

Inside of Optics & Orbit Server/Client System for SuperKEKB

Akio Morita(SAD committer in SuperKEKB commissioning group),
KEK, 1-1, Oho, Tsukuba, Ibaraki 305-0801, Japan

Motivation

In KEKB B-Factory optics/orbit control system, EPICS PVs on different IOCs and files on shared NFS server are used for communication between server and client. Server receiving client data via multiple channels received unexpected data image due to propagation skew among channels. Lock / Unlock primitives for exclusive access implemented on KEKB B-Factory control system had serious lock race. These issues make many troubles during KEKB operation. For analysing such operating system errata, log data of KEKBLog archiver, which is house made EPICS PV archiving system, is not enough because archiver does not record *whydunit*. In such poor software infrastructure, tuning tool programmer can not write correct transaction code.

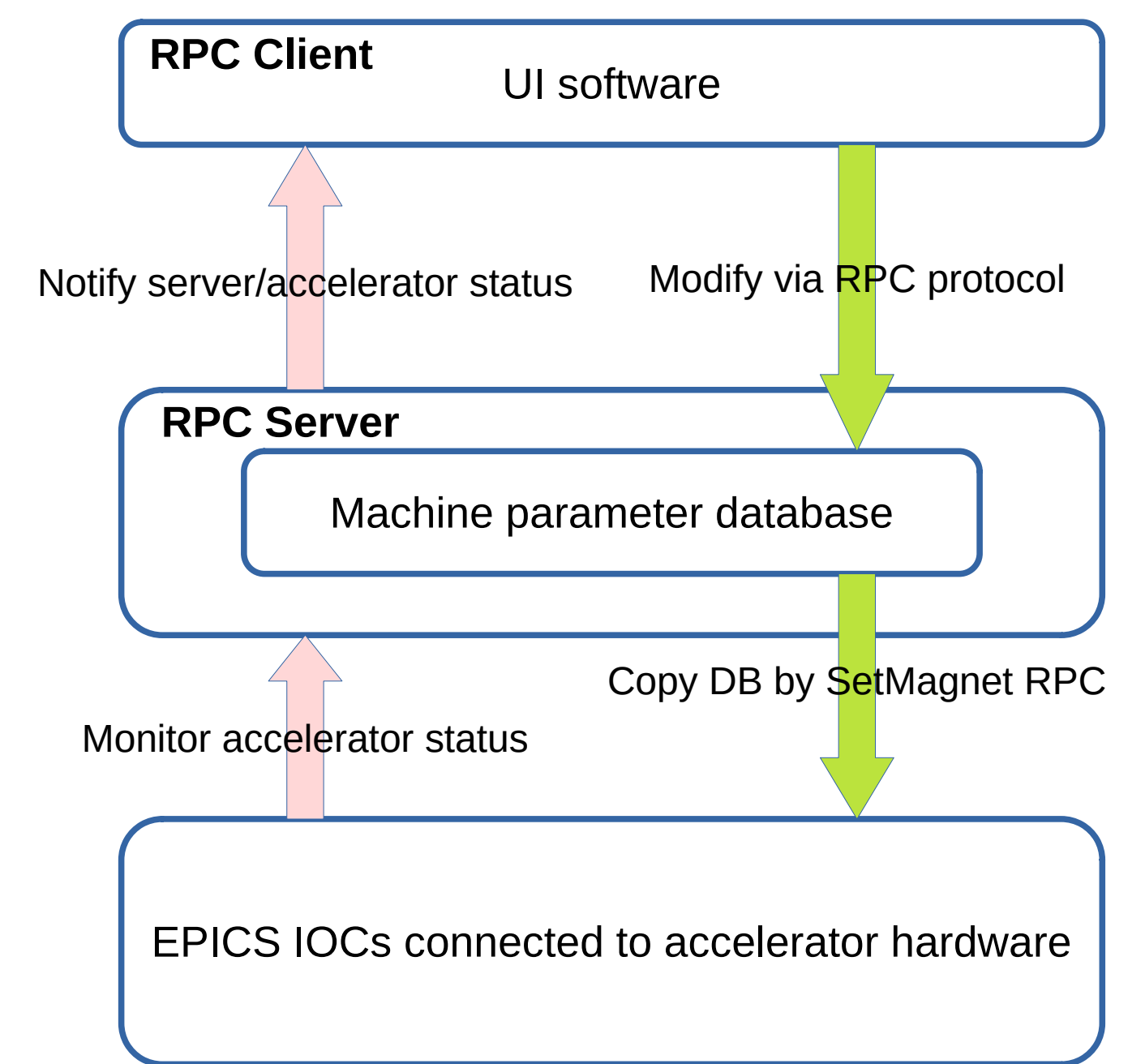
For SuperKEKB control system, consistent control system view and software framework for writing race free transaction are required.

