**ISO/IEC 23090-39:2024**

ISO/IEC JTC 1/SC 29/WG 03

Secretariat: XXXX

**Information technology — Coded representation of immersive media — Part 39: Avatar Representation Format**

WD stage

**Warning for WDs and CDs**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office

CP 401 • Ch. de Blandonnet 8

CH-1214 Vernier, Geneva

Phone: +41 22 749 01 11

Fax: +41 22 749 09 47

Email: copyright@iso.org

Website: www.iso.org

Published in Switzerland

Contents

[Foreword 4](#_Toc185236834)

[Introduction 5](#_Toc185236835)

[1 Scope 1](#_Toc185236836)

[2 Normative references 1](#_Toc185236837)

[3 Terms and definitions 1](#_Toc185236838)

[4 Abbreviated terms 1](#_Toc185236839)

[5 System Description 2](#_Toc185236840)

[5.1 Overview 2](#_Toc185236841)

[6 Avatar Data Model 3](#_Toc185236842)

[7 Avatar Document 3](#_Toc185236843)

[7.1 General 3](#_Toc185236844)

[7.2 Preamble 4](#_Toc185236845)

[7.2.1 Authentication Features 4](#_Toc185236846)

[7.2.2 Supported Animation 5](#_Toc185236847)

[7.3 Metadata 5](#_Toc185236848)

[7.4 Structure 6](#_Toc185236849)

[7.4.1 Asset 6](#_Toc185236850)

[7.4.2 Level of Detail (LOD) 6](#_Toc185236851)

[7.5 Components 6](#_Toc185236852)

[7.5.1 Skeleton 7](#_Toc185236853)

[7.5.2 Skin 7](#_Toc185236854)

[7.5.3 Mesh 7](#_Toc185236855)

[7.5.4 Blendshapes 8](#_Toc185236858)

[7.5.5 Node 8](#_Toc185236861)

[7.6 Data 8](#_Toc185236863)

[7.7 Proprietary Animation 9](#_Toc185236864)

[8 ARF Container Format 9](#_Toc185236865)

[8.1 General 9](#_Toc185236866)

[8.2 ISOBMFF-based container format 10](#_Toc185236867)

[8.3 Zip-based container format 11](#_Toc185236868)

[9 Animation Stream Format 12](#_Toc185236870)

[9.1 General 12](#_Toc185236871)

[9.2 Facial Animation Sample Format 12](#_Toc185236872)

[9.3 Joint Animation Sample Format 13](#_Toc185236873)

[Annex A (normative) ARF Document JSON Schema 14](#_Toc185236874)

[Annex B (normative) Integration into Scene Description 23](#_Toc185236875)

[Bibliography 24](#_Toc185236876)

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](https://www.iso.org/directives-and-policies.html) or [www.iec.ch/members\_experts/refdocs](https://www.iec.ch/members_experts/refdocs)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](https://www.iso.org/iso-standards-and-patents.html)) or the IEC list of patent declarations received (see <https://patents.iec.ch>).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see [www.iso.org/iso/foreword.html](https://www.iso.org/iso/foreword.html). In the IEC, see [www.iec.ch/understanding-standards](https://www.iec.ch/understanding-standards).

This document was prepared by Technical Committee ISO/IEC/JTC 1 *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO 23090 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](https://www.iso.org/members.html) and [www.iec.ch/national-committees](https://www.iec.ch/national-committees).

Introduction

This document defines the MPEG Avatar Representation Format (ARF). It provides a data model for the MPEG Avatar Representation Format, a data document that describe the components of an ARF base avatar model, several container formats for storage and distribution, animation sample formats for transmission of animation parameters, and a binary format for the streaming of the MPEG Avatar Representation Format.

Information technology — Coded representation of immersive media — Part 39: MPEG Avatar Representation Format

# Scope

This document specifies the MPEG Avatar Representation Format (ARF) with the goal of offering an interoperable exchange format for the storage and animation of 3D avatars.

# Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document.

* ISO/IEC 23090-14, Information technology — MPEG-I Scene Description

# Terms and definitions

The following terms and definitions are listed in this document.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at [https://www.iso.org/obp](https://www.iso.org/obp/ui)

