ETSI GS ARF 008 V1.1.1 (2024-12)

GROUP SPECIFICATION

Augmented Reality Framework (ARF); Open APIs for Managing and Querying the World Analysis function

Disclaimer

The present document has been produced and approved by the Augmented Reality Framework (ARF) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGS/ARF-008

Keywords

API, augmented reality, context capturing and analysis, interoperability, real world capture

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the ETSI Search & Browse Standards application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on ETSI deliver repository.

Users should be aware that the present document may be revised or have its status changed, this information is available in the Milestones listing.

If you find errors in the present document, please send your comments to the relevant service listed under Committee Support Staff.

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure (CVD) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Augmented Reality Framework (ARF).

The ISG ARF shares the following understanding for Augmented Reality: Augmented Reality (AR) is the ability to mix in real-time spatially-registered digital content with the real world. The present document specifies the interoperability requirements for Reference Points AR 16 and AR 17 of the reference architecture for AR solutions defined in ETSI GS ARF 003 [1].

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document provides an overview and an introduction to the interface specification for the reference point "AR 8 - Pose" of the AR framework architecture [1] developed by the ETSI Industry Specification Group (ISG) for an Augmented Reality Framework (ARF). The actual interface specification is provided as OpenAPI™ specification [3] and forms the baseline for the present document.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS ARF 003 (V1.1.1): "Augmented Reality Framework (ARF); AR framework architecture".

[2] ETSI GS ARF 004-4 (V1.1.1): "Augmented Reality Framework (ARF); Interoperability Requirements for AR components, systems and services; Part 4: World Analysis, World Storage and Scene Management functions".

[3] ETSI GS ARF 005 (V2.1.1): "Augmented Reality Framework (ARF); Open APIs for the Creation, Management and Querying of the World Representation".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] IETF RFC 4122: "A Universally Unique IDentifier (UUID) URN Namespace".

[i.2] IETF RFC 2616: "Hypertext Transfer Protocol HTTP/1.1".

[i.3] OpenAPI Specification v3.0.0.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

AR experience: real time perception of the mixture of the real world and spatially-registered digital content by user senses

AR system: combination of hardware and software that delivers an AR experience

Augmented Reality (AR): ability to mix in real-time spatially-registered digital content with the real world

feature: characteristics of a real world element that can be searched, recognized or tracked

NOTE: Features can be of different nature without being limited to visual patterns, UWB, Wi-Fi®, Infra Red or sounds.

pose: combination of position and orientation

reference point: point located at the interface of two non-overlapping functions of the AR framework architecture and representing interrelated interactions between those functions

trackable: element of the real world of which features are available and/or could be extracted

NOTE: Features can be made available from an analysis of the element itself (fiducial markers, natural images, 3D point cloud) or processed from a representation of the element (3D CAD model).

world anchor: coordinate system related to an element of the real world on which virtual content stays spatially-registered

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
AR	Augmented Reality
ARF	Augmented Reality Framework
ISG	Industry Specification Group
JSON	JavaScript Object Notation
REST	REpresentational State Transfer
UI	User Interface
UUID	Universally Unique IDentifier
WA	World Analysis
YAML	YAML Ain't Markup Language

4 Basic structure of the World Analysis API for Pose and Capabilities

4.1 Overview

The World Analysis API for Pose (AR 8) allows the Scene Management to retrieve the Pose(s) of the AR device and/or of one or more World Anchors and Trackables to update its AR scene representation at runtime. These Poses are estimated by the World Analysis function which will require relocalization information extracted from the World Graph stored in the World Storage function. The API also gives access to the capabilities of the World Analysis to the Scene Management to adapt and offer the best service to end-users.

Figure 1 shows a typical architecture used by an AR scene management. The Scene Management function loads an AR scene including one or several World Anchor or Trackable references to which AR assets are attached. The API has to assure an easy and elementary access to the World Analysis to retrieve to Poses estimated at runtime (by relying on the World Capture and World Storage functions) as well as its capabilities to adapt the AR scene accordingly and display it to the end user.

