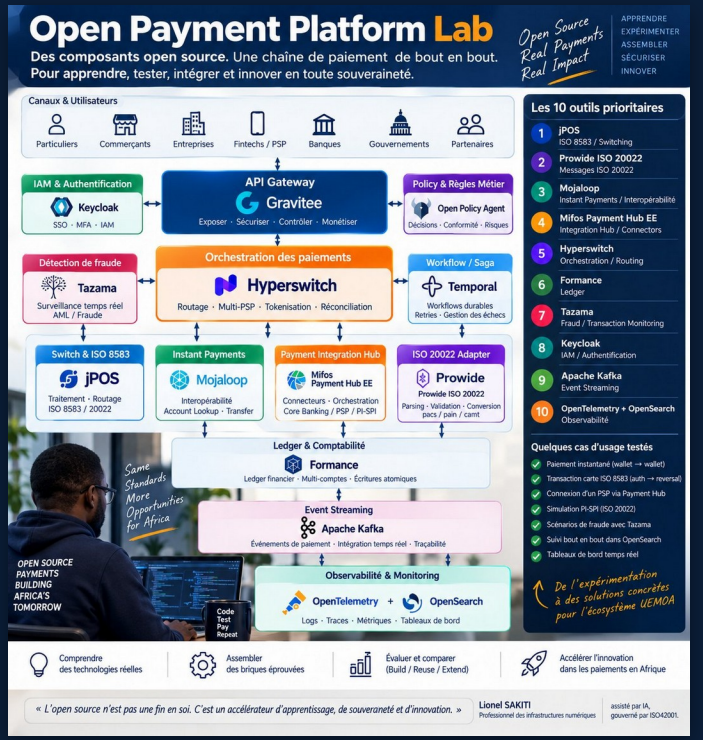
GUIDE DE MISE EN ŒUVRE · DÉPLOIEMENT · DÉVELOPPEMENT

Open Payment Platform Lab

Des composants open source. Une chaîne de paiement de bout en bout. Pour apprendre, tester, intégrer et innover en toute souveraineté.

Open Source · Real Payments · Real Impact

Lionel SAKITI
Professionnel des infrastructures numériques · Septembre 2026



Comprendre les paiements par la pratique

Dans un secteur réglementé, critique et exigeant, évaluer un composant sur sa documentation ne suffit pas. Ce qui compte, c'est son comportement une fois intégré : idempotence, reprise après panne, traçabilité, sécurité.

« L'objectif n'est pas de dire que l'open source peut tout remplacer. »

10 briques prioritaires intégrées

7 cas d'usage testés de bout en bout

11 phases de déploiement reproductibles

Ce que le Lab est — et ce qu'il n'est pas

Le Lab EST

- Un environnement d'apprentissage par la pratique
- Une architecture de référence intégrée, versionnée (GitOps)
- Un banc d'essai pour mesurer et comparer
- Un support de décision Build / Buy / Reuse / Extend / Replace
- Un outil de dialogue DSI – métiers – risques – régulateur

Le Lab N'EST PAS

- Une plateforme de production
- Une affirmation que l'open source remplace tout
- Un substitut aux homologations et certifications
- Un environnement traitant des données clients réelles
- Un projet figé : les briques évoluent

Huit verbes pour structurer la démarche

Installer

Déploiement reproductible

Intégrer

Contrats API & événements

Tester

Cas nominaux et d'erreur

Simuler

Émetteurs, acquéreurs, DFSP

Observer

Traces, métriques, logs

Casser

Chaos engineering

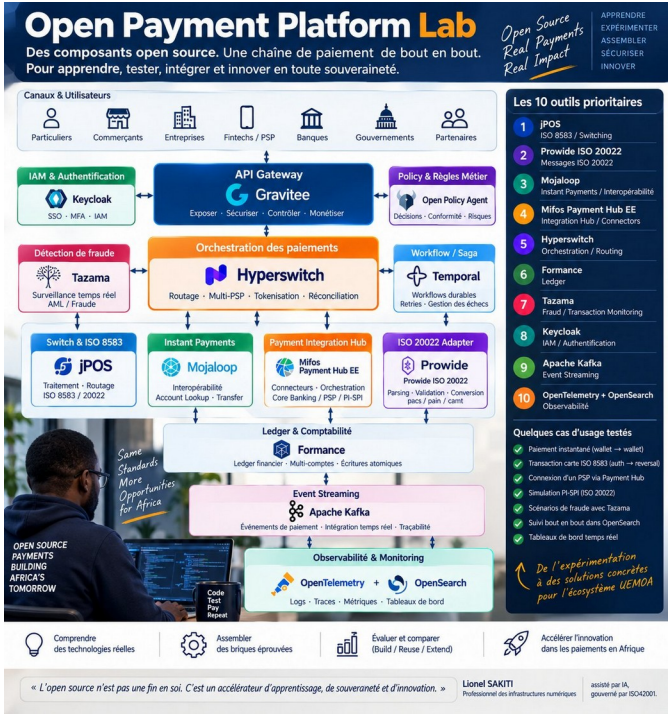
Sécuriser

Sécurité par défaut

Benchmarker

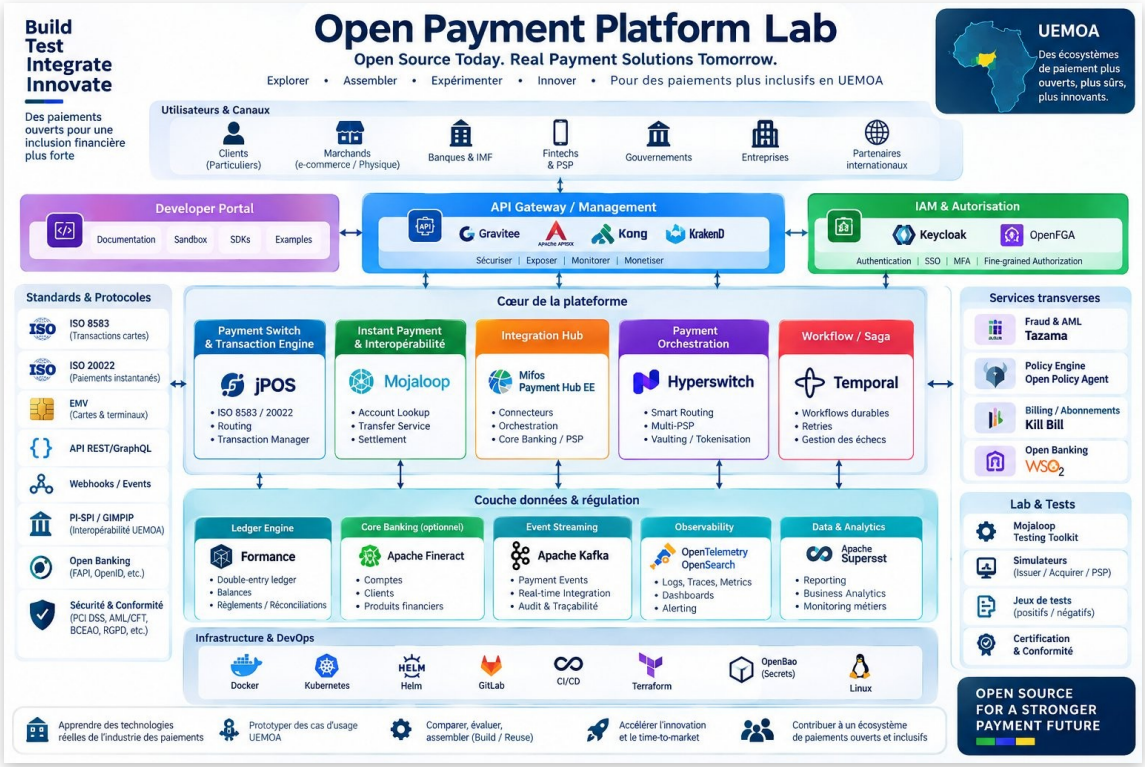
Débit, latence, coût

Architecture de référence



- **Exposition** : Gravitee, Keycloak, Open Policy Agent
- **Orchestration** : Hyperswitch + Temporal (sagas)
- **Rails** : jPOS (ISO 8583), Mojaloop, Payment Hub EE, Prowide ISO 20022
- **Ledger** : Formance, source de vérité financière
- **Événements** : Apache Kafka
- **Observabilité & risque** : OpenTelemetry, OpenSearch, Tazama

