OSG Technology 2018

Brian Bockelman
OSG AHM 2018

2017 in Recap:

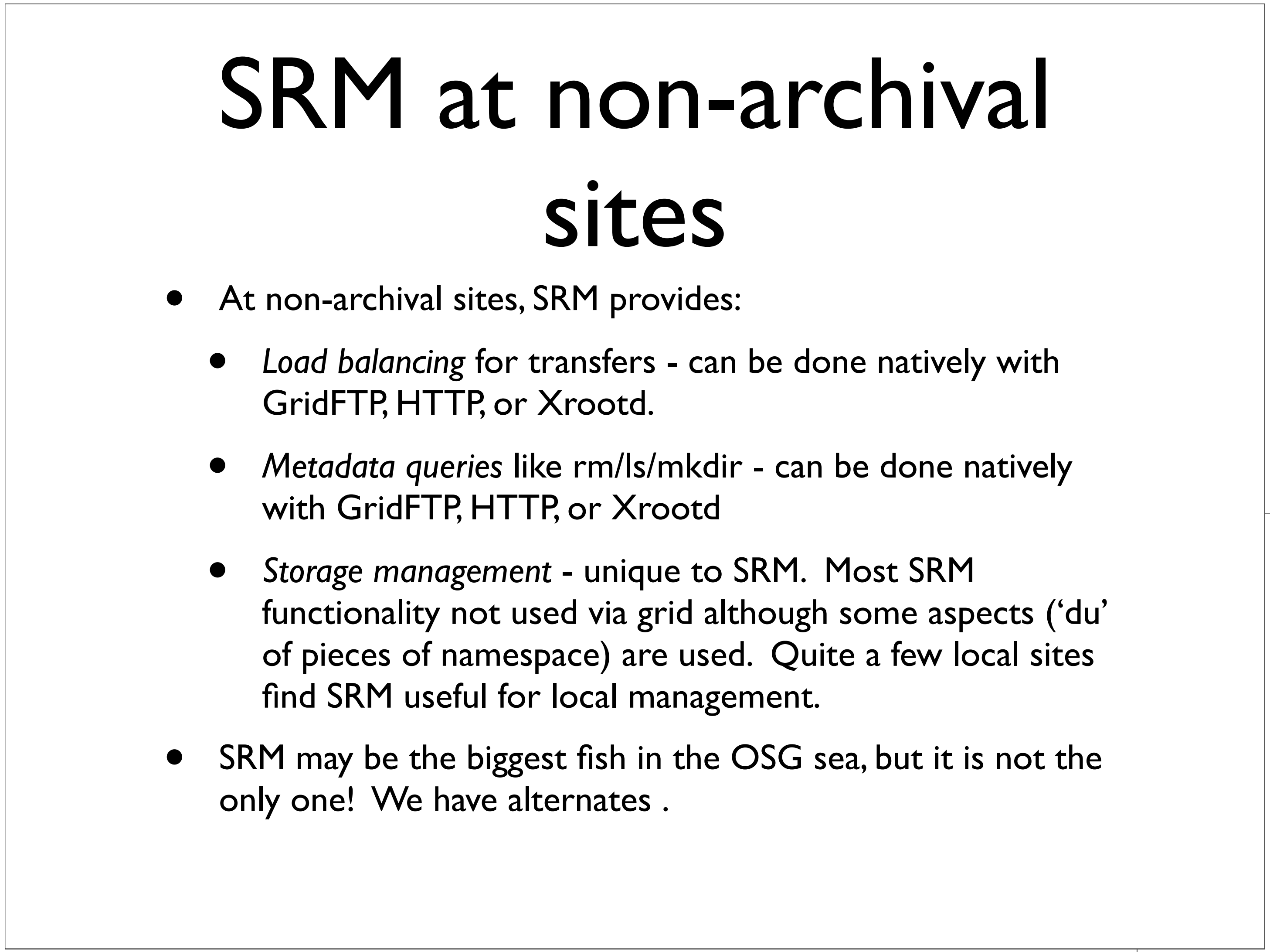
Simplify,
Simplify,
Simplify

2017 was a *fantastic* year for retiring software

In memoriam

- Recall all the friends we've lost in the past year:
  - GRAM, glexec, GIP/BDII, Gratia (central), bestman2, GUMS, lcg-utils, VOMS-Admin.
- These transitions are *important*, require *significant lead-time*, and are *worthwhile*.
  - My first presentation on retiring SRM was in 2012!
  - OSG's support for the bestman2 SRM implementation ends in 75 days.
  - The resulting infrastructure is simpler and reduces maintenance burden.
- Software has **lifetime** beyond its "best used by" date!
  - This final lifecycle stage can entail a good chunk of the support costs. Who pays those?
  - It's been a role of the OSG to help ease these transitions! **We try to plant the seeds many years beforehand**.

