
JNUC
20 Nashville 24
Scaling and Simplifying
How Ohio State Used Jamf Pro to Streamline Device Management

Title slide

Who we are

Marty Winders
Associate Director, Client Services and Engineering
The Ohio State University - Office of Technology and Digital Innovation
I am the service owner of the shared Jamf service at OSU

Ray DeMay
Senior Endpoint Engineer
The Ohio State University - Office of Technology and Digital Innovation
I am the Jamf platform administrator

The leadership at Ohio State had a vision to provide an iPad to every incoming student and wanted to see this to fruition through the CIO, the CIO came to Marty to ask if we had the tools capable for supporting this initiative. The central IT organization was not using Jamf at the time but with an undertaking of this caliber it was identified that we must adopt the Jamf platform. While adopting the Jamf platform a vision was given to offer a consolidation of Jamf Servers as well as to offer Jamf to those areas that could not afford it our did not have the acumen to support it themselves.

Shared Jamf was then implemented as a service to help consolidate the disparate Apple device management across OSU. Our users supported by the OCIO team was migrated from Ivanti to Jamf while 15 other departments decided to collaborate into the Shared Jamf offering and decommissioning their own server. This included departments such as College of Engineering, Arts and Sciences and our Athletic Department.

While the digital flagship initiative did not last after COVID, the Shared Jamf environment continues to grow and be an effective solution across campus.

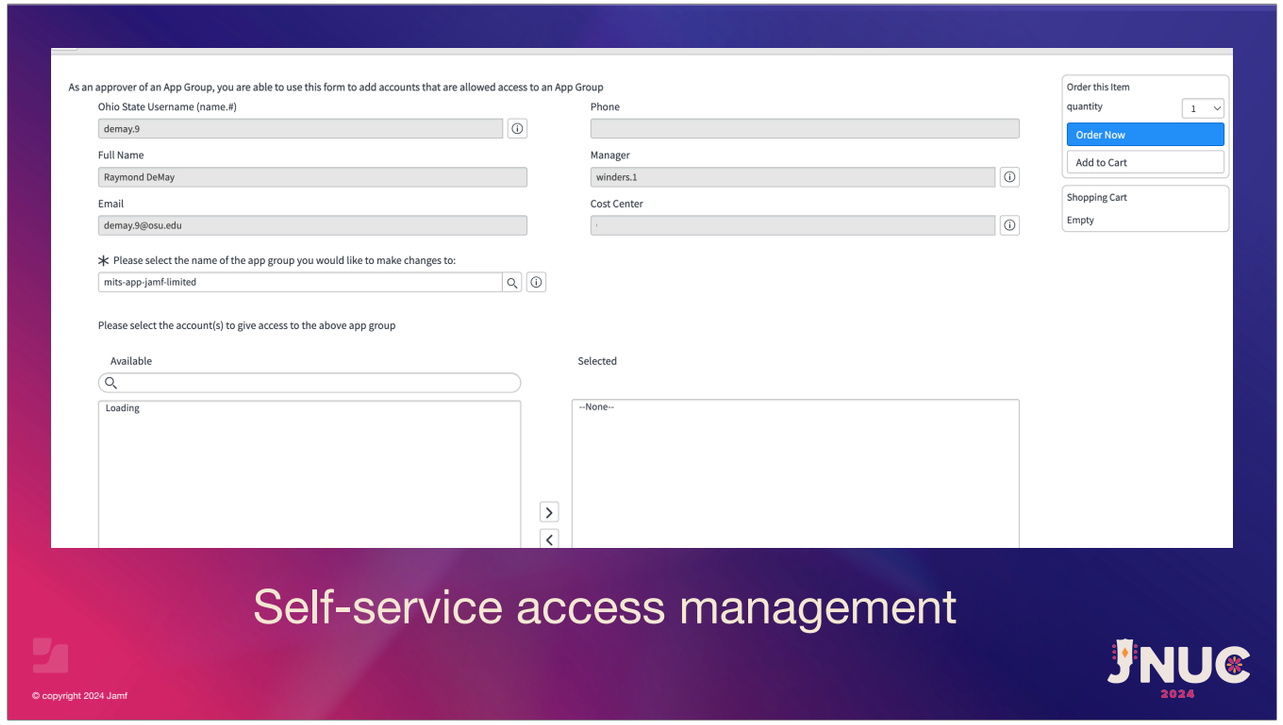
Shared Jamf Service Overview

- One site for each college/business unit
  - Some exceptions to this
- Three tiers of access
  - Site administrator - also has permission to upload packages and scripts
  - Limited - a custom permission set for desktop support/helpdesk
  - Auditor
- API account(s)
- Self-service access management

