

# DevOps/SRE Stack

## DevOps/SRE Stack for Drop (originally FontePay)

“ **Rebrand note (2026-02-14):** FontePay was renamed to Drop. Some references to FontePay remain in this document (metric names, Sentry projects, API URLs). These should be updated when implementing the actual DevOps stack. Drop uses a PSD2 pass-through model — no wallet, no balance held by Drop.

## Table of Contents

1. [Executive Summary](#)
  2. [CI/CD Pipeline](#)
  3. [Testing Strategy](#)
  4. [Monitoring & Observability](#)
  5. [Error Tracking](#)
  6. [Alerting & Incident Management](#)
  7. [Documentation](#)
  8. [Security Operations](#)
  9. [Cost Summary](#)
  10. [Implementation Priority](#)
  11. [Integration Diagram](#)
- 

## 1. Executive Summary

# Stack Philosophy

Drop requires a DevOps/SRE stack that balances:

- **Fintech compliance** (audit trails, security, GDPR)
- **Cost efficiency** for MVP phase
- **Scalability** for growth to 100K+ users
- **EU data residency** where possible
- **Small team maintainability** (1-2 DevOps engineers)

## Recommended Stack Overview

Area	MVP Tool	Scale Tool	Reason
CI/CD	GitHub Actions	GitHub Actions + ArgoCD	Native GitHub, EU runners available
E2E Testing	Playwright	Playwright	Open-source, excellent mobile web
Load Testing	k6	k6 + Grafana Cloud	Grafana ecosystem, scriptable
APM	Grafana Cloud	Grafana Cloud	EU-hosted, cost-effective
Logs	Grafana Loki	Grafana Loki	Part of Grafana stack
Errors	Sentry	Sentry	Best-in-class, EU hosting
Alerts	Slack + PagerDuty	PagerDuty	Start simple, scale
Secrets	AWS Secrets Manager	AWS Secrets Manager	Native AWS, compliant
Security Scan	Snyk	Snyk + DAST	Developer-friendly

Total MVP Monthly Cost: EUR 800-1,200/month

Total Scale Monthly Cost: EUR 2,500-4,000/month

---

## 2. CI/CD Pipeline

### 2.1 Recommendation: GitHub Actions

## Why GitHub Actions over alternatives:

Criteria	GitHub Actions	GitLab CI	CircleCI
Native Integration	Best (GitHub)	Requires migration	Good
EU Runners	Yes (Azure EU)	Yes	Limited
Free Tier	2,000 min/month	400 min/month	6,000 min/month
Secrets Management	Native	Native	Native
Self-hosted Runners	Yes	Yes	Limited
Marketplace	Largest	Growing	Medium
Learning Curve	Low	Medium	Medium
OIDC for AWS	Native	Requires setup	Requires setup

### Decision: GitHub Actions

- Already using GitHub for source control
- Native OIDC integration with AWS (no long-lived credentials)
- EU-hosted runners available
- Excellent ecosystem of actions
- Cost-effective at scale

## 2.2 Pipeline Architecture

```
# .github/workflows/main.yml structure
```

Triggers:

- push to main/develop
- pull request
- manual dispatch

Jobs:

1. lint-and-format
  - ESLint, Prettier
  - Parallel for speed
2. security-scan
  - Snyk dependency check
  - Secret scanning
  - SAST (CodeQL)

3. test-unit
  - Jest (backend/frontend)
  - Coverage threshold: 80%
4. test-integration
  - Database tests
  - API contract tests
5. build
  - Docker image build
  - Multi-arch (amd64/arm64)
6. test-e2e (staging only)
  - Playwright
  - Against staging environment
7. deploy-staging
  - Automatic on develop merge
8. deploy-production
  - Manual approval required
  - Canary deployment

## 2.3 Deployment Strategies

### MVP Phase: Rolling Deployment

- Simple, works with small user base
- Zero-downtime with K8s rolling updates
- Easy rollback

### Scale Phase: Canary Deployment

Production Traffic:

- └─ 95% → Current Version
- └─ 5% → New Version (canary)

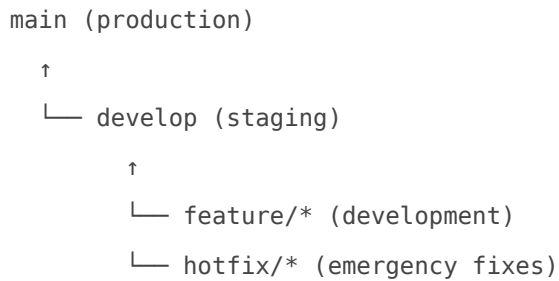
Promotion: Manual after metrics validation

Rollback: Automatic on error rate spike

### Implementation: ArgoCD + Argo Rollouts

- GitOps model (infrastructure as code)
- Automated sync from Git
- Progressive delivery
- Audit trail of all deployments

## 2.4 Branch Strategy



### Rules:

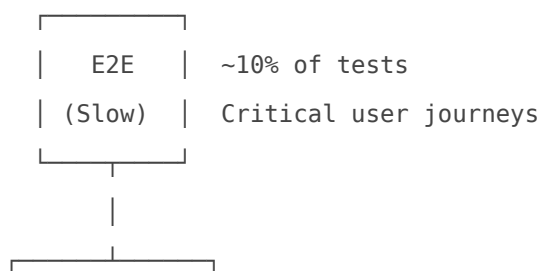
- `main`: Protected, requires PR + approval + passing CI
- `develop`: Protected, requires PR + passing CI
- Feature branches: Deleted after merge
- Hotfixes: Can bypass develop in emergencies

