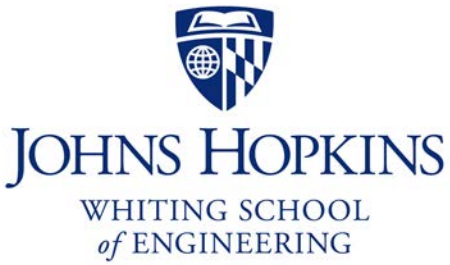
The da Vinci Research Kit (dVRK)

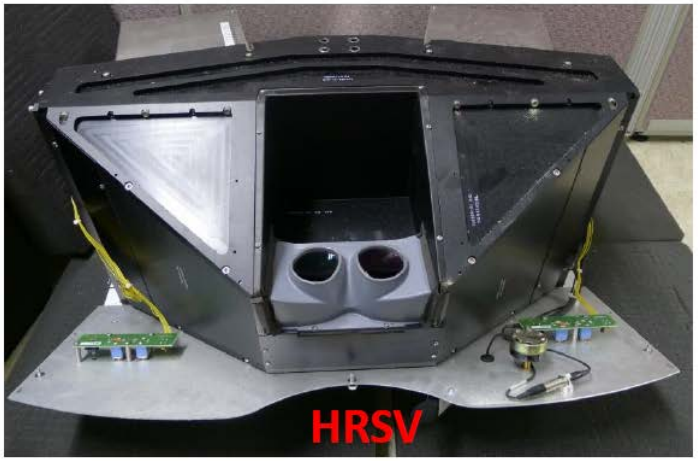
Peter Kazanzides

Dept. of Computer Science
Johns Hopkins University



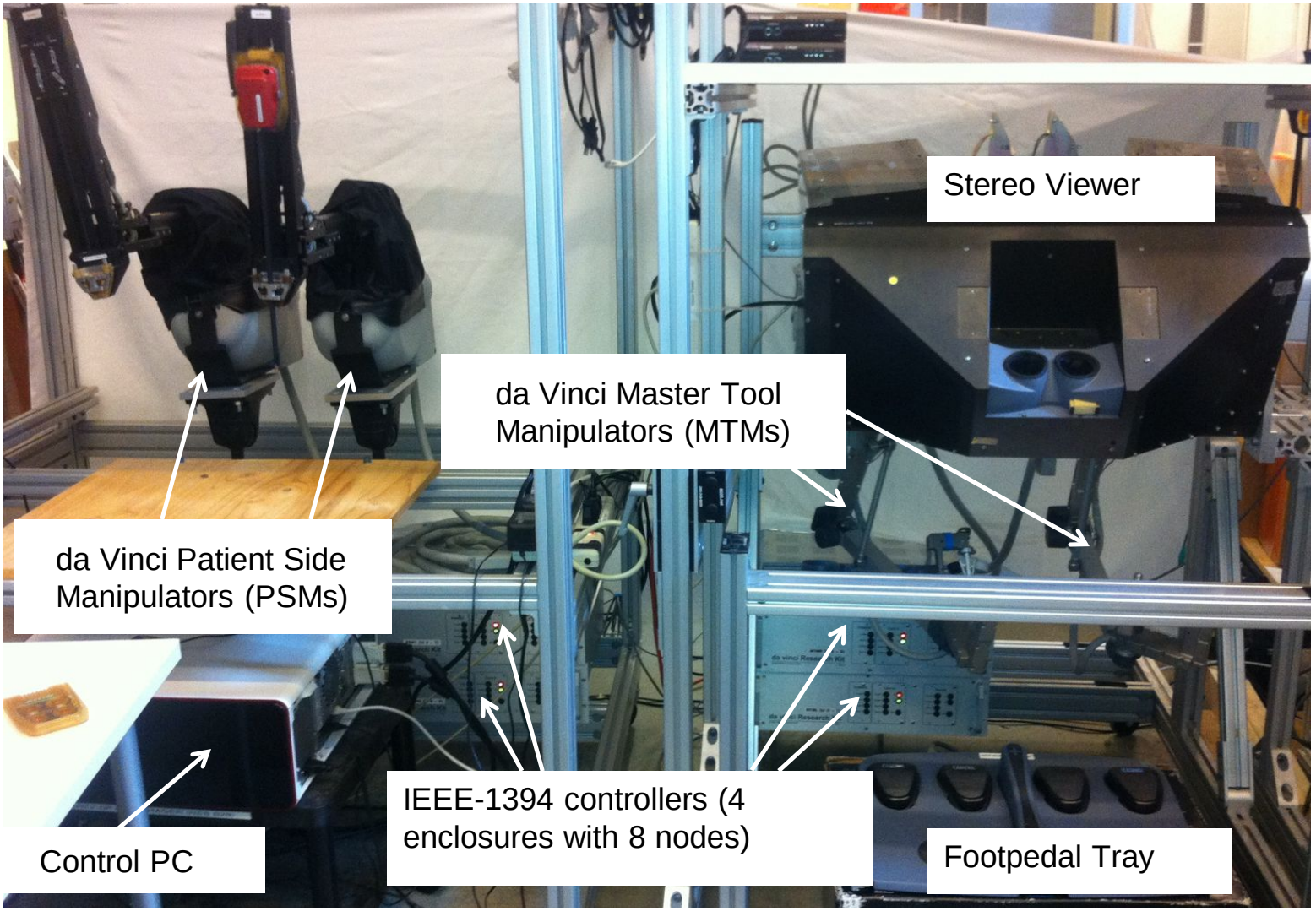
What is the dVRK?

Originally meant just the mechanical hardware components provided by Intuitive Surgical



What is the dVRK?

Then, expanded to include the open source controllers developed by JHU and WPI



What is the dVRK?

Now, can refer to any da Vinci system with open source controllers

dVRK (full da Vinci) at UBC

dVRK Community

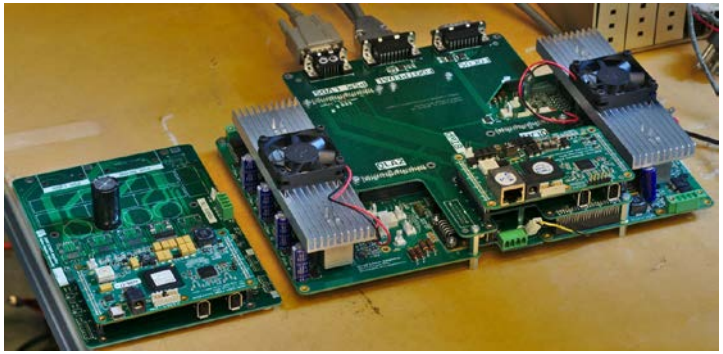
https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki

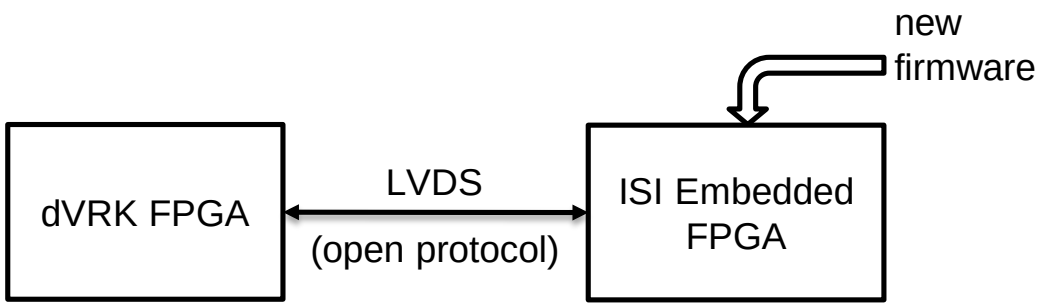
https://research.intusurg.com/index.php/Main_Page

