

BCSE497J Project-I

# Zero-Trust Cloud Collaboration Platform

22BCEO465 – Swarit Jain

22BCTO307 – Akshay Agarwal

22BDSO110 – Shivansh Srivastava



# Literature Review: Foundations of Zero Trust

Our platform's design is informed by cutting-edge research in Zero-Trust architectures and cloud security. We've synthesized key insights from leading academic and industry publications to build a robust and adaptive solution.

## Zero Trust Architecture: A Systematic Literature Review

Explores adaptive access control and contextual decision-making in cloud environments.

## Implementing Zero Trust Security Models in Cloud Computing

Reviews zero trust deployment strategies and operational challenges in cloud.

## Redefining Zero Trust Architecture in Cloud Networks

Discusses dynamic, granular, and context-aware policy enforcement for cloud security.

## Comparing Feature-based and Context-aware Approaches to PII Generalization

Evaluates context-aware models for accurate PII detection.

## Anomaly Detection in Cloud Computing

Systematic review of machine learning techniques for proactive threat identification in the cloud.



# Addressing Critical Gaps in Cloud Security

Our research identified key limitations in existing cloud security paradigms. The Zero-Trust Cloud Collaboration Platform is specifically designed to overcome these challenges, ensuring a more dynamic and intelligent security posture.

## → Static Access Policies

Fail to adapt to evolving user contexts, roles, and environments, creating vulnerabilities.

## → Surface-Level PII Detection

Overlooks nuanced, context-sensitive risks, leading to potential data breaches.

## → Lack of Real-Time Integration

Limits proactive security for policy enforcement and behavioral anomaly detection.

# Our Strategic Objectives

The development of the Zero-Trust Cloud Collaboration Platform is guided by a clear set of objectives, focusing on modularity, intelligence, and real-time responsiveness.

01

## Modular Cloud Platform

Build a modular, microservice-ready cloud platform that ensures continuous, context-sensitive verification.

02

## Automated PII Classification

Implement automated, context-aware NLP-based PII and sensitivity classification for uploaded files.

03

## Adaptive Policy Engine

Develop an adaptive Zero-Trust Policy Engine for informed access decisions based on multiple dynamic factors.

04

## Behavioral Anomaly Detection

Integrate behavioral anomaly detection using ML to spot and respond to insider and external threats in real time.

05

## Live Visualization Dashboard

Provide a demo-ready dashboard for live visualization of access logs and security alerts.

# Platform Specifications: Hardware

Our platform is designed for flexibility, supporting both development and scalable deployment environments with robust hardware requirements.

## 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



- Linux VM (Ubuntu 22.04 LTS or CentOS 8)
- 4 vCPU, 16GB RAM, 200GB SSD
- Scalable up to cluster environments
- Trusted LAN/VPN and untrusted public IP ranges
- NAT/firewall configurations for Zero Trust scenarios



# Platform Specifications: Software Stack

A comprehensive and modern software stack underpins our Zero-Trust platform, leveraging open-source tools for efficiency and scalability.

## Backend & Infrastructure

- Python 3.9+ (FastAPI framework)
- Docker Compose
- Celery (task queue)
- PostgreSQL 13+
- Redis 6+



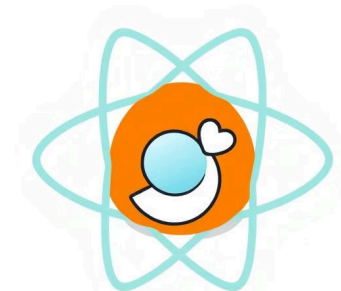
## AI/ML & Security

- Hugging Face Transformers (DistilBERT/RoBERTa)
- scikit-learn (Isolation Forest)
- WebAuthn (FIDO2) libraries
- JWT tokens



