Applying Autonomy and Robotics to Space Exploration

Thomas Krueger

09/09/2021

European Space Agency

"To provide for and promote, for exclusively peaceful purposes, cooperation among European states in space research and technology and their space applications."

Human Robot Interaction Lab inside the ESA since 2011

"Developing and demonstrating cutting-edge robotics in the field of human-robot interaction with focus on telemanipulation and shared autonomy in the context of ESA exploration programs. Our goal is to boost developments of European partners and reduce overall risk."

- Part of robotics with special focus on teleoperation with time delays up to a few seconds
- Small team (9): hands on & in house
- Demonstrators based on COTS
- Reduce risk, better definition and estimation of work packages

History on usage of DDS and converging to RTI DDS

Use of DDS since 2010 in the Human Robot Interaction Lab

- Originally used for co-simulation of the sampling mechanism of the Exomars
- Evaluation of vendors in that time, requirements: multiple OS, no daemons, ready to use RTI

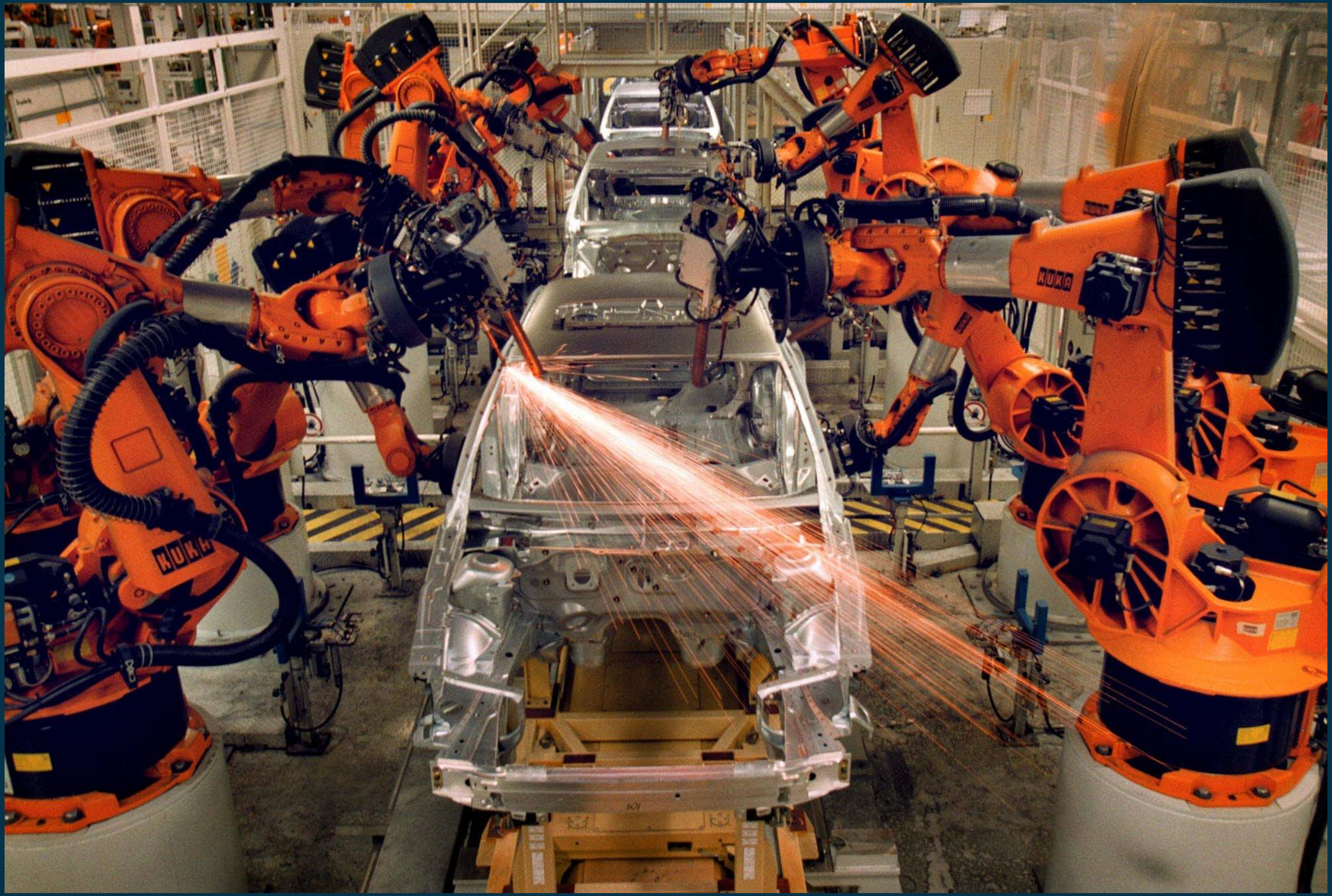
First use in collaboration – control the NASA Robonaut 2 with our haptic devices

- NASA using RTI DDS, with connection to NASA AMES and Mountain View Area

Start-up of the laboratory – key technologies for accelerated development

- Matlab/Simulink with a lab custom developed real-time target
- Use of RTI-DDS with auto generated Simulink blocks enabled to quickly run stable real-time robot control, especially for visiting researches and control experts.

Automation ≠ Autonomy



Definition of autonomy

LEVEL 0

There are no autonomous features.

LEVEL 1

These cars can handle one task at a time, like automatic braking.

LEVEL 2

These cars would have at least two automated functions.

LEVEL 3

These cars handle "dynamic driving tasks" but might still need intervention.

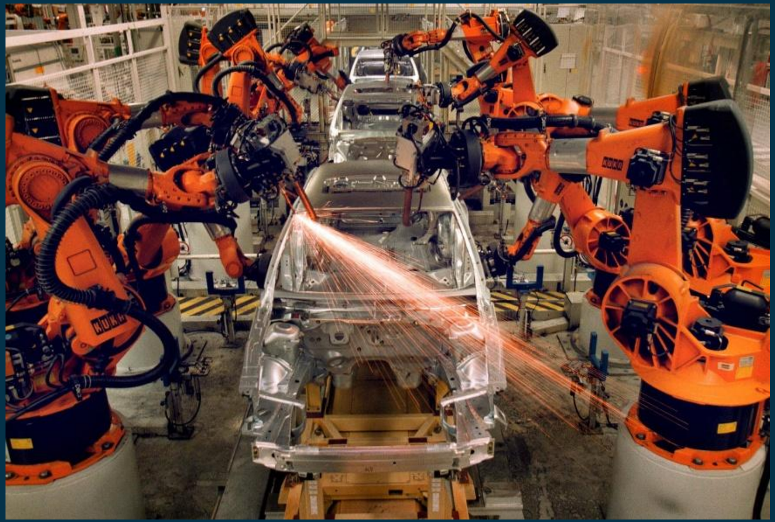
LEVEL 4

These cars are officially driverless in certain environments.

