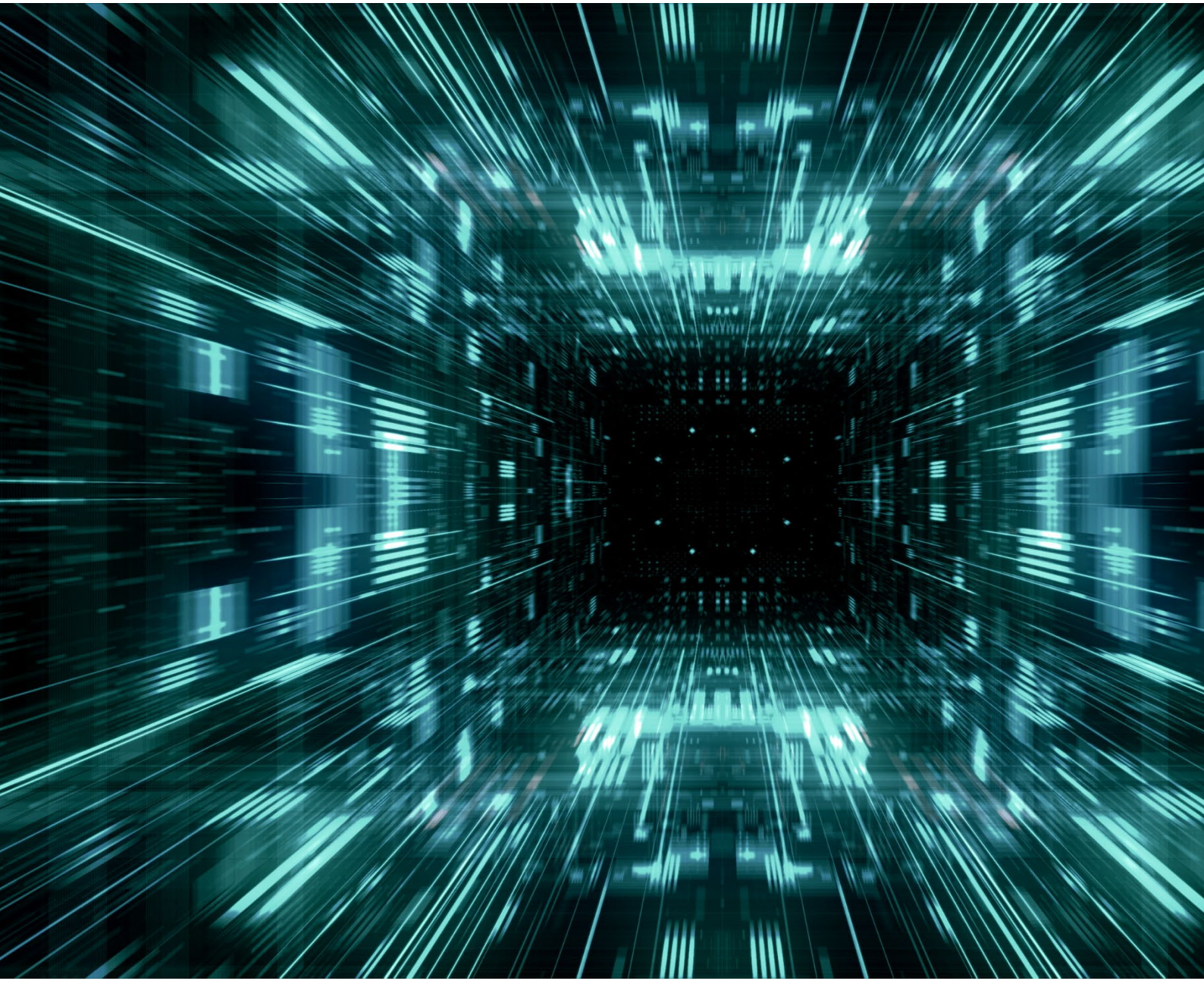
DAIM
injecting insights

DAIM Edge Computing Platform
API Introduction



30.05.2023

DAIM GmbH | Linzer Straße 2-6 | A-4320 Perg | +43 7262 52834 | office@daim.tech | ATU75416337 | FN 531357h

