UNIVERSITY OF CAMBRIDGE
Department of Computer Science and Technology

ITSC_AGENDA_23/06/2025

IT Strategy Committee, Department of Computer Science and Technology
23rd June 2025 at 14:00
GS15, William Gates Building

AGENDA

Membership

Richard Mortier, Chair [RM]
Mark Cresham, Secretary [MC]
Tim Jones, UTO Rep [TJ]
Rob Harle, Director UG Teaching [RH]
Helen Francis, PSS Rep [HF]
Nic Lane, GPU resourcing strategy lead [NL]
Thomas Sauerwald, Deputy HoD [TS]
Daniel Porter, IT Support Manager [DP]
Sam Nallaperuma, Research Staff Rep [SN]
Abraham Martin Campillo, UIS Rep [AM]
Malcolm Scott, IT Infrastructure Specialist [MS]

1. **Welcome & apologies [RM]**

2. **Approval of the Minutes of the previous meeting [RM]**

   Unconfirmed Minutes of the meeting held on 28th April 2025 are attached for approval.

3. **Matters arising [RM]**

4. **Actions from the previous meeting [RM]**

   (i) GPU Upgrades [NL]
   - NL will explore potential usage patterns for Lightning, to give clearer idea of potential pricing structure.

   (ii) Network Upgrades [MS]
   - MS will update the committee on the progress of firewall renewal. An update on this is covered in the document “IT Strategy Committee report: Infrastructure (June 2025)” that is appended to this agenda.

   (iii) Separation of Department Websites [RM]
   - RM to remind mgk25 of the need for alternative proposal and present the findings to the committee for evaluation. The document “Some notes on the departmental web presence” (from mgk25) is appended to this agenda.

5. Standing items

(ii) * UIS update [AM]

- AM to provide a verbal or written update on any relevant developments from UIS, including updates on the door locking system.

(iii) IT Services updates [DP, MS]

- DP and MS to update the committee on recent developments from the CST IT Services: Support Team and Infrastructure Team, respectively. The document "IT Strategy Committee report: Infrastructure (June 2025)" is appended to this agenda.

(iv) Decommissioning of Legacy Services [DP, MS]

- DP and MS to update the committee on progress of decommissioning. Including the potential decommissioning of old printers (DP), desktop.cl.cam.ac.uk (DP) and the managed Wikis (MS).

6. Main business

(i) DS-Print: printer access [RM]

- RM to facilitate a discussion based on the document "Concerns about restricting printer access to DS-Print" (from mgk25), that is appended to this agenda.

7. Any Other Business [RM]

8. Date of next meeting(s) [RM]

(i) Confirmation of the date, time and location of the next meeting(s).

- TBD

Some notes on the departmental web presence

Background

The Computer Laboratory had a website continuously since November 1993, served as static files, from a departmental NFS filespace that itself has existed since the late 1980s.

Serving a web site as static files has numerous advantages, in terms of performance, security, ease of maintenance, ease of backup, ease of long-term preservation, modularity and flexibility. The service has been hugely popular with local users. It allows use of an evolving mix of different content-management strategies and house-style tools and has shown excellent data retention: material published nearly three decades ago (e.g., research-project pages, conference sites or course materials from the late 1990s still remain easily accessible and maintainable today). It hosts seminar video archives just as easily as software repositories or large collections of PDFs. It is very amenable to easy automation using tools familiar from software engineering (e.g. build scripts, makefiles, CI/CD). The hierarchical file tree makes it very easy to repeatedly delegate responsibility for parts of the site.

As of summer 2019, our filer-hosted site offered 48254 HTML pages (7104 of which Ucampas formatted), 44804 PNG images, 8893 JPG images, 11236 GIF images, 30857 PDF files, 2329 PostScript files, 63811 JSON files, 1512 CSS files. etc.

In summer 2020, a tiny fraction of that content (a bit over 300 HTML pages taken from one Subversion repository) were moved to a Drupal 7 hosted service operated by the UIS to try out this central University service.