Slides from AHM 2013



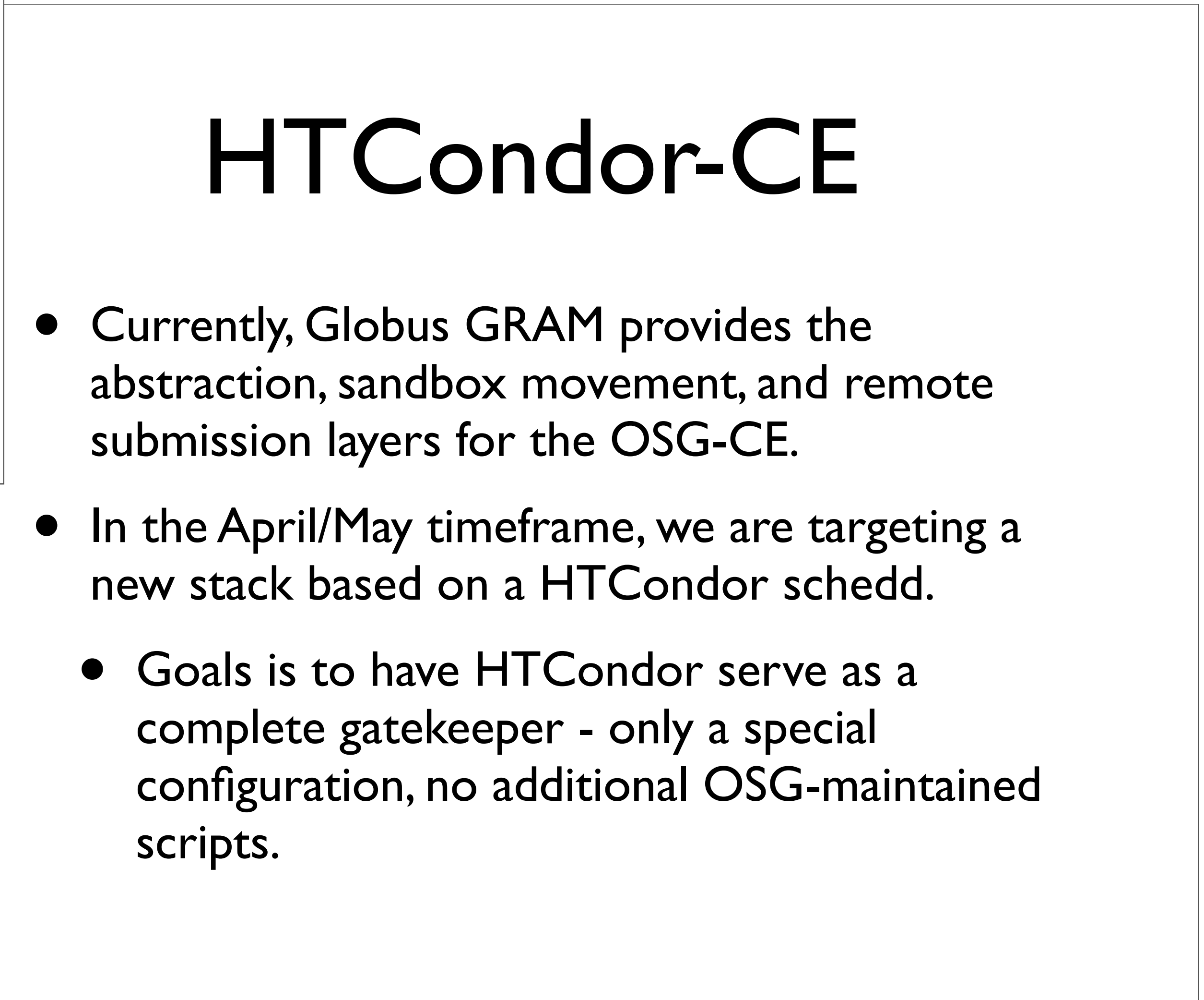
SRM at non-archival sites

- At non-archival sites, SRM provides:
  - *Load balancing* for transfers - can be done natively with GridFTP, HTTP, or Xrootd.
  - *Metadata queries* like rm/ls/mkdir - can be done natively with GridFTP, HTTP, or Xrootd
  - *Storage management* - unique to SRM. Most SRM functionality not used via grid although some aspects ('du' of pieces of namespace) are used. Quite a few local sites find SRM useful for local management.
- SRM may be the biggest fish in the OSG sea, but it is not the only one! We have alternates .

Wednesday, March 13, 13

<- Initial thinking on SRM retirement

Initial announcement on HTCondor-CE ->



HTCondor-CE

- Currently, Globus GRAM provides the abstraction, sandbox movement, and remote submission layers for the OSG-CE.
- In the April/May timeframe, we are targeting a new stack based on a HTCondor schedd.
  - Goals is to have HTCondor serve as a complete gatekeeper - only a special configuration, no additional OSG-maintained scripts.

Globus is going away…

- Last June, Globus announced support for the Globus Toolkit was ending December 2017 (security-only support for another year).
    - Their organization’s services planned to stop using GT components.
    - They didn’t have a mechanism to provide sustainable support for the GT community.
- The GT support community didn’t extend beyond the existing NSF project!

https://opensciencegrid.github.io/technology/policy/globus-toolkit/

