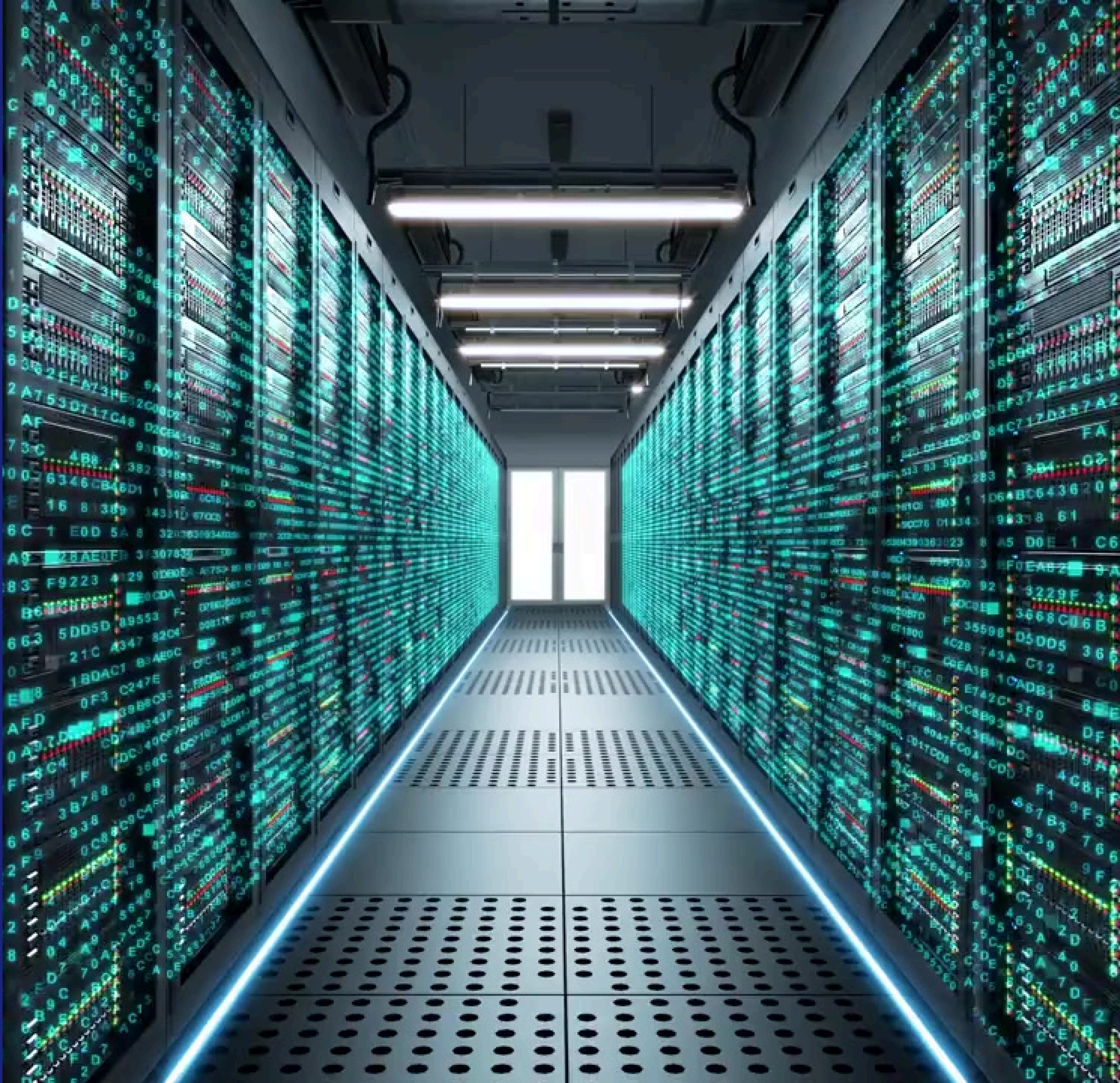
Cloud Computing Redaction Project

Michail Angelos Karvelas



Project Overview

The aim of this project is to establish a secure system that anonymizes sensitive documents, similar to a redaction service.

1. **The Goal**

Create a trusted system that protects privacy by anonymizing documents.
Automatically clear names, emails, and addresses from text.