— IEC Electropedia: available at <https://www.electropedia.org/>

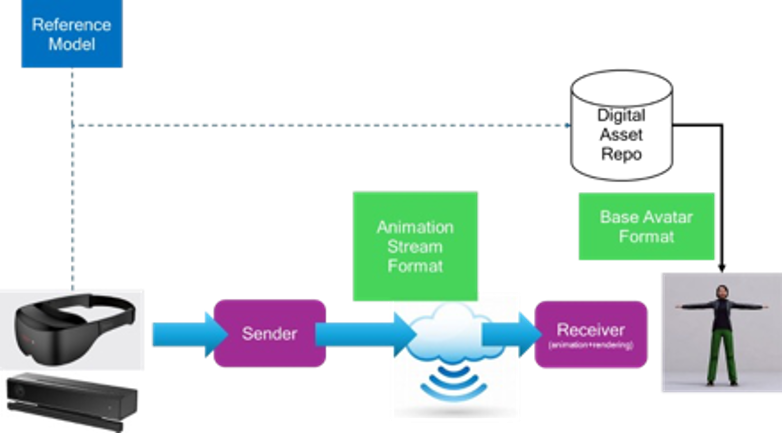
# Abbreviated terms

ISOBMFF ISO base media file format

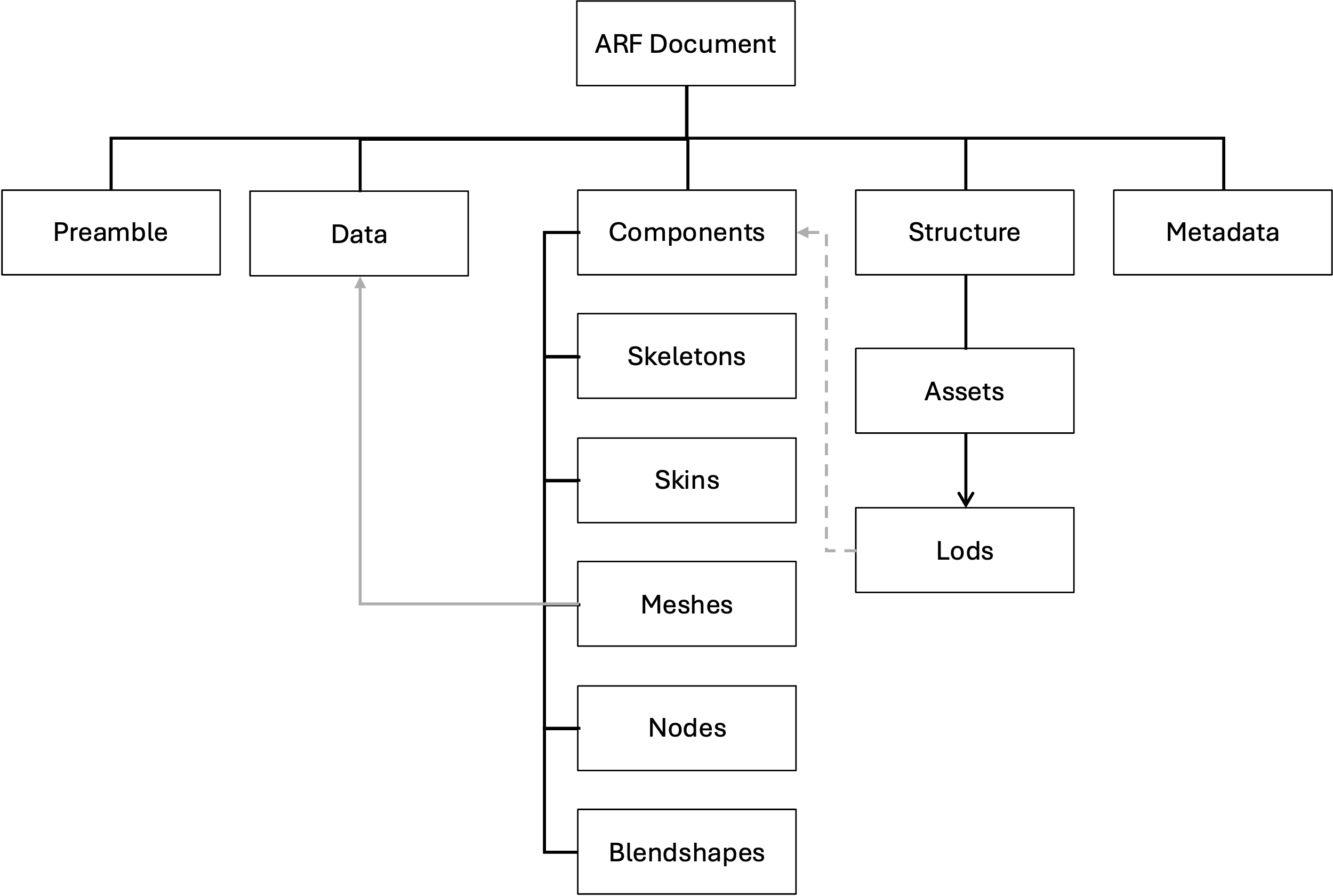
|  |  |
| --- | --- |
| Avatar | 3D graphics-based representation of a user |
| Animation Data | skeletal, blendshape set, and other animation-related information |
| Animation Streams | timed animation data used to animate the base avatar |
| Base avatar model | personalized and animatable 3D model of the user |
| Blendshape | displacements and/or variations of the based avatar model to express key-frame animations |
| LBS | Linear Blend Skinning |
| ARF | Avatar Representation Format |
| ARF container | container that includes all components of the base avatar model, its associated digital assets, and the related metadata |
| ARF document | JSON-formatted document that acts as the entry point to a ARF container |
| Joint | A term used to specify a spatial location of a skeletal joint of the avatar model. |
| Skeleton | A hierarchical representation of joints that are connected with bones to form the skeletal structure of the base avatar model. |

