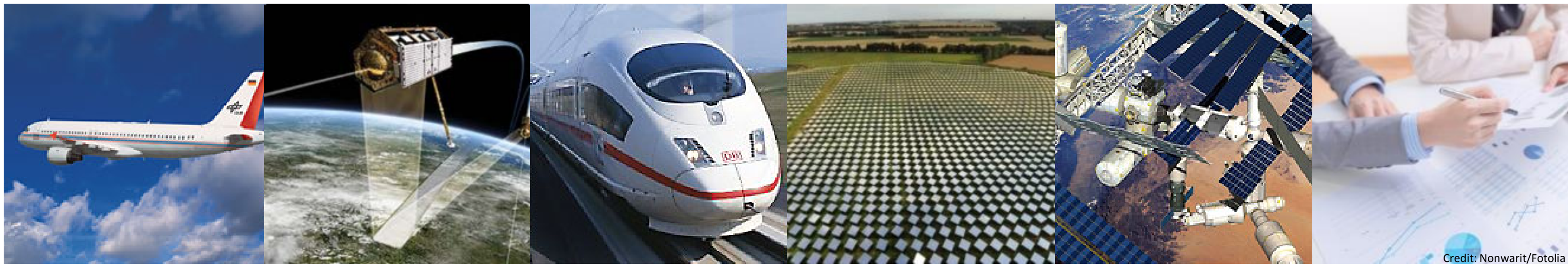
On-board Software Systems

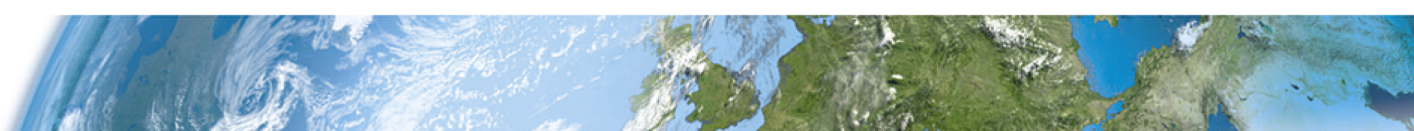
Olaf Maibaum

DLR

German Aerospace Center
One DLR – Three Pillars

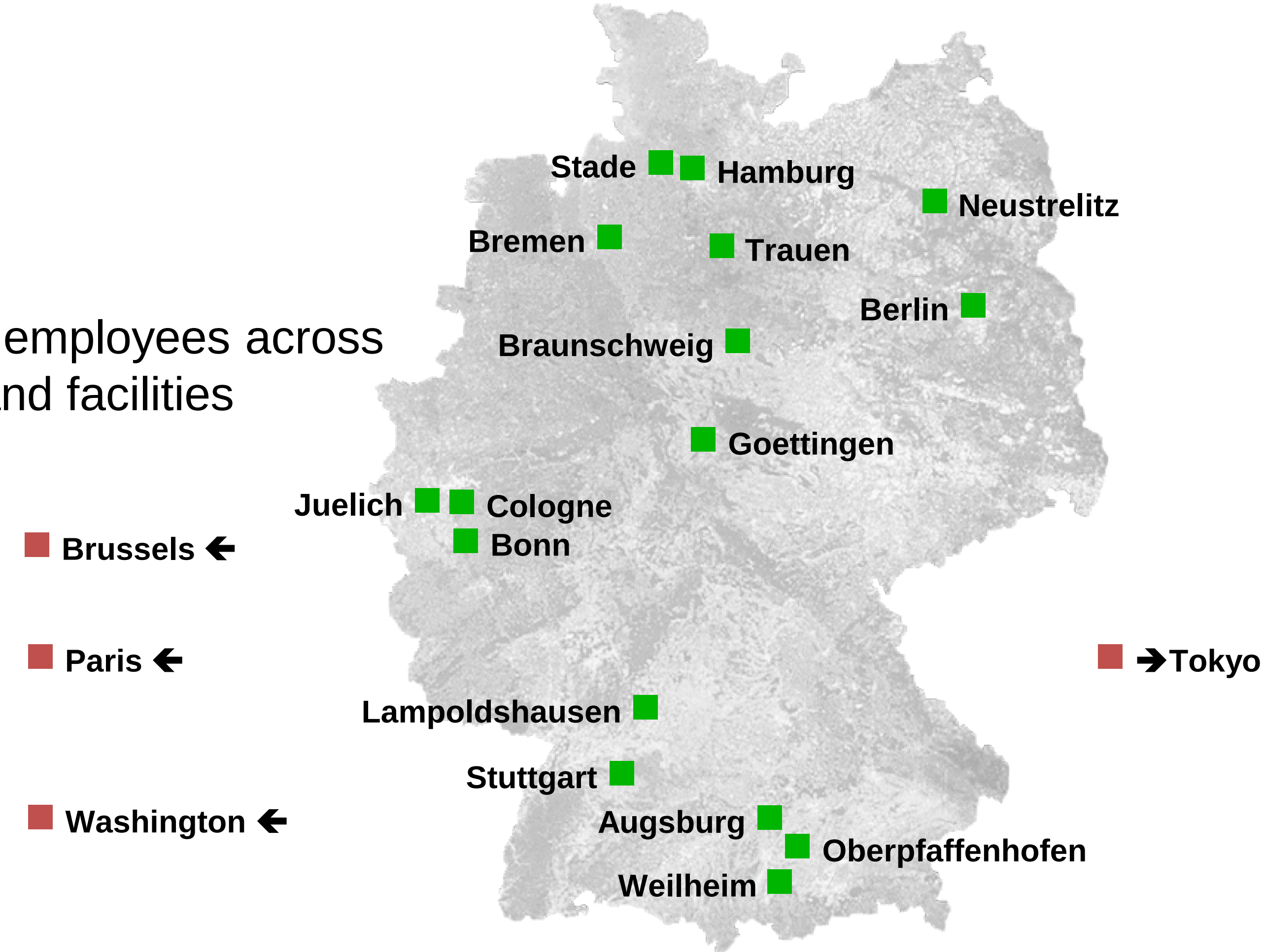


- Research Institution
  - Germany's national research center for aeronautics and space
- Space Agency
  - Responsible for the forward planning and the implementation of the German space program by the German federal government
- Project Management Agency
  - Germany’s largest project-management agency



German Aerospace Center
Locations and Employees

- Locations
  - 16 Sites
  - 4 Offices
- Employees
  - Approx. 8000 employees across 33 institutes and facilities



