Article

ATON: an open-source framework for creating immersive, collaborative and liquid web-apps for Cultural Heritage

Bruno Fanini [1,†,‡,*], Daniele Ferdani [1,‡], Emanuel Demetrescu [1,‡], Simone Berto [2,], and Enzo d'Annibale [1,‡]

[1] National Research Council - Institute of Heritage Science (CNR-ISPC), Italy
[2] University of Padua (UniPD), Italy
* Correspondence: bruno.fanini@ispc.cnr.it
† Current address: CNR Area della Ricerca Roma1 - Via Salaria km 29.300 Monterotondo St. Rome, Italy

Abstract: The web and its recent advancements represent a great opportunity to build universal, rich, multi-user and immersive Web3D/WebXR applications targeting Cultural Heritage field - including 3D presenters, inspection tools, applied VR games, collaborative teaching tools and much more. Such opportunity although, introduces additional challenges besides common issues and limitations typically encountered in this context. The "ideal" Web3D application should be able to reach every device, automatically adapting its interface, rendering and interaction models - resulting in a single, *liquid* product that can be consumed on mobile devices, PCs, Museum kiosks and immersive AR/VR devices, without any installation required for final users. The open-source ATON framework is the result of research and development activities carried out during the last 5 years through national and international projects: it is designed around modern and robust web standards, open specifications and large open-source ecosystems. This paper describes the framework architecture and its components, assessed and validated through different case studies. ATON offers institutions, researchers, professionals a scalable, flexible and modular solution to craft and deploy liquid web-applications, providing novel and advanced features targeting Cultural Heritage field in terms of 3D presentation, annotation, immersive interaction and real-time collaboration.

Keywords: Web3D; WebXR; Immersive VR; collaborative interaction; Semantic 3D; 3D Modelling; Heritage Science

Citation: Fanini, B. et al. ATON. *Appl. Sci.* **2021**, *1*, 0.
https://doi.org/

Received:
Accepted:
Published:

1. Introduction

The presentation and dissemination of interactive 3D content on desktop and mobile web browsers has undergone great advancements during the last few years. This is due to several factors, including: a) improvement of browsers Web3D capabilities; b) browsers integration with devices built-in hardware (webcam, microphones, GPS, compass, etc.); c) adoption of modern standards (e.g. WebXR); d) introduction and standardization of 3D formats tailored to interactive web presentation (e.g. glTF). Web browsers are available on virtually all computing devices, thus users can access interactive applications (web-apps) from any device, anywhere, as long as internet (or local network) connection is present. Web applications are also becoming very appealing for the mobile world and especially the Cultural Heritage field (e.g. Museums, Institutions, etc...), due to users not being forced to install any third-party software from stores, nor require additional components to inspect a 3D model or interact with immersive AR/VR experiences.

The web represents a great opportunity to build cross-device web-apps, although such opportunity introduces additional challenges related not only to the creation of responsive user interfaces (UIs), but also to the automatic and seamless adaptation of the application to the device. The "ideal" web-application should be able to reach every device, automatically adapting its interface and interaction model, thus resulting in a single, liquid product that can be consumed on mobile devices (smartphones, tablets), PC, museum kiosks and immersive VR devices (HMDs). Under this light, there is also a growing need for something more exciting than viewers for mere inspection of 3D objects through a web browser: there is an interest in tools for crafting online Web3D/WebXR experiences, applied 3D games, teaching tools and also collaborative web platforms (like *Mozilla Hubs*) tailored to CH needs and requirements.

It is not unusual to visit museums or archaeological sites that offer technological solutions in the form of mobile applications to augment or enrich the visitors' experience using their own devices. In order to consume such interactive experiences, generally visitors have to install applications or third-party software on their smartphones/tablets. From a developer perspective, stores (e.g.: Apple store, Google Play, etc.) have precise guidelines and review processes that sometimes can be fairly complex or result in prolonged acceptance times, especially regarding immersive VR applications. Another challenge for developers arises from traditional application development, where the re-use of code between native solution, web-application and immersive VR solution is very poor, thus leading to the multiplication of effort. Regarding cross-device applications, from a design standpoint several challenges arise not only related to the responsiveness of the application (different screen sizes require adaptable user interfaces) but also to the interaction model adopted. For instance exploring/inspecting a 3D scene through a multi-touch device and through an HMD (immersive VR) require radically different interaction models [1]. This generally leads to the development of different products for different devices, thus creating - once again - fragmentation. Main challenges usually identified in literature within potentially complex 3D scenes consumed on the web include: a) network traffic; b) web-app memory footprint; c) rendering performance. Immersive computing introduces additional demands and performance requirements for low-latency communication in order to deliver a consistent, smooth and acceptable experience. These have to be taken into strong consideration, especially within the limited resources available on mobile web browsers, compared to desktop counterparts.

A big challenge is represented by 3D formats suitable for the web, taking into account state of the art challenges largely addressed in common literature, but also production pipelines of content creators. Which 3D formats should I use to publish my CH objects on the web, while maintaining extensibility and interoperability on the long run with other software tools? Is there a common standard?

Finally, from a deployment perspective, a common challenge for CH stakeholders who intend to disseminate their 3D content on the web is represented by scale (from small servers up to large infrastructures). Which ecosystems should we embrace that provide the necessary *building blocks* for creating scalable network applications and services to efficiently serve 3D content? Are there best practices in terms of design to maximize hardware at our disposal?

The aim of this paper is to present the ATON framework and its components, including novel features and tools available to institutions, museums, researchers and professionals to craft and deploy rich, universal, liquid Web3D applications targeting Cultural Heritage. The next section will describe the state of the art related to web technologies and modern open standards/specifications for 3D presentation and collaboration on the web. The central section (3) will describe the ATON framework and its components. The section 4 will describe experiments and results carried out on selected case studies to assess specific components of the framework, followed by a discussion section (5).

2. Related Work

Web browsers are becoming more and more integrated with devices' hardware. There is already a vast literature on interactive presentation of 3D content using WebGL - a JavaScript API for rendering high-performance interactive 3D and 2D graphics without the use of plug-ins [2]. However, a web page today can also obtain access to different hardware and sensors, on a variety of devices [3], allowing to create and deliver rich web-based experiences that go beyond the mere 3D presentation, without requiring any additional software for final users. A web-application can access camera and microphone for instance, access user location (geolocation API), device orientation and position (landscape/portrait). A web-app can detect network type and speed, battery status and available memory, as well as preventing mobile device screen to standby (wake lock API). WebRTC [4] introduced an API that gives browsers the ability to communicate in real-time, to stream video, voice and generic data between peers. A few web browsers also already provide support for speech recognition and synthesis, through the Web Speech API[1]. All these integrations can be exploited to craft web-based, interactive CH applications that can be consumed by final users just by opening an URL on their devices. Recent literature is also investigating privacy concerns [5] of HTML5 web-applications. Indeed, depending on the hardware/sensor accessed, specific features require A) a secure connection (certified domain) - for instance accessing built-in microphone or camera; and B) user consent.

Immersive VR and AR on the Web - The introduction of the first WebVR open specification [6] allowed developers and designers to create seamless immersive realities in common web browsers using consumer-level 3-DoF and 6-Dof HMDs. Because of its inherent openness and accessibility, the web represents a great opportunity to enable universal access to immersive VR experiences, without requiring additional software. The specification played a big role in democratizing immersive VR by allowing larger audiences to experience 3D content through low-cost (e.g. cardboards) or high-end headsets (HTC Vive, Oculus Rift, Oculus Quest, etc.) directly from a web page.

The open specification evolved into WebXR[2] [7],[8]: it aims to unify VR and AR (Augmented Reality) worlds, supporting a wide range of user inputs (e.g. voice, gestures) enabling users to navigate and interact with virtual spaces over the web [9],[10]. WebXR allows entirely new, gripping experiences to be built for the web, and it is also fueling content creators who need to test and deploy immersive VR content on the web. WebXR is also being exploited for creative narratives in CH through immersive AR experiences [11] or toolkits for creating XR digital storytelling [12]. It is also possible for a web page to exploit VR controllers haptics, and even track articulated hand poses (at the present time, as experimental feature) allowing hand tracking with supported HMDs (e.g. Oculus Quest) in WebXR spaces.

Standardization of Web3D formats - *glTF*[3] (GL Transmission Format) by Khronos is a royalty-free, open standard for efficient streaming and rendering of 3D models and scenes [13]. It minimizes the size of 3D assets and the runtime processing required to unpack and use them. glTF is an extensible publishing format that streamlines authoring workflows and interactive services by enabling the interoperable use of 3D content across the industry. Particularly interesting is the support to the *PBR* (Physically-Based Rendering) model, crucial within the Cultural Heritage [14] and other fields to simulate advanced materials at runtime by approximating the flow of light. Furthermore, Khronos also recently released PBR extensions[4] for glTF to support volume-based absorption, refraction (see fig. 1, top right), and complex specular reflections to be used by diverse renderers, from real-time rasterization to production-class path-tracing. glTF data like geometry can be also compressed: *Draco*[5] is an open-source library developed by Google for compressing and decompressing 3D meshes and point clouds. It is intended to minimize the storage and improve the transmission of 3D models over network connections. There are several open-source tools to perform Draco compression, and built-in glTF exporters in 3D modeling software (like Blender) already provide compression options for content creators. The glTF format is rapidly spreading and is being largely

[1] https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
[2] https://immersiveweb.dev/
[3] https://www.khronos.org/gltf/
[4] https://www.khronos.org/blog/using-the-new-gltf-extensions-volume-index-of-refraction-and-specular
[5] https://google.github.io/draco/

adopted by many platforms due to its high interoperability and perspectives to address specific archiving challenges [15]. Several 3D modeling software tools (Blender, 3DS Max, Maya, etc.) as well as game engines like *Unreal Engine 4* - can export directly in glTF (including PBR materials), thus boosting the web publishing workflow. Several institutions, including the Smithsonian, already published open-access 3D models using this standard[6]. Furthermore, with the latest specification, Khronos is also preparing glTF to be submitted for transposition to an international standard.

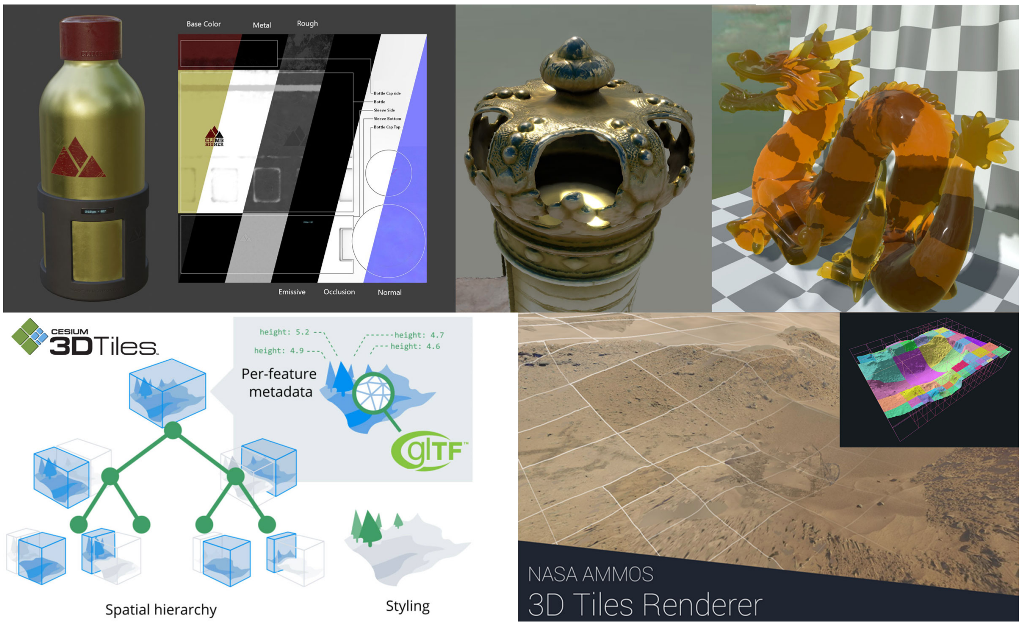

Figure 1. Top row (from left to right): water bottle sample with PBR materials from Khronos glTF samples, detail of a public domain CC0 model from Malopolska's Virtual Museums (https://sketchfab.com/WirtualneMuzeaMalopolski) and support for real-time volumetric refraction and absorption (sample glTF model "Dragon" by Khronos). Bottom row: Cesium 3D Tiles specification and NASA AMMOS open-source 3D Tiles renderer

3D Tiles is an open specification built on glTF developed by Cesium [16] for sharing, visualizing and interacting with massive heterogeneous 3D geospatial content, high-resolution photogrammetry datasets or BIM [17] across desktop, web, and mobile applications [18]. The foundation of 3D Tiles is a spatial data structure that enables Hierarchical Level of Detail (H-LOD) so only visible tiles are streamed and rendered, thus suitable to maintain interactive performances. Single tiles can also adopt Draco compression to further improve transmission size. 3D tiles specification for tilesets, associated tile formats and the associated styling specification are open formats that do not depend on any vendor-specific solution, technology, or products. The open specification is being adopted by the community outside Cesium, including NASA that is developing an open-source 3D tiles renderer [7] for AMMOS project (see fig. 1, bottom). The 3D tiles specification is also being integrated with game engines like Unreal Engine 4[8] allowing interactive visualization of global high-resolution content (photogrammetry, terrain, imagery, and buildings), and with other open-source multi-platform 3D engines like O3DE (https://o3de.org/).

Web3D presentation tools - There are several tools, libraries and platforms to present interactive 3D content on the Web without requiring any additional software.

[6] https://www.khronos.org/news/press/khronos-smithsonian-collaborate-to-diffuse-knowledge-for-education-research-and-creative-use
[7] https://ammos.nasa.gov/
[8] https://cesium.com/platform/cesium-for-unreal/

One of the largest open-source libraries (also in terms of community) is *Three.js*[9][19], often used also for mobile fruition [20] and natively supporting glTF format, WebXR and modern web standards. *A-Frame* (Mozilla)[10] is another widespread web framework [21] for building 3D/AR/VR experiences on the Web: the entity-component system framework provides a familiar authoring tool for web developers and designers, while embracing modern Web standards. *Model-viewer* (Google) is also another open-source solution targeting the presentation of small 3D models for AR and VR through WebXR. *3DHop* [22] is an open-source software which allows the creation of interactive WebGL presentations with special focus on high-resolution 3D models targeting the CH field. Although at the moment, it has no support for WebXR, glTF nor PBR materials. Regarding proprietary platforms, *SketchFab* is a well-known solution to publish, share, discover 3D, VR and AR content on the Web. Its viewer offers an advanced rendering system (including PBR model, screen-space reflections, depth-of-field, refraction and more) and glTF download options. Several institutions, including the British Museum, published collections of 3D models on SketchFab [23]. The platform is also investigating streaming of multi-resolution 3D models through paged hierarchical level-of-detail ("Massive" project). There are also other frameworks and visualization toolboxes already investigating Web3D presentation architectures targeting CH. *Resurrect3D* [24], based on Three.js, aims to offer basic visualization and interaction capabilities, but also customizability for domain experts to develop specific analysis and visualization tools. *Voyager* (Smithsonian) [11] offers material editing, relighting, measurement and annotation tools. Within open source 3D WebGIS, *MayaArch3D*[12] is a virtual research environment that combines aspects of 2D, 3D, GIS, and archeological data into a platform.

Creating and deploying web-applications - Regarding web-applications creation and deployment, a recent set of standards advocated by the Google Web Fundamentals group introduced features such as offline support, background synchronisation, and home-screen installation [25]. Such an approach is known as *Progressive Web-Applications* (PWA) that aims to offer responsive, connectivity independent, app-like, discoverable and linkable web-based solutions. PWAs have completely brought in a new dimension to mobile development, making web apps look, feel and act similar to native and hybrid apps [26]. Web-applications (including those dealing with interactive 3D content) can be deployed in local or networked contexts in order to be consumed by final users. *Node.js*[13] is an open-source, cross-platform, back-end javascript runtime that allows to build scalable network applications [27]: it's often employed for deploying web-applications, including those targeting Cultural Heritage field [24],[28],[29],[30]. It is common in Node.js contexts to offer also REST APIs [31] to perform server-side tasks or allow easy integration with external services or platforms. Furthermore, *microservices* emerged as a new architectural approach, in which distributed applications are broken up into small independently deployable services, each running in its own process and communicating through lightweight mechanisms [32]. These approaches allow us to design robust web-oriented frameworks with modular architectures, easily adaptable to a wide range of requirements and hardware (small servers up to large infrastructures) with the possibility to independently control service components.

Collaborative Web3D - There is indeed a strong interest in literature for CVE (Collaborative Virtual Environments) widely investigated for desktop-based applications [33],[34],[35]. The possibility to share the same virtual 3D space and interact with other people, became even more appealing during the covid-19 pandemic [36]. Most of the recent literature is focused on social VR [37] paradigm, which is gaining more and more attention on desktop-based applications. There is already a robust literature on

[9] https://threejs.org/
[10] https://aframe.io/
[11] https://smithsonian.github.io/dpo-voyager/
[12] https://mayaarch3d.org/en/
[13] https://nodejs.org/

taxonomy [38] supporting the design of these applications, alongside user experience [39] and highlighting opportunities for CH [40]. Several works are also focusing on the importance of avatar representations [41] within social VR, and investigations related to personal spaces [42]. Within Web3D/WebXR contexts, there are a few works investigating synchronous collaboration, in which multiple users access the same virtual (immersive) space using common web browsers [43]. The most prominent example in the open-source panorama is certainly *Mozilla Hubs*[14] (based on A-Frame) that allows users to create online virtual meeting spaces, and it's being used also for live workshops with children [44]. Recent research is also investigating multi-user collaboration through decentralized approaches for live coding environments in WebXR [45]. Regarding proprietary solutions, *LearnBrite*[15] is another collaborative web-based solution focused on micro-learning and instructor led training, allowing to create shared immersive scenarios across multiple devices. From a technical perspective, real-time communications are typically realized through web-socket protocol [46], often adopted for creating multiplayer WebGL games [47] and 360 social VR experiences on the web [48]. These projects offer valid solutions for multi-user collaboration in Web3D/WebXR virtual environments, although not specifically targeting Cultural Heritage, and specific collaborative interactions/tools required for this scope.

