KAOS — Technology Architecture Overview

This document provides a high-level overview of the technology stack and architecture of the KAOS platform.

1. Architectural Principles

The KAOS platform is designed based on the following core principles:

- **Scalability:** A microservices-based architecture designed to handle millions of workflows and tasks concurrently.
- **Reliability:** High availability and fault tolerance built in at every layer, with automated recovery mechanisms.
- **Security:** A zero-trust security model with end-to-end encryption, role-based access control (RBAC), and comprehensive audit logging.
- **Extensibility:** A plugin-based architecture that allows for the rapid development and deployment of new integrations and agents.

2. Technology Stack

Layer	Technology	Rationale
Frontend	React, TypeScript, Tailwind CSS	Modern, component-based UI for a fast and responsive user experience.
Backend	Node.js, TypeScript, Express	Event-driven, non-blocking I/O model ideal for handling a high volume of concurrent workflows.
Database	PostgreSQL, Redis	PostgreSQL for reliable transactional data and Redis for high-performance caching and message queuing.
Infrastructure	Docker, Kubernetes, AWS	Containerization and orchestration for automated scaling, deployment, and management on a leading cloud provider.
CI/CD	GitHub Actions, Terraform	Automated testing, infrastructure-as-code, and continuous deployment to ensure rapid and reliable releases.

---

3. Core Components

Platform Architecture

(Note: A proper architecture diagram should be created and linked here)

- **Workflow Engine:** The heart of the platform, responsible for parsing workflow definitions, managing state, and orchestrating the execution of tasks.
- **Agent Framework:** A standardized framework for building, managing, and deploying both proprietary and third-party AI agents.
- **Integration Layer:** A universal API and a set of pre-built connectors that allow KAOS to communicate with any external system.
- **Governance & Audit Service:** A centralized service that enforces security policies, manages access control, and logs every action taken on the platform.
- **API Gateway:** A single entry point for all external API requests, providing authentication, rate limiting, and routing to the appropriate microservice.

4. Data Flow Example: Invoice Processing

1. **Trigger:** An email with an attached invoice arrives in a designated inbox.
2. **Integration:** The Email Connector receives the email and triggers the “Invoice Processing” workflow.
3. **Orchestration:** The Workflow Engine executes the first step: an AI Agent extracts the invoice data (vendor, amount, due date) using OCR and NLP.
4. **Integration:** The workflow calls the Xero API to create a new bill with the extracted data.
5. **Governance:** The Governance Service logs the entire transaction, including the user who initiated the workflow and the data that was processed.
6. **Automation:** The workflow sends a Slack notification to the finance team for final approval.