http://jhu-cisst.github.io/mechatronics/

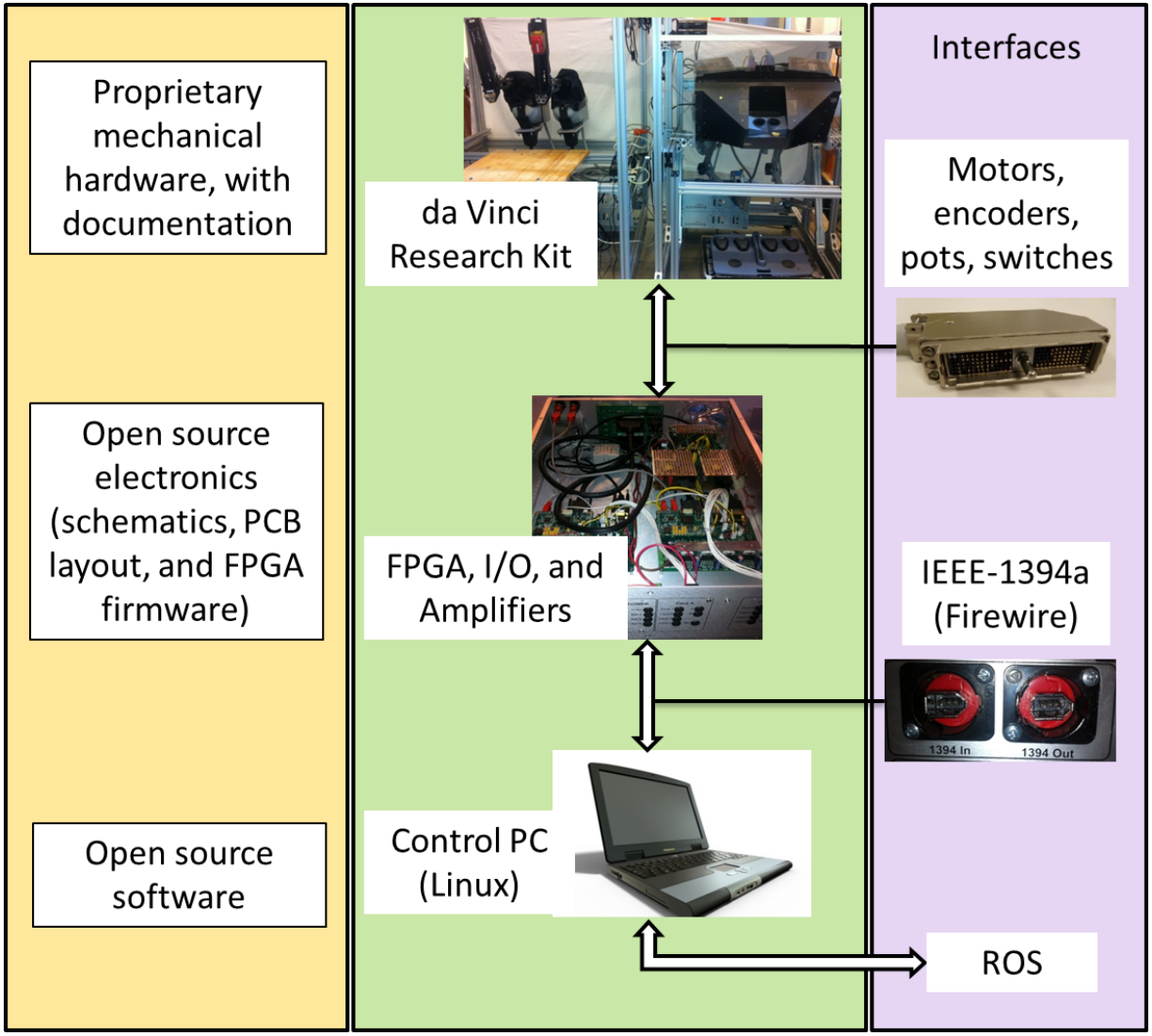
Next Generation: dVRK-S

- da Vinci S/Si:
  - da Vinci S has same MTM as dVRK
- Updated PSM (S/Si):
  - Proprietary electronics embedded in arm, with FPGA to digitize feedback and send via LVDS
- Solution:
  - Create new FPGA firmware, with open protocol
  - Add LVDS interface to dVRK FPGA, new amps





da Vinci Research Kit



Software Architecture

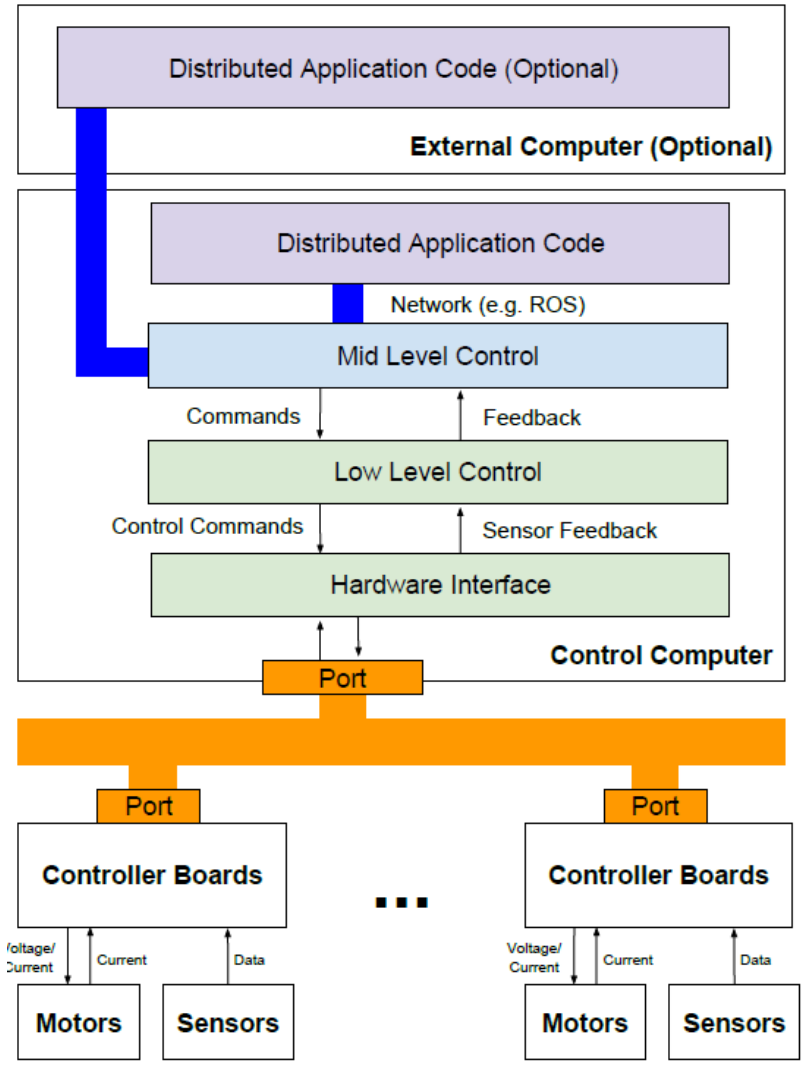
Layer

Application

Real-time Control

Fieldbus

Hardware/Firmware



Key Concerns

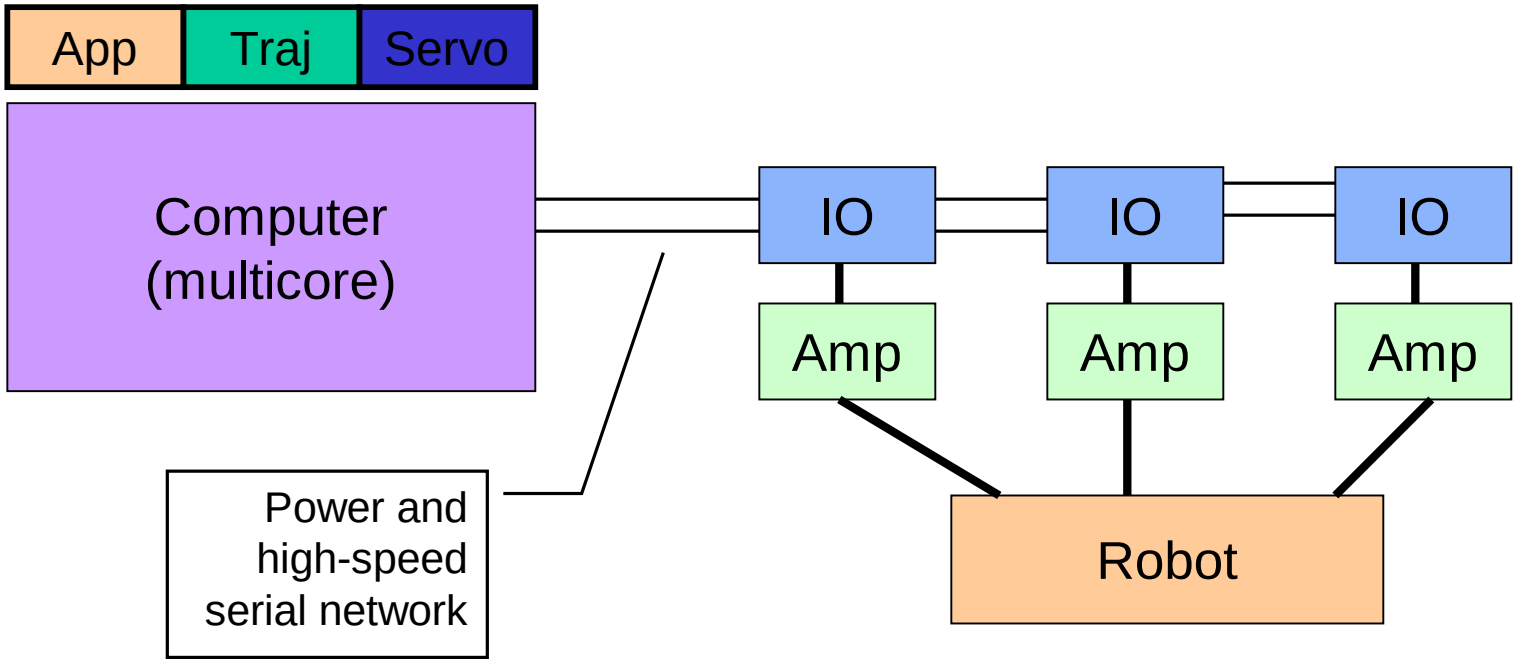
Interoperability (ROS)

