

ADAM Space, managing planetary data on the cloud

A major barrier in the exploitation of Planetary data is the lack of an effective integration of the data and the processing resources supporting the entire data management cycle, namely from data discovery till to the generation and visualization of spatial data products. In the NEANIAS Space services, planetary data is being handled by the ADAM-Space platform, a suite of services for geospatial data exploration -- [<https://explorer-space.adamplatform.eu/>]. Originally conceived to Earth data exploration, in the context of the NEANIAS [<https://www.neanias.eu/>] project the ADAM platform has been extended the support to planetary data missions.

ADAM Space is a thematic customization for planetary data of the ADAM platform to enable access to surface image data from Mars and the Moon provided by NASA and ESA imagery. While ADAM and ADAM-Space are accessible through different web user interfaces [<https://explorer.adamplatform.eu/>, <https://explorer-space.adamplatform.eu/>], both rely on the core backend data access service component. In this article, we will go through some functionalities of ADAM and, more importantly, the concept of the integration of micro-services within NEANIAS and the meaning of ADAM for the scientific community to have planetary-data-as-a-service.

The goal of ADAM Space service is to setup and integrate in the European Open Science Cloud (EOSC) infrastructure a platform to handle Planetary Space Data and provide broad use services (e.g., image mosaicking) as well as CPU/GPU intensive on-demand processing tasks (e.g., landing-sites) that require reliable cloud resources at scale. Making Planetary Space Data and those services publicly available increases the *FAIRness* level and fully supports reproducible data science.

To make data FAIR -- Findable, Accessible, Interoperable and Reusable --, ADAM implements open standards data format and access interfaces as well as high-level interfaces for improved data accessibility. On such interfaces, besides the graphical web-app at <https://explorer-space.adamplatform.eu/>, the user is provided with the ADAM-API, a RESTful interface and Python API for non-interactive, machine-ready access of data.

Architecture

ADAM is composed of the following components:

- Explorer (<https://explorer-space.adamplatform.eu/>), a web application providing an interactive and user-friendly graphic user interface to explore the available data on top of Mars and Moon surface interactive globes;
- Data Access System (DAS) and Data Processing System (DPS), the components responsible to store, discover, access, and process the geospatial data via OGC standard interfaces (OpenSearch, WCS, WMS, WPS);
- ADAM-API (Application Processing Interfaces), providing low-level access to data through a Python library, implemented on top of the OpenAPI specification;

- Jupyter Notebooks Hub, providing a programming environment with the ADAM-API installed as well as access to the DAS and DPS services for advanced data exploration.

ADAM Space service components are represented in Figure 1. Software modules and micro-services are organized into five main layers according to their main functionality:

- GUI layer: this layer provides components needed for building and exposing ADAM functionalities at UI level;
- APIs layer: it includes the APIs for the integration between GUI and Services layers;
- Services layer: it handles services for the APIs and the GUI execution. It is composed by the Data Access System (DAS) for discovering and access data, and the Data Processing System (DPS) to provide processing capabilities on Kubernetes;
- Scalability layer: this is the processing environment which manages Kubernetes PODs, Docker images and instances;
- Storage layer: it provides storage resources;
- Authentication, Authorization and Account provisioning (AAA) layer: for the authentication and authorization of the system users.

The high modularity of the service components and the interoperable layer offered by the API and standard interfaces allows at supporting both centralized and distributed deployments across different IaaS (Infrastructure-as-a-Service) providers. Services composing ADAM Space are split across different infrastructures, between MEEO cloud, GARR cloud [<https://cloud.garr.it>], and Amazon AWS [<https://aws.amazon.com>], according to their role and need for scaling up or privacy.

