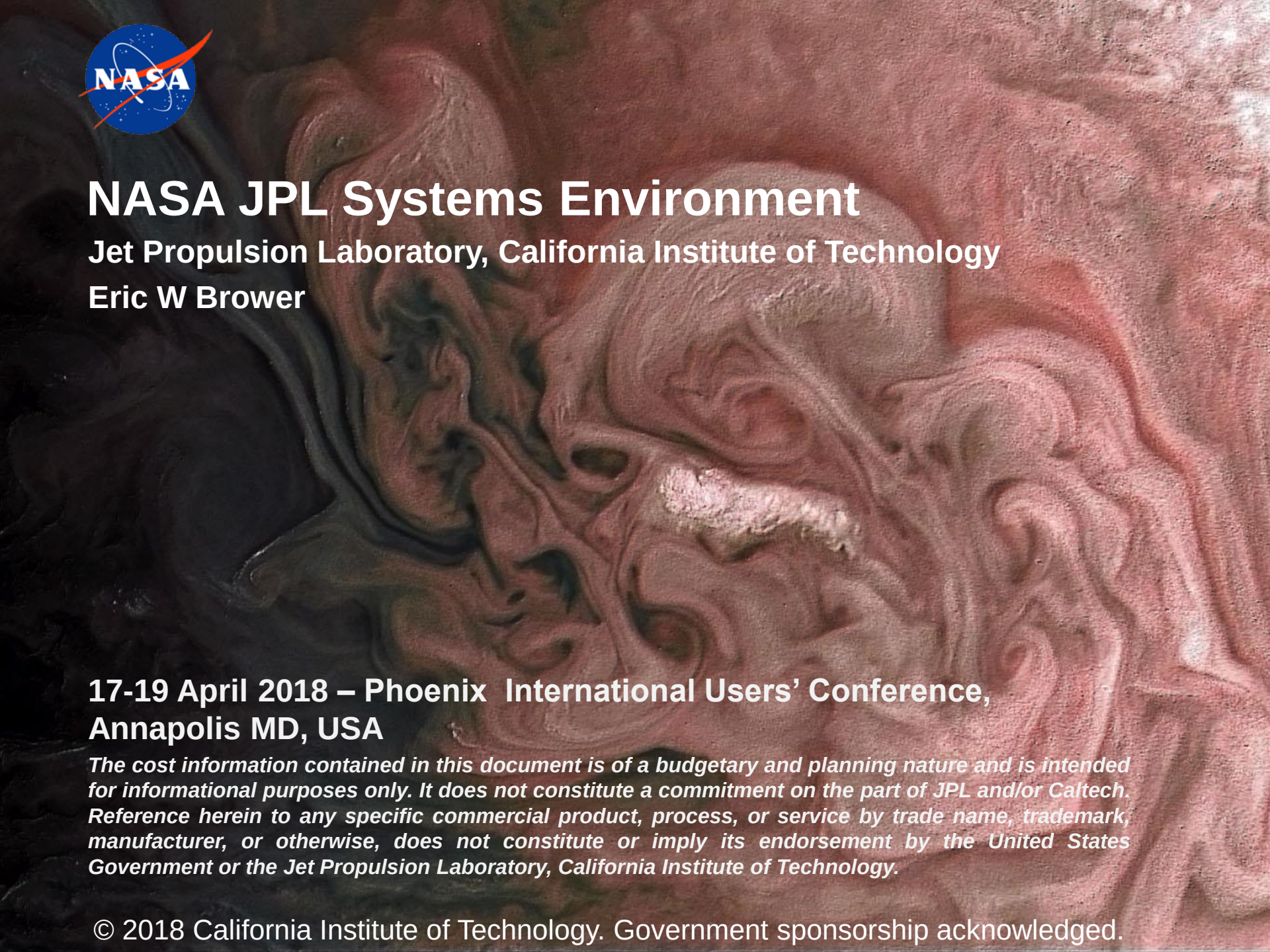


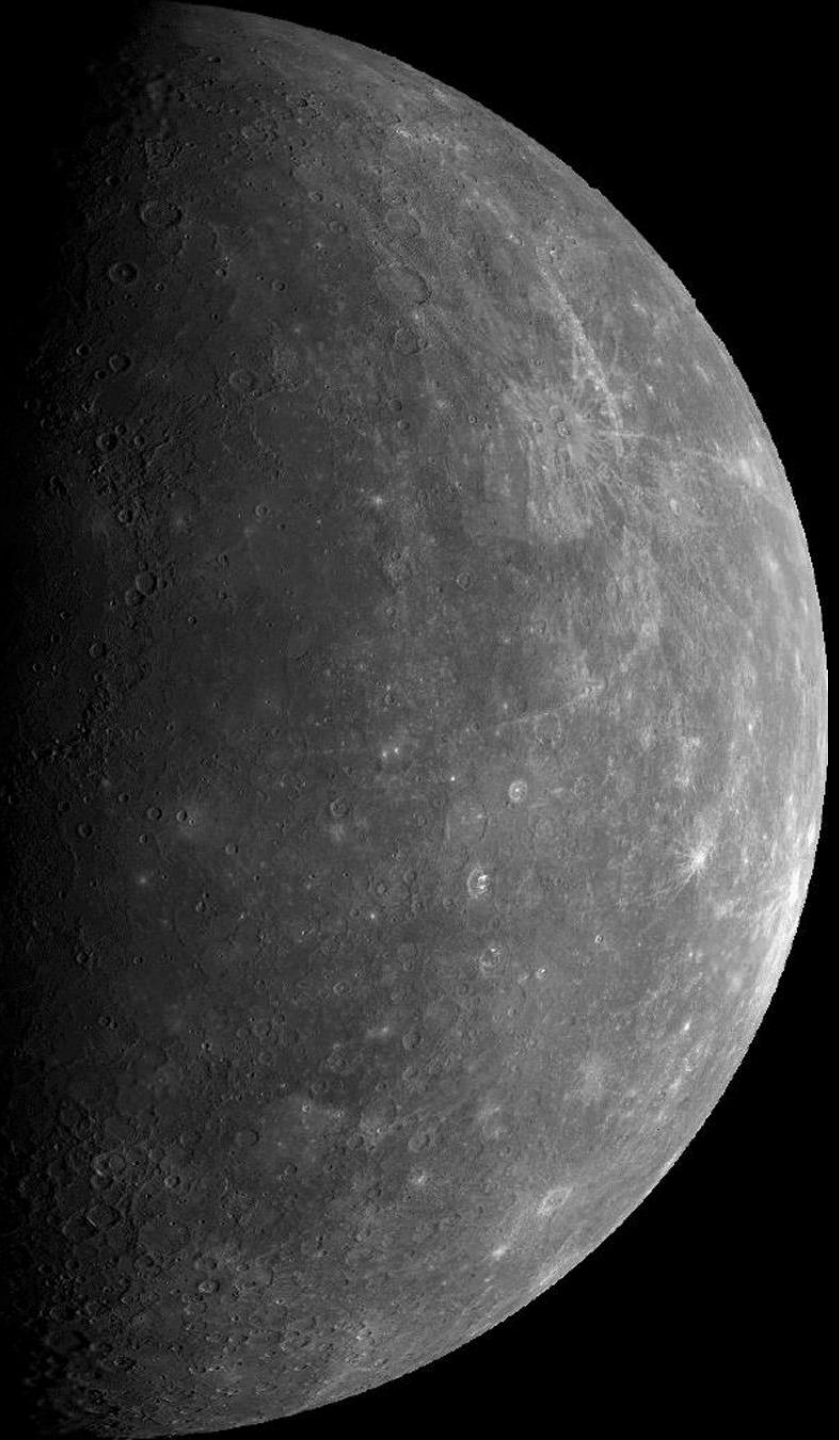
# NASA JPL Systems Environment


**Jet Propulsion Laboratory, California Institute of Technology**

**Eric W Brower**


**17-19 April 2018 – Phoenix International Users' Conference, Annapolis MD, USA**

***The cost information contained in this document is of a budgetary and planning nature and is intended for informational purposes only. It does not constitute a commitment on the part of JPL and/or Caltech. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.***

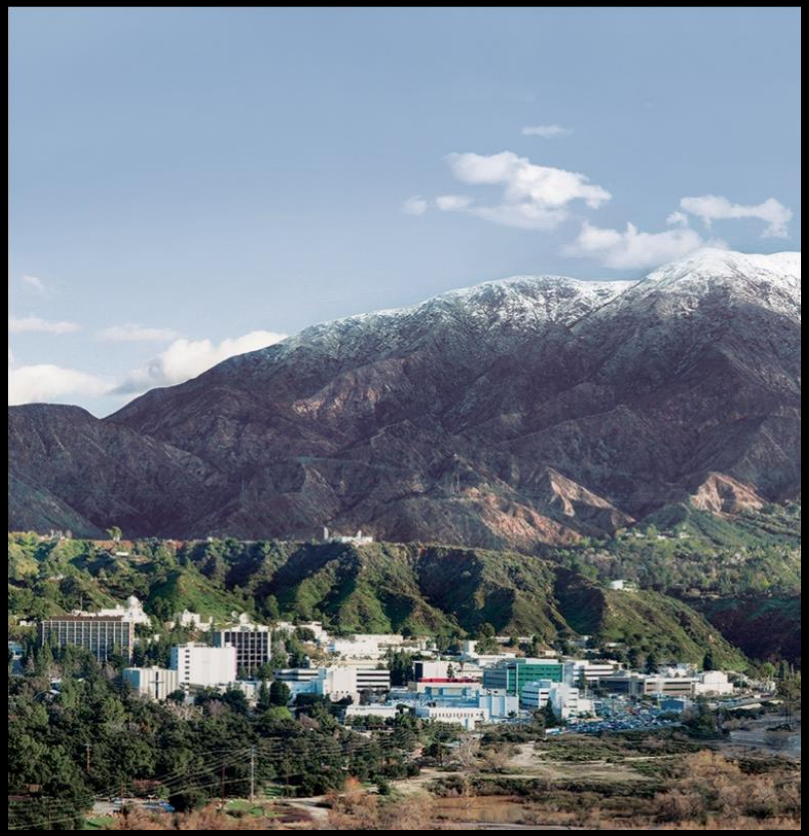


# Agenda

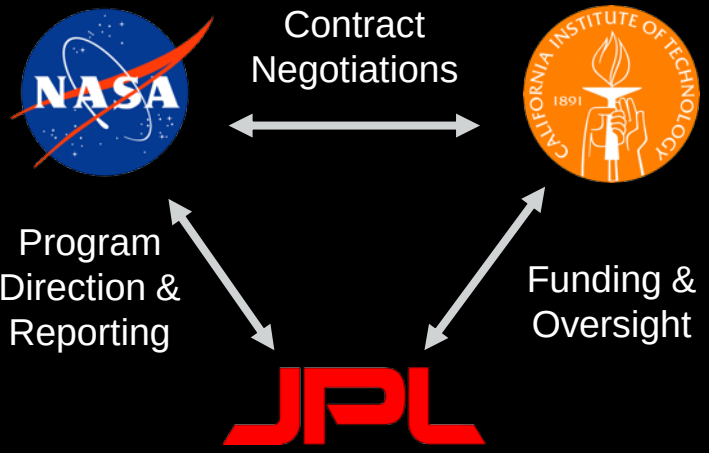
