EntaENGELment · Darstellungsschema (P9-rc1+) — Merged

Stand: jetzt

Kompakte, aktuelle Zusammenführung von Architektur, Datenpfaden, Governance, Resonanz und Skalierung. **Integriert v0.7.4** (Ma'at-Audit, Nil-Zyklus, BoundaryPanel-Live, Takt/Lyra) sowie **P9-rc1** (EEG-Mapping + KPIs + `ev_final_run` Schema).

---

1) Meta-Topologie (Resonanz/Dipol)

Zwei Pole ↔ Vesica-Membran mit **Hard-Gate**. Resonanz-Kern (χ, ω, P, τ). Toroidaler Regel-Loop: **Edge → KPI → Governance → Audit → Explain → Edge**.

---

2) System-Stack (Layer)

L0 Sensorik (Edge): HRV, EDA, Pupille, EEG
L1 Ingest: `phybox/ingest.py` (RMSSD, SCR-Rate, dPupil/dt, EEG α/(β+θ))
L2 Synthese: χ, ω, P, τ (signal_map_v1; EWMA/Clamp/robust_z)
L3 KPIs: σ-Rolling, Artefakt-Rate, Drift-Flags
L4 Governance: Consent-Gating (**Hard-Gate**), Dual-Receipts, AEAD-Bindung
L5 Explain/UI: GovernanceBar, SensorPicker, M13c, Explain-Overlay
L6 Audit: Ledger, Root-Hash-Freeze, Nightly Verify
L7 Skalierung: FL (`fl_proto.yaml`), MPC (`mpc_proto.yaml`)
L8 Resonanz: Resonance Protocol v3 (Archetype-Card), Chronicles
L9 Seeds/Index: CP-π, RCC/Maxwell/DF, Policies, Cards, Protos

---

3) Kernbausteine (kompakt)

- **Hard-Gate**: GateOpen ⇔ (Φ≥Φ* ∧ RCC\:EC ∧ ¬PO ∧ $\|M\|_2$=1) → *kein Stream ohne gültigen Consent*.
- **Dual-Receipts**: Consent-Receipt + Governance-Receipt (UI-Badges).
- **AES-GCM PoC**: AAD = receipt_id | root_hash (Kontext-Bindung).
- **EEG-Term in χ**: EEG=α/(β+θ) mit EWMA (aggregate-only).
- **KPIs**: σ(χ,ω,P), Artefakte 0..1, Drift-Flags {χ,ω,P}.
- **Resonanz v3**: coherence_score, topological_fit, Chronicles-Events.
- ***Neu (v0.7.4-Align)***:
- **Gate-Block (operativ)**: `geff`, `threshold`, `dt_window_ms=[18,40]`, `refractory_ms=120`, `flip_allowed`, `phase∈{T1,T3}`.
- **Type-II / κ**: `typeII = (λ/ξ > 1/√2)`, `kappa = λ/ξ`. Shadow-Ring aktiv bei Type-II.

---

4) Komponenten-Diagramm (Mermaid)

```
flowchart LR
  subgraph Edge[Edge / PhyBox]
    HRV[HRV]:::s --> ING
    EDA[EDA]:::s --> ING
    PUP[Pupille]:::s --> ING
    EEG[EEG]:::s --> ING
    ING[Ingest (phybox/ingest.py)] --> SYN[χ, ω, P, τ]
    SYN --> KPI[KPIs (σ, artifacts, drift)]
  end

  subgraph Gov[Governance]
    CG[Consent Hard-Gate] --> DRX[Dual-Receipts]
    DRX --> ENC[AES-GCM AAD]
    ENC --> LED[Ledger]
  end

  Edge -->|WebSocket| UI[UI: GovernanceBar · M13c · Explain]
  KPI --> UI
  Gov --> UI
  LED --> UI

  subgraph Scale[Skalierung]
    FL[Federated Learning] -->|Secure Agg| AGG[Aggregator]
    MPC[MPC Shares] --> AGG
  end

  SYN --> FL
  KPI --> MPC

  classDef s fill:#eef,stroke:#88f
```

---

5) Datenfluss & Governance-Pfad (Mermaid)

```
sequenceDiagram
  participant U as User
  participant UI as UI (GovernanceBar/Explain)
  participant PB as PhyBox (Edge)
  participant GOV as Governance
  participant LED as Ledger

  U->>UI: Consent erteilen
  UI->>GOV: Consent-Token (Frische prüfen)
  GOV-->>UI: GateOpen / GateClosed
  PB->>GOV: Frame(meta)
  alt GateOpen
```

```
    PB-->>UI: m13c {{χ,ω,P,τ}} + kpi
    GOV->>LED: Dual-Receipts, Root-Freeze
    UI->>UI: Explain Overlay (Gate-Beweise, Resonanz, KPIs)
  else GateClosed
    PB-->>UI: kein Stream
  end
```

---

6) Datenmodelle (key-level)

6.1 `metrics_v2` (Frame)

```
{
  "chi": 0.64, "omega": 0.41, "P": 0.58, "tau": 0.33,
  "kpi": {
    "sigma": {"chi": 0.05, "omega": 0.04, "P": 0.06},
    "artifact_rate": 0.08,
    "drift_flags": {"chi": false, "omega": false, "P": true}
  },
  "gate": {
    "geff": 0.66, "threshold": 0.60,
    "dt_window_ms": [18, 40],
    "refractory_ms": 120,
    "flip_allowed": true,
    "phase": "T1"
  },
  "typeII": true, "kappa": 0.98,
  "consent": {"status": "valid", "receipt_id": "CONS-…"},
  "root_hash": "rh:…"
}
```

6.2 `ev_final_run` (Aggregate-Event)

Schema: `schema/events/ev_final_run.schema.json` — **inkl. Gate/Type-II/Commits/Evidence**
Beispiel: `events/samples/ev_final_run.json` (aktualisiert).