JNUC 2024

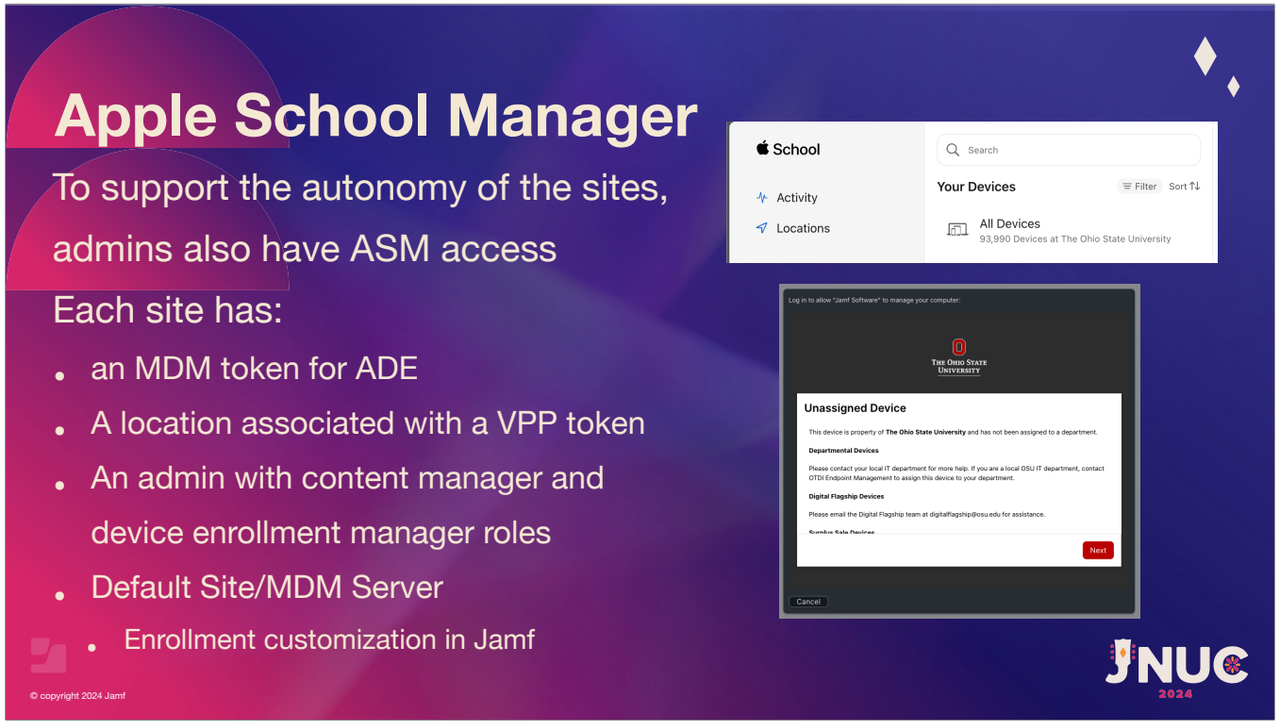
The basics of the shared Jamf are as follows:
Each customer gets one site, with some rare exceptions (marching band site, we have 4 sites)
There are three tiers of access. Site admins get full access to their site, plus package, script, and printer uploads at the top level. Limited is basically create/read/update only. Auditor is read-only (a built-in permissions set).

Admins could request a read-only or read/write API account (some had both)



This is one of the self-service request catalog forms that was built in ServiceNow to handle adding and removing users from the AD groups that Jamf access is derived from.

InCommon Grouper plays middleman, taking the AD groups and tying them to the SSO groups that are passed through Shibboleth authentication. That then grants the user their access in Jamf.



Along with Jamf, there is a shared model for ASM, albeit not as siloed (ASM limitation). We also have accounts for the site admins and some other technicians in ASM. Each site has an ADE and VPP token.

We assign a default server in ASM, which points to an “unassigned” site in Jamf. This site has an enrollment customization that is limited to one LDAP group that nobody is in. That way the device can’t finish enrolling. The messaging tells the person trying to enroll the device of their next steps.

Service Licensing and Cost Model

OTDI owns and maintains the Jamf service. We meet with our CSM, open support tickets, schedule upgrades.

Costs are shared between departments, based on number of devices in each site. We collect counts near renewal time, asking for customers to account for any expected growth.

Cost savings by going to Jamf Cloud and no egress costs. Cost shared between the departments by counts and that our organization maintains that information, purchases all licenses, maintains relationships and costs shares with the other units. By consolidating we also maximized our spend and our price breaks with Jamf.

