An architecture for second screen experiences based upon distributed social networks of people, devices and programs

W3C Web & TV Convergence Workshop,
12-13 March 2014, Munich, Germany

Njål Borch, Norut,
Bin Cheng, NEC,
Dave Raggett, W3C,
Mikel Zorrilla, Vicomtech-IK4

We acknowledge support from the EU via funding for the MediaScape project under grant agreement 610404.
Disclaimer: the ideas and opinions expressed in this paper are those of the authors, and do not represent the view of their affiliated organizations.

Abstract

This paper presents a vision for the future of broadcasting, focusing upon opportunities for enabling interactive social user experiences that exploit second screens, and accommodate the trend for viewers to watch programs at a time of their own choosing. A rich collection of use cases suggests the importance of a social network of people, devices and programs as a basis for coordinating complementary content and services across devices and groups of friends. We propose a distributed architecture along with an abstraction layer that supports a simple model of the social network, and enables underlying optimization of communication mechanisms without burdening developers. We finish with a brief analysis of business models.

Introduction

The Internet is changing the role of broadcasters, and demanding new ways of thinking about the kinds of services offered to their users. Surveys [*Deloitte Global Mobile Consumer Survey, UK data, May 2012*] show that many people are now comfortable with using their phone or tablet at the same time as the TV. This represents an opportunity for offering new kinds of second screen services with richer engagement with viewers. The same goes for time shifting. People are increasingly viewing content when they want, and this shows the demand for active engagement, as compared to passive viewing of content when it is broadcast. This points to opportunities for services featuring high levels of viewer interaction. In the text below, program refers to a TV or radio program and not to application programs, for which we use the terms applications or services. This paper is based upon work being done in the EU FP7 ICT MediaScape project:

- http://mediascapeproject.eu/

Use case analysis

The MediaScape project includes a mix of broadcasters and technology companies. We started by brainstorming about future user experiences and trying to free ourselves from assumptions about current technology and practices. Appendix 1 presents a set of use cases along with comments. We have used these to come up with a proposed architecture and implementation techniques that could enable the use cases. Here is a slightly extended example:

> *"Peter is watching a show from the national broadcaster while commuting. As the train nears the station he pauses. He is quite eager to see the end of the show though, so as soon as he enters home he wants to resume it. The broadcaster's app on the smart TV has his personal list of shows on the front page, with the most recent one on top. Peter simply presses the resume button, forgetting that "So you think you can dance" is about to start. There's an outcry as his daughter realises someone is blocking her plan to watch it. Peter submissively opens the show on his laptop too. For a brief moment the show is presented in tight synchrony by the TV and the laptop. At this point Peter can safely vacate the smart TV."*

The set of use cases prompted the following set of functional requirements. The technical implications are described in the next section.

1. Devices are part of a conceptual framework that includes their location, usage rules and relationships to people, programs and other devices
2. Devices can communicate with each other through this framework
3. Broadcasters can provide support for time shifting programs, allowing users to view content at a time of their own choosing, to start at a given time offset from the start of the program, and to pause and resume content
4. Broadcast programs are associated with metadata including time and spatial regions
5. The same is true for time shifted programs
6. The metadata links programs to complementary content and services
7. These services can provide a user interface to interact with them
8. Programs are associated with a session whilst they are running
9. Associated content and services can run on one or more devices, and can be moved from one device to another with a slick transition
10. Services may consist of subsidiary services that can be placed on different devices
11. Devices can be tightly synchronised to the program session
12. UI transitions can be tightly synchronized across devices
13. The user experience can be adapted to the class of device
14. The user experience can be adapted to user preferences
15. The conceptual framework in requirement 1 includes the set of friends for each person
16. People can form temporary social relationships e.g. to participate together in a game or competition

17. Notifications can be sent to users via their devices
18. Notifications can include a user interface, e.g. inviting the user to initiate an action
19. Notifications can also be sent to devices to initiate some specified behaviour

Architectural Choices

One major choice is between direct peer to peer messaging over a local network, or via a cloud based service. This paper concentrates on the latter approach and allows for a mix of local and remote devices, e.g. for shared experiences across a group of friends even though they are not at the same location. In principle the distinction between cloud and peer to peer can be blurred by introducing abstractions that hide the differences. This is something we hope to explore.