- Introduction
- OpenCAE Approach
- Open Source Contributions
- Questions

Jet Propulsion Laboratory
California Institute of Technology

# NASA Jet Propulsion Laboratory (JPL)



- Located in Pasadena, CA
- NASA-owned *"Federally-Funded Research and Development Center"*
- University-operated
- ~5,000 employees

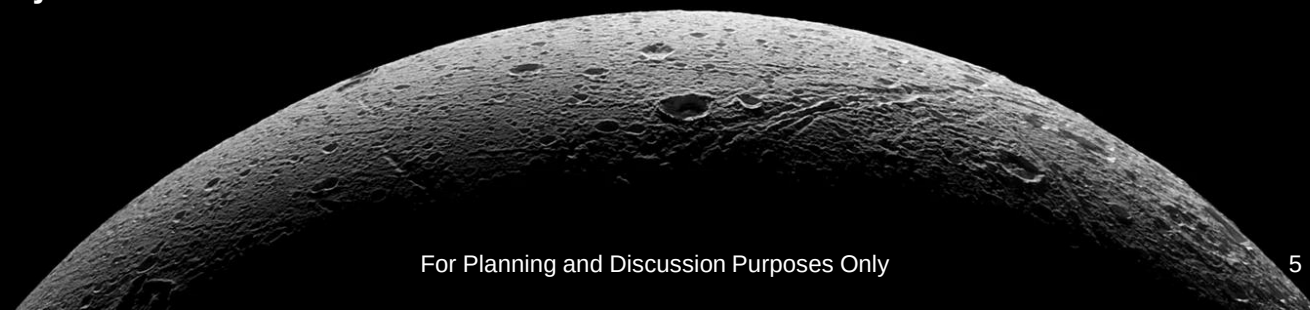


# Computer Aided Engineering (CAE)

- Computer Aided Engineering provides the Laboratory's Engineering Staff and Scientific communities with tools and technical expertise
- Four Environments:
  - Systems Environment
  - Software Environment
  - Mechanical Environment
  - Electrical Environment

