Java multi-mission simulation framework: evolutions and improvements

Pierre BORNUAT (CS SI)
Thierry WARROT (CNES)

26/03/2015

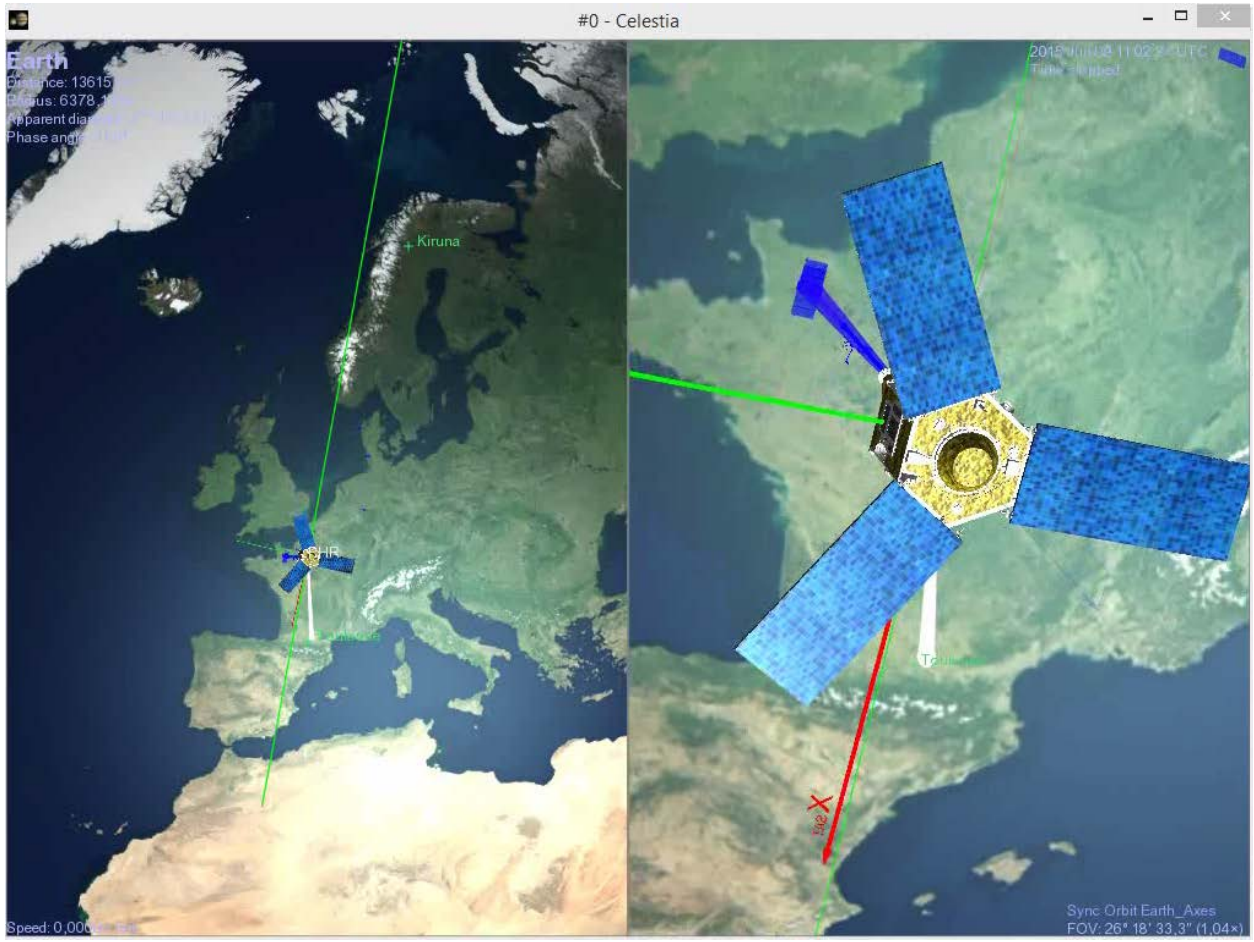
AGENDA

- Mission Simulation at CNES
- ALIS Infrastructure Characteristics
- ALIS Improvement of Computation Capacities
- ALIS Versatility: Emphasis on COTS Integration
- Conclusions

ALIS: Atelier Logiciel pour l'Ingénierie Système
Simulation Framework for Mission Engineering

Mission Simulation at CNES
(Context / Acquisitions)

Aim at elaborating the payload mission plan regarding on board capabilities, transmission channel and ground resources

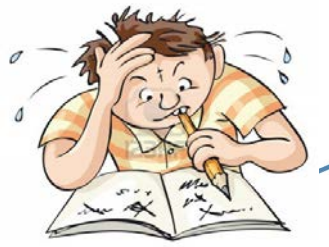


Mission Simulation at CNES

(Mission Programming engineer's task)

