



D7.1: BRIDGET Authoring Tools and Player –Report – Version A

Project ref. no.	FP7-ICT-610691
Project acronym	BRIDGET
Start date of project (duration)	2013-11-01 (36 months)
Document Date	2014-12-19
Version/svn revision	D7.1_final.docx
WP / Task responsible	WP7
Leader of this document	Giovanni Cordara
Other contributors	Nicola Piotto, Sergio García Lobo, Francisco Morán Burgos, Davide Bertola, Leonardo Chiariglione, Peter Grosche, Marius Preda, Milos Markovic, Alberto Messina, Adrian Gabrielli, Veronica Scurtu
Reply to	
Document status	Final
Document name	D7.1: BRIDGET Authoring Tools and Player – Report – Version A
Number of pages	41
Abstract	According to BRIDGET's DoW, the current D7.1 deliverable is a report to be delivered by 2015-04-30 describing version A of BRIDGET's Authoring Tools and Player, and presenting a progress report of the work

	conducted within WP7 in the 2013-11-01/2015-04-30 reporting period. D7.1 is accompanied by D7.2, that contains the corresponding SW.
Keywords	WP7, BRIDGET Authoring Tools, BRIDGET Player, Media Rendering

Revision History

Version	Date	Reason of change
0.1	2014-12-03	Giovanni Cordara (HUA) – Document structure
0.2	2014-12-10	Sergio García Lobo and Francisco Morán Burgos (UPM) – Section 4.3
0.3	2014-12-11	Peter Grosche and Milos Markovic (HUA) - Sections 4.1 and 4.2
0.4	2014-12-13	Leonardo Chiariglione and Davide Bertola (CED) – Sections 3.2 and 5.2
0.5	2014-12-16	Marius Preda (IMT) – Sections 2.1, 2.2, 3.1,3.3, 5.1 and 6
0.6	2014-12-17	Giovanni Cordara (HUA) – Section 1, intergation and first review
0.6.1	2014-12-17	Albert Messina (RAI) – Section 2.3
0.7	2014-12-18	Francisco Morán Burgos (UPM) – second review
0.8	2014-12-18	Marius Preda (IMT) – additions to Sections 2.1, 3.1 and 5.1
0.9	2014-12-19	Sergio García Lobo and Francisco Morán Burgos (UPM) – additions to Section 4.3
1.0	2014-12-19	Giovanni Cordara – final review of Interim Report
1.1	2015-03-30	Giovanni Cordara - First draft of the deliverable (incrementally extending Interim Report)
1.2	2015-04-04	Leonardo Chiariglione – Davide Bertola – update of sections 3.2, 3.4, 5.2
1.3	2015-04-08	Alberto Messina – update of section 2.3
1.4	2015-04-10	Marius Preda, Adrian Gabrielli, Veronica Scurtu – update of sections 2.1, 2.2, 3.1, 5.1
1.5	2015-04-10	Milos Markovic – update of section 4.2
1.6	2015-04-11	Giovanni Cordara – integration and first clean-up
1.7	2015-04-19	Milos Markovic – refinement of section 4.2
1.8	2015-04-19	Veronica Scurtu and Adrian Gabrielli – refinement of section 2 and 3
1.9	2015-04-20	Francisco Moran Burgos and Sergio Garcia Lobo – overall review and refinement of section 4.3
2.0	2015-04-22	Integration of new contribution, and sanity check – draft delivered for QAT
2.1	2015-04-29	Ingo Feldmann – internal review
Final	2015-04-30	Giovanni Cordara – final update based on internal review

Table of Contents

1	Introduction	7
2	T7.1 - User Interfaces and Bridget Presentation	8
2.1	Design Principles.....	8
2.1.1	Bridgets User Interface Design	8
2.1.2	Authoring Tool User Interface Design	12
2.2	User Interface Implementation	19
2.2.1	News UI.....	20
2.2.2	Documentary UI	21
2.2.3	Entertainment Show UI.....	21
2.3	Ongoing Usability Experts Analysis.....	22
3	T7.2 - Authoring Tools.....	23
3.1	Architecture Design and Implementation.....	23
3.2	Authoring Tool Frontend	24
3.3	Authoring Tool Backend	26
3.4	A BRIDGET Authoring Tool for Internet Video Distribution.....	28
4	T7.3 - Immersive Media Rendering	30
4.1	Audio Fingerprint Technology.....	30
4.1.1	Implemented Approach	31
4.1.2	Performances	32
4.2	Optimizations for Binaural Playback on Mobile Devices.....	32
4.2.1	Audio engine Demo User Interface	33
4.2.2	Definition of 3D Audio Engine Interfaces.....	34
4.3	Optimized Rendering of Synthetic 3D Models	34
5	T7.4 - Multi-screen player	36
5.1	Architecture Design and Implementation.....	36
5.2	A BRIDGET Player for Internet Video Distribution.....	39
6	T7.5 – Standardization	41
7	Conclusions	41

Table of Figures

Figure 1: Linear vs bridged consumption experience.....	11
Figure 2: In real situations bridgets are overlapping.....	11
Figure 3: Illustration of a bridget.....	12
Figure 4: Conceptual design of the AT UI	13
Figure 5: Conceptual design of the AT UI for Programmes overview	14
Figure 6: Conceptual design of the AT UI for Programmes editing	14
Figure 7: Conceptual design of the AT UI for Programme creation.....	15
Figure 8: Conceptual design of the AT UI for editing an existing bridget	16
Figure 9: Conceptual design of the AT UI for creating a new bridget.....	16
Figure 10: Conceptual design of the AT UI for visual search enrichment.....	17
Figure 11: Conceptual design of the AT UI for 3D reconstruction	18
Figure 12: Conceptual design of the AT UI for editing the bridget presentation layout.....	18
Figure 13: Current RAI programs selected to BRIDGET experiments.....	19
Figure 14: Identification of the hot spots and collection of relevant content	19
Figure 15: Illustrations of the first UI design as presented on the tablet for one of the RAI programs	20
Figure 16: Illustrations of the second UI design as presented on the tablet for the news report.....	20
Figure 17: Illustrations of the second UI design as presented on the tablet for a documentary on Torino architecture.....	21
Figure 18: Illustrations of the second UI design as presented on the tablet for a TV entertainment show	22
Figure 19: BRIDGET AT frontend and backend architecture.....	24
Figure 20: Video is played to find source content for a bridget.....	25
Figure 21: A bridget is created.....	25
Figure 22: Images suggested by CDVS search	26
Figure 23: Viewing all bridgets in a program	26
Figure 24: Authoring tools database structure.....	28
Figure 25: Current WimTV architecture.....	29
Figure 26: WimTV extended with new bridget-oriented applications	29
Figure 27: Overview of the BRIDGET Audio Fingerprint Engine	31
Figure 28: Block diagram of the audio rendering engine based on binaural synthesis	32
Figure 29: User interface for spatial audio rendering of a dynamic scene.....	33
Figure 30: Snapshots of different the BRDIGET Player for Android: a traditional point cloud and its derived splat model (top), plus a full view and a close-up of a splat-based 3D model (bottom).....	36
Figure 31: Full views of different splat-based 3D models rendered in the BRIDGET Player for Android	36
Figure 32: AFP Architecture	37
Figure 33: BRIDGET AFP PROTO	37
Figure 34: RemAud PROTO (ARAF).....	39
Figure 35: A model for internet distribution of video programs enriched with bridgets.....	39
Figure 36: A user experience of internet distribution of video programs enriched with bridgets.....	40
Figure 37. The screen of the WimView+ mobile app.....	41

Acronyms

AAC:	Advanced Audio Coding
AFP:	Audio Fingerprint
ARAF:	Augmented Reality Application Format
AT:	Authoring Tool
AVC:	Advanced Video Coding
BIFS:	BIInary Format for Scenes
DoW:	Disposition of Work
GUI:	Graphical User Interface
HEVC:	High Efficiency Video Coding
HRTF:	Head Related Transfer Functions
JPEG:	Joint Picture Experts Group
MLAF:	Media Linking Application Format
MPEG:	Moving Picture Experts Group
MXM:	MPEG Extensible Middleware
PNG:	Portable Network Graphics
UI:	User Interface

1 Introduction

The current **D7.1 deliverable**, due to 2015-04-30, describes version A of BRIDGET's Authoring Tools and Player and presents a report of the work carried out within WP7 in the first 18 months of the project. D7.1 is released together with D7.2, containing the corresponding SW and related documentation.

The current deliverable extends and completes the **interim** deliverable D7.1i, prepared by the work-package partners in January 2015 as a supporting material for the first year's review.

Within the BRIDGET project, WP7 is assigned a threefold scope: *i)* designing seamless and effective user interfaces (Task -T- 7.1) for BRIDGET tools; *ii)* conducting research on innovative media rendering technologies, with particular emphasis on low complexity methods well suited to mobile devices (T7.3); and *iii)* integrating the results of WP4-6 research into a unified software/hardware architecture (T7.2 and T7.4).

The main results of this first 18 project months for each of those directions can be summarized as follows:

- **Conceptual design of BRIDGET User Interface (T7.1):** BRIDGET is introducing a new concept for fruition of second screen content. This requires the design of a novel dedicated UI for enjoying such content effectively, without interfering with the main content displayed on the TV. Section 2 describes the studies conducted in order to design such user experience, maximize user appreciation and provide an easily customizable interface. Usability experts were consulted for these tasks; they reviewed several prototypes of the UI, providing feedbacks and suggestions that will be taken into consideration for next releases of the BRIDGET tools.
- **Research on low complexity media rendering (T7.3):** several optimizations have been studied, in order to enable real-time rendering of advanced media formats, namely synthetic 3D models and binaural audio, in resource-constrained environments such as tablets and other mobile devices. In both cases, algorithmic optimizations were necessary, as well as the introduction of new data formats, in order to make the fruition of those formats possible within the BRIDGET player. Progress report on the conducted work is reported in Section 4.
- **Development of first version of BRIDGET tools:** in line with the DoW schedule, first releases of the BRIDGET Authoring Tool (AT) and BRIDGET Player have been completed and integrated into a full platform architecture (Sections 3 and 5) as designed by WP3 and described in D3.1, "BRIDGET System Architecture and Interfaces". The **first release of the Authoring Tool (T7.2)** integrates a vast number of functionalities, namely: bridget creation, possibly also based on templates, usage of hierarchical temporal segmentation and audio fingerprinting in order to set up timings and triggers for bridget presentation, visual search to detect relevant images from image archives, encapsulation and compression of bridget-enriched content into bitstreams compatible with MPEG's Augmented Reality Application Format (ARAF; formally, ISO/IEC 23000-13 [1]). The **first release of the BRIDGET multi-screen player (T7.3)** is able to play bridgets encapsulated into an ARAF format, and render media content, 3D models (in the splat format defined in the project) and metadata. Synchronization with the main broadcast content is guaranteed by the integration of the audio fingerprint engine. As an important highlight of this reporting period and first step towards BRIDGET commercial exploitation, the integration of the BRIDGET architecture and player into an existing Web TV platform (WimTV) [2] has been carried out and was demonstrated at Y1 review.

In summary, the structure of the present document reflects the organization foreseen for WP7 in the DoW: for each task, main achievements are described, together with a report of the progress obtained in the first project year. In particular, section 2 describes the design of the user interfaces developed for the project (for bridgets and AT). Section 3 is dedicated to the architectural description of AT frontend and backend, whereas section 4 reports the progress achieved so far on the study of innovative media rendering algorithms. Finally, section 4 describes the working principles of the Bridget Player and section 5 the contributions to standards.

2 T7.1 - User Interfaces and Bridget Presentation

This task was active in this reporting period between January 2014 and April 2015.

Task 7.1 was the first one starting in WP7 and, indeed, at the beginning of the project major attention was devoted to the design of the BRIDGET user experience, as a new concept of fruition of second screen applications.

Within this reporting period, first concepts of the UIs for bridget production (AT) and consumption (Player) have been designed and integrated into BRIDGET architecture. Such concepts have been continuously refined, thanks to the feedbacks received by usability experts. Starting from these design rules, exemplary bridgets addressing the use cases defined in WP2 and the proof-of-concept studied in WP8 were implemented aiming at optimizing user experience for the audience targeted in each case.

Particular attention has been devoted to the usability, in particular on the player, in order to enable pleasant and effective bridget consumption without, at the same time, interfering with linear content displayed on the main screen.

Section 2.1 illustrates the studied design principles, while Section 2.2 describes the implementation details of the current UIs. As mentioned before, such UIs were already shown to usability experts, whose feedbacks are reported in Section 2.3.

2.1 Design Principles

2.1.1 Bridgets User Interface Design

A study carried out by Google¹ shows that 77% of TV viewers are using an additional screen while watching TV. Such results imply that the BRIDGET solution will promote second screen environment even further.

Despite the fact that second screens emerged in the last years as an important form of content consumption, the research concerning the user experience in this area is limited. Much of this previous work has been focusing on investigating the behaviour of the user during the consumption of second screen content, concerning mainly the social aspects and how the user attention is split. The most relevant work in this field is by Holmes et al., who used a head-mounted eye-tracker to monitor the attention of people viewing a drama or documentary programme with an 'app' running on a second-screen [3]. Some research also focused on examining the usability of dual screens systems [4][5][6].

The BBC has a dedicated Research & Development Department that has conducted several second screen experiments, one of which was launched a couple of years ago for the nature show Autumnwatch². The difference with this app was that it is created for a laptop and not a mobile device. Some of the recommendations that emerged from their experiment include the following:

- An additional content should be presented on a single short page/screen, without any scrolling and including simple interactions and navigation;
- The timing of the content is critical and the synchronisation should be as good as possible;
- The additional content presented on the second screen should be relevant;
- Create rich media experiences that mirror the multitasking behaviour of the users.

This study report proves to be extremely relevant for BRIDGET scenarios as well. In fact, one of the BRIDGET goals is to develop innovative functionalities to enjoy multimedia content by connecting it to related content, augmenting it with virtual information of interest, and helping navigate the 3D reconstruction of the captured scene. The way in which such functionalities are presented to the user on the second screen (e.g., a tablet) represents one of the differentiating factors for the quality of experience. Therefore, the design of a specific ergonomic UI for the BRIDGET Player represents an important requirement, that was addressed since the beginning of the project.

¹ http://services.google.com/fh/files/misc/multiscreenworld_final.pdf

² <http://www.bbc.co.uk/blogs/legacy/researchanddevelopment/2010/11/the-autumnwatch-tv-companion-e.shtml>

One of the main drivers for designing BRIDGET UI concerns the user attention that has to be alternatively dedicated to the first and second screens. This entails to carefully design how and when the content on the second screen is displayed. The second screen should be used as a natural continuation and extension of the experience, and not as a means of interrupting the experience. Additionally, the interactive experiences on the second screen should be short enough in order to allow the user resynchronize with the main story.

