

## Smart Integration of Energy Storages in Local Multi Energy Systems for maximising the Share of Renewables in Europe's Energy Mix

Grant Agreement No. 730936  
Start Date: 01/12/2016 – Duration 36 Months

### Description of D6.3, D6.4 – Services for SmILES Data Management

<b>Work Package No./ Title</b>	WP 6 Shared Data and Information Platform
<b>Task No/title</b>	T6.3: Implementation of content services
<b>Deliverable Number</b>	D6.3, D6.4
<b>Delivery Date</b>	M30 (shifted to M32)
<b>Date of completion</b>	28/10/2019
<b>Document Type</b>	OTHER
<b>Dissemination Level</b>	PU - Public
<b>Issued by</b>	KIT
<b>Author (s)</b>	Clemens Döpmeier
<b>Contributors</b>	Shadi Shahoud, Isabelle J. Südmeyer

### Summary

This document gives a short overview on the work performed in Task 6.3 for implementing the content services for the SDIP platform. SDIP allows using either the Liferay internal content and asset management service or the external Generic Data Services (GDS, [5]) of the Energy Lab 2.0 for data and information storage within SDIP. Therefore, SDIP can be either used independent from other platforms or easily integrated into bigger engineering and research environments using the GDS integration as an example.

The implementation was done in several phases by first integrating the content, asset and schema GDS service into Liferay by implementing a CMIS (Content Management Interoperability Services, [4]) interface for the GDS master data and schema service and wrapping the REST API's of the GDS services into Liferay provided services for secure and easy access from UI frontend components. In a second step, the time series service was integrated into Liferay by implementing a Liferay service wrapper for its REST API. The availability of the CMIS and GDS service wrapper implementations and their integration into the SmILES Liferay server mark the delivery of Deliverables D6.3 "Data services for time series data management and background data" and D6.4 "Data services for content and asset management of system configurations, models, etc." of type Other (e.g. software implementation / package). The integration was done using a first prototype version at the end of January 2019. In August 2019, an updated version was installed which is now used by the production system.

### Approval

	<b>First Author</b>	<b>1<sup>st</sup> Revision by</b>	<b>2<sup>nd</sup> Revision</b>	<b>Final Revision by WP leader</b>	<b>Project Coordinator</b>
<b>Date</b>	Clemens Döpmeier			Clemens Döpmeier	Isabelle J. Südmeyer
	30/07/2019			28/10/2019	04/11/2019



# 1 Introduction

## 1.1 Purpose of the document

This document summarizes the work conducted in task T6.3 of work package WP6 of the SmILES project for setting up the data and content services needed by the Shared Data and Information Platform (SDIP) for storing SmILES related data and information. SDIP allows using either the Liferay internal content and asset management service or the external Generic Data Services (GDS) of the Energy Lab 2.0 for data and information storage within SDIP. Therefore, SDIP can be either used independent from other platforms or easily integrated into bigger engineering and research environments, such as the Energy Lab 2.0 software platform, using the GDS integration as an example.

The Shared Data and Information Platform implemented in WP6 serves as a common entry point for accessing all project results and the data collected within the project. The platform integrates advanced visualization tools and search technology to explore SmILES methodology, data and project results easily through the web interface of the platform. This will be of great value for users looking for information about performing complex energy system modelling and analysis, or are in the need of data and sub-models for creating own models.

The shared data and information platform and the tools implemented for the platform will provide a good foundation for the dissemination activities and for further projects in which the data and information base contained in the platform can be enhanced to cover other smart energy technologies beside storage and renewables as well. In this respect the platform can be extended towards a central information hub on smart energy technology concepts, data and best practices.

The implementation of the data and content services in Task 6.3 was done in several phases by first integrating the content, asset and schema service into Liferay by implementing a CMIS (Content Management Interoperability Services, [5]) interface for the GDS master data and schema service and wrapping the REST API's of the GDS services into Liferay provided internal services for secure and easy access from UI frontend components. In a second step, the time series service was integrated into Liferay by implementing a Liferay service wrapper for its REST API. The availability of the CMIS and GDS service wrapper implementations and their integration into the SmILES Liferay server mark the delivery of Deliverables D6.3 "Data services for time series data management and background data" and D6.4 "Data services for content and asset management of system configurations, models, etc." of type Other (e.g. software implementation / package). The integration was done using a first prototype version at the end of January 2019. In August 2019, an updated version was installed which is now used by the production system.

