CHAPTER 3

Proposed Solution

Good design is obvious. Great design is transparent.
— Joe Sparano

This chapter proposes a user-centered solution for XR-assisted VRU safety, namely the possible interactions with CAVs, guided by usability standards. It first states the concrete needs of pedestrians and cyclists in mixed traffic, then shows how the architecture, interaction endpoints, connectivity layer, and service offloading are designed to meet those needs in real-life scenarios.

3.1 Envisioned Architecture

Figure 3.1 summarizes the set of topics that this thesis brings together for the proposed solution, presenting how they relate to each other. Based on the gaps emerged in chapter 2, this diagram acts as an arranged distribution from the background concepts and state-of-the-art presented in chapter 2 to the design choices presented next: types of cooperative perception given possible sensors on XR devices (see section 3.2), interaction endpoints on XR devices in section 3.3, the connectivity layer of connected VRUs in section 3.4, and service offloading in section 3.5. The figure also provides traceability to the thesis objectives (OBJ1–OBJ6) described in chapter 1, highlighting where each objective is addressed.

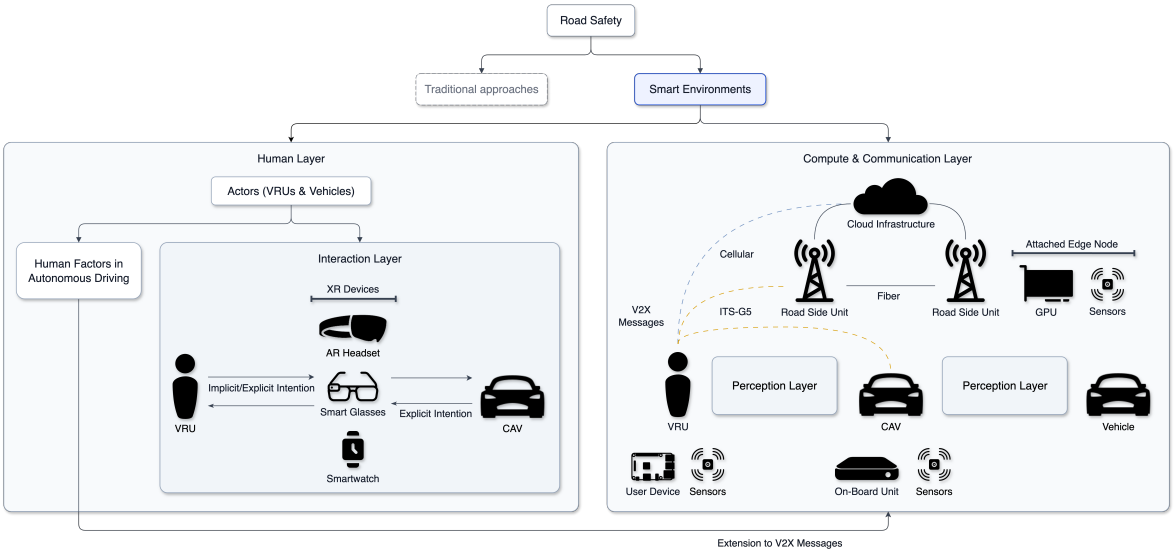

Figure 3.1: The topics tackled in this thesis and their relationships.

The architecture spans from actors to the interaction and computation layer, with explicit data and control planes. The goal is to develop a complete safety system that is technology-agnostic in this section. The structure consists of four main components: VRU perception; human interactions; V2X connectivity; and computation offloading. Each element in Figure 3.1 is unpacked in section 3.2–section 3.5, linking user needs to concrete design choices explicitly.

