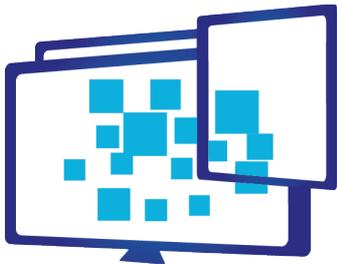


Instrument: Collaborative Project (STREP)

Call Identifier: FP7-ICT2013-10

Grant Agreement: 610404



MediaScape

Dynamic Media Service Creation,
Adaptation and Publishing on Every Device

WP6 – TEST- Prototyping and Testing

Deliverable Number	D6.4
Title	Final functional and technical testing report
Version	1.1
Date	11th May
Status	FINAL

Restricted

Project Partners: VIC, IRT, NEC, BBC, W3C, NOR, BR

Contributors: VIC, IRT, BBC

Every effort has been made to ensure that all statements and information contained herein are accurate; however the Partners accept no liability for any error or omission in the same.

© Copyright in this document remains vested in the Project Partners

Project

Dynamic Media Service Creation, Adaptation and Publishing on Every Device
MediaScape (610404)

Document Title

D6.4 Final functional and technical
testing report

Version

1.1

Date

11th May

Author

VIC, IRT, BBC

Document Control

Version	Date	Author	Modifications
0.1	25 June 2015	Angel Martin (VIC)	Initial draft Definition of the first ToC
0.2	13 November 2015	Kerstin Heinz (BR)	Methodology vision Scrollyteller strategy
0.3	16 November 2015	Angel Martin (VIC)	Contributions consolidation
0.4	18 November 2015	Libby Miller (BBC)	Methodology Vision General review
0.5	20 January 2016	Mikel Zorrilla (VIC)	Strategy for the Elections prototype
0.6	21 January 2016	Mikel Zorrilla (VIC)	Technical testing definition and description
0.7	21 March 2016	Ana Dominguez (VIC)	Technical testing report for Elections prototype
0.8	21 March 2016	Ion Alberdi (VIC)	Review and contributons
0.9	22 March 2016	Benedikt Vogel (IRT) and Alex Erk (IRT)	Contributions for the Scrollyteller prototype
1.0	22 March 2016	Mikel Zorrilla (VIC)	Final contributions
1.1	11 May 2016	Mikel Zorrilla (VIC)	Conclusions and general review



Project

Dynamic Media Service Creation, Adaptation and Publishing on Every Device
MediaScape (610404)

Document Title

D6.4 Final functional and technical
testing report

Version

1.1

Date

11th May

Author

VIC, IRT, BBC

Table of Contents

1.INTRODUCTION.....	3
1.1.RELATION WITH OTHER DELIVERABLES.....	3
2.TESTING METHODOLOGY.....	4
2.1.FUNCTIONAL TESTING METHODOLOGY.....	4
2.2.TECHNICAL TESTING METHODOLOGY.....	5
3.MEDIASCAPE PROTOTYPES.....	6
3.1.ELECTIONS.....	6
3.1.1. <i>Definition</i>	6
3.1.2. <i>Functional and Operational Description</i>	6
3.1.3. <i>Testing metrics</i>	10
3.1.4. <i>Testing results</i>	13
3.1.5. <i>Conclusions</i>	25
3.2.SCROLLYTELLER.....	26
3.2.1. <i>Definition</i>	26
3.2.2. <i>Functional and Operational Description</i>	26
3.2.3. <i>Testing metrics</i>	27
3.2.4. <i>Testing results</i>	29
3.2.5. <i>Conclusions</i>	31
4.CONCLUSIONS.....	32

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

1. Introduction

Situating the technologies and libraries developed in WP3, WP4 and WP5 in the context of real user needs and using the four scenarios defined in WP2 as a basis, three final prototypes have been defined and implemented in MediaScape:

- Scrollyteller prototype: IRT and BR are the main contributors to this prototype which is based on the HybridRadio scenario. A broadcaster delivering a radio programme with second screen visual information synchronised with the radio programme. It can also be extended to a third device, such as TV, having two visual screens synchronised with the radio show. BR is leading this prototype with the technical support of IRT.
- EuroMeme prototype: Developed by BBC, it is based on the ExtraMedia scenario with some social aspects covered by the MultiQuiz and the SocialTV scenarios. It provides a synchronised second screen service for the Eurovision TV show to create clip videos from the second screen device and share them to friends. BBC is leading this prototype.
- Elections prototype: Vicomtech-IK4 is main contributor to this prototype which is based on the ExtraMedia scenario. A broadcaster delivering a live TV show during the elections night and providing related content such as other streaming cameras from the headquarters of the political parties, vote counting real-time information, statistics, graphics, social networks services, etc. The end users will consume it with multiple devices at the same time. Vicomtech-IK4 is leading this prototype in a close collaboration with EITB, the public Basque broadcaster, part of the EIKEN cluster (member of the Advisory Board).

Once the prototypes have been developed, it is necessary to go through a testing process to on the one hand, identify and correct the weak points of the prototypes and on the other hand, and with a broader view, to analyse the performance of MediaScape technologies as an integrated solution. This process requires the adoption of some metrics to tackle the functional, technical or end-user testing validation.

This deliverable describes functional and technical testing activities, together with the results and conclusions obtained from them. From the three developed prototypes, since Euromeme is clearly oriented to user experience across the application, functional and technical tests have not been performed, applying only user tests that will be documented in D6.5 together with user tests of Scrollyteller and Elections.

Therefore, this document provides the functional and technical testing activities across ScrollyTeller and Elections prototypes. Section 2 describes the methodologies used for each one of the tests while Section 3 provides a functional and operational description, the parameters and metrics to analyse, the obtained results and conclusions for each prototype.

1.1. Relation with other deliverables

All the employed technologies and libraries in the prototypes are the result of the work done along the project in technical work packages, that is, WP3, WP4 and WP5. Therefore, all the API details are documented in “D3.3 - Final Version of multi-connection mechanisms and multi-device authentication”, “D4.3 - Final implementation of multi-device synchronisation services” and “D5.3 - Final Version of multi-device adaptation” respectively. Deliverable “D6.3 - Final Prototypes” provides a full description of each prototype regarding the definition process and the main functionalities and architecture.

This document presents the testing activities and results obtained on the functional and technical tests of ScrollyTeller and Elections prototypes. And finally Deliverable “D6.5 - Final user testing report” will provide the user testing activities and results of the three final MediaScape prototypes.

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

2. Testing methodology

2.1. Functional testing methodology

The main goal of the functional testing activity is to evaluate the functionality of a prototype. It means to verify that the prototype provides the functionality that has been designed for.

For that purpose, this deliverable provides a functional and operational description for the Scrollyteller and the Elections prototypes developed in MediaScape. The functional description provides the features of the possibilities that the prototype offers to the end user: how to launch the application, add new devices, interact with the application, etc. The operational description describes how the prototype works to provide that functionality.

The methodology defined for the functional testing activities is a black-box testing method, which consists on defining tasks to be performed by the Scrollyteller and Elections prototypes without peering into their internal structures or workings. These tasks will reflect the main functionality that the prototypes offer. For example, if a prototype enables different cameras apart from the main TV content, one task could be to add to the TV another camera.

