



Event-driven Processing of Earth Observation Data

Matthes Rieke (m.rieke@52north.org)¹, Sebastian Drost², Simon Jirka¹, and Arne Vogt²

¹52°North GmbH, Münster, Germany

²Geodesy Department, University of Applied Science Bochum, Bochum, Germany

Earth Observation data has become available and obtainable in continuously increasing quality as well as spatial and temporal coverage. To deal with the massive amounts of data, the WaCoDiS project aims to develop an architecture that allows its automated processing. The project focuses on the development of innovative water management analytic services based on Earth Observation data such as provided by the Copernicus Sentinel missions.

Goals of the WaCoDiS Project

Water management associations currently face different challenges regarding the quality control of drinking water. More frequently, they have to deal with pollutant inputs into surface waters. Reasons for this are increased fertilization of agricultural areas and extreme weather events, which, in particular, lead to nitrate leaching. In order to develop case specific strategies for reducing pollutant inputs, the development of an improved water monitoring is needed. The broader availability of Earth Observation data as well as the evolution of different Copernicus satellite data platforms, such as the Copernicus Data and Exploitation Platform – Deutschland (CODE-DE), contribute to this task. Innovative analytic services can use those platforms' processing capabilities to operate on Big Earth Observation Data in order to provide different Earth Observation products for a hydrologic application context. WaCoDiS aims to develop such a monitoring system for the Wupper region in North-Rhine Westfalia, Germany. This is the project's area of study. The Wupperverband is the water authority responsible for a 813 km² catchment area and operates several river dams and sewage treatment plants. The goal is to improve hydrological models

including, but not limited to, a) identification of the catchment areas responsible for pollutant and sediment inputs, b) detection of turbidity sources in water bodies and rivers.

Automatic Processing Workflows for Earth Observation Data

Since most of the Wupperversband’s hydrologic expert users have only little or no knowledge of processing Earth Observation data, there is a substantial need for automatic processing workflows. Therefore the WaCoDiS System comprises dedicated components. Each one is responsible for a specific task within a processing workflow that automatically generates Earth Observation products. The entire processing workflow is depicted in Figure 1 and described as follows.

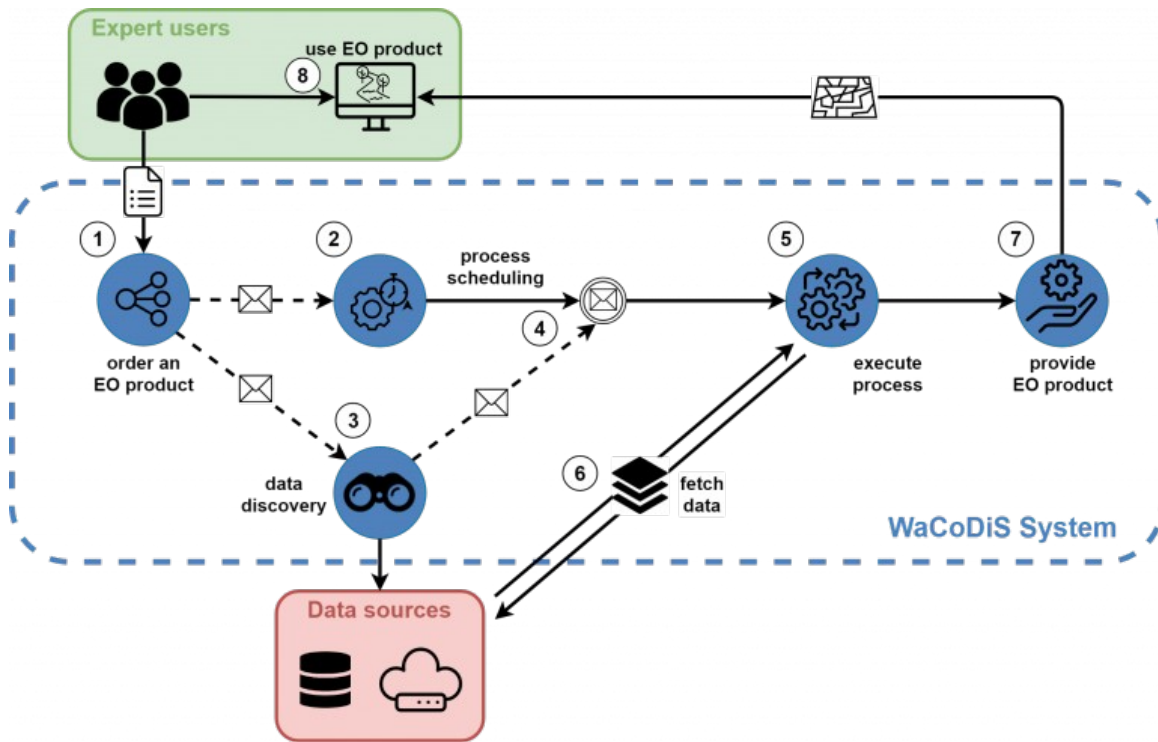


Figure 1: Processing workflow of the WaCoDiS System.

