**Explanation of Digital Twin Network Data Model**

This section describes the Schema of the proposed DTN data model to represent the a data center in the Edge-Cloud Continuum. All the entities of the data center jointly represent the: physical and the software components. The proposed data model follows a bottom-up approach (for instance, starting the representation of the smallest virtual component, called container, to the level of Cluster or cite), is able to fully represent the Edge-Cloud Continuum. The under-consideration cluster is managed and orchestrated through Kubernetes in any active site/domain. This domain may reside in any Edge (micro data center), Cloud data center, or in private organization premises.

The below schema specification also explains how some of the attributes identify the relation to other entities (components of a cluster) and refer them through meaningful additional attributes. For instance, the container will be always deployed on the top of the computational node, so there is a reference attribute in the container schema, and vice versa, to record this relation.

In following entities, schema, and relations are described one-by-one.

* ***Computational Node***

Link:<https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/computational_node.json>

The computational Node is the main entity that supports the computation of various sorts of workloads in data centers. The hardware specifications such as CPU core, storage capacity, I/o speed, etc. represent the capabilities. The other identifiers of a node are its location, vendor, operating system, and the date of integration or removal for the data center assets. This schema can characterize any node across the continuum (i.e., node residing in the Cloud or the Edge data center) through the range of specified attributes, for both purposes, identification and metric calibration. The metrics include the consumptions (bandwidth, energy, CO2 emission), and latency of the node software stack.

Note that in the node schema, there are references of the five schemas further. These references identify the deployed containers, the room in which the node exists, the rack, and the other computational node to which the target node has a dependency relation.

"deployedContainers": {

"type": "array",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/container.json"

},

"description": "a reference to the identification of the deployed container in this computational node [reference to the container schema in Edge-Cloud Continuum Model]"

},

"roomId": {

"type": ["string", "null"],

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Room.json"

},

"description": "The identifier of the Room hosting the computational node (in the case of a server), null in the case if the computational node is IoT object."

},

"rackNumber": {

"type": "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Rack.json"

},

"description": "The rack number on which the node is plugged in a rack[reference to the rack JSON schema]"

}

"physicalDependencies": {

"type": "array",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/computational\_node.json"

},

"description": "List of nodes with which the node has dependencies. reference to the schema of the computational itself. The instances, (technically a tuple in data characterizing the node) can use this schema"

}

"deployedServices": {

"type": "array",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/service.json"

},

"description": "List of services located at the node level."

}

* ***Rack***

Link : <https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Rack.json>

This JSON schema defines the structure for representing a rack in a data model. It specifies properties such as rack identification, the rack geometry (physical coordinates of the rack object), capacity\_using\_power, capacity\_using\_intensity, energy consumption, and CO2 emission.

Note that this schema includes a pointer to the node schema and an array structure to record all of the nodes identification within the Rack entity:

"installedNode": {

"type": "array",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/computational\_node.json"

},

"description": "The computational nodes in the given rack [ref node schema of the Edge-Cloud Continuum Data model ]"

} }

* ***Room***

Link: <https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Room.json>

The Room schema includes attributes such as room identifier, customer organization, number of racks in the room, capacity metrics, monitoring metrics, real-time data for energy consumption, temperature, electrical intensity, humidity, and CO2 emission rate.

"available\_racks": {

"type": "array",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Rack.json"

},

"description": "Number of available racks in the room [reference to the racks schema of Edge-Cloud Continuum data model ]"

},

* ***Site***

link : <https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/site.json>

A site entity represent the aggregate of the different rooms in a data center. The entity has some identification attributes (ID, concerned customers, rooms list) and the rest includes metrics like energy consumption, power consumption, CO2 emission, and site temperature). The list of room attributes is a pointer to the room schema in the model, the customers of the site are identified through the customer schema reference.

"list\_Rooms": {

"type": "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Room.json"

},

"description": "List of rooms in the site [reference to the rooms schema in the Edge-Cloud Continuum data model]"

},

"Customer": {

"type": "string",

"description": "The customer organization associated with the site [ reference to the customer schema of Edge-Cloud Continuum data model]"

}

* ***Service***

Link: <https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/service.json>

The service schema represents the functionality that a VM, container, or node offers. There is the possibility that a service is offered through the coordination of more than one virtualization unit (for example single microservice is deployed in more than one container). The attribute “deployedRefContainers” is a reference to the container executing the service functions.

"deployedRefContainers": {

"type": "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/computational\_node.json"

},

"description": " container to which a service is deployed, reference to the container model of Edge-Cloud Continuum data model"

}

}

* ***Application***

<https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Application.json>

Application is the set of services (microservices) that make it functions complete to offer the end user. The entity schema has the identification of the application, URL, status, list of the services that are making this application, and the criticality level. The service list attribute points to the reference schema of the service in the model as depicted below.

