AITHOR – C4 Architecture Model

# 1. Executive Overview

## 1.1 Roles and Usage Patterns

* JC Admin – platform governance + full project control (companies, JC/client users, RBAC matrix, subscriptions/billing/manual payments, security/incident monitoring, tickets, pending requests, CMS/library/guidance, troubleshooting).
* JC PM – project execution + publish no platform governance no user invitations/seat allocation.
* Client Admin – client workspace owner invites and manages up to 5 Editors + 5 Viewers allocates seats and enforces client portal access controls manages client-level users and roles.
* Client Editor (5 seats) – upload, create/edit projects/folders/file, analytics, dashboards/reports, AI usage, publish where allowed no user admin.
* Client Viewer (5 seats) – view projects/folders/files and download where allowed view dashboards/reports AI read-only where applicable.

## 1.2 Key non-functional constraints

* On‑prem Keycloak integration for authentication and MFA/OTP (Keycloak setup out of scope).
* Access control + seat management requirements in client portal (SOW) including IP restrictions and concurrent login restrictions.
* Standardized errors, retries, DLQ handling and redrive/redraft semantics for long-running operations.
* Operational dashboards: queue errors, job failures, prompt failures, uploads in progress, AI activity logs.

# 2. C4 Level 1 – System Context

This diagram shows AITHOR boundary, users, and external dependencies.

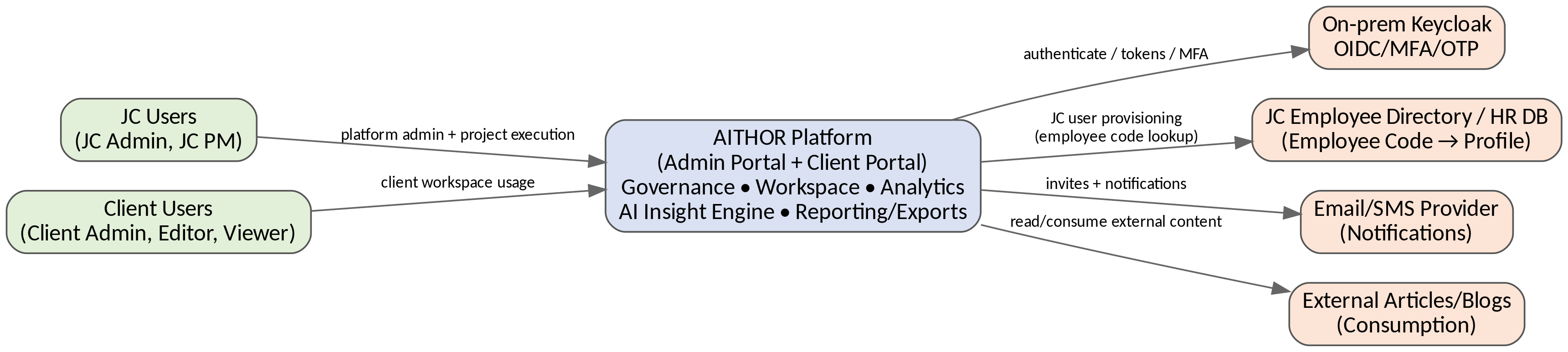


Figure 2-1. System Context (C4 Level 1)

## 2.1 External Dependencies

|  |  |  |
| --- | --- | --- |
| **Dependency** | **Purpose** | **Notes** |
| Keycloak (on‑prem) | OIDC auth, MFA/OTP, token issuance, password reset flows | Integration only AITHOR does not implement its own credential store. |
| JC Employee Directory / HR DB | JC user provisioning via employee code lookup | MS2 fetches profile attributes on JC user creation/invite. |
| Email/SMS Provider | Invitations, notifications, alerts | Adapters in MS5 in-app notification always available. |
| External Articles/Blogs | User consumption/references | Role-gated access AITHOR does not store external content. |

# 3. C4 Level 2 – Container Architecture

This diagram includes all deployable units and all data stores described in the Data Model and design pack.