## 1.2 Scope of the document

The work description of Task 6.3 of the SmILES project defines the following subtasks:

- Implement / setup data services for storing and accessing time series data and other background data based on available functionality in the Energy Lab 2.0 (D6.3).
- Implement / setup an asset management system for storing and accessing structured documents (scenario descriptions, best practice documents, etc.) and binary assets such as simulation models in the shared data and information platform (D6.4).

This document describes the overall concepts for the implementation of the data and content services and summarizes the implementation work done for both sub-tasks.

### **1.3 Structure of the document**

In Section 2, the overall architecture of the Shared Data and Information Platform (SDIP) is explained.

Section 3 describes the underlying concepts for the implementation of the content services and gives a short overview on key aspects of the implementation.

The document concludes with a short summary and outlook in Section 4.

## 2 Overall Architecture of the Shared Data and Information Platform (SDIP)

The Shared Data and Information Platform is fully integrated with the SmILES project web site and is implemented by using a Liferay web portal server as technical platform for delivering the web pages of the user interface. As shown in Figure 1, the SmILES web portal (e.g. the Liferay server and SDIP-UI-application layer) sits on top of the Energy Lab 2.0 ICT infrastructure as runtime infrastructure.

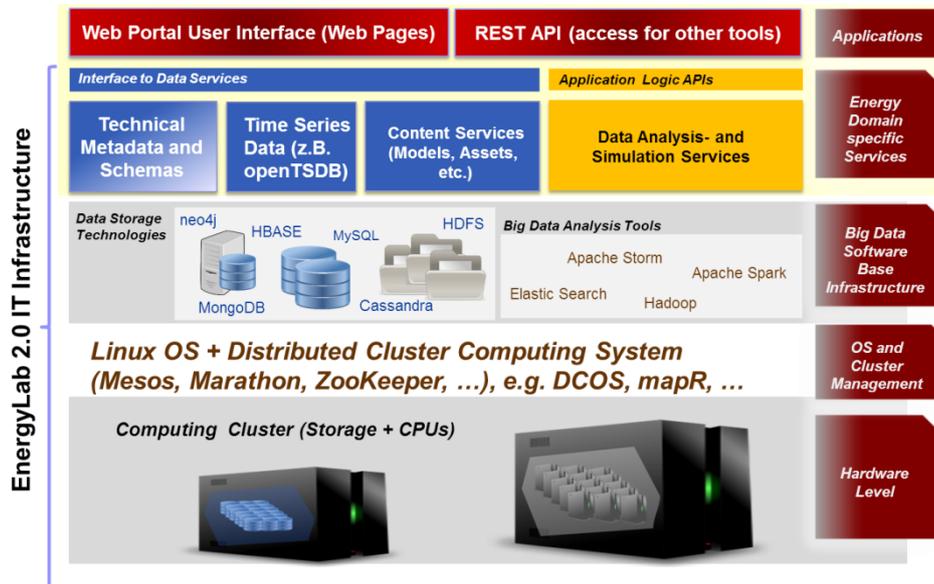


Figure 1: ICT Infrastructure for hosting the SDIP Platform

The Energy Lab 2.0 runtime infrastructure consists of several computing clusters for providing computing and storage capabilities for the software applications of the Energy Lab 2.0 at the KIT Campus North which is managed by the Institute of Automation and Applied Computer Science (IAI) at KIT.