The transformation of the traditional linear experience of consuming the TV content is illustrated in Figure 1. Here, one single bridget is illustrated: S_1 is the start time and E_1^1 the end time if the user is not interacting with the content. If there is interaction, then the bridget end time can be any of E_1^n ($n > 1$). One may observe that selecting the starting time (S_1) is fundamental for a proper experience, and this can be set by the content designer. The same situation occurs for E_1^1 , while for E_1^n ($n > 1$) the content designer has no direct control. However, he can design the bridget in such a manner that E_1^n ($n > 1$) will be in a certain range.

Linear content

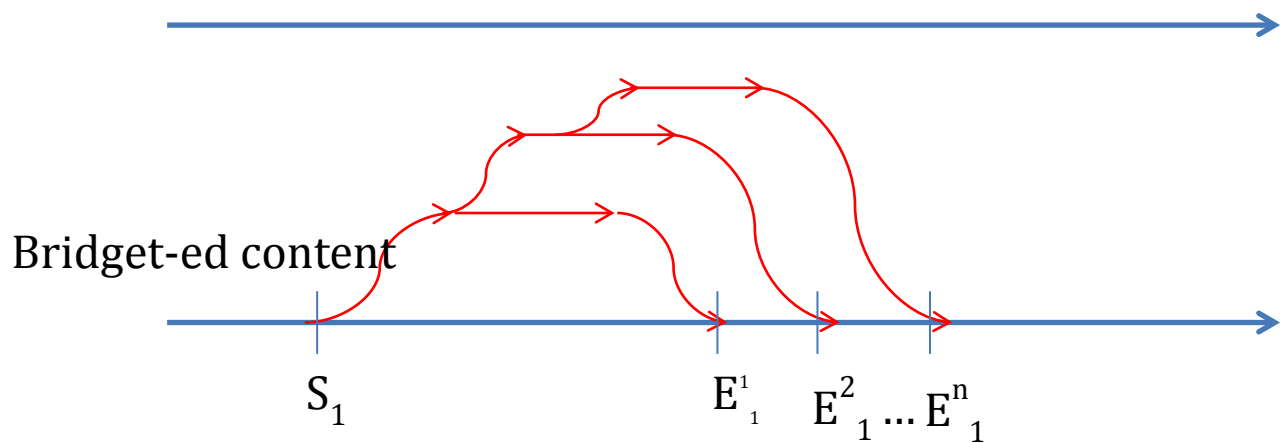


Figure 1: Linear vs bridgeted consumption experience

The situation is much more complicated in real cases. This is mainly due to the fact that several bridgets are designed for a program and that overlapping may occur (as illustrated in Figure 2).

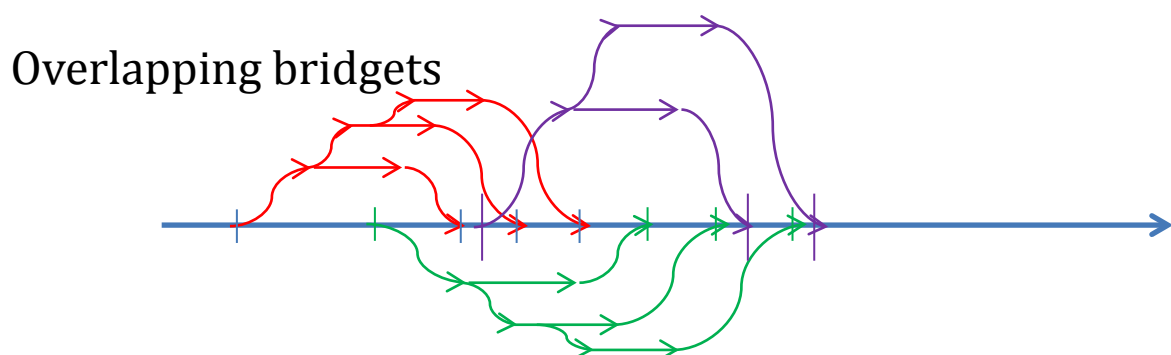


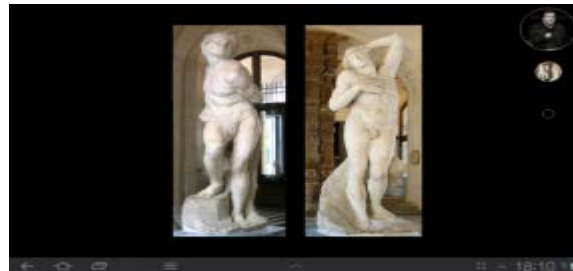
Figure 2: In real situations bridgets are overlapping

This temporal analysis made us derive the first set of rules for designing bridgets:

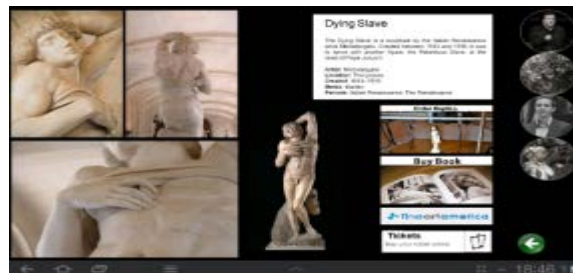
- A bridget should start with one screen presenting the main topic (enough information but comprehensible in one second);
- The complex bridgets are interactive – details are made available "on click" (higher resolution images, videos, text, 3D graphics, web-sites etc.);

- Bridgets have a life time;
- A special view should indicate what bridgets are available at current time.

Remarkably such design principles are strongly correlated with those emerging from the BBC study, though meant for being applied in general purpose scenarios on mobile devices. These rules were implemented in several prototypes: we present in Figure 3 three screens for a bridget providing additional information related to the Dying Slave sculpture (the main screen is showing a TV program about the Louvre museum).



a) a bridget main screen



b) additional data presented "on click"



c) enlarged view on a media component of one bridget, also available "on click"

Figure 3: Illustration of a bridget.

The activity on this task is continuing with the objective of extracting additional design rules for presenting bridgets.

During the first 18 months of the project, several bridget UIs were developed in order to test various features and functionalities of bridget presentation and consumption. These UIs are presented in detail in section 2.2.

2.1.2 Authoring Tool User Interface Design

During the first 18 months of the project, an additional activity was carried on within task 2.1, namely the design of the professional AT UI. Based on the analysis of the production workflow, several conceptual designs were proposed and discussed with the consortium partners and usability experts (section 2.3). At the end of this study phase, we structured the AT UI into four design layers as presented in Figure 4.

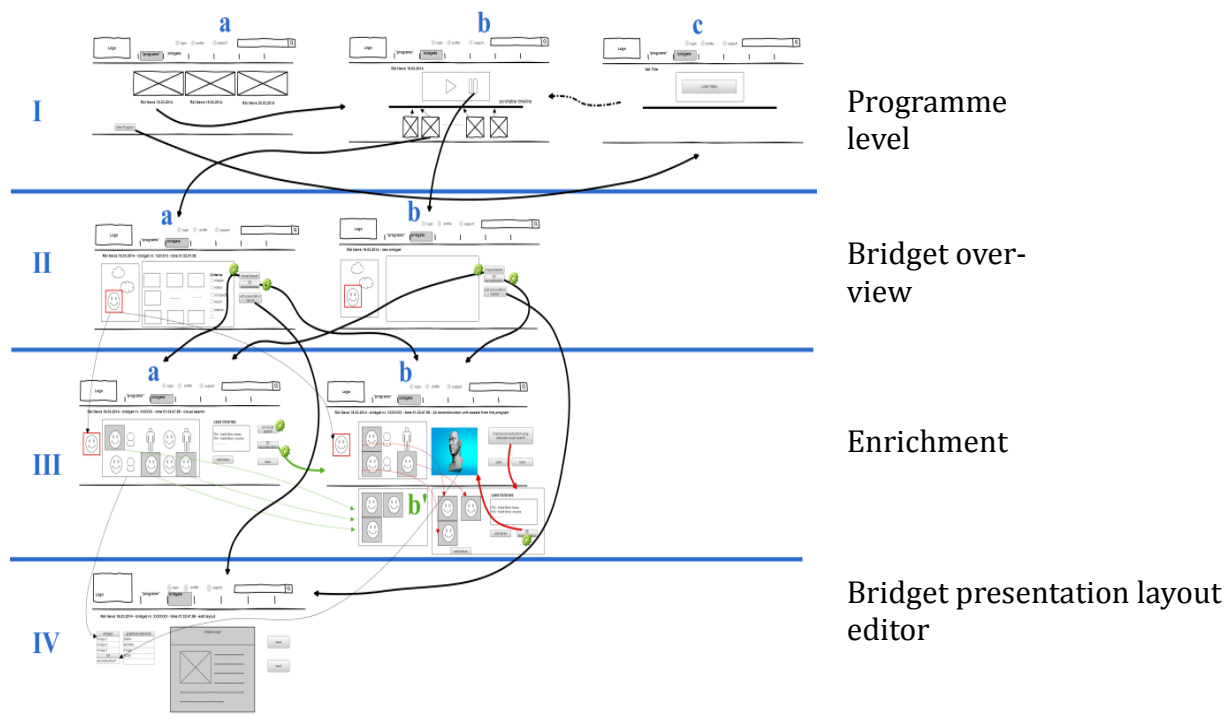


Figure 4: Conceptual design of the AT UI

- I. The top level is the "*Programme level*". Here, the professional designer retrieves the entire main media (audio-video) and indications where existing bridgets are positioned (if any). He can also select manually a position (frame or shot) that he wants to enrich. Several functionalities are supported at this layer, through different interfaces:
 - a *Programme overview* (see Figure 5, interface *I.a.* is presented to professional designer):
 - The professional designer is presented all the available Programmes (that are accessible based on user permissions);
 - b The professional designer has the option to enrich one of them or add a new Programme to be enriched. The *Programme editing* interface (see Figure 6, interface *I.b.*):
 - Presents the professional designer the Programme's video and the corresponding timeline containing references to pre-existing bridgets;
 - Allows the professional designer to select one of the pre-existing bridgets for editing;
 - Allows the professional designer to pause the video, and select a frame or a segment for enrichment – and then add a new bridget;
 - Allows the professional designer to edit the Programme's details (broadcast title, broadcast date, broadcast time, etc.).
 - c *Programme Creation* (see Figure 7, interface *I.c.* is presented to professional designer):
 - The professional designer is allowed to load a new video associated to a Programme and specify the Programme's details (broadcast title, broadcast date, broadcast time, etc.);

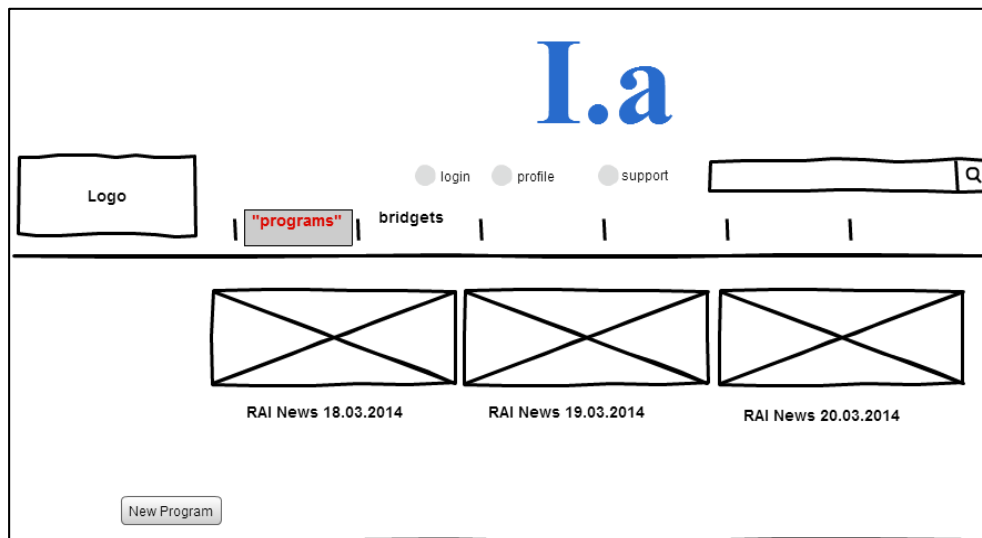


Figure 5: Conceptual design of the AT UI for Programmes overview

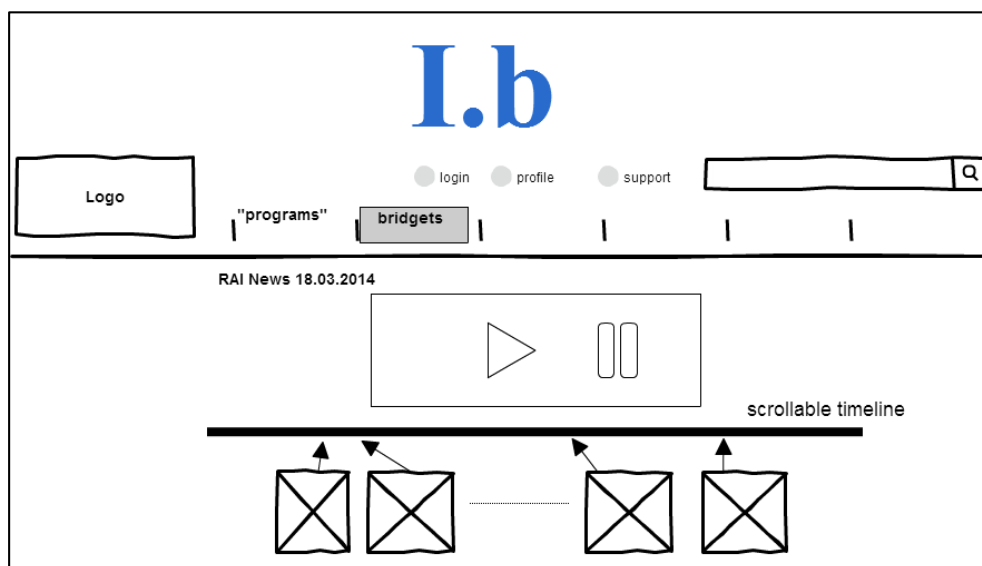


Figure 6: Conceptual design of the AT UI for Programmes editing

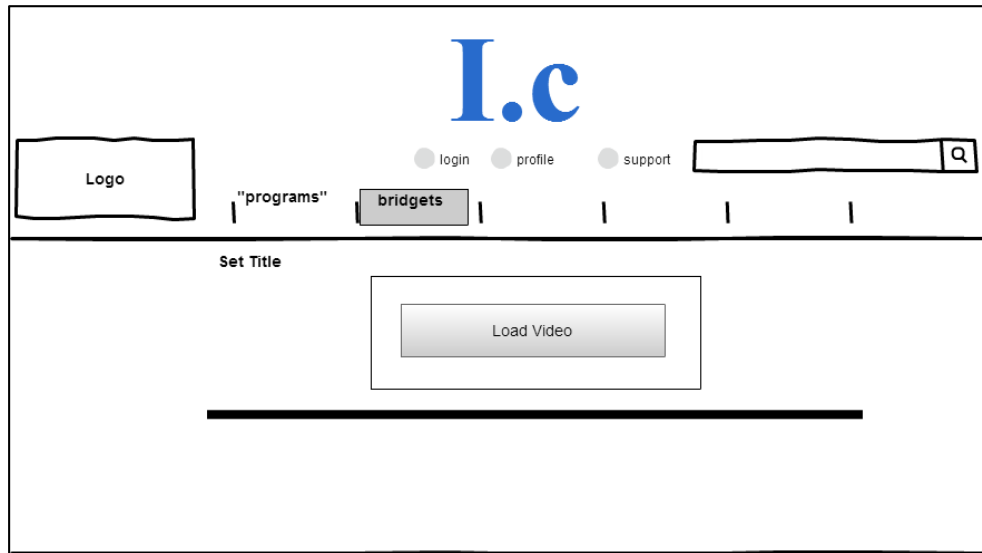


Figure 7: Conceptual design of the AT UI for Programme creation

- II. At the second level, the "*Bridget Overview*" level, all the information related to a particular bridget is presented. It is possible here to create a new bridget or to edit an existent one.
- a *Editing an existing bridget* (see Figure 8, interface *II.a.* is presented to professional designer):
 - The professional designer is presented with the corresponding segment having already selected the region used for enrichment;
 - The professional designer is presented with the associated media (i.e. the bridget destination content) already used for enriching the selected region (images, 3D objects). The professional designer has also the possibility to browse through this associated media by category and search by metadata;
 - The professional designer has the option of running another visual search based on the selected region from the current segment;
 - The professional designer has the option of running the 3D reconstruction based on a set of images obtained by visual search using a default set of indexed content repositories;
 - The professional designer has the option of editing the bridget's presentation layout (the information displayed to the user when selecting the bridget from the Bridget Player in the end user's BRIDGET Application).
 - b *Creating a new bridget* (see Figure 9, interface *II.b.* is presented to professional designer):
 - The professional designer is allowed to select the region for enrichment from the selected video segment;
 - The professional designer is presented with the same options as *II.a.*

