



DVB-I Webinar Series

Part 3 of 3

6 April 2020

Introduction

Paul Higgs – TM-I Chair - Huawei

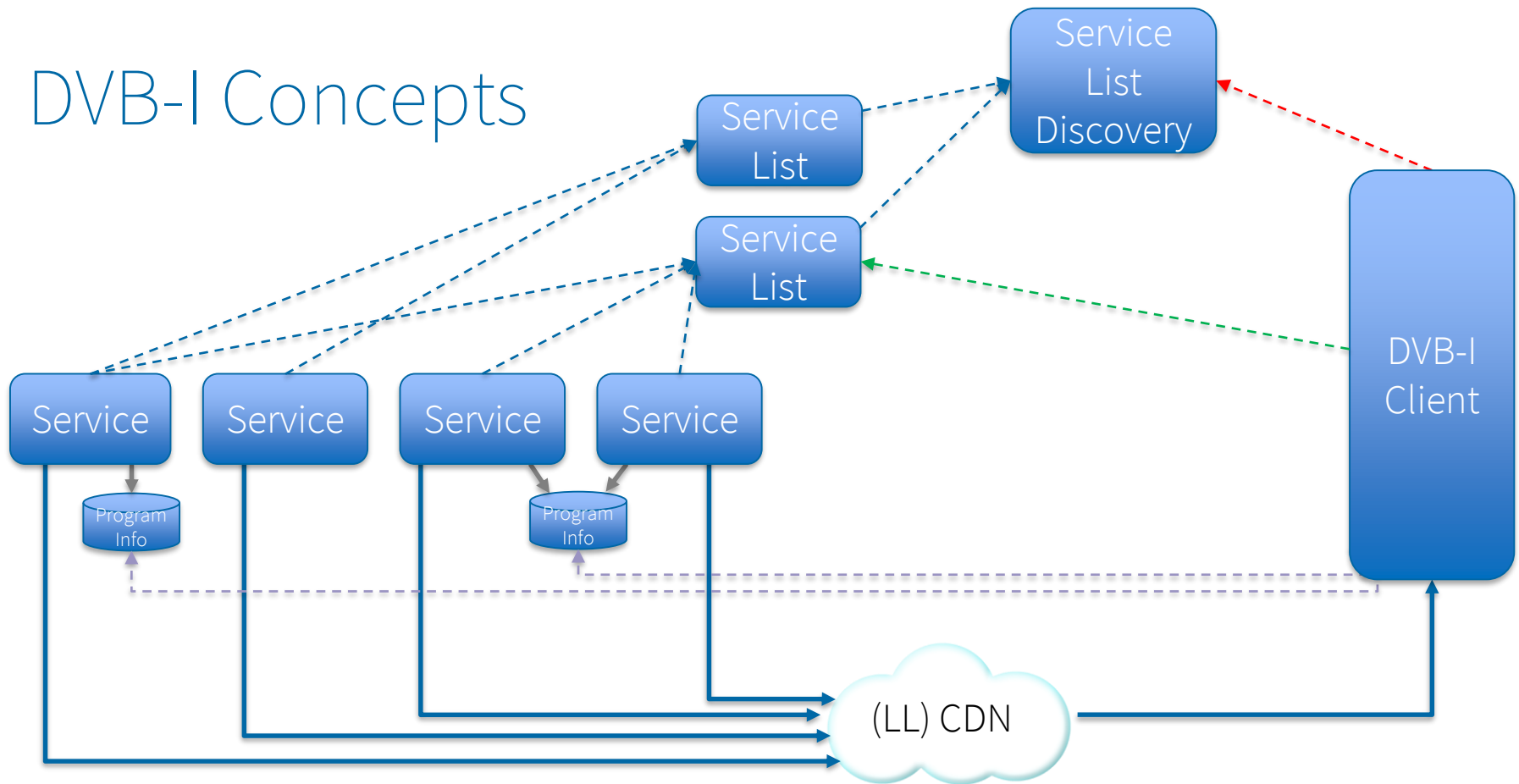
Road to DVB-I

- DVB-C, DVB-T, DVB-S
 - Broadcast television services
- HbbTV
 - Add interactive elements and ondemand content to existing broadcast services
- DVB-IPTV
 - Television services delivered over managed/private IP networks
- DVB-I
 - Broadcast television services delivered over the Internet