Design Concepts

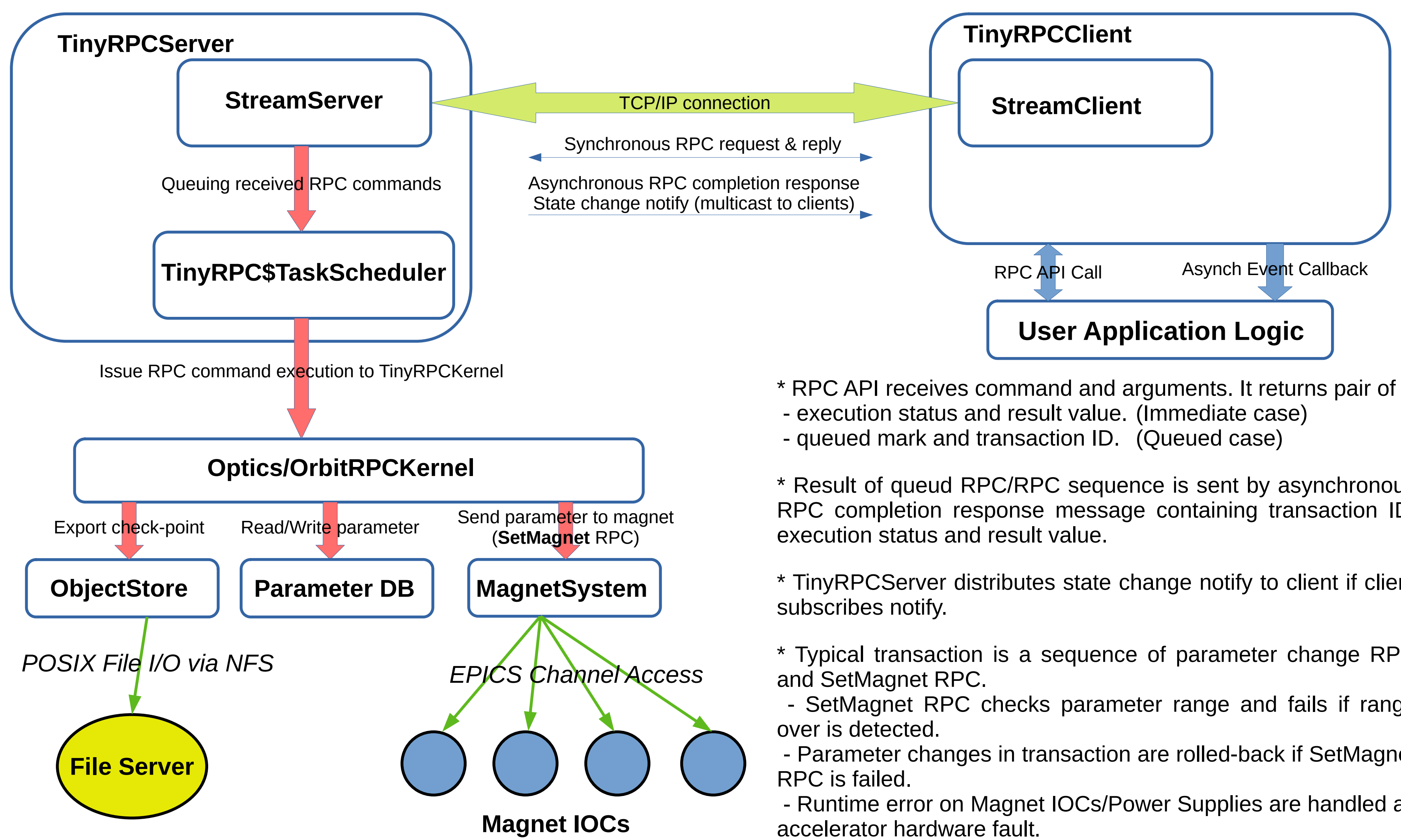
- * Single server / multi client system architecture.
 - Provide simple accelerator control mode.
 - Server side unified hardware control and error handling.
- * Remote Procedure Call(RPC) over TCP/IP network.
 - Data race free communication between server and client.
- * Serial execution semantics by single queue.
- * Transaction support for RPC command sequence set.
- * Named / runtime-creatable optics fudge & orbit bump data set.
 - Provide simple view for linear composable tuning knobs.
 - Provide easy remove action for temporary knobs.
- * Server side MUTEX resources.
 - Provide exclusiveness for GUI client panel instance and parameter manipulation.
- * RPC command level logging.
 - Logging *whodunit* & *whydunit*.
- * Checkpoint logging per accelerator state changes.
 - Provide machine parameter change history with command log.
- * Persistent & race free checkpoint database on POSIX FS.
 - Checkpoint database SHOULD be never corrupted except administrator's miss operation, file system layer bug and block device corruption.