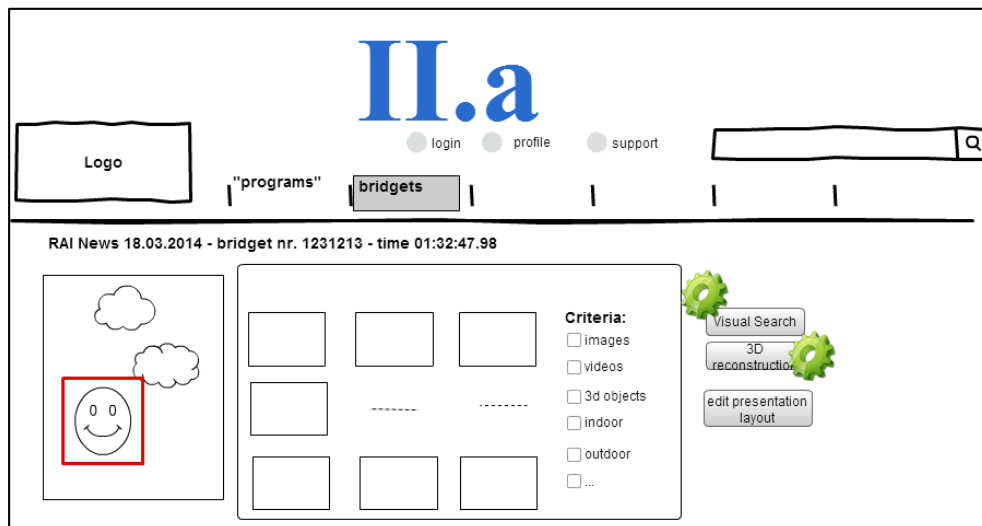


Figure 8: Conceptual design of the AT UI for editing an existing bridget

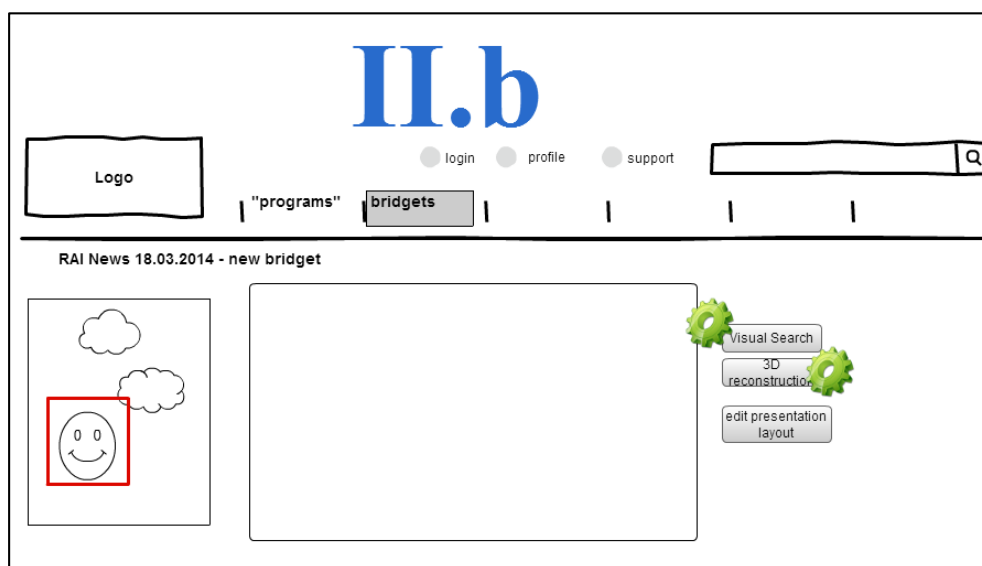


Figure 9: Conceptual design of the AT UI for creating a new bridget

III. The third level, the "*Enrichment*" is reached when additional media repositories are scanned to find similar images and when the set of similar images are used to reconstruct a 3D model.

a *Visual search* (see Figure 10, interface *III.a*):

- The professional designer is presented with the segment selected for enrichment;
- The professional designer is presented with the list of already selected images obtained from visual search and validated by the author;
- The professional designer has the option of adding or removing repositories for visual search and re-running the visual search on the new set of repositories;
- The professional designer has the option of running the 3D reconstruction on a selected set of images obtained from visual search;
- The professional designer has the option to save the images as destination content of the bridget.

b *3D reconstruction* using the default visual search repositories (see Figure 11, interface *III.b*):

- The professional designer is presented with the segment selected for enrichment;

- The professional designer is presented with the list of images obtained from visual search (using the default repositories for visual search) and used for generating the 3D object;
 - The professional designer has the option of improving the reconstruction by adding extra repositories for visual search or adding specific images;
 - The professional designer has the option of selecting specific images from the visual search results to be used in the reconstruction;
 - The professional designer is presented with the resulting 3D reconstruction;
 - The professional designer has the option to save the 3D reconstruction as a destination content of the bridget.
- b' *3D reconstruction* using selected images obtained from running the visual search algorithm (see Figure 11):
- The professional designer is presented with the images selected from the results of the visual search algorithm;
 - The professional designer has the same options as for *III.b*.

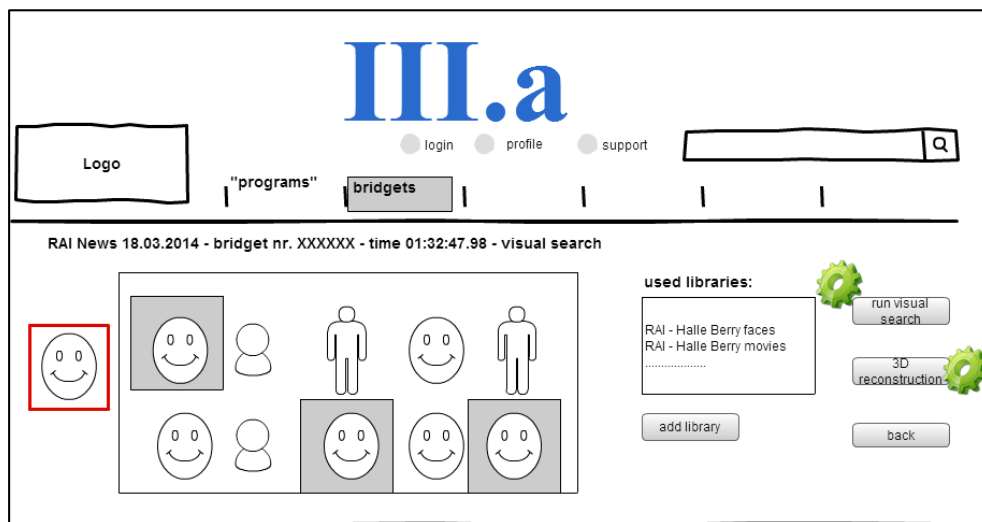


Figure 10: Conceptual design of the AT UI for visual search enrichment

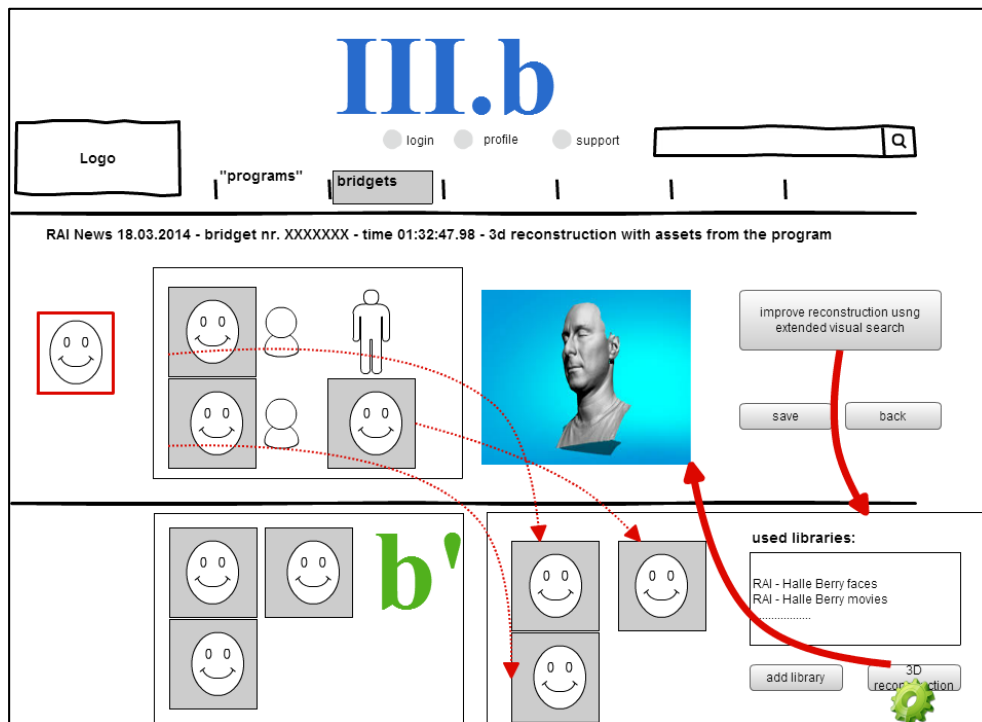


Figure 11: Conceptual design of the AT UI for 3D reconstruction

IV. The forth level "*Bridget presentation layout editor*" (illustrated in Figure 12, interface IV) is the one allowing to edit the layout of a bridget (selecting the interactive behaviour of the bridget as well as the spatial and temporal arrangements of the media elements composing the bridget).

- The professional designer is presented with a special area which allows manipulation (drag and drop, resize, etc...) of images, 3D objects, buttons and text fields;
- The professional designer is presented with a list of associated images and 3D objects obtained from visual search and 3D reconstruction;
- The professional designer is presented with a list of widgets that can be added to the layout: button, text field, image, audio, etc.

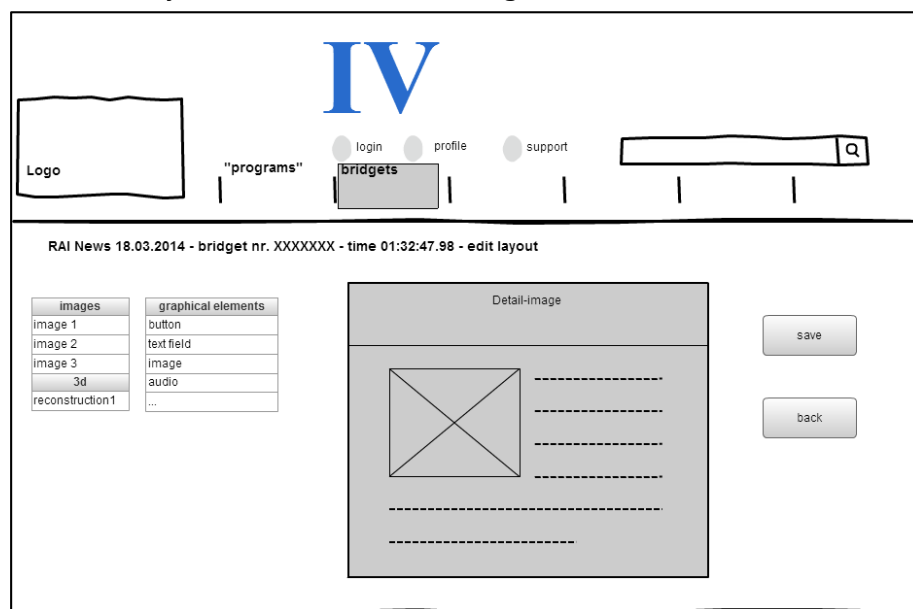


Figure 12: Conceptual design of the AT UI for editing the bridget presentation layout

The activity on this task is continuing and in collaboration with T7.2, whose objective is to implement, test and validate these concepts with professional content producers.

Snapshots of the implemented AT UIs are displayed in section 2.2.

2.2 User Interface Implementation

The design guidelines summarized in the previous section were used to enrich the consumption experience with second screen content for four use cases, provided by RAI and representing various kinds of programs: news, a documentary on Torino architecture, a TV talk show about the Concordia accident, and a TV entertainment show. These programs in combination with the related UI design will be introduced in more detail in the following sub-sections.

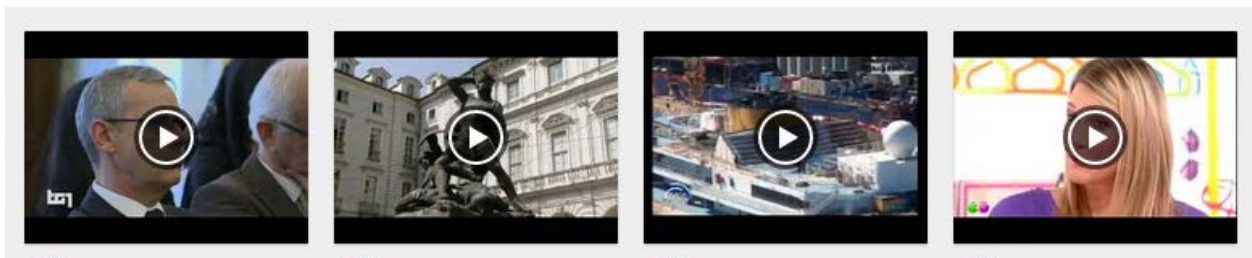


Figure 13: Current RAI programs selected to BRIDGET experiments

For each program, a detailed analysis of the content was performed, interesting (temporal) hot points were detected and related content was collected from various sources. Figure 14 shows this enrichment process for the News program.