2. **The Solution**

A simple service that receives text and sends back a fully cleaned PDF.
Powered by language models that find sensitive details with context.

3. **The Architecture**

Built for the cloud with containers and Kubernetes.
Ready for high demand with smart routing and traffic control.
Sensitive content is erased before any file is generated.

The Problem & Solution

- The Challenge: Handling sensitive documents (PII) securely is difficult. Manual redaction is slow and error-prone.
- The Solution: An automated Microservice Architecture that detects and permanently removes PII.
- Key Capabilities:
  - Intelligent Detection: Combines Regex (Email/Phone) and AI (Names/Locations).
  - Visual Redaction: Physically draws black boxes on PDFs (not just covering text).

Scalable: Runs on a self-healing Cloud Cluster (AKS).

System Architecture

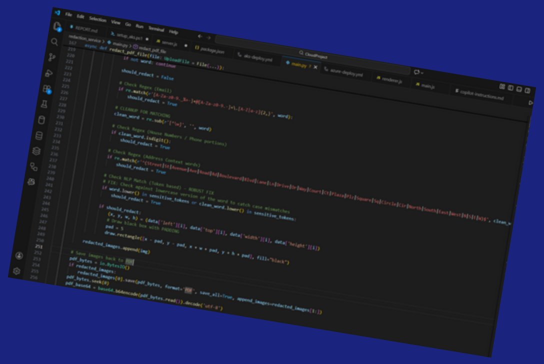
- Frontend (Consumer):
  User Interface for uploading files.

- Backend (Producer):
  Python FastAPI service for OCR and NLP processing.

- Service Discovery:
  Consul allows the Frontend to find the Backend dynamically.

System Architecture

Backend Explanation



```
async def redact_pdf_file(file: UploadFile = File(...)):
            if not word: continue

            should_redact = False

            # Check Regex (Email)
            if re.match(r'[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}', word):
                should_redact = True

            # CLEANUP FOR MATCHING
            clean_word = re.sub(r'[^\w]', '', word)

            # Check Regex (House Numbers / Phone portions)
            if clean_word.isdigit():
                should_redact = True

            # Check Regex (Address Context words)
            if re.match(r'^(Street|St|Avenue|Ave|Road|Rd|Boulevard|Blvd|Lane|Ln|Drive|Dr|Way|Court|Ct|Plaza|Plz|Square|Sq|Circle|Cir|North|South|East|West|N|S|E|W)$', clean_w
                should_redact = True

            # Check NLP Match (Token based) - ROBUST FIX
            # FIX: Check against lowercase version of the word to catch case mismatches
            if word.lower() in sensitive_tokens or clean_word.lower() in sensitive_tokens:
                should_redact = True

            if should_redact:
                (x, y, w, h) = (data['left'][i], data['top'][i], data['width'][i], data['height'][i])
                # Draw black box with PADDING
                pad = 5
                draw.rectangle([x - pad, y - pad, x + w + pad, y + h + pad], fill="black")
        redacted_images.append(img)
    # Save images back to PDF
    pdf_bytes = io.BytesIO()
    if redacted_images:
        redacted_images[0].save(pdf_bytes, format='PDF', save_all=True, append_images=redacted_images[1:])
    pdf_bytes.seek(0)
    pdf_base64 = base64.b64encode(pdf_bytes.read()).decode('utf-8')
```

Regex: Used for deterministic patterns (Emails, Phones). High speed, 100% accuracy for fixed formats.

