PLOT – Sandbox Cloud Integration Plan (v1.0)

Zweck: Reproduzierbare, EU-konforme Sandbox-Umgebung für PLOT, um Integrationen (ERP, DMS/Plancloud, BIM, Telematik) sicher zu testen. Fokus auf API-Kohärenz, Event-Flows, RBAC, Observability und Datenminimierung.

---

1) Ziele & Leitplanken

- **Realitätsnah**: Repräsentative Datenmodelle und Eventraten, ohne Produktivsysteme zu berühren.
- **Sicher**: Keine produktiven Identitäten/Secrets; PII minimieren; EU-Region.
- **Reproduzierbar**: IaC (Terraform/Helm), deterministische Seeds, idempotente Pipelines.
- **Messbar**: KPIs (OTD, Turnaround, Comm-Laufzeit, Nacharbeit) als Sandbox-Dashboards.

---

2) Topologie & Umgebungen

- **local** (Entwickler): Docker Compose, Everything-in-a-Box.
- **sandbox** (Cloud): geteilte Testumgebung pro Branch/Feature über **Ephemeral Namespaces**.
- **staging**: Pre-Prod, reale Drittsystem-Sandboxes, Lasttests.

Netz: VPC `eu-central-1` (oder äquivalent), Subnetze `public`/`private`, egress via NAT. WAF vor Ingress.

Kubernetes: 1× Cluster `plot-sbx`, Namespaces je Feature: `plot-sbx-<branch>`.

---

3) Kern-Services (Sandbox Deploy)

- **plot-core**: Deliveries, Slots, Logistics Comms, KPIs.
- **plot-auth**: OIDC Provider (Sandbox), RBAC/Policy Engine (OPA).
- **plot-erp-adapter**: ERP-Reader/Writer gegen Stub oder reales ERP-Test.
- **plot-docs**: DMS/Plancloud Connector (signed URLs, revisions).
- **plot-bim**: IFC/BCF Links, Viewer-Proxy.
- **plot-telemetry**: Webhooks für Carrier, Geofences.
- **schema-registry**: Event-Schemas (JSON/Avro), Validierung.
- **event-bus**: Redpanda/Kafka (Sandbox), Topics unter `sbx.*`.
- **db**: Postgres, `pg_partman` für Event-/Audit-Tabellen.
- **objstore**: MinIO (S3-kompatibel) für Attachments/Pläne.

---

4) Integrationen & Stubs

ERP

- Modi: `MOCK` (WireMock), `REPLAY` (gespeicherte Antworten), `SANDBOX` (echte ERP-Testumgebung).
- Endpunkte (Beispiele):
- `GET /api/erp/purchase-orders?updated_since=...`
- `POST /api/erp/events` (PLOT→ERP; idempotent via `X-Idempotency-Key`).

DMS/Plancloud

- Signed URL-Fluss: `POST /api/docs/signed-upload`, `GET /api/docs/{id}/signed-download`.
- Plan-Revisionen als Metadaten (Rev, gültig-ab, superseded).

BIM

- BCF-Mock: `POST /api/bim/bcf/issues`, `GET /api/bim/bcf/issues/{id}`.
- IFC-Referenzen: Zuordnung `delivery_item.ifc_guid`.

Telematik/Carrier

- Webhook: `POST /api/carriers/{id}/telemetry` (events: `geofence.enter/exit`, `eta.update`).
- Simulator (ratebasiert): 5–20 Fahrzeuge/Projekt; p50 ETA-Update alle 60 s.

---

5) Events & Schemata

Konvention: `sbx.<domäne>.<ereignis>.v1` (z. B. `sbx.delivery.slot_booked.v1`).