Once each prototype has a list of tasks to be performed by the prototype, described in this document, the functional testing consists on performing those tasks and learn if some functional and operational aspects should be improved in order to provide an application whose all functionalities work properly. These mentioned tasks will be grouped in the following domains:

- General testing: tasks defined in this domain will check that basic features like starting or finishing the application or opening different sessions, work correctly in a user level on each target device.
- User interface testing: tasks defined in this domain will check the correct design and work of the whole user interface, that is the menu and configuration panel, the components, the controller and the layouts.
- Association testing: tasks defined in this domain will test how association of multiple devices features work.
- Local control testing: tasks defined in this domain will test the application when local interactions are performed. In MediaScape applications, all the devices run the same application and each device can interact with itself as well as with each other. So, all the tasks in the local testing will be related to interactions to the same device.
- Remote control testing: tasks defined in this domain will test all the features that could be performed in the application when it is running on a group of devices at the same time. Unlike in local testing, interactions to remote devices will be mainly analysed.
- Timeline testing: task defined in this domain will test timeline related features as playing, pausing or time-shifting the application.

Therefore, the conclusions that this methodology aims to achieve is to verify that all the functionalities that each prototype wants to cover are well implemented. It will be possible to perform small improvements if some small functionality gaps are detected during the testing process. All the functionality should be perfectly implemented across the prototypes at the last stage of the testing activities.

Project	Document Title	
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)	D6.4 Final functional and technical testing report	
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

2.2. Technical testing methodology

The main objective of the technical testing activities is to measure the technical implementation of the prototypes, testing the technical implementation of the developed libraries being used across the prototypes. This document describes the architectural definition of the Scrollyteller and Elections prototypes, detailing the different MediaScape libraries and APIs being used on each prototype. Due to the differences on the involved libraries on each prototype, different metrics are defined for each case. However, the same pillars are followed across them:

- Operational domain:
 - Responsiveness: measure how responsive is the user interface of the prototype across different devices of different screen sizes.
 - Interoperability: measure the interoperability of the prototype across different devices (TVs, mobiles, laptops), their OS (Android, iOS, etc.) and the different browsers (Google Chrome, Firefox, Opera, etc.)
- Performance domain:
 - Scalability and concurrence: measure the possibility of having different users using the application and how many devices at the same session within a user.
 - Latency and jitter: measure the latency and jitter of the different parameters of the prototype. Could be the time to move a component from one device to another, the time to show something on a device or the synchronisation accuracy across devices.
- Technological domain:
 - Universal Vs. Experimental: Measure the technologies requiring an experimental or specific environment, such as a specific implementation of a specific Web browser, OS dependencies, etc.
 - Effectiveness: Measure the network communication and the time to load overhead added by the MediaScape libraries
- Prototyping:
 - Flexibility and modularity: measure how flexible are the libraries from a developer point of view to create prototypes or to make modifications to the functionalities of the existing prototype
 - Expressiveness: measure how easy and adequate is the nomenclature used in the APIs. For example, how intuitive is to define the multi-device behaviour of the application through the Adaptation libraries.
 - Openness: measure how open the libraries are to use them in their own, to combine with other proprietary or existing libraries (not developed in MediaScape), to extend the libraries, add new rules, etc.

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

3. MediaScape Prototypes

3.1. Elections

3.1.1. Definition

The Elections prototype, created by Vicomtech-IK4 in collaboration with the Basque public broadcaster (EiTB¹), provides a multi-device experience for a Live TV Programme broadcasting the elections night, and it can be consumed on demand, to perceive the same experience as live the day after. The prototype starts with a Connected TV (such as an HbbTV television) with an application signalled by the broadcaster. The user is able to interact with the app through the TV or associate new devices to the experience (laptops, tablets, smartphones, etc.). The user is able to consume a single experience though multiple devices at the same time in a dynamic set-up that can be updated during the experience.

The Elections application targets real-time stream of extra media incorporating the multi-screen experience into viewing entertainment. A complete second screen experience with devices around a TV has been designed and it serves audiences an increased interaction and immersion with an Elections program.

The possibilities available for the users include:

- different cameras from the headquarters
- radio signal
- real-time counting information
- Infographics and historical results
- social activity

The elections prototype has been conceived by means of technologies developed in MediaScape making use of the WP3 discovery and association, WP4 sync and WP5 adaptation. More accurately, the party creation joining different devices to the elections app is supported by the Discovery API and the Association API from WP3. Once the communication is established a data channel is deployed through the Mapping Service and the Shared State from WP4. These are also employed to store and propagate the experience customization parameters. Concerning the dynamic components, the timing from WP4 provides multiple live media synchronisation. Last but not least, the dynamic experience is built on top of the Agent and the Application Context from WP4 while the responsive behaviour comes through the Adaptation Engine and the User Interface Engine from WP5.

3.1.2. Functional and Operational Description

This section lists the Elections prototype's main features and possibilities:

- App availability and ignition
 - The availability of an app related with the event is marked with an icon at the top left corner of the TV.
 - To launch it, a button on the remote controller will activate a control panel, the app main menu.
- App main menu and configuration panel

¹<http://www.eitb.eus/en/>

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScope (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

- A slide-hiding dashboard appears to provide seven options to the user:
 1. Hide. Close the menu, end the set-up and continue with the experience.
 2. Add Device. Join another devices to a shared experience.
 - Different mechanisms are performed to get all the devices into the same application session to build a synchronous, seamless and consistent experience. Some are almost transparent for the user, while another ones will require a more explicit user interaction.
 - When a device joins or leaves the app experience, the other get feedback with a notification of connection or disconnection.
 3. Layouts. Set up the preferred info arrangement
 - The user can easily select the visual arrangement that better fits his preferences. The elections prototype has four different layouts available:
 - PiP (Picture-in-Picture)
 - Grid
 - Carousel
 - Divided
 - From any of the devices that compose the experience he/she can configure it.
 - On top of the interface the user can see all the time which device he/she is setting up.
 - Default templates for visual scheduling are provided at each device when the application is started.
 4. Cameras. Decide which video signals are more interesting and compose personalised elections night experience.
 - Five video components associated to this section
 - They show the video signal of the TV or the one of each party-headquarter.
 - They can be switched on/off.
 - They can be muted and unmuted.
 - They are synchronised.
 - From any of the devices that compose the experience the user can configure them.
 - On top of the interface the user can see all the time which device he is setting up.
 - According to the video and audio set up the user can start enjoying the show by using the “hide” option where all the video signals are synchronised.
 - *Based on WP4 Motion a common clock will achieve a perfect synchronisation of the different cameras.*
 5. Twitter. Watch live social activity with specific hashtags and trending topics. Two components are associated to this section:
 - Twitter viewer

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

- It shows the available tweets in real-time related to a selected hashtag.
 - It can be switched on/off.
 - A hashtag filter/search can be performed.
 - Trending map
 - It shows which is the trending political party in each moment for every province and for the whole region.
 - It can be switched on/off.
 - From any of the devices that compose the experience the user can configure them.
 - On top of the interface the user can see all the time which device he is setting up.
 - According to the chosen configuration the user can start enjoying the show by using the “hide” option.
 - *Based on WP4 timing a common clock will achieve a perfect synchronisation of the information.*
6. Radio. Switch the audio to a radio signal.
- A radio component is associated to this section.
 - It shows an equalizer and the logo of the active radio broadcaster.
 - The equalizer shows if it is muted or not.
 - It can be switched on/off.
 - The broadcaster can be selected.
 - From any of the devices that compose the experience the user can configure it.
 - On top of the interface the user can see all the time which device he is setting up.
 - According to the chosen audio the user can start enjoying the show by using the “hide” option.
 - *Based on WP4 timing its timeline enables a perfect synchronisation of the audio.*
7. Graphics. Customize live infographics including past results, time evolution, specific parties or localizations. Two components types are related to this section:
- Table component
 - It shows real-time counting results data in a table format
 - Vote count
 - 2015 real-time count highlight dynamism (evolving value).
 - 2011 past results.
 - Party name.
 - Party logo.
 - Vote count.

