Mobiliya

Security and Surveillance

December 2017

Table of Contents

© 2017 Mobiliya. All Rights Reserved

1. About this Document

This document contains API reference information for Security and Surveillance solution.

This document also contains Database Design with Entity-Relationship information.

2. Intended Audience

This document is intended for programmer, manager, architect to understand how to utilize reference implementation for Security and Surveillance solution

3. Pre-requisites

Reader should be aware of full stack development to interpret API reference information and Database design.

4. Introduction

This document will provide API reference for below modules:

Backend:

This module will provide api to North bound interfaces like web application and South bound interfaces are for Aggregator and Compute Engine

Aggregator:

Aggregator is part of South bound section. This module will communicate with DVR and provide stream services for various cameras and push images for analytics.

Compute Engine:

Compute Engine is part of South bound section. This module will make analytics and deep learning detection of camera images.

Web app:

Web app is part of North bound section. It will mainly interact with user and backend module for numerous services from UI.

Api Version	/v1/
Base Path	/heimdall/
SOLUTIONURL	http://<URL>
Socket URL	http:// <URL>:5006
Back end MQTT	http:// <URL>:1885

© 2017 Mobiliya. All Rights Reserved

5. Back End:

This module will provide two types of API's

North Bound API (NB):

North Bound API set includes communication between backend module and application layer like web application, mobile application, custom application etc. Classification will help us to identify api which belong to back end to application layer.

South bound API (SB):

South Bound API set includes communication between backend module to lower layer like Aggregator, Compute engines (like Jetson/Movidius/Cloud based DL algorithm). This classification will help us to identify api's which belong to backend to lower layer.

1. To Add Compute Engine (NB)

POST /computeengine

Description

This api is part of North Bound Api set. It is used by web application to onboard compute engine on solution.

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
type	Required	Type of compute engine: Jetsontx2, Movidius, Cloud algorithm	String	Jetsontx2
name	Required	Name/Label of compute engine	String	Jetson 2
detectionAlgorithm	Optional	Detection algorithm running on compute engine	String	Human detection
location	Optional	Location of compute engine	String	Location details: pillar number 25

Responses

Return Value	Description
201	Created compute engine successfully
409	Compute engine already exist

Return format

String

Errors

None

Example

```
request

https://<url>/ComputeEngine


body

{

type: “jetson tx2”,

name:”Jetson 1”,

location:”Pillar 23”

}


response

{

“return”:"success",

“id”:”1234”

}
```

2. To delete Compute Engine (NB)

DELETE /computeengine

Description

This api is part of North Bound Api set. It is used by web application to offboard compute engine on solution.

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
computeEngineId	Required	Compute engine ID to delete	String	123

Responses

Return Value	**Description**
201	Deleted compute engine successfully
409	Compute engine is in use, please remove dependent camera/algorithm

Return format

String

Errors

None

Example

```
request

https://<url>/ComputeEngine

body

{

“computeEngineId”:”123”

}



response

{

“return”:"success"

}
```

3. To Add Aggregator (NB)

POST /aggregator

Description

This api is part of North Bound API set. It is consumed by web application to onboard Aggregator to solution.

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
name	Required	Name/Label of aggregator	String	AggregatorDvr2
streamingUrl	Required	Base URL of DVR/NVR to stream camera	String	<valid URL>
channelId	Required	Number of total camera supported	String	16
location	Optional	Location of DVR/NVR	String	Location details: pillar number 25

Responses

Return Value	Description
201	Created aggregator successfully
409	aggregator already exist

Return format

String

Errors

None

Example

```
request
https://<url>/aggregator


body
{
“name”:”aggregatorDvr2”,
“streamingUrl”:
“rtsp://admin:admin@192.168.1.20/user=admin&password=admin&channel=1&stream=0.sdp?”,
“channelId”:”32”,
“location”:” Pillar 23”
}


response
{
“return”:"success",
“aggregatorId”:”1234”
}
```

© 2017 Mobiliya. All Rights Reserved

4. To offboard Aggregator (NB)

DELETE /aggregator

Description

This api is part of North Bound API set. It is consumed by web application to offboard Aggregator to solution.