Vue étendue de la plateforme



Sept couches, une chaîne de bout en bout



Applications de démonstration, simulateurs

Gravitee · Keycloak · OPA

Hyperswitch · Temporal

jPOS · Mojaloop · Payment Hub EE · Prowide

Formance

Apache Kafka

OpenTelemetry · OpenSearch · Tazama

Les 10 briques prioritaires

1

jPOS

ISO 8583 / Switching

2

Prowide ISO 20022

Messages ISO 20022

3

Mojaloop

Instantané / Interopérabilité

4

Mifos Payment Hub EE

Intégration / Connecteurs

5

Hyperswitch

Orchestration / Routage

6

Formance

Ledger

7

Tazama

Fraude / Monitoring

8

Keycloak

IAM / Authentification

9

Apache Kafka

Event streaming

10

OpenTelemetry + OpenSearch

Observabilité

I

Les briques

Dix fiches : rôle, licence, place dans le Lab, points d'attention.

jPOS — Switch ISO 8583

1

AGPL v3 / commerciale · Java · Q2

- Autorisations 0100/0110, annulations 0400/0410
- TransactionManager : chaîne de participants (champs, politique, fraude, ledger)
- **Attention** : AGPL — analyse juridique ou licence commerciale avant production

jPOS Q2 — switch ISO 8583 (log)

```
<log realm="opp-switch/10.42.3.18:52114" at="2026-09-08T10:51:03.114">
  <receive>
    <isomsg direction="incoming">
      <field id="0"  value="0100"/>              <!-- demande d'autorisation -->
      <field id="2"  value="4761*******0019"/>  <!-- PAN masqué (test) -->
      <field id="3"  value="000000"/>
      <field id="4"  value="000000015000"/>
      <field id="11" value="000417"/>
      <field id="41" value="TERM0001"/>
      <field id="49" value="952"/>                <!-- XOF -->
    </isomsg>
  </receive>
</log>
<log realm="txnmgr" at="2026-09-08T10:51:03.162" lifespan="48ms">
  PrepareContext [PREPARED] -> CheckFields [PREPARED] -> OpaPolicyCheck [PREPARED]
  -> FraudScore(Tazama) [PREPARED] -> LedgerHold(Formance) [PREPARED] -> SendResponse [COMMITTED]
</log>
<log realm="opp-switch/10.42.3.18:52114" at="2026-09-08T10:51:03.165">
  <send> <field id="0" value="0110"/> <field id="11" value="000417"/> <field id="38" value="A1B2C3"/> <field id="39" value="00"/> </send>
</log>
<log realm="opp-switch" at="2026-09-08T10:51:44.020"> 0400 reversal STAN 000417 -> LedgerRelease [COMMITTED] -> 0410 / 39=00 </log>
```

Maquette illustrative · OPP Lab

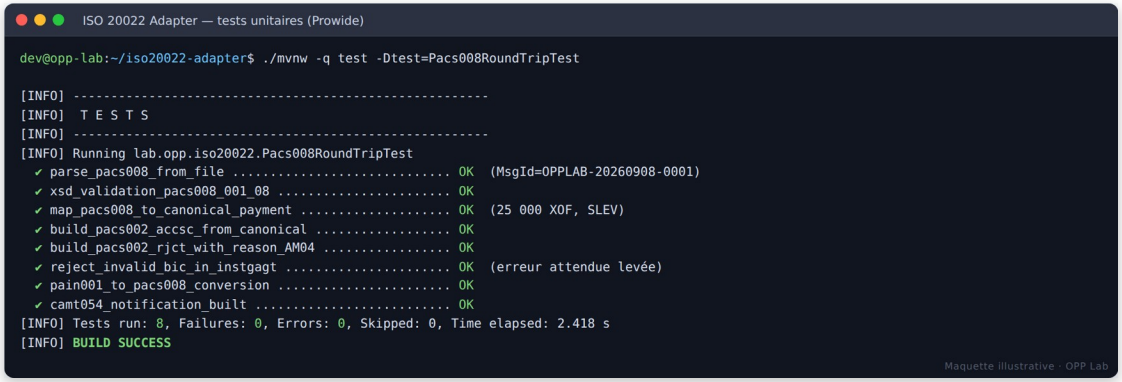
Capture illustrative — maquette reconstituée, chiffres fictifs

Prowide ISO 20022 — Messages

2

Apache 2.0 · Java · Maven

- Parser, construire, convertir pacs / pain / camt
- Fondation de l'adaptateur ISO 20022 du Lab
- **Attention** : validation des règles de marché à construire ou acquérir



Capture illustrative — maquette reconstituée, chiffres fictifs

Mojaloop — Instantané & interopérabilité

3

Apache 2.0 · Node.js · Helm

- Lookup d'alias, devis, transferts, règlement
- Testing Toolkit : DFSP simulés et collections de tests
- **Attention** : brique la plus lourde ; support pédagogique, pas une réplique de PI-SPI



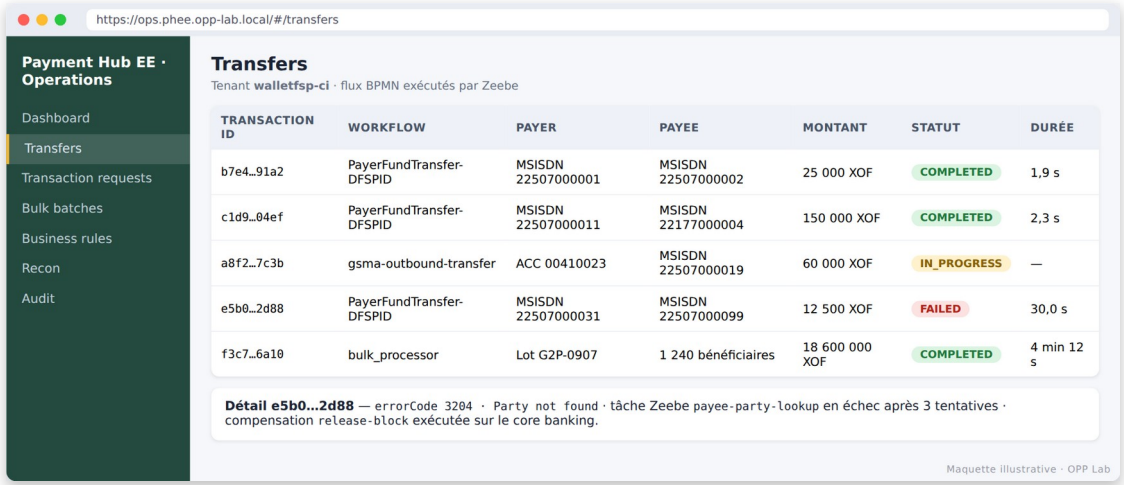
Capture illustrative — maquette reconstituée, chiffres fictifs

Mifos Payment Hub EE — Intégration

4

MPL 2.0 · Java · Zeebe (BPMN)

- Côté participant : connecteurs core banking, Mojaloop, GSMA
- Paiements de masse (G2P, salaires)
- **Attention** : Zeebe ≥ 8.6 soumis à licence pour la production



Capture illustrative — maquette reconstituée, chiffres fictifs

Hyperswitch — Orchestration & routage

5

Apache 2.0 · Rust

- Routage multi-PSP par règles, bascule automatique
- Tokenisation, réconciliation, analyse des taux d'autorisation
- **Opportunité** : contribuer des connecteurs pour les rails africains



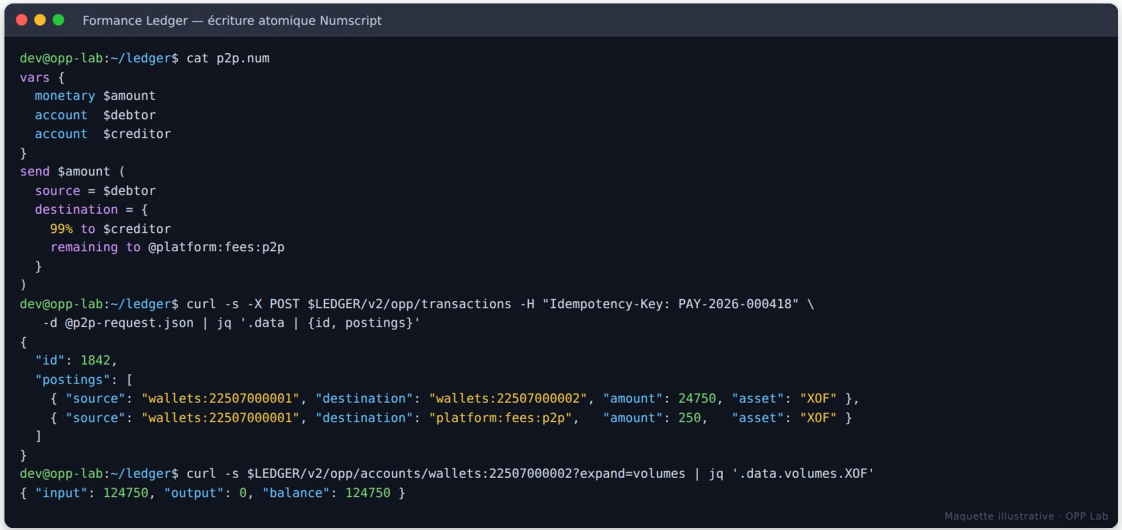
Capture illustrative — maquette reconstituée, chiffres fictifs

