|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding by Artificial Intelligence  www.mpai.community |

|  |  |
| --- | --- |
| **N726** | 2022/06/22 |
| **Source** | Daniele Bortoluzzi, Andrea Basso |
| **Title** | The MPAI Ontology V1.1 |
| **Target** | MPAI Members |

This document provides elements for the MPAI Ontology referenced by the MPAI-AIF Technical Specification V1, specifically, for

1. Resource Policy
2. Authentication
3. Protocol
4. Architecture
5. OS
6. OSVersion

|  |  |
| --- | --- |
| **ResourcePolicy** |  |
| If “Name” == “Memory” then | {  “Name”: “Memory”  “Minimum”: int (bytes)  “Maximum”: int (bytes)  “Request”: int (bytes)  } |
| If “Name” == “CPU:Number” then | {  “Name”: “CPU:Number”  “Minimum”: int (>=0, 0 means “all”)  “Maximum”: int (>=0)  “Request”: int (>=0)  } |
| If “Name” == “CPU:Class” then | {  “Name”: “CPU:Class”  “Minimum”: enum(“Low”, “Medium”, “High”)  “Maximum”: enum(“Low”, “Medium”, “High”)  “Request”: enum(“Low”, “Medium”, “High”)  } |
| If “Name” == “GPU:Number” then | {  “Name”: “GPU:Number”  “Minimum”: int (>=0, 0 means “all”)  “Maximum”: int (>=0)  “Request”: int (>=0)  } |
| If “Name” == “GPU:CUDA:Class” then | {  “Name”: “GPU:CUDA:Class”  “Minimum”: enum(“SM10”, “SM11”, “SM12”, “SM13”, “SM20”, “SM21”, “SM30”, “SM32”, “SM35”, “SM37”, “SM50”, “SM52”, “SM53”, “SM60”, “SM61”, “SM62”, “SM70”, “SM72”, “SM75”, “SM80”, “SM86”, “SM87”, “SM90”)  “Maximum”: enum(“SM10”, “SM11”, “SM12”, “SM13”, “SM20”, “SM21”, “SM30”, “SM32”, “SM35”, “SM37”, “SM50”, “SM52”, “SM53”, “SM60”, “SM61”, “SM62”, “SM70”, “SM72”, “SM75”, “SM80”, “SM86”, “SM87”, “SM90”)  “Request”: enum(“SM10”, “SM11”, “SM12”, “SM13”, “SM20”, “SM21”, “SM30”, “SM32”, “SM35”, “SM37”, “SM50”, “SM52”, “SM53”, “SM60”, “SM61”, “SM62”, “SM70”, “SM72”, “SM75”, “SM80”, “SM86”, “SM87”, “SM90”)  } |
| If “Name” == “GPU:CUDA:FrameBuffer” then | {  “Name”: “GPU:CUDA:FrameBuffer”  “Minimum”: float “GB\_” enum(“GDDR”, “GDDR2”, “GDDR3”, “GDDR4”, “GDDR5”, “GDDR5X”, “GDDR6”, “GDDR6X”)  “Maximum”: float “GB\_” enum(“GDDR”, “GDDR2”, “GDDR3”, “GDDR4”, “GDDR5”, “GDDR5X”, “GDDR6”, “GDDR6X”)  “Request”: float “GB\_” enum(“GDDR”, “GDDR2”, “GDDR3”, “GDDR4”, “GDDR5”, “GDDR5X”, “GDDR6”, “GDDR6X”)  } |
| If “Name” == “GPU:CUDA:MemorySpeed” then | {  “Name”: “GPU:CUDA:MemorySpeed”  “Minimum”: float “GHz”  “Maximum”: float “GHz”  “Request”: float “GHz”  } |
| **Authentication** | enum(“Token Authentication”, “Certificate-based Authentication”, “Password-based Authentication”) |
| **Protocol** | enum(“UDP”, “TCP”, “HTTP”, “HTTPS”, “CoAP”, “CoAPS”, “L2CAP”, “BNEP”, “RFCOMM”, “SDP”, “RPC”, “RTP”, “RTCP”) |
| **Architecture** | enum(“x86”, “x86-64”, “ARM”, “ARM64”, “MIPS”, “RISC-V”) |
| **OS** | enum(“Windows”, “MacOS”, “Linux”, “Android”, “Zephyr”, “RTOS”, “Oniro”, “iOS”) |
| **OSVersion** | {  “MinimumOSver”: string  ) |

**JSON representation**

The ontology representation in JSON is depitched below.

