Admissions system background
Reading for 17-313 design document practice, October 9, 2019

Background: Assume we have built a new admissions system from scratch. The new system supports collecting general application information applicable to all programs, which applicants will only need to complete once; applicants then apply to specific programs by completing program specific supplements (with questions selected by the admission committee for the program).

In this new system, the entire platform is virtualized, including networking, server, and storage (via a VMWare cluster). Each physical server hosts several virtual machines; one central server, the VirtualCenter management server, is responsible for assigning the virtual machines to the servers.

This choice enables several key quality requirements:

- Cost. We reuse existing CMU vSphere installation, an efficient use of existing resources; we also avoid cloud costs.
- Security: storing data on premises is more secure (and FERPA compliant) than a cloud storage system, and ensures direct control over the data; deploying on premises minimizes the attack surface.
- Modifiability: the virtual cluster does not assume specific hardware; VMs can be created, destroyed, or provisioned quickly, adapting to changing needs.

Services: Figure 3 shows the integration and reuse of existing CMU services:

- Oracle. CMU has an existing Oracle installation that is already maintained by dedicated staff members, backed up, and redundant.
- Payment Processing , for application fees.
- Authentication For applicants and recommenders, we will be using a Single Sign-On (SSO) service, making use of their existing university credentials; we will use CMU's login service for reviewers and admissions staff.
- STMP Mail Server, of which CMU has 12.
- CMU's SIO, to automatically create new student profiles for admitted students.

We secure the TCP communications between our API server and existing services using SSL.

Deployment. We implemented infrastructure automation to ensure availability and performance. Our infrastructure automation architecture consists of a single virtual instance running Ansible Tower (an orchestration tool), interfaces with underlying services/virtual instances (Figure 5).

Figures taken from a previous year's 17-313 (anonymized) homework submission.

Admissions system background
Reading for 17-313 design document practice, October 9, 2019

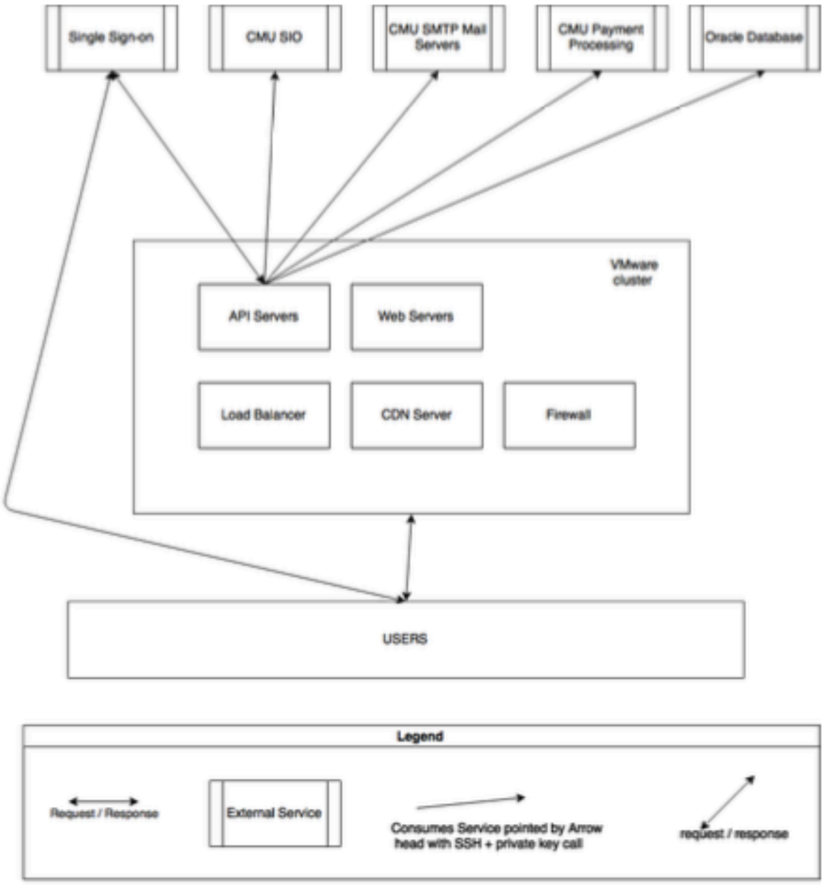

Figure 3: Service Architecture

Figure 3. Service Architecture

Figures taken from a previous year's 17-313 (anonymized) homework submission.

