Open Forecast – Security Assessment

www.open-forecast.eu
@OpenForecastEU

Co-financed by the Connecting Europe Facility of the European Union

Project	Open Forecast
Action number	2017-DE-IA-0170
Report title	Open Forecast – Security Assessment
Report number	OF_R3.2
Date of publication	31.5.2021
Revision	V1.0
Lead Partner	HLRS
Authors	Vladimir Popov, Sven Bingert
Dissemination level	Public

Executive Summary

This document describes the security assessment for the Open Forecast project. The Open Forecast Service as the result of the project is composed of a set of other services. Each service requires a detailed few on the security assessment. Each component of the Open Forecast Service is designed and deployed following latest security requirements also following the GDPR[1] requirements.

[1] https://gdpr.eu (last visited April 6th 2021)

1 Table of Contents

Executive Summary 2
1 Table of Contents 3
List of Figures 4
1 Introduction 5
2 HLRS HPC Resources 5
1.1 Access control for HLRS network 5
1.2 User registration 5
1.3 Minimum requirements for computers 6
2 HPC API 6
2.1 HPC API access control 7
3 Flowable 8
4 Sen2-Agri System 8
5 GeoServer 10
6 Summary 10
7 Acknowledgements 10
8 References 10

List of Figures

Figure 1: Design of the HPC API. It consists of three components: the user or workflow (left) submitting an HPC job, the HPC API (middle) receiving the jobs, and the HPC environment that pulls the jobs from the HPC API. 6
Figure 2: HPC API example flow diagram. The authentication takes place when a job is submitted or pulled. 7
Figure 3: Example workflow desigend in the flowable modeler app. The given example includes HPC resources by using the HPC API 8
Figure 4: Schematic view the Sen2Agri system 9

1 Introduction

The goal of a security assessment (also known as a security audit, security review, or network assessment), is to ensure that necessary security controls are integrated into the design and implementation of a project.

2 HLRS HPC Resources

The HLRS operates mainframe computers and the associated infrastructure for scientific and industrial customers. The network is protected by a whitelisting firewall, which filters based on IP addresses and ports, for both directions of traffic, ingoing and outgoing. User access is via direct connections that end directly on the border routers on the HLRS. Basically, all traffic that is not explicitly allowed is prevented.

1.1 Access control for HLRS network

Access to various services in the HLRS network and outside of it is via the access paths required for operation and for providing the service offered to customers limited.

From a technical point of view, there are two main authentication methods:

- the classic username and password principle;
- the public-key-based authentication, which is based on asymmetrical, cryptographic procedures.

For external user access, only those services are permitted where the authentication cannot be read by third parties. Authentication of users on the network can only be done in an encrypted way (for example, ssh, vpn, GridFTP, uftp). It is also possible for access using private keys that do not need to be passed as part of the authentication (this also includes certificates). The transmission of the username and password must be encrypted.

1.2 User registration

With HLRS, subgroups are implemented in the accounting system for the user accounts. An account is only allowed for one person. External users are not allowed to access HLRS resources via shared account. Group accounts are not permitted, i.e. accounts that are used jointly by several users are not permitted. However, users can very well have several groups. A person can also have several accounts for different tasks. User accounts are brought to the systems by the respective resource provider via its user administration. The UIDs / GIDs only need to be unique locally on the system. If passwords are used for authentication, users undertake to keep their password secret and not to pass it on to third parties. Also, undertake to change the start password as soon as possible. The users have to choose suitable / secure passwords.

1.3 Minimum requirements for computers

The minimum requirements for computers that users may use to access systems in the HLRS network are:

- An operating system for which security updates are offered;
- These security updates are regularly and promptly automated or manually imported;
- Suitable software must be installed and active in order to detect any malware (antivirus software, application firewall or IPS);
- In order to be able to use the computer, you have to authenticate yourself (e.g. with username and password);
- If sensitive data such as passwords are saved, they must be protected additionally.

