Horizon Europe Grant Agreement nº 101135025

Funded by the European Union

A toolset for hyper-realistic and XR-based human-human and human-machine interactions, PRESENCE

Grant Agreement nº 101135025

HE Call identifier: HORIZON-CL4-2023-HUMAN-01-CNECT
Topic: HORIZON-CL4-2023-HUMAN-01-21

Type of action: HORIZON Research and Innovation Actions

D3.1 Haptics technologies report I

DISSEMINATION LEVEL

☒	**PU**	**Public**
	SEN	*Confidential, only for members of the consortium (including the Commission Services)*

Grant Agreement nº:	Project Acronym:	Project title:
101135025	PRESENCE	A toolset for hyper-realistic and XR-based human-human and human-machine interactions

Lead Beneficiary:	Document version:
ACTRO	V1.4

Work package:
WP3 – Digital Touch

Deliverable title:
D3.1: Haptics technologies report I

Start date of the project:	Contractual delivery date:	Actual delivery date:
1st of January 2024	28 of February 2025	31st of March 2025

Editor(s):
Damien Faux (ACTRO)

LIST OF CONTRIBUTORS

PARTNER	CONTRIBUTOR
ACTRO	Damien Faux, Xavier de Tinguy
SG	Gijs den Butter, Max Lammers
INTER	Detjon Brahimaj

LIST OF REVIEWERS

PARTNER	REVIEWER/S
CAPGEMINI	Denis Denisov
I2CAT	Sergi Fernández

CHANGE HISTORY

VERSION	DATE	PARTNERS	DESCRIPTION/COMMENTS
V1.0	12 – 02 - 2025	ACTRO, SG, INTER	Deliverable first consolidated version before internal review
V1.1	25 – 02 – 2025	Capgemini, ACTRO, SG, INTER	Internal review, final consolidated version
V1.2	27 – 02 – 2025	I2CAT	First formatting, indexes and cross-references fixing, version for consolidation, Ethics and Annexes added
V1.3	27 – 03 – 2025	ACTRO, SG, INTER	Internal review, second and final consolidated version
V1.4	31 – 04 – 2025	I2CAT	Final formatting, version submitted through the Participant Portal

Executive summary

The **Digital Touch Technologies Report I (D3.1)** provides an update on PRESENCE Work Package 3 (WP3), which focuses on developing a unified tactile input and haptic feedback technology chain. The current status shows significant progress in developing the **Presence Haptics API**, integrating multiple haptic devices such as the SenseGlove Nova 2, Actronika's Skinetic vest and Meta Quest 3 & Pro controllers. The project has successfully linked haptic content creation tools, developed a transcoding pipeline for haptic data conversion, and established an emitter-receiver body mapping system for improved interaction fidelity. An XR scenario where all three haptic devices can be used within one scene has been created and demonstrated during the general assembly in Januari. However, some key performance indicators, particularly on the multiplayer functionality is still work in progress. Presence is world's first project to implement haptic standardization through the MPEG .hjif format for multiple haptic devices, ensuring interoperability across various haptic devices.

Looking ahead, WP3 will focus on **testing and refining the multiplayer functionality** to support at least six simultaneous users. The transcoding pipeline will be further developed to enable seamless integration between different haptic formats, and additional firmware improvements such as Bluetooth Low Energy communication will enhance real-time rendering. Planned experiments will validate the fidelity of the MPEG. hjif transcoding format, while upcoming publications and pilot studies aim to establish the Presence API as an industry-standard framework for haptic content creation and multi-device interoperability.

Table of contents

List of Figures

1. Introduction

1.1. Purpose, scope and structure of document

The purpose of this deliverable D3.1: Digital Touch Technologies Report I is to provide an overview of the objectives and associated tasks of Work Package (WP) 3. It will outline the required tasks and the roadmap to fulfil them as described in the DoA. This document aims at fulfilling the Milestone 3: Technological pillars 1st delivery and WP3 middle delivery. It provides details of the first iteration of SDK developed within WP3 to allow haptic integration within the interactions of WP5 use cases.

Figure 1: PRESENCE's developer testing the first implementation with both the SenseGlove Nova 2 haptic gloves and the Skinetic haptic vest.

The document is structured as follows:

- *Section 2* provides WP3's objectives, key performance indicators (KPIs), and tasks according to the DoA. In addition, a detailed system architecture will illustrate the components of each task at a higher level, the hardware required, and the connection between the tasks.
- *Section 3* presents a review of the related works in the industry and scientific community.
- *Section 4* elaborates on the i) developments and key achievements, ii) KPIs status, and if applicable, iii) deviations and mitigation plan of each task within WP3.
- *Section 5* presents a first evaluation of the proposed approaches through the prism of industry standards.
- *Section 6* provides an outlook and planned experiments and publications within WP3.

2. WP3 - Digital Touch

2.1. Objectives and KPIs

This WP focuses on the development of a tactile input and haptic feedback end-to-end technology chain. This WP aims to achieve several objectives:

- Facilitate real-life tactile interactions between humans and objects through the use of haptic technologies,
- Enhance the current state-of-the-art devices,
- Develop a new abstracted haptic data processing flow that goes beyond the limitations of the haptic devices being used,
- Enable haptic and tactile interactions between remote users,
- Create a set of technologies and libraries that will support integration WPs for user-oriented testing and the implementation of demonstrators.

These objectives are supported by three KPIs to ensure the progress of the works towards their aims:

- **KPI 3.1**: Deliver a tactile input system with higher-level haptic pattern composition using open APIs for, at least, six (6) simultaneous users ready to control and interface in a real-time XR scenario (WP3, T3.1, T3.2, D3.2).
- **KPI 3.** : Improve the current state of the art for haptic engines and providers delivering open APIs for at least 6 (six) simultaneous users and ≥3 (three) haptic modalities (as combinations of active, passive, tactile, kinaesthetic) in a real-time scenario (WP3, T3.3, D3.2).
- **KPI 3.3**: Deliver one set of haptic technologies APIs to integrate haptics with the two other pillars, in the two (2) demonstrators and with additional scenarios (WP5, T5.1, D5.3).

2.2. Tasks

This WP is divided into three tasks, driven in parallel:

- **T3.1 - Tracking and tactile capturing** (M04-M32, Lead: SG, Contr.: ACTRO, INTER, UB):

  Task T3.1 will focus on processing the set of inputs needed to enable tactile/haptics interactions.

  The work will implement i) capturing methods for tactile actions on digital objects; ii) volume recognition and visual tracking methods for human actions and linking this to existing pre-programmed haptic data.

  The work developed within this task will be fed into T3.2 (interpretation and transmission of the inputs collected) and T3.3 (haptic rendering of the inputs to the users).

  In addition, this task will make use of the human representation developed in WP2 (T2.2) and WP4 (T4.1), linking the haptic input collection to volumetric video and also to 3D human models, providing an integrated coherent multisensory capture solution of both visual and tactile data for multi haptic perceptions.

- **T3.2 - Interpretation and transmission of tactile data** (M04-M32, Lead: INTER, Contributors: ACTRO, SG):

  When combined with point-clouds, visual imaging or audio, the tactile data (from T3.1) is enriched with properties such as shapes, texture, velocity.

  This task will focus on i) interpreting the combined data for a deep understanding of the captured properties and the correlation between social cues and haptic rendering strategies, ii) translate the interpretation into a higher level of abstraction for the Interhaptics Engine and Providers component (T3.3) and iii) exploring novel transmission formats of any haptic and tactile data between different components and the network.

  Furthermore, the task will evaluate different transmission architectures based on KPIs like latency, scalability and resource usage.

- **T3.3 - Rendering of haptic data** (M04-M32, Lead: ACTRO, Contributors: SG, INTER, UB, IMEC):

  The main focus of this task is on the rendering of tactile data (captured in T3.1 and interpreted and transmitted in T3.2) utilising tactile displays (as gloves, vests and controllers). The goal is to improve haptic fidelity in order to enhance the perceived presence in social interactions.

  This will be done by developing a software component that is able to translate individual haptic patterns toward different displays and enable multi-device rendering transcending device-specific limitations.

  Further a set of software tools, providing haptic design capabilities to manage specific social haptic interaction patterns and the rendering for multi-device and multi user environments, will be designed and developed to facilitate integration with XR applications (WP1) and demonstrators (WP5).

2.3. Architecture

The PRESENCE Haptics API consists of 3 layers - the Materials & Body Layer, the Core Haptic API, the underlying Device Specific Implementation - and is supplemented by several Haptic Creation Tools.

The Haptic Creation tools consist of the Interhaptics Composer and the Skinetic Studio, which provide several means to generate the haptic sensations. The generated files store the sensations either as signals or parametric descriptors. The Interhaptics Composer and Skinetic Studio can handle. hap, .hjif and .wav files which can then be imported into the client’s project as haptic materials which are models for the rendering of haptic effects.

