

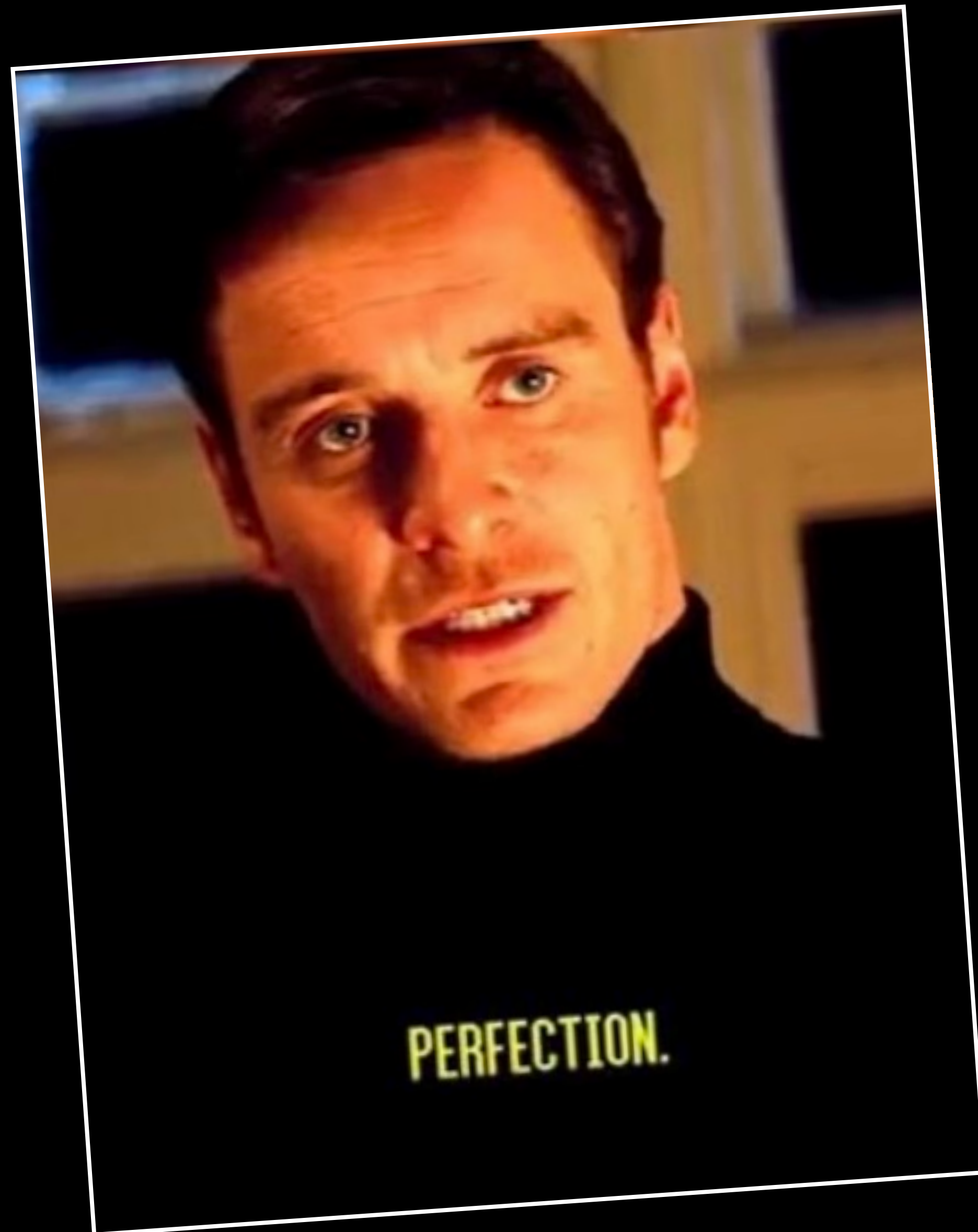
APEX / GLOBUS · SEPT 14, 2026

Anvil MEP • A Testbed for Innovation

A Million and One Ways to Build Python Environments

Geoffrey Lentner

Principal AI Scientist



React dashboard

single-page UI

FastAPI · REST

API routing

OAuth2

auth gateway

Airflow

workflow scheduler

Python / DAG

workflow definition

PostgreSQL

persistence

Redis + Celery

broker + comms

Kubernetes

orchestration

Slurm

resource provisioning

Apptainer

containers

... ALL TO RUN

```
for sample in samples: analyze(sample)
