

# **SPICE-enhanced Cosmographia SEMINAR**

**Marc Costa Sitjà**  
**ESA SPICE Support Engineer**

**ESAC, ESA, Villanueva de la Cañada, Spain**  
**2017/03/09**

Issue/Revision: 1.0

Reference: Presentation Reference

Status: Issued

ESA UNCLASSIFIED - Releasable to the Public

## SCOPE

- Avoid the ESA SPICE Service (a.k.a. ME) dozens of e-mails on installing, configuring and running Cosmographia.

## AGENDA

- 16:00-16:20 Introduction to SPICE at ESA and Cosmographia
- 16:20-16:50 Configuration of Cosmographia for ESA Missions
- 17:00-17:40 Controls and applications of Cosmographia
- 17:40-18:00 Q&A and brainstorming

## PRE-REQUISITES

- Have an installed copy of Cosmographia in your computer and having Webex configured in your computer. You can obtain Cosmographia from the following links:
  - <https://www.cosmos.esa.int/web/spice/cosmographia>
  - [https://naif.jpl.nasa.gov/naif/cosmographia\\_components.html](https://naif.jpl.nasa.gov/naif/cosmographia_components.html)

## IMPORTANT DISCLAIMER:

This is a very personal approach of using and setting up Cosmographia and is not carved in stone neither blessed by NAIF aimed to facilitate both your lives and my life and getting you introduced in Cosmographia. Sentences like *"Oh but Marc said that this works like this"* or *"Marc said it was like that"* should not be used ☺

# Introduction – SPICE



- SPICE is an information system that uses *ancillary data* to provide Solar System geometry information to scientists and engineers for planetary missions in order to plan and analyze scientific observations from space-born instruments.
- When we talk about “ancillary data” we talk, *minimum* of spacecraft trajectory and orientation.
- SPICE was originally developed and is maintained by the Navigation and Ancillary Information Facility (NAIF) team of the Jet Propulsion Laboratory (NASA).
  - **SPICE** provides users a large suite of SW used to read SPICE ancillary data files to compute observation geometry.
  - **SPICE** is used to organize and package these data in a collection of files called “kernels”
  - **SPICE** includes SW for writing, reading kernels and computing observation geometry from kernels
- Learn more about SPICE here:
  - [spice.esac.esa.int](http://spice.esac.esa.int)
  - [naif.jpl.nasa.gov](http://naif.jpl.nasa.gov)
- SPICE Training classes are available in the US in a ~yearly basis and in ESA/Europa in a ~biannual basis.
  - <https://naif.jpl.nasa.gov/naif/training.html>
  - <https://www.cosmos.esa.int/web/spice/training>
- You can also teach SPICE yourself: [https://naif.jpl.nasa.gov/naif/self\\_training.html](https://naif.jpl.nasa.gov/naif/self_training.html)

# Introduction – ESA SPICE Service



- The ESA SPICE Service is responsible for the generation and maintenance of SPICE data along with providing consultancy, training and utilities related to exploiting SPICE data for ExoMars 2016, BepiColombo, JUICE, Solar Orbiter, Mars Express, Rosetta and Venus Express.
- The ESA Home page for SPICE:
  - **spice.esac.esa.int** or **http://www.cosmos.esa.int/spice**
- The ESA Home for SPICE contains some information on the ESA SPICE Service and points to:
  - The available operational SPICE kernel collections
  - Available archived SPICE kernels
  - Original data used to produce the SPICE kernels
  - Contains directions to use WebGeocalc and Cosmographia.
- ExoMars 2016 SPICE Server:
  - **ftp://spiftp.esac.esa.int/data/SPICE**

The screenshot shows the 'WELCOME TO THE ESA HOME FOR SPICE' page. It features a navigation menu on the left with links for Home, SPICE for ExoMars 2016, SPICE for Rosetta, SPICE for MEX, SPICE for BepiColombo, SPICE for JUICE, SPICE for VEX, SPICE for SMART-1, WebGeocalc, Cosmographia, and Training. The main content area includes sections for 'ABOUT SPICE', 'DATA', 'OPERATIONAL KERNELS', 'ARCHIVED KERNELS', and 'SERVICES'. The 'ABOUT SPICE' section explains that SPICE is an information system for planning and analyzing spacecraft observations. The 'OPERATIONAL KERNELS' section lists various mission-specific kernels like MEX-E-R-SPICE-V1.0 and ROS-E-R-C-SPICE-V1.0. The 'SERVICES' section mentions WebGeocalc and Cosmographia.