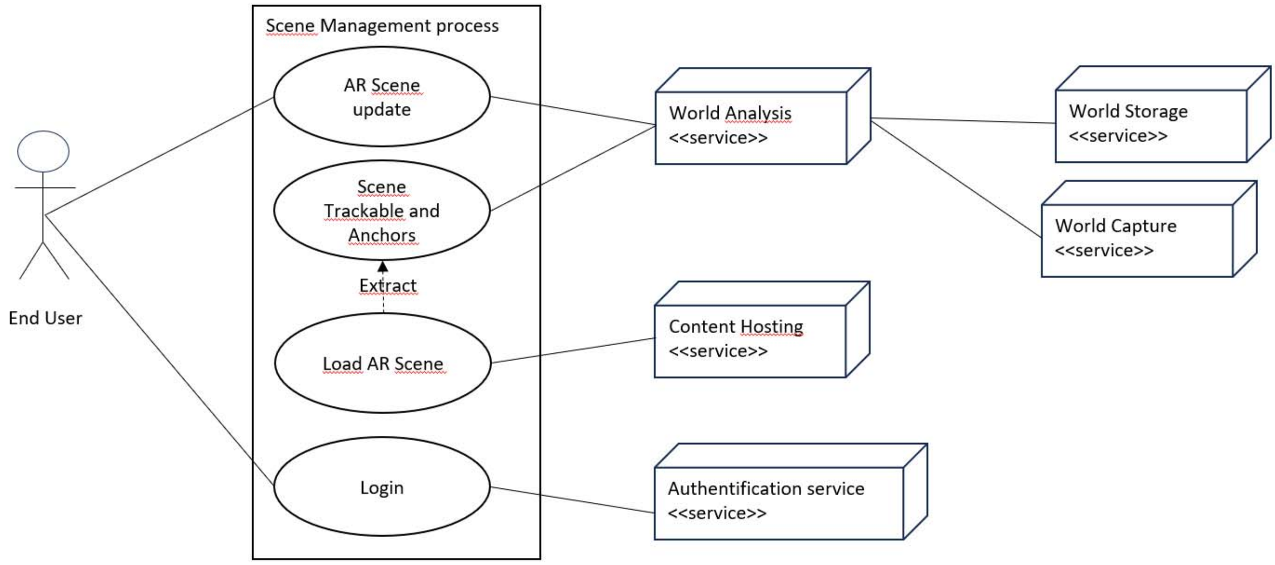

Scene Management process
AR Scene update
Scene Trackable and Anchors
Extract
Load AR Scene
Login
End User
World Analysis <<service>>
World Storage <<service>>
World Capture <<service>>
Content Hosting <<service>>
Authentification service <<service>>

Figure 1: Architecture of a typical AR scene management system

4.2 Pose estimation

Augmented Reality (AR) registers AR assets in 3D with the real world in real-time. To achieve this, an AR system estimates the 3D transform of the AR device in relation to the real world objects, or conversely. This so called Pose is given in a specific format depending on the use-case. When the Scene Management requests the Pose of a given element from the World Analysis, the latter combines the information provided by the World Capture (obtained from the device sensors) with the relocalization information provided by the World Storage, to compute the resulting Pose.

To provide different possibilities on various devices and AR ecosystems, a Pose is associated with a mode that determines if it corresponds to the Pose of an AR device in relation to a World Anchor or a Trackable, or conversely a Trackable or World Anchor in relation to the AR device. An Augmented Reality developer should select one mode, adapted to the scenario.

Depending on the use-case (for instance outdoor or indoor), a pose can be expressed in various formats according to the request of the Scene Management. Those include but are not limited to a Geodetic reference frame or a cartesian reference frame.

An AR system can exploit the context in which the Pose was collected by the World Analysis to adapt the AR experience. Such a context can include the timestamp of the Pose, the estimation state (tracked, not tracked), the confidence of the Pose or even instructions to improve the estimation (for instance move slower, increase lighting, etc.). Such context metadata are associated with a Pose.

The World Analysis API allows to request the Pose in the different modes and formats and to collect the different metadata associated with a given Pose.

4.3 Structure of the API

The World Analysis API is part of the ARF Open API v2.0.0 and available on the ETSI Forge at the following URL:

- https://forge.etsi.org/rep/arf/openapi/-/tree/2.0.0

The API is composed of a main OpenAPI yaml file located here:

- https://forge.etsi.org/rep/arf/openapi/-/blob/2.0.0/API/worldanalysis/worldanalysisopenapi.yaml

It also relies on another OpenAPI yaml file defining common data formats with the World Storage API located here:

- https://forge.etsi.org/rep/arf/openapi/-/blob/2.0.0/API/common/commonopenapi.yaml

The structure of these files is based on the OpenAPI™ 3.0.0 specification [i.3].

