Clas12 Reconstruction and Analysis Framework

V. Gyurjyan S. Mancilla

JLab

- Thomas Jefferson National Accelerator Facility (TJNAF), commonly called Jefferson Lab or JLab, is a U.S. national laboratory located in Newport News, Virginia
- Superconducting RF technology based accelerator will provide a 12 GeV continuous electron beam with a bunch length of less than 1 picosecond.
- Nuclear physics experiments in 4 end-stations (A,B,C,D)
- CLAS is a large acceptance spectrometer installed in Hall B to study
  - Quark-gluon interactions with nuclei
  - Nucleon-nucleon correlations
  - Nucleon quark structure imaging,
  - etc.

CLAS12 Computing Requirements

- Enhance utilization, accessibility, contribution and collaboration
  - Reusability of components
  - On-demand data processing.
  - Location independent resource pooling.
  - Software agility
- Utilization of multicore processor systems
  - Multi-threading
- Ability to expand computing power with minimal capital expenditure
  - Dynamic elasticity.
  - Utilization of IT resources of collaborating Universities.
  - Take advantage of available commercial computing resources.

Ideas Adopted From GAUDI

- Clear separation between data and algorithms
- Services encapsulate algorithms
- Services communicate data
- Three basic types of data: event, detector, statistics
- Clear separation between persistent and transient data

No code or algorithmic solution was borrowed.

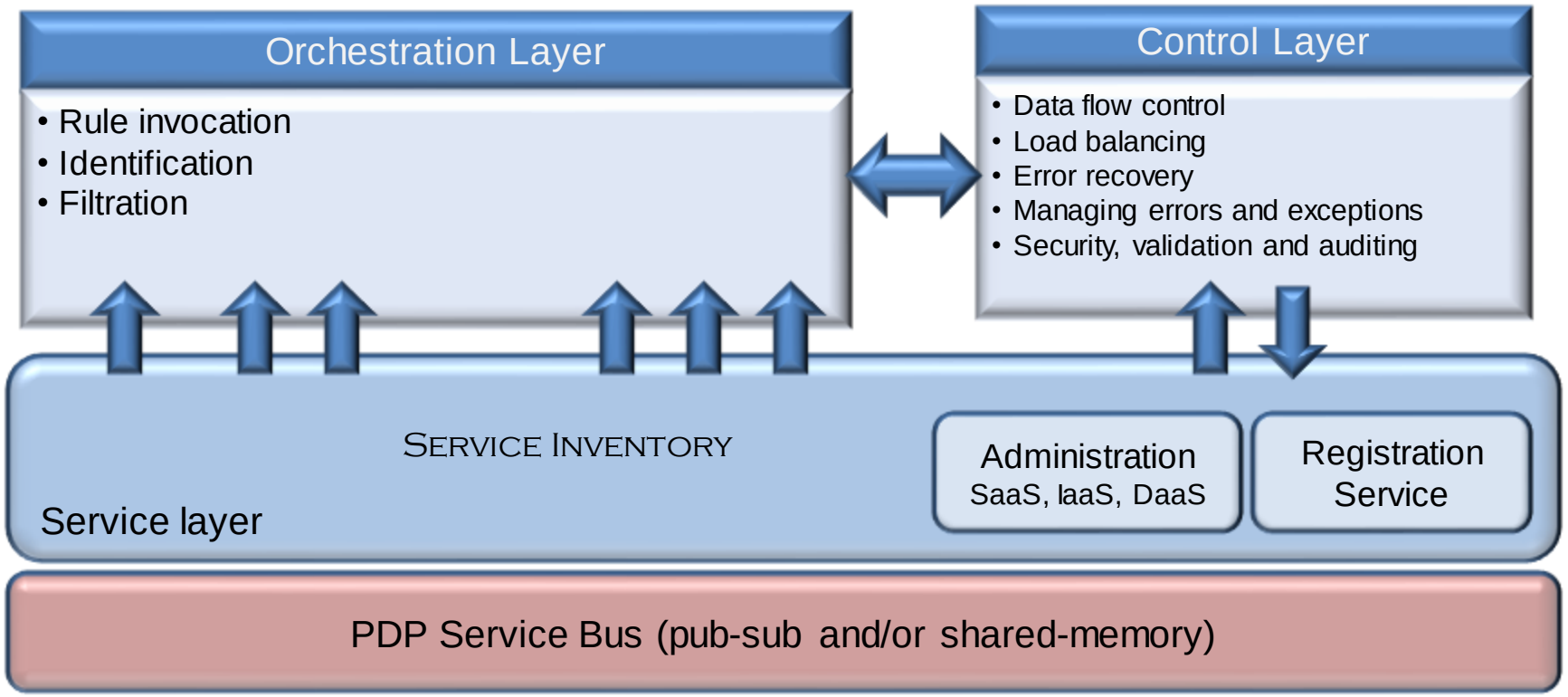
Computing Model and Architecture Choice

- ClaRA is an implementation of the SOA
- Data processing major components as services
  - Multilingual support
    - Services can be written in C++, Java and Python
- Physics application design/composition based on services
- Supports both traditional and cloud computing models
  - Single process as well as distributed application design modes
  - Centralized batch processing
  - Distributed cloud processing
- Multi-Threaded

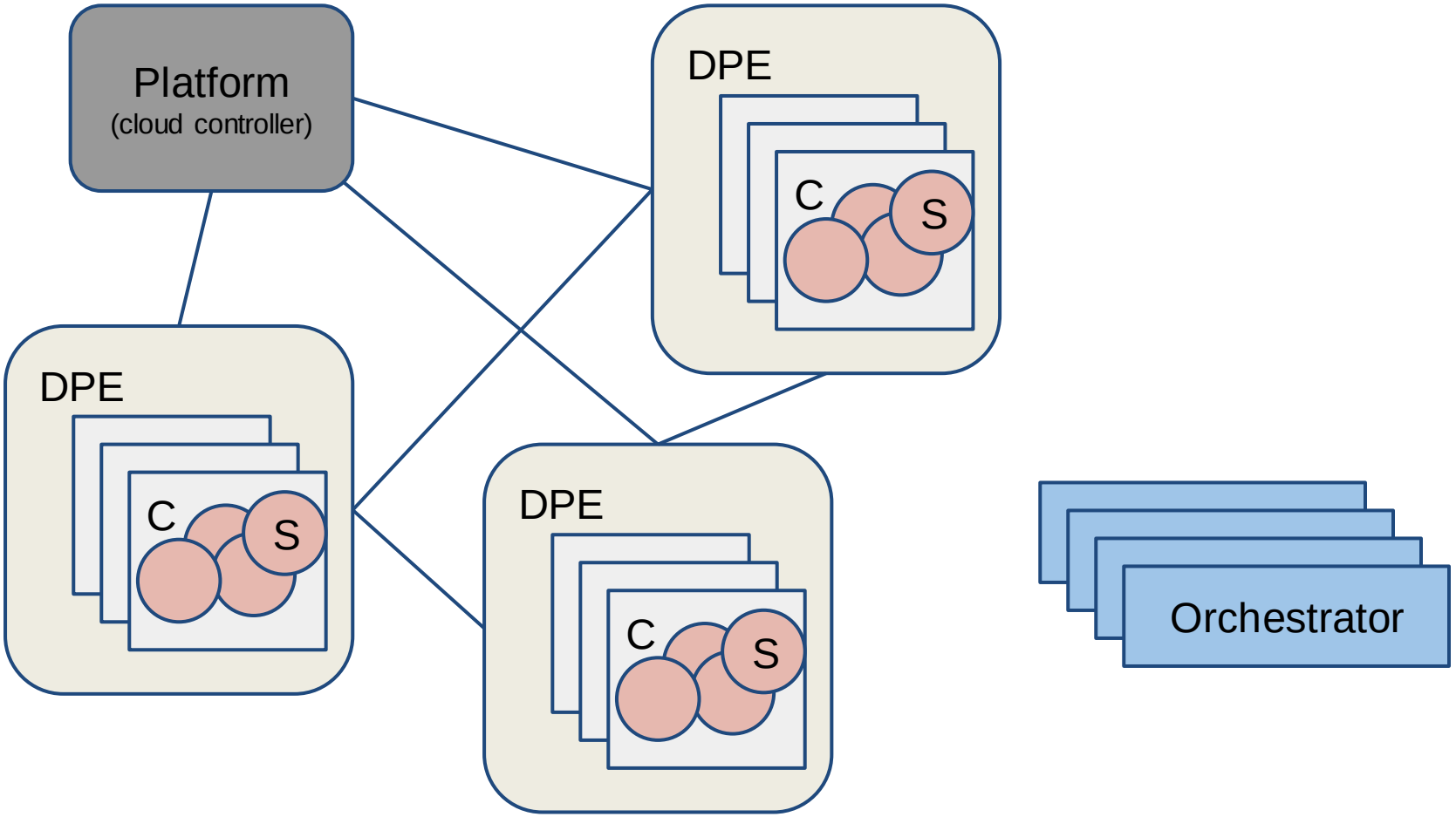
Attributes of ClaRA Services

- Communicate data through shared memory and /or pub-sub messaging system
  - Well defined, easy-to-use, data-centric interface
- Self-contained with no dependencies on other services
  - Loose coupling. Coupled through communicating data.
- Always available but idle until requests arrival
- Location transparent
  - Services are defined by unique names, and are discovered through discovery services
- Combine existing services into composite services or applications
  - Services can also be combined in a single process (run-time environment), communicating data through shared memory (traditional computing model).