3.2 Cooperative Perception

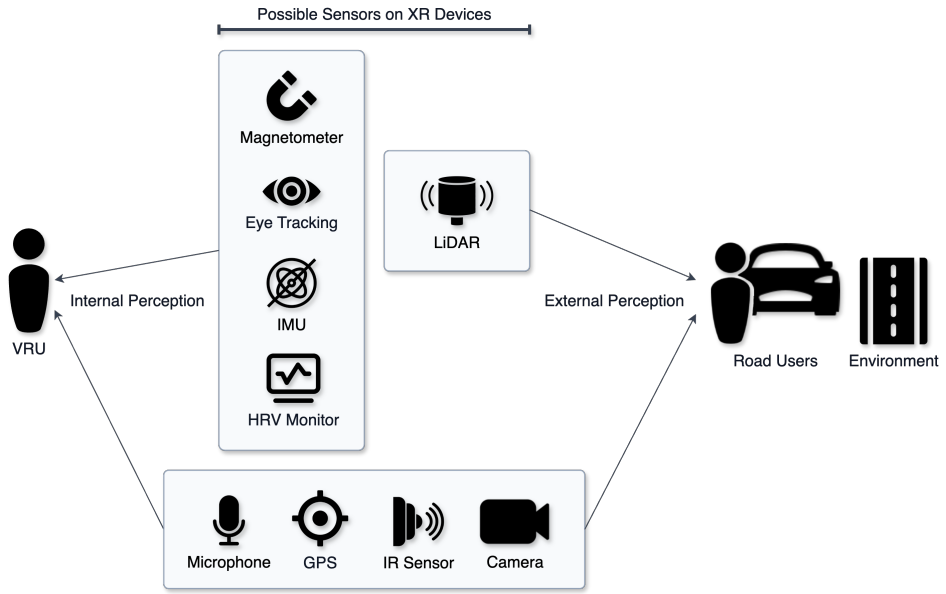

Figure 3.2: Types of sensors available in XR-enabled VRUs and their purposes.

This section implements the users as a sensor opportunity identified in subsection 2.4.3.Figure 3.2 illustrates a cooperative perception system where road users act as mobile sensor units on the road, more specifically, the new notion of leveraging VRUs and their sensors. This system leverages an XR-enabled VRU within a smart city, performing both internal and external perception through an environmental layer representing factors like occlusions, weather, and lighting.

In traditional V2X, vehicles and infrastructure are the primary agents for collective perception (e.g., an AV sharing their sensor data through CPMs). Here, the VRU's XR device is included as a sensor node which captures events from the pedestrian's point of view, which is often unique. For example, a VRU's smart glasses camera might capture an approaching vehicle on a street crossing. The device can send a CPM to alert nearby vehicles of that vehicle's presence (effectively, the pedestrian is helping to detect other vehicles). More commonly, the VRU device will detect and broadcast VRU-self information (as VAM), but augmented with sensor confirmation (e.g., I (the VRU) am approaching a crosswalk and I'm seeing a vehicle on the left). The envisioned use cases focus on immediate safety (short-term perception sharing), so the VRU's device will typically send detections of on-road agents and let the edge system decide if/how to use them.

While RSUs have long-range sensors like Radars, Radars, and high resolution cameras, these are fixed to road infrastructure, and therefore affected by their limited geographical coverage, further hindered by no LoS scenarios. Vehicles also possess these sensors while being mobile; however, they might miss objects because of their relative speed and location to other non-motorized road users. VRUs, being on the ground, have a different perspective as they can often see other pedestrians on sidewalks, or hear things that a sealed car might not. By being mobile, their cameras might capture angles that infrastructure cameras cannot. Thus, it is expected that VRU-provided data to be partial but valuable (e.g., confirming the presence or motion of a vehicle from another angle, or detecting VRUs that vehicles have not yet picked up).

Moreover, the XR devices are meant to run lightweight models that process on-body signals for interaction and safety cues. Core inputs are GPS and IMU for position and heading, eye gaze from the XR device, and hand gestures, which some XR devices are already optimized for their Software Development Kit (SDK) to detect. These signals can usually be inferred in short windows to infer high-level actions. Other metrics that cannot be processed in a timely manner are offloaded (more in section 3.5) using the connectivity backbone of the system.

Figure 3.2 categorizes sensors from internal perception, external perception, and multi-purpose. Sensors like the magnetometer, IMU, Heart Rate Variability (HRV) monitor, and eye tracking only provide information about the person who's wearing the device, while the LiDAR normally delivers information about the surroundings. Infrared Radition (IR)/depth sensors and cameras can be used for both. For instance, detecting an object in LoS with the user, but also recognizing hand gestures. These features depend on the device's number and placement of sensors. Similarly, the microphone may capture the user's voice, but could also detect honking from nearby vehicles. The GPS can directly inform the position of the

user who's carrying the device, and indirectly be used to compute an external road user's coordinates (e.g., combining information from the depth sensor, video camera, and GPS coordinates).

Table 3.1: Representative on-device sensors across the XR device classes considered in this thesis.

