Joe Takeshi Valerio Yachachin

I. Resumen

La propuesta plantea una **arquitectura de integración híbrida** que permita al banco avanzar en su proceso de modernización, garantizando la convivencia entre el core bancario tradicional y un nuevo core digital.

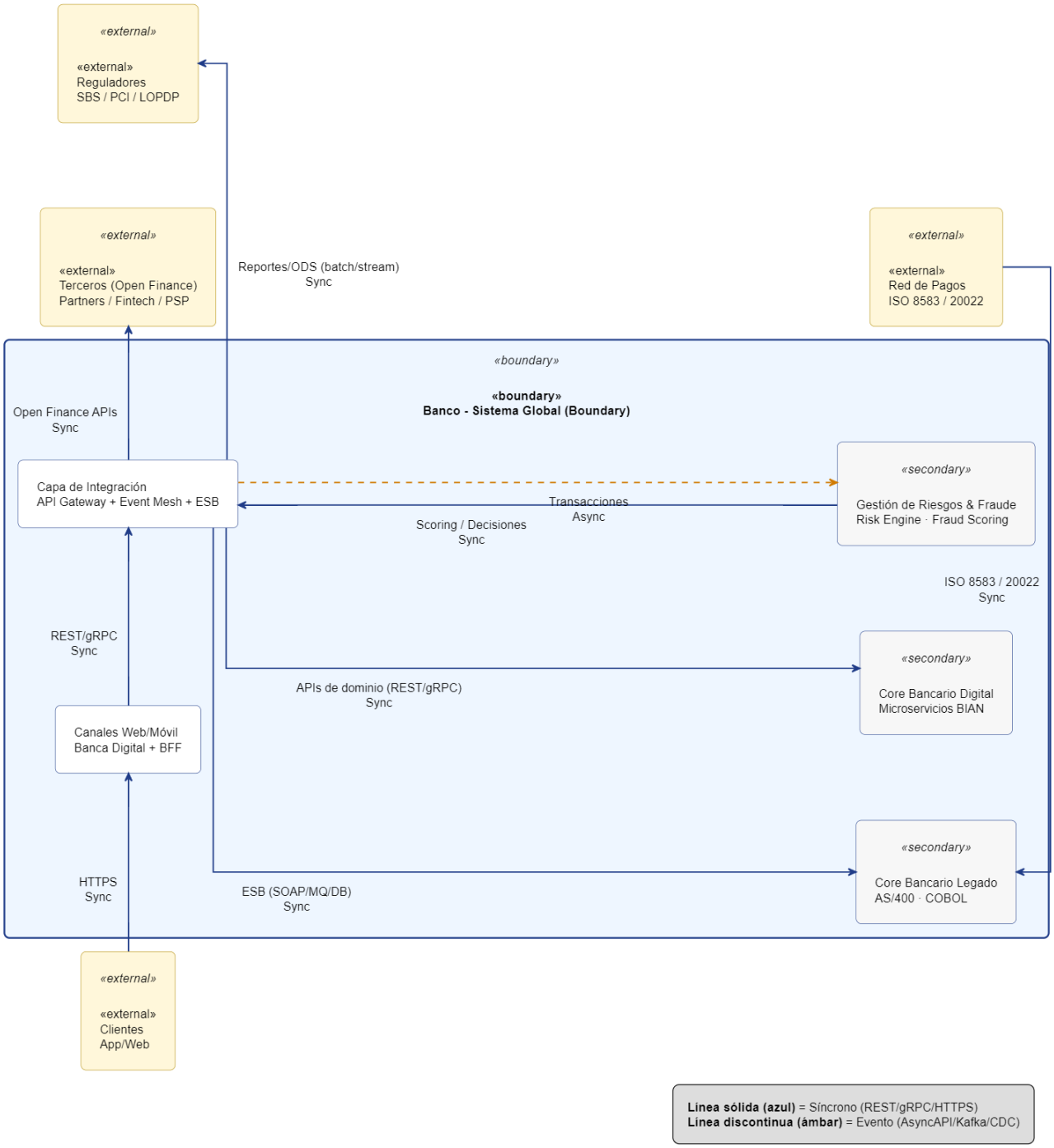
El diseño se apoya en tres pilares clave:

1. **Estandarización e interoperabilidad**
    - Uso de **APIs abiertas** (OpenAPI/AsyncAPI) y **eventos de negocio** para habilitar la comunicación ágil y segura entre los distintos sistemas.
    - Aplicación de **patrones de integración desacoplados** (Strangler Fig, Anti-Corruption Layer) que reducen la complejidad y el riesgo en la evolución tecnológica.
2. **Gobernanza y seguridad**
    - Modelo de **gobierno de APIs** alineado al estándar **BIAN**, lo que asegura consistencia, escalabilidad y cumplimiento regulatorio.
    - Seguridad basada en **OAuth2/OIDC**, **mTLS interno** y cumplimiento de marcos regulatorios y normativos: **LOPDP (Ley N.º 29733)**, **PCI DSS** e **ISO 27001**.
3. **Disponibilidad y migración controlada**
    - Estrategia de **alta disponibilidad (HA)** y **recuperación ante desastres (DR)** con un esquema **activo-activo Multi-AZ**, asegurando continuidad del servicio.
    - **Plan de migración gradual por dominios funcionales** (pagos, cuentas, clientes), diseñado para minimizar riesgos y garantizar una transición ordenada.

En conjunto, esta arquitectura proporciona una base sólida para que el banco evolucione hacia un **ecosistema digital más ágil, seguro y resiliente**, respondiendo a las nuevas demandas de clientes, reguladores y del mercado financiero.

II. C4 – Nivel 1: Diagrama de Contexto

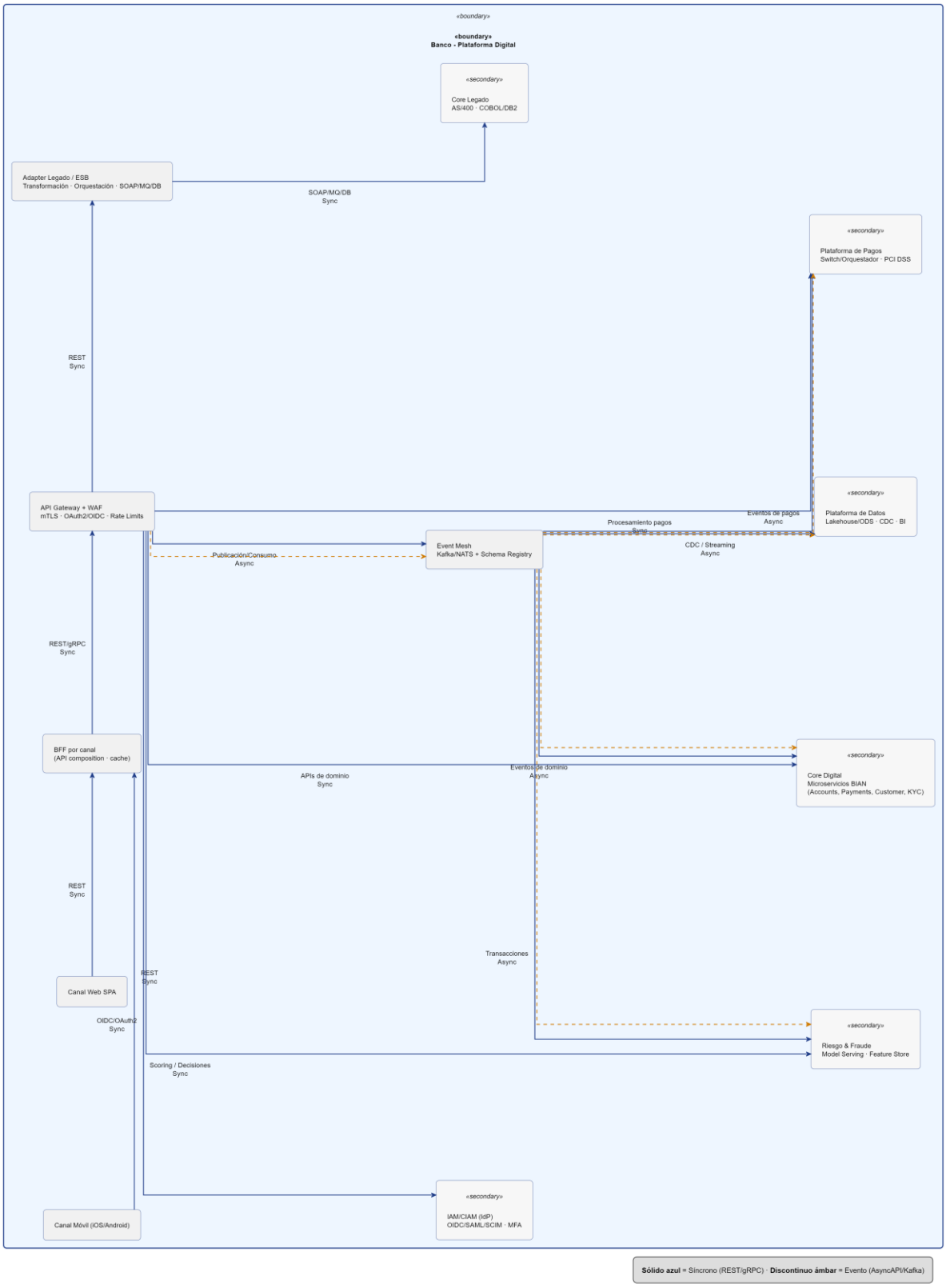
Relación de actores externos, canales, cores y servicios transversales (riesgo/fraude, pagos, reguladores)