ClaRA Design Architecture

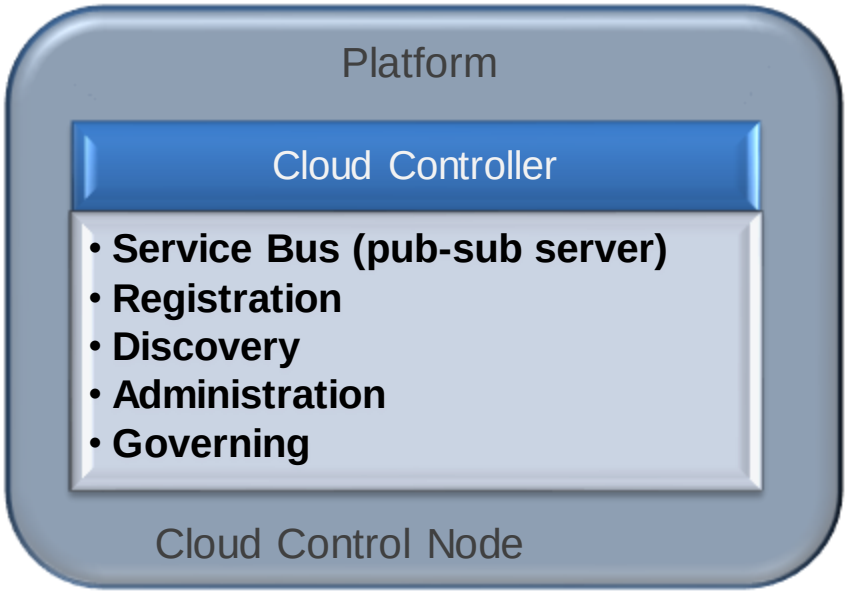


ClaRA Components



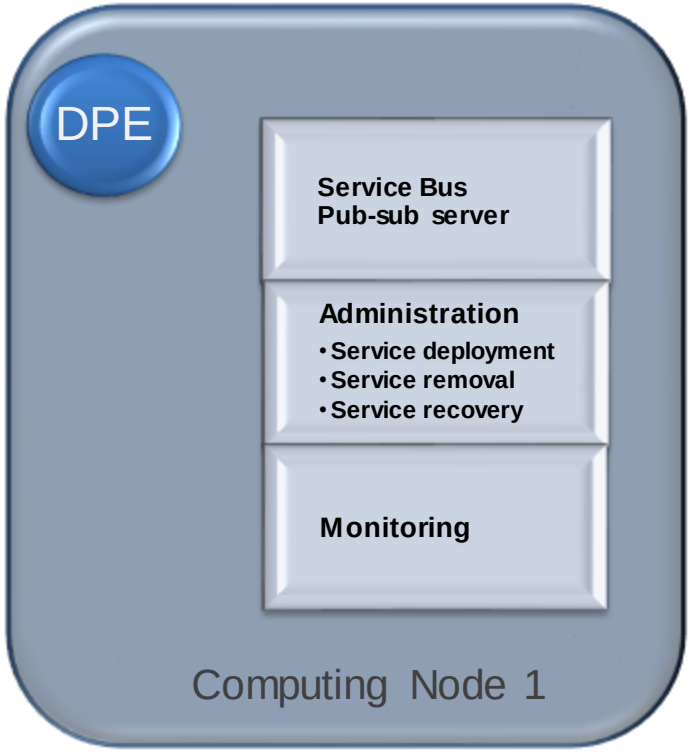
Platform (Cloud Controller)

- ClaRA administration
- Service registration and discovery.
- Keeps an inventory of all running DPEs and all deployed services.
- Used by orchestrators to discover and check services availability and distribution.



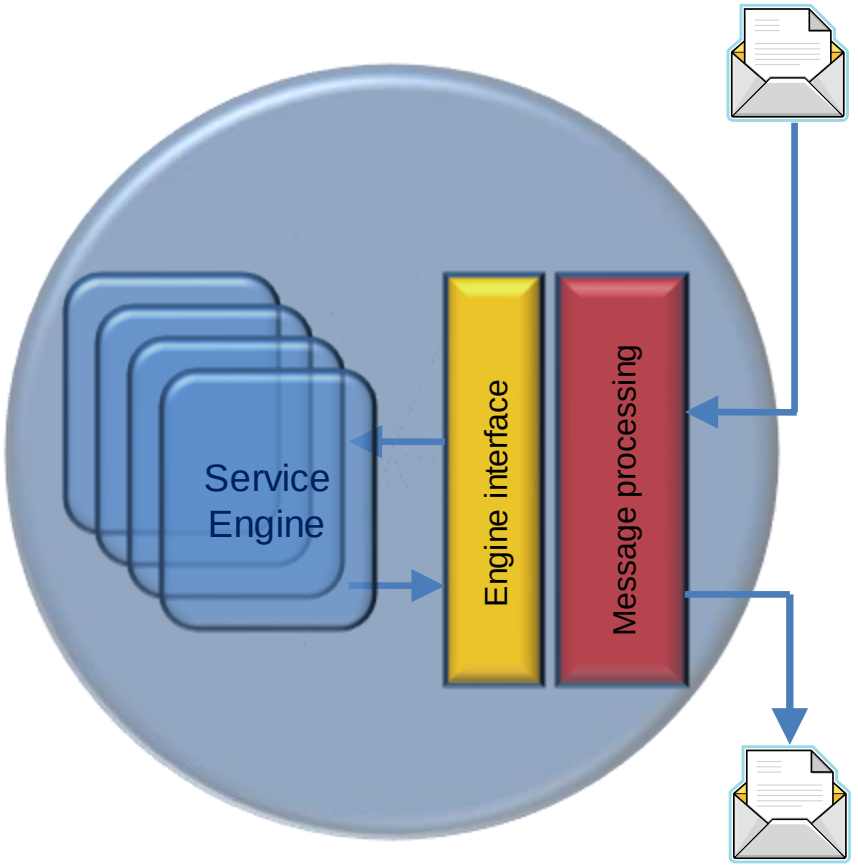
Data Processing Environment (DPE)

- Main ClaRA processes.
- Each node acts as a DPE.
- All services are deployed and executed by threads inside the DPE process.
- Global memory to share data between services.