Beispiel (JSON Schema)

```
{
  "$id": "https://schemas.plot/sbx.delivery.slot_booked.v1.json",
  "type": "object",
  "required": ["event", "occurred_at", "actor", "data", "correlation_id"],
  "properties": {
    "event": {"const": "sbx.delivery.slot_booked.v1"},
    "occurred_at": {"type": "string", "format": "date-time"},
    "actor": {"type": "object", "properties": {"user_id": {"type":
"string"}, "role": {"type": "string"}}},
    "correlation_id": {"type": "string"},
    "data": {
      "type": "object",
      "properties": {
        "project_id": {"type": "string"},
        "delivery_id": {"type": "string"},
        "slot": {
```

```
          "type": "object",
          "properties": {
            "start": {"type": "string", "format": "date-time"},
            "end": {"type": "string", "format": "date-time"},
            "zone": {"type": "string"}
          },
          "required": ["start", "end", "zone"]
        }
      },
      "required": ["project_id", "delivery_id", "slot"]
    }
  }
}
```

---

6) RBAC & Testidentitäten

Rollen: `SUI`, `FE`, `SUB`, `CAR`, `AUD`.

Test-Tenants & User - Tenant: `ten-SBX-ACME`, Projekte: `prj-SBX-001`, `prj-SBX-002`. - Nutzer: `sui@sbx.plot`, `fe@sbx.plot`, `sub@sbx.plot`, `car@sbx.plot`, `aud@sbx.plot`.

Policies (OPA-Snippet)

```
package plot.authz

default allow = false

allow {
  input.role == "SUB"
  input.action == "read"
  input.resource.type == "delivery"
  input.resource.owner_org == input.subject.org
}

allow {
  input.role == "SUI"
  input.action in {"create", "update", "delete", "read"}
}
```

---

7) KPI-Erhebung in der Sandbox

- **OTD%**: Gate-Zeitpunkt ≤ `slot_start + 15m`.
- **Turnaround**: `dock_out - gate_in` (Median/p90 je Projekt/Zone).
- **Comm-Laufzeit**: `ack - broadcast` pro Rolle.
- **Nacharbeit**: `discrepancy=true` Anteil je Warengruppe.

Synthetic KPI-Seeder: Generiert realistische Verteilungen (Stoßzeiten 7–9/12–14 Uhr, Ausreißer, Wochenmuster).

---

8) Compliance & Audit (Sandbox)

- **Audit-Log**: append-only, Hash-Kette, Export `JSON/CSV`.
- **PII-Hygiene**: Platzhalter-MSISDNs, Test-E-Mails `@example.test`.
- **Datenresidenz**: EU-Buckets, regionale KMS-Keys.
- **Legal Hold** Flags auch in Sandbox testbar.

---

9) Observability

- **Tracing**: OpenTelemetry (HTTP, DB, Kafka), Trace-IDs in Logs.
- **Metriken**: Services (`latency`, `error_rate`), Business KPIs.
- **Dashboards**: „Sandbox Logistics“: OTD (Woche), p90 Turnaround, Top-Zonen nach Stau.
- **Alerting (low-noise)**: Schwellen & SLO-Budget, Stummschaltung je Namespace.

---

10) CI/CD (Ephemeral Envs)

Pipeline (Auszug) 1. Lint & Unit Tests. 2. Contract Tests (Pact) gegen ERP/DMS/BIM/Carrier Stubs. 3. Build Images, signieren (Sigstore/Cosign). 4. Deploy `plot-sbx-<branch>` Namespace via Helm. 5. Smoke Tests (Readiness, Health, Sample Flow `book-slot`). 6. Integrationstests (happy path + Fehlerpfade). 7. Promotion Gate → `sandbox` shared.

Feature Flags: z. B. `telemetry.geofence` toggelbar.

