Chapter 4
The iTEC Technical Artefacts, Architecture and Educational Cloud

Frans Van Assche, Luis Anido-Rifón, Jean-Noël Colin, David Griffiths, and Bernd Simon

Abstract This chapter introduces the technical artefacts of the iTEC project in the context of a cloud architecture. The rationale for the technology developed in the iTEC project follows from its overall aim to re-engineer the uptake of ICT in schools. To that end, iTEC focused (a) on some important barriers for the uptake of ICT such the effort that teachers must make in redesigning their teaching and finding the right resources for that, and (b) on enablers for the uptake of ICT, such as providing engaging experiences both for the learner and teacher. The technical innovations are centred around three themes: innovations in the support of learning design, innovations by using a-typical resources, and innovations in the integration and management of learning services and resources. Next this chapter presents the cloud architecture adopted by all technology providers, including a shared user management and control system, the shared data models and interoperability solutions. The technical artefacts and then further elaborated in the ensuing chapters.

Keywords Uptake of ICT • Schools • Technical architecture • Authentication • Authorisation • Exchange protocols

F. Van Assche (✉)
Department of Computer Science, University of Leuven, Leuven, Belgium
e-mail: frans.van.assche@gmail.com

L. Anido-Rifón
Telematics Engineering Department, ETSI Telecommunication, University of Vigo, Vigo, Spain
e-mail: lanido@det.uvigo.es

J.-N. Colin
University of Namur, Namur, Belgium
e-mail: jean-noel.colin@unamur.be

D. Griffiths
Institute of Educational Cybernatics, University of Bolton, Bolton, UK
e-mail: D.E.Griffiths@bolton.ac.uk

B. Simon
Knowledge Markets Consulting G.m.b.H., Wien, Austria
e-mail: bernd.simon@km.co.at

© The Author(s) 2015

F. Van Assche et al. (eds.), *Re-engineering the Uptake of ICT in Schools*,
DOI 10.1007/978-3-319-19366-3_4

Rationale for the Educational Cloud and Technical Artefacts

Whereas Chap. 1 elaborates the rationale for re-engineering the uptake of ICT in schools, this section introduces the choice of artefacts developed in iTEC, the architecture for these artefacts, and how these fit together in what we call the iTEC Educational Cloud (IEC).

Barriers to the mainstreaming of technologies have been studied since the beginning of TEL. For example the first large scale European project about TEL in schools (Van Assche 1998) reported already the limited time of teachers, teacher training, the curriculum, etc. Other research added lack of teacher confidence (teachers being scared and intimidated by their student's increasing knowledge about Internet and communication devices), lack of pedagogical teacher training; lack of suitable educational software, limited access to ICT; rigid structure of traditional education systems, etc.

However, as many practitioners will testify (e.g. see in Van Assche 1998; Van Assche et al. 2006), the barrier most mentioned is the burden to teachers (often expressed as lack of time) when they have to explore and absorb emerging technologies. This in turn seems to influence other cited problems. Therefore, iTEC decided to explore how teachers can be helped in the following three areas.

Firstly, we noted that teachers reported in earlier projects that they spend most of their time, apart from contact hours in the classroom, in lesson preparation and assessment. The introduction of new technologies increases the burden by requiring established lesson plans to be revised, and by introducing elements into the planning process whose implications for the classroom process are unknown to teachers. iTEC sought to alleviate this problem by providing support in carrying out lesson planning which involved new technologies. An investigation with Ministries of Education (MoE) revealed that many countries and regions have lively teacher communities that exchange lesson plans and ideas. For example the lektion.se community in Sweden alone has more than 220,000 members. However, the challenge is to share lesson plans and ideas across national and regional boundaries. Therefore, iTEC decided to explore how **de-contextualized learning designs** (including lesson plans)—in iTEC called scenarios—could make ideas and elaborated designs more shareable. In addition, de-contextualisation would facilitate the introduction of emerging technologies without the need to refer to specific products. This was achieved by providing requirements for a lesson plan in an intentional way instead of an extensional way,[1] which has the additional advantage of making the requirements more resilient to changing technologies. The intentional way means that for example the scenarios refer to kinds of resources in a descriptive way, while the lesson plan will typically refer to specific resources.

Secondly, iTEC investigated how **learning can be made more engaging** by providing non-traditional resources through the use of ICT. While, the ambient

[1] "Intension" indicates the internal content of a term or concept that constitutes its formal definition; and "extension" indicates its range of applicability by naming the particular objects that it denotes.

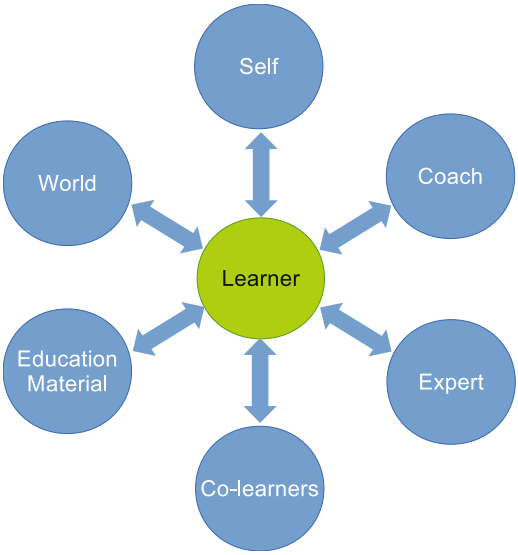

Fig. 4.1 Interactions of the learner

intelligent vision from 2001 (see Chap. 1) was unrealistic, it was indicative of a shift to different forms of more learner-centred, ICT-facilitated approaches including personal learning, individual learning, self-regulated learning, and ambient schooling (Van Assche 2004). Within such a learner-centred approach the levers for engagement come from interactions. The learning experience can only be influenced through interactions, and it is at these points of contact that we seek to identify the opportunities for creating and facilitating engagement. These opportunities are summarised in Fig. 4.1.

