Hardware Emulation for Iterative EPICS Development

Aidan Boisvert
Oak Ridge National Labs
September 19, 2024

ACU NEXT Lab

NEXT
Nuclear Energy eXperimental Testing

Nuclear Energy eXperimental Testing

Intending to build a molten salt reactor on campus by 2026.

Operating under a sponsored research agreement with Natura Resources

ACU Science and Engineering Research Center

Molten Salt Test System (MSTS)

NEXT
Nuclear Energy eXperimental Testing

~200 Liter FLiNaK Salt Loop

Technological stepping stone for Natura's MSR-1.

- Test bed for scientific monitoring systems & control mechanisms for the reactor.

Built using many EPICS IOCs on distributed hardware.

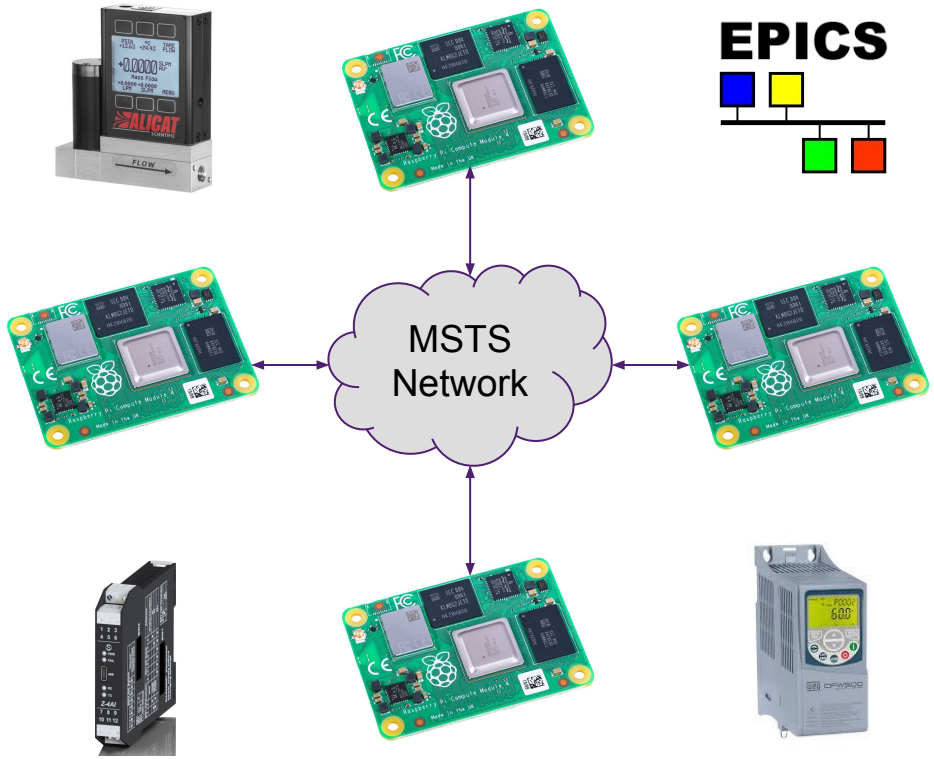
MSTS Drain Tank, prior to heater installation

Instrumentation & Controls

NEXT
Nuclear Energy eXperimental Testing

Leveraging only a few EPICS modules, we have been able to construct a very capable experimental system.

- asyn
  - modbus
  - streamdevice
- SNL Sequencer



EPICS based on Raspberry Pi CM4s managing subsystems of the MSTS.

MSTS Control and Instrumentation Systems

NEXT
Nuclear Energy eXperimental Testing

Major Subsystems of MSTS:

- Gas Control System
- Heat Control System
- Pump Control System
- Level Measurement System
- HF Gas Detection System

Development Reality

NEXT
Nuclear Energy eXperimental Testing

We are a small university without a graduate computer science program.