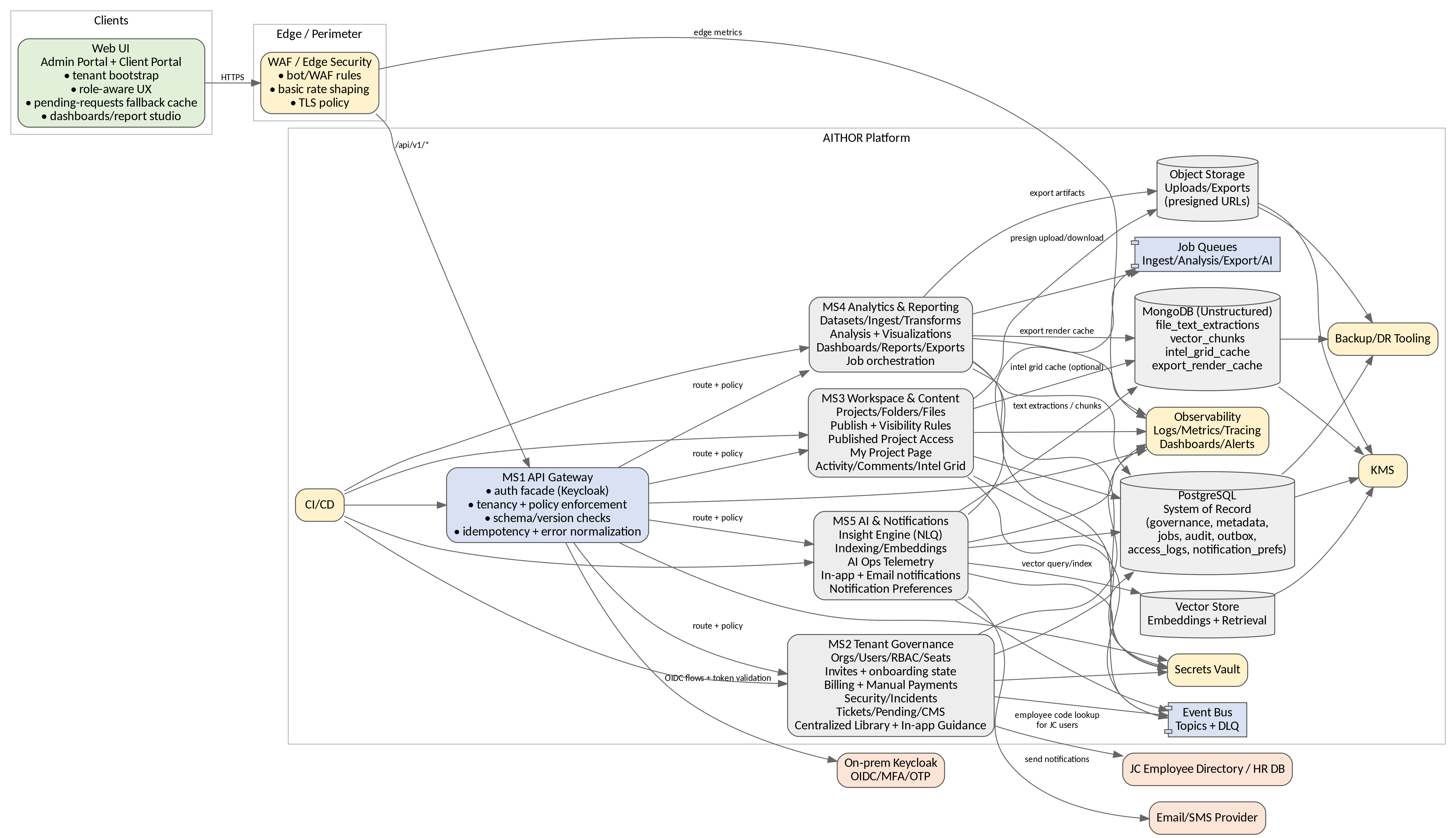


Figure 3-1. Container Diagram (C4 Level 2)