It is articulated in two JSON schemas. AIF-metadata.schema.json and the AIW-AIM-metadata.schema.json

AIF-metadata.schema.json

{

"$schema": "http://json-schema.org/draft-07/schema#",

"$id": "https://mpai.community/standards/MPAI-AIF/V1/AIF-metadata.schema.json",

"title": "MPAI-AIF V1 AIF metadata",

"type": "object",

"properties": {

"ImplementerID": {

"description": "A numeric ID identifying the Implementer. Provided by MPAI Store",

"type": "integer"

},

"Version": {

"description": "Provided by the Implementer. Replaced by '\*' in technical specifications",

"type": "string"

},

"APIProfile": {

"description": "Provided by MPAI. Selected by the Implementer",

"type": "string",

"enum": [

"Base",

"Main",

"High"

]

},

"ResourcePolicies": {

"description": "A set of policies describing computing resources made available to AIWs",

"type": "array",

"items": {

"description": "A policy describing computing resources made available to AIWs",

"type": "object",

"properties": {

"Name": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string"

},

"Minimum": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string"

},

"Maximum": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string"

}

},

"required": [

"Name"

],

"allOf": [

{

"if": {

"properties": {

"Name": {

"const": "Memory"

}

}

},

"then": {

"properties": {

"Minimum": {

"pattern": "[0-9]+"

},

"Maximum": {

"pattern": "[0-9]+"

},

"Request": {

"pattern": "[0-9]+"

}

},

"required": [

"Minimum",

"Maximum",

"Request"

]

}

},

{

"if": {

"properties": {

"Name": {

"const": "CPU:Number"

}

}

},

"then": {

"properties": {

"Minimum": {

"pattern": "[0-9]+"

},

"Maximum": {

"pattern": "[0-9]+"

},

"Request": {

"pattern": "[0-9]+"

}

},

"required": [

"Minimum",

"Maximum",

"Request"

]

}

},

{

"if": {

"properties": {

"Name": {

"const": "CPU:Class"

}

}

},

"then": {

"properties": {

"Minimum": {

"pattern": "Low|Medium|High"

},

"Maximum": {

"pattern": "Low|Medium|High"

},

"Request": {

"pattern": "Low|Medium|High"

}

},

"required": [

"Minimum",

"Maximum",

"Request"

]

}

},

{

"if": {

"properties": {

"Name": {

"const": "GPU:Number"

}

}

},

"then": {

"properties": {

"Minimum": {

"pattern": "[0-9]+"

},

"Maximum": {

"pattern": "[0-9]+"

},

"Request": {

"pattern": "[0-9]+"

}

},

"required": [

"Minimum",

"Maximum",

"Request"

]

}

},

{

"if": {

"properties": {

"Name": {

"const": "GPU:CUDA:Class"

}

}

},

"then": {

"properties": {

"Minimum": {

"pattern": "SM10|SM11|SM12|SM13|SM20|SM21|SM30|SM32|SM35|SM37|SM50|SM52|SM53|SM60|SM61|SM62|SM70|SM72|SM75|SM80|SM86|SM87|SM90"

},

"Maximum": {

"pattern": "SM10|SM11|SM12|SM13|SM20|SM21|SM30|SM32|SM35|SM37|SM50|SM52|SM53|SM60|SM61|SM62|SM70|SM72|SM75|SM80|SM86|SM87|SM90"

},

"Request": {

"pattern": "SM10|SM11|SM12|SM13|SM20|SM21|SM30|SM32|SM35|SM37|SM50|SM52|SM53|SM60|SM61|SM62|SM70|SM72|SM75|SM80|SM86|SM87|SM90"

}

},

"required": [

"Minimum",

"Maximum",

"Request"

]

}

},

{

"if": {

"properties": {

"Name": {

"const": "GPU:CUDA:FrameBuffer"

}

}

},

"then": {

"properties": {

"Minimum": {

"pattern": "[0-9]+\_[GDDR|GDDR2|GDDR3|GDDR4|GDDR5|GDDR5X|GDDR6|GDDR6X]"

