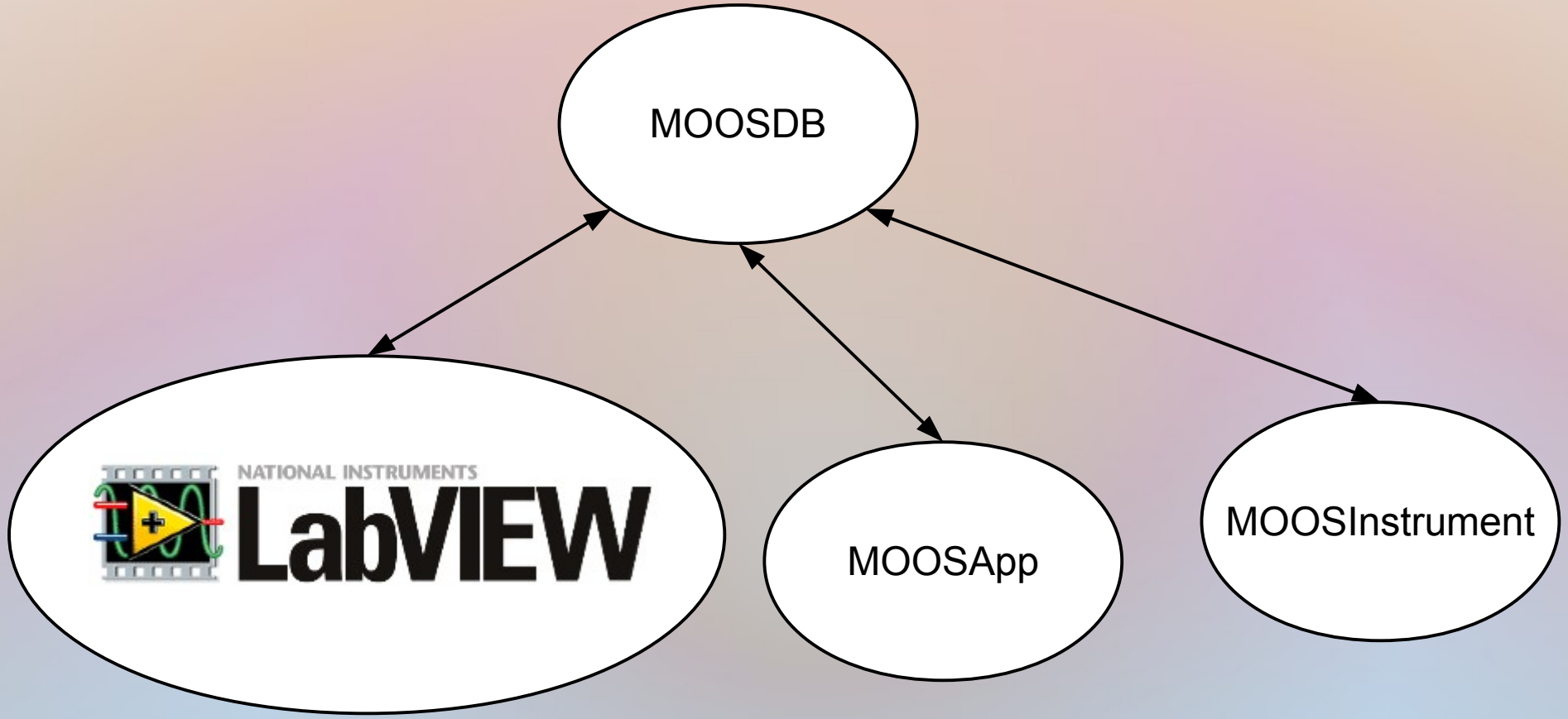
STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

