Nanospace, an open source tool to help concurrent engineering teaming in cubesat preliminary design

Thibault Gateau, Lucien Senaneuch

OSCW 2020 - December 11, 2020

Phase 0/A: Preliminary Design

Specialized skills

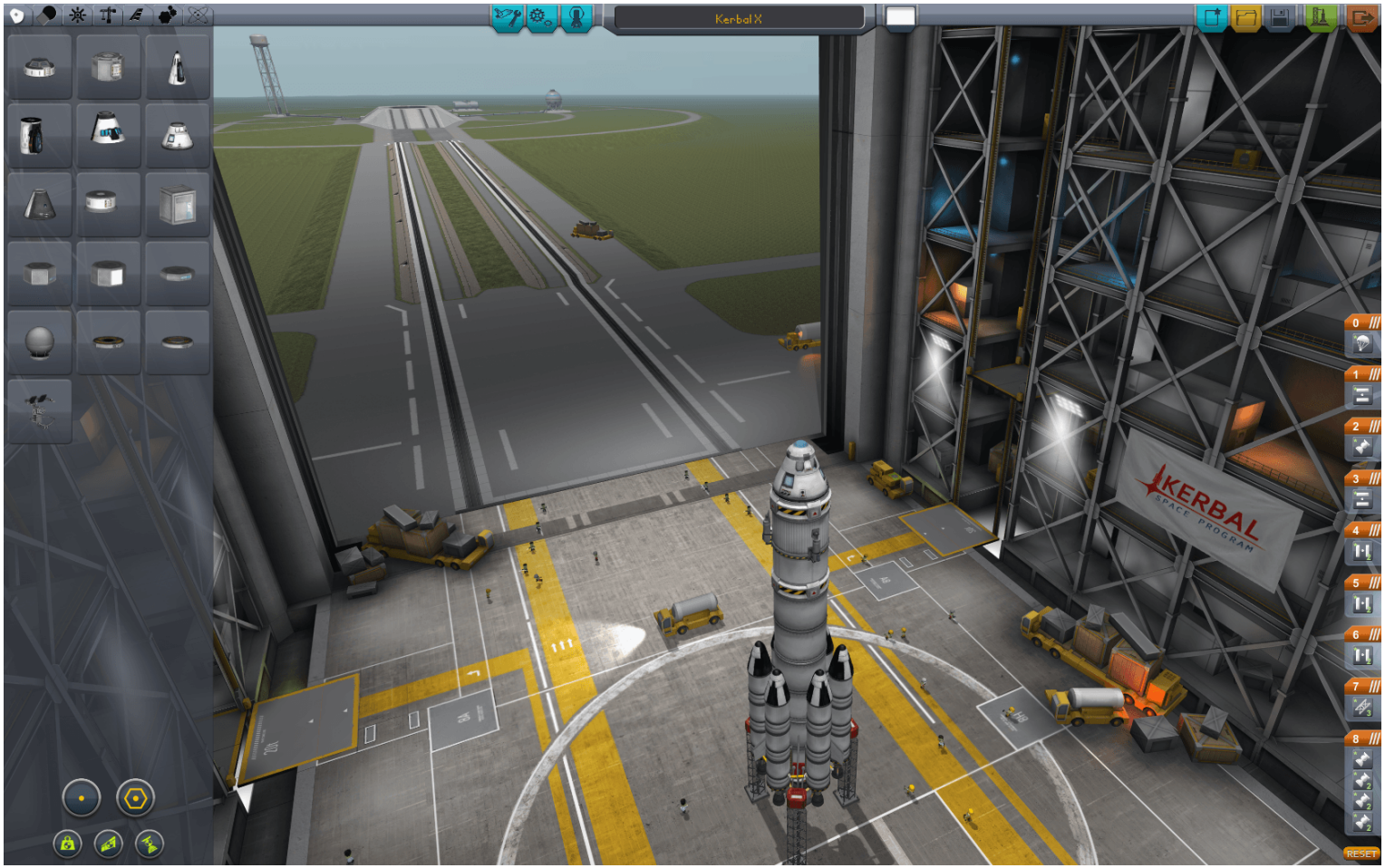
- Sub-systems budgets
- Sub-systems architectures

Team work

- Communication
- Management

Eyesat synaptic view

Expectations *vs* reality



https://www.kerbalspaceprogram.com/ - January 2018

Expectations *vs* reality

Spacecraft mission "NSS TEST CUBESAT"

Preliminary implementation of the cubesat 1U mission with the basic sensor as a payload without a propulsion system.

Mission start date	20/06/2020	
Mission end date	20/07/2020	
Operation time	30,00	days

Mission Launch Segment

Launch date	20/06/2020
Launch vehicle name	Soyuz
Launch site name	Baikonur

Spacecraft Name

Payload

Type	sensor	
Required power	10	W
Generated data	0	Mbit
Weight	0,2	kg

Construction

Structure type	1U	
Spacecraft height	100	mm
Spacecraft width	100	mm
Spacecraft depth	114	mm
Structure mass	0,1	kg
Total mass	1,600	kg
Ixx	0,00307	kg.m^2
Iyy	0,00267	kg.m^2
Izz	0,00307	kg.m^2

Mass Budget

	Without margin, kg	Margin, %	Margin, kg	Including margin, kg	% of total
Structure	0,1	20 %	0,020	0,120	6,25 %
Power	0,658	13,61 %	0,090	0,748	41,13 %
Communication	0,1095	5 %	0,005	0,115	6,84 %
OBC	0,075	10 %	0,008	0,083	4,69 %

Flight profile

Orbit type	LEO	
Argument of Perigee (w):	180	deg
RAAN (Omega):	7,13482	deg
Mean Anomaly (M):	0	deg
Period:	5428,14	sec
On-time per orbit	480	sec

Communication

Communication system

Product name	NanoCom AX100	
Weight	0,0245	kg
Energy consumption	3,2	W
Elevation Angle (d):	5	deg
Slant Range (S):	1491,67	km
Uplink Frequency Choice:	145,8	MHz
Downlink Frequency Choice:	437,45	MHz
Data rate	0,0012	Mbps

Data budget

Sybsystem	Telemetry type	Number	Sample rate, Hz	Word size, bit	transmission speed,
Power supply system	Voltage	3	0,02	16	0,96
	Current	3	0,02	8	0,48
Motherboard	Status	1	0,02	16	0,32
	Time	1	4	32	128
ADCS	Accelerometer	3	4	10	120
	Magnetometer	3	4	16	192
	Gyroscope	3	4	16	192
	Light sensor, RGBW	4x6	4	8	768

Power budget

System mode - NOMINAL	Without margin, W	Margin, %	Including margin, W
Power	0,2	10 %	0,22
Communication	3	10 %	3,3
OBC	3,2	10 %	3,52

Motion & orientation control

On-board computer

Product name	Nanodock DMC 3	
Weight	0,051	kg
Power consumption	-	W

Sensor

Sensor type	magnetomerter	
Product name	HMR2300	
Weight	0,098	kg
Power consumption	0,45	W

ADCS actuators

Type	Reaction wheel	
Product name	NanoTorque GSW-600	
Weight	0,18	kg
Power consumption	2,5	W
Max Torque	0,019	N.m.s
Max Angular Momentum	0,002	N.m

Propulsion system

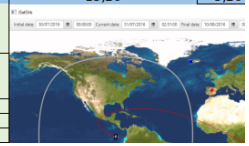
Type	x
Product name	x
Weight	0 kg
Power consumption	0 W

System dissipa

	Withou margir W
Power	x
Payload	x
Total dissipated power without any margin	x
Total dissipated power without system margins	x
Total dissipated system power margin	x
Total dissipated power including system margins	x

Link budget

Type	UHF/VHF	S-B
Orbit alttitude	0	
C_N0_UP	82,56	
Eb_N0_UP	42,74	32,74
margin_UP	17,54	7,54
C_N0_Down	57,65	47,65
Eb_N0_Down	26,86	16,86
margin_Down	15,26	5,26
Alternative	15,26	5,26



Preliminary 1U design by Anton Poltoradnev - ISAE-Supaero

Software Tools and libraries

Candidate Software List Specification for Nanostar

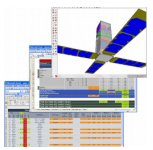
Power budget :
- IDM-CIC
- Home made scripts

Structure :
- IDM-CIC
- Catia [X]

Mission analysis :
- GMAT
- Satorb
- DOCKS
- libs :
  Celestlab
  Poliastro
  Orekit

Visualization Tools :
- Home made
- VTS
- Celestia
- Cosmographia

Dynamic database :
- IDM-CIC (ECSS Standard)
- Valispace database
- Home made database