## 3.1 Container Responsibilities (Strict)

|  |  |  |
| --- | --- | --- |
| **Container** | **Responsibilities** | **Key Data/Integrations** |
| UI (Admin Portal + Client Portal) | Tenant bootstrap, invite onboarding UI, pending-requests fallback cache view + refresh, role-aware navigation, dashboards/report studio UI, file upload via presigned URLs | Calls MS1 direct upload/download to object storage. |
| WAF/Edge | Perimeter protection, baseline rate shaping, TLS policy enforcement | Feeds edge metrics to observability. |
| MS1 API Gateway | Auth facade, session controls (expiry/forced logout), schema validation, versioning, tenancy guard, RBAC/entitlements/seats enforcement, IP/concurrency enforcement, idempotency, standardized errors, access logging | Keycloak PG access\_logs routes to MS2–MS5. |
| MS2 Tenant Governance | Companies/orgs, user lifecycle, invites (resend/expire), RBAC matrix, seat allocation, subscriptions/billing/manual payments, security incidents, tickets/pending, CMS + in-app guidance + centralized library, troubleshooting tooling | PG (system of record) event bus via outbox HR/employee directory integration. |
| MS3 Workspace & Content | Projects/folders/files, upload lifecycle, publish + visibility rules, published project access, My Project Page (default folders + cards), activity/comments, Intel Grid | PG object storage Mongo intel grid cache event bus via outbox. |
| MS4 Analytics & Reporting | Dataset ingest/clean/transform, analysis, visualization, dashboards, reports, exports, job orchestration, export render caching | PG object storage Mongo export cache queue event bus. |
| MS5 AI & Notifications | AI ingest/index/query, extractions and chunks, embeddings, NLQ/insight generation, prompt-failure telemetry, in-app/email notifications, notification preferences | PG Mongo unstructured vector store queue event bus email/SMS. |

# 4. C4 Level 3 – Component Architecture

## 4.1 UI (Admin Portal + Client Portal)

* Bootstrap: load /me and /session/context enforce org/project selection fallback dropdown if context missing.
* Invitation onboarding: open invite link, validate token, create password handle invite expiry and resend prompts.
* Session behavior: handle session expiry and forced logout show secure re-auth experience.
* Pending Requests UX: show latest list if fetch fails, show last cached snapshot with “last updated” auto-refresh at intervals (e.g., 5 minutes) per sequence flows.
* My Project Page: tiles → project page → cards routing (Report Studio/Dataset Viewer/Media Viewer/internal tools) + default folders (Deliverables, Data Analysis, Media, Visualization, Appendix) + recent activity/comments.
* Role-aware modules: JC Admin governance modules JC PM project execution modules client admin/editor/viewer capability gating.
* In-app guidance + centralized library: render help content, guides, reusable assets role-gated to ensure correct visibility.

## 4.2 MS1 – API Gateway (Components)

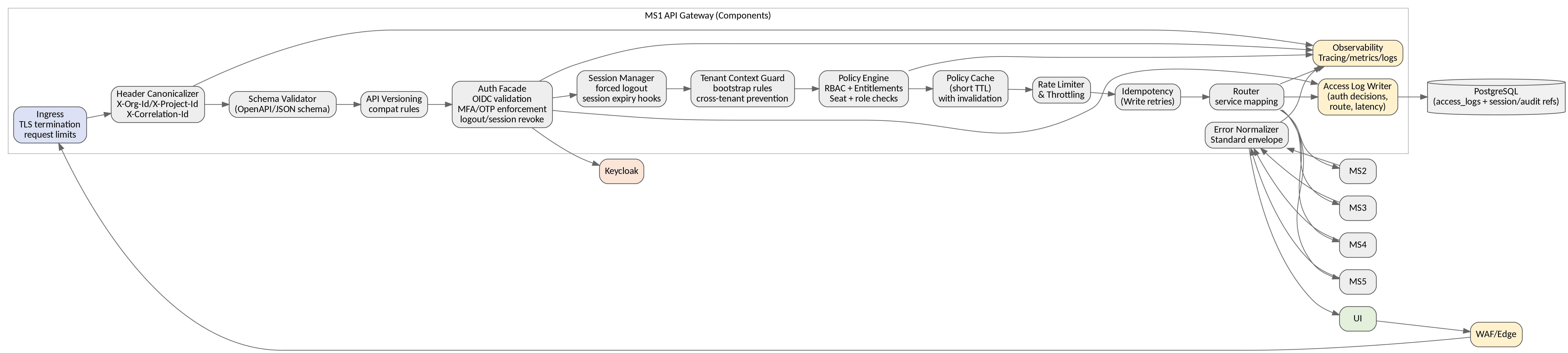


Figure 4-1. MS1 Components

* Schema validation: validate inbound payloads against OpenAPI/JSON schemas reject invalid requests with standardized errors.
* Versioning: enforce /api/v1 contract rules provide compatibility guardrails for future versions.
* Session controls: integrate logout/session revoke support forced logout and session expiry handling signals to UI.
* Authorization: RBAC + subscription entitlements + seat rules policy caching with invalidation from governance changes.
* Security controls: IP restrictions and concurrent session policies (configured via MS2) enforced at gateway.
* Access logging: write access logs (route, decision, latency, correlationId) to PG and observability stack.