Platform is owned and maintained by OTDI (you) along with our site and all of the top level configurations in partnership with College of Arts and Sciences

```python
for id, site_name in site_ids.items():
    site_computer_count = site_mobile_device_count = site_apple_tv_count = 0
    for i in range(0, 3):
        site_computer_object_api_endpoint = f"{jamf_server}/api/v1/sites/{id}/objects?page={i}&page-size=2000&sort=objectType%3Aasc&filter=objectType%3D%3D%22Computer%22"
        site_computer_objects = requests.get(
            site_computer_object_api_endpoint, headers=headers
        )
        site_mobile_device_object_api_endpoint = f"{jamf_server}/api/v1/sites/{id}/objects?page={i}&page-size=2000&sort=objectType%3Aasc&filter=objectType%3D%3D%22Mobile%20Device%22"
        site_mobile_device_objects = requests.get(
            site_mobile_device_object_api_endpoint, headers=headers
        )
        site_apple_tv_object_api_endpoint = f"{jamf_server}/api/v1/sites/{id}/objects?page={i}&page-size=2000&sort=objectType%3Aasc&filter=objectType%3D%3D%22Apple%20TV%22"
        site_apple_tv__objects = requests.get(
            site_apple_tv_object_api_endpoint, headers=headers
        )
        site_computer_count += len(site_computer_objects.json())
        object_ids += [item["objectId"] for item in site_computer_objects.json()]
        site_mobile_device_count += len(site_mobile_device_objects.json())
        site_apple_tv_count += len(site_apple_tv__objects.json())
```

We collect current site counts in April with a Python script, then ask each customer to confirm the counts, encouraging them to clean up old devices before final counts are sent for renewal.

https://github.com/raydemay/jamf-site-counts

JNUC 2024

I wrote a Python script to get me device counts in each site, by type (computer, mobile device, Apple TV)

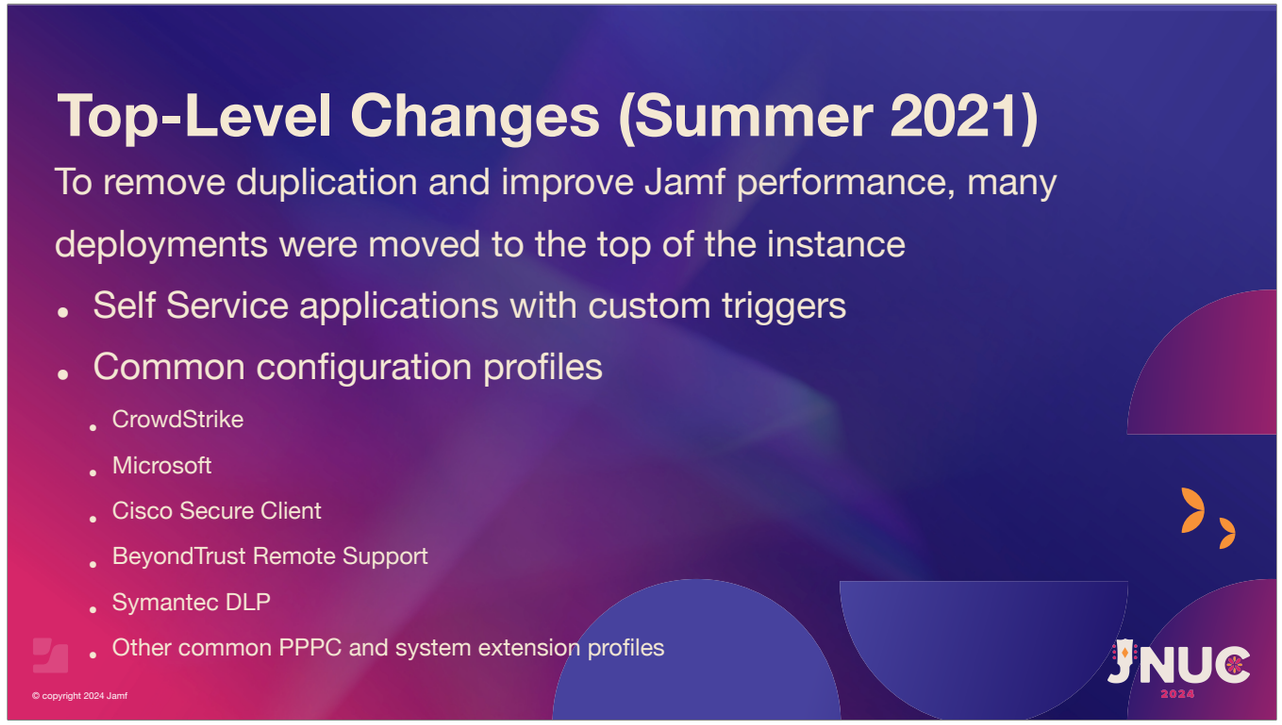
Early Challenges

Move from on-premises to cloud was ***rocky***

Shared service didn't have a lot of sharing

- Every site had the same kinds of config profiles and policies
- Package and script duplication
- No mechanism to share ideas
- Performance issues

Some of the early bumps in the road with the service. Units were using the service correctly, or at least in the way that they were sold, but it turns out that the approach was wrong from the beginning. Moving to Jamf Pro Premium Cloud helped a bit, but the actual move was a disaster.