```

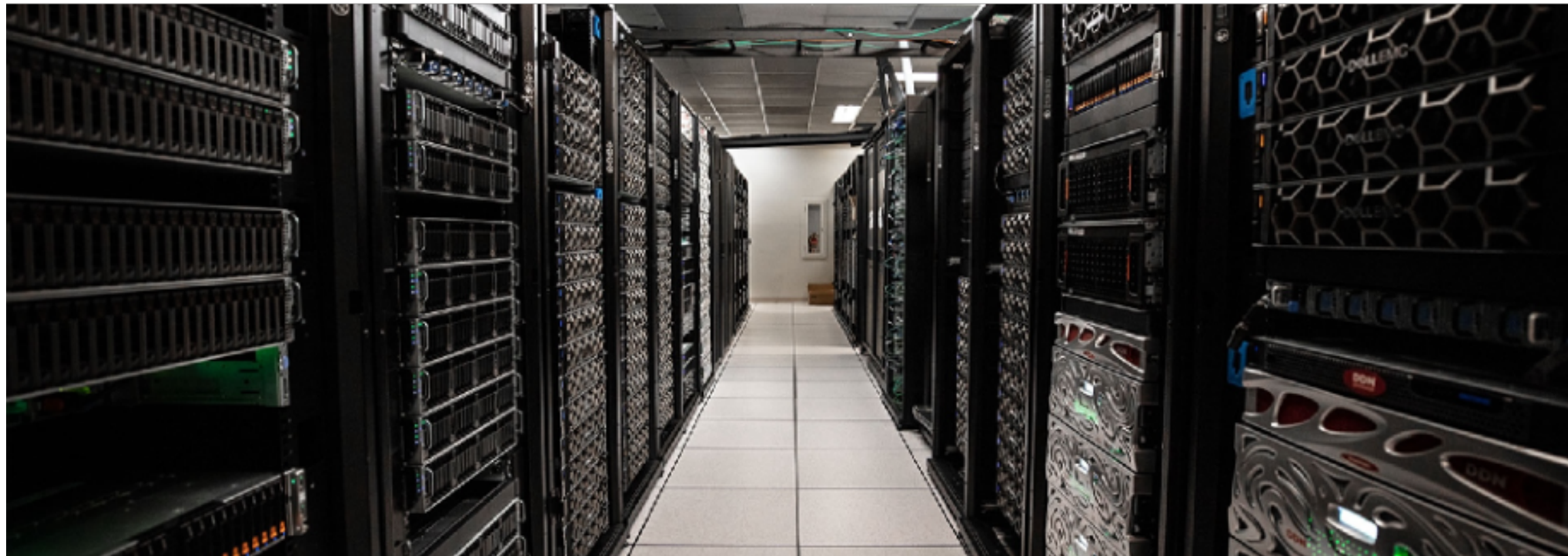


TLDR;

- **Anvil MEP** - not new but still notable,
- Wish list from last year now live,
- UV shims to tame Pythons,
- We can do better - with **uv-manager**,
- **Globus-MCP** for compute,
- Bonus material - maybe

Rosen Center for Advanced Computing

Research computing and data services at Purdue University



WHERE I SIT

Systems

clusters · storage · network

Science

domains · methods · data

Software

RSE · workflows · agents



Inside RCAC: data center aisle · open rack · Fortress archive mural.

