MundoVR: A Hybrid AI-VR Architecture for Real-Time Gamified Second Language Acquisition

Ashley Naka (819029)
Jennifer Awounou (819021)
Azamkhon Khudoyberdiev (819025)

Abstract

This paper presents MundoVR, a hybrid AI-VR architecture designed to address the latency, scalability, and cost challenges of integrating Large Language Models into immersive educational environments. By strategically distributing workloads between lightweight on-device processing and cloud-based services, the proposed architecture targets a p99 latency under 1.5 s while supporting 10,000 concurrent users at estimated costs below $0.10 per session. We provide formal SysML specifications and Architecture Decision Records, offering a reusable blueprint for scalable, pedagogically adaptive AI-VR systems that align with Second Language Acquisition principles.

Keywords

Virtual Reality, Large Language Models, Software Architecture, Second Language Acquisition, Microservices, Edge Computing, SysML

1 Introduction

1.1 Motivation and Context

Second Language Acquisition (SLA) research consistently demonstrates that authentic conversational practice is crucial for developing fluency [9]. However, traditional Computer-Assisted Language Learning (CALL) systems fail to provide the spontaneous, contextually rich interactions necessary for developing conversational competence. Meanwhile, recent breakthroughs in Large Language Models (LLMs) and automatic speech recognition have created unprecedented opportunities for intelligent, adaptive dialogue systems. Virtual Reality technology offers unique affordances for language learning: presence, embodiment, and contextual learning. Studies show VR environments improve vocabulary retention by 35-50% compared to traditional methods [15], while reducing learner anxiety through safe, judgment-free practice spaces [2]. The convergence of these technologies creates the possibility of realistic conversational practice systems. However, integrating them poses significant software engineering challenges: real-time AI inference must maintain VR immersion, handle complex natural language understanding within 1.5 seconds, scale to thousands of concurrent users, and remain cost-effective.

1.2 Problem Statement

Current AI-VR language learning systems face three critical limitations. First, cloud-based LLMs introduce 2-5 second delays, breaking VR immersion, while on-device models lack the contextual reasoning for pedagogically sound conversations. Second, GPU-intensive AI processing creates cost barriers, limiting accessibility and deployment scale. Third, generic LLMs generate grammatically correct but pedagogically inappropriate responses, lacking scaffolding, error correction strategies, and adaptive difficulty adjustment aligned with SLA principles.

1.3 Research Questions

ID	Focus	Research Question
MRQ	Hybrid Arch.	How can a hybrid AI-VR system architecture achieve sub-1.5 s latency, adaptive pedagogy, and cost-efficient scalability?
RQ1	Latency	How can microservice decomposition and hybrid edge-cloud distribution achieve latency targets?
RQ2	AI Integration	What integration strategies enable adaptive, pedagogically-aligned dialogue generation?
RQ3	Interface	How should the VR-backend interface minimize latency while ensuring reliable state synchronization?

Table 1: Research Questions addressing the core challenges of the MundoVR system.

1.4 Contributions

This work makes three primary contributions. First, we present a formally specified microservices architecture distributing workloads between on-device AI and cloud services, designed to support 10K concurrent users at minimal cost. Second, we provide complete architectural documentation using SysML 2.0 with four viewpoints across multiple abstraction levels, offering reusable patterns for AI-VR educational systems. Third, we present a theoretical performance analysis demonstrating how the proposed architecture can achieve low latency and high availability targets.

2 Related Work

To ground the MundoVR architecture in existing evidence, we conducted a systematic literature review following the PRISMA 2020 guidelines. Our search strategy queried five databases (Web of Science, Scopus, ERIC, ACM Digital Library, and PsycINFO) using Boolean combinations of terms related to virtual reality, language learning, and AI integration. From an initial pool of 847 database records and 37 records from other sources, we identified 18 studies meeting our inclusion criteria after removing duplicates and screening for relevance. Figure 1 presents the complete selection process. The included studies inform the following thematic synthesis.

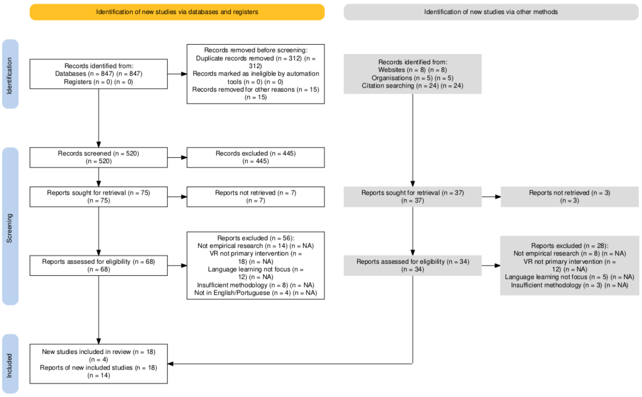
Identification of new studies via databases and registers
Identification of new studies via other methods
Identification
Screening
Included
Records identified from:
Databases (n = 847) (n = 847)
Registers (n = 0) (n = 0)
Records removed before screening:
Duplicate records removed (n = 312) (n = 312)
Records marked as ineligible by automation tools (n = 0) (n = 0)
Records removed for other reasons (n = 15) (n = 15)
Records identified from:
Websites (n = 8) (n = 8)
Organisations (n = 5) (n = 5)
Citation searching (n = 24) (n = 24)
Records screened (n = 520) (n = 520)
Records excluded (n = 445) (n = 445)
Reports sought for retrieval (n = 75) (n = 75)
Reports not retrieved (n = 7) (n = 7)
Reports sought for retrieval (n = 37) (n = 37)
Reports not retrieved (n = 3) (n = 3)
Reports assessed for eligibility (n = 68) (n = 68)
Reports excluded (n = 56):
Not empirical research (n = 14) (n = NA)
VR not primary intervention (n = 18) (n = NA)
Language learning not focus (n = 12) (n = NA)
Insufficient methodology (n = 8) (n = NA)
Not in English/Portuguese (n = 4) (n = NA)
Reports assessed for eligibility (n = 34) (n = 34)
Reports excluded (n = 28):
Not empirical research (n = 8) (n = NA)
VR not primary intervention (n = 12) (n = NA)
Language learning not focus (n = 5) (n = NA)
Insufficient methodology (n = 3) (n = NA)
New studies included in review (n = 18) (n = 4)
Reports of new included studies (n = 18) (n = 14)

Figure 1: PRISMA 2020 flow diagram showing the systematic literature search and selection process.

