From Gamers to Tango Dancers

Bridging Games Engines and Distributed Control System Frameworks for Virtual Reality (VR) based scientific simulations

FOSDEM16, Brussels, 31 January 2016

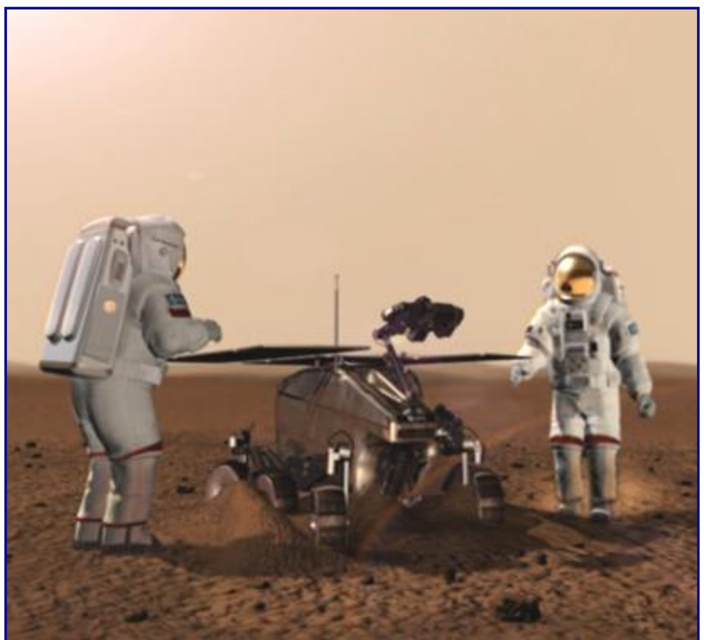
Acknowledgements

Thanks to:

- The Italian Mars Society (**IMS**) under whose affiliation great part of the work was performed
- The Austrian Space Forum (**OeWF**) for organizing the analog missions where field testing was carried on
- **Google** and **ESA** for sponsoring some of the students who contributed to the project

Mission critical Command, Control and Communication (C3) systems

- Crewed missions, such as planetary exploration, or urban search-and-rescue missions are characterized as complex and highly-demanding.



Mission critical Command, Control and Communication (C3) systems

- The overall HW/SW system supporting such missions, referred as Command, Control and Communication (C3), must be able to integrate many modules, including sensors, graphical interfaces and navigation software.

The C3 System

The C3 system must provide the data processing and communications functionalities required to:

- monitor and control the crew headquarters
- monitor and maintain crew health and safety
- communicate with mission support, robots and crewmembers
- support data processing related to the mission objectives
- host the core part of the crew operations planning and scheduling support system

C3 as a Distributed Control System

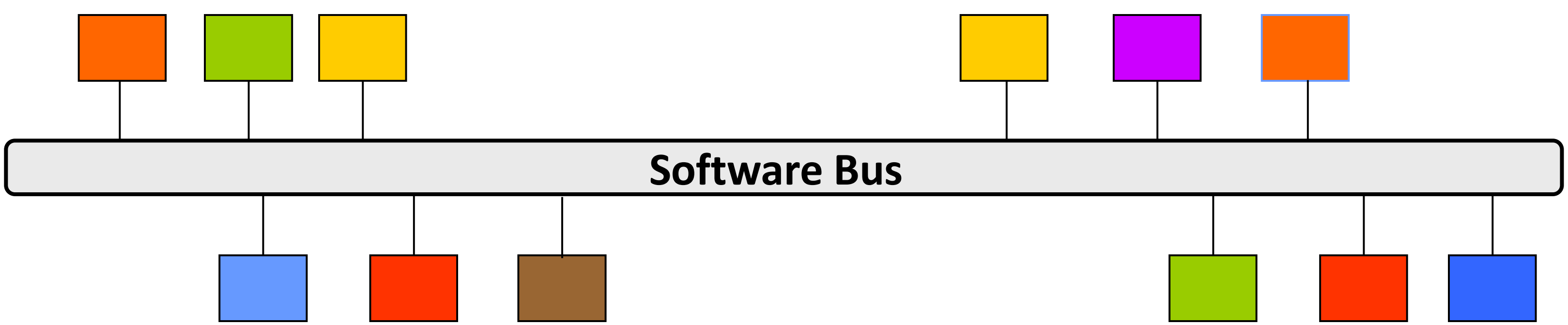
- For C3 a common architecture and standardization will be a key elements.
- As part of this systematic standardization effort the C3 system has been classified within the heterogeneous Distributed Control System (DCS) domain. DCS fundamental services are:
    - Communication
    - Online database
    - Configuration database
    - Logging
    - Recording & Replay

Control Frameworks

- The common services and tools needed in all DCS are grouped into a Framework.
- The specific applications are then built on top of it by using and/or configuring its available tools.
- By using Frameworks, time can be spent on the project-specific solutions (i.e. the functional part) rather than on fundamental implementations.