The Core API is the main entry point of the haptic API and allows it to perform basic operations such as loading haptic materials, playing and stopping effects. Playing a haptic effect consists of defining which effect to play, and where to play it. Developers using the Presence Haptics API will mainly interface with this layer.

Additionally, the core automatically handles the various haptic files and performs conversion between them. That Transcoding is automatically performed as the files are imported within Unity Editor.

The Device Layer contains the individual implementations of the Core API for specific haptic devices. When an effect is played through the Core API, these implementations take their own version of the signal, and play it through their proprietary API when applicable. For instance, when the SenseGlove implementation is asked to play a vibration on the hand palm, it retrieves the “SenseGlove version” of that signal from said effect and plays it on the Nova Glove using the SenseGlove API. Each implementation must also include the conversion from said device’s signals into a more generic format and vice versa.

The “Materials & Bodies Layer” is a combination of components that are used to send and receive haptic effects based on collision events. HapticReceivers are linked to a body part, and are meant to be added to a player’s Avatar. HapticEmitters can be added to colliders that play one or more haptic effects on a body part when touched; for example; a table or a cup.

All parts of the API are unified under the “Presence” namespace, and all classes related to the Presence Haptics API are prefixed with “PHap_,” except for scripts created by device implementers.

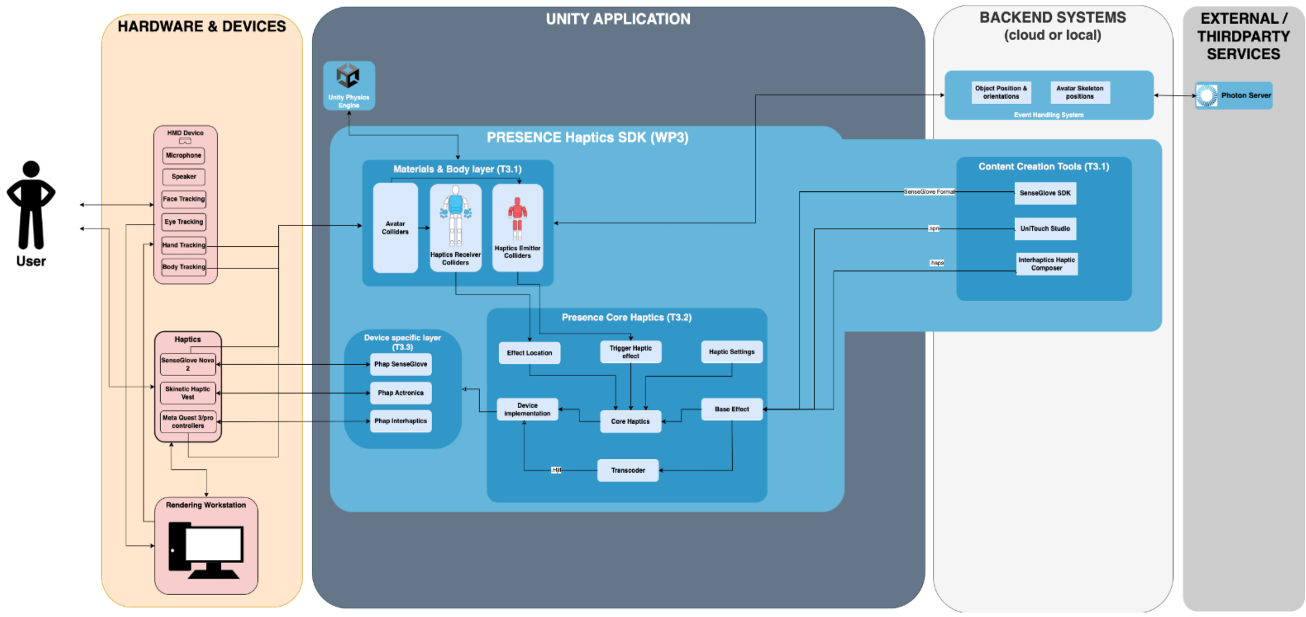


Figure 2: Architecture DIagram in the same style as other PRESENCE work packages of WP3

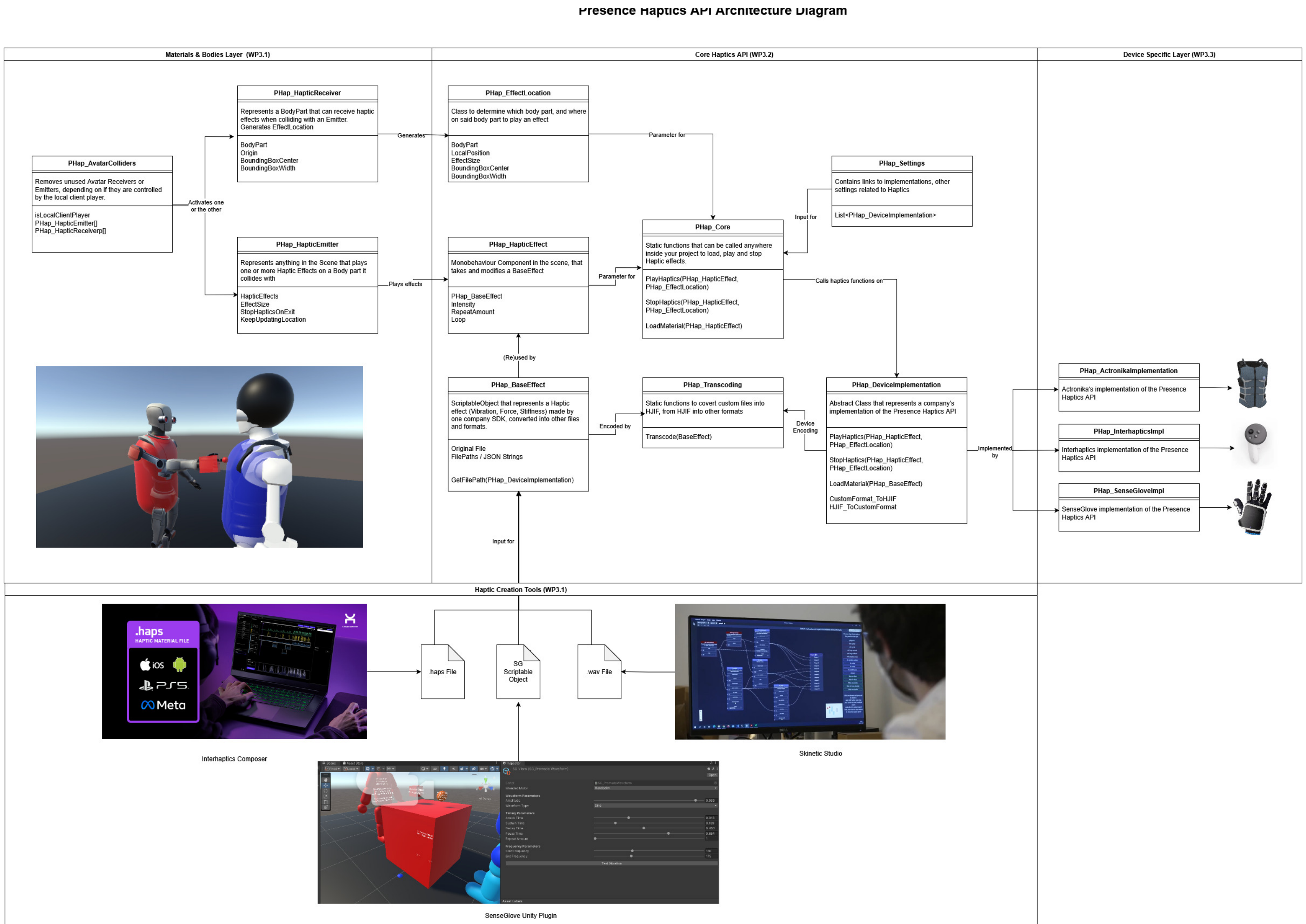


Figure 3: UML diagram of the PRESENCE Haptics API.

3. Related work

3.1. Haptics State of the Art

Within PRESENCE, we have selected the SenseGlove Nova 2, Actronika's Skinetic vest, and Meta Quest controllers as the primary haptic devices due to their superior fidelity, ease of use, and suitability for multiplayer industry applications. These devices offer a balance between wearability, precision, and user experience, making them ideal for enterprise use-cases and training scenarios.

The SenseGlove Nova 2 represents a state-of-the-art haptic glove designed to provide a high level of tactile realism. It integrates force feedback through a magnetic friction brake system, capable of exerting up to 20N of resistance per finger, allowing users to perceive object stiffness and sizes in virtual environments. Additionally, it employs vibrotactile feedback via linear resonance actuators (LRA's) to simulate interactions such as button clicks and surface textures. The glove's active strap mechanism ensures a secure fit while enhancing the perception of object manipulation, contributing to its high wearability for extended use in training and simulation applications.