1. Domain expert users define the need for a certain EO product by specifying the product type and the spatio-temporal coverage for input datasets.
2. The WaCoDiS System takes the process description and starts scheduling the process execution.
3. At the same time, dedicated components discover external data sources for new datasets required and store their metadata.
4. As soon as the required datasets become available, the scheduling component is notified.
5. If all preconditions (required datasets are available and execution time is reached) are fulfilled, the process execution is started.

6. The datasets required are fetched from the data sources and provided as inputs to the EO processes that generate an EO product.
7. The EO product generated is disseminated by a standardized web service (e.g. a OGC Web Coverage Service) that is accessible within the domain user's system infrastructure.
8. Domain expert users can use the EO product e.g. within their own domain applications via web services.

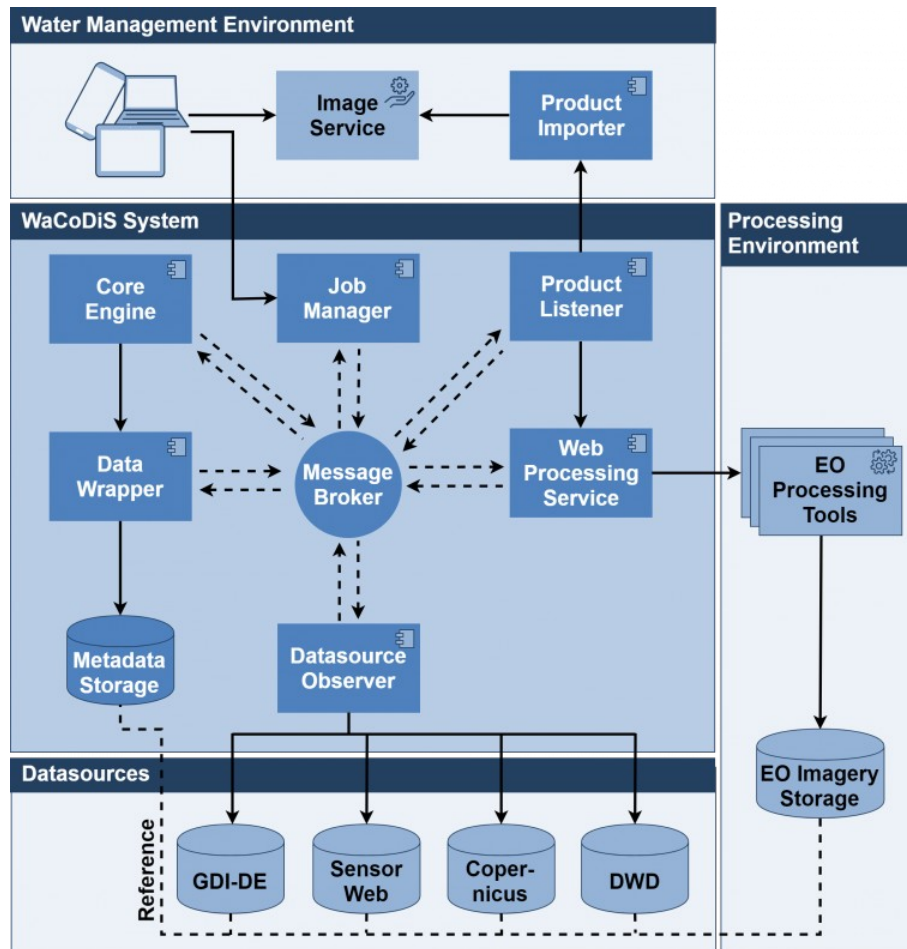


Figure 2: Microservice architecture of the WaCoDiS System.

The main benefit of the automated processing workflow is that a user must not have any knowledge of Earth Observation data processing. The user only needs to define what product type he demands for which area and temporal extent. The scheduling system handles this description by executing each single task so that the user receives the product needed at the end of the whole processing workflow. Thus, the system reduces manual work (e.g. identification of relevant input data, execution of algorithms) and minimizes the required interaction of domain users. In addition, once

a Processing Job is registered within the system, the user can track its status (e.g. when it was last executed, if an error occurred) and will eventually be informed when new processing results are available.

An Event-driven Microservice Design for the WaCoDiS System Architecture

In order to realize such an automatic processing workflow, we designed the system architecture according to the Microservice architecture pattern (Figure 2). Small components fulfill different tasks and responsibilities.

