RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

KOSMOS - NIMPH

ROMAIN BOYER

KOSMOS DAY

24 SEPTEMBER 2026

AGENDA

01 NIMPH MISSION

02 RECOVERY APPROACH

03 ARCHITECTURE

04 INTEGRATION AND VERSION MANAGEMENT

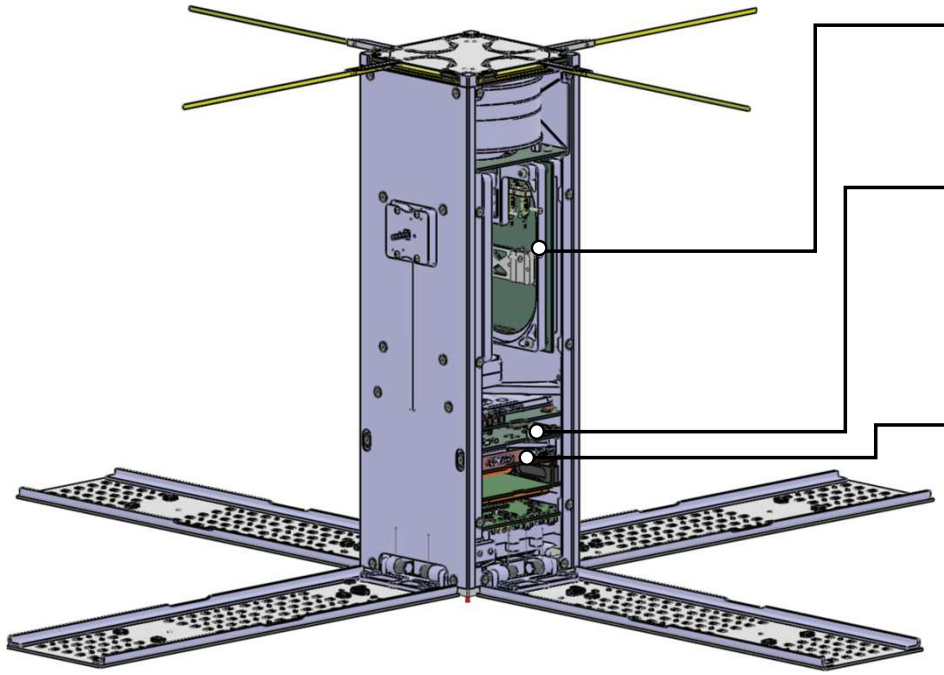
05 CONCLUSION

01

NIMPH MISSION

NIMPH

Nanosatellite to Investigate Microwave Photonics Hardware



EDMON [Erbium Doped fiber optic amplifier MONitoring]

Qualification of an Erbium-Doped Fiber Optical Amplifier (EDFA) for Space Applications

THINGSAT

868 MHz LoRa Transceiver Designed by CSUG (Academic Space Center of Grenoble)

M2M [Machine 2 Machine]

Proof of Concept for a Machine-to-Machine Relay via the Satellite UHF Link: Application to a Medical Use Case

RAM [Radio AMateur]