When compared to alternative haptic gloves, the Nova 2 demonstrates superior wearability and versatility. Manus VR gloves, while known for precise hand tracking, primarily rely on vibration motors for feedback, lacking force feedback capabilities. Weart gloves provide tactile feedback with a focus on material perception but do not incorporate force feedback mechanisms. HaptX gloves, in contrast, offer highly detailed haptic interactions with an exoskeleton design that delivers up to 40 pounds of force per hand. However, the bulkier form factor and reliance on an external pneumatic system limit their usability in a mobile, multiplayer XR setting. The Nova 2 balances haptic fidelity and practicality, making it well-suited for industrial collaboration and training scenarios.

Figure 4: SenseGlove Nova 2, force and haptic feedback glove used within PRESENCE.

For whole-body haptic feedback, the Actronika Skinetic vest was chosen over competing solutions due to its high-definition (HD) haptics and superior fidelity. Unlike the bHaptics vest, which relies on eccentric rotating mass (ERM) motors to generate vibrations, the Skinetic vest incorporates 20 high-resolution voice coil actuators. These actuators enable a wider range of sensations, from subtle environmental effects like raindrops to strong impact simulations, resulting in a more immersive experience. Additionally, the vest supports wireless connectivity, long battery life, and lightweight ergonomics, making it an ideal choice for dynamic, multiplayer use cases.

Figure 5: Actronica Skinetic HD haptic vest, used within PRESENCE

Electro-tactile haptic gloves and vests were considered but ultimately excluded from the PRESENCE project due to concerns regarding fidelity, safety, and ethical considerations. Electro-tactile feedback, which stimulates the skin using electrical signals, often struggles with consistency and precision in generating realistic sensations. Furthermore, the potential for unintended muscle contractions or discomfort raises safety concerns, particularly in long-duration use. From an ethical standpoint, ensuring user comfort and mitigating any potential adverse effects are paramount, making mechanical and vibrotactile-based haptic systems the preferred choice.

The decision to prioritize wearable haptic solutions in a multiplayer XR setting was driven by the need for mobility, scalability, and natural user interaction. Wearable devices such as the SenseGlove Nova 2 and Skinetic vest allow users to move freely without being tethered to stationary feedback systems, enabling a seamless collaborative experience. Their wireless nature ensures ease of deployment across multiple users, making them particularly well-suited for enterprise training, remote collaboration, and simulation environments.

3.2. Foundation of haptics standardization

The collaboration in WP3 traces its origins to the Haptics Industry Forum, where the founders of Actronica, Interhaptics, and SenseGlove identified a significant gap in knowledge among XR developers and project managers regarding the effective implementation of haptics in XR scenarios. To address this, they authored *XR Haptics: Implementation & Design Guidelines*, providing a foundational reference for best practices. Building upon this effort, the need for standardization emerged, particularly in haptic file formats and XR scenario implementations. This collaboration played a key role in shaping the MPEG haptics file format, culminating in WP3's first cross-device reference implementation. Additionally, the partners contributed to efforts toward an OpenXR haptics standard, actively engaging as an advisory panel to Khronos on the development of an OpenXR extension for haptic feedback devices. Their collective work continues to drive advancements in standardization and interoperability within the XR haptics ecosystem.

4. Status

4.1. Tracking and tactile capturing

The primary objective of T3.1 is to deliver a tactile input system capable of higher-level haptic pattern composition using open APIs, supporting at least six simultaneous users in real-time XR scenarios. To achieve this, the work package focuses on two main areas: the haptic content creation tools and the materials and bodies layer.

Haptic Content Creation tools:

Currently, there are no generic haptic content creation tool sets, as most tools are device-specific. In this project, Presence aims to bridge this gap by enabling these tools to transcode haptic signals into the MPEG .hjif file format. While the actual transcoding is addressed in T3.2, T3.1 focuses on its integration, including unifying the taxonomy of haptic modalities. Additionally, this task involves enhancing the haptic content creation tools developed by project partners, such as Actronica's Unitouch Studio, Interhaptics' Haptic Composer, and SenseGlove's Interaction SDK. As the project progresses, a subset of pre-defined materials will be developed for use within the Presence API, potentially including captured materials to enrich the toolset.

Materials and bodies layer:

To ensure seamless content creation and device interoperability, developers must define the body part for which a haptic effect is designed and the triggering events. Presence addresses this by creating a bodymapping system that links haptic effects to avatar systems. It also includes the development of a haptic emitter and receiver system for Unity. This system leverages Unity's physics engine to trigger integrated haptic effects, where haptic emitters—assigned to virtual objects or body parts—carry a haptic signal, effectively creating virtual materials. Haptic receivers are associated with haptic devices; when they collide with an emitter, they interpret the signal, check for connected haptic devices on the specified body part, and trigger the appropriate API event to play the received haptic signal.

4.1.1. Development keys and achievements

Haptic Content Creation tools:

Haptic content creation tools are software used to design and integrate touch sensations into digital environments. Unitouch Studio / Skinetic studio (Actronika) offers templates and AI-assisted tools for creating haptic sequences, while the Interhaptics Composer provides scalable tools for rendering complex haptic effects. Additionally, the SenseGlove Interaction SDK enables precise hand tracking, force feedback, contact feedback and vibrotactile feedback integration, allowing to build rich haptic interactions. All three tools are essential to the PRESENCE Haptics API, enabling consistent, multi-device haptic experiences.

The key advancements in this task involved linking three distinct content creation tools to the transcoding pipeline developed in T3.2. Actronika introduced a preliminary version of the Unitouch Studio, Interhaptics integrated their Haptic Composer and, and SenseGlove established integration with their Unity Interaction SDK. As a result, three different haptic modalities (KPI 3.2)—vibrations, stiffness, and contact feedback—can now be effectively encoded into the MPEG .hjif format and linked to the haptic emitter system in order to create a unified “virtual material”.

Next to the integration of the three different tools, the tools have been improved:

Actronika has improved several aspects of the Unitouch Studio tool for haptic sensations design. Templates of haptic sequences for various types of interactions contexts, and associated tutorials, have been added to ease the design approach and simplify the learning of the different tools. In the same spirit, the possibility to export the designed sequence in an audio format has been added to allow the user to use their design with a wide range of devices. Pushing further this idea of simplifying the haptic design steps for the developers, an AI tool to automatically generate a haptic sequence from a context prompt has been implemented, in an early development stage. The overall user interface has also been improved to enhance the user experience through the intuitiveness of use of the different tools. Finally, an audio-to-haptics tool has also been released to provide a seamless/Plug&Play haptic experience, without any prior haptic design requirements.

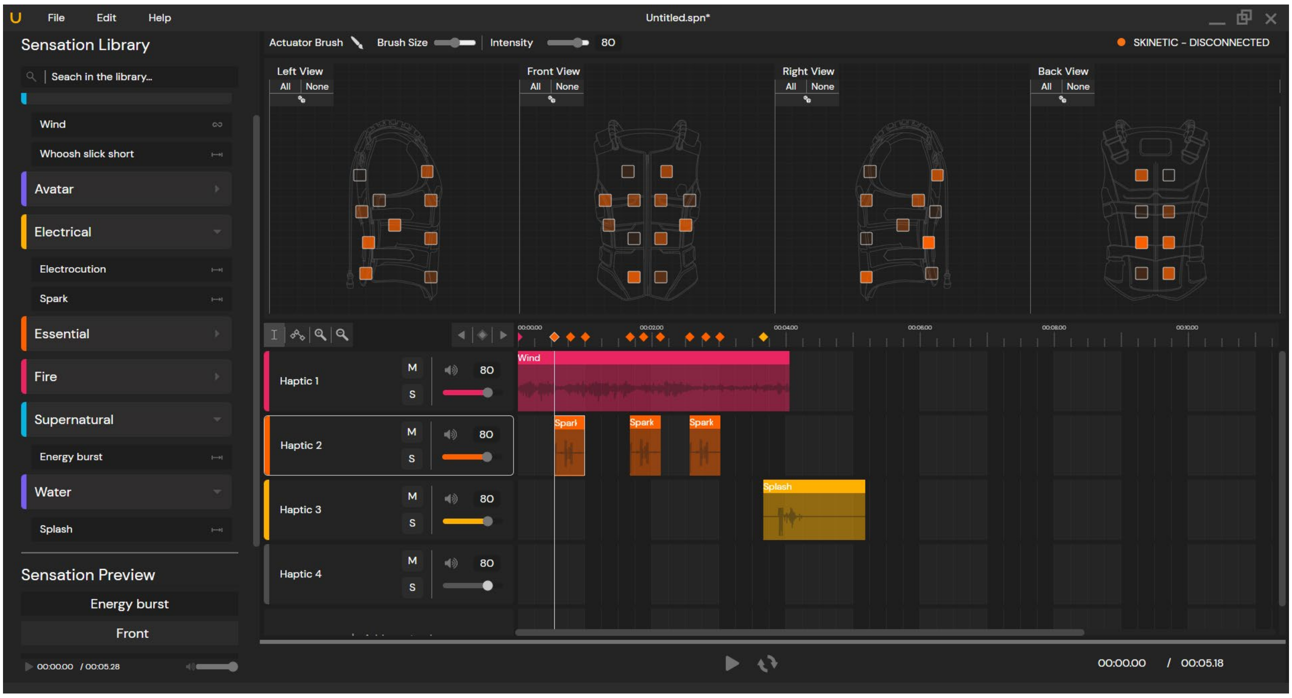


Figure 6: Interface overview of Unitouch studio from Actronica.

