

MULTIMEDIA WEB

Webでできるマルチメディア表現

折原 レオナルド 賢

 CyberAgent.
adtech **studio**



AI Messenger

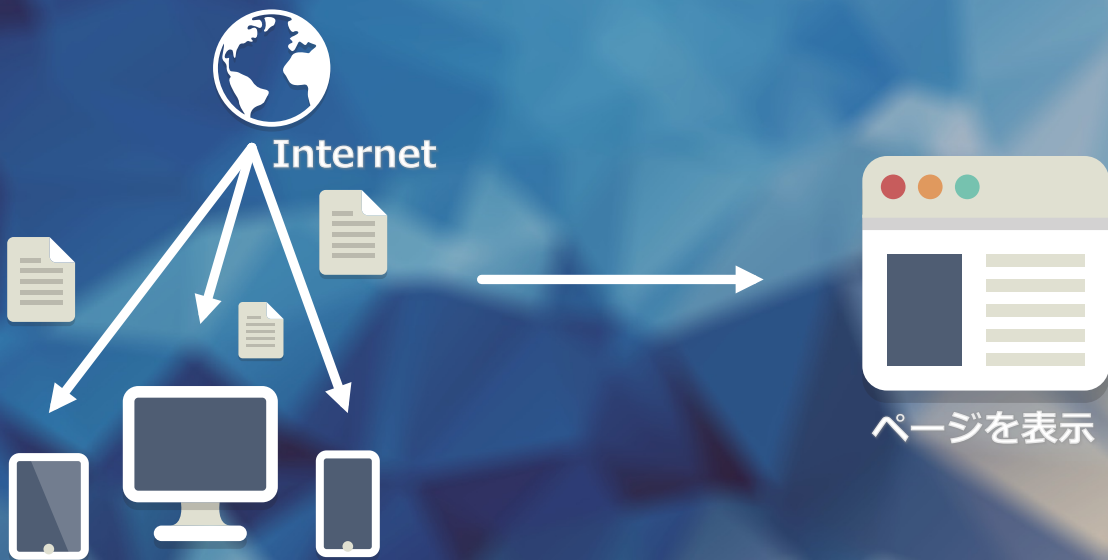
普段はアドテクスタジオのAI Messengerというプロダクトをやっています



Scroll Down

そもそもWebって

▶ 文書を配信するために規格化されたシステム



そもそもWebって

▶ 文書を配信するために規格化されたシステム



Payment Request API

Web GL

Web Audio API

Web USB API

Web Bluetooth API

Service Worker

Web MIDI API

今では多くのことができるように

etc ...

Progressive Web App

▶ デバイスやAPIのできることに応じて段階的に高機能に



Webで仕事向けアプリが動いたり
ゲームが動いたりする

今回取り扱うAPI

マルチメディアに Focus !

- **Web Audio API**
 - JavaScript上で音のミキシング、編集、フィルタリング
- **Web MIDI API**
 - MIDIの入出力をJavaScriptで制御
- **WEB GL (three.js)**
 - Canvas要素を使って三次元グラフィックスを生成
- **Service Worker**
 - HTMLページとは別で動作するプログラム

Web Audio API

DEMO 1

<https://youtu.be/L14tUQe7MWM>

Web Audio API

```
const context = new window.AudioContext();  
const sourceNode = context.createMediaElementSource();  
const gainNode = context.createGain();  
const biquadFilterNode = context.createBiquadFilter();  
const analyserNode = context.createAnalyser();
```

source
Node

gain
Node

biquad
Filter
Node

analyser
Node



Web Audio API

```
sourceNode.connect( gainNode );
```



実機の音響機器をつなぐようにconnectしていきます

Web Audio API

```
sourceNode.connect( gainNode );  
gainNode.connect( biquadFilterNode );
```



実機の音響機器をつなぐようにconnectしていきます

Web Audio API

```
sourceNode.connect( gainNode );  
gainNode.connect( biquadFilterNode );  
biquadFilterNode.connect( analyserNode );
```



