NASA JPL Systems Environment

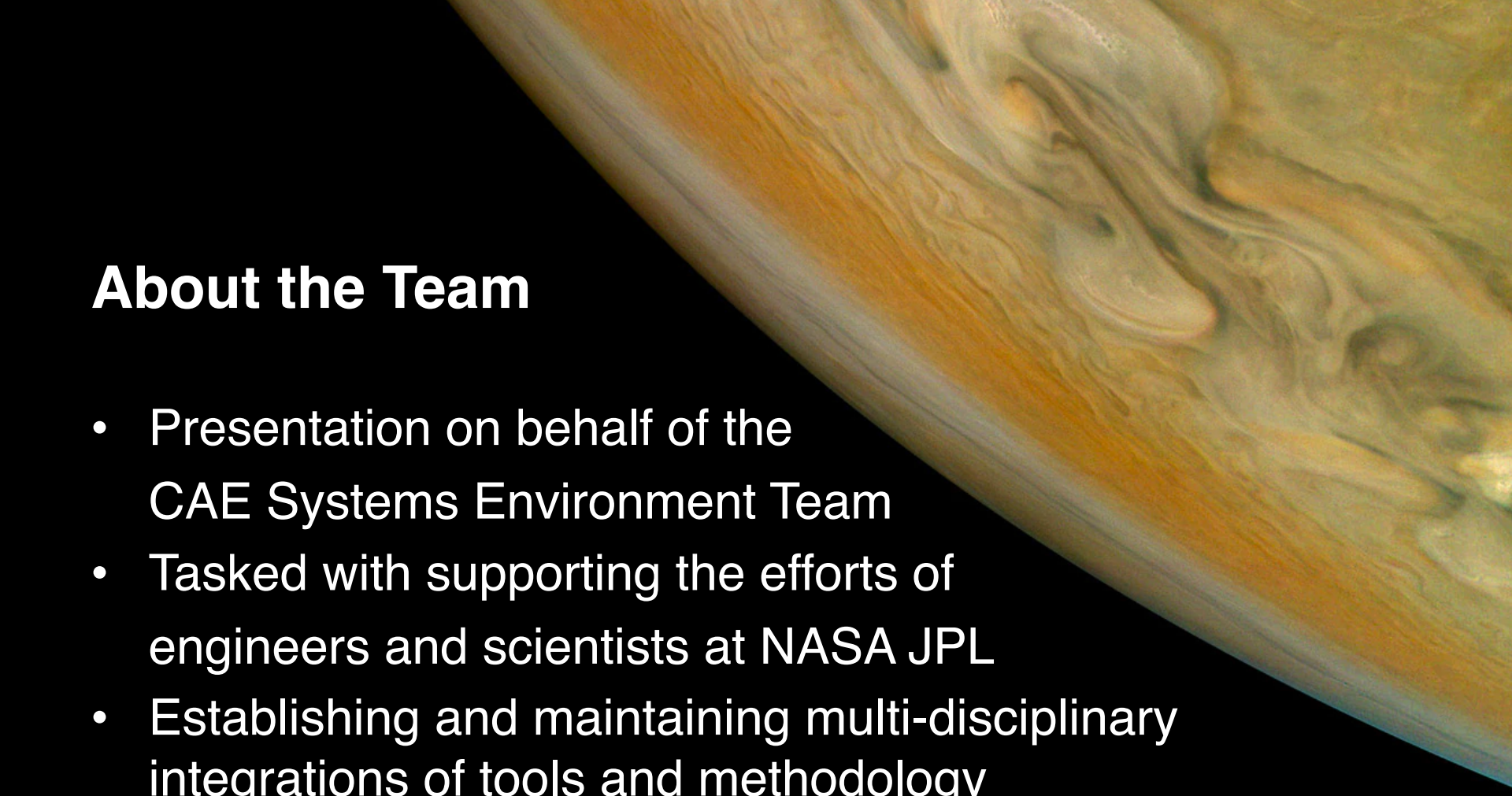
Robert Karban, *CAE Project Systems Engineer*
Jet Propulsion Laboratory,
California Institute of Technology

December 2018 – OMG Technical Meeting – Seattle, WA, USA

Agenda

- Introduction
- OpenCAE Approach
- Open Source Contributions
- Questions

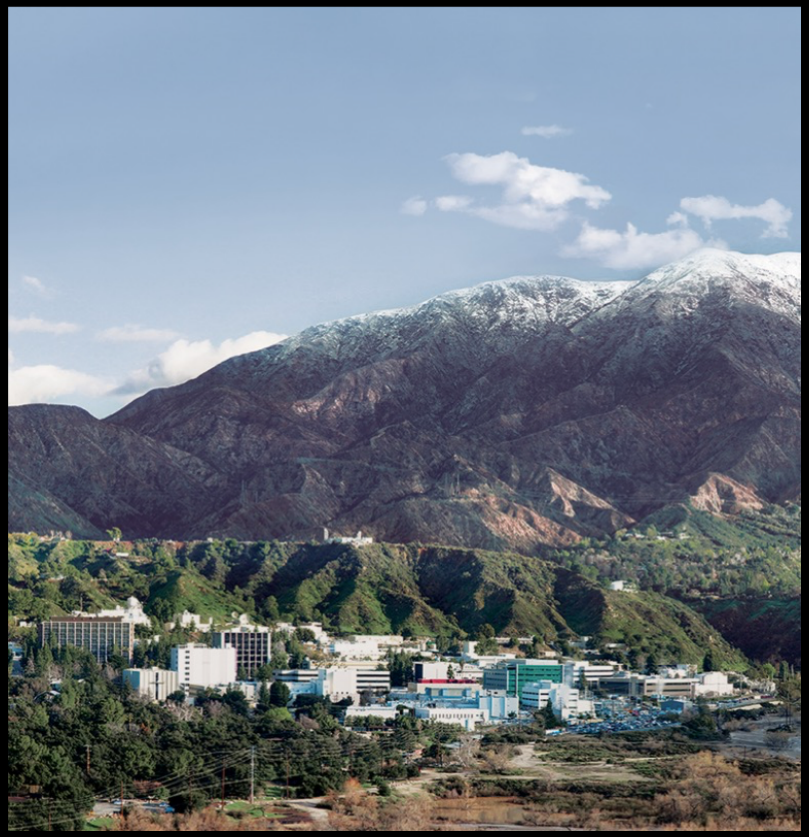
Jet Propulsion Laboratory
California Institute of Technology



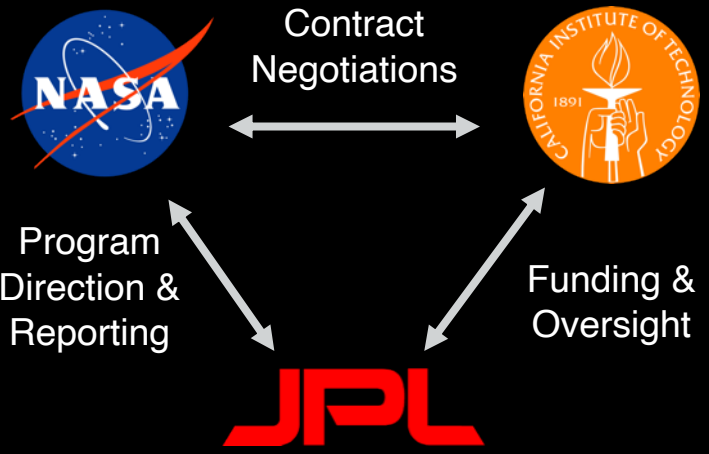
About the Team

- Presentation on behalf of the CAE Systems Environment Team
- Tasked with supporting the efforts of engineers and scientists at NASA JPL
- Establishing and maintaining multi-disciplinary integrations of tools and methodology

NASA Jet Propulsion Laboratory (JPL)

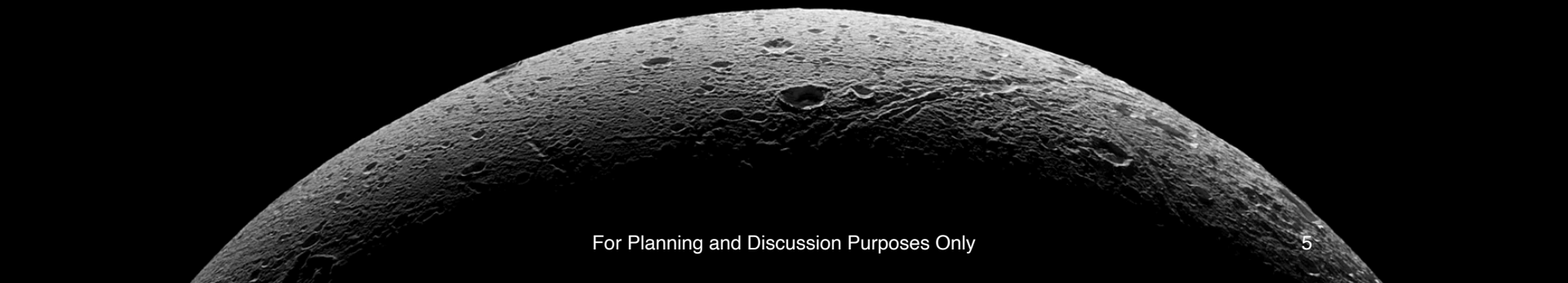


- Located in Pasadena, CA
- NASA-owned *"Federally-Funded Research and Development Center"*
- University-operated
- ~5,000 employees