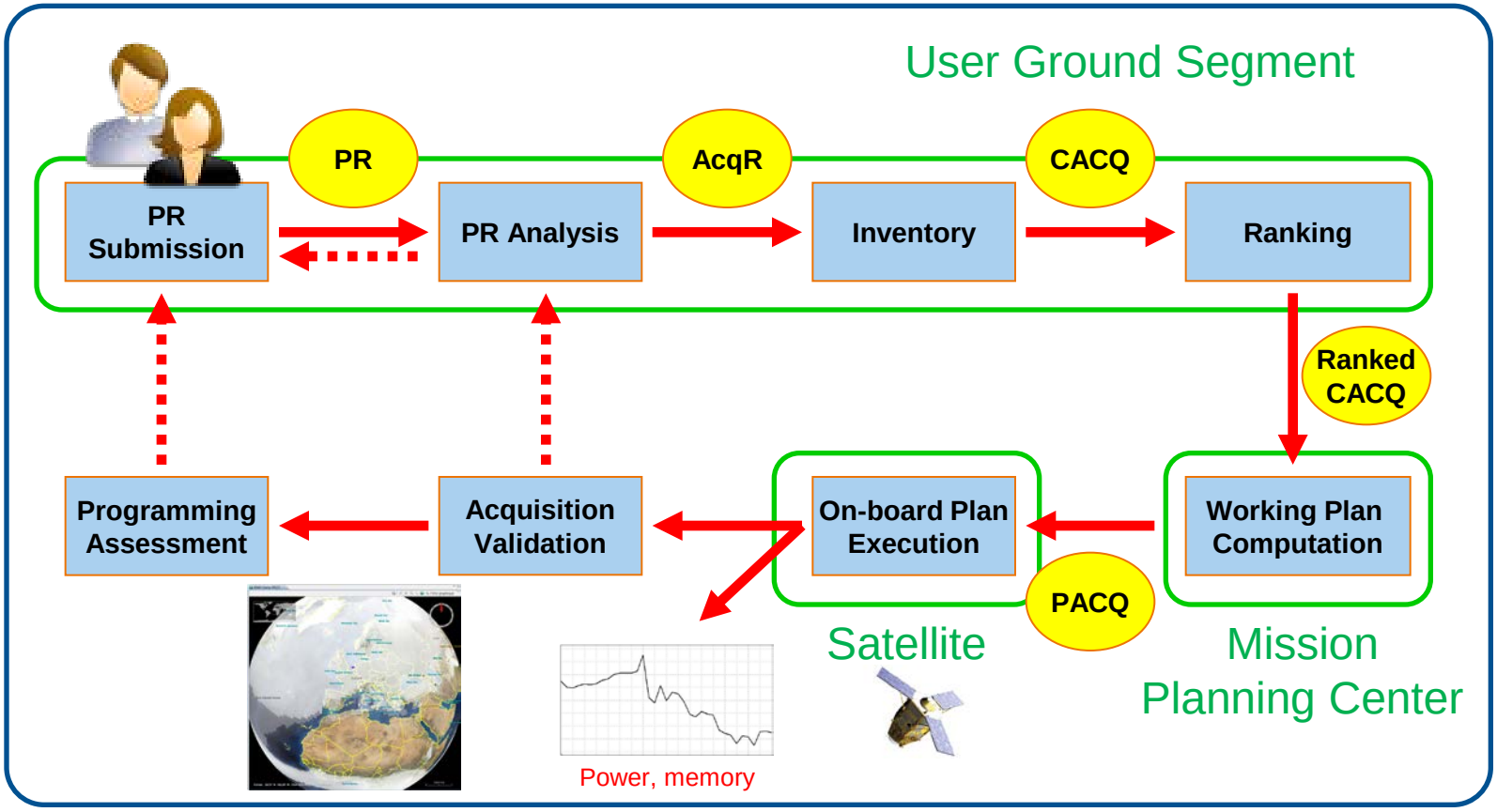
- Examples of typical problems to deal with:
  - System sizing: ground networks, satellite configuration, agility…
  - Scheduling optimization according to:
    - » Number of images to take
    - » Customer Priority Requests
  - On board Memory Capacity and electrical power consumption

  - Average delay between request deposit and user delivery
  - Amount of 3D “useful” images accessible during 6 months, according to usual climate
  - Average System capability on a region, along one orbit
  - …
- Multiple satellites, ground networks, thousands of images to acquire
  - ➔ Combination is so enormous and complexity so high that mission planning simulators are mandatory



ALIS

Mission Simulation at CNES
(Mission planning loop)



