consider it

co-creating the digital future

C-ITS Forum 2025

Sichere Kreuzung mittels C-ITS-Technologien

Leutrim Mustafa

Head of consider innovation

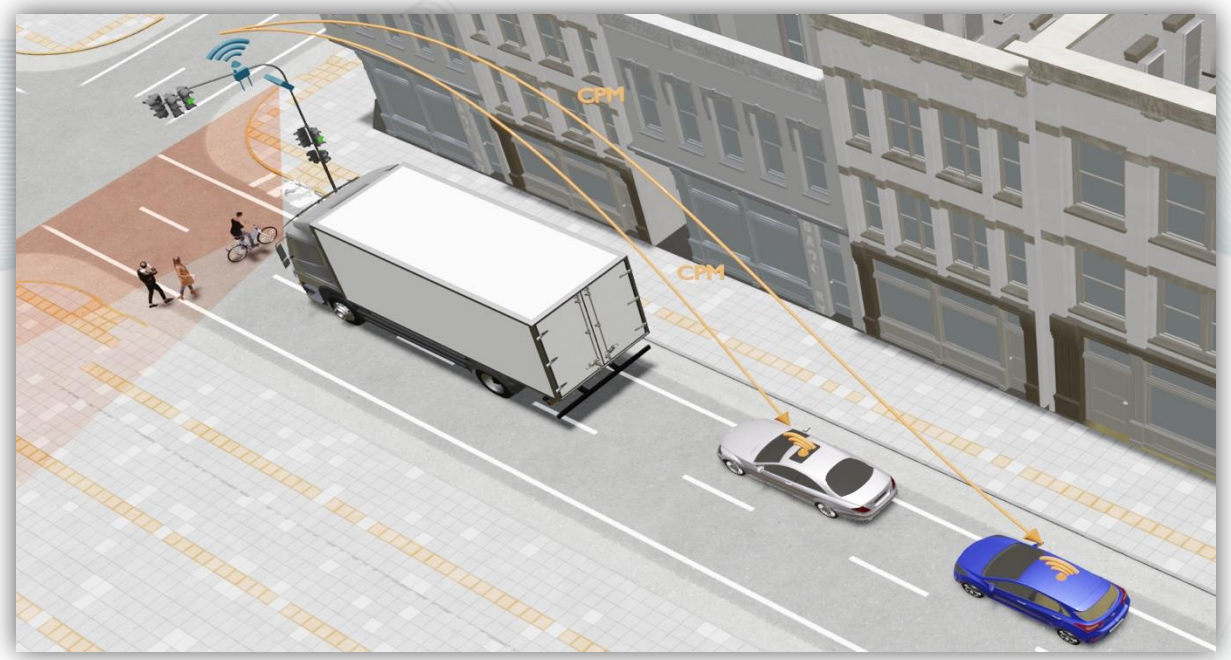
mustafa@consider-it.de

+49 176 4398 6409

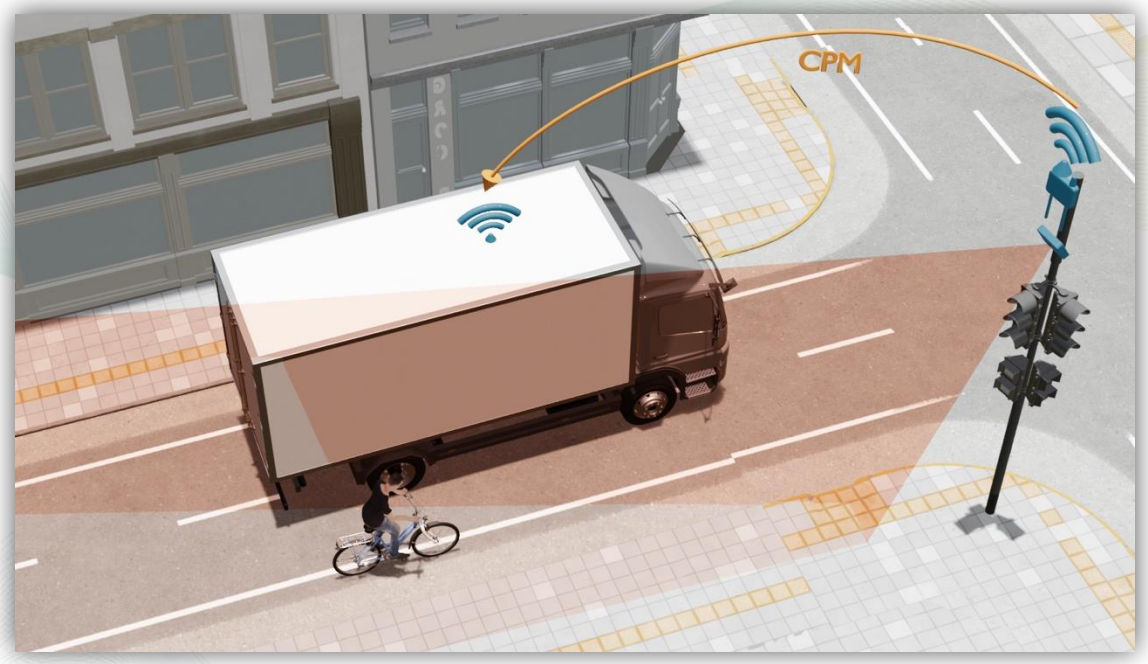
26. Februar 2025

Ziel des Collective-Perception Service

1. Wahrnehmung der VRUs.



2. Informationen für Entscheidungen.



VRU Erkennung

CPM Erzeugung

CPM Verarbeitung

Warnmeldung

Datenstruktur einer CPM

1. Header

- protocolVersion – Protokollversion
- messageID – Eindeutige Nachricht-ID
- stationID – ID der sendenden Station
