SMARTCHAIN

SMART SOLUTIONS IN SHORT FOOD SUPPLY CHAINS

D6.10 Implementation of the inventory of innovations and related interactive tools (ii)

Work Package 6

IPB

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 773785

Document Identification

Project Acronym	SMARTCHAIN
Project Full Title	Towards Innovation - driven and smart solutions in short food supply chains
Project ID	773785
Starting Date	01.09.2018
Duration	36 months
H2020 Call ID & Topic	SFS-34-2017 - Innovative agri-food chains: unlocking the potential for competitiveness and sustainability
Project Website	http://www.smartchain-h2020.eu/
Project Coordinator	University of Hohenheim (UHOH)
Work Package No. & Title	WP6 Innovation platform
Work Package Leader	ISEKI-Food Association (IFA)
Deliverable No. & Title	D6.4 Implementation of the inventory of innovations and related interactive tools (ii)
Responsible Partner	Institute of Physics Belgrade (IPB)
Author (s)	Dušan Vudragović, Antun Balaž (IPB)
Review & Edit	Katherine Flynn, Foteini Chrysanthopoulou, Gunter Greil (IFA); Javier Casado (UHOH)
Type	Report
Dissemination Level	PU – Public
Date	16.08.2021
Version	1.0 Dušan Vudragović (IPB)
	1.1 Dušan Vudragović (IPB), Javier Casado (UHOH)
Status	Final

Executive Summary

The focus of this deliverable, as defined in the project's Description of Action [1], is to present the implementation of the inventory of innovations and related interactive tools. Our implementation followed the functional requirements and the initial design of the system presented in the deliverable D6.2 - Design of the inventory of innovations and related interactive tools [2]. We have shown the development stage of the system and its components in the middle of the project lifetime in the deliverable D6.4 - Implementation of the inventory of innovations, related interactive tools [3]. In this deliverable, we present the final architecture of the innovation inventory and related interactive tools. The content of the previous deliverable is updated here to reflect the current status of the system and related tools, as this is the second version of the document.

In this document we have outlined the main technical details and described the implementation of components that are structured into three layers: the front-end, the back-end, and the underlying infrastructure. For each component, we have reported its function and implementation details. The significant number of components are based on existing widely used open-source solutions, such as Elasticsearch [4] [5], Apache Tika [6][7], LevelDB [8], and Casbi [9]. For the project purposes, these are made use of by developing a set of related interactive tools: document store API, REST API, innovation inventory portal, and innovation description template. Together, all these components produce a workflow that realizes the SMARTCHAIN inventory of innovations.

This workflow is exposed for the front-end components through the REST API to consume the service. One of the front-end components is the innovation inventory portal that allows querying of the stored innovations. The portal is incorporated into the SMARTCHAIN innovation platform. Contrary to the innovation inventory portal, whose interface is designed to support a wide range of communities outside the project, the document store API is mainly oriented towards the project participants. Therefore, the innovation platform sends read-only requests to the innovation inventory, while the document store API also supports write requests, i.e., project's hub managers and WP leaders can store new and edit existing information within the inventory through the document store API and REST API.

After a brief introduction in Section 1, Section 2 of this deliverable gives updated details of the system's architecture, describes functions of components, and lists open-source solutions that are used for the implementation of the inventory of innovations. Section 3 gives technical details on related interactive tools, components that are developed within the framework of the project and whose main purpose is to orchestrate processes and to enable interaction between different components in the system. Section 4 presents the deliverable conclusions. Appendix A and Appendix B specify the underlying infrastructure and REST API more technically. Appendix C contains the innovation description template, a minimal set of metadata we have used for the description of the innovation, and finally, Appendix D lists innovations registered within the system.

Table of Contents

Glossary

ACL	Access Control Lists
API	Application program interface
CRUD	Create, Read, Update and Delete
DB	Database
DSL	Domain Specific Language
HTTP	HyperText Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDT	Innovation description template
IO	Input/output
IP	Intellectual property
JSON	JavaScript object notation
PDF	Portable document format
REST	Representational state transfer
SFSC	Short food supply chains
WP	Work package

1. Introduction

The development of the innovation inventory and related interactive tools started at the beginning of the project by identifying community requirements. We collected initial requirements directly from stakeholders during the multi-actor workshops and extended these with consortium-specific demands and suggestions during the project lifetime. Finally, facing these requirements with currently available technology solutions, we developed the SMARTCHAIN inventory of innovations.

The primary purpose of the SMARTCHAIN inventory of innovations is to enable knowledge transfer, innovation, and cooperation between the involved stakeholders of the studied short food supply chains. Due to the increasingly varied nature and practice of short supply chains, dependencies on different geographic conditions (culture, climate, resources, governing structures, available infrastructure, market, etc.), the consortium primarily focused on 18 preselected case studies, existing short food supply chains, from 9 countries (2 case studies per country). During the analysis of these case studies, the project identified innovative and practical solutions relevant to the short food supply chain scale up. These were documented for WP6 purposes using an innovation description template and exposed through the SMARTCHAIN inventory of innovations.

The innovation inventory is implemented as a document organization and retrieval system, which supports quick finding and discovery of information related to short food supply chains. Through it, the users can upload, share, and discover innovations, patents, IPs, and other materials related to food supply chains. The target user group are farmers and agricultural organizations looking to optimize their operations, as well as innovation donors, i.e., researchers, technology providers, etc., who wish to raise the visibility of their innovations within a highly interested audience. The front-end of the inventory is an interactive online portal (https://www.smartchain-platform.eu/en/innovation-inventory) oriented towards all the stakeholders and actors and incorporated into the SMARTCHAIN innovation platform. The front-end allows storing, generating, sharing, and utilizing information on innovations, facilitating communication between the innovation hubs. The back-end is the inventory (database) of available innovations, solutions, and recommendations.

2. Architecture of the inventory

During the multi-actor workshops coordinated by the WP1 team, hub managers identified and collected in total 109 feedbacks about expectations and 68 feedbacks on user needs via a predefined questionnaire and in discussion with participants. We presented the summary of these results in the deliverable D6.2 - Design of the inventory of innovations and related interactive tools [2]. Participants' expectations are grouped into 20 main categories, which allows us to identify the three most important ones: analysis of the current situation, knowledge of actors and stakeholders about the short food supply chains, and feedback and sharing of experiences. These expectations highlighted the necessity of an accurate analysis of the state-of-the-art and a need for the knowledge transfer and sharing of experiences.

We followed the feedback about expectations collected within the WP1 during the development of the inventory architecture. Also, the architecture, data structure, and content of the inventory were validated by end-users during the project's lifetime. Since the information about innovations are mostly text-based data, such as text documents (e.g., scientific papers, patents, case studies, product descriptions, farm descriptions, and technology descriptions), scanned paper documents, presentations, and spreadsheets, our goal was to store these data formats, and made them fully searchable via the innovation platform. In order to enable an efficient search for large quantities of unstructured data, the system needs to build a data structure called index. The index holds searchable information extracted from documents and is organized to support a full-text search on every available piece of information. The process of creating such a data structure is known as indexing. It goes through all data records, extracts a list of terms that appear in them, and makes a note of which records hold it for each term. Such an approach enables quick lookup of records that hold the terms of a search query.

However, before the indexing can take place, the document analysis step must extract all the data that will be searchable. The extraction transforms raw information into a suitable format (usually a semi-structured document). This processing step is heavily dependent on the type of raw data being ingested. For example, the extraction will differ significantly in the case of HTML pages and scanned documents. While the former is directly parse-able, the latter must go through an image processing step and optical character recognition (OCR) to extract textual content.

The search processing consists of two components. First, the search query that the user has entered in natural language or via an optional formal syntax is translated into the actual set of terms and constraints that will be looked for in the index. This process can include word stemming, lemmatization, removal of stop words (such as "the"), and other language-specific transformations to make the query more flexible for matching in the index. The second component of the search is assigning the rank of relevance to each result retrieved. This can be controlled by tweaking the weights of fields in the index records. The relevance is also calculated using the predetermined metrics, such as TF-IDF (term frequency-inverse document frequency).

There were several high-quality open source solutions for the implementation of the search facility in the form of libraries, services, and engines, and we performed a detailed analysis of each of them. In general, we find out that libraries provide the greatest flexibility to developers, giving them just the parts needed to implement the entire system from scratch. On the other hand, services make libraries more encapsulated and easier to use and scale, while still giving enough flexibility to design the rest of the system that uses the search service. The engines are complete solutions for document management, whose flexibility is limited by their established design and implementation. They usually support limited customization through configuration or extensions of some specific components. In particular, we tested Apache Lucene and Xapian libraries, Elasticsearch, Solr and Sphinx services, and Ambar and OpenSemanticSearch engines. Details about these are given in the deliverable D6.2 - Design of the inventory of innovations and related interactive tools.

After analysis of functional requirements and available technologies, we started with the inventory development, which through several iterations led to the architecture illustrated in

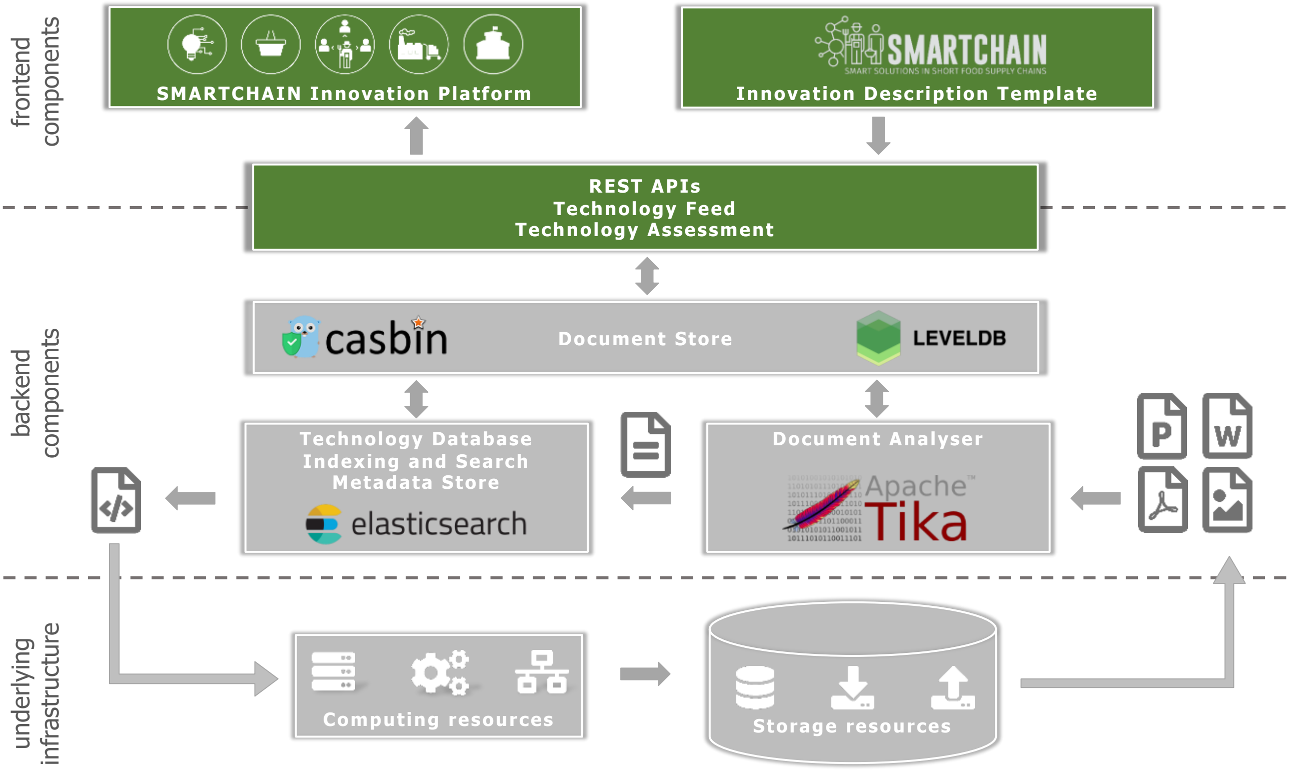


Figure *1*. The architecture is structured into three main layers: the underlying infrastructure, the back-end, and the front-end (related interactive tools).