Communication Service

- The communication service constitutes the foundations of a Framework and can be identified as a specific layer of software (middleware) between clients and control server processes.
- The Communication Middleware implements what can be referred to as a “Framework Software Bus”.
- All C3 applications use the Software Bus to communicate with each other and are oblivious to the underlying communications mechanism.



Control Framework Selection

- Framework selection criteria applied included:
  - Required services
  - Performance needs
  - Desired operating systems
  - Real-time requirement
  - Provided hardware integration support
  - Software engineering tools & development support.
- Various options have been considered and the TANGO DCS Framework (http://www.tango-controls.org) was selected

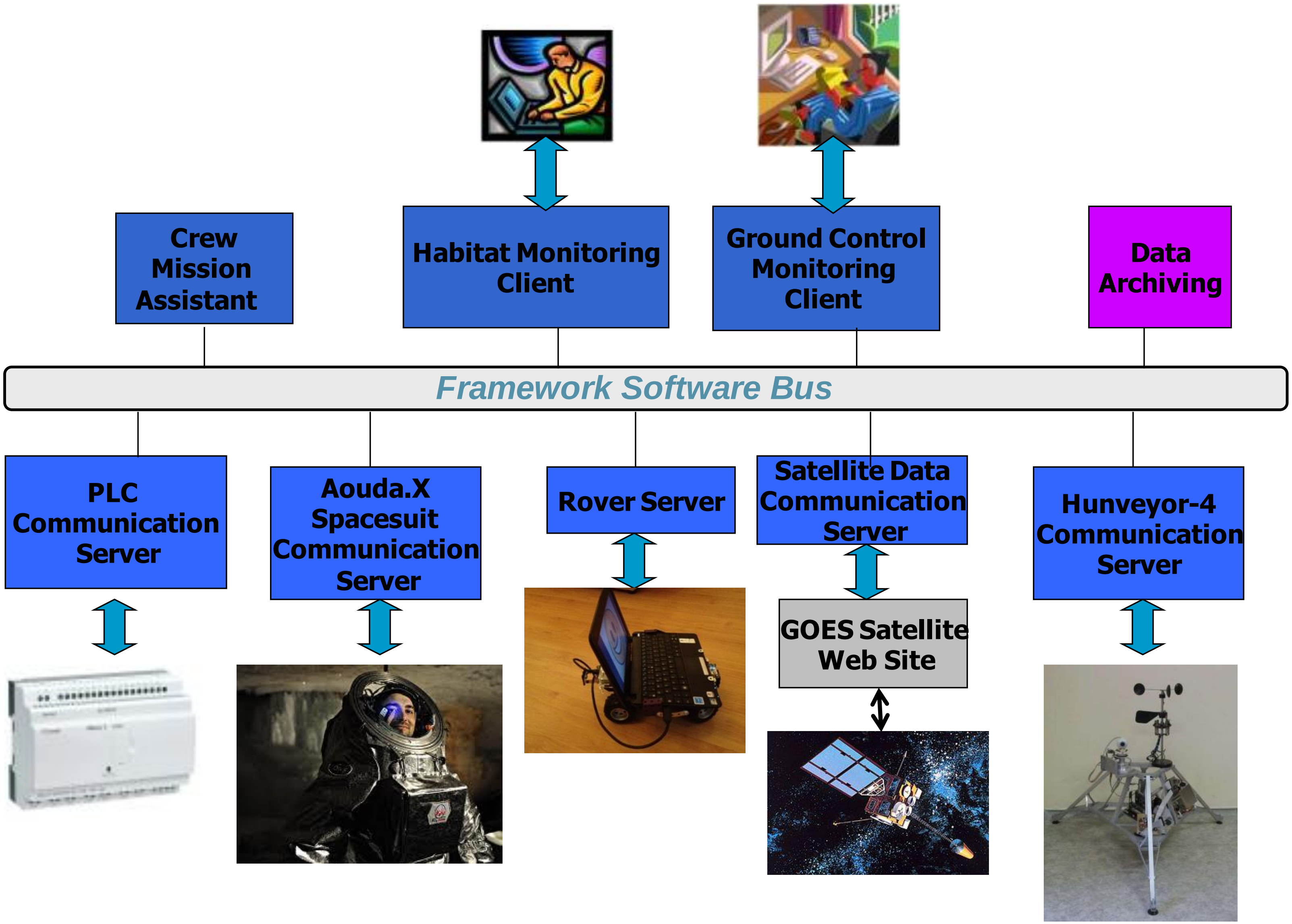
Connecting things together

Control Frameworks Evaluation Metrics

Metric / Tango evaluation

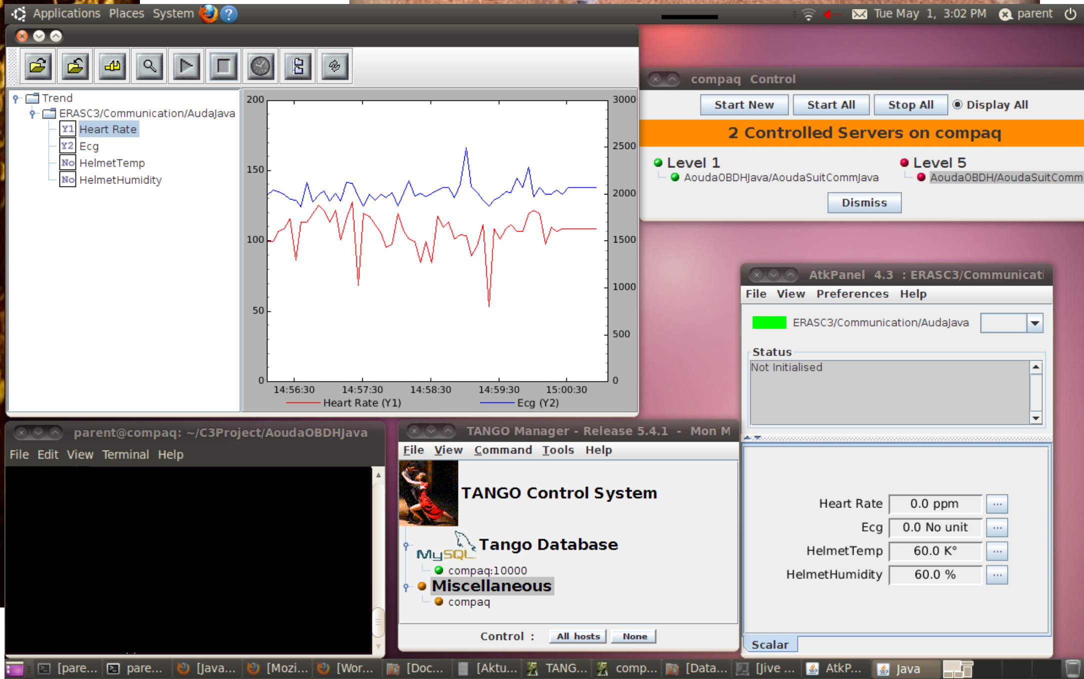
- Required services /(all standard DCS services available)
- Performance needs / (no performance problems experienced so far)
- Desired operating systems /(very easy apt-get based installation on Ubuntu, well supported on Windows)
- Software engineering tools & development support / (very useful tools as pogo and server/client templates)
- Learning Curve / (very good encapsulation of Corba and ZeroMQ complexity)
- Scalability (Number of Servers) / (good scalability, very easy redeployment)

A C3 Prototype Functional Diagram



C3 Prototype field tests

- The C3 prototype has been put in operation during various OeWF analogue field missions.
- The standard plotting facilities embedded in Tango has been used for the visualization of the data.



Virtual Reality (VR) Simulations

Development and Testing of C3 systems is a major challenge. VR based Simulators can play an important role in this respect. In VR, a model of any future mission in an extreme simulated environment can be created.

