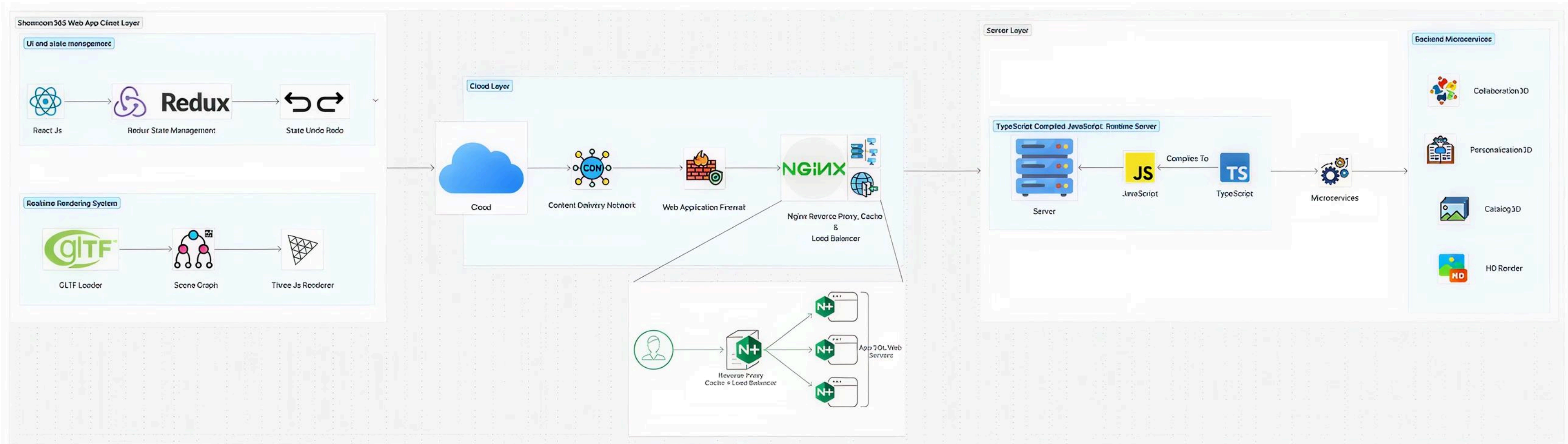


Spaces365 Core

Executive Summary

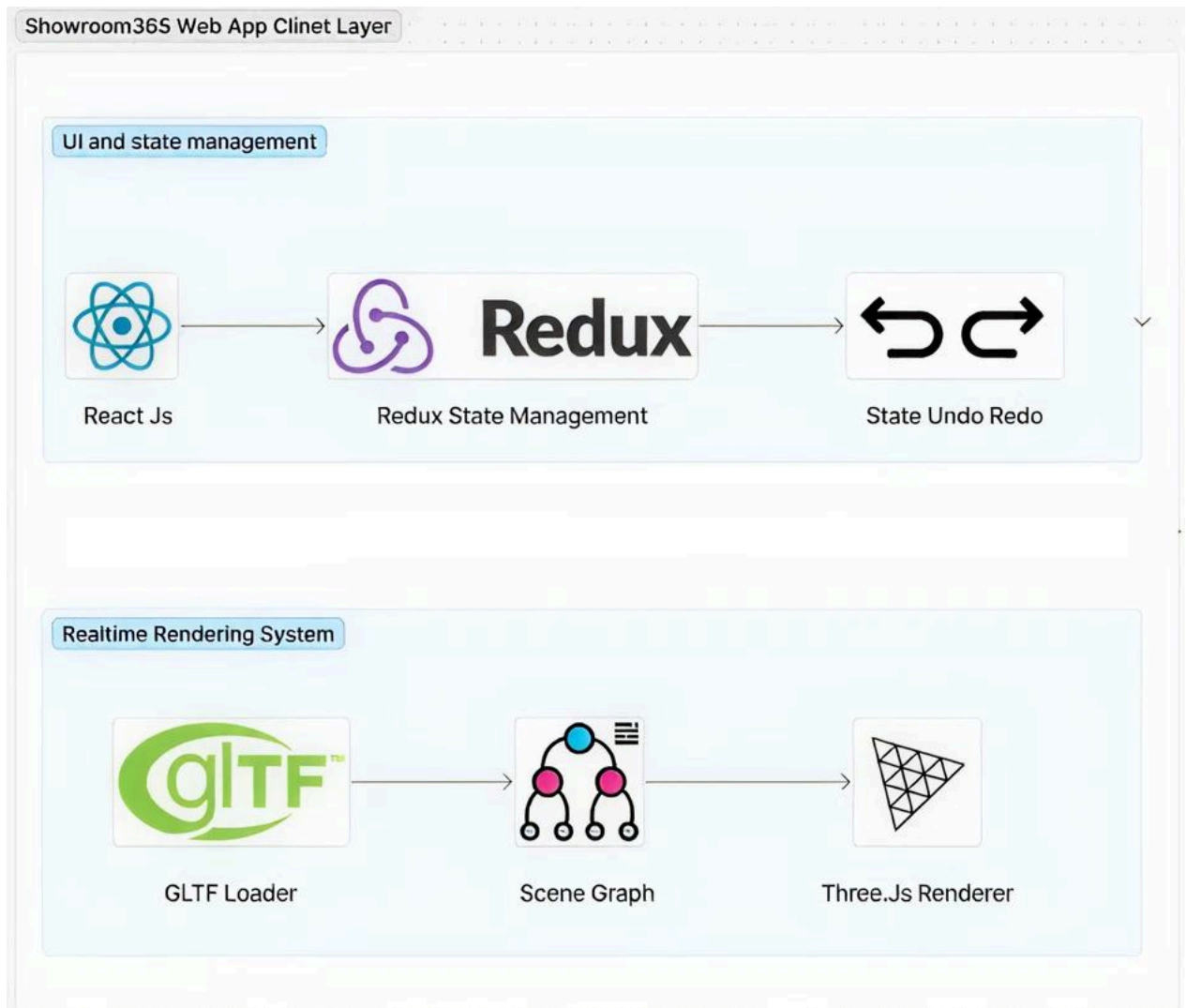
The Spaces365 Core architecture defines the client-side foundation of our 3D visualization platform. It is designed to deliver a high performance, interactive 3D experience directly in the browser, minimizing server dependency for real-time rendering and scene interaction.

