Monitoring of CERN Data Interchange Protocol System

Brice Copy, E. Mandilara, I. Prieto Barreiro, F. Varela Rodriguez

BE Beams Department

THPHA162

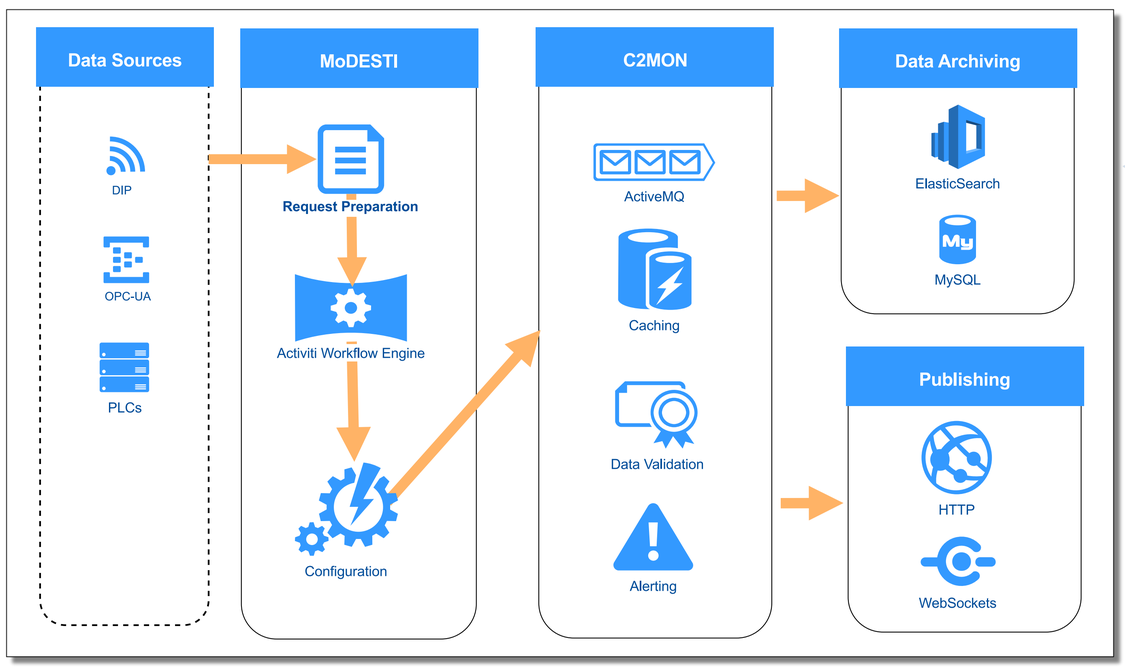
Abstract

CERN's Data Interchange Protocol (DIP) [1] is a publish-subscribe middleware infrastructure developed at CERN to allow lightweight communications between distinct industrial control systems (such as detector control systems or gas control systems).
DIP is a rudimentary data exchange protocol with a very flat and short learning curve and a stable specification. It also lacks support for access control, smoothing or data archiving.This paper presents a mechanism which has been implemented to keep track of every single publisher or subscriber node active in the DIP infrastructure, along with the DIP name servers supporting it. Since DIP supports more than 55,000 publications, regrouping hundreds of industrial control processes, keeping track of the system activity requires advanced visualization mechanisms (e.g. connectivity maps, live historical charts) and a scalable web-based interface to render this information is essential.

Objective

Allow data publishers and subscribers to agree formally on data formats and naming, by integrating DIP data publication definitions into the MoDESTI workflow engine.

