Thomas J. Lampoltshammer / Christian Sageder / Thomas Heistracher

The Openlaws Platform

An Open Architecture for Big Open Legal Data

Free access to legal information is imperative for evolving and supporting democracy. National and European initiatives such as RIS (Legal Information System of the Republic of Austria) and EUR-Lex form the first steps towards a sustainable and open environment for legal data. However, in the long run, community engagement and cooperation between legal fractions have to be established to foster rethinking and positively reshaping of jurisprudence in the 21st century. Therefore, the openlaws.eu project will provide a platform motivated by the requirements of the community through combination of open software, open data, and open innovation. This paper provides architectural and technical considerations and an outline of the initial architecture for the openlaws.eu platform.

EDITIONS WEBLAW

Contents

1   Introduction

[Rz 1] Free access to legal information is imperative for evolving and supporting democracy. National and European initiatives such as RIS (Legal Information System of the Republic of Austria) and EUR-Lex form the pivotal first steps towards a sustainable and open environment for legal information. However, in the long run, community engagement and cooperation between legal fractions have to be established to foster rethinking and positively reshaping of jurisprudence in the 21st century. The European project *openlaws.eu* therefore aims at fostering and extending this crucial engagement and cooperation by building on the three pillars of open data, open innovation, and open source software. The idea is not only to make access to legal information easier, but also to present a way to organize legal information according to individual needs and to share this information with other users. This forms a social layer that extends existing legal information systems starting from national databases up to the European level. *Openlaws.eu* will not replace these systems, but supports them to distribute their content to a larger user group in a faster and more intuitive way (Wass et al., 2013).

2   Open Innovation — A Path Towards A Sustainable Platform

[Rz 2] Innovation is the driving force for all successful businesses. If the process of development for new concepts and ideas is extended beyond the boundaries of a single organisation or company, we encompass the concept of *open innovation*. The European Commission also favours this concept as it is anticipated within the European Innovation System; within this system, the idea is to create innovative ecosystems in which all involved stakeholders interact and work together for the benefit of all. In this context, the *openlaws.eu* project aims at gathering outside knowledge from experts of the legal community and integrate it into the *openlaws.eu* Web platform. This «wisdom of the crowd» enables the creation of new links between people and legislation, an easier search through the body of legislation, as well as support for the handling of interesting legal items (mass-customisation). By this approach, the system is not only considering results from computer-driven analyses, but also from human expertise.