Immersive Virtual Reality (IVR) Station Development

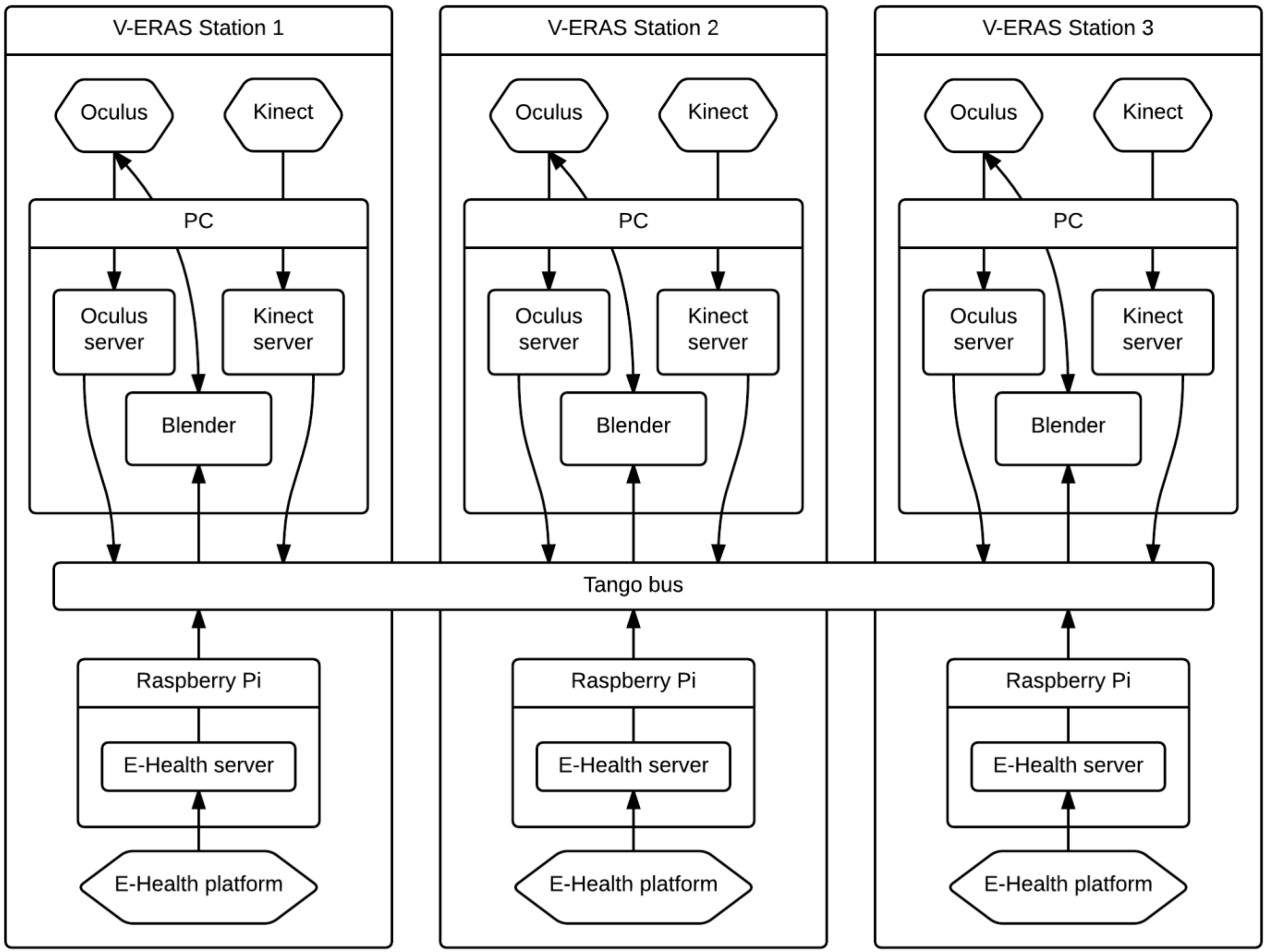
The development effort was divided in:

- Simulations development using an appropriate game engine supporting a virtual reality headset
- Full body and gestures tracking
- Integration of an omnidirectional treadmill
- Support for multiuser

