Go-Lab

Global Online Science Labs for Inquiry Learning at School

Collaborative Project in European Union's Seventh Framework Programme
Grant Agreement no. 317601

Deliverable D4.2

Specifications of the Learning Analytics, Scaffolding Services, and Add-on Services – initial

Editors	Tobias Hecking (UDE)
	Yiwei Cao (IMC)
Date	29th April, 2014
Dissemination Level	Public
Status	Final

The Go-Lab Consortium

Beneficiary Number	Beneficiary Name	Beneficiary short name	Country
1	University Twente	UT	The Netherlands
2	Ellinogermaniki Agogi Scholi Panagea Savva AE	EA	Greece
3	École Polytechnique Fédérale de Lausanne	EPFL	Switzerland
4	EUN Partnership AISBL	EUN	Belgium
5	IMC AG	IMC	Germany
6	Reseau Menon E.E.I.G.	MENON	Belgium
7	Universidad Nacional de Educación a Distancia	UNED	Spain
8	University of Leicester	ULEIC	United Kingdom
9	University of Cyprus	UCY	Cyprus
10	Universität Duisburg-Essen	UDE	Germany
11	Centre for Research and Technology Hellas	CERTH	Greece
12	Universidad de la Iglesia de Deusto	UDEUSTO	Spain
13	Fachhochschule Kärnten - Gemeinnützige Privatstiftung	CUAS	Austria
14	Tartu Ulikool	UTE	Estonia
15	European Organization for Nuclear Research	CERN	Switzerland
16	European Space Agency	ESA	France
17	University of South Wales	USW	United Kingdom
18	Institute of Accelerating Systems and Applications	IASA	Greece
19	Núcleo Interactivo de Astronomia	NUCLIO	Portugal

Contributors

Name	Institution
Tobias Hecking, Sven Manske, H. Ulrich Hoppe	UDE
Yiwei Cao, Nils Faltin	IMC
Sten Govaerts, Andrii Vozniuk & Christophe Salzmann	EPFL
Lars Bollen & Anjo Anjewierden	UT
Elio San Cristobal	UNED
Danilo Garbi Zutin	CUAS
Pablo Orduna	UDEUSTO

Legal Notices

Executive Summary

This deliverable describes the initial version of the specifications of learning analytics, scaffolding, and add-on services of Go-Lab. All these services provide additional functionality to teachers, students and lab owners using the Go-Lab portal (see D5.2). This deliverable consists of two major parts: (i) the learning analytics and scaffolding services and (ii) the add-on services.

Learning analytics aims to collect and analyse user activities to make learning and learning environments more effective and efficient. The Go-Lab learning analytics services provide means to track user activities and analyse this tracked data. This provides the foundation for guidance mechanisms for students through scaffolding, as well as intelligent decision support for teachers and lab owners. More specifically, the learning analytics services provide support for recommendations, intelligent feedback for students, and analytical reports that help to design better inquiry based learning scenarios and spaces.

This deliverable describes the architecture of the learning analytics services in detail. Furthermore, it explains how this service integrates with the Go-Lab portal and a mechanism that enables privacy of the tracked data controlled by the teacher. We regard the learning analytics service as an enabler of scaffolding applications and thus the learning analytics services and its feedback loop together provide the scaffolding services.

The add-on services consist of the bartering platform and the lab booking system to support Go-Lab Portal in different aspects.

The bartering platform offers teachers peer assistance through a tutor social platform for expertise sharing related to online labs and inquiry learning spaces. Teachers are motivated to help other teachers and share their skills and knowledge about online labs on the bartering platform. Furthermore, the bartering platform also attempts to make the Go-Lab Portal sustainable and usable by many schools through active interactions among teacher communities and a credit system, ranging from social rating to payment mechanisms.

Since remote labs can only be used by a limited number of users at the same time, the Go-Lab Portal needs services to arrange which users can use a lab at a given time. Therefore, a Go-Lab calendar-based booking system is offered to manage remote lab booking tasks. In the booking system three Go-Lab booking schemes are specified for use by remote labs in order to get Go-Lab as well as external remote labs ready for Go-Lab users' use.

In this document usage scenarios, requirements and initial component specifications are described and contextualised with existing research.

The specifications in this deliverable will be updated and finalised in D4.6 (M33). Furthermore, a first version of the implemented prototypes will be delivered in

D4.4 (M24) and the final implementation will be described in D4.8 (M36). As a preview for D4.4, Appendix B briefly describes the implementation efforts, achieved up till M18.

Table of Contents

1. Introduction

This deliverable specifies the learning analytics, scaffolding and add-on services in Go-Lab. These services provide additional functionality to the Go-Lab portal to support teachers, students and lab-owners. The Go-Lab portal is an inquiry learning portal where teachers can discover, create, edit and share ILS making use of online labs and apps appropriate for their courses. The portal also allows teachers to use such ILS, without installation effort, in their course with students so students can acquire domain knowledge and scientific methodology skills, while doing experiments using online labs and the inquiry learning methodology (layed out by WP1 in D1.1) supported by inquiry based learning apps (see D5.1 & D5.3). The high-level architecture of the Go-Lab portal (Govaerts et al., 2013) consists of two main components (see D5.2), namely the lab repository[1] and the ILS platform[2]. The lab repository provides a collection of online labs, apps and ILS. Such ILS can be shared by the teacher. These shared ILS can be imported in the ILS platform, which enables the teacher to create and edit his ILS using apps and labs from the lab repository combined with external learning material (e.g. videos and slides). Through the ILS platform, students also use the ILS prepared by the teacher in the classroom. Although, the Go-Lab portal provides a wide range of functionality, the value of the Go-Lab portal can be further enhanced by services fostering learning, collaboration efficiency and effectiveness as well as usability. Learning analytics, scaffolding and add-on services can be seen as such services for the Go-Lab portal, which concentrate on different aspects of user support in Go-Lab.

The learning analytics services support the Go-Lab portal by laying the foundation for guidance mechanisms for students through scaffolds, as well as intelligent decision support for teachers and lab owners. More concretely, the learning analytics services provide methods and technologies for recommendations, intelligent feedback for students through analytics supported interventions, and analytical reports that help to design better learning scenarios and ILS. To support the learning activities of a student in an ILS, instructional scaffolding apps are being developed, which allow personalised guidance of a student during the inquiry cycle, for instance in the form of immediate feedback (e.g. help messages) or adaptation of apps to the learning process (see D1.1 and D5.1 for more information). Scaffolding is mostly interleaved with the learning analytics services in such way that modelling and analysis of learners' activities is the basis to generate personalised feedback. Therefore, we regard the learning analytics service as an enabler of scaffolding applications and thus the learning analytics services and its feedback loop together provide the scaffolding services for scaffolding apps.

The functionality of the Go-Lab add-on services is twofold: (i) to support peer assistance among teachers, and (ii) booking mechanisms for remote labs. Operating remote labs or getting acquainted with inquiry learning methodologies can be a burden. To alleviate this, the bartering platform supports peer assis-

---

[1]The lab repository is available at `http://www.golabz.eu`
[2]The ILS Platform is available at `http://graasp.epfl.ch`

tance for teachers to operate online labs and design ILS. Teachers can request help sessions with peers or tutors through different communication channels. Social rating and a credit system will be integrated to reward tutors. Teachers are supported to become tutors after they get expertise from peer assistance.

Booking mechanisms are crucial for a remote lab because of the limited physical resources. Besides that, teachers want to be able to reserve a remote lab exactly during their class. Remote labs may employ different booking schemes because of different usage scenarios and implementations. In order to enable lab booking in a federation of online labs the add-on services specify a unified Go-Lab lab booking system with calendar.

The remainder of this deliverable is structured in two major sections. While Section 2 outlines the architecture and specifications of the learning analytics and scaffolding services, Section 3 describes the architecture and specifications of the mentioned add-on services. Both parts are introduced with objectives of the specific services and user scenarios. Initial specifications of software architectures and processes are derived from a requirement analysis for learning analytics and scaffolding services, and add-on services respectively. Finally, the appendices provide some extra details on the learning analytics feedback loop, the implementation work accomplished so far and surveys we want to conduct to get a better understanding in what users want for the bartering platform UI and lab owners' requirements on the booking system.

2. Learning analytics and scaffolding services

The following sections outline the initial specification of learning analytics and scaffolding services in Go-Lab. After introducing learning analytics and scaffolding and its objectives, user scenarios in the context of Go-Lab are described. In order to design useful services, requirements are derived from the scenarios by identifying the information needs of the stakeholders and requirements for the architecture. Based on the requirements and specifications for the Go-Lab learning analytics and scaffolding services are presented. Finally, we give an overview of the current state of work and an outlook.

2.1. Introduction to learning analytics and scaffolding services

Siemens (2012) defines learning analytics as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs". To achieve these goals, learning analytics brings together different fields, i.e. business intelligence, web analytics, educational data mining and recommender systems (Ferguson, 2012). Methods comprise (1) content analysis of produced artefacts by learners (such as concept maps (de Jong et al., 2010; Hoppe et al., 2012; Clariana et al., 2013) and texts (De Wever et al., 2006; Liddo et al., 2011; Southavilay et al., 2013)), (2) learner behaviour analysis (Zaıane & Luo, 2001; Facca & Lanzi, 2005; Duval, 2011) and (3) social network analysis (Laat et al., 2007; Zhang et al., 2007; Harrer et al., 2007). A major challenge, also for the Go-Lab project, is to combine these approaches in a flexible infrastructure in order to achieve a productive synergy (Suthers et al., 2013).

In the Go-Lab portal as an environment for various kinds of inquiry-based learning activities many types of data are produced including traces of interaction with the system and results of the learning process. The learning analytics services make use of such data to provide analytical information for the Go-Lab portal (D5.2) and add-on services (section 3) in order to foster awareness, create individual scaffolds and recommendations for students as well as supervision support for teachers.

The focus of the analytics and scaffolding services in Go-Lab is on interaction and content analysis. For a better overview the learning analytics services can be aligned to three aspects of the Go-Lab portal where different activities take place that can be supported. One aspect is the portal as a whole. In the portal, the learning analytics services can help teachers to find appropriate resources, for example through recommendations of apps, labs and ILS templates. The second aspect is to monitor the students behaviour at the level of inquiry learning spaces (ILS) in order to get a clear picture of their overall learning activities. This includes log protocols comprising of time stamped events like the access of a resource and app usage. The analysis of the student actions when using particular apps and the thereby produced data is a third aspect which leads to tailored scaffolding and feedback mechanisms.

Scaffolding mechanisms assist learners in tasks that they cannot solve alone

without guidance. Based on previous analysis and modelling of learner behaviour it is possible to adapt the scaffolds to the needs of each particular learner. Typical scaffolding mechanisms are immediate feedback for example recommendations or the adaptation of an app according to the learner model. Scaffolding services for learners rely on information about the labs, its users and their user activities as well as the subject domain of the lab. Therefore scaffolding apps are dependent on the lab metadata scheme developed in WP5 as well as the smart device and gateway for remote labs (see D4.1).

2.2. User scenarios

The following user scenarios highlight how the stakeholders can benefit from the learning analytics services within the Go-Lab portal and apps.

Scenario LA-1: Learning analytics for the lab repository. Physics teacher Norman is recommended by his colleagues to check out the Go-Lab lab repository to engage his students more in his course. On the landing page Norman finds a set of the most popular labs, apps and shared ILS. Norman is attracted by a lab on radioactivity because he cannot do such an experiment in his classroom. On the detail page of this lab, he finds next to more detailed information on the lab, a wide variety of additional resources he can use in combination with the lab. For instance, there is an available list of the most used inquiry learning apps that other teachers have used together with this lab in their ILS. Furthermore, a list of popular ILS created by other teachers using this lab is shown, where Norman can check what learning activities other teachers created. In a sidebar some statistics are shown that illustrate how often and when the lab has been used as well as a small map to illustrate where. Norman sees that this lab is actually used most in his country so he assumes it will be well aligned with his curriculum. Next to this, there is a list of experts who can assist him with this lab. He discovers that some of them are fellow teachers who have taught several courses with the lab and even the software developer of the lab is available. For now Norman decides to try things out before requesting help and creates an ILS with this lab.

Scenario LA-2: Learning analytics on the ILS platform level. While Norman builds an ILS in the portal for his upcoming lesson, he is adding relevant material. Norman wants to monitor the progress of his students and see how much time each student spent on the single phases of the inquiry learning cycle and whether some students are left behind. Additionally he wants to see which learning resources are used by his students. Therefore Norman searches the lab repository and finds two apps that visualise the time spend in phases and statistics on resource usage. Now he can assemble a teacher dashboard from these analytics apps in a dedicated subspace of his ILS. The analytics apps require the tracking of the user activities inside the ILS. This is done by a dedicated user tracking agent. This agent is an artificial space member, and hence can access all activities inside the ILS. It can be added and removed by the teacher,

in order to control the tracking of user activities. Hence, before the course session starts Norman makes sure that the tracking agent is member of the space as an observer. After Norman has successfully finished the ILS, he shares a secret URL with his students so they can access the ILS. During the session, he realises by observing the teacher dashboard, that students that have used the reading material in the orientation phase finish the experimentation phase more quickly with better results. After the course is over he recommends these reading materials to his colleagues who create similar ILS.

Scenario LA-3: Learning analytics in apps. Teacher Norman has additionally created an ILS on electric circuits for his physics class. The students are all around 16 years old. They should experiment with a circuit simulator and examine the effects of resistors in different arrangements of the circuit. For the conceptualisation of the experiments, the students should create a concept map to model their knowledge on the interplay between the different factors in electricity like voltage, electric current and resistors. In addition, the students should formalise their hypotheses. Therefore, Norman chooses the concept mapping app and the hypothesis scratchpad from the lab repository and adds it into the corresponding space. These apps are connected to the analytics service and track actions and produced artifacts. In this ILS, a student, Max, created a concept map and now he creates a hypothesis: "If the resistor value increases, the electrical current decreases." In the experimentation phase he takes longer than his peers to conclude the experiment. The system notices that he relates concepts in his hypothesis but did not connect them in his concept map previously, and gives him a hint that he might go back to the conceptualisation phase and plan the experiment more accurately, because some inconsistencies between the concept relations and the created hypothesis have been detected. He reviews the reading material again and adds the missing relations to his concept map. With this step, it becomes clear to him that the resistors influence the electrical current, and that this can be shown if the voltage is kept constant. This helps him to finish the experiment.

After class, the teacher wants to review the previous session in order to detect potential lack of knowledge of the students and to get information on how future sessions can be improved. As described in scenario LA-2, he has added some pre-configured analytics apps to his personal space. He uses one of these apps to investigate individual concept maps created by the students but also an aggregated concept map which is an overlay of all created maps. This helps him to detect a common misunderstanding of the relation between electric current and voltage especially in the early phase of the session. Additionally, another analytics app that displays frequent sequential patterns of student actions while interacting with the lab, displays that many of the students do not adjust the voltmeter after they adjusted the resistors. This convinces him that his students have a common misconception of the relations between electric current, voltage and the influence of resistors and he decides to revise these things again in with his students in the next class.

2.3. Requirements

In this section we identify different stakeholders of learning analytics in Go-Lab. When students use the ILS platform, they produce data, which can be analysed by the system to produce an immediate intervention to support the learning process (feedback loop). On the one hand, the architecture and infrastructure for learning analytics in Go-Lab needs to support the feedback loop through logging and notifications. On the other hand, the services need to meet the legal requirements, because storing and processing personal data might concern privacy of the learners. The described requirements are likely to be extended in the final specification D4.6 when user experiences are available.

2.3.1. Stakeholders and their information needs

We consider that the production of useful analysis processes is not an easy task, and it needs different methods and apps for each one of these different actors. There are at least four types of users who can benefit from analytics and scaffolding services.

Students.   The main information need of students is awareness on the learning process and guidance. Thus, the learning analytics services should enable self-reflection and recommend activities and resources to the learner individually.

Teachers.   The teacher's perspective focuses on monitoring and instrumentation for classroom management, especially awareness of student behaviour in an ILS. In addition, Go-Lab supports teachers in finding and structuring learning material.

Lab owners.   Analysis of lab usage can provide valuable information to lab owners. Hence, lab usage statistics can also be very useful in resource planning and lab booking. Insights into the usage patterns of labs helps to assess whether the lab is used as intended or modifications and user guidance are needed.

Researchers.   Learning analytics can support Go-Lab researchers in decision making regarding the Go-Lab development. The large-scale data collection can foster new insights in the research field of technology-enhanced learning. Furthermore, the learning analytics services, will be exploited by the development of other services like personalisation and recommendation (see D5.1).

2.3.2. Relevant data sources for learning analytics and scaffolding

Students, teachers and lab-owners produce many digital traces. The goal of the analytics services is to make use of this data in order to provide the stakeholders mentioned in 2.3.1 with intelligent information targeting their information needs.

Table 1.: Sources of data in Go-Lab

	Inquiry learning spaces	Apps	Lab repository
Students	Resource access and usage, e.g. number of downloads of a document, Produced artefacts, e.g. notes, hypotheses	Action sequences, Usage data of apps, e.g. time spent on an app, Interaction with online labs through an app	
Teachers	Resource assembly	Usage data of teacher apps	ILS Template creation, ILS Template adaptation
Lab owners		Information about smart devices (D4.1), Mathematical models of labs	ILS Template creation, Lab metadata

For a better overview which data are available on different levels of the Go-Lab portal, table 1 summarises the types of data and their producers that can be used for learning analytics purposes. On the level of ILS it should be recorded which resources and apps students use in order to observe their learning behaviour and to get a clear picture of how students deal with the various learning opportunities in ILS. An example for this would be scenario LA-2 where records of student behaviour on space level are presented to the teacher in form of a dashboard. Information about which resources, apps and labs teachers assemble in ILS can be used to generate recommendations as described in LA-1. While data produced on the level of inquiry learning spaces reflect the general use of the space content, action log protocols of interaction with inquiry learning apps and online labs allow deeper insights into concrete learning activities. This can be used to create scaffolds in the form of user feedback by the system as in scenario LA-3. Since learning activities always take place in the context of a certain subject domain, and are intended to be centred around an online lab, it is often necessary to obtain this information also. Otherwise it would not be possible to generate domain dependent feedback for learners and teachers. On app level this can be information from the smart devices (D4.1), for example a list of possible actions in an online lab. Subject domains, keywords and other information that can be used to tailor analysis and learner feedback for different contexts, are available as metadata in the lab repository and are often provided by the lab owners.

A major challenge for analytics services in Go-Lab is to aggregate this heterogeneous and multidimensional data and apply mixed analytic methods.

2.3.3. Privacy and anonymity

Tracking user activities and analysing them is twofold. On the one hand it envisions the goal of supporting and improving learning, on the other hand it leads to concerns of privacy of an individual. Data processing and data retention collide with privacy and therefore these aspects are under legal restrictions. A framework for these aspects is given through the EU directives 95/46/EC (Data Protection Directive) and 2006/24/EC (Data Retention Directive). These directives are implemented in different countries in a different way. The bottom line of these directives is that personal, identifiable information (PII), namely information that leads to the identity of a learner in a direct or indirect way, should not be exposed through the system and only be retained if necessary for the use. Typically, research purposes soften these limitations a bit, e.g. in the law for schools in NRW, Germany (SchulGes NRW[1], §121). Examples for PII are a user's real name or real ID-Numbers (Passport, Social Security Number, ...), Email address, Personal characteristics (including photographic image) and any indirection that leads to one of these criteria. One country-independent approach to cover all the directives and guidelines is to provide a "Do not track" option to disable the logging of the user actions. For handling sensitive data when tracking users, the "OECD Recommendations For Data Protection" [2] provide good guidelines for assuring privacy by design for Go-Lab.