# System Description

## Overview



# Avatar Data Model



# Avatar Document

# General

The ARF document is a JSON-formatted document that describes the user’s base avatar model. The following table describes the high-level component objects of the ARF document.

|  |  |  |  |
| --- | --- | --- | --- |
| ARF Property | Type | Use | Description |
| preamble | Preamble | M | Contains data that uniquely the format and characteristics of the ARF container. |
| metadata | Metadata | M | Contains metadata related to the base avatar model. |
| structure | Structure | M | Contains the data structures of the ARF container. |
| components | Components | M | Contains the core elements of the base avatar model. It lists the main ARF containers to represent and animate the base avatar. |
| data | Data | M | Contains the data for each element of the “components” ARF container. |

# Preamble

The preamble is used to uniquely identify the format and characteristics of the MPEG Avatar Representation Format. It carries a unique signature as well as information about the compatible animation frameworks that work with this base avatar model.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| signature | string | M | Uniquely identifies the ARF. |
| version | string | M | Specifies the version of the MPEG Avatar Representation Format. |
| authentication\_features | array(AuthenticationFeatures) | O | An array of features that are used to identify the owner of this base avatar. |
| supportedAnimation | SupportedAnimation | M | Contains information about the supported animation types. |

# Authentication Features

The authentication features are used to uniquely associate a base avatar model in ARF format to its owner.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| public\_key | URI | M | A URL to the public key that is used to decrypt the features. |
| facial\_feature | string | O | A base64 encoded feature vector of floats. This can be used to match extracted facial features during a communication session. The facial feature shall be encoded with the user’s private key to preserve authenticity. |
| voice\_feature | string | O | A base64 encoded feature vector of floats. This can be used to match extracted voice features during a communication session. The voice feature shall be encoded with the user’s private key to preserve authenticity. |

# Supported Animation

The supported animation identifies the type of animation supported by the avatar format.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| faceAnimation | array(uri) | M | Lists the supported face animation types. Each item in the array is a string representing a supported face animation type.    Each identifier should be formatted as a URN that includes an identifier of the framework, followed by an identifier of the facial blendshape set. An example is: “urn:khronos:openxr:facial-animation:fb-tracking2”. |
| bodyAnimation | array(uri) | M | Lists the supported body animation types. Each item in the array is a string representing a supported body animation type.    Each identifier should be formatted as a URN that includes an identifier of the body animation/tracking framework, followed by an identifier of the body joint set. An example is: “urn:khronos:openxr:body-animation:fb-body”. |
| handAnimation | array(uri) | M | Lists the supported hand animation types. Each item in the array is a string representing a supported hand animation type.      Each identifier should be formatted as a URN that includes an identifier of the body animation/tracking framework, followed by an identifier of the body joint set. An example is: “urn:khronos:openxr:hand-animation:hand”. |

# Metadata

The metadata component contains information about the owner of the base avatar model, some physical characteristics of the base avatar, such as sex, age and height, as well as other metadata related to security and protection of the base avatar model.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| name | string | M | A string that describes the name of the avatar. |
| id | string | M | A string that uniquely identifies the avatar. |
| age | integer | M | An integer value to define the age of the avatar. |
| gender | string | M | A string that describes the gender of the avatar. |

# Structure

The structure component describes the structure of the ARF container. It lists the assets and levels of detail included in this ARF container. It also provides information about the required encryption scheme to decrypt the components of this ARF container that are encrypted.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| assets | Array(Asset) | M | List the assets included in the ARF container. |

### Asset

The assets constitute the key part of the ARF container. An ARF container can contain multiple assets that define the base avatar model of the user or that are associated with it (e.g. digital assets like garments and wearables). Each asset can be accessed and extracted individually.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| name | string | M | The name of the asset. |
| lods | array(LOD) | M | A list of level of details available for this asset in the ARF container. |

### Level of Detail (LOD)

The Level of Detail object provides a link to all components of an asset at a specific level of detail. This should facilitate partial access to the ARF container by allowing to extract the desired assets at the desired level of detail.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| name | string | M | The name of the LOD. |
| skins | array(number) | M | List of references to all skins that are part of this asset. |
| meshes | array(number) | M | List of non-skinned meshes that are part of this asset. |
| skeletons | array(number) | M | List of references to skeletons in the ARF container. |
| blendshapes | array(number) | M | List of references to blendshape sets in the ARF container. |