Performance, ease of programming

Scalability, performance

Low latency

Centralized Computation and Distributed I/O

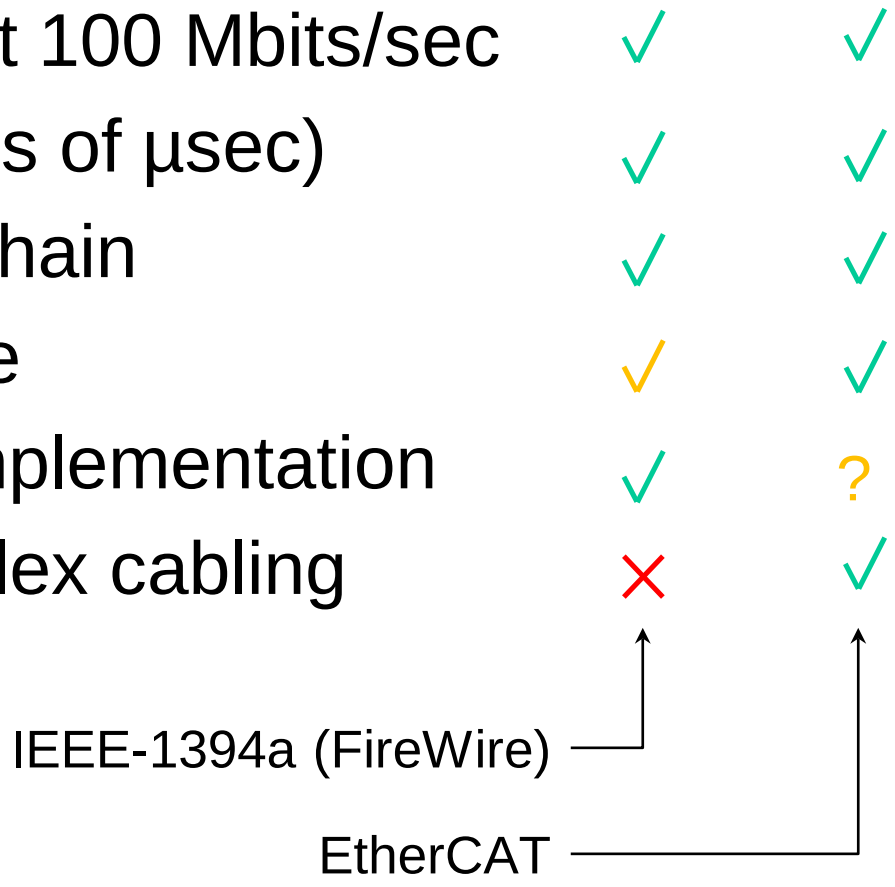


- Possible with high-performance networks/computers
  - All the benefits of distributed I/O (reduced cabling)
  - Computation on familiar development platform (flexibility)

Kazanzides, P., Thienphrapa, P., "Centralized Processing and Distributed I/O for Robot Control," *IEEE Intl. Conf. on Technologies for Practical Robot Applications (TePRA)*, 2008.

Fieldbus Requirements

- Speed of at least 100 Mbits/sec
- Low latency (tens of µsec)
- Ability to daisy-chain
- Readily available
- Simple FPGA implementation
- Option for high-flex cabling