2.3.4. Requirements for the Go-Lab learning analytics infrastructure

Currently architectures for learning analytics infrastructures are being developed in different contexts. This incorporates also business analytics and data mining tools (Kraan & Sherlock, n.d.). Some tools are designed for specific types of learning systems like learning management systems (LMS) (Fortenbacher et al., 2013). Tools like Crunch [3] are more focused on the development and the offering of targeted analytics services. Another promising approach for the aggregation of learning activity data across different environments is the Tin Can API [4]. In order to prevent fragmentation of the analytics services in Go-Lab the specified infrastructure should be able to integrate those different concepts in an extensible manner. The Open Learning Analytics project (Siemens et al., 2011) advocates for modular systems that allow openness of process, algorithms, and technologies which is an important feature in a heterogeneous field as learning analytics. This should also be the line followed for the analytics architecture in Go-Lab. There are various opportunities to use the Go-Lab portal to create inquiry scenarios with virtual and remote labs. This requires the possibility to create custom analytics solutions as well as the offering of common services by integrating existing systems.
While the mentioned systems meet the demand of modularity, they dismiss the

---

[1] http://www.schulministerium.nrw.de/docs/Recht/Schulrecht/Schulgesetz/
  Schulgesetz.pdf, retrieved 28th February 2014.
[2] http://www.oecd.org/internet/ieconomy/oecdguidelinesontheprotectionofprivacyandtransborderflows
  .htm
[3] http://crunch.kmi.open.ac.uk/
[4] http://tincanapi.com/

chance to tailor learning analytics to multiple stakeholders. Analytics services can be used for ex-post analysis by researchers to get insights into learning processes or to design better mechanisms to guide students via scaffolds. In contrast to the perspective of ex-post analyses, the learners can also immediately benefit from such systems, typically through interventions. Learning analytics processes can be modelled as a cycle (Clow, 2012), where analysis and feedback steps are interleaved with learning. This leads to the following functional requirements for the learning analytics and scaffolding architecture in Go-Lab.

Action Logging.   The scenarios LA-2 and LA-3 describe situations in which the behaviour of the students within the system is analysed. Before such analyses can be performed, the user activities need to be captured through the system, which can be achieved through action logging. Action logs must consistently reflect the users actions in the inquiry learning space. This comprises user behaviour on the space level, for example access to resources and subspaces as well as specific actions when using an app. The logs have to be in an agreed format so that analysis methods can be developed independently. For the LA-1 scenario, action logging is also required, but on a different level. Since lab repository users are often not logged into the system, we do not know their user identity and will be tracked anonymously, mainly focused on page access.

User feedback.   Referring to the learning analytics cycle, (Clow, 2012) describes the key to the successful application of learning analytics as "Closing the loop' by feeding back this product to learners through one or more interventions". In order to provide immediate feedback to the learners as described in scenario LA-3, action logging alone is not sufficient. Therefore, appropriate channels need to be established, where the analysis results can be fed back to the learner. To use analysis results for immediate intervention, analysis components should be triggered in such way, that notifications can be generated and fed back to the learner on time. Scaffolding apps have to be able to handle different kinds of notifications ranging from prompts to reconfiguration or adaptation of apps according to the learners' needs. In the technical sense this means to implement mechanisms that transform the analysis results produced by algorithms into human understandable feedback, for example messages that recommend a certain action. Apart from that, a challenge is to design the feedback meaningful and comprehensible in a pedagogical way. This should go along with the research of the pedagogical partners in Go-Lab.

Data storage and ex-post analysis   Furthermore, the learning analytics services should improve the Go-Lab portal as a whole by providing decision support to teachers and lab owners. Learning analytics and educational data mining can be used in such case to acquire knowledge about the learners in a larger scale. The intervention does not immediately affect the learners that produce the data, but following generations of learners. Such ex-post analyses are
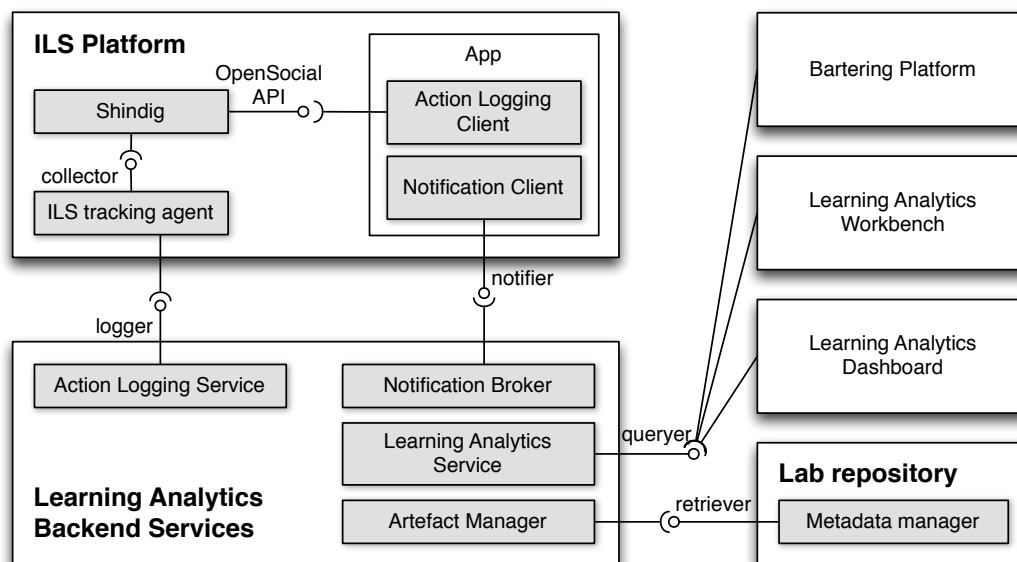
Figure 1.: Components of the learning analytics infrastructure.

useful, to develop methods to provide and tailor guidance. This requires an adequate data management where data from different sources can be aggregated for analysis purposes. The gathered data must be processable by different analytics technologies. A typical impact of retrospective analysis of data gathered over a certain period in time are recommendations as described in scenario LA-1.

2.4. Architecture and specifications

This section describes a specification of the Go-Lab learning analytics infrastructure. Based on the requirements above, the architecture is intended to be modular, and open for extensions. First an overview on the general services and components is given. After that specific solutions for an agent based architecture, data retrieval and the generation of user feedback are given. This is followed by an initial specification of action logging and notification formats. Applications that use the described infrastructure and privacy mechanisms are presented at the end of this section.

2.4.1. Components overview

Figure 1 depicts an overview of the learning analytics infrastructure. It can be divided in server-side services, namely the backend services, and client side services in the ILS platform in the form of a user tracking agent and client APIs for action logging and notifications.

2.4.1.1. Services of the learning analytics backend

The backend services provide four interface components for different aspects of data acquisition, analysis and feedback mechanisms that are connected to the other components of the Go-Lab portal, for instance the user tracking agent for learning analytics in ILS (AngeLA) and apps. The Activity Logging Service

establishes an endpoint for the tracking agent and apps to push event logs of user activities (see scenarios LA-2 and LA-3). This service delegates the handling of these data to the internal components (see section 2.4.2) where it can be either be stored or processed directly. Another component for the acquisition of data is the Artefact Manager. This service can be used by the internal analytics components to gather artefacts from external data sources. In contrast to action logs, artefacts can be any kind of structured data that are persistently stored outside of the learning analytics backend. This includes learning material, textual resources produced by learners and metadata of the content stored in the lab repository. For example, the lab repository offers a web service to access metadata about apps, online labs and other resources (D5.2). The Artefact Manager can then be used to gather additional information about labs, like the subject domain from the lab repository in order to create tailored scaffolds through messages or reconfiguration of apps. In some cases such scaffolds can be directly created by apps. However, to decouple the analysis of user actions from a particular app, it is desirable to use the server-side analytics components. The Notification Broker can then be used to send messages to particular clients where they can be displayed (scenario LA-3) or be used to configure a scaffolding app. Therefore, clients (apps) can establish a socket connection via Socket.io[5] with the Notification Broker by using the Notification Client API. This allows the learning analytics backend services to send messages to the services in the ILS platform actively. Further there will be data harvested over a certain period of time which can then be used for ex-post analysis that reflect longer term information. Hence, the Analytics Services interface which allows other components to access data and analytics routines in the learning analytics backend. This allows the flexible connection of a variety of tools and services. This can be for example a recommendation engine (scenario LA-1), the bartering platform (see section 3) or the "analytics workbench" (Göhnert et al., 2013) as an already existing analysis tool (see section 2.4.7).

2.4.1.2. Services in the inquiry learning spaces platform

Within the ILS platform there are three services that play an important role for learning analytics and scaffolding. Inquiry learning apps, especially scaffolding apps like the concept mapper app and the hypothesis scratchpad (see D5.3) can use dedicated Javascript APIs to handle action logging and notifications in a unified manner. In Figure 1, these services are named Action Logging Client API and Notification Client API. The tracking agent for learning analytics (AngeLA) listens for log events of apps as well as activities on space level like the access of resources e.g. videos and the subspaces for the different inquiry learning phases. Therefore, this agent uses the inherent tracking mechanisms of the existing ILS platform. It is able to send all the gathered activities to the Action Logging Service of the learning analytics backend services as explained above.

---

[5]Socket.io, `http://socket.io`
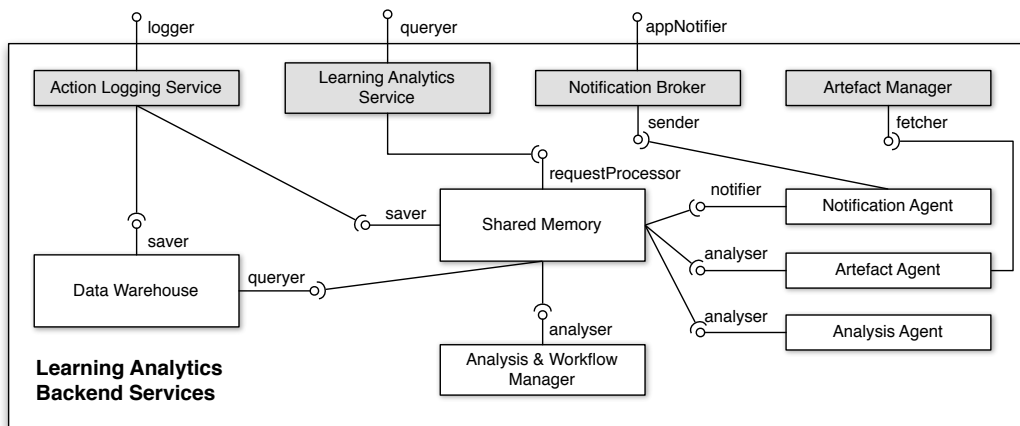
Figure 2.: Components of the learning analytics infrastructure.

2.4.2. The analytics backend services agent system

The internal components of the learning analytics backend are depicted in figure 2. The architecture is based on a multi-agent system with a distributed shared memory implementing the Tuple Spaces concept (Gelernter, 1985), namely SQLSpaces (Weinbrenner, 2012). This component provides a shared memory for agent coordination and communication, but also a workspace for analysis. Basically it can be seen as a blackboard where agents exchange messages in the form of tuples (ordered lists of data objects). Analysis agents, for example an agent that analyses action log data produced in inquiry learning spaces can register listeners by specifying tuple templates. Whenever a tuple that represents an action log is written to the space and matches the template, the agent is notified by SQLSpaces. This enables a loose coupling of components because data exchange and communication is completely mediated by the shared memory, manifesting an implicit protocol for agent communication. Agents can be designed to perform analyses and data acquisition autonomously or on-demand (see Figure 2). This approach has been used successfully in other inquiry learning environments especially in the context of the SCY project (Giemza, Weinbrenner, Engler, & Hoppe, 2007; Anjewierden, 2012). For Go-Lab the shared memory is intended for temporary storage of tuples. For persistent data storage we propose a data warehouse approach (Inmon, 1992). This is a common way to aggregate heterogeneous data from different sources for analytics purposes. The Activity Logging Service (Figure 2) writes incoming activity logs to the shared memory for direct analysis but also into the data warehouse for long term storage. In the data warehouse these activity logs can be enriched by resource content gathered by the Artefact Agent that uses the Artefact Manager as connector to external components. The data in the data warehouse can then be used for long term ex-post learning analytics and made available for specialised analysis tools.

2.4.3. Data access and retrieval in the Go-Lab portal

As described in the requirements section 2.3 the learning analytics infrastructure should be able to gather data from different sources relevant for analytics. In this section, we present the user activity tracking solutions for use in the ILS platform, while preserving privacy, and for anonymous tracking in the lab repository. Furthermore, technical possibilities to retrieve metadata about resources in the lab repository (online labs, app, ILS templates) are presented.

2.4.3.1. Activity tracking in the lab repository

As mentioned, lab repository users will often not be logged in, so detailed user information will not be available while tracking. Therefore, we will rely on existing web analytics solutions. We selected Piwik[6] because it is open source, free and allows us to store the tracked data on our own servers to ensure better privacy than with cloud-based solutions like Google Analytics.[7]

2.4.3.2. Activity tracking in the inquiry learning spaces platform

To supply the LA backend services with activity logs, user actions need to be recorded in the Go-Lab ILS platform and sent to the LA backend. Such user actions are represented in the ActivityStream format (as specified in D5.1). For instance, an app (mostly OpenSocial widgets in Go-Lab, see D5.1) tracks user activities by propagating an ActivityStreams object to the OpenSocial ActivityStreams API, which channels this request to the Graasp (ILS platform) API that saves the ActivityStream object in the Graasp database. Among the recorded actions are the following: add, update, invite, join, remove, access. The portal stores recorded actions according to the ActivityStreams format in a normalised form inside of MySQL database. Encoding action logs in the ActivityStream format will be discussed in section 2.4.6 in more detail.

Trust in the Go-Lab portal privacy is crucial for the platform adoption. To enforce the privacy policy, the concept of Learning Analytics Agent (AngeLA) was introduced. AngeLA provides a way to configure action logging policy on per-space basis. AngeLA is represented in the Go-Lab portal in the same way as any other real user. The agent can be invited or removed from a space. This mimics a physical classroom where an external person can be invited to monitor the situation and record observations.

The actions are only recorded within spaces that have the learning analytics agent as a member. The Go-Lab portal collects activity logs as they happen (in real-time) from all spaces with AngeLA as a member and sends them via HTTP post requests to the LA backend for further processing.

Since teachers can decide to turn of learning analytics user tracking, the apps making use of learning analytics data and results, should be intelligently and robustly designed. Apps should be both aware that tracking is turned off and the learning analytics backend services can be unavailable for unforseeable

---

[6]Piwik, `http://piwik.org/`
[7]Google Analytics, `http://www.google.com/analytics/`

reasons.

2.4.3.3. *Metadata retrieval from the lab repository*

As described in D5.2 the lab repository contains various information about apps, labs and ILS templates. It offers interfaces to retrieve this metadata. As described in Section 2.4.1.1 the Artifact retrieval service can use this interface to retrieve information about apps, labs and space templates in order to enrich the data gathered by the activity tracking services.

2.4.4. Learning analytics strategies for apps

Due to the flexibility of the learning analytics service and its modular architecture, apps can make use of the learning analytics service and the tracked user activity via different channels. In this section, we will briefly describe the possibilities:

- Apps can make use of the OpenSocial API of the ILS platform to retrieve tracked ActivityStream objects. This retrieved activity could be used for simple visualisations or analysis, e.g. for dashboard visualisations.

- Apps can use the 'queryer' interface to retrieve results from the learning analytics service. This can be useful for an app to directly access the required analysis results from the backend services, e.g. for more advanced visualisations and basic recommendations.

- Apps can be automatically notified when new analysis results will become available in the learning analytics backend services. This case is explained in detail in the next section.

2.4.5. Data flow and feedback loop

In order to fulfil the requirement of an effective feedback loop for immediate intervention as required in Section 2.3.4 this section outlines the typical information flow when feedback should be given to a student directly by scaffolding apps. Figure 3 depicts the complete data flow cycle when activity logging in the portal and backend analysis is involved. Given scenario LA-3 where the student Max uses the concept mapping and the hypothesis creation apps. The concept mapping app uses the notification API to subscribe to the Notification Broker as a listener for messages from the analytics backend services when it becomes active (1.1). Whenever the student adds a concept to the concept map, adds a relation between concepts or draws a hypothesis the action is logged by the corresponding app. The user tracking agent AngeLA (scenario LA-2) takes these logs (1.2) and sends them to the Action Logging Service (2) which itself delegates the log to the Action Logging Broker (3). This broker stores the received logs in the data warehouse for long-term storage (4.1) but also in the form of tuples in the shared memory (SQL spaces) (4.2). The action logs can be allocated to specific apps using the generator field in the activity logs (see section 2.4.6). A dedicated concept mapping analysis agent listens for tuples that have been send by the concept mapping and hypothesis creation apps, and hence it is triggered whenever action logs from these apps are written into the SQL

spaces (5). When the agent detects a misconception between the hypothesis and the concept map as described in scenario (LA-3) it sends a message back to the app by inserting a notification tuple into the SQL spaces (6). Therefore it uses the addressing scheme described in Section 2.4.6. Then the Notification Agent becomes actively notified by the SQL spaces that there is a new notification (7). This agent then uses the Notification Broker to send the message to the right client (8). Because of the addressing scheme the Notification Broker can choose the right socket connection to emit the message (9). The final handling/displaying of a message is under the responsibility of each particular app. Appendix A contains a more elaborately modelled version in UML.

2.4.6. Logging and notification format specifications (initial)

Logging Format   D5.1 specified that action logs in Go-Lab should follow the ActivityStreams format [8]. In the ActivityStreams format, a user action has subject-verb-object-target semantics like "Max (Actor) adds (verb) a concept (object) to a concept map (target)". Additional context information can be added, by providing additional fields e.g. "generator". Some fields allow nested structures, e.g. an actor can have specific fields for id, name, etc. Consequently the common serialisation for activity streams is the JSON format [9]. Listing 2.1 shows an example of the mentioned action that is encoded in the ActivityStreams format.

---

[8]http://activitystrea.ms/
[9]http://json.org
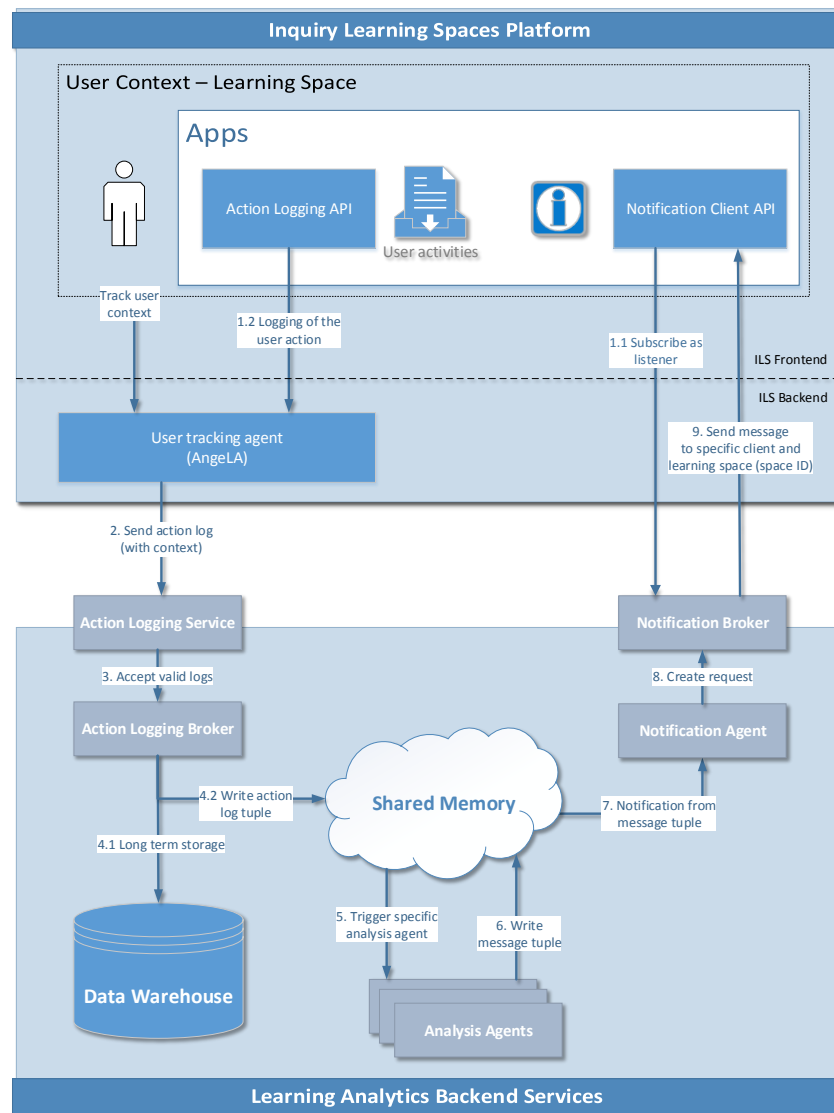
Figure 3.: Data flow from logging over analysis to immediate user feedback.

```
{
    "published": "2014-02-10T15:04:55Z",
    "actor": {
      "objectType" : "person",
      "id": "4fd5fe62-2c5d-4cfd-bbb9-eeecc4b147bc"
    },
    "verb": "update",
    "object" : {
        "objectType" : "concept",
      "id": "1f53e17c-90a3-4e27-8f48-09c09b694326"
        "content": "fluid density"
    },
    "target" : {
      "objectType": "concept_map",
      "id": "b4ab6f3a-74e9-4e8a-bea0-0d242626d8f2",
      "displayName": "my Buoyancy concept map"
     },
    "generator" : {
        "objectType": "application"
        "url": "http://go-lab.gw.utwente.nl/..."
        "displayName": "conceptMapper"
    },
    "provider": {
    "objectType": "ILS",
    "id": b4ab6f3a-74e9-4e8a-bea0-0d241236d8f2
    }
}
```

Listing 2.1: An example action log of a user, who created a concept in a concept map, encoded in the ActivityStreams format.

