# MISSIONIZATION

# PROCESS FOR MULTI-MISSION VEHICLES

## Final Presentation (FP)



INNOVATING SOLUTIONS

# Project Technical Approach

# Project Background and Objectives

- The Flight SoftWare (FSW) of a recurrent operated space vehicle, requires a big amount of effort especially during the qualification phase.
- The on-board SW includes a relevant set of parameters that depend on the specific mission the vehicle is going to perform. The values to be assigned to these parameters are normally computed well before the flight using dedicated tools and specific know-how.
- This aspect is particularly relevant when addressing the main part of the on-board SW that concerns with the Guidance, Navigation and Control Function characterized by several parameters that are mission dependent

# Proposal Background and Objectives

- Is not viable to repeat the whole **QUALIFICATION PROCESS** for each mission for both costs and schedule limitations

- The main goal for an affordable SW production should count on an executable code that has been already qualified in a well-defined domain and that remains unchanged through the different missions.

- Whenever a new Mission Data (MD) is produced it should be verified that the new MD belongs to the qualification envelope and linked the unchanged core FSW

- This missionized SW then shall enter into an acceptance process aiming at verifying that the mission requirements are fulfilled by the new missionization

# Project Background and Objectives

- This missionized SW then shall enter into an **ACCPTANCE PROCESS** aiming at verifying that the mission requirements are fulfilled by the new missionization.

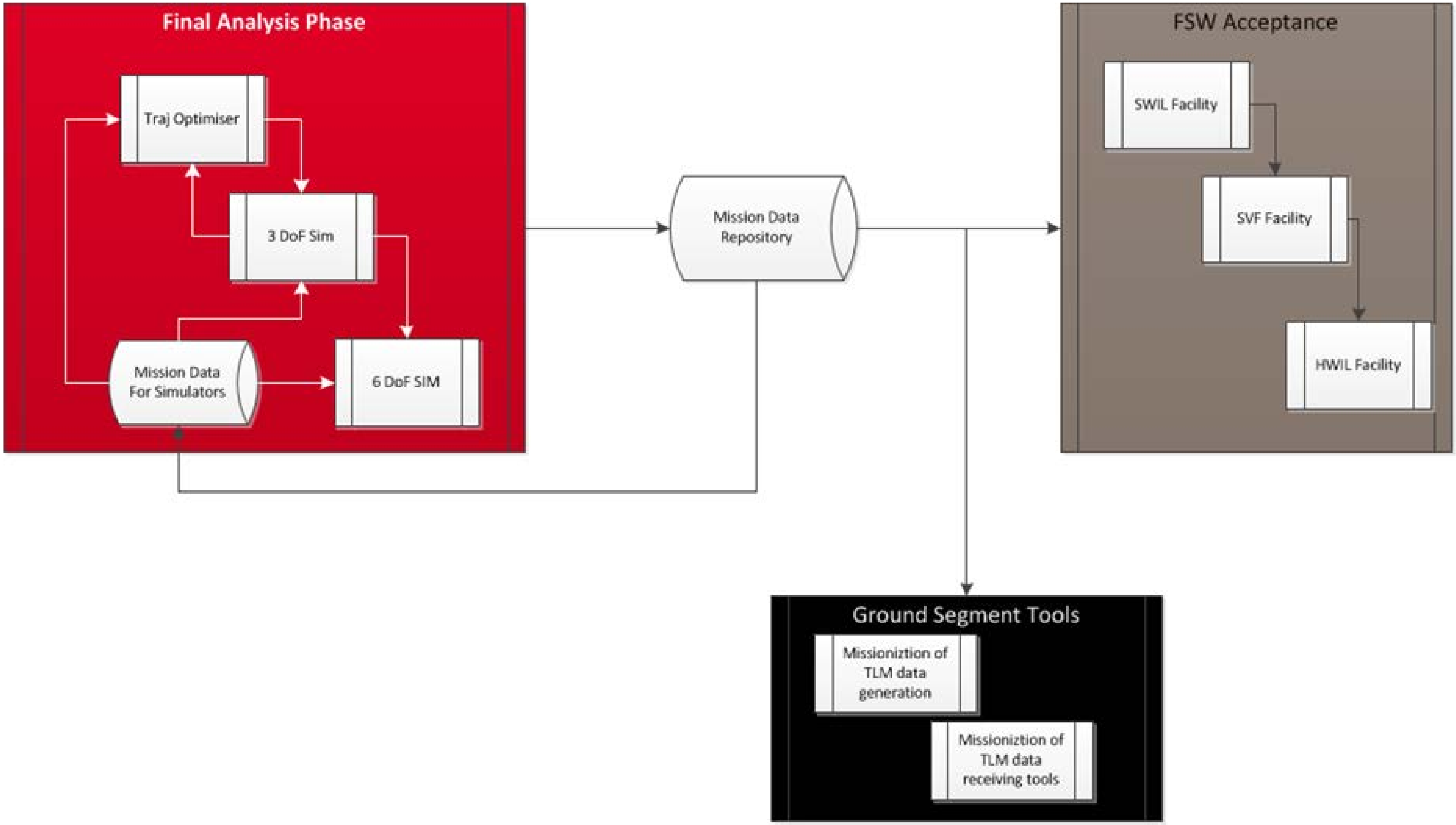
- To attain this relevant goal two main aspects are emphasized in the SoW:

  – Software Architecture Design. The whole FSW should be designed under the requirements of parameterization and modularity versus the missionization parameters.

  – Mission data Management.
  A properly repository architecture shall be defined in the frame of this contract for the mission data repository in order to guarantee the correct data flow and interchange protocols.

# Project Background and Objectives

- The mission data repository in fact should be able to interface with the different elements of the acceptance process: Simulators, Facilities, Check tools



VEMIPRO-PDR

# Missionization Process Definition

# Missionization Concept Description (1)

- Produce and verify FSW and associated elements (eg simulator configuration)
- Set of Parameters computed from
  - Know-How + Tools
  - Different Actors
- Interface rules between them and Data Repository
- Missionization Process is divided in 3 levels:
  - Phases (sequential)
    - Processes (1st level)
      - Activities (2nd level)
- Phases:
  - Preliminary Analysis – check feasibility, compatibility
  - Final Analysis – Formal approval through different simulators (cascade process)
  - Production – progressive coding from prototype into target / linking of code to MD
  - Acceptance – activities are connected with facilities : SVF, SWIL, HWIL

# Missionization Concept Description (2)

**From Qualification to Acceptance**

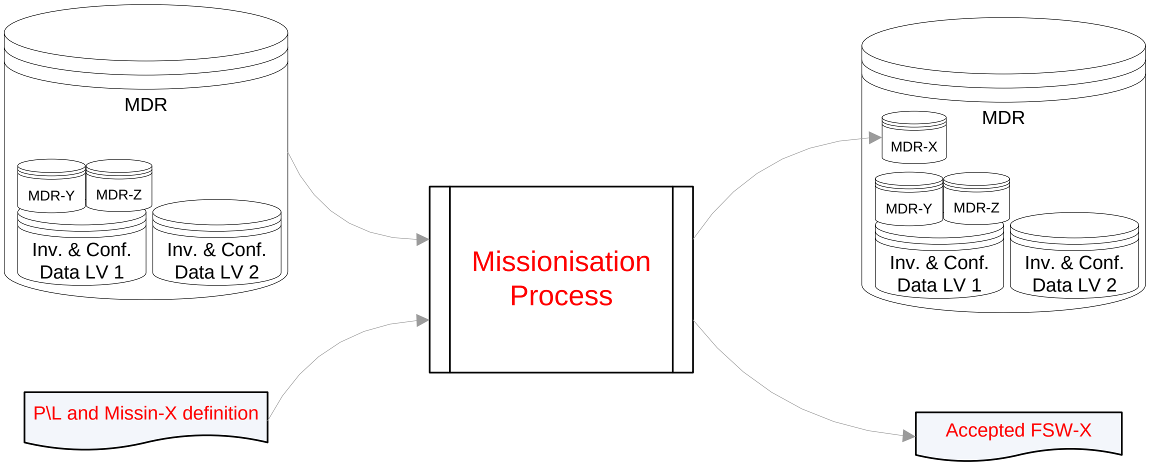
- Qualification
  - The executable core of FSW should be qualified in a domain
- Acceptance
  - New Mission: Mission Data is verified to be within the envelope
  - Link new MD to the core code – acceptance is a simplified qualification process

Design architecture for:

- Software – parametrization and modularity
- Mission Data Management – ensure data flow, consistency
  - Tools and Facilities, Simulators, Ground Tools

**Mission Data Repository**

- Database supporting the mission data
- Several specific missions
- Common data for multiple missions / vehicles
- **Invariant Data**
- **Configuration Data**



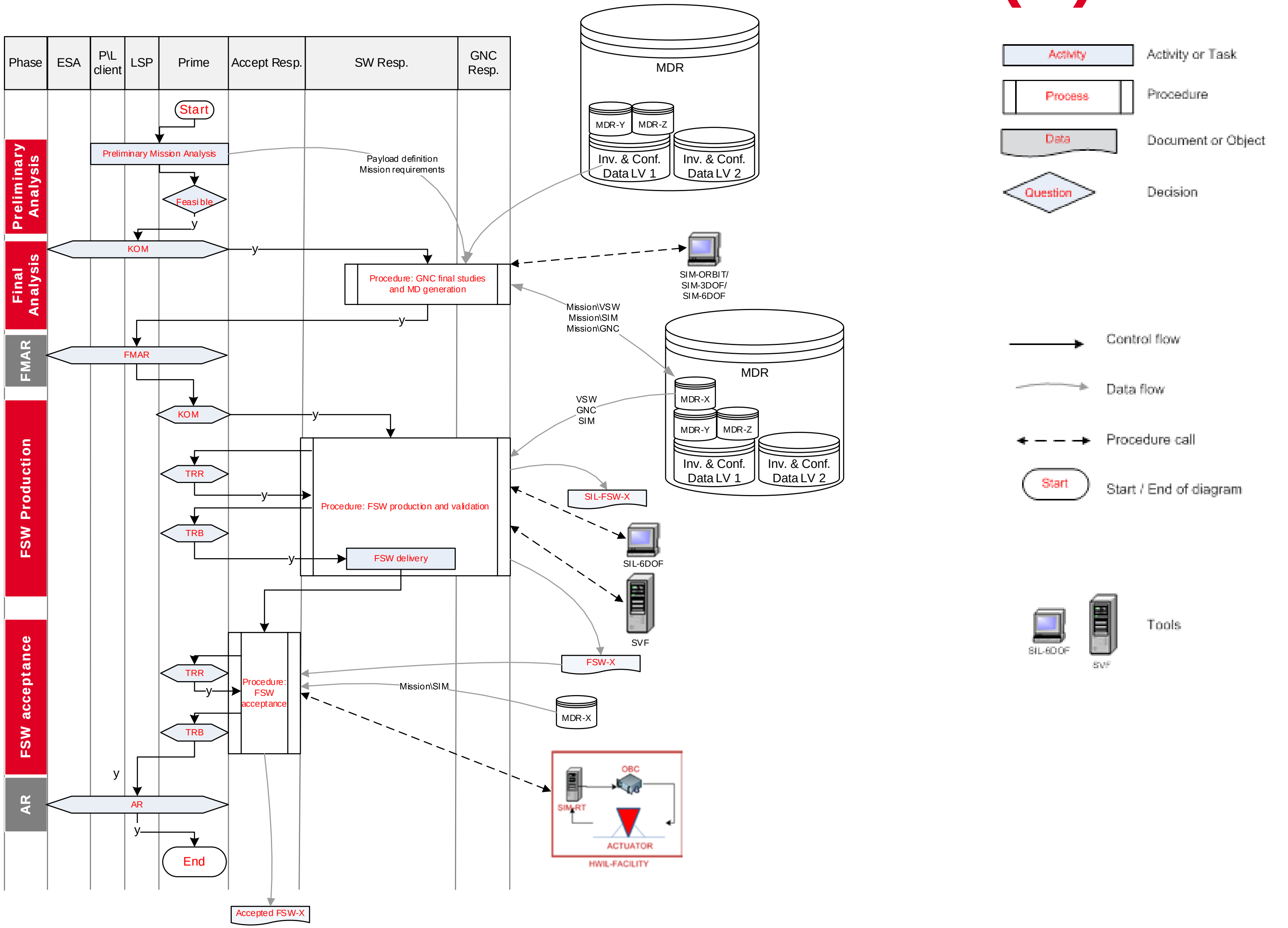
# Missionization Process Overview

**Phases**

- Preliminary Studies
  - Performed by the Service Provider Authority (support from prime and clients)
  - Early integration studies (flight envelope)
  - Qualitative Analysis and Checks with pre-existing Tables and quick MA tools
- Final Analysis
  - Definition of the Final Mission (data for FSW, GS)
  - Cascade / Iterative Process – using sequentially more complex tools
  - Approval of the Mission Data – Final Mission Acceptance Review (FMAR)
    - Responsibility - FSW development industrial prime – mainly to the GNC team
- Production of the Flight Software (FSW)
  - Generation of the binary and validation on SVF target
  - Facilities to sequentially validate and go from prototype to binary on target
  - Responsibilities (Binary Production - TRR , validation of SVF target – TRB ):
    - Mainly SW Responsible – SW Architect, SW Developer, Verification and Validation and ISVV
    - GNC Team
- Acceptance of the Flight Software
  - Acceptance in the HWIL facility
  - Responsibility is on Industrial Prime – operates HWIL facility

# Missionization Process Overview (3)



VEMIPRO-FR

# Missionisation Tool Requirements and Design

INNOVATING SOLUTIONS

# Missionization Tool Use Cases

Missionization Tool is to be used, with main use cases

- Creation of a new mission instance in the repository;
- Create a new version of a given mission/LV configuration instance in a mission repository;
- Population and/or modification of the repository;
- Data extraction from the repository for different purposes and targets. The following targets are considered:
  - Generation of missionized FSW, including data conversion to high level languages like Ada or C based on templates.
  - Test and SVF tools, including data conversion to Matlab, Perl, Python, csv and/or textual (readable by Matlab or other modelling tool) formats.
  - Ground Segment tools, including data conversion to TLM or Ground Decommutation File (GDF) formats.
  - Formal documents, in the form of tables with textual description of the data.

