



# CTA WAVE

## **HTML5 Issues on Embedded CE Devices Survey Results**

W3C TPAC – September 11, 2023

John Riviello, CTA WAVE HTML5 API Task Force Chair

# HTML5 Issues on Embedded CE Devices Surveys

- Surveys intended to assess the need for a performant common platform for TV Web application development
- Surveys were sent on March 3:
  - [CE Device Manufacturer Survey](https://www.surveymonkey.com/r/Q68JVR7) - <https://www.surveymonkey.com/r/Q68JVR7>
  - [App Developer Survey](https://www.surveymonkey.com/r/LDSRG66) - <https://www.surveymonkey.com/r/LDSRG66>
- Surveys were sent to:
  - CTA WAVE Mailing List
  - W3C Media & Entertainment Interest Group
  - HbbTV
  - Individual contacts
- Deadline was April 14<sup>th</sup>, but surveys are still open for responses
- 9 responses received (5 App Developers, 4 CE Device Manufacturers)

# Problem Statement

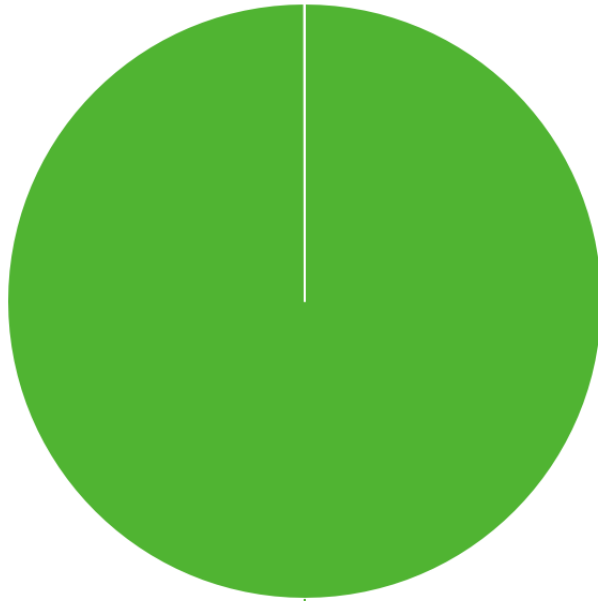
HTML5 media applications could be the way to enable universal apps across all TVs and other CE devices. However, app developers in the streaming industry have turned to **proprietary native application development** to ensure their apps **perform well on TVs and other CE devices**. This may result in **slower development and release cycles** as development teams need to customize their native TV application for each CE platform. This also **may slow innovation** because it necessitates **verifying enhanced streaming services across multiple devices**.

There are ways to address performance concerns while still leveraging web technologies. Some current web browsers support rendering and computing solutions beyond basic HTML, CSS and JavaScript that approach native application speeds and offer developers more control and flexibility in resource management with **WebAssembly** or **WebGL** toolsets. Other potential approaches are **web-based MiniApp frameworks** that run inside Super Apps and provide a mix of native and web-based components, or browsers, such as Cobalt, that support a **limited but useful set of features**.