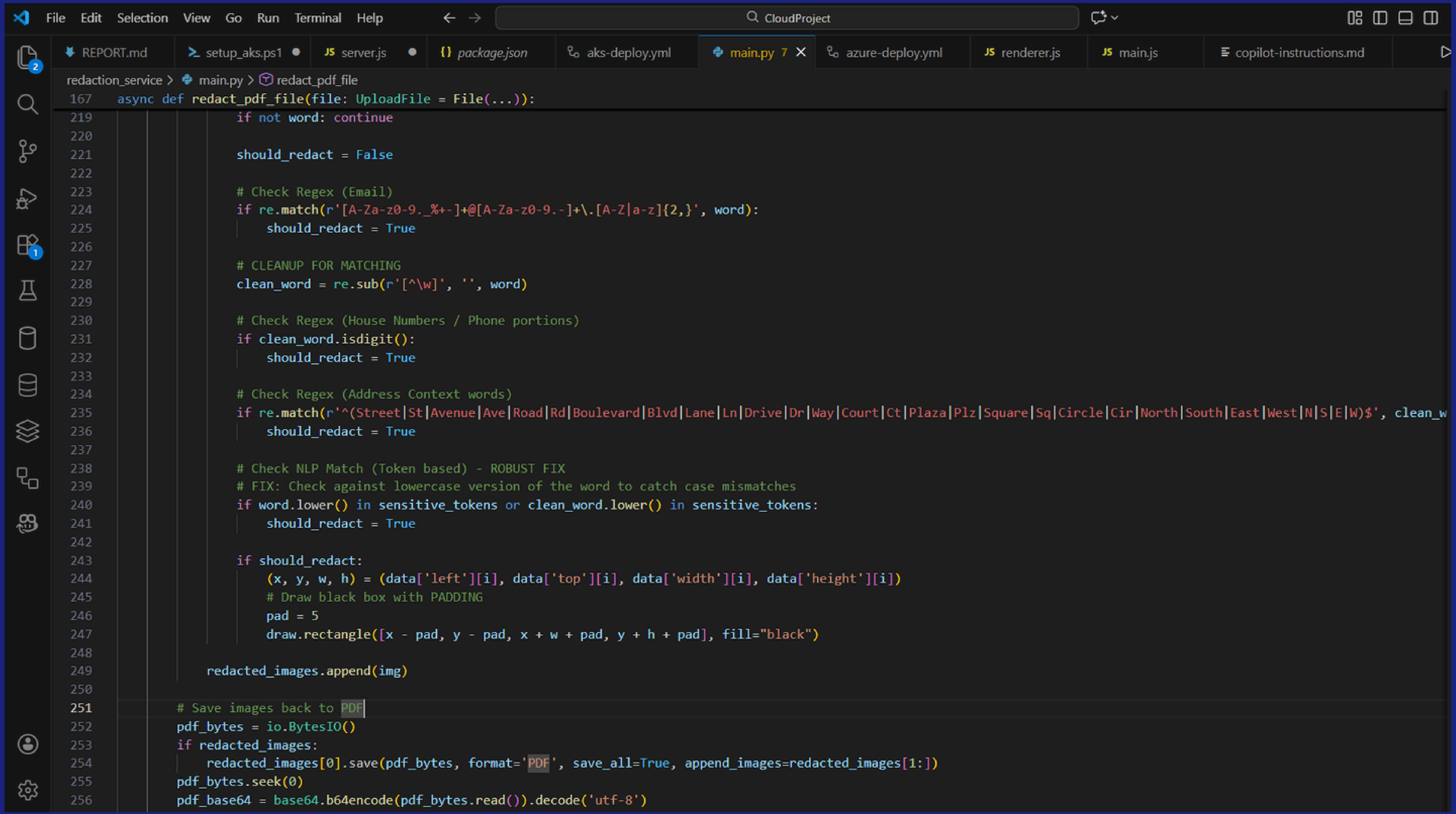
NLP (spaCy): Used for context-aware entities (Names, Cities). Capable of distinguishing "Park" (Name) from "Park" (Location).

Why Hybrid?

Combining both covers the weaknesses of each, ensuring maximum accuracy.