# Missionization Tool Principles Design

- **Modularity** is related to the easy and clean capability of substituting existing parts and components of the VEMIPRO Missionization Tool.

  – High level design:
    - Goal and responsibilities of each component of the tool's control logic is well defined providing a clear separation of functionalities.
    - Each component is designed to have only simple I/F interactions with the other components.

  – Mission data repository design:
    - Stored data is precisely categorized and identified.
    - Parameters divided into different hierarchical level following the logic Category -> Family -> Sub-Family.
    - Parameters further separated by Mission and Mission Version.

# Missionization Tool Principles Design

- **Configurability:** possibility to easily configure and adapt the tool to:
  - specific missions & launch vehicles
  - different templates for exporting/importing mission data
  - different users with different roles
  - different physical sites.

# Missionization Tool Principles Design

- **Maintainability:** Tool is designed as a mission data repository that can continuously evolve adding new missions, mission versions launch vehicles and their configurations in the frame of future activities or missions.
- Furthermore, new users, data checks, internal/external interfaces and input/output formats can be added without hurting the tool's functionality.

  – logic components simple, with precise functionalities and easily identifiable

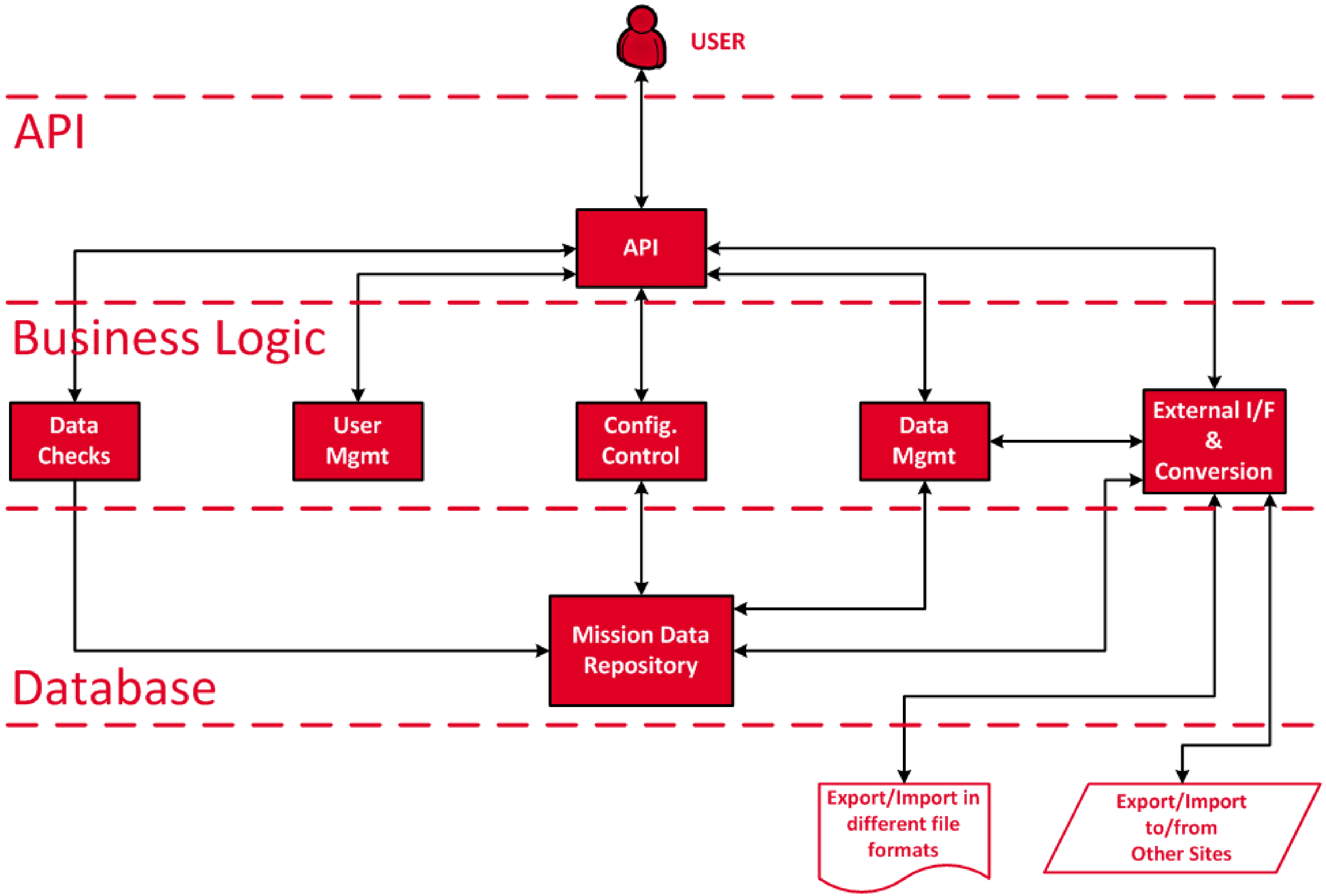
  – mission data stored in the tool's repository is separated according an hierarchical scheme

  – not specific to any mission or launch vehicle

  – any changes performed on the mission data stored in tool's repository are registered

# Missionization Tool High Level Architecture



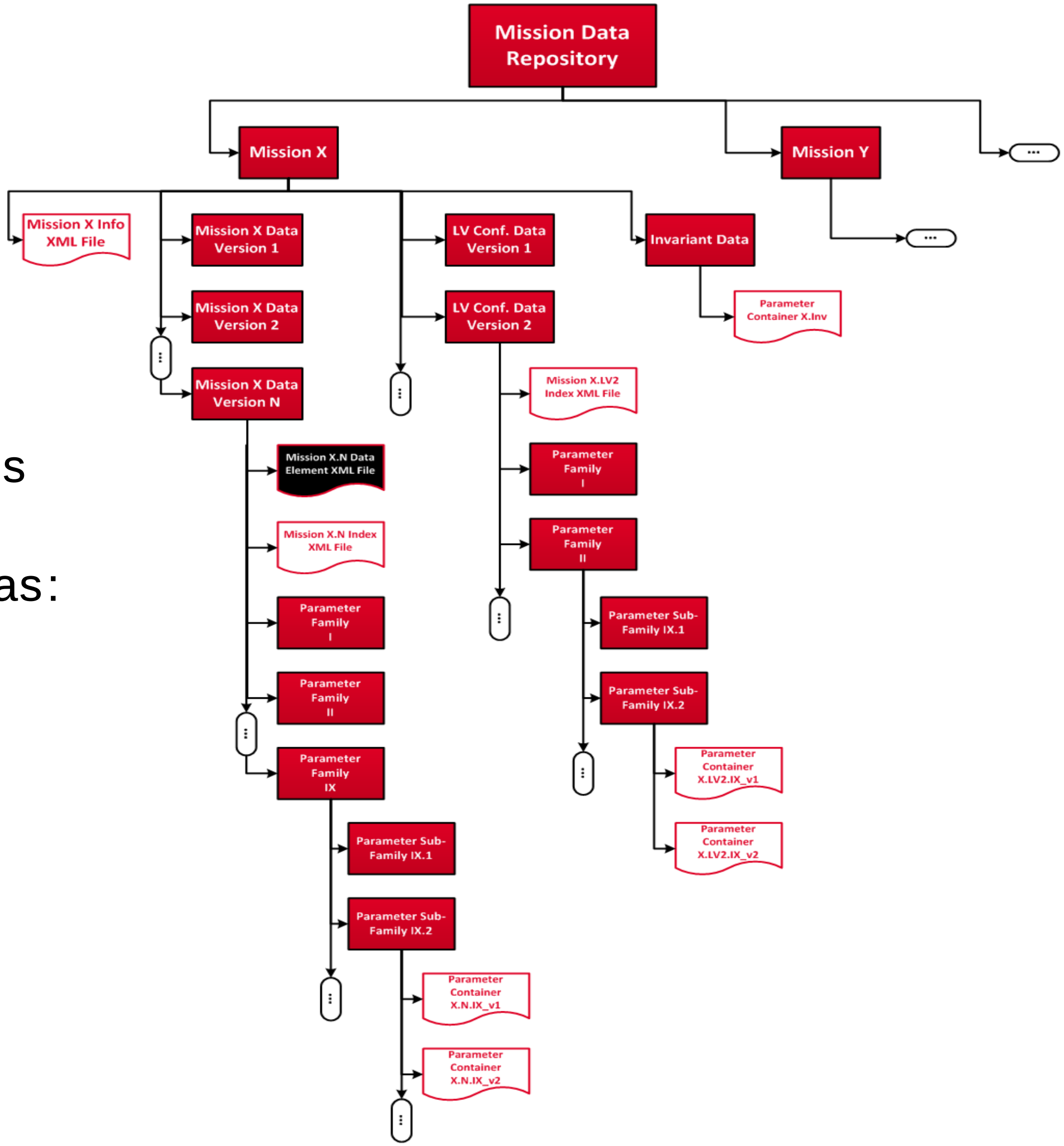
# Missionization Tool Modules

- **API**
  - Responsible for the user interface with the Tool and providing a Graphical User Interface (GUI) and a Command Line Interface (CLI)
- **User Management**
  - Responsible for storing the information related to each user (e.g. authentications, permissions, logging history, etc.)
- **Data Management**
  - Responsible for handling the data in the mission data repository (e.g. import data, compare, modify, synchronize, etc.)
- **Data Checks**
  - Responsible for performing automatic checks on the data stored in the mission data repository (e.g. data checksum, integrity, format, syntax, external checks, etc.)
- **Configuration Control**
  - Responsible for keeping a record of the changes performed on the data repository (e.g. versioning, tagging, history, grouping, etc.)
- **External I/F & Conversion**
  - Responsible import/export of data from/to local areas and conversion of the mission repository data into textual format

# Missionization Tool Modules

- **Missionization Data Repository**
  - Responsible for the persistence of the mission data inside the VEMIPRO Missionization Tool
  - Many missions and versions can be stored
  - Each mission is organized as:
    - Mission Folder
    - Configuration Folder
    - Invariant data Folder
  - Parameters family and subfamily folders
  - Data stored in XML files
  - Mission Data Element XML file