rcac.purdue.edu



About Me

Background and Roles



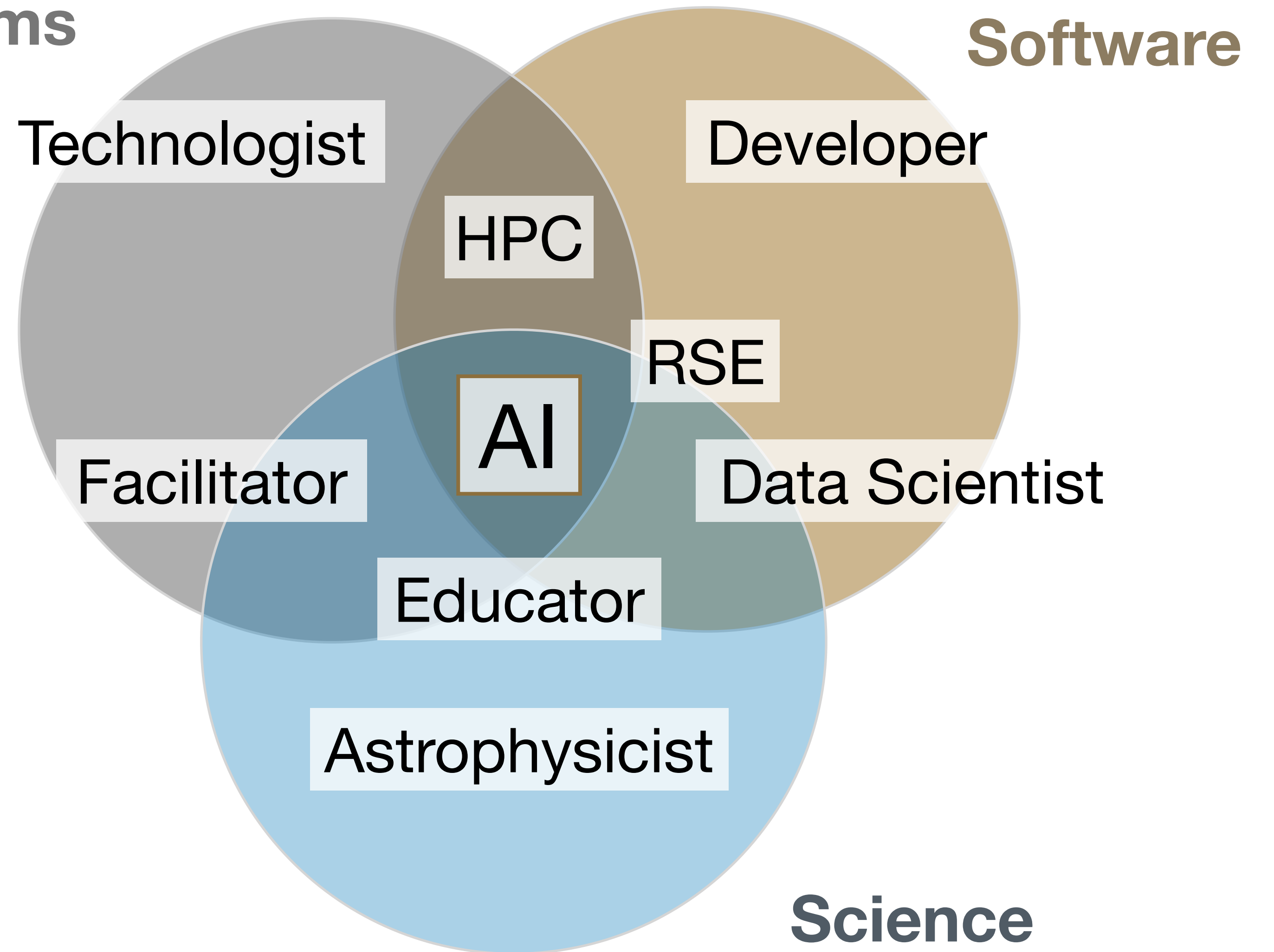
Geoffrey Lentner

Principal AI Scientist · RCAC

glentner@purdue.edu

rcac.purdue.edu/about/staff/glentner

Systems

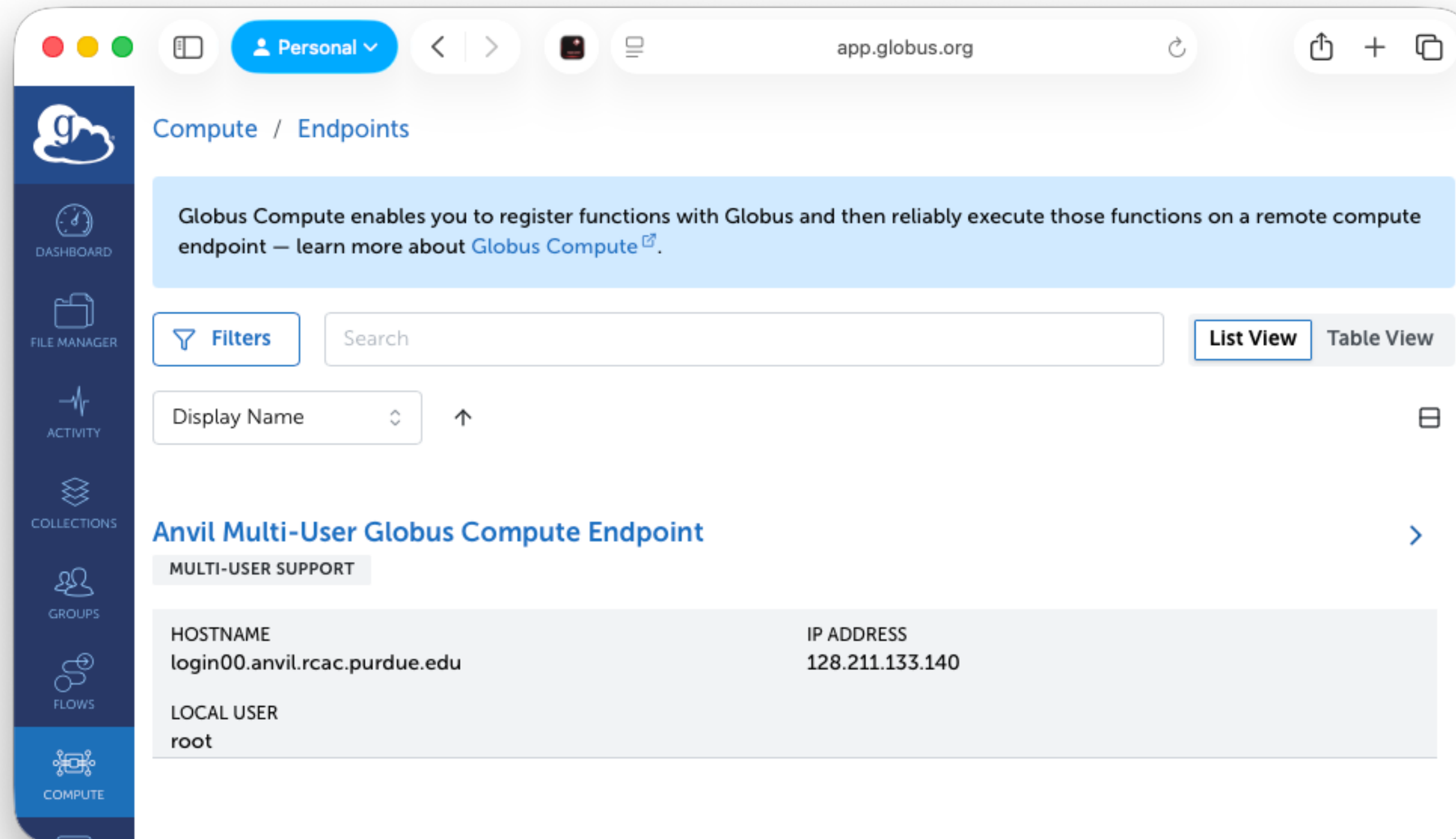


Anvil MEP

One year in review

Anvil MEP

A *Public* Globus Compute Endpoint — with special powers!



- Live for ~18 months
- Updated ~~months~~ *days* ago
- Magic default ***worker_init***
- Magic UEP shim
- More to come!

Anvil MEP

A Public Globus Compute Endpoint with special powers