DLR Simulation and Software Technology
Organization

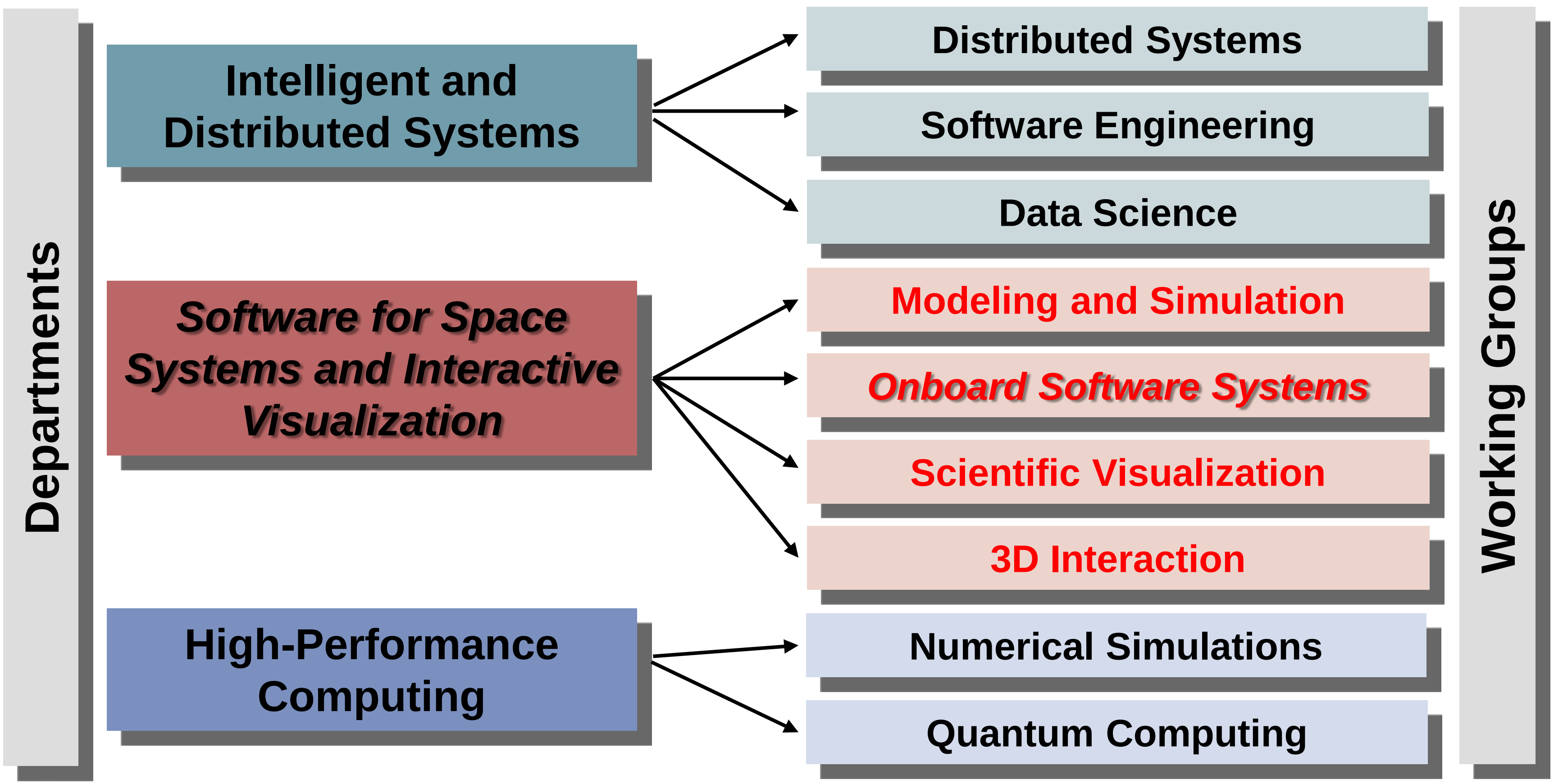
Director: Rolf Hempel

- **Intelligent and Distributed Systems**
  Andreas Schreiber (Cologne/Berlin)
- **Software for Space Systems and Interactive Visualization**
  Dr. Andreas Gerndt (Braunschweig)
- **High-Performance Computing**
  Dr. Achim Basermann (Cologne)

Sites: Cologne, Braunschweig, (Berlin)

DLR Simulation and Software Technology

Scientific Themes and Working Groups



Simulation and Software Technology
CROSS DRIVE – Collaborative Planet Exploration

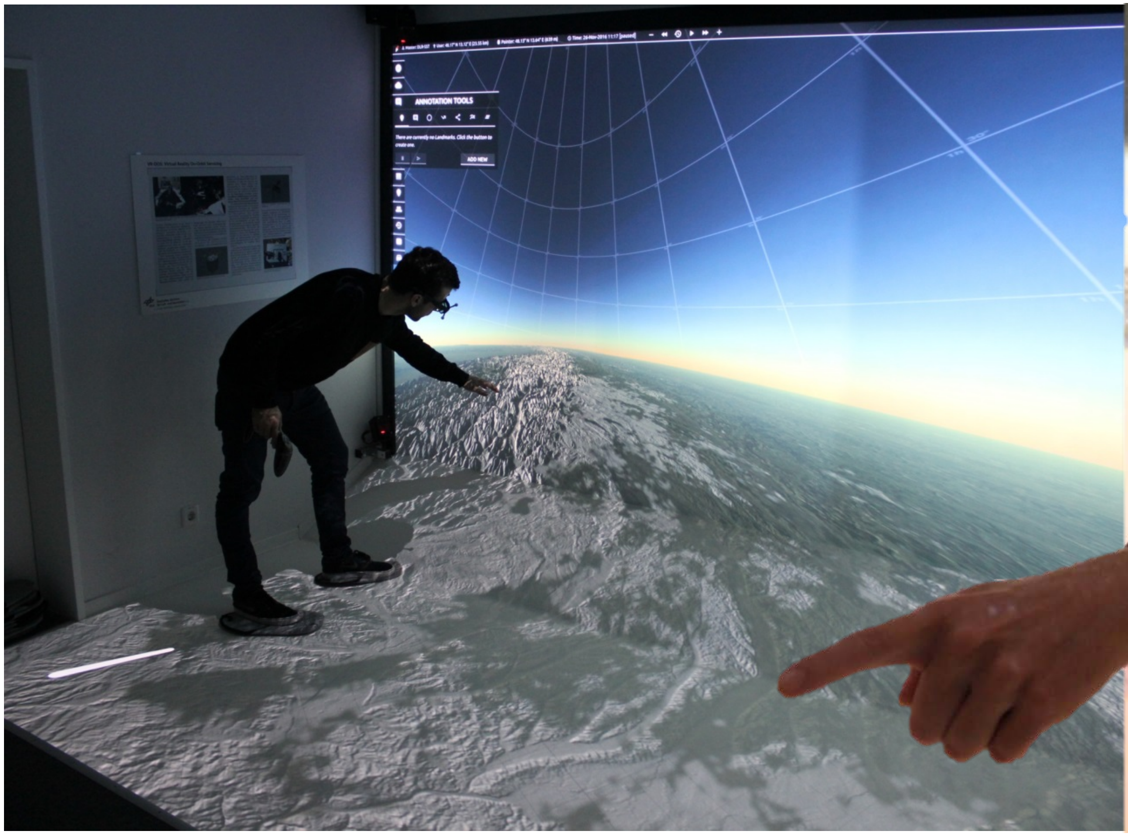
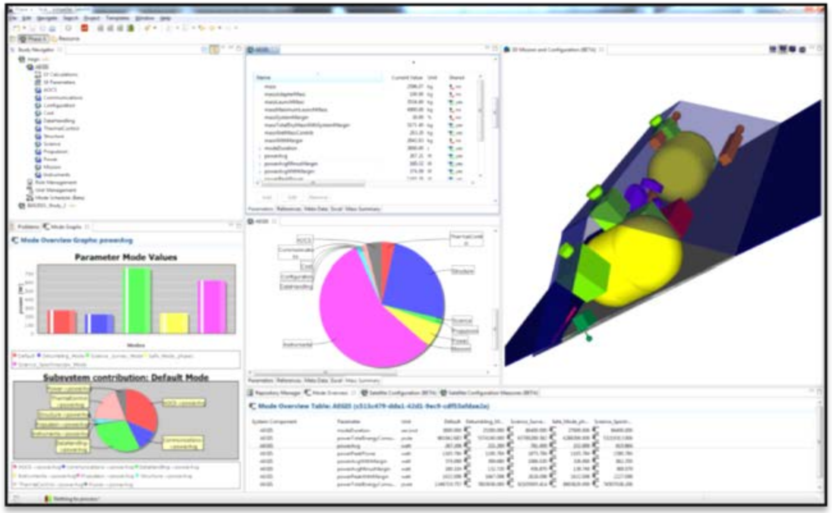