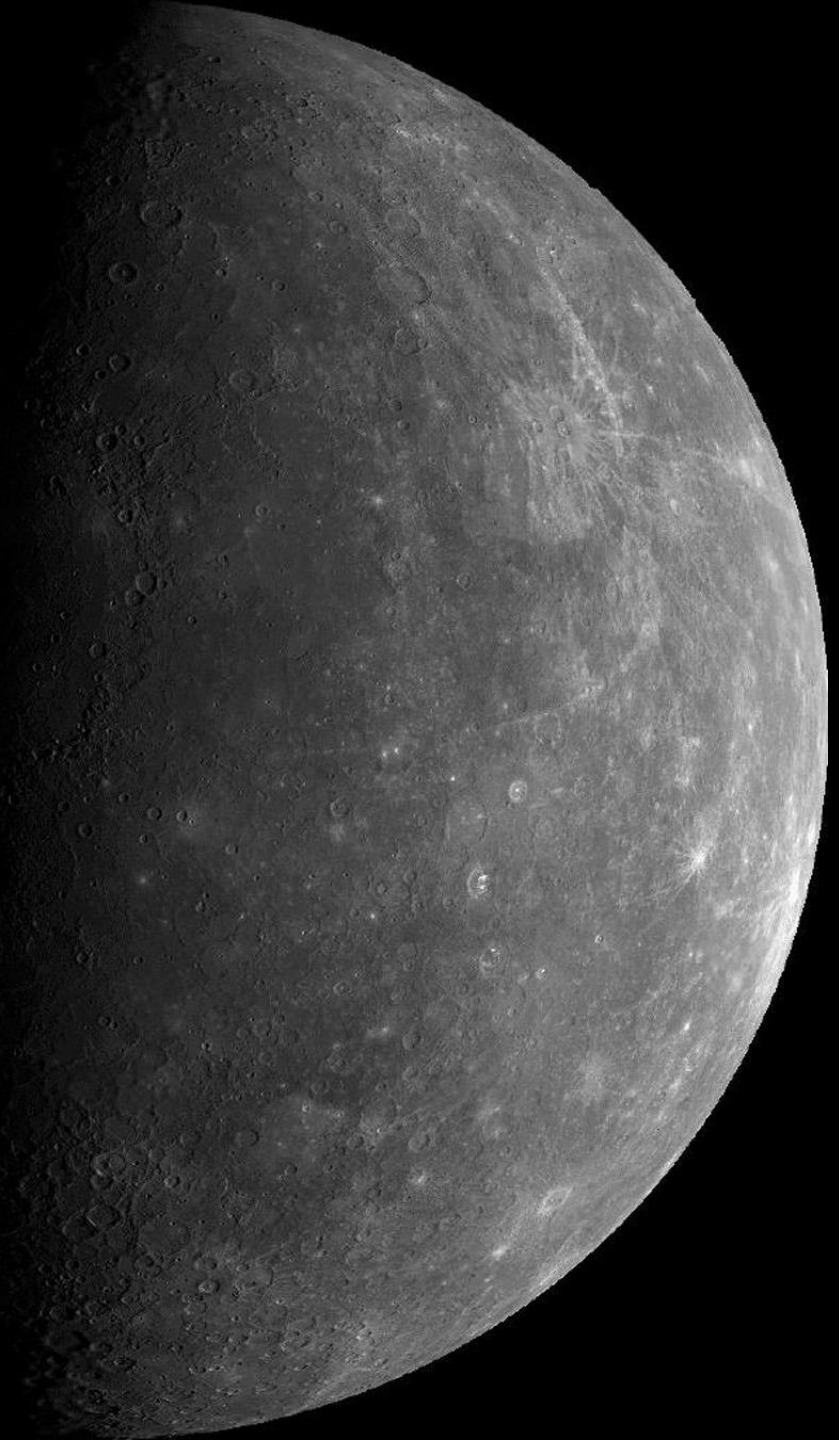
# OpenCAE Vision

- Provide an open portfolio in a shared environment that seamlessly connects engineers developing missions and systems.
  - Open - The portfolio that CAE provides is open in every sense of the sprit of open source. Our processes, code, apps, services and artifacts are accessible by JPL users as well as vendors and partners.
  - Shared - CAE is more than a collection of licenses and tools, its a shared environment for engineering. The diverse community of users, developers partners and vendors are able to contribute innovation and work more effectively by reducing the overhead.
  - Connected - the CAE Environment connects engineers allowing them to collaboratively construct and analyze the precision products needed to develop Missions and Systems at JPL using the CAE environment. This is done without the overhead of traditional manual exchanges of information. Engineers can connect with each other and find relevant engineering data and information reducing redundancy and increasing value of the engineering products and analysis produced by the flight project.



# OpenCAE Mission

- Develop the CAE environment from a user centered architecture leveraging vendor partnerships using robust life cycle processes.
  - Vendor partnerships – CAE works closely with Vendors providing them crucial feedback and insight into how their products are serving the needs of engineers and developers
  - User centered architecture – to achieve the vision of Open CAE, the technical architecture for CAE is driven by the needs of the practitioners who use the environment and the needs of the projects that are served by it
  - Life-cycle process – the life-cycle processes for CAE provide the integrity of the the applications services and support provided by CAE



# Agenda

- Introduction
- OpenCAE Approach
- Open Source Contributions
- Questions

Jet Propulsion Laboratory
California Institute of Technology

# Applications of Model-Based Engineering at JPL

JPL is applying MBE practice in several projects

- Missions to Europa
  - Europa Clipper
  - Europa Lander
- Missions to Mars
  - Mars 2020
  - InSight
  - Mars Sample Return (MSR)
- Thirty Meter Telescope
- Ground Data Systems
- Psyche
- MAIA

Engineering Products

- MELs, PELs
- Resource allocation analysis
- System decomposition,
- Libraries / reusable models

Not just spacecraft missions! Not just early phases of design!