Content	Speakers	Where ?	Television report done by
The letter : http://www.mef.gov.it/documenti-allegati/2014/Letter_Padoan_final.pdf	Pier Carlo Padoan  is an Italian economist and politician. Mr. Pier Carlo Padoan has been a Deputy Secretary-General of Organisation for Economic Co-operation and Development (OECD) since June 2007. Mr. Padoan is responsible for developing the strategic vision of OECD, the innovation strategy and its strategic response to the economic crisis. He is In Charge of OECD's relations with other international organizations, as well as local development, small and medium-sized enterprises, trade and agriculture, science and technology, and tax issues. He is Minister of Italy	5:48	- Edition : Gianpiero Scarpati  - Service : Chiara Anselmi - Documentation : Daniele Vergari - Graphics : Grazia Pietrasanta
	Graziano Delrio  is an Italian medical doctor and politician, who served as minister for regional affairs and autonomy from late April 2013 to February 2014. He is the state secretary to the Prime Minister and the mayor of Reggio Emilia. Links : https://twitter.com/graziano_delrio	7:00	- Edition : Marco Frittella  Links : http://it.wikipedia.org/wiki/Marco_Frittella https://twitter.com/mfrittella
	Sergio Chiamparino  is the current President of Piedmont from 2014, and was the mayor of Turin, Italy from 2001 to 2011. A graduate in political sciences at the University of Turin, where he worked as a researcher until 1975, Chiamparino started his political career that same year as head of the Italian Communist Party in the Town Council of Moncalieri, his native city. He joined the Democratic Party of the Left on its formation and was elected to the Chamber of Deputies in 1996, following a surprise defeat in 1994 to the centre-right candidate Alessandro Meluzzi in the left-leaning district of Mirafiori. He was elected mayor of Turin in 2001, succeeding to Valentino Castellani and then re-elected in May 2006 with 66.6% of votes, defeating the centre-right candidate Rocco Buttiglione. Links : https://twitter.com/sergiocchiampa	7:22	- Edition : Alessandro Diana

Figure 14: Identification of the hot spots and collection of relevant content

The implementation of this design was carried out in 2 stages: first, an initial UI design was created as shown in Figure 15. This phase focused more on enabling the features and functionalities of the UI on the BRIDGET Player.



Figure 15: Illustrations of the first UI design as presented on the tablet for one of the RAI programs

The second phase of the UI design concentrated on the presentation of such functionalities on the second screen, in order to optimize the user experience. Therefore, three different scenarios were elaborated for three of the RAI programs: news, a documentary on Torino architecture and a TV entertainment show. The UI design was centred not only on the content itself, but also on the targeted audience.

Let us note that the UI for the TV talk show “Porta a Porta” was not redesigned, since the first version of the UI was found to be suitable for the presented content.

2.2.1 News UI

Figure 16 shows the current version of the UI design for the news report. The main difference from the first version of the UI consists in the fact that, in the later version, each bridget is providing enrichment to only one of the subjects discussed in the news report. The user can browse at any time the summary of the show, in order to see the list of the discussed subjects. Users don’t have access to all the bridgets from the beginning, but only when such bridgets become “available”, i.e. when the related subject is presented in the show (following the principles described in Section 2.1).



Figure 16: Illustrations of the second UI design as presented on the tablet for the news report

2.2.2 Documentary UI

Figure 17 presents the current version of the UI for the documentary on Torino architecture. In this case, as the documentary is presenting the landmarks of a city, the entire UI design was conceived around the map of the city. Each time a bridget becomes available, i.e. a new landmark is presented in the documentary, it is positioned on the map. A new feature implemented within this design is the possibility to view the restaurants and hotels situated in the landmark's neighbourhood. At the end, the user is presented with a small quiz related to the presented content.

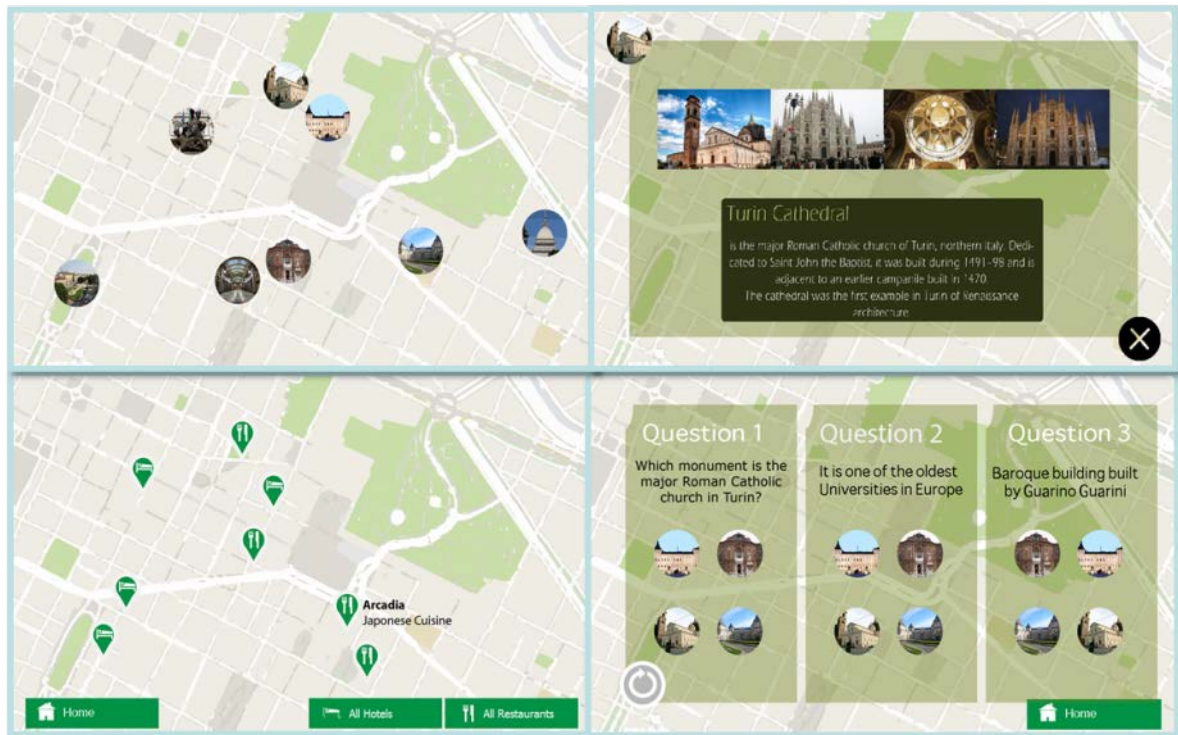


Figure 17: Illustrations of the second UI design as presented on the tablet for a documentary on Torino architecture

2.2.3 Entertainment Show UI

Figure 18 presents the current version of the UI design for the TV entertainment show. Since the target audience of this show is teenage girls, the features and design were focused on the presented subjects, i.e. street style; make up, look and hair styles. At the end of the program, the user also is presented with a dressing game. The game consists of creating different outfits for a character by combining garments from the proposed gallery. The game provides a feedback for each created outfit regarding the user's skills in matching different clothes (e.g. "The shoes don't match", "Try again", "This is amazing!", "It's so fancy!", "Love the colours" etc).



Figure 18: Illustrations of the second UI design as presented on the tablet for a TV entertainment show

2.3 Ongoing Usability Experts Analysis

The BRIDGET user experience is being continuously improved through the involvement of usability experts coming from the media production environment and in collaboration with WP8. They have been involved in the first design phase and they have already seen the first release of the BRIDGET tools and gave a first feedback from the usability point of view; the feedbacks and comments received will be used as drivers to improve user experience in next releases of the BRIDGET tools.

The first prototype of the BRIDGET Player was shown to the usability experts in October 2014 and an improved version has been presented during the project's integration meeting in December 2014. During the same meeting also a story board for a new enriched programme about lyric opera "Madama Butterfly" was presented by RAI's production department. The first feedback received is the following:

- It would be nice to have an alarm every time a new time-aligned bridget gets in scope (a vibration or a sound bip).
- During the bridget experience we noticed that sometimes the attention is really captured by the second screen with the risk that users need to continuously change context from the primary screen to the second screen. As an effect it is not unlikely to miss some important linkage elements between the primary content and the bridget-ed content. To limit this effect a suggestion received is to include in the second screen application also some sort of "representative element" of the source, like for example a (list of) key frames. Basically this could be implemented on the authoring tool as an optional bridget proposed by default every time a new time aligned bridget is created. The author is always free to discard the option. In general it is suggested to define methods to drive the attention of the user during the bridget experience, especially when related content is video and audio, without interfering too much with the main stream.
- Although already sufficient for a good first engaging experience, the set of available media types for the destination content seems limited. It is suggested to include also generic text, PDF files and structured metadata (e.g., titles, descriptions, credits).
- It is suggested to implement a way to classify bridglets into a set of user-defined classes, decided by the author of the bridglets. The aim is to use a visual cue (like for example a logo or a colour setting) for each class during the bridget experience. Examples of such classes are: information, curiosity, related material.

- It is also proposed to consider the implementation of synchronized bridgets, i.e. time aligned bridgets that change state (e.g., paging long text, selecting the portion of information presented) during the main programme timeline. In some specific cases like using an alternative camera shooting as destination content, it is useful to have a tighter synchronization between the primary and secondary stream. A way to implement this would be through autostart of destination content.

A second round of evaluations has been done in early 2015 in preparation of the user trials, taking place in the second year of the project. This second evaluation brought in several others inputs regarding the completeness of the functionalities exposed by the Professional Authoring Environment w.r.t. typical production workflows. This represents a step ahead in the process of making these tools acceptable and widely usable by professionals inside the broadcasters' organisations. The inputs have been evaluated by WP7 and decisions have been made about what additional functionalities had to be included for the user trials. Specifically, the main new functionalities regard the possibility to modify authorial decisions about bridgets, the extension of the descriptive metadata set for bridgets, the possibility to re-use existing bridgets in the context of a programme. At the end of this second cycle, WP8 believes that the Authoring Tool is ready for a first meaningful evaluation by professionals.

3 T7.2 - Authoring Tools

This Task, which started in M5, has two objectives: *i)* to design and implement full-featured studio version AT meant for professional broadcasters (*professional*) and content providers, *ii)* to design and implement a simplified (*mini*) AT usable by consumers on end-users terminals. During the first 18 months of the project, the activity was focused on the *professional AT*. Implementation of the *mini AT* will start in the remaining months of year 2.

Section 3.1 describes the overall architecture of the developed AT; more details about the frontend and the backend systems are provided respectively in 3.2 and 3.3.

3.1 Architecture Design and Implementation

Professional authors goal is to present on the second screen content that has semantic connections with the first screen content: additional information about people, places, events, ... In order to do so, they can use traditional, text based, search when the semantic concept can be formalized as text. However, the answer to this kind of search is in most of the cases presented also as textual information. While including this classic mechanism in the authoring tool, BRIDGET goes beyond this paradigm allowing also visual links performed by using visual search technologies developed in WP5: by using an image as query, the professional author obtains a set of similar images from the repository, images that may be already connected to other metadata (text or other media). He can then chose the ones he want to include in the bridget and select what kind of information will be presented (the media itself, metadata that comes with the media, ...) and how this will be presented (the layout).

All those functionalities are made available through the *professional AT*, implemented as a web tool based on the MyMultimediaWorld [7] platform. This framework, property of IMT, manages multimedia content and the associated descriptions in a cloud-friendly manner. This task was dedicated to the integration of the novel functionalities developed in WP[4-6] in the form of corresponding plug-ins. Each plug-in corresponds to an engine in the MXM functional architecture designed by WP3 and described in D3.1. The components already integrated at the end of the 18 months are: temporal segmenter, visual search, audio fingerprint extraction and matching. The integration activity is continuing for the other BRIDGET engines; however the integration rules are already established and adopted by the entire consortium.

The AT has access to additional multi-media repositories, according to the architecture depicted in Figure Figure 19: the content of such repositories (*Program storage*) is previously analysed, in order to automatically extract descriptors uniquely representing such content, that are stored on the *Bridget storage*. Therefore, when the media to enrich is uploaded, an automatic processing chain may be started,

searching for associations in the repository, collecting the corresponding metadata (*Metadata storage*) and proposing to the content producer several types of enrichments (images, text, videos). This process may be also started for only a specific part of the program.

Finally, the AT packages the assets, UIs, links to external services and resources in a standardised and consistent manner, and those packages are ready for delivery and consumption at the player side. The standard used as a basis for the data format created by the authoring tool is MPEG-A Part 13 (Augmented Reality Application Format) [1], a standard initiated by the BRIDGET partners before the project started and continuously improved by considering the feedback from BRIDGET experiments. The activity related to the bridget layout editor is planned in two phases: in the first phase bridget templates are used, and in the second phase a full editor will be integrated in the AT.

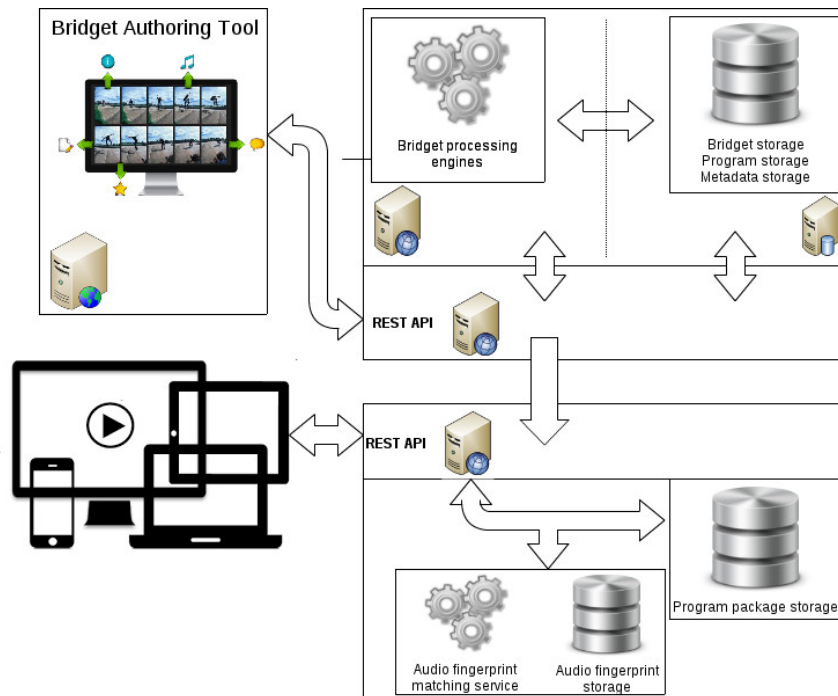


Figure 19: BRIDGET AT frontend and backend architecture.