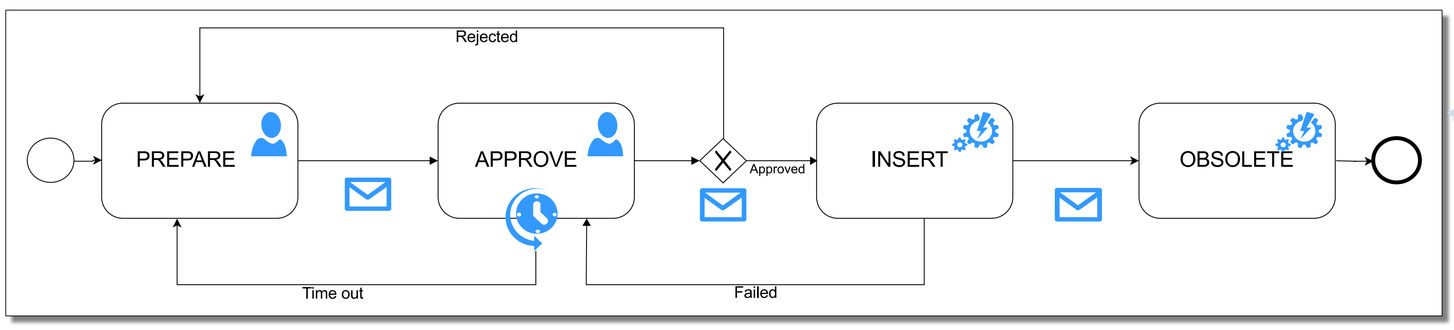
MoDESTI and C2MON Integration



To address different use cases, MoDESTI supports modular plug-ins. Each MoDESTI plug-in contains a schema describing the data domain and its constraints, and also provides a Business Process Model and Notation (BPMN) workflow specification.
In the case of DIPCM, a dedicated plug-in was created in order to cover the application's workflow as described earlier.

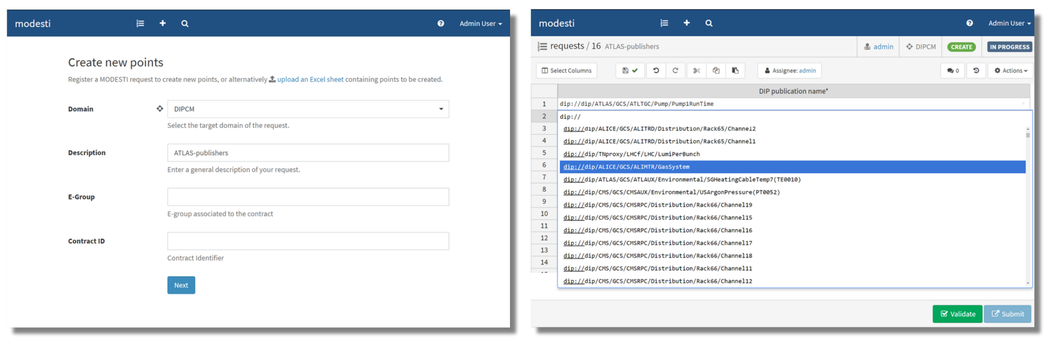
C2MON has a persistent storage mechanism implemented with Elasticsearch, that is used to store alarms or other data that are wanted to be monitored, under the interface of a data tag. The data tag is a C2MON Tag coming from an external data source, while a C2MON Tag is an object destined to be updated, logged, published to clients and used in data views. It corresponds to a single external data point, represented by a Java primitive type (String, Float, Integer, etc.). Every DataTag has to provide a so-called "hardware address", which contains the information needed for subscribing to this data point. After this subscription, and with the necessary Data Acquisition (DAQ) module in function, updates about data changes can be coming in and stored into Elasticsearch.

DIP Contract Monitoring Workflow



DIP Contract Monitoring relies on a simple workflow that backs the application's front-end user interface through the same aforementioned three basic stages: registration, approval, and insertion into a C2MON configuration database.
The first stage is the **preparation** of a contract, which is done by visiting the DIP Contract Monitoring System web application. At the end of this procedure, the second stage, **approval**, is triggered.
At the beginning of this approval step, a notification is dispatched to the stakeholders that are responsible for the individual publications included to a contract. These stakeholders are invited to review the contract details in MoDESTI and approve or reject the requested contract. If the contract is rejected, its creator needs to edit the information they provided as per the approver's comments. If the request is approved, it progresses in the workflow to the final stage, which is the configuration of the contract into C2MON [4], which will allow it to be actively monitored.