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
aggregatorId	Required	ID of aggregator	String	1234

Responses

Return Value	**Description**
201	deleted aggregator successfully
409	aggregator in use

Return format

String

Errors

None

Example

```
request

https://<url>/aggregator


body

{

“aggregatorId”:”123”

}


response

{

“return”: “deleted successfully",

}
```

5. To Add Camera (NB)

POST /cameras

Description

This is part of North Bound API set. To add camera to solution from web application

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
name	Required	Name/Label of camera	String	receptionCamera
aggregatorId	Optional	DVR where this camera belongs to	String	123
type	Required	Type of camera : IP/IR	String	ip
computeEngineId	Optional	Compute engine where we would like to process camra image	String	25

Responses

Return Value	Description
201	Created camera successfully
409	camera already exist

Return format

String

Errors

None

Example

```
request

https://<url>/camera


body

{

“name”:”receptioncamera”,

“aggregatorId”:”123”,

“computeEngineId”:”23”,

“type”:”ip”

}


response

{

“return”:"success",

“cameraId”:”1234”

}
```

6. To delete Camera (NB)

DELETE /camera

Description

This is part of North Bound API set. To remove camera from solution from web application

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
cameraId	Optional	Camera id to be removed	String	25

Responses

Return Value	**Description**
201	removed camera successfully
409	camera already in use

Return format

String

Errors

None

Example

```
request

https://<url>/camera



body

{

 “cameraId”:”123”

}



response

{

“return”: “removed successfully"

}
```

7. To register Algorithm (SB)

POST /devices/computeengine/algorithm

Description

This is part of South Bound API. Compute engine will register its capability in terms of algorithm supported to backend by using this api. Below table is considered by Backend for Algorithm supported. Compute engine need to pass parameter id based on below table.

Algorithm	ID
Human Detection	1
Face Detection	2
Object Detection	3

Below table shows different shapes is supported by algorithm

Shape	ID
Rectangle	1
Line	2
Triangle	3
Circle	4

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
algorithmId	Required	Algorithm id for respective name	integer	1
fps	Optional	FPS supported by algorithm	integer	2
computeEngineId	Required	Compute engine where algorithm is running	String	23
supportedShapes	Required	Shape supported by algorithm	String	[{1: rectangle}, {2: line}]

Responses

Return Value	Description
201	register algorithm successfully
409	Algorithm is already existing on selected compute engine

Return format

String

Errors

None

© 2017 Mobiliya. All Rights Reserved

Example

```
request

https://<url>/devices/computeengine/algorithm


body

{

“algorithmId”:”1”,

“fps”:”2”,

“computeEngineId”:”12”,

“supportedShapes”:” [{1:rectangle},{2:line}]”

}


response

{

“return”:"success",

}
```

8. To unregister Algorithm (SB)

POST /devices/computeengine/algorithm

Description

This is part of South Bound API. Compute engine will unregister its capability in terms of algorithm supported to backend by using this api. Below table is considered by Backend for Algorithm supported. Compute engine need to pass parameter id based on below table.

Algorithm	**ID**
Human Detection	1
Face Detection	2
Object Detection	3

Requires authentication

Valid bearer token should be provided

Resource Information

Type	**HTTP/HTTPS**
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
algorithmId	Required	Algorithm id for respective name	integer	1
computeEngineId	Required	Compute engine where algorithm is running	String	23

Responses

Return Value	**Description**
201	unregister algorithm successfully
409	Invalid algorithm request

Return format

String

Errors

None

Example

```
request

https://<url>/devices/computeengine/algorithm



body

{

“algorithmId”:”1”,

 “computeEngineId”:”12”

}



response

{

“return”:"success",

}
```

9. To get list of supported Algorithm (NB)

GET /supportedalgorithms

Description

This is part of North Bound API. From web application we can get list of algorithms like human detection, face detection, object detection etc. supported by particular compute engine. Compute engine might be Jetson device, movidius or cloud based algorithm.

Below table is considered by Backend for Algorithm supported

Algorithm	ID
Human Detection	1
Face Detection	2
Object Detection	3

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
ComputeEngineId	Required	Compute engine where algorithm is running	String	23

Responses

Return Value	Description
201	List of algorithms supported
409	Invalid compute engine

Return format

String

Errors

None

Example

```
request

https://<url>/supportedalgorithms

body

{

 “computeEngineId”:”12”

}



response

{

“algorithmId”:"1"

}
```

10. To Enable/Disable Algorithm (NB)

POST /algorithm

Description

This is part of North Bound API. From web application we can enable/disable compute engine algorithm like human detection, face detection, object detection etc.