Interhaptics has made significant improvements on the Haptic composer development across multiple fronts. Optimization efforts have enhanced the overall performance, ensuring smoother and more efficient operations. The graphics have been upgraded to provide a more visually appealing and immersive experience, while improvements in UX have made the software more intuitive and user-friendly. The general speed of use has been notably increased, allowing for faster workflows and reduced latency, bolstering the software's stability and scalability. Additionally, the introduction of new rendering capabilities has expanded the creative possibilities for users. At the current stage we are performing QA validation prior to release to the public.

Figure 7: Interface overview of the Interhaptics composer.

SenseGlove has made significant improvements to their Interaction SDK, which now supports the use of OpenXR tracking devices within the SDK. Additionally, new features have been implemented, including force- feedback based on the SenseGlove Nova 2 active strap, custom waveforms for high definition vibrotactile-feedback, and optimized methods for defining stiffness curves.

Materials and bodies layer:

The collision and body mapping system has been successfully integrated into the Presence Haptics API. It has been tested and demonstrated through a VR scene showcasing the API's capabilities, including the body mapping and emitter-receiver collision system. Initial optimizations have also been implemented, addressing variations in avatar sizing and supporting multiple actuator localizations.

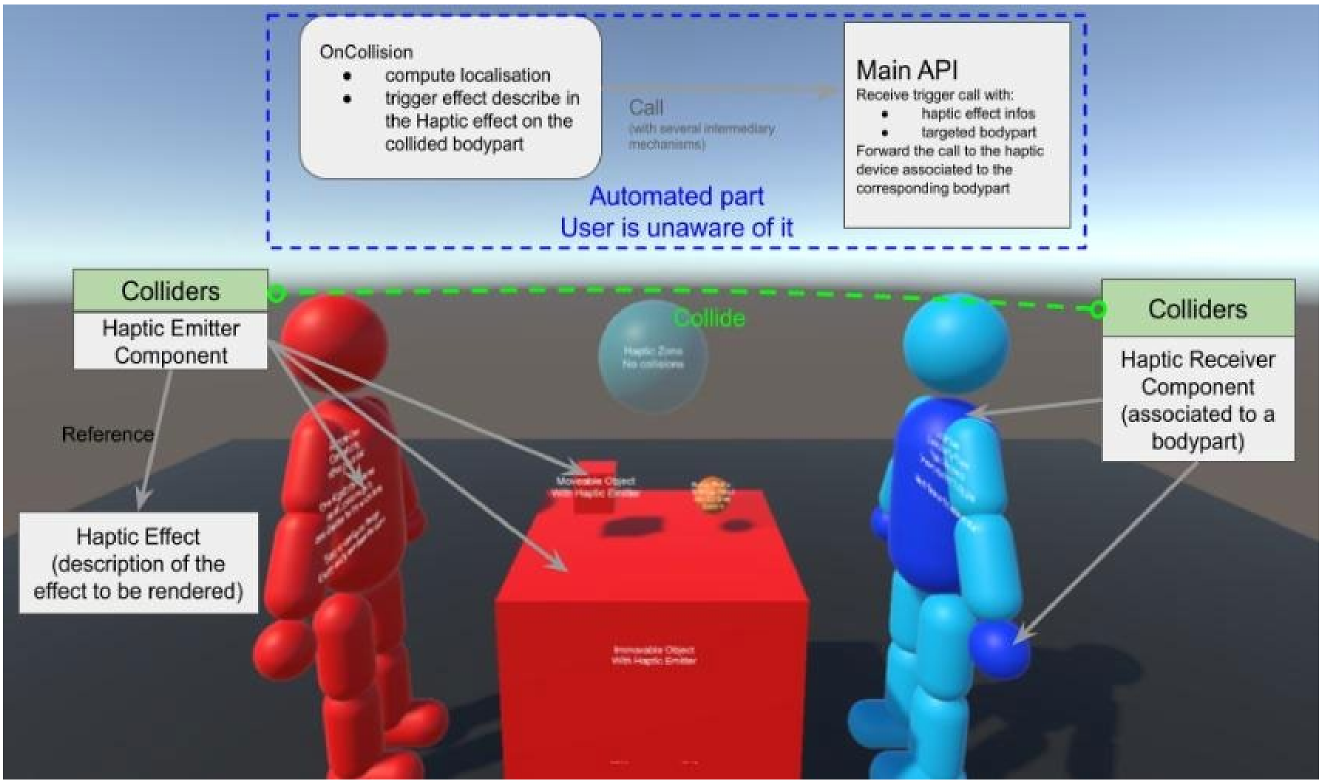


Figure 8: Haptic Emitter, Receiver & Bodymapping system of Presence.

To accommodate the directionality and spatiality of haptic effects on body parts with multiple actuators, the system takes target and radius as input information for rendering. This allows each effect to be assigned a specific location, ensuring that if a device has actuators within the defined target and radius, it can accurately render the effect. By using this spatial information, the system enhances precision and immersion in haptic feedback. The implementation and rendering of these effects are part of Task 3.3.

4.1.2. KPIs status

KPI 3.1: *Deliver a* ***tactile input system*** *with higher-level haptic pattern composition using* ***open APIs*** *for, at least,* ***six (6) simultaneous users*** *ready to control and interface in a real-time XR scenario (WP3, T3.1, T3.2, D3.2).*

The current implementation of KPI 3.1, which aims to deliver a tactile input system supporting at least six simultaneous users in a real-time XR scenario, has not yet been fully tested with six users. However, there is strong confidence that this KPI can be successfully met with the existing system architecture. The primary reason for this confidence is that all haptic rendering is performed on the client-side, which means that the number of users present in the scene does not impact the rendering process itself. From the perspective of the haptic system, there is no difference between two users or six users interacting within the same environment.

The client-side rendering approach ensures that haptic signals are processed and rendered locally on each user's device. As a result, the only interaction data transmitted over the network relates to position and collision events rather than the haptic signals themselves, which minimizes network overhead and latency. This architecture is inherently scalable, allowing the haptic system to maintain high performance and responsiveness regardless of the number of users.

Additionally, the use of the Presence Haptics API further supports this scalability by providing a unified framework for managing haptic interactions independent of the number of participants. Therefore, while the formal testing of six users is still pending, there is a clear technical basis for believing that this KPI will be achieved once full multiplayer testing is conducted. Also, the transcoding of haptic effects is done at creation of the scenario and is not executed live.

KPI 3.2: *Improve the current state of the art for **haptic engines and providers** delivering **open APIs** for at least **6 (six) simultaneous users** and **≥3 (three) haptic modalities** (as combinations of active, passive, tactile, kinaesthetic) in a real-time scenario (WP3, T3.3, D3.2).*

This is a shared KPI between task T3.1 and T3.2. Three haptic modalities are functionally implemented into the transcoding pipeline. (Vibration, stiffness and force). It is now tested with one user, the multiplayer functionalities are due for the next iteration and the creation of template scene as an example integration for WP3, that is on the scheduled tasks for the next phase.

4.1.3. Deviations and mitigation plan

Deviations:

Within the architecture of Presence, there is no provision for tactile capturing. This decision is the consequence of the overarching decision that holo-ported humans will be captured, and if these individuals are not wearing haptic displays, they will appear as non-touchable holograms. As no objects are being captured, there is neither a reason nor a use case to capture live tactile data.

Consequently, T3.1 has shifted its focus toward pre-set haptic materials, as well as the development of the haptic content pipeline and associated tools.

As described in the DoA developments i and ii of T3.1, i) *“capturing methods for tactile actions on digital objects;* and ii) *“volume recognition and visual tracking methods for human actions and linking this to existing pre-programmed haptic data”.* will be replaced by: "Enhancing and linking existing content creation tools for haptics to a generic haptics file format, enabling content creators to develop for any device using any tool."

