook).** Congelar escrituras no críticas → promover Aurora → escalar EKS y calentar Redis → habilitar SQS/consumidores → cambiar orígenes (CF/ALB) y DNS (Route 53) → monitorear y aplicar contingencias.
- **Consistencia.** Idempotencia (DynamoDB), Transactional Outbox (SQS/EventBridge) y job de reconciliación (DLQ + verificación ledger Aurora) post-failover.
- **Pruebas & evidencias.** Canarios por flujo (login/saldo/transferencia), SLO y burn-rate, game days/chaos (derribo de AZ, caída Core/BCE, failover Aurora).
- **Seguridad en DR.** WAF/Shield en ambas regiones, IAM separado por región, CloudTrail/Config multi-región y copia de logs en cuenta de seguridad.
- **Costos.** Warm standby = mínimo en secundaria y escalado on-demand; control con Budgets y etiquetado FinOps.



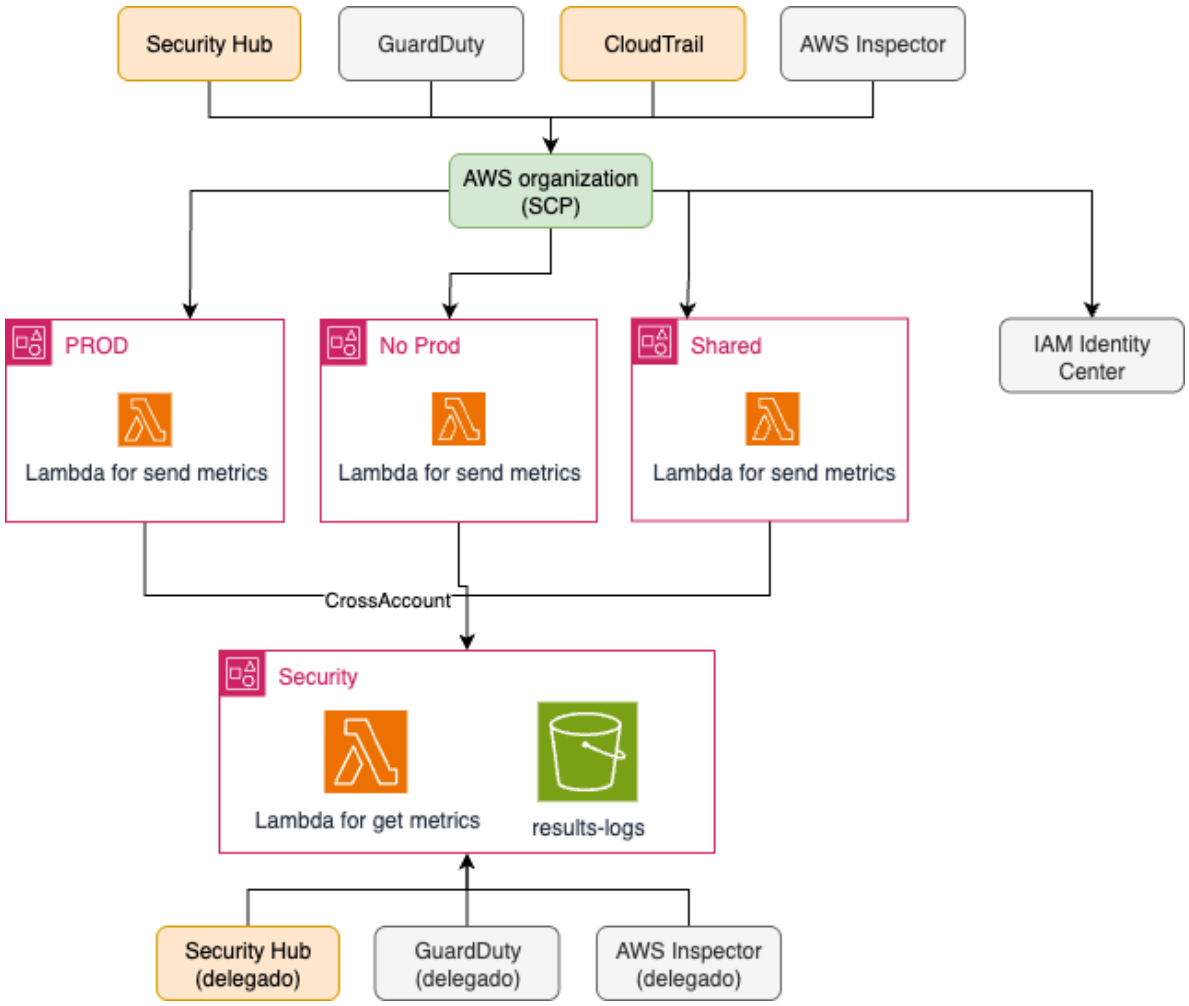
7.1. Observabilidad y monitoreo

- **Qué medimos.** Métricas (latencia, errores, tráfico, saturación), logs estructurados (sin PII), trazas E2E (W3C trace context) y **canarios**.
- **Cómo se instrumenta. OpenTelemetry** (SDK + **ADOT Collector** en EKS) publica en **CloudWatch Metrics/Logs**, **X-Ray** (traces) y **OpenSearch**; **AMP + Grafana** para dashboards. **RUM** para Web y **Synthetics** multi-región.
- **Alertado. CloudWatch Alarms → SNS → NOC/On-call** con rutas P1/P2/P3 y *burn-rate* SLO.
- **Cumplimiento.** Cifrado en tránsito/reposo, retención, **no PII** en logs, acceso por roles.



7.2. Seguridad y cumplimiento normativo

- **Gobierno. Multi-cuenta** con **SCPs**, **Cuenta de Seguridad** centraliza CloudTrail (S3 WORM), **Security Hub** y **GuardDuty** agregados. Accesos con **IAM Identity Center (MFA)**.
