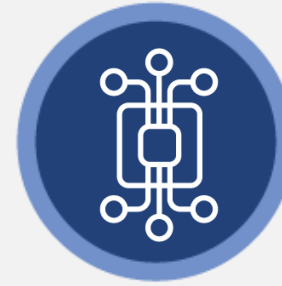


# Globus Compute Panel

MW RCD Annual Meeting  
May 7, 2026



## What's New in Globus Compute

Lev Gorenstein, Solutions Architect  
[lev@globus.org](mailto:lev@globus.org)

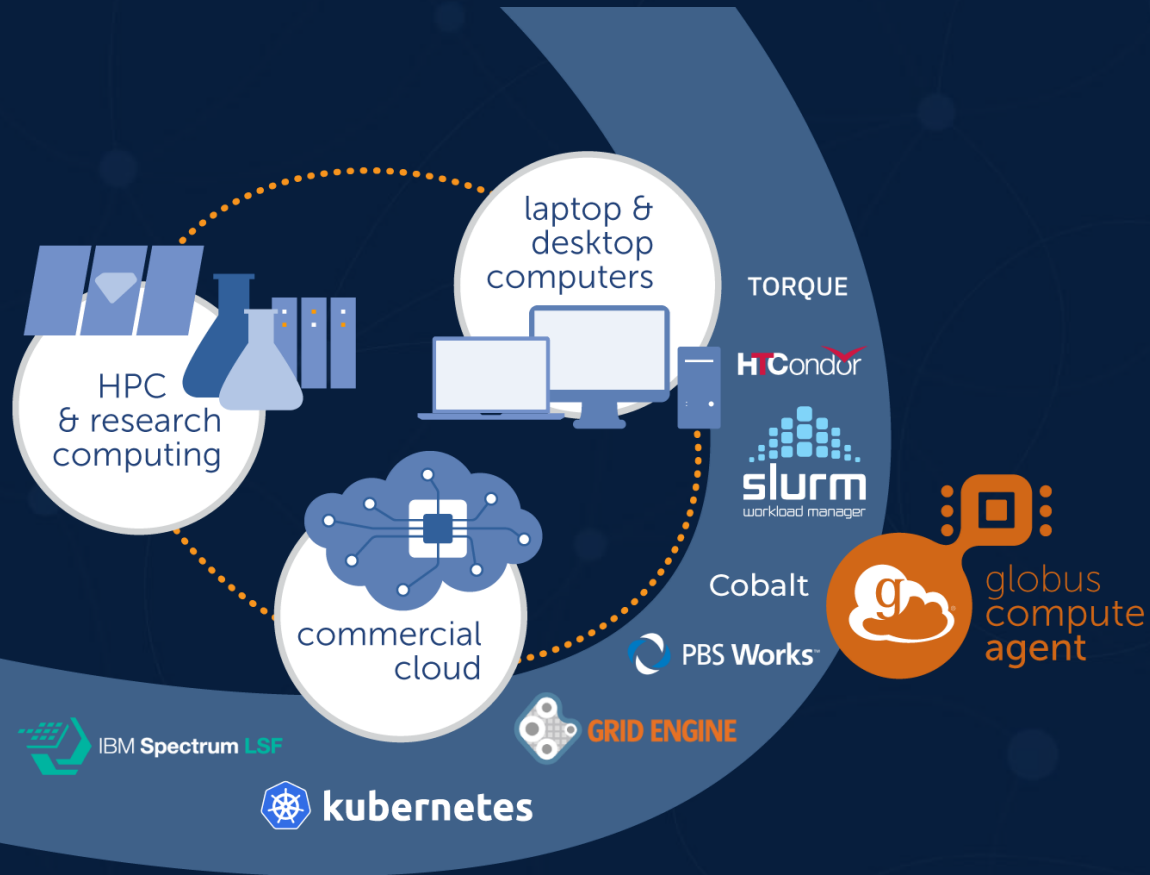


THE UNIVERSITY OF  
**CHICAGO**





# What is Globus Compute?



- A service that enables programmatic access to compute resources using the FaaS model
  - Secure, reliable orchestration of compute tasks
  - Consistent user interface across diverse resources