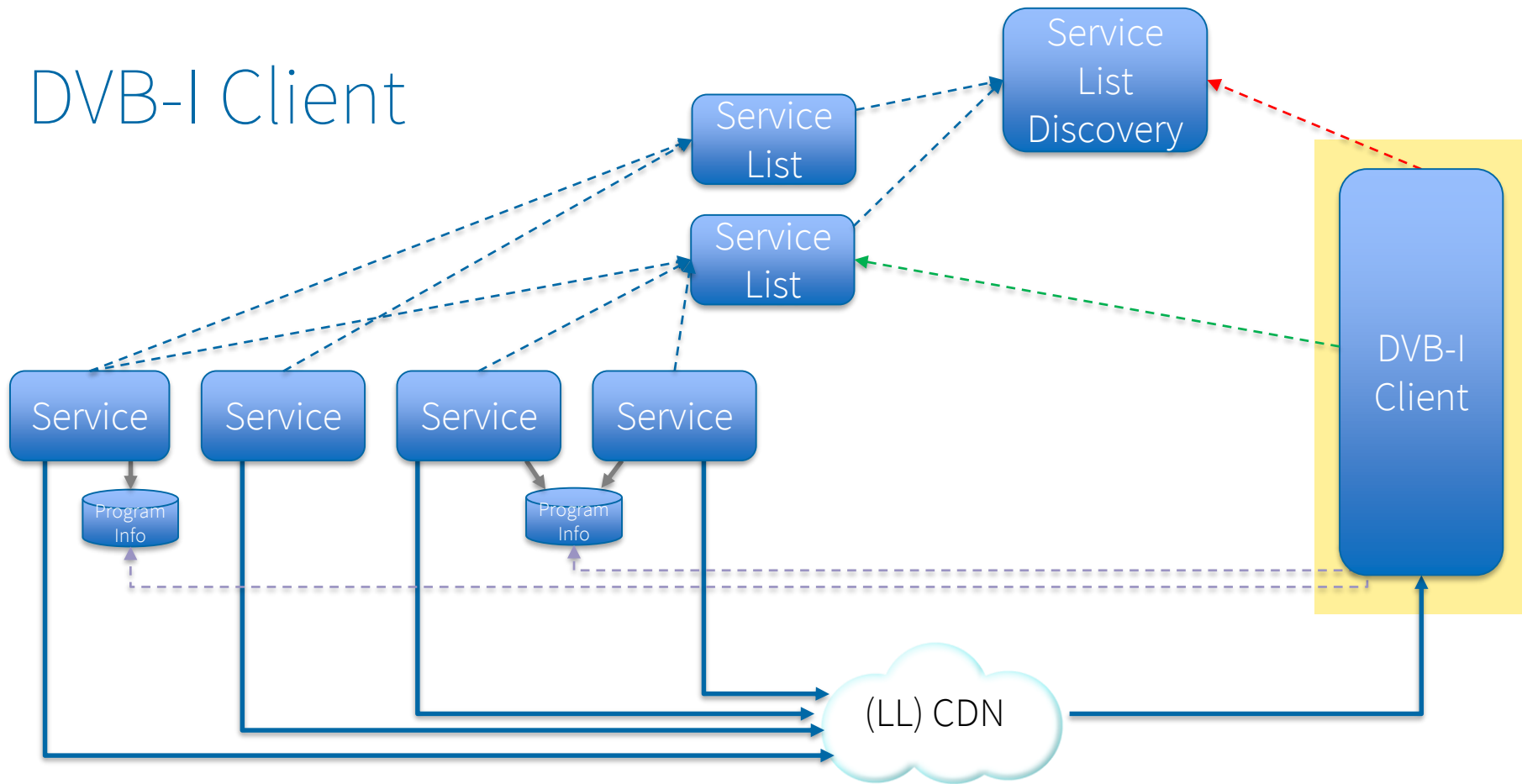
The ecosystem

- A177 – DVB-I Service Discovery and Programme Information
 - The “internet channel scan” and EPG
- A168 – DVB-DASH with Low Latency
 - Media format for linear broadband television services (On Demand also)
- A176 – Multicast ABR (DVB-mABR)
 - Efficient delivery of DVB DASH over IP networks
- A178 – Targeted Advertising (DVB-TA)
 - Personalizing advertisements for television services

DVB-I Concepts



DVB-I Client



Part 3: DVB-I Clients

Juha Joki - Sofia Digital

Tomi Sarparanta – Sofia Digital

Topics

- Background and purpose
- Client architecture
- Client functionality
- HbbTV (Operator Application) and Android implementation
- Backend
- Additional information and lessons learned