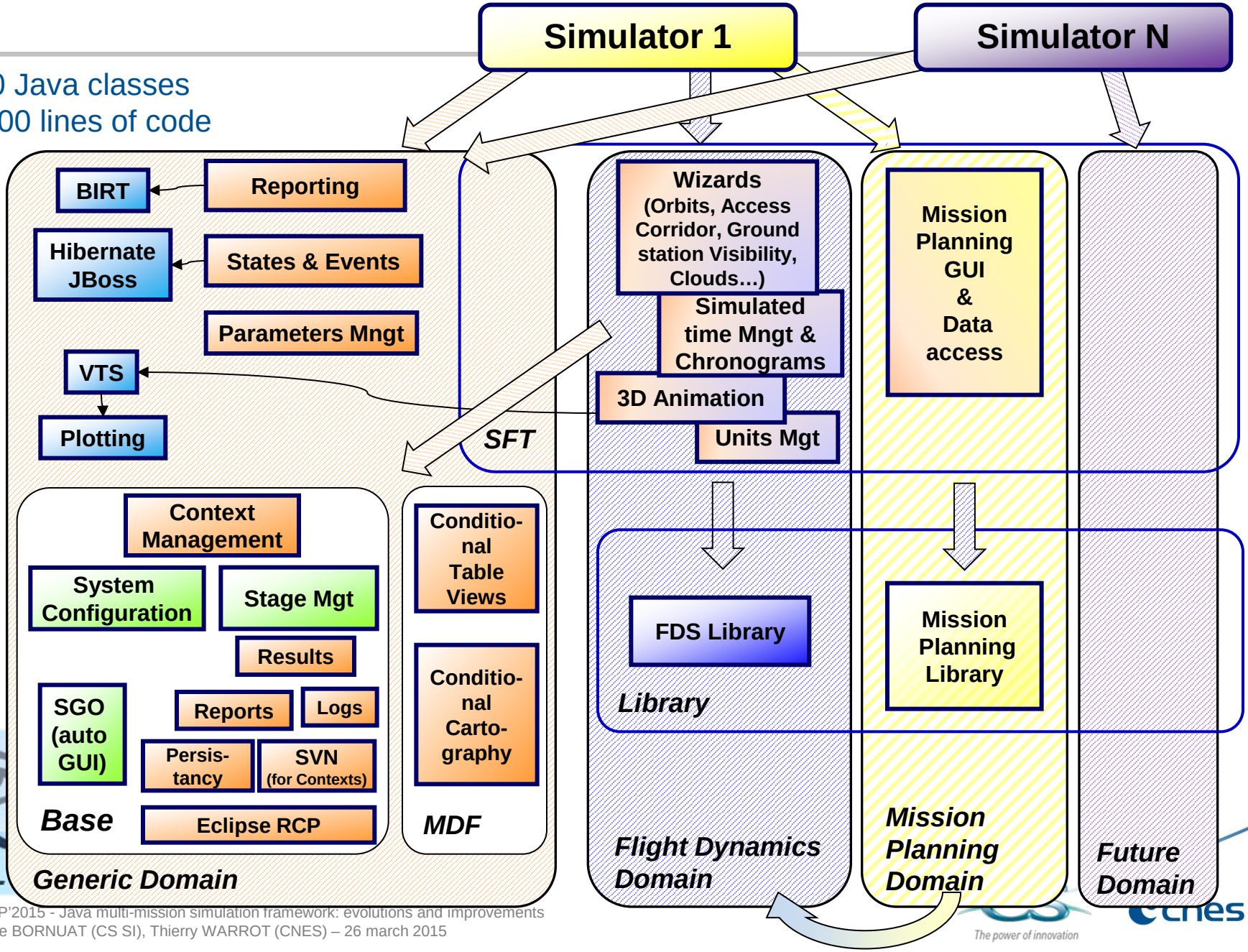
Mission Planning Simulator

PR: Programming Request
AcqR: Acquisition Request

CACQ: Candidate ACQuisition
PACQ: Planned /Programmed ACQuisition

ALIS Infrastructure Characteristics

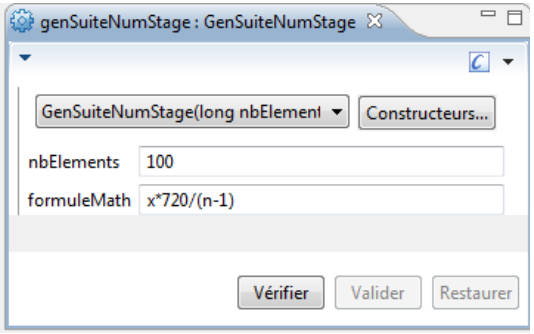
23 000 Java classes
500 000 lines of code



ALIS Infrastructure Characteristics
(ALIS versatility / SGO)

- **SGO**: *Show Generic Objet*
  - Automatic generation of Graphical User Interfaces through source code inspection

```
public class GenSuiteNumStage {
   protected long _nbElements;
   private String _formuleMath;
...
```

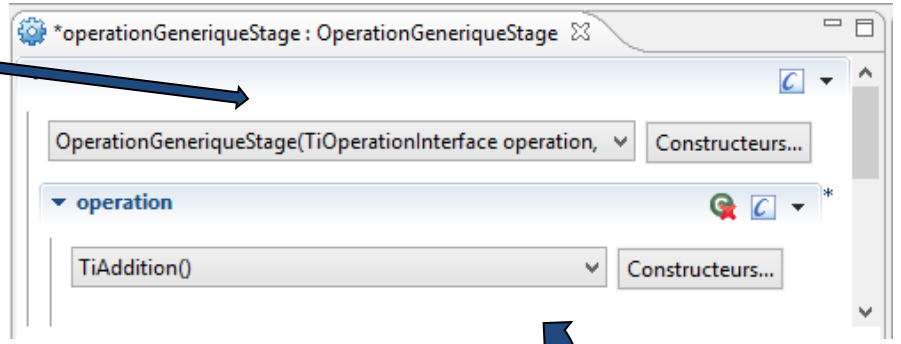


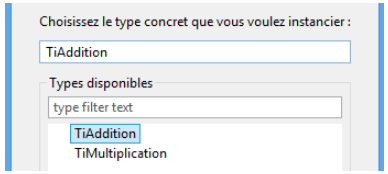
  - Strategy Design Pattern

```
public OperationGeneriqueStage(
TiOperationInterface operation)
{
…
}
```

```
public class TiAddition implements
TiOperationInterface
{
…
}
```

```
public class TiMuliplication
implements TiOperationInterface
{
…
}
```

