
Publish and Share

Concept Manual

Version 1.0

2019-08-01



Email: contact@its4land.com

Web: <https://its4land.com/>

Table of Contents

INTRODUCTION	2
OVERVIEW	3
LAND ADMINISTRATION CONTEXT	6
THE PUBLISH AND SHARE PUBLIC API	7
THE PUBLISH AND SHARE PUBLIC API	7
USING THE PUBLIC API	8
PROJECT STRUCTURE	8
SUPPORT FOR LADM CONCEPTS	9
TOOLS AND RUNTIME	11
THE DATA DISSEMINATION INTERFACE	12
USAGE MODELS OF PUBLISH AND SHARE	14
SUPPORT FROM PUBLISH AND SHARE FOR THE APPLICATION MODEL	15
SUPPORT FROM PUBLISH AND SHARE FOR THE INTEGRATION MODEL	16
SUPPORT FROM PUBLISH AND SHARE FOR THE TOOLS MODEL	17
SUPPORT FROM PUBLISH AND SHARE FOR THE PLATFORM EXTENSION MODEL	18
CHARACTERISTICS OF THE MODELS AND SUGGESTIONS WHICH MODEL SHOULD BE USED IN WHICH SITUATION	20
APPLICATION VS. INTEGRATION	20
TOOLS VS. PLATFORM EXTENSION MODEL	21
SUGGESTED READING	24

Introduction

This handbook documents the main concepts of the Publish and Share platform and provides a technical overview of the architecture of the system. The manual is intended for any stakeholder involved in development and operation of land administration workflows on Publish and Share.

The Publish and Share platform is primarily intended as a development platform to be used in land administration related projects. Typical users of its4land Publish and Share are independent software vendors (ISV) or integrators to create services / applications for end-users.

Publish and Share offers a set of high level geocloud based services for developers of land administration systems to use or integrate spatial references for tenure registration. Using these geocloud services allows a vendor to concentrate on functionalities required by their customer instead of re-implementing common solutions for common problems. All these services are based on the concepts introduced by LADM.

Four different usage models have been developed to improve Land Administration workflows and tasks by Publish and Share:

Application: A self-contained application for an end-user.

Integration: Publish and Share functionality is integrated into a Land Administration workflow.

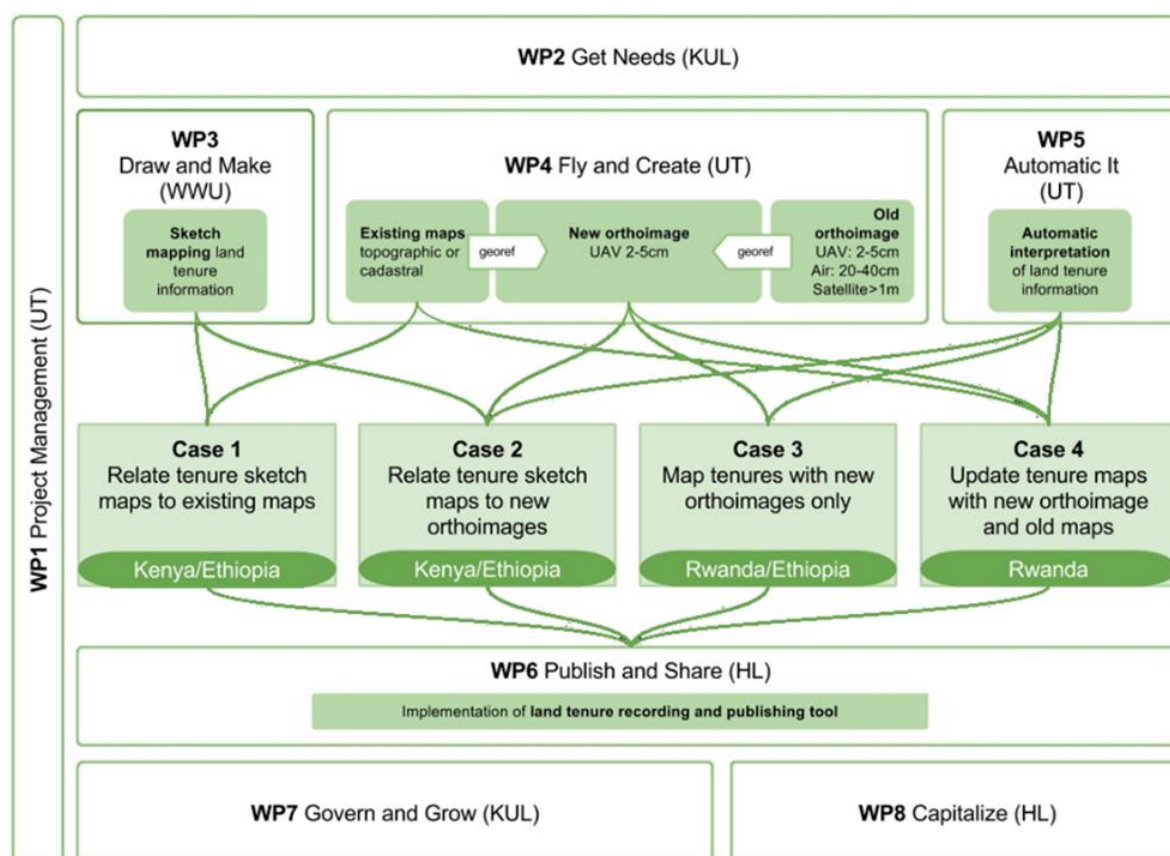
Tools: A self-contained functionality used for integration or applications.

Platform extension: New core functionalities of the platform.

Apart from the technical aspects, the different usage models provide a base for different models for a later commercialization and use of the its4land toolbox by addressing different stakeholders in a commercial setting.

Overview

The Publish and Share platform can be considered on the one hand as a runtime environment for the tools developed in its4land and on the other hand as a provider of data and information for existing land administration systems (LAS) or other tools. The platform will be accessible via service interfaces based on standards from Open Geospatial Consortium (OGC) and World-Wide-Web Consortium (W3C). The modelling of the interfaces follows the concepts introduced by Land Administration Domain Model (LADM). External systems like LAS or planning systems can use the service interfaces to integrate data into their own processes, based on specific national rules. The usage scenarios and workflows to combine the its4land tools with land administration systems will be defined and implemented to a prototype level. For demonstration purpose Publish and Share combines the tools and methods developed in the other its4land work packages (*Draw and Make*, *Fly and Create* and *Automate It*) in a technical platform (see Error! Reference source not found.).