Typically a learner interacts with a coach (usually the teacher), a subject expert (usually the teacher), co-learners, education material, the world outside the closed educational environment, and with the traces of their own earlier activities. In this context of interactions, iTEC exploited the fact that ICT provides the means to go beyond the classroom setting. For example to be able to chat with an astronaut about space travel, participate in a distant experiment in CERN, get coaching support from a grandmother living a 100 km away, have access to simulation and serious games, and consult same-age learners abroad about how to pronounce a foreign language. As such, engagement can arise from the person, material, or environment one interacts with and/or the interaction conduit itself. Again from the early Web for Schools project up to recent TEL projects such as the Stellar project, research has pointed to the engaging potential of ICT.[2] iTEC therefore explores to what extent interactions other than the traditional classroom interactions can possibly enhance engagement.

[2] In the Stellar 'Big Meeting' of February 2012 there was only one factor mentioned by all business stakeholders: the engagement potential of TEL.

Thirdly, iTEC tackled the substantial **burden that comes with the integration** of emerging technologies. Whereas innovators and early adopters are prepared to put up with a range of integration problems, these are a real barrier for the early majority, the late majority, and laggards. If we want to cross the mainstreaming chasm, it is essential to reduce the integration burden. This burden originates from the lack of interoperability between platforms and applications running on these platforms as well as between applications themselves. iTEC aimed to provide easy integration for at least 50 % of the installed platforms for education including container technologies such as the Virtual Learning Environments Moodle[3] and DotLRN,[4] and for the interactive whiteboard software OpenSankoré.[5]

Given these three areas in which interventions can be made to improve the uptake of ICT in schools, the iTEC artefacts can be presented, together with their rationales:

- *Ready-made scenarios*: iTEC created a set of scenarios (i.e. de-contextualised structured narrative learning designs) that aim to help teachers to go beyond their usual classroom activities and to explore emerging technologies. iTEC proposes that if teachers are provided with examples of effective use of new technologies, it will be easier for them to start using such new technologies in their own classes. These scenarios are adapted by teachers to their own local context.
- *Ready-made learning activities*: Learning stories consist of learning activities and are further elaborations of scenarios as concrete instantiations whose purpose is to make the resource (material, people, events) requirements more concrete. By providing different levels of abstraction, teachers and learners can choose the appropriate level for their purpose.
- *A Future Classroom Scenario Method*: As iTEC was concerned with systemic change, it also created a method with procedures and techniques for developing such scenarios. An important part of this toolkit is the Future Classroom Maturity Model (see Chap. 2) that allows teachers, head-teachers, ICT co-ordinators, and MoE to assess where they are with respect to four innovation dimensions, and develop scenarios that facilitate taking the next step.
- *The Learning Activity Design Method*, that guides teachers in how to find and use an archive of Learning Stories and Learning Activities which are derived from iTEC scenarios. It is focused on enabling the adoption of advanced pedagogical approaches by teachers, supported by appropriate technologies and other resources. The Learning Activity Design Toolkit is used by individual teachers and collaborative communities.
- *A Widget Store*: The iTEC Widget Store provides access to a collection of small ready-to-use educational apps that can be deployed in a range of ‘shells’ which act as containers for widgets (see also later). The W3C specification for widgets was adopted in order to maximise interoperability, and support is provided for embedding widgets from the iTEC Store in Moodle, DotLrn, OpenSankoré, and even ordinary browsers.

[3] https://moodle.org/

[4] http://dotlrn.org/

[5] http://open-sankore.org/

- *A number of technical artefacts, including services and specifications*: These artefacts, elaborated in the next section, offer, inter alia, automated help in finding adequate resources, activities, and scenarios; automated support for localisation; finding more easily other types of resources such as people and events; play applications in the form of widgets; plug and play authentication and authorisation; support in establishing new collaborations, and last but not least the iTEC Educational Cloud (IEC).

All these iTEC artefacts have a **common characteristic**: facilitating the uptake of ICT in schools. However, the benefits are not restricted to this. For example some of the technical artefacts (see next section) are also beneficial to technology providers, standardization bodies, researchers, etc.

Technical Artefacts

In this section we focus on the *technical* artefacts. These artefacts primarily aim to support teachers in their learning design and assessment activities. A typical workflow is that the teacher selects an iTEC scenario, and then defines a number of learning activities based upon the scenario which together constitute a learning story. When the teacher finally puts the learning story into practice, the system assists in translating abstract requirements into concrete resources, that fit her pedagogical goals. While describing the technical artefacts, the innovations are highlighted.

Innovations in Support for Learning Design[6]

The aim of this iTEC technology is to support teachers in discovering the opportunities and limitations for the implementation of learning stories and activities within their technical contexts, and to assist them in the identifying learning stories and activities which are practicable given the technological resources available to them. In order to achieve this, iTEC created a Scenario Development Engine (SDE). This is a novel approach in this domain, as previous systems provided, at most, lesson plans that required a given collection of tools to be implemented. In other words, state-of-the-art systems did not provide assistance in discovering lesson plans that could be implemented with the tools available to the teacher. In addition to providing support in assessing feasibility, the SDE also provides recommendations on the three types of resources (people, events, and learning material) that can be used to implement learning stories and activities, namely technological tools including software applications, and events (see also next paragraph). The SDE offers the typical functionality of a traditional recommendation system (Ricci et al. 2011).