3.2 Authoring Tool Frontend

The AT frontend is a browser-based web application that allows a professional designer (author) to create and store bridgets. It has been developed as an HTML5 application running entirely in the browser. This allows an easy implementation and efficient logical separation between frontend and backend. It uses REACT.js [4] to exploit efficient client-side templating and redraw on each application state transition. It interacts with the backend using pure HTTP REST APIs.

The development process carried out so far can be summarized in the following breakdown of activities:

- Interaction with RAI experts and professionals to select user requirements and critical aspects of a user interface for the authoring of bridgets, as described in section 2.1 and 2.3;
- Translation of user requirements into functional requirement and drafting the user workflow then shared with the partners for agreement. Such requirements are aligned with the general BRIDGET requirements described in deliverable 2.2;
- Analysis of software solution and libraries to support browser client-side application development and creation of rich user interface;
- Definition of a first set of APIs to support the first minimum bridget workflow;
- Creation of interfaces and logic for source and destination content upload and description (metadata);

- Creation of user interfaces and application logic to view enriched multimedia contents with associated bridgets on the timeline;
- Addition of advanced UI functionalities for:
 - Enabling user selection of time segments in the multimedia player;
 - Conversion of a video frame to an image using HTML5 Canvas apis;
 - Enabling the user selection of an area of the image to select specific objects inside a scene.
- Creation of the user workflow to:
 - Register/Login/Logout/Unregister;
 - Create and remove bridgets;
 - Edit metadata;
 - Run visual searches and select results;

As an example, the Figures below demonstrate how some of the BRIDGET AT functionalities can be accessed.

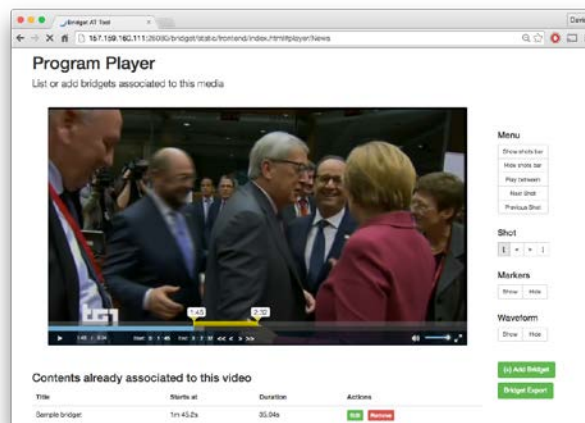


Figure 20: Video is played to find source content for a bridget

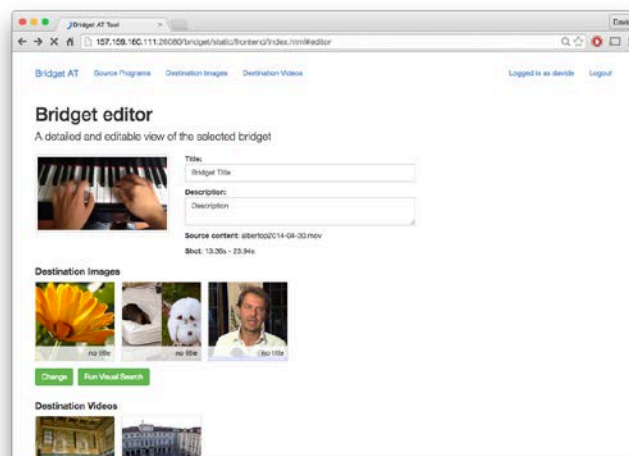


Figure 21: A bridget is created

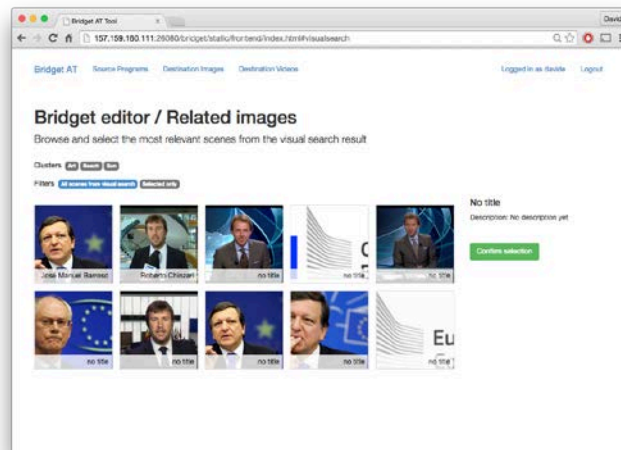


Figure 22: Images suggested by CDVS search

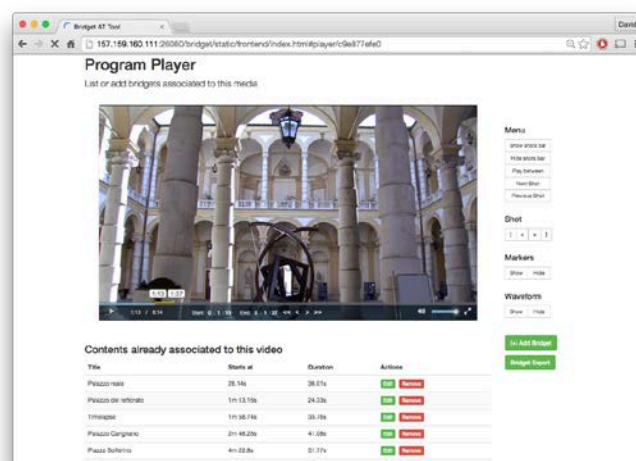


Figure 23: Viewing all bridgets in a program

3.3 Authoring Tool Backend

The AT backend is implementing the architecture defined in WP3 and described in details in D3.1, "BRIDGET System Architecture and Interfaces".

At the time of writing this deliverable, the following components were integrated (as proto-engines, adopting naming conventions of D3.1).

Proto Engine	Overall Description	Current AT integration status	Source
Media Structure Analysis	Analyses a media content producing a hierarchical temporal structure representation of the content, as well as low-level descriptors (e.g. visual content-based), high-level annotations (e.g. scene classification), and quality measures (e.g. visual or audio quality based).	<i>Hierarchical temporal segmentation and dense shot sampling for Visual Search</i>	WP4
Visual Search	Supports search for visual objects in large	<i>Full integra-</i>	WP5

Proto Engine	Overall Description	Current AT integration status	Source
	image and video libraries: analyses images and videos in the content libraries, extracts compact descriptors, builds descriptor databases and indexing schemes, ranks results	<i>tion: Compact Descriptors for Visual Search</i>	
Fingerprint Extractor	Extracts an Audio FingerPrint (AFP) from an uncompressed chunk of audio data	<i>Full integration</i>	WP7
MediaFramework	The MediaFramework is a high level MPEG-M Engine, grouping together several media specific engines such as Video, Image, Audio, and File Format Engine. It also implements common functionalities (independent on the media type) such as resource loading and saving.	<i>Full integration</i>	MPEG-M Part 2

The following engines are planned to be integrated in the second year of the project.

Proto Engine	Overall Description	Source
Media Annotation	Analyses a media content producing low-level descriptors (e.g. visual content-based), high-level annotations (e.g. scene classification)	WP4
Media Quality Assessment	Analyses a media content producing quality measurements (e.g. visual or audio quality based)	WP4
Media Analysis – Face detection	Analyses a media content extracting semantic information about detection of faces present in the video	WP4
Media Analysis – Dialog detection	Analyses a media content extracting semantic information about extrapolation of dialogues in audio tracks	WP4
3D Reconstruction	Creates a 3D model from input images and/or videos	WP6
Audio bridgets	Creates audio bridget with spatial audio that can be rendered through binaural playback	WP6
Bridget Description	Provides access to bridget data structures and the associated metadata	WP7
3D Compression	Produces a compressed 3D graphics data structure from a 3D model that can be efficiently transmitted, and later decompressed and rendered to screen	WP6

The AT uses an internal data model which is mapped to a distributed database. During the authoring process, data is extracted from the database and exported in a format interpreted by the BRIDGET Player. The data model used by the Player is based on MPEG ARAF (the consortium is currently conducting a

standardisation effort in creating a dedicated format for bridget content, called MLAF [16]). The database behind the AT is structured at a conceptual level as depicted in Figure 24.

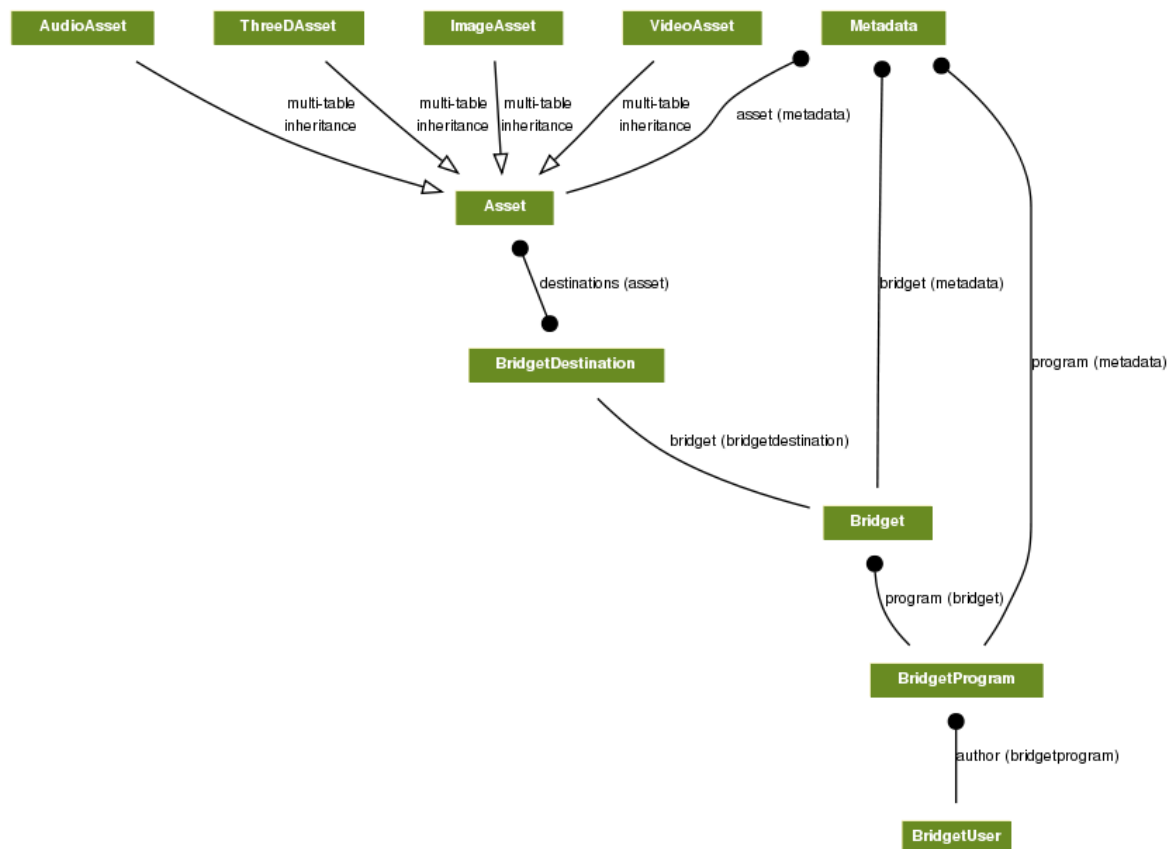


Figure 24: Authoring tools database structure

3.4 A BRIDGET Authoring Tool for Internet Video Distribution

In order to show integration of BRIDGET platform also for internet TV, and to complete a first step towards commercial exploitation of BRIDGET technology, the integration of the current AT with existing WimTV platform was also accomplished in this first reporting period.

CEDEO operates WimTV[2], a commercial platform for content management, trading and distribution on the internet. The WimTV architecture is depicted in Figure 32: AFP Architecture:

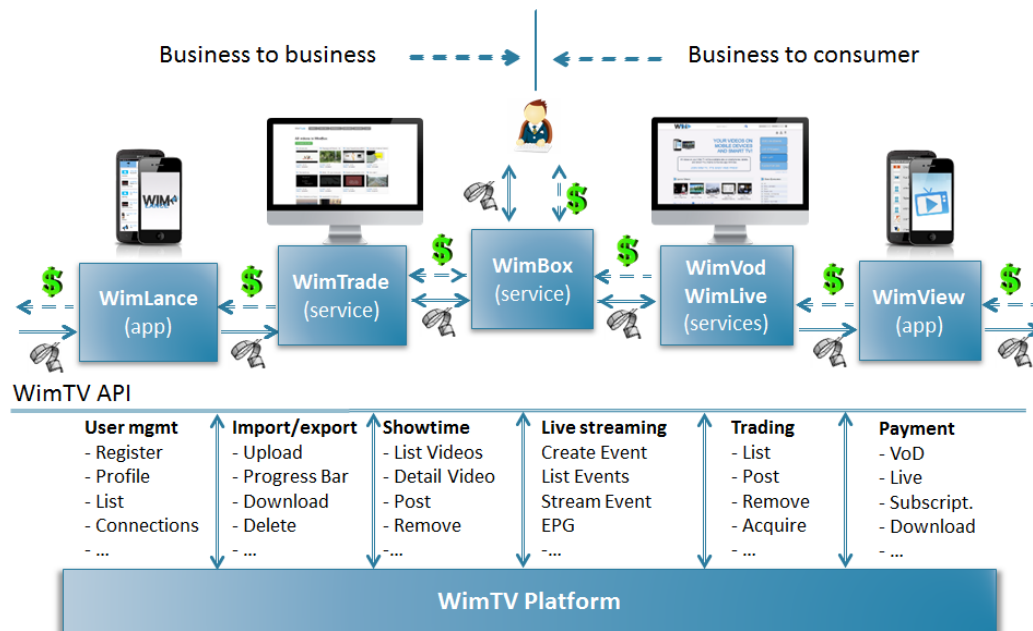


Figure 25: Current WimTV architecture

The WimTV platform exposes a rich set of APIs that support a number of services and applications:

- WimBox: management of user content;
- WimVod: on demand publication of user content;
- WimLive: publication of live events or user content streamed as live at scheduled times;
- WimView: application for viewing content on mobile and Smart TV;
- WimTrade: posting of user content for sale (licensing);
- WimLance: application for responding to requests for content.

The WimTV APIs have been slightly extended to support a new WimTV application: WimBridge, depicted in Figure 26.