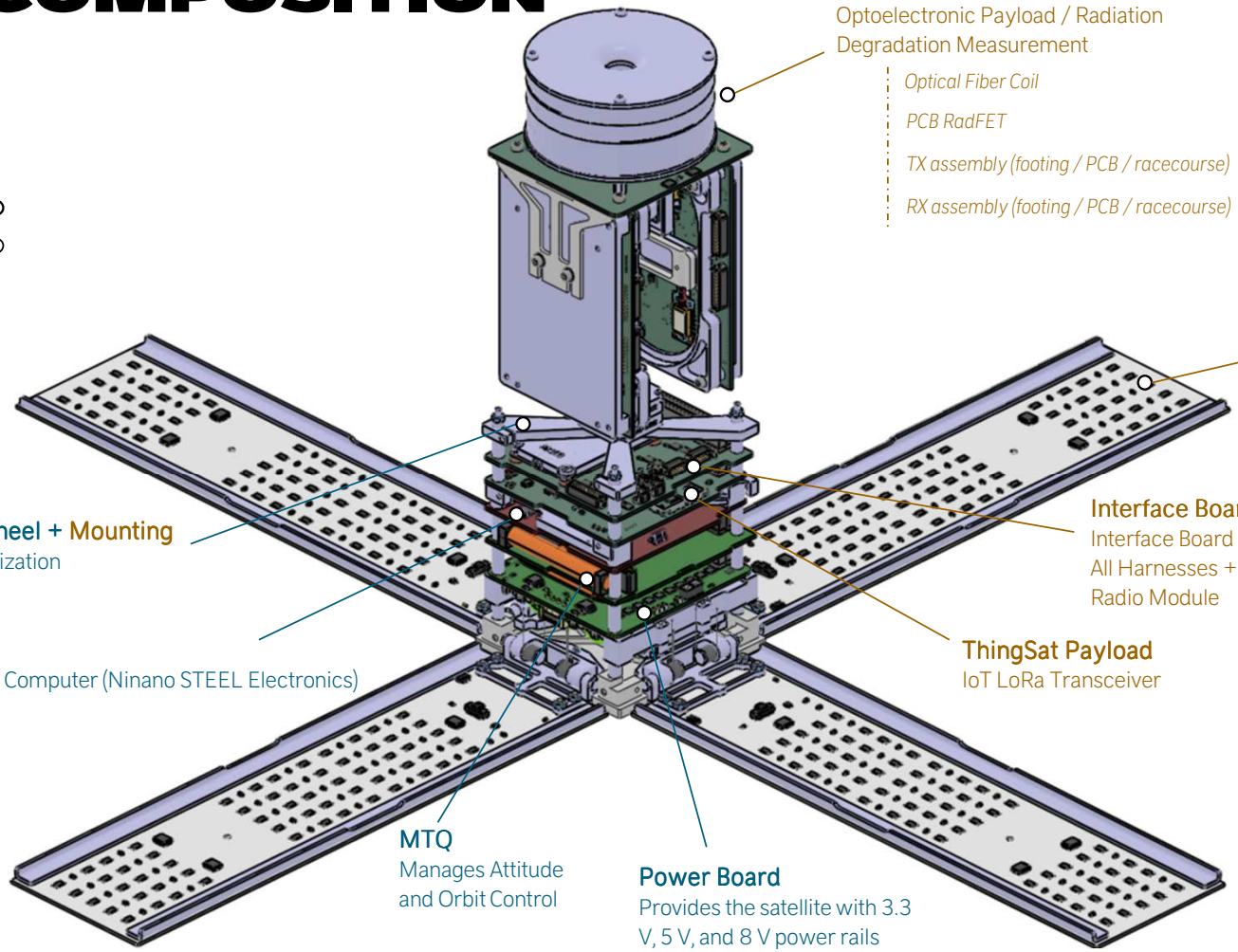
Data Exchange System for International Amateur Radio Operators (HAM Radio) through the Satellite's UHF Link

NIMPH – COMPOSITION

Legend :

CNES Delivery (SEED)

CSUT Design



EDMON Payload
Optoelectronic Payload / Radiation Degradation Measurement

- *Optical Fiber Coil*
- *PCB RadFET*
- *TX assembly (footing / PCB / racecourse)*
- *RX assembly (footing / PCB / racecourse)*

Solar Arrays
Conversion of Solar Energy into Electrical Energy

Interface Board
Interface Board Connecting All Harnesses + AX100 Radio Module

ThingSat Payload
IoT LoRa Transceiver

Reaction Wheel + Mounting
Satellite Stabilization

OBC
Onboard Computer (Ninano STEEL Electronics)

MTQ
Manages Attitude and Orbit Control

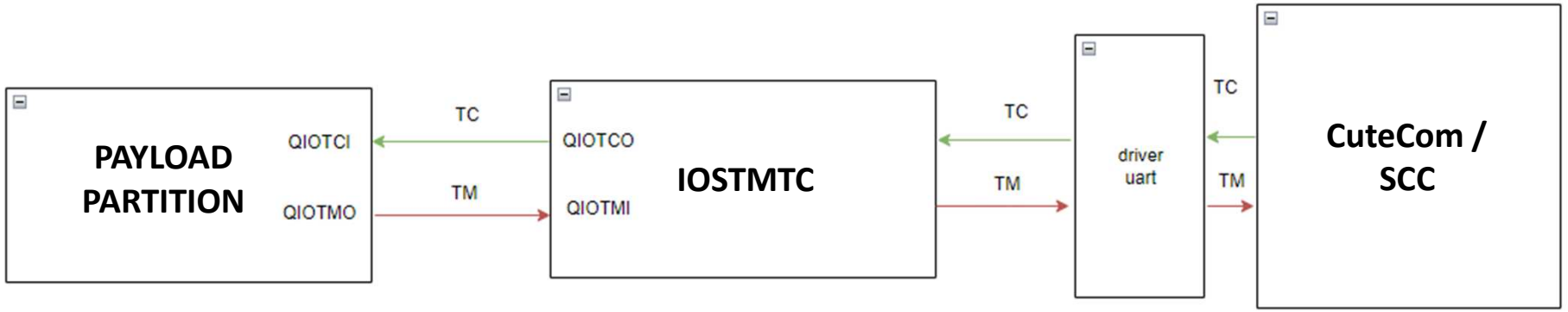
Power Board
Provides the satellite with 3.3 V, 5 V, and 8 V power rails