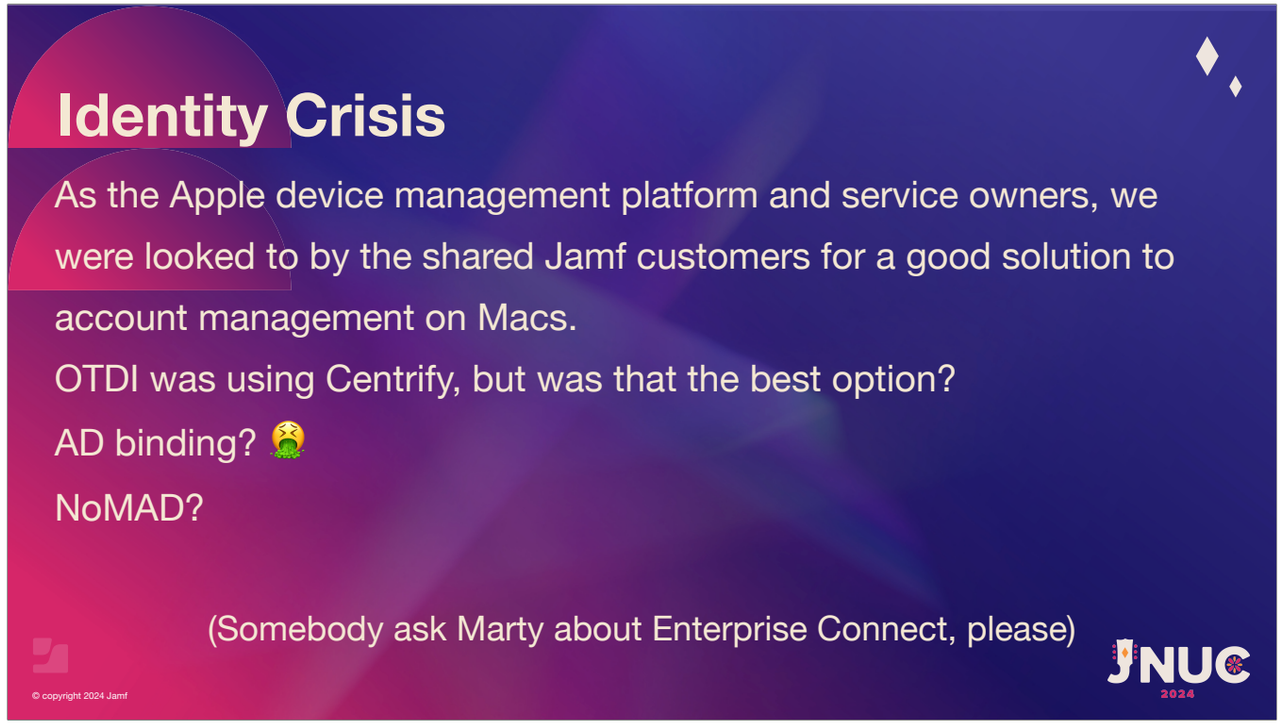


To both improve performance in Jamf, and to provide more value to customers, some common policies and config profiles were moved to the Full Jamf Pro level and scoped to all devices (with exclusions)

Providing top level Self Service in Shared Jamf allows us to eliminate redundancies in duplicate package/policy creation to provide a more streamlined management experience for users and admins

Shared configuration profiles meant that the most common config profiles weren’t being distributed 18 times, cutting back on the bloat in Jamf.

We still can’t create a full Jamf Pro summary, even after all of this work!

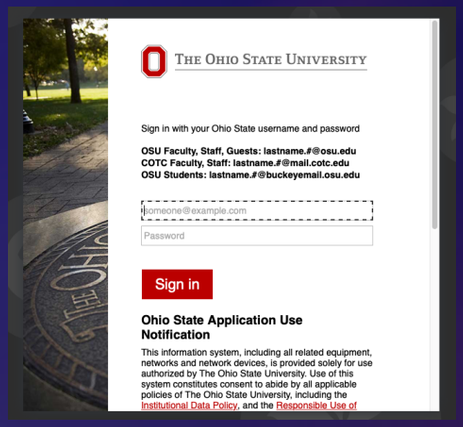


OSU is a mess when it comes to identity. Multiple domains, decentralized IT, different levels of administration skill and effort. As owners of the Apple device management service, we wanted to be able to prescribe a good solution for account management on Macs that worked for every use case

Jamf Connect

We landed on Jamf Connect to manage identities on Macs.