One large storage cluster is used for Big Data Management and Big Data Analytics applications: e.g. it hosts several NoSQL database applications and a distributed HDFS filesystem (e.g. the distributed file system HDFS, MongoDB as document-oriented database, Hbase as column store, the Elasticsearch document-oriented database and index store, a MySQL cluster as relational database cluster, and a neo4J graph database as graph store) for large scale data management and several data analytics frameworks (e.g. Apache Spark, Storm, Hadoop and Elasticsearch) for building and running Big Data analytics applications. Another storage cluster, the Ceph-Cluster provides a large scale distributed filesystem and object (asset) storage by using the Ceph distributed filesystem software. Ceph provides low level API's to operating systems, i.e. a Ceph cluster can be used by virtualized computing nodes on a cluster as low level block devices for providing virtual storage disks for filesystems. It also supports standard network filesystem interfaces for Windows or Linux operating environments (e.g. NFS or SMB). Additionally, it supports cloud storage compatible interfaces (e.g. the Amazon S3 API) for modern Internet applications for storing structured objects and larger binary files in e.g. an own Cloud storage.

Third, the Energy Lab ICT infrastructure contains Linux and Windows computing clusters for server applications. The Linux cluster is equipped with the public domain Openstack cloud operating environment which allows running cloud infrastructure compatible applications on premise on own computing hardware. Especially, the Openstack cluster instruments Docker and Kubernetes for providing software container based automation of server applications (e.g. services, web applications, etc.).

The Openstack cluster is used as runtime environment for the service and application layer of the Energy Lab 2.0 software infrastructures (see the top 2 layers in Figure 1). At the one hand it hosts the Generic Data Services (GDS) for storing Energy related data and information files in the underlying database infrastructure layer. On the other hand, it provides dedicated services for running complex scientific calculations (e.g. forecasting or data analytics workflows) and/or complex simulation workflows (e.g. simulations, co-simulations) as distributed workflows on the underlying container infrastructure with a distributed scientific workflow engine called PROOF (PROcess Operation

Framework). Furthermore, the Docker and Kubernetes infrastructure is used for automating the management of running other applications, such as web-applications, e.g. the SmILES web portal.

Both, the GDS services and PROOF, are Microservice-oriented software applications developed by IAI as server-side backends for developing web-based engineering applications for complex energy system related research on top of these services.

The SmILES web portal and SDIP platform fully instruments the Energy 2.0 runtime infrastructure. The Liferay portal server which is the central web based user interface (see the Web-UI- and Application-layer in Figure 2) to the SmILES web site and SDIP platform runs as automated application on the Openstack cluster in the Docker / Kubernetes environment.

