www.datamarket.at

Initial Release of Foundational Service Technology Prototypes

Deliverable number	*D6.2*
Dissemination level	*Public*
Delivery date	*31st of March 2018*
Status	*Draft*
Author(s)	*Heimo Gursch (KNOW)* *Herwig Zeiner (JRS)* *Hermann Fürntratt (JRS)* *Artem Revenko (SWC)* *Thomas Lampoltshammer (DUK)* *Lőrinc Thurnay (DUK)* *Bernhard Niedermayer (Catalyst)*

FFG

The Data Market Austria Project has received funding from the programme "ICT of the Future" of the Austrian Research Promotion Agency (FFG) and the Austrian Ministry for Transport, Innovation and Technology (Project 855404)

bmvit

Executive Summary

The data driven economy needs not only data but also intelligent services putting the data to use. The efforts in work package 6 (WP6) are targeted at populating DMA with intelligent and easy to use services. WP6 deals with two different aspect. Firstly, WP 6 develops tools to simplify the integration of services into DMA. Hence, the burden for publishers of services becomes lower when entering their services into DMA. Secondly, several services are developed in WP6 which will be the first available products once DMA is launched. These services should be role models for other services joining DMA in the future, motivating other people to use them in their projects, and also to motivate others to develop new services and publish them via DMA as well. In the first category of tools simplifying the integration of services into DMA fall two undertakings of WP7:

- The **Service Ingestion** defines the lifecycle of services in DMA. The lifecycle starts with of the publishing of a service into DMA. It is then followed by the maintenance, updates, and changes of the service when they are necessary. The lifecycle ends with the retirement of a service from DMA. The service ingestion offers tools to assist this lifecycle as well as tools to describe the API of a service, its legal requirements and manages its metadata so that a service can be found by the potential customers. Furthermore, the service ingestion offers a code generation for developers desiring to integrate a DMA service in their own project.
- The **Semantic Enrichment and Entity Linking** component deals with the metadata describing the services ingested into DMA. It processes the service metadata and extracts concepts out of the metadata characterising the services and their features. These characterisations are an important input for the search and recommendation in DMA. To unify the description of services, the concepts are validated by a thesaurus.

The second category of services provide the first services consumable via DMA. They can be assigned to the earth observation, mobility, data processing, and information science.

- The **Data Clustering and Slicing Services** offers the ability to extract test datasets out of much larger dataset for the purpose of testing and evaluation of a dataset without the need to investigate the complete dataset at once. The service works in two steps. Firstly, the data is clustered in different smaller partitions (e.g. by means of spatial clustering). Secondly, the different portions are provided individuality by the service.
- The **Earth Observation Services** are composed by three services for the processing of satellite data from the ESA. The forest change monitoring service calculates the changes of forests due to natural growths unfinanced by environmental disturbances like heavy gales. The influence of wind also in the focus of the second service, highlight the storm damage resilience of forests. The third service deals with rockfall propagation modelling.
- The **Heat Map Service** is targeted at the visualisation of geographic intensity information. The service is aimed at visualising taxi demands on the city map.
- The **Taxi Ride Sharing** will offer the ability to combine separate taxi rides into one. Taxi users can share their ride to cut costs on their trips. The service is targeted at combining separate taxi rides with the least impact in comfort for the taxi users.
- The **Recommendation and Search** service offers a scalable and distributed recommender system. The recommender system can handle different types of information (e.g. item descriptions, location information, transaction data) and be configured to the needs of the different recommendation tasks. It also offers an extensive search functionality.

This deliverable is composed after the first half of the project time is already passed. For all of the described services the design phase and the first development iteration are concluded. The services are currently extensive tested, and extensions and further functionality are implemented. The second project phase will see the execution of the said implementations with further functionality. It will also lay an emphasis on the integration of the developed services into the Docker based DMA infrastructure.

Table of Contents

List of Abbreviations

ALS	Airborne Laserscanning
ASA	All Source Area
API	Application Programming Interface
CB	Content-Based filtering
CF	Collaborative Filtering
CPU	Central Processing Unit
CSV	Comma-Separated Values
DML	Data Modification Layer
DSM	Digital Surface Model
DTM	Digital Terrain Model
ESA	European Space Agency
F1-score	measure of a test's accuracy, also called F-score or F-measure
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
ID	Identifier
IMPACT	Image Processing and Classification Toolkit
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MP	Most Popular
MSI	Multi-Spectral Instrument
nDCG	Normalised Discounted Cumulative Gain
nDSM	normalised Digital Surface Model
POC	Proof Of Concept
RAM	Random-Access Memory
RC	Recommender Customiser
RE	Recommendation Engine
REV	Recommender Evaluator
RSG	Remote Sensing Software Package Graz
SAGA	System for Automated Geoscientific Analyses
SD	Standard Deviation
SSA	Single Source Area
XML	Extensible Markup Language
YAML	Yet Another Markup Language

1 Introduction

The advent of Big Data highlighted the potentials hidden in datasets. To leverage these potentials, the matching services to the data are needed. DMA aims at bringing services and datasets together creating a data driven market. Work package 6 (WP6) concentrates its efforts on the services for DMA. The efforts in WP6 can be summarised into two categories. Firstly, WP6 works on the so-called central-services for DMA, which are tools for DMA participants allowing them to submit their services to DMA marketplace and infrastructure. Secondly, WP6 also develops the first services which will be offered in DMA to potential customers.

When looking at the central services, i.e. the tools for the participations, of DMA, they should create the infrastructure and framework for DMA to become a lively marketplace with reach interactions. The central service which is required to the add, update, maintain, and retire a service offer from DMA is therefore one of the essential parts of DMA and is called service ingestion. The current status of the service ingestion development is described in Chapter 2.

Developers publishing services in DMA is just one aspect in DMA, but potential customers also need to find the services they are looking for. The semantic enrichment and entity linking processes the metadata describing the services. This processing is an enrichment of the metadata making services easier to find by potential customers. The work of this topic is reported in Chapter 3.

The remaining Chapters 4, 5, 6, 7, and 8 describe the current work on the services. These services will be the first offers available in DMA and should provide the participants in DMA with an incentive to use the services in their projects or to also create and offer services on their own. Many of these services will also be used in the pilots developed in WP 8 and WP9.

This deliverable reports the status of the affairs in WP6 after the first half of the project has finished. In this document the current status for each service developed in WP6 is described in detail. Also, there is an outlook for the upcoming work on the said service which will be undertaken in the second half of the project.

2 Service Ingestion

2.1 2.1 DMA Service Description

Data services and its application programming interfaces (APIs) become more and more important in a closer connected world. In order to provide consistent and robust interfaces, the API life cycle has to be kept in mind. In the light of the Data Market Austria, its API life cycle comprises the following stages:

- Create a data service with its API
- Publish it on the Data Market Austria (service ingestion)
- Contract with customers and manage access
- Update services
- Keep different versions
- Deprecate services, and finally
- Retire services

Data Market Austria promotes the use of robust APIs based on established standards, such as the OpenAPI standard[1]. With the extension of the OpenAPI standard, the DMA RESTful service description allows to contain all three important description parts in a compact way in a single file:

1. The specification of the application programming interface between client and server – from a software engineering perspective: with version information, which calls are available, minimum requirements that have to be implemented on the clients' side, security models, URLs for testing, staging and deployment.
2. Information about legal aspects – license models, service level agreements, what is expected from the customer, what is guaranteed by the vendor (quality of service).
3. Metadata for brokerage – which enables quick discoverage through search, metadata enrichment from additional sources for proper ranking according to search queries, and recommendation of services.

The DMA service description is both – human understandable, as well as machine readable, which allows to auto-generate source code from it. Although the majority of programming languages is supported by the proposed code generator, user defined source code templates can be created, in case the code generator does not provide a proper template for the required programming language. If customised templates require too much effort, manual implementation of the API is still an option.

Furthermore, the whole service API documentation can be auto-generated as well.

So, with this compact service description we can achieve an easy and comprehensive acquisition of all metadata. At the same time, offering a repository and tools to handle such descriptions binds the service API vendor more tightly to the existing DMA platform.

2.2 Current State of Service Ingestion

At present, we have implemented a DMA Service API editor which allows to

- create new DMA service descriptions
- upload an existing service description file- or URL based
- update service metadata

---

[1] For more information, see: https://github.com/OAI/OpenAPI-Specification (verified 20. 2. 2018)

- show the API documentation, and
- allows to test API calls prototypically

Possible Service description file formats are YAML and JSON, although YAML format is the preferred one. The Service API Editor uses a build-in DMA schema for validation and only correctly validated description data is submitted to the DMA Metadata Converter & Harvester (see Figure 1).

For service documentation, another software module has been prepared, which allows to visualise the API calls in a decent layout. An additional feature provides the possibility to switch between all minor versions of a service API in order to display its differences.

Last, but most convenient tool in the toolset is the service API code generator. It allows to generate client and server stub code for the following languages / frameworks:

- API clients: ActionScript, Ada, Apex, Bash, C# (.net 2.0, 3.5 or later), C++ (cpprest, Qt5, Tizen), Clojure, Dart, Elixir, Elm, Eiffel, Erlang, Go, Groovy, Haskell (http-client, Servant), Java (Jersey1.x, Jersey2.x, OkHttp, Retrofit1.x, Retrofit2.x, Feign, RestTemplate, RESTEasy, Vertx, Google API Client Library for Java, Rest-assured), Kotlin, Lua, Node.js (ES5, ES6, AngularJS with Google Closure Compiler annotations) Objective-C, Perl, PHP, PowerShell, Python, R, Ruby, Rust (rust, rust-server), Scala (akka, http4s, swagger-async-httpclient), Swift (2.x, 3.x, 4.x), Typescript (Angular1.x, Angular2.x, Fetch, jQuery, Node)
- Server stubs: Ada, C# (ASP.NET Core, NancyFx), C++ (Pistache, Restbed), Erlang, Go, Haskell (Servant), Java (MSF4J, Spring, Undertow, JAX-RS: CDI, CXF, Inflector, RestEasy, Play Framework, PKMST), Kotlin, PHP (Lumen, Slim, Silex, Symfony, Zend Expressive), Python (Flask), NodeJS, Ruby (Sinatra, Rails5), Rust (rust-server), Scala (Finch, Lagom, Scalatra)[1]

All modules are fully containerised and can be integrated into the DMA ecosystem according to Figure 1.

2.3 Interfaces and APIs

After user authentication / authorisation, the portal allows to select further actions, which can be

- Create a new service API description
- Add an existing service API description
- Update an existing service API description
- Delete a service API description
- Show documentation of a certain service API major version with all of its minor versions
- Start code generator with certain service API description, and
- Submit service API description to Metadata Converter & Harvester.

[1] For more information, see: https://github.com/swagger-api/swagger-codegen (verified 20. 2. 2018)

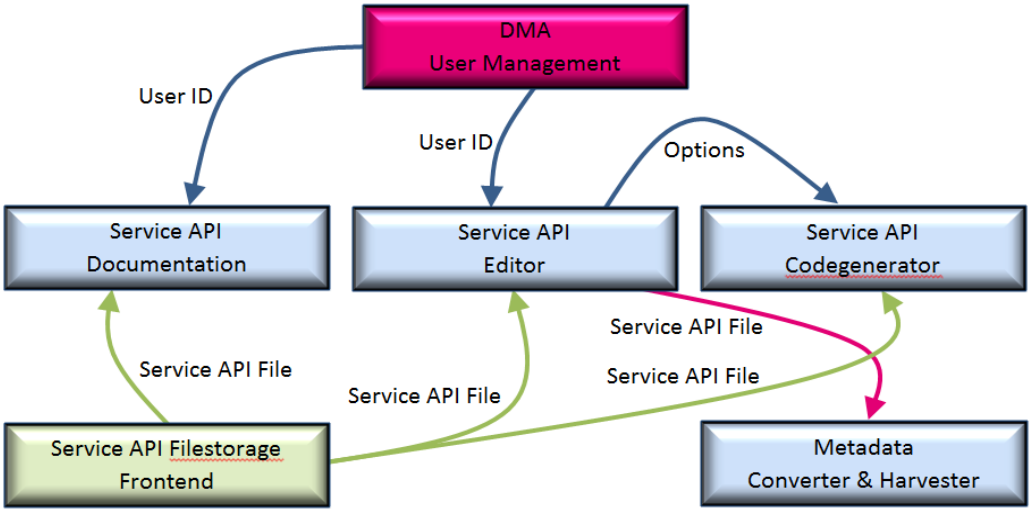

Figure 1: Functional dependencies between service API ingestion tools and portal components

2.3.1 Create, Add or Update a Service Description

The following code fragment depicts an example on how to integrate the Service API editor into the DMA workflow. The HTML file instantiates the service API editor with new or existing service API description. Currently, this is not yet implemented but it will be implemented until autumn 2018.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>DMA Service API Editor</title>
<link href="./dist/swagger-editor.css" rel="stylesheet">
<link rel="icon" type="image/png" href="./dist/favicon-
32x32.png" sizes="32x32" />
<link rel="icon" type="image/png" href="./dist/favicon-
16x16.png" sizes="16x16" />
</head>
<body>
<div id="swagger-editor"></div>
<script src="./dist/swagger-editor-bundle.js"> </script>
<script src="./dist/swagger-editor-standalone-preset.js">
</script>
<script>
window.onload = function() {
const editor = SwaggerEditorBundle({
dom_id: '#swagger-editor',
layout: 'StandaloneLayout',
presets: [
SwaggerEditorStandalonePreset
],
// This URL represents the selected DMA service API file
```