A social network of people, their devices and programs

The use cases point to the value of a social network service for people, devices and programs that allows friends to see what each other are watching or intending to watch, and to share comments and suggestions on what to watch. Each device connects to this service and provides presence information that can be shared with its friends. This includes the device's location. For instance, a device could detect when it is at home based upon the SSID of the WiFi network connection or its external IP address if assigned statically. Smart phones may be able to report their latitude and longitude based upon embedded GPS hardware. Further possibilities include Bluetooth beacons and other location sensing technologies.

The service consists of clients that run in the devices, and a cloud based component that runs on servers. The clients need to be able to handle asynchronous notifications, including being able to launch a browser for a given URI, and even to be able to wake a device from its standby power saving mode. This suggests the potential for a background agent that complements the MediaScape library used by web applications in the browser, providing capabilities that aren't possible in the browser. On Linux, the agent could be a daemon written in C, on Android a background service written in Java. Other possibilities include Node,js and even service workers invoked by web browsers. An important consideration is to make it easy for users to start using MediaScape on as broad a range of devices as possible, for instance on Android devices, by installing the MediaScape app from Google Play and registering the device with the social network on first run.

In some cases it may make sense to implement a virtual device as a basis for using devices which don't support the MediaScape architecture directly. In this situation, a MediaScape compatible device hosts a virtual device using drivers for proprietary interfaces to communicate with the associated physical device.

The social network provides a means for devices to pass messages to one another without needing to know the details of the underlying transport mechanism involved. This simplifies application development, e.g. avoiding the need for developers to worry about passage through firewalls, or whether a device is connected via WiFi or a cellular network. This involves a means for the social network to expose an API to applications through a JavaScript library.

One possibility would be via browser extensions. Unfortunately these are not supported on Android browsers. In principle, you could have a local HTTP server running on the device and use XMLHttpRequest to access it. However, it would be sufficient for apps to use a JavaScript library to communicate with the cloud component of the social network.

Including programs as entities in the social network allows users to search for programs, to see what complementary content and services are available for them, to read reviews and comment on them, and to receive notifications sent out by broadcasters about programs.

Programs, metadata and associated complementary content and services

For every program we need a way to access its metadata. This could be associated with the program as a whole, or with a particular time offset from the start of the program, or with time intervals during the program, or with regions, e.g. tracking the location of an actor throughout a sequence of frames. Metadata could be embedded as part of program content, or held separately. The metadata includes links to complementary content and services, for example, audio tracks in different languages, scene descriptions for people with sight impairments, reviews, links to related programs and so forth. A brief survey and discussion is given in Appendix 2.

Services are the basis for user interaction that goes beyond passively watching or listening to content. Broadcasters, program developers or third parties could provide services as Web applications. When a program is running (either live or time shifted), there is a program session which can be used as a basis for coordinating multiple devices. If you are participating as a group of friends, then you and your friends' devices are part of the same program session. The social network is aware of what is being shown on each device and whether the device is personal as for a tablet and smartphone, or shared as for a TV.

Synchronization and adaptation of user interfaces

The user interface for content and services running as part of the same program session can be synchronized across the participating devices. This is based upon dynamic measurement of network latency and delaying actions by the maximum latency for all of the devices. For web applications, this can be implemented with JavaScript and Web Sockets or HTTP.

HTML5 allows for animation with scripts (e.g. with the Canvas element) and with CSS, where it may be triggered by UI events such as mousing over a given element. For script driven animation, performance can be improved by allowing the browser to choose when to render changes. There is considerable variation in performance across different devices.

The use cases point to the need to be able to dynamically alter which services are running on what devices. Services can be implemented as separate web applications, or as components of the same web application. Adaptation to the device or to the user preferences could be implemented with client-side scripts that respond to changes and revise the document structure or styling. This is where the developer can take advantage of the application programming interface (API) exposed by the social network.