[Rz 3] The development and realisation of such an ambitious system's architecture is no a trivial task. Therefore, it is most important to base its continuous development and deployment on a solid basis. To achieve this task, the *openlaws.eu* project is embedded in a tangible cooperation network, involving all necessary stakeholders to guarantee sustainable results. Figure 1 describes the strategic development path pursued within the project.
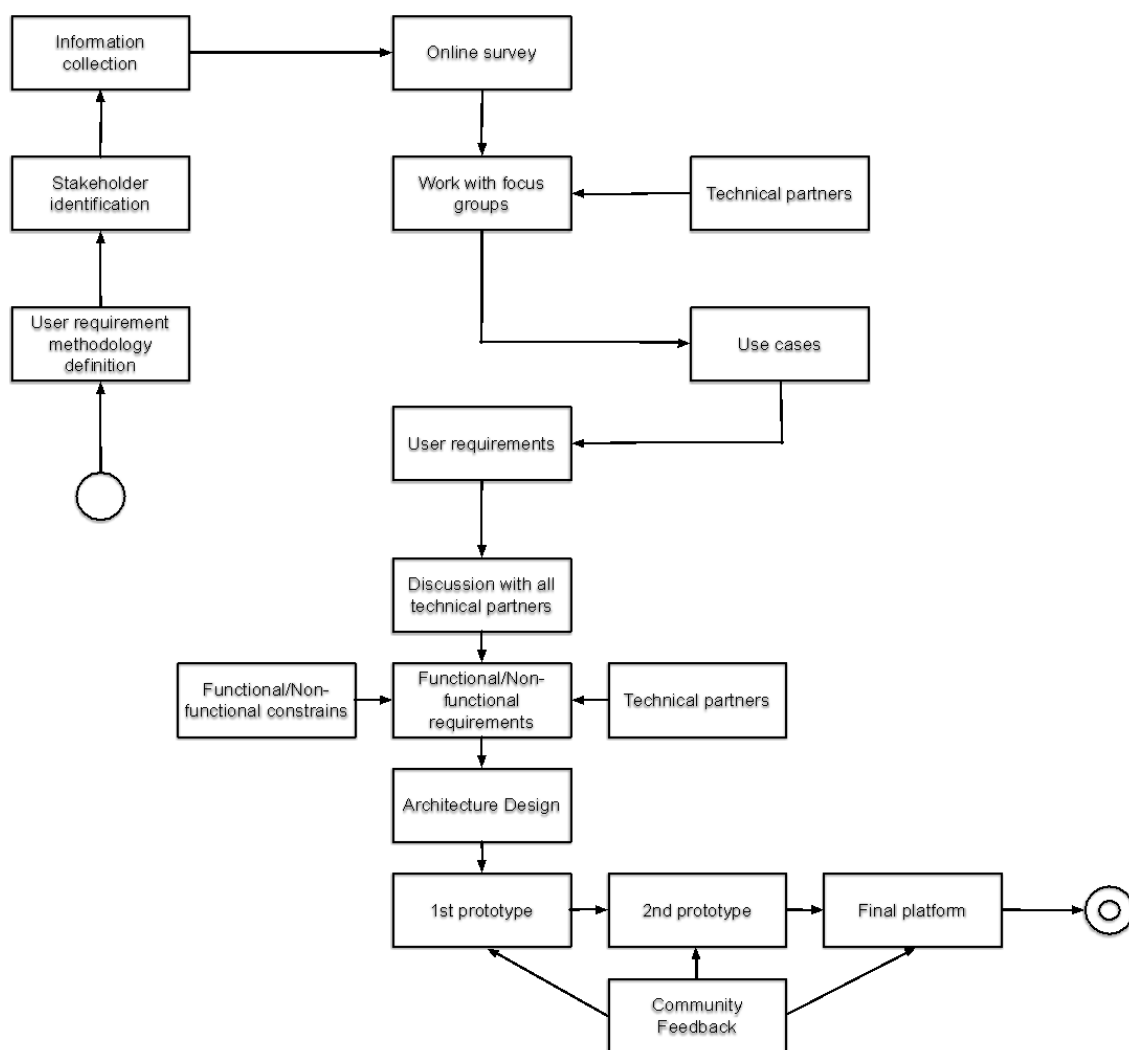
```
┌──────────────┐                    ┌──────────────┐
│ Information   │───────────────────▶│ Online survey│
│ collection    │                    └──────┬───────┘
└──────▲───────┘                            │
       │                                    ▼
┌──────────────┐                    ┌──────────────┐     ┌──────────────────┐
│ Stakeholder  │                    │ Work with    │◀────│ Technical partners│
│ identification│                   │ focus groups │     └──────────────────┘
└──────▲───────┘                    └──────┬───────┘
       │                                   │
┌──────────────┐                           │          ┌──────────────┐
│ User requirement│                        └─────────▶│ Use cases    │
│ methodology   │                                     └──────┬───────┘
│ definition    │                                            │
└──────▲───────┘                    ┌──────────────┐         │
       │                            │ User         │◀────────┘
      ◯                             │ requirements │
                                    └──────┬───────┘
                                           ▼
                                    ┌──────────────┐
                                    │ Discussion   │
                                    │ with all     │
                                    │ technical    │
                                    │ partners     │
                                    └──────┬───────┘
┌──────────────────┐     ┌──────────────┐         ┌──────────────────┐
│ Functional/Non-  │────▶│ Functional/  │◀────────│ Technical partners│
│ functional       │     │ Non-functional│        └──────────────────┘
│ constrains       │     │ requirements │
└──────────────────┘     └──────┬───────┘
                         ┌──────────────┐
                         │ Architecture │
                         │ Design       │
                         └──────┬───────┘
                                ▼
        ┌──────────┐    ┌──────────┐    ┌──────────────┐
        │ 1st      │───▶│ 2nd      │───▶│ Final platform│──▶ ◎
        │ prototype│    │ prototype│    └──────────────┘
        └──────────┘    └──────────┘
                   ┌──────────────┐
                   │ Community    │
                   │ Feedback     │
                   └──────────────┘
```

Figure 1: Strategy path to define architecture

[Rz 4] After the initial setup of the entire user requirement methodology, the stakeholders are identified and associated detail information is collected. Next, an online survey is conducted for identification of user needs from the legal community. These results are presented and extended by feedback from focus groups, in cooperation with all technical partners.