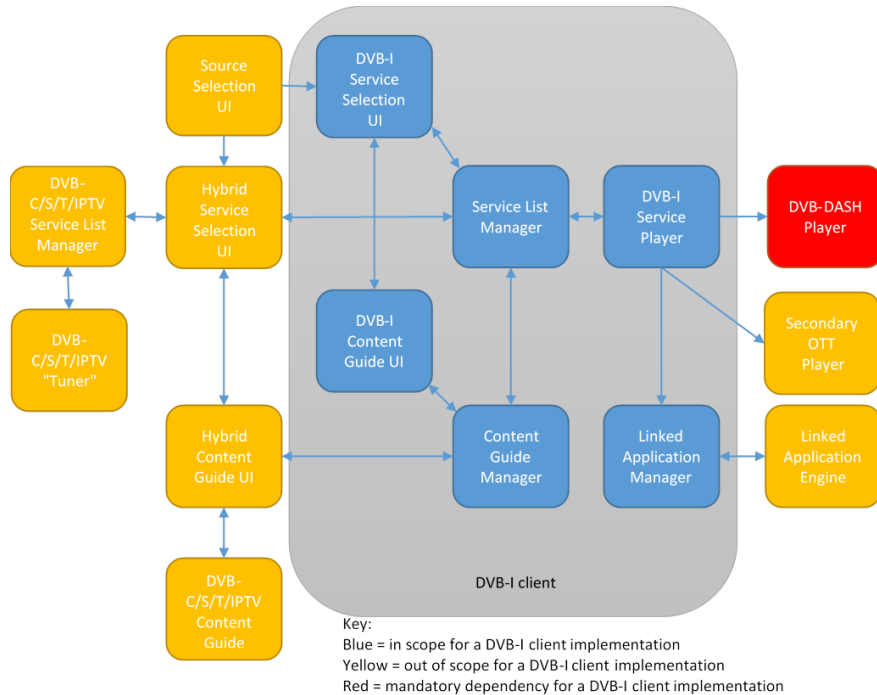
Background and purpose

DVB-I Reference Client Background

- Work started together with DVB in January 2020
- First version was released end of January
- “Phase 1”, a stable demo meant for DVB World was completed early March. Work is now on-going for Phase 2, due to be completed some time in May
- Regular updates since, now at version 2.4
- Android mobiles and tablets (7, 8, 9, 10 – just try it!) and HbbTV OpApp versions officially supported
 - Other possible target platforms could be (but not in scope this time): androidTV, iOS, SmartTV platforms, or anything with DVB DASH + javascript + HTML5
- Includes a backend component to create and edit the necessary metadata
- Publicly hosted in DVB’s GitHub, licenced under MIT license
 - <https://github.com/DVBproject/DVB-I-Reference-Client>
- Easily deployed anywhere with basic web skills
- Based on the DVB-I Spec A177 – Service Discovery and Programme Information



Application scope and purpose



- The DVB-I reference client purpose is to verify the A177 spec (service lists, content metadata, regions, languages, and so on) can be used to build an end-to-end solution
 - “DVB-I is the first major specification to benefit from DVB’s new Verification and Validation activity that aims at accelerating the implementation of its technologies in the marketplace”
- At this point it is *not* exhaustive
- It is a work in progress – as is the specification
- Many things are unfinished or not complete
- Final version will be a *reference client*, not something that is meant to be used as-is, but perhaps providing building blocks and for and confidence in commencing a commercial product or ecosystem

Project development

- Work started with a simple backend component to generate valid DVB-I Service lists and Guide data
- Existing HbbTV OpApp service list browser and DASH player component were adapted to use DVB-I metadata formats
- Finally, Android Progressive Web Application was developed utilizing the dash.js player (<https://github.com/Dash-Industry-Forum/dash.js/wiki>)

Client architecture

DVB-I Reference client architecture

- All Service provider, service list and schedule parsing is implemented in a common codebase used by the HbbTV and Android clients
- Reference client codebase is divided to common, android and hbbtv directories
 - HbbTV implementation is divided to launcher (channel list) and epg directories
 - Android implementation has everything in separate js-files in the js-directory
- XML is parsed to JavaScript objects

DVB-I Reference client architecture

- Main functions of the DVB-I Reference client are divided to three main javascript library files
 - `common.js`
 - Provides basic functions for date parsing, navigation, resource loading, layout helpers, etc.
 - `dvbi-common.js`
 - Provides service list registry query and channel list query functions
 - `channel-common.js`
 - Provides TV-Anytime schedule parsing

Service provider list

- Service provider list URL is implemented inside the `dvbi-common.js`
- Service provider list parsing function is the `parseServiceListProvider` in `dvbi-common.js`
- `GenerateServiceListQuery` in `dvbi-common.js` can be used to generate a filtered service provider query