Formance — Ledger financier

6

MIT · Go · PostgreSQL

- Écritures atomiques à double entrée, Numscript lisible par les métiers
- Idempotence native par clé
- **Attention** : définir le plan de comptes avant de coder



Capture illustrative — maquette reconstituée, chiffres fictifs

Tazama — Fraude & monitoring

7

Apache 2.0 · Linux Foundation · NATS

- Règles → typologies → décision (TADProc) en temps réel
- Messages ISO 20022 : pacs.008, pacs.002, pain.001, pain.013
- **Attention** : pile de référence non destinée à la production

Tazama — évaluation temps réel d'un pacs.008

```
fraud@opp-lab:~/tazama$ curl -s -X POST http://tazama-tms-api.opp-fraud:5000/v1/evaluate/iso20022/pacs.008.001.10 \
    -H "Content-Type: application/json" -d @samples/pacs008-velocity-burst-07.json | jq '.message'
"Transaction is valid"

fraud@opp-lab:~/tazama$ nats sub 'interdiction-service' --count 1 | jq '.report | {status, typologyResult}'
{
  "status": "ALRT",
  "typologyResult": [
    { "id": "typology-processor@1.0.0", "cfg": "028@1.0.0",
      "result": 520, "workflow": { "alertThreshold": 400, "interdictionThreshold": 500 },
      "ruleResults": [
        { "id": "rule-018", "subRuleRef": ".03", "reason": "Montant anormalement élevé vs historique" },
        { "id": "rule-044", "subRuleRef": ".02", "reason": "Nouveau bénéficiaire + rafale de transactions" },
        { "id": "rule-078", "subRuleRef": ".01", "reason": "Compte récemment créé" }
      ] } ],
  "tadpResult": { "prcgTm": 41 }
}
# Relais NATS -> Kafka : alerte publiée sur opp.fraud.alerts.v1 ; le workflow Temporal compense.
```

Maquette illustrative · OPP Lab

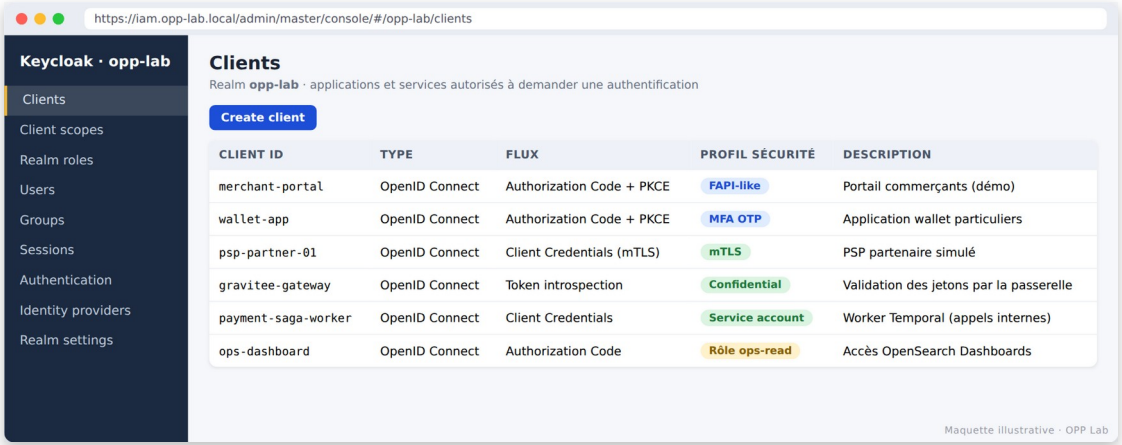
Capture illustrative — maquette reconstituée, chiffres fictifs

Keycloak — IAM & authentification

8

Apache 2.0 · CNCF · Quarkus

- OIDC / OAuth2, MFA, comptes de service, mTLS partenaires
- Realm opp-lab géré en code (Git)
- **Règle** : séparer realm master et realm applicatif



Capture illustrative — maquette reconstituée, chiffres fictifs

Apache Kafka — Événements de paiement

9

Apache 2.0 · KRaft · Strimzi

- Événements CloudEvents, clé de partition = paymentId
- Piste d'audit technique, rejouabilité
- **Choix** : at-least-once + consommateurs idempotents

Kafka — topics et événements de paiement

```
ops@opp-lab:~$ kubectl -n opp-streaming exec opp-kafka-broker-0 -- bin/kafka-topics.sh --bootstrap-server localhost:9092 --list
opp.payments.initiated.v1
opp.payments.authorized.v1
opp.payments.settled.v1
opp.payments.failed.v1
opp.ledger.postings.v1
opp.fraud.alerts.v1
opp.iso8583.messages.v1
opp.iso20022.messages.v1
opp.audit.trail.v1
ops@opp-lab:~$ kcat -b opp-kafka-bootstrap:9092 -t opp.payments.settled.v1 -C -o -1 -e | jq
{
  "specversion": "1.0",
  "type": "opp.payment.settled",
  "source": "/opp/workflow/payment-saga",
  "id": "evt-01J9Z6Q4X3K8M2N5P7R1S0T9V",
  "time": "2026-09-08T10:42:17.381Z",
  "traceparent": "00-4bf92f3577b34da6a3ce929d0e0e4736-00f067aa0ba902b7-01",
  "data": {
    "paymentId": "PAY-2026-000418", "rail": "MOJALOOP",
    "amount": { "value": 25000, "currency": "XOF" },
    "debtor": "wallet:+22507000001", "creditor": "wallet:+22507000002",
    "ledgerTxId": 1842, "fraudScore": 0.12, "status": "SETTLED"
  }
}
```

Maquette illustrative · OPP Lab

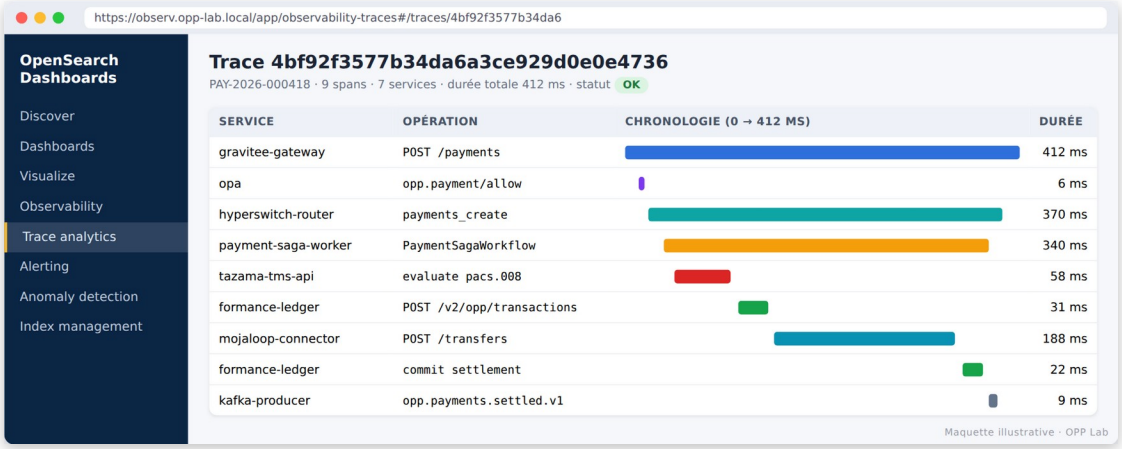
Capture illustrative — maquette reconstituée, chiffres fictifs

OpenTelemetry + OpenSearch — Observabilité

10

Apache 2.0 · OTLP · Dashboards

- Une trace unique du canal jusqu'au ledger
- Masquage des données sensibles dans le Collector
- **Condition** : on ne teste ni ne casse ce qu'on n'observe pas



Capture illustrative — maquette reconstituée, chiffres fictifs

II

Déploiement

Trois niveaux de Lab, onze phases, tout en code.