Mitigation plan:

Within Presence WP3 the three project partner all have their own tactile content creation tool. These tools will be implemented and enhanced in the presented architecture, allowing for ease of integration of haptic devices without live tactile capturing. In order to validate the deviation mentioned above and test the set KPI’s a template scene will be created using the Presence Haptic API in a multiplayer setting. In that way also the 6 players objective of the KPI can be tested that was not feasible in the first version. With the chosen implementation of the Emitter/Receiver body mapping system we are able to handle all haptic calls on the client side. Only the event trigger should be derived from the server based position of the Emitters & Receivers. Since this is normal position and collision data we do not expect any overhead from the (transcoded) haptic signals. Also different effects from different content creation tools will be loaded into the template scene.

Identified Risks:

To utilize the SenseGlove Nova 2 effectively, a manipulation system is required to detect actions such as grabbing and releasing (haptic) objects. Ideally, the Presence Haptics API should remain independent of the manipulation system being used. To achieve this, SenseGlove has begun implementing OpenXR hand tracking formats. While the implementation of these formats into their manipulation system has been successful, full recognition as an OpenXR device requires the creation of an OpenXR subsystem. This process involves significant time and costs, including those associated with Khronos membership. For now, this risk is mitigated by using the SenseGlove manipulation system with the Presence Haptics API for projects involving SenseGlove. Currently, manipulation can be achieved with both SenseGloves and controllers. However, for future implementations, the goal is to enable compatibility with any third-party manipulation system that supports OpenXR tracking formats.

4.2. Interpretation and transmission of tactile data

The primary objective of T3.2 is to advance the current state of haptic engines and providers by using open APIs. This will support at least six simultaneous users and three haptic modalities, combining active, passive, tactile, and kinaesthetic feedback in real-time scenarios. To achieve these goals, the work package focuses on two main objectives: the design and transcoding pipeline, and the communication pipeline. This is contained in the **Core Haptics API** component.

Transcoding pipeline

Currently, the haptic market is fragmented into separate technology silos, with multiple devices and numerous file formats for transmitting device-specific haptic content. While audio and video can deliver a multisensory experience with high fidelity, haptics is not yet encapsulated within encoding formats. However, with efforts from Interhaptics and industrial partners, a reference MPEG haptic software was published in May 2024, giving haptics the same status as audio and video in the technological landscape.

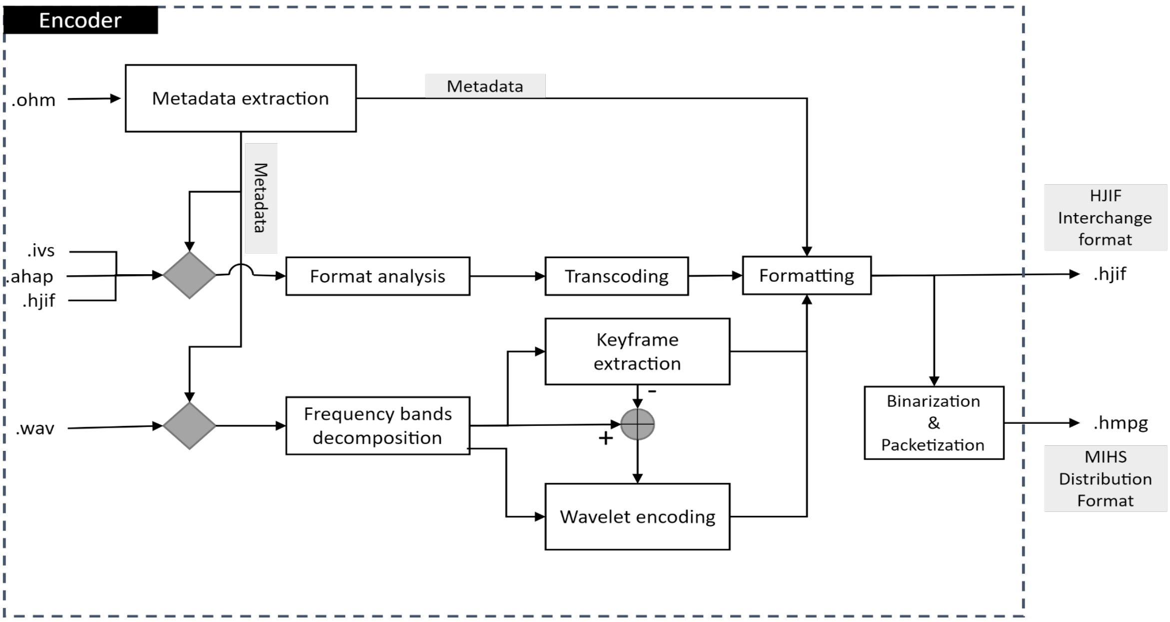


Figure 9: Haptic MPEG Encoder structure.

In PRESENCE, we decided to adopt this standard to facilitate easy implementation by reformatting haptic patterns and intentions over the human avatar. This choice enables the work package to transition from multiple custom formats to the HJIF haptic standard format, integrating multimodal haptics (passive/active tactile, stiffness, and force) into a unique haptic API (PRESENCE Haptic API).

Communication pipeline

The communication pipeline involves decompressing the HJIF MPEG haptic file. When the collision system is triggered and a haptic effect is set to play, the decompression system will decode the HJIF haptic effect into the appropriate device-dependent file format before rendering the effect.

4.2.1. Development keys and achievements

Link to the Haptic Content Creation tools and Device specific tools

To enable end users to utilize the haptic API without managing device-specific operations, each device owner has integrated their own SDK. These SDKs provide consistent features for loading and unloading haptic material, rendering haptic effects, and controlling parameters such as intensity, looping, and the number of repetitions of the target haptic material. Designers can focus on creating and attaching the material to its receiver game object, while the API handles the background implementation of the SDK owner.

The Presence API architecture uses body mapping to target devices. This approach allows developers to trigger any haptic effect on any body part, with the API automatically recognizing the haptic modality, rendering the haptic material, and addressing the necessary device. This device-agnostic approach enables developers to design scenes and associated haptic operations without targeting a specific device.

Link to bodymapping abstraction layer

Inside the API; the PHap_EffectLocation parameter represents the target body part, a local position, and an effect size, that help a device implementer determine which actuator(s) to use. To account for different avatar sizes, a bounding box parameter is also passed to normalize the effect location. The PHap_HapticReceiver is set up to generate this parameter based on a collision with a PHap_HapticEmitter. This is particularly useful when dealing with haptic vests with many actuators.

Development of the transcoding pipeline

One of the key advancements in this task involved providing an API able to encode between the three different custom haptic formats within WP3. This API is able to take as input one of the custom haptic formats (for example .haps) and provide as output the same signal encoded so that other device-specific APIs can use it (.wav or a ScriptableObject, in this case).

This is achieved by first converting one's proprietary format into a "standard" MPEG .hjif file format. Other device implementers translate this format back into their own. This architecture means that each device implementer only needs to be able to convert between their own format and the .hjif format; no knowledge of or reference to other APIs is required. Within the Presence Core API; each device implementor must implement these functions.

When a PHap_BaseEffect is created with a link to an original file, the PHap_Transcoding class is responsible for finding the corresponding PHap_DeviceImplementation and ensuring the file is converted into other formats, provided they support it. It does so through the PHap_Settings class, which contains references to the PHap_DeviceImplementations inside the project.

The PHap_BaseEffect contains links to the file formats of each implementation so they can be accessed at runtime. This only needs to happen when the original file is imported or changed. At runtime, these (serialized) effects will be loaded into their respective APIs so they are already converted or parsed when it has to be played.

As a result, three different haptic modalities (KPI 3.2) can now be effectively encoded into the MPEG .hjif format. The transcoding system was then linked in T3.1 with the haptic emitter system creating an unified haptic virtual material. PHap_BaseEffect are linked to a PHap_HapticEffect inside the scene. This component can modify some parameters of the original, such as the amount of times it is played, or at what intensity. One or more Phap_HapticEffects can then be linked to a PHap_HapticEmitter to send a signal to a PHap_HapticReceiver upon collision.

The transcoding and communication channel has been successfully integrated in the Presence Haptic API. With all work package partners, we have tested and demonstrated the capability of all the integrated sub-systems (T3.1, T3.2 and T3.3) of the Presence Haptic API through a VR scene.

4.2.2. KPIs status

KPI 3.2: *Improve the current state of the art for **haptic engines and providers** delivering **open APIs** for at least **6 (six) simultaneous users** and **≥3 (three) haptic modalities** (as combinations of active, passive, tactile, kinaesthetic) in a real-time scenario (WP3, T3.3, D3.2).*

The improved state of the art is linked to the transcoding pipeline which pushes for an adoption of a standard haptic format delivering an open API. This KPI is also shared between task 3.2 and T3.1. With the current implementation, three haptic modalities are functionally implemented into the transcoding pipeline (Vibration, stiffness & force). The system has been tested with one user while the multiplayer functionalities are due for the next iteration.