- We actually did the thing!
- Python and SDK version
- Hash “requirements”
- Sync *venv* with *uv*



Wouldn't it be cool if...

- ▶ Specify dependencies from the client side
- ▶ Magical default *worker_init* materializes at start

Endpoint configuration magic

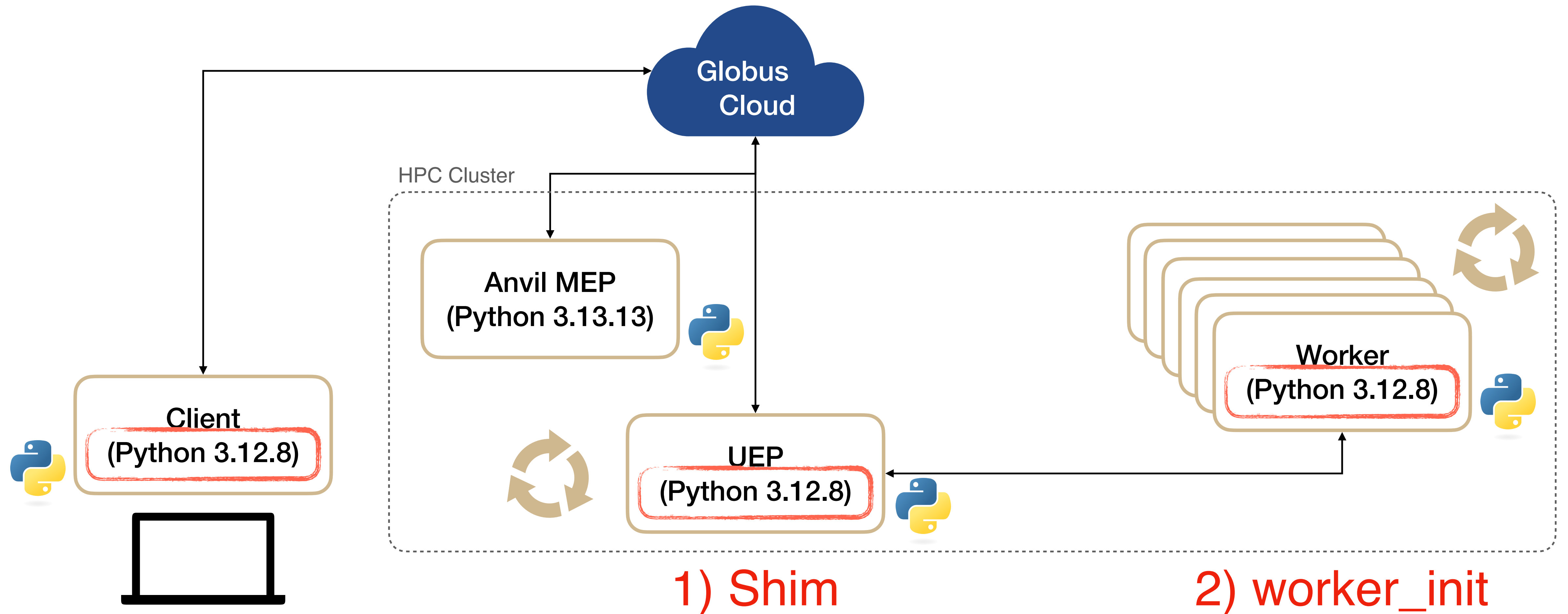
```
config = {  
    'account': 'asc170016',  
    'partition': 'shared',  
    'walltime': '00:05:00',  
    'python_version': '3.13',  
    'python_dependencies': """  
torch ≥ 2.8.0  
matplotlib ≥ 3.10.0  
"""  
}
```