High-Level Core Architecture Overview Diagram

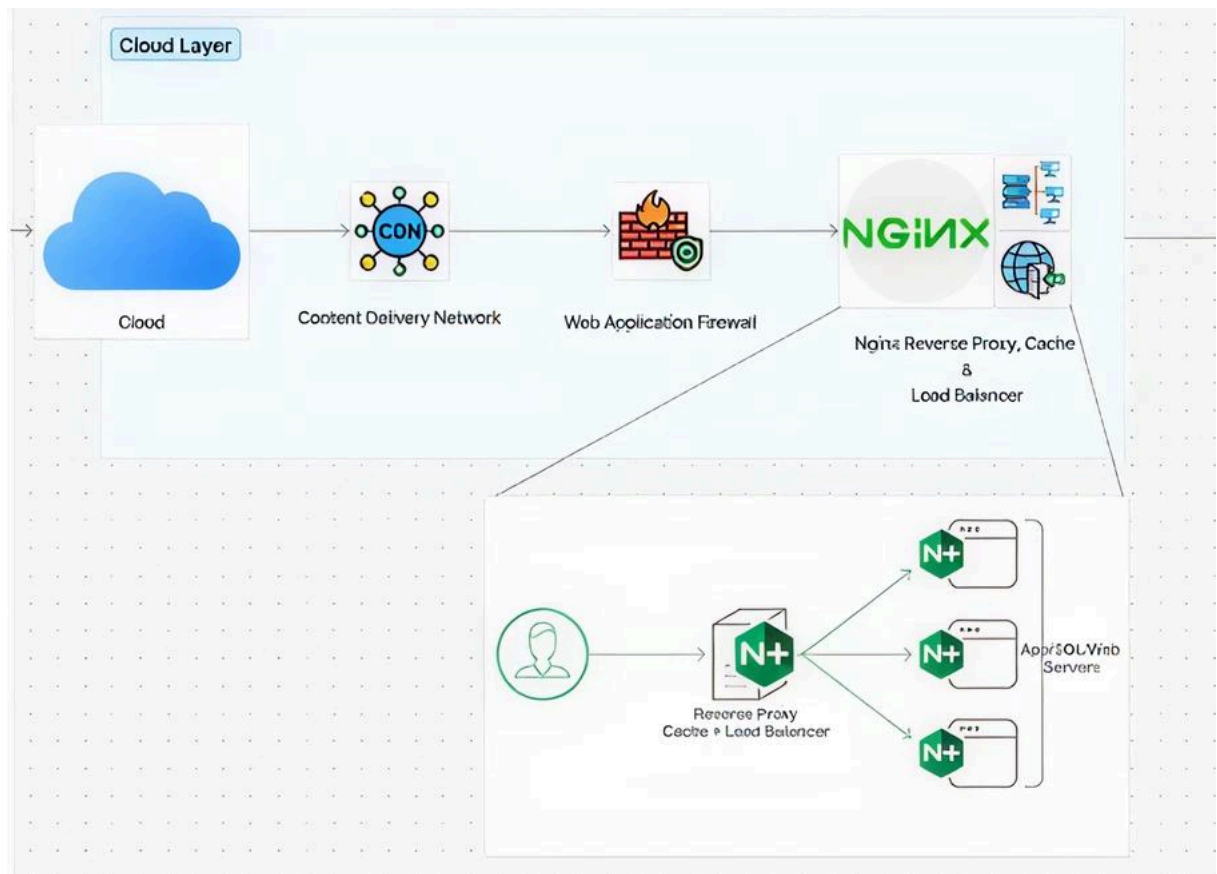


Key Components & Interactions

