WEB DAY 2026

Building WebXR Experiences with React and Babylon.js

Simone De Vittorio
XR Technologist
Reactlyon

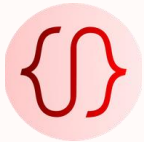
improove

WEB DAY 2026

Kudos

beSharp. we make IT run.

Sponsor



Partner

Paradigm Shift

2007: THE MOBILE SHIFT

ACCESSIBILITY AND PORTABILITY

Era of the rectangle: design for pocket-sized glass.

2026: THE SPATIAL SHIFT

INTELLIGENT, IMMERSIVE REALITIES

Era of the presence: design for the world around us.

improove

Escape from the "Flatland" Prison

- The 2D web is still trapped in rectangles and `<div>`
- We fake depth with shadows and parallax, but we're hitting a ceiling
- Huge data needs a new way to be represented

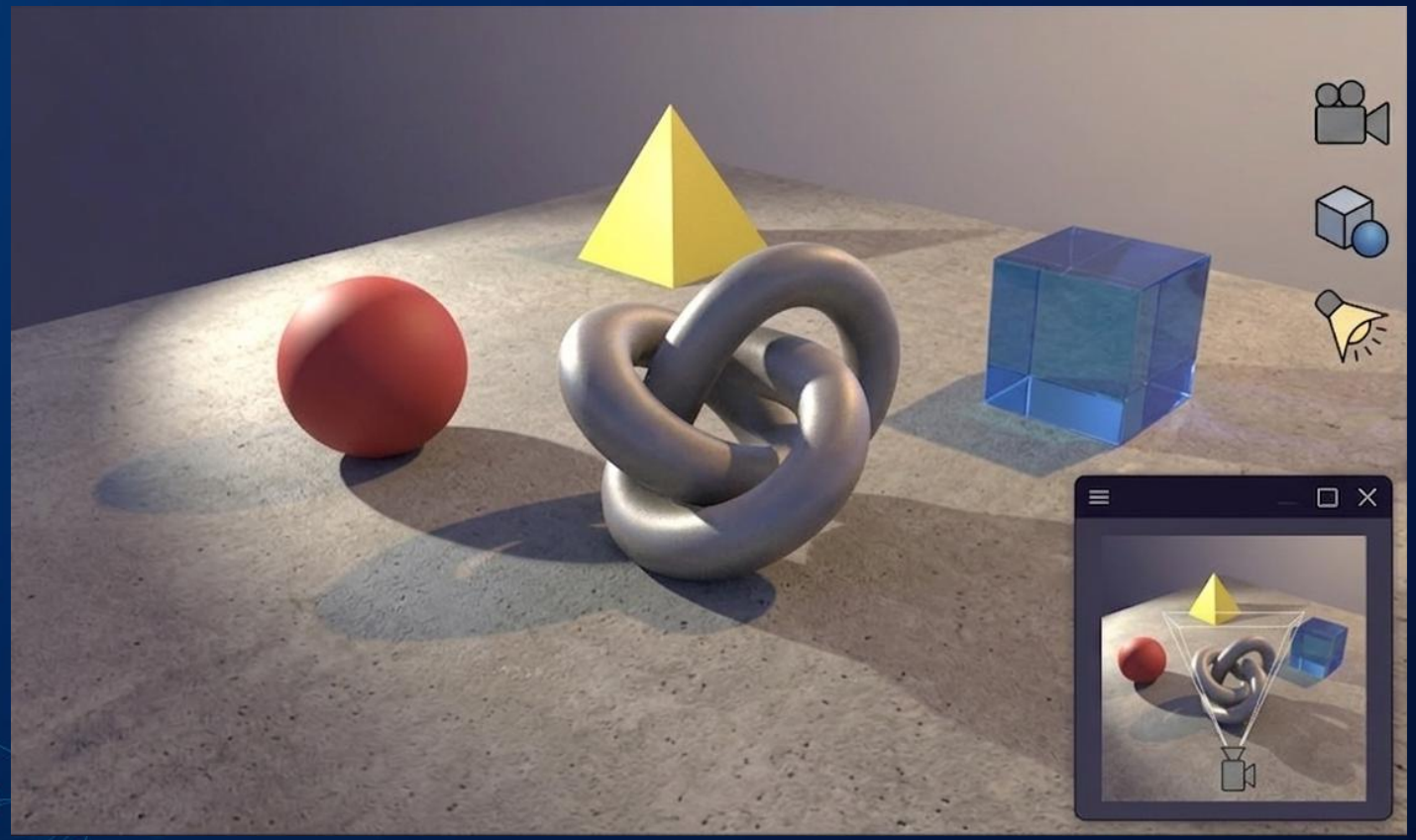
Humans live in 3D. Why don't our apps?