LOS :
- Stella

Thermal :
- Systema thermica
- Home made scripts

Link Budget :
- AMSAT
- Home made

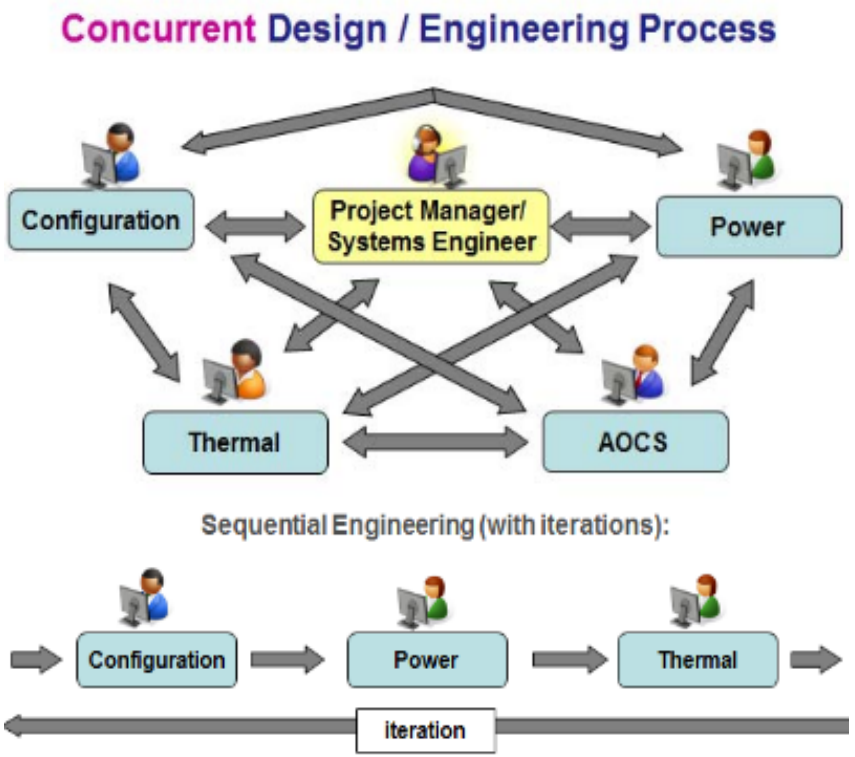
Trendy way: Concurrent Design Engineering

Efficient data management and exchanges

- create data models
- shared common models
- data update should be propagated in each expert tool

Strategy?

- take an existing CDE?
- adapt from libs?
- adapt from MBSE?



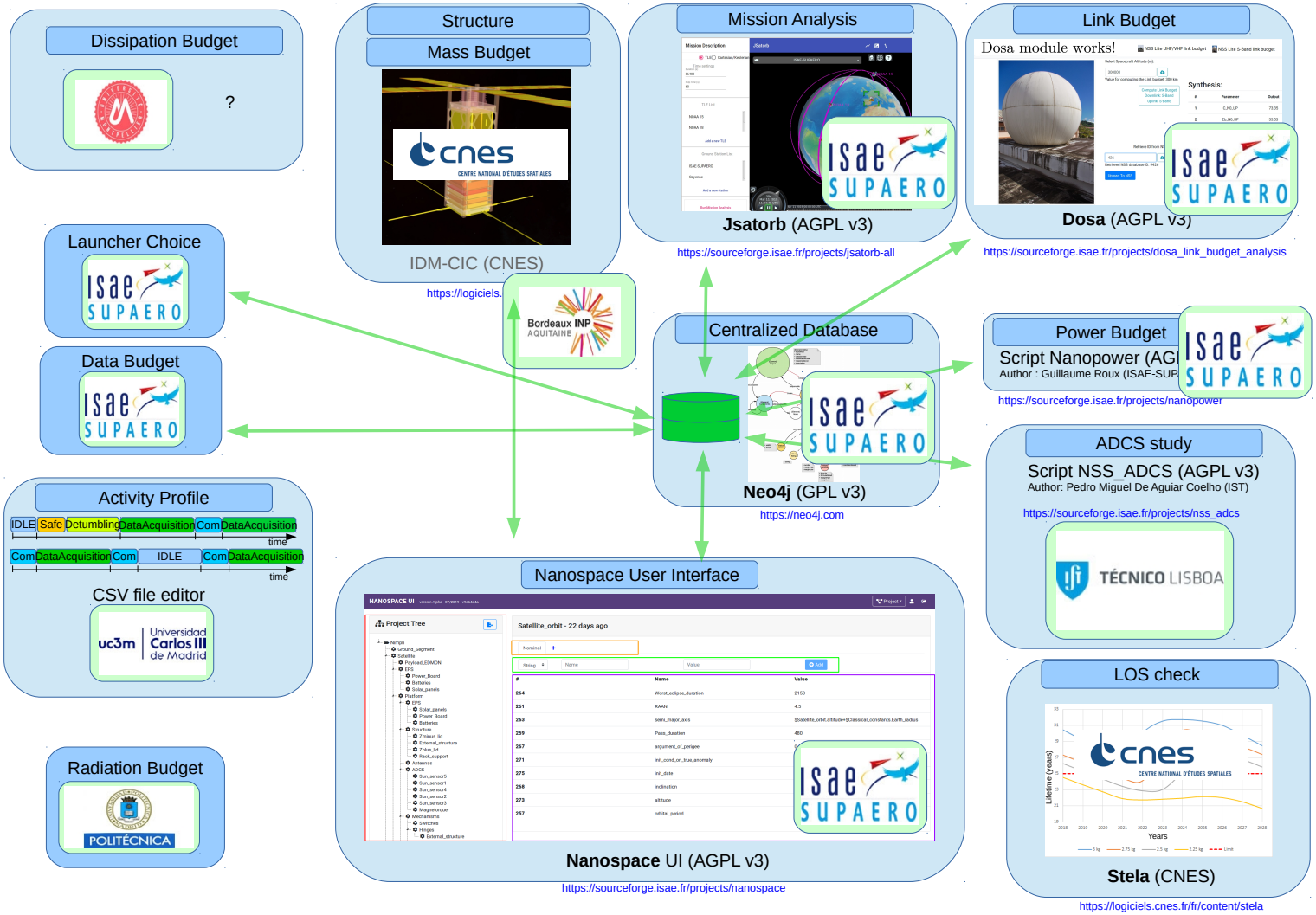
[Di Domizio and Gaudenzi, 2008]

Review [Knoll et al., 2018]

Goal: Concurrent design engineering with some requirements

- Student challenges on cubesats preliminary design
- Multiple access from multiple sites (5 institutions)
- Database rather than spread-sheets [Gordon, 1999]
- Allow users to keep their favorite expert tools
- Student friendly & **Open-source**

NSS: Nanostar Software Suite



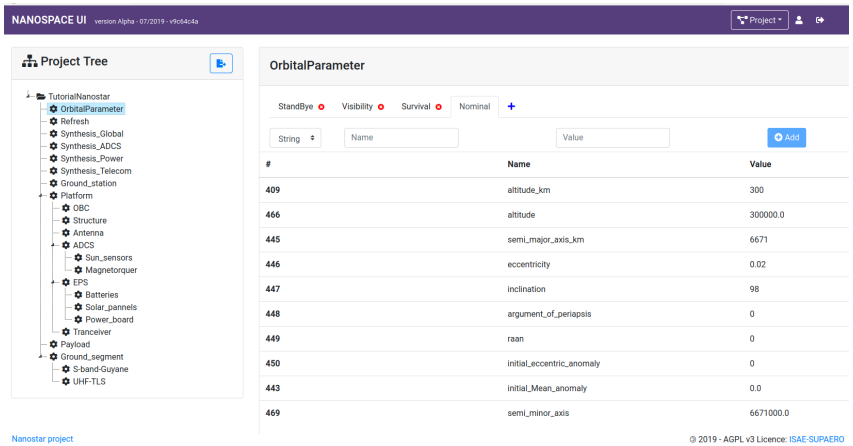
1. Nanospace - Demo
   - Demo: First connection
   - Nanospace-UI
   - Demo: First interactions

2. Nanospace - Architecture
   - Technical choices
   - Running Architecture
   - Interaction with third party applications

