Horizon 2020 LC-SPACE-04-EO-2019-2020

Copernicus Evolution – Research for harmonised and Transitional-water Observation (CERTO)

Project Number: 870349

Deliverable No: D7.2		Work Package:	
Date:	10-DEC-2020	Contract delivery due date	31-DEC-2020
Title:	Solution Architecture Design Document		
Lead Partner for Deliverable	PML Applications		
Author(s):	Amy Westlake, Oliver Clements, Ben Calton		
Dissemination level (PU=public, RE=restricted, CO=confidential)			PU
Report Status (DR = Draft, FI = FINAL)			FI

Acknowledgements
This project has received funding from the European Union's Horizon 2020 research and innovation programme grant agreement N° 870349

Table of Contents

Reference Documents

RD1	CERTO Deliverable: D2.2 Technical requirements of the CERTO prototype
RD2	CERTO Deliverable: D7.1 Initial Report on DIAS & Cloud Provider Capabilities

Glossary

API	Application Programming Interface
C3S	Copernicus Climate Change Service
CMEMS	Copernicus Marine Environment Monitoring Service
CLMS	Copernicus Land Monitoring Service
CLI	Command Line Interface
DIAS	Data and Information Access Services
EO	Earth observation
FTP	File Transfer Protocol
HPC	High Performance Computing
OGC	Open Geospatial Consortium
OpenDAP	Open-source Project for a Network Data Access Protocol
SaaS	Software as a Service
SNAP	The Sentinel Application Platform
VM	Virtual Machine
WCS	Web Coverage Service
WMS	Web Map Service

1 Executive Summary

Water quality is a key worldwide issue relevant to human food consumption and production, industry, nature, and recreation. The European Copernicus programme includes satellite sensors designed to observe water quality and services to provide data and information to end-users in industry, policy, monitoring agencies, and science. However, water-quality data production is split across three services, Copernicus Marine, Copernicus Climate Change, and Copernicus Land, with different methods and approaches used and some areas, notably transitional waters, are not supported by any service.

The CERTO project aims to address this lack of harmonisation by undertaking research and development necessary to produce harmonised water-quality data from each service and extend Copernicus to the large number of stakeholders operating in transitional waters. The main output of the project will be a prototype system that can be "plugged into" the existing Copernicus services, developing Data and Information Access Services (DIAS), or popular open-source software used widely by the community (e.g. SNAP).

This design document lays out the intended approach to the implementation of the improved algorithms within the prototype and the processing infrastructure that will deliver the resulting satellite data products. It should be viewed in conjunction with the technical requirements document [RD1] produced in WP2 that provides details of the regions of interest, the variables that will be produced, timeliness, data format, and data access methods.

2 Introduction

This document outlines the proposed architecture for the Software as a Service (SaaS) stack for use within the CERTO project, henceforth referred to as the 'CERTO prototype'. This architecture will allow the processing "chunks" to be executed on any virtualised cloud hosting infrastructure, for instance one of the Copernicus DIAS, to deliver Earth observation (EO) water quality data products required in the case study regions. The architecture consists of several parts:

- Workflow definition engine
- Runner, to send jobs to virtual machines
- User interface, allowing users to initiate and customise processing

By combining these concepts and technologies, the CERTO architecture will allow the creation of processing systems, the dynamic execution of processing "granules" as well as the provisioning of output data to the user. In the initial design, the user interface will not be automated and will require CERTO processing partners to work with the prototype development team, as outlined in [RD1].

3 Context

Within the Copernicus programme, water quality is monitored through three distinct services: Copernicus Climate Change Service (C3S), Copernicus Marine Environment Service (CMEMS), and Copernicus Land Monitoring Service (CLMS). C3S provides a global water quality product focussing on oceanic and shelf water, CMEMS focusses on global data and regional products for European seas, whilst the CLMS provides data for inland water bodies. CERTO will build a prototype that will enable these services to be brought together, also bridging the gaps in current data provision, providing water-quality data on all water bodies, especially transitional waters in the coastal zone.

The production of EO data products results from the operation of processing chains; this is the combination of a suite of software packages, libraries, and custom scripts drawn together with a configuration that defines each step required to generate the products. Each of these existing services have their own processing chains which are often executed on dedicated high-performance computing (HPC) hardware with a closely coupled link between computing infrastructure and the software that runs on it.