The Holy Trinity: WebGL, WebGPU & WebXR

- **WebGL**: The reliable foundation of browser 3D
- **WebGPU**: The next-gen leap for high-performance compute and graphics
- **WebXR**: The magic glue that connects our code to the lenses of an XR headset or smart glasses

The result: console-grade graphics, directly in a URL.

The Anatomy of a 3D World



Engine

↓

Scene(s) → global container(s)

- **Camera** → point of view
- **Lights** → illumination, shadows
- **Meshes** → 3D objects
- **Materials** → color, texture
- **Audio** → global, spatial
- **Physics** -> simulations
- **Animations**
- **Particles**
- **Post processes**
- **Events**
- **GUI**
- **XR Session**
- ...

Meet the Titan: Babylon.js

- **Graphic Engine**: Open-source, Microsoft-backed, and feature-complete
- **Performance**: Optimized for desktop, mobile, and standalone XR devices
- **Versatility**: NASA, CAD, indie games, e-commerce, portfolio
- **Cross Platform**: Web, PWA, mobile, and VR/AR/MR headsets

The 3D Friction

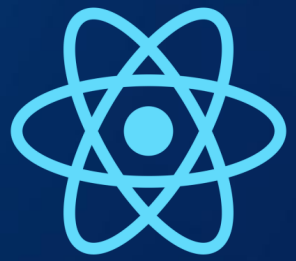
- **Manual lifecycle management**: Every object must be created, updated, and disposed explicitly, making code repetitive, and error-prone
- **Poor Scalability**: What starts simple quickly turns into a tangled scene graph that is difficult to understand, evolve, and maintain
- **Developer Experience Gap**:
  - No declarative mental model
  - No native HMR: Changes require a full scene reload
  - Logic-Rendering Coupling

React to the rescue!

- **Declarative**: Describe *what* you want, not *how* to do it
- **Component & Hooks**: Encapsulate complex 3D behaviors (physics, shaders, interactions)
- **HMR**: Live edits without scene reload
- **Ecosystem**: State management, linting, testing, UI library

Reactylon: What is it?

REACT



UI library for building **component-based** interfaces.