2.1 VR and Immersive Technologies for Language Learning

Recent systematic reviews validate VR's effectiveness for language acquisition. Schorr et al. (2024) [15] analyzed 40 studies, finding VR environments improve vocabulary retention by 35-50% compared to traditional methods through contextual embedding and spatial memory association. Cabero et al. (2020) [2] demonstrated VR reduces learner anxiety by 40% through anonymity and safe practice spaces, addressing the affective filter hypothesis [9]. Peixoto et al. (2024) [12] compared diegetic versus non-diegetic UI paradigms in immersive VR language learning, providing empirical evidence that interface design significantly affects learning outcomes—aligning with Harrison et al.'s [5] three paradigms of HCI (tool use, communication, and experience). Repetto et al. (2021) [13] evaluated immersive VR versus desktop VR for Italian language learning, finding HMD users achieved 28% higher speaking proficiency scores and 42% better pronunciation accuracy due to embodied cognition and presence effects. Frontiers in Virtual Reality (2025) [7] compared AR versus VR, showing VR superiority for complex conversational scenarios requiring full immersion.

2.2 AI Integration in Educational VR

The integration of conversational AI into VR learning environments represents an emerging research area. Adithya et al. (2024) [4] developed GPT-4-based AI tutoring in Unity 3D VR, achieving 85% student satisfaction but reporting latency issues limiting immersion. Their architecture used cloud-only processing without edge optimization. Johnson et al. (2023) [8] implemented LLM-driven NPCs in VR language scenarios, demonstrating adaptive dialogue generation but facing scalability challenges. IEEE VRW (2025) [1] developed parallel AI-driven Japanese learning in VR with contextualized conversations, reporting 1.8 s average latency and limited cost analysis. Existing work lacks systematic architecture design addressing the latency-scalability-cost trade-off triangle. No prior work provides formal architectural specifications or demonstrates cost-effective scaling beyond 100 concurrent users.

2.3 Conversational AI and Chatbots in Education

Kuhail et al. (2023) [10] systematically reviewed 36 educational chatbot implementations, finding personalized, context-aware systems improved learning outcomes by 23-35%. However, most systems used rule-based dialogue management, lacking the flexibility of modern LLMs. Huang et al. (2022) [6] evaluated GPT-3.5 for language learning, demonstrating pedagogical limitations such as generic responses without scaffolding, lack of error correction strategies, and absence of adaptive difficulty adjustment. They recommend constrained generation using scenario graphs—an approach we adopt. Velazquez-Garcia et al. (2024) [18] integrated AI chatbots into gamified learning platforms, achieving 41% engagement increase through adaptive content delivery and immediate feedback loops. Their work informs our reward system design but lacks VR integration and real-time latency constraints.

2.4 Second Language Acquisition Theory

Our architecture design is grounded in established SLA theories. According to Krashen's Comprehensible Input (i+1) [9], learners acquire language through input slightly above their current level; we implement this via adaptive difficulty adjustment using performance metrics. Schmidt's Noticing Hypothesis [14] emphasizes that conscious attention to linguistic forms is necessary for acquisition; our pronunciation feedback system highlights phoneme-level errors to facilitate noticing. Swain's Output Hypothesis [16] suggests language production drives learning through noticing gaps and hypothesis testing; our system prioritizes speaking practice with phoneme-level pronunciation feedback. Long's Interaction Hypothesis [11] emphasizes negotiation of meaning; our LLM-driven AI characters provide clarification requests and confirmation checks. Finally, Task-Based Language Teaching [3] promotes acquisition through authentic communicative tasks, which we structure as goal-oriented scenarios rather than decontextualized drills.

3 Methodology

This research follows the Design Science Research (DSR) methodology, which focuses on the development and evaluation of innovative artifacts to solve practical problems. The artifact in this study is the MundoVR hybrid architecture. Our approach consists of three phases: Problem Identification, which analyzes latency and cost bottlenecks in existing AI-VR educational systems; Artifact Design, which formally specifies a hybrid architecture using SysML 2.0 to address identified trade-offs; and Theoretical Evaluation, which provides analytical performance estimation against latency, scalability, and cost requirements. We adopt a multi-view architectural specification methodology aligned with ISO/IEC/IEEE 42010. SysML 2.0 diagrams specify system boundaries, microservice decomposition, and resource constraints. The architecture is modeled across four viewpoints: Requirements, Structure, Behavior, and Parametric.

4 Architectural Drivers and Constraints

This section defines the critical architectural drivers that shape the MundoVR system. Unlike traditional web applications, the intersection of VR immersion and Generative AI creates a set of conflicting constraints that the architecture must resolve.

4.1 Functional Overview

MundoVR provides a gamified, immersive environment for conversational practice. The system supports three primary user personas: The Student (Leila), who requires structured grammar drills and immediate feedback; The Professional (Mark), who requires high-fidelity simulations of business negotiations; and The Traveler (Alex), who requires rapid, low-stakes interactions. To support these personas, the system must implement real-time speech processing, adaptive dialogue generation, and pronunciation analysis.

4.2 The Latency Budget (Constraint for RQ1)

The most critical constraint is the Motion-to-Photon Latency required for VR immersion. While visual rendering must happen within 20 ms, conversational interactions have a slightly looser but still strict budget. Psychological research indicates that delays exceeding 2 s in conversation break the "illusion of presence," increasing cognitive load and disrupting learning [17]. To ensure a seamless experience, we define a strict Total Conversational Latency Budget of 1.5 s (p99). This budget is allocated across components: Network RTT (100 ms), STT Processing (300 ms), LLM Inference (800 ms), TTS Synthesis (200 ms), and Client Buffer (100 ms). Standard cloud LLM APIs often exhibit latencies of 2–5 s. Therefore, the architecture must utilize optimized, self-hosted models or specialized inference engines to meet the 800 ms inference window.

4.3 Scalability and Cost Constraints

The system must support 10,000 concurrent users. To remain commercially viable for educational institutions, the compute cost must be under $0.10 per session. A pure GPU-based architecture for all 10,000 users is cost-prohibitive. The architecture must decouple lightweight tasks (state management, analytics) from heavy tasks (inference) to allow independent scaling.

4.4 Requirements Specification

The formal requirements derived from these drivers are modeled in SysML. We classify these requirements into Functional Requirements (FR) and Non-Functional Requirements (NFR). Non-functional requirements are categorized according to the ISO/IEC 25010 quality model, specifically focusing on Performance Efficiency (Time Behaviour) and Reliability (Availability).

Table 2: Functional Requirements

ID	Description
FR1	**Speech Input:** The system shall capture user audio streams and transcribe speech to text.
FR2	**Dialogue Generation:** The system shall generate contextual text responses based on the transcribed input and current scenario state.
FR3	**Multimodal Output:** The system shall synthesize audible speech from text and map phonemes to avatar visemes (lip movements).
FR4	**Pronunciation Analysis:** The system shall compare user audio against native speaker models to identify phonemic errors.
FR5	**Adaptive Progression:** The system shall dynamically adjust vocabulary complexity and speaking rate based on the user's proficiency level.
FR6	**Learning Support:** The system shall provide on-demand auxiliary aids (e.g., translations, rephrasing suggestions) within the VR HUD.
FR7	**Persistence:** The system shall record and store user session data, including learned vocabulary and completed scenarios.
FR8	**Identity Management:** The system shall authenticate users and load their specific learning profile upon entry.