- **Accesos mínimos.** Roles por servicio (IRSA en EKS), rotación de secretos y duración corta.
- **Datos personales.** LOPDP: minimización, consentimiento biométrico, retención/borrado y control de transferencias internacionales.
- **Cripto & red.** TLS 1.2+, KMS/MRK, ACM/Private CA, VPC endpoints, WAF/Shield, PrivateLink/Direct Connect cuando aplica.
- **DevSecOps.** Inspector (ECR/Lambda/EC2), OPA/Gatekeeper, SAST/DAST/IaC scanning, SBOM y firmas (cosign).
- **Respuesta a incidentes.** Security Hub+GuardDuty → playbooks (SNS/SSM), Detective forense, reporte a regulador si aplica.
- **Automatización de envio de hallazgos:** La estrategia consiste en centralizar los hallazgos de seguridad de todas las cuentas principales en una **cuenta de seguridad** dedicada.
    - **Delegación de servicios de seguridad**
      Primero, se delegan los servicios de seguridad (como GuardDuty, Security Hub, etc.) hacia la cuenta de seguridad, para que actúe como punto central de gobierno.
    - **Envío de hallazgos mediante Cross-Account**
      Los hallazgos generados en las cuentas principales se envían hacia la cuenta de seguridad a través de **Lambdas** que escriben directamente en un **bucket S3** de dicha cuenta, usando permisos cross-account.
    - **Procesamiento centralizado**
      Una vez almacenados en S3, se utiliza otro Lambda en la cuenta de seguridad para procesar la información y normalizarla si es necesario.
    - **Consulta de logs y hallazgos**
      Finalmente, los datos procesados en S3 pueden analizarse fácilmente con **Athena**, lo que permite consultas SQL sobre los hallazgos de seguridad consolidados.