The majority of our interns are either:

- Electrical Engineering Undergraduates
- Computer Science Undergraduates

Generally Inexperienced with C/C++ and need to be trained in EPICS development

Ensure Code Functionality

NEXT
Nuclear Energy eXperimental Testing

When training students to work in the EPICS ecosystem, the most common software failures are:

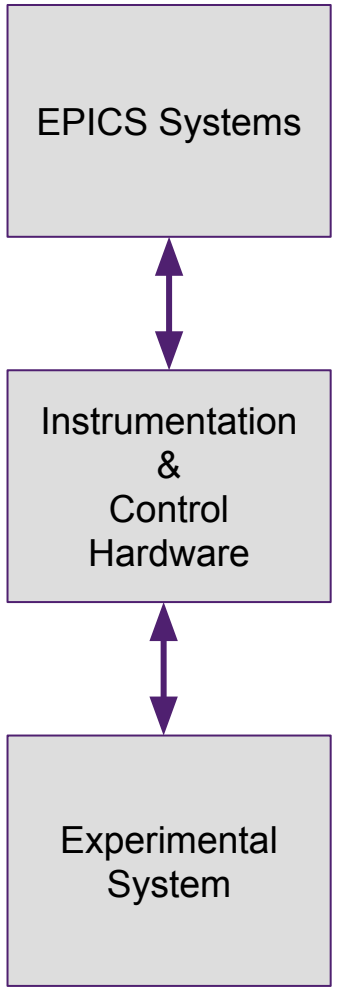
- Incorrect Modbus Configuration
- Incorrect Stream protocols
- Poorly conceived / constructed EPICS database relationships

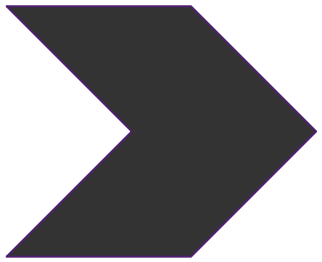
In order to quickly teach new developers and test new software, we needed to disconnect our development process from experimental hardware.

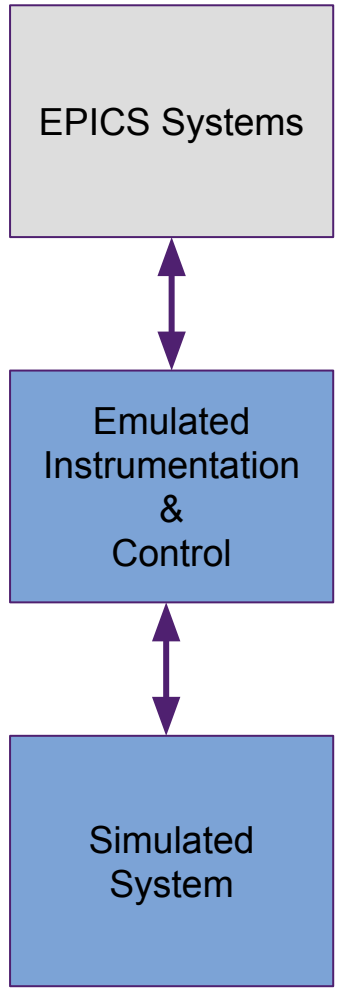
Approach

EPICS In Production (At NEXT)

- Distributed
- Communicates with physical hardware on experiment
- Untested code should be avoided to ensure reliability
- ARM Linux based environment
- Misconfigured modbus / stream commands could cause unintended controls behavior







