HbbTV Test Applications Webinar

Mark Riley & Alex Russell

1st July 2025

Freely

Secondary Video Window OpApp

Mark Riley

1st July 2025

Introduction

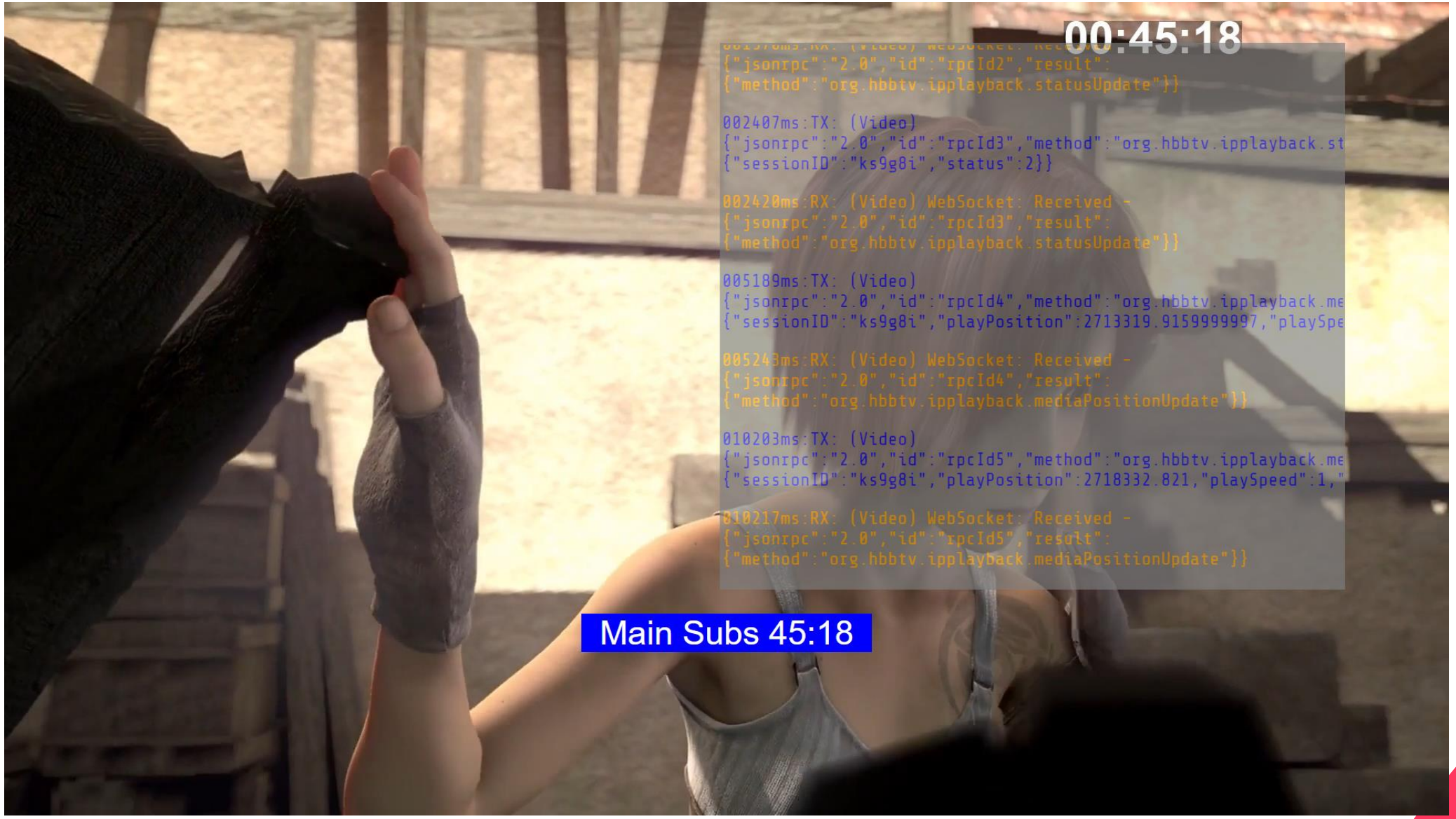
This simple opapp is intended to help the manufacturer commence initial development.

- First created during spec development of feature – to bootstrap manufacturer implementation
- Prototyping / PoC
- Exercising New Features
- Dev and Debugging