The screenshot shows the 'spice' page for BepiColombo. It features a navigation menu on the left with links for Home, SPICE for ExoMars 2016, SPICE for Rosetta, SPICE for MEX, SPICE for BepiColombo, SPICE for JUICE, SPICE for VEX, SPICE for SMART-1, WebGeocalc, Cosmographia, and Training. The main content area includes sections for 'ABOUT SPICE', 'DATA', 'OPERATIONAL KERNELS', 'ARCHIVED KERNELS', and 'SERVICES'. The 'ABOUT SPICE' section explains that SPICE is an information system for planning and analyzing spacecraft observations. The 'OPERATIONAL KERNELS' section lists various mission-specific kernels like MEX-E-R-SPICE-V1.0 and ROS-E-R-C-SPICE-V1.0. The 'SERVICES' section mentions WebGeocalc and Cosmographia.

Index of <ftp://spiftp.esac.esa.int/data/SPICE/JUICE/kernels/>

[Up to higher level directory](#)

Name	Size	Last Modified
<a href="#">aareadm.txt</a>	8 KB	19/08/16 00:00:00
<a href="#">ck</a>		01/02/17 14:20:00
<a href="#">dsk</a>		29/09/16 08:36:00
<a href="#">fk</a>		01/02/17 15:08:00
<a href="#">lk</a>		16/02/17 17:27:00
<a href="#">lsk</a>		01/02/17 14:21:00
<a href="#">mkk</a>		13/02/17 16:25:00
<a href="#">pck</a>		01/02/17 14:21:00
<a href="#">sck</a>		01/02/17 14:21:00
<a href="#">spk</a>		13/02/17 16:40:00

# Introduction – Cosmographia



- In Space missions design, planning, operations and data analysis the ability to visualize and interpret the geometry of a given mission in an easy and comprehensive manner is key.
- **SPICE-enhanced Cosmographia**<sup>1</sup> is a downloadable tool used to help visualize and analyze astrodynamics and science aspects of a mission in a 3D scenario (it is a spin-off of the well known Celestia tool)
- It can depict such things as:
  - spacecraft trajectory and orientation
  - target body ephemeris and size, shape and orientation
  - reference frames (coordinate systems)
  - vectors and angles
  - ranges
  - instrument view cones and footprints
  - ... and more.
- The SPICE-enhanced Cosmographia distributed by NAIF makes use of SPICE kernels.
- Everything you need is to install Cosmographia, and download the configuration files and SPICE kernels for the mission you want to use. Should be easy right?
- Does anybody need a brief introduction to SPICE? If, so please check the BACKUP slides at the end of the presentation.

*<sup>1</sup>It is important to stress the "SPICE-enhanced" part. Please do so when you reference the tool.*

# Configuration - Resources



## *“It did not have to happen”*

There are enough resources provided by NAIF and by the ESA SPICE such that you can do this on your own, nevertheless It can take a substantial amount of time and you might need some SPICE-knowledge that you might not be supposed to have. So be the Seminar.

### **0-AVAILABLE RESOURCES:**

- <https://www.cosmos.esa.int/web/spice/cosmographia>
- <https://naif.jpl.nasa.gov/naif/cosmographia.html>

Lets look at what the ESA SPICE Service offers:

- Relatively nice web page with a couple of nice pictures
- A beautiful Readme file: <ftp://spiftp.esac.esa.int/cosmographia/missions/aareadme.txt>
- A link to the Cosmographia User Guide: <https://cosmoguide.org/>

**If you carefully looked at those two documents and if you have a basic knowledge of SPICE, this is it, you can leave this seminar. Godspeed!**

Lets look at what NAIF offers:

- NAIF has several links that point out all the information you need, being the most important of all:
- [https://naif.jpl.nasa.gov/naif/cosmographia\\_details.htm](https://naif.jpl.nasa.gov/naif/cosmographia_details.htm)
- [https://naif.jpl.nasa.gov/naif/cosmographia\\_components.html](https://naif.jpl.nasa.gov/naif/cosmographia_components.html)

