BCSE497J Project-I

ZERO-TRUST CLOUD COLLABORATION PLATFORM

22BCE0465	**Swarit Jain**
22BCT0307	**Akshay Agarwal**
22BDS0110	**Shivansh Srivastava**

Under the Supervision of

Yokesh Babu S

Associate Professor Grade 1

School of Computer Science and Engineering (SCOPE)

B.Tech.

in

Computer Science and Engineering

School of Computer Science and Engineering
(Core, IOT, Data Science)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

September 2025

ABSTRACT

The proposed "Zero-Trust Cloud Collaboration Platform" is designed as a modern framework to ensure secure, privacy-first collaboration in cloud environments through the integration of three intelligent modules: Contextual PII Analysis, Adaptive Zero-Trust Policy Engine, and Behavioral Anomaly Detection. The solution addresses critical security needs in decentralized infrastructures by leveraging advanced authentication, microservice-based architecture, and machine learning-powered analytics. The first phase hardens the backend with containerized services, robust APIs, and passwordless WebAuthn authentication. Next, an NLP-driven module—using tools like Hugging Face Transformers—classifies the sensitivity of uploaded files based on detected PII, ensuring that data risk is contextually understood and handled asynchronously. The heart of the system—the Zero-Trust Policy Engine—enforces access controls informed by user roles, data sensitivity, and network trust, with context-aware decision-making for every access request. Policies are enforced dynamically at the middleware level, safeguarding data in real time. Behavioral Anomaly Detection further strengthens security by tracking user activities, identifying irregular or malicious patterns using Isolation Forest, and enabling rapid threat response. The final dashboard provides transparent, actionable insights into policy decisions and security threats. By combining cutting-edge NLP, adaptive access control, anomaly detection, and intuitive visualization, this project advances the state-of-the-art in cloud collaboration security—ensuring regulatory compliance, operational resilience, and granular, dynamic protection against emerging cyber threats

TABLE OF CONTENTS

INTRODUCTION

BACKGROUND

Modern cloud collaboration platforms present unparalleled flexibility but introduce significant risks related to data privacy, unauthorized access, and insider threats as the traditional network perimeter dissolves. Zero Trust Security (ZTS) has emerged as a foundational alternative, adopting a “never trust, always verify” mindset that is especially suited for cloud architectures where users and data are decentralized.

MOTIVATION

The exponential increase in sensitive data shared online, stringent compliance regulations (e.g., GDPR), and the evolving threat landscape demand a security paradigm capable of providing continuous verification, context-aware protection, and rapid incident response. Existing solutions often rely on static policies and surface-level PII detection, leaving organizations exposed to sophisticated threats and regulatory penalties

SCOPE OF THE PROJECT

This capstone targets the full security lifecycle within a cloud collaboration context—from advanced authentication and data sensitivity classification to adaptive access control and anomaly detection. It emphasizes modular backend architecture, real-time middleware enforcement, and the integration of machine learning for threat detection and decision-making

PROJECT DESCRIPTION AND GOALS

LITERATURE REVIEW

- **Zero Trust Architecture: A Systematic Literature Review** – Explores adaptive access control and contextual decision-making in cloud environments.
  https://arxiv.org/html/2503.11659v1

- **Implementing Zero Trust Security Models in Cloud Computing** – Reviews zero trust deployment strategies and operational challenges in cloud
  https://wjarr.com/sites/default/files/WJARR-2024-3500.pdf

- **Redefining Zero Trust Architecture in Cloud Networks** – Discusses dynamic, granular, and context-aware policy enforcement for cloud security.
  https://magnascientiapub.com/journals/msarr/sites/default/files/MSARR-2021-0032.pdf

- **Comparing Feature-based and Context-aware Approaches to PII Generalization** – Evaluates context-aware models for accurate PII detection https://arxiv.org/html/2407.02837v1

- **Anomaly Detection in Cloud Computing** – Systematic review of machine learning techniques for proactive threat identification in the cloud https://ijsrset.com/index.php/home/article/download/IJSRSET2512402/IJSRSET2512402/1260

GAPS IDENTIFIED

- Static access policies fail to adapt to evolving user contexts, roles, and environments.
- Surface-level PII detection overlooks nuanced, context-sensitive risks.
- Lack of real-time workflow integration for policy enforcement and behavioral anomaly detection limits proactive security.

