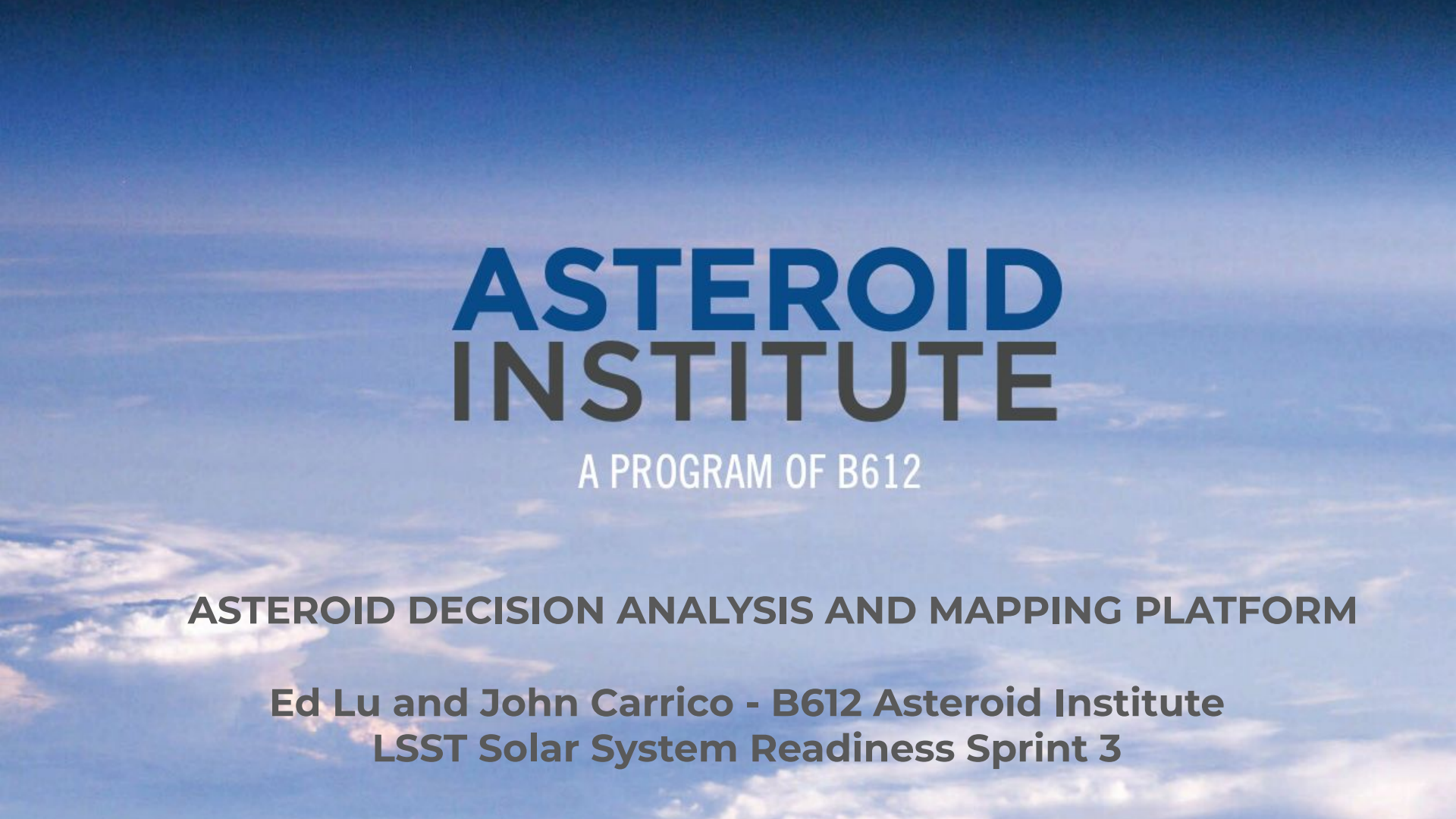


ASTEROID DECISION ANALYSIS AND MAPPING PLATFORM

Ed Lu and John Carrico - B612 Asteroid Institute

LSST Solar System Readiness Sprint 3

Goal: Scaleable Astrodynamics Compute Engine

ASTEROID INSTITUTE
A PROGRAM OF B612

- Astrodynamics functions that just work, at scale
- Orbit propagation, uncertainty propagation (covariance, Monte-Carlo), State Vector targeting, orbit determination
- Interface Python scripts via API or web interface
- Open and well documented APIs - strong Python support
- Handles various coordinate systems
- Architected to run at scale
    - Cloud based - running on Google Cloud using Docker
    - System manages batch jobs, spooling up of CPUs etc.