# Globus Compute components

- **Compute service** — Highly available cloud-hosted service for managed function execution
- **Compute endpoint** — Locally-deployed software that abstracts access to compute resources, from edge device to supercomputer
- **Compute SDK** — Python interface for interacting with the service, using the familiar (to many) Globus look and feel

# Interactive (or not) computing via familiar tools

- Define function like any other code
- Invocation runs code on compute resource
- Wait for results, or access later asynchronously
- Not just Python (support for MPI and basic shell operations)

```
# Define the function for remote execution
def (name: str):
    return f'Hello, {name}!'

from globus_compute_sdk import Executor

MY_COMPUTE_ENDPOINT = "ad4f48be-9c03-49bc-9dc4-e240bc599bef"

# Run the function on the remote compute resource
with Executor(endpoint_id=MY_COMPUTE_ENDPOINT) as ex:
    future = ex.submit(hello_name, 'you')

print(future.result())
```

# Why should researchers use Globus Compute?

- **Interact with HTC/HPC resources via familiar tools**
  - Jupyter notebooks, Python scripts
- **Reliably manage repeated tasks**
  - Parameter sweeps, Monte Carlo simulations
- **Simplify execution of jobs on diverse compute resources**
  - Scale up/out on different platforms with minimal effort
- **Create fully automated research pipelines**
  - Combine robust data management with computation
- **No need to “learn HPC”**
  - Hide complexity of underlying environment



# What's new in Globus Compute?



**Managed remote execution**



# A few stats

## Totals (+ "new this week")

Running Users: 1,840 (+7 [0.38%])  
 Endpoints: 46,102 (+1,518 [3.29%])  
 Functions: 1,353,316 (+313 [0.02%])  
 Tasks: 67,311,314 (+710,058 [1.05%])

## Active this Week

Users: 90  
 Endpoints: 320  
 Functions: 342

## All Users this Week, by Task Count

1:	aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa	691,353
2:	bbbbbbbb-bbbb-bbbb-bbbb-bbbbbbbbbbbb	5,971
3:	cccccccc-cccc-cccc-cccc-cccccccccccc	4,862
4:	dddddddd-dddd-dddd-dddd-dddddddddddd	3,606
5:	eeeeeeee-eeee-eeee-eeee-eeeeeeeeeeee	1,552

Inference  
 @Polaris

<https://arxiv.org/html/2510.13724v1> (2025)

The Globus Compute dataset – André Bauer; Haochen Pan, et al.  
 Future Gener. Comput. Syst. (2024) <https://doi.org/10.1016/j.future.2023.12.007>

# TL;DR: Major news up to current **4.10.2**

- Operational changes - **SEP / MEP / Templatability**
- **Daemonization** changes
- **High-Assurance** (HA) support
- Capabilities for fancy **UEP customization on the fly** (decrease serialization warnings)
  - Many **template variables** for runtime introspection
  - **Environment variables** passed into UEP and workers
- **Source code** serialization capabilities
- Miscellaneous **quality of life** improvements

[globus-compute.readthedocs.io/en/stable/changelog.html](https://globus-compute.readthedocs.io/en/stable/changelog.html)



# SEP / MEP / Templates timeline and confusion

- 1.0.12 (March 2023 - still FuncX)
  - Non-templatable SEPs (“single-trick ponies”).
  - Hidden and unadvertised templatable multi-tenant capability.
- 2.0.0 – 2.20.0 (2023-2024, Globus Compute):
  - Non-templatable SEPs (“single-trick ponies”).
  - Hidden and unadvertised templatable multi-tenant capability.
- **2.20.0 – 3.16.0 (May 2024 – October 2025) – “officially MEP”**
  - The **--multi-user** flag. Official MEPs and template capabilities, and RPM/DEB packages.
  - The Zoo:
    - Templatable MEPs,
    - Non-templatable “true SEPs”,
    - Templatable “pseudo-MEP” SEPs (if configured with **--multi-user**).
  - Behaved differently: MEPs map/fork/drop privs/spawn UEPs, templatable “pseudo-MEP” SEPs did the same, “true SEPs” didn’t.