実機の音響機器をつなぐようにconnectしていきます

Web Audio API

```
sourceNode.connect( gainNode );  
gainNode.connect( biquadFilterNode );  
biquadFilterNode.connect( analyserNode );  
analyserNode.connect( context.destination );
```



実機の音響機器をつなぐようにconnectしていきます

Web Audio API

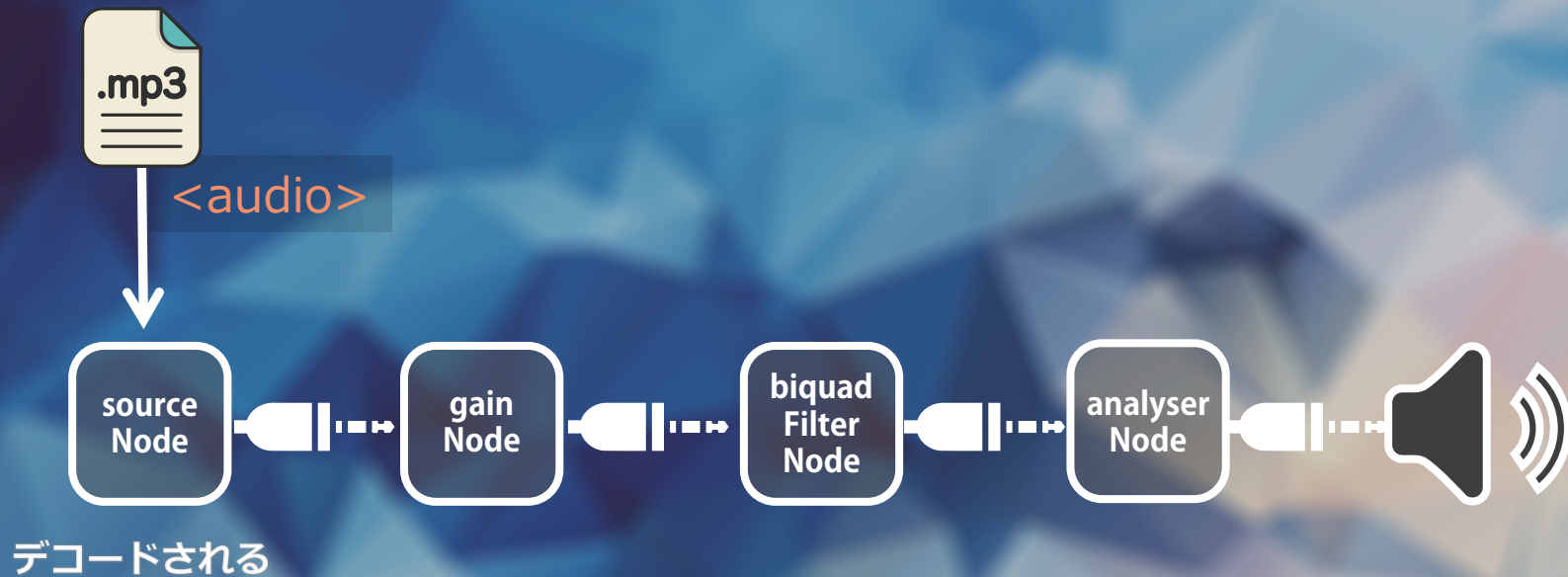
```
sourceNode.connect( gainNode );  
gainNode.connect( biquadFilterNode );  
biquadFilterNode.connect( analyserNode );  
analyserNode.connect( context.destination );
```