# Configuration - Download



## 1-DOWNLOAD COSMOGRAPHIA:

- <https://www.cosmos.esa.int/web/spice/cosmographia>
- [https://naif.jpl.nasa.gov/naif/cosmographia\\_components.html](https://naif.jpl.nasa.gov/naif/cosmographia_components.html)
  - [SPICE-enhanced Cosmographia for Mac OSX \(64-bit\)](#)
  - [SPICE-enhanced Cosmographia for Windows \(64-bit\)](#)
  - [SPICE-enhanced Cosmographia for Linux \(64-bit\)](#)

**First showstopper: You need to have a 64-bit machine  
Workaround: Use a Linux Virtual Machine (Ubutnu with Oracle's Virtual Box)**

- Extract it and install it in your favorite OS.
- Preferably put the **cosmographia3** directory under a higher level **cosmographia** directory where you will put the rest of the files
- We see three important documents inside the cosmographia directory:
  - `userguides/CosmographiaUsersGuide_v8` => You can use it alternatively to the on-line one
  - `Aareadme` => You can ignore it for the time being for is more NAIF specific
  - `templates/` => You can use it if you want to generate your own configuration
- Lets do a dry run of Cosmographia.

# Configuration – Mission JSONs 1



## 2-DOWNLOAD MISSION CONFIGURATION FILES:

- Download you preferred ESA Planetary Mission -or all alternatively- from the ESA SPICE page:

Mission	Repository Browser	Zipped package
JUICE	<a href="#">cosmographia/missions/JUICE/</a>	<a href="#">JUICE</a>
ExoMars 2016	<a href="#">cosmographia/missions/EM16/</a>	<a href="#">EM16</a>
BepiColombo	<a href="#">cosmographia/missions/BEPIC</a>	<a href="#">BEPIC</a>
MARS-EXPRESS	<a href="#">cosmographia/missions/MEX</a>	<a href="#">MEX</a>
ROSETTA	<a href="#">cosmographia/missions/ROS</a>	<a href="#">ROS</a>
VENUS-EXPRESS	<a href="#">cosmographia/missions/VEX</a>	<a href="#">VEX</a>
All Missions	<a href="#">cosmographia/missions</a>	<a href="#">All Missions</a>

- Unzip the directory and generate a missions directory under the cosmographia directory where you will put the mission specific subdirectory. Your Cosmographia directory will look like:

- cosmographia/
  - missions/
    - MEX/JUICE/EM16/BEPIC/VEX/ROS
  - cosmographia3/
  - [cosmo\\_spice/ \(If you also download the NAIF missions\)](#)

- Now we take a look at the structure of the Cosmographia Configuration files for a given mission

# Configuration – Mission JSONs 2



The set of files under each mission subdirectory includes the following directories JSON and supporting files:

- **config** subdirectory containing configuration JSON catalog files `spice_*.json` and `spacecraft_*.json`
  - **config/spice\_\*.json** JSON file defining the SPICE kernels set to be loaded. Normally all mission MKs and, for some missions, the latest generic PCK provided under "kernels" are listed in this file. The SPICE name of the mission is normally the only token following "spice\_" in the JSON file name.
  - **config/spacecraft\_\*.json** JSON file(s) defining spacecraft(s). One or more of these files can be present. The SPICE name of the spacecraft is the first or only token following "spacecraft\_". The additional suffix "\_arcs" indicates that the JSON file defines the spacecraft trajectory with more than one arc.
  - **config/body\_\*.json** JSON file(s) defining natural body(es). One or more of these files can be present. The SPICE name of the body is the first or only token after "body\_". The additional suffix "\_arcs" indicates that the JSON file defines the body trajectory with more than one arc, normally as a SPICE-based arc only during the SPICE archive coverage and as Cosmographia built-in arcs before and after that.
- **kernels** -- subdirectory containing meta-kernels from the mission SPICE archive. These MKs have absolute paths set that need to be set up as indicated later in this document. If you are only using SPICE for Cosmographia. I recommend you to use this directory to store the SPICE kernels as well.
- **models** subdirectory containing spacecraft and/or body 3D models pointed to by spacecraft and body JSON files.

