



IE MANAGER

Requirements Specifications

Version *1.2*

Mentor A Siemens Business

Version history

Version #	Implemented By	Revision Date	Reviewed By	Approved By	Reason
0.0	<i>Amr El Sehemly, Mohamed Hatem</i>	<09/22/19>	Ghada Bahig	Ghada Bahig	Initial Design Definition draft
1.0	<i>Amr El Sehemly, Mohamed Hatem Ahmed Ibrahim</i>	<10/07/2019>	Ghada Bahig, Emmanuel Petit	Ghada Bahig	Initial Design and specifications document
1.1	<i>Amr El Sehemly, Mohamed Hatem Ahmed Ibrahim</i>	<10/31/2019>	Ghada Bahig	Ghada Bahig	Upgraded specifications, designs and assumptions
1.2	<i>Amr El Sehemly, Mohamed Hatem Ahmed Ibrahim</i>	<12/03/2019>	Ghada GBahig, Emmanuel Petit		Upgraded specifications, designs and assumptions

Table of Contents

Version history	0
Table of Contents	2
Introduction	5
Definitions	6
Assumptions	7
Architecture	7
Workflow and user stories	8
User stories	8
Workflows	9
IEManager Download new IEMS workflow	9
IEManager Download new IEMS workflow	10
IEMS Successful Checkout license for an IE device workflow	11
IEMS Failed Checkout license for an IE device workflow	11
IEManager Purchase to a prepaid package workflow	12
IEManager Package decommissioning	13
Services	14
1. Download IEMS package	15
Request	15
Response	16
2. Configure	17
Request	17
Response	19
3. Regenerate IEMS/OVA download URI	20
Request	20
Response	20
4. Generate iso configuration file download URI	21
Request	21
Response	21
5. Get Configurations	22
Request	22
Response	22
6. Get Configurations	23
Request	23

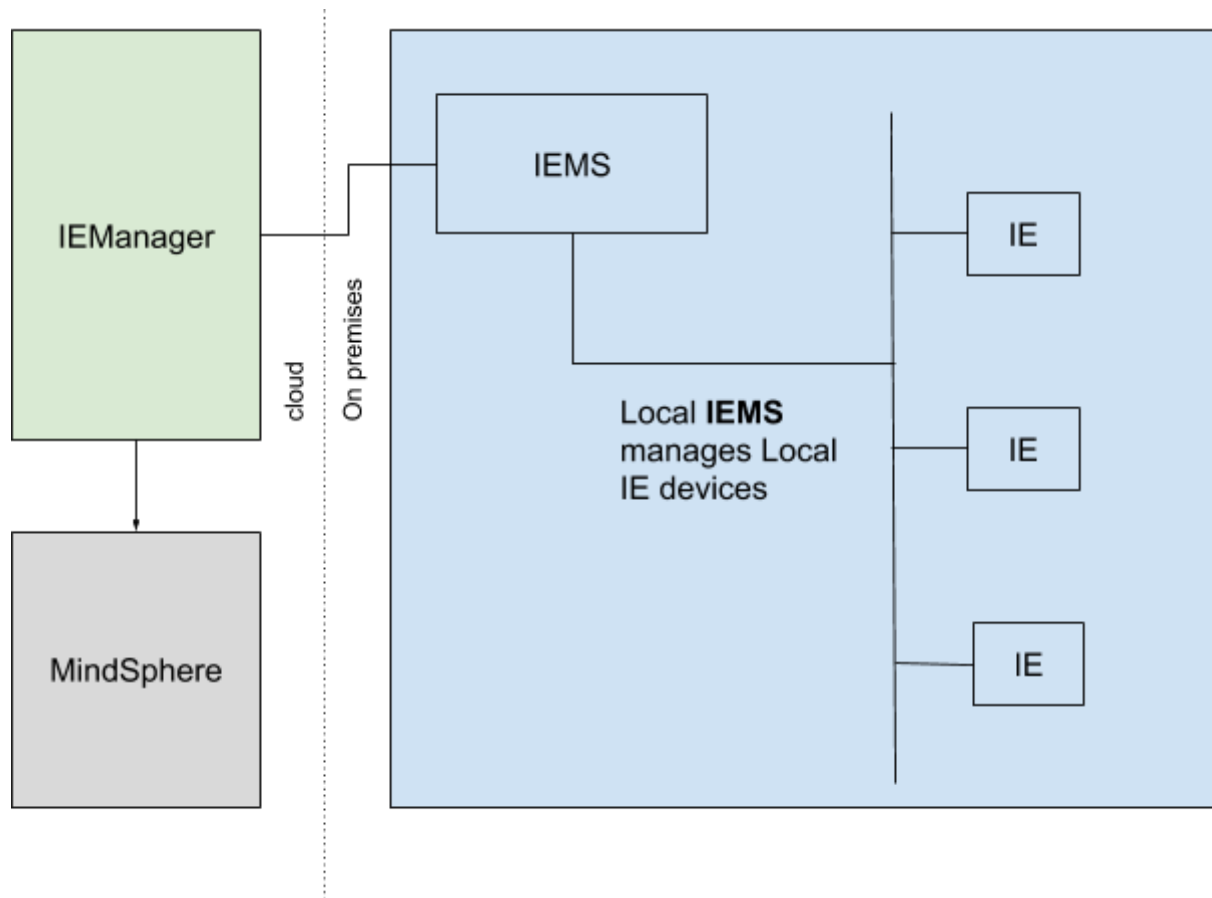
Response	23
7. Get Configurations	24
Request	24
Response	24
8. Get user iems instances	25
Request	25
9. Get user iems instance by id	25
Request	25
10. Get user iems transactions/activities	27
Request	27
11. Get user iems devices	28
Request	28
12. Get Subscription packages	29
Request	29
Response	29
13. Purchase	30
Request	30
Response	30
14. Check out device license	31
Response	32
15. Check in license device	33
Response	33
16. Check-in all devices per IEMS	34
Response	34
17. Get User Subscribed packages:	35
Request	35
18. Decommission package	36
Request	36
Response	36
19. Get IE devices transactions history	37
Request	37
Response	37
20. Get activities (All transactions)	38
Request	38
Response	38
Data Model	39
Data Model Diagram:	39
UI	39
Mock-ups	41

Dashboard	42
User Activities	42
VM DistributionManager	43
IE Manager	44
IE Package Manager	45
Appendix	46
IEMS config JSON schema	46
Response example applying HAL conventions	50

Introduction

IEManager is a front gate cloud based system for industrial edge customers to manage IE user activities via the following features

- subscribe to industrial edge management packages
- download IEMS software
- checkout license/check-in license of IE management packages
- track transaction history.