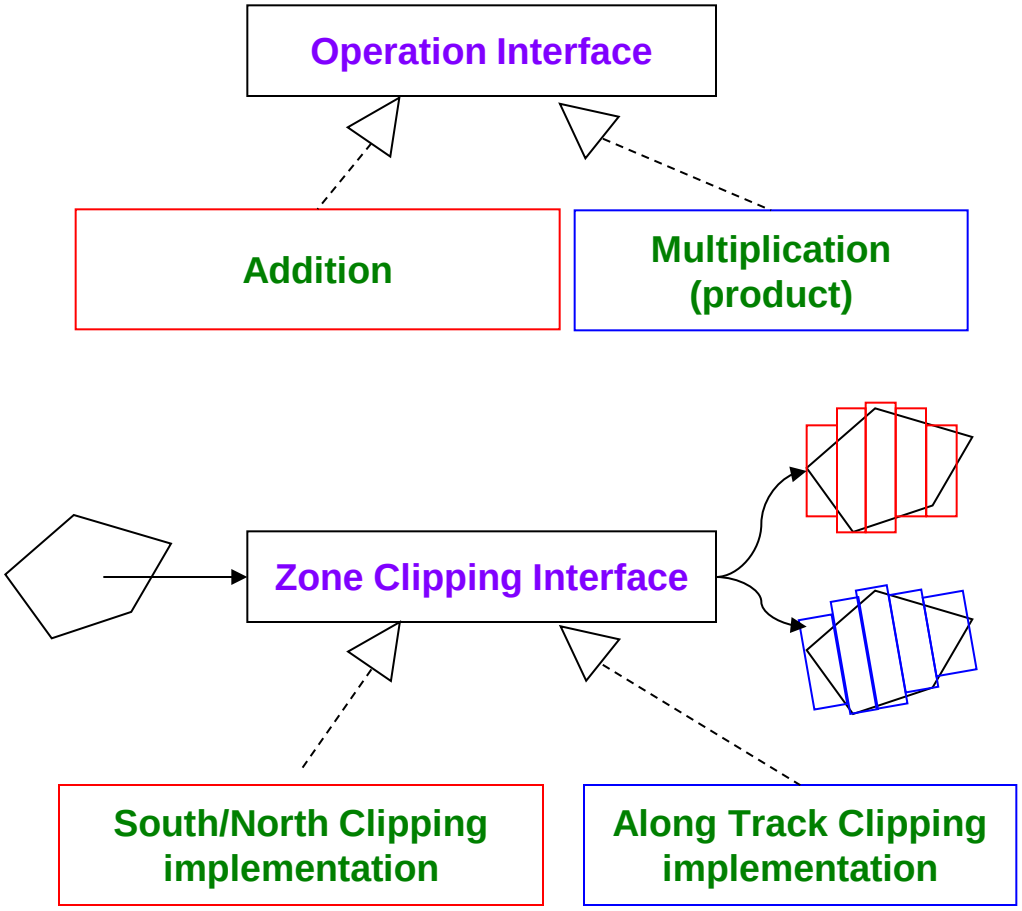




ALIS Infrastructure Characteristics
(ALIS versatility / SGO)

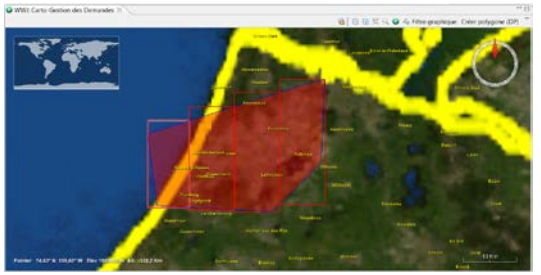
- ⇨ Mission Simulation at CNES
- ⇨ **ALIS Infrastructure**
- ⇨ Improve Computation Capacities
- ⇨ ALIS Versatility / COTS integration
- ⇨ Conclusions

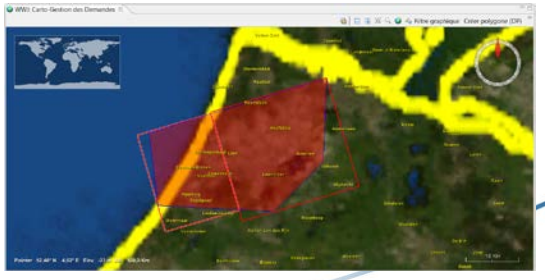
- SGO: *Show Generic Objet*
  - ✦ Strategy Design Pattern



+

*





ALIS Improvement of Computation Capacities

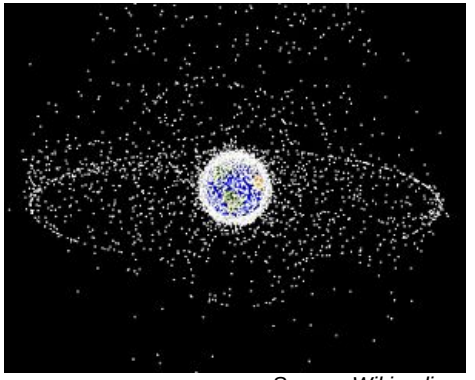
(Need for high performances)

Initial design: standalone infrastructure

- Simulation execution engine embedded with GUI layer

SSA simulator: ALIS' user with its own specifics

- Very large number of space objects (> 20 000 debris)
- Wide quantity of orbitographic data
- Synchronous / asynchronous processes integrating SSA algorithms



