Simulacrum: A Model-Based Simulator for the LCLS Accelerator

Matt Gibbs, William Colocho, Jane Shtalenkova, and Ahmed Osman
SLAC National Accelerator Laboratory

U.S. DEPARTMENT OF ENERGY | Stanford University

SLAC NATIONAL ACCELERATOR LABORATORY

First, A Brief Trip to the Dictionary

Simulacrum (Noun):

1. An image or representation of someone or something.

'a small-scale simulacrum of a skyscraper'

1.1 An unsatisfactory imitation or substitute.

'a bland simulacrum of American soul music'

--Oxford English Dictionary

LCLS

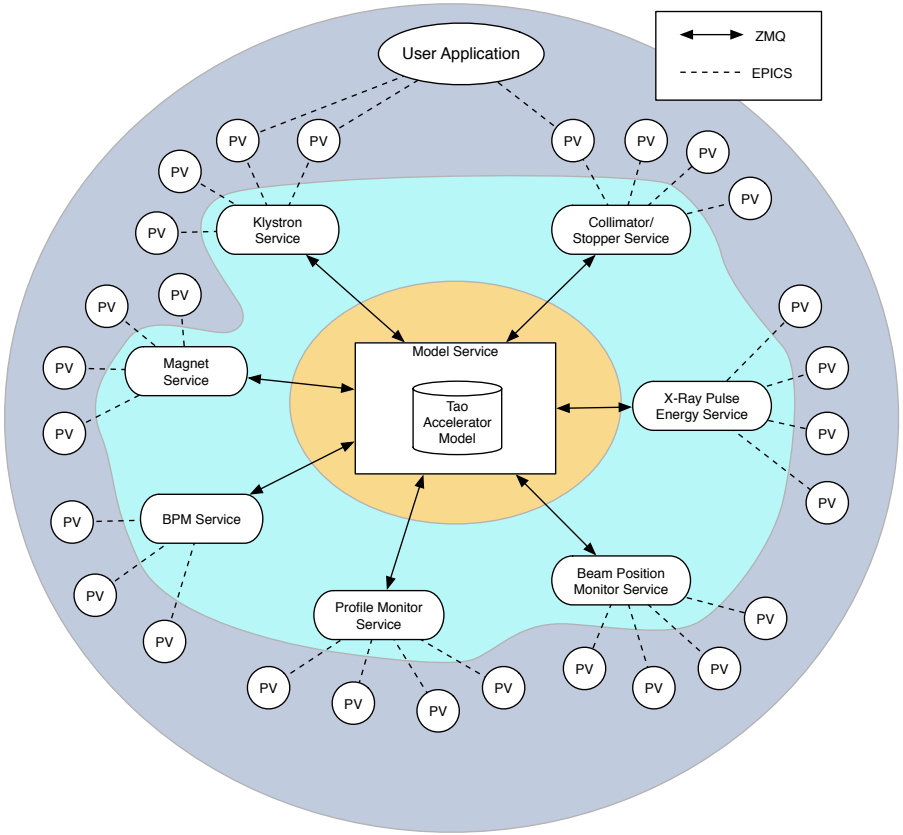
- 78 Klystrons
- 191 Quadrupole Magnets
- 314 Corrector Dipoles
- 39 Dipole Bend Magnets
- 175 Beam Position Monitors
- 20 Profile Monitors
- 19 Wire Scanners
- 32 Collimators (16 movable)
- …And much much more!

Simulacrum

Simulacrum is a system of many small services, written in Python, that work together to simulate the LCLS accelerator. There are three facets to the simulation:

1. **Electron beam dynamics**
2. **The accelerator hardware** (devices like magnets, RF cavities and amplifiers, beam position monitors, beam profile monitors, etc.)
3. **The EPICS control system interface to the hardware** (PVs like “BPMS:LI24:801:X”, which are used to interact with devices)