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

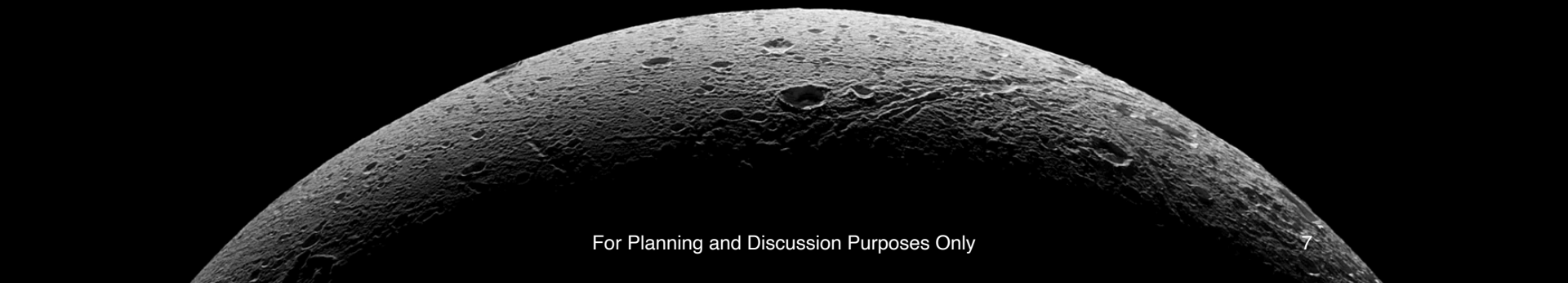


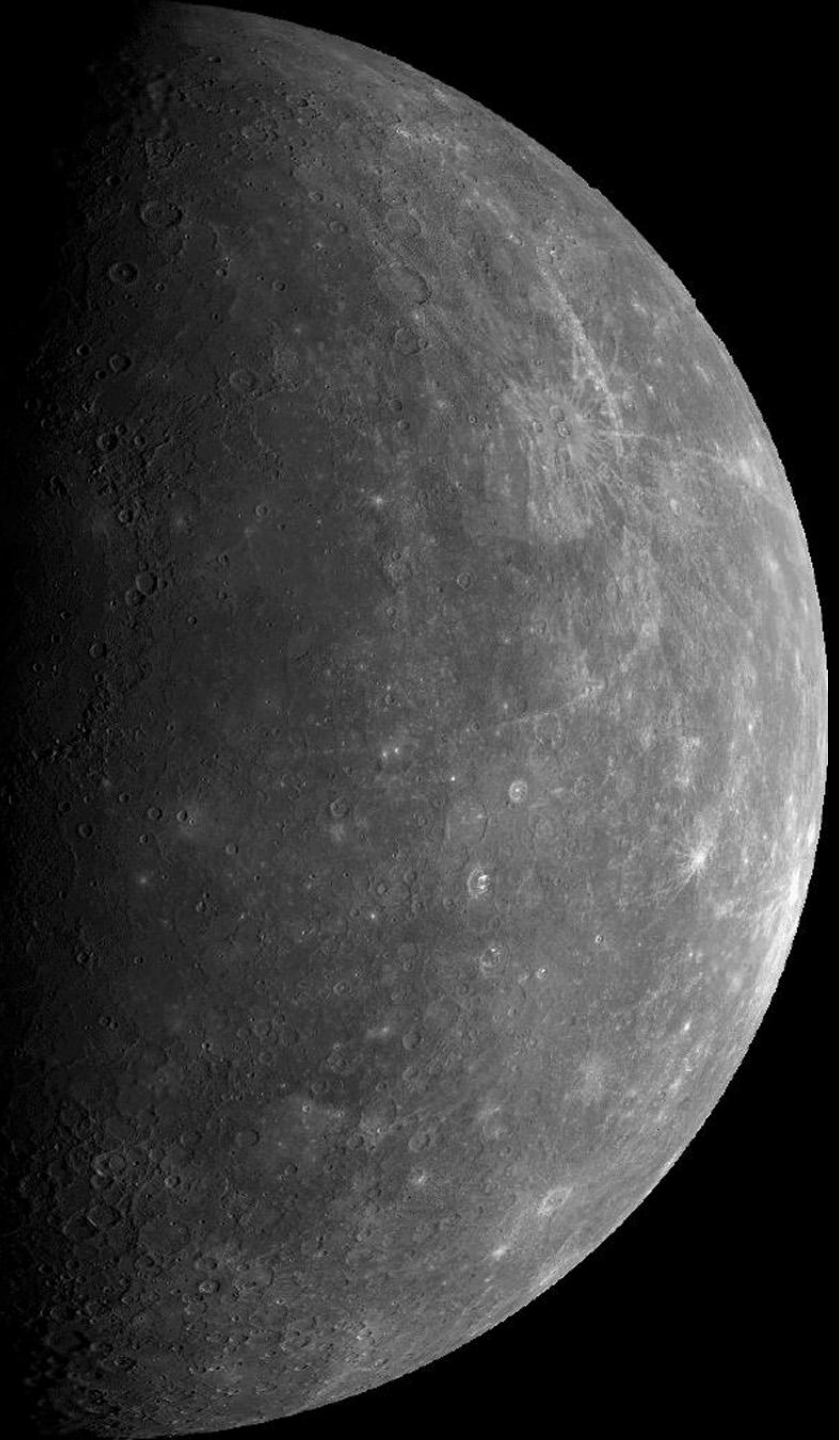
OpenCAE Vision

- Provide an open portfolio in a shared environment that seamlessly connects engineers developing missions and systems.
  - Open - Processes, code, apps, services and artifacts are accessible by JPL users as well as vendors and partners
  - Shared - The diverse community of users, developers partners and vendors are able to contribute
  - Connected - Collaboratively construct and analyze the same precision products needed to develop Missions and Systems at JPL using the CAE environment.

OpenCAE Mission

- Develop the CAE environment from a user centered architecture leveraging vendor partnerships using robust life cycle processes.
  - Vendor partnerships – Crucial feedback and insight into how Vendor products are serving the needs of engineers and developers
  - User centered – Architecture for CAE is driven by the needs of the practitioners and projects
  - Life-cycle process – Provide the integrity of the the applications services and support





Agenda

- Introduction
- OpenCAE Approach
- Open Source Contributions
- Questions

Jet Propulsion Laboratory
California Institute of Technology

Model-Based Engineering applied by Projects for delivering Engineering Products

Projects:

- Europa Clipper
- Europa Lander mission concept
- InSight
- Mars 2020
- Mars Sample Return (MSR) – potential missions concept
- Thirty Meter Telescope
- Ground Data Systems
- Psyche
- MAIA

Products:

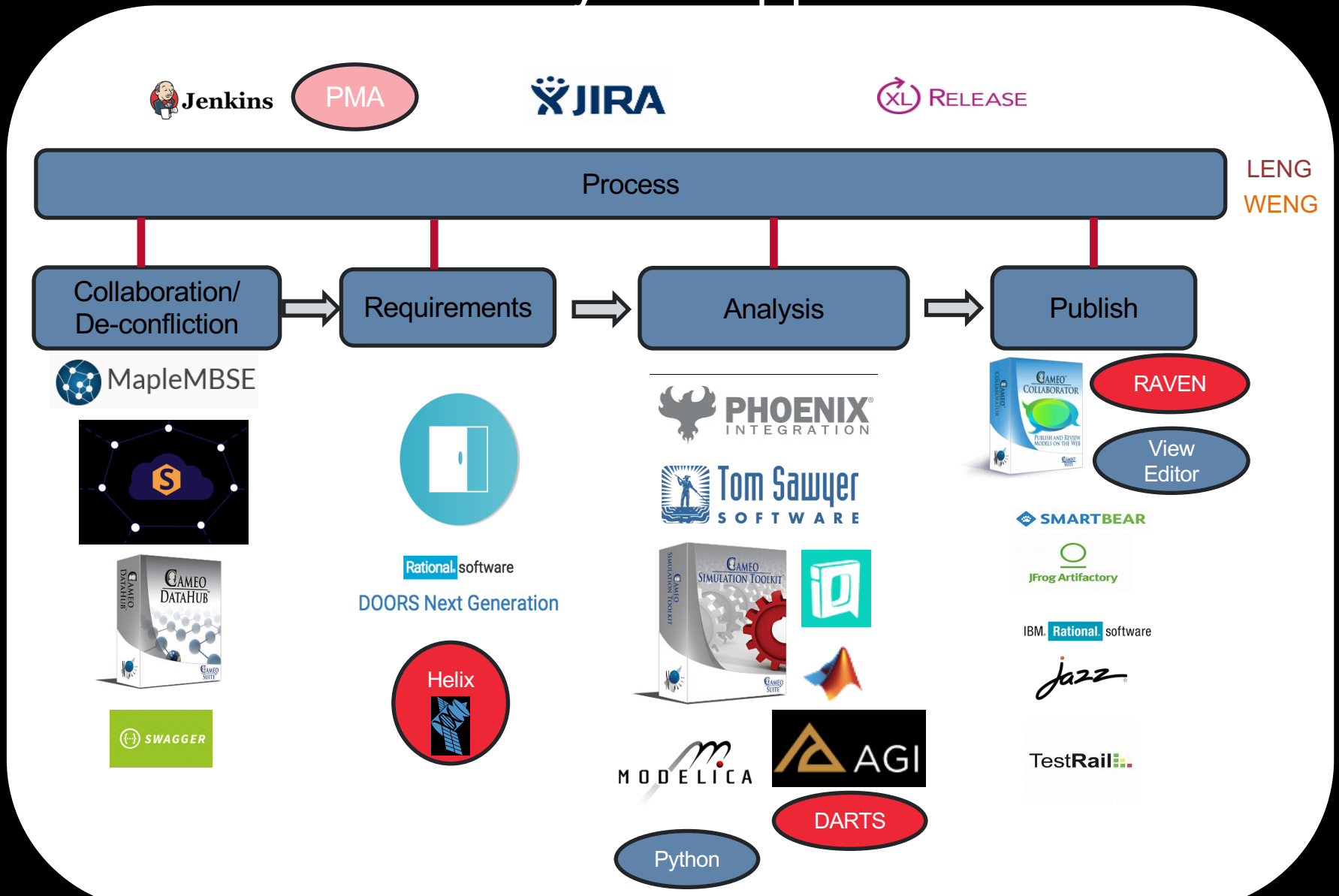
- MELs, PELs
- Resource allocation analysis
- System decomposition,
- Libraries / Reusable models

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

OpenCAE provides the engineering platform

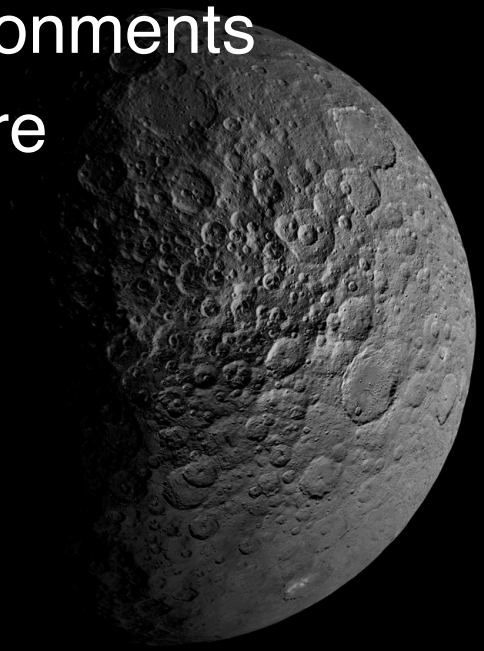
- A platform for engineering tools to work together
- Incorporate tooling from systems, software, mechanical, and electrical domains
- Platform integrates heterogenous data sources
- Emphasize standards for data interchange
- Case studies inform the architecture of the engineering environments
- Multi-model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

