Web-based Signage 端末
性能要件

Release Candidate Rev.

2016 年 3 月 11 日

作成： Web プラットフォーム性能ベンチマーク検討会

発行： 慶應義塾大学 SFC 研究所 先端ウェブアーキテクチャ・ラボ

目次

1 性能要求条件の概要

本文書は、Web-based Signage(ウェブ/HTML5 技術に基づくデジタルサイネージ)普及を目的に、Web-based Signage に求められるプレーヤー端末の性能に関して、ウェブ/HTML5 技術関連の要求条件を整理したものである。本文書では、３章から１０章に各要求条件が定義されている。各要件ごとの評価テスト手順については、「Web-based Signage テストスペック」文書に記されている。

1.1 性能評価の範囲

Web-based Signage は、コンテンツを管理する CMS（コンテンツ登録やプレイリストなどを管理）、コンテンツを配信する CDN（配信クラウドとネットワーク）、そして、コンテンツを再生する端末から構成される。本資料は、これら Web-based Signage システム全体のうち、端末を評価の対象にする。

端末は、以下の要素で構成される。

ハードウェア

物理的に端末を構成する構成要素を指す。具体的には、パネル、SoC（CPU, GPU, メモリなど）、入出力インタフェース（HDMI, USB, 電源など）で構成される。

オペレーティングシステム

Web-based Signage として端末ハードウェアを利用するためにインストールされる基本ソフトを指す。具体的には、Windows, Android, iOS, Linux, FreeBSD, Firefox OS などを指す。

ブラウザーランタイム

HTML, CSS, JavaScript などの Web 標準技術を動作させることができるアプリケーション基盤を指す。具体的にはウェブブラウザーを指すが、必ずしも、一般的なウェブブラウザーである必要はない。例えば、Android の WebView など、ランタイムとして用意された環境も含む。

JS プレーヤー

プレイリストに基づいてコンテンツを再生するためのアプリケーションを指す。Web-based Signage では、ブラウザーランタイム上で動作するウェブアプリケーションとして作られる。具体的には、HTML, CSS, JavaScript を駆使して開発される。

コンテンツ

コンテンツは、JS プレーヤー上で表示または実行され、サイネージ端末の前にいるユーザーに見えるものである。コンテンツは、単体画像、動画、または、ウェブアプリケーションとして制作される。中には、これらコンテンツを表示しつつ、音声を再生するものも存在する。

本資料の性能評価の対象である端末とは、上記構成要素のうち、ハードウェア、オペレーティングシステム、ブラウザーランタイムまでを意味する。

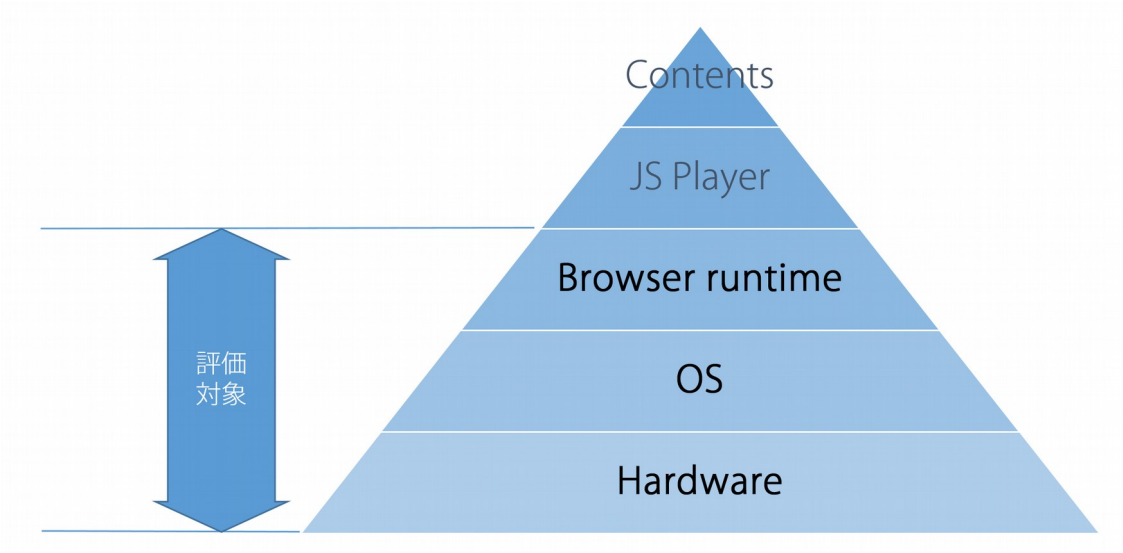
また、本性能評価は、これら構成要素を個別に評価するのではなく、これらを組み合わせた状態、つまり、実際の端末商品として総合的に評価することとする。