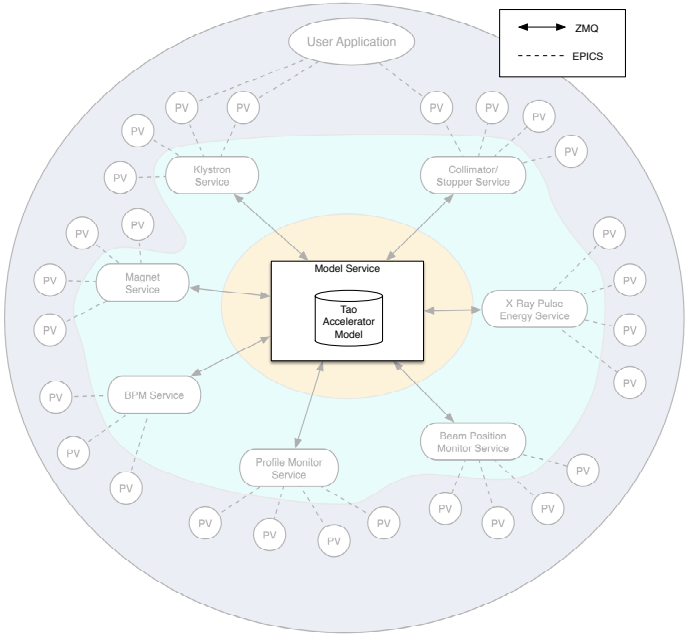
Architecture



Architecture: Model Service

The “Model Service” provides the electron beam dynamics simulation.

- Runs an instance of “Tao” [1], an accelerator modeling environment.
- A ZeroMQ request/reply socket receives Tao commands, executes them, and sends back the results
- A ZeroMQ publish/subscribe socket broadcasts electron beam information as the model changes



[1] https://www.classe.cornell.edu/bmad/tao.html

Tao Commands

A few Tao Command examples:

```
show element Quad::*
```

Returns device information for every quadrupole magnet

```
set element XC00 hkick=0.003
```

Sets the horizontal kick (in radians) for corrector magnet XC00

```
show lat UNDSTART:UNDTERM
```

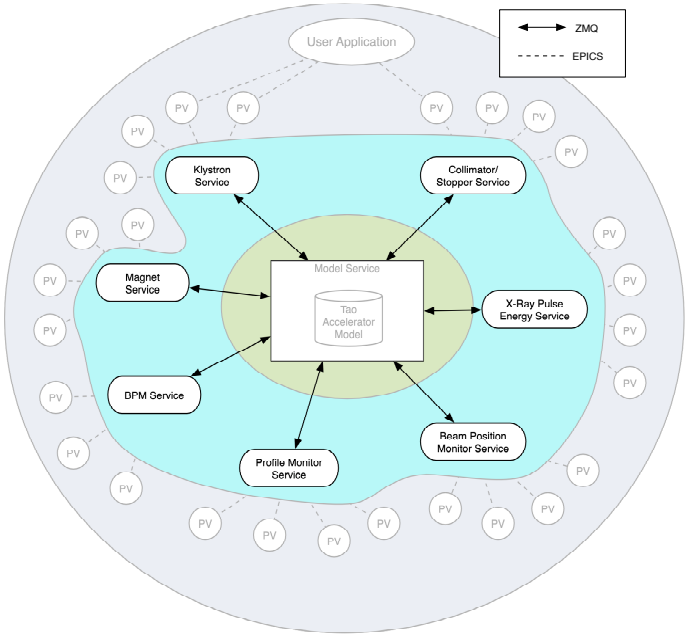
Returns beam information (twiss parameters, momentum, transverse position, and 6x6 transport matrix) at every element between “UNDSTART” and “UNDTERM” marker points

Architecture: Device Services

Device services simulate accelerator hardware.

- Send commands to the Model Service via ZeroMQ request/reply
- Subscribe to beam information published by the Model Service
- Simulate device logic (on/off commands, interlocks, setpoints, etc.)

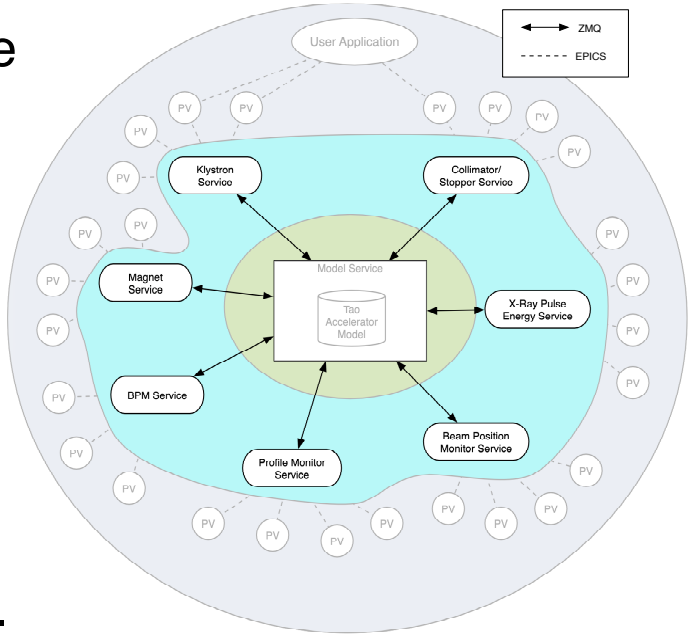
Services currently exist for Magnets, Beam Position Monitors, Klystrons, Collimators, Profile Monitors, and X-Ray Pulse Energy Measurements



