

## Introduction

The SSRL beamline is controlled by the DCS protocol or distributed control system, which consists of three parts: The Blu-Ice GUI, the DCSS or the distributed control system server, and the DHS or distributed hardware servers.

Some of the drawbacks of Blu-Ice is that the GUI scripted from Tcl/Tk, which is a very old and rarely used language. DCSS/DHS are programmed in C++, but it is very complex and hard to understand, thus making maintenance quite difficult. The purpose of this project is to integrate as much of Blu-Ice into EPICS as we can, with the eventual goal of replacing all of DCSS/DHS with EPICS and replacing the Blu-Ice GUI with one written in Python.

## Research

Initially, I spent time learning how EPICS worked. EPICS uses IOC's (Input Output Controllers) which interface to equipment and devices. The IOC runs on a database of PVs (or records), process variables that represent instruments and their controllable aspects. Clients can connect to the IOC via Channel Access protocol and then interact with the hardware.

Once we were able to get access to the Blu-Ice source code, I started parsing through it to find the best way to integrate EPICS into Blu-Ice, as well as trying to understand how Blu-Ice worked.

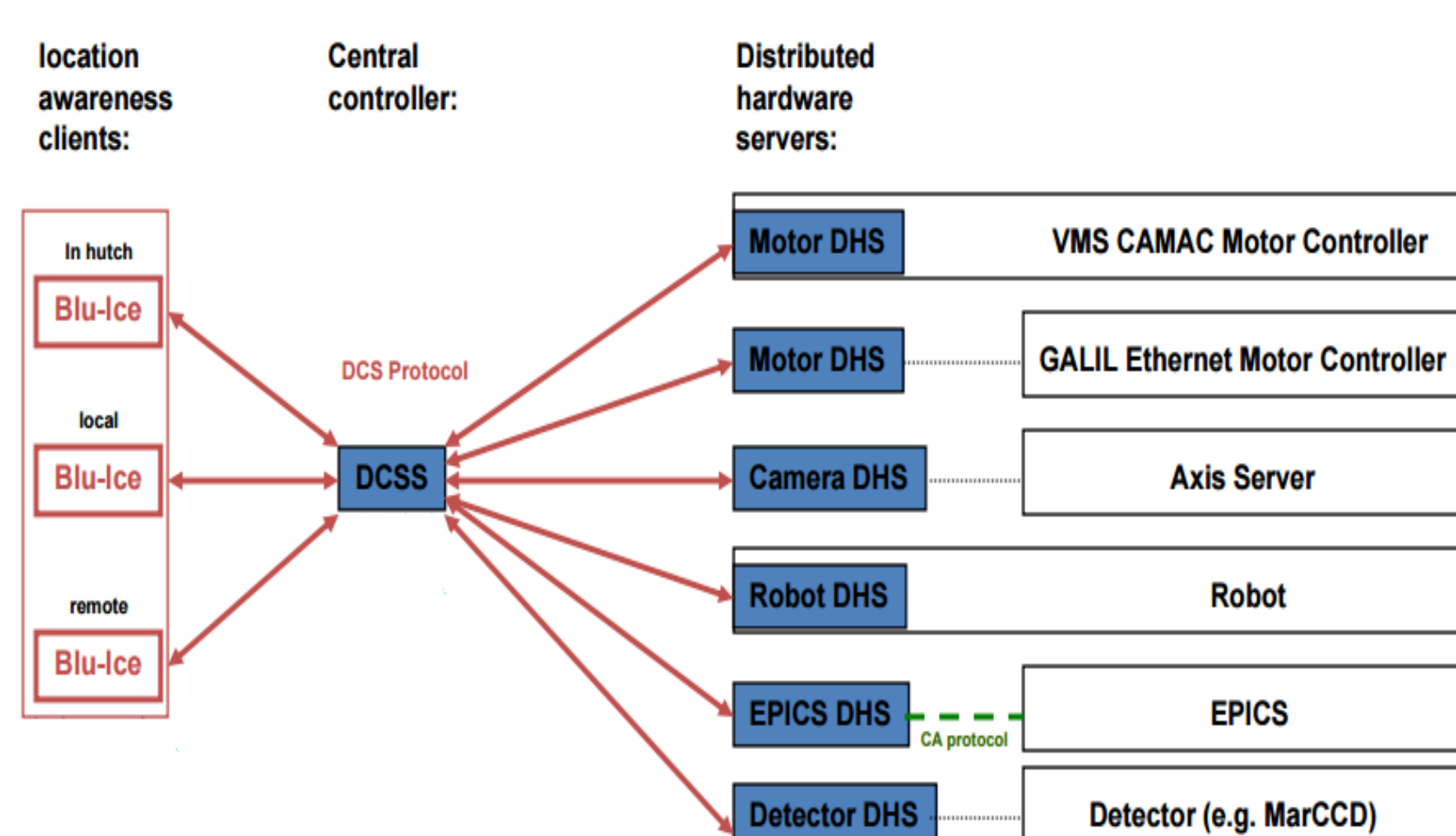


Fig 1. DCS architecture. Blu-Ice GUIs connect to the DCSS, which sends instructions to the DHS.

The DCS server, or DCSS communicates with the Blu-Ice GUI and DHS. The GUI sends instructions to DCSS, which tells the DHS what to do. The DHS has low-level control over physical devices and simply executes instructions from the DCSS.

We hoped to initially replace the stoh and htos messages (Server to Hardware and Hardware to Server messages) that the DCSS sends/receives from the DHS to control hardware by redirecting it to modify/change EPICS PVs via Channel Access. This would be simplest and easiest way to connect DCSS to EPICS.

We found a gateway that sent DCS messages into EPICS, and were hoping to integrate that into Blu-Ice. The gateway was already written and allowed for stoh and htos messages to be sent to EPICS. The trick would be properly integrating it into Blu-Ice. The DCSS is a very complex system as it manages several GUI clients as well as several hardware clients. The DCSS also allows custom scripts to be run and executed, which also adds another dimension of complexity. The most logical way to integrate EPICS would be to create a single gateway that parses and executes all messages being sent to the server (gtos or GUI to server). However, upon observation of the DCSS structure, this became unfeasible as each GUI client has its own message handler, meaning that a separate gateway would have to be created for each client.

## New Discovery

As I was researching information on Blu-Ice, I stumbled upon a project that a team from Argonne National Laboratory did (JBluIce), which essentially removed DCSS/DCS from Blu-ICE and replaced it with EPICS. They also replaced the Tcl/Tk GUI with one written in Java. This was a major discovery as it basically encompassed our project goals (Integrating EPICS into Blu-Ice and replacing the GUI with one written in a more modern programming language).

Currently, we are working on launching the JBluIce and testing it. There are a lot of missing packages that need to be installed and the configuration of JBluIce will have to be changed to suit the SSRL beamlines. We are hoping that we can deploy JBluIce this summer.

I'm also in the process learning Python as we hope to replace the Java GUI with one written in Python.

## Conclusions

We wanted to integrate EPICS into Blu-Ice to make the source code more readable and maintainable.

This is still a work in progress. There already exists a viable solution for integrating Blu-Ice with EPICS (JBluIce). We are currently working on launching and testing it.

Eventually, we hope to create a new GUI written in Python.

## Acknowledgments

Use of the Linac Coherent Light Source (LCLS), SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515.