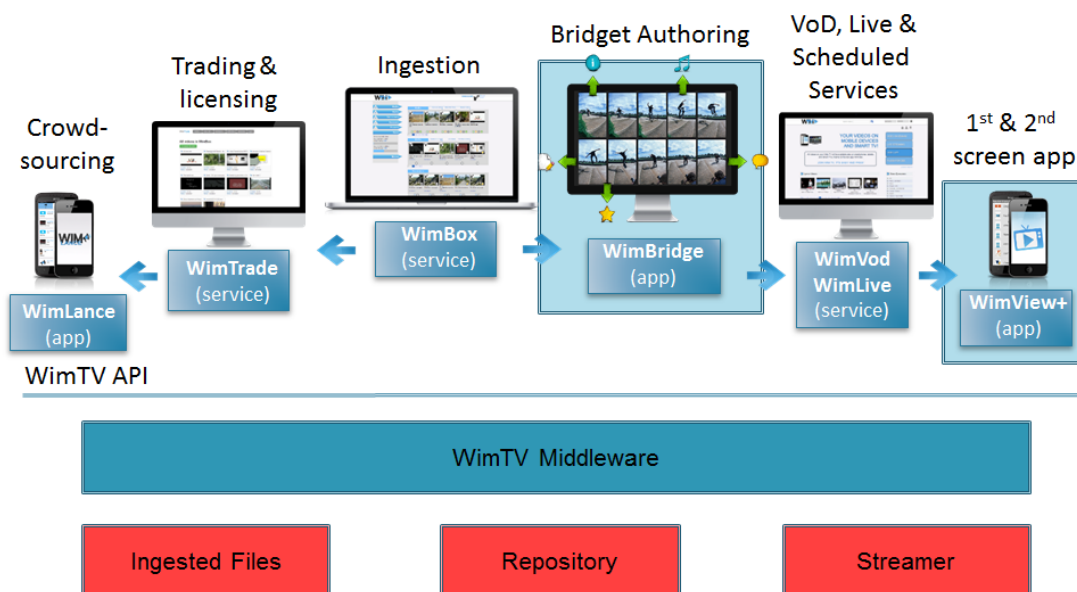


Figure 26: WimTV extended with new bridget-oriented applications

Now a (*professional*) user of the WimTV platform can ingest a video (much like before) to the Ingested Files area. Ingested Files are then transcoded to the internal standard format and stored to the Repository. A user can then use WimBridge to

1. Create Bridget Destination Templates (BDT), i.e. bridgets that contain candidate destination content and corresponding metadata, using the video just ingested;
2. Augment the ingested video with previously created BDTs, duly integrated with references to the selected portions of the video being augmented (source content) and appropriate metadata;
3. Store the bridgets to the repository;
4. Create a scheduled WimTV service made up of bridget-ed videos;
5. Post the scheduled service to the WimLive event guide.

At the scheduled time the streamer will stream the sequence of augmented video as if it were a live event.

It is believed by project partners that this integration step can pave the way for wide adoption of BRIDGET technology in commercial services for several reasons, namely:

- It proves that BRIDGET is applicable to traditional broadcast TV programs as well as internet TV providers;
- It shows how the BRIDGET backend can remain independent from the external platform backend (provided that some contact point and related APIs are agreed);
- It shows simplicity of integration;
- It shows how destination content and related BRIDGET templates can be easily exported.

4 T7.3 - Immersive Media Rendering

This task was active in this reporting period between March 2014 and April 2015.

In this first project year, research has been conducted towards three main directions:

1. Development of an audio fingerprint solution in order to enable synchronization between first and second screen (4.1);
2. Optimizations for binaural playback on mobile devices (4.2);
3. Optimized rendering of synthetic 3D models (4.3).

The optimizations for binaural playback and rendering of synthetic 3D models are in line with the planned DoW activities, and aiming at improving the experience of advanced media fruition on mobile devices. The spatial audio technology generated for enabling binaural playback on the mobile devices is not integrated yet into the overall BRIDGET framework: it is planned, however, to complete such integration within 2015.

The audio fingerprint technology was not originally planned in the DoW. However, it has been developed by the project partners as most reliable and precise way to enable synchronization between first and second screen. Due to the low complexity of the implemented solution, the developed algorithm has been already integrated in both BRIDGET Player and AT and it is currently used as main enabler of the synchronization engine.

4.1 Audio Fingerprint Technology

Temporal synchronization of first and second screen is an essential requirement for BRIDGET.

Audio-based content identification and synchronization is a well established research direction: its adoption in several commercially available technologies represents a hint about the robustness and applicability of such a technology. Also, very low computational complexity enables the usage of audio-based synchronization in real-time on mobile devices, and the low required latency for content recognition makes this technology suitable for BRIDGET scope.

When BRIDGET project started, therefore, such considerations represented a solid background for deciding on adoption of audio fingerprint as main enabler of the BRIDGET synchronization engine.

In the Bridget scenario, the goal is to: *i)* identify the content watched by the listener on the first screen, *ii)* to obtain temporal synchronization with any available BRIDGETs for this content. Audio fingerprints of portions of bridget-ed media are also extracted on the authoring side by professional users, in order to identify triggers for bridgets presentation.

A general overview of the BRIDGET audio fingerprint engine is shown in Figure 27. The end user BRIDGET application running on the second screen device monitors the audio being recorded live from the broadcasted programme using the built-in microphone. First, in Step 1, a short audio fragment corresponding to two to five seconds of the audio signal is recorded. Then, in Step 2, the actual audio fingerprints for the audio fragment are computed. In Step 3, the computed fingerprints are compared to reference fingerprints for the source content previously downloaded, identifying the triggering times for bridget presentation. Information about whether the fingerprints of the current audio fragment match any of the reference fingerprints for the source content is obtained in Step 4. If the computed fingerprints match the fingerprints obtained for a time point in the source content, synchronization of the enriched programme is achieved and the corresponding second screen bridget is presented in a time-synchronous manner together with the programme in Step 5.

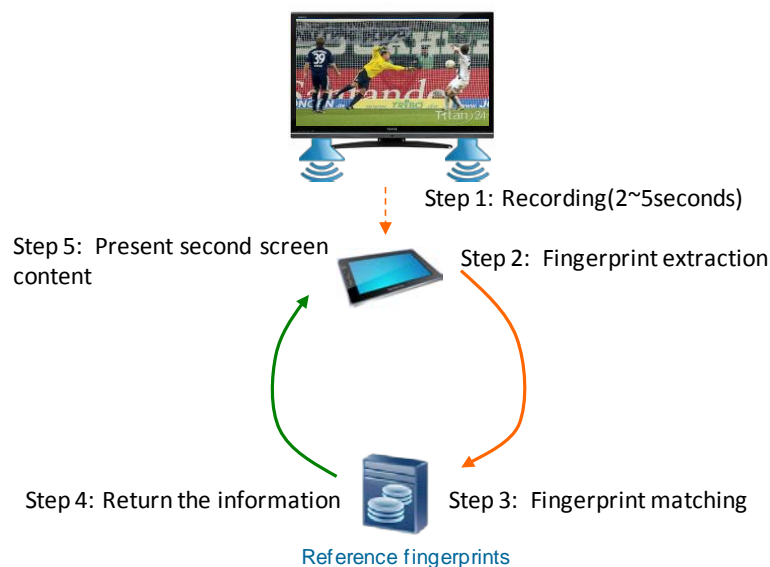


Figure 27: Overview of the BRIDGET Audio Fingerprint Engine

4.1.1 Implemented Approach

The implemented approach relies on a spectrogram computed from the audio stream through a short-time Fourier transform [17]. Then, a peak-picking strategy is applied that extracts locally predominant points in the magnitude spectrogram and reduces the complex spectrogram to a “constellation map”, which is a low-dimensional sparse representation of the original signal by means of a small set of time-frequency points. Although this step constitutes a significant reduction in data, the resulting peaks are highly characteristic, reproducible, and robust against many, even significant, distortions of the signal.

For performing fingerprint look-ups in large databases, index-based techniques such as a Hash table have to be considered in order to reduce the computational complexity and the resulting retrieval time. A hash value is obtained for pairs of peaks as a combination of both frequency values and the time difference between the peaks. Such a combinatorial hashing strategy has three advantages:

- The resulting fingerprints are more specific than single peaks, leading to an acceleration of the retrieval as fewer exact hits are found.
- The fingerprints are translation-invariant as no absolute timing information is captured.

- The combinatorial multiplication of the number of fingerprints introduced by considering pairs of peaks as well as the local nature of the peak pairs increases the robustness to signal degradations.

4.1.2 Performances

For detecting matching fingerprints with high recognition rates in noisy environments, this fingerprinting strategy relies on analyzing fragments of the audio stream of length 2~5 seconds. Once a match is found, a synchronization accuracy of around 40 milliseconds can be achieved. Such results are perfectly matching BRIDGET requirements; besides, all the tests conducted utilizing audio sequences used in the exemplary use cases described in section 2 were successful. Audio synchronization was correctly detected even in presence of (moderately) noisy environment with percentage of success close to 100%.

On the basis of those considerations, no further investigation is planned on this topic.

4.2 Optimizations for Binaural Playback on Mobile Devices

Spatial sound represents an important aspect of immersive multimedia technologies. The ability to recognize spatial position of virtually created sound sources, in combination with visual cues, creates “close to real life” user experience. Different technologies for spatial sound synthesis (acquisition and rendering) have been developed in previous years. They all differ in number of properties e.g. computational complexity and reproduction equipment. Since the requirements for BRIDGET involve spatial audio synthesis and reproduction over tablets and mobile devices, an audio engine based on binaural technology is designed.

Inputs for the spatial sound (hereby referred as *3D audio*) engine are a number of mono audio signals and spatial position and orientation of the corresponding sound sources. According to the position of the source relative to the listener (calculations based on position and orientation data) a pair of Head Related Transfer Functions (HRTF) [14] that corresponds to the calculated direction is selected from a database. HRTFs vary according to individuals: in order to enable general fruition of spatial audio content on mobile devices a generic set of HRTFs is used based on non-individual functions. When the correct pair of HRTFs is selected, it is convolved with the mono audio signal. The result of convolution is two-channel audio signal (binaural signal) that now contains spatial information of the perceived sound source, i.e. the direction of sound is easily noticeable. In addition, an artificial reverberation is added in order to increase the feeling of being in the specific acoustical environment. Reverberation is implemented in terms of room impulse responses. Binaural signal is then reproduced over a set of equalised headphones. Equalisation is done in order to flatten the frequency response of the headphones so that it doesn't influence the frequency content of the binaural signal. Any frequency modulation of binaural signal can influence a perceived direction of a virtual sound source. Block diagram of the audio rendering engine is given at Figure 28.

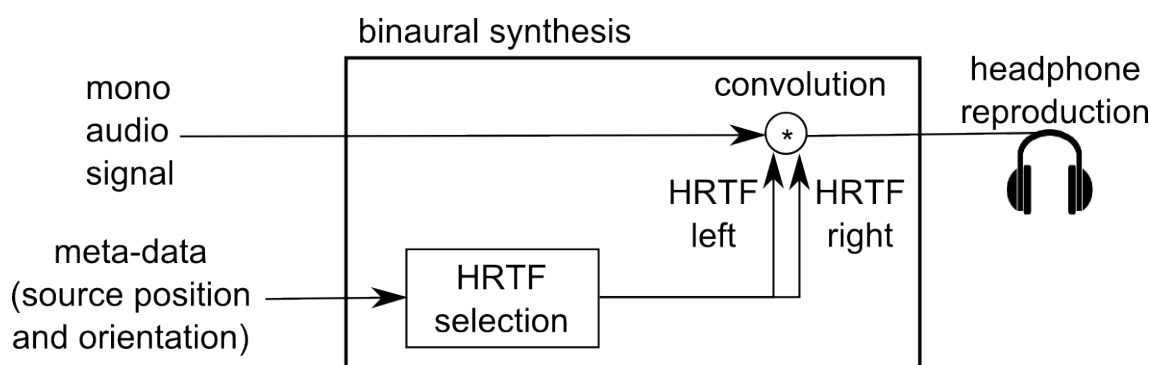


Figure 28: Block diagram of the audio rendering engine based on binaural synthesis

For a spatial audio synthesis of a dynamic environment (listener allowed to move within the scene), binaural synthesis needs to be computed in real-time. Also, due to the possible network delay, binaural synthesis needs to be done at the end of a processing chain right before the reproduction on the mobile devices. More insight about the adopted approach is provided in D6.1.

Within T7.3, in order to reduce complexity and enable low-complexity binaural rendering, the algorithm was optimized utilizing the following principles and techniques:

- The original HRTFs are approximated with a short FIR filters with the length of 256 or 512 taps. In this way, the time and computational power is saved on convolution process, while the spatial features of HRTFs are perceived.
- Convolution is done in frequency domain. First, the Fourier transform of audio signal and corresponding HRTFs is performed and the results are divided and then the Inverse Fourier transform gives the binaural signal in time domain ready for reproduction.
- Room impulse response is split in three groups of reflections: direct sound, early reflections and late reverberations. During the convolution, each group is treated separately which speeds up the process.

Optimized native implementation is being completed in order to enable binaural playback on mobile devices.

4.2.1 Audio engine Demo User Interface

A demonstrative user interface showing the behaviour of the *3D audio engine* was implemented, and it is depicted in Figure 29. User is allowed to freely move within the scene by changing the position of the red dot. Spatial sound is created in real time according to its relative position to the sound sources – gray dots.

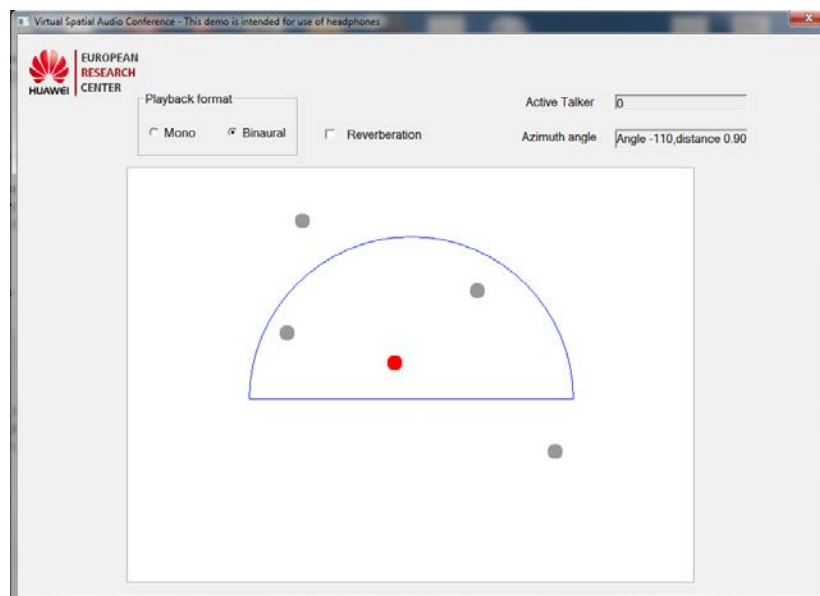


Figure 29: User interface for spatial audio rendering of a dynamic scene

Such interface has been developed as a standalone demo application (released by deliverable D7.2), in order to show the potential of the audio bridget technology at a time in which the audio chain is not integrated yet within the BRIDGET tools and architecture. Conceptually similar interface, however, will be also designed and encapsulated in an ARAF compliant scene, in order to enable the same interaction on the tablet for the consumption of audio bridgets.