7.3. Costos y optimización

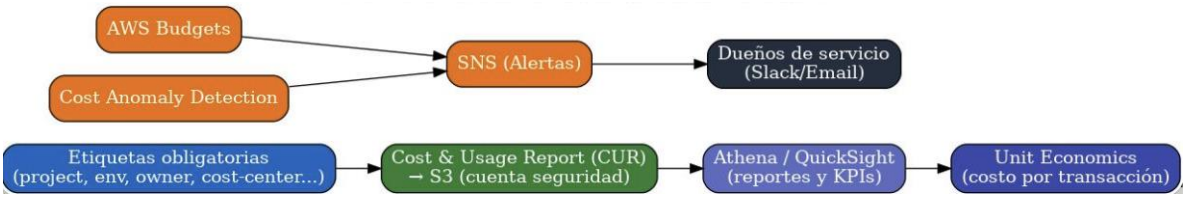
 Visibilidad. Etiquetas obligatorias (project, env, owner, cost-center…), **CUR a S3**, consultas con **Athena/QuickSight** y **KPIs por transacción** (login, saldo, transferencia, notificación).

Control. AWS Budgets y **Cost Anomaly Detection** con alertas a **SNS** hacia dueños de servicio.

Optimización.

- Cómputo: rightsizing/HPA/Karpenter, **Savings Plans** para base.
- Datos: Aurora (índices correctos, RIs), DynamoDB (TTL y autoscaling RCU/WCU), Redis (evitar hot keys).
- Red: minimizar NAT/egress, CloudFront para bajar salida.

- Observabilidad: retención hot→cold y muestreo de trazas.