# Components

The components component is the core of the ARF document. It lists all the components of the ARF container and provides sufficient information to access and use these components for the reconstruction and animation of the base avatar model.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| skeletons | array(Skeleton) | M | A list of skeletons used to describe the avatar skeletal asset. |
| skins | array(Skin) | M | A list of assets that are stored in this ARF container. |
| meshes | array(Mesh) | M | A list of geometries used to describe the avatar asset. |
| nodes | array(Node) | M | A list of nodes used to organize, merge and describe and transform the avatar components. |
| blendshapes | array(Blendshapes) | M | A list of blendshape sets used to describe the avatar animations. |

# Skeleton

The skeleton component describes a partial or complete skeleton that is used in the ARF container. The skeleton describes the joints and their relationships.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| name | string | M | The name of the skeleton. |
| root | number | M | Reference to the root joint for the skeleton in the nodes collection. |
| joints | array(number) | M | List of references to the list of joints in node collection of the ARF container. |
| inverseBindMatrix | number | M | References an item in the data collection of the ARF container that contains the inverse bind matrices for the joints in the same order as the joints. |

# Skin

The skin component is a skinned mesh representing a part of the Avatar body or an associated digital asset. A skin defines the mapping between a mesh and a skeleton, enabling mesh deformation through a skeletal animation system.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| name | string | M | The name of the skin. |
| mapping | string | M | this contains a path indicator that can be used to assign this skinned mesh to a particular node in the scene graph. |
| skeleton | number | M | a reference to the skeleton. |
| mesh | number | M | a reference to the mesh of the skin. |
| weights | number | M | reference to an item in the data collection that contains the weights associated with every joint of the skeleton. |

# Mesh

The component mesh defines the 3D geometrical primitive of the avatar containing its topology and 3D shape.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| name | string | M | The name of the mesh. |
| id | number | M | The identifier of the mesh. |
| path | string | M | A string that represents a hierarchical path that can be used to associate the mesh with a node in the external scene graph. |
| data | array(Data) | M | A reference into a data item that contains the mesh data. |

# Blendshapes

The blendshapes component defines a set of shapes that deform a given basis mesh.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| name | string | M | The name of the blendshape set. |
| id | number | M | A unique identifier of the blendshape set. This id is used in the facial animation to associate the weights with the shapes. |
| shapes | array(number) | M | An array of references to data items that contain each blendshape’s data. |
| basisMesh | number | M | A reference to a data item that contains the basis mesh for this blendshape set. |

# Node

The node component defines the skeletal joints hierarchy and structure for the ARF container. Each skeleton in the ARF container makes reference to a set of nodes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Use** | **Description** |
| name | string | M | The name of the node. |
| mapping | string | M | The joint type or semantics e.g., “full\_body/upper\_body/right\_arm”. |
| parent | number | O | If present, the identifier of the parent node of this node. This attribute shall be present for all nodes, except for the root. |
| children | array | O | if present, a list of identifiers of the children nodes of this node. |
| scale | array(number) | O | The node’s non-uniform scale, given as the scaling factors along the x,y and z axes. |
| rotation | array(number) | O | The node’s unit quaternion rotation in the order (x,y,z,w), where w is the scalar. |
| translation | array(number) | O | The node’s translation along the x,y and z axes. |
| transform | array(number) | M | Provides a 4x4 transformation matrix for the node to define its position and orientation. |

# Data

The data component contains low-level content of the ARF container e.g., meshes, tensors, images, or other data. Each data component may be compressed and/or encrypted.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |
| name | string | M | a string that defines the name of this data. |
| type | string | M | a string that provides the mime type of the data. |
| uri | string | M | a string that defines the data content or reference to the data content depending on type. |
| offset | integer | O | defines the number of bytes used as offset into the data content as pointed to by uri. |
| byteLength | integer | O | defines the number of bytes to use in data content. |
| compression | string | O | an identifier of the compressor used to compress this LoD representation of the mesh. |
| protection | number | O | an identifier of the protection configuration that is applied to encrypt this LoD representation of the mesh. |

# Proprietary Animation

The component proprietary animation provides information on how to use external models to reconstruct or animation the ARF container.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Use | Description |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| proprietary\_animation | | object | O | This object may provide information about an ML-based proprietary model for reconstruction and animation of the user’s avatar. |
|  | id | number | M | A unique identifier of this proprietary animation scheme. |
|  | scheme | URI | M | A vendor-specific URN to identify the proprietary reconstruction and animation scheme. |
|  | items | array(number) | M | A list of data item references, e.g. pretrained models or model weights, that are used by this proprietary reconstruction and animation scheme. |

# ARF Container Format

# General

The ARF container is a key component of the MPEG Avatar Representation Format (ARF), which is designed to facilitate efficient and flexible avatar representation and transmission in communication and shared space sessions. It acts as a structured repository for all the elements that constitute the user’s base avatar model, thus enabling seamless integration and animation across platforms and applications.

The ARF document as defined in clause 4 shall be marked as the entry point to the ARF container. The ARF document describes all the components that make up the user’s base avatar model. All components that are described by the ARF document shall be stroed in the ARF container and the addressing scheme shall allow for locating these components within the ARF container.