Demo: First connection

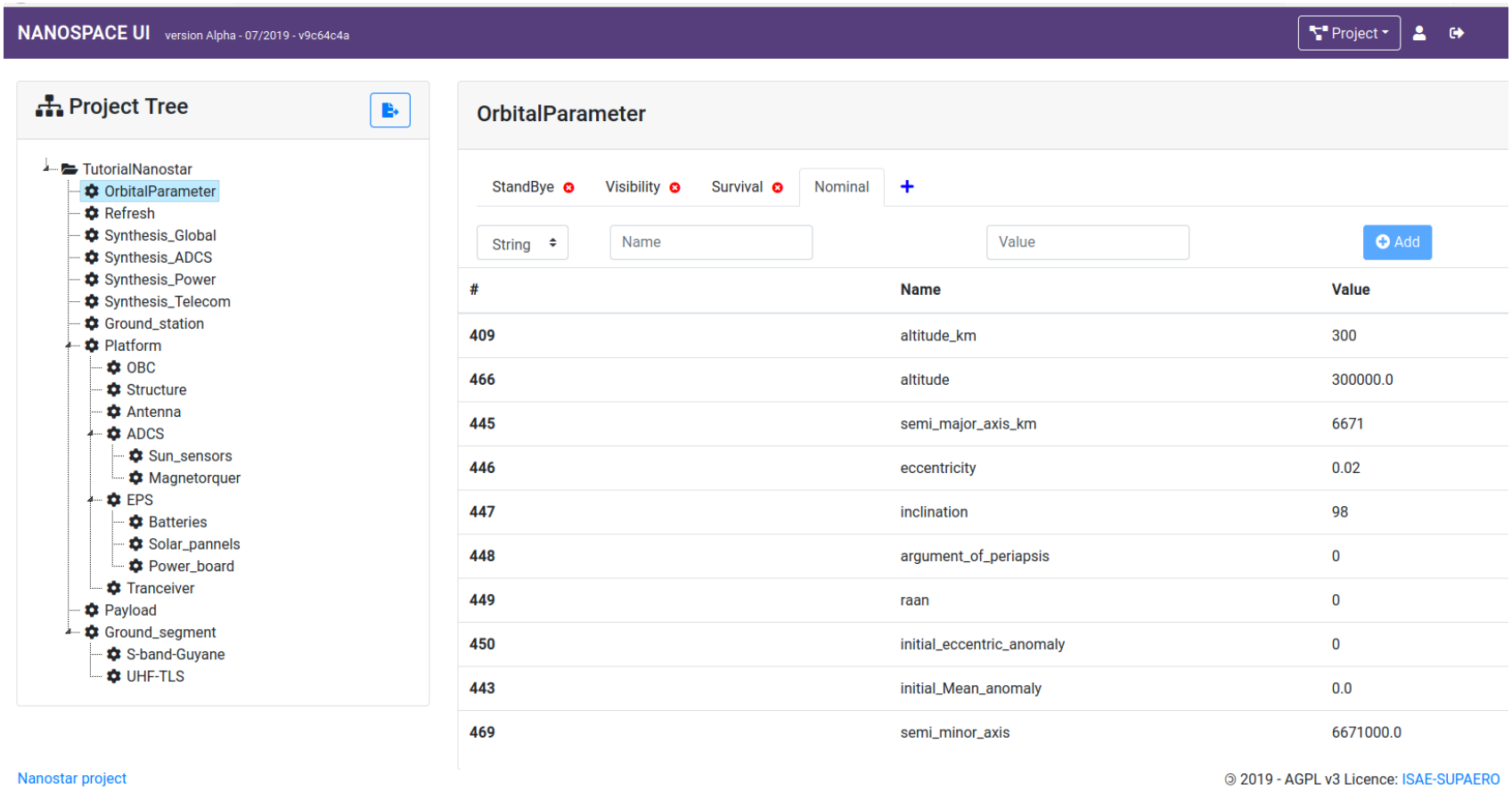
Using your favorite Web Browser

- Go to `https://dcas-nanostar.isae.fr`
- You can **Subscribe** and **Login**
- Import an example project json project file: `https://gitlab.isae-supaero.fr/nanostar/nanospace/nanospace-user`



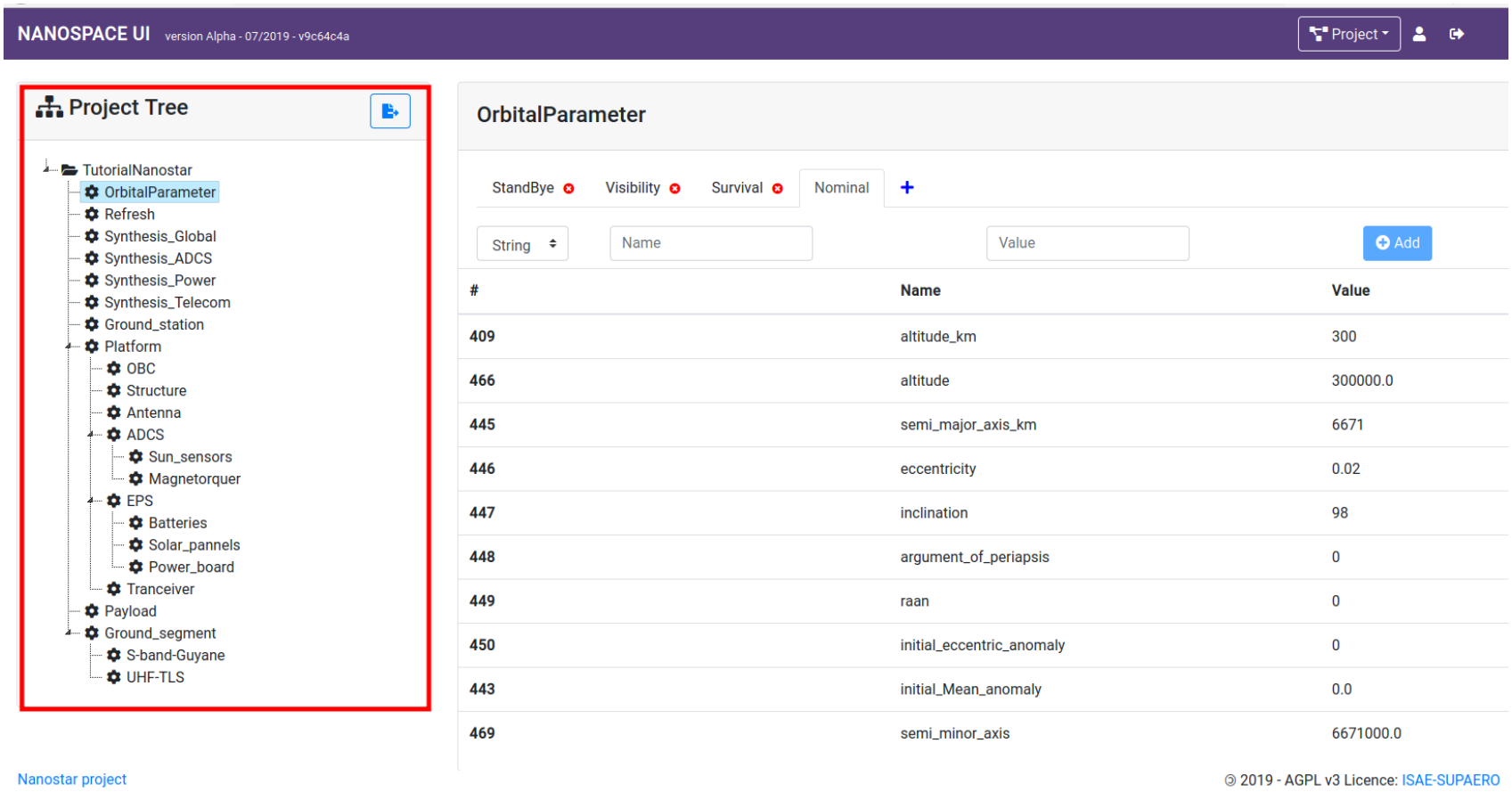
Interreg Sudoe
EUROPEAN UNION
NANOSTAR
European Regional Development Fund