https://software.xsede.org/news/xsede-response-globus-toolkit-end-support-announcement

… But the community isn’t!

- There are several organizations that rely on similar functionality out of the Globus Toolkit — CERN, EGI, OSG, PRACE, XSEDE.
- Members of these organizations banded together to create the **Grid Community Forum** in order to maintain a fork of the Globus Toolkit, the **Grid Community Toolkit**.
- This mechanism will provide baseline support for the functionality we need.
  - Given the maturity level of the software, effort level is fairly manageable … until OpenSSL breaks its ABI.
  - This happens every 3-4 years: hence, we have a reasonable amount of time to plan for the future.
- Note that GridCF could potentially include other software stacks under its umbrella in the future.

Looking Ahead

- If simplifying the software stack saves us* time and money, what have we been doing with it?
  - Transitioning to a new bulk transfer model.
  - Fixing our authorization model.
  - Advancing portability of application environments.
  - Tackling the “data management problem” - caches and organized replica management.

*** (OSG, sites, community)**

Looking Ahead (With Buzzwords)

- If simplifying the software stack saves us* time and money, what have we been doing with it?
  - HTTPS! (Transitioning to a new bulk transfer model.)
  - SciTokens! (Fixing our authorization model.)
  - Singularity! (Advancing portability of application environments.)
  - StashCache! Rucio! (Tackling the “data management problem” - caches and organized replica management.)