Fig.: Terrain rendering on a multi pipe system

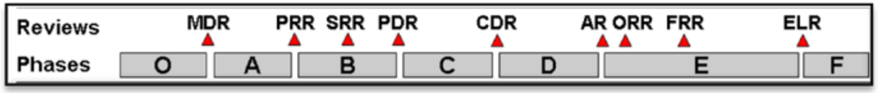
Fig.: Exploring the mars via VR glasses

Space Systems and Interactive Visualization
Group: Modeling and Simulation

- Dedicated Software to support Concurrent Engineering



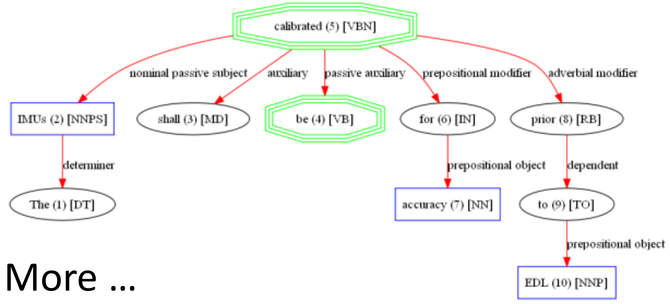
- Covering whole Lifecycle
  - Closing Gaps between Phases



- Model-based Systems Engineering
  - Different Views from One Single Point of Truth
  - Further Benefits
    - Sensitivity Analysis

	$m_0 = 3000kg$			$m_0 = 3kg$		
	m_p	V_t	m_t	m_p	V_t	m_t
m_0	1.00	1.00	1.00	1.00	1.00	1.00
Δv	0.96	0.96	0.96	0.96	0.96	0.96
c	-0.96	-0.96	-0.96	-0.96	-0.96	-0.96
ρ_N2H4	0.00	-1.00	-1.00	0.00	-1.00	-1.00
ρ_titan	0.00	0.00	1.00	0.00	0.00	1.00
p_burst	0.00	0.00	1.00	0.00	0.00	1.00
σ_titan	0.00	0.00	-1.00	0.00	0.00	-1.00

- (Concurrent) Model Checking



- More …

DLR

Space Systems and Interactive Visualization
Group: Onboard Software Systems

Mission:

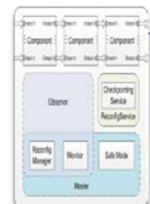
- Provide reliable real-time software for embedded space systems

Research Focus:

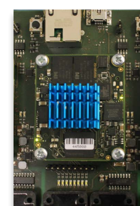
- Model-driven Software Development
- (Real-time) Execution Platforms
- Reconfigurable Distributed Onboard Systems
- Verification/Validation and Quality Assurance

Fig.: TET-1 and BIROS = FireBird

Onboard Applications



Middleware



Operating System

Hardware Driver

DLR

Our Use Cases

- New On-board Computer Architectures
  - OBC-NG
  - ScOSA
- Attitude (and Orbit) Control
  - BIRD
  - OOV-TET
  - BIROS
  - Eu:CROPIS
- Command & Data Handling
  - Eu:CROPIS
  - MAIUS
- (Optical) Navigation
  - SHEFEX
  - ATON
- Payload Software
  - MAIUS