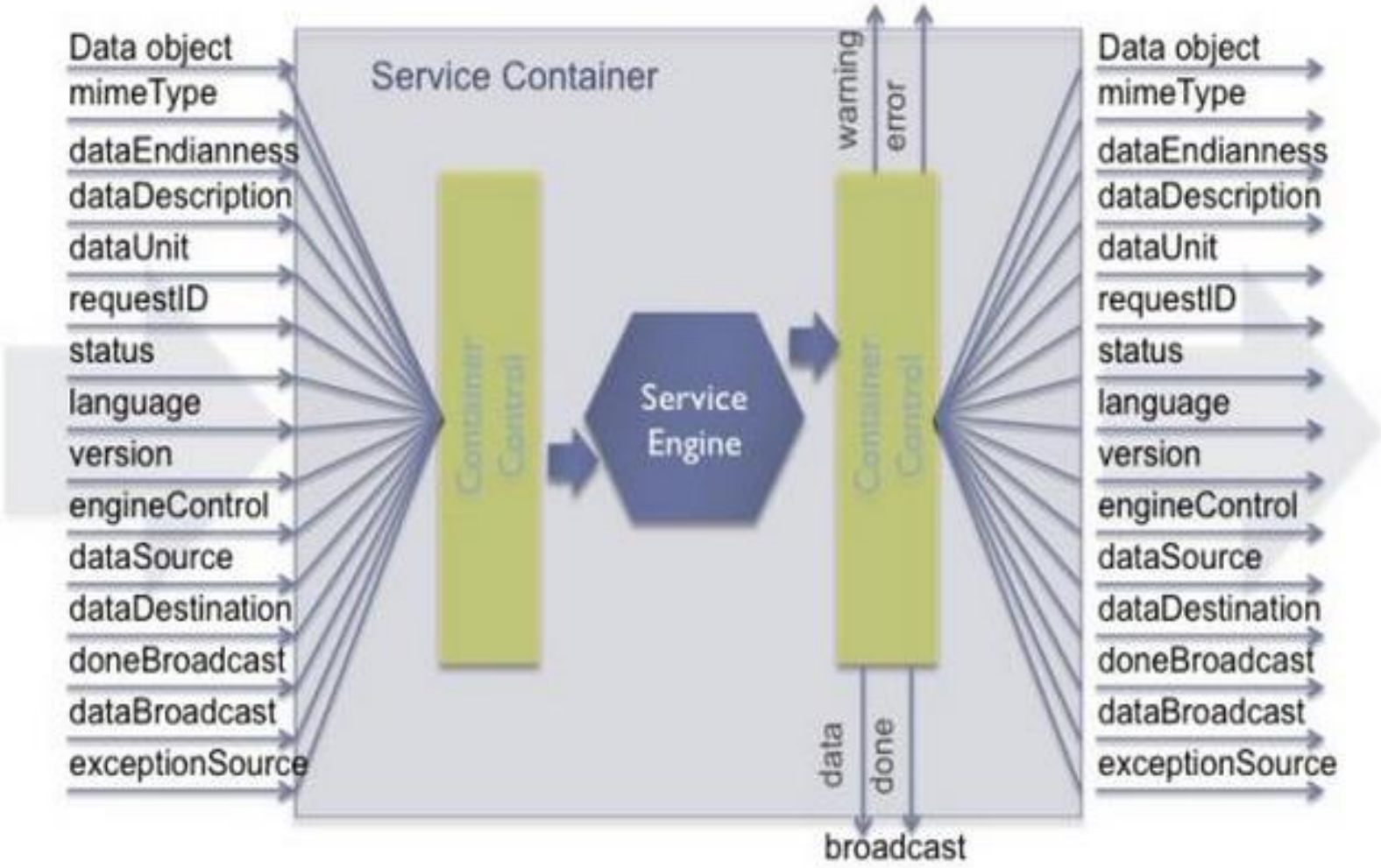


Service Container

- Group and manage services in a DPE
- Can be used as namespaces to separate services.
  - The same service engine can be deployed in different containers in the same DPE.
- Handle service execution and its output.
- Service container presents a user engine as an SOA service (SaaS implementation).



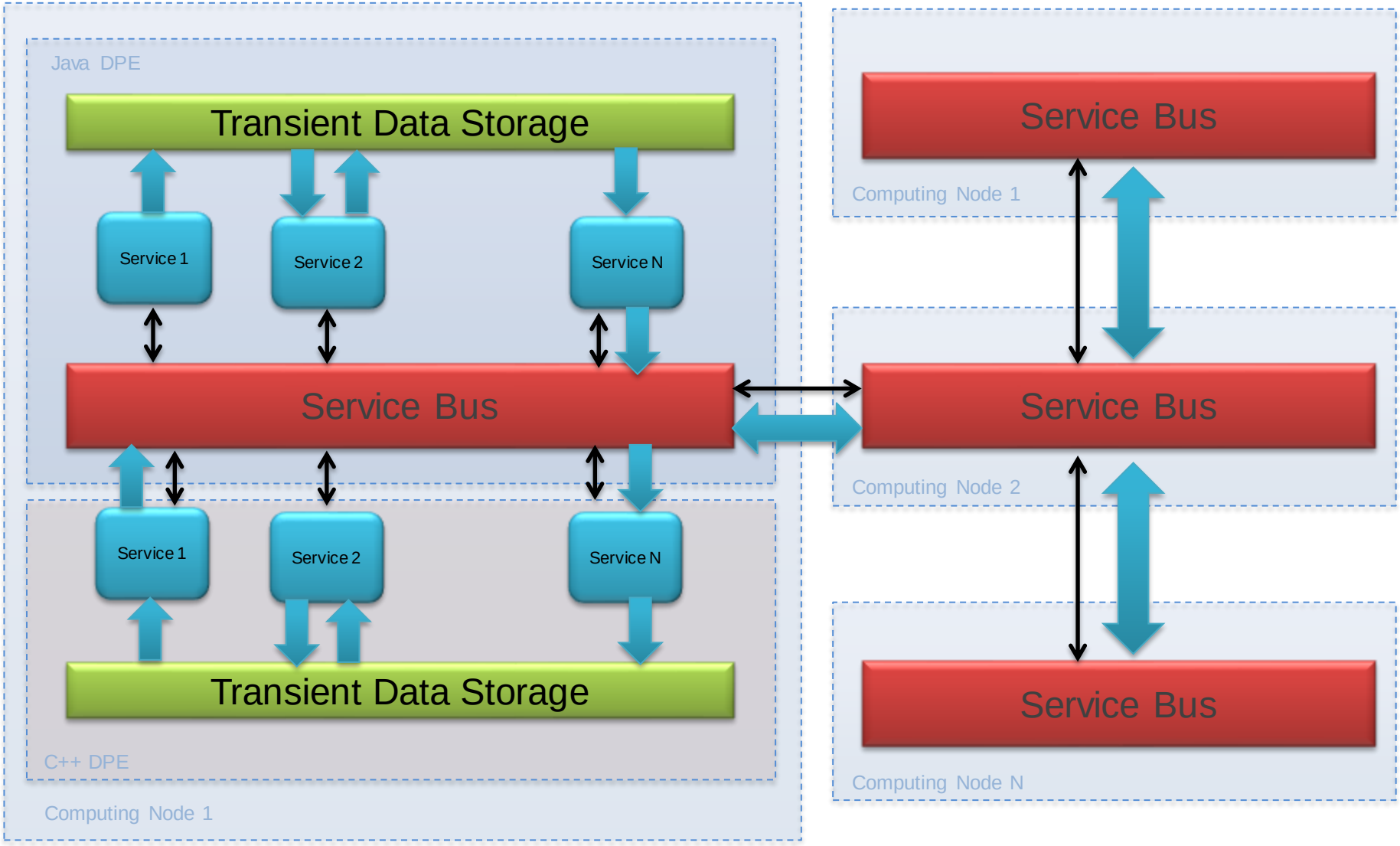
Transient Data Envelope



Transient Data Object

- EVIO 4.1 is the default event format.
  - Data is just a byte buffer (avoid serialization).
  - Complete API (Java, C++,Python) to get data from the buffer.
  - A set of wrappers to work with the common CLAS12 bank format.

Service Communication



Service Engine

- The fundamental unit of ClaRA based application.
- Receives an input data in an envelope, and generates an output data.
  - The data envelope is the same for all services.
- Implements ClaRA standard interface
  - A configure method
  - An execute method.
  - Several description/identification methods.
- Must be *thread-safe*.
  - The same service engine can be executed in parallel multiple times.