- Custom ADFS webpage with JavaScript to handle the different usernames
- Previously had to have separate configurations for faculty/staff and students, making mixed use devices impossible



This is Jamf Connect as it currently stands for OSU. We are working to move to Entra ID with password hash sync. Hoping to include privilege elevation, but that might not be ready for our uses *quite* yet.

Adding More Value (2022-Current)

I became the Jamf platform owner in January 2022

Changes since then:

- More top-level config profiles and policies
- 802.1x device authentication
  - Was only for OTDI at first
- More useful extension attributes

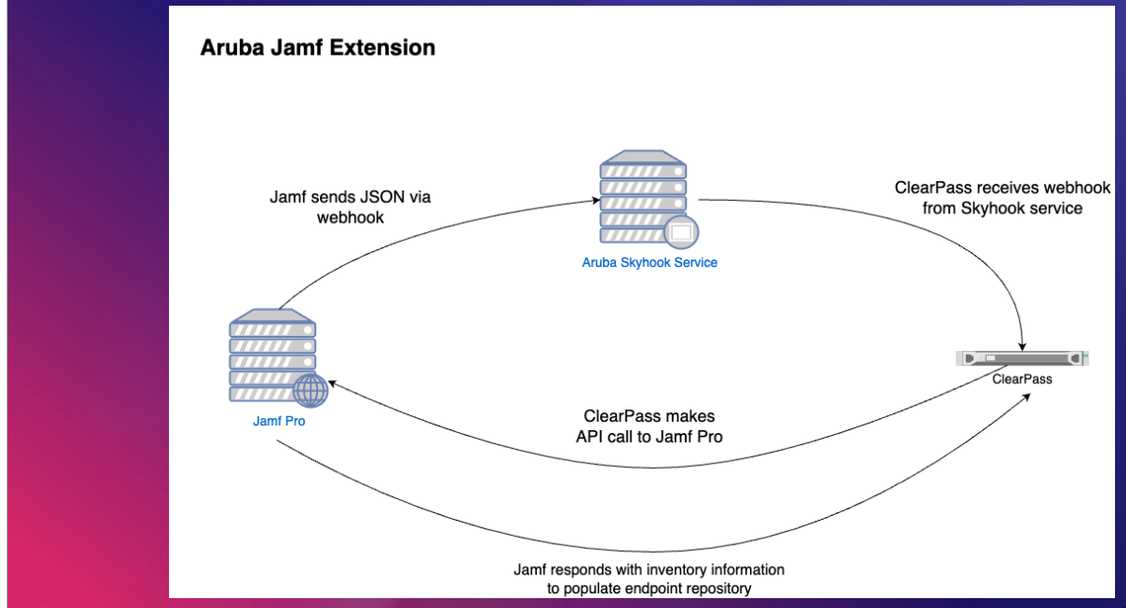
JNUC 2024

More changes since I took over. The big one is eduroam and 802.1x. Our 802.1x implementation is what I am going to cover in more detail in the next few slides.

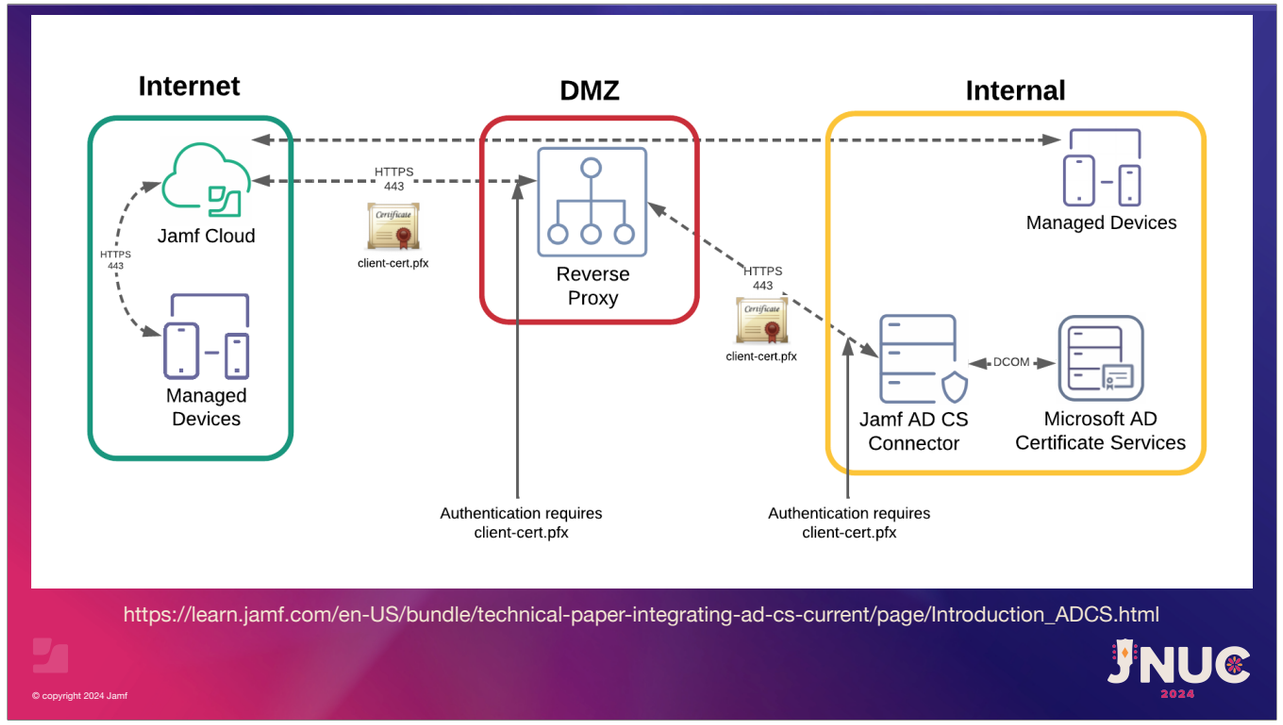
802.1x Device Certificate Authentication

