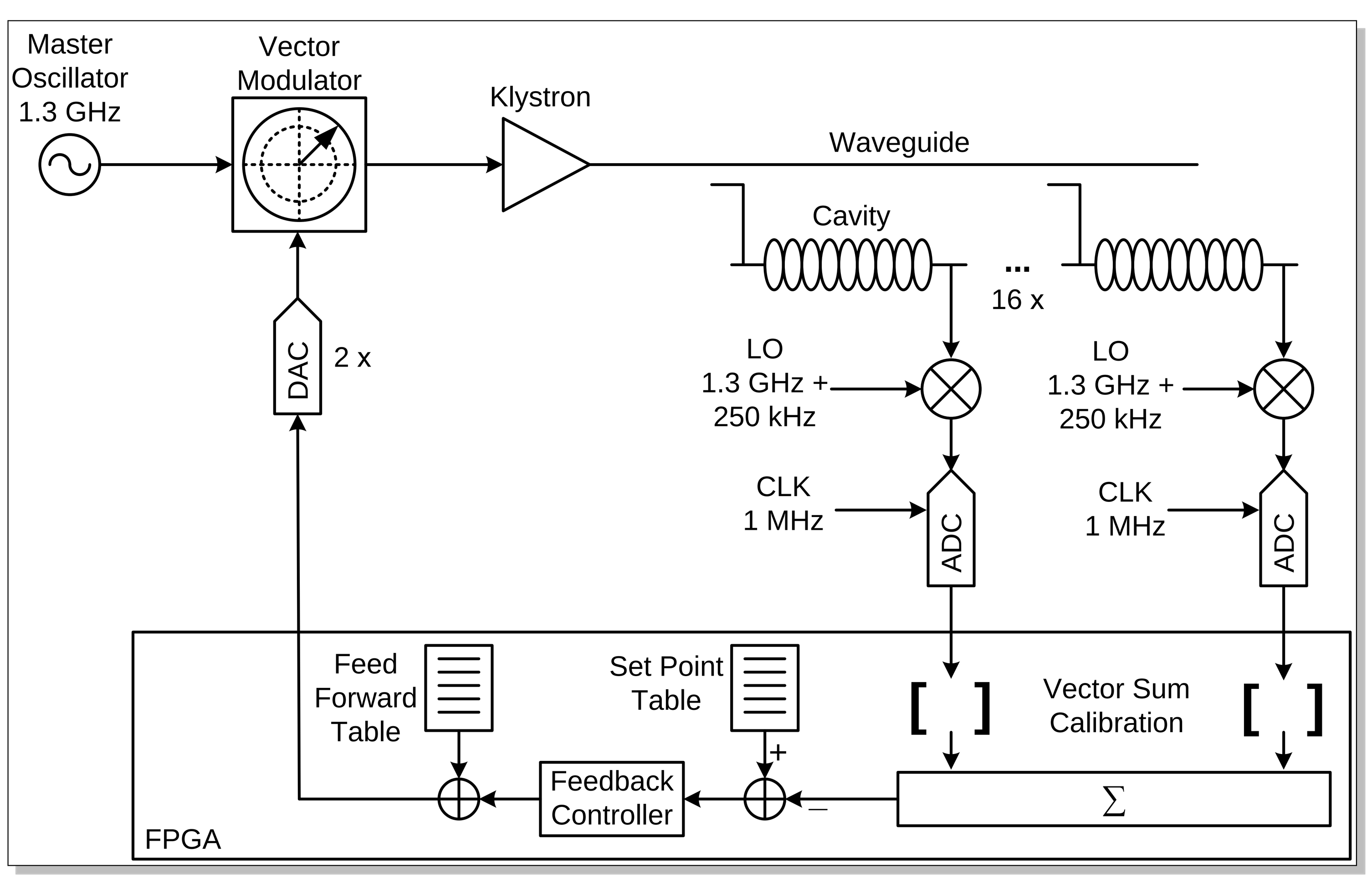
SLAC NATIONAL ACCELERATOR LABORATORY

Architecture Design of the Application Software for the Low-Level RF Control System of the Free-Electron Laser at Hamburg