```
url: 'https://dma.serviceAPI.service/openAPI-sha256-
1234567890-v1.yaml'
})
window.editor = editor
}
</script>
// This represents the save and submit button for service API
descriptions
<div id="save-and-submit-to-metadata-converter-and-harvester">
<form action="https://url-to-dma-metadata-converter-and-
harvester.com">
     <input type="submit" value="Start DMA metadata converter
& harvester" />
</form>
</div>
</body>
</html>
```

Listing 1: DMA Service Editor with validation - submission triggers metadata harvesting

2.3.2 Deprecate and Retire a Service

Update existing service description by adding a `deprecated` token to the operations description.

Retirement of a service currently means to remove the service description from the service repository and to trigger a metadata harvesting pass.

2.4 Further Development

Currently the DMA schema does not contain any controlled vocabulary for

- `category`, mapped to `dcat:themeTaxonomy`, test tags are `Category1, .. Category3`
- `theme`, mapped to `dcat:themeTaxonomy`, test tags are `Theme1, .. Theme3`, and
- `tags`, mapped to `dcat:keyword`, test tags are `Tag1,.. Tag3`.

3 Semantic Enrichment and Entity Linking

3.1 Semantic Enrichment

The interpretation of data depends on the context. The metadata is generally not verbose. Semantic Enrichment aims at providing additional context, therefore facilitating related activities, e.g. similarity assessment, recommendation services, search. The enrichment employs the knowledge contained in a thesaurus or, more generally, an ontology.

A thesaurus contains concepts, relations between concepts, and attributes of concepts. Attributes usually contain labels of the concept with a distinguished preferred label. The labels present lexical forms of the concept, for example, “marital status”[1]. Typically a concept is referred to by the usage of its preferred label, i.e. the concept “marital status”. The concepts may have other labels in their attributes: alternative and/or hidden labels.

The relations contain hierarchical (transitive) relations and non-hierarchical. Hierarchical relations between concepts constitute a partial order on the concept, a hierarchy. The concept “marital status” has an ascending branch with concepts “family”, “social questions”. We typically use broader/narrower to denote hierarchical relations. These hierarchical relations enable hierarchical enrichment, i.e. enrichment with broader concepts. For example, if the value of a metadata field contains “description of marital status of population under 30 years” it would be possible to extract additional information that the mentioned description is also about family and social questions. The exact interpretation and implementation of such enrichment depends on the use case. Moreover, hierarchical relations encourage introduction of similarity measures between concepts, for example Resnik similarity (Resnik, 1995), Lin similarity(Lin, 1998).

In the course of the semantic enrichment process, concepts are going to be extracted from the metadata via their labels. Only the concepts contained in the thesaurus are extracted, therefore, the quality of the thesaurus is essential for the concept extraction, hence for semantic enrichment. EuroVoc[2] is used in DMA as the main thesaurus. EuroVoc is a multilingual, multidisciplinary thesaurus covering the activities of the EU. It contains terms in 23 EU languages (Bulgarian, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish and Swedish), plus in three languages of countries which are candidate for EU accession: македонски (mk), shqip (sq) and српски (sr). The total number of concepts is 7159.

At a later stage of the project (M30 of the project) it is planned to employ domain specific thesauri. The domain specific thesauri cover different subject areas and enhance enrichment with domain specific information, labels, and relations. In DMA domain specific thesauri come from pilots, i.e. WP8 and WP9. Following the procedure established for including domain specific thesauri for pilots further domain specific thesauri may be included in the semantic enrichment process.

3.1.1 Implementation: PoolParty eXtractor

After the data is ingested and accepted in the data ingestion pipeline, the metadata is enriched. At this stage we can assume that metadata is validated, and has already the DMA-specific metadata predicates and entities (in the case of controlled vocabularies). The entities which come from controlled vocabularies are saved in the form of URI (see Section 3.2 Entity Linking).

---

[1] http://profit.poolparty.biz/profit_thesaurus/1538

[2] http://eurovoc.europa.eu/

The metadata of datasets and service is enriched. For both types of assets the following metadata fields are being enriched:

- Description (`dct:description`)
- Title (`dct:title`)
- User-generated tags (`dcat:keyword`)

For adding the enrichment to the metadata the stand-off annotation are used, i.e. the URIs of the extracted concepts are stored separately and the original titles, description, tags are not modified. Therefore, any service working with metadata (in particular the recommender service) may continue operating to the original values. This is especially useful in the unusual case when no concepts are found, i.e. no enrichment possible.

NLP interchange format[1] is used for annotations. The predicate `nif:annotation` is used to provide a reference to the knowledge base. Moreover, the positions of the extracted concepts as well as their surface forms are recorded in the metadata. The format of including annotations is best illustrated with an example.

Let the incoming dataset contain the following description:

```
<dataset_uri> dct:description "The Q1 report of the power consumption of
households in Vienna 2016"^^xsd:string.
```

Then the enriched metadata is going to look the following way:

```
@prefix nif:   <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .

<dataset_uri> dct:description <_:description_of_dataset_uri> .
<_:description_of_dataset_uri>
a             nif:Context, nif:RFC5147String ;
nif:isString "The Q1 report of the power consumption of households in
Vienna 2016"^^xsd:string .

<_:description_of_dataset_uri#char=7,12>
a                     nif:RFC5147String ;
nif:anchorOf          "report"^^xsd:string ;
nif:beginIndex        "7"^^xsd:int ;
nif:endIndex          "12"^^xsd:int ;
nif:annotation    <http://eurovoc.europa.eu/2891>;
nif:referenceContext  <_:description_of_dataset_uri> .
<_:description_of_dataset_uri#char=27,37>
a                     nif:RFC5147String ;
nif:anchorOf          "consumption"^^xsd:string ;
nif:beginIndex        "27"^^xsd:int ;
nif:endIndex          "37"^^xsd:int ;
nif:annotation    <http://eurovoc.europa.eu/139>;
nif:referenceContext  <_:description_of_dataset_uri> .
<_:description_of_dataset_uri#char=42,51>
a                     nif:RFC5147String ;
nif:anchorOf          "household"^^xsd:string ;
nif:beginIndex        "42"^^xsd:int ;
nif:endIndex          "51"^^xsd:int ;
nif:annotation    <http://eurovoc.europa.eu/1864>;
nif:referenceContext  <_:description_of_dataset_uri> .
```

[1] http://persistence.uni-leipzig.org/nlp2rdf/