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

- Vote percentage.
 - Seats.
 - It can be switched on/off.
 - It is possible to select the city (main cities in the Basque area):
 - IRUÑA – PAMPLONA
 - VITORIA – GASTEIZ
 - BILBO – BILBAO
 - DONOSTIA – SAN SEBASTIAN
 - It is possible to select present or past data.
- Four graphics components (one per main city, with the graphic results of the counting information)
 - They show:
 - City or region name.
 - 2015 bars of real-time count highlight dynamism (evolving value).
 - 2011 bars of past results.
 - Party Icon.
 - They can be switched on/off.
- From any of the devices that compose the experience the user can configure them.
- On top of the interface the user can see all the time which device he is setting up.
- According to the chosen configuration the user can start enjoying the show by using the “hide” option.
 - *Based on WP4 timing its timeline enables a perfect synchronisation of all the different contents.*
- Controller: there is a controller element that can be unfolded on the bottom of the screen when the configuration panel is closed and that allows to manage the timeline of the application with the following functionalities:
 - Play/Pause.
 - Slider (progress bar of the video)
 - Include bookmarks.
 - Go Back or Forward to bookmarks of the video (when a politician is talking, etc.).
 - Local time.

Because the different situations where the different elements of the application (visual components, the menu, the panel and the controller) will be displayed, they satisfy a responsive design by means of media queries from 100x100px to 1280x1024px. These situations will depend on the display size for all the elements and on the

Project

Dynamic Media Service Creation, Adaptation and Publishing on Every Device
MediaScope (610404)

Document Title

D6.4 Final functional and technical testing report

Version

1.1

Date

11th May

Author

VIC, IRT, BBC

layout for the visual components.

3.1.3. Testing metrics

This section provides a complete description of the functional and technical metrics to be analysed in elections prototype in order to see how it works.

Functional testing metrics

The following table shows the functional tasks that the prototype should be able to fulfil.

Testing domain	Task
General testing	Start the application.
	Open different application sessions.
	Detection of the device type.
User interface testing	Device icon visualization at the top-left corner on a TV, STB, laptop, tablet and smartphone to identify each device.
	Show Menu and configuration panel at any time on a TV, STB, laptop, tablet and smartphone by clicking the device icon.
	Hide Menu and configuration panel at any time on a TV, STB, laptop, tablet and smartphone by clicking the hide option in the menu.
	Menu shows the device identification icon at the top side.
	Navigate to each section on the menu directly for the Laptop, Tablet and Smartphone while using Up, Down, Select and Back for the TV.
	Different colour for the current option in the menu.
	Visualization of the joined devices at the top while navigation over the different options in the menu. The device being configured is represented with a different colour.
	Open Add Device section: a QR code and a shortened link should appear.
	Open Layouts section: The set of available templates should appear, each one represented with an icon. The active layout present a different border colour.
	In layout section navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting a camera then video or audio. While navigating through the layouts Back goes to the "Menu".
	Open Cameras section: The set of available "Cameras" should appear identified with a logo and including a video on/off (show/hide) and an audio on/off (mute).
	In cameras section navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting a camera then video or audio. While navigating through the cameras Back goes to the "Menu".
	Open twitter section: the hashtag filter and the activators of twitter viewer trending map should appear.
In twitter section navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting the element then their different setup elements. While navigating through these elements Back goes to the "Menu".	
Open radio section: The radio activator and the active broadcaster logo should appear on the left column while the set of available radio broadcasters appear on the right column. The selected broadcaster logo should be in colour while the remaining available broadcasters' logos in grey palette.	
In radio section navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting the column then their different setup elements. While navigating	

Project

Dynamic Media Service Creation, Adaptation and Publishing on Every Device
MediaScope (610404)

Document Title

D6.4 Final functional and technical testing report

Version

1.1

Date

11th May

Author

VIC, IRT, BBC

Testing domain	Task
	<p>through these elements Back goes to the "Menu".</p> <p>Open graphics section: a table activator should appear on the left column being possible to select the city and the year of the data to show. The selected city should have a different colour border and the selected year too. Also four graphic activator should appear on the right column, one for each city.</p> <p>In graphics section, navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting the column (for the table or the graphs) and then their different setup elements. While navigating through these elements Back goes to the "Menu".</p> <p>Video components: One should show the main TV show and the other, the party-headquarters cameras.</p> <p>Twitter viewer component: it should show the tweets related to each moment, with the user name, the photo and the text.</p> <p>Trending map component: it should show the trending political party in each moment for every province and for the whole region.</p> <p>Radio component: it should show a radio interface allowing to hear the radio audio.</p> <p>Table component: it should show in a table format the counting information in each moment (party, vote number, % and seats) under a configuration (year and place).</p> <p>Graphic components: it should show the counting results in bar graph format in each moment comparing them with past results.</p> <p>PIP layout: the component with the lowest order should appear in fullscreen while the other in a reduced size in a column on the right side of the screen. When a component from the column is clicked it is changed to the fullscreen position replacing its position.</p> <p>Grid layout: components should be organized in a grid, taking into account the preferred width, aspect ratio and order. The screen should be divided in cells of different sizes, leaving empty slots at the bottom side.</p> <p>Carousel layout: components should be organized in a carousel being able to slide them to left and right.</p> <p>Divided layout: components should be organised dividing the screen in equal size areas, one for each component. The components should be ordered taking into account the priority from the left to the right and from the top to the bottom.</p> <p>Tab visualization on the bottom of the screen to unfold the controller on a TV, STB, laptop, tablet and smartphone.</p> <p>Open the controller by clicking the tab on the bottom of the screen.</p> <p>Play, pause, rewind and forward buttons visualization to control the timeline.</p> <p>Visualization of a slider for timeline of the application with time marks.</p> <p>Numerical visualization of the moment time.</p> <p>Close the controller by clicking the tab again.</p>
Association testing	<p>Associate a new device using qr code.</p> <p>Associate a new device using shorted url.</p> <p>Notifications received when a device is connected/disconnected.</p>
Local control testing	<p>Add/remove a camera to the same device.</p> <p>Mute/Unmute an added camera to the same device.</p> <p>Add/remove the twitter viewer to the same device.</p> <p>Add/remove the trending map to the same device.</p> <p>Add/remove the radio to the same device.</p> <p>Add/remove the table to the same device.</p>

Testing domain	Task
Remote control testing	Add/remove a graphic to the same device.
	Change the layout of the same device.
	Detection of the devices types.
	From any associated device other devices can be configured (no master and slaves).
	Add/remove a camera to a remote device.
	Mute/Unmute an added camera to a remote device.
	Add/remove the radio to a remote device.
	Add/remove the twitter viewer to a remote device.
	Add/remove the trending map to a remote device.
	Add/remove the table to a remote device.
	Add/remove a graphic of a remote device.
	Remote device's configuration panel is updated when some of its component is added/removed.
	Change the layout of a remote device.
Slide carousel to change components visualization of a remote device.	
Remote device's configuration panel is updated when its layout is changed.	
Timeline testing	Play/Pause the application.
	Rewind/ Forward in the timeline.
	Time jumps in the timeline (time-shifting)

Table 1: Functional testing tasks

Technical testing metrics

The technical metrics will rest on these features.

