

Annual Review for OpenAssetIO

Rob Fanner, David Feltell
2024-08-21

/* ACADEMY
SOFTWARE
FOUNDATION

OpenAssetIO

Brief Description:

An interoperability standard for the tools and content management systems used in media production.

TSC Chairperson:

David Feltell <david.feltell@foundry.com>

TSC Members and Affiliations:

Matt Daw (MovieLabs), Ondřej Samohel (Ynput), Peri Friend (Foundry), Mathieu Mazerolle (Foundry), Tom Cowland (Cinesite)

Contributed by:

Foundry

Key Links:

Github: <https://github.com/OpenAssetIO>

Documentation:

<https://openassetio.github.io/OpenAssetIO/>

Artwork: [OpenQAssetIO-logo.svg](#)

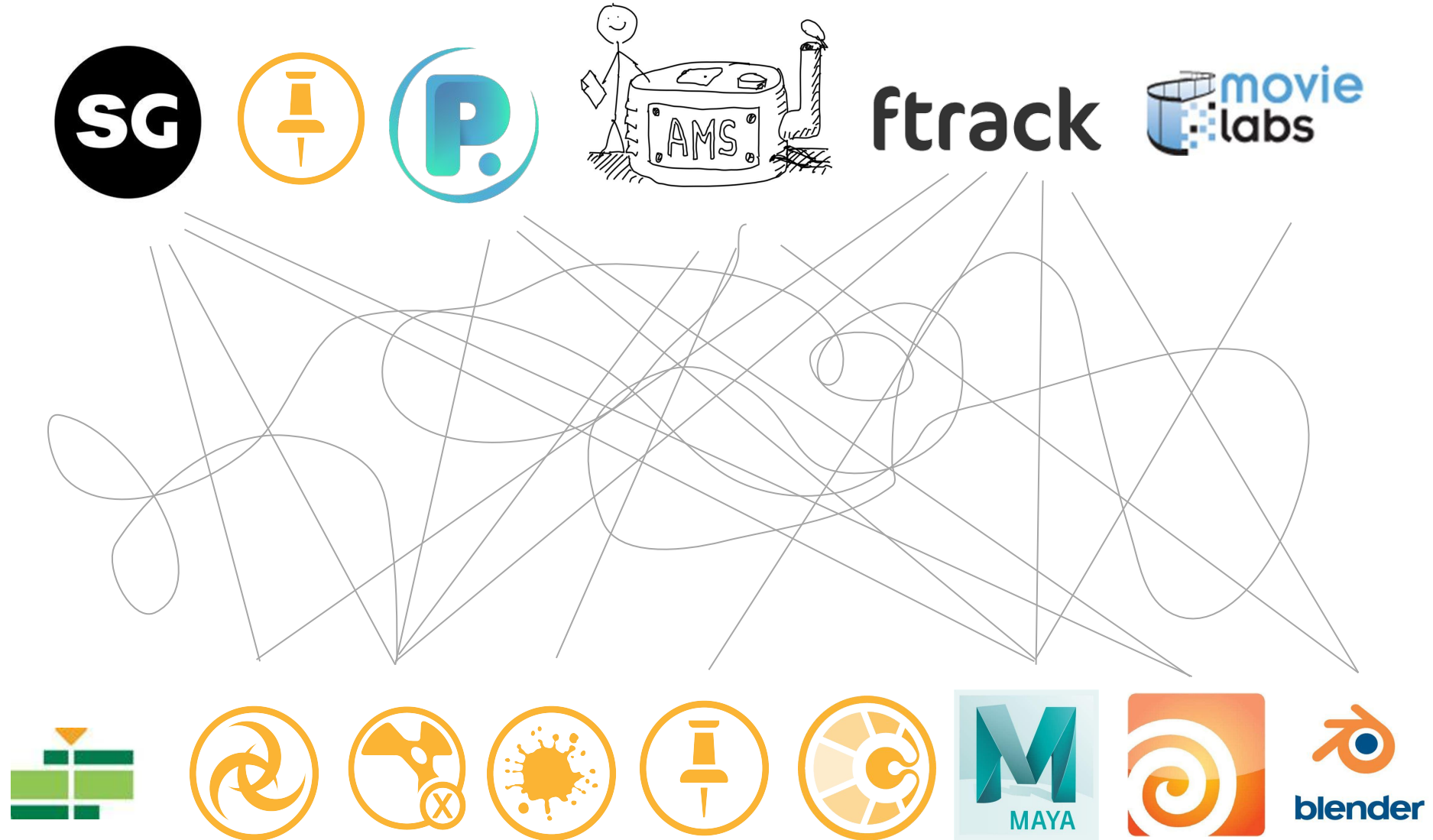
Mailing lists:

- [openassetio-discussion](#)
- [#openassetio](#)

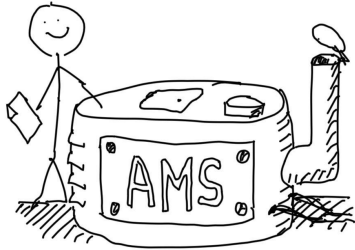
OpenSSF Best Practice Badge URL:

<https://bestpractices.coreinfrastructure.org/en/projects/6046>

OpenAssetIO



OpenAssetIO



ftrack



 **OPENASSETIO**



Incubation Project review criteria

Note: No changes since 2023 review. Our primary focus areas have been on a) dogfooding via DCC integrations and b) locking in a v1 API.

To be accepted at the Incubation stage, a project must meet the Sandbox requirements plus:

- ✓ Have completed and approved the Technical Charter and agree to transfer any relevant trademarks to The Linux Foundation or its affiliate, LF Projects, LLC, and to assist in filing for any relevant unregistered ones. The ASWF maintains a template for projects to use.
- Have defined its technical governance, including:
 - ✓ A LICENSE file in every code repository, with the license chosen an OSI-approved license.
 - ✓ A README file welcoming new community members to the project and explaining why the project is useful and how to get started.
 - ✓ A CONTRIBUTING file explaining to other developers and your community of users how to contribute to the project. The file should explain what types of contributions are needed and how the process works.
 - ✓ A CODEOWNERS or COMMITTERS file to define individuals or teams that are responsible for code in a repository; document current project owners and current and emeritus committers.
 - ✗ A CODE_OF_CONDUCT file that sets the ground rules for participants' behavior associated and helps to facilitate a friendly, welcoming environment. By default, projects should leverage the Linux Foundation Code of Conduct unless an alternate Code of Conduct was previously approved.
 - ✓ A RELEASE file that provides documentation on the release methodology, cadence, criteria, etc.
 - ✗ A GOVERNANCE file that documents the project's technical governance.
 - ✗ A SUPPORT file to let users and developers know about ways to get help with your project.

Incubation Project review criteria (cont.)

Note: No changes since 2023 review. Our primary focus areas have been on a) dogfooding via DCC integrations and b) locking in a v1 API.

- ✓ Have achieved and maintained an [OpenSSF Best Practices Badge](#) at the passing level.
- ✓ Have had a successful license scan with any critical issues remedied.
- ✓ Have a defined project mission and scope
- ✓ An overview of the project's architecture and features defined.
- ✓ A project roadmap defined, which should address the following questions.
 - What use cases are possible now?
 - What does the next year look like in terms of additional features and use cases covered?
- Community and contributor growth assessment
 - ✓ The current number of contributors and committers, and the number of different organizations contributing to the project.
 - ✓ Demonstrate a sustained flow of commits / merged contributions
 - ✓ A credible plan for developing a thriving user community, in particular expanding the number of committers and contributors?
 - Outline of the plan for the project to complete the requirements for Adopted Stage
- ✗ Obtain an affirmative vote of the TAC.

