

# MyPPP System : AI-Driven Public-Private Partnership Proposal Management System

Swastik Ram (Developer)

[swastikram1818@gmail.com](mailto:swastikram1818@gmail.com)

<https://github.com/DEEPML1818>

Navinder Dinesh Ram (DevOps)

[navinder.tech@gmail.com](mailto:navinder.tech@gmail.com)

<https://www.navinder.tech/>

**Abstract.** The **MyPPP System** is an AI-driven and blockchain-integrated platform designed to streamline the management of Public-Private Partnership (PPP) project proposals. Developed for the Public-Private Partnership Unit (UKAS), the system automates the manual processes of proposal submission, evaluation, and decision-making. Utilizing Natural Language Processing (NLP) for proposal parsing, AI for similarity analysis, and blockchain for tamper-proof transparency, the system reduces the workload on government officers while ensuring efficiency and accountability. The MyPPP System features an end-to-end proposal management lifecycle, real-time tracking, and advanced reporting, making it a scalable solution for both state and national rollouts.

# Executive Summary

The **MyPPP System** addresses key inefficiencies in the manual handling of PPP project proposals by UKAS. Traditionally, officers have relied on time-consuming processes for reviewing project submissions, assessing compliance, and locating similar past proposals. The MyPPP System introduces automation at every stage, from submission to final decision, leveraging AI-powered screening to ensure proposals meet predefined criteria and comparing new submissions with a historical database using deep learning models. The integration of blockchain guarantees transparency, providing an immutable audit trail of every proposal's journey. Additionally, the system's CRM functionality supports real-time lifecycle management and monitoring through user-friendly dashboards. This digital transformation aims to improve productivity, reduce delays, and facilitate better decision-making, ultimately enabling the Malaysian government to evaluate and implement impactful projects more efficiently. Following successful pilot tests, the system will be scaled statewide and eventually rolled out nationwide.

## Problem Statement

**General Information:** UKAS is a central agency responsible for planning, evaluating, coordinating, and negotiating the terms of PPP project implementation in Malaysia. Between 2010 and August 2024, a total of 444 project proposals were submitted to UKAS, and based on initial evaluations, the recommendations are as follows:

- 241 (54%) rejected
- 96 (22%) no feedback
- 5 (1%) postponed
- 102 (23%) proceeded for further review

Project proposal evaluations are currently performed manually, resulting in wasted time, human resources, and financial inefficiencies for both the government and private sector. Moreover, viable projects that can generate positive impacts are delayed in implementation.

## Key Challenges:

The digital issues faced are:

- No tool for initial screening of PPP project proposals.
- All project proposals are in technical paper format, which takes a long time to review and assess manually by UKAS officers.
- Difficulty in identifying similarities between new PPP project proposals and previously evaluated ones due to manual review processes.

## Target Group/Beneficiaries:

The system's target users are:

- Ministries/user agencies and private companies submitting PPP projects to UKAS
- UKAS officers responsible for receiving, evaluating, and processing all PPP project proposals, as well as UKAS management as overseers.

## Pain Points:

The main pain points are:

- **Manual evaluation of project proposals**, requiring diverse expert officers and long processing times
- Difficulty finding the right experts to evaluate the proposals, as the scope of proposals varies depending on the creativity of the proposing company.
- Difficulty identifying similar project proposals written in different contexts, manually.
- **Project proposals not following the proper writing criteria**, making the evaluation time-consuming and challenging.

## Solution Target:

We need a solution to:

- **An AI-driven system** that allows applicants (ministries, user agencies, and private companies) to prepare project proposals according to predefined criteria.
- The system should analyze the **similarity** of new project proposals by comparing them with the context of previously evaluated proposals.
- The system should **manage, track status**, and provide **visualization** of project proposals for UKAS management.
- This will be a **custom CRM system integrated with AI**, easing project management and tracking for the UKAS team while offering management, tracking, and visualization capabilities for all stakeholders.
-

## Direct Impact:

The main impacts of this solution will be:

- **Faster processing** of project proposals.
  - **Increased productivity** for UKAS officers through time and resource savings in proposal evaluation.  
UKAS management will be able to **observe proposal trends** from ministries/user agencies and private companies, aiding in the formulation of PPP project policies and strategies.
- 

## Technical Solution Outline:

### 1. AI-Powered Proposal Screening System:

- **Natural Language Processing (NLP)** will be employed to parse and understand project proposals submitted in various formats, such as PDFs or documents. This will enable the system to screen and assess whether the proposals meet predefined PPP criteria.
- **Machine Learning (ML)** algorithms can be trained on historical project proposal data to automatically assess the viability and relevance of new project proposals. These models will be capable of identifying keywords, proposal structures, and compliance with UKAS guidelines.

## 2. Automated Similarity Analysis:

- Implement a **contextual similarity engine** using deep learning models like BERT or GPT-based language models. This feature will compare new proposals with a database of previously submitted projects to detect similarities in objectives, financial models, risk-sharing mechanisms, or implementation strategies.
- **Clustering algorithms** could be used to group similar project proposals, making it easier for UKAS officers to review past assessments and streamline decision-making.

## 3. Custom CRM with AI Integration:

- Develop a **Custom CRM system** specifically designed to handle the lifecycle of PPP project proposals. This CRM will manage project intake, evaluation, feedback, and tracking.
  - AI integration within the CRM will facilitate proposal routing to the appropriate experts and provide **automated alerts** when proposals deviate from established criteria.
- The CRM should also include features for **data visualization**, allowing UKAS management to monitor proposal statuses, track evaluation progress, and identify trends in PPP project submissions over time.

## 4. Project Proposal Management and Tracking Dashboard:

- A **centralized dashboard** that provides a real-time view of the status of all proposals, including submitted, in-review, accepted, or rejected.
- The dashboard should offer **visual insights** into project trends, department-wise proposal submissions, time spent in each review stage, and expert assignments. Integration with AI-driven forecasting models can help management predict future proposal trends and resource requirements.

### 5. AI-Based Compliance Checker:

- The system will feature an **AI-driven compliance checker** that verifies if the submitted project proposals adhere to UKAS' PPP proposal format and guidelines. This will save time in manual reviews and ensure that only compliant proposals proceed to the evaluation stage.

### 6. Data-Driven Insights and Reports:

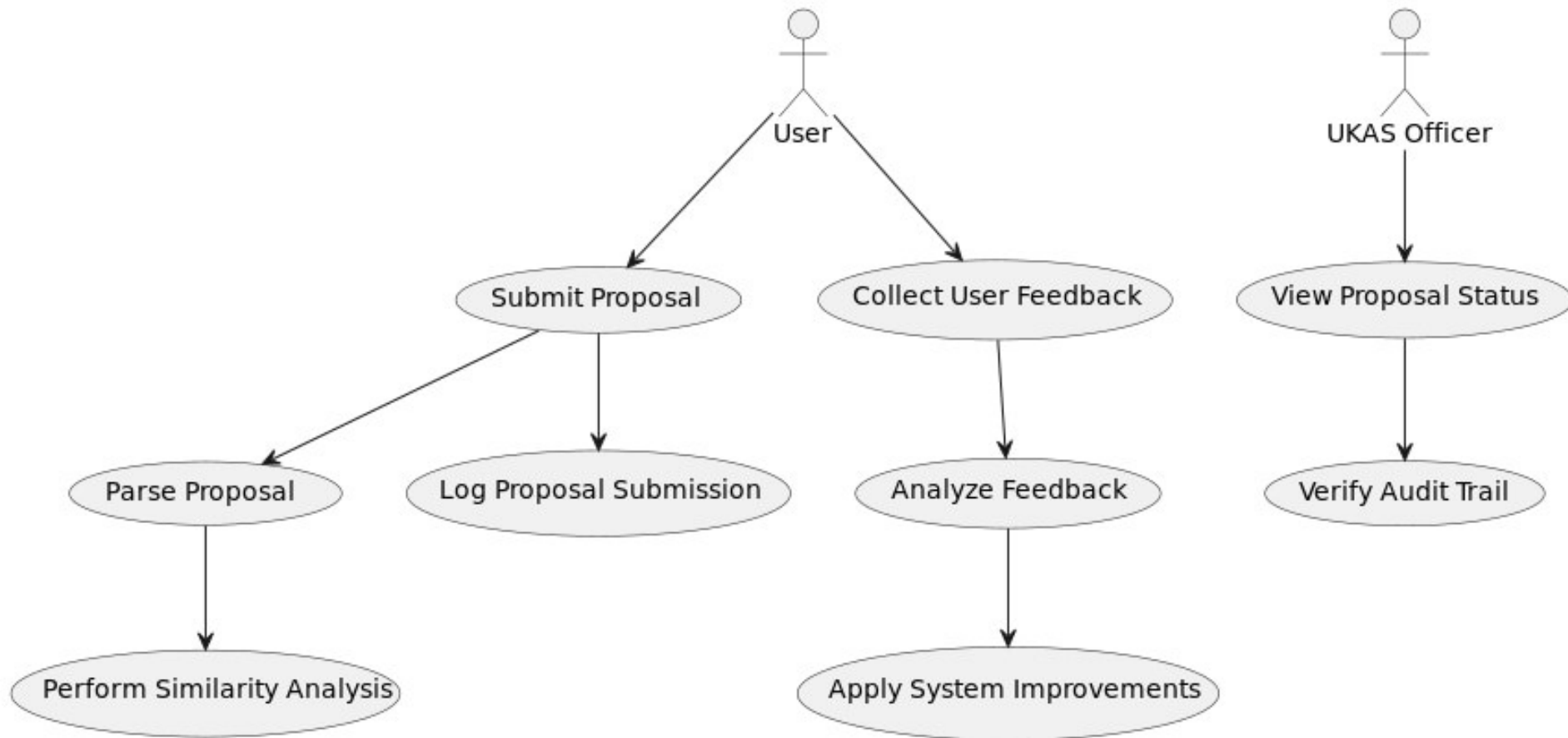
- Provide **data analytics and reporting tools** within the CRM, enabling UKAS officers and management to generate reports on project trends, evaluation bottlenecks, and sector-specific data insights
- These reports can help UKAS make **data-driven decisions** about strategic project selection and resource allocation.

### 7. Blockchain Integration for Proposal Transparency:

- Consider integrating **blockchain technology** for proposal tracking and transparency. Each proposal's submission, review, and evaluation could be recorded on a tamper-proof blockchain, ensuring accountability and transparency in the proposal evaluation process.

This AI-powered **MyPPP system** will drastically reduce the manual workload of UKAS officers, ensure faster project proposal reviews, and provide comprehensive insights for better management and strategic planning of PPP projects in Malaysia. The combination of AI for automated assessment, machine learning for similarity analysis, and custom CRM for project lifecycle management will lead to a highly efficient digital solution for managing PPP projects.

# Use Case



*This diagram highlights the main use cases in the MyPPP System, showcasing the interaction between users (e.g., Users, UKAS Officers) and the system's core functionalities, including feedback collection and iteration.*



# Information Systems

## 1. Technology Stack

### Backend

- **Python/Django:** Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It will be used to build the backend API and business logic for the system.
- **PostgreSQL:** An open-source relational database that is highly scalable and integrates well with Django. It will store all project proposal data, user information, proposal status, and audit trails.
- **Celery:** Distributed task queue system for managing background tasks (e.g., sending notifications, performing similarity analysis).
- **Redis:** In-memory data store used for caching and task queueing (via Celery).

### AI and Machine Learning

- **SpaCy:** An open-source library for advanced NLP. It will handle proposal parsing and initial screening.
- **BERT (via Hugging Face):** A pre-trained deep learning model for language understanding. It will power the similarity analysis for comparing proposals.
- **Scikit-learn:** A machine learning library used for training models on historical project proposal data to predict the viability of new proposals.

## Frontend

- **React.js:** A popular open-source JavaScript library for building dynamic user interfaces. It will be used for creating the project proposal management dashboard and user interfaces for the CRM system.
- **Tailwind CSS:** A utility-first CSS framework to build responsive, modern designs for the frontend.

## Blockchain

- **Hyperledger Fabric:** An open-source blockchain framework designed for enterprise use. It will track proposal submissions, reviews, and evaluations to ensure transparency.

## Other Integrations

- **Elasticsearch:** An open-source search engine used for full-text search and filtering project proposals in the database.
- **Kibana:** A data visualization dashboard that integrates with Elasticsearch to monitor trends and display key metrics for UKAS management.

## DevOps

- **Docker:** A platform for developing, shipping, and running applications in containers. Used for containerizing the application for consistent deployments.
  - **Kubernetes:** For container orchestration and scaling the application as needed.
  - **GitLab CI/CD:** For automated testing and deployment pipelines.
-

## 2. Functionalities

### AI-Powered Proposal Screening

- **NLP Parsing:** Proposals submitted in various formats (PDF, DOCX) will be parsed using SpaCy to extract meaningful information.
- **Criteria Check:** The system will automatically check if the proposal aligns with predefined PPP criteria, such as project structure, financial models, and risk-sharing mechanisms.

### Automated Similarity Analysis

- **Proposal Comparison:** Using BERT, the system will compare new project proposals against previously submitted ones to identify similarities in objectives, risks, and execution strategies.
- **Clustering:** The system groups similar proposals using clustering algorithms to assist UKAS officers in locating past evaluations quickly.