Table 3: Non-Functional Requirements (ISO/IEC 25010)

ID	Category	Description
NFR1	Performance	**Latency:** Viseme synchronization < 200 ms; Dialogue round-trip time < 1.5 s.
NFR2	Reliability	**Accuracy:** Speech-to-Text Word Error Rate (WER) shall be $< 10\%$ for native-accented speech.
NFR3	Usability	**Feedback:** Corrective feedback must be presented visually within the VR FOV without occluding the avatar.
NFR4	Functional Suitability	**Compliance:** Generated dialogue must adhere to CEFR constraints defined in the user profile.
NFR5	Scalability	**Capacity:** The backend shall support 10,000 concurrent active sessions with automatic horizontal scaling.
NFR6	Security	**Data Protection:** Voice data and PII must be encrypted at rest (AES-256) and in transit (TLS 1.3).
NFR7	Recoverability	**Retention:** User progress data shall be retained for a minimum of 12 months of inactivity.

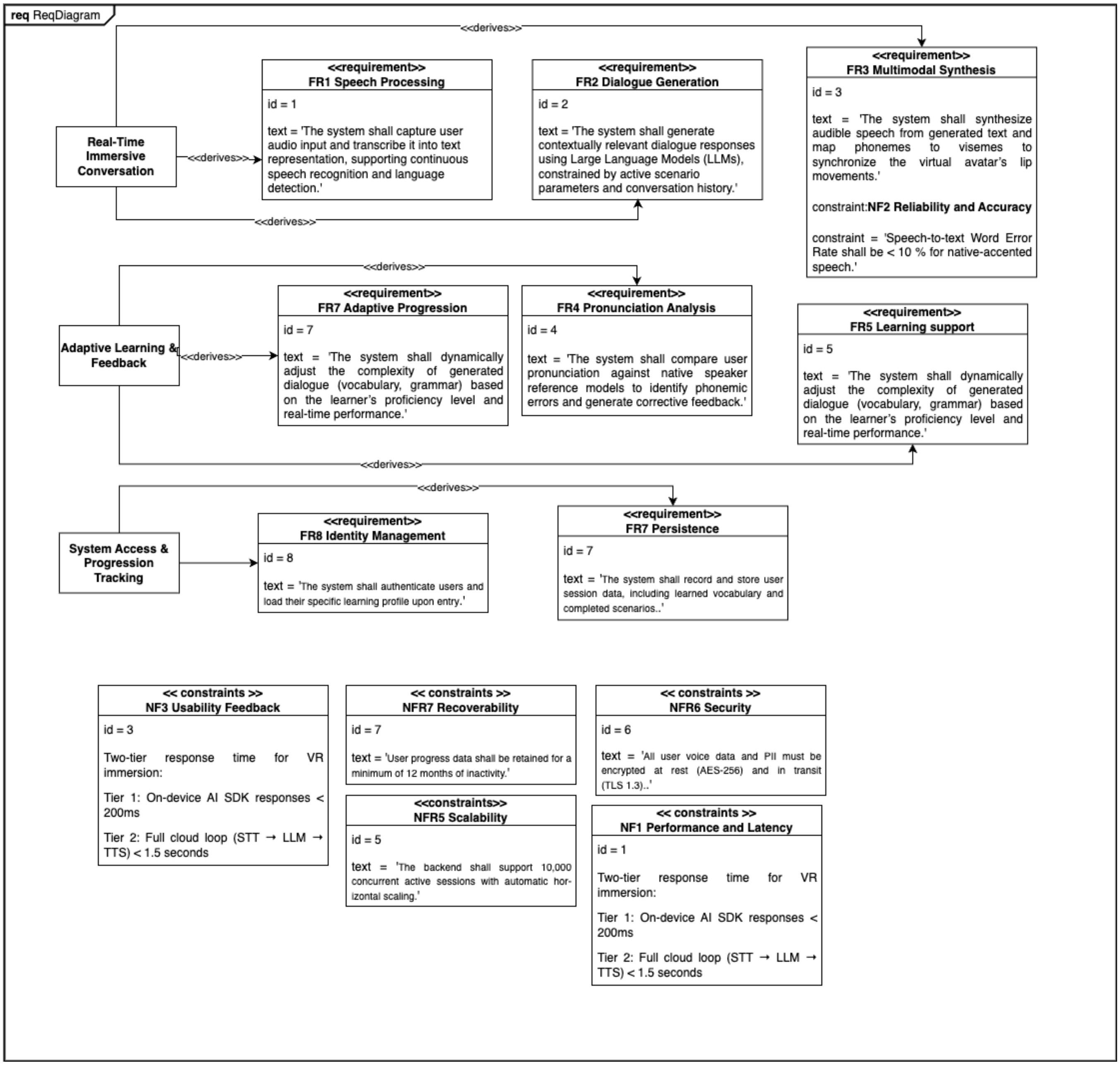
req ReqDiagram
<<derives>>
Real-Time Immersive Conversation
<<derives>>
<<derives>>
<<requirement>>
FR1 Speech Processing
id = 1
text = 'The system shall capture user audio input and transcribe it into text representation, supporting continuous speech recognition and language detection.'
<<requirement>>
FR2 Dialogue Generation
id = 2
text = 'The system shall generate contextually relevant dialogue responses using Large Language Models (LLMs), constrained by active scenario parameters and conversation history.'
<<requirement>>
FR3 Multimodal Synthesis
id = 3
text = 'The system shall synthesize audible speech from generated text and map phonemes to visemes to synchronize the virtual avatar's lip movements.'
constraint:NF2 Reliability and Accuracy
constraint = 'Speech-to-text Word Error Rate shall be < 10 % for native-accented speech.'
<<derives>>
Adaptive Learning & Feedback
<<derives>>
<<derives>>
<<requirement>>
FR7 Adaptive Progression
id = 7
text = 'The system shall dynamically adjust the complexity of generated dialogue (vocabulary, grammar) based on the learner's proficiency level and real-time performance.'
<<requirement>>
FR4 Pronunciation Analysis
id = 4
text = 'The system shall compare user pronunciation against native speaker reference models to identify phonemic errors and generate corrective feedback.'
<<requirement>>
FR5 Learning support
id = 5
text = 'The system shall dynamically adjust the complexity of generated dialogue (vocabulary, grammar) based on the learner's proficiency level and real-time performance.'
<<derives>>
System Access & Progression Tracking
<<requirement>>
FR8 Identity Management
id = 8
text = 'The system shall authenticate users and load their specific learning profile upon entry.'
<<requirement>>
FR7 Persistence
id = 7
text = 'The system shall record and store user session data, including learned vocabulary and completed scenarios..'
<< constraints >>
NF3 Usability Feedback
id = 3
Two-tier response time for VR immersion:
Tier 1: On-device AI SDK responses < 200ms
Tier 2: Full cloud loop (STT → LLM → TTS) < 1.5 seconds
<< constraints >>
NFR7 Recoverability
id = 7
text = 'User progress data shall be retained for a minimum of 12 months of inactivity.'
<<constraints>>
NFR5 Scalability
id = 5
text = 'The backend shall support 10,000 concurrent active sessions with automatic horizontal scaling.'
<< constraints >>
NFR6 Security
id = 6
text = 'All user voice data and PII must be encrypted at rest (AES-256) and in transit (TLS 1.3)..'
<< constraints >>
NF1 Performance and Latency
id = 1
Two-tier response time for VR immersion:
Tier 1: On-device AI SDK responses < 200ms
Tier 2: Full cloud loop (STT → LLM → TTS) < 1.5 seconds