Admissions system background
Reading for 17-313 design document practice, October 9, 2019

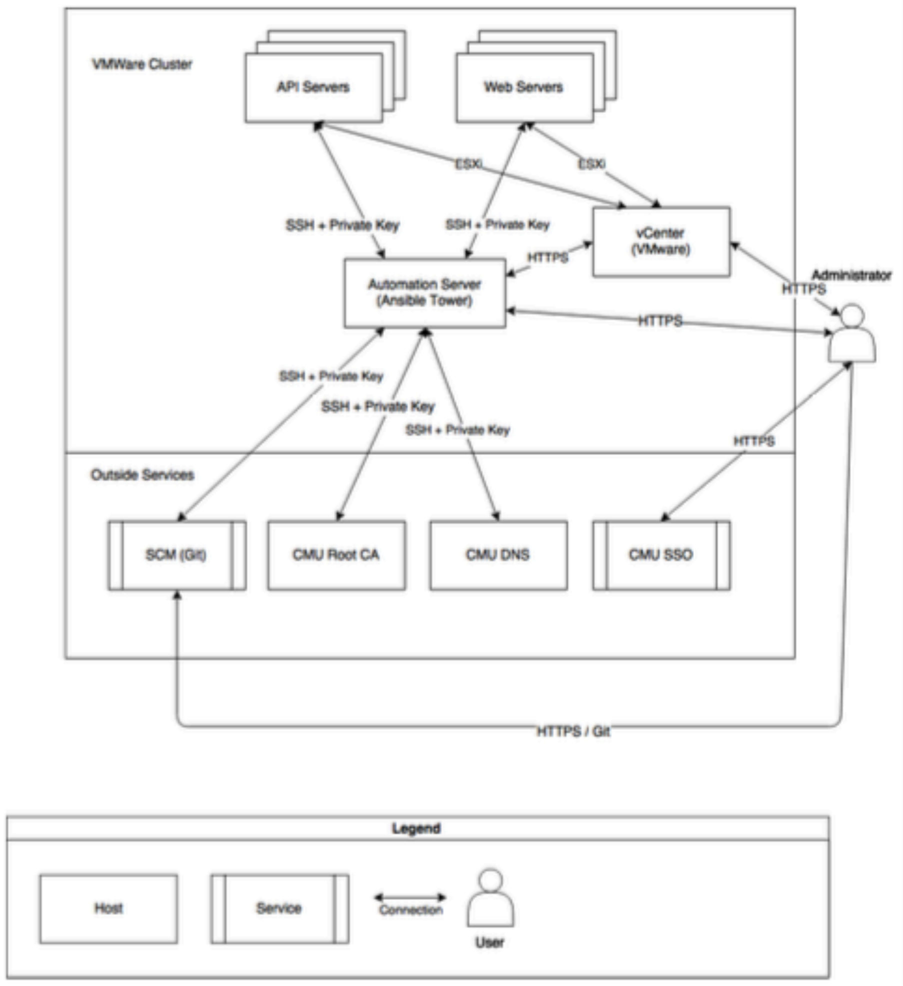

Figure 5. Deployment Architecture

Problem: Our new multi-program admissions system is very successful at CMU, and a number of other organizations are interested (e.g., other colleges in and around Pittsburgh). These organizations have similar functional needs as CMU. However, the new customers have different environments that must be adapted to, such as different internal account systems and SIO handling.

Our present architecture is modifiable by construction. Here, we consider deployment. We have several competing concerns:

1. CMU does not want to host the system for the new customers.
2. We must be very careful with security, once we leave the safety of CMU's [ed note: nonexistent] firewall.
3. We can no longer have significant scheduled maintenance downtimes, as we move into new time zones.

How should we stage our multi-site deployment, given these concerns?

Figures taken from a previous year's 17-313 (anonymized) homework submission.