4.2.3. Deviations and mitigation plan

Deviations

Within the architecture of the Presence Haptic API, there will not be the use of the Intehaptics engine and providers previously planned. This choice was made to construct a common Haptic API instead that relies on the architecture shown in Figure 2 diagram where all partners use the Core API, a system used to load, convert and play the haptic effects. This deviation however does not impact the “abstraction of a haptic event for a higher-level control and haptic toolbox for haptic stimuli design and spatialization across multi-device (device agnostic approach)” as defined in the DOa.

Mitigation plan

The architecture of the Presence Haptic API allows a high level abstraction layer for the composition of haptic feedback patterns independently from client HW. Indeed, the abstraction of haptic event and spatialization are achieved with the architecture choice of using body mapping to target haptic actuators and devices while the transcoding pipeline allows the achievement of multi-device (device agnostic approach) and the haptic toolbox for haptic stimuli design thanks to the adoption of the standard hjif haptic MPEG format.

For the multiplayer settings, a template will be created using the Presence Haptic API to test at least six simultaneous users. For these tests we do not foresee any major issue or delays as the current architecture implementation allows us to handle the haptic rendering on the client side which ensures synchronization. Any additional delay on the haptic feedback could be related to the event triggered which is handled by the server (position of relative emitter and receiver). We expect this to be handled by the integration partners that should ensure low latency communication between server and client.

4.3. Rendering of haptic data

The main focus of this task is on the rendering of tactile data (captured in T3.1 and interpreted and transmitted in T3.2) utilising tactile displays (as gloves, vests and controllers). The goal is to improve haptic fidelity in order to enhance the perceived presence in social interactions. This will be done by developing a software component that is able to translate individual haptic patterns toward different displays and enable multi-device rendering transcending device-specific limitations. Further a set of SW tools, providing haptic design capabilities to manage specific social haptic interaction patterns and the rendering for multi-device and multi user environments, will be designed and developed to facilitate integration with XR applications (WP1) and demonstrators (WP5).

4.3.1. Development keys and achievements

Multi-device rendering: device owners SDK abstraction

A standardized API was established, incorporating the fundamental features of the haptic rendering layer. Each device owner is responsible for integrating their SDK, which handles rendering for their specific device while enabling the required features. These features include loading and unloading haptic materials, playing and stopping haptic effects based on a material, and configuring additional parameters such as intensity, repetition count, and looping mode. When a haptic operation is executed via the API, the corresponding implementation for each device is triggered and responds accordingly. The dispatching process within the API occurs automatically, remaining transparent to users.

Device Abstraction: Bodymap-Based Triggering

The API abstracts haptic devices by segmenting the body into distinct parts and haptic modalities, ensuring seamless interaction regardless of which devices a user is wearing. When a haptic effect is triggered on a specific body part, the WP3 implementation automatically identifies the appropriate device based on the haptic modality of the material and the targeted body part. If no haptic device is present for a given body part, the trigger has no effect.

Spatialization abstraction: device end-points abstraction

Within task 3.3 a specific implementation is made that targets device specific end points (actuators). As each device can have variable end-points (actuator of any kind) located anywhere on a body part, the actual positioning of these end-points are fully abstracted. Users do not have to know which and how the device is present in a bodypart and simply need to define a 3D localisation (position, orientation, scale) on the bodypart where the haptic effect should be applied. The WP3 implementation is then responsible to convert this location into the appropriate format for each device.

Skinetic implementations

The Skinetic plugin has been reformatted to comply with the API's structure and feature requirements. As an initial implementation to secure rapid iterations, the proprietary .spn format has been implemented for use, pending the development of a full transcoding solution for broader compatibility for rich vibrotactile data formats.

SenseGlove implementations

The SenseGlove plugin has been adjusted to accommodate for the Phap API structure and has been set-up to decode the HJIF file format into the SenseGlove proprietary format. This decoding pipeline is extensively tested. The SenseGlove devices and SKD's are made fully compatible with the WP3.2 transcoding pipeline. Furthermore for latency optimization, SenseGlove has developed an alpha version of their Bluetooth Low Energy communications protocol, which is designed to reduce the latency of haptic rendering when their gloves are used with standalone VR headsets. Initial tests have demonstrated improved performance, and an official firmware version is scheduled for release later this year. This bluetooth low energy implementation will secure more stable haptic rendering from both windows and android operated systems.

4.3.2. KPIs status

KPI 3.3: *Deliver one set of* ***haptic technologies APIs*** *to integrate haptics with the two other pillars, in the* ***two (2) demonstrators*** *and with additional scenarios (WP5, T5.1, D5.3).*

The current version of the haptic API automatically registers and connects the different haptic devices if they are available which allows to trigger haptic effects of any kind on each device. Therefore, it secures interoperability between the three implemented devices. The full integration with other pillars is not achieved yet. This is planned for the next stage of the project.

4.3.3. Deviations and mitigation plan

Deviations

A first deviation is linked to the fact that each device owner has their own custom communication mechanism and rendering for their device which prevents a single rendering engine to handle every device.

A second deviation is concerning that several haptic formats (.ahap, .ohm. .wav .haps and SenseGloves proprietary format) are to be compatible with the API and an automatic transcoding has to be done between each format so that each device owner SDK can use any of the available data, the format being equivalent. The transcoding is not complete yet.

Mitigation plan

Regarding the first deviation, each device owner SDKs are then encompassed into a single core API that the user interacts with. This API has specific feature and behavior that the included SDK should conform with.

To address the second deviation, the transcoding being not fully implemented, Actronika's implementation temporarily uses their proprietary format ".spn" as the input for haptic material, since the full transcoding of this format has been more complicated than initially anticipated. There is now chosen for a system to decode the .spn into a .wav file and then transcode the .wav file. This systematic will be implemented in the next phase of the project.

5. Preliminary evaluation

The MPEG group issued a call for proposal in May 2021 with the objective to create a baseline reference design and software implementation for haptic data representation and coding. The outcome of this call for proposal was a standard format that provides a comprehensive framework for delivering haptic signals. Such standards facilitate the development of current and future applications in different domains spanning from gaming, mobile to virtual reality to cite a few. This effort culminated in a joint publication that presents the first phase of the MPEG haptic coding standard. The publication includes an overview of the codec architecture, preliminary results in terms of compression efficiency, and future plans for haptic coding and distribution.

To evaluate the quality and bit-rate requirements for high-quality haptics encoding, various tests were conducted. While the bit-rate depends on the number of encoded effects, it has been shown that most content can be encoded with a bit-rate lower than 8 kbps for typical use cases.

A user study involving 35 subjects was conducted to assess the perceptual fidelity of the encoded haptic signals (both vibrotactile and kinesthetic). Overall, the combined C2VWR coding detailed in the paper offers the best tradeoff, delivering good results across all bit-rates by leveraging the strengths of both vectorial and wavelet methods.

6. Outlook

6.1. Planned experiments

To validate the main objective of enabling content creation tools and haptic device interoperability, an experiment will assess the perceptual fidelity of transcoded MPEG .hjif haptic signals compared to original OEM formats.

The Presence API, which incorporates the world's first multi-device integration of the MPEG .hjif file format and transcoding pipeline, ensures a standardized approach to haptic content delivery. This API is designed to facilitate seamless interoperability between different haptic devices, thereby promoting a unified standard for haptic content creation and delivery.

The experiment, conducted within Work Package 3 (WP3), will involve end-users evaluating haptic signals of varying richness, such as simple effects where only chances in amplitude appear or more complex effects where there are rapid chances in frequency and where the amplitude varies a lot. Participants will be asked to compare original and transcoded signals to identify any perceptual differences. The hypothesis is that no significant difference in perception will be detected, thereby proving the fidelity and effectiveness of the MPEG .hjif transcoding pipeline. Validation of this hypothesis would confirm the potential of the MPEG .hjif format to ensure consistent and high-quality haptic experiences across devices, enabling industry-wide adoption of a unified standard for haptic content creation and multi-device interoperability..

Our specific hypothesis is that the .hjif format will be comparable to the original format targeted at the specific device. To verify this, we aim to measure the confusion rate of users when they are asked to choose which haptic effect (original or encoded format) is most similar to a base effect (original format).

For this experiment, we will implement a Unity project in a virtual reality (VR) environment where user use a SenseGlove Nova 2 and a Skinetic haptic vest. The user will first be presented with a cube to which we attach a haptic effect in its original format. The user will be able to touch the object for no more than X seconds (This will be defined later). After this, the base effect will disappear, and two cubes will appear in its place. The user will then be able to touch the cubes and express their judgment on which cube matches the base effect. The system will collect the participant's response, the base effect tested, and the response time.

Multiple trials will be presented to the user, and the collected data will be used to analyze the participant's confusion rate and accuracy. A high confusion rate will imply that participants are not able to discern between the original format and the encoded one. A low confusion rate with low accuracy will imply that the participant recognizes the encoded format as the base effect. A low confusion rate with high accuracy will imply that the participant is able to recognize a difference between the original and encoded format and that they are able to identify the original format among the two.