### CRM System

- **Lifecycle Management:** The CRM manages the entire lifecycle of a proposal: from submission, screening, expert review, to approval or rejection.
- **Proposal Routing:** AI-powered routing of proposals to the right experts for review based on proposal content and expert availability.
- **Notifications & Alerts:** Automatic alerts for incomplete or non-compliant proposals.

## Dashboard & Tracking

- **Proposal Status Overview:** Real-time dashboard displaying the current status of all submitted proposals (pending, in-review, accepted, rejected).
- **Visualization:** Trends in proposal submissions and evaluations, categorized by departments, time-to-review, etc., are presented visually.

## Blockchain Integration

- **Immutable Tracking:** Every step in the proposal review process is stored on Hyperledger Fabric, ensuring tamper-proof records.
- **Audit Trail:** Blockchain-based audit logs for transparency and accountability.

## Data Analytics & Reporting

- **Data Insights:** Reports generated on project submission trends, evaluation bottlenecks, and resource needs using Kibana.
- **Predictive Analytics:** Forecasting future proposal trends using machine learning models.

### 3. Features

- **Full Proposal Lifecycle Tracking:** From submission to review, decision, and auditing.
- **AI Compliance Checker:** Automated proposal format and compliance checks.
- **Proposal Similarity Analysis:** Deep learning models to identify commonalities between proposals.
- **Customizable Dashboard:** User-friendly interface with real-time updates and visualizations for management.
- **Blockchain Transparency:** Immutable records for proposal submissions and reviews.
- **Automated Routing & Notifications:** Streamlined proposal review and alerts for incomplete or non-compliant submissions.
- **Report Generation:** On-demand reporting on trends, bottlenecks, and outcomes.

### 4. Workflow

1. **Submission:** Ministries, user agencies, and private companies submit proposals via the frontend (React.js) interface.
2. **AI Compliance Check:** Proposals are screened for compliance with predefined PPP criteria.
3. **Similarity Analysis:** The system runs a similarity check against previously submitted proposals.
4. **Routing:** The proposal is routed to the appropriate UKAS officers for review based on the content and expertise required.

5. **Review:** UKAS officers review the proposal, aided by the system's suggestion engines and insights into similar proposals.
6. **Decision:** The proposal is either accepted, rejected, or returned for revision.
7. **Blockchain Record:** All key actions (submission, review, and decision) are recorded on Hyperledger Fabric.
8. **Tracking & Reporting:** The dashboard provides real-time status tracking and generates periodic reports.

## 5. Product or Modules (Itemized)

### 1. Proposal Submission Module:

- Allows users to submit project proposals via a web interface.
- AI-powered initial screening for compliance.

### 2. AI Compliance Checker Module:

- Checks if proposals meet PPP criteria and flags incomplete ones.

### 3. Similarity Analysis Module:

- Compares new proposals to existing ones using NLP and machine learning.
- Groups similar proposals for quick reference.

### 4. Custom CRM Module:

- Manages the entire proposal lifecycle from submission to review, routing, and final decision.

## 5. Dashboard and Tracking Module:

- Provides UKAS with a visual dashboard of proposal status and trends.

## 6. Blockchain Integration Module:

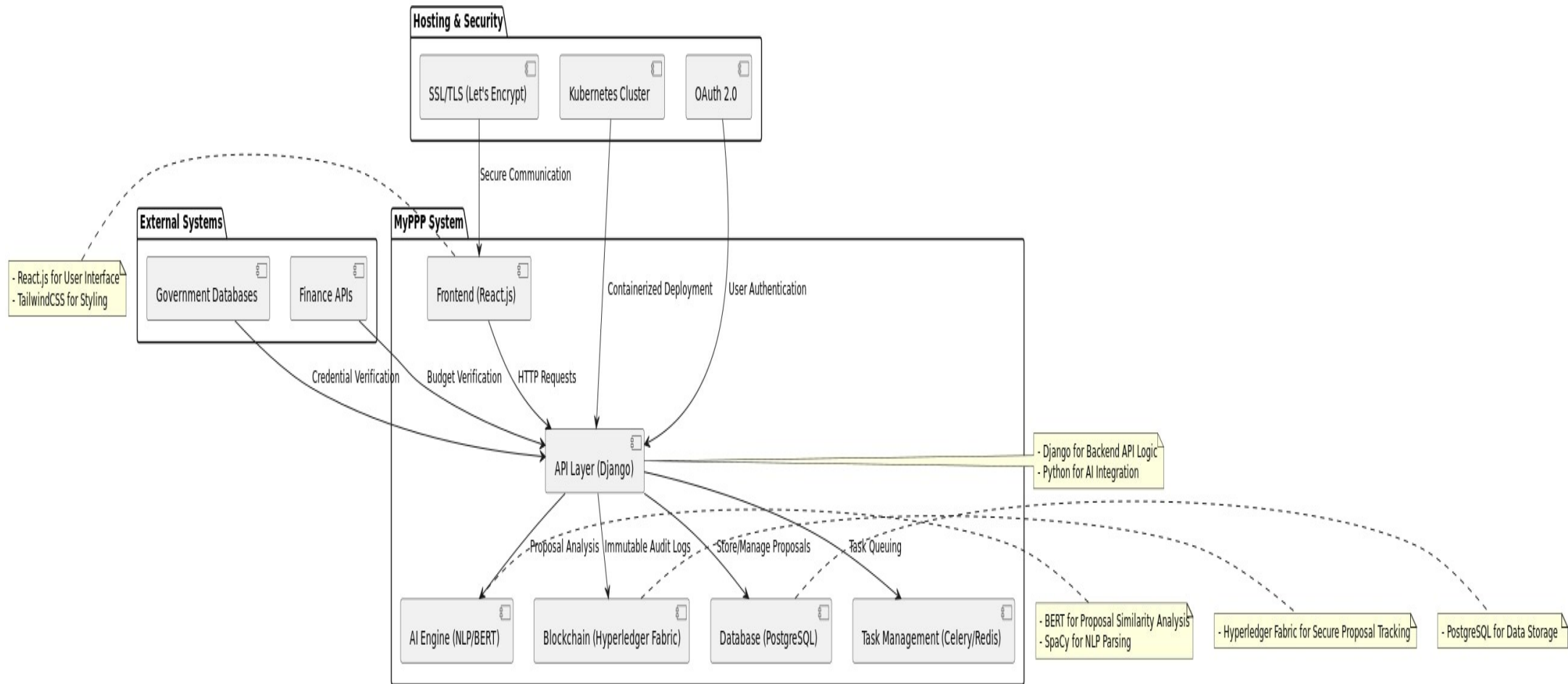
- Logs proposal submission and review events on a tamper-proof blockchain.

## 7. Reporting & Analytics Module:

- Generates insights and trend reports for UKAS management.

This open-source architecture provides a robust, scalable, and AI-driven solution to streamline PPP project proposals, significantly improving the efficiency of the review and evaluation process for UKAS.

# Architecture



*This diagram showcases the core components of the MyPPP System, including the frontend, backend, AI engine, blockchain, and external integrations, and how they interact to manage PPP proposals.*



# Technical Implementation Design

## 1. Overview

The **MyPPP System** is designed to digitize the process of managing Public-Private Partnership (PPP) project proposals. The system aims to reduce manual processes, provide AI-driven screening and analysis of project proposals, and offer real-time tracking, status updates, and data visualization for UKAS officers and management. Key features include NLP-based proposal screening, similarity analysis using AI models, a custom CRM system, and integration with blockchain for transparency.

### Key Goals:

1. Automate proposal submissions and reviews.
2. Provide a robust AI engine for initial screening, similarity analysis, and routing.
3. Deliver a centralized tracking system with dashboard visualization for UKAS management.
4. Ensure secure and transparent operations via blockchain.

## 2. Architecture Overview

### 2.1 High-Level Architecture:

The system will follow a **microservices architecture**, using containerization (Docker) to isolate different system components, and orchestrate them using **Kubernetes** for scalability. The system comprises the following components:

- **Frontend Service** (React.js): User interface for submitting proposals, tracking status, and managing interactions.
- **Backend Service** (Django): Core business logic, API services, and task coordination.
- **AI Engine** (NLP and ML): Parses, analyzes, and screens proposals.
- **Database Service** (PostgreSQL): Central repository for storing project data, user information, and historical proposals.
- **Blockchain Service** (Hyperledger Fabric): Stores immutable logs of project submissions, reviews, and decisions.
- **Message Broker/Queue** (Celery + Redis): Asynchronous task management for AI screening, similarity analysis, and background processes.

## 2.2 Detailed System Components:

### Frontend (React.js + Tailwind CSS)

- **User Interface:** Ministries, user agencies, and private companies submit project proposals and track their progress.
- **Admin Interface:** UKAS officers can manage, review, and approve or reject proposals. The interface provides visualization and reporting dashboards for management.
- **API Communication:** Communicates with the backend API to fetch data, submit proposals, and trigger workflows.

### Backend (Django)

- **Core API:** RESTful APIs that handle requests from the frontend, including:
  - Proposal submission
  - Proposal status tracking
  - Expert review workflows
  - Proposal comparison and similarity analysis
- **Task Management:** Uses **Celery** with **Redis** for managing long-running processes like AI-driven screening and similarity analysis in the background.

## AI-Powered Proposal Screening (NLP + ML)

### 1. NLP Engine (SpaCy):

- Parses and extracts data from unstructured proposals (PDFs, DOCX, etc.) to identify key sections (e.g., project objectives, financials).
- Verifies that the proposal follows predefined PPP structure and guidelines.

### 2. Proposal Classification:

- Uses **Scikit-learn** to classify proposals based on historical data, determining their initial viability and adherence to guidelines.

### 3. Similarity Analysis (BERT):

- Contextual analysis of new proposals using **BERT (via Hugging Face)** to compare against a database of previously evaluated projects.
- Outputs similarity scores, highlighting areas where new proposals match past projects.

### 4. Clustering (K-Means):

- Groups similar proposals for easy access to past reviews, reducing manual review times.

## Database (PostgreSQL)

- **Relational Data Model:** Stores structured data on submitted proposals, users, evaluation histories, and expert reviews.
- **Proposal Metadata:** Stores extracted metadata from NLP parsing, making it easy to query and visualize proposal data.

### **Blockchain Integration (Hyperledger Fabric)**

- **Proposal Submission Logs:** Each submission event is logged on the blockchain, providing a tamper-proof record.
- **Evaluation Audit Trail:** Logs the status and results of each proposal review, ensuring transparency for UKAS officers and stakeholders.
- **Smart Contracts:** Automates the submission, review, and decision-making workflow via smart contracts.

### **Search Engine (Elasticsearch)**

- **Full-Text Search:** Enables fast searching of project proposals based on extracted metadata and key text segments.
- **Proposal Filters:** Provides advanced filtering capabilities (e.g., by project type, sector, date submitted, etc.).

### **Data Visualization & Reporting (Kibana)**

- **Dashboard:** Displays visual insights into project proposal status, bottlenecks, trends in submissions, and department-specific data.
- **Reports:** Generates reports on time spent in review stages, sector-specific trends, and evaluation efficiency, helping UKAS management make data-driven decisions.

### Task Queue (Celery + Redis)

- **Asynchronous Processing:** Offloads heavy tasks like proposal parsing, similarity analysis, and notifications to background workers.
- **Real-Time Notifications:** Sends alerts to UKAS officers when new proposals are submitted, reviewed, or when manual intervention is needed.

## 3. Functionalities

### 3.1 AI Proposal Screening:

- **NLP Parsing:** Automatic extraction of key details (objectives, financials, implementation plan) from submitted documents.
- **PPP Criteria Compliance Check:** Checks the extracted content against predefined PPP project guidelines, flagging non-compliant proposals.

### 3.2 Similarity Analysis:

- **Contextual Comparison:** Uses AI models to assess similarities between newly submitted proposals and past projects, helping to avoid redundancy.
- **Cluster Grouping:** Similar proposals are grouped together, and officers can reference past reviews to expedite new evaluations.

### 3.3 CRM System:

- **Proposal Lifecycle Management:** Tracks the entire lifecycle of each project proposal, from submission to review, approval, or rejection.
- **Expert Review Assignment:** Automatically routes proposals to appropriate expert reviewers based on content.

### 3.4 Dashboard & Status Tracking:

- **Real-Time Status:** Provides real-time updates on the status of all proposals (submitted, in-review, approved, rejected).
- **Trend Visualization:** Displays proposal submission trends by sector, type, and department.

### 3.5 Blockchain Audit & Transparency:

- **Immutable Records:** Logs key proposal lifecycle events (submission, review, approval) on the blockchain.
  - **Audit Trails:** Enables UKAS officers and management to review a tamper-proof audit trail of all proposal-related actions.
-

## 4. Workflow

### 1. Proposal Submission:

- Users submit project proposals via the React.js frontend.
- Proposals are uploaded in PDF or DOCX format and sent to the backend for processing.
- The AI engine parses the proposal and checks for compliance with PPP guidelines.
- Proposals that do not comply are flagged for revision, and notifications are sent to the user.

### 2. NLP Parsing & Compliance Check:

- The backend runs similarity checks using BERT-based contextual analysis.
- If similar projects are found, they are clustered and displayed to UKAS officers.