8. Operación y gobierno

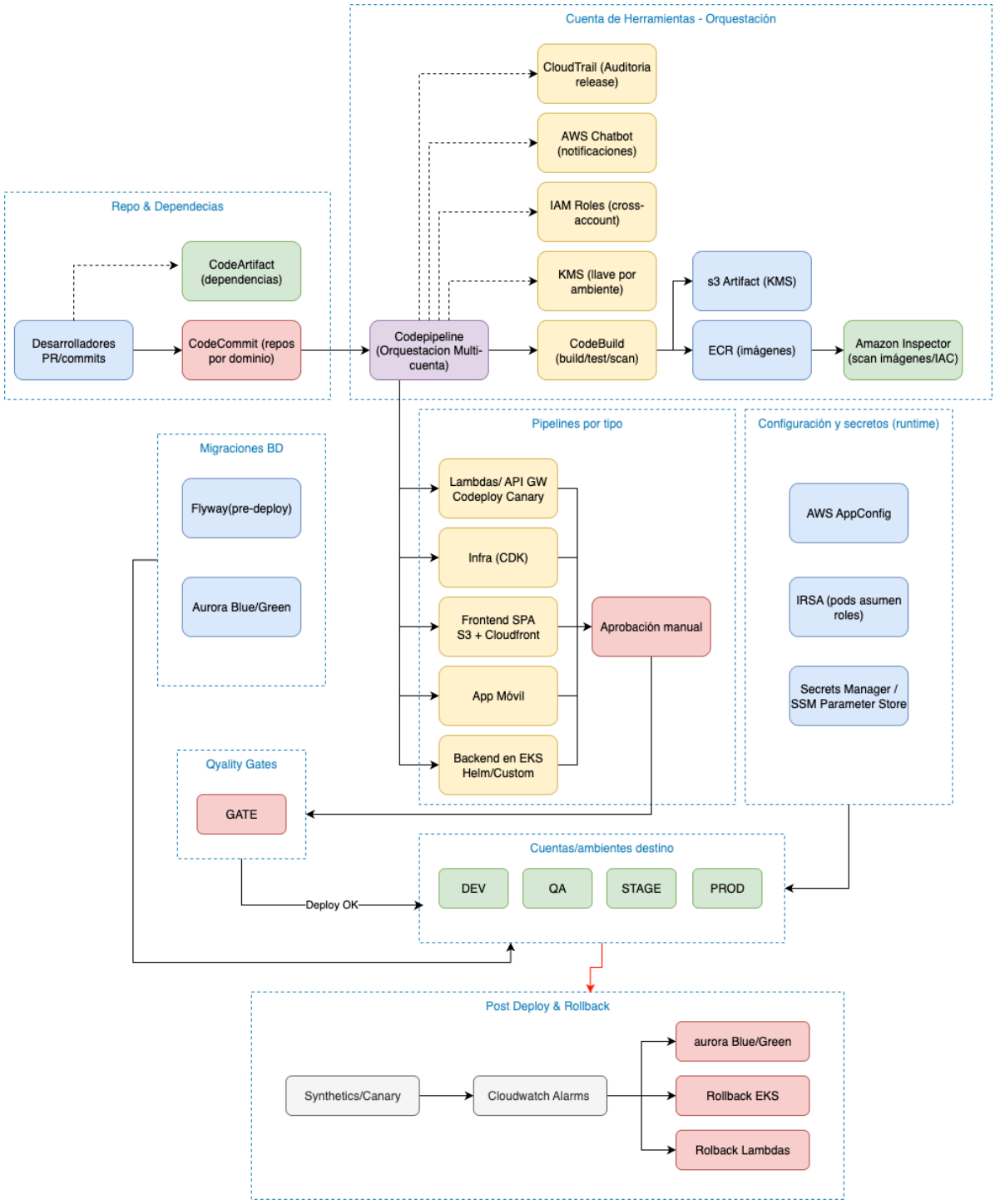
8.0. CI/CD y gobernanza

Objetivo. Entregar cambios **seguros, trazables y reversibles** desde el commit hasta producción, con **segregación por cuentas/ambientes** y controles de cambio auditables.

Implementación en AWS:

- **Repos & ramas.** CodeCommit por dominio; trunk-based con PR y 2 revisores → menos fricción y mejor auditoría (CloudTrail).
- **Orquestación multi-cuenta.** CodePipeline/CodeBuild en **cuenta de herramientas**; despliegues **cross-account** a dev/qa/stage/prod mediante roles IAM y KMS por ambiente → separación de funciones (SoD).
- **Artefactos y cadena de suministro.**
  - S3 (artifact) + CodeArtifact (deps).
  - ECR (imágenes) con **Inspector**.
  - **SBOM + firma (cosign)** y verificación en admisión (EKS/OPA).
- **Pipelines por tipo:**
  - **Frontend SPA:** build → S3 → invalidación selectiva de CloudFront → canarios (Lighthouse/login/saldos) → aprobación manual stage→prod.
  - **App móvil:** build firmado (llaves en Secrets Manager) → artefacto a S3 → pruebas UI/emu + canarios → publicación manual a stores.
  - **Backend EKS:** unit + contract + integración → push ECR (firmado/escaneado) → deploy Helm/Kustomize con IRSA → **canary por ALB** → **gates por Alarms/X-Ray** → rollback auto.
  - **Lambdas/API Gateway:** CodeDeploy **canary 10%→100%** con rollback automático por alarmas.
  - **Infra (CDK):** synth + **cdk-nag** → diff → **aprobación manual** → deploy por ambiente (Infra as Code auditada).
- **Migraciones BD.** Flyway/Liquibase en pre-deploy con patrón **expand** → **deploy** → **contract**; **Aurora Blue/Green** para cambios críticos.

- **Configuración y secretos. AppConfig** con **safe rollouts (1%→5%→100%)**; secretos y parámetros en **Secrets Manager/SSM**, inyectados en runtime vía **IRSA** (sin secretos en imagen).
- **Gobernanza de cambios.** Aprobación dual (Negocio + Operaciones/TI), evidencias (PR, pruebas, SBOM), bitácora (CodePipeline + CloudTrail + ticket), notificaciones con **AWS Chatbot**.
- **Post-deploy.** Smoke tests (canarios), **gates por alarmas** (p95, 5xx, colas, lag DB) y **rollback**: Lambda (CodeDeploy), EKS (helm rollback), Aurora (Blue/Green). MTTR bajo.