[6] Here the term 'learning design' is used as a generic term, not to be confused with IMS-Learning Design.

However, unlike typical recommendation systems, which base their operation on the computation of an estimated utility level for a given user, the SDE provides recommendations taking into account the technical and pedagogical context in which learning stories and activities will be developed. This approach is inherently more complex, as the 'suitability' of a resource in our case is more difficult to determine, because it cannot be computed according to the tastes or interests of a particular person, but rather depends on the assessments of a community of experts.

The SDE combines two state-of-the-art technologies. First, the SDE is based on multi-criteria recommendation techniques (Matsatsinis et al. 2007; Lakiotaki et al. 2008) that consider several factors (identified and ranked by the community of experts) to compute the relevance of resources. Second, like other recommendation systems (Peis et al. 2008), semantic technologies are used to represent the information managed by the system to improve the handling and integration of data from different sources, and above all, to update the underlying models. Note that these models have to be updated frequently, as new rules or resource types (e.g., new types of tools or events) may appear at any time.

Innovations in the Use of A-Typical Resources for Learning

Figure 4.1 describes five interactions that can be used as levers for engagement. For example be able to chat with an astronaut, seek help from a retired person willing to assist with mathematics, being able to participate to events organised by others. iTEC investigated whether new forms of interactions can be integrated in the classroom in an easier way and whether the approach can be scaled. While this may not be the first time that some of these interactions have been proposed, they are certainly not mainstream. The aim of iTEC was to identify the barriers to creating these interactions and to find ways to overcome them. By doing so, iTEC sought to facilitate the exploration of new ICT enabled scenarios, new roles, and new situations in the learning process.

The basic instrument is a People and Events repository that allows users to find People who are willing to contribute to a learning activity or Events organized by others and in which learners and/or teachers can participate. Whereas professional networks—such as LinkedIn—have already existed for some time, they are too generic for this purpose, and do not fulfil the requirements of the educational sector for professional networking. Similarly, the technology—a repository with faceted search—is not new, it is the application of this technology which is of interest. More specifically, iTEC investigated the following questions:

- To what extent is there an interest in sharing information on People and Events?
- Which types of People and Events are of interest?
- What information about People and Events should be gathered, using which vocabularies?
- What level of sharing is appropriate: in schools, region/country, or in Europe?

- To what extent do teachers make use of the opportunity to find people and events and/or recommendations for learning activities?
- What are the barriers and enablers?

Innovation in the Integration and Management of Learning Services and Resources

One of the main bottlenecks in mainstreaming technologies is the integration of technologies into the environment that the teacher is familiar with and/or which she is required to use. Innovative tools and services are often designed for a particular combination of operating system, hardware (PC, tablet, mobile phone, whiteboard), and software (e.g. Moodle, Blackboard, Facebook). Proprietary systems, also used elsewhere (Govaerts and Dahrendorf 2011), exist which resolve part of this problem, such as the Apple App Store and Google Gadgets, but they are restricted to particular platforms. Consequently, in order to facilitate the integration of new applications into as wide a range as possible of real-life classroom environments, iTEC chose to support the delivery of services through non-proprietary interoperability specifications and software. It was decided that the most effective and sustainable solution would be to use the W3C specification for Packaged Web Apps (Widgets), which is expected to facilitate the interoperability of a wider range of platforms.

Beyond the need to support this technical integration, it is also necessary to enable teachers and students to find and deploy the widgets which they would like to use. iTEC has developed a **Widget Store** to meet this need, which can be embedded in any web platform with a modest programming effort. This enables widgets to be described either formally, using the iTEC classification, or informally using tags. Paradata on the use of the widgets is cumulated across various instances of the store. The Widget Store has an API which provides access to this data, which can be processed by recommender engines (including, but not limited to, the SDE), or in learning analytics applications. The Widget Store and its underlying servers are all open source, and are built using Apache Wookie and the Edukapp server software. iTEC has been a leading contributor to both of these projects (Wilson et al. 2011; Griffiths et al. 2012).

This vision of making use of the W3C widget specification to deliver flexible services across platforms was set out in the iTEC project proposal, and has been realised in the Widget Store outlined above. The widget package is itself a rather simple structure, consisting of some HTML, some JavaScript and some image files. However, its very simplicity means that it can be used in a number of different ways, and as a consequence it may be misleading simply to state that iTEC makes use of widgets. It is more valuable to consider the approaches which can be taken to providing functionality with widgets. We may distinguish the following approaches:

- As a platform for delivery of single user applications (e.g. a task timer)
- As means of accessing services provided by the Wookie server which underlies the Widget Store. This manages user identity and enables applications to provide

threaded multi-user services which can be deployed over multiple platforms. These may be relatively simple (e.g. voting), or more complex
- As a simple way of accessing information tools (e.g. time servers, 'this day in history')
- As a means of integrating more complex external services (for example Etherpad)

In addition, part of the project vision for widgets was that tools would be provided which enabled teachers and learners to create their own widgets. The Widget Store supports using three principal approaches:

- As a way of delivering open content from the Internet, embedded into widgets
- As a way of publishing small websites created by teachers and students
- As an interoperability platform (e.g. uploading a Flash file and making it available as a widget)

The iTEC Technical Architecture

Even when the functionalities described earlier in this section are made available, it is still challenging for teachers and ICT coordinators to integrate such services. Therefore, iTEC has adopted the cloud approach—the iTEC Educational Cloud (IEC)—such that the described services are available without cumbersome installations by teachers, learners, or ICT coordinators.