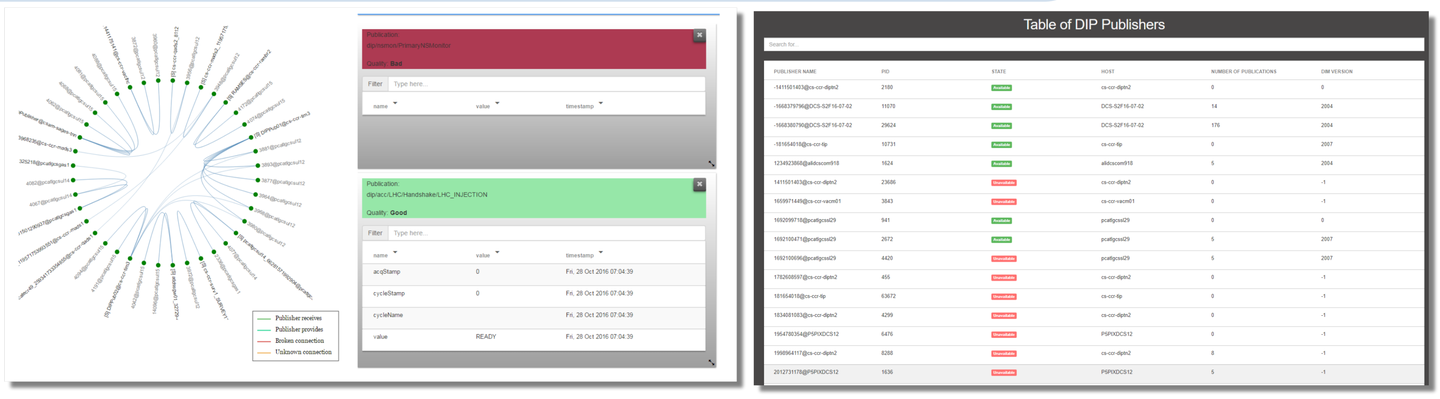
MoDESTI User Interface Integration



To address different use cases, MoDESTI supports modular plug-ins. Each MoDESTI plug-in contains a schema describing the data domain and its constraints, and also provides a Business Process Model and Notation (BPMN) workflow specification.
In the case of DIPCM, a dedicated plug-in was created in order to cover the application's workflow as described earlier.

C2MON has a persistent storage mechanism implemented with Elasticsearch, that is used to store alarms or other data that are wanted to be monitored, under the interface of a data tag. The **data tag** is a C2MON Tag coming from an external data source, while a C2MON Tag is an object destined to be updated, logged, published to clients and used in data views. It corresponds to a single external data point, represented by a Java primitive type (String, Float, Integer, etc.).

DIP Infrastructure Monitoring Web Interface



Since DIP enables the real-time communication between heterogeneous systems, it is being widely used and the demand for more data exchange is continuously growing. At the same time, the state and flow of information is often ignored while they are not negligible. Information exchanges therefore need to be monitored so that, for instance, invalid publications can be detected, invalid name formats can be reported upon and in general, to keep track of what information is being published.

For this reason, four web-based tools were implemented: DIP Web Browser, DIP Publishers Monitor, Connectivity Maps, DIPNS Monitor.

The front ends of all these tools were created making use of Polymer Web Components [7], which allows to create reusable widgets or components, and very simply reuse them in multiple web pages.

References

[1] W. Salter et al., "DIP Description" LDIWG (2004), https://cern.ch/dip/.
[2] C. Gaspar et al., "DIM, a portable, light weight package for information publishing, data transfer and inter-process communication", Computer Physics Communications 140 1+2 102-9, 2001.
[3] R. Martini et al., "Tools and Procedures for High Quality Technical Infrastructure Monitoring reference Data at CERN", WEPGF141, Oct 2015 , ICALECPS'15, Melbourne, Australia.
[4] M. Braeger et al., "High availability monitoring and big data : using Java clustering and caching technologies to meet complex monitoring scenarios", MOPPC140, Oct 2013, ICALEPCS'13, San Francisco, USA.
[5] T. Rademakers, "Introducing the activiti framework", in Activiti in action, Shelter Island, NY, USA, Manning publications, 2012.
[6] C. Walls, "Deploying Spring Boot applications", in Spring Boot in action, Shelter Island, NY, USA, Manning publications, 2015.
[7] D. Glazkov and H. Ito, "Introduction to Webcomponents", 24 July 2014, http://www.w3.org/TR/components-intro/
[8] B. Copy et al., "Scalable web broadcasrting for historical industrial controls data", WEPGF042, Oct 2015, ICALECPS'15, Melbourne, Australia.
[9] S. Murray, "Introducing D3", in Interactive Data Visualizations for the web, Sebastopol, CA, O'Reilly Media, 2017.
[10] I. Hickson, "The WebSocket API", W3C Consortium, 20 September 2012, http://www.w3.org/TR/2012/CR-websockets-20120920.

Brice Copy, E. Mandilara, I. Prieto Barreiro, F. Varela Rodriguez
Email : brice.copy@cern.ch
CERN BE-ICS Group

ICALEPCS2017
Barcelona · Spain, October 8-13 · Palau de Congressos de Catalunya