Domain	Topic	Metric	Parameters
Operational	UI responsiveness	Configuration panel's responsiveness	A subjective perception of a developer with Web development skills to appreciate the responsiveness of the configuration panel. 0% would be not responsive and 100% very responsive (acceptable on all devices).
		Controller's responsiveness	A subjective perception of a developer with Web development skills to appreciate the responsiveness of the controller. 0% would be not responsive and 100% very responsive (acceptable on all devices).
		Components' responsiveness	A subjective perception of a developer with Web development skills to appreciate the responsiveness of each component. 0% would be not responsive and 100% very responsive (sizes having sense).
		Layouts' responsiveness	A subjective perception of a developer with Web development skills to appreciate the responsiveness of each layout. 0% would be not responsive and 100% very responsive (acceptable on all devices).
	Interoperability	Platforms supported	OS
Browsers			
Platforms			
Performance	Scalability and Concurrence	Number of devices in the same session	Measure that the prototypes reaches a maximum of 5 devices in the same session, which is considered a dense configuration for a single user.
		Number of users using different sessions	Measure that the prototypes reaches more than 5 users in different sessions. Evaluate the architectural design to enable multiple user at the same time.

Domain	Topic	Metric	Parameters
	Latency	For local setup	Time for switching on a component in the same device.
			Time for changing the layout in the same device.
			Time to receive a motion update in the same device.
		For remote setup	Time for switching on a component in a remote device.
			Time for changing the layout in a remote device.
			Detection of the connection of a new device.
Jitter	For local media	Non appreciable delay between components in the same device.	
	For remote media	Bearable synchronization between different devices.	
Technological	Universal vs Experimental	Standard stack level of compliance/correctness	Technologies requiring experimental/limited environment over. Specific implementations for specific environments.
	Standard vs Ad-hoc	Standards involved	Mention the employed standards.
	Effectiveness	MediaScape libraries overhead	MediaScape libraries loading. Media content loading.
Prototyping	Flexibility and modularity	Flexibility of the libraries to create prototypes	A subjective perception of a developer with Web development skills to use the libraries to create a prototype.
		Flexibility of the prototype to add new functionalities.	A subjective perception of a developer with Web development skills to change the functionalities of the prototype.
	Expressiveness	The nomenclature of the libraries	A subjective perception of a developer with Web development skills to use the libraries, if the nomenclature is adequate.
	Openness	How interoperable are the libraries	A subjective perception of a developer with Web development skills to use the libraries separately or to exchange one of them with another existing library.

Table 2: Technical testing metrics

3.1.4. Testing results

This section presents the obtained results after having performed both functional and technical tests in elections prototype.

Functional testing results

Testing domain	Task	Result (✓ ~ ✗)
General testing	Start the application.	✓
	Open different application sessions.	✓
	Detection of the device type.	✓
User interface testing	Device icon visualization at the top-left corner on a TV, STB, laptop, tablet and smartphone to identify each device.	✓
	Show Menu and configuration panel at any time on a TV, STB, laptop, tablet and smartphone by clicking the device icon.	✓
	Hide Menu and configuration panel at any time on a TV, STB, laptop, tablet and smartphone by clicking the hide option in the menu.	✓

Testing domain	Task	Result (✓ / ~ / x)
	Menu shows the device identification icon at the top side.	✓
	Navigate to each section on the menu directly for the Laptop, Tablet and Smartphone while using Up, Down, Select and Back for the TV	~
	Different colour for the current option in the menu.	✓
	Visualization of the joined devices at the top while navigation over the different options in the menu. The device being configured is represented with a different colour.	✓
	Open Add Device section: a QR code and a shortened link should appear.	✓
	Open Layouts section: The set of available templates should appear, each one represented with an icon. The active layout present a different border colour.	✓
	In layout section navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting a camera then video or audio. While navigating through the layouts Back goes to the "Menu".	~
	Open Cameras section: The set of available "Cameras" should appear identified with a logo and including a video on/off (show/hide) and an audio on/off (mute).	✓
	In cameras section navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting a camera then video or audio. While navigating through the cameras Back goes to the "Menu".	~
	Open twitter section: the hashtag filter and the activators of twitter viewer trending map should appear.	✓
	In twitter section navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting the element then their different setup elements. While navigating through these elements Back goes to the "Menu".	~
	Open radio section: The radio activator and the active broadcaster logo should appear on the left column while the set of available radio broadcasters appear on the right column. The selected broadcaster logo should be in colour while the remaining available broadcasters' logos in grey palette.	✓
	In radio section navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting the column then their different setup elements. While navigating through these elements Back goes to the "Menu".	~
	Open graphics section: a table activator should appear on the left column being possible to select the city and the year of the data to show. The selected city should have a different colour border and the selected year too. Also four graphic	✓

Testing domain	Task	Result (✓ / ~ / x)
	activator should appear on the right column, one for each city.	
	In graphics section, navigate to each item directly for the Laptop, Tablet and Smartphone while Up, Down, Left, Right, Select and Back for the TV, first selecting the column (for the table or the graphs) and then their different setup elements. While navigating through these elements Back goes to the "Menu".	~
	Video components: One should show the main TV show and the other, the party-headquarters cameras.	✓
	Twitter viewer component: it should show the tweets related to each moment, with the user name, the photo and the text.	✓
	Trending map component: it should show the trending political party in each moment for every province and for the whole region.	✓
	Radio component: it should show a radio interface allowing to hear the radio audio.	✓
	Table component: it should show in a table format the counting information in each moment (party, vote number, % and seats) under a configuration (year and place).	✓
	Graphic components: it should show the counting results in bar graph format in each moment comparing them with past results.	✓
	PiP layout: the component with the lowest order should appear in fullscreen while the other in a reduced size in a column on the right side of the screen. When a component from the column is clicked it is changed to the fullscreen position replacing its position.	✓
	Grid layout: components should be organized in a grid, taking into account the preferred width, aspect ratio and order. The screen should be divided in cells of different sizes, leaving empty slots at the bottom side.	✓
	Carousel layout: components should be organized in a carousel being able to slide them to left and right.	✓
	Divided layout: components should be organised dividing the screen in equal size areas, one for each component. The components should be ordered taking into account the priority from the left to the right and from the top to the bottom.	✓
	Tab visualization on the bottom of the screen to unfold the controller on a TV, STB, laptop, tablet and smartphone.	✓
	Open the controller by clicking the tab on the bottom of the screen.	✓
	Play, pause, rewind and forward buttons visualization to control the timeline.	✓
	Visualization of a slider for timeline of the application with time marks.	✓
	Numerical visualization of the moment time.	✓
	Close the controller by clicking the tab again.	✓

Testing domain	Task	Result (✓ / ~ / x)
Association testing	Associate a new device using qr code.	✓
	Notifications received when a device is connected/disconnected.	✓
Local control testing	Add/remove a camera to the same device.	✓
	Mute/Unmute an added camera to the same device.	✓
	Add/remove the twitter viewer to the same device.	✓
	Add/remove the trending map to the same device.	✓
	Add/remove the radio to the same device.	✓
	Add/remove the table to the same device.	✓
	Add/remove a graphic to the same device.	✓
	Change the layout of the same device.	✓
Remote control testing	Detection of the devices types.	✓
	From any associated device other devices can be configured (no master and slaves).	✓
	Add/remove a camera to a remote device.	✓
	Mute/Unmute an added camera to a remote device.	✓
	Add/remove the radio to a remote device.	✓
	Add/remove the twitter viewer to a remote device.	✓
	Add/remove the trending map to a remote device.	✓
	Add/remove the table to a remote device.	✓
	Add/remove a graphic of a remote device.	✓
	Remote device's configuration panel is updated when some of its component is added/removed.	✓
	Change the layout of a remote device.	✓
	Slide carousel to change components visualization of a remote device.	✓
Remote device's configuration panel is updated when its layout is changed.	✓	
Timeline testing	Play/Pause the application.	✓
	Rewind/ Forward in the timeline.	✓
	Time jumps in the timeline.	✓

Table 3: Functional testing results

Every task on the table has been executed by a developer and as it can be seen the prototype is able to perform most of them. So, it can be said that it fulfils the main functional requirements described in section 3.1.2. Only navigation problems have been detected in the television, where the functionality itself is developed to work with the remote control, but it has to be developed a more intuitive order to be able to navigate in an easier way.