Our filer hosted web server (www.cl.cam.ac.uk) continues to serve the vast majority of the department's URLs (in the region of ~98%, excluding auto-generated Drupal URLs such as for individual calendar events, also excluding ~crsid personal web spaces on the filer).

But Drupal 7 is now end of live and the service will be discontinued by the UIS sometimes in 2026.

The UIS has recently outsourced Drupal hosting to a U.S. company (pantheon.io).

A new service called Cambridge Web Platform (based on Drupal 10 or 11) is under development, but has been long delayed and currently still offers very limited facilities compared to even its predecessor. It is not going to be backwards

compatible with UIS Drupal 7, meaning that web content would have to be manually ported to the new system, likely leading to a loss of version history and Drupal-7 specific functionality.

The 2020 split of the departmental web site across two services and domains has been quite disruptive: there is now a complex and difficult to maintain tangle of redirects in place, there are now two independent navigation trees that we manually try to keep in sync, web search and navigation has become very ineffective across both sites, and it is unclear to external users what the two domain names, cst vs cl, even signify in our URLs.

The UIS, following repeated service shutdowns without creating compatible and history-preserving successor options, has little track record on long-term continuity, i.e. offering platforms that span entire academic careers. (This may be hindered by high staff turnover, increasing dependency on external suppliers and short-term consultants.)

Proposal

We should commit to maintaining our filer-hosted site as a long-term resource that is able to preserve the public record of entire academic careers, aiming for vibrant use and growth well into the 2040s.

We should plan our web strategy accordingly with such timespans in mind. Keeping our web presence predominantly as a static file tree, maintained via open-source tools and standardized protocols, will significantly help with fulfilling that ambition.

We should familiarise and encourage new members of the department to develop their web presence on the filer-hosted web site. (This doesn't appear to have happened since ~2020 and risks encouraging the kinds of project-specific SQL/CMS VMs that rarely survive the departure of the PhD student or post-doc who set it up.)

We should discourage adding new public content of long-term value on UIS Drupal, Sharepoint or other services whose state can't simply be preserved and transferred using standard POSIX-file-tree archiving tools along the lines of zip, rsync, scp or git.

We should start to migrate material currently hosted on Drupal 7 back onto the filer, starting with existing static HTML material aimed at in-house users, located under /teaching/ and /local/, in order to reduce the complexity of the redirect lists.

We should look into technical options for merging both the current cst and cl sites into a single domain name.

We should look into installing an in-browser CKEditor for editing filer-hosted version-controlled web pages (which seems to have been the main feature that attracted some users to Drupal).

The news-blog and calendaring parts of the site and the presentation of their headline summaries on the front page should probably remain on some SQL-hosted CMS, such as Drupal or WordPress.

Action items

I will need admin access to the current Drupal 7 CST site, to be able survey the current content and navigation structure, and explore potential interfaces and tooling for automating future content migration. Where do we keep local backups?

We should set up a web working group that brings together academics looking after site structure, content, tooling and training, with sys-admins looking after server infrastructure and user support.

Immediate priorities will be dealing with the upcoming deadlines imposed onto us by the ongoing shutdown of UIS services:

- Managed Wiki Service – by 1 August 2025, at least four departmental wikis affected

- talks.cam – may be retired soon unless we can help the UIS with refurbishing its code base

- Drupal 7 shutdown – hopefully not before sometimes in 2026

Markus Kuhn, 20 June 2025

IT Strategy Committee report: Infrastructure

Malcolm Scott
June 2025

Incident report on VM outage

Early in the morning of Monday 16th June, a RAM module failed in the storage server that holds VM disks the newer of our main departmental VM pools. All VMs on that pool, including several core services such as the SQL Server database and dbwebserver, stopped. The server was brought back online mid-morning. The same VMs underwent a planned shutdown after hours on Tuesday 17th June so that we could replace the failed RAM module and bring the server back to its full complement of RAM. No ongoing impact is expected from this outage.