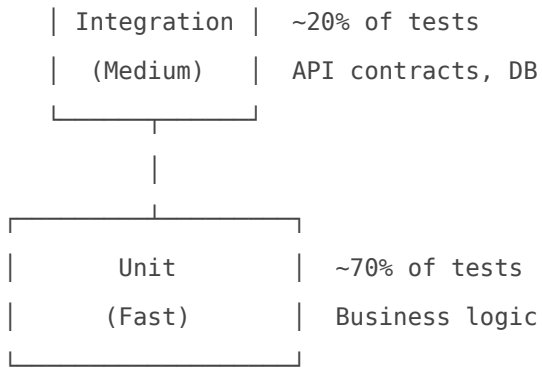
## 2.5 GitHub Actions Cost Estimate

Phase	Minutes/Month	Cost
MVP (5 devs)	~3,000	Free (2,000) + EUR 20
Scale (15 devs)	~15,000	EUR 120/month

## 3. Testing Strategy

### 3.1 Testing Pyramid





## 3.2 Unit Testing

**Current Stack:** Jest (already configured)

**Coverage Requirements:**

Component	Minimum	Target
Business Logic	90%	95%
API Controllers	80%	90%
Utilities	70%	80%
UI Components	60%	70%

**Best Practices:**

- Test business logic, not implementation
- Mock external dependencies
- Use factories for test data
- Run on every commit

## 3.3 Integration Testing

**Tools:**

- **Testcontainers** - Spin up PostgreSQL, Redis in Docker
- **Supertest** - HTTP assertions for API testing
- **Pact** - Contract testing between services

**What to Test:**

- Database queries (with real PostgreSQL)
- Redis caching behavior
- API contract between services

- BaaS webhook handlers
- Payment flow integration (sandbox)

## 3.4 E2E Testing

### Recommendation: Playwright

Criteria	Playwright	Cypress
Browser Support	All major + mobile	Chrome, Firefox, Edge
Speed	Faster (parallel)	Slower
Auto-wait	Built-in	Built-in
Mobile Testing	Better (device emulation)	Limited
CI Integration	Excellent	Good
Cost	Free	Free (cloud paid)
Learning Curve	Medium	Lower

### Decision: Playwright

- Better mobile web testing (critical for Drop)
- True parallel execution
- Multiple browser contexts
- API testing built-in
- Network interception for mocking

### Critical User Journeys to Test:

1. User registration + KYC start
2. Login flow (email + biometric)
3. View balance and transactions
4. Send P2P transfer
5. Card top-up flow
6. Card freeze/unfreeze
7. SEPA transfer initiation

### Playwright Configuration:

```
// playwright.config.ts
{
  projects: [
    { name: 'Desktop Chrome', use: { ...devices['Desktop Chrome'] } },
    { name: 'Mobile Safari', use: { ...devices['iPhone 14'] } },
  ],
}
```

```
{ name: 'Mobile Chrome', use: { ...devices['Pixel 7'] } },  
],  
retries: 2,  
reporter: [['html'], ['junit', { outputFile: 'results.xml' }]],  
}
```

## 3.5 Load Testing

### Recommendation: k6

#### Why k6:

- Open-source, scriptable in JavaScript
- Integrates with Grafana (our monitoring stack)
- Cloud option available for distributed load
- Can run locally or in CI/CD

#### Load Test Scenarios:

Scenario	Virtual Users	Duration	Success Criteria
Baseline	50	5 min	p95 < 500ms
Peak	200	10 min	p95 < 1000ms
Stress	500	5 min	No crashes
Soak	100	1 hour	No memory leaks

#### Critical Endpoints:

- `POST /api/auth/login` - 100 req/sec target
- `GET /api/accounts/balance` - 500 req/sec target
- `POST /api/transfers` - 50 req/sec target
- `GET /api/transactions` - 200 req/sec target

## 3.6 Security Testing

#### SAST (Static Analysis):

- **CodeQL** (GitHub native) - Free, good coverage
- **Snyk Code** - Better for JavaScript/TypeScript
- **SonarQube** - Alternative if self-hosted preferred

#### DAST (Dynamic Analysis):

- **OWASP ZAP** - Free, CI-integrated
- **Burp Suite** - For manual penetration testing

### Dependency Scanning:

- **Snyk** - Primary recommendation
- **Dependabot** - Free, GitHub native (backup)

### Schedule:

Test Type	Frequency	Blocker?
SAST	Every PR	Yes (high severity)
Dependency Scan	Daily	Yes (critical)
DAST	Weekly	No (review)
Pen Test	Quarterly	N/A (manual)

## 4. Monitoring & Observability

### 4.1 Strategy: Unified Grafana Stack

#### Why Grafana Cloud over alternatives:

Criteria	Grafana Cloud	Datadog	New Relic
EU Hosting	Yes (Frankfurt)	Yes	Yes
Pricing Model	Usage-based	Per-host	Per-user
MVP Cost	EUR 0-200	EUR 400+	EUR 300+
Scale Cost	EUR 500-1,000	EUR 2,000+	EUR 1,500+
Open Standards	Full (Prometheus, OTel)	Partial	Partial
Vendor Lock-in	Low	High	High
Self-host Option	Yes (fallback)	No	No