EPICS In Testing

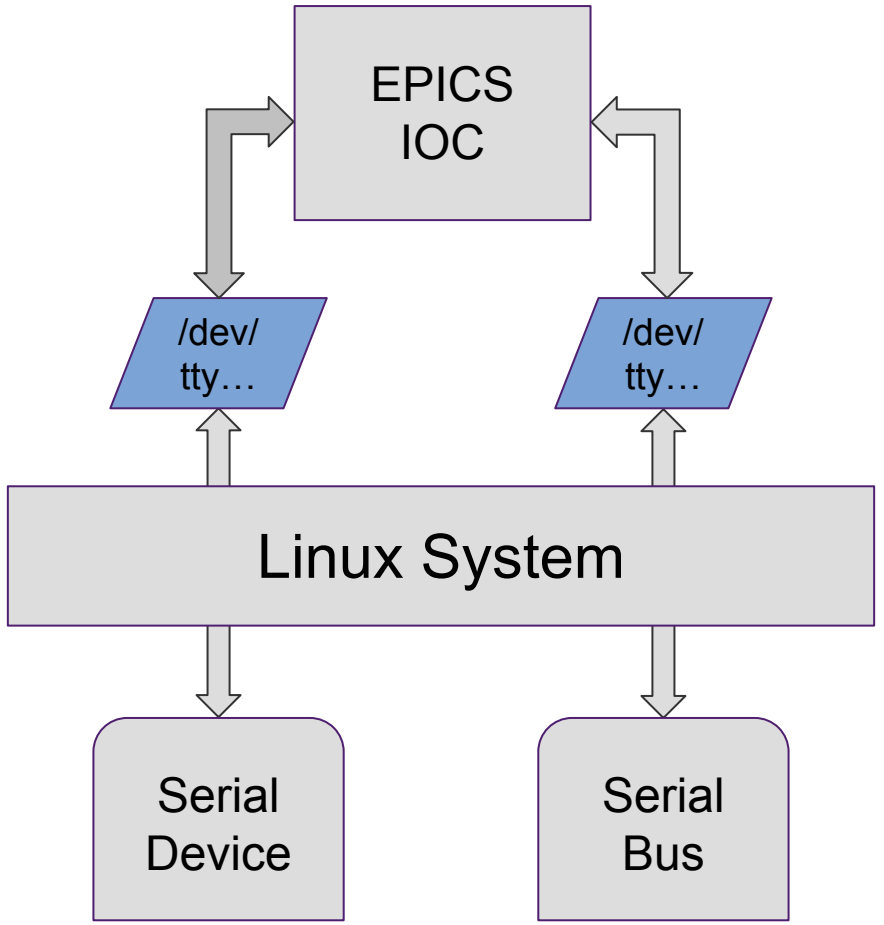
- Many IOCs operating on one host machine.
- Disconnected from physical hardware.
- Emulated I&C behavior
- x86 Linux environment
- Able to rapidly iterate and test modbus and stream communication.