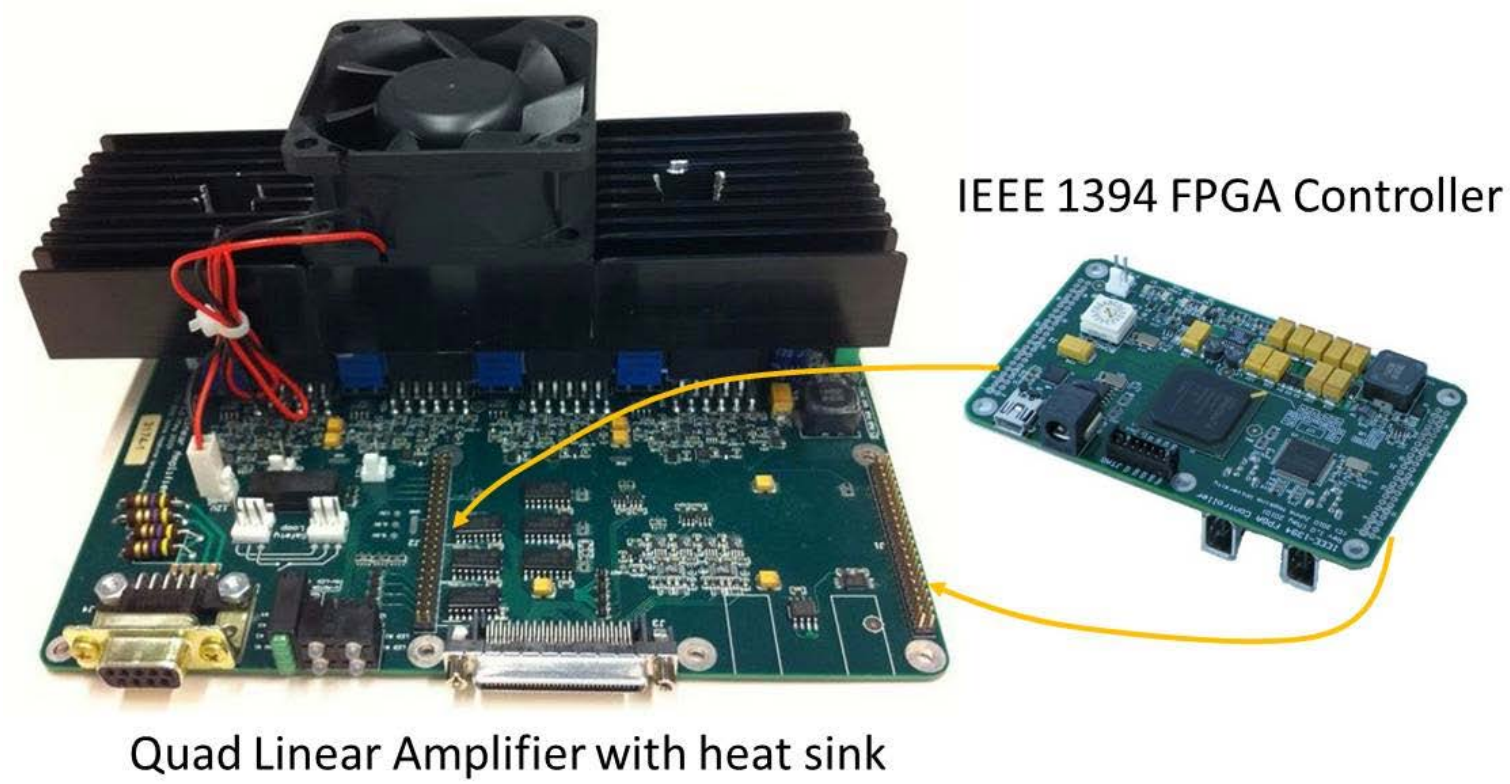


In 2006, FireWire appeared to be best choice

Today, other good options, such as EtherCAT

Hardware/Firmware layer

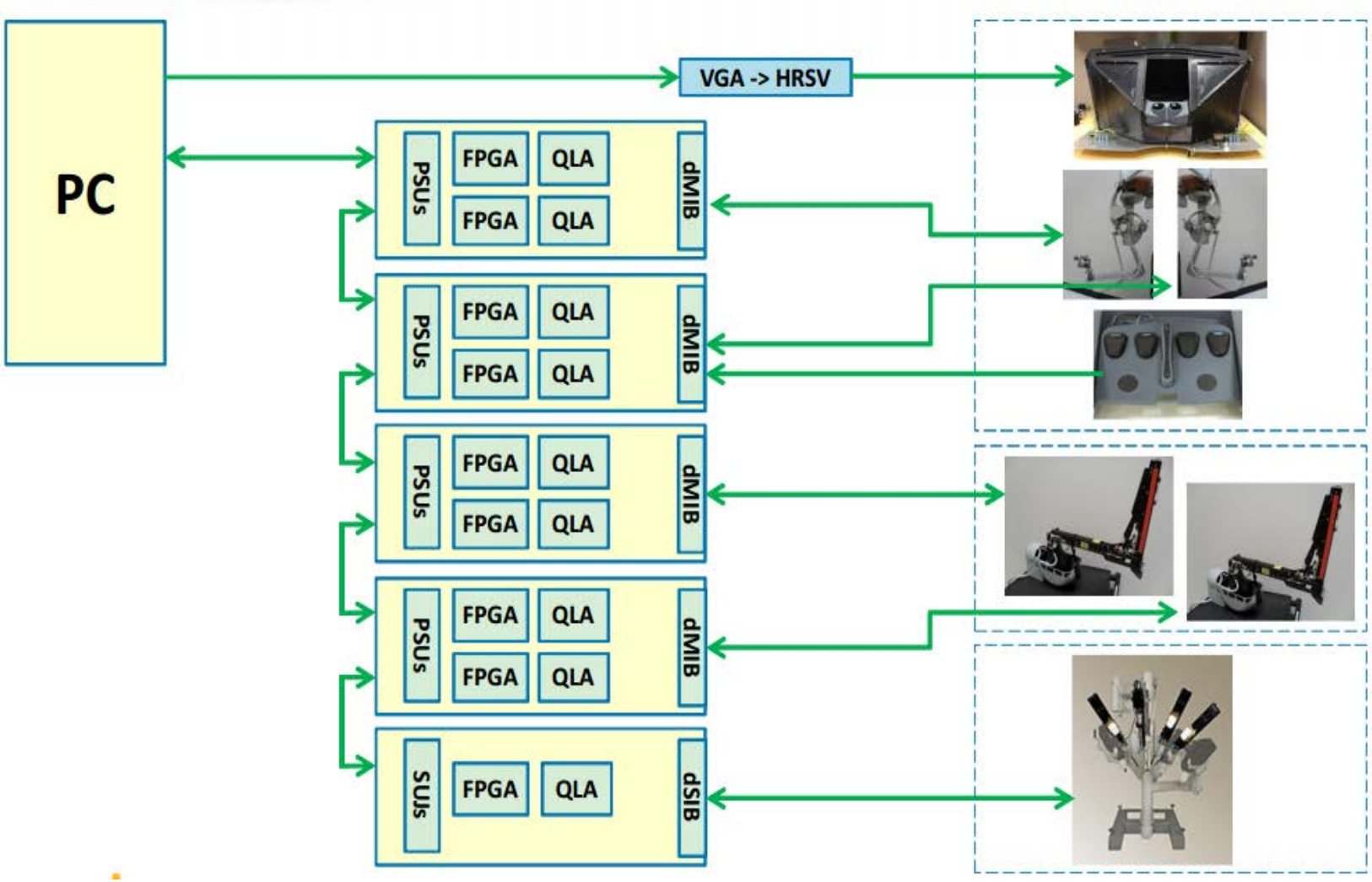
FPGA/QLA



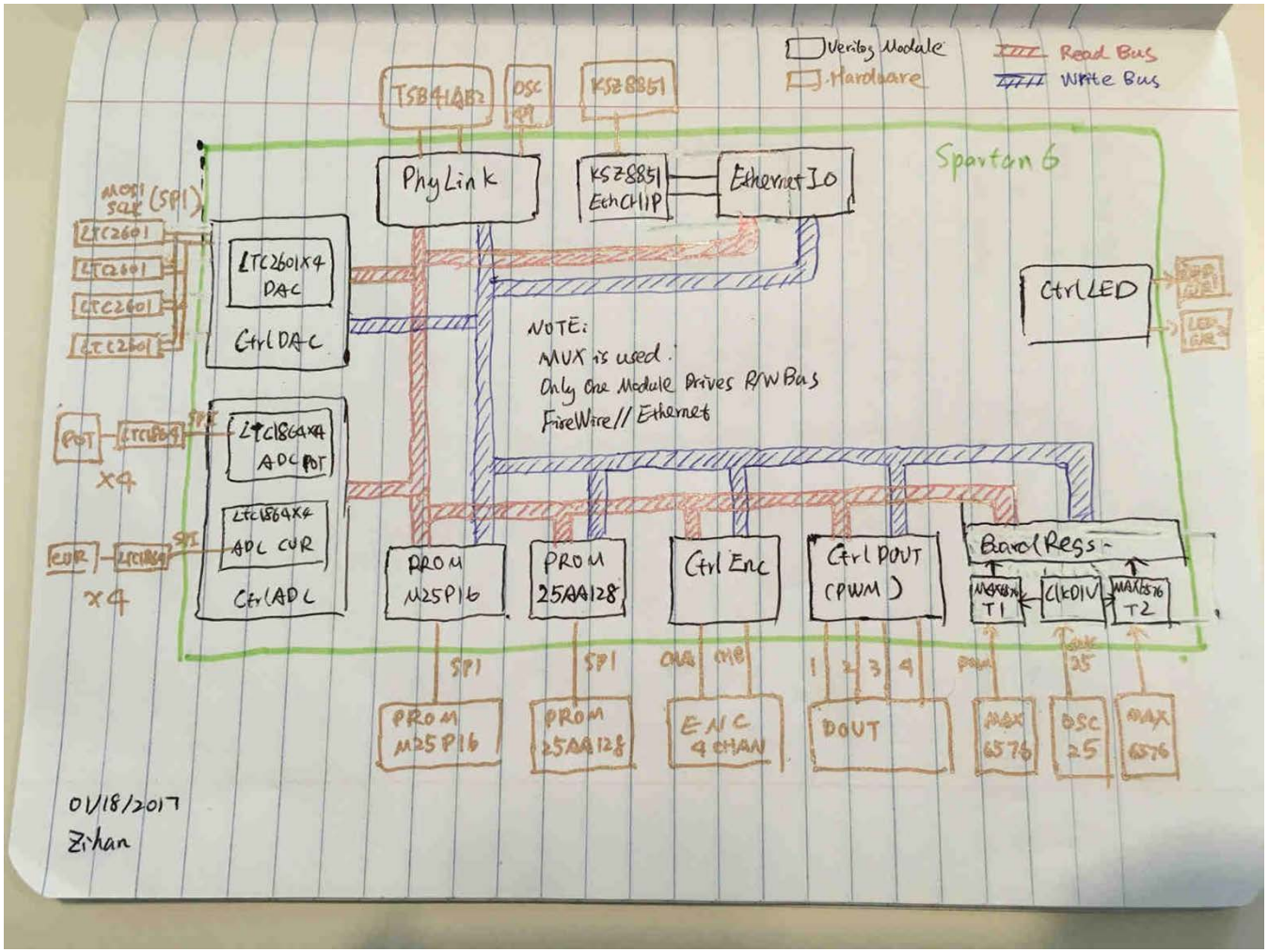
Low Latency: IEEE 1394 link layer implemented in FPGA (from packets to hardware and vice-versa)

Schematics, PCB layout, firmware (Verilog) at http://jhu-cisst.github.io/mechatronics/

da Vinci Research Kit



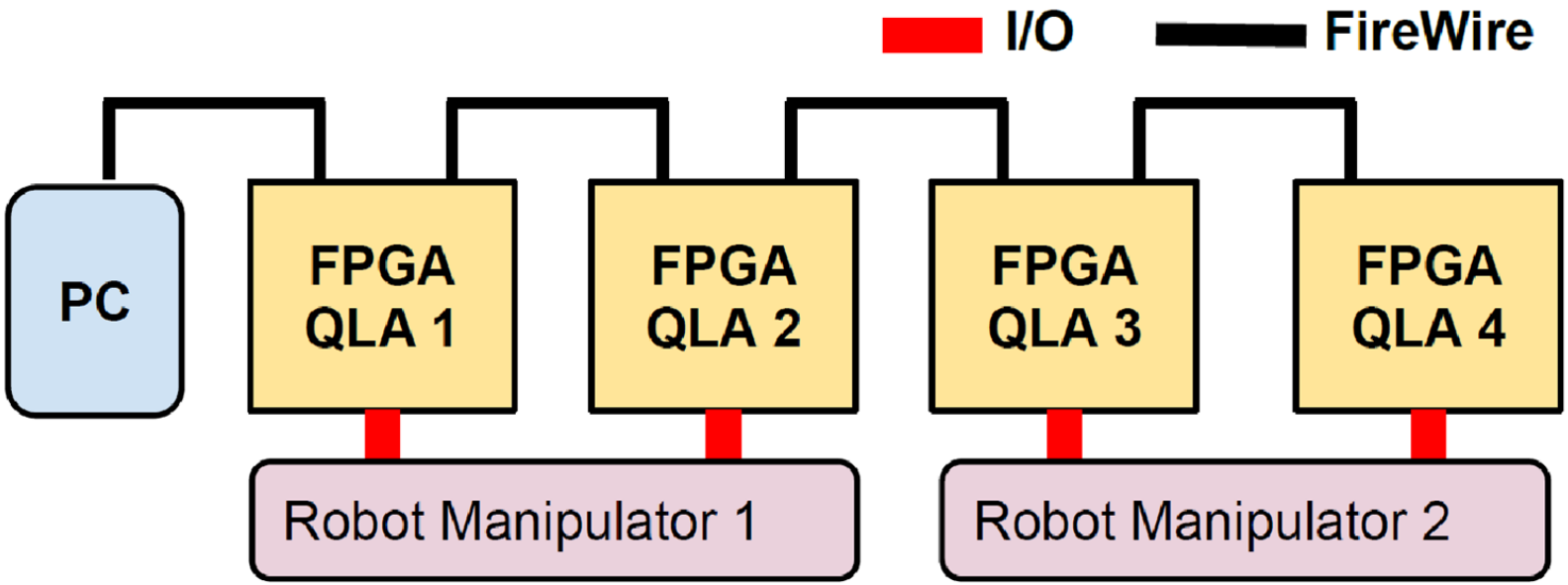
Firmware (Verilog)



https://github.com/jhu-cisst/mechatronics-firmware/wiki

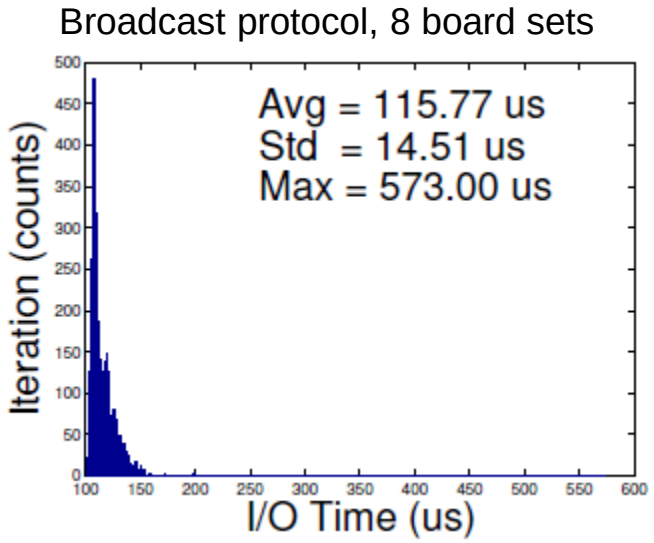
Fieldbus Layer

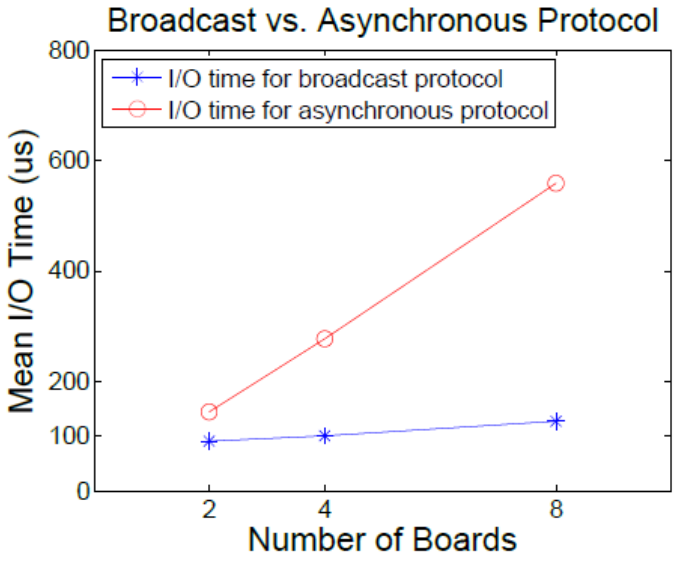
FireWire (IEEE 1394)