Architecture: EPICS Interface

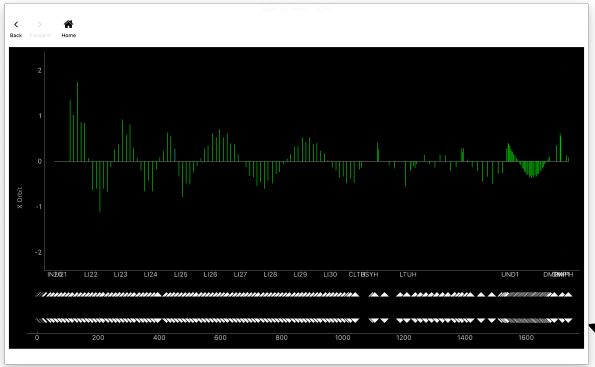
Device services also host EPICS PVs to interact with the devices

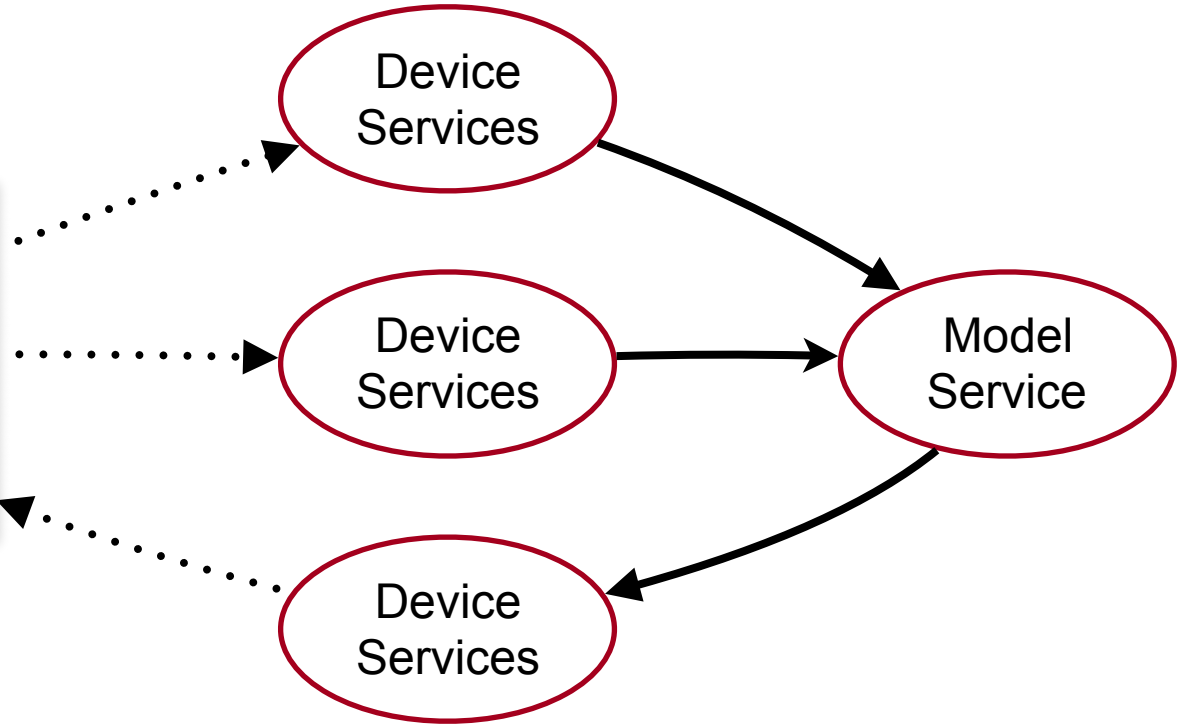
- Uses “caproto” [1] Python module to handle EPICS communication
- Callbacks on PV read/write operations change device state and send Tao commands to inform the model of the changes
- New beam information updates device state, triggering PVs to publish new values.