# Streamlining the model

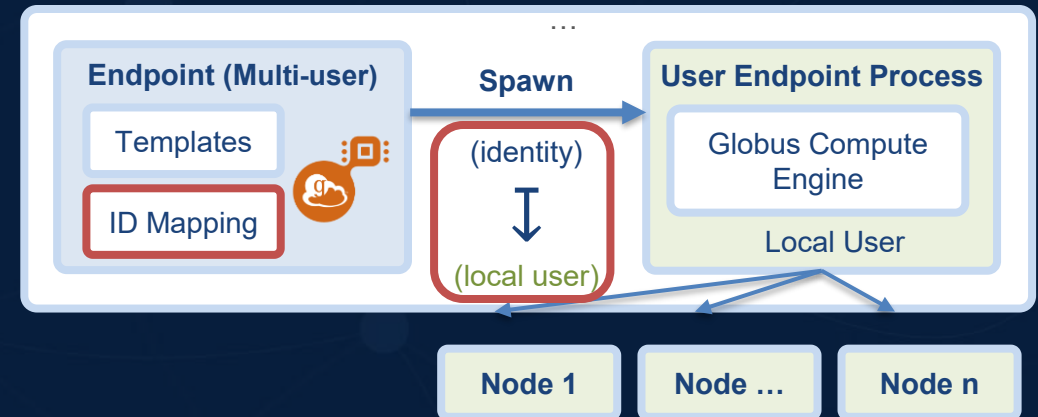
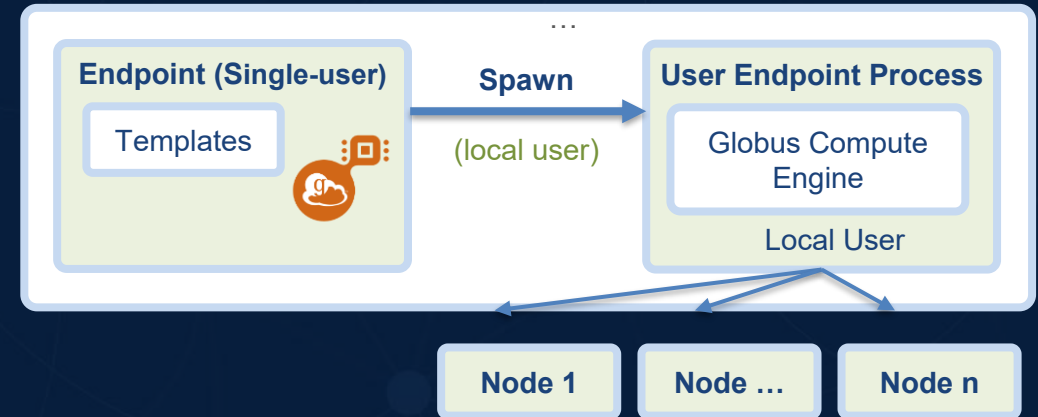
- 4.0.0+ (since October 2025)
  - All endpoints are templatable
  - All endpoints behave identically
    - Map identity → fork → drop privs → render UEP
      - The only difference is that unprivileged (non-root) endpoints skip the mapping and the drop.
  - **TL;DR:** they are all just “endpoints” now
    - No need to differentiate between SEPs and MEPs except in the context of installation and identity mapping.



# Globus Compute Endpoint – two modes



- **Single-user mode**
  - “Personal” endpoint
  - No admin privileges to install
  - Core process runs as user
  - No identity mapping (simply forks)
  
- **Multi-user mode**
  - “System-wide” endpoint
  - Installed by sysadmins
  - Core process runs as root
  - Performs identity mapping (then forks and drops privileges)





# Daemonization changes

- **Pre-4.0.0 (October 2025) – easy:**
  - Systemd for MEPs (**enable-on-boot** and **disable-on-boot** subcommands to generate and manage service files)
  - SEPs auto-daemonized on start (convenient!)
- **4.0.0 – 4.5.0 – slightly harder:**
  - Systemd services for MEPs as before
  - SEPs lost auto-daemonization – needed [proper but non-intuitive] incantations:  
`globus-compute-endpoint start myep &> /dev/null &`  
`disown %1`
- **4.6.0+ (February 2026) – easy again:**
  - Systemd services for MEPs as before
  - Easy '**--detach**' flag for SEPs:  
`globus-compute-endpoint start myep --detach`