Play

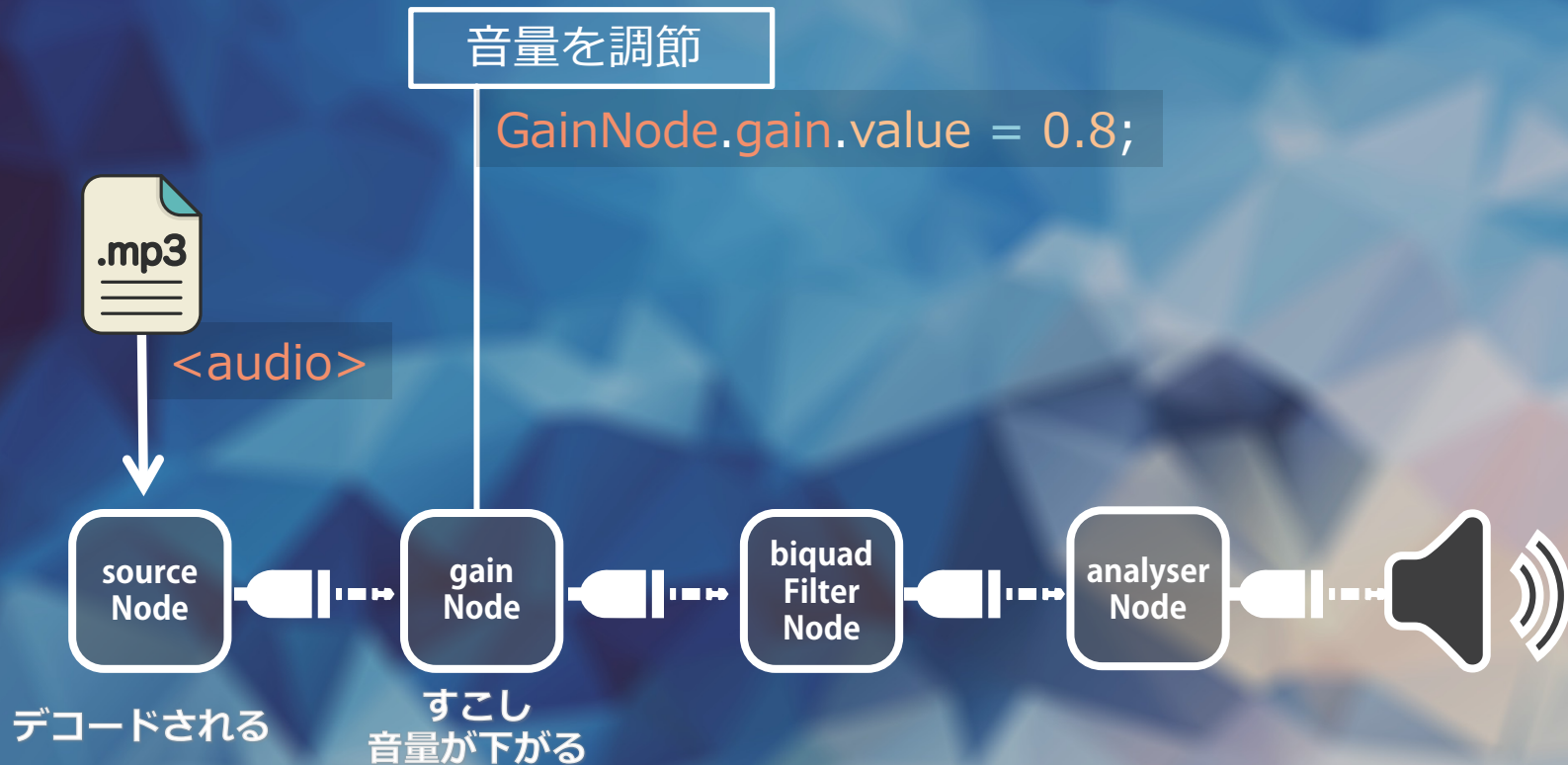


実機の音響機器をつなぐようにconnectしていきます

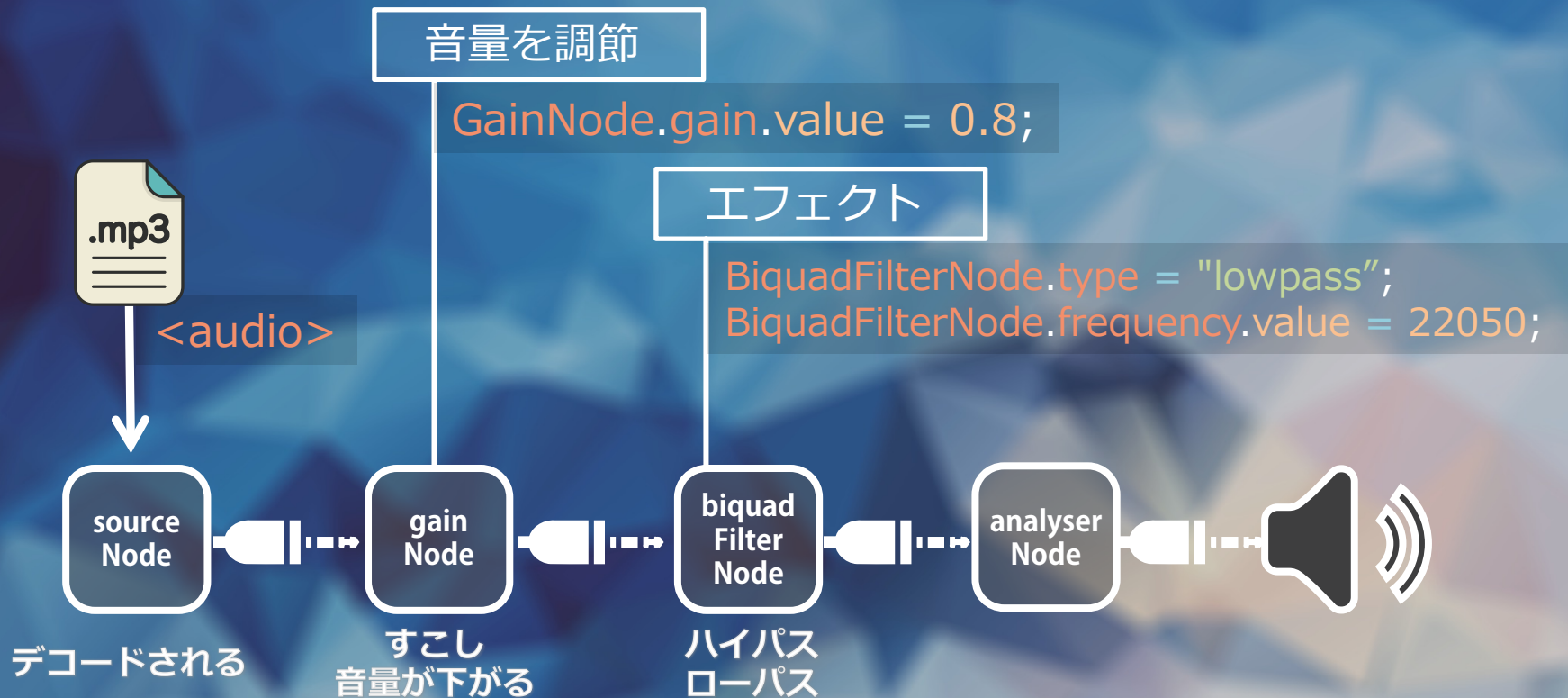
Web Audio API



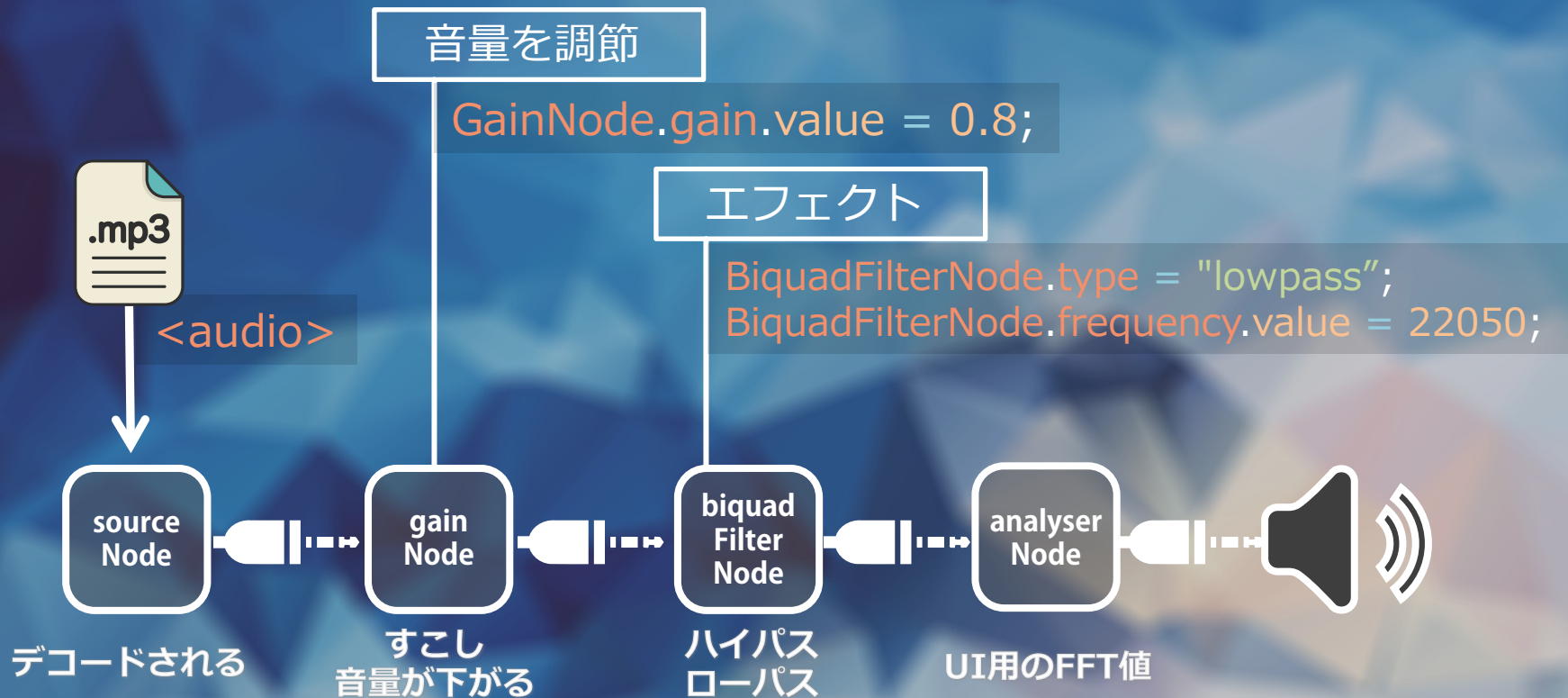
Web Audio API



Web Audio API



Web Audio API



Web MIDI API

MIDI機能を有した機器とJavaScriptを接続できる



MIDIコントローラ
キーボードとか



MIDIメッセージ



Web MIDI API

MIDIについて



MIDI端子



MIDIコントローラー

Web MIDI API

DEMO 2

<https://youtu.be/yH-JNIVVCME>

Web MIDI API

```
navigator.requestMIDIAccess({  
  sysex: true  
}).then( onMIDISuccess, onMIDIFailure );
```

```
const onMIDISuccess = ( midi ) => {  
  midi.inputs.onmidimessage = onMIDIMessage;  
}
```

```
const onMIDIMessage = ( event ) => {  
  console.log( event.data );  
}
```

MIDI信号がくると
このfunctionが呼ばれる



MIDIコントローラ
キーボードとか



MIDIメッセージ



Web

WEB GL (three.js)