2 HPC API

At most HPC Centers the access is limited to secure-shell (ssh) to a login node (c.f. 1.1). From this login node HPC jobs can be submitted or other nodes can be accessed. In order to include HPC resources into generic workflows we designed and implemented a REST API[2] to allow jobs submission without directly using ssh logins [1]. A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer. A REST API is a way for two computer systems to communicate over HTTP in a similar way to web browsers and servers.

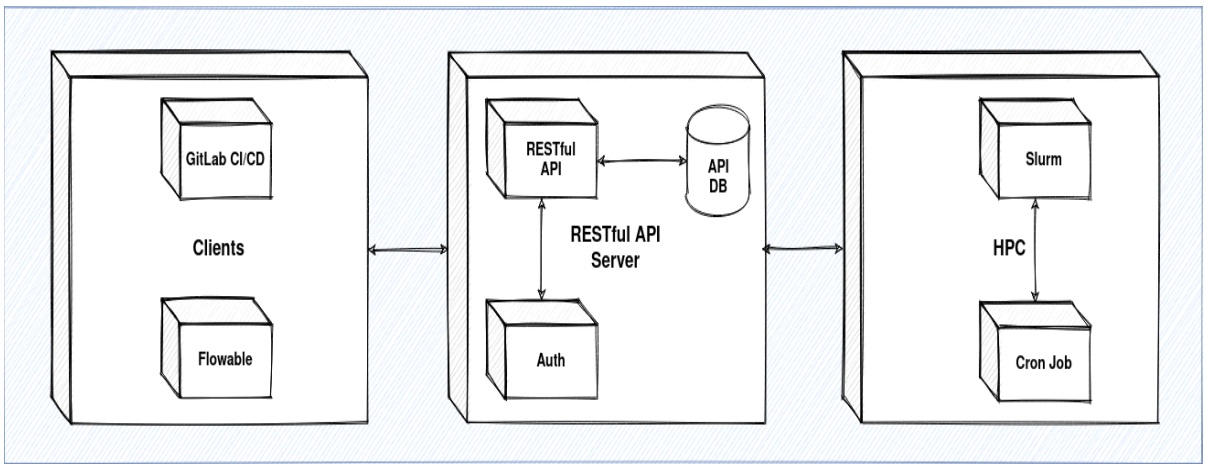

Figure 1: Design of the HPC API. It consists of three components: the user or workflow (left) submitting an HPC job, the HPC API (middle) receiving the jobs, and the HPC environment that pulls the jobs from the HPC API.