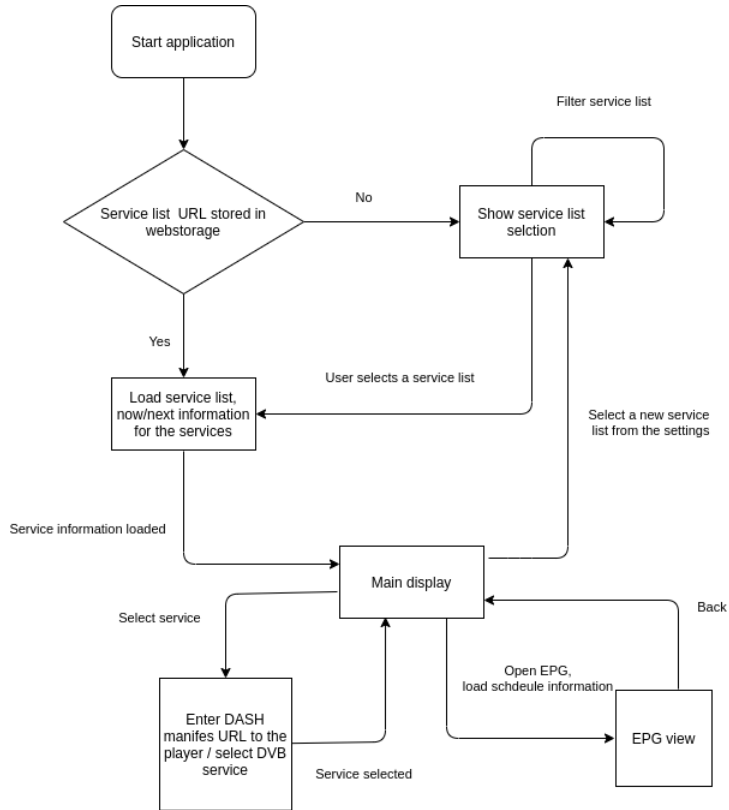
Service List

- Service list parsing is done in `parseServiceList` in `dvbi-common.js`
- `dvbChannels`-parameter in the `parseServiceList` function should be a `HbbTVChannelList` object
- `parseServiceList` will try to match the DVB services in the DVB-I service list with the ones that are found in the device channel list
- Android client only lists services with a DASH source, HbbTV client list services with a DASH source and those with DVB-T/S/C source that are listed in the device channel list
- HbbTV client also lists DVB services that are not in the DVB-I service list
 - assigned a new LCN starting from 1000 ($1000 + n$) and are visually different from the services listed in the DVB-I service list
 - This is purely a design decision – the client can be modified to work differently, to filter out premium services, or to filter out all services not present in the DVB-I service list etc.

Schedule information

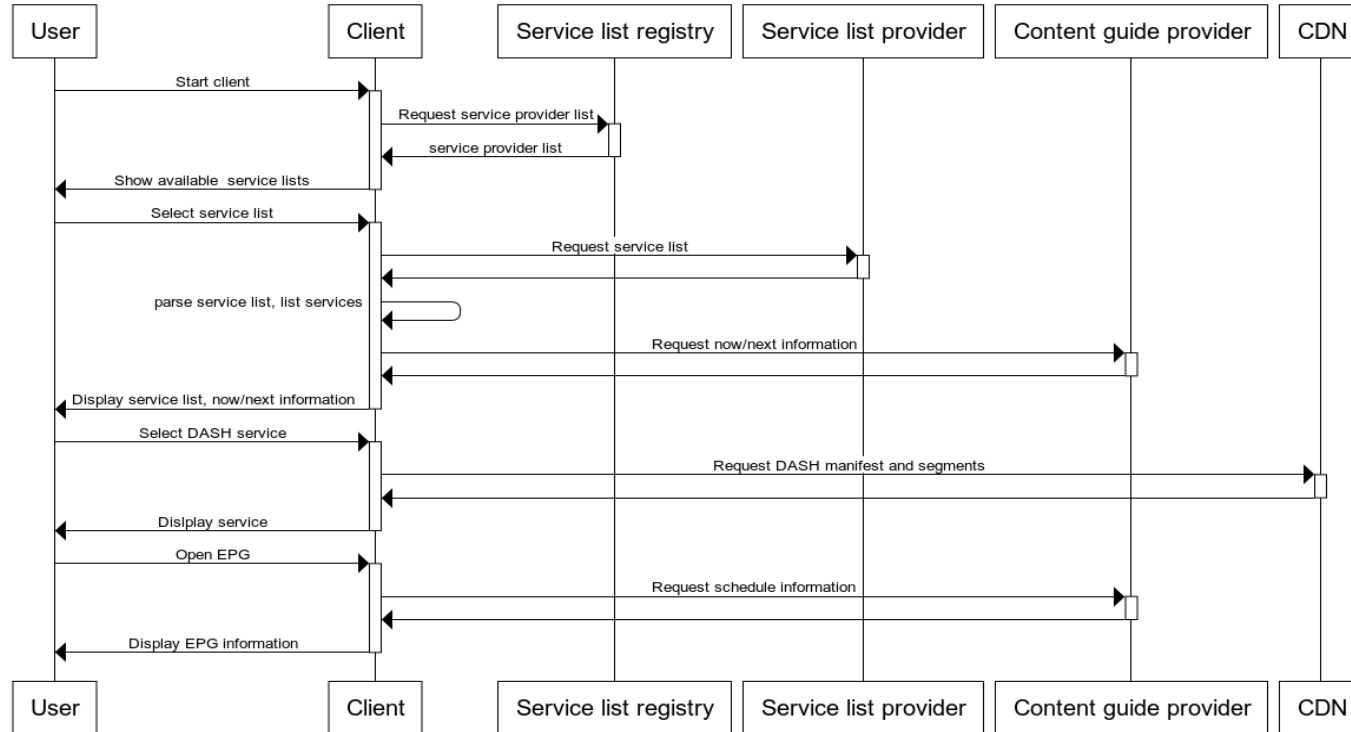
- Schedule information requests are always made for a single service
- Schedule response parsing implementation is found in `parseSchedule` in `channel-common.js`
- Both `now/next` and `schedule` requests use the same response schema
- Current and next program can be requested with the `now_next=true` parameter
- Schedule can be requested with `start` and `end` parameters