CAE Systems Environment Provides Integrated Life-Cycle Support



OpenCAE evolves through 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



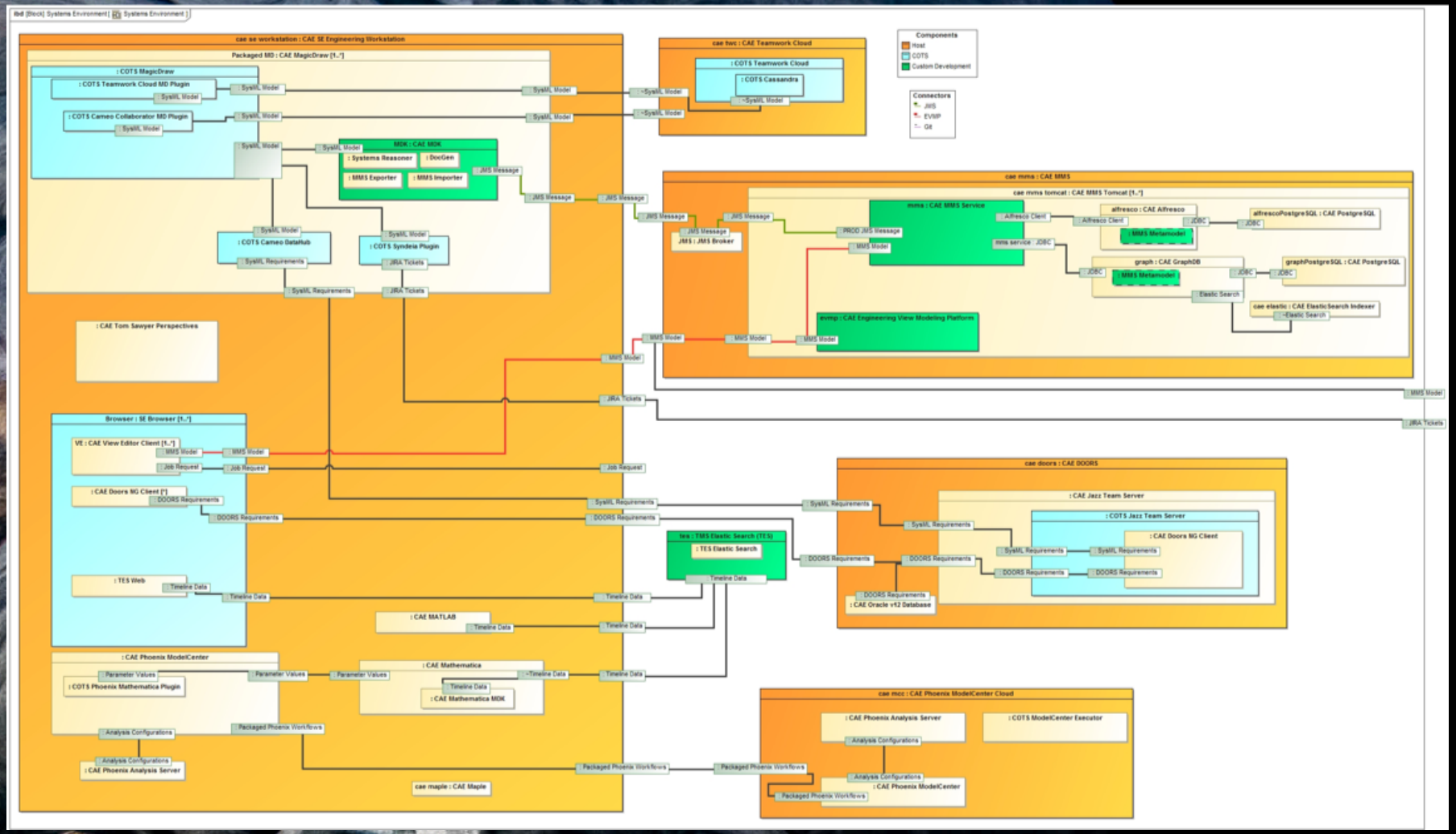
OpenCAE engages the User Community

- 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

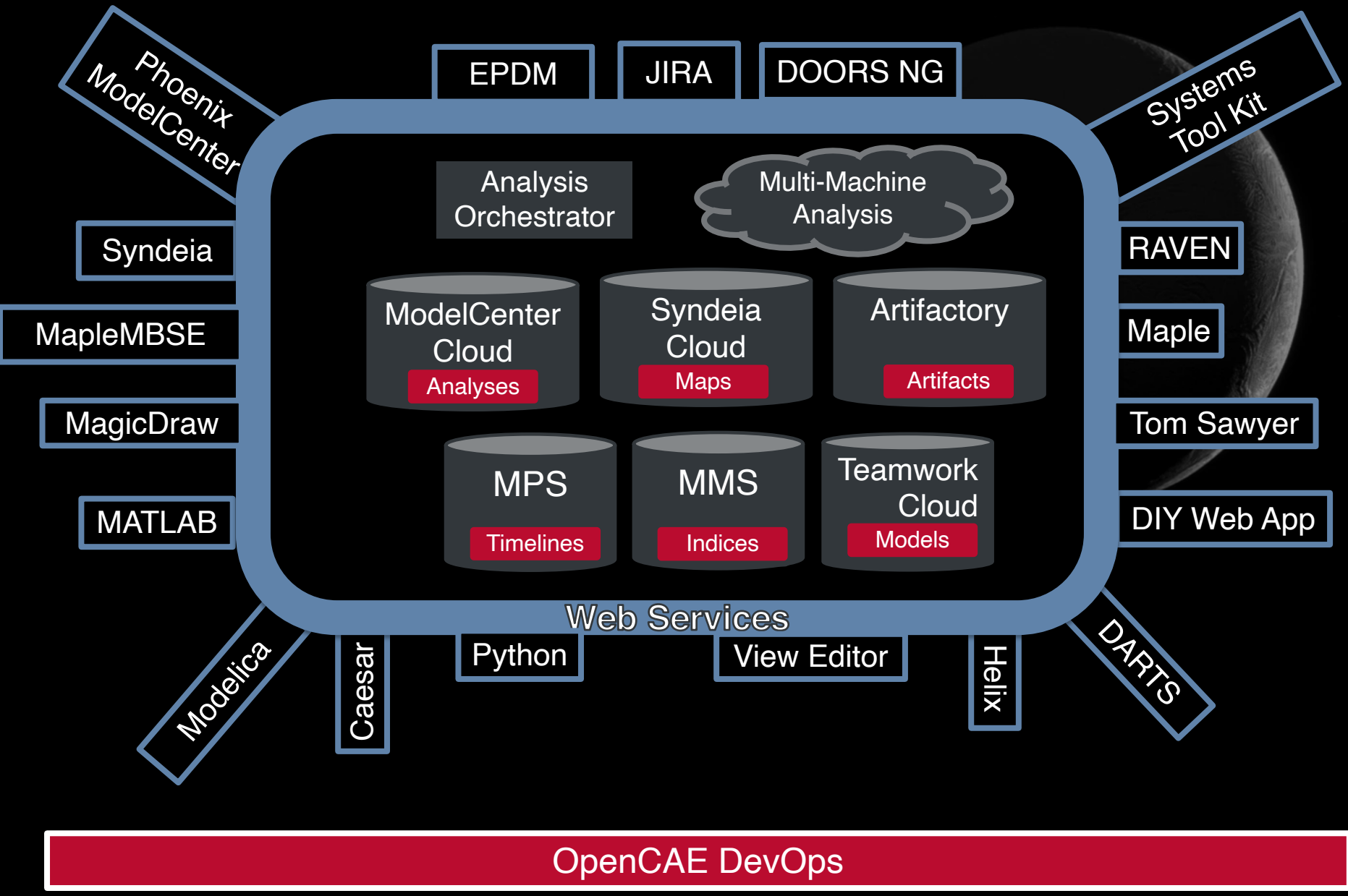
OpenCAE develops requirements through Case Studies

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

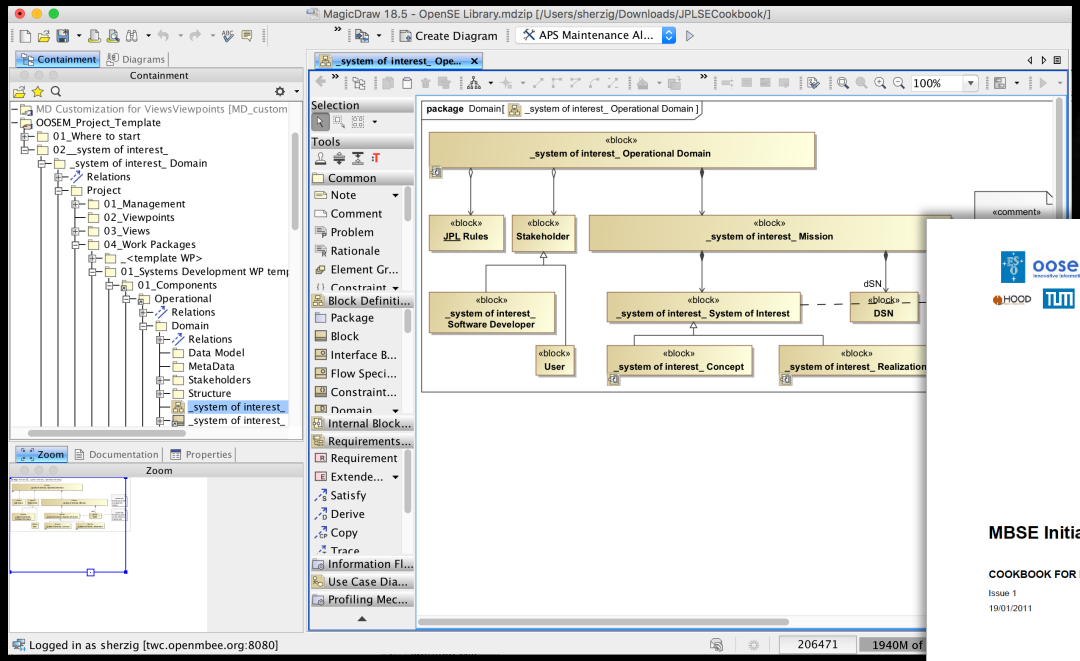
Interactions Within CAE Systems Environment



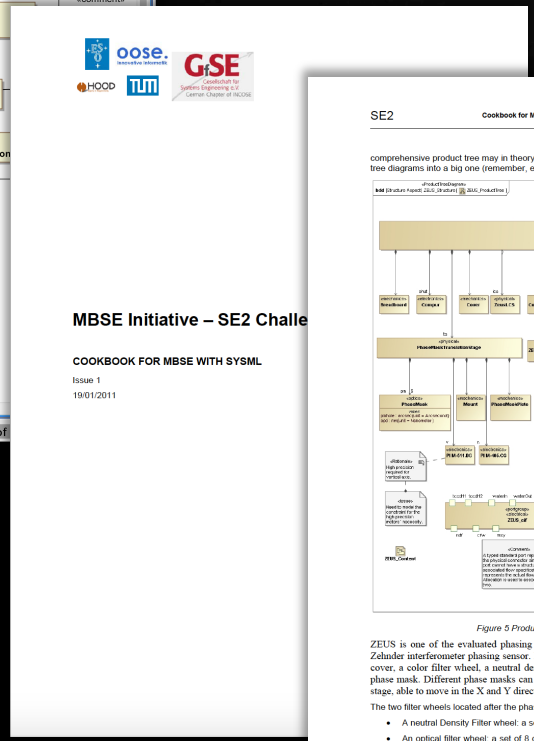
CAE delivers an integrated Systems Environment