# OpenCAE: Realization

- A platform for engineering tools to work together
- Incorporate tooling from systems, software, mechanical, and electrical domains
- Platform integrates heterogenous data sources
- Emphasize standards for data interchange
- Case studies inform the architecture of the engineering environments

# OpenCAE: User Centered Design

- User Centered Design steers the development of the OpenCAE infrastructure
- Continuous communication with users to understand their experience in the OpenCAE environments
- Users evaluate solutions before they are implemented
- Following standard UX practices

# OpenCAE: User Communication

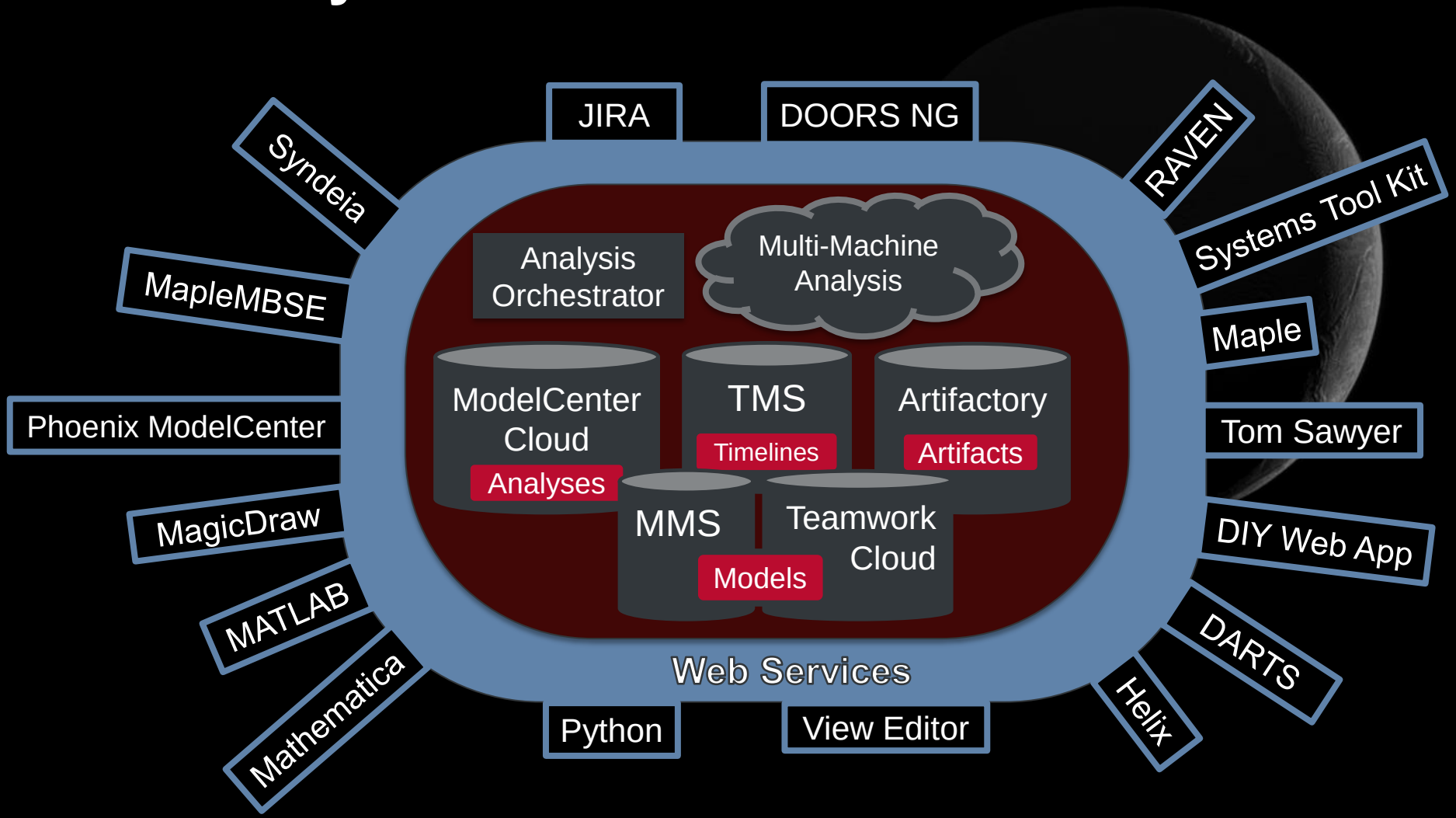
- Mailing lists generated by tool license use
- Slack channels per each tool for general questions (with vendors)
- Technical Working Groups held biweekly with vendors for tool-specific questions
- OpenCAE Systems Environment Team Office Hours held biweekly for general questions and support

# CAE Systems Environment

## Support SE Activities:

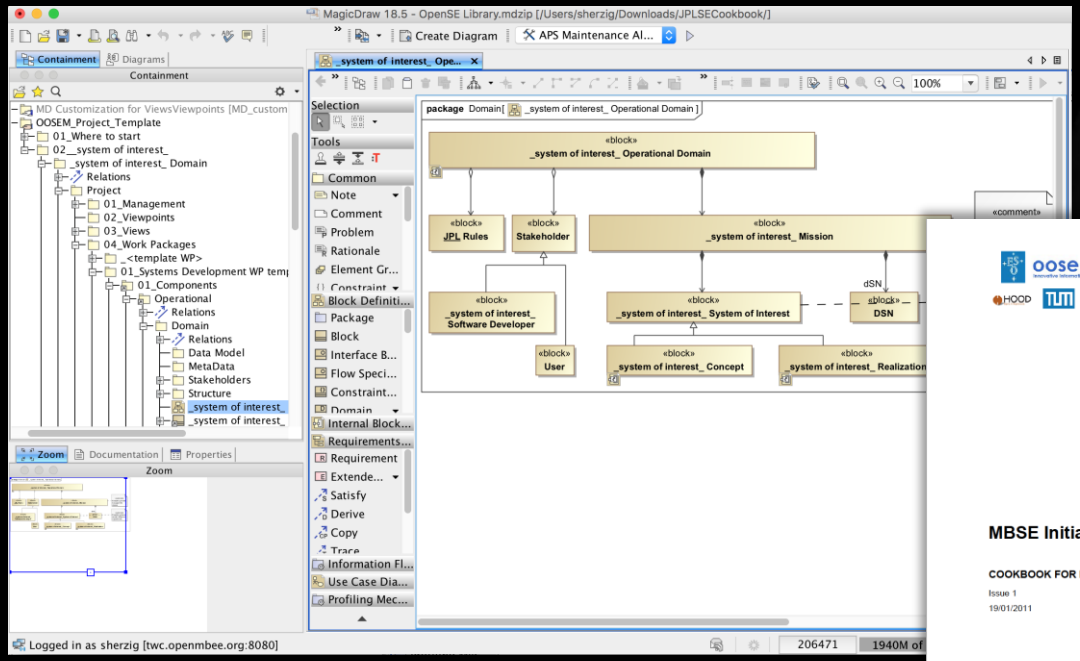
- Requirements Management
- Interface Management
- Design Management
- Trade Studies
- Interdisciplinary Integration
- Analysis Management
- Resource Management

