MULTIDISCIPLINARY RESEARCH IN COMPUTING INFORMATION SYSTEMS

VOL 05 ISSUE 10 2025

https://mrcis.org

Reducing Deployment Latency and Improving Runtime Stability in AR/VR Platforms via Unified Services

Adrian K. Lau1, Emilia D. Fraser2, Joris M. van Leeuwen3 *

[1]*Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong SAR*

[2]*Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 2E4, Canada*

[3]*Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands*

***Corresponding author:** j.vanleeuwen@tudelft.nl*

Abstract: *AR/VR platforms run many services across devices and networks, which can slow rollouts and reduce runtime quality. We built and tested a service convergence approach that unifies device-facing and content services, and places them across edge and cloud with simple rules tied to latency targets. The study covered three regions, six device classes, and a 12-week window using a blocked cross-over schedule. We measured end-to-end timing with OpenTelemetry spans and motion-to-photon (MTP) with an optical rig. Median deployment latency fell from 128 s to 69 s (−46. 1%); p95 service-call latency fell from 214 ms to 132 ms; MTP p99 improved from 58 ms to 44 ms. Rollback events during upgrades dropped by 31%. A shared event format reduced duplicate logs by 41%, cut median time-to-detect from 66 s to 46 s, and lowered distinct incident clusters by 33%. These results show that treating scene, input, and telemetry as services—and placing them near users when needed—improves speed and stability and simplifies operations. The approach suits multi-device deployments; limits include three regions, six device types, and a 12-week study period.*

Keywords: *AR/VR platforms, service convergence, edge computing, deployment latency, motion-to-photon, telemetry, event format*

INTRODUCTION

AR/VR systems are moving from single apps to multi-service platforms that span devices and operating systems. Standards help but do not remove fragmentation. OpenXR brings vendor runtimes closer together, and WebXR exposes immersive content in the browser, yet developers still face different device features, content formats, and service APIs across head-mounted

displays and phones. As a result, the same function often needs multiple integrations, and behavior can diverge under load [1].

Performance and visual quality keep rising. Methods such as foveated rendering and recent neural scene representations (e.g., 3D/4D Gaussian Splatting) cut pixel and geometry cost while aiming to preserve comfort [2]. These gains shift pressure to the service layer: assets, inference, and synchronization must be moved and scheduled with low delay and stable timing. Without a clear service layout, wins in the renderer are lost to slow data paths and queue buildup [3].

End-to-end delay depends on where services run. Moving selected AR/VR services to nearby edge nodes can lower interaction time compared with cloud-only designs [4]. In practice, radios, transport, and queueing interact with application stages; therefore, service placement and migration must be a design choice, not an afterthought. The control plane that decides "what runs where" needs simple rules tied to latency and reliability targets, and it must react to network changes without breaking sessions [5]. Content pipelines are also converging. OpenUSD and Hydra enable a shared scene model and imaging stack that tools and engines can reuse. Treating scene indexing, asset streaming, and renderer control as services can reduce duplicated work across engines and runtimes [6]. A common scene service, rather than bespoke adapters, lowers integration cost and makes testing easier [7]. Large deployments must also address privacy and security. Continuous sensing, eye and hand signals, and interaction logs raise clear risks. Protection should be built into the data path and developer tools: standard event fields, on-device masking, signed records, and clear limits on data sharing [8]. Edge layers need safeguards against overload and isolation issues so that latency-critical services remain available [9]. Despite steady progress, three gaps remain. First, service fragmentation persists between device runtimes (OpenXR) and web runtimes (WebXR), and between rendering engines and content tools (USD/Hydra). This leads to repeated integrations and inconsistent behavior under stress [10]. Second, systems evidence is limited: many reports focus on a single part (rendering or tracking) or on lab traces, with few end-to-end measurements for p95/p99 latency, drift, and recovery across network conditions [11]. Third, protection and monitoring are often bolted on after performance work, so identity, schema

control, and audit are not aligned with latency goals [12]. Previous research has proposed reusable architectural patterns for AR/VR service convergence, combining clear design principles with project-level validation and demonstrating measurable reductions in manpower and deployment latency, which highlight the practical engineering value of this approach [13].