Technical testing results

Domain	Topic	Metric	Parameters	Result
Operational	UI responsiveness	Configuration panel's responsiveness	A subjective perception of a developer with Web development skills to appreciate the responsiveness of the configuration panel. 0% would be not responsive and 100% very responsive (acceptable on all devices).	100%
		Controller's responsiveness	A subjective perception of a developer with Web development skills to appreciate the responsiveness of the controller. 0% would be not responsive and 100% very responsive (acceptable on all devices).	100%
		Components' responsiveness	A subjective perception of a developer with Web development skills to appreciate the responsiveness of each component. 0% would be not responsive and 100% very responsive (sizes having sense).	100%
		Layouts' responsiveness	A subjective perception of a developer with Web development skills to appreciate the responsiveness of each layout. 0% would be not responsive and 100% very responsive (acceptable on all devices).	100%
	Interoperability	Platforms supported	OS	Windows/Linux/Android
			Browsers	Chrome/Opera/Firefox (polyfills)
			TV Platforms	HbbTV, Android Dongles
Performance	Scalability and Concurrency	Number of devices in the same session	Measure that the prototypes reaches a maximum of 5 devices in the same session, which is considered a dense configuration for a single user.	✓
		Number of users using different sessions	Measure that the prototypes reaches more than 5 users in different sessions. Evaluate the architectural design to enable multiple user at the same time.	✓
	Latency	For local setup	Time for switching on a component in the same device.	Always under 250ms
			Time for changing the layout in the same device.	Always under 200ms
			Time to receive a motion update in the same device.	Average under 50ms
		For remote setup	Time for switching on a component in a remote device.	Always under 500ms
			Time for changing the layout in a remote device.	Always under 550ms
			Detection of the connection of a new device.	Always under 1s
	Jitter	For local media	Non appreciable delay between components in the same device.	Fully synchronized
		For remote media	Bearable synchronization between different devices.	Average desynchronization 12.76 ms
Technological	Universal vs Experimental	Standard stack of level compliance/correctness	Technologies requiring experimental/limited environment over.	Unified media pipeline in Android Web Platform features
			Specific implementations for specific environments.	HbbTV 1.5 functionalities layer
	Standard vs Ad-hoc	Standards involved	Mention the employed standards.	Mainly based in HTML5

Domain	Topic	Metric	Parameters	Result
	Effectiveness	MediaScape libraries overhead	MediaScape libraries loading	Different depending on the device
			Media content loading	Different domains for media contents are needed in order to avoid bottlenecks at the loading.
Prototyping	Flexibility and modularity	Flexibility of the libraries to create prototypes	A subjective perception of a developer with Web development skills to use the libraries to create a prototype.	High
		Flexibility of the prototype to add new functionalities.	A subjective perception of a developer with Web development skills to change the functionalities of the prototype.	High
	Expressiveness	The nomenclature of the libraries	A subjective perception of a developer with Web development skills to use the libraries, if the nomenclature is adequate.	Good
	Openness	How interoperable are the libraries	A subjective perception of a developer with Web development skills to use the libraries separately or to exchange one of them with another existing library.	Good

Table 4: Technical testing results

User Interface responsiveness

The prototype has been tested in many different devices as PCs, Tablets, SmartPhones and televisions and it has been checked that the user interface is totally responsive in reasonable conditions (it does not make sense to see a web component smaller than 100x100px). Web components, the configuration panel, the controller and the layouts are correctly adapted in every checked situation.

Interoperability

The prototype has also been tested in different browsers, OS and platforms:

- Browsers: Chrome/Opera and Firefox
- OS: Windows, Linux, Android
- Platforms: HbbTV (Panasonic TX-42ASW654 LED TV with HbbTV 1.5)

It has to be said that the development of the prototype has been centered in Chrome/Opera browsers where it works in a native way, but it also could run in Firefox, with a poorer performance because of some native inconsistencies. For example, some polyfills are needed in order to be able to use Web Components. This fact clearly affects to the general performance of the prototype.

In the case of operative systems, all the accomplished tests have worked correctly without any significant difference in the between Windows or Linux or Android.

Finally, the prototype also works in HbbTV with all its functionalities, but with a limited performance. This is due to the additional polyfills and changes it needs. Apart from the Web Components polyfill needed for Firefox, a polyfill is required for the promise functionality. Moreover, in order to be able to synchronize the media elements, mediaSync module has to be adapted². And finally HbbTV 1.5 functionalities have to be integrated using an adapter layer. Even so, with all these changes the prototype performs all the main functionalities.

²VIC and NOR worked together successfully, cooperating WP4 and WP6 to improve the timing mechanisms to be fully operative with HbbTV.

Scalability and concurrence

In order to test the scalability of the prototype two tests have been performed. On one hand, it has been checked that it supports until 5 devices in the same session, and on the other hand, that it supports more 5 different sessions. Both test have had a positive result without any problem, so it can be said that Elections prototype is a scalable prototype.

Latency

The objective here has been to know how fast reacts the prototype to user interaction or devices connection. For that five different tests have been accomplished.

Time to change the layout in the same device

It has been measured the time needed by the prototype in order to change the layout when clicking in the same device layout the change is wanted. For that, the time since the click event is caught until the layout is rendered has been taken into account. These are the obtained results:

Repetition	1	2	3	4	5	6	7	8	9	10
Time (ms)	190	133	114	101	159	125	186	119	154	159
Average	144 ms									

Table 5: Time to change the layout in the same device

The obtained values vary between 101 and 190ms with an average value of 144ms. All the layouts in the prototype have not the same complexity, so this explains the differences between the values. Even so, all the values are under 200ms which is a more than acceptable value at user level.

Time to change the layout in a remote device

It has been measured the time needed by the prototype in order to change the layout when clicking in a device for a remote change. For that, the time since the click event is caught in the first device until the layout is rendered in the second device has been taken into account. The devices clocks are supposed to be synchronized. These are the obtained results:

Repetition	1	2	3	4	5	6	7	8	9	10
Time (ms)	310	268	550	303	460	297	435	383	326	298
Average	363 ms									

Table 6: Time to change the layout in a remote device

The obtained results vary between 268 and 550ms with an average value of 363ms. Again, all the layouts have not the same complexity, so we find differences between the values. It can also be appreciated that the needed time for a remote change is two or three times the needed for a local change, which is reasonable. In this case, all the values are under 550ms which is also an acceptable value at user level.