The two files exhibit 5 main parts:

- Header part (copyrights, info, servers, tags).
- Paths for administration (/**ping**, /**admin**, /**version**).
- Paths for collecting Poses (**/pose**), and paths for managing subscriptions to Poses (**/pose/subscriptions**).
- Paths for retrieving the capabilities supported by the World Analysis (**/capabilities**).
- Components part - containing the various schemas (in/out, mode and geometry parameters) and HTTP responses (client and server errors, warnings). All responses are now packaged into a schema with a separate "message" string component.

JSON should be used as coding protocol for parameter and object structures. The functions' return values for single UUIDs shall be non-encoded strings. Other text as well as binary data formats may also be used if some systems request this.

NOTE: Information about the definition and the characteristics of UUIDs can be found in [i.1]. The HTTP semantics are described in [i.2].

The administration paths can be used to test a server, to get the state of a server, and to request the API version.

The API may be tested in the "rendered file" view on ETSI Forge, directly from the repository.

An implementation of the API shall follow the specification in the linked OpenAPI yaml file.

5 Description of the API

5.1 Pose

The OpenAPI path **/pose** in the YAML file "worldanalysisopenapi.yaml" defines the operation to collect the Poses of AR devices, Trackables and World Anchors and manage subscriptions to these Poses (as described in the AR Framework architecture document [1] as the reference point "AR 8-Pose" and following the requirements detailed in [2]).

Operations:

- Requesting the last Pose of one or multiple World Anchors and Trackables will return one or a list of Pose(s) with the more up-to-date information collected by the World Analysis (Pose value, timestamp, confidence, estimation state, etc.) as a JSON document.
- Subscribe to collect the Pose of AR devices, World Anchors and Trackables when they are modified will return a subscription that includes the callback url on which the updated Poses will be sent by the World Analysis as a JSON document. Callbacks can be managed through Webhooks, WebSockets and HTTP.
- Update a subscription information (i.e. validity, callback url, etc.) will return the modified subscription with its updated parameters as a JSON document.
- Deleting a subscription will stop the associated callback for a pose. When the Pose is changed no information will be sent to the client anymore.
- Specifying the minimum frame rate for Pose estimation for given trackable types; will return with success if this frame rate is supported (PoseConfiguration) or will return an error code.

A server shall implement all operations for managing Poses and subscriptions. It may refrain from collecting Poses for given Trackables or World Anchors depending on the end user access rights.

NOTE 1: Webhooks and WebSockets: If a client requests a subscription and will use a Webhook, the client has to implement the Webhook functionalities itself and start the service. If the client will use WebSockets, it has to open actively the connection to the World Analysis server and wait for new Pose message(s) from the server. The pose has to be coded in a JSON format using the API Pose schema.
Aside from that, the client has to handle the unsubscription mechanism and close all mechanisms for both ways of communication.

NOTE 2: To not overload the bandwidth of the World Analysis, the client may create one WebSocket client and handle the incoming data according to its various Pose subscriptions.

5.2 Capabilities

The OpenAPI path **/capabilities** in the YAML file "worldanalysisopenapi.yaml" defines the operations to retrieve the supported capabilities of the World Analysis (as described in the AR Framework architecture document [1] as the reference point "AR 8-Pose" and following the requirements detailed in [2]).

Operations:

- Accessing all Capabilities of the World Analysis will return a JSON document with all the supported Trackable Types and associated parameters (encoding information, accuracy, latency, framerate).
- Accessing all capabilities associated with the Trackables that are supported to obtain the Pose of a given Trackable or World Anchor will return a JSON document with all the supported Trackable Types and associated parameters.

A server shall implement all operations for managing Capabilities.

Annex A (informative): Example Usage of the World Analysis API

A.1 Introduction

This annex provides an example (Validation Application Server Maintenance) illustrating the application of the World Analysis API file specified in the present document. The example is kept simple and uses a frequent scenario where an AR system spatially registers virtual content on physical objects (in the real world).

A.2 Specification of the Validation Application 'Server Maintenance'

A large space (e.g. the inside and the outside of a building) is to be enriched with some virtual content via the help of AR techniques to guide and assist a data center technician.

Use case: A staff member of the building managing a data center wants to guide and assist an external technician for the replacement of a defective equipment (i.e. server or switch) in a given rack. The external worker needs to be as autonomous as possible and does not know anything about the building. Then, the staff member decides to use AR to display user interfaces, textual information and meshes to successively:

1) Guide the technician from outside the building to the reception for requesting the access to the inside of the building.

