Photon Science Controls: A Flexible and Distributed LabVIEW™ Framework for Laser Systems

ICALEPCS 2021: MOBL05

Brock Davis
Control Systems Engineer

11 October 2021

LLNL-PRES-827859
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

LabVIEW™ is an attractive option for mid-scale distributed control systems – with caveats

- NI provides an extensive hardware ecosystem
  - cDAQ for non-deterministic applications
  - cRIO (Linux RT) for soft real-time
  - cRIO FPGA for hard real-time

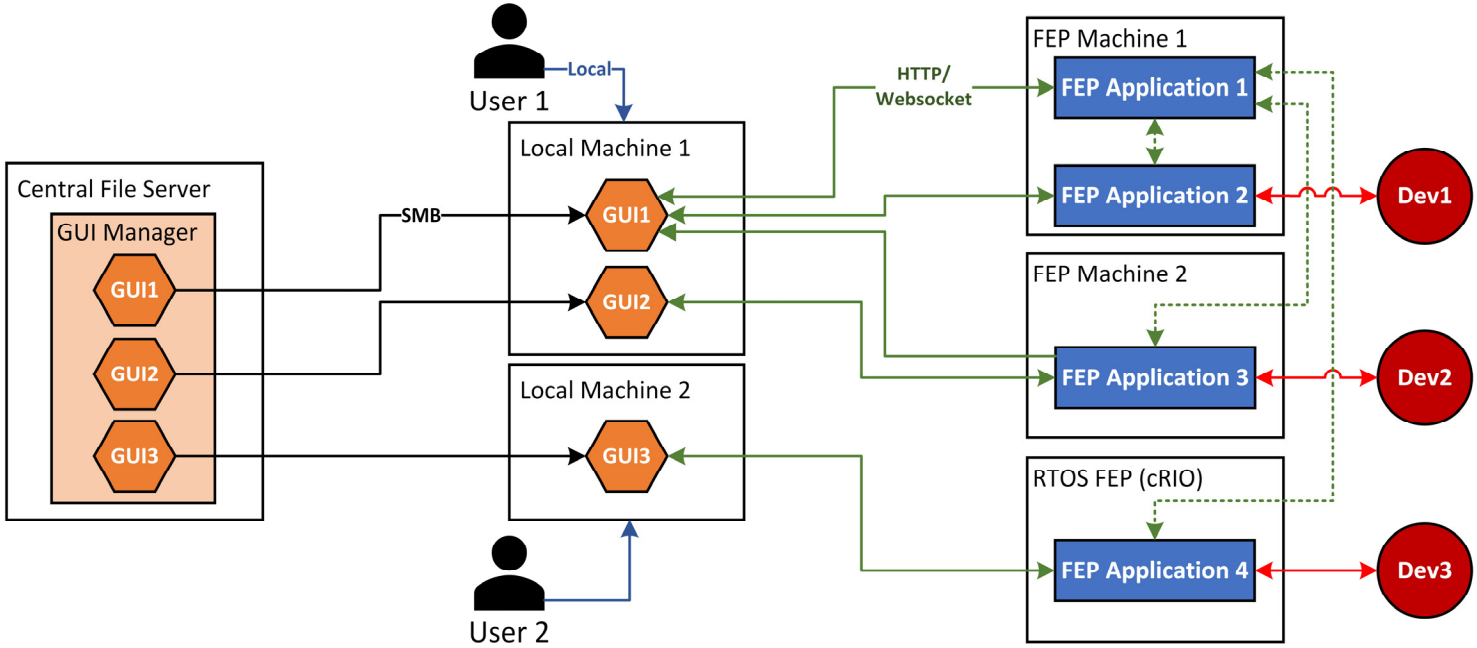
https://www.ni.com/en-us/shop/hardware/products/compactrio-controller.html