As examples, an application could query the device type to adapt the layout to best suit that device. If you are using a smartphone as a second screen, the application might choose to put some parts of the UI and content on the TV and other parts on the phone according to the judgement of the developers. This could be overridden by the user preferences. Notifications for users could be presented in different ways on different devices. Applications can adjust the presentation according to whether you are at home, at work or somewhere else.

Distributed versus Centralised Approaches

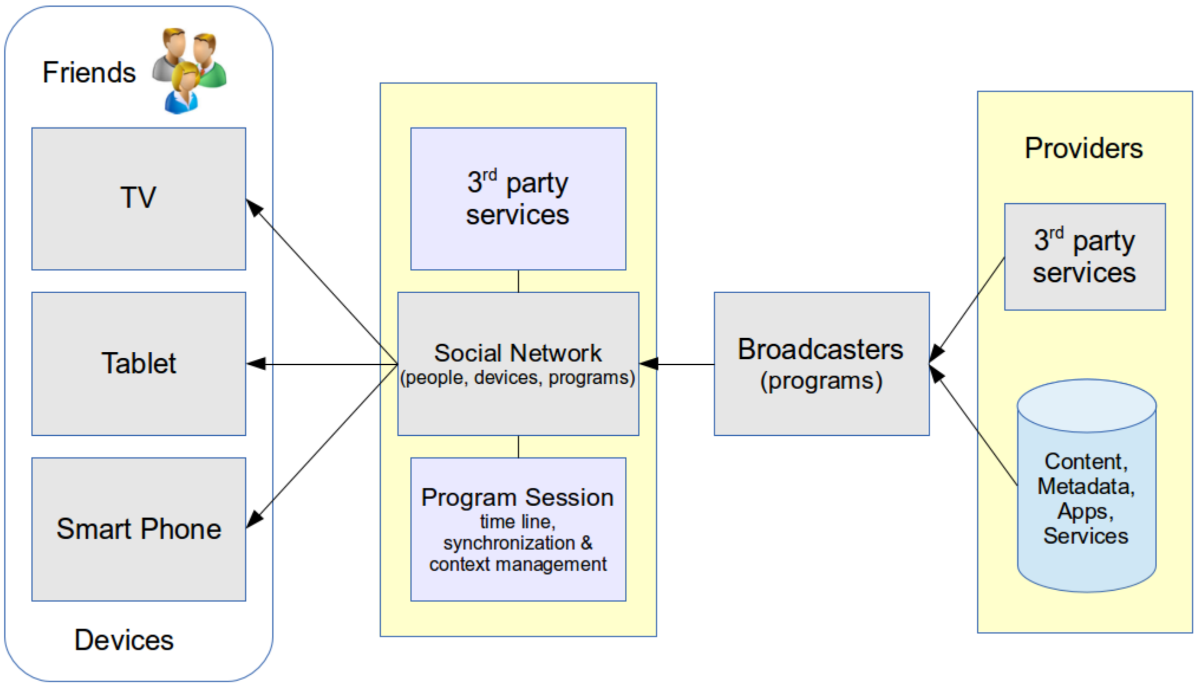
A social network of people, devices and programs could be implemented by a single vendor with a single website following the precedent set by Facebook, and this might be easier to explain to would be investors. However, sustained innovation over the long term is more likely to be attained through a rich ecosystem based upon a distributed approach together with the means for third parties to compete to provide value added services. This would rely upon a shared use of open standards and an open market of services.

A starting point for a distributed solution is to require each entity in the social network to have its own URI along with a standard means for updating the presence information, receiving notifications, accessing social relationships and so forth. Beyond the core set of APIs necessary for the social network, there are value added services, for instance, to search for programs that match a user's interests across a range of broadcasters. To satisfy the needs for privacy, communications are encrypted and entities are authenticated via public key cryptography. This is an active area of development, e.g. the work at the FIDO Alliance, but is outside the scope of this paper.

A distributed approach can also incorporate delegation. This can be exploited to support direct peer to peer communication where appropriate, e.g. from a tablet to a TV on the same WiFi network. To simplify applications and services, an abstraction layer is used to support a simple conceptual model of the social network and to hide the underlying communication mechanisms, thereby allowing for optimizations without burdening app and service developers.