# Configuration – Mission JSONs 3



- **observations** subdirectory containing JSON files defining observations. These observations are defined by the Science Ground Segments of every mission.
  - **observations/obs\_\*.json** JSON files defining observations (stored in a subdirectory named observations). The SPICE name of the sensor carrier is the first token after "obs\_". The SPICE name of the sensor is the second token after "obs\_". The SPICE name of the sensor's target body is the third token separated from the first one by "\_". The starting and end date of the observation in YYYYMMDD format is the third token also separated by "-". The last token is an index number.

**ADVANCED TIP: These are the files that connect the mission scenario with the current planned or performed observation scheduled by the mission SGS.**

- **sensors** subdirectory containing JSON files defining sensors.
  - **sensors/sensor\_\*.json** JSON files defining sensors. The SPICE name of the sensor carrier is the first token after "sensor\_". The SPICE name of the sensor is the second one and the target body of the sensor is the last one.

**ADVANCED TIP: These are the files that connect Cosmographia with the SPICE Instrument Kernels of each instrument.**

# Configuration – Mission JSONs 2



- **scenarios** subdirectory containing JSON files defining scenarios loadable by Cosmographia.
  - **scenarios/load\_\*.json** JSON files that load multiple JSON files together in the right order to make a set of relevant objects available in Cosmographia at once. The "load\_" JSON files that have only the mission name in the file name load only the "spice\_" JSON, one of the "spacecraft\_" JSONs, and one or more "body\_" JSONs. The "load\_" JSON files that have ALL\_SENSORS in the file name load all "sensors\_" JSONs in addition to "spice\_", "spacecraft\_", and "body\_" JSONs but do not load any observations.

**If you don't run scripts, this is the main file you will load.**

- **scripts** subdirectory containing python and command line.
  - **scripts/\*.py** Python script that loads a Cosmographia scenario and automatically manipulates and navigates through Cosmographia. Typically the name of the script refers directly to the scenario name.
  - **scripts/run\_\*.csh/.bat** Unix/Windows command line script to start Cosmographia.

**Python scripts (which are very powerful as you will see) don't run with the Windows version**

**To run the Command line scripts you need to add Cosmographia to your path. This is the typical thing I forget about everytime that I do so do not ask me.**

# Configuration – SPICE kernels 1



## 3-DOWNLOAD SPICE KERNELS:

***NO KERNELS NO GLORY***

- Kernels for ESA missions can be obtained from different sources and they depend on the “kernel status”. There are two types of kernels:
  - **Operational kernels:** Used in the daily operations of the mission SGS and PI teams. They always contain the latest information and coverage available.
  - **Archived kernels:** Peer reviewed and “high-quality” kernels part of the science archive of the mission. Subject to releases and they do not contain the latest information available.
- Operational kernels are obtained from the operational FTP: <ftp://spiftp.esac.esa.int/data/SPICE/> This is the option recommended to use Cosmographia for ESA Missions.
- Archived kernels can be obtained from the PSA or from NAIF.
  - <https://www.cosmos.esa.int/web/psa/ancillary-data>
  - [https://naif.jpl.nasa.gov/naif/data\\_archived.html](https://naif.jpl.nasa.gov/naif/data_archived.html)
- You only need to download the kernels specified in the meta-kernel. A meta-kernel is provided for each mission in the Mission directory that you have downloaded

**META-KERNEL (a.k.a FURNISH kernel) lists all the kernels needed to load a certain mission scenario. A user should ALWAYS use it as a reference and should not embark the adventure of sorting out for his own which are the kernels to be used.**

- You can install the kernels in any location of your computer because of this all meta-kernels will have to be edited to replace the path in the PATH\_VALUES keyword with the actual absolute path to the archive's data directory.

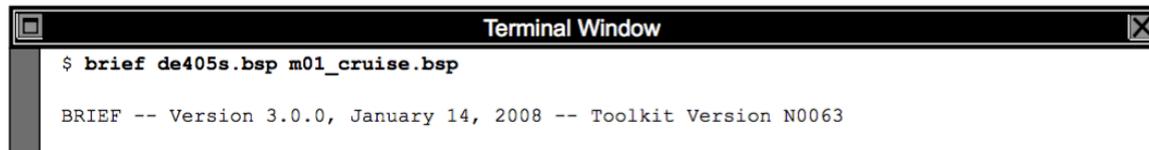
# Configuration – SPICE kernels 2



- It is recommended to use a tool such as WinSCP, FileZilla or alternatively other synchronization command lines tools.
- The ESA SPICE Service admits that facilitating kernel management for users is a pending task and it will be addressed in the future
- Alternatively and without making a lot of noise.... The ESA SPICE Service can provide you with a python script by Alexey Malakov from the ACS ExoMars 2016 team in IKI.
  - <ftp://spiftp.esac.esa.int/temp/spicesync/>

