AV API Initiative

AVEngineers - BYU AV Services

Vision and Goals

“When an instructor walks into a classroom, the system knows the instructor’s preferences and what class they’re teaching; in-class technology will function accordingly, providing a dynamic, unique and catered learning experience”

Unified RESTful API that exposes all attributes and functionality of AV equipment over IP network.

- Classroom control via local touch panel
- API control via any connected IP network
- Programmatically handle monitoring, administration
- Gather all events in a DB: metrics, etc.
- Possibilities we haven’t thought of yet...

The Current Jungle

—

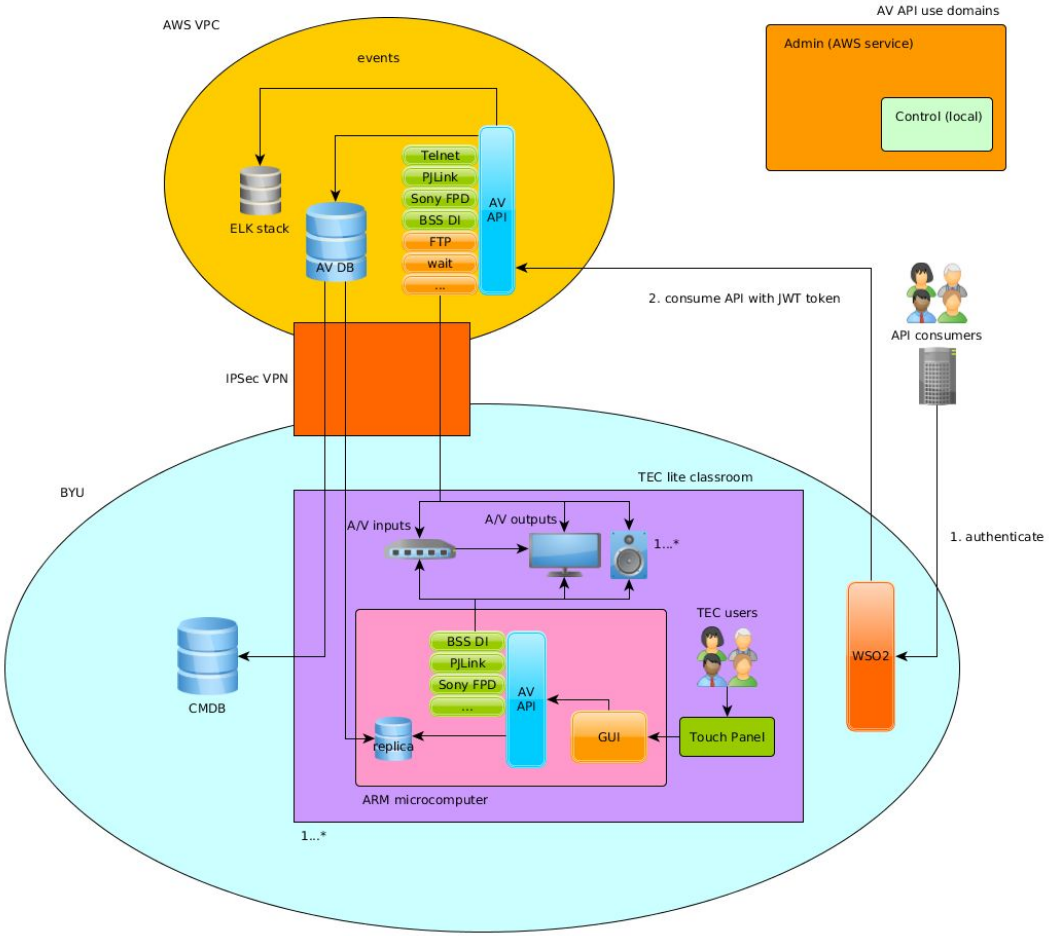
- Wide variety of devices, and how they are controlled: from RS-232 to various protocols across IP
- Sector changing fast; some vendors starting to move towards RESTful, but still a big mess

Start with low-hanging fruit; begin with easy cases first, eventually work towards comprehensive coverage.

System Architecture

- Modular to accommodate disparate devices and protocols - > microservices
- Best practice: each service has very specific task. KISS (keep it simple stupid)
- ‘Classroom logic/paradigm’ presented in a fronting AV service.
- AWS Beanstalk, other AWS services for ease of deployment and management
- Autonomous classroom systems.

High-level System Architecture



Software Architecture

GO programming language: well suited for web microservices.

Deployment Pipeline

Continuous integration; fully automated pipeline from check-in to production

- Code is checked in to public GitHub repo
- GitHub fires a git hook to CircleCI
- CircleCI compiles and tests builds; pending success, builds docker containers and pushes to Docker Hub
- CircleCI tells AWS Beanstalk to tear re-build application(s) with new containers
- New containers are also deployed to Raspberry Pis in classrooms

Monitoring and Admin

AWS Beanstalk takes care of much of the web application management and infrastructure for us:

- maintain application health
- abstracts underlying resources
- dynamic auto-scaling, etc

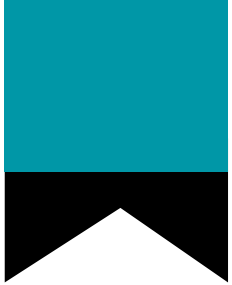
Custom Slackbot that notifies relevant channels if application is sick

All events passing through the API are pushed to our ELK stack: failure notifications for operations, metrics, historical data, etc.

Now and the future

- Several functional microservices: projector, Sony TV, etc.
- Recently ran a demo demonstrating basic control of classroom via API on ARM microcomputer (input change, on/off, volume)
- Well-defined RESTful interface
- Functional vertical slice
- Controlled access via WSO2

- Continue growing the devices we can talk to
- User-based authentication - AD
- Securing intra-microservice communication
- Update mechanisms for syncing AV DB with CMDB
- Many related projects that will leverage the API...



Documentation

Using GitHub repo wiki - natural place to document. We have pages on: system/software design, deployment pipeline, etc. Intention is to store team’s collective knowledge here

Also utilizing the GitHub issues system for tracking/assigning mainly tasks, but also issues.