- generationDeltaTime – Zeitstempel der Nachricht

2. cpmParameters

ManagementContainer

- stationDataContainer – Informationen zur Station
- referencePosition – Georeferenzierte Position

SensorInformationContainer

- Sensordaten, die von der Station gesammelt werden

3. PerceivedObjectContainer *(Erfasste Objekte)*

Objekterkennung und Klassifizierung

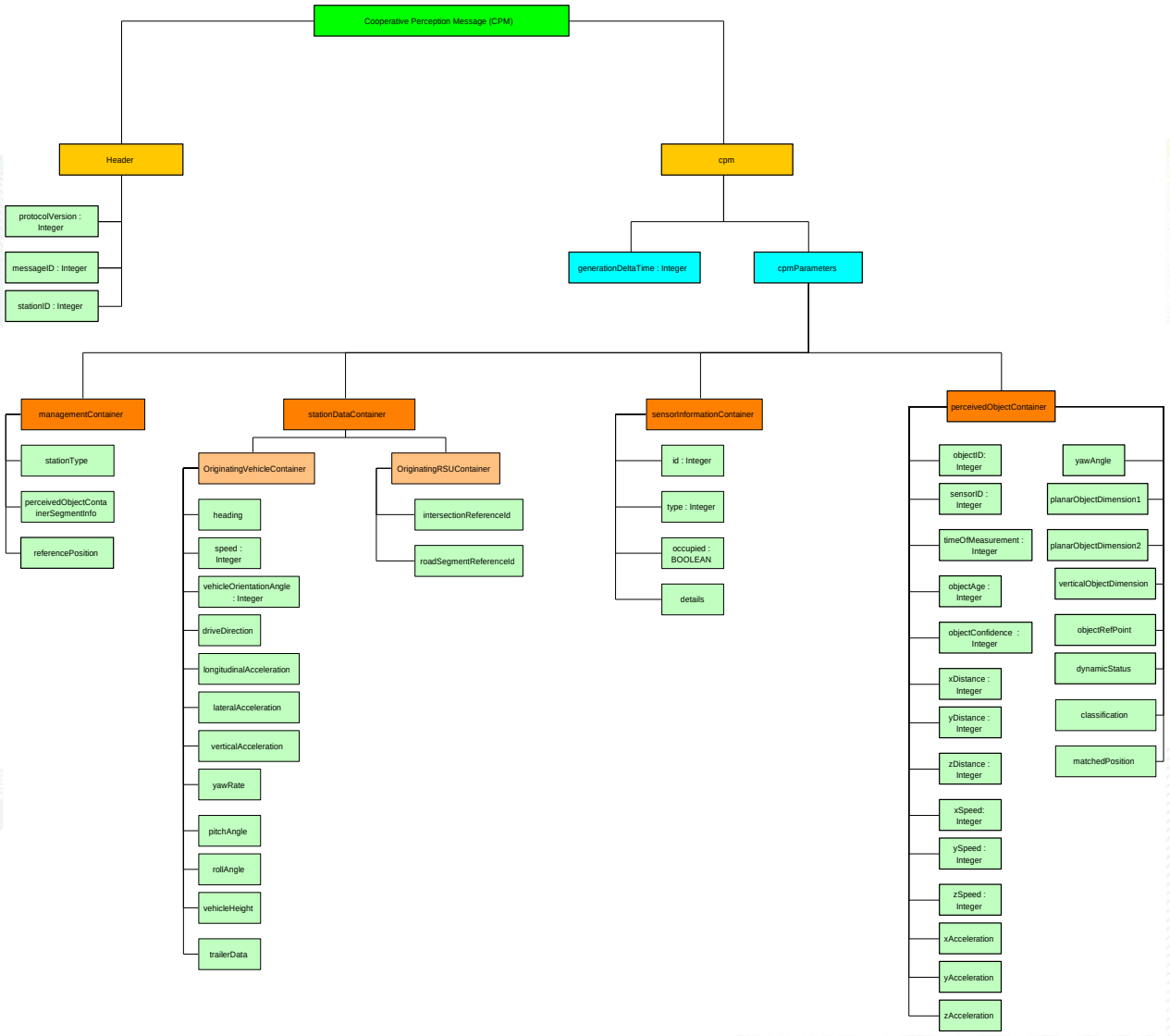
- objectID – Eindeutige Objektkennung
- sensorID – ID des Sensors, der das Objekt erkannt hat
- timeOfMeasurement – Zeitpunkt der Messung
- objectAge – Alter der Objekterkennung
- objectConfidence – Zuverlässigkeit der Erkennung

