

跨平台移动Web AR的关键技术 介绍及应用

黄亚坤

2021.07.24

网络与交换技术国家重点实验室
北京邮电大学

目录

1

AR发展动态及新的趋势

2

Web AR关键技术及应用

3

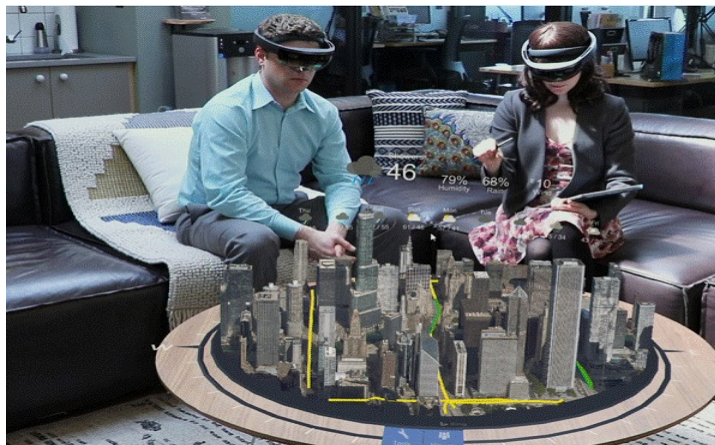
Web AR相关标准化进展

现有AR解决方案

❖ 基于专用头显/眼镜设备的AR（重量级）解决方案

- 虚实融合体验好 • 沉浸感强
- 价格昂贵 • 便携性差 • 移动性差

这种方案目前适用于固定场所的展示与体验，无法实现大规模移动化普及推广和应用

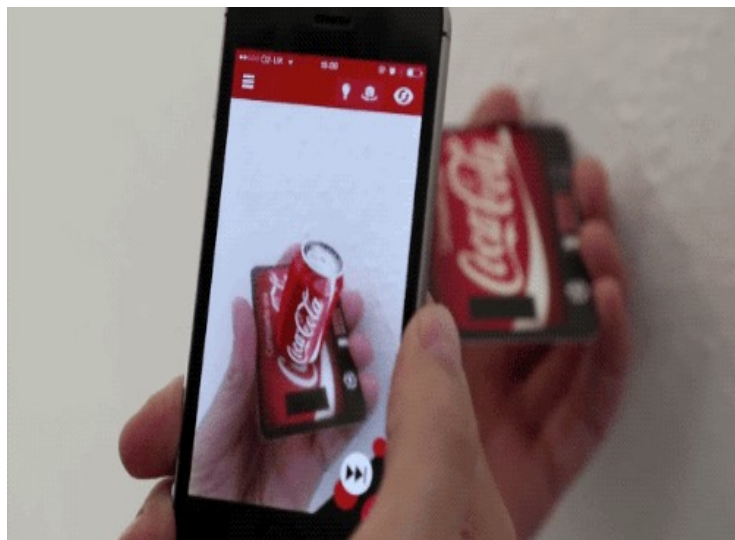


智能手机终端逐渐成为一种AR/VR普适化的终端

现有AR解决方案

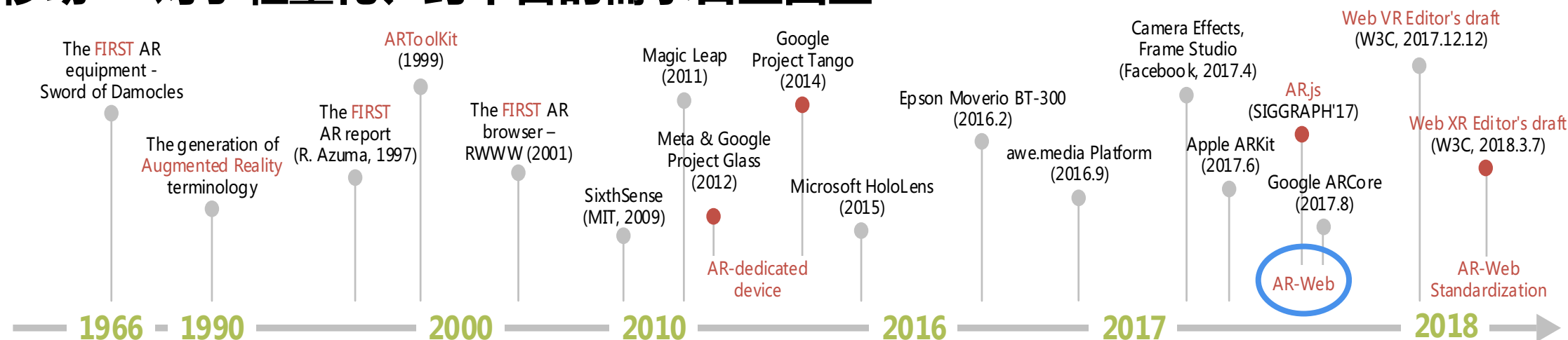
❖ 基于App的AR解决方案

- 普适性得到明显改善，虚实融合效果比较好，沉浸感差一些
- （独立App的模式）**用户准入门槛高**（需要额外的下载与安装）->**推广成本高**
- （搭载到主流App的模式）**跨平台性差，跨App差**



AR的发展历程和趋势

移动AR对于轻量化、跨平台的需求日益凸显

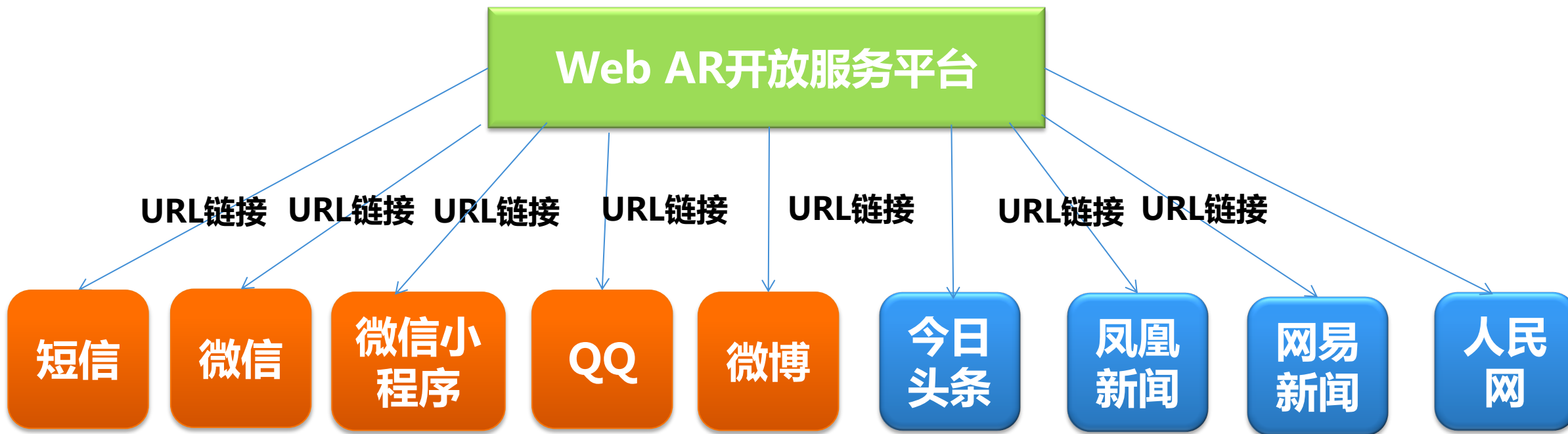


Web 由于具有天然的跨平台性，因此，基于Web的AR技术开始萌芽

- 1998 AR 首次应用于电视直播 (1st & Ten)
- 2012 AR 专用头显与移动终端 (Google Glass, Project Tango)
- 2016 App-based AR 游戏 (Pokémon Go , 8周超过5亿下载量)
- 2017 基于Web的AR解决方案 (AR.js , WebARonARKit & WebARonARCore)