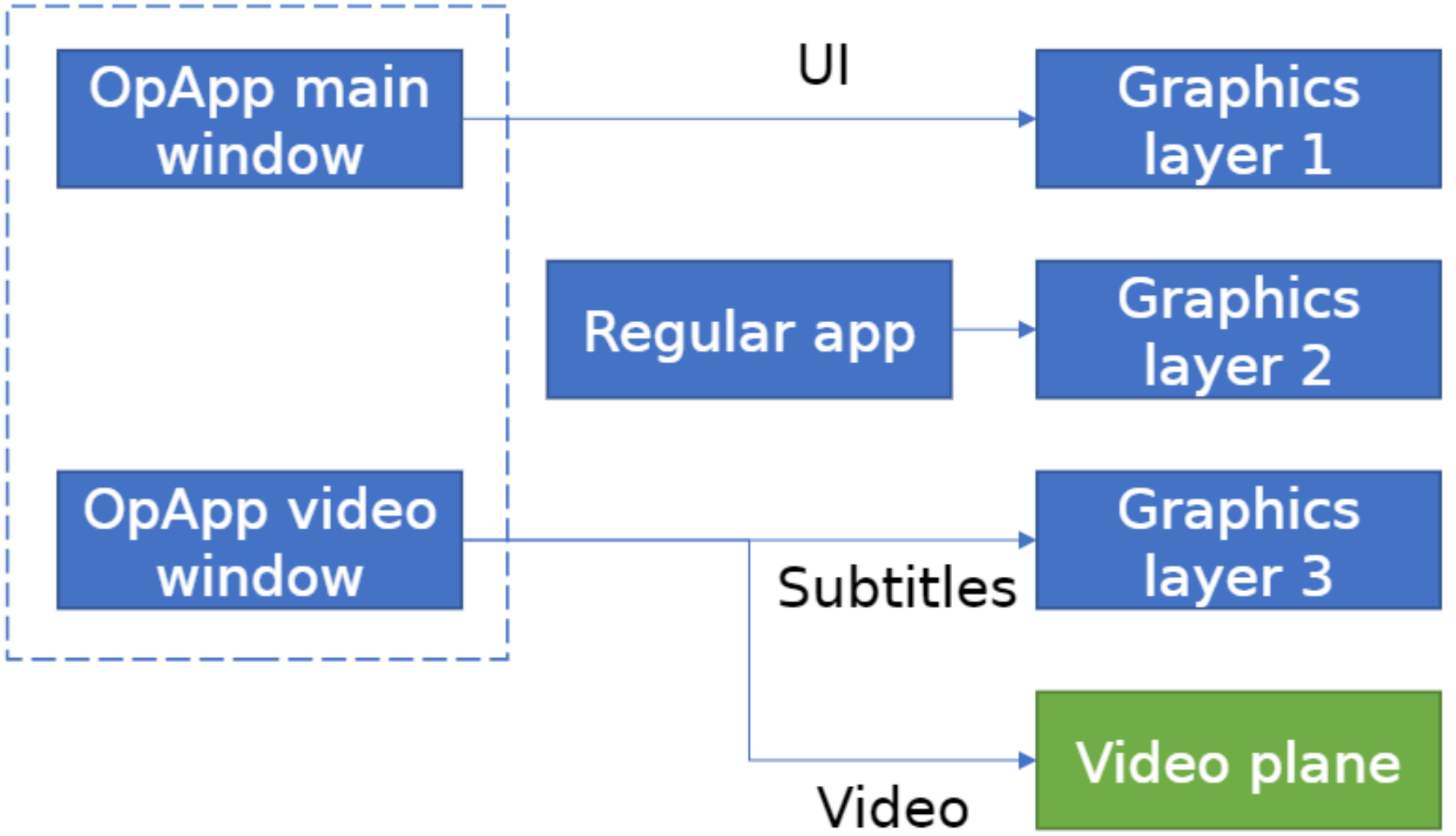
Main Window (index.html)

```
ApplicationManager Object. Error: Cannot read properties of
undefined (reading 'isObjectSupported')
000295ms:WARN: (hbbtvOpApp)Exception when getting the owner
Application object. Error: Cannot read properties of null
(reading 'getOwnerApplication')
000295ms:WARN: (hbbtvOpApp)Exception when creating creating
oipfConfiguration Object. Error: oipfObjectFactory is not
defined
000296ms:WARN: Exception when creating creating
oipfCapabilities Object. Error: oipfObjectFactory is not
defined
000301ms:WARN: DRM: DRM: application/oipfDrmAgent is NOT
supported
000302ms:WARN: DRM: setActiveDRM: not supported
000310ms:Elapsed: OpApp loaded (v4.33)
000607ms:RX: (Video) Child msg:
{"jsonrpc":"2.0","id":"rpcId1","method":"childInitialised"}
000608ms:TX: (Video)
{"jsonrpc":"2.0","id":"rpcId1","method":"SetupVideo","params":
{"hbbtvCfg":"null","jsonRPCServerURL":""}}
000842ms:RX: (Video) Child msg: {"id":"3db4577a-a038-4662-b557-
ad42955bb8b5:1","result":[null]}
001152ms:WARN: (hbbtvOpApp)No Broadcast Channel List present!
002371ms:TX: (Video)
{"jsonrpc":"2.0","id":"rpcId2","method":"Play","params":
{"src":"https://refplayer-
dev.cloud.digitaluk.co.uk/dynamic/ngp-ssai.mpd"}}
```