Definitions

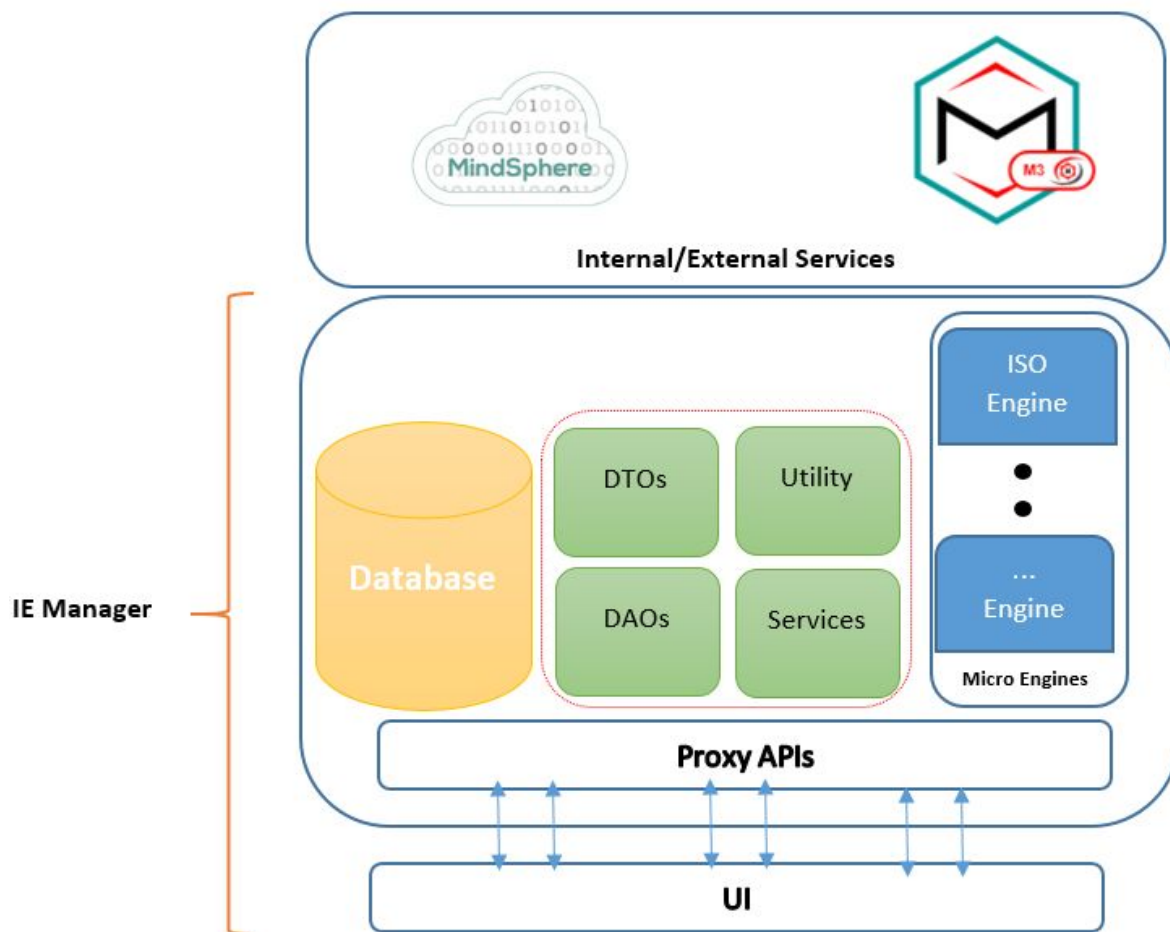
- User : *User (IE user) as one of these end users*
 - User who will be using the IEManager portal to manage subscriptions/IEMS instances/track IE devices activities and history
 - User who will be using IEMS VM images to provision his industrial edge device (we clarify the distinction between these users as needed in user stories and workflows)
- IEManager : *Industrial Edge Manager, which is the system used to*
 - Manage package subscription
 - Verify user packages while checkin/checkout license activities take place on IE devices
 - Configure IEMS instance and download the configuration as a **.iso** file that will be used as a guest disk image to be attached to the IEMS instance .
 - Download IEMS image
 - Track activities/transactions of IE user with regards to the previous functionalities
 - Track package subscription activities
 - Track checkin/checkout license activities
 - Track download IEMS images activities
- IEManager portal : Frontface UI pages which utilize the IEManager backend services to support IEManager functionalities .

Assumptions

1. User is not allowed to reuse his subscription quota assigned to a specific device even if the device is out of service or offboarded by the user.
2. IEMS essential configuration is to be validated and generated from IEManager according to user customization as per [this schema](#) as a .iso disk image that will be used with IEMS VM image

Architecture

IEManager Architecture is mainly separated into 2 modules back-end and front-end, Back-end core is structured as 3 basic layers DAOs (Data Access Objects layer), DTOs (Data Transfer Objects layer) and core services as shown below. Back-end is calling some external services related to file management from Mindsphere cloud APIs



Workflow and user stories

The system could be described from an abstract business view as a set of user stories which summarizes the required features and expectations as follows

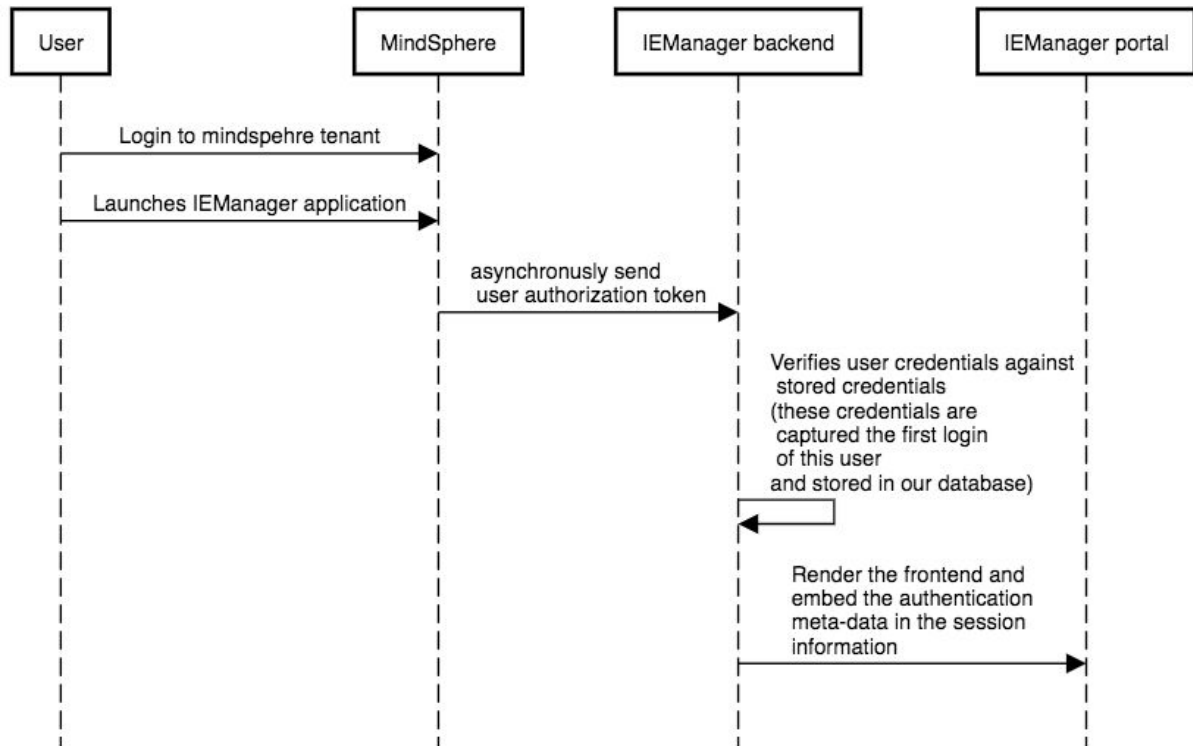
User stories

1. As an **IE user (customer)**, I need to **launch IEManager application through MindSphere and view a management dashboard**, so that I can analyze, control/manage my subscription and local edge management system distributions(IEMS).
2. As an **IE user (customer)**, I need **to have a VM distribution manager page** in the portal so that I can configure and download the necessary resources for using my IEMS image.
3. As an **IE user (customer)**, I need **to have “user-activities” page** in the portal, so that I can track my billing activities, check-in , checkout device license activities and my downloaded IEMS image.
4. As an **IE user (IEMS user)** I need to check out a package license when a device onboarding takes place so that my device is activated and capable of utilizing the power of IE software.
5. As an **IE user (IEMS user)** I need to check in a package license when a device offboarding takes place so that my device is deactivated.
6. As an **IE user (Customer)** I need to **subscribe to** an IE package so that I can consume it's available licenses for onboarding IE devices to a certain IEMS.
7. As an **IE user (Customer)** I need to **decommission** a subscribed package so that it doesn't automatically get renewed at it's renewal date.