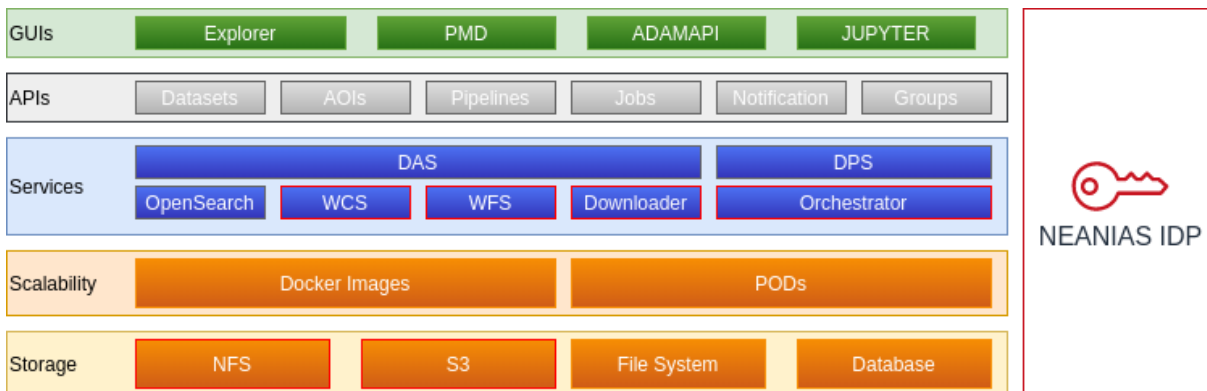


Figure 1: ADAM infrastructure.

To support the Single Sign-On approach, internal or external Identity Providers can be integrated and configured. Figure 2 shows the available options for the ADAM Space instance.

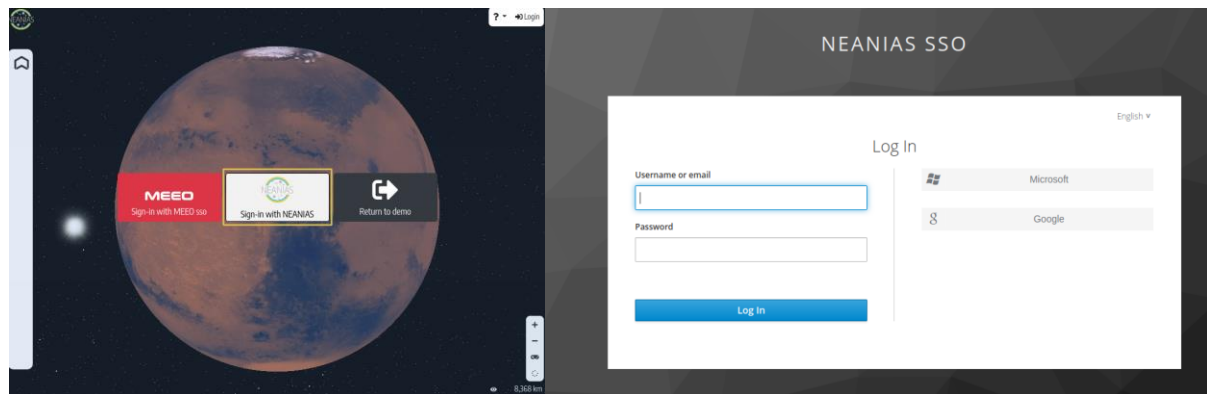


Figure 2: ADAM Login (left) and NEANIAS AAI Sign-in federation (right)

NEANIAS AAI provides the possibility to access service by the Microsoft and Google Identity Providers as shown in Figure 2

A logged user has the authorization for working on a set of datasets (for more details see the section Data). Search and discovery functions are provided by the Graphical User Interface as shown in Figure 3 and Figure 4.