[1] https://caproto.github.io/caproto/

Data Flow




Device Services
Device Services
Device Services
Model Service

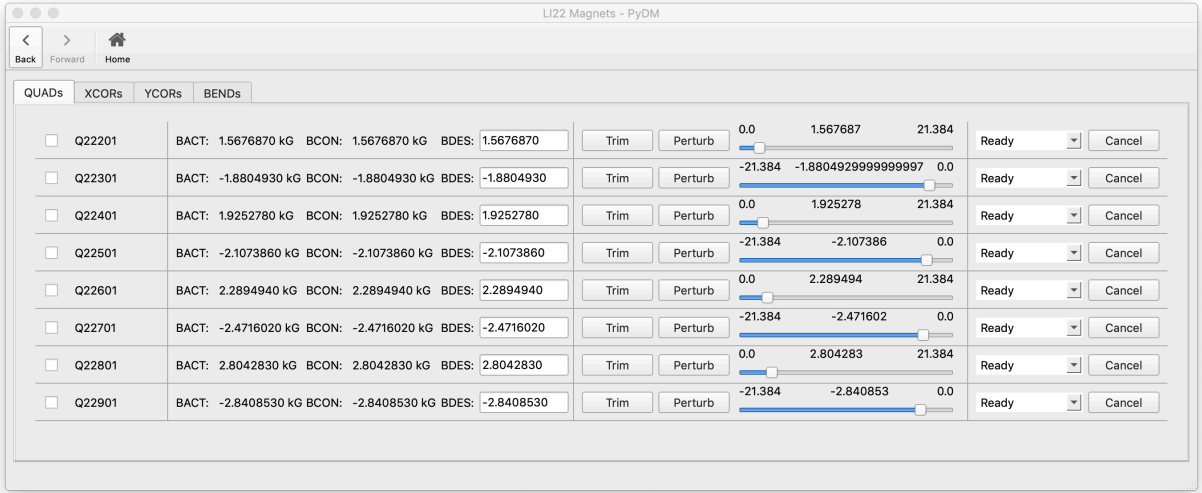
Implementation Details

SLAC

- All Simulacrum services are written in Python 3, heavily utilizing the ‘asyncio’ module.
- At SLAC, a dedicated server runs all of the services on the same machine. Splitting up services to run on different machines is possible.
- By adjusting environment variables for ZeroMQ ports and EPICS ports, multiple instances can run on the same machine without interfering with one another.
- Simulacrum easily runs on a laptop too. Play with an accelerator on the beach!

Use Case: Control System Display Development

The LCLS is currently undergoing upgrades for the LCLS-II Project. Entirely new electron and photon beam lines are being added. We've used Simulacrum as a way to prototype new control system displays for magnets.



Use Case: Improving Optimization Tools

SLAC

A multidimensional optimization tool called Ocelot [1] is frequently used at LCLS to optimize FEL pulse intensity by adjusting quadrupole magnets. A new “Gaussian Process” optimization method was tested during accelerator downtime by running Ocelot against Simulacrum PVs.

[1] S. Tomin et. al, “Progress in Automatic Software-Based Optimization of Accelerator Performance,” *Proceedings of IPAC 2016, Busan, Korea, May 8-13 2016.*

Use Case: Validating Beam-Based Alignment of LCLS Undulators

SLAC

LCLS must precisely align > 20 undulator segments to overlap the electron beam with the x-rays produced by the undulators.

By injecting known misalignments into Simulacrum's Tao model of the undulators, then running Beam-Based Alignment (BBA) software, we can compare the corrections calculated by BBA to the known misalignments, and verify that the software will function properly on the real machine.

Use Case: Training Scenarios for New Operators

SLAC

Simulacrum can be an easy way for new operators to gain hands-on experience with the software used to run the accelerator. Scripts can set up “scenarios” for an operator to tackle:

- Use manual steering controls to flatten the electron beam trajectory in the linac (Complete)
- Diagnose the cause of poor FEL performance (mis-matched accelerator optics, tripped-off klystron or quadrupole magnet) (Planned)

Adapting to Other Accelerators

SLAC

- The Model Service is nearly “machine agnostic”, it will run with any BMAD lattice file. Translators from XSIF and MAD lattice files to BMAD lattice files are provided with Tao.
- The existing device services implement SLAC’s (linac) EPICS interfaces. LCLS-specific code is used by to translate from element names in the lattice to PV prefixes: “XC00” becomes “XCOR:IN20:121”

Conclusion

- Simulacrum is an accelerator simulator for LCLS
- Built modularly, with many different intercommunicating processes simulating different aspects of the machine.
- The “end result” are PVs that provide the same EPICS interface as the real accelerator hardware. Accelerator software can be run using Simulacrum PVs instead of the real machine PVs.
- Simulacrum is a useful tool for software development and operator training.

Thank You!

Questions?