Cuenta de Herramientas - Orquestación
CloudTrail (Auditoria release)
AWS Chatbot (notificaciones)
IAM Roles (cross-account)
KMS (llave por ambiente)
s3 Artifact (KMS)
Repo & Dependecias
CodeArtifact (dependencias)
Desarrolladores PR/commits
CodeCommit (repos por dominio)
Codepipeline (Orquestacion Multi-cuenta)
CodeBuild (build/test/scan)
ECR (imágenes)
Amazon Inspector (scan imágenes/IAC)
Migraciones BD
Flyway(pre-deploy)
Aurora Blue/Green
Pipelines por tipo
Lambdas/ API GW Codeploy Canary
Infra (CDK)
Frontend SPA S3 + Cloudfront
App Móvil
Backend en EKS Helm/Custom
Aprobación manual
Configuración y secretos (runtime)
AWS AppConfig
IRSA (pods asumen roles)
Secrets Manager / SSM Parameter Store
Qyality Gates
GATE
Deploy OK
Cuentas/ambientes destino
DEV
QA
STAGE
PROD
Post Deploy & Rollback
Synthetics/Canary
Cloudwatch Alarms
aurora Blue/Green
Rollback EKS
Rolback Lambdas

8.1. Pruebas y calidad

Objetivo. Asegurar funcionalidad, rendimiento, seguridad, accesibilidad y cumplimiento **antes** de producción, con **gates automáticos** en el pipeline y evidencia auditada.

Estrategia práctica:

- **Pirámide de pruebas:** unitarias → contrato (Pact) → integración → E2E (Playwright/Device Farm).
- **Ambientes efímeros por PR** (namespace EKS): integraciones y E2E aisladas.
- **Datos de prueba sintéticos/enmascarados (LOPDP)**; cero PII real en no-prod.
- **Performance** con k6 (soak, stress, picos) y telemetría OTel/X-Ray; bloquear promoción si p95 excede +20% del objetivo.
- **Resiliencia/Chaos** con AWS FIS: caída AZ, latencia/errores a Core/BCE/KYC, saturación colas, killing de pods; validar circuit breaker, sagas e idempotencia.
- **Seguridad:** SAST/DAST, Inspector (imágenes), **SBOM + firma**, OPA/Gatekeeper (no root, FS read-only, límites), OWASP ASVS L2. **Cero críticos/altos** abiertos.
- **Accesibilidad/UX:** axe + Playwright (WCAG 2.1 AA), RUM para LCP/CLS/INP; sin violaciones críticas y LCP p75 ≤ 2.5 s.
- **UAT y regulatorias (Ecuador):** evidencias LOPDP, pruebas AML/CFT y plantillas de notificación revisadas.
- **Observabilidad de pruebas:** correlación traceId/txnId, logs estructurados, evidencias en S3 WORM y paneles de release.

9. Roadmap de implementación

Fases y hitos (visión general)

