

AI-Assisted Emergency Awareness in XRTC

Nuria Oyaga^{1,*}, Pablo Pérez¹, Ignacio Benito¹, Marta Orduna¹ and Álvaro Villegas¹

¹Nokia Spain, Calle de María Tubau 9, Centro Empresarial Bilma, 28050 Madrid, Spain

Abstract

eXtended Reality Team Communication (XRTC) enables a remote expert to be virtually transported into a local scene and to guide people on site under time pressure. This position paper proposes an extension of XRTC for emergency scenarios that combines a remotely controllable PTZ camera, Edge AI perception, temporal agent reasoning, and LLM-based semantic review. Candidate emergency cues are surfaced to the remote expert as decision support: human judgment remains central, with the aim of reducing analysis time. We outline a three-part architecture (local XRTC scene, Edge AI layer, remote expert experience), discuss configurable detection and review policies, and highlight open challenges around false positives, privacy, and expert-facing configuration.

Keywords

extended reality, XRTC, emergency response, Edge AI, human-AI teaming, remote assistance, PTZ camera

1. Introduction

XRTC (eXtended Reality Team Communication) provides a system that, through immersive communication, can virtually transport a person to a specific remote environment [1, 2]. A remote expert can then access the location of a situation within their area of expertise in a more direct and immersive manner, providing assistance and guidance to the people physically present. Although this capability can benefit multiple disciplines, this work focuses on emergency scenarios [3], where an expert can assess the situation, make decisions under time pressure, and guide on-site personnel in responding to the emergency.

This work explores an extension of XRTC [1] that focuses attention on specific areas while using AI assistance to detect potential emergency situations. A PTZ camera connected to the local XRTC environment provides an additional video source that can be controlled by the remote expert to focus on critical points of interest. The video stream is processed at the Edge using AI algorithms [4] to detect potential emergency indicators, such as a person lying on the ground or a body position consistent with a fall [5]. These indications are treated as candidate events to be reviewed and contextualized using an LLM [6], rather than as automated decisions. The resulting information is presented to the remote expert, who can combine it with remote presence at the scene to provide assistance when necessary, in line with broader uses of AI for emergency triage [7].

The main objective is to support human judgment by aiming to reduce analysis time for decision-making and enabling more thorough assistance [8]. By combining immersive presence at the emergency location, focused PTZ observation, AI-based perception, temporal agent reasoning [9], and semantic review, the system can direct the expert's attention towards potentially relevant situations while keeping the expert responsible for the final decision.

The proposed system will be used in the Public Protection and Disaster Relief (PPDR) use-case framework of EURO-3C, which develops federated, interoperable Telco-Edge-Cloud services for a European Connected Collaborative Computing Network (3CN). In this context, the emergency scenario provides an application-level driver for low-latency connectivity, Edge inference, and policy-controlled infrastructure orchestration.

2nd International XR4CE Workshop, November 3, 2026, Valencia, Spain

*Corresponding author.

✉ nuria.oyaga@nokia.com (N. Oyaga); pablo.perez@nokia.com (P. Pérez); ignacio.benito_frontelo@nokia.com (I. Benito); marta.orduna@nokia.com (M. Orduna); alvaro.villegas@nokia.com (A. Villegas)

ORCID 0000-0002-6389-4430 (N. Oyaga); 0000-0002-3502-6791 (P. Pérez); 0000-0001-8775-0733 (I. Benito); 0000-0002-3475-4594 (M. Orduna); 0000-0002-0999-0309 (A. Villegas)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