3. ATON Framework

The overall design of the framework is conceived to be highly modular, in order to accommodate different scenarios and requirements of museums, institutions, researchers, professionals and other audiences intending to deploy interactive 3D experiences through a wide range of hardware. The ATON architecture and its components are the product of national and international projects, experiences and user feedback gathered during the last 5 years, which allowed the framework to evolve into the current state (version 3.0). The very first version (1.0) was developed under the ARIADNE European project [49] for visualization of large 3D landscapes online [50]. Regarding presentation layer (client-side), the framework stands on the shoulders of open-source 3D libraries (such as Three.js) leveraging modern standards such as WebXR to present rich, immersive 3D experiences to final users. Service layer (server-side) stands on top of Node.js and several modules (Express.js, socket.io, Passport.js[16] and others) to provide scalable network services handling 3D content streaming, user authentication, collaborative sessions and much more. Specific focus for ATON was placed on ease of deployment, scalability, interoperability, out-of-the-box tools and presentation features for CH stakeholders, while offering a simple but powerful API to developers.

[14] https://hubs.mozilla.com/
[15] https://www2.learnbrite.com/
[16] http://www.passportjs.org/

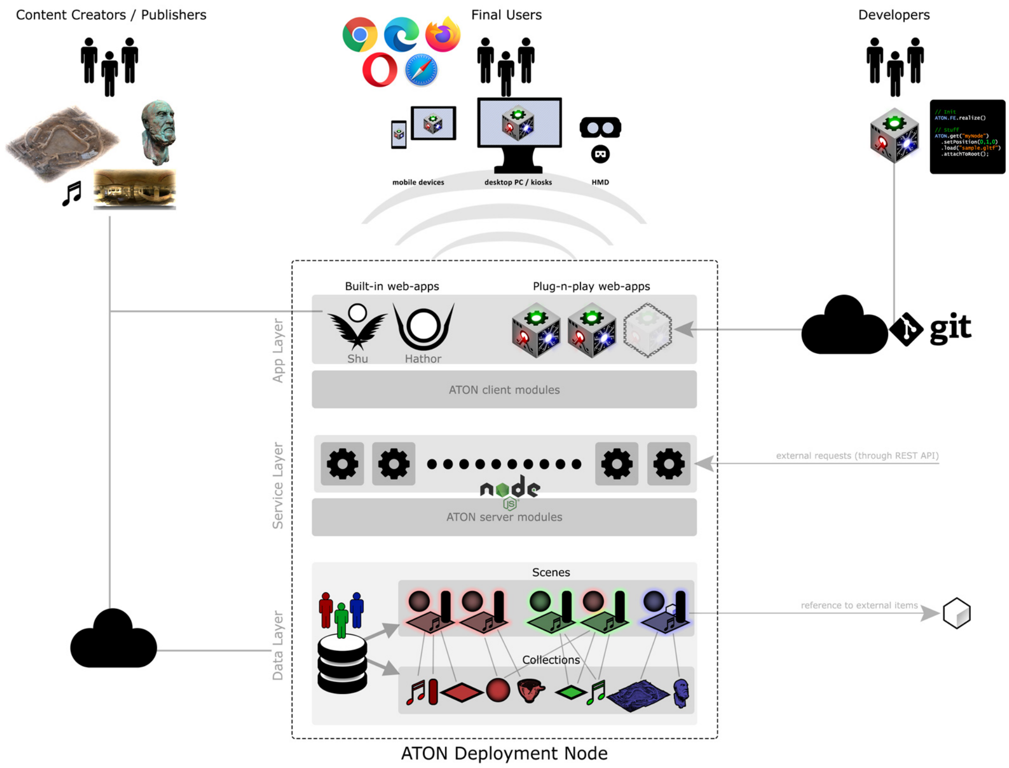

Figure 2. Overview of the ATON Framework architecture

This section describes core modules of the framework, highlighting specific client and server components, as well as integration with 3D content workflows (fig. 2). Deployment scenarios are first introduced in section 3.1, then data layer (collections and scenes) in section 3.2. Deployment of web-applications is then discussed using a *plug-n-play* approach for the architecture in section 3.3. The liquid presentation layer is then discussed in section 3.6, including cross-device rendering capabilities, semantic annotations and client components for interacting with 3D content. Components enabling collaborative sessions among remote users are discussed in section 3.7, and finally a built-in front-end described in section 3.8.

3.1. Deployment Node

We first define the *Deployment Node* (DN) as the virtual or physical machine where one instance of the ATON framework is deployed. Thanks to Node.js portability and network scalability, a wide range of hardware solutions can be employed as DN (see fig. 3, top). These range from low-cost single board computers (like a Raspberry Pi[17]), up to laptops/PCs in local networks (e.g. classrooms), up to small servers or large infrastructures available over internet connection, with world-wide reachability. Depending on requirements and specific demands, this also allows the entire framework to be deployed on all existing cloud services that support Node.js like Google Cloud[18], Amazon Web Services (AWS)[19], Heroku[20] and many more.

17 https://www.raspberrypi.org/
18 https://cloud.google.com/
19 https://aws.amazon.com/
20 https://www.heroku.com/

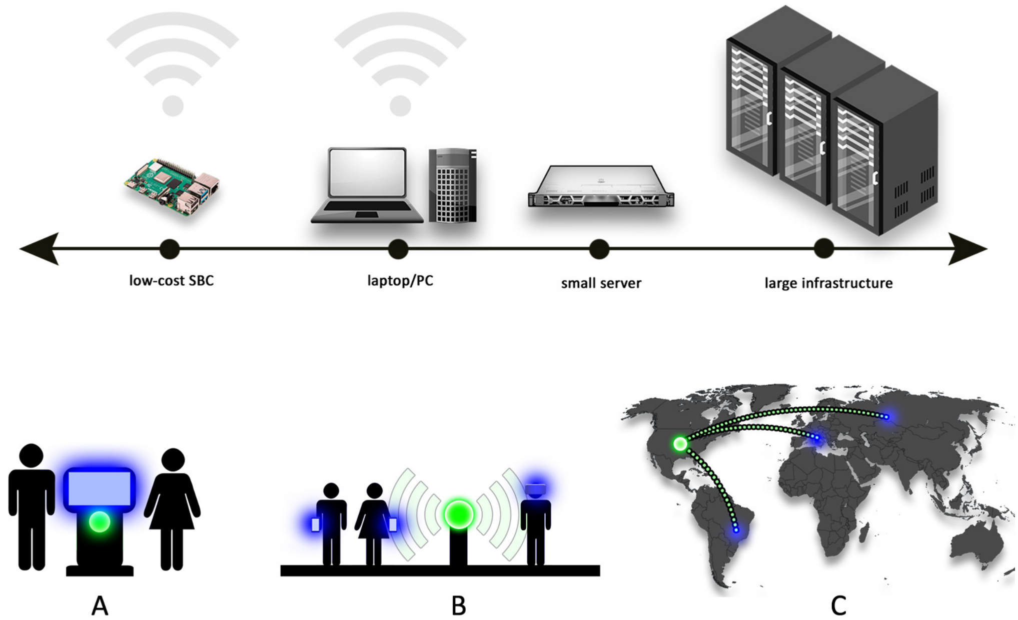

Figure 3. Deployment hardware (top) and three different deployment scenarios (bottom)

The default setup consists in a DN delivering content (e.g. 3D models) and web-applications to remote consumers (users) using a multitude of different devices, through local or internet network connection. There are also scenarios where a DN is used also as presentation device (e.g. museums kiosks) thus deploying and consuming the web-app on the same hardware (see fig. 3, A), without any local network or internet connection required from the institution side. There are different services operating on a DN when the framework is up and running. They are designed to scale automatically with the DN hardware, exploiting - where available - clustering and load balancing capabilities offered by process managers like PM2 [51]. Such services have different roles and tasks including content streaming, user authentication, collaborative sessions (discussed in the next sections), and much more. Since a *microservice* model is adopted, they can be easily configured or independently disabled to fulfill a wide range of requirement scenarios (see fig. 3, bottom). A documented REST API[21] allows local (or external) web-apps to perform different operations on the current DN: this is also specifically designed to facilitate integration and communication with external CH tools or services within federated infrastructures or projects. Describing in detail each service is out of scope for the paper, thus for more technical details we suggest official documentation.

3.2. Collections and Scenes

The framework defines an important distinction between collection and scene concepts.

A *collection* is a set of items - including 3D models, panoramas, audio sources, etc. - that we intend to use to create an interactive 3D presentation or space. Formats of collection items within the framework must be suitable for the web (e.g. png, jpg, webm, mp4, mp3 - just to name a few related to multimedia content). The main adopted format for 3D models is glTF with Draco compression (see state of the art section). Due to its interoperability, such standard offers content creators smooth integrations with several 3D software tools and engines (like Blender, Maya, Unreal Engine 4, etc...) while open-source tools (e.g. Cesium tools - https://github.com/CesiumGS) can be exploited to automate ingestion of desktop formats into collections. For more complex items (e.g. massive heterogeneous 3D geospatial content, or large photogrammetry models)

[21] https://aton.ispc.cnr.it/apidoc/server/

Cesium 3D tiles open specification is adopted, thus offering smooth streaming of large multi-resolution items over the web (see section 2).

A *scene* on the other hand, is an arrangement of collection items, exploiting hierarchical organization and transformations offered by scene-graphs. A scene may indeed include specific viewpoints (POVs), keywords, semantics, soundscape, and much more. Such separation is quite common in several game-engines (e.g. Unreal Engine 4) where a collection of assets can be used and referenced to arrange multiple levels (scenes) in the project.

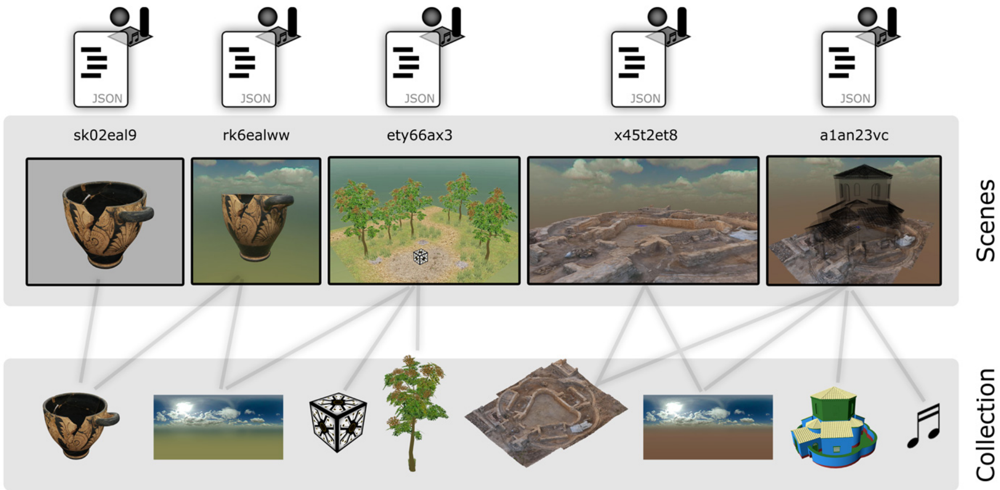

Figure 4. Scenes and Collections

A scene may indeed also consist of a single item (e.g. a 3D model) exploitable for 3D galleries of online virtual museums, with each scene corresponding to a single collection item. Within a DN, each scene is assigned a unique identifier (a Scene-ID, or "*sid*" for short) in the form of a string (e.g.: "e45huoj78", "demo/skyphos", "user3/site2/reconstruction") which can be used by any ATON-based web-app to address a specific published 3D scene (see fig. 4). The scene itself is stored as a compact JSON file (scene descriptor), similar to the *svx JSON* used by Voyager[22]. Scenes are thus very cheap in terms of storage (data layer) compared to collections. The JSON format offers direct manipulations by local ATON services, third-party services or web-based tools - as well as guaranteeing full extensibility. Several libraries (e.g. Three.js, Babylon.js, etc.) already provide import/export routines for entire scenes in their own JSON format, although they usually store information not required for CH scopes. ATON has the goal of keeping the JSON scene descriptor very light, since it targets a specific subset of features targeting Cultural Heritage (scene-graph, semantic graphs, soundscape, viewpoints, general scene information, etc.) that will be discussed in the next sections.

Such an approach differs from other Web3D solutions (like SketchFab for instance) where the concepts of 3D model and scene overlap. The strict separation in ATON between scenes and collections has several advantages:

- Re-use: the same item (3D model, panorama, audio, etc.) can be employed (and re-styled) in different scenes, avoiding unnecessary duplication in terms of storage
- Update: an item (e.g. a 3D model, a panorama, etc.) can be improved by content creators and easily updated in the collection, automatically affecting all the scenes in which such item is referenced
- Caching: in web-based scenarios, this approach allows a) to facilitate browser caching (e.g. when switching to different scenes referencing the same asset) and b)

[22] https://smithsonian.github.io/dpo-voyager/explorer/scene-create/

to avoid duplicate client requests in the same scene (e.g. multiple instances of the same 3D model, like a tree)

- Cloning: a scene can be easily cloned within the DN, maintaining a very small footprint in terms of storage, and allowing users to work on different copies or hypotheses of a 3D virtual environment
- External references: a scene may even contain references to cross-domain sources (e.g. a 3D model or tileset located in another DN, or accessible through a public url) thus allowing the distribution of resources across multiple DNs or servers

3.3. Web-Applications layer

The application layer of a DN (see fig. 2) can host multiple web-applications (and their logic): these are consumed on demand by clients (users) who access the CH application (tool, 3D virtual museum, immersive experience) on their own devices, without any installation. The framework offers a few built-in web-apps: a basic back-end "*Shu*" (described in the next section) and an official front-end (called "*Hathor*") to present 3D scenes that will be discussed in detail in section 3.8.

The ATON architecture allows to host and deploy custom web-applications: this is crucial since museums, institutions, professionals, etc. may have different requirements in terms of user interface, 3D content presentation, semantics and much more. The framework thus offers developers a *plug-n-play* architecture where web-apps can be easily deployed or transferred to other DNs. Each application lives in a specific folder of the DN, thus enabling different integrations for developers with git repositories, sFTP, cloud storages, etc.

The framework offers a basic web-app template (PWA-compliant) as a robust foundation to build custom, cross-device, liquid web-applications. This approach also avoids core components duplication (e.g. presentation modules) since each web-app has direct access to ATON client components and services. Each web-application possesses a unique ID (or "*app-id*") thus offering a consistent mapping within the DN of collections, scenes and applications. Single applications can rely on content in the app folder (e.g. 3D models, panoramas, media content, etc.), or access centralized collections and scenes on the instance. The flexibility of such approach from the perspective of museums, institutions or professionals is that they are offered different scenarios to meet their needs:

- Exploit the built-in ATON front-end (if it fulfills their requirements) and use it "as it is" to present 3D models and scenes to final users, without any code development required
- Extend the built-in front-end with custom functionalities
- Develop and deploy a custom web-app through the *plug-and-play* architecture

3.4. Access, manage and publish content

The framework provides a built-in authentication system that allows content creators and publishers to access, manage and modify their own collections and scenes on the deployed instance. The most basic setup allows to place and organize such content directly in the main collection folder of the DN (e.g. 3D models, audio, panoramas, etc.). The framework offers a built-in lightweight, responsive back-end ("*Shu*") where authors, editors or content creators can authenticate to publish and manage 3D scenes with ease. The local authentication middleware is based on passport.js, allowing a fine-grained control on requests involving content access, modification or other tasks. Furthermore, passport.js middleware offers a wide range of integrations with different authentication strategies (Facebook, Google OAuth, and many more) thus providing a flexible and extensible system for more advanced back-ends.