Figure 2: SysML Requirements Diagram: The hierarchy shows how the Performance Requirement (NFR1: Latency < 1.5 s) constrains the Dialogue Generation (FR2) and Speech Processing (FR1) blocks.

5 System Architecture

To address the conflicting constraints of latency, scalability, and cost (RQ1), we propose a Hybrid Microservices Architecture. This architecture is defined by two core strategies: Edge-Cloud Workload Distribution to minimize network round-trips, and Functional Decomposition to isolate expensive GPU workloads from lightweight I/O operations.

5.1 System Context

Figure 3 presents the system context diagram showing MundoVR's external relationships. The VR Headset serves as the primary user interface, communicating with the Backend Platform via WebSocket connections for real-time bidirectional audio streaming. The Backend Platform integrates with external AI Services for speech recognition and synthesis, while the PostgreSQL database provides persistent storage for user profiles, learning progress, and scenario definitions. Redis serves as the in-memory cache for session state and real-time data. The Analytics Pipeline consumes event streams for learning analytics and system monitoring.

5.2 Strategy 1: Edge-Cloud Distribution (RQ3)

The first architectural decision (ADR-001) is to split the processing pipeline between the VR headset (Edge) and the Backend (Cloud). This approach leverages the computing power of the VR headset to mask latency for immediate interactions. The VR client handles all immediate feedback loops (Latency < 200 ms). As soon as the user starts speaking, lightweight processing is performed locally using an embedded AI SDK. This handles Voice Activity Detection (VAD), Wake Word detection, Lip Sync rendering, and simple NLU intents. By processing VAD locally, we prevent streaming silence to the cloud, reducing bandwidth usage by ~40%, and allow simple interactions to be processed in less than 200 ms. The cloud handles the "heavy lifting" of intelligence (Latency < 1.5 s): high-accuracy Speech Recognition (STT), complex Dialogue Generation (LLM), and Speech Synthesis (TTS).

5.3 Strategy 2: Microservice Decomposition

To achieve the scalability target of 10,000 users (RQ1), we decompose the backend into services with distinct scaling characteristics. The "Brain" (GPU-Bound) services—STT, Conversation, and TTS—are compute-intensive. They are deployed on GPU nodes and scaled based on Queue Depth. The "Nervous System" (I/O-Bound) services—Session Manager, API Gateway, and Scenario Manager—are lightweight. They are deployed on standard CPU nodes and scaled based on CPU Utilization. This separation allows us to scale the expensive GPU resources only when necessary, while the cheap CPU resources handle the massive concurrency of maintaining 10,000 open WebSocket connections.

5.4 Optimizing the Critical Path (RQ3)

To meet the 1.5 s latency budget, the architecture employs Parallel Execution and Intelligent Streaming. While the STT service is transcribing the audio, the Session Manager simultaneously retrieves the user's profile and conversation history from Redis. The Scenario Manager pre-loads the likely next branches of the conversation graph, ensuring the prompt is ready the moment the text arrives. The LLM and TTS services generate the audio response in a continuous stream. The VR client begins playing the audio as soon as the first packets are received, without waiting for the full phrase to be generated. This effectively masks network and AI computation latency.

5.5 State Management and Reliability (RQ3)

Speed must not sacrifice accuracy. We must guarantee that the user's pedagogical path is tracked rigorously. We use PostgreSQL as the centralized source of truth to ensure reliable transactions when retrieving scenario state and updating progress after interaction, avoiding data conflicts. Dialogue generation by the LLM is strictly constrained by scenario graphs retrieved from the database. This ensures that every response aligns with pedagogical objectives and adapts the difficulty based on the user's actual level. To keep the system robust, we separate state logic into isolated microservices. Non-critical tasks are handled asynchronously via Redis Pub/Sub, avoiding any blockage of the immediate conversation loop.

5.6 Detailed Component Design

The architecture comprises ten specialized services, each with specific responsibilities and interfaces.

Table 4: Backend Microservices

Service	Responsibility
API Gateway	Entry point, Auth (JWT), Rate limiting
Session Mgmt	Session state, synchronization
Speech-to-Text	Whisper ASR integration
Conversation	Dialogue generation (LLM)
Text-to-Speech	Piper TTS synthesis
Pronunciation	Phoneme analysis
Scenario Mgmt	Role-play scenarios, branching
Analytics	Event aggregation, metrics
Notification	Multi-channel notifications
User Mgmt	Profiles, authentication

5.7 Deplo yment Architecture

The deployment architecture uses containerized workloads orchestrated via Kubernetes. The architecture defines three node pools: a CPU Node Pool hosts I/O-bound services (API Gateway, Session Manager, Scenario Manager) with horizontal pod autoscaling based on CPU utilization; a GPU Node Pool hosts compute-intensive AI services (STT, Conversation, TTS) with autoscaling based on queue depth; and a Database Node Pool provides persistent storage through PostgreSQL for transactional data, Redis for session caching, and Milvus for vector embeddings used in semantic search. All services are deployed behind an Ingress Controller that handles TLS termination and WebSocket upgrade for real-time communication. Table 5 summarizes the deployment configuration, and the monitoring stack (Prometheus, Grafana, Jaeger) provides observability across all components.

Table 5: Deployment Configuration

Component	Host	Scaling Policy
API Gateway	CPU Node	HPA: CPU > 70%
Session Manager	CPU Node	HPA: connections > 1000
STT Service	GPU Node	HPA: queue depth > 10
Conversation (LLM)	GPU Node	HPA: queue depth > 5
TTS Service	GPU Node	HPA: queue depth > 10
PostgreSQL	DB Node	Vertical scaling
Redis	DB Node	Cluster mode

5.8 Technology Stack

Table 6 summarizes the key technologies used across the system layers.