The CERTO prototype will develop a new processing chain which will take the best elements from existing chains, plus the improved optical water type algorithms, the improved land-sea interface/atmospheric correction techniques and indicators; this will be implemented using a Software as a Service (SaaS) model capable of being run on almost any hosting infrastructure. This will be achieved by creating a modular processing chain that can be “packaged” into a series of loosely coupled containers, with all the software and processing tasks and their dependencies included, which can be run independently from any hardware. Each container or module will have clearly defined interfaces so that inputs and outputs of each element will be interoperable.

3.1 Example Processing Chain – CALIMNOS

The Calimnos processing chain currently operated by PML is used to generate products for CLMS and the ESA Lakes Climate Change Initiative project, as well as previous Horizon 2020 projects (EOMORES, TAPAS) and provides a solid foundation upon which to build. Calimnos is being used to process data from OLCI at 300m resolution, and developments in the TAPAS project have extended this further to allow processing of Sentinel-2 Multi-Spectral Imager (MSI) data. Calimnos already has all the main functional elements of a processing chain capable of producing CERTO products and extending this well developed and well tested chain into a modular container-based SaaS will result in a powerful, yet flexible, solution.

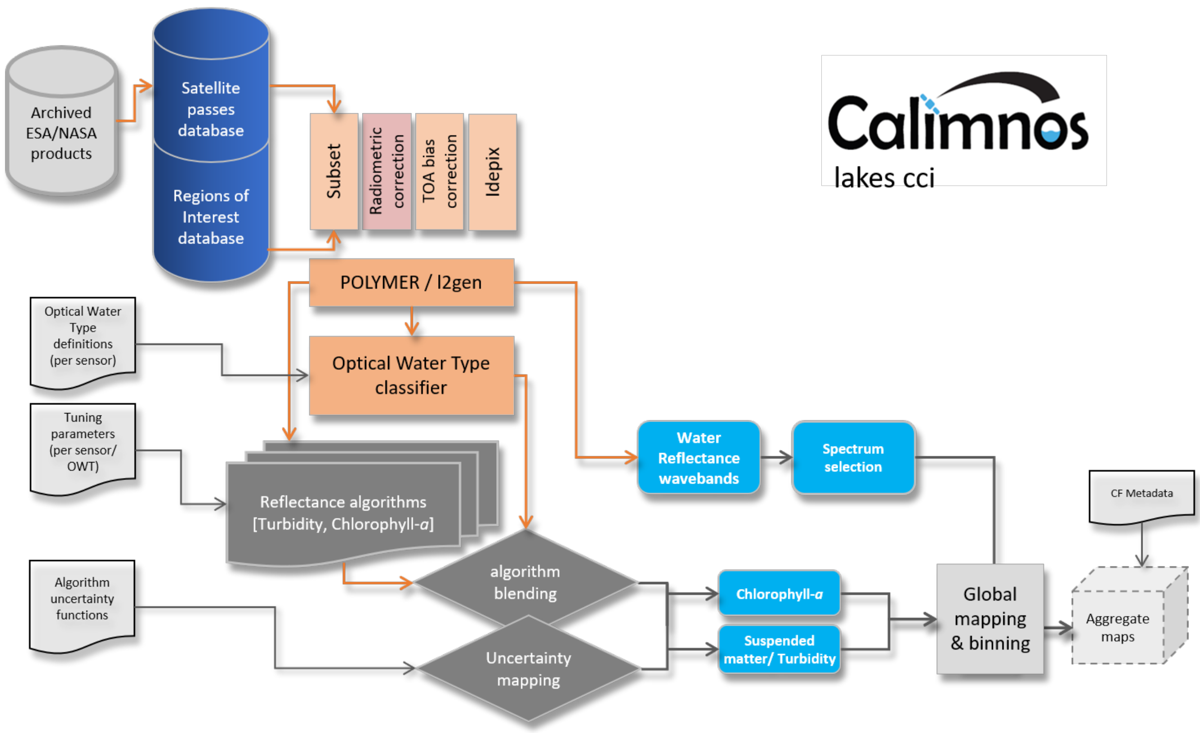

Figure 1 - Calimnos processor overview

The Calimnos chain implements several steps to produce the final products. These include:

- Data discovery
- Data subsetting by target area (individual water bodies)
- Radiometric corrections
- Pixel identification and masking (e.g. land/cloud/water/ice)
- Atmospheric correction
- Optical water type classification
- Algorithm mapping and blending
- Uncertainty mapping
- Final product aggregation and mosaicing

A schematic overview of Calimnos is given in Figure 1 showing the software components that are required to execute the processing chain. The architecture, as defined, provides a suitable environment to execute CERTO processing elements such as the Calimnos chain outlined above.

