Cloud Based Discovery and Processing of Geospatial Data

Arne Vogt
Bochum University of Applied Sciences, Department of Geodesy
Lennershofstraße 140, 44801 Bochum, Germany
arne.vogt@hs-bochum.de

Andreas Wytzisk-Arens
Bochum University of Applied Sciences, Department of Geodesy
Lennershofstraße 140, 44801 Bochum, Germany
andreas.wytzisk@hs-bochum.de

Sebastian Drost
Bochum University of Applied Sciences, Department of Geodesy
Lennershofstraße 140, 44801 Bochum, Germany
sebastian.drost@hs-bochum.de

Simon Jirka
52°North Initiative for Geospatial Open Source Software GmbH
Martin-Luther-King-Weg 24, 48155 Münster, Germany
jirka@52north.org

Abstract

Cloud Computing becomes increasingly important in information technology. Many applications and data sets have already been transferred to a cloud environment. This technical development is therefore also relevant for spatial data applications. Existing approaches that outline search and processing of geospatial data optimized for cloud computing will be addressed and compared with the characteristics of current spatial data infrastructures. Further, a software architecture is proposed which covers approaches for geodata discovery and geodata processing workflows in cloud computing environments.
Keywords: spatial data infrastructure, cloud computing, spatio-temporal data, data discovery, geodata processing, software architecture.

1 Introduction

Due to advanced technical possibilities the collection of data has increased significantly in recent years. Also in the domain of geoinformatics, large amounts of spatio-temporal data are collected. For example, high-resolution imagery of the earth's surface is generated by earth observation satellites like the *Sentinel* satellites of the *Copernicus* program. Furthermore, sensor networks constantly publish a great variety of measurement values. In addition, due to Open-Data efforts, large data sets from the public sector become increasingly available. A possible solution for handling large amounts of data is the utilization of cloud computing. While cloud infrastructures are already used commonly to store spatio-temporal, data processing workflows mostly have not changed. Therefore, the capabilities of cloud computing are not exploited fully. It is still common for users to download (spatio-temporal) data and apply algorithms in a local environment, e.g. a desktop computer. To endorse the utilization of cloud computing, new approaches to organize discoverability and exchange of spatio-temporal data are emerging.

2 Related Work

Gorelick et al. (2017) introduce the platform *Google Earth Engine* (GEE) which allows on-demand processing of spatial data in the cloud. GEE gives an example of how spatial data processing workflows can be realized without the use of local hardware. Krämer (2018) demonstrates a software architecture that uses microservices for processing large sets of spatial data. The architecture is suitable for deployment in a cloud environment. As with GEE, users can select data and algorithms from an internal catalogue. Holmes (2017) gives an outline of how discovery and processing of geodata could be structured in the future to better match the characteristics of cloud computing. The specification *SpatioTemporal Asset Catalog* (STAC) (SpatioTemporal Asset Catalog, 2019) is conceived to promote the implementation of these approaches. The proposed software architecture adopts the STAC specification (see section 4). *Spatial Data on the Web Best Practices* (Tandy, van den Brink and Barnaghi, 2017) is a guideline on how to make geodata easily accessible on the web. Many of the best practices provided by this guideline are considered by STAC and are therefore also relevant in the context of the proposed software architecture.

3 Current SDIs and emerging Concepts

Many existing spatial data infrastructures (SDI) are designed as a decentralized service-oriented architecture. These SDIs implement the *Publish-Find-Bind* pattern. Existing standards harmonize metadata descriptions as well as search and exchange of geodata. Standards specified by the *Open Geospatial Consortium* (OGC) are constantly implemented by data providers of the public sector (e.g. in the context of INSPIRE), but many other data providers define their own APIs to search and access geodata. So in practice a data user still needs to know relevant data providers and their specific APIs in order to access data sets of interest. Holmes (2017) envisions a scenario where each set of spatial data is available in a cloud environment. The data provider supplies brief metadata files for each of its data sets (see section 3.1). Potential users of available data can register and index published metadata in customized catalogues, e.g. by web crawling. In contrast to the Publish-Find-Bind pattern users create their own catalogues instead of using several different (provider-specific) APIs.