Below table is considered by Backend for Algorithm supported

Algorithm	**ID**
Human Detection	1
Face Detection	2
Object Detection	3

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
algorithmId	Required	Algorithm id for respective name	integer	1
fsp	Optional	FPS selected by use	String	2
computeEngineId	Required	Compute engine where algorithm need to be enable/start	String	23
action	Required	Enable/disable (1,0 respectively)	integer	1

Responses

Return Value	Description
201	Enabled/disable algorithm successfully
409	Algorithm is already existing on selected compute engine

Return format

String

Errors

None

Example

```
request

https://<url>/algorithm


body

{

“algorithmId”:”1”,

“fps”:”2”,

“computeEngineId”:”12”,

“action”:”1”

}


response

{

“return”:"success",

}
```

11. Web Socket raw image (NB)

Description

This is part of North Bound API set. This feature helps to communicate the raw image from the backend to the webapp.

Assumption:

Connected to SOCKETURL

Topics

Subscriber topic - rawImage

Requires authentication

Secure web socket

Resource Information

Type	Web socket
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
imageBase64Payload	Required	The raw image to be displayed on webapp for bounding box selection	BSON	Base64 payload
rawImageName	Required	Raw image name	string	<image name>

Responses

NA

Return format

NA

Errors

None

Example

Connection request

https://<SOCKETURL>

Topic to subscribe

rawImage

Response Body

{

“rawImageName”:<image name>,

“imageBase64Payload”: <Base64 payload>

}

12. Web Socket Result image/metadata Update (NB)

Description

This is part of North Bound API set. This feature helps to communicate the result image (along with result metadata) from backend to the webapp.

Assumption:

Connected to SOCKETURL

Topics

Subscriber topic - liveImage

Requires authentication

Secure web socket

Resource Information

Type	Web socket
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
imgBase64	required	The result image to be displayed on webapp	BSON	Base64 payload
bBox	required	Bounding boxes of detected objects	string	{“x”:”10”,“y”:”100”,“x2”:”150”,“y2”:”157”}
Result	required	Number of detections with respect to area of interest	integer	{“result”: [“1”]}

Responses

NA

Return format

NA

Errors

None

© 2017 Mobiliya. All Rights Reserved

Example

Connection request

https://<SOCKETURL>

Topic to subscribe

liveImage

Response Body

```
{
“result”: [“2”],
“bBox”:[{“x”:”24”,“y”:”50”,“x2”:”67”,“y2”:”100”},{“x”:”150”,“y”:”150”,“x2”:”198”,“y2”:”450”}],
“imgBase64”: <Base64 payload>
}
```

13. Web Socket add camera response (NB)

Description

Whenever the notification from camera is received from aggregator, this socket response will notify the webapp if the camera is valid or not. This is part of North Bound API set.

Assumption:

Connected to SOCKETURL

Topics

Subscriber topic - addCameraResponse

Requires authentication

Secure web socket

Resource Information

Type	Web socket
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
status	required	If 0 - Invalid entry <camera name> 1 - Camera added <camera name>	string	{“status”: “1”}
cameraId	required	Camera id	string	{“camname”:<cameraid>}

Responses

NA

Return format

NA

Errors

None

Example

Connection request

https://<SOCKETURL>

Topic to subscribe

addCameraResponse

Response Body

```
{

“status”:”1”,

“cameraId”: <camera id>

}
```

14. Web Socket Background camera results (NB)

Description

This feature helps to communicate the results of the background streaming cameras, from backend to the webapp. This is part of North Bound API set.

Assumption:

Connected to SOCKETURL

Topics

Subscriber topic - backgroundCameraResults

Requires authentication

Secure web socket

Resource Information

Type	Web socket
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
cameraName	required	Name of the camera which is streaming in the background	string	{“camname”:”Reception”}
result	required	Number of detections with respect to area of interest	integer	{“result”: “1”}

Responses

NA

Return format

NA

Errors

None

© 2017 Mobiliya. All Rights Reserved

Example

Connection request

https://<SOCKETURL>

Topic to subscribe

backgroundCameraResults

Response Body

```
{
“result”:”3”,
“cameraName”: <Camera name>
}
```

15. Web Socket notifications (NB)

Description

This feature notifies the messages from backend to the webapp. This is part of North Bound API set.