## 4.3 MS2 – Tenant Governance (Components)

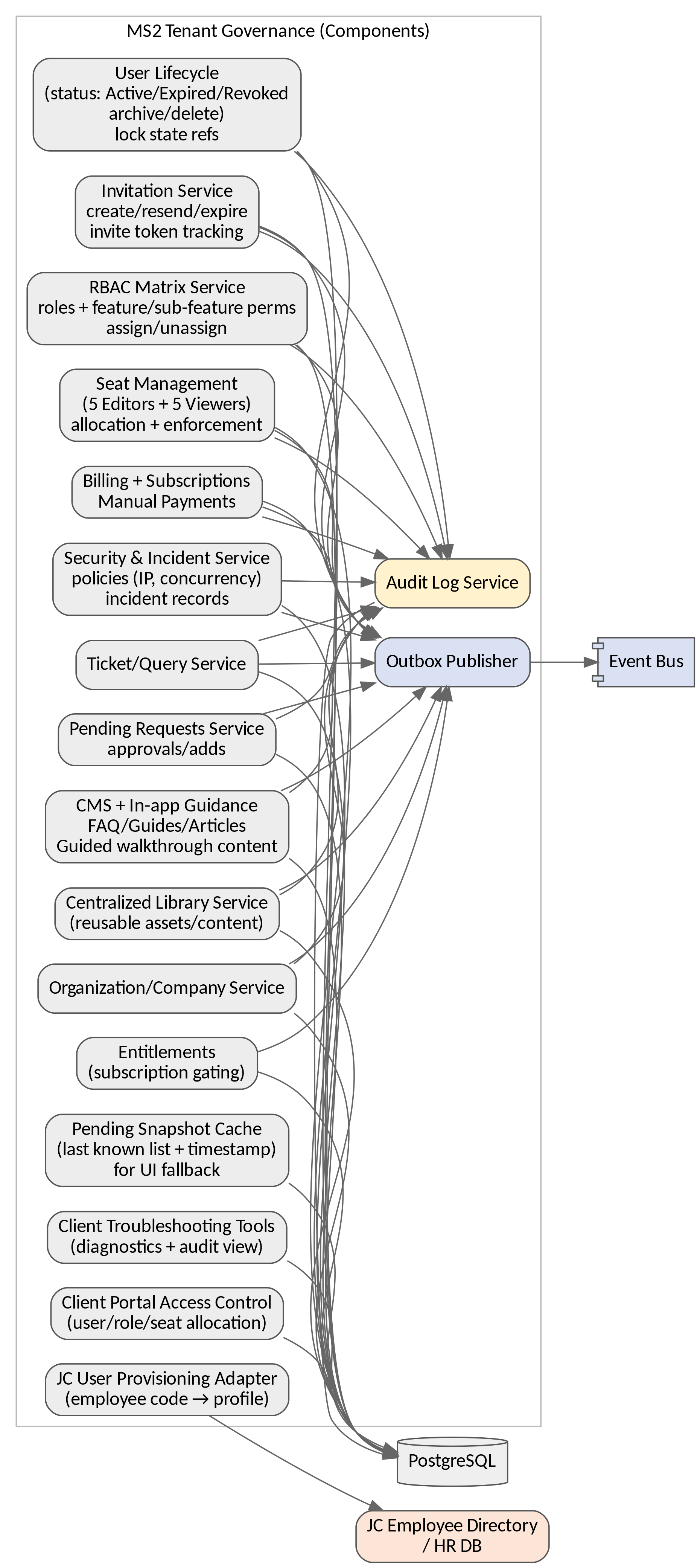


Figure 4-2. MS2 Components

* Invite lifecycle: create/resend/expire invites track onboarding state audit all changes.
* JC user provisioning: accept employee code fetch remaining profile attributes from employee directory/HR DB.
* User lifecycle: active/expired/revoked archive/delete bulk operations audit trail.
* RBAC matrix: roles + feature/sub-feature permissions assign users edit/delete roles admin-driven.
* Seat management: enforce client admin caps (5 editors + 5 viewers) and seat allocation policies.
* Billing: subscriptions and manual payment tracking entitlement changes propagate to MS1 enforcement.
* Pending requests: maintain current list and persist snapshot cache for UI fallback schedule refresh/retry.
* CMS + Guidance + Library: manage FAQ/Guides/Articles, in-app walkthrough content, centralized library assets.
* Security & incidents + troubleshooting: policy management, incident records, support diagnostics (audit-driven).

