

# **ARCHITECTURE AND SYNCHRONIZATION MODEL OF A DISTRIBUTED WEB-BASED GAMING VIDEO CONFERENCING PLATFORM USING AUGMENTED REALITY**

*Uzenbek E.R (China)*

*Northwestern Polytechnical University*

*Xi'an, China*

The development of web technologies and the widespread adoption of video conferencing have led to a transition from passive forms of remote communication to interactive collaborative environments. Modern video conferencing systems based on WebRTC provide low-latency audio and video transmission; however, they remain limited in terms of spatial interaction between users and virtual objects.

Augmented Reality (AR) technologies enable the integration of three-dimensional objects into the user's visual environment, creating new scenarios for collaborative interaction. A key challenge in such systems is ensuring the consistency of the virtual scene state within a distributed environment under conditions of network latency, bandwidth variability, and concurrent user actions.

The objective of this study is to develop the architecture of a distributed web-based platform for gamified video conferencing with AR functionality and to formalize a synchronization model of the scene state that accounts for the temporal characteristics of the system.

Real-time collaborative interactive systems are studied within the fields of distributed computing, collaborative systems, and immersive environments.

WebRTC is widely used for low-latency media transmission in browser-based applications [1]. However, most existing research primarily focuses on audio and video delivery, without addressing the problem of consistent replication of shared virtual state.

In the field of collaborative AR, research investigates models of spatial synchronization and object replication in shared virtual environments [2,3]. The main challenges include:

- maintaining state consistency under concurrent user actions;
- compensating for network latency;
- ensuring architectural scalability.

Distributed systems employ various consistency models, including strong consistency, eventual consistency, and CRDT-based approaches [4]. For interactive AR systems, strict consistency often results in increased latency, making eventual consistency with conflict resolution mechanisms a more suitable model.

Thus, existing solutions do not provide a unified architecture for browser-based AR video conferencing with a formalized latency and synchronization model, which highlights the relevance of the present study.

The proposed distributed web-based platform for gamified video conferencing with augmented reality is formalized as a system:

$$S = (U, C, O, E, \Phi)$$

where  $U$  denotes the set of users,  $C$  represents the set of network connections implemented via WebRTC,  $O$  is the set of virtual AR objects,  $E$  is the set of interaction events, and  $\Phi$  is the scene state update function.

Each user maintains a local replica of the scene state  $State_i(t)$ , consisting of a collection of objects  $O_k$  described by position, orientation, scale, ownership, and timestamp parameters. State updates are performed using an event-driven model:

$$State(t + 1) = \Phi(State(t), E_l),$$

where the event  $E_l$  is defined as a tuple  $(user_i, object_j, action, ts_l)$ . This formalization allows the system to be interpreted as a distributed event-driven environment with state replication.

The platform architecture consists of three functional layers. The client layer performs video capture, gesture recognition using MediaPipe Hands, interaction logic

processing through a finite state machine (FSM), and three-dimensional scene rendering using Three.js. The signaling layer is implemented using Node.js and Socket.IO and is responsible for establishing WebRTC connections, managing rooms, and exchanging control messages. The synchronization layer handles the generation, transmission, and application of interaction events, ensuring consistency of local AR scene replicas across all participants.

The use of an event-based replication model minimizes the amount of transmitted data and enables interactive collaboration in small distributed groups while maintaining acceptable temporal performance characteristics.

The scene state at time  $t$  is defined as:

$$State(t) = \{O_1(t), O_2(t), \dots, O_m(t)\}$$

Each object is represented as:

$$O_i = (x_i, y_i, z_i, r_i, s_i, owner_i, ts_i)$$

where  $x_i, y_i, z_i$  denote spatial coordinates,  $r_i$  represents orientation,  $s_i$  denotes scale,  $owner_i$  identifies the current owner of the object, and  $ts_i$  is the associated timestamp.

An interaction event is defined as:

$$E_k = (user_j, object_i, action, params, ts_k)$$

The state transition function is defined as:

$$State(t + 1) = \Phi(State(t), E_k)$$

To prevent conflicts during concurrent interactions, an ownership model is applied. When a Grab action occurs, the object is assigned to the initiating user as its temporary owner. Any modification attempts by other users are ignored until a corresponding Release event is received. Timestamps are used to ensure proper temporal ordering of events and to resolve potential race conditions.