```
<_:description_of_dataset_uri#char=56,61>
a                       nif:RFC5147String ;
nif:anchorOf            "Vienna"^^xsd:string ;
nif:beginIndex          "56"^^xsd:int ;
nif:endIndex            "61"^^xsd:int ;
nif:annotation     <http://eurovoc.europa.eu/6202>;
nif:referenceContext   <_:description_of_dataset_uri> .
```

The EuroVoc thesaurus is already available at dma.poolparty.biz, the project ID of EuroVoc is `1DF19BCD-681D-0001-6277-102018CC1615`. In order to extract concepts the APIs of PoolParty are implemented, deployed and available. The description may be found at https://dma.poolparty.biz/extractor/api and https://dma.poolparty.biz/PoolParty/api. In particular, the method https://dma.poolparty.biz/extractor/api/extract (description: https://dma.poolparty.biz/extractor/api?method=extract-d9a1833dd1dd59d1baf943359e474cfd) is used to extract the concepts, return their URIs and positions.

In order to work with the enriched metadata the following API call is available:

https://dma.poolparty.biz/PoolParty/api/thesaurus/{projectID}/concept (description: https://dma.poolparty.biz/PoolParty/api?method=getConcept-7b41dba9c8550a3bb8fef4d3bb5fc4de) returns the information about the concept, including broader and narrower concepts, related concepts, labels, description and possible notes.

A PoolParty instance for supporting the DMA project is deployed at SWC premises and already available for usage. Until September 2018 it is planned to implement an annotation service that would wrap the extract call and return the annotations using NIF (as described above). Until 2019 it is planned to provide a dockerised PoolParty instance to be deployed at the DMA infrastructure.

3.2 Entity Linking

When a service or a dataset is being ingested into DMA platform its metadata is validated. One of the validation steps is a check of the values (objects of triples) of the metadata. Some values are not restricted, i.e. can be any strings (for example, user generated tags) whereas others are restricted (for example, language). For the restricted ones PoolParty is used to store all the allowed values. This way we ensure interoperability and conformity of metadata.

The metadata values can readily be equal to one of the allowed values or there could be an entity linking step in order to transform the incoming values into the allowed ones. In order to prepare such linking rules an entity linking tool will be implemented and deployed at DMA infrastructure until February 2019.

3.2.1 Implementation: PoolParty Thesaurus Server

Currently it is possible to check if a value is contained among the allowed values or not. For this purpose a controlled vocabulary containing all the allowed values should be modelled and stored in PoolParty. In order to perform this check the method https://dma.poolparty.biz/extractor/api/suggest (description https://dma.poolparty.biz/extractor/api?method=suggest-8558e880b146229e43a60f20627f1f74) shall be called with the parameters:

- `language=en`
- `searchParameters=[{"matchingStrategy":"EXACT","searchString":<desired search string>}]`

If the response contains any results then a concept with the specified label exists in the thesaurus.

The following metadata fields expect a value from a controlled vocabulary:

- Tags (`dcat:keyword`)
- Theme (`dcat:theme`)
- Access rights (`dct:accessRights`)
- License (`dct:license`)
- Price model (`dmav:PriceModel`)

In the entity linking process the values are checked. If the checks are successful all the string values are substituted by URIs of the respective concepts from the DMA controlled vocabularies.

Currently PoolParty instance is already available. As the next step the entity linking interface is planned. The interface will enabled the DMA users to prepare mapping rules via web interface. The interface will be developed and deployed until February 2019.

4 Data Clustering and Slicing Service

The data clustering and slicing service is a non-core service (see Figure 2) to be offered by DMA to its data customers. The data clustering component identifies naturally-occuring clusters in CSV datasets, while the data slicing component lets data customers download subsets of given datasets, based on the identified clusters. The objective of the data clustering and slicing service is that in case of very large datasets, users would not have to purchase, download, and process the entire dataset. Instead, they will be able to purchase subsets of it, containing those items, which are most relevant to them (thus potentially saving money, time, and computational resources).

Cluster analysis is a resource demanding task, especially for very large datasets. Therefore, the data clustering and slicing service runs cluster analyses only once per resource and saves the results of the analyses for further reuse. Saving each subset as self-contained CSV files would increase the size exponentially with the number of dimensions. Hence, the storage space requirements for storing all CSV datasets in the DMA infrastructure is impracticable in the general case. Since the data clustering and slicing service is primarily intended to be used for very large datasets, the increased need for storage space would be impractical. Therefore, the results of the cluster analyses are saved in a separate CSV file (one file for each resource), with each row storing the respective cluster ID of the original resource, while being the same order as the original file.

The generation of the subsets takes place on-the-fly: once the customer selects a cluster for download, the data slicing component parses the dataset and the cluster data file row-by-row and sends only those rows to the customer that are in the cluster they are interested in.

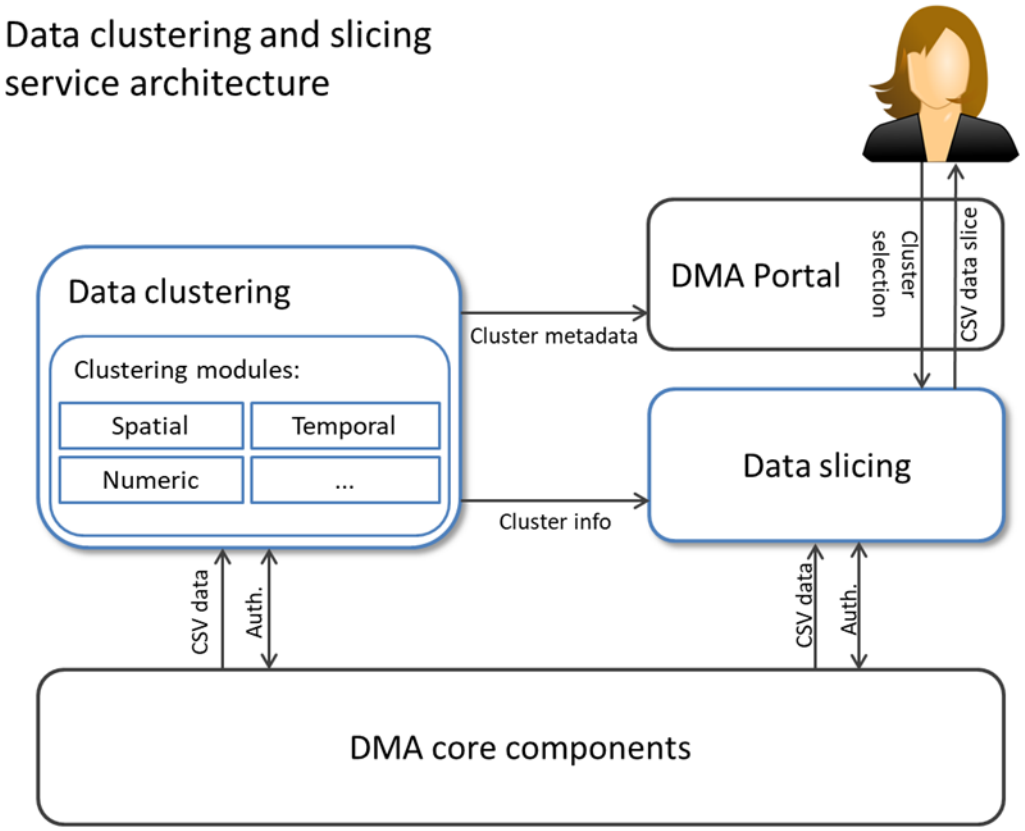

Figure 2: Data flow between the components of the data clustering and slicing service

4.1 Current status

Currently, a prototype of the data clustering and slicing service is deployed to the DMA cloud. This is a proof of concept user interface is available at http://slicendice-slicer-sandbox.apps.dma-cloud.catalysts.cc/poc/slicer.php

The data clustering component as well as the data slicing component are implemented as two separate microservices, on the DMA cloud. The separation of concerns regarding these components is due to the following reasons:

- their performance profiles differ (the data clustering runs once per dataset, but is resource-demanding; the data slicing runs every time a dataset's subset is queried but is not resource demanding)
- data analyses of the data clustering is implement in Python Flask, and while web delivery tasks of the data slicer is implement in PHP (data clusterer is implemented by Python Flask and Connexion, data slicer is by PHP and Slim framework, the basis of both generated by Swagger Code Generator)
- the analyses produced by the data clustering might in the future be reused by other services, independent of the data slicing.

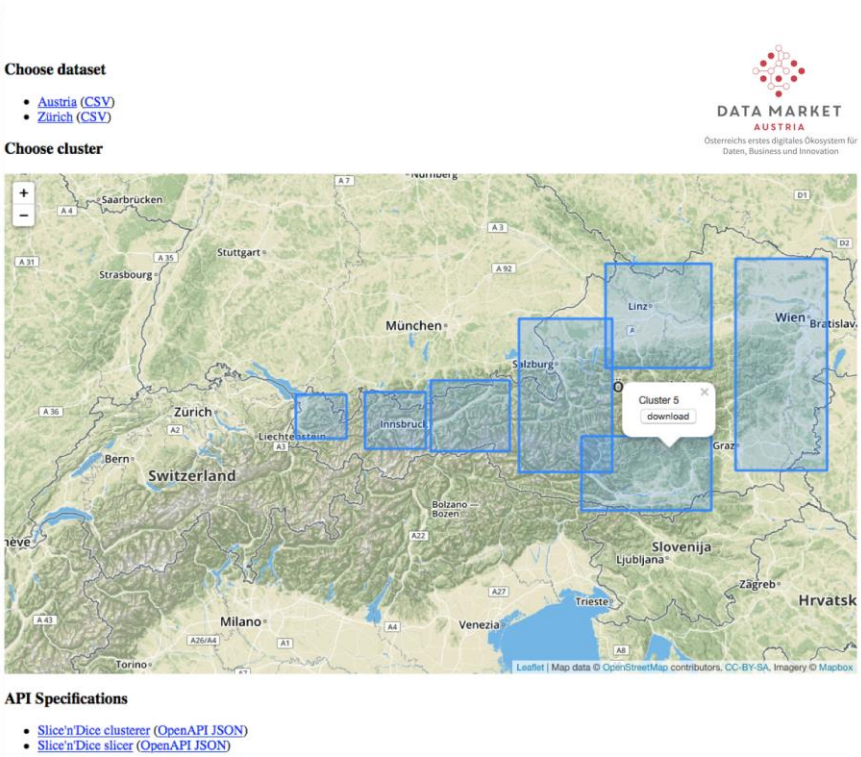

Figure 3: Example result visualisation of the service

4.1.1 Data clustering component

Spatial clustering (on the basis of geographic coordinates) was the use-case chosen for the data clustering and slicing service prototype, with the aim of identifying naturally-occuring geographical clusters in datasets.

A controlled vocabulary is used for identifying CSV columns that contain coordinates by comparing the column header value with the controlled vocabulary. The content of relevant columns are assumed to be in the WGS 84 standard (Decker, 1986), the standard used by most contemporary GPS

devices. Coordinate records that do not contain standard coordinate information are not analysed - the clusterer saves an invalid value (-1) for such rows.

Once columns containing geographical information are identified, the data clustering component identifies clusters in them using k-means clustering (Hastie et al., 2001), currently identifying a pre-defined number of clusters, with the analysis of the ideal number of clusters to be implemented later. Cluster information is then saved in a separate CSV file, each row containing the cluster ID of the corresponding row of the original CSV file. Once the clusters are identified, the bounding boxes of each cluster are calculated and saved in a JSON metadata file.

If the data clustering component is called to return a dataset's cluster information or cluster bounding box metadata, the data clustering component first looks if cluster information and bounding boxes have already been identified and saved for the requested dataset. If yes, it returns the stored files, if not, it performs the cluster analysis, saves the files and returns the queried information.

4.1.2 Data slicing component

The data slicing component is to be queried with the dataset's ID and the cluster ID to be downloaded. It fetches the CSV file containing cluster information from the data clustering component, and fetches the dataset from DMA ecosystem (to be implemented - for proof of concept the component currently fetches the same dataset used by the clustering component, hosted by the service). The slicing component then parses the two CSV files row-by-row parallelly and when the cluster ID in the cluster CSV matches the cluster ID requested by the customer, it pushes the corresponding line from the dataset CSV to the customer.

The data slicing component is implemented, using only stream-based file operations for good performance and scalability. The source CSV files are not downloaded by the service as a whole - they are downloaded, read and then discarded row by row. The output CSV subset is not buffered by the service - the CSV rows are pushed to the user on-the-fly, using HTTP1.1 chunked transfer encoding. Thanks to stream-based file operations the data slicing component performs well with datasets of any size.

The data clustering and slicing service is currently not fully integrated in the DMA ecosystem. It runs on the DMAcloud and service discovery is enabled, but datasets are not fetched from Conduit[1] and authentication is not implemented, since during the time of the development the ecosystem was not mature enough to integrate services with it. Instead a few example datasets are integrated in the service, to prove the concept, and authentication is disabled.

4.2 API Description

4.2.1 Data Clustering Component

The implemented, detailed specifications can be found at http://slicendice-clusterer-sandbox.apps.dma-cloud.catalysts.cc/api/v1.0/ui/#!/enhancement/get_cluster_data

4.2.1.1 Bounding boxes of identified clusters

Request

[1] See DMA deliverable D5.3

Method	**URL**
POST	api/v1.0/slicendice/coordinates/getBoundingBox/<process_id>
Parameters/Request Body	
Type	**Params**
HEAD	auth_key
auth_key The auth_key that was given in response to /api/login	
QUERY	process_id
process_id The process_id Process ID of the dataset submission process	
Response	
Status	**Response**

200	[
	{
	"cluster_id": 0,
	"maxx": 16.577513,
	"maxy": 48.322666,
	"minx": 16.181831,
	"miny": 48.117908
	},
	{
	"cluster_id": 1,
	"maxx": 17.577513,
	"maxy": 49.322666,
	"minx": 17.181831,
	"miny": 49.117908
	}
	]

202	{"error":"Clustering for this dataset is already being processed. Try later"}
400	{"error":"Process ID does not exist / is missing."}
401	{"error":"Unauthorised. Auth key is missing"}
404	{"error":"No asset with the given job process ID found."}
500	{"error":"An error occurred."}

4.2.1.2 Get Cluster Data

Returns a CSV file of cluster information on the source dataset.

Request	
Method	**URL**
POST	api/v1.0/slicendice/getClusterData/<process_id>
Parameters/Request Body	
Type	**Params**
HEAD	auth_key
auth_key The auth_key that was given in response to /api/login	
QUERY	process_id
process_id The process_id Process ID of the dataset submission process	
Response	
Status	**Response**
200	X-DMA-cluster 0 6 -1 6 2
202	{"error":"Clustering for this dataset is already being processed. Try later"}
400	{"error":"Process ID does not exist / is missing."}
401	{"error":"Unauthorised. Auth key is missing"}
404	{"error":"No asset with the given job process ID found."}
500	{"error":"An error occurred."}

4.2.2 Data slicing component

The implemented, detailed specifications can be found at http://slicendice-slicer-sandbox.apps.dma-cloud.catalysts.cc/poc/OpenAPISpec.html

4.2.2.1 Get slice

Parses the original resource and the cluster data CSV file parallelly and streams only those rows of the original file that have the selected cluster ID in the same row of the cluster data CSV file.

Request	
Method	**URL**
POST	api/v1.0/slicendice/getSlice/<process_id>/<cluster_id>
Parameters/Request Body	
Type	**Params**
HEAD	auth_key
auth_key The auth_key that was given in response to /api/login	
QUERY	process_id
process_id The process_id Process ID of the dataset submission process	
QUERY	cluster_id
The cluster_id of the cluster of the dataset to be downloaded.	
Response	
Status	**Response**
200	CSV content with only the relevant rows
400	{"error":"Process ID does not exist / is missing."}
401	{"error":"Unauthorised. Auth key is missing"}
404	{"error":"No asset with the given job process ID found."}
500	{"error":"An error occurred."}

4.2.3 Service discovery