Fig.: TET-1 and BIROS = FireBird

Fig.: Eu:CROPIS

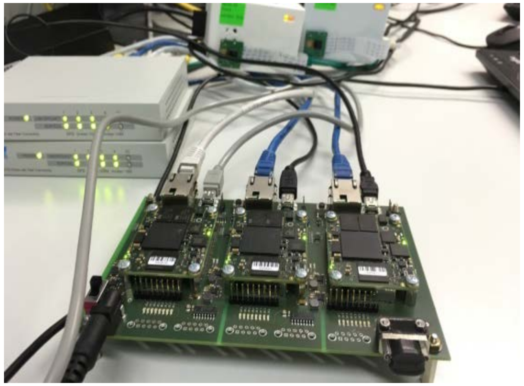
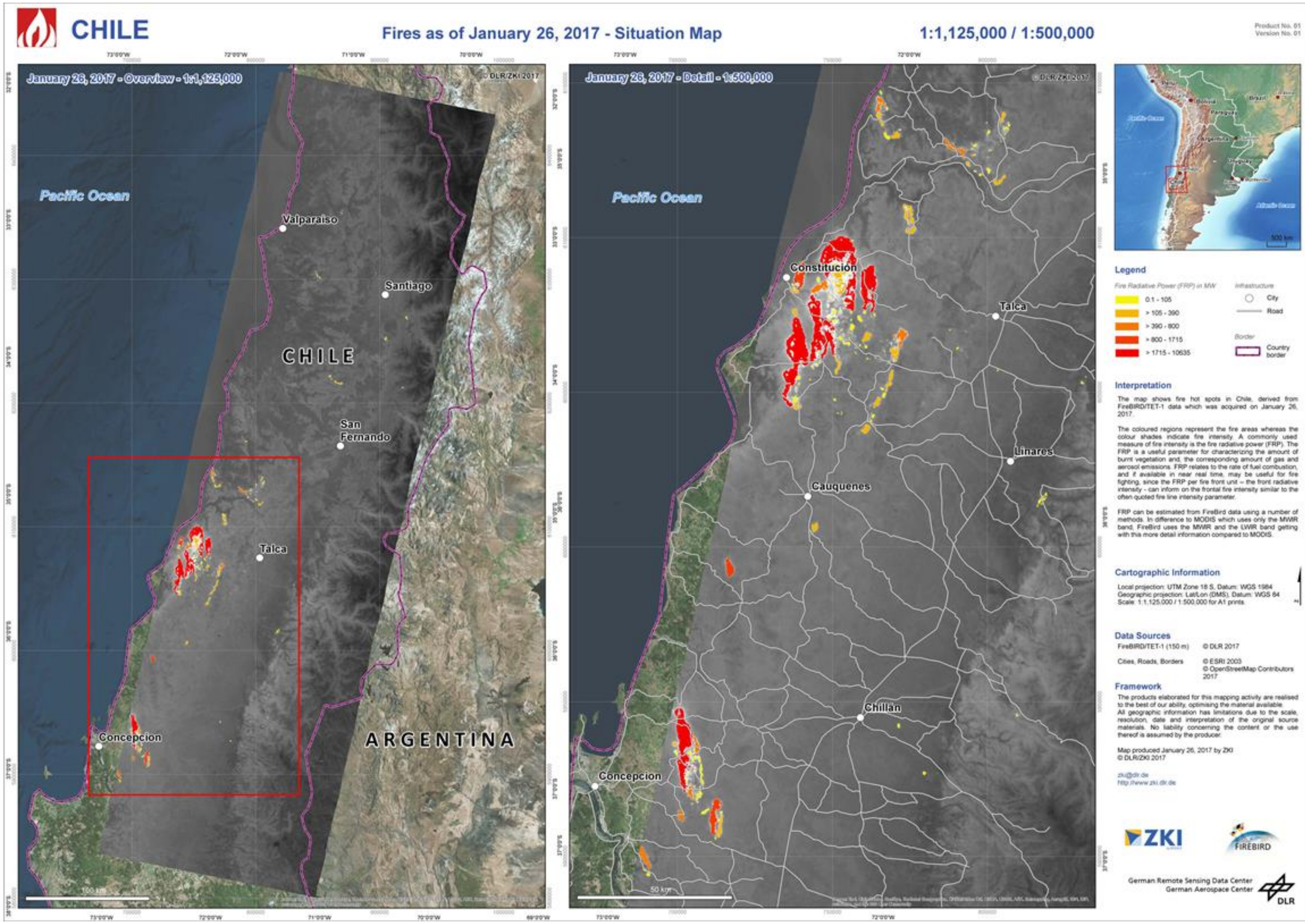


Fig.: OBC-NG

FireBird


CHILE
Fires as of January 26, 2017 - Situation Map
1:1,125,000 / 1:500,000
January 26, 2017 - Overview - 1:1,125,000
January 26, 2017 - Detail - 1:500,000
Pacific Ocean
Valparaiso
Santiago
CHILE
San Fernando
Talca
Concepcion
ARGENTINA
Constitución
Linares
Cauquenes
Chillán
Legend
Interpretation
Cartographic Information
Data Sources
Framework
ZKI
FIREBIRD
German Remote Sensing Data Center
German Aerospace Center
DLR

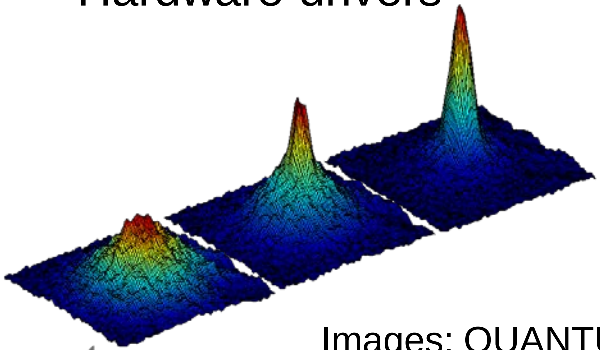
MAIUS / QUANTUS
Payload Software / Experiment Control

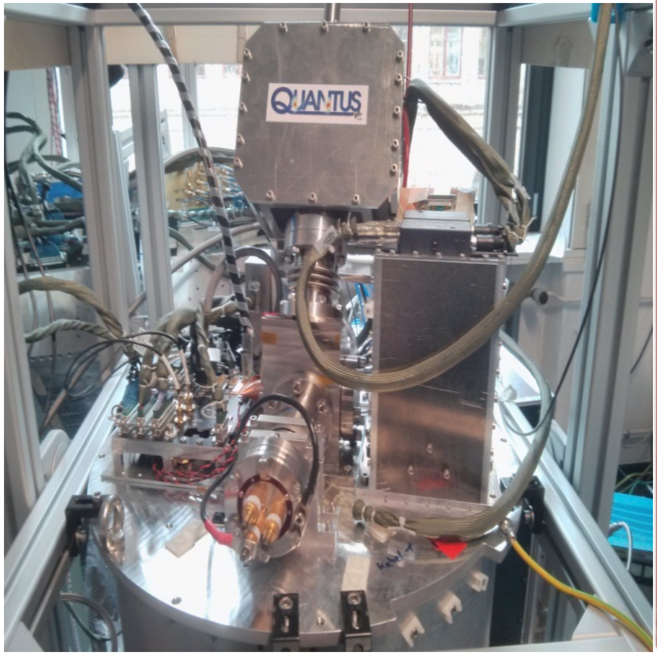
Mission

- Bringing Bose-Einstein-Condensate and atom interferometers to space (Geodesy, Fundamental Physics)

Payload Control and Data Handling

- Software for on-board computer
- Experiment selection
- Tele-Command / Telemetry
- Hardware drivers

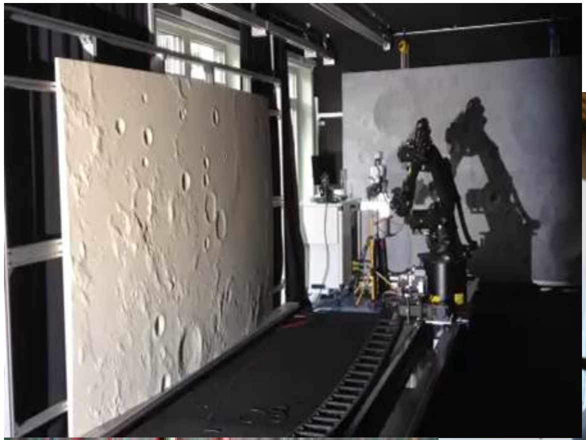


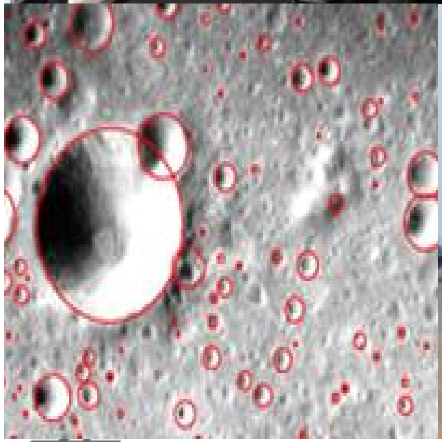


Images: QUANTUS Consortium

Autonomous Terrain-based Optical Navigation (ATON)