- LabVIEW™ software has a shallow learning curve and an intuitive “dataflow” order of execution
- Many common data acquisition and processing workflows are well-documented and simple to implement
- Simple control systems can be stood up quickly with LabVIEW™, but often suffer from a lack of scalability and extensibility
- Creating flexible LabVIEW™ software requires significant skill and time investment

The Photon Science Controls (PSC) framework fills the niche between small- and large-scale distributed control systems

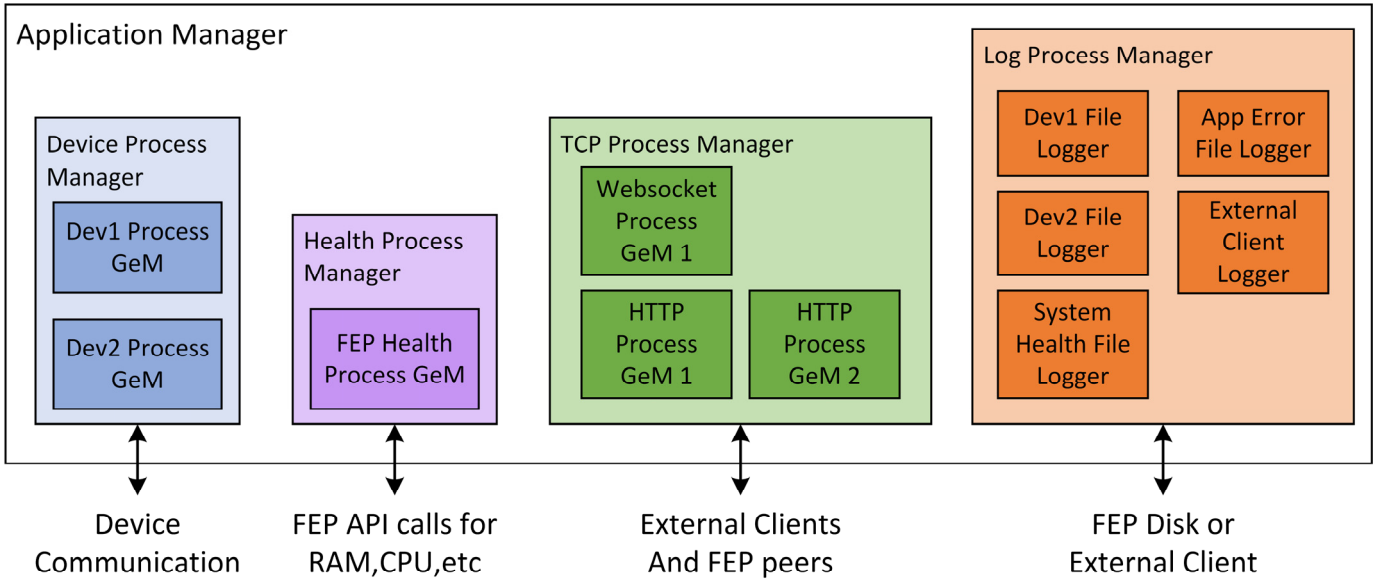
- Small-scale systems (single FEP) can use LabVIEW with no concerns for extensibility/scalability
- Large-scale systems (facility level) have funding, staffing, and timelines to invest in custom software and hardware to exactly meet needs
- Mid-scale systems occupy a challenging space where scalability is a necessity, but resource constraints often prevent development of custom software and hardware
- The PSC framework bridges this gap by providing an easy-to-use LabVIEW™ architecture that works across the determinism spectrum
- Native object orientation allows for scaling and extension through inheritance and polymorphism

