Int. j. adv. multidisc. res. stud. 2026; 6(3):964-968

Received: 08-04-2026
Accepted: 18-05-2026

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

The Development of Architectural Approaches for Creating Interactive Scenarios for VR and AR Use

[1] Alevizopoulos A, [2] Savvatianopoulos M, [3] Dandolou D, [4] Kritikos I, [5] Alexandrides T, [6] Alevizopoulos G

[1, 3, 4, 6] 3rd University Psychiatric Clinic, GONK, Aghioi Anargyroi, NKUA, Greece

[2] Premium Consulting, Greece

[5] Omega Technology, Greece

Corresponding Author: **Alevizopoulos G**

Abstract

Background: As immersive virtual reality (VR) and augmented reality (AR) environments evolve, their architectural frameworks must adapt to support dynamic interactivity and user agency.

System Overview: This manuscript introduces the VIODET platform, which advances the design of interactive scenarios by offering two complementary architectural approaches unified under a single cross-platform Unity client. The platform deploys seamlessly across the Meta Quest 3, mobile devices in Mixed Reality environments, and Windows PCs.

Architectural Approaches:

- **AI-Powered Dynamic Interaction:** The first approach leverages cloud-based artificial intelligence, including Speech-to-Text, Large Language Models, and Text-to-Speech, to facilitate natural, free-form dialogues and adaptive narrative progression. It utilizes a dual-stream animation system that blends cloud-driven lip synchronization with locally managed body movements, effectively mitigating network latency artifacts.
- **Non-AI Deterministic Authoring:** The second approach relies on a backend graphical scenario design tool to create predetermined multiple-choice dialogue graphs and event mappings. This deterministic framework guarantees complete outcome predictability and testability, making it highly suitable for compliance-driven applications, medical simulations, and high-stakes training.

Conclusion: By decoupling content creation from the runtime environment via a modular client-server model, VIODET successfully reconciles the demand for emergent, exploratory user experiences with the necessity for authorial control and production efficiency. This dual-architecture strategy serves as a resilient and accessible framework for future extended reality (XR) development across educational, entertainment, and industrial domains.

Keywords: Virtual Reality (VR), Artificial Intelligence (AI), Interactive Scenarios, Narrative Design, Cross-Platform Architecture

Introduction

The evolution of architectural approaches for creating interactive scenarios in virtual reality (VR) and augmented reality (AR) reflects a multidisciplinary convergence of design, technology, and user experience. As immersive environments transition from experimental prototypes to practical applications across entertainment, education, and industry, the architectural frameworks underpinning these spaces have become increasingly sophisticated. Early VR and AR architectures primarily focused on replicating physical spaces digitally; however, contemporary developments emphasize dynamic interactivity and seamless integration with real-world contexts. This shift necessitates a reevaluation of traditional architectural principles to accommodate spatial fluidity, sensory engagement, and user agency within digital realms [1-3]. Central to this progression is the adoption of user-centered design methodologies that prioritize intuitive navigation and meaningful interaction within virtual environments [4]. Technological innovations such as large language models facilitating JSON-driven scene generation exemplify how automation and context-aware systems reduce complexity while enhancing creative control [5]. These advancements enable designers to craft intricate XR worlds without extensive coding expertise, thus democratizing content creation [6]. Furthermore, analyzing successful VR and AR projects reveals patterns in effective spatial organization, narrative

integration, and technological interoperability that inform best practices in virtual architecture. Looking forward, emerging trends suggest an increasing emphasis on adaptive environments capable of responding dynamically to user behavior and environmental data streams. The interplay between evolving hardware capabilities—such as improved sensors and display technologies—and innovative software architectures promises to redefine interactive spatial experiences fundamentally. Understanding this trajectory provides critical insight into how architectural approaches can continue to evolve in response to both technological potentialities and human-centered imperatives within immersive media landscapes.

The VIODET Platform

The VIODET platform supports two distinct architectural approaches for creating interactive scenarios, which share a common foundation on the client side (Unity application) but differ fundamentally in how content is produced and managed.