The design of the IEC reported in this chapter has been guided by the following key design principles:

- Collaborative and social functionality
- Accelerated feature delivery
- Open integration protocols
- Serving multiple tenants, a tenant being a group of users (e.g. a school, region, or country) sharing the same view on the technology-enhanced learning environment providing ease-of-use in configuring and customizing such an environment

These principles are characteristic of cloud computing and more particular for Software as a Service (SAAS) models.

It is however not sufficient to develop the architecture according to the vogue of the time. The architecture should serve a relevant user community and follow a solid methodology. A number of efforts have been made to describe and guide construction-oriented research processes (Hevner 2007; Vaishnavi and Kuechler 2007; Takeda et al. 1990). iTEC opted to adopt design science research which is a research paradigm in which the researchers seek answers to their questions about the problem in focus through the creation of innovative artefacts (Hevner 2010; March and Smith 1995).

By making use of a design science research methodology, we ensured that the value of our solution to the general problem (i.e. a need to improve the uptake of ICT in schools) was evident to practitioners and researchers, in order to promote

commitment to the solution and acceptance of the results. In the design process for the IEC we have identified the following stakeholder roles:

- *Learner*: A Learner is a person who is actively engaged in Learning Activities to enhance their knowledge, skills, and competences. A Learner interacts with the Resources provided to her via a Shell.
- *Teacher*: A Teacher is a learning facilitator who supports pupils in their Learning Activities. A Teacher administers a group of Learners via a Shell and stimulates learning by re-using Resources.
- *Learning Designer*: A Learning Designer is a role that can for example be adopted by advanced teachers, head masters, or faculty at universities. A Learning Designer inspires other teachers to adopt pedagogical innovation mediated by Learning Story and Activity Designs.
- *Technical Pedagogical Coordinator* (TPC): A TPC is in charge of inspiring the teachers in their organisation(s) to adopt pedagogical innovation mediated by Learning Stories and Activities. Coordinators are also in charge of administering and deploying the technical infrastructure that supports the facilitation of learning.

The IEC encompasses all the services that are made accessible to its user, whether directly or indirectly. It includes user-end services, back-end services and also some horizontal services that securely connect end-user technologies to form a single, homogeneous and consistent activity space. More specifically, the IEC consists of the following core services; for the sake of clarity not all them depicted in the Architecture Overview of Fig. 4.2:

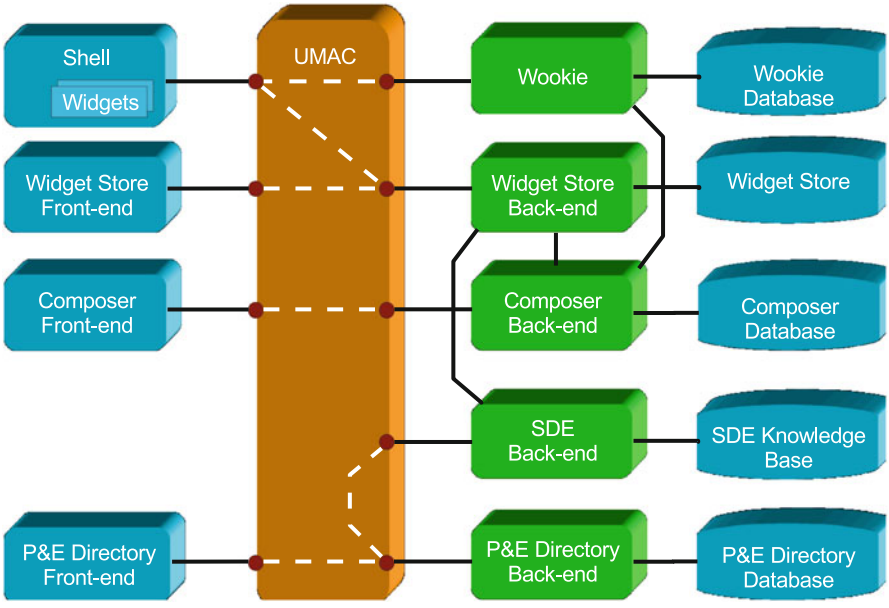

Fig. 4.2 The iTEC educational cloud architecture

- *Shell*: a configurable software container that (as the name suggests) acts as an empty shell allowing users to identify and add their own Resources and to integrate them in order to meet the educational objectives of a Learning Activity.
- *Composer*: an application that supports technical pedagogical coordinators as well as advanced teachers in accomplishing three main tasks: (1) composing Learning Activities and Learning Stories, (2) managing Learning Resources such as Content, and Tools, (3) administering Technical Settings of learning environments.
- *Scenario Development Engine* (SDE): a software component offering back-end services related to technical localisation, i.e., identifying which Learning Activities can be implemented in a school. The SDE also supports resource planning, providing recommendations on the best Learning Resources with which to fulfil the requirements included in a Learning Activity.
- *Widget*: a Web-technology based container for Resources that comes with a graphical user interface for displaying information arrangements and provides standardized methods for data manipulation. Widgets can run in a Shell (described above) supported by the Apache Wookie run-time environment.
- *Widget Store*: a software component that supports creation, upload, tagging, and searching for Learning Resources in the form of Widgets.
- *People & Events Directory*: a directory where users can find Contributors to a Learning Activity, and potentially useful Events.
- *User Management and Access Control* (UMAC): a set of components that supports user authentication and authorization throughout the IEC. It comprises three main modules: an authentication server, an authorization server and an authorization filter that controls access to the above mentioned components. Once a user is authenticated, she can use the different services dependent on her authorization.

Our use of the Software as a Service concept is clarified in Table 4.1, which provides a mapping of the main characteristics of SaaS to our approach.