轻量级、普适化的跨平台移动Web AR

轻量级、普适化的跨平台移动Web AR技术及开放服务



目录

1

AR发展动态及新的趋势

2

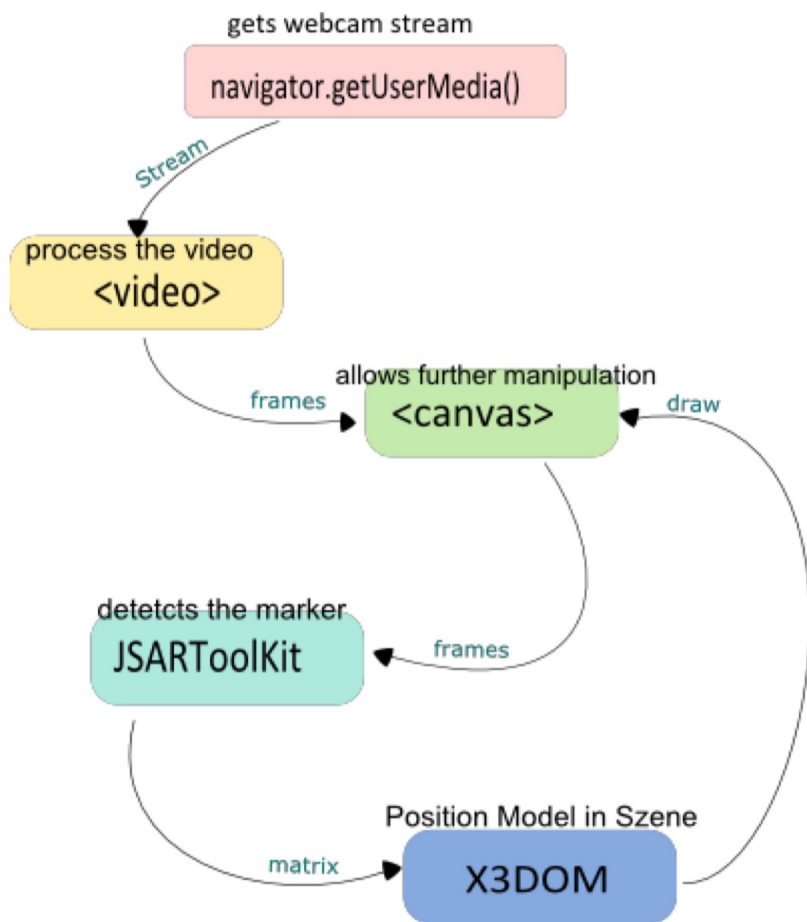
Web AR关键技术及应用

3

Web AR相关标准化进展

Web AR的技术路线之一

❖ 纯前端解决方案



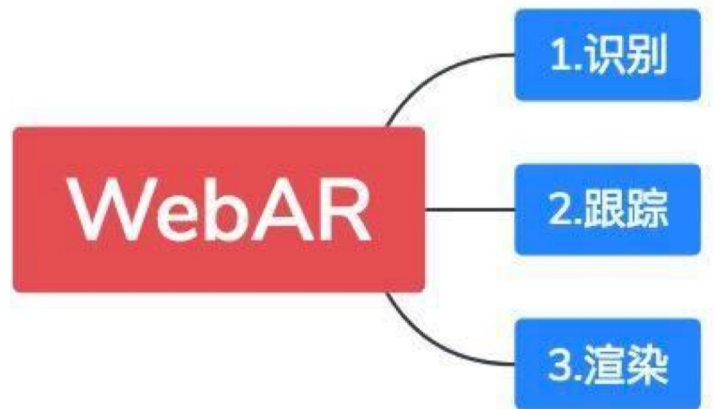
以JSARToolKit为例:

- 使用WebRTC获取摄像头信息，然后在canvas画布上绘制原图;
- JSARToolKit计算姿态矩阵，进而渲染虚拟信息



Web AR的技术路线之一

❖ 纯前端解决方案



核心步骤：识别、跟踪、3D 物体的渲染。

- （识别）WebRTC默认获取的是前置摄像头，如果想获取后置摄像头的视频流，需要用navigator.mediaDevices.enumerateDevices() 遍历设备 camera 得到。且要用 https 打开网页才能访问摄像头；
- （跟踪）Tracking.js、JSFeat、ConvNetJS、deeplearn.js、keras.js；
- （渲染）A-Frame、Three.js、Pixi.js、Babylon.js。



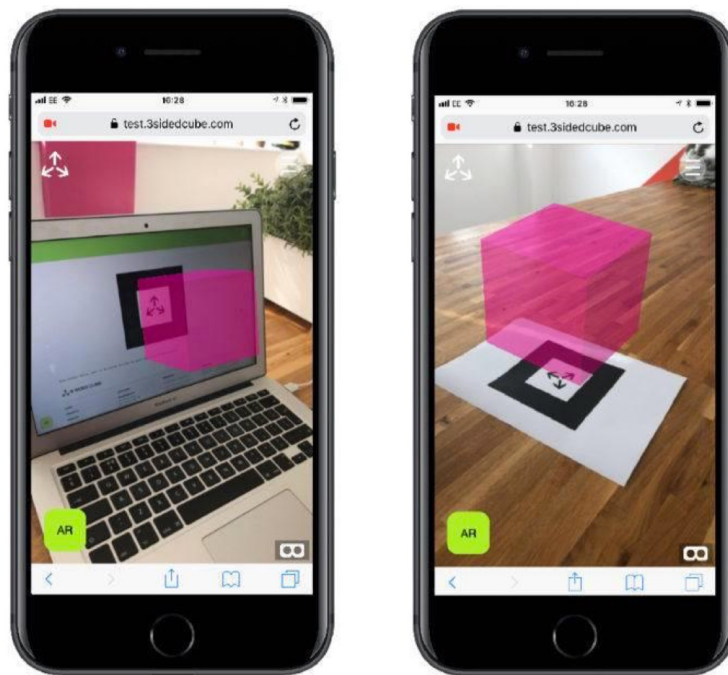
Web AR的技术路线之一

❖ AR.js

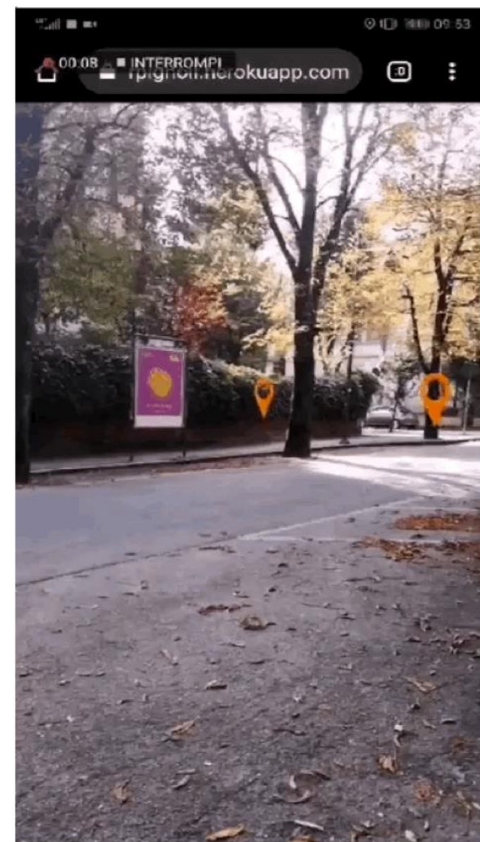
优点：

- 跨浏览器兼容性
- 达到 60fps 的高性能
- 基于 Web，无需安装
- 使用 WebGL 和 WebRTC
- 无需额外或不常见的硬件
- 可以用不到 10 行 HTML 来完成

- 基于标记的AR，通过扫描识别图出现AR模型效果。



- 基于位置的AR，通过使用手机传感器的方向和位置，在相机上显示其“物理”位置上每个位置的内容。



Web AR的技术路线之一

❖ JSARToolKit (ARToolkit的js版本, 支持pat marker和nft marker的识别与跟踪)



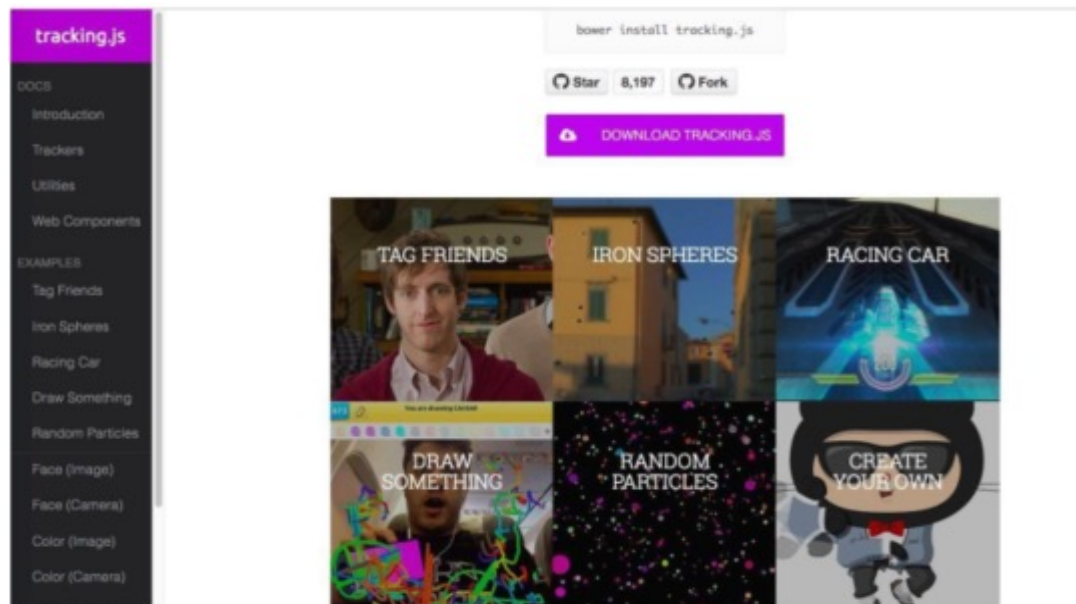
运行机制：

- JSARToolKit 在画布元素上运行
- 将画布传递给 JSARToolKit 进行分析
- 在 WebGL 场景中绘制视频帧并在其上方绘制对象
- 对每一帧都执行此操作，就可以进行视频 AR 跟踪