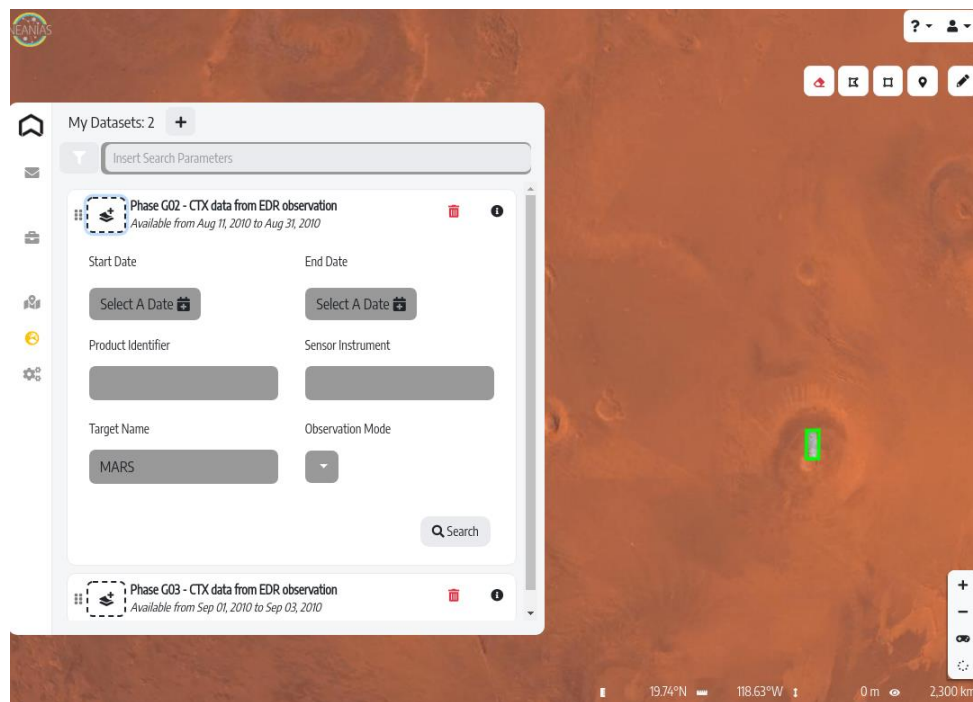


Figure 3: Data search

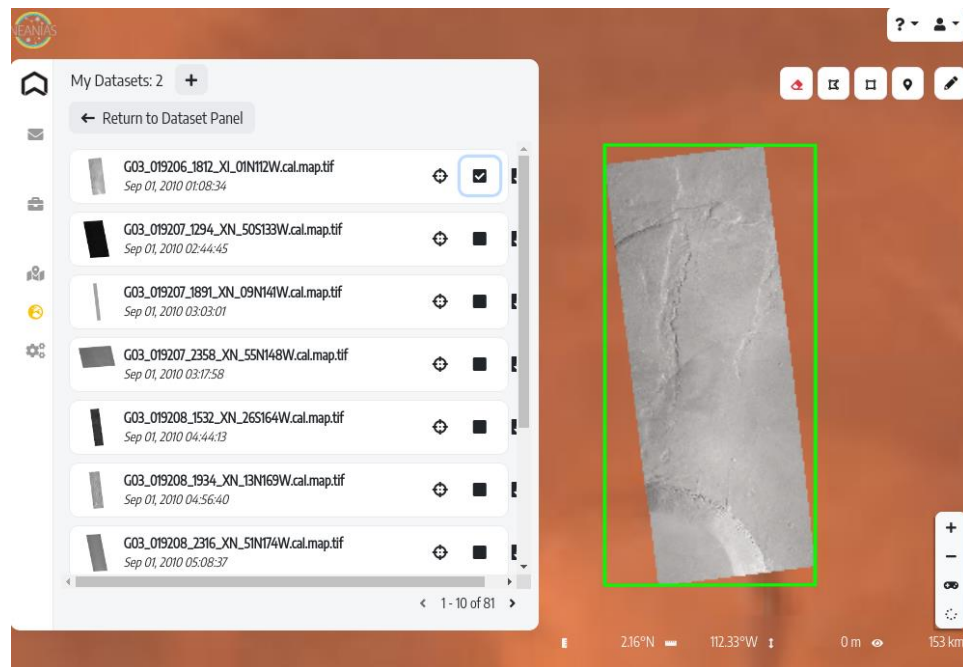


Figure 4: Data discovery

Planetary Data Science

Planetary data is originally provided by Planetary Data System (PDS) nodes from NASA/USGS. Image data is copied to our systems and prepared to be used by the thematic services we are providing for the NEANIAS community: mosaicking, landing-sites evaluation, map-making.

