PERLE Progress Review (PPR-3)
PERLE Command Control Architecture

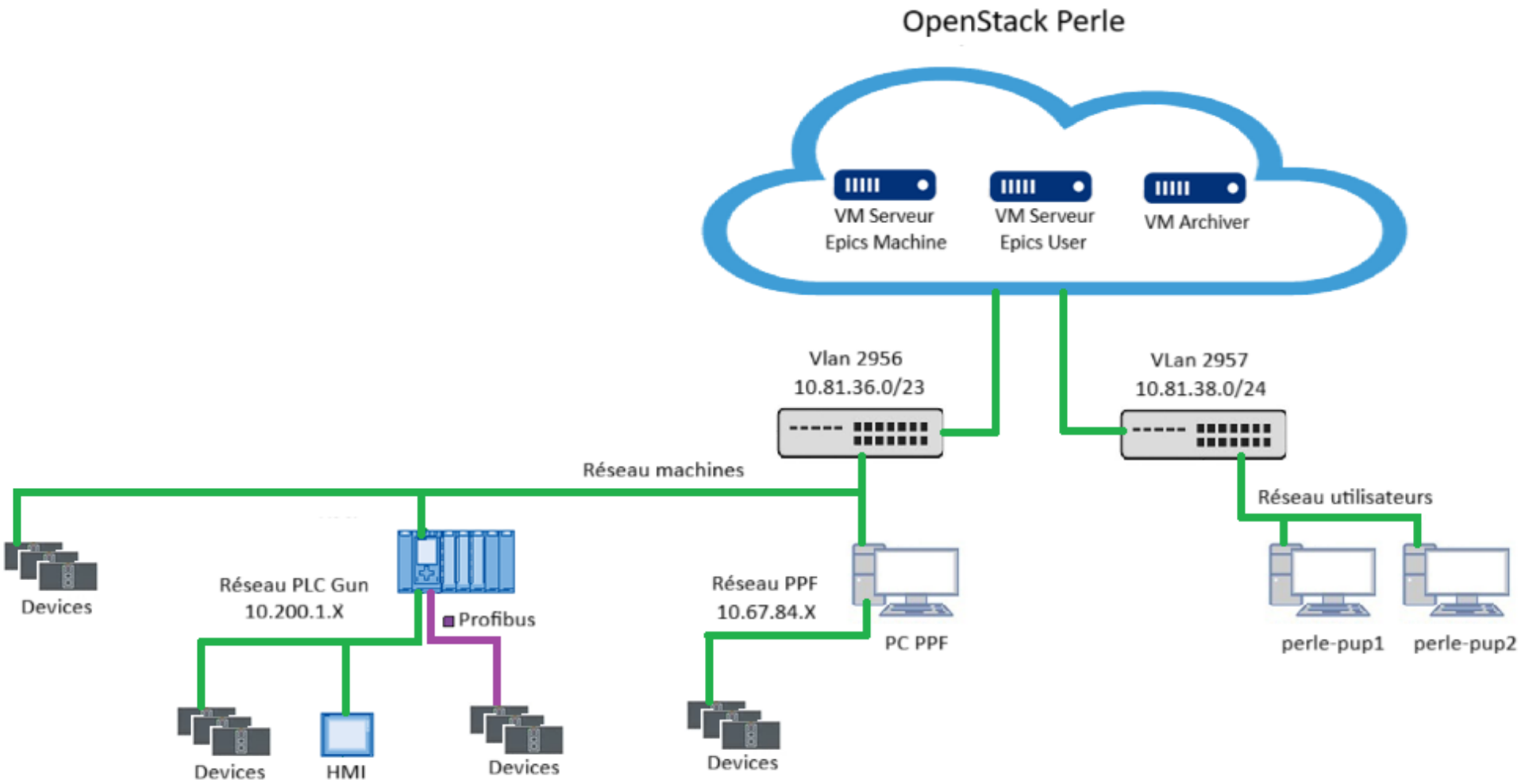
https://indico.ijclab.in2p3.fr/event/13822/

Vincent LAFAGE

29/05/2026

Point sur l'avancée

Architecture en place (y compris le lien avec VirtualData)



merci O. Dalifard, N. Barre, P. Lejeannic, G. Philippon, G. Marchal

Point sur l'avancée

Archivage

- Accueil d'un stagiaire de BUT (Clément Krawiec)
- ⇒ *Spring EPICS collaboration meeting* 2026
  - 20–24 avr. 2026
  - `https://indico.in2p3.fr/event/37441/`
  - tout le groupe contrôle-commande

FONCTIONNEMENT (1/2)

L'innovation principale de l'Archiver Appliance : les données ne vont pas directement sur un grand disque dur. Elles passent par trois étages :