- Optical navigation for autonomous landing on celestial bodies





Images: Project ATON

Reconfigurable Distributed Onboard Systems
Motivation

Missing On-board Computing Power

- Number of space-qualified processors and FPGAs is low
- Increasing requirements for more computing power in the areas:
    - Earth observations
    - Robotics
    - …

Redundancy Concepts Often Limited to Subsystems

- Each computing unit has usually its dedicated redundant counterpart
- Standby systems can not take over tasks of computers in other subsystems

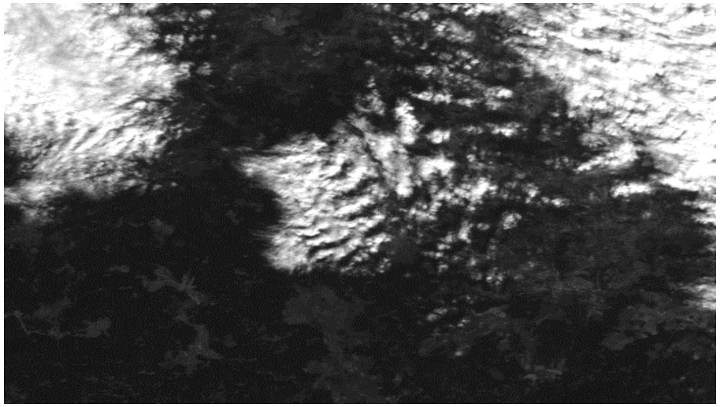


Fig.: Sample image "Cloud Detection"

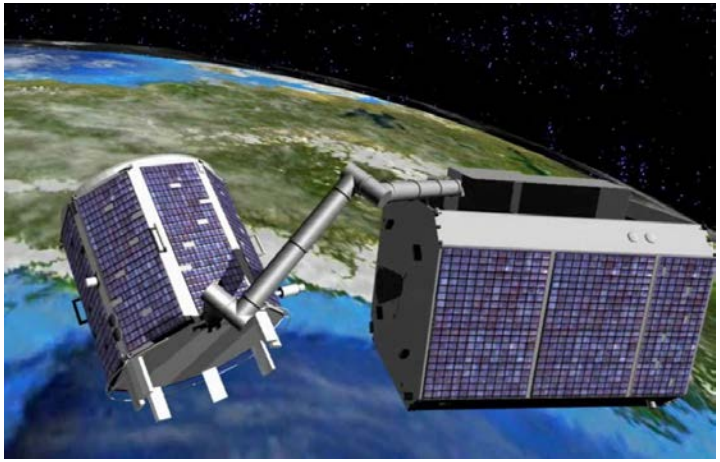


Fig.: DEOS (On-Orbit Servicing)

Reconfigurable Distributed Onboard Systems

Vision

Resource Utilization
Using all available computing resources

Redundancy
Migration of applications across subsystems

Reconfiguration
Software and hardware reconfiguration for different mission phases and error mitigation

Evaluation of COTS (Commercial Off-The-Shelf)

Reconfigurable Distributed Onboard Systems

Scalable On-board Computing for Space Avionics

Fault tolerance

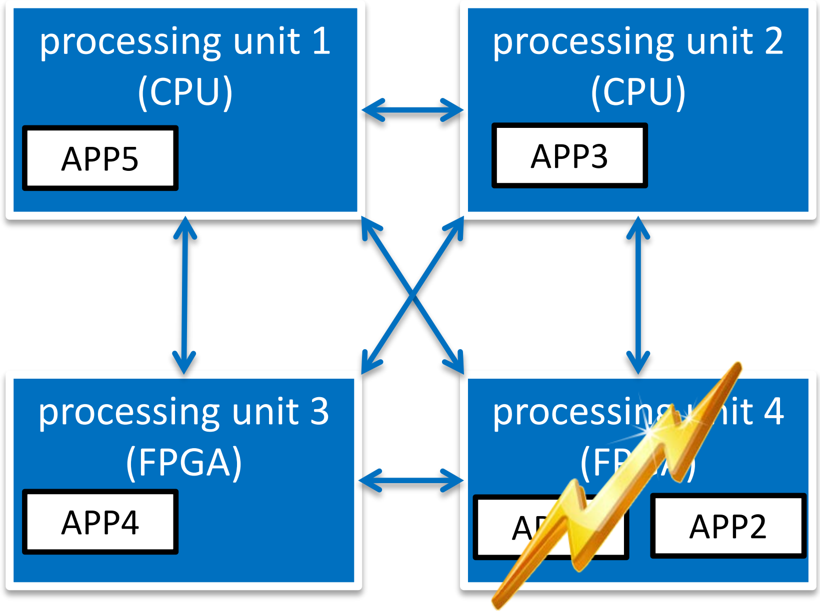
- Reconfiguration due to failure

Scalability

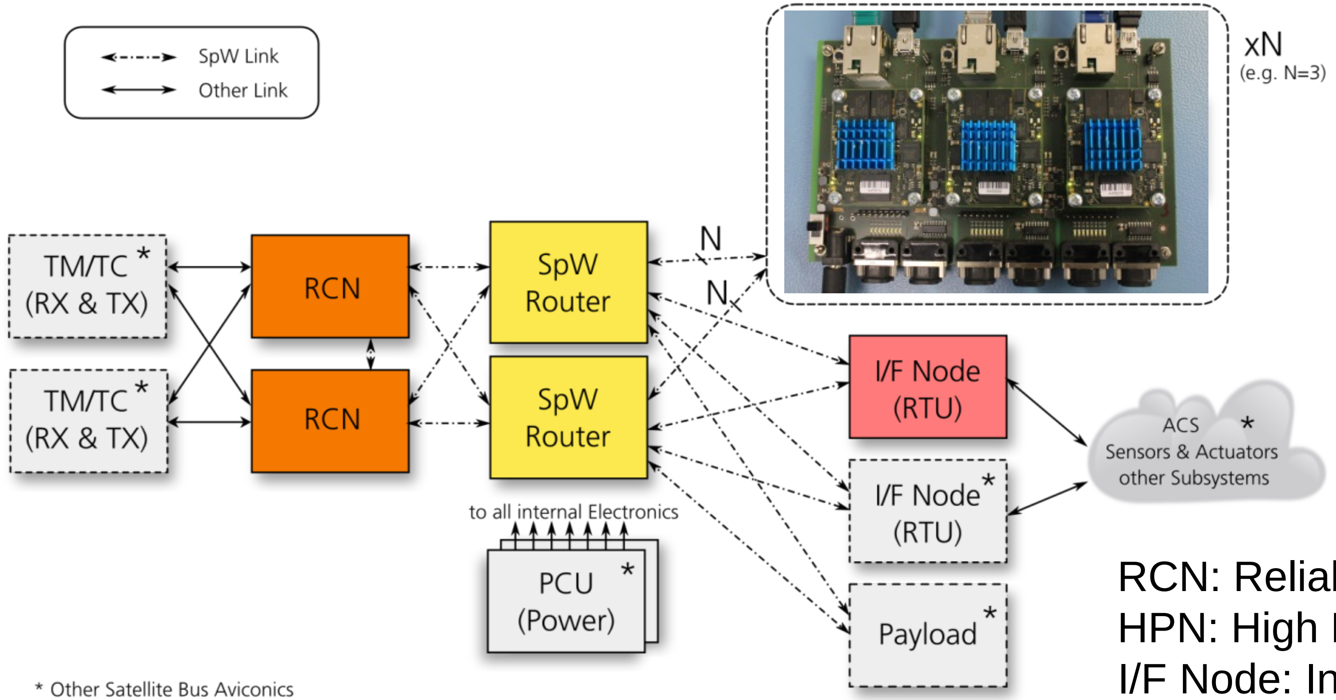
- Heterogeneous Architectures (CPU, FPGA, DSP)
- Combination of COTS and radiation tolerant hardware
- Reconfiguration for new mission phase