VEMIPRO-FR

# Study Cases Definition and Scenarios

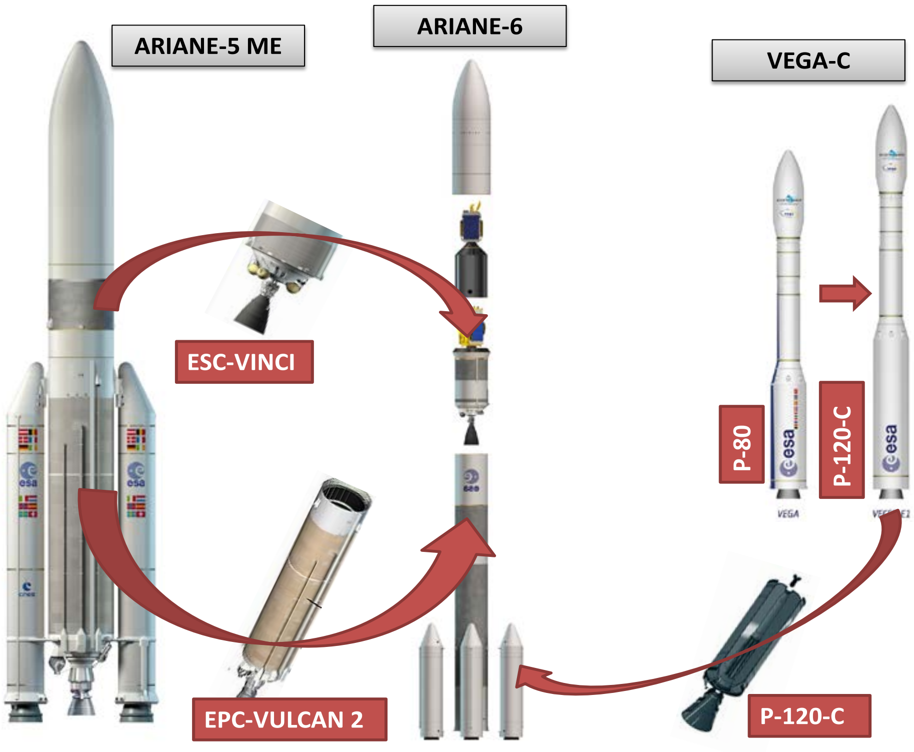
INNOVATING SOLUTIONS

# Study Case: ARIANE-6

# Study case: Ariane 6

- Ariane-6 launcher version, approved to the last ministerial council shall be hence based on:
  - Lower Stage: a cryogenic main stage (LOX/LH2), propelled by Vulcain engine (in its version 2+).
  - Strap on Boosters: two or four Solid Rocket Boosters (SRB), in the class of 120t of propellant (dubbed P120C), which will be common with Vega-C.
  - Upper Stage: a cryogenic upper stage (LOX/LH2) propelled by Vinci engine, based on the Ariane 5 ME Upper Stage



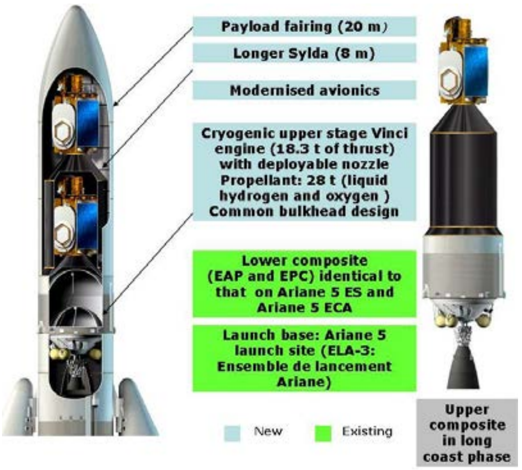

 

# Ariane-6 Configuration

- In its current version Ariane 6 considers two main configurations:

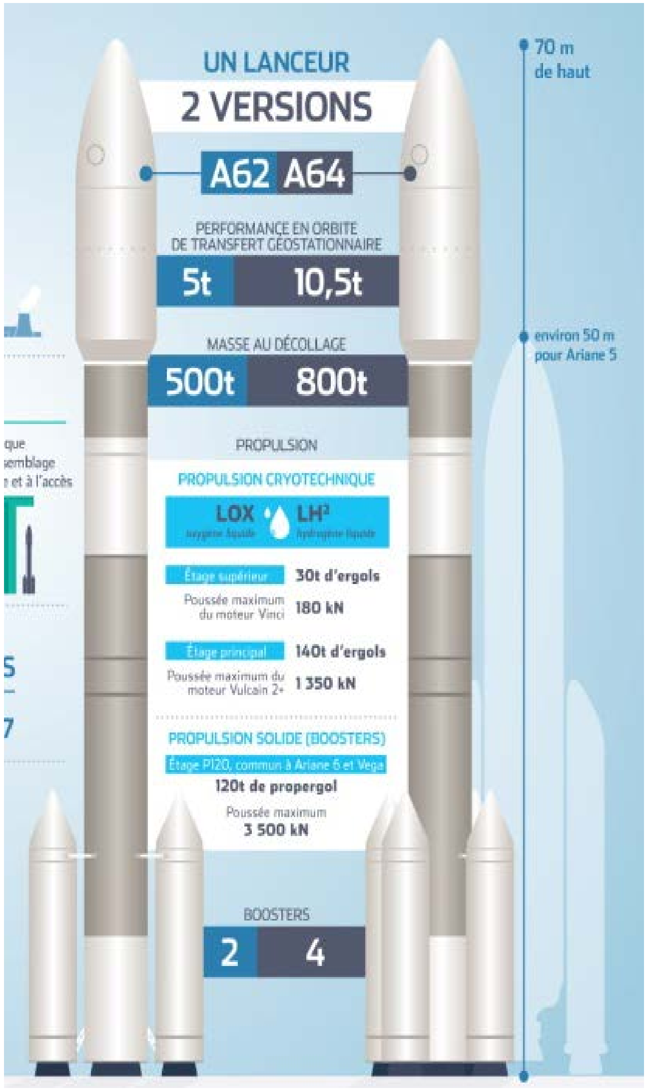
- A62 equipped integrating
  - Two Solid Rocket Booster based on VEGA P120-C stage
  - A short Fairing for a single payload release on to GTO orbit
  - A single payload adapter.

- A64 equipped integrating
  - Four Solid Rocket Booster based on VEGA P120-C stage
  - A long fairing for a dual payloads release on to GTO orbit
  - A very long SYLDA adapter shall be derived from A5ME program to allow the dual launch




# Ariane-6 Vehicle

| Configuration | Performance |
|---|---|
| A-62 | 5.8t GTO |
| A-64 | 10.9t GTO |
| A-62/A-64 | ≥4.5t SSO |



# Ariane-6: Mission & Main Mission Phases





# Ariane-6 Mission

- Ariane-6 specifications require the launcher to attain several orbits: LEO, MEO, GEO, HEO, GTO, GTO+, Hyperbolic

| Secondary Missions | Type | Configuration | Qualification to be achieved in less than (months) | Orbital Parameters |
| --- | --- | --- | --- | --- |
| MEO-Galileo | Single Launch | A62 | 24 | Circular orbit<br>22620 - 23220 km<br>Inclination: 54° - 58° |
| LEO/ATV | Single Launch | A62 | 24 | Circular orbit (260 to 400 km)<br>Inclination: 51.6° |
| GTO+/GTO ++ | Dual Launch | A64 | 6 | 180 km ≤ Perigee L ≤ Perigee U ≤ 35286 km<br>Apogee: 35.286 km |
| Direct GTO+ | Single Launch | A64 | 15 | 2000 km ≤ Perigee ≤ 35286 km<br>Apogee: 35.286 km |
| GTO Single | Single Launch | A62 | 15 | Perigee: 180 – 250 km<br>Apogee: 35.286 / 36.286 km |
| Escape Lagrange | Single Launch | A64 | 24 | Apogee: 1.300.000 km |
| Lunar Transfer | Single Launch | A64 | 24 | Apogee: 350.000 km<br>Incl. 18° to 29° |
| Escape Mars | Single Launch | A64 | 24 | $V_\infty$: 2 to 4 km/sec |
| Escape Mercury | Single Launch | A64 | 24 | $V_\infty$: 3.8 to 4.2 km/sec |
| Escape Jupiter | Single Launch | A64 | 24 | $V_\infty$: 3 to 4 km/sec<br>Launch declination 0° to -5° |
| GTO / Escape | Dual Launch | A64 | 24 | Upper S/C as GTO<br>Lower S/C as for * or ** or *** |
| SSO | Single Launch | A62 | 24 | Circular orbit<br>600 – 800 km<br>Inclination: 96° - 99° |
| GEO mission | Single Launch | A64 | 15 | Circular orbit<br>35.286 / 36.286 km<br>Incl. 0° |
| GTO-OURS | Dual Launch | A64 | 6 | Upper S/C and Lower S/C as GTO<br>(with a higher performance reached using statistical reserves down to 50% of exhaustion probability of US) |
| GTO/GTO+OURS | Dual Launch | A64 | 6 | Upper S/C and Lower S/C as GTO/GTO+<br>(with a higher performance reached using statistical reserves down to 50% of exhaustion probability of US) |
| GTO/SuperGTO+ | Dual Launch | A64 | 15 | Upper S/C as GTO (with incl. 6°)<br>Lower S/C Super-synchron 24 hours 6° with perigee of 3000 km |

# Ariane-6 Missionization Parameters

| Category | Description |
|---|---|
| Configuration | Data pertaining to a specific family of the vehicle and which does not change between missions using the same version of the vehicle (e.g., number of boosters, amount of liquid propellant, service modules) |
| Mission | Data which in principle changes from mission to mission (e.g., mass of the payload) |
| Invariant | Data that never changes from mission to mission and regardless of the vehicle being used (e.g., universal constants). Nevertheless, it is listed here for sake of completeness |

| Family | Sub-Family | Description |
|---|---|---|
| GNC<br>Guidance, Navigation and Control | NAV | Navigation |
| | GUI | Guidance |
| | RCS | RCS Control |
| | FM | Flight Management |
| | MEC | Mechanisms |
| | TVC | Thrust Vector Control |
| | TVC-P120C | TVC for SRMs |
| | TVC-EPC | TVC for main cryogenic stage |
| | TVC-ESC | TVC for upper stage |
| | RCS-EPC | RCS for Exo-EPC phase |
| | RCS-ESC | RCS for ESC phase |
| VMS<br>Vehicle Management Software | VM | Vehicle management |
| | FDIR | Fault Detection, Isolation and Recovery |
| | TLM | Telemetry data to be transmitted during the mission. |
| SIM<br>Simulation of Vehicle and Environment | USV | Vehicle configuration |
| | ENV | Environment configuration |
| GS<br>Ground Segment | GS | Data needed for missionisation of Ground tools and generate the FED files, etc |

# Ariane-6 Missionization Parameters List

| Category | Family | Sub-family | Name | Description | Comment |
|---|---|---|---|---|---|
| Configuration | GNC | FM | C_TIME_PITCHOVER_START | Time for Pitch Over Manoever | Associated to cleareance of launch-pad. Can change between configurations |
| Configuration | GNC | GUI | C_P120C_NUMBERS | Number of P120-C boosters | 2 for A62 , 4 for A64 |
| Configuration | GNC | GUI | C_FAIRING_MASS | Fairing mass | Different fairings for different A6 configurations |
| Configuration | GNC | RCS | C_RCT_ANGLE | Thruster direction angle | RCS Thrusters direction could evolve towards different configuration |
| Configuration | GNC | RCS | C_RCT_POSITION | Thrust position | RCS Thrusters position could be modified |
| Configuration | GNC | RCS/TVC | C_FAIRING_INERTIA | Fairing Inertia (matrix) | Different fairings for different A6 configurations |
| Configuration | GNC | RCS/TVC | C_FAIRING_COG | Fairing CoG | Different fairings for different A6 configurations |
| Configuration | GNC | TVC | C_ESC_PROPMASS | Upper stage total liquid propellant | ESC total amount of stored propellant could change depending on the launcher configuration |
| Configuration | GNC | TVC | C_SYLDA_MASS | SYLDA Mass | SYLDA, the A6 multipayload adapter could have different configuration |
| Configuration | GNC | TVC | C_SYLDA_COG | SYLDA COG | SYLDA, the A6 multipayload adapter could have different configuration |
| Configuration | GNC | TVC | C_SYLDA_INERTIA | SYLDA Inertia (mass) | Different payload adapters for different A6 configurations |
| Configuration | GNC | TVC | C_PAYLOAD_ADAPT_MASS | Payload Adapter Mass | Different Payload Adapters could be considered for different launcher configuration |
| Configuration | GNC | TVC | C_PAYLOAD_ADAPT_COG | Payload Adapter CoG | Different Payload Adapters could be considered for different launcher configuration |
| Configuration | GNC | TVC | C_PAYLOAD_ADAPT_INERTIA | Payload adapter Inertia | Different Payload Adapters could be considered for different launcher configuration |
| Configuration | SIM | USV | C_MAX_HEATRATE_LF | Maximum Heat Rate Long Fairing | Total heat rate is a parameter normally shaping the trajectory optimisation. It could depend on the fairing and hence on the Launcher configuration |

# Study Case: SPARTAN

# SPARTAN-FP7

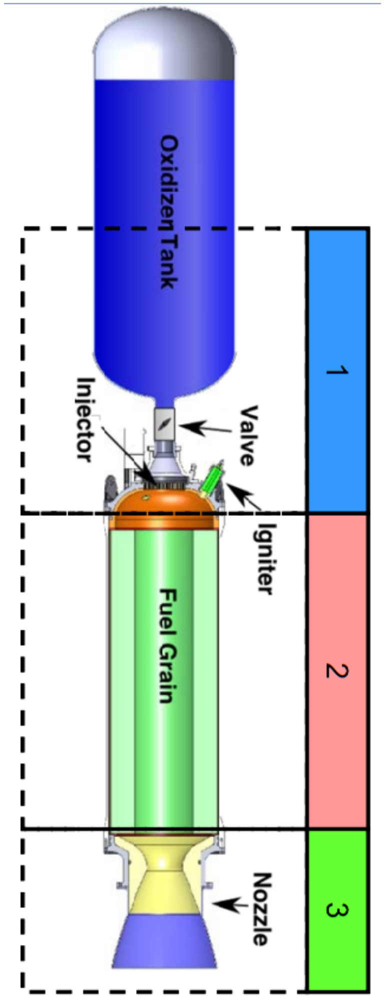



Figure 1: Perspective view.