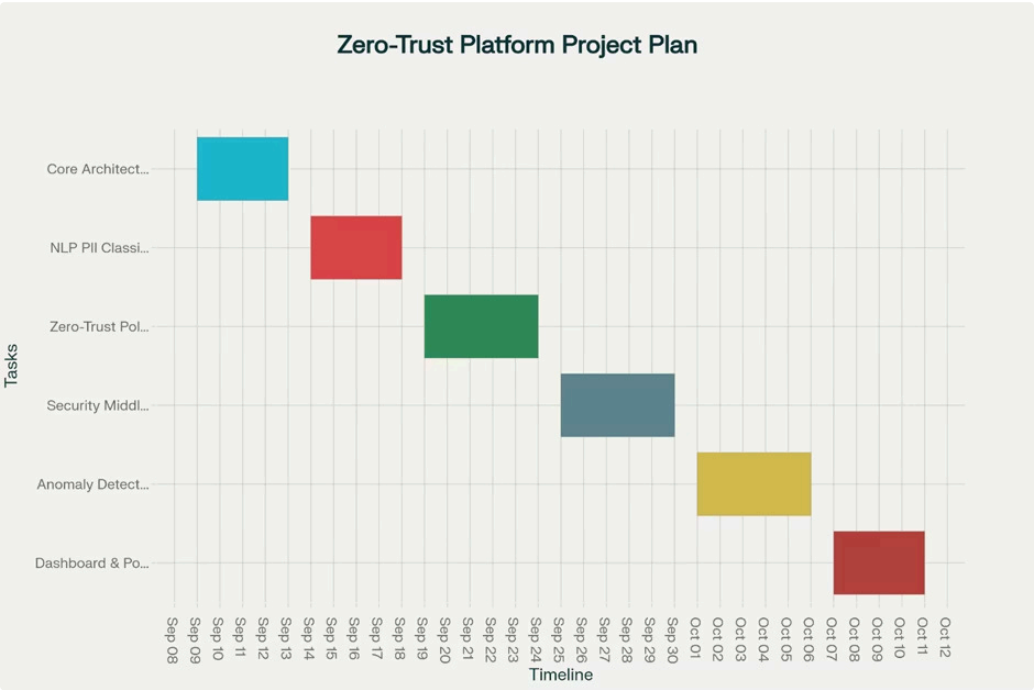
## Frontend & DevOps

- React.js/Vue.js (latest)
- Chart.js/D3.js for dashboard visuals
- Git, CI/CD tools (GitHub Actions/CI)
- VS Code/PyCharm IDE
- 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

# Requirement Analysis: Functional Capabilities

The platform's core functionalities are designed to provide comprehensive security and seamless collaboration within a Zero-Trust framework.

1

## Secure User Authentication

Via passwordless WebAuthn for enhanced security.

2

## File Management APIs

Upload, download, list, delete with strict user validation.

3

## Background File Classification

Using NLP for PII and sensitivity levels.

4

## Dynamic Policy Enforcement

Allow, deny, read-only via Zero-Trust engine.

5

## Real-Time Monitoring

Activity logging and anomaly detection alerts.

6

## Responsive Dashboard

For access decisions and threat notifications.



# Requirement Analysis: Non-Functional & Constraints

Beyond core features, the platform adheres to critical non-functional requirements and operates within defined project constraints to ensure quality and timely delivery.

## Non-Functional Requirements



### Scalability

Modular microservice architecture supporting load balancing and container orchestration.

### Reliability

Robust error handling and transaction logs for data integrity.

### Usability

Clean UI/UX for dashboard, fast API response for user satisfaction.

### Security

End-to-end encryption (TLS), strict context-aware access control.

### Compliance

Designed for GDPR and other regulatory frameworks.

## Project Constraints



### Open-Source Preference

Prioritizing modern, open-source libraries and tools.