*** (OSG, sites, community)**

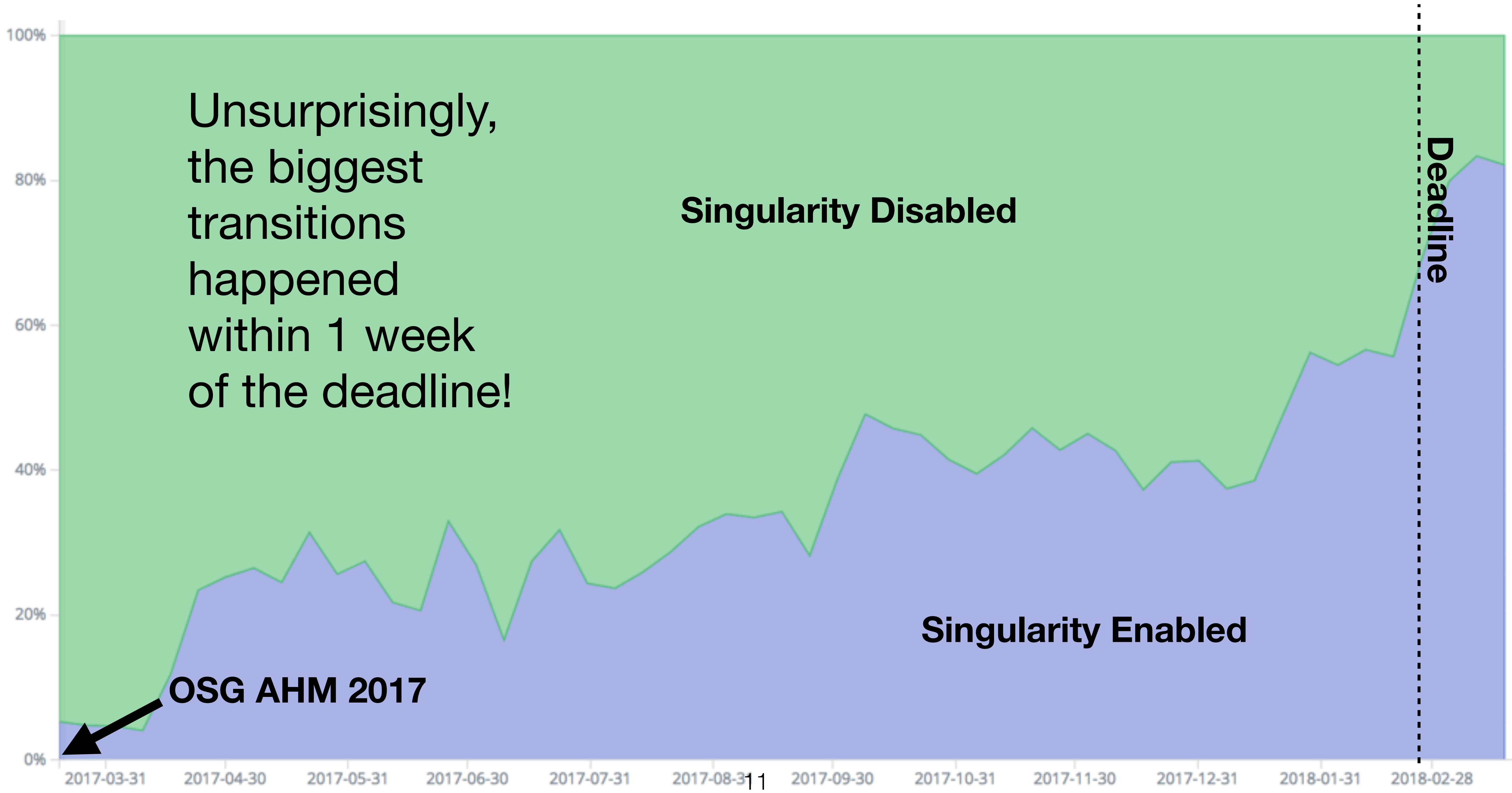
Portable Applications

- **In the beginning**, there was `a.out`: the application was a statically linked executable.
  - Perfectly pleasant to move between execution environments.
- Then the Linux community discovered shared libraries and modules.
  - Had many great properties. Portability is not one of them.
  - An entire generation of developers was trained on development styles that didn't include portability.
- **What is old is new again**: with Linux containers, users can

Containers on OSG

- This isn't your grandpa's `a.out`: the average container size used on OSG is 3.7GB (uncompressed).
    - Building compact containers is still an art.
    - Distribution is a *challenge*. We have a reasonable solution for WLCG-like sites: we have yet to meet the challenge for sites without CVMFS.
- We currently use Singularity as the runtime for our containers. Started contributions to the upstream project in 2016.
- Singularity / containers solves portability issues: opportunities remain to better integrate it in the runtime stack (error handling / translation).