Sensor	**AR headset**	**Smart glasses** (voice-only)	**Smart glasses** (voice + HUD)	**Smartwatch**
IMU	✓	✓	✓	✓
Magnetometer	✓	⊖	✓	⊖
Microphone	✓	✓	✓	✓
Camera	✓	—	⊖	—
Depth/IR Sensor	✓	—	—	—
LiDAR	⊖	—	—	—
GPS	⊖	⊖	⊖	⊖
Heart Rate	—	—	—	✓
Eye Tracking	✓	—	—	—

Legend: ✓ = typically present; ⊖ = device dependent; — = not typical.

Table 3.1 describes the possible on-device sensors for the four representative classes of XR devices, using three categories: typically present, device dependent, and not typical. One of the goals is to build a general, device-agnostic system capable of processing the data of all possible sensors across different devices. This abstraction will unify use cases that will depend on this internal or external perception.

While features such as gaze and eye-tracking can serve as input modalities, they also raise privacy concerns for users [121]. To mitigate this, the system performs these sensitive tasks on-device and has opt-in controls with user-visible toggles so that such features are only active with explicit user consent. This approach aligns with recent research and industry practices, which prioritize local processing and user control to preserve privacy in XR systems [122].

3.2.1 Extended Reality Devices

The following subsections describe the characteristics and capabilities of two XR devices used in the current work. The Microsoft HoloLens 2, a AR headset, and a Brilliant Labs Frame, a monocular smart-glass platform aimed at developers.

Microsoft HoloLens 2

The Microsoft HoloLens 2 is a prominent device in the AR category both in research and commercial applications. This is largely due to being the first "complete" AR device with visual see-through technology. Unfortunately, they have recently been discontinued[1] with no evidence of Microsoft following up with a new generation. Nevertheless, these are still heavily used, especially in the research community, as there is no real competitor that can match the HoloLens's capabilities in either hardware or software.

[1] `https://uploadvr.com/microsoft-discontinuing-hololens-2`. Accessed 18 August 2025.

Hardware-wise, the HoloLens is equipped with a Snapdragon 850, which is an ARM-based processor developed for Windows laptops and developed by Qualcomm. It also contains a Graphics Processing Unit (GPU) Adreno 630, and a Holographic Processing Unit (HPU), which is specifically designed to handle data from the headset's sensors, and displays the digital content to the user. The headset also features 4 GB of LPDDR4x memory and 64 GB of storage. It supports Wi-Fi 802.11ac, Bluetooth 5.0, and USB-C for power and connectivity. The HMD has a 1440×936 resolution per eye at 60 Hz, with an FOV of 43° horizontal[11].

Software-wise, the HoloLens has a research mode which allows developers access to raw data from the headset's sensors, including a depth camera, four head tracking cameras, two infrared cameras for eye tracking, an IMU, a magnetometer, microphones, and an RGB camera, among others. Ungureanu *et al.* [123] have demonstrated the potential of the research mode in enabling computer vision research, namely, high-fidelity spatial mapping and real-time object recognition. This mode allows researchers to build mixed reality applications based on processing sensor data and also combine it with the built-in eye and hand tracking capabilities of the headset. The *hl2ss* library by Dibene and Dunn [124] is a popular library that implements the Application Programming Interface (API) described by Microsoft in [123]. This library allows a Python client to access stream sensor data over Transmission Control Protocol (TCP), therefore supporting every OS capable of running Python. Configurable to actively connect to an external host that is waiting for the data, or make the host request the data via the HoloLens' Internet Protocol (IP) address. This feature is crucial for use case scenarios that use cloud services, by not requiring the HoloLens' IP to be public. Additionally, the library can be used as a *dll* plugin for XR applications, or run as a standalone application with the unique purpose of broadcasting sensor data.

Brilliant Labs Frame

The Frame is a lightweight, monocular smart glass. Akin to the HoloLens, it also offers an optical see-through display, but only on the right eye. The display is a 0.23" micro-OLED, with a resolution of 640×400 with an approximately 20° FOV. It is powered by an Field Programmable Gate Array (FPGA) to handle the HMD and the on-board sensors, which include a microphone and an image sensor that captures images at 1280×720, but crops the image in the pipeline to 720×720 for computer vision applications. For motion sensing capabilities, it offers an IMU and a magnetometer. For connectivity, it uses Bluetooth 5.3 as its only way of communication[17]. However, unlike the HoloLens, this is a very lightweight and inconspicuous device, which the normal person may describe as some "quirky" glasses. They are designed to be worn the whole day and help in daily activities.

Software-wise, Frame runs a full open-source Lua-based firmware. The official SDK exposes Bluetooth connectivity between the device and an application host. Therefore, it opens the program application to be developed via Python, Flutter, web technologies, and much more. Another benefit of its open-source nature is that this device allows developers to run their own custom firmware if necessary.