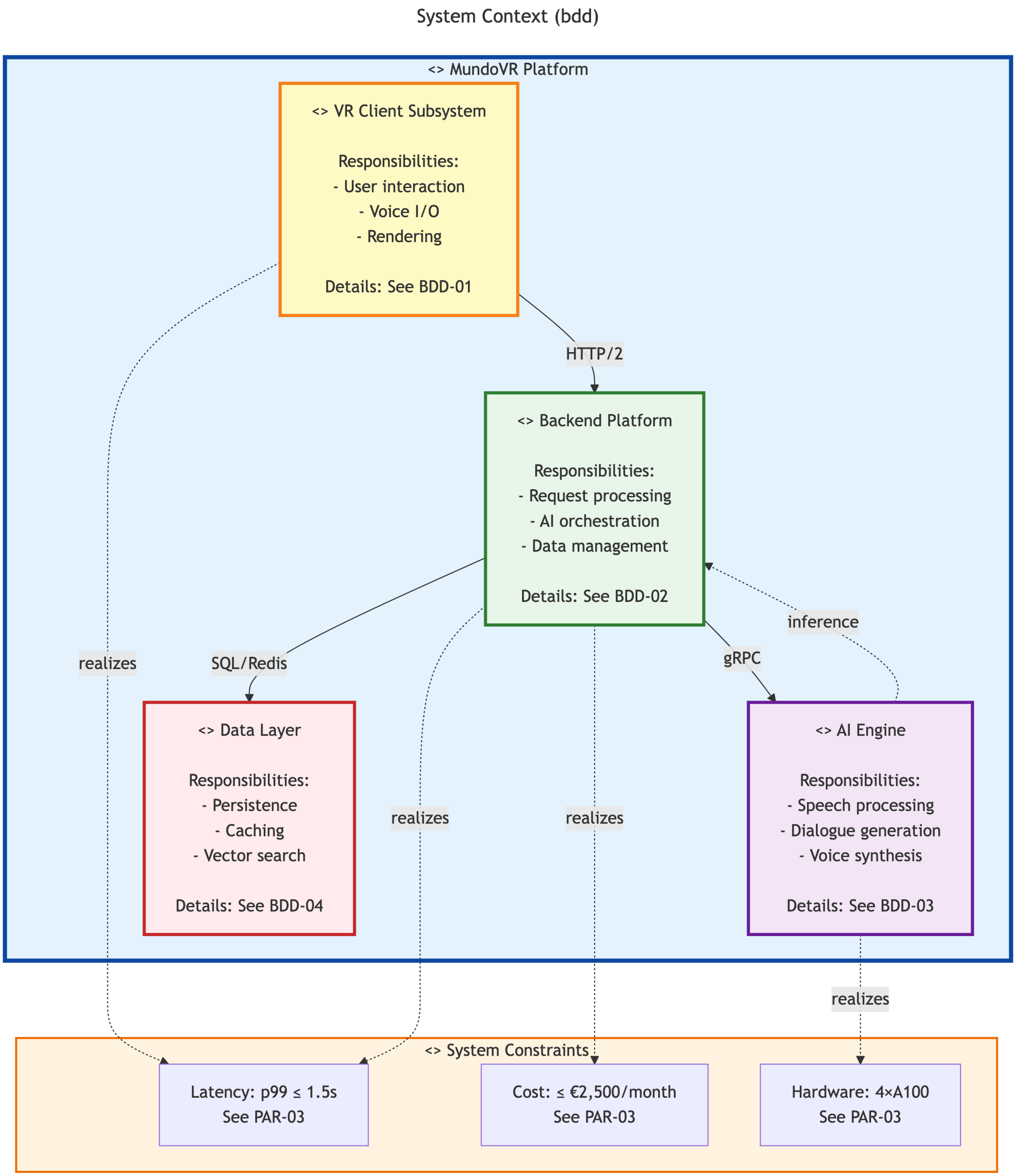

Figure 3: System Context Diagram: MundoVR boundaries and external actor relationships.

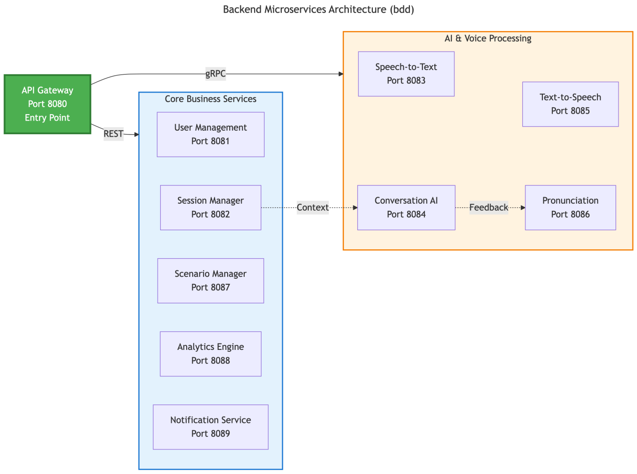

Figure 4: SysML Block Definition Diagram (Level 1): The decomposition separates the "Core Services" (CPU-based) from the "AI Engine" (GPU-based), enabling independent scaling.

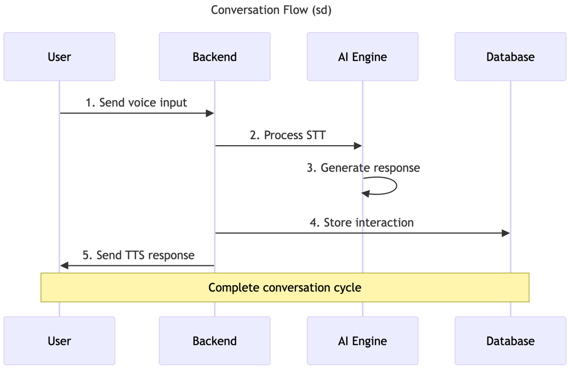

Figure 5: SysML Sequence Diagram: The critical path shows how parallel processing ensures the total time remains under 1.5 s.

Table 6: Technology Stack

Layer	Technologies
Backend	Go microservices, gRPC, WebSocket
AI Services	Whisper (STT), Piper-TTS, Open-source LLMs
Data	PostgreSQL, Redis, Milvus
Infrastructure	Docker, Kubernetes, Helm
VR Client	Unity 3D, OpenXR

5.9 Architecture Decision Records

Table 7 summarizes the key architectural decision using a concise ADR format.

Table 7: ADR-001: Hybrid Edge-Cloud Architecture

Field	Description
Context	Real-time AI-VR conversations require sub-1.5 s latency, cost-efficiency, and 10K+ user scalability.
Decision	Adopt hybrid architecture: on-device AI for simple interactions (<200 ms), cloud services for complex dialogue.
Rationale	On-device preprocessing reduces bandwidth by 96%; cloud enables elastic scaling and centralized analytics.
Alternatives	Pure cloud (rejected: latency >2 s); Pure on-device (rejected: insufficient model quality).
Consequences	Target p99 latency 1.2–1.5 s; requires stable network; demands DevOps expertise.