- **STS** (Short Term Store)
  - Sur de la RAM ou SSD ultra-rapide
  - Garde quelques heures de données
  - C'est là que les nouvelles mesures arrivent d'abord

- **MTS** (Medium Term Store)
  - Sur SSD classique
  - Garde quelques semaines/mois

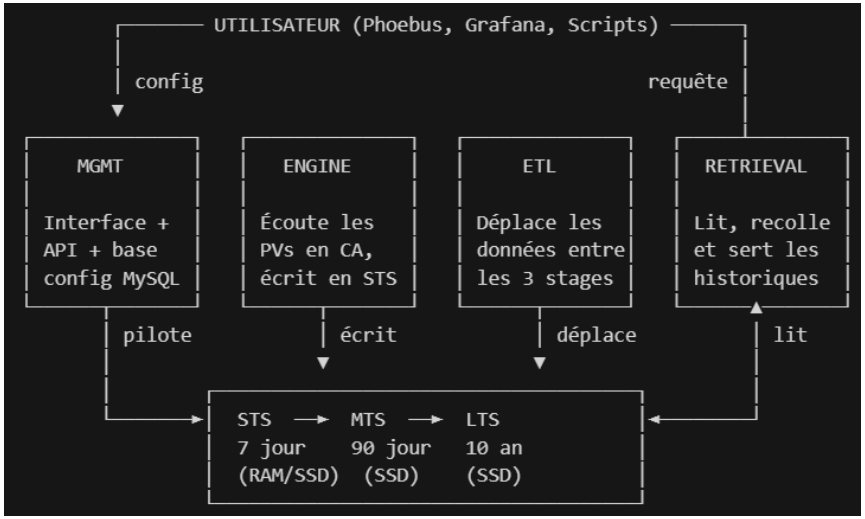
- **LTS** (Long Term Store)
  - Sur disque dur classique ou stockage réseau
  - Garde des années de données

Un mécanisme nommé ETL déplace et agrège automatiquement les données d'un étage à l'autre en arrière-plan.

FONCTIONNEMENT (2/2)

L'Archiver Appliance est une application Java déployée dans un serveur Apache Tomcat. Elle s'articule autour de 4 services qui collaborent avec un rôle précis :

- **MGMT** — Interface web, gestion des PVs à archiver, communication avec l'utilisateur
- **ENGINE** — Écoute en permanence tous les capteurs sur le réseau EPICS et enregistre les nouvelles valeurs
- **ETL** — Déplace les données entre les trois niveaux de stockage
- **RETRIEVAL** — Quand quelqu'un veut consulter des données historiques, c'est lui qui va les chercher dans les trois étages et les renvoie

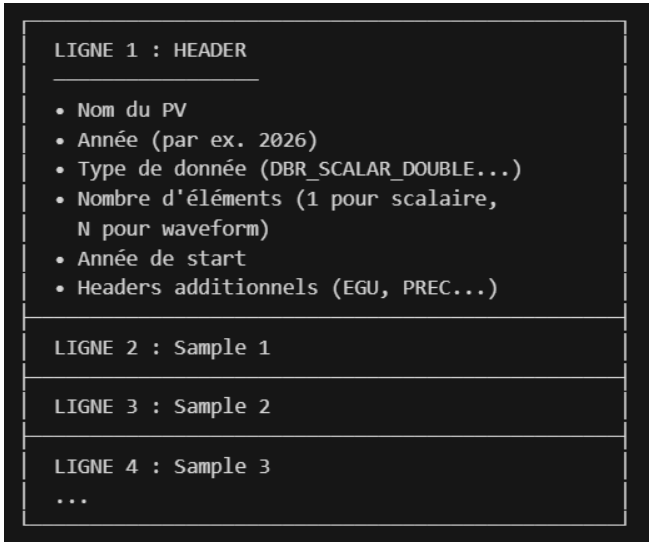


STOCKAGE DE LA DONNÉE

Pour stocker les mesures sur disque, l'archiver utilise Protocol Buffers, un format de sérialisation créé par Google en 2008 qui permet de convertir les données en flux d'octets binaires optimisés.

