Hour 2 – High-Level Architecture Exploration

Goal: Map Hour-1 requirements into a reference architecture with component boundaries, data flows, performance budgets, and a phased delivery plan.

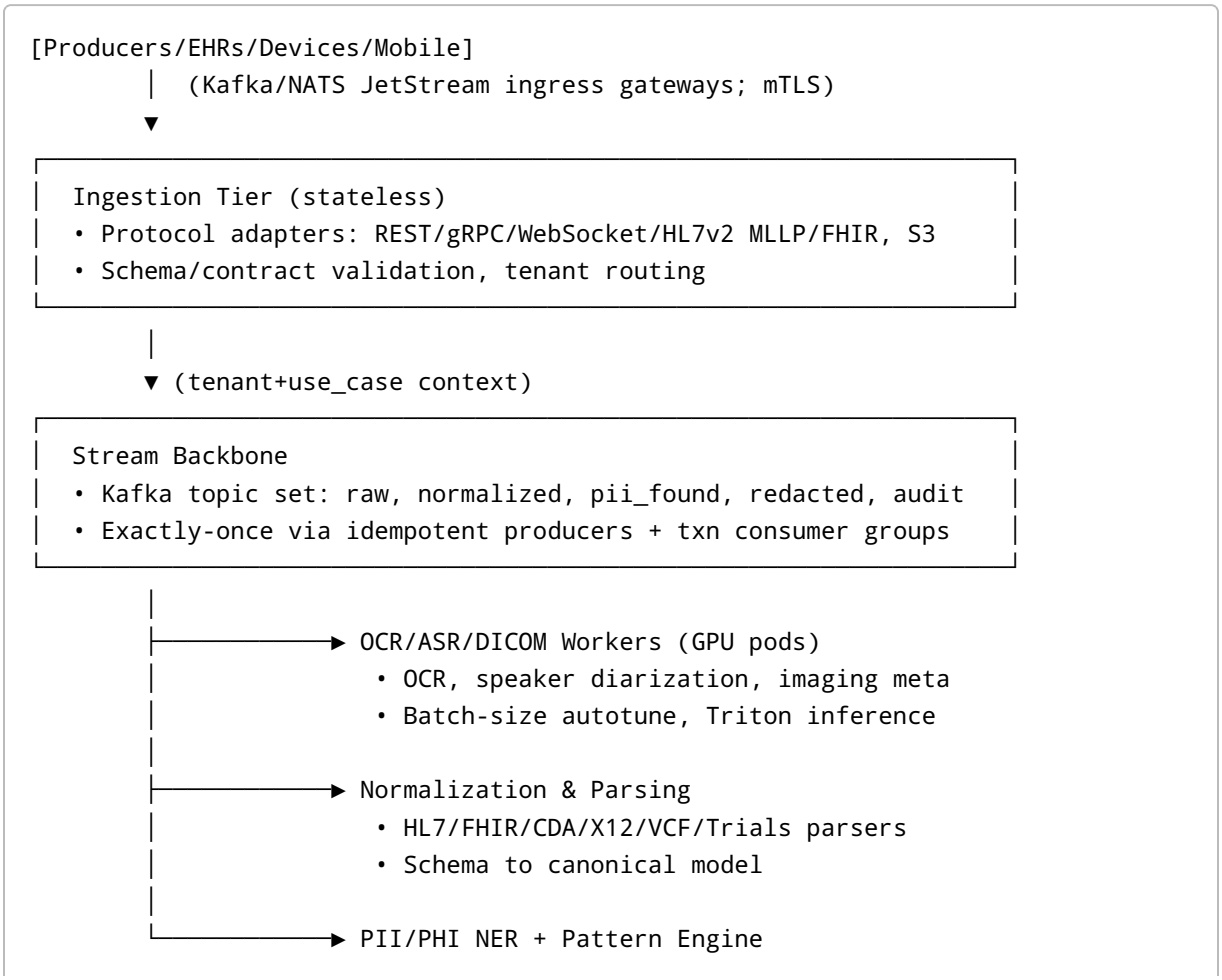
---

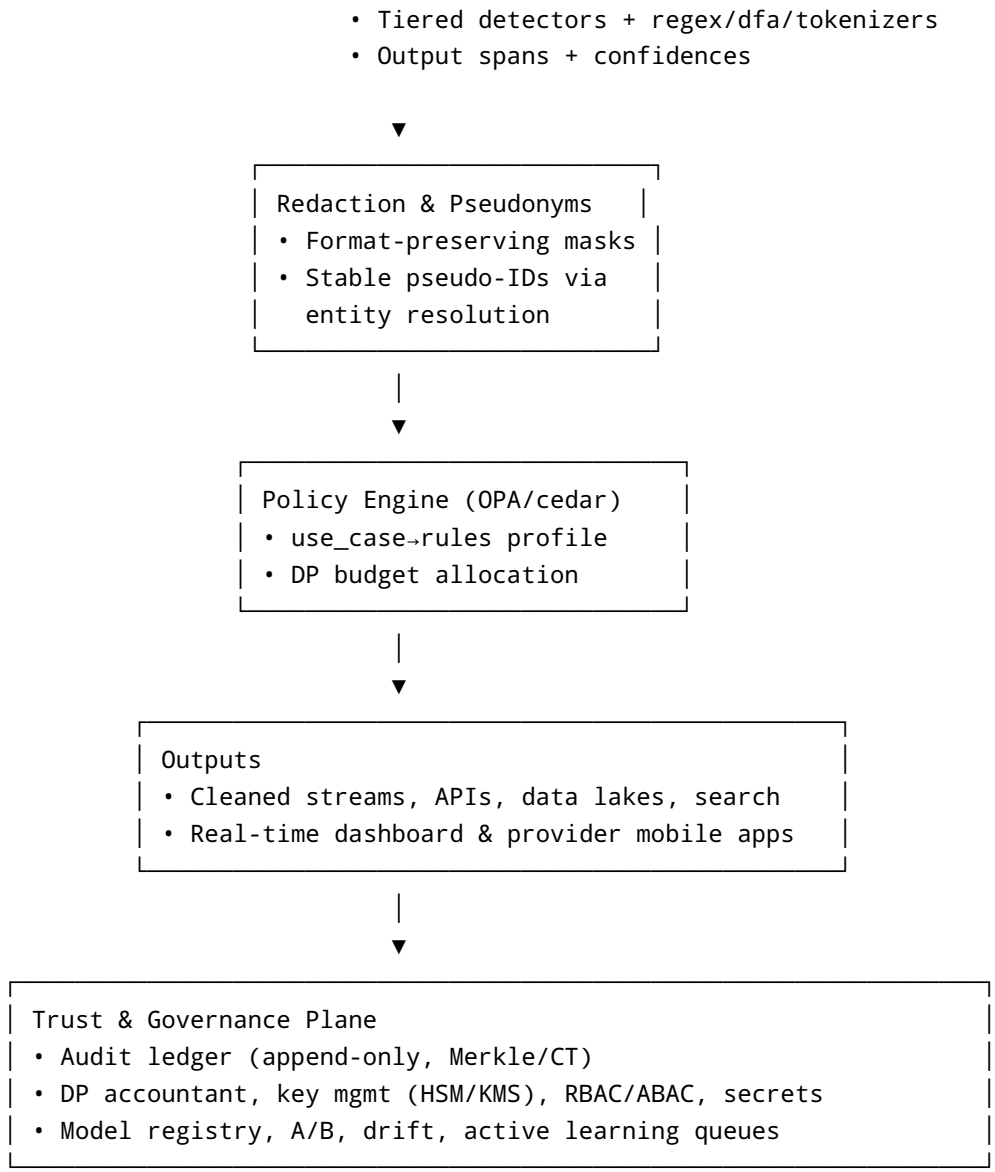
0) Architecture Guardrails

- **Privacy-first by design**: default-deny, minimum-necessary, RBAC/ABAC at every hop.
- **Provable guarantees**: (ε, δ) DP accounting, cryptographic audit proofs.
- **Horizontal everything**: stateless compute, sharded state, elastic queues.
- **Multi-tenant isolation**: hard isolation at data plane; soft isolation at control plane.
- **Deterministic latency**: per-stage SLOs to meet <100 ms p95 end-to-end.
- **Extensibility**: plugin model for modalities (parsers, detectors, redact rules).

---

1) Reference Architecture (Bird's-eye)





---

2) Component Breakdown

2.1 Ingestion Tier

- **Adapters**: REST, GraphQL, gRPC, WebSocket, HL7v2 (MLLP), FHIR (REST), S3/GCS dropboxes, SFTP.
- **Responsibilities**: contract validation (OpenAPI/Protobuf), tenant tagging, schema stamping, back-pressure.
- **Security**: mTLS, JWT/OIDC, per-tenant API keys, request-level `use_case` required.

2.2 Stream Backbone

- **Kafka/NATS** with topic families per tenant & modality. Retention tuned by use case. Exactly-once via idempotent producers and transactional consumers.
- **Schema registry** (Avro/Protobuf/JSON-Schema) for canonical events.

2.3 Modality Processing Workers

- **OCR**: GPU pods (Triton) + layout reconstruction; PDF images → text spans.
- **Audio**: streaming ASR, speaker diarization (VAD + x-vector/ECAPA), timestamps.
- **Imaging (DICOM)**: metadata extraction, PHI tag scrubbing, pixel-level overlays detection.
- **Structured**: HL7/FHIR/CDA/X12/VCF/Trials → canonical healthcare model.