Both microservices expose the following endpoint that returns the OpenAPI specification of the API.

Request	
Method	**URL**
GET	api/v1.0/slicendice/OpenAPISpec/

Parameters/Request Body	
Type	**Params**
`HEAD`	`auth_key`
The `auth_key` that was given in response to `/api/login`	
Response	
Status	**Response**
`200`	The OpenAPI specification of the API in JSON format

4.3 Development Plan

The future development of the data clustering and slicing service during the runtime of the DMA project foresees the following steps:

1. Full integration of the service into the DMA ecosystem (fetching and sending data, authentication)
2. Create clusterers for other data types in addition to geographical coordinates (i.e., date and time values, numbers, taxonomies, etc.)
3. Integrate the data slicing component in the DMA portal: creating a standard UI.
4. Expanding the controlled vocabulary used for identifying CSV column containing coordinates by header value
5. Consider the possibility of supporting other coordinate standards in addition to WGS 84.

In addition to the data clustering and slicing service, two additional non-core DMA services are planned by DUK, based on the outlines and requirements as well limitations stated in Deliverable 6.1. The first of these services is a kind of Once-only download of datasets, to facilitate different ways of accessing static as well as dynamic data (i.e. streaming data) to support additional service and product models on DMA. The second service is to offer Dockerised templates, schemes and tools that enable third-parties to quickly start interacting with the DMA platform programmatically, as well as building services to be connected with DMA. However, these two services and their feasibility strongly depend on the outcome and possibilities as well as limitations provided by the platform/portal development out of WP4.

5 Earth Observation Services

5.1 Motivation

Forests play an important role for society; they prevent floods, support water production, offer protection from avalanches and rockfall, are part of the global CO_2 cycle, provide recreational space and lay the foundation for a whole industry branch (cf. Mayer 1999, 296). In Austria 47,6 % of the total area are covered with forests, with the province of Styria having the highest coverage (61,4%) of all provinces (cf. Sebauer 2013, 14; Russ 2011, 3). The forest in Austria has a strong protection capability against erosion and subsequent risks of flooding, mudslides and avalanches and is essential for the preservation of the Alpine ecosystems. In the context of forest damage windfall is a sudden event very often followed by subsequent bark beetle infestations. Both events impact the structure of forests and, thus, diminish the protective forest functions mentioned. To react adequately, forest managers and environmental agencies require sudden or regularly updates on forest parameters such as crown closure or forest gaps, as well as the size of damaged areas (including loss of timber volume in damaged areas) to set up the respective forest management activities and logistics. Due to the high cost of data collection for conventional inventories these parameters were prepared only for small forest sites and at infrequent intervals or from sample data of the Austrian Forest Inventory (ÖFI). While providing key statistical data on larger regions, these surveys do not allow the derivation of the regional (wall to wall) distribution of the various forest parameters.

Relevant for wind induced damage to forests in Europe are winter storms, foehn storms and summer storms. With a diameter of 1000-3000 km winter storms possess the largest scale for potential damage (cf. Schindler et al 2012, 57). As for storm intensity, intensity of damage may be attributed to wind gusts more than to mean wind speed (cf. Gebhardt et al 2011, 1125). Nevertheless not only wind but also tree attributes, stand situation, site conditions and environmental factors are important factors for tree vulnerability.

Rockfall is a rapid and high-energy geomorphic process which may cause substantial damage and even fatalities. Although single rockfall events usually affect only small areas, especially in alpine regions wide areas are potentially endangered. Thus, decision makers and stakeholders are craving for models and tools producing spatially distributed process information (usually digital layers and maps) supporting rockfall assessment (Chung & Fabbri 2006). Depending on (i) scale, (ii) availability and quality of input data and (iii) financial efforts, different types and levels of rockfall zonation are recommended. At regional scale (1:25,000; 1:50,000), indicative hazard maps have turned out to be useful pre-disaster mitigation tools (Bell et al. 2013, Guzzetti et al. 2003). Such maps are rather evaluated on the reliability with regard to frequently occurring low-magnitude events (e.g. small boulder sizes with short runout length) than on high-magnitude (worst case) scenarios (maximum potential boulder size with maximum runout length). However, low-magnitude events are very sensitive to changes of the vegetation cover as small changes may cause substantial effects on the runout lengths. Frequent updating of the required forest parameters thus is needed for reliable hazard information.

The assessment of forest damage by means of SENTINEL 2 satellite data can therefore be considered as an important step forward for a sustainable management of forests (cf. international Alpine Convention).

5.2 Service Descriptions

Within the DMA pilot “Complex Data Management for Forestry Services” (WP9) four services are defined, namly Forest(Change) Monitoring, Storm Damage Resilience, and Updated Rockfall

Propagation Modelling. These services can be part of Forest Management and Action Planning plattform.

5.2.1 Forest (Change) Monitoring

The monitoring of forest areas offers a very efficient tool to obtain information of possible change caused by biotic or abiotic mechanisms. However, it is a-priori not known where changes or forest damage can occur. In order to overcome this drawback it is necessary to observe very large areas on a regular base. In this context the new SENTINEL satellite system offers a very powerful tool with high spatial, spectral and temporal resolutions. As this system has a swath width of up to 290 km and a repetition rate of every 5 days the frequency for updates can fully satisfy forest monitoring needs. Storage capacities have to be adapted to the huge amount of data recorded for all systems, including Landsat TM.

5.2.2 Storm Damage Resilience

Nowadays complex data sets are available for many forest related applications. With respect to storm events this situation is a prerequisite to analyse storm damage resilience. Forest experts emphasise the need of certain key parameters for this evaluation. For example, data on wind speed, topography or forest features are a prerequisite, together with satellite derived information, to model this resilience. From the availability of the data these parameters can be grouped into:

- environmental factors (e.g. such as wind speed, geomorphological features, geological / pedological information)
- tree attributes (e.g. such as tree type involving the root system, tree height)
- stand situation (e.g. development stage, species mixture, crown pattern, stand structure, intensity of thinning)

From the above listed input data sets the resilience of forest areas against potential storm events can be modelled and thus the foresters can obtain important information for their management plans.

5.2.3 Updated Rockfall Propagation Modelling

Area-wide high-resolution rockfall modelling results based on LiDAR data are available for the whole Province of Styria. Two main aspects were taken into account: (a) the identification of potential source zones, and (b) the estimation of rockfall propagation zones. The runout distances were modelled by velocity calculation based on a one parameter friction model. Whereas potential source zones do not change within short time, the modelled run-out zones strongly depend on friction values based on terrain roughness and forest characteristics (e.g. treetop number per unit area, crown coverage, height of upper layer, vertical forest structure). The service will consist of updated rockfall propagation models whenever relevant forest changes are recorded (Service 1).

5.2.4 Forest Management and Action Planning

Frequently updated data sources as represented by the outputs of Services 1 to 3 will give rise to optimised services and decision support systems for action planning and implementation – not only for forestry purposes but as well as for e.g. hazard and risk prevention using technical measures. Stakeholders and forest experts can then react very fast, using this type of information, in order to take countermeasures and to obtain an overview of the economic and environmental impact of the storm damage.

5.3 Primary Content/Data Involved

As already mentioned three service cases are envisaged to be developed in order to support the forest authorities with necessary information for their forest management plans. They encompass a change map, updated regularly or on demand, a forest resilience map, showing the resilience of forest areas against the impact of storm events, and finally an updated rockfall propagation map, which is an update of existing rockfall maps. The respective DMA user for these services is the Forestry Board of the Government of Styria ("Landesforstdirektion") which has a mandate for forest management in Styria. They support the service development with input data from various sources, such as LiDAR data (see also below).

5.3.1 Up-to-date Satellite Data

The possibility to get cost free data from Sentinel 2 A/B satellites brought advantages like:

- high recording frequency with recording intervals of 5 days,
- huge swath width of 290km,
- adequate spatial resolution of 10m, and
- high spectral resolution with 12 bands.

As can be seen in Figure 4, the spatial resolution of SENTINEL 2 is significantly better than of the Landsat Thematic Mapper. Although it is not very high resolution imagery its resolution fits very well to the services to be developed.

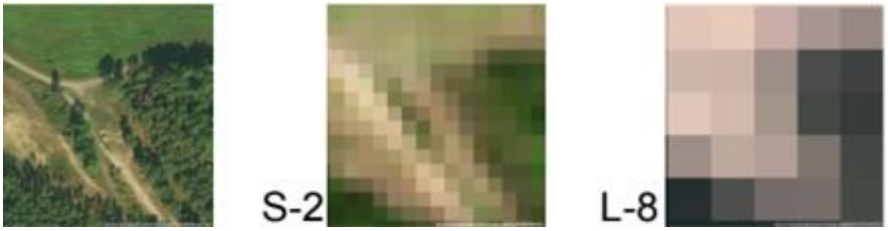

Figure 4: Comparsion of spatial resolution of different optical sensors.

Sentinel-2A in combination with Sentinel-2B provides a 5-day repeat coverage of Earth's land areas, where Landsat-8 has a 16 days repeat coverage. A high frequency is a prerequisite for change detection on demand and is also needed if the forest of interest is located in a highly cloudy region.

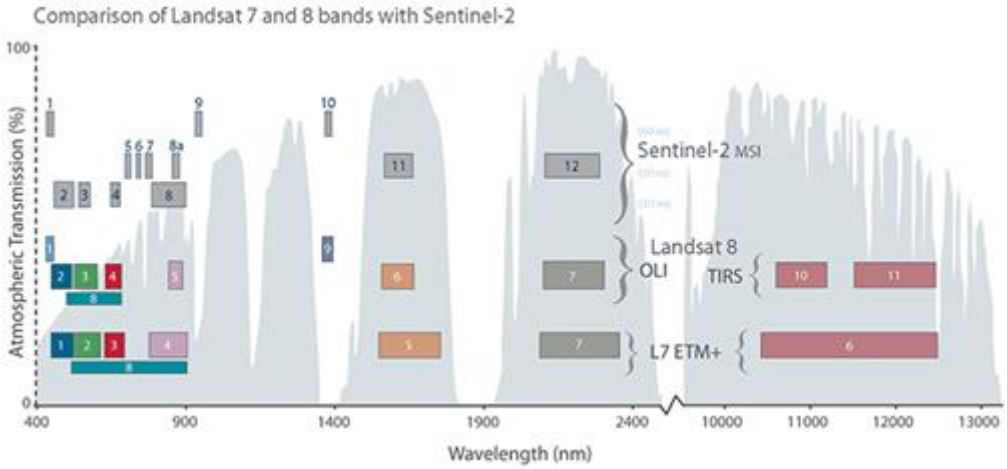

Figure 5: Comparison of Landsat 7 and 8 bands with Sentinel-2.

The Sentinel-2A/B sensors monitor wavelengths in the "red-edge" (red and near-infrared bands) are specifically designed to monitor land cover. As well as monitoring plant development and change, Sentinel-2 is used to map changes in land cover and to monitor the world's forests. For instance, it also provides information on pollution in lakes and coastal waters. Images of floods, volcanic eruptions and landslides so to contribute to disaster mapping and helping humanitarian relief efforts.

5.3.2 Surface Model from ALS Data

All height data are derived from the laserscanner campaign and are received from the department of Geoinformation Steiermark. The different tiles were acquired between 2009-2012.

- DSM (Digital Surface Model): Additionally to x,y values each pixel has a z value representing elevation including vegetation and man-made features, i.e. height of features "above ground".
- DTM (Digital Terrain Model): As opposed to a DSM, a DTM has z values for ground surface heights, i.e. "ground" height".
- nDSM (normalised DSM): An nDSM is the difference between a DSM and a DTM. It therefore contains z values for heights above ground, i.e. tree height.

5.3.3 High Precision Forest Parameter from Airborne Laserscanning (ALS)Data

From the above described input nDSM several forest parameters were derived within the project „Ableitung forstlicher Parameter aus Laserscanner Daten und Erstellung von Gefahrenhinweiskarten für die Steiermark". They encompass the parameters upper height of trees, vertical structure, mean height of trees, species mixture or growing class.

For the rockfall propagation modelling, the required friction parameters are deduced from relevant forest parameters (treetop number per unit area, crown coverage, height of upper layer, and vertical forest structure) which are based on the high resolution Airborne Laserscanning (ALS) Surface Model.

5.3.4 Forest Maps

Currently forest maps are not available for the investigations.

5.3.5 Geological Data

A DTM based on ALS with a spatial resolution of 1 m, provided by Geoinformation Steiermark, is available. This dataset is used both for the identification of potential rockfall detachment areas and the terrain dependent modelling of the runout zones.