#### Decision: Grafana Cloud

- Best cost/value for startup
- EU data residency (Frankfurt region)
- Open standards (can migrate if needed)
- Unified platform (metrics, logs, traces)
- Free tier generous for MVP

## 4.2 Metrics (Prometheus + Grafana)

### Infrastructure Metrics:

- CPU, Memory, Disk, Network
- Kubernetes pod health
- Database connections, query latency
- Redis hit/miss ratio

### Application Metrics:

- Request rate, latency, error rate (RED)
- Active users (DAU/MAU)
- Transaction volume and value
- KYC conversion funnel
- Card activation rate

### Business Metrics (Custom):

```
fontelepay_transactions_total{type="p2p|sepa|card"}
fontelepay_transaction_value_eur{type="p2p|sepa|card"}
fontelepay_users_registered_total
fontelepay_users_kyc_passed_total
fontelepay_cards_issued_total{type="virtual|physical"}
fontelepay_api_latency_seconds{endpoint="/api/..."}
```

## 4.3 Log Aggregation (Loki)

### Why Loki:

- Part of Grafana stack (unified UI)
- Cost-effective (indexes labels, not content)
- Kubernetes native
- Query language similar to Prometheus

### Log Structure (JSON):

```
{
  "timestamp": "2026-02-05T10:30:00Z",
  "level": "info",
  "service": "payment-service",
  "trace_id": "abc123",
  "user_id": "usr_xxx", // pseudonymized
```

```
"message": "Transfer initiated",  
"amount_eur": 100,  
"transfer_type": "sepa"  
}
```

### Retention Policy:

Log Type	Retention	Reason
Application	30 days	Debugging
Security/Audit	7 years	Compliance
Access Logs	90 days	Security review

### GDPR Considerations:

- No PII in logs (use pseudonymized IDs)
- User IDs hashed or tokenized
- IP addresses masked after 30 days

## 4.4 Distributed Tracing (Tempo)

### Implementation: OpenTelemetry

#### Why OpenTelemetry:

- Vendor-neutral standard
- Supports all our languages (Java, Node.js, Dart)
- Auto-instrumentation available
- Future-proof (industry standard)

#### Trace Critical Paths:

1. User login (app -> API -> auth -> DB)
2. Payment initiation (app -> API -> payment -> BaaS -> ledger)
3. Card transaction (webhook -> processor -> notification)

#### Sampling Strategy:

- 100% for errors
- 100% for slow requests (>1s)
- 10% for successful requests (MVP)
- 1% for successful requests (scale)

## 4.5 Real User Monitoring (RUM)

## For Web (Next.js):

- Grafana Faro (free, part of Grafana)
- Captures: Page load, Web Vitals, JS errors

## For Mobile (Flutter):

- Custom implementation with OpenTelemetry
- Track: App start time, screen transitions, API calls

## Key Metrics:

Metric	Target	Threshold
LCP (Largest Contentful Paint)	<2.5s	<4s
FID (First Input Delay)	<100ms	<300ms
CLS (Cumulative Layout Shift)	<0.1	<0.25
App Cold Start	<2s	<3s
API Response (p95)	<500ms	<1s

## 4.6 Grafana Cloud Cost Estimate

Component	MVP Usage	MVP Cost	Scale Usage	Scale Cost
Metrics	10K series	Free	50K series	EUR 150
Logs	50 GB/mo	Free	200 GB/mo	EUR 200
Traces	10 GB/mo	Free	50 GB/mo	EUR 100
<b>Total</b>	-	<b>EUR 0-50</b>	-	<b>EUR 450</b>

## 5. Error Tracking

### 5.1 Recommendation: Sentry

#### Comparison:

Criteria	Sentry	Bugsnag	Rollbar
EU Hosting	Yes	Yes	No
Flutter SDK	Excellent	Good	Limited

Criteria	Sentry	Bugsnag	Rollbar
Source Maps	Automatic	Automatic	Manual
Performance	Included	Separate	Included
Pricing (MVP)	Free	EUR 100	EUR 100
Pricing (Scale)	EUR 300	EUR 400	EUR 350
Slack Integration	Native	Native	Native
Issue Grouping	Best	Good	Good

### Decision: Sentry

- Best Flutter support (critical for mobile)
- EU data residency available
- Excellent source map integration
- Issue grouping reduces noise
- Performance monitoring included
- Generous free tier (5K errors/month)

## 5.2 Sentry Configuration

### Projects:

- `fontelepay-web` (Next.js frontend)
- `fontelepay-api` (Node.js/Java backend)
- `fontelepay-mobile` (Flutter app)

### Settings:

```
// sentry.config.js
{
  dsn: "https://xxx@sentry.io/xxx",
  environment: process.env.NODE_ENV,
  release: process.env.GIT_SHA,
  tracesSampleRate: 0.1, // 10% of transactions

  // Filter sensitive data
  beforeSend(event) {
    // Remove PII
    if (event.user) {
      delete event.user.email;
      delete event.user.ip_address;
    }
  }
}
```

```
}  
  return event;  
}  
}
```