# HTML5 Issues on Embedded CE Devices Surveys

5. Would you be willing to participate in a collaborative effort to address these issues?

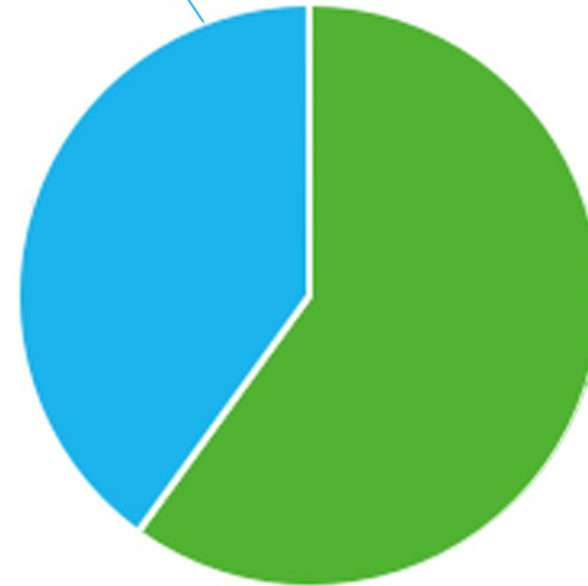
**CE Device Manufacturers**



**Yes, and my company already participates in CTA WAVE: 4**

**App Developers**

**Yes, and I would like more information about participating: 2**



**Yes, and my company already participates in CTA WAVE: 3**

# HTML5 Issues on Embedded CE Devices Surveys

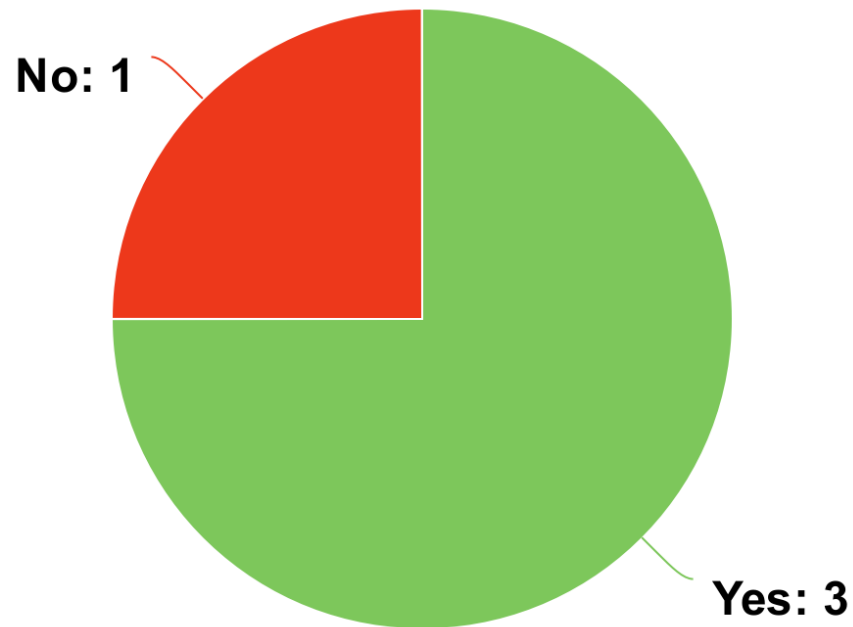
## Executive Summary

- Majority of App Developers report there are performance problems
- Majority of CE Device Manufacturers report the specialized (i.e., individual) applications for major streaming providers present a problem
- Majority of App Developers and 50% of CE Device Manufacturers agree there is a business need for a common web-based platform to encourage app developers to build their applications on a common platform
  - WebGL had the most favorable comments
  - Split on benefit of WebAssembly
  - Clear the solution has to be web-based, standardized & widely available
  - There must be interest from streaming providers for the business need to exist

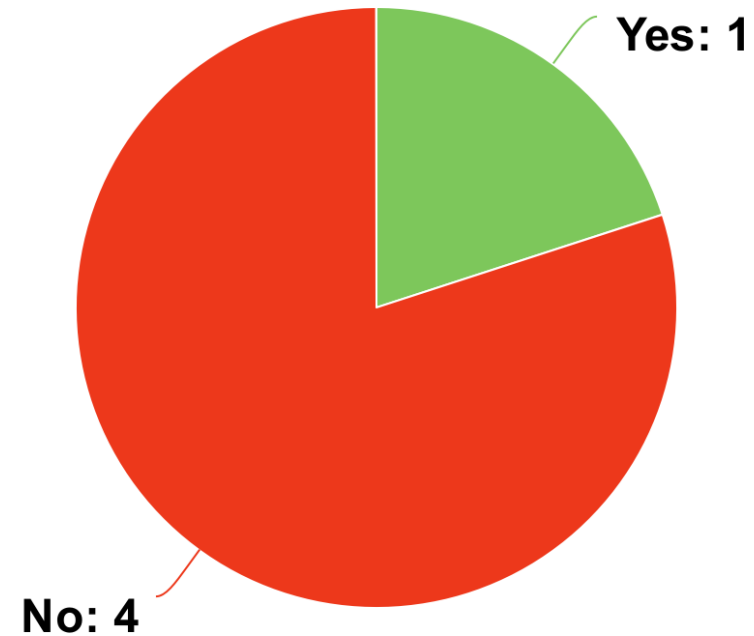
# HTML5 Issues on Embedded CE Devices Surveys