3.1 SpatioTemporal Asset Catalog

The SpatioTemporal Assets Catalog specification (2019), which is currently under development, aims to standardize the publication and search of spatio-temporal data. STAC is

a lean metadata specification that uses the JSON format. The STAC specification comprises essentially two types of documents. Each document is exposed on the web and can be referenced by a unique URL.

A *STAC-Item* (Figure 1) is a simple metadata description of a geospatial asset that is compliant with the GeoJSON specification. An example for an asset is a scene generated by a Sentinel satellite. STAC-Items contain hyperlinks to the actual geodata. Those references provide a common, simple method to access geodata. A spatial and a temporal attribute are mandatory for each asset. Technical properties are specified in additional extensions. Further, a STAC-Item links *STAC-Catalogs* or other STAC-items.

A STAC-Catalog (Figure 2) is used to structure metadata documents by linking STAC-Items or further STAC-Catalogs. In Addition, a STAC-Catalog includes generally valid information about the data and the data provider.

Figure 1: STAC-Item for a Landsat scene, linking a geotiff image for each spectral band (B1, B2, …)

```
{
  "id": "LC81530252014153LGN00",
  "type": "Feature",
  "bbox": [...],

  "geometry": {
    "type": "Polygon",
    "coordinates": [...]
  },

  "properties": {
    "datetime": "2014-06-02T09:22:02Z",
    "description": "Landat 8 imagery ...",
    "eo:gsd": 30.0,
    "eo:cloud_cover" : 10,
    "landsat:wrs_path": 153,
    "landsat:wrs_row": 25,
    ...
  },

  "links": [
    {
      "rel": "self",
      "href": "http://[...]/L8/LC81530252014153LGN00.json"
    },
    {
      "rel":"root",
      "href": "http://[...]/L8/catalog.json"
    }
  ],

  "assets": {
    "thumbnail": {
      "href": "http://[...]/L8/LC81530252014153LGN00_T.jpeg",
      "type": "jpeg"
    }
    "B1": {
      "href": "http://[...]/L8//LC81530252014153LGN00_B1.TIF",
      "type": "geotiff",
    },
    "B2": {
      "href": "http://[...]/L8/LC81530252014153LGN00_B2.TIF",
      "type": "geotiff",
    },
    ...

  }
}
```

Figure 2: STAC-Catalog containing links to STAC-Items and provider information

```
{
  "id": "landsat_sample",
  "title": "Landsat8 catalog",
  "description": "Catalog for L8 imagery",

  "links": [
    {
      "href": "http://[...]/L8/catalog.json",
      "rel": "self"
    },
    {
      "href": "http://[...]/L8/catalog.json",
      "rel": "root"
    },
    {
      "href": "http://[...]/L8/LC81530252014153LGN00.json",
      "rel": "item"
    },
    ...
  ],

  ...
  "formats": ["geotiff"],
  "provider": {
    "scheme": "s3",
    "region": "us-east-1"
  }
}
```

4 Architecture for Handling Geodata in Cloud Enviroments

This section describes a proposed software architecture which is devised to fully accomplish search, exchange and processing of spatio-temporal data in cloud environments. This architecture comprises two basic components. The first component realizes the discovery of geodata. The second component provides capabilities to automate geodata processing in cloud environments. The proposed architecture is designed to use the STAC specification.

4.1 Discovery

Web Crawler: A web crawler downloads documents from the web and extracts all relevant information and references (hyperlinks) to other documents. In the next step the web crawler downloads the documents that are referenced and handles them like the initial document (seed). Within the context of the proposed architecture, the documents that are downloaded meet the STAC specification. Since each STAC document contains references to other STAC-Items or STAC-Catalogs, entire metadata collections can be captured by applying web crawling. In addition to the references, the web crawler also extracts the contained metadata from the STAC-Items.

Search Capabilities: The proposed architecture provides that the metadata extracted by the web crawler is added to a search index. Hence, the web crawler forwards extracted metadata to a search index. Figure 3 depicts the workflow that comprises the crawling and indexing of metadata.