The User Management and Access Control system

While in the ensuing chapters the full functionality of the services shown in Fig. 4.2 is described, in this section we discuss the UMAC shared middleware service.

As described in the previous sections, iTEC integrates a wide variety of components, including shells, web applications, self-contained widgets, and widget-based applications. This integration raises some questions in terms of user management and access control:

- User authentication may take place at the shell level, but also, some integrated services may require some form of authentication or at least be aware of the visiting user's identity. This implies the need for an central authentication mechanism that can span the range of components and provide consistent information about the user.

Table 4.1 Mapping of SaaS components to the IEC components

SaaS characteristic	Educational cloud components
Collaborative and social functionality	The three main subsystems: the composer, the people and events directory, and the widget store provide collaborative and social functionality. The composer supports the sharing of resources such as learning activities and learning stories; the P&E directory together with the Widget Store supports sharing of *people*, *events* and widgets, and also implements a full set of social metadata. In particular, the Widget Store, acts as a marketplace for learning resources, in content and tools targeting teachers and learners
Accelerated feature delivery	The IEC architecture combines various application service providers, allowing each to rapidly deliver new functionalities. In order to offer an integrated service, integration protocols (see below) are required
Open integration protocols	The IEC architecture combines the offerings of various application service providers, including the Composer, the P&E directory, the SDE, the Widget Store and UMAC. These are integrated using integration protocols. In Fig. 4.2, the communication between the components is shown as lines. This communication may or may not be controlled by UMAC. In the latter case, the service is itself responsible for the authorization handling of its API. In addition these protocols for integration are *open* for other applications to integrate with Examples of the open integration protocols are (a) the P&E API for updating and retrieving information about people and events, and (b) interfaces provided by the shell to be exploited by the widgets, for example Widget APIs and inter-widget communication capabilities Apart from the fact that each service comes with its own set of protocols, some protocols are common and are used by multiple IEC components; viz. the iTEC Protocol for Data Harvesting (iTEC-PDH) and the UMAC API for user management and access control
Serving multiple tenants	A *tenant* is a group of users sharing the same view on the technology-enhanced learning environment. Within the IEC multiple tenants (e.g. schools, regions, or countries) are served at the same time. One of the key features for achieving this is the provision of multilingual services based on shared multi-lingual vocabularies as well as customization and configuration features
Ease-of-use in configuring and customizing such an environment	Customization and configuration may be required for a context which includes multiple tenants, and it is certainly true in the present case. Therefore the IEC is built for easy configuration and customization through (a) its Shell that allows the IEC to be delivered with different application run-time environments such as Moodle and DotLRN, and (b) the widget engine that allows configuring one's own technology-enhanced learning environment

- Access control policies may be defined centrally, at the iTEC Cloud level, but these policies have to co-exist and be consistent with those defined at the shell level, or at the integrated services level, if any. Again, this requires an authorisation mechanism that integrates at the various levels of the architecture.

Because end-users are highly sensitive to authentication and authorisation mechanisms and difficulties they may encounter in using them, we ran a survey among

iTEC teachers, and collected 269 responses from 17 European countries. One of the main conclusions of the study was that using iTEC services should not add extra authentication burden on users. Rather, iTEC will have to extend existing infrastructure and offer the possibility of re-using credentials that users may already possess with external identity providers. However, because some users are concerned that re-using credentials might constitute a security risk, it is important to propose a mixed approach.

Complete results of the survey are presented in Colin and Simon (2012).

Our goal was thus to design a system that meets the following requirements:

- Allow user authentication at the shell level, and convey the user information to sub-components (widgets and back-end services)
- Allow access policies to be defined globally to the IEC, based on a Role-Based Access Control (Ferraiolo et al. 2001) model
- From the global access rules, provision local policies to every iTEC sub-component
- Support interoperability with major service providers, like Google, Facebook, Yahoo…

Designed Solution

The interoperability requirements led us to focus on open standards and protocols to build authentication and authorisation mechanisms. We performed a thorough study, and identified candidate protocols like SAMLv2,[7] OpenID[8] and oAuth.[9] Due to their technological maturity, their relative simplicity, their support for web interactions, the availability of libraries and their wide adoption by main actors on the net, we selected oAuthv2 and OpenIDv2 as the basis for our solution. The fact that users are warned when an application wants to access protected data was also an element of choice.

OpenIDv2 (OpenID Foundation 2007) is an open and standard protocol for signing on to websites using one single set of credentials. The protocol has been developed for many years and adopted by major players on the Internet, like Google. It relies on the assumption that users have an identity defined with an Identity Provider (IdP), and want to use that identity to access various services offered by Service Providers (SP). The typical flow is a user visiting a Service Provider that requires authentication; SP prompts the user for her identity or that of her IdP. The user is then redirected to the IdP to authenticate, and if authentication succeeds, the user is sent back to the SP with the proof that successful authentication did take place. Optionally, the IdP may provide additional information about the user (this requires some protocol extensions).