On the app level the field provider indicates the space in which an app lives. The field generator specifies the app and the field target specifies a concrete object e.g. a particular concept map. With this hierarchical indexing it is possible to allocate each log to a specific user and instance of an app. When a user starts an app within an ILS, the fields actor, generator, provider and in most cases target can be initialised directly by the activity logging API because the values of these fields will not change for the logs generated during the app usage. In contrast to that, object, verb and published (the timestamp) can differ in every log. These fields can be set by app by using the logging API.

Notifications   As activity logs, notifications can be serialised in JSON format. In this initial specification the notification message must contain four fields, namely type, importance, target and content. Listing 2.2 displays a notification message in more detail. In this initial specification the type of a notification can

be a textual message, a concrete resource e.g. for recommendations or configuration parameters for scaffolding apps. It should also be possible to specify importance levels for messages which can result in different handling of a message on the client side. The target must contain the id of a receiving client. An app can be addressed uniquely by the ID of the inquiry learning space in which it lives and its own identifier within the ILS. Until now, the receiving clients are apps in the ILS platform. However, in the future an ILS itself could also be a client, listening to notifications. Feedback can then be displayed on a more upper level not related to a particular app, for example resource recommendations. Hence, the target should also contain a type argument which indicates the type of receiver. Consequently the content field comprises message text, resource URLs or configuration parameter according to the type of notification.

```
{
        type: "prompt | resource | configuration",
        importance: "1...10",
        target: {
                type: "ILS | app | ...",
                id: "{uuid}"
},
        content: {
                text: "Consider the following concepts..."
                url: "http:\\..."
                image-url: "http:\\..."
                configuration: { property-value list }
        }
}
```

Listing 2.2: Structure of a notification message in JSON format.

2.4.7. Applications using the learning analytics infrastructure

As described before, the described analytics infrastructure should not be a closed system. It is rather intended to be connected with various other components by offering targeted services. The following, some already existing, or under development, applications are intended to be integrated with the learning analytics infrastructure.

Analytics Workbench: Major parts of the analytics workbench (Göhnert et al., 2013) as it is now available have been developed in the context of the EU project SiSOB [10]. This project, funded under the Science in Society theme, aimed at measuring impact of science and research on society beyond classical bibliometrics using a network-analytic approach. The overall goal of the workbench development was to provide a generic and extensible analysis framework with an integrated user interface that would allow even non-computer experts to

---

[10]"An Observatorium for Science in Society based in Social Models", `http://sisob.lcc.uma.es/`

access the full analytical power behind the tool through a visual form of specification and that would also allow for reusing and sharing the created analysis work flows. The workbench comes with a web-based user interface for designing analysis processes. The workflows are represented in a visual language based on the pipes-and-filters metaphor. Here, the filters represent analysis steps and links (pipes) between these filter modules describe the data flow. Workflows can be stored, loaded, and also shared with other users. This user interface is backed by a multi-agent system and each of the modules in the visual language corresponds to one agent in the backend.

A wide range of analysis modules are currently available for various functions including general data handling facilities, many standard and also more specialised network analysis techniques, modules for the processing and analysis of activity logs (including modules for deriving statistical information, creating networks, and doing sequence analysis), as well as a wide range of different visualisations.

In Go-Lab, we make use of the workbench to specify, generate and test various kinds of analysis procedures, with a special focus on sequential analysis of user actions.

The left side of Figure 4 shows an example workflow, where different concept maps of a single session are used to build an aggregated graph (scenario LA-3), which is displayed in the end as analytics app for the teacher (right side of figure 4).

The main benefit for Go-Lab to integrate this workbench is to enforce a multi-stakeholder perspective on learning analytics which goes along with the requirements. A separation of analysis (authoring of workflows) and target platform (displaying the results) helps to address different target groups as students, teachers, researchers and lab owners. The outcome of the integration is a system that creates portable widgets automatically out of workflows. These small applications can be embedded in widget platforms, particularly the Go-Lab ILS-Platform (Graasp) as can be seen in figure 4. While the widget is authored through a graphical programming language, the created widget can be used by a teacher to foster collaborative work in the classroom supported through the analytics system.

Besides the graphical approach, the workbench offers in its integrated version the possibility to create analytics services through an externally triggered execution of a workflow. A widget, which is created from such workflow, does not simply encapsulate static results of an execution of a workflow at a single point in time. Furthermore it encloses an external representation (JSON-based) of such a workflow and it can modify or trigger this workflow at any time. The analytics services (see section 2.4.2) delegate such a request through the shared memory to trigger the workflow execution.

The approach is more general, which gives the possibility to target other platforms in the future with low efforts, e.g. Piwik, which will be used to monitor the lab repository. From an architectural perspective, the workbench is integrated into the backend services, sharing the same infrastructure, namely the shared

Figure 4.: Left: visual representation of an analytics workflow. Right: Result visualisation in the Graasp platform.

memory for agent coordination and data transportation. By using the Analytics Service interface it can also access data from the data warehouse, see section 2.4.1.1.

Teacher dashboard: According to Stephen Few (Few, 2007) a dashboard is a visual display of the most important information needed to achieve one or more objectives, consolidated and arranged on a single screen so the information can be monitored at glance. Following the definition, the teacher dashboard provides a teacher with the possibility to observe learning activities on individual and class level, allowing the teacher to intervene when necessary. Such learning analytics dashboards have proven to be useful and effective for teachers in many settings (Verbert et al., 2013).

We plan to build teacher dashboards from portable apps implementing OpenSocial and ActivityStreams specifications (Vozniuk, Govaerts, & Gillet, 2013). For instance, one app could be used to show how many students are currently at each phase of the inquiry cycle, another app could display average time spent on each task (scenario LA-2). When several widgets are organised as a dashboard, such a dashboard can be exported as an app itself. This app consisting of other apps is called a metawidget (Blattner, Glinert, Jorge, & Ormsby, 1992). When bundling a dashboard as a metawidget, it becomes portable and can be deployed and ran on any learning platform implementing OpenSocial and ActivityStreams specifications.

Once built, such a dashboard can be customised and reused in multiple ILSs. To encourage reuse and customisation of teacher dashboards by other teachers, both dashboards metawidgets and individual apps will be shared on the lab repository.

The Recommender System: The recommender system will make use of the data collected by the learning analytics backend services and metadata available in the lab repository. As scenario LA-1 described the main task for the recommender will be: (i) recommendations of teacher community members and lab owners as tutors for creating inquiry learning spaces for the bartering platform and (2) recommendations of labs, apps and ILS for the lab repository and ILS platform. In D5.1, we described the planned recommender system in some details. The year 1 review report contained the following statement related to this plan: "Considering the relatively small number of remote and virtual labs identified so far, the effort and resources foreseen for activities related to metadata compilation and tagging as well as the underlying automatic recommending mechanisms may need some downsizing." Based on these suggestions, we plan to start from a simpler recommendation approach to save resources. The implementation of the recommender has not started yet, but design is on-going. We plan to use existing libraries, such as Apache Mahout[11] or ElasticSearch[12], to be able to develop a basic recommender with less resources. Initially, popularity based on basic usage traces (e.g. resources posted and consumed) and the structure of ILS will be used by the recommender to recommend mainly labs and ILS. In a later stage, the recommender results can be improved by using social interaction traces (e.g. rating, tagging and commenting). This data is generated by teachers and thus causes less privacy concerns than student generated data. The implementation details of the recommender system will be described in D5.5 (M32).

2.4.8. Assuring privacy

To fulfil the privacy requirements, described in Section 2.3.3 we will provide a (technical) framework and a policy for enforcing privacy in the system. From the technical perspective this is handled through different mechanisms:

- Pseudonymisation of user IDs. The ILS platform obfuscates the user ID for the communication with the analytics backend services. To preserve unique user identity, a (hidden) mapping is retained in the ILS platform, as indicated in Figure 1. This mapping is necessary to enable the feedback loop, especially for addressing the notifications to the right user.

- Separation of portal user profiles and analytics profiles. The profiles of users in the analytics backend only contain the data that is necessary for analysis.

- "Do not track". This option can be enabled by a teacher for an ILS. This deactivates the tracking of users, particularly the logging of their actions both in the ILS platform and the learning analytics backend services.

On the non-technical side we plan to have a transparency policy, where we make clear to the users of the ILS platform, particularly the teachers, which kind of information is tracked and how it is processed through the system. This

---

[11]Apache Mahout, `https://mahout.apache.org/`
[12]ElasticSearch, `http://www.elasticsearch.org/`

is especially necessary in some European countries, where transparency of external storage platforms is a requirement by law, if teachers want to use it with sensitive data.

2.5. Summary and outlook

In this part of the deliverable we presented the initial specification of the learning analytics and scaffolding services. Students, teachers, researchers and lab owners have been identified as main stakeholders of learning analytics. A basic requirement for learning analytics is to capture of the learners' activities in a comprehensive way. To enable scaffolding of the learning process through immediate feedback by the system, a feedback channel from the analytics infrastructure to the learning environment is needed. Apart from that, relevant data for analytics like action logs of students need to be stored persistently and be accessible by analytics tools for retrospective ex-post analysis.

In order to achieve a general framework that enables the implementation of the required services, the envisaged architecture of the learning analytics backend services is based on a modular multi-agent system with agent communication over a shared memory. This enables a flexible integration of components because the complete coordination of agents is mediated by the shared memory. In the ILS platform activity tracking is done using the user tracking agent (AngeLA). AngeLA is technically a space member of an ILS. The agent records actions happening within an ILS where it is a member. This allows for configurable user tracking which is in control of the teacher as space creator. Apps in the ILS portal can use dedicated APIs for action logging and notifications that simplifies the connection to the learning analytics backend. After the technical realisation of the specified learning analytics infrastructure, future work will cover the development of specific analytics methods for user activities in ILS and recommendations in the lab repository. There will also be a detailed planning of useful analytics enabled scaffolding mechanisms in coordination with the pedagogical partners in Go-Lab. First case studies of learning analytics in Go-Lab with a prototypical implementation of the specified services will be presented in D4.4.

3. Add-on services

The Go-Lab add-on services provide value-adding, easy-to-use services to the Go-Lab Portal. These add-on services reduce the complexity of using online labs and inquiry learning spaces. These services promote and enable a wider adoption of Go-Lab. The term "add-on" itself also indicates that those services and systems are not bound to Go-Lab Portal, but are plugable if necessary.

3.1. Introduction to add-on services

The add-on services enable Go-Lab users to book lab usage time and to barter for skills and competences needed when using the Go-Lab Portal (see D5.2) or inquiry learning. In the remainder of this deliverable we will refer to both services as (1) the booking system and (2) the bartering platform. D1.1 defines three types of online labs in Go-Lab. Only remote labs require users to book usage time because of resource constraints. Therefore, the booking system only targets remote labs. The booking system provides a reservation mechanism to schedule lab usage time. The bartering platform helps users find experts who can help with conducting inquiry learning using online labs. Those experts could be experienced teachers or any user who has sufficient knowledge and skills using online labs and inquiry spaces. Furthermore, the booking system can be employed by the bartering platform to reserve users' help sessions as well.

First, why will we specify booking system for Go-Lab infrastructure in Work Package 4? There are various existing booking schemes for the existing remote labs, e.g. iLab and WebLab Deusto. Users can go to each lab web page to book the lab step by step. However, users often face problems such as how to find a lab available for use or how to book a remote lab. The Go-Lab Portal provides access to online labs from different lab owners. Accordingly, how to book the labs with different booking mechanisms in a unified way can become a challenge. The booking system needs to make a specification for Go-Lab remote labs; whereas it needs to adapt the existing booking schemes of remote labs in order to integrate those existing labs. From this viewpoint, it is important to have a Go-Lab booking system to simplify and unify the booking experience for users.

Second, why does Go-Lab Portal need a bartering platform? The Go-Lab Portal supports comprehensive inquiry learning spaces using online labs, which might not be straightforward to operate or turn into a classroom activity (Gillet et al., 2013). Lack of ICT skills is one of the barriers to lower school teachers' using Internet technologies (Breuer et al., 2009) (see also D6.3). To partially support teachers with their ICT skills, the portal offers diverse apps for scaffolding and guidance of students practising inquiry learning. However, apps may not always be able to cover all assistance which students and teachers require. In the bartering platform, experts with experience, knowledge and skills in operating specific online labs, using these labs in the classroom or in inquiry learning shall offer support and help to teachers. What motivate teachers to help each other? Siemens points out that nature of reward, engagement, participation,

and diversity of network as such factors that influence learning processes in (Siemens, 2008). The bartering platform uses reward mechanisms to engage teachers with peer assistance. In detail, the teachers that get help from an expert can award this expert with some form of credit, e.g. social media badges or virtual currency. Furthermore, the bartering platform is helpful to enhance school teachers' ICT skills and to support teachers lifelong learning. Teachers can become an expert to help the other teachers after being trained by the experts for several times.

We call these experts who offer help *tutors* in the bartering platform. *Bartering* refers to the whole process from users and tutors looking for each other, agreeing on how help is conducted, to having finally an online help session. Thus, the bartering process includes a lot of communication activities, including credits exchange activities.

A main component of the bartering platform is the *tutor social platform* to facilitate teacher community build-up. The other component, the *credit system*, will be implemented based on the development of the business model together with Work Package 9 (ref. D9.3, M24), as an exploitation strategy for Go-Lab Project's sustainability. The development of the credit system will be planned for three phases in order to develop Go-Lab user communities gradually. Above all, teachers' use of Go-Lab bartering platform is free of charge, if teachers register to Go-Lab as a Go-Lab user. Lifelong learners out of Go-Lab users may eventually pay to get tutors' help, which maintains sustainability of the Go-Lab project.

The next sections are organised as follows: first, the user scenarios sketch how add-on services are used in the Go-Lab portal. Accordingly, requirements are analysed for the booking systems and the bartering platform respectively. The state-of-the-art services related to booking and bartering are compared. Then the overall architecture of the add-on services show the functions and relationships to the Go-Lab Portal. Architecture of the booking system and the bartering platform are specified in detail. Finally, we summarise this specifications of the Go-Lab add-on services.

3.2. User scenarios

Four user scenarios are illustrated to show how the add-on services work together in the Go-Lab portal. They can be seen as a whole.

Scenario AO-1: John is a school teacher and wants to give his students a physics class with an online lab on buoyancy. He logs in to the ILS platform and creates an inquiry learning space. He adds the WebLab Aquarium remote lab into his ILS to investigate the density of fluids. He wants to demonstrate the WebLab Aquarium to his students during one of his physics classes on Wednesday morning in one of the coming three weeks. Thus, John goes to the lab repository, browses to the detailed description page of this lab and clicks the "How to book" button. After which he follows a wizard with instructions how to book

the lab and clicks the button "Book the lab". A calendar of WebLab Aquarium pops up and shows there is only one time slot available in two weeks that fits his physics class. He books it right away by adding the class name and number of students. He is notified that 3 remote lab instances are recommended and booked for his class of 25 students. He also receives an email and notification in his inquiry learning space which hosts this lab. During his class, he connects to the lab transparently without further setup and is able to use the remote lab user interface with his students to control different objects floating or sinking in the aquarium to observe the force.

Scenario AO-2:   The next days, John finds it would be nice if he could show some live observation of the galaxy during his physics class. In the Go-Lab lab repository, he finds the Faulkes Telescope lab. Again, he clicks the "How to book" button and gets an instructional page that describes that he first has to go to the Faulkes project web site and register. After registration, he should send an email to the Faulkes project team to book a time session. With this clear booking process instruction on the Go-Lab portal, John is able to book a session with the Faulkes project team smoothly. John finds it convenient that the lab repository provides guidelines to book this specific lab.

Scenario AO-3:   Another school teacher, Jane, wants to use WebLab Aquarium on the next day. However, WebLab Aquarium is already booked by other users and is not available for her class. Jane searches on the lab repository by inputting her class time and the class subject. A list of labs available at the requested time and for her topic is shown. Jane is glad to find one and books it for her class (following the procedure described in AO-1).

Scenario AO-4:   John wants to use the RED lab during his class, but he does not know how to operate the lab. He finds a list of lab tutors on the lab repository page of the RED lab. He clicks the first one in that list, Chris, and retrieves Chris' profile page that shows that Chris has helped over 10 Go-Lab users and shows very positive feedback from the people he helped. Chris offers online video sessions. John finds Chris' profile promising and books a help session on the next day right away. Then during the help session, both John and Chris use a video calling tool of the bartering platform to communicate with each other. This video calling tool will be offered by some well-known vendors and its user interface should be familiar to teachers, so that teachers are not required additional ICT skills. Besides asking questions face-to-face in the video call, Chris is also able to share his laptop screen about how RED works. John gets very good instructions and can ask directly questions to Chris. Chris even helps John adapt his ILS together with the RED lab. After the help session, John gives a good rating to Chris' help and writes a positive review. Now Chris' profile has achieved an "excellent tutor" badge after having helped the 15th happy Go-Lab user.

3.3. Requirements

The aforementioned user scenarios focus on the users' viewpoints and hide the technical complexity from the technical viewpoints. For example, the Smart device and Smart Gateway (see D4.1) are involved in AO-1 for the communication between Go-Lab Portal and the existing lab management system of WebLab Aquarium. The requirements are analysed thoroughly based on both users' and technical aspects and are grouped in functional and non-functional requirements. First, we discuss the common requirements of all add-on services and then elaborate on specific requirements for each service.

3.3.1. Non-functional requirements on add-on services

Overall, add-on services need to meet these non-functional requirements (NFR) (Chung & Prado Leite, 2009).

Interoperability and portability: Go-Lab add-on services need to be interoperable with the Go-Lab portal; between the add-on services themselves; and with online labs (through the smart device and smart gateway specification). For example, the Go-Lab booking system needs to be interoperable with the booking mechanisms of specific online labs. Furthermore, portability ensures other lab platforms to easily adopt Go-Lab Add-on Services, so they do not have to implement their own booking services. This requirement supports a large-scale federation of online labs across platforms and online lab user communities.