3.2.2 Current Work

Preliminary work in developing the user as a sensor concept has resulted in the publication of a demonstration paper in IEEE Vehicular Networking Conference 2025 [3]. This work showcases a framework that integrates AR headsets and mobile object detection to enhance the safety of road users in a smart city environment. The framework leverages AR devices, such as the Microsoft HoloLens 2, as mobile sensing hubs to capture real-time sensor data. By equipping VRUs with these devices, the system enables dynamic sensing and data contribution, addressing the limitations of fixed infrastructure sensors. The proposed architecture employs microservices to ensure modularity and real-time data processing, meeting the latency requirements for safety-related systems, of under 300 ms. The demonstration features real-world tests, showcasing its potential to improve urban safety and mobility applications. The real demonstration can be visualized in this link[2].

Figure 3.3 presents the demonstration scenario. An AR-equipped cyclist acting as a mobile sensor by using the video and depth camera of the AR headset to detect two road users: an AV and a pedestrian. The headset offloads detections to the smart city marked as yellow markers in the ATCLL map. The AV, already connected to the platform, appears in purple and is overlapped by the perceived vehicle marker in yellow. This indicates that VRU-derived perception data closely matches ground truth data.

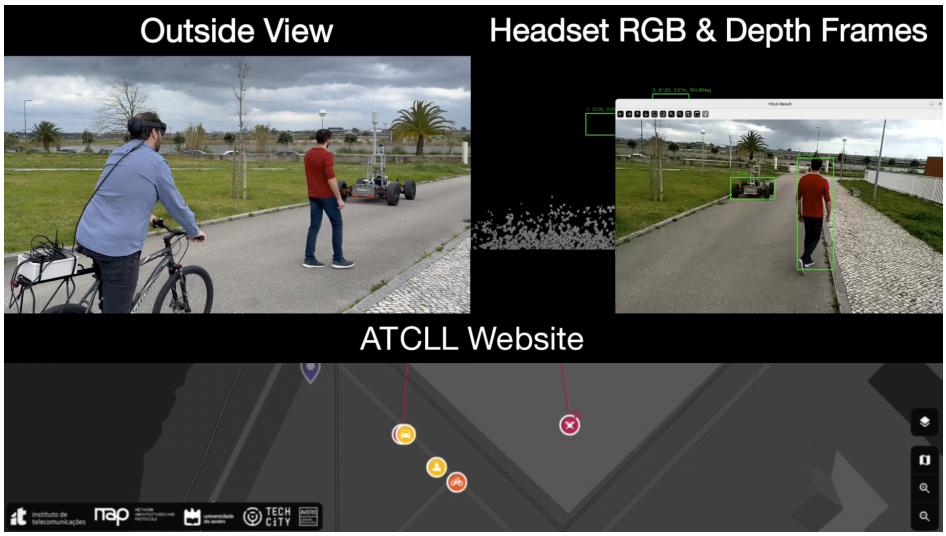

Figure 3.3: User as a sensor demo overview. Left: outside view with an AR-equipped cyclist. Top-right: on-headset RGB and depth frames with object detections. Bottom: ATCLL website showing the detections on the map [3].

As a next step beyond the user as sensor demo, this next work targets the occlusion problem at urban intersections. As urban mobility increasingly integrates micromobility solutions such as bicycles, innovative road safety solutions have become a priority for these VRUs. This work explores the use of AR and V2X communications to enhance cyclist safety at intersections with obstructed visibility.