[Rz 5] Out of this pool, use cases are drafted, which present the basis for the first set of user requirements. These requirements are transformed into functional and non-functional requirements for the discussion and decision process with the technical partners.

[Rz 6] After a set of use cases for the first initial architecture is found, the actual design process is started, which leads eventually to the first prototypical implementation. In the second step, additional functionalities are added to the prototype until the final state for this project has been reached. During the entire development and implementation phase, the legal community is involved within the process. The feedback associated to the intermediate versions is evaluated and incorporated into the next succeeding version of the *openlaws.eu* platform.

3   Open Source Software — Enabling An Extensible Architecture

[Rz 7] The inherent structure of the architectural description is based on the ISO/IEC/IEEE 42010:2011 standard entitled «Systems and software engineering — Architecture description» (ISO/IEC/IEEE, 2011). This document contains actions regarding the analysis, creation, and sustainability aspects of software architectures. In addition, terminology, key aspects, as well as examples comprise this standard. Figure 2 depicts the overall concept of an architectural description.
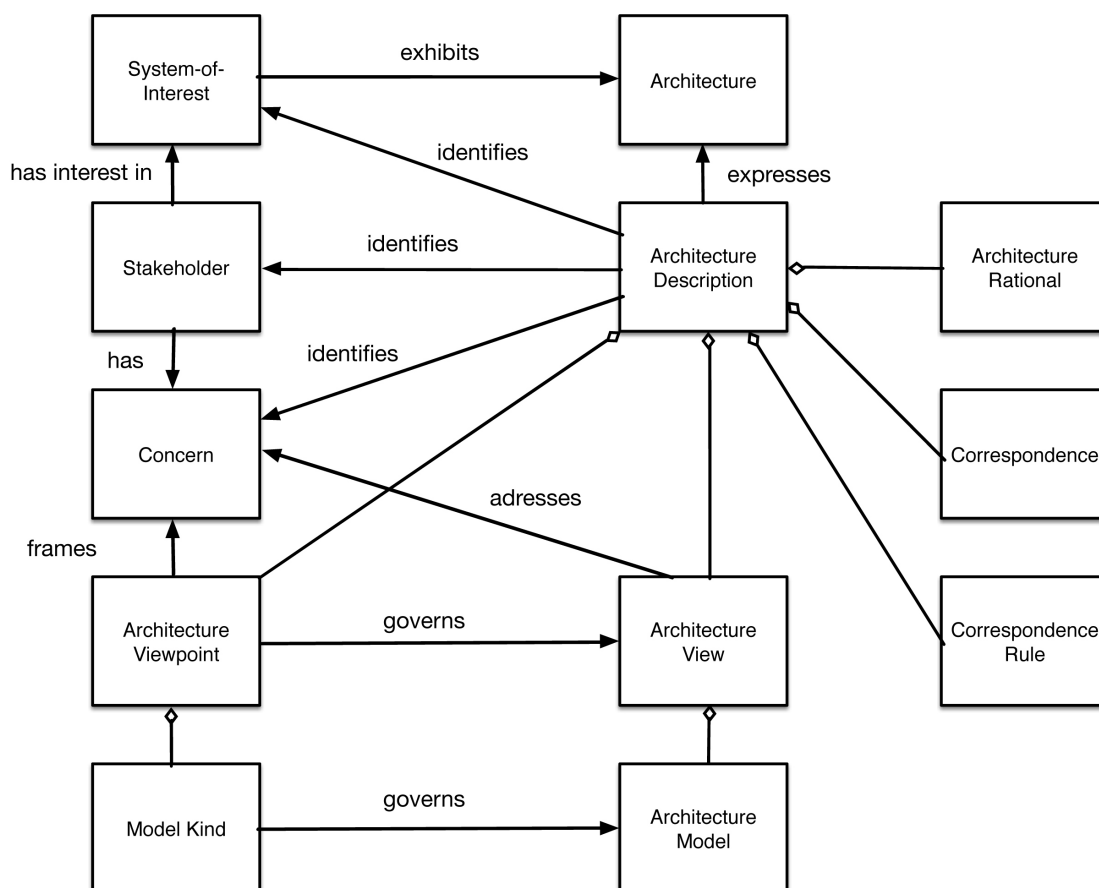
Figure 2: Concept of an architectural description, taken from (ISO/IEC/IEEE, 2011)

[Rz 8] The main difference between a system's architecture and its description is related to its respective meta level. While a system's architecture is an abstract concept of components, its description binds it to actual technology. This approach can be compared to the concept of a Model-driven Architecture (MDA) (LIDDLE, 2011). As viewpoints on the architecture strongly depend on project-specific aspects such as stakeholders, their requirements and so forth, the ISO/IEC/IEEE 42010:2011 standard does provide only a basic, non-specific description of these viewpoints.