Easy Integration and openness: As additional services, they need to be easily integrated in the Go-Lab portal. For those labs and inquiry learning spaces that want to use Add-on Services time to time, Add-on Services can be disconnected and reconnected easily.

Scalability: Add-on services are required to support a large number of users, e.g. to barter teachers' skills to help each other, or to support small as well as big school classes to use remote labs.

User-friendliness and transparency: A user-friendly user interface is essential for a successful and widely-adopted booking system and bartering platform. The complexity of the different external systems that add-on services have to communicate with, should be invisible for the end-users.

Security & Privacy: Only authenticated users are allowed to use the Go-Lab Add-on Services. User authentication of Add-On Services complies with Go-Lab Portal user management system. Information about booking and bartering activities need to be managed and secured by the add-on services. Privacy of registered users should be preserved.

3.3.2. Functional requirements on booking systems

The target users of the booking system are teachers and lab owners. Teachers book remote labs for their STEM classes with the booking system. Although students are the main users of the remote labs, they are not authorised to book labs individually. Because the remote labs are booked once for a whole class, teachers will take the responsibility to book labs. This is also in line with the Go-Lab user authentication scheme which specifies that only teachers can possess a user account in the Go-Lab Portal, while students can use an ILS without login.

Lab owners, who use the Go-Lab booking system, are willing to share their remote labs with Go-Lab Consortium. There is an agreement between Go-Lab Consortium and the lab owners to allocate certain lab usage time slots to Go-Lab users only.

Based on User Scenarios AO-1 to AO-3 from the previous section and with regard to some potential functions, functional requirements on Go-Lab Booking System include:

- **Single sign-on.** The add-on services need a simple authentication mechanism for user friendliness.

- **Booking a lab.** To provide a consistent user experience, the complexity and diversity of the different booking systems of remote labs should have a unified UI for teachers to book a lab.

- **Consulting lab calendar.** Teachers can access the calendar with lab availability and make a reservation.

- **Administer lab calendar.** Lab owners can add and edit time slots when their lab will be available to Go-Lab users in the lab calendar. Lab owners can indicate how many physical instances they provide access to. For instance, there are many physical copies of the RED lab[1] available that could be booked separately.

- **Booking for multiple sessions.** Teachers can input the number of students when booking. The booking system considers how many sessions or how many physical instances of this remote lab are appropriate for students' use at the class.

- **Running a booked lab in an ILS.** Students and teachers need to be able to effortlessly execute a booked remote lab in their ILS. Booking information and data communication of the experiment and its progress has to pass through the ILS.

- **Notifying booking.** Teachers need to be notified via email or in the Go-Lab portal about their booking at the time of the booking and a few hours before the booking for awareness cues.

- **Notifying lab booking progress in ILS.** Students and teachers need to

---

[1]RED lab, `http://www.golabz.eu/lab/red-lab`

be provided with time left and total time information of the booking in the lab client app in the ILS.

- **Cancelling booking.** Teachers need to be able to cancel their lab booking to deal with mistakes. This function may take place in two circumstances. During the lab booking process, teachers can cancel the booking at any step before they confirm the booking. During the time period between a successful booking and the lab usage time, teachers are able to cancel their bookings, if they don't need the labs later. The occupation status of the labs will be changed for future re-booking by other teachers.

- **Closing a booked lab session.** Once the reservation time has expired, the remote lab experiment has to be gracefully ended in the ILS. The lab could for instance only allow observation and no longer operation.

- **Searching ready-to-book labs.** This booking system provides a time-based search to find labs available within a certain time slot.

- **Booking statistics.** The booking system will track the use of labs. This can be further analysed by the learning analytics service.

- **Booking of other resources than labs.** The booking system can also be used to book lab tutor time and other scarce resources.

3.3.3. Functional requirements on the bartering platform

The target users of the bartering platform are teachers, students and lab owners. Teachers may need support and help when they encounter problems during the creation of an ILS or when using online labs in class. On the other hand, teachers would like to help other teachers with their expertise. Lab owners can use the platform to offer their professional support to lab users. Students can be interested in getting help to carry out experiments on their own and experienced students could help others. In the context of the bartering platform we refer to the users who provide help as tutors.

Based on Scenario AO-4, the functional requirements on Go-Lab Bartering Platform include:

- **Single sign-on.** Although authentication is needed for most functionality of the bartering platform (e.g., not for searching), it should be user friendly, thus the same login information as in the Go-Lab Portal is reused which most bartering platform users will already have.

- **Managing a tutor profile.** A tutor can create a profile and update it according to tutors' help offers.

- **Managing a user profile.** A user profile contains information of a person who is requested or looks for help of a tutor. A user profile can be created, updated and managed according to their help requests and the expertise that users have gained.

- **Commenting and rating.** Users could comment and rate the tutors after they get help and support from the tutors. This will be reflected in the

tutors' profiles.

- **Contacting, bartering, and communicating at tutor time.** Communication channels (e.g. email and video or audio chat tools) are required for contact information and bartering process between tutors and Go-Lab users. Above all, this communication channel enables tutors to assist Go-Lab users. A video chatting channel is provided to create a real-time, face-to-face like help from tutors to Go-Lab users. Tutors also require a resource upload tool to share learning resources with the users who need tutoring, such as video resources. Screen sharing and documents sharing can further facilitate the help session.

- **Booking tutor time.** Besides contacting tutors via email, a booking functionality is also provided that gives users a clear overview of tutors' availability via a calendar. This booking process can be also cancelled accordingly, if some changes happen later.

- **Recommending tutors.** Go-Lab users will be provided with recommendations of potential experts that can help them with operation of a specific lab or creation of an ILS.

- **Searching tutors.** Go-lab users can search the tutors for certain labs or ILS'.

- **Listing tutors.** A list of experienced tutors is provided per lab and ILS in the lab repository.

- **Assigning credits to users.** Users get a certain number of credits when they start using the bartering platform in order to book tutors' help session. This functionality extends the complete bartering process. Credits could be social media badges, vouchers, and currency.

- **Exchange credits among Go-Lab users and tutors.** As a sequence of *assigning credits to users* tutors offer their help sessions against users' vouchers. Also tutors are able to use them to get help from other tutors if necessary.

3.4. State of the art

The existing lab booking schemes and bartering platforms are surveyed to help specify the Go-Lab add-on services.

3.4.1. Existing lab booking schemes

Any existing remote laboratory used by students in production supports some booking schemes (also called scheduling schemes) (Orduña & García-Zubia, 2011). Three mechanisms can be found in the literature:

1. *Queueing: First Come First Served (FCFS)* manages exclusivity of one or several copies of an online lab through a queue that can have a priority model (e.g. local students go first). It is useful in batch laboratories where students submit a task and they are notified of the results when finished and wherever the interaction is short (for a few minutes). With longer in-

teraction time or number of students, the queue time could become longer and hardly to estimate.

2. *Time session specific booking* enables teachers to book lab usage sessions for specific time periods, e.g. next Tuesday at 11:00. The teachers and their students are guaranteed to have exclusive access during that lab usage time.

3. *No scheduling* is useful when the laboratory can be used by a large number of users, e.g. in some online labs where students can only observe the experiments instead of controling the lab instances by themselves.

Examples of the mechanism *queueing* are most WebLab-Deusto laboratories (Orduña et al., 2011), iLab Shared Architecture batch laboratories (Harward et al., 2008), ISILab (Ponta et al., 2009) or RemotElectLab (Sousa et al., 2010). Examples of the mechanism *time session specific booking* are iLab Shared Architecture interactive laboratories (Harward et al., 2008), RLab (Safaric et al., 2005), or certain laboratories which rely on external systems for scheduling such as Moodle[2].

However, the granularity of all these systems are on the student level. Their concern is to make it possible to guarantee that only one student can access one resource, so no conflict happens during the interaction. Certain systems, such as WebLab-Deusto, provide multiple concurrent users access to a single laboratory (Orduña et al., 2012), but not collaboratively in group.

In Go-Lab Portal, students are using the online lab in class. Hence, only queues with short-session laboratories can potentially be used. Other mechanisms available in the literature are enabling booking at the group level. For instance, in CASPiE[3] a teacher can book a time slot of 3-4 hours, and only their students can use it. Inside this time slot, there can be different mechanisms (e.g., a queue only for those students, or simply a conflict indicating that the resource is being used by somebody else). This mechanism is suitable as a booking scheme. As of March 2014, this mechanism is also being developed in WebLab-Deusto for all their labs.

On top of these mechanisms, there are mixed approaches in the literature:

The VISIR lab   [4] It internally uses a fast queue where each measurement takes less than a second. While it is an interactive laboratory, students are actually offline (not really using the laboratory) when building a circuit. When they take a measurement, this internal queue is used. In this way, multiple users can share the opinion that they are using the laboratory at the same time. However, if many users are using the system at the very same time (as in ISEP (Alves et al., 2011)), students may not have the same perception of interactivity. For this reason, the original VISIR management system supports a booking mechanism

---

[2]http://www.moodle.org/

[3]http://www.caspie.org/

[4]http://openlabs.bth.se/

for teachers at the group level as explained above. This guarantees that only those teachers' students can use the laboratory at certain time. Groups can be created with a low number of users (e.g. 30-60 users), and they keep the good interactivity perception. Internally, inside these sessions, the fast queue is still used. When used inside WebLab-Deusto, the load balancing is managed among different federated environments using VISIR. So the number of students is split among them (Orduña et al., 2013; Orduña, 2013).

The Labshare Sahara system   It supports using both queues and calendars for the same time slots (Lowe & Orou, 2012). A user may book a session for a particular date or queue it, and a uniform scheme is provided.

3.4.2. Existing bartering platforms

Bartering and exchange activities take place very often in various online platforms where user communities are involved. Even common social media platforms, such as Facebook, Twitter, Instagram, or Pinterest can be used to announce services or goods for bartering, although they do not support the transaction process explicitly.

The existing bartering platforms which support trading, mainly motivate users to save money by trading anything from cooking skills, baby sitters, smartphones, to houses and real-estate. They can cover a large range of goods or services for exchange and bartering, e.g. BarterQuest[5], TradeYa[6], and Swapit[7]. These bartering platforms use points or miles instead of money to equalize trades or acquire items or services. Points and miles can be purchased, which enables these websites non-cash trading.

If we only observe services, knowledge and skill bartering, these bartering platforms are evolved from helpdesks or call centres. Instead of these conventional face-to-face or telephone line based, video chatting has become a promising tool for real-time help sessions between experts and users who have a problem and require a help session. The recently rolled-out Google Helpouts[8] integrates Google Hangouts for video chatting and offers help session with experts (e.g. in cooking or repairing your computer) for free or for monetary payments. The free bartering platform Just Answer[9] supports chatting to solve problems by experts, with the slogan "A new question is answered every 9 seconds". In the context of schools, Trade school[10] is a real world (not online) bartering platform. It makes a community of self-organised schools running on barter in many cities worldwide. Learners barter knowledge and experiences acquired in those schools with barter items such as food.

---

[5]http://www.barterquest.com/

[6]http://tradeya.com/http://www.swapit.co.uk/

[7]http://www.swapit.co.uk/

[8]https://helpouts.google.com

[9]http://www.justanswer.com/

[10]http://tradeschool.coop/

Table 2.: A comparison of the selective existing bartering platforms and Go-Lab Bartering Platform as well as platform

	Bartering objects	Listing	Video chatting	Rating & commenting	Free or payment	Community based
TradeYa	goods + services	yes	no	yes	both	no
Google Helpouts	services (expertise)	yes	yes	yes	both	no
Just Answer	services (expertise)	yes	no	yes	free	no
Trade School	knowledge (expertise)	no	no	no	free	yes
Go-Lab BPF	services (expertise)	yes	yes	yes	both	yes

Table 2 compares Go-Lab Bartering Platform requirements with some selective existing bartering platforms mentioned before. Go-Lab supports the specific user communities who have or search for expertise in online labs. Hence, it is capable of delivering community specific bartering services. At the same time, Go-Lab Bartering Platform can benefit of the dynamics of a social network, for instance regular users that are highly rated can be upgraded to tutors based on this community (cf. (Cao et al., 2010)).

The credits on barter range from social media badges, virtual currency, to real currency. Mozilla has launched the Open Badges open source platform[11] develops the Open Badges standard for online assessment. Similar to badges in FourSquare[12], learners are motivated to learn by gaining widely-accepted Open Badges as an incentive method. (Limpens & Gillet, 2011) proposed a competence model with a virtual currency based decentralised credit system to make incentives for self-regulated learner communities. Google Helpouts employ Google Wallet for the payment system with real currency. In comparison, social media badges motivate users through gamification approaches, while currency-based exchanges bring users monetary profits.
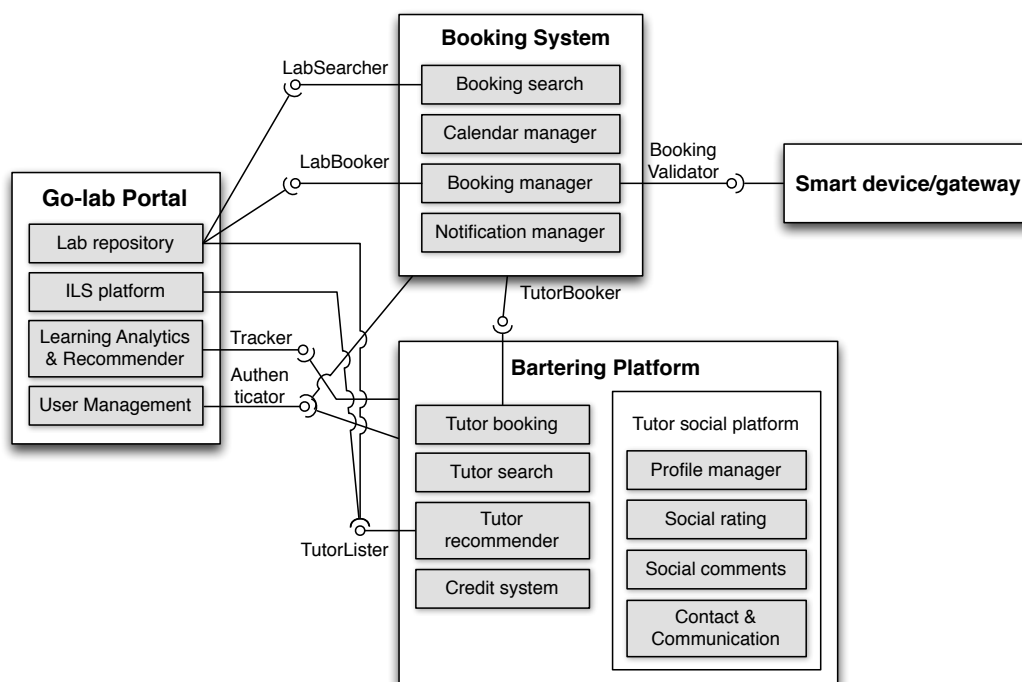
3.5. Architecture and specifications

Based on the requirements analysis, we present an overall architecture of Add-on services with its subsystems in detail in the following sub sections.

---

[11]http://openbadges.org/
[12]http://foursquare.com/

Figure 5.: Architecture of Go-Lab Add-on Services.

3.5.1. The overall architecture

The Go-Lab add-on services consist of two subsystems booking and bartering. Figure 5 depicts the component diagram of Add-on Services and interfaces to remote labs using Smart Gateway and Smart Device specifications (cf. D4.1) and the Go-Lab Portal (Govaerts et al., 2013).

Go-Lab Booking System consists of the booking search, calendar manager, booking manager, and notification manager components. The booking search component offers a search service to find labs available at a given time of a class to ease finding an appropriate lab to book. Such a search can be done in the portal via the LabSearcher interface. The calendar manager provides a lab schedule calendar to both teachers and lab owners and makes use of a centralised booking database. Lab owners can use the calendar manager to edit the time slots where they allow Go-Lab users to use their labs. The booking manager offers all functionality for making a booking to the portal via the LabBooker interface. The booking is validated with the remote lab via the BookingValidator interface with the smart device and gateway. It enables users to use the booked lab within the lab usage time slots specified by the lab owner in the calendar manager. The notification manager notifies users of the booking status and synchronised with smart devices in that lab usage time slot.

Go-Lab Bartering Platform consists of a tutor social platform, a credit system, and a set of components to find tutors. The tutor social platform manages user and tutor profiles and provides social features such as user comments and ratings on the user profiles and the help sessions. The contact & communica-

tion component provides different contact channels between users and tutors, e.g. email, chatrooms or a video chatting tool to conduct the help session. In the bartering platform, users can book a help session with a tutor via the tutor booking component, which supports calendar-based booking through the Go-Lab booking system. Furthermore, one can search for tutors via the tutor search component and get recommendations of tutors by cooperation between tutor recommender component and the Go-Lab Portal via the TutorLister interface. The credit system provides mechanisms to award tutors for the provided help. Besides users' social reputation growth in the tutor social platform (e.g. via gamification badges or scoring systems), the credit system attempts to explore potential business models for experts and users requesting help. More details about the credit systems will be discussed in Section 3.5.3.3.

Both the booking system and the bartering platform use single sign-on, which is realised via the User Management component of the portal via its Authenticator interface. To provide user activity traces to the learning analytics service, user actions are tracked in the bartering platform via the Tracker interface of the Learning Analytics service in the portal.

3.5.2. Architecture and specifications of the the Go-Lab booking system

As mentioned before, to have a remote lab use the Go-Lab booking system, the lab owner needs to agree with the Go-Lab consortium to allow only Go-Lab users to make use of the lab owner's lab at the time slots specified by the lab owner. Surveys are prepared to communicate with lab owners. A complete survey for lab owners can be found in Appendix C.2. Furthermore, every booking by a Go-Lab teacher will be done with one single Go-Lab user identity in the specific booking system of the remote lab. This strategy is chosen as it limits the complexity of providing interoperability between existing types of booking mechanisms. For instance, the strategy allows lab owners to still continue using their existing booking mechanisms in parallel with the Go-Lab booking system.

The Go-Lab booking system will only be used at the time slots allocated to Go-Lab, which simplifies calendar syncing issues between the Go-Lab booking system and legacy booking systems. In addition, by using a single user identity to do all Go-Lab bookings, teachers are not required to create accounts on the legacy booking systems and the interfacing with such legacy booking systems can be made transparent for teachers. The drawback is that lab owners will not know who has been using their lab, but we can provide them with detailed statistics that will provide them likely with deeper insights, since Go-Lab can provide more information than just plain lab stats, e.g. with which apps labs have been used or in which ILS.

3.5.2.1. Go-Lab booking schemes

Due to the variety of existing booking mechanisms with which the Go-Lab booking system needs to be compatible, different integration solutions are presented in Table 3. Generally, the Go-Lab booking system offers three kinds of booking schemes based on a Go-Lab booking calendar. There is also one external booking scheme to provide basic booking help with labs that do not want to

Table 3.: Go-Lab booking schemes

Booking schemes		Description	Lab examples	Realisation summary
External booking		no cooperation in booking; or lab owner is contacted personally via email to book	Faulkes telescope	Go-Lab Portal provides help page on booking or forward booking emails to lab owners
Go-Lab booking	only calendar integrated	the lab owner allows Go-Lab to use his lab on fixed dates, but there is no technical integration between lab booking system and Go-lab booking system	Methyl Orange	using the Go-lab calendar manager, but uses the lab's booking system
	transformed	to integrate some existing (legacy) remote labs, to make the booking compatible. the level of compatibility with the Go-Lab booking system depends on the legacy lab	iLab, Web-Deusto, Lab-Share	transform booking via the smart gateway
	fully integrated	to integrate fully with the Go-Lab booking system	LED	using the Go-Lab booking system and the smart device

make use of the Go-Lab booking system. For instance, this external scheme will provide a description for teachers on how to book the lab with the lab owner's external booking mechanism (e.g. a step-by-step walkthrough to book a lab or email contacts of the lab owner who can further assist teachers), as described in Scenario AO-2.

