

# Systems-Facing call, 2026-09-17 @ 1p ET/ 12p CT/ 11a MT/ 10a PT

## Topic: What is Aleph?

Aleph is a proof-of-concept multi-model inference platform built by Rahim Khoja and Karim Ali in the University of Alberta's Research Computing team, running on Vulcan. It provides shared, local access to scientific and general-purpose AI through one endpoint and API key. Familiar tools—including Claude Code, Open WebUI, notebooks, and research applications—can connect through OpenAI- and Anthropic-compatible APIs, keeping inference and the research context it processes on the infrastructure we operate.

A central motivation is to make scientific model inference a straightforward part of research code. A Slurm job can call a model, receive a result, and use it in the next computation. A domain researcher studying proteins or materials should be able to use relevant models without first becoming an expert in serving runtimes, dependency management, or GPU deployment. Aleph handles that infrastructure, while researchers choose appropriate models and evaluate their results. Sharing running model servers also reduces repeated setup, loading, and separate GPU reservations.

The catalog spans proteins, genomics, materials, weather, vision, and more, but most demand today comes from everyday AI: chat, reasoning, coding, and agents. Both uses support the same direction—making locally hosted AI a reusable service for people, applications, and computational workflows.

The project draws inspiration from CERN's model-serving work, Argonne's inference service, and the U.S. Department of Energy's Genesis Mission. Our longer-term goal is an autonomous researcher that lives on the cluster: an agent that reviews literature and previous experiments, submits jobs, calls scientific models, checks results, and maintains an auditable record. Aleph supplies one part of that foundation, giving sustained research workflows access to local inference with control over data and compute resources.

This session covers why we built Aleph, how its architecture fits alongside HPC, and what real workloads are teaching us. We'll walk through provisioning, GPU sharing, model deployment and routing, scaling, and usage accounting, and discuss what remains experimental as we explore shared inference as a research computing service.

[https://docs.google.com/document/d/16cgFUUC3\\_WdlmIsnmUeY1hc5lRJB1gY17hnhNhRdnAM/edit?usp=sharing](https://docs.google.com/document/d/16cgFUUC3_WdlmIsnmUeY1hc5lRJB1gY17hnhNhRdnAM/edit?usp=sharing) Presentation Cards being Read

<https://inference.vulcan.alliancecan.ca/> This is Aleph the Live Service

<https://docs.alliancecan.ca/wiki/Aleph> This is the How to use Aleph ( Getting a Key )

<https://github.com/ualberta-rcg/aleph> This is the Aleph Public Repo ( Can you tell its a Vibe Coded Monstrosity? )

<https://github.com/ualberta-rcg/warewulf-rke2-hami> This is the image we boot to via WareWulf

### **Speakers:**

Rahim Khoja, HPC & AI Infrastructure Engineer, University of Alberta

Karim Ali, HPC & AI Infrastructure Engineer, University of Alberta

### **Link To Video Recording:**

<https://www.youtube.com/watch?v=0ScCZZBs22o>

### **Zoom Meeting Room:**

<https://utah.zoom.us/my/carcc?pwd=TjFuR3VVM2d5eE5zWnEvWWxDTFBCUT09>

Meeting ID: 824 051 8198

Password: 31415926

One tap mobile

+13462487799,,8240518198#,,#31415926# US (Houston)

+16699006833,,8240518198#,,#31415926# US (San Jose)

Dial by [your location](#)

+1 346 248 7799 US (Houston)

+1 669 900 6833 US (San Jose)

**NOTE: This meeting will be recorded via Zoom.**

### **Link To Slides:**

**TBA**

**CaRCC YouTube Channel:** <https://www.youtube.com/carcc>

### **CaRCC YouTube Channel Systems-Facing Playlist:**

[https://www.youtube.com/playlist?list=PLV-SC0CHLTwehApeMg\\_fj5VbYndET3yT](https://www.youtube.com/playlist?list=PLV-SC0CHLTwehApeMg_fj5VbYndET3yT)