A simple REST API (Wrapper API) is introduced to the architecture in order to encapsulate the access to the underlying search index. By including this interface, the architecture does not impose a dependency to any specific search engine

technology. The Wrapper API allows the definition of basic spatial, temporal and attribute-based filters.

4.2 Processing

Algorithms and Processing Services: The processing component comprises the various algorithms that can be applied to spatio-temporal data. Furthermore, this component handles the access to the input data which is demanded by a processing workflow. For the envisaged fully automated processing of spatial data it is necessary that accessing data works without manual interactions. For this reason, the concept of *processing services* is introduced to the architecture. A processing service encapsulates one or more algorithms and is also responsible for matching input data with the parameters of a specific algorithm. To find the appropriate input data, the *processing services* interacts with the described discovery component. Geodata is accessed by dereferencing the hyperlinks contained in STAC-Items.

An important goal of the proposed architecture is to allow processing of geodata close to the data directly in the same cloud infrastructure in which it is stored. Transferring or downloading data is often a limiting factor in the performance of data processing systems. To avoid this limitation, data must be used where it is hosted instead of moving it to an external processing unit. By transferring algorithms, the special network infrastructure provided by the cloud environment can be exploited. Hence, it must be possible to port the processing services to another environment effortlessly. A multi-cloud scenario must be supported because the different data providers do not always store their data in the same cloud infrastructure (Figure 4). It is proposed to utilize containerization solutions, e.g. *Docker*, in order to achieve the necessary portability. Containerization provides a method to encapsulate processing services. As most cloud providers already support popular containerization software, this kind of encapsulation facilitates transferring software components between different environments.

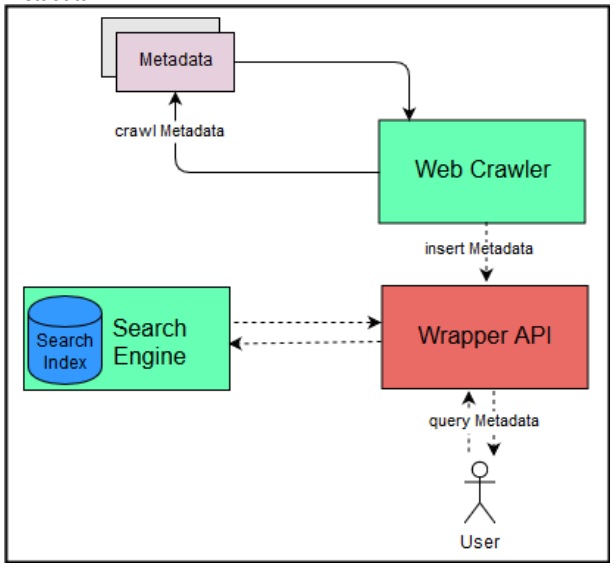
Metadata Generation: The proposed architecture also includes the automatic generation of metadata. As for input data and metadata, it is envisaged that the products of data processing will be stored in a cloud-based storage. As output data is added to the storage, new metadata files are created automatically. Many cloud storage solutions allow the registration of event handlers that are executed when data is added to storage. Event handlers can for example be implemented as Function as a Service (FaaS). FaaS is the event-based execution of short code scripts. In this scenario, a script generates appropriate metadata documents (STAC-items) when processing outputs become available. As provided in the STAC specification, the metadata for processed data contains hyperlinks to the metadata of the original data

Processing Chains: As for metadata generation, it should be possible for processing services or external applications to immediately handle new output data when it becomes available. This can also be achieved by facilitating FaaS. A script can be registered which then calls another application. Alternatively, a more elaborated messaging system can be installed. Such a system has the advantage that messages can be distributed according to more precise patterns. Many cloud providers already offer services for implementing a message-oriented architecture. An event system enables the automatic execution of processing chains comprising multiple processing steps. Ultimately, this should lead to fully automatic geodata processing workflows in the cloud.