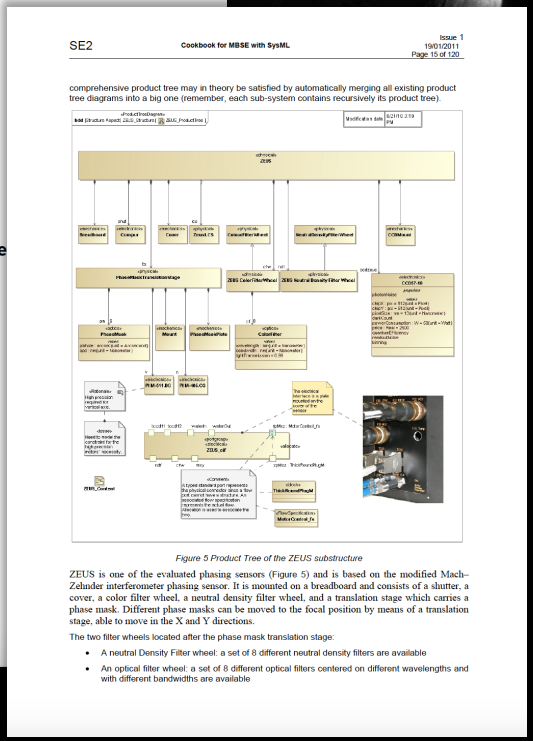


CAE Systems Environment: Cookbook and Template Model



"Cookbook" for modeling methodology and patterns





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

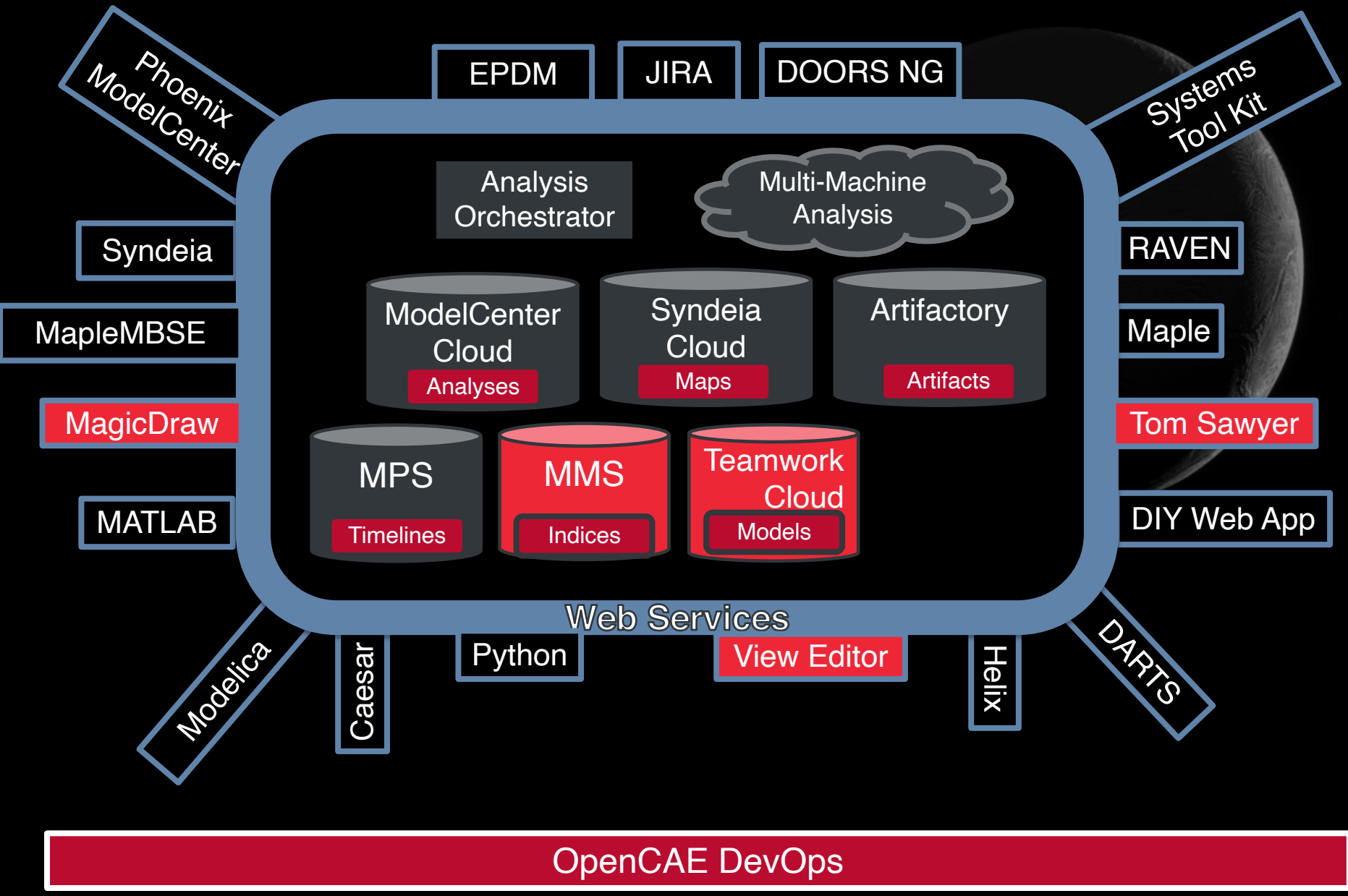
CAE offers project and domain specific adaptions

- 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

Mars2020 Embedded Role

- Ground Data System (GDS) Consolidation
- Flight System Model integrity
- Integration for generated artifacts
- Customized documents support
- Testing and Deploying various customizations of CAE Systems Environment

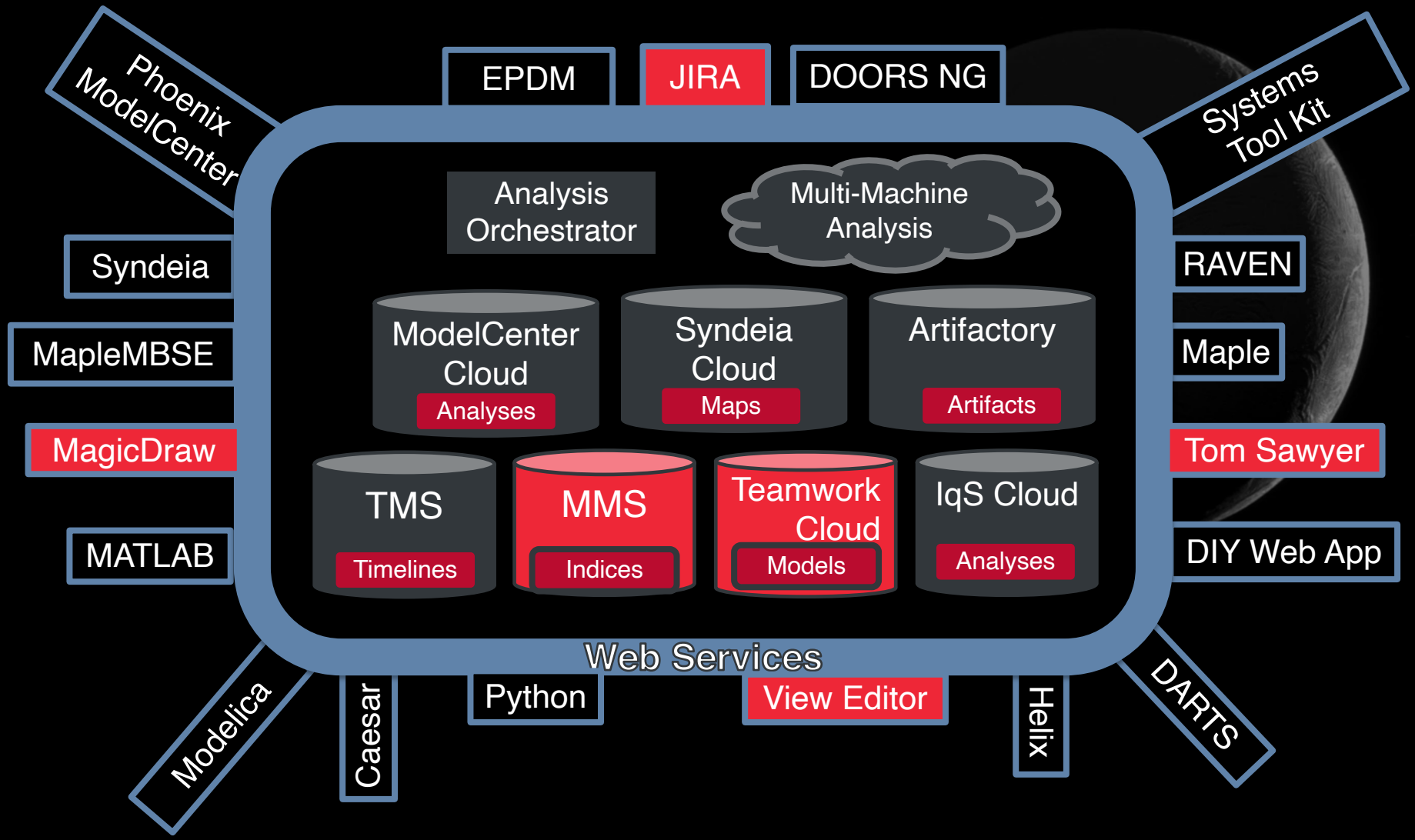
CAE delivers an integrated Systems Environment



NESC MBSE Pathfinder Embedded Role

- Apply Best Modeling Practices
- Investigate and use various workflows in environment in a restricted environment
- Specified Training and Tutorials
- Team meetings and evaluation support
- On Call Support