Trois niveaux de Lab

1 Poste développeur

Docker / Podman Compose

- 1 personne, ≥ 32 Go RAM
- 3 à 5 briques par cas d'usage
- Découverte, formation

2 Lab d'équipe

Kubernetes léger (k3s / RKE2)

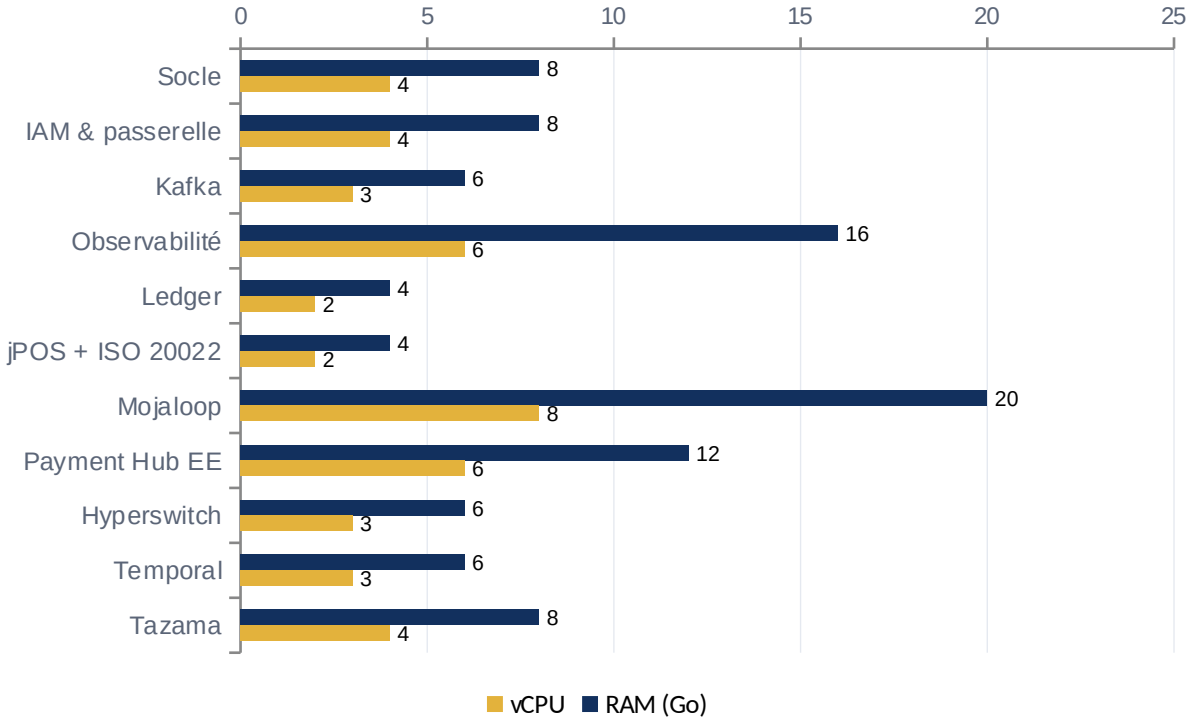
- 5 à 20 personnes, 3 nœuds
- Les 10 briques + socle
- Intégration bout en bout

3 Lab de performance

Kubernetes multi-nœuds HA

- Équipe plateforme
- Configuration haute dispo
- Benchmark, chaos

Dimensionnement indicatif — Lab de niveau 2



≈ 45

vCPU (requests)