Notas: Se adopta BIAN como referencia de dominios (Service Domains) para estandarizar interacciones. Open Finance expone APIs externas con políticas y contratos claros vía API Gateway y API Manager.

III. C4 – Nivel 2: Diagrama de Contenedores

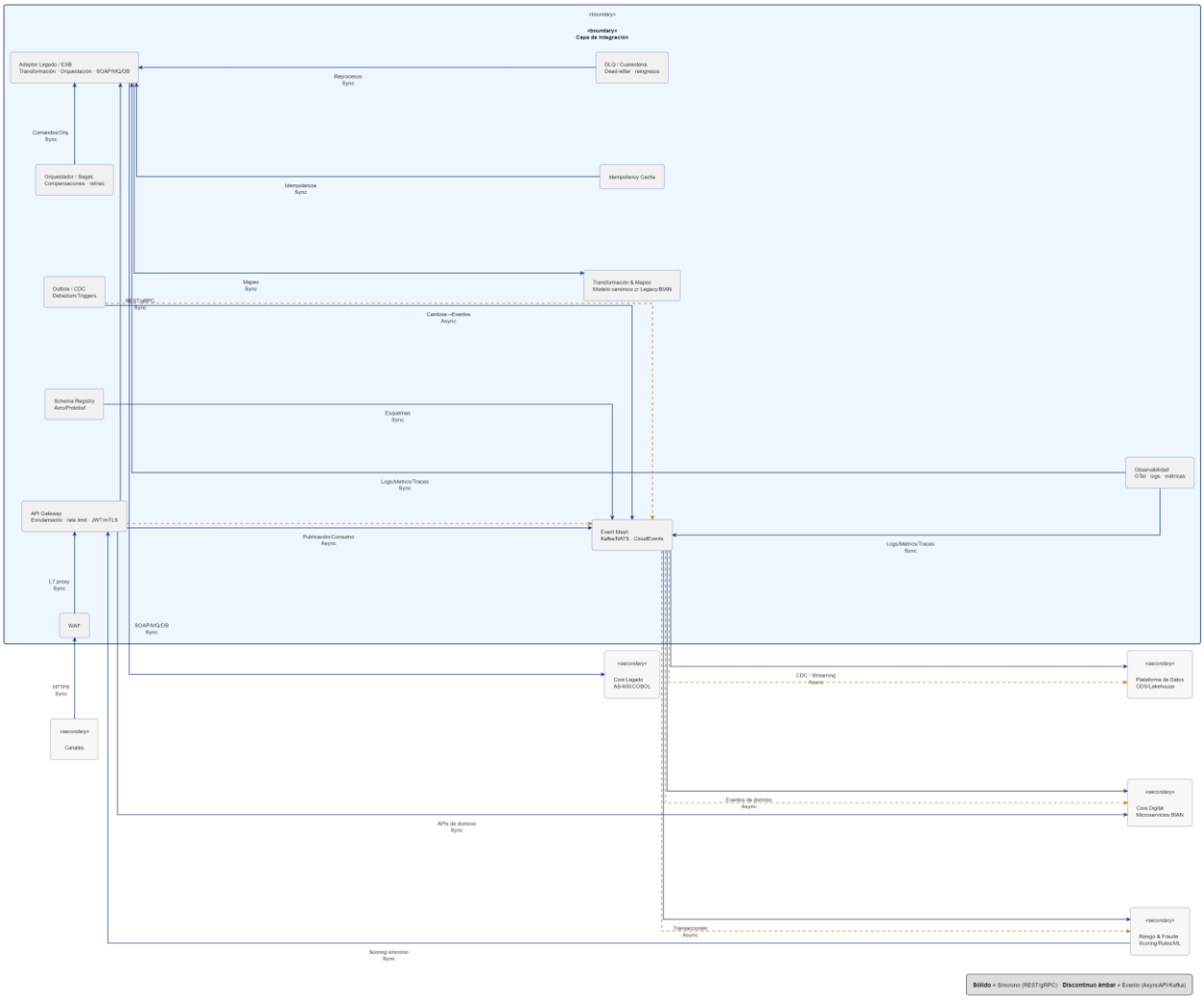
Vista de contenedores: canales, borde API, malla de eventos/ESB, cores, pagos, datos e IAM.