Business Models

Broadcasters can make money through subscriptions (the license fee in the case of the BBC), pay as you go for individual programs or series, or from advertising. The bigger challenge is how to monetize the social network of people, devices and programs. One way is to exploit the data collected on user's viewing habits. Many people have shown themselves willing to allow this in exchange for free services, subject to agreeable privacy policies. An alternative is a subscription fee. Users may also be willing to pay extra for value added services. If there are standard interfaces for plugging in such services, then independent developers won't need to incur the cost of developing custom solutions from scratch for each social network operator.



The above diagram illustrates the role of broadcasters for producing programs with the help of the social network for coordination across devices and friends. 3rd party services are used by broadcasters to enhance their programs with independently developed content and services, and by the social network for services that add value to the core capabilities, including search across multiple broadcasters, something that individual broadcasters are unlikely to offer. An ecosystem of 3rd parties will be important for healthy innovation and competition.

Conclusion

This paper describes a high level architecture for rich engagement of users in social contexts with broadcast and timeshifted content, and second screens such as a TV with a tablet or smartphone. The approach is based upon a social network involving people, devices and programs, along with the means to support tight synchronization across devices. The key to success will be the ability for broadcasters and program developers to provide value added content and services that engage users' appetite for interaction, and to link them to programs via metadata.

Appendix 1 - Use Cases and Comments

The comments on each use case were created as an intermediate step for coming up with the functional and architectural requirements.

1. Commuting quiz show

   A man is on the bus, participating in the quiz with friends while watching a gameshow. When he gets home, his smart TV gives him the option to continue to watch the show and continue to play the quiz.

   Start on a smartphone, continue on home TV. Phone detects home WiFi network, and knows about home TV, and can send message to turn TV on and switch it to a given channel. Is this logic integrated into the smart phone's viewer app, or is it a separate agent running on the phone, on the TV on a home gateway, or even in the cloud?

2. Surrounding radio at home

   Andy browses to his favourite radio station's web page, which detects his nearby physical radio device and allows him to play the live show through it.

   Start on radio station website, discover local radio on WiFi network, offer button in website page to start playing given show. Can we assume that this logic is integrated into the web site's web page, or is it "injected" by a browser extension or agent as explained for use case 23?

3. Marauders Glastonbury Map

   While at the Glastonbury music festival, friends can see each others' locations and collaboratively decide which bands to see, with contextual information about which stages are busy and what's on. Accelerometers allow armchair participants to see where the best bits are.

   Assume each phone is online to a social network and that their presence information includes their location. A social app provided by the festival allows them to see each other's position on a festival site map, to browse and vote for which bands they want to see and how busy each of the stages are likely to be based upon vote data aggregated anonymously from all conference goers. The social app streams acceleration data whilst users are dancing. Friends stuck at home can thus see how much fun the friends at the festival are having. There are tough issues of battery life, given the scarcity of charging points at festivals!

4. Israeli Eurovision Night
Beatrix and friends create their own Eurovision night with a shared selection of songs they do Karaoke to, upload, share as a programme with a live custom voiceover, vote on, and share more widely.

This could be a karaoke app with a means to browse songs to add to the playlist, and to record audio to blend with the prerecorded music video for each song. The new recordings are saved to the cloud and available via a social app that allows viewers to vote on best performances.

5. Debate Newspaper
While watching a political debate on TV, Alex's phone notifies him that his favourite newspaper offers live commentary and statistics on the debate, as an overlay on his TV.

Not sure why the notification/offer is to the phone rather than to the TV that Alex is currently watching, which would seem more likely. Does the news site track what he is watching, or is there a background service triggered by the broadcast that knows about or which discovers sources of live commentary?

6. Video Call
Stephanie is in a video call on her phone with her father, while her son watches TV. She switches the phone's display to the TV so that her son can see and hear his grandfather and vice versa; the TV pauses while the call takes place.

The phone app for the video call discovers the TV (perhaps in advance) and is able to direct the TV to the video stream from the app's provider. The TV is able to pause and later resume the stream it was playing, assuming the distribution channel for the stream supports this functionality.