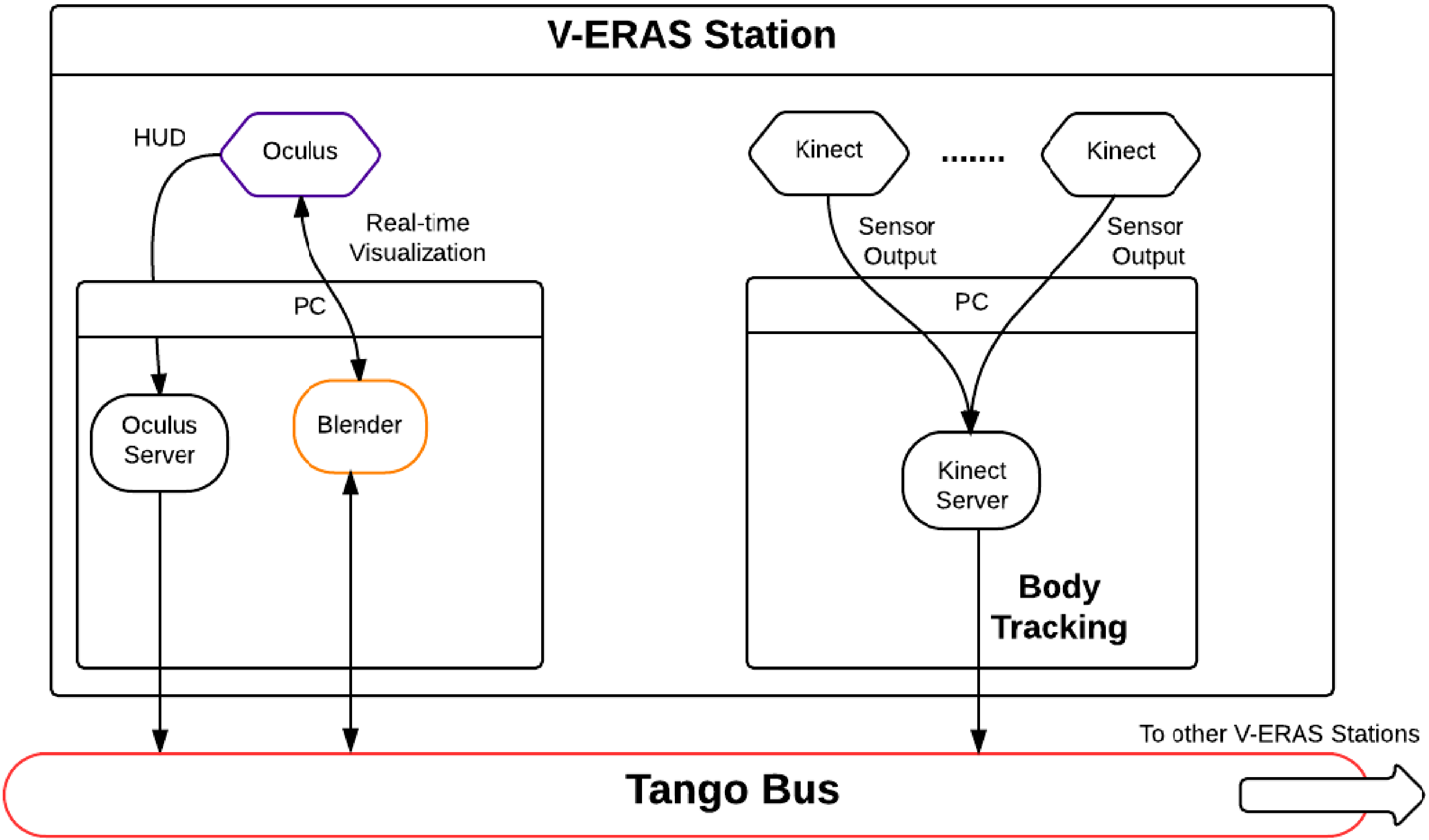
The Immersive Virtual Reality (IVR) Station

Multiuser support and initial Blender/Tango integration

E. Melotti thesis:
https://www.theseus.fi/bitstream/handle/10024/81342/Melotti_Ezio.pdf



Single IVR station latest architecture



Fielding IVR simulations

During the OeWF Amadee15 analog campaign in Austria the use of IVR has proven to be very helpful in training crews for EVAs, enabling them to familiarize themselves in advance with the terrain where their mission is to take place.

BUT……

Strong limitations directly or indirectly linked with the Blender choice remain:

- Rotation of the user's avatar in the blender scene is limited. This is a non solved issue of our Kinect integration in Blender.
- Long time was needed to port our Oculus integration from DK1 to DK2 (and is not yet completed….)
- etc. etc.

So basically most of the effort is spent in developing and maintaining up to date the VR devices integration in Blender and little time is left for science.

The Unity option

- All of this would get solved by building the IVR simulation on top of commercial game engines such as Unity. Most of the VR devices have dedicated third party plug-ins or are even natively supported.
- With Unity such things like moving to a different headset (PlayStation VR) could be relatively smooth

Open Questions

- Is there a way to move to Unity preserving some part of the development done so far in Python within Blender?
- Is there a chance to proceed in using Python for scripting development in Unity or C# is an absolute must?
- Are there impacts on the open source (BSD) licensing we have been using so far by using Unity?
- ......

Thanks for your attention!