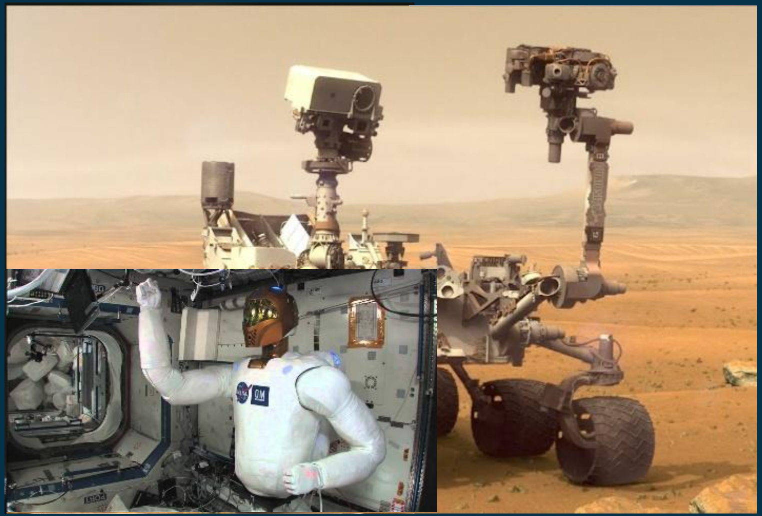
LEVEL 5

These cars can operate entirely on their own without any driver presence.

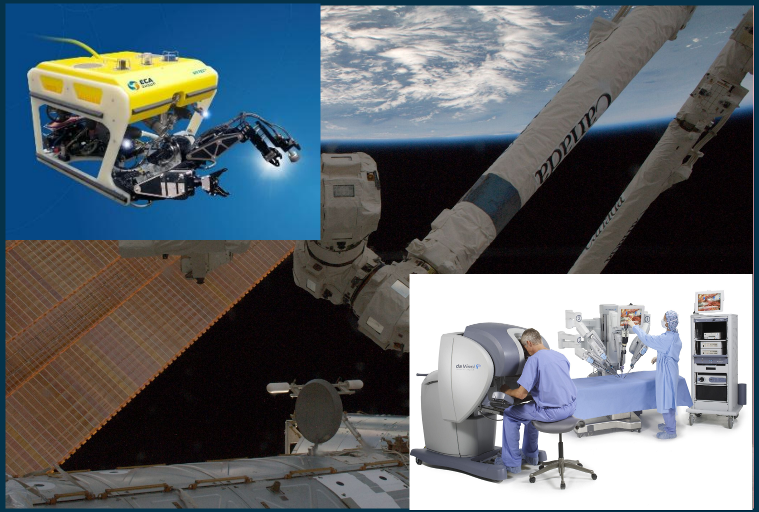
The diverse field of robotic



Industrial automation



(semi) Autonomous agents

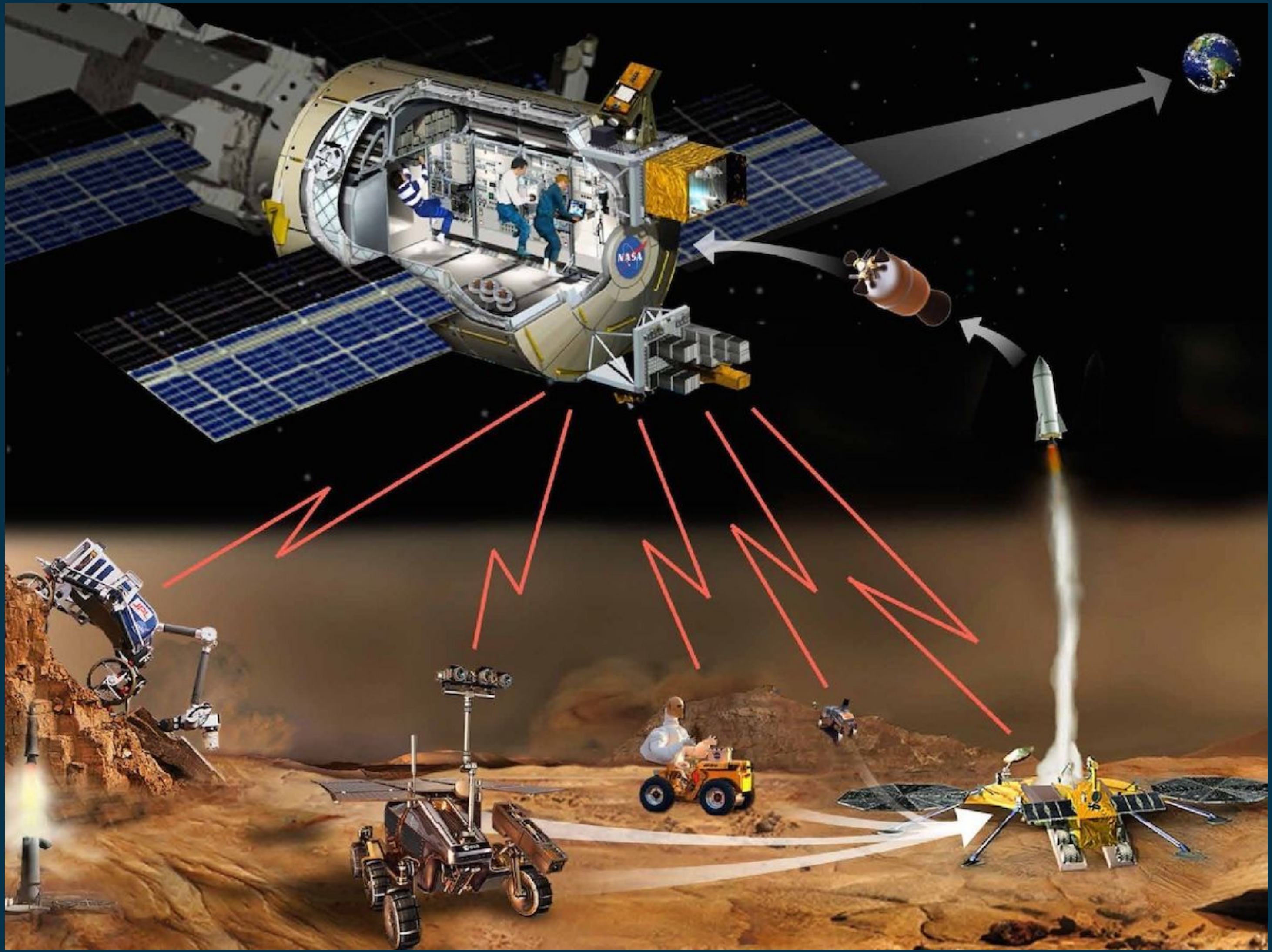


Teleoperation

Commanded

Guided

A case for teleoperation – NASA and ESA METERON (2010)



(Image credit: NASA/GSFC)

Distant, Dirty, Dull, Dangerous

Teleoperation combines the human perception with the advantages of robots to get the job done

A realistic perspective

"A human geologist can do in a week what the Mars rovers can do in a year." [Frances Westall 2018]

User perspectives

After ANALOG-1 an experiment to use direct teleoperation to collect rock samples, L. Parmitano said that it would be nice to initiate certain task and let the robot notify me when done. [2019]

After SUPVIS-Justin an experiment to autonomy to perform maintenance - A. Gerst said that it would be nice to directly jump in and command the robot directly. [2017]

Scenario for lunar exploration

A lunar exploration scenario □ creating an Analog with the ISS

The robot

dexterous view & cameras

communication

locomotion

manipulation arm + gripper

power
control algorithms
localization
safety
diagnose
logging
…

Usually it is all about the robots …

… the Astronaut, the science, safety, control algorithms etc.