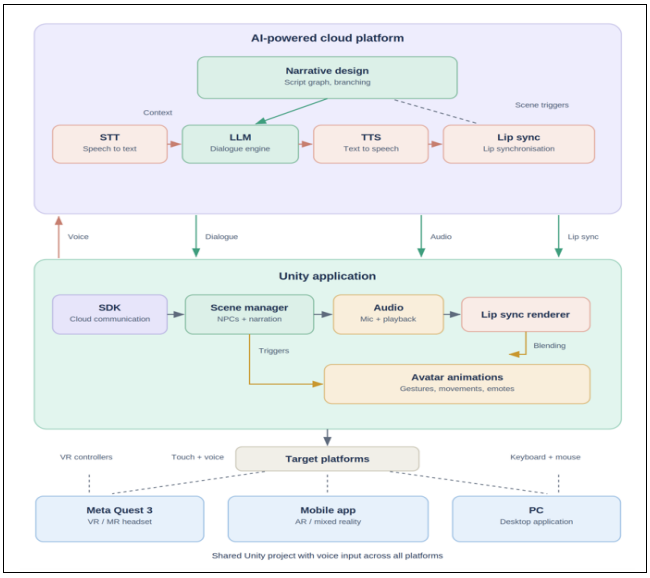

Diagram 1: AI-powered platform

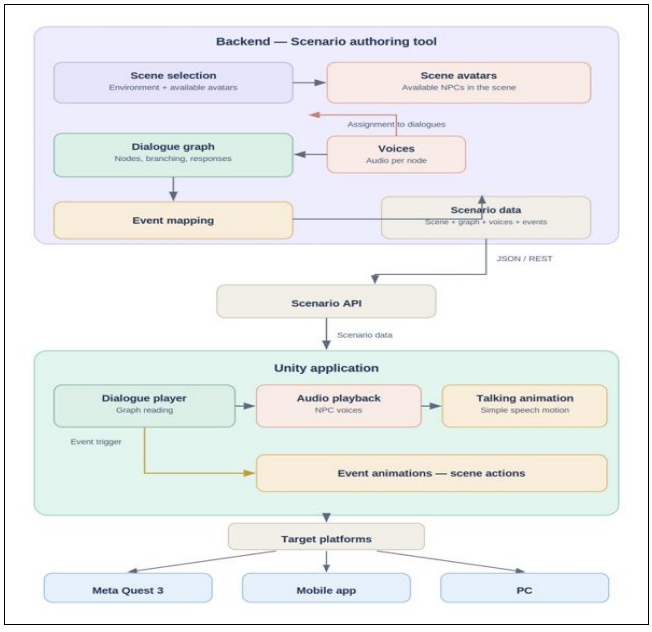

Diagram 2: Scenario authoring tool

The first approach utilizes cloud-based artificial intelligence services for dynamic, real-time dialogue generation. User interaction with the characters is free and non-predetermined: the user speaks naturally, their speech is transcribed to text, a large language model (LLM) generates the character's response based on the narrative context, and the response is returned as synthetic speech with lip-sync. The progression of the scenario is driven by a Narrative Design, which defines the scene structure and transition conditions (Diagram 1).

The second approach is based on pre-designed multiple-choice scenarios without the use of artificial intelligence. A backend tool allows the creation of scenarios through a dialogue graph, where every phrase, user (player) responses, and scene animations are defined in advance by the author. The data is transferred to the Unity application via a REST API, and playback faithfully follows the predetermined graph (Diagram 2).

In both cases, the Unity application is developed from a single project and runs on Meta Quest 3, mobile devices, and Windows PCs. The application is based on a client-server architecture combining cloud AI services with a Unity runtime environment, deployed on Meta Quest 3, mobile devices, and Windows PCs.

Cloud Platform with Artificial Intelligence

The cloud tier provides four core services:

- Narrative Design: Defines the scenario structure as a scene graph with branches, character backgrounds, and trigger conditions that control how the interactive experience unfolds.
- Speech-to-Text (STT): Transcribes the user's speech into text.
- Large Language Model (LLM): Generates appropriate character dialogue based on the narrative state. The LLM also returns scene triggers to the Narrative Design, enabling dynamic scenario progression based on user interactions.
- Text-to-Speech (TTS): Synthesizes the response into vocal audio per character.
- Lip Sync: Produces real-time lip synchronization data, synced with the generated speech.