JSARToolKit的性能：

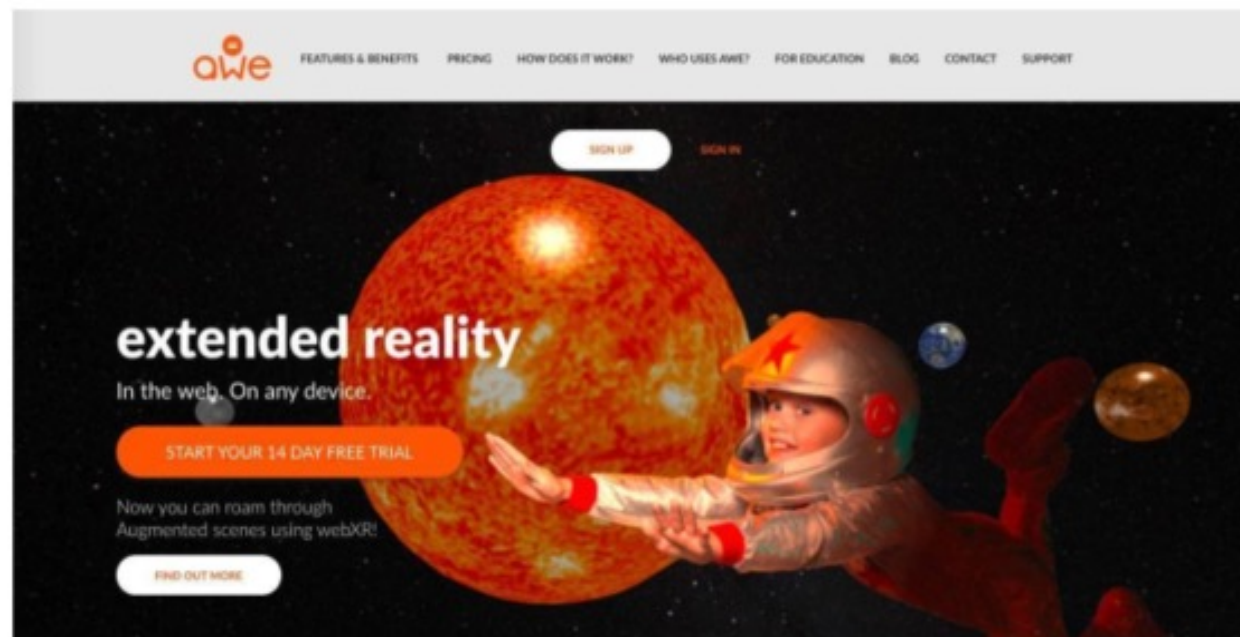
	NoWasm (ms)	WASM (ms)	Gain
Start videostream	1134.1	671.1	41%
Test time	3431.5	3039.5	11%
Load marker	60.80	54.70	10%
Average video process	22.16	23.05	-4%
Cleanup time	13.60	3.40	75%

Web AR的技术路线之一



❖ tracking.js

tracking.js 库将不同的计算机视觉算法和技术引入浏览器环境。通过使用现代 HTML5 规范，使开发者能够进行实时颜色跟踪，面部检测等。

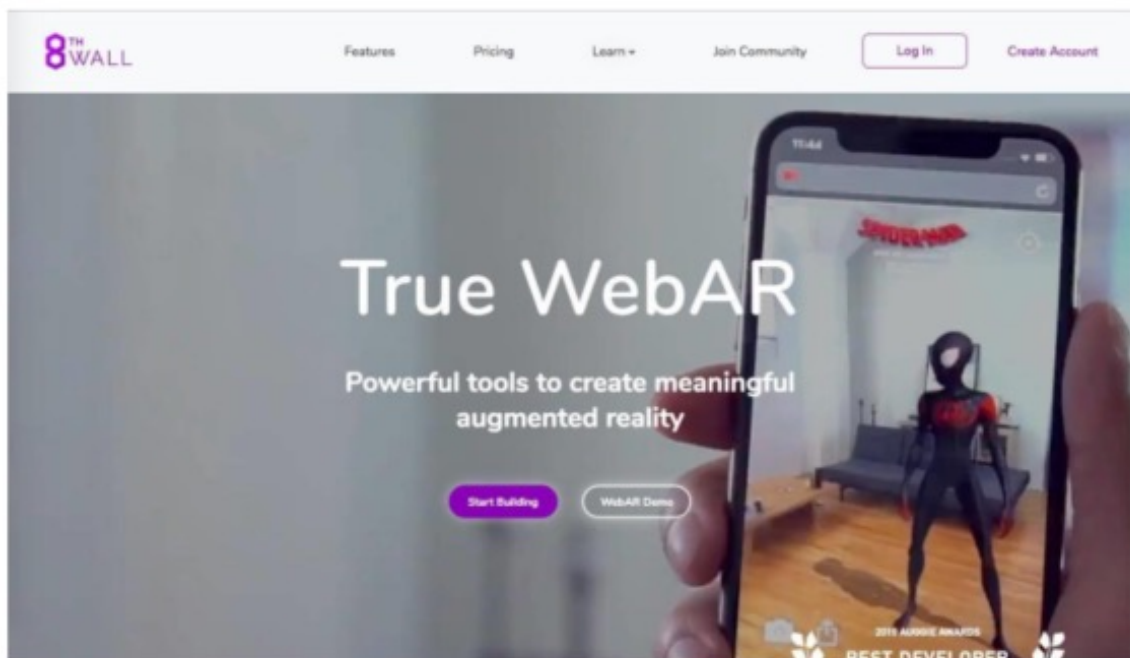


❖ awe.js

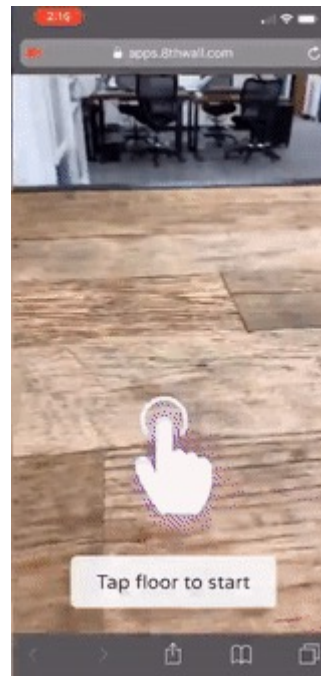
awe.js 提供了一些其他类型的 AR 体验，如增强现实标记，基于位置，和 leap motion 传感器 AR。

Web AR的技术路线之一

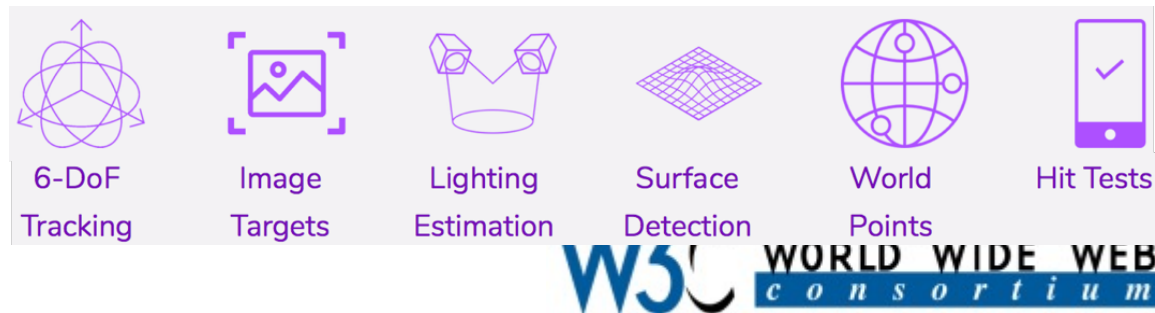
❖ 8th Wall



- 8th Wall Web 使用符合标准的JavaScript 和 WebGL 构建，加入了同步定位和 SLAM 功能，并针对移动浏览器上的实时 AR 进行优化。