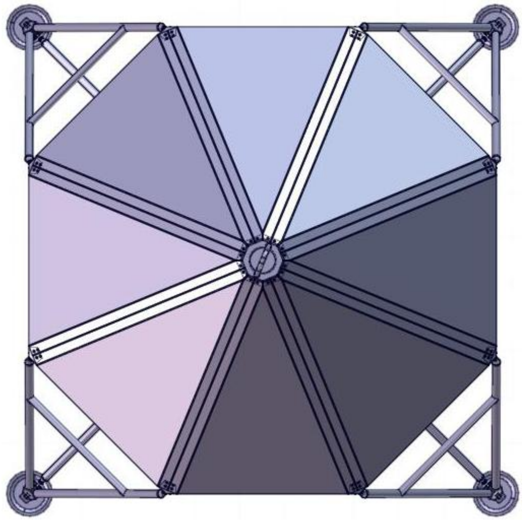
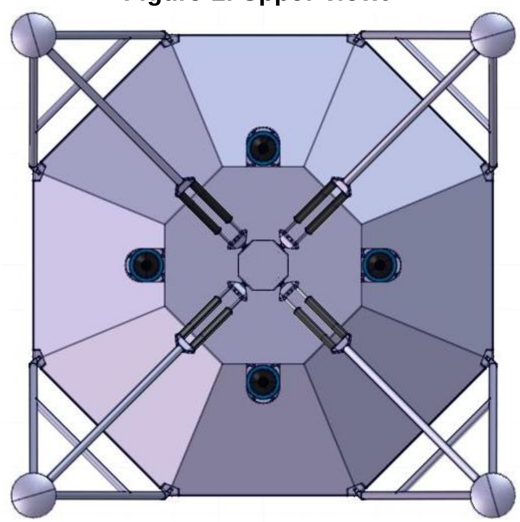
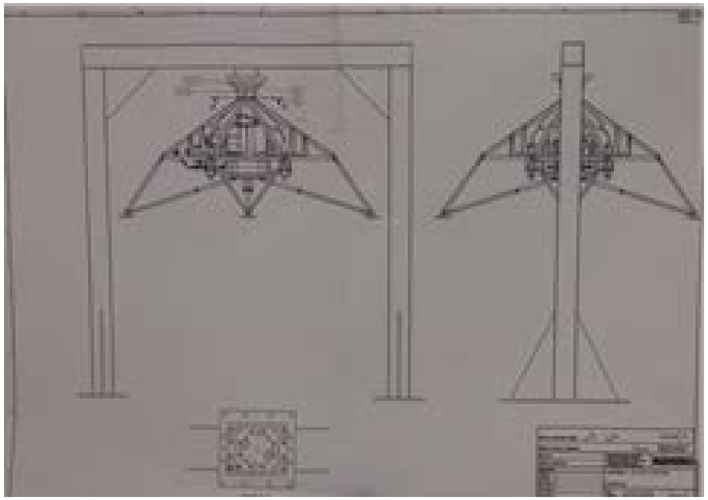


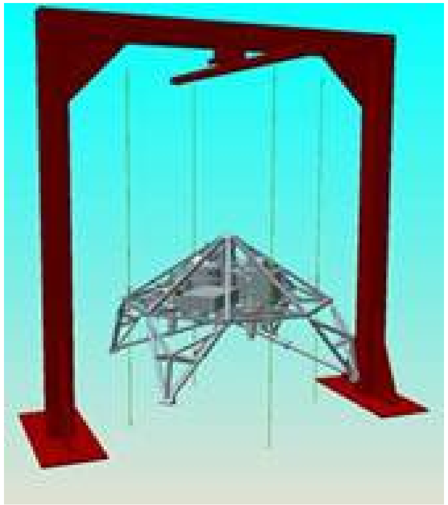
Figure 2: Upper view.

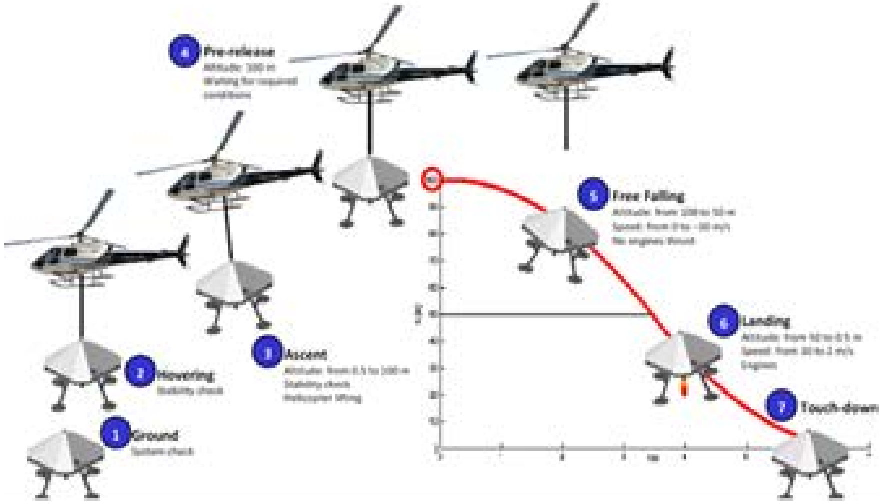


# Several Tests and a unique OBSW







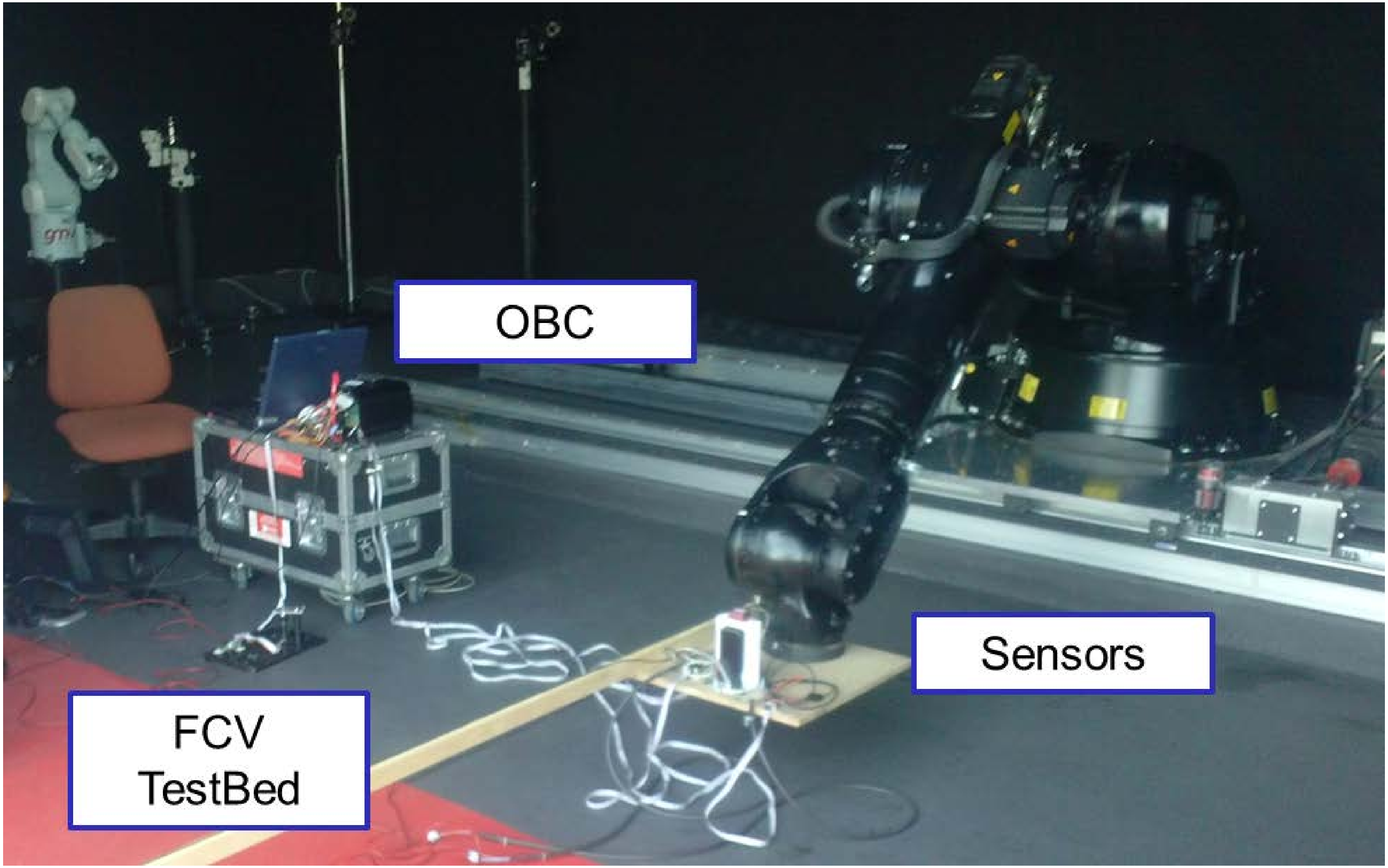

 