CMS Singularity Rollout: Last 12 Months

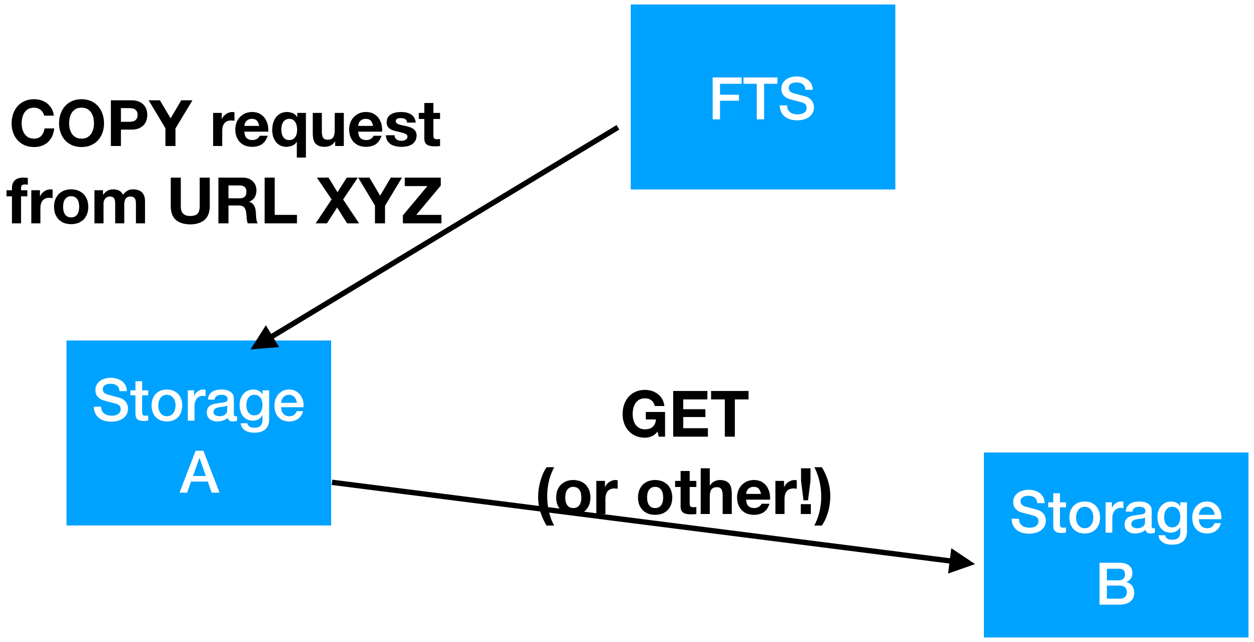


Unsurprisingly, the biggest transitions happened within 1 week of the deadline!

OSG AHM 2017

WebDAV TPC

- WebDAV TPC is done by FTS contacting one storage endpoint, asking it to COPY to/from a given URL.
    - The active endpoint performs the transfer, typically a HTTP GET or POST.
    - Important: **ANY URL** can be given, including GridFTP or XRootD.
    - "Storage B" needs to know *nothing* about WebDAV TPC; only needs GET/PUT semantics. Allows transfers with S3, for example.
- Already widely implemented, including plugin available for XRootD (`xrootd-tpc` in osg-upcoming).
- Tricky part: *authorization* with Storage B. For this, we are working on a concurrent transition away from X509 to bearer-token based.
- This work is just beginning: lots of things to do in areas like performance. **Perfect for external collaboration!**



Authz revolution:

- **Identity-based:** authorization based on mapping who you are.
- **Capability/Token-based**: authorization based on something you are able to present.