Native LabVIEW MOOS interface

Alexander Sedunov

Native LabVIEW MOOS interface

- MOOS communication library based only on built-in Labview functions

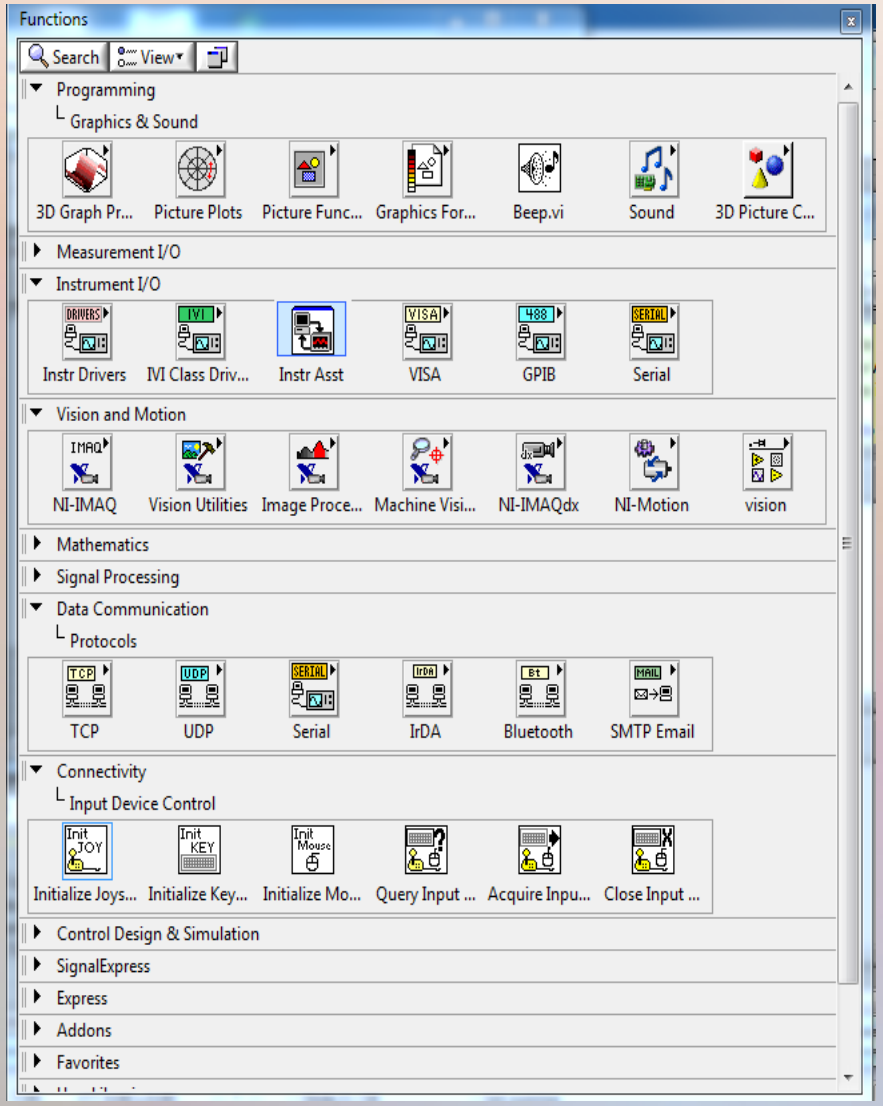


Presentation outline

- Why use LabVIEW?
- Labview interface
- Why native implementation?
- Replicating CMOOSCommClient
- Maintaining the interface
- Demonstration

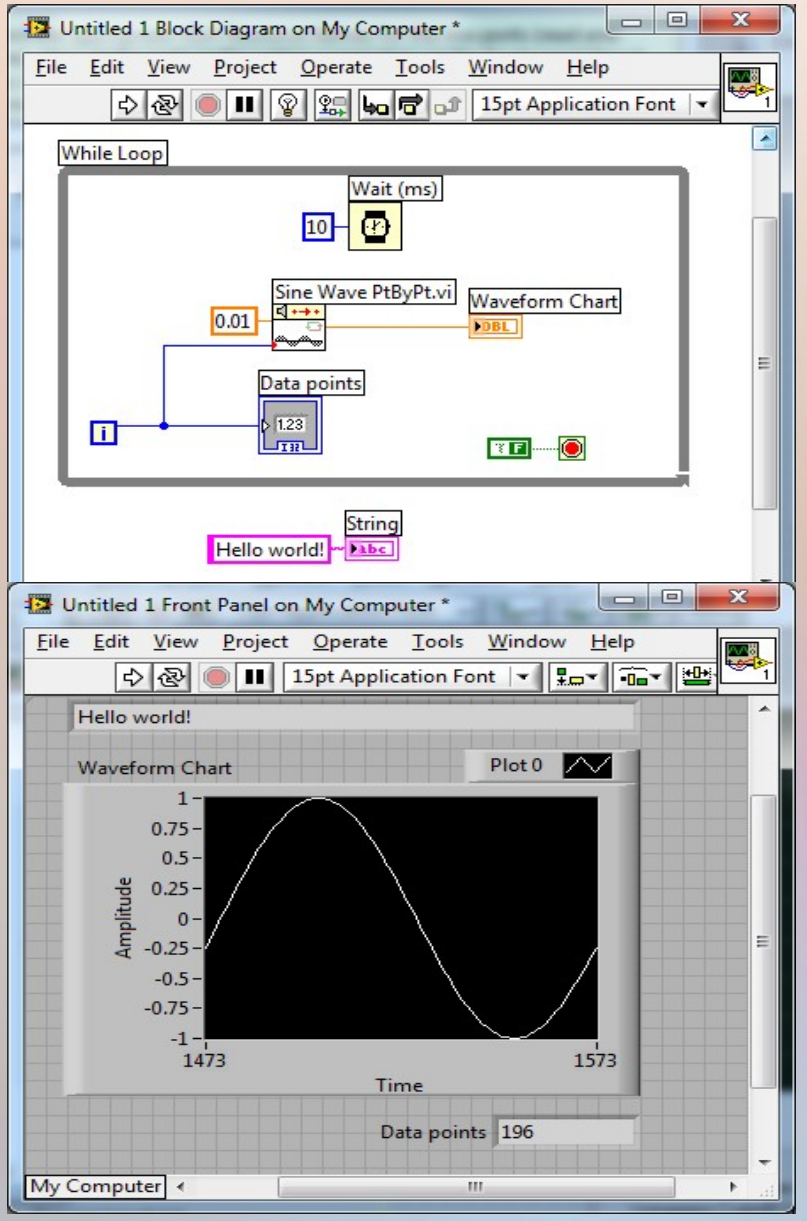
Why use NI LabVIEW

- Rapid prototyping
- Hardware interfacing
- Instant portability to supported platforms