JNUC 2024

Section slide to indicate the topic



This is my understanding of how Aruba’s ClearPass Jamf extension works with web hooks triggering the API calls back to Jamf



We utilize the AD CS connector for getting the 802.1x certificates. This is the diagram that Jamf provides in their documentation.

802.1x Device Auth

Single configuration profile for wired and wireless

- All mobile devices also have a wireless profile

Wireless is configured for eduroam

- https://internet2.edu/eduroam-global-roaming-access-service/

Certificate CN is $SERIALNUMBER@jamf.it.osu.edu

Certificate SAN is the serial number

- Don’t need to care about MAC address

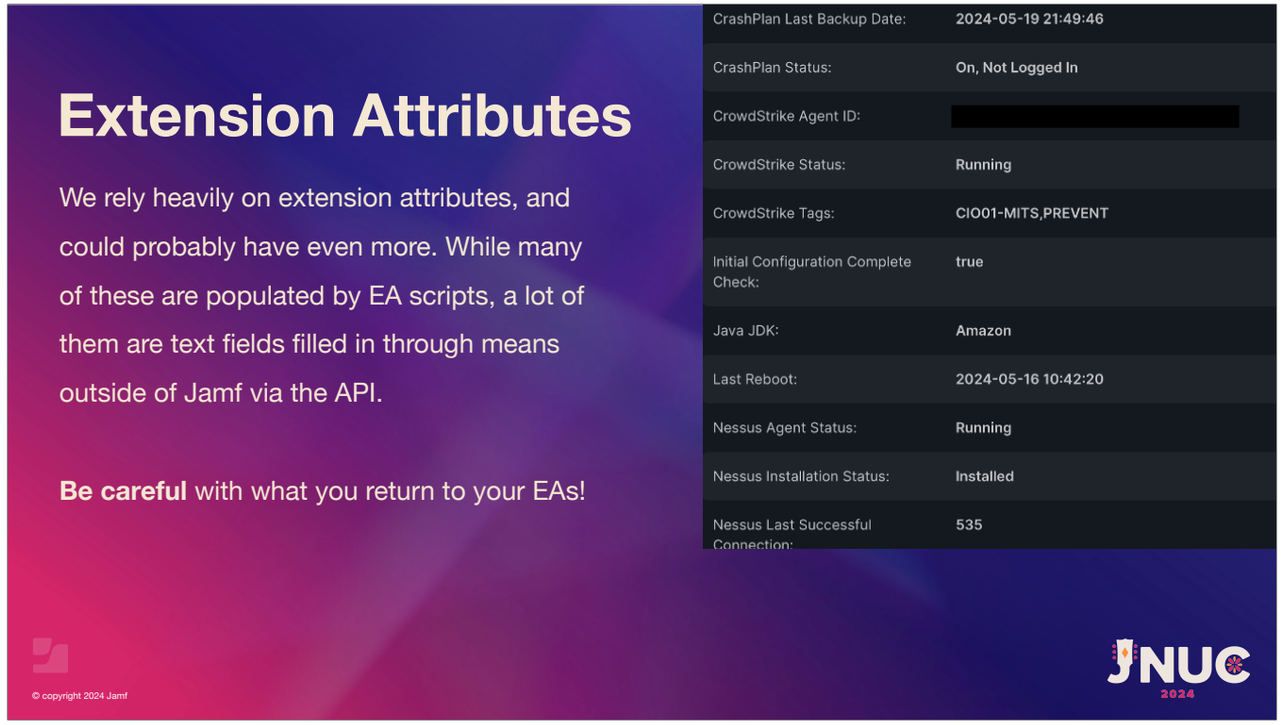
JNUC 2024

One last slide to cover the config profile side of the 802.1x implementation

Mentioning eduroam since this is something other higher education institutions should look into

