Euan Murray, Head of Learning Spaces Technology. May 2019

[www.teachingspaces.ed.ac.uk](http://www.teachingspaces.ed.ac.uk)

**The potential for Teams as a central hub for support communication**

Providing timely, proximate and convenient support is crucial to help our academic staff overcome technical issues when operating the AV and IT systems in Teaching Spaces. Over the past few years, our Teaching Space Technology Support Service has undergone a rapid expansion. This has been driven by the desire to bring more and more of our vast and varied Teaching Space estate under central management, funding and support. In 2015 we were operating 220 spaces over 3 support sites, by 2020 this could be in the region of 550 spaces across 9 separate sites. This will make us one of, if not the largest central support service in the UK. To enable this change we need to find efficiencies in how we operate, both in terms of technology and the model we use to triage and deliver that support.

The Central Area physical response times as of Summer 2019



**The development of remote support tools**

The direction of travel in the sector is a move to harness the IP connectivity of modern AV systems to provide both proactive and reactive support at a distance. These tools allow us to fix a range of user awareness issues very quickly and easily. Even if an issue cannot be fixed remotely and still requires a technician to be despatched to the venue, these remote tools allow us to collect valuable information that speeds up either the resolution or the escalation of an incident.



Extron’s Global Viewer Enterprise

**Adapting our service desk support model**

Our current support model is to field telephone calls from venues, either from mobile or a fixed telephone in the room. Our service desk will then ask the user some very basic questions, if this doesn’t resolve the issue, they keep the user on hold which they phone the local team base to find a technician to despatch. They are then kept on hold until a technician is found. If one can’t be found at the local base, a user is advised that a technical will be found as soon and possible and the service desk then start calling the mobiles of the technicians stationed at that campus. This can lead to a drop in the response time if the team is very busy and out on existing callouts, events, and maintenance tasks.

The downside when using remote tools is the added time taken by our service desk to access the equipment and potentially talk the end user through any steps they can take. This process takes time and only adds to the delay in the resolution time if a technical still has to be found. In order to deliver this efficiently, we needed to find a way to communicate with technicians without holding up the telephone call with the user.

The obvious solution to this was to wrap communications between our service desk and technicians into the same remote support interface we are operating while still on the phone. These messages have to be quick, seamless and easy enough not to distract the service desk from attempting a remote fix.

Phone call

Traditional support model

Remote support model

Basic steps

Technician comms

Phone call

Remote

support

Leave from base

Arrive at venue

Technician

comms

Leave from base

Arrive at venue

Simplified summary of the differences in support model

**Moving from telephones to notifications**

The major change in support model that underpins the new service is the move to notifications as an alert system for technicians rather than a phone ringing. This allows us to break free from the manual physical aspect of calling individual sites, we can now alert the whole team from one button within our remote support interface.



****

Desktop client

Mobile client

****

The link from GVE allowing one-touch broadcast of alerts within Teams

Our remote support interface is using AutoIT to launch various interfaces to AV equipment. We are using this same scripting interface to launch a PowerShell script that initiates a JSON message card POST to Teams through the Graph API using an application with delegated access [1]. This post then appears in Teams as if it was manually created by the service desk user.

[1] <https://docs.microsoft.com/en-us/graph/api/channel-post-chatmessage?view=graph-rest-beta>

**The Virtual Campus within Teams**

The success of notifications as an incident alert system depends on the isolation of these notifications against any other a technician will see. The way we are isolating alerts is by using channel notifications within Teams. This works in the same way as following groups in Facebook, a technician will get a notification on the creation of an alert but will only receive further notifications of activity on that thread if they replied to it.

For notifications to work, a technician should only see notifications for alerts in the campus in which they are based in. This is made more complicated by our technicians working on a shift rota and moving between sites each day. This is where things need to get a bit more homebrew.

Each of our support sites have their own virtual campus “teams”. We have built an app using the PowerShell Teams module [2] that allows our service desk to move the technicians between sites quickly on demand. This will be used every morning to put technicians into either their respective teams or remove them from all campus teams if they are out of office. This ensures that the technicians don’t have to manually move between teams, as this would be a point of failure for notifications.

[2] <https://docs.microsoft.com/en-us/microsoftteams/teams-powershell-overview>



Teams Campus Manager app built with PowerShell

**What’s next?**

We are planning a launch of our new service desk for September 2019 in time for the start of the next academic term. Although there are still technical and procedure issues to work through, Teams offers us some fantastic opportunities and a way to quickly scale our communications.

Some of the future potential we have started to identify:

* Automating activity within UniDesk, our incident management system
* Using USB devices to help with notification visibility: <https://luxafor.com>
* Using the “Shifts” app within Teams to manage our shift rota, which is currently manual
* Managing holidays and finance requests through Teams using Flow