### 3. Similarity Analysis:

- Proposals are routed to appropriate UKAS officers based on content and expert availability.
- Officers review the proposals, and the system tracks their actions in real-time.

### 4. Routing & Review:

- Once a decision is made, it is logged on the blockchain, ensuring an immutable audit trail.
- Approved, rejected, or revised proposals are automatically updated on the dashboard.
- UKAS officers can monitor the status of all proposals in real-time via the dashboard.

- The system generates reports on bottlenecks, review times, and other key metrics for management.

### 5. Decision & Blockchain Logging:

### 6. Tracking & Reporting:



## 6. Scalability and Future Enhancements

### 6.1 Scalability:

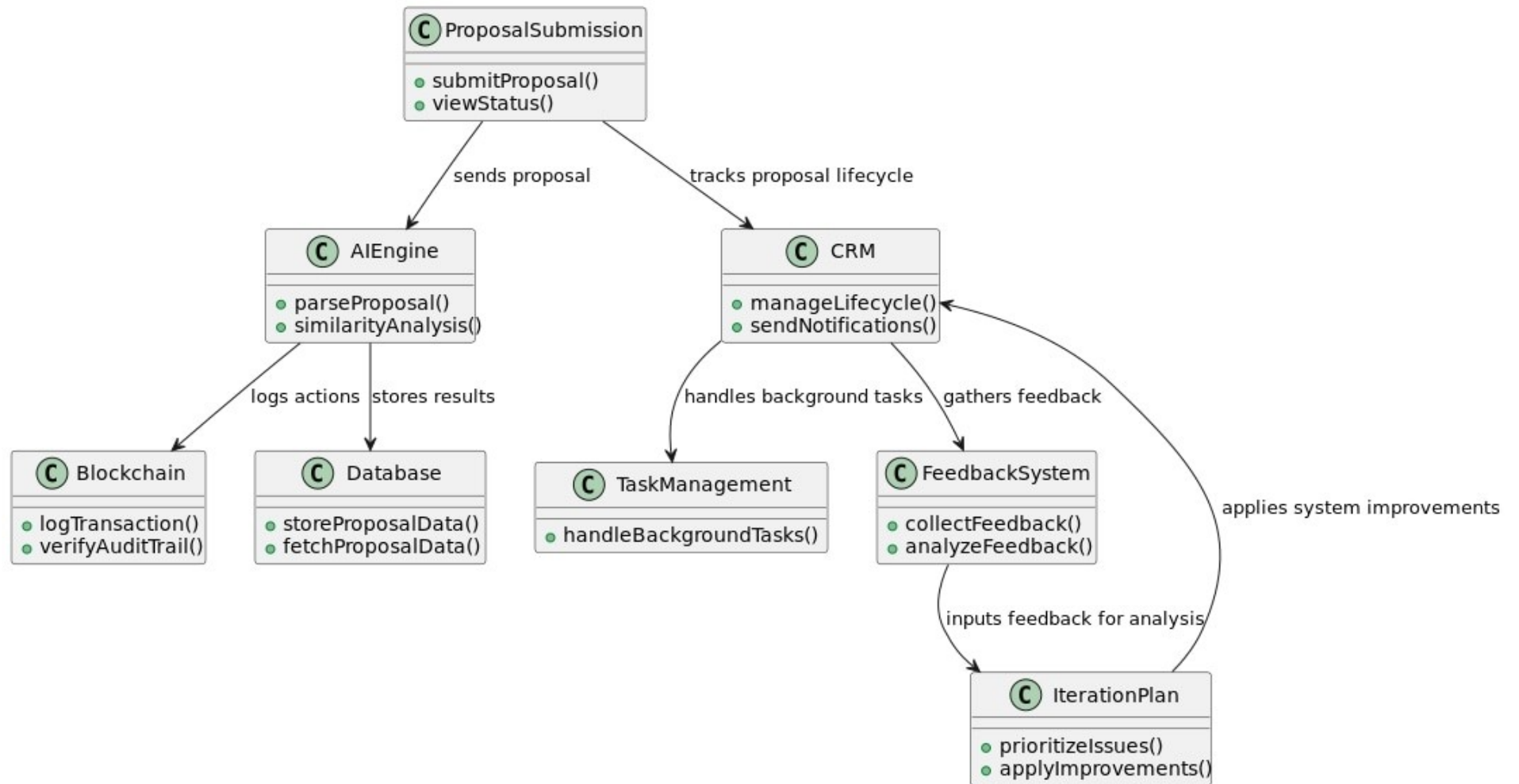
- **Microservices & Kubernetes:** As the number of proposals grows, containerized services can be scaled independently, allowing for horizontal scaling across the AI, frontend, and backend services.
- **Database Sharding:** PostgreSQL sharding techniques can be applied as the proposal data set grows to ensure fast queries and smooth operations.

### 6.2 Future Enhancements:

- **Automated Decision Making:** Implement advanced machine learning models to predict the success or failure of PPP projects based on historical data.
- **Integration with Government Databases:** Seamless integration with existing government systems for real-time verification of private partner credentials.
- **Enhanced Security:** Adding multi-factor authentication (MFA) and end-to-end encryption for sensitive proposal data.

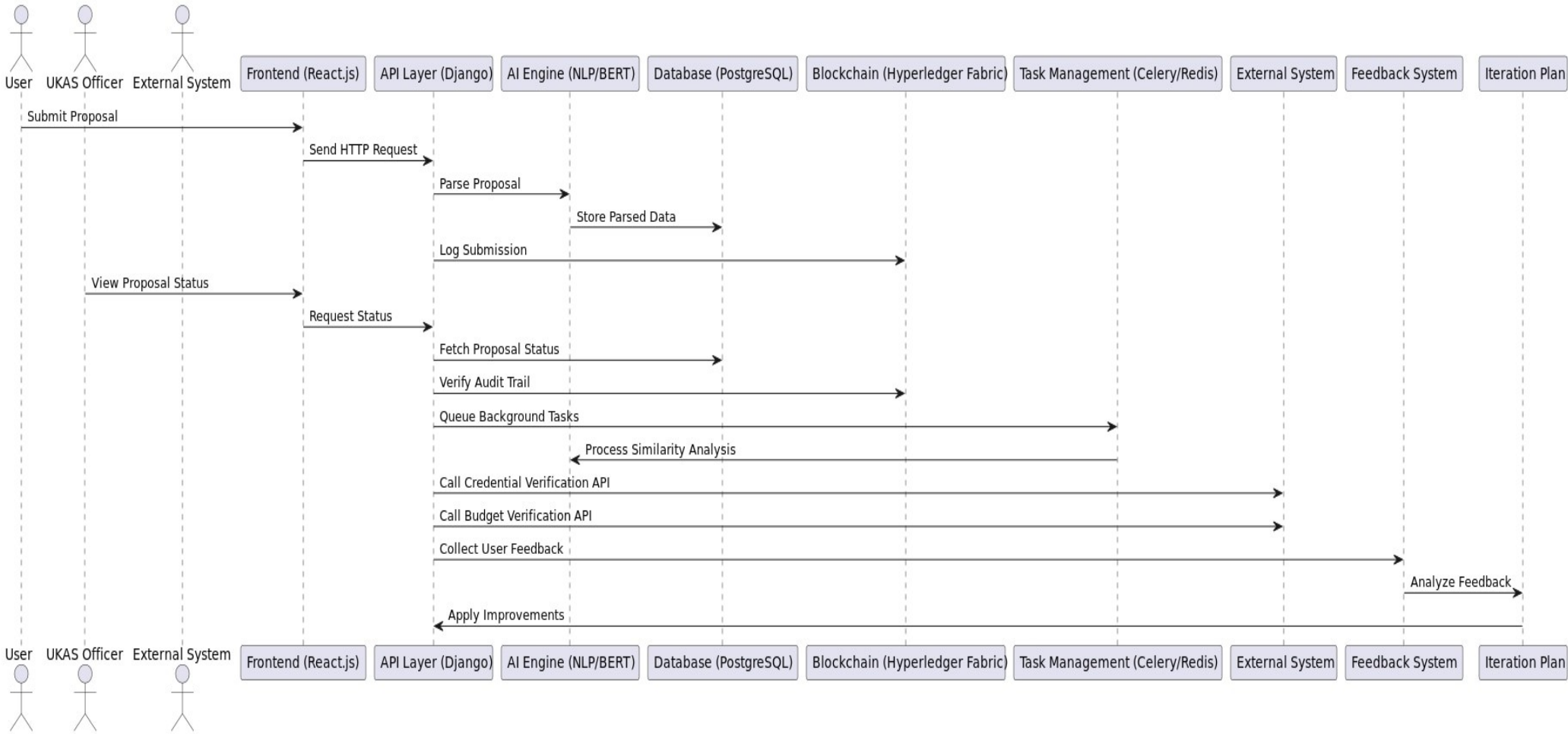
The **MyPPP System** provides a streamlined, AI-driven, and blockchain-integrated platform for managing PPP project proposals. It automates manual processes, ensures transparency through blockchain, and provides UKAS officers with a scalable and efficient solution for reviewing and approving proposals.

# Class



*This Class Diagram models the main components of the MyPPP System, including the system's key functionalities and feedback/iteration mechanisms, as extracted from the document.*

# Sequence



*This diagram showcases the flow of interactions within the MyPPP System, from proposal submission to lifecycle management, including integration with external systems for verification and support.*

# Types of Datasets

- **Project Proposal Dataset:**
  - Contains simulated project proposals from various agencies (ministries, companies). Each proposal should include sections like project objectives, financial model, risk analysis, and compliance with PPP guidelines.
  - **Fields:** Project Title, Submitting Agency, Submission Date, Project Objectives, Financial Model, Risk Analysis, Project Summary, and Status (Submitted, In Review, Rejected, Approved).
- **Historical Proposals Dataset:**
  - Includes previously reviewed project proposals. This data is critical for testing the AI-driven similarity engine.
  - **Fields:** Proposal ID, Project Title, Submitting Agency, Year of Submission, Reviewer Comments, Final Decision (Approved, Rejected, etc.), Similarity Index.
- **Expert Review Dataset:**
  - A dataset of reviewers or evaluators from UKAS. Each expert's specialization can be tied to certain types of project proposals.
  - **Fields:** Reviewer ID, Name, Specialization, Assigned Proposals, Evaluation Comments, Decision Date.

**Compliance Rules Dataset:**

- This dataset should include the compliance rules or guidelines that project proposals must follow. It will be used to validate whether a proposal adheres to UKAS' requirements.
- **Fields:** Rule ID, Rule Description, Rule Category (e.g., Financial, Legal, Risk).

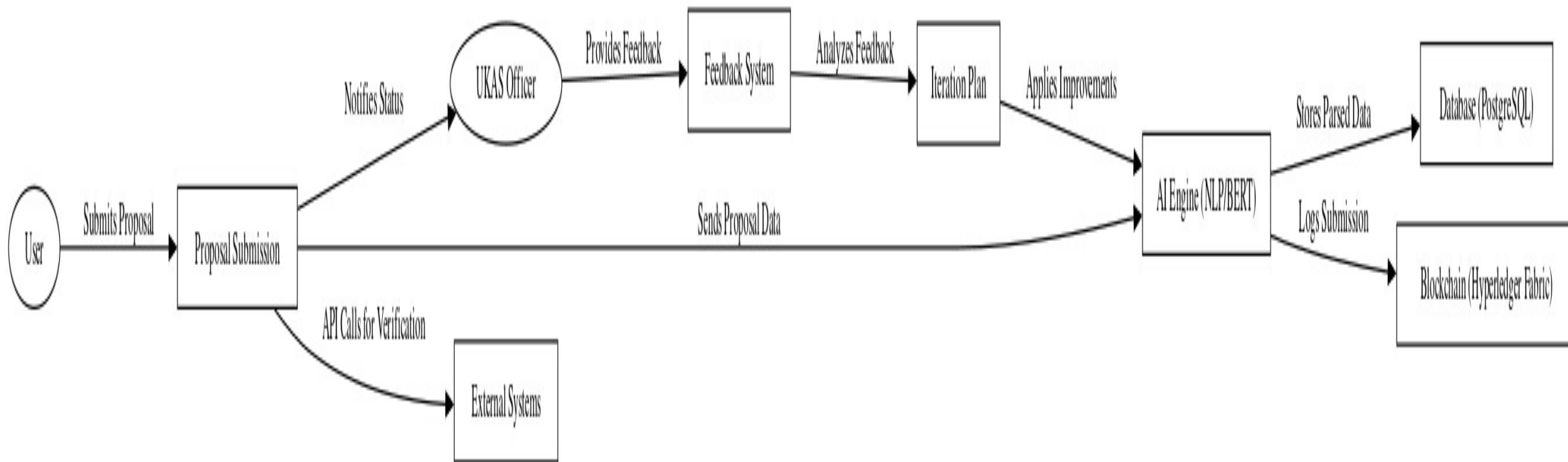
**Proposal Status Tracking Dataset:**

- Tracks the progress of proposals as they move through the review process.
- **Fields:** Proposal ID, Current Status (e.g., In Review, Awaiting Approval), Assigned Reviewer, Date of Last Update.