なお、ウェブ/HTML5 技術を使ったサイネージ端末のうち、ネットワーク接続を有しない、または、ネットワーク接続を有するもののネットワークを使わずにコンテンツを再生するスタンドアローン型のサイネージ端末（USB ストレージに保存したコンテンツを再生するタイプの端末）は、評価の対象外とする。

本性能要件文書の記載対象は、ウェブ/HTML5 の表示関連の性能のみに絞っている。例えば長期安定性（ハングアップしないこと）やタスクスケジューラ関連など、サイネージ端末に対する一般的な要件については対象としない。また、外部デバイスとの連携機能(デバイス API など)については、現状のブラウザでの実装状況をかんがみ、本版では対象としていない。

1.2 評価の方針

Web-based Signage の用途は多岐にわたる。必ずしも、端末が本資料の性能要求条件すべてを満たす必要はない。そのため、本資料では、端末を単一的なレベル評価ではなく、性能レベルを端末の用途に応じた適合用途として分類し、それぞれの適合要件を定義することとする。



2 適合要件の分類と性能評価

端末の本評価要件では、まず評価分類を定義し、その評価分類ごとに性能を軸とした適合分類に端末を分類する。分類の概要は下表のとおり。

分類表

評価分類	適合分類	説明
解像度	小密度型	離れた場所から見ることを前提にした旧来型のサイネージを指す。小さな文字を表示するコンテンツには向かない。
	中密度型	至近距離から見られることを前提に、小さな文字（映画の字幕より小さい文字）でも明瞭に読めるレベルのサイネージを指す。また、パネルのサイズが小さければ、ブランディングを重視したコンテンツにも使われる。
	高密度型	ブランディングやリアリティを重視したコンテンツを表示できるレベルのサイネージを指す。
ストレージ	キャッシュ型	コンテンツとなる画像ファイルや動画ファイルを保存する機能を有しない端末を指す。
	蓄積型	コンテンツとなる画像ファイルや動画ファイルを保存する機能を有する端末を指すが、保存サイズが小さいものを指す。
	大容量蓄積型	コンテンツとなる画像ファイルや動画ファイルを保存する機能を有する端末を指すが、保存サイズが大きいものを指す。
描画速度	低速型	ティッカー、ウェブコンテンツアニメーション、全画面遷移（移動やフェード）などのレンダリングにおいて、誰が見てもコマ落ちしているのが良く分かるレベル。 静止画しか表示しないケースにおいては、低速型で十分対応可能。
	中速型	ティッカー、ウェブコンテンツアニメーション、全画面遷移（移動やフェード）などのレンダリングにおいて、違和感がないレベル。
	高速型	ティッカー、ウェブコンテンツアニメーション、全画面遷移（移動やフェード）などのレンダリングにおいて、一般の人がコマ落ちに全く気づけないレベル。
並行処理	シングル型	同時並行処理に効果が無い端末を指す。静止画を表示するだけのシンプルな用途であればこれで十分といえる。
	マルチ対応型	同時並行処理をサポートする端末を指す。並行処理が発生しても、動画やアニメーションの再生を妨げることがないレベル。
ビデオ再生	少機能型	ブラウザーランタイムがサポートする解像度より小さいサイズのビデオであれば再生できるレベル。
	中機能型	ブラウザーランタイムがサポートする解像度のビデオを全画面で再生できるレベル。
	高機能型	ブラウザーランタイムがサポートする解像度のビデオを全画面で再生できるだけでなく、縮小または拡大表示、回転表示、移動表示したまま再生できるレベル。
オーディオ再生	少機能型	単にオーディオを再生できるレベル。
	高機能型	オーディオの生成、合成、再生、エフェクトなどに対応したレベル。
再生開始遅延	低速型	再生開始コマンドを送ってから実際に再生が開始されるまでの遅延がある程度以上存在するもの。
	高速型	再生開始コマンドを送ってから実際に再生が開始されるまでの遅延が小さいもの。