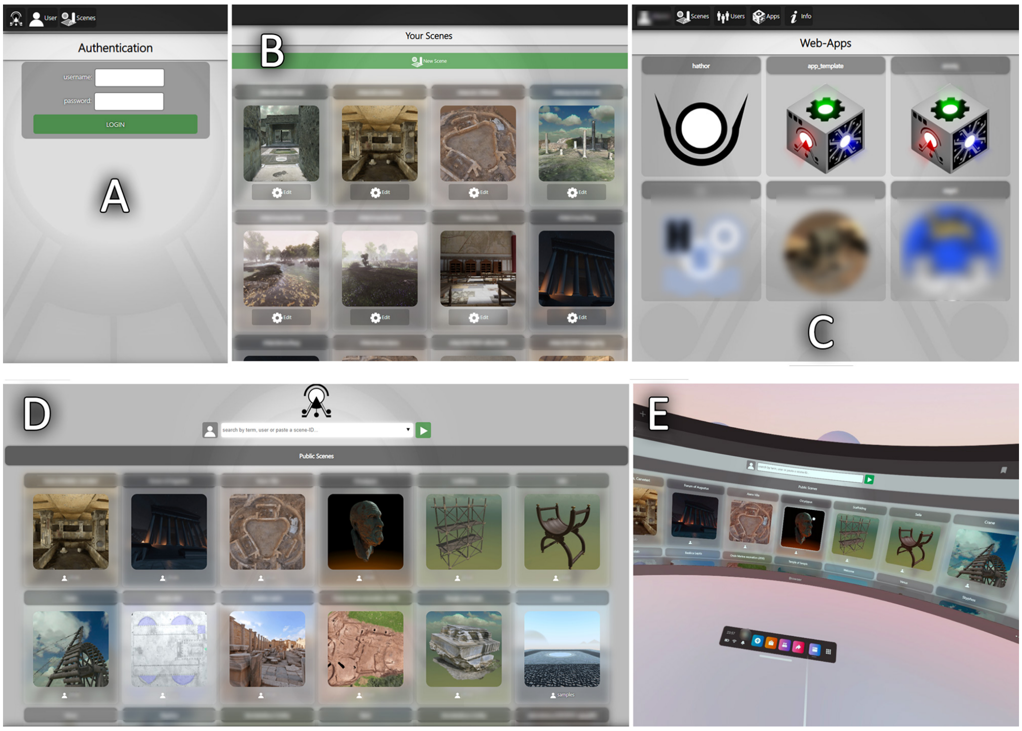

Figure 5. Shu back-end. Authentication (A); private scenes gallery (B); web-applications gallery (C); public landing page on standard browser (D) and immersive browser (E) using the Oculus Quest 2

Shu allows authenticated users (fig. 5, A) to create online scenes with ease starting from their collections (or remote content) in private galleries (fig. 5, B). Authenticated users can then publish their scenes on the main landing page (fig. 5, D and E) with public access, thus allowing remote users to consume the 3D scene on every device. The landing page also provides a search box to filter public scenes by author, term or keyword, very useful to create custom galleries (e.g. collection of museum objects). Administrators have additional control on the instance and web-apps currently hosted by the framework (fig. 5, C). The standard workflow thus involves authenticated users to upload, manage or modify their content into collections, then arrange and publish a 3D scene. Several options enabling remote access to local collections and scenes are available, although a very flexible and comfortable solution for content creators, editors and publishers is the cloud integration (see assessment section). Indeed, the REST API and the modular structure of the framework allow the development of custom or more advanced back-end solutions to access and manipulate content.

3.5. Modifying published scenes

Once a 3D scene is published (i.e.: it has a unique ID assigned), updating or manipulating its items *at runtime* using a front-end or presenter is possible through proper interfaces (e.g. transforming objects, adding annotations, etc.) but making these changes persistent requires direct intervention on the corresponding JSON descriptor (see 3.2). It is quite natural to expect routines that allow authenticated users to change, update, annotate, fine-tune or modify their scene in a persistent manner. Scene patches allow client web-apps to send compact partial edits (patches) to the DN through the REST API in order to modify a given scene and its corresponding JSON file. Such approach is based on *JSON patches* [52], a format for describing changes to a JSON document, that fully suits the scene descriptors of the ATON framework.

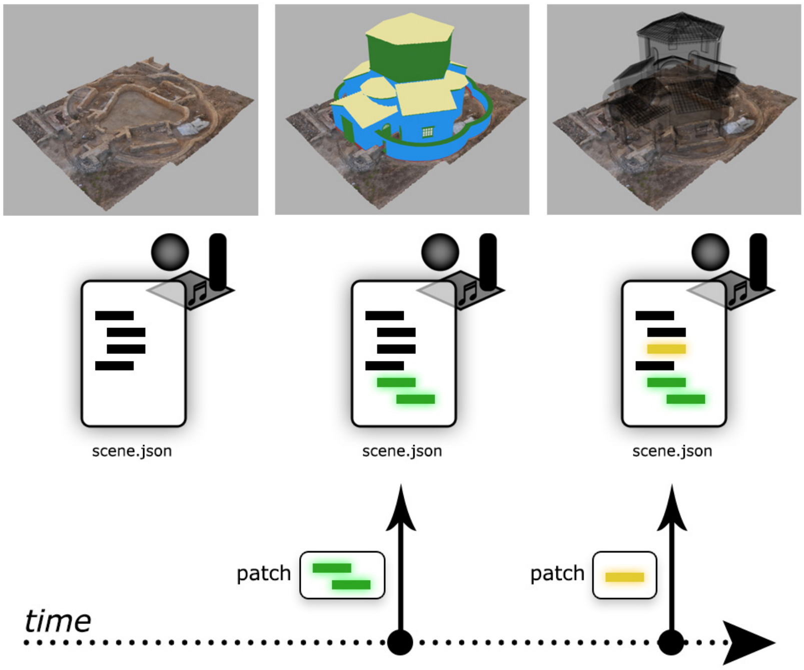

Figure 6. Sample JSON patches (add node, update material) sent over time to the server to apply partial modifications to the 3D scene descriptor

This enables authenticated clients to perform scene updates at runtime by sending over the network small patches that persistently modify the JSON file on the server (see fig. 6). Common examples include applying scene-graphs modifications (adding/removing/transforming scene nodes); adding/modifying scene description or semantic annotations; modifying lighting or environment and more in general, anything in the JSON scene descriptor. A web-application can thus provide usable interfaces allowing users to transparently perform such tasks and apply changes to a scene. This approach not only allows web-apps to perform arbitrary edits to a scene through compact JSON patches exchanged over the network, but it is also a robust approach for future changes involving the JSON scene descriptor, custom JSON descriptors or to modify other JSON files. Furthermore, the DN may easily keep track of 3D scene modifications performed on the descriptor by creating lightweight snapshots, also exploitable for instance for *undo* operations.

3.6. Presentation layer

This section focuses on the presentation side (client devices) of the ATON framework and its available components to build "liquid" web-apps, embracing modern web standards. There are several challenges to face for 3D presentation, that go beyond classics such as performance (e.g. dealing with mobile web browsers and their limited resources) and content streaming over the network. Among these, a big challenge is related to the variety of devices employed by final users. It is not only a matter of providing *responsiveness* [53] (user interface elements) for different screen sizes - but also dealing with different interaction models.

3.6.1. Device profiling

The role of the ATON profiler component is to automatically detect user device capabilities at the very beginning of the experience delivered through a web-app. This includes detection of user device built-in sensors and accessible hardware, as well as connection type, since specific features (such as WebXR presentation, or accessing microphone, camera, GPS, etc.) require secure connections. The main goal of such profiling is allowing other ATON components to automatically adapt A) the user interface (UI); B) the rendering system and C) the interaction model (e.g. navigation) - to the capabilities of the current device (see fig. 7). This plays a crucial role into offering final users universal, *liquid* web-applications.

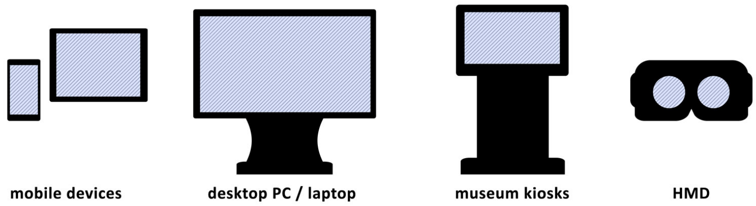

Figure 7. Device classes to consume ATON content

3.6.2. Rendering system

The core rendering system is based on Three.js (see section 2), exploiting advanced features offered by the open-source library for the presentation of CH objects, archaeological sites or generic 3D virtual environments. As previously anticipated, the interactive rendering fully supports the PBR model, thus providing advanced simulation of materials and how they react to the surrounding environment. The PBR model is widely adopted and well-established, especially within desktop-based content creation pipelines targeting applied games [54],[55] and Web3D platforms like SketchFab. The workflow is well-defined within the glTF standard, with several properties (e.g. roughness, metalness, emissive, etc.) that - just like desktop-based pipelines - are generally encoded into multiple textures. Three.js PBR model is also compatible with WebXR, thus offering the final user a consistent simulation of the flow of light when consuming content through a stereoscopic device (i.e.: high-end HMDs or cardboards).

Common Web3D solutions (like SketchFab) offer a single light-probe [56] system to simulate a general response of 3D model surfaces to the environment. In ATON, a multiple light-probe system is provided: this allows to place and arrange *multiple* light-probes in a 3D scene, thus leading to a more consistent simulation of PBR materials in relation to the surrounding environment. Light-probes (LP) independently capture their surroundings, with each mesh automatically assigned to an LP depending on proximity, following a similar policy of game engines like Unreal Engine 4. This drastically improves the final result, with more consistent reflections and overall illumination for 3D items scattered across the scene (detailed discussion later in section 4.2).

Figure 8. A few captures from interactive 3D scenes in ATON. Top row: real-time shadows and advanced effects (bloom, ambient occlusion); Middle row: multiple light-probes system and PBR materials; Bottom row: depth-of-field effects, real-time volumetric refraction and absorption (Khronos glTF extension) and multi-resolution dataset (Cesium 3D Tiles)

The lighting system also supports dynamic shadows (provided by Three.js), to further improve the overall quality of the presentation, from small objects (e.g. artifacts) up to large environments (e.g. archeological sites). ATON rendering system also supports dynamic pixel density, to control or fine-tune framerate on devices with poor graphics performances. A similar approach is employed for immersive WebXR sessions through the framebuffer scale factor, while WebXR open specification is currently aiming to embrace *foveated rendering* [57]. Web-applications based on ATON automatically adapt and fine-tune these parameters according to the profiler, in order to maintain a consistent framerate, or directly control them for specific requirements. These can be particularly useful for small museums employing low-cost or cheap GPU hardware kiosks to present 3D items.

3.6.3. Navigation system

This is a central component of ATON, particularly advanced since it is designed to adapt to several devices ranging from mobile devices up to HMDs for immersive VR. Thanks to the profiler, different interaction models are offered and automatically adapted: on mobile devices for instance (smartphones and tablets) and touch-screens (e.g. museum kiosks) a multi-touch interaction is provided, while on desktop devices (laptops, PCs) a keyboard+mouse model is enabled. On immersive VR/AR devices, different interaction models are automatically enabled, depending on HMD degrees-of-freedom (3-DoF or 6-DoF). There are different navigation modes in ATON that can be activated, available depending on the typology of the user device: A) orbit mode (default); B) first-person mode; C) device-orientation mode; and C) immersive VR navigation mode.

Orbit mode is a classic navigation model offered by the vast majority of Web3D presenters: in ATON it offers re-targeting features (double-tap/double click on surfaces to adjust camera target) with smooth transitions for a good user experience. *First-person mode* allows the user to explore the environment through a common point-and-go model (see fig. 9, top right). If no custom constraints or interaction models are provided, eligible locomotion areas are determined at runtime through an algorithm similar to the one adopted by SketchFab, depending on surface normals. *Device orientation mode* (available on mobile devices) accesses user device built-in sensors and uses such information to control the virtual camera (see fig. 9, middle row): a model often used for augmented experiences targeting tourism [58], or the augmentation of museum displays [59]. All these navigation models take into account well-established and validated approaches in Web3D literature [60],[58] to interact with 3D content.

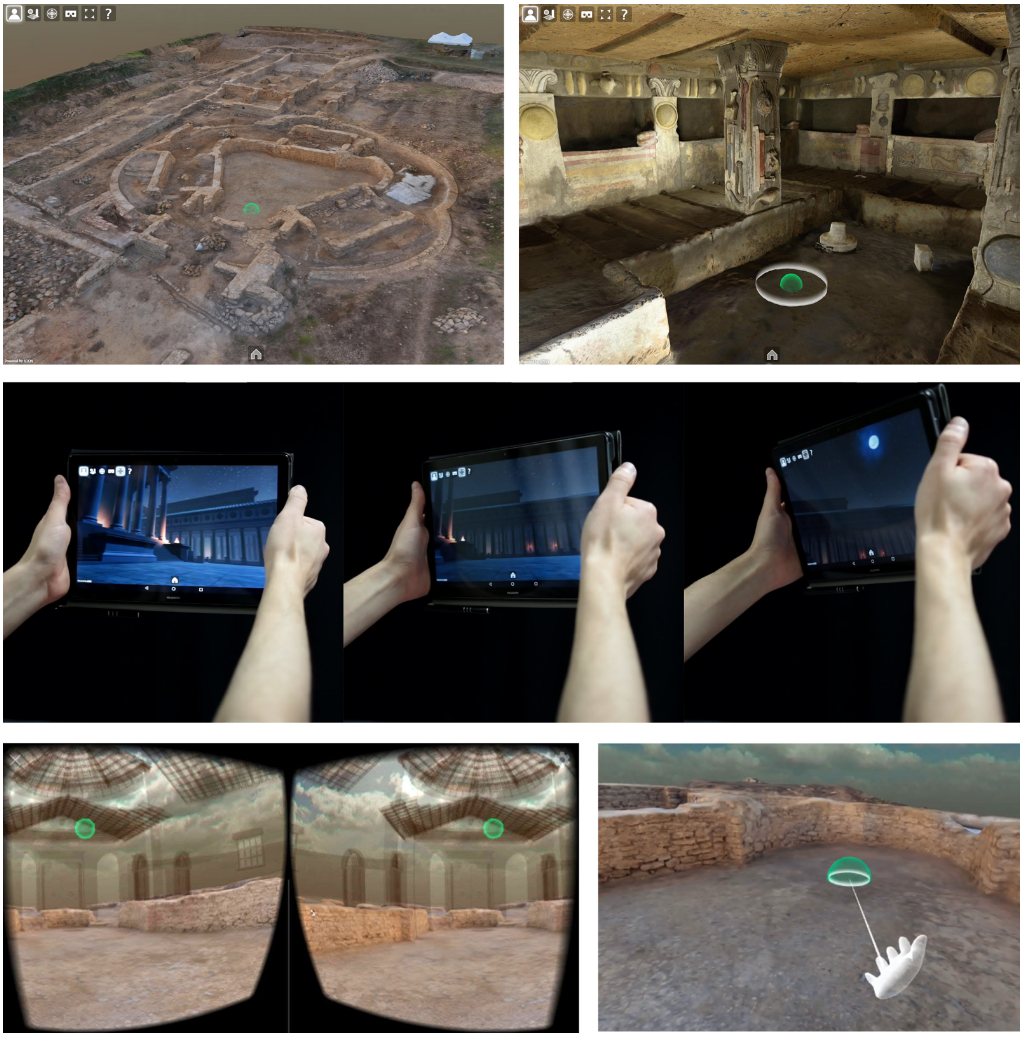

Figure 9. Different navigation modes. Top row: orbit (left) and first person (right); Middle row: device orientation mode; Bottom row: sample immersive view-aligned query/pointing on 3-DoF devices like cardboards (bottom left) and through 6-DoF VR controllers for locomotion on high-end HMDs (bottom right)

Regarding immersive VR, a locomotion technique based on teleport [61] is offered with specific transitions to minimize motion sickness [62]. Without specific constraints (e.g. locomotion nodes [63]) or interfaces, locomotion areas are automatically determined using the same approach of first-person mode. The system automatically adapts to 3-DoF and 6-Dof HMDs, switching pointing methods accordingly also depending on the presence of VR controllers (see fig. 9, bottom row). When no controllers are present (e.g. cardboards) a view-aligned / gaze pointing is activated, otherwise, one of the VR controllers is used [64],[65]. This allows the system to seamlessly adapt to 3-DoF (e.g. cardboards) and 6-DoF interaction models offered by high-end HMDs (e.g. Oculus Quest, HTC Vive, etc.).

Navigation system also provides structured viewpoints (or POV), which consist of eye location, target location and field-of-view. Particular focus was put on smooth transitions to guarantee a good user experience, also including field-of-view transitions. Furthermore, viewpoint transition requests are also correctly handled by the system in immersive VR modes (duration and orientation). Each POV in ATON possesses a specific ID, thus it can be easily recalled by a web-application, or updated in a JSON scene descriptor. A special POV is the home viewpoint, which - if not directly provided - is automatically computed by the navigation system, guaranteeing a correct initial location once all assets are loaded.

The system maintains a current POV (consistent with all navigation modes) that can be accessed and manipulated by custom routines. One example is the application of navigation constraints (to limit users movements) or adoption of locomotion graphs (move only into specific locations of the virtual environment), depending on the application requirements. These features are specifically designed to *quantize* the navigation in the 3D space - where needed - for all devices.

3.6.4. Query system

CH-oriented Web3D applications should provide interactive methods to query the virtual environment during exploration or inspection tasks. This is vital for semantic annotations (described in the next section), measuring tools or generic inspection: interactive routines are needed to perform intersections with complex or basic shapes. While for basic 3D models this is not particularly challenging (most Web3D libraries provide intersection methods with the underlying scene-graphs), several issues arise when the virtual environment becomes more complex or device resources are limited. For desktop-based web browsers, performance can be slightly impacted (small framerate drops) while for mobile devices and WebXR this can be devastating. The latter strictly require low-latency response times to deliver a consistent, smooth and acceptable experience for the final user while querying the space. A basic solution for Web3D/WebXR applications in general is to perform queries only on geometrically-simple shapes, although certain tasks (e.g. measuring) require intersection with more complex geometries.