Time to switch-on a component in the same device

It has been measured the time needed by the prototype in order to activate a component when clicking in the same device the activation is wanted. For that, the time since the click event is caught until the component is rendered has been measured. These are the obtained results:

Repetition	1	2	3	4	5	6	7	8	9	10
Time (ms)	157	151	148	238	232	216	198	187	202	221
Average	195 ms									

Table 7: Time to switch on a component in the same device

The obtained results vary between 148 and 238ms with an average value of 195ms. The variation could be due to the differences between the components that have been activated (video, twitter, etc). In this case, all the values are under 250ms, thus, an acceptable value at user level.

Time to switch-on a component in a remote device

It has been measured the time needed by the prototype in order to activate a component when clicking in a device for a remote change. For that, the time since the click event is caught in the first device until the component is rendered in the second device has been measured. The devices clocks are supposed to be synchronized. These are the obtained results:

Repetition	1	2	3	4	5	6	7	8	9	10
Time (ms)	232	478	454	231	304	333	413	498	267	324
Average	353 ms									

Table 8: Time to switch on a component in a remote device

The obtained results vary between 232 and 498ms with an average value of 353ms. The variation could be due to the differences between the components that have been activated (video, twitter, etc). It can also be appreciated that the needed time for a remote change is two or three times the needed for a local change, which is reasonable. In this case, all the values are under 500ms, thus, again an acceptable value at user level.

Time to detect the connection of a new device

It has been measured how much time a device that is running the application needs to be aware that another device has been connected to the same session. For that, the time since the application is started in the new device until the first device obtains the notification and changes its configuration panel has been measured. The devices clocks are supposed to be synchronized. These are the obtained results:

Repetition	1	2	3	4	5	6	7	8	9	10
Time (ms)	800	918	584	789	498	425	439	581	474	630
Average	614 ms									

Table 9: Time to detect the connection of a new device

The obtained results vary from 425 to 918ms, with an average value of 614ms, so always under a second.

Time to receive a motion update in the same device

In this case, it has been measured the time needed by the prototype in order to play/pause the timeline in the clicked device. For that, the time since the click event is caught until controller of the same device receives the *timeupdate* event which specifies the motion's velocity change has been measured. These are the obtained results:

Project

Dynamic Media Service Creation, Adaptation and Publishing on Every Device
MediaScape (610404)

Document Title

D6.4 Final functional and technical testing report

Version

1.1

Date

11th May

Author

VIC, IRT, BBC

Repetition	1	2	3	4	5	6	7	8	9	10
Time (ms)	43	45	43	41	45	41	45	43	41	44
Average	43 ms									

Table 10: Time to receive a motion update in the same device

The obtained results vary between 41 and 45ms with an average value of 43ms. The values do not present high variations and suppose acceptable values for user interaction.

Time to receive a motion update in the remote device

It has been measured the time needed by the prototype in order to play/pause the timeline in remote device. For that, the time since the click event is caught in the clicked device until controller of the remote device receives the *timeupdate* event which specifies the motion's velocity change has been measured. The devices clocks are supposed to be synchronized. These are the obtained results:

Repetition	1	2	3	4	5	6	7	8	9	10
Time (ms)	91	81	95	98	100	88	90	94	99	95
Average	93 ms									

Table 11: Time to receive a motion update in the remote device

The obtained results vary between 81 and 100ms with an average value of 93ms and the values do not present high variations. As it can be appreciated, the needed time to be aware of a motion update coming from a remote client is twice the needed time when the click is done in the same device. This makes sense as in the remote case the client has to request the change to online provider and then the provider has to notify it to the second client, while in the local case, the same client is the one who requests the update. In this case all the values are under 100ms which is also an acceptable value for user interaction.

Jitter

Here the synchronisation delays have been measured both between components in the same device or in different devices. For that two test have been performed:

Delay between different media elements in the same device

For testing the synchronization between components in the same device, it has been analysed the position in motion's *timeupdate* event that arrives to each component. These have been the obtained results:

e.pos in Comp A (s)	e.pos in comp B (s)	Difference (ms)
22.1918	22.1918	0
22.3901	22.2301	0
22.5901	22.5901	0
22.7902	22.7902	0
22.9903	22.9903	0
23.1903	23.1903	0
23.3904	23.3904	0
23.5903	23.5903	0
23.7921	23.7921	0
23.9901	23.9901	0

Table 12: Delay between components in the same device

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

Looking at the results in the table above, it can be appreciated that the synchronization between components in the same device is up to the ms. Therefore, it can be said that there is a non appreciable delay between components running in the same device.

Delay between components in different devices

For testing the synchronization between components in different devices, it has been analysed the position in motion's *timeupdate* event arrived to the component in each device. Coming the reference clock from the same online provider, the difference in ms has been measured.

e.pos in Comp A (s) Device 1	e.pos in comp B (s) Device 2	Difference (ms)
172.4036	172.4150	11.4
172.6036	172.6150	11.4
172.8035	172.8208	17.3
173.0035	173.0151	11.6
173.2035	173.2148	11.3
173.4036	173.4195	15.9
173.6036	173.6149	11.3
173.8039	173.8149	11
174.0038	174.0189	15.1
174.2036	174.2149	11.3
Average		12.76

Table 13: Delay between components in different devices

Looking at the results in the table above, it can be appreciated that the desynchronisation between components in the different devices is bigger but not a high value. Taking into account that we start appreciating the audio desynchronisation at 50ms approximately, we can say that components are synchronised and therefore, that there is a non appreciable delay between components running in different devices.

It is true that in media elements some jumps have been detected when the application runs in more than one device. However this issue could be related to a loading problem of the browser.

Universal vs experimental

The prototype itself does not require any experimental environment to fulfil the basic features but, there are some flags in Chrome, which their activation improves the performance of the prototype:

- Unified media pipeline in Android: enables the possibility of using variable playback rates on Android allowing much tighter synchronization and nicer user experience.
- Web platform features: in order to be able to use the grid layout correctly, it is necessary to activate this flag, which makes possible the CSS Grid Layout algorithm to work. This way the components will be automatically placed into an unoccupied space in the grid container taking into account their priority and size.

On the other hand, and as it has been said before, some layers have to be added to the prototype in order to add the HbbTV 1.5 functionalities.

Standard vs ad-hoc

Since one of the objectives of MediaScape is to build interoperable applications, this prototype has been

built on top of HTML5 standard, which works increasingly on a wider range of devices.

Apart from that, Elections prototype uses Web Components standard for its atomic units which present a wide set of advantages for the multi-device behaviour.

Finally, in the synchronization part, Shared Motion provides the multi-device timing support based on the timing object, that although it is not a standard, W3C advocates by its standardization in Multi-device Timing Community Group.

Effectiveness

In order to analyse how heavy is the prototype for the browser in a indicative way, two main events have been tracked in different devices:

- DOMContentLoaded: this event is triggered when the DOM and CSSOM are ready which means that the render tree can be built.
- Load: this event is triggered when all the needed resources to start the application are loaded.