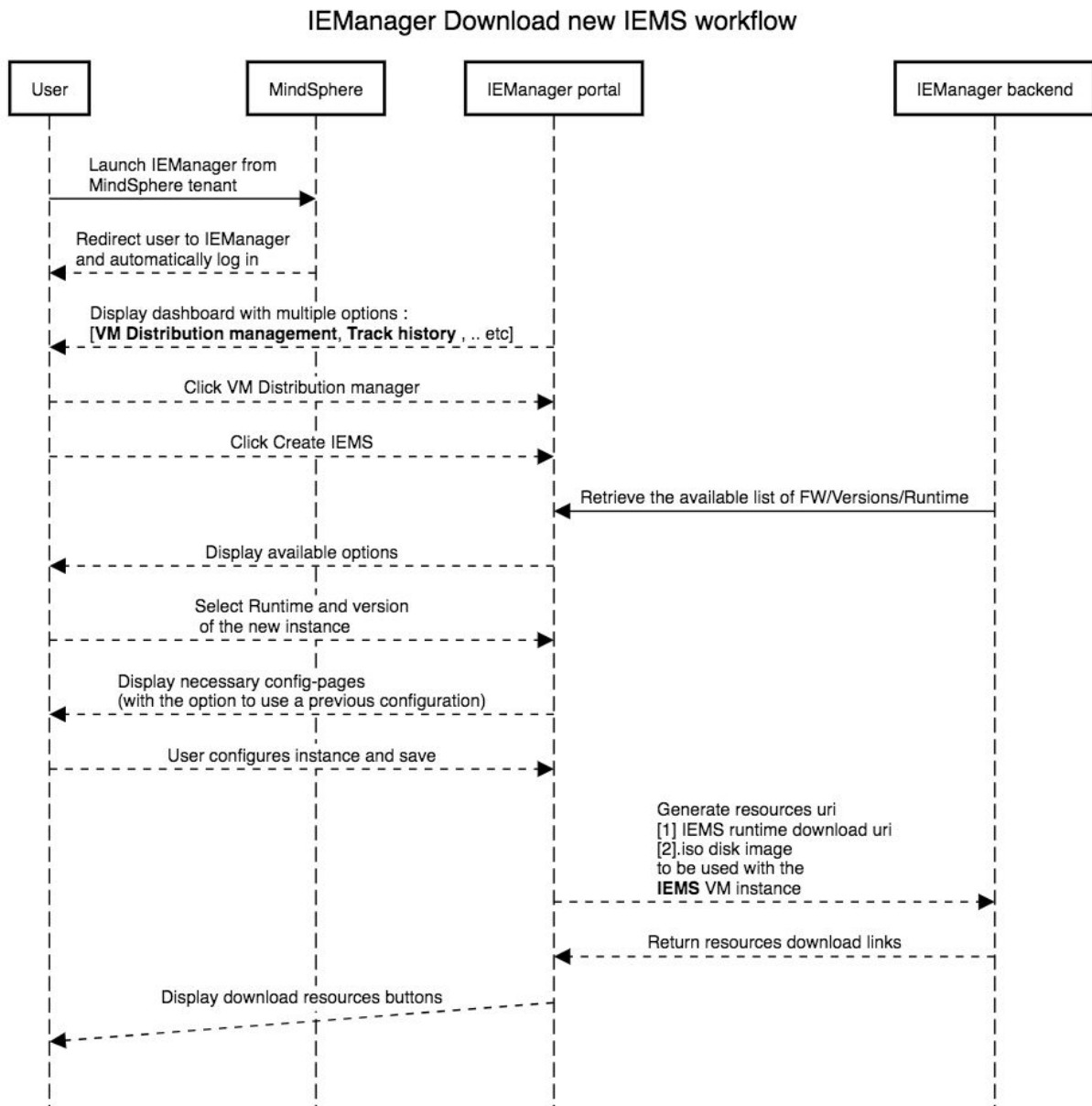
Workflows

A typical user (Industrial edge customer who will use IEManager portal) is expected to go through the following workflows in runtime utilizing IEManager system