3.1   Core Software Components of the openlaws.eu Platform

[Rz 9] The set of viewpoints being considered for the *openlaws.eu* project is based on the work of Rozanski & Woods (2011). For this paper, we will concentrate on the information view and the deployment view. Figure 3 depicts a combination of these viewpoints.

[Rz 10] For the web-based frontend of the platform, AngularJS (Angular, 2015) was chosen. This Model-View-ViewModel-based JavaScript Framework provides the necessary flexibility needed to access data and the associated meta-data from the platform in a fast, fluid, convenient way. In combination with the Bootstrap Framework (Bootstrap, 2015) for a responsive layout, the plat-form can be accessed over a plethora of different devices. In order to keep a low communication load profile, we employ a REST-based interface (Fielding, 2000) to exchange data. In addition, this approach provides better control over the access towards the platform and the data held.

[Rz 11] All interactions within and towards the backend system are handled with the Spring Security Framework (Spring, 2015). This provides the necessary basis for authentication and au-thorisation within the platform. To import open legal data from various legal data sources (e.g. Eur-Lex, RIS), custom importer services are developed for each external source. After an initial bulk import as a data basis, incremental updates are executed once the original data source has changed or was updated with new data items. After the data was acquired and its consistency check was positive, it is mapped to the internal data model (see Fig. 4) and imported into the platform's data repository. To handle the scheduling of all periodic process within the system, the scheduling software Quarz (Quarz, 2015) was chosen.

[Rz 12] The data repository of the platform is based on the Neo4j database (Neo4, 2015). This graph-based database does not only support the development team in transferring the project's use cases directly towards the data model, but also features highly useful approaches to imple-ment a recommendation engine based on the data model and the data inside the repository. The search functionalities of the system are covered by the Elasticsearch ELK Stack (Eleasticsearch, 2015). This comprehensive search optimization framework does not only allow complex queries but also facilitates fast response times, what is important for the overall usability of the *open-laws.eu* platform.

[Rz 13] Future extensions will include advanced repositories for user-generated documents (Jackrabbit Oak, 2015), advanced inline citation functionalities, as well as a business process engine (Apache Syncope, 2015) to support workflows for teams within the platform.

3.2   Community Engagement Trough Open Source Software

[Rz 14] To succeed in the challenging process of not only establishing a legal community platform such as *openlaws.eu*, but also to extend it beyond organisational borders, a multiple perspectives' view is necessary regarding the innovation process (West & Lakhani, 2008). An integration of both internal and external innovation inputs represents one of the most sustainable approaches towards this endeavour (Lakhani & Panetta, 2007). As all components within our platform are based on open source software projects, we provide the community the opportunity to actively contribute and positively influence the platform itself. However, motivations for contributing in open source projects strongly differ (Lerner & Tirole, 2002). Some developers support projects just for the sake of altruism, while others try to push their own reputation. Many contributors also want to improve the systems that they use on a personal basis (Bogers & West, 2012). The