# High-Assurance (HA) support

- **3.6.0 - 3.8.0 (April-May 2025)**
  - Ability to declare endpoint as **HA** (and associate with **HA Auth policy**)
  - **HA functions** (once declared, can only run on specific endpoint)
    - HA functions are cleared from Globus-owned infrastructure (e.g., DBs, logs) after 90 days of inactivity
    - Note: **inactivity**, not creation date (so an HA function *could* live in perpetuity if accessed/used every 89 days)
  - AMQP no longer sends results directly to the Executor (only notifies that results are ready, and Executor itself contacts the service to retrieve them)

# More template variables for introspection

- **3.9.0, 3.18.0 (2025)**

- **parent\_config**

- **user\_runtime.\*\*\*:**

- user\_runtime.globus\_sdk\_version
    - user\_runtime.globus\_compute\_sdk\_version
    - user\_runtime.python.\*\*\*:
      - version, version\_tuple, implementation,...
    - user\_runtime.platform.\*\*\*:
      - architecture, machine, node, release,...

- **mapped\_identity.\*\*\*:**

- local.uname, local.uid, local.gid, local.shell,...

- **Full list:**

[globus-compute.readthedocs.io/en/stable/endpoints/templates.html#reserved-variables](https://globus-compute.readthedocs.io/en/stable/endpoints/templates.html#reserved-variables)

```
engine:  
  type: GlobusComputeEngine  
  provider:  
    type: SlurmProvider  
{% if 1001 in mapped_identity.local.groups %}  
  partition: {{ partition }}  
{% else %}  
  partition: default  
{% endif %}
```



# More on-start customization

- **Template variables work at the config/template level**
- **4.9.0 (March 2026)**
  - Env vars **GC\_USER\_PYTHON\_VERSION** and **GC\_USER\_SDK\_VERSION** exported into the UEP (at UEP/worker/task level)
    - `endpoint_init`, `worker_init` and `workers` can use to provision proper Python environment on the fly
    - Fewer serialization issues and Parsl mismatches
- **Globus Compute Tutorial Endpoint dogfoods this**
- **So does Purdue Anvil ( 🙌 Geoffrey 🙌 )**

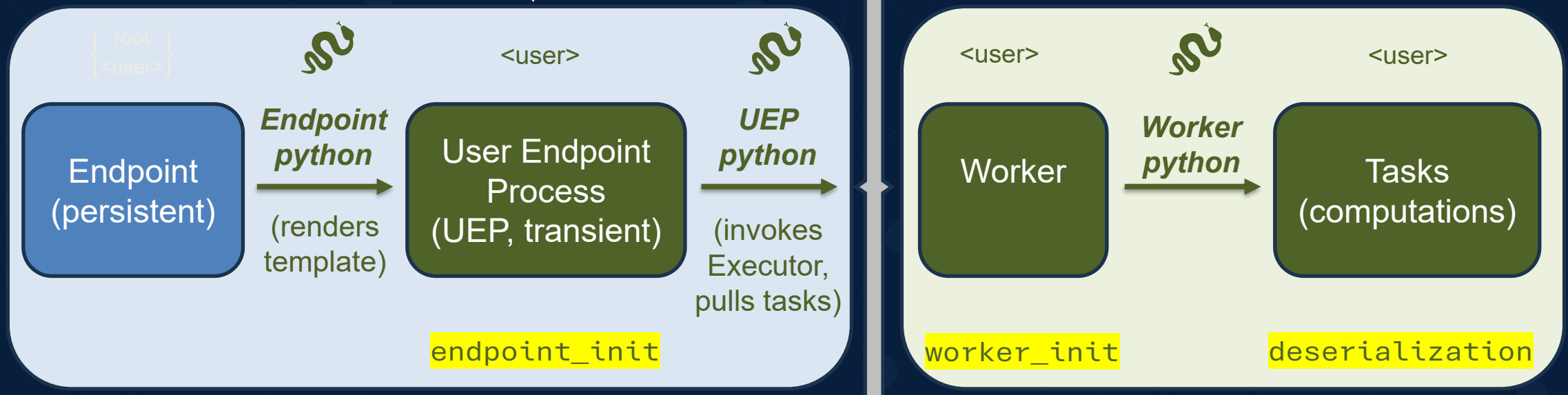
[globus-compute.readthedocs.io/en/stable/endpoints/endpoints.html#advanced-environment-customization](https://globus-compute.readthedocs.io/en/stable/endpoints/endpoints.html#advanced-environment-customization)