Source: Wikipedia

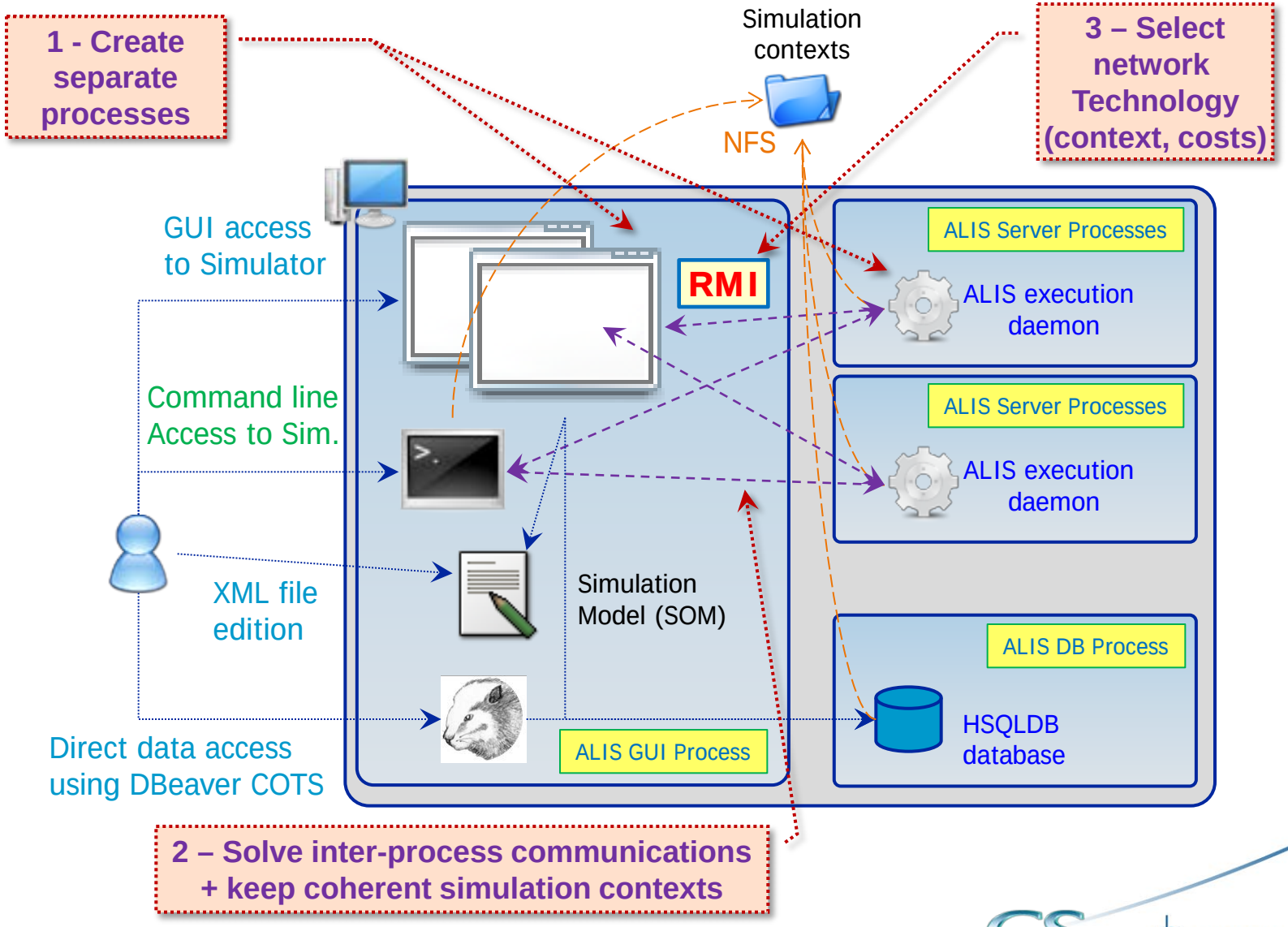
Consequence: need for **increased computation capabilities**

- Use on a Linux cluster, with technical constraints
- Needed split of simulation execution engine and GUI
- Deep ALIS' architecture re-engineering required
- Move to a distributed application (multi-process)

ALIS Improvement of Computation Capacities

(New ALIS distributed architecture)

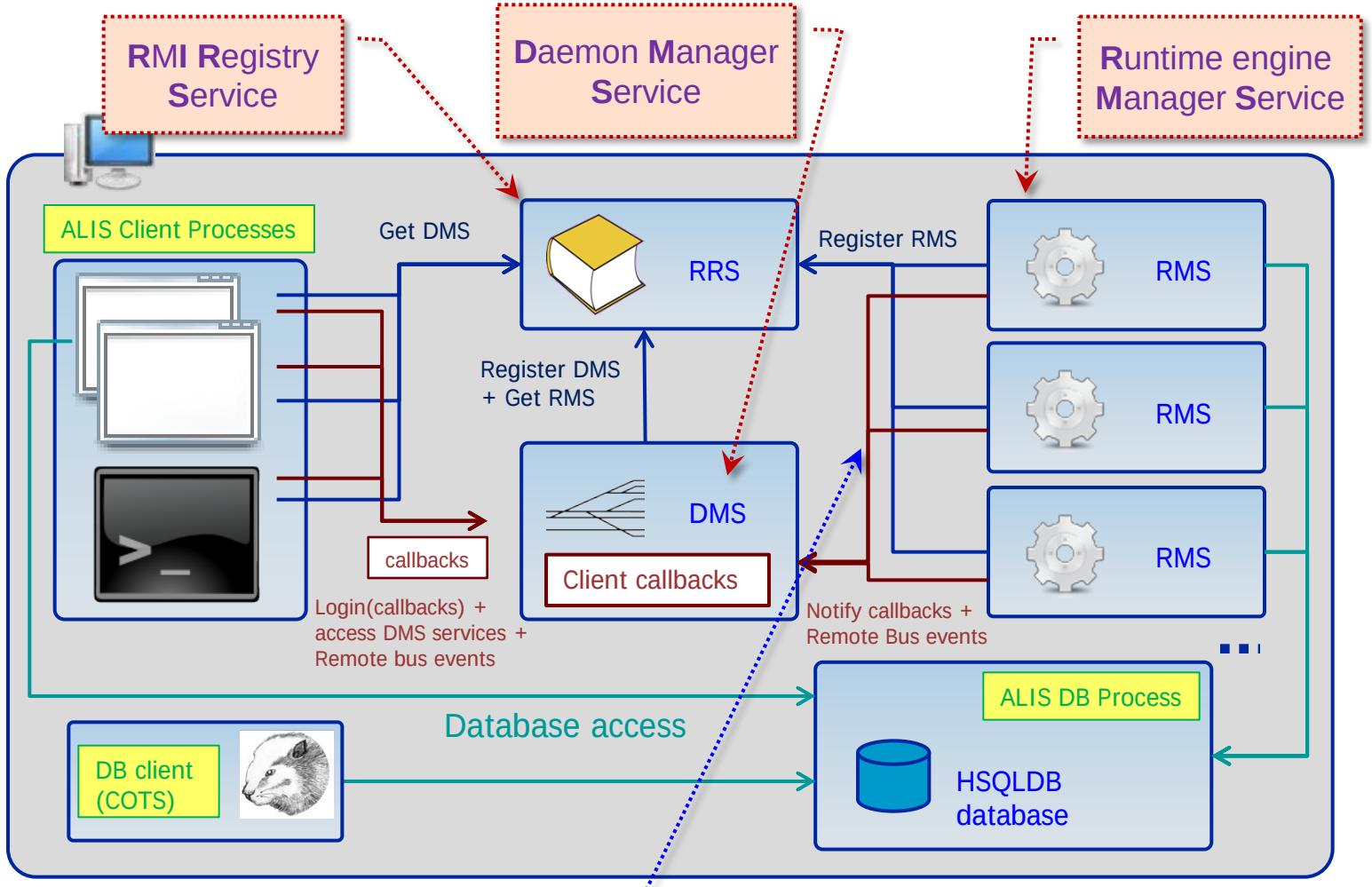
- Mission Simulation at CNES
- ALIS Infrastructure
- **Improve Computation Capacities**
- ALIS Versatility / COTS integration
- Conclusions