3 解像度

解像度の分類と評価基準は、下表のとおりとする。

性能要件

分類	評価基準
小密度型	ブラウザーランタイムのビューポート解像度が画素数換算で 720P 相当（1280 × 720 ピクセル）以下の解像度のものを指す。
中密度型	ブラウザーランタイムのビューポート解像度が、画素数換算で 720P 相当（1280 × 720 ピクセル）より大きく 1080P 相当（1920 x 1080 ピクセル）以下のものを指す。
高密度型	ブラウザーランタイムのビューポート解像度が、画素数換算で 1080P 相当（1920 x 1080 ピクセル）を超えるものを指す。

ブラウザーランタイムのビューポート解像度は、ブラウザーに割り当てられるビデオメモリーのサイズに依存する。また、解像度を高くするには、ブラウザーランタイムが GPU アクセラレーションをサポートしないと、期待通りのパフォーマンスを上げることができない。

本項目は、ブラウザーランタイムのビューポート解像度の大きさによって分類する。すべての端末が、本分類のいずれかに属する。

3.1 小密度型

離れた場所から見ることを前提にした旧来型のサイネージを指す。小さな文字を表示するコンテンツには向かない。具体的には、ブラウザーランタイムのビューポート解像度が画素数換算で 720P 相当（1280 × 720 ピクセル）以下の解像度を指す。この HD 相当の解像度は、一般的なサイネージの用途として、もっとも普及している解像度であり、ほとんどのライトウェイトのデジタルサイネージの用途として、十分なサイズである。

3.2 中密度型

至近距離から見られることを前提に、小さな文字（映画の字幕より小さい文字）でも明瞭に読めるレベルのサイネージを指す。また、パネルのサイズが小さければ、ブランディングを重視したコンテンツにも使われる。具体的には、ブラウザーランタイムのビューポート解像度が、画素数換算で 1280x720 を超えて 1080P(フル HD)相当（1920 × 1080 ピクセル）以下のものを指す。

中密度型は、小密度型と比べて、詳細なピクセル表現が可能となり、とりわけ、近い距離なら小さな文字でも明瞭に読み取れる。フロア案内などのメッセージボードにも活用が可能である。また、42 インチ程度のパネルであれば、ブランディングを重視したコンテンツにも十二分に活用できる。

中密度型は、直近の Web-based Signage では必須の機能となっていくと考えられる。

3.3 高密度型

ブランディングやリアリティを重視したコンテンツを表示できるレベルのサイネージを指す。具体的には、ブラウザーランタイムのビューポート解像度が、画素数換算でフル HD 相当（1920 × 1080 ピクセル）を超えるものを指す。

高密度型は、大型パネルを使いつつも、粗が目立たない。また、パネルのサイズが小さければ、本物と見間違うほどのリアリティを再現できる。既存のハイエンドのデジタルサイネージシステムと同等の品質を提供する。