Application workflow diagram



1. Application is loaded
2. Checking for pre-selected service list is performed
3. Service list selected/parsed
4. Main display is shown
5. User can select another service list / open EPG / select or tune to a DVB-DASH or broadcast service

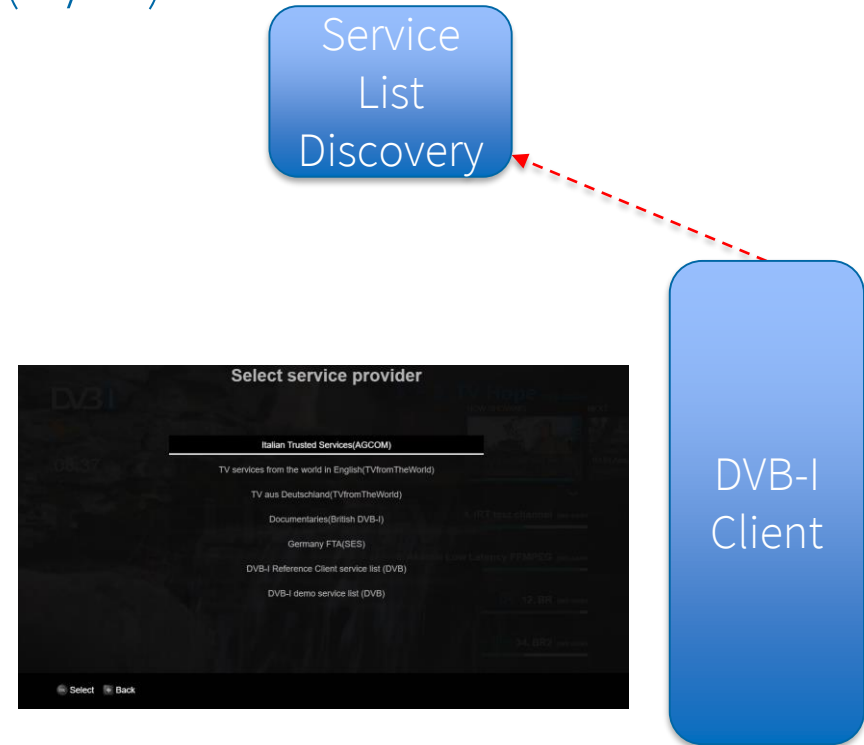
Sequence example



Client functionality

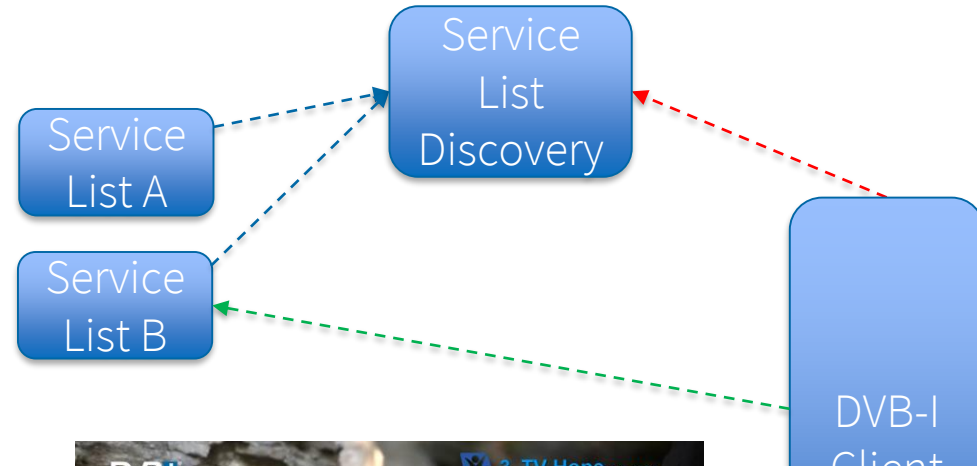
Client basic functions (1/3)

- Service list browser
 - Accesses the Service list registry server
 - A JS function travels through a pre-defined master service list
 - Filters: regulatorlist, providername, genre, target country, language
 - If there's not a previous service list selected, this screen is the initial view of the operational application. From this screen the service list is selected
 - The selected service list is stored in local (web)storage or as a fallback in a cookie



Client basic functions (2/3)

- Service browser
 - Parses through the service list selected in the service list browser
 - In Android, only DASH-services of the selected service list are listed
 - In HbbTV, three types of services are listed:
 - DVB-C/S/T services installed in the TV, and listed in the DVB-I Service list
 - DVB DASH only services
 - DVB-C/S/T services installed in the TV but not listed in the DVB-I service list
 - Allows the selections (tuning/starting) the DVB / Live DVB DASH services
 - Both Android and HbbTV versions use the same service list registry and service lists
 - Service list includes Low Latency sources as well from related projects around DVB-I (ffmpeg/dashll)



Client basic functions (3/3)