Nanospace-UI



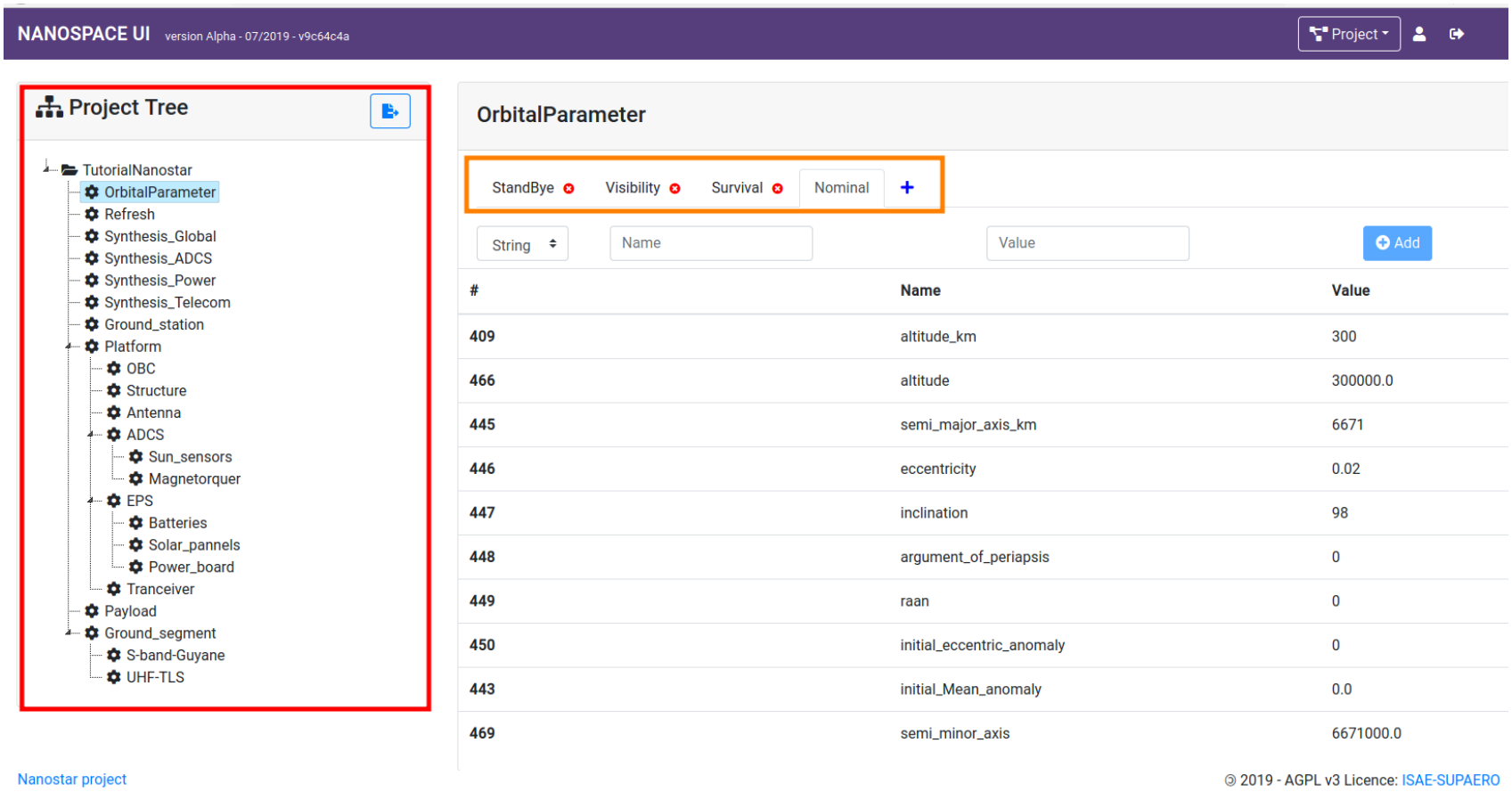
View of a project in a browser

First interactions



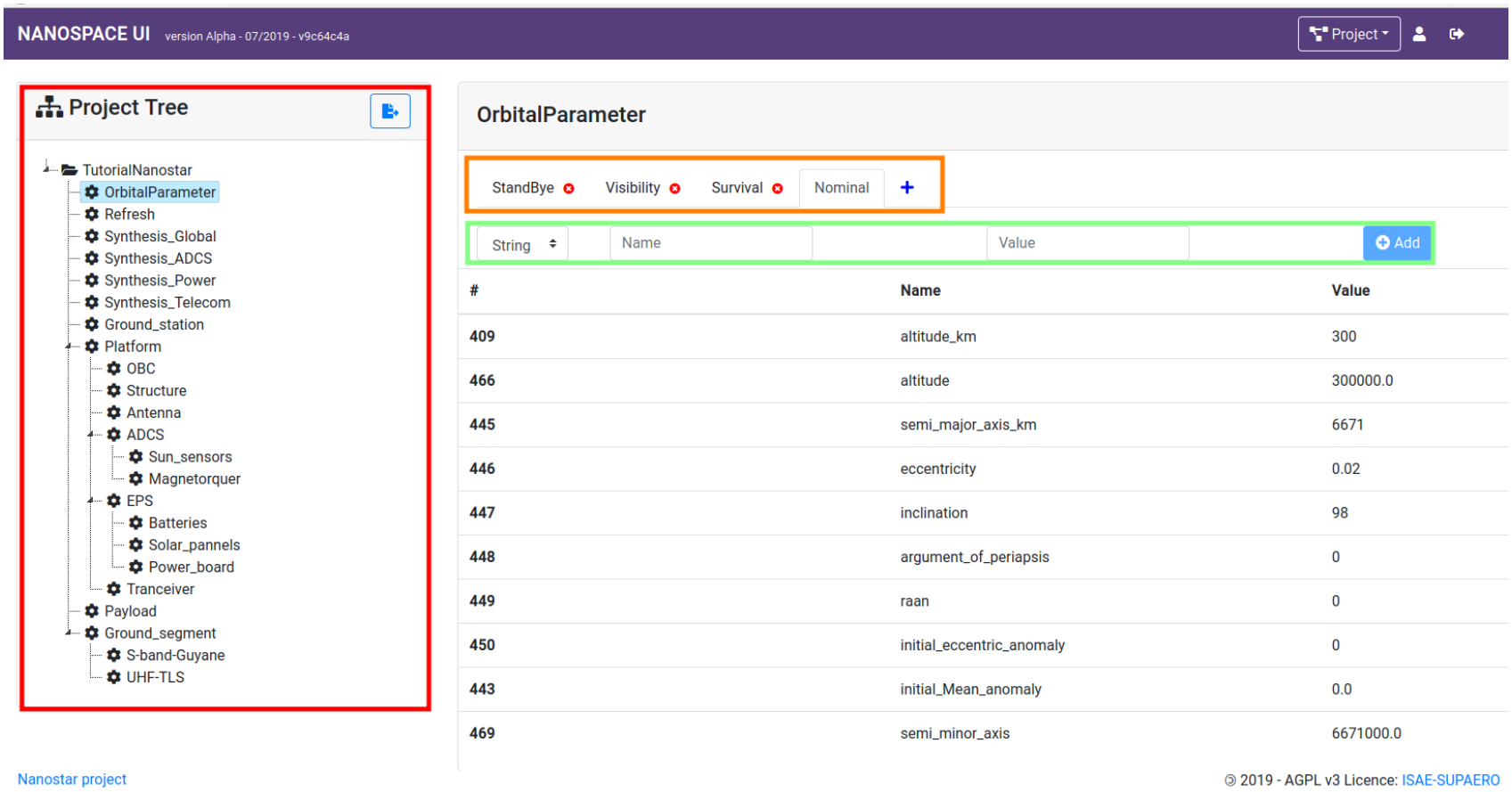
View of a project in a browser

First interactions



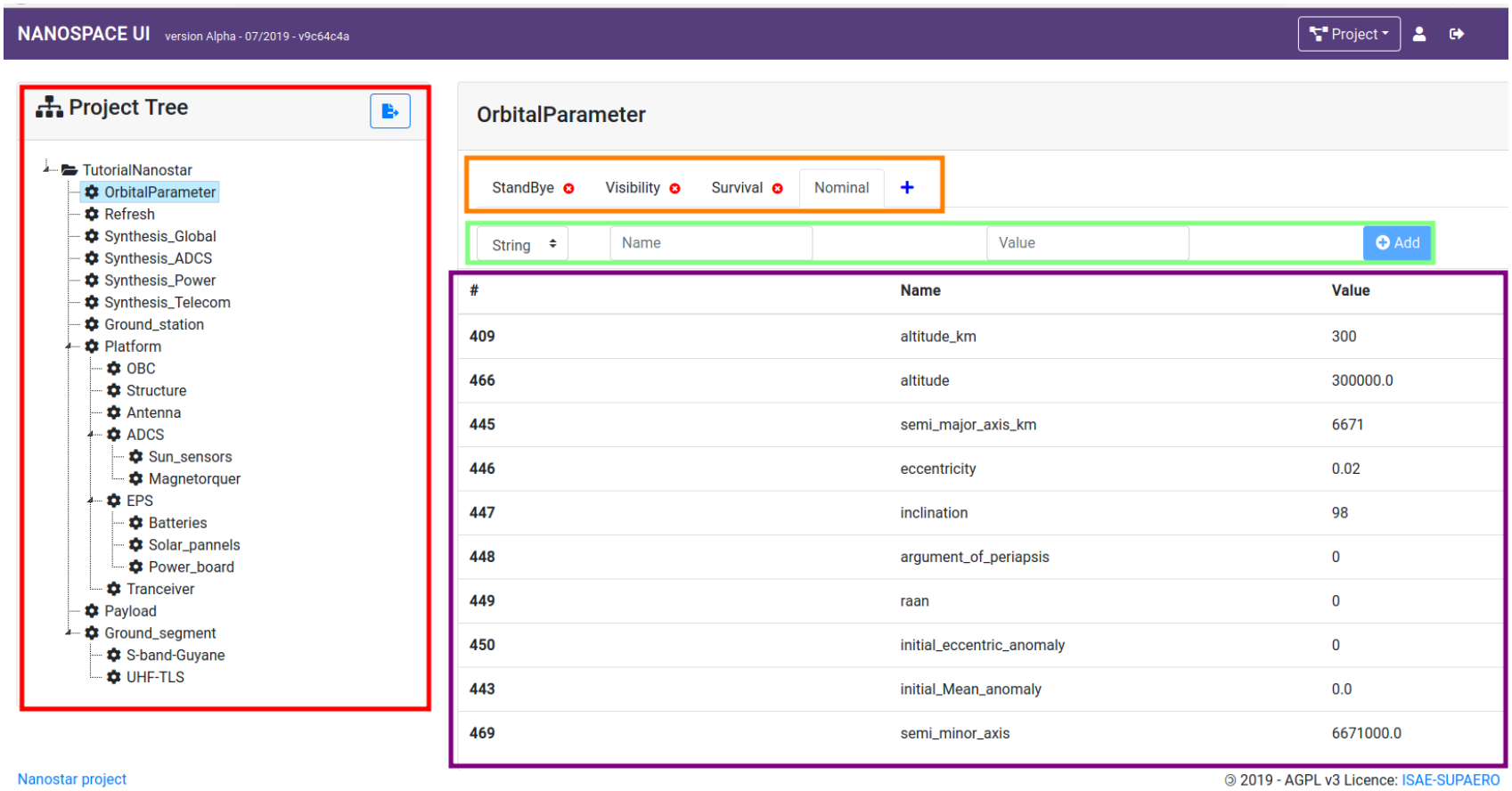
View of a project in a browser

First interactions



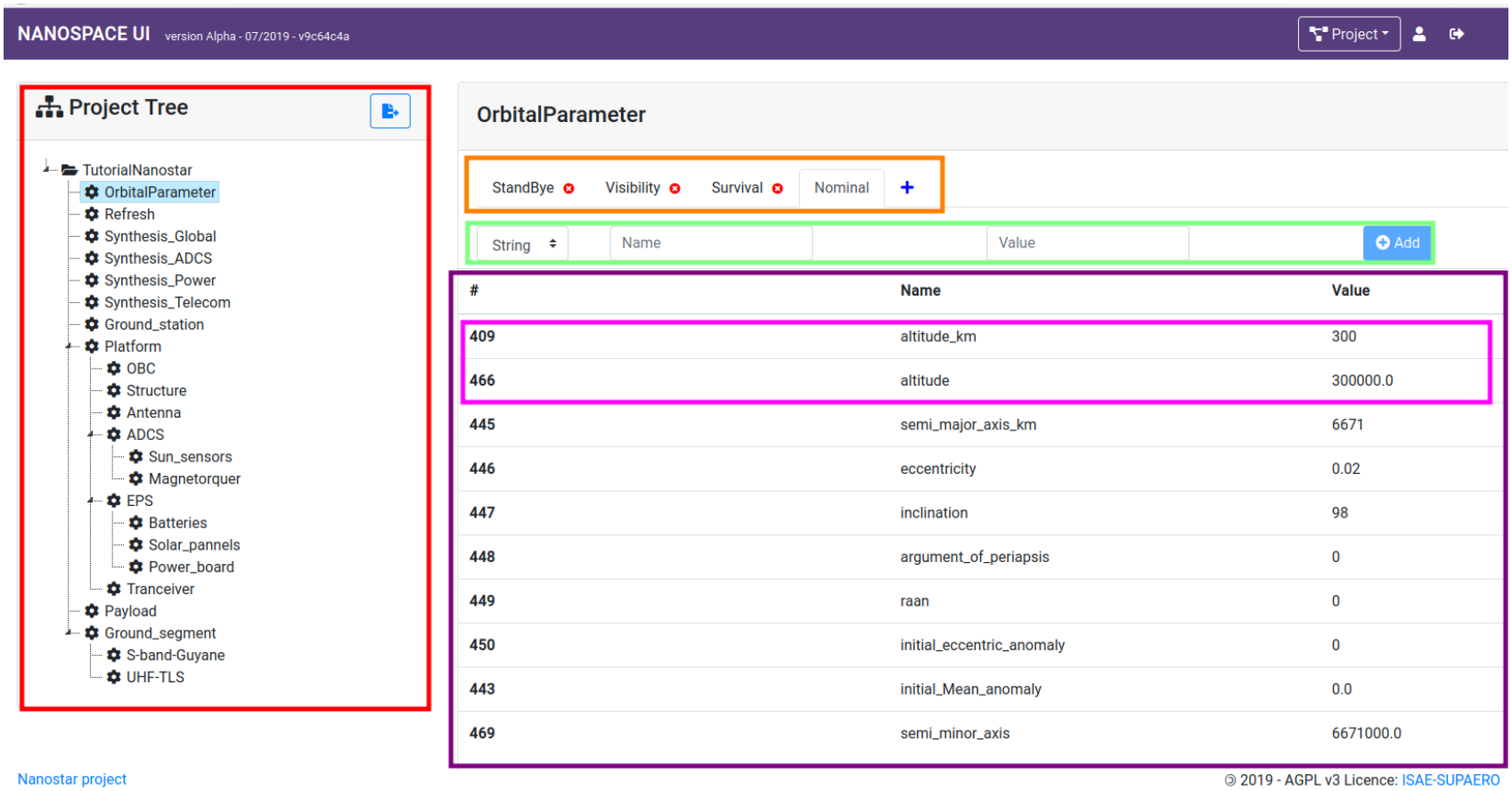
View of a project in a browser