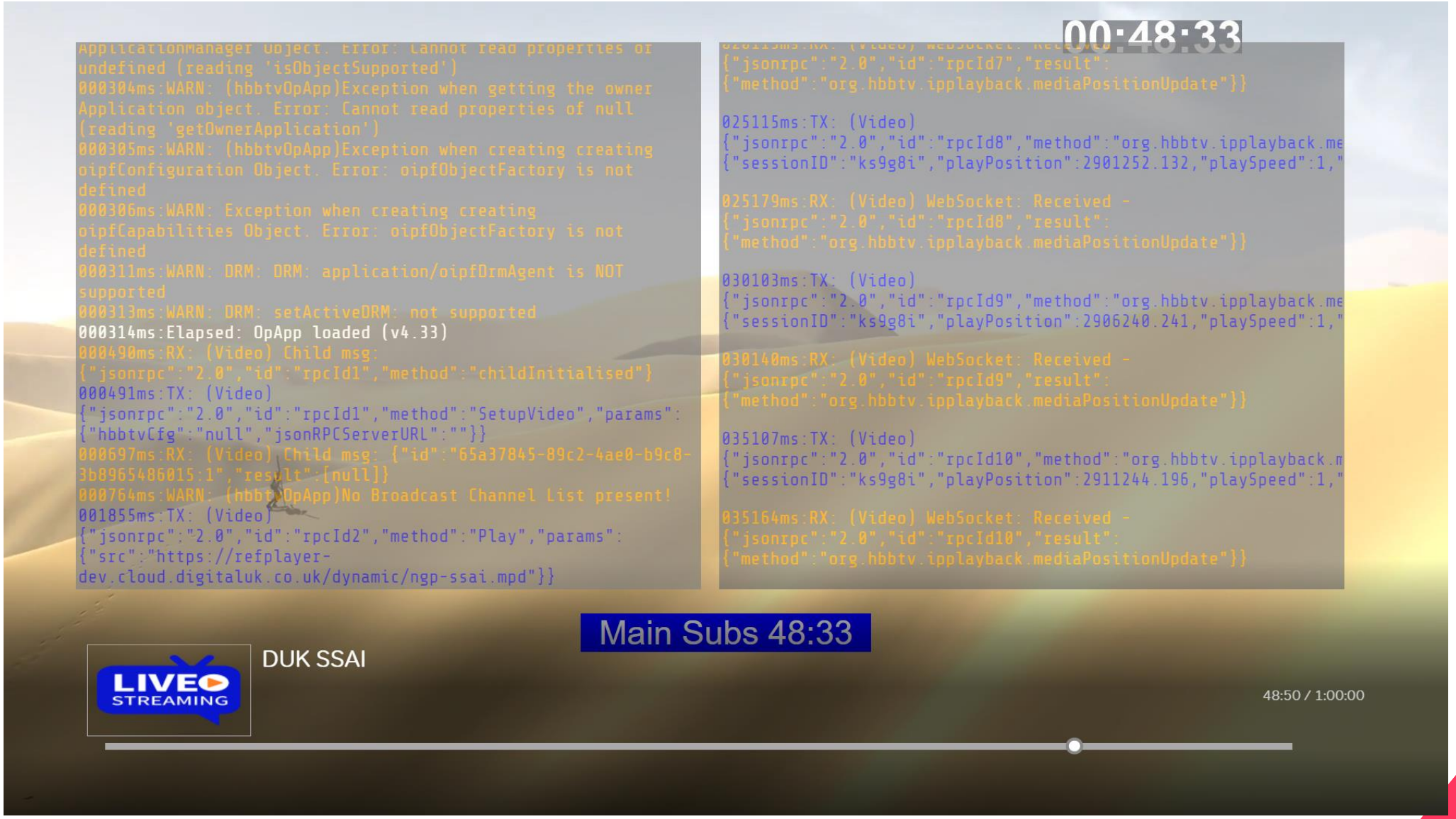
Video Window (video.html)



Plane / Layer Ordering



Composited Layers



Playback

- Playback via video window
- Plays DASH content
- DRM
  - SL2000 / SL3000
  - Temporary / Persistent licences
- MSE / EME API
- Uses dash.js v4.7.4 (and v5.x currently being tested)

JSON RPC

***OpApp** to **Terminal** calls:*

- `org.hbbtv.negotiateMethods`
- `org.hbbtv.ipplayback.statusUpdate` *(where: 2 presenting, 3 stopped)*
- `org.hbbtv.ipplayback.mediaPositionUpdate`
- `org.hbbtv.ipplayback.setComponents`

***Terminal** to **OpApp** calls:*

- `org.hbbtv.ipplayer.selectChannel`
- `org.hbbtv.ipplayer.play`
- `org.hbbtv.ipplayer.stop`
- `org.hbbtv.ipplayer.pause`
- `org.hbbtv.ipplayer.resume`
- `org.hbbtv.ipplayer.setVideoWindow`
- `org.hbbtv.ipplayer.seek`
- `org.hbbtv.ipplayer.selectComponents`

Socket

The WebSocket properties are exposed via the **application/oipfCapabilities** xml doc.

The app checks for the following element:

```
<json_rpc_server url="ws-url" version="spec-version" opapp_version="spec-version"/>
```

```
videoWindow = window.open(window.globalURL + "video.html", "_opappvideo");
```

Note the second parameter `_opappvideo` (this is how the terminal knows that this is the OpApp Video Window)

The App creates a WebSocket, using the standard JavaScript API, eg:

```
__webSocket = new WebSocket(cfg.jsonRPCServerURL);
```

Negotiation

Example of list of methods required by the OpApp video window

```
"method": "org.hbbtv.negotiateMethods",
"params": {
    "appToTerminal": [
        "org.hbbtv.ipplayback.statusUpdate",
        "org.hbbtv.ipplayback.mediaPositionUpdate",
        "org.hbbtv.ipplayback.setComponents"
    ],
    "terminalToApp": [
        "org.hbbtv.ipplayer.selectChannel",
        "org.hbbtv.ipplayer.play",
        "org.hbbtv.ipplayer.stop",
        "org.hbbtv.ipplayer.pause",
        "org.hbbtv.ipplayer.resume",
        "org.hbbtv.ipplayer.seek",
        "org.hbbtv.ipplayer.setRelativeVolume",
        "org.hbbtv.ipplayer.setVideoWindow",
        "org.hbbtv.ipplayer.selectComponents"
    ]
```

Select Channel

Some example messages...

```
"method": "org.hbbtv.ipplayer.selectChannel",
"params": {
    "channelType": 0,
    "idType": 41,
    "ipBroadcastID": "dvb://233a..d04f"
},
```