- EPG
 - Content is based on the TV-Anytime specification
 - Broadcast and On-line Services Search, select and rightful use of content (“TV-Anytime”); Part3:Metadata; Sub-part 1: Phase 1 –Metadata schemas, ETSI TS 102 822-3-1 v1.11.2 (2019-06)
 - In HbbTV there’s a Grid-based traditional EPG, accessible with the green colour key
 - In Android there’s a list-based EPG, opened via a user interface element
- Info banner
 - In HbbTV, the info banner is a traditional info banner displaying “Now/Next” programming information from the TV-Anytime
 - In Android, the user interface overlay includes info banner information as well

HbbTV Operator app vs. Android app

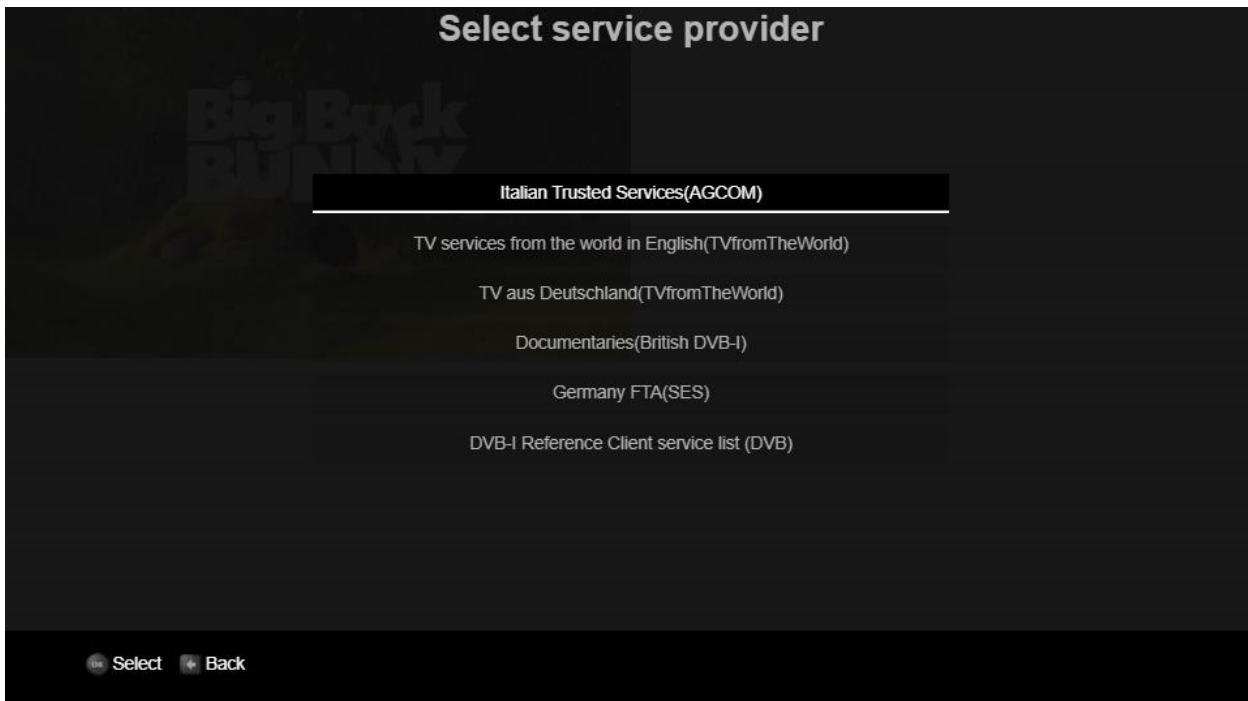
HbbTV OpApp DVB-I Client

- Utilizes the Operator App APIs (Privileged)
 - Enables registering additional key events like P+/P-, INFO and GUIDE for Operator application use
 - Enables replacing some of the normal TV User interface
 - Additional channel list management
 - Many OpApp functionalities require agreement with the receiver manufacturer
- Info banner / Channel list browser works in “transient” and background modes
 - Transient mode has OpApp visible for a limited time, fullscreen video is available, can be overlaid with native UI components
 - Background mode has the OpApp in invisible mode, listening for key events
- EPG works in the “foreground” mode
 - Works in full-screen mode, broadcast video available only heavily scaled
- Works also as a “normal” broadcast HbbTV v2 application, utilizing only normal key events like the color keys
 - Transport stream is provided for effortless testing
- Navigation is always done with remote control key events
- HTML5 video object –based player
 - App relies on the native DASH player on the TV – some limitations are bound to happen (i.e. the latest advancements from dash.js player are not available)