3.5.2.2. Interface specifications and detailed interaction between components

This section will provide more details on the interaction between the different architecture components and the interfaces defined in Figure 5. The main functional requirements of the booking system are addressed in the following UML sequence diagrams.

Booking a lab & Notifying of booking: Figure 6 shows a sequence diagram of the procedure to book a lab that uses a smart device (see D4.1). When a

sd Book a Smart Device lab
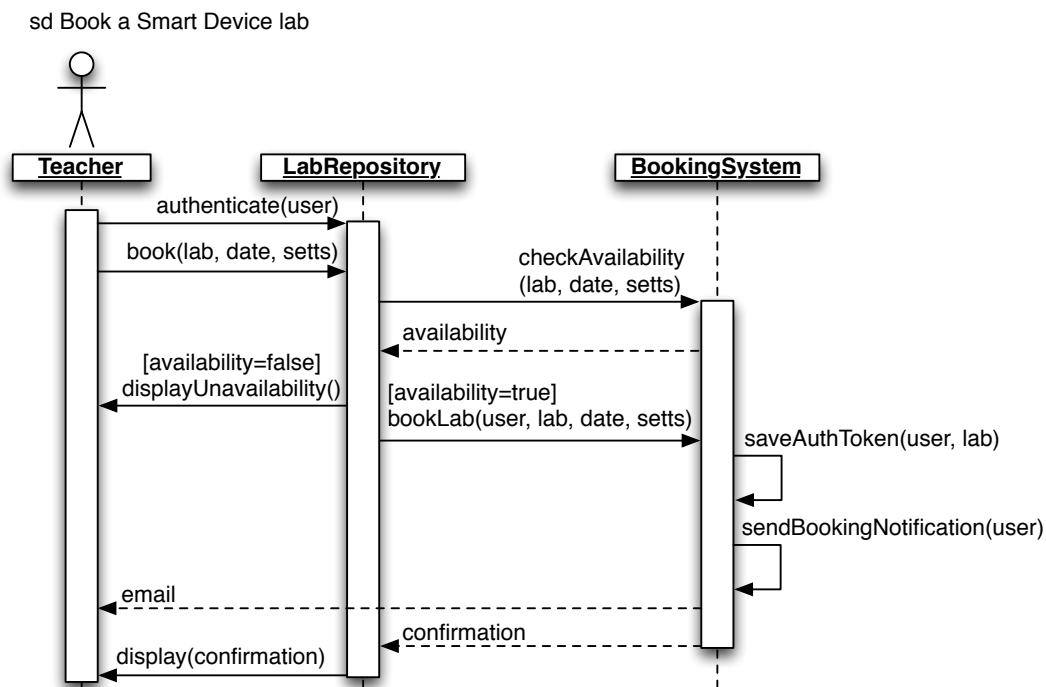
Figure 6.: A sequence diagram that models the booking of a lab using a smart device.

teacher wants to book a lab, she needs to be authenticated to the lab repository using the `Authenticator` interface of Figure 5. The lab repository contains the booking UI. A lab is booked for a given time slot and settings (e.g. the number of students or lab instances) using the `LabBooker` interface. First the availability of the lab is checked in the calendar manager of the booking system. The booking system is aware of the number of lab instances available. When the lab is unavailable, a message in the lab repository UI is displayed to the teacher. In case the lab is available, the lab can be booked using the booking manager and an authentication token is saved to enable the lab access at the specified booking time (see the 'Running a booked lab in an ILS' section below). A booking notification is sent to the user by the notification manager. This notification can be an email and/or a message that pops up in the lab repository or the ILS platform.

The procedure for a lab using the smart gateway (see D4.1) is a bit different. In the case of the smart device, the calendar is completely managed by the booking system. With the smart gateway, a legacy lab can have its own calendar. The lab owner who shared his lab with Go-Lab, agreed to allow Go-Lab teachers to access their labs at configured time slots only accessible to Go-Lab, which enables synchronised calendars between the booking system and the lab. Figure 7 illustrates how a lab behind the smart gateway can be booked and how a booking can be propagated to the legacy lab (although this is not required). Similar to Figure 6, a teacher needs to authenticate, then book a lab for a time slot with given settings. The availability is checked in the booking system. If
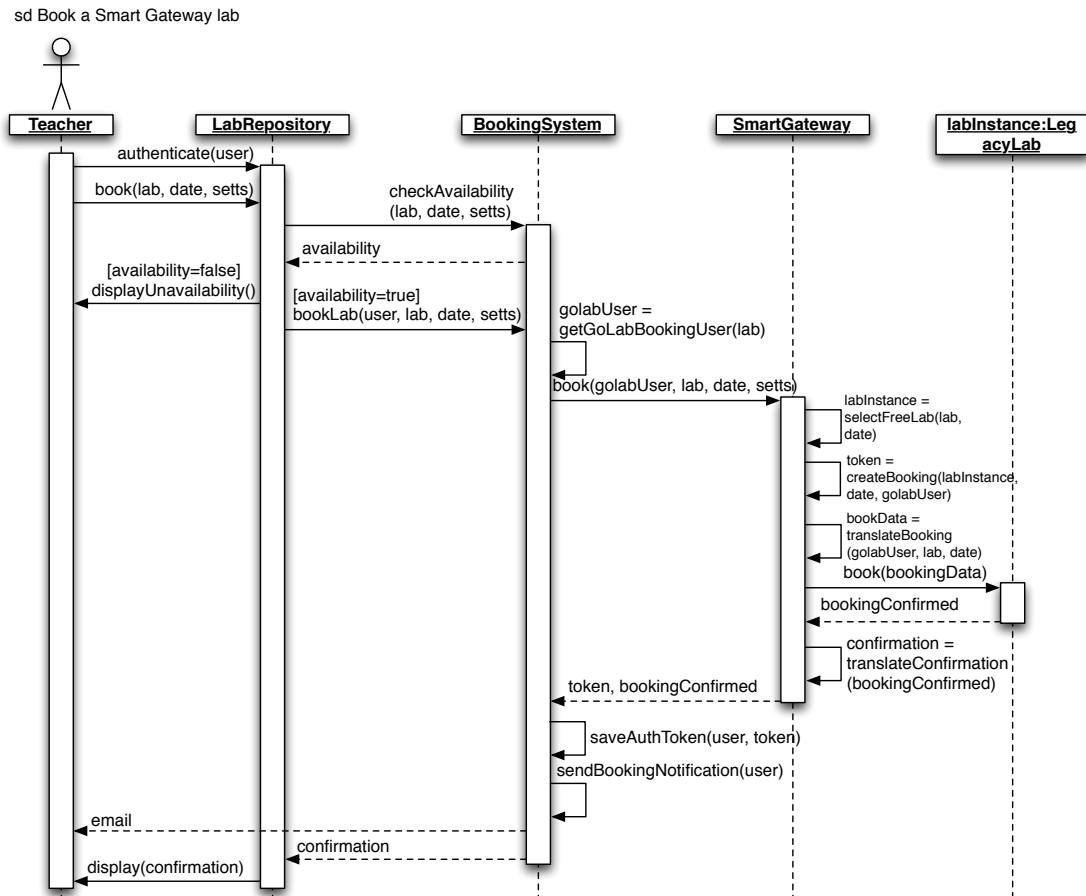
Figure 7.: A sequence diagram that models the booking of a lab using the smart gateway.

the lab is available, the lab can be booked. The integration of the booking with the smart gateway is optional. The lab owner might require such an integration, so we have modelled it as follows. To facilitate an easier implementation of the smart gateway, one lab user is created for each lab that handles the booking. In this way, we do not need to burden the teacher to create a specific account for each lab. With this Go-Lab-wide lab user, the lab is booked via the smart gateway, a specific lab instance is selected that is then booked, which provides an authentication token. Additionally, some translations of the data are made to accommodate the specific booking system of the legacy lab. Finally, the confirmation can be translated if necessary and the booking confirmation is returned together with the authentication token to the booking system. Upon arrival, the authentication token is saved and the notification is returned to the teacher (identically to the smart device mechanism).

Running a booked lab in an ILS: Figure 8 and 9 illustrate the procedure to run a booked lab using, respectively a smart device and the smart gateway, inside an ILS. First, we will discuss the smart device case (see Figure 8).

When a student connects to an ILS in the classroom, she is asked to enter a

sd Use a Smart Device lab
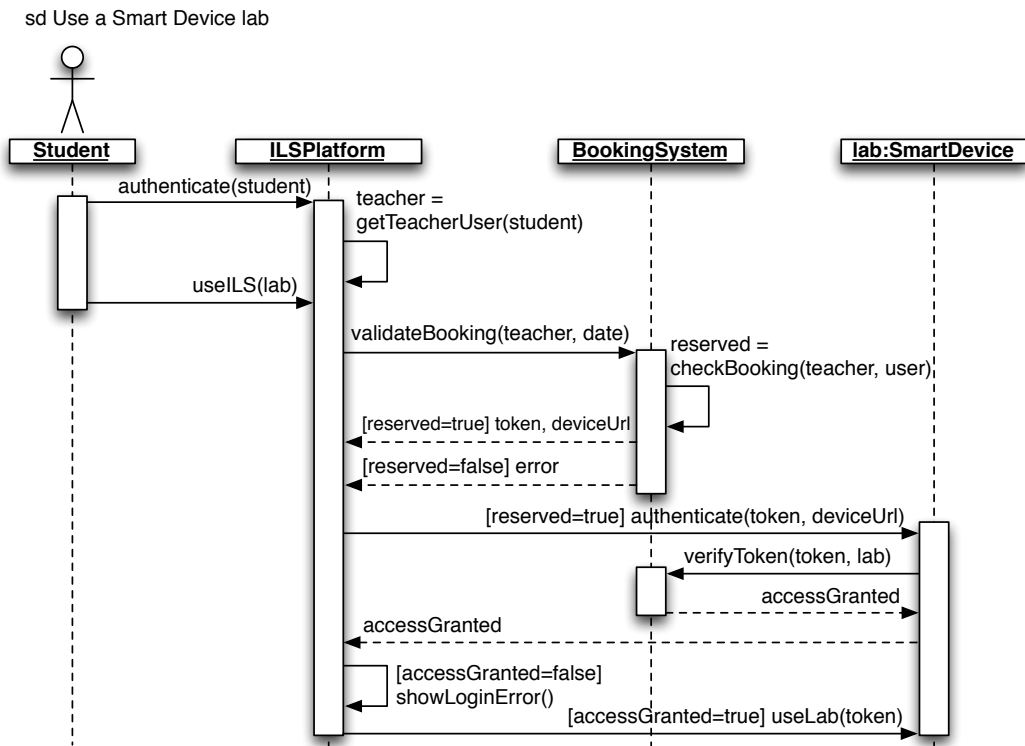
Figure 8.: A sequence diagram that models how a booked lab using a smart device is used in an ILS.
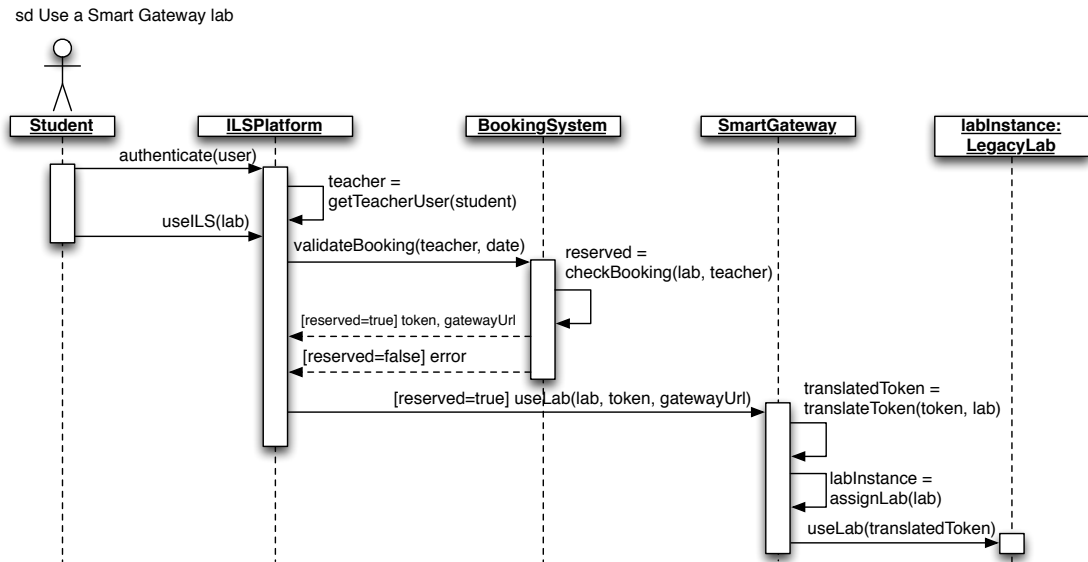
sd Use a Smart Gateway lab

Figure 9.: A sequence diagram that models how a booked lab using a smart gateway is used in an ILS.

nickname (see D5.2 for a discussion on the difference between teacher and student authentication schemes). From the ILS, the teacher user that made the booking can be retrieved. The ILS platform runs the app(s) to operate the remote lab (i.e. one or several OpenSocial gadgets that contain the UI to interact with the remote lab, further referred to as a *lab app*). Thus, the ILS platform validates the booking using the teacher user who made the booking with the booking system. If the booking system confirms the booking, an authentication token and the URL to the booked smart device is sent to the app on the ILS platform. By passing the smart device URL, the booking system can be made responsible of assigning multiple instances (when available) of the same online lab. Now the app can use this token to get access to the experiments of the remote lab through the smart device services. The smart device confirms the token with the booking system. If there was no booking found, an error is displayed to the teacher. In the other case, the user can conduct her experiments with the smart device.

Figure 9 shows the procedure for labs using the smart gateway. The procedure is identical, up till where the lab app on the ILS platform has received the token and connects to the smart device to start the experiment with the token on the smart gateway. After this request, the smart gateway translates this token into an appropriate access token to the legacy lab. Whether the result of this translation is another token or some other form of credentials depends on the requirements of the legacy lab and will differ per legacy lab. Essentially, this translation step provides the interoperability with the legacy lab. Furthermore, the smart gateway has the application logic to select a free instance of the lab in case multiple instances exist and connects the user to the right instance.

Administering lab calendars: Figure 10 illustrates the procedures to administer a lab calendar by lab owners. First, lab owners who want to use the Go-Lab booking system can create a calendar for their lab using the lab repository UI. Note, they have to be logged in. In the booking system, a new calendar is created in the calendar manager and is displayed in the lab repository UI to the lab owner. Such a calendar can be made for every instance of the lab. A user interface could be implemented to simplify this repetitive task, but the creation of each calendar of each instance will be identical to this sequence diagram.

With this lab calendar UI in the lab repository the lab owner can add time slots to this calendar to indicate when she wants to allow Go-Lab teachers to use her lab. The time slot is added to the calendar manager in the booking system and notification is displayed, as well as a confirmation email is sent, to the lab owner to confirm the agreement.

Notifying lab booking progess & Closing a booked lab session: Figure 11 illustrates two functional requirements: the notification of the booking time progress and the closing of the booked lab session. Note that Figure 11 describes what happens after an ILS with a booked lab is being used and the booking ses-

sd Add lab to Go-Lab calendar
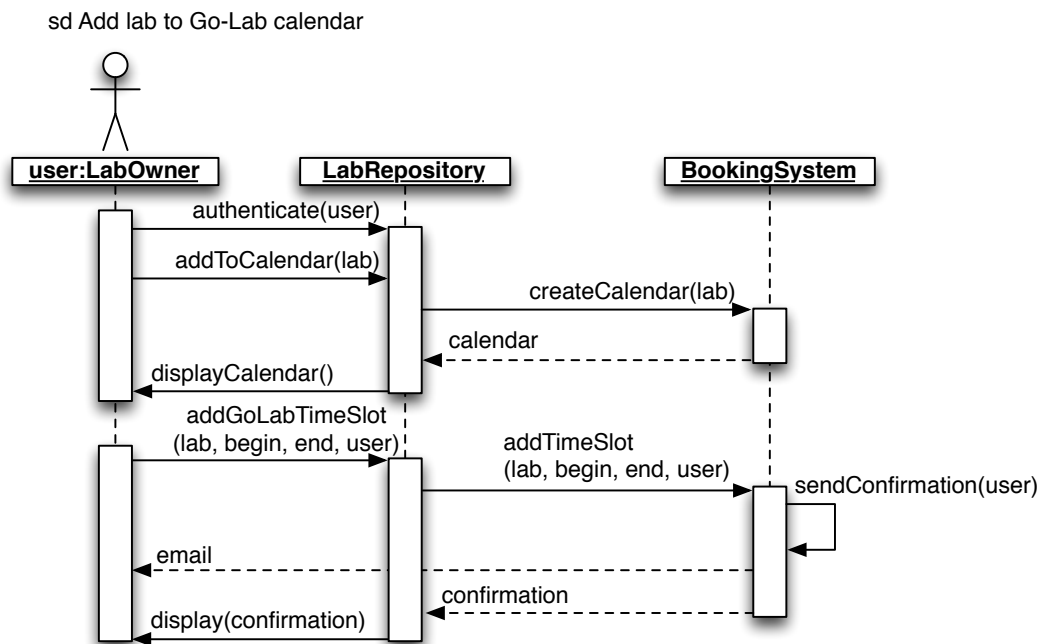
Figure 10.: A sequence diagram that models how a lab owner can administer the booking calendar of a remote lab.

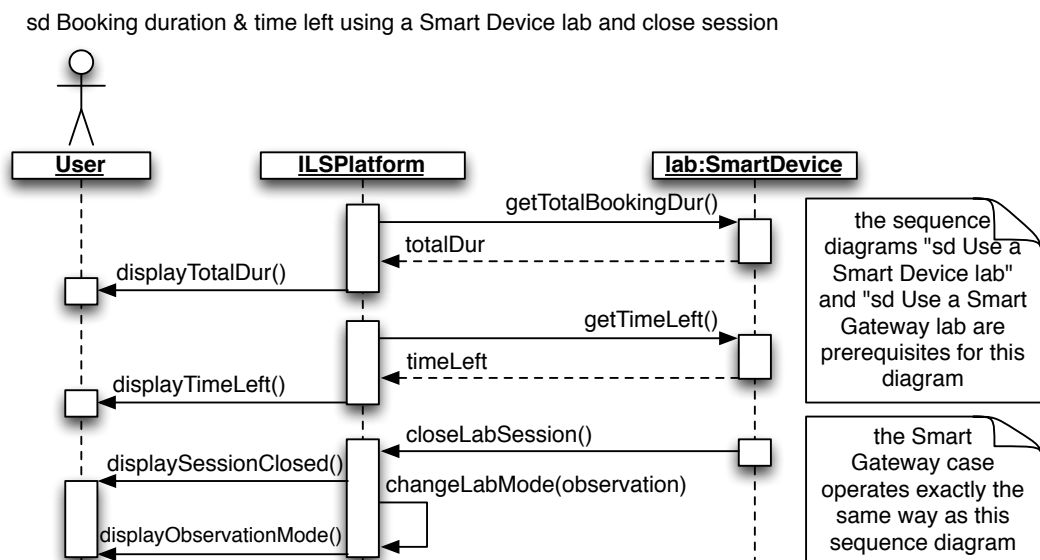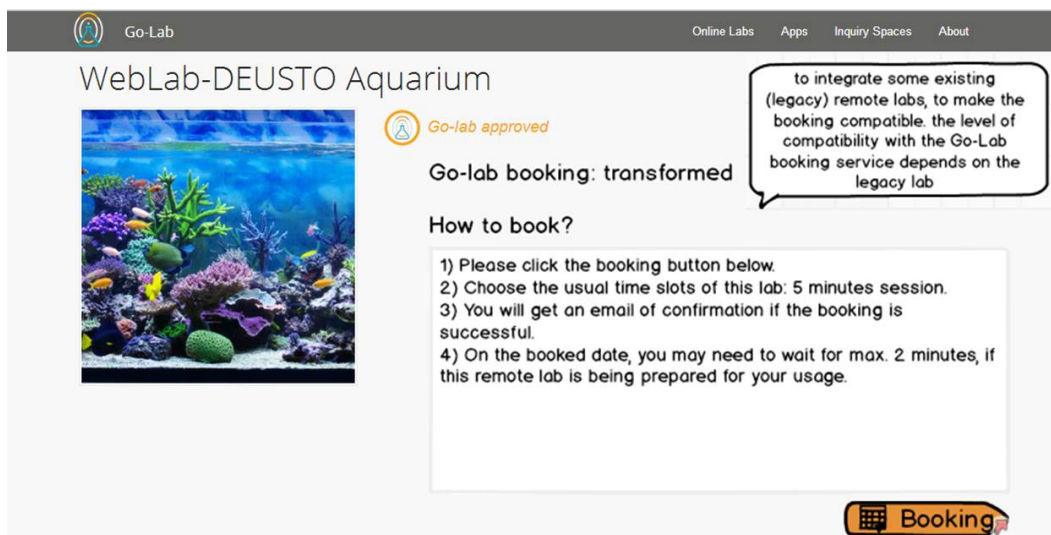sd Booking duration & time left using a Smart Device lab and close session

Figure 11.: A sequence diagram that models how the total booking duration and time left is updated and a booked session is closed.