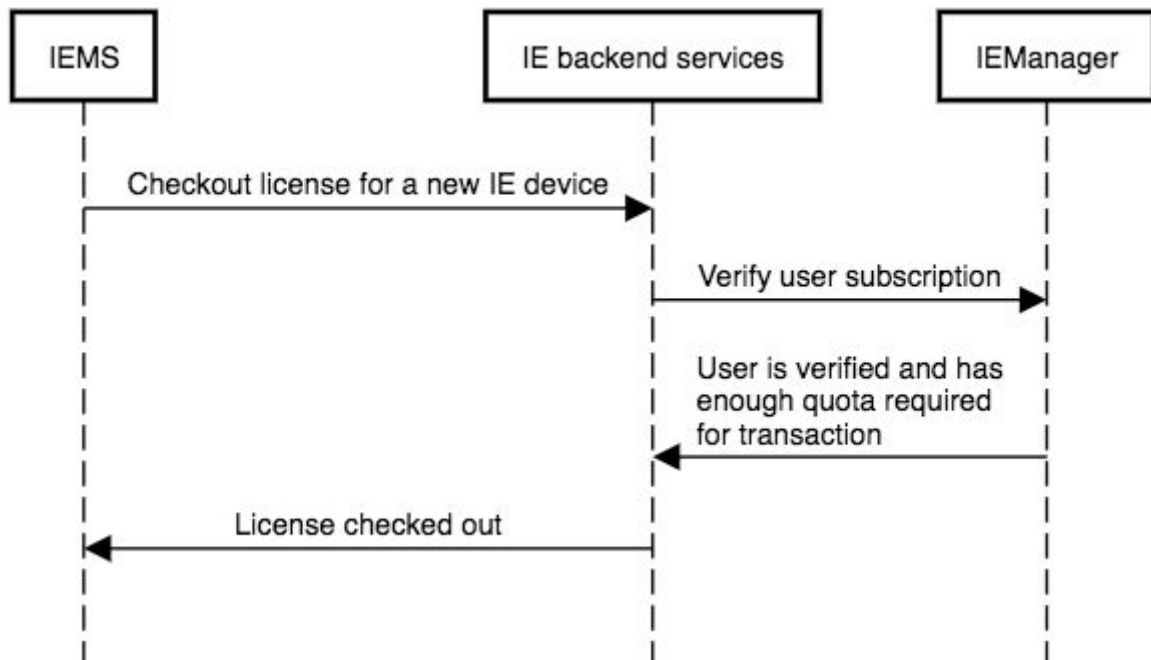
IEManager Download new IEMS workflow



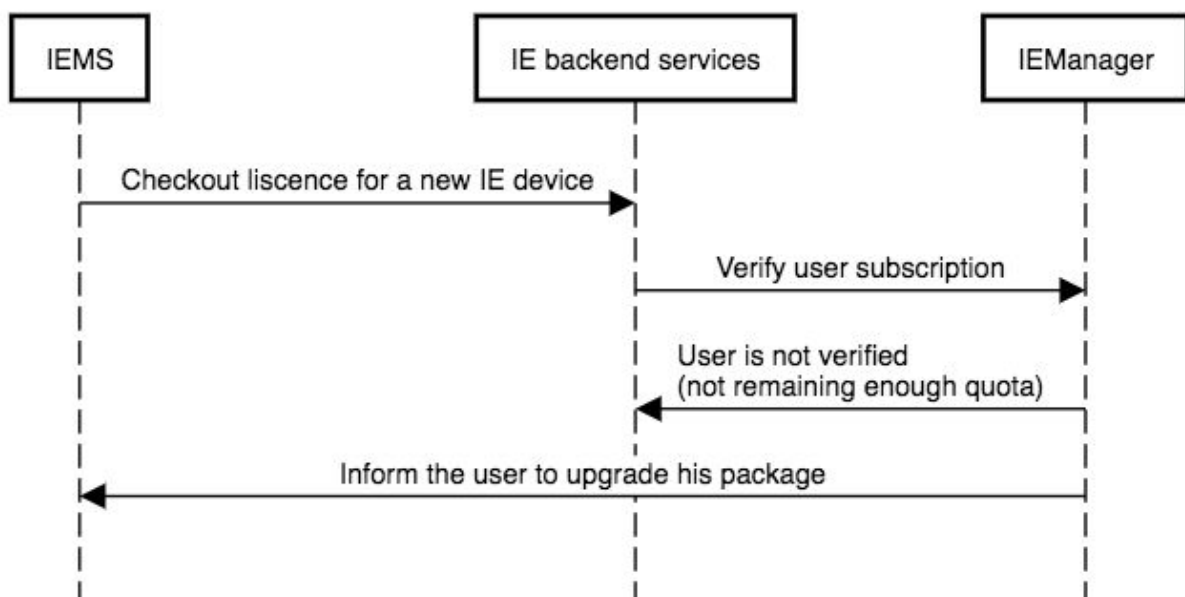
IEManager Download new IEMS workflow



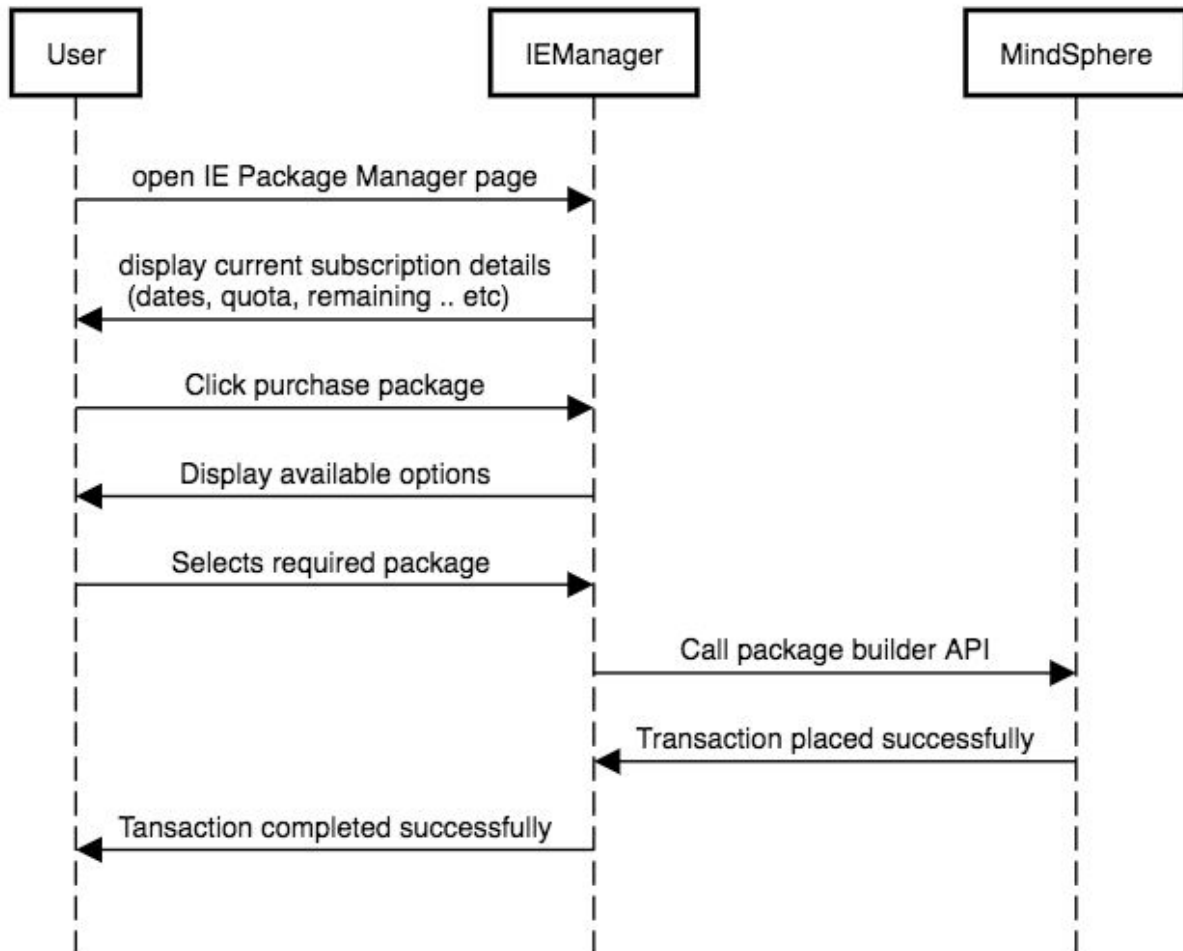
IEMS Successful Checkout license for an IE device workflow



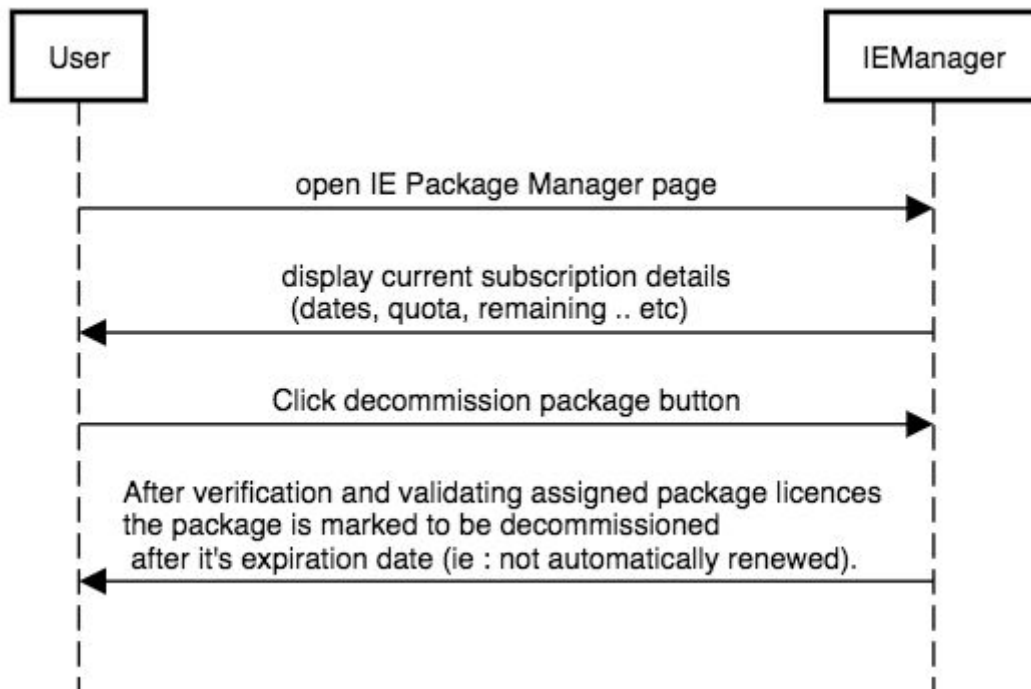
IEMS Failed Checkout license for an IE device workflow



IEManager Purchase to a prepaid package workflow



IEManager Package decommissioning



Services

Services are the collection of exposed APIs supporting the IEManger's capabilities. They can be used through the IEManager portal or called through the IEMS (as a client to IEManager services). Services are following API first strategy and the HAL Conventions as per [this example](#).

For simplicity, we write the response examples of the following APIs without the hal conventions to easily read and understand the purpose of each API.

1. Download IEMS package

This service is responsible for creating an IEMS resource instance in our database and generating a download link for the IESM Package. It will call an internal service in MS to access MS storage and allow user to download the package (IEMS image).

Request

Method	URL
GET	iemanager/api/v1/iems-download/:instanceId?<version>

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string
QUERY_PARAM	<version>	number

version

The `version` is an optional param in case need to download specific version of IEMS package.