The implementation of the Publish and Share platform follows a toolbox approach and will provide a framework of common APIs and services used by all its4land tools. From this toolbox, a user can select those its4land tools fitting his tasks best.

As per the paradigm of geocloud, the tools will be implemented as services accessible via an Application Programming Interface (API) based on web standards. Tools which are capable of calling a REST API can make use of any kind of Publish and Share service, like public APIs or storage services. For 3rd party tools developed outside the scope of its4land, it is the responsibility of the 3rd party tool vendor or creator to adapt their tools to Publish and Share.

Its4land's claim is the development of state of the art methods for recording land rights with special consideration of the needs of local stakeholders in developing countries. To achieve a close integration of local stakeholders, adopting needs and land tenure concepts is required.

Publish and Share is a geocloud platform with the capability to host and integrate tools and data to facilitate land tenure recording services and applications. With Publish and Share, UAV image capturing, qualitative data processing and boundary delineation can be integrated into existing or new land administration workflows. For seamless integration with existing systems, Publish and Share is implemented on standards like REST, LADM and OGC OWS. Publish and Share is not primarily intended as an out of the box software for end-users. As a geocloud platform, Publish and Share provides technical services to support land administration projects in implementing their own tools, applications and workflows.

It is composed of four technical components:

1. A set of public REST-APIs that allows tools and applications to interact with the Publish and Share platform
2. A tool runtime environment for smart sketchmaps, image processing, boundary delineation and other tools
3. A data repository for alphanumeric, geo, binary and image data
4. OGC services for data dissemination

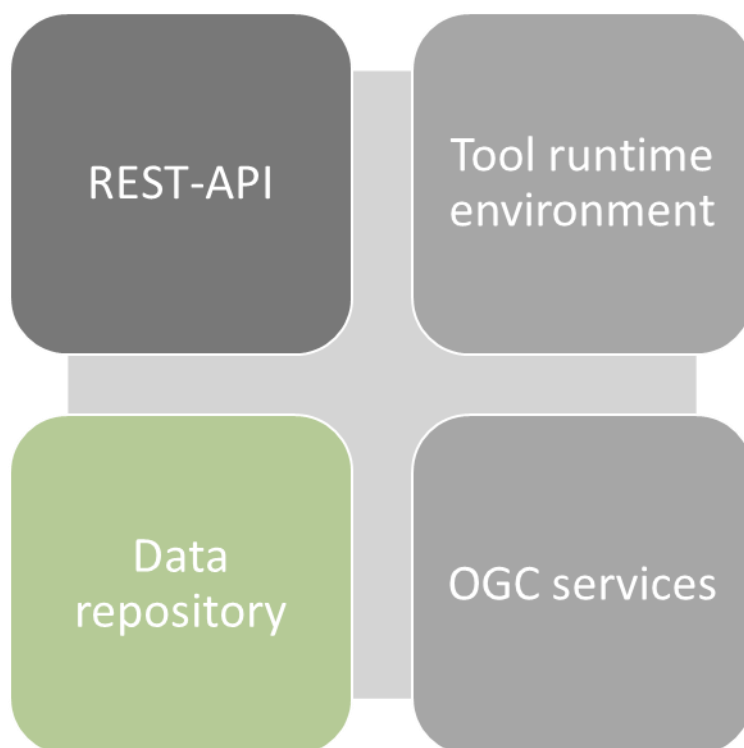


Figure 1: **Components of the Publish and Share platform**

All components developed for Publish and Share are open source. The components can be hosted in cloud environment like Amazon Web Services, Microsoft Azure or a private cloud environment.

Land Administration Context

The establishment of a Land Administration System is a complex and challenging task. Land Administration Systems are never out of the box systems. Differing legal, social, historical and technical aspects between different countries mean that these systems are highly specific and localized. Over the past few years, different attempts were undertaken to streamline the concepts of Land Administration Systems and provide guidance for development and implementation, like Land Administration Domain Model (LADM) [1] and Fit for Purpose (FFP) [2]. LADM establishes a common ontology of land administration concepts. Fit for Purpose focuses on needs of people and their relationship to the land, to support security of tenure for all and to sustainably manage land use and natural resources.

Different stakeholders have different views on a land administration system. A common ontology simplifies communication. Publish and Share has a general software engineering perspective on land administration systems and assumes no specific software development process or technique. From this perspective typical stakeholders/users who are involved in such a process are:

- Operator of such a system
- (Software) developer, including business analyst, data modeller, and software architects, etc.
- Domain specialist, like surveyors, lawyers, etc.
- Other software systems, that have to communicate with a system under development.

Both LADM and FFP provide a strong conceptual foundation for land administration systems. However, on the level of the technical implementation as software, most land administration systems are one of a kind. Its4land Publish and Share is developed as a general and universal platform for operating, integrating and disseminating land administration workflows and functionalities with a focus on base data capturing. Publish and Share offers a set of high level geocloud based services for developers of land administration systems.

Using these geocloud services allows the developer to concentrate on functionalities required by their customer instead of re-implementing common solutions for common problems. All these services are based on the concepts introduced by LADM. A developer who is familiar with LADM will recognize known concepts like SpatialUnits, AdminSources, etc. These concepts are available in the Publish and Share API. Their localized meaning depends on country-specific implementation.

More details about LADM and Publish and Share can be found in “Publish and Share: LADM Manual”.

The Publish and Share Public API

Publish and Share provides two types of interfaces for external systems – the REST based API called Public API, and OGC web services called Data Dissemination Interface (DDI).

The Publish and Share Public API

The Publish and Share platform offers developers and system integrators a Public API on top of which custom applications can be built or platform features integrated into existing land administration systems. The API provides an implementation for LADM concepts like *SpatialUnits* among others, such that they can be used in a programmatic fashion in an information system. The API itself is a REST API [3] and is served over HTTP(S) [4] [5]. Implementing a REST API makes integration on any platform or programming language seamless. Figure 2 shows the main user interface of the online documentation of the Public API.