# Elements

# Intermediate Validation

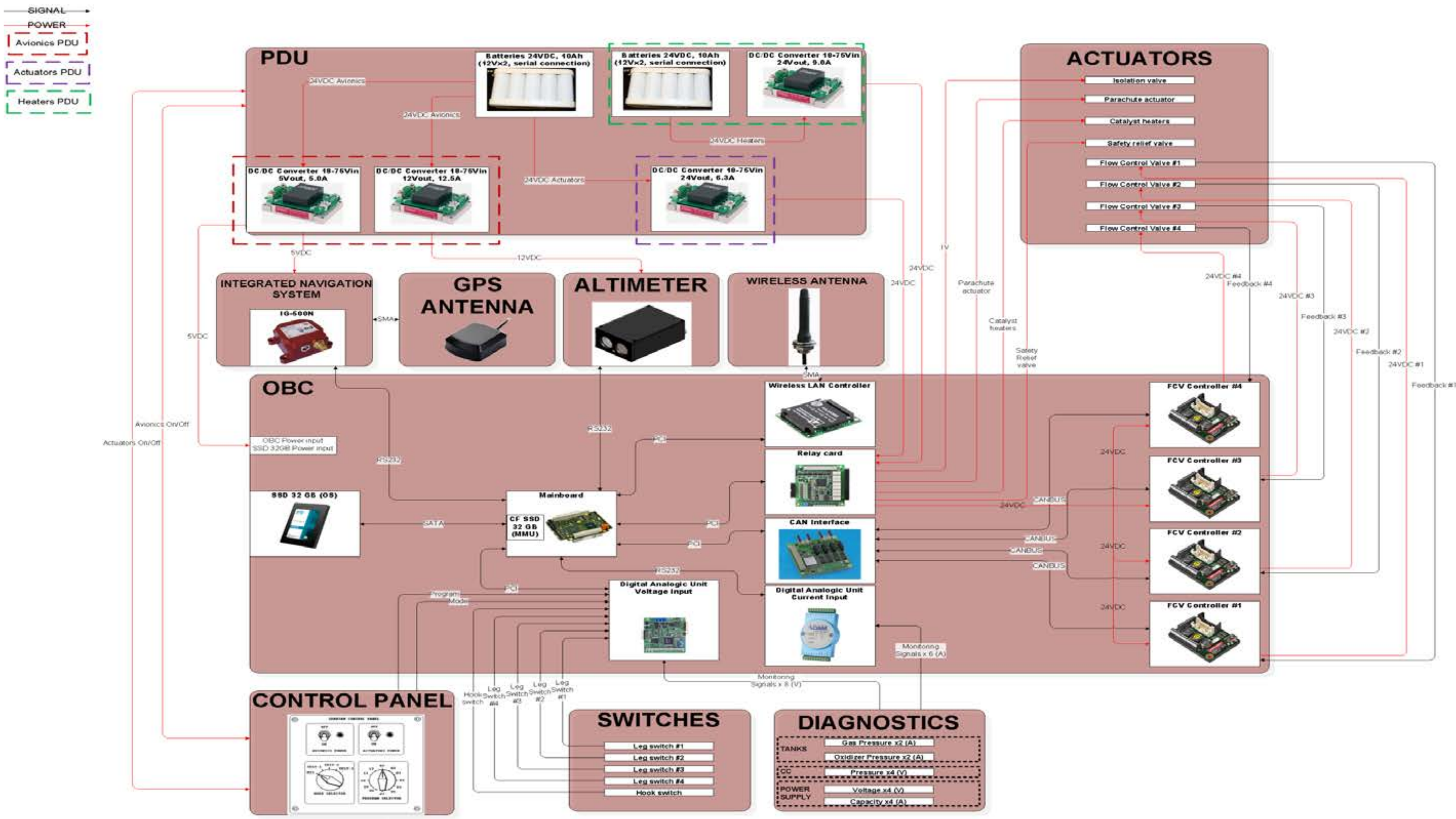
 

# Intermediate Validation

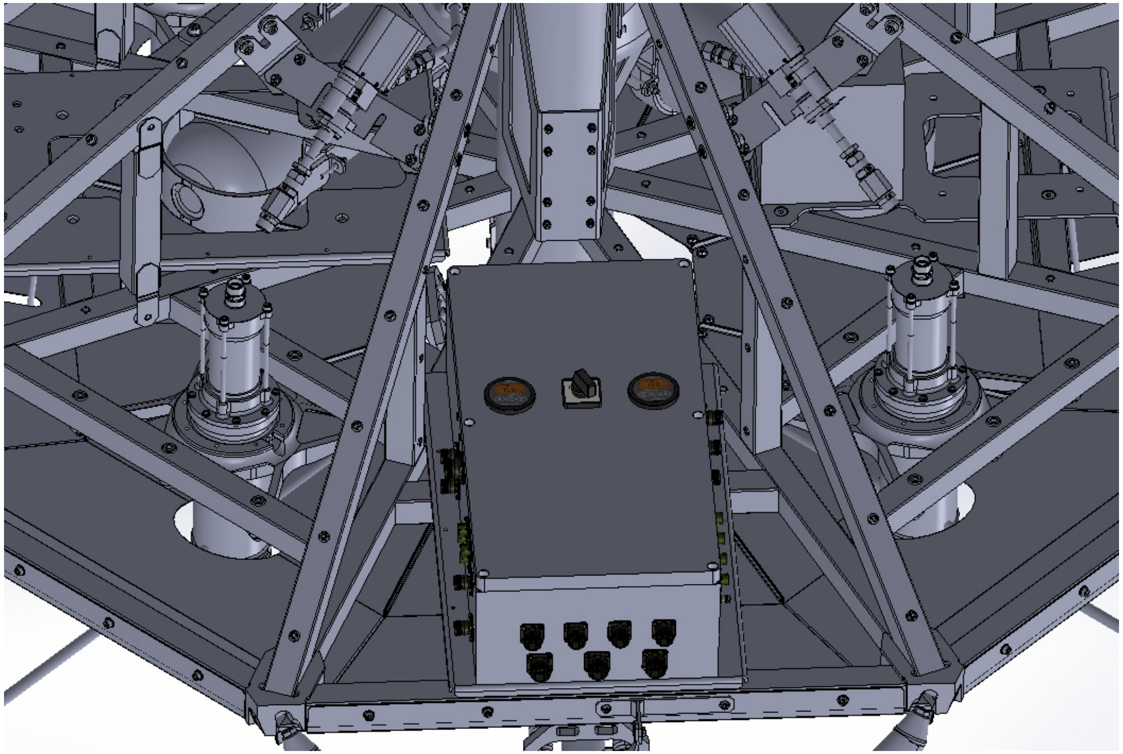


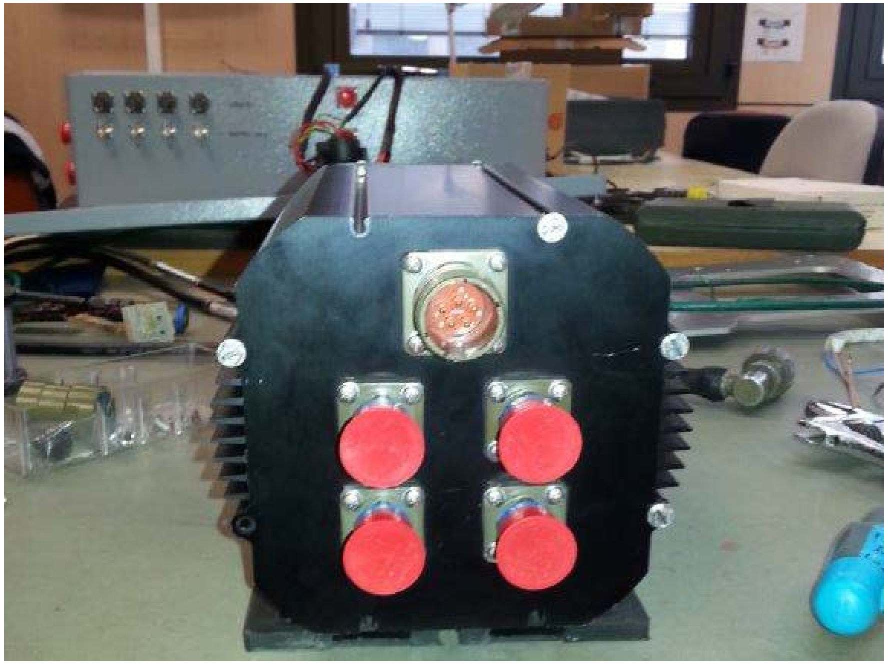
# Avionics Architecture



# Avionics Box

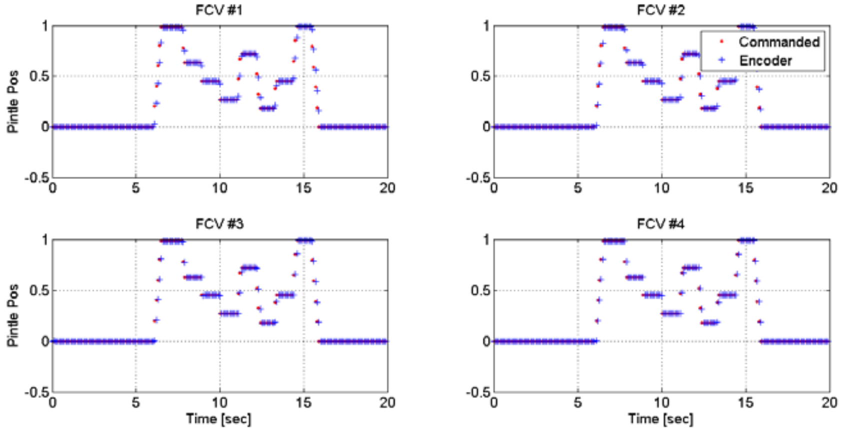



# OBC



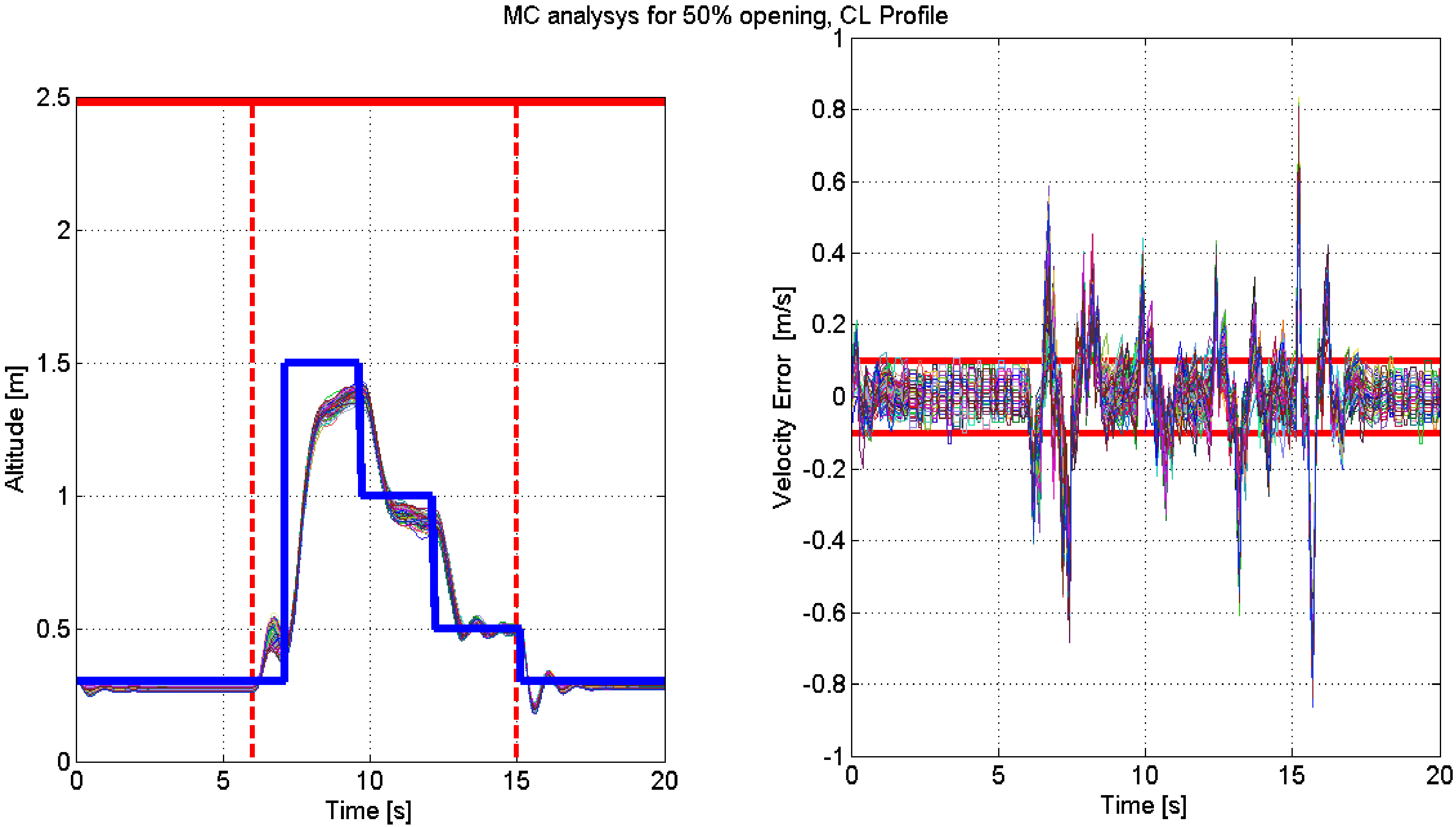
 

# Static Tests



# TR4 Simulation in FES

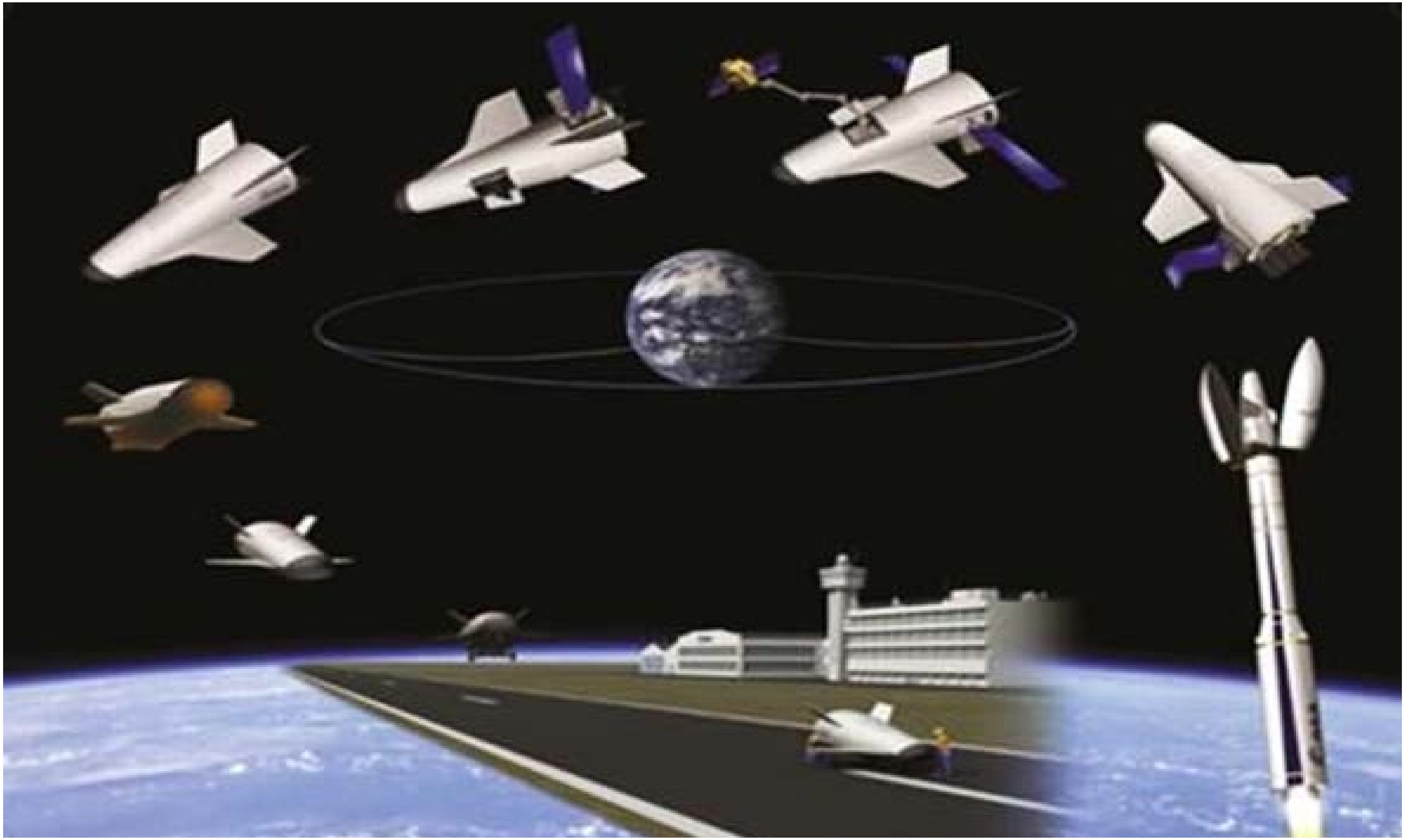


# SPARTAN Parameters List

| Category | Family | Sub-family | Name | Description | Comment |
|---|---|---|---|---|---|
| Configuration | CTR | THR | uu | FCV throttling steps table | Configurable parameter – changing it will result in control redesign |
| Configuration | CTR | THR | RR | Grain regression table | Configurable parameter – changing it will result in control redesign |
| Configuration | CTR | THR | CC | FCV throttling steps percentage of nominal thrust table | Configurable parameter – changing it will result in control redesign |
| Configuration | FM | FM | N | Flight software frequency | It can change depending on mission requirements but it would lead to a complete re-evaluation of the GNC. |
| Configuration | GUI | GUI | eps | FCV commands quantization | Configurable parameter – changing it will result in control redesign |
| Configuration | GUI | GUI | dC_max | FCV commands rate limiter | Specific valve characteristic |
| Configuration | Time | Sample | dt | Sample Time | It can change depending on mission requirements but it would lead to a complete re-evaluation of the GNC. |
| Invariant | Body | Body | pos | Position of the fuel and oxidizer tanks in the lander body frame | Fixed for every mission |
| Invariant | Const | Const | PI | PI constant divided by 180 | Constant |
| Invariant | CTR | FUE | L | Length of solid fuel in tank | Fixed for every mission (engine characteristic) |
| Invariant | CTR | FUE | R0 | Internal radius of solid fuel | Fixed for every mission (engine characteristic) |
| Invariant | CTR | FUE | R1 | External radius of solid fuel | Fixed for every mission (engine characteristic) |
| Invariant | CTR | FUE | rho | Density of solid fuel | Fixed for every mission (engine characteristic) |
| Invariant | CTR | FUE | a | Nozzle efficiency parameter | Fixed for every mission (engine characteristic) |
| Invariant | CTR | FUE | n | Nozzle efficiency parameter | Fixed for every mission (engine characteristic) |
| Invariant | CTR | OXI | dm_max | Maximum mass flow rate of oxidizer | Fixed for every mission (engine characteristic) |
| Invariant | EPS | EPS | eps | Epsilon value used for machine tolerance | Fixed for every mission |
| Invariant | GUI | CL | r | Engine distance from the body reference system | Fixed for every mission |
| Invariant | GUI | CL | g | gravity | Physical constant |
| Invariant | INF | INF | inf | Infinity value used for machine tolerance | Fixed for every mission |
| Invariant | NAV | NAV | g0 | Value of the local gravity acceleration | Physical constant – vector |

# Study Case: PRIDE



# IOSV Study Case - PRIDE

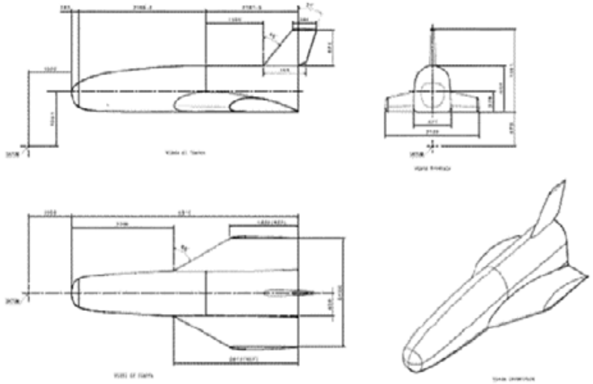
- **System Description**
  - Vehicle – PRIDE ( High L/D re-entry follow-up of IXV)
    - From Phase A for Vega-launched demonstration
    - General description – shape, mass, dimensions