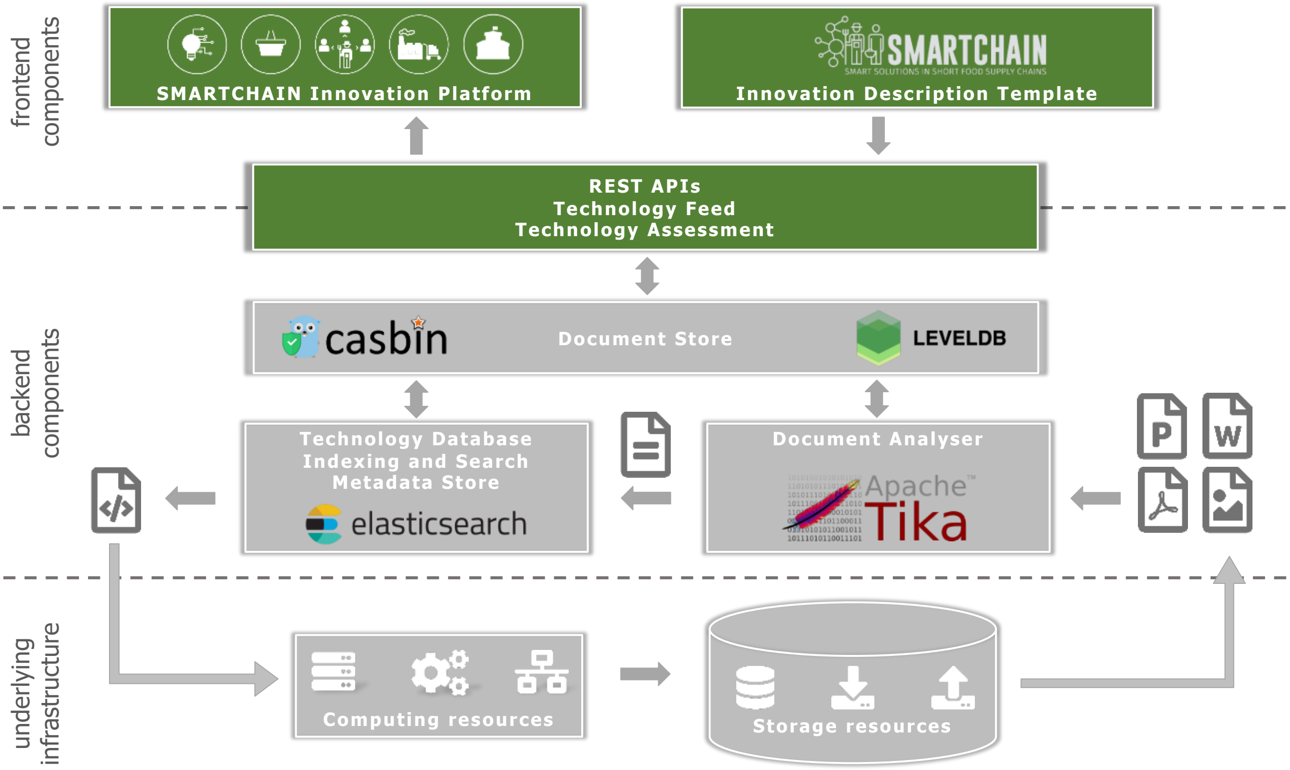


Figure 1: Architecture of the SMARTCHAIN inventory.

The underlying infrastructure layer is based on the computing and storage capacity of the PARADOX-IV cluster at the Institute of Physics Belgrade. It consists of 106 compute nodes and a storage system of 100 TB storage space connected with 10 Gbps Internet connection. Technical details of the cluster are given in Appendix A.

The central component of the backend layer is the SMARTCHAIN document store. It brings together all backend services and databases, in particular, technology database, indexing and search engine, metadata store, and document analyzer. In this layer, information is organized into JSON document structures, which can be extended by an arbitrary number of additional fields. The system is developed to support such an extension, ensuring that new fields are searchable either via free-form queries, which do the full-text search, or structured queries, which can give more specific match criteria. The documents can also have file attachments, which can be image data, PDFs, Word documents, spreadsheets, document scans, etc. All these attachments are processed in the background by the document analyzer to extract any searchable text content from them. Each attachment is associated with a field in the document structure, where the extracted text and file metadata are stored. The attachment files themselves are stored in an integrated LevelDB database [8] that resides in the underlying infrastructure layer. In order to expose the attachments through the REST API, a corresponding unique key is assigned per attachment within the JSON document.

LevelDB was chosen as a file storage because of its simple interface and ability to be fully integrated into the document store service binary. This reduces the number of components inside the system. In essence, it is a fast key-value store that can use any binary string as either key or data. This allows us to store the attachment metadata and files without additional encoding into other formats (such as base64), which would possibly increase the size of the data. In addition to this, a direct usage of the file system would impose the need for an additional metadata store, so LevelDB proved as a more consistent approach for this.

Besides the document management, the SMARTCHAIN document search also has a support for management of system users. This is an administrative feature that is used to control the level of access to the service. Users can be created, updated, deactivated, and their access can be controlled on a REST API path basis.

Elasticsearch technology [4] [5] is used for the creation and maintenance of a search index that holds all the document structures that are put into the SMARTCHAIN Document Store. It also holds text contents from the attached files, which are extracted by the document analyzer component. Such an index enables full-text search on any part of the document and returns matching documents ranked by how well they correspond to the search query. More specific queries, that can give more strict control on how matches are made, can be specified in the Elasticsearch query DSL syntax. In this syntax, a query is a JSON object, which has a number of specified fields that control matching, filtering, and paging of the results.

Since the search engine within the SMARTCHAIN Document Store can only work with text data, we use the document analyzer component to extract text information from all the files that are attached to the documents in the store. It is based on the Apache Tika library [6] [7], which can extract text content from a wide range of file formats. According to the documentation, it is very versatile as it supports over one thousand file types. The extracted text is stored in the index on corresponding documents and is included in full-text searches. The Tika service is also used to determine the mime type of attachments at upload time.

All components for interaction with the inventory: document store API, REST API, innovation platform, and innovation description template, we developed as related interactive tools, which are described in the following section.

3. Related interactive tools

In this section, we describe interactive tools developed within the framework of the project: document store API, REST API, innovation inventory portal, and innovation description template. These are all components that users could use for uploading, sharing, and discovering innovations, patents, IPs, and other related materials related to food supply chains.

3.1 Document store API

The document store is a backend service that integrates all components needed to support the SMARTCHAIN innovation platform. It is implemented in the Go programming language [10] and has the following components:

- Embedded web server that publishes the REST API;
- Component handling the search via Elasticsearch;
- Text extraction component that uses Apache Tika;
- File storage component based on LevelDB.

The entire system is compiled and statically links into a single binary, but it has a dependency on Elasticsearch and Tika services, which also depend on the Java platform. The Go programming language was chosen as an implementation platform because it enables easier asynchronous programming, which was used to orchestrate all the background work that gets executed by the system in response to API requests. Another helpful feature of it was that it can make static binaries that do not require any dependencies to be installed on the deployment target.

The HTTP server from the Go standard library's NET/HTTP package was chosen for hosting the REST API. It has asynchronous processing of HTTP requests, which enables it to scale to a much larger number of connections than it would if it instantiated a thread or process per request. It is configured to support HTTPS protocol if the SSL certificate is available. The location of the server host certificate is given via the—`cert` argument and the private key is passed through the—`key` argument. The details of the hosted API are given in Section 3.2 and Appendix B.

The authorization is implemented using the Casbin library [9], which supports the PERM (Policy, Effect, Request, Matchers) metamodel for specifying authorization schemes. In our implementation, we use the authorization model based on access control lists (ACL) on API paths, that define the access level for each user role. It also has an admin superuser who can access all API calls. Anonymous users are allowed GET access on paths relevant to serving content in a read-only fashion. All requests are authenticated via the HTTP basic authentication.

The search engine is built around Elasticsearch which is a service wrapper around the Lucene library [12], that handles index operations, scaling, and fault tolerance. The interface it exposes is reminiscent of a database, where tables correspond to separate indices and document fields to table columns. The index used by the document store is named `sc-innovations`, and it is configured to use the edge n-gram tokenizer in its default analyzer for all fields. This setting allows searching and matching of incomplete phrases in order to give meaningful results, even if there are typos in the search query.

Since the search in the SMARTCHAIN document store can only work with text data, we use Apache Tika to extract text information from all files that are attached to the documents in the store. Tika is a toolkit for text extraction, which can read text from more than one thousand different file types. Similar to Elasticsearch, it runs as a standalone server in the background and the document store invokes its services through an asynchronous queue.

This queue is implemented using a Go programming language channel on which each newly uploaded file is wrapped and sent in a job structure that holds its data, the document ID, and the property in the document to be updated. The job structures are consumed by worker goroutines, which are similar to threads, but are asynchronous, and in IO-bound tasks, many of which can run concurrently per single thread. The processing in these workers invokes Apache Tika to extract any text information from the given file and to determine its mime type. The text contents and original file names are stored into the document in the index, on the specified property, which makes the information contained in the file searchable. The mime type is added as metadata into the file storage, where the original file data is also stored. This process is shown in a sequence diagram in **Figure 2**. The text extraction component depends on the Tika standalone server jar being present, and, if not, the appropriate version will be downloaded automatically and started.

The files of the attachments added to documents in the index are stored in a LevelDB storage [8]. LevelDB was chosen as a fast key-value store, which can be included in the binary of the Document Store as a static library, not to impose any new dependencies for the deployment. Along with the data from each file, we also store additional metadata including the mime type and the original file name. These metadata are used later, when serving a file to properly set response headers, so the client can render them correctly. Physically, all data are stored in the storage component of the underlying infrastructure.

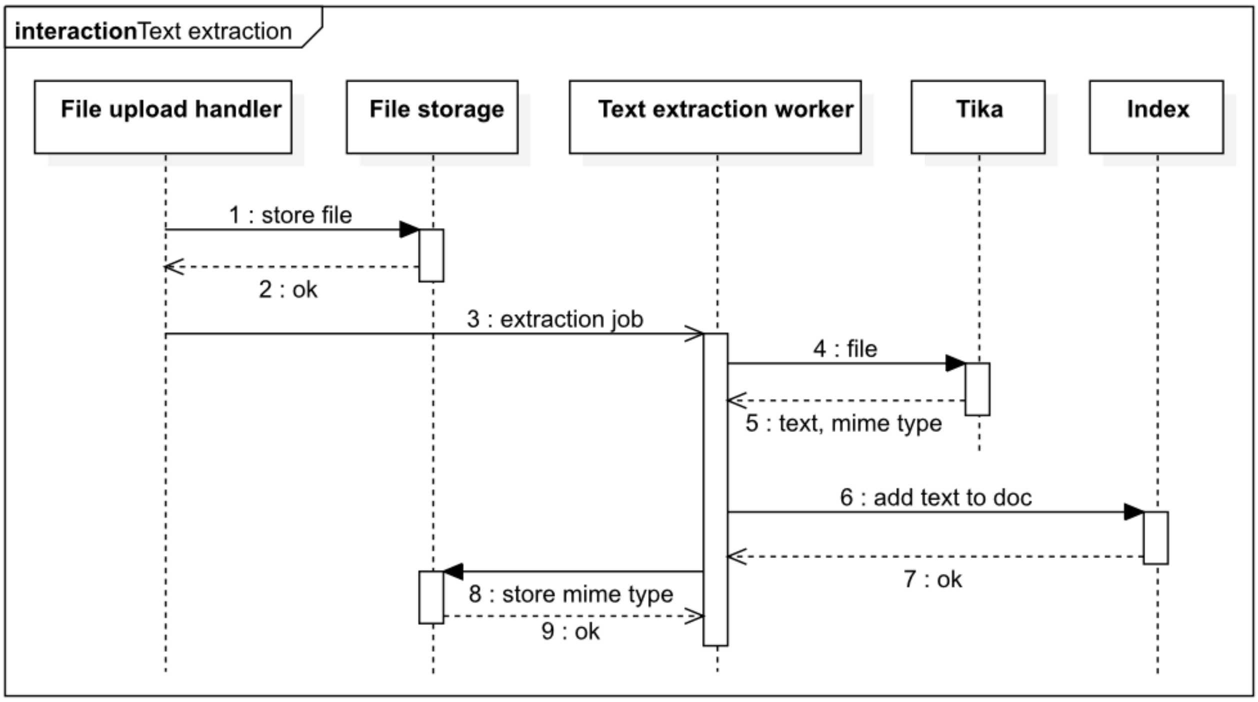


Figure 2: A sequence diagram of the file upload and text extraction process.

3.2 REST API

The REST API provides a unified interface to the document store backend. It supports standard create-read-update-delete operations on documents stored in the system. The REST API is available on the HTTPS protocol and, depending on the configuration, could be optionally protected by the basic HTTP authentication scheme. The API exposes four types of resources: Documents, Attachments, Search, and User.

The central entity of the information schema in the document store is a **Document**. It is a JSON object with arbitrary properties, except for the ones which begin with an underscore (_), as they are subject to additional processing in the system. Currently, reserved properties are `_pictures` and `_documents`, which are intended for the picture and file attachments. After processing, pictures and documents properties are populated with arrays containing the corresponding relative URLs. Also, the original fields are populated with objects that contain file metadata and extracted text contents for full-text search. Documents are identified by `_id` field, which is assigned at document creation in the search index. The API endpoint for documents is on `/api/doc` path, and it supports CRUD (Create, Read, Update and Delete) operations on the documents using the standard HTTP verbs, as prescribed by REST:

- POST creates a document;
- GET fetches the document;
- PUT updates an existing document;
- DELETE removes a document.

A more detailed description of each operation, along with example requests and responses, is given in Appendix B.1.

The **Attachments** are files that are associated with an innovation. They are tied to a specific document on one of its properties. The property they attach to is specified during the upload, and it has structure of an array, in order to support upload of multiple files at once. Subsequent uploads to the same property are appended to the array. The attachments have two separate endpoints:

- `/api/upload` - this path accepts POST request with multipart/form-data encoding that contains the target document ID, the target property on which to place the attachments, and the attachment files;
- `/api/attachment/{key}` - this path responds to GET requests and returns the file of the attachment by the given key. The key is generated from the document ID, property, and index in the array of attachments.