[2] `https://youtu.be/SQbYsRSChRM`

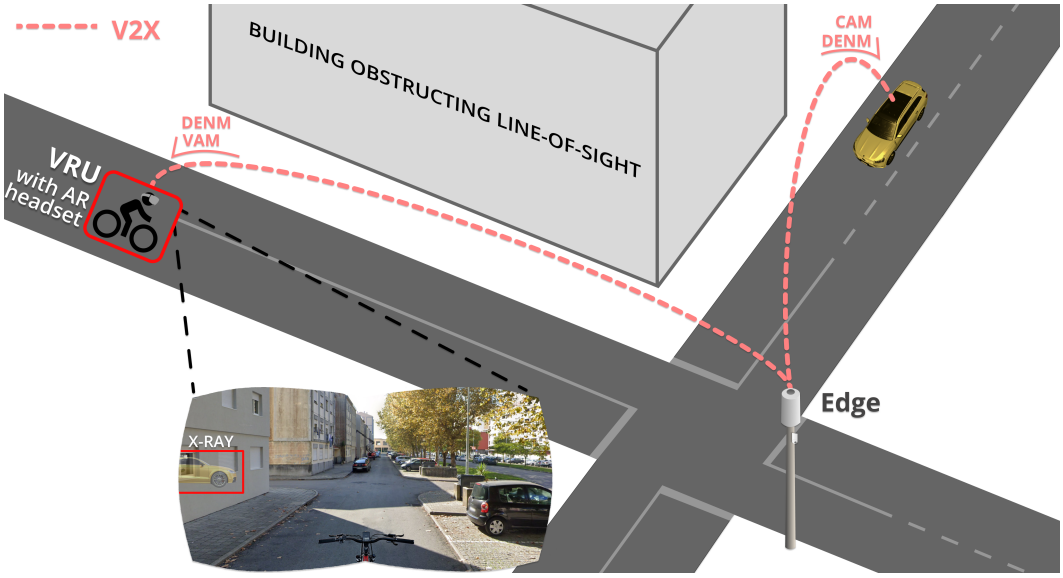

Figure 3.4: A cyclist wearing an AR headset interacting with traffic through vehicular messages to show an obstructed vehicle [2].

Figure 3.4 illustrates a scenario involving road users, their communications, and the UI presented to the VRU via an AR headset to assist with navigation through traffic. This use case addresses situations where cyclists receive assistance at intersections with limited LoS, enabling the visualization of occluded vehicles through an 'X-ray'-like view in the AR headset. This system leverages V2X communications for real-time interaction, and a computational device that processes the AR headset's video camera to intelligently control the 'X-ray' system based on LoS. The system processes CAMs and VAMs to understand the road users' relative position to each other. The video from the AR headset's camera is offloaded to an edge device on the bicycle, which detects external road users. This information can be used to know if the vehicle is out of LoS from the user or not, intelligently turning the 'X-ray' system on or off. The 'X-ray' itself is a virtual asset that superimposes the real vehicle's coordinates received through V2X communications.

To evaluate the system's feasibility, the proposed system is tested in a real-world demonstration in the ATCLL platform using a Microsoft HoloLens 2 AR headset and an NVIDIA Jetson-based object detection pipeline. Results also show that the representation of the 'X-ray' vehicle is done under the 300 ms threshold set by ETSI for road safety applications, the system operates on 3.75 FPS, and that increasing the frame rate of the stream does not impact the resource and power consumption. The real demonstration can be visualized in this link[3].

3.3 Interaction Endpoints

During the autonomous driving transition period, it is anticipated that road traffic will host traditional vehicles (i.e., human-operated, or level 0), connected vehicles, and fully AVs. Therefore, it is crucial that interactions between VRUs and CAVs are seamless, multimodal,

[3] `https://youtu.be/837cFBvnPX4`

and reliable. This section defines the interaction endpoints that enable bidirectional communication between VRUs and CAVs, using cooperative perception from XR-enabled VRUs. The sensors in use have been previously described in section 3.2.

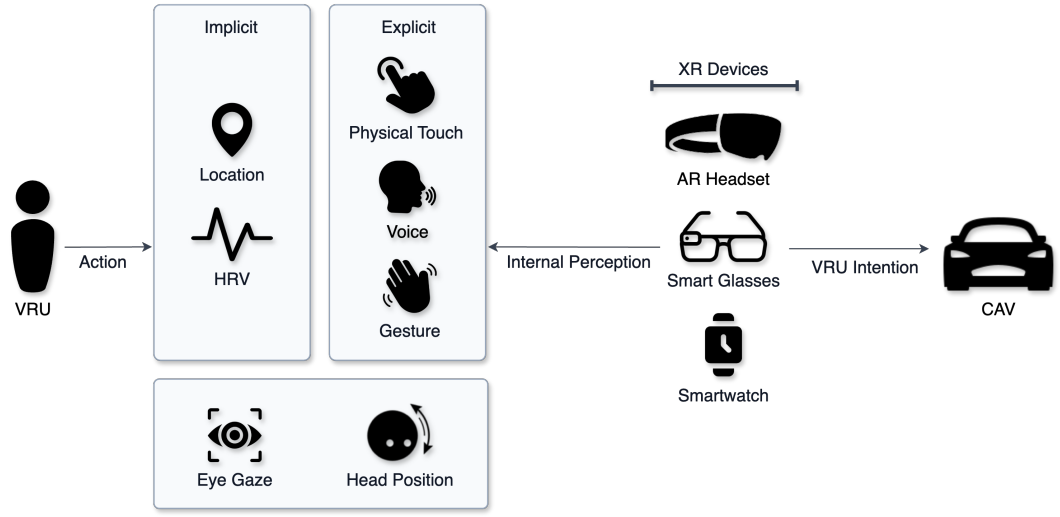

Figure 3.5: Types of input modalities available in XR-enabled VRUs.