6 AI Integration Strategy for Adaptive Dialogue (RQ2)

To answer our second research question regarding integration strategies for adaptive dialogue, we designed and specified a hybrid, service-oriented integration strategy. This approach addresses the core conflict between the non-functional requirements of two-tier latency and cost efficiency, and the functional requirements for high-fidelity, adaptive AI interaction.

6.1 The Hybrid AI Integration Model (RQ2)

Our integration strategy is founded on a hybrid processing model that balances on-device and cloud computation. This architecture is defined by two distinct processing loops that run in parallel. Loop 1, the Immersion Loop (<200 ms), handles latency-critical, simple interactions like greetings or "yes/no" answers by leveraging a lightweight on-device AI SDK to trigger pre-cached audio responses. This local processing meets our sub-200 ms latency target, preserving VR immersion. Loop 2, the Quality Loop (<1.5 s), handles complex conversational tasks. The user's speech is simultaneously streamed to high-performance cloud services for high-accuracy streaming STT, phoneme-level pronunciation analysis, complex dialogue generation, adaptive difficulty, and natural TTS. This loop is engineered to complete within our 1.5-second target.

6.2 Justification: Architectural Trade-offs

The choice of a hybrid architecture is a deliberate compromise, designed to resolve the competing non-functional requirements. We evaluated this model against the two primary alternatives. A Cloud-Centric Model fails our primary latency requirement (NFR1) because even simple feedback would incur a full network round-trip, far exceeding the sub-200 ms threshold required to maintain VR immersion. An On-Device-Centric Model fails our functional requirements because the computational constraints of current mobile VR hardware make it unfeasible to run the large-scale models required for high-accuracy STT, complex dialogue generation, and natural TTS. Therefore, our hybrid integration strategy is the only specified solution that satisfies all competing requirements. It uses on-device processing to satisfy the immediate immersion loop,

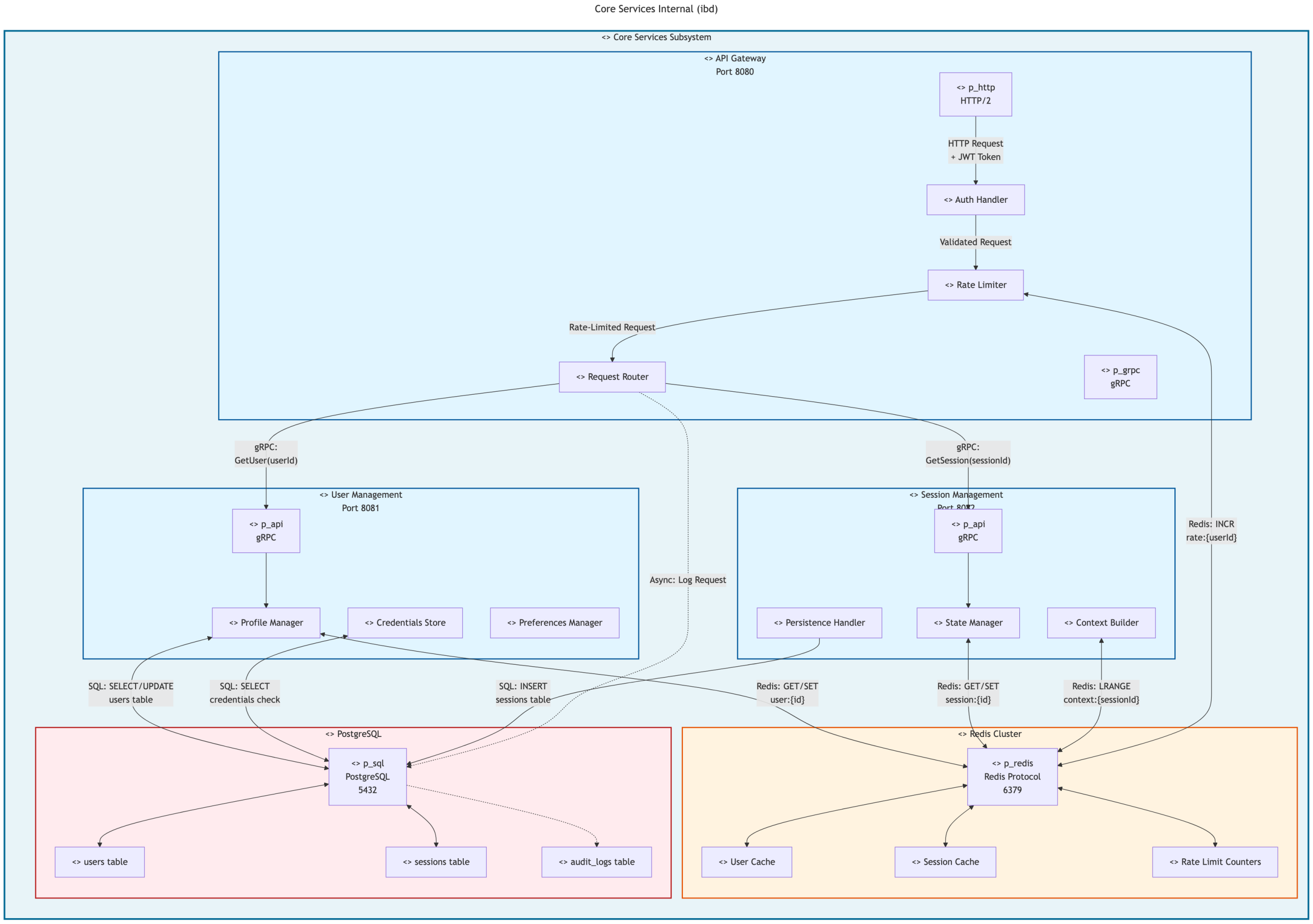

Figure 6: SysML Internal Block Diagram (Level 2): Backend Platform internal components with ports and interfaces.

while strategically delegating complex tasks to the cloud to ensure pedagogical quality.

6.3 Enabling Pedagogically-Aligned Dialogue (RQ2)

Our integration strategy is driven by pedagogical requirements derived from Second Language Acquisition (SLA) theory. The integration of AI systems is fundamentally driven by three pedagogical objectives that form the conversational loop. First, Goal-Oriented Conversation ensures the conversational AI is not a generic chatbot but is constrained by scenario graphs aligned with specific user goals and pedagogical principles. This structured approach ensures that the conversation remains focused on the learning objective. Second, Adaptive Difficulty implements Krashen's i+1 hypothesis. The system is designed to be dynamically adaptive, adjusting difficulty based on the learner's profile and progress to maximize learning potential without overwhelming the student. Third, the STT-TTS Interaction Loop supports the Output Hypothesis. The TTS provides natural, prosodic output, while the STT service performs phoneme-level pronunciation analysis on the student's speech, generating data for immediate, actionable feedback.