### Alert Rules:

Condition	Action	Priority
New issue (high severity)	Slack + PagerDuty	P1
Issue spike (>10x baseline)	Slack + PagerDuty	P1
New issue (medium)	Slack only	P2
Regression (resolved reopened)	Slack	P2

## 5.3 Source Maps

### Web (Next.js):

- Automatic upload via `@sentry/nextjs`
- Hidden from production (security)

### Mobile (Flutter):

- Upload dSYM (iOS) and mapping files (Android)
- Integrated with CI/CD

## 5.4 Sentry Cost Estimate

Phase	Events/Month	Cost
MVP	<5,000	Free
Growth	~50,000	EUR 26/month
Scale	~500,000	EUR 300/month

# 6. Alerting & Incident Management

## 6.1 Phased Approach

**MVP (Team <5): Slack + Grafana Alerts**

- Simple, no additional cost
- On-call rotation manual
- Suitable for low traffic

### Growth (Team 5-15): Add PagerDuty

- Proper escalation policies
- On-call schedules
- Mobile alerts
- Incident timeline

### Scale (Team 15+): Full Incident Management

- PagerDuty + Statuspage
- War room automation
- Post-incident reviews

## 6.2 Alert Levels

Level	Response Time	Examples	Notification
P1 - Critical	15 min	Payment processing down, data breach	PagerDuty + Slack + SMS
P2 - High	1 hour	High error rate, degraded performance	PagerDuty + Slack
P3 - Medium	4 hours	Non-critical service degraded	Slack only
P4 - Low	Next business day	Warning thresholds	Slack (daily digest)

## 6.3 Critical Alerts (P1)

Alert	Condition	Action
API Down	0 successful requests for 2 min	Page on-call
Payment Failures	>5% failure rate for 5 min	Page on-call
Database Unreachable	Connection failures >10/min	Page on-call
Security Event	Suspicious activity detected	Page on-call + security
Error Spike	10x baseline errors	Page on-call

## 6.4 On-Call Rotation

### MVP Setup:

Week 1: Dev A (primary)  
Week 2: Dev B (primary)  
Week 3: Dev A (primary)  
...

Escalation:

- 0-15 min: Primary on-call
- 15-30 min: Secondary on-call
- 30+ min: Engineering lead

**PagerDuty Cost:**

Plan	Cost	Features
Free	EUR 0	5 users, basic
Professional	EUR 21/user/mo	Full features

**MVP: Free tier (5 users) Scale: Professional for core team**

## 6.5 Incident Response Runbook Template

```
## Incident: [Title]

### Detection
- Alert source: [Grafana/Sentry/PagerDuty]
- Time detected: [timestamp]
- Severity: [P1/P2/P3]

### Impact
- Users affected: [estimate]
- Services affected: [list]
- Financial impact: [if applicable]

### Timeline
- HH:MM - [Event]
- HH:MM - [Event]

### Root Cause
[Description]
```

### Resolution

[Steps taken]

### Action Items

- [ ] [Preventive measure]
- [ ] [Process improvement]

### Participants

- Incident Commander: [name]
- Responders: [names]

# 7. Documentation

## 7.1 API Documentation

**Recommendation: OpenAPI 3.1 + Swagger UI**

**Why:**

- Industry standard
- Auto-generated from code annotations
- Interactive testing
- Client SDK generation

**Implementation:**

```
# openapi.yaml (partial)
openapi: 3.1.0
info:
  title: Drop API
  version: 1.0.0
  description: Mobile banking API

servers:
  - url: https://api.fontelepay.com/v1
    description: Production
  - url: https://api.staging.fontelepay.com/v1
    description: Staging
```

```
security:
  - bearerAuth: []

paths:
  /accounts/{id}/balance:
    get:
      summary: Get account balance
      tags: [Accounts]
      ...
```

### Hosting:

- Swagger UI at `/docs` endpoint
- Redoc as alternative (cleaner for external)
- Postman collection export for testing

## 7.2 Runbooks

**Location:** `/docs/runbooks/` in repository

### Required Runbooks:

Runbook	Purpose
<code>deploy-production.md</code>	Production deployment steps
<code>rollback.md</code>	How to rollback a bad deploy
<code>database-migration.md</code>	Safe DB migration process
<code>incident-response.md</code>	General incident handling
<code>scaling.md</code>	How to scale services
<code>secrets-rotation.md</code>	Rotating API keys, certs
<code>disaster-recovery.md</code>	Full recovery procedures

### Runbook Template:

```
# Runbook: [Title]

## Overview
[What this runbook covers]

## Prerequisites
```

- [ ] Access to [system]
- [ ] Permissions: [list]

### ## Steps

1. [Step with command examples]
2. [Step with verification]

### ## Verification

[How to confirm success]

### ## Rollback

[If something goes wrong]

### ## Contacts

- Primary: [name/slack]
- Escalation: [name/slack]

## 7.3 Architecture Decision Records (ADRs)

**Location:** `/docs/adr/` in repository

### Format:

```
# ADR-001: Use PostgreSQL as Primary Database

## Status
Accepted

## Context
We need a reliable, ACID-compliant database for financial transactions.

## Decision
Use PostgreSQL 16 as our primary database.

## Consequences
### Positive
- Strong ACID compliance
- Excellent JSON support
- Proven in fintech
```

