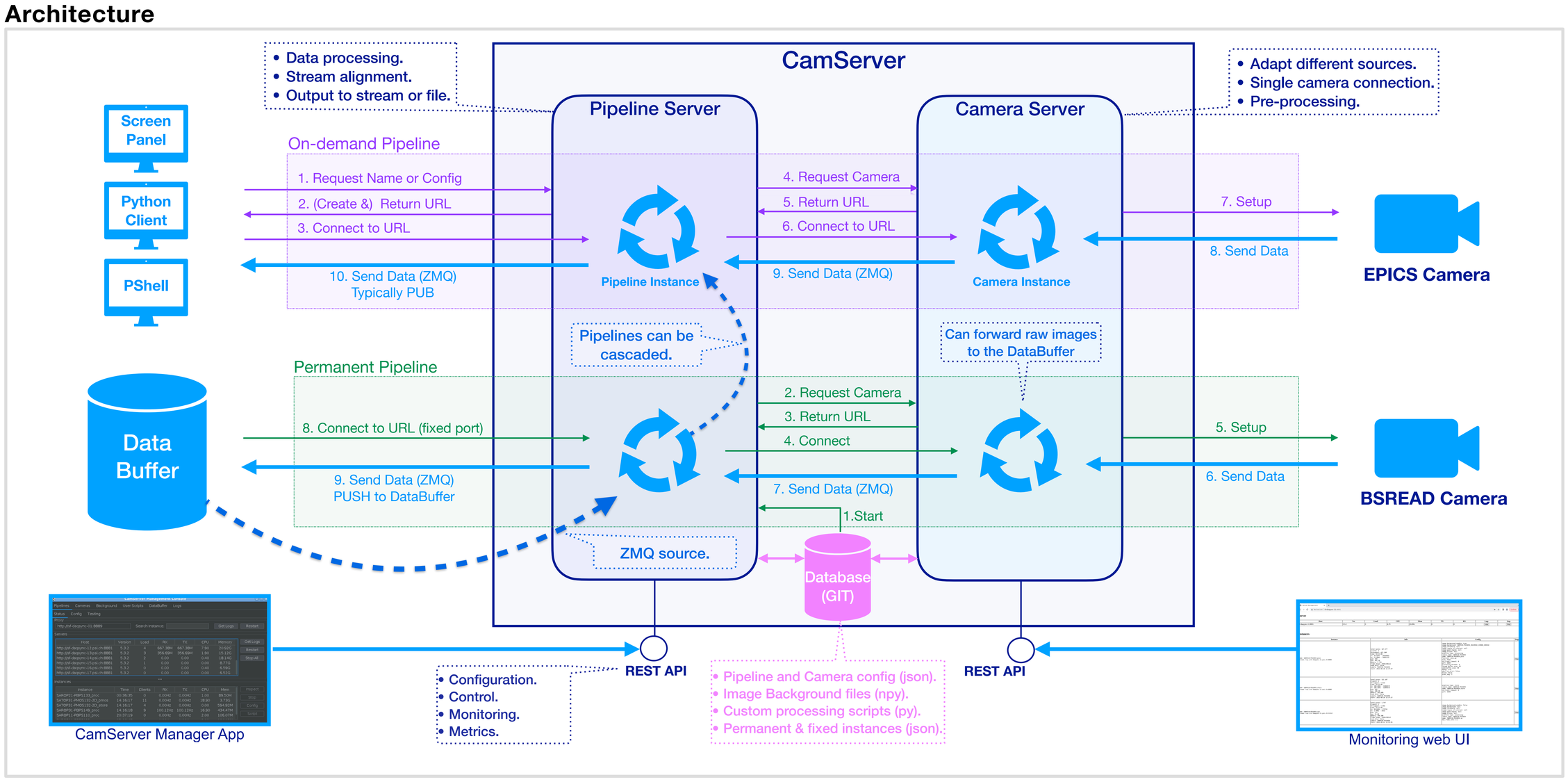
CamServer: Stream Processing @ SwissFEL and SLS 2.0