OBJECTIVES

- Build a modular, microservice-ready cloud platform that ensures continuous, context-sensitive verification.
- Implement automated, context-aware NLP-based PII and sensitivity classification for uploaded files.
- Develop an adaptive Zero-Trust Policy Engine for informed access decisions based on multiple dynamic factors.
- Integrate behavioral anomaly detection using ML to spot and respond to insider and external threats in real time.
- Provide a demo-ready dashboard for live visualization of access logs and security alerts.

PROBLEM STATEMENT

Traditional security models for cloud collaboration are insufficient to address emerging threats because they lack context-aware data classification, adaptive policy enforcement, and proactive anomaly detection—resulting in increased risks of data breach, non-compliance, and operational disruptions.

PROJECT PLAN

Week 1: Hardened Core & Advanced Authentication (Approx. 28 hours)

- **Microservice-Ready Architecture:**
  - Design the backend using containerization (Docker Compose).
  - Configure Python API, PostgreSQL, and Redis as separate services for scalability and maintainability.
- **Advanced Authentication:**
  - Implement passwordless WebAuthn authentication to replace traditional username/password systems.
  - Integrate registration and login workflows using public key cryptography,
- **Core File Management APIs:**
  - Build essential APIs: upload, list, download, delete.
  - Add error handling and strict user ownership check for data access.

Week 2: The Intelligent Data Classifier (Approx. 28 hours)

- **NLP Engine:**
  - Use Hugging Face Transformers (DistilBERT or RoBERTa) for extracting, flagging, and classifying PII in files.
  - Potentially fine-tune models for dataset relevance.
- **Automated Data Classification:**
  - Assign Data Sensitivity Levels (Public, Internal, Confidential, Restricted) based on model output.
  - Store sensitivity metadata in the database for each file.
- **Asynchronous Processing:**
  - Implement background processing using Celery for file classification, ensuring a responsive frontend even during heavy analysis.

Week 3: Zero-Trust Policy Engine (Approx. 32 hours)

- **Policy Engine Module:**
  - Develop a standalone backend module that consumes request context (user role, data sensitivity, network trust status).
  - Returns policy decision: ALLOW, DENY, or READ_ONLY.

- **Context Vectors:**
    - Factor in user's role, file sensitivity, and network location (corporate VPN vs. untrusted IP).
- **Policy Rules:**
    - Hardcode initial rules prioritizing safety:
        - **Public files:** Unrestricted access.
        - **Confidential:** Employees get download access only from trusted networks.
        - **Restricted:** No downloads, only READ_ONLY.

Week 4: Middleware Integration & Enforcement (Approx. 32 hours)

- **Security Middleware:**
    - Implement middleware in FastAPI or equivalent, intercepting critical API calls (e.g., download, view).
- **Policy Enforcement:**
    - Query Policy Engine for every request, enforce returned access decision in real time.
    - Deliver watermarked or non-downloadable file previews when appropriate.

Week 5: Behavioral Anomaly Detection (Proof-of-Concept) (Approx. 32 hours)

- **Activity Logging:**
    - Log key user actions (login, file management, sharing) with full context to a separate database table.
- **Anomaly Detection Model:**
    - Use Isolation Forest (scikit-learn) for unsupervised anomaly detection based on activity features (e.g., files accessed per hour, data downloaded per session).
- **Simulation and Testing:**
    - Script to generate simulated logs for "normal" activity and separate "attack" patterns (e.g., rapid mass downloads).
    - Validate model's effectiveness in flagging anomalies.

Week 6: Advanced Dashboard & Final Polish (Approx. 28 hours)

- **Security Dashboard:**
  - Build a frontend page showing:
    - Live log of Policy Engine access decisions.
    - Real-time alerts for threats detected via Behavioral Anomaly model.
- **Polish and Demo Preparation:**
  - Refine codebase, architecture documentation, and testing.
  - Prepare end-to-end demonstration covering regular and threat scenarios, highlighting adaptability and intelligence of the platform.

REQUIREMENT ANALYSIS (SRS)