Furthermore, a geological basemap with 1:50.000 reference scale (provided by Geoinformation Steiermark) is used for the disposition modelling (identification of potential rockfall detachment areas), and the estimation of boulder sizes. The identification of rockfall detachment areas is based on the definition of slope threshold values. Of course, this approach may be refined by considering morphometric parameters, the geotechnical properties of the release areas, and the tectonic situation (e.g. fabric and discontinuities to be investigated in the field) for more detailed analyses on local scale.

5.3.6 Result from Windfall Modelling

Windfall data are not available due to processing problems at ZAMG. As a consequence of this fact the input variable “wind data” was not used for the modelling.

5.4 Current Implementation

5.4.1 Implementation Forest (Change) Monitoring

For the “Forest Change Monitoring” a concept was elaborated and the relevant modules identified. Currently there are two main modules in realisation in order to successfully apply the service. They comprise the “Minnaert Correction” and the “Change Detection” module.

The “Minnaert Correction” module (see Figure 6) is very important for the change detection process because it normalises the illumination effects, which are especially in mountainous environments have a strong impact on the land surface. This approach is, thus, a prerequisite for any remote sensing processing tasks over large areas. The “Change Detection” module will be developed within the next period of the project and will complete the service “Forest Change Monitoring”.

For the “Forest Resilience” a modelling approach was developed which is based on the above listed variables. Literature names numerous variables concerning vulnerability. As already mentioned they can be divided into:

- tree attributes (e.g. tree type involving the root system, tree height)
- stand situation (e.g. species mixture, stand structure, intensity of thinning, edges between stands)
- site conditions (e.g. geological/ pedological information, topography)
- environmental factors (e.g. wind (gust) speed, water logging following intensive precipitation, temperature affecting frozen soil)

Not all variables have the same explanatory power: the most important factor seems to be tree height (cf. Gardiner et al. 2013a, 4 and Hale et al 2016, 28/37). A higher percentage of conifers seem to make a stand more vulnerable as conifers appear to be more susceptible (cf. Indermühle et al. 2005, 9; Gardiner et al 2013, 38; Schmoeckel 2005, 17). Concerning vertical structure authors do not agree, some consider it important (cf. Hanewinkel et al. 2014, 531; Dvořák et al 2001, 447), whereas others believe it to be negligible (cf. Gardiner et al. 2013, 4). In several studies forest operations, disease, insect outbreaks and previous damage from snow or wind had a temporary destabilizing effect on the (neighboring) trees (cf. Schmoeckel 2005, 17; Pasztor et al .2015, 11; Suvanto et al. 2016, 23). Wind load is generally highest behind edges, the depth of penetration depending on stand

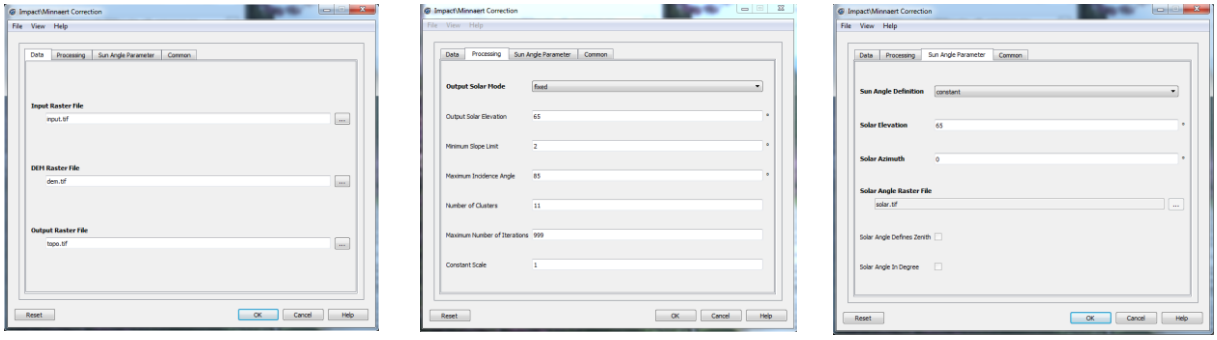

Figure 6: GUI for the "Minnaert Correction"module.

density (cf. Mayer et al. 2010, 73). Topography plays an important role for exposure to wind, causing the wind to accelerate or slow down and offering shelter or exposure.

The goal for this application is to weight the variables that are important and where data is available resulting in overall vulnerability for the test area. Currently, for this model the following variables are considered to be important, as there are:

- stand height
- percentage of conifers
- vertical structure
- slope
- edges and corresponding buffers for exposure
- TOPEX (topographic exposure)

After calculation or extraction all variables are available on pixel level with a resolution of 10x10m. The weighting of these variables is conducted following their rating in literature and is displayed in Figure 7.

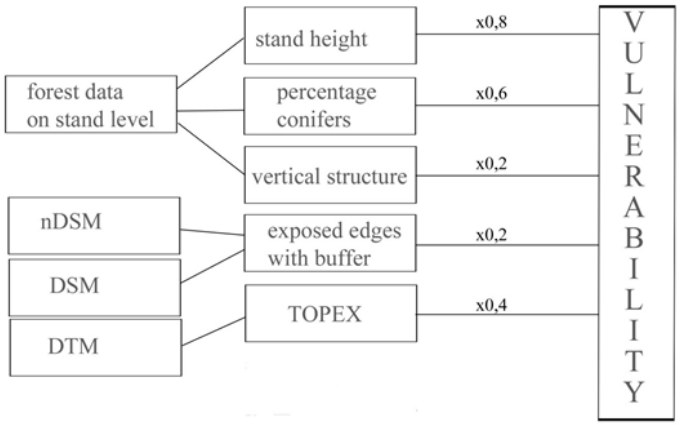

Figure 7: Weighting of variables for total vulnerability (Elisabeth Hafner 2017).

The input variables for the model need various computing and procssing resources and it has to be recognised that for large areas the hard- and software facilities have to be adapted. The percentage of coniferous was derived from satellite data (RAPIDEYE) for the entire Styrian region. The parameters stand height and vertical structure were derived from LiDAR data. The vertical structure (see Figure 8) is a measure of the vertical structure within a stand and defines if a stand has either a uniform or a non-uniform structure. This parameter, in three categories, has a strong influence on the stability of a forest stand.

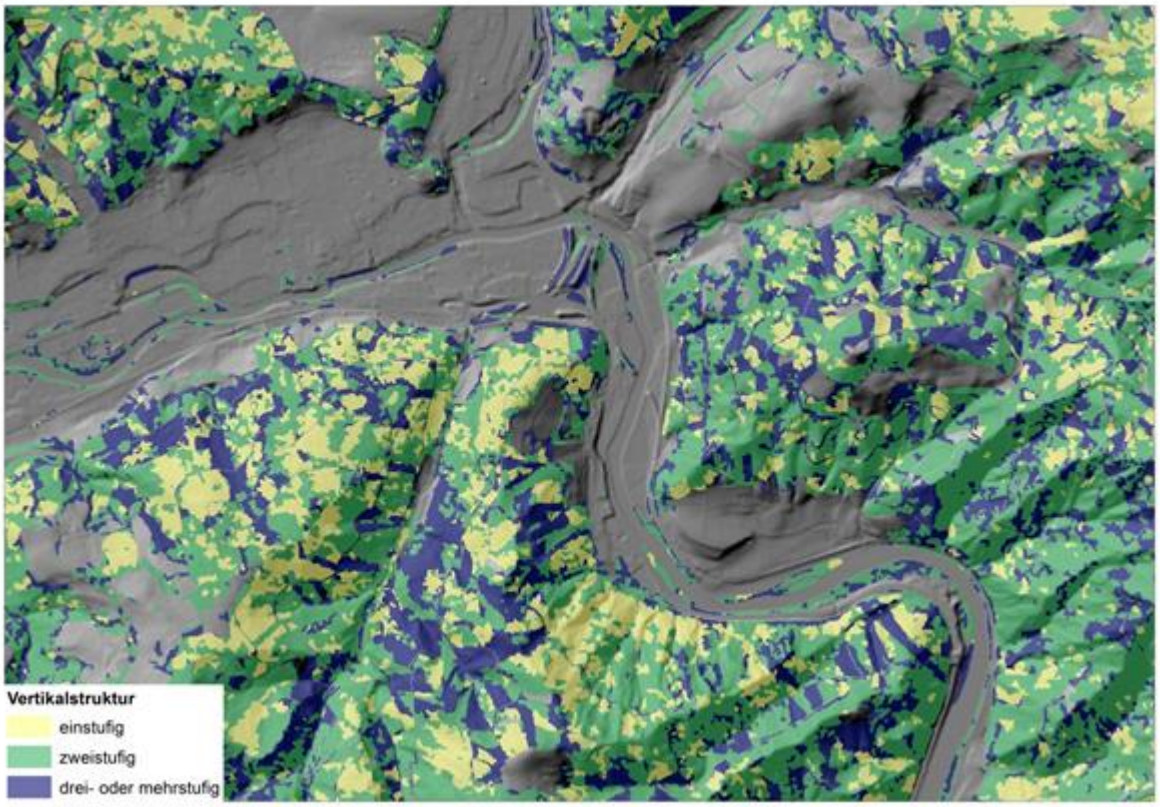

Figure 8: Vertical structure (three categories) within the test area (Elisabeth Hafner 2017)

The variable “exposed edges with buffer” (see Figure 8) was calculated from the LiDAR nDSM and shows the exposed edges in eight different directions. This parameter is the most important one, because the exposure determines whether the wind force can penetrate a forest stand or not. According to forest experts, it is not likely that damage occurs if edges are not exposed.

The varibale TOPEX is a measure for the exposure of a stand positions and was calculated with a distance of 1000m, as a so-called “TOPEX to Distance” (see Figure 10) parameter. The distance was calculated for the eight main cardinal directions, and was accumulated to one single map. It can nicely be observed that the most exposed regions are on the peak and ridge areas and the least exposed ones in the valleys.

After the calculation of the six variables it was a necessary step to analyse these parameters according to their relevance before they were incorporated into the model. That means, each of the parameters was analysed with respect to the locality and the storm event taken place in the region. This investigation was based on a past storm event where the damage was derived and mapped. The damaged and non-damaged areas were compared with the maps of the variables in order to find a correlation between them. Based on the outcome of this analysis the variables were weighted accordingly and incorporated into the model.

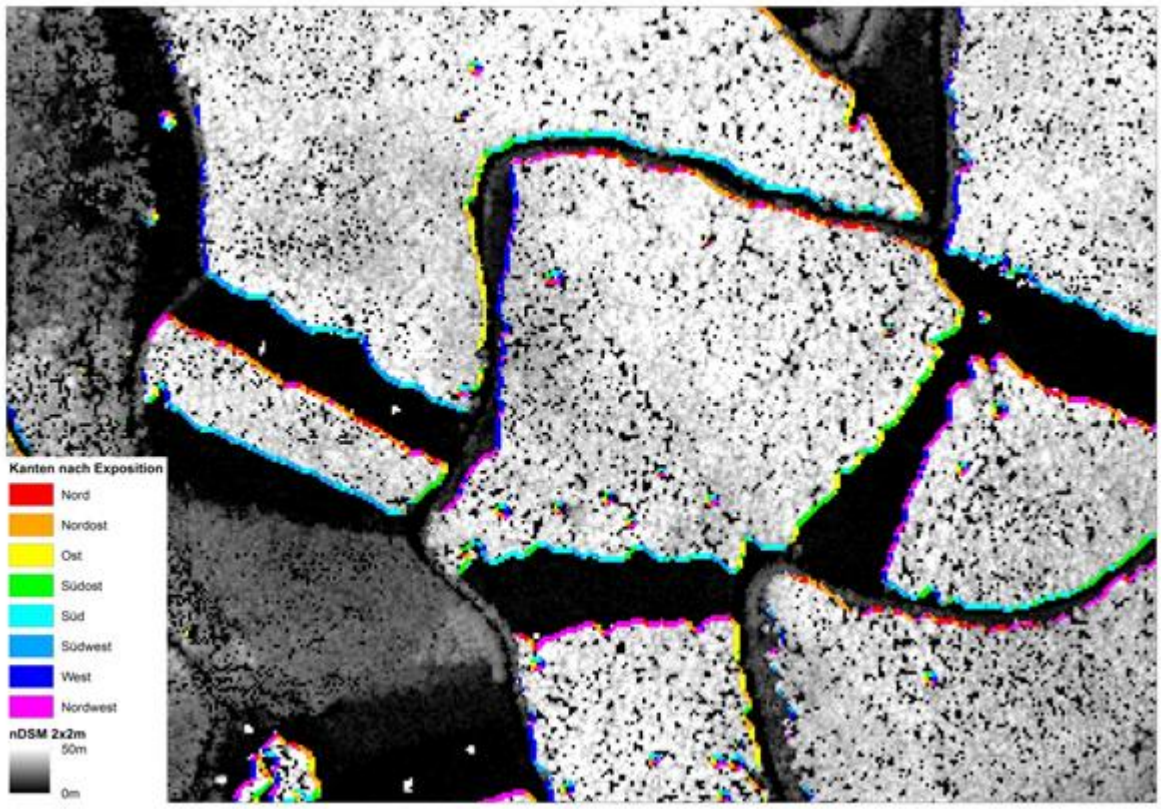

Figure 10: Exposed edges in eight categories within the test area (Elisabeth Hafner 2017)

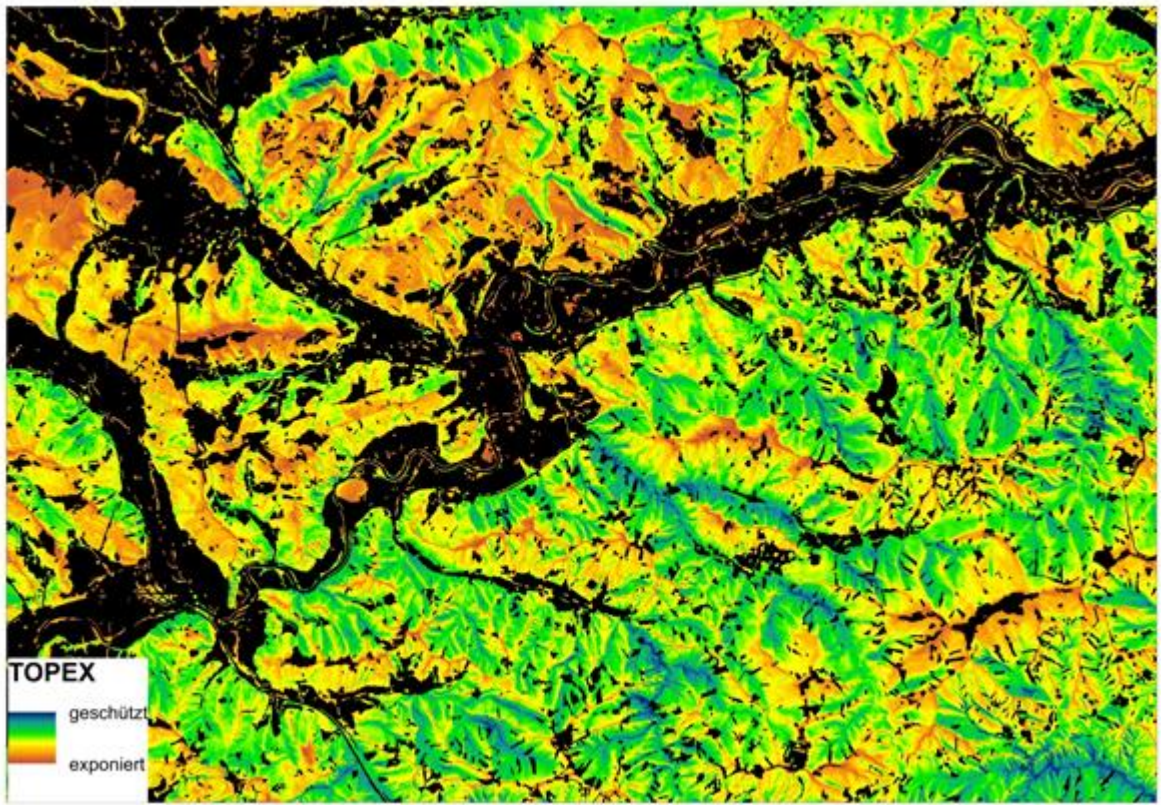

Figure 9: Resulting map of the "TOPEX to Distance" parameter within the test area (Elisabeth Hafner 2017).

5.4.2 Implementation of the Storm Damage Resilience Service

The vulnerability of the region was finally calculated based on the modelling results and can be seen in Figure 11. The service "Storm Damage Resilience" is built-upon a knowledge based model which has been developed on expert knowledge and scientific literature. A novel edge detection algorithm has been developed in order to classify the very high resolution LiDAR data. A validation was performed in a region where a past storm event (summer thunderstorm 8th Juli 2015) caused damage. These damaged areas were used to compare the outcomes of the modelling (as can be observed in Figure 11).

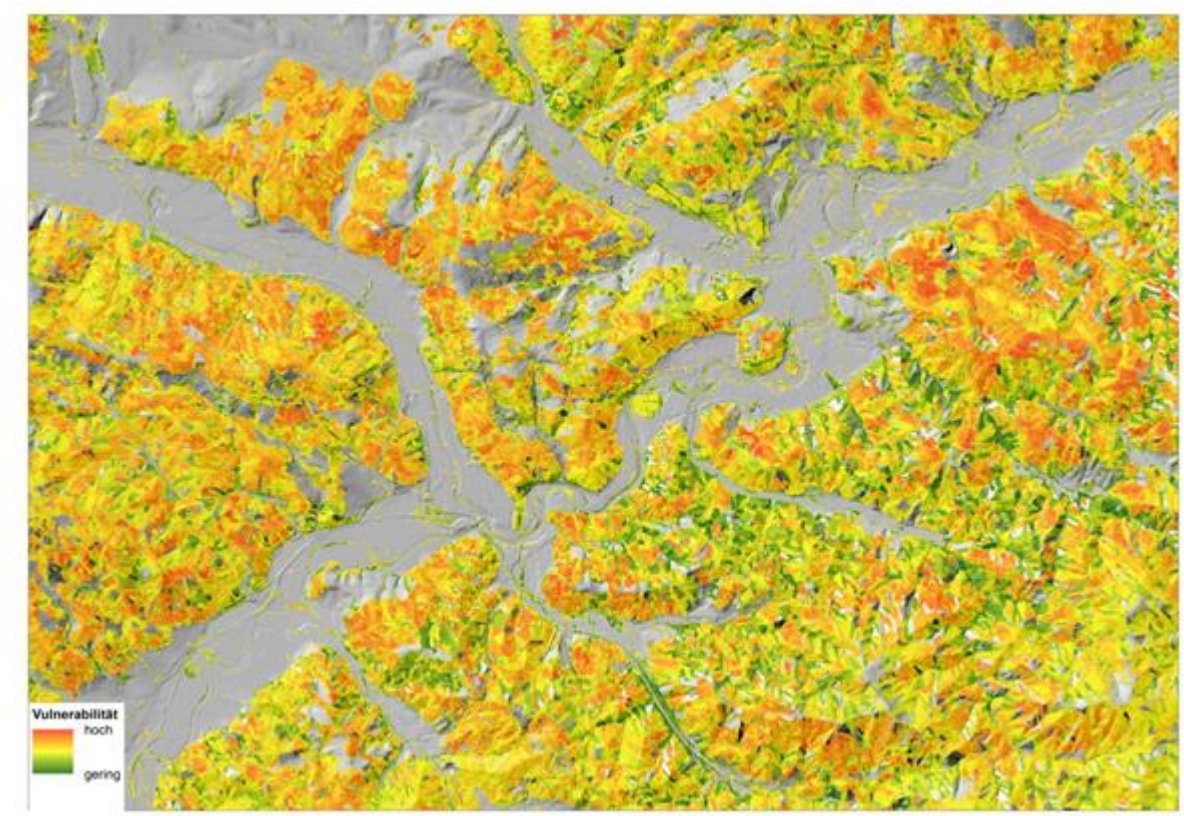

Figure 11: Resulting map of the vulnerability within the test area (Elisabeth Hafner 2017).

5.4.3 Implementation of the Rockfall Propagation Service

For the "Updated Rockfall Propagation" modelling a grid based slope profile velocity calculation incorporated in a stochastic multiple flow direction system is used. The required topographic information is derived from a DTM based on ALS. As the used rockfall algorithm is not able to calculate potential upward trajectories, sinks in the DTM are filled using a standard hydrological correction algorithm.

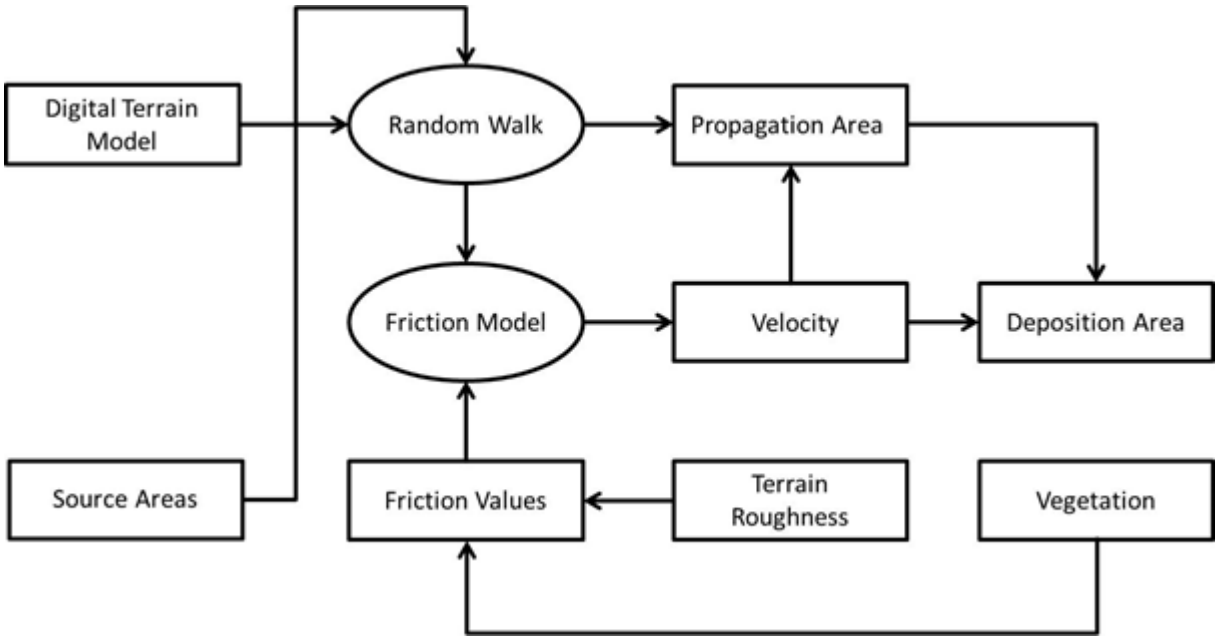

Figure 12: Workflow of the Rockfall Propagation Methodology

The following steps are applied to calculate updated rockfall propagation areas:

1. **Estimation of the runout distances by velocity calculation based on a one parameter friction model.** The impact of the falling rock on the slope surface and the associated energy reduction has significant influence on the runout distances. Based on velocity reduction, Broilli (1974) specifies the absorption of energy generated by an impact with 75-85 %. Three approaches to calculate energy reduction were tested based on the works of (a) Kirkby & Statham (1975), (b) Meißl (1998) and (c) Scheidegger (1975). The method chosen has an obvious effect on the propagation distance. Maximum distances were achieved with the algorithm of Kirkby & Statham, minimum distances with the algorithm of Meißl. Based on empirical model calibration, the best fitting propagation distance was achieved by using the energy reduction algorithm of Scheidegger. The motions after the impact are controlled by the contact with the slope surface – bouncing, rolling, and sliding. The grid is divided into triangles connecting the centroid point of each cell to integrate suitable equations which are generally based on the Coulomb´s law of friction in a grid based model. Thus, the distances between the preceding cell and the processing cell, are predetermined by the raster resolution. The velocity calculation is continued until the radicand of the equations gets negative. In this case the velocity is zero and the calculation stops.
2. **Estimation of process paths by using a multiple flow direction algorithm.** The positions of possible rockfall start cells in the detachment areas as defined from the disposition modelling are stored in an ancillary grid (source area file). Each neighbouring cell is coded in slope gradients, based on an eight direction flow model in hydrologic modelling. The assignment of adjacent cells as transition cells depends on: (a) slope threshold, (b) exponent of divergence, and (c) factor of persistence. These values have to be calibrated by an iterative approach taking into account real and documented events. The iterative calibration of model-parameters was done in several test areas with typical geomorphological and geological conditions applying two scenarios: (1) only one source area pixel was selected (single source area (SSA) - this is a theoretical assumption and is used mainly for illustration purposes) and (2) all source area pixels as defined by the disposition modelling (all source area (ASA)).
   By weighting the direction of the preceding cell, the system results in a first order Markov-chain. One must consider that a single modelled trajectory is based on random variables and

conditional probabilities (cf. Figure 13). Thus, implementation of random walk and Markov-chain in rockfall modelling is only applicable if a statistically sufficient number of random walks is performed so that the totalised potential process area is close to ergodic distribution of real events (Monte Carlo simulation).

Aiming at analysing the effect of iterations (10-10,000) for both model setups (SSA & ASA), the cumulative propagation area of 10 model-runs, with 100,000 iterations each was assumed as ergodic distribution. Low recognition rate mean values as well as high and fluctuating standard deviation (SD) values within the SSA-results indicate that the number of iterations is not sufficient to cover the high count of potential trajectories due to the high resolution DTM and relief conditions iteration dependency decreases with a high number of iterations Considering all source area pixels (ASA), the recognition rate curve ascends rapidly up to 1,000 iterations (> 95 %). Above this value, the curve tends towards an asymptote (cf. Figure 15).

3. **Introduction and calibration of friction coefficients for each forest class and for each defined block size.** According to Coulomb's law, the friction force is influenced by surface topography, (block)-mass and a dimensionless coefficient of friction. The effect of friction is controlled by surface roughness and block size. Including both, surface (roughness) and block characteristics (e.g. block size), the coefficient of friction can be used to simulate block size variations and distribution of rockfall material. The coefficients of friction are obtained by comparing simulated stops with data derived from field campaigns and remote sensing data (e.g. high resolution (0.2x0.2 m) orthophotos).

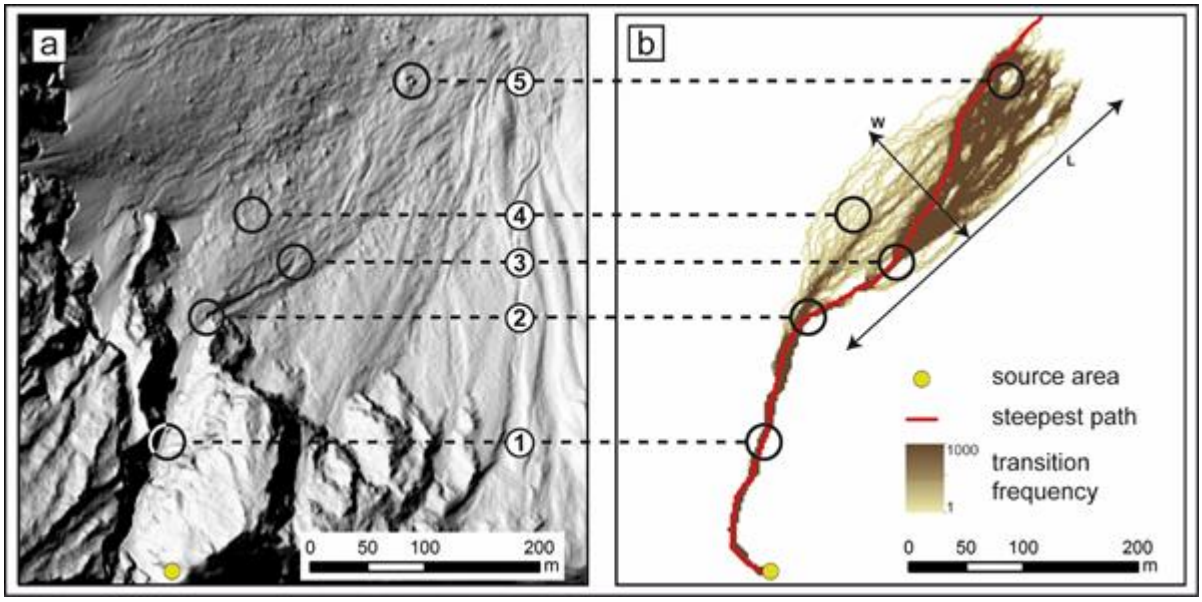

Figure 13: Test area "Ochsenkar": Transition frequencies of the modelling compared to the steepest path within the propagation area. For illustration purposes the modelling was performed with a single source area pixel (theoretical assumption) a: Hillshade visualisation based on ALS data with 1x1 m raster resolution; b: Rock HazardZone Model with 1,000 iterations.

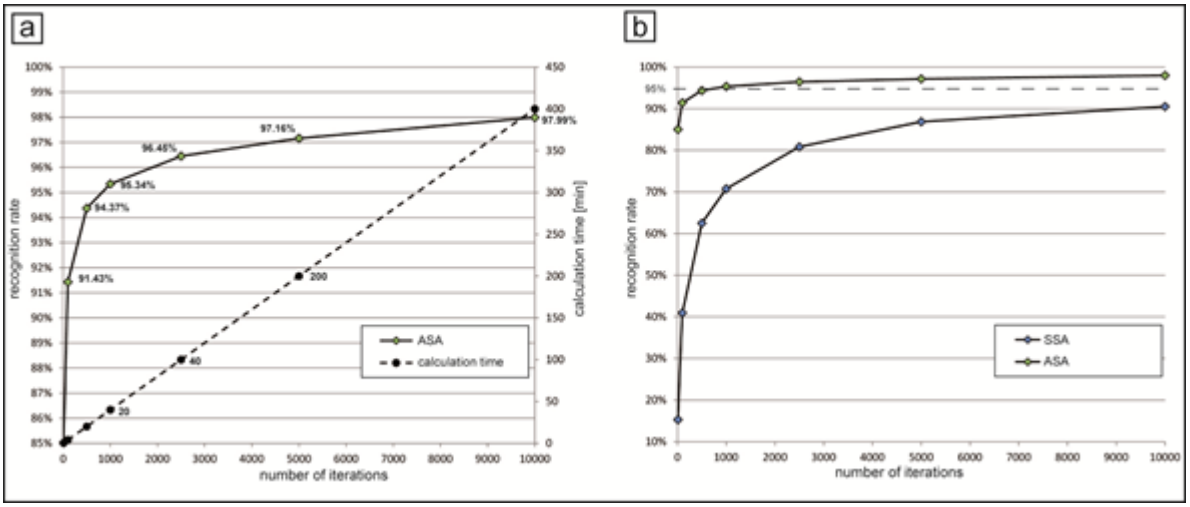

Figure 15: Effects of iterations. a: ASA-recognition rate and computer calculation time using an Intel(R) Xeon(R) CPU E5-1620v2@3.70 GHz and 64gb RAM. The calculation time demonstrates a strict linear correlation. b: Recognition rates of SSA and ASA model setups.