### ### Negative

- Requires more ops than managed NoSQL
- Horizontal scaling more complex

### ## Alternatives Considered

- MySQL: Less JSON support
- MongoDB: Not ACID by default
- CockroachDB: Higher cost, complexity

## Key ADRs to Create:

- ADR-001: Database selection (PostgreSQL)
- ADR-002: Cloud provider (AWS)
- ADR-003: BaaS provider (Swan)
- ADR-004: Mobile framework (Flutter)
- ADR-005: Monitoring stack (Grafana)
- ADR-006: CI/CD platform (GitHub Actions)

## 7.4 Documentation Tooling

Type	Tool	Cost
API Docs	Swagger/OpenAPI	Free
Internal Docs	Notion or Confluence	Free-EUR 50/mo
Runbooks	Git repository	Free
Diagrams	Mermaid (in Markdown)	Free
Postmortems	Notion template	Free

## 8. Security Operations

### 8.1 Dependency Scanning

**Recommendation: Snyk**

#### Why Snyk:

- Best JavaScript/TypeScript support
- Dart/Flutter support
- Automatic PR fixes

- License compliance
- Container scanning

### Integration:

```
# .github/workflows/security.yml
- name: Snyk Security Scan
  uses: snyk/actions/node@master
  with:
    args: --severity-threshold=high
```

### Policy:

Severity	Action	SLA
Critical	Block PR, fix immediately	24 hours
High	Block PR, fix before merge	72 hours
Medium	Warning, fix in sprint	2 weeks
Low	Track, fix when convenient	1 month

### Snyk Cost:

Plan	Cost	Limits
Free	EUR 0	200 tests/month
Team	EUR 52/dev/mo	Unlimited

**MVP: Free tier Scale: Team plan**

## 8.2 Secret Management

**Recommendation: AWS Secrets Manager**

### Why AWS Secrets Manager:

- Native AWS integration (using AWS already)
- Automatic rotation support
- Audit trail via CloudTrail
- GDPR compliant (EU region)
- No additional infrastructure

### Alternative: HashiCorp Vault

- More features but more operational overhead

- Consider for Scale phase if multi-cloud

### Secrets to Manage:

Secret	Rotation	Access
Database credentials	90 days	Backend services
API keys (Swan, Stripe)	180 days	Backend services
JWT signing keys	365 days	Auth service
Encryption keys	Never (versioned)	All services

### Implementation:

```
// secrets.ts
import { SecretsManager } from '@aws-sdk/client-secrets-manager';

const client = new SecretsManager({ region: 'eu-central-1' });

export async function getSecret(name: string): Promise<string> {
  const response = await client.getSecretValue({ SecretId: name });
  return response.SecretString!;
}
```

### AWS Secrets Manager Cost:

Secrets	Cost
10 secrets	EUR 4/month
50 secrets	EUR 20/month
100 secrets	EUR 40/month

## 8.3 Penetration Testing

### Schedule:

Test Type	Frequency	Provider
Automated DAST	Weekly	OWASP ZAP
Web App Pen Test	Quarterly	External firm
Mobile App Pen Test	Quarterly	External firm
Infrastructure Pen Test	Annually	External firm

## Budget:

Test	Cost
Web + API Pen Test	EUR 5,000-10,000
Mobile Pen Test	EUR 5,000-8,000
Infrastructure	EUR 8,000-15,000
<b>Annual Total</b>	<b>EUR 25,000-45,000</b>

## EU-Based Pen Testing Firms:

- Cure53 (Germany) - Excellent reputation
- Securitum (Poland) - Cost-effective
- WithSecure (Finland) - Enterprise grade
- Secura (Netherlands) - Banking expertise

# 8.4 Security Monitoring

## SIEM Considerations:

- MVP: CloudWatch + Grafana alerts (sufficient)
- Scale: Consider AWS Security Hub or Elastic SIEM

## Security Alerts:

Event	Action
Failed login spike	Alert + temp block
New device login	User notification
Large transfer	Manual review queue
Admin action	Audit log + alert
API key usage anomaly	Alert + investigate

# 8.5 Compliance Automation

## Tools:

- **AWS Config** - Configuration compliance
- **Prowler** - AWS security assessment (free)
- **Checkov** - Infrastructure as code scanning

## Automated Checks:

- S3 buckets not public
- Encryption at rest enabled
- Security groups not overly permissive
- IAM policies least-privilege
- Audit logging enabled

## 9. Cost Summary

### 9.1 MVP Phase (Monthly)

Category	Tool	Cost (EUR)
CI/CD	GitHub Actions	20-50
Monitoring	Grafana Cloud (free tier)	0-50
Error Tracking	Sentry (free tier)	0
Alerting	Slack + PagerDuty Free	0
Security	Snyk (free tier)	0
Secrets	AWS Secrets Manager	10
Testing	Playwright, k6 (OSS)	0
<b>Total</b>		<b>EUR 30-110</b>

### 9.2 Growth Phase (Monthly)

Category	Tool	Cost (EUR)
CI/CD	GitHub Actions	100-150
Monitoring	Grafana Cloud	200-400
Error Tracking	Sentry Team	100-300
Alerting	PagerDuty Professional	100-200
Security	Snyk Team	200-400
Secrets	AWS Secrets Manager	20-40
Testing	k6 Cloud (load testing)	100-200
<b>Total</b>		<b>EUR 820-1,690</b>