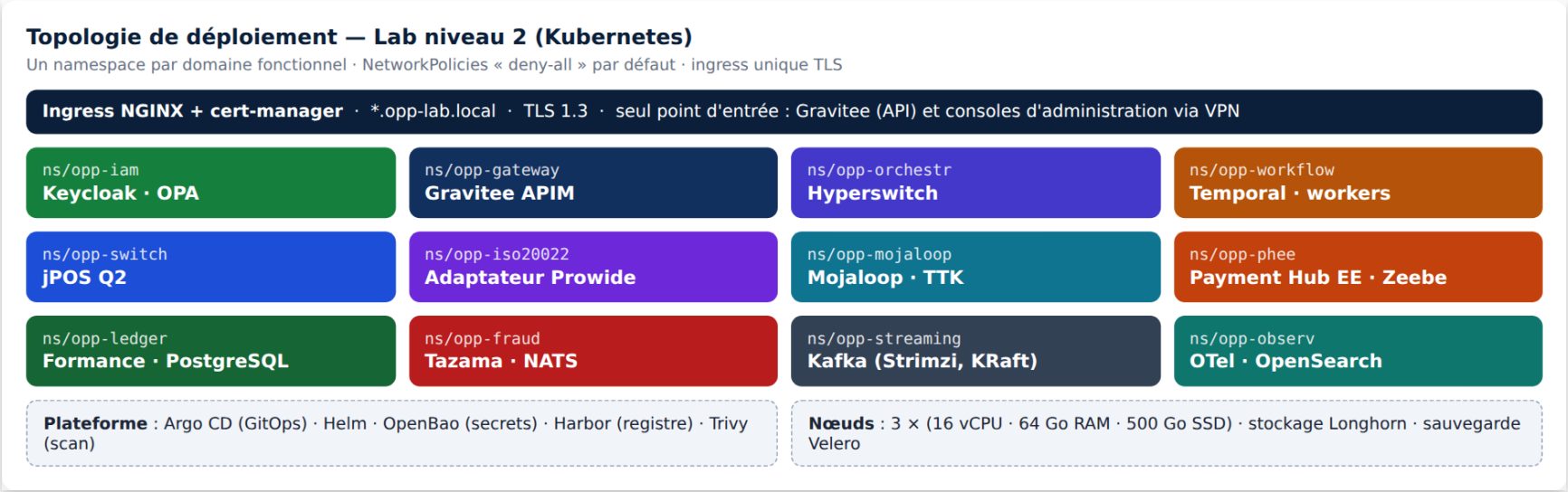
≈ 98 Go

RAM

≈ 690 Go

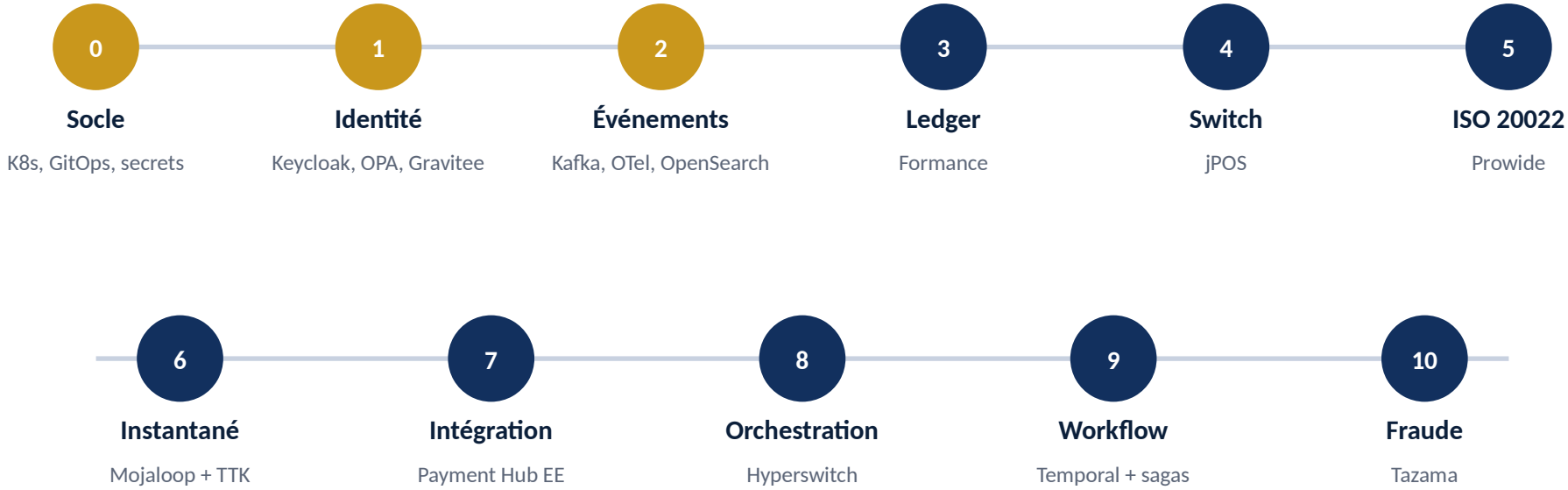
Stockage

Un namespace par domaine, deny-all par défaut



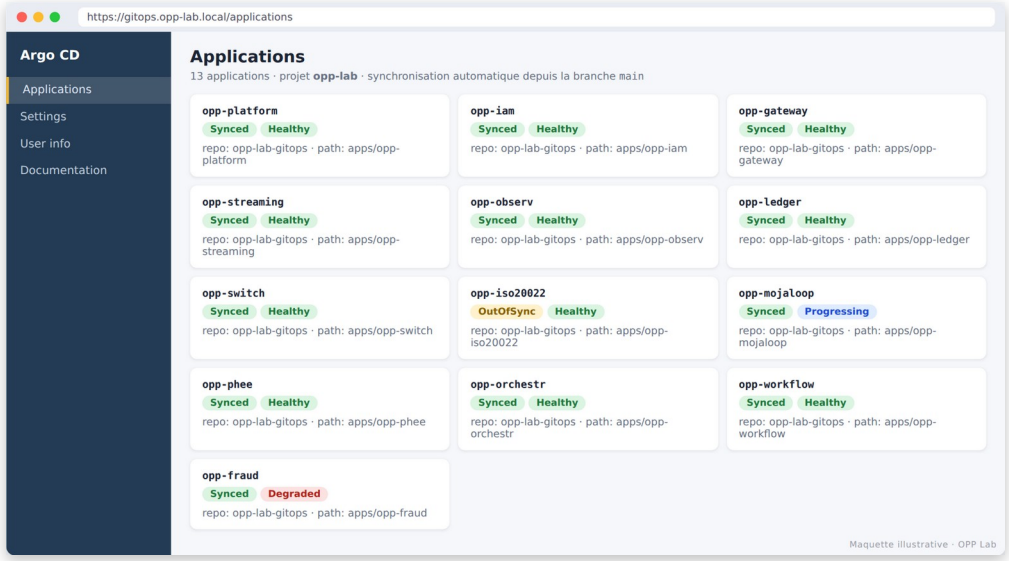
Onze phases de déploiement

Durée indicative : ≈ 23 jours-ingénieur · chaque phase a des prérequis, une procédure, un critère de sortie et une preuve archivée



Tout en code : GitOps, secrets et chaîne logicielle

- **Argo CD** synchronise le dépôt opp-lab-gitops (app-of-apps)
- **OpenBao + External Secrets** : aucun secret dans Git
- **cert-manager** : autorité de certification interne, TLS partout
- **Harbor + Trivy + cosign** : images scannées et signées
- Reconstruction complète du Lab : **< 1 journée**



Capture illustrative — maquette reconstituée, chiffres fictifs

Points d’accès du Lab

Service	URL du Lab	Port interne	Accès
Gravitee (passerelle)	api.opp-lab.local	8082	Public Lab
Keycloak	iam.opp-lab.local	8080	OIDC public / admin VPN
OPA	interne	8181	Namespaces autorisés
Hyperswitch / Control Center	pay.opp-lab.local	8080 / 9000	Interne / VPN
Temporal frontend / UI	workflow.opp-lab.local	7233 / 8080	Interne / VPN
jPOS ISO 8583	switch.opp-lab.local	10000 (config.)	Simulateurs
Tazama TMS API	interne	5000	Worker de saga
OpenSearch / Dashboards	observ.opp-lab.local	9200 / 5601	VPN, rôle ops
OTel Collector	interne	4317 / 4318	Tous namespaces

III

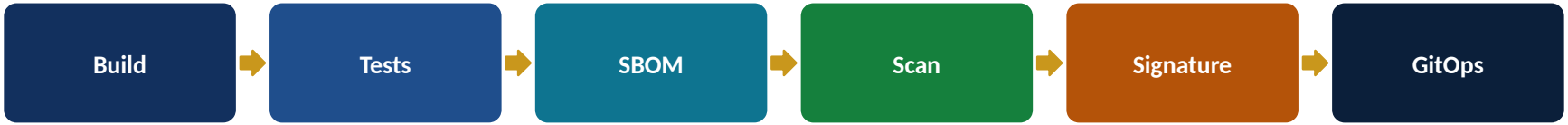
Développement

Le code d’intégration : là où se trouve la vraie valeur du Lab.

Monorepo applicatif et chaîne de livraison

```
opp-lab-services/
├── contracts/  (OpenAPI, AsyncAPI, schémas)
├── canonical-model/
├── payment-saga-worker/     # Temporal
├── iso20022-adapter/        # Prowide
├── switch-participants/     # jPOS
├── phee-workers/            # Zeebe
├── hyperswitch-connectors/
├── policies/                # Rego + tests
├── simulators/  data-generator/  perf/
```

- Tests unitaires, de contrat et d'intégration à chaque merge request
- SBOM CycloneDX + inventaire des licences automatique
- Seules les images signées sont admises dans le cluster



Un modèle canonique pour dix langages

```
record CanonicalPayment(
  String paymentId,
  String idempotencyKey,
  Rail rail,     // CARD|INSTANT|ISO20022|PSP
  Money amount,  // unités mineures + ISO 4217
  Party debtor, Party creditor,
  String purpose, Instant createdAt,
  Map<String,String> metadata)
```

Canonique	ISO 8583	ISO 20022	Mojaloop
paymentId	STAN + RRN	EndToEndId	transactionId
montant	champ 4	IntrBkSttlmAmt	amount
devise	champ 49 (952)	@Ccy	currency
statut	champ 39	pacs.002 TxSts	transferState

Montants en entiers, jamais de flottant · XOF : 0 décimale

Contrat d'événements : CloudEvents sur Kafka

Topic	Producteur	Consommateurs
opp.payments.initiated.v1	Saga	Analytique, fraude, audit
opp.payments.settled.v1	Saga	Réconciliation, notifications
opp.payments.failed.v1	Saga	Support, alerting
opp.ledger.postings.v1	Saga	Réconciliation, reporting
opp.fraud.alerts.v1	Relais Tazama	Saga, case management
opp.iso8583.messages.v1	jPOS (masqué)	Audit, rejeu
opp.audit.trail.v1	Tous	Archivage OpenSearch

Enveloppe

CloudEvents 1.0 + traceparent

Garantie

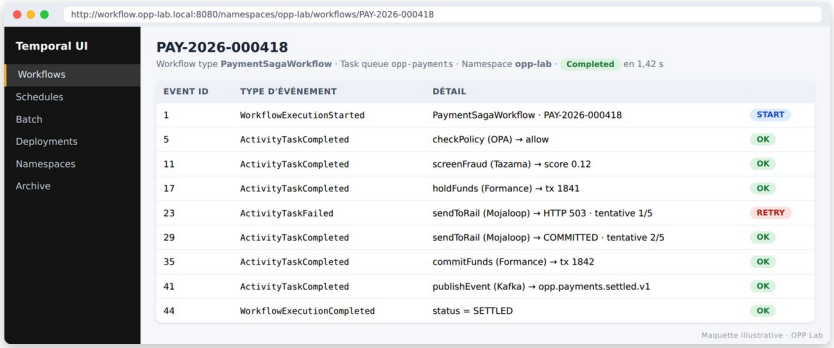
At-least-once, consommateurs idempotents

Résilience

3 réplicas, min.insync = 2

La saga de paiement avec Temporal

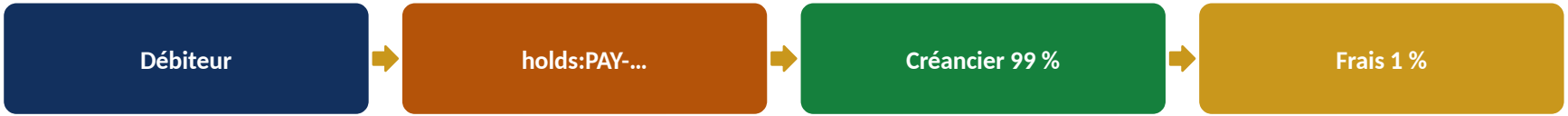
```
Saga saga = new Saga(opts);
if (!act.checkPolicy(p).allow())
    return fail("POLICY_DENIED");
if (act.screenFraud(p).blocked())
    return fail("FRAUD_BLOCKED");
String holdId = act.holdFunds(p);
saga.addCompensation(act::releaseFunds, holdId);
RailResult r = act.sendToRail(p); // retries
if (!r.accepted()) { saga.compensate(); ... }
long tx = act.commitFunds(holdId, p);
act.publishEvent("settled", p, tx);
```



Capture illustrative — maquette reconstituée, chiffres fictifs

Compensation garantie, même après redémarrage du worker

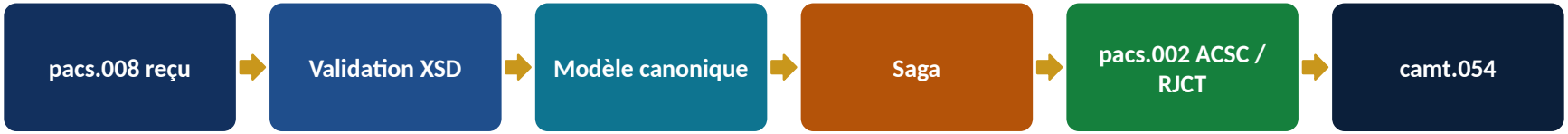
Le ledger : réserver, régler, compenser




```
// commit.num — règlement avec frais
send $amount (
  source = $hold
  destination = {
    99% to $creditor
    remaining to @platform:fees:p2p
  }
)
```