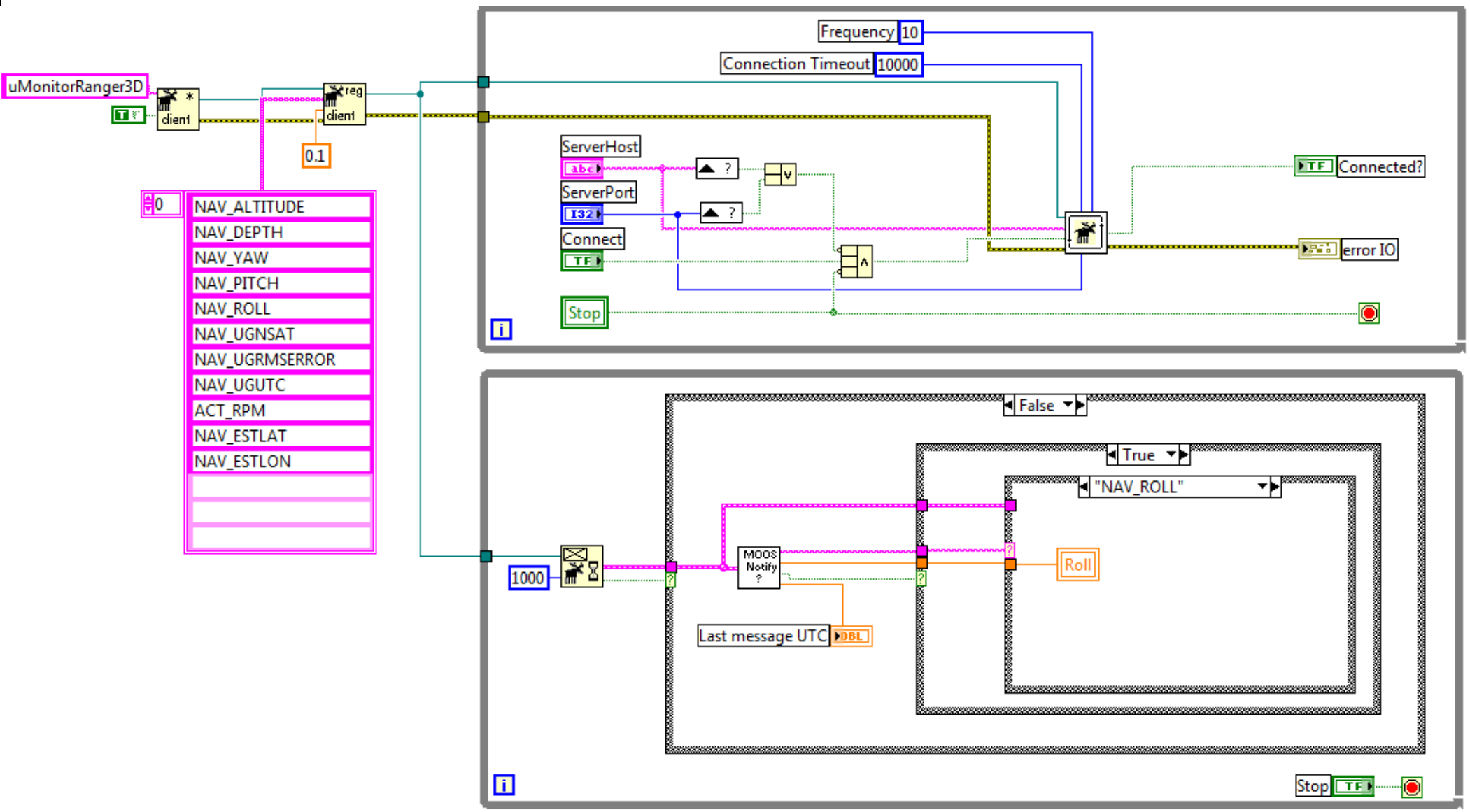
Virtual Instruments (VI)

- Programs in Labview are called ”Virtual Instruments”
- Consist of diagram and Front Panels