Assumption:

Connected to SOCKETURL

Topics

Subscriber topic - notification

Requires authentication

secure web socket

Resource Information

Type	Web socket
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
notification	required	Notification to webapp	NA	{“message”: “Camera is down”}

Responses

NA

Return format

NA

Errors

None

© 2017 Mobiliya. All Rights Reserved

Example

Connection request

https://<SOCKETURL>

Topic to subscribe

notification

Response Body

```
{
“message”: ”Camera is down”
}
```

© 2017 Mobiliya. All Rights Reserved

16. POST getRawImage (NB)

Description

This is North Bound api. Web app would request backend to provide RawImage. This API notifies the backend to fetch the raw image from the aggregator and the image will be received on socket.

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
cameraId	required	Unique ID generated for the device	String	12
feature	required	To specify the detection algorithm	String	“personDetection”
streamingUrl	required	Camera URL to stream the camera	String	<camera Url>

Responses

Return Value	Description
201	Image send successfully
401	Invalid request

Return format

String

Errors

None

Example

```
request
https://<url>/api/getRawImage


body
{
 “cameraId”: <Unique ID>,
“feature”: “personDetection”,
“streamingUrl”: <Camera URL>
}


Response
NA
```

17. MQTT get raw image (SB)

Description

This is part of Southbound API set. This feature publishes the required data to aggregator via topic getRawImage (Backend→Aggregator) and receives the base64 raw image from the aggregator via topic rawMQTT (Aggregator→Backend).

Topics

Publisher topic - getRawImage

Subscriber topic - rawMQTT

Requires authentication

Certificate based

Resource Information

Type	MQTT
Response formats	JSON

Parameters (for getRawImage)

Name	**Required**	**Description**	**Data Type**	**Example**
cameraId	required	Unique ID generated for the device	String	12
feature	required	To specify the detection algorithm	String	“personDetection”
streamingUrl	required	Camera URL to stream the camera	String	<camera Url>

Parameters(for rawMQTT)

Name	**Required**	**Description**	**Data Type**	**Example**
imgName	required	The image name	String	<image name>
imgBase64	required	Base64 payload of an image	BSON	<image payload>

Responses

NA

Return format

NA

Errors

None

Example

Connection request

https://<MQTT url>

Topic - getRawImage

Request Body

```
{
“cameraId”: <Unique ID>,
“feature”: “personDetection”,
“streamingUrl”: <Camera URL>
}
```

Topic - rawMQTT

Request Body

```
{
“imgName”: <image Name>,
“imgBase64”: <Base64 payload>
}
```

18. POST getResult (SB)

Description

This is part of Southbound API set, this is consumed by Compute Engine to post result of Image to Backend(ComputeEngine→ Backend).

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
imgName	required	The image name	String	<image name>
bBox	required	The bounding box for the detection	String	{“bbox”:[{“x”:”10”,“y”:”100”,“x2”:”115”,“y2”:”300”}]}
Result	required	Number of detections with respect to area of interest	integer	{“result” : [“1”]}

Responses

Return Value	**Description**
201	Result received successfully
401	Invalid request

Return format

String

Errors

None

© 2017 Mobiliya. All Rights Reserved

Example

```
request

https://<url>/api/getResult


body

{

   “imageName”: <image Name>',

   “result”: [“2”],

“bBox”:[{“x”:”24”,“y”:”50”,“x2”:”67”,“y2”:”100”},{“x”:”150”,“y”:”150”,“x2”:”198”,“y2”:”450”}]

}

Response

{

“201”:”Image result received”

}
```

Refer to "Web Socket Result Update” section to cross-verify this functionality

19. To register Aggregator (SB)

POST /devices/aggregator

Description

This api is part of South Bound API set. It is used by Aggregator module. On boot aggregator will register itself with solution by using this api.

Requires authentication

Valid bearer token should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
aggregatorName	required	The aggregator name	String	<aggregator name>
url	required	Base URL of Aggregator	String	<url>
Ip	required	IP address of Aggregator	String	192.168.123.12
macAddr	Optional	Mac id of Aggregator	String	12:23:23:23:43:34
location	Optional	Physical location where aggregator is installed	String	Floor3
channelId	Optional	Channel id for DVR/NVR	string	32

Responses

Return Value	**Description**
201	deleted aggregator successfully
409	aggregator is whitelisted

Return format

String

Errors

None

Example

```
request

https://<url>/devices/aggregator



body

{

“aggregatorName”:”aggregatorDvr2”,

“url”:
“rtsp://admin:admin@192.168.1.20/user=admin&password=admin&channel=1&stream=0.sdp?”,

“ip”:”192.158.123.23”,

“macId”:”23:23:23:23:23”,

“channelId”:”32”,

“location”:”Pillar 23”

}



response

{

“return”: “registered successfully",

}
```

20. POST Unregister aggregator (SB)

POST /aggregators

Description

This api is part of South Bound API set. It is used by Aggregator module. On exit sequence aggregator will de-register itself with solution by using this api.