Linux Serial Ports

NEXT Instrumentation and Control:

- Serial
- TCP/IP

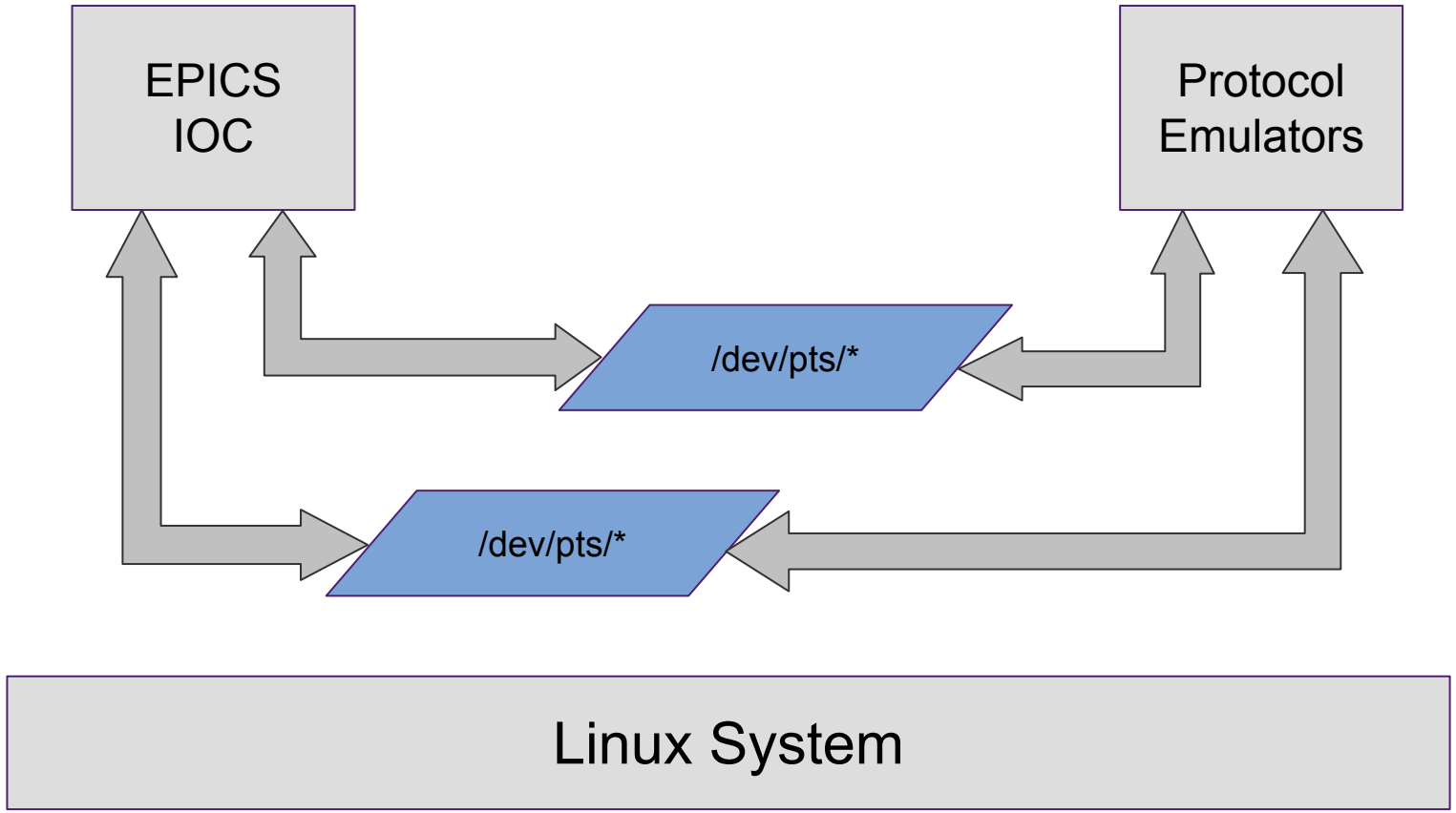
We have focused our hardware emulation efforts towards asyn serial and asyn IP.