Unity Application

The client-side Unity application includes an SDK that manages communication with the cloud platform. The Scene Manager coordinates NPC behavior and narrative flow, receiving dialogue responses from the SDK and routing them to the appropriate subsystems. The audio system manages microphone capture for user input and playback of the NPCs' synthetic speech.

Avatar animation is driven by two independent but complementary sources:

- The Lip Sync Renderer maps incoming data from the cloud platform to facial blend shapes, producing accurate lip movements synchronized with speech.
- Concurrently, a custom avatar animation system, triggered locally by the Scene Manager, controls body gestures, idle movements, emotes, and scripted actions via the Unity Animator.

These two streams are blended for each character, allowing the facial animation to remain closely tied to the cloud-generated speech, while the body animation remains responsive and independent of network latency.

Deployment Platforms

The application is developed for three platforms: as a standalone VR experience on the Meta Quest 3, as a mobile app offering the same functionality in Mixed Reality environments, and as a PC application for desktop users, all from a single Unity project. User input returns to the system via voice capture and per-device interaction methods: VR controllers on the Quest 3, touch and voice on mobile, and keyboard and mouse on PC.

Non-AI Architecture

The application relies on a client-server architecture combining a scenario design backend (Web application) and a Unity runtime environment, for Meta Quest 3, mobile devices, and Windows PCs.

Backend: Scenario Design Tool

The backend provides a graphical environment for creating interactive scenarios.

- Workflow: The workflow begins with scene selection: the author selects the environment in which the scenario unfolds, and each scene defines which avatars are present in it. The scene avatars then become available as NPCs that the author can assign to dialogue nodes.
- Dialogue Graph: Each scenario is represented as a dialogue graph, where nodes correspond to NPC phrases and branches correspond to the user's alternative responses. For each dialogue node, the author assigns one of the scene avatars as the speaker, while a voice audio file can be added to any node.
- Events: Additionally, any node or response can be linked to a scene event, so that triggering a specific point in the dialogue fires a corresponding action within the application.

All data — scene definition, graph structure, audio files, and event mappings — are stored in the backend and sent to the application via API.

Scenario API

A REST API transfers each scenario's data from the backend to the Unity application, including the dialogue graph structure, audio files, and event mappings.

Unity Application

The Unity application receives the scenario data from the API and plays back the scenario in real-time. The Dialogue Player navigates through the graph's nodes, playing the corresponding voices through the audio system, while characters perform simple talking animations during each phrase. For every event defined in the backend, there is a corresponding animation in Unity that automatically triggers when the user reaches the relevant dialogue node or selects the linked response.

Deployment Platforms

The application is developed from a single Unity project for three platforms: Meta Quest 3 as a VR experience, a mobile app in Mixed Reality environments, and a PC application for desktop users.

Discussion

The VIODET platform represents a significant advancement in the architectural design of interactive VR and AR scenarios by offering two complementary yet fundamentally distinct approaches to content creation and delivery, both unified under a single, cross-platform Unity client. This dual-architecture strategy addresses a core tension in immersive media development: the desire for rich, emergent user experiences on one hand, and the need for authorial control, predictability, and production efficiency on the other. By grounding both approaches in the same robust client-server foundation, VIODET achieves remarkable flexibility while minimizing development overhead and deployment complexity.

Architectural Foundations and Shared Strengths

At its core, VIODET leverages a modular client-server model that decouples content authoring and management from the runtime experience. The Unity application—deployed as a standalone VR experience on Meta Quest 3, a Mixed Reality mobile app, and a desktop PC application from a single project—serves as the universal runtime layer. This "write once, deploy anywhere" philosophy is particularly noteworthy in an industry where fragmentation across hardware ecosystems often inflates costs and delays iteration. The application's SDK handles cloud or backend communication, while specialized subsystems (Scene Manager, audio pipeline, and dual animation streams) ensure responsive, believable character interactions regardless of the chosen content pipeline.