DEMO3

<https://youtu.be/5wdT8o6n30s>

WEB GL (three.js)

```
const scene = new THREE.Scene();
```



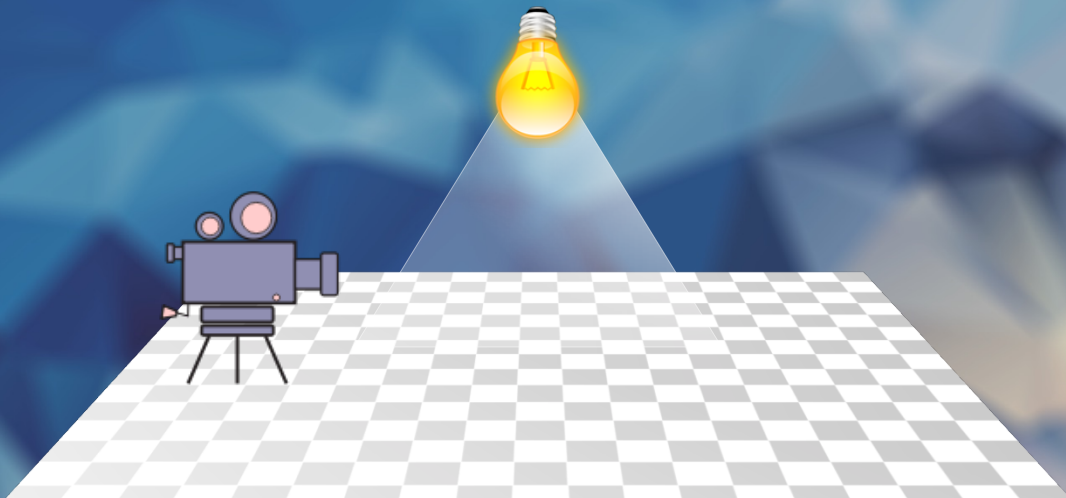
WEB GL (three.js)

```
const scene = new THREE.Scene();  
const camera = new THREE.PerspectiveCamera( fov, aspect, near, far );
```

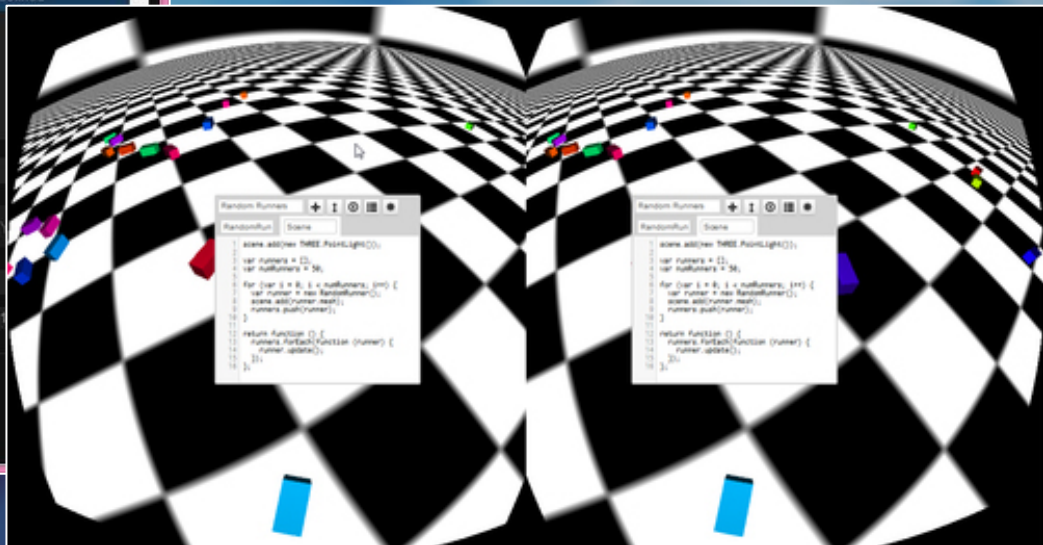
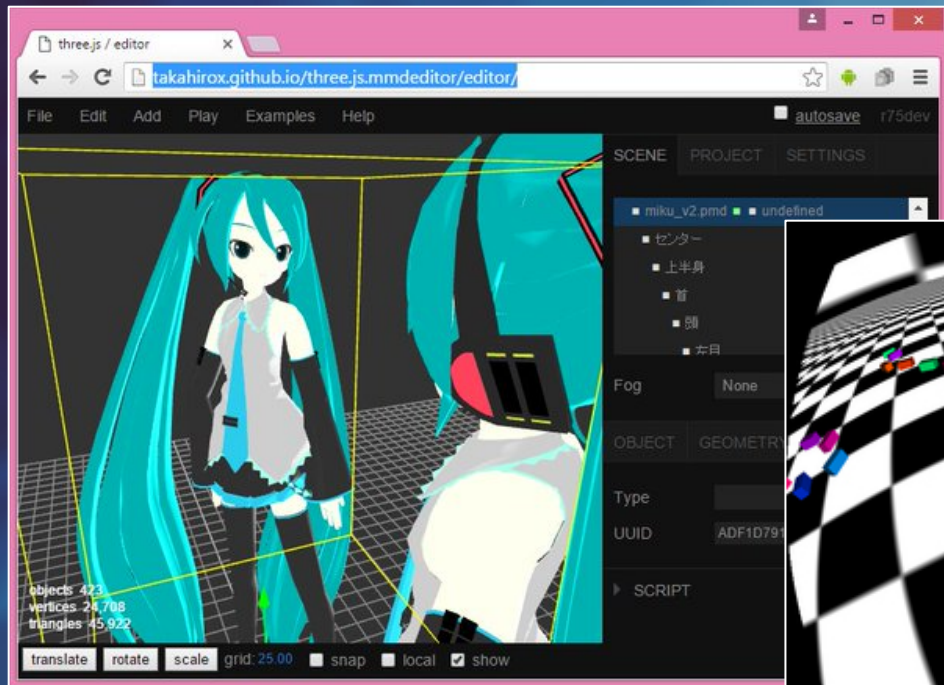