Terminal >>> Video Window

Video Window >>> Terminal

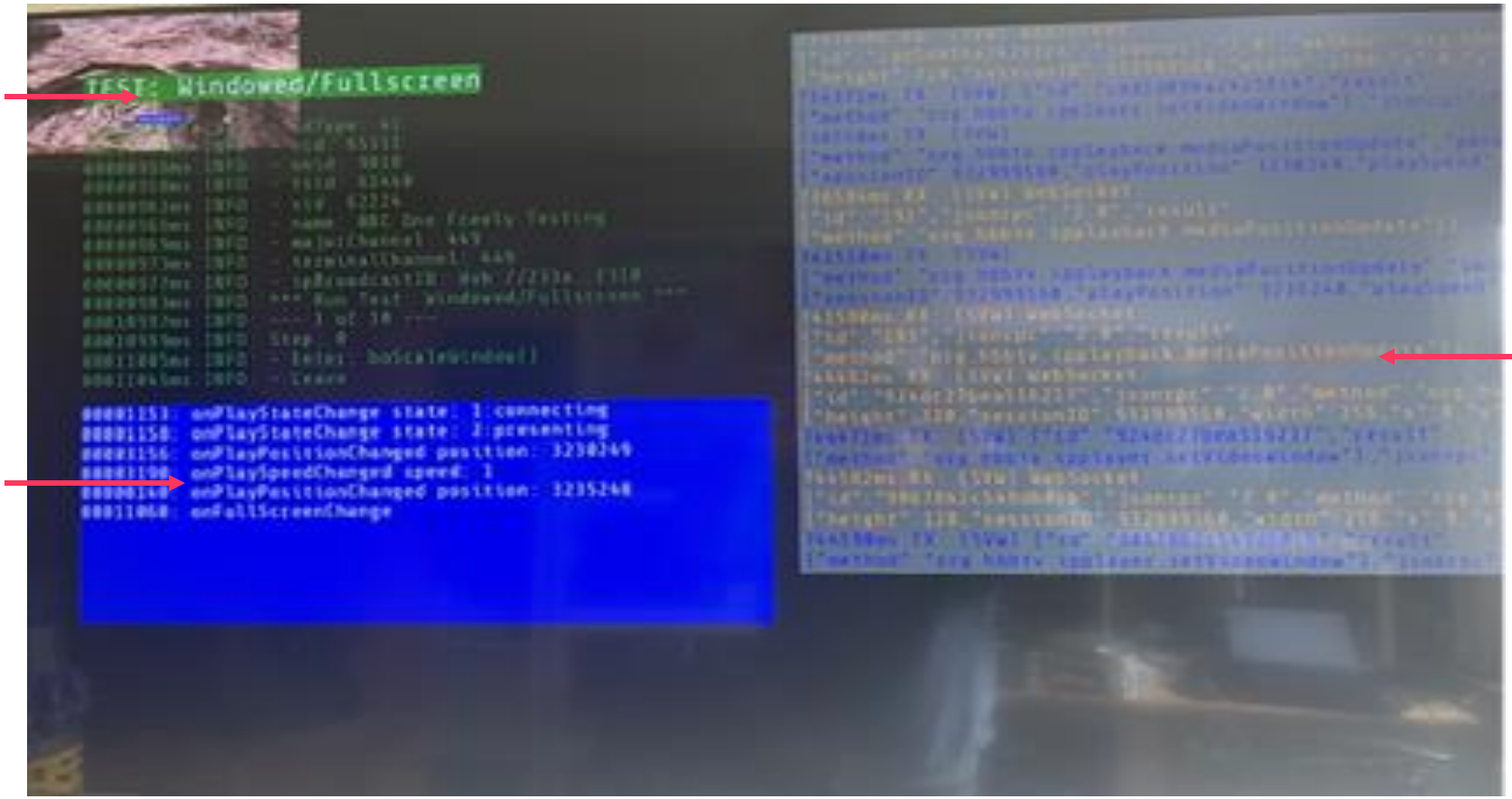
```
"method": "org.hbbtv.ipplayback.statusUpdate",
"params": {
    "sessionID": 1513807178,
    "status": 2
},
```