Figure 12.: A unified user interface to show lab booking information.

sion has started (see the paragraph *Running a booked lab in an ILS*). Once the booked session has started and the ILS is successfully using the lab, the app that operates the lab requests the total booking duration from the smart device and provides the user with an indicator of the total time she can spend operating the lab. Similarly, this lab app can also request the time left to complete the experiment with the online lab and display this information to the user.

When the lab booking time slot has expired, the smart device sends a request to the lab app on the ILS platform to close the lab session. After which, the user is notified and the lab app can change its mode to only allow observation of the remote lab (i.e. when this functionality is provided by the remote lab and the lab app). This would allow the user to observe experiments by her peers.

3.5.2.3. Go-Lab booking system user interface

The Go-Lab booking user interface aims to offer users a unified user interface to book an online lab session. As mentioned before, all labs (even including those with external booking scheme) will provide information on how to book the lab to assist teachers with the lab booking procedure.

Figure 12 illustrates the user interface. To help the teacher, a description of the booking scheme is displayed with the most interesting information for teachers on how to book this lab. In the displayed booking scheme, i.e. *transformed* (cf. Table 3), users can book with the Go-Lab booking system by using the booking calendar.

3.5.3. Architecture and specifications of the Go-Lab bartering platform

Go-Lab Bartering Platform makes users get experts' help and also helps Go-Lab user communities become online lab experts, we call them tutors in Go-Lab. Amongst, the tutor social platform supports the users' acquisition process of online lab knowledge and skills. This section specifies the bartering platform architecture in detail with UML sequence diagrams and implementation plans.

3.5.3.1. *Interface specifications and detailed interaction between components*

This section provides more details on the interaction between the different bartering platform architecture components and the interfaces defined in Figure 5. The main functional requirements of the bartering platform are addressed in the following UML sequence diagrams.
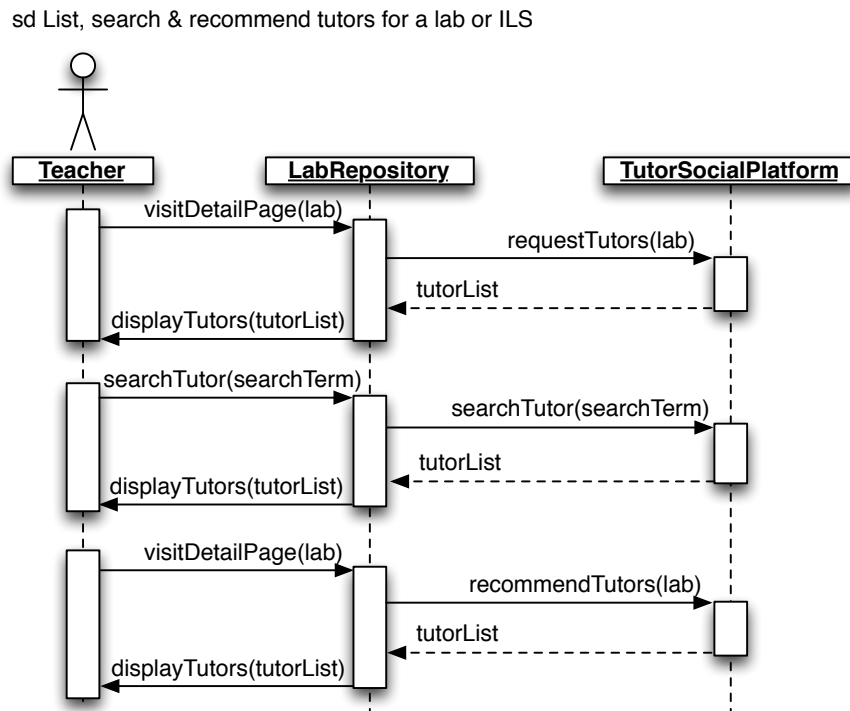
Listing of tutors, Searching tutors & Recommending tutors: Figure 13 illustrates three functional requirements: listing, searching and recommending tutors. First, we will elaborate on listing tutors on a detail page of the lab repository (see D5.2).

When a teacher visits a detail page of a lab (or an ILS, but this is not modelled in Figure 13) on the lab repository, a list of tutors that are available for this lab are retrieved from the tutor social platform and displayed to the teacher in the lab repository UI.

Similarly, for searching a tutor, the teacher's search term is passed to the tutor search component in Figure 5, after which the tutor search results are listed in the lab repository UI. The recommendation case works identically. Based on a lab (or ILS) the `TutorLister` interface returns a list of recommended tutors, which are displayed in the lab repository. The listing and searching case do not have a specific interface modelled in the architecture of Figure 5. This was omitted for clarity in Figure 5, but those interfaces are similar to the `TutorLister`.

Booking tutor time: Figure 14 illustrates how a teacher can book a time slot for a help session with a tutor and pay the tutor credits. To be able to book such a help session, the teacher has to log on to the lab repository, where she can visit the detailed page of a lab (or ILS, but this is not modelled in Figure 14). From the tutor list provided on the lab detail page (see Figure 13 for details), the teacher selects a tutor, for which the tutor profile is retrieved from the profile manager of the tutor social platform. Figure 14 just refers to the Bartering Platform for brevity. This profile is displayed on the lab repository to the teacher, where she clicks a *book* button after selecting an appropriate time slot in the availability calendar of the tutor.

The booking itself consists of different checks. First, the bartering platform checks in its credit system whether the teacher has sufficient credits to pay for this help session. If this is not the case, then the transaction aborts and an error is displayed to the teacher. In case there are sufficient credits, the availability of the tutor at the requested time slot is validated by the tutor booking component using the booking system logic via the `TutorBooker` interface. If the tutor is in the meantime unavailable, the transaction is aborted and the teacher is notified. Only when the tutor is available and there is sufficient credits, the credits are transferred from the teacher to the tutor profile using the credit system. In case an error occurs with this payment, the teacher is notified. Otherwise, the tutor

sd List, search & recommend tutors for a lab or ILS

Figure 13.: A sequence diagram that models the listing, searching and recommending of tutors in the bartering platform.

booking is made and a notification is sent upon success. This notification can be an email and a notification in the lab repository UI. Note, that this booking procedure will be treated as an atomic transaction and the tutor availability will be locked during the transaction to counter double bookings. Moreover, the check on credits will not be included in the first prototype, but will be integrated in the latter development phases (see Section 3.5.3.3).

Other functional requirements: The other functional requirements are not modelled in detail here, since some are quite simple or are already included in Figure 14 (e.g. 'Assigning credits to users' and 'Managing a tutor profile' are partially included). Additionally, some details require further clarification during implementation. For instance, for 'Contacting, bartering, and communicating at tutor time' the communication tools we re-use (e.g. Google Hangouts) will influence the sequence diagrams. More details will be included in the prototype deliverable D4.4 (M24) and the final specifications deliverable D4.5 (M30). Below, we briefly discuss some details of the profile manager and the contact & communication component.

The profile manager of the Social Tutor Platform manages user profiles that can be edited by both owner and any other user. A user profile includes the name, contact information, a short description, and expertise in related online labs and inquiry spaces, and an activity log. All these fields, except the activity log, can be edited by the profile owner. Other users can write comments related to the
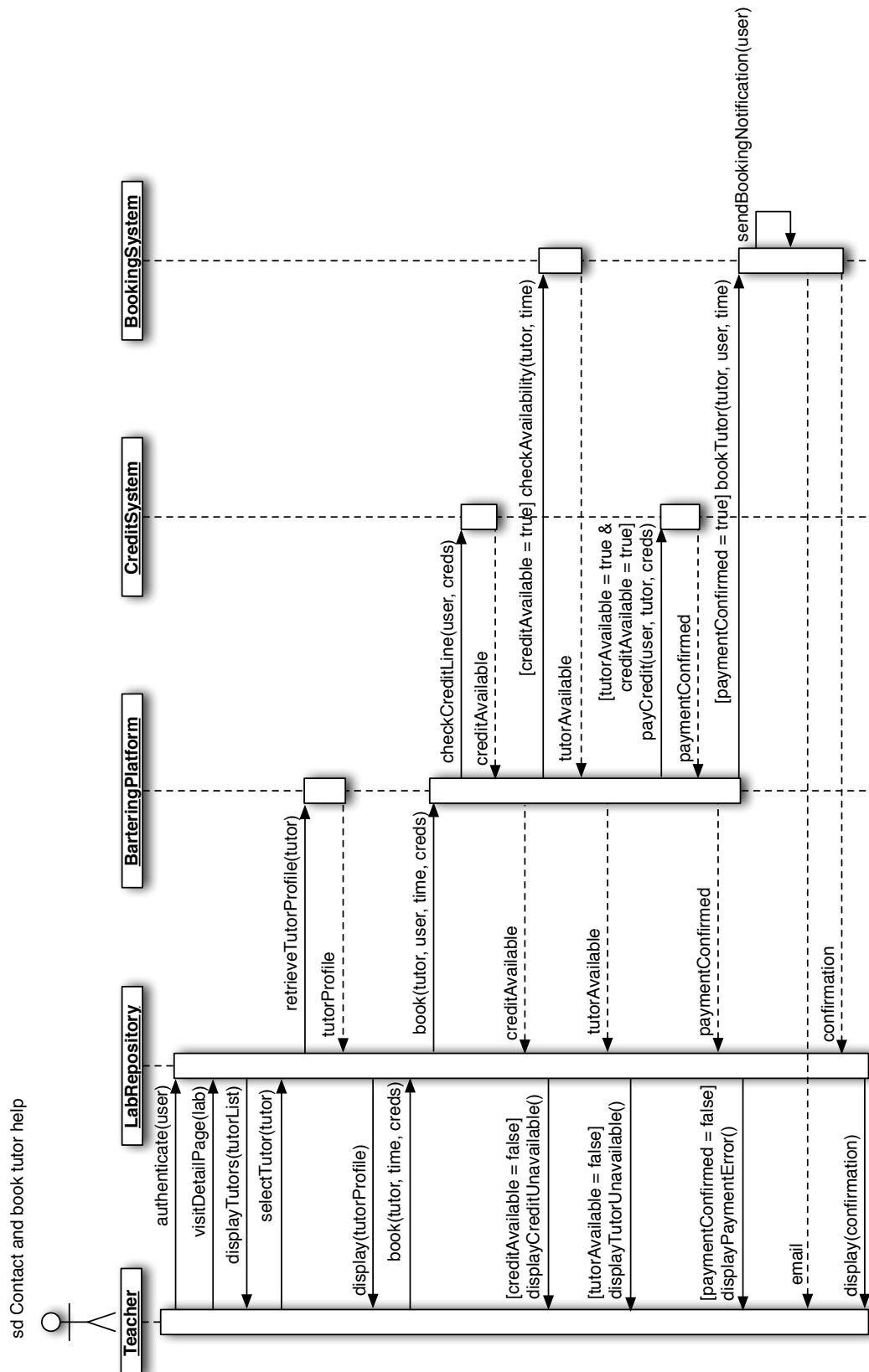
Figure 14.: A sequence diagram that models how to make a booking with a tutor.

Figure 15.: List tutors in Go-Lab Portal.

help session and rate the help sessions and the tutor using a five star rating. The average rating is calculated and listed in each user profile.

The 'contact & communication' component supports various channels to enable communication between users and tutors. Such channels are required to contact, to barter for a help session, and finally to conduct the help session. They comprise emails, contact forms, chat rooms, screen-sharing, and video-chatting; and can be used in combination. For example, one can email a tutor to make an appointment of a help session, while the help session itself is done through the video chatting tool.

3.5.3.2. Go-Lab bartering platform user interface

Go-Lab Bartering Platform user interface is mainly the 'social tutor system' where users (usually teachers) can search tutors, view tutors' profiles, contact and book help sessions. It also provides a tutor list to the Go-Lab Portal. When a registered user views the online labs and inquiry spaces, they may also see the tutor list with their ratings on the right side (see Figure 15). By clicking the individual tutor or the orange button, teachers can access the bartering platform.

Each tutor's profile is managed in the bartering platform. Other users can see the tutor's profile page as depicted in Figure 16. It displays a basic description of a tutor with contact information and average ratings as well as labs and inquiry spaces in which the tutor has expertise. The timeline panel offers a flexible view of tutors' activities in the bartering platform. Tutors can post their help offers in their timeline, which can be further linked to the time schedule for booking.

Tutors will have a centralised booking calendar by clicking the booking button on the upper-right side, which employs the calendar manager in the booking system. Other users (teachers) can comment and rate this tutor. Users' ratings will be calculated as the average rating. Tutors can also decide to re-post some favourite comments by other users or contact activities with teachers.

3.5.3.3. Implementation plan

As an add-on service to support the sustainability of the Go-Lab project and an exploitation strategy platform, the bartering platform employs an implementation plan specified with several development phases. After each phase, a user survey, in coordination with WP3, will be planned to collect improvement feedback for the next phase. Thus, the development process of the bartering platform will be interleaved and adapted in an agile way.

Figure 17 depicts the road map of the bartering platform with information about development phases of different components (cf. the add-on service architecture in Figure 5). Within the Go-Lab project, a free bartering platform supports the teacher community. The paid bartering platform provides an exploitation plan for sustainability beyond the end of the Go-Lab project.

Credit System of the bartering platform   The bartering platform provides on the one hand assistance to teachers that need support to operate online labs and ILS; and on the other hand a social platform for tutors and experts that enables them to improve the visibility of their expertise. The tutor's social profile together with the credit system to award tutors, will enable this visibility. *Implicit* bartering and currency-free bartering will be used to award tutors for the provided help. This bartering is supported by social rating based social media badges. Social media badges indicate tutors' expertise or trust score. For example, quality labels at national and European levels are assigned to motivate teachers in eTwinning [13] and social help platforms, such as the Q&A site StackExchange[14], use social rating mechanisms to rate the best answer and rate the users who provide the answers. Such ratings are then used to compute an overall trust score of the expertise of a *tutor*, which often provides extra motivation for these *tutors* (Chang & Chuang, 2011).

To make this work, a credit system or point system is needed to conduct the bartering process. In the future, the credit system of the bartering platform will further implement the business model of the bartering platform, details of which will be conceptualised and documented in D9.3.

The credit system will be implemented in three phases and is initially optional for the bartering platform. This phase-based development process ensures school teachers to receive tutors' support for free. It will involve lifelong learners gradually through payment, which maintains Go-Lab resources as well. In detail, the implementation plan is as follows, with three phases labelled in Figure 17:

---

[13]eTwinning, `http://www.etwinning.net/`
[14]StackExchange, `http://stackexchange.com/`

Figure 16.: Tutor profile management and display in the bartering platform.
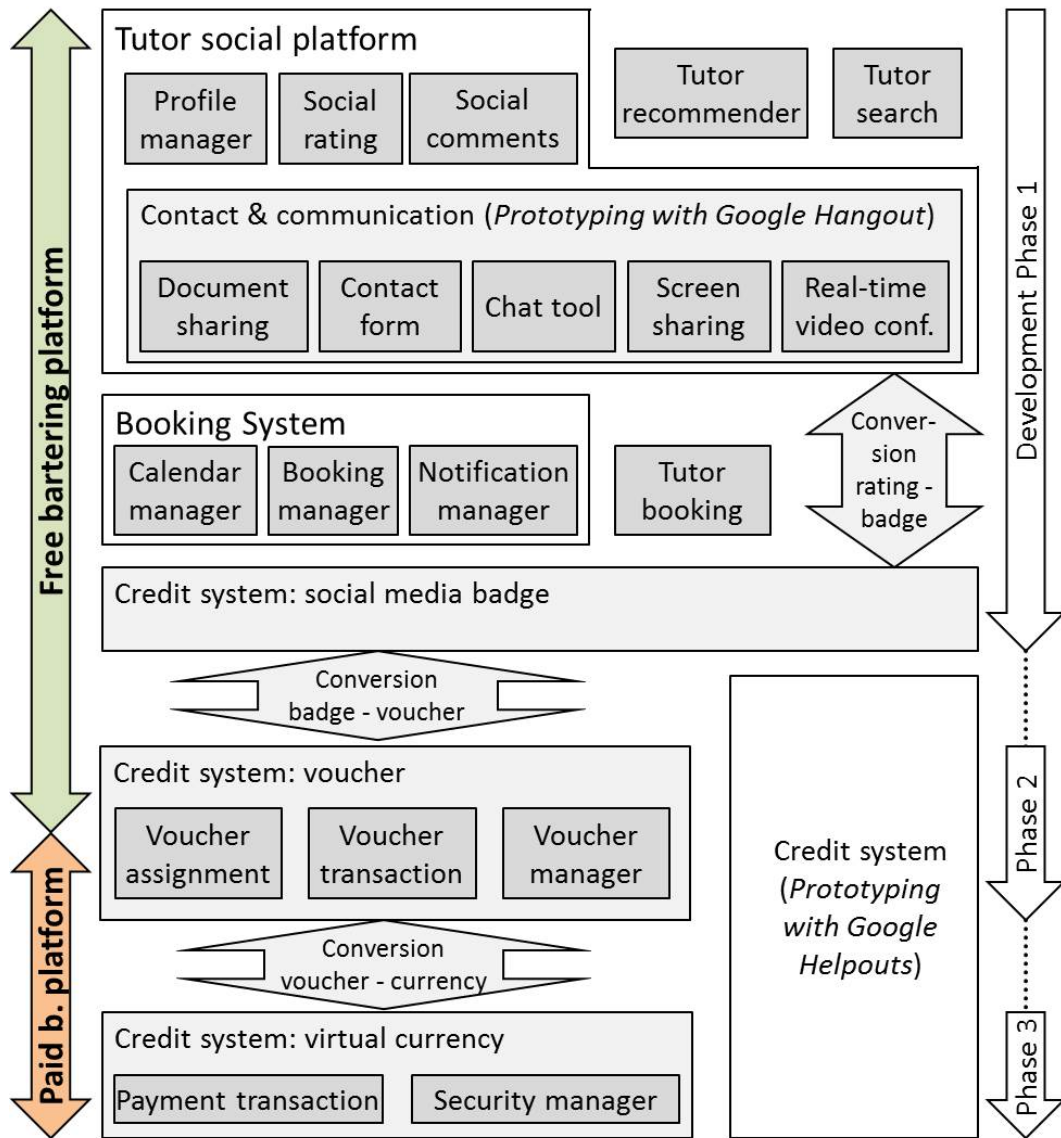
Figure 17.: A road map of development of the bartering platform

- *Phase One* implements a credit system based on social media badges. The bartering platform awards tutors with social media badges based on how many users the tutors have helped and what the social ratings are. Go-Lab bartering platform specifies the conversion from user social ratings to badges. Social media badges only show a tutor's social reputation. They are not used for exchange of tutor help sessions.