Pour une mesure de type ScalarDouble, voici les 6 champs stockés pour chaque sample :

- **secondsintoyear** — Le timestamp en secondes depuis le 1er janvier de l'année
- **nanos** — Précision sous-seconde, en nanosecondes (0 à 999 999 999)
- **val** — La valeur elle-même
- **severity** — Niveau d'alarme (0 = OK, 1 = MINOR, 2 = MAJOR, 3 = INVALID)
- **status** — Code d'état EPICS (HIHI, LOW, COMM, etc.)
- **repeatcount** — on stocke un seul sample avec ce champ pour N répétitions consécutives



DÉPLOIEMENT POUR PERLE

Le déploiement est conteneurisé avec Docker, ce qui permet de :

- Répliquer l'entièreté du déploiement en moins de 3 minutes via docker compose up
- Configurer facilement et avant installation l'archiver dans le docker-compose.yml
- Isoler totalement les dépendances du système hôte (Java, Tomcat, MySQL, Grafana)

Utilisation d'un volume monté pour la persistance et la sécurité des données (actuellement 800 Gio) :

- /data/archiver → pour sauvegarder les fichiers .pb
- /data/grafana → pour sauvegarder les configurations des éléments de grafana

Le catalogue des PVs à archiver est versionné en JSON et importé automatiquement au démarrage, garantissant la reproductibilité de la configuration et évitant toute saisie manuelle.


```
.
├── .env                          # Configuration (à créer depuis .env.example)
├── .env.example
├── .gitignore
├── docker-compose.yml
├── archiver/
│   ├── Dockerfile                # Image Tomcat 10 + Archiver Appliance 2.3.1
│   ├── archiver-entrypoint.sh    # Génère appliance.xml, attend l'API, importe les PVs
│   └── pvs/
│       └── generated-catalog.json  # Liste des PVs à archiver
├── mysql/
│   └── init/
│       └── 01-schema.sql         # Création des tables au premier démarrage
├── grafana/
│   └── provisioning/
│       └── datasources/
│           └── archiver.yaml     # Datasource Archiver Appliance pré-configurée
└── data/                         # Créé au runtime (mysql/)
```

TACHES EFFECTUÉES

Mise en place de l'infrastructure :

- Création de 2 VMs sur OpenStack pour héberger les services :
  - perle-ioc → hébergement des IOC PLCGUN, LASER et l'IOC intermédiaire Libera
  - perle-archiver → hébergement de l'archiver appliance, MySQL et Grafana
- Résolution des problèmes réseau pour permettre la communication Channel Access entre les serveurs

Conteneurisation de l'archiver appliance :

- Création d'un Dockerfile embarquant l'archiver appliance avec son serveur Tomcat
- Mise en place d'un archiver-entrypoint.sh automatisant au démarrage :
  - La génération du fichier appliance.xml
  - La création des volumes de stockage (STS / MTS / LTS)
  - L'import du catalogue de PVs via l'API REST de l'archiver
- Approche Infrastructure as Code : tout le déploiement est versionné et reproductible par un simple docker compose up

Visualisation et exploitation :

- Intégration de Grafana dans la même stack Docker
- Installation du plugin Archiver Appliance (Sasaki) pour récupérer les données archivées
- Création possible de dashboards pour visualiser plusieurs données simultanément : PLCGUN, LASER et les BPM Libera

IJCLab
Irène Joliot-Curie
Laboratoire de Physique des 2 Infinis

DÉFIS A VENIR

Espace de stockage :

- Optimiser l'occupation disque (actuellement aucune compression côté archiver appliance)
  - Gain attendu : 60 à 80 % de réduction sur des données EPICS typiques

Mise en place d'un système d'alarme :