System Architecture

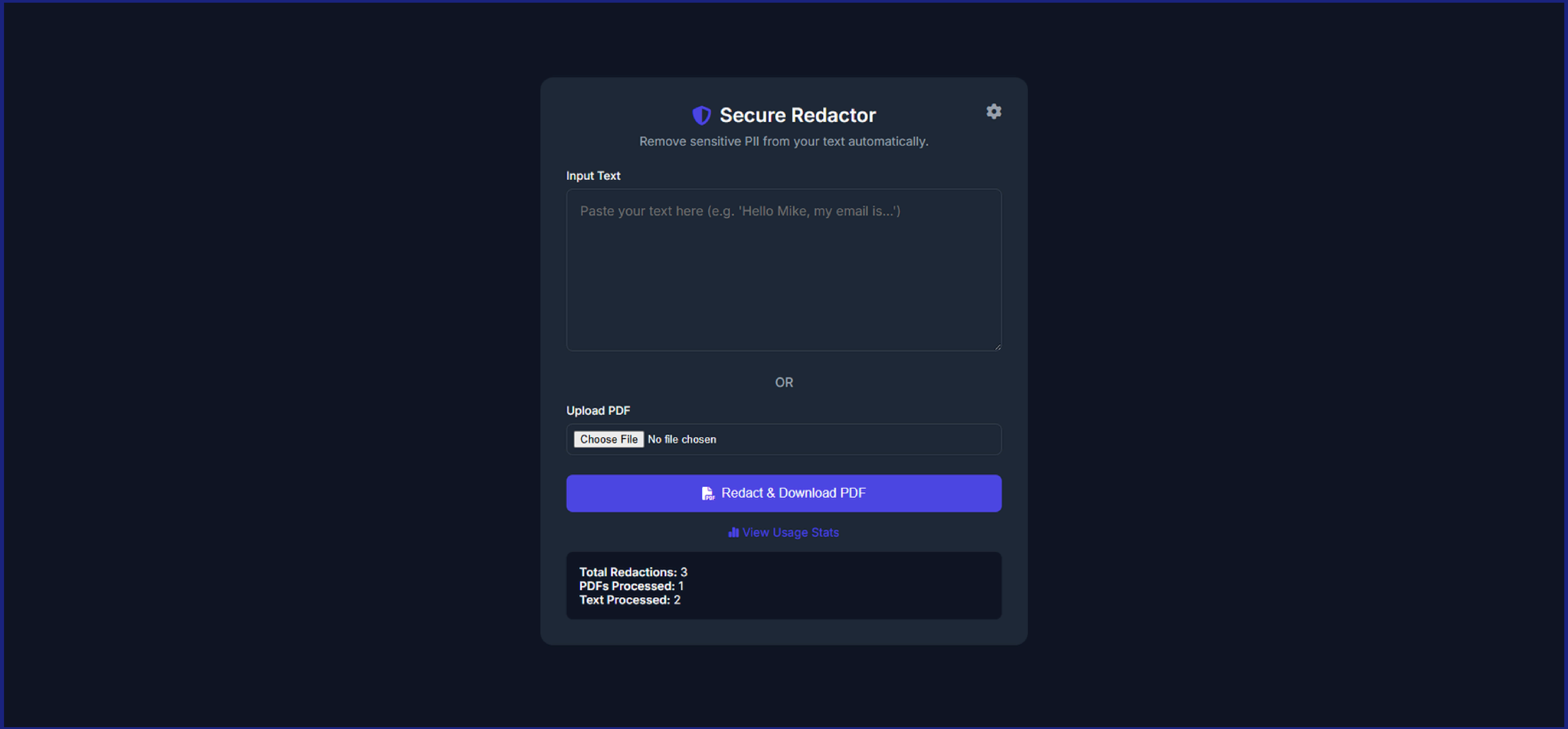
Backend Code Screenshot



```python
async def redact_pdf_file(file: UploadFile = File(...)):
                if not word: continue

                should_redact = False

                # Check Regex (Email)
                if re.match(r'[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}', word):
                    should_redact = True

                # CLEANUP FOR MATCHING
                clean_word = re.sub(r'[^\w]', '', word)

                # Check Regex (House Numbers / Phone portions)
                if clean_word.isdigit():
                    should_redact = True

                # Check Regex (Address Context words)
                if re.match(r'^(Street|St|Avenue|Ave|Road|Rd|Boulevard|Blvd|Lane|Ln|Drive|Dr|Way|Court|Ct|Plaza|Plz|Square|Sq|Circle|Cir|North|South|East|West|N|S|E|W)$', clean_w
                    should_redact = True

                # Check NLP Match (Token based) - ROBUST FIX
                # FIX: Check against lowercase version of the word to catch case mismatches
                if word.lower() in sensitive_tokens or clean_word.lower() in sensitive_tokens:
                    should_redact = True

                if should_redact:
                    (x, y, w, h) = (data['left'][i], data['top'][i], data['width'][i], data['height'][i])
                    # Draw black box with PADDING
                    pad = 5
                    draw.rectangle([x - pad, y - pad, x + w + pad, y + h + pad], fill="black")

            redacted_images.append(img)

        # Save images back to PDF
        pdf_bytes = io.BytesIO()
        if redacted_images:
            redacted_images[0].save(pdf_bytes, format='PDF', save_all=True, append_images=redacted_images[1:])
        pdf_bytes.seek(0)
        pdf_base64 = base64.b64encode(pdf_bytes.read()).decode('utf-8')
```