## 4-CHECK SPICE KERNELS:

- Now we need to modify the meta-kernel PATH\_VALUE with the absolute path where we have placed our kernels.
- After this I would NOT recommend to start using Cosmographia but instead I would first “check” the kernels outside Cosmographia. We will use the NAIF utility **brief** to do so.
- *brief* is a simple command line program for summarizing the contents of SPK or binary PCK files
- We obtain *brief* from here: <https://naif.jpl.nasa.gov/naif/utilities.html> by downloading the version for our OS and copying it into where the meta-kernel is
- We open a terminal window and we run it on the mk like this:

A screenshot of a terminal window titled "Terminal Window". The window shows a command prompt with the command "\$ brief de405s.bsp m01\_cruise.bsp" and the output "BRIEF -- Version 3.0.0, January 14, 2008 -- Toolkit Version N0063".

```
Terminal Window
$ brief de405s.bsp m01_cruise.bsp
BRIEF -- Version 3.0.0, January 14, 2008 -- Toolkit Version N0063
```

# Configuration – Final Setup



## 5-FINAL SETUP:

There are a couple of additional steps we need to perform to run Cosmographia scripts:

- We need to modify the paths within the python scripts
- We need to include Cosmographia in the path

## 5-RUN COSMOGRAPHIA:

There are three ways to load a mission scenario in Cosmographia:

1. Open Cosmographia and Load a Scenario file
2. Open Cosmographia and run a Python Script (Sorry only available to MAC/LINUX users)
3. Run a bash/cshell script that will open Cosmographia and open a scenario file.

**RESOURCES: Cosmographia performs a very intensive usage of your GPU and you have to be careful because it will suck the battery out of your laptop**

# Usage – DEMO

**DEMO by Marc Costa** –not to be done by the participants-:

1. Run Rosetta script: run\_150214-ROS\_NAC-CG.csh
2. Run Rosetta Python script for all mission
3. Run Philae descent script: run\_141112-ROS\_NAC-CG.csh
4. Look for Lutetia NAC images at the archive
5. Look for JUICE closest approach to Europa in WebGeocalc and load the scenario
6. Run BepiColombo Science Phase example. Use to activate vectors and angles
7. Load an ExoMars 2016 scenario and show loading and unloading sensors
8. Load MEX on top of the ExoMars 2016 scenario go to 2017-Mar-06 MCO2.2
9. Load VEX scenario.



# Usage – Interesting stuff



- **OBSERVATION CATALOGUES:** Observations are used to visualize when a given instrument or suits of instruments have performed a science observation, it is often more visual with cameras and imagers. An observation needs a sensor definition and it can determine whether if a sensor Field-of-view and frustrum is shown or not.
  - Most of the ESA missions DO NOT have observations defined. Currently Rosetta and BepiColombo have observations defined for demo purposes which do not correspond to actual observations
  - JUICE has a Cosmographia plug-in written by Rafael Andres (ESAC/ESA) which generates Observation catalogues from input science planning files –and other inputs-, for the time being this is only used by the JUICE SOC. Future will tell what is done.
  - Observation definition and generation is one of the most attractive capabilities of Cosmographia to be exploited.
  
- **SENSOR CATALOGUES:** Sensors can be “stand-alone” or alternatively they can be connected to an observation. There is a tag in the sensor catalogue in which it can be specified whether if it is only shown during an observation or all the time. Also if you look at the sensors available for all missions you will see that there are multiple copies of the same sensor with different “targets” in the filename and some ending with “obs”.
  - In order to extract information out of an observation from a given sensor the target of the sensor needs to be specified (Mars) and if there are multiple targets (Mars, Phobos), multiple catalogues need to be defined.