### 9.3 Scale Phase (Monthly)

Category	Tool	Cost (EUR)
CI/CD	GitHub Actions + ArgoCD	200-300
Monitoring	Grafana Cloud	500-1,000
Error Tracking	Sentry Business	300-500
Alerting	PagerDuty + Statuspage	300-500
Security	Snyk + DAST	500-800
Secrets	AWS Secrets Manager	40-60
Testing	k6 Cloud	200-400
Documentation	Confluence	50-100
<b>Total</b>		<b>EUR 2,090-3,660</b>

## 9.4 Annual Security Costs

Item	Cost (EUR)
Penetration Testing (4x/year)	25,000-45,000
Compliance Audit (annual)	10,000-20,000
Security Training	2,000-5,000
<b>Total</b>	<b>EUR 37,000-70,000</b>

# 10. Implementation Priority

## 10.1 Phase 1: Foundation (Week 1-2)

### Must Have:

- GitHub Actions basic pipeline (lint, test, build)
- Sentry error tracking (all environments)
- Basic Slack alerting
- AWS Secrets Manager setup
- Snyk dependency scanning

**Outcome:** Can deploy safely with visibility into errors

## 10.2 Phase 2: Observability (Week 3-4)

### Must Have:

- Grafana Cloud setup (metrics, logs)
- Prometheus metrics in application
- Structured logging (JSON)
- Basic dashboards (RED metrics)
- Critical alerts configured

**Outcome:** Can monitor application health

## 10.3 Phase 3: Testing (Week 5-6)

### Must Have:

- Unit test coverage >70%
- Integration tests for critical paths
- Playwright E2E for happy paths
- k6 load test baseline
- Test runs in CI/CD

**Outcome:** Confidence in deployments

## 10.4 Phase 4: Security (Week 7-8)

### Must Have:

- CodeQL SAST enabled
- OWASP ZAP in staging
- Security headers configured
- Audit logging implemented
- First penetration test scheduled

**Outcome:** Security baseline established

## 10.5 Phase 5: Operations (Week 9-12)

### Should Have:

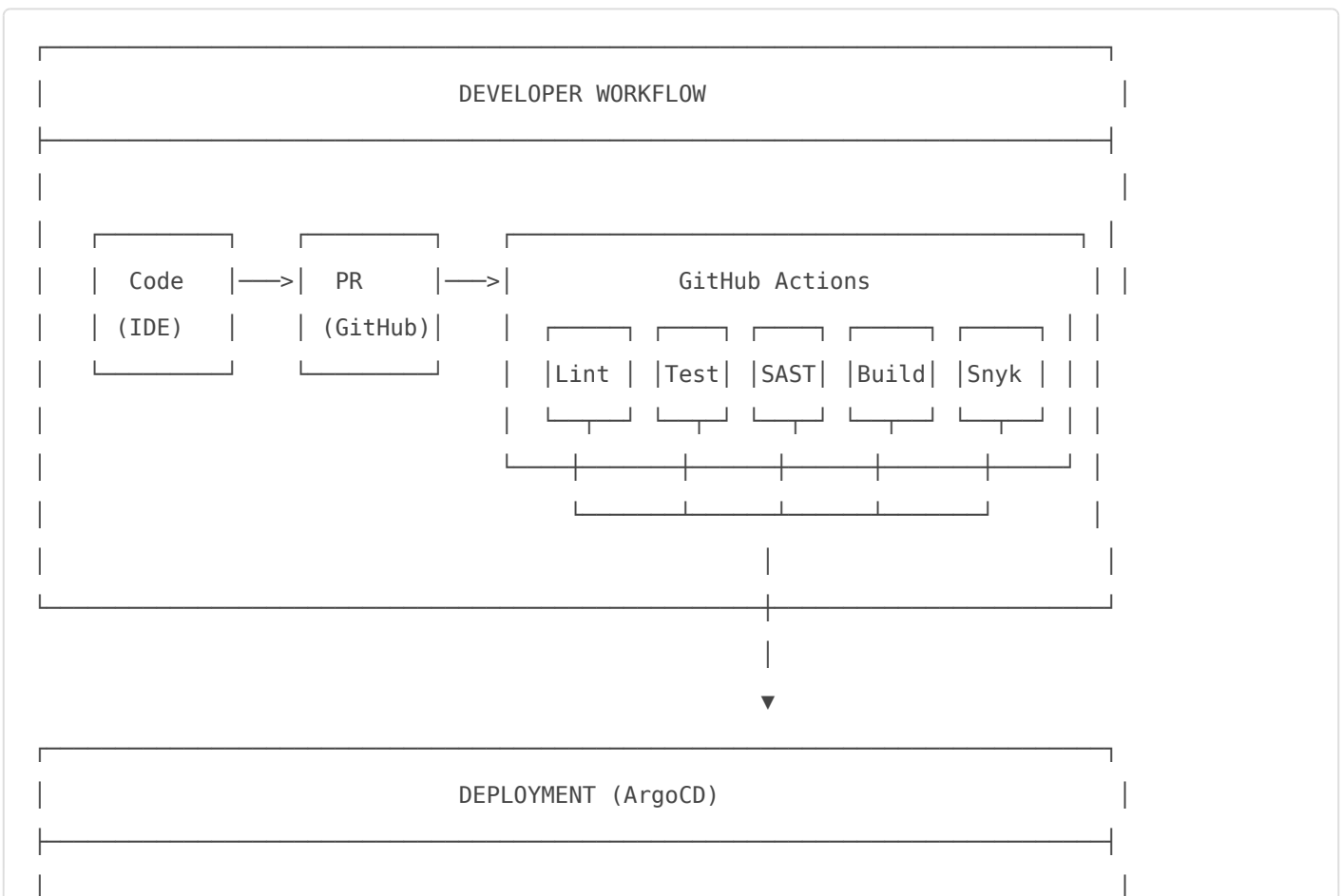
- PagerDuty on-call rotation
- Runbooks for critical scenarios
- Disaster recovery tested
- OpenAPI documentation complete
- ADRs documented