When fetching an attachment, it will be served with an appropriate mime/type and original file name in response headers. Technical specification and example of the attachments' resources are given in Appendix B.2.

The underlying search engine in the document store accepts search queries specified as JSON objects. The **Search** endpoint is on `/api/search` path and it accepts POST requests through JSON query in the request body. The format of these queries is specified in Elasticsearch Query DSL, and it supports many options to control the result matching, filtering, paging, etc. More details on common queries and a request example can be found in Appendix B.3. Search results are served in an abbreviated form in a JSON array. Each element contains the following fields:

- `docid` - document ID;
- `title` - innovation title;
- `summary` - innovation description, shortened;
- `pictures` - the array of URLs of the attached pictures from the `_pictures` property.

The results are sorted by relevance score, and the main intended use for these data are to be rendered on the search results page and to provide enough information to link to the full document behind the match.

Administrative operations on the document store service include system **User** administration. These are the users who can access the REST API, and their access is controlled by ACL's rules on API paths. The authorization rules are configured in the `auth_model.conf` and `policy.csv` files, which are outside the scope of the REST API. A user entity contains the following fields:

- `username` - unique username;
- `password` - only filled out on user creation or update, otherwise blank;
- `role` - role name that controls the authorization of the user;

- `mail` - mail address;
- `active` - boolean that specifies if the user is enabled.

The requests available for the users' resource are:

- POST on `/api/users` creates a new user. The password in the request is expected to be in clear text and it will be stored hashed on the backend. It will never be sent in a response to any other request, it is only used for authenticating requests.
- GET on `/api/users` fetches the list of all system users, with password fields left blank.
- GET on `/api/users/{username}` fetches a specific user, also without a password.
- PUT on `api/users` updates a given user to the field values specified in a JSON object in the request body. Updates are total, if any field is left blank or omitted, it will be cleared in the database as well.

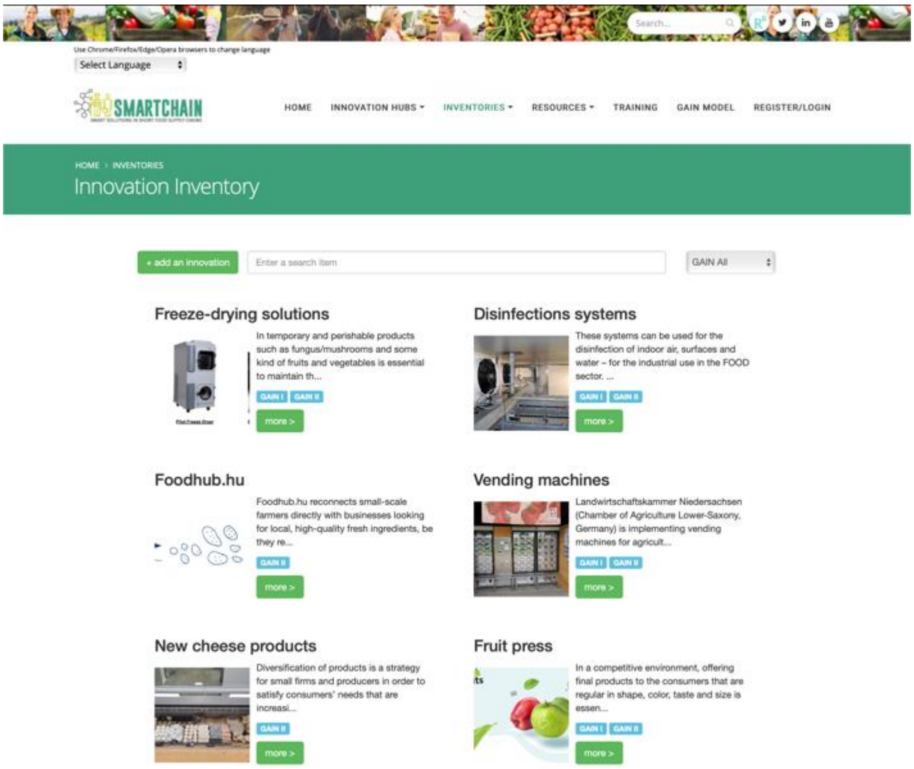
More details and example requests and responses are given in Appendix B.4.

3.3 Innovation inventory portal

The innovation inventory portal (https://www.smartchain-platform.eu/en/innovation-inventory) is a web front-end that enables read/write access to the Document Store for the SMARTCHAIN hub managers and WP leaders. It is implemented in C++ using the Wt web toolkit [13]. The application provides a view to search and show results from the document store, and a detailed view for specific documents. The innovation inventory portal search page is illustrated in **Figure 3**.

For each innovation registered in the system, the user can get an innovation page with further information about the innovation. For some innovations, this is provided in the form of a web page (**Figure 4**), while for others as PDF document. However, the following information is associated with each innovation in the inventory:

- title of the innovation;
- picture that visually describes the innovation;
- type of innovation, whether it is technological, social, environmental, etc.;
- end users who might be interested in the innovation;
- short summary for practitioners;
- website for additional information;
- countries of origin;
- documents and publications;
- technology readiness level;
- GAIN model level;
- people involved, who are able to provide additional information and support.



Freeze-drying solutions

In temporary and perishable products such as fungus/mushrooms and some kind of fruits and vegetables is essential to maintain th...

Disinfections systems

These systems can be used for the disinfection of indoor air, surfaces and water – for the industrial use in the FOOD sector. ...

Foodhub.hu

Foodhub.hu reconnects small-scale farmers directly with businesses looking for local, high-quality fresh ingredients, be they re...

Vending machines

Landwirtschaftskammer Niedersachsen (Chamber of Agriculture Lower-Saxony, Germany) is implementing vending machines for agricult...

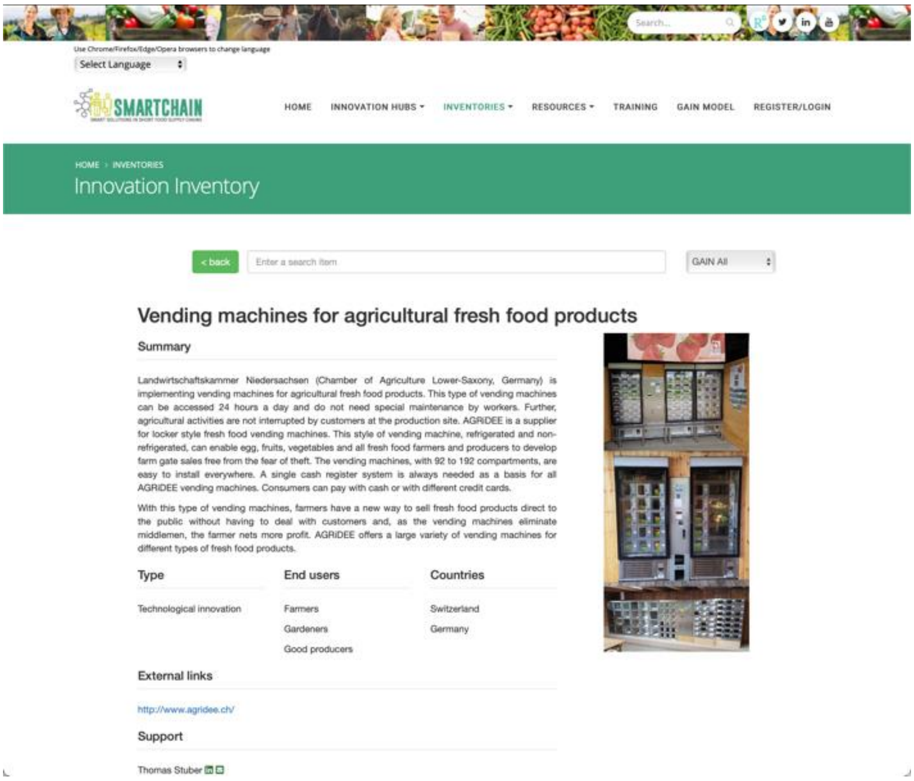
New cheese products

Diversification of products is a strategy for small firms and producers in order to satisfy consumers' needs that are increasi...

Fruit press

In a competitive environment, offering final products to the consumers that are regular in shape, color, taste and size is essen...

Figure 3: The innovation portal incorporated into SMARTCHAIN innovation platform.



Vending machines for agricultural fresh food products

Summary

Landwirtschaftskammer Niedersachsen (Chamber of Agriculture Lower-Saxony, Germany) is implementing vending machines for agricultural fresh food products. This type of vending machines can be accessed 24 hours a day and do not need special maintenance by workers. Further, agricultural activities are not interrupted by customers at the production site. AGRiDEE is a supplier for locker style fresh food vending machines. This style of vending machine, refrigerated and non-refrigerated, can enable egg, fruits, vegetables and all fresh food farmers and producers to develop farm gate sales free from the fear of theft. The vending machines, with 92 to 192 compartments, are easy to install everywhere. A single cash register system is always needed as a basis for all AGRiDEE vending machines. Consumers can pay with cash or with different credit cards.

With this type of vending machines, farmers have a new way to sell fresh food products direct to the public without having to deal with customers and, as the vending machines eliminate middlemen, the farmer nets more profit. AGRiDEE offers a large variety of vending machines for different types of fresh food products.

Type	End users	Countries
Technological innovation	Farmers Gardeners Good producers	Switzerland Germany

External links

http://www.agridee.ch/

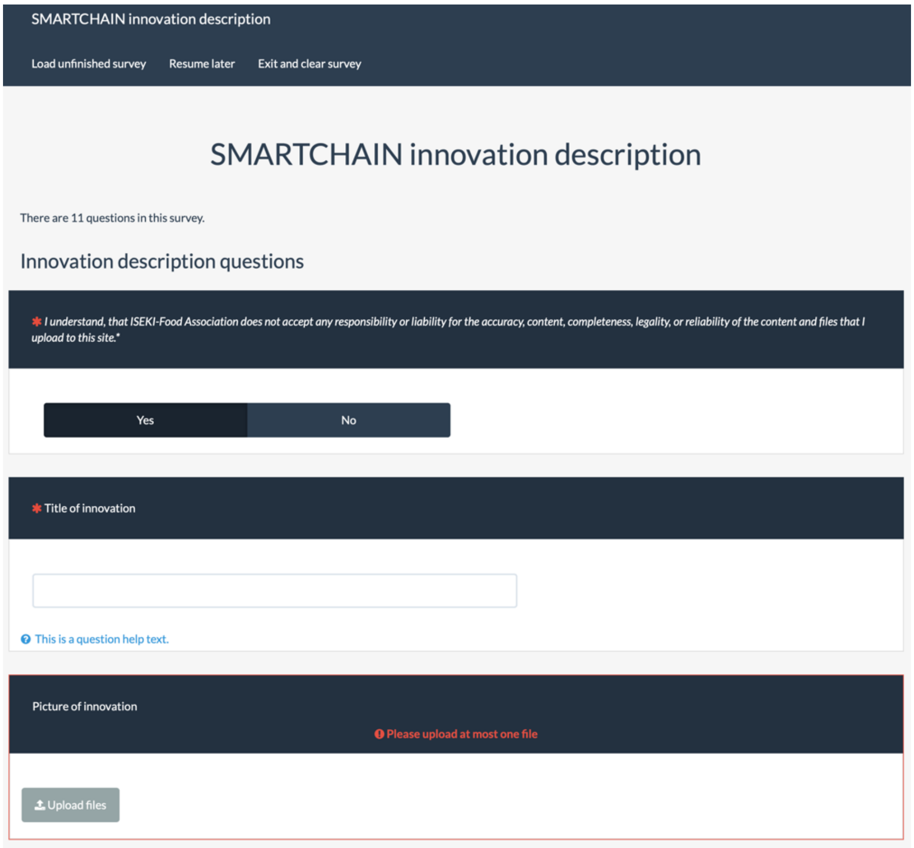
Support

Thomas Stuber

Figure 4: The innovation portal document page.

3.4 Innovation description template

The main content in the innovation inventory system is uploaded by innovation donors, project's hub managers, and WP leaders through the innovation inventory portal. However, innovations could be registered by stakeholders outside the consortium as well. This is done using the Innovation Description Template (IDT). The template was initially developed as an offline document, and today, it is the online web form implemented in the form of a survey (**Figure *5***). Besides innovation descriptions, IDT supports additional data entry, such as geographical location, technology readiness level, potential customers, patent information, related documentation, photos, videos, etc. The current version of the IDT is given in Appendix C. All collected data are used to better gauge the relevance of the innovation to various search queries, which are submitted via the innovation inventory portal.


SMARTCHAIN innovation description

SMARTCHAIN innovation description

There are 11 questions in this survey.

Innovation description questions

* *I understand, that ISEKI-Food Association does not accept any responsibility or liability for the accuracy, content, completeness, legality, or reliability of the content and files that I upload to this site.**

* Title of innovation

This is a question help text.

Picture of innovation

Please upload at most one file

Figure 5: Innovation description online form.

4. Conclusions

As it is described in the SMARTCHAIN DoA, the project aims to foster and accelerate the shift towards collaborative short food supply chains and to introduce new robust business models and innovative practical solutions that enhance the competitiveness and sustainability of the European agri-food system. This is realized by the analysis of the technological and non-technological (WP2), social (WP3), consumer (WP4), environmental (WP5), and business and policy (WP7) specific factors related to short food supply chains, which resulted in identification of the key parameters that influence sustainable food production and rural development.