ALIS Improvement of Computation Capacities

(Close-up on processes interactions)

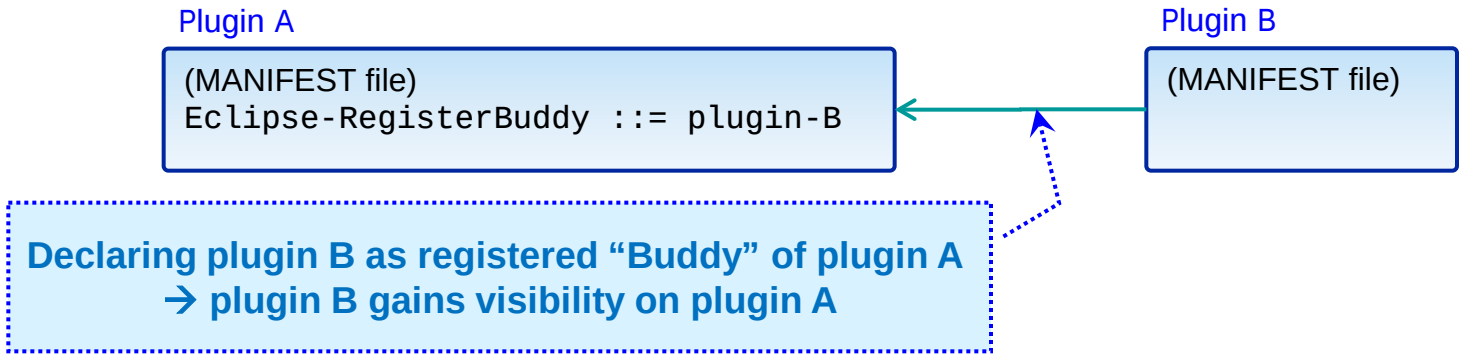
- Mission Simulation at CNES
- ALIS Infrastructure
- **Improve Computation Capacities**
- ALIS Versatility / COTS integration
- Conclusions



ALIS Improvement of Computation Capacities
(Technical issues)

RMI in an Eclipse-RCP / Spring context

- Two different dependencies resolution paradigms
  - Eclipse OSGi environment → dependencies solved using OSGi manifests
  - Other technologies (RMI, Spring, XStream) → standard Java classpath mechanisms to solve dependencies
- Eclipse provides the "Buddy Class loading" mechanism
  - Integration strategy to dynamically discover and load classes

