The AIDA-2020 Advanced European Infrastructures for Detectors at Accelerators project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement no. 654168.

This work is part of AIDA-2020 Work Package 3: Advanced software.

The electronic version of this AIDA-2020 Publication is available via the AIDA-2020 web site <http://aida2020.web.cern.ch> or on the CERN Document Server at the following URL: <http://cds.cern.ch/search?p=AIDA-2020-MS19>

Copyright © CERN for the benefit of the AIDA-2020 Consortium
MILESTONE REPORT

DESIGN DOCUMENT FOR EVENT DATA MODEL TOOLKIT

MILESTONE: MS19

Document identifier: AIDA-2020-MS19
Due date of deliverable: End of Month 14 (June 2016)
Report release date: 02/09/2016
Work package: WP3: Advanced Software
Lead beneficiary: DESY
Document status: Final

Abstract:
The goal of the MS19 milestone is a design document for an event data model toolkit. The design document has been created, together with a first implementation in form of the PODIO C++ library. This Milestone Report describes the design work and work on the implementation that was performed to this end thus far. In addition, it summarizes first test-user experience.
AIDA-2020 Consortium, 2016
For more information on AIDA-2020, its partners and contributors please see www.cern.ch/AIDA2020

The Advanced European Infrastructures for Detectors at Accelerators (AIDA-2020) project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement no. 654168. AIDA-2020 began in May 2015 and will run for 4 years.

### Delivery Slip

<table>
<thead>
<tr>
<th>Name</th>
<th>Partner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authored by</strong></td>
<td>B.Hegner</td>
<td>CERN</td>
</tr>
<tr>
<td><strong>Edited by</strong></td>
<td>F.Gaede</td>
<td>DESY</td>
</tr>
<tr>
<td><strong>Reviewed by</strong></td>
<td>F.Gaede [WP coordinator]</td>
<td>DESY</td>
</tr>
<tr>
<td><strong>Approved by</strong></td>
<td>Scientific Coordinator</td>
<td></td>
</tr>
</tbody>
</table>

Approved by Scientific Coordinator 02/09/16
# TABLE OF CONTENTS

1. INTRODUCTION .......................................................................................................................... 4
2. DESIGN AND IMPLEMENTATION ............................................................................................... 5
3. REFERENCES .................................................................................................................................. 6
ANNEX: GLOSSARY ............................................................................................................................ 6
Executive summary

To allow the efficient storage and reading of high-energy physics event data the current implementations of event data models need to be reworked. After evaluating existing implementations, a data model design was created and tested in form of a C++ implementation. The design and the implementation have been presented and documented and are now being evaluated by various potential clients in the high-energy-physics community.

1. INTRODUCTION

Efficient storage and retrieval of experiment data is one of the key concerns of high energy physics data analysis. In addition to hardware resources, the software used for the storage needs to be highly efficient.

Based on the experience of first and second generation object-oriented data models and I/O libraries, a reworked design and approach for storing event data is being prepared. It takes the move to increased hardware parallelization in form of many-core processors and vectorization into account, as well as effects of memory optimization and memory access latencies.

One of the key concerns of current data model libraries and toolkits is the high abstraction level and subsequent complexity of implementations. This makes retrofitting new approaches for memory cache friendly designs very hard, if not impossible.

The present milestone addresses the creation of a design for a new event data model toolkit that takes changes in the computer hardware into account while supporting the already existing use cases of data analysis at the Large-Hadron-Collider (LHC) or for studies of future accelerators and experiments.
2. DESIGN AND IMPLEMENTATION

After an analysis of existing data model libraries from the experiments at the LHC and the efforts of the linear collider software community, an initial design for an experiment independent data model library was created. Based on the idea of employing simpler data structures, so-called PODs, a potential design has been discussed with the key stakeholders of the HEP software community and the status of the then-created PODIO software project \cite{1,2} presented at various occasions \cite{3-5}. A document describing its design and documentation was created subsequently \cite{6}.

PODIO, or plain-old-data I/O, is a C++ library to support the creation and handling of data models in particle physics. It is based on the idea of employing plain-old-data (POD) data structures wherever possible, while avoiding deep-object hierarchies and virtual inheritance. This is to both improve runtime performance and simplify the implementation of persistency services. At the same time it provides the necessary high-level functionality to the physicist, such as support for inter-object relations, and automatic memory-management. In addition, it provides a (ROOT assisted) Python interface. To simplify the creation of efficient data models, PODIO employs code generation from simple YAML-based markup syntax. To support the usage of modern software technologies, PODIO was written with concurrency in mind and gives basic support for vectorization technologies. This document describes first how to define and create a specific data model, then how to use the created data. Afterwards it will explain the overall design and a few of the technical details of the implementation. Many of the design choices are inspired by previous experience of the LCIO package used for the studies of the international linear collider, and the Gaudi Object Description applied in the LHCb collaboration at the LHC.

Currently the first version of PODIO is being used as underlying data model library for the studies of the future circular collider (FCC) \cite{7,8}.

There are ongoing efforts with the linear collider software project on whether PODIO could become the next evolutionary step of the rather successful LCIO package. For this, a few additional features will need to be added to PODIO. In addition, the upgrade plans of the LHCb experiment include the evaluation of PODIO as a library for their event data storage. In particular, more realistic and thorough tests of PODIO’s runtime performance will be carried out.
3. REFERENCES


ANNEX: GLOSSARY

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PODIO</td>
<td>Plain-Old-Data I/O Library</td>
</tr>
<tr>
<td>FCC</td>
<td>Future Circular Collider</td>
</tr>
<tr>
<td>YAML</td>
<td>Yet Another Markup Language</td>
</tr>
</tbody>
</table>