+

BABYLON.JS

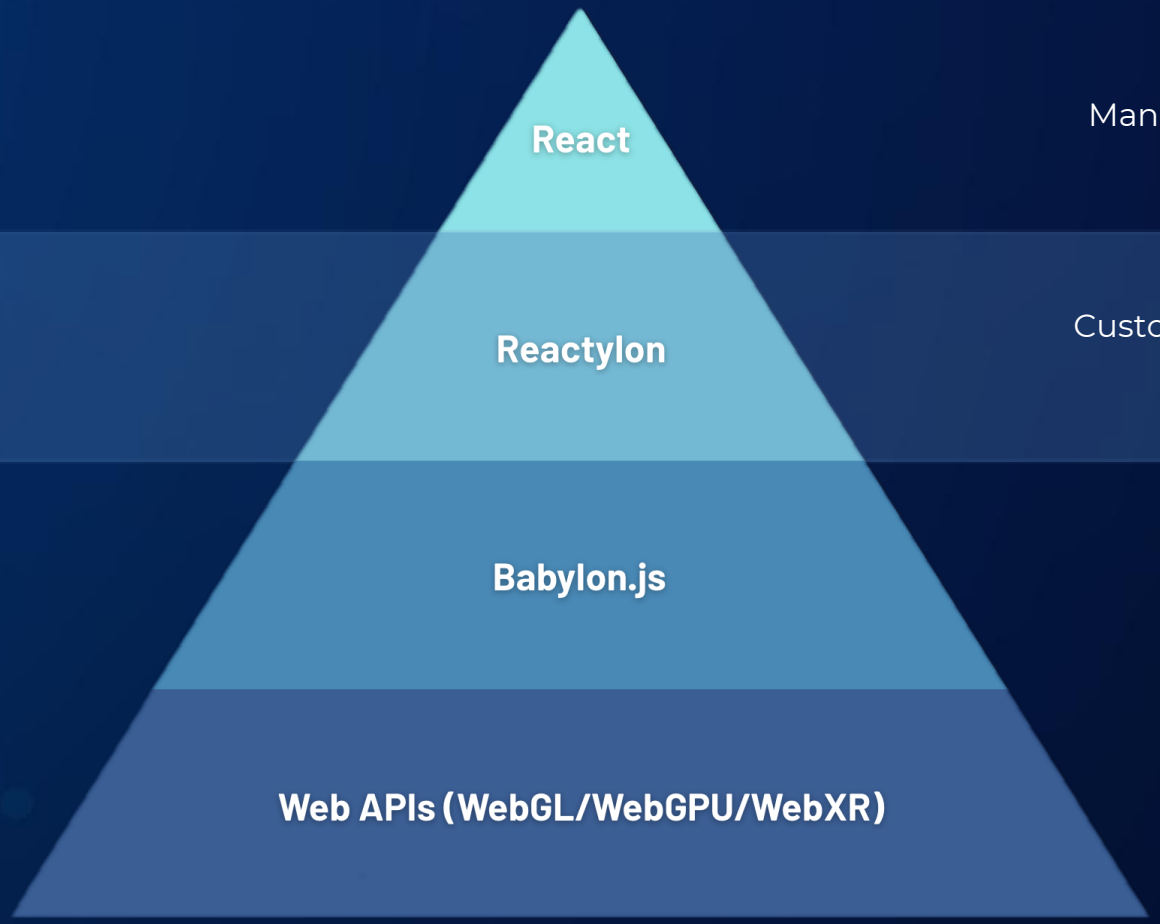
JavaScript engine for building and rendering real-time **3D** and **XR** experiences.

=

REACTYLON

An open-source multiplatform framework, designed to create interactive and immersive **3D/XR experiences**.

Where it fits



Manages components **state**, **props**, and **lifecycle** deciding when UI updates happen.

Custom **React renderer** that turns JSX into live 3D scene objects and manages their lifecycle.

High-level **3D engine** that owns scenes, cameras, lights, meshes, materials, and post-processing.

Browser's **low-level bridge** to GPU and XR devices.

improove

Reactylon: Why use it?

- Declarative **JSX syntax** for Babylon.js
- Full **TypeScript** support
- Automatic **objects management**
- Built-in **WebXR** support (VR/AR/MR)
- Interactive **documentation** (125+ live sandboxes)
- **Reactylon** (web) + **Reactylon Native** (desktop/iOS/Android/headset)

Babylon.js

```
const canvas = document.getElementById("renderCanvas");
const engine = new Engine(canvas, true);
const scene = new Scene(engine);

const camera = new UniversalCamera("camera", new Vector3(0, 5, -10), scene);
camera.target = new Vector3(0, 0, 0);
camera.attachControl();

const light = new HemisphericLight("light", new Vector3(0, 1, 0), scene);
light.intensity = 0.7;

const box = MeshBuilder.CreateBox("box", { size: 1 }, scene);
box.position.y = 1;

const material = StandardMaterial("material", scene);
material.diffuseColor = Color3.Red();
box.material = material;

// register a render loop to repeatedly render the scene
engine.runRenderLoop(function () {
    scene.render();
});

// watch for browser/canvas resize events
window.addEventListener("resize", function () {
    engine.resize();
});
```