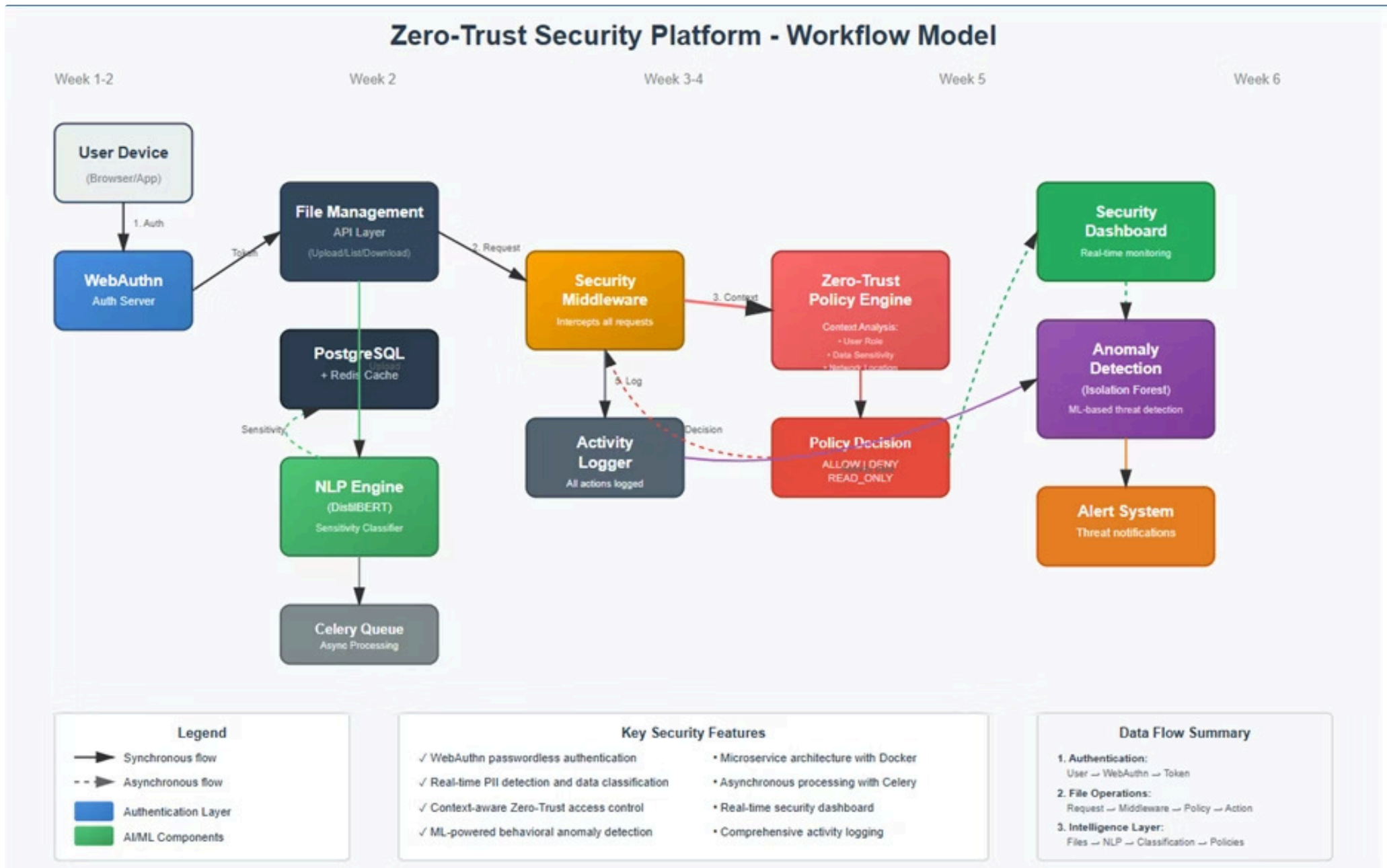
### Timeframe

Project delivered within 1.5 months.

### Deployment Flexibility

Cloud-ready or on-prem deployment options.

# Workflow Model



# Module Design & Implementation: Week 1 Focus

The initial phase of development focuses on establishing the core architecture and foundational security mechanisms.



## Docker-based Service Architecture

Python backend, PostgreSQL (users/files), Redis (cache, Celery broker). Compose files with clear service boundaries, network isolation.



