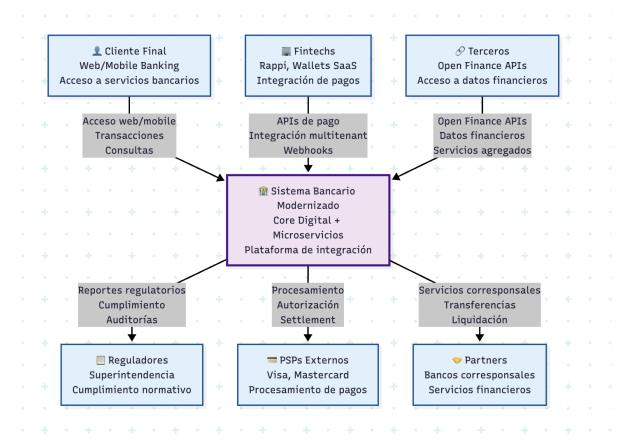
Diagrama de Contexto

Actores Externos:

- Cliente Final: Usuarios que acceden a servicios bancarios a través de web y mobile
- Fintechs: Empresas como que integran servicios de pago
- Terceros: Proveedores que acceden a APIs de Open Finance
- Reguladores: Superintendencias que requieren cumplimiento normativo
- PSPs Externos: Procesadores como Visa/Mastercard para pagos
- Partners: Bancos corresponsales para servicios financieros



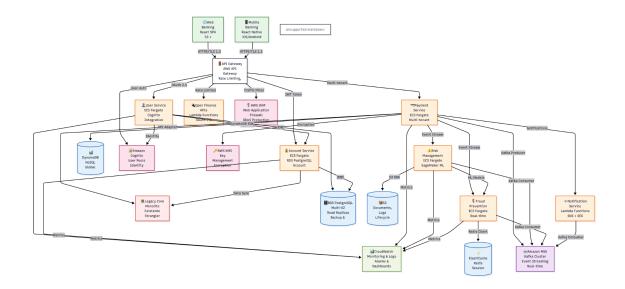
Flujos Principales:

- Acceso de Clientes: Cliente → Sistema Bancario (Web/Mobile Banking, APIs REST)
- Integración Fintech: Fintech → Sistema Bancario (APIs REST, Webhooks, OAuth 2.0)
- Open Finance: Terceros → Sistema Bancario (Open Finance APIs, OAuth 2.0)
- Cumplimiento: Sistema Bancario → Reguladores (APIs de reporte, SFTP)

Diagrama de Contexto

Diagrama de Contenedores

La arquitectura de contenedores detalla los componentes principales, tecnologías AWS y sus interacciones.



Capas de la Arquitectura:

- 1 Front End Layer:
 - Web Banking: React SPA con CDN global
 - Mobile Banking: React Native cross-platform

2 - API Gateway Layer:

- AWS API Gateway: Punto de entrada único
- Características: Rate limiting, autenticación, autorización
- Protocolos: OAuth 2.0, JWT, HTTPS/TLS 1.3

3 - Microservicios Core:

- User Service: Gestión de identidad y usuarios
- Account Service: Gestión de cuentas bancarias
- Payment Service: Procesamiento de pagos multitenant
- Risk Service: Evaluación de riesgos con ML
- Fraud Service: Detección de fraude en tiempo real
- Notification Service: Notificaciones multicanal

4 - Legacy Integration:

- Legacy Core: Monolito existente
- Patrón: Strangler Fig para migración gradual

Diagrama de Contexto

5 - Data Layer

• RDS PostgreSQL: Datos transaccionales críticos

• DynamoDB: Datos de sesiones y dinámicos

• S3: Documentos y logs

• ElastiCache: Cache y sesiones

6 - Messaging

• Amazon MSK: Event streaming con Kafka

• Patrón: Event-driven architecture

Tecnologías AWS Utilizadas:

Servicio	Propósito	Características
ECS Fargate	Orquestación de contenedores	Serverless, auto-scaling
API Gateway	Punto de entrada de APIs	Rate limiting, caching
Lambda	Funciones serverless	Notificaciones, Open Finance
RDS PostgreSQL	Base de datos relacional	Multi-AZ, read replicas
DynamoDB	Base de datos NoSQL	Global tables, auto-scaling
MSK	Event streaming	Kafka, real-time processing
SageMaker	Machine Learning	Modelos de riesgo y fraude