    - AOCS / GNC Equipment – assumes ADR package in cargo bay (<50kg)
  - Mission – Active Debris Removal
    - Adapt baseline to deorbiting small (~150 kg) satellites using net system (looser contraints on rv accuracy, passive detumbling)
    - Assume high degree of autonomy , INS+GNSS-based navigation
  - SW Architecture / GNC definition
    - Adapted from IXV
    - GNC modes for LEOP, RV, Re-entry defined

- **Instantiation**
    - Sample list of Missionization /Configuration/Invariant Variables
    - Divided into Families / Subfamilies

# System Description

- **Vehicle – PRIDE USV3**
  - – From Phase A for demonstration LEO mission:
    - Dimensions – 5.31 m long, 2.2 m wide (VEGA)
    - Wet mass: 1600 Kg, dry 1440 Kg
    - L/D : hypersonic 1.2 , subsonic 4
    - ADR package in cargo bay (max 50 kg) : DVS camera plus Net system – ANdROiD/PATENDER heritage)

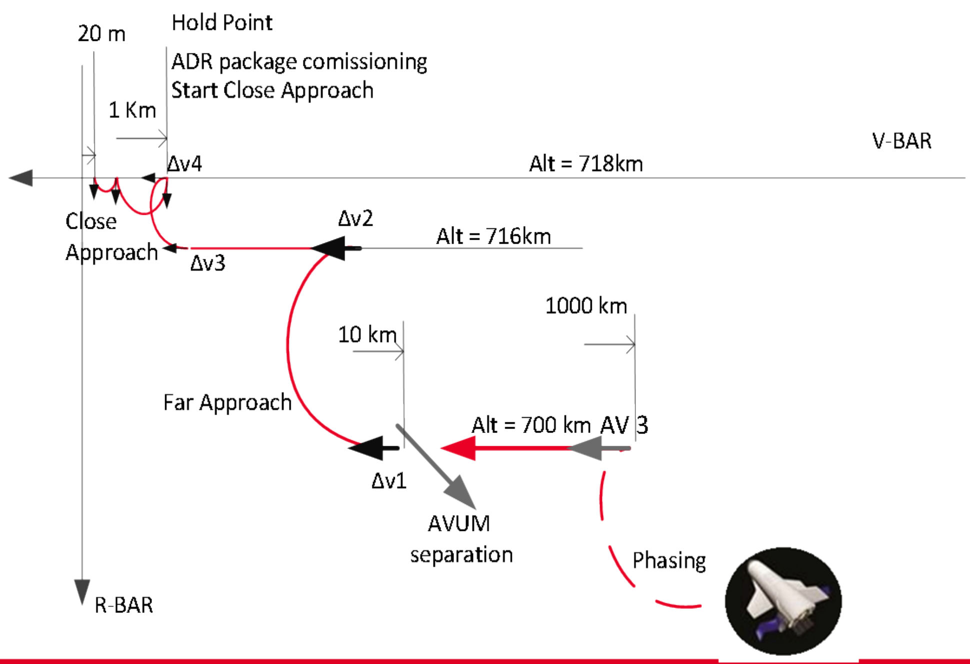


| | |
|---|---|
| **RCS** – proximity manoeuvres, orbital and hypersonic re-entry | Set of **6 hydrazine 400 N Thrusters** |
| **ASCS** - active during late atmospheric re-entry and landing. | Derived from the IXV Flap Control System |
| Main **navigation sensor** for all-round orbital operations and re-entry. | Navigation-Grade Inertial with **Tight GPS / INS** integration, including differential GPS for autonomous landing, eg.: Honeywell SIGI (COTS) X37 |
| **Sensor for Proximity/Rendez-vous** | Navigation Camera with integrated image processing (eg.: DVS from TSD – PRISMA heritage) |
| **Sensor for TAEM/Approach and Landing** | Barometer / Radar Altimeter / dGPS |

# System Description

- **Mission–** (very autonomous) **Active Debris Removal**
  - **Pre-launch, LEOP and Phasing** with AVUM to trailing orbit
    - Sample target: Proba 2 – 150 kg , in SSO)
  - **Rendez-vous and capture** concept from ANdROiD (scaled):
    - Far Range approach to 1 km supported by GPS+INS.
    - Hold point and ADR package comissioning – *smart* camera and net capture
    - Close Range (hoping) supported by camera to ~30 m from target
    - Capture via net (10x10 m system), passive detumbling
    - Hold point and ADR package comissioning – *smart* camera and net capture
    - Close Range (hoping) supported by camera to ~30 m from target
    - Capture via net (10x10 m system), passive detumbling
    - Deorbiting via series of perigee lowering burns (tethered)
    - Separation and insertion into re-entry ballistic arc



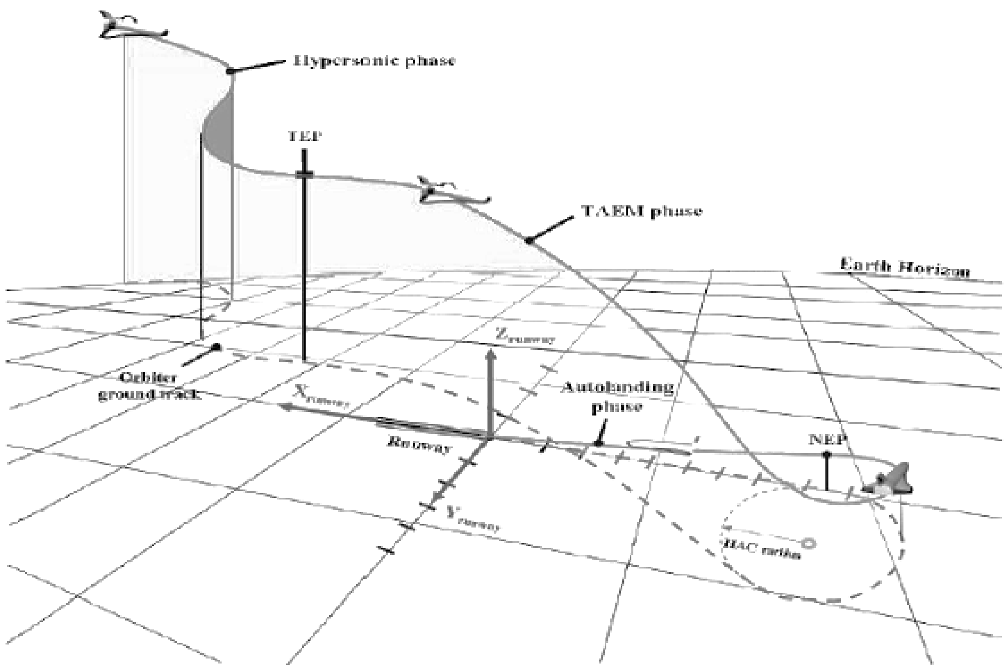

 

gmv INNOVATING SOLUTIONS

# System Description

– **Re-Entry and Auto-Landing:**

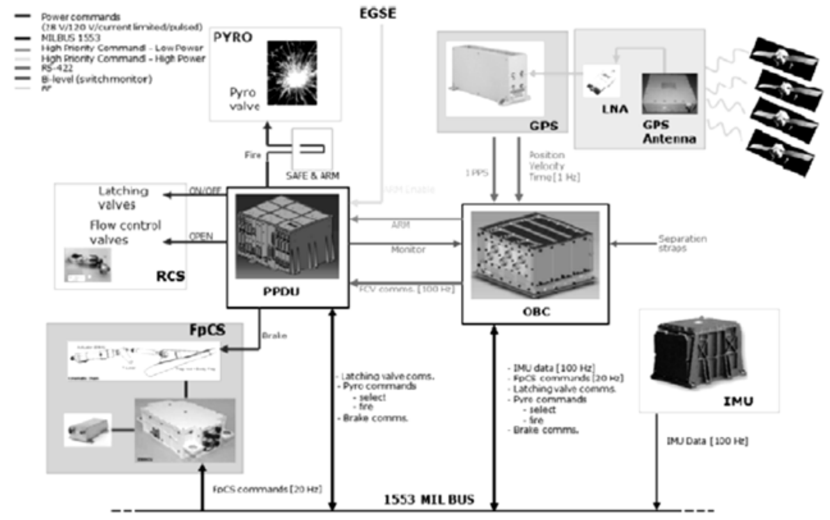
- **Hypersonic re-entry** - 120 km to 20 km, Mach 20 to Mach 2, high AoA
  - The GNC minds thermodynamic and mechanical constraints.
  - Attitude controlled through RCS thrusters.
- **Terminal Area Energy Management** (TAEM) – from TAEM Entry Point (TEP), to Mach 0.5, the Nominal Exit Point (NEP)
  - S-turns to dissipate potential and kinetic energy
  - Aligns the vehicle's heading with the runway
  - Navigation radar and GPS/dGPS , still RCS-based attitude, lower AoA
- **Aproach and Landing** (A/L) - steep/shallow glideslopes, a parabolic pullup and final sink rate control. ASCS actuated (similar to conventional aircraft)