2) Guide the technician from the reception to the entrance of the data center.

3) Indicate at the entrance of the data center the safety instructions and make them validate they did take them into account.

4) Indicate in the data center the place where the equipment to be installed is stored.

5) Indicate in a given rack the defective equipment that needs to be replaced.

A.3 World Graph and AR Scene Graph of the Validation Application

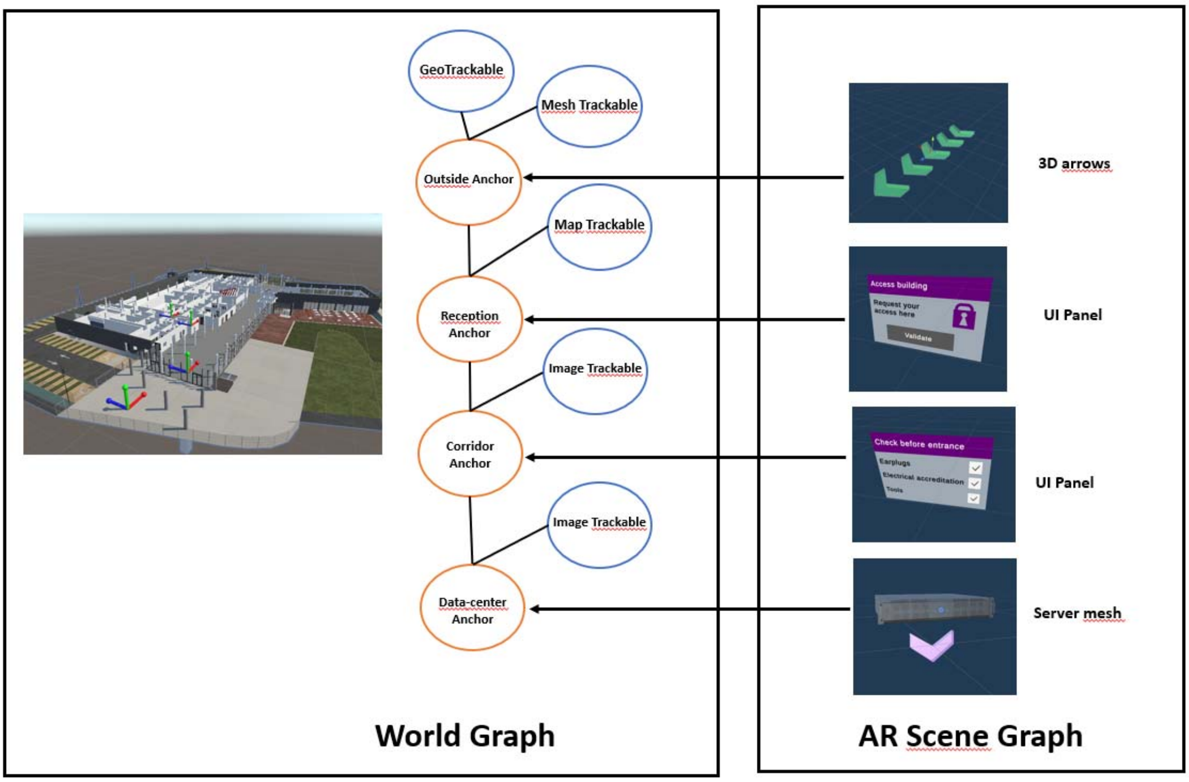


Figure A.1: Example of World Graph for the data center technician validation application

Figure A.1 represents the data center technician use case:

1) The left picture shows the location where the AR elements will be deployed with some visible World Anchors.

2) The World Graph (middle of the picture) shows the objects and their relations. The lines represent the World Links. Trackables are drawn in a blue circle and World Anchors in orange ones.

3) The right part in the figure (AR Scene Graph) represents the AR content that the user will see in the final application.

NOTE: Due to differences in the locations of the AR experience and the number of in-between steps, the World Graph may differ from case to case.