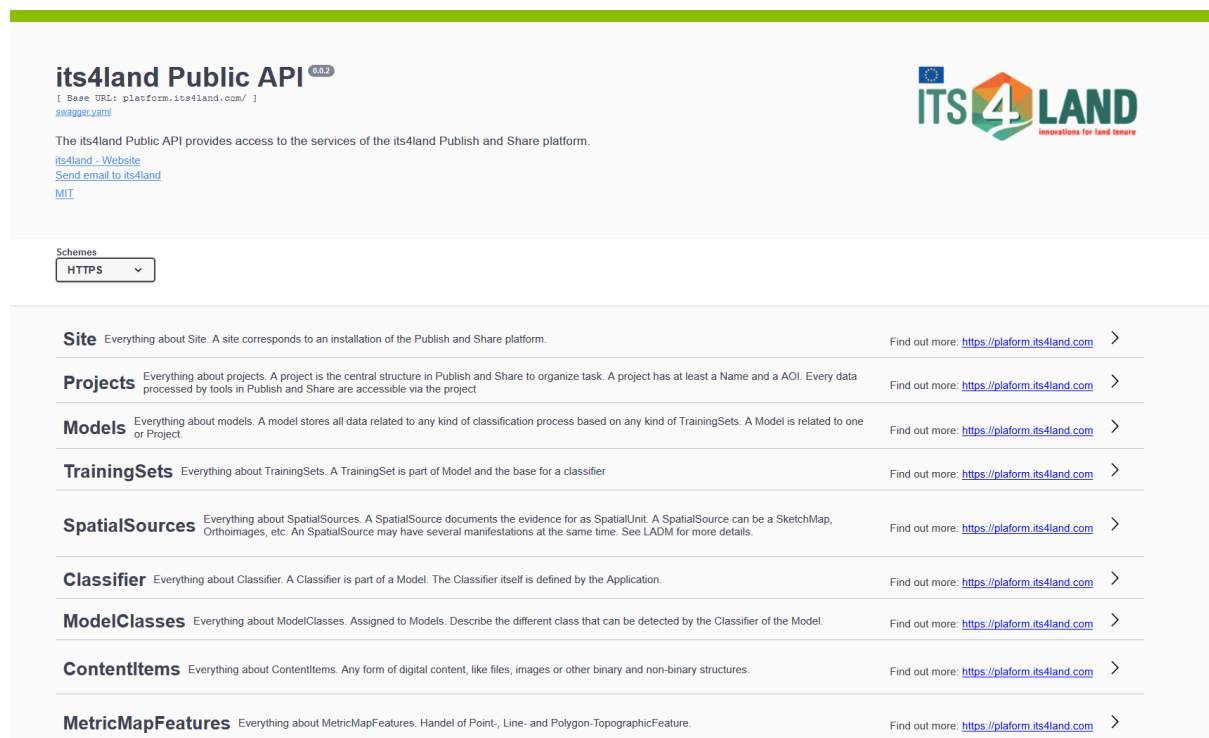


Figure 2: Publish and Share Public API documentation

The Public API lends itself to multiple usage models which are discussed in the following section. These can include an end-user who interacts using only a GUI client, a developer wishing to extend their legacy system with features of Publish and Share or wanting to extend the platform to customize it to their needs. The API is designed to achieve these

workflows by following an open standard for specification and documentation which includes example requests and responses.

Using the Public API

Using the API consists of making one or more HTTP requests via the standard methods – GET, POST, PUT, PATCH or DELETE [6]. Each request is stateless and does not retain information from previous requests. The API uses the JavaScript Object Notation a.k.a JSON format for data interchange. In response to a request, the server sends a response. The response contains HTTP status codes which indicate the request status and optionally a JSON payload in the response body if the request was successful. It is up to the client to formulate a request and interpret the response as per the API specification.

Project Structure

Users of the Public API will first start by creating a *Project*. A Project holds contextual information such as the input and output and related metadata supplied by the user, like name, description, timestamps, tags, etc. Additionally, a project defines the spatial extent of the area of interest, with its geometry described by a polygon following the GeoJSON specification [7]. Following this convention ensures that the project area is also viewable in third party tools (e.g. QGIS) via a simple API call.

Creation of a Project is as simple as supplying the necessary metadata to the */projects* endpoint via an HTTP POST request. The project area is defined as an area of interest (AOI). Every project that is created gets a unique ID. This ID can be used to get detailed information about the project as well as to add further resources such as spatial sources and other documents to the project. Figure 3 illustrates the sequence when interacting with the API's *projects* endpoint. The client initiates requests by providing required data and in turn, the server responds by creating a new project resource (when using the POST method) or retrieving information about existing projects (using GET method).