# CAE Systems Environment: Overview

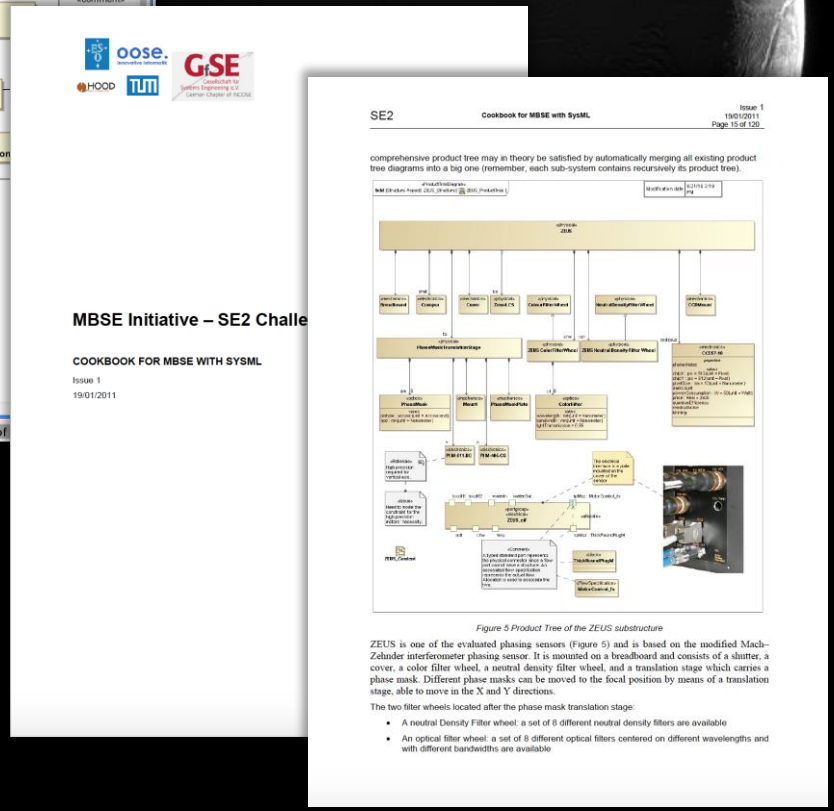



OpenCAE DevOps

# CAE Systems Environment: Cookbook and Template Model




**“Cookbook” for modeling methodology and patterns**




**Template models to be used by projects as a starting point, with recommended organization, model libraries, etc.**

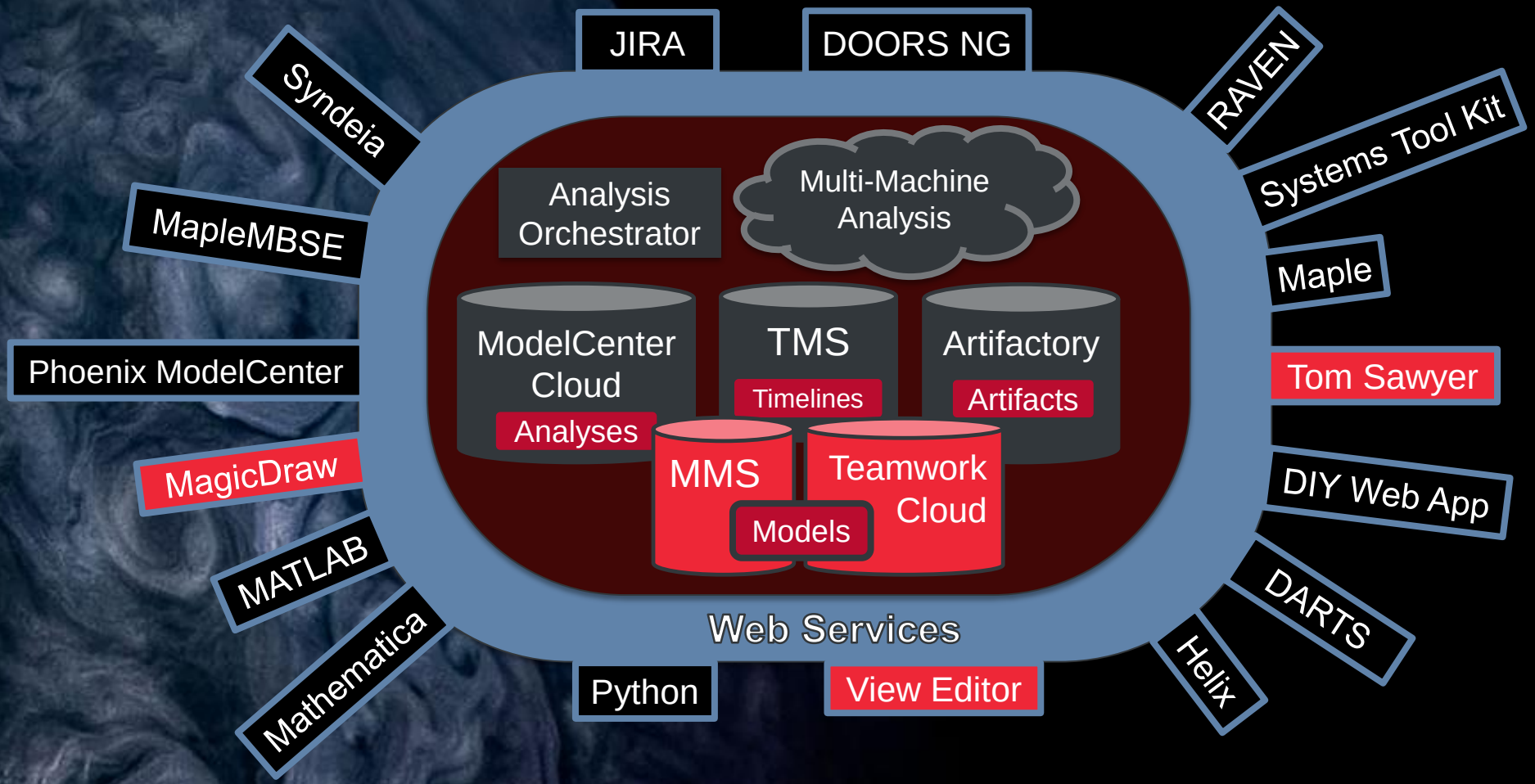
# CAE Embedded Roles

- CAE provides the same environment to all its customers (engineers and scientists)
- Embedded roles work directly on projects to adapt the standard environment specific to the project goals or methodology
- Embedded roles capture needs in general case studies which inform the CAE architecture

# Europa Lander Embedded Role

- Need:
  - Generate orderly and palatable diagrams from a system model describing the Lander
  - SE products should never be out of sync with the system model
- Approach:
  - Leverage Tom Sawyer plugin for MagicDraw development effort
  - Supply requirements directly from the project to the vendor
  - Coordinate with CAE development team on the use case for Tom Sawyer integration with DocGen and View Editor