Requires authentication

Valid bearer token should be provided in the headers

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
aggregatorId	required	The aggregator id	String	Agg23

Responses

Return Value	**Description**
201	Aggregator un-registered successfully
401	Invalid request

Return format

String

Errors

None

Example

```
request

https://<url>/aggregators


Request Body

{

“aggregatorId”:”agg23”

}


Response

{

201 : unregister successfully

}
```

21. POST Register Compute Engine (SB)

POST /devices/computeengines

Description

This API helps to register Compute Engine. This is part of South Bound API set. It is used by compute engine to register itself in bootup sequence.

Requires authentication

Valid bearer token should be provided in the headers

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
name	required	The name of device	String	<device name>
deviceType	required	Type of device	String	<device type>
timestamp	required	Current timestamp	date	<epoch timestamp>
macId	required	Mac id of device	String	<mac id>
ipAddress	required	Ip Address of device	String	<ip>
location	Optional	Physical location where aggregator is installed	String	Floor3

Responses

Return Value	Description
201	Compute engine registered successfully
401	Invalid request

Return format

String

Errors

None

Example

```
Request:   https://<url> /devices/computeengines


Request Body

{

“name” : “jetsontx2”,

“deviceType” : “jetson”,

“timestamp”:”123123123”,

“macId” : “1231231123”,

“ipAddress”:”192.168.21.21”,

“location”:”floor3”

}


Response

201 : Created
```

22. POST Unregister Compute Engine (SB)

POST /devices/computeengines

Description

This API helps to deregister Compute Engine. This is part of South Bound API set. It is used by compute engine to deregister itself in shutdown sequence.

Requires authentication

Valid bearer token should be provided in the headers

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
computeEngineId	required	Id of compute engine	String	12

Responses

Return Value	**Description**
201	Compute engine deregistered successfully
401	Invalid request or in use

Return format

String

Errors

None

Example

```
Request:   https://<url> /computeengines


Request Body

{

“computeEngineId”:”12”

}


Response

201 : unregistered
```

23. GET Maximum number of Camera Supported (SB)

POST /computeengines/camerasupported

Description

This API is part of South Bound API set. It is mainly used by compute engines to update capability of how many cameras can be supported by each compute engine. This is runtime capability show up by compute engine.

Requires authentication

Valid bearer token should be provided in the headers

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
Id	required	Id of compute engine device	String	123
timestamp	required	Current timestamp	date	<epoch timestamp>
macId	Optional	Mac id of device	String	*00:0a:95:9d:68:16*
numberOfCameraSupported	required	Number of camera supported by compute engine	Integer	3
location	Optional	Physical location where aggregator is installed	String	Floor3

Responses

Return Value	**Description**
201	Compute engine updated successfully
401	Invalid request

© 2017 Mobiliya. All Rights Reserved

Return format

String

Errors

None

Example

```
“return”:"success",

Request:   https://<url> /computeengines/camerasupported



Request Body

{

“id” : “123”,

 “timestamp”:”123123123”,

“macId” : “1231231123”,

“numberOfCameraSupported”:”2”,

“location”:”floor3”

}



Response

201 : updated
```

24. MQTT heartbeat update status (SB)

Description

This is part of South Bound Api. This feature enables devices such as compute engine, aggregator, etc to publish if they are alive or in running state.

Topics

Publisher topic - ping

Subscriber topic - ping_ack

Requires authentication

Certificate based

Resource Information

Type	MQTT
Response formats	JSON

Parameters(for ping)

Name	**Required**	**Description**	**Data Type**	**Example**
aggregatorId	optional	Aggregator Id	String	123
computeEngineId	optional	Compute Engine Id	String	23

Parameters(for ping_ack)

Name	**Required**	**Description**	**Data Type**	**Example**
aggregatorId	optional	Aggregator Id	String	23
computeEngineId	optional	Compute Engine Id	String	32

Responses

NA

Return format

NA

Errors

None

Example

Connection request

https://<MQTT url>

Topic - ping

Request Body

```
{

“aggregatorId”:”123”,

“computeEngineId”: ”123”

}
```

Topic - ping_ack

Request Body

```
{

“aggregatorId”: ”123”

}
```

25. MQTT check camera (SB)

Description

This is part of South bound api set. This is used by backend to ensure the newly added camera is validated by aggregator.