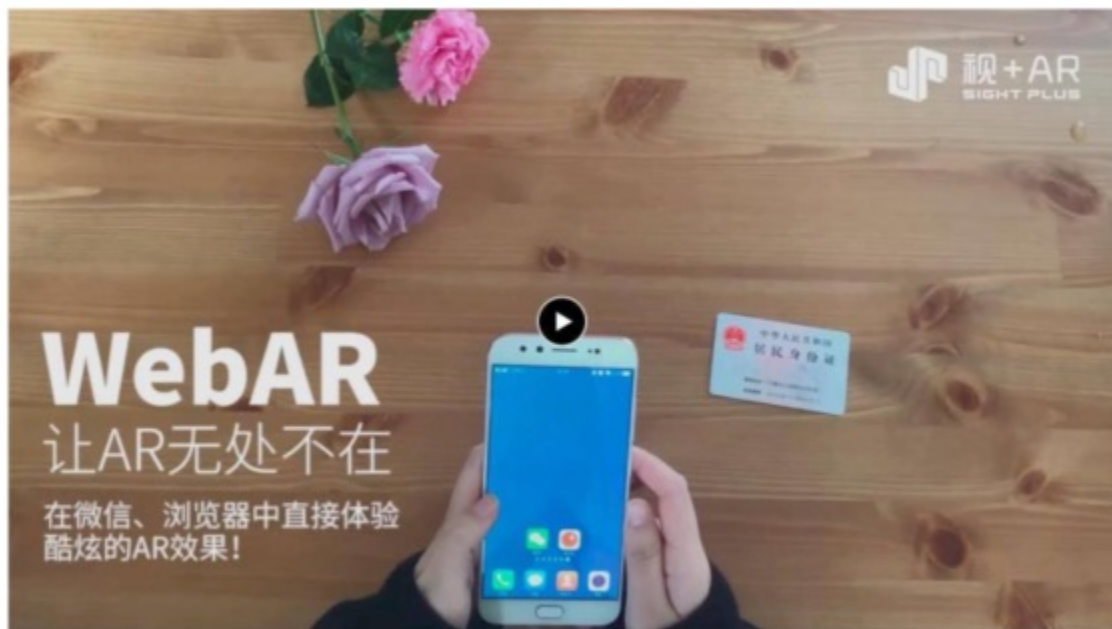


8th Wall SLAM 引擎：



Web AR的技术路线之一

❖ EasyAR



- 自主 AR 引擎：EasyAR 引擎针对微信小程序进行算法深度优化，只为打造卓越小程序 AR 体验
- Web3D 引擎算法深度优化：网络三维引擎针对小程序进行算法深度优化，实现极致的 3D 渲染效果，打造逼真的 AR 体验
- 大数据&精准营销：与腾讯深度合作，为品牌提供大数据和精准营销服务
- 极速开发，快速落地：凭借领先技术和丰富经验，视+ AR 可实现快速开发，即时发布WebAR 项目

Web AR的技术路线之一

❖ AR Quick Look

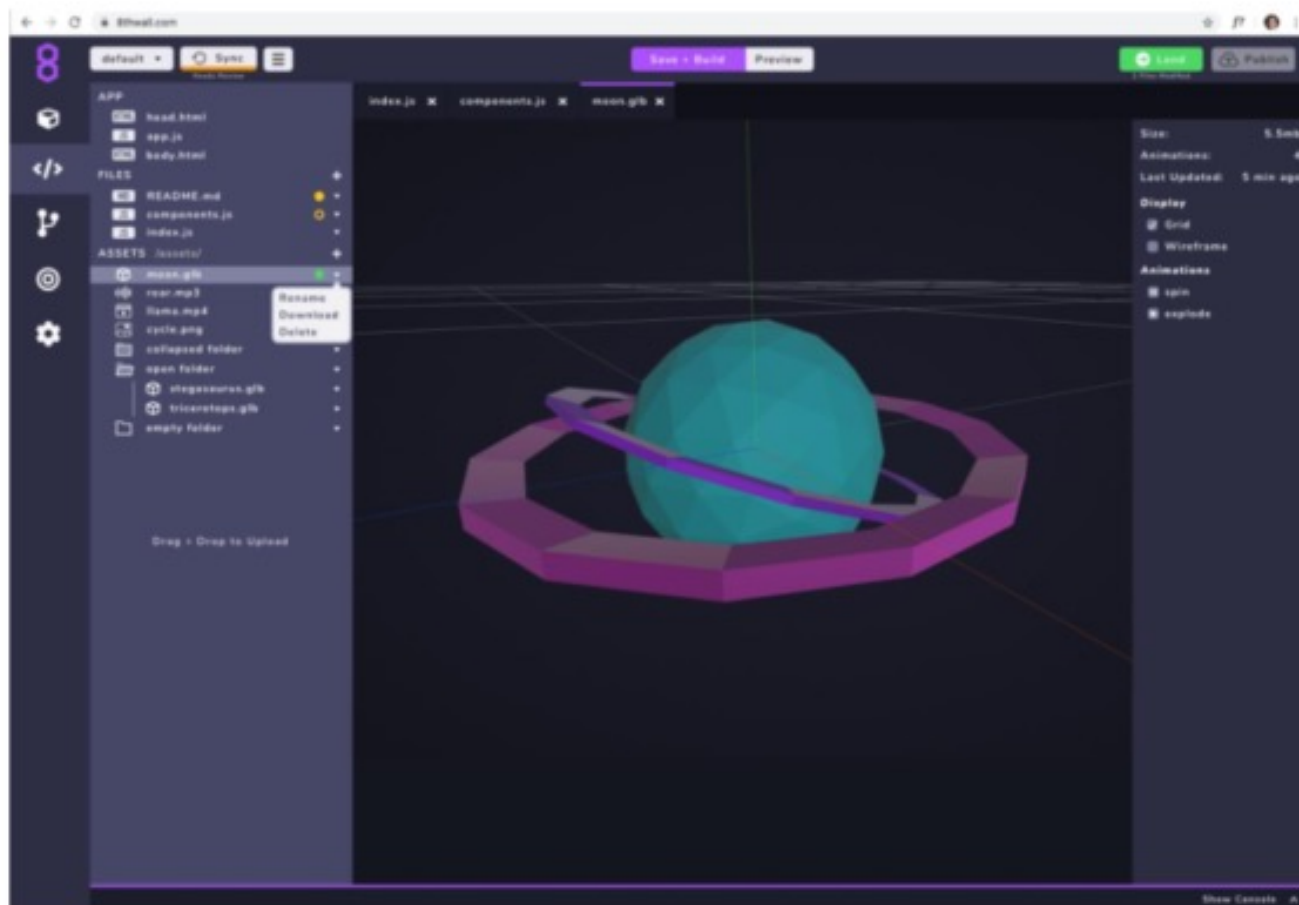
- AR Quick Look 是用于在 AR 场景中预览 3D 模型的技术，通过它开发者能快速在自己的 App 中展示高质量的 3D 内容。
- ARKit 作为 iOS 11 的一部分首次亮相，使开发人员可以轻松地将 AR 体验添加到他们的应用程序中。



Web AR的技术路线之一

❖ 8th Wall即时开发协作工具

- 8th Wall 创建了一个端到端云解决方案，用于创建、协作和即时发布基于浏览器的 WebAR 项目。
- 通过对模板的访问，跨团队工作的能力，以及内置的托管，开发人员能够快速地将大规模的 AR 体验带到市场中。



Web AR的技术路线之二

❖ 扩展浏览器内核的解决方案

在浏览器内核层面扩展，将增强现实计算密集算法嵌入到扩展内核中，在上层提供JavaScript接口。对一般开发者而言，不必关心内部算法实现。

目前还处于开发和内测阶段，而且各个浏览器之间的上层应用接口不统一；也无法跨平台的使用

单靠浏览器的话，计算力还是有限、电池能耗也是问题；



腾讯的TBS AR计划

W3C WORLD WIDE WEB
consortium

Web AR的技术路线之二

❖ WebARonARKit and WebARonARCore



❖ WebARonARKit和WebARonARCore底层使用ARKit和ARCore作为AR引擎，上层封装Web接口



Web AR的技术路线之二

❖ 扩展标签的解决方案

- 在浏览器中扩展标签，唤起本地的AR服务。浏览器仅充当入口的作用。苹果的AR Quick Look即此方案，使用QLPreviewController做控制器。

目前体验效果好，苹果、华为等手机厂正探索该方案。



Web AR的技术路线之二

❖ WebXR Device API

Web XR Device API 与WebVR的两个核心区别：

- 为了支持各种用户输入（例如语音和手势），为用户提供了在虚拟空间中进行导航和交互的选项
- 为开发AR体验建立技术基础，让创作者将真实世界的媒体与提升体验的上下文覆盖层进行整合。