A key feature of the ARF container format is its support for partial access. This means that depending on the specific requirements of the application or on the network conditions, only a subset of the user’s base avatar components need to be downloaded. The selection of the components is based on factors like the desired level of detail (LoD), the target bitrate, the user’s selection (e.g. the skinned meshes that represent garments).

The ARF container format plays a crucial role in enabling real-time avatar-based communication and shared experiences. By providing a standardized and interoperable way to store and transmit avatar data, it streamlines the process of sharing and animating avatars across different platforms and applications. In a typical scenario, a user would first create and upload their base avatar model to a central server. When participating in a communication or shared experience session, the user's avatar information, including the location of the ARF container, is shared with other participants. Based on the received information and the negotiated access level, the other participants can then download the container with only the necessary/authorized components of the user's avatar and animate it in real time using the transmitted animation streams.

In this specification, we define two ARF container formats for the storage of the user’s base avatar model. The first one is ISOBMFF-based and the second is Zip-based.

# ISOBMFF-based container format

ISO/IEC 14496-12 defines the concept of brands, which may be indicated in the FileTypeBox.

This specification defines the following brands:

|  |  |  |
| --- | --- | --- |
| Brand | Description | Compatibility Level |
| ARF | file level non-timed metadata items | every ISOBMFF-based container shall declare ARF as the major brand. |
| maas | ARF + timed animation streams | Files that contain stored animation streams shall declare maas among their compatibility brands. |

When stored in an ISOBMFF-based container, the user’s base model shall be stored as metadata items with the MetaBox being declared at the file level. A PrimaryItemBox shall be present and shall contain the item identifier of the item that contains the ARF document.

The following shall apply:

* The HandlerBox shall have a handler\_type set to ‘**ARF** ’
* The primary item shall declare content\_type of “**model/ARF+json**”
* It may contain an item protection box that defines the encryption for the components of the base avatar model that are protected.
* each component of the base avatar model, including the different LoD variants,  shall be stored as an independent item.

When animation streams are also stored as part of the ARF container, at least one metadata track shall be present in the file and shall carry the avatar animation samples. The following requirements shall be fulfilled:

* ‘meta’ handler type shall be used in the HandlerBox of the MediaBox
* The sample entry format shall be ‘**urim**’
* Independent animation samples shall be marked as sync samples
* The URI identifying the type of the metadata is ‘**urn:mpeg:avatar:animation**’

The sample entry for the animation timed metadata track shall be as follows:

|  |
| --- |
| aligned(8) class AvatarAnimationSampleEntry() extends DataSampleEntry('anim') {  unsigned int(3) precision;  bits(5) reserved;  unit(8)[256] animation\_profile;  float timescale;  uint(8) avatar\_id;  uint(8) lod\_id;  } |

* precision – specifies the length in bytes of the correspondence values within each sample. The value of precision shall be greater than 0 and smaller or equal to 4.
* animation\_profile: is a character string with the name of the profile that generated stream conforms to.
* timescale: is the number of ticks per second.
* control\_precision\_minus1: plus 1 specifies the size in bytes of the target avatar index in control AAUs. The value of this field shall be greater than 0 and smaller than or equal to 3.
* avatar\_id: is an integer identifying the avatar to animate.
* lod\_id: is an integer identifying the level-of-detail (LoD) of the avatar to animate.

Samples may be grouped to indicate a sequence of associated animation codes that are stored and ready for playback. The sample group shall be signaled using the group type ‘**aasq**’. Each animation sample group shall have a description about the pre-stored animation sequence, e.g. “smile”, “dance”.

The sample format for an animation sample is defined in clause 6.

# Zip-based container format

An alternative to the ISOBMFF-based container format is the zip-based container format. A Zip container shall be formatted according to ISO/IEC 21320-1. All components of the base avatar model shall be included in the Zip file. The references to these components shall be relative to the location of the ARF document. The ARF document shall be in the root folder of the Zip container and shall be named ***arf.json***.

If present, animation sequences shall be stored as individual binary files  with file extension “**.bin**” under a folder named “animations”. The format of each of these animation files shall be as follows:

|  |  |
| --- | --- |
| animation\_file() { | Descriptor |
| num\_animation\_sequences | int(16) |
| for(i=0;i<num\_animation\_sequences;i++) { |  |
| num\_chars\_in\_description | int(16) |
| description[num\_chars\_in\_description] | b(8) |
| num\_facial\_animations | int(16) |
| for(j=0;j<num\_facial\_animations;j++) { |  |
| facial\_animation\_sample | See clause 6 |
| } |  |
| num\_body\_animations | int(16) |
| for(j=0;j<num\_facial\_animations;j++) { |  |
| body\_animation\_sample | See clause 6 |
| } |  |
| num\_hand\_animations | int(16) |
| for(j=0;j<num\_facial\_animations;j++) { |  |
| hand\_animation\_sample | See clause 6 |
| } |  |
| } |  |

