TechNova Solutions Pvt. Ltd.

Internal Projects Documentation

Prepared for: Internal Engineering & Product Teams
Prepared by: TechNova - Engineering Documentation Team

Date: November 23, 2025

Table of Contents

1. Project Atlas – Unified Employee Management System

Overview

Project Atlas is TechNova's centralized human capital management platform designed to consolidate employee data, automate HR workflows, and provide real-time insights into workforce operations. Atlas aims to replace disparate spreadsheets and legacy systems with a single source of truth for employee records, payroll, attendance, performance reviews, and compliance reporting.

Key Features

• Employee master data (personal info, employment history, role & team mappings).

• Time & attendance (biometric integration, shift management, over-time calculations).

• Payroll & benefits (salary components, taxes, statutory deductions, payslip generation).

• Performance management (OKRs, goal tracking, 360-degree feedback workflow).

• Leave & approvals (multi-level approval flows, holiday calendars, accrual policies).

• Reporting & dashboards (turnover analytics, headcount trends, payroll spend).

Architecture & Data Flow

Atlas is implemented as a modular microservice system. The core services include: Identity Service, Payroll Service, Attendance Service, Performance Service, and Reporting Service. Services communicate via RESTful APIs and an internal message bus (Kafka) for asynchronous events such as 'attendance_processed' and 'payslip_generated'. PostgreSQL serves as the primary relational store for structured HR data, while Redis is used for caching frequently accessed lookup tables and session data.

Tech Stack & Integrations

Backend: Java Spring Boot microservices.

Frontend: React + TypeScript with a component library for consistent UI.

Database: PostgreSQL for transactional data and TimescaleDB extension for time-series attendance analytics.

Messaging: Kafka for event-driven flows.

Auth: OAuth2 / Keycloak for SSO and fine-grained role-based access control.

Third-party: Biometric vendor SDKs, payroll compliance APIs, and an HRIS import/export connector.

Security & Compliance

Atlas enforces least-privilege access, row-level security for sensitive data, and strong encryption-at-rest and in-transit. Payroll data access is restricted to authorized HR roles, and audit trails are stored immutably. Regular compliance checks are performed for statutory payroll rules and personal data protection regulations.

Operational Notes & Runbooks

Daily jobs: nightly payroll calculation, weekly headcount sync, daily biometric attendance ingestion.

Monitoring: Prometheus metrics for service health, Grafana dashboards for SLA and latency monitoring.

Backup & Recovery: Point-in-time recovery enabled for PostgreSQL; nightly snapshots stored off-site.

Roadmap & Enhancements

Next milestones include self-service onboarding automation, manager analytics for performance calibration, and integrating a compensation planning module with predictive budgeting.

2. Project Horizon – AI-Based Customer Support Assistant

Overview

Project Horizon is an internal AI assistant designed to automate first-line customer support for product teams by answering common tickets, suggesting responses, and routing complex issues to the right engineering owners. Horizon uses a RAG pipeline to ground its answers in internal documentation, past tickets, and product release notes, minimizing hallucinations while providing context-rich responses.

Key Features

• RAG-powered response generation (retriever + generator).

• Ticket summarization & suggested replies.

• Automated tagging and intent classification.

• Escalation routing with suggested assignees based on past ownership.

• Feedback loop to incrementally update knowledge base from resolved tickets.

Architecture & Pipeline

Horizon's pipeline follows: Ingest → Split → Embed → Index → Retrieve → Rerank → Generate.

Ingestion sources include internal docs, Confluence pages, closed tickets from the helpdesk, and product changelogs. Documents are split with a hybrid semantic + fixed-size splitter to preserve context while maintaining efficient retrieval. Embeddings are computed using a sentence-transformer model and stored in FAISS with metadata fields (project, date, doc_type).

Modeling & Prompting

The generation model is accessed via the LLM API (configurable — e.g., Gemini / OpenAI). The prompt template includes: retrieved context (top k passages), a short ticket summary, and system instructions enforcing source citation. Re-ranking leverages an additional lightweight model that scores retrieved passages for relevance before the generation step.

Evaluation & Metrics

Horizon is evaluated on response accuracy (% of correct answers), ticket resolution uplift (reduction in human handling time), and user satisfaction scores collected via post-resolution surveys. A/B tests compare fully automated reply vs. human-in-the-loop suggestions.

Key metrics: Precision@1 for retrieval, ROUGE/BLEU for summaries (where applicable), and human-annotated faithfulness checks.

Operational Considerations

Continuous ingestion pipeline ensures new docs are embedded nightly. A feedback ingestion job captures agent-corrected answers to retrain the re-ranker and update embeddings. Throttling, rate-limiting, and query caching reduce cost and latency. A manual override UI lets support leads review and approve automated responses before sending.

Safety & Guardrails

Strict policies define what Horizon can and cannot respond to (e.g., account deletions, billing disputes require manual handling). Sensitive PII is redacted at ingestion, and the system logs provenance for all generated responses to allow audits.

3. Project NovaPay – Secure Payment Gateway

Overview

NovaPay is an internal payments solution designed to handle vendor payouts, internal billing, and integration with customer-facing billing systems. It supports recurring billing, invoice lifecycle management, reconciliation, and multiple payment rails including UPI and NetBanking. The service emphasizes security and regulatory compliance across transaction handling.

Key Features

• Recurring payments & subscription plans.

• Invoice generation & templating engine.

• Multi-rail payment support (UPI, bank transfer, card passthrough via tokenization).

• Reconciliation engine to match incoming transactions with invoices.

• Fraud detection & anomaly scoring based on transaction patterns.

Payment Flow & Architecture

Payment requests are accepted via authenticated APIs, validated against merchant configuration, and routed to the appropriate payment provider adapter. The system uses idempotent operations to handle retries safely. A reconciliation batch job consumes settlement reports and attempts to auto-match with ledger entries; unmatched items raise alerts for manual investigation.