# Usage – Most used Controls 1



## Most used commands for Loading and unloading configurations:

Action	Control	Shortcut	Utility
Open Scenario	File: Open Catalog...	CMMD + O	Loads a given mission scenario
Run Python Script	File: Run Script...	CMMD + ALT + R	Loads and launches a python script.
Exit Cosmographia	Cosmographia: Quit...	CMMD + Q	Exits Cosmographia, quite useful to restart it when things go wrong
Load Catalog	File: Open Catalog...	CMMD + O	After loading the main mission scenario you can keep loading catalog files (Sensor catalogs or other missions!)
Unload Catalog	File: Unload Last Catalog	CMMD + U	Unloads the latest catalog file you have loaded, such as a sensor!

## Most used commands for position control:

Action	Control	Shortcut	Utility
Select to an object	Air menu Looking glass	CMMD + F	Allows you to search for an object
Make object central	Air Menu Looking glass +	CMMD + C	The objects becomes the center
Jump to an object	Air Menu Looking glass +	CMMD + G	Allows you to jump to a certain object
Jump to Point of View around object	Camera: Jump to point of view	CMMD + V	Allows you to specify a point of view relative to the selected object

# Usage – Most used Controls 2



## Most used commands for time control:

Action	Control	Shortcut	Utility
Jump to a point in time	Time: Set Time...	CMMD + T	Allows you to input a certain time.
10x Faster	Time: 10x Faster	CMMD + L	Accelerates time rate 10 times
10x Slower	Time: 10x Slower	CMMD + K	Slows down the time rate 10 times
Reverse time	Time: Reverse	CMMD + J	Reverses time. <b>TIP: This last three controls are one to each other and they are very handy J - K - L</b>
Pause	Time: Pause	CMMD + P	Pauses time. Quite useful for fast flybys!

## Most common use cases of Cosmographia:

- Validate SPICE kernels in development (FK, IK, SPK, CK) [ESA SPICE Service]
- Assist the design of Pointings and or Observations [JUICE SOC]
- Validate a given Study mission scenario [BepiColombo SGS]
- Understand the geometry of a given mission scenario
- PR purposes

**User/Client interaction:** Feel free to send suggestions/ideas and your own catalogue files.

## Future Developments

- Periodically update the Configuration Files and the meta-kernel (that lists the relevant SPICE kernels to be used).
- Provide an integrated and one-click installation executable for a given mission or a Virtual Machine.
- Provide an interface to download the relevant kernels for a given mission scenario.
- Define a cross-mission approach for observations.

## Updates and announcements

Stay tuned for future updates and announcements. Future updates might include:

- Periodical update of meta-kernels
- Updates of catalogue files due to updates in the SPICE kernels
- Enhancements and corrections in catalogue files
- Population of observation catalogue files
- Update of the spacecraft 3D models

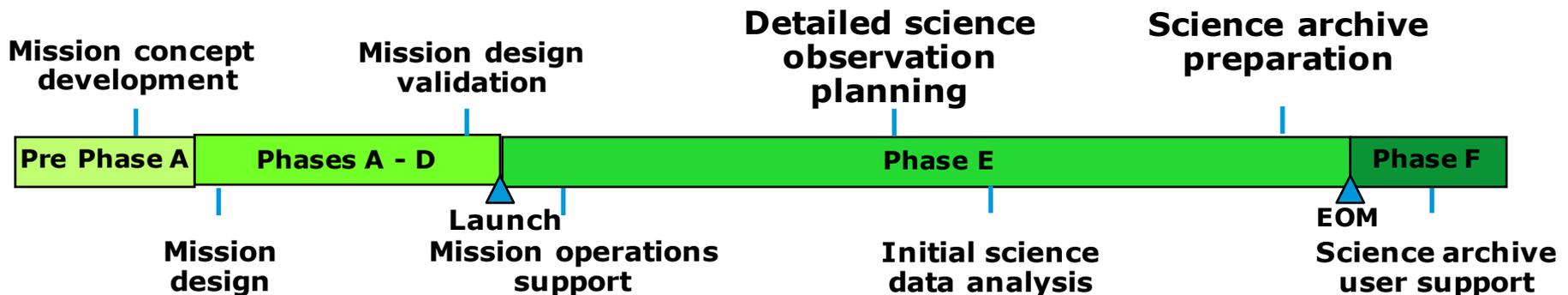
They will be made via:

MAINLY: SPICE mail list (SPICE User Group and mission specific mail list you can request to join them)