[2] hpc-api.open-forecast.eu

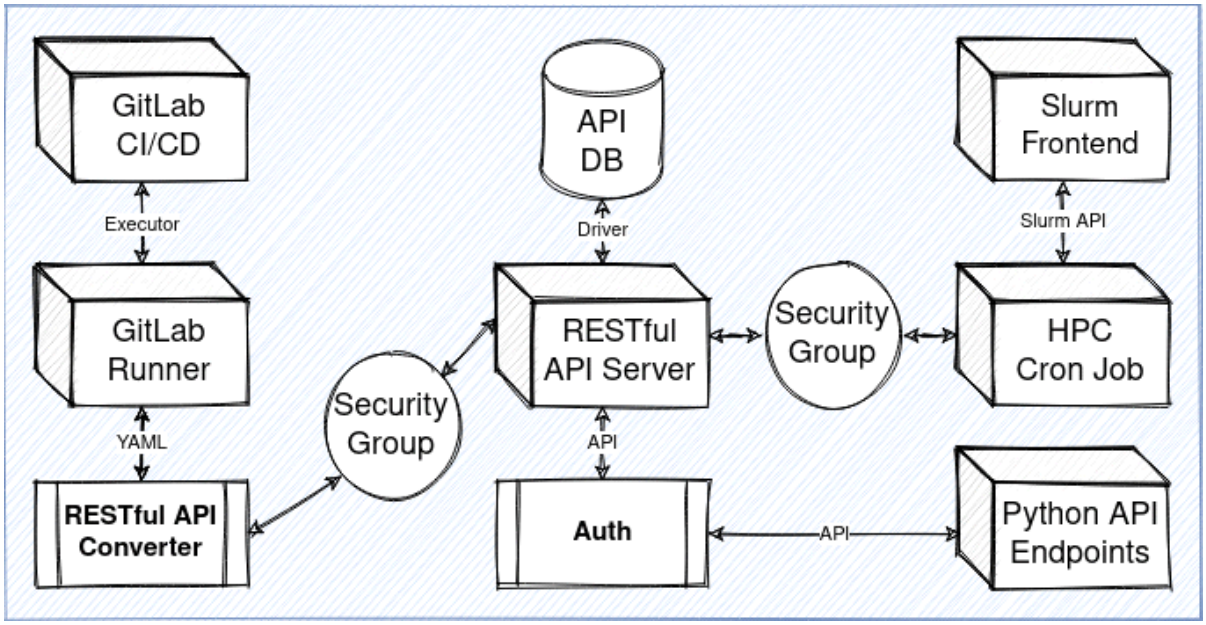

Figure 2: HPC API example flow diagram. The authentication takes place when a job is submitted or pulled.

2.1 HPC API access control

The communication to the API is done by using a security token. A security token is currently managed by the service owner and stored as encrypted file on the server. In further developments the management of the tokens will be handled by an Identity Management System (IDM). Via the IDM the user can create, delete and manage security tokens. The GWDG IDM system will include the token management as a self-service via the GWDG portal. The tokens will then not be stored on the server any more, which reduces this risk of data loss.

The security token is used to submit a job to the API. The API checks username and security token by either comparing with the reference or asking the IDM system. Remaining risks are that a token is lost or published. In these cases, it is important to have user "Role-Based Access Control" in the IDM system. For each role in the IDM, a list of its basic powers is generated. This will prevent unauthorized corruption, deletion or publication of the security token. In case of the loss of a security token, the user is obliged to immediately inform the system administrator about this in order to block and issue a new security token. In case of tokens are used in a workflow system (Flowable or GIT) the access to the token by a 3rd person needs to be prevented.

Next important point is to use a cron job. A cron job within the HPC environment is started in the user space and pulls jobs from the API. It is a way to pull automatically in a given frequency from a source. The idea is that the HPC system does not “wait” for the user to log in and to submit the

job but to look for a job at a specific resource. The cron job will use the same token as for the job submission to authenticate the user and can the assigned jobs. The token management and usage should be improved. Therefore a final thesis is offered to design and implement a token management and integrate it with the GWDG identity management system.

3 Flowable

Flowable is a light-weight business process engine written in Java. The Flowable process engine allows to deploy BPMN 2.0 process definitions (an industry XML standard for defining processes), creating process instances of those process definitions, running queries, accessing active or historical process instances and related data, plus much more. Flowable is extremely flexible when it comes to adding it to your application/services/architecture. You can embed the engine in your application or service by including the Flowable library, which is available as a JAR. Since it's a JAR, you can add it easily to any Java environment: Java SE; servlet containers, such as Tomcat or Jetty, Spring; Java EE servers, such as JBoss or WebSphere, and so on. Alternatively, you can use the Flowable REST API to communicate over HTTP. Common to all the ways of setting up Flowable is the core engine, which can be seen as a collection of services that expose APIs to manage and execute business processes.