1. Do you feel HTML5 can be implemented fully with acceptable performance on the full range of CE platforms? (same question for both surveys)

**CE Device Manufacturers**



**App Developers**



# HTML5 Issues on Embedded CE Devices Surveys

1. Do you feel HTML5 can be implemented fully with acceptable performance on the full range of CE platforms? (same question for both surveys)

## CE Device Manufacturers

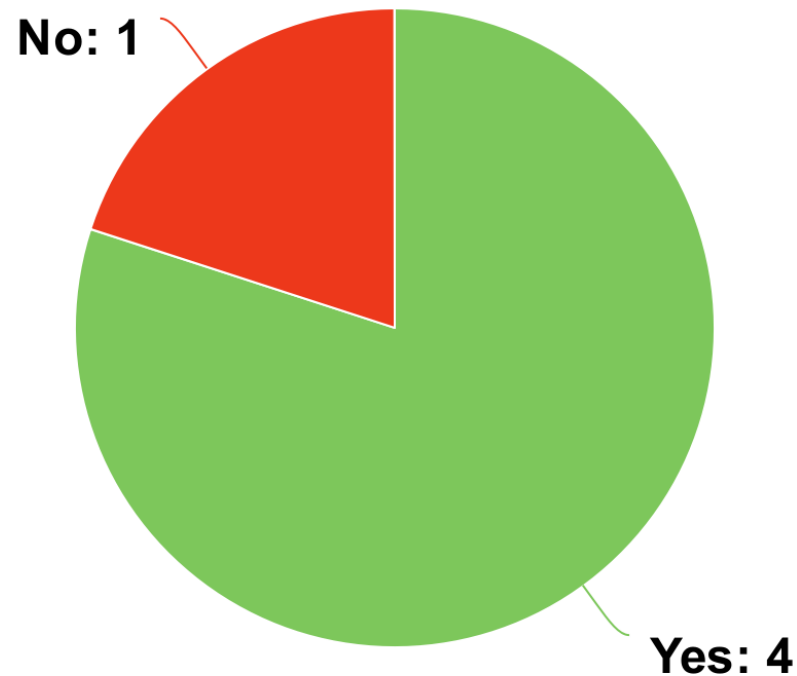
- Some content providers / aggregators may have more demanding definitions of "acceptable". It might be interesting to see what they consider to be "acceptable".
- Careful use of features, layers and frameworks should make it possible to create performant applications. Critical areas can be improved by using WebAssembly and / or WebGL.
- It depends on the complexity of the application, the level (skill) of the application creator, and the performance of the CE device.

## App Developers

- Due to interoperability differences and backwards compatibility issues, devices are classified in functional groups that are served different functionalities.
- HTML rendering performance can not handle the animations to make a polished UI.
- Performance has improved over time, but there is still considerable variability between devices and manufacturers, particularly in horizontal markets and the longevity of devices in the market (e.g., supporting 10-year-old devices)
- CSS transitions in Chromium perform at least as smoothly as alternative solutions where animations are performed via WebGL+Canvas driven by JavaScript.
- There is a challenge of finding ways lower level media capability controls, such as Web Audio, Web Codecs, etc. can be profiled and implemented in CE devices cost-effectively, while enabling an acceptable level of performance.
- There are gaps for an application to fully do what a CE platform currently does today natively (e.g. captions display, parental handling, content encryption)

# HTML5 Issues on Embedded CE Devices Surveys

2. App Developers: Is the runtime performance of web applications on TVs and other consumer devices a problem, in your opinion?