Response

Status	Response
200	<pre>{ "id": "d5e3eb70-bg45-11ea-]]1q-45512b032864", "name": "test", "userId": 193, "customerId": 193, "accountId": 193, "iemsVmId": "558ssb-85e5-4755-ad2c-19dcb46cb907", "hostname": "iems.mentor.io", "state": 1, "downloadURI": "download-uri" }</pre>
400	<pre>{"error": "Invalid version no."}</pre>
401	<pre>{"error": "Invalid Auth key."}</pre>
500	<pre>{"error": "Error description"}</pre>

2. Configure

After user customizes the configuration, it will be stored in DB and then ISO engine module will generate an iso file containing the json configuration. This service will store this configurations in DB to enable user to retrieve the configuration in future instances for re-use and to track configurations for created VM images.

Request

Method	URL
POST	iemanager/api/v1/iems/<iems-id>/configuration
Body	<pre>{ "version": 1, "keyboard-config": "en", "hostname": "example.host.mentor.com", "dns-settings": { "nameservers": ["137.202.38.2", "8.8.8.8"], "searches": ["mentor.com"] }, "network-settings": { "dhcp-mode": "auto", "manual-settings": { "ip-address" : "1.1.1.1/16", "gateway" : "10.10.10.10" } }, "proxy-settings": { "http": "http://admin:password@139.126.38.2", "https": "http://admin:password@139.126.38.2", "no_proxy": ["localhost", "xyz.com", "12.14.15.6"] }, }</pre>

	<pre> "certificates": [{ "type": "backend", "private-key": "", "certificate": "" }], "cloud-credentials": { "client-id": "ZYDPLLBSK3MVQJSIYHB10R2JXCY0X2C5UJ2QAR2MAAIT5Q", "client-secret": "YXBpVXJsOiAiaHR0cHM6Ly9teS5hcGkuY29tL2FwaS92MSIKdXNlcm5hbWU6IHt7dXNlcm5hbWV9fQpwYXNzd29yZDoge3twYXNzd29yZHI9", "customer" : "api.poc.edge.siemens.cloud", "account-id" : 1000, "iems-id" : "0000-0000-0000-0000-0000-0000-0000-0000" } </pre>
--	--

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string

Response

Status	Response
201	<pre>{ "id": "00273490-]m,b-11ea-bc56-45512b032864", "iemsId": "<iems-id>", "metadata": "<configuration-metadata>", "version": 2, "physicalResourceName": "<resource-name>", "date": "<date>" }</pre>
401	<pre>{"error": "Invalid Auth key."}</pre>
500	<pre>{"error": "Error description"}</pre>

3. Regenerate IEMS/OVA download URI

This service is responsible for re-creating a download link for the ova resource linked to a certain IEMS resource

Request

Method	URL
GET	iemanager/api/v1/firmwares/:iemsId/ uri

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string

Response

Status	Response
200	["uri"]
401	{"error":"Invalid Auth key."}
500	{"error":"Error description"}

4.Generate iso configuration file download URI

This service is responsible for creating a download link for the iso file resource of a certain configuration object

Request

Method	URL
GET	iemanager/api/v1/configurations/:resourceName/ uri

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string

Response

Status	Response
200	["uri"]
400	{"error":"Invalid version no."}
401	{"error":"Invalid Auth key."}
500	{"error":"Error description"}

5. Get Configurations

List the history of configuration generated

Request

Method	URL
GET	iemanager/api/v1/configurations

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string

Response

Status	Response
200	<p>Response will be an object containing the list of configuration(array). Each item in the array has the following structure/example.</p> <pre>{ "id": "00273490-]m,b-11ea-bc56-45512b032864", "iemsId": "<iems-id>", "metadata": "<configuration-metadata>", "version": 2, "physicalResourceName": "<resource-name>", "date": "<date>" }</pre>
401	{"error": "Invalid Auth key."}
500	{"error": "Error description"}

6. Get Configurations

Get specific configurations object by id

Request

Method	URL
GET	iemanager/api/v1/configurations/:id

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string

Response

Status	Response
200	<p>Response will be an object containing the configuration metadata. as per the following structure/example.</p> <pre>{ "id": "00273490-]m,b-11ea-bc56-45512b032864", "iemsId": "<iems-id>", "metadata": "<configuration-metadata>", "version": 2, "physicalResourceName": "<resource-name>", "date": "<date>" }</pre>
401	{"error": "Invalid Auth key."}
500	{"error": "Error description"}

7. Get Configurations

List the history of configuration versions for a specific iems

Request

Method	URL
GET	iemanager/api/v1/iems/:iems-id/configurations

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string

Response

Status	Response
200	<p>Response will be an object containing the list of configuration(array). Each item in the array has the following structure/example.</p> <pre>{ "id": "00273490-]m,b-11ea-bc56-45512b032864", "iemsId": "<iems-id>", "metadata": "<configuration-metadata>", "version": 2, "physicalResourceName": "<resource-name>", "date": "<date>" }</pre>
401	{"error": "Invalid Auth key."}
500	{"error": "Error description"}

8. Get user iems instances

Request

Method	URL
GET	iemanager/api/v1/iems

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string

Response

Status	Response
200	<p>Response will be an object containing the list of iems instances (array). Each item in the array has a structure as per the following example.</p> <pre>{ "id": "2c8c2200-144d-11ea-8f2c-01148a51cc6a", "name": "test", "userId": 193, "customerId": 193, "accountId": 193, "iemsVmId": "a7aade67-85e5-4755-ad2c-19dcb46cb907", "hostname": "example.host.mentor.com", "state": "commissioned" }</pre>
401	{"error": "Invalid auth key."}
500	{"error": "Error description"}

9. Get user iems instance by id

Request

Method	URL
GET	iemanager/api/v1/iems/:id

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string

Response

Status	Response
200	<p>Response will be an object containing the iems instances as per the following example.</p> <pre>{ "id": "2c8c2200-144d-11ea-8f2c-01148a51cc6a", "name": "test", "userId": 193, "customerId": 193, "accountId": 193, "iemsVmId": "a7aade67-85e5-4755-ad2c-19dcb46cb907", "hostname": "example.host.mentor.com", "state": "commissioned" }</pre>
401	<pre>{"error": "Invalid auth key."}</pre>
500	<pre>{"error": "Error description"}</pre>

10. Get user iems transactions/activities

Get all user iems activities.

Request

Method	URL
GET	iemanager/api/v1/transactions/iems?<id>

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string
QUERY_PARAM	id	string

Response

Status	Response
200	<p>Response will be an object containing the list of transactions(array). Each item in the array has the following structure.</p> <pre>{ "id": "aaaassss-16c1-11ea-bc56-45512b032864", "userId": 12, "customerId": 12, "accountId": 12, "transactionTypeId": 5, "transactionType": "IEMS_CONFIGURATION_TRANSACTION", "iemsId": "kjkjkjkj-5544-11ea-bc56-45512b032864", "iemsConfigId": "nnnnhhhh-16c1-11ea-bc56-45512b032864", "transactionDate": "1575482877680", "iemsHostname": "<hostname>" }</pre>
401	{"error": "Invalid auth key."}
500	{"error": "Error description"}

11. Get user iems devices

Request

Method	URL
GET	iemanager/api/v1/iems/<iems-id>/devices

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string

Response

Status	Response
200	<p>Response will be an object containing the list of devices (array). Each item in the array has the following structure.</p> <pre>{ "id": "<id>", "serialNo": "100100102", "name": "<name>", "iemsId": "<id>", "licenseId": "<id>", "status": "onboarded", "licenseStatus": "active" }</pre>
401	{"error": "Invalid auth key."}
500	{"error": "Error description"}

12. Get Subscription packages

List all Subscription packages available to allow user subscribe.