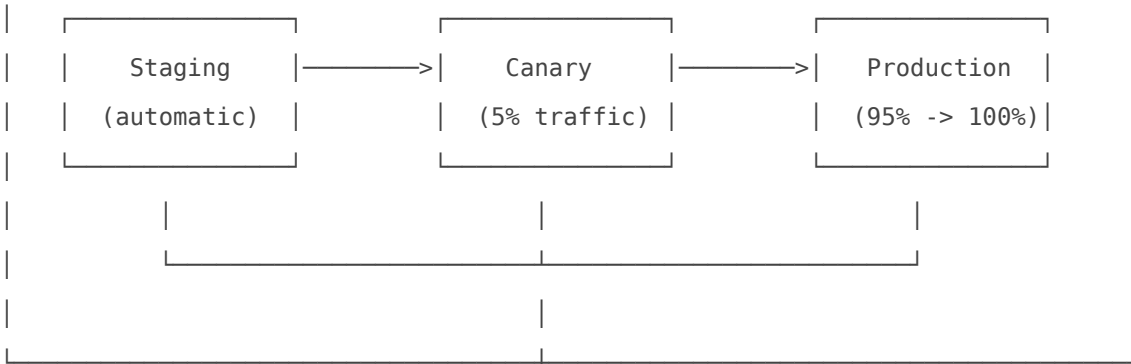
**Outcome:** Production-ready operations

## 10.6 Checklist Summary

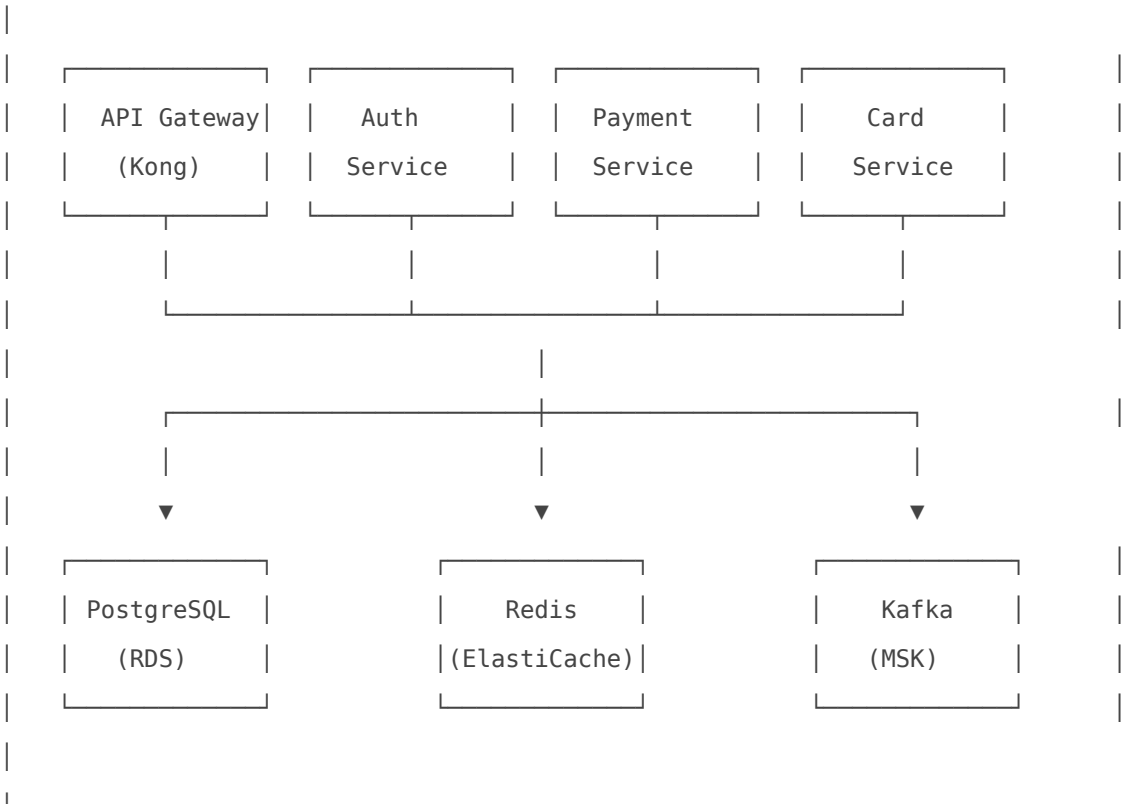
Week 1-2: CI/CD + Errors + Secrets  
 Week 3-4: Monitoring + Logs + Alerts  
 Week 5-6: Tests + E2E + Load  
 Week 7-8: Security + Audit + Pen Test  
 Week 9-12: On-call + Docs + DR