Positions- und Bewegungsdaten

- xDistance, yDistance, zDistance – Objektposition
- xSpeed, ySpeed, zSpeed – Geschwindigkeit entlang der Achsen
- xAcceleration, yAcceleration, zAcceleration – Beschleunigung

Objektattribute

- classification – Typ des Objekts (z.B. Fahrzeug, Fußgänger)
- objectRefPoint – Referenzpunkt des Objekts



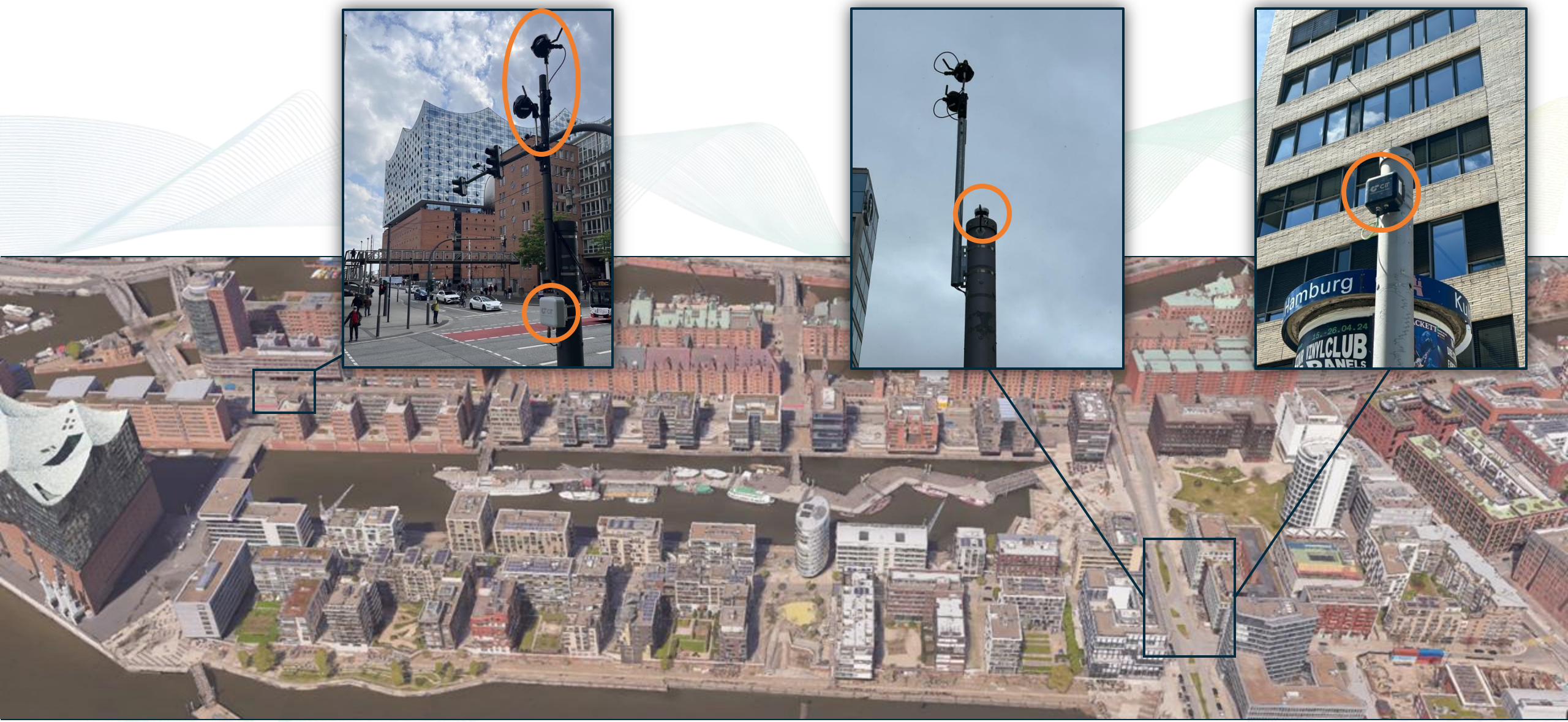
Beispiel CPM

```
CPM {
    header: ItsPduHeader {
        protocol_version: 1,
        message_id: 14,
        station_id: StationID(
            3200371753,
        ),
    },
    cpm: CollectivePerceptionMessage {
        generation_delta_time: GenerationDeltaTime(
            0,
        ),
        cpm_parameters: CpmParameters {
            management_container: CpmManagementContainer {
                station_type: StationType(
                    15,
                ),
                perceived_object_container_segment_info: None,
                reference_position: ReferencePosition {
                    latitude: Latitude(
                        535419159,
                    ),
                    longitude: Longitude(
                        99933856,
                    ),
                    position_confidence_ellipse: PosConfidenceEllipse {
                        semi_major_confidence: SemiAxisLength(
                            4095,
                        ),
                        semi_minor_confidence: SemiAxisLength(
                            4095,
                        ),
                        semi_major_orientation: HeadingValue(
                            3601,
                        ),
                    },
                    altitude: Altitude {
                        altitude_value: AltitudeValue(
                            982,
                        ),
                        altitude_confidence: unavailable,
                    },
                },
            },
```

```
CPM {
    cpm: CollectivePerceptionMessage {
        cpm_parameters: CpmParameters {
            },
            station_data_container: None,
            sensor_information_container: Some(
                SensorInformationContainer(
                    [
                        SensorInformation {
                            sensor_id: Identifier(
                                3,
                            ),
                            type: SensorType(
                                8,
                            ),
                            detection_area: stationarySensorPolygon(
                                AreaPolygon {
                                    poly_point_list: PolyPointList(
                                        [
                                            OffsetPoint {
                                                node_offset_pointxy: node_XY5(
                                                    NodeXY28b {
                                                        x: OffsetB14(
                                                            -4922,
                                                        ),
                                                        y: OffsetB14(
                                                            -2845,
                                                        ),
