



Aerie — Collaborative and modular mission planning and scheduling

- Open source and zero cost
- Cloud native enabling distributed collaboration at scale
- User-friendly UI and low-code constraint authoring

- Automated and manual workflows
- Complete solution from plan creation to command generation
- Leverages modern, quality open source libraries

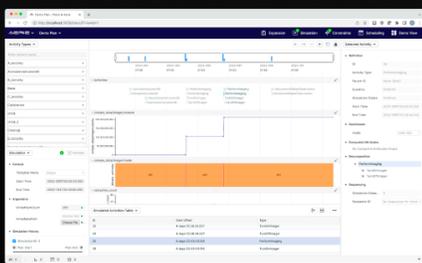
Github
github.com/NASA-AMMOS/aerie
 Request a demo
eric.w.ferguson@jpl.nasa.gov
 Get support on Slack
[NASA-AMMOS #aerie-users](#)

Who is it for?

- Systems Engineers**
Analyze and inform spacecraft design and architecture during development.
- Mission Planners**
Create a viable plan that meets mission objectives while adhering to project and spacecraft constraints throughout the project lifecycle.
- Spacecraft Operators**
Build an integrated plan and set of commands that meets plan goals and constraints.

Aerie UI

Aerie offers a user-friendly interface that incorporates data visualization, modular timelines, and customizable views, facilitating the creation and analysis of plans. The adaptable layout accommodates diverse tasks performed by varied user roles. The UI uses modern web technologies like subscriptions to enable collaboration.

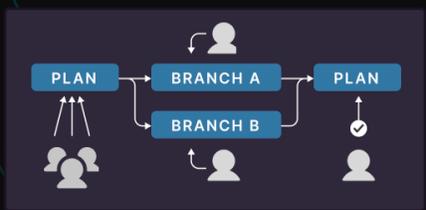


Aerie API

A GraphQL API enables clients to request precise data, enhancing efficiency with fewer requests and reduced bandwidth. All Aerie components utilize this API, facilitating integration with external tools and the Aerie command line interface. The open-source Hasura engine streamlines API maintenance by building it directly from Aerie's data model.

Collaborative Planning

Aerie supports multiple modes of collaboration that enable highly distributed planning and allow it to be tailored to a mission's operations planning process. Collaborate on one plan in real time, or make branches, edit them asynchronously, and merge them when you're ready. Contributors can edit freely or define a system with permissions for certain roles.



Constraint Checking

Planners can author constraints, evaluate them against simulation data, and visualize violations on Aerie's plan timeline. Constraints are defined within an expression language embedded in TypeScript and can describe complex relationships between activities and resources. Constraints can be evaluated against external data pushed into Aerie's PostgreSQL database via Aerie's API.

```

1 Real.Resource('/rate/a')
2   .plus(Real.Resource('/rate/b'))
3   .lessThan(10)
    
```

Aerie Extended Universe

The Aerie project has begun to build a marketplace of extensions that missions can use to jump start their modeling and integration efforts. New extensions are provided by core Aerie collaborators, missions, and other members of the community.

Automated Scheduling

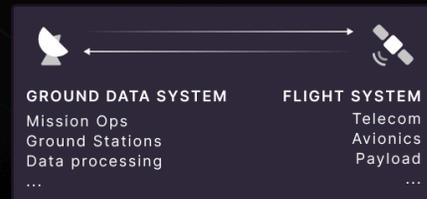
Aerie incorporates a scheduling system that auto-assigns activities in a plan based on declarative goals set by planners. Planners use TypeScript to define three goal types: Recurrence Goals for scheduling activities at a specific cadence, Coexistence Goals for coordinating activities with given resource conditions or other activities, and Composite Goals, which AND/OR the previous goal types for concurrent processing.

Sequence Editing and Expansion

Sequence engineers can author and edit sequences using the open source Monaco editor (which powers VS Code). This provides out of the box Intellisense, and is fully compatible with the open source seq-json sequence specification. Operators can link activities to corresponding sequences and commands for spacecraft execution. Expansion logic is defined using a TypeScript-embedded Domain Specific Language (eDSL).

Mission Model Framework

Aerie offers missions a Java-based framework to define system behavior. Modelers can access the Java ecosystem, including editors, libraries, and tools, to create a JAR file compatible with Aerie's interface. Users can then simulate plan effects using Aerie's discrete event simulation engine through its UI or API.



Aerie aims to provide configurable multi-mission models for spacecraft and ground systems, streamlining mission model development for various projects.