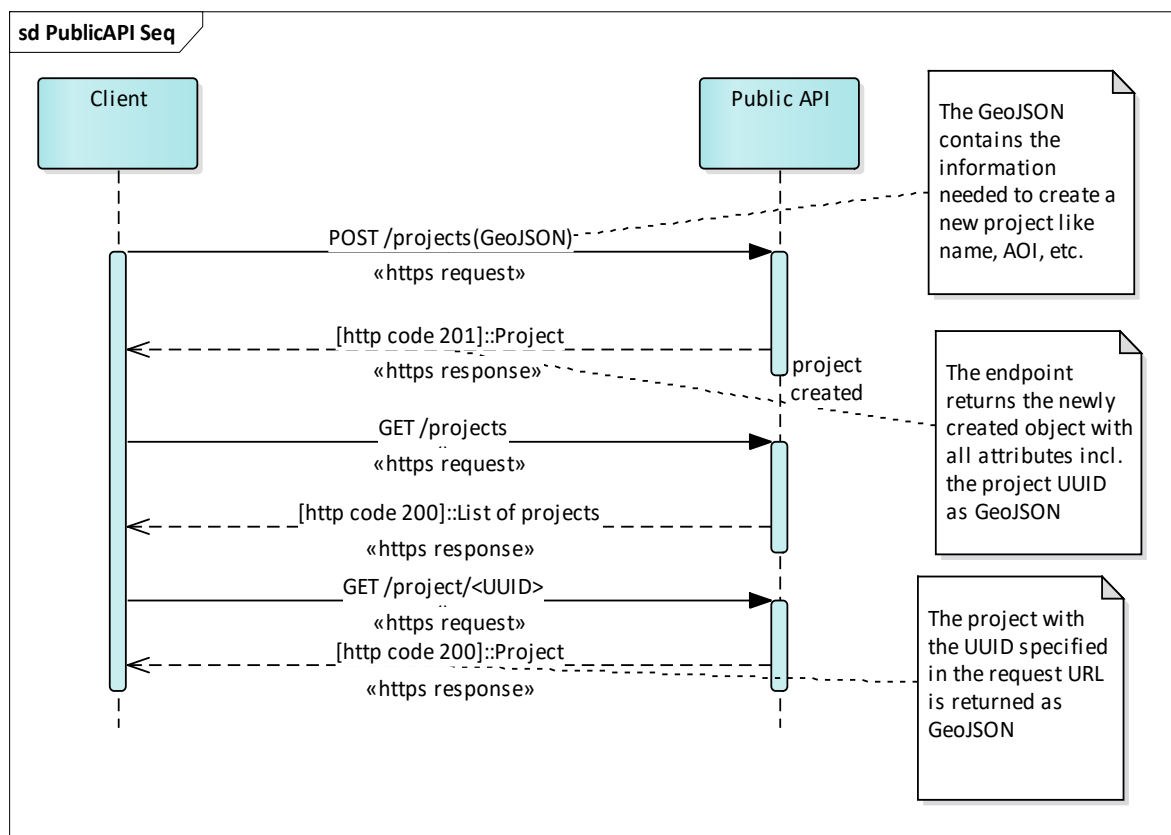


Figure 3: Sequence diagram for the project endpoint

Support for LADM concepts

Adding an LADM concept such as a SpatialSource involves a few more API calls. A SpatialSource itself is an abstract entity which digitally can take the form of a file in some given format. Hence, the actual contents of the SpatialSource are stored via a separate ContentItems endpoint, which uploads the file and saves it to the backend storage returning a ContentItem ID. Creating a spatial source involves using this ContentItem ID and making a POST request to a SpatialSource endpoint to register the file/item as a spatial source. Spatial sources themselves can be created standalone and added to one or more projects later. Another way is to create a spatial source as a sub-resource of an existing Project. The Publish and Share API allows one to obtain the SpatialSource associated with a project or as standalone objects. These calls are illustrated in the sequence diagram shown in Figure 4.