MQTT Topics

Publisher topic - checkCamera

Subscriber topic - checkCameraResponce

Requires authentication

Certificate based

Resource Information

Type	MQTT
Response formats	JSON

Parameters (for checkCamera)

Name	**Required**	**Description**	**Data Type**	**Example**
cameraId	required	Unique ID generated for the device	String	Unique ID
streamingUrl	required	Camera URL to stream the camera	String	<camera Url>
userId	Optional	User id who has initiated the request	String	123

Parameters(for checkCameraResponce)

Name	Required	Description	Data Type	Example
cameraId	required	Unique ID generated for the device	String	Unique ID
streamingUrl	required	Camera URL to stream the camera	String	<camera Url>
userId	Optional	User id who has initiated the request	String	123
cameraStatus	Required	Status of camera : Online-1 /Offline - 0	Int	1

Responses

NA

Return format

NA

Errors

None

Example

Connection request

https://<MQTT url>

Topic - checkCamera

Request Body

{

“cameraId”: <Unique ID>,

“streamingUrl”: <streamingUrl>,

“userId”: “123”

}

Topic - checkCameraResponse

Request Body

{

“cameraId”: <Unique ID>,

“streamingUrl”: <streamingUrl>,

“userId”: “123”,

“cameraStatus”: “0/1”

}

26. MQTT check multiple camera status: online/offline (SB)

Description

This is part of South bound api set. This is used by backend to ensure the selected camera is online/offline with aggregator. This is runtime check for all added cameras.

Topics

Publisher topic - cameraUrls

Subscriber topic - cameraStatus

Requires authentication

Certificate based

Resource Information

Type	MQTT
Response formats	JSON

Parameters (for cameraUrls)

Name	**Required**	**Description**	**Data Type**	**Example**
cameraId	required	Unique ID generated for the device	String	Unique ID
streamingUrl	required	Camera URL to stream the camera	String	<camera Url>
userId	Optional	User id who has initiated the request	String	123

© 2017 Mobiliya. All Rights Reserved

Parameters (for cameraStatus)

Name	Required	Description	Data Type	Example
cameraId	required	Unique ID generated for the device	String	Unique ID
streamingUrl	required	Camera URL to stream the camera	String	<camera Url>
userId	Optional	User id who has initiated the request	String	123
cameraStatus	Required	Status of camera : Online-1 /Offline - 0	Int	1

Responses

NA

Return format

NA

Errors

None

© 2017 Mobiliya. All Rights Reserved

Example

Connection request

https://<MQTT url>

Topic - cameraUrls

Request Body

```
[
{
“cameraId”: “12”,
“streamingUrl”: <streamingUrl>,
“userId”: “123”
},
{
“cameraId”: “13”
“streamingUrl”: <streamingUrl>,
“userId”: “123”
}
]
```

Topic - cameraStatus

Request Body

```
[
{
“cameraId”: “12”,
“streamingUrl”: <streamingUrl>,
“userId”: “123”,
“cameraStatus”: “0/1”
},
{
```

```
“cameraId”: “13”,

“streamingUrl”: <streamingUrl>,

“userId”: “123”,

“cameraStatus”: “0/1”

}

]
```

27. MQTT To select area of interest and start streaming engine (SB)

Description

This is part of South Bound Api set. This is used by backend to pass on information about area of interest to Compute engine and ask Aggregator to start streaming. This will get confirmation from compute engine and Aggregator by MQTT subscriber topic.