Compte	Usage
wallets:<msisdn>	Wallets simulés
holds:<paymentId>	Fonds réservés
settlement:<rail>	Positions par rail
platform:fees:<produit>	Frais perçus
platform:suspense	Écarts à analyser

Adaptateur ISO 20022 : du pacs.008 au pacs.002




```
MxPacs00800108 mx = MxPacs00800108.parse(xml);
CreditTransferTransaction39 tx =
    mx.getFIToFICstmrCdtTrf().getCdtTrfTxInf().get(0);
Money amount = new Money(
    tx.getIntrBkSttlmAmt().getValue().longValueExact(),
    tx.getIntrBkSttlmAmt().getCcy());
PaymentResult r = sagaClient.startAndWait(canonical);
return Pacs002Builder.fromResult(mx, r);
```

- Rejets testés : devise invalide, montant décimal, BIC inconnu, doublon de MsgId
- Versions de messages épinglées (.001.08, .001.10…)
- Règles de marché : cas d’école Build / Buy

Politiques métier en code : Open Policy Agent

OPA — décision de politique paiement

```
dev@opp-lab:~/policies$ cat payment/limits.rego
package opp.payment

import rego.v1

default allow := false

allow if {
    input.channel in {"wallet", "merchant", "psp"}
    input.amount.value <= data.limits[input.customer.kyc_level].per_tx
    not sanctioned
}

sanctioned if input.creditor.country in data.sanctions.countries

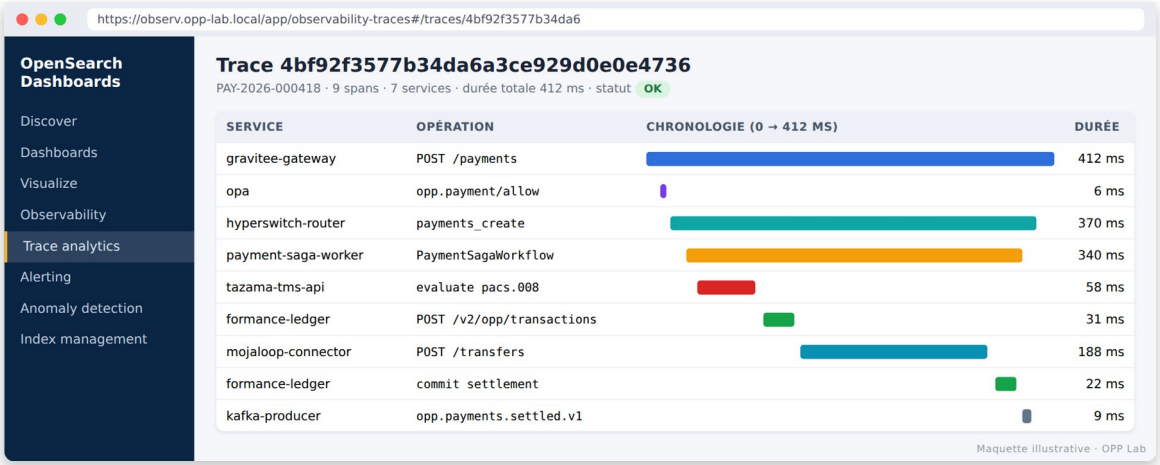
dev@opp-lab:~/policies$ curl -s -X POST http://opa.opp-iam:8181/v1/data/opp/payment \
    -d '{"input":{"channel":"wallet","amount":{"value":750000,"currency":"XOF"},
         "customer":{"kyc_level":"L1"},"creditor":{"country":"CI"}}}' | jq
{
  "result": {
    "allow": false,
    "reasons": [ "amount 750000 XOF exceeds per_tx limit 500000 for kyc_level L1" ],
    "sanctioned": false
  },
  "decision_id": "3f1c9a2e-8b4d-4c61-9e0f-5a7d2b6c8e11"
}
```

Maquette illustrative · OPP Lab

Capture illustrative — maquette reconstituée, chiffres fictifs

- **Logique** en Rego, **données** en JSON (plafonds KYC, listes, horaires)
- Décision motivée : raison lisible par le support
- Politiques testées par « opa test » dans la CI
- Appelée par Gravitee, la saga et jPOS

Instrumenter pour voir chaque transaction



Capture illustrative — maquette reconstituée, chiffres fictifs

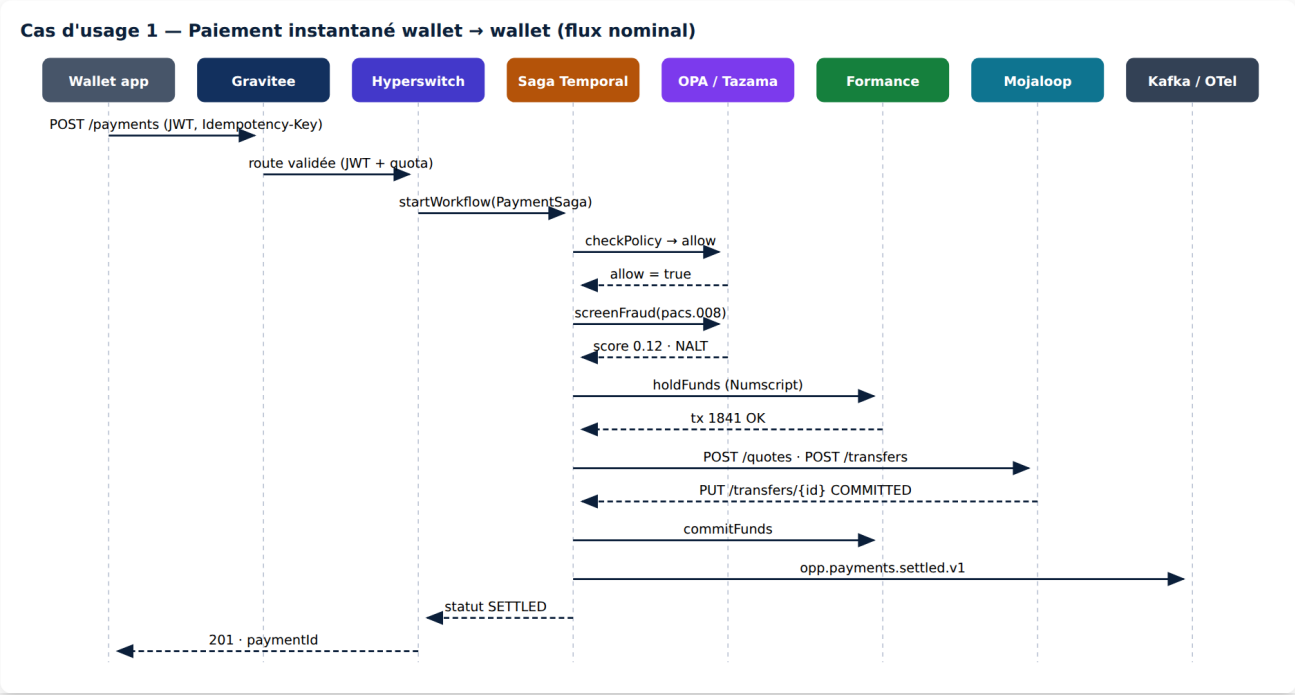
- Agent Java : zéro code pour HTTP, JDBC, Kafka
- Spans métier : opp.payment_id, opp.rail, montant
- traceparent propagé dans Kafka
- Masquage dans le Collector

IV

Cas d’usage

Sept scénarios automatisés, observés et documentés.

