High-Level Application Development and Production Infrastructure at TRIUMF

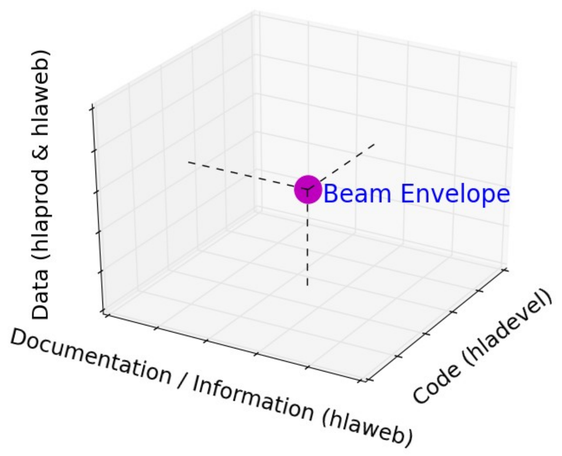
Evgeniy Tikhomolov, Y. Bylinskiy, A.C. Morton, T. Planche, T. Tateyama, J. Lee., P. Jung,*
TRIUMF Canada's National Laboratory for Particle and Nuclear Physics

Abstract
TRIUMF users and operators use a number of high- level applications (HLAs) written in different languages, having rather complicated graphics user interfaces, to carry out tasks related to delivering ion beams with required characteristics and to process data from TRIUMF's EPICS-based and legacy cyclotron control systems. Some applications have been developed by the EPICS community, and some at TRIUMF. These applications run on different production computers and are developed on different machines. This model no longer satisfies the TRIUMF's needs because of the growing number of applications, the long times required for data processing on current machines, the lack of real-time visualization of beam properties and so on. We present recently implemented infrastructure to solve these tasks, which contains three powerful servers. The servers run web applications and have tools to run calculation-intensive stand-alone applications remotely. Data fromTRIUMF EPICS-based control systems are taken via local Soft IOCs. The documentation is maintained by using developed content management system. The installed system is working reliably and has room for further expansion.

Motivations

TRIUMF doesn't have a dedicated group of software developers tasked with immediate response to issues that arise during day-to-day beam delivery. Such issues are resolved by operators and physicists themselves. Thus, a flexible and simple ("user-friendly") software development environment was the main request when it was decided to set up High-Level Application (HLA) development and production infrastructure. At the same time, developers come with different experience, backgrounds and their own favorite tools for development. Thus, the HLA environment should provide some "default set" of tools which are rather common and compatible with other widely-used tools.

The projects that are under development at TRIUMF usually have three components: the code itself (in a number of programming languages), documentation and information, and input/output data. Thus, the common project can be represented as a point in a "three-dimensional space" where each component defines an axis:

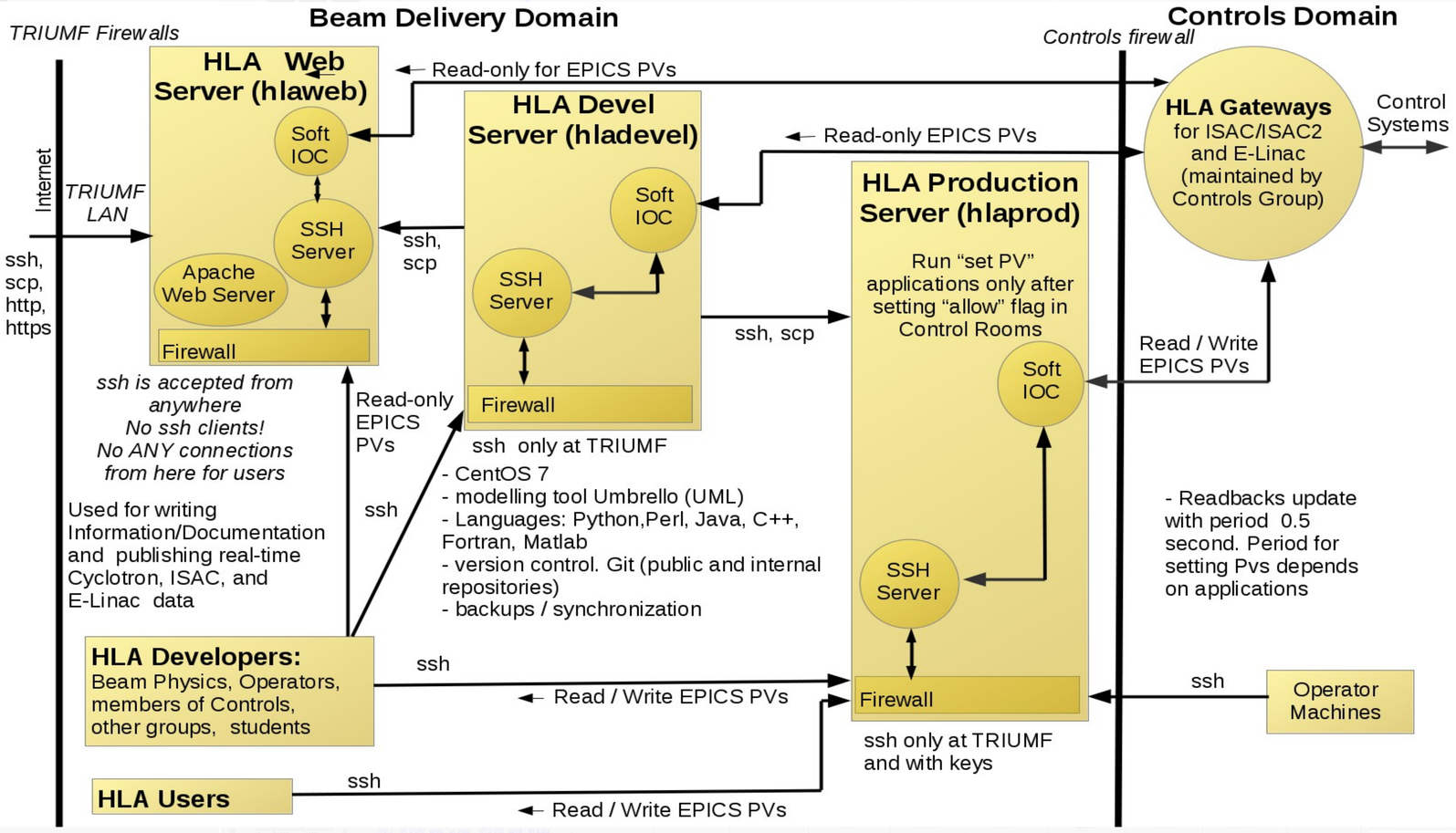


Any information and documentation about the project is kept on the HLA web server hlaweb. To write information on line an in-house content management system (CMS) using TinyMCE JavaScript package (https://www.tinymce.com/) was developed and set up:

Users Development Environment

Users can start light-weight remote desktops Xfce (https://www.xfce.org/). Graphics-intensive applications (like Matlab (http://www.mathworks.com/products/matlab/), Opera (http://operafea.com/)) which run very slowly via X-forwarding are started in X2Go (http://wiki.x2go.org/doku.php) session in the mode "Single Application":

Development (hladevel) and production servers (hlaprod and hlaweb) have absolutely identical setups for directories and software. The differences between the servers are defined only by their different roles:



Interface between HLAs and EPICS is done in two steps: HLA servers run local EPICS Soft IOCs which communicate with EPICS via gateways. Then HLAs get and set data from or in local Soft IOCs. Such a two-level design provides flexibility and complete decoupling from TRIUMF's Controls Group. Setting values for control system devices is allowed only from the production server hlaprod and only after the operators in Control Rooms set an "allow" flag with a time-out of 1 hour. The HLA development server hladevel and production server hlaprod are accessible only from the TRIUMF network. The web server hlaweb is accessible via the Internet and users can ssh to their accounts.

HLA Development Flow

UML: Umbrello (https://umbrello.kde.org/)
Languages: Python, Perl, Java, C++, Fortran, Matlab. Codeblocks (http://codeblocks.org/) is suggested as an IDE for code development
Web applications, client side: JavaScript, jQuery (http://learn.jquery.com/), HighCharts (http://www.highcharts.com/)
For version control Git (https://git-scm.com/) is used. Public and private remote repositories are located on hlaweb and hladevel, respectively. For projects which are expected to involve large number of developers Github (https://github.com) may also be used
Rsync and Backups. The development server hladevel is used for the synchronization (using rsync) of data and user's scratch directories from hlaweb every 4 hours. Data directories from hlaprod are also synchronized to hladevel. Nightly backups are done only on hladevel both on local USB drives (for fast restoration) and to a remote location (using TRIUMF's Amanda system (http://www.amanda.org/)

CONCLUSION

The created HLA infrastructure allows software developers to develop applications both by using the default set of tools installed on HLA servers and remotely on the computers of individual users. The sequential versions of projects are saved on HLA servers and are easily accessible by developers. Documentation for projects can be written on-line and is maintained in a very simple way using a dedicated HLA CMS. Several methods are used to run HLAs (which work with devices in TRIUMF control systems) in a secure manner. If the load on the servers will be increased the expansion of existing infrastructures is quite easy and won't create significant additional time for the maintenance.

Welcome to TRIUMF High-Level Applications web site: http://hlaweb.triumf.ca

Presented at:

PCaPAC 2016
11th International Workshop on Personal Computers and Particle Accelerator Controls
October 25th – 28th, 2016
Campinas, Brazil

*Present address: Facility for Rare Isotope Beams Michigan State University 640 South Shaw Lane East Lansing, MI 48824