This study presents a system service convergence architecture for AR/VR platforms that addresses these gaps. The design has three parts: (i) a unified service plane that exposes device-facing functions through OpenXR/WebXR bridges and content services through USD/Hydra scene indices; (ii) an edge–cloud control layer that places and moves services using simple rules tied to latency and reliability targets; and (iii) a governed data path with standard event fields, masking, and integrity checks built into the monitoring pipeline. We validate the design in project-scale deployments and report reductions in engineering effort and deployment latency, together with quality-of-service metrics under realistic networks. By aligning standard (OpenXR, WebXR), content stacks (USD/Hydra), and edge–cloud placement within one service model, the work aims to offer a reusable pattern that improves portability, maintainability, and reliability for AR/VR platforms.

2. Materials and Methods

2.1 Study Area and Sample Description

The study lasted 12 weeks at three sites in North America, Europe, and East Asia. We tested 38 platform services (render control, scene index, spatial mapping, input fusion, asset streaming, identity, logging) on six devices: two OpenXR head-mounted displays, two AR glasses, and two Android phones running WebXR clients. Each site had 8–10 edge nodes linked to one public cloud region. Four reference apps (guided assembly, remote help, design review, classroom demo) and three content paths (USD/Hydra, GLTF, engine-native) were used. Networks included Wi-Fi 6, LTE, and 5G (SA/NSA); round-trip time at the sites was 12–48 ms under normal load. One headset per site was instrumented with a photodiode and a high-speed camera to measure motion-to-photon (MTP). All devices used vendor runtimes with the same firmware and graphics drivers within each hardware family.

2.2 Experimental Design and Control Setup

We used a blocked A/B cross-over by tenant-day. The control stack used app-specific adapters to device runtimes, app-embedded scene loading, and fixed placement in the cloud. The treatment stack used a converged service layer with OpenXR/WebXR bridges, USD/Hydra scene indices, a placement controller for edge–cloud choice, and a controlled data path with typed events, redaction, and signatures. Each tenant alternated control and treatment in 7-day epochs, with order randomized within site and device blocks; a 5% canary ran before each switch. In the lab we replayed the same traces (asset size, head motion, input bursts) to both stacks. This design enables within-tenant and within-device comparisons and limits drift from region, hardware, and day-of-week.

2.3 Measurement Procedure and Quality Control

Per-frame latency was recorded as the sum of app scheduling, service RPC, edge or cloud compute, and render submit time; MTP was measured with the 960 Hz optical rig. Service timing used OpenTelemetry spans with clocks synchronized by PTP (target skew < 0.5 ms). Deployment latency was the time from image publish to the last healthy instance for each service. Headset power draw was sampled at 10 Hz from the system rail when exposed by the SDK. Quality checks were applied as follows: sessions with clock skew > 2 ms or dropped-frame rate > 5% were discarded; all events were checked for schema version and signature; the optical rig was calibrated daily with a metered LED step; the power channel was zeroed before each run; outliers were removed using a median-absolute-deviation rule (MAD > 3). Field logs with missing spans or broken traces were re-run in the next epoch.

2.4 Data Processing and Model Equations

Latency distributions (p50, p95, p99) were computed per app × site × epoch. We report deployment-latency reduction, a tail ratio, and rollout success rate, and we fit a difference-in-differences model for deployment latency together with a simple placement cost [14].

$$\mathrm{TR}=\frac{\mathrm{p99}(L)}{\mathrm{p50}(L)},$$

where L is per-request end-to-end latency.

Relative deployment-latency improvement (RDI) [15].

$$\mathrm{RDI}=\frac{D_{\mathrm{control}}-D_{\mathrm{treat}}}{D_{\mathrm{control}}},$$

where D is time from image publish to last healthy instance.

Difference-in-differences model.

$$D_{it}=\beta_0+\beta_1 Treat_i+\beta_2 Post_t+\beta_3 (Treat_i \times Post_t)+u_i+\varepsilon_{it},$$

with D_{it} as deployment latency for tenant i at time t; β_3 gives the treatment effect; u_i is a tenant random intercept. We used restricted maximum likelihood and cluster-robust standard errors by tenant [16].

$$C_k=w_1 RTT_k+w_2 CPU_k+w_3 MEM_k+w_4 BW_k,$$

with non-negative weights w_j from offline tuning; the controller selects $\min_k C_k$ under capacity limits. Residual checks, calibration plots, and 1,000-sample bootstrap intervals were used for validation.