Topics

Publisher topic - startStreaming

Subscriber topic - ackStartStreaming

Requires authentication

Certificate based

Resource Information

Type	MQTT
Response formats	JSON

Parameters (for startStreaming)

Name	**Required**	**Description**	**Data Type**	**Example**
aggregatorId	Required	Aggregator id to start streaming	String	123
computeEngineId	Required	Compute engine selection	String	231
cameraId	required	Select camera	String	Unique ID
bBox	Required	Area of interest from Northbound	Array of objects	
streamingUrl	required	Camera URL to stream the camera	String	<camera Url>
userId	Optional	User id who has initiated the request	String	123

Parameters (for ackStartStreaming)

Name	Required	Description	Data Type	Example
aggregatorId	Required	Aggregator id to start streaming	String	123
computeEngineId	Required	Compute engine selection	String	231
cameraId	required	Select camera	String	Unique ID
userId	Optional	User id who has initiated the request	String	123

Responses

NA

Return format

NA

Errors

None

Example

Connection request

https://<MQTT url>

Topic - startStreaming

Request Body

```
{

“cameraId”: “13”,

“streamingUrl”: <streamingUrl>,

“userId”: “123”,

“bBox”:[{“x”:”24”,“y”:”50”,“x2”:”67”,“y2”:”100”}],

“computeEngineId”:”123”,

“aggregatorId”:”231”

}
```

Topic - ackStartStreaming

Request Body

```
{

“cameraId”: “13”,

 “userId”: “123”,

“computeEngineId”:”123”,

“aggregatorId”:”231”

}
```

28. MQTT to stop cameras (SB)

Description

This is part of South Bound Api set. This is used by backend to stop camera streaming and processing.

Topics

Publisher topic - stopCameras

Requires authentication

Certificate based

Resource Information

Type	MQTT
Response formats	JSON

Parameters (for stopCameras)

Name	**Required**	**Description**	**Data Type**	**Example**
aggregatorId	Required	Aggregator id to stop streaming	String	123
cameraId	Required	Select camera/s	Array	Unique ID
userId	Optional	User id who has initiated the request	string	123
computeEngineId	Required	Compute engine selection	String	231

Responses

NA

Return format

NA

Errors

None

Example

Connection request

https://<MQTT url>

Topic - stopCameras

Request Body

```
{

“cameraId”: [“13”,”45”],

 “userId”: “123”,

“computeEngineId”:”123”,

“aggregatorId”:”231”

}
```

User API (North Bound)

This section will cover below operations with respect to user

Signup

Login

Logout

Forgot password

© 2017 Mobiliya. All Rights Reserved

29. POST Signup (NB)

POST /user/signup

Description

This is North Bound API set. This feature helps to perform signup activity for new user to solution. This api allows new user to sign up on solution portal

Requires authentication

Valid password should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
userName	Required	Username of user	String	mobiliyatest
emailId	Required	Email id to of user, Could you use for activation	String	user@mobiliya.com
mobileNumber	Optional	This could be used for 2FA	integer	9898989898
password	Required	Password for user account	String	
organization	Optional	Organization of user	String	

© 2017 Mobiliya. All Rights Reserved

Responses

Return Value	Description
201	Created user successfully
409	User already exist

Return format

String

Errors

None

Example

```
request

https://<url>/user/signup


body

{

“userName”:”mobiliyauser1”,

“mobileNumber”:”9898988998”,

“emailId”:”user@mobiliya.com”,

“password”:”1d*23234(*”,

“organization”:”mobiliya”

}


response

"success"
```

30. POST login (NB)

POST /user/login

Description

This feature helps to perform login activity for user to solution. This is part of North Bound API set.

Requires authentication

Valid bearer token based

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
userName	Required	Username of user	String	mobiliyatest
password	Required	Password for user account	String	
organization	Optional	Organization of user	String	

Responses

Return Value	**Description**
201	Created user successfully
409	User already exist

Return format

String

Errors

None

Example

```
request
https://<url>/user/login


body
{
“userName”:”mobiliyauser1”,
 “password”:”1d*23234(*”,
“organization”:”mobiliya”
}


Response
{
“return”:"success",
}
```

31. POST reset password (NB)

POST /user/resetpassword

Description

This api is part of North Bound API set. It is used to reset password of user.

Requires authentication

Valid password should be provided

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	Required	Description	Data Type	Example
userName	optional	Username of user	String	mobiliyatest
emailId	Required	Email id to of user, Could you use for activation	String	user@mobiliya.com

Responses

Return Value	Description
201	Rese password successfully
409	User does not exist

Return format

String

Errors

None

Example

```
request
https://<url>/user/resetpassword

body
{
“userName”:”mobiliyauser1”,
“mobileNumber”:”9898988998”,
}

response
"success"
```

32. POST logout (NB)

POST /user/logout

Description

This feature helps to perform logout activity for user to solution. This is part of North Bound API set.