Fieldbus scalability

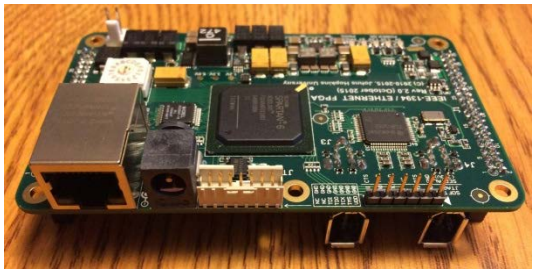
- Latency primarily due to overhead on PC
  - ~35 µs per asynchronous transaction
  - Individual read/write to 8 boards: ~530 µs
  - Taking advantage of broadcast and peer-to-peer transfers reduces I/O to ~116 µs

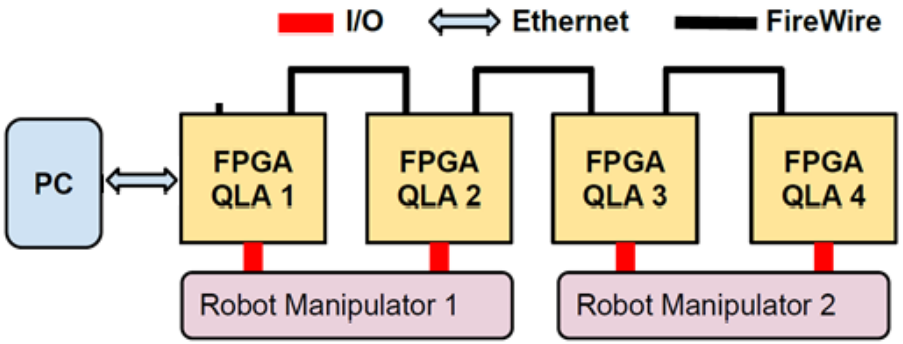


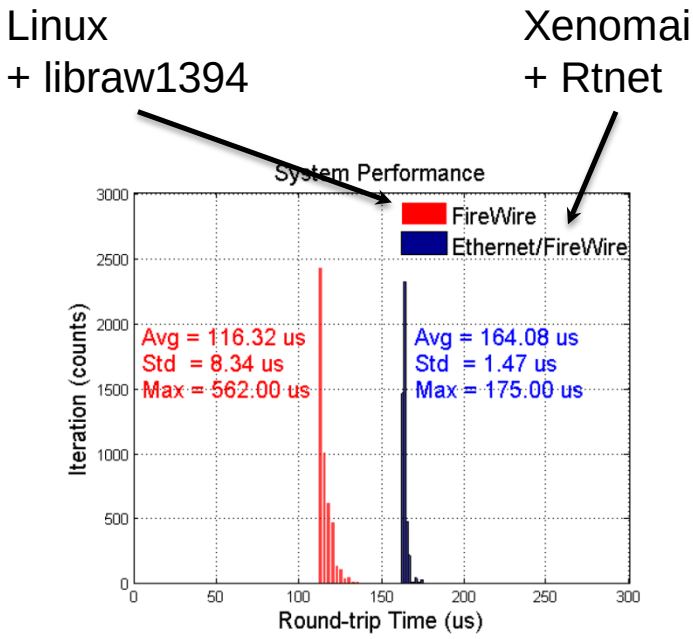


Ethernet Interface (in process)

- Ethernet is supported everywhere, including RTOS, Simulink Real-Time (Matlab xPC)
- Real-time drivers for Ethernet readily available







Qian, L., Chen, Z., Kazanzides, P., "An Ethernet to FireWire bridge for real-time control of the da Vinci Research Kit (dVRK)", *IEEE Conf. on Emerging Technologies and Factory Automation (ETFA)*, 2015.

Performance Summary

- Full da Vinci: 39 axes of control
  - Two 7-axis MTMs
  - Three 7-axis PSMs
  - One 4-axis ECM
- Desired closed loop control of 1+ kHz
  - Achieved 3 kHz in practice
- Open source and open standards
  - Ethernet and FireWire

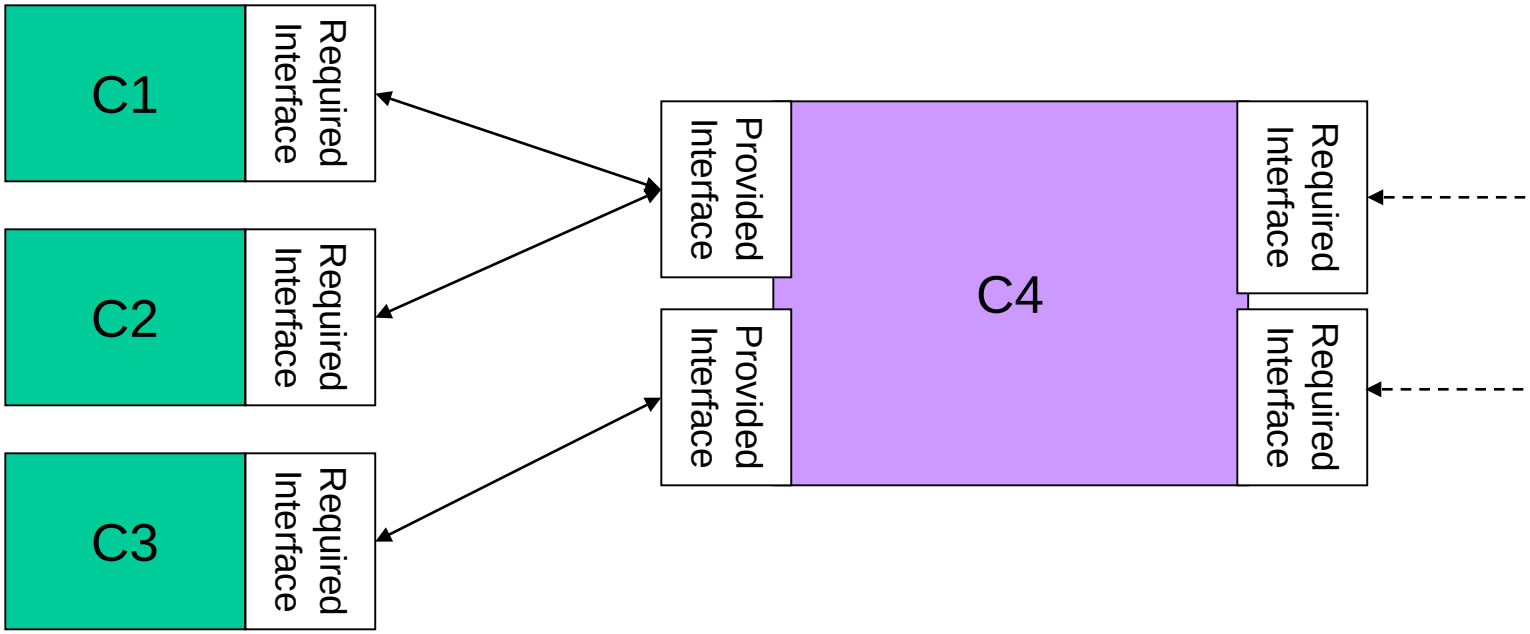
Real-Time Control Layer

Real-Time Control

- Component-based design, supporting multiple components in a single process
- Efficient communication between components in single process
- Ability to schedule components in a single thread
- Available in Orocos, cisst/SAW, …