Communication links are overlooked in robotics - *"we will figure it out when we get there"*

- Uncontrollable & uncomfortable element tendency to build own infrastructure
- Real world conditions are limiting or provide opportunity to be solved

Addressing concerns after the research phase

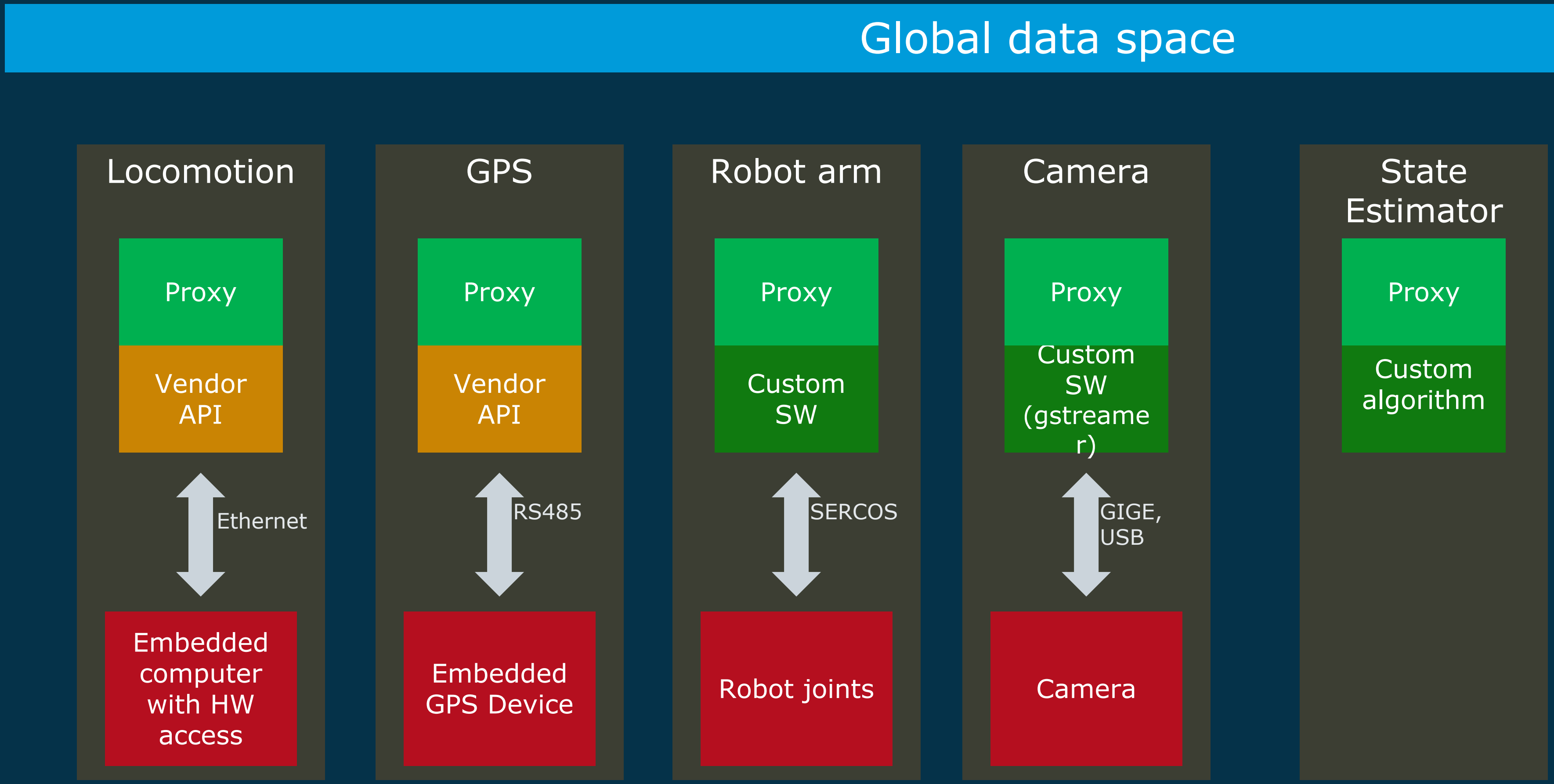
- Maintainability
- Scalability

§ Disclaimer §
Guilty roboticist

Teleoperation and distributed system architecture as driving factor for our design choices

Concept of proxies (robotic microservices)

- First step is to create a “proxy” with a DDS interface for each device



Contenders

- Industrial PLC systems
- Dedicated Embedded Real-Time systems
- Standard Linux Distribution(s)

Finding the sweet spot for:

- Interfacing with hardware
- Real-time capabilities
- Multi purpose, libraries and chance to pass qualification tests
- Manageable and understandable by a small team (configuration, update, toolchains, SDK, CI)

Buildroot with only required packages and custom configuration

Logical communication architecture of the control station

DDS - global data space

Station Control

- DDS IF
- Simulink auto - gen code
- ↕ USB
- Haptic Device, Joystick

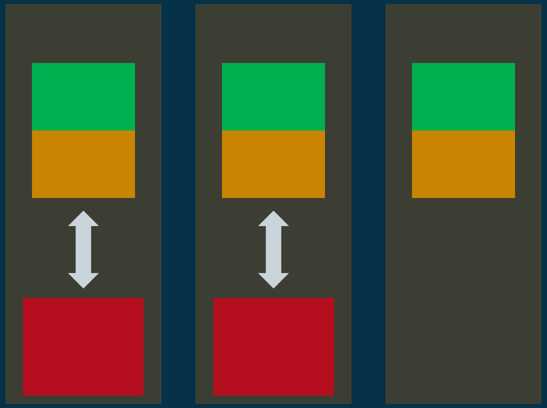
Operator UI

- DDS IF
- Qt applicatio n

Ready to go – forgetting about the communication link

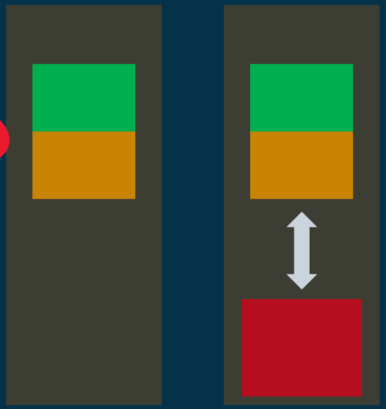
Everything is in the DDS global data space – we can be agnostic of the link in between

global data space (robot)