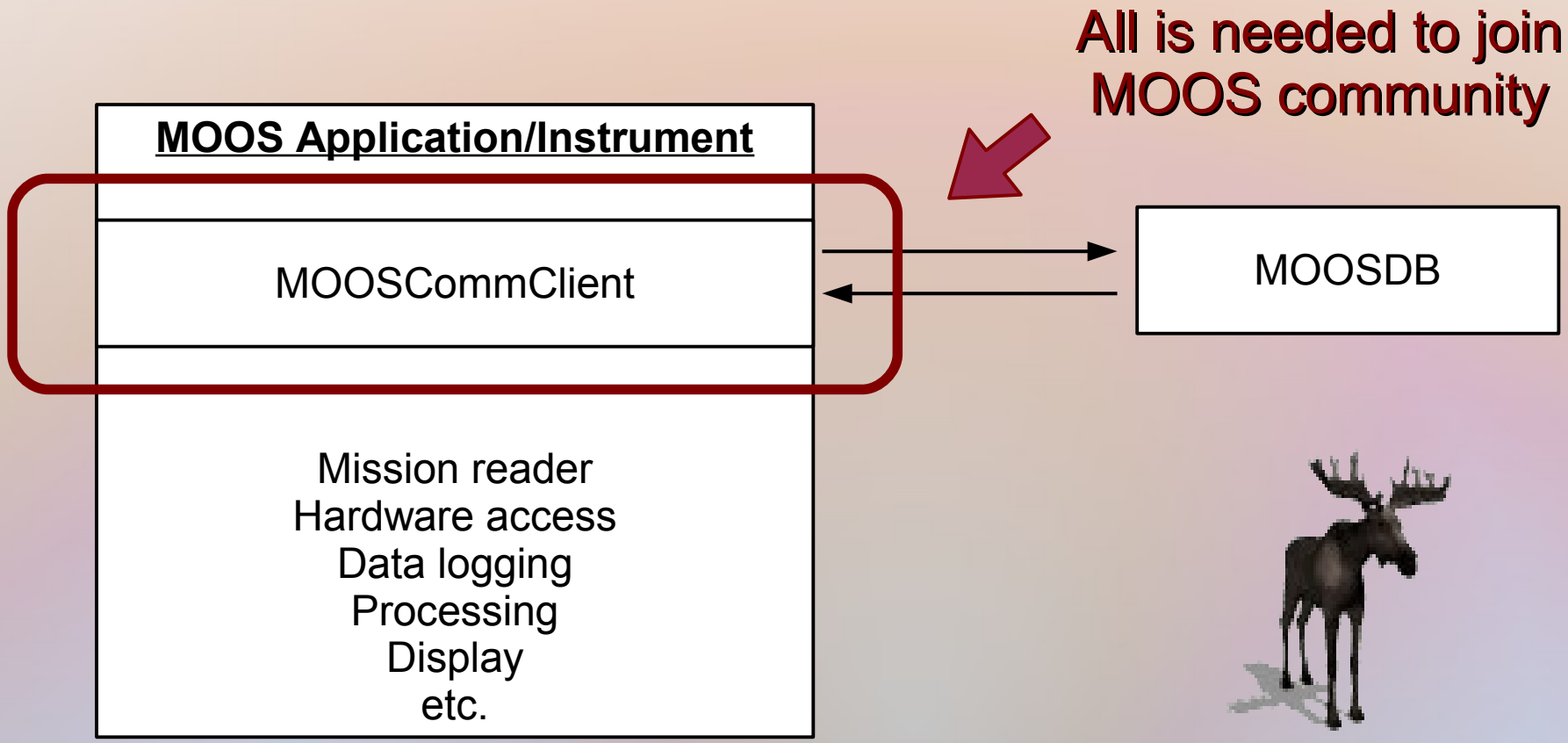


The Native MOOS Interface API

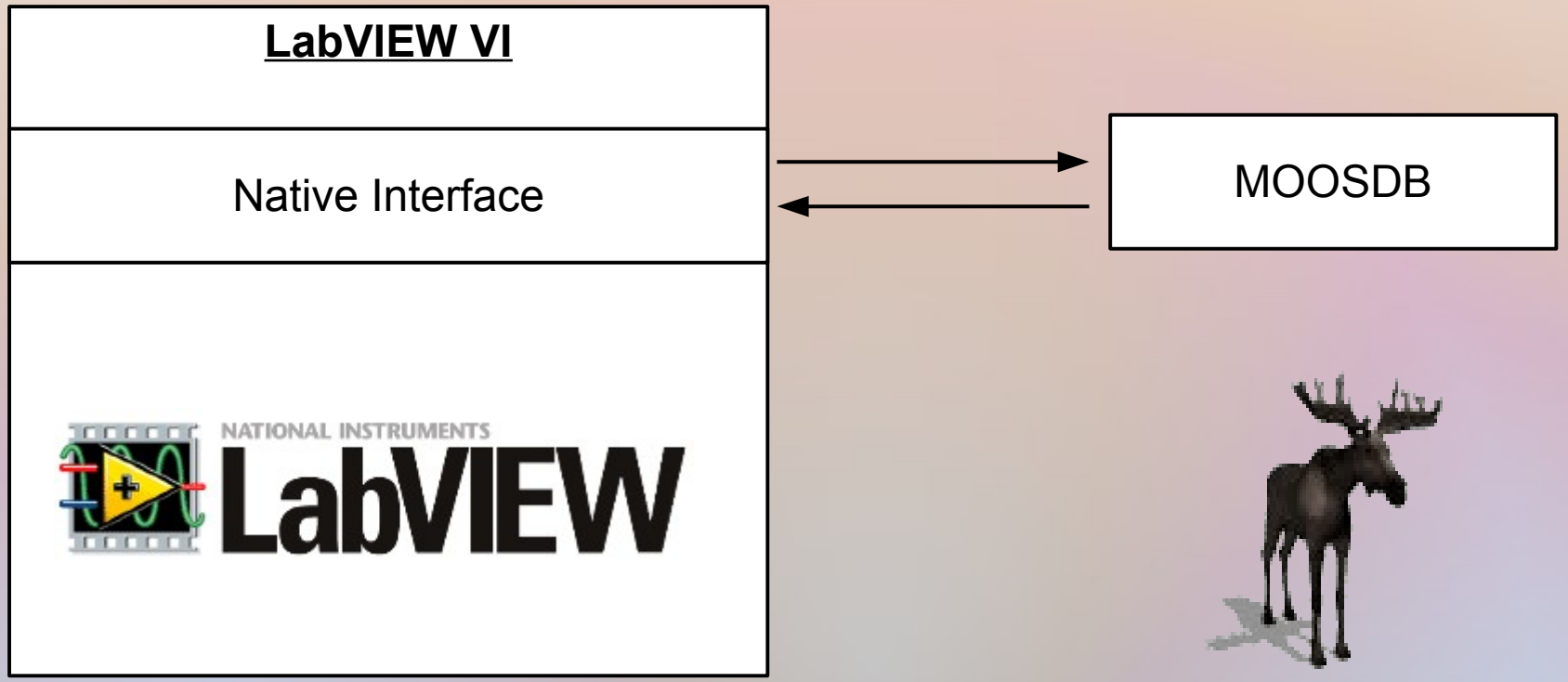
Sample LabVIEW diagram



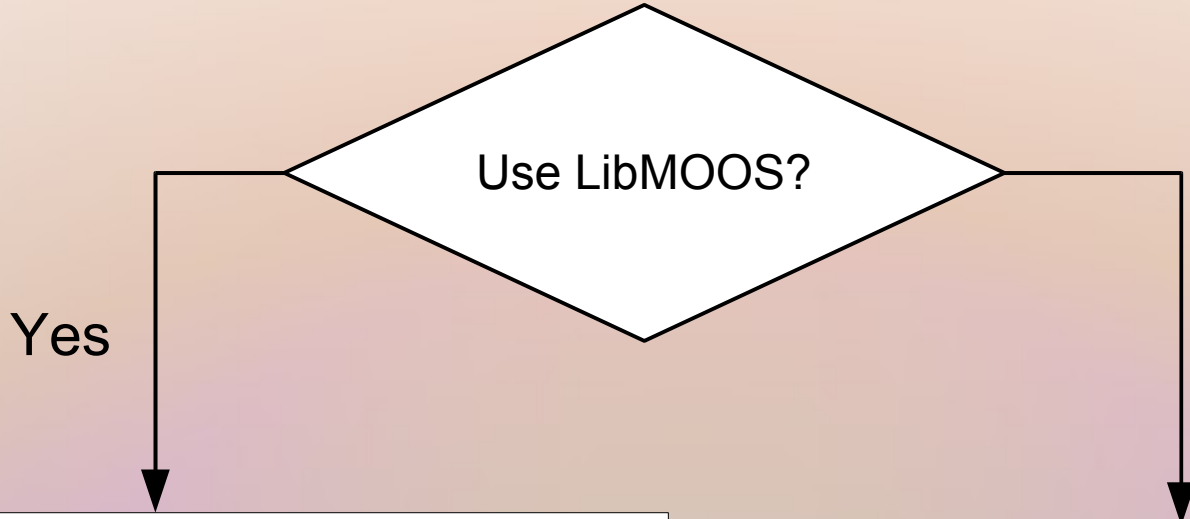
MOOS Community



Native LabVIEW MOOS Interface



Alternative ways of integration