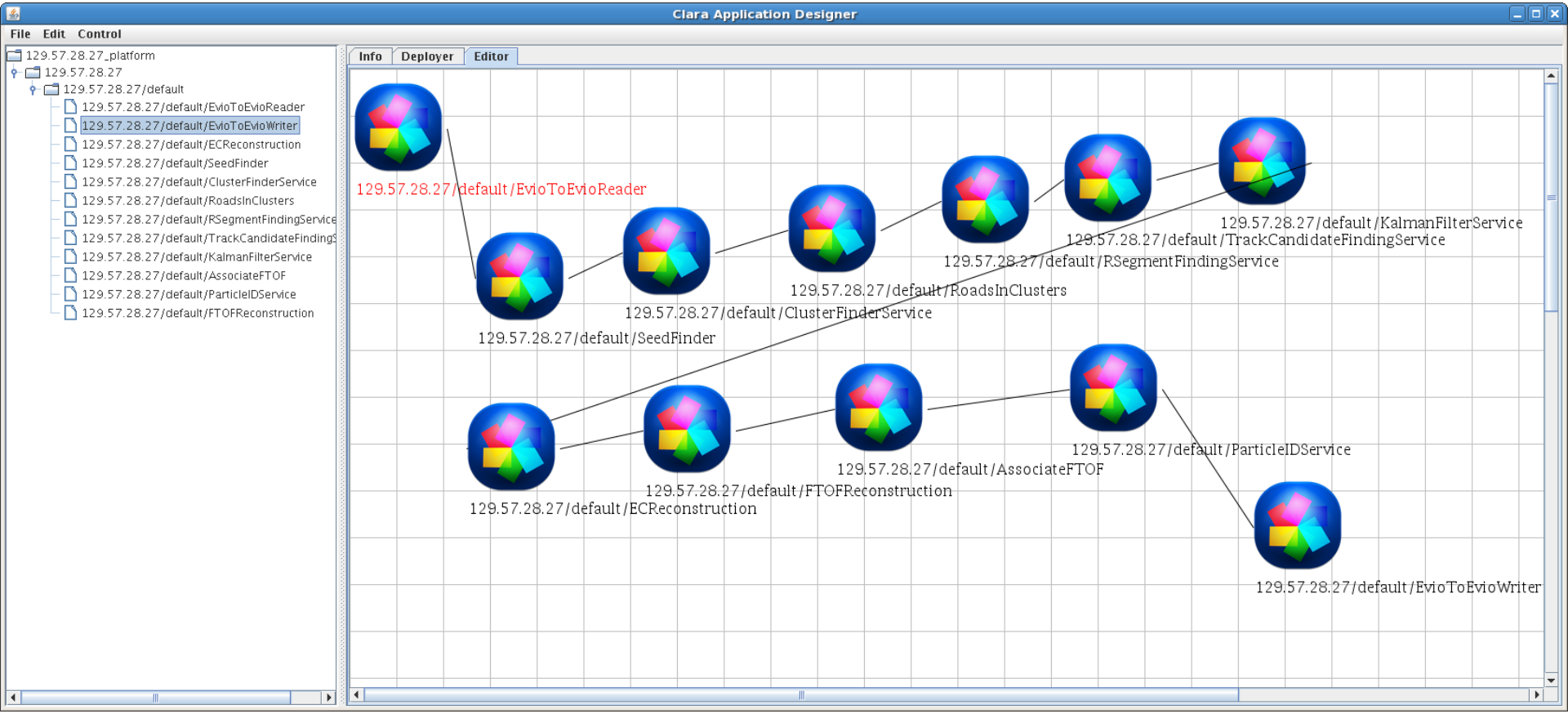
Orchestrator

- Design and control ClaRA applications
- Coordinate services execution and data flow.
- Usually run outside of the DPE.
- Deploy services to DPEs.
  - Each deployed service is identified by the following canonical name:*dpe_name/container_name/service_engine_name*
- Link services together.
  - The output of a service is sent as the input to its linked service.

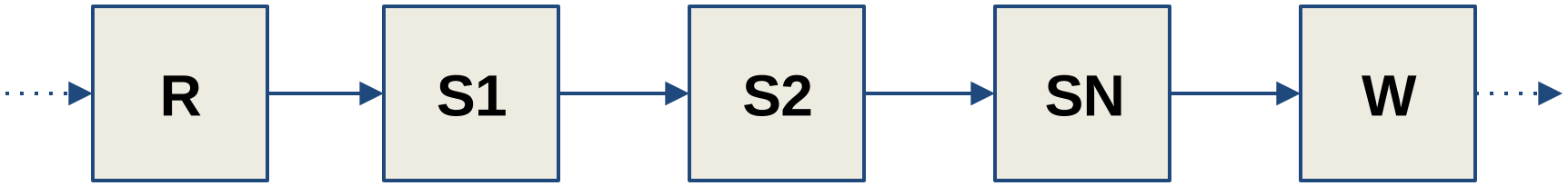
Orchestrator

- Request services execution.
  - *Async requests:* service output is sent to all its linked services.
  - *Sync request:* service output is returned to the requester.
- Monitor services execution.
  - Data, done, warning and error monitoring.
  - Run custom callback code when a notification is received.

