



# iTEC

Designing the future  
classroom



Co-funded by the  
7th Framework Programme  
of the European Union

## iTEC Shells and the Widget Store

Will Ellis, Dai Griffiths, Gill Leahy and Ainhoa Marcos, Bern Simon

<http://itec.eun.org>



## Table of Contents

<b>1</b>	<b>PURPOSE AND SCOPE.....</b>	<b>3</b>
1.1	SHELL PROTOTYPES.....	3
1.2	THE ITEC WIDGET STORE.....	3
<b>2</b>	<b>SHELL DEVELOPMENTS AND CONTENT INTEGRATION.....</b>	<b>4</b>
2.1	OPEN SANKORÉ .....	5
2.2	DOTLRN .....	6
2.3	SMART TECHNOLOGIES: WIDGETS AND SMART AMP.....	9
2.4	PROMETHEAN: CLASSFLOW.....	10
2.5	USER MANAGEMENT AND ACCESS CONTROL .....	12
<b>3</b>	<b>ITEC WIDGET STORE 1.5 RELEASE NOTES.....</b>	<b>14</b>
3.1	INTRODUCTORY COMMENTS .....	14
3.2	NEW FEATURES IN THE 2014 VERSION.....	14
3.2.1	<i>Front Page</i> .....	14
3.2.2	<i>Navigation</i> .....	15
3.2.3	<i>Categories</i> .....	15
3.2.4	<i>Favourites</i> .....	16
3.2.5	<i>My Widgets</i> .....	18
3.2.6	<i>Creator</i> .....	20
3.3	SERVER SIDE AND CODE BASE CONFIGURATIONS.....	21
3.3.1	<i>LTI</i> .....	21
3.3.2	<i>Language Support</i> .....	21

# 1 PURPOSE AND SCOPE

This document describes some of the key technical outputs derived from the iTEC project. Its purpose is to support the further use and development of these tools and the underlying technologies and technical principles.

## 1.1 Shell prototypes

Knowledge Markets – one of iTEC’s technical partners – produced the following technologies to investigate the integration of learning content with open source platforms.

- Edutheek: An LTI-based market place solution for learning management systems
- DotLrn: iTEC’s shell based on learning management systems

Further information on access to these tools can be obtained from Knowledge Markets: <http://www.km.co.at/km/en/index>

The technologies produced by University of Namur in iTEC were the following:

- UMAC server for a central management of the authentication and authorization processes for all iTEC components
- UMAC filter to protect individual web services in a transparent way
- UMAC widget library to help develop secured widgets that work with the UMAC subsystem
- an experimental porting of the Open Sankoré shell to work with the widget store and hence with all other iTEC technologies

They are critical for a secure deployment of an education cloud.

## 1.2 The iTEC Widget Store

The iTEC Widget Store supports the W3C Widget specification and OpenSocial Gadgets, as well as supporting IMS Learning Tools Interoperability (LTI). It provides functionality for hosting, creating, describing, discovering and deploying widgets. To provide this extensive functionality, the Store has built on a number of established open source software systems, and has initiated new code projects where necessary. The software components which have been included in the Widget Store, and extended and adapted by iTEC are:

- Edukapp 10: provides the Store service itself is based upon Edukapp 10 . This software has been extended to include a dedicated pure REST API and also to include some model requirements particularly to describe functionalities.
- Apache Wookie 7: houses, parses, manages and delivers W3C widgets.
- Solr 8 – used for search indexing and query language.
- Shindig 9 – houses, parses and manages OpenSocial gadgets.

The user interface for the store is implemented as a client in a separate software package. The architecture of the Store was designed by iTEC in order to make this separation of concerns possible. The architecture is centred on the REST API (Documented on the IEC WidgetStore Web Site) implemented by the project in Edukapp, which manages communication between the store and the clients. Implementation of the architecture also led to extensive enhancements to the code of Apache Wookie and Edukapp. In this case a pure html/javascript client has been written and packaged as a W3C Widget.

This approach provides great flexibility in the way that the store can be embedded in client platforms. This has been demonstrated through the development of a Moodle block that communicates with the store through the API, through a Web site based Widget Store, and a full implementation of the Store as an embeddable widget.

The iTEC Demonstration platform can be accessed at: <http://wookie.eun.org/Store/>

## 2 SHELL DEVELOPMENTS AND CONTENT INTEGRATION

One of the technical goals of the iTEC project was to enable teachers to integrate the learning tools and content they need in order to set up a tailored learning environment, while supporting sharing across different education institutions and administrations that have widely differing requirements and technological infrastructure. An iTEC Shell is an application that has the capability to select and deliver iTEC resources and services, and to combine them with already existing, "native" applications.

It was a requirement that iTEC should be based on the use of open specifications, and at an early stage in the project the W3C Widget specification was selected as the enabling technology which would support the integration of resources and services into iTEC shells. One advantage of this approach was that there was an existing and well-established open source project which provided some of the necessary functionality: Apache Wookie. As described in the deliverables of WP8, iTEC has extended the functionality of Apache Wookie in order to meet the requirements of the project. The creation of an iTEC Shell involved, therefore, developing a 'connector', i.e. code which enabled an application to select and display W3C widgets.

Work on Wookie resolved the technical delivery of resources and services to different systems and users. However, two other challenges remained. Firstly, it was necessary to enable teachers to describe and discover the resources that they wanted to include in their tailored environments. Secondly, support was needed for teachers and pedagogic coordinators who wanted to create their own resources. Both these needs were addressed by the development and deployment of the iTEC Widget Store. The Widget Store adopts the paradigm of an App Store, which is familiar to all users of mobile platforms such as iOS and Android. Since the development of the Widget Store, the user interface of Apache Wookie has been eliminated, and the Widget Store (or an equivalent application) is the only way that users select and deploy widgets.

The iTEC Widget Store has been implemented with an application programming interface (API) which enables applications to create their own interfaces with which to carry out operations on the Store, and to interrogate the store about its contents. The iTEC Widget Store has also been implemented as a widget, which offers another approach to integration. The Widget Store Widget can itself be integrated into a range of Web based platforms. This creates opportunities for a number of approaches to integration, best illustrated by describing work carried out with the iTEC shells.

The iTEC Widget Store can also be deployed as a stand-alone application. In this mode the user can create, describe and discover widgets, and can copy embed codes which they can embed in other Web applications (for example WordPress). This requires a minimal understanding of HTML, which is not needed for other integration approaches.

In order to test and demonstrate the capabilities of the Widget Store in creating shells, partner Bolton incorporated the Store into Moodle in three different ways, with differing degrees of integration, and which demonstrate differing approaches to integration with a shell.

1. Widget store as a widget. In this approach the iTEC Widget Store is embedded into a Moodle page. Using this approach, widgets held on the Store can be created, reviewed, tagged, described and discovered, but there is no support for installing them. This level of integration is very easy to achieve in any Web based platform.
2. Widget store Moodle block. Within Moodle a block is a component of a page which allows users to select a Moodle resource (for example a forum, or a piece of content). Integration involved development of a new block, which does not give access to a piece of internal Moodle functionality, but rather communicates with the API of the iTEC Widget Store. The indexed search facilities in the store are used to populate the interface, enabling a widget to be installed directly into a block in any page format. Because of the properties of blocks in Moodle, the user is constrained to including a single widget in each block.
3. Widget store Moodle page format. In this approach the iTEC Widget Store is integrated into a custom Moodle page format, building on work carried out within the Omelette project. When the user adds a new widget into the page the store is displayed in an overlay (popup window) with all its features and an additional install option next to each widget. This approach has the additional advantages of enabling multiple widgets to be embedded in the same page, and to export and import collections of widgets using Open Mashup Description Language (OMDL).

The changes implemented in the iTEC Widget Store in the final year of the project are detailed in the *Release notes to the iTEC Widget Store 1.5*, included as an appendix to this deliverable.