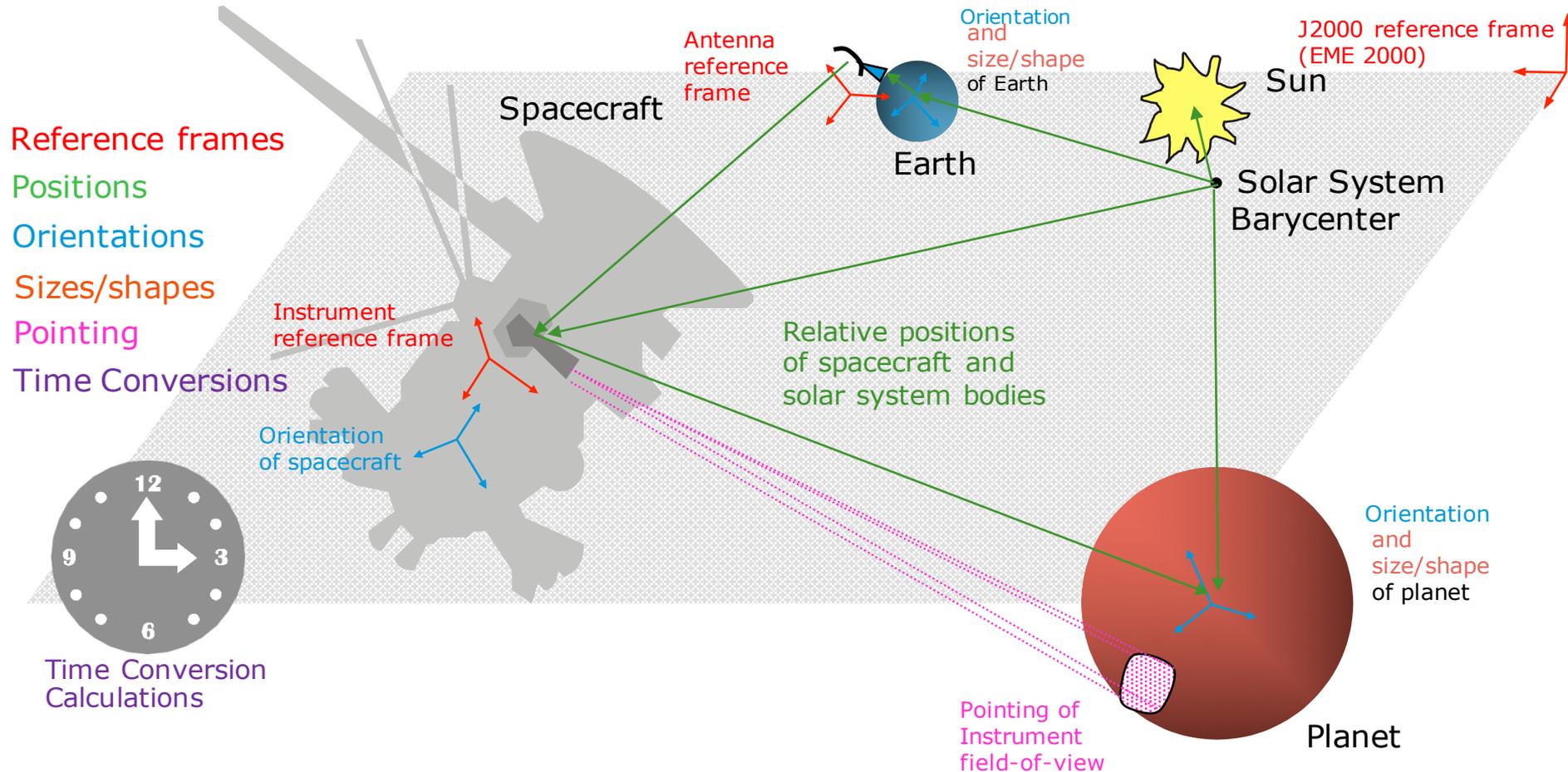
ALSO: #spice channel in OpenPlanetary Slack Community.