4 Deliverables & Milestones

Progress of the development of the prototype is tracked using deliverables and milestones. Following the delivery of this architecture design document at month 12, the first release of the software is due six months later at M18. An important milestone occurs at M24 with the handover of the spectral water class libraries and optional algorithm matrix for implementation into prototype. Two further releases of the software are then due; the 2nd release at M28 and a final release at M36.

5 Architecture

5.1 System Requirements

The CERTO processing system has a set of defined user requirements as set out in [RD1] and the aim of this section is to define the approach that will be taken to address these user requirements translating them to technical requirements. The main requirements for processing of data are:

- Easily configurable – the user should be able to select or define a spatial extent, the variables required, and the temporal range
- Proximity to the required data – the processing should take place where the input data are stored to avoid having to download and store large volumes of data
- Ability to run partners' processing code – the prototype will be modular in construction allowing partners to exchange modules for alternatives if required
- System should be deployable in any cloud system – this is a key consideration for the successful operation of the prototype. The entire system should very easily be run on any suitable cloud hosting provider's infrastructure
- Make results available to the user – data products generated by the prototype will be available for users in a variety of ways; details of data access methods can be found in section 9 of [RD1]

As discussed in the report on cloud computing providers [RD2] there are several DIAS operators with the data required by CERTO users. This leaves the much broader requirement of being able to run/execute partner code. This requirement is met by utilising a containerised code system, a method that provides a level of abstraction between the code/software and the infrastructure in which it runs.

Containers provide a method whereby the code is given its own instance of a system (including software and code libraries) self-contained and isolated within the container. For the CERTO prototype we will deploy software utilising Docker and Singularity; these two software packages do largely the same task, i.e. they provide a method to package software into containers and a method to run the containers, and a further explanation of their use follows in section 5.2.1.

The wider system should also be deployable and runnable on any of the available cloud infrastructure available to CERTO partners. To satisfy this, the CERTO architecture focuses on using the Linux operating system (available from all cloud providers). All the required software can be installed an executed on the Linux operating system.

To automate the process of deploying the CERTO system an automatic script will be developed using Ansible configuration files.

5.2 System Design

To satisfy the requirements as specified in user requirements document [RD1] and using the software and tools discussed above we propose the following system design.