7. The Question
Tiara participates in an open journalism radio show over a period of time, and is able to participate and view comments even when watching on-demand or offline.

Not clear what is meant by participation. An app associated with the radio program could be used to view and add comments. This involves a means to synchronize the comments as part of a timeline.

8. Concert upload UGC
Jo and her friends participate in a singing contest which they found by capturing a link from an advert for a live show. They upload their versions

of songs to a site. They watch the live show by a singer, vote on what the singer sings next, and the best participant sings live from home with the band in the interval.

Link scanned by smartphone's QR code reader and used to start the show's app on the TV (if at home) or on their smart phone. The app allows users to upload their own wording for existing songs, and to stream themselves singing along with the band when it comes to their turn. This use case needs to be clarified!

9. Modularised Construction

Covering a high stakes Chess match, the producer is provided with a number of different visualizations. They can all be controlled by shared timing information, and can as such be "dragged and dropped" into a web page with no interaction or "glue" between the visualizations and can be personalised to different groups.

This is a set of alternative video streams of the same event. Users can dynamically switch between streams for different perspectives. The use case needs clarification.

10. Concert Video Wall

At a concert, the audience takes pictures of the band on stage. When they've taken the picture, it appears on the giant screen at the back of the stage making the audience feel part of the show.

This requires a social app for taking photos and uploading them to the concert's site. The site allows a moderator to select which photos to show on the big screen and in what order.

11. Public Screen Know More on Mobile Phone

Charlie arrives at the train station after work. He sees a news clip on the big screen and wants to know more. He connects his phone to the big screen. He needs to catch his train, but he can see more about the story on the train.

The big screen is in the station concourse. It includes a barcode that the phone can scan and use to launch the story. Other solutions include the use of Bluetooth Low Energy or even Infrared to signal the URI. In these cases, the phone provides a notification offering Charlie the choice on watching the news item on his phone.

12. Series Keeping Score

Eddie and Frank love the 'Great British Bake Off' TV programme. Every year they each pick a winner in the first episode. This year, Eddie is team Jon, and Frank is team Tom. Each week they pick a winner too, especially if someone is having a good day.

This is a social app provided by the broadcaster. Users can invite their friends to participate and pick winners at the start of each show.

13. Game Map Review

Eve and Mark find useful information about a game by moving their phones near an advert for it, including the location of shops that sell it, plus online reviews. The results are fed back to a server which uses the data to improve the experience for others.

This sounds like an NFC tag with an icon on the game's advert. The tag provides a discount coupon as well a link to information on where to purchase the game and user contributed reviews. The app involved uses info provided by the tag to identify the advert's geolocation.

14. Multi-source Group Recommendations / Playlist

Andy and Christine are suggested an evening's viewing from live and on-demand content by their TV, which remembers their preferences and can request group recommendations from an online service; their TV and can also be controlled by a connected device.

A suggestion service tracks what people watch, both anonymously, and for groups of friends. This is used to compute suggestions based upon statistical techniques driven from past behaviour.

15. Advertising as Social Games

While watching a boring show, Sarah gets asked to join a different channel, where there's a competition to spot how many animals appear in the ad break. She joins the show with her friends and they compete to see who spots the most. At the end she gets her score and some statistics for the game.

A social app could propose that you change channel based upon what your friends are watching. This involves sharing what you watching with the social app provider. A separate app is used for the competition.

16. Social TV Experience

A user can see what his friends are watching or plan to watch – both for on-demand content (so he can watch out for spoilers / have something to talk about at work) and for live content (so he can join and chat).

A social app tracks what people are watching and allows friends to see what each other are viewing, and to share notes on what they plan to watch. The app include support for comments and live chat.

17. Producing Partial Media

BBC offers audio commentary for F1 production. They may sell this as a standalone product even if they do not have the rights to broadcasting the video content. Foreign broadcasters with such rights may want to buy BBC's live audio commentary - because it's good - as can individual viewers across the world.

The BBC's website provides an app that adds the synchronized commentary to the F1 production from another site. Users may need to pay to view the video.