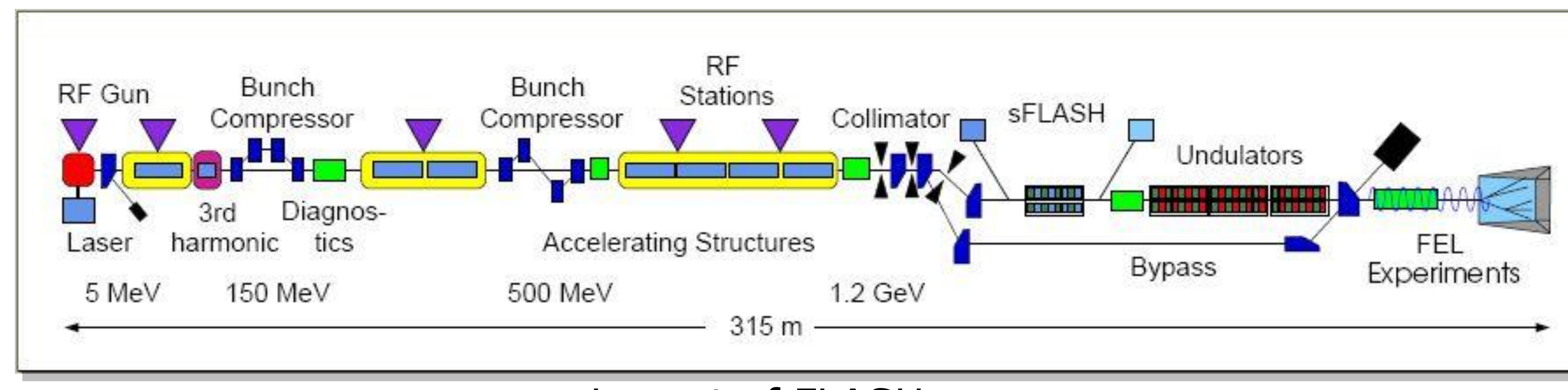
Z. Geng, SLAC, Menlo Park, California, U.S.A.
V. Ayvazyan, DESY, Hamburg, Germany
S. Simrock, ITER Organization, St. Paul lez Durance, France

Introduction

The superconducting linear accelerator of the Free-Electron Laser at Hamburg (FLASH) provides high performance electron beams to the lasing system to generate synchrotron radiation to various users. The Low-Level RF (LLRF) system is used to maintain the beam stabilities by stabilizing the RF field in the superconducting cavities with feedback and feed forward algorithms. The LLRF applications are sets of software to perform RF system model identification, control parameters optimization, exception detection and handling, so as to improve the precision, robustness and operability of the LLRF system. In order to implement the LLRF applications in the hardware with multiple distributed processors, an optimized architecture of the software is required for good understandability, maintainability and extendibility. This paper presents the design of the LLRF application software architecture based on the software engineering approach for FLASH.



FLASH RF station including LLRF



Layout of FLASH

LLRF Applications

LLRF applications are sets of software to facilitate the LLRF control loop for better precision, robustness and availability. The main goals of the LLRF applications include :

- Improve the RF field stabilities by optimizing the parameters of the RF field controller.
- Improve the robustness and availability of the LLRF system by system diagnostics, exception detection and handling.
- Support automation for easy operation.