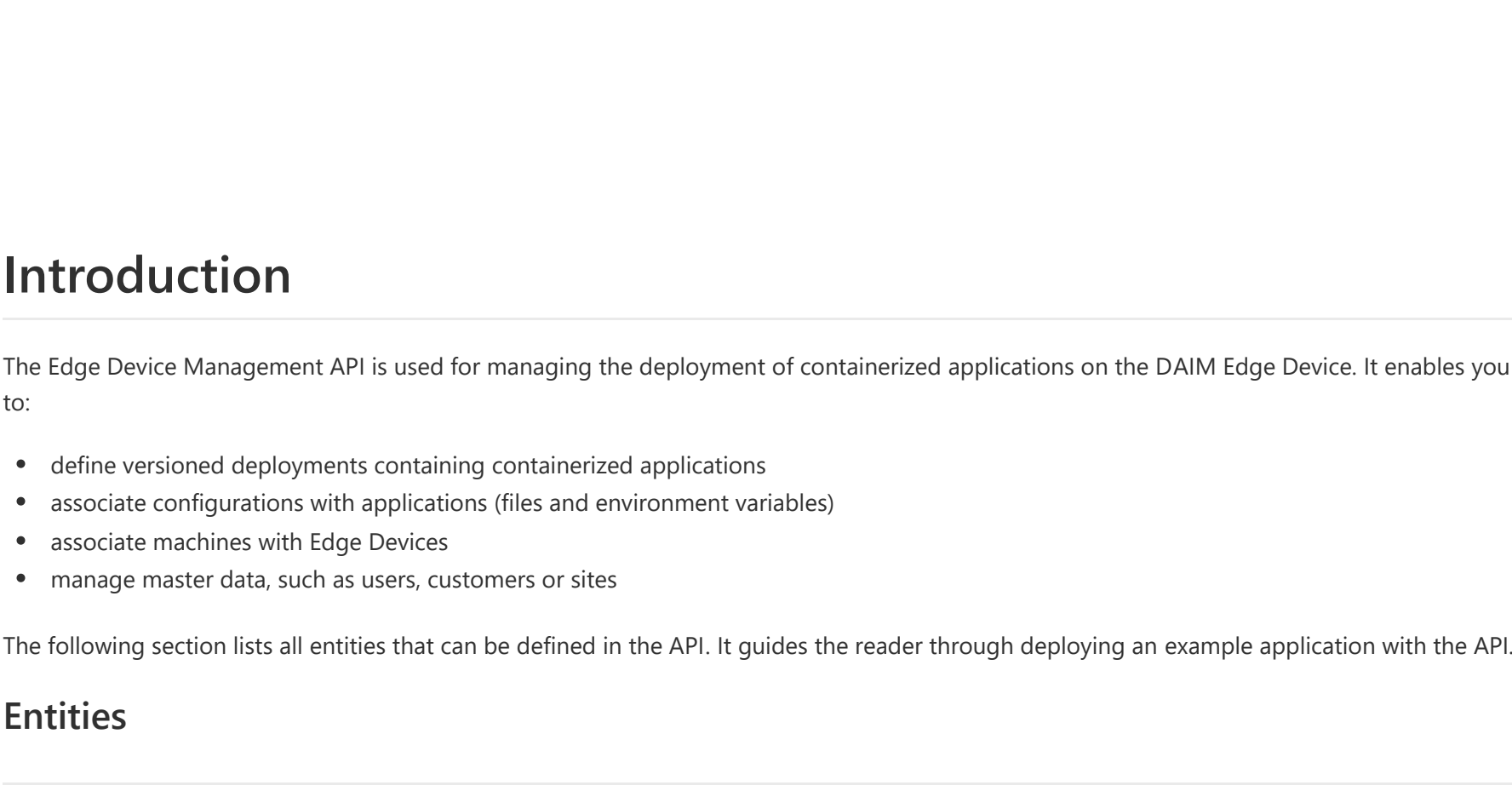
Introduction

The Edge Device Management API is used for managing the deployment of containerized applications on the DAIM Edge Device. It enables you to:

- define versioned deployments containing containerized applications
- associate configurations with applications (files and environment variables)
- associate machines with Edge Devices
- manage master data, such as users, customers or sites

The following section lists all entities that can be defined in the API. It guides the reader through deploying an example application with the API.