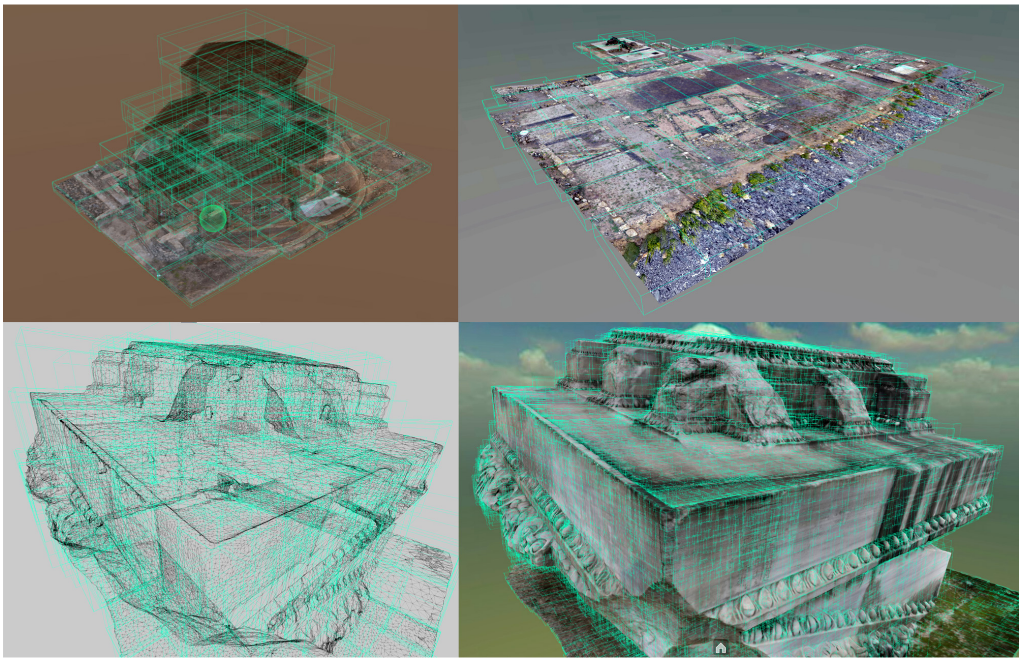

Figure 10. Sample BVH trees in ATON (green) to accelerate 3D queries

In order to overcome these issues meshes need to be spatially indexed [66]. *BVH* (Bounding Volume Hierarchy) trees are employed in ATON (see fig. 10) to query complex geometries very efficiently, while maintaining high frame rates [67]. This solution allows web-applications to query 3D surfaces very efficiently, for different reasons, including navigation purposes (orbit mode re-targeting, first-person locomotion), semantic annotations, measuring and spatial user interface elements. Furthermore, BVH trees can be also adopted for Cesium' 3D Tiles (see fig. 10, bottom row) to improve performances of interactive queries on large multi-resolution datasets. Current BVH implementation in the framework is based on an open-source library (https://github.com/gkjohnson/three-mesh-bvh) to accelerate ray-casting routines in Three.js. ATON offers an interactive 3D selector (visually represented as a sphere) with a location and radius that can be used to perform different tasks on queried surfaces.

3.6.5. Semantic annotations

Interactive Web3D applications in general [68] and specifically those targeting Cultural Heritage have strong requirements for the annotation of 3D models [69], linking sub-portion of a 3D object or scene to some related information presented to final users. Previous research already showed the importance of separating semantics from 3D representations [70],[71],[72]. Having a separated *semantic-graph* offers great advantages and flexibility when dealing with different data granularity, specifically separating the 3D rendering requirements of visible scene-graph (multi-resolution, hierarchical culling, cascading transformations, etc.) from semantic segmentation requirements. Building from previous literature, ATON adopts semantic 3D shapes as primary means to link information, with several advantages:

- They can be organized hierarchically (semantic graph), thus exploiting instancing and cascading transformations
- They can be produced by external 3D modeling software, semi-automatic algorithms or interactively generated by users at runtime
- They are suitable for 3D queries performed in immersive VR/AR sessions (through 3D intersection routines)
- They can be exploited as base elements to build more advanced formalisms

Each semantic node in ATON possesses a specific ID (e.g. "eyes", "floor01", etc.) with one or more children shapes, offering simple routines for web-applications to define their own behaviours when hovering or selecting shapes belonging to that ID. Common examples include showing a popup containing linked information, sliding informative panels, audio playback and much more.

Figure 11. Top row: basic (spherical) annotations interactively added using current 3D selector location and radius, with multiple shapes under the same semantic node ID (e.g.: “eyes”, top right). Bottom row: free-form semantic annotations interactively created at runtime using multiple surface points at different scales

Regarding user-generated shapes, the framework allows two different approaches for interactive annotation at runtime: 1) *Basic*: spherical shapes (location and radius) and 2) *Free-form*: shapes progressively built from points interactively placed on queried surfaces or in mid-air (convex-hull algorithm). In terms of network transmission, the first is indeed more compact, since a single shape can be described by 4 values (location coordinates + radius), while a free-form shape requires a list of 3D coordinates (4 points at least). Both semantic shape types can be exported into glTF or OBJ formats, directly using underneath ATON routines: a Web3D tool can thus provide user interfaces (UIs) to selectively download nodes in the *semantic-graph*. For specific workflows with dedicated UIs, such a feature allows professionals to interactively and easily create semantic shapes (annotations) - using mouse, fingers, stylus pens or VR controllers - and then reuse such shapes in other 3D modeling software.

3.6.6. User interface blueprints and spatial UI

Regarding *user interface*, ATON offers several built-in UI elements (HTML5) to boost the creation or prototyping of web-applications. They consist of buttons (e.g. home viewpoint, enter VR mode, etc.), toolbars, modal popups, etc. - with fully responsive support, guaranteeing a smooth, automatic adaptation to different screen sizes. They can be easily themed with custom CSS (cascading style sheets), while developers can attach custom routines to them. Furthermore, built-in elements (like buttons) work in combination with the profiler, enabling the creation of consistent user interfaces across multiple devices. A few examples are the immersive VR mode (only showing on secure connections and supported devices) or device-orientation navigation mode (only showing on mobile devices supporting this feature). The web-application can thus exploit these UI elements, or create their own using common HTML5 and javascript/ES6 functionalities.

The framework provides a built-in *spatial UI* (user interface elements living in the 3D space) specifically targeting immersive AR/VR sessions. These components were designed on top of existing guidelines [73] and design patterns related to 3D user interfaces [1], immersive VR principles targeting education [74] and immersive UIs for virtual museums [75]. The spatial UI module provides developers with buttons, 3D toolbars, labels, panels, and dynamic 3D text rendering features that allows web-applications to arrange interactive elements inside the scene.

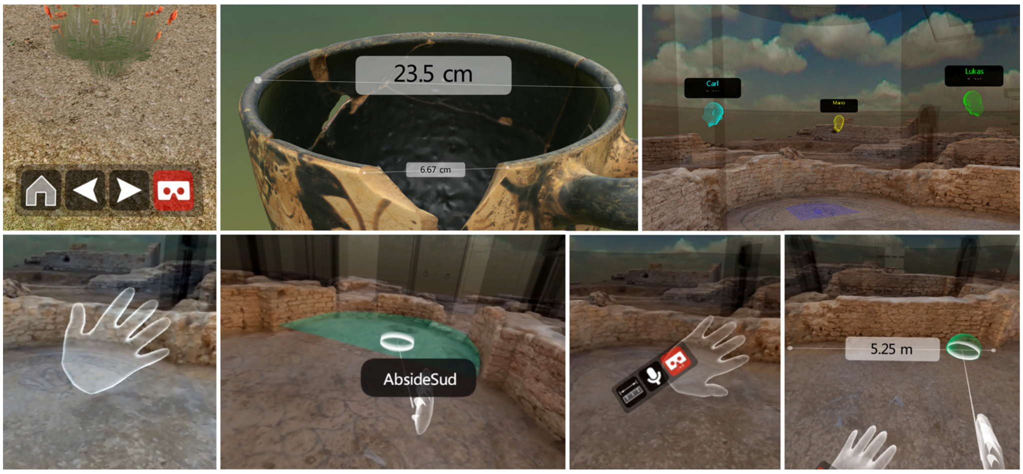

Figure 12. A few applications of spatial UI elements. Top row (from left to right): 3D toolbars in the virtual space with custom events, multiple measurements, 3D floating labels. Bottom row (from left to right): immersive VR hands, semantic labels (VR), wrist interfaces and immersive VR measurements

Since they are designed as nodes, they are managed within a UI-graph, thus they can be freely reorganized or transformed for different purposes. A few examples involve arranging informative panels inside the environment, attaching floating 3D labels to semantic shapes or placing buttons (triggering custom events) within local or absolute coordinate systems. These elements are consistent with - and specifically indicated for - immersive VR visualization. They also allow the creation of wrist interfaces [cit M. Alger] for virtual hands with ease, attaching specific functionalities. A few examples are related to measurement in WebXR sessions, teleport to predefined locations, enabling/disabling temporal layers, and much more. The spatial UI component offers blueprints to build custom spatial interfaces, depending on the specific WebXR application requirements.

3.7. Collaborative Sessions

One of the main contributions of the framework is the built-in collaboration component, enabling remote users to access synchronous, real-time collaborative sessions within virtual 3D environments - using a web browser. As anticipated in the state of the art section, a few open-source projects are already investigating this type of solutions (see "Mozilla Hubs" in section 2) and due to the covid-19 pandemic, such features are even more desired for distance-learning and online tools for education. ATON framework provides a collaborative system targeting Cultural Heritage field called "*VRoadcast*" consisting of client-side and server-side components. Such contribution has the goal of developing a social VR layer on top of existing features described in the previous sections. The most adopted approach by web-based and desktop-based solutions for social VR (see for instance VRChat, Mozilla Hubs, etc.) is the *room* concept: multiple users access a uniquely identified virtual space where they can interact in a synchronous manner, also inviting other users (typically by sharing a link). Within the ATON framework this is a session ID - that usually corresponds to a scene ID (see section 3.2) - exploited to manage multiple collaborative sessions on the DN. From a technical perspective, real-time communications are realized through web-socket protocol, specifically exploiting Socket.io [76] that is widely used for the development of node.js applications which include real-time communications. Multiple collaborative sessions can be created on a single DN (with different participants for each scene), thus depending on the scenario (see section 3.1), users may interact with other users in local networks (no internet connection required) or through an internet connection (world-wide). This allows great flexibility and, more importantly, no dependency on external (or proprietary) services. Furthermore, thanks to the microservice design (see section 2) it is possible to disable or independently control the service without impacting other services operating on the DN. The collaborative service was already employed and assessed in several online classrooms and workshops, with several students connecting to a public server node from different locations.

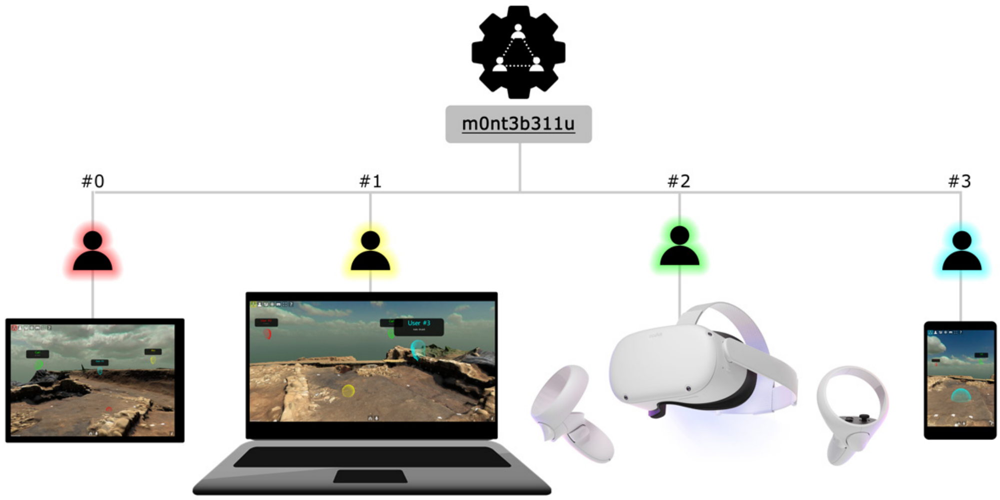

Figure 13. A collaborative session with ID "m0nt3b311u" involving multiple remote users, with different devices

In a given collaborative scene, the service allows to:

- map in real-time other users locations and orientations in the 3D space (represented as basic avatars);
- broadcast user states' updates (e.g. username);
- stream audio (talking through the device built-in microphone);
- perform collaborative modifications to the scene thanks to scene patches (see section 3.5).

Regarding user states' attributes that change very frequently (e.g. location, orientation) there are indeed several approaches already explored in literature to optimize network traffic. VRoadcast communications exploit existing design patterns, also including data quantization and state interpolation.

Users in a collaborative session can indeed interact through completely different devices, thanks to the liquid presentation layer (see section 3.6). It is thus possible to collaborate together in the same 3D scene using mobile devices (smartphones/tablets), PC/laptops, museum kiosks or immersive VR devices (HMDs) connected to the same DN - without any installation required. As shown by other works and results presented later in this paper, it elevates the experience creating collaborative Web3D/WebXR spaces where users can virtually meet to discuss, while operating from remote locations.

3.7.1. Requesting a collaborative session

When one user requests to join a collaborative session ID, the VRoadcast service retrieves the session (if it already exists) or creates it. Once the request is accepted, the service assigns a unique ID to the user that is maintained until he/she leaves the session. Each user is assigned a specific color (6 cyclic colors are employed): this visually facilitates the identification of other participants in the 3D space. There is an upper bound capacity of 255 users per scene, although such quantities are hardly reached in practical tests and applications (especially for simple scenes). The service takes care of users entering or leaving the scene, appropriately broadcasting specific messages to scene participants. When the last user leaves a given 3D scene, the related session is destroyed.

3.7.2. Customization and extensibility

VRoadcast provides built-in communications for transmitting user states and audio data, alongside basic messages exchanged between the DN and the clients. After previous assessments although, the standard set of collaborative features soon became a limitation for web-applications willing to define their own custom events. For instance, specific CH web-applications, applied games or tools may need to broadcast certain communications to other peers (e.g. toggling a node/layer, adding annotations, measurements, etc.). For this reason, a crucial step was to introduce in the framework the possibility for client applications to easily define their own network logic, enabling custom collaborative behaviors. This is in practice realized through a simple API to fire or subscribe to network events, opening endless opportunities for developers to define their own collaborative events with custom data exchanged in real-time within a 3D scene. Furthermore, since they are defined at web-applications level (client-side) they do not interfere with other collaborative web-applications deployed on the same DN. A vivid example of web-application taking advantage of the collaborative layer customization is the front-end "*Hathor*": an overview of this web-application is described in the next section.

3.8. Hathor

"*Hathor*" is the official, built-in front-end of the ATON framework, taking advantage of all the features described in the previous sections. The need for such web-application comes from requirements highlighted by the communities during the development of ATON and during previous projects and experiences. With the inclusion of Hathor, museums, professionals and general stakeholders, have three available scenarios with Hathor:

1. They have a built-in, maintained web-application to present 3D scenes and collections with advanced features (real-time collaboration, presentation settings, interactive annotations, etc.) with no coding requirements or developers involved (use "as it is")
2. They extend or adapt the functionalities of Hathor with little coding efforts
3. They develop their own solution (custom web-application) on top of ATON components depending on specific requirements

The main goal of the front-end is to provide a web-application to present 3D scenes to different users, exploiting the underneath ATON components. Hathor consumes just one parameter, a scene ID, to load and present the virtual environment and its associated data (viewpoints, semantic annotations, etc.).

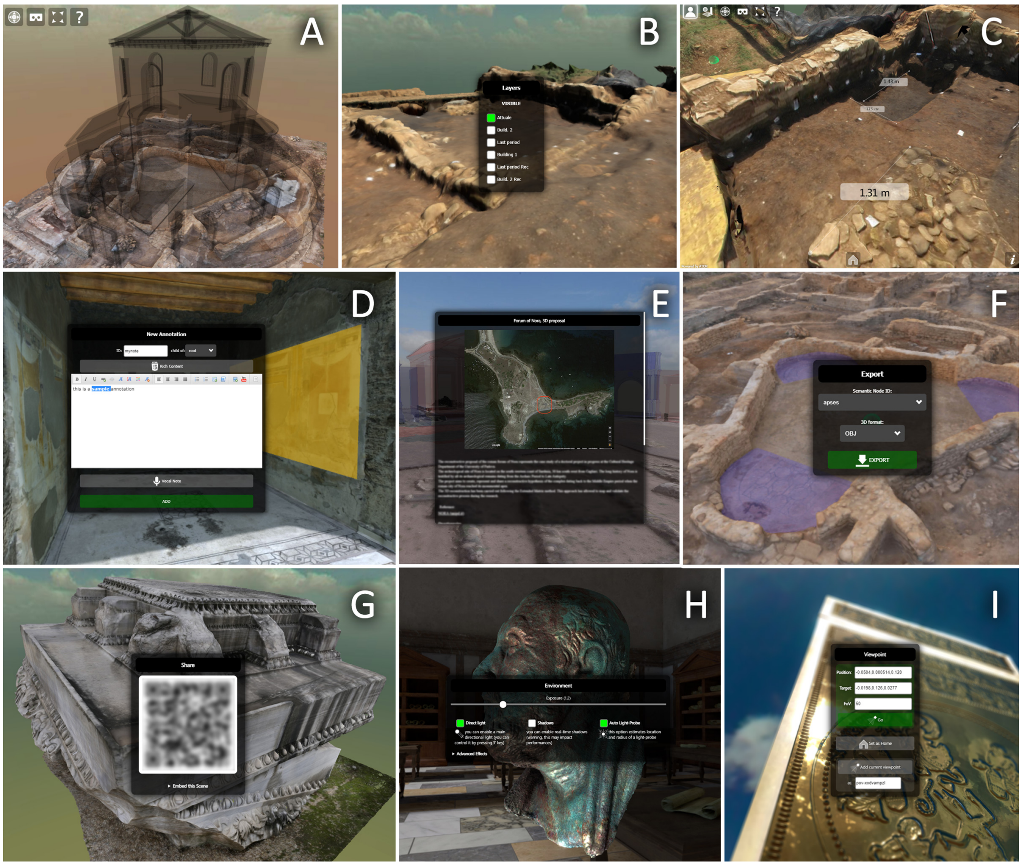

Figure 14. Sample captures from Hathor front-end. A) sample scene presentation and basic UI; B) layer switching; C) multiple measurements added by the user; D) HTML5 built-in editor and vocal notes for semantic annotations; E) user-created rich HTML5 content; F) semantic shapes export; G) sharing options; H) environment and lighting settings; I) viewpoint options

Hathor, as well as other web-apps developed on top of the ATON framework, is compliant with the PWA model (see section 2) thus developing a new model of app distribution within the mobile world, with a growing integration with the device. Furthermore, the front-end is compliant with the Open Graph protocol by Facebook (https://ogp.me/), thus providing improved sharing features when consuming 3D scenes, besides automatic QR-code (see fig. 14, G) and embed options. The basic interface (for the general public and non-authenticated users) offers basic tasks like navigation (viewpoints, immersive VR mode), layers control (show/hide scene-graph nodes - for instance switching between present and reconstruction), environment settings (dynamic lighting, shadows, advanced effects, etc.) and sharing options (embed interactive 3D view, QR-code). A built-in help is available to illustrate different functionalities and keyboard shortcuts, it also shows contextualized support depending on the detected device.