Broadcast Related App Coexistence

OpApp + Broad Related App

Scaled Secondary Video Window

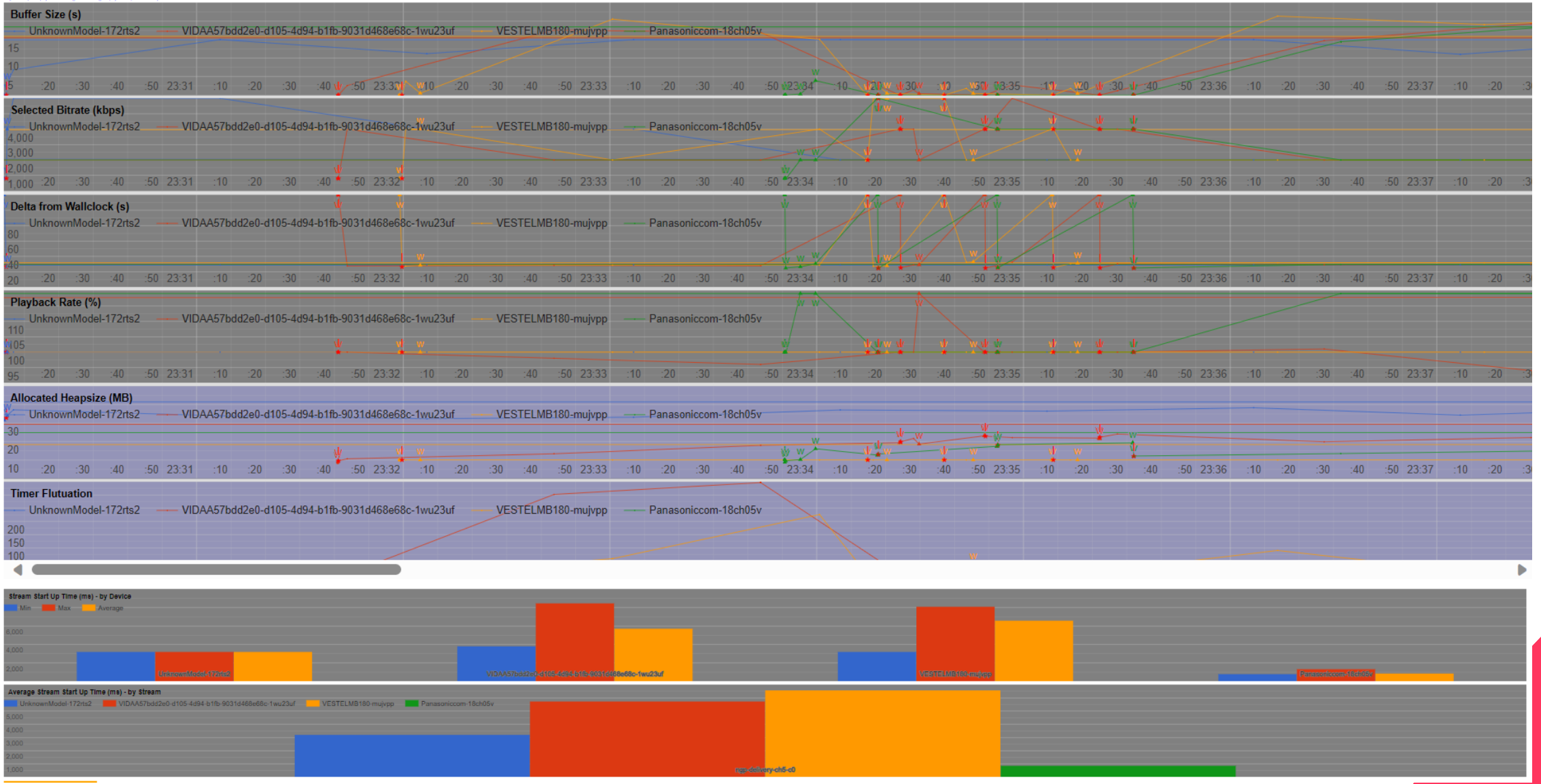
Broadcast Related App



Video window debug

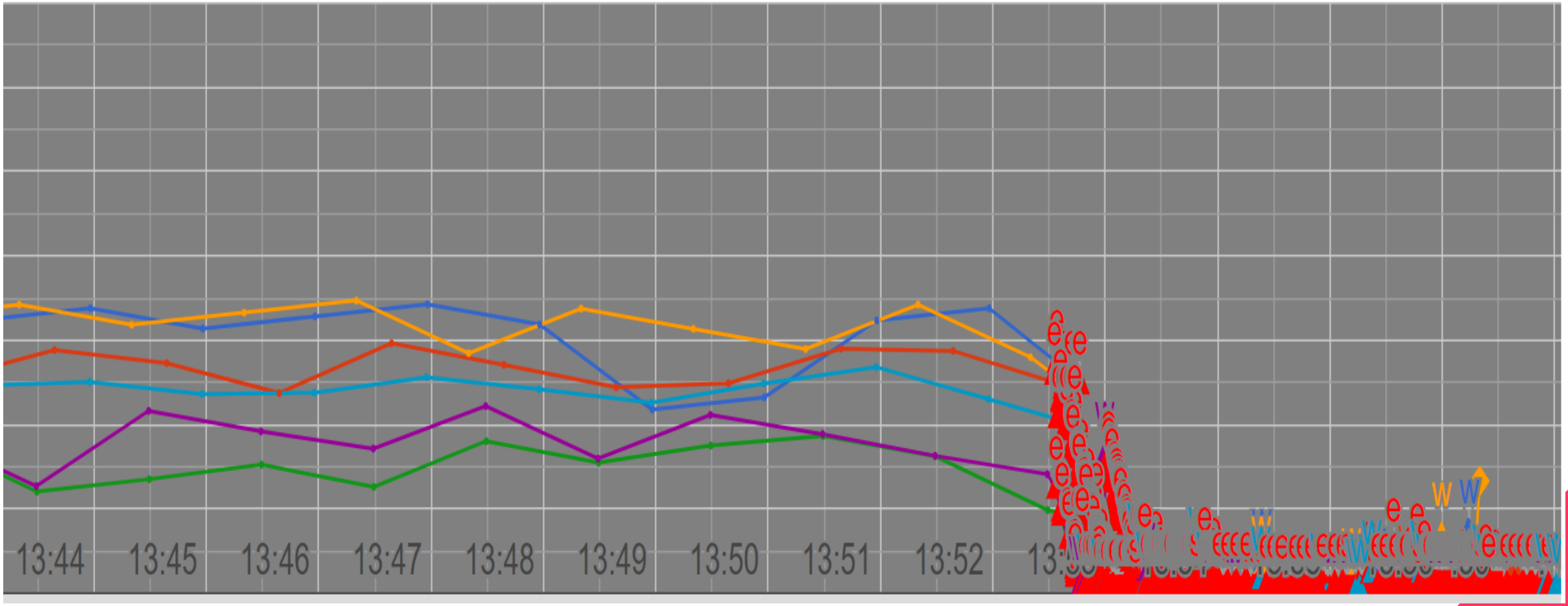
OpApp Main Window in background state

Playback Metrics



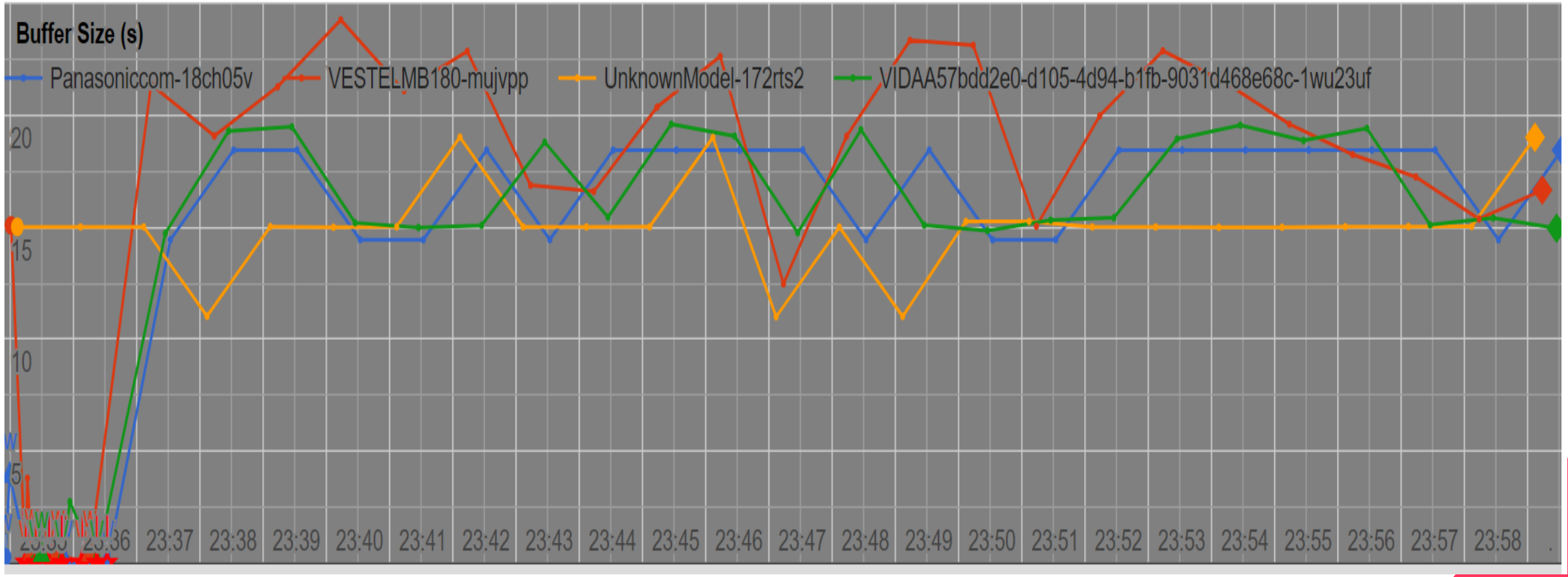
Capturing Fundamental Stream Issues