Globus Compute + ParslFest'25
August 29, 2025

17

Anvil MEP

A Public Globus Compute Endpoint with special powers



Anvil MEP

A Public Globus Compute Endpoint with special powers

```
# Anvil user configuration template
# Standard Slurm cluster with dynamic worker_init to provision venvs
#
# Additional properties:
# - requirements: multi-line string (literally requirements.txt).
# - venvs_dir: path to venvs directory (default: ~/.globus_compute/venvs).

{% set major, minor, micro = user_runtime.python.version_tuple %}

{% set default_worker_init -%}
export PATH="$HOME/.local/bin:$PATH"
if ! command -v uv &> /dev/null; then
    echo "uv command not found; installing..."
    curl -LsSf https://astral.sh/uv/install.sh | sh
fi

ENV_SITE={{ venvs_dir | default("$HOME/.globus_compute/venvs") }}
ENV_HASH=$(echo {{ requirements | default() }} | md5sum | awk '{print $1}')
ENV_NAME="py{{ major }}.{{ minor }}.{{ micro }}-${ENV_HASH}"
ENV_PATH="$ENV_SITE/$ENV_NAME"

uv venv "$ENV_PATH" --python {{ major }}.{{ minor }}.{{ micro }}
source "$ENV_PATH/bin/activate"

export UV_LINK_MODE=hardlink
export UV_COMPILE_BYTECODE=1
export UV_TORCH_BACKEND=auto
uv pip install globus-compute-endpoint={{ user_runtime.globus_compute_sdk_version }}
uv pip install -r <(echo -e {{ requirements | default() }})
{%- endset %}
```

- Jinja template includes dynamic behavior to automatically construction environment for functions.
- Users can of course override this.

Anvil MEP

A Public Globus Compute Endpoint with special powers

```
# Anvil user configuration template
# Standard Slurm cluster with dynamic worker_init to provision venvs
#
# Additional properties:
# - requirements: multi-line string (literally requirements.txt).
# - venvs_dir: path to venvs directory (default: ~/.globus_compute/venvs).
```

```
{% set major, minor, micro = user_runtime.python.version_tuple %}
```

```
{% set default_worker_init -%}
export PATH="$HOME/.local/bin:$PATH"
if ! command -v uv &> /dev/null; then
    echo "uv command not found; installing..."
    curl -Lsf https://astral.sh/uv/install.sh | sh
fi
```

```
ENV_SITE={{ venvs_dir | default("$HOME/.globus_compute/venvs") }}
ENV_HASH=$(echo {{ requirements | default() }} | md5sum | awk '{print $1}')
ENV_NAME="py{{ major }}.{{ minor }}.{{ micro }}-$ENV_HASH"
ENV_PATH="$ENV_SITE/$ENV_NAME"
```

```
uv venv "$ENV_PATH" --python {{ major }}.{{ minor }}.{{ micro }}
source "$ENV_PATH/bin/activate"
```

```
export UV_LINK_MODE=hardlink
export UV_COMPILE_BYTECODE=1
export UV_TORCH_BACKEND=auto
uv pip install globus-compute-endpoint={{ user_runtime.globus_compute_sdk_version }}
uv pip install -r <(echo -e {{ requirements | default() }})
{% - endset %}
```

Metadata from client

Install uv

Setup virtual env

Anvil MEP

A Public Globus Compute Endpoint with special powers

```
# Anvil user configuration template
# Standard Slurm cluster with dynamic worker_init to provision venvs
#
# Additional properties:
# - requirements: multi-line string (literally requirements.txt).
# - venvs_dir: path to venvs directory (default: ~/.globus_compute/venvs).

{% set major, minor, micro = user_runtime.python.version_tuple %}

{% set default_worker_init -%}
export PATH="$HOME/.local/bin:$PATH"
if ! command -v uv &> /dev/null; then
    echo "uv command not found; installing..."
    curl -LsSf https://astral.sh/uv/install.sh | sh
fi

ENV_SITE={{ venvs_dir | default("$HOME/.globus_compute/venvs") }}
ENV_HASH=$(echo {{ requirements | default() }} | md5sum | awk '{print $1}')
ENV_NAME="py{{ major }}.{{ minor }}.{{ micro }}-${ENV_HASH}"
ENV_PATH="$ENV_SITE/$ENV_NAME"

uv venv "$ENV_PATH" --python {{ major }}.{{ minor }}.{{ micro }}
source "$ENV_PATH/bin/activate"

export UV_LINK_MODE=hardlink
export UV_COMPILE_BYTECODE=1
export UV_TORCH_BACKEND=auto
uv pip install globus-compute-endpoint={{ user_runtime.globus_compute_sdk_version }}
uv pip install -r <(echo -e {{ requirements | default() }})
{%- endset %}
```

Higher-level parameters

- requirements

List of packages/versions

literally (requirements.txt)

So what now?

