Hall Pass: Per-action Authorization for Shared Agents

Alexander Zhang · Alok Kamatar (advisor), Kyle Chard (advisor)
University of Chicago

THE UNIVERSITY OF CHICAGO
DEPARTMENT OF COMPUTER SCIENCE

SC26 Chicago, hpc unites.
globus labs

Problem

- Science is collaborative, so agents get shared.
- Some agents are costly: a SAM 3 agent on an A100 takes 52 s to start, then 1.2 s to segment 20 images.
- Academy [3] can share an agent with a Globus group, but then every member can call every action and terminate it.
- So owners choose between a private copy per user, each paying the startup again, and sharing all of it.

Background

- **Academy** is middleware for running agents across research infrastructure [3]. Each agent has a mailbox on an exchange. Callers invoke the agent's actions by messaging it, and the exchange routes the messages.
- Its exchange validates each caller's **Globus Auth** token [5] and looks up the caller's **Globus Groups** [2] itself, so a caller cannot claim a membership.

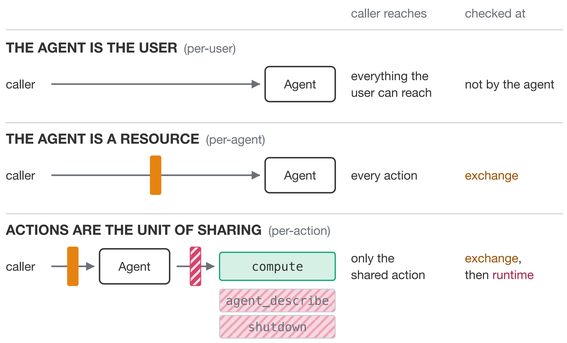
What Hall Pass adds

Hall Pass makes the action the unit of sharing in Academy. My work, PR #430; token checks, group lookup and whole-mailbox sharing were already there.

- The owner declares which Globus groups may call each action, `@action(sharing=[g])`. Undeclared actions stay owner-only.
- The exchange admits a caller who owns the mailbox or is in a group it is shared with. It never sees which action is called.
- The exchange stamps the caller's verified groups on each message, and the agent's runtime checks them against the action called.
- Lifecycle is separate: `control_groups` may shut down or cancel without calling any action; any group the agent is shared with may ping.
- A mailbox with no owner record now admits no one (it admitted anyone before), and every denial is logged with a reason.

Sharing models

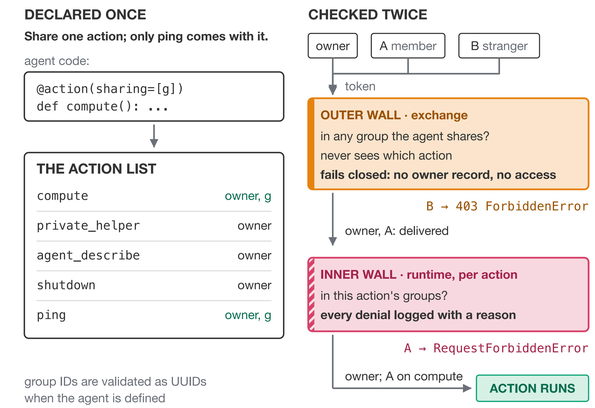
There are three ways to share an agent, and where the check sits decides what a caller can reach [4]. Hall Pass is the third, and the only one measured.



Design

Two checks: may this caller reach the agent, and may it run this action?

- **Trust and scope.** Trusted: the exchange, Globus Auth and Groups. Stopped: non-members, and members calling past their actions. Not stopped: a permitted action runs with the owner's credentials, and removing a member takes up to 60 s.



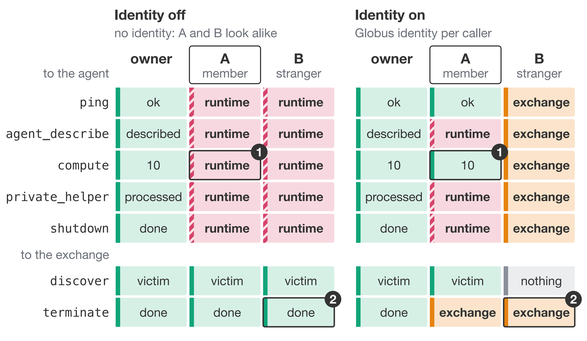
The owner is the user who launched the agent; the agent has no Globus identity of its own.

Who can do what

The same probes and agent code, with the exchange's identity check off, then on:

- One agent, the "victim", is probed by its owner, member A and stranger B: three Globus accounts held by one person.
- The runtime always serves the owner's mailbox, which registered the agent.