WEB GL (three.js)

```
const scene = new THREE.Scene();  
const camera = new THREE.PerspectiveCamera( fov, aspect, near, far );  
const light = new THREE.DirectionalLight( 0xffffff );
```

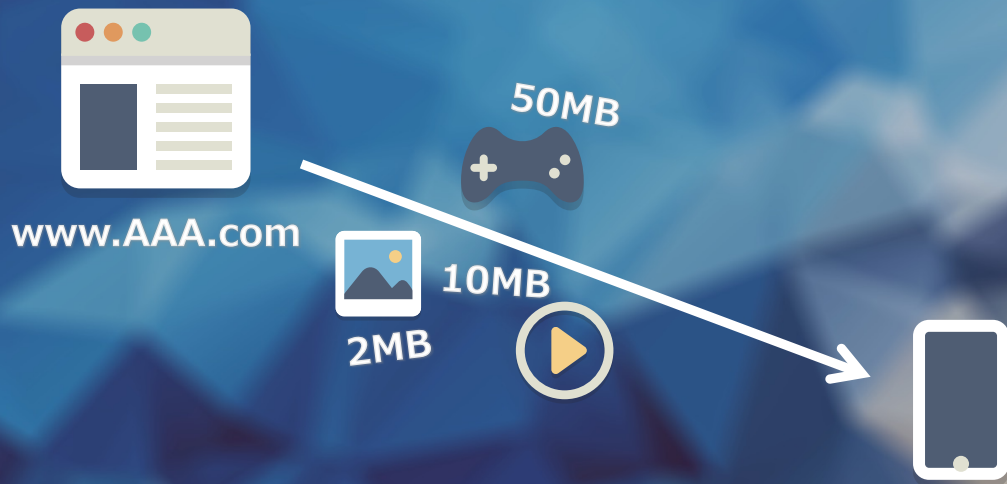


WEB GL (three.js)



マルチメディアの困るところ

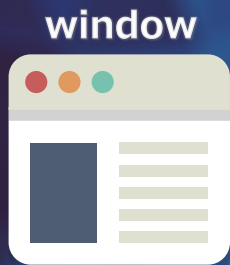
▶ リンクひとつ踏んだだけで大容量ファイルをDLされるのは困る



リッチなコンテンツはネイティブでやるべき？

Service Worker

▶ HTMLページとは別で動作するプログラム



```
navigator.serviceWorker.register( 'service-worker.js' );
```