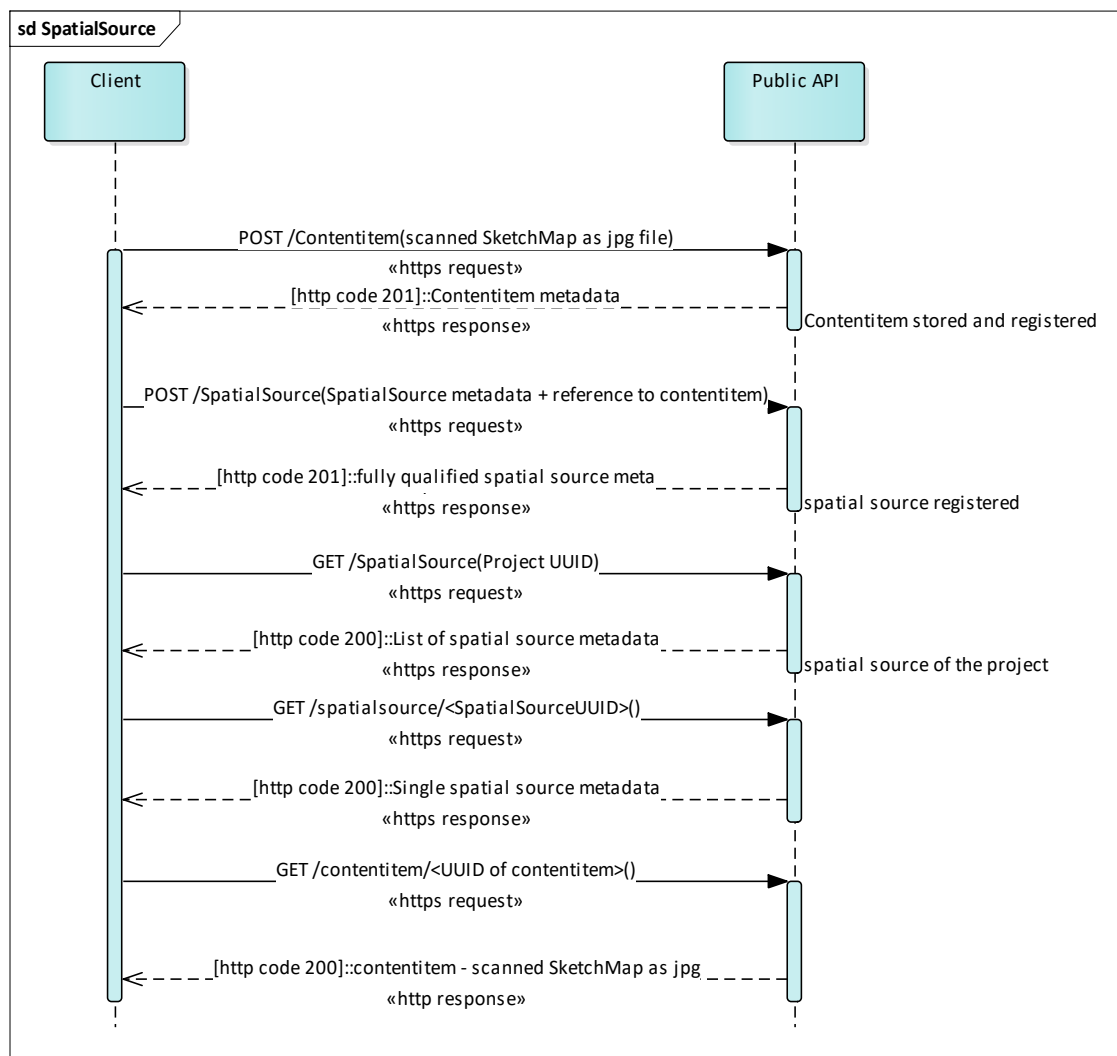


Figure 4: Sequence diagram for spatial source usage

Tools and runtime

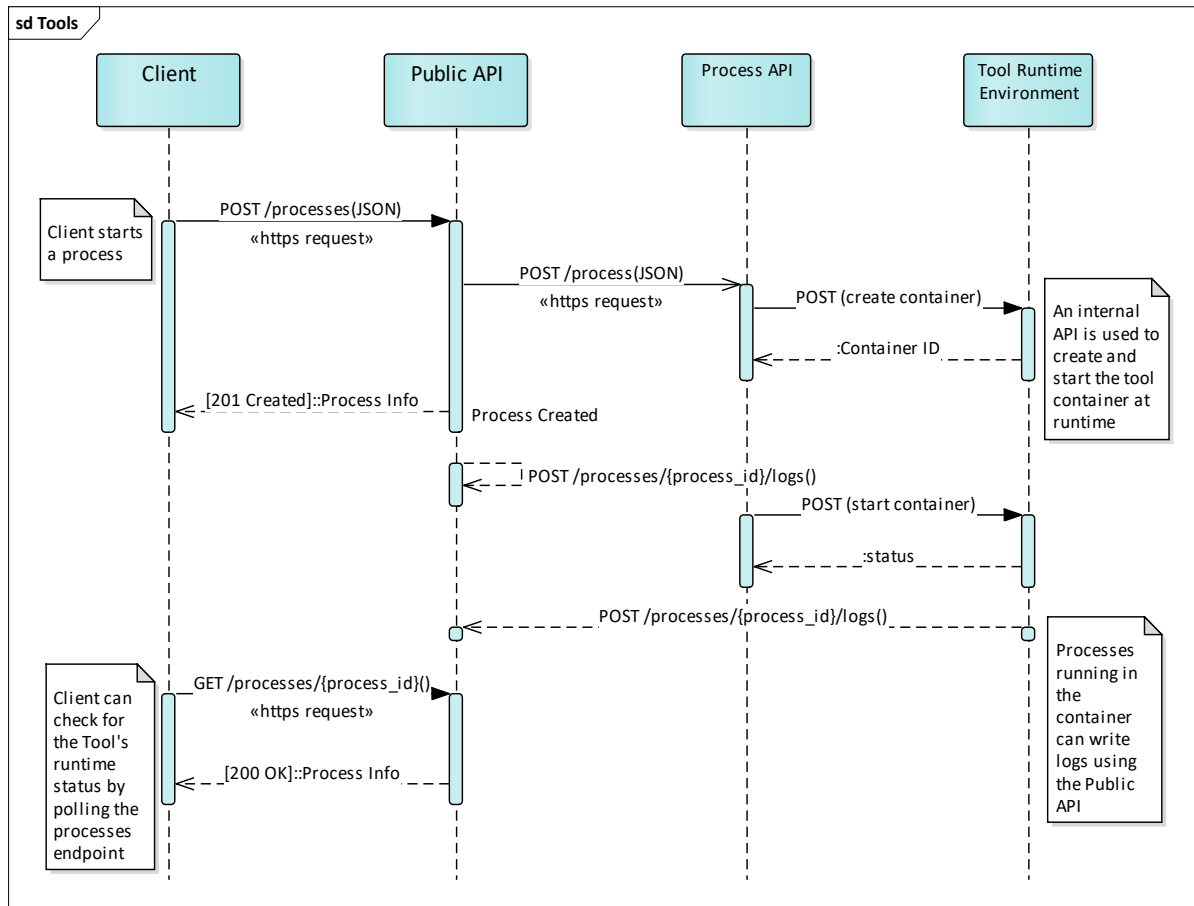


Figure 5: Sequence diagram for process handling

One of the most powerful features of the Publish and Share platform is the ability to run resource heavy workloads for land administration using independently developed tools. Examples for such tools are *Draw and Make* and *Fly and Create* developed as part of its4land. Resources can need lots of storage space and require computationally demanding tools to process them. Storage requires persistence and is handled by back-end stores in a cloud based environment. Computation is more transient and requires processor time for only as long as a tool is running.

To run a tool in Publish and Share, one creates a *Process* using the Public API. The Public API, in turn, invokes an internal API which is used for process management. The internal API then spawns containers and starts tools with the appropriate arguments passed to it by the client. Containers run the tool in an isolated manner, and can only access platform resources via

API calls. A logging facility is available to update the process status. Figure 5 illustrates this functionality:

1. The client creates a *Process* via the POST */processes* endpoint. The response contains a unique id assigned to the newly created process – *{id}*
2. The Public API wraps the request by the client and passes it on to an internal Process API. It also logs this information.
3. The Process API creates a container for the requested tool and starts it with the supplied arguments.
4. The tool starts running and is able to log the status via POST to the */processes/{id}/logs* endpoint.
5. The client is able to query for the runtime status of the started process by querying GET to the */processes/{id}* endpoint.

Using containers for running tools provides the benefits of scalability, security and allows tool writers greater flexibility. These advantages are further discussed in later sections.

The data dissemination interface

The DDI provides several WMS and WFS services for data stored in Publish and Share. In comparison to the Public API, the DDI only allows read only access to data in Publish and Share. The DDI is limited to spatial features like spatial units or boundary face strings. DDI also provides high performance access to orthoimages and other raster data stored in Publish and Share via WMS services.

The main advantage of the DDI is the ease of use. Nearly every GIS Tool on the market can use OGC services out of the box. This allows a lightweight data usage by workflows developed by the application or integration usage model.

For the visualisation of spatial raster data the DDI should be preferred instead of the Public API, since the WMS provides data regarding coordinate systems, the extents and the resolution, in such a way that they can be directly used in map. Spatial raster data that should be processed by an algorithm is better retrieved by the Public API, since the Public API provides the data in full resolution and extent.

Access to spatial vector data depends on the capabilities of the client software. If the client software is already capable of using WFS, this could be used. The Public API requires more efforts on the client side to handle requests and responses. But on the other hand, the

Public API allows better control, like advanced filtering capabilities compared to the WFS interface. If data should be manipulated, the use of the Public API is required.

Table 1 provides an overview of the available spatial data classes in Publish and Share and their access services in DDI.

Data	WMS	WFS	Remarks
Spatial Unit	X	X	Only to 2D polygon profile
Boundary Face String	X	X	
Metric Map Feature	X	X	
Orthomosaic	X		
Other raster data	X		
Other vector data	X	X	

Table 1: **DDI access methods**

The data dissemination interface is built on top of GeoServer.

The DDI service of Publish and Share instance is available via:

<https://<URL Publish and Share Platform>/DDI/WMS> for the WMS service instance and
<https://<URL Publish and Share Platform>/DDI/WFS> for the WFS service instance.

The Public API provides endpoints to register already georeferenced and stored orthoimages as DDI layer.