4 ストレージ

ストレージの分類と評価基準は、下表のとおりとする。

性能要件

分類	評価基準
キャッシュ型	コンテンツとなる画像ファイルや動画ファイルを保存する機能を有しない、または、保存機能は有するが、ブラウザーランタイムが W3C で規定された Indexed Database API[※1] または File API: Directories and System[※2] のいずれをも実装していない端末を指す。
蓄積型	コンテンツとなる画像ファイルや動画ファイルを保存する機能を有し、かつ、ブラウザーランタイムが W3C で規定された Indexed Database API[※1] または File API: Directories and System[※2] のいずれかを実装している端末を指すが、そのうち、保存サイズが 1024MB 未満のものを指す。
大容量蓄積型	コンテンツとなる画像ファイルや動画ファイルを保存する機能を有し、かつ、ブラウザーランタイムが W3C で規定された Indexed Database API[※1] または File API: Directories and System[※2] のいずれかを実装している端末を指すが、そのうち、保存サイズが 1024MB 以上のものを指す。

端末がストレージをサポートするかどうかによって、ネットワークトラフィック、端末起動からコンテンツ再生までの時間、安定的な長時間運転などのパフォーマンスに影響する。

本項目は、ストレージの有無、また、蓄積可能な容量によって分類する。すべての端末が、本分類のいずれかに属する。

なお、端末のストレージに事前蓄積してから再生するには、HTML5(Javascript 含む)の範囲外の手法で実現することもできる。本要求条件文書では HTML5 ベースの方法を対象としているため、この項目もそれを用いたものとなっている。

4.1 キャッシュ型

キャッシュ型は、ストレージを持たない、または、ストレージを有するものの、ブラウザーランタイムが W3C で規定された Indexed Database API[※1] および File API: Directories and System[※2] のいずれも実装していない端末を指す。

キャッシュ型は、端末を起動するたびにコンテンツのダウンロードが必要となり、コンテンツ再生までに時間がかかる。ただし、ブラウザーキャッシュによって、次回起動時におけるダウンロードデータ量は抑えられる可能性があるが、本分類では、ブラウザーキャッシュの有無は考慮しない。なお、ここで言うブラウザーキャッシュとは、Application Cache ではないので注意して欲しい。

また、ネットワークの状況によっては動画の再生が不安定になるなどのデメリットが想定される。一方、ストレージが無い分だけ、端末のハードウェアコストが抑えられる。また、コンテンツ配信サーバーと端末との間のネットワーク環境が良好であれば（オンプレミス型配信環境など）、さほど問題にならない。

4.2 蓄積型

蓄積型は、ストレージを持ち、かつ、ブラウザーランタイムが W3C で規定された Indexed Database API[※1]または File API: Directories and System[※2]のいずれかを実装している端末を指す。

蓄積型は、表示するコンテンツをストレージに蓄積し、コンテンツ再生時には、蓄積されたコンテンツを使うため、ネットワーク環境に依存しない安定再生が期待できる。また、コンテンツを都度ダウンロードする必要がないため、コンテンツが更新されていない限り、端末を起動してからコンテンツの再生までの時間が大幅に短縮される。

4.3 大容量蓄積型

大容量蓄積型は、蓄積型のうち、蓄積容量が 1024MB 以上のものを指す。

※1 W3C Indexed Database API

http://www.w3.org/TR/IndexedDB/

ストレージ型においては、Indexed Database API だけでなく、Blob オブジェクトの保存、および、File API 仕様で規定された Blob URL の生成（createObjectURL()メソッド）もサポートしなければいけない。

http://www.w3.org/TR/FileAPI/#dfn-createObjectURL

※2 W3C File API: Directories and System

http://dev.w3.org/2009/dap/file-system/file-dir-sys.html

2016 年 3 月現在、W3C File API: Directories and System は廃止となり、新たに File System API が考案されている。

http://w3c.github.io/filesystem-api/

ただし、まだ公開版草案になっておらず、ブラウザーの実装もないことから、しばらくは、File API: Directories and System が使われることになる。将来的には、File System API に置き換えられることになる。