We acknowledge poor communication about this issue, largely due to problems with our Instatus incident reporting website which the team usually relies on internally for updates about known issues.

We are using the free tier of Instatus's service. We have had a few problems with it lately. We don't currently know of a better alternative.

Recruitment

We will shortly be advertising for an IT Specialist (primarily a Linux system adminsitrator) at grade 8, aiming to interview in August.

ACS / Part III GPU usage report

This year almost all ACS / Part III projects that needed us to provide GPUs used a combination of HPC and our local shared development server. GPU usage was substantially lower than last year.

GPU VMs

Only two students used a departmental GPU VM – very much lower than previous years.

At the start of this academic year we bought two servers to host ACS GPU VMs, with four L40S GPUs each. This was, unexpectedly, much more than we needed this year. One of these servers turned out not to be required this year so was never fully set up; it was left turned off all year. We have recently discovered that it is actually faulty and are currently working with Dell to resolve this; it is a complex fault to diagnose.

HPC usage

24 students were given HPC access. 18 made non-trivial usage of their account.

Unlike all previous years, no student went substantially over the 300 hour allocation. One student asked for and was granted an extra 60 hours (but only actually used 21 extra, i.e. 321 in total). Another student used an extra 25 hours (i.e. 325 in total) without communicating this in advance.

For comparison, last year 40 students used HPC, and 15 went over the standard allocation of 300 hours – some quite substantially.

The total cost of ACS students' HPC usage this year (so far, pending any further usage e.g. to write up their work) was £1720.27, again much lower than usual.

Shared development server

At the start of this academic year, as well as the new GPU VM hosts, we also bought a dedicated development server for the ACS course with 8 low-power L4 GPUs. This was experimental as we had no experience of low-power GPUs being used for development – though last year we only provided P100 GPUs (initially just one, upgraded to four mid-year) which despite being high-power, high-end GPUs are in many ways slower than the much more modern L4 GPU.

We were prepared to redeploy the spare L40S GPU VM host as a development server if the L4 GPUs proved problematic, but did not do so.

I do not know how the students found the low-power GPUs; I heard no complaints (but on a one-year course, they would not have known anything different). We did however detect some jobs running on GPUs for a long time (hours/days) despite the students being instructed to not start long-running (>2 minute) resource-intensive tasks. Generally these tasks tied up a single GPU (of eight), preventing one-eighth of others from doing much using their randomly-assigned GPU.

It may be that the upgraded development server explains the reduced usage of HPC and personal GPU VMs this year. Other possible explanations (other than the lower student count, which cannot account for the observed change alone) are welcome.

Non-GPU usage

Three students needed VMs without GPUs. Two of these had projects involving Kubernetes clusters so they asked for 20 and 10 VMs respectively. The other student needed just one VM.

In order to accommodate the Kubernetes VMs, we took the GPUs out of the first generation of our GPU VM hosts (two machines, formerly with four GPUs each)

and redeployed them to host this fleet of 30 CPU-only VMs. The remove GPUs (Tesla P100) are obsolete but the servers themselves are still usable for the moment – though they are large and power-hungry quad-CPU machines.

We have limited capacity to provide large clusters of VMs or servers for ACS projects, though a couple per year is likely to be sustainable without specific investment for at least a few more years.

Several students (more than usual) made substantial usage of the HPC's CPU cluster. In all cases but one, they also used the GPU cluster. This is not a concern; CPU time is relatively cheap compared to GPU, and that cluster has generally been lightly loaded – though we did hear from one researcher recently (not an ACS student) that queueing times for SL3 on the CPU cluster are starting to become significant (e.g. 17 hours for a fairly small job).

GN09 cooling capacity

As an update to previous reports on GN09 being at capacity, the recent spell of hot weather has been right on the brink of exceeding the resilient capacity of the building's chiller. Temperature of GN09 has been largely unaffected, but whilst outdoor temperatures exceed approximately 25°C one of the chiller's two circuits runs at about 100% of its capacity.