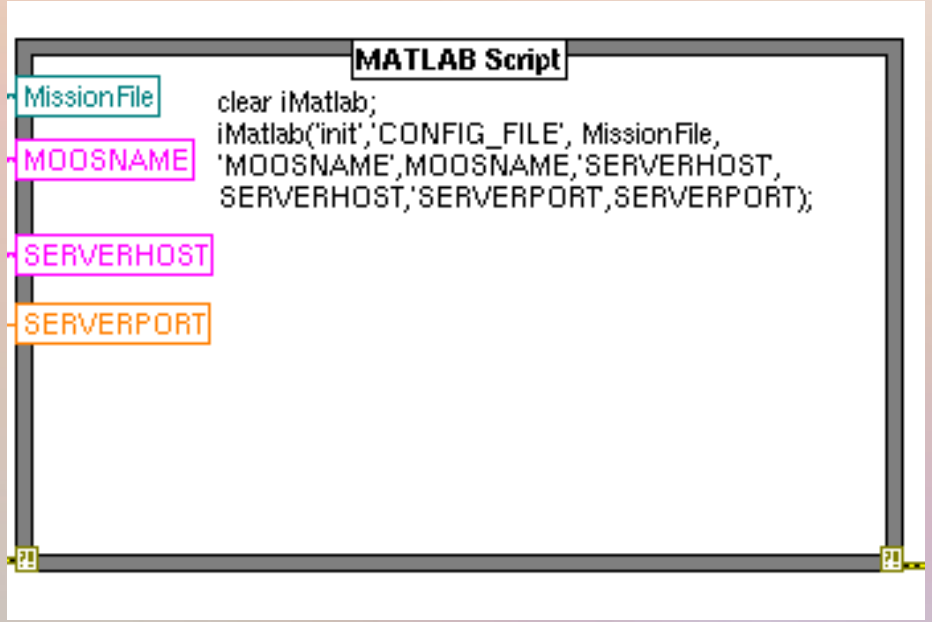


- Call iMatlab via ActiveX
- Use CIN
  (Code Interface Node)

- Reimplement MOOSClient in Labview

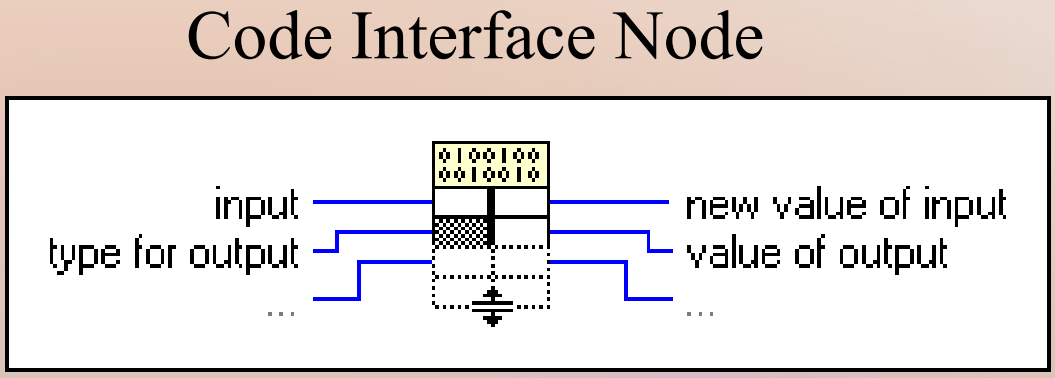
Call iMatlab via ActiveX

- ActiveX is Windows only
- Considerable overhead
- Only one instance per computer!