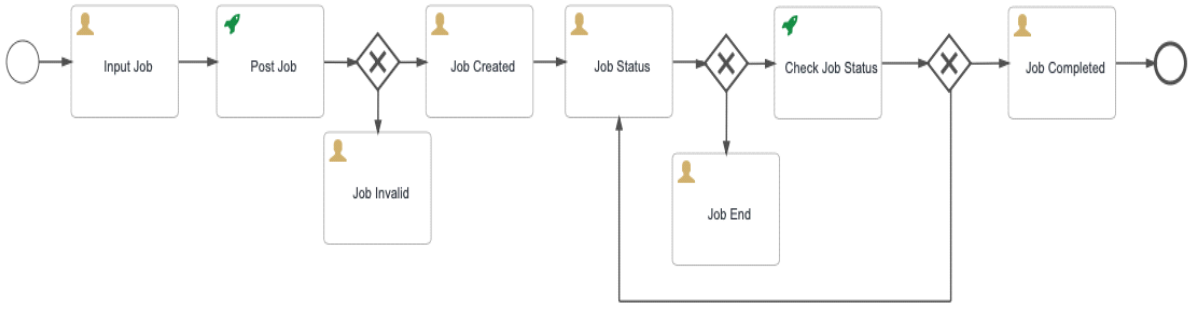

Figure 3: Example workflow desigend in the flowable modeler app. The given example includes HPC resources by using the HPC API

Currently, access to the system is organized through local user management. There is also "Role-Based Access Control", which minimizes the risk of exceeding the authorized permissions of users in the system. As designing workflows is an expert task only a limited number of people get access to the modeler. Production ready workflows can then by published as applications. These can then be accessible with or without authentication.

4 Sen2-Agri System

The Sen2-Agri system is an operational standalone processing system generating agricultural products from Sentinel-2 (A&B) and Landsat 8 time series along the growing season. The Sen2-

Agri system is free and open source, allowing any user generating near real time products tailored to his needs at its own premises or on cloud computing infrastructure. It consists in:

- A set of independent processing modules – one for each type of product and an additional one dedicated to atmospheric correction. The building blocks of the software solution for the modules rely on the Orfeo Toolbox (OTB), a high-resolution image processing toolbox, open source, portable and able to be extended to complement the suite of the already developed algorithms;
- An Orchestrator, a central component that manages the automated execution of these components from input EO data download through processing up to product delivery along the season basing upon user supplied parameters. Based on SLURM, an open-source resource manager. It is also in charge of controlling resource allocation (memory, CPU) to execute jobs and dispatching them on execution nodes (i.e. a single machine or several ones whenever possible);
- A Graphical User Interface (GUI) designed to simplify the setting of mandatory parameters (site extent and season dates) or more advanced ones, to monitor the system activity, the pre-visualise the output products, to request additional products.