HARDWARE AND SOFTWARE SPECIFICATIONS

HARDWARE

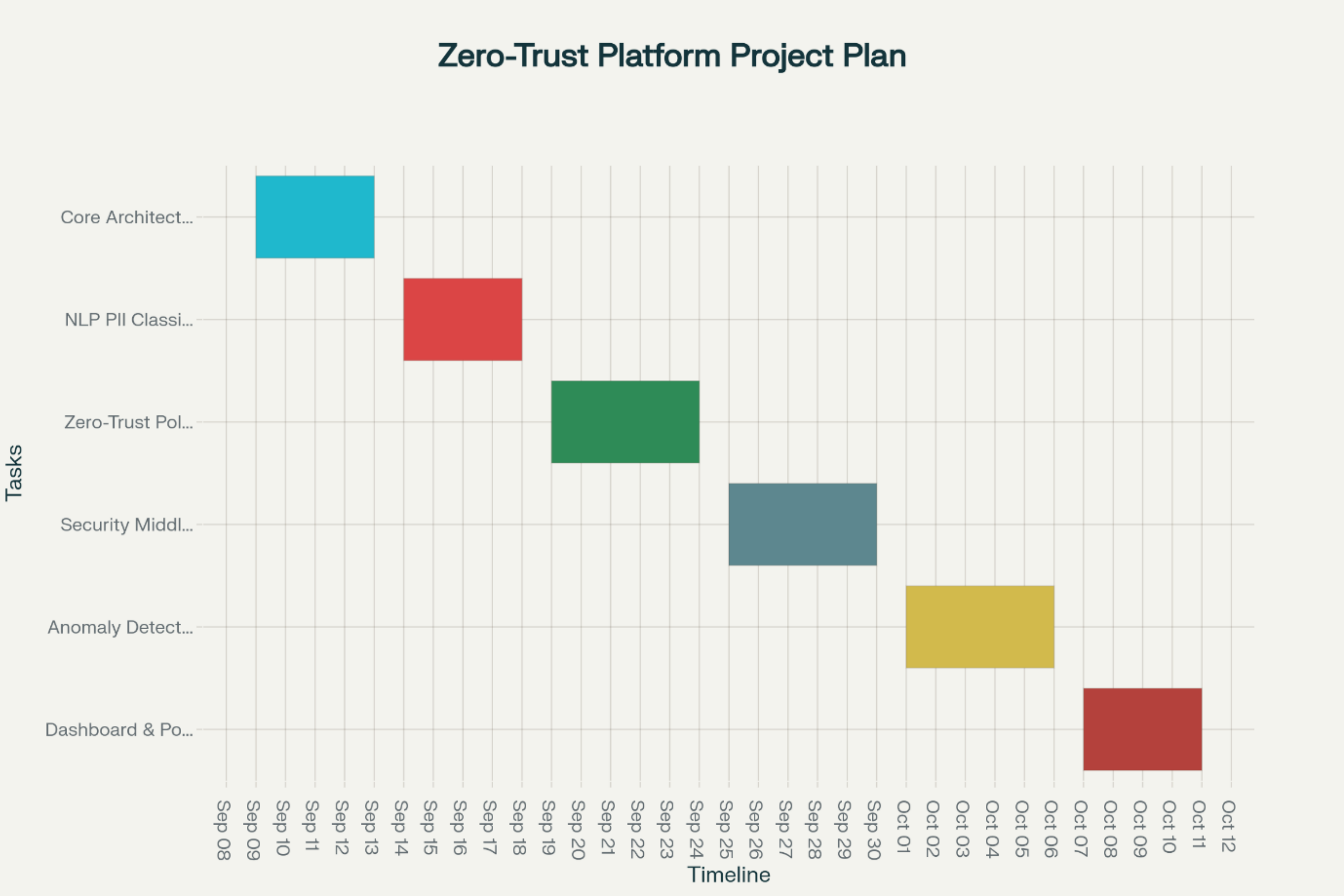
- **Development Machines:** Standard x86-64 architecture systems, minimum Quad-core CPU @2.0 GHz, 16GB RAM, SSD storage (256GB+), Ethernet/Wi-Fi connectivity, supporting container runtime (Docker).
- **Server Deployment (for cloud or on-prem):**
  - Linux VM (Ubuntu 22.04 LTS or CentOS 8), 4 vCPU, 16GB RAM, 200GB SSD; scalable up to cluster environments.
- **Networking:** Trusted LAN/VPN and untrusted public IP ranges for simulating Zero Trust scenarios. NAT/firewall configurations.

SOFTWARE

- **Backend:** Python 3.9+ (FastAPI framework), Docker Compose, Celery (task queue), PostgreSQL 13+, Redis 6+
- **PII/NLP Analysis:** Hugging Face Transformers (DistilBERT/RoBERTa), scikit-learn (Isolation Forest)
- **Authentication:** WebAuthn (FIDO2) libraries, JWT tokens
- **Frontend:** React.js/Vue.js (latest), Chart.js/D3.js for dashboard visuals
- **DevOps:** Docker Compose, Git, CI/CD tools (GitHub Actions/CI), VS Code/PyCharm IDE
- **Utilities:** Nginx (reverse proxy), REST API tools (Postman), Testing frameworks (pytest, unittest)

GANTT CHART WITH WORK BREAKDOWN STRUCTURE

This Gantt chart visualizes timelines, task dependencies, and weekly milestones. Below is a project schedule (September–October 2025):



The project spans approximately 14 weeks total, with sequential phases building upon each other. Each colored bar represents a different project component with specific start and end dates.