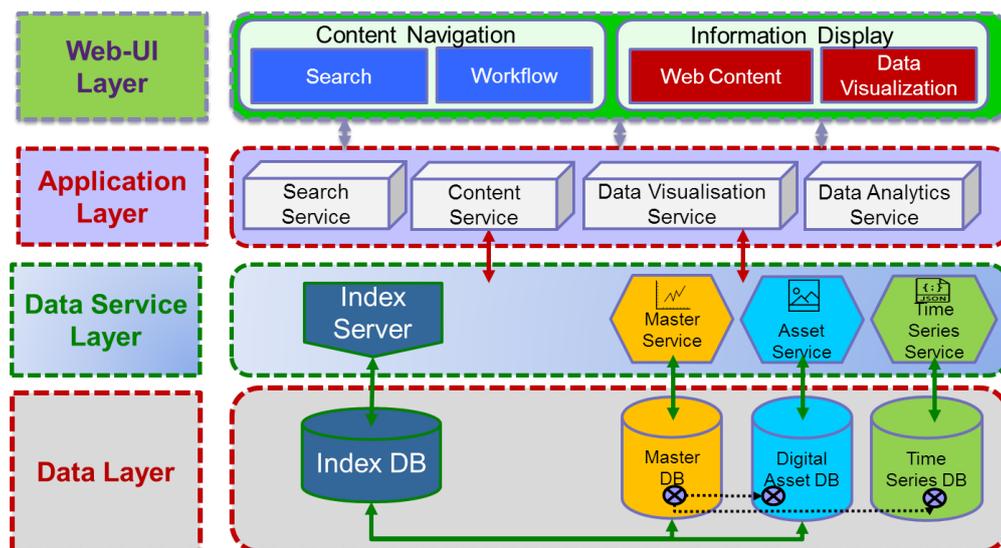


Figure 2: SmILES Portal and SDIP Architecture

An “Application Layer” situated in the application server of the Liferay server provides interfaces to the GDS services of the Energy Lab 2.0 infrastructure (Data Service Layer and Data Layer) for easily accessing and storing data and information in the GDS services. The “Application Layer” implements proxy APIs provided by Liferay for easing a secure and authorized access to the GDS services from web user components and JavaScript applications embedded into the Liferay provided web pages. The Application Layer implemented in the Liferay application server also implements adapter, which provide access to GDS content via standard Liferay APIs and internal interfaces for accessing web content and assets (The content and asset management APIs of Liferay). This allows web authors using the SmILES Liferay portal server to create and access GDS content and assets as if they were internally stored within the Liferay server storage layer. Therefore, the storage ecosystem of the SmILES Liferay server provides users the opportunity to store and access information content either within the Liferay internal storage layer or within the GDS services using the same authoring and access environment.

Liferay itself provides very good measures for setting up structured content elements and adding arbitrary metadata to assets uploaded to the server via its content management interfaces (e.g. the standard authoring environment of Liferay). Those are used by SDIP to provide structured content and metadata forms to authors for uploading assets (e.g. binary data) and creating content elements according to the structural content specifications for SDIP contents as defined in Deliverable D6.2. The underlying implemented “Application Layer” transparently stores those content and asset elements either in Liferay or the GDS services of the Energy Lab 2.0 infrastructure as introduced about.

### 3 Data and Content Services for SDIP

Within task T6.3 the data layer for storing the information which should be provided and managed by the Shared Data and Information Platform were integrated into the Liferay platform. A first analysis showed that information related to SmILES methodology can either be represented as semi-structured document, as text or binary file which should be additionally described by certain metadata attributes, or as data files providing e.g. time series data (see also Deliverable 6.2, [2]). Content of the first type “semi-structured” are e.g. informational descriptions of SmILES methodology (tasks) and their inputs and outputs, which consist mainly of textual description parts, but also contain some structured fields for e.g. describing configuration parameters or the like, and can additionally contain references to one or more asset descriptions (e.g. describing binary files). In a web based information system such descriptions can be implemented as a structure of linked semi-structured web content objects, which can then be used for generating integrated web pages describing these kinds of information elements: e.g. pages describing system configurations and use case descriptions, specifications of control functions, modelling components, system models or test case specifications.

Assets, such as word documents or PDF files, media files, e.g. images or videos, model files for simulators, such as Matlab or Modelica variants, have as one ingredient the binary file but also need some additional structured metadata, e.g. information about the author of the document, the title, a short abstract, for implementing adequate search and access interfaces to such assets. Finally, data files containing time series data can be stored and managed as assets or – in case of large time series data – better be stored in special time series data services for easy visualization and access functionality.