5 描画速度

描画速度の分類と評価基準は、下表のとおりとする。

性能要件

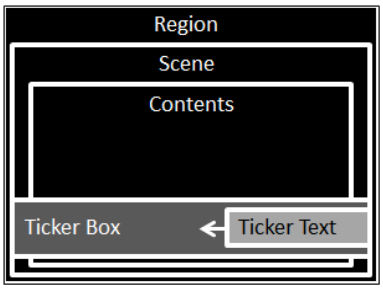
分類	評価基準
低速型	CSS transform / transition を利用して、ティッカーテキスト(画像ではなくフォントで実現し、高さが画面高の 10%)の水平移動をした場合及び静止画像（全画面）の水平移動を行った場合いずれかで、15 秒間でのコマ落ち(フレーム落ち)が合計 10 フレームを超えるものとする。
中速型	CSS transform / transition を利用して、ティッカーテキスト(画像ではなくフォントで実現し、高さが画面高の 10%)の水平移動をした場合及び静止画像（全画面）の水平移動を行った場合のいずれかで、15 秒間でのコマ落ち(フレーム落ち)の合計が 3 フレームを超えて、両方の場合で、合計 10 フレーム以下となるものとする。
高速型	CSS transform / transition を利用して、ティッカーテキスト(画像ではなくフォントで実現し、高さが画面高の 10%)の水平移動をした場合及び静止画像（全画面）の水平移動を行った場合の両方で、15 秒間でのコマ落ち(フレーム落ち)が合計 3 フレーム以下となるものとする。

コンテンツアニメーションにおいてコマ落ち(フレーム落ち)は、パフォーマンスを測る上で、重要な指標となる。本項目は、コマ落ち(フレーム落ち)を基準に分類する。すべての端末が、本分類のいずれかに属する。

本評価は、以下の検証を総合的に判断して分類分けを行うこととする。

ティッカー

ティッカーはシーン領域上に用意されたティッカー領域（Ticker Box）の中を、ティッカーテキスト（Ticker Text）が右から左に人が読める速度で移動する。



パネル横置きを前提とし、全画面静止画コンテンツをシーン領域に表示した状態で、高さが全体の 10%を占めるティッカー領域内で、ティッカーテキストが一定の時間をかけて右端から左端に到達する速度で移動するものとする。この状況で、ティッカーの動きを評価する。使用するティッカーテキストは、画像ではなくフォントで実現することとし、また十分な長さを有することとする。

遷移効果の検証には、CSS TransformとCSS Transitionを採用する。

CSSの例
遷移前：transform: translateX(1920px); transition: transform 30s linear 0s;
遷移時：transform: translateX(-3840px);
※ 遷移前のtranslateX()の引数とtransformのdurationは画面横幅とテキストの幅を考慮して調整すること

画面遷移

シーン切り替えに使う遷移効果を検証材料とする。本評価においては、ブラウザーランタイムのビューポート解像度と同サイズの画像を使い、スライドの画面遷移を検証することとする。実際のサイネージコンテンツでは、1秒以下の遷移を使うが、ここではパフォーマンス評価を目的とするため、計測可能な秒数で評価する。画面遷移にはスライドの他にフェードインやスケールもある。以下に事例を示す。

遷移効果の検証には、CSS TransformとCSS Transitionを採用する。

フェードイン

CSSの例
遷移前：opacity: 0; transition: opacity 5s linear 0s;
遷移時：opacity: 1;

スライド

CSSの例
遷移前：transform: translateX(-1920px); transition: transform 5s linear 0s;
遷移時：transform: translateX(0px);
※ 遷移前のtranslateX()の引数は画面横幅に合わせること

スケール

CSS の例

遷移前：transform: scale(0); transition: transform 5s linear 0s;
遷移時：transform: scale(1)