---

11) IaC-Skeleton

Terraform Struktur

```
infra/
  modules/
    network/
    k8s/
    postgres/
    minio/
    redpanda/
    observability/
  envs/
    sandbox/
      main.tf
      variables.tf
      backend.tf
```

Helm Values (Beispiel)

```
plot-core:
  image: registry.example/plot-core:{{ .Chart.AppVersion }}
  env:
    REGION: eu-central
    DB_URL: postgres://plot:***@postgres.sbx:5432/plot
    EVENT_BROKER: kafka://redpanda.sbx:9092
    KPI_GRACE_MINUTES: 15
```

---

12) Docker Compose (Local Mirror)

```
version: "3.9"
services:
  postgres:
    image: postgres:16
    environment:
      POSTGRES_DB: plot
      POSTGRES_USER: plot
      POSTGRES_PASSWORD: plot
    ports: ["5432:5432"]
  minio:
    image: minio/minio
    command: server /data
    environment:
      MINIO_ROOT_USER: minio
      MINIO_ROOT_PASSWORD: minio123
    ports: ["9000:9000", "9001:9001"]
  redpanda:
    image: redpandadata/redpanda:latest
    command: redpanda start --overprovisioned --smp 1 --memory 1G --reserve-
memory 0M
    ports: ["9092:9092"]
  erp-mock:
    image: wiremock/wiremock:3
    ports: ["8081:8080"]
    volumes:
      - ./mocks/erp:/home/wiremock
  dms-mock:
    image: wiremock/wiremock:3
    ports: ["8082:8080"]
    volumes:
      - ./mocks/dms:/home/wiremock
  telemetry-sim:
    image: ghcr.io/example/plot-telemetry-sim:latest
    environment:
      TARGET_URL: http://localhost:8080/api/carriers/sbx/telemetry
```

```
    RATE: 10 # events/min
plot-core:
  image: ghcr.io/example/plot-core:dev
  depends_on: [postgres, redpanda]
  ports: ["8080:8080"]
```

---

13) Beispiel-APIs (Sandbox)

Slot buchen

```
POST /api/deliveries/{id}/book-slot
Authorization: Bearer <sandbox-jwt>
X-Idempotency-Key: slot-dly-9b2f-2025-09-20T08:00Z

{
  "start": "2025-09-20T08:00:00Z",
  "end": "2025-09-20T08:30:00Z",
  "zone": "Gate-North"
}
```

Telemetry Webhook

```
POST /api/carriers/sbx/telemetry
{
  "vehicle_id": "TRK-4711",
  "delivery_id": "dly-9b2f",
  "event": "geofence.enter",
  "geofence_id": "gate-north",
  "timestamp": "2025-09-18T08:44:01Z"
}
```

---

14) Testdaten & Seeding

- **Referenzsätze**: 3 Projekte, 12 Lieferzonen, 150 POs (davon 20 Long-Lead), 600 Deliveries/Woche.
- **Zeitmuster**: Anlieferpeaks 07:30, 08:15, 12:30; Ausreißer (Wetter, Stau) simuliert.
- **Datenqualität**: 3–5 % Discrepancies, 1–2 % No-Shows.

Seeding via `plot-core seed --profile sbx-eu` (idempotent, bei erneutem Lauf nur Deltas).

---

15) Sicherheits-Checkliste (Sandbox)

- [ ] Getrennte Secrets (`sandbox.*`) in Secret Manager/KMS.
- [ ] Kein Versand realer SMS/E-Mails: Test-Provider/Blocklisten.

- [ ] IP-Access-Lists für Admin-UIs.
- [ ] Rotierende Schlüssel, Policy „least privilege“ je Namespace.

---

16) „Sandbox Ready“ – Abnahmekriterien

- **Deploy** in <15 min, Ephemeral Namespace aus PR.
- **Green** Smoke-Suite (Slot buchen, Telemetry empfangen, KPI berechnen).
- **Dash** zeigt OTD/Turnaround live.
- **Audit**-Export abrufbar, Hash-Kette validiert.
- **Docs**: Postman-Collection + Quickstart.

---

17) Nächste Schritte

1) Terraform Backend & Cloud-Konto verbinden, Cluster `plot-sbx` ausrollen. 2) Helm Charts deployen, Seeding fahren, Dashboards importieren. 3) Contract Tests mit Partnern (ERP/DMS/BIM/Carrier) aktivieren. 4) Feature Flags für Telemetrie und Logistics-Broadcasts toggeln.

> Ergebnis: Eine belastbare, EU-konforme Spielwiese, in der Integration echt wirkt – ohne Produktionsrisiko.