The propagation algorithms first were adapted for rockfall by Wichmann (2006) and compiled in the System for Automated Geoscientific Analyses (SAGA) (Conrad et al. 2015). SAGA is programmed in the object oriented C++ language and supports the implementation of new functions with a very effective Application Programming Interface (API). Functions are organised as modules in framework independent Module Libraries and can be accessed via SAGA's Graphical User Interface (GUI) or various scripting environments. SAGA generally is free software which may be redistributed and/or modified under the terms of the GNU General Public License as published by the Free Software Foundation (version 2 or later). However, the respective module Rock HazardZone was

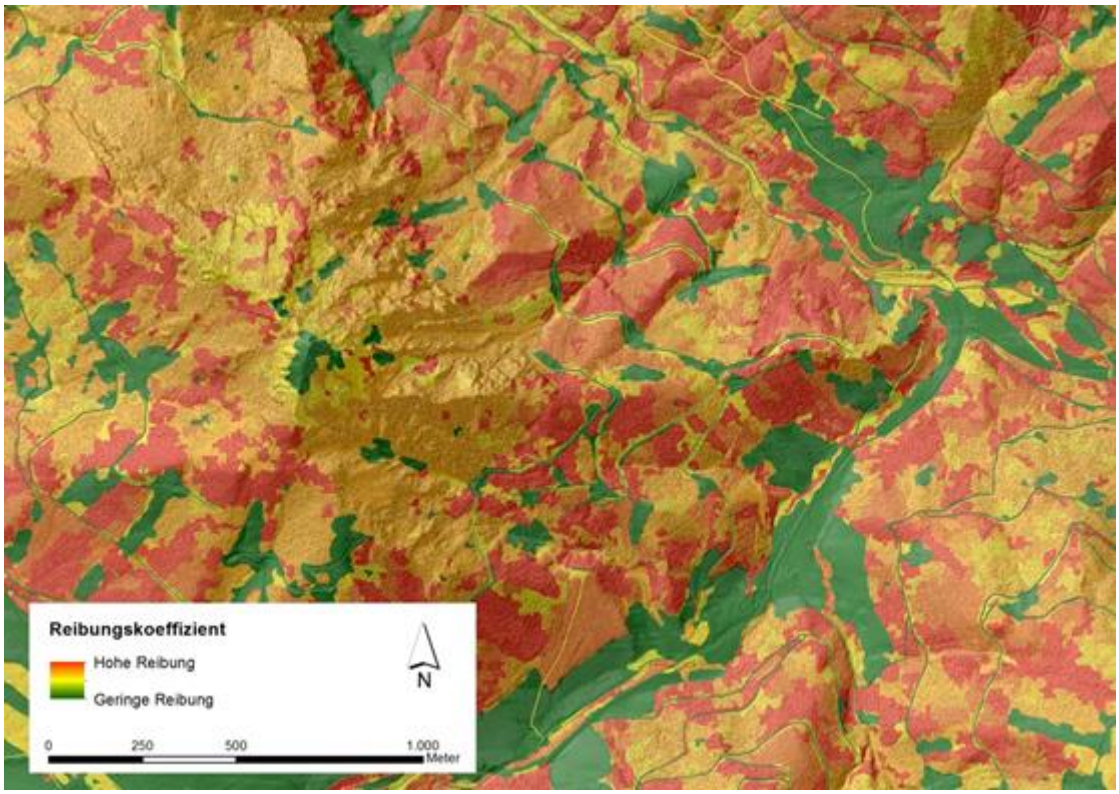

Figure 14: Assignment of Coefficients of friction according to relevant forest parameters in test area "Triebenstein" as derived from Airborne Laserscan data 2008 (green: low friction, red: high friction).

distributed in a purchasable version only In the SAGA 6.0.0 version (released 13/10/2017) an updated version of this module (called Gravitational Process Path Model) first was available open-source (Wichmann 2017). The current version is SAGA 6.2.0 (released 21/12/2017).

Within the frame of this project the algorithms will be integrated in the IMPACT software package of JOANNEUM RESEARCH.

The next steps will focus on integrating the updated forest parameters and the adaptation of friction parameters according to the identified changes. This will first be realised in areas affected by recent storm events. The area-wide and near-realtime application depends on the availability of the results of the Service “Forest Change Monitoring”.

5.5 Data and Service Prerequisite

5.5.1 Software Packages

At the Institute DIGITAL, Department of Remote Sensing and Geoinformation, at JOANNEUM RESEARCH, a long term expertise exists for the geometric processing of remote sensing image data. Moreover, outstanding expertise is being built up in the field of advanced image analysis. These fields of expertise have materialised in the development of

- The Remote Sensing Software Package Graz (RSG), which is designed for geometric processing options for almost any kind of remote sensing image data, acquired from airborne as well as space-borne platforms and comprising the geometric aspects of optical line scanners, SAR systems, perspective images and rational polynomial modelling. RSG covers nearly all of the geometric and radiometric processing options being specified for the VA processor components of the processor suite.
- The Image Processing and Classification Toolkit (IMPACT), which is designed for higher level image analyses capabilities and considers sophisticated methods and algorithms which ease the use and which enhance the applicability of remotely sensed imagery. IMPACT covers the majority of image processing, display and analysis necessities as specified for the VA processor components of the processor suite.

Both software toolboxes are subject to ongoing developments and algorithmic upgrades arising from the ongoing research at the Institute. The planned services will be implemented in the IMPACT environment, which is shortly described below:

At the Institute DIGITAL, Department of Remote Sensing and Geoinformation most of the processing tasks which involve higher level image analysis capabilities are performed within the IMPACT software framework. IMPACT has been designed from the beginning to be a framework which supports experts in their goal of develop new highly sophisticated methods and algorithms which ease the use and enhance the applicability of remotely sensed imagery.

One of the primary goals of IMPACT has been the use of an object oriented programming language (C++) and the utilisation of this language to reduce the development effort as much as possible. Hence the products and programs built within the IMPACT framework are highly modular and reusable. This modularisation of reusable components plays the key role when creating customised processing chains for our customers, since the development effort is reduced considerably.

The IMPACT framework consists of numerous components which are developed to cover specific tasks and simultaneously provide modularised parts for other research and development topics. The most relevant components (among others) are listed below:

- Generic large image handling facilities (multi-gigabyte files are no problem).
- Basic image manipulation filter library (Scaling, Arithmetic processing, …)

- Statistical analyses library (Regressions, Histogram matching, …)
- SAR image manipulation library (Speckle reduction, …)
- Library for object based analyses (Interest points, classification, …)
- Visualisation and statistical analyses of large images (IMPACTViewer).

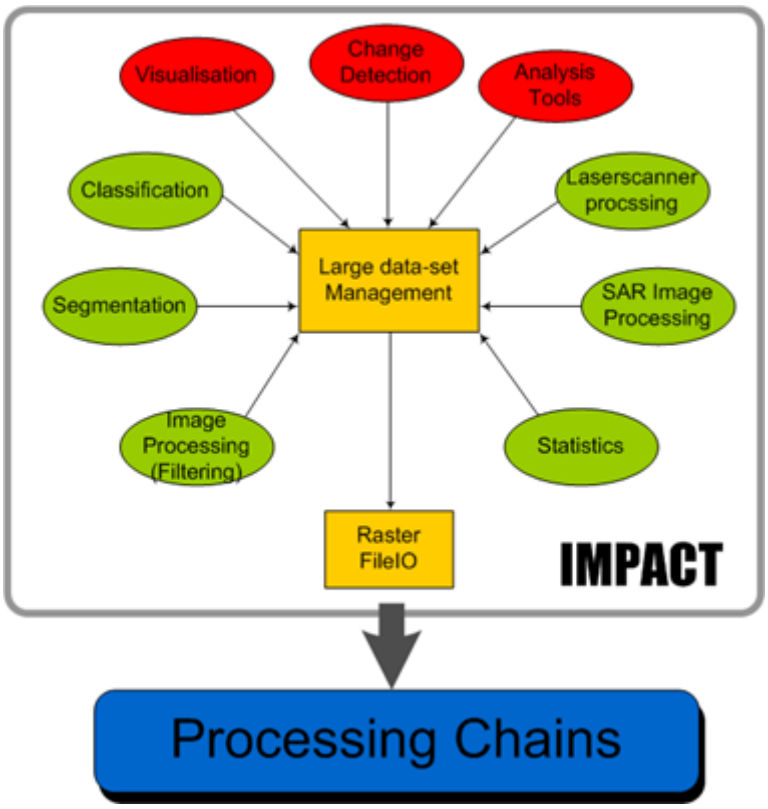

Figure 16: Technical structure of IMPACT.

Data sources

The LiDAR data (DTM, DSM) have been put at disposal by the Regional Government of Styria. The Sentinel-2 images are public available data, owned by the European Commission, and are distributed by the European Space Agency (ESA). Copernicus Sentinel-2 Multispectral Instrument (MSI) data are wide-swath, high-resolution, multi-spectral image data with the objective to support Copernicus Land Monitoring studies, including the monitoring of vegetation, soil and water cover, as well as observation of inland waterways and coastal areas. The MSI takes samples in 13 spectral bands, four bands at 10 meter, six bands at 20 meter and three bands at 60 meter spatial resolution. Sentinel-2 MSI Level-1C tiles are 100x100km2 ortho-images in UTM/WGS84 projection.

Wind hazard maps for the test area were foreseen to be included into the service “Storm Damage Resilience”. However, this data set was planned to be processed by ZAMG, but were not delivered and thus excluded from the modelling.

Data Set Used	real or “fake”[1]	owner	holder	volume	transfer rate	format	sensitive[2]

[1] Is there ‘fake-data’ in place or is all of the data real data?

[2] Is the data sensitive (privacy, commercial, etc…)?

DTM (Digital Terrain Model)	real	Federal State Steiermark	JR	300 GB		GEOTIFF	yes
DSM (Digital Surface Model)	real	Federal State Steiermark	JR	300 GB		GEOTIFF	yes
Geological Basemap	real	Federal State Steiermark	JR	40 MB		SHAPEFILE	no
Copernicus Sentinel-2 MSI data	real	European Commision / ESA	EODC	75 GB * 1.2 PB	36 GB/year * 1.7 TB/day	SENTINEL-SAFE format including JPEG2000 image data	no

* Values indicating the absolute amount of data

5.6 Development Plan

The development of the “Storm Damage Resilience” is completed. The “Change Detection” module will be developed within the next period of the project and will complete the service “Forest Change Monitoring”. The next steps on the “Rockfall Propagation” service will focus on integrating the updated forest parameters and the adaptation of friction parameters according to the identified changes. This will first be realised in areas affected by recent storm events. The area-wide and near-realtime application depends on the availability of the results of the Service “Forest Change Monitoring”.

6 Heat Map Service

6.1 Description of the Service

One of the core responsibilities of taxi fleet management is to maximise utilisation of the taxi fleet. This fleet utilisation is strongly linked with the ability to predict demand for taxis in the immediate future. Correctly assessing demand minimises waiting times between passengers for drivers, thus increasing utilisation. The proposed heatmap service supports taxi fleet management by generating such predictions.

The heatmap service will gather information on the current state of a specified area by consuming a variety of data services available via DMA such as services that provide the current distribution of people, current weather and forecasts, scheduled events and public transport arrival times, amongst others. It will then temporally and spatially unify these data sources to provide a standardised basis for further processing. Using pre-trained prediction networks and specifically built algorithms, the taxi demand is then predicted from the current state of the observed area. For the prediction the area is regularly subdivided into a grid and the demand is predicted for every square of the grid.

The basis for the prediction algorithm is an analysis of the standard distribution of taxi demand from the historical data of partnered taxi fleet managers (Taxi 40100) to provide us with a baseline demand. For every input type we build models for how they affect this baseline demand. For example, for the hour after a concert ends we will expect increased taxi demand at the venue of the concert. Another example: If a regular long-distance train should arrive at 22:00 but is delayed to 22:30. The taxi demand generated by people arriving at the station with this train will occur 30 minutes later than usual. Also a hypothesis that we will follow is that the demand for taxis will increase for such an event, as people are trying to compensate for the train delay be choosing a faster inner-city means of transportation.

In the context of DMA Work Package 8, the heatmap service is one of the planned demonstrator services of Task 8.3. The purpose of these demonstrator services is twofold. On the one hand they serve as a study on how DMA can support complex data-driven applications and drastically decrease their development time through a ecosystem of standardised data-provides. On the other hand, they will provide example implementations that show how to consume DMA services on a technical level.

Other data products and services the Heatmap Services depends on are:

- Taxi rides
    - During training
    - Used to validate hypothesis
    - Provided by Taxi 40100
- People distribution
    - During training & prediction
    - Provided by T-Mobile
- Weather conditions
    - During training & prediction
    - Provided by ZAMG
- Events
    - During training & prediction
    - Obtained from public sources
- Inter-city transportation
    - During training & prediction
    - Obtained from public sources

6.2 Current Implementation

There exists a proof-of-concept (POC) implementation of the service. At the time of writing the DMA platform is not yet available in a form that allows us to consume data from DMA services. Thus, the POC uses local data sources to validate our methods for the prediction process.

This implementation already includes models for how irregular events and delays in public transport affect the baseline demand as described above in the description of the service. The following image provides a map visualisation of the derived demand changes generated by our POC service. Models that predict the effect of weather on taxi demand are currently in development. The necessary data to generate the baseline taxi demand was not yet made available to us; we use simulated data in the current implementation.

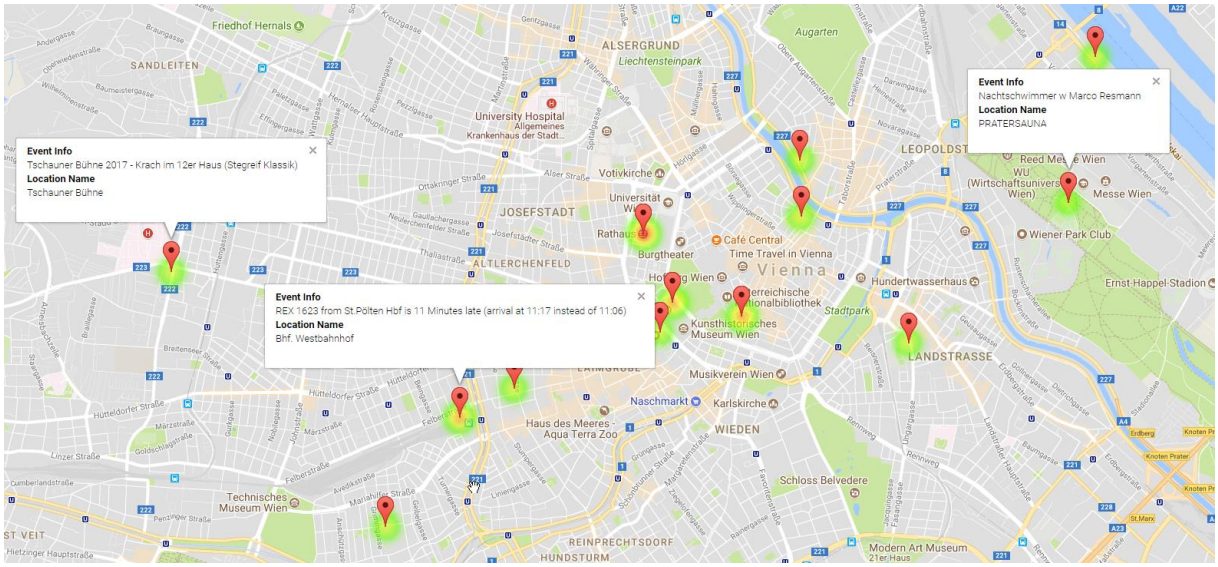

Figure 17

6.3 Interfaces and APIs

This chaper describes the planned first version of the API for the heatmap service. This specification is derived from the current proof-of-concept implementation and subject to change pending further developments. All requests and responses are supmitted via HTTP.

Request	
Method	**URL**
`GET`	`api/v1.0/taxidemand/`
Parameters/Request Body	
Type	**Params**
`HEAD`	`auth_key`
`auth_key` The `auth_key` that was given in response to /api/login	

QUERY	lon_e
lon_e [number] The longitude of the eastern border of the target region	
QUERY	lat_n
lat_n [number] The latitude of the northern border of the target region	
QUERY	lon_w
lon_w [number] The longitude of the western border of the target region	
QUERY	lat_s
lat_s [number] The latitude of the southern border of the target region	
QUERY	res
res [integer] The resolution of the prediction grid. The grid consists of squares with sides of length res meters.	
QUERY	predict_for
predict_for The time coordinate for which the prediction shall be made. If no timezone information is given, UTC is assumed, Date and Time in ISO 8601 format	
Response	
Status	**Response**
200	The grid of predicted demand in GeoJSON format. Each region in the grid is given as a GeoJSON feature: ● The geometry is defined as a GeoJSON point that specifies the center of the region. ● The demand property contains the predicted demand of taxis. The demand is given as the number of needed taxis in the 60 minutes. ● The baseline property contains the baseline demand calculated for this region. Again as the number of taxis needed in the next 60 minutes.
400	Error message detailing the issue with the request.
403	Empty Response

6.4 Development plan for the remaining project time

Further development will proceed on the basis of the POC service. We plan to reuse significant parts of it for the final heatmap service. The plan for turning our POC implementation into a DMA service with the functionality described above has two major parts:

6.4.1 Migrate POC to DMA platform

Ultimately, the heatmap service will be part of the DMA platform as a DMA compatible service. Realising this requires the following migrations from our local POC service:

- Switch from local data sources to consuming DMA data services. As this demonstrator will be one of the first services to do this on the DMA platform, this will likely involve the development of general components that facilitate service orchestration.