5 Future Work

The proposed software architecture presents concepts how discovery and processing of spatio-temporal data in a multi-cloud context can be achieved. The implementation of this architecture is in a prototype phase. Individual parts of the architecture, such as the discovery component, are already realized for test purposes. The current prototype should be continuously extended in the future in order to further validate and refine the underlying concepts. Especially the infrastructure of *Amazon Web Services* has been used so far for prototypical implementations. It is planned to examine the infrastructure of other cloud providers in depth in order to prove the portability with further application scenarios. This should mainly address the replaceability of cloud-specific services, e.g. FaaS capabilities. A relevant environment would be, for example, *Mundi Web Services* as it provides access to the Sentinel earth observation data.

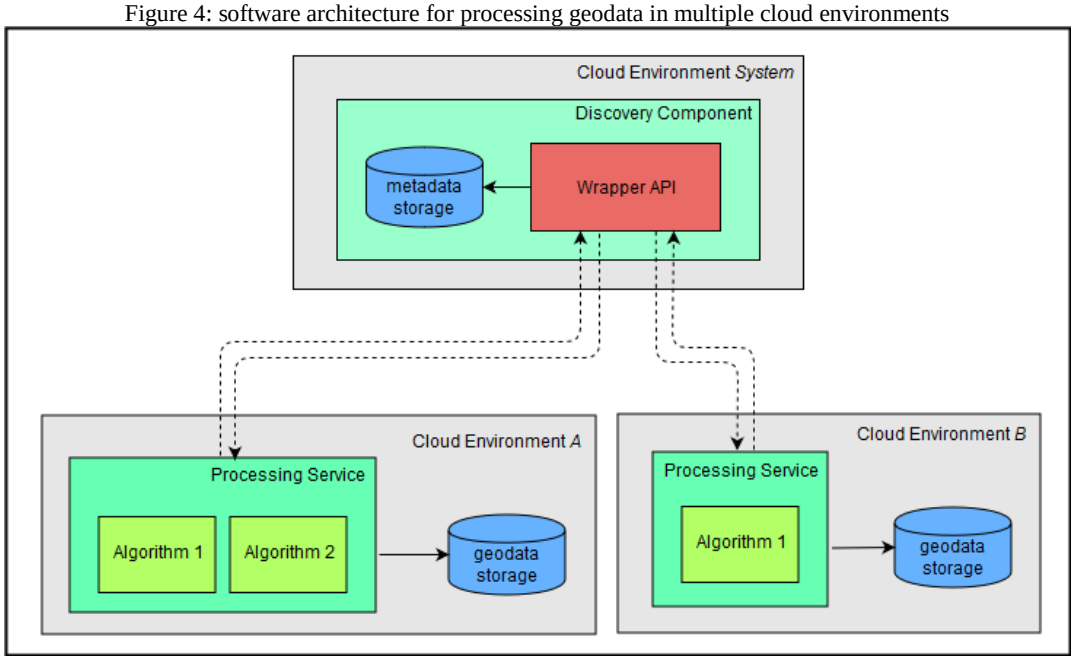
Figure 3: software component for collecting and querying metadata



6 Acknowledgment

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

Figure 4: software architecture for processing geodata in multiple cloud environments



References

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D. and Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. Elsevier.

Holmes, C. (2017). Cloud Native Geospatial Part 1. [Online] Available from: https://medium.com/planet-stories/cloud-native-geospatial-part-1-basic-assumptions-and-workflows-aa67b6156b53 [Accessed 20 February 2019].

Krämer, M. (2018). A Microservice Architecture for the Processing of Large Geospatial Data in the Cloud (Doctoral dissertation). Technische Universität Darmstadt.

SpatioTemporal Asset Catalog (2019). SpatioTemporal Asset Catalog. [Online] Available from: https://radiantearth.github.io/stac-site/ [Accessed 20 February 2019].

Tandy, J., van den Brink, L. and Barnaghi, P. (2017). Spatial Data on the Web Best Practices. [Online] Available from: https://www.w3.org/TR/sdw-bp/ [Accessed 21 February 2019].