Decisiones clave: BFF por canal para optimizar latencia; gRPC interno y REST en el borde; eventos para dominios de alta asincronía; ESB para encapsular protocolos del legado.

IV. C4 – Nivel 3: Componentes – Capa de Integración (Adapter Legado + Eventing)

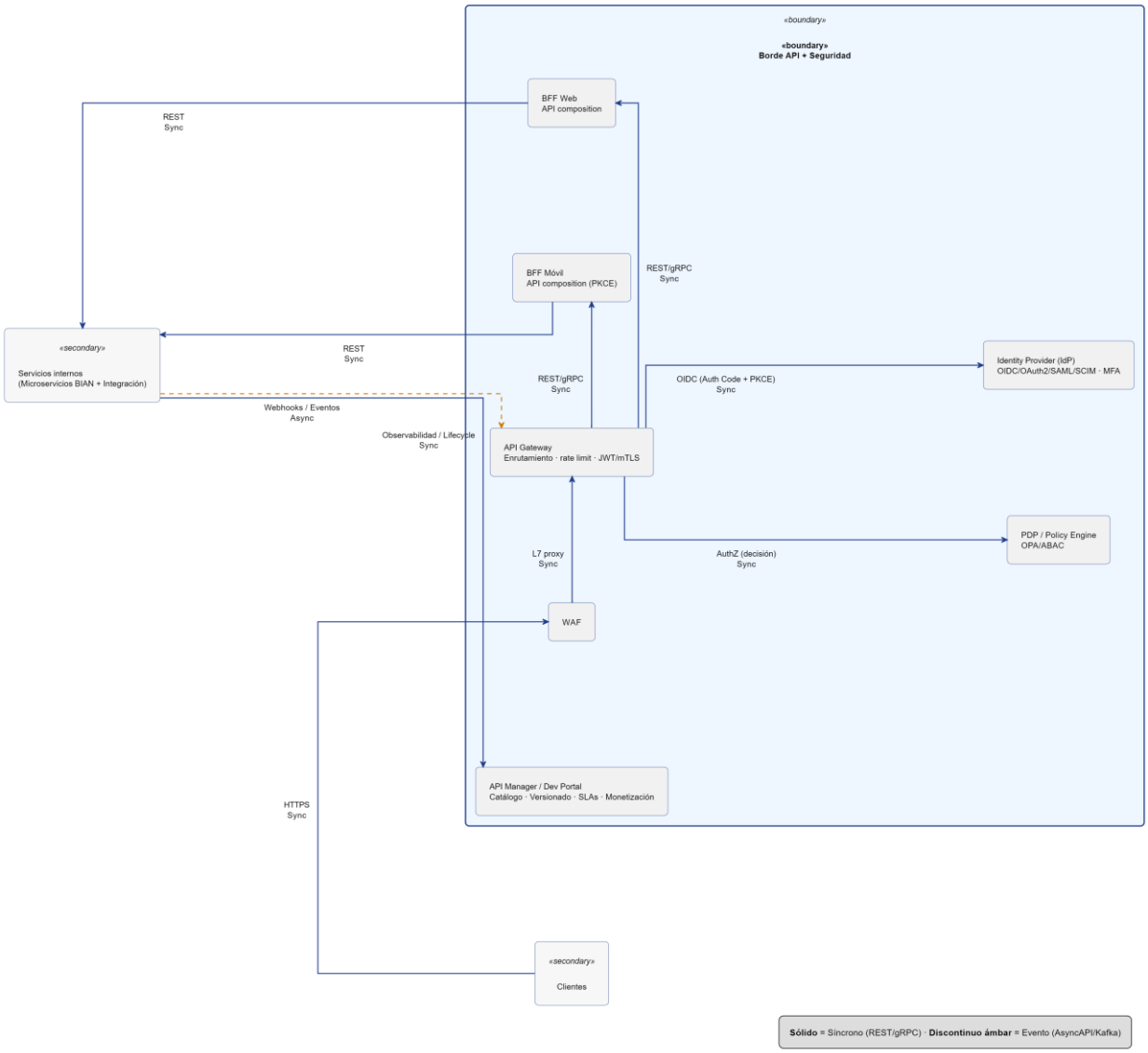
Muestra componentes esenciales para desacoplar el core legado y convertir cambios en eventos de negocio.