**Similarity Analysis Dataset:**

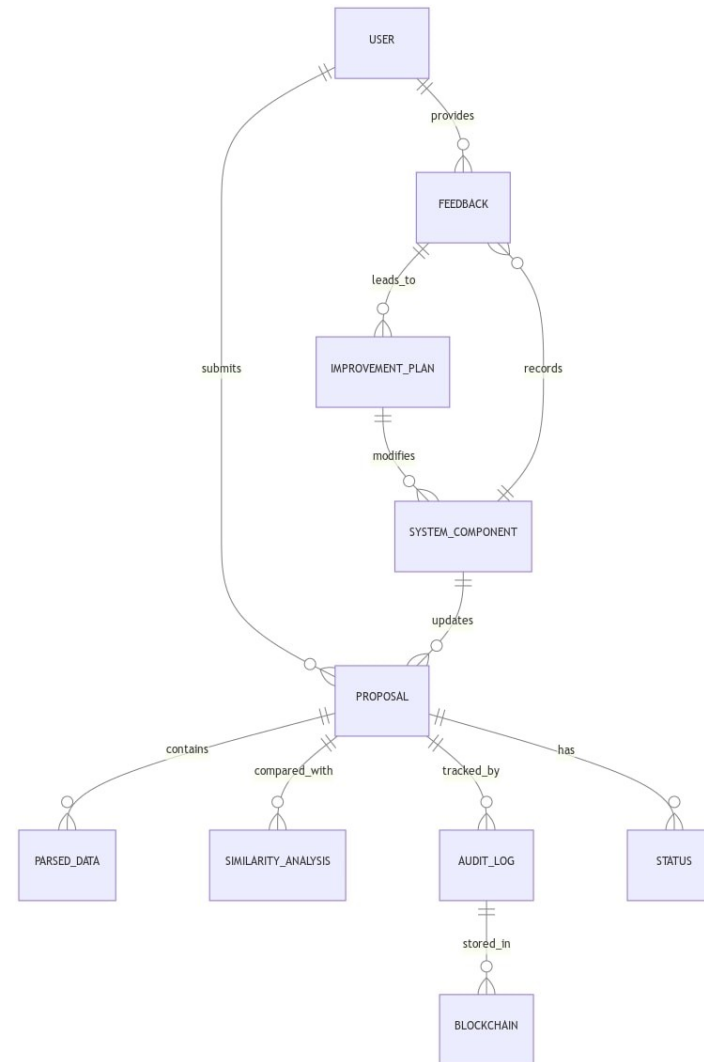
- For the AI-based similarity engine to work, you'll need a dataset containing keywords and proposal themes that the system can use to compare proposals.
- **Fields:** Keywords, Proposal Themes, Similarity Score.

## Data Flow



*This diagram illustrates the flow of data within the MyPPP System during the pilot deployment, showcasing how proposals are submitted, processed, and reviewed, along with the interactions with external systems.*

# Entity Relationship



*This diagram shows the relationships between key entities in the MyPPP System, including proposals, feedback, users, and how feedback leads to iterative improvements in the system.*

# Pilot Deployment Phases

## Phase 1: Preparation and Environment Setup

**Objective:** Set up the necessary infrastructure and environments to ensure the system can run smoothly from initial pilot deployment to eventual scaling.

### 1. Infrastructure Setup

- **Cloud Services:** Provision cloud infrastructure (e.g., AWS, Google Cloud, Azure) to host microservices, databases, and blockchain nodes.

- **Containerization and Orchestration:**

- Use **Docker** to containerize each component (React.js frontend, Django backend, AI engine, PostgreSQL, Hyperledger Fabric).
- Deploy **Kubernetes** clusters to manage containerized services and enable scaling.

**Load Balancers and Scaling Configurations:**

- - Set up load balancers to manage incoming traffic and ensure high availability.
  - Configure Kubernetes for automatic scaling to handle sudden increases in load.



## 2. Development Environment

- Finalize backend (Django) and frontend (React.js) environments for code deployment.
- **AI Engine** (NLP with SpaCy and BERT): Ensure models are pre-trained and deployed within the environment.
- **Database Configuration**: Set up **PostgreSQL** for handling proposal metadata.
- **Blockchain** (Hyperledger Fabric): Set up nodes for the blockchain network to track proposals and create audit trails.

## 3. Security Measures

- Implement basic security layers: HTTPS, encryption for sensitive data (proposal documents, user data).
  - Use **RBAC (Role-Based Access Control)** to manage system user permissions.
  - Enable **logging and monitoring** to track system behavior using Elasticsearch and Kibana dashboards.
-

## Phase 2: Pilot Deployment (Limited Scope)

**Objective:** Test the MyPPP System with a small group of users and ministries, focusing on stability and core functionality.

### 1. Limited User Group

- Select 1-2 ministries or agencies to use the system for submitting, reviewing, and approving PPP project proposals.
- Limit the number of proposals submitted during this phase to ensure manageable load.

### 2. Core Features Activated

- **Proposal Submission:** Ministries will submit proposals via the **React.js frontend**.
- **AI-Based Screening:** The **NLP engine** will parse and screen proposals.
- **Blockchain Logging:** Begin recording proposal submissions and review actions on the **Hyperledger Fabric** blockchain.
- **Dashboard and Status Tracking:** Activate dashboards for real-time tracking of proposal status for UKAS officers.

### 3. Monitoring and Issue Resolution

- Closely monitor system performance (CPU, memory, response times) via **Kibana dashboards**.
- Implement a feedback loop for users and developers to identify bugs, performance issues, and usability concerns.

#### 4. Performance Tuning

- Based on pilot usage data, fine-tune the **AI models** (BERT similarity analysis, K-Means clustering) to improve accuracy and processing speed.
- Optimize database queries, blockchain response times, and Kubernetes autoscaling to improve system responsiveness.

### Phase 3: Full Pilot Expansion

**Objective:** Expand to more ministries and agencies, with an increased number of proposals and system users.

#### 1. Wider Rollout

- Gradually add more agencies and increase the number of proposals being handled by the system.
- Test the system under more complex conditions, such as multiple simultaneous proposals, large submissions, and high concurrent user access.

#### 2. Full Feature Activation

- **Proposal Similarity Analysis:** Enable similarity checks for incoming proposals using BERT.
- **Clustering of Proposals:** Implement K-Means clustering to group similar proposals, making them easier to review.
- **CRM Features:** Activate the proposal lifecycle management and expert reviewer assignment features.

### 3. Stress Testing

- Perform stress testing to simulate large volumes of submissions and review workflows.
- Monitor system stability, and ensure that scaling mechanisms via Kubernetes and load balancing can handle increased loads.

### 4. Advanced Security

- Introduce **multi-factor authentication (MFA)** and additional data encryption for sensitive proposal data.
- Harden API access and blockchain nodes against potential security threats.

## Phase 4: Gradual Scaling and National Rollout

**Objective:** Scale the system to handle national-level submissions, fully implement all services, and ensure seamless integration with existing government systems.

### 1. Horizontal Scaling

- Use Kubernetes to scale out additional services as the load increases. Scale the AI engine, backend services, and the blockchain network independently based on load.
- Employ **PostgreSQL database sharding** to optimize performance for large-scale proposal data.

## 2. Onboarding Additional Agencies

- Gradually bring more government agencies and private companies into the system.
- Provide training sessions and onboarding support to ensure users understand system features and workflows.

## 3. Full Blockchain Integration

- Fully integrate **smart contracts** to automate the submission and review process.
- Enable seamless, tamper-proof audit trails for all proposals.

## 4. Integration with Government Databases

- Start integrating with existing government databases to verify credentials and project details in real-time.
- Set up API connections to pull necessary data for faster verification and project approval processes.

## Phase 5: Continuous Improvement and Future Enhancements

**Objective:** Regularly enhance the system based on user feedback and introduce machine learning models for predictive analysis.

### 1. Data-Driven Improvements

- Use insights from Kibana dashboards to optimize review processes, reduce bottlenecks, and improve system efficiency.
- Continuously enhance the AI models for better proposal screening and classification.

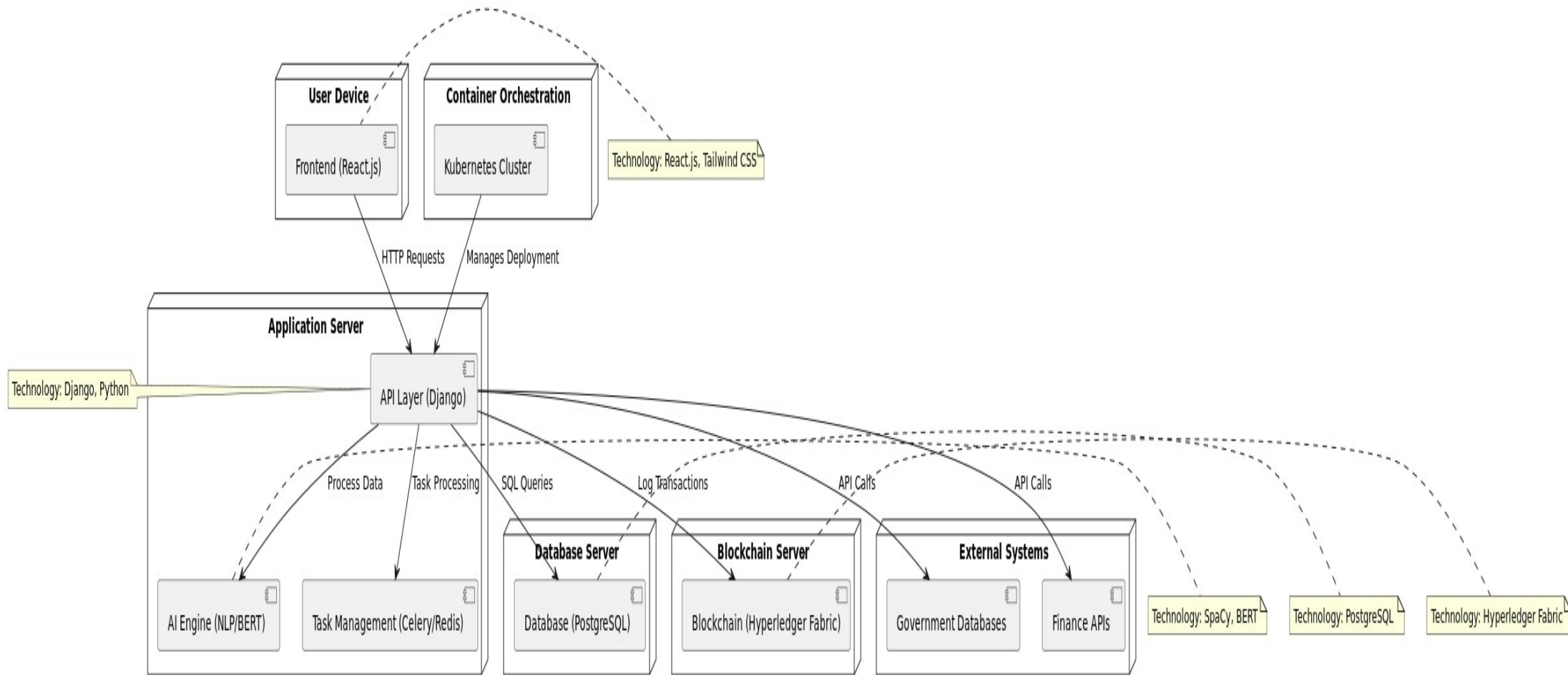
### 2. Automated Decision-Making Models

- Implement **machine learning models** to predict the success or failure of PPP projects based on historical data.
- Automate initial decision-making and recommendations to help UKAS officers make faster, more informed decisions.