CAE delivers an integrated Systems Environment

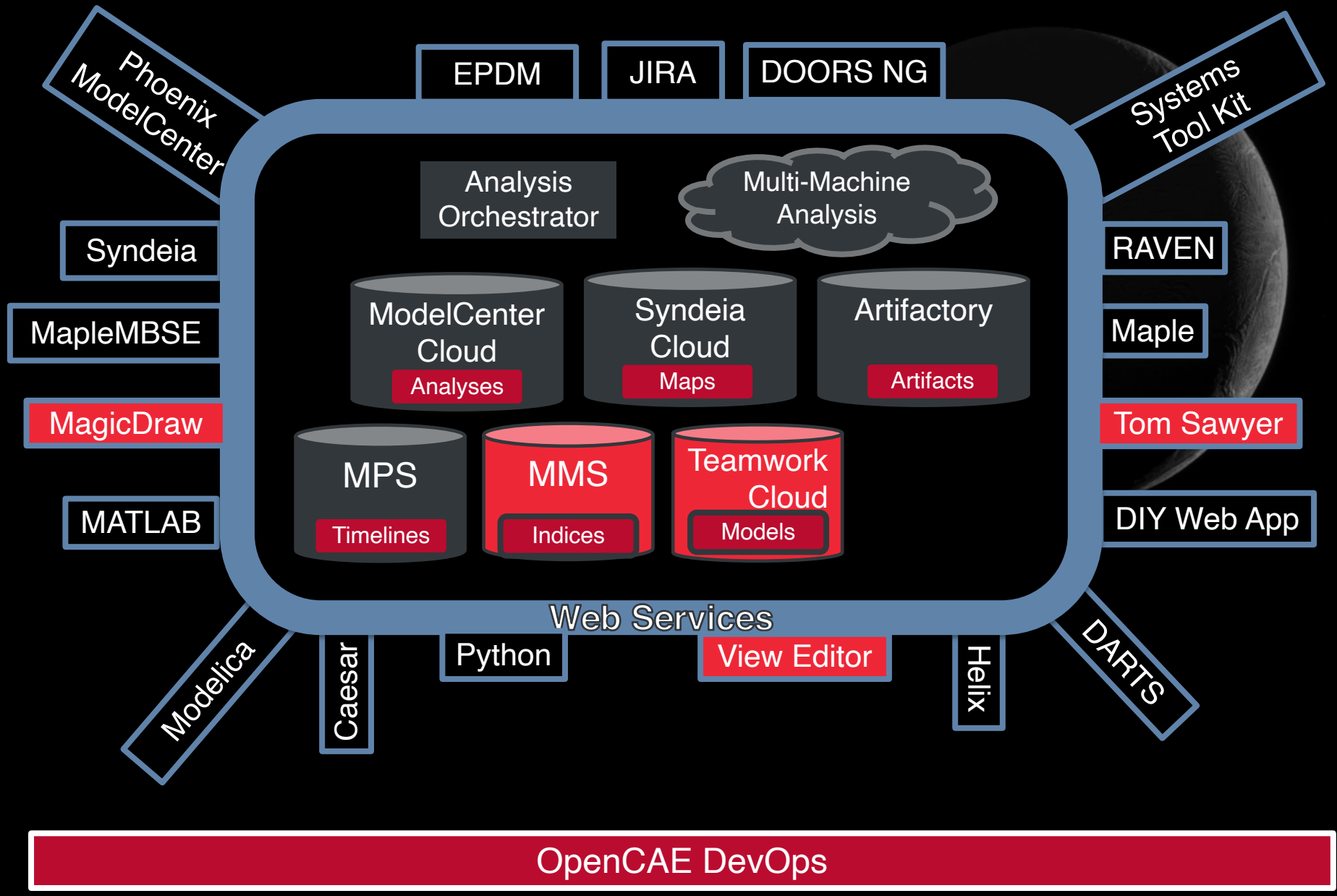


OpenCAE DevOps

Europa Lander Embedded Role

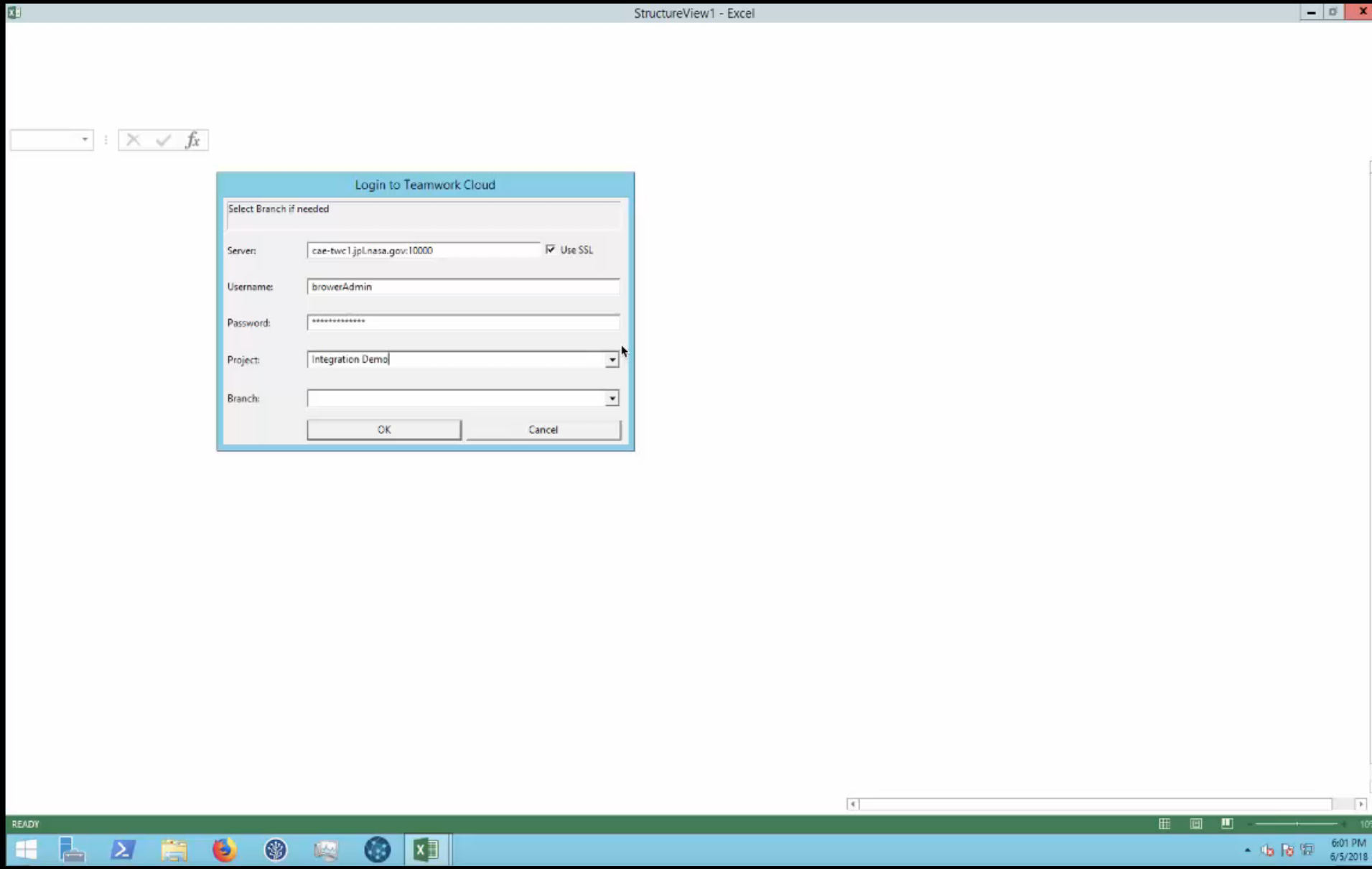
- Need:
  - Generate orderly and palatable diagrams from a system model describing the Lander
  - SE products should never be out of sync with the system model
- Approach:
  - Leverage Tom Sawyer plugin for MagicDraw development effort
  - Supply requirements directly from the project to the vendor
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