## 4.4 MS3 – Workspace & Content (Components)

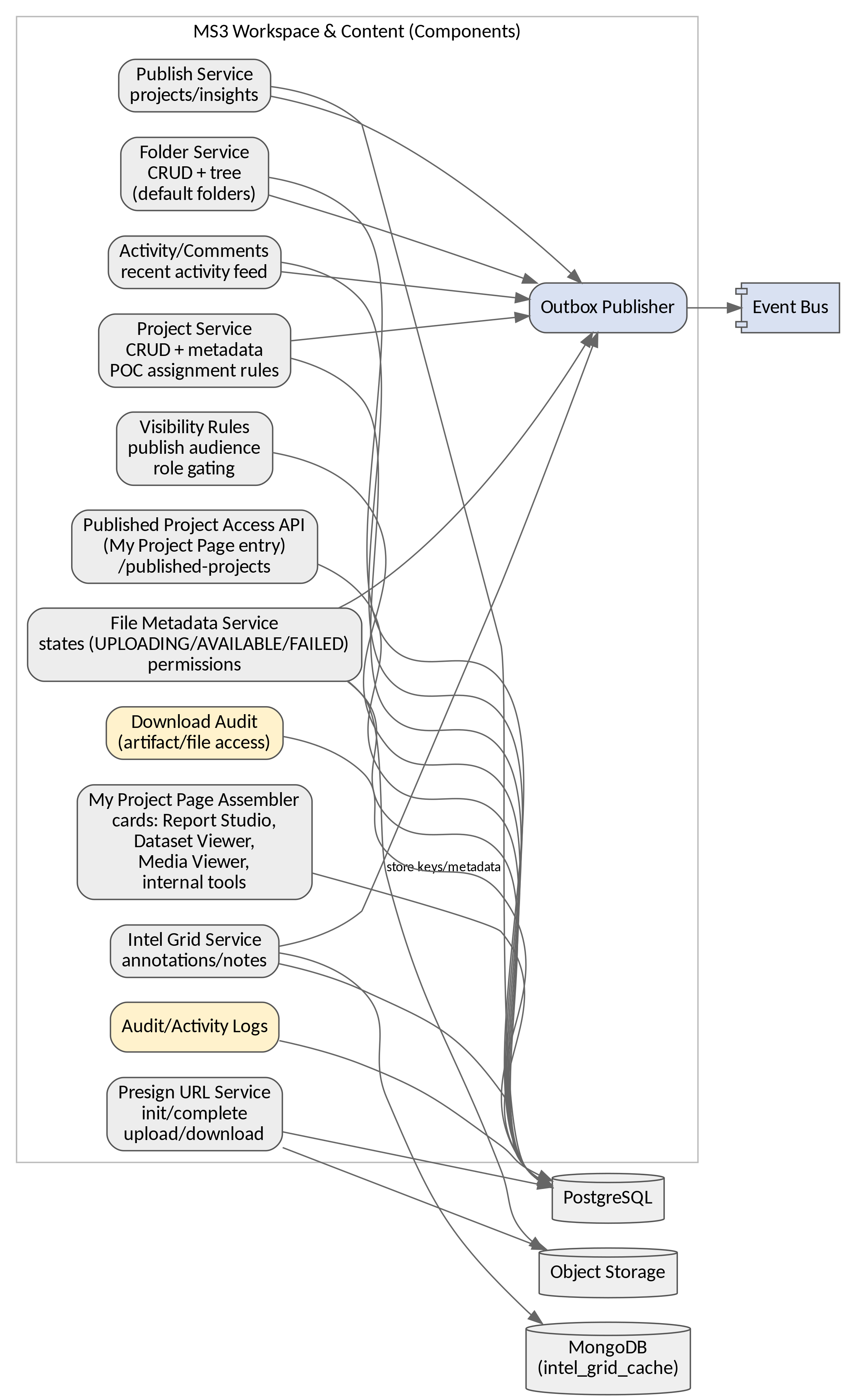


Figure 4-3. MS3 Components

* Project lifecycle: create/edit/view/delete + metadata updates apply POC auto-population rules where defined audit changes.
* Publish + visibility rules: publish projects/insights and calculate who can see them expose Published Project Access APIs.
* Folder + default folders: CRUD and tree create default folder structure where required role-gate deletion.
* File lifecycle: upload init/complete states UPLOADING/AVAILABLE/FAILED presigned URL issuance download audit logging.
* My Project Page assembler: cards to Report Studio/Dataset Viewer/Media Viewer/internal tools + recent activity/comments.
* Intel Grid + collaboration: annotations/notes cache as per data model audit events via outbox.