The Data Access System (DAS) module enables discovery and access functionalities, and it is fully configurable to manage both local and remote data catalogues. Utilities are available for ingesting geospatial data and for the analysis of intermediate and final results of the (pre-)processing pipelines. All data archived in storage resources are accessible via standard interfaces:

- OGC: Catalogue Service for the Web (CSW),
- OpenSearch: with OGC OpenSearch Geo and Time extension,
- OGC: Web Coverage Service (WCS).

Data *reduction* is a series of steps to normalize data from different datasets. Besides homogenizing file formats and other technical aspects like image intensity range and datatype, for instance, instrumental noise and physical units conversion is also part of the data preparation for internal (i.e., ADAM data store) archival for future use. The execution of those steps is managed by ADAM *orchestrator*, described in section **Processing**.

ADAM's *Downloader* component is responsible for copying of *Datasets* from long-term data storage dedicated in the MEEC Cloud, it does so by querying the Data Access Service interface for data products in a given region of the planet. In case the retrieved data products are not yet in the local data store, the dataset manager can launch a pipeline through ADAM's Orchestrator which subsequently downloads, process and ingest the data for internal, long-term storage.

Eventually, NEANIAS Space Planetary data products are available (1) interactive exploration back-referencing the original/source data products where the user can trace and use, (2) download of our reduced image, and (3) further use by the thematic services provided by ADAM/NEANIAS Space planetary.

Processing

When we are talking about a pipeline, we mean the minimum unit of processing composed by four main steps (list, download, process and load) which communicate by GeoJSON standard. It can be executed in a scalable environment (AWS or Kubernetes) using the ADAM DPS component. For the DPS are not relevant the language used for the pipeline implementation and the specific area (planetary, astrophysics and so on) related to its processing but there are some criteria that pipeline must follow:

- communication between steps by geojson standard;
- the possibility to create a docker image of the pipeline code;
- working on raster data.

ADAM-API

The users might prefer a CLI/standalone solution therefore the ADAM Space service provides the `adamapi` python-library that can be used to discover, access and computing data from a command line interface or in Jupyter notebook.

This `adamapi` package is available through the Python Package Index¹. Currently the API package provides the following functionalities:

- Auth, the module to configure the environmental variables and a method to manage the authorization;
- Datasets, the module to discover the datasets available in the ADAM Space service, including all properties (e.g. description, start/end date, spatial coverage);
- Search, the module to discover the products available for a specific dataset, including filter options by supported parameters (e.g. geometry, attributes, tile, ...);
- GetData, the module to access a product or a timeseries of products, up to pixel granularity.

With `adamapi` installed and their private API key at hand, users can query, access and process data provided by ADAM wherever their custom software solution is deployed – Amazon cloud or their personal computer – it is transparent for the data.

Conclusion

In this article we presented the ADAM Space service and its key role on scaling up scientific planetary data to EOSC through the NEANIAS project, aiming to provide a seamless set of services for the geological community.

¹ <https://pypi.org/project/adamapi>



In the next releases, processing services will be enabled in ADAM Space as well as Planetary Science Data offer will be extended to incorporate different Martian imagery from different instruments for a smoother integration to users data analysis workflow.

Sure enough, all users willing to explore and interact with Martian data through ADAM Space service with their Google or Microsoft accounts, or integrate their Python routines with ADAM-API are invited to do so and contribute to the Martian cloud exploration.

Carlos H. Brandt
Jacobs University Bremen

Laura Vettorello, Simone Mantovani
Meteorological Environmental Earth Observation (MEE0)