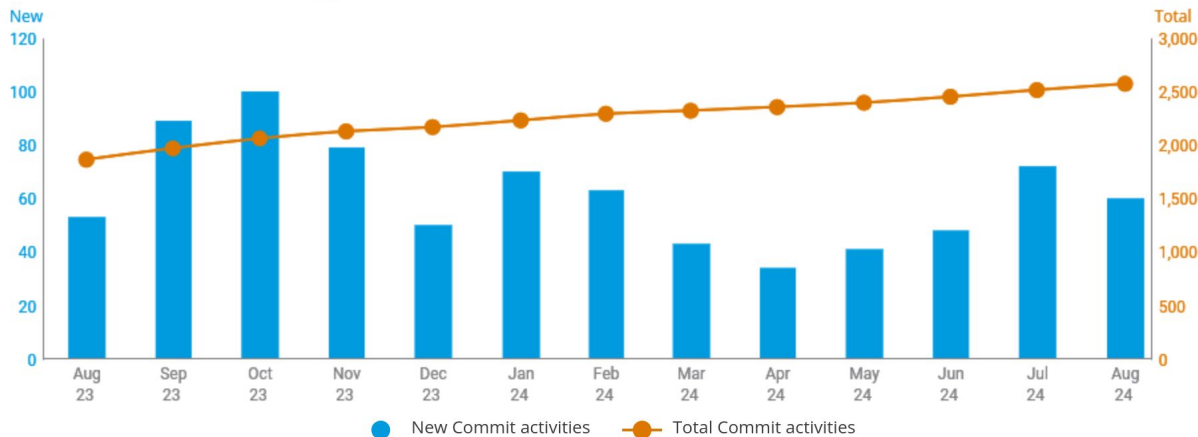
Contributions

Last 12 months:

- 701 commits across 8 repos
 - 413 in OpenAssetIO repo / 377 in other repos
- 272 GitHub issues closed vs. 243 issues opened
- Core contributors down from 3 to 1 currently
 - Looking to gear up again as and when we can
 - Major contributions on developer experience, review and validation via Foundry product teams

Commit Activities ⓘ

Commit activities **decreased by 13%** 📉 vs. the previous time period.



Contributor Leaderboard ⓘ

Pull Request Comments are **down** 📉 by 20.94% vs. the previous time period.

Metric: Pull Request Comments ▼

Rank ▲	Contributor	Source	Pull Request Comments	Change	% Contributions
1	feltech		919	-53	45.07%
2	elliottcmorris		624	-16	30.6%
3	foundrytom		296	-573	14.52%
4	foundry-markf		87	+42	4.27%
5	themissingcow		43	+43	2.11%
6	tomc-cinesite		31	+31	1.52%
7	SamCrooksFoundry		15	+15	0.74%
8	brycebrazen		14	+14	0.69%
9	DuncanHopkinsFoundry		6	+5	0.29%
10	lpapp-foundry		1	-5	0.05%
11	antirator		1	0	0.05%
12	hartmutbehrens		1	+1	0.05%

Organizations contributing and/or using in production

FOUNDRY.



ftrack

+?



Key Achievements in the past year

- **OpenAssetIO v1.0 (beta)**
 - C++ Manager plugins (previously Python-only)
 - Hybrid C++ & Python Manager plugins
 - Utilities to convert URL-to-path and path-to-URL
 - Token substitution for assets made of file sequences
 - Hoping to lock-down API over the next quarter
- **OpenAssetIO MediaCreation (alpha)**
 - Building common ontology of traits for VFX industry
 - Future: Can be based on Movielabs Ontology for VFX use cases
- **New Industry Integrations**
 - Nuke: Basic support available now in Nuke 15.1v1
 - Katana: Work in progress on Katana support via Asset API
 - ftrack: Forked our prototype and now maintain themselves
- **Pending...**
 - UI Delegation loosely couples DCCs and Asset Systems
 - Further improvements to Nuke & Katana integrations

Hosts: Hello OpenAssetIO

This notebook illustrates how to initialise an OpenAssetIO session, resulting in a `Manager` instance for communication with a particular asset management system's plugin. We then perform a simple query of the manager, to `resolve` a property of an entity.

Setup

The core OpenAssetIO library, `openassetio`, provides the "verbs" for interaction with a manager. I.e. it is the API we use to communicate with the manager plugin. Most data exchanged with the manager is opaque from the perspective of the core API.

The industry-specific MediaCreation library, `openassetio-mediacreation`, provides the "nouns" for describing the data exchanged with the manager. I.e. it provides the structure of the data exchanged with the manager.

```
In [1]: try:
import openassetio
import openassetio_mediacreation
except ImportError:
print("This notebook requires the packages listed in `resources/requirements.txt` to be installed")
raise
```

We'll also pull in a helpers library to help us format our output - this is a library just for these notebooks and not part of the OpenAssetIO ecosystem generally.

```
In [2]: from resources import helpers
```