Control Model Concepts

- * Layered architecture
 - Virtual accelerator layer
 - + Machine parameter database on **Server**
 - Real accelerator layer
 - + EPICS online database
- * Triggered copy from virtual to real accelerator
 - Copy parameter set on server database to IOCs via EPICS channel access protocol at SetMagnet RPC timing.

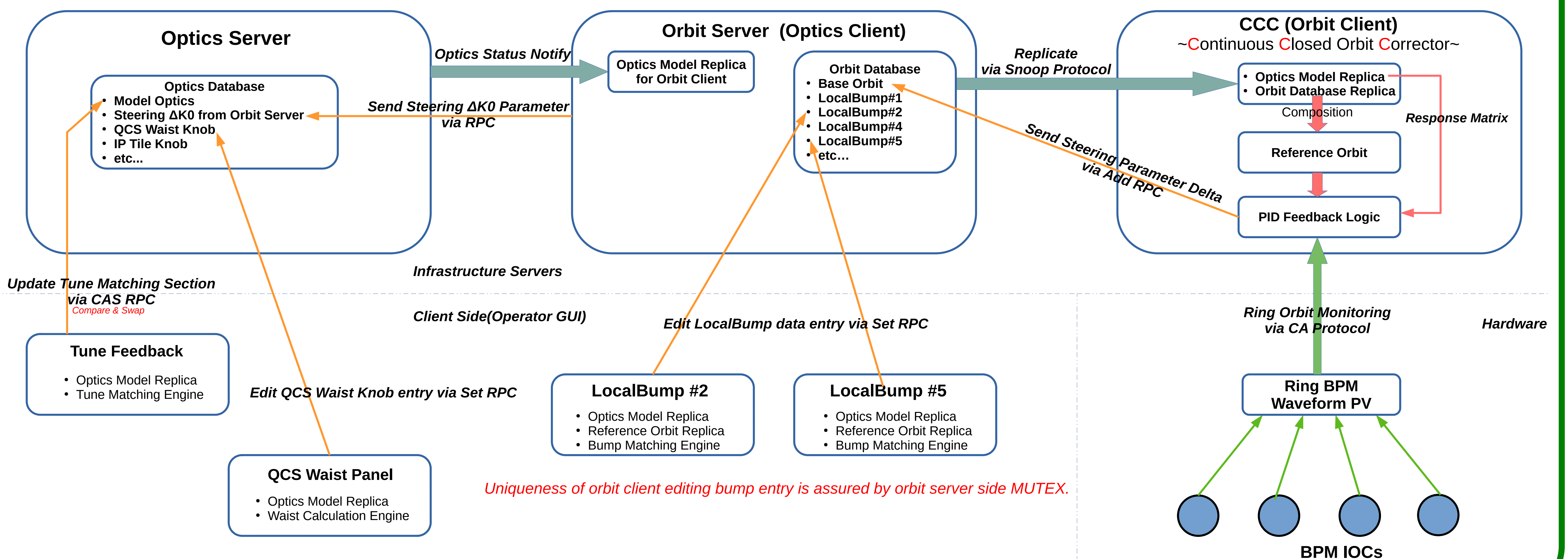


Modular Designed RPC Server/Client System



- [Framework]**
- * **StreamServer/Client** class
 - Message packet exchange layer on TCP/IP socket.
 - + Connection management and Callback per message packet arrival.
 - * **TinyRPCServer/Client** class
 - Remote Procedure Call(RPC) framework on packet exchange.
 - + Session management & MUTEX primitives for lock manipulations
 - + Transaction of RPC sequence supported by TaskScheduler.
 - + Asynchronous RPC completion response & State change notify.
 - Individual RPC commands are injected from TinyRPCKernel instance.
 - * **TinyRPC\$TaskScheduler** class
 - RPC batch execution & queuing framework for TinyRPCServer.
 - * **TinyRPCKernel** class
 - RPC command implementation provider template class.
- [Optics/Orbit Server Backend]**
- * **Optics/OrbitRPCKernel** class
 - Delivered class for SuperKEKB operation from TinyRPCKernel
 - Support accelerator parameter database Get/Set/Add/CAS operation.
 - Support SetMagnet operation to export database to Magnet IOCs.
 - Store server state check-point via ObjectStore class.
 - + Including RPC command level logging & previous state link pointer.
 - * **MagnetSystem** class
 - Abstraction layer for controlling Magnet IOCs.
 - + Synchronous/Asynchronous setter operator.
 - + Magnet IOC state monitoring & Error handling.
 - * **ObjectStore** class
 - Time indexed object database with dependence chain on POSIX FS.
 - + Based on exclusive creation & atomic rename operation.
 - + Support symbolic name tags as root node of object chains.

Optics & Orbit Control System Data Flow



Uniqueness of orbit client editing bump entry is assured by orbit server side MUTEX.