### 3. Security and Privacy Enhancements

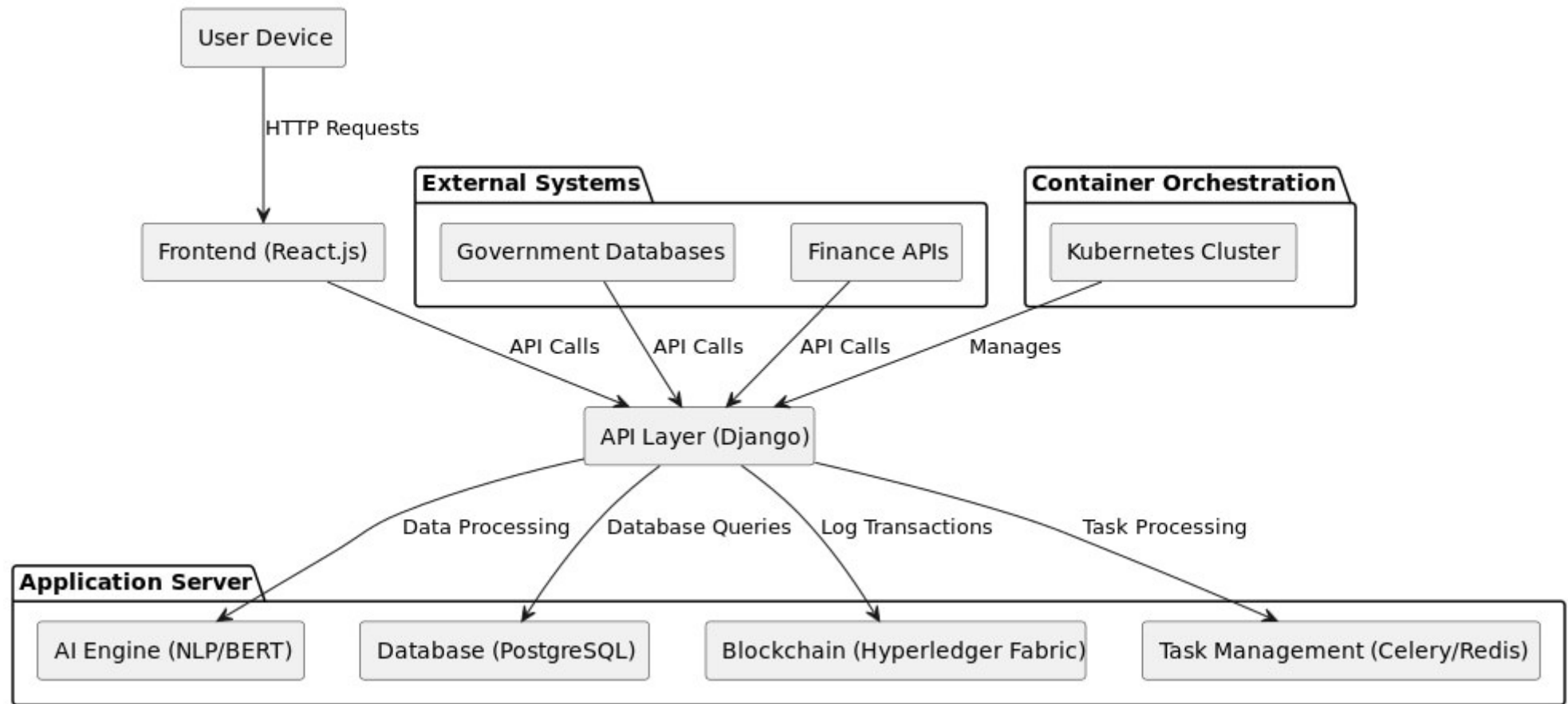
- Introduce end-to-end encryption for sensitive data and further improve access controls.
- Regularly update security protocols to protect against evolving threats.

# Deployment



*This diagram illustrates the deployment architecture of the MyPPP System, detailing how components are distributed across different servers and environments during the pilot and full rollout phases.*

# Network



*This diagram depicts the network architecture of the MyPPP System, illustrating how different components communicate over the network and interact with external services.*



# Integration & Support

## 1. API Access for Real-World Integration

To ensure smooth operation, the **MyPPP System** needs to integrate with various external systems through APIs. These APIs facilitate data exchange, streamline processes, and enable real-time decision-making.

### a. API Types and Their Roles:

#### 1. Government Databases API

- **Purpose:** To verify the credentials of private companies or public agencies submitting proposals.  
**Example:** Integration with a national business registry API for real-time verification of company registration and status.
- **Technical Considerations:**
  - **RESTful API** with secure endpoints for retrieving company details (name, registration number, status).
  - **OAuth 2.0** or other secure authentication protocols for access control.
  - Response caching to minimize API calls for frequently queried data.

#### 2. Finance and Budget APIs

- **Purpose:** To access financial information related to government budgets and compare it against proposed project costs.  
**Example:** Integration with government budgetary APIs to ensure that project proposals fall within budgetary limits.
- **Technical Considerations:**
  - Querying financial records via a **GraphQL API** for more efficient, tailored data retrieval.

- **Automated alerts** when a proposal exceeds the available budget limits set by governmental financial authorities.

### 3. Public Sector Workforce API

- **Purpose:** To cross-reference project proposals with the available workforce in various governmental departments.
- **Example:** Accessing workforce APIs to check if a project proposal requiring specialized skills has the necessary human resources available within the public sector.
- **Technical Considerations:**
  - **Rate-limiting** mechanisms to ensure the system isn't overwhelmed during high-volume API calls.
  - **Data synchronization** to update workforce availability in real-time.

### 4. Blockchain API (Hyperledger Fabric)

- **Purpose:** To ensure a transparent, tamper-proof record of proposal submissions, approvals, and reviews.  
**Example:** UKAS or external agencies need access to view proposal histories, audit trails, and final decisions.
- **Technical Considerations:**
  - **Smart Contracts** as APIs that trigger specific workflows (e.g., when a proposal status changes).
  - Exposing a **Hyperledger Fabric API** endpoint for external applications to fetch tamper-proof logs.

## b. Role-Based API Access

To maintain security and proper data governance, **role-based API access** is essential for UKAS officers, external agencies, and private entities submitting proposals. Different roles might have different API access levels:

- **UKAS Management:** Full access to all system APIs for viewing, approving, and auditing proposals.
- **Review Officers:** Limited API access to submit evaluations and recommendations.
- **Public Agencies and Companies:** Limited access to submission-related APIs to track proposal status and compliance feedback.

### Security Measures:

- OAuth 2.0 or JWT (JSON Web Tokens) for authentication and access control.
- API throttling and rate-limiting to prevent misuse or overloading of the system.

## 2. Database Integrations

Database integrations ensure seamless data flow between the **MyPPP System** and external databases managed by government agencies, facilitating data enrichment, proposal verification, and report generation.

### a. External Databases Integration:

#### 1. National Business Registry Database

- **Purpose:** To verify the legal status, financial health, and credentials of private companies. **Database Type:** Likely to be a centralized relational database (e.g., **PostgreSQL** or **Oracle**).
- **Integration Strategy:**
- - Establish **read-only database connections** to retrieve data without compromising the integrity of the external database.
  - Use **JDBC/ODBC** connectors or **RESTful APIs** for secure data querying.

## 2. Government Financial Systems

- **Purpose:** To check available budgets and financial allocations related to PPP projects.
- **Integration Strategy:**
  - Real-time data queries using **SQL/NoSQL databases**.
  - API-based integration with **ERP systems** or **budgetary tools** used by the government.
  - **Encryption** for financial data transmitted between systems to ensure data security.

## 3. PPP Project History Database

- **Purpose:** To store and reference data from past PPP projects for similarity analysis and AI-powered screening.
- **Integration Strategy:**
- - Importing historical data into **PostgreSQL** or using a **data warehouse** solution (e.g., **Amazon Redshift**) to ensure fast access for AI-driven insights.
  - **Full-text indexing** (using **Elasticsearch**) to speed up proposal similarity searches.

## b. Data Enrichment and Synchronization

- **Real-time Synchronization:** Ensuring data pulled from external systems is always up-to-date by synchronizing regularly through **scheduled tasks** (e.g., via **Celery** background jobs).
- **Data Enrichment:** Using external data sources to enrich internal data. For example, a company's financial health could be enriched with real-time data from the national financial registry.

### Technical Considerations:

- Establish **API-based data synchronization** with real-time push or pull models.
- Use of **ETL (Extract, Transform, Load) pipelines** to aggregate data from multiple sources, ensuring it's available for system-wide usage and reporting.

## 3. Ongoing Support

To ensure long-term functionality, **ongoing support** from UKAS and other governmental bodies will be necessary, especially in the following areas:

### a. System Maintenance and Updates

#### 1. Regular API Updates

- External APIs (e.g., national business registry, financial APIs) may change, requiring the system to adapt quickly.
- **API version management:** Use versioning to prevent breaking changes when external systems are updated.

## 2. Database Maintenance

- Ongoing management of **PostgreSQL** and blockchain databases will be needed for optimization, sharding, and archiving old proposals.
- **Backup and Disaster Recovery:** Implement automated backup solutions to safeguard critical data.

## 3. System Audits and Security Updates

- UKAS or IT service providers will need to regularly audit the system for security vulnerabilities, especially with sensitive proposal and financial data.
- Regular **blockchain audits** to ensure the immutability and integrity of proposal submissions and reviews.

## b. AI and Machine Learning Model Updates

### 1. Ongoing Model Training

- The **NLP and similarity analysis models (SpaCy, BERT)** will need continuous updates with new training data as more proposals are processed.
- UKAS or an IT partner will need to **maintain and retrain** the machine learning models to ensure accurate proposal screening and classification.

### 2. Model Performance Monitoring

- Performance monitoring tools should be in place to track the effectiveness of AI models in processing proposals.
- Implement **AI drift detection** to alert when models need retraining due to outdated data.

## c. User Support and Training

### 1. User Onboarding

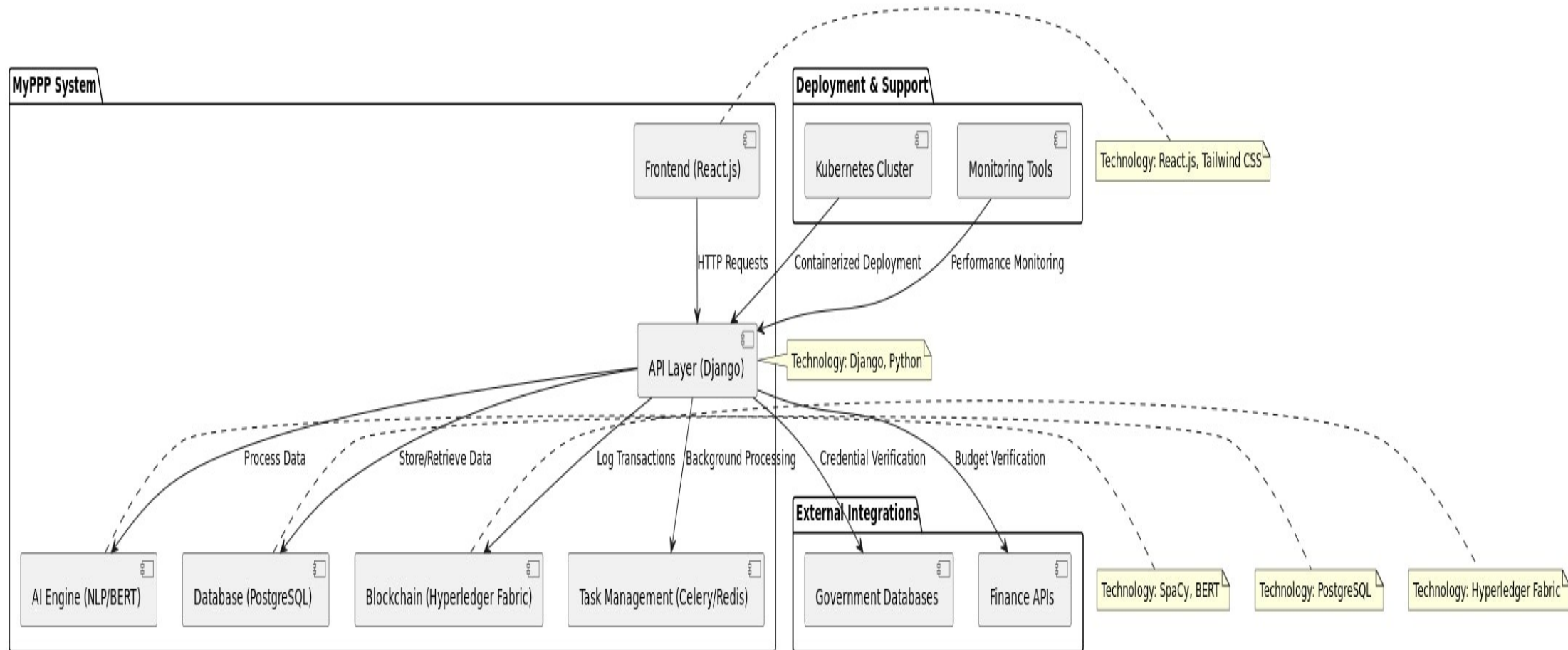
- As more governmental bodies adopt the MyPPP System, training sessions and detailed documentation (API documentation, user manuals) will be essential.
- Provide **API developer portals** for developers to test API endpoints and understand system integration options.

### 2. Helpdesk and IT Support

- Continuous IT support will be needed for resolving issues, managing updates, and troubleshooting API or database integration problems.
- Implement a **ticketing system** for UKAS officers and external users to report issues or request enhancements.

Effective API access, database integration, and ongoing support are critical for the success of the MyPPP System in a real-world governmental setting. **Role-based API access**, secure and seamless **database integrations**, and constant updates to **AI models** and security protocols will ensure the system functions as intended and scales with the growing demands of UKAS and other governmental bodies. Additionally, proactive system maintenance, coupled with strong user support and training, will be essential for long-term adoption and operational efficiency.

# Components



*This diagram outlines the major components of the MyPPP System, including their interactions and dependencies, as well as highlighting the technology stack and integration strategies for deployment.*



# Feedback and Iteration Plan

## Phase 1: Pilot Feedback Collection

The first step after the pilot is collecting structured feedback from all stakeholders involved, such as UKAS officers, external users (ministries, private companies), IT teams, and system administrators.

### 1. Feedback Sources

#### 1. User Surveys and Questionnaires

- **Target Audience:** UKAS officers, ministry personnel, private sector proposal submitters.
- **Survey Focus:**
  - Ease of use of the system's frontend (React.js) for submitting proposals.
  - User experience with the AI-based proposal screening and similarity analysis.
  - Satisfaction with the real-time status tracking and dashboards.
  - Suggestions for features or improvements.
- **Methods:** Distribute digital surveys through email or directly within the MyPPP System's user portal.