Component-Based Software

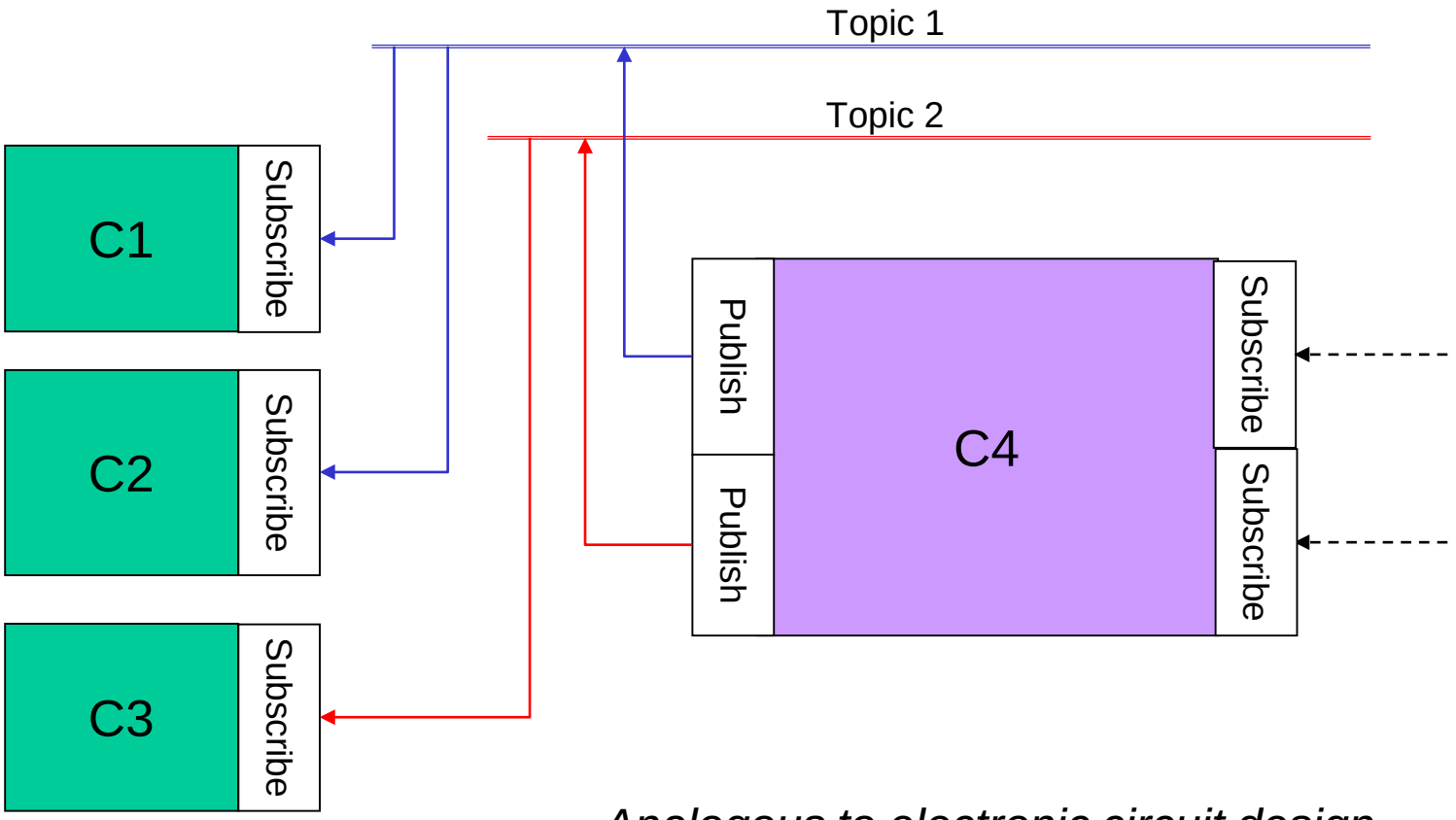
- Client/Server communication



Analogous to electronic circuit design

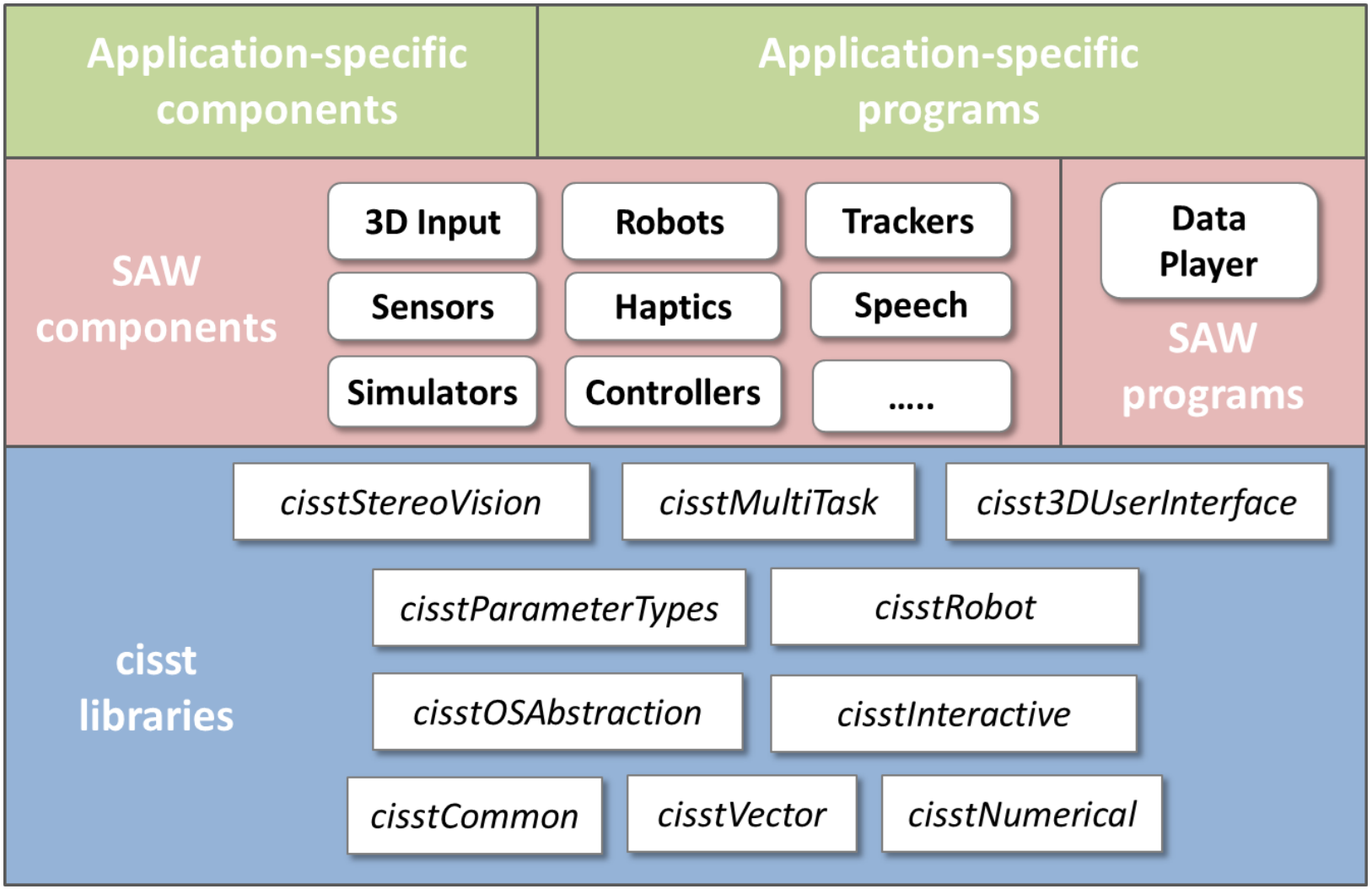
Component-Based Software

- Publish/Subscribe communication



Analogous to electronic circuit design

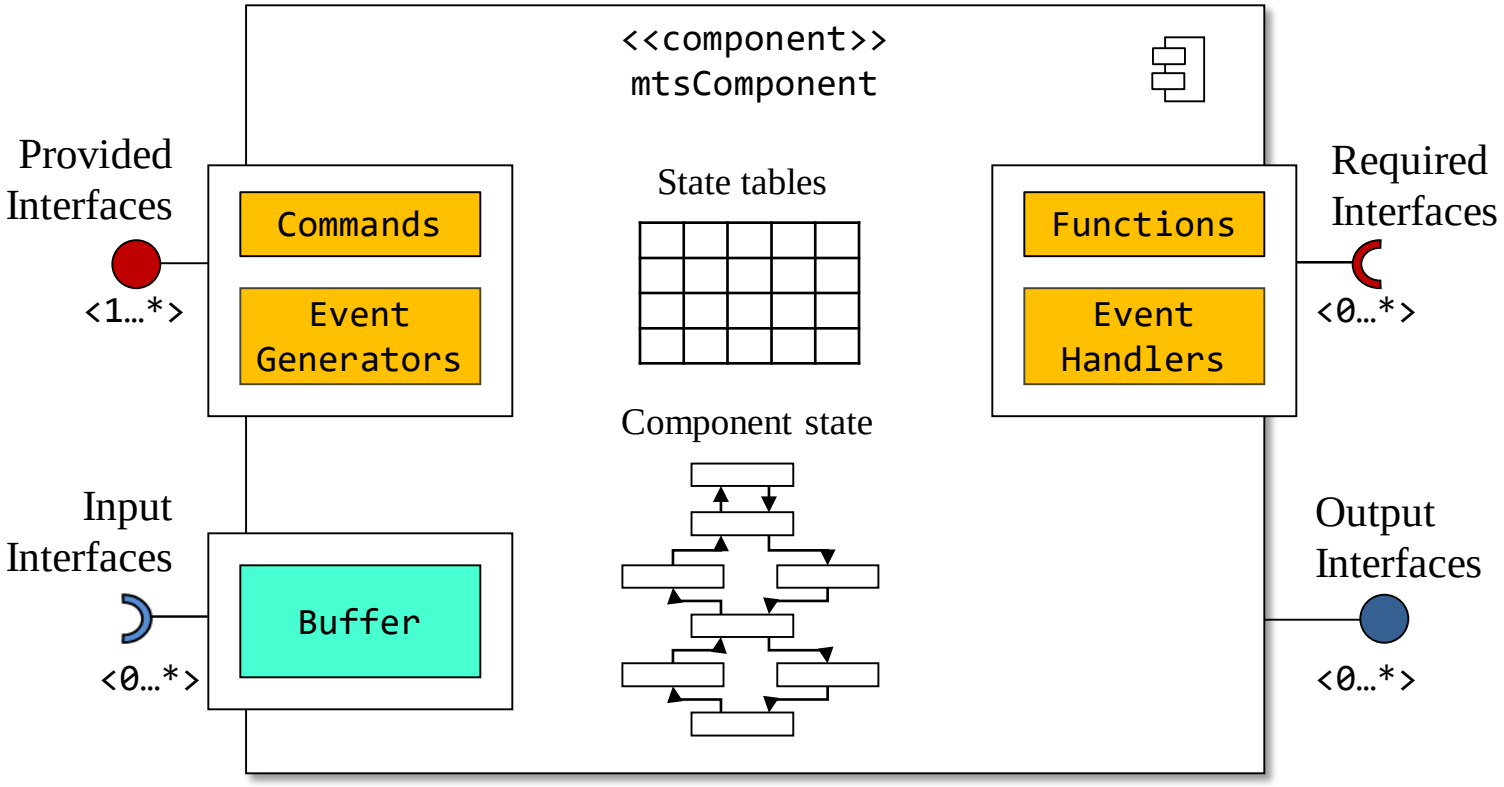
cisst/SAW Software Overview



http://github.com/jhu-cisst

http://github.com/jhu-saw

cisst/SAW Component Model

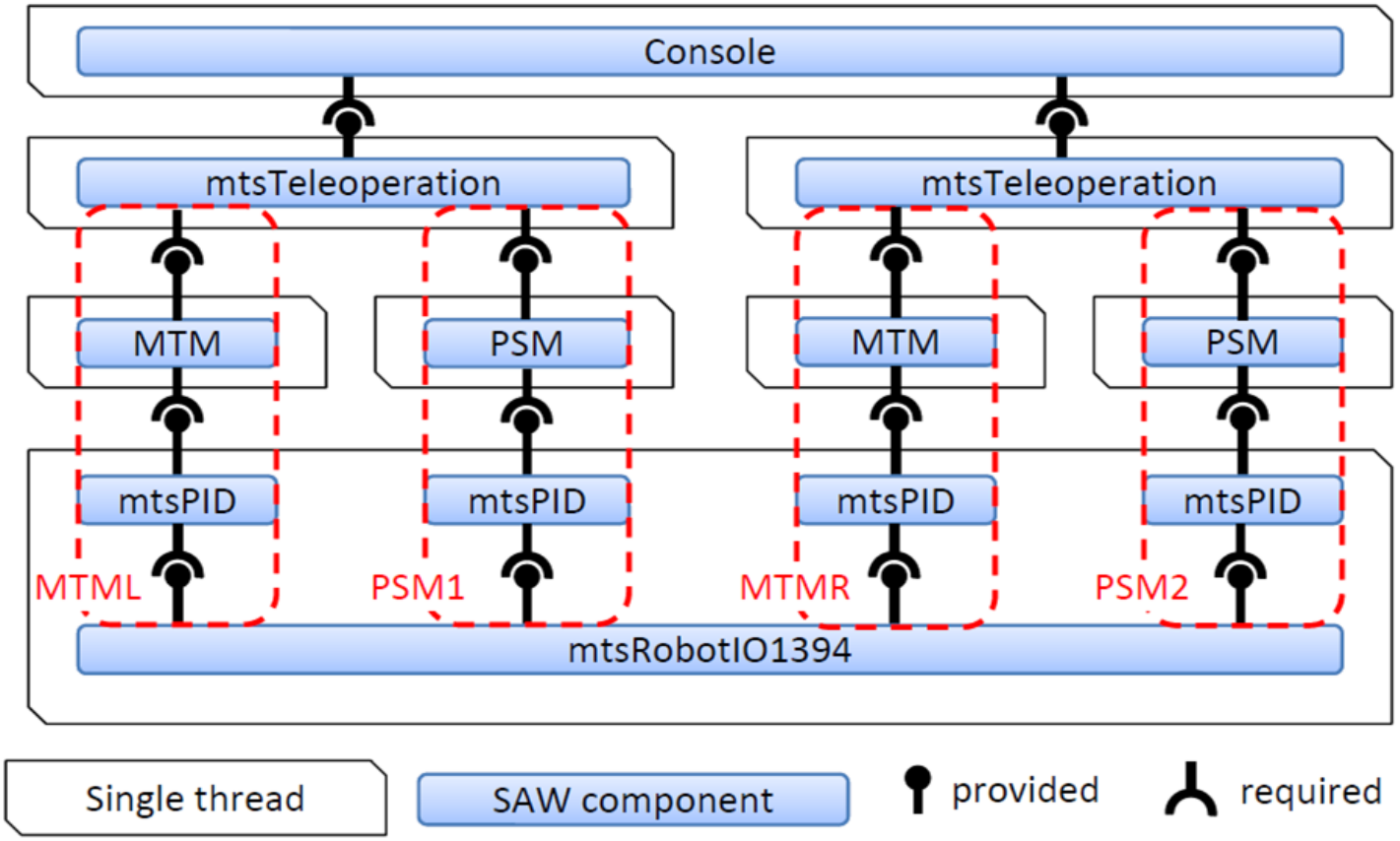


cisst commands ≈ ROS Services