---

7) CP-π (Gate-Kontrollpunkte)

- **σ-Stabilität**: σ für χ, ω, P in Range (PASS/HOLD/FAIL).
- **Artefakt-Rate**: ≤ 10 % (Green).
- **Drift**: *keine Doppel-Flags* (gleichzeitig χ & ω).
- **Consent-Frische**: gültig (nicht abgelaufen).
- **Root-Freeze**: Index-Root gematcht.
- **Δt/Refraktär**: Fenster [18,40] ms, τ_ref=120 ms eingehalten.

---

8) Seeds & Cards (Archetypen)

- **signal_map_v1 (Card):** Biosignal → (χ, ω, P, τ)
- **resonance_protocol_v3 (Card):** Resonanzlogik/Chronicles
- **gate_policy_v1 / alert_policy_v1 (Policies)**
- **fl_proto.yaml / mpc_proto.yaml (Skalierungs-Seeds)**
- ***Neu:*** `takt_config.json` (BPM, Bias-Map, θ), `lyra_beat_map.csv` (64 Beats), `metronome_84bpm.wav`

---

9) Ledger (Audit/Auswertung)

- **Root-Hash Freeze** (`index.master.json`) · **Nightly Verify** (`tools/nightly_verify.py`)
- **Receipts:** `receipts/*` (Consent, Ledger, SecAgg)
- **Explain Overlay:** Gate-Beweise (RCC/Maxwell), KPI-Snapshot, **Type-II-Status**
- ***Neu:*** Boundary-Artefakte: `BoundaryPanel_live.json`, `BoundaryPanel_timeseries.csv`, `live_events.jsonl`
  — Rolling-Fenster 60 s, 5-s-Bins; Metriken *false_flips/min*, *refractory_viol/min*; **WS-Stub** `wss://panel.local/boundary`

---

10) DR (Disaster Recovery / Data Retention)

Snapshots (WORM), Rotation (Keys/Receipts), Retention: **aggregate-only**, keine Rohdaten (Privacy by Design).

---

11) UI-Bausteine

GovernanceBar: Consent-Badge, Root-Hash-Chip, KPI-Dropdown
SensorPicker: Sensor-Auswahl + EEG-Sparkline
M13c: χ↔ω Scatter/Summary + Drift-Badges + Explain-Button
KPI-Tiles: Stability (σ), Artifacts, Drift
Explain-Overlay: What/Why/So-What + **SigmaSliceBadge** (PASS $\leq$0.60 · HOLD $\leq$0.75 · FAIL >0.75)
BoundaryPanel (Live): Rolling-Rates, Events-Feed, Δt/τ_ref-Checks

---

12) Glossar

χ, ω, P, τ · **RCC**:EC**, **¬PO**, **$||M||_2=1$**, Maxwell-Boundary, CP-π, Seeds, DR · **Type-II/κ**: κ=λ/ξ, Type-II⇔κ>1/$\sqrt{2}$.
geff: effektives Gate = gate + bias(note); **θ**: Schwelle (=0.60); Δt=[18,40] ms; τ_ref=120 ms.

---

13) Pfad/Artefakte (Keywords)

Mapping `angel/cards/signal_map.yaml` · **Ingest** `phybox/ingest.py` · **KPIs** `phybox/daemon.py` ·
UI `ui/src/components/*`, `types/metrics_v2.ts` · **Governance** `policies/gate_policy_v1.json`, `tools/runpack_gate.py` ·
Audit `tools/ledger_verify.py`, `index.master.json`, `receipts/*` · **Skalierung** `protos/fl_proto.yaml`, `protos/mpc_proto.yaml` ·
Events `schema/events/ev_final_run.schema.json`, `events/samples/ev_final_run.json` ·
Panel `BoundaryPanel_live.json`, `BoundaryPanel_timeseries.csv`, `live_events.jsonl` ·
Takt `takt_config.json`, `lyra_beat_map.csv`, `metronome_84bpm.wav`

---

14) P9-Release-Bundle — EEG-Mapping + KPIs + `ev_final_run` Schema (v1.0)

- **EEG-Term in χ (Edge, lokal)** inkl. Mapping & Tests
- **Bio-KPIs** (σ-Rolling, artifact_rate, drift_flags) mit Daemon-Verdrahtung
- ``** (aggregate-only)** + Beispiel
- **UI-Hooks** (Tiles/Overlay) + Smoke-Gate + Runbook
- **Compliance**: Aggregatwerte potentiell personenbezogen → ephemere IDs, kurze Retention, AAD-Bindung

---

15) Master-Katalog Δ-Protokoll (Auszug, v1.3/NSR 1.3.NSR)

- §1 Gate/Leakage — π-Gate, RCC-Guards, 1-Edge-Policy ✅📶
- §3 Doublet-Flipper — Δ-Fenster, Refraktär, T1/T3-Flags ✅☸️
- §14 Explain-Overlay — Gate-Beweise, KPI-Snapshot ✅☸️
- §16 Maxwell & Chaos-Pendulum — L*-Schranken ✅📶
- §18 Zettelkasten-Brücke — DF/Zeta in Wissensgraph ✅♾️

---

16) Urteil & Nächste Schritte

Urteil: kohärent, leakage-frei, meta-backprop-kompatibel.
Next-3: (1) T1.1.2 Lyra Linearity-Cal · (2) Panel Ring-Buffer + Gate-Loudness · (3) Evidence-IDs für Spell-125 (Primärhandschriften).