Eg: Incorrect CORs headers for Ads



Capturing Buffering Levels

Buffering levels over time – for a contended network (5 devices, capped at 16mbps)



Freely

Test and Conformance OpApp

Alex Russell

1st July 2025

API testing with an OpApp

Freely

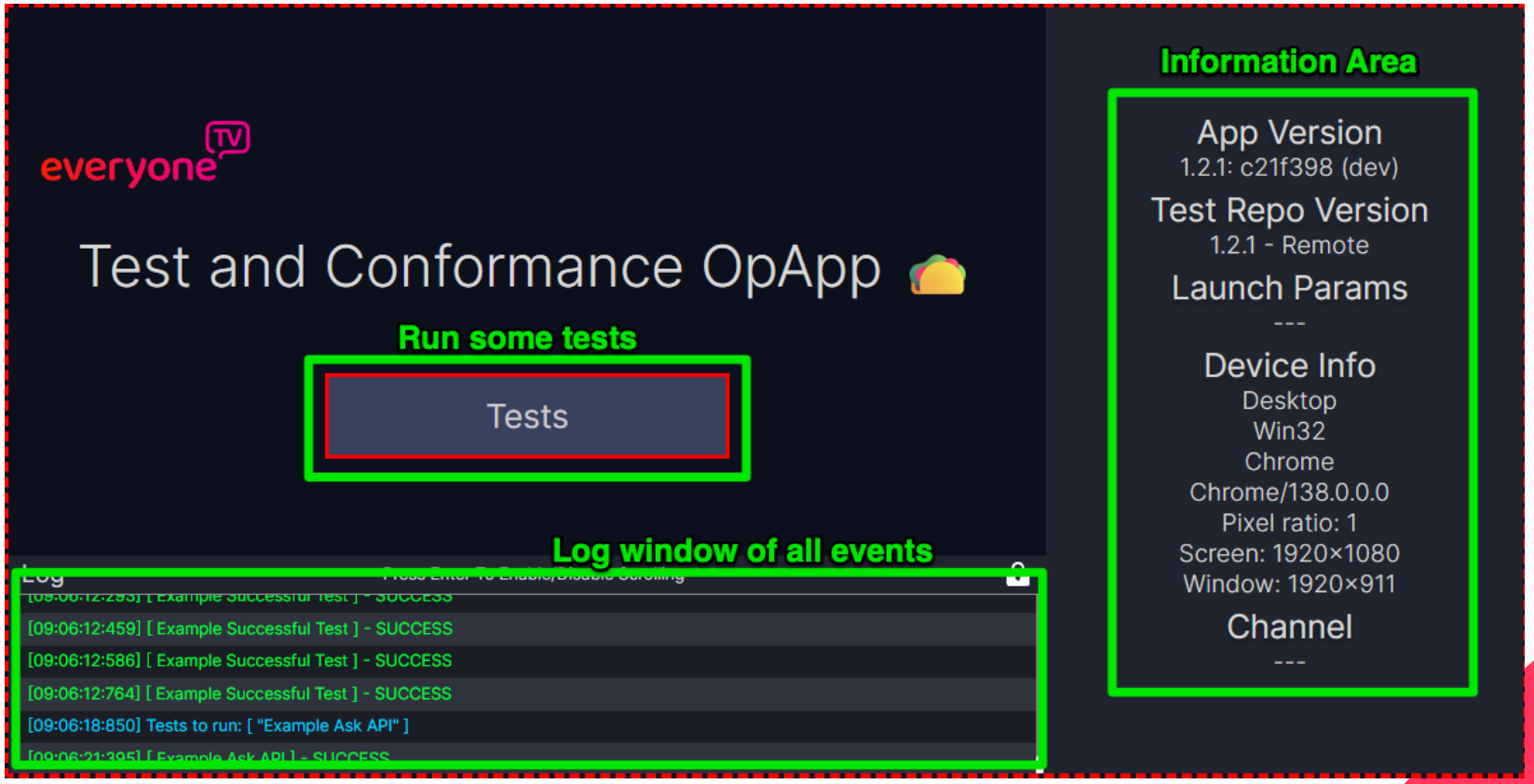
ETV also developed the Test and Conformance OpApp - TaCO