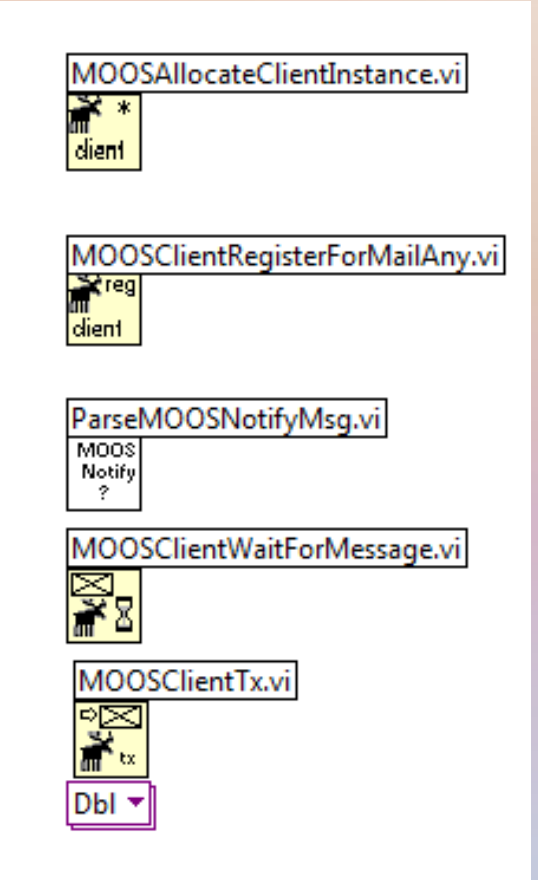
Code Interface Node

- Single entry point
- Callbacks for loading and unloading VI
- Everything has to be encapsulated into single function