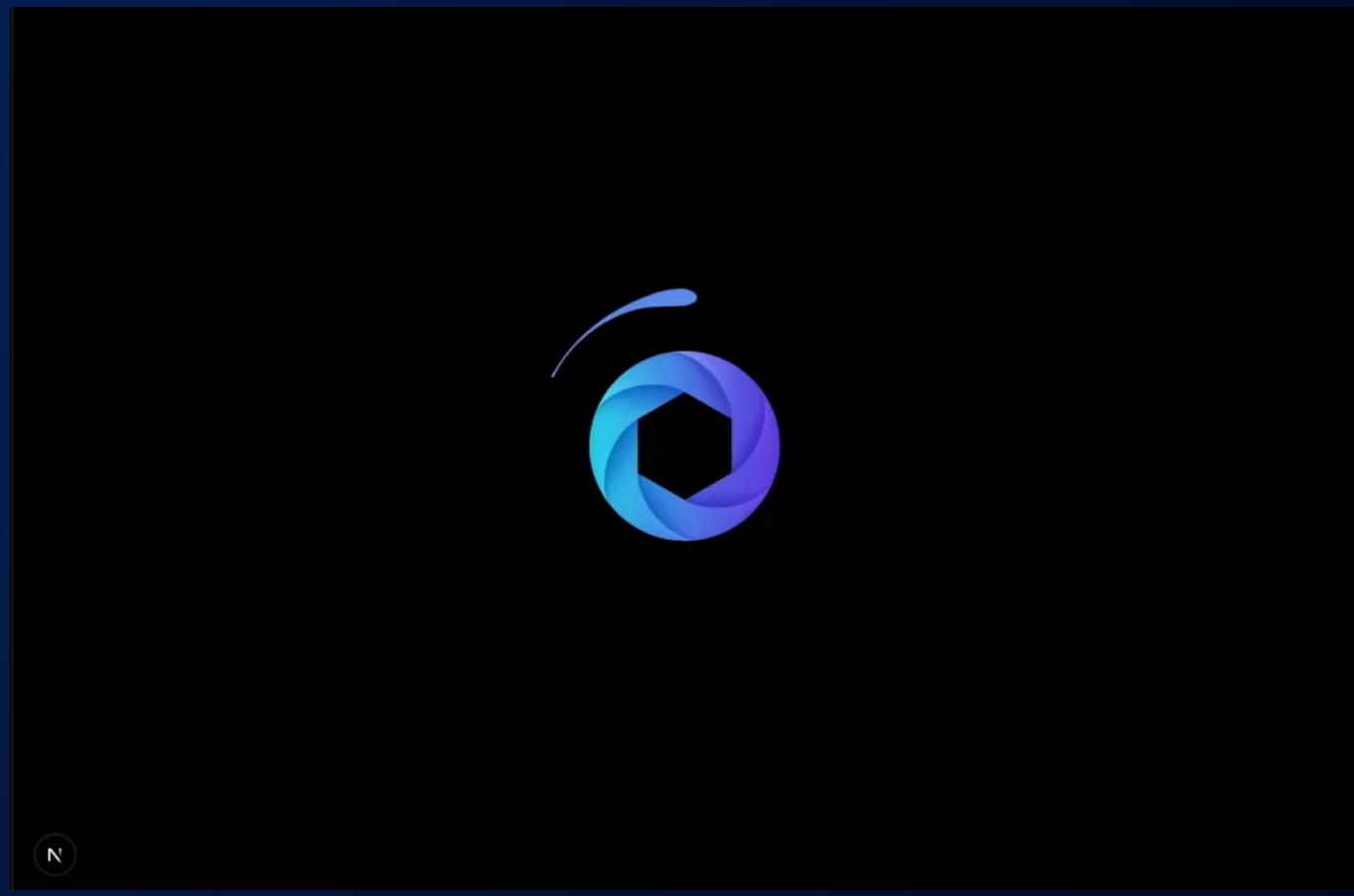
Reactylon

```
<Engine antialias>
    <Scene>
        <universalCamera
            name="camera"
            position={new Vector3(0, 5, -10)}
            target={new Vector3(0, 0, 0)}
            onCreate={camera => camera.attachControl()}
        />
        <hemisphericLight name="light" direction={new Vector3(0, 1, 0)} intensity={0.7} />
        <box name="box" options={{ size: 1 }} positionY={1}>
            <standardMaterial name="material" diffuseColor={Color3.Red()} />
        </box>
    </Scene>
</Engine>
```