Usage models of Publish and Share

The Publish and Share platform is primarily intended as a development platform to be used in land administration related projects. Typical users of its4land Publish and Share are independent software vendors (ISV) or integrators to create services / applications for end-users. Four main usage models have been identified in the its4land project and are supported by Publish and Share.

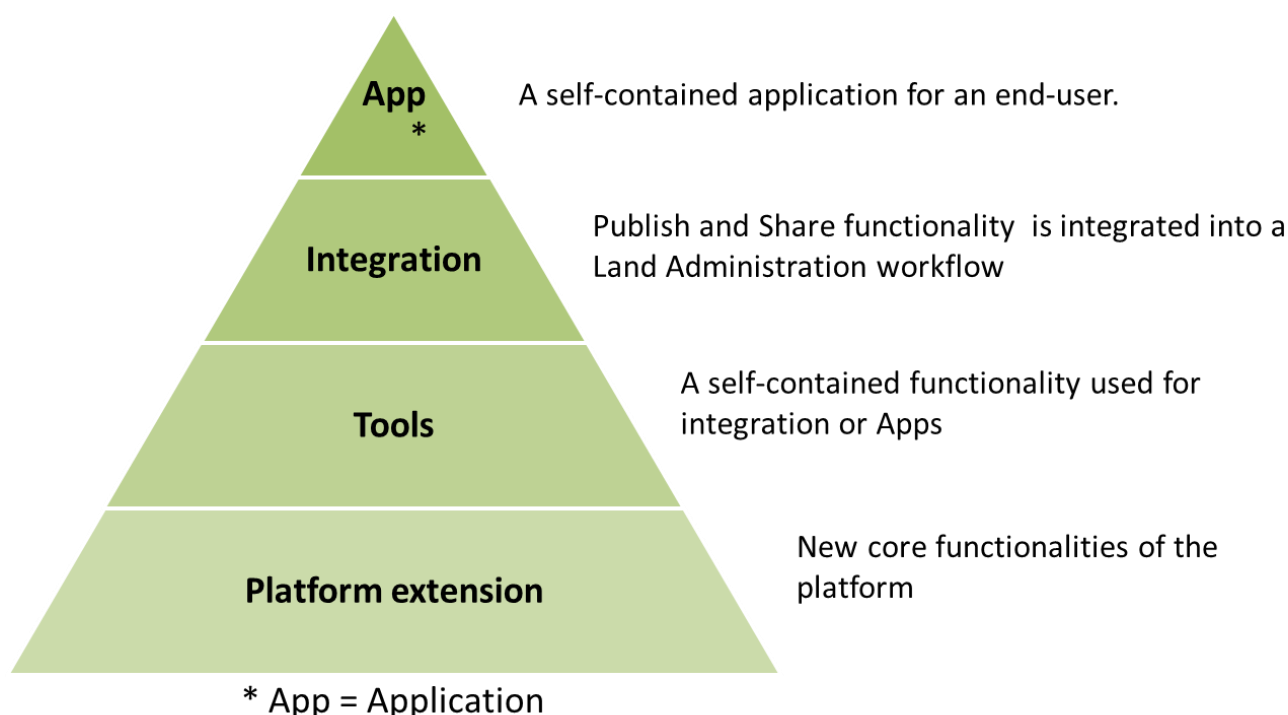


Figure 6: Overview of the usage models of Publish and Share

Application: An application targets an end-user. Typically an application is developed as a single page web application or mobile app and relies on the API provided by the platform. Other types of applications are possible; e.g. a QGIS-plugin or an ArcGIS application that interacts with Publish and Share. An application typically implements a domain or user specific workflow.

Integration: An existing Land Administration System integrates Publish and Share functionality into its own workflows. This integration is done on the programmatic API level. An example for this development model is the use of the boundary delineation methods from WP5 in a parcel based cadastral application to improve parcel determination workflows. An integration into an existing land administration system highly depends on the context of a country specific land administration workflow.

Tools: A tool implements self-contained functionality. Typically these are CPU and storage intensive tasks with little or no interactive usage. Tools make use of the API and data models specified by the platform for input/output and to organize information, but otherwise operate in an isolated fashion.

Platform extension: A platform extension implements functionalities at the level of the platform API. A typical platform extension could be a new API method that provides data from Publish and Share in specific way for a land administration system.

Support from Publish and Share for the application model

Complex calculation in a Run Time Environment – Client applications typically run on end-user devices such as desktops or mobile devices such as tablets. These are typically not suited to handle the workloads of geospatial applications, which can involve demanding processing requirements. Such tasks can take several hours to complete. In such situations, the heavy processing tasks can be offloaded to the Publish and Share platform which runs the resource demanding workload in a cloud based environment.

Storage – The same argument regarding the limited computing capacity of a client device can be extended to its storage as well. A typical aerial image of a target area can range from several hundred megabytes to gigabytes in size. Publish and Share offers centralized storage for the input spatial sources such as imagery, sketch maps and others. The cloud based storage also allows for the dissemination of processed data to a wider audience. The only requirement is a working internet connection with a suitable bandwidth.

API – The Public API provides the necessary scaffolding in order to construct an application in the domain of land administration systems. API endpoints provide the means for adding structures, data and metadata to the system following LADM concepts, along with necessary abilities to tag them, start computationally heavy processes and log them. Ample documentation is provided to enable application developers to look up the necessary parameters and their datatypes to perform requests and view expected responses from the platform.

Dissemination – One of the goals of the its4land project is to provide capabilities for the dissemination of (processed) data. The centralized storage makes this possible by making both spatial sources and processed data accessible via the API as Content Items. Content Items are accessible (downloadable) via URLs. In addition to accessing stored items, the platform allows applications to register data sources such as aerial imagery via tools and use OGC standards such as WMS to offer them as a service.

Structure (Projects) - A Project holds contextual information and related metadata supplied by the user. Information in the project structure of Publish and Share includes its name and description, spatial boundary, spatial sources, models created for tools and related tags. This information is wrapped in a GeoJSON structure allowing the spatial extents to be visualized directly by third party application which supports this format. Applications using the API, however, are free to use only the required information and define their own native structure for storing project information.

Support from Publish and Share for the integration model

In addition to the features supported for applications, as described in previous section, Publish and Share offers a number of benefits for users wanting to integrate it in into their existing land administration systems.