- *Phase Two* implements a voucher-based credit system. A voucher represents a kind of credit which is used to exchange help sessions among users and tutors in the bartering platform. A conversion mechanism can be specified between social media badges and vouchers, especially at the initial phase of introduction of vouchers. Users can only *"buy"* help sessions with vouchers they possess. After a Go-Lab user has created a user profile in the bartering platform, she gets a fixed number of vouchers. Users spend vouchers on receiving help sessions. Tutors earn vouchers through offering help sessions. Most important, vouchers are assigned to school teachers for free, while other users (out of schools) get only limited-number of vouchers for free. The credit system does not support monetary voucher exchange, yet.

- *Phase Three* implements a payment-based credit system. Real-world currency is used to buy vouchers and this transaction is supported by the credit system. With this potential business model for the bartering platform, any user (also outside of the Go-Lab community) can join, use and pay for Go-Lab online lab resources. Users' vouchers could also be exchanged to money, while teachers' free vouchers will be sponsored by different sponsorships, e.g. Go-Lab Consortium, national or regional Ministry of Education, or even enterprise sponsoring, and school students' parents. The sponsorship model will be elaborated in WP 9.

Prototyping with Google Hangouts and Helpouts   We have developed some preliminary plans for integration and reuse of existing advanced communication tools. A good video chatting tool is vital for communication in the help session between tutors and users. Preferably, a video chat with the ability to share the tutor's screen so demonstrations can be easily given. The Go-Lab Bartering Platform is not going to develop its own video chatting tool, but applying the existing advanced video chatting tools e.g. Google Hangouts [15], which supports even live group conversation, document sharing, chatrooms, and screen sharing etc. The bartering platform can use Google Hangouts API, if users give their basic Google user account information, i.e. the email address. This is planed for development Phase 1.

A further plan for agile prototyping in Phase 2 and Phase 3 is to use the platform of Google Helpouts. As mentioned in Section 3.4, Google Helpouts does not support special communities. But it reaches a very wide user communities. It also uses Google Wallet for the payment process, which could simplify the

---

[15]http://www.google.com/+/learnmore/hangouts

development process of the credit system of the bartering platform. In summary, Go-Lab Bartering Platform employs Google technologies in order to develop the first prototype with few Go-Lab development resources. If the developed prototype functionality is beneficial to the users, but privacy is a major concern, in-house custom solutions might be investigated based on the remaining project resources.

Participatory design survey for the bartering platform   In order to get feedback from user communities after each development phase, participatory design (PD) surveys are designed to enquire about and evaluate the current prototype and assess whether the user requirement assumptions are correct. This survey will be conducted within some of the planned PD workshops organised by WP3 in year 2. These PD activities will take place after the submission of this deliverable, since year 1 PD activities focused on the Go-Lab portal and year 2 PD activities started only at the beginning of March which did not allow enough time for data collection.

The participatory design workshops for the bartering platform will follow this structure:

1. The Go-Lab Portal is introduced to participants.

2. The participants are asked to conduct participatory design activities related to ILS and online labs related to other WPs.

3. The bartering platform concept and its prototype with basic functionality, (e.g. tutor listing and profiles) are presented.

4. The participants fill in the survey.

5. The participants are asked to complete the remaining participatory design activities related to ILS and online labs.

6. The participants are asked to contact the tutors in Step 3 after the workshop.

After the workshop, the questionnaires will be evaluated. Moreover, the interactions between participants and workshop organisers will be observed, as the practical aspect. Do the participants need help in accomplishing the assigned tasks during the participatory design activities on ILS and online labs? Do the participants contact the tutors? These activities of asking questions to workshop organisers and of contacting tutors are similar to the help sessions taking place in the bartering platform. We would like to see whether there is any conflict between the survey results and participants' interaction. We plan to count the occurrences of help sessions between workshop tutors and participants during and after the workshop. This might provide more insights in how many teachers would actually use the bartering platform.

This process can be repeated in other participatory design workshops in the future, together with the further development progress of Go-Lab Portal, online labs, and the bartering platform. The target groups for the survey will be school

teachers, researchers in research areas of TEL, lab owners, and other online users. Additional questions will be specified for different target groups, based on the general questions in Appendix C.1.

3.6. Summary and Outlook

The Go-Lab Add-on Services bring added-value to the Go-Lab Portal. The Go-Lab Booking System offers diverse booking schemes for remote labs booking. The Go-Lab Booking System provides remote labs a booking system cloud service, if those remote labs don't have sufficient resources to implement their own. Thus, it provides a unified and simplified way to book online labs although they have different booking mechanisms. The Go-Lab Bartering Platform supports the dynamic development of Go-Lab user communities. Users help each other in operating online labs as well as design and use of inquiry spaces. Any user is able to become a tutor for online labs with the growth of their knowledge and skills. This process is validated through social rating and social commenting. Here the most users will be school teachers. Thus, the bartering platform makes the Go-Lab Portal comprise a sustainable market place of knowledge and skills about online labs for teachers. It is a promising solution to supporting teachers' lifelong learning. Teachers' tutoring activities in the bartering platform will add more interactions among teachers to support teachers' activities in Go-Lab Portal.

This section described the initial specification for Go-Lab Add-on Services. Besides this specification, user surveys will be conducted to validate the specified requirements. The surveys will take place in Go-Lab workshops related to academic events in the research area of technology enhanced learning as well as in participatory design workshops with school teachers.

4. Conclusion

In this deliverable we outlined our initial architecture and specifications for learning analytics, scaffolding, and add-on services in Go-Lab. As a basis for these specifications, user scenarios and requirements related to these services and their integration with the Go-Lab portal have been presented.

To support learning analytics and scaffolding in the Go-Lab portal we proposed an open infrastructure for data gathering and analysis processes to assist the different stakeholders and pedagogical scenarios. In addition, we have demonstrated a feedback loop mechanism provided by the learning analytics service, of which a first version has been implemented (see Appendix B). We are currently, discussing with the pedagogical cluster (WP1) how to properly integrate the power of learning analytics in inquiry-based learning and our scaffolding apps. Together with the pedagogical cluster, we will design the solutions to be presented in deliverable D4.4 and will collaborate with WP1 to design scaffolding apps.

For the add-on services we have described the booking and bartering services. To design the booking service, we surveyed existing booking mechanisms and designed three Go-Lab booking schemes to meet the booking requirements of existing remote labs and the needs of teachers. The presented solution will also provide specific help to assist teachers to book labs that will not be integrated with the Go-Lab booking system.

The bartering platform fosters the evolution of specific user communities who have or search for expertise and help for operating online labs or creating inquiry learning spaces. We have proposed several communication channels to provide assistance and a credit system to award tutors for their time. The business model of the bartering platform and its credit system will be further elaborated in WP 9.

Furthermore, some assumptions we took during the requirement analysis of the booking and bartering services will be further validated with teachers and lab owners and when needed we will adjust our specifications. To validate our assumptions, we have designed two surveys: one for the bartering platform and one for the booking system (see Appendix C).

This deliverable is a starting point to implement the learning analytics, scaffolding and add-on services. Part of the implementation has already started (see Appendix B) and the first release of these services will be documented in D4.4 (M24). The specifications will be finalised in D4.6 (M33).

A. Appendix A - The Learning Analytics Feedback Loop

This appendix provides a more detailed technical specification of the data flow in the learning analytics service and feedback loop created by this service. This was already discussed in Section 2.4.5 and Figure 3 above. In Figure 18, a detailed UML sequence diagram is shown providing more details than Figure 3. The next paragraph will briefly describe the sequence diagram.

The sequence diagram actually starts not on the left side as usual but towards the right with *listenTo(appPattern)*. This was done to keep the feedback loop visually clear. Before any app makes use of the learning analytics service, different agents are registered with the shared memory. Such agents subscribe to a specific pattern of user activities that are saved in the learning analytics service.

Once a user executes an app (see on the left side of Figure 18), app subscribes itself via the notification client to the notification broker of the learning analytics service to receive learning analytics based notifications. When a user does an action in the app, the app can log this by sending info on the action and the user to the action logging client, which will transform the action into a ActivityStreams object, which is passed via Shindig (the OpenSocial container used by the ILS platform) to the ILS platform. There the ILS tracking agent, when present, collects all ActivityStreams objects and retrieves some context details, which are together send to the action logging service of the learning analytics service. From here the ActivityStreams object is saved in the data warehouse and the shared memory (before being transformed to a shared memory compatible tuple). Upon arrival in the shared memory, the tuple is matched against all patterns stored by different analysis and notification agents. If a match is found the agent it matches with is triggered and will analyse the tuple asynchronously. After this analysis is finished, the result can be saved back in the shared memory. If this result tuple matches with one of the notification patterns registered by a notification agent, this notification agent will be notified and this agent will create an appropriate message. This message is then sent from the agent to the notification broker to the app, which can display this message to the user.
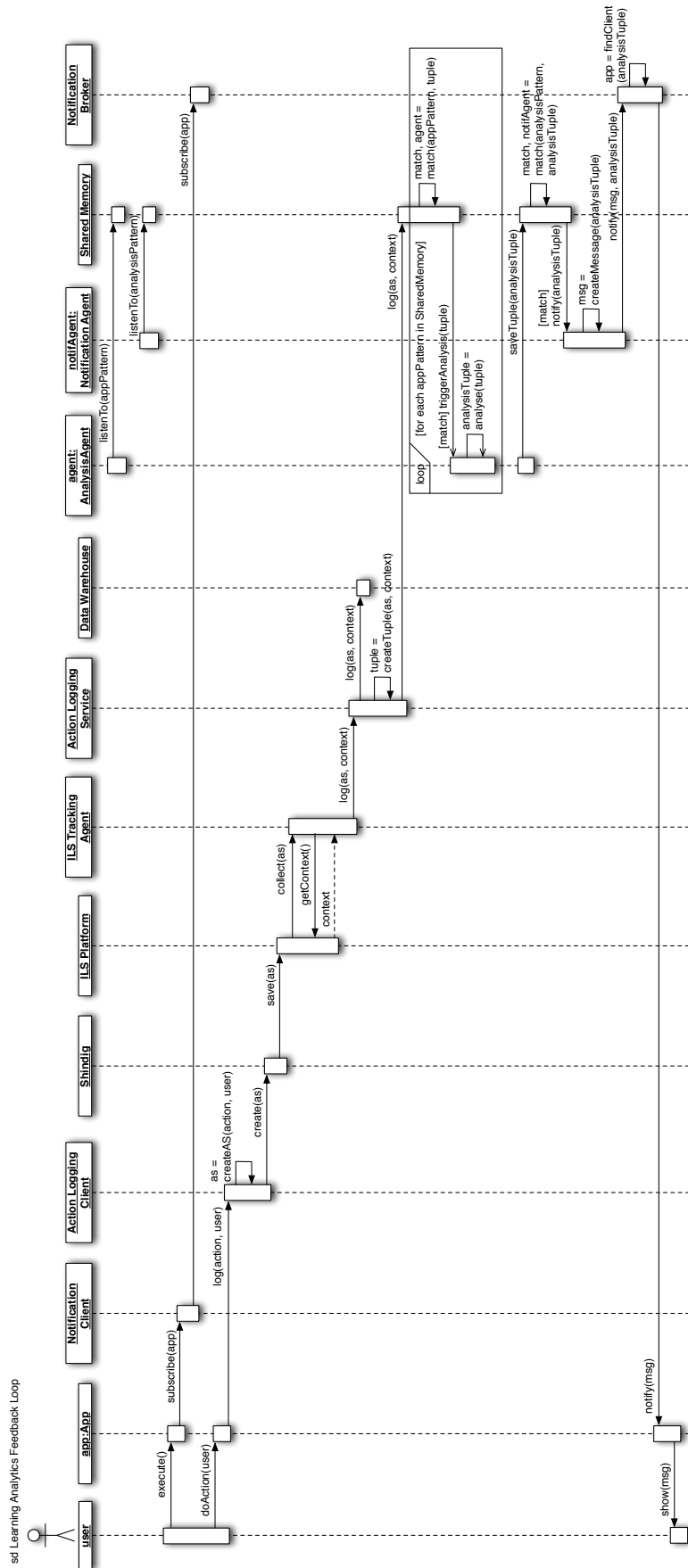
Figure 18.: A sequence diagram that models the data flow in the learning analytics service and demonstrates the feedback loop.

B. Appendix B - Technical implementation of Learning Analytics services

B.1. Introduction

This appendix describes the current state of the technical implementation of the learning analytics infrastructure based on the initial specification. An outlook and future work for the implementation will be given. The work for this deliverable reflects the technical foundation for the next steps on learning analytics, particularly the specification of concrete guidance mechanisms beyond a technical proof-of-concept and the application of learning analytics methods to real data in the Go-Lab ecosystem. The structure of this appendix is as follows: Section B.2 gives an overview of the exposed learning analytics services, which connect the Go-Lab portal to the learning analytics infrastructure. User activities are captured through a logging service, which will be realised on the portal side through AngeLA, the ILS Tracking Agent, which is explained in B.3. Section B.4 lines out the integration of the existing learning analytics workbench and the adaptation to Go-Lab-specific needs. In section B.5 the process necessary to provide a framework for interventions in terms of the learning analytics cycle is described. Each section, provided an outlook on future work for the initial implementation and the final specification.

### B.2. Learning Analytics Backend Services Prototype

A prototypical implementation of the described services of the learning analytics backend is already available as starting point for the further developments. The Action Logging Service described in section 2.4.1.1 accepts action logs encoded in the JSON based activity stream format that are send to the webservice url `http://golab.collide.info/activity`. Those logs are stored in the data warehouse in form of a MongoDB[1] No-SQL database. A web tool that can be used to send example logs to the learning analytics backend can be found under `http://golab.collide.info`. Analytics agents that perform analyses on concept maps and issue notifications to the client, using the Notification Broker are also implemented. This will be described in more detail in appendix section B.5. Further developments will focus on the implementation of more complex analysis and feedback features. This should then include analysis on aggregated datasets, consisting of action logs from different apps enriched with resource metadata gathered from the lab repository using the Artefact Retrieval Service. Such analytics services are currently available and exposed through the Learning Analytics Workbench prototype (cf. B.4), which is also integrated in the Learning Analytics infrastructure. All (prototypical) services are implemented and deployed forming the basis of the learning analytics infrastructure.

B.3. The AngeLA Prototype

This section illustrates the key elements of the Learning Analytics Tracking Agent (AngeLA) functionality implemented until now:

---

[1] `https://www.mongodb.org/`
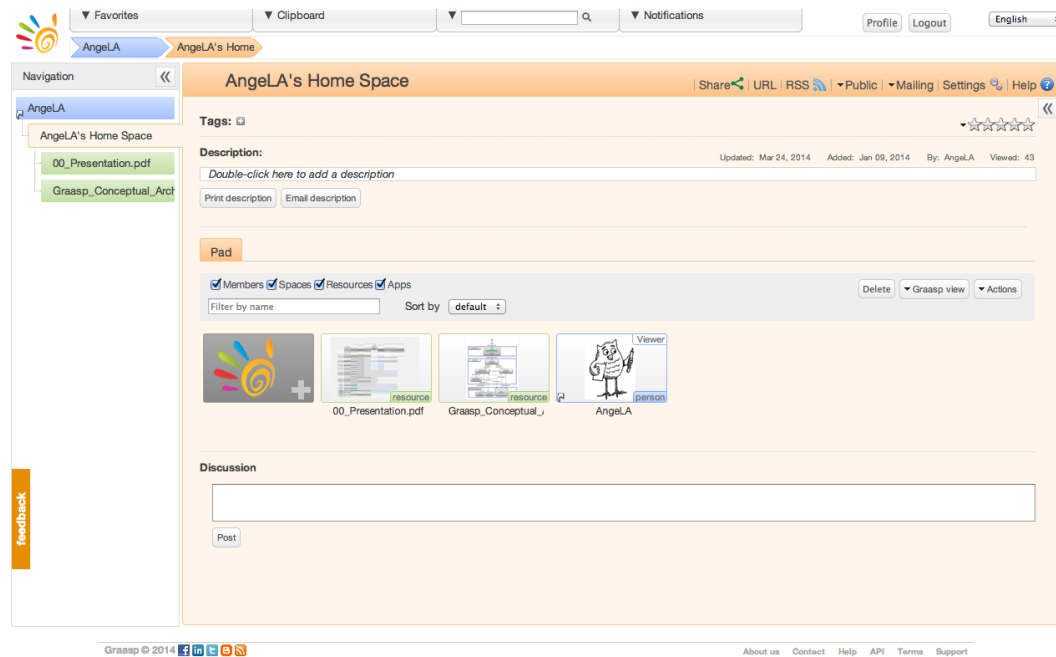
Figure 19.: A screenshot of the Learning Analytics Tracking Agent in a space.

1. **Tracking permission management:** AngeLA aggregates activity logs only from the spaces where it is a member. This provides an easy to use way to manage user tracking permissions. To enable the activity tracking in a space a user just needs to invite AngeLA to this space. Figure 19 shows a screenshot of such a space with AngeLA as a member. When AngeLA is removed from the space, the tracking is disabled for that space. This behaviour is intuitive for the teacher, since the teacher is expecting all members of a space to be aware of activities happening inside.

2. **Cross-space activity data collection:** AngeLA continuously aggregates activity logs of the Graasp users across the spaces where it is a member. The activities are aggregated into a single activity stream.

3. **Data transmission to the Learning Analytics Backend:** All the activity records collected are sent to the LA backend for further processing. The Activity Streams format is used to represent the actions during the transmission.

The functionality presented above was implemented and deployed on the production version of Graasp. The future work includes (1) implementing the default policy for placing AngeLA into the space and (2) collecting and sending the student and teacher actions happening in the Inquiry Learning Space. The latter requires the identities of anonymous students to be represented in Graasp, for instance by implementing support of temporary users.

B.4. Learning Analytics Workbench Prototype

The official version of the workbench described in section 2.4.7 is available for testing under `https://workbench.collide.info/`. The Go-Lab version of the analytics workbench comprises a more Go-Lab oriented set of analysis components and offers the described feature to export workflows as Open Social widgets. It is available at `http://golab.collide.info:9000/`. However, this version is under continuous development and the authors can not guarantee functionality and availability at any time.

Future plans involve the development of more Go-Lab specific analyses, such as filters for interaction analysis and processing of action logs, i.e. sequential pattern mining.

B.5. Learning Analytics Feedback Loop Prototype

This appendix demonstrates and documents the progress on the Learning Analytics Feedback Loop Prototype. The purpose of the Feedback Loop Prototype is to evaluate the specifications and first implementations towards their practicability and integration capability within the overall Go-Lab architecture. This includes:

1. **client-side action logging:** A client-side app is responsible for generating user action logs, as described in deliverable D5.3. Following the ActivityStreams specifications, this includes specifying the "verb" and the "object" of an action. As a next step, this information is passed to an instance of the ActionLogger.

2. **relaying action logs to the LA backend:** The ActionLogger relays an ActivityStream object either directly to the LA backend, or relays it to the tracking agent in the ILS (AngeLA) for further processing (e.g. anonymization) and further relaying. In the future, all ActivityStream objects will be relayed to the ILS tracking agent to enforce our privacy policies.

3. **storing and processing action logs in the LA backend:** Once an action log item arrives at the LA backend (cf. Figure 1), it is passed to the Data Warehouse and to the Shared Memory, where a set of agents analyze the incoming action logs.

4. **generating notifications in the LA backend:** When certain rules or conditions are met, one or more Analysis Agents decide to sent a Notification to the learner. This notification may inform the learner about an important concept which is missing in her concept map.