Entities



API Entities and relations (UML class diagram)

Edge Device

- Edge Devices are managed entities that represent a physical device.
- Each device is owned by a Customer.
- One or multiple Machines can be attached to an Edge Device.
- Each Edge Device has a dedicated System Version that defines the OS packages, file system structure and management services that are running.
- Each Edge Device has a dedicated Deployment that defines, which Edge Applications Images can be running as Containers on the Edge Device.
- Edge Device can have debugMode enabled which allows to modify running Containers via the Device Management API by HTTP PUT, POST, PATCH and DELETE requests on Containers and CustomConfigurationFiles

Deployment

- Each Deployment consists of a set of dedicated Applications that are to be deployed as Edge Applications Containers on an Edge Device
- Each Edge Device can only be assigned to one dedicated deployment or to no deployment

Application

- Each Application defines a unique name that can be running on an Edge Device as a Container (aka docker container).
- Each Application contains Image information of the docker image as well as environment variables, user/group information of the docker container and ConfigurationFiles to be mounted
- Each Image defines a dedicated Image URL (docker repository, image name and tag) that is used to pull the docker image to the Edge Device during deployment

Configuration

- Configurations are the central element of how to configure a running edge application. It can contains a set of environment variables and 0..n ConfigurationFiles.
- env dict property (key-value-map) defines ENVIRONMENT variables that will be set when container is started.
- data can contain an arbitrary string for holding custom UI specific configuration details (like default values, paths, etc.)

ConfigurationFile

- A file with given name and path will be created and mounted inside docker container on startup. It contains fileContent as content.
- Type can be either TEXT, BINARY, J2_TEMPLATE
- If type is J2_TEMPLATE the file extension ".j2" will be added to the file name when pushed to the git repository. Ansible will process these files by replacing Jinja2 variables (e.g. '{{ var1 }}') before mounting the files to its Container on the Edge Device. Such variables can be defined in Applications, Configurations and Containers (if debugMode is enabled).

Container

- Containers represent deployed Applications on dedicated Edge Devices
- Containers contain user/group information as well as variables, environment variables and CustomConfigurationFile that are currently mounted additionally to the docker container
- If the Edge Device has "debugMode" enabled, these container fields and CustomConfigurationFile can be modified via HTTP PUT/POST/PATCH/DELETE requests, otherwise the information is read-only and there are no CustomConfigurationFiles

CustomConfigurationFile

- CustomConfigurationFile represent additional mounted files that have been added after the deployment. By default, a container has mounted only defined ConfigurationFiles that are associated with the corresponding Application.

Machine

- Machines represent physical machines that are owned by exactly one or no Customer
- Machines can be grouped in Sites, each Machine can only be assigned to one Site or no Site.
- Machines can be attached to exactly one Edge Device or no Edge Device.

Customer

- A Customer represents the customer that owns Machines and Edge Devices
- Customers can be associated to Machines, Sites, and Edge Devices
- Customers can have users that may be given permission to access the Device Management Web Portal or Device Management API

Reserved ports on Edge Device

The following TCP ports are already occupied on the DAIM edge device by DAIM management services:

Port	Interfaces	Description
22	all	sshd if maintenance is enabled
53	lo, support	dnsmasq for onbboarding UI
80	support	onboarding UI
5355	all	systemd-resolve

Port	Interfaces	Description
8090	lo	connsuite HTTP requests
8125	lo	netdata
8443	all	nifi (if UI is enabled)
18080	all	onboardingd
19999	lo	netdata