AI-assisted emergency awareness in XRTC

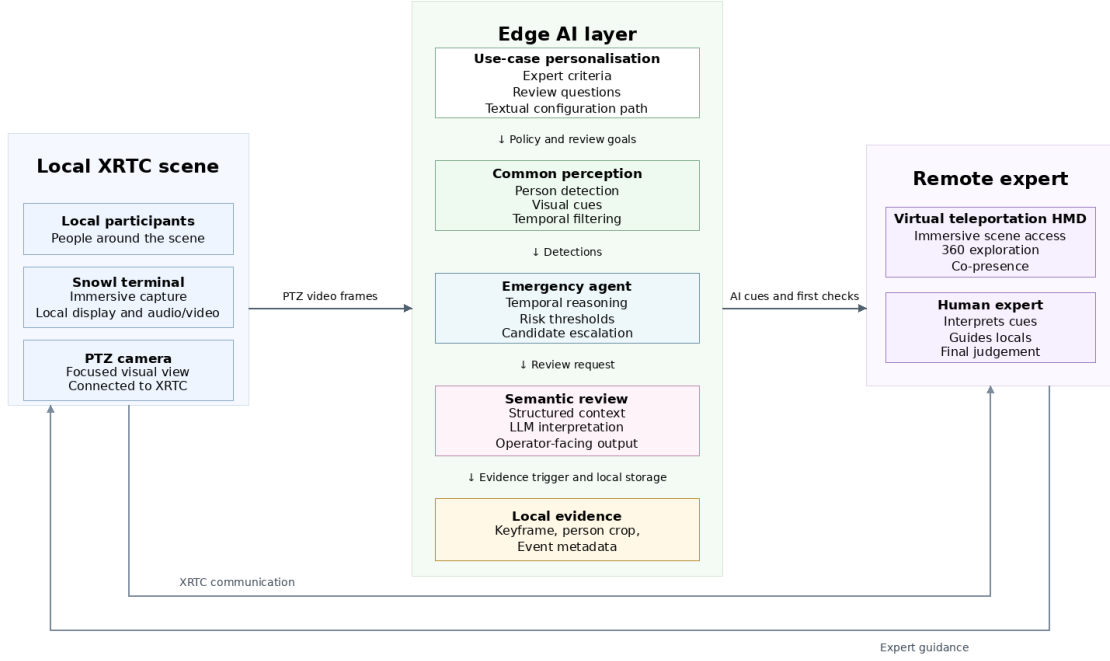


Figure 1: AI-assisted emergency awareness in XRTC architecture.

2. AI-assisted emergency awareness in XRTC

The extension of XRTC with an emergency assistance system that combines AI-based processing with a remote expert who uses the AI-analysis information is structured around a three-part architecture, as shown in Figure 1. The local XRTC scene represents the physical environment in which an emergency may occur and where the XRTC system, including an integrated PTZ camera, is deployed. An AI layer deployed at the Edge consumes the video stream provided by the PTZ camera and performs an AI-based analysis aimed at supporting decision-making and the remote expert during assistance. Finally, on the remote expert side, the expert connects immersively to the scene, analyzes the situation with the AI-analysis information, and provides the required assistance.

2.1. Local XRTC scene: capture, communication, and PTZ observation

The local scene represents the physical environment to be monitored for the detection of potential emergency situations. Within this space, the XRTC system is deployed to capture the physical scene and enable real-time communication between local participants and the remote expert. To support AI-assisted emergency detection, a PTZ camera is integrated into the local XRTC environment as an additional video source.

In the developed system, the PTZ camera video stream is transmitted independently from the other media streams established within XRTC, while relying on the same communication architecture and backend infrastructure. The PTZ video is therefore handled as an additional stream within the XRTC system and can be consumed concurrently by the AI layer deployed at the Edge and by any other authorized consumer connected to the system. This architecture supports both direct visualization of the PTZ video stream and more advanced processing and analysis of the captured scene.

The XRTC media streams, PTZ video, and camera-control messages traverse an L4S-enabled network path, which limits queuing delay and congestion loss under load. The same connectivity and Edge AI pipeline has been evaluated for teleoperated driving [10]. L4S provides the low-latency path, while Edge

execution keeps visual inference close to the video source and reduces the delay between observation and alert generation.