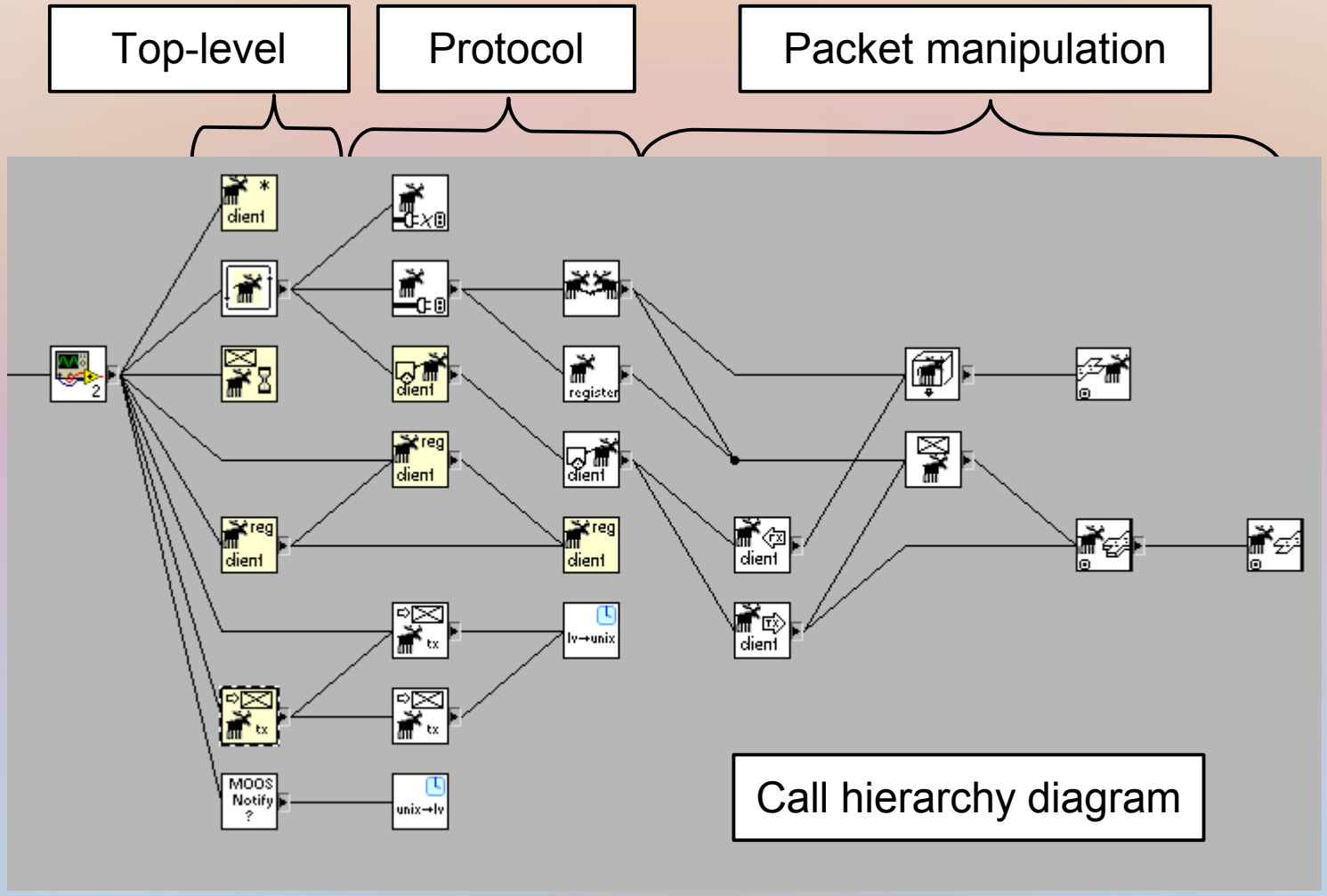
Reimplement MOOSClient in Labview

- Resources managed automatically
- Unlimited instances per machine
- Low-level Vis exposed

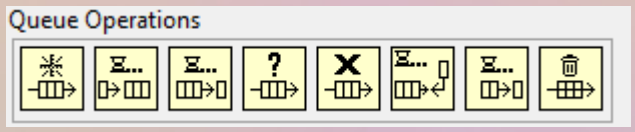
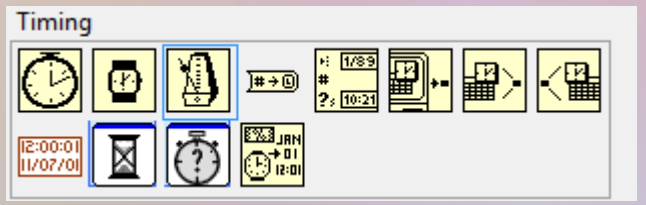
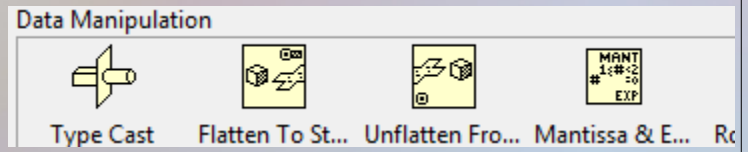