Regarding *semantic annotations* (for authenticated users), it is possible to interactively add basic or free-form shapes on top of queried surfaces, and assign them rich HTML5 content. This is possible through a built-in WYSIWYG[23] editor (see fig. 14, D), that gives users complete freedom in terms of content type and complexity (formatted text, images, youtube videos, audio, embedded pages or generic HTML5) as seen in fig. 14, E. Authors and editors are presented with an easy to use interface, while rich content is stored into the JSON scene descriptor. It is also possible to record vocal notes assigned with specific semantic nodes: this is particularly indicated for scenes targeting immersive VR, where users query the 3D space and listen to vocal notes made by remote editors. The entire

[23] What You See Is What You Get

semantic-graph (imported or user-generated shapes) can be exported directly (see fig. 14, F) from the browser (see section 3.6.5).

Hathor offers the possibility to add multiple measurements into the 3D space, a feature specifically useful for CH professionals. These exploit the spatial UI offered by ATON to be consistent with immersive AR/VR sessions, and automatically adapt to different scales (see fig. 14, C).

In order to apply scene changes at runtime (see section 3.5), Hathor provides *temporary* changes (changes that do not modify the server-side JSON scene descriptor) and *persistent* changes (modifications altering the JSON descriptor). For the general public (without authentication on the DN) only temporary changes are possible (they are lost on page refresh) while editors are able to enable persistent modifications. This is particularly useful during collaborative sessions that do not intend to alter the original 3D scene setup, but rather show other participants temporary modifications (e.g. showing/hiding a layer, adding temporary annotations or measurements, etc.) for various purposes.

Hathor fully exploits the *collaborative* components of the ATON framework (see section 3.7). A single button allows the user to switch between "single" and "collaborative" session, joining a specific session associated with the current scene ID. A basic chat panel is provided between participants, and it is possible to talk in real-time with others through the built-in microphone of the device (mobile, desktop PC, HMD) similarly to *Mozilla Hubs*. This is possible through a WebRTC library that allows to stream audio, video or screen activity as well - and broadcast them to other users in the scene.

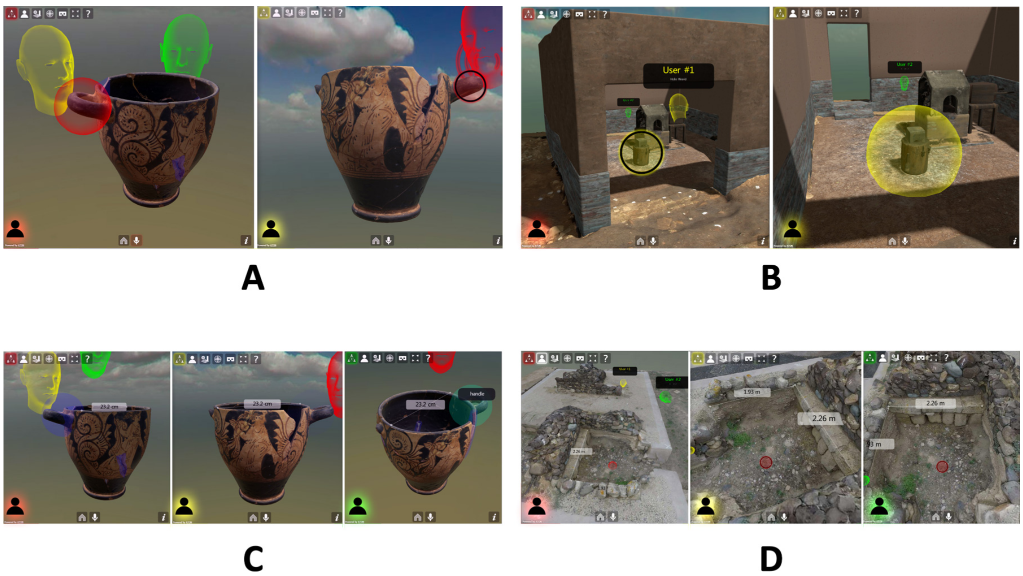

Figure 15. Sample collaborative sessions in Hathor with 3 users (red, yellow and green). A) User 0 (red, left view) is streaming its focus to other participants, right is yellow user view. B) User 1 (yellow) is streaming its focus, and changed lighting settings at runtime (red view left, yellow view right). C, D) All three users perform annotations and measurement tasks at different scales

During a collaborative session, scene modifications are broadcast to participants: environment settings, lighting changes, semantic annotations, measurements, etc. are all synchronized in real-time, thanks to scene patches (see section 3.5). It is thus possible to collaboratively enrich the scene with semantic annotations, measurements, as well as switch layers, testing different lighting setups and much more. A particularly useful tool for collaborative CH sessions offered by Hathor is the *focus streaming*. It is possible for a participant to broadcast his/her focal point in the form of a pulsating visual 3D indicator on the location where he/she intends to raise attention (see fig. 15). As highlighted by different experiments, these features enable rich collaborative experiences, for instance discussing a specific 3D scene or reconstruction hypothesis, directly online, with every device and without any installation required for participants.

4. Experiments and Results

Different components of the ATON framework and their interplay have been already investigated and assessed by previous work during the past few years. More specifically: A) Framework integration with cloud architectures for publishing multi-resolution 3D landscapes within European infrastructures [50] and its exploitation for cross-device, online virtual museums [77]; B) Presentation of virtual 3D collections and augmentation of past museum displays on mobile devices (Capitoline Museum in Rome) [59]; C) Creation of applied CH games as web-applications [78],[79]; and D) Visual/immersive analytics architectures for remote inspection of interactive WebXR sessions [80],[81].

This section thus describes and reports a selection of experiments carried out on different case studies, to assess novel components of the ATON framework (light-probing, annotations, cross-device presentation and collaborative modules) with related results and discussion. Each section reports results for the given case study, while a final discussion section (5) will overview obtained results, their interplay with the other framework components, and implications in a broadest context.

4.1. Deployment Node setup

For carrying out the experiments in the next sections, a lab server with a public IP and a certified domain (for secure connection) was used. The machine was equipped with an Intel Core i7-3820 3.6GHz and 16Gb RAM. The server was configured with a debian-based Linux OS and Node.js to deploy the ATON services through the PM2 process manager (see section 3.1) in cluster mode with 4 cores (2 threads each) handling incoming client connections, remote requests and collaborative sessions (including users' states and audio streaming). The framework was configured with a few users who created, edited and published several 3D scenes through Shu back-end (see section 3.4). Access to the collections was possible through an integration of data folder with *NextCloud* (https://nextcloud.com/) - on the same server - allowing users to easily and privately manage their content.

4.2. Chrysippus head

The *Chrysippus of Soli* is a 1st century AD bronze head (overall height of 14.5 cm) originally located in the library of the Forum of Peace in Rome and nowadays exposed permanently in the Museum of the Imperial Fora, in the near Trajan's Markets. This portrait of the well-known philosopher of antiquity has a largely oxidised surface, hence the characteristic green colour. This work of art, well known in antiquity, was digitised for research applications using photogrammetric techniques in 2012 (software: Photoscan 1.0 from Agisoft) as part of the European 3D-ICONS (3dicons-project.eu) project and published in the Europeana digital library (www.europeana.eu). In 2021, the dataset was re-processed with the current photogrammetric tools (Metashape 1.7) and re-elaborated according to the PBR material surface representation paradigm (see section 2). The model creation workflow then focused on the creation of albedo, roughness, metalness and normal maps with a resolution of 4096×4096 pixels (see Fig. 16).

area (sqm)	original tris	tris n.	tris/m	tex res	uv ratio	mm/pixel
0.04	91M	50k	1.3M	4096	0.6	0.06

Table 1: Metrics of the model of Chrysippus: area expressed in sqm, number of triangles of the original and reduced models as well as the geometrical resolution (number of tris per meter) and the texture resolution (considering that the uv mapping uses actually just 60% of the full atlas texture).

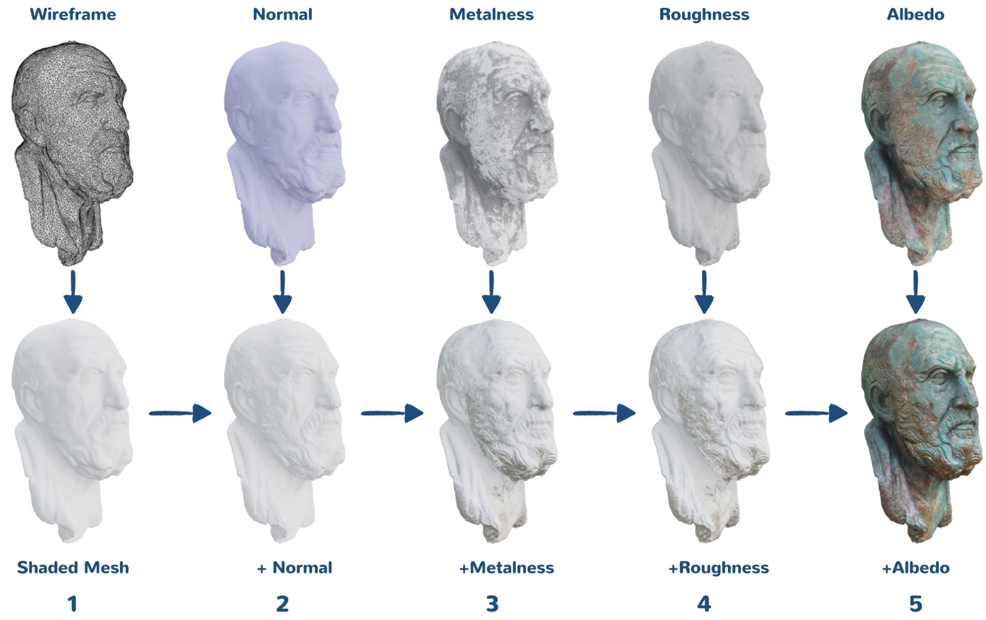

Figure 16. Full Chrysippus 3D model workflow

The photogrammetric survey produced a model with a high geometric resolution (91 million polygons for an object of 14.5 cm in height). The optimisation process that was followed involved the creation of a simplified mesh (53k polygons) on which the geometric information was reported through the use of a normal map. Through the use of reference images of the original object, taken under different light conditions (grazing light, direct light, etc.) and through chromatic filtering operated on the albedo map, the metalness maps (the green areas are equivalent to oxidised material, while the remaining areas are the visible intact bronze) and the roughness maps (extraction of the micro-relief from the colour information contained in the albedo) were obtained. The albedo was obtained by means of delighting algorithms (software: Metashape de-lighter 1.7) starting from the diffuse channel already obtained by transferring the photographs onto the geometry within the Metashape photogrammetry software.

4.2.1. Results

The PBR model made it possible to address a very hot topic in the field of virtual reconstructions of antiquity, namely the difference in perception of the artefacts by ancient versus contemporary audiences. As already mentioned, originally the Chrysippus was located inside the library of the *Forum Pacis* while it is now inside a museum case in the nearby museum of Trajan's markets. The two different locations dramatically influence the lighting conditions and, consequently, the chromatic perception of this object (see Fig. 17).

In other words, placing the same PBR model within two different scenarios (ancient and modern contexts) results in two different visualizations. The potential of an online dissemination capable of conveying these nuances opens up new communication perspectives by guaranteeing a more realistic and scientifically validated perception of an ancient artefact.

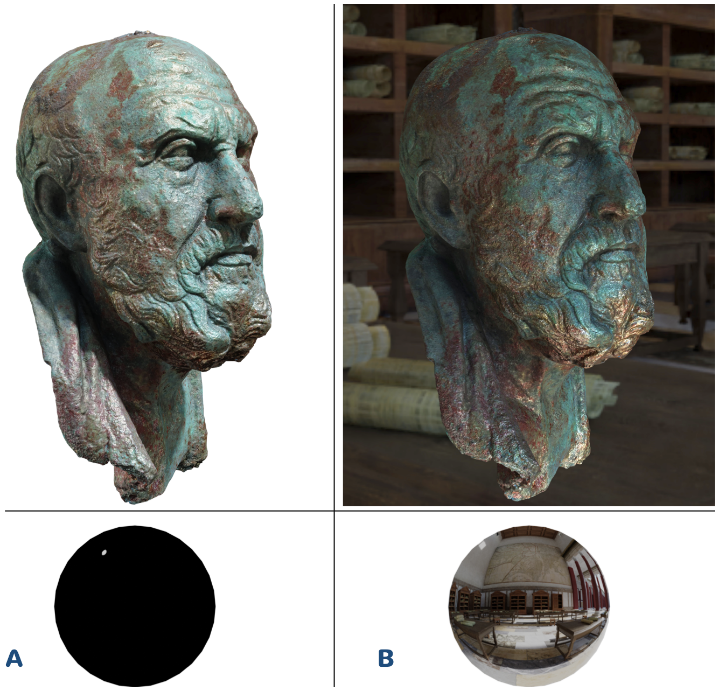

Figure 17. Chrysippus model rendered using a physically, path tracing, based production renderer (Cycles within Blender 2.93) and environment lights used: A) a spot light similar to the museum exposition and B) a reconstructed environment (E.Demetrescu) of the Forum of Pacis.

In ATON, two different scenes were created referencing the same 3D asset "Chrysippus head" in glTF format (see section 3.2) including all PBR textures. The two scenes allowed to simulate different scenarios and lighting conditions, modern lighting setup (A) and original context (B) - see fig. 18. For the first scene (A), a black background and a directional light were used to simulate the spot used to illuminate the bronze head in the Museum. For the second scene (B), an HDR panorama of the *Forum Pacis* was employed to simulate original lighting conditions.

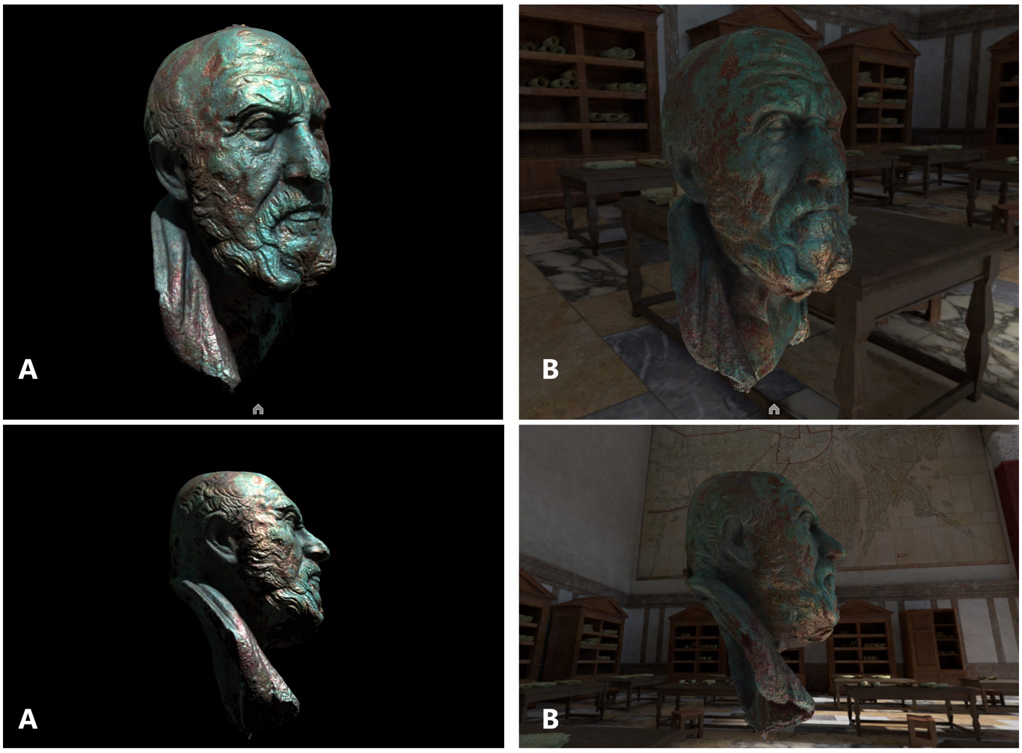

Figure 18. Interactive visualization in ATON of the same item using two different environments and lighting setups

A different setup with tinted panels was also created to assess multiple light-probes (see section 3.6.2) arranged in a 3D scene with 3 instances of the Chrysippus head. Four light-probes per instance (a total of 12) were placed in the 3D scene (see fig. 19, top) in order to simulate local reflections and illumination. As shown in fig. 19 (middle) the three instances react consistently to their surroundings (tinted panels), thanks to light-probes capturing local details and applying them to associated 3D meshes and their PBR model. The testbed (publicly accessible on https://aton.ispc.cnr.it/examples/lightprobes/) also allows to switch individual sections and panels at runtime, to prove how these elements affect local reflections and illumination. It is shown for instance in fig. 19 (bottom) how hiding or showing the white panel affects the back lighting on the 3 heads (and reflective spheres).