CAE delivers an integrated Systems Environment

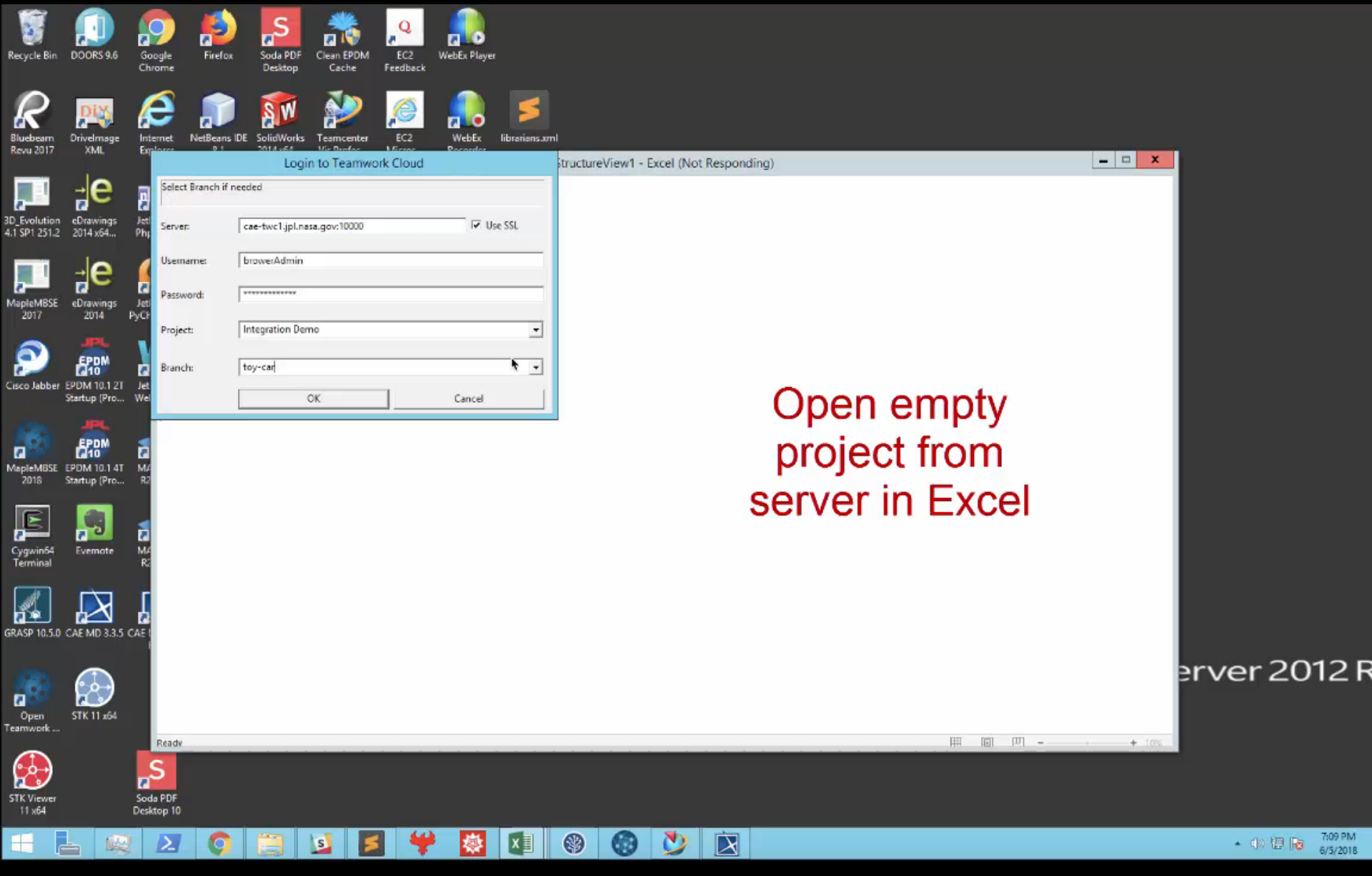


Managed Excel 1/3

Managed Excel 2/3



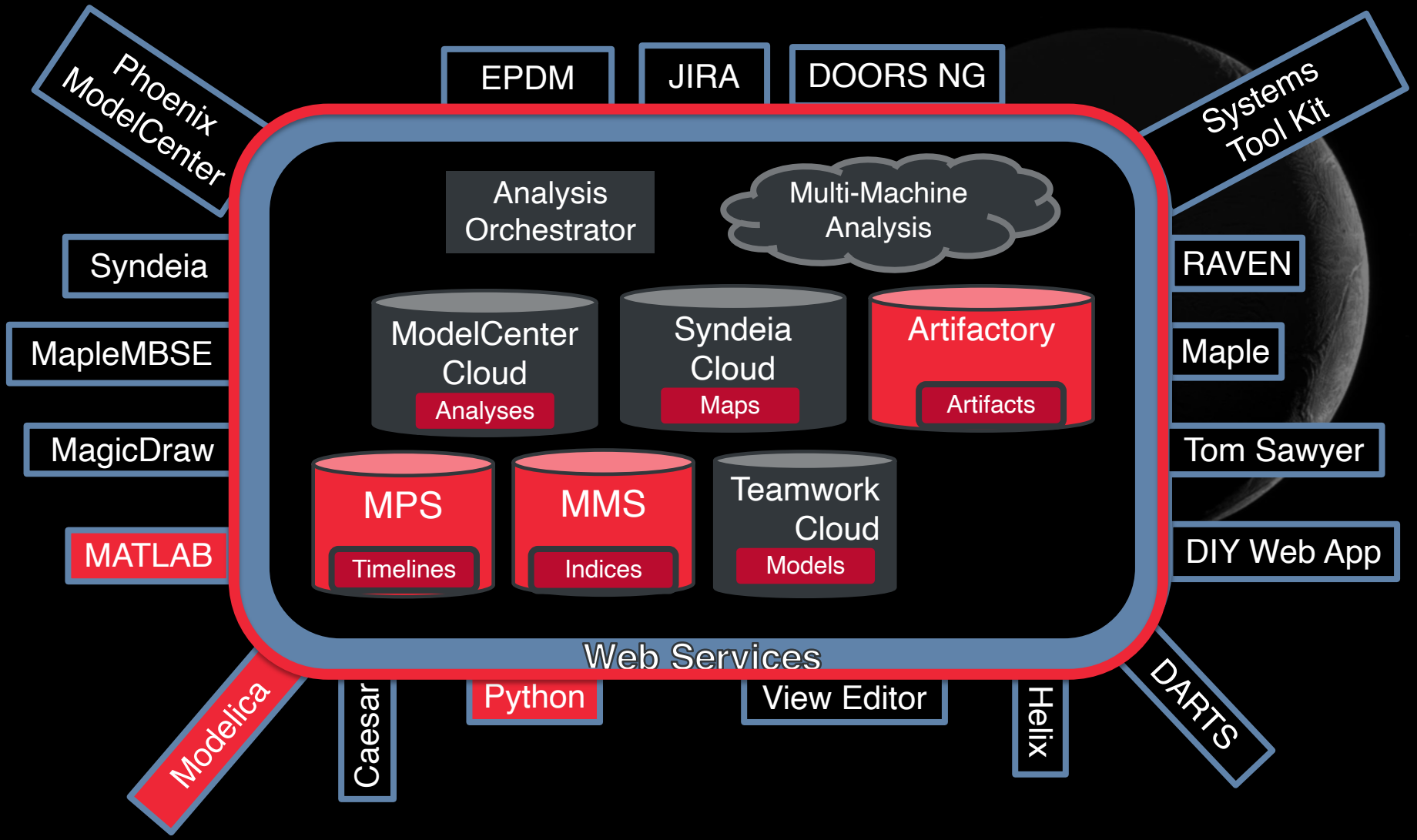
Managed Excel 3/3



Europa Clipper Embedded Role

- Need:
  - 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 delivers an integrated Systems Environment

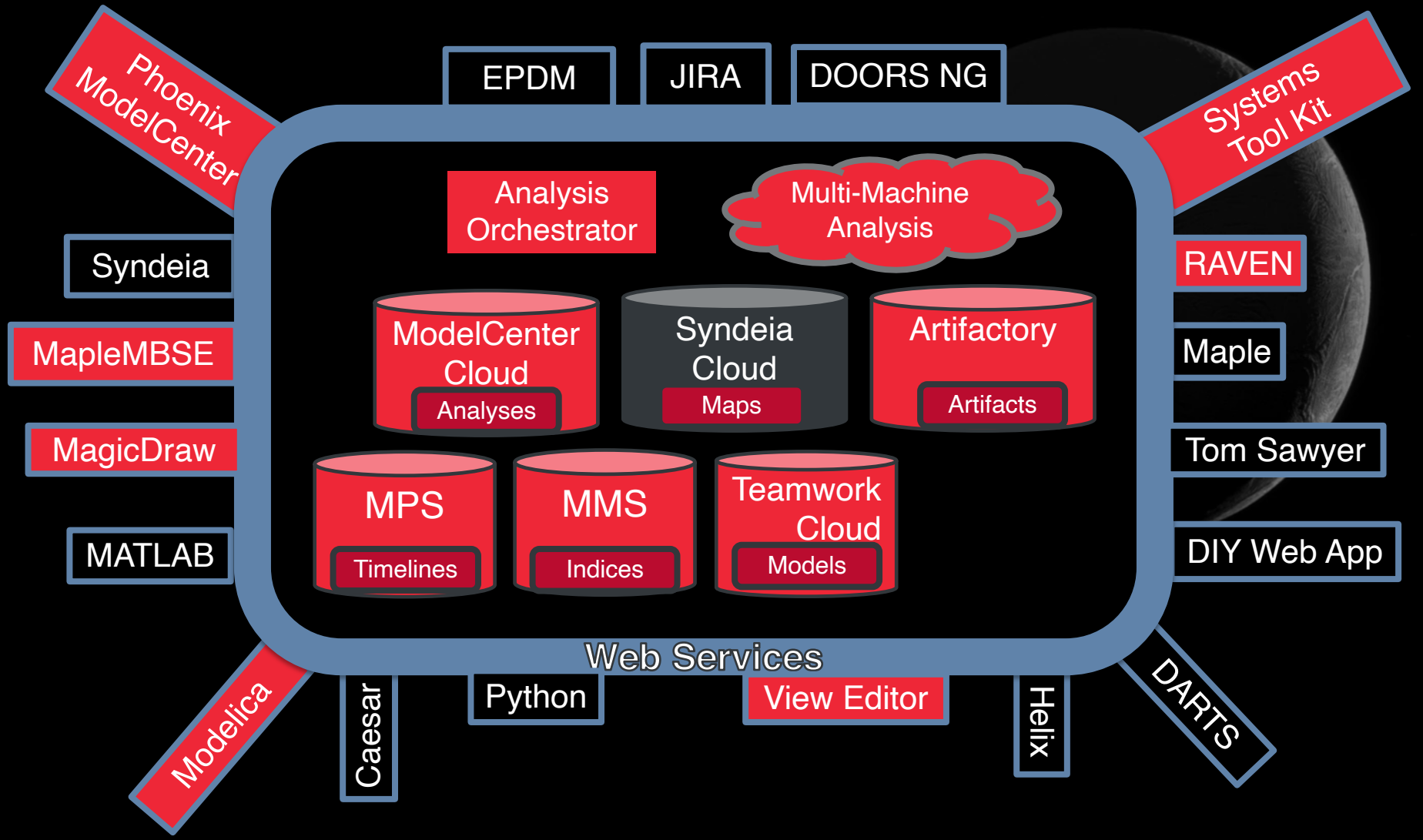


OpenCAE DevOps

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 delivers an integrated Systems Environment