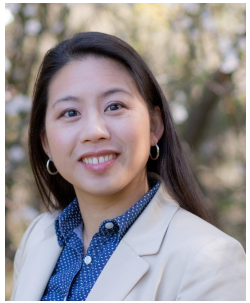
Asteroid Institute

ASTEROID INSTITUTE
A PROGRAM OF B612

Dr. Ed Lu, Asteroid Institute Executive Director

John Carrico, ADAM Program Manager & Institute Astrodynamicist

Carise Fernandez, ADAM Senior Software Engineer



Dr. Mario Juric, Associate Professor of Astronomy, University of Washington

Dr. Sarah Greenstreet, Senior Researcher

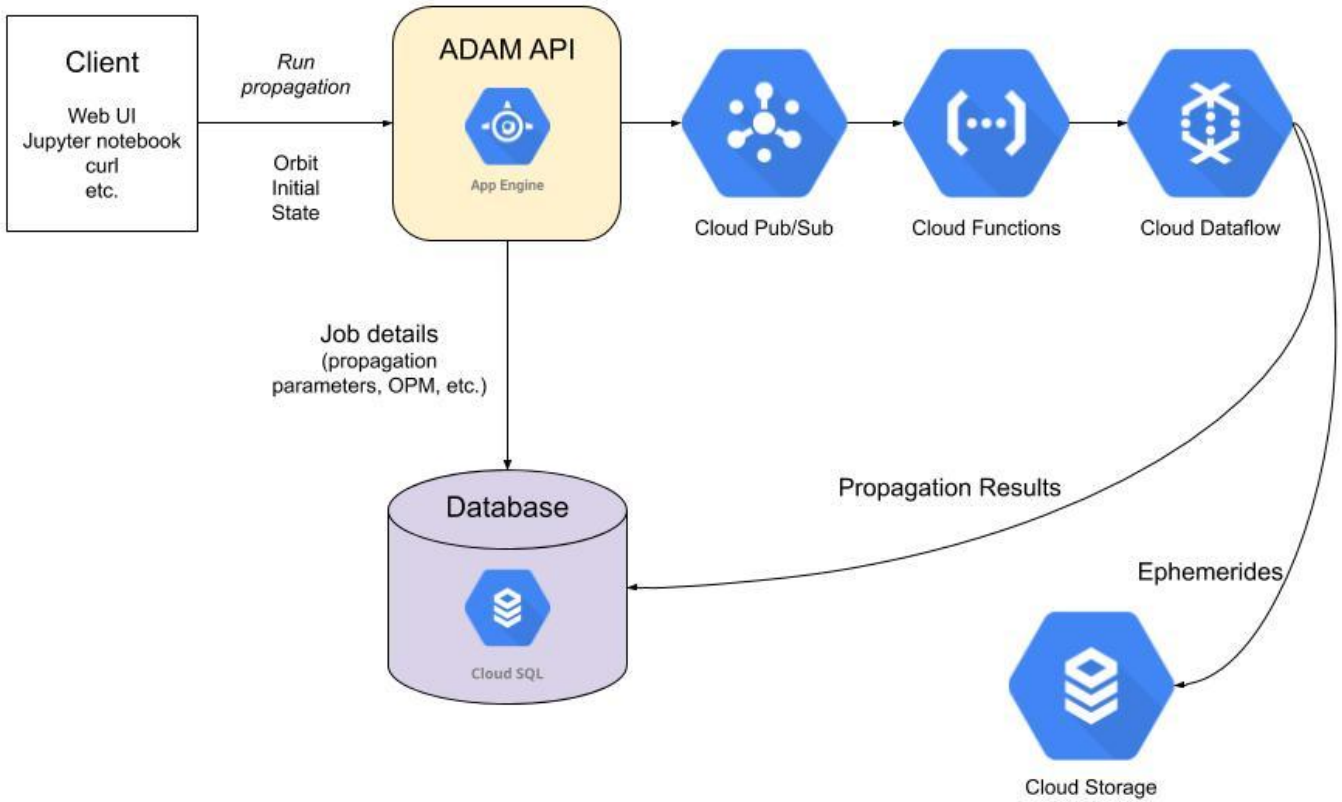
Hank Grabowski, Software and Aerospace Engineer