#### 2. In-Depth User Interviews

- **Target Audience:** Key users such as decision-makers, proposal reviewers, and technical administrators.
- **Focus Areas:**
  - Detailed insights into the challenges faced while using the system.

- Perceived gaps in the current workflow (e.g., AI screening accuracy, dashboard reporting).
- Usability issues in specific processes (proposal tracking, clustering, etc.).

**Methods:** Conduct video interviews or one-on-one meetings to gather deeper qualitative insights.

### 3. Usage Analytics and System Logs

- **System Monitoring Tools** (Kibana, Elasticsearch):
  - Gather quantitative data on user behavior: system response times, dashboard load times, API call volumes, and proposal throughput.
  - Track **AI performance**: Measure accuracy of the NLP parsing, classification errors, and similarity analysis effectiveness.
- **Error Logs and Performance Monitoring**:
  - Use **log analysis tools** to identify technical issues like API failures, proposal submission errors, and Celery task timeouts.

### 4. IT Team and Developer Feedback

- **Internal Evaluation**:
  - Get feedback from the IT teams managing the system, particularly regarding issues such as server loads, system scalability, and integration with external databases.
- **Developers' Postmortem**:
  - Conduct internal review sessions with the development team to document challenges in the codebase, database, AI engine, and blockchain integrations.

## Phase 2: Feedback Analysis and Prioritization

After collecting feedback, it's crucial to properly analyze and prioritize the improvements needed for the MyPPP System. This phase focuses on categorizing and understanding feedback in a structured way.

### 1. Categorizing Feedback

#### 1. Usability & User Experience

- Problems with UI/UX of the frontend (React.js).
  - Feedback on the proposal submission flow, navigation, and ease of interaction.
- #### 2. Performance & System Efficiency
- Technical issues, including response time, load management, and bottlenecks.
  - AI-related problems, such as inaccuracies in NLP parsing or similarity analysis.
- #### 3. System Functionality Gaps
- Missing features, e.g., more advanced reporting filters, better proposal grouping, etc.
  - Feedback on dashboard views, reporting, or data visualization (Kibana).
- #### 4. Integration & Data Access
- Issues with API integration, such as delays in retrieving data from external systems.
  - Problems related to external database syncing or real-time access to government records.

### 2. Prioritization of Issues

Once categorized, feedback should be prioritized based on **impact** and **feasibility** for improvements. Use the following prioritization method:

### 1. Critical (High Impact, Low Effort):

- Example: Fixing errors in proposal submission or AI screening bugs.
- Prioritize these issues first as they directly impact user experience and are easy to resolve.

### 2. High Impact, High Effort:

- Example: Improving AI model accuracy, restructuring the blockchain system for faster response times.
- These issues are crucial for the system's overall success, but require more resources. Start addressing these right after critical issues.

### 3. Low Impact, Low Effort:

- Example: Minor UI/UX adjustments, like repositioning buttons or updating tooltips.
- Fix these issues after high-priority tasks are completed.

### 4. Low Impact, High Effort:

- Example: Integrating new external databases or redesigning certain workflows.
- These can be scheduled for future releases or major version upgrades.

## 3. Establish a Feedback Loop

- **Feedback Triage System:** Set up an internal feedback triage system where a dedicated team categorizes and prioritizes user feedback weekly.
- **Progress Reports:** Share regular progress updates with stakeholders (UKAS management, key users) on the status of resolved and upcoming changes.

## Phase 3: Iterative Improvements and System Updates

Once feedback has been analyzed and prioritized, the next step is to make the necessary changes and improvements. This phase involves **iterative development cycles** to refine the MyPPP System based on user feedback.

### 1. Iterative Development and Releases

Adopt an **agile development approach** to continuously release updates and improvements based on the prioritized feedback.

#### 1. Sprint Planning (2-4 weeks per sprint)

- At the start of each sprint, select a set of critical issues, bugs, or features to be fixed or improved.
- Involve cross-functional teams: developers, UX designers, AI/ML engineers, blockchain experts.

#### 2. Internal Testing & QA

- After each sprint, thoroughly test the new features or fixes in a **staging environment**.
- Ensure that AI models are re-trained and tested with **new data** to verify improvements.
- Use performance monitoring tools to stress-test any changes to the system infrastructure (Kubernetes scaling, PostgreSQL optimization).

### 3. User Acceptance Testing (UAT)

- Allow a small set of pilot users to test the improved system during each sprint to gather early feedback.
- Focus on verifying that UI changes are intuitive, AI improvements deliver more accurate results, and performance is enhanced.

### 4. Incremental Rollout

- After successful testing, incrementally roll out the updates to the live system.
- Gradually release features or updates to a broader audience to minimize disruptions and allow for rollback if issues arise.

## 2. Continuous AI Model Training and Optimization

As part of the iterative improvement, focus on continuously optimizing the AI models:

### 1. New Data Integration

- Use new proposal data and feedback to **retrain NLP models** (SpaCy) and the **BERT-based similarity analysis**.
- Continuously improve **proposal classification accuracy** and reduce false positives.

### 2. AI Monitoring

- Implement AI monitoring tools to track the model's real-time performance. For example, flagging incorrect classifications or similarity scores will alert the IT team when the model is drifting.

### 3. Model Version Control

- Use **version control for AI models** to easily switch between model versions and roll back if new model versions underperform.

### 3. Infrastructure and Scalability Enhancements

Post-pilot improvements should also focus on scaling the system infrastructure to support a larger user base.

#### 1. Database Sharding and Optimization

- As proposal data grows, consider implementing **PostgreSQL sharding** to optimize database performance.  
Apply caching strategies to frequently accessed data for faster retrieval.

- 

#### 2. Kubernetes Auto-scaling Enhancements

- Refine the auto-scaling policies for Kubernetes clusters based on the pilot feedback and analytics (CPU usage, memory, request throughput).
- Ensure the **Celery + Redis** task queue is optimized for handling a larger number of proposals, especially for background AI tasks.

#### 3. Blockchain Scalability

- Enhance the performance of the **Hyperledger Fabric blockchain**, focusing on faster transaction times and response for proposal submission logging.
- Consider creating **lightweight blockchain nodes** for agencies requiring quicker access.

## Phase 4: Statewide and Nationwide Rollout

Once the system has been iteratively improved based on the feedback and technical fixes, it will be ready for a broader deployment across the state or nationwide.

### 1. Phased Rollout

#### 1. Statewide Rollout

- Start with a **region-by-region deployment** within a state. Onboard more ministries and public agencies gradually. Ensure each region has **dedicated support teams** to handle user inquiries, system issues, and further feedback collection.
- 

#### 2. Nationwide Rollout

- Once the statewide rollout is successful and stable, plan for a **nationwide rollout**. Use the same phased approach but allow for real-time data integration with national databases and systems.
-



## 2. Ongoing Feedback and Improvement Cycle

Even after the full rollout, it is crucial to maintain a feedback and improvement cycle:

- Continue **monitoring usage patterns** and collecting feedback.
- Use **analytics tools** (Kibana, Elasticsearch) to continuously improve system performance, AI accuracy, and user experience.
- Regularly release updates and new features based on ongoing feedback.

This **Feedback and Iteration Plan** ensures that the MyPPP System evolves based on real-world user experiences and technical insights gathered from the pilot phase. With a clear path for gathering feedback, analyzing it, and prioritizing improvements, the system can be iteratively enhanced, leading to a successful statewide and nationwide rollout. Continuous user feedback and technical iteration will ensure the MyPPP System remains adaptable, scalable, and functional as it expands to meet the growing demands of public-private partnerships.

# Team Introduction

Our team comprises highly skilled professionals with profound expertise in cutting-edge technologies like blockchain, artificial intelligence, and decentralized systems. Drawing from prestigious technical backgrounds and years of experience, each member brings a wealth of knowledge and practical solutions that align with our mission of innovation and technological advancement.

## Swastik — Full Stack Developer and Programmer

With five years of experience in full stack development, Swastik is passionate about blockchain technology and transparent systems. His proficiency spans multiple programming languages, including Solidity, Rust, Golang, Vyper, Python, JavaScript, TypeScript, Ruby, PHP, C, and C++. Swastik has consistently focused on building secure and efficient decentralized applications, particularly those that leverage the Ethereum Virtual Machine (EVM) and Layer 2 solutions.

He has successfully developed a blockchain-based HR payroll management system to streamline payment processes via decentralized platforms, and has pioneered **NFTInu**, a comprehensive NFT rental service. Currently, he is leading the development of **nftxau.com**, an innovative phygital blockchain solution that links digital blockchain assets with tangible real-world items using advanced database systems. Swastik's unique blend of technical acumen and vision positions him as a driving force in bringing blockchain solutions to real-world challenges.

## Navinder — Project Manager and DevOps

Navinder brings over 21 years of extensive experience as a Solution Architect, specializing in Information and Computer Science. His expertise spans across ICT, artificial intelligence (AI), artificial general intelligence (AGI), Internet of Things (IoT), and embedded systems engineering. A seasoned professional in the design, development, and management of complex IT systems, Navinder excels in integrating diverse technologies into cohesive, scalable solutions.

Currently, his focus lies in Data Science and blockchain programming, utilizing Python and R to drive innovations in machine learning, deep learning, neural networks, and algorithmic sciences. His role as a Blockchain advisor includes smart contract-based fintech development, where he leverages his expertise in high-level programming, backend development, and CI/CD pipelines.

Navinder's command over ICT infrastructure, cybersecurity, and information systems, combined with his project management skills, ensures the delivery of reliable, high-quality solutions. He serves as a Subject Matter Expert, ensuring that all aspects of project delivery meet the highest standards of technical excellence.

Together, Swastik and Navinder form a dynamic team dedicated to pushing the boundaries of what is possible in blockchain, data science, and decentralized applications, ensuring that our projects are not only technically robust but also innovative and aligned with the future of digital ecosystems.

## References

### 1. MyPPP System - Open Source Implementation Document

Public-Private Partnership (PPP) project proposal management solution for UKAS, Malaysia. This document details the technical architecture, functionalities, and implementation roadmap for the MyPPP System.

**Published by:** Internal Development Team

**Technologies Used:** Python/Django, React.js, PostgreSQL, SpaCy, BERT, Hyperledger Fabric, Celery, Redis, Elasticsearch, Kibana

**Date of Publication:** September 2024

## 2. SpaCy

An open-source library for advanced Natural Language Processing (NLP). SpaCy is used in the MyPPP System to parse and extract data from PPP project proposals, enabling automated screening checking.

**Website:**

## 3. BERT (Bidirectional Encoder Representations from Transformers)

A deep learning model for NLP, utilized for similarity analysis in the MyPPP System. BERT compares new project proposals with past submissions to identify commonalities and improve review efficiency.

**Website:** [https://huggingface.co/transformers/model\\_doc/bert.html](https://huggingface.co/transformers/model_doc/bert.html)

## 4. Hyperledger Fabric

An enterprise-grade, open-source blockchain framework used in the MyPPP System to track all proposal-related actions, ensuring transparency and creating a tamper-proof audit trail.

**Website:** <https://www.hyperledger.org/use/fabric>

## 5. PostgreSQL

An open-source relational database used to store project proposal data, user information, and audit logs in the MyPPP System, ensuring data consistency and scalability.

**Website:**

## 6. Elasticsearch and Kibana

Elasticsearch is a search engine used in the MyPPP System to facilitate full-text searches and data filtering, while Kibana provides a data visualization dashboard to monitor proposal trends and key metrics.

**Website:**

## 7. Docker and Kubernetes

Docker is used for containerizing the MyPPP System components to ensure consistency across environments, while Kubernetes is employed for orchestrating and scaling these containers as needed.

**Website:** and

# **MyPPP System**

**AI & Blockchain-Driven Platform for Public-Private Partnership Proposal  
Management**

**By:  
Swastik  
&  
Navinder**

**29<sup>th</sup> September 2024**

# Introduction

The MyPPP System is an AI-driven platform that streamlines the management of Public-Private Partnership (PPP) project proposals in Malaysia. By automating submissions and utilizing Natural Language Processing (NLP) for compliance checks, along with blockchain for secure records, it enhances efficiency and transparency, enabling UKAS officers to make informed decisions quickly.

# Team Introduction

- Our team is a powerhouse of innovation and technical expertise, with deep roots in cutting-edge fields like blockchain, artificial intelligence, and decentralized systems.
- **Swastik**, a blockchain visionary with five years of full-stack development experience, has created groundbreaking decentralized applications, including NFT platforms and phygital blockchain solutions that seamlessly connect the digital and physical worlds.
- **Navinder**, with over 21 years of experience as a solution architect, is a master of integrating advanced AI, IoT, and data science technologies into scalable, transformative solutions.
- Together, we are the driving force behind a future-focused digital ecosystem, blending innovation with technical excellence. Our proven track record and passion for pushing the limits of what's possible make us the perfect team to spearhead this transformative project.

# Problem

## Manual PPP Proposal Handling Creates Inefficiencies

The current manual process of handling Public-Private Partnership (PPP) project proposals in Malaysia is slow and inefficient.

UKAS officers spend too much time reviewing lengthy proposals and searching for relevant past projects, leading to delays in decision-making and lost opportunities.