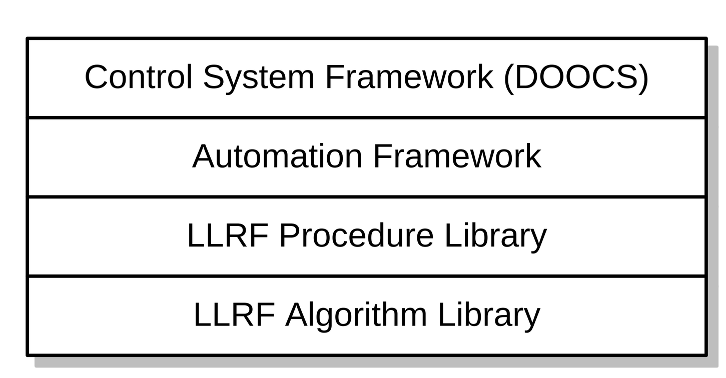
Table 1: Examples of LLRF applications

Category	Applications
Diagnostics	- Beam phase and beam current measurement - Loop phase and loop gain measurement - Cavity loaded quality factor and detuning measurement
Signal Calibration	- Vector sum calibration - Cavity gradient and phase calibration - RF gun field calibration
System Identification	- RF system dynamic model identification - Cavity model identification - Klystron non-linearity characterization
Control Parameters Optimization	- Adaptive feed forward - Feedback gain scheduling
Exception Detection	- Cavity quench detection - Cavity operational limit exceeded detection - RF system components failure detection
Exception Handling	- Recovery from cavity quench - Adjust cavity gradient

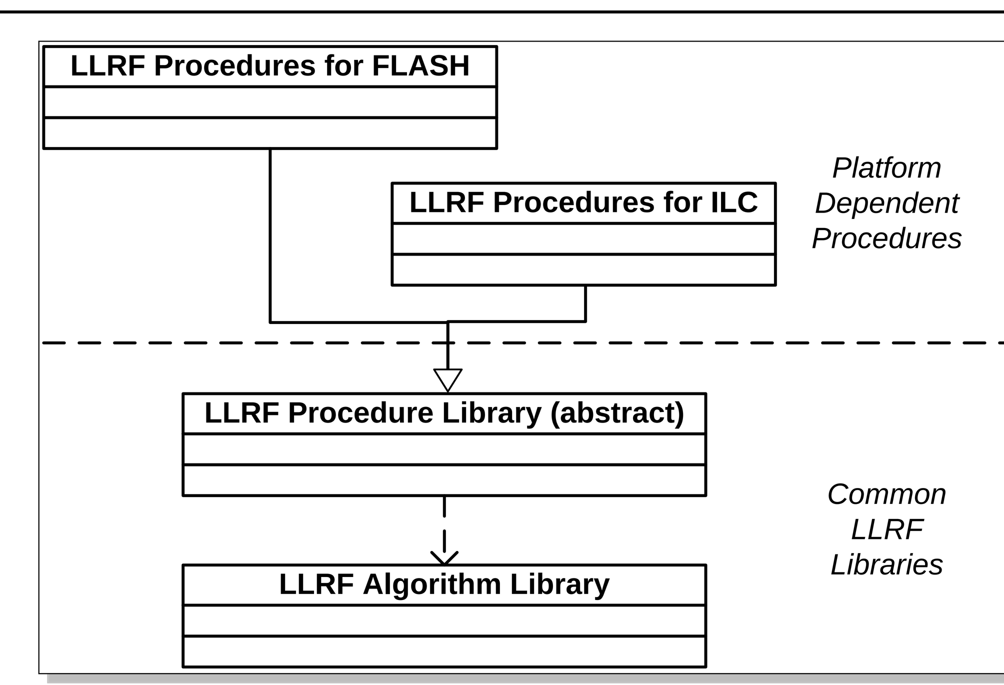
Software Architecture Design

Several key points of the software architecture for LLRF applications:

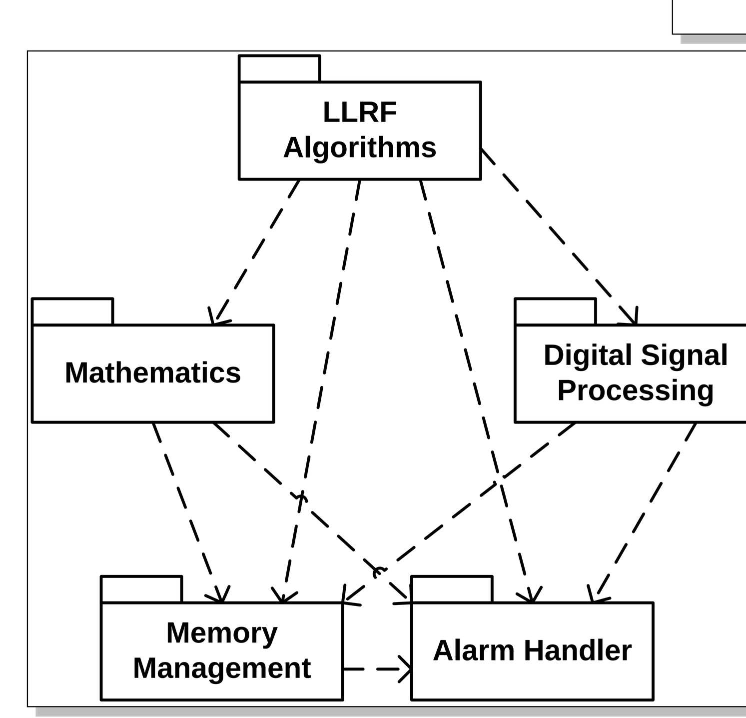
- Layered architecture. Interaction only happens between the adjacent layers.
- Define libraries (LLRF Procedure Library and LLRF Algorithm Library) to capture LLRF domain knowledge for high reusability.
- The LLRF Procedure Library is designed in C++ language for better extendibility. It realizes the procedures for the tasks (use cases) of the LLRF applications.
- The LLRF Algorithm Library is designed in C language for better performance in math calculation and lower level processor (MCU or DSP) compatibility.
- The LLRF Algorithm Library is broken down based on the domains. The idea is to pack the code for a specified domain (like digital signal processing) so that the whole package can be reused in other applications.
- For each domain package, the codes are divided into three parts: "Available Interface", "Library Body" and "Required Interface". The interactions between domain packages are only through the interfaces.



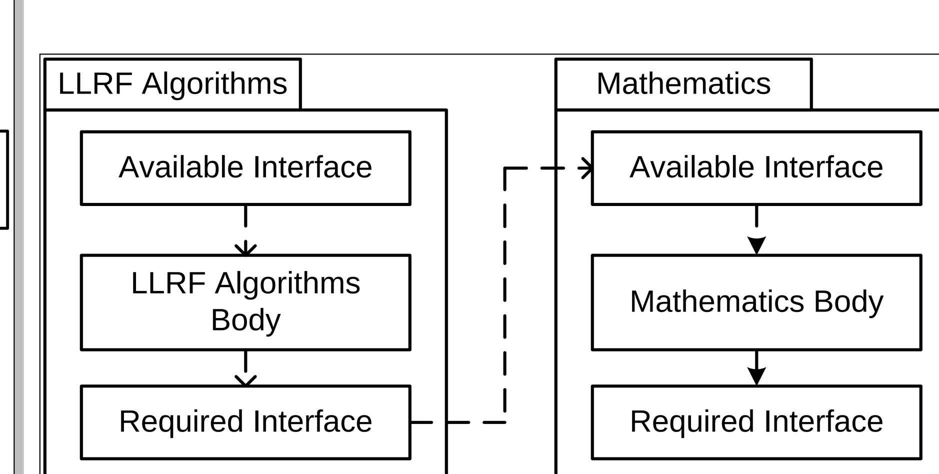
Layers of the architecture for LLRF Applications



