

# Local Deployment Emulation and Enhanced Testing for the HSF Conditions Database

**Fellow:** Sambridhi Deo([deosambridhi@gmail.com](mailto:deosambridhi@gmail.com))

**Project Mentors:** Lino Gerlach ([lino.oscar.gerlach@cern.ch](mailto:lino.oscar.gerlach@cern.ch))

**Project Duration:** 12 weeks(Dates to be confirmed)

## Project Description

High-energy physics (HEP) experiments require access to calibration and other non-event data during event data processing. These data are referred to as Conditions data i.e. any additional data needed to process event data from particle physics experiments. Conditions data are heterogeneous, change over time and are often accessed at very high frequencies.

The HSF Conditions Database strives to serve as an experiment-agnostic implementation that can be adopted by various HEP experiments. It is an open source software project that consists of a server- and a client-side component, the prior can be deployed on a kubernetes cluster for performance scaling.

**nopayloaddb:** This is the server-side component implemented as a Django REST API application. It handles metadata queries and provides information about the locations which is a unique identifier of the payload data, but does not store the actual payload data itself.

**nopayloadclient:** This is the client-side component implemented as a C++ library. It communicates with the nopayloaddb server through REST API. The nopayloadclient retrieves the payload locations from the server and is responsible for fetching the actual payload data from those locations. It also handles local caching of payload data and provides an interface for experiments to access and process the conditions data payloads.

## Objective

The goal of this IRIS-HEP Fellowship project is to enhance the functional tests that ensure the compatibility between the server- and client-side by implementing a local deployment emulation that is very similar to the environment in which the system is deployed in production. This can be achieved by running a kubernetes cluster inside a single container and allows to also test the deployment configuration locally. If time permits, the local deployment emulation will be used to develop and test different server-side caching strategies.

## Deliverables

After the completion of the project, the following deliverables will be publicly available:

1. **Local Deployment Emulation and Enhanced Testing for the HSF Conditions Database:** This can be, for example, a kubernetes cluster that can be run inside a single container and allows to

deploy the server-side component in a production-like setting locally. It should also be used for the automated functional tests.

2. **Documentation:** We aim to provide comprehensive documentation for the local deployment emulation mentioned above.
3. **Summary Presentation:** Presentations will be given on the status of the work to the IRIS-HEP collaboration at various meetings as part of the IRIS-HEP Fellow program.

## Timeline

Weeks	Goals
1-3	<ul style="list-style-type: none"><li>● Run both client and server components locally</li><li>● Literature review, evaluate existing tools for local kubernetes deployment</li></ul>
4-6	<ul style="list-style-type: none"><li>● Develop first prototype of local kubernetes deployment of the server component<ul style="list-style-type: none"><li>○ Decide on a tool for local kubernetes deployment (e.g. minikube or kind)</li><li>○ Run ‘empty’ kubernetes cluster locally as minimal example</li><li>○ Deploy <b>nopayloaddb</b> on that cluster</li><li>○ Manually run some functional tests to verify the deployment</li></ul></li></ul>
7-8	<ul style="list-style-type: none"><li>● Iterate &amp; refine implementation based on mentor feedback</li></ul>
9-10	<ul style="list-style-type: none"><li>● Write documentation &amp; include local kubernetes deployment in automated tests</li></ul>
11-12	<ul style="list-style-type: none"><li>● Prepare final presentation</li></ul>