The WP6 supported these activities by developing a virtual environment for knowledge transfer, innovation, and cooperation for all the stakeholders of the short food supply chain. From the beginning of the project, we have expected a lot of unstructured information to be stored within the inventory system. Therefore, to enable an efficient search for large quantities of unstructured data, we have created the index that holds searchable information extracted from documents or manually uploaded to the system. It is organized to support full-text search on every available piece of information. We have identified available open-source solutions that could be reused for project purposes and developed a set of related tools that orchestrate the workflow in a seamless manner.

In this document, we have reported technical details of the front-end and back-end components. The central part of the system is the document store API that integrates all components within the SMARTCHAIN inventory of innovations, and the REST API that allows frontend components to consume the service. In addition to these, we have documented the innovation inventory portal, which utilizes the provided APIs. Also, we presented the innovation description template and its online implementation that allows the registration of innovations. Following the innovation description template structure, we have registered 153 innovations listed in Appending D.

A. PARADOX-IV cluster

PARADOX-IV cluster represents the fourth major upgrade of the PARADOX cluster (illustrated in the figure below) and became operational during September 2013. The cluster consists of 106 working nodes and 3 service nodes. Working nodes (HP ProLiant SL250s Gen8, 2U height) are configured with two Intel Xeon E5-2670 8-core Sandy Bridge processors, at a frequency of 2.6 GHz and 32 GB of RAM (2 GB per CPU-core). The total number of new processor-cores in the cluster is 1696. Each working node contains an additional GP-GPU card (NVIDIA Tesla M2090) with 6 GB of RAM. With a total of 106 NVIDIA Tesla M2090 graphics cards, PARADOX is a premier computer resource in the wider region, which provides access to a large production GPU cluster and new technology. The peak computing power of PARADOX is 105 TFlops. One service node (HP DL380p Gen8), equipped with an uplink of 10 Gbps, is dedicated to cluster management and user access (gateway machine). All cluster nodes are interconnected via Infiniband QDR technology, through a non-blocking 144-port Mellanox QDR Infiniband switch. The communication speed of all nodes is 40 Gbps in both directions, which is a qualitative step forward over the previous (Gigabit Ethernet) PARADOX installation. The administration of the cluster is enabled by an independent network connection through the iLO (Integrated Lights-Out) interface integrated on motherboards of all nodes.

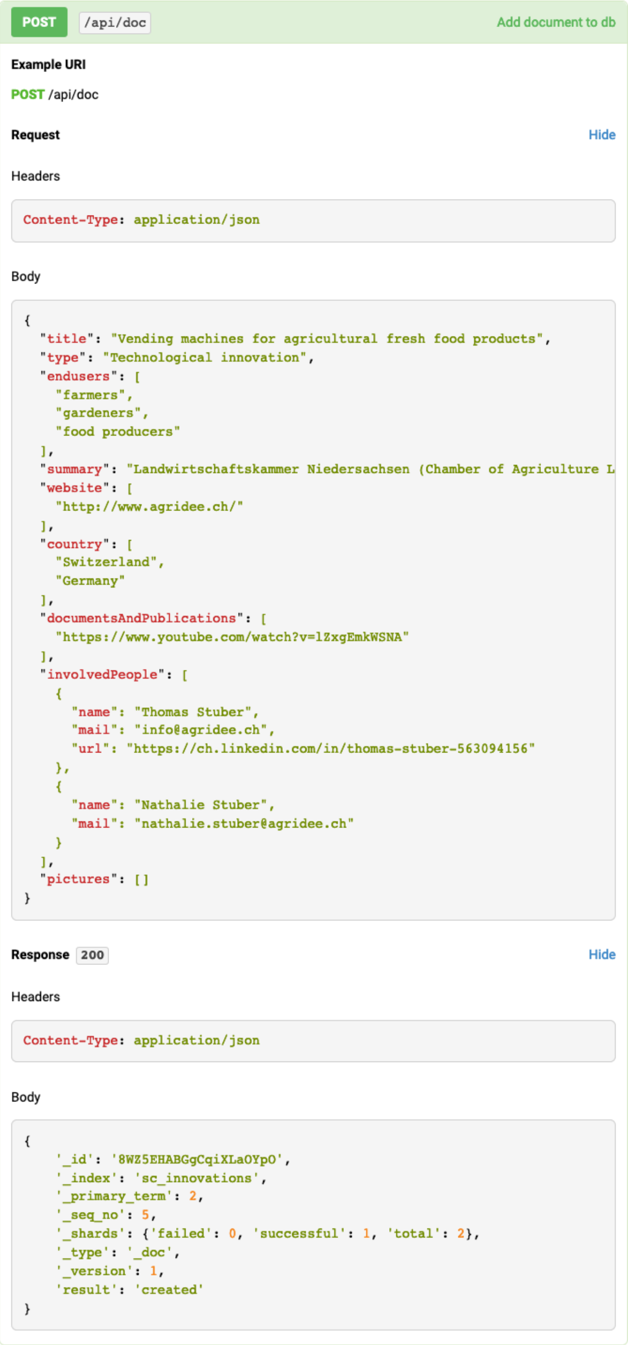
PARADOX cluster is installed in four water-cooled racks. The cooling system consists of 4 cooling modules (one within each rack), which are connected via a system of pipes with a large industrial chiller and configured so as to minimize power consumption.

PARADOX installation at the Scientific Computing Laboratory of Institute of Physics Belgrade.

B. REST API specification

This appendix gives a more technical specification of the developed SMARTCHAIN REST API.

B.1 Documents



POST `/api/doc` Add document to db

Example URI

POST /api/doc

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "title": "Vending machines for agricultural fresh food products",
  "type": "Technological innovation",
  "endusers": [
    "farmers",
    "gardeners",
    "food producers"
  ],
  "summary": "Landwirtschaftskammer Niedersachsen (Chamber of Agriculture L
  "website": [
    "http://www.agridee.ch/"
  ],
  "country": [
    "Switzerland",
    "Germany"
  ],
  "documentsAndPublications": [
    "https://www.youtube.com/watch?v=lZxgEmkWSNA"
  ],
  "involvedPeople": [
    {
      "name": "Thomas Stuber",
      "mail": "info@agridee.ch",
      "url": "https://ch.linkedin.com/in/thomas-stuber-563094156"
    },
    {
      "name": "Nathalie Stuber",
      "mail": "nathalie.stuber@agridee.ch"
    }
  ],
  "pictures": []
}
```

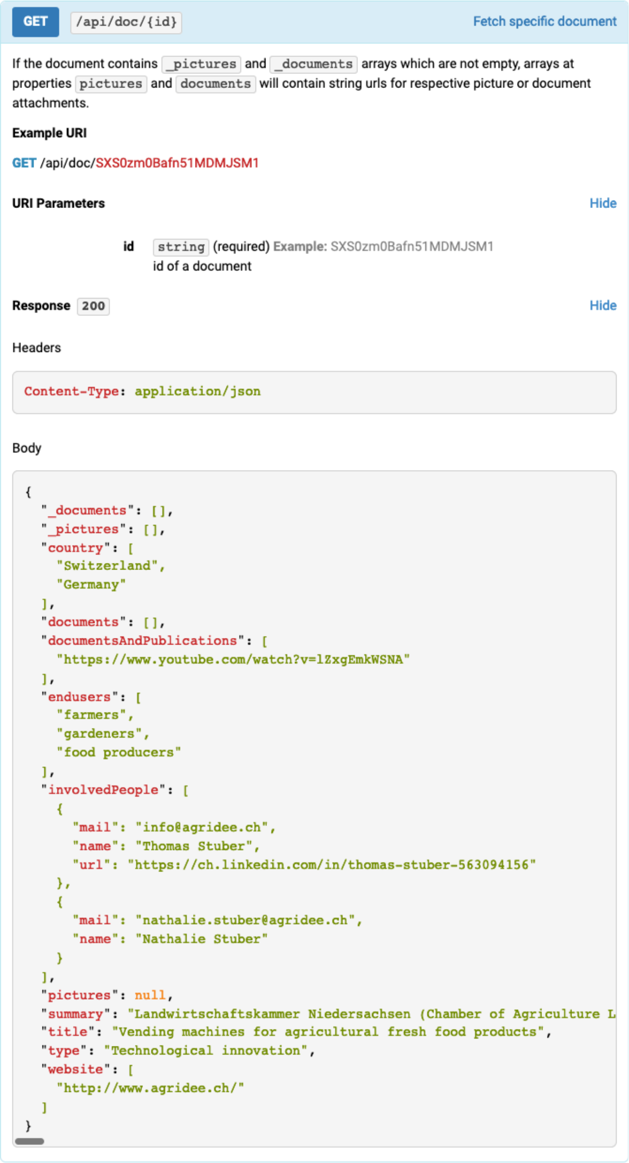
Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    '_id': '8WZ5EHABGgCqiXLaOYpO',
    '_index': 'sc_innovations',
    '_primary_term': 2,
    '_seq_no': 5,
    '_shards': {'failed': 0, 'successful': 1, 'total': 2},
    '_type': '_doc',
    '_version': 1,
    'result': 'created'
}
```



GET `/api/doc/{id}` Fetch specific document

If the document contains `_pictures` and `_documents` arrays which are not empty, arrays at properties `pictures` and `documents` will contain string urls for respective picture or document attachments.

Example URI

GET /api/doc/SXS0zm0Bafn51MDMJSM1

URI Parameters Hide

id `string` (required) **Example:** SXS0zm0Bafn51MDMJSM1
id of a document

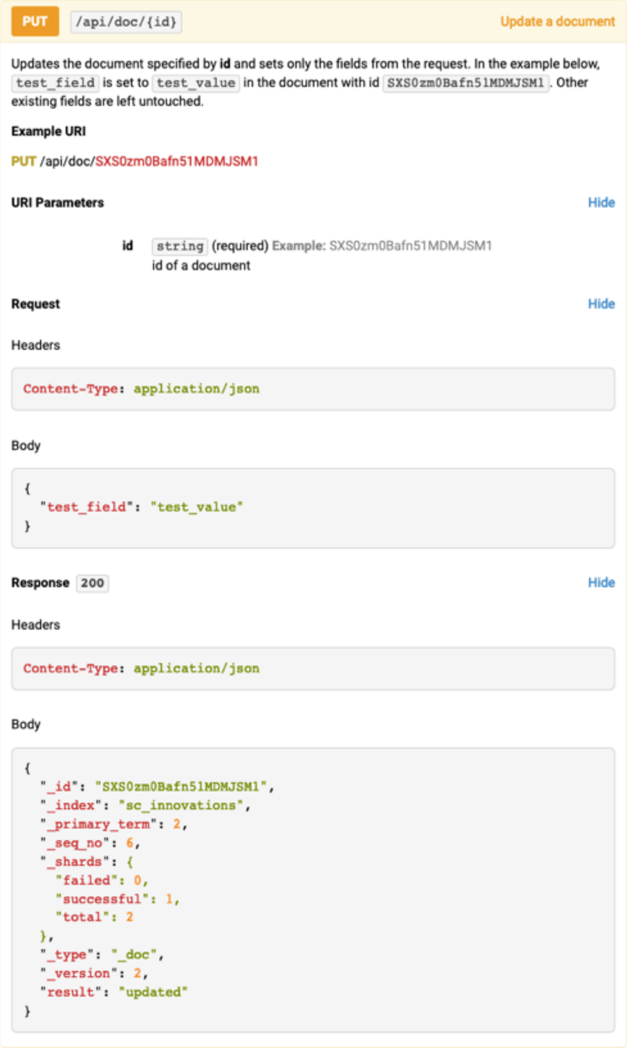
Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "_documents": [],
  "_pictures": [],
  "country": [
    "Switzerland",
    "Germany"
  ],
  "documents": [],
  "documentsAndPublications": [
    "https://www.youtube.com/watch?v=lZxgEmkWSNA"
  ],
  "endusers": [
    "farmers",
    "gardeners",
    "food producers"
  ],
  "involvedPeople": [
    {
      "mail": "info@agridee.ch",
      "name": "Thomas Stuber",
      "url": "https://ch.linkedin.com/in/thomas-stuber-563094156"
    },
    {
      "mail": "nathalie.stuber@agridee.ch",
      "name": "Nathalie Stuber"
    }
  ],
  "pictures": null,
  "summary": "Landwirtschaftskammer Niedersachsen (Chamber of Agriculture L
  "title": "Vending machines for agricultural fresh food products",
  "type": "Technological innovation",
  "website": [
    "http://www.agridee.ch/"
  ]
}
```



PUT `/api/doc/{id}` Update a document

Updates the document specified by **id** and sets only the fields from the request. In the example below, `test_field` is set to `test_value` in the document with id `SXS0zm0Bafn51MDMJSM1`. Other existing fields are left untouched.

Example URI

PUT /api/doc/SXS0zm0Bafn51MDMJSM1

URI Parameters Hide

id `string` (required) **Example:** SXS0zm0Bafn51MDMJSM1
id of a document

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "test_field": "test_value"
}
```