Bootstrap OpenAssetIO

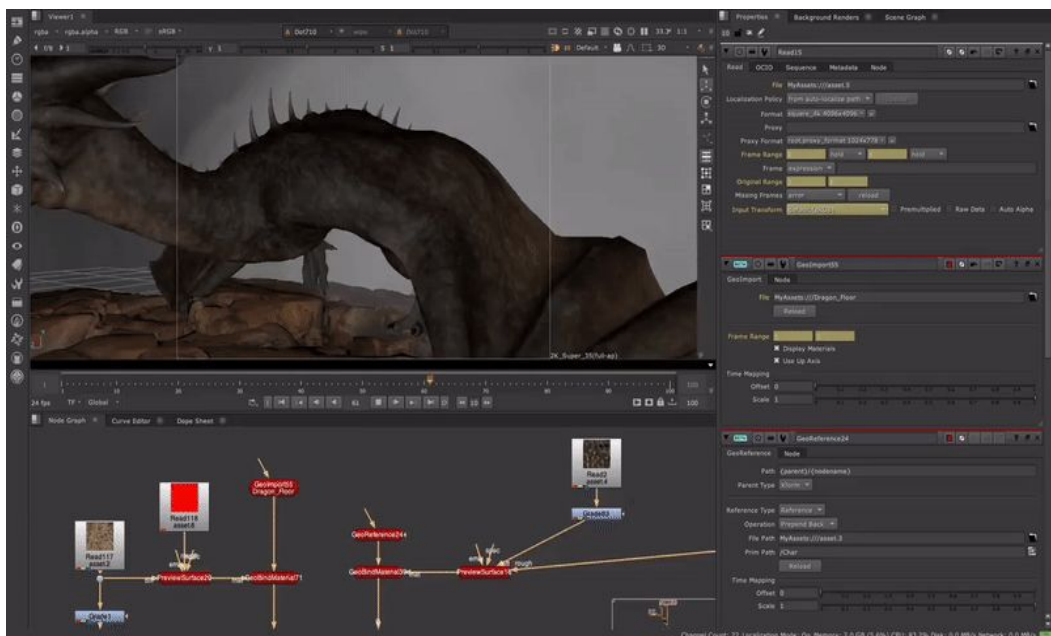
The code initializing the OpenAssetIO API is known as a "Host".

More docs, examples, and [Jupyter notebooks](#) available for playing with the API