Week	Task	Subtasks
1	Core Architecture & Auth	Docker Setup, WebAuthn, File API development
2	NLP PII Classifier	Hugging Face, Model tuning, Celery background
3	Zero-Trust Policy Engine	Context vectors, Policy rule encoding
4	Security Middleware	API interception, Policy enforcement
5	Anomaly Detection	Logging, Data simulation, Isolation Forest ML
6	Dashboard & Final Polish	Dashboard design, Alerts, Documentation

FUNCTIONAL REQUIREMENTS

- Secure user authentication via passwordless WebAuthn
- File management APIs (upload, download, list, delete) with strict user validation
- Background classification of files using NLP for PII and sensitivity levels
- Dynamic policy enforcement (allow, deny, read-only) via Zero-Trust engine
- Real-time activity logging and anomaly detection alerts
- Responsive dashboard for access decisions and threat notifications

NON-FUNCTIONAL REQUIREMENTS

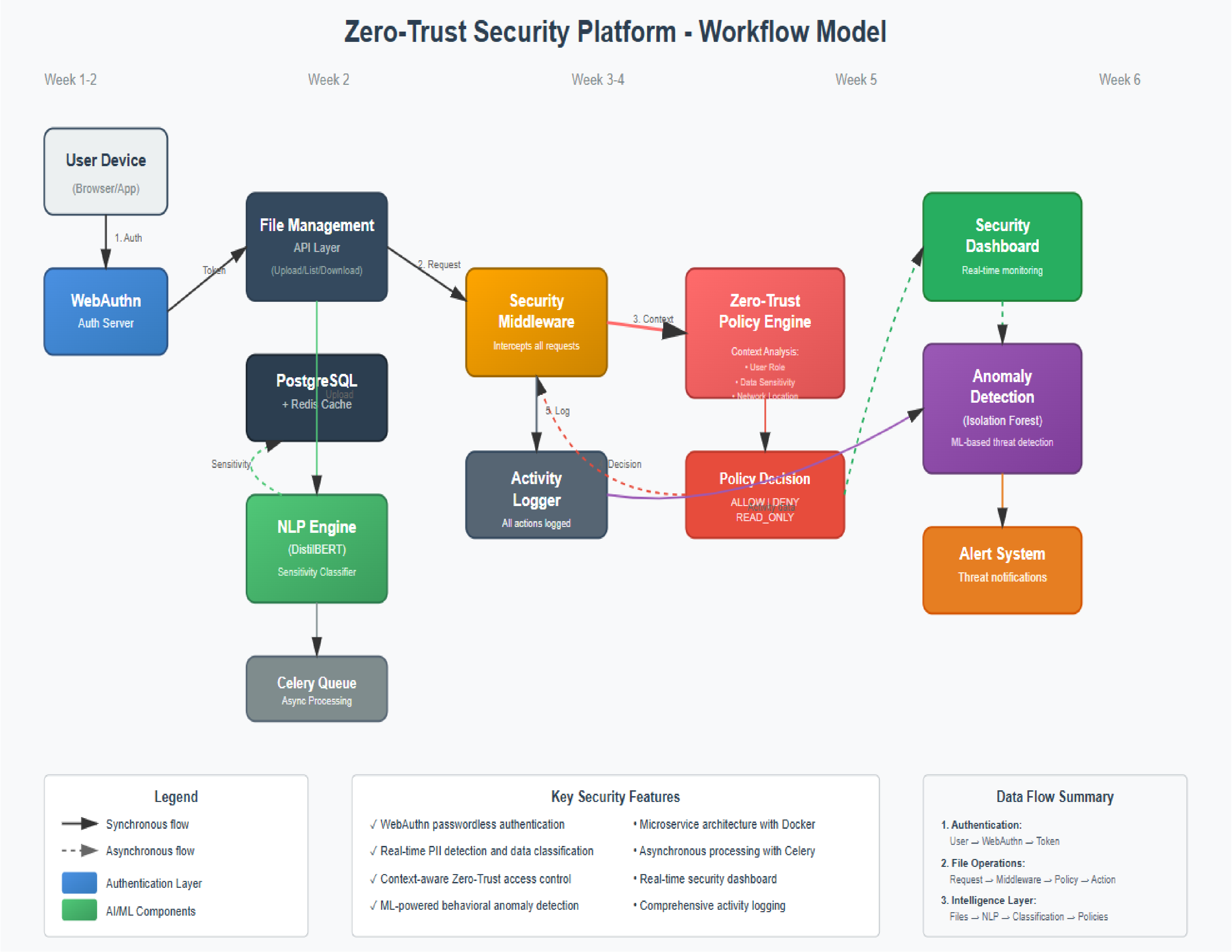
- Scalability: Modular microservice architecture supporting load balancing and container orchestration
- Reliability: Robust error handling and transaction logs
- Usability: Clean UI/UX for dashboard, fast API response
- Security: End-to-end encryption (TLS), strict context-aware access control
- Compliance: Designed for GDPR and other regulatory frameworks