5.1 低速型

いずれの評価においても、誰が見てもコマ落ちが気になるレベル。ティッカーは読みづらい。

5.2 中速型

いずれの評価においても、実用に支障がないレベルだが、よく見れば、一般の人でもコマ落ちが認識できるレベル。

5.3 高速型

いずれの評価においても、期待通りの表現が再現され、一般の人ではコマ落ちを認識することができないレベル。

6 並行処理

並行処理の分類と評価基準は、下表のとおりとする。

性能要件

分類	評価基準
シングル型	同時並行処理に効果が無い端末を指す。静止画を表示するだけのシンプルな用途であればこれで十分といえる。具体的には、ブラウザーランタイムが Web Workers を実装していない、または実装していたとしても、ワーカースレッド 2 個を使用して分散処理した場合に、1 個での処理に比べ、トータルの処理時間が短縮しないものを指す。
マルチ対応型	同時並行処理をサポートする端末を指す。並行処理が発生しても、動画やアニメーションの再生を妨げることがないレベル。具体的には、ブラウザーランタイムが Web Workers を実装しており、ワーカースレッド 2 個を使用して分散処理した場合に、1 個での処理に比べ、トータルの処理時間が短縮するものを指す。

Web-based Signage 用 JS プレーヤーは、プレイリストに基づいたメディア表示や再生だけではなく、平行してさまざまな処理を行う。また、マルチリージョンに対応した JS プレーヤーでは、複数の画面を分割表示しつつ、それぞれを独立して再生する必要がある。

この場合、レンダリングを伴わない処理をバックグラウンドで実行することで、メディア表示や再生を妨げることなく、さまざまな並行処理が可能となる。

本項目は、スレッドによる分割処理によって、トータルの処理時間の短縮が認められるかどうかで判断する。すべての端末が、本分類のいずれかに属する。

6.1 シングル型

シングル型は、ウェブランタイムが Web Workers を実装していない、または、実装しているが、スレッド処理をしても効果があまり認められないレベルを指す。一般的に、CPU がシングルコアの場合はこれに相当する。

6.2 マルチ対応型

マルチ対応型は、ブラウザーランタイムが Web Workers を実装しており、2 つ以上のスレッドを同時に処理することが可能であり、トータルの処理速度の短縮が認められるレベルを指す。一般的に、CPU がマルチコアの場合はこれに相当する。

マルチ対応型に属するためには、1 つのワーカースレッドでの処理時間に対して、2 つのワーカースレッドで分散処理した時にトータルの処理時間の短縮が認められることを条件とする。

7 ビデオ再生

ビデオ再生の分類と評価基準は、下表のとおりとする。

性能要件

分類	評価基準
少機能型	次を満たすこと。（1）mp4(H.264/AAC/MP4)もしくは webm(VP8/Vorbis/WebM)の少なくともいずれか一方をサポートすること。（2）Javascript から video 要素のオブジェクトの play()メソッドを実行したら、ビデオの再生が開始すること。 ビデオ解像度に関しては、ブラウザーランタイムがサポートする解像度のビデオを全画面で再生できなくてもよいとする。
中機能型	上記（1）（2）の条件に加え、ブラウザーランタイムがサポートする解像度の 30fps 以上のフレームレートのビデオをコマ落ちなく全画面で再生できること。
高機能型	上記（1）（2）の条件に加え、ブラウザーランタイムがサポートする解像度の 30fps 以上のフレームレートのビデオをコマ落ちなく全画面で再生でき、かつ、縮小または拡大後の再生、回転後の再生、移動後の再生（具体的には CSS Transforms を video 要素に適用しての再生）ができるレベル。

ビデオを再生するためには、それなりの CPU パワーと GPU パワーを要求する。特にブラウザーランタイムがサポートする解像度のビデオを全画面で再生するためには、ブラウザーランタイムがハードウェアデコードおよび GPU アクセラレーションに対応している必要がある。この差が、ビデオ再生のパフォーマンスに大きく影響を及ぼす。

ビデオ再生の分類に属するためには、以下の要件をすべて満たす必要がある。

1. 以下のビデオタイプのうち最低でも一つをサポートすること

MIME-Type	ビデオコーデック	オーディオコーデック	コンテナ
video/mp4	H.264	AAC	MP4
video/webm	VP8	Vorbis	WebM