### **CaRCC YouTube Channel Data-Facing Playlist:**

<https://www.youtube.com/playlist?list=PLV-SC0CHLTwdrPXN8zdres6aovVAoEJlq>

**CaRCC Code of Conduct:** <https://carcc.org/about/carcc-code-of-conduct/>

**Join the CaRCC Slack Workspace:**

[https://join.slack.com/t/carcc/shared\\_invite/zt-9lhv4cfm-wy7RpbNI7V9qLJcvtzq8hA](https://join.slack.com/t/carcc/shared_invite/zt-9lhv4cfm-wy7RpbNI7V9qLJcvtzq8hA)

**Sign-In (Name / Affiliation / Email): (Max on call: 56)**

1. Ric Anderson / Sys Admin / ric@arizona.edu
2. Jim Leous / Consultant / [jim.leous@gmail.com](mailto:jim.leous@gmail.com)
3. Steve Anthony / Lehigh University / sma310@lehigh.edu
4. Ross Dickson / ACENET & Dalhousie U / ross.dickson@ace-net.ca
5. Michael Tang / University of British Columbia / michael.tang@ubc.ca
6. Evan Linde / Oklahoma State University / elinde@okstate.edu
7. Michael Kinach / ACENET & Memorial University of Newfoundland / michael.kinach@ace-net.ca
8. Chris Endemann / University of Wisconsin-Madison / endemann@wisc.edu
9. Mike Cammilleri / University of Wisconsin-Madison / mjcammilleri@wisc.edu
10. Dan Bongert / University of Wisconsin - Madison / dbongert@wisc.edu
11. Scott Delinger / Altadel Consulting Ltd. / scott@altadel.com
12. Matthew Lambert / University of Tennessee / mlambe28@utk.edu
13. Alan Sill / Texas Tech University / alan.sill@ttu.edu
14. Robert Yelle / Oregon State University / robert.yelle@oregonstate.edu
15. Nick Eggleston / Kansas State University / njeggleston@ksu.edu
16. Brahm White-Gluz / Digital Research Alliance of Canada / brahm.whitegluz@alliancecan.ca
17. Michael Ewan / Portland State University / mewan@pdx.edu
18. Jacob Boschee / University of British Columbia / jacob.boschee@ubc.ca
19. Ben Winjum / UCLA / bwinjum@oarc.ucla.edu
20. Sajid Ali / NYU / ss19980@nyu.edu
21. Davide DelVento / Quantinuum / [delvento@gmail.com](mailto:delvento@gmail.com)
22. Fernanda Foertter / University of Alabama / [ffoertter@ua.edu](mailto:ffoertter@ua.edu)
23. Mike Hutcheson / Baylor University / mike\_hutcheson@baylor.edu
24. Katherine Holcomb / University of Virginia/ kah3f@virginia.edu
25. Patrick Schmitz / Semper Cogito & Internet2 Consultant / patrick@sempercogito.com
26. Nicholas Polys / Virginia Tech
27. Sarah Huber / University of Victoria / sahuber@uvic.ca
28. Rose Stein / CIQ / [rstein@ciq.com](mailto:rstein@ciq.com)
29. Grigory Shamov / UManitoba / [Grigory.Shamov@umanitoba.ca](mailto:Grigory.Shamov@umanitoba.ca)
30. Tyler Collins / SHARCNET - tk11br@sharcnet.ca
31. Heechang Na / Ohio Supercomputer Center / hna@osc.edu
32. Ryan Sparkes / Digital Research Alliance of Canada / ryan.sparkes@alliancecan.ca
33. Kurt Roberts / University of Cincinnati / Kurt.Roberts@uc.edu
34. Ben Nickell / Idaho National Laboratory / ben.nickell@inl.gov
35. Kim Wilkins / University of Nevada Reno / kwilkins@unr.edu