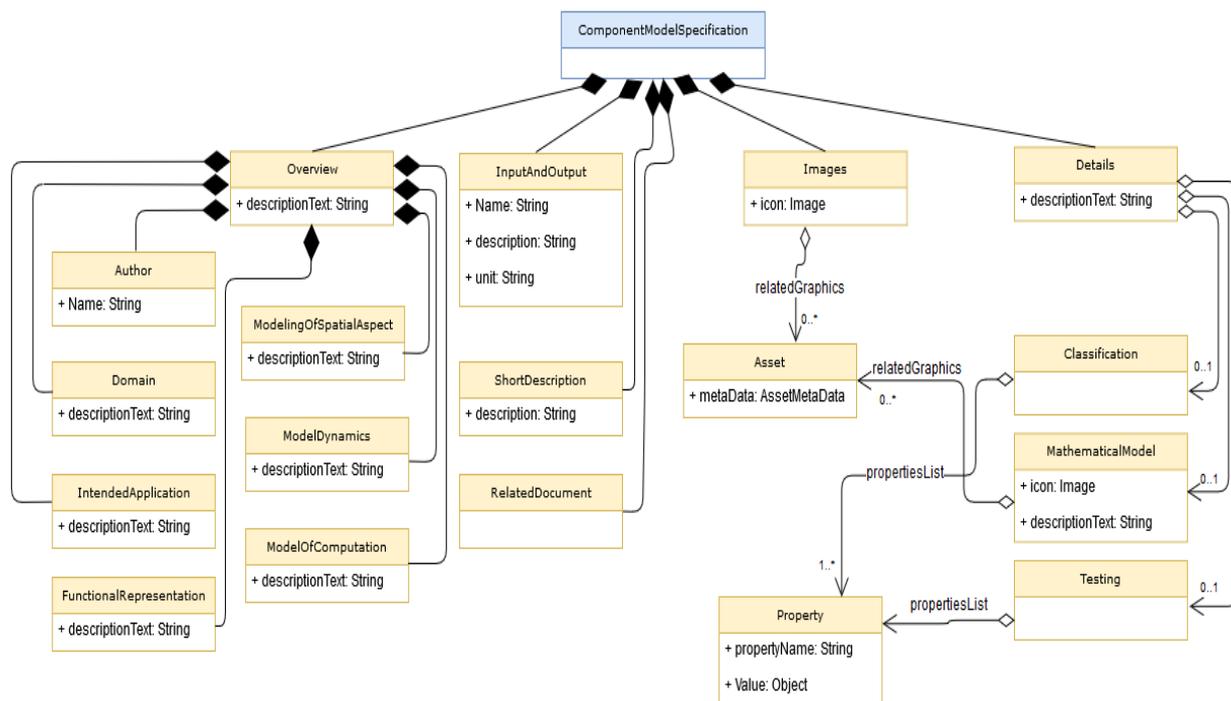


Figure 3: Structure of a Component Model Description

Let’s demonstrate this structure of SDIP elements using a “Component Model Specification” as example (see Figure 3 or Deliverable D6.2 for more information about the structure of SDIP information elements). A “Component Model Description” has an “Overview” information part, which describes some key information about the model. It contains e.g. the fields Author, Domain, IntendedApplication, FunctionalRepresentation, ModelingOfSpatialAspect(s), ModelDynamics and ModelOfComputation, which are described as text attributes; but some of them are defined for the authors as enumeration of text labels (e.g. ModelOfComputation, see also screenshot below).

Additionally, each Component Model Specification describes the InputAndOutput parameters of the model, has a short summary description and can point to related documents (RelatedDocument), e.g. a PDF or Word file, with more information. The Model Specification can further point to one or more

image assets illustrating the model with different graphical representations. Each (image) asset has further metadata: e.g. information about the author, file type, an image tile used as capture and a description of the image explaining the graphics. A more detailed description can be given in further description parts Details, Classification, MathematicalModel and Testing. Those parts can also contain property lists which describe essential properties of the model by listing the property name, its type and a short description of its meaning for the model.

The following screenshot shows how the information in this semi-structured description is used by SDIP to generate a web page showing a Component Model Specification in the web browser.

**Model Overview**  
**Author / organization:** Zhichao Wu/KIT  
**Domain:** Electrical storage  
**Intended application:** Testing of process simulations  
**Modelling of spatial aspects:** Lumped (single device)  
**Model dynamics:** Dynamic  
**Model of computation:**  

- Time-continuous
- Discrete-event

**Functional representation:** Implicit

**Input and Output**  
**Input variables :** Real  $P_{batt}$ : Battery power setpoint, given by controller [kW]  
**Output variables:**  

- Real  $SOC_{batt}$ : State of Charge of the battery [%]
- Real  $U_{batt}$ : Output voltage of the battery [V]

**Related Documents**  
[D4-1 Definition of a minimal set of component models.pdf](#)

**Short Description**  
An equivalent circuit model of a battery, including a sensor that measures the power and voltage of the battery. The battery is modelled as an equivalent circuit with OCV lookup table and four impedances connected in series. Since one single battery as defined in the model is usually not enough to provide the required power, a pack typically consists of several identical batteries connected in series.