Application Graphical Designer

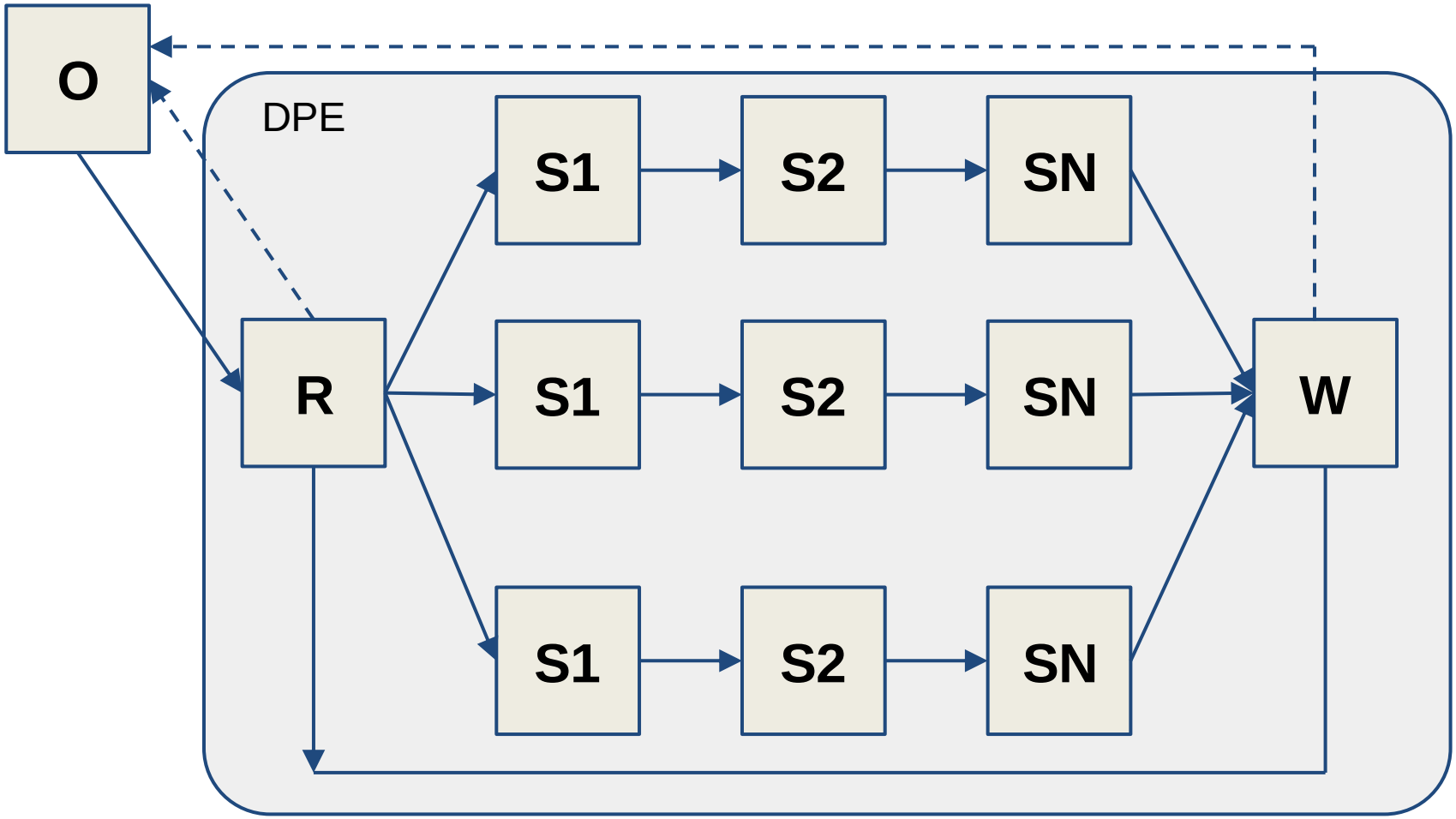


Single Event Reconstruction

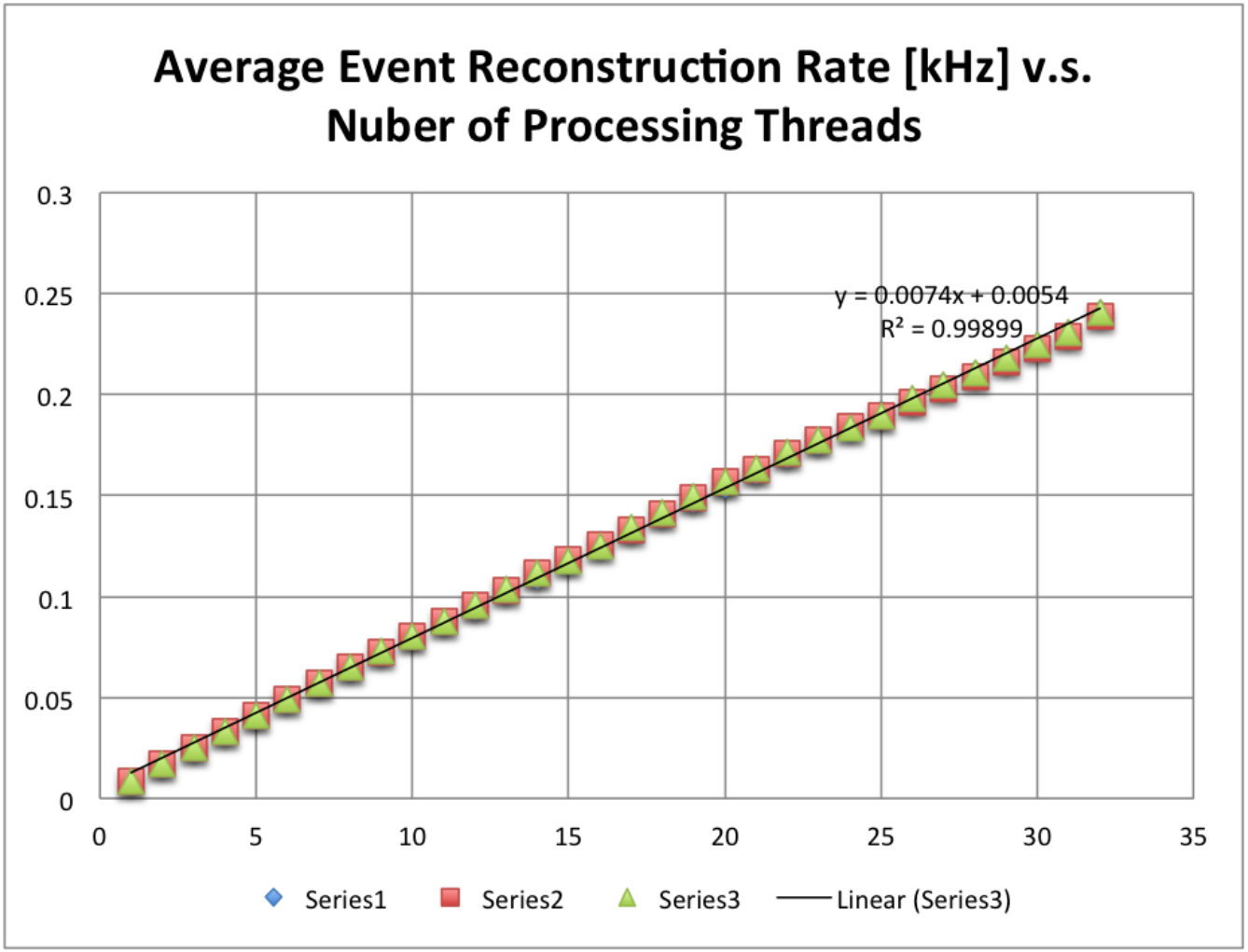


- Read EVIO events from input file.
- Events pass from service to service in the chain.
  - Services add more banks to the event.
- Write events to output file.