Resource Utilization

- Using all available computing resources
- Reintegrate recovered nodes into network



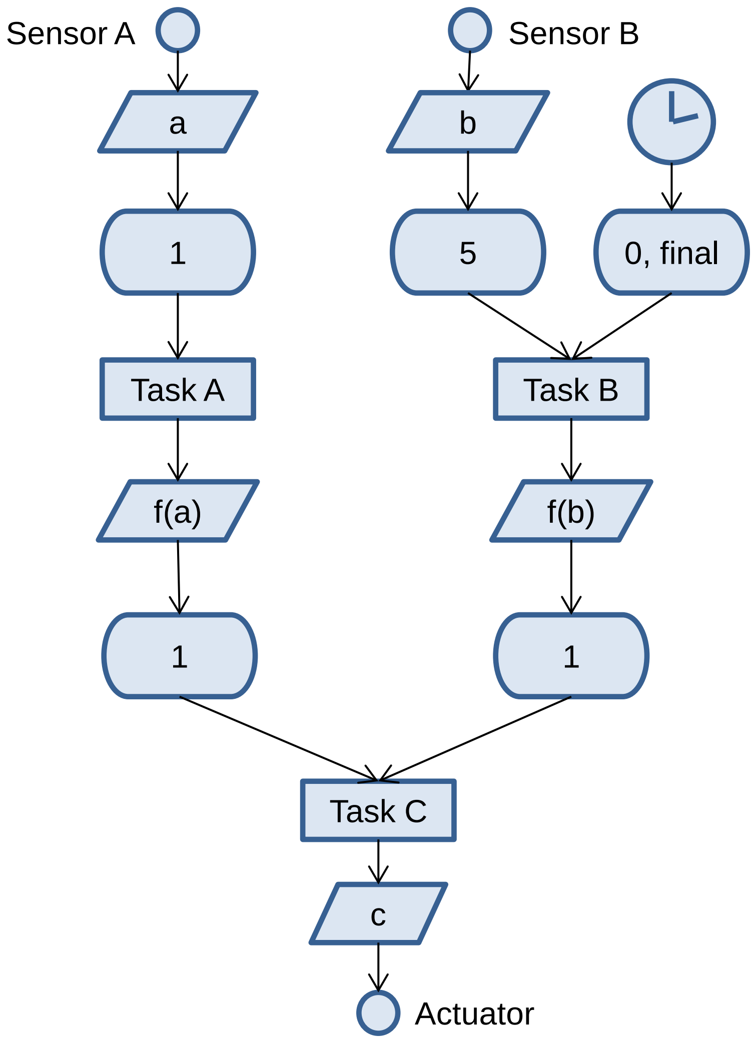
Reconfigurable Distributed Onboard Systems
Scalable On-board Computing for Space Avionics



RCN: Reliable Computer Node
HPN: High Performance Node
I/F Node: Interface Node
SpW: SpaceWire
RTU: Remote Terminal Unit
PCU: Power Converter Unit
TM/TC: Telemetry/Telecommand
ACS: Attitude Control System

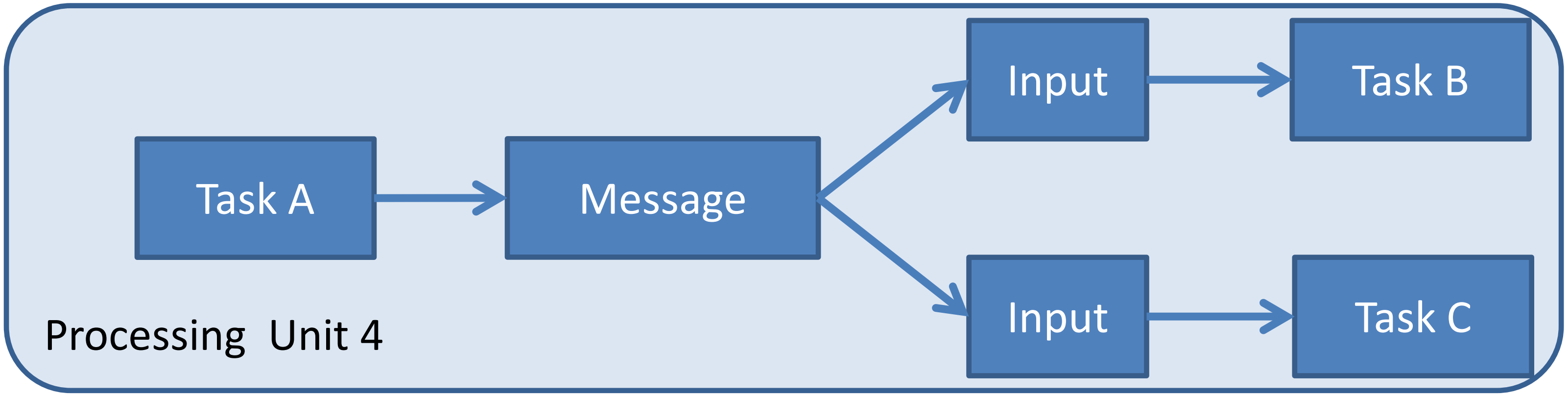
(Real-time) Execution Platforms
Tasking Framework

- Event based execution framework
- Inspired by Petri Nets
- Start of computation tasks by
  - availability of data
  - time events
- Parallelization for
  - multi core
  - distributed systems
- Scheduler available for different platforms
  - Linux
  - RTEMS
  - Xilinx bare metal (ongoing)