The following components play important roles:

- **Job Manager:**
The Job Manager enables system users to define a processing job, including the repetition rules, input data and the targeted Earth Observation product.
- **Datasource Observer:**
Based on the job definitions, this component observes the CODE-DE platform (or other equivalent data centers) to identify matching input data. The datasets found are referenced within the WaCoDiS System by storing their metadata within another component.
- **Core Engine:**
The Core Engine considers the repetition rules for processing jobs in order to schedule their execution. It checks fulfillment of input data requirements and executes the EO processing algorithms.
- **EO Processing Tools:**
The processing algorithms (e.g. landcover classification, forest vitality change, determination of chemical and physical parameters within water bodies) operate within the processing environment of satellite data platforms and generate the Earth Observation products.
- **Web Processing Service:**
An OGC Web Processing Service Interface encapsulates the processing algorithms. It enables an interoperable execution of these algorithms by means of describing the required process input parameters and outputs. It also supports the exploitation of several existing tools, such as the Sentinel Toolboxes or GDAL, to pre-process Earth Observation data.
- **Product Listener/Product Importer:**
A Product Listener waits for processing algorithms to finish and is notified if a new Earth Observation product is available. A Product Importer, which has access to the expert users' target system infrastructure, injects the final product within an Image Service.

The different components are loosely coupled and react to messages and events that are published on a central message broker component. This allows for a flexible scaling and deployment of the system. For example, the management components can run on different physical locations than the processing algorithms. New processing algorithms can be also easily integrated using the OGC Web Processing Service Interface.

Cloud deployment patterns

A first priority was the design of a component architecture to consider deployment in native cloud environments. The Microservice pattern serves this well and allows for elasticity and scalability aspects in the system. All components are available as Docker images and are configurable to meet the deployment environment's requirement. The different functional modules (e.g. persistence layer, scheduling, process execution) are fully decoupled. Therefore, the system can be deployed in Platform as a Service (PaaS) environments, such as the Copernicus DIAS nodes, or on Infrastructures as a Service (IaaS), e.g. Google Cloud Platform (GCP) or AWS (see Figure 3), as well as on a local or bare metal server. Our proof of concept comprised an installation on a vanilla Kubernetes cluster. During the remainder of the project, we will install the system on the CODE-DE platform (also considering the current re-launch and its native APIs) and on GCP premises. The Message Broker establishes all communication between the components, thus enabling their distributed deployment. This is of particular importance if the EO data processing should be executed near the data. For example, the data observation and scheduling components could run in a self-hosted environment while the EO processing tool components (i.e. WPS + tool backend) are deployed on e.g. CODE-DE.

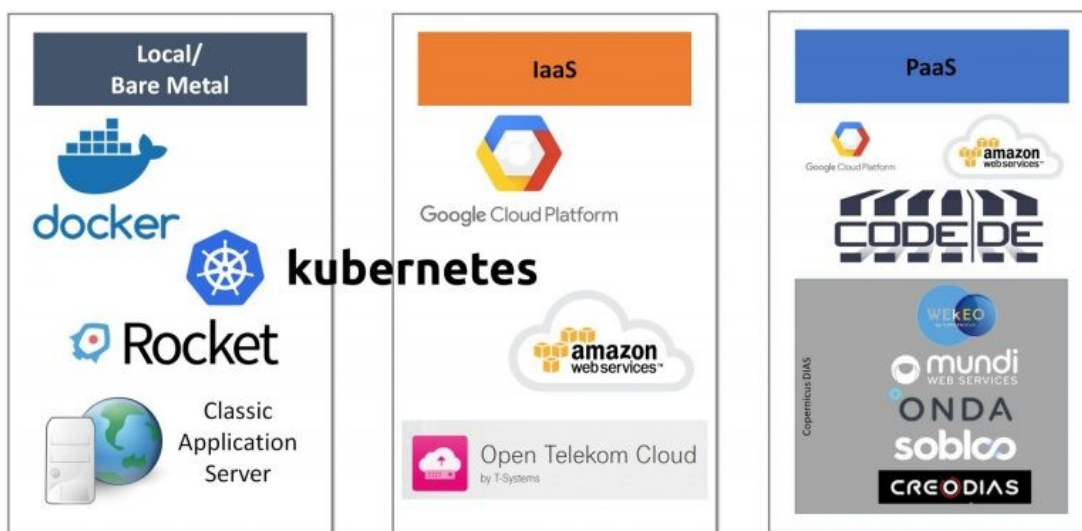


Figure 3: Deployment patterns for the WaCoDiS System.

In summary, this work targets the development of a system that allows the automated and event-driven creation of Earth Observation products. It is suitable to run on IaaS or PaaS (e.g. CODE-DE or Copernicus DIAS nodes) as well as dedicated environments such as Kubernetes clusters. Future work will focus on the system's usability aspects. A dedicated browser-based application will be developed for managing the different processing jobs. Several improvements in the component architecture (e.g. EO tool and product catalog) will support these developments.

This work has been funded by the Federal Ministry of Transport and Digital Infrastructure (Germany), BMVI, as part of the mFund programme.