## 2.1 Open Sankoré

Open Sankoré is a free open source digital teaching software which is compatible with most of interactive whiteboard hardware. Although the software did not support W3C widgets by default, the University of Namur has extended it to enable such support and allow it to integrate iTEC components. The integration approach has been discussed with the Open Sankoré team. It has been challenging because of the different technologies used to build Open Sankoré. The overall result is a connected Interactive Whiteboard application that embeds the iTEC Widget Store, and

allows the users to select individual widgets or a complete Learning Story. Behind the scenes, the integration links to the other iTEC developments through the common User Management and Access Control component (see section 1.7 below). This work has been experimental and some integration issues still remain. The goal was to demonstrate the technical feasibility and the interest of using standards to quickly integrate different technologies.

Thanks to an architecture based on open-standards and technologies, integration of iTEC components has been greatly facilitated. Through the approach of making content and tools available as widgets, it has been relatively easy to modify shells, as containers, to host any of the other iTEC components. The Widget Store being the central place to access widgets, and being exposed as a widget itself, it has been integrated in the Open Sankoré software, giving access to the full collection of iTEC resources.

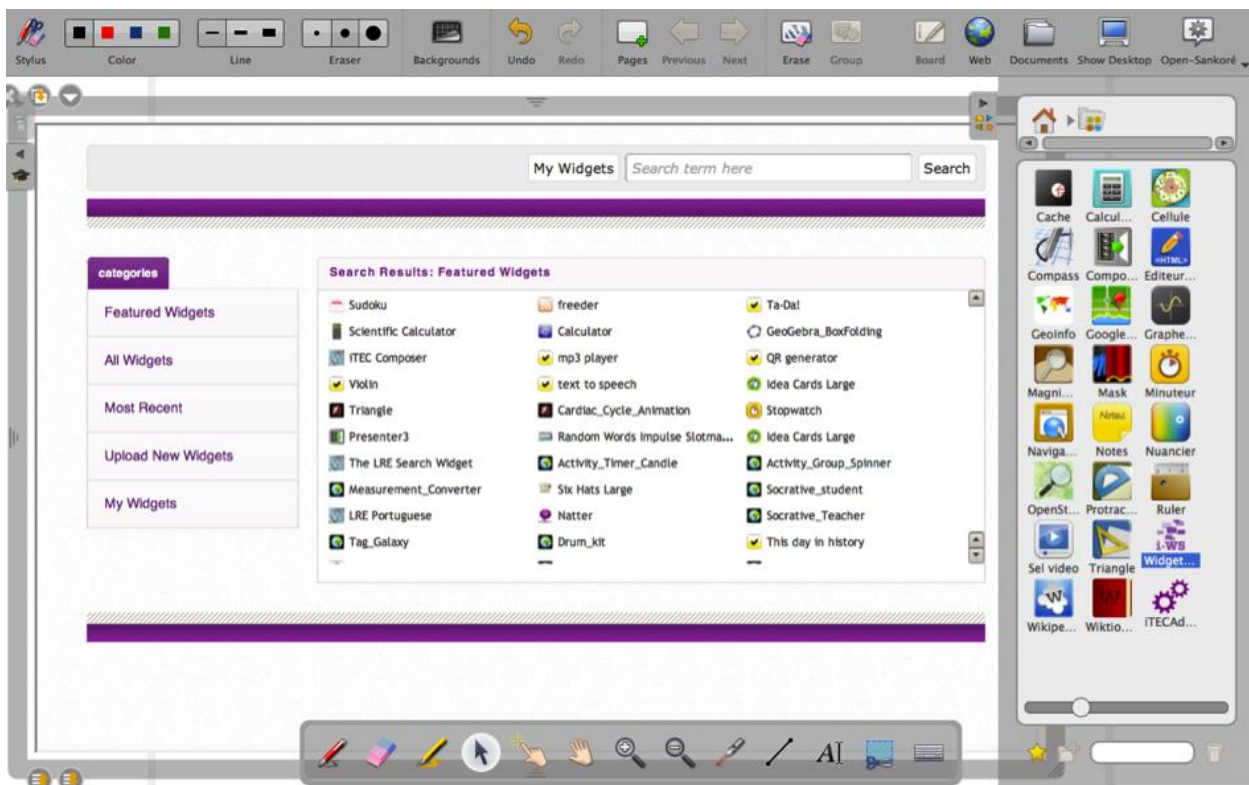


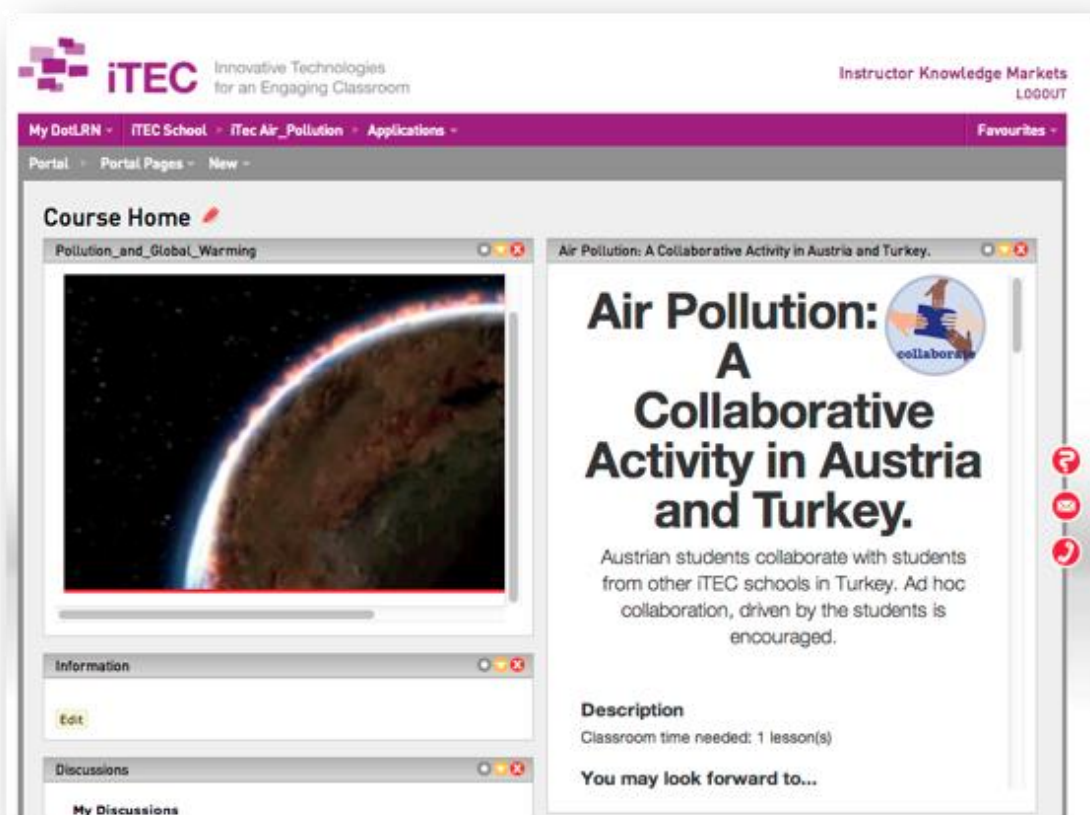
Fig. 1: The iTEC Widget Store in the Open Sankoré IWB software

## 2.2 DotLrn

DotLrn is an open source enterprise-class learning management system (LMS). DotLrn is the technological foundation of Austria's leading LMS solution, lms.at. lms.at serves about 1,200 schools in Austria. The first wave of adoption of lms.at was facilitated by the support of tailor-made use cases of technology-enhanced learning. Beyond the provision of content, lms.at supported an increased transparency of the grading process as well as the exchange of good practices of organising school life via calendar and file sharing. Within iTEC, the providers of DotLrn, Knowledge

Markets, sought to innovate on existing LMS functionality by enhanced availability of standardized content and tools as described in the following sections. As well as integrating the Widget Store, DotLrn was also developed to make use of the existing Learning Tools Interoperability specification (LTI).