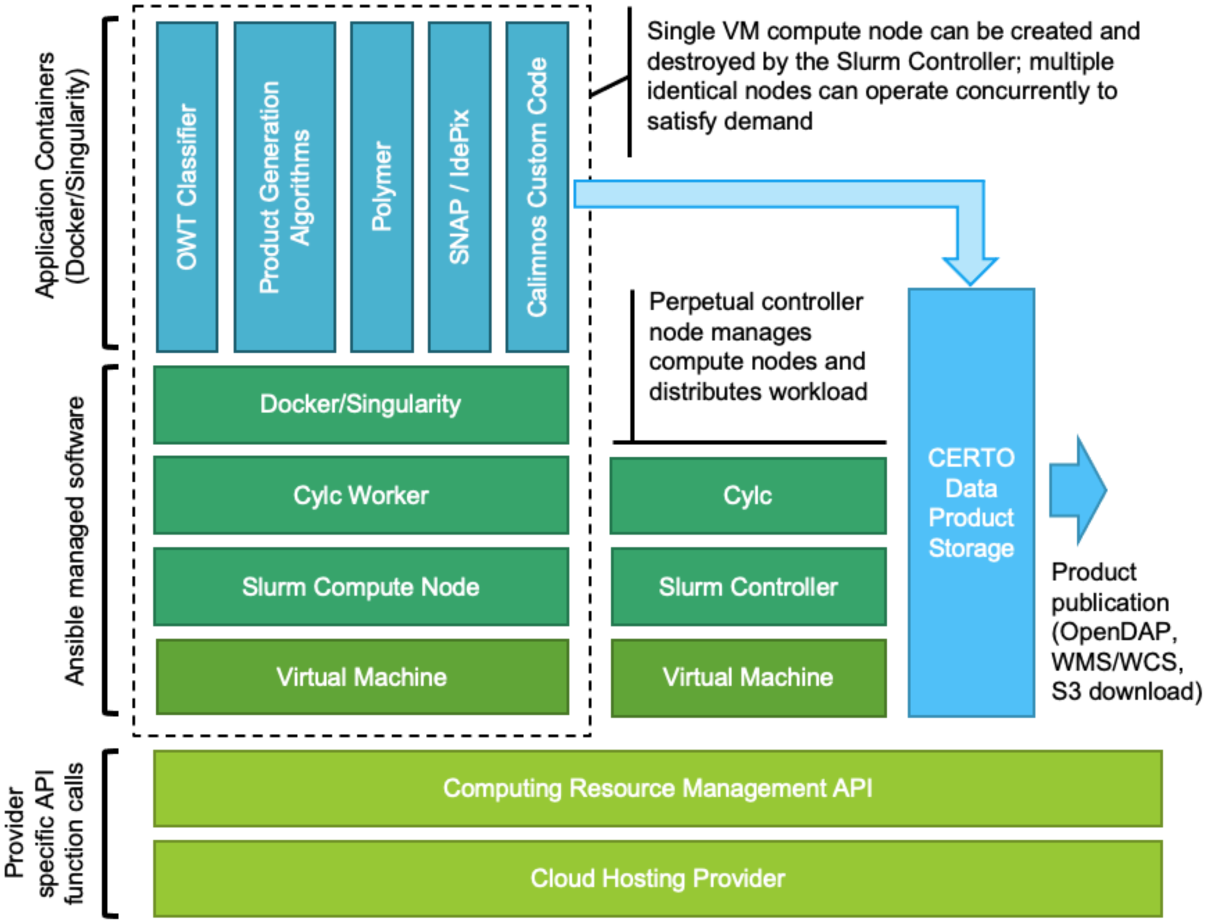

Figure 2 - System component overview

The system described in Figure 2 outlines the initial design for the CERTO processing environment. The lower two tiers of the diagram refer to the cloud hosting provider and the API they provide to programmatically deploy computing infrastructure and as previously mentioned, there are differences in the approach taken by the different providers; therefore, these interactions with this API will be provider specific. Once an API call has been made to create a virtual machine all subsequent setup is agnostic of provider and fully automated.

The setup of a compute node is managed using Ansible which installs and configure Slurm, Clyc and Docker in a fully automated manner, and with these software installed the node will be ready to start running the CERTO processing chain. The processing chain software itself will be deployed in containers which will be pulled from an external container registry, e.g. Docker Hub or a GitLab Container Registry. It is anticipated that the processing environment will change very little from the initial release, however, the processing chain will evolve significantly and regularly. Taking this approach to system design ensures that changes to the processing chain elements can be rapidly deployed without having to alter the processing environment.

The processing infrastructure requires one perpetual VM to be available so that data processing requests can be submitted at any time. This perpetual machine acts as the Cylc workflow engine – to manage which tasks run and in what order – and as the Slurm controller

– to manage where the tasks run within the compute resource pool. Cylc and Slurm will work together to manage the entire process from handling the initial request through to alerting the user when the final products are available to view/download.

Providing access to the data that is produced by the CERTO prototype will utilise existing standard interfaces. Data access methods are discussed in the user requirements document [RD1] and include direct download via an API, WMS/WCS, and OpenDAP. There are a variety of community tools and software that can interface with these protocols, such as ArcGIS, QGIS, which would allow users the ability to visualise and further analyse the outputs.

5.3 Technical Components

This section details the technologies incorporated into the architecture including a short description of each. The technical components are split into two categories; components required to deliver the processing environment, and components of the processing chain.

5.3.1 Processing Environment Components

Cloud Hosting Provider API

Each of the commercial cloud computing providers and the DIAS providers offers an API which can be used to automatically build a pre-defined computing infrastructure consisting of virtual machines, networks, firewalls, load balancing, etc. The CERTO prototype will define example configurations which can be used build a processing environment suitable for the CERTO processor. The APIs provided by each of the cloud providers differ slightly in terms of function names and arguments required, and each provider offers different specification options for virtual machines, storage, and networking; therefore, the examples provided will be specific to an individual provider – this is the only element of the prototype which is not provider agnostic.

Ansible

Ansible is an automation engine specifically designed for the provision of computing infrastructure. Once the basic virtual machine has been configured with a minimal operating system via the cloud providers' API, Ansible will be used to install and configure all other required software by taking a human readable script containing a list of software and dependencies. This eliminates the need for manually installing multiple programs on different machines and keeps work environments consistent. Ansible becomes especially useful when operating in a cloud environment as it allows easy creation of virtual machines in a fully automated way and, when they are no longer required, virtual machines can be deleted to save expense.