System Architecture

FrontEnd



Secure Redactor

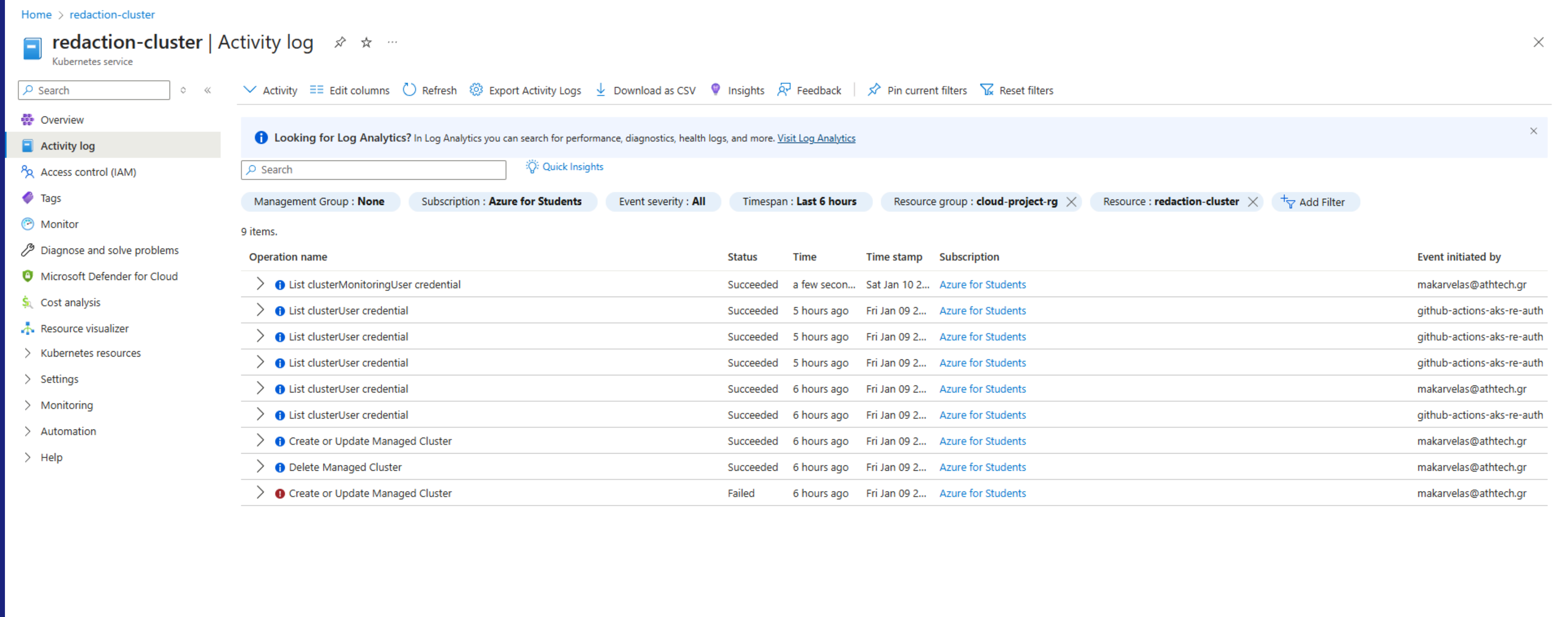
Remove sensitive PII from your text automatically.

Total Redactions: 3
PDFs Processed: 1
Text Processed: 2

Kubernetes Infrastructure (AKS)

- Key Specs:
  - Cluster: Azure Kubernetes Service (AKS)
  - Nodes: 2 Nodes (Standard_B2s) for hardware redundancy
- Networking:
  - LoadBalancer: Exposes the Frontend to the public internet

ClusterIP: Keeps Backend communication private and secure



redaction-cluster | Activity log

Kubernetes service

Looking for Log Analytics? In Log Analytics you can search for performance, diagnostics, health logs, and more. Visit Log Analytics

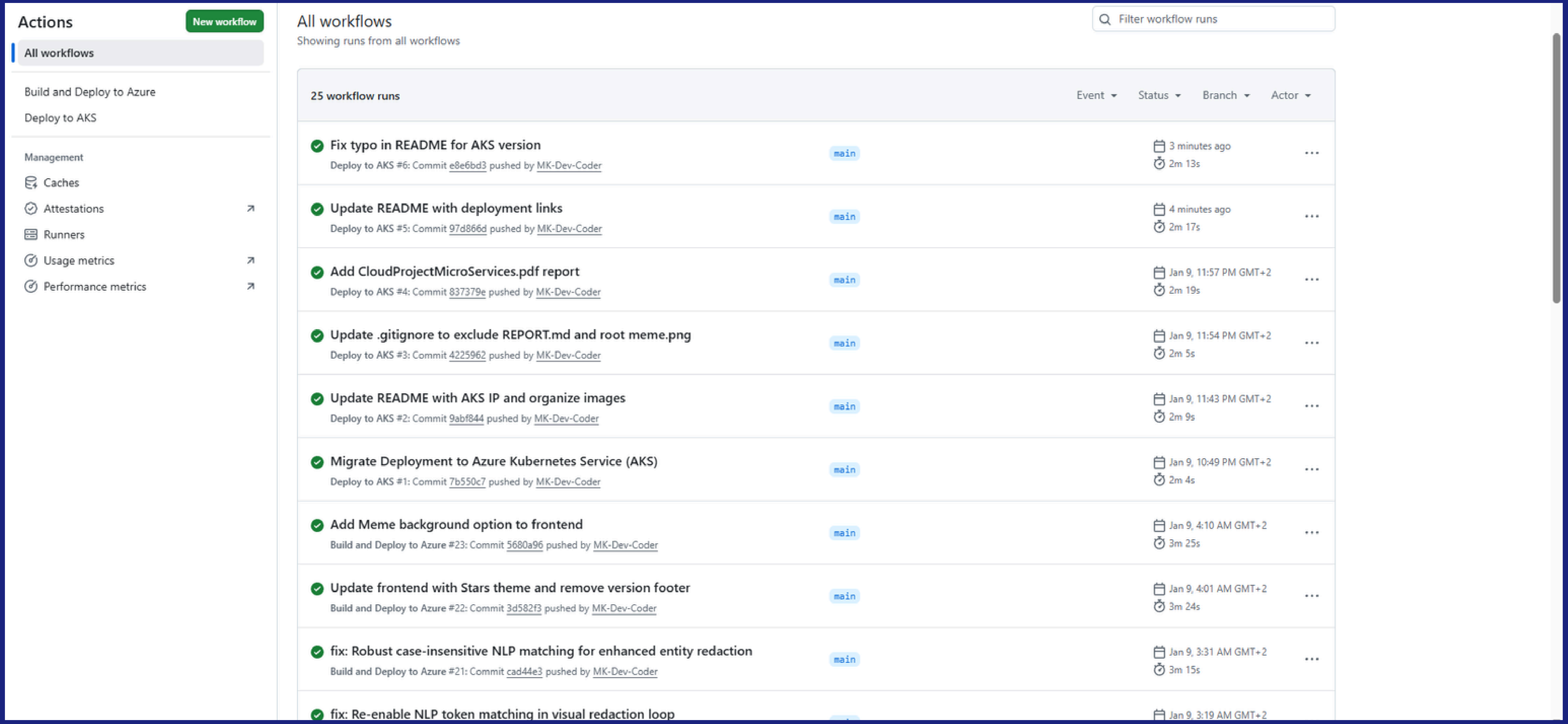
Management Group : **None** · Subscription : **Azure for Students** · Event severity : **All** · Timespan : **Last 6 hours** · Resource group : **cloud-project-rg** · Resource : **redaction-cluster**

9 items.

Operation name	Status	Time	Time stamp	Subscription	Event initiated by
List clusterMonitoringUser credential	Succeeded	a few secon...	Sat Jan 10 2...	Azure for Students	makarvelas@athtech.gr
List clusterUser credential	Succeeded	5 hours ago	Fri Jan 09 2...	Azure for Students	github-actions-aks-re-auth
List clusterUser credential	Succeeded	5 hours ago	Fri Jan 09 2...	Azure for Students	github-actions-aks-re-auth
List clusterUser credential	Succeeded	5 hours ago	Fri Jan 09 2...	Azure for Students	github-actions-aks-re-auth
List clusterUser credential	Succeeded	6 hours ago	Fri Jan 09 2...	Azure for Students	makarvelas@athtech.gr
List clusterUser credential	Succeeded	6 hours ago	Fri Jan 09 2...	Azure for Students	github-actions-aks-re-auth
Create or Update Managed Cluster	Succeeded	6 hours ago	Fri Jan 09 2...	Azure for Students	makarvelas@athtech.gr
Delete Managed Cluster	Succeeded	6 hours ago	Fri Jan 09 2...	Azure for Students	makarvelas@athtech.gr
Create or Update Managed Cluster	Failed	6 hours ago	Fri Jan 09 2...	Azure for Students	makarvelas@athtech.gr

CI/CD Pipeline (DevOps)

- Push: Code committed to GitHub triggers the pipeline.
- Build: Docker Image created and pushed to Azure Container Registry (ACR).
- Deploy: AKS cluster pulls the new image and updates the deployment.



All workflows

Showing runs from all workflows

25 workflow runs

Fix typo in README for AKS version
Deploy to AKS #6: Commit e8e6bd3 pushed by MK-Dev-Coder — main — 3 minutes ago — 2m 13s

Update README with deployment links
Deploy to AKS #5: Commit 97d866d pushed by MK-Dev-Coder — main — 4 minutes ago — 2m 17s

Add CloudProjectMicroServices.pdf report
Deploy to AKS #4: Commit 837379e pushed by MK-Dev-Coder — main — Jan 9, 11:57 PM GMT+2 — 2m 19s

Update .gitignore to exclude REPORT.md and root meme.png
Deploy to AKS #3: Commit 4225962 pushed by MK-Dev-Coder — main — Jan 9, 11:54 PM GMT+2 — 2m 5s

Update README with AKS IP and organize images
Deploy to AKS #2: Commit 9abf844 pushed by MK-Dev-Coder — main — Jan 9, 11:43 PM GMT+2 — 2m 9s

Migrate Deployment to Azure Kubernetes Service (AKS)
Deploy to AKS #1: Commit 7b550c7 pushed by MK-Dev-Coder — main — Jan 9, 10:49 PM GMT+2 — 2m 4s

Add Meme background option to frontend
Build and Deploy to Azure #23: Commit 5680a96 pushed by MK-Dev-Coder — main — Jan 9, 4:10 AM GMT+2 — 3m 25s

Update frontend with Stars theme and remove version footer
Build and Deploy to Azure #22: Commit 3d582f3 pushed by MK-Dev-Coder — main — Jan 9, 4:01 AM GMT+2 — 3m 24s

fix: Robust case-insensitive NLP matching for enhanced entity redaction
Build and Deploy to Azure #21: Commit cad44e3 pushed by MK-Dev-Coder — main — Jan 9, 3:31 AM GMT+2 — 3m 15s

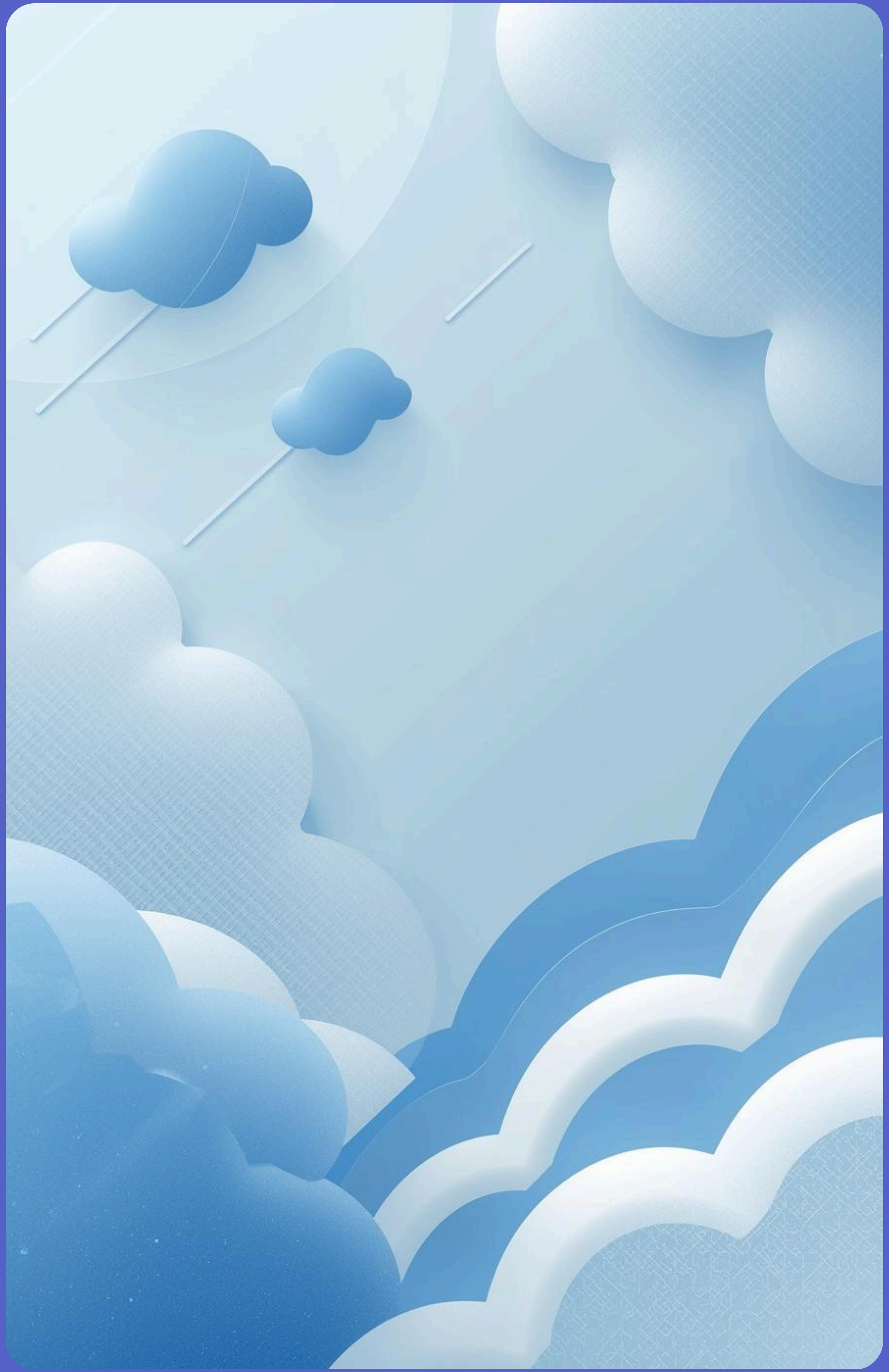
fix: Re-enable NLP token matching in visual redaction loop — Jan 9, 3:19 AM GMT+2

Implementation Steps and Phases

Development began locally, where I built and tested the backend and frontend microservices to ensure stability. After confirming that all components functioned correctly in isolation, I deployed the full solution to Microsoft Azure, making the web application accessible via the public cloud."

Future Improvements

- Advanced Visual Privacy (Scene Text Recognition):
    - Current State: The system works on scanned documents (white background, black text).
    - Future: Implement Object Detection (e.g., using YOLO or CRAFT) to identify and blur sensitive information in natural photos.
    - Use Case: Automatically blurring house numbers, license plates, and street signs in uploaded photos to protect location privacy.
- Multi-Language Support:
    - Future: Integrate additional spaCy models (e.g., es_core_news_sm for Spanish) to support international document redaction.
- User Authentication (RBAC):
- Future: Implement OAuth2 (Login with Google/Microsoft) so that only authorized employees can redact documents, with audit logs tracking who redacted what.



Thank you for your attention