Initially DotLrn did not support the W3C widget standard, but thanks to iTEC the widget technology became integrated also into this LMS. The iTEC year 3 version of the DotLrn widget integration supported not only the instantiation of single widget. Whole packages of widgets representing a learning activity can now be selected and instantiated within a DotLrn course, thereby reducing the effort for setting up DotLrn for a widget-supported learning experience. The following figure shows a DotLrn course that is based on the learning activity “Air Pollution”.



*Fig.2: A DotLrn Course based on the Learning Activity “Air Pollution”*

Overall, DotLrn benefits from the integration of third party components based on two standards: W3C widgets and IMS LTI standard. As a result, DotLrn users benefit from a wider choice of available learning resources, but also from a new user experience when it comes to managing DotLrn Portlets. The basic use case behind LTI is to allow the seamless connection of web-based, externally hosted applications and content – summarized under the concept (remote) "Tool" – to platforms that present these Tools to users. For example, it allows the secure integration of external assessment applications or hosted premium content with learning management systems without having to develop and maintain custom integrations for each application or content provider.



Based on this "Tool" concept, LTI distinguishes between the "Tool Consumer", in many cases the Learning Management System "consuming" the Tool, and the "Tool Provider", as it "provides" the Tool for use in the Tool Consumer (IMS 2012). At the Tool Consumer, a Tool is launched within a certain "Context" relevant to the Tool Provider. Examples for a relevant Context are a course, or a school organisation representing groups of users sometimes with different privileges (e.g. instructors, students).

When it comes to the integration of external tools, the iTEC project supported the following requirements, with access management implications:

- R1. Acquire access to a plethora of high-quality learning resources: To maximize the benefits for learners and teachers, an LMS provider needs to offer a broad variety of learning resources from all kinds of relevant publishers and other institutional content providers.
- R2. Maximize coverage for school book publishers: On the other hand, in order to reach out to its customers, it is in the highest interest of a publishing house to make its learning resources available within as many (technology-enhanced) learning environments as possible.
- R3. Keep publishers in control of the provided content: One important requirement that has been identified is to ensure that a publisher of premium content stays in full control of learning resources provided at any time within its lifecycle (i.e. creation, sharing, delivery, termination).
- R4. Allow publishers to manage access to resources: Another requirement stemming from our publisher's business model is the need for staying in control of content access policies. Given the fact, that a subset of the learning resources offered might be considered premium content – and therefore the publisher has no interest in making it freely available – a third party must not be in charge of managing access control.
- R5. Support sharing of different learning resource types: In the school sector, relevant content varies significantly and, as a consequence, the integration approach needs to be able to support the hosting of all different kinds of materials. While some educators prefer content that is optimized for a paper-based learning environment (e.g. PDF documents) others feel the need to share standard formats (e.g. PowerPoint) for electronic delivery. Others might even require the exchange of e-learning content in SCORM or other formats.
- R6. Allow easy integration in learning management systems: From the perspective of the provider of a learning management system, the integration of learning resources should be as seamless and easy as possible. Providers would like to avoid developing and maintaining custom integrations for each content-providing institution.

Based on the above-mentioned requirements, our integration approach has been carried out and tested in the context of the following case studies:

Austria's leading publishing house in terms of market share, Veritas, agreed to provide all its school-book accompanying online materials to the marketplace using LTI. Veritas is now able to seamlessly integrate an archive of LTI links to its online material within Austria's leading LMS. This archive is un-packaged, the metadata extracted together with the LTI links. The set of LTI links is then made available to the end user. From the Tool Consumer's point-of-view, i.e. the LMS, the LTI integration allows an automated login and authorization at the Veritas web site where the premium content is hosted.



## 2.3 SMART Technologies: Widgets and SMART amp

From the beginning of the project SMART Technologies developed three iTEC widgets based in the pedagogical part of each cycle scenarios to support the development and adoption of Learning Activities.

- Random Words for lateral thinking
- Six thinking hats for changing perspectives and reflection
- Idea cards to trigger observation methods (to support design and inquiry methods)

The three widgets were developed in two formats, W3C for being implemented in the Widget Store and also in any other IWB, VLE, PLE; and also in SMART Notebook format for being displayed in SMART Notebook learning platform. The widgets were translated into English, German, Spanish, Polish, Hungarian, and were used by teachers in cycles 3-5 of pilots.

Those widgets are available for teachers in both the Widget Store and in SMART Exchange <http://exchange.smarttech.com/>

Widget is displayed  
inside SMART Notebook



*Fig.3: A SMART Widget displayed in SMART Notebook.*

This special SMART Notebook Widget was developed specifically to run iTEC widgets (W3C) within SMART Notebook software.

Using this specific widget, teachers and students are able to embed any iTEC widget from the Widget Store in their lessons using SMART Notebook software. SMART Notebook software is a learning platform that has been used as a shell for developing the iTEC LA and LS.

The ability to include iTEC Widgets into SMART Notebook collaborative learning software has allowed teachers to develop their iTEC Learning Activities using one learning platform.

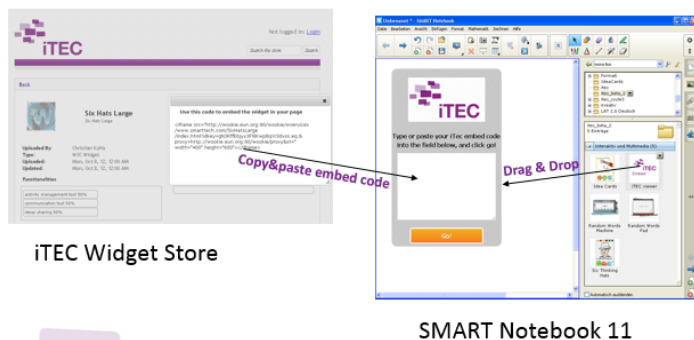


Fig. 4: iTEC Widget Store used to add widgets to SMART Notebook.

The iTEC Widget Store concept, along with the design of Learning Scenarios and Learning Activities, has supported the development of the SMART amp collaborative learning software. SMART amp is a cloud based tool designed specifically for educators. Its power is unlocking the potential of student's devices in the classroom. Most of the iTEC concepts like collaboration, reflection, peer evaluation, design...are behind SMART amp concept. SMART amp enhances collaboration, content creation, communication, sharing and assessment from anywhere, on whatever device is being used. Lessons can begin with a huge, blank workspace, ready for student contributions, or can be prepared with specific learning paths in place so SMART amp has become the perfect tool for developing iTEC learning scenarios in different levels and subjects where collaboration is the key anywhere, at any time and in any device.

SMART amp supports not only trends in education that have been developed during the project like BYOD, but also is breaking the classroom barriers, allowing teachers and students to work together, collaborate and share inside and outside the classroom. As it lives in the cloud, it connects any web-enabled device and gives students freedom to learn.

## 2.4 Promethean: ClassFlow

Involvement in the iTEC project has significantly influenced the development of ClassFlow. [www.classflow.com](http://www.classflow.com) by the ICT Supplier Promethean.

This novel software was developed and tested by teachers during the project and taken to large scale public BETA in January 2014 at the BETT show in the UK.

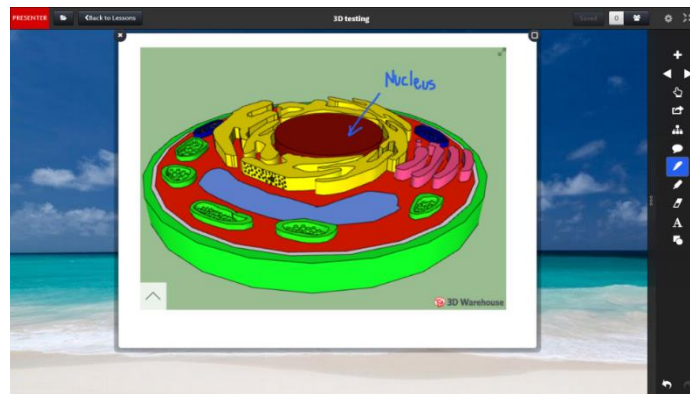
Key technological and pedagogical themes explored and researched in iTEC are now embedded into its design, including collaboration and student centred creativity, web widgets, cloud services and use of emerging social media connections as a method of sharing lesson ideas and projects.