### Key Challenges:

- ➔ No automated system for initial proposal screening.
- ➔ Lack of transparency and tamper-proof tracking.
- ➔ Difficulty finding similar past proposals manually.



# Our Solution

## Introducing MyPPP System

- ➔ A fully automated platform that uses AI and blockchain to streamline the PPP proposal process.
- ➔ Automates submissions, screening, and tracking, reducing the workload for UKAS officers.
- ➔ Ensures full transparency using blockchain technology for tamper-proof record keeping.

# Core Functionalities

## 1. AI-Powered Proposal Screening

- ➔ **NLP** (Natural Language Processing) automatically parses proposals and checks them against predefined PPP criteria.

## 2. Similarity Analysis

- ➔ **AI** compares new proposals with past submissions, identifying similarities in risk models, financials, and objectives using deep learning (**BERT**).

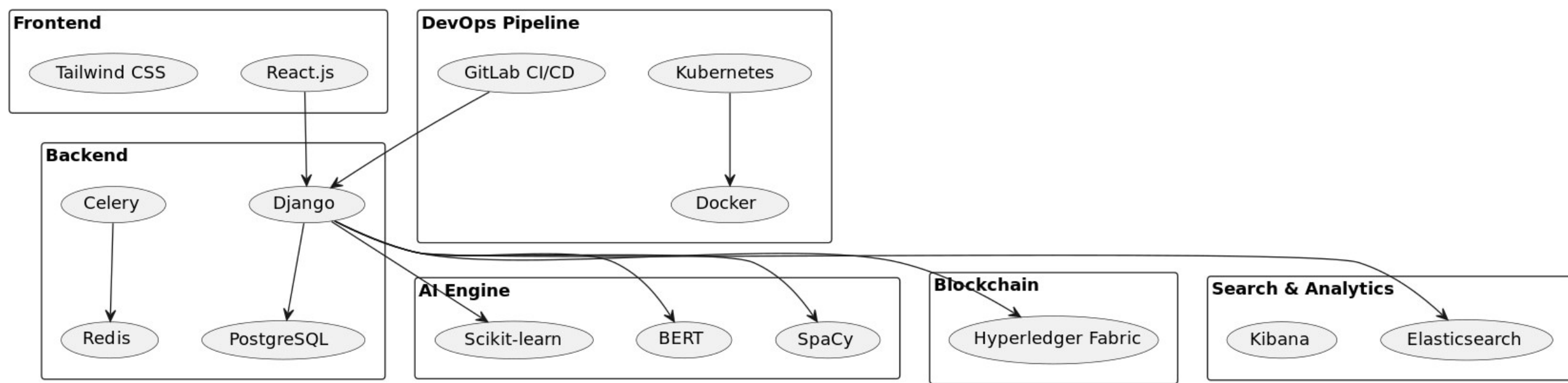
## 3. Blockchain Integration

- ➔ Every step of the proposal lifecycle is recorded on a tamper-proof **blockchain**, ensuring transparency and auditability.

## 4. CRM Lifecycle Management

- ➔ Tracks each proposal from submission to final decision, with real-time updates, alerts, and routing to the appropriate experts.

# Solution Architecture

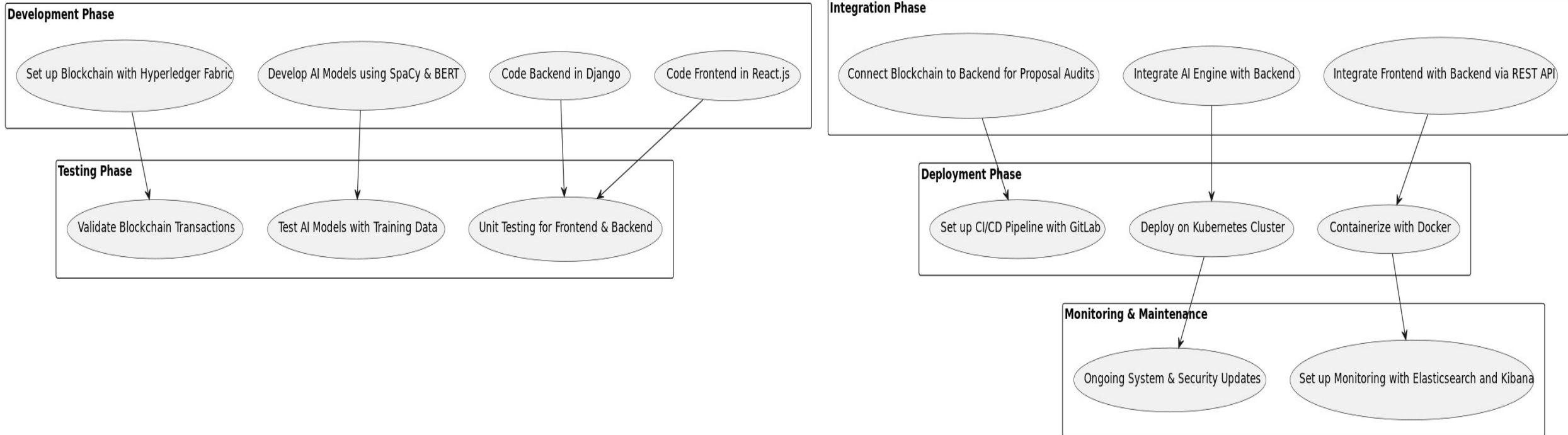


This diagram visually represents the **complete solution architecture** of the MyPPP System

# Technical Architecture

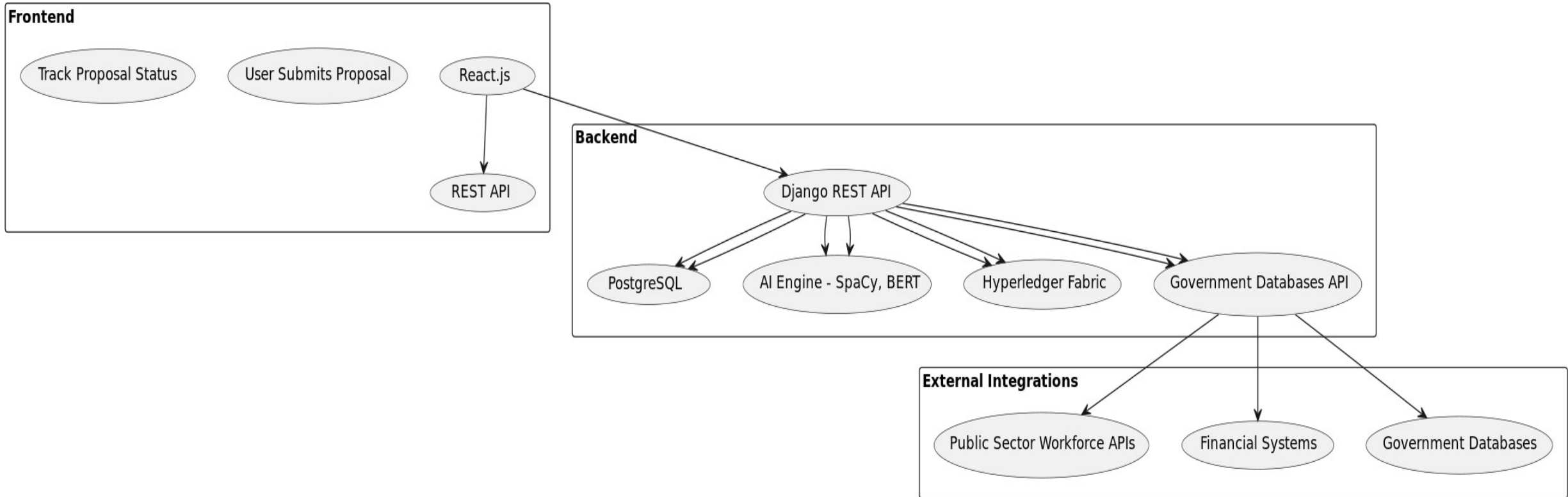
- ➔ Frontend (**React.js**): Provides the interface for submitting and tracking proposals.
- ➔ Backend (**Django**): Manages API requests, business logic, and data processing.
- ➔ AI Engine: NLP (**SpaCy**) for parsing, and BERT for similarity analysis.
- ➔ Blockchain (**Hyperledger Fabric**): Ensures immutable records of proposal submissions and decisions.
- ➔ Task Management (**Celery** + **Redis**): Handles background tasks like similarity analysis and notifications.
- ➔ Database (**PostgreSQL**): Stores all structured proposal data.

# Function Implementation



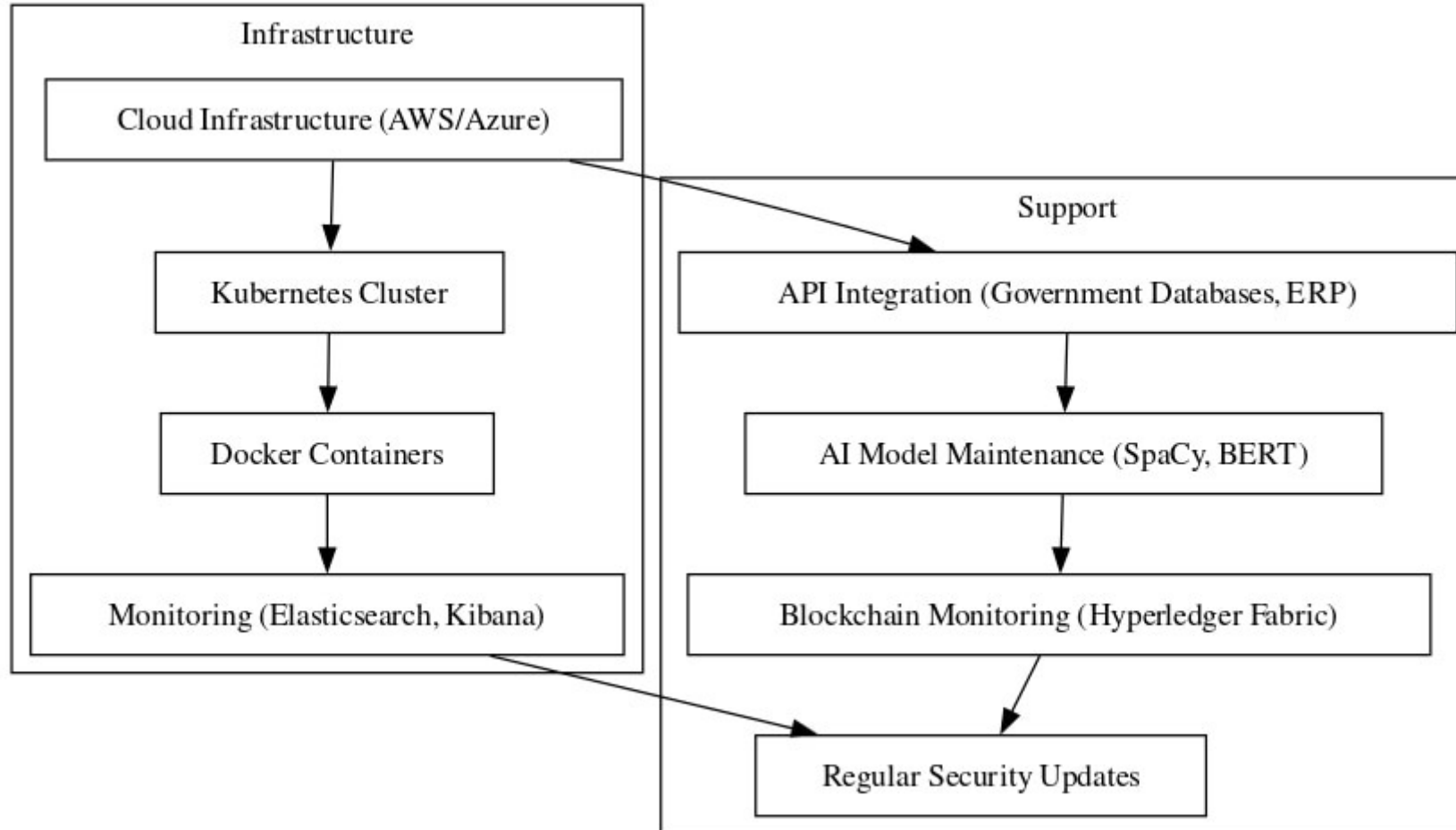
*This diagram illustrates the step-by-step implementation phases of the MyPPP System, from development and testing through integration and deployment, followed by monitoring and continuous maintenance.*

# Integration Solution



*This diagram shows how the frontend and backend of the MyPPP System integrate through REST APIs, connecting to databases, AI engines, blockchain, and external government systems for seamless functionality.*