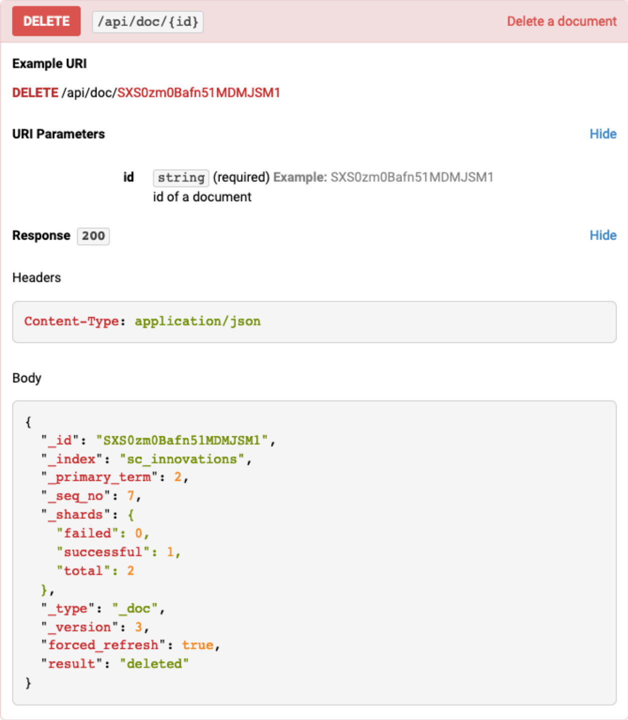
Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "_id": "SXS0zm0Bafn51MDMJSM1",
  "_index": "sc_innovations",
  "_primary_term": 2,
  "_seq_no": 6,
  "_shards": {
    "failed": 0,
    "successful": 1,
    "total": 2
  },
  "_type": "_doc",
  "_version": 2,
  "result": "updated"
}
```



DELETE `/api/doc/{id}` Delete a document

Example URI

DELETE /api/doc/SXS0zm0Bafn51MDMJSM1

URI Parameters Hide

id `string` (required) **Example:** SXS0zm0Bafn51MDMJSM1
id of a document

Response `200` Hide

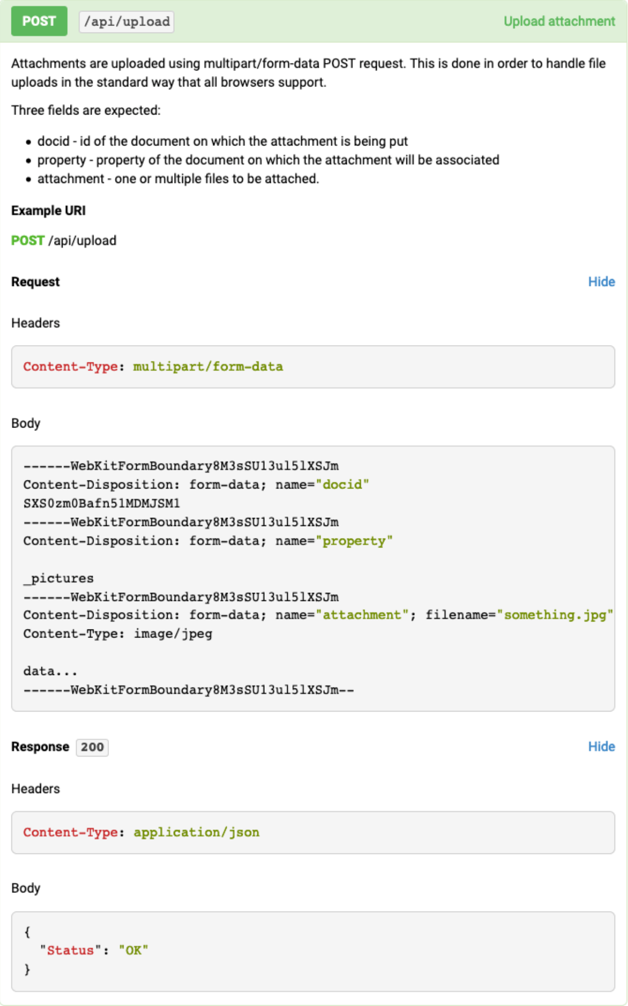
Headers

```
Content-Type: application/json
```

Body

```
{
  "_id": "SXS0zm0Bafn51MDMJSM1",
  "_index": "sc_innovations",
  "_primary_term": 2,
  "_seq_no": 7,
  "_shards": {
    "failed": 0,
    "successful": 1,
    "total": 2
  },
  "_type": "_doc",
  "_version": 3,
  "forced_refresh": true,
  "result": "deleted"
}
```

B.2 Attachments



POST /api/upload Upload attachment

Attachments are uploaded using multipart/form-data POST request. This is done in order to handle file uploads in the standard way that all browsers support.

Three fields are expected:

- docid - id of the document on which the attachment is being put
- property - property of the document on which the attachment will be associated
- attachment - one or multiple files to be attached.

Example URI

POST /api/upload

Request Hide

Headers

```
Content-Type: multipart/form-data
```

Body

```
------WebKitFormBoundary8M3sSU13ul5lXSJm
Content-Disposition: form-data; name="docid"
SXS0zm0Bafn51MDMJSM1
------WebKitFormBoundary8M3sSU13ul5lXSJm
Content-Disposition: form-data; name="property"

_pictures
------WebKitFormBoundary8M3sSU13ul5lXSJm
Content-Disposition: form-data; name="attachment"; filename="something.jpg"
Content-Type: image/jpeg

data...
------WebKitFormBoundary8M3sSU13ul5lXSJm--
```

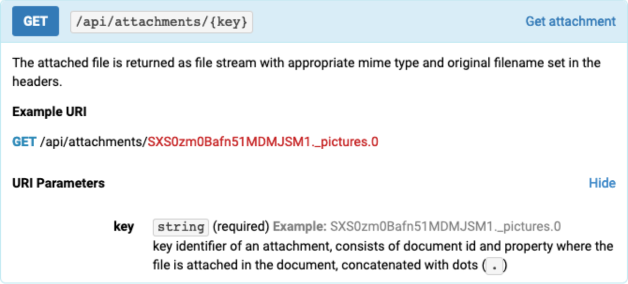
Response 200 Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "Status": "OK"
}
```



GET /api/attachments/{key} Get attachment

The attached file is returned as file stream with appropriate mime type and original filename set in the headers.

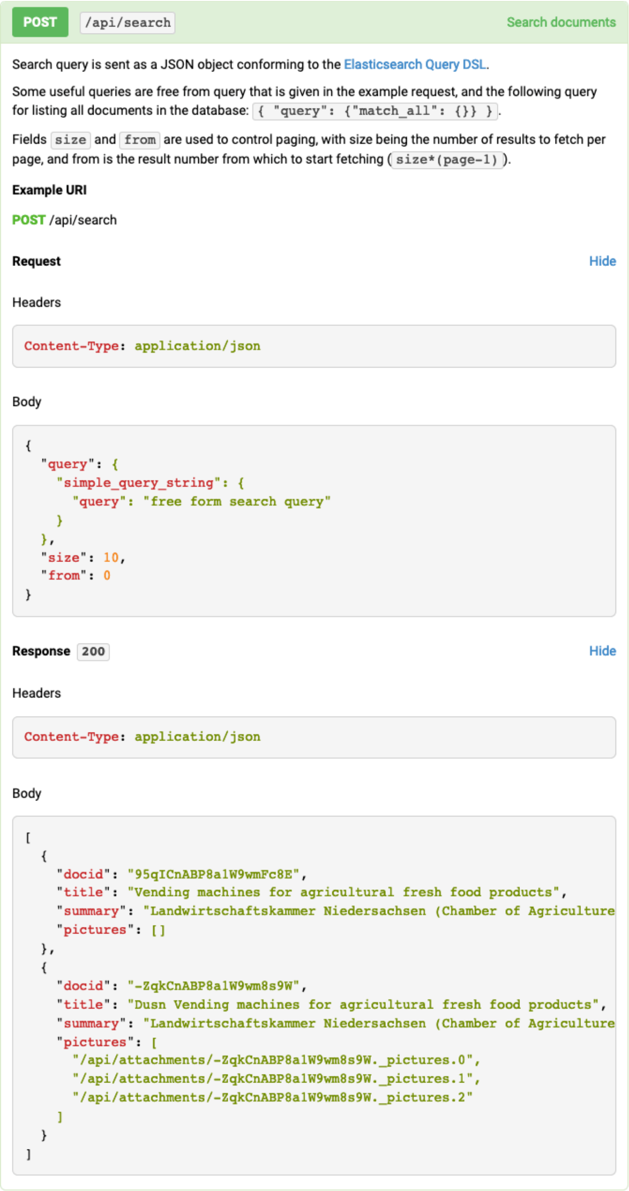
Example URI

GET /api/attachments/SXS0zm0Bafn51MDMJSM1._pictures.0

URI Parameters Hide

key `string` (required) **Example:** SXS0zm0Bafn51MDMJSM1._pictures.0
key identifier of an attachment, consists of document id and property where the file is attached in the document, concatenated with dots (`.`)

B.3 Search



POST `/api/search` Search documents

Search query is sent as a JSON object conforming to the Elasticsearch Query DSL.

Some useful queries are free from query that is given in the example request, and the following query for listing all documents in the database: `{ "query": {"match_all": {}} }`.

Fields `size` and `from` are used to control paging, with size being the number of results to fetch per page, and from is the result number from which to start fetching (`size*(page-1)`).

Example URI

POST /api/search

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "query": {
    "simple_query_string": {
      "query": "free form search query"
    }
  },
  "size": 10,
  "from": 0
}
```

Response 200 Hide

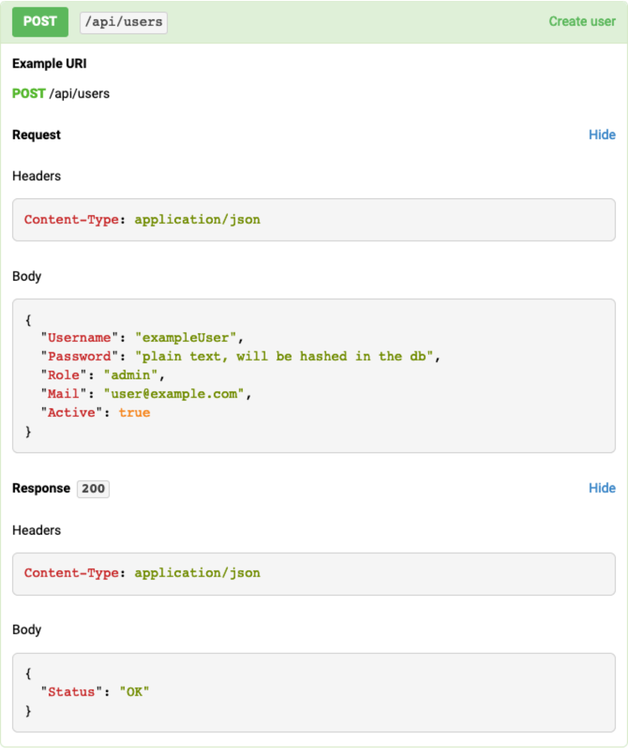
Headers

```
Content-Type: application/json
```

Body

```
[
  {
    "docid": "95qICnABP8alW9wmFc8E",
    "title": "Vending machines for agricultural fresh food products",
    "summary": "Landwirtschaftskammer Niedersachsen (Chamber of Agriculture
    "pictures": []
  },
  {
    "docid": "-ZqkCnABP8alW9wm8s9W",
    "title": "Dusn Vending machines for agricultural fresh food products",
    "summary": "Landwirtschaftskammer Niedersachsen (Chamber of Agriculture
    "pictures": [
      "/api/attachments/-ZqkCnABP8alW9wm8s9W._pictures.0",
      "/api/attachments/-ZqkCnABP8alW9wm8s9W._pictures.1",
      "/api/attachments/-ZqkCnABP8alW9wm8s9W._pictures.2"
    ]
  }
]
```

B.4 Administrative



POST /api/users Create user

Example URI

POST /api/users

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "Username": "exampleUser",
  "Password": "plain text, will be hashed in the db",
  "Role": "admin",
  "Mail": "user@example.com",
  "Active": true
}
```

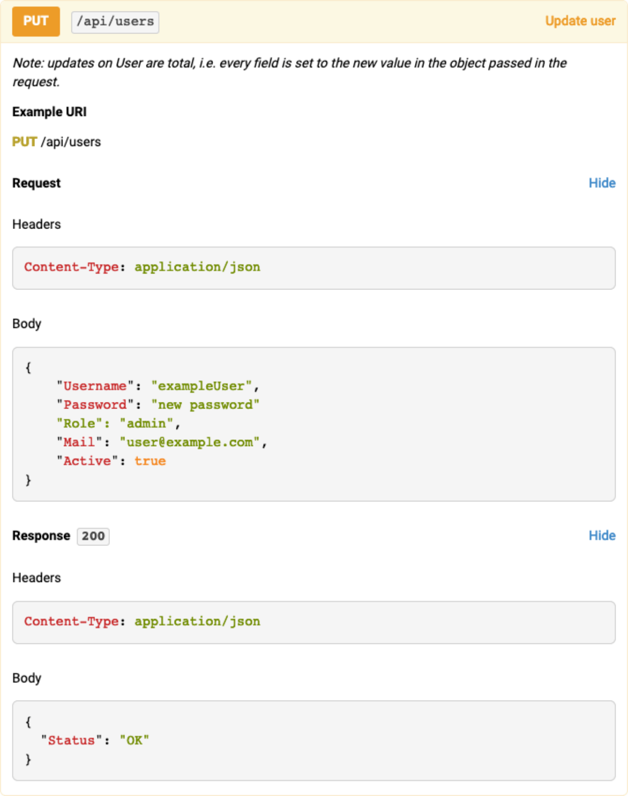
Response 200 Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "Status": "OK"
}
```



PUT /api/users Update user

Note: updates on User are total, i.e. every field is set to the new value in the object passed in the request.

Example URI

PUT /api/users

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    "Username": "exampleUser",
    "Password": "new password"
    "Role": "admin",
    "Mail": "user@example.com",
    "Active": true
}
```