Joachim Moeyens, Univ. of Washington Postdoctoral Researcher

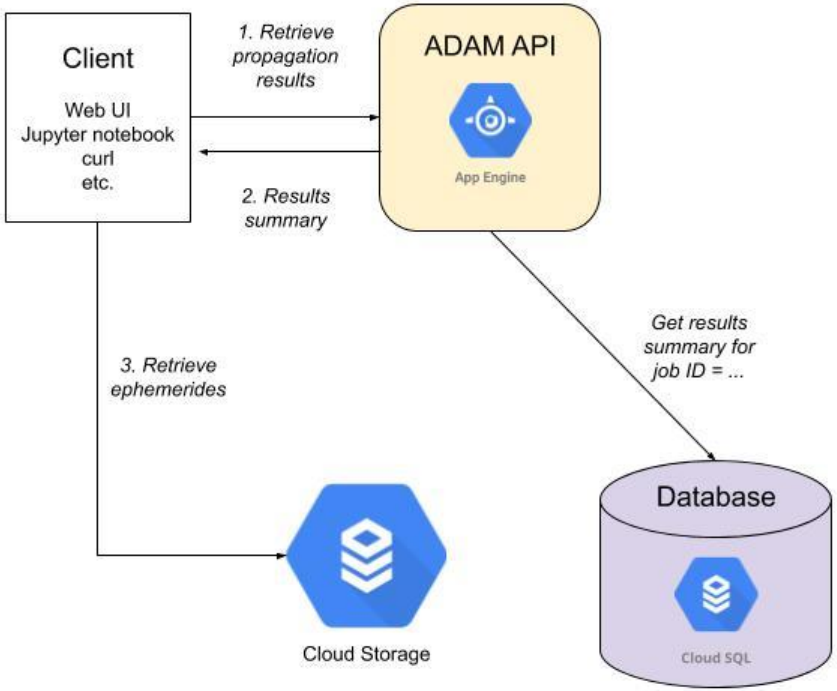
Emmie King

Allan Posner, Systems Engineer and Astrodynamicist

ADAM Job Dataflow



ADAM Data Retrieval



ADAM Project Status

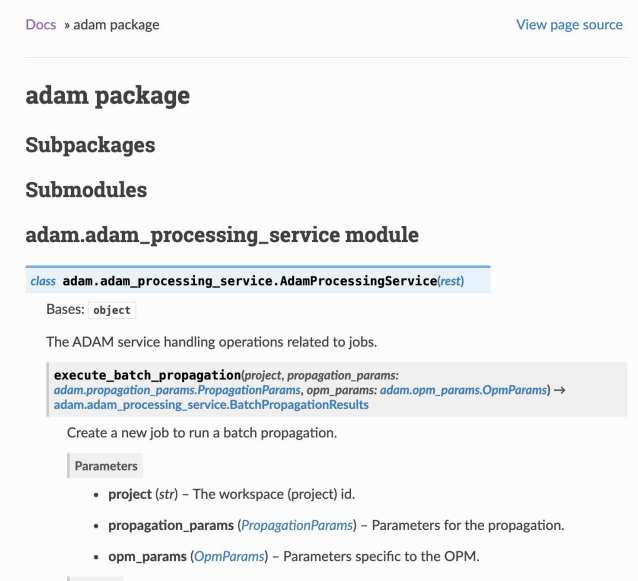
ASTEROID INSTITUTE
A PROGRAM OF B612

- API's running: Propagation (OpenOrb, STK), State Vector targeting (STK)
- API's in work: Orbit Determination (OpenOrb, FindOrb)
- Visualization: SpaceKit open source Javascript
- Future work: Back end infrastructure, Observation linkage at scale

Example - API for Monte-Carlo Propagation

ASTEROID INSTITUTE
A PROGRAM OF B612

- Propagating a state vector with covariance
- Python call to



Docs » adam package View page source

adam package

Subpackages

Submodules

adam.adam_processing_service module

class **adam.adam_processing_service.AdamProcessingService**(*rest*)

Bases: `object`

The ADAM service handling operations related to jobs.

execute_batch_propagation(*project*, *propagation_params: adam.propagation_params.PropagationParams*, *opm_params: adam.opm_params.OpmParams*) → adam.adam_processing_service.BatchPropagationResults

Create a new job to run a batch propagation.

Parameters

- **project** (*str*) – The workspace (project) id.
- **propagation_params** (*PropagationParams*) – Parameters for the propagation.
- **opm_params** (*OpmParams*) – Parameters specific to the OPM.

```
batch_run = aps.execute_batch_propagation(config['workspace'], propagation_params, opm_params)
```

Demo

ASTEROID INSTITUTE
A PROGRAM OF B612

ADAM Call for LSST-SSSC Alpha-testers

ASTEROID INSTITUTE
A PROGRAM OF B612

- Looking for use-case feedback
- Initial chance to “kick the tires and see what ADAM can do”
- Expectation: users will be familiar with Jupyter notebooks - we will provide a sample notebook to start with
- Goal: Hack together a demo. Provide use-case feedback.

FAQ

ASTEROID INSTITUTE
A PROGRAM OF B612

1) How can I use it? Call for alpha testers - use case feedback.
2) Who pays for it? Currently 100% from donations to B612 Foundation.
3) When can I use it? See call for alpha testers. Plan is to begin opening up on a limited basis later this year.
4) Open source? Mix of open source and commercial algorithms.
5) Running on private cluster? Eventually, not currently.