# 11. Integration Diagram





KUBERNETES CLUSTER (AWS EKS)

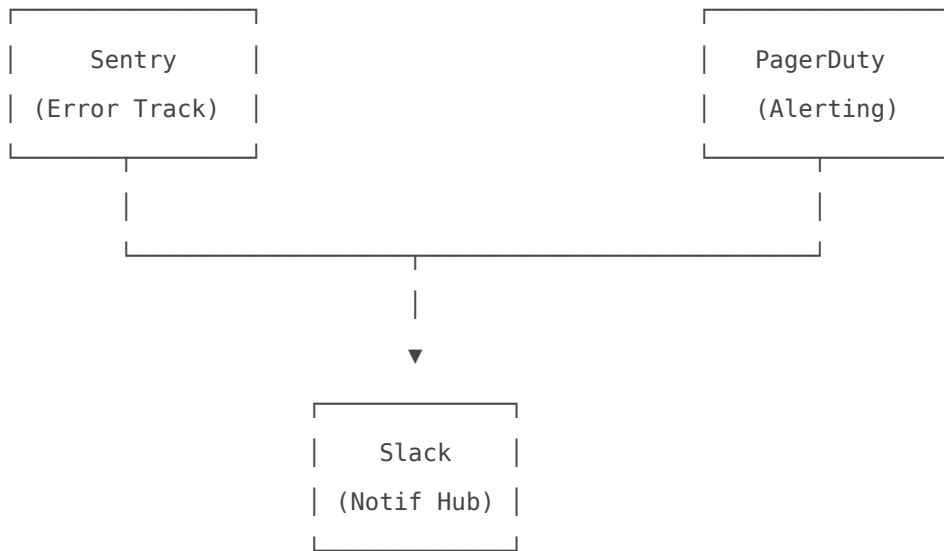
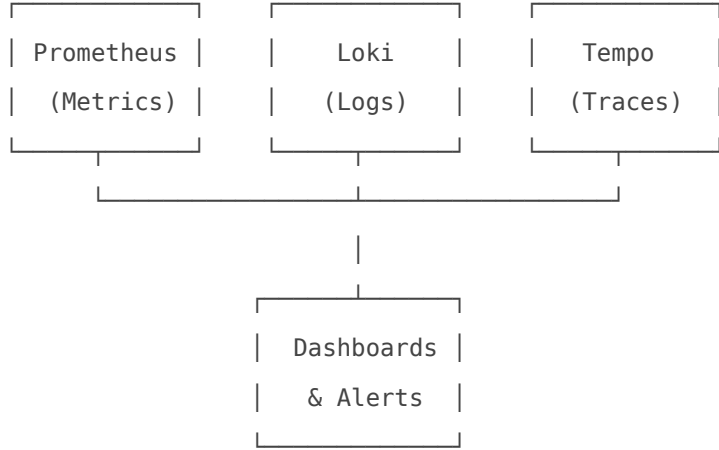


Telemetry



OBSERVABILITY STACK





SECURITY LAYER



# Appendix A: Tool Links

Tool	URL	Purpose
GitHub Actions	github.com/features/actions	CI/CD
ArgoCD	argoproj.github.io/cd	GitOps deployment
Grafana Cloud	grafana.com/cloud	Monitoring
Sentry	sentry.io	Error tracking
PagerDuty	pagerduty.com	Incident management
Snyk	snyk.io	Security scanning
Playwright	playwright.dev	E2E testing
k6	k6.io	Load testing
OpenTelemetry	opentelemetry.io	Observability

# Appendix B: Decision Matrix

Decision	Options Considered	Winner	Key Factor
CI/CD	GitHub Actions, GitLab, CircleCI	GitHub Actions	Native GitHub, EU runners
Monitoring	Datadog, New Relic, Grafana	Grafana Cloud	Cost, EU hosting, open standards
E2E Testing	Playwright, Cypress	Playwright	Mobile web support, speed
Error Tracking	Sentry, Bugsnag, Rollbar	Sentry	Flutter SDK, EU hosting
Alerting	PagerDuty, Opsgenie, Slack	PagerDuty	Industry standard, free tier
Secrets	AWS SM, Vault, GCP SM	AWS Secrets Manager	Already on AWS, simple
Security	Snyk, Dependabot, Sonar	Snyk	Best JS/TS coverage

# Appendix C: Compliance Mapping

Requirement	Solution	Evidence
PCI DSS 10.x (Logging)	Grafana Loki, 7yr retention	CloudTrail + Loki
GDPR (Data Residency)	Grafana EU, Sentry EU	Region configs

Requirement	Solution	Evidence
GDPR (Right to Erasure)	Pseudonymized logs	No PII in logs
SOC 2 (Change Mgmt)	GitHub PRs, ArgoCD	Audit trail
ISO 27001 (Incident)	PagerDuty, Runbooks	Incident records

---

*Document created: 2026-02-05 Last updated: 2026-02-05 Author: DevOps Research*

---

Revision #4

Created 2026-02-18 08:44:29 UTC by John

Updated 2026-05-23 10:58:50 UTC by John