API Usage in Sites

Section slide to indicate the topic



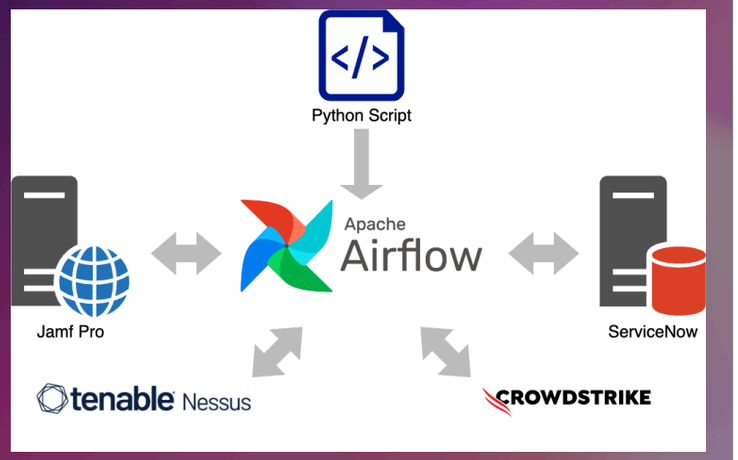
Extension attributes

We use these. A lot. From reporting on different agents on the device, to handling local admin and network authorization roles, and CMDB information. We learned the hard way to not try to overload EAs with a lot of data, and to be picky with what is added.

ServiceNow

We integrate ServiceNow into our device management using Python scripts that are orchestrated with Airflow

- Jamf
- CrowdStrike
- Nessus



ServiceNow integration

Instead of using the ServiceNow app, we have a custom API workflow utilizing Apache Airflow for orchestration. Python scripts run to get or post data via Jamf or other device management APIs. This image leaves out Active Directory, but there is a lot of AD integration for the Windows side of the house

Arts and Sciences is doing similar things with TeamDynamix and a “jss-api-worker” server running Flask

ServiceNow

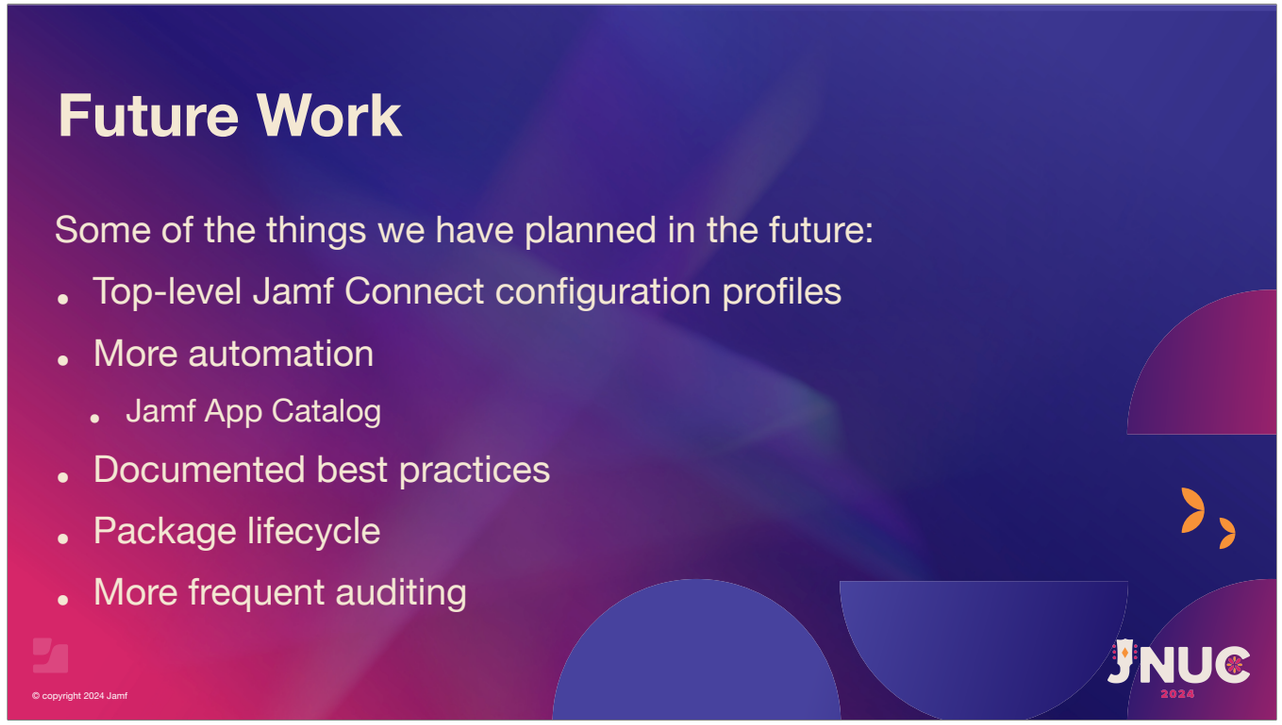
In the sites that we directly manage, we use ServiceNow in Jamf for:

- Local admin requests (Privileges)
- Network authorization roles
- Software update cycles and exceptions
- Inventory info based on the CI
  - Asset tag
  - Assigned user
  - Status

More details about what we do with the custom API workflow. This is mainly about reading/writing EAs. I will mention our plan for CrowdStrike here.