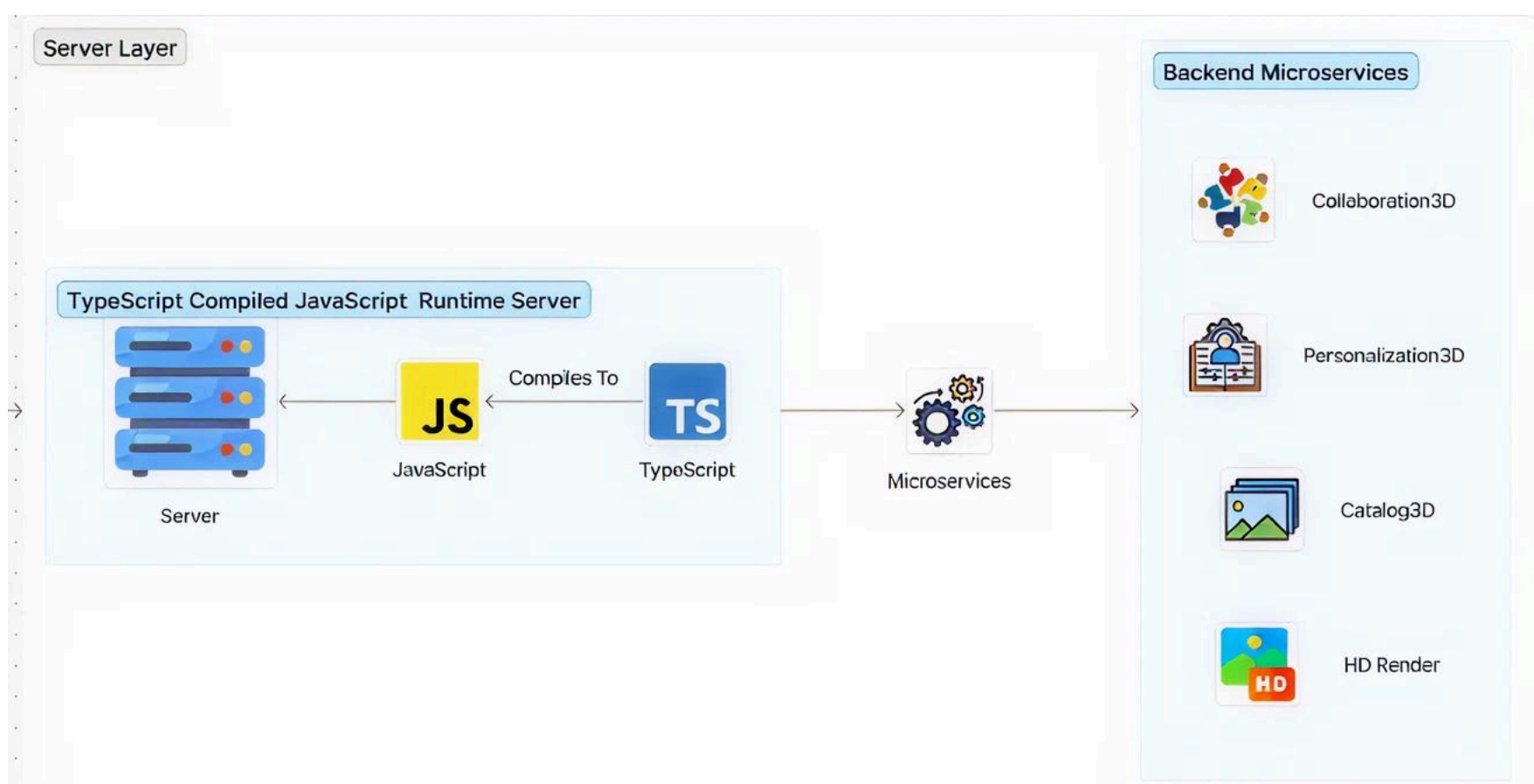
Spaces365 is a web-application supported by any HTML5 compliant web-browser.