18. Transferring Presentation Between Devices

Peter is watching TV on his phone while commuting, pauses it as he leaves the train, resumes at home on the TV, then transfers it to a laptop when he gets kicked off the TV by his daughter.

This assumes support for time shifting by the broadcaster, as well as a means to discover when the phone is at home and the local TV is available. The laptop could ask the phone or the TV for the stream and time to start it from.

19. Switching Content Backend

Anne is watching a comedy show live over DVB-T. She pauses the stream to go make herself a coffee. Anne resumes playback, which is fetched over the Internet from a live-window. She has to pause again to drive her son Anthony to soccer practice. When she comes back home, she again resumes the show. This is now fetched from the catch-up TV service, but she still does not notice any difference.

This assumes time shifting support that masks the source of the stream, and requires a source of metadata to choose the appropriate source.

20. Custom Extra Media

Two people are watching TV together. They're interested in finding out more about the programme – about a particular plot point. Both can see an extended service on their devices. One continues with the extended information and the other goes to an on-demand programme because he's lost the thread of what's happening in the main story.

The personal devices can run an app that asks the TV for metadata that can be used to search for related content.

21. Audio commentary in different language/different volume level
Umberto watches the film “Cars” with his children on the TV set on the VoD service. The sound system is set to the dubbed Italian version of the film. He opens the browser on his smartphone, logs in and selects the original soundtrack for himself.

The VoD service tracks what you are watching and when you open the service on a second device, offers you complementary content. This use case shows the need for tight synchronization across the two devices involved.

22. On Demand First Class Citzen
Sofia watches a debate on TV, timeshifted and enriched by user generated media (provided by earlier viewers). On her iPad she can see an infographic showing the time distribution of likes and user comments dynamically updated as new data becomes available. She realises that something extraordinary happens about two minutes into the debate, since there is a remarkable spike on the graph. By clicking the graph, both the TV and her smart phone updates to the new position.

The broadcaster’s website offers complementary content and services for its programming on its time shifting service. Users can decide which of these they want to show on each of their devices. This use case benefits from synchronized presentations across the devices.

23. Moving Visual Elements
Jake watches the news, in which a dog has bitten a child. Comments from viewers and readers (of their earlier Web based coverage) are shown scrolling over the screen. Jack wants to have a better view and transfers the comments to his tablet. The transition is smooth and natural, giving him the feeling that the visual element moved (as opposed to just disappearing on one and appearing on another).

Broadcast content is linked to complementary content and services, In this case, Jake’s preferences are set to show users’ comments as an overlay. The UI on the TV allows you to shift selected complementary content and services to a second screen device. Synchronization across devices enables this to be done in a satisfyingly smooth manner. This involves an agent running on the tablet that listens for messages from the TV.

24. Sports
Jack and Jill watch an F1 race. They have the video on the TV and a tablet showing a GPS map of all cars, synchronized to the TV stream. They also have the pulse of each of their favourite driver on their SmartPhone (Jack has Jenson Buttons, Jill has

Fernando Alonsos) playing as an audio track, making for excellent entertainment as Fernando repeatedly fails (then succeeds!) overtaking Jenson.

Broadcast content is linked to complementary content and services. The TV and tablet are both part of an application session, and show selected content/services tightly synchronized across the two devices.

25. Device Suggestions

Alfred registers his SmartTV, his MediaScape loudspeaker, work computer, smart phone and tablet with MediaScape. When applicable, MediaScape enabled services discover the availability of nearby devices and suggest using them. Alfred opens a Quiz on the iPlayer on his tablet, which suggests watching the content on the TV. He chooses to do so, the video is moved to the TV and the Quiz is left on his tablet.

This presumes a cloud based personal service where you can register your devices. Each device can track its whereabouts via the WiFi SSID. The service exposes this as context data to the apps and services provided by the broadcaster. The use case also requires coordinated transfer of content views from one device to another.

26. Conflicting Interests

Both Alfred and Bonnie has registered their SmartTV to their MediaScape profile. Bonnie arrives home while Alfred is watching a quiz using both the TV and his tablet. She uses her smartphone to find the latest news programme and manually selects to have it shown on the TV. In response to this, Alfred’s quiz video must leave the TV and reappears on his tablet together with the Quiz graphics and Bonnie can watch the news.