# BACKUP: SPICE in a nutshell - I

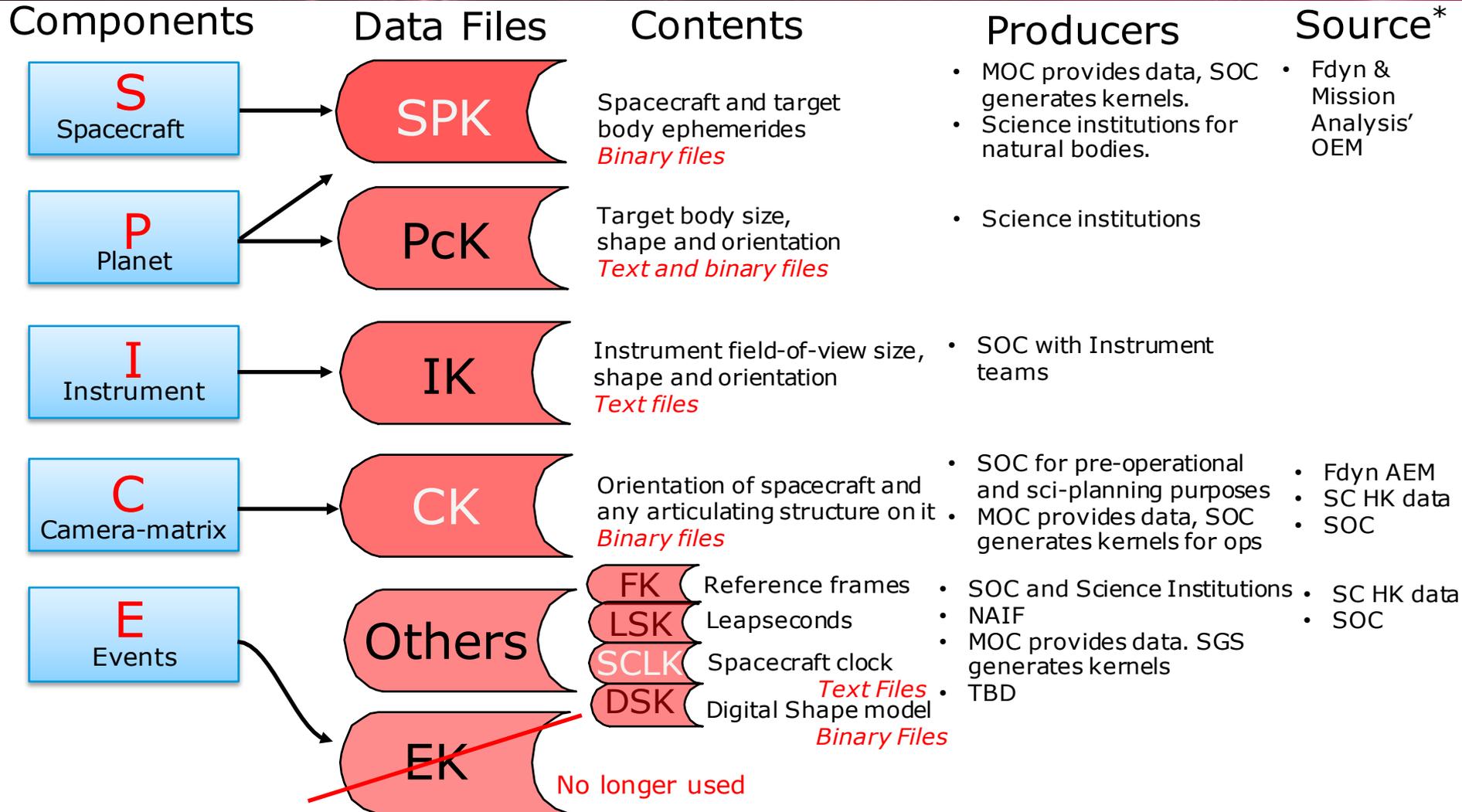
- SPICE is an information system that uses *auxiliary data* to provide Solar System geometry information to scientists and engineers for planetary missions in order to plan and analyze scientific observations from space-born instruments.
- When we talk about “ancillary data” we talk, *minimum* of spacecraft trajectory and orientation.
- SPICE was originally developed and maintained by the Navigation and Ancillary Information Facility (NAIF) team of the Jet Propulsion Laboratory (NASA).
  - **SPICE** provides users a large suite of SW used to read SPICE ancillary data files to compute observation geometry.
  - **SPICE** is used to organize and package these data in a collection of files called “kernels”
  - **SPICE** includes SW for writing, reading kernels and computing observation geometry from kernels



# BACKUP: SPICE in a nutshell - II



# BACKUP: SPICE in a nutshell - III



\*For binary data. Text files are manually produced.