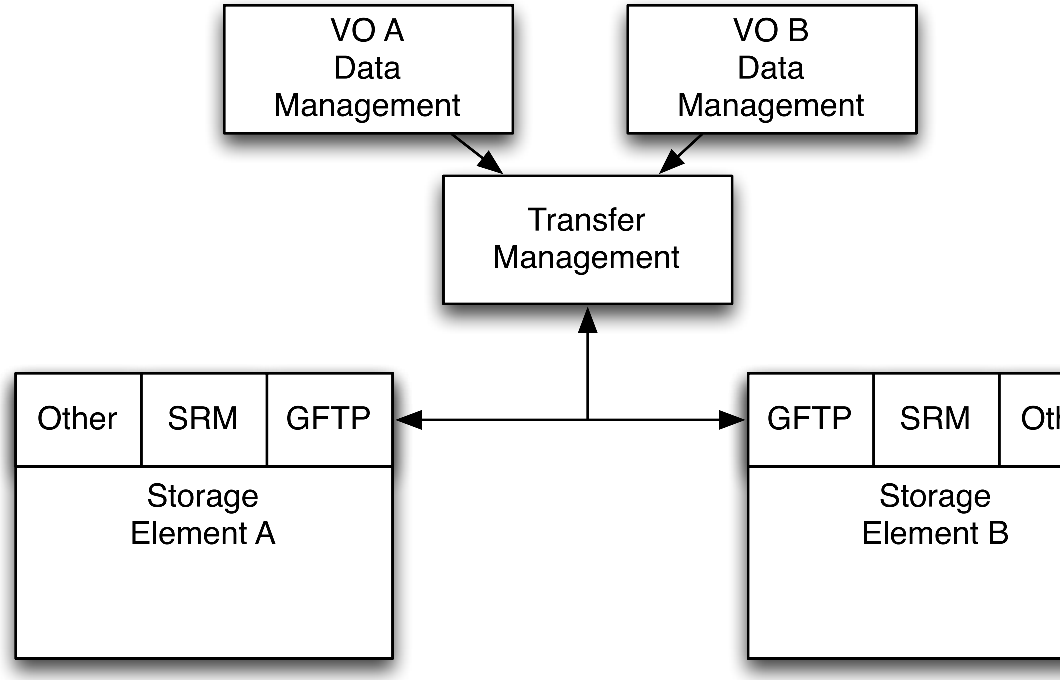
Capability-based authorization

- Currently, sites figure out who you are (identity), then decide what you’re allowed to do.
    - Most sites don’t care at the level - they want to say “CMS can write into /mnt/foo” and let CMS take care of the rest.
- In the ecosystem we are working on with the SciTokens team:
    - Storage software is able to validate the signature is associated with a VO.
    - Capabilities allow CMS to sign authorizations for activities within its storage areas.
- Example token payload:

```
{
“iss":"https://scitokens.org/cms",    # Token issuer
“scp":["write:/store/user/clundst","read:/store"],   # Storage authz
“sub":"clundst",   # Subject name, for traceability.
“jti”:"b8d54a62-cd33-4b4b-bb64-11b804272f1d",  # Token ID.
“exp":1521561382,   # Expiration and validity time.
"iat":1521557782,
"nbf":1521557782
}
```