2.4 PII/PHI Detection Layer

- **Tiered detectors**: hybrid NER (transformers) + deterministic patterns (DFAs/regex) + checksum/validators.
- **Targets**: Tier-1 (names, SSN/equivalent, MRN, phone, email) recall $\geq$99.5%; Tier-4 (quasi-identifiers) $\geq$90%.
- **Calibration**: per-tier thresholds; near-miss JSON emission for failures.

2.5 Redaction & Pseudonymization

- **Format-preserving**: masks maintain glyph/shape (e.g., phone, dates, IDs). Locale aware.
- **Stable pseudonyms**: deterministic salted hashing + entity graph IDs; cross-modality & family relations.
- **Reversibility**: escrowed mapping in HSM with time-boxed access policies.

2.6 Policy Engine

- **Inputs**: `tenant`, `use_case`, `role`, `purpose`, DP budget state.
- **Outputs**: redaction recipe, fields to preserve, DP ε allocation, retention TTL, export permissions.
- **Tech**: OPA/Rego or Cedar policies; hot-reloadable bundles per tenant.

2.7 Trust & Governance Plane

- **Audit ledger**: append-only store; each decision/event hashed into a Merkle tree; periodic anchor to external timestamping service.
- **DP Accountant**: per-identity and per-dataset budget tracking; composition rules; alerts when budgets near thresholds.
- **Key Management**: KMS/HSM envelopes; multi-party approval for de-pseudonymization.

2.8 Serving & Interfaces

- **APIs**: REST/GraphQL for CRUD & queries, gRPC for high-throughput ingestion, WebSocket for realtime.
- **Sinks**: Cleaned topics, parquet/Delta/Iceberg lakes, vector search (PHI-free), analytics warehouse (DP-sanitized).
- **UX**: Real-time ops dashboard (SLOs, lag, budgets) + provider mobile apps (FHIR-aware views).

2.9 ML Platform

- **Model zoo**: multimodal transformer(s) + specialized heads (NER, diarization, OCR-LM fusion).
- **Training**: FL coordinator (cross-site), DP-SGD, adversarial training, evaluation harness.
- **Inference**: Triton server, dynamic batching, rate/latency autotune.
- **MLOps**: model registry, shadow/A-B, drift detection, active learning loop with PII-safe annotation UI.

---

3) End-to-End Data Flows (Happy Paths)

3.1 Text/PDF with PHI

1) Ingest via REST → `raw.text` topic. 2) OCR (if needed) → spans & layout → `normalized.text`. 3) NER+patterns → entity spans → `pii_found.text`. 4) Policy engine applies recipe (based on `use_case`) → redacted doc + pseudo-IDs → `redacted.text`. 5) Audit entries written (hash of input, spans, policy hash, output hash) → ledger. 6) Downstream sink: search index (PHI-free), lake storage, API response.

3.2 Audio Stream

1) gRPC streaming → `raw.audio`. 2) VAD/ASR/diarization (streaming) → `normalized.audio` with speaker turns. 3) NER on transcripts + time codes → `pii_found.audio`. 4) Redaction: token-level bleeping + transcript masking; speaker pseudo-IDs. 5) Emit audit & sanitized stream to WebSocket subscribers.

3.3 DICOM

1) S3 dropbox watch → `raw.dicom`. 2) Metadata + pixel PHI detection → overlays removed. 3) Policy + redaction of headers → `redacted.dicom`. 4) Immutable audit + registry update.

---

4) Performance & Latency Budget (p95 targets)

- Ingress + auth: **5 ms**
- Queueing (steady-state): **<5 ms** (multi-partition, warm consumers)
- OCR/ASR/DICOM (if applicable): **30 ms** (GPU, batch 4–8, quantized models)
- NER + patterns: **20 ms** (fused kernels; ONNX/TensorRT)
- Policy evaluation: **2 ms**
- Redaction & serialization: **10 ms**
- DP accounting + audit write (async): **≤8 ms** visible; full commit async within 50 ms
- Egress (API/stream write): **5 ms Total:** ~**77–85 ms** p95 (headroom to 100 ms). p99 target: <200 ms.

Throughput shaping: - Kafka partitions sized for ≥ 2× peak (e.g., 400 partitions for 100k docs/s with 250 msg/s/partition). - Autoscaling on lag + GPU/CPU utilization; predictive scaling from workload calendar.