5. **relaying notifications to the client app:** The Notification Broker in the LA backend relays the notification to the client-side NotificationHandler, which is identified by a combination of user-id, session-id (ILS), and artefact-id.

6. **processing notifications in the client app:** Once the notification reaches the client, it is the app's responsibility to process it and to react accordingly. This includes showing a prompt to the learner in the form of a pop-up dialog, or by changing the configuration of a tool.
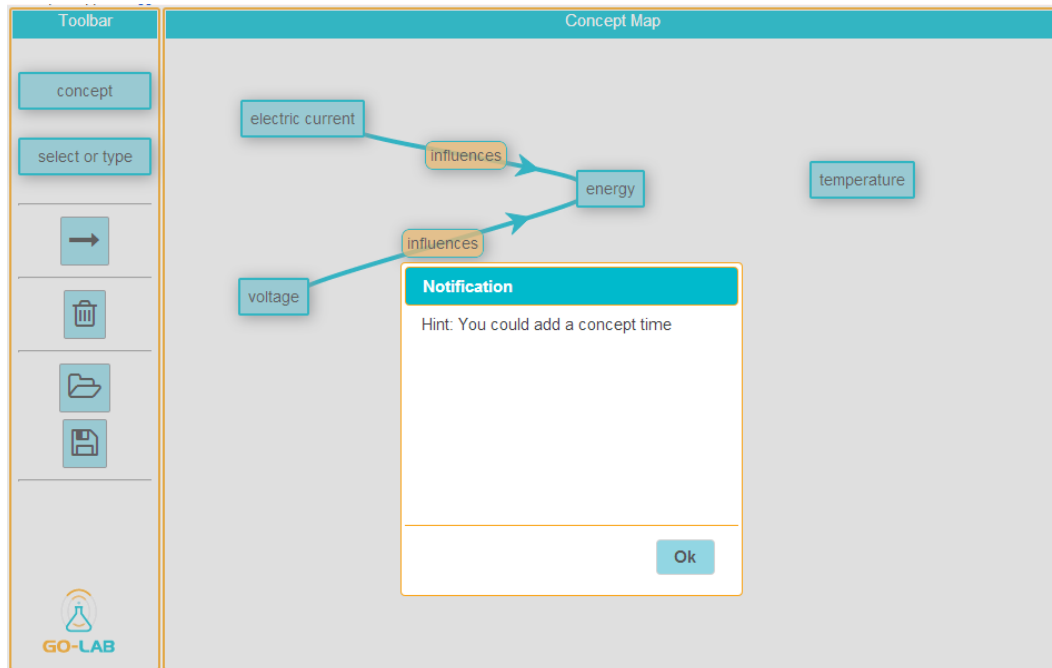
Figure 20.: Screenshot of the Concept Mapper tool receiving a notification from the LA Analysis Agents.

All steps 1-6 as described above have been successfully implemented and tested as a proof-of-concept for the Learning Analytics infrastructure, on client-side and in the backend. Figure 20 shows a screenshot of the Concept Mapping tool, demonstrating the full LA Feedback Loop including action logging, analysis on the LA servers and showing a resulting notification.

Future work includes the generation of pedagogically meaningful intervention rules (in cooperation with the Pedagogical Cluster), which will be implemented as a set of Analysis Agents.

C. Appendix C - Add-on services surveys

C.1. Survey on the Go-Lab Bartering Platform

Dear survey participants, would you like to share your experiences with online laboratories? It takes about 10 minutes. We believe it would be very helpful for us to consider your opinions to improve the development of our Go-Lab Bartering Platform.

Go-Lab Bartering Platform supports peer assistance for teachers to operate online labs and design ILS. Teachers can request help sessions with peers or tutors through different communication channels. Social rating and a credit system will be integrated to reward tutors. Teachers can become tutors while they grow their expertise after being helped.

Please feel free to disseminate this survey: `goo.gl/aIqtJ2`

Thank you for your cooperation!

Yours sincerely

Go-Lab team

Questions on laboratory experiments and online labs

Have you conducted any physical hands-on laboratory experiment by yourself?

For example, have you made a chemical experiment in the chemistry classroom or in an institute laboratory?

(Yes/No)

Are/Were you fond of conducting physical hands-on scientific experiments in the classroom?

(from '1 - I dont́ like it at all' to '5 - I like it very much.')

Is/was it difficult for you to conduct physical hands-on laboratory experiments?

Do you think it difficult for you to conduct physical hands-on laboratory experiments?

(from '1 - Very difficult' to '5 - Very easy')

The use of laboratory experiments influences school students to choose STEM studies as further education topics.

(from '1 - Strongly disagree' to '5 - Strongly agree')

Are you familiar with online labs? Do you know the concept of online labs?

(from '1 - I am not familiar with online labs at all.' to '5 - I am very familiar with online labs.')

Do you think that online labs can replace physical hands-on experiments?

(from '1 - Don't agree' to '5 - Strongly agree')

Do you think that online labs are more difficult to operate/conduct than real lab settings?

(from '1 - Don't agree' to '5 - Strongly agree')

Before participating in this survey, did you know the Go-Lab project?

(Yes/No)

If yes, how did you learn about the Go-Lab project?

- I know Go-Lab from search engines.
- I know Go-Lab from friends and colleagues.
- I know Go-Lab from school teachers.
- I know Go-Lab by some other means.
- Other:

What could be your favourite online labs?

Please give at least one example of the labs you have seen or you wish to have.

Questions on bartering platforms

If you have any issues with online labs or inquiry learning spaces, which kind of help is most useful for you?

Please rank the following help means: (with '1 - absolutely inappropriate', '2 - inappropriate', '3 - neutral', '4 - appropriate', '5 - absolutely appropriate')

- Face-to-face help of an expert
- Personal online meetings to offer help by expert, e.g. via Skype
- A helpdesk
- Online discussion forums
- Social media channels, e.g. twitter, FB, etc.
- Online search
- Other help means

What could be the incentives to motivate users/tutors help other users? (rank with '1 - absolutely inappropriate', '2 - inappropriate', '3 - neutral', '4 - appropriate', '5 - absolutely appropriate')

- Tutors like to share their expertise for free.
- Tutors get social media badges.

- Tutors get other tutors' help sessions in return.
- Tutors get paid.
- Other incentives

Teachers' questions

Which grade levels or students groups do you teach?

On which topic/subject do you teach?

Please specify as detailed as possible, e.g. electromagnetic force in physics.

Do you conduct some experiments physically at your classes?

(Yes/No)

If yes, what kind of experiments are they?

Do you experience problems while conducting some experiments physically in the classroom?

(Yes/No)

It is useful to create an inquiry learning space including online labs.

An inquiry learning space provides a structure for inquiry learning activities with various learning resources like online labs and apps.

(from '1 - absolutely inappropriate' to '5 - absolutely appropriate')

Thank you!

You are welcome to leave your contact information, so that we could send you the survey results.

Your name and email address:

C.2. Lab owner's Survey (A complete survey)

This survey aims to analyse lab owners' needs and requests, including issues related to Go-Lab booking systems. The first part of the survey has been rolled out with `goo.gl/zYcnXt`. The complete version of the questions is as follows:

The Go-Lab Project (Global Online Science Labs for Inquiry Learning at School; www.go-lab-project.eu) is a European project co-funded by the European Commission (Seventh Framework Programme) with 19 partner organisations from twelve countries. Go-Lab aims at providing access to online laboratories in order to enrich classroom experience for Science, Technology, Engineering and Math (STEM) education.

The typical users of Go-Lab will be teachers looking for freely accessible support resources for classroom activities. The students will be able to exploit the online labs selected by their teachers without registration. Lab owners will be able to share their resources openly using the Go-Lab portal. The project targets to reach 10,000 students in Europe.

If a laboratory is included, it will potentially benefit from:

- Higher visibility of the laboratory
- New pedagogic contents and scaffolds
- Increased user base and feedback

IMPORTANT NOTE

This document is neither a binding commitment stating that you will share your labs, nor about obtaining access to your labs. It is just a survey regarding the willingness and the constraints of the lab owner community regarding sharing schemes as envisioned in Go-Lab.

Thank you very much!

Part 1

What laboratories do you have that could fit in the curriculum of primary and secondary schools?

Would you share your labs with the Go-Lab community?

(from '1 - Not likely' to '6 - Very likely')

Would you share your labs to increase their visibility?

(from '1 - Not likely' to '6 - Very likely')

Would you share your labs to get third parties involved to create pedagogical content using the labs?

Example: teachers creating pedagogic contents for your laboratory

(from '1 - Not likely' to '6 - Very likely')

Would you share your labs without requiring direct economic remuneration?

(from '1 - Not likely' to '6 - Very likely')

Under which Remote Laboratory Management System do you have these labs, if any?

- iLabs Shared Architecture
- Labshare Sahara

- WebLab-Deusto

- None

- Other:

Which scheduling mechanism does your system use?

- Calendar

- Queue

- None

- Don't know

- Other:

Does your system support concurrent access by multiple students to the remote laboratory?

Concurrent access: multiple users using the same equipment at the same time

(Yes/No)

References

Alves, G. R., Marques, M. A., Viegas, C., Lobo, C., Barral, R. G., Couto, R. J., . . . Covita, D. S. (2011). Using visir in a large undergraduate course: Preliminary assessment results. In *Global engineering education conference (educon), 2011 ieee* (pp. 1125–1132).

Anjewierden, A. (2012). *Explorations in fine-grained learning analytics* (Unpublished doctoral dissertation). University of Twente.

Blattner, M., Glinert, E., Jorge, J., & Ormsby, G. (1992). Metawidgets: towards a theory of multimodal interface design. In *Computer software and applications conference, 1992. compsac'92. proceedings., sixteenth annual international* (pp. 115–120).

Breuer, R., Klamma, R., Cao, Y., & Vuorikari, R. (2009). Social network analysis of 45,000 schools: A case study of technology enhanced learning in europe. In U. Cress, V. Dimitrova, & M. Specht (Eds.), *Learning in the synergy of multiple disciplines* (Vol. 5794, p. 166-180). Springer Berlin Heidelberg. Retrieved from `http://dx.doi.org/10.1007/978-3-642-04636-0_18` doi: 10.1007/978-3-642-04636-0_18

Cao, Y., Hannemann, A., Klamma, R., Kovachev, D., & Renzel, D. (2010). Mobile multimedia management for community-aware storytelling. In T. Hara, C. S. Jensen, V. Kumar, S. Madria, & D. Zeinalipour-Yazti (Eds.), *Mobile data management* (p. 59-64). IEEE Computer Society.

Chang, H. H., & Chuang, S.-S. (2011, Jan). Social capital and individual motivations on knowledge sharing: Participant involvement as a moderator. *Information and Management*, *48*(1), 9–18. doi: 10.1016/j.im.2010.11.001

Chung, L., & Prado Leite, J. C. (2009). Conceptual modeling: Foundations and applications. In A. T. Borgida, V. K. Chaudhri, P. Giorgini, & E. S. Yu

(Eds.), (pp. 363–379). Berlin, Heidelberg: Springer-Verlag. Retrieved from `http://dx.doi.org/10.1007/978-3-642-02463-4_19` doi: 10.1007/ 978-3-642-02463-4_19

Clariana, R., Engelmann, T., & Yu, W. (2013). Using centrality of concept maps as a measure of problem space states in computer-supported collaborative problem solving. *Educational Technology Research and Development*, *61*(3), 423-442. Retrieved from `http://dx.doi.org/10.1007/ s11423-013-9293-6`

Clow, D. (2012). The learning analytics cycle: Closing the loop effectively. In *Proceedings of the 2nd international conference on learning analytics and knowledge* (pp. 134–138). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/2330601.2330636` doi: 10.1145/2330601 .2330636

de Jong, T., van Joolingen, W. R., Giemza, A., Girault, I., Hoppe, U., Kindermann, J., ... van der Zanden, M. (2010). Learning by creating and exchanging objects: the SCY experience. *British Journal of Educational Technology*, *41*, 909–921.

De Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (2006, January). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Comput. Educ.*, *46*(1), 6–28. Retrieved from `http://dx.doi.org/10.1016/j.compedu.2005.04.005` doi: 10.1016/ j.compedu.2005.04.005

Duval, E. (2011). Attention please! learning analytics for visualization and recommendation. In *1st international conference on learning analytics and knowledge (lak 2011)* (pp. 9–17). Banff, Canada: ACM.

Facca, F. M., & Lanzi, P. L. (2005). Mining interesting knowledge from weblogs: a survey. *Data Knowl. Eng.*, *53*(3), 225-241.

Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, *4*(5), 304–317.

Few, S. (2007). Dashboard confusion revisited. *Perceptual Edge*.

Fortenbacher, A., Beuster, L., Elkina, M., Kappe, L., Merceron, A., Pursian, A., ... Wenzlaff, B. (2013, Sept). Lemo: A learning analytics application focussing on user path analysis and interactive visualization. In *Intelligent data acquisition and advanced computing systems (idaacs), 2013 ieee 7th international conference on* (Vol. 02, p. 748-753). doi: 10.1109/IDAACS .2013.6663025

Gelernter, D. (1985). Generative communication in linda. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, *7*(1), 80–112.

Giemza, A., Weinbrenner, S., Engler, J., & Hoppe, H. U. (2007). Tuple spaces as a flexible integration platform for distributed learning environments. In *Proceedings of the 2007 conference on supporting learning flow through integrative technologies* (pp. 313–320). Amsterdam, The Netherlands, The Netherlands: IOS Press. Retrieved from `http://dl.acm.org/ citation.cfm?id=1565478.1565541`

Gillet, D., de Jong, T., Sotirou, S., & Salzmann, C. (2013, March). Personalised learning spaces and federated online labs for stem education at school.

In *Global engineering education conference (educon), 2013 ieee* (p. 769-773). doi: 10.1109/EduCon.2013.6530194

Göhnert, T., Harrer, A., Hecking, T., & Hoppe, H. U. (2013). A workbench to construct and re-use network analysis workflows - concept, implementation, and example case. Niagara Falls, Canada.

Govaerts, S., Cao, Y. C., Vozniuk, A. V., Holzer, A., Garbi Zutin, D., San Cristobal Ruiz, E., ... Gilet, D. (2013). Towards an online lab portal for inquiry-based stem learning at school. In *Proceedings of the 12th international conference on web-based learning (to appear, accepted 06/2013).*

Harrer, A., Malzahn, N., Zeini, S., & Hoppe, H. U. (2007). Combining social network analysis with semantic relations to support the evolution of a scientific community. In C. Chinn, G. Erkens, & S. Puntambekar (Eds.), *Mice, minds, and society - the computer supported collaborative learning (cscl) conference 2007* (p. 267-276). International Society of the Learning Sciences.

Harward, V., del Alamo, J., Lerman, S., Bailey, P., Carpenter, J., DeLong, K., ... Zych, D. (2008). The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories. *Proceedings of the IEEE*, *96*(6), 931–950.

Hoppe, H. U., Engler, J., & Weinbrenner, S. (2012, 07/2012). The impact of structural characteristics of concept maps on automatic quality measurement. Sydney, Australia.

Inmon, W. H. (1992). *Building the data warehouse*. New York, NY, USA: John Wiley & Sons, Inc.

Kraan, W., & Sherlock, D. (n.d.). Analytics tools and infrastructure.

Laat, M. D., Lally, V., Lipponen, L., & Simons, R.-J. (2007). Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for social network analysis. *I. J. Computer-Supported Collaborative Learning*, *2*(1), 87-103.

Liddo, A. D., Shum, S. B., Quinto, I., Bachler, M., & Cannavacciuolo, L. (2011). Discourse-centric learning analytics. In *Lak* (p. 23-33).

Limpens, F., & Gillet, D. (2011). A competence bartering platform for learners. In H. Leung, E. Popescu, Y. Cao, R. W. H. Lau, & W. Nejdl (Eds.), *Icwl* (Vol. 7048, p. 148-153). Springer.

Lowe, D., & Orou, N. (2012). Interdependence of booking and queuing in remote laboratory scheduling. In *Remote engineering and virtual instrumentation (rev), 2012 9th international conference on* (pp. 1–6).

Orduña, P. (2013). *Transitive and scalable federation model for remote laboratories* (Doctoral dissertation, Universidad de Deusto, Bilbao, Spain). Retrieved from `http://paginaspersonales.deusto.es/porduna/phd/`

Orduña, P., & García-Zubia, J. (2011). Scheduling schemes among internet laboratories ecosystems.

Orduña, P., Irurzun, J., Rodriguez-Gil, L., García-Zubia, J., Gazzola, F., & López-de Ipiña, D. (2011). Adding new features to new and existing remote experiments through their integration in WebLab-Deusto. *iJOE*, *7*(S2), 33-39.

Orduña, P., Rodriguez-Gil, L., Angulo, I., Dziabenko, O., López-de Ipina, D., & García-Zubia, J. (2012). Exploring students collaboration in remote laboratory infrastructures. In *Remote engineering and virtual instrumentation (rev), 2012 9th international conference on* (pp. 1–5).

Orduña, P., Rodriguez-Gil, L., López-de Ipiña, D., & Garcia-Zubia, J. (2013). Sharing remote labs: A case study. *International Journal of Online Engineering*, *9*.

Ponta, D., Scapolla, A., & Buschiazzo, P. (2009). Survey of remote laboratories using service oriented architectures. *International Journal of Online Engineering (iJOE). ISSN*, *2121*.

Safaric, R., Truntič, M., Hercog, D., & Pačnik, G. (2005). Control and robotics remote laboratory for engineering education. *International Journal of Online Engineering (iJOE)*, *1*(1).

Siemens, G. (2008). Learning and knowing in networks. changing roles for educators and designers. In *ITFORUM for discussion* (p. 1-26).

Siemens, G. (2012). Learning analytics: envisioning a research discipline and a domain of practice. In *Proceedings of the 2nd international conference on learning analytics and knowledge* (pp. 4–8). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/2330601.2330605` doi: 10 .1145/2330601.2330605

Siemens, G., Gasevic, D., Haythornthwaite, C., Dawson, S., Shum, S. B., Ferguson, R., . . . Baker, R. (2011). Open learning analytics: an integrated & modularized platform. *Proposal to design, implement and evaluate an open platform to integrate heterogeneous learning analytics techniques*.

Sousa, N., Alves, G., & Gericota, M. (2010). An integrated reusable remote laboratory to complement electronics teaching. *Learning Technologies, IEEE Transactions on*, *3*(3), 265–271.

Southavilay, V., Yacef, K., Reimann, P., & Calvo, R. A. (2013). Analysis of collaborative writing processes using revision maps and probabilistic topic models. In *Lak* (p. 38-47).

Suthers, D. D., Lund, K., Rosé, C. P., Teplovs, C., & Law, N. (2013). Productive multivocality in the analysis of group interactions. *New York, NY: Springer. CSCL book series. doi*, *10*(1007), 978–1.

Verbert, K., Govaerts, S., Duval, E., Santos Odriozola, J. L., Van Assche, F., Parra Chico, G. A., & Klerkx, J. (2013). Learning dashboards: an overview and future research opportunities. *Personal and Ubiquitous Computing*, 1–16. Retrieved from `https://lirias.kuleuven.be/handle/123456789/440777` (Accepted) doi: 10.1007/s00779-013-0751-2

Vozniuk, A., Govaerts, S., & Gillet, D. (2013). Towards portable learning analytics dashboards. In *Proceedings of the 13th IEEE International Conference on Advanced Learning Technologies.*

Weinbrenner, S. (2012). *Sqlspaces: a platform for flexible language heterogeneous multi-agent systems*. München: Hut. (Zugl.: Duisburg, Essen, Univ., Diss., 2012)

Zaıane, O. R., & Luo, J. (2001). Web usage mining for a better web-based learning environment. In *Proceedings of conference on advanced technology*

for education (pp. 60–64).

Zhang, J., Ackerman, M. S., & Adamic, L. (2007). Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on world wide web* (pp. 221–230). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/1242572.1242603` doi: 10.1145/1242572.1242603