Fase	Objetivo	Duración estimada	Entregables clave	Criterio de salida (Go/No-Go)
0. Preparación	Alinear alcance, riesgos, regulador y equipos	1–2 semanas	Plan del proyecto, matriz de riesgos, responsables, cronograma base	Aprobación de Negocio, Riesgos, TI y Seguridad
1. Descubrimiento & Diseño	Historias de usuario y experiencia; diagrama C4 y flujos principales	2–3 semanas	Historias priorizadas, wireframes, C4 actualizado, plan de datos y seguridad	Revisión de stakeholders y señal verde de Seguridad
2. POV (prueba de valor)	Probar la idea punta a punta a bajo costo	4–6 semanas	Login + consulta de saldos + transferencia propia en ambiente de prueba	Demostración funcional, métricas p95 aceptables, sin bloqueos críticos
3. MVP	Versión mínima utilizable por clientes	8–12 semanas	Web/App, saldos/movimientos, transferencias propias, notificaciones email/SMS, auditoría	UAT aprobado, pentest sin hallazgos críticos, mesa de ayuda lista
4. Piloto interno	Probar con colaboradores del banco	2 semanas	100–300 usuarios internos, manuales rápidos, canal de soporte	Tasa de éxito ≥ 95%, defectos corregidos, satisfacción ≥ 4/5
5. Piloto controlado (clientes)	Salir con un grupo pequeño real	4–6 semanas	1–5% de clientes, monitoreo 24/7, comunicación clara	SLO cumplidos (disponibilidad/latencia), quejas < 1%, sin incidentes de seguridad
6. Producción Fase 1	Go-Live amplio y estabilización	2 + 4 semanas	Lanzamiento general, operación 24/7, runbooks y on-call	Estabilización (4 semanas) con SLO ≥ 99.9%, sin reprocesos monetarios

Fase	Objetivo	Duración estimada	Entregables clave	Criterio de salida (Go/No-Go)
7. Ampliaciones	Más funciones y canales	8–12+ semanas	Interbancarias, pagos de servicios, WhatsApp, mejoras UX, accesibilidad	Entregables por sprint con KPIs de uso y costo por transacción
8. Operación & Mejora continua	Optimizar y crecer	Permanente	Reportes mensuales, FinOps, seguridad continua, roadmap trimestral	Revisiones trimestrales con negocio y regulatorio interno

Notas: Duraciones son estimadas; se afinan con el banco. Cada fase incluye: seguridad, privacidad (LOPDP), accesibilidad y evidencia para auditoría.

Qué haremos en cada fase

- Preparación: alinear objetivos, armar el equipo, definir presupuesto y riesgos principales.
- Descubrimiento & Diseño: acordar la experiencia del cliente, dibujar “cómo conversa todo” (diagramas C4), y elegir qué entra primero.
- POV: montar lo mínimo en AWS para probar la idea end-to-end con pocos usuarios.
- MVP: construir lo necesario para que el cliente ya pueda usar banca en línea sin llamadas al banco.
- Pilotos: primero colaboradores, luego un grupo pequeño de clientes; medir, aprender y corregir.
- Producción Fase 1: salir para todos con monitoreo 24/7 y equipo de soporte listo.
- Ampliaciones: agregar pagos, interbancarias, WhatsApp, mejoras de usabilidad y accesibilidad.
- Operación: revisar métricas de negocio, costos y seguridad cada mes; planificar el siguiente trimestre.

Roles y responsabilidades

- Negocio: prioriza funcionalidades y aprueba mensajes/comunicaciones.
- Riesgos/Fraude: define umbrales, reglas y excepciones.
- Seguridad/Privacidad: revisa controles (MFA, cifrado, LOPDP).
- TI/Arquitectura: diseña y valida la solución en AWS.
- Desarrollo (Front/Back): construye y prueba.
- Operaciones/Soporte: runbooks, monitoreo y mesa de ayuda.
- Comunicaciones/Marketing: plan de lanzamiento y tutoriales.
- Legal/Compliance: valida textos, disclaimers y contratos.

Comunicación y adopción

- Antes del piloto: FAQs, tutorial en video (2–3 min), guía paso a paso.
- Durante piloto: canal abierto (WhatsApp/Chat del banco) y tiempos de respuesta claros.
- Go-Live: correos y notificaciones en App con mensajes simples (“qué hay de nuevo” y “cómo hacerlo”).
- Capacitación interna: sesiones de 1–2 horas para canales, call center y sucursales.

Puertas de calidad (Go/No-Go) en cada salida

- Funcional: casos críticos ok (login, saldos, transferencias).
- Rendimiento: p95 dentro de objetivo.
- Seguridad: sin hallazgos críticos.
- Soporte: mesa de ayuda y runbooks listos.
- Comunicación: materiales publicados.