Figure 3.5 demonstrates what actions VRUs can perform to trigger a VRU intention. These can be categorized as implicit, explicit, or multipurpose. Implicit interactions include location and HRV. These are classified as indirect because they are transparent and not explicitly thought out by the user. While user location might be used to indicate a VRU's intention (e.g., approaching in the direction of a crosswalk indicates wanting to use it), HRV gives insights into the VRUs's affective state and workload, which can determine how confidently an inferred intention is interpreted. Explicit interactions are those that the VRU needs to purposely perform to convey an intention. These include physical touch, speaking, and gesturing. They can be utilized to initiate intent (e.g., shouting "please stop," or raising a hand to signal the intention to cross the street) but also to acknowledge something (e.g., tapping the device frame or a button to confirm an action). Multipurpose inputs like eye gaze and head position can be both implicit and explicit depending on the context. For example, knowing the VRU's head position in addition to location might indicate the VRU is at a crosswalk looking left. On the other hand, head motion like nodding might acknowledge a prompt given to the user on the HUD.

One crucial component of interaction systems is their usability, and in this case, interactions must be low-effort, unambiguous, and robust to sensing errors more common in outdoor scenarios. In practice, this favors binary or ternary cues (e.g., go, wait, or caution), which are shorter inputs. The aforementioned modalities show both promises and pitfalls; therefore, an interaction system favors the use of multiple interactions to reduce false positives, especially if combining an implicit and explicit input or two explicit inputs. While two explicit interactions create a more robust approach, combining an implicit with an explicit interaction prevents users from performing elaborate actions while increasing the certainty of the VRU's intention.

An example would be prioritizing voice and physical taps in low light scenarios or using gestures and eye gaze in noise scenarios as input. Other strategies may emerge as form of event confirmation or continuous reassurance during the interaction.

Additionally, when multiple wearables are present, cross-device pairing is advantageous. A head-worn device paired with a smartwatch can capture eye gaze, while the smartwatch can be used for physical touch or wrist gestures as explicit actions [125].

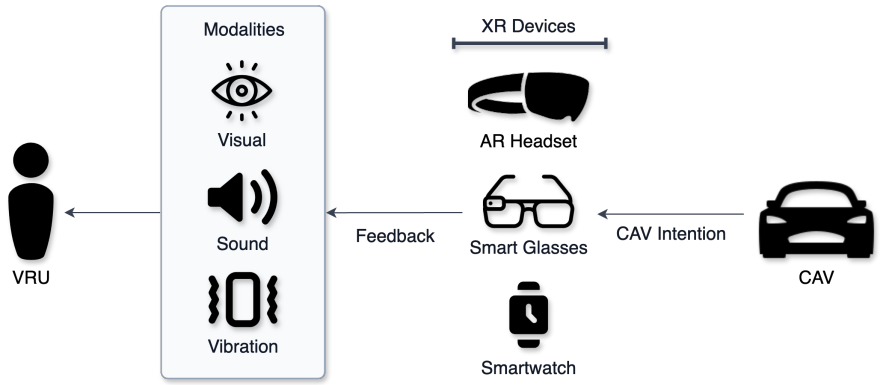

Figure 3.6: Types of output modalities available in XR-enabled VRUs.

In this work, usability follows the International Organization for Standardization (ISO) standard ISO 9241-11 [126] as "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use", in this case, during use. User Experience (UX) follows ISO 9241-210 [127] as a "person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service". This indicates that UX spans before, during, and after use (expectations, feelings during interaction, and lingering impressions).

Figure 3.6 demonstrates output modalities to VRUs wearing XR devices. Following human senses, this can be visual, either 2D or 3D, audio encompassing ear and bone speakers, and haptics. AR serves as one of the most strong output modalities, since human perception shows robust visual dominance in multisensory settings [128], acting like a high-tech traffic sign [73]. Combining at least two output channels for (e.g., brief haptic pulse and audio tone), especially outdoors where either light or noise may be problematic, can provide redundant feedback. Additionally, combining modalities gives clearer feedback; an unambiguous acknowledgment to avoid hesitation or guessing in critical situations. These choices also follow the standard for interaction principles [127], by being intuitive, fitted with user expectations, error tolerant and suitability for different possible tasks.