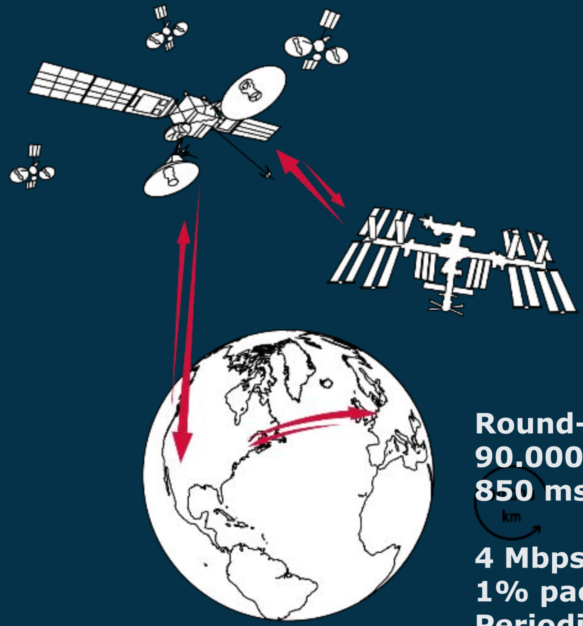
global data space (control station)

Can we?



WiFi in the lab
Something in the other building, city, country, …

Something on the Space Station

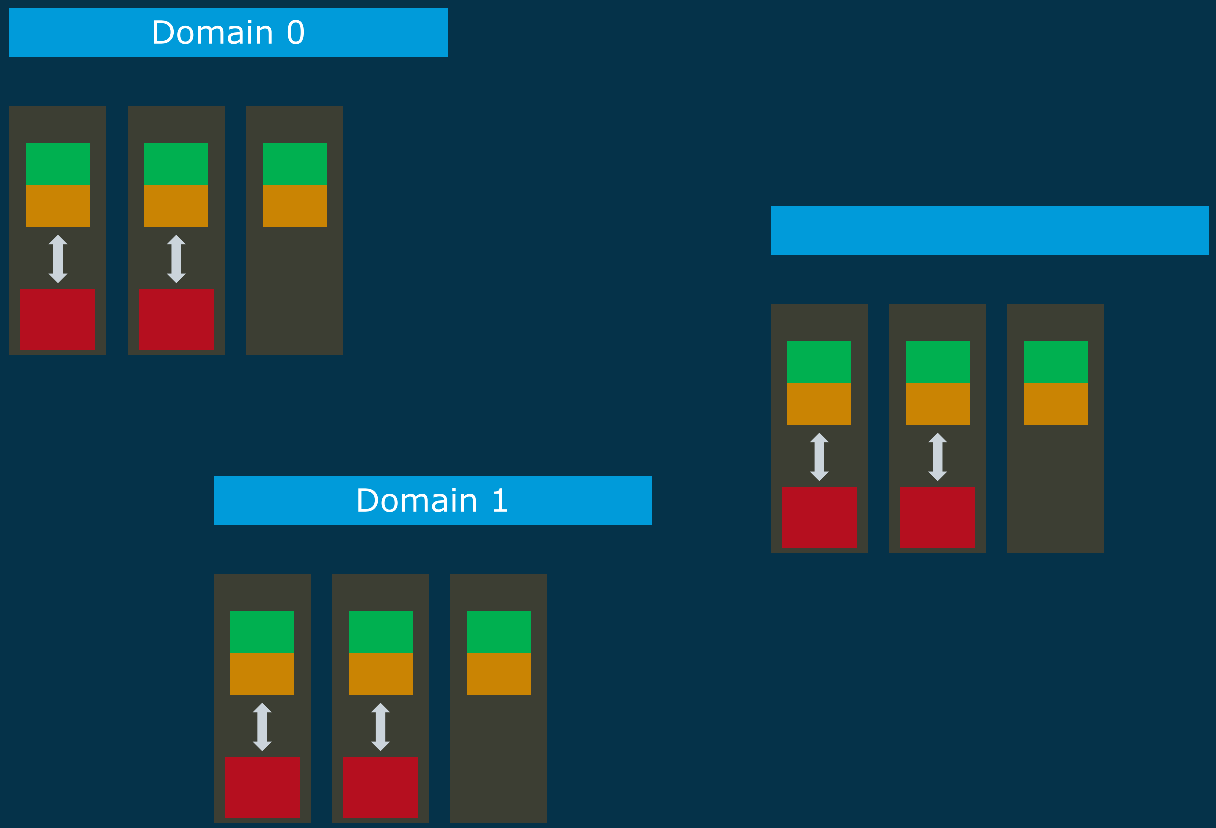


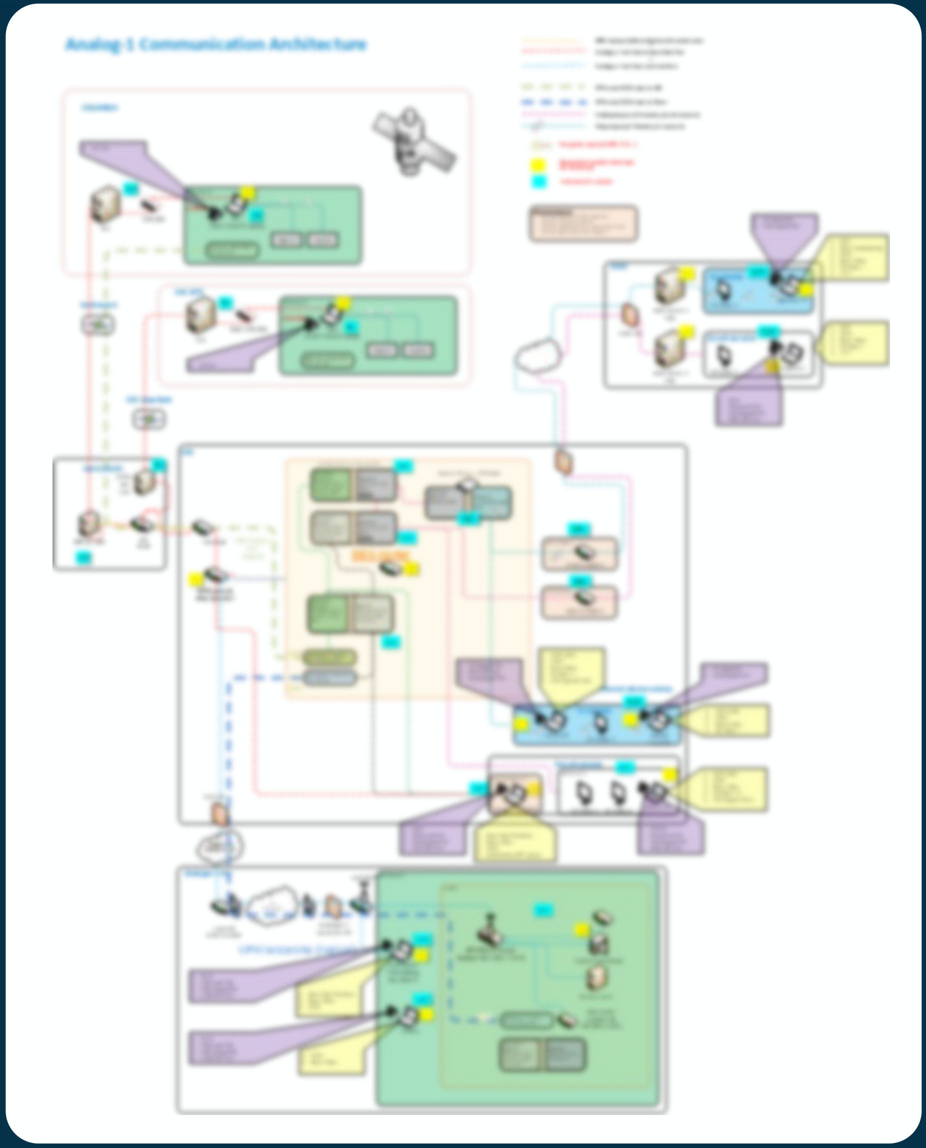
Round-trip:
90.000 km
850 ms

4 Mbps
1% packet loss
Periodic handovers

Challenge: Balancing the needs of various stakeholders

Creating an architecture in an very conservative and safety/security driven environment where serverless and publish/subscribe technologies are not common and satisfying the requirements of various stakeholders





From simple to complicated

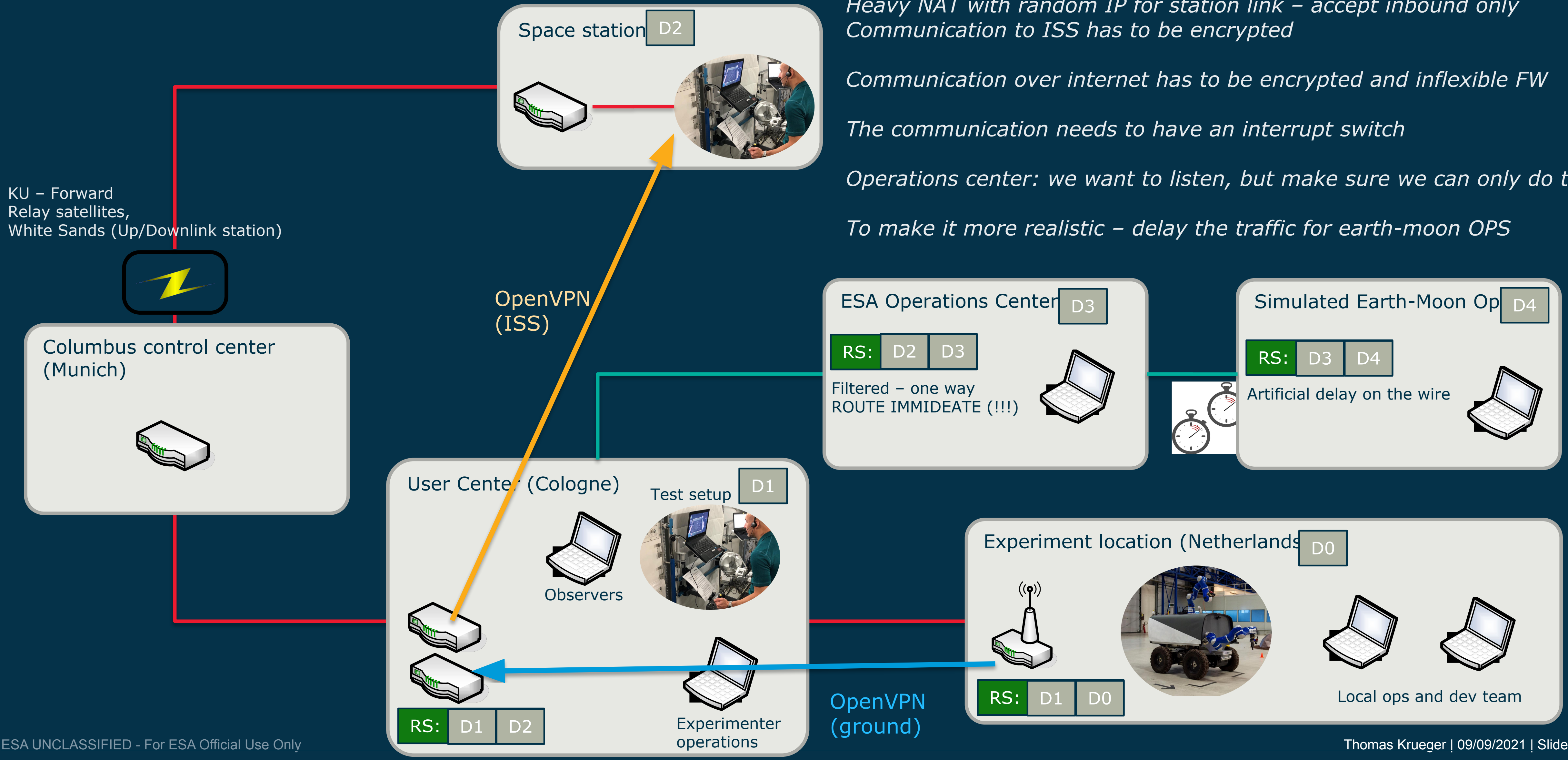
Heavy NAT with random IP for station link – accept inbound only
Communication to ISS has to be encrypted

Communication over internet has to be encrypted and inflexible FW

The communication needs to have an interrupt switch

Operations center: we want to listen, but make sure we can only do that

To make it more realistic – delay the traffic for earth-moon OPS



The good – almost exclusive usage of DDS for data transfer

- Telemetry, commands, Real-time data for haptic
- Video over DDS (H264 encoded)
- Use of RTI Recording service to log all data (consistent logging)
- Use of RTI Routing Service
- Tooling applies at all stages (admin console, spy)
- Architecture and system design based on DDS

"Almost"

- OpenVPN usage to cross "difficult network segments" (good performance, but …)
- File transfers via FTP

The Good

Routing services allow network separation/simplification possible to handle with a small team

- Most machines with generic configuration:
    - Very simple network config (subnet, one standard gateway)
    - NO discovery peers
    - NO advanced networking

- Few powerful machines run routing services
    - Avoid complex IP based routing (complex IP behavior not clear to everyone)
    - Configure routing services only with peers and only here advanced routing and FW
    - Easy storage of those configurations

Design drawbacks

- Despite good performance VPNs reduce the MTU, thus more overhead/issues, limiting ourselves
- Configuration effort and system administration (2 VPNs, minimum 2 RS)
- Multiple routing services with route on "IMMEDIATE" cause dangerous locks in case of type changes
    - □ Routing services allow since 6.0.1 more than 2 participants!
    - □ Post experiment evaluation session with RTI-support
- The justification for VPN as means for encryption (□ Connext DDS Secure)
- Performance issues with *iptables* and *OpenVPN*

Lessons learned

- Trigger discussion with support and system architects at early stage
- Remove the VPN where possible
- More effort to advertise RTI DDS as TRL9