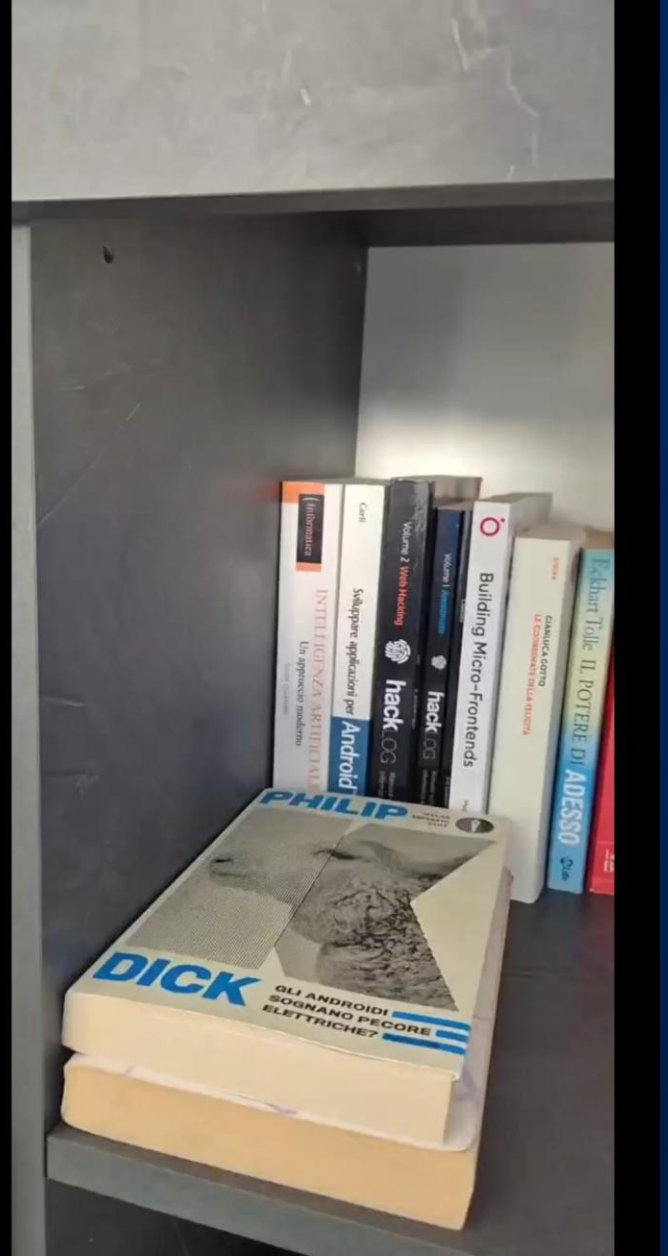
Use Case: Interactive 3D

WEB - MOBILE

Deliver **interactive, real-time visuals** that let users inspect, configure, and understand products or data from any angle - right in the browser or PWA-with smooth animation, lighting, and near-photoreal materials. Beyond configuration, interactive 3D also powers **games and simulations**: physics, collisions, particles, skeletal/morph animation, pathfinding, audio, input, save/load, and multiplayer.