# Animation Stream Format

## 9.1 General

## 9.2 Facial Animation Sample Format

The facial animation sample shall follow the format specified in the following table:

|  |  |
| --- | --- |
| facial\_animation\_sample() { | Descriptor |
| timestamp | int(64) |
| blendshape\_set\_id | int(16) |
| confidence\_present | int(1) |
| reserved | int(7) |
| num\_blendshapes | int(16) |
| for(i=0;i<num\_blendshapes;i++) { |  |
| blendshape\_id | int(16) |
| weight | float(32) |
| if (confidence\_present) { |  |
| confidence | float(32) |
| } |  |
| } |  |
| } |  |

## 9.3 Joint Animation Sample Format

The joint animation sample shall follow the format specified in the following table:

|  |  |
| --- | --- |
| joint\_animation\_sample() { | Descriptor |
| timestamp | int(64) |
| joint\_set\_id | int(16) |
| velocity\_present | int(1) |
| reserved | int(7) |
| num\_joints | int(16) |
| for(i=0;i<num\_joints;i++) { |  |
| location\_matrix[16] | float(32) |
| if (velocity\_present) { |  |
| velocity\_matrix[16] | float(32) |
| } |  |
| } |  |
| } |  |

1. (normative)  
     
   ARF Document JSON Schema

The following table contains the JSON Schema for the ARF document.

|  |
| --- |
| {    "$schema": "http://json-schema.org/draft-07/schema#",    "type": "object",    "title": "ARF Container Schema",    "required": [      "preamble",      "metadata",      "structure",      "components",      "data"    ],    "properties": {      "preamble": {        "$ref": "arf-preamble.schema.json",        "description": "Contains data that uniquely the format and characteristics of the ARF container"      },      "metadata": {        "$ref": "arf-metadata.schema.json",        "description": "Contains metadata related to the base avatar model"      },      "structure": {        "$ref": "arf-structure.schema.json",        "description": "Contains the data structures of the ARF container"      },      "components": {        "$ref": "arf-components.schema.json",        "description": "Contains the core elements of the base avatar model. It lists the main ARF containers to represent and animate the base avatar"      },      "data": {        "$ref": "arf-data.schema.json",        "description": "Contains the data for each element of the 'components' ARF container"      }    }  } |

The schema for Preamble is provided in the following table:

|  |
| --- |
| {      "$schema": "http://json-schema.org/draft-07/schema#",      "type": "object",      "title": "Preamble Schema",      "required": [          "signature",          "version",          "supportedAnimation"      ],      "properties": {          "signature": {              "type": "string",              "description": "Uniquely identifies the ARF"          },          "version": {              "type": "string",              "description": "Specifies the version of the MPEG Avatar Representation Format"          },          "authentication\_features": {              "type": "array",              "description": "An array of features that are used to identify the owner of this base avatar",              "items": {                  "$ref": "#/components/schemas/AuthenticationFeatures"              }          },          "supportedAnimation": {              "$ref": "#/components/schemas/SupportedAnimation"          }      },      "components": {          "schemas": {              "AuthenticationFeatures": {                  "type": "object",                  "required": [                      "public\_key"                  ],                  "properties": {                      "public\_key": {                          "type": "string",                          "format": "uri",                          "description": "A URL to the public key that is used to decrypt the features"                      },                      "facial\_feature": {                          "type": "string",                          "format": "base64",                          "description": "A base64 encoded feature vector of floats. This can be used to match extracted facial features during a communication session. The facial feature shall be encoded with the user's private key to preserve authenticity"                      },                      "voice\_feature": {                          "type": "string",                          "format": "base64",                          "description": "A base64 encoded feature vector of floats. This can be used to match extracted voice features during a communication session. The voice feature shall be encoded with the user's private key to preserve authenticity"                      }                  }              },              "SupportedAnimation": {                  "type": "object",                  "required": [                      "public\_key"                  ],                  "properties": {                      "public\_key": {                          "type": "string",                          "format": "uri",                          "description": "A URL to the public key that is used to decrypt the features"                      },                      "facial\_feature": {                          "type": "string",                          "format": "base64",                          "description": "A base64 encoded feature vector of floats. This can be used to match extracted facial features during a communication session. The facial feature shall be encoded with the user's private key to preserve authenticity"                      },                      "voice\_feature": {                          "type": "string",                          "format": "base64",                          "description": "A base64 encoded feature vector of floats. This can be used to match extracted voice features during a communication session. The voice feature shall be encoded with the user's private key to preserve authenticity"                      }                  }              }          }      }  } |