36. Addis O'Connor / Stanford University / [addiso@stanford.edu](mailto:addiso@stanford.edu)
37. Terry Stilwell / Old Dominion University / [tstilwel@odu.edu](mailto:tstilwel@odu.edu)
38. Graham Morley / Morley Media Group / [graham@morleymedia.dev](mailto:graham@morleymedia.dev)
39. Jim Burton / Clemson University / [jburt2@clemson.edu](mailto:jburt2@clemson.edu)
40. Jaynal pervez /UT San Antonio /[pervez@uthscsa.edu](mailto:pervez@uthscsa.edu)
41. Umesh Upadhyaya /HPCNepal/ [umesh@hpcnepal.org](mailto:umesh@hpcnepal.org)
42. Mike Wolfe/ Florida State University / [mw26l@fsu.edu](mailto:mw26l@fsu.edu)
- 43.
- 44.
- 45.
- 46.
- 47.
- 48.
- 49.
- 50.
- 51.
- 52.
- 53.
- 54.
- 55.
- 56.
- 57.
- 58.
- 59.
- 60.
- 61.
- 62.
- 63.
- 64.
- 65.
- 66.
- 67.

---

Volunteer to contribute to Brag Day session (celebrate you or your teams' successes (project/planning/initiative, etc):

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

7.

### Call Notes:

- Rahim Khoja and Karim Ali (University of Alberta) Presenting
- The Upper Bound Conference in Edmonton, Alberta, CA will soon host our presenters if you want to meet them in person (<https://www.upperbound.ai/>)

### Intro to Aleph

- Aleph is a local inference server, simply a webserver connected to GPUs that server tokens
- Everything is locally hosted
- Aleph is also able to scale vertically and horizontally with load using pods
- Aleph uses similar technology to OpenAI, but serves multiple models and users can request the model that they feel will best serve their needs
- Models stay in memory after the first request to serve multiple users that want the same models
- Aleph was not necessarily designed to serve "generalist models" although it can. Aleph is more geared toward scientific computing, allowing researchers to "just ask for the model they need and walk away"
- Aleph as a service takes away the burden of learning how to construct the "systems" aspect of serving and querying a model. It also removes the duplication of effort and resources across the institution.
- The backend is a SLURM job
- Who uses it? People. Also applications with agent to agent traffic and Claude Code is also run in this environment.
- 7 nodes with L40s GPUs are available for inference. The models are allowed to scale to 0 to save resources when not in use but models can also scale to take up the full pool.
- Funding for hardware is a blocker at the moment and the resource is rarely empty
- The original intention was to be a "science backend", but it mostly isn't being used that way. Most of the use is chat based queries, serving 11B tokens in the first week.
- Original inspiration came from the CERN project K-Serve. Further research was done into mostly US-based science based institutions doing similar work. Rahim admits they are a team of "two guys reading research papers and nodding their heads".
- The end goal for this project is to become a resource for "autonomous science", automating the "boring parts" of doing science and allowing researchers to focus on what they're good at

### System Architecture

- Warewulf builds the base compute nodes, allowing for a stateless node provisioning letting a node become a SLURM node or Aleph worker dynamically
- RKE2 then orchestrates models on the worker nodes and scale. Project HAMi then allows for GPU "slicing" meaning a whole model won't take up a whole GPU if it's not required, making better use of hardware.
- KServe describes the container, storage, and other pod level setup tasks. Knative decides how many their should be. At no point is a GPU "consulted".

- From there vLLM, Text Embedding Inference, and nvidia's NIM, and other custom servers are loaded based on need. Model weights live on an NFS share all nodes can access.
- Adding a new model can sometimes present a challenge. Model weights and other data live in pvc.yaml. Inferenceservice.yaml holds the startup info for the engine. Details.yaml is the "model card", providing a config map for the gateway to model. Some models use different verbiage such as "thinking" or "reasoning" for the same operation so this file helps to standardize for APIs querying this service.
- Claude Code has allowed for rapid deployment of this service, creating the bulk of these yaml files and allowing engineers to verify its output instead of create the info from 0.
- There are six checkpoints before your tokens reach a GPU. MetalLB, Traefik, cert-manager, Tyk, gateway translation, Istio, and then the Model itself all handle specific parts of the chain, providing for graceful failures if the hardware is full or model is unavailable for some reason.