Linux Serial File Properties:

- Bidirectional
- Buffered
- Configurable Bitrates
- Start/Stop bits
- Just another file

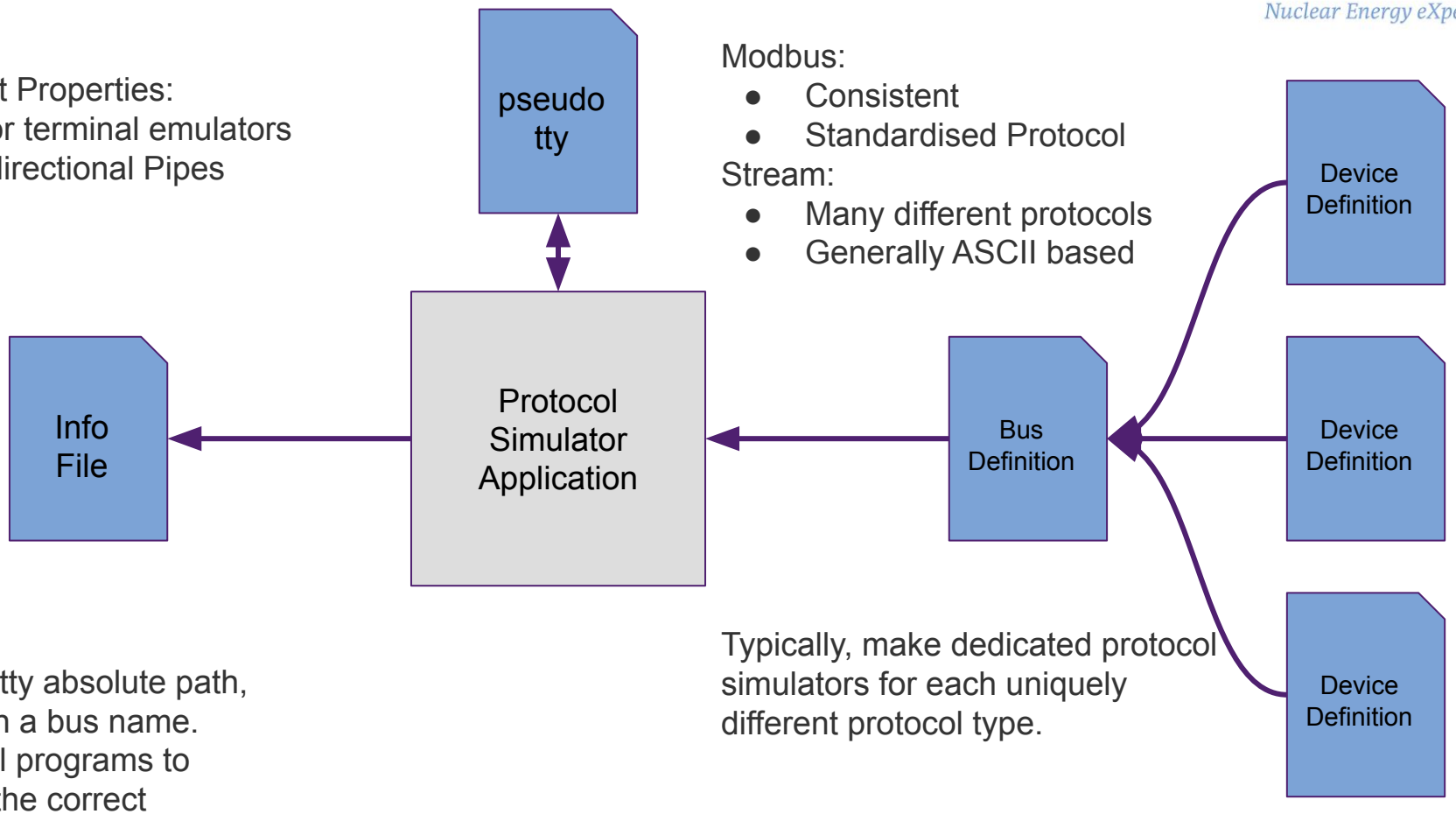
Emulated Serial Ports



Protocol Emulation