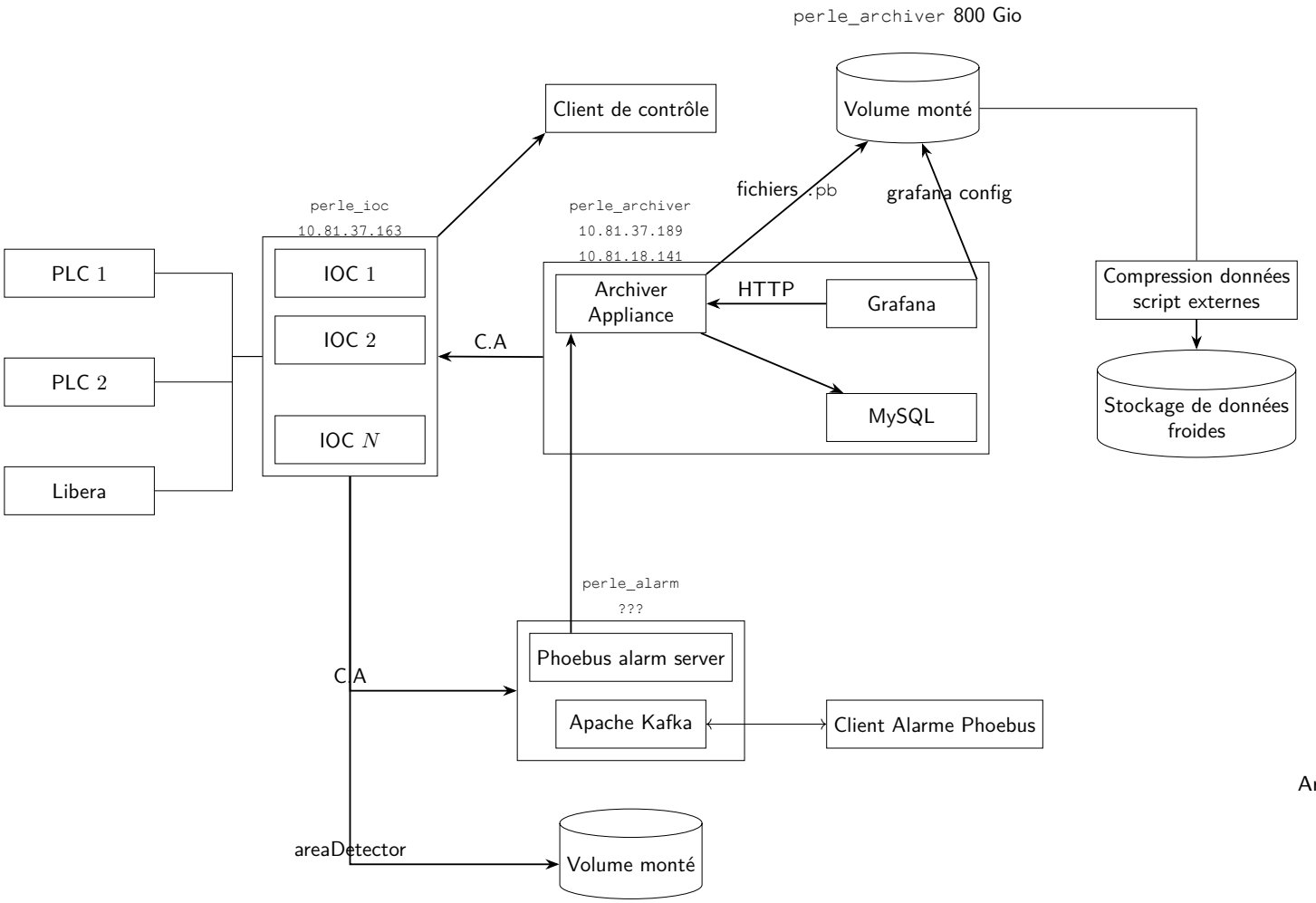
- Déployer Alarm Phoebus Server (stack lourde : Kafka + Phoebus alarm server)
- Gestion centralisée des alarmes actives, historique, acquittement et notifications

Stockage des images caméra :

- L'archiver appliance n'est pas adapté au stockage d'images (waveforms volumineuses, débit élevé)
- Besoin d'une solution dédiée : HDF5 ou autre
- À étudier : intégration avec areaDetector (standard EPICS pour l'acquisition d'images)
- Volume estimé à anticiper selon résolution × FPS × nombre de caméras

IJCLab
Irène Joliot-Curie
Laboratoire de Physique des 2 Infinis

Point sur l'avancée Architecture



Stockage des images en HDF5

Architecture comprenant :

- Gestion / archivage des alarmes
- Archivages des données de l'expérience
- Archivage des images caméras

Cette architecture, déjà complexe, constitue la version la plus légère permise par le standard EPICS pour répondre à ces besoins.

AURA : BENCHMARK

Benchmark AURA vs EPICS Archiver Appliance

Comparaison sur matériel équivalent (laptop bureau, 8 cœurs, 16 Go RAM)

Métrique	EPICS Archiver Appliance	AURA (Rust)	Gain
Démarrage + abonnement 30 000 PVs	~5-7 minutes	**5 secondes**	**98,33 % (× 60)**
Performance max sur 1 PV	~100 Hz	**4 000 Hz**	**3 900 % (× 40)**
Throughput soutenu	~50 000 events/s (estimé)	**200 000 events/s**	**300 % (× 4)**
Stockage & requêtage	Fichiers .pb plats, API REST propriétaire	**TimescaleDB + SQL natif**	**Standard universel**
Compression données	Aucune (21 o/sample)	**Colonnaire auto (2-4 o/sample)**	**85 % (×5 à ×10)**

High Performance · Memory-Safe · Event-Driven

IJCLab Irène Joliot-Curie Laboratoire de Physique des 2 Infinis

Point sur l'avancée

Archivage

- L'architecture est bien avancée
- ⇒ rédaction du TDR
  - Formation EPICS
  - début septembre
  - tout le groupe contrôle-commande
- ⇒ RH souhaités pour la suite (nouveau système d'archivage, alarmes, aspect caméras)