cisst events ≈ ROS topics

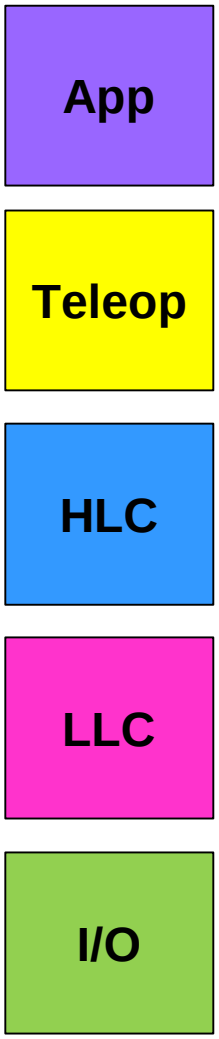
Application Layer

(Middleware)

da Vinci Teleoperation



SAW Components for dVRK



mtsIntuitiveResearchKitConsole (dVRK)

mtsTeleoperation (generic)

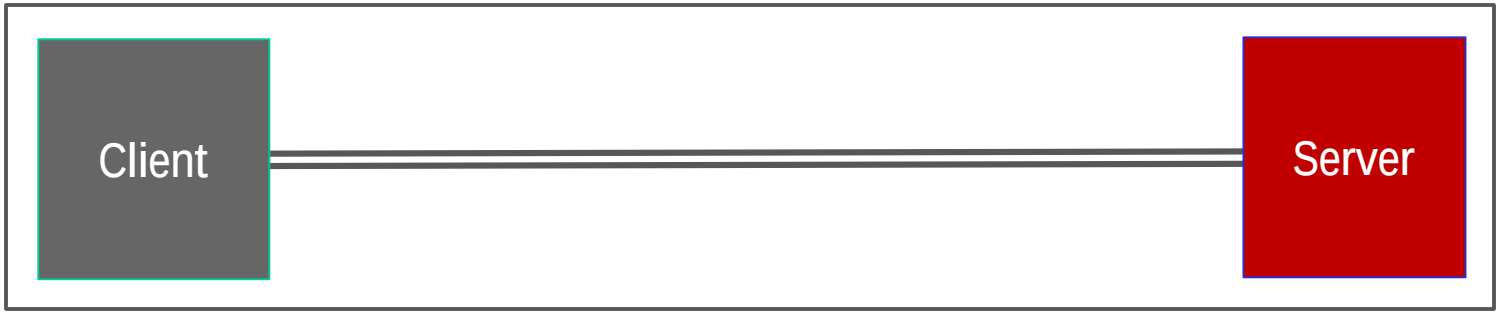
mtsIntuitiveResearchKitMTM (dVRK)
mtsIntuitiveResearchKitPSM (dVRK)

mtsPID (generic)

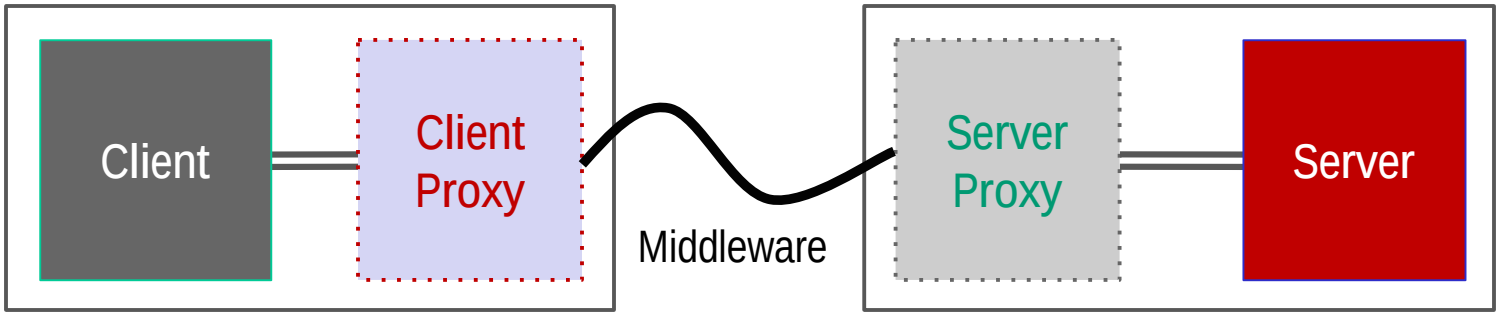
mtsRobotIO1394 (generic)

Inter-Component Communication

- Within process, cisst uses internal mechanisms (state table and queues)