- Focusses on validating the API integration between an OpApp and the terminal
- Does not attempt to demonstrate video playback (as far as possible) as SVW OpApp provides this testing
- TaCO was created in two parts:
  - A reference OpApp with a set of features implemented
  - A set of tests downloaded at run-time using these features

TaCO launch screen

TaCO is designed to:

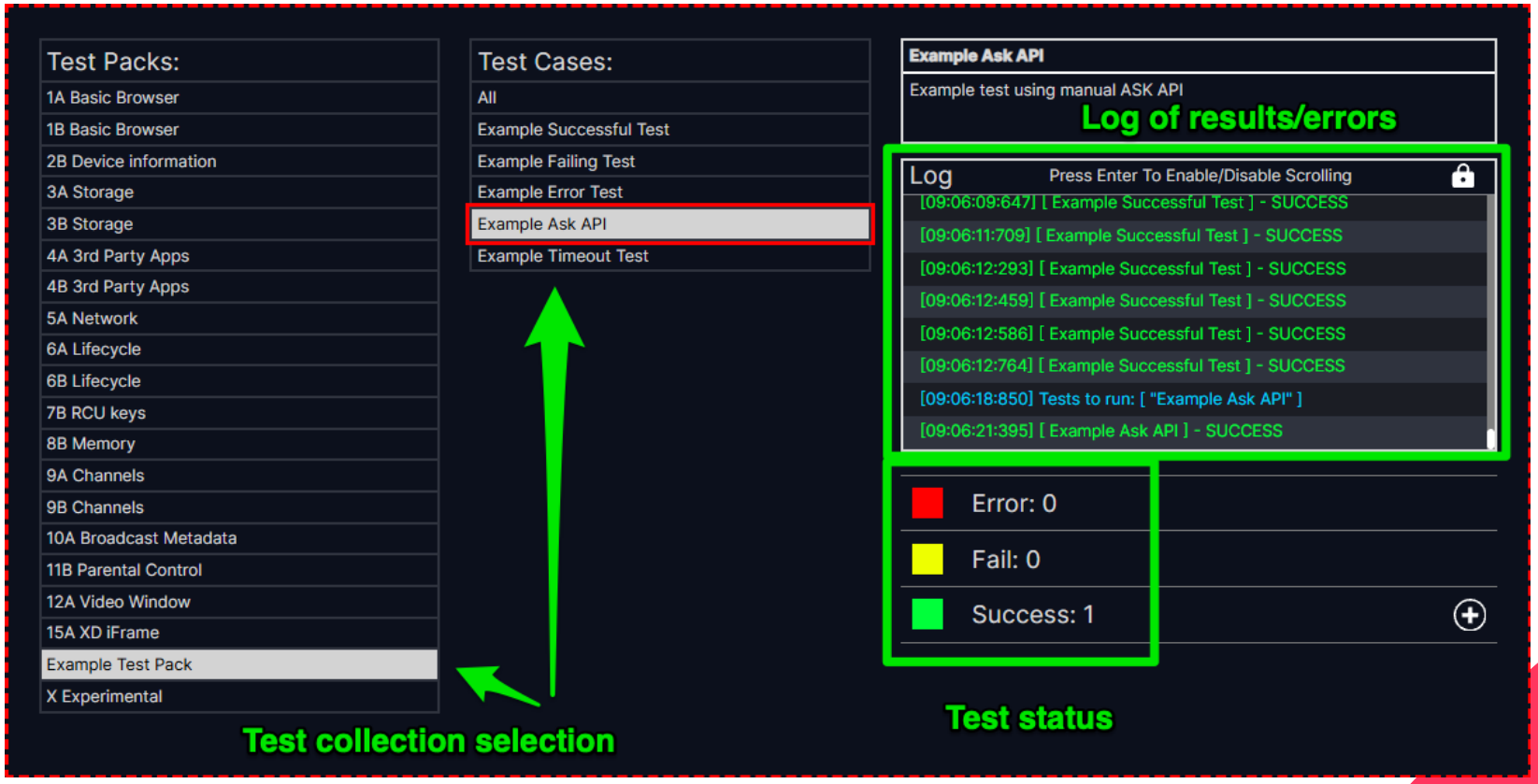
- Provide as much on screen and console feedback to the developer/tester as possible
- Simply launching TaCO on a real Freely device provides useful info on the right hand side
- Allows SLOC/LLOC combinations to be validated, check UA etc.