02

RECOVERY APPROACH

AVAILABLE SOFTWARE

- Two Flight Software Available :
  - LV Starter kit Nano CU
  - LV PF Nanolab



LV Starter kit Nano CU Schematic

SELECTED SOFTWARE

- Software Selection based on…

Pros | Cons	LV Starter kit Nano CU	LV PF Nanolab
++	Easy to get started	Integration of avionics modules
++	Limited scope to the partition → fewer side effects	Reduced integration risk
--	Integration effort later in the project	Updates are more difficult
--	Fewer testing opportunities with the SEED rack	Inheritance of all modules

- **LV PF Nanolab was selected** based on CSUT previous experience with simplified versions and its compatibility with avionics integration.

INITIAL DEVELOPMENT

PARALLEL PAYLOAD PARTITION DEVELOPMENT

- Partition Addition
  - Add a partition based on APPDKPUS
  - Add a LithOS process
  - Add a PUS service

- OneWire Device Integration
  - EDMON payload temperature sensor
  - Heater control switch

- Initial development was **challenging**, requiring a **best-effort approach** with interns and significant turnover

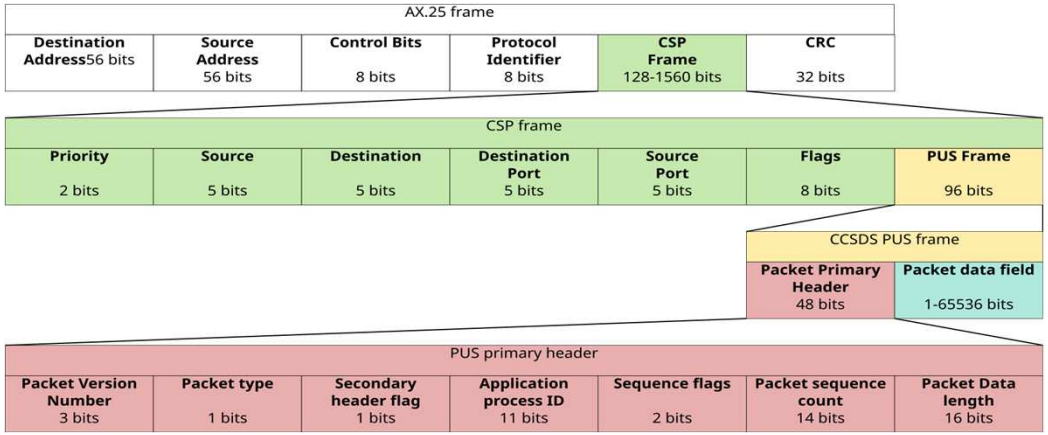
SPECIFIC DESIGN

ADAPT FS SCOPE TO CSUT NEEDS

- Payload Driver Integration
  - I²C driver
  - PS UART driver
  - GPIO driver
    - => Direct driver control from payload partitions

- PS CAN Driver Integration for the UHF Transceiver
  - Interface with the IOSTMTC partition
  - Integration of transceiver protocols (CSP, CFP)

- New FINAL_UHF Mode
  - Interface with the PS CAN driver
  - IOSTMTC task parameters linked to RF parameters