ClassFlow is innovative in the level of 'frictionless' real time interaction and collaboration it achieves both within the web environment and between student devices. HTML5 and associated widgets have been a focus of iTEC and are maximised in ClassFlow to enable a diversity of student devices to connect to each other and to the lesson. Camera and drawing functions enable students to co-create content in real time and a simple URL & 5 letter code enables a very simple way of enabling an interactive classroom.

The product was released as version 1.0 in summer 2014 and has since won a number of education software awards including ISTE “Best in Show” at the largest Education Technology Show in the USA.

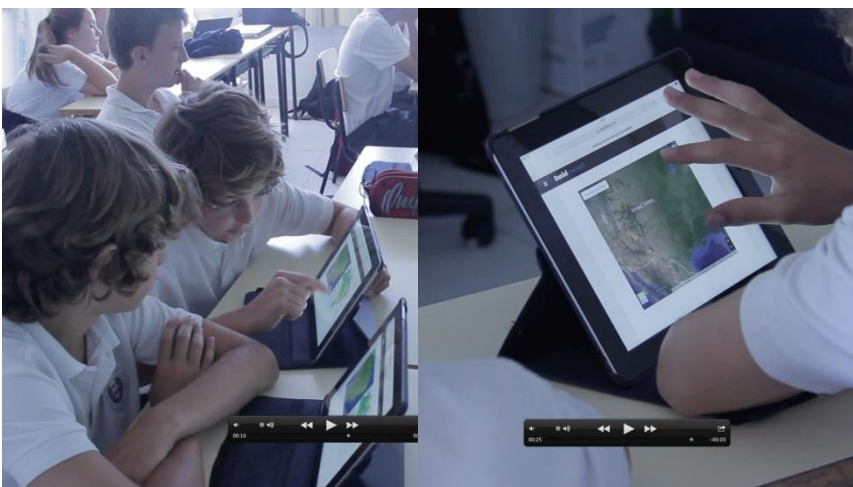
Widgets, as proposed by ITEC, are supported in ClassFlow and prototype lesson ideas were demonstrated early in the project using Promethean’s Inspire software. A demonstration lesson early in the project used widgets to show how real time maps of traffic and pollution data could be combined or ‘mashed up’ into a single lesson. These ideas are now embedded into ClassFlow.

A number of widgets available in the ITEC widget store work directly with ClassFlow and widgets from popular tools like Google Maps, GeoGebra, etc. also work.



*Fig. 5: An interactive 3D model of a cell made by a student and embedded into ClassFlow.*

In addition to viewing widgets embedded into ClassFlow cards, ClassFlow uniquely allows a teacher or student to annotate over the widget. Teachers or students can also “share cards” and if the widgets are interactive, each student can work with them independently. The rich sharing is a feature of ClassFlow.



*Fig. 6: Students using embedded map widgets shared by their teacher in ClassFlow in an ITEC pilot school*

## 2.5 User Management and Access Control

iTEC integrates a wide variety of components: shells, web applications, self-contained widgets, 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 authentication mechanism that can span the range of components and provide consistent information about the user.
- 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.

In addition to those considerations, a survey run among teachers in Europe showed that a significant number of them were not willing to create yet-another-user-account to access a service but would rather re-use an existing account they may already have, with their school or any third party provider (like Facebook, Google or Yahoo). A review of existing approaches and standards hinted at the use of OAuth2 and OpenID open protocols. Not only are those widely adopted but also are they designed with web interactions at their core, which particularly fit the overall iTEC Cloud architecture.

To integrate those protocols into the iTEC environment, we designed the UMAC (User Management and Access Control) framework, which is comprised of the following components:

- the UMAC server is responsible for user authentication, issuance of authorization 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. It links with external services like Facebook or Google for user authentication, where applicable.
- 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 widgets are a collection of widgets that make it possible to access and manage authentication and authorisation information in the iTEC Cloud. These widgets make it possible 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 without knowledge of the various protocol implementations.

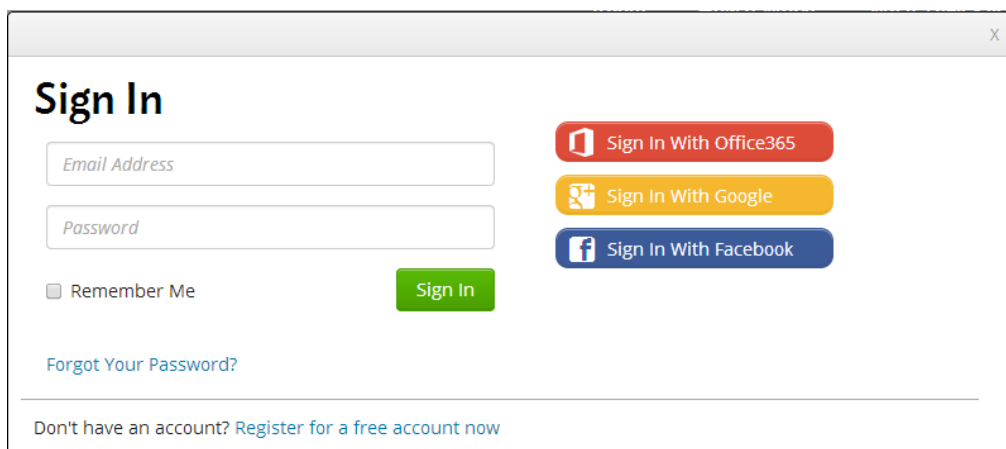
Based on this framework, other iTEC components could be easily integrated; user authentication was delegated to the UMAC server through the OpenID protocol while web services were protected using the UMAC filter, in an almost completely transparent way for the service developer.

Although the proposed approach did not design any new protocol, it is believed that it's innovation lies in the way it brings together various protocols and uses them to bridge the technology gap between different components.

- The Widget Store is designed to inherit the authentication of users from the container platform. Consequently it works smoothly as an embedded application without the need for further authentication. However, a stand-alone version of the Widget Store has also been implemented which has its own authentication mechanism.
- In the case of DotLrn, the project also successfully experimented with the IMS Learning Tools Interoperability (LTI) specification (IMS 2012). The basic use case behind LTI is to allow the seamless connection of web-based, externally hosted applications and content – summarized under the concept (remote) "Tool" – to platforms that present these Tools to users. For example, it allows the secure integration of external assessment applications or hosted premium content with learning management systems without having to develop and maintain custom integrations for each application or content provider.

Based on this "Tool" concept, LTI distinguishes between the "Tool Consumer", in many cases the Learning Management System "consuming" the Tool, and the "Tool Provider", as it "provides" the Tool for use in the Tool Consumer (IMS 2012). At the Tool Consumer, a Tool is launched within a certain "Context" relevant to the Tool Provider. Examples for a relevant Context are a course, or a school organisation representing groups of users sometimes with different privileges (e.g. instructors, students).

- Promethean's ClassFlow has an extensive API and is also deeply integrated with a number of other services via APIs to provide a seamless, single experience. As a result, Web search and cloud folders are fully integrated into ClassFlow which means teachers do not need to juggle applications or windows to create or manage a lesson. All content is held in the cloud and a user can access it from any devices using a login. ClassFlow currently supports login/authentication with Office 365 for Education, Google Apps for Education and Facebook as well as its own accounts. Sign on with popular cloud productivity solutions provides a route to millions of other education users and helps further in taking iTEC ideas "to scale"



The image shows a 'Sign In' window for ClassFlow. It features a title bar with a close button (X). The main content area has a 'Sign In' heading. Below the heading are two input fields: 'Email Address' and 'Password'. To the right of these fields are three social login buttons: 'Sign In With Office365' (red), 'Sign In With Google' (yellow), and 'Sign In With Facebook' (blue). Below the input fields is a 'Remember Me' checkbox and a green 'Sign In' button. At the bottom left, there is a link 'Forgot Your Password?'. At the bottom right, there is a link 'Don't have an account? Register for a free account now'.