Client-side architecture is built on React2 for UI rendering and Redux3 for centralized state management, featuring a built-in undo/reedo mechanism for state-level history tracking. For the 3D layer, we utilize a GLTF4 loader to ingest assets and construct a dynamic scene graph, enabling real-time rendering with ThreeJS5.



Web Server A NGINX9 webserver configured as reverse proxy and connected to a backend NodeJS10 server serves requests from client-tier via http.



Core-server constitutes a “NodeJS” runtime along with a Typescript11 compiler. Typescript ensures type-safety to JavaScript code.

Catalog3D hosts catalog consisting of 3D digital twins of SKUs. They are organized in a taxonomy consisting of categories, sub categories etc. This is like how most eCommerce stores organize SKUs. These 3D digital twins offer fidelity of size, shape, material and other characteristics such as light intensity, beam shape, dynamics, material variants etc. Its built on venture12

Personalization3D service offers a digital twin of a users’ space. This could be an “office room”, “living room”, “hotel lobby”, “landscape exteriors” etc. This digital twin is created in a semi-automatic process from user supplied CAD files or plain photos shot with a smartphone. These inputs are used to re-construct a digital twin with high fidelity of size, shape and material. Its built on owncloud12

Collaboration3D service enables collaboration of the 3D scene in real-time. Real time collaboration is based on using CRDT’s. (Content Replicated Data Type). The popular yjs14 framework is used for implementation.

HDRender creates a photo-realistic version of the 3D scene in Spaces365. It does this using blender15-equivalent versions of glTF objects in the scene along with world class renderer “cycles16”.

Deployment

Spaces365 has been developed and deployed on AWS16. (Amazon Web Services) using the foundation of Ubuntu17 OS. It can be deployed on any of the hyper-scalers such as Microsoft Azure, Google Cloud etc. VisCommerce will support a vendor if they choose to deploy on their private cloud as well.

Security & Compliance World class security infrastructure such as firewalls and routers serves as the foundation. 3D assets are in glTF compliant to khronos’s standard for 3DCommerce19.

Business benefits & alignment Spaces365 is designed to work with existing eCommerce storefronts and to integrate with other enterprise software such as websites, billing systems and CRM systems.

References

1. HTML5 - <https://www.w3.org/TR/2011/WD-html5-20110405/>
2. ReactJS - <https://react.dev/>
3. ReduxJS - <https://redux.js.org/>
4. glTF - <https://www.khronos.org/glTF>
5. ThreeJS - <https://threejs.org/>
6. CDN - <https://aws.amazon.com/cloudfront/>
7. WAF - <https://aws.amazon.com/waf/>
8. LB - <https://aws.amazon.com/elasticloadbalancing/>
9. Nginx - <https://nginx.org/>
10. Nodejs - <https://nodejs.org>
11. Typescript <https://www.typescriptlang.org/>
12. Vendure - <https://vendure.io/>
13. Owncloud - <https://github.com/owncloud>
14. Yjs - <https://yjs.dev/>
15. Blender - <https://docs.blender.org/api>
16. Cycles - <https://www.cycles-renderer.org/>
17. AWS - <https://aws.amazon.com>
18. Ubuntu - <https://ubuntu.com/>
19. 3DCommerce Asset Guidelines - <https://github.com/KhronosGroup/3DC-Asset-Creation>