

E-Banking 3.0 (moderne, microservices, SOAP, GraphQL & Devops)

Objectif : moderniser l'application E-Banking existante (les anciens projets) en :

- **Améliorant l'UX** (SPA + GraphQL pour requêtes agrégées),
- **Découpant** la plateforme en microservices indépendants (scalabilité, tests isolés),
- **Conservant l'interopérabilité** avec systèmes bancaires legacy via **SOAP**,
- **Ajoutant** services innovants (portefeuille crypto, assistant IA, biométrie).

Modernisation (les anciens projets) :

Espaces Client / Agent / Admin, virements, recharges, gestion comptes, RGPD, journaux d'audit.

Nouveautés :

Portefeuille crypto (intégration API partenaires, historique, taux temps-réel).

Assistant bancaire IA (chat / FAQ — intégration GPT/Dialogflow).

Paiement biométrique & QR (authentification forte) Apple Pay.

Graphiques et gestion budgétaire (dashboards interactifs)

Architecture technique (haut niveau)

1. **Frontend SPA** (Angular + Apollo GraphQL) \longleftrightarrow **API Gateway GraphQL**
2. **API Gateway GraphQL** : agrège et unifie les données de plusieurs microservices (Account, Payments, Crypto, User, Notification, Analytics).
3. **Microservices** : REST internal + events via Kafka.
4. **Adapter SOAP** : un microservice “Legacy-Adapter” réalise les appels SOAP vers le cœur bancaire existant (en lecture/écriture) et normalise la réponse pour les microservices.
5. **Auth / IAM** : Keycloak (OAuth2 or OpenID Connect) pour SSO, MFA et gestion des rôles (client, agent, admin).
6. **Event Bus** : Kafka pour événements (transaction completed, notification, audit).
7. **Persistence** : PostgreSQL (comptes, transactions), MongoDB (logs & historique crypto), Redis (cache, sessions).
8. **Blockchain**.

9. **Observability** : Prometheus + Grafana, ELK (Elastic) pour logs, Jaeger pour traces distribuées.
10. **CI/CD & Infra** : Docker → Kubernetes (Helm) ; pipeline GitHub Actions or **Jenkins** pour build/test/deploy.

Liste détaillée des microservices

- **auth-service** : s'interface avec Keycloak, gère MFA, sessions.
- **user-service** : profils, KYC, consentements RGPD.
- **account-service** : comptes, soldes, relevés.
- **payment-service** : virements, virements instantanés, règles anti-fraude.
- **legacy-adaptor-service (SOAP)** : wrapper SOAP pour le core bancaire (expose endpoints REST internes).
- **crypto-service** : opérations crypto (achat/vente via partenaires).
- **notification-service** : e-mail, SMS, push.
- **analytics-service** : agrégation pour dashboard et alertes, expose graphiques.
- **ai-assistant-service** : webhook vers ChatGPT/Dialogflow + logs des conversations.
- **audit-service**

Sécurité & conformité

- **OAuth2 / OIDC** (Keycloak) + **MFA** (SMS + biométrie).
- **RGPD** : consentements, droit à l'effacement, journaux d'accès, anonymisation des logs.

DevOps & pipeline (CI/CD + Infra)

- **Repo** : un repo par microservice.
- **CI** : lint → build → unit tests → contract tests → sonarqube → image build (Docker) → push to registry.
- **CD** : image → staging deployment (K8s) → integration tests → manual/automated approval → production via Helm charts.
- **Infra as Code** : Terraform pour infra cloud (K8s cluster, managed DB), Helm pour apps.
- **Observability**: Prometheus (metrics), Grafana (dashboards), Elastic Stack (logs), Jaeger (traces).
- **Alerting** : Alertmanager + Slack/Email.

Tests & qualité

- **Unitaires + Integration** pour chaque service
- End-to-end tests