A.4 Example Usage of the API (Validation Application)

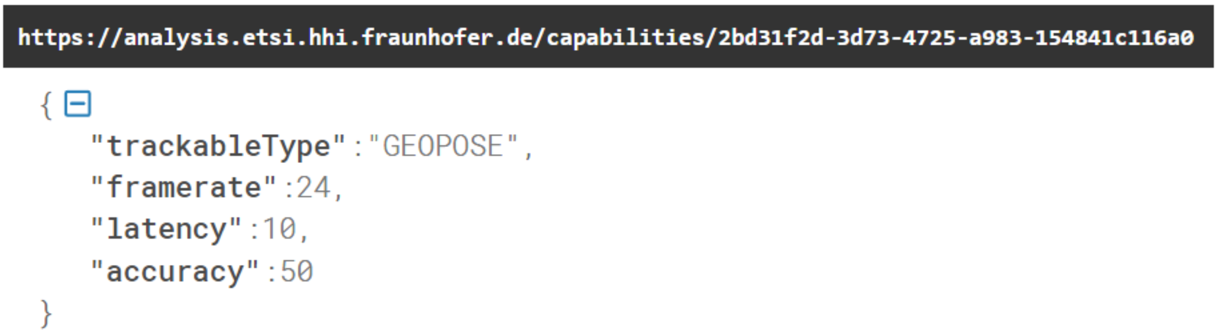
This clause contains examples of requests that can be sent between the Scene Management and the World Analysis. They match the requests exchanged in the context of the use case described in clause A.2. These requests can be implemented through a REST HTTP server or through direct calls (in C++, C# or other) if the World Analysis is instantiated locally on the device.

The following examples of requests rely on an HTTP server implementing the World Analysis API.

After using the World Storage API for instantiating the World Graph shown in Figure A.1, the author will setup an AR scene (with a dedicated tool such as Unity or in a 3D file such as a glTF file) where the 3D content references the World Anchors and the Trackables of the World Graph.

1) The Scene Management requests the capabilities associated with the Trackables and World Anchors referenced by the AR Scene Graph (based on their uuids).

When the application starts, a request is sent per UUID found in the AR Scene Graph. Below in Figure A.2 is an example of REST request. The request concerns the capabilities associated with the "Outside Building" World Anchor. Here the World Analysis only supports GeoTrackables so that a Mesh Trackable cannot be used to locate the World Anchor.



```
https://analysis.etsi.hhi.fraunhofer.de/capabilities/2bd31f2d-3d73-4725-a983-154841c116a0
```

```
{ ⊟
   "trackableType":"GEOPOSE",
   "framerate":24,
   "latency":10,
   "accuracy":50
}
```

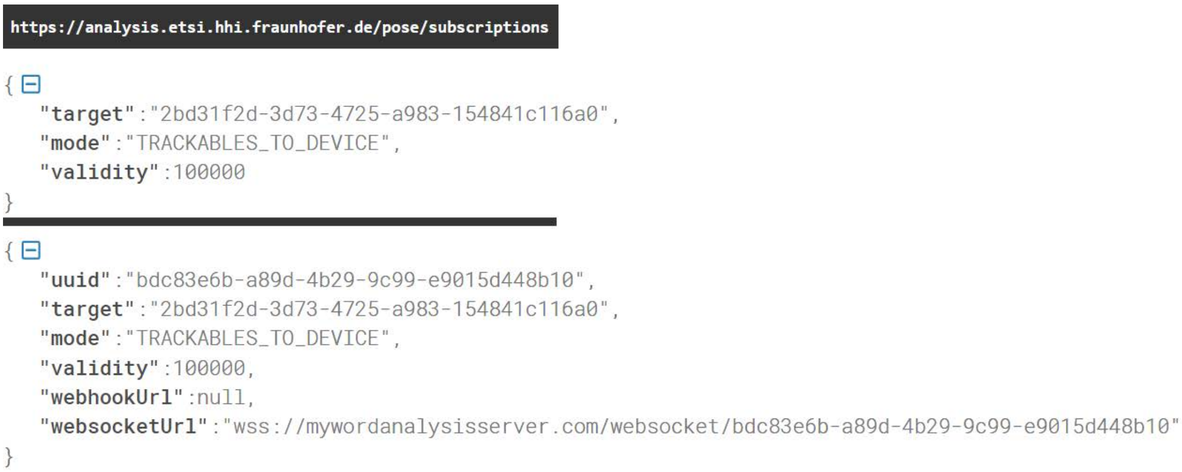
Figure A.2: GET request sent to a World Analysis server and response body

The top part of Figure A.2 is the URL on which the request is sent containing the UUID of the reference World Anchor. Below is the response body, a JSON containing the list of Trackables (and associated properties) that can be used by the World Analysis to estimate the pose estimation of the World Anchor.