Cas 1 — Paiement instantané de wallet à wallet



Attendu : SETTLED en moins de 2 s · hold puis commit équilibrés · événement publié · trace complète

Cas 2 — Carte ISO 8583 : autorisation puis annulation



jPOS Q2 — switch ISO 8583 (log)

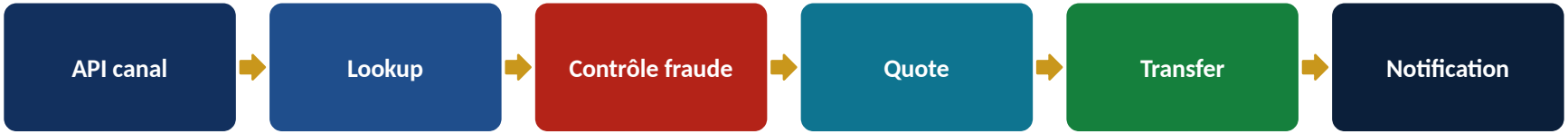
```
<log realm="opp-switch/10.42.3.18:52114" at="2026-09-08T10:51:03.114">
  <receive>
    <isomsg direction="incoming">
      <field id="0"  value="0100"/>              <!-- demande d'autorisation -->
      <field id="2"  value="4761********0019"/>  <!-- PAN masqué (test) -->
      <field id="3"  value="000000"/>
      <field id="4"  value="000000015000"/>
      <field id="11" value="000417"/>
      <field id="41" value="TERM0001"/>
      <field id="49" value="952"/>                <!-- XOF -->
    </isomsg>
  </receive>
</log>
<log realm="txnmgr" at="2026-09-08T10:51:03.162" lifespan="48ms">
  PrepareContext [PREPARED] -> CheckFields [PREPARED] -> OpaPolicyCheck [PREPARED]
  -> FraudScore(Tazama) [PREPARED] -> LedgerHold(Formance) [PREPARED] -> SendResponse [COMMITTED]
</log>
<log realm="opp-switch/10.42.3.18:52114" at="2026-09-08T10:51:03.165">
  <send> <field id="0" value="0110"/> <field id="11" value="000417"/> <field id="38" value="A1B2C3"/> <field id="39" value="00"/> </send>
</log>
<log realm="opp-switch" at="2026-09-08T10:51:44.020"> 0400 reversal STAN 000417 -> LedgerRelease [COMMITTED] -> 0410 / 39=00 </log>
```

Maquette illustrative · OPP Lab

Capture illustrative — maquette reconstituée, chiffres fictifs

- Chaîne : champs → OPA → Tazama → hold Formance
- Compensation locale dans abort()
- Messages masqués publiés sur Kafka

Cas 3 — Connexion d'un PSP via Payment Hub EE



Nominal

Transfert COMPLETED visible dans Operations

Échec

MSISDN inconnu → compensation release-block côté core banking

Masse

Lot G2P de 1 000 bénéficiaires : durée mesurée

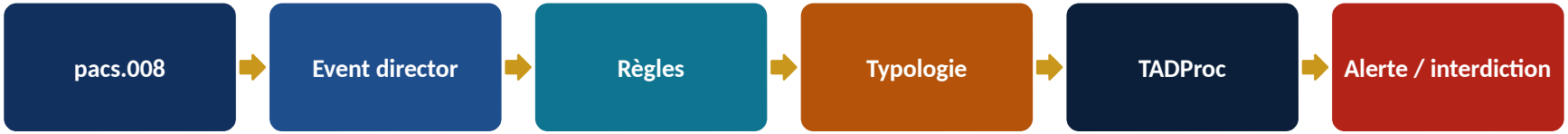
Cas 4 — Simulation PI-SPI (ISO 20022)



Portée de la simulation Le Lab simule les principes d'un paiement instantané interopérable fondé sur ISO 20022 afin de préparer les équipes. Toute connexion effective relève exclusivement du processus d'homologation de la BCEAO.

- Rejets motivés testés : devise, montant décimal en XOF, BIC inconnu, doublon
- Transactions irrévocables après validation

Cas 5 — Scénarios de fraude avec Tazama



Tazama — évaluation temps réel d'un pacs.008

```
fraud@opp-lab:~/tazama$ curl -s -X POST http://tazama-tms-api.opp-fraud:5000/v1/evaluate/iso20022/pacs.008.001.10 \
    -H "Content-Type: application/json" -d @samples/pacs008-velocity-burst-07.json | jq '.message'
"Transaction is valid"

fraud@opp-lab:~/tazama$ nats sub 'interdiction-service' --count 1 | jq '.report | {status, typologyResult}'
{
  "status": "ALRT",
  "typologyResult": [
    { "id": "typology-processor@1.0.0", "cfg": "028@1.0.0",
      "result": 520, "workflow": { "alertThreshold": 400, "interdictionThreshold": 500 },
      "ruleResults": [
        { "id": "rule-018", "subRuleRef": ".03", "reason": "Montant anormalement élevé vs historique" },
        { "id": "rule-044", "subRuleRef": ".02", "reason": "Nouveau bénéficiaire + rafale de transactions" },
        { "id": "rule-078", "subRuleRef": ".01", "reason": "Compte récemment créé" }
      ] } ],
  "tadpResult": { "prcgTm": 41 }
}
# Relais NATS -> Kafka : alerte publiée sur opp.fraud.alerts.v1 ; le workflow Temporal compense.
```

Maquette illustrative · OPP Lab

Capture illustrative — maquette reconstituée, chiffres fictifs

Asynchrone

Petits montants : détection sans latence client

Synchrone

Montants élevés : interdiction en ligne, latence à mesurer

Cas 6 & 7 — Suivi bout en bout et tableaux de bord



Capture illustrative — maquette reconstituée, chiffres fictifs

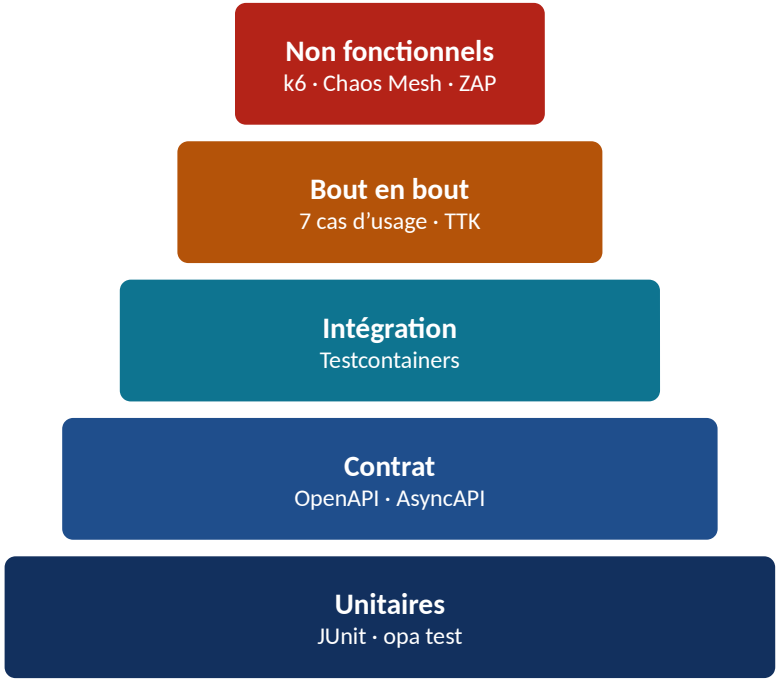
- Recherche par paymentId → trace → journaux → événement
- Volumes, succès, latence p95, rail, causes d'échec
- Réconciliation des volumes affichés avec Formance

V

Tester, sécuriser, décider

Tests, chaos, benchmark, conformité et grille de décision.

Tester et « casser » méthodiquement



Expérience de chaos	Critère de succès
Suppression du worker de saga	0 doublon, 0 perte
Latence 800 ms vers Mojaloop	Échecs < 2 %
Arrêt d'un broker Kafka	Aucune perte
Ledger en lecture seule	Aucun paiement à moitié réglé
Certificats proches de l'expiration	Alerte ≥ 7 jours avant

Benchmarker pour comparer, pas pour battre des records

Benchmark — k6 sur /payments (profil « instantané »)

```
perf@opp-lab:~/bench$ k6 run --vus 200 --duration 10m scenarios/instant-p2p.js

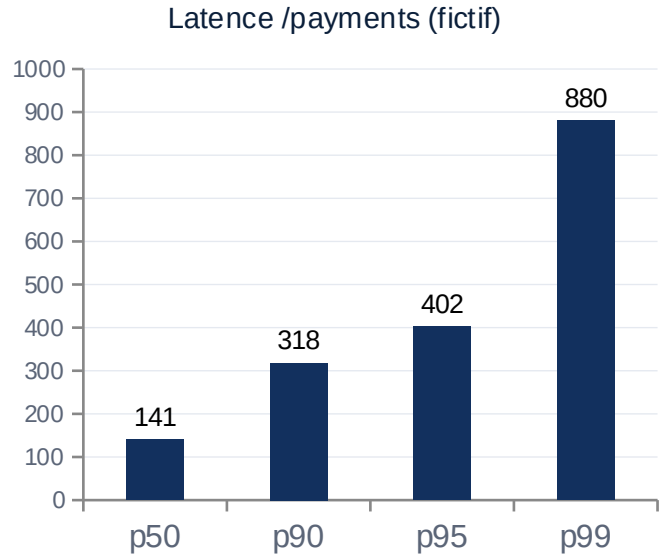
  scenarios: (100.00%) 1 scenario, 200 max VUs, 10m30s max duration

     ✓ status is 201 ........................... 99.41%  ✓ 351 902  ✗ 2 088
     ✓ has paymentId ........................... 100.00%
     ✓ ledger balanced (post-check) ............ 100.00%

     http_req_duration..............: avg=182ms  med=141ms  p(90)=318ms  p(95)=402ms  p(99)=880ms
     http_reqs......................: 353990  589.9/s
     iteration_duration.............: avg=339ms  min=61ms   max=5.02s
     ✓ thresholds: http_req_duration p(95)<500 ... PASSED
     ✗ thresholds: http_req_failed rate<0.5% ..... FAILED (0.59%)
```

Maquette illustrative · OPP Lab

Capture illustrative — maquette reconstituée, chiffres fictifs



Chiffres illustratifs : ils dépendent du matériel, des versions et de la configuration.