"servicesList": {

"type": "array",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/service.json"

},

"description": "List of services that constitute the application. [reference to the service schema of the Edge-Cloud Continuum data model ]"

}

* ***Virtual Machine***

<https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Virtual_Machine.json>

The virtual machine entity represents the system-level virtual instance in a computational node. Alongside the identification of the VM, the entity includes the measurement metrics which include the energy consumption, CO2 emission, CPU, memory utilization, bandwidth, and the latency induced in the service response. Additionally, the entity also includes a parameter to define that this VM is installed on which computational node in the cluster. Moreover, the schema also includes the reference Rack and the Cluster (as abstract-level identification of the VM)

"underlying\_computational\_Node": {

"type": "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/computational\_node.json"

},

"description": "The computational node on which the VM is executing and offering a service [Reference to the schema of the of computational node here from the Edge-Cloud Data Model]"

},

"referenceCluster": {

"type": "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/cluster.json"

},

"description": "[Reference to the schema of the of Cluster here from the Edge-Cloud Data Model]"

},

"referenceRack": {

"type": "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Rack.json"

},

"description": "[Reference to the Rack schema here from the Edge-Cloud Data Model]"

}

* ***Container***

link:<https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/container.json>

The container entity schema is described through all identification of the container (e.g., containerId), and the metrics that represent the measurements of a container. This includes the resource utilization of a container (i.e., CPU, RAM, energy) and others like the latency of the functions deployed in a container, carbon emission of a single container, the bandwidth available, and network interface data (in, and out) rates.

Note that an additional attribute in the schema, named deployedInNode is to append the reference of node in which the container is executing. The other attribute is referenceInstalledApplication, a pointer to the application whose function/components are actually containerized.

"deployedInNode": {

"type": "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/computational\_node.json"

},

"description": "The computational node in which the container is deployed[ref node schwema of the Edge-Cloud Continuum Data model ]"

},

"referenceInstalledApplication": {

"type" : "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Application.json"

},

"description" : "This attribute represents the reference application: whose functionality is executed in this container"

}

* ***Application\_Pod***

Link : <https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Application_pod.json>

This is referred to as Kubernetes' smallest unit and is widely discussed terminology in application deployments. A single Pod may contain multiple containers performing a common function and cooperating for single objectives. The pod has attributes for the identification, importantly the name, and other performance metrics like consumptions (CPU, memory, network transmission, energy), disk input-output, and Co2 footprint. One of the reference models in the description of the pod is a computational node.

"deployedInNode": {

"type": "string",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/computational\_node.json"

},

"description": "The computational node in which the Pod is deployed[ref node schema of the Edge-Cloud Continuum Data model ]"

}

}

* ***Pod***

<https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Pod.json>

In data center terminology, different from the Kubernetes deployment, a Pod represents a set of racks. It has the identification attribute (id) and other metrics including CO2 emission, energy consumption, list of the racks, and Pod electrical intensity. The listRack attribute is a reference to the rack schema for cataloging the racks with a single pod in the data center.

Note that the interpretation of the Pod in the context of data center infrastructure and application deployment is different.

"listRacks": {

"type": "array",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Rack.json"

},

"description": "List of racks constituting the POD in the data center [reference to the rack schema of Edge-Cloud Continuum Data model]"

}

* ***Cluster***

<https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/cluster.json>

The cluster is a representation entity for the logical aggregate of the data centers. This is an abstract entity in the data model. It has identification through a unique cluster ID and a list of the concerned customers. The metrics of the cluster energy consumption, and CO2 emission. The customer and site list attributes points to the reference of the customer and the site schema in the Edge-Cloud Continuum data model.

"referenceCustomer": {

"type": "string",

"description": "The client organization associated with the site[Reference to the customer schema of the Edge\_Cloud Continuum data model]"

}

"List\_Sites": {

"type": "array",

"items": {

"$ref": "https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/site.json"

},

"description": "List of sites within the cluster [reference to the site schema]"

}

* **Customer**

Link : <https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/Customer.json>

This schema represents the customer of services offered by a service provider on the top of virtual resources of a data center. The main components for the customer are the ID, location, and the service it interacts.

"subscribredServices": {

"type": "array",

"items": {

"$ref":"https://github.com/SyedMohsan/Edge-Cloud-Data-Model/blob/main/service.json"

},

"description": "List of subscribed services[these reference to subscribed services is necessary to offer the view to customer

]"

}