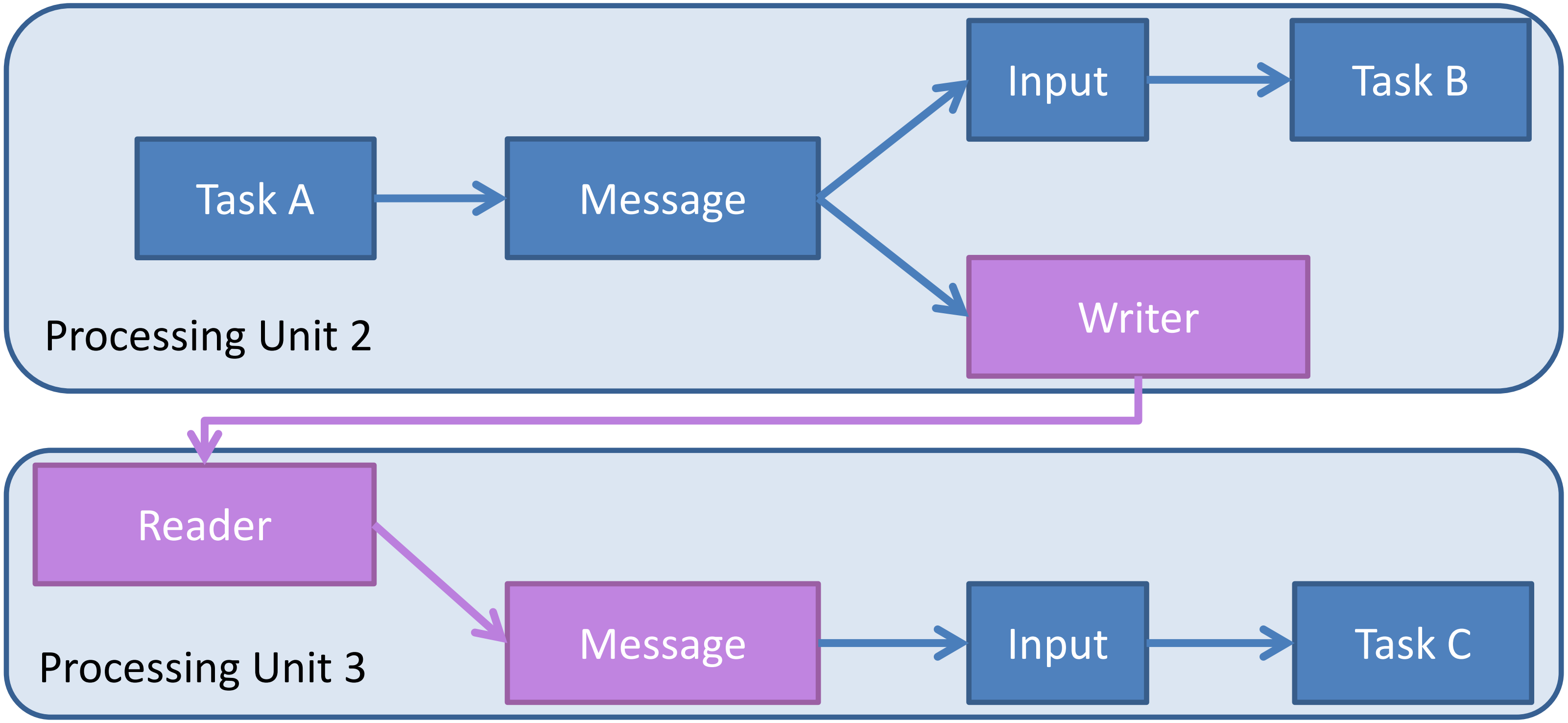
OBC-NG: Task Reconfiguration

- Tasks running on one computing node:



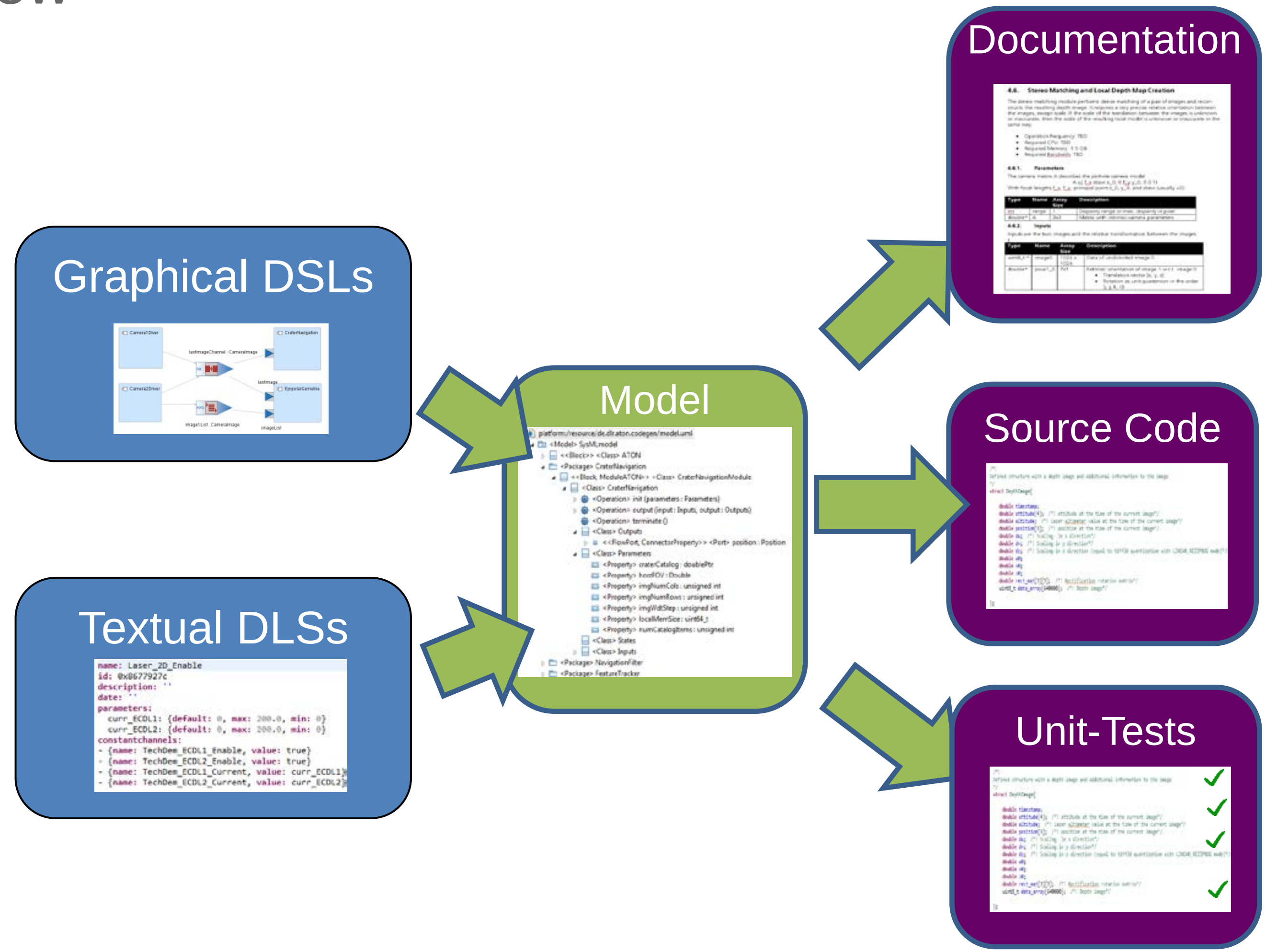
OBC-NG: Task Reconfiguration

- Tasks running on two computing nodes:



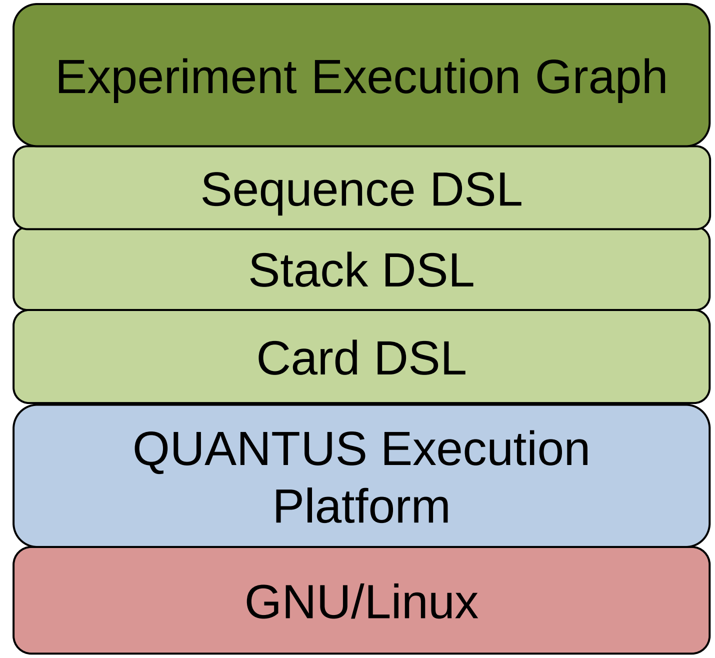
Model-Driven Software Development (MDSD)

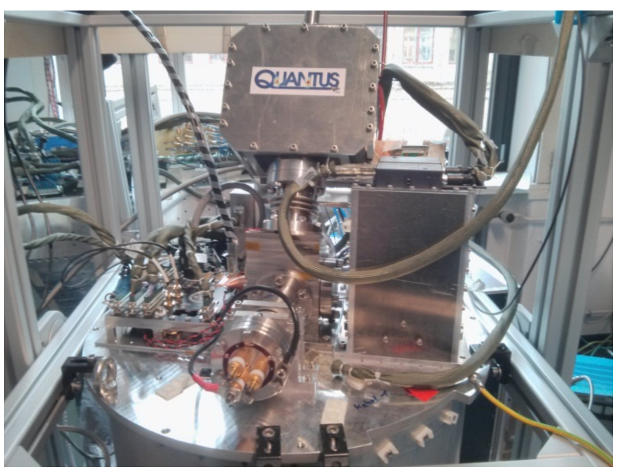
Overview

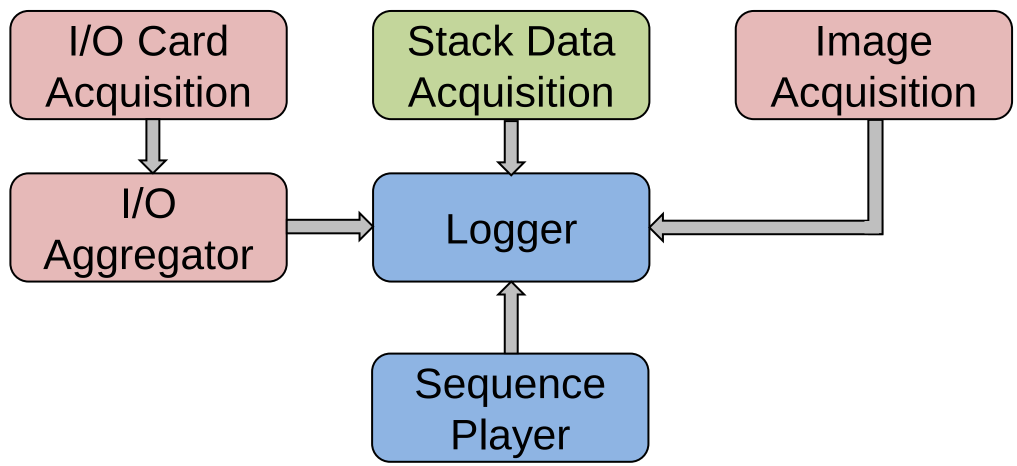


(Real-time) Execution Platforms
Experiment Control Platform

- Software framework to support autonomous execution of complex experiments
- Features:
  - Sequence player
  - Graph-based experiment control
- Use case: quantum optical experiments on a sounding rocket







Model-Driven Software Development (MDSD)
Example: Hardware Driver Description

- Textual DSL to describe behavior of hardware
  - Content presented with textual DSLs can be edited faster
- Textual DSL to describe how different hardware is physically organized
- DSL descriptions are transformed to C++ source code

- Now, hardware developers are able to provide also the software drivers!

LaserCurrentDri | laserStack.stac | laserStackWithC | testSequence.se

```
device LaserCurrentDriver1_0
  parameter currentA float 0.0 to 0.999984741 default = 0.0
  parameter currentB float 0.0 to 0.999984741 default = 0.0
  parameter globalLaserEnable boolean default = false
  parameter laserADisable boolean default = true
  parameter laserBDisable boolean default = true
  parameter powerDownModeA integer 0 to 3 default=0
  parameter powerDownModeB integer 0 to 3 default=0
  parameter signalSourceA integer 0 to 15 default=13
  parameter signalSourceB integer 0 to 15 default=13
  parameter tempSens boolean default=fa
  parameter readA float read only defau
  parameter readB float read only defau
  parameter readTemp float read only de

  process getLaserCurrent
    out readTemp
    out tempSens
    out readA
    out readB
    pattern w 1 1 0 1 0 0 0 1
    pattern w 1 0 0 1 0 0 0 1
    pattern w 1 0 1 1 0 0 0 0
    pattern w 1 0 1 1 0 0 0 1
    repeat from -1 to  -3
      pattern r 1 0 0 1 readTemp[repeti
```

Fig.: DSL for laser driver definition (MAIUS mission)

DLR


Questions?