                                                    },
                                                ),
                                                node_offset_point_z: None,
                                            },
                                            OffsetPoint {
                                                node_offset_pointxy: node_XY4(
                                                    NodeXY26b {
                                                        x: OffsetB13(
                                                            -628,
                                                        ),
                                                        y: OffsetB13(
                                                            -2505,
                                                        ),
                                                    },
                                                ),
                                                node_offset_point_z: None,
                                            },
```

```
CPM {
    cpm: CollectivePerceptionMessage {
        cpm_parameters: CpmParameters {
            perceived_object_container: Some(
                PerceivedObjectContainer(
                    [
                        PerceivedObject {
                            object_id: Identifier(
                                36,
                            ),
                            sensor_id_list: Some(
                                SensorIdList(
                                    [
                                        Identifier(
                                            3,
                                        ),
                                    ],
                                ),
                            ),
                            time_of_measurement: TimeOfMeasurement(
                                1500,
                            ),
                            object_age: None,
                            object_confidence: Some(
                                ObjectConfidence(
                                    101,
                                ),
                            ),
                            x_distance: ObjectDistanceWithConfidence {
                                value: DistanceValue(
                                    2544,
                                ),
                                confidence: DistanceConfidence(
                                    102,
                                ),
                            },
                            y_distance: ObjectDistanceWithConfidence {
                                value: DistanceValue(
                                    -4590,
                                ),
                                confidence: DistanceConfidence(
                                    102,
                                ),
                            },
                            z_distance: None,
                            x_speed: SpeedExtended {
                                value: SpeedValueExtended(
                                    41,
                                ),
                                confidence: SpeedConfidence(
```