#### Tracking usage

- This remains a pain-point for the project. Tracking input-output of tokens, who us using, what model they're using, duration/GPUs held, whether a novel or unloaded model has to get loaded, etc. Currently this is not a billed service but tracking usage is important to show value to the institution and one day a charge for use will likely have to come

#### What's next?

- Batch Inference. Hosting an API to do many-to-one computation of lots of data to the same model
- Aleph to Aleph routing. Being able to access models running on another Aleph system to make use of more powerful hardware or specialized models at another institution without having to have an actual account or instance running on that outside Aleph instance
- Karmada. Orchestrating scaling of models up or down across geographic boundaries or multiple institutions

#### Audience Questions

- Heechang Na (Ohio) - How do you track GPU usage? Are you tracking at a user or group level?
  - At the moment, GPU usage to make the API response is captured and stored locally and will be used in the future for the billing aspect. Also being tracked is which models and keys are being used by each user
- How does this project compare to Dynamo?
  - Neither presenter was familiar with this project but Science Inference Clusters were researched in creating this project and no really good published project was found
- Steve Anthony - How do you deal with periods of high demand for both Slurm and Aleph workers? Is one preempted or prioritized over the other? Are Slurm jobs killed to respond to real time Aleph chat requests?
  - This is what we hope to get to, complete autonomy. Warewulf *can* do this, but the mechanics of making that work are still in progress. Currently it's more manual.
- Davide Del Vento - Have you considered deploying workloads with SLURM instead of RKE2
  - This was considered but RKE2 was implemented for better scaling of load
  - HAMi also is better implemented in Kubernetes which helps keep utilization of hardware as high as possible

- Jaynal Pervez - How can I get some help implementing a similar project at my institution?
  - You can reach out to us. We are a team of 2 so our time may be limited but we love sharing what we know and what we've done.
- Tyler Collins - What do you do for security reviews? If someone starts abusing the resource and DDoSing the resource?
  - There is currently no SLA for this because it's not currently a production service. Currently there hasn't been a need for this but as this becomes a more formalized offering this may get revisited.
  - We also haven't focused much on "security" both from a people standpoint or a model standpoint.
- Chris Endeman - How many users do you think you can support using this for agentic coding? I think those context windows add up fast. I'd love to offer this as service to researchers, but I'm wondering how large of a GPU fleet we'll need.
  - Currently 7 nodes each with 4 L40s GPUs are dedicated to Aleph. This can serve 4000 output tokens per second comfortably but this can only load smaller models due to the memory limitations of the GPUs. Calculating this need really depends on the models you want to run.
  - Currently there are 40 regular users. Open WebUI uses an incredible amount of tokens as well as some other GUIs and APIs and currently they're well supported with the current infrastructure
- Nick Eggleston - Do you have any model limitations on what you can host? Mostly asking about things like Deepseek or anything else that may try to escape its sandbox?
  - No. We don't stop anything that may inhibit science and we don't look at what the users are doing. We'll deploy any model that will fit on our servers.
- Alan Sill - do any of the AI-driven features that are used to operate aleph at the system level running as root, or are they done in non-privileged accounts by API calls?
  - None of the servers hosting Aleph are connected to the rest of the institution. The containers are currently run as root, but this was a calculated risk and is being evaluated to be mitigated. The worst thing that a rogue agent could get is token usage from these systems so the risk is low.
- Qi - Are most of the tokens consumed during working hours? So at night the GPUs are kinda rested.
  - Some workloads will run for multiple days depending on what the users are running. General interactive work will spike during waking hours but workloads that are "set and forget" may run longer.
- Fernanda Foertter- How much money do you think you might be saving the users?
  - There is no upper bound on the number of tokens you can spend (tongue in cheek), but the biggest selling point of this system isn't currently the cost but that it's hosted on-site and allows a secure environment for research.

#### **Upcoming Presentation Plans:**

- 

**Suggest future topics for a Systems-Facing call (Feel free to add one!):**

•