The PTZ camera lets the remote operator control the camera according to a predefined control scheme. Control commands generated at the remote side are transmitted using the same streaming infrastructure employed for the audio and video flows, using a text-based structure. On the local XRTC side, an integrated wrapper interprets these command streams and translates them into the corresponding PTZ camera control operations, including camera movement and zoom.

As a potential extension of the proposed architecture, information generated by the AI processing layer at the Edge may also support automated PTZ camera adjustments. AI-derived information could trigger specific camera operations, such as adjusting the zoom level or repositioning the camera to focus on an area identified as potentially relevant.

2.2. Edge AI layer: perception, agent reasoning, personalisation, and semantic support

The Edge AI layer converts the PTZ video stream into structured alerts and contextual information for the remote expert.

The implementation of this layer is structured into three interconnected components that communicate using the same messaging system and backend infrastructure employed by XRTC: the detection module, the agent module, and the semantic analysis module.

The detection module receives the video stream directly from the PTZ camera and applies an AI-based detection algorithm to identify elements within the scene that may be associated with an emergency situation, primarily people. Algorithms are configurable through a configuration file defined prior to module execution, allowing the user to customize the type of detector to be used and the different thresholds. The current implementation employs YOLO-based detectors [11], although support for additional detection algorithms is considered as a potential extension in future developments. Once the detections have been generated, they are not interpreted independently. Instead, they are forwarded to the agent module, which operates as the operational reasoning layer of the system.

The agent receives the detections and evaluates them over time according to a set of parameters defined in an emergency policy configured in a separate file. These configurable parameters include detection confidence, temporal persistence, body position, posture-related cues, cooldown periods, and additional configurable thresholds. Based on this temporal analysis, the agent determines whether the observed situation should be disregarded or considered a potential emergency. The agent thus transforms low-level visual detections into a structured emergency context that can trigger an alert. This context is subsequently forwarded to the semantic analysis module to obtain a more detailed interpretation of the detected situation.

The semantic analysis module employs an LLM to provide the remote expert with more specific and detailed information about the potential emergency. The amount and type of information delivered to the expert are also configurable through a separate configuration file prior to module execution. This information may include a concise summary of the emergency, a risk priority level, a recommended initial verification step, potential uncertainties, and an explanation of why the observed situation requires attention.

A proposed development direction for the system is to evolve from the current configuration approach, based on multiple independent configuration files, towards a text-based configuration model in which emergency experts can directly describe the situations and conditions that they want the system to monitor. This approach would allow the same perception and reasoning architecture to be reused across different emergency scenarios without requiring substantial technical modifications. The pipeline is cue-agnostic: new indicators can be introduced by pairing alternative detectors (or detector outputs) with updated agent policies and semantic-review prompts, without redesigning the three-module architecture. The current prototype focuses on person presence and fall-related posture; near-term extensions target visual cues such as visible smoke, fire, or flooding, and behavioural indicators including prolonged immobility or distressed crowd movement.