4.2.2 Definition of 3D Audio Engine Interfaces

The inclusion of audio bridgets within the overall BRIDGET architecture entails the definition of a novel (at the best of our knowledge, we are the first including spatial audio rendering within a second screen experience) description of audio bridgets information, that needs to be packaged in a format understandable by any BRIDGET player. The interface defined, in order to support such scenario, is the following:

Input (bridget data package)

- Number of virtual sound sources (numerical type - integer);
- Mono audio signal – the audio content of each virtual source within the scene (one channel audio file - 16-bit PCM with 48 kHz sampling frequency);
- Meta data for each sound source – virtual source position in coordinate system of the scene given in Cartesian coordinates (numerical type - single);
- Meta data for the listener – listener position in coordinate system of the scene given in Cartesian coordinates (numerical type - single) and listener orientation in the local spherical coordinate system given by yaw, pitch and roll (numerical type - integer).

In the next release of the BRIDGET tools, it is envisaged to add some more metadata, describing in a compact and explicative way the acoustic properties of the audio rendering environment. Such metadata format is also novel, so an *ad hoc* syntax will be defined for this scope. This will allow users on the tablets to experience how the sound would vary when played in different acoustic environments.

Output (computed in real-time at the player side)

- Binaural audio – mix of all input audio channels positioned in space around the listener (two channel audio file - 16-bit PCM with 48 kHz sampling frequency).

4.3 Optimized Rendering of Synthetic 3D Models

One of the primary aims of T7.3, in what concerns the rendering of synthetic/reconstructed 3D models, is the integration of the splat-based 3D rendering techniques developed within T6.3 for high-end PCs (equipped with powerful CPUs and GPUs) with the BRIDGET Player designed for mobile platforms running the Android operating system. This requires using the embedded systems version of the OpenGL graphic library, namely OpenGL ES 2.0.

First of all, a thorough study of the existing GPAC framework [9], which is the basis for the BRIDGET Player, was conducted. This included the review of the general data flow scheme, as well as an in-depth analysis of the “compositor” sub-project of libgpac, where the actual rendering of 3D objects is performed. Moreover, the Osmo4 application for Android, which enables the use of GPAC for compliant mobile devices, was also studied and modified to fulfil BRIDGET’s immersive media rendering needs. The results of this analysis and the derived actions are detailed below:

- Regarding the data flow, the splat-based 3D models generated within T6.1 must be provided in a readable format for the existing GPAC infrastructure. Since T6.2 on “3D Media Coding” only started at the very end of the first year of the project, the provisional data format chosen to store 3D models, and in particular 3D point clouds, was sub-optimally derived from Binary Format for Scenes (BIFS), also known as MPEG-4 Part 11 (more formally, ISO/IEC 14496-11). This specification already provided means to store and load *traditional* point sets, where *traditional* is meant to include only locations and colours. For the sake of BRIDGET’s needs, an extension of BIFS has been designed which includes new attributes to support splats, namely major and minor ellipse axes. Once this information is retrieved from the “extended BIFS files” containing splat-based 3D point clouds, it is stored in memory in order to be later accessed by the rendering modules.
- Concerning the rendering core of GPAC, i.e., the “compositor” sub-project, one of the leading conclusions was the need for upgrading to version 2.0 the currently used version (1.0) of the OpenGL ES

graphic library, since v1.0 only allows for a fixed graphic pipeline rendering with very limited customization. However, although OpenGL ES 2.0 supports programmable vertex and fragment shaders, it is not backwards compatible with v1.0, and therefore a shader implementation is mandatory for every desired rendering effect. This requires developing specific shaders for tasks such as point- and mesh-based rendering, model lighting and texture mapping. During the first year of the project, implementations for rendering *traditional* and splat-based 3D point models have been completed, and the remaining tasks are expected to be developed during the second year to achieve a full upgrade of GPAG's "compositor" sub-project to OpenGL ES 2.0. Besides, all actions detailed in T6.3 (see D6.1) have been included in the appropriate GPAC functions, together with the developed vertex and fragment shaders. The only exception is the alpha-blending technique for smoothing the splats' edges and colours. This alpha-blending technique currently needs that the primitives be sorted according to their distance to the viewpoint at every rendering instant, which is a highly demanding task not yet suitable for embedded systems. However, as stated in T6.3, research is being carried to circumvent this issue.

Regarding the user experience, some modifications of the original Osmo4 GUI have been performed to improve the immersive effect of the BRIDGET Player. Osmo4's interface initially allowed only for a rotation of the rendered 3D model, but the classic possibilities of performing translations and in/out zooms are now implemented as well. Zooms are achieved by the typical "two-finger pinch gesture" (the field of view of the projection matrix is consequently modified to increase or decrease the region of the 3D model displayed on the viewport). As for the translation movement, it is launched with a "one-finger long press gesture", which then allows for a horizontal or vertical shift of the 3D model. In order to differentiate this gesture from the rotation action, once the "translation mode" is turned on, the background is automatically changed from black to white, and conversely.

The top row of Figure 30 shows the difference between the original point cloud of Torino's Palazzo Carignano and the corresponding splat-based 3D model generated within WP6, both of them rendered in the BRIDGET Player for Android. The bottom row presents the full splat model of Torino's Arco Valentino and a close-up obtained with the zoom-in action described above. Examples of other splat-based models can be found in Figure 31.



Figure 30: Snapshots of different the BRDIGET Player for Android: a traditional point cloud and its derived splat model (top), plus a full view and a close-up of a splat-based 3D model (bottom)

Finally, regarding the HW platforms used for testing, all algorithms for rendering of splat-based 3D models in mobile environments using the modified GPAC-based BRIDGET Player run on the following two tablets with equally acceptable performance results:

- Huawei MediaPad M1 8.0 (running Android 4.2.2);
- Nvidia Tegra Note 7 (running Android 4.4.2).

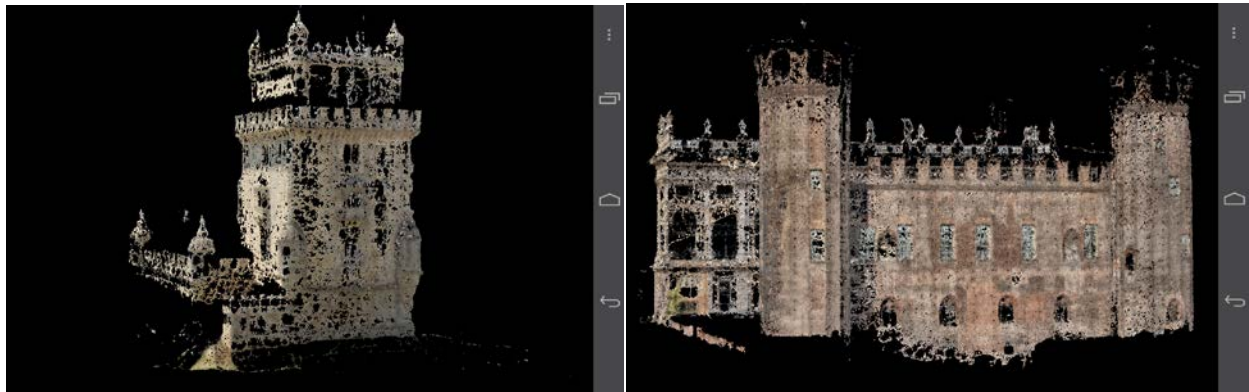


Figure 31: Full views of different splat-based 3D models rendered in the BRIDGET Player for Android

5 T7.4 - Multi-screen player

This task was active in this reporting period between March 2014 and April 2015.

The overall picture about BRIDGET player development has been reported in 5.1. As reported in Section 3.3, the first step towards exploitation of BRIDGET technologies in existing services was also accomplished in the first 18 months, integrating the BRIDGET overall framework with the existing WimTV webTV platform. On the end user's mobile device, therefore, the BRIDGET player has been integrated with the WimTV player into a dedicated application able to seamlessly synchronize and visualize bridgets on top of the existing WimTV service. Details about such work are provided in subsection 5.2.

5.1 Architecture Design and Implementation

This task addresses the architecture design of a BRIDGET Player prototype, as well as its implementation in the reference mobile terminals. The starting point was GPAC, a multimedia interactive framework, property of IMT (but distributed as open source) [9] and the activity was focused on integrating MPEG ARAF functionalities in OSMO4, the GPAC multimedia player. OSMO4 is an open source MPEG-4 compliant multimedia player which is supported on multiple platforms: Windows, Windows mobile, Android, iOS, Linux and MacOSX. OSMO4 supports different multimedia formats as well as a JavaScript execution engine based on the Mozilla SpiderMonkey engine. The implementation supports most of the Script features defined in the MPEG-4 standard and includes full support for interaction with any object in a scene. Additionally, Osmo4 supports PNG and JPEG for static images, MP3 and AAC for audio, H.264/AVC and HEVC for video. Files can be accessed from the local drive or through HTTP. OSMO4 uses OpenGL and OpenGL-ES for rendering mixed environments including 2D and 3D graphic objects. The player has also access to a variety of sensors such as motion sensors, location sensors, microphones, cameras and others. Apart from implementing support of ARAF in OSMO, the main modification needed in this first period, in order to enable full support of BRIDGET functionalities in GPAC, was the introduction of a mechanism for support and management of audio fingerprint (AFP) within the player architecture.

The schema presented in Figure 32 illustrates the AFP functional architecture.

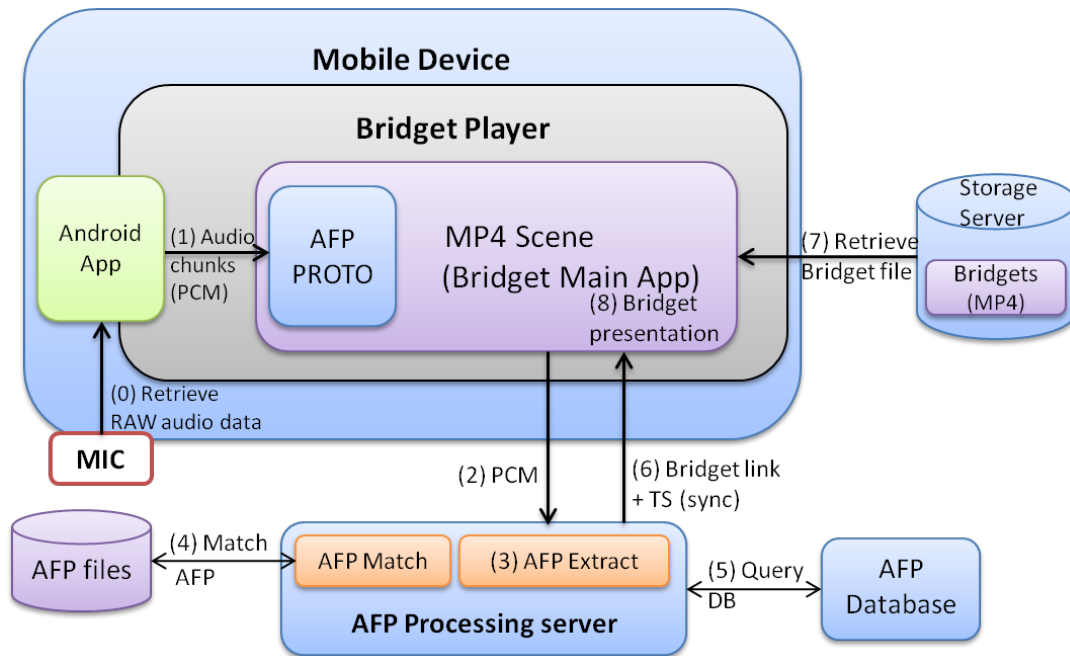


Figure 32: AFP Architecture

In order for the BRIDGET Player to retrieve bridgets and synchronize with the main screen, the following functionalities, described below, have been implemented:

- An AFP PROTO, which is a scene level feature implemented in the Bridget Player.
- A processing server where the audio fingerprint functionalities are running.

The AFP prototype has been designed to connect to the microphone of the device, through the Android application that acts as a wrapper over the native code of the BRIDGET Player. As long as the prototype is enabled, the raw audio data of the microphone is intercepted, chunked to a given size and sent to the BRIDGET scene (the main application).

A simplified schema of the AFP prototype is depicted in Figure 33.

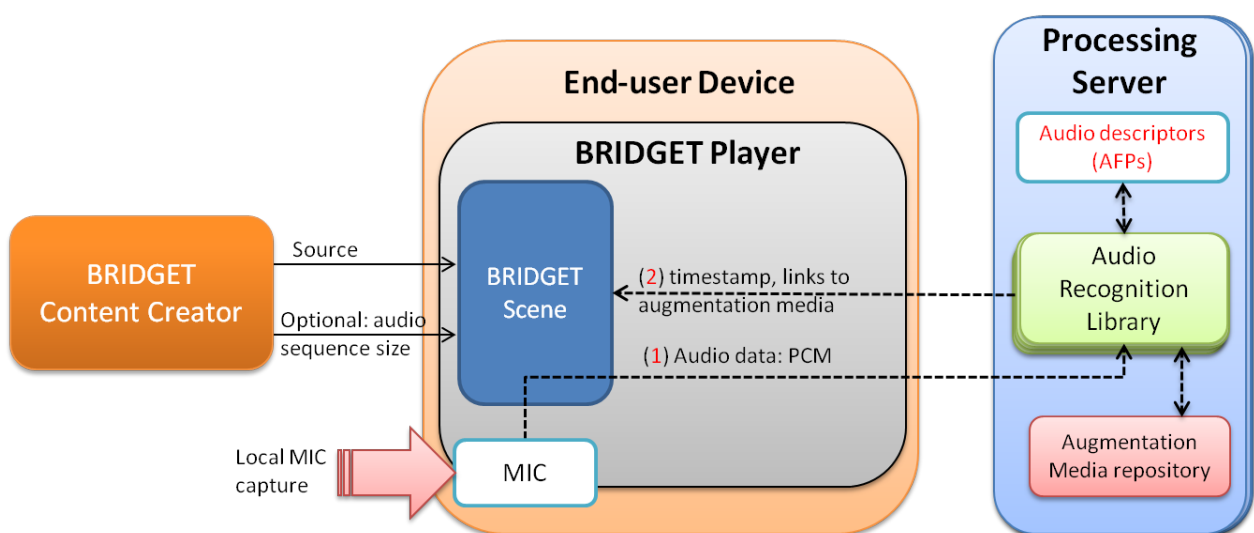


Figure 33: BRIDGET AFP PROTO

Further, the AFP PROTO interface is presented:

```

EXTERNPROTO AFP [
    exposedField MFString      source      []
    exposedField SFInt32       buffer_mic   40000
    exposedField SFBool        isEnabled    FALSE
    eventOut      MFInt32       mic_data
] "urn:inet:gpac:builtin:AFP"

```

The **source** field of the prototype specifies the URL of the device's microphone. The application connects to the microphone and intercepts raw audio data (PCM) as long as **isEnabled** field is TRUE. The PCM chunk size can be also controlled through the prototype instance by using the **buffer_mic** field. The recorder sample rate is hardcoded at 8000 samples per second therefore a 5 second audio chunk corresponds to a **buffer_mic** value of 40.000. Whenever a new audio buffer is filled with data, the PCM array is sent to the Bridget Scene through the **mic_data** field of the prototype. A JavaScript XMLHttpRequest is used to send the audio PCM to the AFP Processing Server where the audio fingerprint services are running. The first operation performed by the processing server is to extract the audio fingerprint by using the Huawei AFP library (section 4.1) and then is to match the result against all the extracted audio fingerprints that are already stored on the storage machine. The Huawei AFP matching library computes the number of common features between the reference fingerprint and the stored ones. If matchings with fingerprints from different programmes are found, a ranking algorithm is applied on the result in order to decide which of the programmes exposes higher audio similarity.