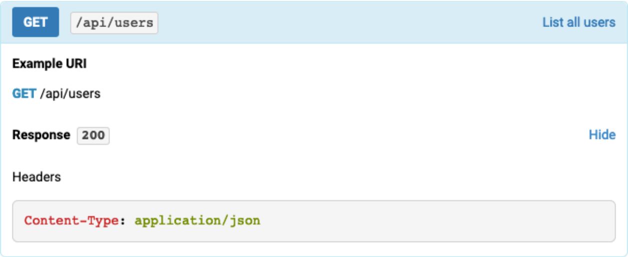
Response 200 Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "Status": "OK"
}
```



GET /api/users List all users

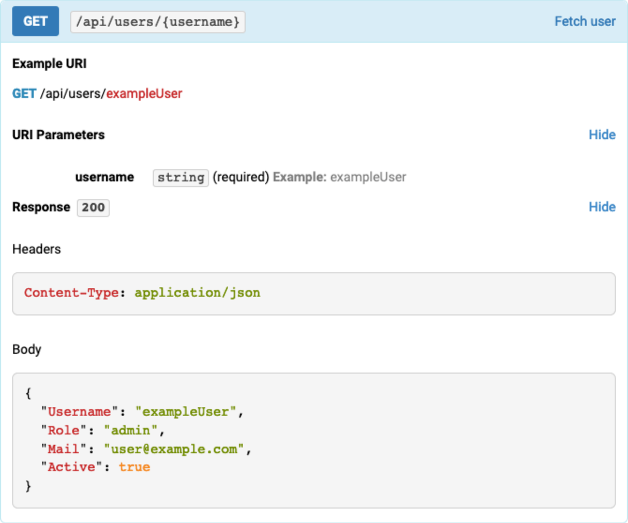
Example URI

GET /api/users

Response 200 Hide

Headers

```
Content-Type: application/json
```



GET /api/users/{username} Fetch user

Example URI

GET /api/users/exampleUser

URI Parameters Hide

username `string` (required) **Example:** exampleUser

Response 200 Hide

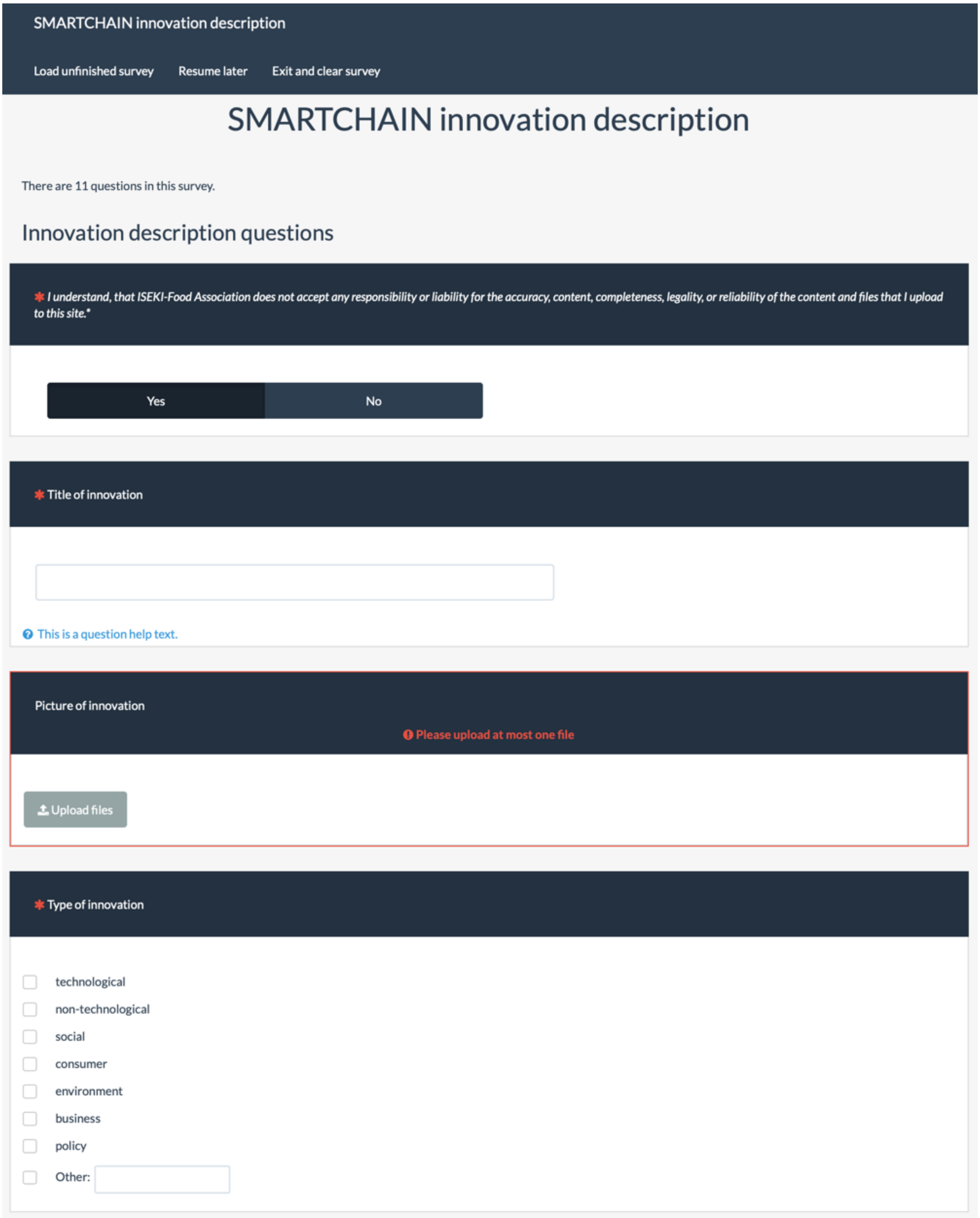
Headers