Request

Method	URL
GET	iemanager/api/v1/subscription-packages

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string

Response

Status	Response
200	<p>Response will be an object containing the list of packages(array). Each item in the array has the following structure.</p> <pre>{ "id": "<id>", "name": "Golden Package", "duration": <duration>, "quota": 10 }</pre>
401	{"error": "Invalid auth key."}
500	{"error": "Error description"}

13. Purchase

Request

Method	URL
POST	iemanager/api/v1/purchasing-packages
Body	<pre>{ "package_id": "a2aaaac98-1aa0-11ea-8d71-362b9e155667" }</pre>

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string

Response

Status	Response
201	<pre>{ "id": "<id>", "transactionTypeId": 1, "transactionType": "PACKAGE_SUBSCRIPTION", "userSubPackageId": "<user-subscription-package-id>", "transactionDate": 1575485215854 }</pre>
401	<pre>{"error": "Invalid auth key."}</pre>
500	<pre>{"error": "Error description"}</pre>

14. Check out device license

This API handles checking out device license request to allow onboarding an IE device and updates the subscription package to reflect on billing system as below:

- Get user subscription package details.
 - in case there is an available licenses the device will be onboarded and the license will be consumed.
 - Otherwise the user will be redirected to subscription page to purchase/subscribe to a new package
- Apply the transaction to MindSphere billing system.
- Store the details in DB.

Request

Method	URL
POST	iemanager/api/v1/iems/<iems-id>/devices/checkout
Body	The body will contain the device object <pre>{ "serialNo": "Abc091231", "name": "device01" }</pre>

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string

Response

Status	Response
201	The created device object with generated uuid { "id": "<id>", "userId": 12, "transactionTypeId": 3, "transactionType": "ONBOARDING_DEVICE", "ieDeviceId": "<id>", "licenceId": "<id>", "transactionDate": 1575485347914 }
401	{"error": "Invalid Auth key."}
500	{"error": "Error description"}

15. Check in license device

This API is used to handle off-boarding device, change the device license status to be inactive and update in subscription package.

Request

Method	URL
POST	iemanager/api/v1/iems/<iems-id>/devices/check-in
Body	The body will contain the device serial number { "serialNo": "Abc091231" }

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string

Response

Status	Response
201	The device object whose license will be checked out { "id": "<id>", "userId": 193, "transactionTypeId": 6, "transactionType": "OFFBOARDING_DEVICE", "ieDeviceId": "<id>", "licenceId": "<id>", "transactionDate": 1575485680035 }
401	{"error": "Invalid Auth key."}
500	{"error": "Error description"}

16.Check-in all devices per IEMS

Check in all devices' licenses per IEMS and update in subscription package to reflect on the billing system.

Request

Method	URL
POST	iemanager/api/v1/iems/<iems-id>/devices/ check-in-devices

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
HEAD	Content-Type	string

Response

Status	Response
201	<p>Response will be an object containing the list of detached devices(array). Each item in the array has the following structure.</p> <pre>{ "id": "<id>", "userId": 193, "transactionTypeId": 6, "transactionType": "OFFBOARDING_DEVICE", "ieDeviceId": "<id>", "licenceId": "<id>", "transactionDate": 1575485680035 }</pre>
401	{"error":"Invalid Auth key."}
500	{"error":"Error description"}

17. Get User Subscribed packages:

List all User subscribed packages.

Request

Method	URL
GET	iemanager/api/v1/user-subscription-packages

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string

Response

Status	Response
200	<p>Response will be an object containing the list of transactions(array). Each item in the array has the following structure.</p> <pre>{ "id": "<id>", "userId": 193, "customerId": 193, "accountId": 193, "subscriptionPackageId": "<id>", "startDate": "1575472993718", "expirationDate": "1607008993718", "deviceCountRemaining": 9, "autoRenewal": true, "packageName": "Golden Package", "packageQuota": 10 }</pre>
401	{"error": "Invalid auth key."}
500	{"error": "Error description"}

18. Decommission package

Check-in user subscribed package.

Request

Method	URL
POST	iemanager/api/v1/subscription-packages/:packageId/decommission
Body	

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string

Response

Status	Response
201	<pre>{ "id": "<id>", "userId": 193, "transactionTypeId": 7, "transactionType": "DECOMMISSION_PACKAGE", "userSubPackageId": "<id>", "transactionDate": 1575491208866 }</pre>
401	<pre>{"error": "Invalid auth key."}</pre>
500	<pre>{"error": "Error description"}</pre>

19. Get IE devices transactions history

Get all devices transactions with subscription packages changes in details

Request

Method	URL
GET	iemanager/api/transactions/devices/:fromDate?<toDate>

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
URL_PARAM	<fromDate>	epoch timestamp
Query_PARAM	<toDate>	epoch timestamp

<fromDate>

It's url param to allow get the billing history from specific date

<toDate>

It's query param to allow get the billing history to specific date, otherwise it will get till now.

Response

Status	Response
200	<p>Response will be an object containing the list of device transaction object (array). Each item in the array is structured as per the following example.</p> <pre>{ "id": "<id>", "userId": 2, "transactionTypeId": 6, "transactionType": "OFFBOARDING_DEVICE", "ieDeviceId": "<id>", "licenceId": "<id>", "transactionDate": "1575485680035" }</pre>
401	<pre>{"error": "Invalid auth key."}</pre>
500	<pre>{"error": "Error description"}</pre>

20. Get activities (All transactions)

List all User activities.

Request

Method	URL
GET	iemanager/api/v1/activities?<transaction-type>

Type	Params	Values
HEAD	customer	string
HEAD	account_id	number
HEAD	authorization	string
Query_PARAM	<transaction_type>	int

<transactios_type>

Transactions type to allow user to filter records i.e. subscription transactions, devices transactions and IEMSttransactions.