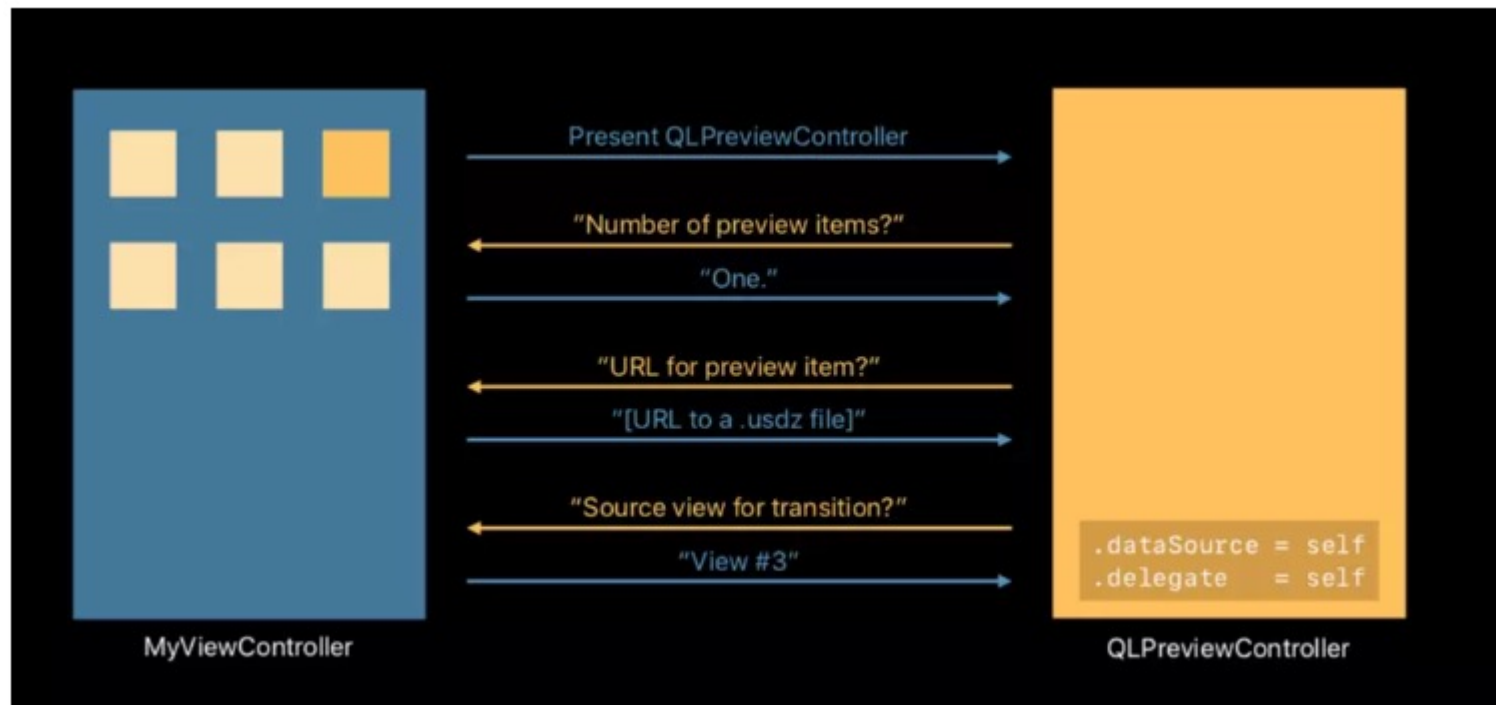
- ❖ Mozilla WebXR团队创建了一个与Firefox和Chrome兼容的WebXR API仿真器浏览器扩展程序，该扩展程序模拟了WebXR API，模拟了各种兼容设备，例如HTC Vive，Oculus Go和Oculus Quest，Samsung Gear。