Such kind of information can for example be used to disable some content (if no Trackables is supported for estimating the pose of a given World Graph) or for giving instructions to the user about what they have to do to help the pose estimation process. Alternatively, it is possible to request all the capabilities supported by the World Analysis Server with the following URL: **http(s)://{servername}:{port}/capabilities**

2) The Scene Management registers to the pose of all the Trackables and World Anchors referenced by the AR Scene Graph.

After accessing the capabilities, a request is sent per UUID to subscribe to all the pose updates of the Trackables and World Anchors that are found in the AR Scene Graph. Below, in Figure A.3, is an example of a REST request for such a subscription.



```
https://analysis.etsi.hhi.fraunhofer.de/pose/subscriptions
```

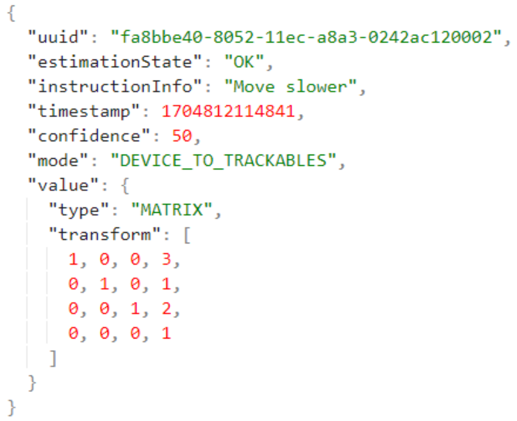
```
{ ⊟
   "target":"2bd31f2d-3d73-4725-a983-154841c116a0",
   "mode":"TRACKABLES_TO_DEVICE",
   "validity":100000
}
```

```
{ ⊟
   "uuid":"bdc83e6b-a89d-4b29-9c99-e9015d448b10",
   "target":"2bd31f2d-3d73-4725-a983-154841c116a0",
   "mode":"TRACKABLES_TO_DEVICE",
   "validity":100000,
   "webhookUrl":null,
   "websocketUrl":"wss://mywordanalysisserver.com/websocket/bdc83e6b-a89d-4b29-9c99-e9015d448b10"
}
```

Figure A.3: POST Request sent to a World Analysis server with the request body and the response body

The top part of Figure A.3 is the URL to which the request is sent, and the part below is the body of the request, a JSON containing all the parameters of the subscription (validity, mode, target uuid). The body can also contain an HTTP(S) URL to which the pose can be sent. In case of success on the server's side the response seen on the bottom of Figure A.3 contains the parameters of the subscription and its UUID. If no URL was provided in the request body, the URL of a WebSocket server is included in the response body. This server will send the pose updates to the Scene Management in a regular interval and upon changes. The communication with the WebSocket may differ from the standard API REST endpoints. In some case, the server may ask the client to first register, with a proprietary protocol, so it can deal with various clients and send them different results. The WebSocket protocol uses the same API objects (schemas) as the REST endpoints, in the JSON string format. It the client proposes a Webhook service then the WA server has to connect itself to this endpoint.

Figure A.4 shows an example of pose sent by the World Analysis to the Scene Management.



```
{
  "uuid": "fa8bbe40-8052-11ec-a8a3-0242ac120002",
  "estimationState": "OK",
  "instructionInfo": "Move slower",
  "timestamp": 1704812114841,
  "confidence": 50,
  "mode": "DEVICE_TO_TRACKABLES",
  "value": {
    "type": "MATRIX",
    "transform": [
      1, 0, 0, 3,
      0, 1, 0, 1,
      0, 0, 1, 2,
      0, 0, 0, 1
    ]
  }
}
```

Figure A.4: Example of Pose that can be sent by the WA

Alternatively, multiple pose subscriptions can be done with a single request with a **POST** request to the following url: **http://{servername}:{port}/pose/subscriptions**

In the same way, the Scene Management can directly request the last estimated pose of a given Trackable or Anchor with a **GET** request to the following url: **http://{servername}:{port}/pose/{TrackableOrWorldAnchorUUID}**

3) The Scene Management can cancel a subscription to a Trackable or a World Anchor.

When the end user wants to end their AR experience or in the case where no AR assets are associated anymore to a given Trackable or Anchor, the Scene Management can request the World Analysis to end a subscription. This can be done with a **DELETE** request to the following url: **http://{servername}:{port}/pose/ subscriptions /{subscriptionUUID}**

The server can then close the WebSockets connection or end the sending of data to the Webhook service.

History

Document history		
V1.1.1	December 2024	Publication