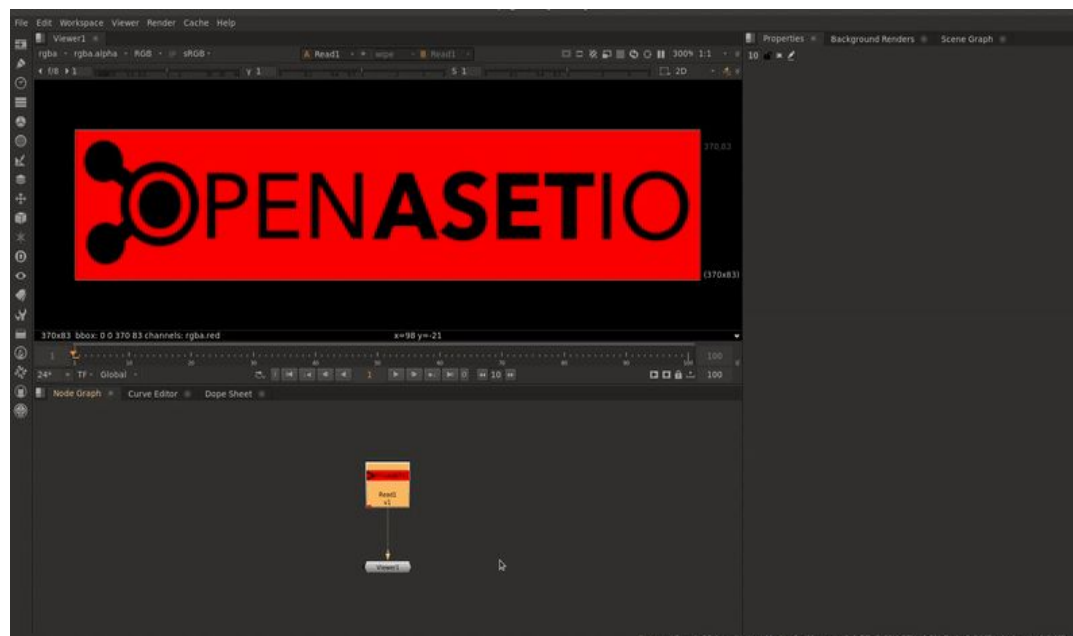
Key Achievements in the past year



OpenAssetIO in Nuke



Asset references with Traits automatically ingested into Nuke



UI Delegation Preview

Connect UI from custom or commercial asset managers over core OpenAssetIO API

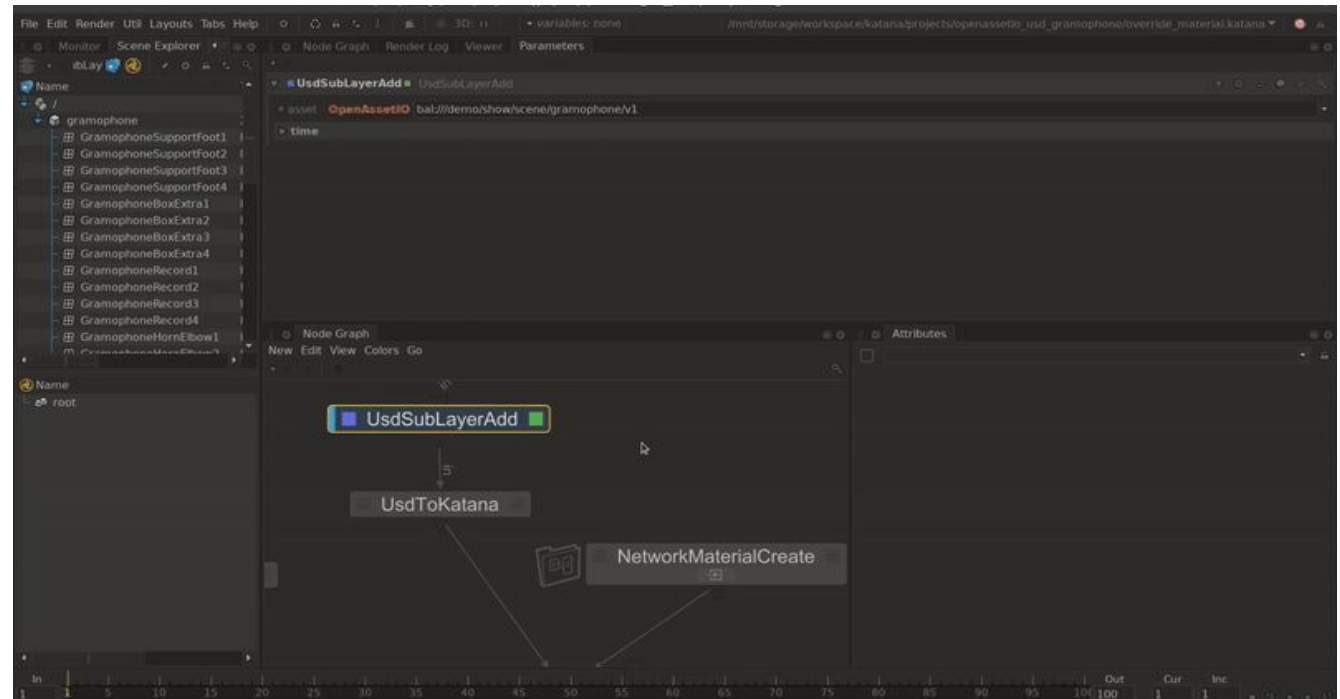
Key Achievements in the past year



OpenAssetIO in Katana

Work in Progress

- Extends Katana Asset API
- Supports USD AR2
- Will be open sourced



Areas the project could use help on

Significant strides made on proving out in-anger DCC integration, and very close to v1.0.0 release now

- Encouraging prototyping, feedback and real-world adoption
 - More DCC vendors, asset management vendors & studios wanted
 - Overall value proposition ramps up significantly as adoption goes up!
- Documentation
 - Already at a high standard, though users are keen on more examples and simple starting points
 - Help with resource on checking/writing documentation & example code would be very welcome
- Trait specifications
 - MovieLabs Ontology starting to carve out a standard
 - Help on embedding the above in OpenAssetIO Traits would be appreciated - helps avoid duplicate standards

TAC Open Discussion

#ASWF

/* ACADEMY SOFTWARE FOUNDATION