Web AR的技术路线之二

❖ AR Quick Look原理简介

- AR Quick Look的核心控制器是QLPreviewController。通过和此控制器多次交互，实现从浏览器到AR Quick Look的跳转。



QLPreviewController原理

Web AR的技术路线之三

❖ 5G时代下 “端+边缘” 融合的解决方案

优势：充分利用云端服务器的计算力和渲染力，弥补Web前端计算力弱的缺点，提高服务的性能；

缺陷：当前4G网络的时延大，难以达到AR所要求的实时跟踪的能力；

“云计算+端计算” 的前后端分离解决方案

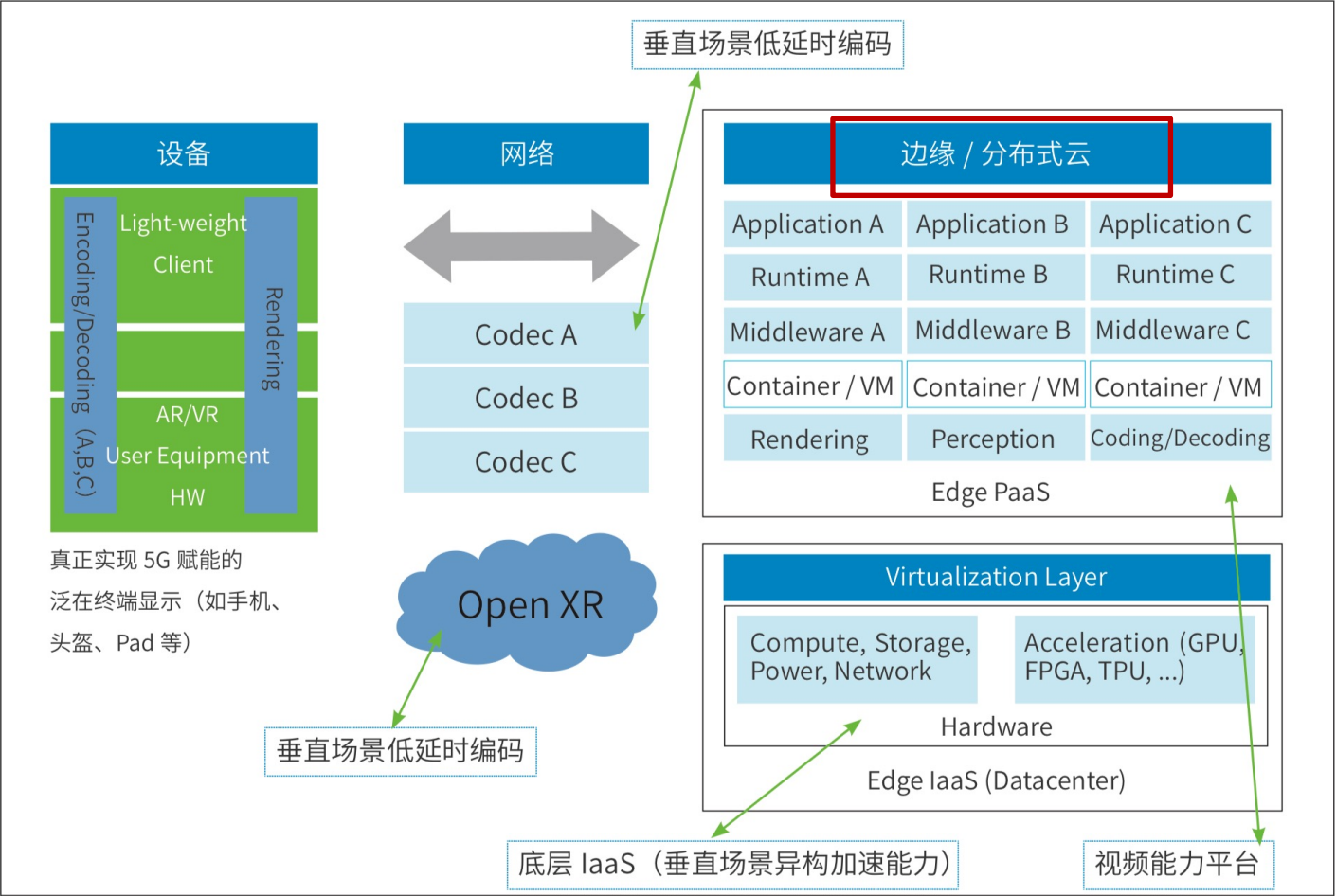


可以满足自然图片/3D物体识别的需求，但难以提供实时跟踪能力

Web AR的技术路线之三

❖ 5G网络中基于边缘云的 AR 系统构架图

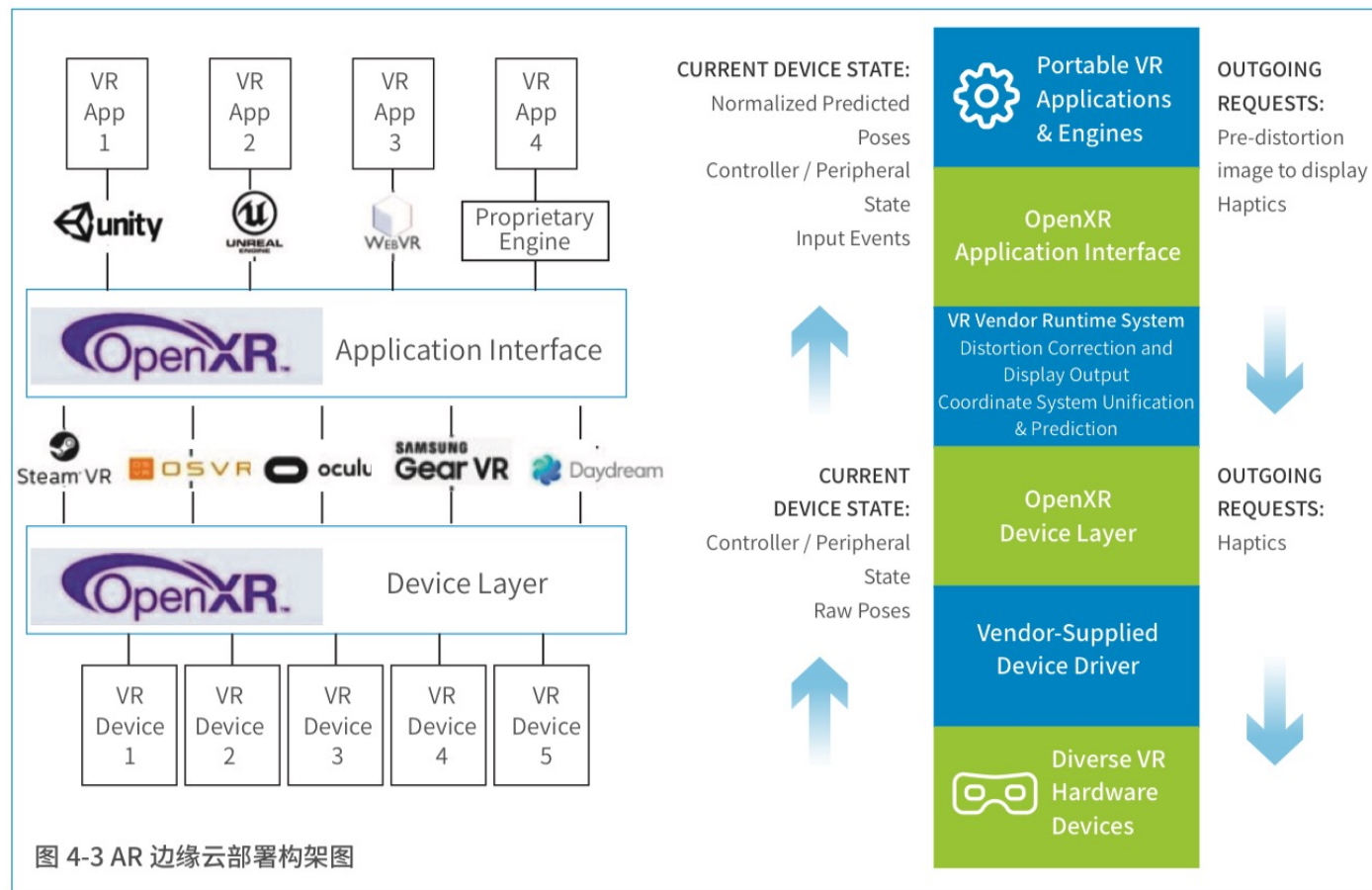
- 低时延编码，云能力赋能终端提供更有吸引力客户体验（如延时和清晰度等指标）。
- 依托 5G 空口低延时，以及高带宽完成待感知现实场景关键数据上传和显示结果下传
- 通过云端强大的服务能力完成更高体验质量的可视化融合展示效果



Web AR的技术路线之三

❖ 一致性能力集成接口

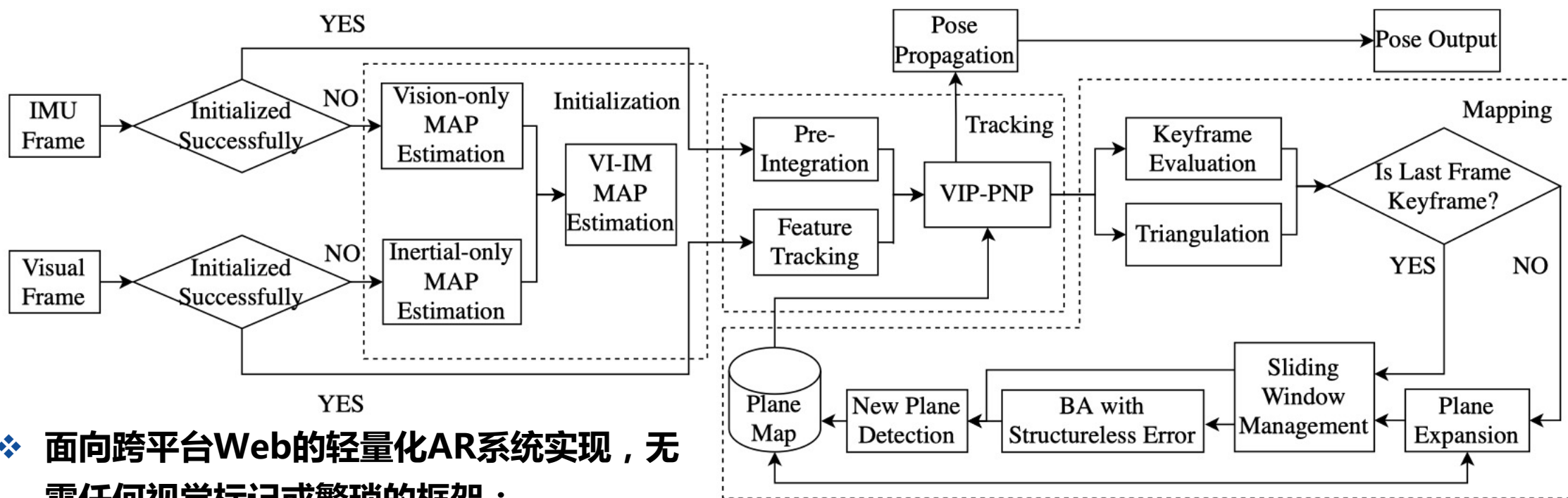
- 实现 5G 为核心和入口的庞大设备群落的体验一致，在之前面临着设备能力，渲染引擎能力接口和开发环境调用接口碎片化的问题。
- Khronos 最新为 VR 和 AR 应用程序和设备创建一个开放且免版税的标准，OpenXR，并发布了标准版本。



Web AR面临的挑战

- ❖ **浏览器弱计算力 vs. AR密集的计算要求**
- ❖ **网络时延 vs. 实时跟踪要求**
- ❖ **面向PC/APP重量级3D渲染技术 vs. 轻量级的Web渲染环境**
- ❖ **浏览器的功能支持程度参差不齐**

轻量化、松耦合的Web AR框架



❖ 面向跨平台Web的轻量化AR系统实现，无需任何视觉标记或繁琐的框架；

❖ 提出了一种单线程视觉惯性里程计（VIO）算法来避免复杂的多线程管理，以适应不同的浏览器，进一步提高效率

■ Yakun Huang, Xiuquan Qiao, et al. WebPoseEstimator: A Fundamental and Flexible Pose Estimation Framework for Mobile Web AR[C]//2021 IEEE Conference on Virtual Reality (VR 2021). IEEE, 2021: 478-479.

■ Zhijie Tan, Xiuquan Qiao, Yakun Huang, et al. WebARS: Enabling the Markerless Augmented Reality System on the Web [C]//IEEE International Symposium on Mixed and Augmented Reality. (ISMAR 2021). (Under Review)