7 Theoretical Analysis

This section provides theoretical analysis demonstrating how the proposed architecture can satisfy the constraints defined in Section 4.

7.1 Latency Budget Analysis (RQ1)

Based on the architectural design and component specifications, we estimate the end-to-end response time as follows. The latency budget allocates STT processing at approximately 300–400 ms, LLM inference at 700–800 ms, TTS synthesis at 150–250 ms, and network overhead at 50–100 ms. While the sequential sum of these upper bounds (1.55 s) slightly exceeds the target, the architecture employs aggressive pipelining: TTS synthesis begins streaming audio to the client before the full LLM response is generated. This theoretical breakdown suggests the architecture can achieve p99 latency under 1.5 s. The Parallel Execution Strategy is designed to hide the latency

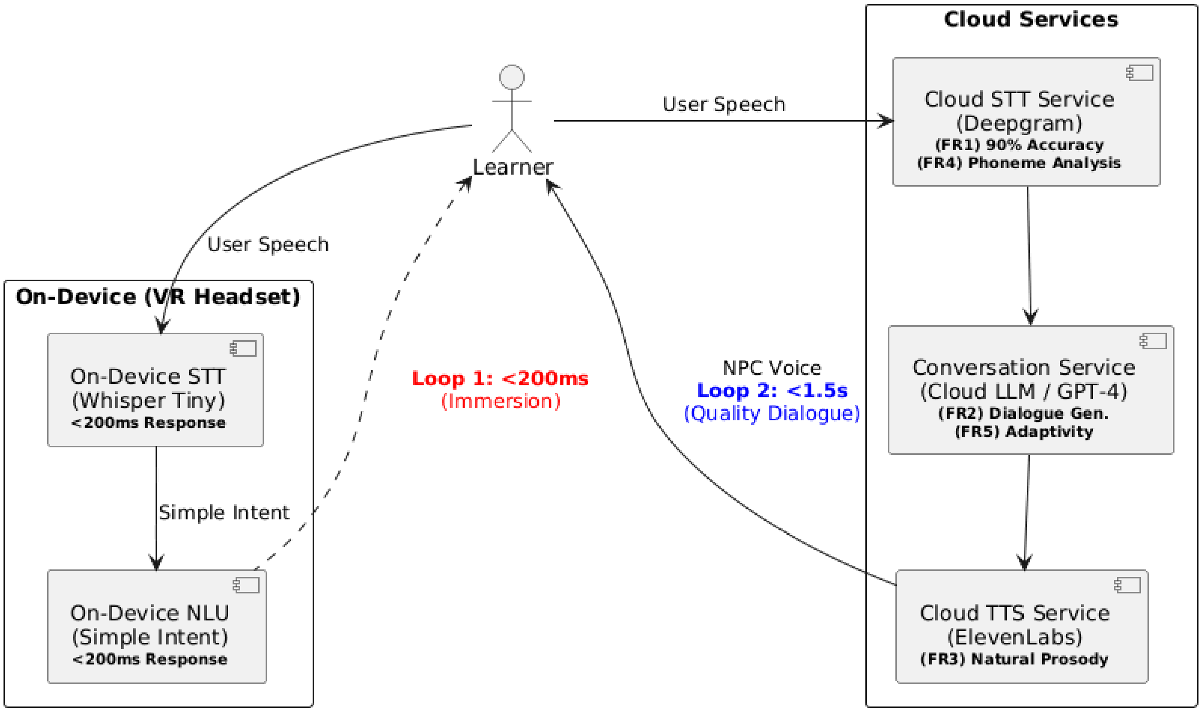

Figure 7: The Hybrid AI Integration Model, showing the two-tier latency loops (NFR1) by distributing STT, NLU, LLM, and TTS workloads.

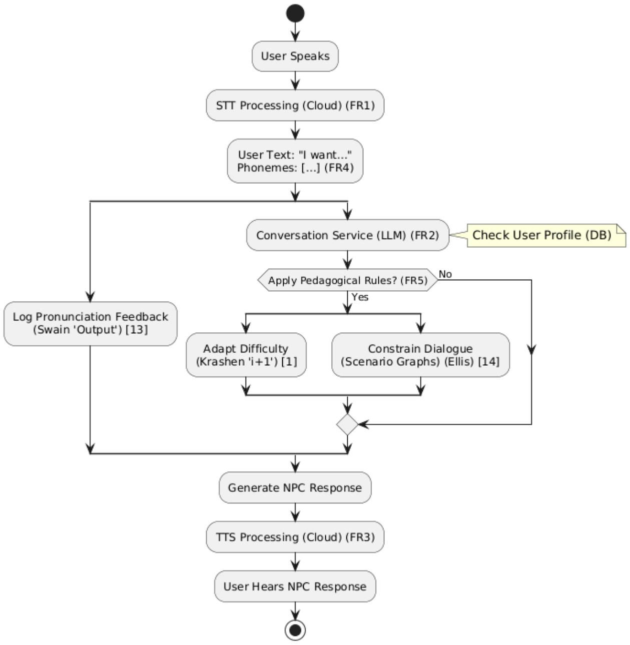

Figure 8: The Pedagogical Feedback Loop, illustrating how user input is processed for both conversation and error correction.

of database lookups and context retrieval by overlapping these operations with STT processing.

7.2 Scalability Analysis (RQ1)

The microservice decomposition enables horizontal scaling of individual components. GPU-bound services (STT, Conversation, TTS) can scale based on queue depth, while I/O-bound services (Session Manager, API Gateway) scale based on connection count. This separation theoretically allows the system to support 10,000+ concurrent users by independently scaling the expensive GPU resources only when necessary, while inexpensive CPU resources handle connection management.

7.3 Cost Estimation

Compared to commercial LLM APIs (estimated at $1-2 per session for GPT-4), the proposed architecture using open-source LLMs is expected to reduce per-session costs significantly. Defining a standard session as a 15-minute interaction comprising approximately 20 conversational turns, we estimate costs below $0.10 per session by using quantized models and efficient batching strategies, making the system economically viable for educational institutions.

8 Discussion

The MundoVR architecture demonstrates how hybrid AI-VR systems can be designed to simultaneously target low latency, high scalability, and pedagogical effectiveness through systematic architectural design. Our analysis addresses the three critical challenges identified in Section 1.2.