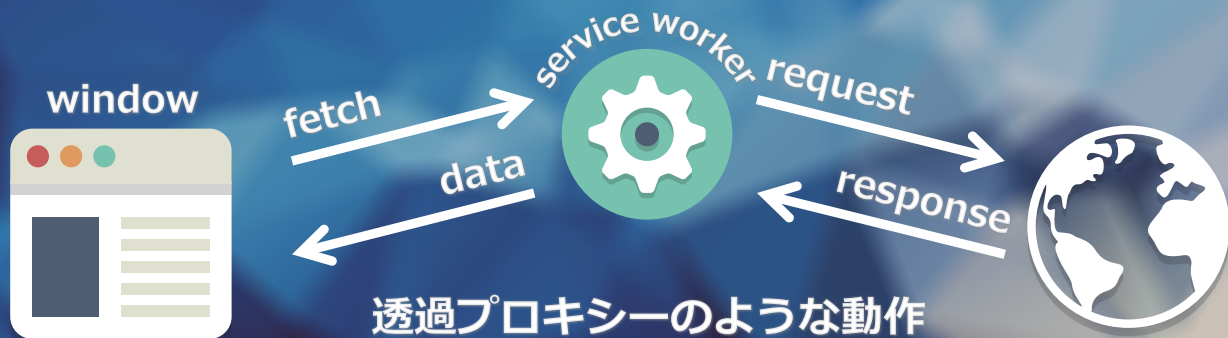
service worker起動

ブラウザのプロセスが活着ている限り、
ページが閉じてもバックグラウンドで動作！

Service Worker

▶ Caching

windowからのFetchを横取り

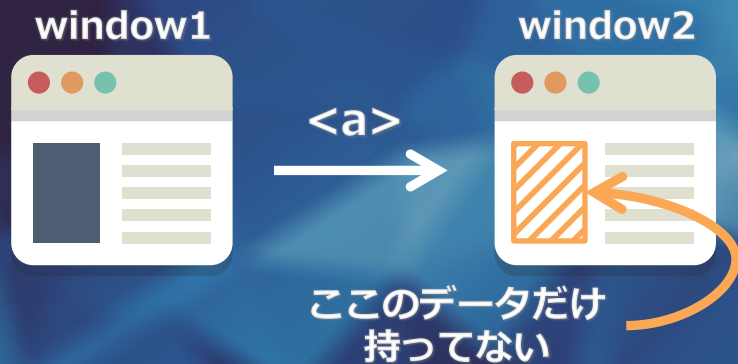


キャッシュを持っていたらRequestすら送らない

Service Worker

▶ Caching

window1 から <a> でwindow2へ遷移 (navigate) したとき



Service Worker

▶ Caching

window1 から <a> でwindow2へ遷移 (navigate) したとき



Service Worker

- ▶ 既に閉じたページからのPush通知
- ▶ offlineで編集したデータをOnlineになったら勝手にSubmit
- ▶ JavaScriptのmainスレッドと別スレッドかつ別LifeCycle

————→ 重い計算を移譲できる

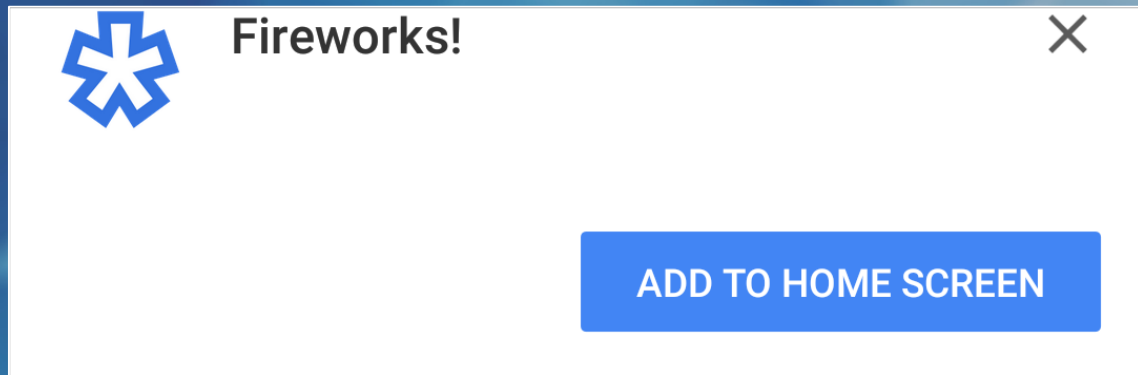
service worker



& more

Service Worker

▶ Android Chromeだと、アプリのようにインストールもできる





おわりです、ありがとうございました！