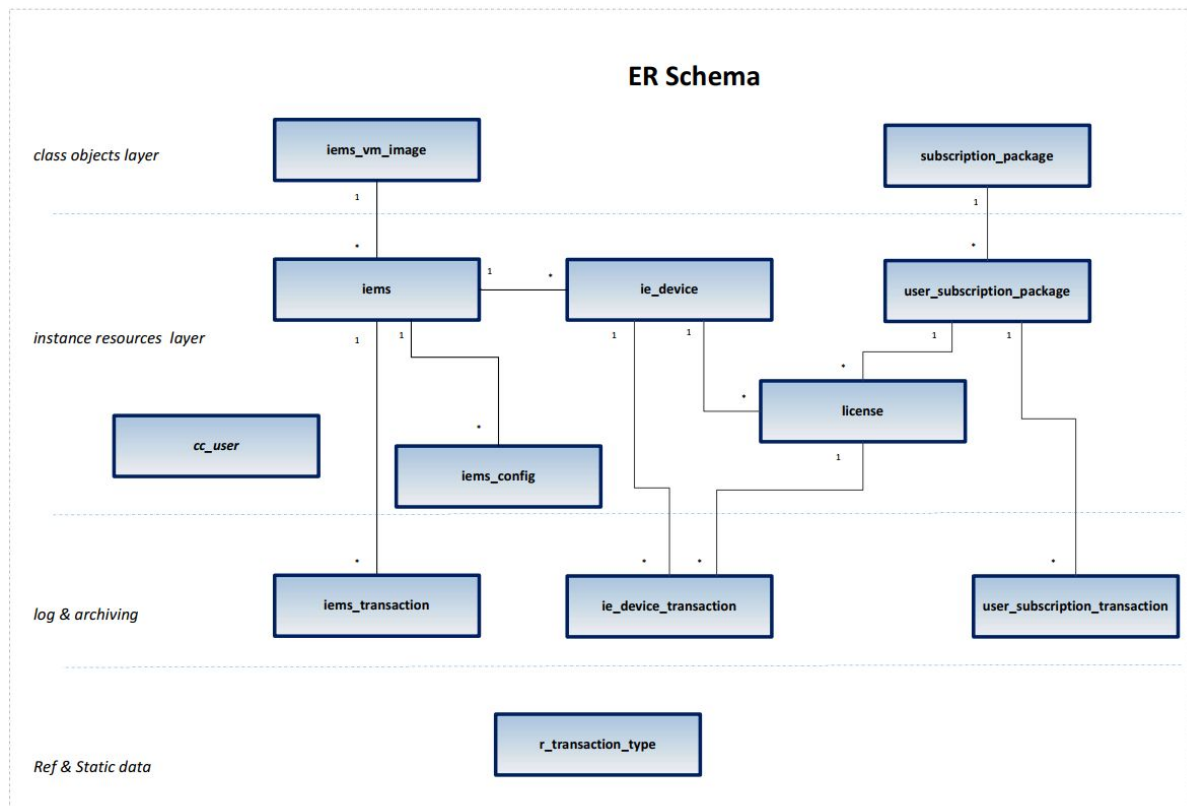
Response

Status	Response
200	<p>Response will be an object containing the list of transactions(array). Each item in the array has the following structure.</p> <pre>{ "transactionId": "<id>", "transactionType": "IEMS_CONFIGURATION_TRANSACTION", "transactionDetails": "IEMS download transaction took place, for an IEMS isntance d5e3eb70-16bf-11ea-bc56-45512b032864, and configuration 00273490-16c1-11ea-bc56-45512b032864", "transactionDate": "1575482877680" }</pre>
401	<pre>{"error": "Invalid auth key."}</pre>
500	<pre>{"error": "Error description"}</pre>

Data Model

Data model handle, store and track customer devices and transactions related to on/off boarding, downloading IESM and any subscription packages stuff.

Data Model Diagram:



Physical Model

iems_vm_image
id *
customer_id
account_id
user_id
name
storage_service {MS, M3, ..}
storage_meta_data
type {}
version
size
supported_architecture
time_tag

subscription_package
id *
name
M5_package_id
offer_expiration
quota
duration (day)
time_tag

iems
id *
name
customer_id
account_id
user_id
iems_vm_image_id
hostname
client_key
client_secret
state {commissioned, downloaded, active}
time_tag

ie_device
id *
serial_no
name
iems_id
license_id
status {onboarded, offboarded, onboarded failed, undefined}
time_tag

user_subscription_package
id *
customer_id
account_id
user_id
subscription_package_id
start_date
expiration_date
remaining_count
auto_renewal_flg
time_tag

license
id *
user_subscription_package_id
status {active, inactive}
time_tag

iems_config
id *
iems_id
meta_data
version {auto increment}
date
physical_resource_name
time_tag

iems_transaction
id *
customer_id
account_id
user_id
transaction_type_id
iems_id
iems_config_id
transaction_date

r_transaction_type
id *
type

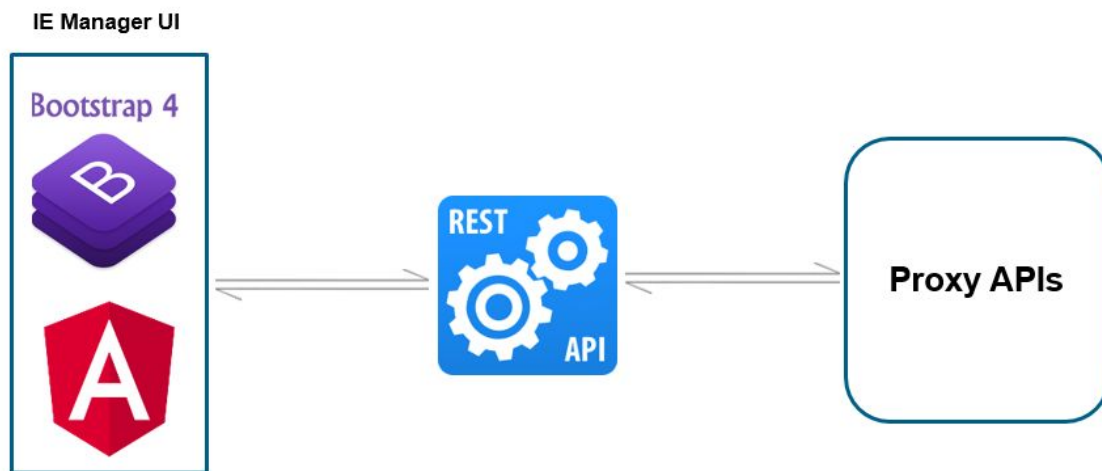
ie_device_transaction
id *
customer_id
account_id
user_id
transaction_type_id
ie_device_id
license_id
transaction_date
details
time_tag

user_subscription_transaction
id *
customer_id
account_id
user_id
transaction_type_id
user_subscription_package_id
transaction_date
time_tag

UI

Our UI application will be a Mindsphere self hosted application.

It will be implemented using Angular 8 and Bootstrap 4 and linked to MindSphere by Proxy APIs via REST calls.



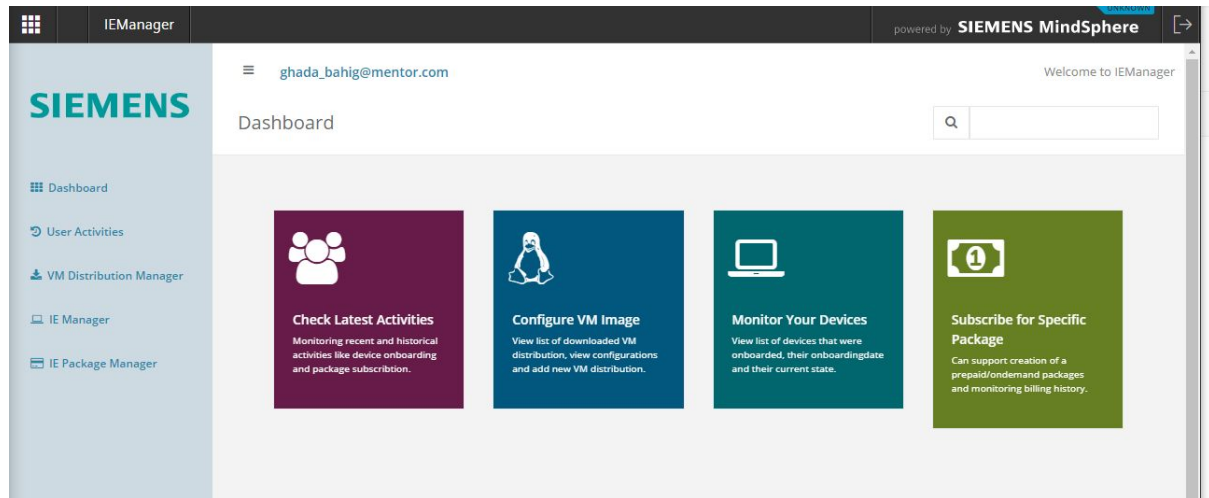
The application will contain 5 pages (as a start) that user can configure and manage VM images, Edge devices and billing options.