Security & PCI Considerations

NovaPay avoids storing raw card data by using tokenization and external card vault providers. All payment communications use TLS 1.2+, and HSM-backed keys are used for sensitive cryptographic operations. Regular PCI-DSS assessments are scheduled, and access to transaction logs is tightly audited.

Monitoring & Auditing

Transaction latency, success rates, and reconciliation deltas are monitored. A dedicated reconciliation dashboard surfaces pending settlements, exceptions, and dispute counts. Audit logs are tamper-evident and retained per the company's retention policy.

Scaling & Reliability

Critical payment components operate in multiple availability zones with a circuit-breaker pattern for downstream provider failures. Queue-based decoupling ensures peak loads are handled gracefully, and rate-limiting prevents provider throttling. Blue/green deployments for payment adapters minimize downtime and risk.

4. Project Sentinel – Real-Time Security Event Monitoring

Overview

Project Sentinel provides centralized observability for security signals across TechNova's infrastructure and applications. It ingests logs, metrics, and traces, applies correlation rules and ML-based anomaly detection, and surfaces prioritized incidents to the security operations team.

Key Capabilities

• Log aggregation and normalization.

• Real-time alerting and incident prioritization.

• ML anomaly detection for unusual patterns (login spikes, exfiltration signatures).

• Forensic timelines and automated containment playbooks.

• Integration with SIEM and ticketing systems for incident lifecycle management.

Data Ingestion & Processing

Sentinel collects logs via Filebeat/Fluentd agents, streams them through Kafka into processing pipelines. Parsers normalize events into a canonical schema; enrichment adds geo/IP reputation, user context, and threat intelligence feeds. A combination of rule-based detectors and unsupervised ML models produces signals with confidence scores.

Detection Models & Playbooks

Detectors include spike detectors, sequence-based anomaly models for user behavior, and signature matches for known threats. For high-confidence incidents, the system can trigger automated containment steps (e.g., isolate a host, revoke tokens) following established playbooks; lower-confidence alerts are queued for analyst review.

Observability & Response

Analysts use a dashboard that shows correlated events, affected assets, and suggested remediation steps. Case management tracks investigations and incident timelines. Integration with chatops tools enables one-click runbook execution for containment actions.

Governance & Compliance

Sentinel maintains an audit trail for all automated actions and analyst interventions. Data retention policies align with regulatory requirements for logs and evidence; sensitive logs are redacted and only accessible to authorized security personnel.

5. Project AeroDocs – Internal Documentation Portal

Overview

AeroDocs is the single-pane documentation platform for internal teams, designed for discoverability, versioning, access control, and rapid authoring. It consolidates guides, runbooks, API references, and onboarding material with a built-in RAG search assistant to surface authoritative answers quickly.

Core Features

• Document versioning & diffs.

• Role-based access control and team workspaces.

• Full-text search with metadata filters (owner, project, last-updated).

• Inline commenting and editorial workflows.

• RAG-powered assistant that cites document passages and ranks authoritative sources.

Content Model & Indexing

Documents are stored using a metadata-rich model: title, slug, tags, owner, team, status (draft/published), and content sections. The indexing pipeline extracts structured sections, tables, and code blocks, creating chunked entries that are embedded and stored in the vector index. Metadata is preserved to ensure retrieval can be filtered by team or document type.

Authoring & Editorial Workflow

Authors create and preview content using a rich editor supporting Markdown and rich blocks (code, diagrams, callouts). An editorial approval workflow routes drafts through reviewers; publish actions update the search index and notify subscribers.

Audit logs track who changed what and when.

Search Assistant Integration

The RAG assistant uses a retrieval policy that biases towards official 'published' documents and senior-engineer-approved how-tos. It surfaces citations and provides quick links to the full document, helping reduce misinterpretation of context.

6. Project SkyBridge – Integration Hub for Microservices

Overview

SkyBridge provides a unified integration layer that simplifies service-to-service communication, authentication, routing, and observability for TechNova's microservices landscape. It centralizes cross-cutting concerns so individual services can remain focused on business logic.

Key Capabilities

• API gateway & routing with intelligent request shaping.

• Centralized auth & identity propagation.

• Service discovery and dynamic routing policies.

• Request/response transformation and protocol bridging.

• Observability: request traces, metrics, and distributed logging correlation.

Design & Implementation

SkyBridge is implemented using Spring Cloud Gateway with custom filters for auth and observability. It integrates with the service registry (Consul/Eureka) and uses Envoy sidecars in advanced deployments for traffic management. Policies for retries, circuit-breakers, and rate-limiting are centrally managed and applied per-service.

Deployment & Scaling

Gateway instances are horizontally scalable; a control plane manages routing rules and certificate rotation. Canary releases and staged rollouts are supported via feature flags and traffic shadowing. Health checks and autoscaling policies are configured through Kubernetes HPA/VPA.

Operational Playbooks

Common operational playbooks include tracing slow requests, diagnosing 502/504 errors, and reverting routing rules. Observability integrates with distributed tracing (Jaeger), metrics (Prometheus), and logs (ELK), enabling rapid root-cause analysis.

Appendix: Deployment & Maintenance Checklists

Deployment Checklist:

• Ensure schema migrations run in a non-blocking fashion and are backward compatible.

• Verify environment variables, secrets, and Key Management access.

• Smoke test critical paths (login, core API endpoints, background job runners).

• Run load tests in staging for expected peak loads.

Maintenance & On-call:

• Define SLOs and SLAs for each service.

• Maintain runbooks for common incidents and escalation paths.
• Weekly sanity checks for batch jobs and ingestion pipelines.
• Monthly security reviews and dependency updates.