},

"Maximum": {

"pattern": "[0-9]+\_[GDDR|GDDR2|GDDR3|GDDR4|GDDR5|GDDR5X|GDDR6|GDDR6X]"

},

"Request": {

"pattern": "[0-9]+\_[GDDR|GDDR2|GDDR3|GDDR4|GDDR5|GDDR5X|GDDR6|GDDR6X]"

}

},

"required": [

"Minimum",

"Maximum",

"Request"

]

}

},

{

"if": {

"properties": {

"Name": {

"const": "GPU:CUDA:MemorySpeed"

}

}

},

"then": {

"properties": {

"Minimum": {

"pattern": "[+-]?([0-9]\*[.])?[0-9]+"

},

"Maximum": {

"pattern": "[+-]?([0-9]\*[.])?[0-9]+"

},

"Request": {

"pattern": "[+-]?([0-9]\*[.])?[0-9]+"

}

},

"required": [

"Minimum",

"Maximum",

"Request"

]

}

}

]

}

},

"Authentication": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string",

"enum": [

"Token Authentication",

"Certificate-based Authentication",

"Password-based Authentication"

]

},

"TimeBase": {

"description": "A protocol providing a time base. If absent, timestamps are interpreted according to the host time clock (absolute time with the appropriate timescale conversion)",

"type": "string",

"enum": [

"NTP",

"RTP",

"RTCP"

]

}

},

"required": [

"ImplementerID",

"Version",

"Authentication"

]

}

AIW-AIM-metadata.schema.json

{

"$schema": "http://json-schema.org/draft-07/schema#",

"$id": "https://mpai.community/standards/MPAI-AIF/V1/AIW-AIM-metadata.schema.json",

"id": "#root",

"title": "MPAI-AIF V1 AIW/AIM metadata",

"type": "object",

"properties": {

"Identifier": {

"id": "#identifier",

"description": "Information uniquely identifying an AIW/AIM implementation",

"type": "object",

"properties": {

"ImplementerID": {

"description": "A numeric ID identifying the Implementer. Provided by MPAI Store",

"type": "integer"

},

"Specification": {

"oneOf": [

{

"description": "An AIW/AIM defined by an MPAI standard",

"type": "object",

"properties": {

"Standard": {

"description": "Defined by the Standard",

"type": "string"

},

"AIW": {

"description": "Defined by the Standard",

"type": "string"

},

"AIM": {

"description": "Same as AIW when the Metadata being defined describes the AIW, otherwise the name of the AIM as defined by the Standard",

"type": "string"

},

"Version": {

"description": "Defined by the Standard",

"type": "string"

},

"Profile": {

"description": "Provided by MPAI. Selected by the Implementer",

"type": "array",

"items": {

"type": "string",

"enum": [

"Base",

"Main",

"High"

]

}

}

},

"required": [

"Standard",

"AIW",

"AIM",

"Version"

]

},

{

"description": "An AIW/AIM defined by an Implementer",

"type": "object",

"properties": {

"Name": {

"description": "Provided by the Implementer",

"type": "string"

},

"Version": {

"description": "Provided by the Implementer",

"type": "string"

}

},

"required": [

"Name",

"Version"

]

}

]

}

},

"required": [

"ImplementerID",

"Specification"

]

},

"APIProfile": {

"description": "Provided by MPAI. Selected by the Implementer",

"type": "string",

"enum": [

"Base",

"Main",

"High"

]

},

"Description": {

"description": "Free text describing the AIM",

"type": "string"

},

"Types": {

"description": "A list of shorthands for Channel data types, defined according to 6.1.1",

"type": "array",

"items": {

"description": "A shorthand for a Channel data type, defined according to 6.1.1",

"type": "object",

"properties": {

"Name": {

"description": "The unique shorthand used for a Channel data type",

"type": "string"

},

"Type": {

"description": "A Channel data type, defined according to 6.1.1",

"type": "string"

}

},

"required": [

"Name",

"Type"

]

}

},

"Ports": {

"description": "A list of AIM Ports",

"type": "array",

"items": {

"description": "A Port, i.e., a physical or logical interface through which the AIM communicates",

"type": "object",

"properties": {

"Name": {

"description": "Implementer-defined name",

"type": "string"

},

"Direction": {

"description": "The direction of the communication flow",

"type": "string",

"enum": [

"OutputInput",

"InputOutput"