A standout technical innovation is the independent yet blended animation architecture [7, 8]. Lip synchronization is driven directly from cloud-generated TTS data (in the AI approach) or pre-recorded audio (in the non-AI approach), while body gestures, idle movements, emotes, and scripted actions are controlled locally by the Scene Manager via Unity's Animator. This separation eliminates network-latency-induced artifacts in facial animation and allows body movements to remain snappy and contextually appropriate, resulting in characters that feel alive and responsive even under variable network conditions.

AI-Powered Dynamic Interaction

The cloud-based AI architecture (Diagram 1) introduces a highly adaptive, conversationally fluid experience. Narrative Design acts as the high-level orchestrator, defining scene graphs, character backstories, and trigger conditions. Real-time user speech is captured, transcribed via STT, processed by an LLM that generates context-aware dialogue and potential scene triggers, converted to speech with TTS, and synchronized with lip data [9]. This closed loop enables genuinely open-ended interactions: users can speak naturally, deviate from expected paths, and still receive coherent, character-consistent responses that can dynamically alter the scenario's progression.

Strengths of this approach include:

- Natural user agency: Free-form dialogue dramatically increases immersion and emotional engagement compared to menu-driven systems [10].
- Scalability of content: Once a Narrative Design is established, the LLM can generate virtually unlimited branches without manual authoring of every possibility [11].
- Rapid prototyping: Designers can iterate on high-level narrative structures rather than exhaustive dialogue trees.

Potential limitations, which future iterations of VIODET could mitigate through hybrid safeguards or guardrail

prompts in the LLM, include dependency on cloud latency, variable quality of generated responses, and the computational cost of real-time STT/TTS/LLM inference [10]. The platform's design already anticipates some of these issues by allowing the LLM to return explicit scene triggers back to the Narrative Design, providing a structured safety net around otherwise open-ended conversations.

Non-AI Deterministic Authoring

The second architecture (Diagram 2) prioritizes authorial precision through a web-based graphical scenario design tool. Authors define environments, assign avatars, construct dialogue graphs (nodes for NPC lines, branches for player choices), attach voice audio files, and map events to Unity-side animations or state changes. All data travels via a clean REST API to the Unity Dialogue Player, which faithfully executes the predetermined graph.

This approach excels in domains requiring strict pedagogical, therapeutic, or compliance-driven outcomes—such as corporate training, medical simulations, language learning, or guided tours—where every possible user path must be vetted and every outcome predetermined. Advantages include:

- Complete predictability and testability: Every interaction path can be exhaustively reviewed and quality-assured before deployment.
- Lower operational cost: No recurring LLM or cloud inference fees.
- Fine-grained multimedia control: Authors can embed custom audio, precise timing, and complex event sequences that might be difficult to achieve reliably through generative AI alone.

The trade-off is reduced spontaneity and higher upfront authoring effort. However, the visual dialogue-graph editor significantly lowers the technical barrier compared to traditional scripting or code-based branching logic.

Comparative Insights and Use-Case Alignment

Both architectures share the same Unity client, deployment targets, and animation blending system, enabling seamless switching or even hybrid deployments within the same project. This convergence is one of VIODET's most elegant architectural achievements: it allows studios or educators to begin with the deterministic tool for rapid, controlled content creation and later evolve selected scenarios toward AI-driven dynamism without rebuilding the client application.

The AI approach aligns naturally with entertainment, exploratory education, and social-simulation use cases where emergent storytelling enhances engagement. The non-AI approach is ideally suited for high-stakes training, accessibility-focused experiences, or scenarios demanding regulatory compliance and reproducibility. Together, they illustrate a mature understanding that no single paradigm dominates immersive content creation; instead, the most effective systems provide authors with a spectrum of control.

