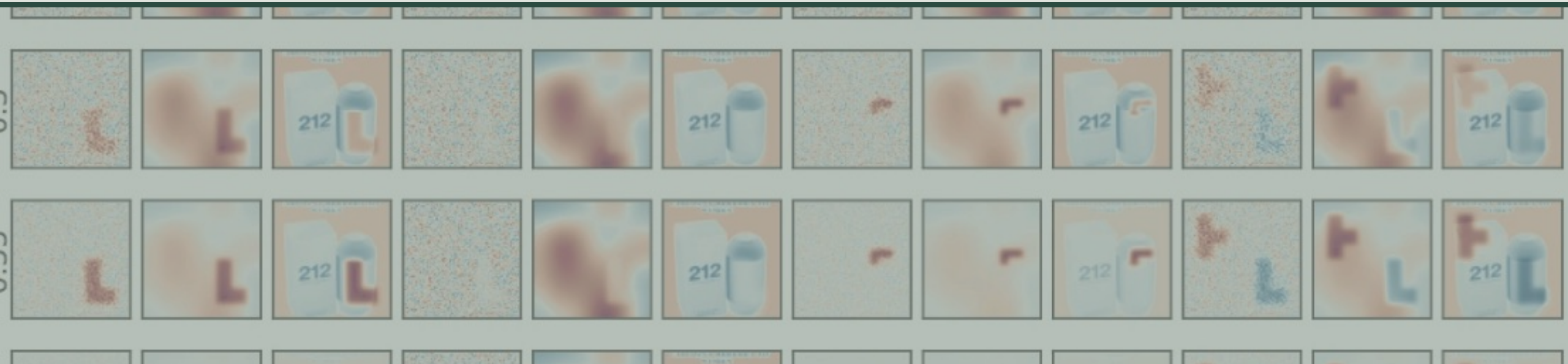
Hosting Options

Infrastructure Comparison for Secure, Scalable, and Maintainable Deployment

June 21st, 2025 - Jan Schneider

EXACT



Hosting Needs of EXACT

EXACT

Compute

Execution of Docker containers for user submissions

Python-based workers (CPU-intensive, potential GPU support)

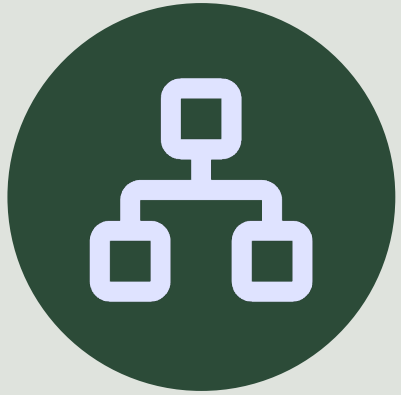
Low to moderate demand for backend (Django) and frontend (Next.js)

Storage

At least 10 GB of storage for datasets and models (e.g., MRI, Tetris)

PostgreSQL database for users, leaderboards, and challenges

Temporary upload space for submitted explanation code



Networking

Public HTTPS access (frontend and API)

Internal communication between containers (Docker Compose)

Secure API endpoints for challenge submission and administration

Security & Maintainability

Container isolation for untrusted code execution (sandboxing)

User/admin authentication and access control

Optional: CI/CD, regular backups, system monitoring and logs

Evaluation Criteria

Cost

Is the solution affordable short- and long-term, are free credits available?

Effort

How much time and expertise is required for setup, updates, and operation?

Security

Does the option offer sufficient protection for data, code execution, and access?

Scalability

Can the system grow with user demand and higher workloads?

Overview

EXACT

AZURE	AWS	Self-Hosted
Academic-friendly, good AI tooling, Docker & PostgreSQL supported, GitHub Actions-native	Similar to AZURE, most flexible, huge ecosystem, but more complex (IAM, VPC, etc.)	Full control, no external dependencies, runs on any server w/ Docker
Strong candidate due to free credits and ease of use	Technically powerful, but higher setup overhead	Ideal for isolated environments, but high effort. Could be useful for early MVPs or isolated use cases

Azure

Overview of cloud-based architecture
on Microsoft Azure

Azure Infrastructure

EXACT

Frontend

- Azure Static Web App

Backend

- Azure Container App
- Service Bus
- Azure Container Job

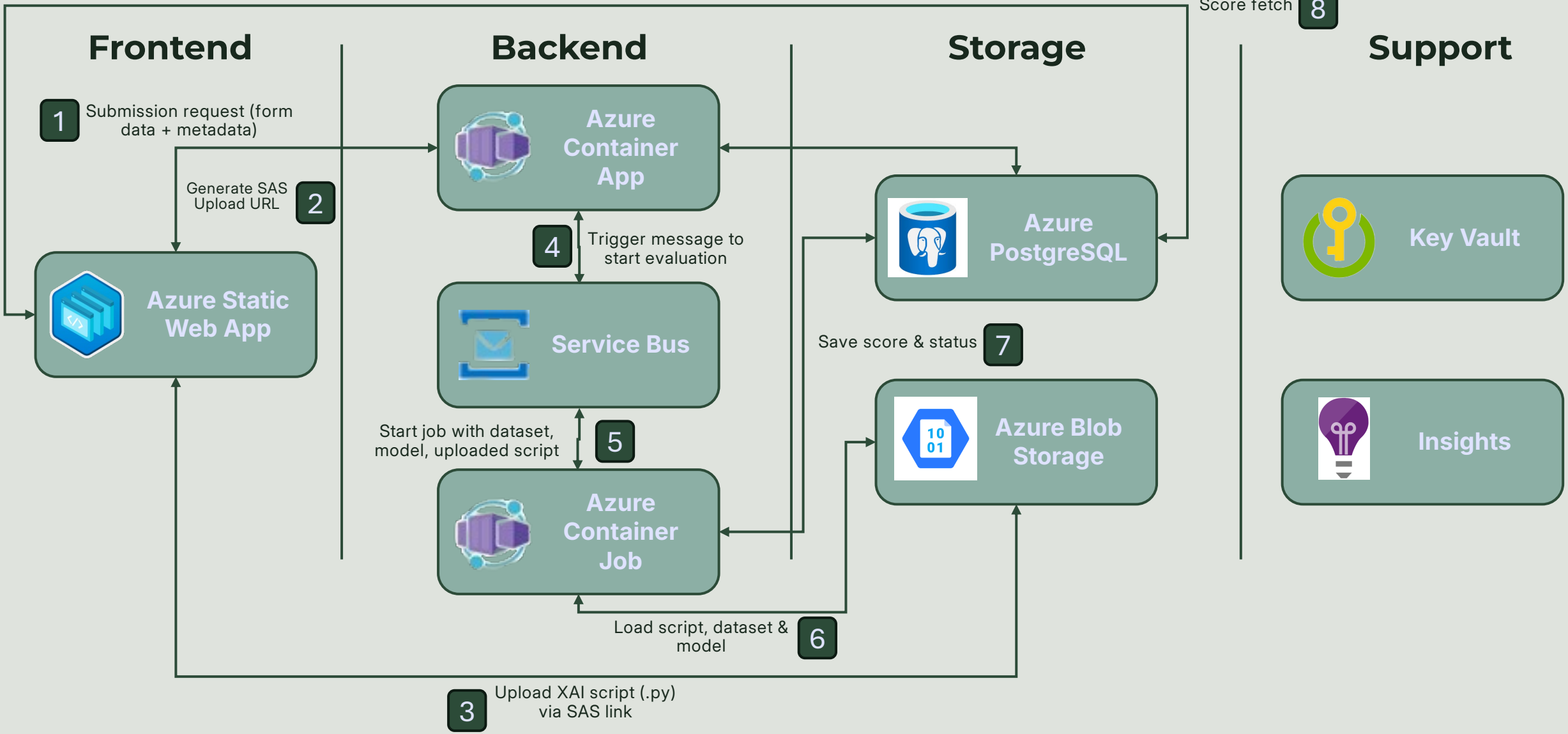
Storage

- Azure PostgreSQL
- Azure Blob Storage

Support

- Key Vault
- Insights

Submission Processing Pipeline



Conclusion Azure

EXACT

Advantages

Scalability

Easily scale compute jobs and services using Azure Container Apps & Jobs

Managed Infrastructure

No manual setup of databases, queues, or storage needed

Security & Compliance

Azure offers built-in TLS, authentication, Key Vault integration

Integrated Monitoring

Centralized insights via Azure Monitor and App Insights

CI/CD Support

Seamless GitHub Actions integration for automatic deployment

Disadvantages

Cost Overhead

Higher baseline costs compared to self-hosted Docker setups

Vendor Lock-In

Azure-specific features (e.g., SAS URLs, Service Bus) reduce portability

Complexity

Managing permissions, services, and billing adds cognitive load

Cold Start Latency

Container Jobs may take time to spin up on-demand

Self-Hosted

Overview of Self-hosted infrastructure
for transparent, flexible evaluation

Self-Hosted Infrastructure

EXACT

Frontend

Next.js

Backend

Django REST Framework

Task Queue

Worker

Storage

Local Filesystem

PostgreSQL

Support

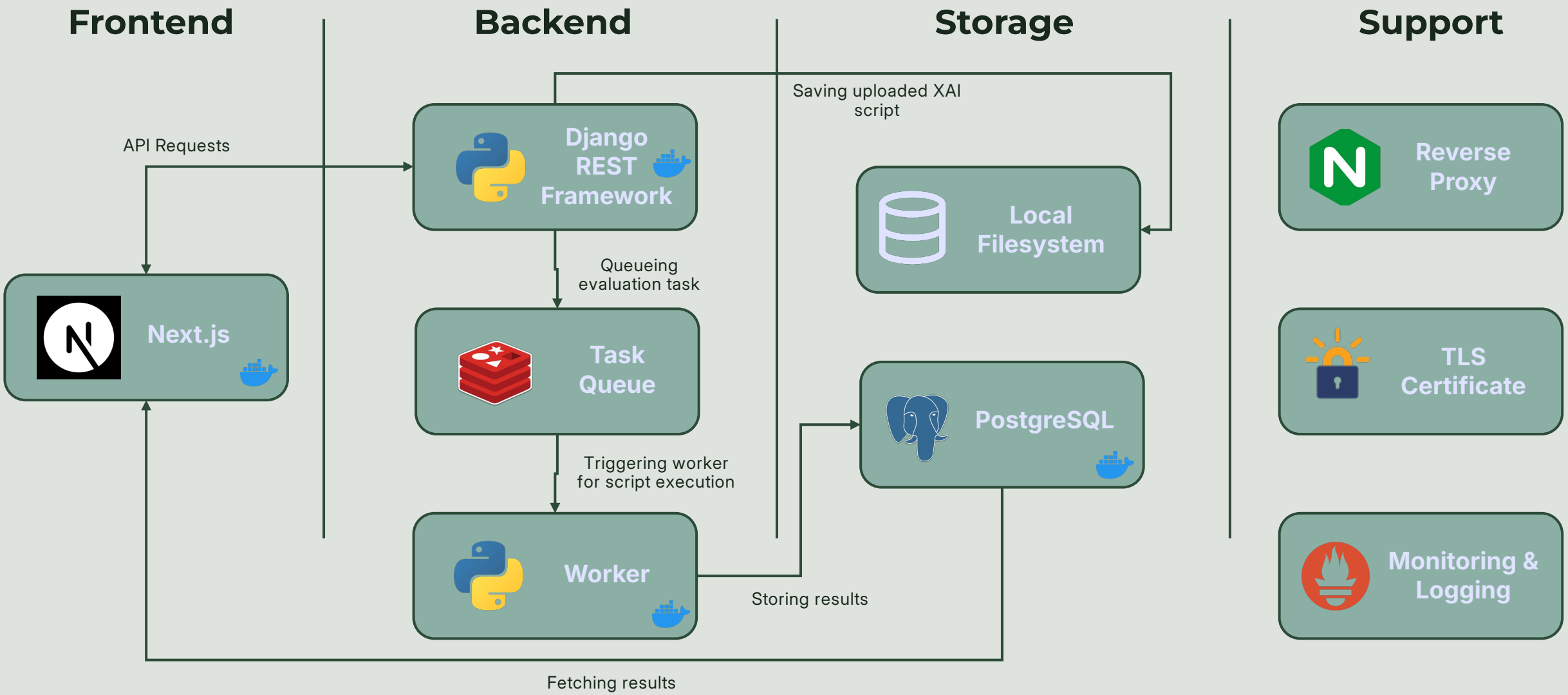
Reverse Proxy

TLS Certificate

Monitoring & Logging

Self-Hosted Infrastructure

EXACT



Summary Self-Hosted

EXACT

Advantages

Flexibility

Full control over architecture and services enables deep customization across all layers of the system

Cost Efficiency

No platform or service fees; only pay for actual compute, storage, and traffic usage

Tech Independence

Vendor-agnostic setup avoids lock-in and facilitates long-term maintainability

Tooling Familiarity

Builds on established open-source tools such as Docker, PostgreSQL, and Redis

Disadvantages

Operational Overhead

Manual provisioning and maintenance of databases, job queues, and file storage

Scalability Complexity

Horizontal scaling, high availability, and load balancing must be managed independently

Security Responsibility

TLS, authentication, and secret management must be implemented and monitored in-house

Monitoring & Observability

Requires integration of custom monitoring stacks (e.g., Prometheus, Grafana, Loki)

Closing Statement

EXACT

For professional or enterprise-grade deployments, I recommend using Azure (or alternatively AWS), as it offers built-in scalability, security, and monitoring – with minimal operational effort.

However, for academic settings, research prototypes, or environments where local server infrastructure is already available, a self-hosted setup can provide more flexibility and cost control – at the expense of increased maintenance responsibilities.