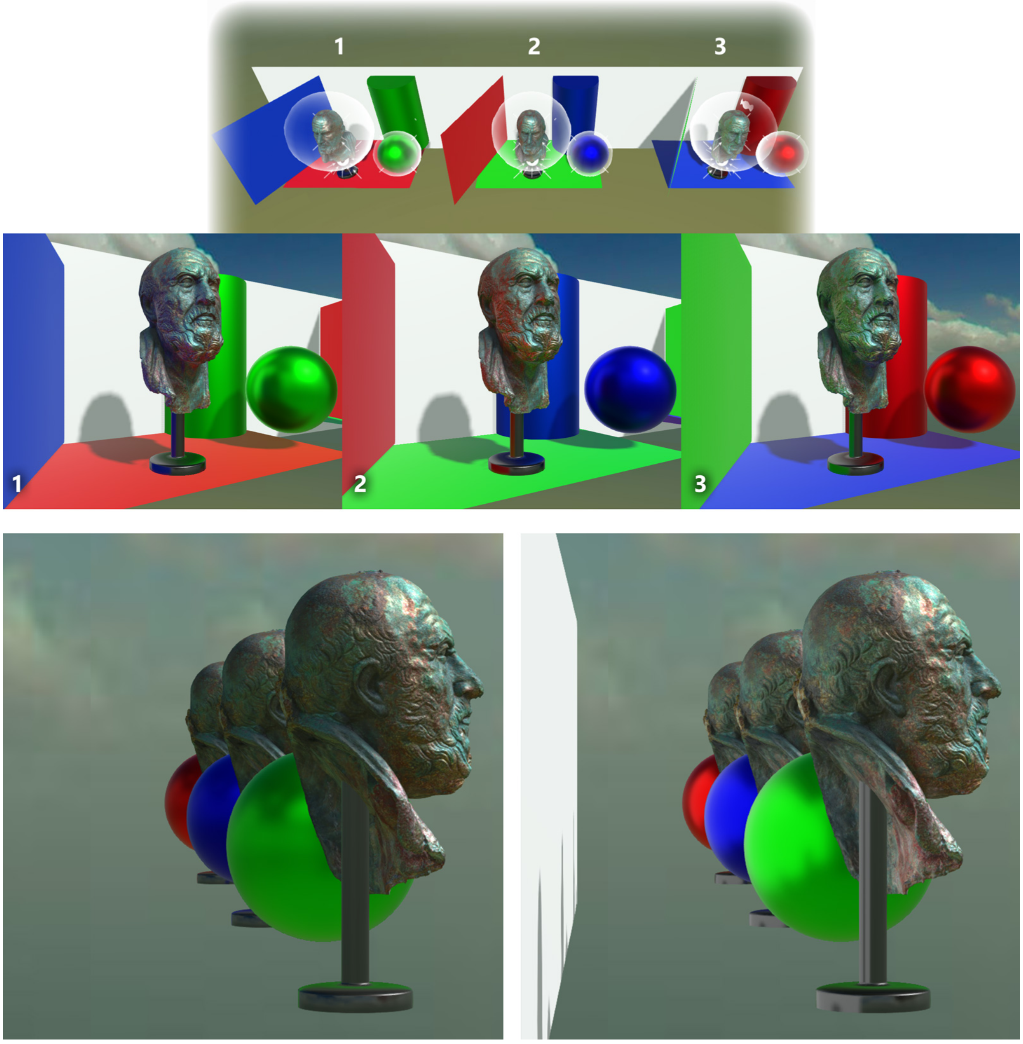

Figure 19. A different setup to assess multiple light-probes on 3 instances of the 3D model

4.3. The roman villa of Aiano

The Roman villa of Aiano is an Italian archaeological site in Tuscany, close to San Gimignano and its remains date back between the end of the 3rd and the 7th century A.D. Since 2005 the villa has been excavated by an Italian-Belgian mission coordinated by the UCLouvain as part of the international project "VII Regio. The Elsa Valley during Roman Age and Late Antiquity" [82]. During the research, a 3D model of the so-called trefoil hall was performed in collaboration with the ISPC-CNR in order to simulate a possible reconstruction of the archaeological remains, characterised by monumental architecture and decorations, and, above all, to better understand the architectural evolution phases of the hall. Three different model have been produced: 1) a digital replica model of the site in its current state of preservation, obtained with image-based modelling techniques; 2) a schematic semantic model which allows to query information and sources used in the reconstructive process and visualise level of certainty using different colour coding to distinguish extant structures from virtual reconstruction; 3) a realistic virtual reconstruction which simulates the building in its formal unity and in its hypothetical aspect at the end of the 4th century A.D., to improve legibility of the hall [83]. The digital replica model was exported in glTF format in order to assess vocal annotations offered by Hathor (see section 3.8) by means of semantic shapes on an ATON scene.

Different vocal notes have been recorded and associated with semantic shapes, placed on specific points of interest like architectural decorations or structures. Figure 20 shows an example of this workflow: the Western apse of the hall shows an *opus signinum* floor decoration in which the *tesserae* form the image of a vase (*Kantaros*). Using the free-form shape annotation tool, the area of the apse was drawn. Then, a vocal note, in which the peculiarities of the floor decoration are highlighted, has been recorded using the built-in PC microphone.

Figure 20. Top row: vocal annotation workflow: free-form shape annotation of the floor decoration, new annotation ID and voice recording. Bottom row: the Aiano 3D scene with audio playback on user activation of annotated areas, from PC and HMD (Oculus Quest) using VR controllers

4.3.1. Results

The identified area for the experiment was segmented in 16 blocks for the reality based layer (modern), in addition to the reconstruction layer (semi-transparent volumes) on top (see fig. 20, bottom left). The blocks (obj format) and referenced textures were converted to the glTF format with Draco compression for geometries, populating the cloud collection in order to assemble the scene. As shown in table 2 Draco compression provided by Cesium tools (https://github.com/CesiumGS/gltf-pipeline) was particularly effective on original obj size (106Mb compressed in 2.43Mb) using compression level = 4.

layer	tris n.	textures n.	texture res.	geom (OBJ)	geom (Draco)
Modern (RB)	854k	57	2048	106Mb	2.43Mb
Reconstruction	81k	0	-	8Mb	7Kb

Table 2: Metrics about the 3D assets referenced in the scene (identified area for the experiment)

After the creation and publication of the 3D scene on the server, the entire annotation workflow was carried out autonomously by an authenticated researcher directly through Hathor front-end, using a web browser. First, scene cloning (see section 3.2) allowed the researcher to duplicate the original 3D scene (available at https://aton.ispc.cnr.it/s/vhlab/demo/aiano) for the experiment, thus reusing the same assets in a different scene (with a different scene ID). On the new scene, the time spent to annotate 5 different areas using the free-form annotation tool (see fig. 20) was around 6 minutes, using mouse and keyboard on a desktop PC equipped with a NVIDIA GTX 980 GPU.

Once edited, the 3D scene including voice annotations was accessed using different devices (PC, mobile smartphone and tablet, HMD) recording average fps for 4 minutes. In order to assess cross-device presentation features, the same 3D scene was accessed and consumed by mobile devices (android smartphone and tablet) and one HMD (Oculus

Quest 2). All devices allowed a group of 6 people to explore and query annotated areas (see section 3.6.5) at interactive framerates (as reported in table 3) and listen to vocal notes created by the researcher. For Oculus Quest HMDs the official Oculus browser was used, while other devices adopted Chrome.

Device	OS	GPU	avg. FPS
Honor v10 (smartphone)	Android 9.0	Mali-G51	56.3
Huawei MediaPad M5 (tablet)	Android 8.0	Mali-G71	54.2
Apple iPhone 12 (smartphone)	iOS	Apple GPU	49.7
PC (workstation)	Windows 10	NVIDIA GTX 980	59.9
Oculus Quest v1 (HMD)	based on Android	Adreno 540	66.4
Oculus Quest v2 (HMD)	based on Android	Adreno 650	72.1

Table 3: Average framerates (4 minutes) for each device

4.4. The Forum of Nora

The roman forum of Nora constitutes the case study of a doctoral project in progress at the Cultural Heritage Department of the University of Padua on behalf of the author of this paragraph. The project aims to propose a 3D reconstruction of the entire forum by following the Extended Matrix (EM) approach [84]. The research purpose consisted of improving the general comprehension of the archaeological remains of the context and creating an online 3D scene accessible on the web. The site of Nora is located on a peninsula on the southwestern coast of Sardinia, about 30 km southwest of Cagliari. The forum, the ancient roman square used as case study for this work, occupies an area of approximately 3500 sqm (including its annexes) in the eastern sector of the ancient city [85].

In order to assess collaborative features offered by the framework, we present in this section outcomes from a multi-user session where 11 users accessed the same 3D scene online, from different physical locations and with different devices. The main goal was to share with multiple participants a guided tour of the 3D reconstruction of the roman forum of Nora, using *Hathor* front-end (see section 3.8). The tests aimed to assess the collaborative service during a multi-user session and also to collect a general feedback of the experience. During the tour users had the possibility to freely explore the scene and, most importantly, to interact with each other using their voice and other tools.

The virtual scene was composed by 3D models organized in 5 layers ("Contemporary", "Period IV", "Period IV rec", "Period V", "Period V rec"). The first layer, the base from which the 3D reconstructive proposal was then realized, represents the photogrammetric model of the area (Contemporary layer). The additional 4 layers correspond to the extant archaeological remains ("Period IV": red, and "Period V": blue), dealing with the two chronological phases analyzed with the project, and their virtual reconstruction ("Period IV rec": light red, and "Period V rec": light blue). The entire reconstructive proposal of the forum was carried out by using both archaeological, geometric and bibliographic data arranged and linked together following the EM approach. This method allows to map and visually represent the reconstructive process behind a virtual reconstruction of an archaeological context. With the exception of the contemporary layer, a semi-transparent material was used for the volumes of the additional 4 layers.

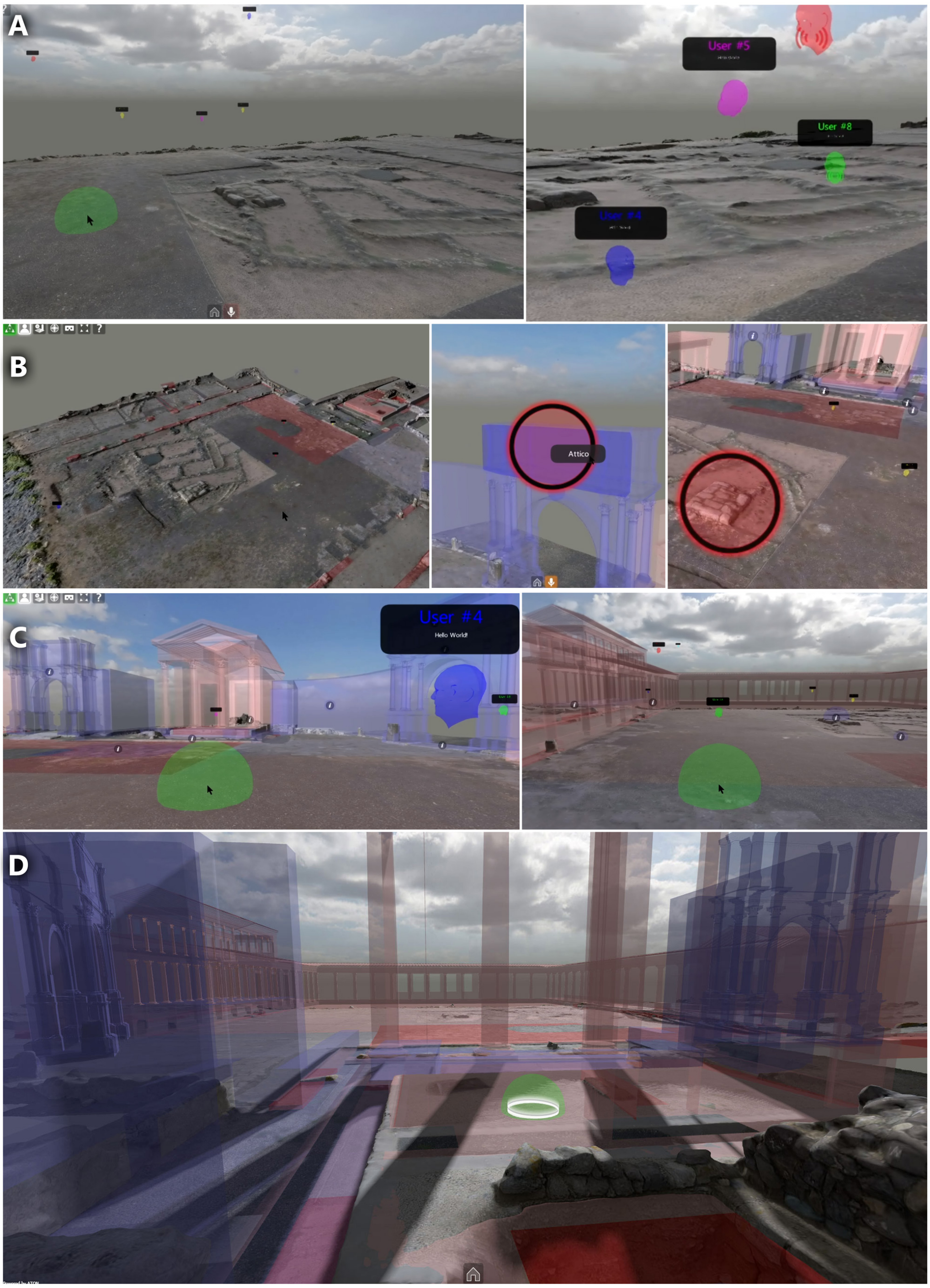

Figure 21. Captures from the collaborative session using a web browser. A) Participants enter the session and gather as the virtual guide (red) starts the explanation; B) The guide progressively activates reconstruction layers (semi-transparent volumes) and uses focus streaming to raise attention on specific hotspots; C) virtual discussion phases; D) The 3D scene with all layers activated

The test consisted in a virtual tour of the forum (see fig. 21) carried out by a guide (one of the users) in the reconstructive proposal, with different collaborative tools offered by Hathor available to participants. Before starting the tour, "navigation modes" (see section 3.6.3) and collaborative tools were introduced to participants to provide all the basics, with a "virtual guide" leading the tour in the 3D scene. Hathor offers the ability to enable persistent changes to the scene (see section 3.8), thus allowing both the guide and all the users to perform modifications in a synchronous manner. During the visit, both basic (spherical) and complex (free-form) annotation shapes were used to record information on the scene. The virtual guide employed basic annotations to signal the different reconstructed structures (temple, monumental accesses, basilica, curia). Annotated content was also enriched with images and other HTML5 media, with the intent of creating rich points of interest. For instance, to improve the understanding of the archaeological area from both a topographical and chronological perspective, basic annotations were created in the center of the forum with the reconstructive plan of the context. A similar annotation was placed as well inside the Basilica and enriched with images in order to provide further information on the mosaic floor. Complex annotations instead, were used to highlight structures, specific scene portions or areas. Furthermore, during the tour, the *focus streaming* feature (see section 3.8) was also employed by the virtual guide with different radii to point or raise the attention on details and areas while describing them. All the participants had the possibility to personally experiment the use of these tools (annotation, focus, layers and lighting) during the visit. This solution enabled a consistent interaction between them and the virtual guide. At the end of the tour, a questionnaire was administered to the participants: we report the outcomes from this survey to support the evaluation of the collaborative session.

4.4.1. Results

At the end of the collaborative session within the reconstructive proposal of the forum of Nora, statistics from the questionnaire and suggestions reported by users, were collected. In this section, the outcomes of this general evaluation of the service will be discussed. In particular, we focused on users locations, devices (hardware) used, collaborative features used including audio (talk) and overall feedback.

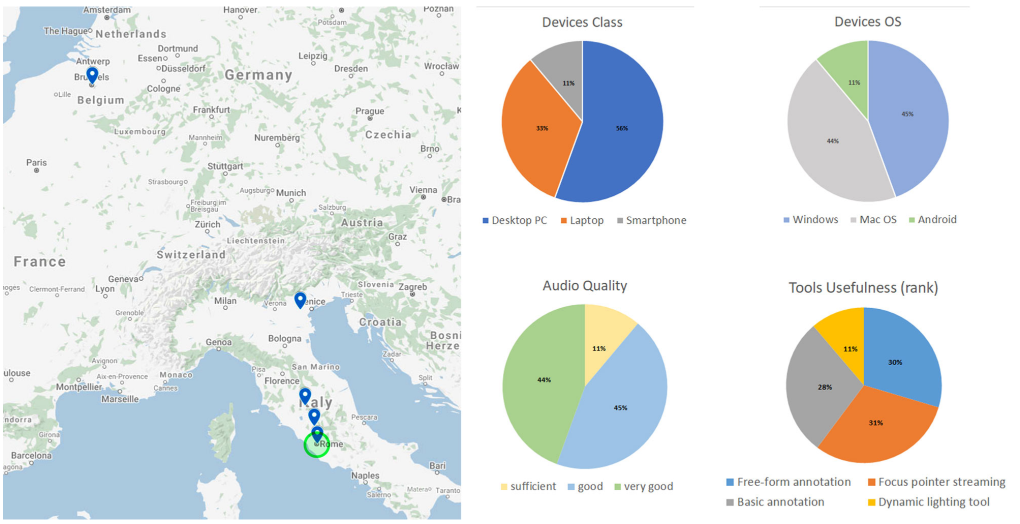

Figure 22. Nora collaborative experiment results

Geographical distribution. Most of the participants (90%) connected from Italy, one from Belgium. Within the Italian territory, 5 users were located in northern Italy and 3 in the center, with the server node (DN) located in Rome.

Hardware and software. The collaborative session was experienced from 3 different classes of devices: PC (56%), laptop (33%) and mobile smartphone (11%). Windows and Mac-OS were equally distributed as main OSes in 8 devices. With the exception of the mobile device (android smartphone), all the other devices employed Google Chrome to interact with the 3D scene.