Nuclear Energy eXperimental Testing

Pseudo TTY Port Properties:
- Typically for terminal emulators
- Named Bidirectional Pipes



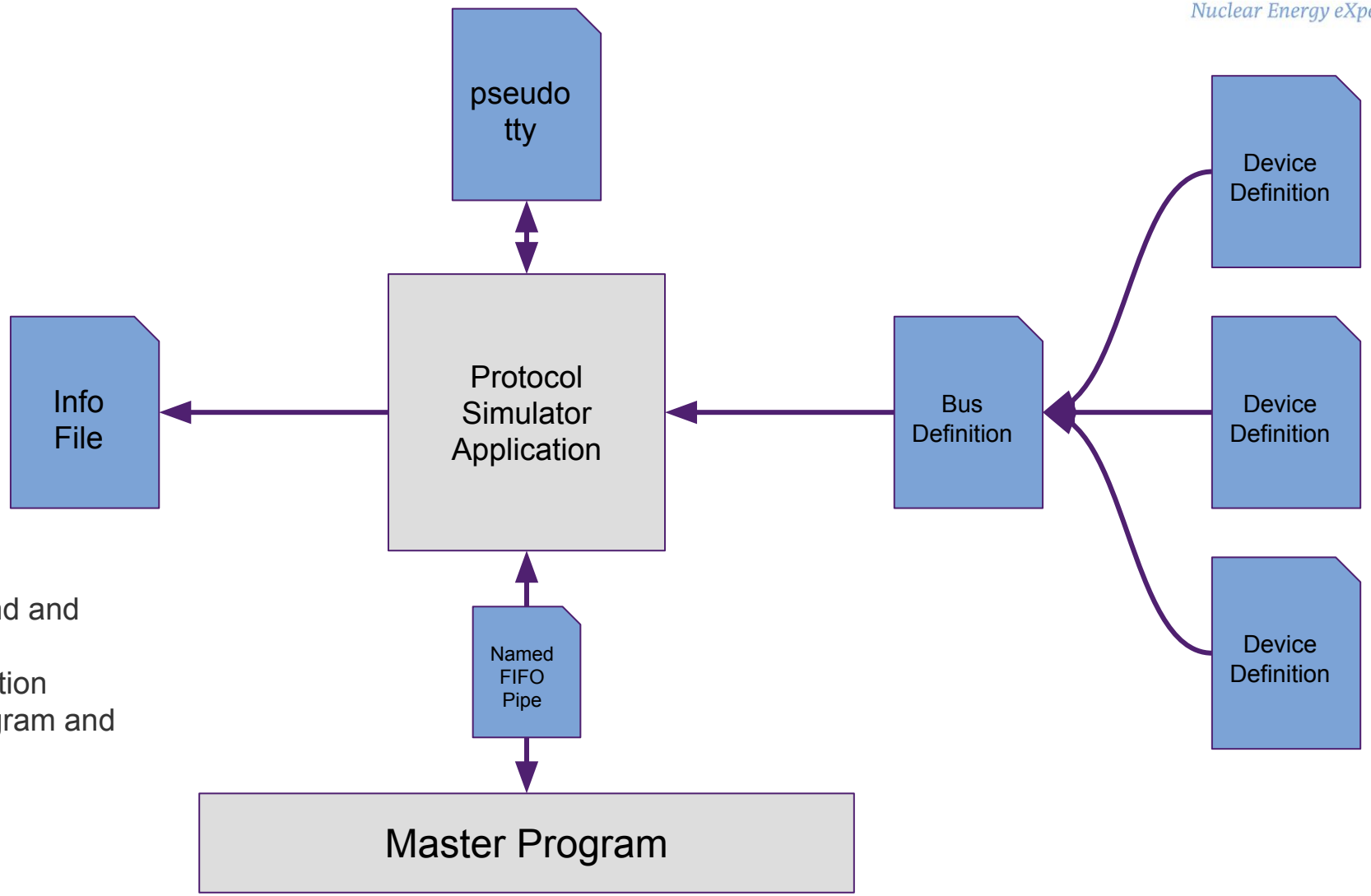
Modbus:
- Consistent
- Standardised Protocol