First interactions



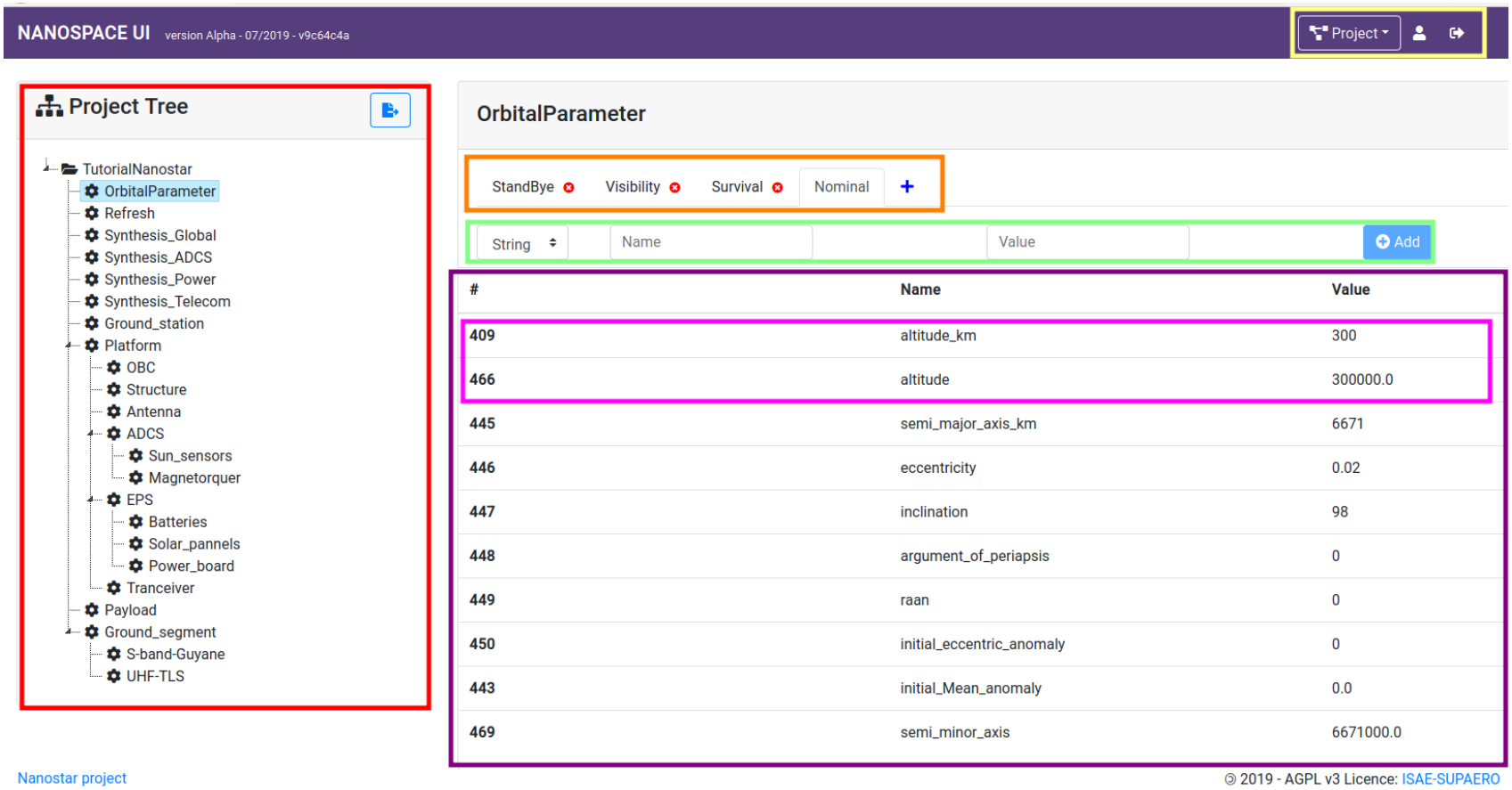
View of a project in a browser

First interactions



View of a project in a browser

First interactions



View of a project in a browser

1. Nanospace - Demo

2. Nanospace - Architecture
   - Technical choices
   - Running Architecture
   - Interaction with third party applications