The total system latency is defined as:

$$T_{total} = T_{capture} + T_{gesture} + T_{Encoding} + T_{network} + T_{sync} + T_{render}$$

The analysis indicates that when the number of participants is small, the dominant components of latency are  $T_{gesture}$  and  $T_{render}$ , as most processing occurs locally on the client side. However, as the number of participants  $n$  increases, the contribution of  $T_{network}$  becomes significant due to the growth in the number of active connections and event transmissions.

In a distributed AR system, the scene state is replicated across all participating clients. When users perform concurrent actions, several issues may arise, including state inconsistency, violations of causal ordering, and object “jumps” resulting from state divergence.

Let the scene state on client  $k$  at time  $t$  be denoted as:

$$State_k(t)$$

The consistency requirement can be expressed as:

$$\lim_{t \rightarrow \infty} State_i(t) = State_j(t)$$

for any clients  $i$  and  $j$ , provided that no new interaction events are generated.

This corresponds to the eventual consistency model.

The proposed system implements eventual consistency combined with a temporary object ownership mechanism. Let  $O_i$  denote a scene object, and let  $Owner(O_i, t)$  be a function defining the owner of the object at time  $t$ .

The correctness condition is defined as:

$$\forall O_i, \exists \leq 1 user_j: Owner(O_i, t) = user_j$$

Thus, at any given moment, each object has at most one active owner. After the interaction is completed, ownership is released:

$$Owner(O_i, t + \Delta t) = \emptyset$$

This mechanism reduces the probability of conflicts and ensures deterministic state updates across distributed clients.

The previously introduced latency decomposition is given by:

$$T_{total} = T_{capture} + T_{gesture} + T_{encoding} + T_{network} + T_{sync} + T_{render}$$

The network component can be further decomposed as:

$$T_{network} = T_{uplink} + T_{server} + T_{downlink}$$

In a mesh-based architecture, the computational and communication load on each client grows proportionally to:

$$Load \propto n - 1$$

The total number of peer-to-peer connections in the system is defined as:

$$\frac{C = n(n - 1)}{2}$$

Therefore, as the number of participants increases, the total number of connections grows quadratically, while the load on each individual client increases linearly. This characteristic inherently limits the scalability of mesh-based AR video conferencing systems and motivates the consideration of alternative architectural models for larger groups.

This study presents the architecture of a distributed web-based platform for gamified video conferencing with integrated augmented reality technologies. A formal system model has been proposed, incorporating users, network connections, virtual objects, and interaction events, along with an event-driven synchronization model of the AR scene state featuring ordering and temporary object ownership mechanisms. In addition, a decompositional model of total system latency has been formulated, enabling evaluation of the contribution of individual components to the overall interactivity of the environment.

The scalability analysis of the mesh architecture and the experimental evaluation of temporal characteristics demonstrate that the proposed solution provides acceptable latency levels and stable performance for small groups of users. The results confirm the feasibility of implementing collaborative AR environments within a browser-based setting using WebRTC, without requiring specialized hardware or native applications.

The proposed architecture establishes a foundation for further research in distributed real-time AR systems, including the development of advanced state

consistency models, improved scalability strategies, and adaptive network delay compensation mechanisms.

Reference:

1. Alvestrand H. Overview: Real Time Protocols for Browser-Based Applications. IETF RFC 8825, 2021.
2. Billinghurst M., Clark A., Lee G. A Survey of Augmented Reality // Foundations and Trends in Human-Computer Interaction. 2015. Vol. 8(2-3).
3. Ens B., Lanir J., Tang A., Bateman S. Revisiting collaboration through augmented reality: The evolution of groupware // IEEE Computer Graphics and Applications. 2019.
4. Shapiro M., Preguiça N., Baquero C., Zawirski M. Conflict-Free Replicated Data Types // Stabilization, Safety, and Security of Distributed Systems. 2011.
5. Jesup R., Loreto S., O’Callaghan J. WebRTC 1.0: Real-Time Communication Between Browsers. W3C Recommendation, 2021.
6. Claypool M., Claypool K. Latency and Player Actions in Online Games // Communications of the ACM. 2006.