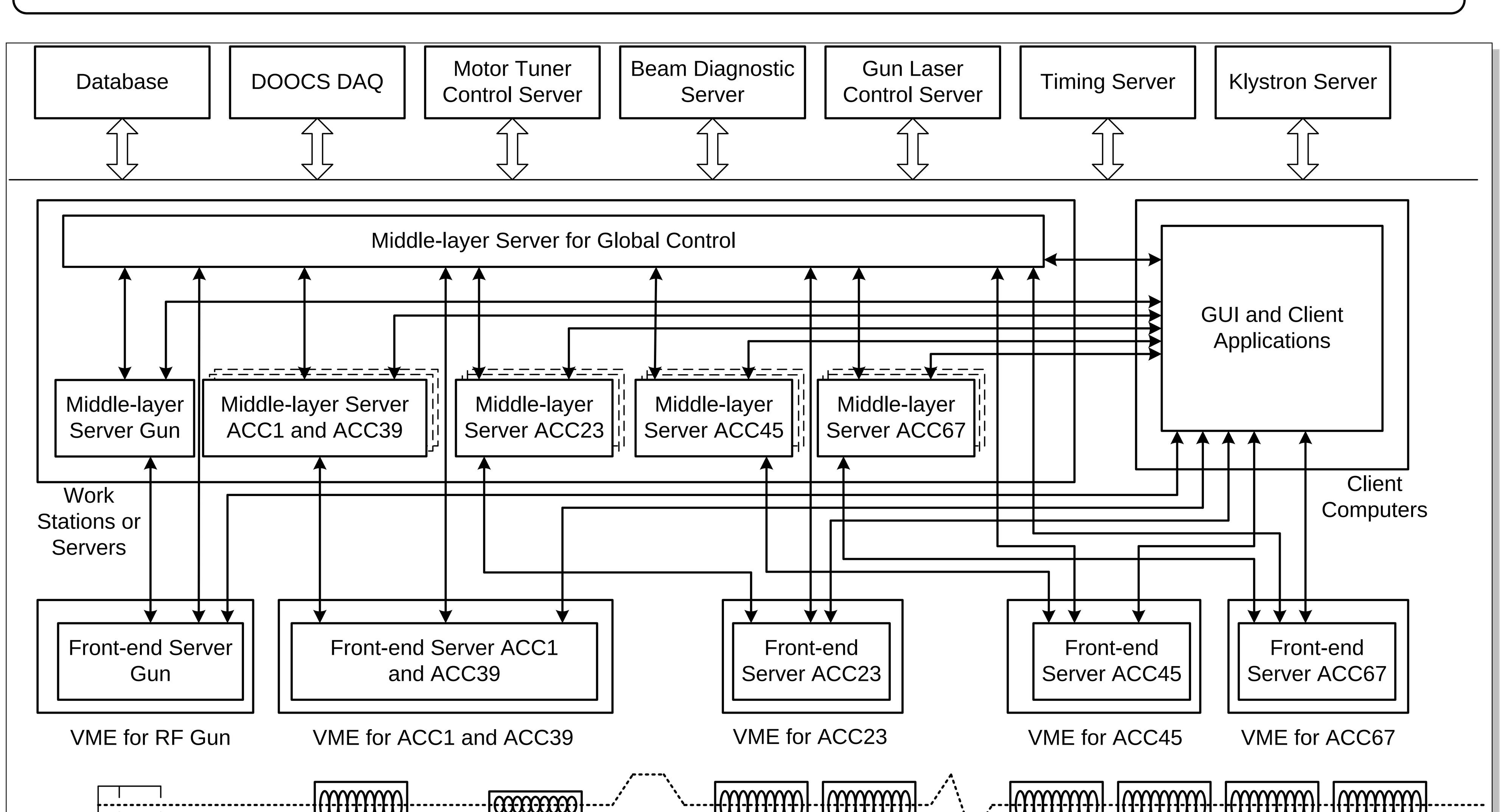
LLRF libraries



Domain breakdown of LLRF Algorithm Library



Dependency between domains



Allocation of LLRF applications to DOOCS servers

Conclusion

Software architecture is designed for LLRF applications at FLASH. This is the first time to implement the LLRF applications in a systematic way. The LLRF Algorithm Library has been developed and several applications have been successfully implemented and tested at FLASH, including the control table generation, vector sum calibration, RF gun calibration, RF system identification and loop phase and loop gain control. The architecture is proved successful for good understandability, maintainability and extendibility, which will be the reference design for the application software of the LLRF system for the European XFEL project. The experiences gained are also useful for the LLRF system design for other machines like ILC and normal conducting accelerators.