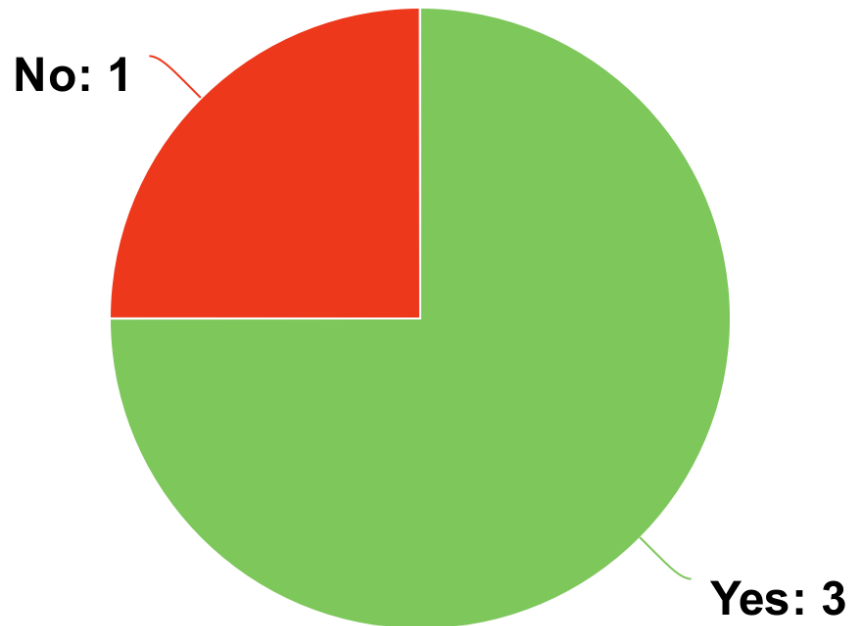


- Not necessarily a problem as basic functions needed for media applications do perform. Obviously, there is performance difference between newer/older or baseline/flagship devices.
- LightningJS performance is acceptable and JavaScript runtime is fast.
- Implementations do vary in their degree of optimisation. Some of this is due to hardware capability but much is also related to the degree to which a browser implementation has been properly optimised to utilise underlying hardware capabilities. This challenge can be magnified in horizontal markets where there may be substantial variations in price-point (and therefore underlying hardware capabilities) in devices that all purport to meet the same set of standards.
- Generally, simple applications run ok, but as soon as you add SVG, PNG, JPEG on high resolution TVs and you have to consider a single threaded browser competing with other resources on the TV, then things starts to slow down dramatically.