OpenCAE DevOps

OpenCAE: What Has Worked

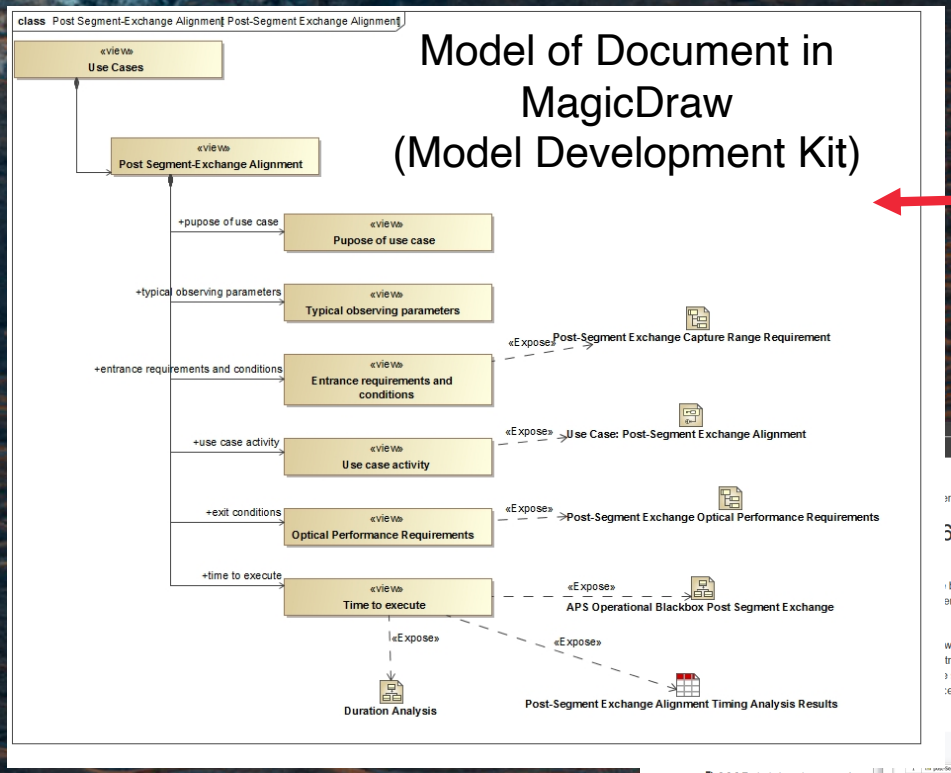
- Domain specific adaptations
  - Managed Services with vendors
  - Embedded roles
- Server-side operations preferred
  - Easier to update a server than many clients
  - COTS connections between services
- Speak the same language (SysML, FMI)
- OpenAPI REST specification
  - Generate clients for users'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

Agenda

- Introduction
- OpenCAE Approach
- Open Source Contributions
- Questions

Jet Propulsion Laboratory
California Institute of Technology

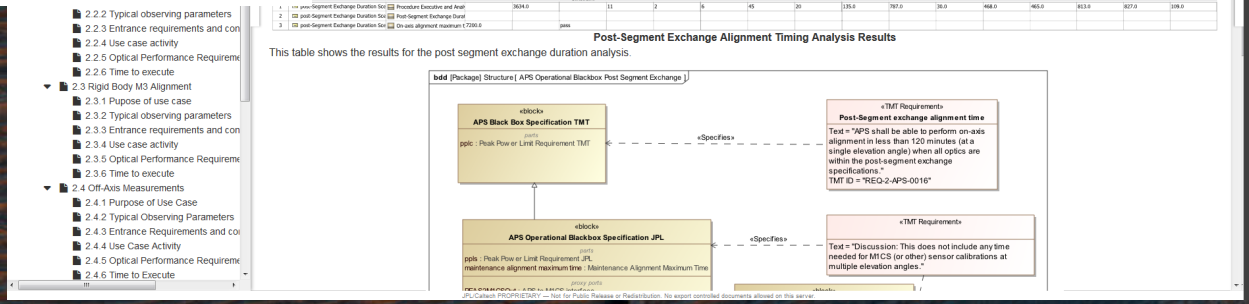
Core Integration of MMS, MDK, and VE



Model of Document in MagicDraw (Model Development Kit)

Model Repository (MMS)

Rendered and editable document in Web interface (View Editor)



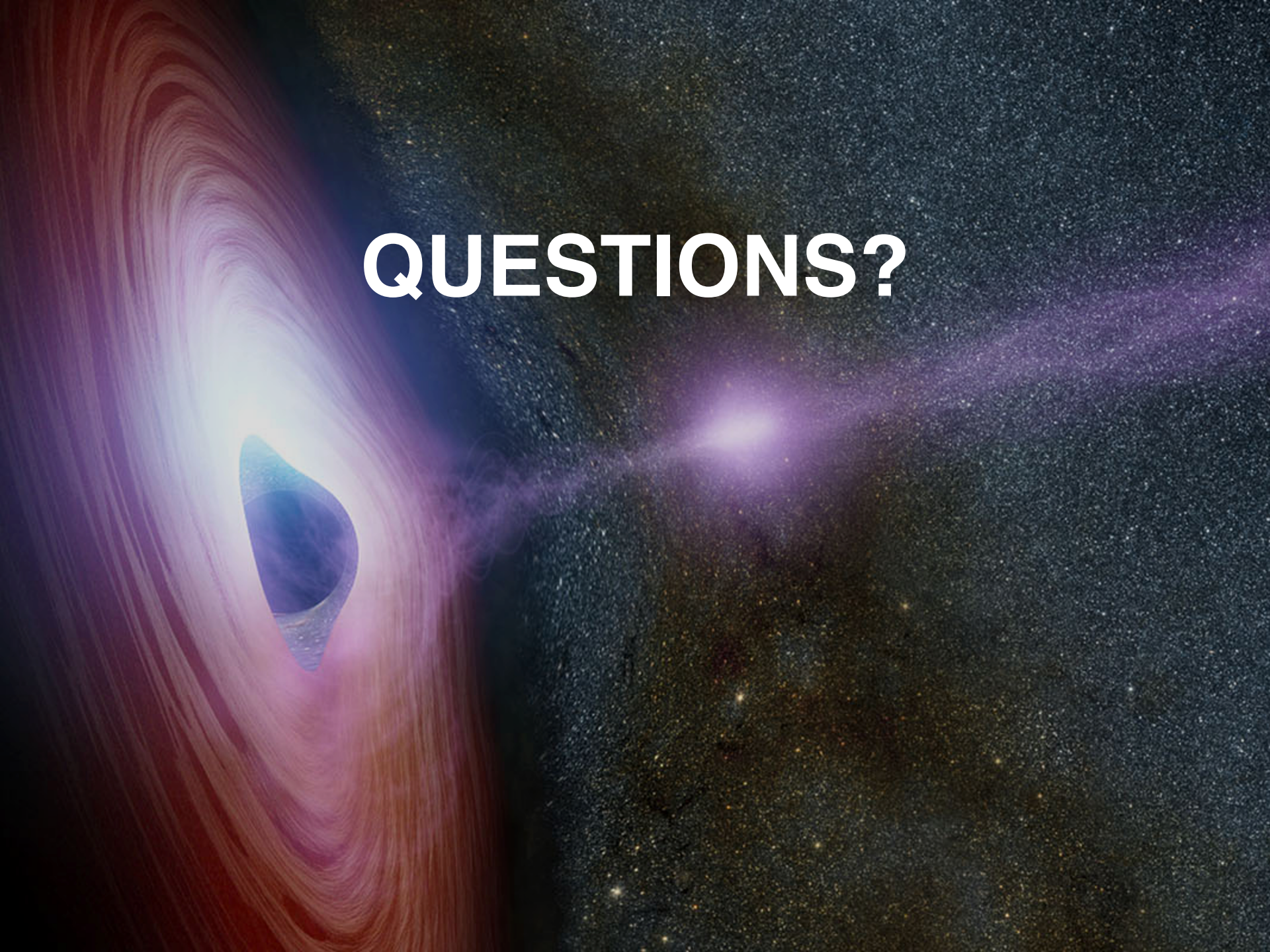
openmbee.org

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
- Along with Boeing, Lockheed, OMG, NavAir, Ford, Stevens, GaTech, ESO
- Vendor participants
- ~200 members

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/


QUESTIONS?

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov