

Firefly on CANFAR

Successful deployment and a test of Interoperability (and a suggestion for platform API standardization)

IVOA Interop, College Park, June 2025

CADC: Brian Major, Shiny Brar, Stephen Gwyn Caltech: Gregory Dubois-Felsmann, Loi Ly





Summary

- > Firefly now available on CANFAR deployments
- > Interesting interoperability challenges
 - Authentication and Authorization: an application on a science platform (Firefly) using a variety of potentially protected TAP services
 - Container standardization across platforms
 - Interacting with other tools running on the platform and other platforms – standard discovery needed

Firefly on CANFAR

- Firefly: https://github.com/Caltech-IPAC/firefly
 - "Web-based UI library for astronomical data archive access and visualization developed at Caltech"
 - Catalogs and rich IVOA support Registry, DAL protocols, including TAP
- CANFAR: https://github.com/opencadc/science-platform
 - container-based interactive and batch science-platform

Çanfar		Documentation Services ▼ About Open Source Support ▼ major
cience Portal		
firefly1 skaha/firefly20 started: 2025-05-28 17:15 expires: 2025-06-01 17:15 memory: <none> CPU cores: <none> CPU cores: <none></none></none></none>	ning) 25.2 UTC UTC (/4G /0.1	K
New Session	Help	Platform Load
type vontainer image vontainer image vontainer image vontainer vo	firefly ~ skaha ~ firefly:2025.2 ~	CPU usage 0 500 1,000 1,500 2,000 2,702 Available RAM: 783.05GB / 10019.05GB used free Memory usage
session name 🕄	firefly2 Launch Reset	0 2.000 4,000 6,000 8,000 10,019.05 Running Instances: 391 asssion desktopApp headless instances
		0 50 100 150 200 250 300 350 391 last update: 2025-05-28 17:14 UTC





IC C 1 of 337 > >I (1 - 100 of 33,626)

🔮 7 Tr 🖬 🕞 🛈 🕸 🍾

Interoperability Considerations

- 1. AAI: an application on a science platform (Firefly) using a variety of potentially protected TAP services
- 2. (Interactive) container standardization across platforms
- 3. Interacting with other tools running on the platform and other platforms

1. Authentication and Authorization

- Authentication and Authorization to run Firefly and use compute resources handled upfront by CANFAR and group membership checks (OIDC and GMS).
- But how to make authenticated queries from Firefly to protected TAP services? (eg, CADC user-managed YouCat tables "TAP-next").
- And is this same problem other tools (eg TOPCAT) face?

1. Authentication and Authorization (cont)

- Current Solution: Use Firefly Token Relay plugin support

- Decide which credentials to send to the chosen TAP server
- But opens up questions about token reuse:
 - When should tokens be relayed? Same domain?
 - Does the scope of the token need to be modified?
 - Does the Token exchange protocol help?
- Limitation: Only CANFAR credentials available, so can only interact with CANFAR services.
- TOPCAT: built-in (prototype) support for negotiating with services via HTTP Challenges (401) to acquire credentials for services from associated organizations.
- Open question: how to support authentication to multiple organizations in tools like Firefly that expect AAI to be handled at a higher layer?

2. Interactive Container Compatibility

- See previous talk in Malta for context:
 - <u>https://wiki.ivoa.net/internal/IVOA/InterOpNov2024GWS/IVOA-Malta-ImageMetad</u> <u>ata.pdf</u>
- Compatible container execution across platforms
 - Batch Execution easy: command and arguments built in, or provided. No networking to deal with.
 - Interactive more difficult...
- Core interactive "types": notebook desktop carta firefly contributed
 - contributed an interface with an *attempt* of standard rules for user-contributed interactive containers
 - batch/headless: no type required because no interactivity

2. Interactive Container Compatibility (cont) Firefly vs Contributed Interactive "Types"

	Firefly	"Contributed" Rule	Summary
Ingress: container port	 default 8080 (can be changed) 	 listen at 5000 	 Apps not likely to agree on universal port Deployments allow the specification of port
Ingress: URL base path	 always expects baseURL + /firefly need URL path translation in ingress 	 access path should be self discovered 	 Most apps require base path to be configured Some don't: vscode, pluto Path a runtime consideration Apps should self discover base path
Startup	 container entrypoint (or custom startup) plugin jar files provided at build time basic app customization through runtime ENV 	 entrypoint or /skaha/startup.sh 	 Use container entrypoint ENV for runtime config

2. Interactive Container Compatibility (cont) Summary

"Contributed" definition should be modified to:

- Platforms should allow for the specification of port(s)
 This aligns with metadata in execution broker
- Apps should self discover base path
 - Not be required to run at / or a specific subpath
- Use container entrypoint
 - But allow config overrides via ENV vars at runtime

3. Interacting with other tools running on the platform or other platforms

- Firefly has an API Jupyter Notebooks running in CANFAR can use the API of their Firefly session.
- All containers mount user-storage data visible to all via:
 - o /cavern/home/...
 - o /cavern/projects/...
- cavern is a POSIX based VOSpace so that API is available too
- CANFAR API (and CLI) allows the discovery of sessions across multiple deployments. Exposes URLs (thus their APIs) to all running interactive sessions.

3. Interacting with other tools running on the platform or other platforms (cont)

- CANFAR API (skaha), and CANFAR CLI, allow the discovery of sessions across multiple deployments. Exposes URLs (thus their APIs) to all running interactive sessions.
- With a session (and tool APIs) running at multiple platforms, users can have "local" data access in multiple locations.
- Standardizing the session launch and discovery API would allow distributed computation across platforms.
 - Should the DSP working group start this work?

Thank you

brian.major@nrc-cnrc.gc.ca

nrc.canada.ca • info@nrc-cnrc.gc.ca