Collaborative tools. During the session a few collaborative tools were introduced to users: 1) dynamic lighting; 2) focus pointer streaming; 3) layer switch; 4) basic (spherical) annotation; 5) free-form annotation. At the end of the session, users were invited to rank all the tools used from the most to the least useful. Users with mobile devices did not have the possibility to annotate on the scene (by design), thus this feature was evaluated by 8 users only. Outcomes revealed that the *focus pointer* was rated by participants as the most useful (31%), followed by free-form annotation (30%) and basic annotation tools (28%).

Audio quality. Throughout the virtual tour, communications were mainly made through voice. Thanks to the microphone of their devices, users easily spoke to each other as they were effectively visiting the site as in real life. Despite the presence of a limited delay, the audio quality was rated as good (45%) and very good (44%). A few users however, disagreed with the use of spatialized audio as it was perceived as uncomfortable.

Usefulness. Overall, the whole group of participants involved in the tour appreciated the collaborative session. From their personal experience most of them (66%) would use this online service for research purposes (for example, team collaboration). Others (44%) believe this service is also a useful instrument for education (virtual tours) and teaching (as a support during the lessons).

In general, all users appreciated the collaborative session: all of them agreed on the advantage of the experience being completely online, without the need for the installation of any external software. Some users disliked the use of spatialized audio for talking avatars, and also with the visual presence of avatar's heads. For the latter, they proposed an option to hide/show avatars' heads - excluding the virtual guide, considered as a reference point during the tour.

5. Discussion

This section presents an overall discussion drawn from the case studies and involved components employed during the experiments.

In terms of architecture, experiments in this paper (section 4) and previous works adopting the framework, highlighted the adaptability and responsiveness of both client and server components. On the presentation side, ATON profiler and scalable rendering system (see section 3.6.2) guarantee a responsive, universal and *liquid* consumption of 3D content on the web, ranging from mobile devices, PC, museum kiosks and 3-DoF or 6-DoF HMDs for immersive VR (e.g. Oculus Quest) which all reported interactive framerates. For instance the Aiano experiment (see section 4.3), did show how a published 3D scene in ATON can be consumed and queried interactively on a wide range of devices. Furthermore, the adoption of a *physically-based rendering* (PBR) model, the light-probing system offered by ATON and the glTF standard, as highlighted by the Chrysippus case study (section 4.2), allow to better simulate the 3D object material properties in relation to the surrounding environment and with drastically different lighting conditions. The introduction of the digital replication concept among tools used by CH sciences responds to the need to go beyond the documentation of geometry and diffuse colour to include the material characteristics of artefacts (how surface features react to light). In other words, with digital replication, there was a need for a formalism that could describe the surface material in a compact and simple way so as to make available a visual substitute for the original object. The metaphor of the PBR, now an industry standard, responds perfectly to this need, allowing, in addition, to transform the digital assets of Cultural Heritage into reusable assets for the creative industry. The greatest challenge related to the PBR metaphor is the lack of a universal workflow in the literature for creating roughness and metalness maps for three-dimensional objects. In general, roughness maps are obtained by [86]:

- painting them directly using 2D applications (Photoshop, Krita, Gimp, Blender, Zbrush, Substance Painter, ...);
- generating them from photographs (Substance B2M, Quixel, ...);
- generating them from procedural materials (Substance Designer, Blender, 3DMax, Maya, ...).

Acquiring the roughness value of a surface from real world data is possible: there are instruments such as box scanners for PBR materials, although they are currently designed to acquire flat surfaces. Creating roughness maps for 3D models requires semi-automatic approaches based on features extraction from images and specific skills in both CH and Computer Graphics in order to perform fine-tuning by comparison with real photos taken under different lighting conditions. Moreover, PBR models are also useful within *Augmented Reality* WebXR sessions [87], where lighting information can be estimated from the real world [88] creating a more consistent AR presentation and visual appearance (see fig. 23, A). The glTF standard including PBR materials moreover, is well supported on several software tools and game engines (like aforementioned Unreal Engine 4) allowing pipelines that directly export PBR assets to the web (see fig. 23, B).

Regarding Web3D rendering, Three.js (and other similar open-source libraries) allow more advanced effects targeting interactive presentation on the web, like *Screen-Space Reflections* (SSR) already adopted in proprietary platforms like SketchFab, or *Ground Truth Ambient Occlusion* (GTAO) [89], although introducing additional costs in terms of performance. Large open-source libraries like Three.js with active communities allow indeed to create a robust foundation for ATON, including custom visualization models (e.g. through the use of shaders) targeting the Cultural Heritage field, like real-time cross-sections or virtual lenses [67], slope gradients, outlines, etc. It also creates a friendly development environment for those who already employed or studied such well-known library for different Web3D projects, thus facilitating adoption of the framework.

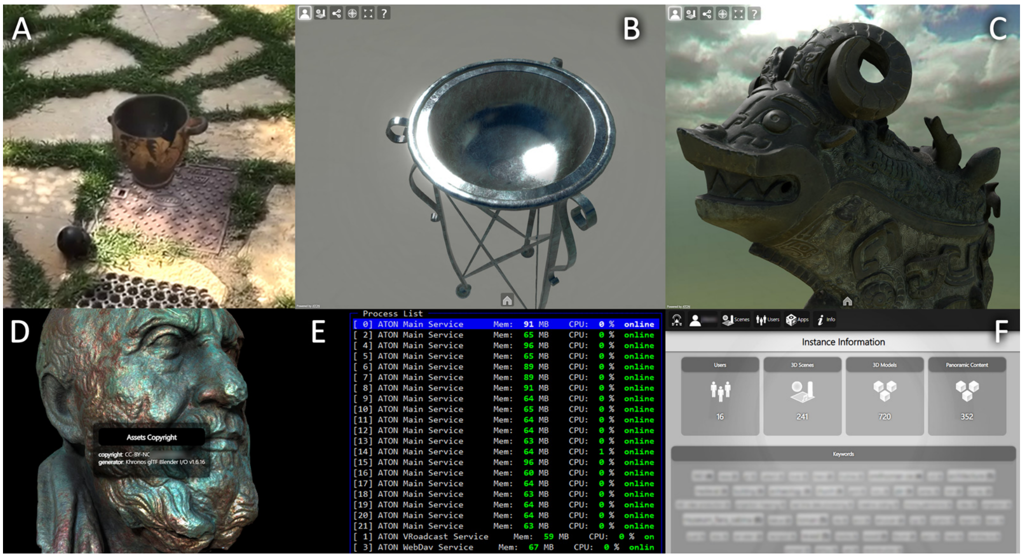

Figure 23. A) AR Presentation of a 3D scene; B) A PBR asset directly exported from Unreal Engine 4; C) A 3D scene referencing an external 3D model from Smithsonian CC0 collection; D) Inline copyright from glTF; E) Workload distribution among available cores; F) Instance content statistics

The decoupling of collections and scenes (see section 3.2) highlighted several advantages in terms of organization and reuse of existing assets for online publication (3D models, panoramic content, etc.). Both scenes and collections are assigned unique IDs within a given ATON instance, thus allowing to uniquely address these resources for public (or local) dissemination. The scene cloning was particularly useful to test different arrangements, hypotheses or lighting setups referencing the same resources (see sections 4.2 and 4.3). The lightweight scene descriptor (JSON) also easily allows versioning approaches on the server node, for instance to provide *snapshot* capabilities to client web-applications for a given 3D scene. Furthermore, the scene descriptor proved to be a powerful approach to reference hybrid content from local and remote collections, including public resources (for instance the open-access CC0 Smithsonian 3D models collection in glTF - see fig. 23, C). This also allows to reference assets across different

ATON servers in order to distribute content in multiple infrastructure nodes, where needed. Regarding referenced 3D items in scenes moreover, since the glTF format is highly extensible, it already allowed an initial support for *inline copyright* data (when present): these fields are already provided in glTF models published by SketchFab and many others, and the topic is indeed actively discussed in glTF communities. These fields are automatically recognised and extracted by ATON loading routines (see fig. 23, D), with copyright information such as author, license, source, etc. that can be presented (e.g. in Hathor) to final users.

Regarding interactive semantic annotation workflow for 3D scenes (see section 3.6.5), Aiano case study showed how easily a researcher can annotate different areas (section 4.3) using tools offered by the framework, but also highlighted a few interesting aspects. Besides the obvious outcomes in dissemination, such an approach has interesting practical implications in the scientific field regarding ATON front-ends like *Hathor*. Such tools can be easily used to study and discuss with other remote researchers, who autonomously (and without any expertise in 3D modeling) can add personal interpretations and spatialized annotations, directly editable into the 3D scene. Furthermore semantic 3D shape approach adopted in the architecture provides all the building blocks to create more sophisticated applications for professionals, supporting advanced semantic formalisms targeting CH, like the *Extended Matrix* [84] and the resulting EMviq tool[24] based on ATON, developed under SSHOC european project.

The Node.js ecosystem and *microservice* design of the framework (see section 3.1) on the other hand, guarantee maximum scalability in terms of deployment - ranging from low-cost hardware (such as single-board computers like Raspberry Pi) up to large national or international infrastructures. Regarding the experiments in this paper (see section 4) for instance, the memory footprint on the configured DN was fairly compact for each microservice, regularly in the 40-100 Mb range. When using more advanced server hardware, process managers such as PM2 (see section 3.1) allow to distribute workload among available cores (see fig. 23, E). In order to enable an advanced integration with users' mobile hardware and sensors (like microphone, camera, compass, gyroscope, etc.) a further step is indeed required to certificate (SSL) the server node or DN. This is also now mandatory for WebXR sessions (AR and VR), and possibly for accessing other device hardware in the near future.

A few instances of the framework are already deployed and publicly accessible (such as https://aton.ispc.cnr.it/), used by different institutions, researchers and professionals. The framework will be soon deployed also on different nodes of the E-RIHS European infrastructure [90] (http://www.e-rihs.it/). In particular, the features offered by the framework and the *plug-n-play* architecture of web-apps (see section 3.3) are already allowing a few projects (e.g. "H_2O" by free University of Bolzano [79]) to switch from game engines like Unity to ATON, to deploy immersive gaming experiences directly on the web, without any installation required.

The collaborative module (client and server components - see section 3.7) proved its effectiveness for the enrichment of the online 3D experience. It was already employed in several online courses and workshops with students connecting from remote locations during the covid-19 pandemic, and also for private meetings discussing the framework itself. Regarding the Nora case study (section 4.4), participants clearly appreciated the collaborative session and the tools offered by Hathor (see section 3.8) without any installation required on their devices. Besides collaborative annotation and dynamic lighting, specific tools developed for the CH field like *focus streaming* - to point or raise the attention on specific locations or areas - proved to be extremely effective for teaching in a multi-user 3D space. The virtual tour model also - as already proven by previous literature - highlighted the importance of a professional (virtual guide) describing a specific 3D reconstruction or hypothesis, as well as the "being there together" aspect,

24 http://osiris.itabc.cnr.it/scenebaker/index.php/projects/emviq/

especially valuable during the covid-19 pandemic. Compared to other open-source solutions, ATON is one of the few to offer *built-in* collaborative features specifically designed for the Cultural Heritage field, accessible from every device. Furthermore, the API allows custom web-applications to easily develop their own collaborative logic (see section 3.7.2), offering developers huge flexibility to craft powerful multi-user web-applications.

6. Conclusions

ATON is an open-source framework born in 2016: during the last few years it has grown into a rich, modular and flexible tool to craft powerful Web3D and WebXR applications for Cultural Heritage (3D presenters, inspection tools, applied games, collaborative teaching tools, etc.) that can be consumed on every device, from mobile up to head-mounted displays (HMDs). In this paper we describe the ATON framework architecture and its components, while presenting and assessing novel features, besides components already investigated in previous works. The framework was employed in several national and international projects that already exploited several components of the architecture, but more importantly, provided crucial feedback to evolve the framework itself.

The entire ATON framework is designed around modern and robust web standards, open specifications and large open-source libraries. The framework fully embraces the WebXR specification, which has become the standard to present 3D content on AR and VR devices through a web browser, with growing adoption by several online solutions. This enables web-applications crafted on top of ATON to be consumed on HMDs and AR devices, automatically adapting interaction models and interfaces. The adoption of Khronos glTF standard for 3D content delivery guarantees maximum flexibility, exchange, customization, durability, reuse and improved workflow due to its growing adoption. Within the framework special attention is given to advanced materials representation, thanks to the exploitation of a *physically-based rendering* (PBR) model that meets the 3D presentation requirements of the Heritage Science fields. The adoption of Cesium 3D Tiles OGC standard on the other hand, guarantees a robust streaming on all devices for massive multi-resolution datasets (photogrammetry, 3D buildings, BIM/CAD, instanced features, and point clouds) referenced in published 3D scenes. The framework provides built-in support for such standard, and thanks to international collaborations and community support, it already offers support for WebXR sessions that will be discussed in a separate paper.

ATON is designed to host and deploy multiple web-applications on the same instance leveraging on framework components (see section 3.3): the adopted *plug-n-play* architecture offers developers maximum flexibility and customization in terms of application logic and user interface. Furthermore, ATON web-apps adopt the PWA model (a set of standards developed by the Google Web Fundamentals group) aiming to offer responsive, app-like, discoverable and linkable web-based solutions. In the paper we specifically presented and discussed "*Hathor*" (the official ATON front-end) and what it offers "out of the box" in terms of features and tools for CH professionals, researchers and institutions.

The framework offers built-in components (server and client side) for *collaborative*, synchronous interactions among users. Special focus was given to the design of these components in order to provide developers with an extensible system to craft their own multi-user logic for web-apps. These features allow to elevate the online 3D experience from single to multi-user, opening incredible opportunities for the Cultural Heritage field in terms of teaching, virtual discussion spaces, collaborative tools or multi-user applied 3D gaming.

From a deployment perspective, the adoption of Node.js ecosystem guarantees high scalability for the framework on single-board computers, laptops, small servers and large infrastructures like E-RIHS [90]. Furthermore, several existing solutions like

Google Cloud, Amazon Web Services (AWS), Heroku (and much more) can be adopted, thus offering a wide range of options to CH institutions and professionals to disseminate Web3D/WebXR applications and 3D content on the web. The framework REST API provides a robust interface for integration with external services and platforms within the Heritage Sciences domain.

During the last few years ATON was also exploited as research playground to implement and investigate interaction techniques (like *Temporal Lensing* [67]), tools to inspect 4D virtual environments enriched with Graph-DBs [72] and other topics related to 3D visualization and interaction for CH, thus becoming a web-based "open lab".

As already highlighted in previous works [91], transforming a research tool into a product usable by the heterogeneous CH community is not an easy task, and a significant amount of resources is required. Community around the latest release of ATON is progressively building up and more developers and institutions are embracing the framework to create interactive 3D experiences, basic 3D presenters for museum collections, applied VR games and CH tools on the web. Current architecture design allows to distribute services in multiple deployment nodes: we plan to carry out extensive and in-depth assessments on services federation and their interplay with existing infrastructures and platforms. An in-depth assessment will be indeed carried out on the presentation features for massive 3D datasets offered by the framework, following (and possibly contributing to) the Cesium 3D Tiles specification. A few international collaborations are also allowing to develop new processing tools and services targeting this OGC standard. Regarding user interface (UI) and user experience (UX), it will be crucial to involve more content creators in order to improve or add editing tools, and on the other hand users/institutions to improve UI elements and their versatility to the different types of scenarios. We plan indeed an in-depth assessment specifically targeting *Hathor* front-end - and more in general ATON UI elements - that will be addressed in a separate paper.

Author Contributions: Conceptualization, B.F.; methodology, all; software, B.F.; validation, D.F., E.D., B.F. and S.B.; formal analysis, all; investigation, all; resources, D.F., E.D., S.B, E.A.; data curation, D.F., E.D., S.B and E.A.; writing—original draft preparation, B.F.; writing—review and editing, D.F. B.F., E.D. S.B.; supervision, B.F.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. Projects which have contributed to the progressive development of the framework, are mentioned in section 3 and 4 were funded by different cultural institutions over the time; related information can be found in references.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: All data and software involved are available through the official ATON framework website http://osiris.itabc.cnr.it/aton/, the framework github https://github.com/phoenixbf/aton or through the main instance https://aton.ispc.cnr.it/

Acknowledgments: Authors would like to acknowledge all the colleagues and the Institutions who made their assets available for our experiments or participated in the projects from which the case studies presented were taken: CNR - ISPC (IT); Université catholique de Louvain - UCLouvain (BE); Museo dei Fori Imperiali - Mercati di Traiano (IT), VRTron (MT); Soprintendenza Archeologia del Lazio e dell'Etruria Meridionale (IT); Necropoli della Banditaccia (IT); Museo Archeologico di Fara in Sabina (IT); University of Lund, Department of Archaeology and Ancient History (SE); prof. J. Bonetto and prof. G. Salemi of the University of Padua (IT). Authors would like to thank also the Three.js community for their support during development and implementation phases.

Conflicts of Interest: Authors declare no conflict of interest.