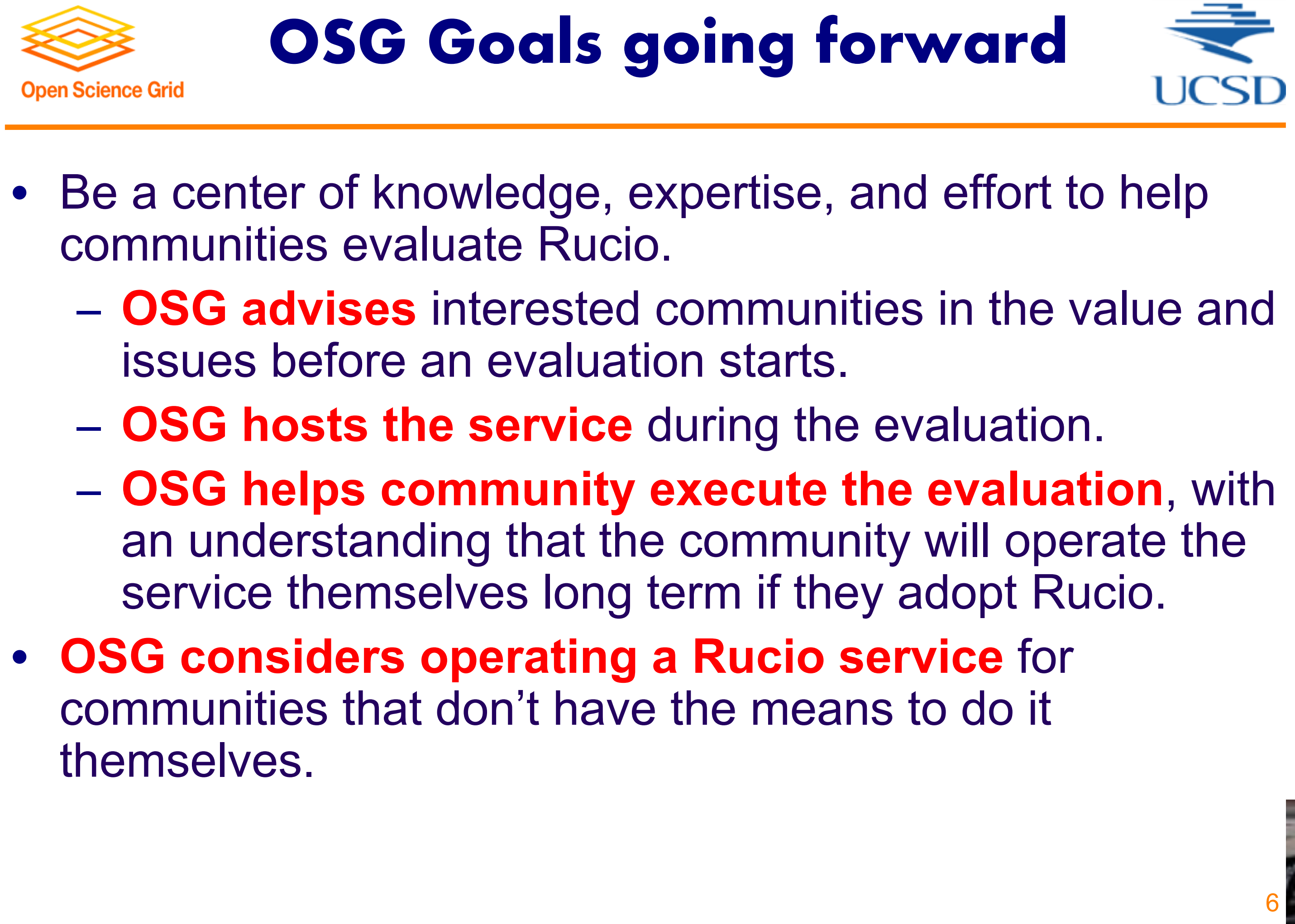
Rucio - Data Replication Management

- I think almost everyone here has seen my rant on how the storage element model has failed opportunistic VOs.
    - In truth, it's not really been successful for small VOs with dedicated storage either!
    - Why? **TOO HARD** and too complex.
- Rucio is a promising piece of software from ATLAS that:
    - Allows the VO to describe its replica policy at a relatively high-level.
    - Well-implemented and leverages transfer layers (FTS) that have begun to mature.
    - Manages the complexity. Includes many functionalities VOs have had to do themselves.
- For technical details, see Benedikt's presentation from Monday: https://docs.google.com/presentation/d/1-U-19bwKHNB0uXmfxPNk0Cakvd-hnebmw6cpSJQSUKg/edit#slide=id.g35932472d5_1_131



Rucio - Growing Community

- Rucio is an ATLAS project, but has been working hard to transform into a community project. First community workshop this month!
- OSG has been working to enable communities that want to evaluate Rucio.
- Lots of potential for joint collaborative projects: both in terms of “scaling down” to make it easier and develop new capabilities (such as SciTokens integration).