PSC systems are peer-to-peer over LAN



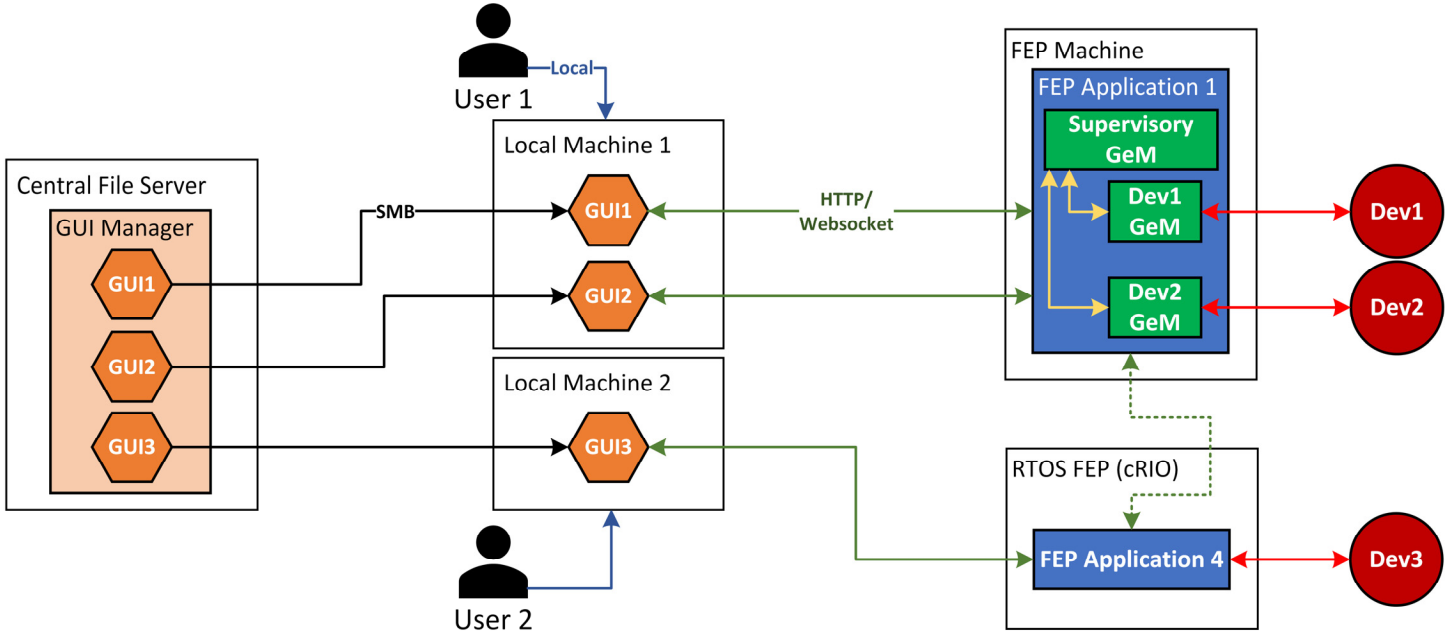
Peer-to-peer architecture allows for complete data access for any peer or client. Users can control the system from any machine on the LAN.

PSC applications consist of asynchronous functional processes known as Generic Messengers (GeMs)