combination of all these motivations is key to establishing a system *by* the community *for* the community. Still, participation alone is not enough; modularity and an open mind set towards new ideas are crucial to foster the establishment of the platform (Lerner & Tirole, 2002).
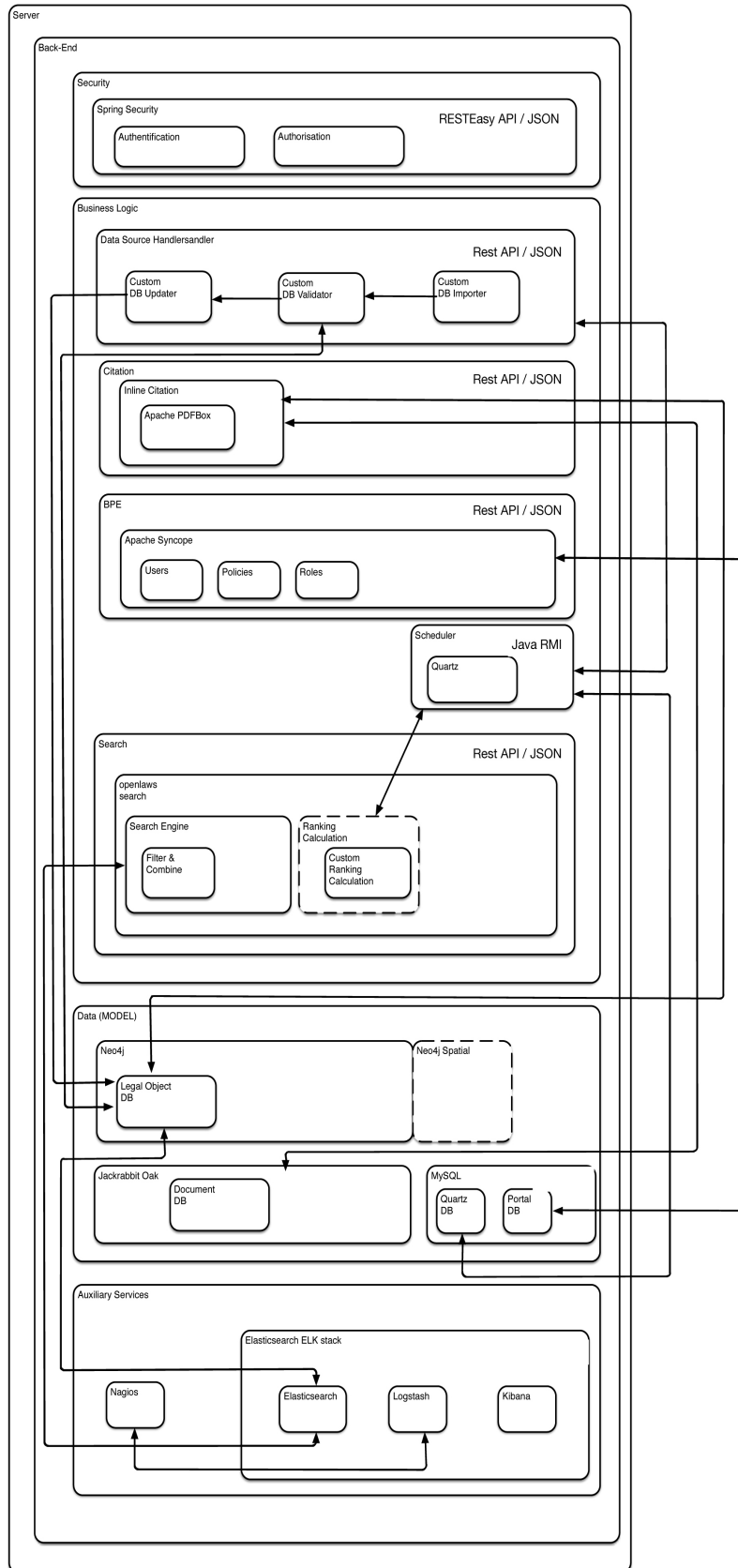
Figure 3: Openlaws.eu technology stack

[Rz 15] The *openlaws.eu* platform achieves this on the one side through the incorporation of before-mentioned open technologies while on the other side providing a «sandbox» environment called «openlaws labs» so everyone can create additional modules or can provide connectors to additional data sources.

4   Open Legal Data — The Openlaws.eu Data Model

[Rz 16] This section describes the first version of the data model employed within the *openlaws.eu* platform. A schematic overview of the suggest data model can be seen in Fig. 4.

Figure 4: Openlaws.eu graph-based data model

The data model within the graph database covers both, legal data as well as the data of the users of the platform. Users can be organised in teams for distribution of legal items within a closed group. For the sake of grouping users and teams, both classes can be associated to an organisation. On the other side, there are the legal items called *legal objects* within our data model. A legal object

covers the meta data of a legal item as well as its actual content. The data model employed within the *openlaws.eu* platform utilizes hierarchical inheritance. This means that every child class of a legal object automatically subsumes all attributes of the parent in addition to its own individual attributes. There exist several kinds of legal objects. One category of legal objects belongs to user- generated legal items such as publications for example. The other legal object types refer to legislation items out of different legal information databases as well as additional user-generated meta data. In addition, every legal item from a legal database is (if available) comprised out of its distinct articles.

[Rz 17] For the first prototype launch event in March 2015, it is planned to include the Eur-Lex legislation items as well as the Austrian RIS. In the next steps, legislation from the UK and the Netherlands will be included during the project.