- Implement the full HTTP interface as defined above. Extended documentation and clearly defined behaviour will be necessary in a service ecosystem such as DMA.
- We intend to host the service on our (Catalysts) in-house infrastructure. Achieving full integration with the DMA platform this way will require additional efforts on the infrastructure level. However, we do expect that compliance with these requirements will involve development effort for the service.

6.4.2 Real world baseline demand and prediction models

Calculating the baseline taxi demand, as described above, is still unsolved. Historical data for taxi demand will be available via the DMA platform and we plan to develop the baseline demand system based on this data once we can access it.

While some of the models that predict a data sources' effect on taxi demand are already developed, others are still missing. In particular, models for predicting the effect of weather (temperature, precipitation and wind speed) on the demand for taxis are currently in development. The second important prediction model that still needs to be developed is the effect of the distribution of people. We expect both of these factors to have significant impact on demand.

7 Taxi Ride Sharing

7.1 Description of the Service

The time and distance optimisation of transports, particularly transports of persons when looking at taxis, is a significant contribution to the efficiency and quality improvement of the overall data oriented services model. The overall objectives of taxi ride sharing is to combine several individual trips in one trip in an efficient way. Since the short-term changes have to be taken into account at all times, adherence to time constraints is essential for the underlying algorithms. In order to meet these requirements, the planned transport optimisation system must, on one hand, be based on intelligent, goal-oriented and practicable as well as very fast optimisation algorithms (e.g. scheduling should be done in 1-2 minutes), and, on the other hand, on an efficient state-of-the-art software architecture for integration this scheduling service is required.

The underlying model of the service can be described around four main entities.

Rides: The time and distance optimisation of a transport request for a customer consists of the starting point, the destination point and a desired pickup time. Furthermore, a maximum deviation time should be considered as well. As a result, time windows can be defined for the pick-up at the start node or at the destination point. Due to the definition of the time windows, a premature arrival at the destination is not possible, while a delayed arrival at the destination is possible. The time windows are strict. However, only deviations from the critical destinations (e.g. airport) are very critical, deviations for non-critical destination points are not punished in the same way.

Vehicles: The vehicles are placed on one or more depots. The vehicles of a fleet are equipped differently (heterogeneous fleet). Each vehicle has a certain capacity for each type of transport (e.g. person or material)

Constraints: An important constraint of scheduling dynamic transport orders is that not all information is known at the beginning of the planning process. The number of transport orders and the traffic condition, for example, are constantly changing in such a service.

Computing: The shared ride algorithm works in such a way that it reschedules the available transport resources (trips, vehicles ...). The calculations that lead to this allocation are carried out at regular intervals, so that all new and also short-term intervals can be processed. All available transport resources within a defined planning horizon are considered. The transport resources must then be informed of their next scheduled and/or re-scheduled orders via their mobile devices.

7.2 Current implementation

The scheduling algorithm assigns the taxi rides to the available transport vehicle. This assignment is carried out at a regular interval. In this way, all new and also new entered short-term transport rides are taken into account. In each of these optimisation runs, all available vehicles are considered. The vehicle must be informed via their mobile devices about their next scheduled rides.

7.2.1 Objectives

In the scheduling algorithm various options are calculated, which ride or which rides could be combined with each other. These allocation options are then evaluated and the optimal combinations are selected. The combination function considers various criteria of the requirements of drivers and the management platform.

The following criteria can in principle be included in the evaluation:

- Arrival time required by the vehicle to get from the current location to the starting point of the new ride
- Travel time for the whole ride including loading and unloading times for each customer
- Delay calculate the time-out for the pickup place or delivery place for each ride
- Earliness calculate the earlier arrival time for the pickup place or delivery place for each ride
- Savings - this criterion only applies to those transport resources that can handle more than one ride at the same time. The calculation includes the time savings that would be obtained in comparison to the sequential processing of orders.
- Optimal is the options with the best value of the valuation function.

7.2.2 Combination strategies

Firstly, we consider empty vehicles and all possible allocation of these vehicles by considering the capacity and the compatibility. In order to limit the number of variants to be considered, only those combinations of two rides are considered whose pickup and delivery destination are sufficiently close to each other. The evaluation functions consider the following evaluation criteria's:

1. Arrival time to the pickup location
2. Travel time to fulfill rides
3. Savings compared to the sequential processing of rides

Secondly, in the case of occupied vehicles, these vehicles are tested for available capacity (see next paragraph) and, depending on the result of the capacity check for a particular vehicle, decided whether to add an additional ride to the planned route. The evaluation functions consider the following evaluation criteria:

1. Calculate additional time spent by combining two rides for each customer
2. Number of common pickup and delivery locations

Thirdly, during this step the capacity and compatibility is verified. We consider allocation options (e.g. certain combinations of rides and vehicles). It checks, for example, whether there is still enough space in the vehicle. The capacity check is carried out in a forward-looking way, for example the already planned pickup and delivery of persons are considered

7.2.3 Scheduling time

In one assignment cycle, all rides can be assigned which are entered in the system and have the status "open" and whose planned start time is within a certain time window. The size of this time window can be individually adapted to the circumstances of the management platform. Explanation: The scheduled start time of an order is either entered during order entry ("Start time is relevant") or, if the end time (= arrival time at the destination) is relevant. Average ride duration is calculated from the delivery time and an estimated starting time.

7.2.4 Strategies for rides allocation

At the heart of the scheduling algorithm is the search for the best possible combinations of several rides, the evaluation of these combinations, and the creation of an allocation plan (that is, which vehicle gets which order next). Depending on the rides status (e.g. the relationship between the rides to be awarded and the available vehicle) following strategies are considered:

- Award the best possible combinations of rides that gets rides as cheaply as possible for the customer
- Distribute the shared rides equally to the vehicles

These two optimisation steps can be done either individually (each one by itself) or in combination.

7.3 Data and Service Prerequisite

The services need several prerequisite to execute the models as described in the model description above. Furthermore, a distance map should be available. In the first iteration, we used a static version. In the next iteration, we will use a dynamic version which be build up the map on runtime.

7.4 Development Plan

First, there exists a proof-of-concept (POC) implementation of the ride-sharing service. The proof-of-concept version uses local data sources to validate our methods for the scheduling process.

In the next development, cycle the service will be tested with data from the Taxi 40100 which will be provided via the DMA portal. In this next step, the scalability of the service will be evaluated too.

Finally, we carry out evaluation of the services. First, the result should be close to the optimum and the probability of obtaining a poor result should be low. Second, a critical point is the required computing time. This criterion is objective and depends heavily on the available data and the required quality of the solution. Third, the understanding and the simple comprehensibility of the results contribute to the acceptance and thus to a wider circle of users. This also helps in the evaluation of the results achieved.

8 Recommendation and Search

8.1 Overview and Current Developments

Recommender Services, or recommenders for short, are commonly used today. Especially online shopping platforms make use of recommenders to provide their customers with suggestions about offers in their stores. Generally speaking, recommenders assist humans to find what they are looking for in large collections of products, service offers, or documents. The Know-Center develops and maintains its own recommender framework called ScaR, which is short for Scalable Recommendation-as-a-service[1]. The ScaR framework uses Apache Solr[2] as its storage and search engine. Currently, the source code of the ScaR Framework is hosted at a public Git repository operated by the Know-Center[3]. The source code is open to the general public under the GNU Affero General Public License. The ScaR framework is a customisable basis for a specialised recommender in any domain.

For offering the ScaR framework as a service on the DMA platform, the versatility of the ScaR framework has been increased. Now it is possible to customise the ScaR recommendation service without the need of programming but just by means of configuration. This is the first step to make the ScaR framework available as an DMA service. This adaption required extensive testing also undertaken in this year.

8.2 Architecture

The ScaR framework follows a microservice architecture, meaning that the ScaR recommender is split into several smaller services, each with dedicated and clearly defined responsibilities. Figure 18 shows the overall architecture of the Scar Framework. It consists of five microservices. The microservices communicate with each other, and with other systems, via the HTTP. The HTTP communication of the microservices is facilitated by a Jetty[4] web server running in each microservice in embedded mode.

The microservice architecture is designed to scale horizontally. Each of the five microservices can be replicated and spread their workload over multiple instances. If it is necessary, each instance can be deployed on a seperate machine to cope with the demand. Apache Zookeeper[5] is used to monitor and control all instances.

Each microservice is explained in further detail in the following subsections of this section.

---

[1] http://scar.know-center.tugraz.at/

[2] https://lucene.apache.org/solr/

[3] https://git.know-center.tugraz.at/docs/?r=scar-framework.git

[4] https://www.eclipse.org/jetty/

[5] https://zookeeper.apache.org/

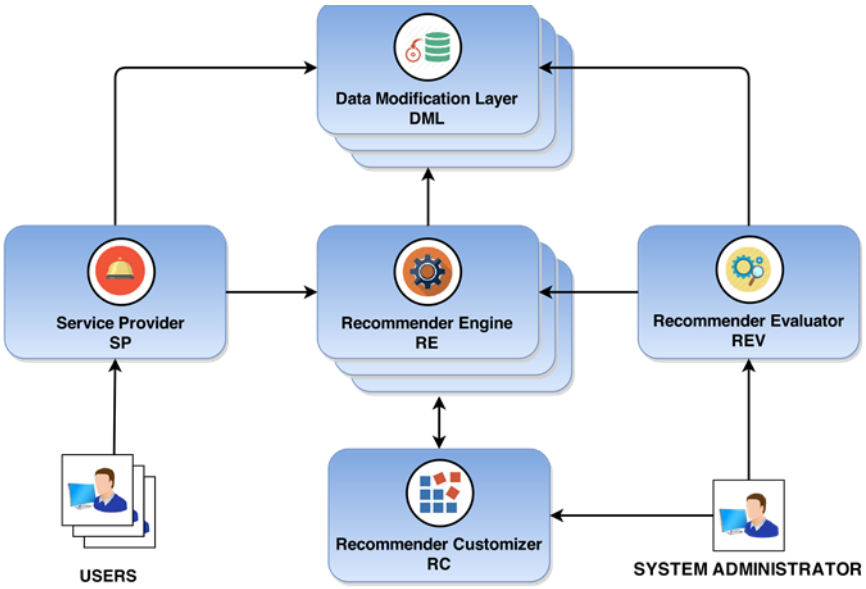

Figure 18: Microservice system architecture of the ScaR framework. The microservices use HTTP to communicate with each other. The architecture scales horizontally; all microservices can be instantiated multiple times and run on different machines if required.

8.2.1 Data Modification Layer (DML)

The complete data used to generate the recommendations is stored in Apache Solr. The Data Modification Layer (DML) communicates with the Apache Solr storage backend. The data handled by the DML and stored in Apache Solr can be of various types. Simple textual data (e.g., textual description of items), user interactions and transactions (e.g., rating given by user, items bought by a user), and localisation data (e.g., geo-locations encoded in latitude and longitude coordinates) are all accepted data types by the DML. The aggregation of the submitted data forms the data corpus used to calculate the recommendations. Depending of the application scenario, not all different data types might be used, even though it would be possible. The DML also interfaces the search functionally of Apache Solr. This allows to use Apache Solr queries to generate post-filtered and personalised recommendations.

8.2.2 Recommendation Engine (RE)