Permite migración progresiva; evita contaminar dominio del nuevo c

V. C4 – Nivel 3: Componentes – Borde API + Seguridad

Control de acceso, autenticación/autorización y gobierno del ciclo de vida de APIs.



VI. Patrones de integración y tecnologías

Área	Patrón/Decisión	Justificación / Tecnologías
Modernización	Strangler Fig + Anti-corruption Layer	Permite migración progresiva; evita contaminar el dominio del nuevo core. ESB/API Façade para legado.
Consistencia	Saga (orquestada y coreografiada)	Compensaciones en procesos distribuidos (ej. pagos). Orquestador ligero o coreografía por eventos.
Mensajería	Outbox + Idempotency + DLQ	Entrega exactamente-una-vez lógico, reprocesos seguros. Kafka/NATS + Schema Registry (Avro/Protobuf).

Lectura/Escritura	CQRS selectivo	Optimiza consultas en canales; reduce acoplamiento de escritura transaccional.
API Edge	BFF, API Composition	Optimiza payloads y latencia por canal. gRPC interno, REST externo, GraphQL opcional.
Resiliencia	Circuit breaker, Bulkhead, Retry/Backoff	Aísla fallas y evita cascadas. Librerías de resiliencia / Service Mesh.
Datos	CDC (Debezium) + Lakehouse	Replica cambios para analítica casi en tiempo real y ODS regulatorio.

VII. Seguridad, cumplimiento normativo y protección de datos

- Identidad y acceso: Implementación de OAuth2/OIDC con IdP para clientes (CIAM) y colaboradores (IAM), uso de MFA, mTLS interno y provisión mediante SCIM.
- Protección de datos: Cumplimiento de la LOPDP (Ley N° 29733) y su Reglamento, aplicando los principios de consentimiento, finalidad, minimización y almacenamiento seguro.
- Pagos: Adopción de PCI DSS, con tokenización de PAN, segmentación de red y gestión segura de secretos en vaults.
- Gobierno de seguridad: Alineamiento a normas internacionales como ISO/IEC 27001 y NIST CSF.
- Seguridad en tránsito y en reposo: Uso de TLS 1.2+ para comunicaciones seguras y cifrado AES-256 en reposo, con administración de claves mediante KMS/HSM.
- Telemetría y monitoreo: Implementación de SIEM, detección de anomalías, auditoría con registros inmutables, DLP y clasificación de datos.
- Privacidad por diseño: Pseudonimización de información sensible, políticas de retención y borrado seguro, gestión de registros de consentimiento y evaluación de impacto para terceros (TIA).