Fig. 9: Sign on options for ClassFlow

## 3 ITEC WIDGET STORE 1.5 RELEASE NOTES

### 3.1 Introductory comments

This is the final release of the iTEC Widget Store at version 1.5. Subsequent releases will be transferred to the Widget Store, it's successor.

These release notes are an amalgamation of the release notes for 2014 to date and subsume the notes for the January release. The release notes are based upon a series of screen shots with descriptions. These notes assume that the user is aware of previous versions of the store documented in the release notes for version 1.

The screenshots in this document have been made using local (development) instance of the store so the widgets and data are different from the iTEC instance. The screenshots included here are all taken from the point of view of a signed-in user, as without sign-in the 'Favourites', 'My Widgets' and Creator tabs would not be visible. Most people who log into the store will not be able to see the admin tab.

### 3.2 New features in the 2014 Version

#### 3.2.1 Front Page

The usability research undertaken at the end of year 3 of the project indicated some difficulties with users finding widgets and seeing relevant information about them. Changes were implemented, increasing the size of icons and adding category and rating information to the placards. The principal changes are as follows:

- The way widgets are displayed on the front of the store has been modified and now includes categories and ratings on the Favourites and Featured widgets sections.
- The layout of the store is also new with the left side navigation removed and tabs added to show the different areas of the store.
- The four buttons above the widget display area change the widgets that are displayed in this front page.
- My Favourites will be empty until you have chosen some favourites (see below).
- Most Recent show the 50 most recently created widgets with small icons.
- All Widgets shows all the widgets in the store as small icons.





### 3.2.2 Navigation

The usability research undertaken at the end of year 3 of the project showed the need to change the navigation. The new interface to the store offers tabs which take the user to the different sections of the store. These tabs are always displayed no matter which section you are in and they also indicate which section you are in by the highlight state.

### 3.2.3 Categories

Several different users from different user groups requested better searching and categories. The response to these requests was combined in a categories faceted search facility. Implementation of this has involved:

- Addition of twenty-four categories to the store. The creator of the widget can categorize widgets in the my widgets section which is described below.
- The categories section allows widgets to be discovered in a way which is intended to be more meaningful to the user, and perhaps simpler. The categories are divided into three groups.
- Within each group of categories the discovery is accumulative, and between the groups it is subtractive. This faceted approach allows users to tailor the discovery of the widgets to best suit their needs. It acts as way of filtering down to the subjects, skills and age ranges in which you are interested.




Widget Store
Categories
Favourites
My Widgets
Creator
Admin


**Subject**  
☐ Numbers  
☒ Words  
☐ Sciences  
☐ Arts  
☐ Humanities  
☐ Technologies

**Age**  
☒ Higher Education  
☒ 16 to 18  
☒ 11 to 14  
☐ 8 to 11  
☐ 5 to 8  
☐ Pre School  
☐ Infant

**21st Century Skills**  
☐ Creativity  
☐ Innovation  
☐ Critical Thinking  
☐ Problem Solving  
☐ Creativity  
☒ Decision Making  
☐ Communication  
☐ Collaboration  
☐ Research  
☐ Adaptability  
☐ Self-Direction

**Results**


**You decide**  
Words  
Not yet rated


**Ta-Da!**  
Higher Education  
Not yet rated

### 3.2.4 Favourites

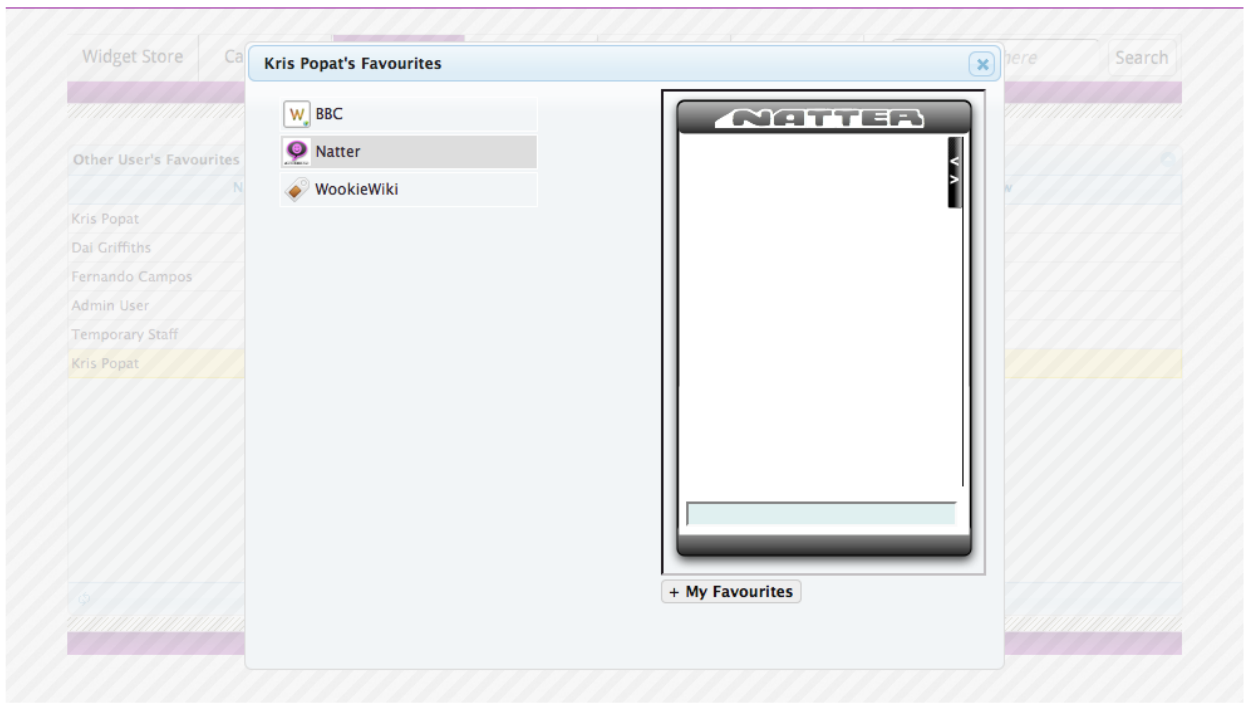
Discussions within the team about how the store had been used by the pilot groups led us to add a favourites facility, to support sharing between users. This section is where the user can see the favourites that other people have chosen. It shows a list of users. Clicking a Favourites button will bring up a dialog with a list of that user's favourites.

