

Open-source has become a mainstream asset in most parts of today's economy, including space. Its use is widespread and in every projects it is customary to look what existing open-source tools could be used as building blocks. Both private companies and public organizations are heavy users of open-source. The **benefits** are well-known by now (reliability, vendor-neutrality, standardization, maintainability, ...).

Creating an open-source project on the other hand is quite a different topic. For the entity that does the initial work, what could be the incentive to publish freely some cutting edge know-how? If it is a private company in a highly competitive environment and it has to pay for the development, what could be the rationale behind this move?

The OREKIT flight dynamics library is a very good example: this project, from its inception as an internal tool in a private company, **CS-SI**, up to its wide adoption in the space domain by now. It explains the reasons for the move to open-source and the rationale behind the **license selection**. This poster presents the expectations at project publications, what worked and what did not work, and the various phases **from closed source to full open governance** with a **meritocratic model**, to obtain a successful and good quality product that is now widely used !

OREKIT overview

- First Flight Dynamics **100% Open Source library** (Apache V2 non contaminating license)
- It includes Core objects (orbital parameters, propagation, frames, time scales, attitude, force models...)
- The Object-oriented architecture adapted to Flight Dynamics needs
- Contributions from FD experts from around the world
- Free software (Apache V2 non contaminating license)
- Widely adopted by industrial and Agencies & operationally used:
 - > ATV (CNES/ESA), Patrius (CNES), Skat (Eumetsat), DSST (ESA)...
- Numerous other uses, some we don't even know about...

Phase 1: In-House Product

In 2001 CS decided to develop a **new core Flight Dynamics library**

It was not intended to be a standalone product by itself, but rather a necessary asset for the company, on top of which custom systems could be developed

It aimed to answer to new identified needs in Flight dynamics systems with new technology, easy to adapt, up to date with respect to recent space flight dynamics models and still compatible with older models.

OREKIT was Developed using only internal funding & human resources and matured from a small set of core components to a collection of core classes and associated algorithms

... But Orekit was just another closed-product in an already crowded flight dynamics market, not easy to sell

Phase 3: Collaborative Library

In 2011, CS decided to extend Orekit as a **Collaborative product**.

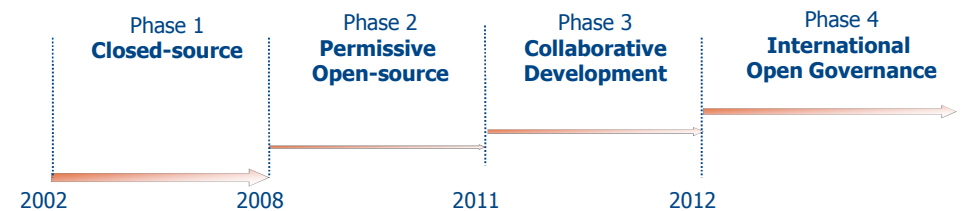
Objective: Reduce communications problems

Creation of a public forge, with public access to activity, bug reports system, source code repository, documentation, and downloads; creation of a project blog for announcements; set-up of Several public mailing lists; creation of a Public source code repository that allows Users to follow developments in real-time, but write access remained limited to the Orekit team.

- OREKIT selected for the next G space FDS at CNES (SIRIUS project) and used by Eumetsat for long term SK analysis tool (SKAT)

... But this governance model is very simple one and is effective during the early stages of a project, but it becomes difficult to keep as more people get involved, it is clearly not a model that is well suited on the long term.

Project history: a continuous Evolution



Phase 2: Open-Source License

In 2008, CS decided to publish Orekit as an **open-source license**.

Objective: The investments previously made on Orekit will be secured by using a classical service-based business model rather than a license based one.

It was necessary to avoid copy-left licenses (like GPL or CeCILL) that can be contaminating, so we chose a **permissive license** : Apache License V2.0

- OREKIT use immediately began to spread, with first external contributions and bug reporting
- Growth of OREKIT experts team inside CS

... But Some limitations (OREKIT releases decided upon CS' team, Code hosted by CS internal server)

Phase 4: Open Governance

In 2012, CS decided to improve Orekit as an **Open Governance** project.

Objective: transition to a Meritocratic model for governance

- This decision was Inspired from the Apache Software Foundation meritocratic model: "*Those who do, decide*", "*Merit is based on actual accomplishments within a project*"

Orekit governance now includes Corporate representatives in addition to individual experts from around the world

The Governance Model is based on Several roles:

- User: simple user of the library, report bugs, ask questions
- Contributor: registered to the various collaborative tools (mailing lists, forge): fix bugs, answer questions, contribute code, write documentation
- Committer: proven or regular contributor-> get direct commit access to the source code repository
- Project Management Committee: People who decide OREKIT management lines

NOW: OREKIT is worldly used !

Community is spreading...

