Architectural Decision Record

Title: 011: Use of the microservices architectural style for the Make the Grade system

Status: Proposed

Context:

Make the Grade is a test-taking system that needs high levels of responsiveness, fault tolerance, elasticity, and data integrity. Because there are separate parts of the system (admin, reporting, grading, and test taking) that require different architectural characteristics, a distributed architecture is appropriate. The two choices are microservices and event-driven architecture.

Decision:

We will use the microservices architectural style.

Microservices provides the necessary fault tolerance, elasticity, and scalability.

Performance deficiencies and high responsiveness needs are addressed through minimal inter-service communication, caching to minimize data retrieval needs (student information, test questions, and test answer keys), and asynchronous communication for automatic grading and storing students' answers.

Data integrity (preventing data loss) is addressed by using persistent queues between the Capture Answer and Automatic Grading components, along with client acknowledgment mode in the Automatic Grading component, to make sure that each student answer stays on the queue until it is persistent in the Student Answer Database.

The test administration functionality will be a single microservice that combines the test scheduling, test maintenance, and student maintenance functionalities. Reporting will be a single microservice as well.

Consequences:

Technically partitioned teams will need to be reorganized into cross-functional teams and will work in parallel in order to finish the system in six months.

We will need to use in-memory caching to address the system's performance, elasticity, and data sharing needs.

We will need additional infrastructure to support microservices: specifically, a service orchestrator like Kubernetes and a more effective CI/CD deployment pipeline.