Relying on the quality of RTI Connext

Pushing functionality into the middleware where possible

- Filters (content, flow)
- Ownership control (override autonomy with manual control)
- Automatic failover

Development tools

- Admin console: visualize data, visualize the distributed, system, Check matching
- Recording service: log data and replay (automatic consistency)

A wish working with embedded headless systems – tools with local web servers and web interface to "see the local systems perspective"

EAC Control Room

The experiment went flawless,

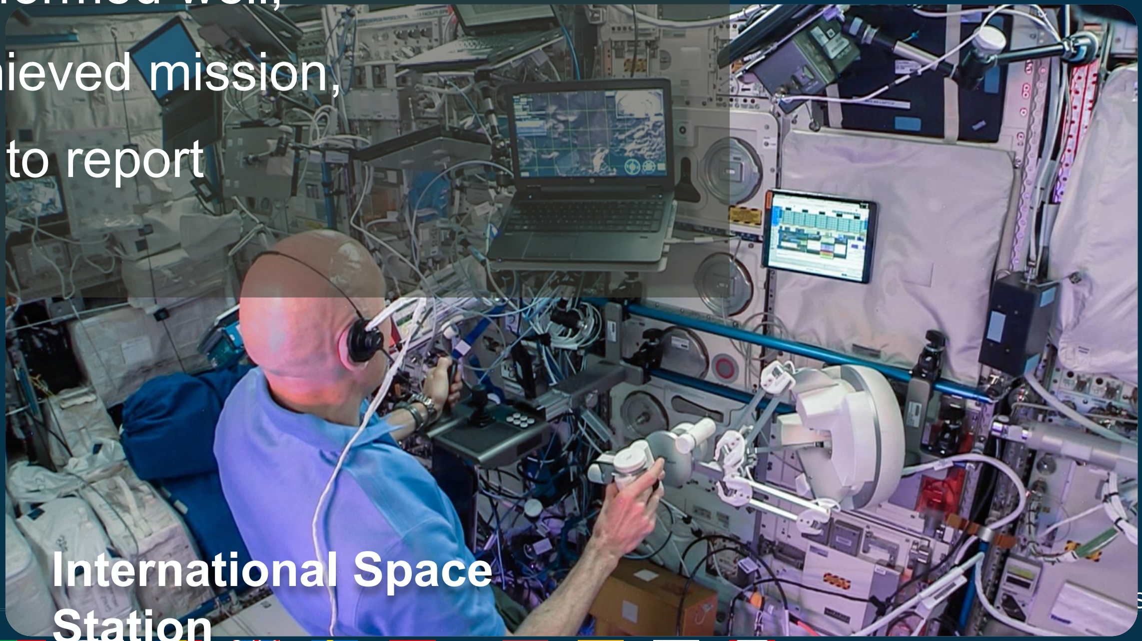
no re-connection problems after satellite handovers,

Robots performed well,

Astronaut achieved mission,

nothing to report

EAC Science Room



International Space Station

Outlook after successful Run

Transition to more autonomy

Teleoperation serves “by accident” as ideal base to advance to autonomy and is fallback

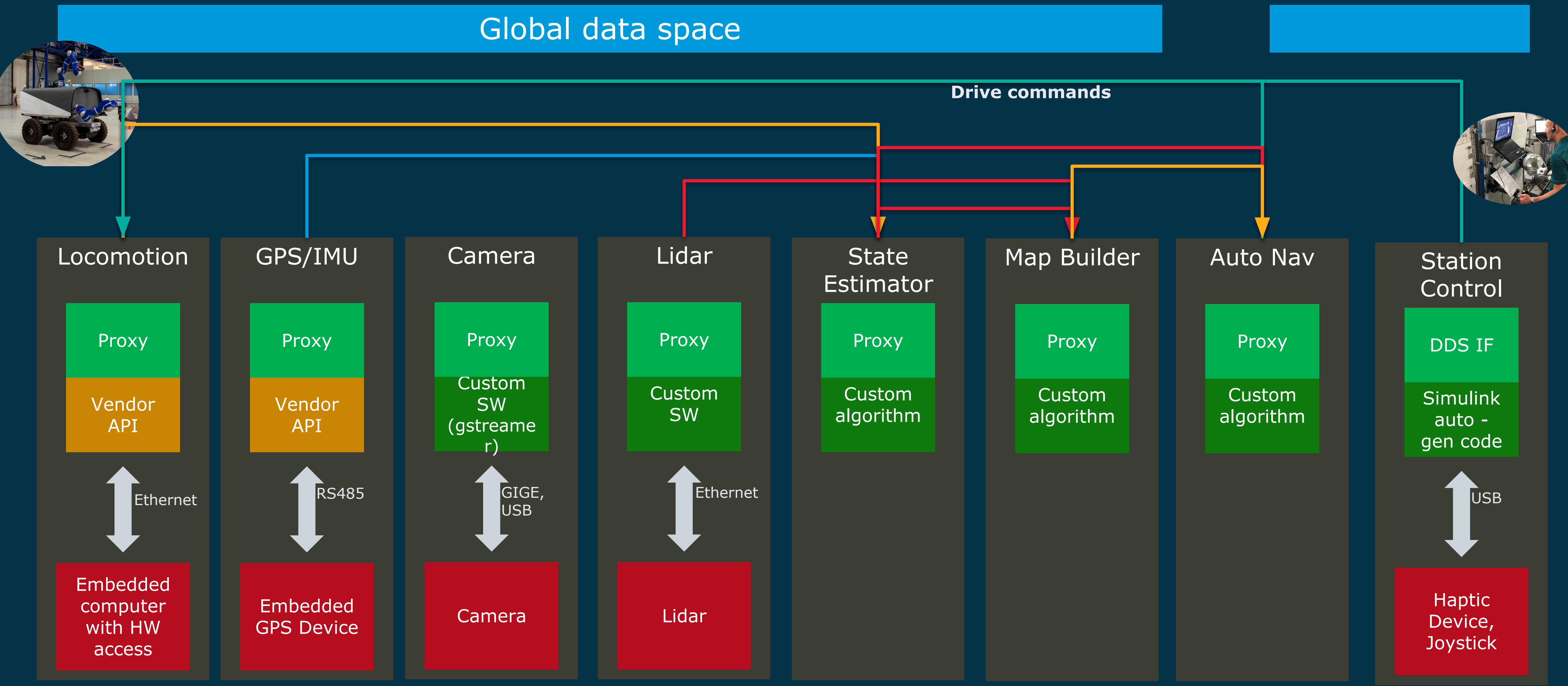
Collaborations to improve single components