Below, we illustrate a hypothetical scene that represents a trial example during the experiment. This scene will help visualize the process and provide a clear understanding of the experimental setup and objectives.

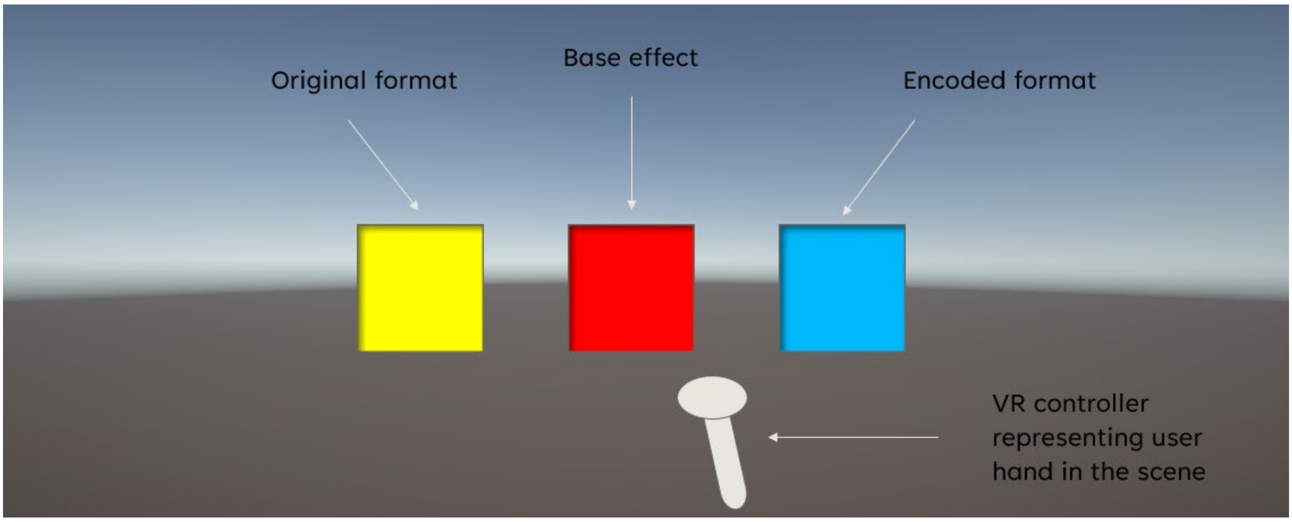


Figure 10: Concept sketch of virtual environment that can represent a scenario during the planned experiment.

The colors of the cubes have been chosen only for representative purposes. During the experiment, the cubes will be of the same color to avoid any bias due to participants' prior preferences. Similarly, also the position of the encoded and original format will be randomised.

To evaluate the richness of the original haptic file compared to its encoded format, we will vary the base haptic effects in terms of frequency, amplitude, duration, and waveform type. By systematically altering one variable at a time while keeping the others constant, we can analyze whether any of these specific variables impact the participant's accuracy in identifying the haptic effects. This approach will help us determine if certain parameters are more influential in maintaining the perceptual fidelity of the haptic signals.

6.2. Planned publications

The results of the experiment will be documented and submitted to leading journals in the fields of haptics, human-computer interaction, multimedia systems or human perception. Potential publication venues include the IEEE Transactions on Haptics, ACM Transactions on Applied Perception, and the Journal of Multimodal User Interfaces. The findings may also be presented at prominent conferences such as the IEEE World Haptics Conference and the ACM Symposium on User Interface Software and Technology (UIST). These publications will detail the methodology, results, and implications of the experiment, highlighting the effectiveness of the MPEG .hjif transcoding pipeline and its potential for industry-wide adoption assuming our hypothesis is confirmed.

6.3. Planned pilots

Following the validation experiment, pilot studies will be conducted within the university setting to further assess the perceptual fidelity of the Presence API and the MPEG .hjif format. These pilots will involve students and faculty members comparing transcoded and original haptic signals in a haptic-enabled application. The focus will be on evaluating user perception to validate the fidelity of the .hjif transcoded format to the OEM format .

6.4. Planned Activities

6.4.1. Planned Activities T3.1

Planned Activities with regards to the SenseGlove interaction SDK. (Content creation)

In the coming period, we are looking into expanding the SenseGlove Interaction SDK with a range of improvements aimed at enhancing flexibility and compatibility. This includes exploring an official Optical Hand Tracking integration (such as Meta Quest) and assessing compatibility with additional tracking devices, for wider compatibility. Efforts may also be directed toward making certain elements of the plugin—especially tracking—more aligned with the Unity XR Interaction System. Furthermore, potential refinements to the Virtual Hand Model and its shaders are being considered. Another area of interest is expanding haptic capabilities, including the ability to create multi-patterned haptic effects on a timeline, offering more dynamic feedback compared to the current sequential vibration model. Allowing to make richer haptic signals that can also be processed and measured during the planned experiments.

Planned Activities with regards to UniTouch Studio (Content creation)

Following the release of effect creation templates, based on contextual design suggestion, combining, sequencing and spatializing different sensation signals in order to propose a “hands on” approach for complex spatialized haptic patterns. We plan to enhance the creation horizon by introducing the possibility to import other samples of various formats. In addition to the 200+ existing contextualized haptic sensations in the Unitouch Studio library, this feature will facilitate the haptic design workflow and offer designers a greater freedom to use different tools.

Planned Activities with regards to Interhaptics Composer (Content creation)

Following the release of the optimization stage, our development roadmap includes significant enhancements to the Haptic Composer. We plan to introduce granular body part targeting, enabling users to design and apply specialized haptic effects to specific areas. This will allow for the creation of highly nuanced and realistic haptic experiences. Furthermore, we will integrate a multi-effect layering feature, simplifying the design of complex spatialized haptic patterns. Users will be able to easily combine and manipulate multiple effects to achieve sophisticated and immersive results. Imagine that you would like to feel both wind and rain together on your haptic vest, then you might want to combine two signals at the same time. This mixing feature will allow actuators to understand two or more effects at the same time. Crucially, these new features will be directly testable within the Composer itself, providing immediate feedback and facilitating rapid iteration during the design process.

Planned Activities with regards to the Material and Bodies layer.

The implementation of the Material and Bodies layer will involve developing a comprehensive reference implementation that leverages the SenseGlove Interaction SDK and the Photon multiplayer engine. The goal is to create a scalable template scene within Unity that demonstrates the practical use of the PRESENCE Haptics API for multiplayer haptic interactions.

This reference implementation will integrate the SenseGlove Interaction SDK to provide accurate and easy to integrate interactions based on a combination of physics based interaction and snap and animation based interaction. By utilizing the SDK’s capabilities to detect hand movements, grabbing, releasing, two handed manipulation, snapzones, and many more pre-programmed interactions, the system will link these interactions to haptic emitters and receivers defined within the PRESENCE Haptics API.

The Photon multiplayer engine will be incorporated to manage networked interactions, allowing multiple users (at least six) to interact within the same virtual environment. This will involve creating a server-client architecture where collision data, positional tracking, and interaction events are managed centrally, while haptic rendering and transcoding of the haptic effects remains client-side to minimize latency.

The template scene will include various interactive objects equipped with haptic emitters, each linked to predefined haptic materials created using tools like Unitouch Studio and the Interhaptics Composer. When users wearing SenseGlove devices interact with these objects, the PRESENCE

Haptics API will trigger appropriate haptic responses based on the collision data and material properties.

Additionally, the scene will include a body mapping system that links haptic signals to specific body parts, ensuring precise haptic rendering even when multiple users interact simultaneously. By providing developers with this comprehensive template, they will have a practical framework to build multiplayer haptic XR applications, demonstrating the power of the PRESENCE Haptics API in supporting real-time, multi-device interactions.

6.4.2. Planned Activities T3.2

Planned activities with regards to the transcoding pipeline

Presence Haptic API, future work will focus on expanding its capabilities, particularly in supporting multiplayer functionality and enhancing format compatibility. Testing will be conducted to ensure the API can effectively handle interactions involving at least six simultaneous users, with specific attention given to minimizing latency through close collaboration with integration partners. To further improve compatibility and standardization, efforts will be directed toward validating the encoding fidelity of the HJIF MPEG haptic format by comparing it against original proprietary formats. This evaluation aims to establish the HJIF format as a reliable industry standard, promoting wider adoption. Additionally, the development process will include continuous refinement of the Haptic API, informed by feedback from integration partners to enhance usability and performance. A key aspect of the upcoming work will be the integration of the full transcoding pipeline for the .spn format used by Actronika. This will be achieved by converting .spn files to .wav files and then utilizing the existing infrastructure to transcode the .wav files into the HJIF format. This approach ensures compatibility with the broader Presence Haptics API framework, enabling seamless communication between diverse haptic devices and enhancing the overall robustness of the system.