Tables 14 and 15 show the results for both events in next devices:

- PC: Processor Intel® Core™ i5-4590 CPU @ 3.30GHz × 4
- Tablet: Samsung Tab SM-T320
- SmartPhone: MotoG (2nd Generation)

DOMContentLoaded event			
Repetition	Time PC (s)	Time Tablet (s)	Time SmartPhone (s)
1	1.12	3.63	4.25
2	0.924	2.26	4.37
3	0.890	1.96	4.42
4	1.01	5.29	4.37
5	0.893	2.12	4.31
6	0.9	2.08	4.26
7	0.901	2.09	4.37
8	0.875	2.10	4.93
9	0.853	3.44	4.23
10	0.975	2.42	4.38
Average	0.9341	2.739	4.389

Table 14: DOMContentLoaded event results

Load event			
Repetition	Time PC (s)	Time Tablet (s)	Time SmartPhone (s)
1	4.18	9.47	7.47
2	3.94	8.14	7.65
3	3.90	7.08	7.53
4	4.02	11.61	7.60
5	4.06	7.33	7.60
6	3.91	8.01	7.62
7	3.91	7.46	7.57
8	3.89	7.40	8.17
9	3.93	9.72	7.47
10	3.99	8.16	7.59
Average	3.973	8.438	7.627

Table 15: Load event results

As it can be seen the time needed by the prototype for both events is more than twice in a tablet than in a PC, while in the SmartPhone the loading is similar to the tablet. This results are logical as tablets and SmartPhones always have lower processing capacity.

Furthermore, an important fact related to the loading is the number of media elements having the same source. When executing in Chrome (This fact has not been analysed in other browsers), the browser only allows to have 6 sockets opened with the same dominion at the same time. Therefore, when the videos start preloading, a bottleneck is generated and it remains waiting for the rest of the resources causing a timeout and avoiding the libraries loading. So, in order to solve this problem media elements should be in different domains.

Flexibility and modularity

The prototype presents a high flexibility and modularity as on the one hand, all the employed libraries have been developed with the objective of being part of a modular architecture and, on the other hand, it has been built using require.js in order to load the different modules it uses. Thus, it is not difficult to add new modules or libraries to the prototype.

Apart from that, since it has been built on top of Web Components, which also contribute to the modularity, it is easy to add or remove functional blocks to the application.

Expressiveness

The prototype clearly satisfies expressiveness issues, as all the nomenclature used in the libraries describes adequately the performed tasks by each one of them. This eases the possibility of adding new rules for example for the multi-device behaviour.

Moreover, the fact of building the prototype using Web Components, gives additional expressiveness as they use custom tags in order to define each functional block of the application. Even more, the use of Polymer library to build those components eases the understanding of each one of them.

Openness

The openness of the main functionalities of the prototype is granted as all employed libraries have been developed in JavaScript and always in an open way, thinking about the possibility of extending or combining them with other libraries not developed in MediaScape. However, it has to be taken into account that additional external libraries have been used for instance, for the design of the configuration panel or the generation of the



Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

qr code.

3.1.5. Conclusions

Once the test have been performed, this section provides some general conclusions for the Elections prototype.

In general lines, we have seen that all the established functional requirements are fulfilled while from the technical testing we can conclude that Elections prototype is an interoperable, scalable and responsive multi-device Web application that allows local and remote interaction with non appreciable or bearable delays. Furthermore, the application is correctly synchronized between components both in the same or different devices. And finally, it achieves all the aforementioned with a high degree of flexibility, modularity, expressiveness and openness.

As a future work, it could be analysed how to improve the performance in different browsers or HbbTV as well as the loading process, but depending in part on the browsers evolution in their native implementations (e.g. Firefox uses polyfills for Web Components).

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

3.2. Scrollyteller

3.2.1. Definition

The goal of the Scrollyteller, created by BR and IRT, is to demonstrate a novel way for users to consume a radio show. On top of the linear audio played by the radio, the Scrollyteller app provides a second screen experience for the listener, including additional information such as text, images and even video streams, related to the ongoing radio program. For the end user, a nearly seamless user experience is required, which should:

- Discover the available radio devices in the home network.
- Provide a transition from the standard broadcasters web page to the Radio-UI with the second screen application.
- Offer a synchronized presentation of additional material to the listener based on standard HTML.
- Offer the choice for the listener to include other display devices into the presentation or even choose to switch completely to different devices (e.g. HbbTV, FireTV ...).

All this has been achieved by the use of MediaScape developments mainly from work packages WP3 and WP4. In detail the discovery of the in home radio device was realized through the SSDP developments. Service transition from the radios second screen to other display systems such as HbbTV and/or FireTV has been achieved by using DAIL functionalities.

For the end user testing (see deliverable D6.5) additionally a pure web based online version of the Scrollyteller prototype was developed. As this setup could not be tested on the DABerry prototype radios with the implemented discovery mechanisms, the testing situation was simulated with a combination of laptop / PC and the mobile device (smartphone/tablet). No additional discovery/pairing was possible after the initial PIN based device pairing. In order to reach the widest possible audience, the challenge was to successfully implement the Scrollyteller prototype on a wide range of possible devices.

During the end user testing at Bayerischer Rundfunk, additionally a live test with of the shelf radio devices was performed (see deliverable D6.5). The test-environment therefore was changed, as a simple QR code scanning needs to be used for the initial device pairing with the second screen. No additional discovery/pairing was possible after the initial PIN based device pairing.

3.2.2. Functional and Operational Description

In the following, the main features and user functionalities of the Scrollyteller application are described

- Initial bootstrap of the listener to the Scrollyteller use case.
 - The listener is able to use the standard broadcasters webpage to discover the in house radio and it's second screen capable UI.
 - When pressing the listen live button on the web page, the prototype implementation offers either the standard IP streaming in the webpage or the discovered radio device in the home network.
 - When the user chooses to listen on the dedicated radio device a HTML page with the radio UI is offered in the browser.
 - In the online user test, the listener starts an introduction web page, which guides him into the pairing of the mobile and PC/laptop parts of the test.

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScope (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

- When both devices show the correct test entry web page a pairing code is required.
 - In the live test at Bayerischer Rundfunk, the radio device shows a QR code transmitted as DAB Slideshow to the DAB Radio, which needs to be scanned by a standard QR code reading application.
 - When the radio show with the additional content is starting the listener can choose if the presentation should start in the radio's own UI or on an additional available display device in the home.
 - When the Scrollyteller content is available, a red button appears in the radio UI, giving the listener the choice to start the second screen application embedded into the UI of the radio.
 - If other display devices are discovered in the home network (HbbTV, FireTV), additionally enable the end user to select these for the content presentation.
- Once the Storyteller content presentation is started, the listener can follow the linear presentation of the accompanying content.
 - The broadcaster can choose to include interactive elements into the linear presentation, such as votings, questionnaires or comment buttons.

3.2.3. Testing metrics

The following section describes the metrics used for the functional and technical evaluation of the prototype.

Functional testing metrics

Testing domain	Task
General testing	Open the broadcaster's web page.
	Press "Play" button on the web page to listen live to the radio programme.
	Select the radio device (DABerry) as listening source of the radio programme.
	For online Testing:
	Browse to the introduction web page and use the displayed pairing code to pair the laptop/PC and mobile devices
	For live testing:
	Use QR code reader to perform the pairing of the mobile device
User interface testing	Check if the radio UI is responsive and general functionalities such as volume control, DynamicLabel (Radio Text) and Slideshow are working as expected.
	RadioWEB button on the radio UI is displayed, indicating the availability of the additional content.
	Press the RadioWEB button and start the content playback embedded in the radio UI.
	Follow the audio of the radio programme and the additional content presentation and evaluate, if the presentation is correctly timed and the heard and watched content somehow relate to each other.
Device discovery	Check if interactive pages in the Scrollyteller content are working when displayed.
	Radio device (DABerry) is found automatically by the broadcaster's web page
	Additional devices (if available) are offered as potential displays for the Scrollyteller presentation. <ul style="list-style-type: none"> • HbbTV televisions • Amazon FireTV

Table 16: Functional testing metrics

Technical testing metrics

The technical metrics will rest on these features.