轻量化Web AR助力冬奥全球化传播

❖ 轻量化WebAR跟踪技术

❖ 多线程

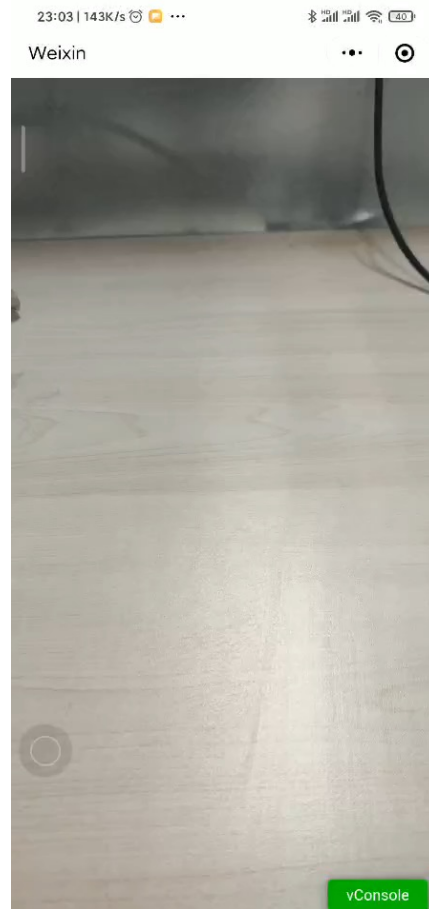
- 将计算任务剥离主线程，保证主线程不会因为计算任务过重而阻塞

❖ IMU数据缓存

- 在Worker Thread计算时，由于IMU高频的特性，会产生一定的数据丢失

❖ World Analysis模块冗余设计

- VIO模块长时间没有返回时，主线程的轻量化姿态估计会接管，保证服务稳定性



放置并转动视角



在平面上行走

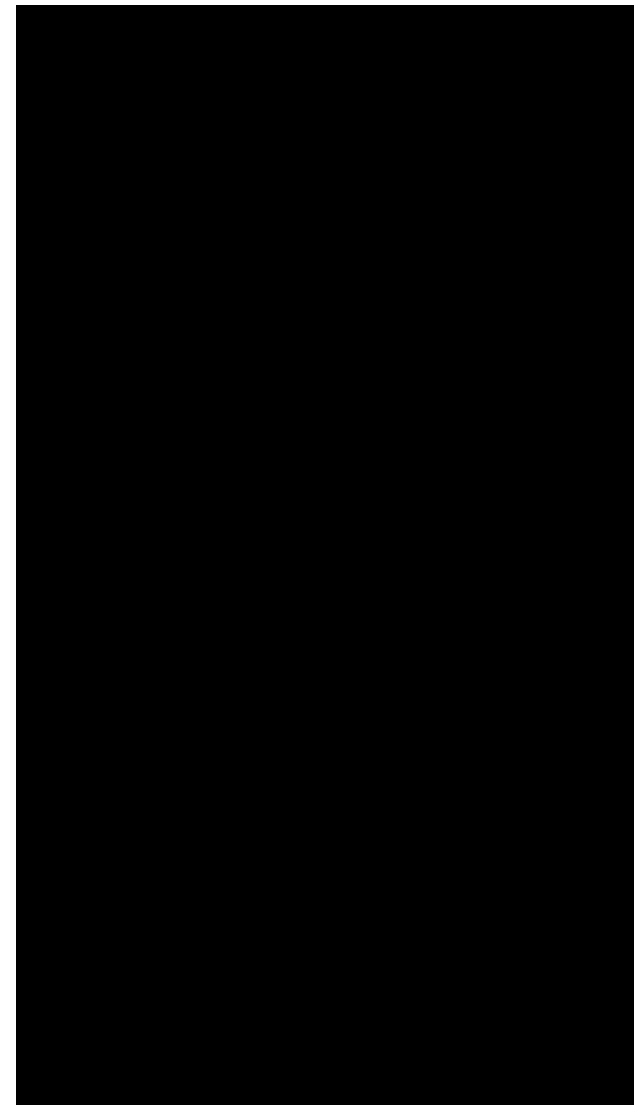
针对Web弱计算环境和有限的底层API支持重新设计、优化的AR架构

轻量化Web AR助力冬奥全球化传播

❖ 实时平面位姿跟踪技术

核心特色

- 通过角点坐标估计平面形变，
获得实时平面位姿信息；
- 模型能够垂直于平面；
- 视频能够完全贴合；
- ...



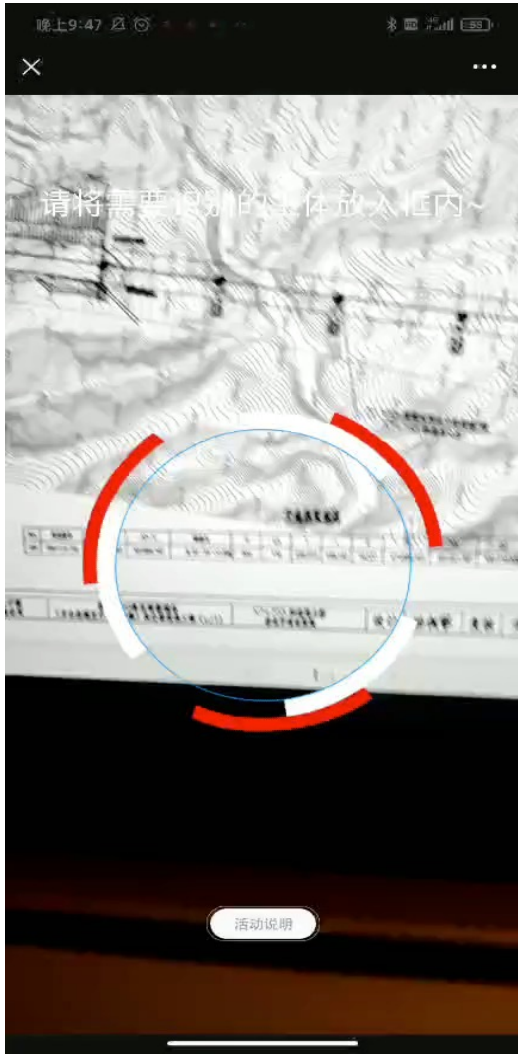
轻量化Web AR助力冬奥全球传播

❖轻量化WebAR渲染技术

大场景模型分块加载

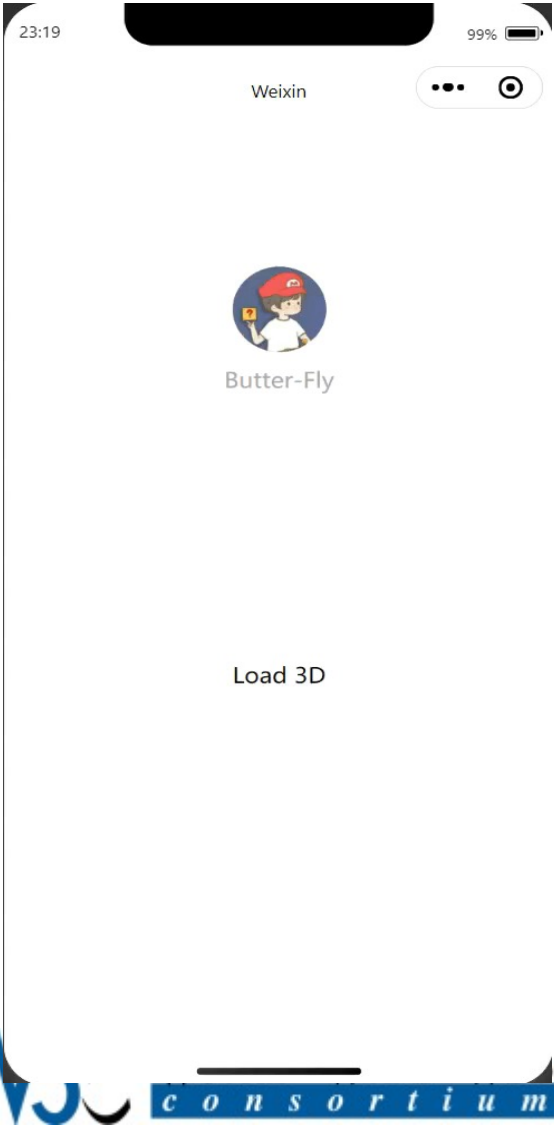
 -46_-39_23.glTF	34.21KB
 -46_-39_24.glTF	35.86KB
 -46_-39_25.glTF	2.39MB
 -46_-40_24.glTF	4.06KB
 -46_-40_25.glTF	2.17MB
 -47_-39_23.glTF	66.59KB
 -47_-39_24.glTF	52.05KB
 -47_-39_25.glTF	6.27MB
 -47_-40_23.glTF	50.41KB
 -47_-40_24.glTF	133.41KB
 -47_-40_25.glTF	9.54MB
 -47_-41_24.glTF	34.22KB
 -47_-41_25.glTF	4.11MB
 -48_-39_23.glTF	34.23KB
 -48_-39_24.glTF	18.03KB
 -48_-40_24.glTF	34.53KB
 -48_-40_25.glTF	5.04MB
 -48_-41_24.glTF	34.23KB
 -48_-41_25.glTF	1.98MB

原ifc模型 175MB
分块结果如图所示



模型和动画分离渲染

- ❖ 模型和动画分离使得同一模型需多次用到的场景下该模型只需加载一次，动画**按需加载**，提高了模型的**复用能力**;
- ❖ 当模型包含多个动画序列时，将模型和动画分离还能一定程度上改善模型的**加载速度**，提升用户体验。



❖ Android Chrome的IMU频率和精度太低

- IMU的频率被限制在60Hz内
 - 60Hz意味着Web每秒获取60组IMU数据，而VINS mono要求频率至少为200hz才能正常启动
 - 若开发人员试图在获取超过60Hz频率的IMU数据，则会抛出异常 “Maximum allowed frequency value for this sensor type is 60 Hz ”
- 加速度精度不足
 - 对于最新版本，从Android Chrome 82开始，IMU的加速度读数仅限于一个小数点
- 我们在以下Android Chrome版本上测试了IMU频率和精度读数：
 - Chrome Beta 92.0.4515.101 (The newest version at the moment)、Chrome 89.0.4389.105、Chrome 86.0.4240.75、Chrome 84.0.4147.89、Chrome 83.0.4103.96 、Chrome 82.0.4077.2 、Chrome 81.0.4044.138、Chrome 76.0.3809.111