Table 3.2: Possible implicit/explicit input and output modalities across the XR device classes considered in this thesis.

Modality	**AR headset**	**Smart glasses** (voice-only)	**Smart glasses** (voice + HUD)	**Smartwatch**
Location	⊖	⊖	⊖	⊖
Heart Rate	—	—	—	✓
Voice	✓	✓	✓	✓
Gestures	✓	—	⊖	✓
Physical touch	⊖	✓	✓	✓
Head position/motion	✓	✓	✓	—
Eye gaze	✓	—	—	—
Visual	✓	—	✓(2D only)	✓
Audio	✓	✓	⊖	⊖
Haptics	⊖	—	—	✓

Legend: ✓ = typically present; ⊖ = device dependent; — = not typical.

Table 3.2 describes the possible input and output modalities across the four representative classes of XR devices, using three categories: typically present, device dependent, and not typical. The table is horizontally divided into implicit input modalities, explicit input modalities, multipurpose input modalities, and output modalities. Once more, four representative classes of XR-capable devices are considered: AR headsets, smart glasses (voice-only), smart glasses with HUD, and smartwatches.

Location is device dependent across all categories because on-board GPS is not guaranteed, as this is not a requirement for any of them; many of these devices rely on the connection to a smartphone for this geographic positioning. Heart rate is typically present on smartwatches due to their health tracking abilities. Voice input is commonly present on all devices via integrated microphones, while gestures are tied to those that possess on-device cameras, and smartwatches if worn on the hand that is performing the gesture. They are not typical of voice-only glasses. Head position/motion is only present on head-worn devices and is derived by IMU and magnetometer data. Eye gaze is typically present on high-end AR headsets, which are bulkier and with dedicated eye tracking sensors. Regarding outputs, visual overlays are typically present on AR headsets in 2D or 3D, while smart glasses with an HUD and smartwatches can present 2D images. Smartwatches, however, are screen-based and not heads-up. Audio feedback is typically present on head-worn devices and device-dependent on watches. Haptics are typically present on smartwatches, built for notification alerts.

Focusing on user needs, Table 3.3 does briefly summarizes the mapping the primary user's needs, in this case, VRUs, with the design choices aforementioned and their associated KPIs and KVIs.

Bringing these input and output modalities together creates an interaction loop that enables both VRU-initiated and CAV-initiated interaction. VRUs can issue low-effort inputs combining implicit and explicit interactions, which can be mapped to an application-level intent via V2X messages. After the response from the CAV, VRU guidance is then rendered

Table 3.3: Primary VRU needs and system implications in the proposed architecture.

User needs	KPIs and KVIs	Design implications
Timely interactions	End-to-end communication under 300 ms [103]	Congestion-aware messaging and MEC offload with local fallback.
Unambiguous interactions	High cue recognition accuracy	Binary/ternary cues and application of redundant modalities.
Low effort input	Task time and NASA-TLX evaluation [129]	Multimodal options and cross-device pairing.
Awareness of intentions	Correct intent inference with false positive/negative rates	Message-level intent fields; cooperative perception from wearables.
Privacy	Opt-in features	On-device computation and user-visible toggles.

with redundant outputs continuously (e.g., haptic pulse while the CAV awaits for the user to cross the street). This interaction paradigm is set to enhance VRU safety in the age of autonomy, complementing AV eHMI solutions with a more personalized and human-driven form of interaction. While challenges of reliability and usability remain, combining these modalities with emerging technologies (like XR) may unlock more natural, bidirectional communication with AVs.

Input and output cues (see Figure 3.5 and Figure 3.6) are design to be usable (i.e, clear, quick and intuitive) and to shape the experience across the full interaction communication pipeline — before (anticipation), during (situational awareness), and after (reassurance).