With identity on, member A can run only the shared action

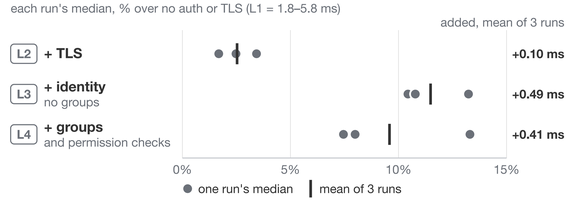


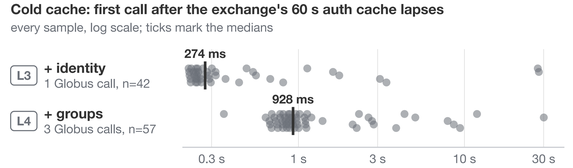
1 A's `compute` is refused with no identity, and served once the exchange knows A is in g.
2 Identity off, stranger B can terminate the agent. Identity on, only the owner can.

Overhead

- We time a request and its response over loopback on one laptop, through four configurations of one exchange: no auth (L1), + TLS (L2), + identity without groups (L3, `resolve_groups=False`), + groups and permission checks (L4).
- Warm, TLS and the checks add about 0.5 ms (TLS alone 0.10 ms): 0.49 ms at L3, 0.41 ms at L4, the mean of 3 runs' medians, or 7–13% in single runs.
- When the 60 s auth cache lapses, the next call makes Globus round trips: median 274 ms for identity, 928 ms with groups. 14 of 57 cold calls with groups took over 2 s, and one timed out at 30 s. At most once a minute per caller.

Warm, TLS and authorization add about half a millisecond





L1–L3 were timed by the owner and L4 by caller A, since the owner skips the per-action check; read no order into warm L3 and L4. Cold cache hits (under 100 ms: 18 of 60 at L3, 3 of 60 at L4) are excluded.

Demo: shared SAM 3 on Anvil

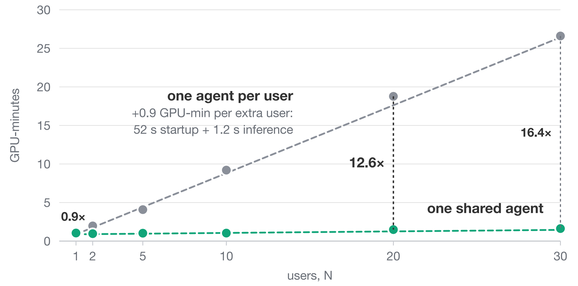
- Each user segments one fixed box in 20 images with SAM 3 [1] on an A100 at Purdue Anvil.
- The agent shares one action, `segment`, with one Globus group; every request passes both checks.
- Every private copy pays the 52 s startup again; one shared agent pays it once.

52 s of setup buys 1.2 s of GPU inference



What sharing saves

Sharing one agent pays startup once, not once per user



At N=20: 18.8 vs 1.49 GPU-minutes.
Per-process timers. Slurm (sacct), which also bills job setup and idle GPU, gives 7.8×.

Medians of 3–8 runs, IQR bars behind the dots. The gap grows with N: each private copy adds a startup. At N=1 the ratio is 0.9×: one user costs slightly more on the shared agent than on a private copy.

- Sharing the whole agent would save the same; Hall Pass gets it while sharing only `segment`. Both setups ran with the checks on, every client signed in as A.
- At N=20 a request waits 17 s longer than alone (19 s at the median), still less than a private copy's 52 s startup.

≈ 0.5 ms
warm cost of TLS + authorization, mean of 3 runs' medians
7–13% in single runs, loopback

7.8–12.6×
the GPU time for 20 private copies as for one shared agent
Slurm accounting to process timers

Conclusion

- Making the action the unit of sharing lets one agent serve several audiences: a member can run the shared action and no other.
- Warm, TLS and the checks add about 0.5 ms. A cold call with groups, at most once a minute per caller, has a median of 928 ms; one timed out at 30 s.
- Next: concurrent callers, other sites, and limits on what shared actions reach.

References

[1] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, et al. 2025. SAM 3: Segment Anything with Concepts. arXiv:2511.16719.
[2] Globus. Globus Groups API. https://docs.globus.org/api/groups/. Accessed August 6, 2026.
[3] Alok Kamatar, J. Gregory Pauloski, Yadu Babuji, Ryan Chard, Mansi Sakarvadia, Daniel Babnigg, Kyle Chard, and Ian Foster. 2026. Empowering Scientific Workflows with Federated Agents. In *Proc. IPDPS 2026*. IEEE, 1403–1418.
[4] Butler W. Lampson. 1974. Protection. *ACM SIGOPS Operating Systems Review* 8, 1, 18–24.
[5] Steven Tuecke, Rachana Ananthakrishnan, Kyle Chard, Mattias Lidman, Brendan McCollam, Stephen Rosen, and Ian Foster. 2016. Globus Auth: A Research Identity and Access Management Platform. In *Proc. IEEE e-Science 2016*. IEEE, 203–212.

Acknowledgements. Claude (Anthropic) was used to assist in preparing select figures. This work used computational resources provided through the NSF ACCESS program under allocation CIS250223, "Multi-site serverless scheduling for science."

Code: PR #430, academy-agents/academy

Contact: azhang3@ucls.uchicago.edu