improove



Use Case: VR/AR/MR

MOBILE – HEADSETS - GLASSES

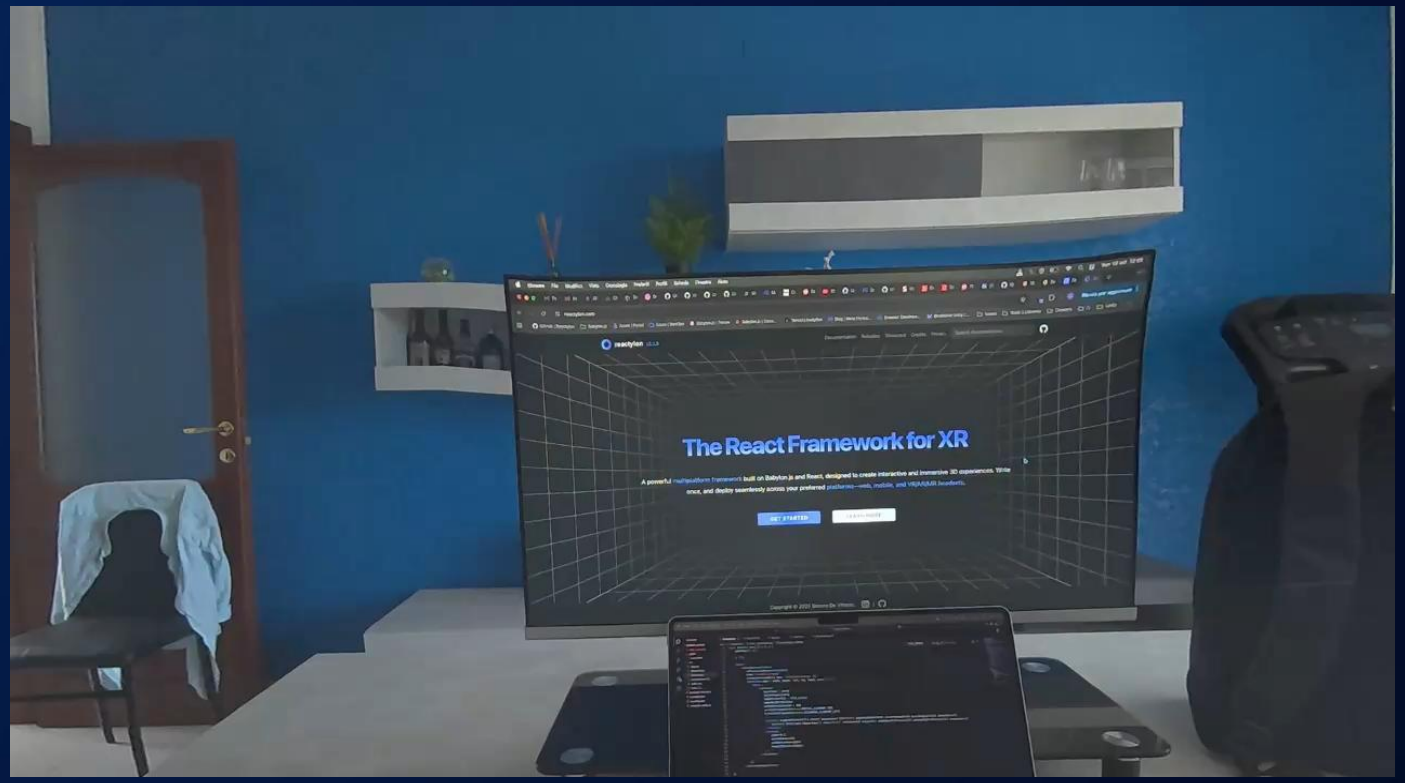
Deliver **immersive experiences** in VR/AR/MR on mobile, headsets, and smart glasses: place and persist content in your space, **blend digital with the real world** via passthrough and occlusion, and interact naturally using hands, controllers, or touch (gestures, rays, grabs, teleportation). Build room-scale scenes with spatial audio, haptics, physics, particles, and animated characters; attach 3D UI or dock standard panels as floating surfaces.

improove

Use Case: Spatial AI Vision

MOBILE – HEADSETS - GLASSES

Deliver Spatial AI Vision on mobile, headsets, and smart glasses: use **passthrough/camera** feeds to run **on-device ML** (object detection, segmentation, pose/hand tracking, OCR) then anchor results into the 3D scene. Overlay labels, outlines, affordances, and guidance arrows; snap UI to detected surfaces; drive interactions with gaze/gestures/controllers; and trigger workflows in real time.



improove

Getting Started

Reactylon ships with a **CLI** that simplifies the process of generating new Babylon.js and Babylon Native projects, making it easier to start in different environments.

Web - Reactylon

npx create-reactylon-app webday

 React ^19  Babylon.js ^8

Mobile - Reactylon Native

npx create-reactylon-app Webday --native

 React ^18  Babylon.js 8.3.0  React Native 0.74.2

improove

Resources & Q&A

Documentation

GitHub

YouTube

improove

WEB DAY 2026

Thanks !

improove