**Present use / development status**  
This model is a simplified prototype of the battery system for model validation and harmonization. The influence of SOH and temperature will be integrated at a later time into the model.

**Model Details**  
Classification  
Mathematical Model  
Testing

Figure 4: Generated SDIP web page showing a Component Model Specification

As can be seen from Figure 4, the model overview description is summarized in a content block on the top left of the page. Below you find the description of the input and output parameters of the model and links to related documents (e.g. document assets stored separately within SDIP). On the right side of the page, there is a short overview description of the model specification and below a slideshow of related graphics. If you click on a graphic in the slideshow, a larger version will be displayed in a Popup Window which also contains the image description.

Even more detailed descriptions are shown when the user opens one of the content areas of the accordion menu beneath “Model details”, which contain the description parts Classification, Mathematical Model and Testing respectively.

SDIP collects the information for different information objects – such as a “Component Model Specification” – from authors by providing structured forms where the author can fill in the semi-structured information into different typed fields according to structural schemas defined in Deliverable 6.2 and implemented in the content and asset management interface of Liferay. For textual information, for which there are predefined values (e.g. an enumeration), the form already pre-defines values from which the author can choose. Other textual fields allow free text entry as single line text or larger HTML formatted text blocks. Properties can be defined to have values of a certain type e.g. float, integer according to the element structure. Binary assets will be stored separately in Liferay in the so called asset management system. Therefore, different asset types with dedicated metadata structures were

defined in Liferay (e.g. Document, Image, Graphics, and Model). Assets can then be created by the authors using also structured forms gathering the metadata belonging to an asset. Additionally, there is an upload-button to upload the binary file and assign it to the asset. Note, that asset management systems also support versioning of assets, which is also the case for Liferay. SDIP uses the asset management interface of Liferay for storing binary files, such as documents, graphics, and model and data files together with according metadata separately from the semi-structured information element descriptions within the data management layer of SDIP. Form fields within the information element forms allow then to link assets to the semi-structured information elements. This allows easy reuse of links to e.g. graphical elements or documents between several SDIP information objects.

Liferay contains two abstractions and internal API-interfaces for accessing and managing semi-structured content and assets (e.g. web content and assets in Liferay terminology) within Liferay. These fit perfectly to these two types of SDIP content. Time series data can be either seen as special asset type and stored within the asset management system, or for really large time series data it's better to store them in a time series database for easy access and usage.

The Liferay abstract interface to the internal content and asset management system allows it to connect third-party software to Liferay as additional asset and/or content storages (e.g. asset management software, such as Alfresco, Documentum or even Microsoft Sharepoint can be easily integrated with Liferay as asset and/or content stores). After integration, Liferay authors can then create and access content and assets, which are either stored in Liferay itself or within the integrated third-party software. While for the SmILES demonstration showcase of SDIP it would be enough to use the Liferay internal data management, the use of SDIP in third-party environments, such as the Energy Lab 2.0 platform, which will also use SDIP, will demand an integration of the third data management with the SDIP data management within Liferay. To show how this can be achieved, the GDS services of the Energy Lab 2.0 were integrated with Liferay for use by SDIP.

The GDS services fit perfectly to SDIP, because they also foresee a data management which is splitted into an asset management service, the master data object service, where semi-structured web content can be considered as master data object, and the time series data service for handling (large-scale) time series data. Additionally, the GDS services contain the Schema service, which provides machine readable semantic descriptions of the structure of information and data objects within the other services, which can be used e.g. for generating forms for authoring data or supporting the access and display of semi-structured information elements.