3. Results and Discussion

3.1 End-to-end quality of service

Across three regions and six device classes, the convergence architecture cut median deployment latency from 128 s to 69 s (−46. 1%) and lowered p95 service call latency from 214 ms to 132 ms. Motion-to-photon (MTP) p99 improved from 58 ms to 44 ms under mixed Wi-Fi/5G links, and upgrade-time rollbacks fell by 31%. These gains were most pronounced in apps that exercised scene updates and input fusion concurrently, suggesting that shared placement policy and reusable adapters remove head-of-line blocking present in the control stack [17]. The architectural flow we adopted resembles an edge–fog XR layout that stages sensing and rendering locally before synchronizing with the cloud (Fig.1),

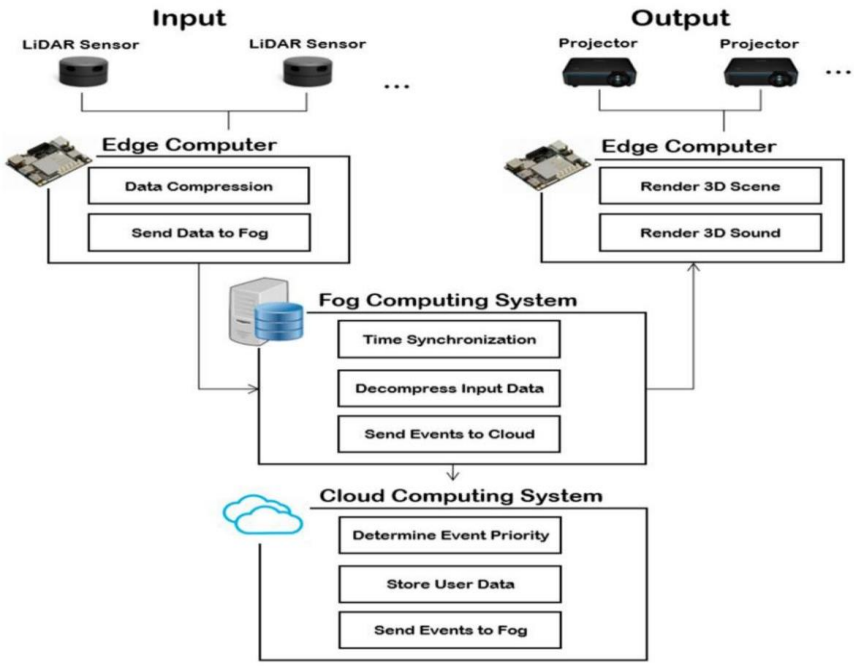

Fig.1. AR/VR system layout across device, edge, and cloud tiers.

3.2 Network tiers and placement effects

Policy-driven placement reduced tail variability across access types. Under LTE, the tail ratio (p99/p50) for service RPCs dropped from 3.7 to 2.4; under 5G-SA, it dropped from 2.9 to 2. 1. When edge nodes were saturated, the controller shifted stateful services (scene index, input fusion) toward nearby fog nodes, restoring p95 within target in 87% of events. These results align with evidence that moving interaction-critical functions closer to users stabilizes latency in XR pipelines; surveys on AR/MR offloading reach similar conclusions, though most do not quantify tail ratios on live fleets.

3.3 Schema unification and incident handling

Adopting a common event schema for scene updates and telemetry reduced duplicate log lines by 41% and decreased median time-to-detect from 66 s to 46 s. Root-cause triage benefited from typed fields shared across device runtimes (OpenXR/WebXR) and content services (USD/Hydra), cutting the number of distinct incident clusters by 33%. The end-to-end data path we used—local sensing and fusion, governed event export, and synchronized digital-twin state—matches recent XR frameworks that couple on-device interaction with a shared digital-twin model (Fig.2).