# CAE Systems Environment

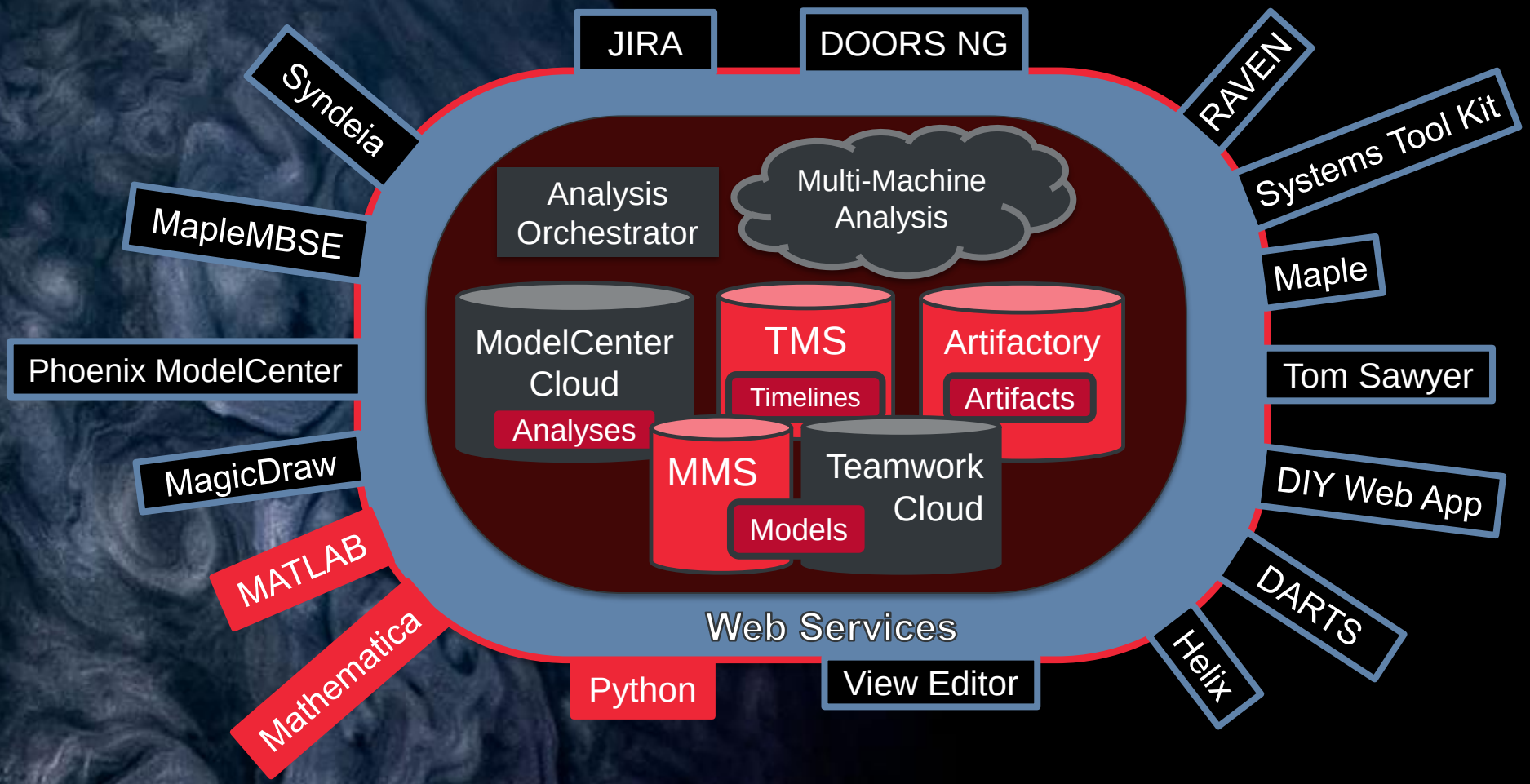



OpenCAE DevOps

# Europa Clipper Embedded Role

- Need:
  - Need to publish artifacts to CAE services (MMS, TES, Artifactory)
- Approach:
  - Express the REST API endpoints of these servers in OpenAPI standard specification
  - Use Swagger codegen to generate clients for specific analysis environments
    - Mathematica, MATLAB, Python, Java
    - More than 20 other languages available