HbbTV OpApp Client

- What is an OpApp?
 - Independent specification complementing HbbTV 2.0.1
 - Based on HTML5, CSS, Javascript
 - Accessed via selecting the OpApp like an HDMI input or a similar familiar access point and offers similar user controls like in a set-top box, including:
 - Channel zapping (+/-)
 - Volume control (up/down)
 - DRM and Adaptive Bitrate Streaming (ABR)
 - UHD, HEVC, Next-gen audio, companion screens, etc.
 - Control of the broadcaster's HbbTV applications
- In short: takes over the regular manufacturer TV UI/UX
 - Example functionalities/ features
 - Live TV, both DVB and OTT
 - Interactive Program Guide
 - Operator defined channel lists
 - Mini TV-guide (i.e. overlay)
 - Catchup / VOD
- OpApp Service discovery is based on country and region selection during First Time Installation
- More information: <https://www.hbbtv.org/wp-content/uploads/2017/12/HbbTV-SPEC-00200-003-Operator-Applications-Explained.pdf>

HbbTV Service list provider selection



HbbTV Service list browser

DVB-I
sofia digital
11:20
Big Buck BUNNY

1000.test

3. TV Hope DVB-DASH

4. IRT test channel DVB-DASH

8. Akamai Low Latency FFmpeg DVB-DASH

11. DVB-T channel 2 DVB-T

NOW SHOWING

09:00 London
40 mins remaining

NEXT

12:00 Like F...
Duration 60 mins

12. BR DVB-DASH


34. BR2 DVB-DASH

HbbTV EPG

DVB-I 08:39

London
09:00 - 12:00

Leslie travels to London to accept an award; Tom hunts down a businessman; Ann and Chris take their relationship to the next level.



TODAY, 6/4 ◀ 9:30 10:00 10:30 11:00 11:30 12:00 ▶

IRT test channel	SportsNation		NASCAR Race Hub	
Akamai	Airstream Dream	Jungles	Wild About Animals	Cities
DVB-T channel 2	London			Like Father, Like Sun
BR	London			Like Father, Like Sun
BR HD	SportsNation		NASCAR Race Hub	

Select -5 +5 First Last Tune to channel

Android client

- PWA-style implementation
 - Progressive Web App => offers a simple “installation” / Add to home screen functionality via a manifest file
 - More information: https://en.wikipedia.org/wiki/Progressive_web_application
- Lists the DVB DASH instances of services only
- Should work on any Android client (Android 7, 8, 9 and 10 tested)
- EPG is list-based instead of grid
- Landscape mode only as an PWA app
- dash.js player implementation
 - Uses the latest stable version (<https://cdn.dashjs.org/latest/dash.all.min.js>)
 - Low latency settings included

Android Service List provider selection

Filter

Provider

Language

Genre

Country

Regulator list

Multiple values can be separated by comma

Select service list

AGCOM
[Italian Trusted Services](#)

TVfromTheWorld
[TV services from the world in English](#)
[TV aus Deutschland](#)

British DVB-I
[Documentaries](#)

SES
[Germany FTA](#)

DVB
[DVB-I Reference Client service list](#)
[DVB-I demo service list](#)

Android main screen with settings screens visible

DVB-I
sofiadigital
Powering Smart Screens

Settings

- Low latency mode

Target latency:

Minimum drift (seconds):

Catch-up playback rate (%):

Select servicelist Stream Info Close

Video resolution: 1280x720 (max:1280x720)
Video bitrate: 2000kbits (max:2000kbits)
Audio bitrate: 192kbits (max:192kbits)
Latency: 8.239s
Settings:
Low latency mode:false Delay:3
Min drift:0.05 Catchup Rate0.5

3 TV Hope
4 IRT test channel
8 Akamai Low Latency FFmpeg
12 BR
34 BR2
324 DASH CBR
640 LL-DASH(number)
641 LL-DASH(time)
642 EyeQ
643 EyeQ3
644 EyeQ4
645 EyeQ5

BR
12. BR
Now: London 49 mins remaining
Next: Like Father, Like Sun 12:00 Duration 60 mins
Open EPG Settings

Android EPG

The screenshot displays an Android EPG application interface with a dark theme. At the top, there is a navigation bar containing a back arrow, the text 'EPG', the version number '6.4', a close icon (X), and a forward arrow. Below the navigation bar, the interface is divided into three columns, each representing a different channel. Each channel has a header with its name and a list of programs with their start times. The first column is for '3 TV Hope', the second for '4 IRT test channel', and the third for '8 Akamai Low Latency FFMPEG'. The '4 IRT test channel' column has a blue highlight on the '07:00 SportsNation' program. The '8 Akamai Low Latency FFMPEG' column has a blue highlight on the '03:00 Alone' program. The Akamai logo is visible in the top right corner of the interface.