The schema for AuthenticationFeature is provided in the following table:

|  |
| --- |
| {      "$schema": "http://json-schema.org/draft-07/schema#",      "type": "object",      "title": "Authentication Features Schema",      "required": [          "public\_key"      ],      "properties": {          "public\_key": {              "type": "string",              "format": "uri",              "description": "A URL to the public key that is used to decrypt the features"          },          "facial\_feature": {              "type": "string",              "format": "base64",              "description": "A base64 encoded feature vector of floats. This can be used to match extracted facial features during a communication session. The facial feature shall be encoded with the user's private key to preserve authenticity"          },          "voice\_feature": {              "type": "string",              "format": "base64",              "description": "A base64 encoded feature vector of floats. This can be used to match extracted voice features during a communication session. The voice feature shall be encoded with the user's private key to preserve authenticity"          }      }  } |

The schema for Structure is provided in the following table:

|  |
| --- |
| {      "$schema": "http://json-schema.org/draft-07/schema#",      "type": "object",      "title": "Structure Schema",      "required": [          "assets"      ],      "properties": {          "assets": {              "type": "array",              "description": "List the assets included in the ARF container",              "items": {                  "$ref": "#/components/schemas/Asset"              }          }      },      "components": {          "schemas": {              "Asset": {                  "type": "object",                  "title": "Asset",                  "required": [                      "name",                      "lods"                  ],                  "properties": {                      "name": {                          "type": "string",                          "description": "The name of the asset"                      },                      "lods": {                          "type": "array",                          "description": "A list of level of details available for this asset in the ARF container",                          "items": {                              "$ref": "#/components/schemas/LOD"                          }                      }                  }              },              "LOD": {                  "type": "object",                  "title": "LOD",                  "required": [                      "name",                      "skins",                      "meshes",                      "skeletons",                      "blendshapes"                  ],                  "properties": {                      "name": {                          "type": "string",                          "description": "The name of the LOD"                      },                      "skins": {                          "type": "array",                          "description": "List of references to all skins that are part of this asset",                          "items": {                              "type": "number"                          }                      },                      "meshes": {                          "type": "array",                          "description": "List of non-skinned meshes that are part of this asset",                          "items": {                              "type": "number"                          }                      },                      "skeletons": {                          "type": "array",                          "description": "List of references to skeletons in the ARF container",                          "items": {                              "type": "number"                          }                      },                      "blendshapes": {                          "type": "array",                          "description": "List of references to blendshape sets in the ARF container",                          "items": {                              "type": "number"                          }                      }                  }              }          }      }  } |

The schema for Components is provided in the following table:

|  |
| --- |
| {      "$schema": "http://json-schema.org/draft-07/schema#",      "type": "object",      "title": "Components Schema",      "required": [          "skeletons",          "skins",          "meshes",          "nodes",          "blendshapes"      ],      "properties": {          "skeletons": {              "type": "array",              "description": "A list of skeletons used to describe the avatar skeletal asset",              "items": {                  "$ref": "#/components/schemas/Skeleton"              }          },          "skins": {              "type": "array",              "description": "A list of assets that are stored in this ARF container",              "items": {                  "$ref": "#/components/schemas/Skin"              }          },          "meshes": {              "type": "array",              "description": "A list of geometries used to describe the avatar asset",              "items": {                  "$ref": "#/components/schemas/Mesh"              }          },          "nodes": {              "type": "array",              "description": "A list of nodes used to organize, merge and describe and transform the avatar components",              "items": {                  "$ref": "#/components/schemas/Node"              }          },          "blendshapes": {              "type": "array",              "description": "A list of blendshape sets used to describe the avatar animations",              "items": {                  "$ref": "#/components/schemas/Blendshapes"              }          }      },      "components": {          "schemas": {              "Skeleton": {                  "type": "object",                  "required": [                      "name",                      "root",                      "joints",                      "inverseBindMatrix"                  ],                  "properties": {                      "name": {                          "type": "string",                          "description": "The name of the skeleton"                      },                      "root": {                          "type": "number",                          "description": "Reference to the root joint for the skeleton in the nodes collection"                      },                      "joints": {                          "type": "array",                          "items": {                              "type": "number"                          },                          "description": "List of references to the list of joints in node collection of the ARF container"                      },                      "inverseBindMatrix": {                          "type": "number",                          "description": "References an item in the data collection of the ARF container that contains the inverse bind matrices for the joints in the same order as the joints"                      }                  }              },              "Skin": {                  "type": "object",                  "required": [                      "name",                      "mapping",                      "skeleton",                      "mesh",                      "weights"                  ],                  "properties": {                      "name": {                          "type": "string",                          "description": "The name of the skin"                      },                      "mapping": {                          "type": "string",                          "description": "This contains a path indicator that can be used to assign this skinned mesh to a particular node in the scene graph"                      },                      "skeleton": {                          "type": "number",                          "description": "A reference to the skeleton"                      },                      "mesh": {                          "type": "number",                          "description": "A reference to the mesh of the skin"                      },                      "weights": {                          "type": "number",                          "description": "Reference to an item in the data collection that contains the weights associated with every joint of the skeleton"                      }                  }              },              "Mesh": {                  "type": "object",                  "required": [                      "name",                      "id",                      "path",                      "data"                  ],                  "properties": {                      "name": {                          "type": "string",                          "description": "The name of the mesh"                      },                      "id": {                          "type": "number",                          "description": "The identifier of the mesh"                      },                      "path": {                          "type": "string",                          "description": "A string that represents a hierarchical path that can be used to associate the mesh with a node in the external scene graph"                      },                      "data": {                          "type": "array",                          "items": {                              "type": "object",                              "title": "Data"                          },                          "description": "A reference into a data item that contains the mesh data"                      }                  }              },              "Blendshapes": {                  "type": "object",                  "required": [                      "name",                      "id",                      "shapes",                      "basisMesh"                  ],                  "properties": {                      "name": {                          "type": "string",                          "description": "The name of the blendshape set"                      },                      "id": {                          "type": "number",                          "description": "A unique identifier of the blendshape set. This id is used in the facial animation to associate the weights with the shapes"                      },                      "shapes": {                          "type": "array",                          "items": {                              "type": "number"                          },                          "description": "An array of references to data items that contain each blendshape's data"                      },                      "basisMesh": {                          "type": "number",                          "description": "A reference to a data item that contains the basis mesh for this blendshape set"                      }                  }              },              "Node": {                  "type": "object",                  "required": [                      "name",                      "mapping",                      "transform"                  ],                  "properties": {                      "name": {                          "type": "string",                          "description": "The name of the node"                      },                      "mapping": {                          "type": "string",                          "description": "The joint type or semantics e.g., 'full\_body/upper\_body/right\_arm'"                      },                      "parent": {                          "type": "number",                          "description": "If present, the identifier of the parent node of this node. This attribute shall be present for all nodes, except for the root"                      },                      "children": {                          "type": "array",                          "description": "If present, a list of identifiers of the children nodes of this node"                      },                      "scale": {                          "type": "array",                          "items": {                              "type": "number"                          },                          "minItems": 3,                          "maxItems": 3,                          "description": "The node's non-uniform scale, given as the scaling factors along the x,y and z axes"                      },                      "rotation": {                          "type": "array",                          "items": {                              "type": "number"                          },                          "minItems": 4,                          "maxItems": 4,                          "description": "The node's unit quaternion rotation in the order (x,y,z,w), where w is the scalar"                      },                      "translation": {                          "type": "array",                          "items": {                              "type": "number"                          },                          "minItems": 3,                          "maxItems": 3,                          "description": "The node's translation along the x,y and z axes"                      },                      "transform": {                          "type": "array",                          "items": {                              "type": "number"                          },                          "minItems": 16,                          "maxItems": 16,                          "description": "Provides a 4x4 transformation matrix for the node to define its position and orientation"                      }                  }              }          }      }  } |

The Data object is defined in the following JSON schema:

|  |
| --- |
| {      "$schema": "http://json-schema.org/draft-07/schema#",      "type": "object",      "title": "Data Schema",      "required": [          "name",          "type",          "uri"      ],      "properties": {          "name": {              "type": "string",              "description": "A string that defines the name of this data"          },          "type": {              "type": "string",              "description": "A string that provides the mime type of the data"          },          "uri": {              "type": "string",              "description": "A string that defines the data content or reference to the data content depending on type"          },          "offset": {              "type": "integer",              "minimum": 0,              "description": "Defines the number of bytes used as offset into the data content as pointed to by uri"          },          "byteLength": {              "type": "integer",              "minimum": 0,              "description": "Defines the number of bytes to use in data content"          },          "compression": {              "type": "string",              "description": "An identifier of the compressor used to compress this LoD representation of the mesh"          },          "protection": {              "type": "number",              "description": "An identifier of the protection configuration that is applied to encrypt this LoD representation of the mesh"          }      }  } |

The ProprietaryAnimation object has the following JSON schema:

|  |
| --- |
| {      "$schema": "http://json-schema.org/draft-07/schema#",      "type": "object",      "title": "Animation Schema",      "properties": {          "proprietary\_animation": {              "type": "object",              "description": "This object may provide information about an ML-based proprietary model for reconstruction and animation of the user's avatar",              "required": [                  "id",                  "scheme",                  "items"              ],              "properties": {                  "id": {                      "type": "number",                      "description": "A unique identifier of this proprietary animation scheme"                  },                  "scheme": {                      "type": "string",                      "format": "uri",                      "description": "A vendor-specific URN to identify the proprietary reconstruction and animation scheme"                  },                  "items": {                      "type": "array",                      "description": "A list of data item references, e.g. pretrained models or model weights, that are used by this proprietary reconstruction and animation scheme",                      "items": {                          "type": "number"                      }                  }              }          }      }  } |

1. (normative)  
   Integration into Scene Description

**Annex C**

**Reference Avatar Client**

Bibliography

1. Khronos, glTF 2.0