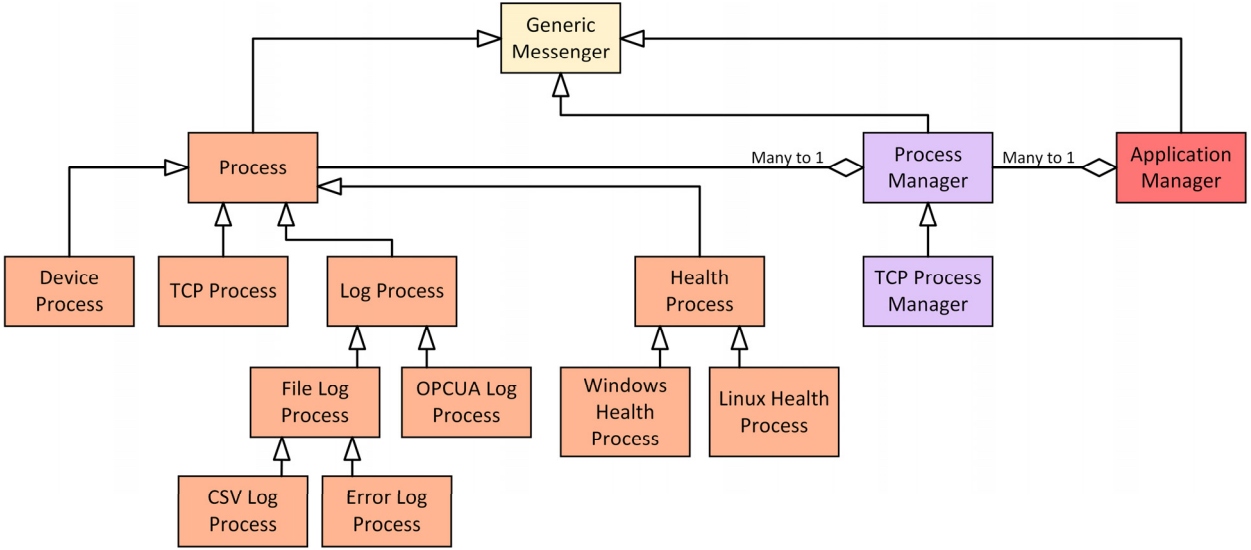
Each GeM is self-contained, asynchronous, and functionally independent. GeM to GeM communication is handled through LabVIEW queues.

GeMs and functionality can be distributed across the system to fulfill needs



Intra-Application communication between GeMs is analogous to inter-application communication between FEP applications

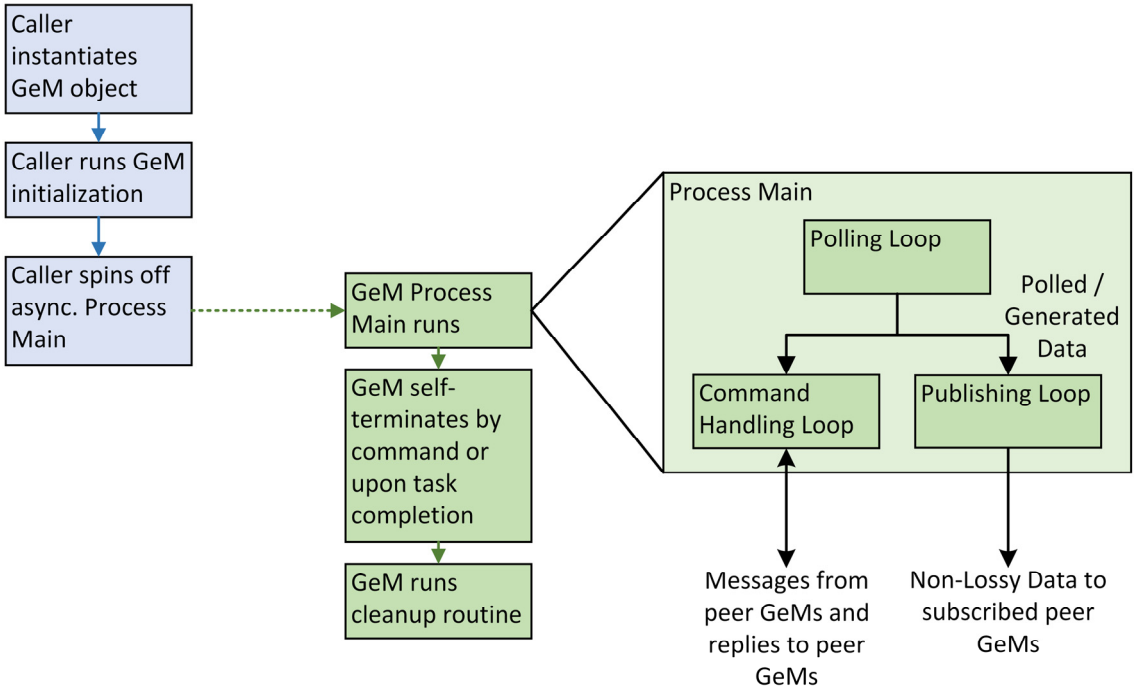
Inheritance and Polymorphism ensure GeM conformity