CONSTRAINTS

- Open-source and modern libraries/tools preferred
- Project delivered within 1.5 months
- Cloud-ready or on-prem deployment flexibility

SYSTEM DESIGN

WORKFLOW MODEL



Zero-Trust Security Platform - Workflow Model

Legend

- Synchronous flow
- Asynchronous flow
- Authentication Layer
- AI/ML Components

Key Security Features

✓ WebAuthn passwordless authentication

✓ Real-time PII detection and data classification

✓ Context-aware Zero-Trust access control

✓ ML-powered behavioral anomaly detection

• Microservice architecture with Docker

• Asynchronous processing with Celery

• Real-time security dashboard

• Comprehensive activity logging

Data Flow Summary

1. Authentication:
User → WebAuthn → Token

2. File Operations:
Request → Middleware → Policy → Action

3. Intelligence Layer:
Files → NLP → Classification → Policies

MODULE DESIGN AND IMPLEMENTATION (UP TO WEEK 2)

Week 1: Hardened Core & Authentication

- **Docker-based Service Architecture**
    - Python backend, PostgreSQL (users/files), Redis (cache, Celery broker)
    - Compose files with clear service boundaries, network isolation
- **WebAuthn Integration**
    - Registration, login endpoints, public key cryptography, FIDO2 libraries
    - JWT token issuance after authentication
- **File Management APIs**
    - Endpoints: /upload, /list, /download, /delete
    - User ownership checks on all ops, error handling for missing/wrong files

Week 2: Intelligent Data Classifier

- **NLP Service**
    - Integration of Hugging Face Transformers (initially DistilBERT/RoBERTa but afterwards self-hosted LLM will be used.)
    - API route to ingest and asynchronously classify files for PII
- **Data Classification Module**
    - Assign sensitivity (Public, Internal, Confidential, Restricted) in DB models
- **Celery Task Integration**
    - Asynchronous background processing for classification, decoupled from API latency
    - Regular updating of the file’s sensitivity tag in the database

REFERENCES

1. National Cyber Security Centre. (n.d.). *Zero trust architecture*. NCSC. Retrieved September 10, 2025, from https://www.ncsc.gov.uk/collection/zero-trust-architecture

2. Duo Security. (n.d.). *How to go from MFA to zero trust*. Duo. Retrieved September 10, 2025, from https://duo.com/resources/ebooks/from-mfa-to-zero-trust

3. National Institute of Standards and Technology. (n.d.). *Implementing a zero trust architecture*. NIST. Retrieved September 10, 2025, from https://www.nccoe.nist.gov/projects/implementing-zero-trust-architecture

4. Zero Trust Architecture: A Systematic Literature Review – Explores adaptive access control and contextual decision-making in cloud environments from https://arxiv.org/html/2503.11659v1

5. Implementing Zero Trust Security Models in Cloud Computing – Reviews zero trust deployment strategies and operational challenges in cloud from https://wjarr.com/sites/default/files/WJARR-2024-3500.pdf

6. Redefining Zero Trust Architecture in Cloud Networks - granular, and context aware policy enforcement for cloud security from https://magnascientiapub.com/journals/msarr/sites/default/files/MSARR-2021-0032.pdf

7. Comparing Feature-based and Context-aware Approaches to PII Generalization – Evaluates context-aware models for accurate PII detection from https://arxiv.org/html/2407.02837v1

8. Anomaly Detection in Cloud Computing-1machine learning techniques for proactive threat identification in the cloud from
https://ijsrset.com/index.php/home/article/download/IJSRSET2512402/IJSRSET2512402/1260