# CAE Systems Environment

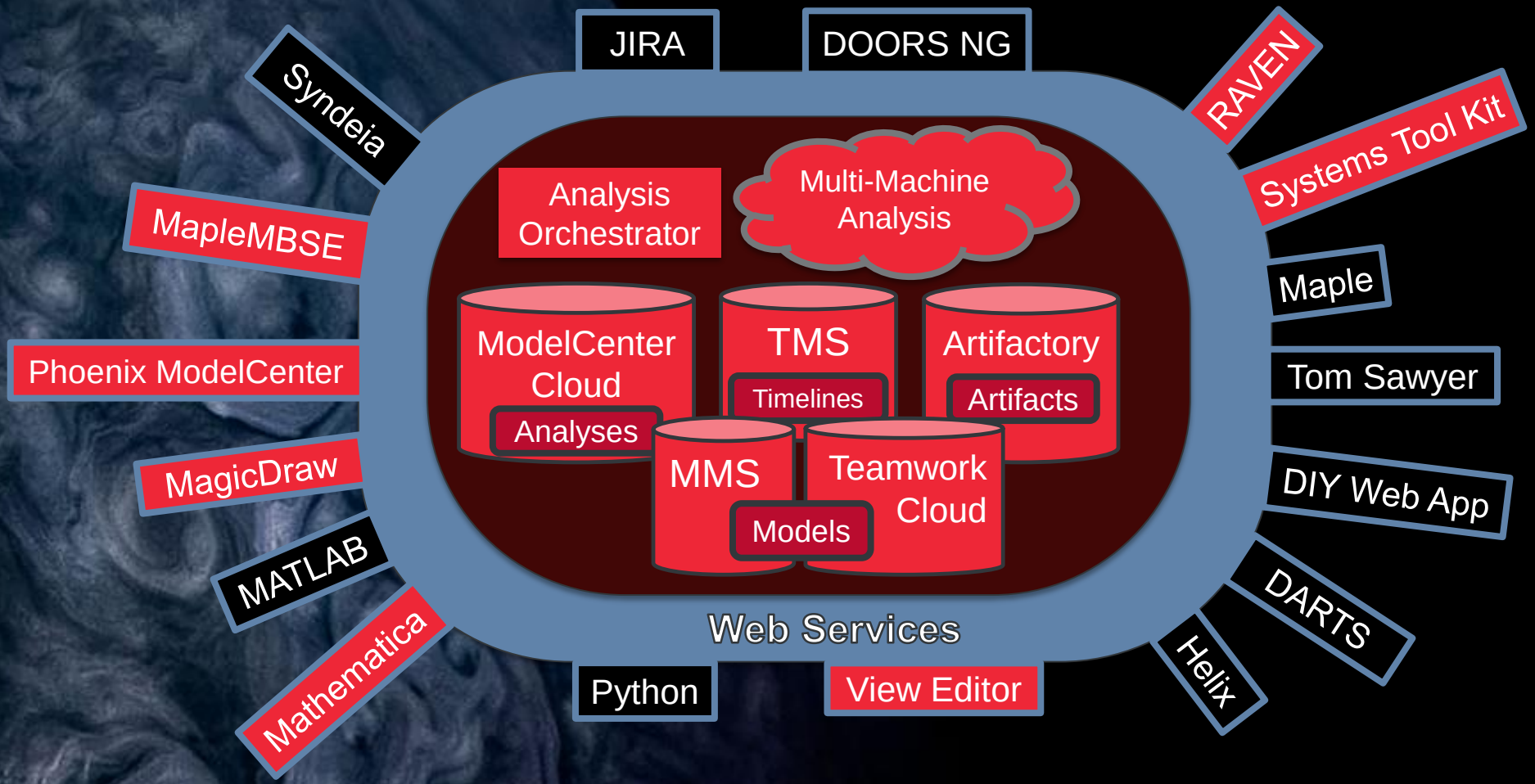



OpenCAE DevOps

# Europa Clipper Embedded Role

- Need:
  - Formalize analysis workflows related to the Clipper Flight System
  - Want to capture the workflows in a model, but also want them to be executable
- Approach:
  - Use Phoenix MBSEPak plugin for MagicDraw to translate the workflow parameters into Phoenix ModelCenter
  - Configure ModelCenter to use shared components in the Analysis Library of ModelCenter Cloud

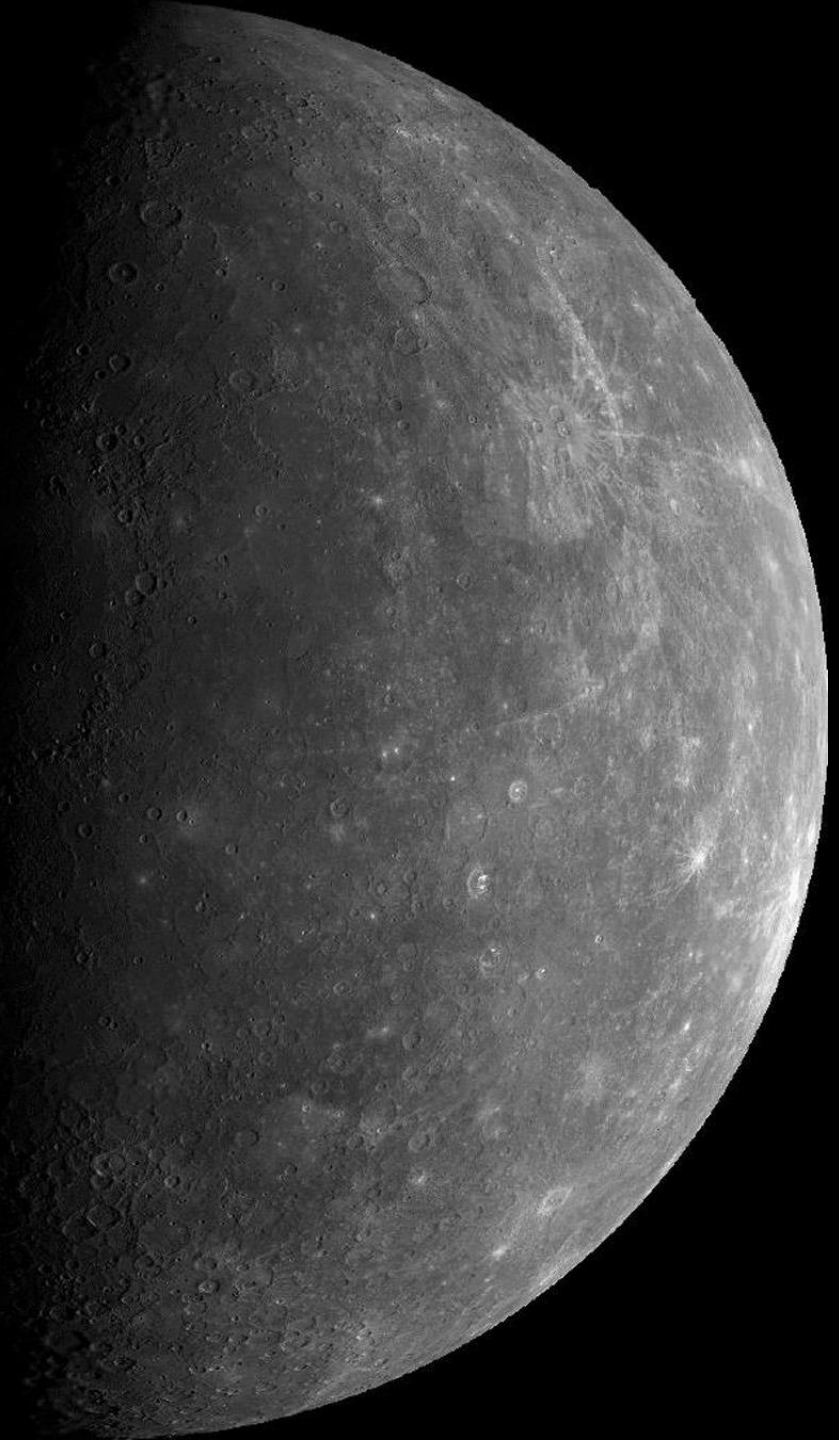
# CAE Systems Environment




OpenCAE DevOps

# OpenCAE: What Has Worked

- Domain specific implementations
  - Managed Services with vendors
  - Embedded roles with JPLers
- Server-side operations preferred
  - Easier to update a server than many clients
  - COTS connections between services
- Speak the same language (SysML, FMI)
- OpenAPI REST specification
  - Generate clients for users' preferred languages
  - Enforces OpenAPI on environment services