Common GeM Functions
Data Polling/Generation
Command Handling
Lossy and Non-Lossy Data Streams
Thresholding and Alarming
Internal Error Handling

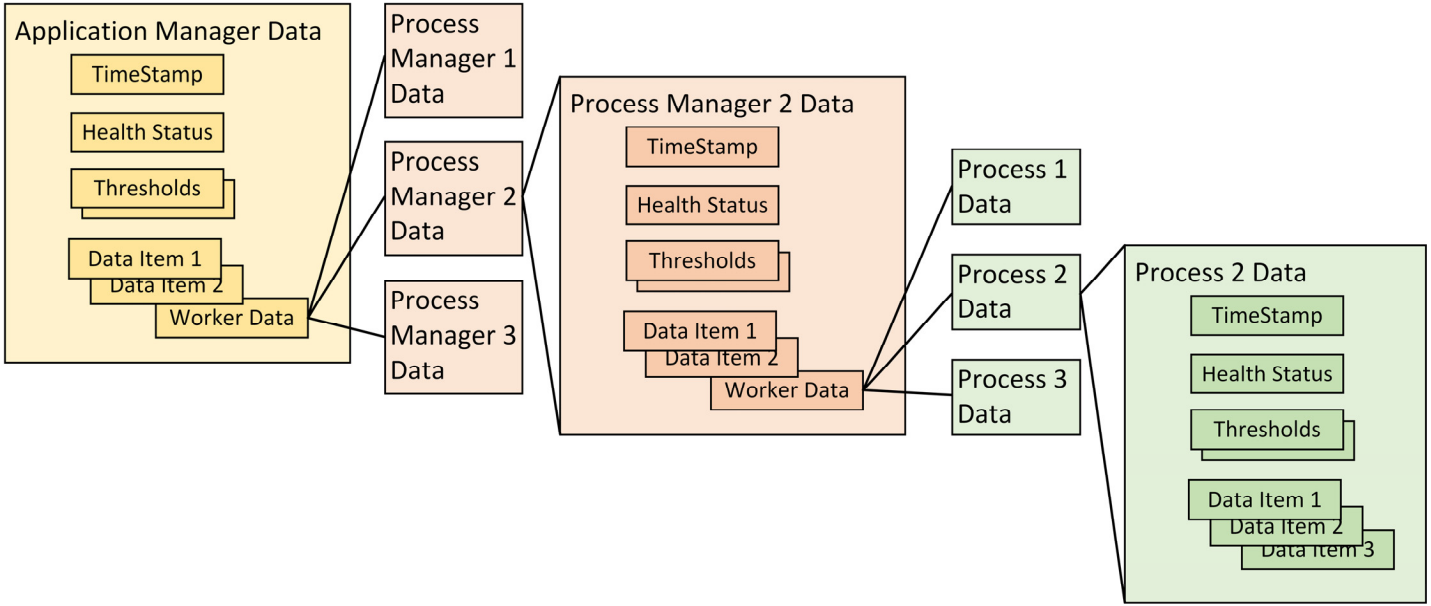
All GeMs share high level functionality inherited from the GeM ancestor class. Each descendant GeM builds upon this functionality with more specific capabilities.

Common Process Main provides consistent behavior across all GeMs



GeMs all have the same interface with peers of the same application. Messaging queues allow for information to be passed between GeMs.

Application data is recursively packaged from worker to manager



The nested data structure allows for full application status to be queried with a single call to the Application Manager GeM.

The PSC framework provides a flexible solution for mid-scale distributed control systems

- Scalability to support distributed systems with a variety of FEPs
- Support for hardware across the spectrum of determinism
- Extensibility for additional functionality, new communication protocols, and new device classes
- Flexible distribution scheme to fulfill requirements by distributing GeM functionality across applications
- Simple workflows for novice developers with sufficient depth to keep advanced developers engaged


Lawrence Livermore
National Laboratory