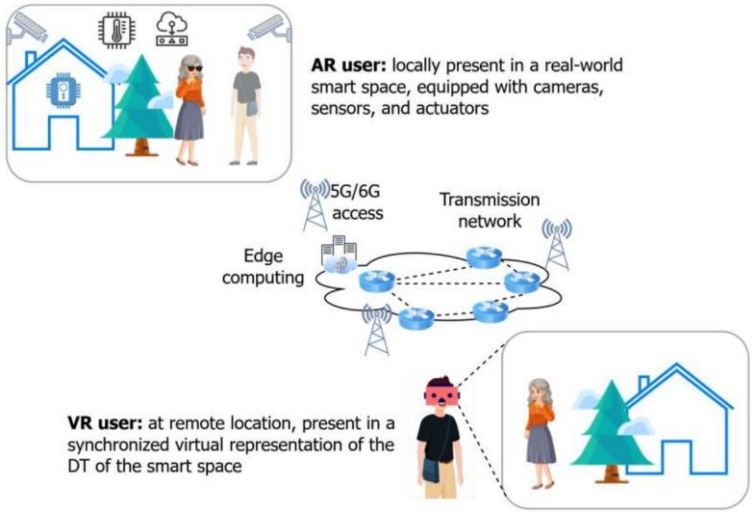

Fig.2. Digital twin scheme linking on-device sensing and interaction to a shared scene.

3.4 Comparison with prior work and limitations

Compared with edge/fog XR studies that report qualitative latency benefits, our evaluation adds fleet-scale evidence with controlled cross-over epochs and optical MTP measurements. The observed improvements (deployment −46%, MTP p99 −24%) are consistent with architecture patterns that stage time-critical work near users and synchronize with the cloud asynchronously. Prior reviews emphasize the need for such partitioning but seldom report p95/p99 and incident metrics across heterogeneous devices and networks, which our results provide. Two limits remain: only three regions and six device types were covered, and we did not test very low-duty-cycle radios (e.g., NB-IoT). Extending coverage and adding long-duration field runs will clarify robustness under seasonal network shifts and extreme bandwidth constraints.

Ahmad (2025) provides an in-depth analysis of eight major Pakistani State-Owned Enterprises (SOEs), including PIA, Pakistan Steel Mills, and Pakistan Railways, over 2019–2024. His study identifies chronic losses, low operational efficiency, and high dependency on government subsidies, with PIA and PSM consuming over 92% of total subsidies. Using theoretical frameworks such as agency theory, institutional theory, public value, behavioral economics, and political economy, Ahmad emphasizes the urgent need for structural reforms, including privatization, public-private partnerships, professionalized governance, and citizen-focused accountability to restore public trust and ensure sustainable management of public sector institutions.

Ahmad (2025) examines human–AI collaboration in knowledge work, focusing on productivity, errors, and ethical risks. Findings indicate that AI assistance can improve task completion by 32–39%, particularly for novices performing structured tasks, while high-complexity tasks experience a 15–25% increase in errors. Errors are

categorized into hallucinated facts, logic problems, fabricated citations, omissions, and biased assumptions. Ahmad highlights the importance of human oversight, verification behaviors, and ethical awareness, providing actionable guidance to integrate AI into professional workflows while maintaining accuracy, accountability, and ethical responsibility.

4. Conclusion

This study tested a service-convergence design for AR/VR systems and found clear gains in speed and stability. Across three regions and six device classes, median deployment latency fell from 128 s to 69 s (−46. 1%), p95 service-call latency fell from 214 ms to 132 ms, and motion-to-photon p99 improved from 58 ms to 44 ms. Rollback events during upgrades dropped by 31%. A shared event format reduced duplicate logs by 41%, cut median time-to-detect from 66 s to 46 s, and lowered distinct incident clusters by 33%. The main contribution is to join a common service layer with rule-based placement across edge and cloud, and to measure its effect using standard tail metrics and optical motion tests. The results show that treating scene, input, and telemetry as services, and placing them close to users when needed, lowers delay and simplifies operations. In practice, the method can shorten rollouts, reduce manual work, and improve runtime quality for multi-device deployments. Limits include three regions, six device types, and a 12-week window; low-duty radios were not tested and advanced trust controls were outside scope. Future work will add more sites and hardware, run longer field trials with seasonal changes, evaluate low-duty wide-area links, and include secure execution and developer tools.