Open Science Grid

OSG Goals going forward

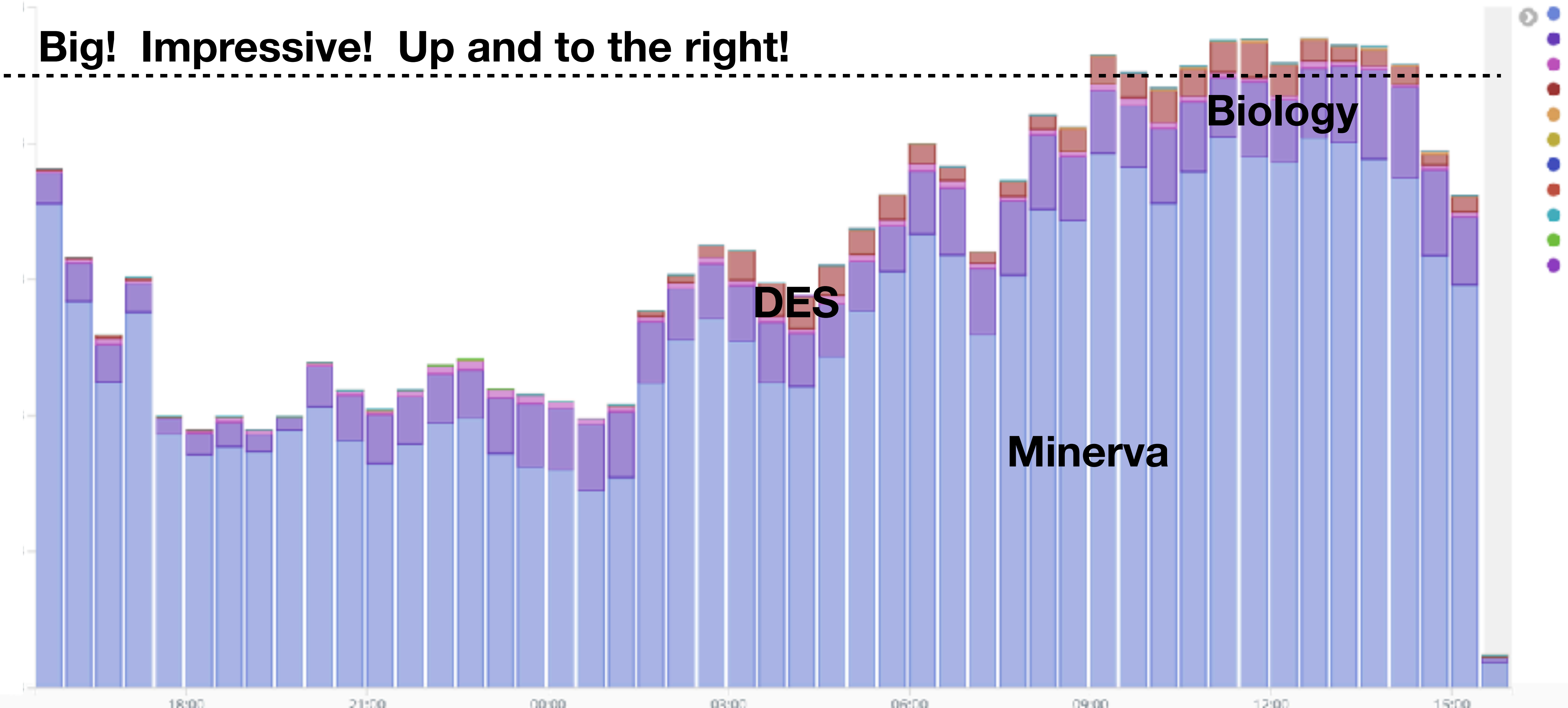
UCSD

- Be a center of knowledge, expertise, and effort to help communities evaluate Rucio.
  - **OSG advises** interested communities in the value and issues before an evaluation starts.
  - **OSG hosts the service** during the evaluation.
  - **OSG helps community execute the evaluation**, with an understanding that the community will operate the service themselves long term if they adopt Rucio.
- **OSG considers operating a Rucio service** for communities that don’t have the means to do it themselves.

6

StashCache

- StashCache is our HTTP- and XRootD-based caching infrastructure
  - Actually spawned from a student project at UChicago in 2014.
- Through 2017, we saw continued adoption of StashCache — both individual users (enabled by user support) and


Big! Impressive! Up and to the right!

DES

Biology

Minerva

StashCache - challenges

- In the past few years, we've been tackling the technical challenges in StashCache:
    - Integrate with documentation and user workflows
    - Add new features (POSIX IO, authenticated StashCache, writable Stash).
    - Stability of the software (tackle those memory leaks!)
    - Monitoring to understand the performance.
- But the strategic challenge remains:
    - The **cache space is a shared resource** which we "manage" through social mechanisms.
    - We need to actually manage the storage and IO: a fundamental problem where we'll need to collaborate with external projects.
    - Currently completely orthogonal from the data replication work with Rucio.

Technology

- Take home messages for the day:
    - Software and Technology team personnel are a core resource us to evolve the OSG technology landscape.
    - We’ve pushed for many years to have a leaner, meaner software stack. This has paid dividends in 2017.
    - With this “simplicity dividend”, we have the effort to tackle challenges such as the support for the Grid Community Toolkit.
        - We have been able to turn the challenges into opportunities for things like authorization models.
    - We’ve also been able to push the boundaries within the OSG in areas like environment portability, data caching, and data replication.