## 4.5 MS4 – Analytics & Reporting (Components)

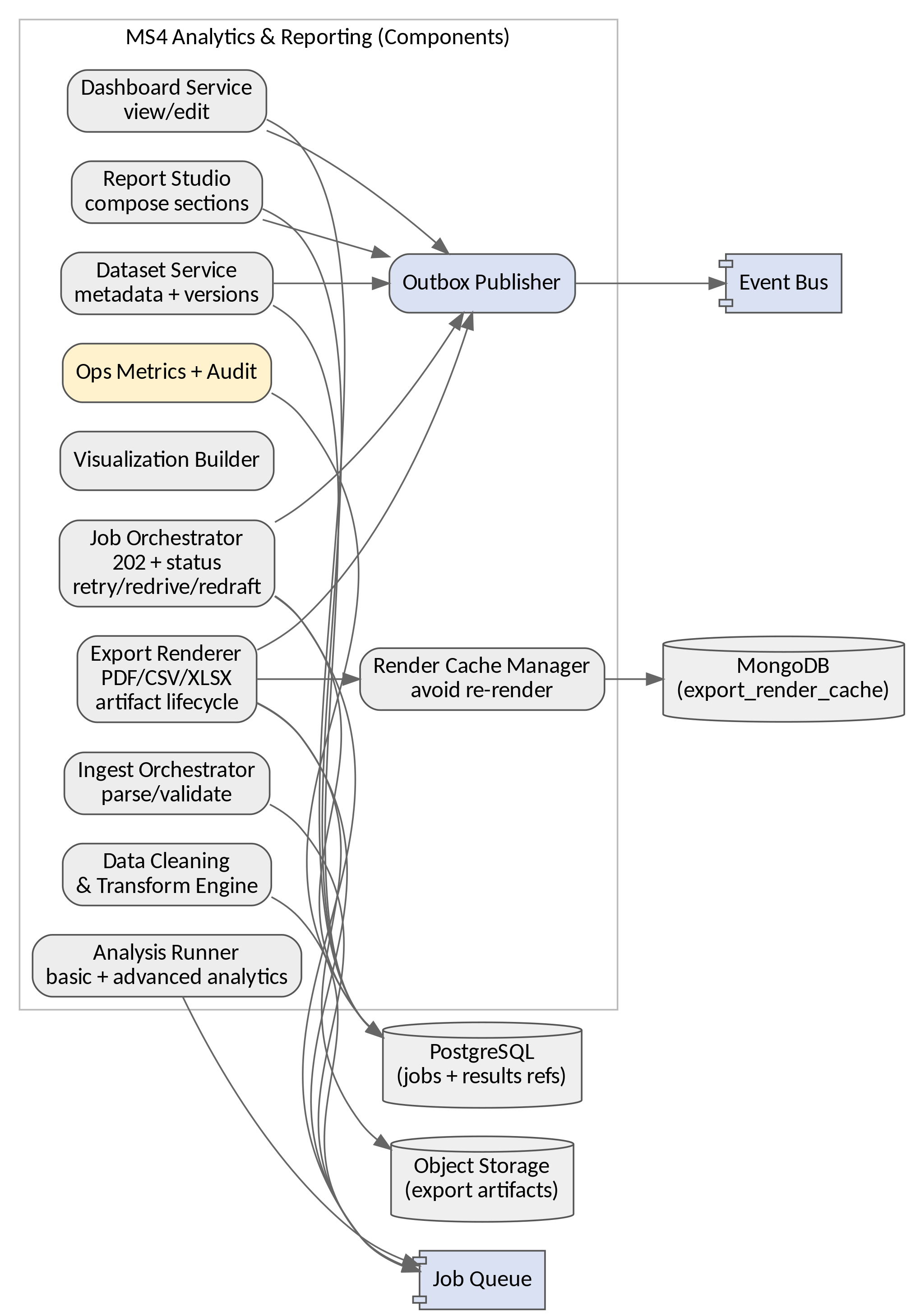


Figure 4-4. MS4 Components

* Ingest + transforms: parse/validate datasets data cleaning/transformation persist results and job state.
* Analysis + visualization: execute analytics and create visualizations integrate into dashboards and reports.
* Dashboards/Reports: view/edit depending on role rules store layouts and report composition.
* Exports: render artifacts (PDF/CSV/XLSX, etc. per API contract) store in object storage maintain retention metadata audit downloads.
* Render caching: use Mongo export\_render\_cache to avoid unnecessary re-render where applicable.
* Job orchestration: return 202 + jobId track status support retries/redrive/redraft as per error strategy.