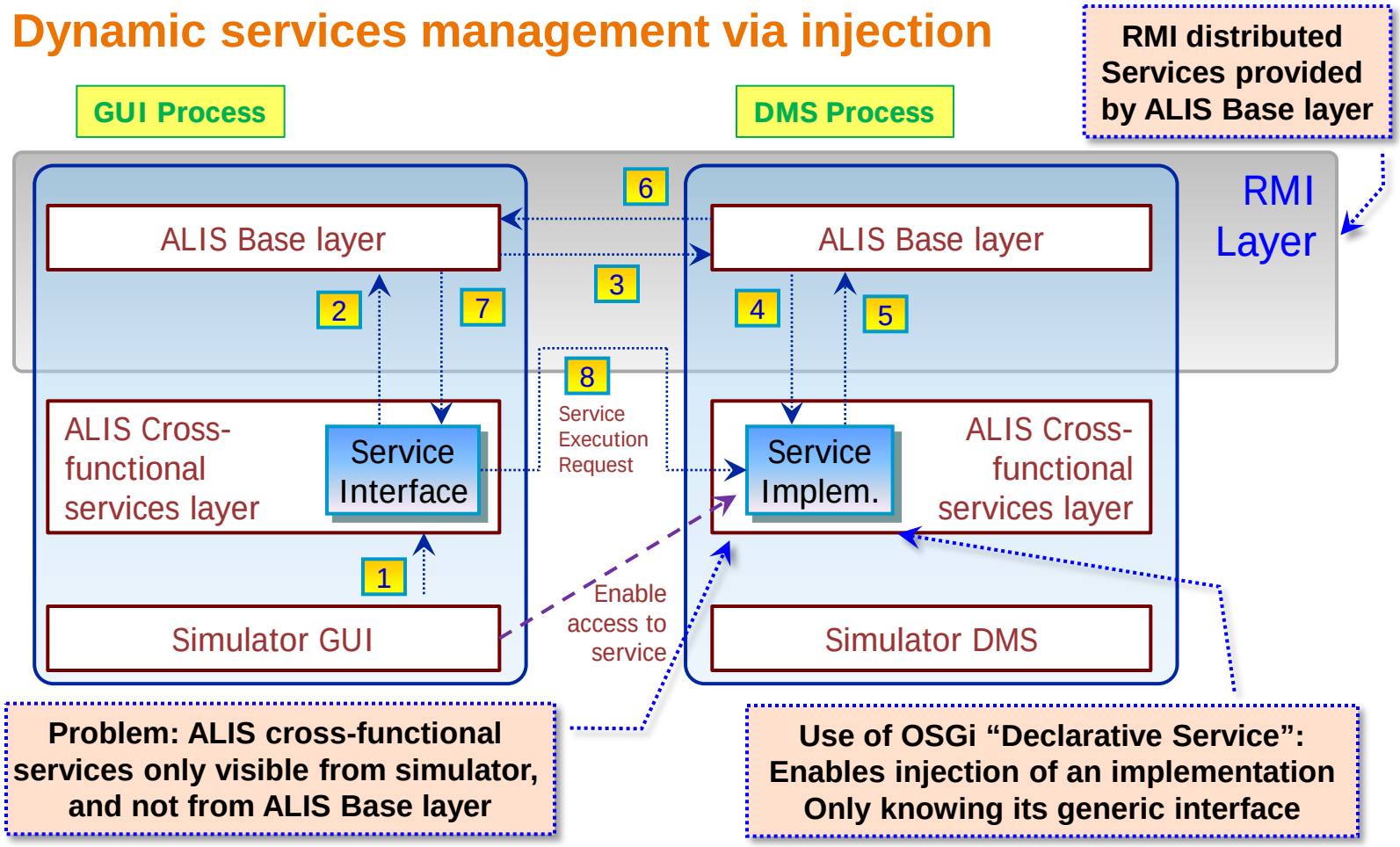


→ Minimizes modifications on existing source code

ALIS Improvement of Computation Capacities
(Technical issues)

- Mission Simulation at CNES
- ALIS Infrastructure
- **Improve Computation Capacities**
- ALIS Versatility / COTS integration
- Conclusions

Dynamic services management via injection



→ Enables extending ALIS remote services offered by the simulator processes without knowing their implementation

ALIS Versatility: Emphasis on COTS integration
(COTS licenses issues)

More and more services and functionalities are required

Strong drivers: development costs + quick availability

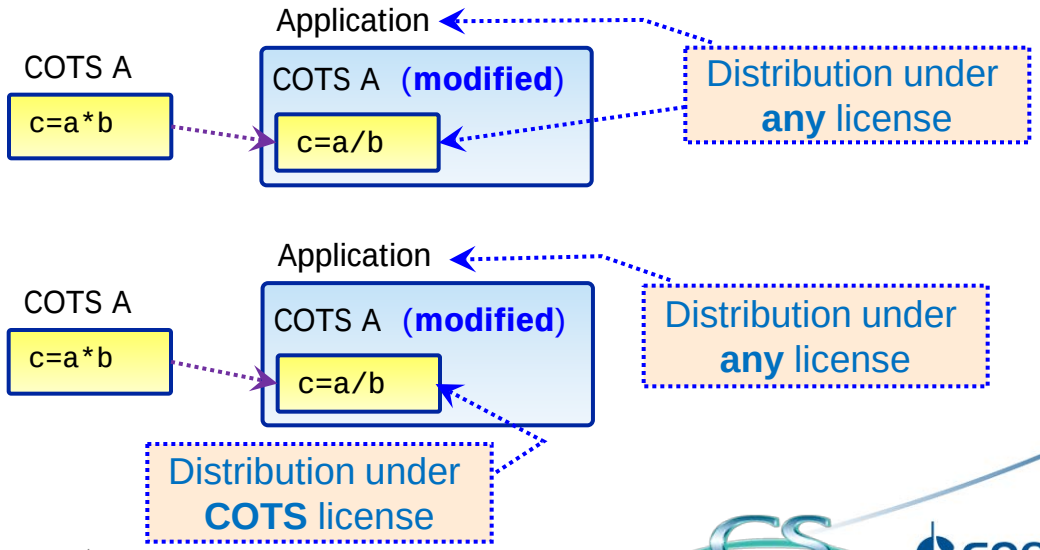
→ Use of COTS with high-readiness level
and adequate licenses

COTS licenses

- Privileged approach: select free open-source COTS with permissive or weak “copyleft” licenses