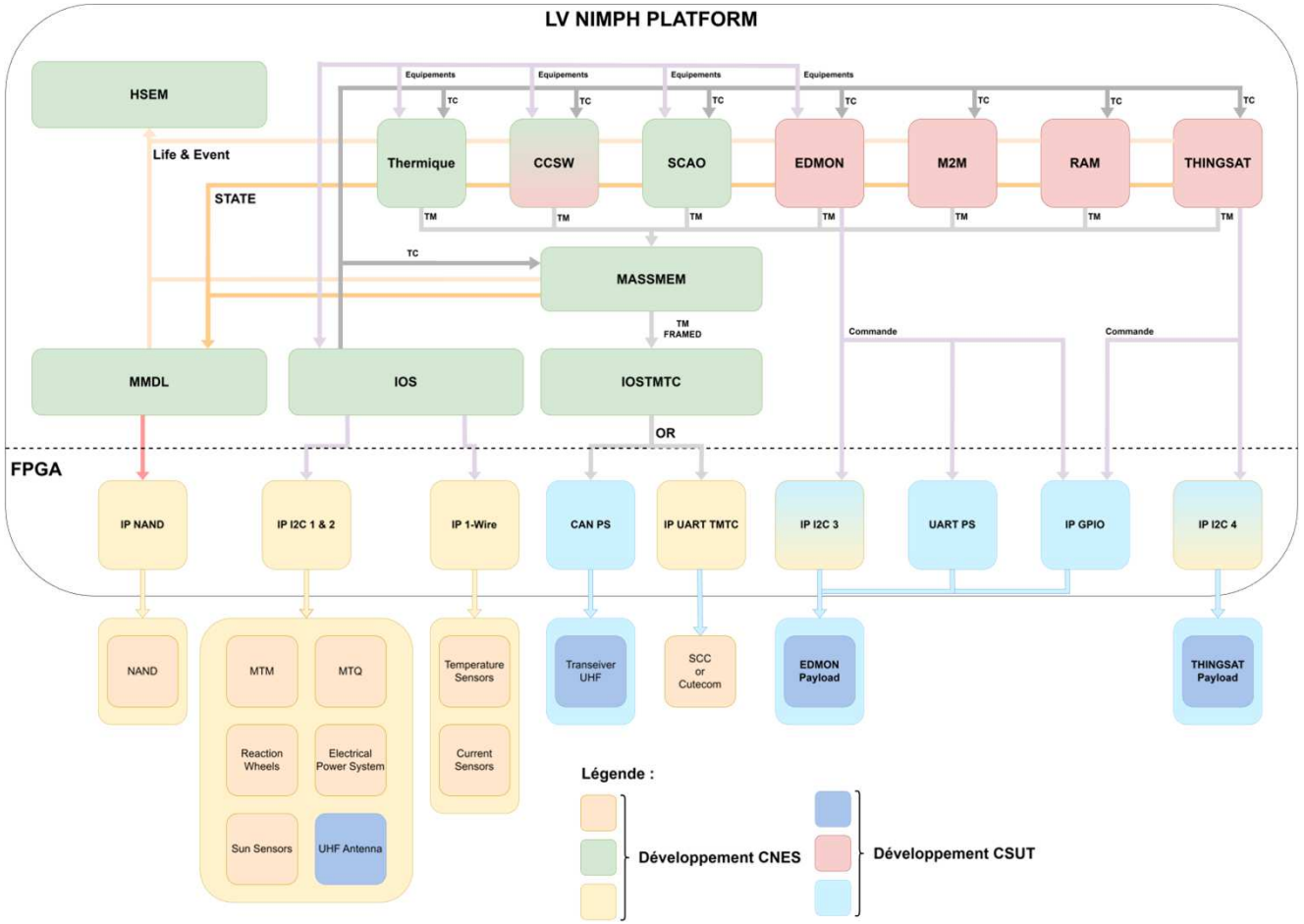


TMTC Encapsulation over UHF Link

03

ARCHITECTURE

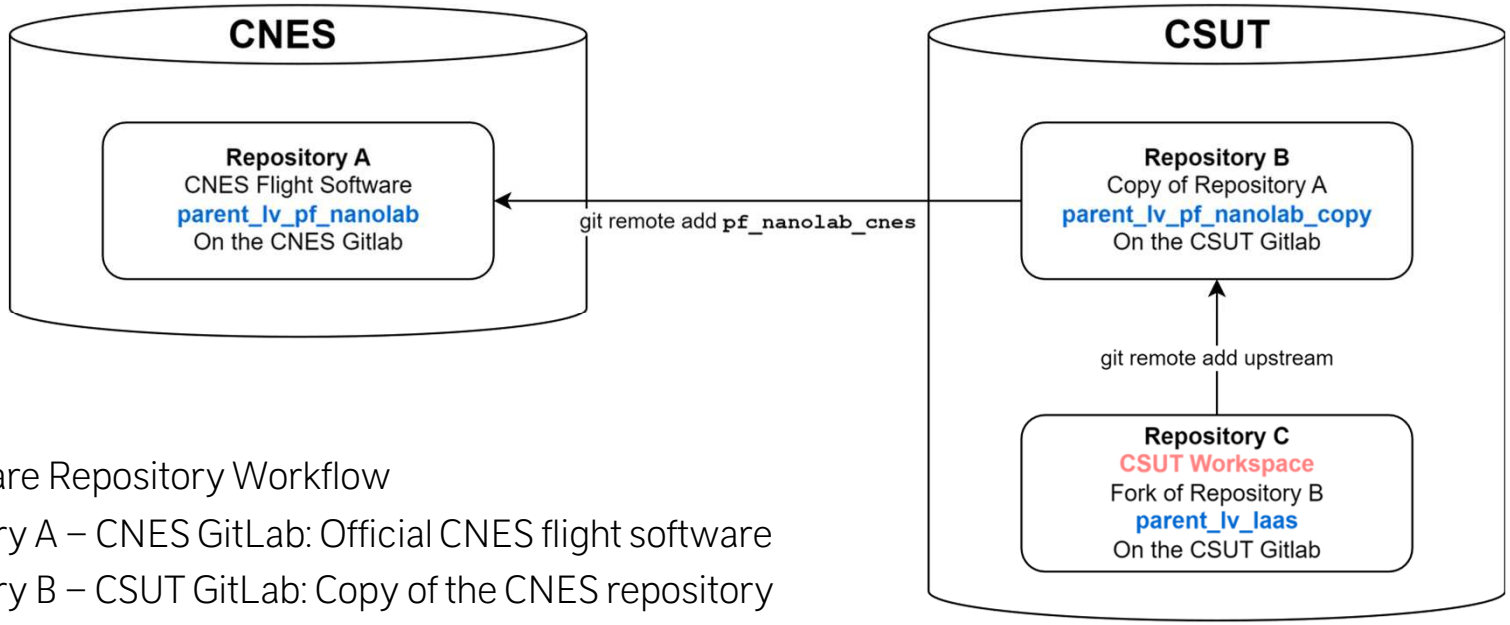
FLIGHT SOFTWARE ARCHITECTURE



04

INTEGRATION AND VERSION MANAGEMENT

MANAGEMENT AND WORKFLOW



- Flight Software Repository Workflow
  - Repository A – CNES GitLab: Official CNES flight software
  - Repository B – CSUT GitLab: Copy of the CNES repository
  - Repository C – CSUT GitLab: Fork used for CSUT developments
- Workflow:
  - CNES baseline → CSUT copy → CSUT development

DAILY VERSION UPDATES

- Update Process
  - Stabilize the CSU software beforehand
  - Review current functionalities (UHF Radio, Payload Bus)
  - Merge individual submodules, then the LV PF with the updated submodule instances

- Error Management
  - Identify the source of the error
  - Compare with the original CNES version when needed
  - Re-verify CSU functionalities after the update

- Support and Follow-Up
  - Weekly meeting open to all CSU teams
  - CNES feature overview for each new LV release

05

CONCLUSION

FLIGHT SOFTWARE REVIEW

A big thank you to the Flight Software team for their support, especially Julien and Eliott!

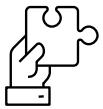


- Works well
  - Software modularity enabling relatively independent development of each partition
  - Onboarding tutorials and support provided by the Flight Software team
  - Generic / specific architecture allowing unchanged modules to be inherited



- Could be improved :
  - Early anticipation of major updates impacting the CSU design
  - Makefile build system
  - Code organization, with some structures and variables difficult to locate

FLIGHT SOFTWARE REVIEW



- CSUT challenges :
  - Training time for a complex project
  - Knowledge transfer with student turnover
  - Limited time and experience of interns



- CSU Approach :
  - One permanent engineer overseeing the flight software development
  - Sub-tasks assigned to interns within the LV

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

KOSMOS - NIMPH

ANY QUESTIONS?

KOSMOS DAY
24 SEPTEMBER 2026