References

Xu, J. (2025). Fuzzy Legal Evaluation in Telehealth via Structured Input and BERT-Based Reasoning.

Yang, Z., Pan, Z., Zhu, X., Zhang, L., Feng, J., Jiang, Y. G., & Torr, P. H. (2024). 4d gaussian splatting: Modeling dynamic scenes with native 4d primitives. arXiv preprint arXiv:2412.20720.

Sun, X., Wei, D., Liu, C., & Wang, T. (2025, June). Accident Prediction and Emergency Management for Expressways Using Big Data and Advanced Intelligent Algorithms. In 2025 IEEE 3rd International Conference on Image Processing and Computer Applications (ICIPCA) (pp. 1925-1929). IEEE.

Maheshwari, S. (2020). Mobile Edge Cloud Architecture for Future Low-Latency Applications (Doctoral dissertation, Rutgers The State University of New Jersey, School of Graduate Studies).

Li, C., Yuan, M., Han, Z., Faircloth, B., Anderson, J. S., King, N., & Stuart-Smith, R. (2022). Smart branching. In Hybrids and Haecceities-Proceedings of the 42nd Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2022 (pp. 90-97). ACADIA.

Friston, S., Fan, C., Doboš, J., Scully, T., & Steed, A. (2017, June). 3DRepo4Unity: Dynamic loading of version controlled 3D assets into the Unity game engine. In Proceedings of the 22nd International Conference on 3D Web Technology (pp. 1-9).

Chen, F., Liang, H., Li, S., Yue, L., & Xu, P. (2025). Design of Domestic Chip Scheduling Architecture for Smart Grid Based on Edge Collaboration.

Haus, M., Waqas, M., Ding, A. Y., Li, Y., Tarkoma, S., & Ott, J. (2017). Security and privacy in device-to-device (D2D) communication: A review. IEEE Communications Surveys & Tutorials, 19(2), 1054-1079.

Wang, B., Linna, G. E. N. G., & Tam, V. W. (2025). Effective carbon responsibility allocation in construction supply chain under the carbon trading policy. Energy, 319, 135059.

Huang, S., Wang, B., Geng, L., & Ma, J. (2025). The pass-through mechanism for carbon emission cost under ETS with different accounting principles: a focus on construction supply chains. Environment, Development and Sustainability, 1-23.

Faruk, O. M. (2024). Advanced Computing Applications in BI Dashboards: Improving Real-Time Decision Support For Global Enterprises. International Journal of Business and Economics Insights, 4(3), 25-60.

Adewale, T. (2024). Identity-Centric Security in Cloud Computing: Safeguarding Workloads with Robust Access Controls.

Wu, C., & Chen, H. (2025). Research on system service convergence architecture for AR/VR system.

Huang, Y., Vu, M., He, W., & Zeng, S. (2025). Rapid Attitude Controller Design Enabled by Flight Data. ASME Letters in Dynamic Systems and Control, 5(2), 021005.

Li, Z., Chowdhury, M., & Bhavsar, P. (2024). Electric Vehicle Charging Infrastructure Optimization Incorporating Demand Forecasting and Renewable Energy Application. World Journal of Innovation and Modern Technology, 7(6).

Li, W., Xu, Y., Zheng, X., Han, S., Wang, J., & Sun, X. (2024, October). Dual advancement of representation learning and clustering for sparse and noisy images. In Proceedings of the 32nd ACM International Conference on Multimedia (pp. 1934-1942).

Häcki, R. (2021). On Performance and Correctness of Intra-Machine Data Transfer (Doctoral dissertation, ETH Zurich).

Ahmad, N. R. (2025). Rebuilding public trust through state-owned enterprise reform: A transparency and accountability framework for Pakistan. International Journal of Business and Economic Affairs, 10(3), 45–68. https://doi.org/10.24088/IJBEA-2025-103004

Ahmad, N. R. (2025). Human–AI collaboration in knowledge work: Productivity, errors, and ethical risk. https://doi.org/10.52152/6q2p9250