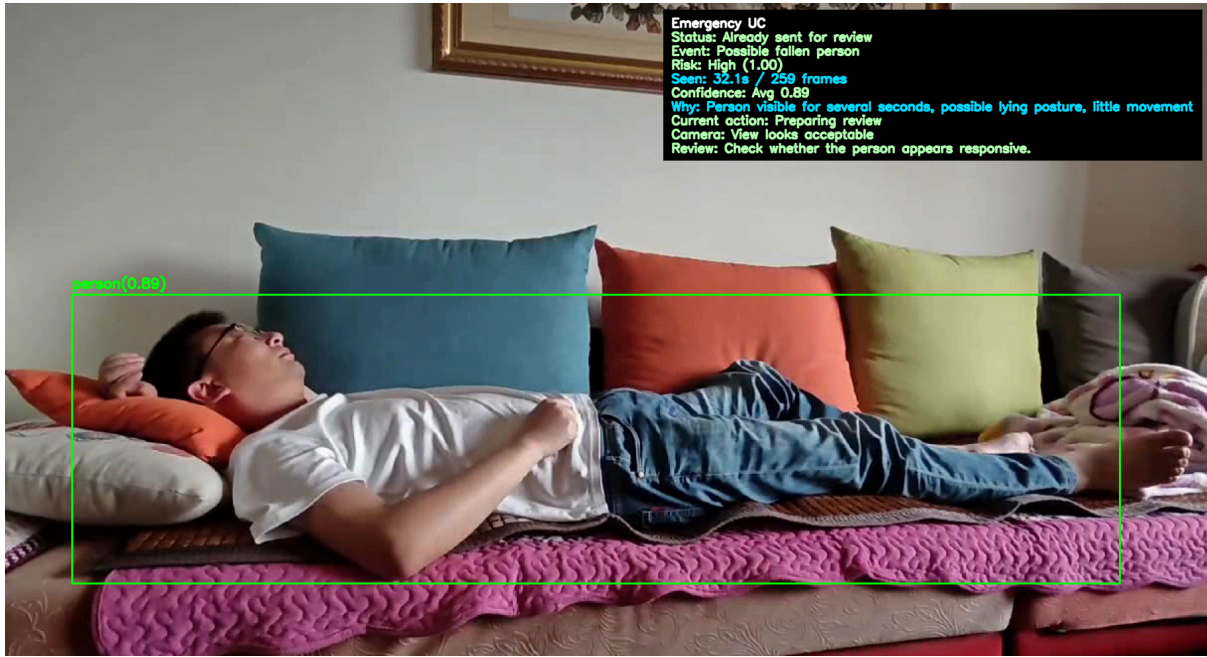


Figure 2: Example visualization of AI cues for the remote expert.

2.3. Remote expert experience: awareness and decision support

On the remote expert side, the expert accesses the scene through an HMD, which provides a sense of presence within the local environment through the XR/TC system, together with a more detailed visualization window supported by the AI layer deployed at the Edge.

The information generated by the AI layer is displayed as an overlay on the analyzed video stream using natural language, while avoiding lower-level computational details, so a faster interpretation by the emergency expert can be performed. In addition to the textual information, bounding boxes are rendered around detected elements that may trigger an alert according to the selected configuration. In the current implementation, this visualization is presented in an additional window within the XR/TC system. However, future integrations may enable AI-generated information to be directly incorporated into the XR/TC interface in the form of alerts, annotations, timeline markers, or contextual panels.

Overall, this approach supports human judgment rather than automating emergency decision-making [8]. The expert remains responsible for assessing the situation, communicating with local participants, and determining the appropriate course of action.

2.4. Initial integration testing

Preliminary integration experiments were conducted in the Nokia Extended Reality laboratory, covering the end-to-end processing chain from PTZ video ingestion through YOLO-based detection, temporal agent evaluation, LLM-based semantic review, and rendering on the expert interface. These sessions involved a remote operator from the development team only; no emergency-domain expert took part at this stage. The operator repositioned the PTZ camera, received candidate alerts comprising visual annotations and natural-language summaries, and assessed them in conjunction with the immersive XR/TC view. The results provide initial evidence that the architectural modules interoperate over the L4S-enabled path and that alert delivery is sufficiently timely to support a future expert-in-the-loop workflow. Qualitative findings included false positives triggered by ambiguous body postures and reduced reliability under adverse viewing angles, in line with the limitations discussed below. Evaluation involving emergency-domain experts is planned separately; quantitative assessment of decision latency and detection accuracy likewise remains for future controlled studies and falls outside the scope of this position paper.

3. Rationale and challenges

Following initial development and testing, several challenges remain to improve system robustness and enable its deployment as an assistance tool in real-world environments. A formal user-based evaluation with emergency-domain experts is planned to assess alert usefulness, false-positive tolerance, and trust calibration under time pressure. One of the main challenges is agile and intuitive use-case configuration, allowing experts to specify in natural language what they want to monitor and how. The system will translate these requirements into the detection criteria and questions required for semantic review and system configuration.