Looking Forward

Section slide to indicate the topic



Some of the future plans we have for the service

We want to put the basic Jamf Connect settings at the top
Utilize Jamf App Catalog in self service at the top
Provide a best practices document specific to shared Jamf

Building Best Practices

In our shared model, the site admins are basically given full reign over their site, but there are some considerations:

- Configuration profiles used by everyone go to the top to lower total number of payloads
- Packages should be named with a unit prefix only if the package is specific to the unit (license files, custom configs, etc.)
- Don’t deploy bad policies (all computers, ongoing frequency)
- Extension attributes have to be approved and uploaded by service owner

JNUC 2024

These are some of the best practices that we have come up with so far. There is a document that we are working on with Arts and Sciences that will have everything to keep in mind

This is our wishlist from Jamf to improve using sites in Jamf Pro.

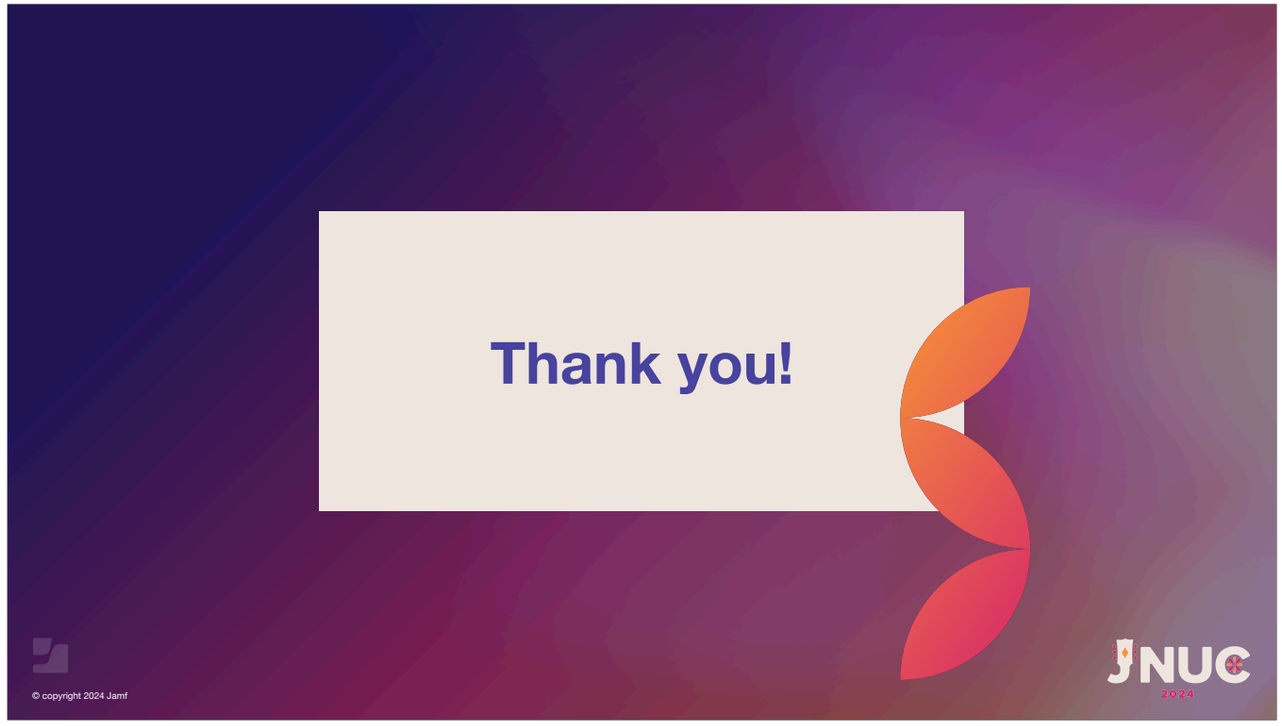
Sharing Our Work

We have had the opportunity to share this with other universities as they look to better manage their Apple devices

- Texas A&M University
- University of Kentucky
- Kansas State University
- Big Ten Schools
  - Penn State University
  - University of Illinois
- Ohio Universities

JNUC 2024

The schools we have had the pleasure of sharing how our shared Jamf model works


Thank you!
© copyright 2024 Jamf
JNUC
2024