Anvil MEP

A Public Globus Compute Endpoint with special powers

- We're installing to `~/.local/bin` (leaking the abstraction)
- We're not guarding against user configuration
- We're filling up `/home` unbounded
- We don't deal with multi-arch (x86 login / aarch64 compute)

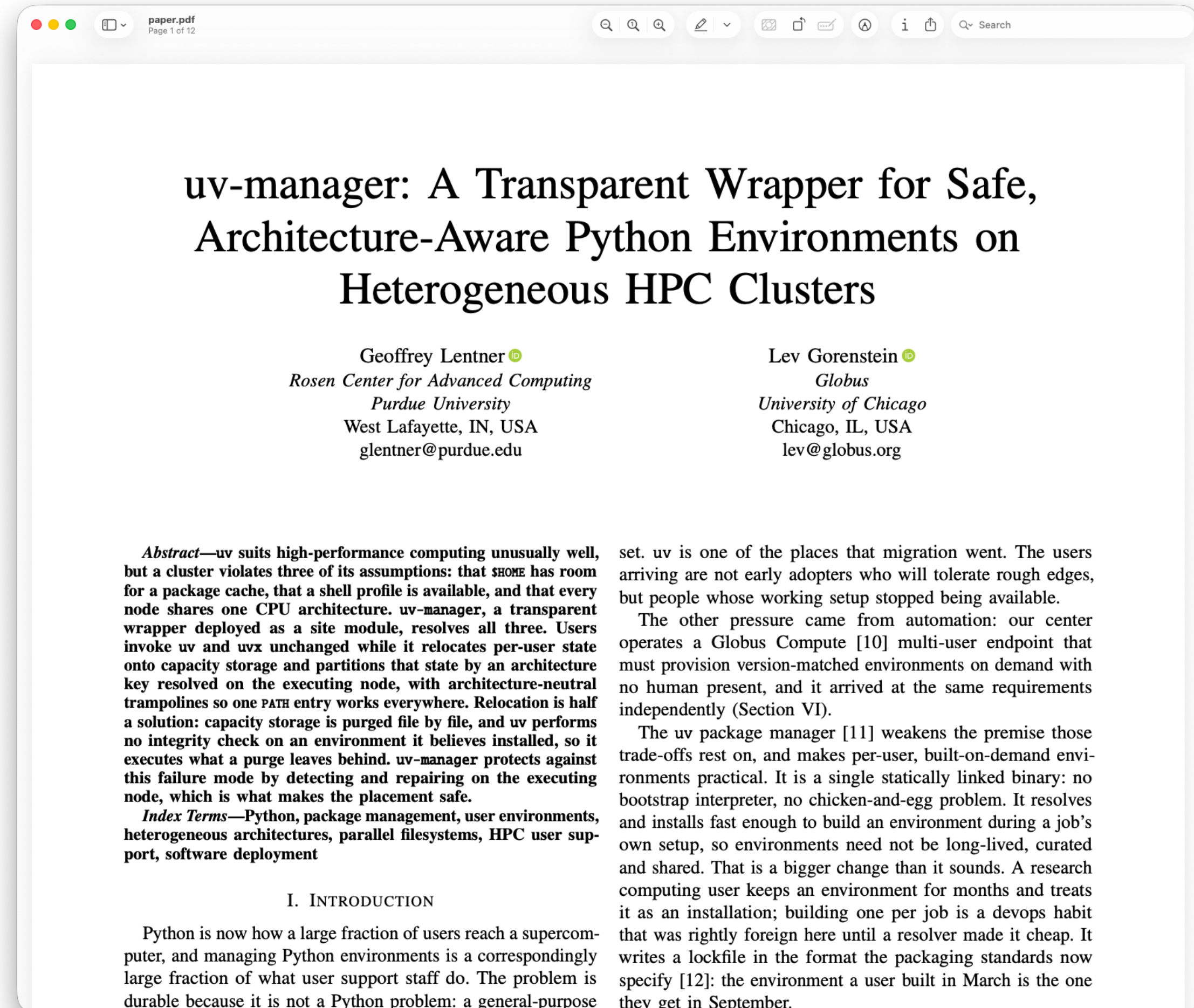
uv-manager

A new tool to manage the abstraction

uv-manager

A new tool to manage uv-based software environments in HPC

- HUST'26 at SC
- Beta - but moving fast



uv-manager

A new tool to manage uv-based software environments in HPC

```
$ module load uv

$ uv --version
uv 0.12.5 (x86_64-unknown-linux-gnu)

$ uvm status
uv-manager:          0.6.1
invoked as:          uvm (/apps/anvil/external/apps/uv/main/bin/uvm)
architecture:        x86_64
state root:           /anvil/scratch/x-lentner/.uv (from $CLUSTER_SCRATCH)
arch root:            /anvil/scratch/x-lentner/.uv/x86_64
selected version:     versions/0.12.5
real uv:              /anvil/scratch/x-lentner/.uv/x86_64/current/uv [present]
real uvx:             /anvil/scratch/x-lentner/.uv/x86_64/current/uvx [present]
resolved version:     uv 0.12.5 (x86_64-unknown-linux-gnu)
installed versions:   0.12.5
UV_CACHE_DIR:         /anvil/scratch/x-lentner/.uv/x86_64/cache
UV_TOOL_DIR:          /anvil/scratch/x-lentner/.uv/x86_64/tools
UV_TOOL_BIN_DIR:      /anvil/scratch/x-lentner/.uv/x86_64/bin/shims
UV_PYTHON_INSTALL_DIR: /anvil/scratch/x-lentner/.uv/x86_64/python
UV_PYTHON_BIN_DIR:    /anvil/scratch/x-lentner/.uv/x86_64/bin/python-shims
trampoline dir:       /anvil/scratch/x-lentner/.uv/bin
XDG_CONFIG_HOME:      <unset – user default: ~/.config>
pin:                  <none – tracks latest>
```

- < 1k lines of Bash
- Not a replacement to uv
- Tries to use **\$SCRATCH**
- Delegates on **basename \$0**

github.com/purduercac/uv-manager

uv-manager

A new tool to manage uv-based software environments in HPC

- Resolve the state root
- Appends arch (`uname -m`)
- Provisions `uv` on first use for that arch