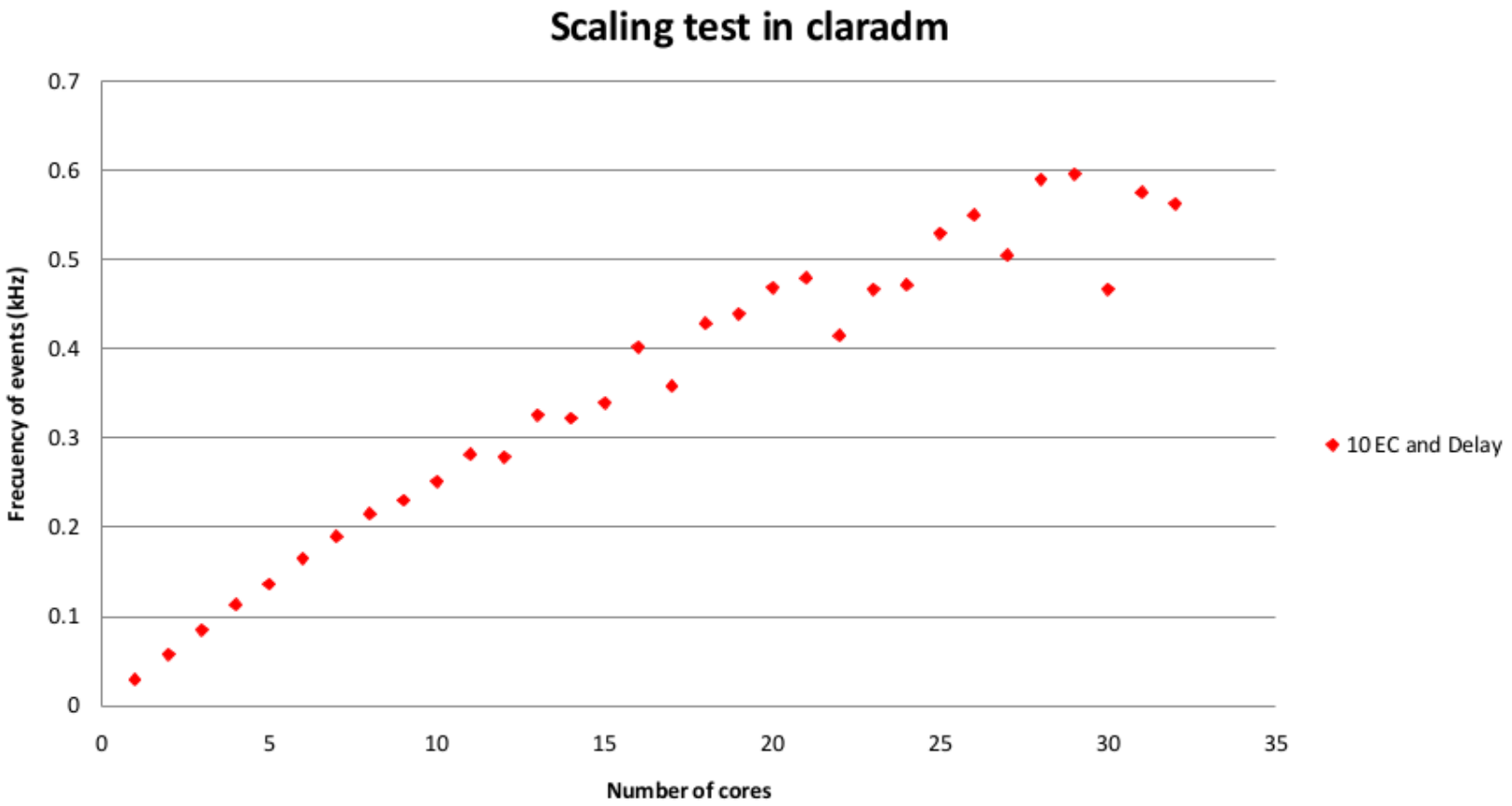
Multi-Core Reconstruction



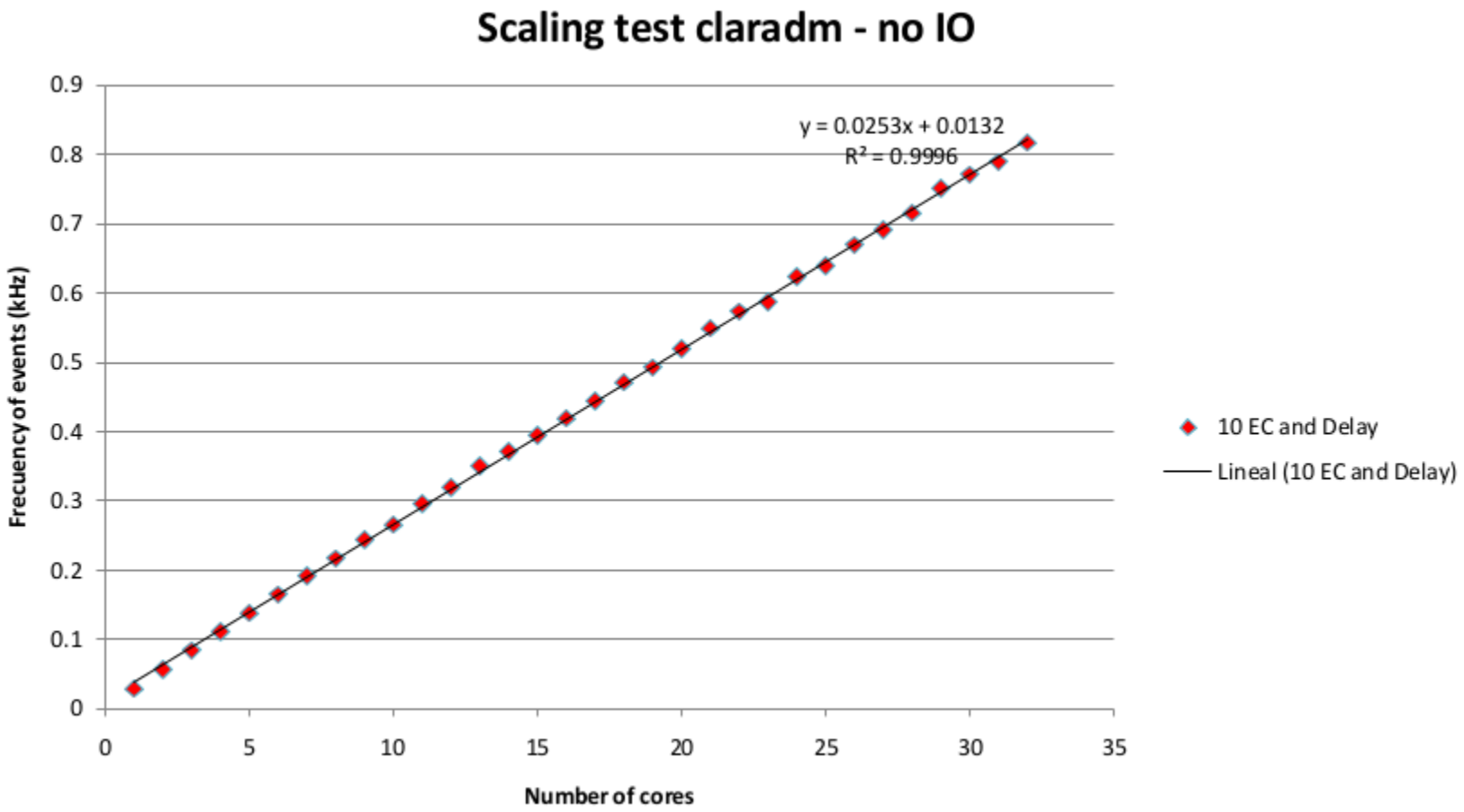
Multi-Core Reconstruction



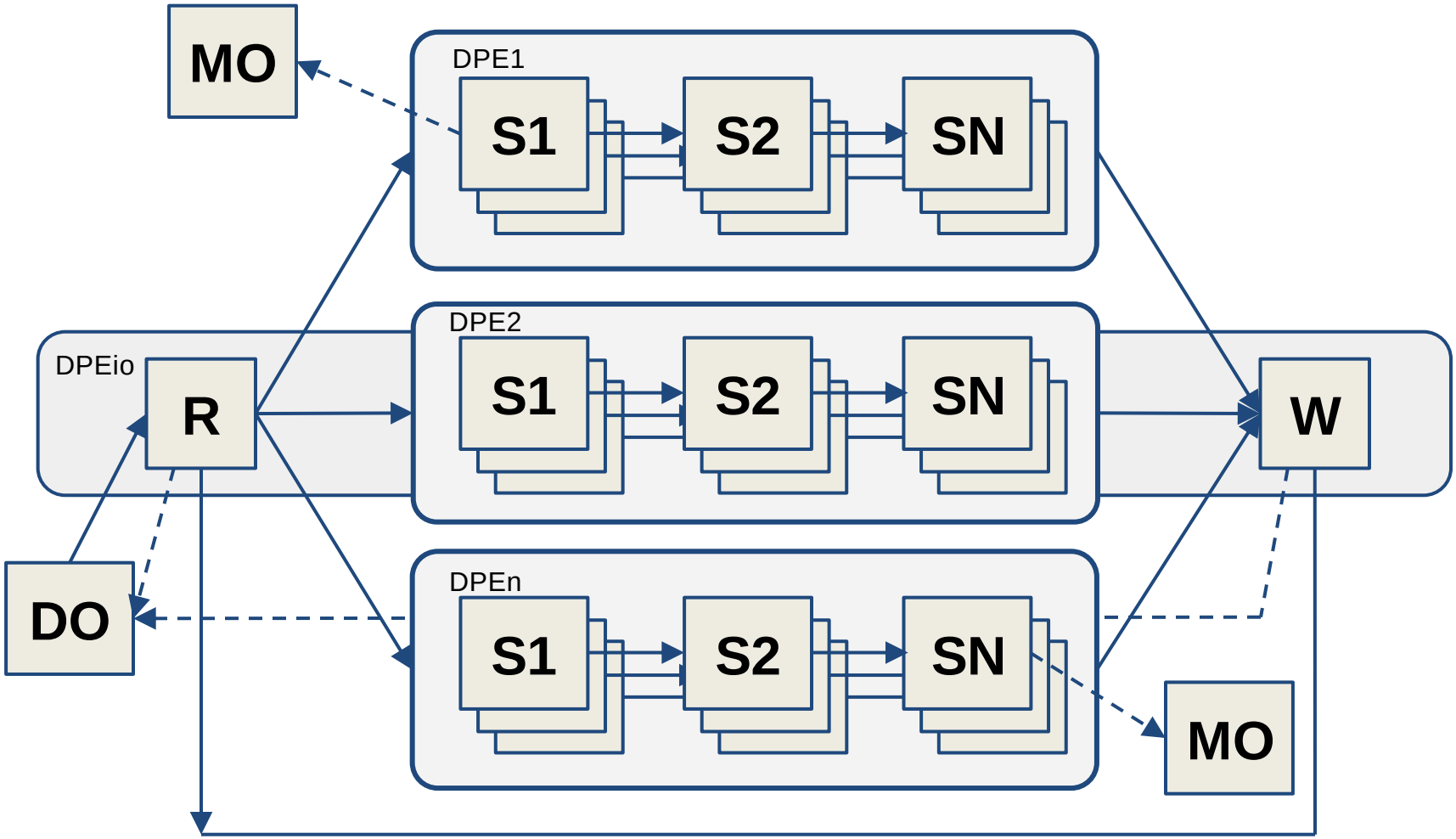
Multi-Core Reconstruction



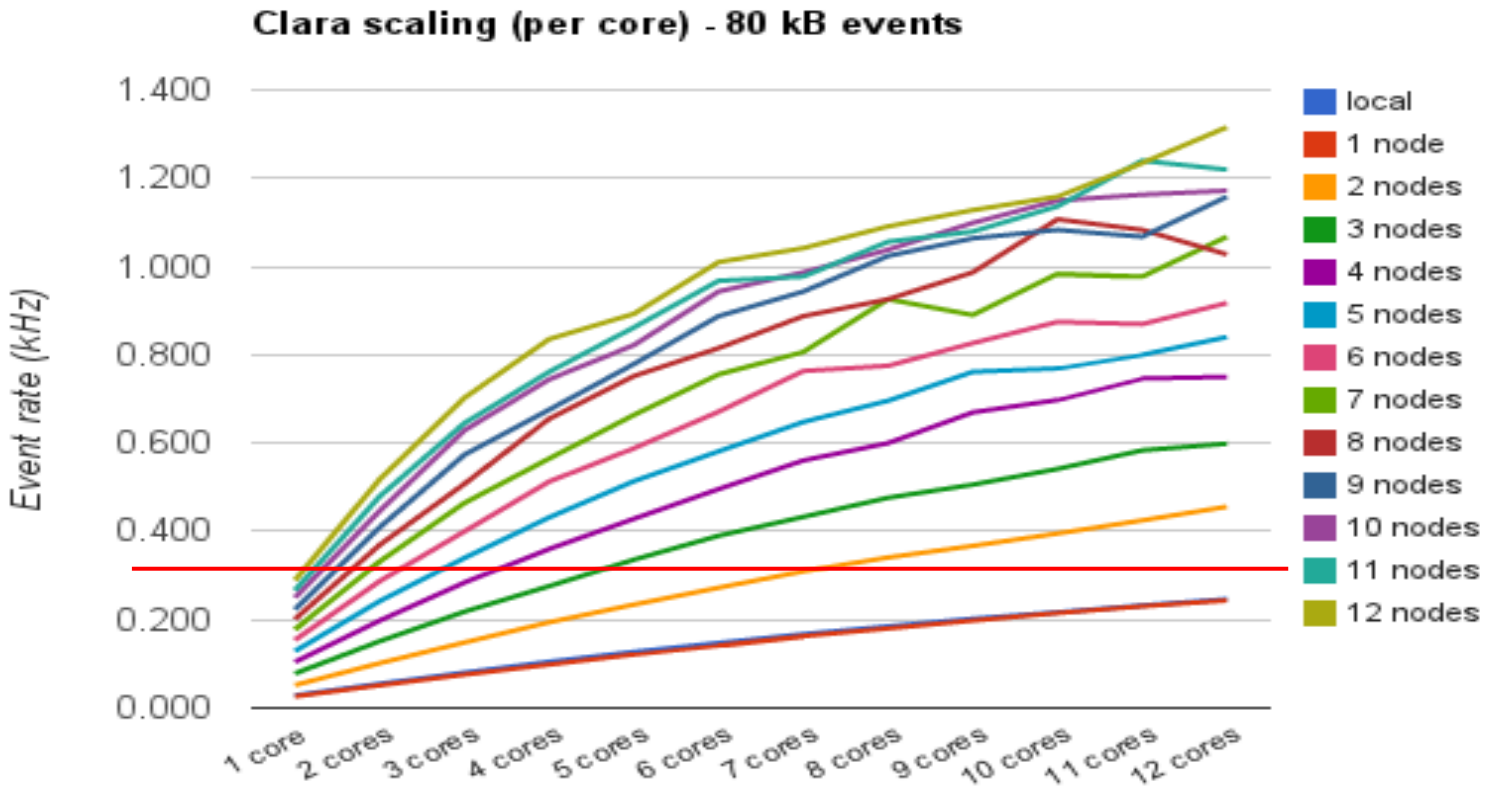
Multi-Core Reconstruction



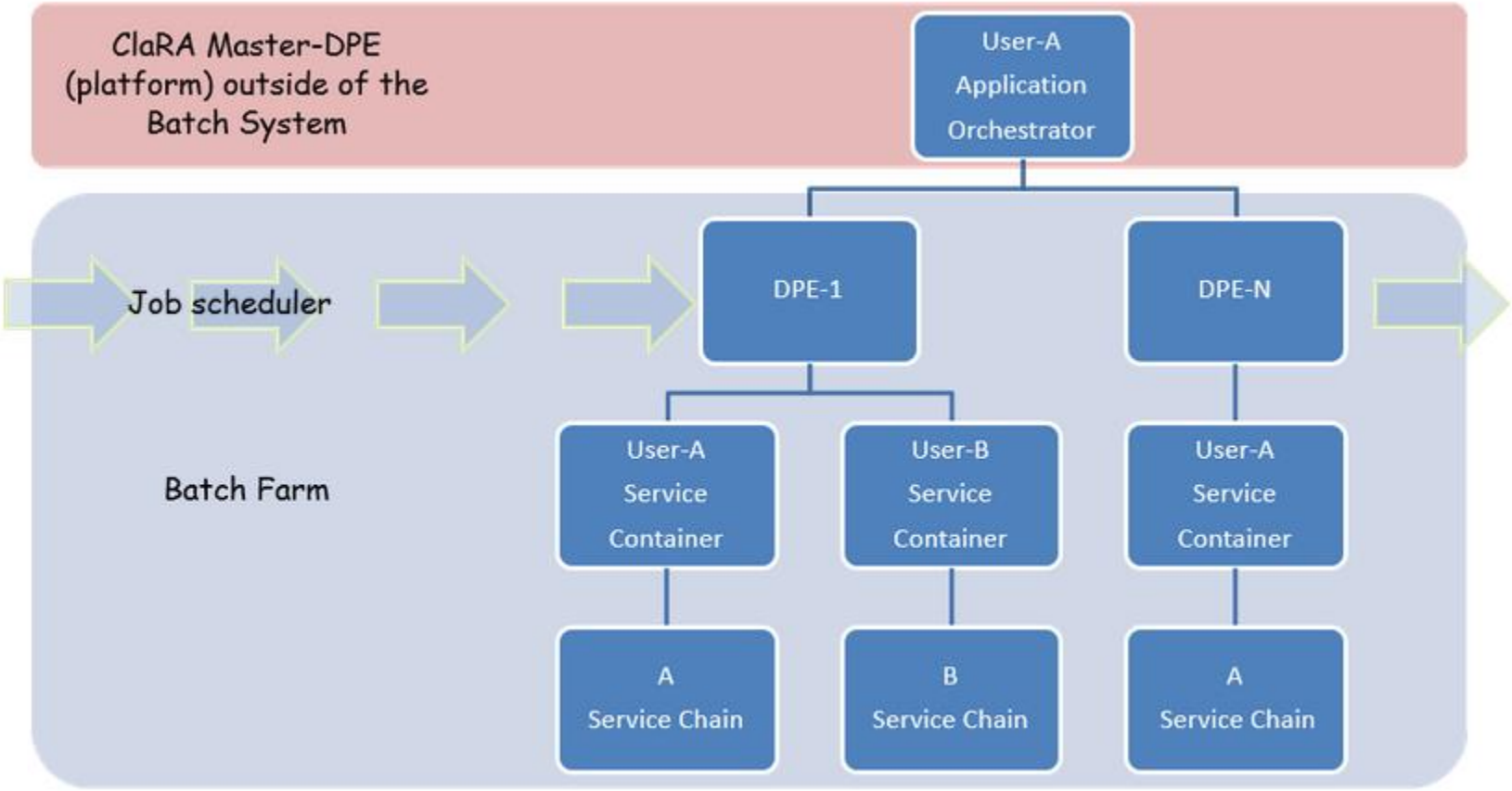
Multi-Node Reconstruction

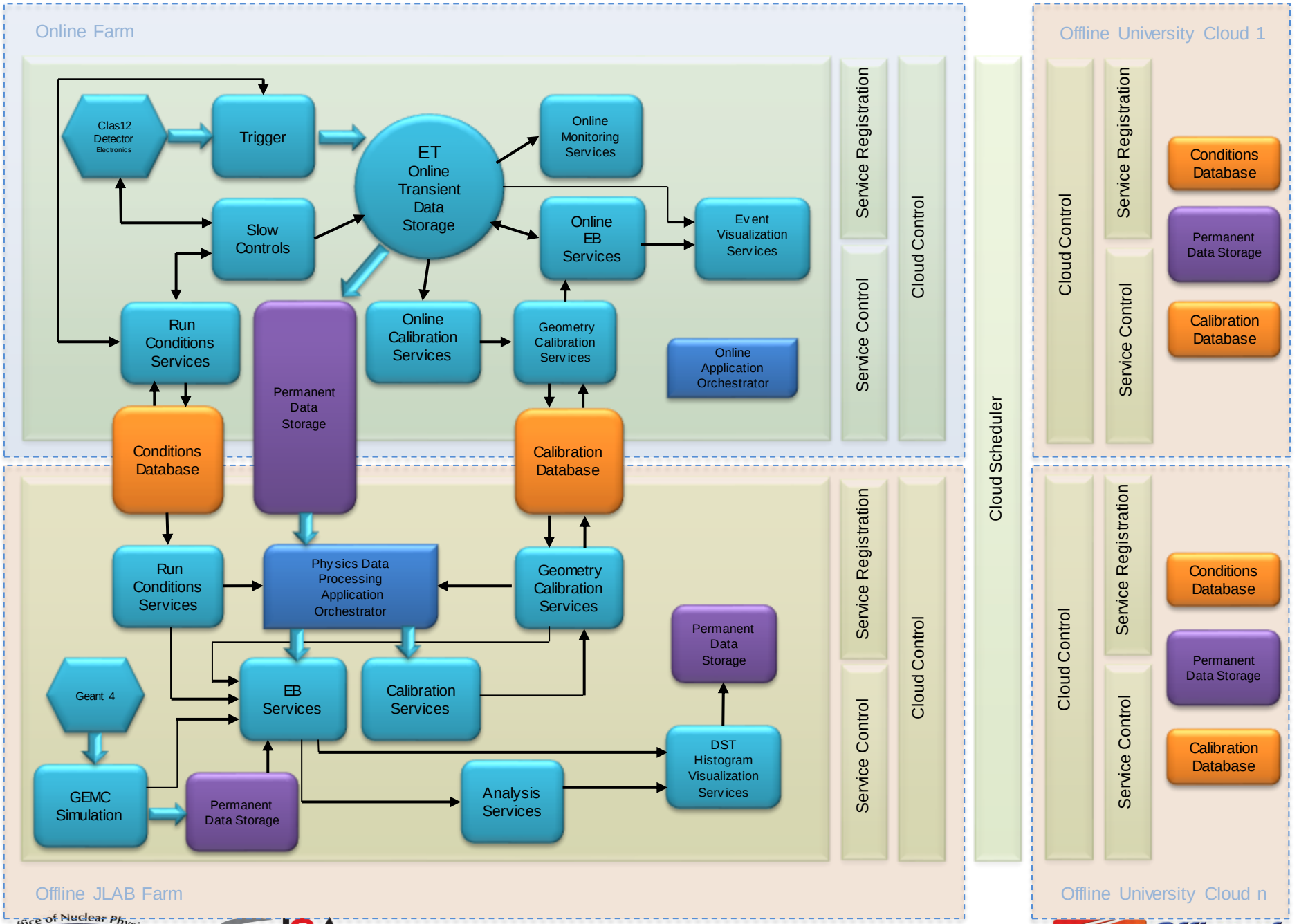


Multi-Node Reconstruction



Batch Deployment




Online Farm
Clas12 Detector
Electronics
Trigger
ET Online Transient Data Storage
Online Monitoring Services
Slow Controls
Online EB Services
Event Visualization Services
Run Conditions Services
Permanent Data Storage
Online Calibration Services
Geometry Calibration Services
Online Application Orchestrator
Conditions Database
Calibration Database
Service Registration
Service Control
Cloud Control
Cloud Scheduler
Offline University Cloud 1
Cloud Control
Service Registration
Service Control