[7] http://saml.xml.org/

[8] http://openid.net/

[9] http://oauth.net/2/

OAuthv2 (Hardt 2012) is a protocol for managing delegation of authorisation. Its main use case is a user (the resource owner) needing to give access to some of its resources hosted on a server (the resource server) to a client, typically another service. To avoid forcing the user to give her credentials to the client, oAuthv2 introduces a workflow where when the user is asked by the client to give access to a resource, she is sent back to an authorisation server where she authenticates and is then asked to grant or deny access. Upon success, the authorisation server issues an access token to the client that it will use to access the resource on behalf of the user. In this way, the user's credentials are never disclosed to the client. This is the protocol that Facebook or Yahoo use for granting access to their services to remote sites, after getting the agreement of the user. oAuthv2 supports various types of 'grants', to support different profiles of this protocol and accommodate different situations:

- **Authorisation Code Grant**: this is the most secure scenario, in which the client directs the resource owner to the authorisation server for authentication and access request; upon success, the authorisation server issues an authorisation code to the client, that the client then exchanges with the authorisation server for an access token, that is finally presented by the client to the resource server to get access to the resource. All interactions with the resource owner go through her user-agent (typically her browser). This scenario supports client authentication by the authorisation server before issuing an access token, and ensures that the access token never reaches the resource owner's user-agent, which could lead to token leakage.
- **Implicit Grant**: this is a simplified version of the previous scenario, in which instead of being issued an authentication code by the authorisation server, the client directly receives an access token. This scenario is targeted at clients implemented in a browser, typically in javascript. In this case, the authorisation server does not authenticate the client, and the access token is exposed to the resource owner or other applications with access to its user-agent.
- **Resource Owner Password Credentials Grant**: this scenario is built on the assumption that there exists a high degree of trust between the resource owner and the client. The resource owner provides the client with her credentials, and the client uses them to request an access token from the authorisation server. This scenario supports client authentication.
- **Client Credentials Grant**: in this scenario, the client is acting on its own behalf, not on behalf of the user. The client authenticates directly to the authorisation server and receives an access token.

It is worthwhile noting that oAuthv2 also supports extension grants that allow to extend the token request mechanism to support different types of credentials, like SAML assertions.

Because we chose to use oAuthv2 to secure widget access to back-end services, and because widgets usually involve client-side computing and get access to the user's environment, the implicit grant is the only option of choice. However, we also successfully implemented the client credentials grant to secure access to the SDE backend service. One of the drawbacks of the implicit grant is the absence of client

authentication, but this can be explained by the nature of widgets, which are running client-side, making available any sensitive information to other components running in the user's environment (user-agent). It would thus not be possible to securely store client credentials at the widget level.

The User Management and Access Control (UMAC) sub-system glues together all IEC components with the above protocols, and comprises the following components:

- The *UMAC server* is responsible for user authentication, issuance of oAuth tokens, and management of user data and privileges; it plays the role of the OpenID's Identity Provider, the oAuth's authorisation server, and implements a back-end service to access, store and manage user data and privilege information.
- The *UMAC filter* is an authorisation guard that sits in front of back-end services; the back-end service represents the oAuth's Resource Server, and the UMAC filter is in charge of validating access tokens.
- The *UMAC management widgets* are a collection of widgets that allow to access and manage authentication and authorisation information in the iTEC Cloud. Those widgets allow to register a new user, to update a user's details, to create sets of users, and to assign iTEC roles.
- The *UMAC library* is a JavaScript library of tools to help the widget developer to easily integrate with the UMAC framework and not care about the various protocols' implementation.

These components are described in greater details in the next sections.

UMAC Server

The UMAC Server serves two main purposes: authenticating users and controlling access to back-end services.

To authenticate users, UMAC Server implements the OpenID Provider specification. It handles authentication requests from iTEC user-facing components (OpenId relying parties), typically shells or web applications, authenticates users, and responds to relying parties; UMAC Server supports SREGv1.0 and AXv1.0 OpenID extensions to provide basic information of logged in user (username, first and last names, email address, language, timezone, country). Authentication is checked against a local database of users.

One of the requirements drawn from the survey described above mandated that iTEC should allow users to login using third-party credentials, namely Google, Facebook or Yahoo. Thus the UMAC Server supports user authentication using any of those systems, by implementing an OpenID Relying Party (in the case of Google and Yahoo) and an oAuth client (in the case of Facebook).

Access control to iTEC services is handled by the UMAC Server. Access requests may come from widgets or web applications, in which case the oAuthv2

scenario implemented is the implicit grant, but requests may also come from standalone applications, which are run in a more controlled environment, and for which the selected scenario is the client credentials grant. Thus the UMAC Server implements the related sections of the oAuthv2 specification, and handles Authorisation Requests (for the implicit grant) and Access Token Requests (for the client credentials grant), issuing access tokens to widgets and controlled applications respectively.

In addition to the authentication and authorisation functionalities, the UMAC server is also used to store user information; this information is made accessible to UMAC widgets and some other IEC components through a REST API, protected by the oAuthv2 protocol, just like any other iTEC back-end service.

Finally, the UMAC server is used to manage user privileges; those privileges span all iTEC services, i.e. apply equally to shells, widgets or back-end services. Six levels of privileges are defined in a strictly hierarchical way: super-user, administrator, coordinator, teacher, student and guest. The level of privilege of a user is passed to the OpenID relying party upon authentication through SREG or AX extensions, where available, and they are checked by the token validation process between the UMAC filter and the UMAC server.

For a seamless user experience, UMAC authentication is propagated to the shell through a plugin mechanism which is dependent on the shell itself. In this way, once the user is authenticated, all shell components (typically widgets) can reuse the user information.

UMAC Filter

The UMAC filter is designed to be deployed in front of back-end services, and interacts with the UMAC server following the oAuthv2 protocol to control access to the services by ensuring that only authorised requests get served. The current implementation of the filter takes the form of a servlet filter, which makes it very easy to integrate and (de)activate and realises a separation of concerns by allowing the service developer to work independently from the access control mechanism.