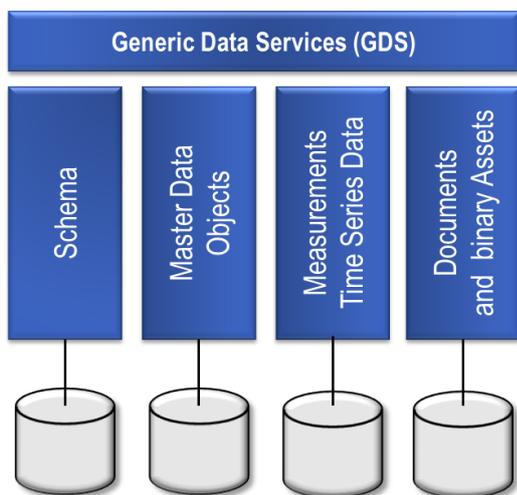


Figure 6: Structure of the Generic Data Services

To integrate the GDS services into Liferay (and therefore SDIP), a CMIS (Content Management Interoperability Services) service interface was implemented as frontend to the GDS services. CMIS let the services together look like a remote hierarchical storage for semi-structured content and asset objects which can then be mounted within Liferay as third-party content and asset store. CMIS is also used for integrations of other commercial asset and content management systems, such as Alfresco or Sharepoint.

Additionally, the REST based service API's of the GDS services were wrapped into Liferay services for easy and

secure access by web based UI components and JavaScript code within the web based user interface components used by Liferay portlets (see also Chapter 2). This allows the implementation of UI components, e.g. for visualization of time series data, which can directly access the data within the time series data service of the GDS without using the CMIS interface which is much more performant.

## 4 Summary and Outlook

This document gives a short overview on the work performed in Task 6.3 for implementing the content services for the SDIP platform. SDIP allows using either the Liferay internal content and asset management service or the external Generic Data Services (GDS) of the Energy Lab 2.0 for data and information storage within SDIP. Therefore, SDIP can be either used independent from other platforms or easily integrated into bigger engineering and research environments using the GDS integration as an example.

The implementation was done in several phases by first integrating the content, asset and schema service into Liferay by implementing a CMIS interface for the GDS master data and schema service and wrapping the REST API's of the GDS service into Liferay provided services for secure and easy access from UI frontend components, such as portlets, web components or JavaScript applications. In a second step, the time series service was integrated into Liferay by implementing a Liferay service wrapper for its REST API. The availability of the CMIS and GDS service wrapper implementations and their integration into the SmILES Liferay server mark the delivery of Deliverables D6.3 "Data services for time series data management and background data" and D6.4 "Data services for content and asset management of system configurations, models, etc." of type Other (e.g. software implementation / package). The integration was done using a first prototype version at the end of January 2019. In Juli 2019, an updated version was installed which is now used by the production system.

Finally, schemas for structured web content and asset objects were defined according to the specification in Deliverable 6.2 for iteratively adding content to the SDIP platform, as it becomes available.

The implementation can be considered as a first version. E.g. while integrating the GDS services only main CMIS functionality was implemented as needed for a first version of SDIP. In further steps after the SmILES project ends, the CMIS interface will successively be enhanced to cover more details. The GDS services are also actively in further development. When new versions will become available, the REST API wrappers for Liferay have to be adapted to the new version too. This work will be done in context of further developments of the Energy Lab energy research platform which will use SmILES as part of its web based user interface.

## Bibliography

- [1] Deliverable D6.1, "D6.1: First version of web portal which contains a 1st version of the Website," H2020 SmILES project, 2017.
- [2] Deliverable D6.2, " D6.2: Specifications of data formats and metadata schemes for the data and information management platform", H2020 SmILES project, 2019.
- [3] Liferay DXP7.1 Server Documentation Online, <https://help.liferay.com/hc/en-us/categories/360000872531>, visited last on 10/31/2019.
- [4] Content Management Interoperability Services (CMIS), Version 1.1 Standard Documents, available online at <http://docs.oasis-open.org/cmisis/CMIS/v1.1/CMIS-v1.1.html>, last visited at 10/31/2019.
- [5] Braun, E., Schlachter, T., Döpmeier, W. C. S., & Stucky, K. U. (2018, April). A Generic Microservice Architecture for Environmental Data Management. In Environmental Software Systems. Computer Science for Environmental Protection: 12th IFIP WG 5.11 International Symposium, ISESS 2017, Zadar, Croatia, May 10-12, 2017, Proceedings (Vol. 507, p. 383). Springer.