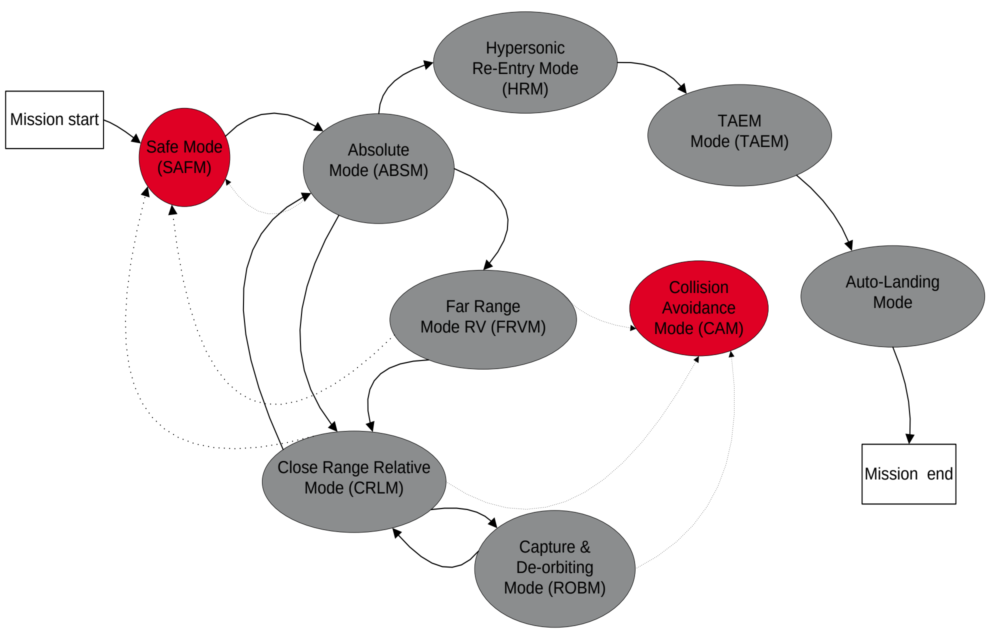


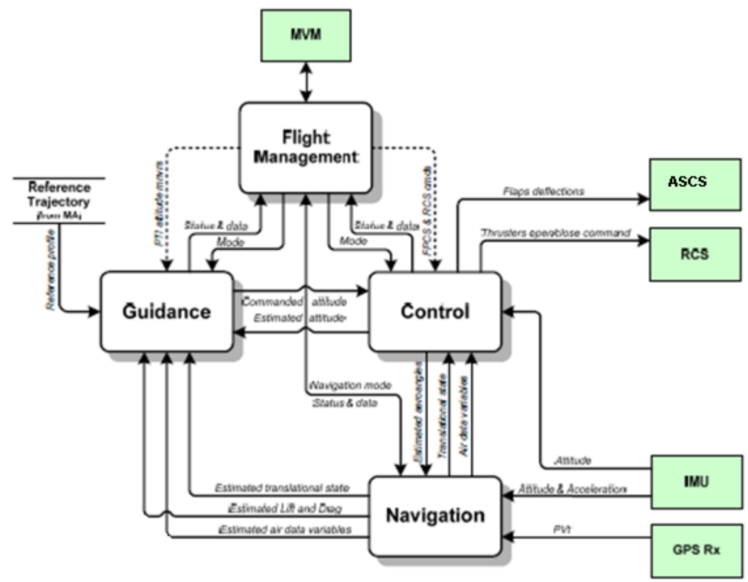

 

# High Level Software Architecture

- Data Handling Subsystem – IXV
- Tasks: GNC, Recording, Transfer
- Supported by: OBC, INS/GPS, RV and Re-entry sensors, DAU, Recorders, connected through 1553 MIL bus
- Actuates: RCS, ASCS
- **GNC**
  - Flight Manager: autonomous program (eg.: PROBA heritage) and monitoring
  - Guidance and Navigation: Orbital/Absolute ; Relative/RV; Re-entry



# Flight Modes

| Mode | Starting Activites | Navigation | Guidance and Control |
|---|---|---|---|
| **Absolute** | During Launch, Phasing (AVUM-controlled) and in the re-entry insertion orbital arch. | Absolute based on INS+GPS (when available). | Initial: LEOP - Passive<br>( Attached to launcher / service module)<br>Final: re-entry ballistic arch – absolute guidance and slew ( RCS ) |
| **Far Range Rendez-Vous** | From AVUM separation until 1 Km from target. | Absolute/Relative based on INS+GPS and updated TLE of target | Scheduled impulsive Hohmann transfers (RCS) |
| **Close Range Rendez-Vous** | Camera acquisition. Comissioning of ADR payload. | Relative based on INS+GPS and VBS integration. | Impulsive hops to 20 m from target. Pointing slew (RCS). |
| **Capture & De-Orbiting** | Capture and passive detumbling. | Absolute/Relative based on INS+GPS and VBS integration. | Series of impulsive perigee lowerings. |
| **Hypersonic Re-Entry** | At EIP, 120 km altitude, end of ballistic arc (done in absolute mode) | Re-entry mode, based on INS+GPS and aerodynamic databases. Outputs states in inertial and atmosphere-relative frames. | Track drag-speed profile / attitude profile for mechanical and aerothermodynamical constraints. |
| **TAEM Mode** | From TAEM conditions (~Mach 2) | Re-entry mode, based on INS+dGPS, barometric, radar and GPS/dGPS. Possibly ILS. | Dissipate kinetic and potential energy, align with track, through RCS and ASCS. |
| **A/L Mode** | From NEP (~Mach 0.5) | | Automated landing sequence( sink rate acquisition, glideslope), ASCS-only. |

# PRIDE Data

- Sample of data from each
  - Category ( Configuration / Mission / Invariant)
    - Family
      - Subfamily

| Family | Sub-Family | Description |
| --- | --- | --- |
| **GNC Guidance, Navigation and Control** | NAV | Navigation |
| | GUI | Guidance |
| | RCS | RCS Control |
| | ASCS | ASCS Control |
| | FM | Flight Management |
| | MEC | Mechanisms |
| **VSW Vehicle Management Software** | VM | Vehicle management |
| | FDIR | Fault Detection, Isolation and Recovery |
| | TLM | Telemetry data to be transmitted during the mission. |
| **SIM Simulation of Vehicle and Environment** | USV | Vehicle configuration |
| | ENV | Environment configuration |
| **GS Ground Segment** | GS | Data needed for missionisation of Ground tools and generate the FED files, etc |

# PRIDE Data

- Example

| Category | Family | Sub-family | Name | Description |
|---|---|---|---|---|
| **Invariant** | GNC | GUI | I_G0 | Gravitational Acceleration |
| **Mission** | GNC | GUI | M_DEORBIT_ORBITS | Intermediate orbital elements as targets of perigee lowering burns |
| **Configuration** | GNC | GUI | C_Q_MAX | Maximum heat flux |
| **Configuration** | GNC | RCS | C_THRUSTER_LOCATION | Location of thrusters in vehicle frame |
| **Configuration** | GNC | RCS | C_THRUSTER_DIRECTION | Direction of thrusters in vehicle frame |
| **Mission** | GNC | RCS | M_MASS_CONSUMPTION_INIT | Initial estimated mass consumption |
| **Mission** | GNC | FM | M_ALTTITUDE_SWITCH | Altitude values to switch modes that are activated via altitude |
| **Mission** | GNC | FM | M_DRAG_SWITCH | Drag values to switch modes that are activated via drag |
| **Mission** | GNC | FM | M_ENTRY_ALTITUDE | Higher altitude to enable DRS triggering computation |
| **Mission** | VSW | VM | M_VEHICLE_MANAGEMENT_FLAG | Table with the values of the vehicle mode management flag |
| **Mission** | VSW | FDIR | M_FDIR_STATUS | Status flag for FDIR |
| **Mission** | VSW | TLM | M_DATA_SELECT | Identifiers of data batches to be transmitted to ground |
| **Configuration** | GNC | NAV | C_IMU_QUANTIZATION | IMU quantization step |
| **Mission** | GNC | NAV | M_MASS_INIT | Initial total mass |
| **Configuration** | SIM | USV | C_AEDB_DRAG | Drag lookup table (drag coefficient from Mach, AoA, Reynolds) |
| **Mission** | SIM | ENV | M_WIND_MODEL | Wind model table |
| **Mission** | GS | GS | M_STATION_MIN_ELEV | Minimum elevation for tracking with each station |
| **Configuration** | GNC | NAV | C_IMU_QUANTIZATION | IMU quantization step |
| **Mission** | GNC | ASCS | M_DYN_PRESSURE_RANGE | Range limits for the dynamic pressure |
| **Mission** | GNC | ASCS | M_VELOCITY_LIMITS | Velocity range limits |

VEMIPRO-FR

# Missionization Tool Prototype

INNOVATING SOLUTIONS

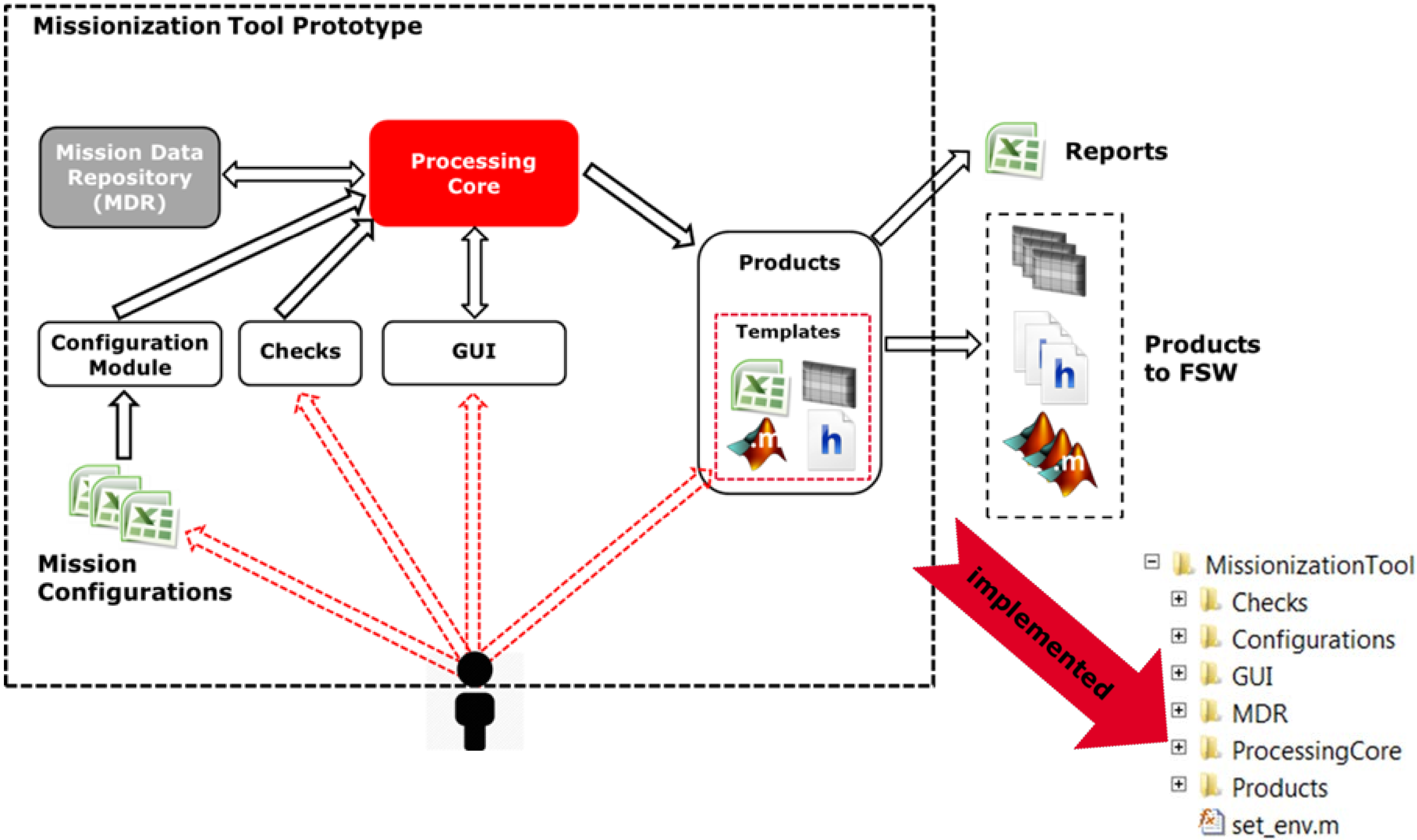
# Prototype Implementation Main Goals

- The main **goals** of missionization tool prototype are:
  - Demonstration of the missionization **main use cases**:
    - Insert new mission data
    - Update existent mission data
    - Generate FSW missionized files for running the FSW in the correspondent facility
    - Generate report of a mission
    - Manage Repository Configuration
    - Perform external user defined checks on the mission data
  - First attempt of missionization tool **implementation** and **configurations** definition
  - First definition of **formats**, **templates** and **products** generation
  - First definition, management and implementation of external user defined **checks**