---

5) Storage & Indexing

- **Hot**: Redis/RocksDB state for stream processors; feature cache.
- **Warm**: Columnar lake (Delta/Iceberg) in object storage; compaction jobs.
- **Cold**: Glacier-class archives with 11-nines durability.
- **Search**: OpenSearch/Solr for PHI-free content & audit queries.
- **Ledger**: Append-only (immudb/Trillian/Tendermint) with Merkle proofs.

---

6) Security Model

- Network: zero-trust, mTLS, SPIFFE IDs.
- AuthN: OIDC, short-lived tokens, mutual TLS for system accounts.
- AuthZ: RBAC + ABAC (OPA/Cedar); purpose binding via `use_case`.
- Secrets: KMS/HSM; envelope encryption; per-tenant keys.
- Data: FPE/FHE where required; all at-rest encrypted.
- Hardening: CIS baselines, SAST/DAST, supply-chain attestation (SLSA), SBOMs.

---

7) Differential Privacy & PPML

- **DP Profiles**: strict (ε=1), balanced (ε=4), analytics (ε=8); δ set per dataset size.
- **Accounting**: per-query/user/object via moments accountant; composition tracked; deny when budget exhausted.
- **FL**: cross-site training with secure aggregation; client drift health checks.
- **HE/SMPC**: scope to high-value inference endpoints with small tensors.

---

8) Observability & Operations

- **Golden signals**: latency, throughput, error rate, saturation, queue lag.
- **SLOs & Alerts**: p95<100 ms, p99<200 ms, audit commit <60 ms, DP budget <10% remaining.
- **Near-miss analytics**: auto-sample false negatives/positives to labeling queue.
- **Chaos & DR**: node kill drills; RTO <30 s, RPO $\leq$ 1 message (transactional).

---

9) Multi-Tenancy & Data Boundaries

- Separate Kafka namespaces/buckets per tenant; per-tenant KMS keys.
- Policy bundles per tenant; rate limits/quotas.
- Option for **dedicated VPC** tenancy for high-sensitivity customers.

---

10) API Surface (Sketch)

- **Ingestion**: `POST /v1/documents?use_case=...`, `grpc Ingest(stream Chunk)`.
- **Realtime**: `wss /v1/streams/{tenant}/{topic}`.
- **Redaction on demand**: `POST /v1/redact` (sync <100 ms SLA for small docs).
- **Search**: `POST /v1/query` (PHI-free).
- **Audit**: `GET /v1/audit/{hash}` → inclusion proof; `POST /v1/audit/query`.
- **DP**: `GET /v1/dp/budgets`, `POST /v1/dp/spend` (idempotent tokens).

---

11) Capacity Planning (Worked Back-of-Envelope)

- **Docs**: 100k docs/s, avg 8 KB payload → 800 MB/s ingress.
- **Kafka**: 400 partitions × 2 MB/s each → 800 MB/s headroom; 3× replication → 2.4 GB/s broker IO.
- **Inference**: NER @ 50k docs/s/GPU with quantized small-LM → ~2 GPUs per 100k docs/s (plus OCR/ASR pool).
- **Storage**: 10 PB/year (raw+redacted+audit) with compaction.

---

12) Risks & Mitigations

- **R1: OCR/ASR latency spikes** → pre-warm GPU pools; admission control; degrade to async for large PDFs.
- **R2: Tier-1 recall shortfall** → cascade detectors; human-in-the-loop review for critical feeds.
- **R3: Policy drift** → policy versioning & canaries; policy unit tests per tenant.
- **R4: DP utility loss** → per-use-case ε tuning; privacy amplification via subsampling.
- **R5: Ledger bottlenecks** → batch tree construction; anchor digests periodically.

---

13) Phased Delivery Roadmap

Phase 0 (2–3 wks) – Skeleton: Kafka, minimal APIs, basic policy engine, stub audit. **Phase 1 (4–6 wks)** – Text/PDF path: OCR, Tier-1 NER, redaction, audit proofs, dashboard v1. **Phase 2 (6–8 wks)** – Audio streaming path + diarization; mobile viewer; DP accountant MVP. **Phase 3 (8–10 wks)** – DICOM path; entity resolution graph; stable pseudonyms. **Phase 4 (ongoing)** – FL/PPML hardening, A/B testing, drift/active learning at scale.

---

14) Open Questions for Stakeholders

1) **Regulatory scope by region** (EU/US/India): differing retention & breach reporting. 2) **MPI availability & trust**: can we consume a hospital's MPI as a hint? 3) **ε/δ defaults per use case**: do we ship opinionated presets? 4) **Ledger anchoring**: internal vs public timestamping cadence & cost sensitivity. 5) **On-prem vs SaaS split**: VPC-hosted options for top-tier clients?

15) Outcome of Hour 2

A concrete, privacy-first reference architecture with latency/throughput budgets, component boundaries, and a pragmatic delivery path aligned to Hour-1 requirements.