```
Content-Type: application/json
```

Body

```
{
  "Username": "exampleUser",
  "Role": "admin",
  "Mail": "user@example.com",
  "Active": true
}
```

C. Innovation Description Template



SMARTCHAIN innovation description

SMARTCHAIN innovation description

There are 11 questions in this survey.

Innovation description questions

** I understand, that ISEKI-Food Association does not accept any responsibility or liability for the accuracy, content, completeness, legality, or reliability of the content and files that I upload to this site.**

Yes No

* Title of innovation

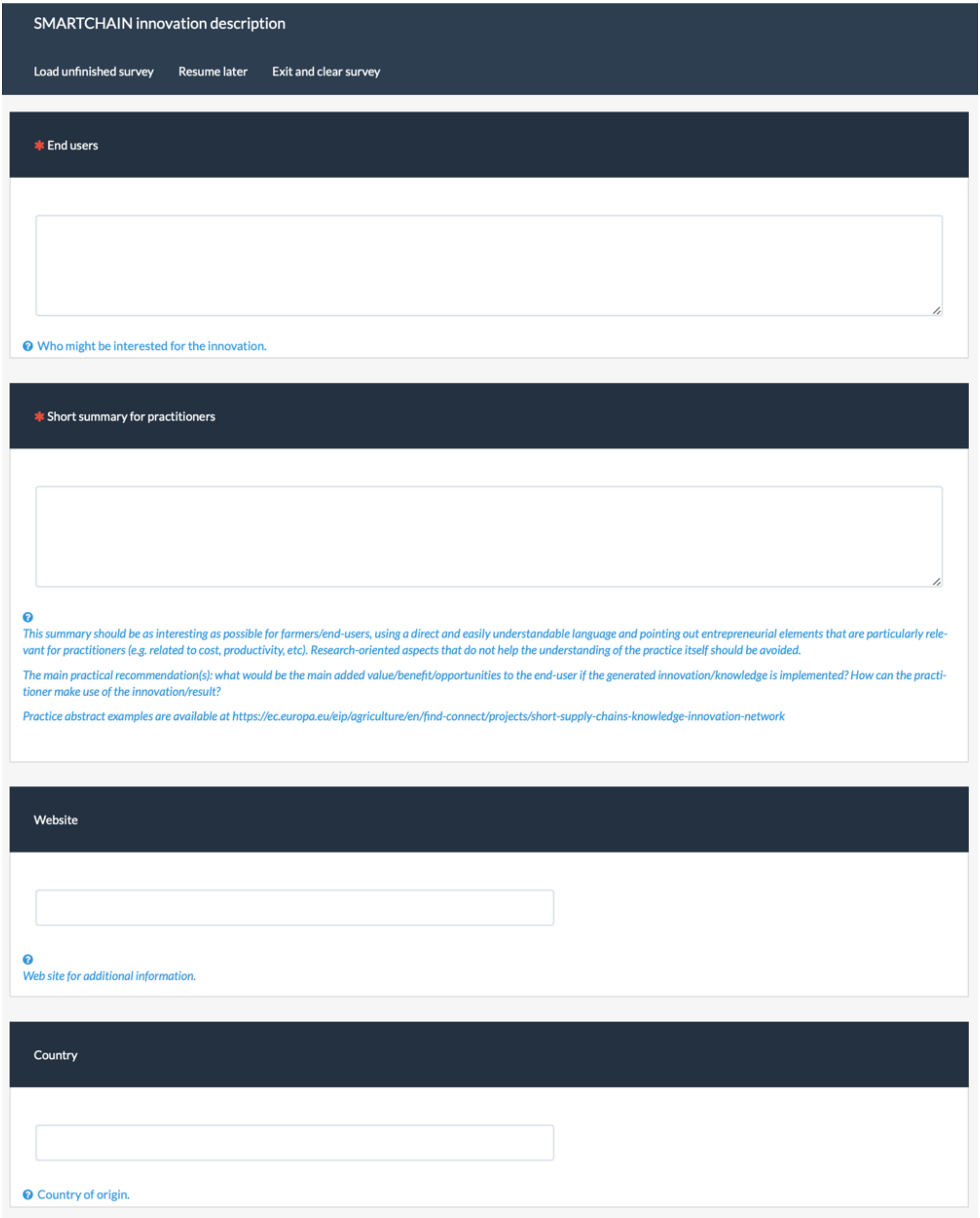
This is a question help text.

Picture of innovation

Please upload at most one file

* Type of innovation

□ technological
□ non-technological
□ social
□ consumer
□ environment
□ business
□ policy
□ Other:



SMARTCHAIN innovation description

* End users

Who might be interested for the innovation.

* Short summary for practitioners

This summary should be as interesting as possible for farmers/end-users, using a direct and easily understandable language and pointing out entrepreneurial elements that are particularly relevant for practitioners (e.g. related to cost, productivity, etc). Research-oriented aspects that do not help the understanding of the practice itself should be avoided.

The main practical recommendation(s): what would be the main added value/benefit/opportunities to the end-user if the generated innovation/knowledge is implemented? How can the practitioner make use of the innovation/result?

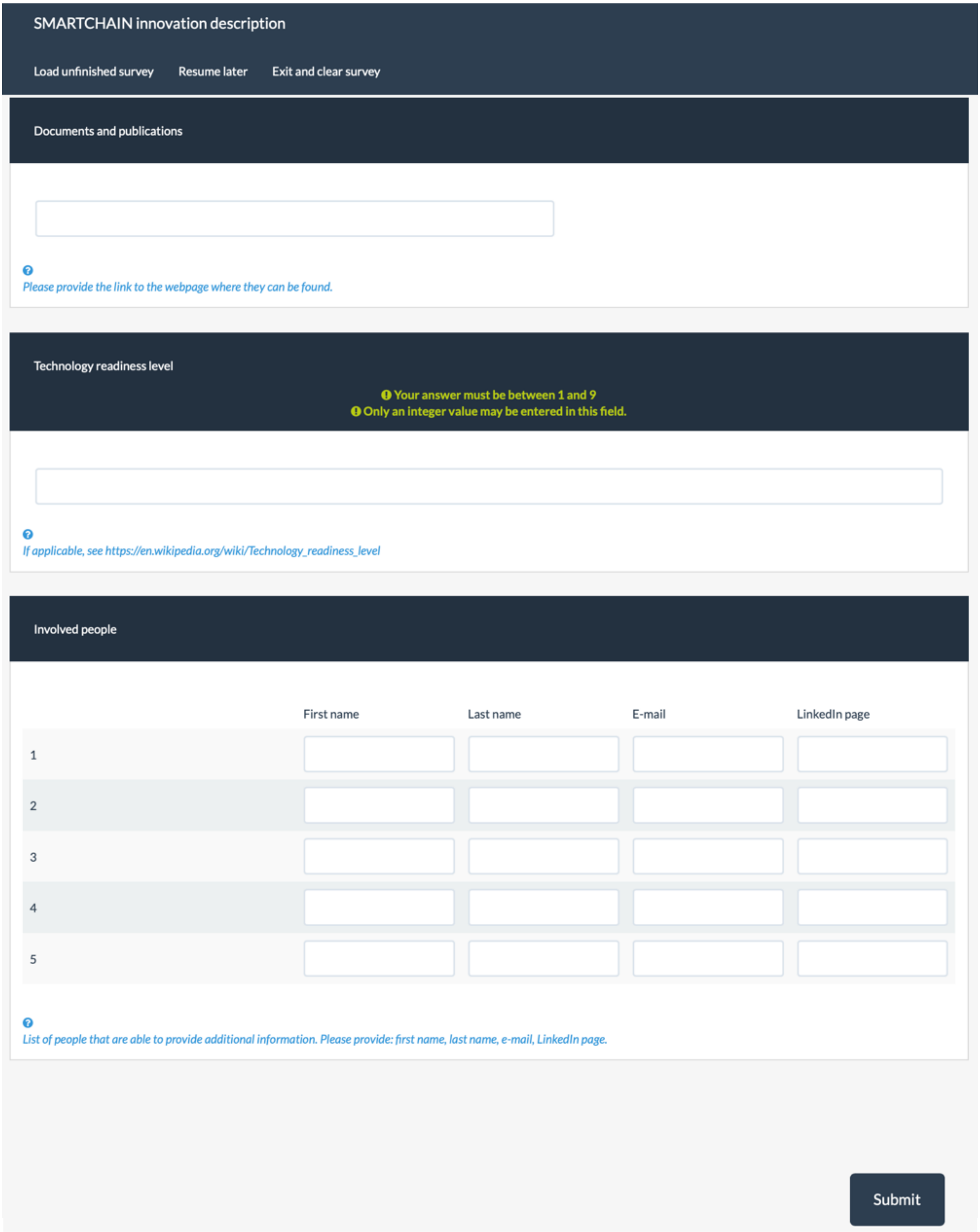
Practice abstract examples are available at https://ec.europa.eu/eip/agriculture/en/find-connect/projects/short-supply-chains-knowledge-innovation-network

Website

Web site for additional information.

Country

Country of origin.



SMARTCHAIN innovation description

Documents and publications

Please provide the link to the webpage where they can be found.

Technology readiness level

Your answer must be between 1 and 9
Only an integer value may be entered in this field.

If applicable, see https://en.wikipedia.org/wiki/Technology_readiness_level

Involved people

	First name	Last name	E-mail	LinkedIn page
1				
2				
3				
4				
5				

List of people that are able to provide additional information. Please provide: first name, last name, e-mail, LinkedIn page.

D. List of innovations registered within the inventory

- Vending machines for agricultural fresh food products, https://scinno.ipb.ac.rs/?_=/docid/TJq7WXABP8a1W9wmHtDT.
- Transforming your grocery store into a springboard for transformational activities, https://scinno.ipb.ac.rs/?_=/docid/TZq7WXABP8a1W9wmJNDB.
- Fruit press, https://scinno.ipb.ac.rs/?_=/docid/Tpq7WXABP8a1W9wmKtA_.
- La Charrette: Carsharing for local products, https://scinno.ipb.ac.rs/?_=/docid/XJq7WXABP8a1W9wmd9Db.
- New cheese products, https://scinno.ipb.ac.rs/?_=/docid/T5q7WXABP8a1W9wmL9CR.
- Dapper Texel. Consumer supported regenerative agriculture, https://scinno.ipb.ac.rs/?_=/docid/TTFv33gBZrz2W_JQ1MCb.
- Disinfections systems for the food sector, https://scinno.ipb.ac.rs/?_=/docid/UJq7WXABP8a1W9wmNND5.
- Foodhub.hu Nonprofit Ltd., https://scinno.ipb.ac.rs/?_=/docid/UZq7WXABP8a1W9wmOtBX.
- Freeze-drying solutions for food products, https://scinno.ipb.ac.rs/?_=/docid/Upq7WXABP8a1W9wmP9DK.
- Reaping the rewards of educational seeds, https://scinno.ipb.ac.rs/?_=/docid/U5q7WXABP8a1W9wmRdCL.
- Grounded festival, https://scinno.ipb.ac.rs/?_=/docid/yLP_JXMB_RFpeCHgnnsY.
- High-pressure processing (HPP) technology for food production, https://scinno.ipb.ac.rs/?_=/docid/VZq7WXABP8a1W9wmUNAj.
- Hermeneus marketplace platform, https://scinno.ipb.ac.rs/?_=/docid/VJq7WXABP8a1W9wmStDm.
- Local2Local's Food Distribution Software and Services, https://scinno.ipb.ac.rs/?_=/docid/VjGW33gBZrz2W_JQRMDv.
- Local2Local Talents, https://scinno.ipb.ac.rs/?_=/docid/VDGL33gBZrz2W_JQ-sC-.
- Mobile shared processing facility, https://scinno.ipb.ac.rs/?_=/docid/XZq7WXABP8a1W9wmfdAi.
- Mobile poultry coops, https://scinno.ipb.ac.rs/?_=/docid/Vpq7WXABP8a1W9wmVdCF.
- Operation Food Freedom, https://scinno.ipb.ac.rs/?_=/docid/UjF_33gBZrz2W_JQX8C4.
- Plant on Demand - The software solution for short food supply chain transactions, https://scinno.ipb.ac.rs/?_=/docid/u7MkCXMB_RFpeCHgU3uk.
- PlayLocal2locaL, https://scinno.ipb.ac.rs/?_=/docid/x7OKJXMB_RFpeCHgenus.
- Sales on webshop - Natuurlijk Vleespakket BV (NV), https://scinno.ipb.ac.rs/?_=/docid/LjEW33gBZrz2W_JQUcB1.
- Smartshortchain, https://scinno.ipb.ac.rs/?_=/docid/ybMTJnMB_RFpeCHgjntc.
- Community Supported Agriculture (CSA) in Germany: Solidarische Landwirtschaft (SoLaWi), https://scinno.ipb.ac.rs/?_=/docid/WJq7WXABP8a1W9wmYNA3.
- Smart greenhouse, https://scinno.ipb.ac.rs/?_=/docid/w7NaH3MB_RFpeCHgHHt1.

- Mini kiwi fruit plantation, https://scinno.ipb.ac.rs/?_=/docid/vLMwH3MB_RFpeCHginsx.
- Szomor-Farm Hungary, https://scinno.ipb.ac.rs/?_=/docid/vbM8H3MB_RFpeCHgNnsA.
- Weather monitoring for agriculture, https://scinno.ipb.ac.rs/?_=/docid/lszNwnIBD2zyg2THyKpq.
- Agrobot berry harvester, https://scinno.ipb.ac.rs/?_=/docid/vrNAH3MB_RFpeCHgKXtu.
- Robotic apple picker, https://scinno.ipb.ac.rs/?_=/docid/v7NEH3MB_RFpeCHgeHv9.
- KisanHub, https://scinno.ipb.ac.rs/?_=/docid/wLNIH3MB_RFpeCHgLXtN.
- Mobile Slaughterhouse, https://scinno.ipb.ac.rs/?_=/docid/wbNSH3MB_RFpeCHgj3vd.
- Automatic milking, grazing and cow health monitoring, https://scinno.ipb.ac.rs/?_=/docid/wrNWH3MB_RFpeCHgcntY.
- Evaluating effective microorganisms for post-harvest fungal contamination, https://scinno.ipb.ac.rs/?_=/docid/xLNiH3MB_RFpeCHgCHsT.
- Valorization of the goat milk into cheese, E.coli detection in goat milk, https://scinno.ipb.ac.rs/?_=/docid/N7OGVHQB_RFpeCHg3Xwk.
- Regulation of the Short Food Supply Chains, https://scinno.ipb.ac.rs/?_=/docid/ObOQVHQB_RFpeCHgQHzo.
- High Hydrostatic Pressure (HHP) processing technology for food production, https://scinno.ipb.ac.rs/?_=/docid/l8z0wnIBD2zyg2THDarz.
- Quality schemes - AGPFGA /Association Gersoise pour la Promotion du Foie Gras/, https://scinno.ipb.ac.rs/?_=/docid/mMwEw3IBD2zyg2TH16ru.
- FreshSense, https://scinno.ipb.ac.rs/?_=/docid/zrNTN3MB_RFpeCHgjnuQ.
- Temperature Monitoring Labels, https://scinno.ipb.ac.rs/?_=/docid/z7NXN3MB_RFpeCHgqXvn.
- Food Safety, Sanitation and Disinfection Indicators, https://scinno.ipb.ac.rs/?_=/docid/0LNbN3MB_RFpeCHgjnvL.
- Food labeling and nutritional analyses without lab tests, https://scinno.ipb.ac.rs/?_=/docid/mswiw3IBD2zyg2THtap0.
- Invivo PS PP, https://scinno.ipb.ac.rs/?_=/docid/xbN6H3MB_RFpeCHgTnsD.
- IFSC, https://scinno.ipb.ac.rs/?_=/docid/xrOAH3MB_RFpeCHg5Hsh.
- Hydro Cooler for loose produce, https://scinno.ipb.ac.rs/?_=/docid/yrM-N3MB_RFpeCHg-HvR.
- Dairy chiller, https://scinno.ipb.ac.rs/?_=/docid/y7NCN3MB_RFpeCHgwnvz.
- Biosensor system (lactate biosensor) that ensures quality and efficiency in the fruit juice industry, https://scinno.ipb.ac.rs/?_=/docid/MrMVTnMB_RFpeCHgg3xj.
- Food radar system for the detection of foreign objects with low density in foods, https://scinno.ipb.ac.rs/?_=/docid/zLNIN3MB_RFpeCHgLnua.
- Portable NIR scanner, https://scinno.ipb.ac.rs/?_=/docid/zbNPN3MB_RFpeCHg2Hsb.
- Hyperspectral analysis for quality control, https://scinno.ipb.ac.rs/?_=/docid/mcwdw3IBD2zyg2THkKoa.

- Vacuum cooling solutions for food products, https://scinno.ipb.ac.rs/?_=/docid/m8wnw3IBD2zyg2THCKpn.
- Biodegradable packaging with natural indicator substances, https://scinno.ipb.ac.rs/?_=/docid/2LOON3MB_RFpeCHgSHtt.
- Ultrasound application: Homogenisation and pasteurisation, https://scinno.ipb.ac.rs/?_=/docid/2bORN3MB_RFpeCHg43sm.
- Ultrasound application: Meat pickling/curing, https://scinno.ipb.ac.rs/?_=/docid/ncw0w3IBD2zyg2THnKqG.
- Mobile juice trailers, https://scinno.ipb.ac.rs/?_=/docid/2rObN3MB_RFpeCHgL3tf.
- Biodegradable active packaging, https://scinno.ipb.ac.rs/?_=/docid/27OfN3MB_RFpeCHgBHvI.
- Biodegradable packaging, https://scinno.ipb.ac.rs/?_=/docid/3LOiN3MB_RFpeCHgvXs1.
- Biodegradable packaging, https://scinno.ipb.ac.rs/?_=/docid/3bOmN3MB_RFpeCHgF3t8.
- Fruit press for fruit juices production, https://scinno.ipb.ac.rs/?_=/docid/3rOqN3MB_RFpeCHgAXvS.
- New ways to sell “too ripe” fruit, https://scinno.ipb.ac.rs/?_=/docid/37OtN3MB_RFpeCHgzntt.
- Tuna loins cutting by ultrasound, https://scinno.ipb.ac.rs/?_=/docid/0bNsN3MB_RFpeCHgmntK.
- Fruit and vegetable juice production, https://scinno.ipb.ac.rs/?_=/docid/0rNwN3MB_RFpeCHgRHtk.
- Diversification of products through new goat cheese production, https://scinno.ipb.ac.rs/?_=/docid/07NzN3MB_RFpeCHgsHsh.
- Freeze drying for food products, https://scinno.ipb.ac.rs/?_=/docid/1LN5N3MB_RFpeCHganu6.
- Modified Atmosphere Packaging (MAP), https://scinno.ipb.ac.rs/?_=/docid/1bOCN3MB_RFpeCHgQHsQ.
- Vacuum-microwave drying technology for food products, https://scinno.ipb.ac.rs/?_=/docid/1rOGN3MB_RFpeCHgXXs9.
- Mackerel gender assessment, https://scinno.ipb.ac.rs/?_=/docid/nMwuw3IBD2zyg2THfKre.