Broader Implications for VR/AR Architectural Practice

VIODET exemplifies several emerging best practices in virtual architecture:

1. User-centered spatial and interaction design: Both pipelines prioritize intuitive voice-first interaction while supporting device-native inputs (controllers, touch, keyboard/mouse), ensuring accessibility across hardware.
2. Democratization of immersive content: By abstracting complex logic into Narrative Design graphs or visual dialogue editors, and by leveraging LLMs for dialogue generation, VIODET reduces the coding expertise required, allowing subject-matter experts (educators, therapists, storytellers) to become primary content creators.
3. Resilient cross-platform architecture: The single-project Unity strategy, combined with cloud/backend decoupling, future-proofs the platform against hardware evolution and supports rapid expansion to new XR devices.
4. Hybrid animation intelligence: The separation of cloud-driven facial performance from locally-driven body animation offers a practical model for other platforms seeking to balance real-time generative capabilities with deterministic performance.

Future Directions

The platform is well-positioned for several evolutionary paths. First, tighter integration between the two architectures—such as using the deterministic graph as a scaffold that the LLM can intelligently expand or contract—could yield the best of both worlds. Second, incorporation of multimodal inputs (gaze, gesture, object interaction) and environmental data streams would enable truly adaptive, context-aware environments as envisioned in the introduction. Third, performance profiling across the three deployment targets will likely reveal opportunities for edge-side inference or selective offline caching of common dialogue paths. Finally, expanding the backend authoring tool with AI-assisted node generation (e.g., LLM suggestions for branching dialogue) could further accelerate content creation while preserving authorial oversight.

In conclusion, the VIODET platform successfully bridges traditional architectural principles of spatial and narrative coherence with modern AI-driven interactivity. Its dual-approach design not only solves immediate production challenges but also charts a thoughtful trajectory for the next generation of immersive experiences—one that remains firmly centered on human creativity while intelligently leveraging artificial intelligence where it adds the greatest value. By maintaining a unified, high-performance client across VR, MR, and desktop, VIODET demonstrates that sophisticated interactive architectures need not sacrifice accessibility, reliability, or creative control. This balanced philosophy positions the platform as a compelling model for future XR development in education, entertainment, industry training, and beyond.

References

1. Slater M, Wilbur S. A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. Presence: Teleoperators and Virtual Environments. 1997; 6(6):603-616.
2. Witmer BG, Singer MJ. Measuring presence in virtual environments: A presence questionnaire. Presence: Teleoperators and Virtual Environments. 1998; 7(3):225-240.
3. Slater M. Immersion and the illusion of presence in virtual reality. British Journal of Psychology. 2018; 109(3):431-433.

4. Jerald J. The VR Book: Human-Centered Design for Virtual Reality. New York, NY, USA: Association for Computing Machinery and Morgan & Claypool, 2015.
5. Chen J, Wu X, Lan T, Li B. LLMER: Crafting interactive extended reality worlds with JSON data generated by large language models. IEEE Transactions on Visualization and Computer Graphics, 2025. Doi: 10.1109/TVCG.2025.3549549
6. Roberts J, Banburski-Fahey A, Lanier J. Steps towards prompt-based creation of virtual worlds. arXiv preprint arXiv:2211.05875, 2022.
7. Pham HX, Wang Y, Pavlovic V. End-to-end learning for 3D facial animation from raw waveforms of speech. arXiv preprint arXiv:1710.00920, 2017.
8. Li X, Wang X, Wang K, Lian S. A novel speech-driven lip-sync model with CNN and LSTM. arXiv preprint arXiv:2205.00916, 2022.
9. Özkaya S, Berrezueta-Guzman S, Wagner S. How LLMs are shaping the future of virtual reality. arXiv preprint arXiv:2508.00737, 2025.
10. Christiansen FR, Hollensberg LN, Jensen NB, Julsgaard K, Jespersen KN, Nikolov I. Exploring presence in interactions with LLM-driven NPCs: A comparative study of speech recognition and dialogue options. In Proc. 30th ACM Symp. Virtual Reality Software and Technology (VRST' 24), 2024, 1-11. Doi: 10.1145/3641825.3687716
11. Van Stegeren J, Myśliwiec J. Fine-tuning GPT-2 on annotated RPG quests for NPC dialogue generation. In Proc. 16th Int. Conf. Foundations of Digital Games (FDG' 21), 2021, Art. no. 2. Doi: 10.1145/3472538.3472595