One of the chiller's circuits has been awaiting repair since April. This was delayed due to scheduling difficulties and by two handovers (Estates changed the University's chiller maintenance contractor and our Building Services Manager also changed shortly afterwards), but Diana Gauvrit (BSM) is now actively working with Estates and Swegon (contractor) to resolve this as quickly as possible. We may be required to shut down servers in GN09 during the repair as we may not have cooling for a time – Swegon initially asked for a two-day outage, but we hope to avoid this.

Systems Management Policy

The central University has introduced a Systems Management Policy to which all multi-user systems and managed computers must adhere. This incorporates a very large number of specific, detailed technical requirements, many of which are contrary to common practice. This policy represents a major change to how IT must be managed in departments of the University; historically we have had the freedom to solve our problems in a manner appropriate to our local requirements and resources, but we are now being very heavily constrained.

The policy applies also to researchers who manage systems for other members of their group.

The deadline for compliance is April 2026. Not all of the technical requirements referenced in the policy have been published yet. Those that have, and those for which we have seen drafts, do not align with our existing system implementations at all. Complying with the policy would be a very significant undertaking due to the very large number of in-house systems in this department; for the current size of the IT infrastructure team, we would certainly miss the deadline and would be able to do little else than review and reimplement existing systems for the next several years. Some draft standards would be impossible to comply with at all given the size of our team (for example the Secure Development draft standard as circulated so far requires *all* software that we develop – including every trivial script to automate a task – to be reviewed by another developer and to have a full CI/CD pipeline). Several of the requirements are poorly-worded and unclear. One of the technical standards (Management of Technical Vulnerabilities) could be interpreted to read that we cannot use any software for which we do not have a support contract – i.e. most free and open-source software seems to be disallowed as such software expressly comes with no warranty or support.

It is possible to apply for exemptions for particular systems from particular parts of the policy via a detailed review process, first reviewed within the School of Technology and then by the CISO. Even enumerating the specific exemptions we would need to apply for will be a very substantial amount of work.

Furthermore, many of the existing problems that we know about and hope to rectify (several of which are security issues) do not actually appear to be in scope of the Systems Management Policy so are at risk of being deprioritised in favour of box-ticking.

We will be discussing this with Head of Department shortly.

Update on firewall project

We have not made much progress with the firewall replacement project, partly due to lack of time and partly due to the company that maintains VyOS, my primary choice of software platform, not engaging with us for some reason.

If we are required to prioritise SMP compliance, we will probably be unable to undertake any other infrastructure projects including the firewall replacement.

Concerns about restricting printer access to DS-Print

There was a recent announcement (5 June 2025) that our existing CUPS print server will be shut down, and that in future printer access will only be available through the UIS DS-Print service.

So I had a closer look at the latter. Summary: this is not an improvement. If fact, it is an unnecessary significant step in the wrong direction. I hope the IT Strategy Committee will urgently reconsider this change and support alternative printer access that does not involve DS-Print, PaperCut NG or PrintDeployClient.

What does the proposed change mean for users?

-

Loss of efficiency: At present with CUPS, each printer has a name and we can send jobs to our preferred printer. The printout starts immediately, or as soon as the job reaches the front of the queue. You visit to the printer just once, when the job is finished. With the DS-Print setup, there is only a single department-wide print queue, called CST_FindMe. You have walk to the printer, authenticate with your University Card, select the job you want to have printed on a touchscreen, and only then will the printout start. You then either have to wait or return a second time to pick up the printout.The CST_FindMe route may well be convenient occasionally for some users (e.g., working from home) and avoid the occasional forgotten printout sitting around. But forcing this mode of operation onto everyone costs additional time for users in a rush who quickly need to print out multiple documents while doing other things, e.g. during meetings with students or visitors (where it has been not uncommon to ask the student or visitor to pick up the printout). The latter will become significantly more cumbersome!