Following the same approach, if several timestamps are returned by the matching library, the algorithm filters the results and returns a single timestamp. Once the program name and the timestamp are identified, the processing server interrogates the database and gets the link pointing to the corresponding BRIDGET file (MP4). The link and the timestamp are sent back to the BRUDGET Scene as a result of the HTTP request previously initiated by the scene. Once the result is available, the BRIDGET Player downloads the BRIDGET file and starts presenting the content considering the timestamp received from the processing server. As long as the AFP proto is enabled, the same process runs continuously. Unless the server returns a link to a new BRIDGET file, the scene only considers the timestamp for synchronisation purposes. If a new link is returned (meaning that another program is detected), the same principle applies as described above.

The architecture and the prototype that have been designed to implement BRIDGET audio fingerprinting functionalities are compliant with one of the prototypes that have been introduced in the second edition of ARAF. The name of the PROTO is **RemAud** and it provides remote audio recognition support in an ARAF Player. The architecture of the RemAud prototype is presented in Figure 34.

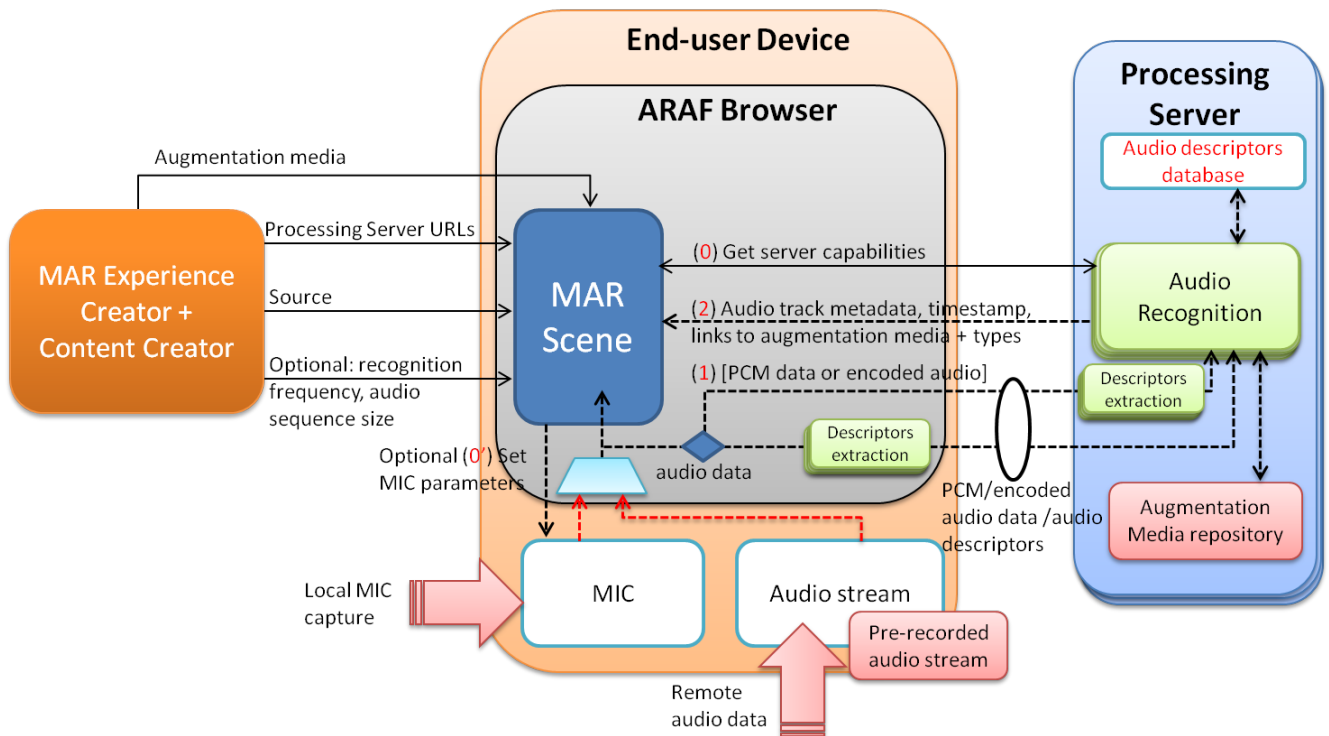


Figure 34: RemAud PROTO (ARAF)

The full description of the prototype can be found in [18], section 4.2.4.12.3.2.

The source, the audio sequence size, the way how the audio data is sent to the processing server and the server response are also defined here and they have exactly the same meaning as described in the AFP prototype functionality.

RemAud provides more functionality than what is necessary in the BRIDGET scenario so far (AFP related); nevertheless RemAud can easily replace the simple prototype that has specifically developed for the BRIDGET use cases.

5.2 A BRIDGET Player for Internet Video Distribution

Following the backend described in section 3.4, a model for internet distribution of video programs enriched with bridgets is given by:

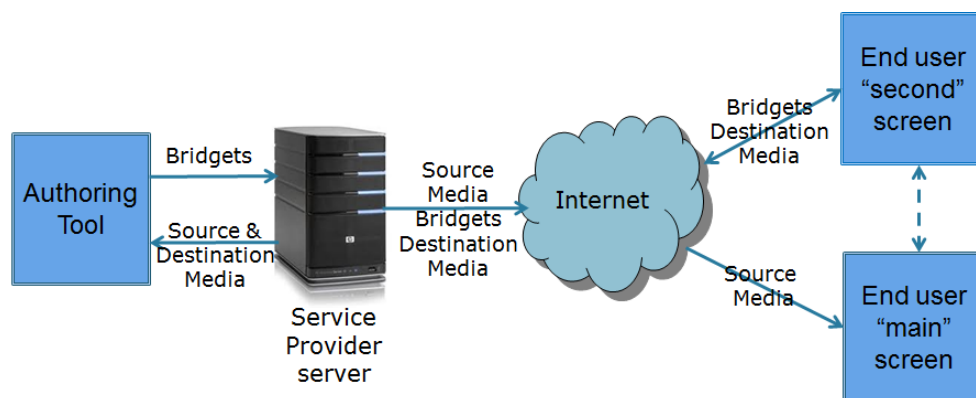


Figure 35: A model for internet distribution of video programs enriched with bridgets

Such an architecture, and in general the concept of linking BRIDGET platform to internet TV, has also implications in the way in which the player needs to behave. In fact, internet offers more opportunities to design different user experiences. The following analysis will be instantiated on the WimTV platform. The most straightforward experience is depicted in Figure 36:

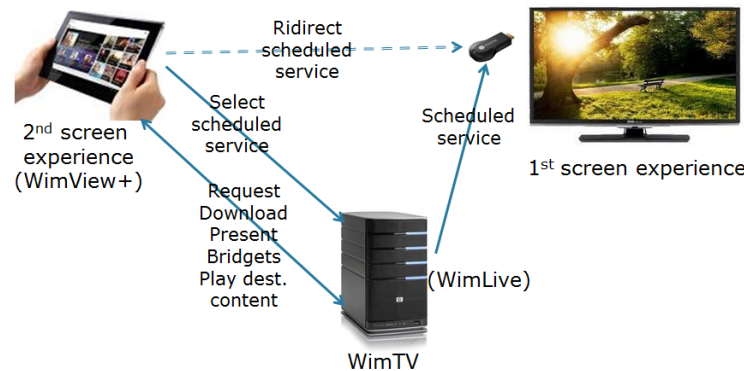


Figure 36: A user experience of internet distribution of video programs enriched with bridgets

In this case, a user starts from the second screen and requests the scheduled video service. The mobile device can redirect the schedule service to the "first" screen (a Smart TV). Bridgets are then consumed on the second screen as in a standard broadcast scenario.

A second, more innovative scenario is provided by Figure 37. This shows the (single) screen of the WimView+ mobile app which is actually composed of 3 screens (windows):

- Top: plays the scheduled service, the equivalent of the broadcast programme;
- Middle: as soon as the app becomes aware that a new bridget becomes active it displays an icon of the bridget here;
- Bottom: the destination content (video) is shown here.

A user starts watching the scheduled service. When the icon of a bridget shows up the user may decide to tap it. Then the destination content (video) starts playing in the bottom window and the audio of the top window is muted. When the user taps the top window the audio of the top window resumes and the audio of the bottom window is muted, but the video (which is actually an on demand video) keeps on playing. If the user taps the bottom window the corresponding audio is resumed and the audio at the top is muted again.

An advantage of this implementation is that the concept doesn't need AFP for synchronization, thus properly working even in extremely noisy environments. Synchronization is controlled by the WimTV server.

Remarkably, also this player integration was accomplished with minimal impact and software implementation effort.

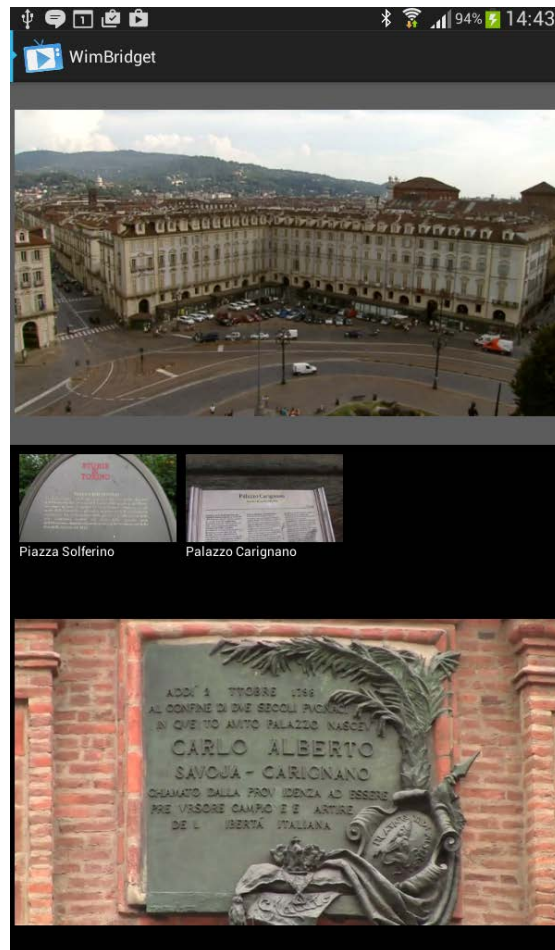


Figure 37. The screen of the WimView+ mobile app

6 T7.5 – Standardization

This task was active in this reporting period between March and October 2014. The effort of the first year has been mainly devoted to ensure compliancy of the AT and player formats to the ARAF standard, and implement single components of player and ATs as MXM engines, as specified in WP3.

As a result of the activities carried out within WP7 (in some cases, jointly with WP6), six proposals co-authored by BRIDGET researchers were submitted to MPEG [10][11][12][13][14][15].

We extended also the ARAF standard to include a mechanism allowing audio fingerprinting, as described in section 5.1. At the current moment, however, the format of audio fingerprint itself is not object of the standard. Next envisaged steps encompass the continuation of such standardisation work with the objective of defining a normative data format for audio fingerprinting.

7 Conclusions

This document summarises the research outcome of WP7 during the first half of BRIDGET's life. WP7 progress is in line with expected plans, without any critical point emerged so far. The planned steps for the upcoming months encompass: *i)* the update of the BRIDGET Tools in order to support of user trials conducted by WP8, *ii)* the development of the first version of the *miniAT iii)* the integration of new modules produced by WP4-6 into the AT backend.

References

- [1] ISO/IEC SC29WG11 23000-13 (MPEG-A) N13182 “DIS of ISO/IEC 23000-13, Augmented Reality Application Format”, Shanghai, October 2012.
- [2] <http://www.wimlabs.com/en/wimtv.html>
- [3] Holmes, M. E., Josephson, S., and Carney, R. E. Visual attention to television programs with a second-screen application. In Proceedings of the Symposium on Eye Tracking Research and Applications, ACM (2012), 397–400
- [4] Hinckley, K., Dixon, M., Sarin, R., Guimbretiere, F., and Balakrishnan, R. Codex: A dual screen tablet computer. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09, ACM (New York, NY, USA, 2009), 1933–1942.
- [5] Bardram, J., Gueddana, S., Houben, S., and Nielsen, S. Reticularspaces: Activity-based computing support for physically distributed and collaborative smart spaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, ACM (New York, NY, USA, 2012), 2845–2854.
- [6] Chen, N., Guimbretiere, F., and Sellen, A. Graduate student use of a multi-slate reading system. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, ACM (New York, NY, USA, 2013), 1799–1808.
- [7] <http://www.mymultimediaworld.com/>
- [8] <http://facebook.github.io/react/>
- [9] <http://gpac.wp.mines-telecom.fr/>
- [10] T. Lavric, M. Preda (IMT), “Updates on M26114 (Initial proto design for server-side processing for augmented reality)”, MPEG contrib. M32336, 107th MPEG mtg., San Jose, CA, US, January 2014.
- [11] T. Lavric, M. Preda (IMT), “Updates on Augmentation Region”, MPEG contrib. M32337, 107th MPEG mtg., San Jose, CA, US, January 2014.
- [12] T. Lavric, M. Preda (IMT), “ARAF guidelines: PROTOs implementations”, MPEG contrib. M33382, 108th MPEG mtg., Valencia, ES, March 2014.
- [13] M. Preda (IMT), G. Kim, C. Perey, “Contributions to MAR RM”, MPEG contrib. M34338, 109th MPEG mtg., Sapporo, JP, July 2014.
- [14] A. Gabrielli, Y. Lehiani, T. Lavric, M. Preda (IMT), “ARAF: remote recognition – analysis and preliminary results”, MPEG contrib. M34360, 109th MPEG mtg., Sapporo, JP, July 2014.
- [15] T. Lavric, M. Preda (IMT), “ARAF: remote audio recognition”, MPEG contrib. M35222, 110th MPEG mtg., Strasbourg, FR, October 2014.
- [16] L. Chiariglione, D. Bertola (CEDEO), A. Messina (RAI), M. Preda, T. Lavric (IMT), “Proposal for a Media Linking Application Format (MLAF)”, MPEG contrib. M35117, 110th MPEG mtg., Strasbourg, FR, October 2014.
- [17] J. B. Allen (June 1977). "Short Time Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform". IEEE Transactions on Acoustics, Speech, and Signal Processing. ASSP-25 (3): 235–238.
- [18] Blauert, J. (1997) “Spatial hearing: the psychophysics of human sound localization”. MIT Press.

- [19] ISO/IEC SC29WG11 23000-13 (MPEG-A) N15293 “Study Text of ISO/IEC CD 23000-13, Augmented Reality Application Format”, Geneva, February 2015.