# HTML5 Issues on Embedded CE Devices Surveys

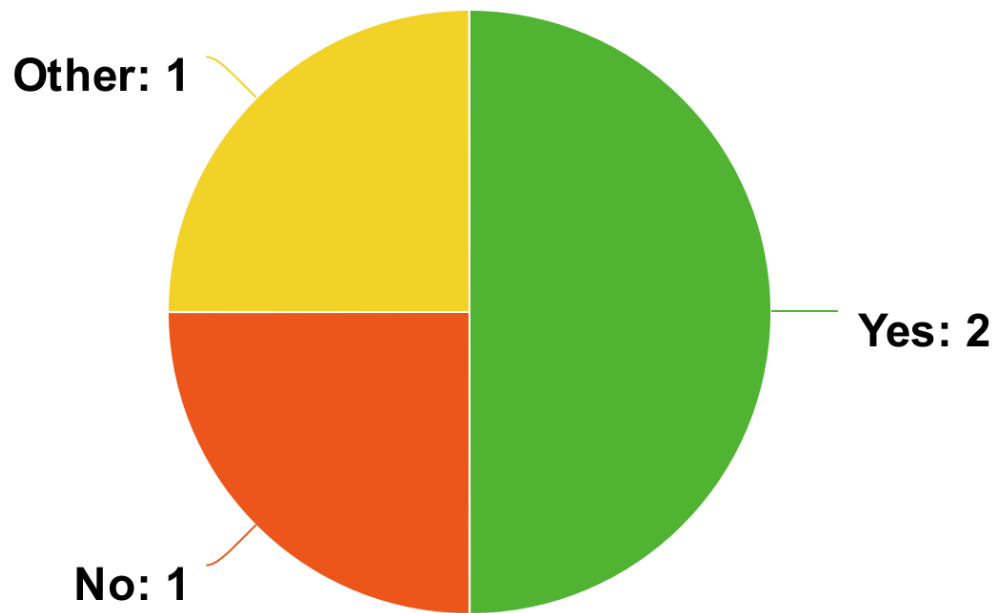
2. CE Device Manufacturers: Does supporting specialized (i.e., individual) applications for major streaming providers present a problem in development or device performance for you?



- We prefer to do as few of them as possible.
- Native applications (also Cobalt based apps) require tight integration to the chipset and drivers, and therefore has to be maintained outside of the App Developers organisation.
  - This may cause monopolistic scenarios where some manufacturers or platforms get great support while others may be left without a certain app.
  - Content owners that support all users and devices maintain multiple versions: dedicated apps for some platforms and HTML for the "long tail".
- A platform or Chipset provider already struggles to include support for major streaming binary apps, and each time a new app emerges, a lot of work has to be done, so this is not a solution that can be leveraged by tier 2 or tier 3 app developers.

# HTML5 Issues on Embedded CE Devices Surveys

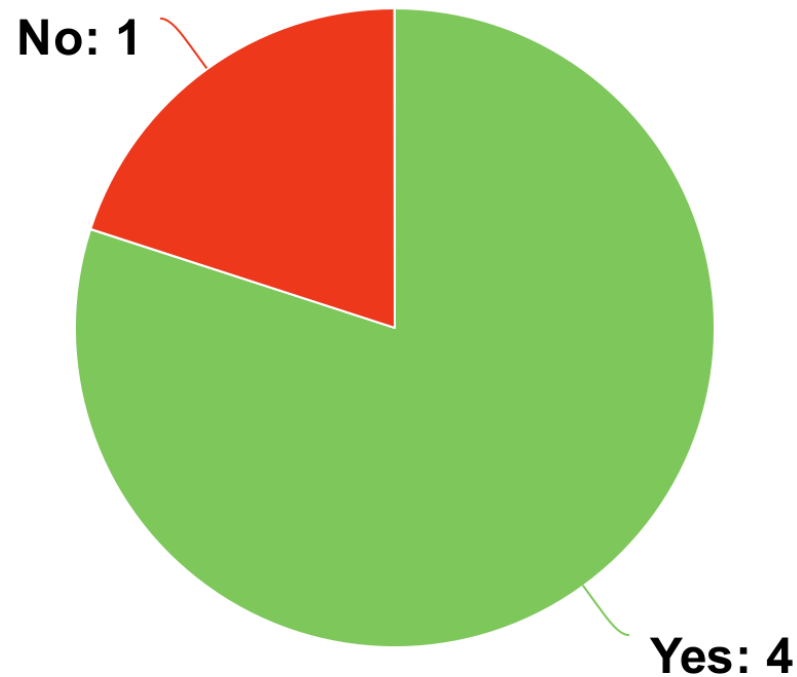
3. CE Device Manufacturers: In your opinion, is there a business need for a common web-based platform to encourage app developers to build their applications on a common platform?