目录

1

AR发展动态及新的趋势

2

Web AR关键技术及应用

3

Web AR相关标准化进展

Web前端相关标准

❖ WebRTC (W3C)

WebRTC 1.0: Real-time Communication Between Browsers

W3C Recommendation 26 January 2021

<https://www.w3.org/TR/webrtc/>

❖ Web Workers (W3C)

W3C Working Group Note, 28 January 2021

<https://www.w3.org/TR/workers/>

❖ WebAssembly (W3C)

WebAssembly Core Specification

W3C Recommendation, 5 December 2019

<https://www.w3.org/TR/wasm-core-1/>

❖ WebGL (Khronos)

WebGL 2.0 Specification

Editor's Draft Fri Dec 4 15:13:39 2020 -0800

<https://www.khronos.org/registry/webgl/specs/latest/2.0/>

Web AR相关的标准化情况

❖ Web前端相关的标准需求

■ 网页开启摄像头的需求

- WebRTC协议标准 (**W3C**)
- 主流浏览器都已经支持，包括Safari11以上、Chrome、QQ浏览器、百度浏览器等
- APP内嵌式Web内核引擎和手机系统自带浏览器目前支持情况还不是很好

■ Web 3D绘图协议：WebGL

■ Web浏览器性能加速的需求

- WebAssembly (W3C)：WebAssembly是一种可以使用非JavaScript编程语言编写代码（如C/C++等）并且能在浏览器上运行的技术。
- Web Workers (W3C)：Web的多线程技术

Web AR相关的标准化情况

❖ 扩展浏览器功能的解决方案

❖ 优势：

- 能够直接调用终端设备硬件资源，AR业务性能表现优于纯前端的解决方案

❖ 缺陷：

- 当前各个浏览器厂商提供了不同的AR内核扩展机制，互不兼容

❖ 当前进展：

目前包括谷歌 (**WebARonARKit, WebARonARCore**)，Mozilla (Web XR)，百度 (DuMix AR)，腾讯 (TBS AR)，360，UC等都提出了各自基于浏览器扩展内核的AR解决方案，通过扩展自身浏览器的内核添加AR的功能，对外来提供JavaScript接口，但目前都还处于实验开发阶段

Web AR相关的标准化情况

❖ 扩展浏览器功能的解决方案

❖ 标准化方向：

- ❖ 针对AR业务的相关浏览器功能模块扩展标准

- ❖ **上层统一的JavaScript API标准**

❖ 现有扩展方案：

- ❖ 通过嵌套具备AR功能的SDK

❖ 方案缺陷：

- ❖ SDK具有差异性，浏览器内核与SDK的嵌套方式也有所差异导致了浏览器间的兼容性较差，导致AR服务无法跨平台体验

Web AR相关的标准化情况

❖ W3C

- **WebXR Device API Editor's Draft** (2020/07/24)
 - This specification describes support for accessing virtual reality (VR) and augmented reality (AR) devices, including sensors and head-mounted displays, on the Web.

TABLE OF CONTENTS

1	Introduction
1.1	Terminology
1.2	Application flow
2	Model
2.1	XR device
3	Initialization
3.1	navigator.xr
3.2	XRSystem
3.3	XRSessionMode
3.4	Feature Dependencies
4	Session
4.1	XRSession
4.2	XRRenderState
4.3	Animation Frames
4.4	The XR Compositor
5	Frame Loop
5.1	XRFrame
6	Spaces
6.1	XRSpace
6.2	XRReferenceSpace
6.3	XRBoundedReferenceSpace
7	Views
7.1	XRView

WebXR Device API

W3C Working Draft, 24 July 2020



This version:

<https://www.w3.org/TR/2020/WD-webxr-20200724/>

Latest published version:

<https://www.w3.org/TR/webxr/>

Editor's Draft:

<https://immersive-web.github.io/webxr/>

Previous Versions:

<https://www.w3.org/TR/2019/WD-webxr-20191010/>

Issue Tracking:

[GitHub](#)

Editors:

[Brandon Jones](#) (Google)

[Manish Goregaokar](#) (Mozilla)

Former Editor:

[Nell Waliczek](#) (Amazon [Microsoft until 2018])

Participate:

[File an issue](#) (open issues)

[Mailing list archive](#)

[W3C's #immersive-web IRC](#)

Copyright © 2020 W3C® (MIT, ERCIM, Keio, Beihang). W3C [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

This specification describes support for accessing virtual reality (VR) and augmented reality (AR) devices, including sensors and head-mounted displays, on the Web.

Web AR相关的标准化情况

❖ 扩展HTML5标签—Google Model Viewer

The screenshot shows the GitHub repository page for GoogleWebComponents/model-viewer. At the top, the repository name is displayed with navigation links for Watch (52), Star (916), and Fork (68). Below this is a navigation bar with links for Code, Issues (210), Pull requests (13), Projects (1), Wiki, Security, and Insights. A large banner promotes joining GitHub today, stating that GitHub is home to over 40 million developers and includes a 'Sign up' button. Below the banner, a description reads 'Easily display interactive 3D models on the web and in AR!' followed by a link to the demo. A series of tags are listed: 3d-models, gltf, model-viewer, glb, three-js, 3d, webcomponents, augmented-reality, magic-leap, and webxr. A statistics bar shows 473 commits, 59 branches, 12 releases, 16 contributors, and the Apache-2.0 license. At the bottom, there are buttons for 'Branch: master', 'New pull request', 'Find File', and 'Clone or download'.

GoogleWebComponents / model-viewer

Watch 52 Star 916 Fork 68

Code Issues 210 Pull requests 13 Projects 1 Wiki Security Insights

Join GitHub today

Dismiss

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

Sign up

Easily display interactive 3D models on the web and in AR! <https://googlewebcomponents.github.io...>

3d-models gltf model-viewer glb three-js 3d webcomponents augmented-reality magic-leap webxr

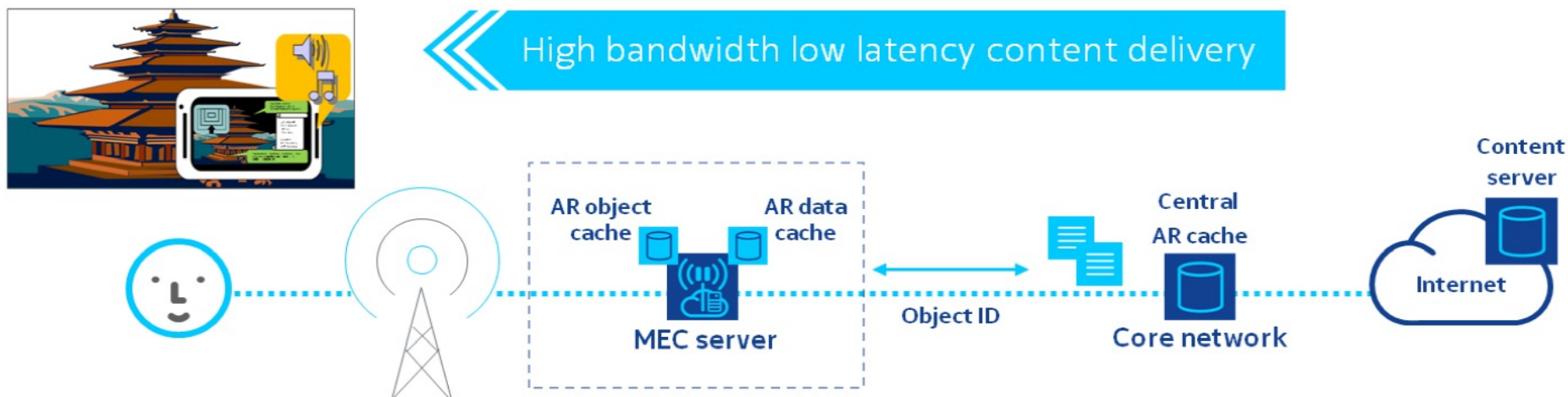
473 commits 59 branches 12 releases 16 contributors Apache-2.0

Branch: master New pull request Find File Clone or download

Web AR相关的标准化情况

❖ 5G时代协同计算技术方案（端+边缘+云）

❖ 现有加速方案（ETSI发布的5G-白皮书）：



❖ 标准化方向---待开展：

- ❖ 5G网络环境下“端+边缘+云”分布式协同的通用AR服务提供技术要求（Browser+MEC+Cloud AR）

谢谢！