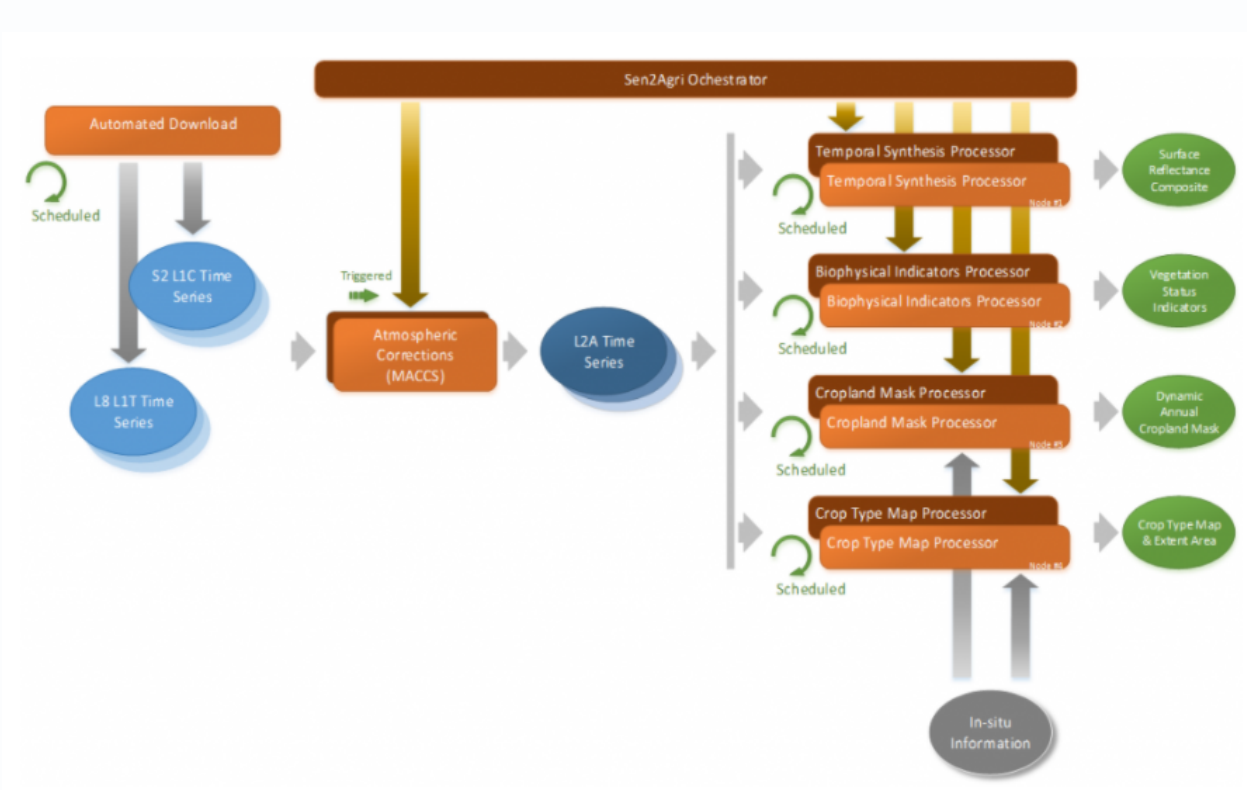

Figure 4: Schematic view the Sen2Agri system

Access to the Sen2-Agri system is organized in the same way as to the Flowable system through a local user management. The access is used for defining an area of interest, time period and

products. The access to the web UI[3] is therefore limited to data experts. The produced data will be made available via the GeoServer service. Accounts required to download the raw data from the ESA repository are stored on the server.

5 GeoServer

GeoServer is an open-source software server written in Java that allows users to share and edit geospatial data. Designed for interoperability, it publishes data from any major spatial data source using open standards. Being a community-driven project, GeoServer is developed, tested, and supported by a diverse group of individuals and organizations from around the world. GeoServer is the reference implementation of the Open Geospatial Consortium (OGC) Web Feature Service (WFS) and Web Coverage Service (WCS) standards, as well as a high performance certified compliant Web Map Service (WMS). GeoServer forms a core component of the Geospatial Web. GeoServer has its own user and access management. For the open forecast service only data experts need to configure the WMS and other services in order to serve the available data.
To access the data there is no need to use accounts, all data is presented as open data.

GeoSever software provides a detailed documentation[4] of all the security measures used in the software.

6 Summary

The services required for the open forecast service (HPC API, Sen2Agri, GeoServer) are running on virtual machines deployed on the GWDG cloud infrastructure. The security measures of the cloud infrastructure, e.g. firewall rules, are used. To protect the cloud servers only public-private key-pair access is allowed and only a very limited of admins have access. The servers are maintained following the standard procedures of the data center.
The services are additional monitored to detect false behavior. Passwords used in services which are not connected to the central GWDG IDM still have to follow the password policies of the GWDG.

7 Acknowledgements

The work carried out by the Open Forecast project is co-financed by the Connecting Europe Facility of the European Union under action number 2017-DE-IA-0170.

8 References

[1] Alamgir, Waqar, 2021, "Design and implementation of an API to ease the use of HPC systems", https://doi.org/10.25625/S3GI5N, Göttingen Research Online / Data, V2

[3] https://sen2agri.open-forecast.eu/login.php
[4] https://docs.geoserver.org/latest/en/user/security/index.html#security (last visited April 6th 2021)