“Microservice architecture” allows third party development on the live systems

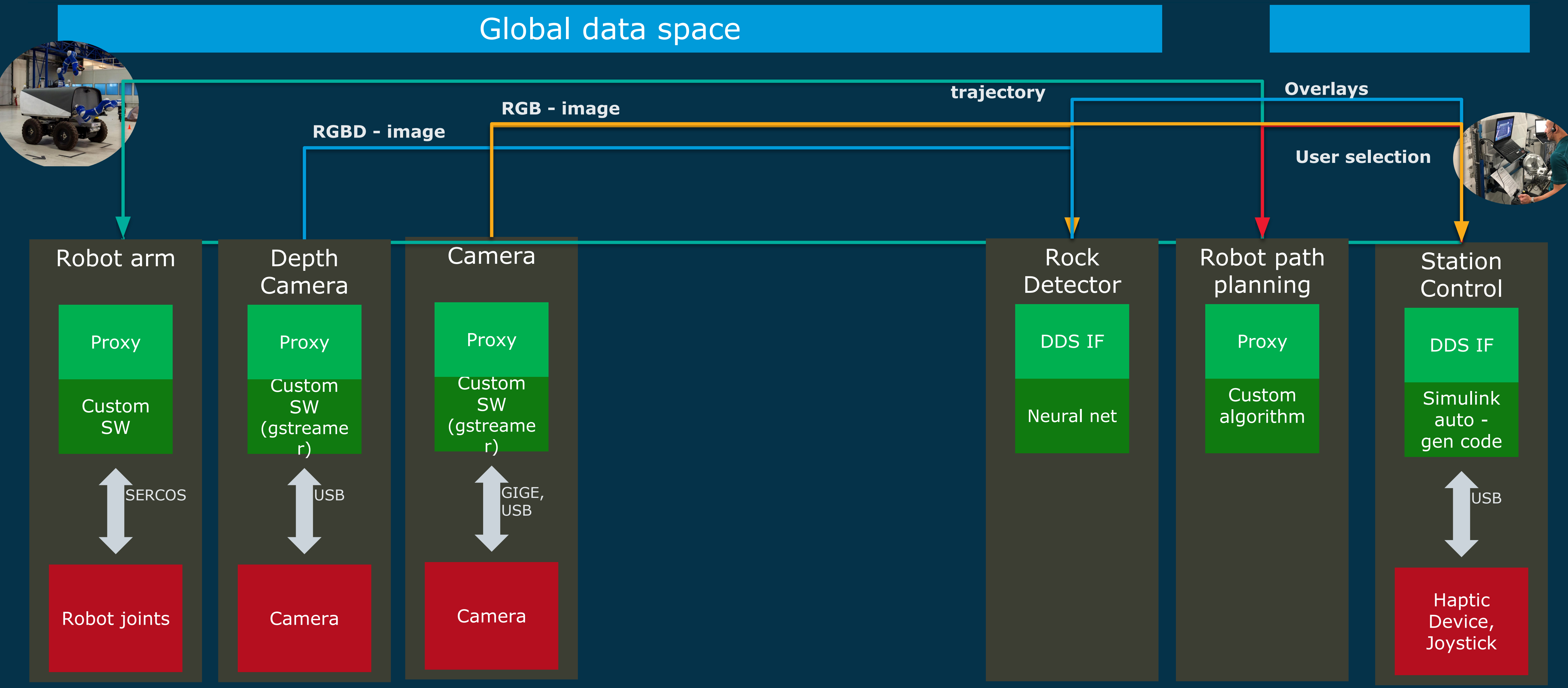
Controlling a fleet of heterogeneous robots

Future project challenge to design data structures and principles without writing bridges or adapt each robot individually

Transition to autonomy – DDS robotic microservices as “fly-by-wire”

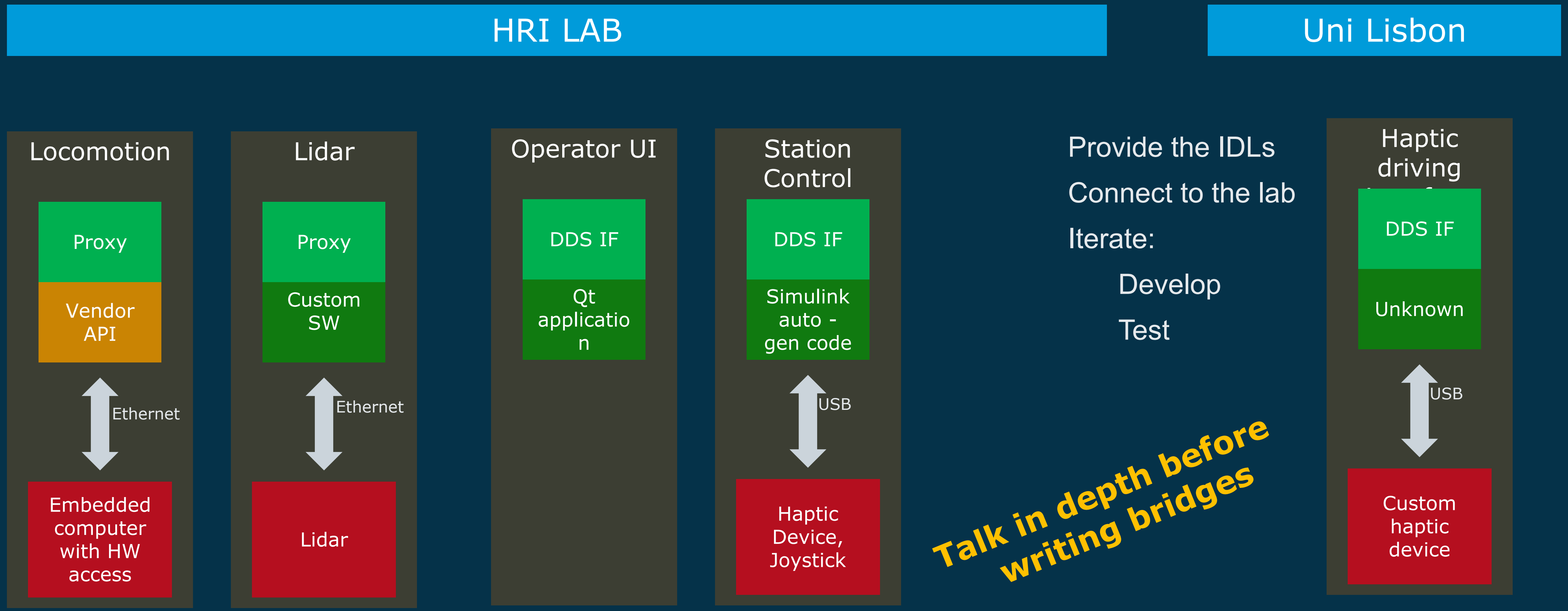

Global data space
Drive commands
Locomotion
Proxy
Vendor API
Ethernet
Embedded computer with HW access
GPS/IMU
Proxy
Vendor API
RS485
Embedded GPS Device
Camera
Proxy
Custom SW (gstreamer)
GIGE, USB
Camera
Lidar
Proxy
Custom SW
Ethernet
Lidar
State Estimator
Proxy
Custom algorithm
Map Builder
Proxy
Custom algorithm
Auto Nav
Proxy
Custom algorithm
Station Control
DDS IF
Simulink auto - gen code
USB
Haptic Device, Joystick