In oAuthv2 terminology, the UMAC filter acts as the protection part of the resource server. It receives requests for access in the form of REST calls (basically http requests), and for each requests, it checks that a valid access token is provided. If no token is present, an error is returned, and it is up to the client to obtain one. If a token is present, its validity is checked by querying the UMAC server through a secure channel, and upon success, the lifetime of the token and the user id of the token owner are returned to the filter. Based on this information, the filter then checks the local access policy that defines the rules for accessing the service. These rules are expressed using the Apache Shiro[10] system. If the rules are evaluated positively, access is granted

[10] http://shiro.apache.org/

and the request is passed to the service. Otherwise, an error is returned. For efficiency reasons, the UMAC filter caches the validated tokens for a period of time to avoid unnecessary roundtrips with the UMAC server.

UMAC Library

The UMAC library is a Javascript library of functions that aims at facilitating the development of widgets and their integration with UMAC authentication service, more precisely, the oAuth authentication endpoint's service. It hides the complexity of the protocol by providing methods to manage the whole authentication process (request for token, redirect to authentication form, token transfer to requesting component and error handling).

Figure 4.3 presents the UMAC components (in gray) as well as the interactions with other iTEC systems. These components are a decomposition of the UMAC component depicted in Fig. 4.2. The UMAC Server is used for authentication (solid lines) either from a shell, widgets or web applications like the Composer or the Persons and Events Directory. This follows the OpenID protocol. Authentication may be local (using the User DB) or rely on third-party authenticators (right-most box). Regarding authorisation (fine dashed lines), UMAC widgets support registration or update of user information through the UMAC REST Web Service, which is protected by the UMAC filter. Similarly, any other iTEC component may access iTEC back-end services which are protected by the UMAC filter (see bottom of the diagram). The UMAC filter validates authorisation with the UMAC server (large dashed lines).

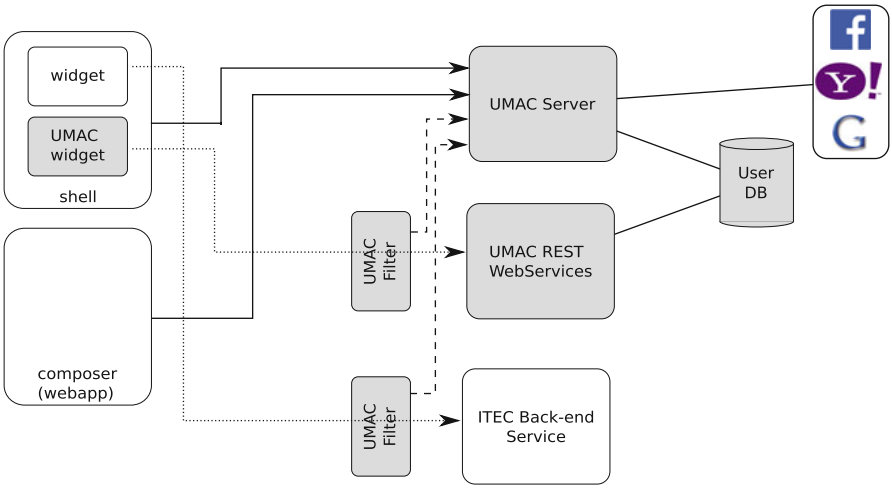

Fig. 4.3 Interactions of UMAC components with other iTEC systems: the example of the composer

Sharing Data

In iTEC, semantic interoperability was achieved by a shared data model for exchange between the iTEC systems and the multilingual vocabularies as described in the appendix of this book. The principle shared object types are:

- Event: a description of interesting Events, maintained in the Persons and Events directory
- Learning Activity: a description of iTEC Learning Activities as provided for example by teachers and maintained in the Composer
- Person: a description of a Person such as an expert, maintained in the Persons and Events directory
- Resource Guide: a description, maintained in the Composer, of resources used with LearningActivities
- Technical Setting: a description, maintained in the Composer, of the technical capabilities of a school or classroom
- Tools: a description of tools used in Learning Activities and Learning Stories, maintained in the composer
- Widget: a description of a widget as recorded in the Widget Store

In addition to the data models, iTEC also implemented a protocol for data harvesting (the iTEC-PDH). Within modern REST interfaces, JSON strings are currently preferred over XML technologies, because JSON facilitates rendering in user interfaces, especially browser-based user interfaces, e.g. W3C widgets. Consequently most REST interfaces in the iTEC architecture are based on JSON strings. The iTEC-PDH follows this approach while borrowing operational semantics from OAI-PMH.

iTEC-PDH request

A service implementing the iTEC-PDH must respond to an http GET request. The GET request has four parts:

- The first part refers to the service—i.e. the harvesting target, e.g. 'http://ariadne.cs.kuleuven.be/itec-directory/api/rest/'.
- The second part specifies the object type. In REST terms, it refers to the collection. For example 'Event'.
- The third part is the string '/harvest'.
- The fourth part is optional and is given as an http query string. It may contain the following elements: 'from=<date-time spec>' and 'until=<date-time spec>'. The <date-time spec> is following the date-time data type (see "Person" in Appendix). As customary the http query string parameters are joined together with an ampersand and follow a question mark. For example '?from=2012-09-15T00:00:00.000+02:00&until=2012-09-16T23:59:59.999+02:00'. As for OAI-PMH the

boundaries must be included in the search results. A service may also implement EPOCH time in milliseconds for these too parameters. For example '?from=1358377200000&until=1358463599999'. The default value for the 'from' value is the beginning of the service. For practical reasons this may be taken as 0 in EPOCH time. The default value for the 'until' parameter is the time the request is received by the service.

iTEC-PDH Response

The response to an iTEC-PDH request is a regular http GET response with a JSON array as the payload. The JSON array contains the update elements as shown in Table 4.2. Each element has

- An identifier labelled "id" with a value following the 'identifier' data type described in "Person" in Appendix.
- A date of last modification labelled "last_mod" with a value following the 'date-time' data type described in "Person" in Appendix.
- The status of the last update, labelled "status" with a value from the value space {"created", "modified", "deleted"}.