Other challenges include improving detection reliability, handling complex camera angles that limit emergency visibility, and distinguishing real emergencies from ambiguous postures while explaining why a situation was flagged [5, 8]. Privacy is also critical, especially when storing visual evidence [12] of an emergency.

Beyond the current implementation, the agent could use emergency confidence and urgency to request temporary preferential treatment for communication and Edge resources through network-exposure APIs such as CAMARA, subject to operator policy. L4S would maintain low queuing delay, while the network APIs would support policy-controlled resource allocation, linking application state to connectivity and Edge orchestration.

Acknowledgments

This work was funded by the European Union under the Horizon Europe EURO-3C project (EUROpean 3C Networks for Secure Telco-Cloud-Edge), Grant Agreement No. 101297599. Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union. The European Union cannot be held responsible for them.

Declaration on Generative AI

In accordance with the CEUR-WS policy on AI-assisting tools, we declare that generative AI tools (Microsoft Copilot, Cursor) were used for grammar and wording refinement and for literature-search support. No core scientific insights or new sections of text were generated by AI. All authors reviewed the manuscript and remain fully accountable for its accuracy and integrity.

References

- [1] P. Pérez, I. Benito, E. González-Sosa, M. Orduna, Á. Villegas, XRTC: Immersive telepresence for first aid and emergency scenarios, in: CHI'26 Workshop on XR for Challenging Environments (XR4CE), Barcelona, Spain, 2026. URL: https://xr4ce-chi26.tech-experience.at/papers/XR4CE_paper_23.pdf.
- [2] A. Villegas, E. Gonzalez-Sosa, P. Perez, J. Torres, The owl: Virtual teleportation through XR, in: 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), IEEE, 2024, pp. 1158–1159.
- [3] V. Lindström, K. Heikkilä, K. Bohm, M. Castrén, A.-C. Falk, Barriers and opportunities in assessing calls to emergency medical communication centres—a qualitative study, *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* 22 (2014) 61. doi:10.1186/s13049-014-0061-3.
- [4] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, *IEEE Internet of Things Journal* 3 (2016) 637–646. doi:10.1109/JIOT.2016.2579198.
- [5] M. M. Islam, O. Tayan, M. R. Islam, M. S. Islam, S. Nooruddin, M. N. Kabir, M. R. Islam, Deep learning based systems developed for fall detection: A review, *IEEE Access* 8 (2020) 166117–166137. doi:10.1109/ACCESS.2020.3021943.

- [6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, *Advances in Neural Information Processing Systems* 33 (2020) 1877–1901.
- [7] A. Tahernejad, A. Sahebi, A. S. S. Abadi, M. Safari, Application of artificial intelligence in triage in emergencies and disasters: a systematic review, *BMC Public Health* 24 (2024) 3203. doi:10.1186/s12889-024-20447-3.
- [8] J. D. Lee, K. A. See, Trust in automation: Designing for appropriate reliance, *Human Factors* 46 (2004) 50–80. doi:10.1518/hfes.46.1.50_30392.
- [9] M. Abou Ali, F. Dornaika, J. Charafeddine, Agentic AI: a comprehensive survey of architectures, applications, and future directions, *Artificial Intelligence Review* 59 (2025) 11.
- [10] I. Benito, M. Orduna, J. J. Ruiz, J. Gutiérrez, P. Pérez, From bitrate and latency to functional QoE in teleoperated driving, in: *Proceedings of the 7th International Conference on Communications, Signal Processing, and their Applications (ICCSPA 2026)*, 2026, pp. 1–6.
- [11] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. doi:10.1109/CVPR.2016.91.
- [12] J. R. Padilla-López, A. A. Chaaraoui, F. Flórez-Revuelta, Visual privacy protection methods: A survey, *Expert Systems with Applications* 42 (2015) 4177–4195. doi:10.1016/j.eswa.2015.01.041.