The five pages are:

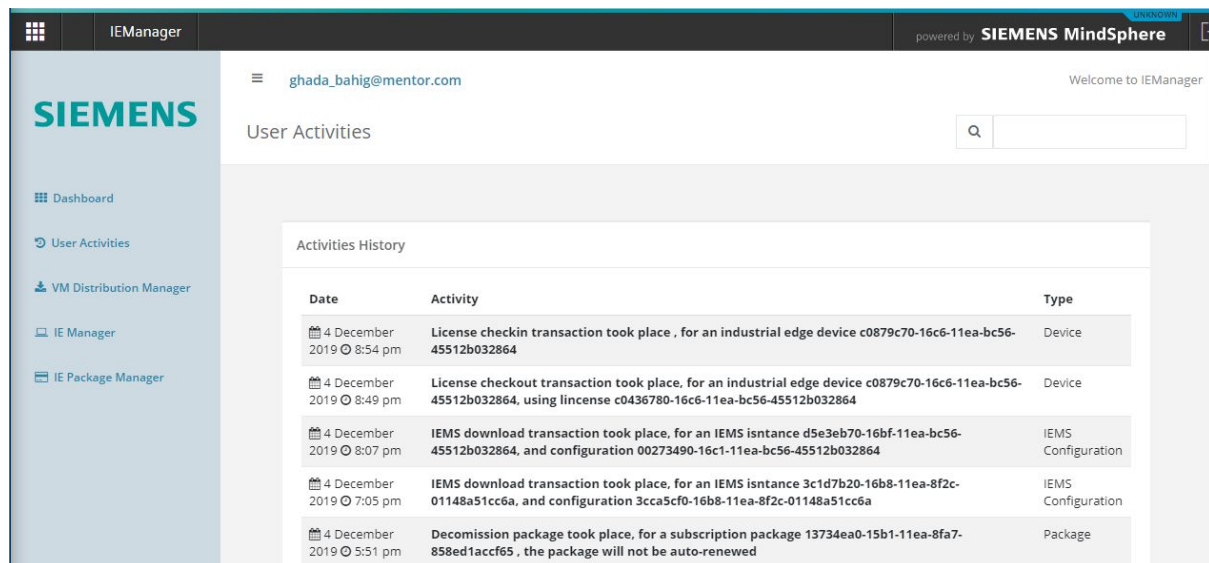
1. Dashboard
2. User Activities
3. VM DistributionManager
4. IE Manager
5. IE Package Manager

Mock-ups

Dashboard



User Activities



VM DistributionManager

IEManager

powered by **SIEMENS MindSphere**

ghada_bahig@mentor.com

Welcome to IEManager

VM Distribution Manager

SIEMENS

Dashboard

User Activities

VM Distribution Manager


IE Manager

IE Package Manager

Configured VM Distribution

Name	IEMS hostname	Configured Date	Action
IEMS-EXAMPLE	example.host.mentor.com	1 December 2019 10:15 am	
IEMS-EXAMPLE	example.host.mentor.com	4 December 2019 7:05 pm	
iem-demo-config	iem-demo.siemens.com	29 November 2019 6:23 pm	

Add new configuration

 IEManager

powered by **SIEMENS MindSphere**

ghada_bahig@mentor.com

Welcome to IEManager

SIEMENS

VM Distribution Manager

Dashboard

User Activities

VM Distribution Manager

IE Manager

IE Package Manager

VM Image Configuration

Name

Select VM Image

ISO Configuration

Hostname

example.host.mentor.com

Keyboard Input

EN

DNS connection

☒ Auto

☐ Manual

Proxy Configuration

http

https

No proxy servers sperated by ', '

Private Key

Choose File

No file chosen


Public Certificate

Choose File

No file chosen

Save changes

IE Manager

 IEManager

powered by **SIEMENS MindSphere**

ghada_bahig@mentor.com

Welcome to IEManager

SIEMENS

IE Manager

Dashboard

User Activities

VM Distribution Manager

IE Manager

IE Package Manager

IEMS


example-test.host.mentor.com

example.host.mentor.com

example-test.host.mentor.com

Device Name	Device Serial	License	Status
demo_dry_run_01	100101	f75fea40-16a3-11ea-93b5-8f6e8ceca8a4	offboarded
demo_dry_run_02	100102	ec5ea3c0-16ad-11ea-be81-e12348f4a5d2	offboarded
device_name	Abc091231	b977c720-169e-11ea-bfe1-f72ff648bbf6	offboarded

IE Package Manager

 IEManager

powered by **SIEMENS MindSphere**

ghada_bahig@mentor.com

Welcome to IEManager

SIEMENS

Dashboard

User Activities

VM Distribution Manager

IE Manager

IE Package Manager

IE Package Manager

Golden Package € 350

10 Devices
365 Days

Subscribe Now

Silver Package € 250

5 Devices
365 Days

Subscribe Now

Billing History

#	Purchase Date	Package	Auto Renewal	Consumption	Remaining	Expiry Date	Action
1	4 December 2019 5:51 pm	Golden Package	✓	0	10	3 December 2020 5:51 pm	Decommission
2	4 December 2019 5:49 pm	Golden Package	✓	1	9	3 December 2020 5:49 pm	Decommission

Appendix

IEMS config JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "version": {
      "type": "integer"
    },
    "keyboard-config": {
      "type": "string"
    },
    "hostname": {
      "type": "string"
    },
    "dns-settings": {
      "type": "object",
      "properties": {
        "nameservers": {
          "type": "array",
          "items": [
            {
              "type": "string"
            },
            {
              "type": "string"
            }
          ]
        },
        "searches": {
          "type": "array",
          "items": [
            {
              "type": "string"
            }
          ]
        }
      }
    }
  }
}
```

```
        "required": [
            "nameservers",
            "searches"
        ]
    },
    "network-settings": {
        "type": "object",
        "properties": {
            "dhcp-mode": {
                "type": "string"
            },
            "manual-settings": {
                "type": "object",
                "properties": {
                    "ip-address": {
                        "type": "string"
                    },
                    "gateway": {
                        "type": "string"
                    }
                },
                "required": [
                    "ip-address",
                    "gateway"
                ]
            }
        },
        "required": [
            "dhcp-mode",
            "manual-settings"
        ]
    },
    "proxy-settings": {
        "type": "object",
        "properties": {
            "http": {
                "type": "string"
            },
            "https": {
                "type": "string"
            }
        }
    }
}
```



```
    },
    "no_proxy": {
      "type": "array",
      "items": [
        {
          "type": "string"
        },
        {
          "type": "string"
        },
        {
          "type": "string"
        }
      ]
    }
  },
  "required": [
    "http",
    "https",
    "no_proxy"
  ]
},
"certificates": {
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "type": {
          "type": "string"
        },
        "private-key": {
          "type": "string"
        },
        "certificate": {
          "type": "string"
        }
      }
    }
  ],
  "required": [
    "type",
```

```

        "private-key",
        "certificate"
    ]
}
]
},
"cloud-credentials": {
    "type": "object",
    "properties": {
        "client-id": {
            "type": "string"
        },
        "client-secret": {
            "type": "string"
        },
        "customer": {
            "type": "string"
        },
        "account-id": {
            "type": "integer"
        },
        "iems-id": {
            "type": "string"
        }
    },
    "required": [
        "client-id",
        "client-secret",
        "customer",
        "account-id",
        "iems-id"
    ]
}
},
"required": [
    "version",
    "keyboard-config",
    "hostname",
    "network-settings",
    "cloud-credentials"
]

```

```
]
}
```

Response example applying HAL conventions

```
{
  "data": [
    {
      "_links": {
        "self": {
          "href": "<gw:uri>/v1/configurations"
        }
      },
      "_embedded": {
        "configurations": [
          {
            "_links": {
              "self": {
                "href":
"<gw:uri>/v1/configurations/0d805650-145d-11ea-bc2e-af4fadfbe670"
              }
            },
            "id": "0d805650-145d-11ea-bc2e-af4fadfbe670",
            "iemsId": "2c8c2200-144d-11ea-8f2c-01148a51cc6a",
            "metadata": "<json-string>",
            "version": 3,
            "physicalResourceName":
"01a62080-145d-11ea-bc2e-af4fadfbe670.iso",
            "date": "1575220047925"
          },
          {
            "_links": {
              "self": {
                "href":
"<gw:uri>/v1/configurations/2d2b4830-144d-11ea-8f2c-01148a51cc6a"
              }
            },
            "id": "2d2b4830-144d-11ea-8f2c-01148a51cc6a",
            "iemsId": "2c8c2200-144d-11ea-8f2c-01148a51cc6a",
            "metadata": "<json-string>",
```

```
        "version": 2,  
        "physicalResourceName":  
"2cb49190-144d-11ea-8f2c-01148a51cc6a.iso",  
        "date": "1575213229107"  
    }  
]  
}  
]  
}
```