Domain	Topic	Metric	Parameters
Operational	UI responsiveness	General Radio UI	A subjective perception of a developer with Web development skills to appreciate the responsiveness of the configuration panel. 0% would be not responsive and 100% very responsive (acceptable on all devices).
		Scrollyteller Contents	A subjective perception of a developer with Web development skills to appreciate the responsiveness of the controller. 0% would be not responsive and 100% very responsive (acceptable on all devices).
	Interoperability	Platforms supported	OS
			Browsers
		Platforms	
Latency	Switch to Radio/HbbTV/FireTV UI	Time for switch from HTML broadcasters page to second scree radio UI	
		Time for switch from second screen Radio UI to HbbTV/FireTV	
	Scrollyteller Contents	Seamless presentation and synchronisation with the linear radio content	
Technological	Universal vs Experimental	Standard stack level of compliance/correctness	Technologies requiring experimental/limited environment over. Specific implementations for specific environments.
	Standard vs Ad-hoc	Standards involved	Mention the employed standards.
	Effectiveness	MediaScape libraries overhead	MediaScape libraries loading. Media content loading.
Prototyping	Flexibility and modularity	Flexibility of the libraries to create prototypes	A subjective perception of a developer with Web development skills to use the libraries to create a prototype.
		Flexibility of the prototype to add new functionalities.	A subjective perception of a developer with Web development skills to change the functionalities of the prototype.
	Expressiveness	The nomenclature of the libraries	A subjective perception of a developer with Web development skills to use the libraries, if the nomenclature is adequate.
	Openness	How interoperable are the libraries	A subjective perception of a developer with Web development skills to use the libraries separately or to exchange one of them with another existing library.

Table 17: Technical testing metrics

3.2.4. Testing results

Testing domain	Task	Result (✓ / ✗)
General testing	Open the broadcasters web page.	✓
	Press "Play" button on the web page to listen live to the radio programme.	✓
	Select the radio device (DABerry) as listening source of the radio programme.	✓
	For online testing	
	Browse to the introduction web page and use the displayed pairing code to pair the laptop/PC and mobile devices	✓
	For live testing	
	Use QR code reader to perform the pairing of the mobile device	✓
	Check if the radio UI is responsive and general functionalities such a volume control, DynamicLabel (Radio Text) and Slideshow are working as expected.	✓
User interface testing	RadioWEB button on the radio UI is displayed, indicating the availability of the additional content.	✓
	Press the RadioWEB button and start the content playback embedded in the radio UI.	✓
	Follow the audio of the radio programme and the additional content presentation and evaluate, if the presentation is correctly timed and the heard and watched content somehow relate to each other.	✓
	Check if interactive pages in the Scrollyteller content are working when displayed.	✓
Device discovery	Radio device (DABerry) is found automatically by the broadcasters web page	✓
	Additional devices (if available) are offered as potential displays for the Scrollyteller presentation. <ul style="list-style-type: none"> HbbTV televisions Amazon FireTV 	✓ ~ As the HbbTV use-case required a relative new HbbTV 2.0 implementation, the reliability of the device discovery depended on the implementation status of the HbbTV 2.0 prototype.

Table 18: Functional testing results

Do main	Topic	Metric	Parameters	Result (✓ / ~ / x)
Operational	UI responsiveness	General Radio UI	A subjective perception of a developer with Web development skills to appreciate the responsiveness of the configuration panel. 0% would be not responsive and 100% very responsive (acceptable on all devices).	100%
		Scrollyteller Contents	A subjective perception of a developer with Web development skills to appreciate the responsiveness of the controller. 0% would be not responsive and 100% very responsive (acceptable on all devices).	100%
	Interoperability	Platforms supported	OS	Android, HbbTV
			Browsers	HbbTV 2.0, Chrome, FireTV browser component
Technological	Latency	Switch to Radio/Hbb TV/FireTV UI	Time for switch from HTML broadcasters page to second scree radio UI	Between 1 to 10 seconds. This depends on the fact, whether the DABradio is already tuned to he broadcast ensemble containing the requested audio programme. Additionally the reception quality also can alter the result, as the radio UI is not displayed completely until the tuning is finished.
			Time for switch from second screen Radio UI to HbbTV/FireTV	Always under 1s
		Scrollyteller Contents	Seamless presentation and synchronisation with the linear radio content	Always under 1s
	Standard vs Ad-hoc	Universal vs Experimental	Standard stack level of compliance /correctness	Technologies requiring experimental/limited environment over.
Specific implementations for specific environments.				
Effectiveness		MediaScap e libraries overhead	Media content loading.	Different depending on the device
Prototyping	Flexibility and modularity	Flexibility of the libraries to create prototypes	A subjective perception of a developer with Web development skills to use the libraries to create a prototype.	High
			A subjective perception of a developer with Web development skills to change the functionalities of the prototype.	Easy
	Expressiveness	The nomenclature of the libraries	A subjective perception of a developer with Web development skills to use the libraries, if the nomenclature is adequate.	Easy
			How interoperable are the libraries	A subjective perception of a developer with Web development skills to use the libraries separately or to exchange one of them with another existing library.

Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

Table 19: Technical testing results

3.2.5. Conclusions

As the Scrollyteller prototype is a very special use-case, which reveals its full functionalities only in an environment with working HbbTV2.0 and SSDP/DIAL capable devices, especially in the online and live test scenarios, the set-up of the test is sometimes hard to understand for test persons.

The discovery mechanisms used, worked fine and in a reliable way once set up correctly. The HbbTV 2.0 prototype used, was sometimes hard to handle from a developers point of view, because the status and the completeness of the necessary implementation was unknown.

One of the biggest challenges during the development of the Scrollyteller prototype was, especially for the wide range of possible devices in the online test, the functional requirement, that video playback has to start automatically on the device. On standard configurations, the installed web browsers prohibit the start of streaming video without explicit user interaction (a security mechanism). In advanced settings, however this can be deactivated, but for the online test this possibility was not an option. Therefore the decision was to find a JavaScript based workaround for that, which caused a very diverse behaviour in video playback handling on the devices.



Project		Document Title
Dynamic Media Service Creation, Adaptation and Publishing on Every Device MediaScape (610404)		D6.4 Final functional and technical testing report
Version	Date	Author
1.1	11th May	VIC, IRT, BBC

4. Conclusions

This document describes the technical testing activities and the functional tests performed to two of the final MediaScape prototypes: Scrollyteller prototype and Elections prototype. The goal of the functional testing has been to validate that the prototypes provide the functionality they were designed for. In both cases the functional tests have provided a positive result and moreover, they have been helpful to improve the developments. On the other hand, the technical tests have been done to objectively measure the technical implementation of the prototypes regarding previously established metrics. Since the prototypes rely on the MediaScape libraries, the technical tests have successfully validated the implemented libraries.

Overall, the technical and functional tests provide a very positive feedback. All the developed libraries has provided the expected functionality to the prototypes and the MediaScape architecture and vision has been followed successfully. MediaScape provides a very flexible, modular and interoperable solution, over Web standards, to provide multi-device media applications. The prototypes, based on the scenarios defined at the beginning of the project, have been implemented and on functionality-gaps appeared.

The interoperability has been addressed as the main issue: from a prototype to a real market adoption, compatibility with all platforms, browsers and devices is a key aspect. The MediaScape architecture has been design thinking in the future, and taking into account the current existing market. In this sense, current HbbTV provides some limitations that will be removed in the future, with a higher convergence of HbbTV and HTML5.