The Recommendation Engine (RE) can be considered the core of the ScaR framework as it generates the recommendations based on different recommendation algorithms. The EL supports user-based Collaborative Filtering (CF), Content-Based (CB) filtering, Most Popular (MP), and hybrid approaches combining two or more of the other recommendation strategies to generate recommendations. The direct communication of the EL with the DML enables the ScaR framework to incorporate changes in the data corpus quickly into the generated recommendations. Hence, there is no need for computational expensive pre-calculations to consolidate the recommendation data corpus on updates.

8.2.3 Recommender Customiser (RC)

The recommendations generation is configured via so called recommender profiles. A profile defines the share of the data corpus, the algorithms, and the algorithm parameters which will be used to generate the recommendations. The Recommender Customiser (RC) manages all the recommender profiles and informs the RE of changes in existing or new recommender profiles.

8.2.4 Recommender Evaluator (REV)

The ScaR framework supports online and offline evaluations to judge the quality of generated recommendations. The Recommender Evaluator (REV) executes the chosen evaluations. It supports A/B testing for online evaluation, meaning that one group of users gets recommendation configured by a particular recommender profile and the second groupe gets recommendations configured by another recommender profile. Offline evaluations can be carried out by splitting the data into training and test sets and by measuring the algorithmic performance with commonly used metrics (e.g., F1-score, nDCG, Coverage, Diversity, Serendipity, and Runtime).

8.2.5 Service Provider (SP)

The Service Provider (SP) is the gateway of the ScaR framework to the other systems. It accepts the requests and delegates them to the respective component. Hence, the service provider is also responsible for load balancing, if multiple instances of a microservice exist.

8.3 Interface Description

This section describes the endpoints of the ScaR recommender Framework to insert data, update data, generate recommendations, search the data corpus, and evaluate the recommendations. All calls are accessible via HTTP version 1.1. Data send to the recommender or received from it is encoded in JSON.

8.3.1 Service Provider (SP)

name	path	methods	description
Recommender Engine Repository Resource	`/bulkData`	`POST`	REST resource for frontend service provider that takes care of all the incoming requests and hadles/redirects them
	`/delete`	`DELETE`	
	`/position`	`GET`	
	`/positionNetwork`	`PUT`	
	`/profile`	`PUT`	
	`/recommendation`	`PUT`	
	`/resource`	`GET`	
	`/review`	`PUT`	
	`/sharedLocation`	`PUT`	
	`/socialInteraction`	`PUT`	
	`/socialStream`	`PUT`	
	`/taskInfo`	`PUT`	

	/updatePost	GET
	/userAction	POST
	/search/searchCore	PUT
		POST

8.3.2 Recommendion Engine (RE)

name	path	methods	description
Recommender Engine Resource	/createProfile	POST	Resource for creating recommendations and wrapping recommender engine calls
	/deleteProfile	GET	
	/getAvailableAlgorithms	GET	
	/recommendation	GET	
	/updateProfile	POST	

8.3.3 Evaluation

name	path	methods	description
Dashboard Information Resource	/information /purchaseDistribution	GET	A service resource for getting the statistical information for the dashboard
	/information /recomendationActionRates	GET	
	/information /recomendationDistribution	GET	
	/information /recomendationTimePeak	GET	
	/information /topCategories	GET	
	/information /topZones	GET	
	/information /zoneDistribution	GET	
Multivariate Testing Resource	/multivariate /addTest	PUT	A service to evaluate mutlivariate tests
	/multivariate /changeTest	PUT	

	/multivariate /info	GET	
	/multivariate /removeTest	PUT	
	/multivariate /statistics	GET	
Recommender Evaluation Resource	/evaluate	PUT	Services for triggering a evaluation and getting the status
	/status	GET	

8.3.4 Recommender Engine Repository

name	path	methods	description
Recommender Engine Repository Resource	/createProfile	POST	Resource for maintaining recommender profiles
	/deleteProfile	GET	
	/getAllProfileIds	GET	
	/getAllProfiles	GET	
	/getProfile	GET	
	/updateProfile	POST	

8.3.5 Data Modification Layer (DML)

Detailed description of available calls of the Data Modification Layer are in the Swagger documentation of the service.

name	description
AbTestingModificationResource	Resource for AB testing.
EvaluationsModificationResource	Resource to store evaluation recommendations
FeedbackLoopModificationResource	Resource to store feedback to generated recommendations
LocationEventsModificationResource	CURD services for Location Event Data
LocationItemModificationResource	CURD services for Location Items Data
PositionNetworkModificationResource	CURD services for Position Network Data
PositionsModificationResource	CURD services for Position Data
ProfileModificationResource	CURD services for Profile Data
ResourcesModificationResource	CURD services for Resource Data
ReviewsModificationResource	CURD services for Reviews Data
SharedLocationsModificationResource	CURD services for Shared Locations Data
SocialActionModificationResource	CURD services for Social Action Data

SocialStreamModificationResource	CURD services for Social Stream Data
SolrDMLSearchResource	Wrapper for Solr Search functionalities
UserActionModificationResource	CURD services for User Action Data

8.4 Development Plan

In the project year past, the focus was on increasing the versatility and configurability of the ScaR framework. This functionality was extensively tested. In the upcoming project year, the ScaR framework will be integrated into the container-based virtualisation environment outlined by DMA for services in the DMA marketplace. This will allow the ScaR framework to be run in the DMA infrastructure. The container-based ScaR framework will then be offered as one of the DMA service consumable via the online portal.

9 Conclusion and Outlook

This deliverable summarises the current state of the work in WP6, written at the end of the first half of the project. The work on all services mentioned in this deliverable is currently ongoing. For each service there exist already a prototype. This means the progress of the work is according to the project plan. The deliverable describes the status of each service in detail. The services are organised according to their purpose. The two services which are part of the DMA core service, namely the service ingestion and the semantic enrichment and entity linking are important tools for developers publishing their service in DMA. The service ingestion helps in managing the lifecycle from the initial publishing of the service, over updates and changes, to the final retirement of the service from DMA. The semantic enrichment and entity linking processed and enriched the metadata of each service to make it easy to find for potential customers. The remaining services described in this deliverable are part of the initial services populating DMA. Many of these services are also used in the pilots developed by WP8 and WP9. Furthermore, these services should motivate DMA participants to either use them in their software products or to also contribute services in to DMA.

In the remaining second half of the project duration, WP6 will cater for further improvement of the services described in this deliverable. Functionality not implemented yet will be implemented and tested; already implemented functionality will be tested and further improved where it is needed. A major part of the efforts in the second half of the project duration will be the integration of the services into the DMA infrastructure. If it is not already done, all services have to be provided as a Docker container to fully comply with DMA service regulations.

10 References

Bell, R., Glade, T., Granica, K., Heiss, G., Leopold, P., Petschko, H., Pomaroli, G., Proske, H. & Schweigl, J. (2013): Landslide Susceptibility Maps for Spatial Planning in Lower Austria. In: Margottini, C., Canuti, P. & Sassa, K. (Eds.) Landslide Science and Practice. Springer, Berlin, Heidelberg: 467-472.

Broilli, L. (1974): Ein Felssturz im Großversuch. – Rock Mechanics. 3: 69-78.

Chung, C-J.F. & Fabbri, A. (2006): Systematic Procedures of landslide hazard mapping fur risk assessment using spatial prediction models. In: Glade, T., Anderson, M. & Crozier, M. (Eds.), Landslide Hazard and Risk. Wiley, Chichester: 139-174.

Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V. & Böhner, J. (2015): System for Automated Geoscientific Analyses (SAGA) v. 2.1.4. – Geosci. Model Dev., 8: 1991-2007, doi:10.5194/gmd-8-1991-2015.

Decker, B. L. (1986). World geodetic system 1984. Defense Mapping Agency Aerospace Center St Louis Afs Mo.

Dekang Lin. (1998). An Information-Theoretic Definition of Similarity. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98), Jude W. Shavlik (Ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 296-304

Dvořák L, Bachmann P, Mandallaz D (2001): Sturmschäden in ungleichförmigen Beständen. In: Schweiz. Z. Forstwes. 152 (2001) 11: 445–452

Gardiner B, Blennow K, Carnus J.M. , Fleischner P, Ingemarson F, Landmann G, Lindner M, & Marzano M, Nicoll B, Orazio C, Peyron J-L, Reviron M.P., Schelhaas M, Schuck A, Spielmann M, Usbeck T (2013). Destructive storms in European Forests: Past and Forthcoming Impacts. Final report to the European Commission - DG Environment, available online: http://ec.europa.eu/environment/forests/pdf/ STORMS %20Final_Report.pdf

Gardiner B, Schuck A, Schelhaas MJ, Orazio C, Blennow K, Nicoll B (eds., 2013a) Living with Storm Damage to Forests. European Forest Institute, Joensu

Gebhardt H, Glaser R, Radtke U, Reuber P (eds., 2011) Geographie; Physische Geographie und Humangeographie. Spektrum Akademischer Verl., Heidelberg

Guzzetti, F., Reichenbach, P. & Wieczorek, G.F. (2003): Rockfall hazard and risk assessment in the Yosemite Valley, California, USA. – Nat. Hazards Earth Syst. Sci. 3: 491-503.

Hale SE, Gardiner B, Peace A, Nicoll B, Taylor P, Pizzirani S (2015): Comparison and validation of three versions of a forest wind risk model. Environmental Modelling & Software 68:27–41.

Hanewinkel M, Kuhn T, Bugmann H, Lanz A, Brang P (2014): Vulnerability of uneven-aged forests to storm damage. Forestry 87:525–534.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). The elements of statistical learning (Vol. 1, pp. 337-387). New York: Springer series in statistics.

Indermühle M, Raetz P, Volz R. (2005): LOTHAR Ursächliche Zusammenhänge und Risikoentwicklung. Synthese des Teilprogramms 6. Umwelt-Materialien Nr. 184. Bundesamt für Umwelt, Wald und Landschaft, Bern.

Kirkby, M.J. & Statham, I. (1975): Surface stone movement and scree formation. – Journal of Geology. 83: 349-362.

Mayer H (1999) Waldbau; Auf soziologisch-ökologischer Grundlage. Springer, [s.l.]

Mayer H, Schindler D, Kunz M, Ruck B (eds.,2010): Strategien zur Reduzierung des Sturmschadensrisikos für Wälder (Verbundprojekt RESTER). Berichte des Meteorologischen Instituts der Albert-Ludwigs-Universität Freiburg Nr. 21.

Meißl, G. (1998): Modellierung der Reichweite von Felsstürzen. Fallbeispiele zur GIS-gestützten Gefahrenbeurteilung aus dem Bayrischen und Tiroler Alpenraum. – Innsbrucker Geografische Studien. 28: 249 pp.

Pasztor F, Matulla C, Zuvela-Aloise M, Rammer W, Lexer MJ (2015): Developing predictive models of wind damage in Austrian forests. Annals of Forest Science 72:289–301.

Philip Resnik (1995). Chris S. Mellish, ed. "Using information content to evaluate semantic similarity in a taxonomy". Proceedings of the 14th international joint conference on Artificial intelligence (IJCAI'95). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1: 448–453.

Russ W (2011): Mehr Wald in Österreich. In: BFW-Praxisinformation 24: 3-5.

Scheidegger, A.E. (1975): Physical aspects of natural catastrophes. Elsevier, Amsterdam / New York, 289 pp.

Schindler D, Grebhan K, Albrecht A, Schönborn J, Kohnle U (2012): GIS-based estimation of the winter storm damage probability in forests; A case study from Baden-Wuerttemberg (Southwest Germany). International journal of biometeorology 56:57–69.

Schmoeckel J (2005): Orographischer Einfluss auf die Strömung abgeleitet aus Sturmschäden im Schwarzwald während des Orkans "Lothar“. Dissertation. Universität Karlsruhe. Online verfügbar unter https://www.imk-tro.kit.edu/download/diss_schmoeckel.pdf

Sebauer V (2013): Querschnittsmaterie Wald. Europäische Forstpolitik. In: Zuschnitt Zeitschrift über Holz als Werkstoff und Werke in Holz 51: Im Wald. 13- 16.

Suvanto S, Henttonen HM, Nöjd P, Mäkinen H (2016): Forest susceptibility to storm damage is affected by similar factors regardless of storm type; Comparison of thunder storms and autumn extra-tropical cyclones in Finland. Forest Ecology and Management 381:17–28.

Wichmann, V. (2006): Modellierung geomorphologischer Prozesse in einem alpinen Einzugsgebiet. Abgrenzung und Klassifizierung der Wirkungsräume von Sturzprozessen und Muren in einem GIS. – Eichstätter Geographische Arbeiten 15: 231 pp.

Wichmann, V. (2017): The Gravitational Process Path (GPP) model (v1.0) – a GIS-based simulation framework for gravitational processes.- Geosci. Model Dev., 10, 3309-3327, 2017. https://doi.org/10.5194/gmd-10-3309-2017.