- Disinfection systems for the food sector, https://scinno.ipb.ac.rs/?_=/docid/17OKN3MB_RFpeCHgtnsx.
- Vending Machines, Automatic distributors of farm products, https://scinno.ipb.ac.rs/?_=/docid/4LO4N3MB_RFpeCHgGntp.
- Refrigerated pickup station – cool lockers, Temperature-controlled lockers for groceries, https://scinno.ipb.ac.rs/?_=/docid/nsw4w3IBD2zyg2TH4qqz.
- Predictive analytics of orders, https://scinno.ipb.ac.rs/?_=/docid/57PpN3MB_RFpeCHgmHs-.
- Logistic based on regional network, https://scinno.ipb.ac.rs/?_=/docid/JTEP33gBZrz2W_JQFsBR.
- Farmers’ market Liliomkert, https://scinno.ipb.ac.rs/?_=/docid/6bPlOHMB_RFpeCHgHHts.
- One-stop shopping for catering professionals, https://scinno.ipb.ac.rs/?_=/docid/6rMBOXMB_RFpeCHgj3sJ.
- Naaber online marketplace, https://scinno.ipb.ac.rs/?_=/docid/67MaOXMB_RFpeCHgznsX.

- YouTyúk, https://scinno.ipb.ac.rs/?_=/docid/7LMdOXMB_RFpeCHg-3vo.
- WePick, https://scinno.ipb.ac.rs/?_=/docid/7bMhOXMB_RFpeCHgW3si.
- All-in-one packaging, https://scinno.ipb.ac.rs/?_=/docid/7rM1OXMB_RFpeCHgonuU.
- Regional corner in the supermarket and in the point of sales, https://scinno.ipb.ac.rs/?_=/docid/77M6OXMB_RFpeCHgvHss.
- Multi-channel sale, https://scinno.ipb.ac.rs/?_=/docid/4bO7N3MB_RFpeCHg6nuB.
- Fast food chains, https://scinno.ipb.ac.rs/?_=/docid/8LNCOXMB_RFpeCHgiXtA.
- Collaboration – "Joint distribution", https://scinno.ipb.ac.rs/?_=/docid/8bNFOXMB_RFpeCHge3va.
- Involvement of the consumers, https://scinno.ipb.ac.rs/?_=/docid/8rNMOXMB_RFpeCHgenuu.
- Agro-tourism chain, https://scinno.ipb.ac.rs/?_=/docid/87NYOXMB_RFpeCHg53tn.
- Moving outlets, https://scinno.ipb.ac.rs/?_=/docid/9LNbOXMB_RFpeCHga3sz.
- Rizikó-Ker Kft., https://scinno.ipb.ac.rs/?_=/docid/9bNeOXMB_RFpeCHgmHvf.
- Deák Mansion, https://scinno.ipb.ac.rs/?_=/docid/9rNhOXMB_RFpeCHgHHu8.
- Pijaca na klik, https://scinno.ipb.ac.rs/?_=/docid/97NlOXMB_RFpeCHgonsB.
- IntelliFood, https://scinno.ipb.ac.rs/?_=/docid/-LNoOXMB_RFpeCHghnsV.
- CHERRY, https://scinno.ipb.ac.rs/?_=/docid/-bN3OXMB_RFpeCHgu3vE.
- Producer's shop, https://scinno.ipb.ac.rs/?_=/docid/4rPAN3MB_RFpeCHgIntw.
- TagItSmart, https://scinno.ipb.ac.rs/?_=/docid/-rN6OXMB_RFpeCHg9Xsj.
- FRISBEE Tool, https://scinno.ipb.ac.rs/?_=/docid/-7N-OXMB_RFpeCHg1nuh.
- Small depots for personalized supply of perishable foods, https://scinno.ipb.ac.rs/?_=/docid/_LOBOXMB_RFpeCHgu3vn.
- Post service cold chain - Natuurlijk Vleespakket BV (NV), https://scinno.ipb.ac.rs/?_=/docid/47PPN3MB_RFpeCHghHvb.
- Demand-driven supply chain, Local2Local, https://scinno.ipb.ac.rs/?_=/docid/5LPTN3MB_RFpeCHgbXtw.
- Shared production facilities for the preservation and packaging of primary agricultural production, https://scinno.ipb.ac.rs/?_=/docid/5bPZN3MB_RFpeCHglntF.
- LANDPACK – Green packaging solutions from grain fields, https://scinno.ipb.ac.rs/?_=/docid/5rPdN3MB_RFpeCHg1HsU.
- Group Integrity - Internal Control System, https://scinno.ipb.ac.rs/?_=/docid/PHozxHYBi5OYF83M-AlK.
- Collective Selling Points (PVC), https://scinno.ipb.ac.rs/?_=/docid/BbPaOXMB_RFpeCHgnnwN.

- Collection of rules and regulations, Guidelines and Good Practices, https://scinno.ipb.ac.rs/?_=/docid/NbNsVHQB_RFpeCHglHyu.
- Small and Mobile Food Processing and Slaughter Houses, https://scinno.ipb.ac.rs/?_=/docid/NLNpVHQB_RFpeCHg_3x3.
- Check Organic - The Integrity Platform, https://scinno.ipb.ac.rs/?_=/docid/OHovxHYBi5OYF83MlQm1.
- Participatory Guarantee Systems as a mechanism for building trust of parties, https://scinno.ipb.ac.rs/?_=/docid/Q3o3xHYBi5OYF83MtQkg.
- Traceability-Vitaproject, https://scinno.ipb.ac.rs/?_=/docid/ALOlOXMB_RFpeCHgFXy3.
- Vitaproject, https://scinno.ipb.ac.rs/?_=/docid/AbOyOXMB_RFpeCHgq3yM.
- Private umbrella brand for traditional food products, https://scinno.ipb.ac.rs/?_=/docid/ArO1OXMB_RFpeCHgZXyL.
- TRACE, https://scinno.ipb.ac.rs/?_=/docid/A7O5OXMB_RFpeCHgGHzZ.
- fTRACE, https://scinno.ipb.ac.rs/?_=/docid/BLPDOXMB_RFpeCHgKHxh.
- Austrian Small Farm Taxation, https://scinno.ipb.ac.rs/?_=/docid/M7NYVHQB_RFpeCHgiHzk.
- Social media marketing - Biofruits, https://scinno.ipb.ac.rs/?_=/docid/BrOuRHMB_RFpeCHge3xB.
- Exposition of products, https://scinno.ipb.ac.rs/?_=/docid/D7PdRHMB_RFpeCHglnyY.
- Wine “orders” and wine trails, https://scinno.ipb.ac.rs/?_=/docid/ELPhRHMB_RFpeCHgkXwM.
- Turn2bio, https://scinno.ipb.ac.rs/?_=/docid/EbPkRHMB_RFpeCHgIXz7.
- Pick-Your-Own farm, https://scinno.ipb.ac.rs/?_=/docid/ErPmRHMB_RFpeCHg1nzJ.
- Biodegradable, home compostable packaging, https://scinno.ipb.ac.rs/?_=/docid/E7PpRHMB_RFpeCHg0ny3.
- Storytelling, https://scinno.ipb.ac.rs/?_=/docid/FLPsRHMB_RFpeCHgPHwv.
- Lead user approach - Alce Nero, https://scinno.ipb.ac.rs/?_=/docid/B7O2RHMB_RFpeCHg13zS.
- Marketing tools – Alce Nero, https://scinno.ipb.ac.rs/?_=/docid/CLO6RHMB_RFpeCHgHXyM.
- Crowdfunding - Natuurlijk Vleespakket BV, https://scinno.ipb.ac.rs/?_=/docid/KTES33gBZrz2W_JQFcAu.
- The Balaton Circle, https://scinno.ipb.ac.rs/?_=/docid/CrPARHMB_RFpeCHg3nxU.
- Éltető Balaton-felvidék brand system, https://scinno.ipb.ac.rs/?_=/docid/C7PLRHMB_RFpeCHgBXxe.
- House of Hungarian Pálinka, https://scinno.ipb.ac.rs/?_=/docid/DLPNRHMB_RFpeCHgzHxk.
- House of Hungarian Wine, https://scinno.ipb.ac.rs/?_=/docid/DbPQRHMB_RFpeCHgsnwq.
- Qualivita, https://scinno.ipb.ac.rs/?_=/docid/DrPTRHMB_RFpeCHgsHxl.

- Community-supported agriculture (Solidarische Landwirtschaft), https://scinno.ipb.ac.rs/?_=/docid/FbM4THMB_RFpeCHgFnxA.
- New packaging for fruit juices, https://scinno.ipb.ac.rs/?_=/docid/HbOYTXMB_RFpeCHgRXw5.
- Implementation of vending machines, https://scinno.ipb.ac.rs/?_=/docid/HrOaTXMB_RFpeCHgkHx8.
- Hermeneus Marketplace Platform, https://scinno.ipb.ac.rs/?_=/docid/FrM7THMB_RFpeCHgQ3wh.
- Cooperative Supermarket, https://scinno.ipb.ac.rs/?_=/docid/F7M_THMB_RFpeCHgBHxf.
- Negotiation power of farmers, https://scinno.ipb.ac.rs/?_=/docid/GLNBTHMB_RFpeCHgk3wY.
- Farm diversification Buffalo Cheese, https://scinno.ipb.ac.rs/?_=/docid/GbODTXMB_RFpeCHgM3xk.
- Farm diversification Tekeres-valley, https://scinno.ipb.ac.rs/?_=/docid/GrOHTXMB_RFpeCHgV3z1.
- Slow food, https://scinno.ipb.ac.rs/?_=/docid/G7ORTXMB_RFpeCHgyXwH.
- OrganicNet, https://scinno.ipb.ac.rs/?_=/docid/HLOVTXMB_RFpeCHgCHw4.
- Rolling price, https://scinno.ipb.ac.rs/?_=/docid/NrNyVHQB_RFpeCHgFnyE.
- Sales on webshop - Natuurlijk Vleespakket BV (NV), https://scinno.ipb.ac.rs/?_=/docid/MjEY33gBZrz2W_JQf8C4.
- PerishABLE, https://scinno.ipb.ac.rs/?_=/docid/J7PPTXMB_RFpeCHgoXzr.
- Babbler, https://scinno.ipb.ac.rs/?_=/docid/KLPRTXMB_RFpeCHgEnzN.
- QIfresh, https://scinno.ipb.ac.rs/?_=/docid/KbPUTXMB_RFpeCHgx3wX.
- Tsenso, https://scinno.ipb.ac.rs/?_=/docid/KrPaTXMB_RFpeCHgjnx8.
- INTOFOOD, https://scinno.ipb.ac.rs/?_=/docid/K7PcTXMB_RFpeCHgN3wo.
- Virtual market platform for farmers, https://scinno.ipb.ac.rs/?_=/docid/n8xOw3IBD2zyg2TH3Kq9.
- Smart label, https://scinno.ipb.ac.rs/?_=/docid/ILPCTXMB_RFpeCHgH3xz.
- AgroNET platform for digital farming solutions, https://scinno.ipb.ac.rs/?_=/docid/IbPETXMB_RFpeCHghnwy.
- Logiqs rolling bench systems for greenhouses, https://scinno.ipb.ac.rs/?_=/docid/IrPGTXMB_RFpeCHgIXwg.
- SenseFly Ag 360 aerial crop analysis system, https://scinno.ipb.ac.rs/?_=/docid/I7PITXMB_RFpeCHgvnyc.
- Fresh Produce Trade App, https://scinno.ipb.ac.rs/?_=/docid/JLPKTXMB_RFpeCHgaXwK.
- FOOODER, https://scinno.ipb.ac.rs/?_=/docid/JbPMTXMB_RFpeCHgxHzg.
- AgriAware, https://scinno.ipb.ac.rs/?_=/docid/JrPOTXMB_RFpeCHgMnzg.

- Vacuum-microwave technology - innovative dehydration technology for new food products, https://scinno.ipb.ac.rs/?_=/docid/WZq7WXABP8a1W9wmZdCe.
- Vacuum cooling solutions for food products, https://scinno.ipb.ac.rs/?_=/docid/Wpq7WXABP8a1W9wmbNC6.
- Voedsel in de buurt (Food in the neighourhood), https://scinno.ipb.ac.rs/?_=/docid/TjFv33gBZrz2W_JQ2cCc.
- Zala Valley Open Farms – community based business model and social coaching, https://scinno.ipb.ac.rs/?_=/docid/W5q7WXABP8a1W9wmctAH.

References

[1] Project SMARTCHAIN 773785 – Annex I - Description of the action.

[2] SMARTCHAIN D6.2 – Design of the inventory of innovations and related interactive tools.

[3] SMARTCHAIN D6.4 – Implementation of the inventory of innovations, related interactive tools.

[4] R. Gheorghe, M. L. Hinman, and R. Russo, Elasticsearch in Action, Manning Publications (2015).

[5] Official Elasticsearch webpage, https://www.elastic.co/.

[6] J. Zitting and C. Mattmann, Tika in Action, Manning Publications (2011).

[7] Apache Tika - a content analysis toolkit, https://tika.apache.org/.

[8] LevelDB - fast key-value storage library, https://github.com/google/leveldb.

[9] Casbin - authorization library, https://casbin.org/.

[10] A. A. A. Donovan, B. W. Kernighan, The Go Programming Language, Addison-Wesley (2015).

[11] S. V. Chekanov, Numeric Computation and Statistical Data Analysis on the Java Platform, Springer (2016).

[12] O. Gospodnetic, E. Hatcher, Lucene in Action, Manning Publications (2010).

[13] Wt, C++ Web Toolkit - Emweb, https://www.webtoolkit.eu/