

Web Remote App

**CREATING A VIRTUAL PRODUCTION TOOL IN FORM
OF AN WEB APPLICATION**

WHAT IS VIRTUAL PRODUCTION?

Virtual Production

- Cost
- Creativity
- Efficiency
- Safety

{Usual Production Pipeline IMG}

{Virtual Production Pipeline IMG}

PREVISUALIZATION

{IMG}

Tools for Previs (VP)

Hardware

- VR-Glasses
- AR-Glasses
- LED-Wall
- Screens(Renderer/PCs)

Software

- Realtime Engines (Unreal Engine/ Unity)
- VR-Scouting Tools
- AR-Scouting Tools
- Applications (Touch OSC)

WHY A WEB-APPLICATION

{IMG OF SET}

Web Remote vs Touch OSC

Touch OSC(as an example)

Pro

- fast so set up
- many components
- Creation-Kit (no coding needed)
- simple layout

Cons

- layout(style) limited
- limited to components
- => limited to functions => possibilities limited

conclusion => Bad for specific and special Use-Cases

Web Remote vs Touch OSC

Web Remote

Pro

- almost infinit possible components and functions
- perfect for specific and special use cases
- accesable from almost everywhere(devices)
- use of Websocket (possibility of more then one clien)
- catching of errors
- space for upgrading

Cons

- takes time to set up

REALTIME ENGINE

UNREAL VS UNITY



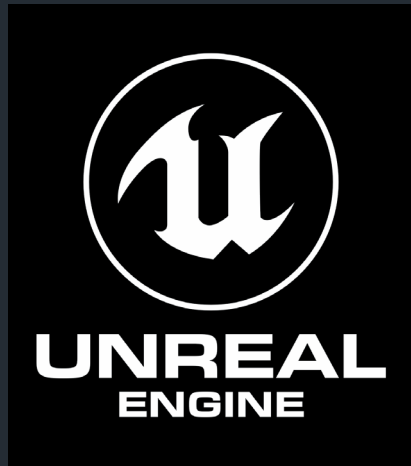


Pro

- more familiar C#
- bigger community
- used for Previs like Lion King

Contra

- no Multi-User Function
- No Advancement for virtuellen Produktionen
- not many VP Tools
- no Web Remote API



Pro

- Model Child for VP
- Multi-User available
- advancement in VP
- has VP Tools
- Quixel integrated

Contra

- not familiar with C++
- smaller Community

{DIAGRAM SHOW UE WORLD
AND EDITOR WOLRD}

Multi User-Diagram

React vs Vue

	<div> VUE</div>	<div> REACT</div>
Virtual DOM	✓	✓
Virtual component-based UI development DOM	✓	✓
Official component library for building mobile apps	✓	✓
Open source	✓	✓
Creator	Evan You	Facebook / Jordan Walk
Initial release	2014	2013
Written in	JavaScript	JavaScript
License	MIT	MIT
Syntax	HTML (default), JSX	JSX
Top features	<ul style="list-style-type: none">• Elegant programming style and patterns• Easy learning curve• Thorough documentation	<ul style="list-style-type: none">• Elegant programming style and patterns• Rich package ecosystem• Widespread usage
Popular applications	<div> </div>	<div> Uber NETFLIX</div>

React

- Harder to learn then Vue
- More developer for react on the market (bigger community)
- Most used JavaScript Technology
- Widespread usage
- Rich package ecosystem
- Elegant programming style and patterns

Vue

- Easier to learn then React
- Elegant programming style and patterns
- Good documentation

DIAGRAMS

{USE CASE DIAGRAM}

{COMPONENT DIAGRAM}

{SEQUENCE DIAGRAM}

WEBSOCKETS VS LONG POLLING

LONG POLLING

- Long polling is a lot more intensive on the server.
- Risk of losing messages

WEBSOCKET

- reduces the expensive data loads
- Risk of losing messages

SHOWING THE APP

Best Case

Worst Case

CONCLUSION

OUTLOOK