LADM describes a number of conceptual ideas for an ideal land administration system. A lot of these concepts tend to be highly abstract and a concrete implementation is needed for them to be integrated into an information system. The API provided by Publish and Share allows for a number of these concepts to be implemented. A few examples are:

- *Spatial sources* - LADM describes any source of Spatial Information as a spatial source. These can include text, imagery, shapefiles, geospatial databases, etc. It is also possible that a parcel of land can have multiple spatial sources. To support the abstract notion of a spatial source, Publish and Share makes use of the idea of content items. A content item is a digital representation of some data which can be stored in the form of a binary file. The *ContentItems* endpoint allows the user to upload, save metadata and download a file, without itself being concerned with the type and format of data. The *SpatialSource* endpoint, in contrast, allows the user to refer to a content item and give the content item a name, description, type and tag it. The higher level object encoded by a *SpatialSource* provides meaning to the raw data of a *ContentItem*.
- *Admin sources* – Similar to how a spatial source captures the spatial part of land registration, an admin source allows for handling its legal aspect. Admin sources similarly can be used in conjunction with a content item to provide meaning to the raw data.
- *Spatial Units* – In the integration model, Publish and Share mostly acts as a provider of information for a land administration system. Publish and Share implements and manages the process of transferring raw geospatial data to spatial units. The land administration system integrates the spatial units from Publish and Share via the Public

API into its own workflow. The spatial units can either be copied into the cadastral database of the land administration system or used there as a spatial reference for the registration of land tenures. The other option is that the spatial unit remains in Publish and Share. The land administration system then registers a land tenure based on the unique identification of a spatial unit in Publish and Share. In this case, Publish and Share acts as the cadastral database for the land administration system. The land administration system itself is only responsible for the legal registration.

An implementation of a LAS will be tailored to local needs and will cherry-pick any needed concepts from LADM and adapt it as required. Likewise, integrators need not use the entire API and are free to choose only those parts of Publish and Share which they want. The advantage of this model is that it reduces the burden on developers to implement individual concepts.

Support from Publish and Share for the tools model

Scalable Runtime Environment – The tools model of Publish and Share offers the ability to run computationally demanding tasks separately and scale them as needed. For this purpose, the tools are packaged in containers using Docker (see deliverable D6.1 [4] and D6.2 [5]). Packaging and running tools in this manner has several advantages:

- Scalability – Many instances of the tool can be spawned at the same time, each with a different set of parameters. It is also possible to run instances of different tools simultaneously.
- Freedom of Implementation – The choice of packaging tools in containers means that tool writers need not be restricted to a limited choice of programming languages. Every container has an *entry point* which determines what gets executed when a container starts running. It is the responsibility of tool writers to specify entry points and their needed arguments and to containerize tools to provide results that can be used in platform workflows.
- Isolation and Security – A container is isolated from the host machine and this ensures that the host machine resources are not accessed by unauthorized means. Shared resources that the platform provides are only accessible via the Public API.

Process management – A tool in Publish and Share is started via a *Process*. Each process has an associated tool and the tool's entry point, along with any necessary parameters passed to the tool. Once started, a process joins a queue and is run when there is sufficient runtime

capacity. Processes are also associated with a project and can be logged at different levels. The core of process management happens in a separate internal-only accessible API and is not exposed to the end user.

Tools can run either in *batch mode* or *interactive mode*. In batch mode, the API expects a number of arguments that are passed to the tool. These include the input data that are accessible in the tool's runtime environment and the parameters and their values to run the tool. Batch mode is ideal for computationally demanding tasks with long running times which do not require user prompts to proceed. When a batch task is finished, it can update the process logs or use other means such as sending an email to notify the user that a task is finished. With interactive mode, the tool provides a service that is accessible via a network port on the host machine. An example of this would be a tool providing a GUI via a web server that is accessible via a browser. It is the tool owner's responsibility to decide upon the runtime mode and structure the tool such that it can be packaged and run in a container and handling the constraints that containerization implies.

While tools are standalone programs and functionally independent units, Publish and Share provides the necessary context to use them in land administration systems. For example, a tool can work with spatial data such as aerial imagery and demarcate features, producing BoundaryFaceStrings that correspond to a SpatialUnit in an LADM. The tool can use the LADM specific concepts provided by the Publish and Share API to designate its output as a SpatialUnit. This makes the output readily usable in the land administration system.

Support from Publish and Share for the platform extension model

Publish and Share intentionally follows open and widely used standards for:

API Specification – The Swagger (aka OpenAPI) [8] standard is used for the specification of the REST API. The API endpoints, parameters, request and return datatypes and HTTP status codes are fully defined in the specification.

Data Serialization – Currently, the only supported format for data serialization is JSON [9]. The format is widely supported in multiple programming languages and frameworks.

Users wishing to extend the platform API to meet their own specialized needs can add to the specification by including more endpoints or modifying and extending the parameters for existing ones. As an alternative to JSON, an extension can include support for other widely used formats for data serialization such as XML [10].

The reference implementation of the server is carried out on the Node.js [11] platform, but it is also possible to implement it using other languages and frameworks.

Amazon Web Services' S3 service [12] is used for storage in the reference implementation for the its4land project. The reason for this being that it is a tried and tested service with a well-documented API, reliability and availability in multiple geographic regions. However, it is possible to switch backend storage providers to other vendors or even to store data on the customer's onsite premises for reasons of redundancy, privacy or security.

We also use an online hosted version of the PostgreSQL database on Amazon's RDS. The database schema followed for Publish and Share is easy to extend. As opposed to using foreign keys to capture relations between tables, which lead to cascading dependencies, we opt to keep track of the relationships in a separate *item2item* table. The structure and the maintenance of relationships are fully controlled by the implementation of endpoints in the Public API. This allows customer defined relationships to reuse the same general database model. E.g. the relationship between Projects and spatial sources in the project is tracked by adding an entry to this table via their respective primary keys.

Characteristics of the models and suggestions which model should be used in which situation

In the previous section, the different usage models of Publish and Share were presented. The four models can be divided into two groups. Application and integration on the one hand and tools and platform extensions on the other. The first two models can be used to implement some land administration related workflows. The other two can be used to implement functional extensions of the Publish and Share platform.

Application vs. integration

Both models are in general used to implement land administration workflows. The main difference between both models is the relationship to a land administration system. In the case of the application model, the coupling is rather weak. Weak in this case means, that the application on Publish and Share and the land administration system can operate completely independent from each other. The connection is mainly realised via the OGC services of the data dissemination interface. The direction of the dataflow is normally from the Publish and Share application to the land administration system. From the perspective of the land administration system, Publish and Share is a service provider. All the processing that is done by the application on Publish and Share is out of the scope and the control of the land administration system. In this model, the land administration system is the consuming part.