Loss of poster self-printing: The DS-Print service appears to be supporting only basic print functionality. In particular, it seems not intended for devices with more advanced options, like our HP inkjet poster printer “gum”. As a result, self-printing of posters is being withdrawn, and users who want to print a poster have to email service-desk and allow for 5 days (!) of turn-around time. Having helped students to make corrections to their poster late at night hours before departure to a conference, I am keenly aware that this alone is a huge restriction compared to the previous self-service offering.

- **Less security:** The authentication data on the very outdated RFID chip in the University Card is quite easy to clone, which some of our students have repeatedly demonstrated. Therefore it may be inadvisable to let confidential documents linger on CST_FindMe, because someone could use a clone of your University Card to print these documents anywhere in the building. In particular, CST_FindMe should never be used for printing test or exam questions!

- **Loss of Compatibility:** So far, it was possible for Unix, Linux, macOS, iOS, Android, ChromeOS etc. users to access our printers entirely with software already built into their operating system. Setting up a printer on Linux, for example, was pretty simple: either add a single-line configuration file (/etc/cups/client.conf) to get all printers from our CUPS print server, or go through a simple GUI dialog to pick the nearest printer's queue from an mDNS/Bonjour advertisement, or manually provide an Internet Print Protocol (IPP) URL for the desired print queue. The latter two options work on pretty much all operating systems.

- It has been common practice for short-term visitors with Eduroam access to simply temporarily add a corridor printer this way.

To use DS-Print, every user is instead now required to install a proprietary, closed-source software called PrintDeployClient:

- **Architecture limitations:** PrintDeployClient is only available for macOS and Windows, and after a long delay there is now also a test version available for certain Debian/Ubuntu versions, but only on x64 architectures. You won't be able to install it on an Android, iOS, Raspberry Pi (ARM), FreeBSD, Arch, Fedora, ChromeOS, etc. device.

Invasive: PrintDeployClient for Linux is a 160 MB large binary-only package that installs services such as a continuously running desktop tray app that appears to act as a non-authenticating IPP proxy.

No visitor access: PrintDeployClient requires UIS web authentication, so it is not accessible to short-term academic visitors who have Eduroam but no Raven login.

Unavailable on servers or via SSH: PrintDeployClient for Linux is a desktop tray app that establishes a tunnel to a UIS Server, over which printer jobs are submitted via IPP. You can't print without a desktop environment running, because without the GUI client there is no tunnel.

Lack of user separation: If you try to print on Linux without a desktop running (e.g., via SSH), the local CUPS server configured by DS-Print will hold your job until someone logs into the desktop. That job will then be sent to the print queue of that next desktop user! Even if it is a different person! (We've tried this on one of the ACS Linux machines, where the next user logging on inherited the previous user's print job, i.e. the print job changed owner! That seems a security risk on multi-user machines and indicates careless programming.)

Compatibility and workarounds: We used to occasionally solve problems with printing PDF documents that contain transparency by using the ability of CUPS to vary the processing steps applied before the document gets to the printer, e.g. use either the PostScript or PDF interpreter of the printer. This seems no longer possibly with PaperCut. Transparency-related artifacts are now always present with PaperCut and can no longer be circumvented via tricks such as "lpr -o raw file.pdf" (happy to explain the gory details omitted here). This is in particular an issue when printing student papers or scientific publications that contain plots produced by e.g. MATLAB, matplotlib, Plots.jl and similar scientific visualization tools that often make use of transparency and/or contain a lot of data points.

Loss of features: the printers offer dozens of configuration options and choices (e.g. colour settings, adjusting the fixer temperature/speed to different media types such as transparencies, labels or glossy paper, printing A6 conference badges, etc.), but the CUPS profile configured by PrintDeployClient (as displayed by

"lpoptions -l") on Linux only supports four very basic options with a small subset of choices: PageSize (no A6!), InputSlot, Duplex, Resolution.