# Infrastructure & Support



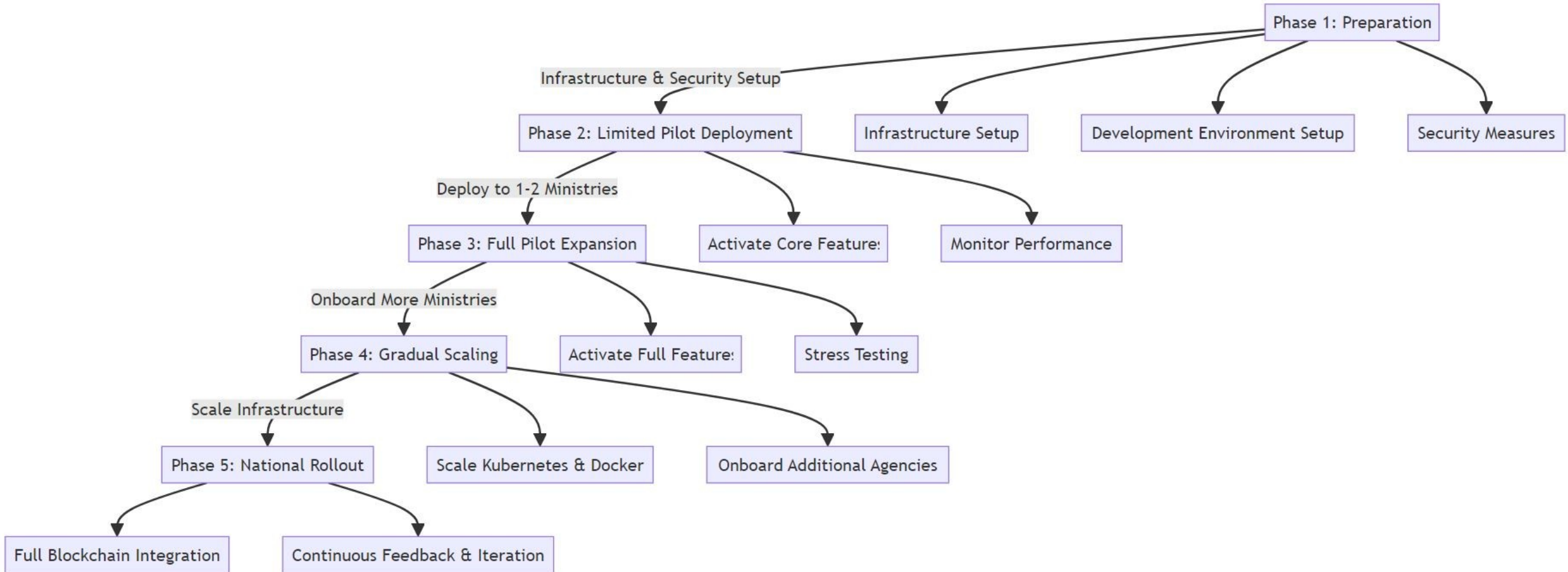
*This diagram shows the infrastructure and support components post-development for the MyPPP System, detailing how cloud services, Kubernetes, Docker, API integrations, AI model maintenance, blockchain monitoring, and security updates are structured and interconnected.*

# Pilot Deployment Plan

1. **Phase 1:** Limited deployment to 1–2 districts to collect initial user feedback and optimize the AI models.
2. **Phase 2:** Gradual rollout across more districts based on the success of the pilot.
3. **Phase 3:** Full statewide rollout with continuous monitoring and system adjustments.



# Pilot Deployment Plan



*This diagram details the complete pilot deployment phases for the MyPPP System, from infrastructure setup through national rollout, focusing on gradual expansion and scaling.*

# Why MyPPP System ?

- **Efficiency:** Automates repetitive tasks, speeding up the review process.
- **Transparency:** Blockchain ensures every action is transparent and recorded.
- **Scalability:** Built on a microservices architecture, the system can scale as it expands nationwide.
- **AI-Powered:** The system intelligently routes proposals, compares them to past projects, and performs compliance checks automatically.

# Final Thoughts

- ➔ MyPPP System addresses the core challenges of PPP proposal management in Malaysia, delivering a scalable, transparent, and efficient solution for UKAS and other stakeholders.
- ➔ By leveraging AI, blockchain, and CRM integration, the system streamlines the entire process, reducing the manual workload and speeding up decision-making.

# Other Informations for Judges References

## Prototype System Demo

Access the system dashboard here:

<https://myppp.onrender.com>

- User:
  - Username : user
  - Password : pass
- Admin:
  - Username : admin
  - Password : pass

## Youtube Video

Video explainer link:

<https://youtu.be/XDFIVkq67SA>

# Q&A Session

(Q&A time)

# Thank You

# MyPPP System

## AI-Driven Public-Private Partnership Proposal Management System

MyPPP System is a comprehensive AI and blockchain-powered platform designed to manage Public-Private Partnership (PPP) project proposals, automating submissions, ensuring transparency, and streamlining the entire lifecycle of project evaluations for the Public-Private Partnership Unit (UKAS).

### Team Introduction

Our team comprises professionals with extensive experience in artificial intelligence (AI), blockchain, CRM development, and public sector digitalization projects. We have implemented systems that combine advanced technologies like Natural Language Processing (NLP), deep learning, and blockchain to automate and secure complex workflows in various industries. Our developers specialize in backend development (Python/Django), frontend development (React.js), AI/ML integration (SpaCy, BERT), and blockchain solutions (Hyperledger Fabric). With a history of successfully delivering scalable, transparent, and innovative solutions, we are equipped to handle the technical and operational needs of the MyPPP System.

### Problem

In Malaysia, the process of managing PPP project proposals is manual, time-consuming, and prone to inefficiencies. UKAS officers must sift through complex technical documents, assess proposal compliance with predefined criteria, and locate similar proposals from the past, all of which demand significant time and human resources. The absence of a digital system to automate initial proposal screenings and streamline decision-making creates bottlenecks, delays project approvals, and increases the risk of overlooked viable projects.

Current challenges include:

- **Manual proposal evaluation:** UKAS officers manually review each proposal, leading to delays and inconsistent evaluations.
- **No automated compliance checks:** Proposals are not screened for completeness or compliance before review.
- **Limited transparency:** There is no centralized tracking system that guarantees the transparency of the review process.
- **Difficulty locating similar past proposals:** Without an AI-driven search, officers struggle to find and compare similar projects.

## The Solution

The **MyPPP System** solves these problems through:

1. **Automated proposal screening:** Using NLP and machine learning, the system automatically parses and evaluates proposals for compliance with PPP criteria.
2. **Blockchain-based transparency:** Every key action in the proposal review process is recorded on a tamper-proof blockchain, ensuring transparency and auditability.
3. **AI-powered similarity analysis:** The system compares new proposals with historical submissions to identify similarities, making it easier for officers to reference past evaluations.
4. **Comprehensive CRM integration:** Manages the entire lifecycle of a project, from submission through review and decision-making, with real-time tracking and automated notifications.

## Core Advantages:

- **Efficiency:** Automates the most time-consuming tasks, reducing the burden on officers.
- **Transparency:** Immutable blockchain records ensure accountability at every step of the process.
- **Scalability:** Built using microservices architecture, the system can scale to handle increasing proposal volumes as it is deployed nationwide.



# Functionalities

The **MyPPP System** offers a range of advanced functionalities to improve the PPP proposal management process:

## 1. Proposal Submission and AI-Powered Initial Screening

- Proposals are submitted through a user-friendly web interface built using **React.js**. Users can upload documents in formats like PDF and DOCX, which are automatically parsed by the system's **AI engine**.
  - NLP Parsing** (via **SpaCy**) extracts key sections from the proposals, such as project objectives, financial models, risk management strategies, and timelines.
- The system runs a **compliance check** to ensure the proposal meets the predefined PPP criteria. This includes verifying the project structure, assessing financial models, and ensuring the proposal addresses risk-sharing mechanisms.

## 2. AI-Based Similarity Analysis

- After parsing, the system leverages **BERT**, a state-of-the-art deep learning model, to perform contextual similarity analysis. This functionality helps UKAS officers by identifying proposals that share similarities with previously evaluated projects. For example, if a proposal has a similar risk model to a past submission, the system will flag this and present it to officers.
- Clustering** algorithms group proposals based on their similarities, allowing officers to quickly locate past evaluations that are relevant to the current project, reducing manual research efforts.

## 3. CRM Lifecycle Management

- The **CRM system** manages each proposal's journey through the submission, screening, review, and decision-making process.
  - It provides a **real-time dashboard** where UKAS officers can view the status of all proposals: pending, in review, accepted, or rejected.
  - The CRM also features **automated routing**, which assigns proposals to the most appropriate experts based on content and workload, ensuring efficient and balanced task distribution.
- Notifications and Alerts** are sent to users and officers at key stages in the proposal process, such as when a proposal requires additional information or when a decision has been made.

#### 4. Blockchain-Based Transparency and Accountability

- Every action related to a proposal, from submission to final decision, is recorded on the **Hyperledger Fabric blockchain**. This ensures a fully transparent, tamper-proof record of every proposal's lifecycle.
  - The system automatically logs submission times, reviews, decisions, and status updates, creating an **immutable audit trail** that can be referenced during evaluations or disputes.
- Blockchain integration also ensures that UKAS officers can demonstrate the integrity of the review process to all stakeholders, including external auditors and public interest groups.

#### 5. Advanced Reporting and Data Visualization

- The system integrates with **Kibana**, allowing UKAS management to visualize key data points, such as proposal submission trends, time-to-decision statistics, and proposal success rates by category.
  - Officers can generate **custom reports** that provide insights into bottlenecks in the review process, allowing for continuous improvement of operations.
- Predictive analytics** powered by machine learning models can forecast future trends in PPP project submissions, helping UKAS better allocate resources and anticipate workload spikes.

## Solution Architecture

The MyPPP System's architecture is built for scalability, security, and performance:

1. **Frontend (React.js)**: Provides an intuitive user interface for submitting proposals and tracking their status. This includes both a user-facing portal for ministries and agencies and an administrative dashboard for UKAS officers.
2. **Backend (Django)**: The core API service handles proposal submission, AI processing, and blockchain integration. The **Django** framework provides the business logic for managing proposals and executing workflows.

### 3. AI Engine (SpaCy and BERT):

- **SpaCy** parses and extracts relevant data from unstructured documents.
- **BERT** compares proposals against previously submitted projects to identify similar proposals and group them for easy reference.

### 4. Blockchain (Hyperledger Fabric):

- Tracks all key proposal lifecycle events and ensures the immutability of the data.
- Each proposal's submission, review, and decision is recorded on the blockchain, ensuring a transparent audit trail.

### 5. Task Management (Celery + Redis):

- Handles long-running tasks like similarity analysis, notifications, and background data processing. Redis ensures low-latency task queuing, while **Celery** manages task distribution across the system.

### 6. Database (PostgreSQL):

- Stores structured data such as user information, proposal metadata, review logs, and decision outcomes.
- Supports the fast retrieval of project proposals and historical records for similarity analysis and reporting.

## Solution Technology for Implementation

The MyPPP System leverages the following technologies:

- **Backend:** Python/Django for developing scalable APIs and managing business logic.
- **Database:** PostgreSQL for storing structured proposal data, decision histories, and audit trails. **AI Engine:** SpaCy (NLP parsing) and BERT (deep learning) for proposal analysis and similarity detection.
- **Frontend:** React.js and Tailwind CSS for an interactive and responsive user experience.
- **Blockchain:** Hyperledger Fabric to provide a tamper-proof, decentralized ledger of proposal transactions.
- **Task Queue:** Celery and Redis to handle background tasks and asynchronous processing efficiently.
- **Search Engine:** Elasticsearch for full-text search capabilities, enabling fast lookup of proposals.

## Integration & Support

The MyPPP System is designed to integrate seamlessly with existing government databases and systems:

- **Government Database Integration:** Through APIs, the system connects to external databases to verify the credentials of submitting agencies and project details.
- **CRM Integration:** The system is built around a robust CRM that manages the entire lifecycle of proposals, offering detailed tracking and reporting.
- **Blockchain Integration:** The blockchain API allows external auditors or stakeholders to access transparent and tamper-proof proposal logs.

For long-term support:

- **API maintenance:** Regular updates to ensure compatibility with government systems.
- **AI model updates:** Continuous learning based on new proposal data to improve accuracy.
- **User support and training:** Includes technical documentation, training sessions, and ongoing helpdesk support.

## Pilot Deployment Plan

1. **Phase 1:** Initial deployment in selected districts, where ministries submit PPP proposals through the system. Feedback will be collected to fine-tune AI models and user experience.
2. **Phase 2:** Gradual expansion to more districts, with full integration of all functionalities.

3. **Phase 3:** System performance will be closely monitored, and adjustments will be made based on real-world data.

## Statewide Rollout Plan

After successful pilot implementation, the system will expand across the state, scaling the backend and blockchain infrastructure to handle larger volumes. Stakeholders will receive training, and the feedback loop will ensure continuous improvement.

## Nationwide Rollout Plan

Upon statewide success, the system will be rolled out nationally, ensuring all relevant ministries and private partners have access to the platform. The national deployment will involve refining the AI models and ensuring seamless integration with all government databases.

## Other Information for Judges

- **Live Demonstration:** A live system demo is available, showcasing the proposal submission process, AI-powered compliance checks, and blockchain-based transparency.
- **Visual Dashboards:** A live demo of web -based dashboards for tracking proposal status and trends.

---

### *Prototype System Demo*

*Access the system dashboard here:*

<https://myppp.onrender.com>

- User:
  - Username : user
  - Password : pass
- Admin:
  - Username : admin
  - Password : pass

*Video explainer link:*

<https://youtu.be/XDFIVkq67SA>