Stream:
- Many different protocols
- Generally ASCII based

Info File:
- PID & pseudo-tty absolute path, associated with a bus name.
- Allows external programs to easily identify the correct protocol simulator.

Typically, make dedicated protocol simulators for each uniquely different protocol type.

Protocol Emulation



Named FIFO Pipes:

- Two pipes for command and response.
- Facilitates communication between “master” program and the protocol simulator

IOC & Protocol Sim Integration

NEXT
Nuclear Energy eXperimental Testing

The only difference between IOCs running on real and emulated hardware using this method is the serial device file that the IOCs communicate with.

Our approach has utilized a separate “sim_st.cmd” file that runs the following commands:

- Copy st.cmd to a temporary cmd file
- Use the “sed” utility to replace serial file paths and asynIP configurations with the necessary pseudo terminal file
- execute the temporary cmd file.

IOC & Protocol Sim Integration

Nuclear Energy eXperimental Testing

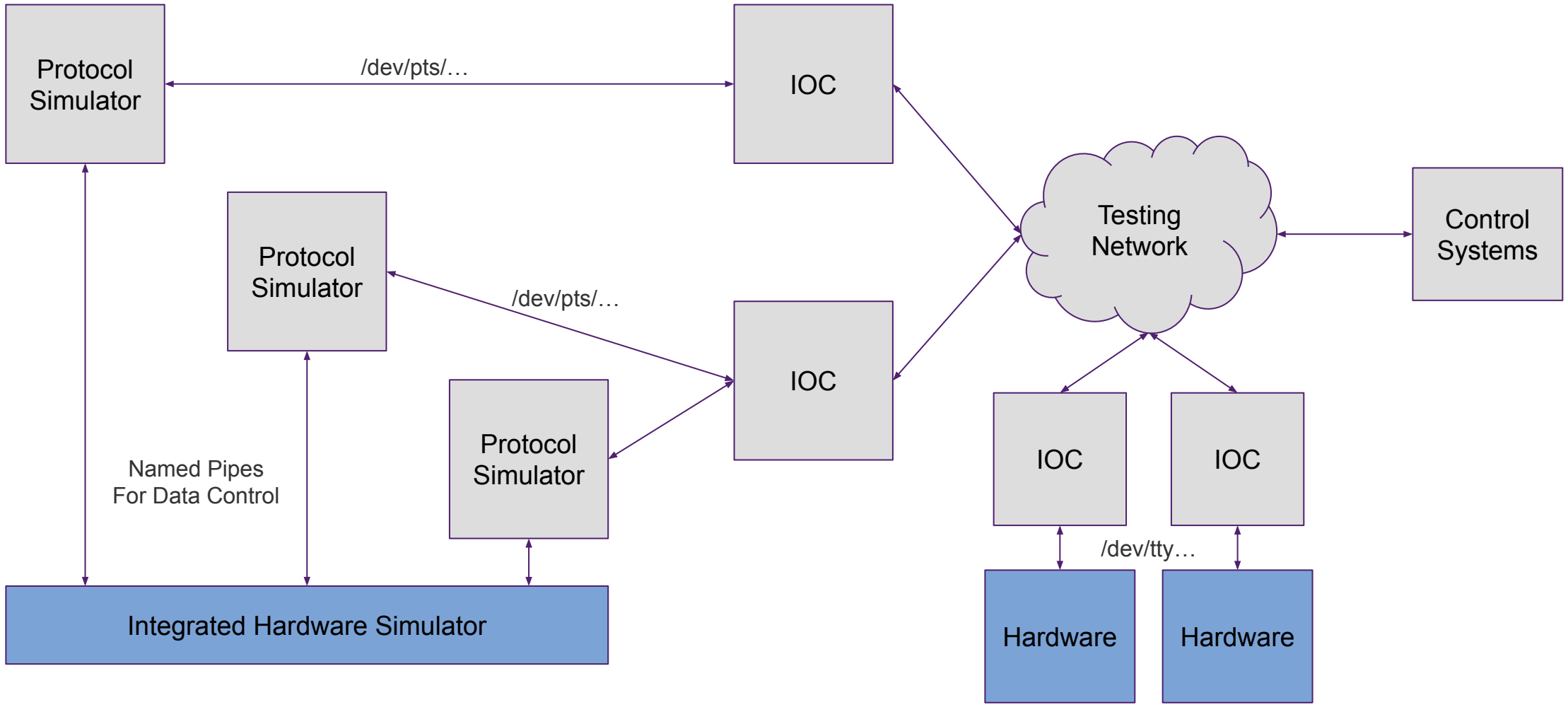
Typical Integration Example:

```
#!/bin/bash
# Copy st.cmd to temporary file
cp st.cmd temp_st.cmd
# Replace the serial file with the simulator’s file via a utility program
sed -i 's|, "/dev/ttyS1"|, "'"$(virtmodbusGetSerial DeviceID -s | tr -d '\n')"'"|' temp_st.cmd
# Execute the temporary file
./temp_st.cmd
```

AsynIP Integration Example:

```
#!/bin/bash
cp st.cmd temp_st.cmd
sed -i 's|drvAsynIPPortConfigure("deviceName","IP:PORT", 0, 0, 0)|drvAsynSerialPortConfigure("deviceName", "/dev/REPLACEME",0,0,0)|' temp_st.cmd
sed -i 's|"/dev/REPLACEME"|"'"$(virtmodbusGetSerial DeviceID -s | tr -d '\n')"'"|' temp_st.cmd
sed -i 's|modbusInterposeConfig("deviceName", 0|modbusInterposeConfig("deviceName", 1|' temp_st.cmd
./temp_st.cmd
rm temp_st.cmd
```

Emulation Integration



Method For Use

NEXT
Nuclear Energy eXperimental Testing

Application towards ACU's Molten Salt Loop:

22 Separate Serial & Network Busses for Instrumentation & Control:

- 8 Modbus Busses
- 14 Stream Devices / Busses

We built a test framework using the tools mentioned previously which could successfully model the I&C system of the MSTS using a node relationship based system.

MSTS With Heating System Installed

Results

NEXT
Nuclear Energy eXperimental Testing

We successfully utilized EPICS at ACU NEXT Lab to build and operate our Molten Salt Test System.

During Mid-August of this year, we ran an eight hour commissioning run of the loop at temperature.

Our EPICS IOCs were developed with heavy usage of the hardware emulation system.

Questions?

Acknowledgements

Special Thanks to: Dale Cox, Austin Geisert

Natura Resources and ACU NEXT Lab

EPICS Contributors and Argonne National Lab

Linux Kernel Contributors