- **Permissive** license

- **Weak copyleft** license



ALIS Versatility: Emphasis on COTS integration

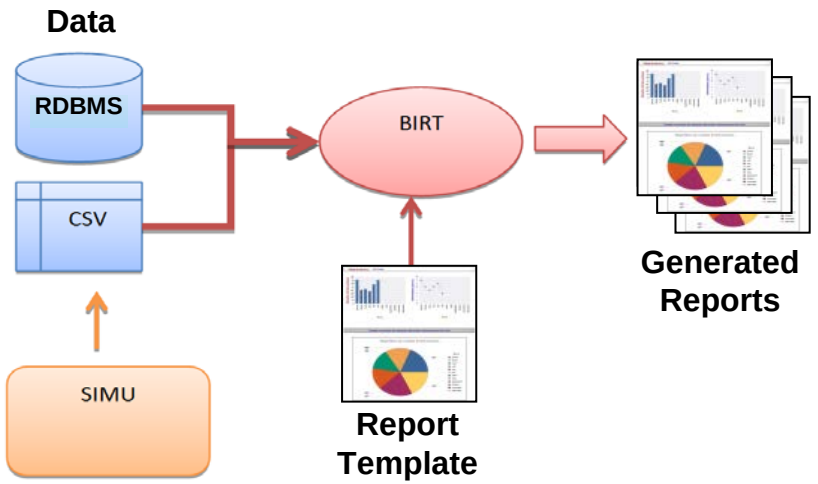
(COTS integration levels / sample)

COTS levels of integration

- From low-level libraries to high-level components with GUI
- Eclipse/OSGi facilitates COTS integration and ALIS increase of high readiness level services to simulators' users / developers

GUI COTS integration with GUI adapters (BIRT Sample)

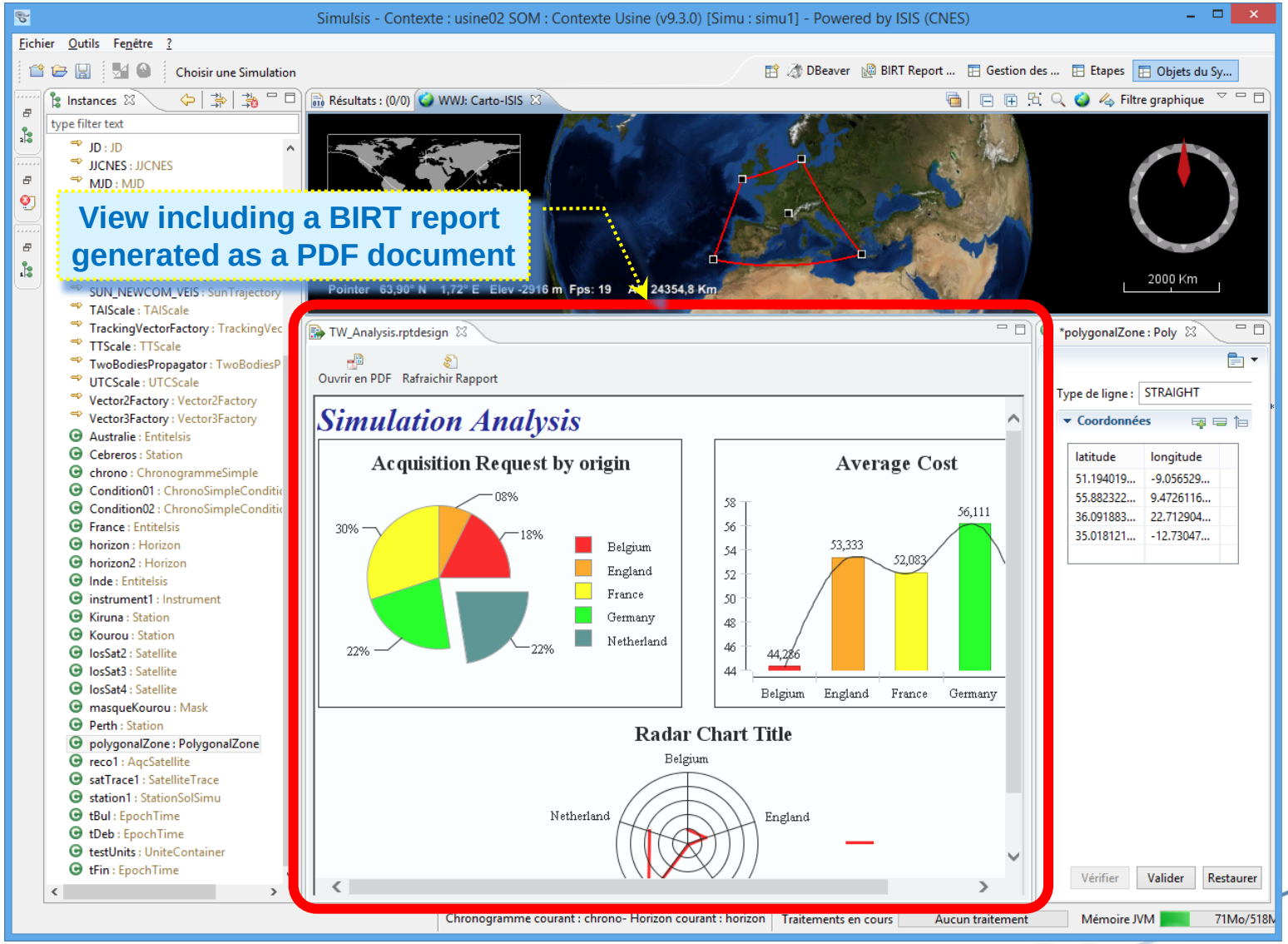
- Users need to produce, collect and analyze statistics about algorithms / parameters
- BIRT (high-level & powerful reporting tool) easily integrated in ALIS
- Improvements added: generate PDF exports and update reports' views during stages runs



ALIS Versatility: Emphasis on COTS integration

(COTS integration sample)

- Mission Simulation at CNES
- ALIS Infrastructure
- Improve Computation Capacities
- **ALIS Versatility / COTS integration**
- Conclusions



Conclusions

- ALIS is a scalable infrastructure designed to build mission simulators, allowing CNES to rapidly conduct Mission Expertise Studies
- CS SI brings to ALIS development its skills and experience to manage complex projects.