Teleoperation as stepping stone to autonomy – rock sampling

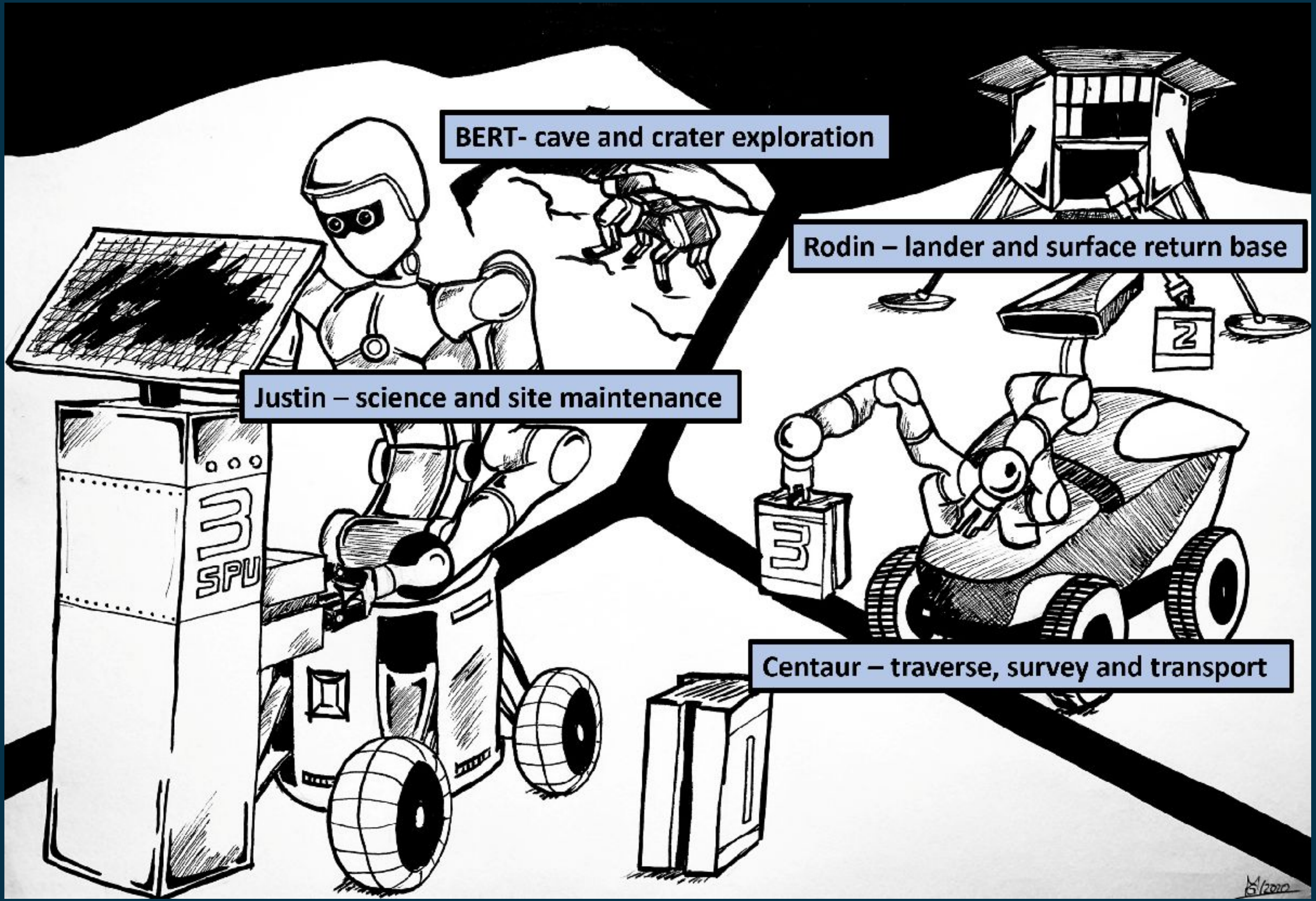


Collaboration – leverage of the system design for collaboration

Ease the connection to collaborators with RTI Cloud Discovery Service to reduce disadvantages of VPN, time lost in firewall discussions, requests and debug



Controlling a fleet of heterogeneous robots



- Multi-site operation
- Parallel and sequential collaboration
- Supervisory command of a robot team
- Teleoperate a robot within a robot team

Confident control a fleet of very similar robots with Connext DDS Keyed topics, QoS, checks, routing and logging …

General challenge to use abstract task and object descriptions and how provide them to the global data space

A word on ROS

Deliberate decision NOT to use ROS history and rationale

- Strong focus and real-time teleoperation and haptics since 2010
- Requirement to have real-time (Xenomai/Preempt RT) - haptics
- Low overhead best effort communication – robot and control are connected

Systems need to pass security checks:

- *“Please remove all unused packages from the system and explain the rest” […] ESA Security officer*
- Anticipation of future safety checks

Acknowledgement that robotics sits in a tricky spot

- High performance safe industrial systems vs. very complex algorithms with intense library usage

We are looking forward to

Rapidly improving infrastructures for robotics on ground (inspection, delivery, …)

- 4G and 5G cellular networks
- Satellite internet constellation in LEO (low latency)
- Systems to automatically swap communication channels (RTI Cloud Discovery)
  e.g. in urban areas from City wifi to 4G to whatever is currently best

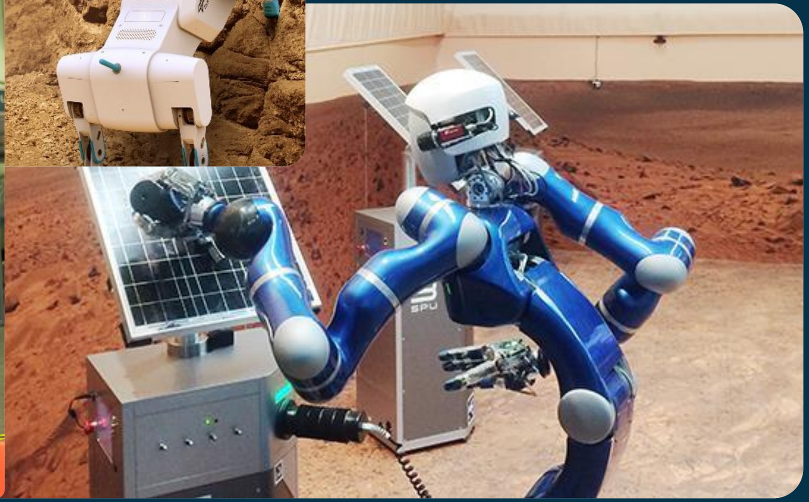
ARM powered devices with GPU cores for video decoding and machine learning

Exiting technologies *- with wish for Connext DDS support ;-)*

- Qt with QML / Flutter
- Golang, Rust
- Simulink blockset roll back or re-invention

Next experiments

- On Mt Etna – ARCHES June/July 2022
- Controlling a fleet of heterogonous robots from Space – MARC-II
- More collaborations



Thank you

&

Questions?

contact: Thomas.Krueger@esa.int