## 4.6 MS5 – AI & Notifications (Components)

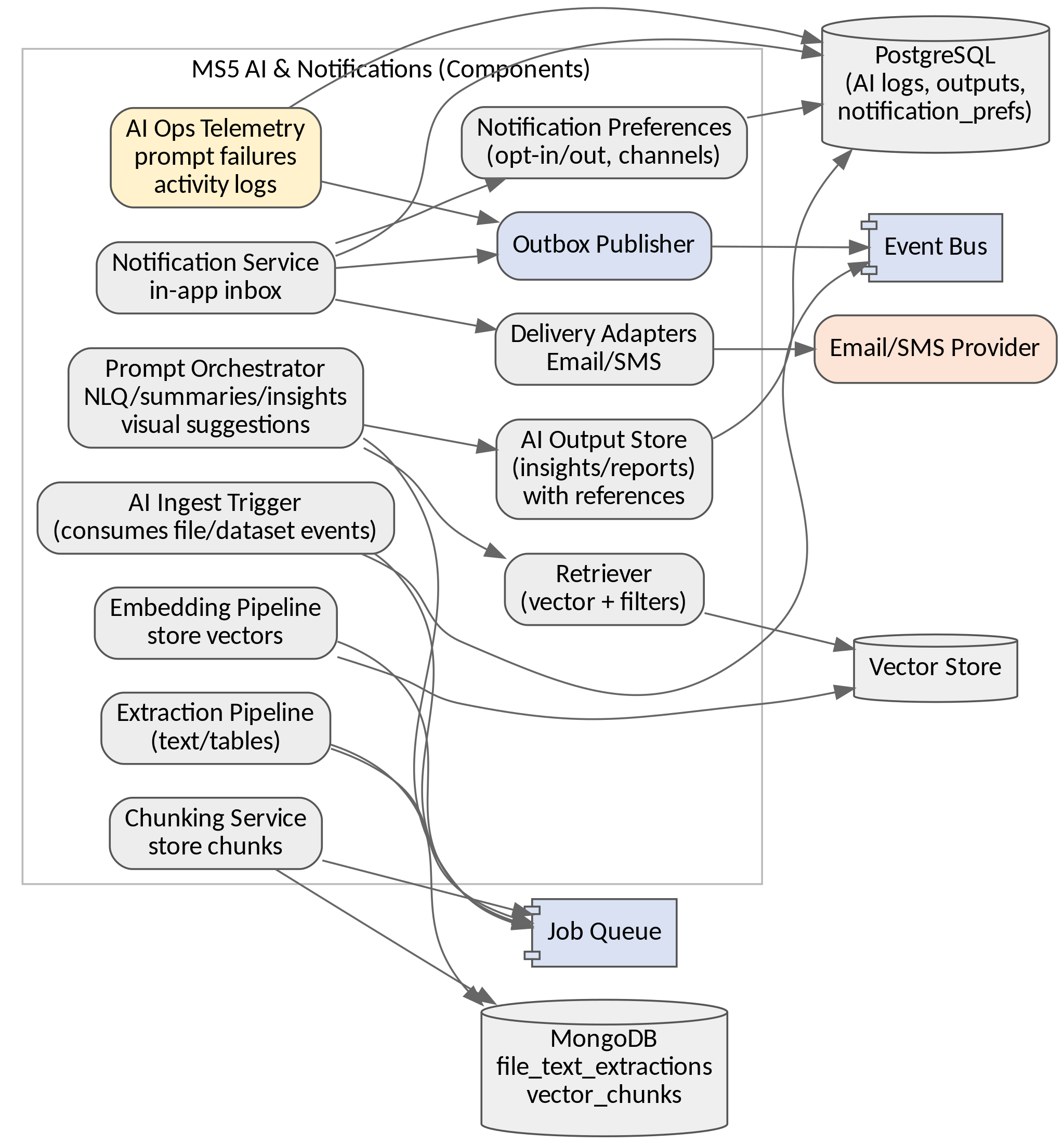


Figure 4-5. MS5 Components

* AI ingest/index: consume file/dataset readiness events extract text chunk embed store vectors persist unstructured extracts/chunks in Mongo.
* Insight Engine: NLQ, summaries, insights, reports, visual suggestions persist outputs with references and operational metadata.
* AI ops telemetry: prompt failures + AI activity logs exposed to operational dashboards aligned with error-handling and event model.
* Notifications: in-app notification inbox + email/SMS delivery store notification preferences support resend/failed delivery handling with DLQ.

# 5. Cross-Cutting Architecture (Strict)

## 5.1 Identity Lifecycle (Invite, Password, Sessions, Audit)

* Invite-based onboarding for non-JC Admin: invite token validation, password creation, invite expiry, resend workflows.
* Password reset/forgot-password flows are executed via Keycloak UI links to Keycloak recovery platform records user status and audit events.
* Session lifecycle: expiry and forced logout supported gateway and UI handle re-auth logout events are audited.
* Login/logout auditing and security telemetry are stored in access/audit logs for support and incident monitoring.
* Lock rules: 3 failed attempts temporary lock (3 minutes), 5 failed permanent lock for non-JC Admin JC Admin exempt (as per design pack).

## 5.2 Authorization and Subscription Enforcement

* RBAC matrix (feature/sub-feature perms) in MS2 is the policy source-of-truth MS1 enforces at runtime.
* Subscription entitlements and seat allocation control availability denials return standardized ENTITLEMENT\_DENIED/PERMISSION\_DENIED errors.
* Client admin seat caps enforced at invite/activation and runtime for concurrent login policies.