Conditions Database
Permanent Data Storage
Calibration Database
Offline JLAB Farm
Run Conditions Services
Physics Data Processing Application Orchestrator
Geometry Calibration Services
Conditions Database
Permanent Data Storage
Calibration Database
Geant 4
EB Services
Calibration Services
Permanent Data Storage
GEMC Simulation
Permanent Data Storage
Analysis Services
DST Histogram Visualization Services
Service Registration
Service Control
Cloud Control
Offline University Cloud n
Cloud Control
Service Registration
Service Control
Conditions Database
Permanent Data Storage
Calibration Database

Challenges

- Increases author and user pools.
  - Management and administration. Strict service canonization rules
- Workloads of different clients may overwhelm a single service.
  - Service and Cloud governance
- Network security
  - Client authentication and message encryption

Summary and Conclusion

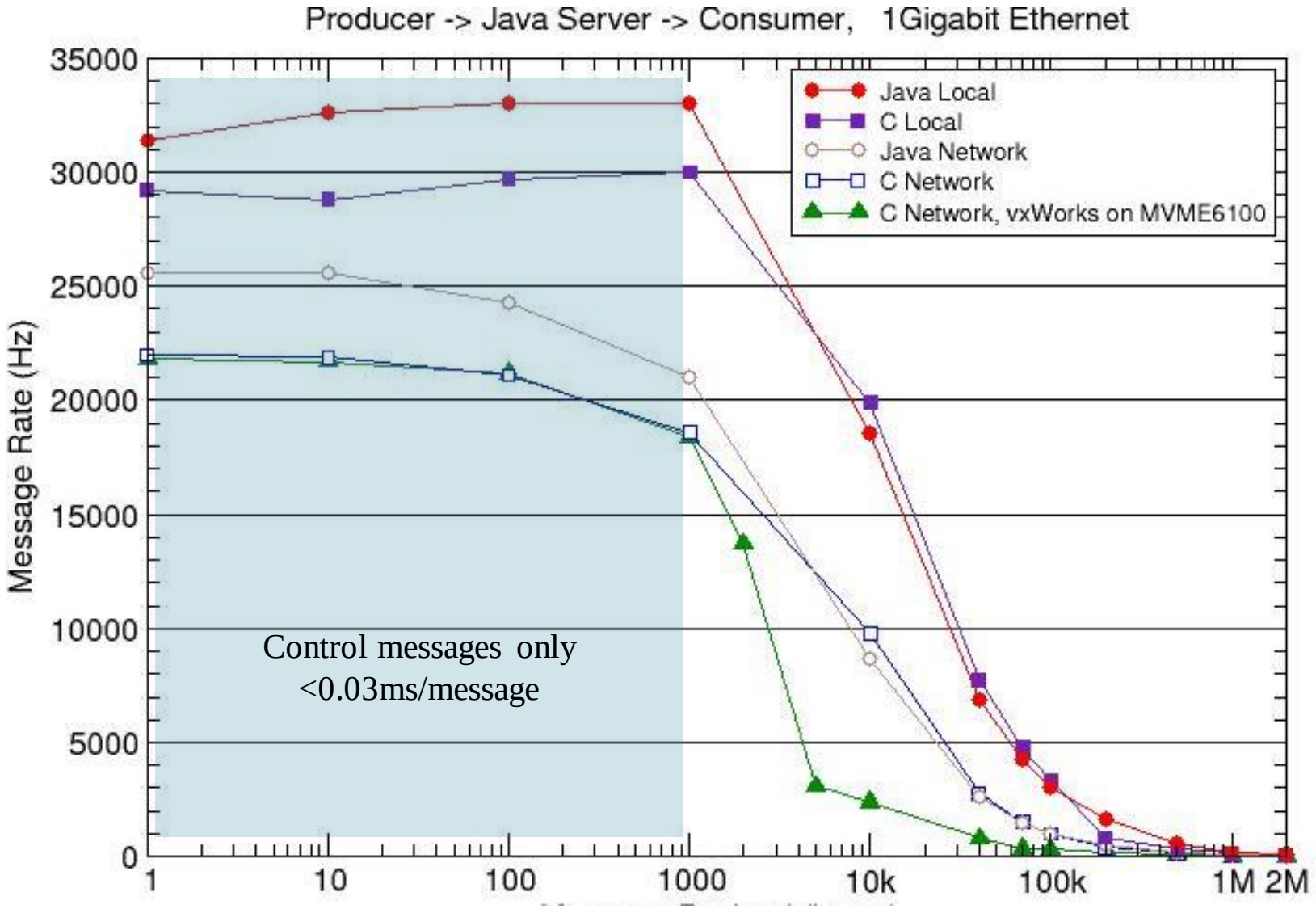
- A multi-treaded analyses framework, based on SOA
- PDP application based on specialized services
  - Small, independent
  - Easy to test, update and maintain
  - Building and running PDP application does not require CS skills.
- List of applications has been developed using the framework
  - Charge particle tracking (central, forward)
  - EC reconstruction
  - FTOF reconstruction
  - PID
  - HTCC
  - PCAL
  - Detector calibration
  - Event Building
  - Histogram services
  - Database application
    - Geometry, calibration constants and run conditions services

ClaRA supports both traditional and cloud computing models and if need be we are ready for cloud deployment.

Links

- https://clasweb.jlab.org/wiki/index.php/CLAS12_Software
- https://clasweb.jlab.org/wiki/index.php/CLARA
- https://clas12svn.jlab.org/repos/

Service Bus Performance measurements



Service Bus Performance measurements