Widget Store
Categories
Favourites
My Widgets
Creator
Admin

**Other User's Favourites**

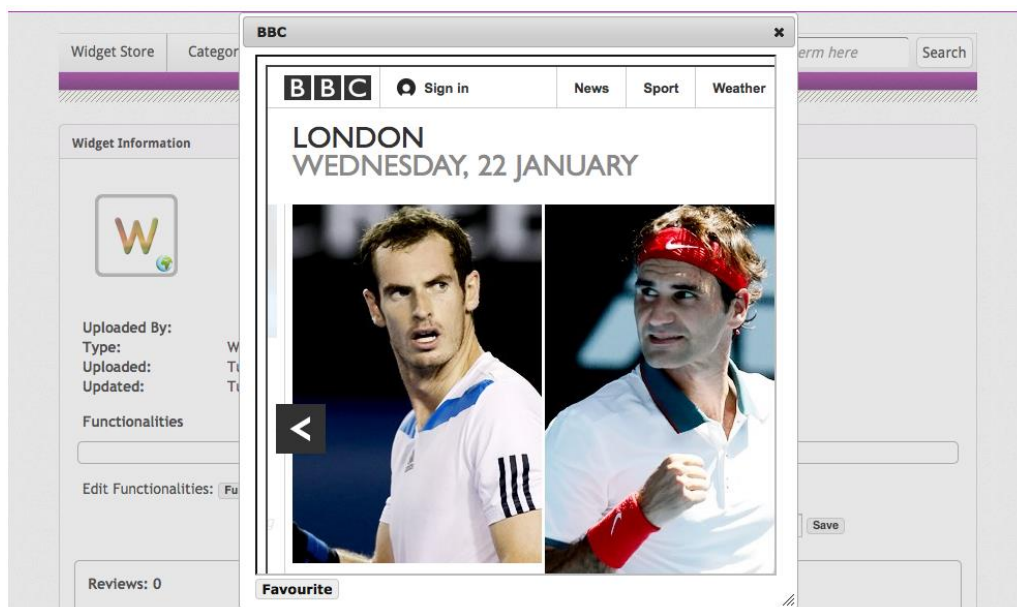
Name ↕	Since	View
Kris Popat	Fri Nov 15 2013	<input type="button" value="Favourites"/>
Dai Griffiths	Tue Nov 19 2013	<input type="button" value="Favourites"/>
Fernando Campos	Tue Nov 19 2013	<input type="button" value="Favourites"/>
Admin User	Thu Nov 21 2013	<input type="button" value="Favourites"/>
Temporary Staff	Fri Dec 06 2013	<input type="button" value="Favourites"/>
Kris Popat	Wed Jan 15 2014	<input type="button" value="Favourites"/>

Here the user can add a widget to their own favourites if they choose.

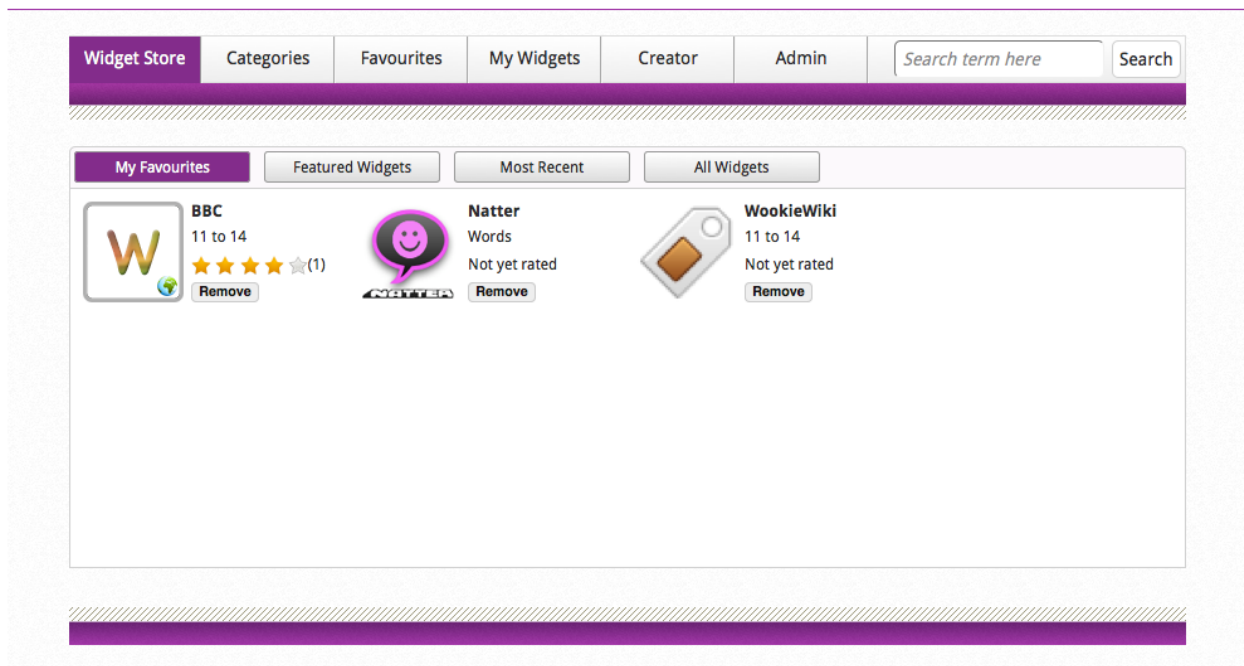


The user can add any widget to their favourites by viewing it on the widget view page. Clicking on a widget icon on the Widget Store front or from search results accesses this.

There is a button at the bottom of the widget viewer. Clicking this button will add the widget to your favourites and take you to the list of your favourites on the front page.

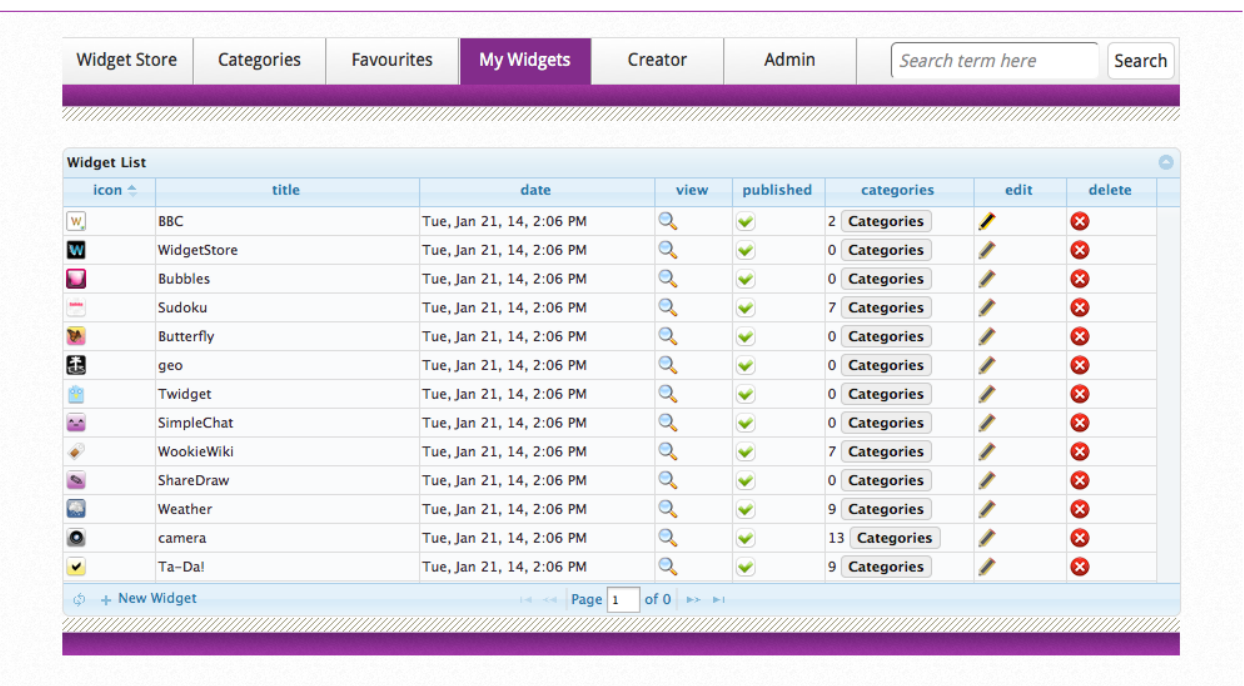


The user can also remove widgets from the favourites collection:



### 3.2.5 My Widgets

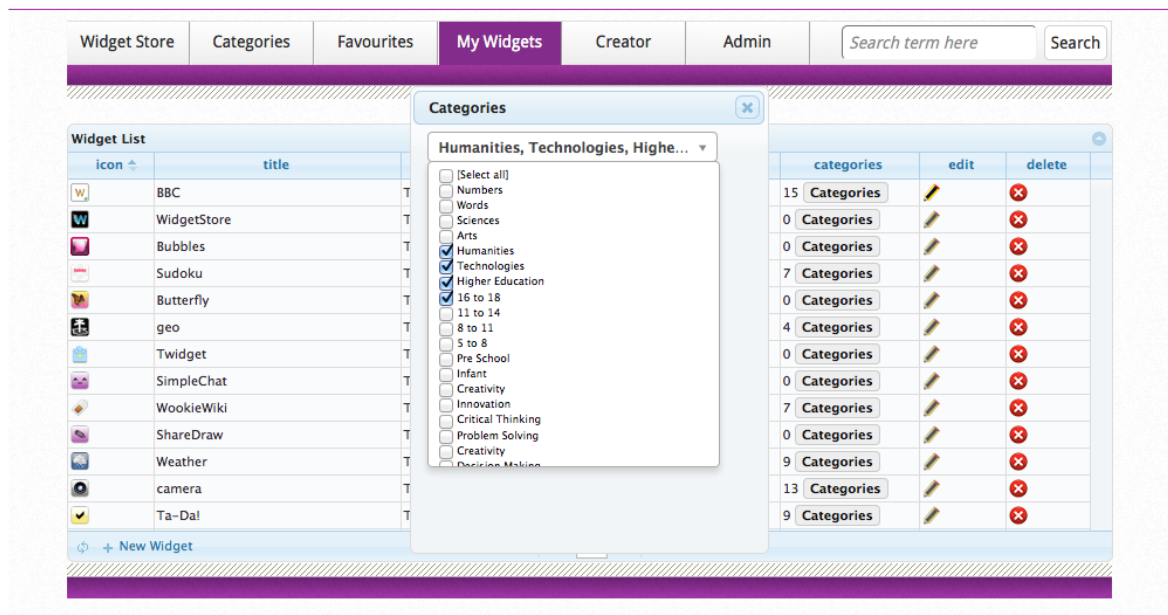
The “My Widgets” section has been completely changed since the last release of the store. The original version had on two tools. This was an attempt to give a more complete set of tools allowing the user to manage their own widgets. It now shows the widgets in a table list with tools to publish, categorize, edit and delete your widgets.



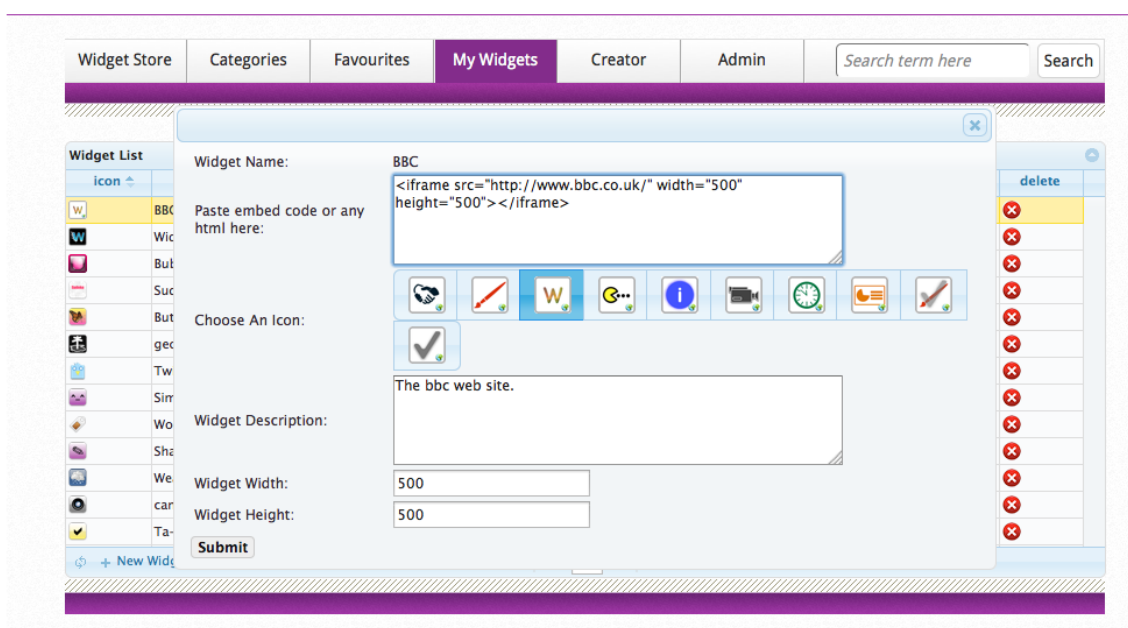
From here the user can view their widget.

The widget can be set as *published* or *unpublished*. In the final version of the iTEC Widget Store direct publication is in the hands of the user, following the iTEC philosophy of open, crowd-sourced, community based publishing. However, workflow controls exist within the system to insert an administrative check and stop for widget publication requests.

The user can categorize their widget with multiple categories. Clicking the categories button will bring up a dialog box in which they can add categories to a widget.



If a widget has been created by the user themselves, using the embedded or web address tool, they can now edit the widget. A click on the pencil for that widget brings up a form with the data originally entered for that widget which can then be changed and saved.



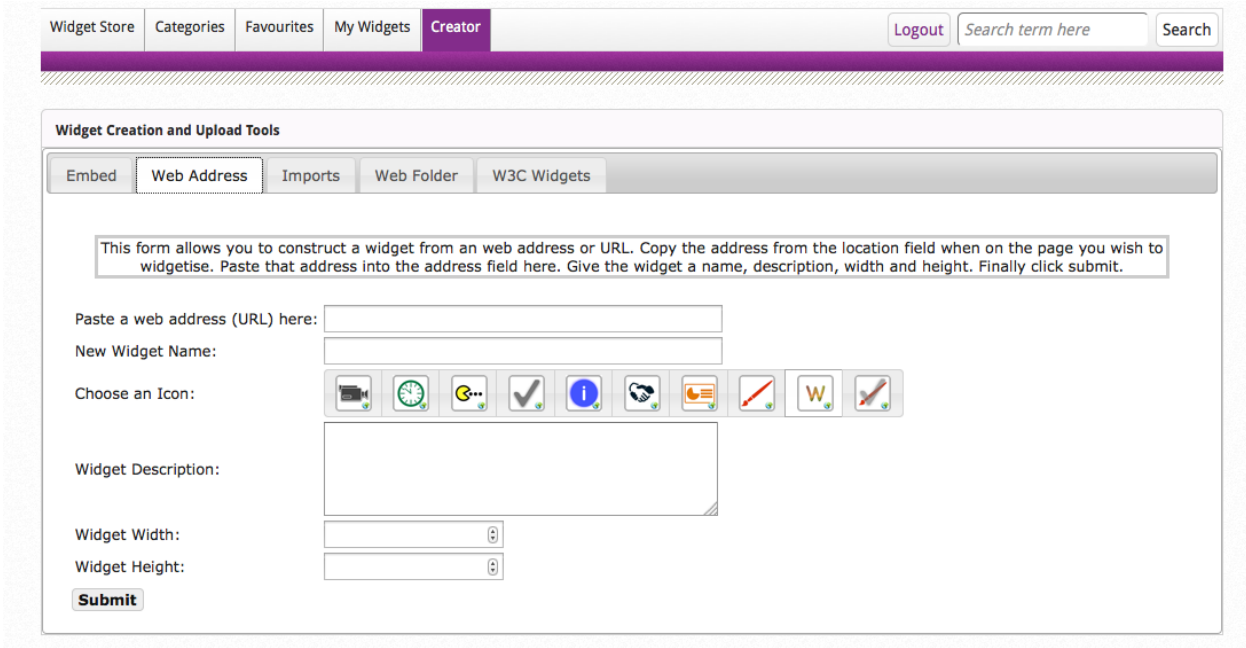
The only difference is that the name cannot be changed here.

### 3.2.6 Creator

In the previous version of the widget store this section was called “Upload”. This name was for historical reasons, as originally uploading new widgets was all you could do. In the present version, however, there are more widget creation tools than upload tools so the name was changed in line with this.