Sécurité par défaut et conformité

Identité

OIDC, MFA, moindre privilège

Réseau

Deny-all, TLS / mTLS, VPN admin

Données

Synthétiques uniquement, masquage

Chaîne logicielle

SBOM, scan, signature

PCI DSS v4.0.1

Concevoir la réduction de périmètre

UEMOA / BCEAO

Règl. 15/2002, PI-SPI, LBC/FT

ISO 27001

Management de la sécurité

ISO/IEC 42001

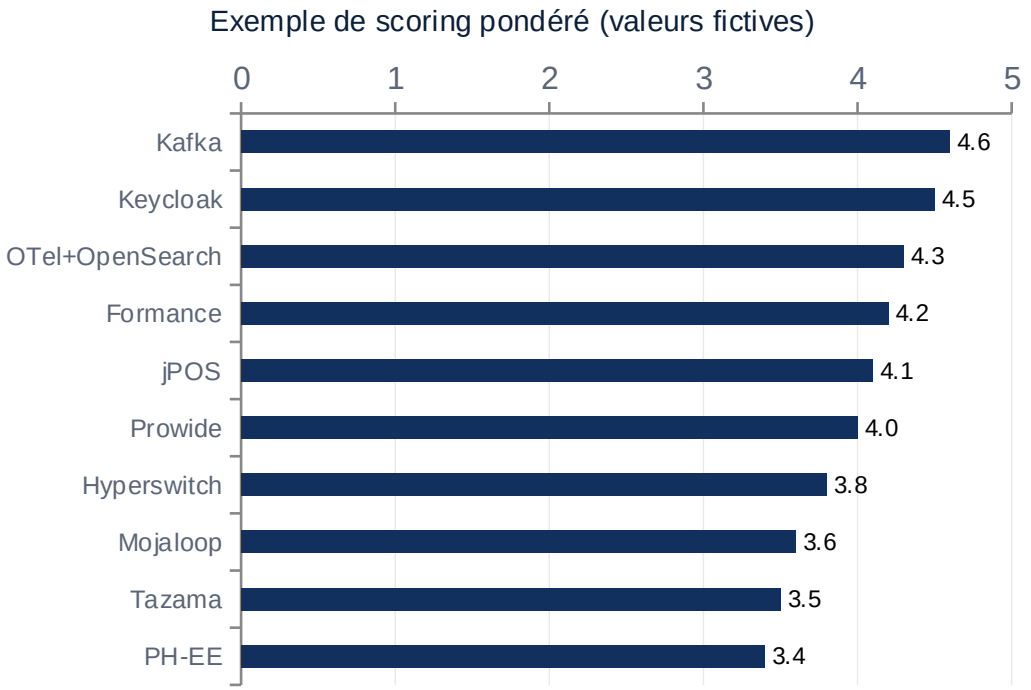
Gouvernance des usages d'IA

Licences : lire avant d'aller plus loin

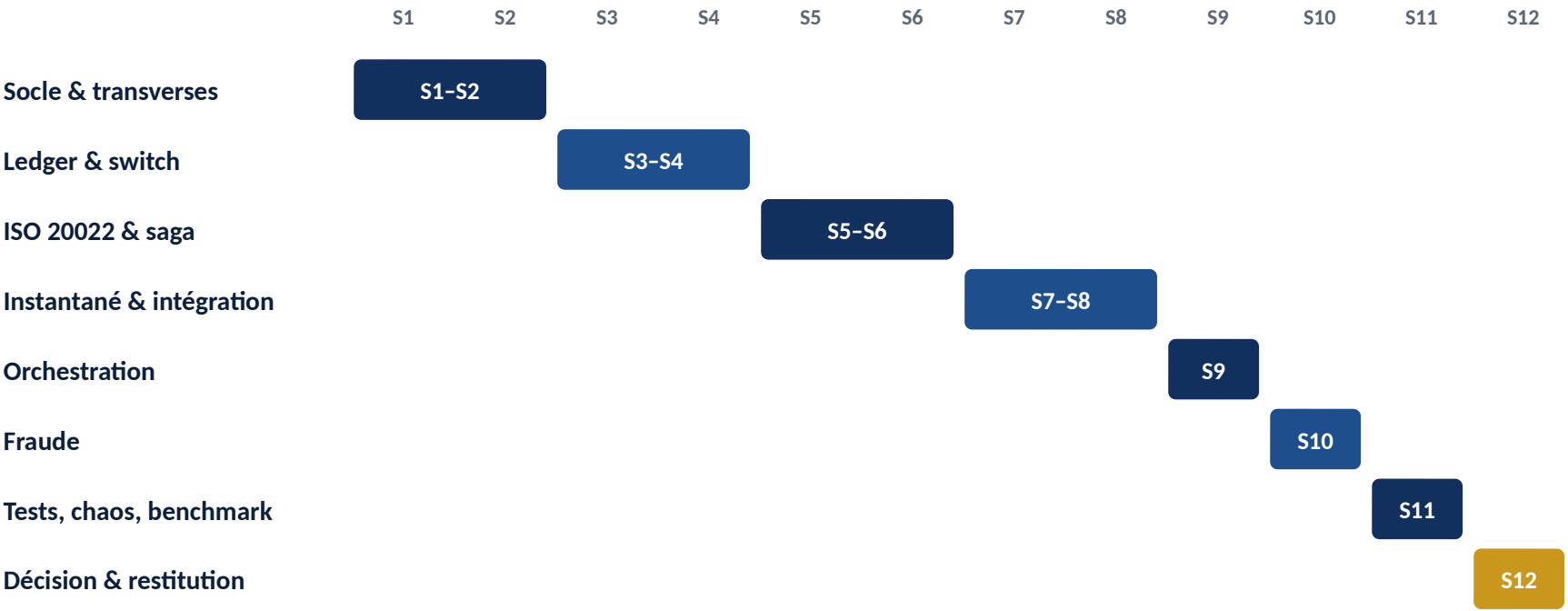
Composant	Licence	Vigilance	Point d'attention
jPOS	AGPL v3 / commerciale	**Élevée**	Diffusion du code modifié si service réseau
Payment Hub EE	MPL 2.0 + Zeebe	**Élevée**	Zeebe ≥ 8.6 : licence requise en production
Prowide ISO 20022	Apache 2.0	**Moyenne**	Fonctions avancées en offre commerciale
Gravitee APIM	Apache 2.0 (CE)	**Moyenne**	Certaines fonctions en édition entreprise
Tazama	Apache 2.0	**Moyenne**	Pile de référence non productive
Hyperswitch, Mojaloop, Keycloak, Kafka, OPA, OTel, OpenSearch	Apache 2.0	**Faible**	Attribution
Formance Ledger, Temporal	MIT	**Faible**	Attribution

Décider composant par composant

- **B** **Build** — Différenciation
- **B** **Buy** — Certification, support
- **R** **Reuse** — Brique mature
- **E** **Extend** — Socle + manques ciblés
- **R** **Replace** — Obsolescence, lock-in



Feuille de route sur douze semaines



Livrable final : grille de décision, ADR, démonstration à la direction

Same Standards · More Opportunities for Africa

« L'open source n'est pas une fin en soi. C'est un accélérateur d'apprentissage, de souveraineté et d'innovation. »



Comprendre
des technologies réelles



Assembler
des briques éprouvées



Évaluer
Build / Reuse / Extend



Accélérer
l'innovation en UEMOA

Lionel SAKITI · Professionnel des infrastructures numériques · assisté par IA, gouverné par ISO/IEC 42001