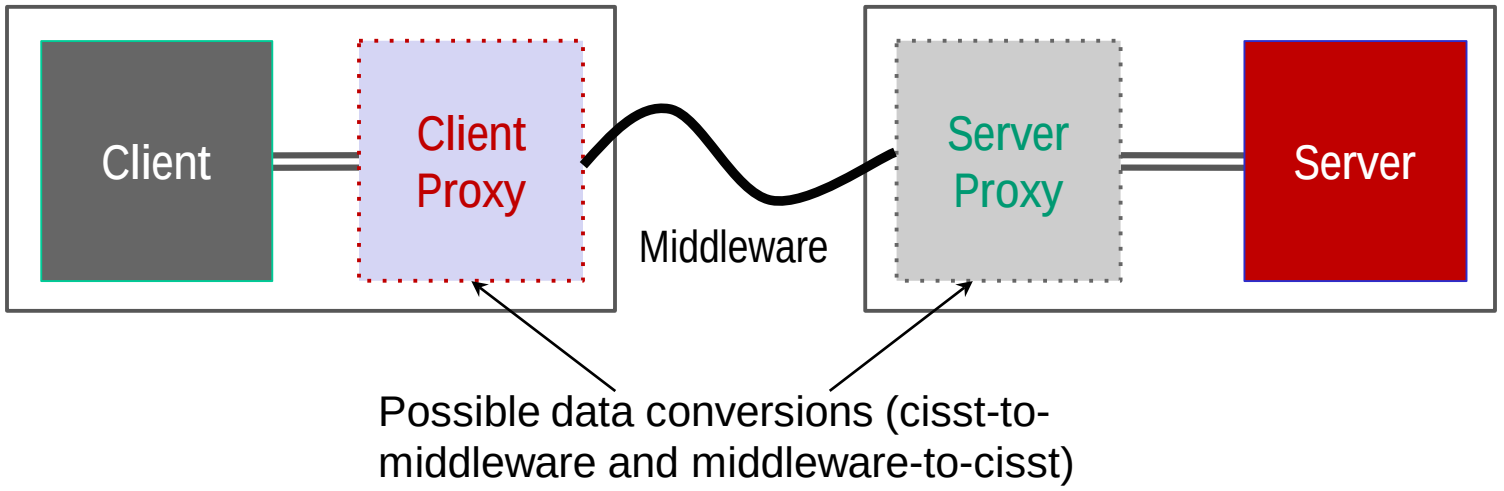


- Between processes, require proxy objects

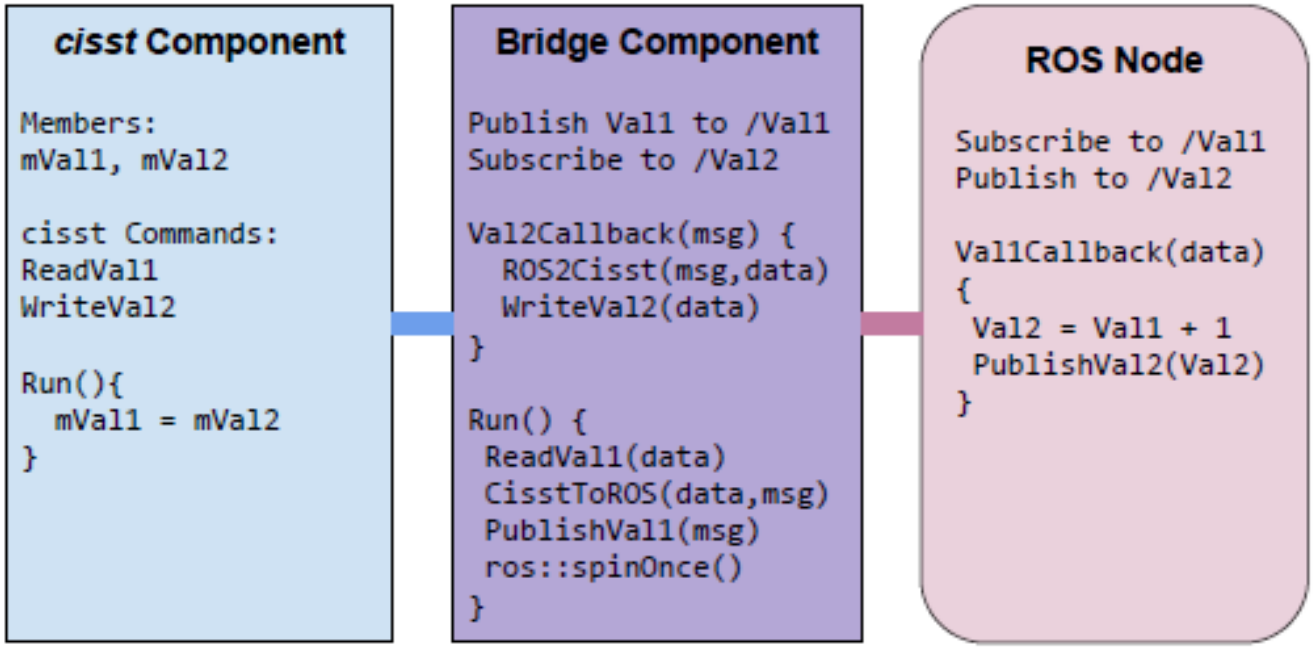


Proxy Component Options

- Robot Operating System (ROS)
- Internet Communication Engine (ICE)
- Raw Sockets (UDP)
- OpenIGTLink
- Others…



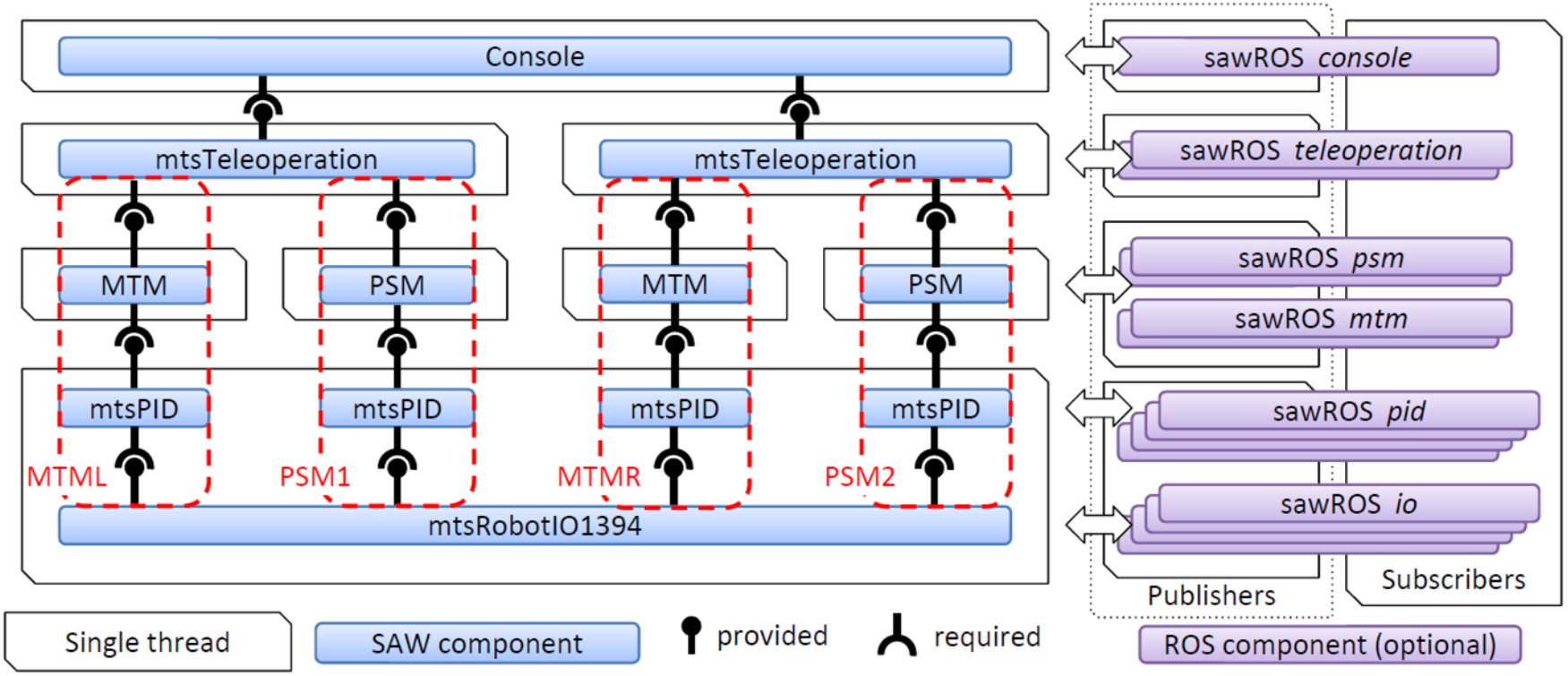
System Integration via ROS



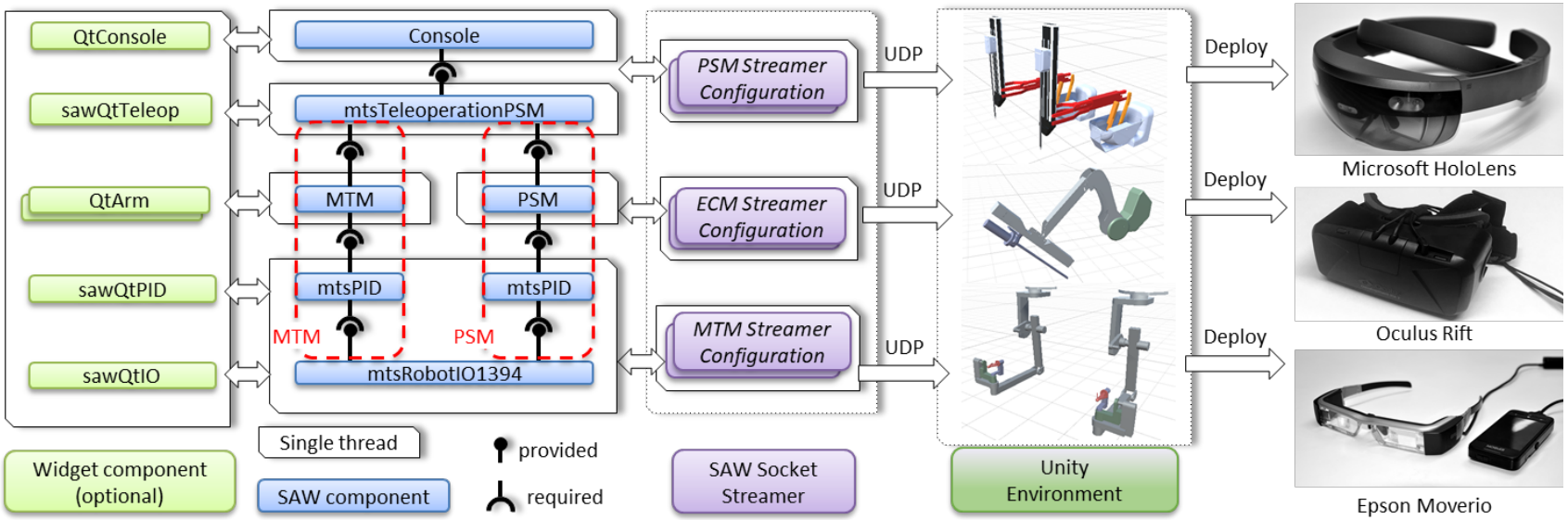
cisst/ROS Bridge Component

Bridges to other middleware, such as OpenIGTLink, also available

da Vinci Teleoperation (with ROS)



dVRK-XR: Socket Streamer to Unity



https://github.com/jhu-dvrk/dvrk-xr