## 5.3 Data Stores and Consistency

* PostgreSQL is the transactional system-of-record: governance, workspace metadata, jobs, audit logs, outbox, access logs, notification preferences.
* MongoDB stores unstructured caches and AI artifacts: file\_text\_extractions, vector\_chunks, intel\_grid\_cache, export\_render\_cache.
* Object storage stores binaries (uploads and exports) accessed via presigned URLs metadata references stored in PG.
* Vector store holds embeddings for semantic retrieval MS5 is owner of embedding/query flows.

## 5.4 Events, DLQ, and Redrive/Redraft

* Outbox pattern publishes domain events from MS2–MS5 into the event bus with tenant context.
* DLQ captures poison messages redrive is controlled redraft creates new artifacts/jobs with new IDs for traceability.
* Operational dashboards surface queue errors, job failures, and prompt failures to enable support workflows.

## 5.5 Error Handling Standard

* All services return standardized error envelopes gateway normalizes dependency errors.
* Idempotency keys required for retryable writes async jobs return 202 + jobId and deterministic job states.

# 6. Deployment, Operations, and NFRs

## 6.1 Monitoring & Alerting

* Central dashboards for: queue depth/errors, job failures, prompt failures, upload failures/in-progress, security alerts, availability/latency SLOs.
* Alerting policies for: elevated 5xx rates, Keycloak unavailability, broker/queue DLQ growth, DB saturation, object-store failures.

## 6.2 Patch, Upgrade, and Release Governance

* CI/CD pipelines enforce build/test/security scans and contract tests releases promoted via environment gates.
* Patch & upgrade strategy: rolling updates for stateless services database migration discipline rollback plans.

## 6.3 Backup/DR

* Backups for PG/Mongo and object storage restore drills and documented runbooks.
* RPO/RTO targets to be finalized with stakeholders.