TaCO test selection

A test pack contains:

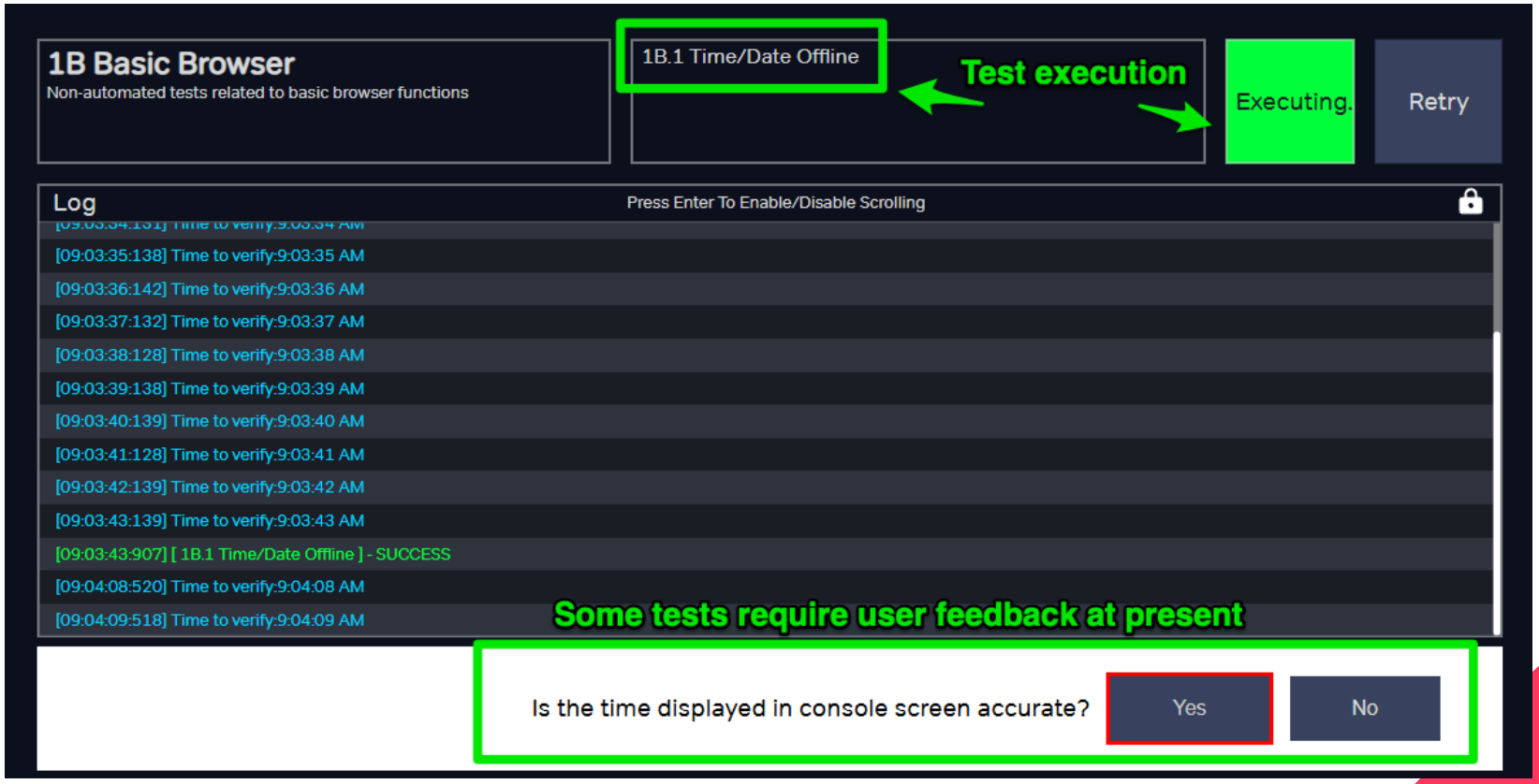
- Groups of related tests which can be run individually or as a set
- Results and errors are shown on screen for immediate feedback
- Further automation hooks are planned but not yet implemented
- ETV would like to consult with partners on how best to allow remote automation of TaCO



TaCO test execution

During test execution:

- TaCO checks for, queries and validates the response for APIs in a specified sequence allowing for threaded execution and server response timings.
- ETV identifies inconsistencies, problems and lack of support for required OpApp APIs
- Some APIs are stubbed out only, some return invalid responses under certain conitions
- Manufacturers are required to pass TaCO tests as part of the conformance regime for Freely.



TaCA – Study Mission Group

To further consider the role test and reference applications play, HbbTV has convened the Study Mission Group (SMG) for Test and Conformance Apps (TaCA).

- Review and summarise the current set of test apps
- How can test/ref. apps be incorporated into HbbTV
- Consider business aspects of development, licensing and validation

First meeting 30th July 11:30am CET

(see HbbTV e-mails for details)

Thank-you