- Exports the `UV_*` storage variables and execs the real binary
- Intercepts `uv self update`
- Maintains arch-neutral trampolines
- *Repairs damaged environments! (WIP)*

```
$UVM_ROOT/  
├── bin/  
│   └── <arch>/  
│       ├── versions/<ver>/  
│       │   └── current → versions/<ver>  
│       ├── bin/shims/  
│       ├── bin/python-shims/  
│       ├── cache/  
│       ├── tools/  
│       └── python/  
└── architecture-neutral trampolines ← safe on PATH  
    uv + uvx, one directory per version  
    tool executables (UV_TOOL_BIN_DIR)  
    managed python links (UV_PYTHON_BIN_DIR)  
    download + wheel cache (UV_CACHE_DIR)  
    tool virtual environments (UV_TOOL_DIR)  
    managed interpreters (UV_PYTHON_INSTALL_DIR)
```

uv-manager

A new tool to manage uv-based software environments in HPC

Launch N
Instances

```
#!/bin/bash  
#SBATCH -N8
```

Application

```
srun uvx thing ...
```

Rank 0 gets the lock and bootstraps/repairs
Ranks 1-N wait, then proceed cheaply

uv-manager

A new tool to manage uv-based software environments in HPC

```
# Anvil user configuration template
# Standard Slurm cluster with dynamic worker_init to provision venvs
#
# Additional properties:
# - requirements: multi-line string (literally requirements.txt).

{% set major, minor, micro = user_runtime.python.version_tuple %}

{% set default_worker_init -%}

export UVM_ROOT=$SCRATCH/.uv
export PATH=$UVM_ROOT/bin:$PATH
curl -LsSf https://github.com/purduercac/uv-manager/blob/main/install.sh | sh

ENV_SITE={{ venvs_dir | default("$UVM_ROOT/venvs") }}
ENV_HASH=$(echo {{ requirements | default() }} | md5sum | awk '{print $1}')
ENV_NAME="py{{ major }}.{{ minor }}.{{ micro }}-$ENV_HASH"
ENV_PATH="$ENV_SITE/$ENV_NAME"

uv venv "$ENV_PATH" --python {{ major }}.{{ minor }}.{{ micro }}
source "$ENV_PATH/bin/activate"

uv pip install globus-compute-endpoint={{ user_runtime.globus_compute_sdk_version }}
uv pip install -r <(echo -e {{ requirements | default() }})

{%- endset %}
```