Channel	Program	Start Time
3 TV Hope	Animal Invaders	00:30
	Face Ripper	01:30
	Mountains	04:01
	Jason Peterson's Into The Wild	05:01
	Alone	09:01
	Jungles	12:00
	We Bought a Zoo	13:30
	Airstream Dream	15:00
	Funnymals	16:00
	Animal Invaders	16:30
	The Pride of the Eastern Cape	17:30
	We Bought a Zoo	18:00
	Airstream Dream	20:30
	Cities	21:30
	Jungles	22:30
Mountains	23:30	
4 IRT test channel	Mobil 1 The Grid	01:00
	Sedona Tire RipSaw Ridge	01:30
	SportsNation	07:00
	NASCAR Race Hub	11:00
	TMZ Sports	12:30
	Sports Scene	13:00
	The Fantasy Show	13:30
	REAL Sports With Bryant Gumbel	14:00
	SportsNation	15:00
	Top Ten	16:00
	BYU Sports Nation	16:30
	SportsCenter With Scott Van Pelt	17:30
	Sports Scene	18:30
	Top Ten	19:00
	Pac-12 Final Score	19:30
SportsCenter	20:30	
SportsNation	21:00	
Outside the Lines	22:00	
SportsCenter With Michael and Jemele	23:00	
8 Akamai Low Latency FFMPEG	Face Ripper	01:00
	Alone	03:00
	Airstream Dream	09:00
	Jungles	10:00
	Wild About Animals	11:00
	Cities	11:30
	Jason Peterson's Into The Wild	12:30
	Jungles	13:00
	We Bought a Zoo	14:00
	Scaly Adventures	16:30
	Mountains	17:00
	Animales Maravillosos	18:00
	Episode 3	19:00
	The Pride of the Eastern Cape	20:00
	Wild About Animals	20:30
We Bought a Zoo	21:00	
Funnymals	23:30	

Backend

Backend

- PHP/JS application running on a web server
- Service list registry, service lists and programme data (TV-A) are hosted in GitHub instance and browsable in the Sofia Digital Backend server
- Example.xml is the “hard-coded/write-protected” demo service list which can be investigated and copied
- Service list creation and exporting is possible
- TV-A data is hardcoded dummy data at the moment
 - Some sample data has been imported during the project, but proper TV-Anytime / EPG sources are not part of this project
- Backend can be easily installed to any web server with PHP and Javascript support

Backend UI

DVB-I Reference Application sofiadigital
Powering Smart Screens

Edit Service list
Subtitle

Service list version: 1
Service list name: DVB-I example servic
Service list provider: DVB
LCN List Target Region: Placeholder
Content Guide ID: cgid-1
Content Guide Schedule Endpoint: https://stage.sofiadig
Content Guide Provider: DVB-I Reference App

Service name: DVB-T channel 1
Service Unique Identifier: sid1
Service version: 1
Service provider: DVB
LCN: 647
Content Guide Service Reference: cgid_1
Service logo URI

Save servicelist
Service list title

Generate XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceList version="11"  
xmlns="urn:dvb:metadata:service:discovery:2019"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema
```

Servicelist
Subtitle

- example-webinar.xml
- example.xml
- example2.xml
- list1.xml
- list2.xml
- simple.xml

Documentation
[DVB-I Specification](#)

- XML can be easily generated and exported to external apps / websites
- Service list registry editor in progress

Live demos

- HbbTV: <https://stage.sofiadigital.fi/dvb/dvb-i-reference-application/frontend/hbbtv/launcher/>
- Android: <https://stage.sofiadigital.fi/dvb/dvb-i-reference-application/frontend/android/player.html>
- Backend: <https://stage.sofiadigital.fi/dvb/dvb-i-reference-application/backend/>

Additional information and lessons
learned

GitHub repository

- All code is published under the DVB GitHub account's DVB-I Reference Client repository
- Everything made for this project is licensed under MIT
 - Other components used have their own license models, obviously
 - These mean jquery and dash.js
- Forking, commenting and contributing is welcomed
- Live demo: <https://github.com/DVBproject/DVB-I-Reference-Client>

Phase 2 work still going on, in the order of appearance

- Parental controls
- Subtitles
- Audio switching
- Language options / localization / regionalization
- dash.js option / Low Latency settings for HbbTV as well
- Series signalling
- Service availability
- SD/HD/UHD switching
- Linked application for service playback
- Live Pause
- DRM option
- Other UX enhancements

Lessons learned

- DVB-I is not rocket science, but can be implemented with well-known internet and TV technologies
 - HTML5 and DASH are mature technologies working fairly well in recently released consumer devices
- Specification is easy to follow and understand
 - TV-Anytime is a complex beast, though
- Building apps against it has proven to be a useful exercise in building an app working in HbbTV and Android
 - DVB-I has proven to work with a lot of devices
- Lots of interest from all around the world

Links

- GitHub repository: <https://github.com/DVBproject/DVB-I-Reference-Client>
- HbbTV Frontend <https://stage.sofiadigital.fi/dvb/dvb-i-reference-application/frontend/hbbtv/launcher/>
- Android PWA installer <https://stage.sofiadigital.fi/dvb/dvb-i-reference-application/frontend/android/>
- Android Player <https://stage.sofiadigital.fi/dvb/dvb-i-reference-application/frontend/android/player.html>
- Backend for Service list generation <https://stage.sofiadigital.fi/dvb/dvb-i-reference-application/backend/>
- TS file download <https://cloud.sofiadigital.fi/index.php/s/qrB6MoFH5cPCDoN>
- Specification <https://dvb.org/?standard=service-discovery-and-programme-metadata-for-dvb-i-services>
- Sofia Digital www.sofiadigital.com

End