2. 自動再生をサポートすること

   近年のスマートフォンやタブレット向け OS のブラウザーでは、ユーザーの操作なしに、JavaScript からビデオの自動再生を禁止しているものがある。本分類に属するためには、この制限を設けてはいけない。

   実際には、JavaScript から video 要素のオブジェクトの play()メソッドを実行したら、ビデオの再生が開始される必要がある。

たとえ、端末がビデオ再生をサポートしていたとしても、以上の要件を満たさない場合は、ビデオ再生の分類には属さない。

7.1 少機能型

少機能型は、主に、ソフトウェアデコードにしか対応していないブラウザーランタイムが該当する。CPU のみでデコード及びレンダリングを行うため、全画面のなめらかな再生は期待できないが、解像度が小さければ問題なく再生できる。

少機能型は、HTML5 仕様で規定された video 要素を使うことを前提としている。ただし、HTML 上にマークアップされた video 要素を再生するのではなく、すべて JavaScript から video 要素の生成と再生開始が行えなければいけない。

7.2 中機能型

中機能型は、前述の少機能型に加え、サイネージとしては最も一般的な用途を想定し、全画面のビデオ再生を必須とする。小さい解像度を引き伸ばして全画面表示する場合は、中機能型に属さない。少なくとも、ブラウザーランタイムがサポートする解像度と同じ解像度のビデオを拡大することなく全画面で滑らかに再生できるレベルを指す。30fps でエンコードされたビデオデータを、コマ落ちすることなく再生できなければいけない。

7.3 高機能型

高機能型は、中機能型をサポートするのはもちろんのこと、そのビデオを縮小または拡大表示、回転表示、移動表示しつつ、滑らかに再生できるレベルを指す。実際には、CSS Transform※を video 要素に適用しても、期待通りの変形を保ったまま再生できるレベルを指す。

※ CSS Transform
http://www.w3.org/TR/css3-transforms/

8 オーディオ再生

オーディオ再生の分類と評価基準は、下表のとおりとする。

性能要件

分類	評価基準
少機能型	次を満たすこと。（1）mp3(MP3/MP3)もしくは mp4(AAC/M4A), webm(Vorbis/WebM), ogg(Vorbis/Ogg)の少なくとも一種類のオーディオ再生をサポートすること。（2）Javascript から audio 要素のオブジェクトの play()メソッドを実行したら、オーディオの再生が開始すること。
高機能型	上記の条件に加え、Web Audio API の各種インターフェースの利用により、オーディオの生成、合成、エフェクトなどに対応したレベル。具体的には、少なくとも次のインターフェースをサポートすること。AudioContext, AudioNode, AudioDestinationNode, AudioBuffer, AudioBufferSourceNode, MediaElementAudioSourceNode, AudioParam, GainNode, DelayNode, OscillatorNode。

オーディオ再生の分類に属するためには、以下の要件をすべて満たす必要がある。

1. 以下のオーディオタイプのうち最低でも一つをサポートすること

MIME-Type	オーディオコーデック	コンテナ
audio/mp3	MP3	MP3
audio/mp4	AAC	M4A
audio/webm	Vorbis	WebM
audio/ogg	Vorbis	Ogg

2. 自動再生をサポートすること

   近年のスマートフォンやタブレット向け OS のブラウザーでは、ユーザーの操作なしに、JavaScript からオーディオの自動再生を禁止しているものがある。本分類に属するためには、この制限を設けてはいけない。

   実際には、JavaScript から audio 要素のオブジェクトの play()メソッドを実行したら、オーディオの再生が開始される必要がある。

たとえ、端末がオーディオ再生をサポートしていたとしても、以上の要件を満たさない場合は、オーディオ再生の分類には属さない。

8.1 少機能型

少機能型は、HTML5 仕様で規定された audio 要素を使うことを前提としている。ただし、HTML 上にマークアップされた audio 要素を再生するのではなく、すべて JavaScript から audio 要素の生成と再生開始が行えなければいけない。