There do exist several other legal objects to enhance the meta data of legal items within the database. Legal objects can, for example, be extended by user-generated tags as well as (if available) by keywords from the original data source. This combination supports the search process and enables filtering for certain types of legislation items. Users can create highlight objects, which can be compared to citations in scientific writing. Legal professionals may also enrich legal items with annotations, which correspond to commentaries of legal texts. History objects provide a change log of annotations in case of alterations or several versions. Last but not least, there exist folder objects. These objects can be used to group items of interest together to a certain *causa* or context. Folders can be private, shared with team members or made public to be available throughout the entire platform.

5 Conclusion and Outlook

[Rz 18] Exiting and innovation-triggering events predominantly happen at collision points of domains. The *openlaws.eu* project is such an example. Many technologies, which already became standards in other fields, have the high potential to elevate the creation, handling, and exchange of open legal data to a new level. The aggregation of legal databases to one entry point, combined with extensive search functionalities and an innovative social layer will provide access to legislation in a powerful new way. In the next step, the first prototype will be presented at a community event in form of a code camp at the University of Applied Sciences in Salzburg in March 2015. The feedback from the development community, paired with the newly gathered ideas and concepts from this code camp will be used to continue the development to the next phase paired with the integration of additional international legal data sources.

6 Acknowledgement

7 References

AngularJS (2015). Available online: https://angularjs.org/ (5 January 2015).

Apache Syncope (2015). Available online: http://syncope.apache.org/ (8 January 2015).

Bogers, M., & West, J. (2012). Managing distributed innovation: Strategic utilization of open and user innovation. Creativity and innovation management, 21(1), pp. 61—75.

Bootstrap (2015). Available online: http://getbootstrap.com/ (5 January 2015).

Elasticsearch (2015). Available online: http://www.elasticsearch.org (9 January 2015).

Fielding, R. (2000). Representational state transfer. Architectural Styles and the Design of Network-based Software Architecture, pp. 76—85.

ISO/IEC/IEEE (2011). Systems and software engineering — Architecture description. ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), pp. 1—46.

Jackrabbit Oak (2015). Available online: http://jackrabbit.apache.org/oak/ (10 January 2015).

Lakhani, K. R., & Panetta, J. A. (2007). The principles of distributed innovation. Innovations, 2(3), pp. 97—112.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. The Journal of Industrial Economics, 50(2), pp. 197—234.

Liddle, S. W. (2011). Model-Driven Software Development Handbook of Conceptual Modeling, Springer, pp. 17—54.

Martin, R.C. (2003). Agile software development: principles, patterns, and practices. Prentice Hall PTR.

Neo4j (2015). Available online: http://www.neo4j.com (5 January 2015).

Quarz (2015). Available online: http://quartz-scheduler.org/ (7 January 2015).

Rozanski, N., Woods, E. (2011) Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives, 2nd Edition. Addison Wesley.

Spring Security (2015). Available online: http://projects.spring.io/spring-security/ (7 January 2015).

Wass, C., Dini, D., Eiser, T., Heistracher, T., Lampoltshammer, T.J., Marcon, G., Sageder, C., Tsiavos, P., & Winkels, R. (2013). «OPENLAWS.EU,» in Proceedings of the 16th International Legal Informatics Symposium IRIS 2013, February 21—23, 2013, Salzburg, Austria, 2013, pp. 209—211.

West, J., & Lakhani, K. R. (2008). Getting clear about communities in open innovation. Industry and Innovation, 15(2), 223—231.

Thomas J. Lampoltshammer, Research associate | Lecturer for Applied Informatics, School of Information Technology and Systems Management, Salzburg University of Applied Sciences, Urstein Süd 1, 5412 Puch / Salzburg, AT, Thomas.Lampoltshammer@fh-salzburg.ac.at; http://www.fh-salzburg.ac.at

Christian Sageder, CTO, BY WASS GMBH, Felix-Ennmoserweg 13, 5061 Elsbethen, AT, christian@sageder.eu; http://www.bywass.at

Thomas Heistracher, Head of Dept. of Informatics and Software Engineering, School of Information Technology and Systems Management, Salzburg University of Applied Sciences, Urstein Süd 1, 5412 Puch / Salzburg, AT, Thomas.Heistracher@fh-salzburg.ac.at; http://www.fh-salzburg.ac.at