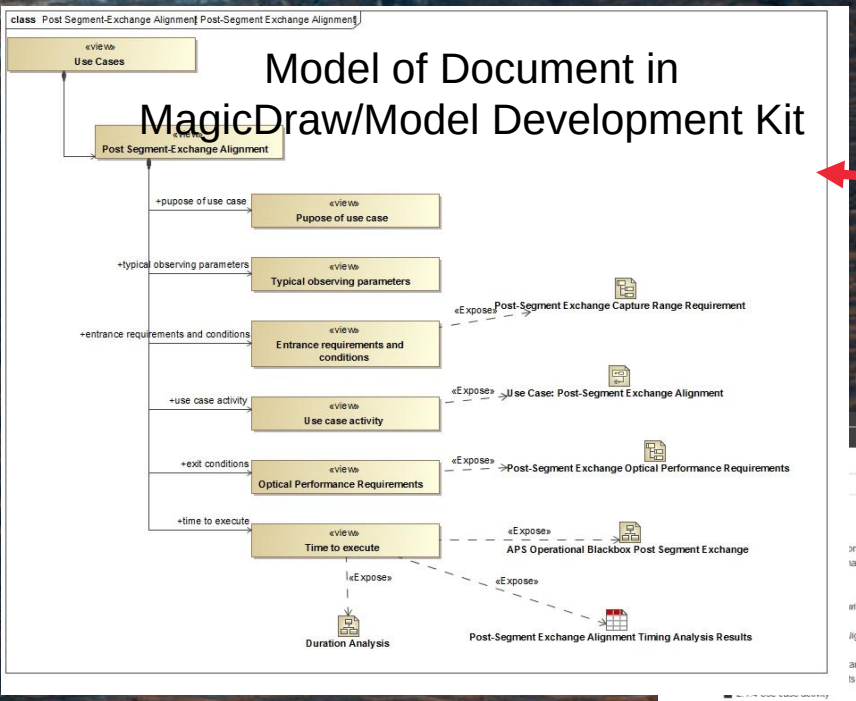


# Agenda

- Introduction
- OpenCAE Approach
- Open Source Contributions
- Questions

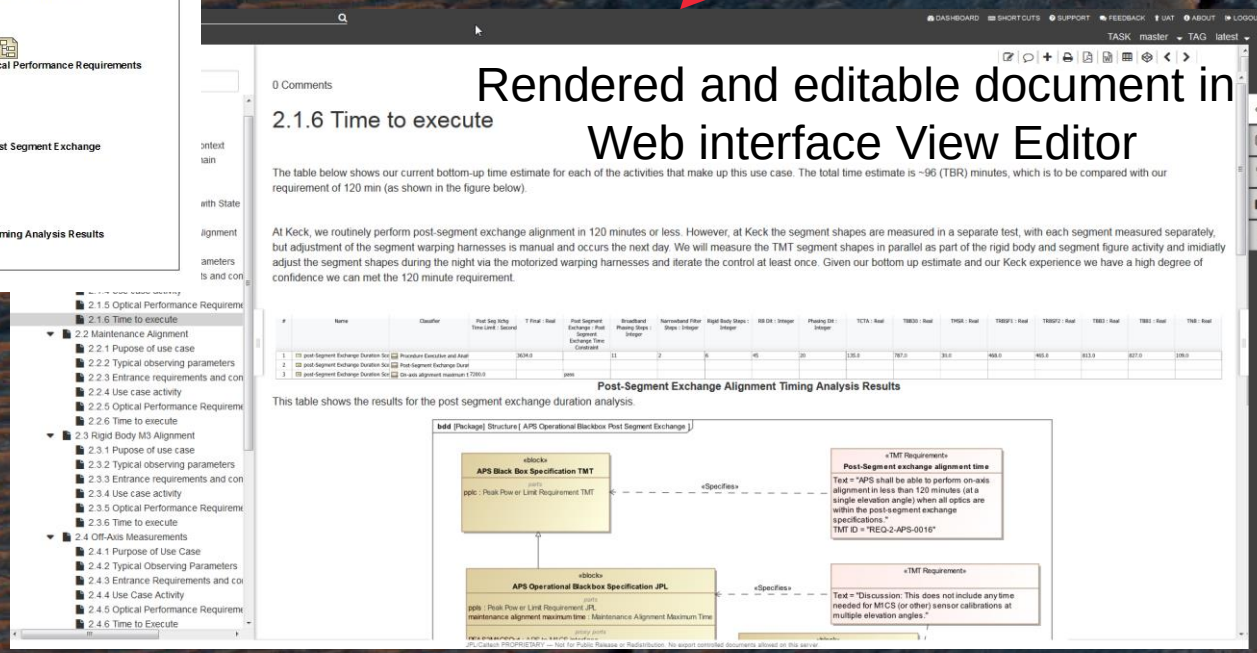
Jet Propulsion Laboratory
California Institute of Technology

# Core Integration of MMS, MDK, and VE




Model of Document in MagicDraw/Model Development Kit

Model Repository




Rendered and editable document in Web interface View Editor

# openmbee.org

## Open Model Based Engineering Environment

- OpenMBEE is a community for open-source modeling software and models
  - Number of open source software activities
  - Number of open source models
- JPL is a participant and adopter of OpenMBEE software and models

# Acknowledgements

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The TMT Project gratefully acknowledges the support of the TMT collaborating institutions. They are the Association of Canadian Universities for Research in Astronomy (ACURA), the California Institute of Technology, the University of California, the National Astronomical Observatory of Japan, the National Astronomical Observatories of China and their consortium partners, and the Department of Science and Technology of India and their supported institutes. This work was supported as well by the Gordon and Betty Moore Foundation, the Canada Foundation for Innovation, the Ontario Ministry of Research and Innovation, the National Research Council of Canada, the Natural Sciences and Engineering Research Council of Canada, the British Columbia Knowledge Development Fund, the Association of Universities for Research in Astronomy (AURA) and the U.S. National Science Foundation.

# References


- Karban, R., Jankevičius, N., Elaasar, M. “ESEM: Automated Systems Analysis using Executable SysML Modeling Patterns”, (to appear in the proceedings of INCOSE International Symposium (IS), Edinburgh, Scotland, 2016.)
- Karban R., Dekens F., Herzig S., Elaasar M, Jankevičius N., “Creating systems engineering products with executable models in a model-based engineering environment”, SPIE, Edinburgh, Scotland, 2016
- Karban, R., “Using Executable SysML Models to Generate Systems Engineering Products”, NoMagic World Symposium, Allen, TX, 2016
- Open Source TMT model: https://github.com/Open-MBEE/TMT-SysML-Model
- Open Source Engineering Environment: https://open-mbee.github.io/
- Docgen, View&ViewPoints: https://github.com/Open-MBEE/mdk/tree/mdk-manual/src/main/dist/manual
- JPL Model-Based Systems Engineering Case Study: http://omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2017:iw_2017_open_mbee.pdf
- A Practical Guide to SysML, 3rd Edition, Chapter 17 by Friedenthal, Moore, and Steiner
- Zwemer, D., “Connecting SysML with PLM/ALM, CAD, Simulation, Requirements, and Project Management Tools”, May 2016
- https://www.jpl.nasa.gov/spaceimages/


**Jet Propulsion Laboratory**
California Institute of Technology

jpl.nasa.gov