A typical workflow implemented following the application model is not under the control of a land administration system. Reasons to develop an application in this way are:

- The workflow is independent of a specific land administration installation. Independent means, that the workflow produces general results without a dependency on any specific land management policy and the workflow does not require any information or data from the land administration system itself. An example is the boundary delineation tool from *Automate It*. This application produces context free boundary face strings. The context is given by the land administration system as part of the usage.
- The application model allows business models that provide a workflow for several customers. The cloud platform of Publish and Share allows worldwide distribution, access and scaling.
- A typical implementation of a workflow in the application model is a mobile app that can be used on smart end-user devices.

The application model is suitable for a vertical workflow approach. The implementation can operate stand alone, without a land administration system, and all data is managed on the Publish and Share platform. The implementation has its own user interface.

The integration model requires a land administration system. The land administration system is the primary system and controls the workflow and provides the user interface. The interaction between the implementation and the land administration system is much closer. This kind of usage model is used when:

- The land administration system should be extended by specific functionality. An example is *Fly and Create*, that extends a land administration system with the functionality to produce and use UAV based orthoimages.
- The integration model allows an integrated workflow. The user can continue using the familiar user interface of the land administration system. The added functionality from Publish and Share is completely hidden.
- The integration model supports business models of integrator companies or independent software vendors who want to extend the functionality of their own product without developing the extension from scratch.
- The integration can be done on nearly all operating systems and development environments since the communication between the land administration system and Publish and Share is done by open and interoperable standards like REST and OGC web services.

The integration model is suitable for horizontal integration. The implementation directly integrates into the context of a large land administration system that implements a specific land management policy.

Tools vs. platform extension model

Both models can be used to extend the Publish and Share platform with functionality independently of a specific workflow. The main difference between both models is the level of complexity that can be implemented.

The tool model allows the implementations of algorithms with high computational and storage requirements. A tool is self-contained software that runs inside the runtime environment of Publish and Share. It can make use of the Public API and the underlying storage structures of Publish and Share. The LADM concepts available in the Public API provides the land administration context to a tool and allows the interaction of a tool with a workflow implemented by the application or integration usage model of Publish and Share. A tool can only make use of those LADM concepts provided by the Public API. To extend Publish and Share, other LADM concepts described in **Error! Reference source not found.** can be implemented by the platform extension model. Both usage models benefit from the implicit advantages of the cloud-based implementation like scalability, flexibility and the world-wide deployment via standard internet protocols. A tool can be implemented in nearly any software development environment. Most components of the existing its4land tools are developed using Python. A platform extension requires JavaScript as a development environment, since the Public API is implemented and hosted on Node.js.

The tool usage model is suitable in situations where:

- A complex algorithm in combination with high computational requirements should be implemented. This usage model is used in its4land to implement all the tools developed in the work packages 4 and 5.
- The tool model can be used to utilize an already existing 3rd party software for Publish and Share. The utilization can be done for the binary package or on the source code level. Publish and Share provides a wrapper that can be modified and adopted to 3rd party software so that the 3rd party software fits into the Publish and Share structures. This 3rd party software can be used like any tools developed in its4land. This method was used to utilize OpenDroneMap [13] as part of the tool developed in the work package *Fly and Create*.
- The tool usage model supports business models for independent software vendors (open source or proprietary) who want to provide specific functionality and distribute it with applications or land administration systems that operate on Publish and Share.

The tool model is suitable when a sophisticated algorithm should be published to the land administration sector. The developer can fully concentrate on the implementation of the algorithm. The connection to the land administration context is achieved by the Publish and Share platform.

The platform extension model extends the Public API of Publish and Share by new REST endpoints. This could be a LADM concept, e.g. a set of endpoints to **LA_BAUnits**. Another option could be to develop endpoints that publish already existing data in Publish and Share in a different way, e.g. an endpoint that combines SpatialUnit and AdminSources to a single response.

The platform extension is used when:

- Additional LADM concepts are required.
- The platform extension model can be used, when a land administration system requires information in a specific form. This would simplify the integration usage model. In the current implementation, the REST endpoints produce JSON as a response. The platform extension model can be used to change the response to XML.
- The platform extension model must be used if a synchronous communication between a client and the server implementation is required.
- The platform extension model supports business models for integrators, independent software vendors or platform operators who want to provide extra functionality.

The platform extension usage model is suitable in situations in which Public API has to serve specific workflow demands, like additional LADM concepts which were not necessary for the its4land project.

Suggested reading

- [1] ISO/FDIS 19152, "Geographic information - Land Administration Domain Model (LADM)," vol. 2012, pp. 1–118, 2012.
- [2] S. Enemark, K. C. Bell, C. Lemmen, and R. McLaren, "Fit-For-Purpose Land Administration," 2014.
- [3] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, Irvine, 2000.
- [4] C. Timm, M. I. Humayun, S. Walter, R. Borchert, S. Crommelinck, and C. Stöcker, "Technical report and software prototype of the mobile image processing system. Its4land Deliverable 6.1," Enschede, 2018.
- [5] C. Timm, M. I. Humayun, S. Walter, M. Chipofya, J. Sahib, and K. Kundert, "Technical report and software prototype of the mobile qualitative data processing system. Its4land Delivery 6.2," Enschede, 2018.
- [6] Network Working Group, "RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1." 1999.
- [7] Internet Engineering Task Force (IETF), "RFC 7946: The GeoJSON Format." 2016.
- [8] SmartBear Software, "OpenAPI Specification Version 2.0," 2014. [Online]. Available: <https://swagger.io/specification/v2/>.
- [9] Internet Engineering Task Force, "RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format." 2017.
- [10] World Wide Web Consortium (W3C), "Extensible Markup Language (XML) 1.1 (Second Edition)." 2006.
- [11] Node.js Foundation, "node.js," 2018. [Online]. Available: <https://nodejs.org/en/>.
- [12] "Amazon Web Services S3 Homepage," 2018. [Online]. Available: https://aws.amazon.com/s3/?nc1=h_ls.
- [13] "OpenDroneMap." [Online]. Available: <https://www.opendronemap.org/>.