# Compute endpoints – the tale of three Pythons

Requests to launch UEP arrive here

Tasks are sent here

Scheduler (or process pool, etc)



job submission host

execution nodes



# Source code serialization capabilities

- **3.16.0 (October 2025)**
  - New methods to register arbitrary Python source code
    - Not sending bytecode ⇒ less sensitive to serialization issues and python version mismatches
    - Don't have to do in-function imports
    - But watch out for dependencies on the worker side instead!

```
Client.register_source_code()  
Executor.register_source_code()
```

[jupyter.demo.globus.org/hub/user-redirect/lab/tree/globus-jupyter-notebooks/Compute\\_Run\\_Task.ipynb](https://jupyter.demo.globus.org/hub/user-redirect/lab/tree/globus-jupyter-notebooks/Compute_Run_Task.ipynb)

# Miscellaneous quality of life improvements

- **3.3.0 (March 2025)**
  - **extends**, **include**, and **import** Jinja tags. Hide sensitive parts from webview (only top-level visible).
- **4.3.0 (January 2026)**
  - Dropped support for Python 3.9 (EOL'd)
- **4.8.0 (March 2026)**
  - **gce** as an alias to the **globus-compute-endpoint** command
- **4.10.1 (April 2026)**
  - Newly-configured MEPs visibility defaults to **public=true**. SEPs visibility is still private (mimicking GCS and GCP).

```
# Hide sensitive gid info and logic.
#
# NB: Renders as user, so put in readable dir
# and specify proper user_config_template_path
# in config.yaml

engine:
  type: GlobusComputeEngine
  provider:
    type: SlurmProvider
{% include "partitions_for_groups.yaml" %}
```

# Benefits of multi-user deployments for clusters

## For Researcher

- No need to log in to the terminal
- No long-running user processes on login nodes

## For Administrators

- Lower barrier for users
- Controllable UEP configurations
  - E.g., pre-choose SlurmProvider, PBSProvider; enforce limits
- Standard Globus Identity Mapping
- Advanced authentication and authorization policies
- Did I say no long-running user processes on login nodes?

# Resources

- **Getting started:**

[jupyter.demo.globus.org/hub/user-redirect/lab/tree/globus-jupyter-notebooks/Compute\\_Introduction.ipynb](https://jupyter.demo.globus.org/hub/user-redirect/lab/tree/globus-jupyter-notebooks/Compute_Introduction.ipynb)

[jupyter.demo.globus.org/hub/user-redirect/lab/tree/globus-jupyter-notebooks/Compute\\_Run\\_Task.ipynb](https://jupyter.demo.globus.org/hub/user-redirect/lab/tree/globus-jupyter-notebooks/Compute_Run_Task.ipynb)

[globus-compute.readthedocs.io/en/stable/quickstart.html](https://globus-compute.readthedocs.io/en/stable/quickstart.html)

- **Endpoint configuration and sample templates:**

[globus-compute.readthedocs.io/en/stable/endpoints/multi\\_user.html](https://globus-compute.readthedocs.io/en/stable/endpoints/multi_user.html)

[globus-compute.readthedocs.io/en/stable/endpoints/endpoint\\_examples.html](https://globus-compute.readthedocs.io/en/stable/endpoints/endpoint_examples.html)



# Ask for help! Really, please.

- **Guidance on best practices**
- **Sounding board for your design/implementation**
- **Assistance with configuring endpoint, templates, etc.**
- **All at no cost to you ...just reach out**
  
- **Contacts:**
  - Me: [lev@globus.org](mailto:lev@globus.org)
  - Support: [support@globus.org](mailto:support@globus.org)
  
- **Mailing list: [discuss@globus.org](mailto:discuss@globus.org)**  
**<https://www.globus.org/mailing-lists>**