The original release version of the store had an embed tool, the imports and the W3C widget upload tool. Additional tools have been added. The Web Address tool creates a widgets from a web url. This effectively creates a mashup portal to another web site.

The web folder tool expects a zipped up folder that contains a mini web site. Meta information is required as with all of the tools to name the widget and specify a width and a height. The tool converts the site into a W3C widget and deploys it.

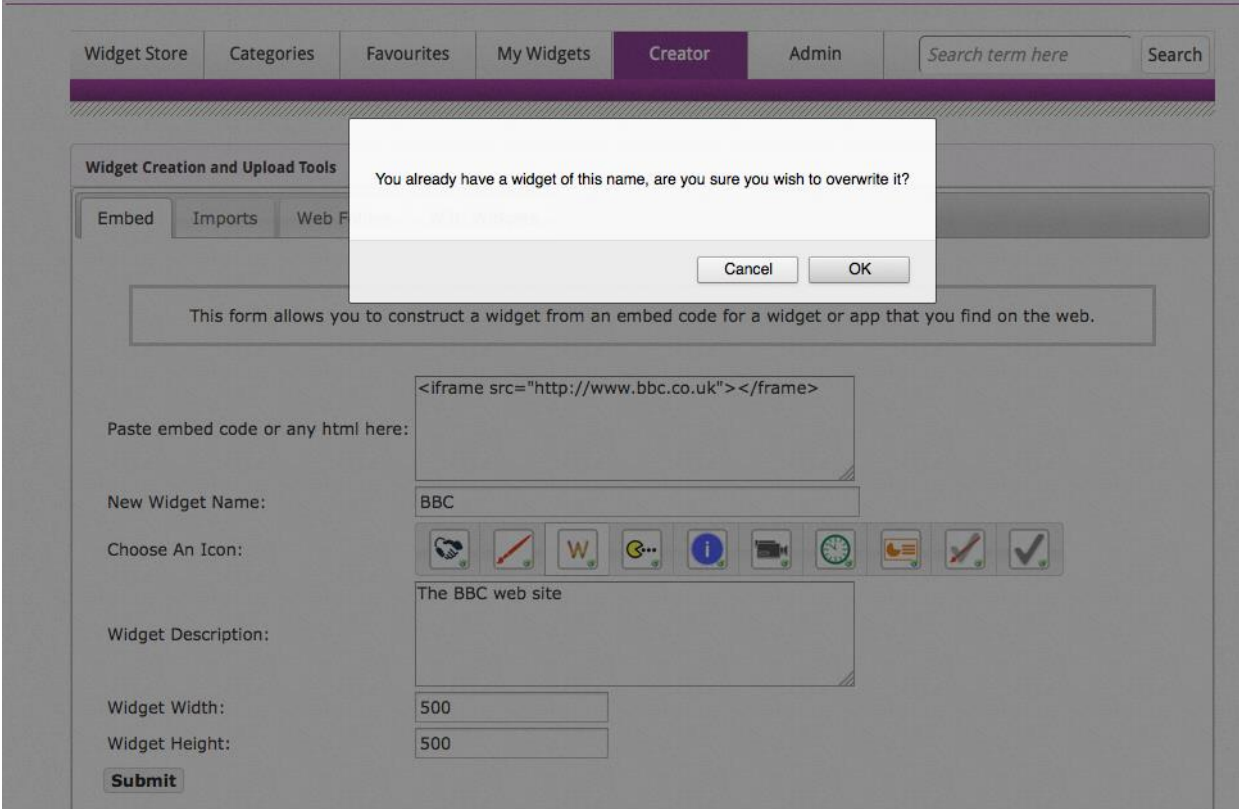


The screenshot shows the 'Widget Creation and Upload Tools' interface. At the top, there is a navigation bar with tabs: 'Widget Store', 'Categories', 'Favourites', 'My Widgets', and 'Creator' (which is highlighted). To the right of the tabs are links for 'Logout', a search bar with the placeholder 'Search term here', and a 'Search' button. Below the navigation bar, the 'Widget Creation and Upload Tools' section is displayed. It has a sub-navigation bar with tabs: 'Embed', 'Web Address' (which is selected), 'Imports', 'Web Folder', and 'W3C Widgets'. A text box explains: 'This form allows you to construct a widget from an web address or URL. Copy the address from the location field when on the page you wish to widgetise. Paste that address into the address field here. Give the widget a name, description, width and height. Finally click submit.' The form includes the following fields and elements:

- 'Paste a web address (URL) here:' followed by a text input field.
- 'New Widget Name:' followed by a text input field.
- 'Choose an Icon:' followed by a row of ten icons: a camera, a clock, a gear, a checkmark, an information icon, a handshake, a list, a pencil, a 'W' icon, and a document with a pencil.
- 'Widget Description:' followed by a large text area.
- 'Widget Width:' followed by a text input field with a small 'x' icon on the right.
- 'Widget Height:' followed by a text input field with a small 'x' icon on the right.
- A 'Submit' button at the bottom left.

There is, however, one major process change. You can now overwrite widgets you have created of the same name. A dialog box warning you that you are about to overwrite the widget comes up.





The screenshot shows the iTEC Creator interface. At the top, there is a navigation bar with tabs: Widget Store, Categories, Favourites, My Widgets, **Creator**, and Admin. To the right of the tabs is a search bar with the placeholder text "Search term here" and a "Search" button. Below the navigation bar, the main area is titled "Widget Creation and Upload Tools". It contains three sub-tabs: "Embed", "Imports", and "Web F". A modal dialog box is open in the center, displaying the message: "You already have a widget of this name, are you sure you wish to overwrite it?". The dialog has "Cancel" and "OK" buttons. Below the dialog, the "Embed" tab is active. It contains a text area for "Paste embed code or any html here:" with the code: `<iframe src="http://www.bbc.co.uk"></frame>`. Below this is a "New Widget Name:" field containing "BBC". Under "Choose An Icon:", there is a row of icons, including a hand, a pencil, a 'W' in a circle, a speech bubble, an 'i' in a circle, a video camera, a clock, a document, a pencil, and a checkmark. Below the icons is a "Widget Description:" field containing "The BBC web site". At the bottom, there are fields for "Widget Width:" (500) and "Widget Height:" (500), followed by a "Submit" button.

This works for Embed, Imports and Web Folder. The update management for W3C widgets is based upon the W3C specification for updating widgets based upon their id. A discussion about this is beyond the scope of these release notes.

At the moment it is recommended that, unless users are W3C widget developers, they should use the other tools to create widgets.

## 3.3 Server Side and Code Base Configurations

### 3.3.1 LTI

The widget store now supports native LTI apps. The administrator must set up the interface to this, as the control information for LTI is secret. LTI apps can be added individually in the LTI configuration files maintained on the server.

The store can be included in an LTI host application, the security information for LTI is contained on the store server in the store configuration files, which must be supplied to the host application for the embedded to work.

### 3.3.2 Language Support

Multiple languages are supported in the user interface for the store. This is a UI translation and does not have any bearing on the language of the content of the store.

The configuration of different languages is within the code of the widget store user interface so some programmer support is required for this at least at the level of recompiling the user interface widget. The language translations are implemented as javascript language files which are in a specific folder in the store ui hierarchy: <widgetstore\_base>/js/lang/*language\_code*.js.

The filename *language\_code* refers to the ISO 639-1 language code specifications so the language file for English, for instance, is called “en.js”.

**Widget Store Development Team - The University of Bolton**

**August 2014**

<http://itec.eun.org>

### **Coordinated by European Schoolnet**

The work presented in this document is co-funded by the European Commission's 7<sup>th</sup> Framework Programme – project iTEC: Innovative Technologies for an Engaging Classroom (Grant agreement N° 257566). The content of this document is the sole responsibility of the consortium members and it does not represent the opinion of the European Commission and the Commission is not responsible for any use that might be made of information contained herein.



Co-funded by the  
7th Framework Programme  
of the European Union