- As a manufacturer, we would love there to be such a business need but there really needs to be interest from streaming providers. If there's no signs of interest from them then there's no business need.
- We already have HTML5, and in Europe we have a richer API with HbbTV. So this is mostly already there. However, some features or APIs may be costly for manufacturers to support, and therefore often not implemented or implemented with limited capabilities. Also, on embedded devices, resource consumption (RAM, CPU, GPU, video decoders) are always a concern.

# HTML5 Issues on Embedded CE Devices Surveys

3. App Developers: Is there a business need for a common web-based platform to help address app performance (or other issues related to TV applications, if any)?



- If certain specific platforms would provide direct access to additional features these can be investigated through proof of concepts and rolled out operationally if they are supported by preferably open standards.
- Having a simpler subset of components specific for TV's would be easier.
- TV-specific features: focus management, remote control input, captions, PIN management, persistent storage through power cycles, broadcast, etc.
- Need standards for how to get web apps onto TVs.
- A common web-based platform remains essential, otherwise there becomes an \*additional\* platform to develop for and support.
  - This would, in effect, double our development costs because we still would have to build for current HTML based devices.
- We would like to see performance improvements in the context of existing Web technologies, and profiling and guidance on use of HTML5 features (e.g. DOM and CSS) to help application developers achieve good performance across all devices.

# HTML5 Issues on Embedded CE Devices Surveys

4. Both Surveys: If you have experience with any of the possible solutions previously mentioned (e.g., MiniApps/Super Apps, WebAssembly, WebGL, limited-feature browsers), what is your opinion on them?

## CE Device Manufacturers

- No.
- No opinion.
- I hope we can stay within a standard browser, without native components.
- MiniApps are ok as long as they don't call out to native code.
- WebAssembly and WebGL are great tools, and we have seen these taken to their full potential yet.
- Introducing a limited browser such as Cobalt sounds good, but in practice (at least for YouTube), even if the browser is open source and fairly standard, as an app developer you would have to get each platform to include your (certified) version of Cobalt to be able to leverage it.

## App Developers

- Not needed.
- Webcomponents improve reusability but do not deliver a performance benefit.
- MiniApps and SuperApps could be interesting if they provide deeper access to CE capabilities in different devices /device classes.
- Rendering via WebGL works really well.
- WebAssembly, on its own does not provide the features an application environment needs. However it could provide some opportunity to write business logic once and compile that across multiple targets (Web, Android, iOS, etc).
- For pure UI development, performance is more constrained by rendering, so WebAssembly doesn't provide a significant performance benefit over JavaScript.
- Limited-feature browsers lack the APIs and integration (e.g., defined lifecycle) to run in both standalone and also alongside linear channel services . It is not cost effective for us to develop the same application twice.



# Web Media API Snapshot Update

September 11, 2023

W3C TPAC

# Annual Web Media API Snapshot (WMAS)

- Documents minimum set of web standards for playback of audio-video content in HTML5 based on the 4 most widely adopted browsers, with an emphasis on adaptive streaming.
  - Consumer electronic capabilities are also considered
- Work occurs within the [W3C Web Media API Community Group](#)
- CTA and W3C have agreement to co-publish annual Web Media API Snapshot
  - W3C Community Group Note
  - CTA WAVE Specification (CTA-5000)
- WMAS was updated every December since original snapshot in 2017
  - For WMAS2023, we are targeting November 1st
- Referenced by other industry standard groups to set their user agent requirements
- WMAS Automated Test Suite helps to ensure devices meet these guidelines

# Anticipated Web Media API 2023 Snapshot Updates

## Added Features

- ECMAScript 2023
- CSS Color Level 4
- Media Fragments URI 1.0 (basic)
  - Only video track ranges is required
- Update references to all WHATWG specs to a recent Review Draft
  - Now includes WebSockets

## Future Considerations:

- WebTransport
- WebAssembly
- Push API

Current Draft: <https://w3c.github.io/webmediaapi/>

Issues: <https://github.com/w3c/webmediaapi/issues>