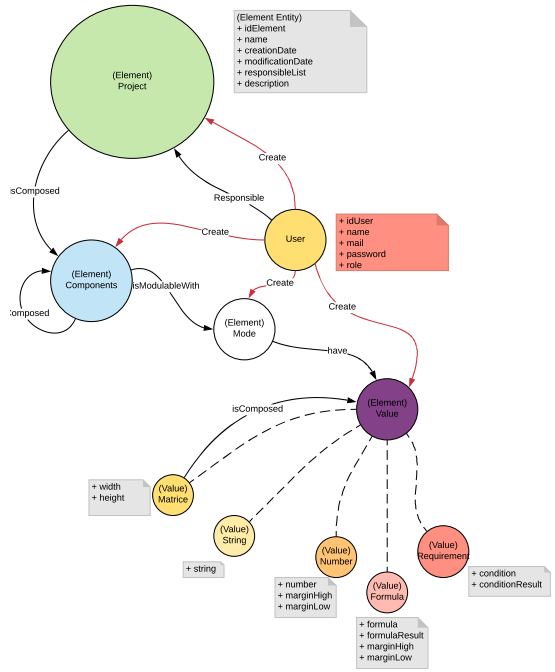
Requirements

- Web App
  <platform independent>
- REST API
  <ease third party connection>
- ACID[1] property
  <concurrent access>

Technical choices

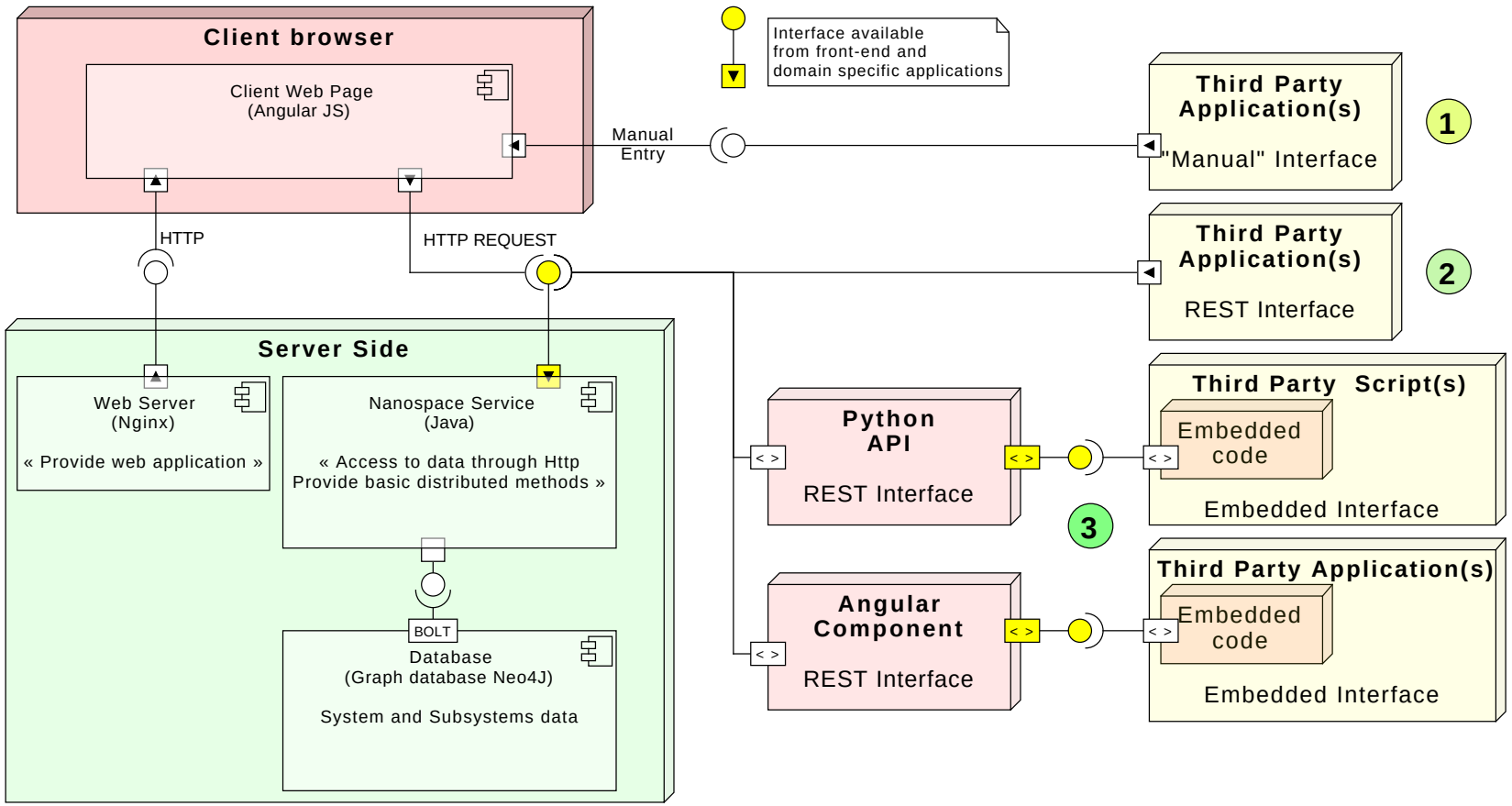
- Database: Neo4j
- Front-end: Angular
- Back-end: Spring Boot
  (Neo4j direct compatibility)

1: Atomicity, Consistency, Isolation, Durability



Database model prototype

Running Architecture



REST interface

Easy to run your own scripts

- Direct connection to the REST interface
- Python API provided (nanospace.py)
- You can use your favorite libs (poliastro, orekit, celestlab...)

Python example

```
2 from nanospace import Nanospace
3 nanospace = Nanospace(srvAddr, usr, pw)
4 altitude =getNanospaceString(nanospace,altitudeID)
5 d = getMarginDown(altitudeKm,dataJsonFile)
6 EB_N0 =  round(d['Eb_N0_Down'],1)
7 nanospace.update string value(Eb N0 DownID, 'EB N0', str(EB N0)+" in [dB]")
```

Adding an Angular component

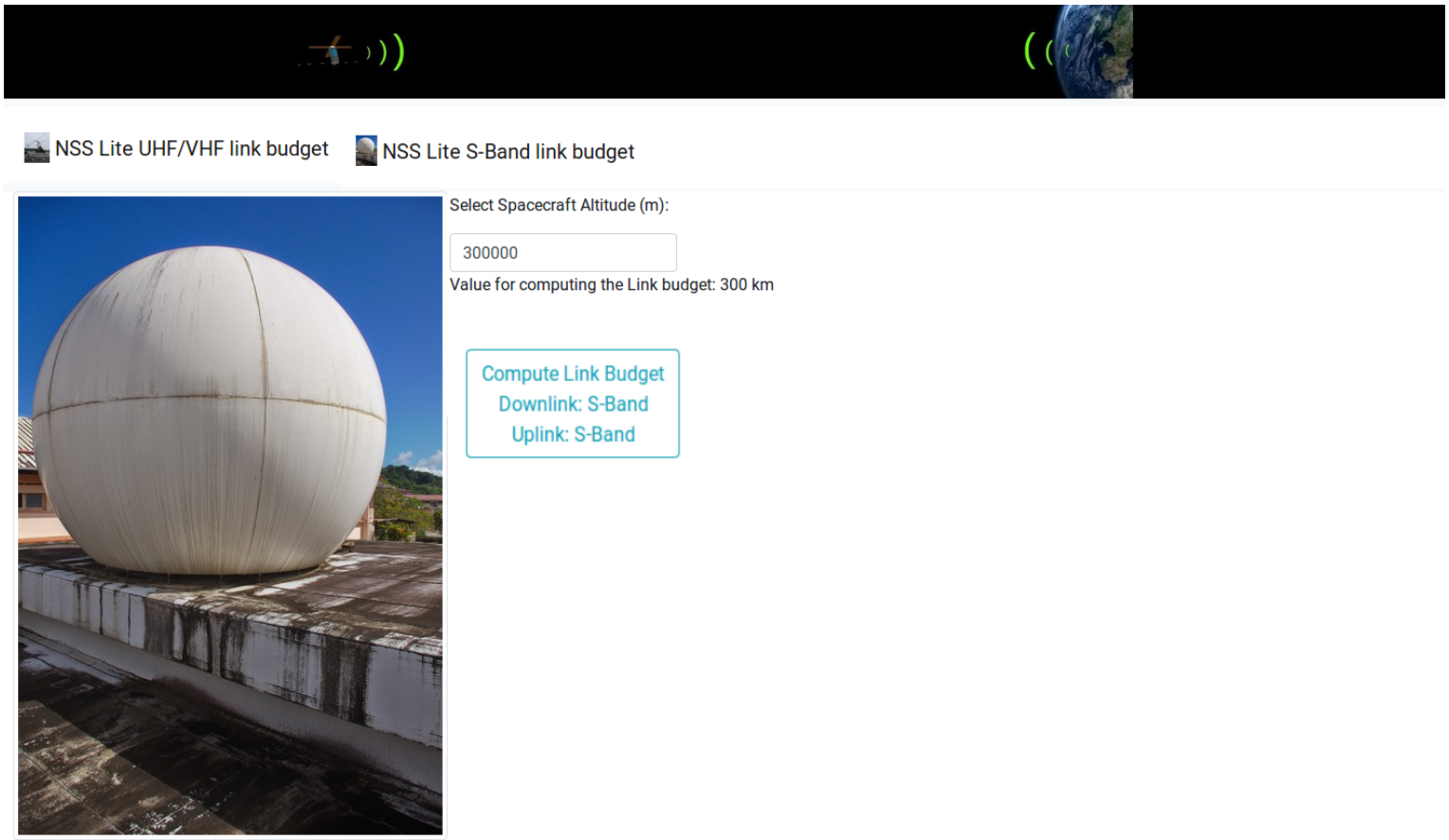
Angular component example available on-line