VIII. Alta disponibilidad (HA) y recuperación ante desastres (DR)

- Topología: Despliegue activo-activo Multi-AZ con balanceo global. Componentes stateful con clustering (Kafka / RDBMS con replicación síncrona o semisíncrona).
- Objetivos: RPO ≤ 5 min (eventos y bases con CDC); RTO ≤ 30 min (automatización vía IaC).
- Estrategias: Blue/Green y canary releases en API Gateway; backups incrementales + snapshots; runbooks de conmutación; pruebas de caos y simulacros de DR.
- Dependencias: SLOs definidos por dominio (latencia p95 ≤ 300ms en canales; disponibilidad ≥ 99.95%).

IX. Estrategia de integración multicore

- Ruteo por dominio/producto: Ej. cuentas en core digital; ciertos pasivos/activos en core legado.
- Modelo canónico: Estándar BIAN para contratos de datos y servicios; Anti-Corruption Layer (ACL) para proteger dominios.

- Sincronización: Eventos como source of truth con réplica eventual y reconciliación diaria.
- Switch transaccional: Implementado en el Adapter para transacciones que permanecen en el legado.

X. Gestión de identidad y acceso (IAM/CIAM)

- Autenticación: OIDC (con PKCE en móviles), MFA / biometría, device binding.
- Autorización: RBAC/ABAC con PDP (ej. OPA) y políticas alineadas a dominios BIAN.
- Seguridad de sesión: JWT de corta duración, refresh tokens rotados; signed cookies en web; detección de anomalías.
- Gestión de usuarios: SCIM para alta/baja/cambios; Just-In-Time provisioning para terceros bajo consentimiento.

XI. Estrategia de APIs internas y externas

- Estándares: OpenAPI 3.x para REST; AsyncAPI para eventos; gRPC para tráfico service-to-service de baja latencia.
- Versionado: SemVer en rutas y contratos; coexistencia v1/v2 con deprecación controlada.
- Mensajería: CloudEvents en encabezados; Avro/Protobuf en bus de eventos; idempotency keys y correlación (trace-id).
- Políticas: Rate limiting por aplicación/usuario; cuotas y monetización para Open Finance; adaptive throttling.

XII. Gobierno de APIs y microservicios

- Catálogo: API Manager con registro de contratos (OpenAPI/AsyncAPI), SLAs y ownership (RACI).
- Calidad: Linters (Spectral), pruebas de seguridad (SAST/DAST), contract testing (Pact), validación de esquemas.
- Lifecycle: Flujo de diseño → revisión → publicación → monitoreo → deprecación; decisiones mediante ADRs.
- Operación: Observabilidad (logs, métricas, trazas); SLOs y error budgets; playbooks y runbooks; gestión centralizada de secretos.

XIII. Plan de migración gradual (minimizando riesgo)

- Fase 0 – Fundaciones: Landing zone, red, observabilidad, CI/CD, API Gateway, Event Mesh, IdP.
- Fase 1 – Strangler por dominios: Iniciar con Pagos (alto valor y aislable), luego Cuentas y Clientes. Activar ACL/Adapters y publicar eventos.
- Fase 2 – Paralelo / Shadow: Doble escritura controlada + reconciliación; feature flags y route by header para usuarios piloto.
- Fase 3 – Corte controlado: Canales apuntan al nuevo core por producto/segmento; plan de rollback con toggles.

- Fase 4 – Decommissioning: Retiro progresivo de interfaces del legado; archivado seguro y evidencias regulatorias.

XIV. Monitoreo y observabilidad

- Métricas: p95/p99, throughput, errores por endpoint/tópico, consumer lag.
- Logs: Estructurados con trace-id / span-id y user-id anonimizado.
- Trazas: OpenTelemetry + collector; vistas de dependencias y costos por llamada.
- Alertas: Basadas en SLOs; gestión de guardias (on-call) con runbooks y autorremediación.

XV. Notas de infraestructura

- La solución puede desplegarse en AWS o Azure, utilizando servicios gestionados:
- Compute / Orquestación: EKS / AKS
- Eventos: MSK / Event Hubs
- APIs: API Gateway / API Management
- Datos: RDS / Managed SQL