6.4.3. Planned activities T3.3

Planned Activities with regards to Skinetic Integration

Regarding Skinetic integration, the focus will be on the implementation of a full transcoding pipeline to allow a spatialized display of haptic effects originating from different design tools. This will secure as well a smooth integration pipeline for the use cases integration within WP5 collaboration and supply the experiment support.
It is also planned to continue to stabilize and release the firmware for all Skinetic devices.

Planned Activities with regards to SenseGlove Integration

For SenseGlove it is key to fully release the Bluetooth low energy implementation on firmware level. The development release shows a marked decrease in communication issues on Android. This also secures the possibilities to play-back all desired haptic effects within acceptable latency on standalone HMD’s It is planned to stabilise the firmware and release it as the new firmware version for all SenseGlove Nova 2 devices.

7. Ethical considerations

As immersive haptic technology enables users to physically interact in virtual environments, ethical considerations become paramount, particularly regarding consent, responsibility, and the impact of touch. A primary concern is the potential for unwanted physical interaction through haptic feedback, as users experience virtual touch in a way that can feel real and personal.

To mitigate risks, PRESENCE adheres to real-world social norms, ensuring that users are responsible for their own avatars. Just as in physical interactions, inappropriate touch is the responsibility of the initiating party, reinforcing accountability and respectful behaviour. Our implementation deliberately restricts haptic feedback to the hands and chest, with the chest being particularly sensitive to inappropriate behaviour. Recognizing the potential for misuse, we have designed the system to ensure that only users equipped with haptic devices can both touch and be touched. This prevents individuals with lower immersion levels from engaging in interactions without fully understanding the real-world impact of their actions, thereby maintaining the integrity of the virtual experience.

Furthermore, PRESENCE prioritizes safety in hardware and software design. All haptic devices integrated into the unified API are CE-certified and developed with inherent safety measures to protect users from physical harm or discomfort. By incorporating these ethical safeguards, we ensure that the PRESENCE haptics framework supports meaningful and immersive haptic interactions while upholding standards of respect, consent, and safety. This structured approach not only fosters a secure and inclusive virtual environment but also establishes best practices for the ethical development of multiplayer haptic experiences.

8. Abbreviations and definitions

4.2. Abbreviations

AI	Artificial Intelligence
API	Application programming interface
BLE	Bluetooth Low Energy
DoA	Description of the Action (Annex I to the contract)
HW	Hardware
HMD	Head Mounted Display
MPEG	Moving Picture Experts Group
OEM	Original Equipment Manufacturer
QA	Quality Assurance
SDK	Software Development Kit
UX	User Experience
VR	Virtual Reality
WP	Work Package
XR	Extended Reality
UIST	User Interface Software and Technology

4.3. Definitions

API Users	Developers using the Haptics API as part of their project (e.g. Zaubar, Vection).
Device Implementers	Companies looking to integrate their Haptic Device(s) and/or plugins inside the Presence Haptics API. (e.g. SenseGlove, Actronika, Interhaptics).
Transcoding	Taking one file format (e.g. .haps) and “translating“ it into the file format(s) that other device implementations can use (e.g. .json, .hjif).
hjif	Haptics Json Interchange Format: A generic (json) format file type containing one or multiple haptic effects.
Normalization (of values)	Mapping a value from a known range [min .. max] to the [0 .. 1] range, to use as a generic input variable.
Manipulation (System)	A system to “grab” and “release” virtual objects based on buttons and/or gestures, and to control their behaviour while grabbed. In some cases, an object may only slide or rotate along one axis. Essentially, the system used to determine if/when an object is “grabable”.
Interaction (System)	A system that combines logic, manipulation, haptics and other forms of feedback to create tasks that can be performed inside a virtual experience (e.g. the Unity XR Interaction Toolkit).

Annex 1 Specification Sheet SenseGlove Nova 2

SENSEGLOVE

SENSEGLOVE NOVA 2 SPEC SHEET
(V1.0 JUN 2024)

Overview of equipment in the Nova 2 Box

SenseGlove Nova 2 Specs:

Features	Specs per glove
Force Feedback	4x 1 DoF passive magnetic brake force feedback for the thumb, index, middle, and ring fingers, with a maximum force resistance of 24N*. The force can be controlled in 100 steps, though linear behavior is not guaranteed. In practical use, 10-15 distinct steps of stiffness can be felt. Note: there is no capability for force control of the passive force feedback. **Our minimum Quality Control (QC) is on 20N; we strive to achieve 24N on at least the thumb and index fingers.*
Haptic Feedback	2x LRA haptic feedback on thumb and index fingertips 2x LRA haptic feedback on the active strap in the palm of the hand one LRA situated on either side of the palm. (Left and right)

1

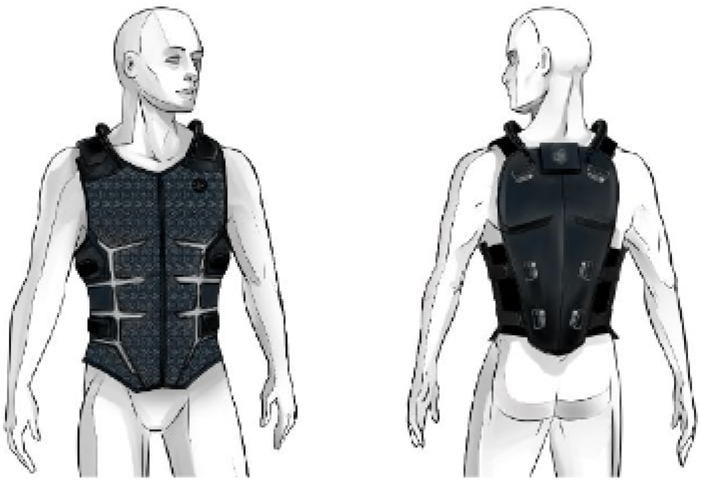
SENSEGLOVE

Contact Feedback	1x Active strap, 1 DoF control pushing against the palm of the hand
Finger tracking	6 DoF finger tracking (2 DoF Thumb, 2 DoF index, 2x 1 DoF middle and ring finger)
Battery Life	3450mAh Lithium-Ion Battery, good for an average of 3 hours of simulation time
Communication	2.4GHz wireless serial communication, 60Hz – 90Hz refresh rate
Software	Compatible with SenseCom device communication program. Compatible with Windows, Linux, and Android devices.
Development Tools	Native C++ API, Development plugins for Unity and Unreal Engine and Ros1 Noetic (update for ROS 2 is planned)
Position Tracking	Third party position tracking required. Available via a controller/ tracker mounted with a single screw to each glove (instructions and parts included). Mounts available for: Meta Quest 2, Quest3 and Pro. HTC Vive Focus/XR Elite wrist bands, HTC Vive Pro trackers and HTC Ultimate tracker.
General	Weight: ~350g (12.3 oz) per glove.

2

Annex 2 Specification Sheet Skinetic - Creators Edition.

Datasheet: Skinetic - Creators Edition



Skinetic Creators Edition is the first haptic suit integrating high definition vibrotactile actuators. With Skinetic, we bring a sense of touch for extended reality.

Skinetic Creators Edition integrated twenty high-definition haptic actuators, enabling you to make you feel every interaction in a virtual environment on the torso.

Through the fusion of fashion and high-technologies, we designed a genderless and unisize wearable device to deliver immersive VR for everyone.

Suggested Applications
• Entertainment: XR, Gaming, Cultural, Music • Enterprise: Training, Simulation, Sensitization

Table 1. Skinetic suggested applications

Feature	Benefits
Haptic	• Wideband motors covering most of tactile sensitivity. • Short response-time
Design	• One-size-fits-all • Fast dressing

Communication	• Integrated Haptic Engine
	  ○ USB, Bluetooth, WiFi
	• Multichannel Audio Interface
	  ○ USB

Table 2. Skinetic Creators Edition kley benefits

Technical Specifications

Vest design

Parameter	Specification	Details
Feedback points	20 Wideband Haptic Actuators	
Size	One-size-fits-all	XS to XXL
Dressing time	< 1 min	
Fitting points	6	2x rigid on shoulders 4x stretch on sides
Weight	2.4kg	Vest: 1.8kg Battery: 610g

Electrical Characteristics

Parameter	Specification	Details
Input voltage	20V	
Battery Energy Value	100Wh	
Battery Autonomy	Up to 8 hours	in standard usage conditions
Battery Charge duration	3 hours	with a 45W charger

Connectivity

Parameter	Specification	Details
USB	Multichannel Audio Interface	USB Audio Class 2.0, 20 channels
USB	HID	Command based
Bluetooth 5.0	Serial	Command based
WiFi protocol	WiFi 4	Command based

Environmental characteristics

Parameter	Specification	Conditions
Operating temperature	[-20 °C; 40 °C]	
Storage temperature	[-40 °C; 70 °C]	