- Angular Component (3 lines of code)
- Available online:
  `www.npmjs.com/package/ngx-nanospace-client-lib`

Embedded component example

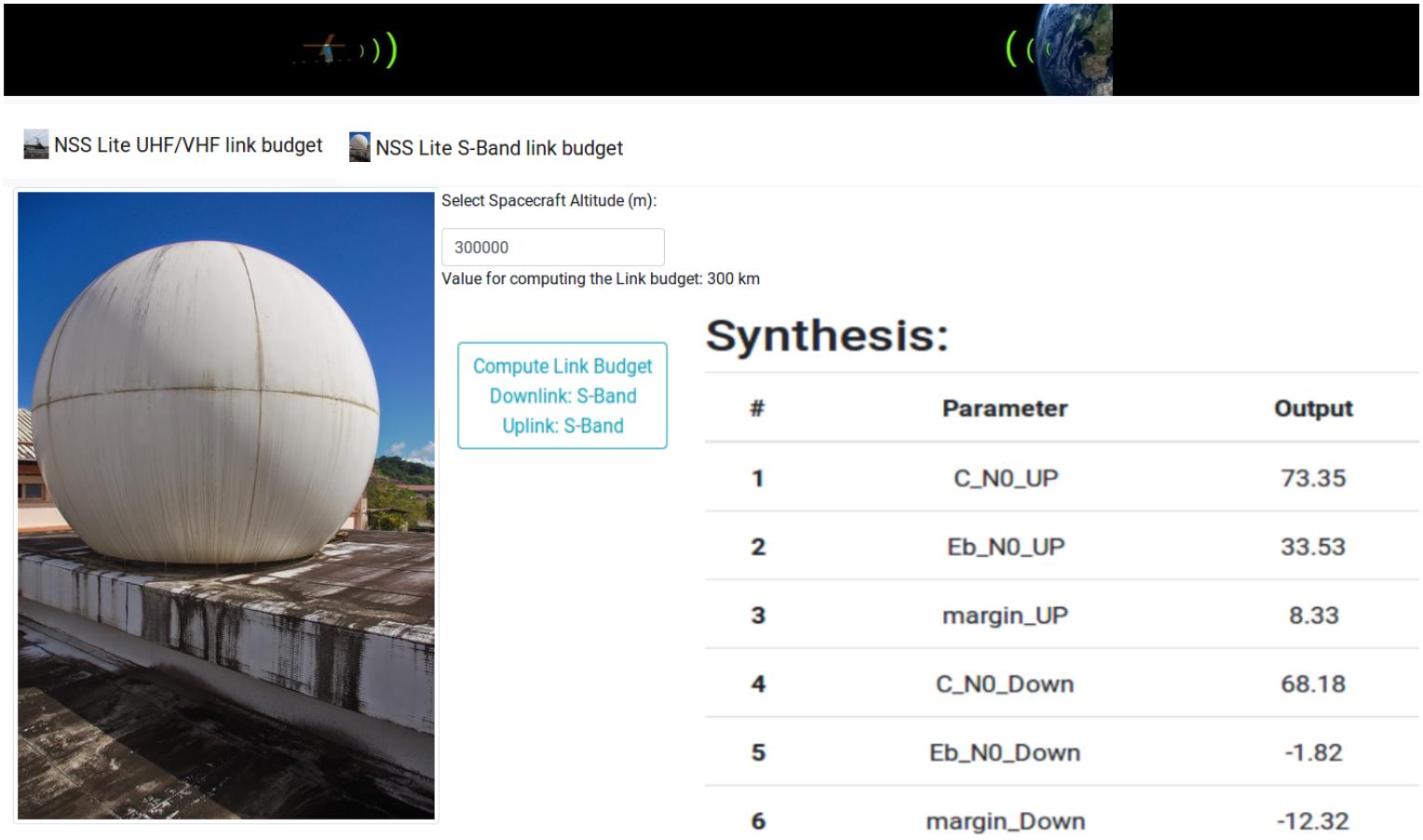
```
/*app.component.html*/
<nano-input-id [(ngModel)]="idImported"></nano-input-id>
<nano-input-value [(ngModel)]="valueImported"></nano-input-value>
<nano-import-export-value [(ngModel)]="valueImported"></nano-import-export-value>
```