Each device is running an agent that tracks the context. When Bonnie uses her phone to ask the TV to show the news programme, the TV can message Alfred’s tablet to arrange the transfer of the Quiz session views to the tablet.

27. Local ad-hoc pairing

Charlie and Dave watch an interactive show, and want to compete. They bump their NFC enabled phones to let them join the same session. The TV infographics now display results for both participants. Eleanor joins them, but lacks NFC on her phone. Dave selects “QR” on his device, shows it to Elanor which scans it and joins in.

The broadcaster has created an app for the show that is loaded when you start the show. This encourages you to invite your friends into the show to compete with one another. NFC and QRCodes allow you to invite people in an ad hoc fashion. You could also send an invite via your social network to your registered friends.

28. Tightly Synchronised Complementary Content

Frankie watches the news, and is informed that there is additional content available. His SmartTV is able to determine the position in the stream accurately, allowing his tablet to join the TV and provide a tightly synchronized experience.

Broadcast content is linked to complementary content and services. The TV and tablet are both part of an application session, and show selected content/services tightly synchronized across the two devices.

Appendix 2 - Metadata

This is brief summary of metadata standards for broadcasting and a discussion on what is needed to support the vision described in this paper.

Existing TV Metadata standards and organizations

Disclaimer: this should not be considered as complete or authoritative.

DAB

Digital radio broadcasts embed string data on the current program.

DVB-SI

This is used for basic information as part of broadcast streams, e.g. the current and next programs, and other events that can be used to synchronize applications even if offline.

TV-Anytime

This can be used for personalized electronic program guides (EPG) and content discovery services. Enhancements include support for multiscreen, mashups etc. TV-Anytime is now maintained by the EBU.

MPEG-7

This is an XML based metadata format for MPEG-4 streams.

MPEG-21

This is essentially a packaging format for sets of metadata including digital rights.

SMPTE

The Society of Motion Picture Technical Experts (SMPTE) maintains a dictionary of terms covering the use of metadata for video, audio and multimedia data. The Material Exchange Format (MXF) wraps audio, video, subtitles, and metadata into a single file. The Broadcast Exchange Format (BXF) is used by broadcasters in relation to program management, traffic, automation and content distribution.

ATSC

The Advanced Television Systems Committee (ATSC) coordinates television standards for different communications media, e.g. digital TV, interaction systems and broadband. ATSC's work has been adopted by North and South American countries as well as South Korea.

EBU

The European Broadcasting Union (EBU) is an organization with members from Europe's national broadcasters. It monitors, analyses and assesses technological development. The EBU has a number of metadata projects, see https://tech.ebu.ch/metadata

Additional discussion

W3C's own work on linked data and the Semantic Web seem very promising for the flexibility we need to link to complementary content and services. Schema.org could be relevant when it comes to indexing services created with HTML5, see for example:

- http://blog.schema.org/2013/12/schemaorg-for-tv-and-radio-markup.html

The BBC uses the programmes Ontology

- http://www.bbc.co.uk/ontologies/programmes/2009-09-07.shtml

A starting point is to be able to associate programs with URIs as a basis for making statements about programs as a whole, events within the program, named regions across a sequence of frames and so forth. One paper that explores this is "APIs and URLs for Social TV" from the W3C Web & TV Workshop in November 2010.

- http://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_60.pdf

One challenge is for broadcasters and video on demand companies to expose metadata that can be used by third parties. This is critical to enabling end users to search for content and services across many providers.

Related to metadata are formats that describe coordination across media resources, one example is W3C's Synchronized Multimedia Interaction Language (SMIL), see:

- http://www.w3.org/TR/smil/

The P2P-Next project worked on synchronising timed metadata with live streamed video over the web using lightweight interactive media objects (LIMO) and a modified browser supporting the HTML5 audio and video elements. See:

- http://ec.europa.eu/information_society/apps/projects/logos/7/216217/080/deliverables/001_p2pnextwp5d541d.pdf