3.3.1 Current Work

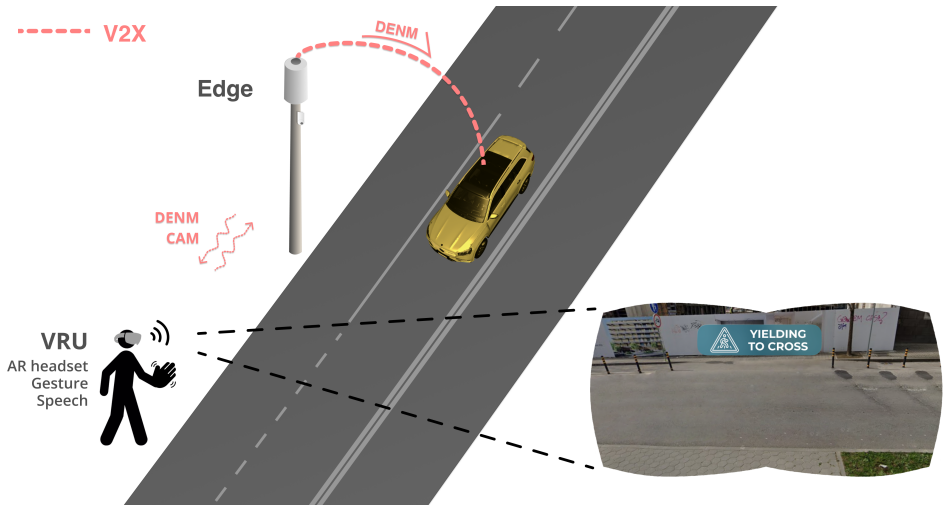

Figure 3.7: Use case scenario showcasing a VRU wearing an AR headset communicating intentions (gestures and speech) to the vehicle through V2X messages.

Building on the input/output taxonomy and the sensors equipped in AR devices described above, a preliminary implementation has been made leveraging the Microsoft HoloLens 2. The headset's microphone array and hand tracking features support two alternative intent channels: voice via wake-word plus a short command, and hand gestures. Each recognized

intent is encoded as an application-level crossing request and sent via a VRU compute device over ITS-G5 to nearby vehicles. Feedback is rendered through an audiovisual cue, consistent with the binary interaction policy. Figure 3.7 presents this use case scenario.

Preliminary measurements in the living-lab show end-to-end latencies above the typical 300 ms safety target; current efforts focus on transport tuning and MEC-assisted ASR for the voice path, and on a lightweight Dynamic Time Warping (DTW)-based recognizer for the gesture recognition. A demonstration video is available at this link[4].

Future work on these systems will report experience, efficiency, acceptability, and workload to evaluate the human-centric side of the interaction. Measures will be collected in outdoor tasks to incorporate real-life environmental factors such as noise and glare.

3.4 Connectivity Layer

This section addresses the challenges associated with VRU-generated messages in road traffic from subsection 2.1.2 and section 2.3.2. The connectivity layer presented in the rightmost side of Figure 3.1 shows all actors involved in the road safety domain (i.e., vehicles, VRUs, RSUs, and cloud infrastructure). The V2X messages are handled by both broadcast and unicast communications over multiple wireless interfaces (i.e., ITS-G5 or 5G). The RSUs act as a multiprotocol gateway, as one can receive ITS-G5 traffic and relay information as a backhaul to the cloud, interacting with cellular traffic. This means that data can take parallel paths. For instance, a warning about an occluded vehicle could be broadcasted through ITS-G5 and 5G, and the VRU will use whichever one that arrives first.

The system builds on the C-ITS framework to represent and share critical information on the road. In particular, VAMs and CAMs for announcing road user location and characteristics, and CPMs for sharing sensor detections. The VRU's user device functions as a personal ITS station, broadcasting both VAMs and CPMs. This user device may be an XR device, wearable, or even a smartphone. All road users may have these perception capabilities, with RSUs and their attached edge nodes leveraging geographical fixed sensing capabilities. In this environment, "smart" road users and infrastructure improve urban mobility, even if the majority of traffic is not connected road users through cooperative perception.

To support advanced interaction, current message standards are set to be extended with custom application-level messages. One key example is a VRU intent message that a pedestrian's device can send. In the envisioned use case, when a pedestrian arrives at a crosswalk and signals intent (either explicitly via user input or implicitly detected by context), an intent message is transmitted to all nearby vehicles. Connected vehicles receiving this request perform an assessment and respond, which then the VRU's device can process and alert the user correctly (e.g., a negative response might be more alerting with audiovisual cues accompanied by haptic feedback). During the crossing, the VRU device continues to send periodic updates like a heartbeat message, with its progress allowing stopped vehicles to monitor if the pedestrian has cleared their lane. Once on the opposite side, the VRU sends a

[4] `https://youtu.be/cYHqxrlKcZg`