Example with Angular component



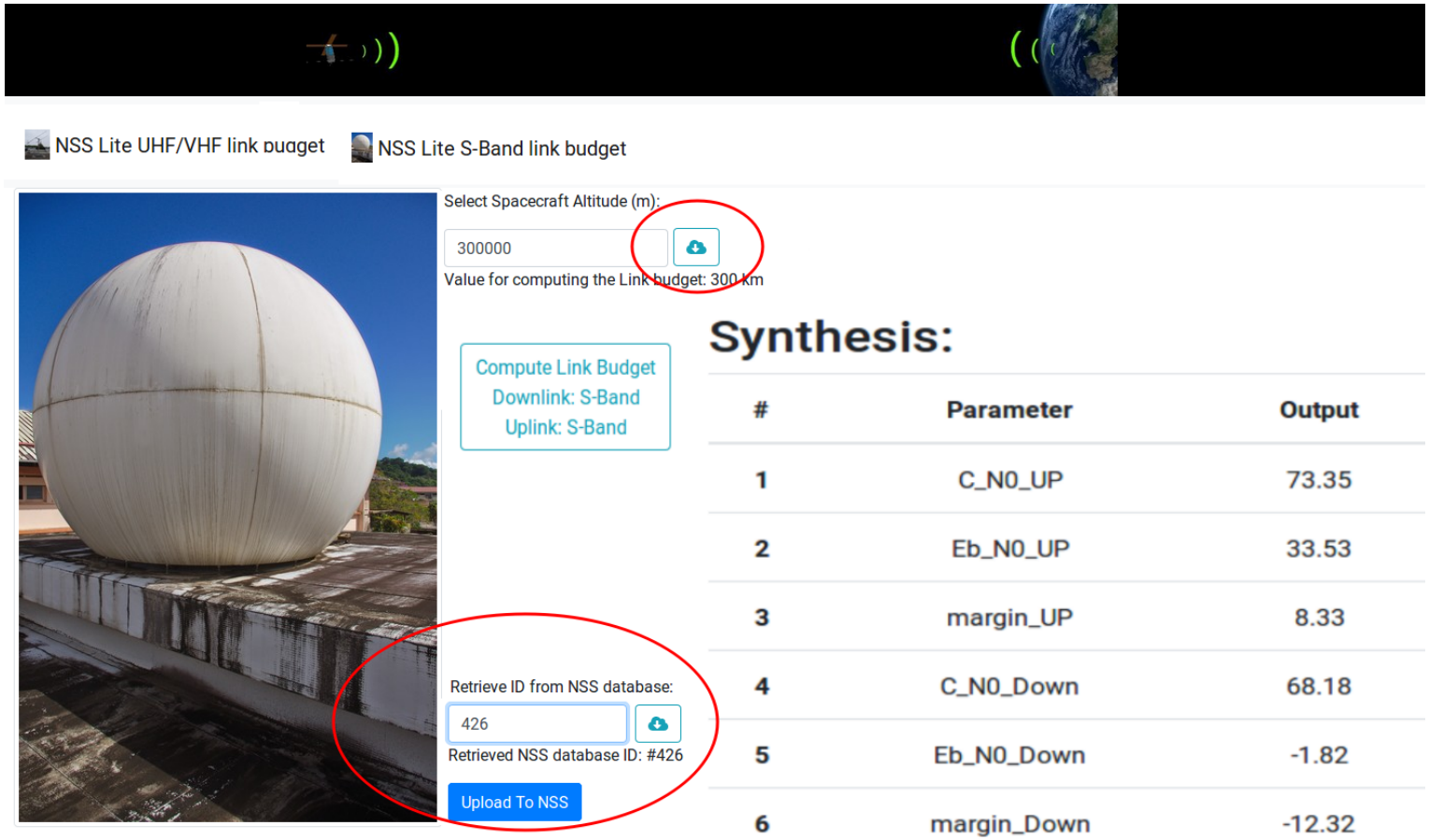
Pedagogical link budget module view

Example with Angular component



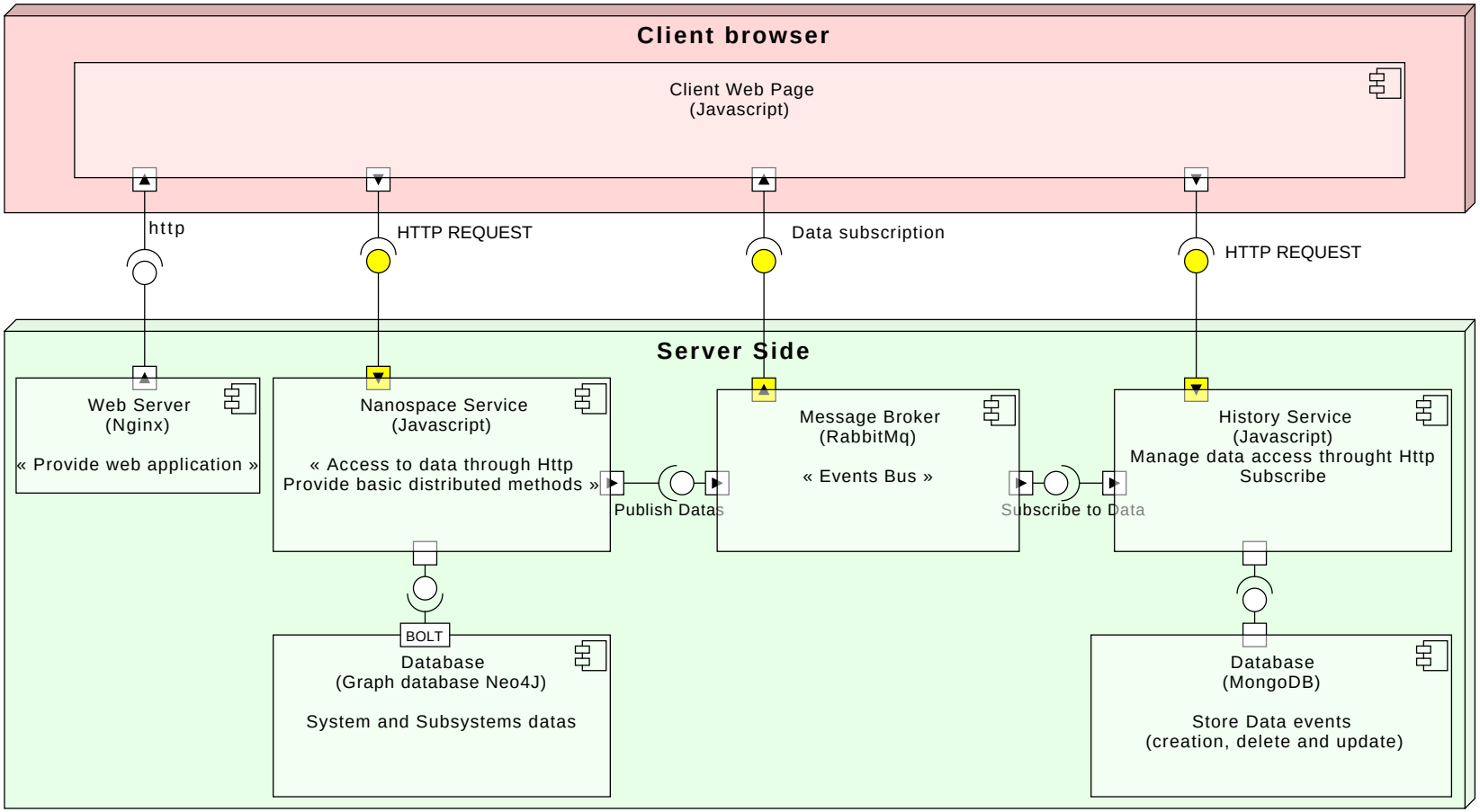
Pedagogical link budget module view

Example with Angular component



Pedagogical link budget module view

Targeted Architecture



Take Home Message

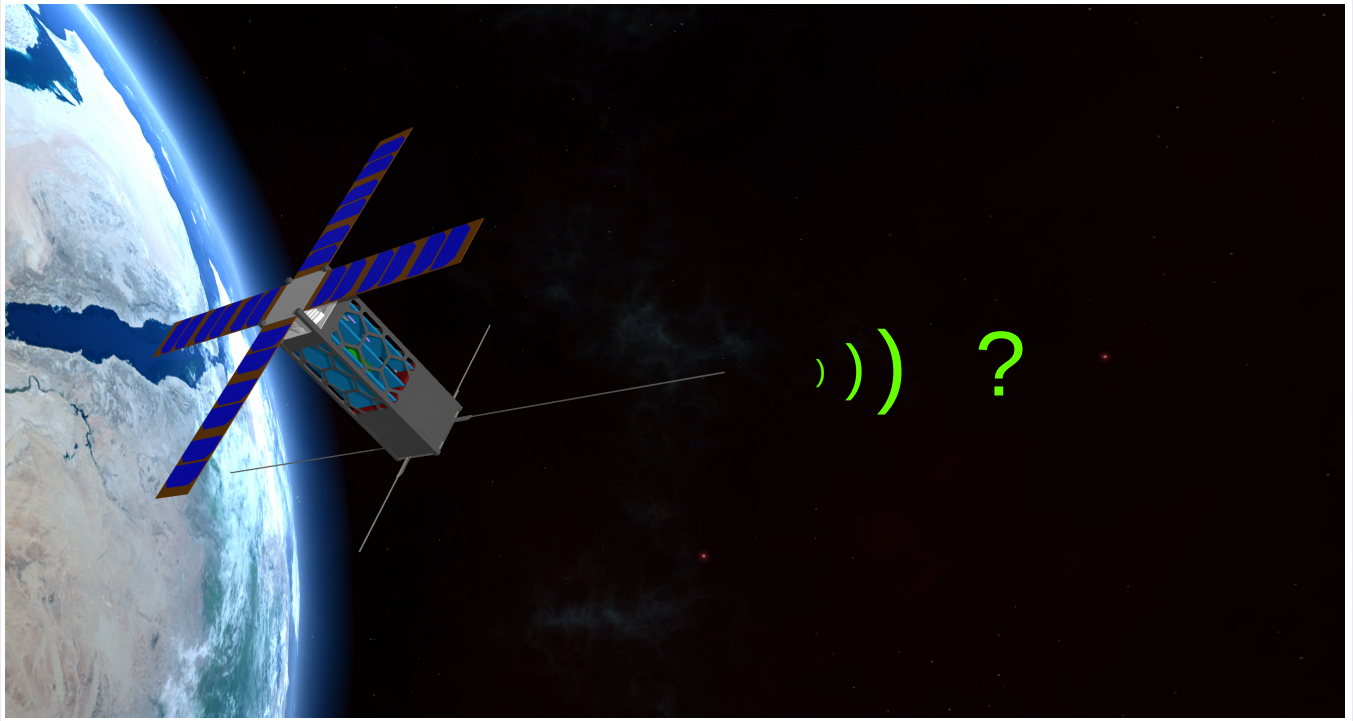
- Easy to integrate to third party application
- Concurrent access, remote-located team context
- Available source code (AGPL v3):
  `https://gitlab.isae-supaero.fr/nanostar/nanospace`
- Available test server:
  `https://dcas-nanostar.isae.fr/`
- Web-service - Docker Version

Future Works

- Event management
- Formal pipeline management, top bottom approach (MBSE?)
- Life cycle beyond phase 0/A (up to C...)
- Check resilience (interaction with DOCKS? GMAT?)

Thank you for your attention!

Any question ?



Special thanks to:

Marie-Carmen Fauré, Frédéric Fal, Maxime Syidalza, Jacques Villemur, Ludovic Bosseaux

Di Domizio, D. and Gaudenzi, P. (2008).
A model for preliminary design procedures of satellite systems.
Concurrent Engineering, 16(2):149–159.

DLR (2019).
Virtual satellite.
https://www.dlr.de/sc/en/desktopdefault.aspx/tabid-5135/8645_read-8374/.

ESA (2019).
Ocdt.
https://ocdt.esa.int.

Gordon, K. J. (1999).
Spreadsheet or database: Which makes more sense?
Journal of Computing in Higher Education, 10(2):111–116.

Knoll, D., Fortin, C., and Golkar, A. (2018).
Review of concurrent engineering design practice in the space sector: state of the art and future perspectives.
In 2018 IEEE International Systems Engineering Symposium (ISSE), pages 1–6. IEEE.

Le Gal, J.-L. and Lopes, P. R. (2016).
Idm-cic.
https://www.clever-age.com/fr/case-studies/cnes-une-application-de-modelisation-3d/.

NASA.
GMAT.
http://gmatcentral.org/.
Accessed: 2017-03-28.

Rheagroup (2019).

Cdp4.
https://www.rheagroup.com/fr/news/cdp4-open-source-community-edition.