Reimplement MOOSClient in Labview

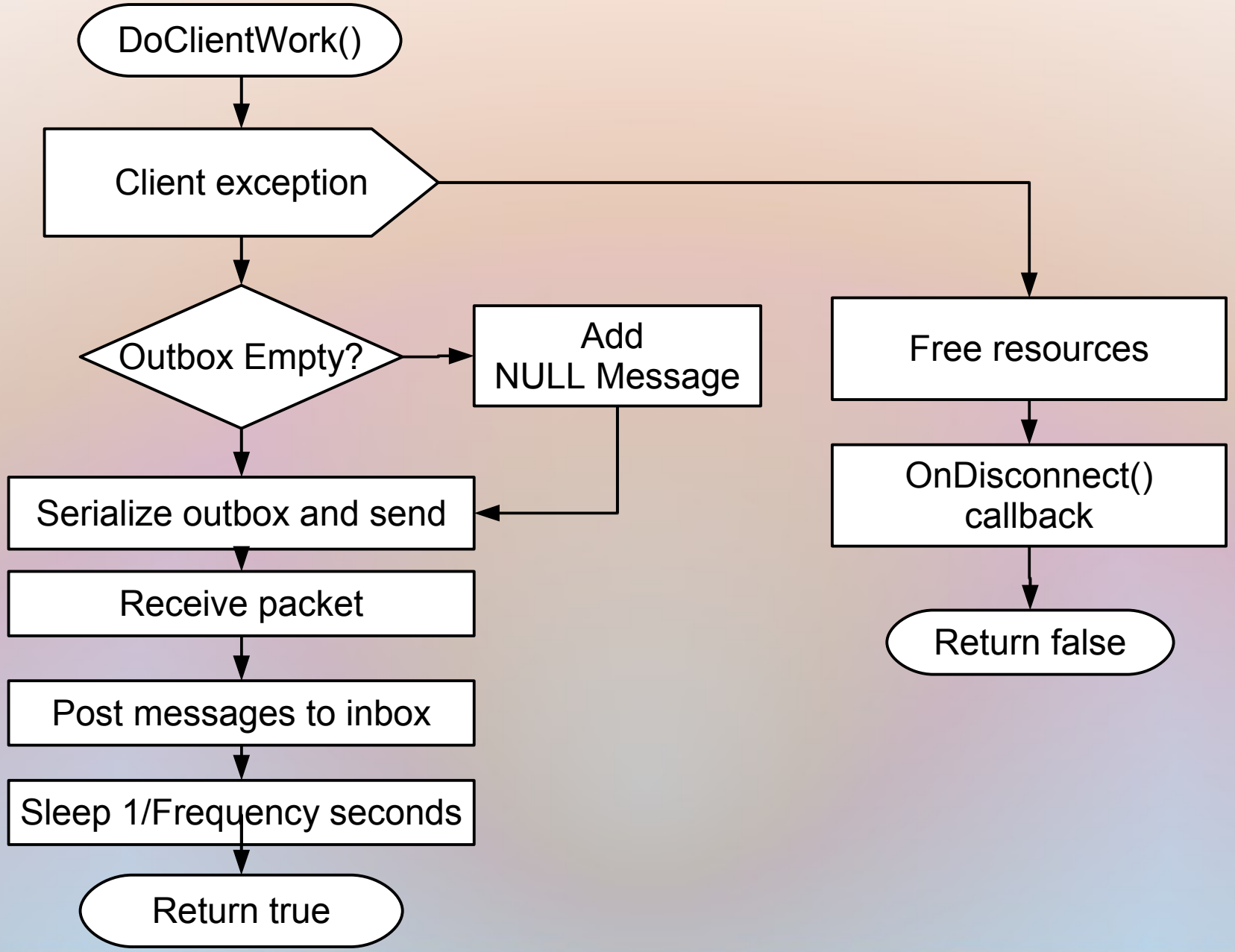
Low-level VIs exposed to developer



LibMOOS vs LabVIEW functionality

Function		NATIONAL INSTRUMENTS LabVIEW
Threads	MOOSThread.h	Implicit parallelism
Sockets	XPCSocket.h	
Queues	MOOSMSG_LIST	
Time	MOOSGenLibGlobal Helper.h	
Serialization	MOOSMsg.h	

Replicating ClientThread



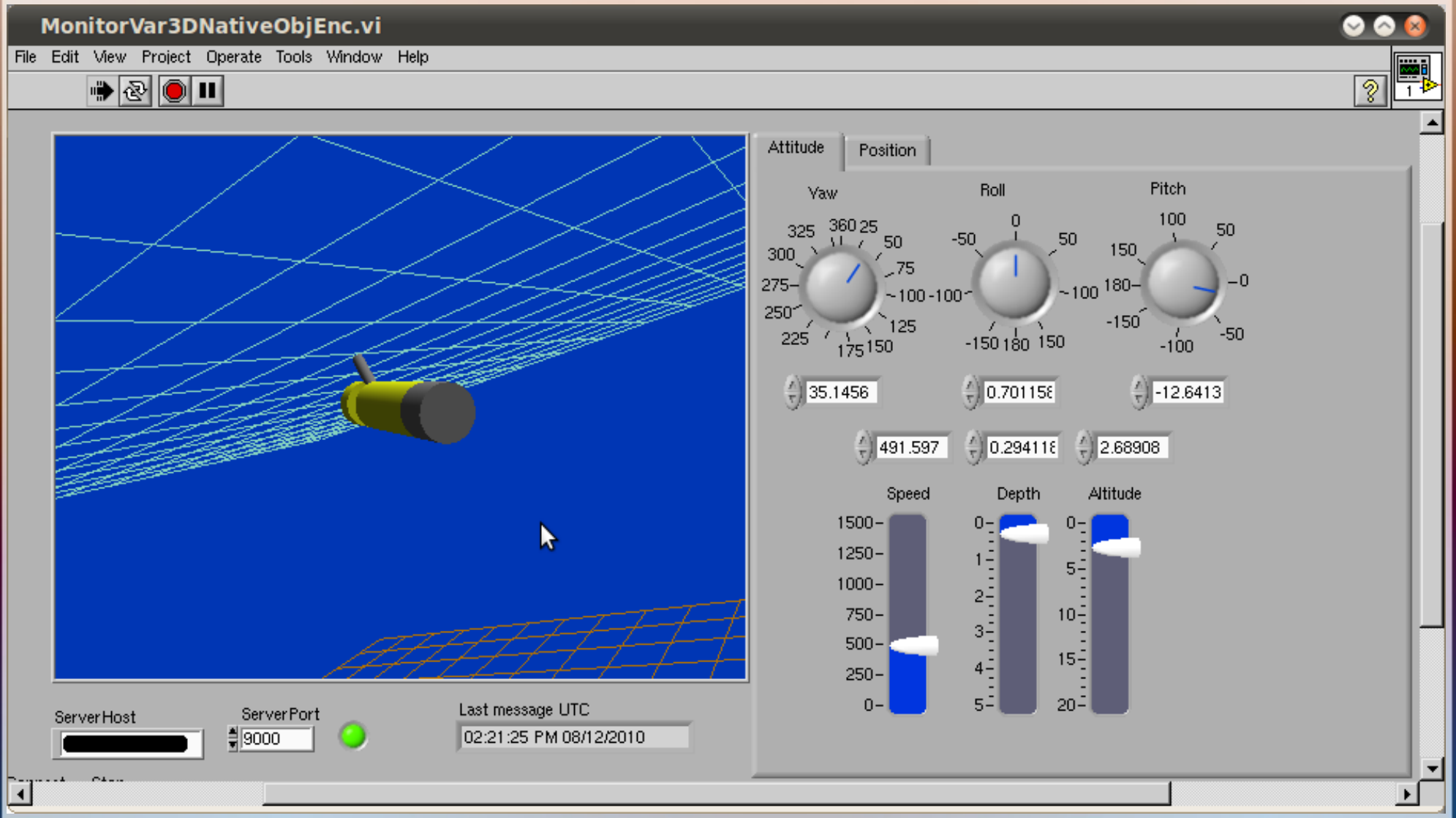
Maintaining separate implemenatation of interface

- Protocol went unchanged for long
- Implementation is confined to few functions

Files to monitor between versions:

- MOOSCommClient.cpp/.h
- MOOSCommObject.cpp/.h
- MOOSCommPkt.cpp/.h
- MOOSMsg.cpp/.h

Demo: LabVIEW-based vehicle attitude display



Acknowledgement

- This work was supported by ONR project #N00014-05-1-0632: Navy Force Protection Technology Assessment Project
- This work would have been impossible without the efforts of Paul Newman and all the contributors to MOOS development.