Security: What PrintDeployClient essentially does is applying a set of changes (to file system, registry, etc.) that it has previously recorded while a printer driver was installed one some example computer at the UIS. If that example computer had a malware infection, that

- could now easily be disseminated via PrintDeployClient to user machines. Likewise, installing security updates for these drivers will require someone at the UIS to repeat this process, with all the things that obviously can go wrong there. PrintDeployClient users can't see what is going on.

PaperCut NG and PrintDeployClient are simply completely the wrong solution for us today, because they were created 26 years ago (1999) to solve problems other than ours:

- PaperCut NG foremost is a page accounting system, introduced to charge undergraduate students a per-page fee, **but we don't charge staff and graduate students per page**.

PrintDeployClient was designed to install vendor-specific printer drivers, **but practically every printer that appeared on the market after about 2016 now enables driverless printing via the IPP 2 protocol, which has built-in support on all common operating systems** (under certification brand names such a AirPrint, Mopria, IPP Everywhere), so the printer-driver installation problem has largely disappeared (or will disappear once we replace our old IPP 1.1 printers, or put an IPP 2 proxy in front of them).

I don't object to replacing our aging printers. In fact, having modern IPP 2.x compliant printer firmware on its own may somewhat simplify setting up printers in the department.

I also don't object to us having a few DS-Print-connected printers or MFDs, e.g. along entry routes into the department, for the benefit of people who want to pick up printouts on the way in, if there really is demand for such a service. (But do we really need a scanner on every corridor? Surely the demand for MFD scanners is at least an order of magnitude lower than for printers? Seems wasteful.)

But I am concerned about users being forced to use a quarter-century-old print-server architecture and product choice that has simply makes no sense in our contemporary context of driver-less printing and not charging for pages.

There are very simple alternatives possible. Some examples:

-

The new printers can serve jobs from DS-Print and jobs from a local print server simultaneously. While DS-Print uses IPP on TCP port 631, the AppSocket TCP port 9100 remains available for independent simultaneous use (e.g. by a local CUPS server). The printers contain their own built-in print servers and can internally queue jobs arriving via different protocols. CUPS and DS-Print are not mutually exclusive.

If there really are good reasons against installing even a basic print server (e.g., CUPS on Ubuntu), why not try making the printers directly available on in-house LANs, without any intervening print server, and advertise them via DNS-SD (aka “Wide-area Bonjour”)? Operating systems would locate the printers directly via DNS, download all the printer specifications and capabilities via IPP, and submit jobs directly. That's what AirPrint and many home-office setups do.

Making printers available via a LAN has become such a simple, routine and basic IT-support task that it is really difficult to see why a world-leading computer science department finds itself in the curious position of having to install restricting proprietary dated software solely to send all its print jobs unnecessarily through another organization. Something clearly has gone wrong here.

Markus Kuhn, 20 June 2025

Appendix: Printer-support improvements

We also identified some documentation and configuration problems with local CUPS printers:

- Some of the web instructions for setting up CUPS connections still predate the release of Windows 10 (2015) or OS X 10.7 Lion (2011) and, if applied in recent years, will likely have caused malfunctions or lead to the installation of decade-old vendor drivers with known bugs and security vulnerabilities. (e.g. RT#131053)

- On Lab-managed Linux machines, an old lpr-wrapper script is routinely installed that, while in principle useful, contains out-of-date information about individual printers and corrupts several command-line options offered by the CUPS version of lpr, e.g. by passing through only the first job option. (I've offered a fixed version in RT#131012)

- We have encountered a case where the wrong vendor's PostScript Printer Definition file was installed on the CUPS server, leading to many job options not working. (e.g. RT#133540)

- Printer firmware has not been kept up to date. (e.g. RT#133540)

Fixing such issues would be easy wins and might help to alleviate problems that may have been incorrectly attributed to CUPS per se. (Some of these tickets were closed unresolved.)