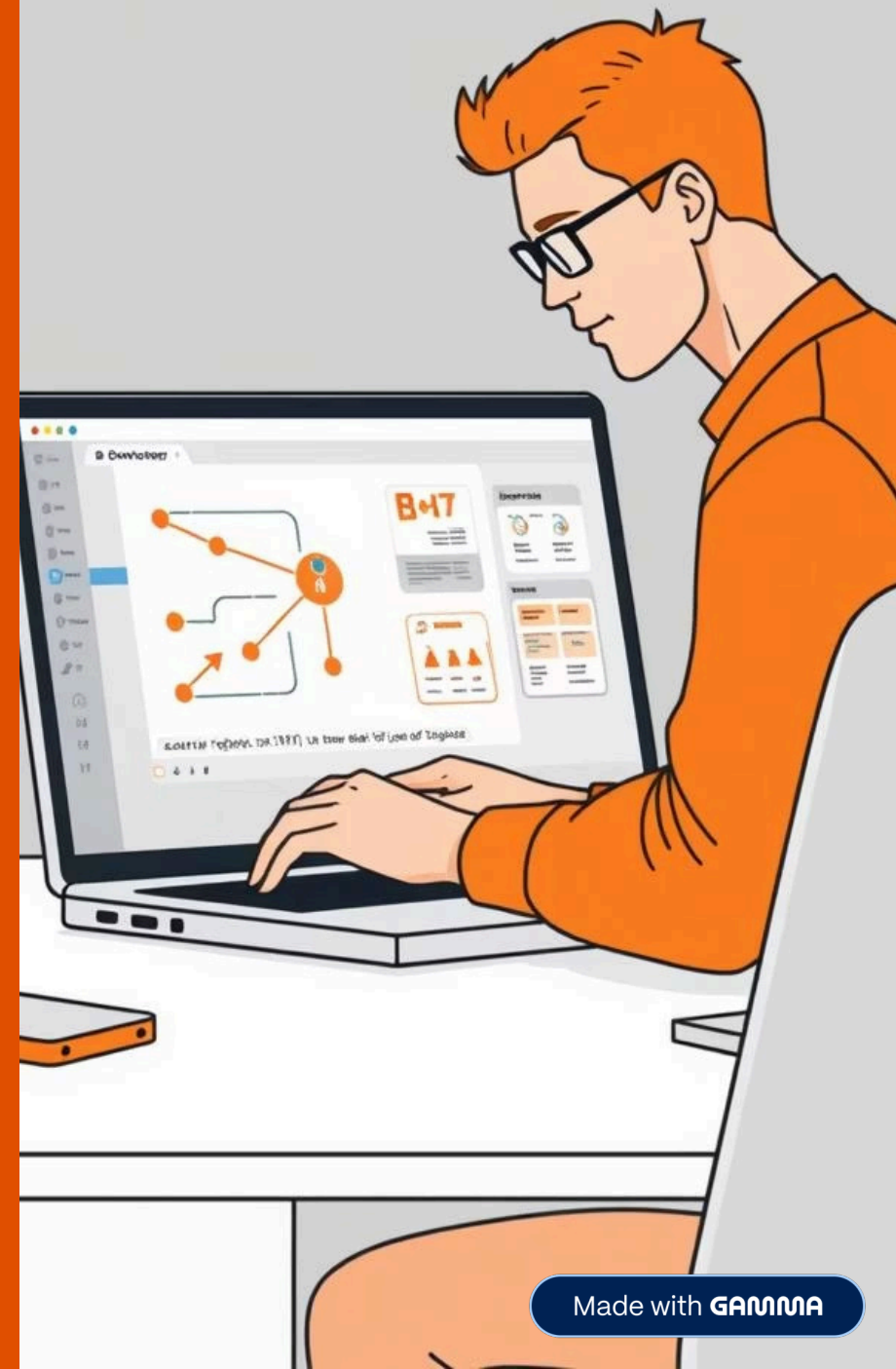
## WebAuthn Integration

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.



# Module Design & Implementation: Week 2 Focus

Building on the core, Week 2 introduces intelligent data classification capabilities, crucial for context-aware security.

- 1

### NLP Service Integration

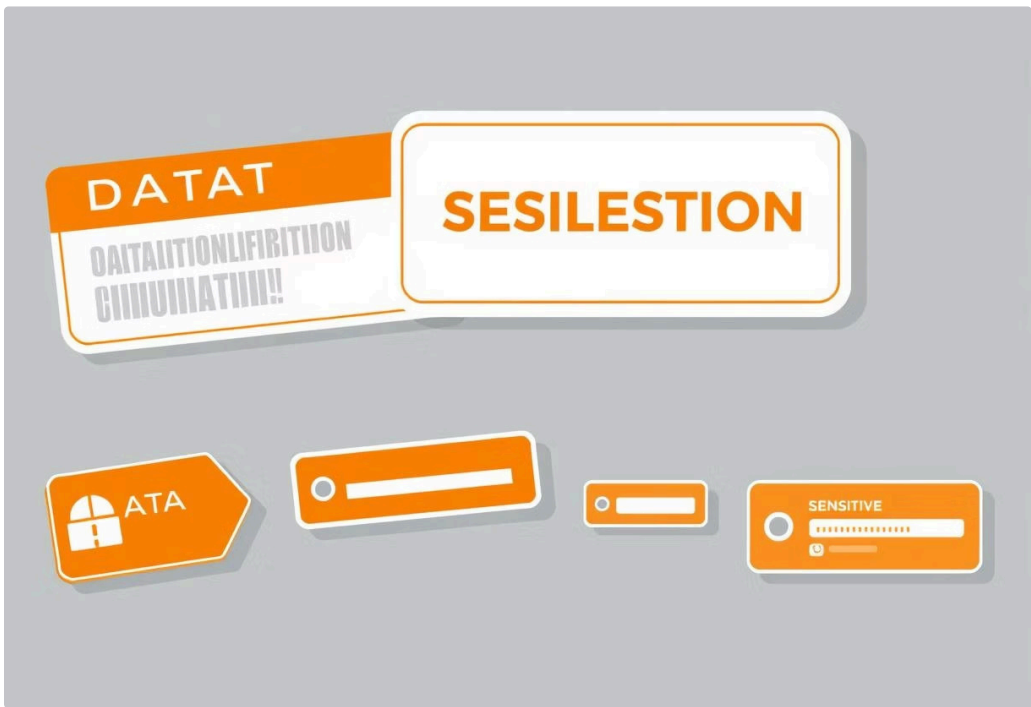
Integration of Hugging Face Transformers (DistilBERT/RoBERTa). API route to ingest and asynchronously classify files for PII.



- 2

### Data Classification Module

Assign sensitivity (**Public, Internal, Confidential, Restricted**) in DB models.



- 3

### Celery Task Integration

Asynchronous background processing for classification, decoupled from API latency. Regular updating of the file's sensitivity tag in the database.