Abbreviations

1203 The following abbreviations are used in this manuscript:

1204

1205

HMD	Head-mounted display
3-Dof, 6-DoF	3 and 6 Degrees of Freedom for HMDs and VR controllers
PBR	Physically-based rendering
BVH	Bounding Volume Hierarchy
POV	Point of view (or viewpoint)
FPS	Frames per second
PWA	Progressive Web Application
CH	Cultural Heritage
DN	Deployment Node
API	Application Programming Interface
UE4	Unreal Engine 4 (game engine)
SBC	Single-board computer

References

1. Kharoub, H.; Lataifeh, M.; Ahmed, N. 3d user interface design and usability for immersive vr. *Applied Sciences* **2019**, *9*, 4861.
2. Evans, A.; Romeo, M.; Bahrehmand, A.; Agenjo, J.; Blat, J. 3D graphics on the web: A survey. *Computers & graphics* **2014**, *41*, 43–61.
3. Gasston, P. *The modern Web: multi-device Web development with HTML5, CSS3, and JavaScript*; No Starch Press, 2013.
4. Nurminen, J.K.; Meyn, A.J.; Jalonen, E.; Raivio, Y.; Marrero, R.G. P2P media streaming with HTML5 and WebRTC. 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2013, pp. 63–64.
5. Diamantaris, M.; Marcantoni, F.; Ioannidis, S.; Polakis, J. The seven deadly sins of the HTML5 WebAPI: a large-scale study on the risks of mobile sensor-based attacks. *ACM Transactions on Privacy and Security (TOPS)* **2020**, *23*, 1–31.
6. Neelakantam, S.; Pant, T. Bringing VR to the Web and WebVR Frameworks. In *Learning Web-based Virtual Reality*; Springer, 2017; pp. 5–9.
7. Jones, B.; Waliczek, N. WebXR device API. *W3C Working Draft* **2019**, *10*.
8. González-Zúñiga, L.D.; O'Shaughnessy, P. Virtual Reality… in the Browser. *VR Developer Gems; CRC Press: Boca Raton, FL, USA* **2019**, p. 101.
9. Maclntyre, B.; Smith, T.F. Thoughts on the Future of WebXR and the Immersive Web. 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). IEEE, 2018, pp. 338–342.
10. Biggio, F. Protocols of immersive web: WebXR APIs and the AR Cloud. Proceedings of the 3rd International Conference on Web Studies, 2020, pp. 27–32.
11. Echavarria, K.R.; Dibble, L.; Bracco, A.; Silverton, E.; Dixon, S. Augmented Reality (AR) Maps for Experiencing Creative Narratives of Cultural Heritage. EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2019), 2019.
12. Jung, K.; Nguyen, V.T.; Lee, J. BlocklyXR: An Interactive Extended Reality Toolkit for Digital Storytelling. *Applied Sciences* **2021**, *11*, 1073.
13. Robinet, F.; Arnaud, R.; Parisi, T.; Cozzi, P. gltf: Designing an open-standard runtime asset format. *GPU Pro* **2014**, *5*, 375–392.
14. Happa, J.; Artusi, A. Studying Illumination and Cultural Heritage., 2020.
15. Vanweddingen, V.; Vastenhoud, C.; Proesmans, M.; Hameeuw, H.; Vandermeulen, B.; Van der Perre, A.; Lemmers, F.; Watteeuw, L.; Van Gool, L. A status quaestionis and future solutions for using multi-light reflectance imaging approaches for preserving cultural heritage artifacts. Euro-Mediterranean Conference. Springer, 2018, pp. 204–211.
16. Schilling, A.; Bolling, J.; Nagel, C. Using glTF for streaming CityGML 3D city models. Proceedings of the 21st International Conference on Web3D Technology, 2016, pp. 109–116.
17. Xu, Z.; Zhang, L.; Li, H.; Lin, Y.H.; Yin, S. Combining IFC and 3D tiles to create 3D visualization for building information modeling. *Automation in Construction* **2020**, *109*, 102995.
18. Mao, B.; Ban, Y.; Laumert, B. Dynamic Online 3D Visualization Framework for Real-Time Energy Simulation Based on 3D Tiles. *ISPRS International Journal of Geo-Information* **2020**, *9*, 166.
19. Danchilla, B. Three. js framework. In *Beginning WebGL for HTML5*; Springer, 2012; pp. 173–203.
20. Münster, S.; Maiwald, F.; Lehmann, C.; Lazariv, T.; Hofmann, M.; Niebling, F. An Automated Pipeline for a Browser-based, City-scale Mobile 4D VR Application based on Historical Images. Proceedings of the 2nd Workshop on Structuring and Understanding of Multimedia heritAge Contents, 2020, pp. 33–40.
21. Gill, A. AFrame: a domain specific language for virtual reality. Proceedings of the 2nd International Workshop on Real World Domain Specific Languages, 2017, pp. 1–1.
22. Potenziani, M.; Callieri, M.; Dellepiane, M.; Corsini, M.; Ponchio, F.; Scopigno, R. 3DHOP: 3D heritage online presenter. *Computers & Graphics* **2015**, *52*, 129–141.
23. Lloyd, J. Contextualizing 3D cultural heritage. Euro-Mediterranean Conference. Springer, 2016, pp. 859–868.
24. Romphf, J.; Neuman-Donihue, E.; Heyworth, G.; Zhu, Y. Resurrect3D: An Open and Customizable Platform for Visualizing and Analyzing Cultural Heritage Artifacts. *arXiv preprint arXiv:2106.09509* **2021**.

25. Biørn-Hansen, A.; Majchrzak, T.A.; Grønli, T.M. Progressive web apps: The possible web-native unifier for mobile development. International Conference on Web Information Systems and Technologies. SciTePress, 2017, Vol. 2, pp. 344–351.
26. Adetunji, O.; Ajaegbu, C.; Otuneme, N.; Omotosho, O.J.; others. Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)* **2020**, *68*, 85–99.
27. Shah, H. Node. js challenges in implementation. *Global Journal of Computer Science and Technology* **2017**.
28. Machidon, O.M.; Tavčar, A.; Gams, M.; Duguleană, M. CulturalERICA: A conversational agent improving the exploration of European cultural heritage. *Journal of Cultural Heritage* **2020**, *41*, 152–165.
29. Huang, C.M.; Guo, Y.A. A Touring and Navigation Service Platform for Mobile Digital Culture Heritage (M-DCH). 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN). IEEE, 2018, pp. 185–192.
30. Bran, E.; Bautu, E.; Popovici, D.M.; Braga, V.; Cojuhari, I. Cultural Heritage Interactive Dissemination through Natural Interaction. RoCHI, 2019, pp. 156–161.
31. Doglio, F.; Doglio.; Corrigan. *REST API Development with Node. js*; Springer, 2018.
32. Aderaldo, C.M.; Mendonça, N.C.; Pahl, C.; Jamshidi, P. Benchmark requirements for microservices architecture research. 2017 IEEE/ACM 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE). IEEE, 2017, pp. 8–13.
33. Liarokapis, F.; Anderson, E.F. Collaborating and Learning in Shared Virtual Environments. *IEEE Computer Graphics and Applications* **2020**, *40*, 8–9.
34. Vincenti, G.; Braman, J. *Teaching through multi-user virtual environments: Applying dynamic elements to the modern classroom: Applying dynamic elements to the modern classroom*; IGI Global, 2010.
35. Barchetti, U.; Bucciero, A.; Santo Sabato, S.; Mainetti, L. A Framework to Generate 3D Learning Experience. In *New Achievements in Technology, Education and Development*; IntechOpen, 2010.
36. Mariani, R. A new virtual. A new reality. How pedagogical approaches are changing. **2020**.
37. Li, J.; Vinayagamoorthy, V.; Williamson, J.; Shamma, D.A.; Cesar, P. Social VR: A New medium for remote communication and collaboration. Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–6.
38. Jonas, M.; Said, S.; Yu, D.; Aiello, C.; Furlo, N.; Zytko, D. Towards a taxonomy of social vr application design. Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts, 2019, pp. 437–444.
39. Kong, Y. User experience in social virtual reality: Exploring methodologies for evaluating user experience in social virtual reality **2018**.
40. Reinhardt, J.; Wolf, I.K. Opportunities of Social VR in Digital Museum Twins. *EVA BERLIN 2018* **2018**, p. 320.
41. Latoschik, M.E.; Roth, D.; Gall, D.; Achenbach, J.; Waltemate, T.; Botsch, M. The effect of avatar realism in immersive social virtual realities. Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology, 2017, pp. 1–10.
42. Bönsch, A.; Radke, S.; Overath, H.; Asché, L.M.; Wendt, J.; Vierjahn, T.; Habel, U.; Kuhlen, T.W. Social VR: How personal space is affected by virtual agents' emotions. 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE, 2018, pp. 199–206.
43. Hudák, M.; Korečko, Š.; Sobota, B. Advanced user interaction for web-based collaborative virtual reality. 2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom). IEEE, 2020, pp. 000343–000348.
44. Bredikhina, L.; Sakaguchi, T.; Shirai, A. Web3D Distance live workshop for children in Mozilla Hubs. The 25th International Conference on 3D Web Technology, 2020, pp. 1–2.
45. Suslov, N. LiveCoding. space: Towards P2P Collaborative Live Programming Environment for WebXR. Proceedings of the Fourth International Conference on Live Coding (ICLC 2019), Medialab Prado, Madrid, Spain, http://iclc. livecodenetwork. org/2019/papers/paper133. pdf, 2019.
46. Fette, I.; Melnikov, A. The websocket protocol, 2011.
47. Chen, B.; Xu, Z. A framework for browser-based Multiplayer Online Games using WebGL and WebSocket. 2011 International Conference on Multimedia Technology. IEEE, 2011, pp. 471–474.
48. Gunkel, S.; Prins, M.; Stokking, H.; Niamut, O. *WebVR meets WebRTC: Towards 360-degree social VR experiences*; IEEE, 2017.
49. Meghini, C.; Scopigno, R.; Richards, J.; Wright, H.; Geser, G.; Cuy, S.; Fihn, J.; Fanini, B.; Hollander, H.; Niccolucci, F.; others. ARIADNE: A research infrastructure for archaeology. *Journal on Computing and Cultural Heritage (JOCCH)* **2017**, *10*, 1–27.
50. Fanini, B.; Pescarin, S.; Palombini, A. A cloud-based architecture for processing and dissemination of 3D landscapes online. *Digital Applications in Archaeology and Cultural Heritage* **2019**, p. e00100.
51. Gonzalez, D. *Developing microservices with node. js*; Packt Publishing, 2016.
52. Bryan, P.; Nottingham, M. Javascript object notation (json) patch. *RFC 6902 (Proposed Standard)* **2013**.
53. Frain, B. *Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques*; Packt Publishing Ltd, 2020.
54. Ferdani, D.; Fanini, B.; Piccioli, M.C.; Carboni, F.; Vigliarolo, P. 3D reconstruction and validation of historical background for immersive VR applications and games: The case study of the Forum of Augustus in Rome. *Journal of Cultural Heritage* **2020**, *43*, 129–143.
55. Pescarin, S.; Fanini, B.; Ferdani, D.; Mifsud, K.; Hamilton, A. Optimising Environmental Educational Narrative Videogames: The Case of 'A Night in the Forum'. *Journal on Computing and Cultural Heritage (JOCCH)* **2020**, *13*, 1–23.

56. Russell, J. HDR Image-Based Lighting on the Web. *WebGL Insights* **2015**, p. 253.
57. Patney, A.; Salvi, M.; Kim, J.; Kaplanyan, A.; Wyman, C.; Benty, N.; Luebke, D.; Lefohn, A. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)* **2016**, *35*, 1–12.
58. Dangkham, P. Mobile augmented reality on web-based for the tourism using HTML5. 2018 International Conference on Information Networking (ICOIN). IEEE, 2018, pp. 482–485.
59. Barsanti, S.G.; Malatesta, S.G.; Lella, F.; Fanini, B.; Sala, F.; Dodero, E.; Petacco, L. The WINCKELMANN300 PROJECT: Dissemination of Culture with Virtual Reality at the Capitoline Museum in Rome. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2018**, *XLII-2*, 371–378. doi:10.5194/isprs-archives-XLII-2-371-2018.
60. Carrozzino, M.; Bruno, N.; Bergamasco, M. Designing interaction metaphors for Web3D cultural dissemination. *Journal of Cultural Heritage* **2013**, *14*, 146–155.
61. Bozgeyikli, E.; Raij, A.; Katkoori, S.; Dubey, R. Point & teleport locomotion technique for virtual reality. Proceedings of the 2016 annual symposium on computer-human interaction in play, 2016, pp. 205–216.
62. Boletsis, C. The new era of virtual reality locomotion: A systematic literature review of techniques and a proposed typology. *Multimodal Technologies and Interaction* **2017**, *1*, 24.
63. Habgood, M.J.; Moore, D.; Wilson, D.; Alapont, S. Rapid, continuous movement between nodes as an accessible virtual reality locomotion technique. 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE, 2018, pp. 371–378.
64. Cournia, N.; Smith, J.D.; Duchowski, A.T. Gaze-vs. hand-based pointing in virtual environments. CHI'03 extended abstracts on Human factors in computing systems, 2003, pp. 772–773.
65. Minakata, K.; Hansen, J.P.; MacKenzie, I.S.; Bækgaard, P.; Rajanna, V. Pointing by gaze, head, and foot in a head-mounted display. Proceedings of the 11th ACM symposium on eye tracking research & applications, 2019, pp. 1–9.
66. Kontakis, K.; Malamos, A.G.; Steiakaki, M.; Panagiotakis, S. Spatial indexing of complex virtual reality scenes in the web. *International Journal of Image and Graphics* **2017**, *17*, 1750009.
67. Fanini, B.; Ferdani, D.; Demetrescu, E. Temporal Lensing: An Interactive and Scalable Technique for Web3D/WebXR Applications in Cultural Heritage. *Heritage* **2021**, *4*, 710–724.
68. Flotyński, J.; Malamos, A.G.; Brutzman, D.; Hamza-Lup, F.G.; Polys, N.F.; Sikos, L.F.; Walczak, K. Recent advances in Web3D semantic modeling. *Recent advances in 3D imaging, modeling, and reconstruction* **2020**, pp. 23–49.
69. Ponchio, F.; Callieri, M.; Dellepiane, M.; Scopigno, R. Effective annotations over 3D models. Computer Graphics Forum. Wiley Online Library, 2020, Vol. 39, pp. 89–105.
70. Tobler, R.F. Separating semantics from rendering: a scene graph based architecture for graphics applications. *The Visual Computer* **2011**, *27*, 687–695.
71. Serna, S.P.; Schmedt, H.; Ritz, M.; Stork, A. Interactive Semantic Enrichment of 3D Cultural Heritage Collections. VAST, 2012, pp. 33–40.
72. Demetrescu, E.; Fanini, B. A white-box framework to oversee archaeological virtual reconstructions in space and time: methods and tools. *Journal of Archaeological Science: Reports* **2017**, *14*, 500–514.
73. Alger, M. Visual design methods for virtual reality. *Ravensbourne. http://aperturesciencellc. com/vr/VisualDesignMethodsforVR_MikeAlger. pdf* **2015**.
74. Johnson-Glenberg, M.C. Immersive VR and education: Embodied design principles that include gesture and hand controls. *Frontiers in Robotics and AI* **2018**, *5*, 81.
75. Hammady, R.; Ma, M. Designing spatial ui as a solution of the narrow fov of microsoft hololens: Prototype of virtual museum guide. In *Augmented reality and virtual reality*; Springer, 2019; pp. 217–231.
76. Rai, R. *Socket. IO Real-time Web Application Development*; Packt Publishing Ltd, 2013.
77. Palombini, A.; Fanini, B.; Pagano, A. The Virtual Museum of the Upper Calore Valley. International and Interdisciplinary Conference on Digital Environments for Education, Arts and Heritage. Springer, 2018, pp. 726–736.
78. Turco, M.L.; Piumatti, P.; Calvano, M.; Giovannini, E.C.; Mafrici, N.; Tomalini, A.; Fanini, B. Interactive Digital Environments for Cultural Heritage and Museums. Building a digital ecosystem to display hidden collections. *Disegnarecon* **2019**, *12*, 7–1.
79. Luigini, A.; Fanini, B.; Basso, A.; Basso, D. Heritage education through serious games. A web-based proposal for primary schools to cope with distance learning. *VITRUVIO-International Journal of Architectural Technology and Sustainability* **2020**, *5*, 73–85.
80. Fanini, B.; Cinque, L. Encoding immersive sessions for online, interactive VR analytics. *Virtual Reality* **2020**, *24*, 423–438.
81. Fanini, B.; Cinque, L. Encoding, Exchange and Manipulation of Captured Immersive VR Sessions for Learning Environments: the PRISMIN Framework. *Applied Sciences* **2020**, *10*, 2026.
82. Cavalieri, M. La villa romana di Aiano-Torraccia di Chiusi, III campagna di scavi 2007. Il progetto internazionale "VII Regio. Il caso della Val d'Elsa in età romana e tardoantica". *FOLD&R FastiOnLine documents & research* **2008**, pp. 1–23.
83. Ferdani, D.; Demetrescu, E.; Cavalieri, M.; Pace, G.; Lenzi, S. 3D Modelling and Visualization in Field Archaeology. From Survey To Interpretation Of The Past Using Digital Technologies. *Groma. Documenting archaeology* **2020**.
84. Demetrescu, E. Virtual reconstruction as a scientific tool. In *Digital research and education in architectural heritage*; Springer, 2017; pp. 102–116.
85. Bonetto, J.; Falezza, G.; Ghiotto, A.; Novello, M. *Nora. Il foro romano. Storia di un'area urbana dall'età fenicia alla tarda antichità (1997-2006), I-IV*, italgraf ed.; Scavi di Nora, 1.

86. Pai, H.Y. Texture designs and workflows for physically based rendering using procedural texture generation. 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2019, pp. 195–198. doi:10.1109/ECICE47484.2019.8942651.
87. Baruah, R. Creating an Augmented Reality Website with Three. js and the WebXR API. In *AR and VR Using the WebXR API*; Springer, 2021; pp. 217–252.
88. Aittala, M. Inverse lighting and photorealistic rendering for augmented reality. *The Visual Computer* **2010**, *26*, 669–678.
89. Waldner, F. Real-time Ray Traced Ambient Occlusion and Animation: Image quality and performance of hardware-accelerated ray traced ambient occlusion, 2021.
90. Striova, J.; Pezzati, L. THE EUROPEAN RESEARCH INFRASTRUCTURE FOR HERITAGE SCIENCE (ERIHS). *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* **2017**, *42*.
91. Potenziani, M.; Callieri, M.; Scopigno, R. Developing and Maintaining a Web 3D Viewer for the CH Community: an Evaluation of the 3DHOP Framework. GCH, 2018, pp. 169–178.