In addition an element with the status "created" or "modified" must have an element labelled "entry" that gives the created or modified entry. The entry itself must follow the data model as specified in the data model as described in the appendix of this book. Note that vocabulary tokens are used if a data element of an entry is of the data type "VocabularyTerm".

It should be noted that an entry may contain an internal identifier such as shown in Table 4.2 "_id".

Conclusions

This chapter has reported on the iTEC architecture and artefacts addressing the most important choking points in the uptake of ICT in schools as well as building on the engaging potential of ICT in learning activities. We have focussed specifically on the innovations in the technical area, and provided and introduction to the Scenario Development Engine, the Widget Store, the People and Events directory, and the iTEC Education Cloud.

Dozens of classroom experiments have led to the identification of both successes and problems for each of the different technical artefacts, and also indicate that as a whole iTEC makes a significant contribution to re-engineering the uptake of ICT in education (See also Chap. 9: Evaluation). It is our belief that the realisation of the future classroom as envisaged by current research efforts can only succeed if sufficient progress is made in technology that will facilitate (and not hamper) the uptake of ICT in schools.

Table 4.2 Example harvesting result


```
[
  {
    "id": "http://itec-directory.eun.org/Person/2305",
    "last_mod": "2012-09-16T10:45:31.190+02:00",
    "status": "modified",
    "entry": {
      "_id": 2305,
      "givenName": "Otto ",
      "familyName": "Leskinen",
      "loginName": "Otto Leskinen",
      "mbox": "otto.leskinen@eduouka.fi",
      "gender": "1",
      "birthDate": null,
      "categories": [
        "teacher"
      ],
      "languageMotherTongue": "fi"
    }
  },
  {
    "id": "http://itec-directory.eun.org/Person/2405",
    "last_mod": "2012-09-07T12:26:15.984+02:00",
    "status": "created",
    "entry": {
      "givenName": "Frans",
      "familyName": "Van Assche",
      "loginName": "fvanassche",
      "gender": "1",
      "description": "Test",
      "birthDate": null,
      "_id": 2405
    }
  },
  {
    "id": "http://itec-directory.eun.org/Person/2605",
    "last_mod": "2012-09-12T18:32:29.884+02:00",
    "status": "deleted"
  }
]
```

Acknowledgement The authors wish to thank Elena Schulman for coordinating the work on the data models and vocabularies, and David Massart for his early work on the architecture.

Open Access This chapter is distributed under the terms of the Creative Commons Attribution Noncommercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

Colin J-N, Simon B (2012) D7.2: second generation of iTEC shells and composer. Project deliverable 7.2, University of Namur

Ferraiolo DF, Sandhu R, Gavrila S, Kuhn DR, Chandramouli R (2001) Proposed NIST standard for role-based access control. ACM Trans Inf Syst Secur 4(3):224–274

Govaerts S, Dahrendorf D (2011) Deliverable D3.4 of the ROLE project, prototype implementation (2nd updated version)

Griffiths D, Johnson M, Popat K, Sharples P, Wilson S (2012) The Wookie Widget Server: a case study of piecemeal integration of tools and services. J Univ Comput Sci 18(11):1432–1453

Hardt D (ed) (2012) The OAuth 2.0 Authorization Framework. RFC 6749, RFC

Hevner AR (2007) A three cycle view of design science research. Scand J Inf Syst 19(2):87

Hevner A, Chatterjee S (2010) Design research in information systems: theory and practice. Springer, Berlin

Lakiotaki K, Tsafarakis S, Matsatsinis N (2008) UTA-Rec: a recommender system based on multiple criteria analysis. In: Proceedings of the 2008 ACM conference on recommender systems, Lausanne, Switzerland, 2008

March ST, Smith GF (1995) Design and natural science research on information technology. Decis Support Syst 15(4):251–266

Matsatsinis NF, Lakiotaki K, Delias P (2007) A system based on multiple criteria analysis for scientific paper recommendation. In: 11th Panhellenic conference in informatics, Patras, Greece, 2007

OpenID Foundation (2007) OpenID authentication specifications 2.0. Openid. http://openid.net/developers/specs/

Peis E, Morales-del-Castillo JM, Delgado-López JA (2008) Semantic recommender systems. Analysis of the state of the topic. Hipertext.net, no. 6

Ricci F, Rokach L, Shapira B, Kantor P (2011) Recommender systems handbook. Springer, New York

Takeda H, Veerkamp P, Tomiyama T, Yoshikawa H (1990) Modeling design processes. AI Mag 11(4):37–48

Vaishnavi VK, Kuechler W (2007) Design science research methods and patterns: innovating information and communication technology, 1st edn. Auerbach, Boca Raton

Van Assche F (ed) (1998) Using the world wide web in secondary schools. ACCO, Belgium

Van Assche F (2004) Towards ambient schooling. In: Delgado Kloos C, Pardo A (eds) EDUTECH: computer-aided design meets computer-aided learning. Kluwer Academic, Boston

Van Assche F et al (2006) iClass Project Educational Vision Statement, deliverable D3.1 of the iClass project, Aug 2006

Wilson S, Sharples P, Griffiths D, Popat K (2011) Augmenting the VLE using widget technologies. Int J Technol Enhanc Learn 3(1):4–20. doi:10.1504/ijtel.2011.039061