]

},

"RecordType": {

"description": "Port data type defined either in the dictionary Types, or according to Section 6.1.1",

"type": "string"

},

"Technology": {

"description": "Whether the Port is implemented in hardware or software",

"type": "string",

"enum": [

"Hardware",

"Software"

]

},

"Protocol": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string",

"enum": [

"UDP",

"TCP",

"HTTP",

"HTTPS",

"CoAP",

"CoAPS",

"L2CAP",

"BNEP",

"RFCOMM",

"SDP",

"RPC",

"RTP",

"RTCP",

]

},

"IsRemote": {

"description": "Boolean specifying whether the port is remote",

"type": "boolean"

}

},

"required": [

"Name",

"Direction",

"RecordType",

"Technology",

"Protocol",

"IsRemote"

]

}

},

"SubAIMs": {

"description": "A list of AIMs in terms of which the current AIM is defined",

"type": "array",

"items": {

"description": "One of the AIMs in terms of which the current AIM is defined",

"type": "object",

"properties": {

"Name": {

"description": "A unique shorthand for the AIM in terms of which the current AIM is defined",

"type": "string"

},

"Identifier": {

"$ref": "#identifier"

}

},

"required": [

"Name",

"Identifier"

]

}

},

"Topology": {

"description": "A list of Channels connecting one Output to one Input Port",

"type": "array",

"items": {

"description": "A Channel connecting one Output to one Input Port",

"type": "object",

"properties": {

"Output": {

"id": "#portID",

"description": "A Port identifier",

"type": "object",

"properties": {

"AIMName": {

"description": "The unique shorthand for a SubAIM",

"type": "string"

},

"PortName": {

"description": "The unique shorthand for one of the SubAIM Ports",

"type": "string"

}

},

"required": [

"AIMName",

"PortName"

]

},

"Input": {

"$ref": "#portID"

}

},

"required": [

"Output",

"Input"

]

}

},

"Implementations": {

"description": "A list of Implementations for the AIM being defined",

"type": "array",

"items": {

"description": "An Implementation for the AIM being defined",

"type": "object",

"properties": {

"BinaryName": {

"description": "Specifies an entry in the archive containing the Implementation downloaded from the MPAI store",

"type": "string"

},

"Architecture": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string",

"enum": [

"x86",

"x86-64",

"ARM",

"ARM64",

"MIPS",

"RISC-V",

]

},

"OperatingSystem": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string",

"enum": [

"Windows",

"MacOS",

"Linux",

"Android",

"Zephyr",

"RTOS",

"Oniro",

"iOS"

]

},

"Version": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string"

},

"OSVersion": {

"type": "object",

"properties": {

"MinimumOSver": {

"description": "Minimum version of OS supported",

"type": "integer"

},

}

},

"Source": {

"description": "Where the AIM Implementation should be found",

"type": "string",

"enum": [

"AIMStorage",

"MPAIStore"

]

},

"Destination": {

"description": "If empty, the Implementation is executed locally. Otherwise, the string must be a valid URI of an MPAI Server",

"type": "string"

}

},

"required": [

"BinaryName",

"Architecture",

"OperatingSystem",

"Version",

"Source",

"Destination"

]

}

},

"ResourcePolicies": {

"description": "A set of policies describing computing resources needed by the AIW/AIF being defined",

"type": "array",

"items": {

"description": "A policy describing computing resources needed by the AIW/AIF being defined",

"type": "object",

"properties": {

"Name": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string"

},

"Minimum": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string"

},

"Maximum": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string"

},

"Request": {

"description": "An entry in the MPAI-specified Ontology",

"type": "string"

}

},

"required": [

"Name"

]

}

},

"Documentation": {

"definition": "A list of references to documents specifying information relevant to the design, implementation and usage of the AIM being defined",

"type": "array",

"items": {

"description": "A reference to a document specifying information relevant to the design, implementation and usage of the AIM being defined",

"type": "object",

"properties": {

"Type": {

"description": "The type of the document",

"type": "string",

"enum": [

"Specification",

"Manual",

"Tutorial",

"Video"

]

},

"URI": {

"description": "A valid URI for the document",

"type": "string"

}

}

}

}

},

"required": [

"Identifier",

"Ports",

"SubAIMs",

"Topology",

"Implementations"

]

}