8.2 高機能型

高機能型は、HTML5 仕様で規定された audio 要素を使ったオーディオを再生できるだけでなく、W3C Web Audio API を使ったオーディオの生成、合成、エフェクトなどに対応したレベルを指す。

Web Audio API を使うことで、音源をスクリプトから生成することでダウンロードデータの低減に寄与するだけでなく、複数の音源を合成することで、多彩なオーディオ表現が可能となる。

Web based Signage では、Web Audio API がカバーする高度なオーディオシンセサイジングのすべてを必要としない。基本的には、以下のユースケースをカバーできれば良い。

- ✓ audio 要素にセットされたオーディオを Web Audio API でコントロールする
- ✓ 複数の音源を同時に再生する
- ✓ 個々の音源に音量調整と遅延を適用する
- ✓ サイン波を指定することで単調な音源をスクリプトから生成する

高機能型に分類されるためには、上記のユースケースを実現する必要がある。少なくとも以下のインタフェースをサポートすれば、実現可能である。

- ✓ AudioContext
- ✓ AudioNode
- ✓ AudioDestinationNode
- ✓ AudioBuffer
- ✓ AudioBufferSourceNode
- ✓ MediaElementAudioSourceNode
- ✓ AudioParam
- ✓ GainNode
- ✓ DelayNode
- ✓ OscillatorNode

9 再生開始遅延

ビデオ・オーディオ再生時の遅延に関する性能要件は、下表のとおりとする。ビデオ・オーディオそれぞれについて評価する。

性能要件

分類	評価基準
低速型	再生開始コマンドを送ってから実際に再生が開始されるまでの遅延が、ビデオ・オーディオのいずれか一方もしくは両方において 0.5 秒以上のもの。
高速型	再生開始コマンドを送ってから実際に再生が開始されるまでの遅延が、ビデオ・オーディオ共に 0.5 秒未満のもの。

本評価は以下の手順で実施することとする。下記は video について記すが、audio も同様。

1. JavaScript から video 要素（HTMLVideoElement オブジェクト）を生成する。
2. 生成した HTMLVideoElement オブジェクトの preload プロパティに"auto"をセットする。
3. 生成した HTMLVideoElement オブジェクトに timeupdate イベントのリスナーをセットする。
4. 生成した HTMLVideoElement オブジェクトの src プロパティにビデオファイルの URL をセットする。
5. 生成した video 要素（HTMLVideoElement オブジェクト）をドキュメントに挿入する。
6. 数秒経過後に、HTMLVideoElement オブジェクトの play()メソッドを実行する。
7. play()メソッドを呼び出してから、最初に生成した timeupdate イベントが発生するまでの時間を、本評価の結果とする。

一般的なブラウザーでは、HTML 上に video 要素や audio 要素を生成し、src 属性にビデオ・オーディオファイルの URL をセットした時点で、再生開始位置から特定の範囲のビデオデータ・オーディオデータをダウンロードしデコード済みの状態でスタンバイする。これによって、いつでもすぐに再生できるようにすることができる（意図的に video 要素・audio 要素の preload 属性に"none"を指定した場合は除く)。近年のスマートフォンやタブレット向け OS のブラウザーでは、プリロードを禁止しているものがあるが、こういった環境では、高速型の要件を満たすことは難しいといえる。

多くのブラウザーでは、ビデオが再生されると timeupdate イベントは 250 ミリ秒ごとに発生する。そのため、play()メソッドを呼び出してから速やかにビデオが再生される環境であれば、500 ミリ秒（0.5 秒）の間に少なくとも 1 回は timeupdate イベントが発生するはずである。

play()メソッドを呼び出すまでに、十分なプリロード分のメディアデータがロードできるよう、本評価においては、十分なネットワーク帯域と、サーバーレスポンスを確保すること。