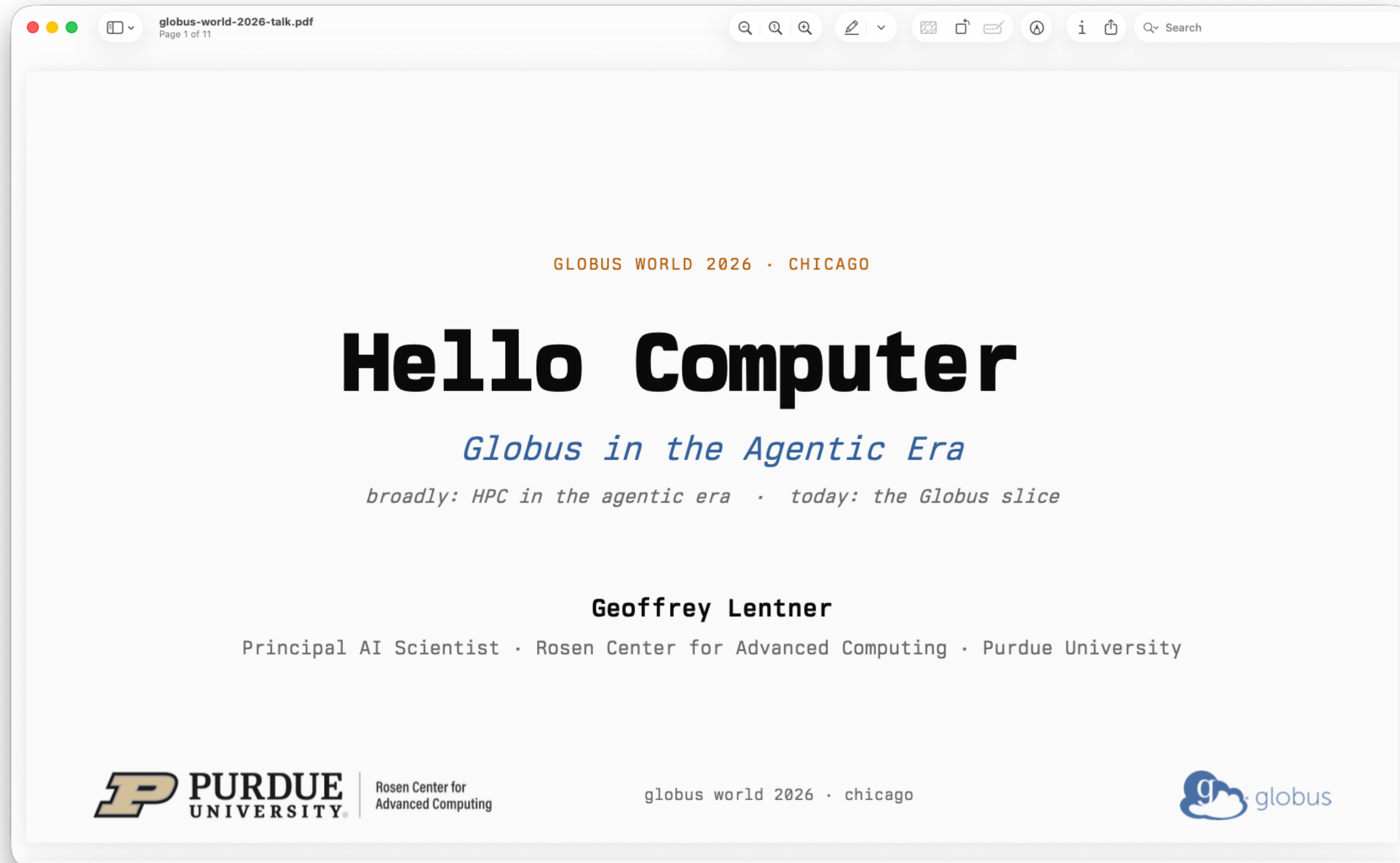
Install wrapper instead

globus-mcp

Agentic workflows with Globus [Compute]

globus-mcp

Globus in the Agentic Era



globus-mcp

Globus in the Agentic Era

Identity & Auth

- `whoami()` — Show logged-in identity
- `globus_login()` — Initiate Globus CLI OAuth login
- `session_consent(scopes)` — Grant endpoint-specific consent

Endpoints

- `endpoint_search(query)` — Find Transfer endpoints by name
- `endpoint_show(endpoint_id)` — Get endpoint details
- `endpoint_local_id()` — Get local GCP endpoint UUID

Filesystem

- `ls(endpoint_id, path)` — List directory contents
- `stat(endpoint_id, path)` — Get file/directory metadata
- `mkdir(endpoint_id, path)` — Create directory
- `rename(endpoint_id, source_path, dest_path)` — Rename or move
- `rm(endpoint_id, path)` — Delete (synchronous)
- `delete(endpoint_id, path)` — Delete (async task)

Transfers

- `transfer(...)` — Submit async transfer task
- `transfer_batch(...)` — Batch transfer (multiple file pairs)
- `task_list()` — List recent tasks
- `task_show(task_id)` — Get task details
- `task_wait(task_id)` — Wait for task completion
- `task_cancel(task_id)` — Cancel a running task
- `task_event_list(task_id)` — Get task events

globus-mcp

Globus in the Agentic Era

Same magic “worker_init”
embedded inside



```
# Submit a Python function to run on a remote HPC endpoint
compute_submit(
    endpoint_id="<uuid>",
    function_source="def analyze(n):\n    ... ",
    function_name="analyze",
    requirements="numpy",
    args=[10000],
)

# Check results later
compute_result("<task-uuid>", timeout=300)
```

Compute (Remote Code Execution)

- `compute_login()` — Authenticate with Globus Compute
- `compute_endpoint_list()` — List accessible Compute endpoints
- `compute_endpoint_status(endpoint_id)` — Check endpoint availability
- `compute_submit(...)` — Submit a Python function for remote execution
- `compute_batch_submit(...)` — Submit multiple functions as a batch
- `compute_status(task_ids)` — Check task status (non-blocking)
- `compute_result(task_id)` — Get task result (optionally wait)

Questions

- Why not use Anaconda?
- Why not just do `<x>`?
- Why are task wait times so long on Anvil?