In the CERTO prototype, the list of software packages will intentionally be very limited as the software dependencies of the processing chain will be packaged in the containers. The primary software will be Docker, Cylc, and Slurm plus their dependencies.

Docker/Singularity

Docker is an open platform that provides a method for deploying and running applications. It offers the ability to package and run an application in a loosely isolated environment called a container. Containers are lightweight objects which bring together *only* the elements required for the application which runs in the container to operate fully: a very minimal operating system, the application itself and its dependencies are the only contents. The container interacts with the host machine's kernel and can interact with other containers through well defined channels. Docker provides an easy way to develop and deploy applications without having to be concerned with dependency version conflicts. Singularity is a very similar application to Docker and can be used as an alternative, especially in situations where greater levels of user permission control are necessary.

In the CERTO prototype each of the processing chain elements will be containerised and each step in the processing chain workflow will be a Docker (or Singularity) command.

Cylc

Cylc is a workflow engine that automatically executes tasks according to their individual schedules and dependencies. Cylc workflows are configured by a single human-readable config file supporting task inheritance and parameterization for larger, complex workflows and it will be the primary method for defining the steps and the order of the tasks in the CERTO prototype. For a more visual understanding and control over workflows, Cylc has a full graphical user interface for visualization, runtime monitoring and control as well as a flexible command-line interface.

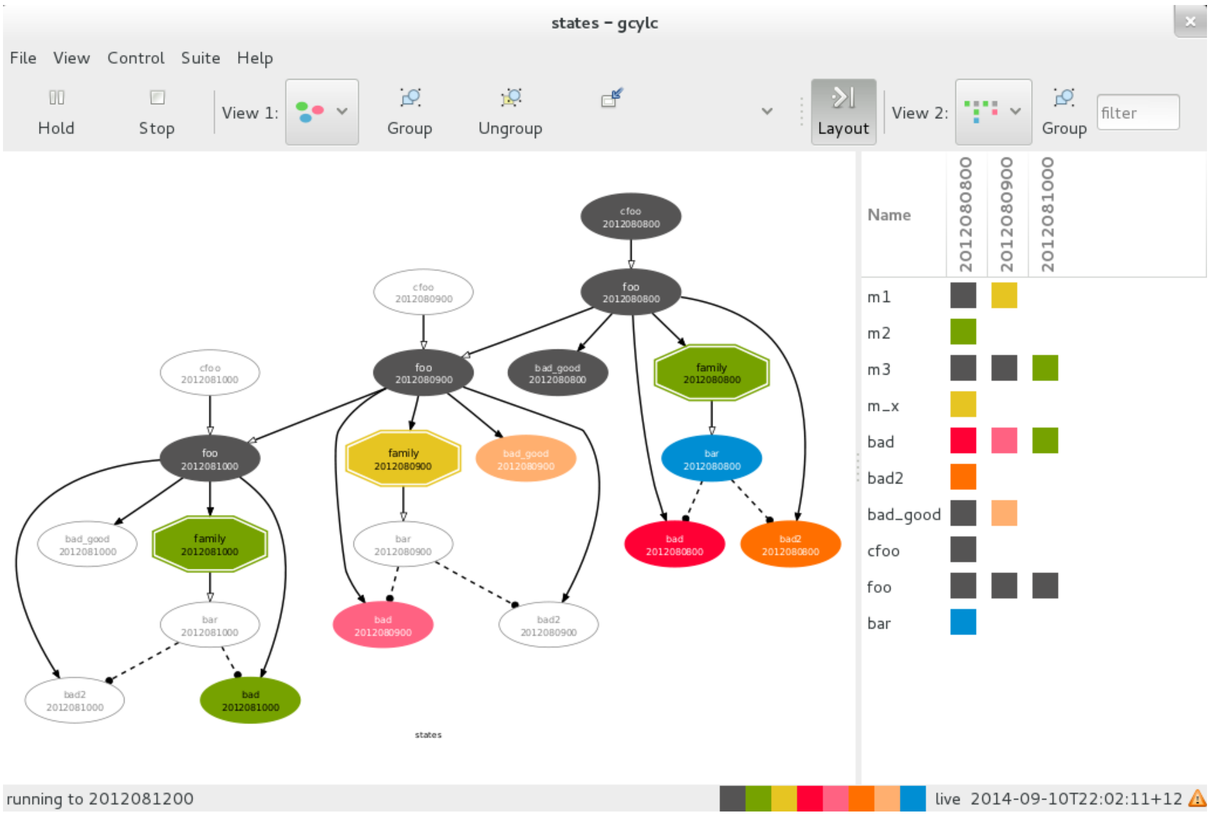

Figure 3 - cylc suite user interface example

Within a Cylc workflow each step is a task which, when completed, can automatically move onto the next task as appropriate. Cylc tasks can branch or fork, repeat, or trigger other tasks depending on certain conditions; they can recover from any failures that occur, and potentially run across multiple computer systems and resource managers.