# Prototype Implementation Architecture

- High level architecture of prototype



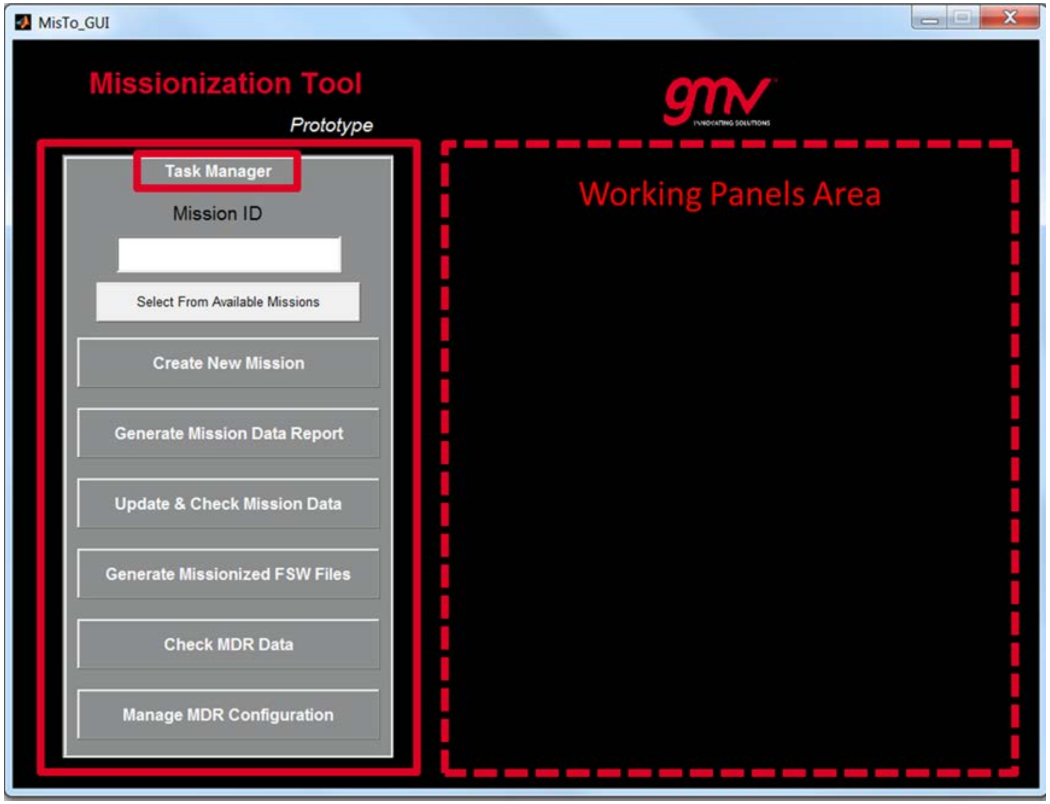
# Missionization Tool Prototype - DEMO

Prototype environment requirements:

- *Matlab (e.g. R2012a)*
- *TortoiseSVN*

Data has been provided for the following missions:

- *ARIANE6*
- *PRIDE*
- *SPARTAN*
- *VEMIPRO (defined in TN01 VEGA like)*

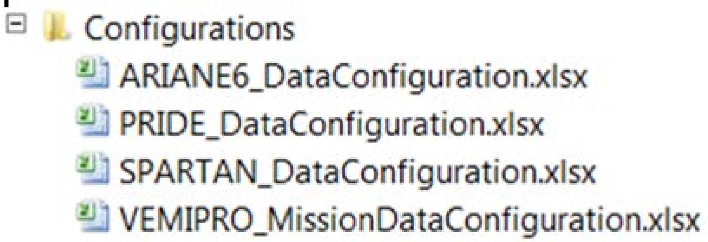


# Prototype Modules – 1/3

## Configurations

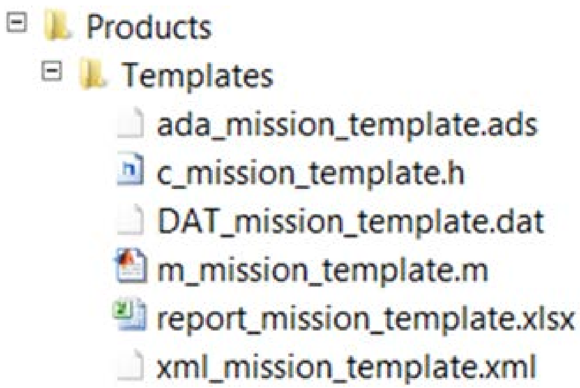
This module contains the configuration files that report the missionization parameters (ARIANE6, PRIDE, SPARTAN and VEMIPRO configuration defined)




## Products

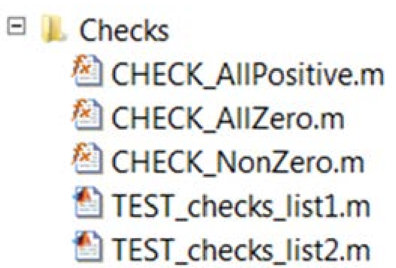
This module contains all the files generated as output of the missionization tool prototype activities (e.g. reports, ASCII files, h files, binary files, etc.). It accounts for templates to generate the products, such as:

- *Matlab mission template*
- *DAT mission template*.
- *XML mission template*.
- *C mission templates*.
- *Report template*.
- ADA mission template




## Checks

This module contains all the Matlab functions that implement user defined checks and scripts containing a lists of pre-defined checks



# Prototype Modules – 2 / 3

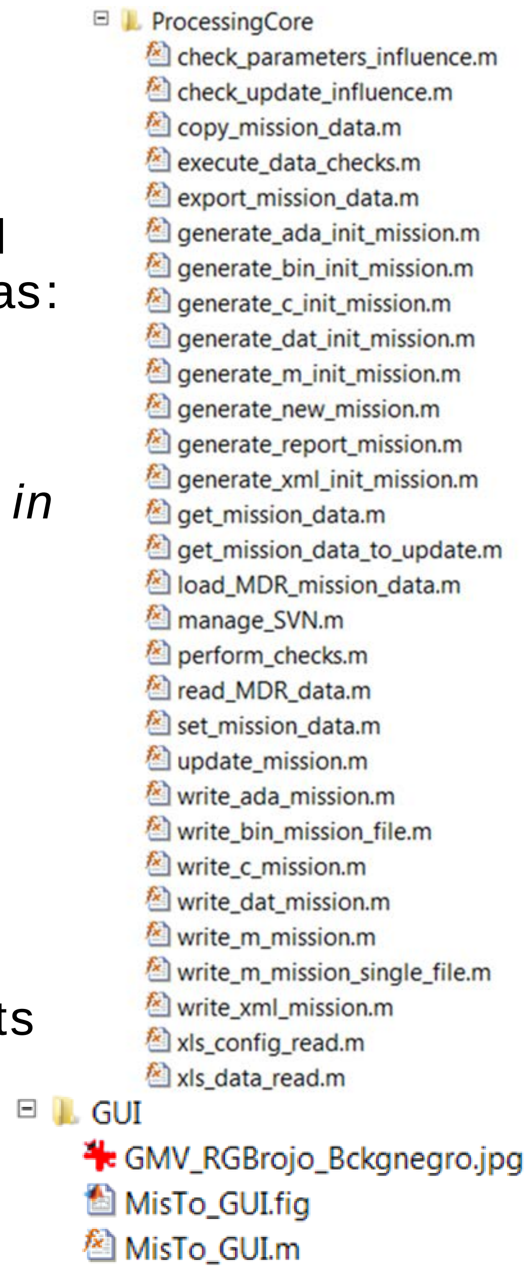
## ProcessingCore

This module contains all the functions for managing all the missionization tool prototype main activities such as:

- *Insert new mission data*
- *Update existent mission data*
- *Generate FSW missionized files for running the FSW in the correspondent facility*
- *Generate report of a mission*
- *Perform checks on mission data*
- *Manage Repository Configuration*

## GUI

This module accounts for a GUI that uses Matlab scripts underneath (e.g. Command Line interface)



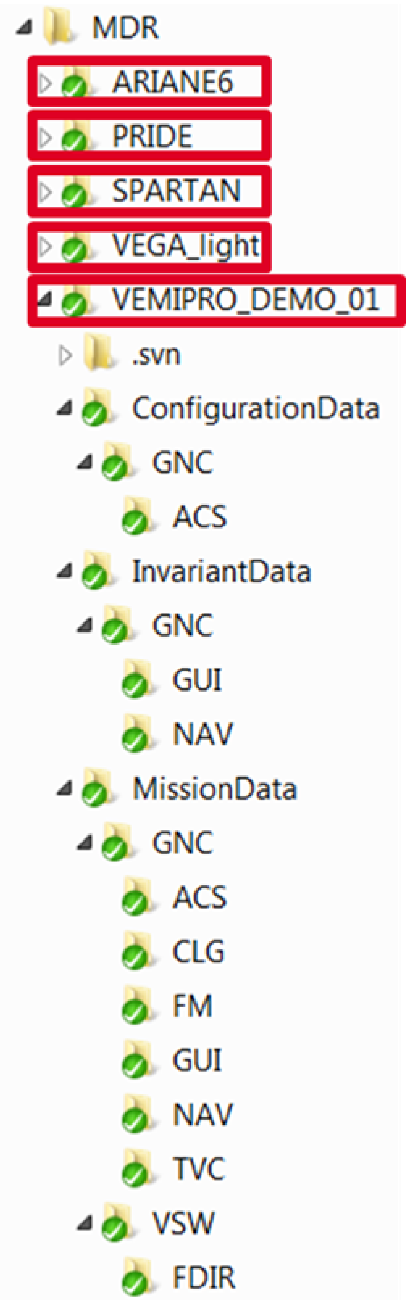
# Prototype Modules – 3 / 3

## MDR

This module contains the mission data repository managed by the tool prototype (under configuration control). The data are organized as directory structure per category, family and sub-family as follows:

- Category (e.g. *MissionData*, *ConfigurationData*, *InvariantData*)
  - Family 1
    - Sub-family 1
      - xml files
    - Sub-family 2
    - ...
    - Sub-family N
  - Family 2
  - ...
  - Family N

The **XML** format is used to store mission data of each sub-family in the correspondent file



# Prototype Implementation Aspects

- **Matlab** has been selected as implementation and scripting tool for the missionization tool prototype
- **SVN** (e.g. TortoiseSVN) has been selected for MDR configuration control
- The SVN is **directly accessed** and **managed** by the prototype GUI
- User **configurable files** and **templates** are used to generate the reports and the missionized FSW files
- The missionization tool prototype can manage **different** missions data and **versions**
- A **GUI** provide the direct access to the prototype functionalities
- A **CLI** can be used for commanding the missionization tool prototype (commands underneath the GUI)
- **Starting point** for a complete and operative missionization tool

# Prototype Implementation Main Restrictions

- The main **restrictions** of missionization tool prototype are:
  - No user permission management is implemented
  - No complete data checks are implemented (e.g. integrity, consistency, checksum, etc.)
  - No complete format and syntax standards are implemented (e.g. numerical formats, exponent-decimal, comma-dot format, data type mapping in C-ADA, etc.)
  - No totally customizable templates for products generation
  - No totally customizable templates for reports generation
  - No complete data type management is implemented

# Thank you

VEMIPRO - TEAM

Space Systems BU

www.gmv.com