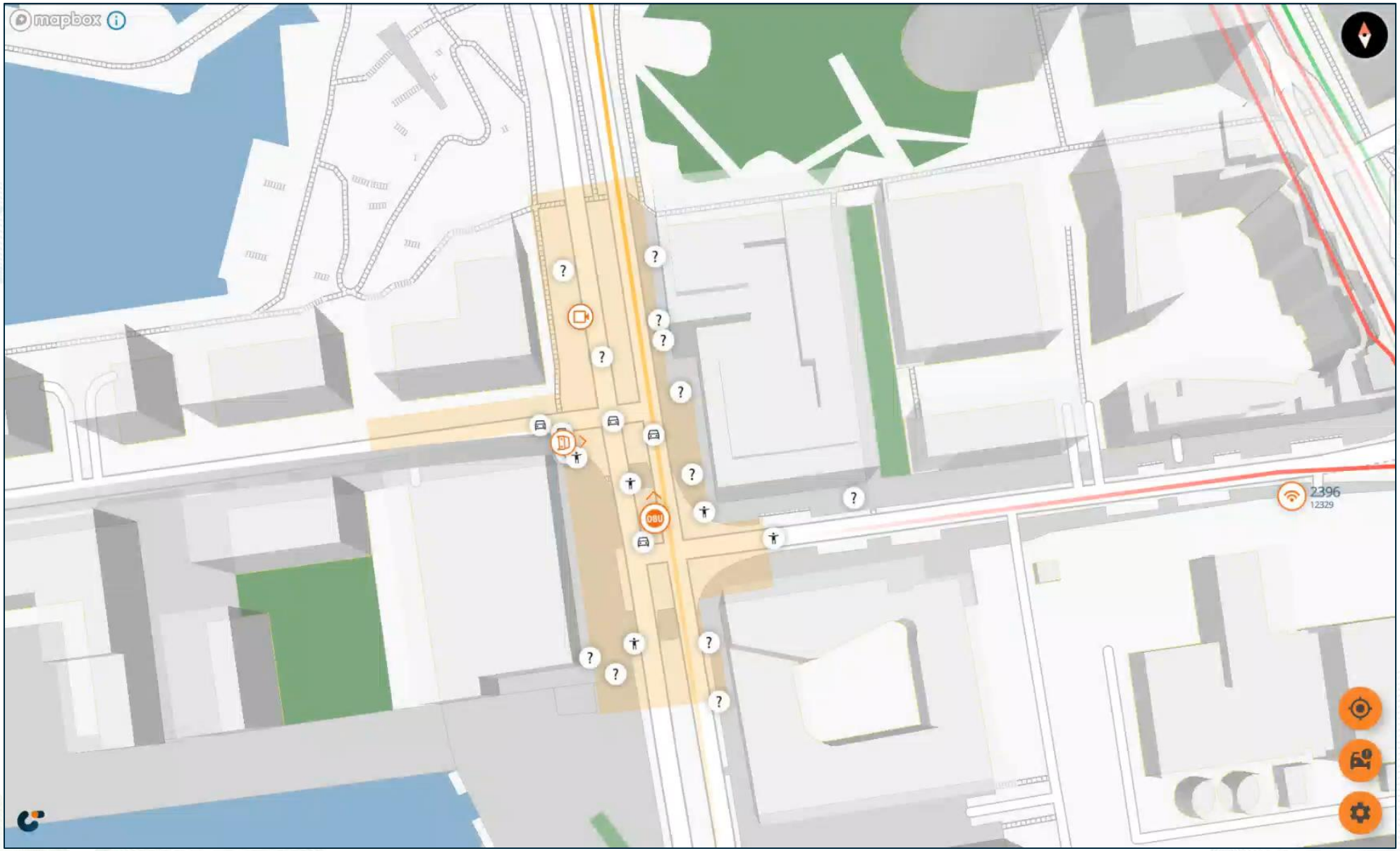
Praxisbeispiele aus Hamburg

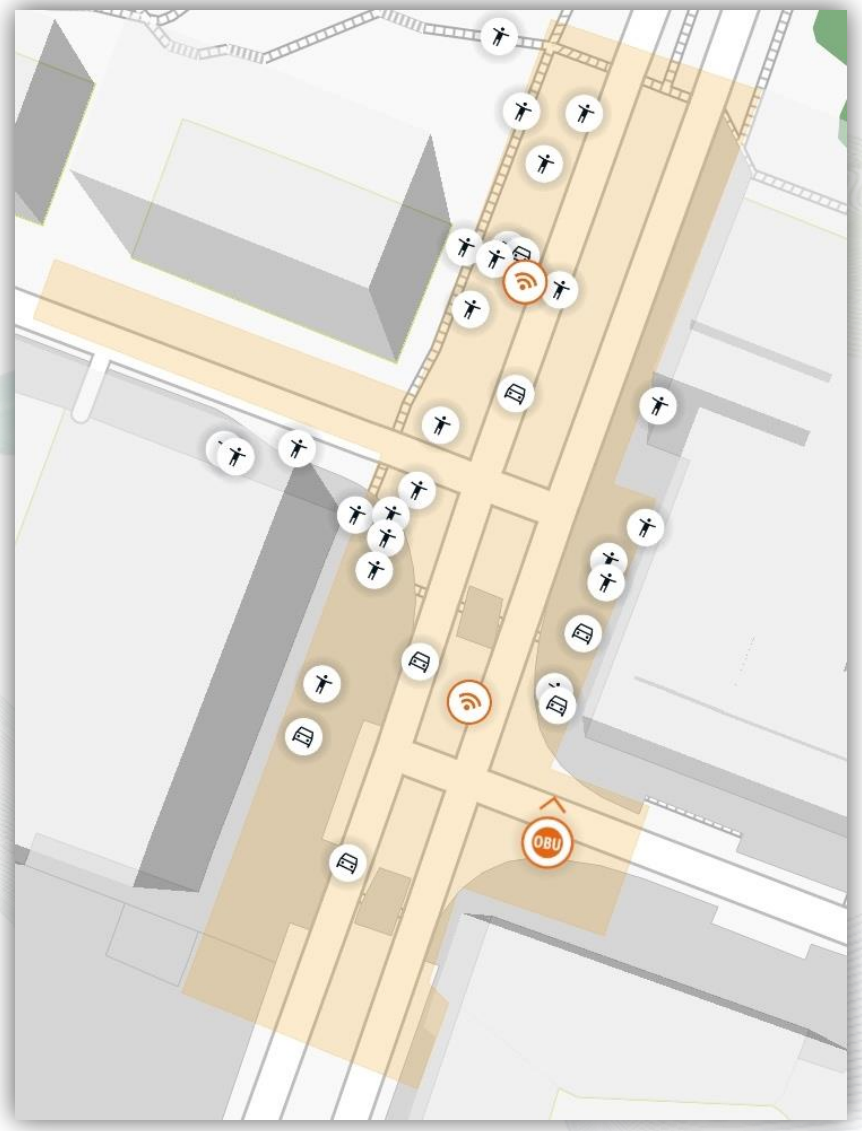


Praxisbeispiele aus Hamburg

Praxisbeispiele aus Hamburg

Praxisbeispiele aus Hamburg



Erkenntnisse

- Ohne Datenfusion gerät die CPM an die Belastungsgrenze.
- Die Latenzzeit von Erhalt der Sensorinformationen bis versendete CPM beträgt im Durchschnitt 46ms.
- Es bestehen starke Abhängigkeiten zur verwendeten Sensorik:
  - Objektklassifizierung
  - Güte der Daten
  - Sensorreichweite