Rose Suite

Rose Suite provides a toolkit to assist with the configuration of tasks. It works by providing a graphical user interface to demonstrate tasks and their dependencies within a suite. Other key features include version control, validating configurations and communicating with Cylc.

To create the tasks and workflow, a configuration file is created which contains the list of tasks, what they do, and their dependencies or inheritance. It is also possible to set variables such as time limits and memory resources.

In the CERTO prototype, Rose Suite will be an ancillary tool used by the development team to create and validate Cylc configurations and will not form a part of the final delivered solution.

Slurm

Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters. As a cluster workload manager, Slurm has three key functions. First, it allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work. Second, it provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes. Finally, it arbitrates contention for resources by managing a queue of pending work.

In the CERTO prototype, Slurm will control which elements of a Cylc configuration are run on which computing infrastructure. A Slurm resource pool is a collection of computing resource to which jobs can be submitted; in this case, it will be a number of virtual machines. When a request for CERTO products is placed, a new Slurm job is created which triggers a Cylc workflow to run in the defined resource pool.

5.3.2 Processing Chain Components

Satellite Passes Database/Catalogue

This database or catalogue contains details of all satellite passes, including the sensor name, processing level, acquisition date/time, geometry and filename. This provides a method to identify which satellite passes are required as inputs to produce CERTO products to satisfy a user-defined request. In the current implementation of the Calimnos processing chain at PML this exists as a PostgreSQL database which is queried using custom Python code; however, there are external resources which can replicate this service. For example, CREODIAS provide an API to allow users to find relevant files either by using a web interface or via the command line.

Workspace Creator

A workspace is simply an allocation of storage where all input data (satellite data, ancillary data, configurations) are stored, intermediate products generated by tasks in the processing chain are stored, and is a space where log files record activity. The workspace creator defines this storage ensuring that all required inputs are present as a prerequisite to processing starting.

Sentinel Application Platform (SNAP)

SNAP is a Java based application developed by Brockmann Consult which offers a rich user interface allowing users to interact with and manipulate satellite data, for example, spatial subsetting, application of algorithms, or variable merging. SNAP is expected to provide the primary route for end-users wishing to undertake their own processing using the CERTO prototype.

It also provides a Graph Processing Framework for creating user defined workflows to automate multiple sequential steps or to perform the same operations on multiple input files, and a Graph Processing Tool (gpt) for automating these workflows via the command line. SNAP includes the ability to add custom plugins which deliver features or functionality not available by default.

In the CERTO prototype, SNAP will be primarily be used via the gpt command line tool where it will be used for subsetting, flagging using the IdePix plugin, L2 product generation, merging, and binning. SNAP, the IdePix plugin and all dependencies will be deployed in a single container potentially with multiple entry point commands.

Polymer

Polymer is atmospheric correction software produced by Hygeos and written in Python. It will be deployed in its own container to ensure that improvements to the software as a result of efforts in WP5 are easily deployable within the processing chain without affecting other elements of the chain. Ensuring this relatively fine-grained modularity will offer the opportunity to test multiple versions of Polymer leaving the rest of the processing chain unaltered.

Custom Python Code

There are many elements within a processing chain that are unique, or at least highly customised, which perform important functions in producing the final outputs. These are generally created as Python libraries designed specifically for the purpose: for example, functions to perform optical water type classification, adding relevant metadata, stitching/merging files, and archiving. It is anticipated that all these custom functions will be deployed in a single container in the CERTO prototype, however, if a single element sees rapid or frequent updates it may be isolated in its own container.

6 Conclusion

The architecture outlined in this document is a first draft of the CERTO Software as a System. The elements of the system, as defined, provide a flexible, agile architecture which can be deployed onto public/research and commercial cloud environments. The architecture will be analysed for fitness during the project and any required changes or additions will be made to work with the CERTO processing partners. These versions of the architecture will be released with their own accompanying document outlining the improvements made at each iteration.

The initial test deployment of the system will be on a Copernicus DIAS chosen based on the available data and pricing models as outlined in [RD2].