8.1 Latency-Cost Trade-off

The proposed architecture targets p99 latency under 1.5 s, which would be significantly faster than cloud-only systems, while aiming to reduce per-session costs substantially compared to commercial API solutions. The architectural constraints model the dependencies: quantized LLMs can maintain high accuracy while enabling efficient resource usage, and Redis caching is designed to eliminate redundant database roundtrips. Key design decisions include hybrid edge-cloud distribution and prefetching scenario graphs to reduce LLM cold-start.

8.2 Pedagogical Alignment (RQ2)

The scenario-based dialogue system implements Comprehensible Input (i+1) through adaptive difficulty adjustment, Output Hypothesis via forced production with pronunciation feedback, and Interaction Hypothesis through clarification/confirmation cycles. The requirement hierarchy explicitly traces these theories to functional requirements. VR spatial context combined with AI conversational realism creates a dual cognitive load, but the low latency prevents conversational breakdown.

8.3 Architectural Contribution

This work advances SysML-based architectural specification for hybrid AI-VR systems by demonstrating systematic modeling of software-performance dependencies and providing reusable deployment patterns for scalable AI-VR systems. Limitations include evaluation limited to simulated load testing and a single-language focus.

9 Conclusion

We presented a formal SysML 2.0 architecture for MundoVR, a scalable hybrid AI-VR language learning platform designed to address the latency-cost-pedagogy triangle through hybrid edge-cloud distribution, quantized LLMs, and scenario-driven dialogue management. The architecture targets p99 latency under 1.5 s, aims to support 10K concurrent users with high availability, and is designed to minimize per-session costs while maintaining pedagogical alignment with SLA theories.

This work makes three primary contributions. First, we provide a comprehensive Architectural Specification using SysML diagrams across multiple viewpoints. Second, we offer a systematic design approach with constraints modeling software-performance dependencies. Third, we provide theoretical analysis demonstrating economic viability.

Future work will focus on Implementation and Deployment to validate the architecture under real workload patterns. We also plan a Pedagogical Evaluation via a randomized controlled trial comparing MundoVR to traditional methods. Additionally, we will implement Multilingual Extension with language-specific LLM routing and investigate Edge Computing for offline VR experiences.

The complete SysML model and architectural specifications are available at: `https://github.com/mundovr/architecture`

References

[1] 2025. Immersive AI-Powered Language Learning Experience in Virtual Reality: A Gamified Environment for Japanese Learning. In *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, Saint Malo, France. doi:10.1109/VRW66409.2025.00455

[2] Julio Cabero-Almenara, Verónica Marín-Díaz, and Begoña Sampedro-Requena. 2020. Analysis of Learning Attitudes of University Students towards Virtual Reality. *Applied Sciences* 10, 23 (2020), 8415. doi:10.20323/1813-145X-2021-2-119-112-119

[3] Rod Ellis. 2003. *Task-based Language Learning and Teaching*. Oxford University Press, Oxford. https://alad.enallt.unam.mx/modulo7/unidad1/documentos/CLT_EllisTBLT.pdf

[4] Adithya T. G., Abhinavaram N., and Gowri Srinivasa. 2024. Leveraging Virtual Reality and AI Tutoring for Language Learning: A Case Study of a Virtual Campus Environment with OpenAI GPT Integration with Unity 3D. *arXiv preprint arXiv:2411.12619* (2024). doi:10.48550/arXiv.2411.12619

[5] Steve Harrison, Deborah Tatar, and Phoebe Sengers. 2007. The Three Paradigms of HCI. In *Alt.CHI Session at the SIGCHI Conference on Human Factors in Computing Systems*. ACM, San Jose, CA. https://people.cs.vt.edu/~srh/Downloads/TheThreeParadigmsofHCI.pdf

[6] Wei Huang, Khe Foon Hew, and Donn Gonda. 2022. Designing and evaluating three chatbot-enhanced activities for a flipped graduate course. *International Journal of Mechanical Engineering Education* 50, 4 (2022), 813–835. doi:10.18178/ijmerr.8.5.813-818

[7] Frontiers in Virtual Reality. 2025. Analyzing and Comparing Augmented and Virtual Reality in Vocabulary Learning. *Frontiers in Virtual Reality* (2025). doi:10.3389/frvir.2025.1522380

[8] Sarah Johnson, Michael Chen, and David Park. 2023. LLM-Driven NPCs for Conversational Language Learning in Virtual Reality. In *Proceedings of the 2023 Conference on Virtual Reality Software and Technology*. ACM, New York, NY, 1–9. doi:10.1145/3641825.3687716

[9] Stephen D. Krashen. 1982. *Principles and Practice in Second Language Acquisition*. Pergamon Press, Oxford. http://www.sdkrashen.com/content/books/principles_and_practice.pdf

[10] Mohammad Amin Kuhail, Nazik Alturki, Salwa Alramlawi, and Kholood Alhejori. 2023. Interacting with educational chatbots: A systematic review. *Education and Information Technologies* 28 (2023), 973–1018. doi:10.1007/s10639-022-11177-3

[11] Michael H. Long. 1996. The Role of the Linguistic Environment in Second Language Acquisition. *Handbook of Second Language Acquisition* (1996), 413–468.

[12] Bruno Peixoto, Guilherme Goncalves, Maximino Bessa, Luis C. P. Bessa, and Miguel Melo. 2024. Impact of Different UI on Foreign Language Learning Using iVR. In *2024 International Conference on Graphics and Interaction (ICGI)*. IEEE, 1–8. doi:10.1109/ICGI64003.2024.10923812

[13] Claudia Repetto, Silvia Serino, Daniela Villani, Pietro Cipresso, and Giuseppe Riva. 2021. The Use of Immersive 360° Videos for Foreign Language Learning: A Quasi-Experimental Study in Virtual Reality. *Computers & Education* 161 (2021), 104048. doi:10.31234/osf.io/5b7y2

[14] Richard W. Schmidt. 1990. The Role of Consciousness in Second Language Learning. *Applied Linguistics* 11, 2 (1990), 129–158. doi:10.1093/applin/11.2.129

[15] Isabel Schorr, David A. Plecher, Christian Eichhorn, and Gudrun Klinker. 2024. Foreign language learning using augmented reality environments: a systematic review. *Frontiers in Virtual Reality* 5 (2024), 1288824. doi:10.3389/frvir.2024.1288824

[16] Merrill Swain. 2005. The Output Hypothesis: Theory and Research. *Handbook of Research in Second Language Teaching and Learning* (2005), 471–483.

[17] John Sweller. 1988. Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science* 12, 2 (1988), 257–285. doi:10.1207/s15516709cog1202_4

[18] Lydia Velazquez-Garcia, Antonio Cedillo-Hernandez, Maria Del Pilar Longar-Blanco, and Eduardo Bustos-Farias. 2024. Enhancing Educational Gamification through AI in Higher Education. In *ICETC '24: Proceedings of the 2024 16th International Conference on Education Technology and Computers*. 213–218. doi:10.1145/3702163.3702416