A. Gobbo[1], A. Babic[1]
[1] Paul Scherrer Institut (PSI), CH-5232 Villigen PSI, Switzerland

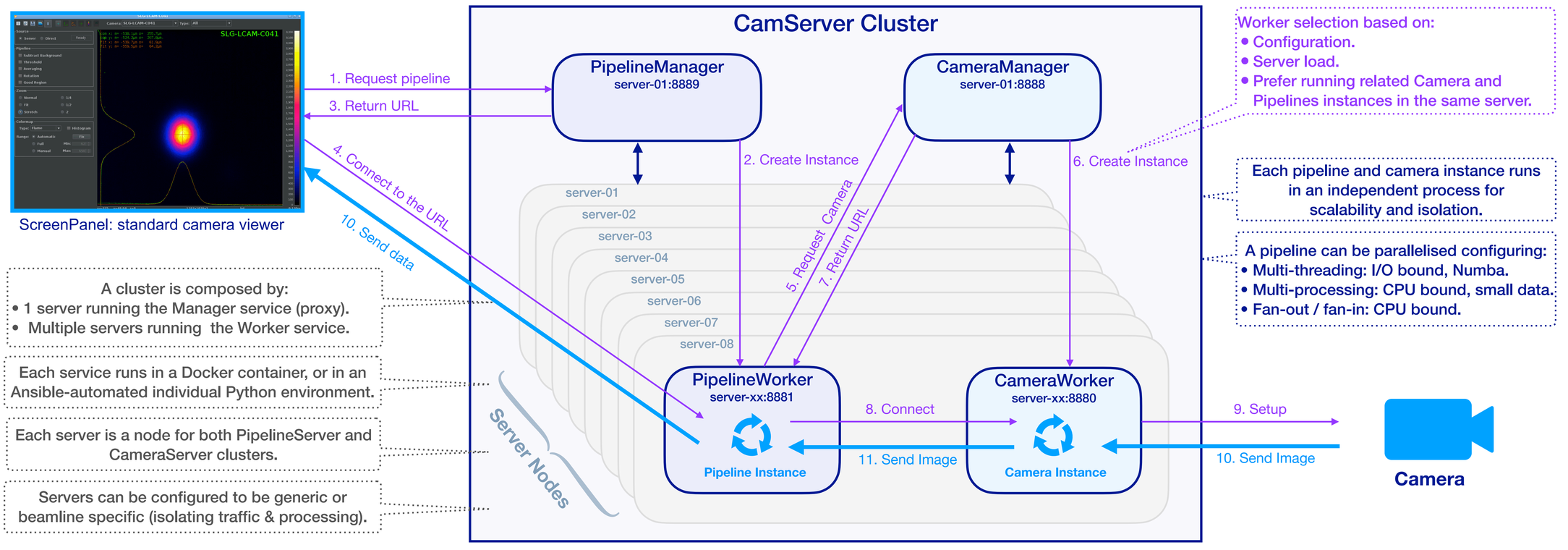
Abstract

CamServer is a Python package for data stream processing developed at Paul Scherrer Institute (PSI). It is a key component of SwissFEL's data acquisition, where it is deployed on a cluster of servers and used for displaying and processing images from all cameras. It scales linearly with the number of servers and is capable of handling multiple high-resolution cameras at 100Hz, as well as a variety of data types and sources. The processing unit, called a pipeline, runs in a private process that can be either permanent or spawned on demand. Pipelines consume and produce ZMQ streams, but input data can be arbitrary using an adapter layer (e.g. EPICS). A proxy server handles requests and creates pipelines on the cluster's worker nodes according to rules. Some processing scripts are available out of the box (e.g. calculation of standard beam metrics) but users can upload custom ones. The system is managed via its REST API, using a client library or a GUI application. CamServer's output data streams are consumed by a variety of client types such as data storage, image visualisation, monitoring and DAQ applications.

Architecture



Deployment



Figures (SwissFEL)

- Currently there are 480 configured processing pipelines and 200 cameras.
- More than 50 permanent pipelines continuously send data to the Data Buffer (temporary storage for DAQ).
- On average more than 10 on-demand pipelines run at a time: the standard SwissFEL camera viewer application is a CamSever client.
- The cluster is composed by 13 servers Intel(R) Xeon(R) Gold 6342 @ 2.80GHz each having 48 cores, 96 threads and 25Gb network adapters.
- At 100Hz each server is limited by network bandwidth to 2 high-resolution cameras (2560x2160) but supports many (>>10) low-resolution cameras.

https://github.com/paulscherrerinstitute/cam_server