Requires authentication

Valid bearer token based

Resource Information

Type	HTTP/HTTPS
Response formats	JSON

Parameters

Name	**Required**	**Description**	**Data Type**	**Example**
userName	Required	Username of user	String	mobiliyatest

Responses

Return Value	**Description**
201	Created user successfully
409	User already exist

Return format

String

Errors

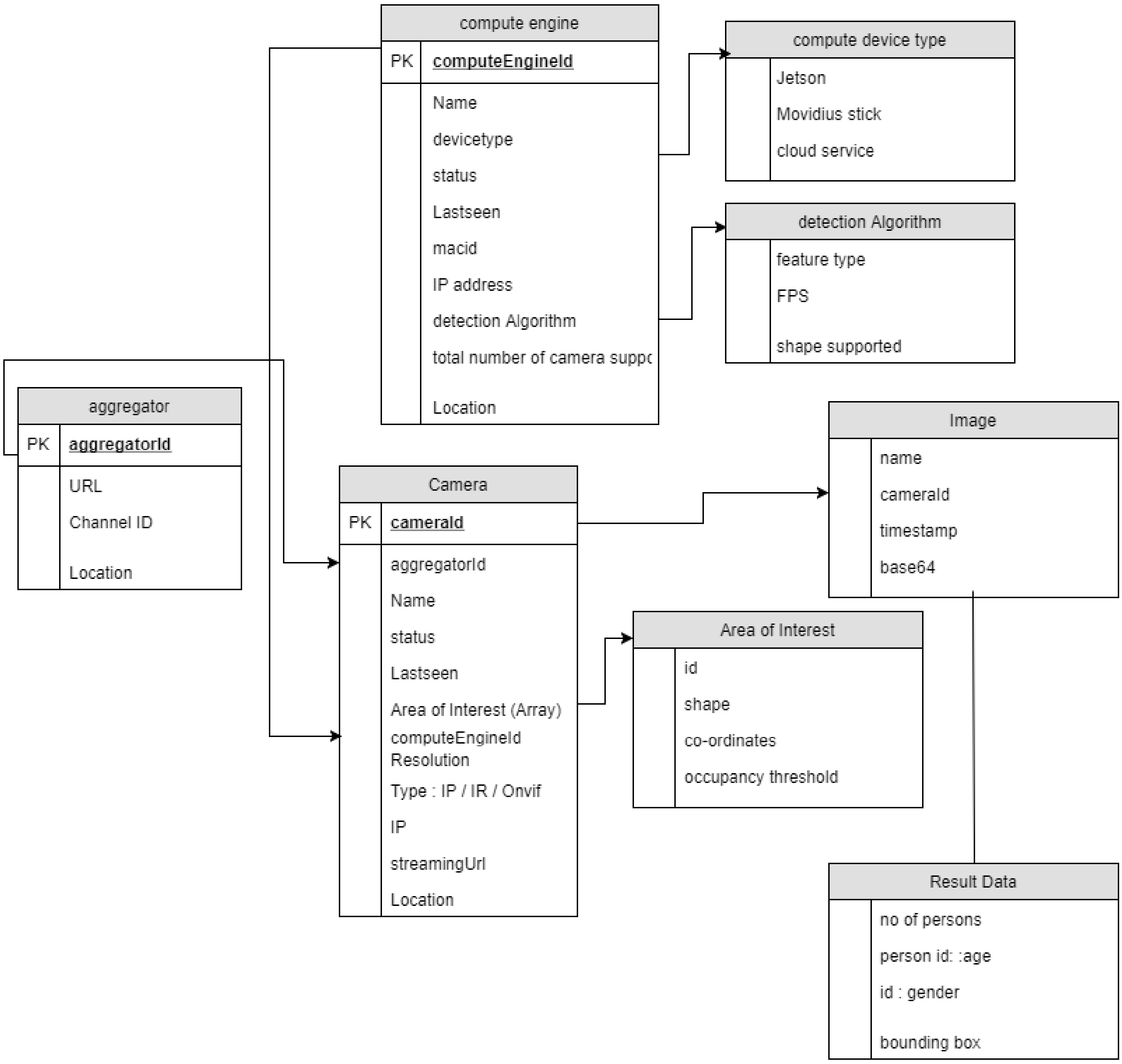
None

Example

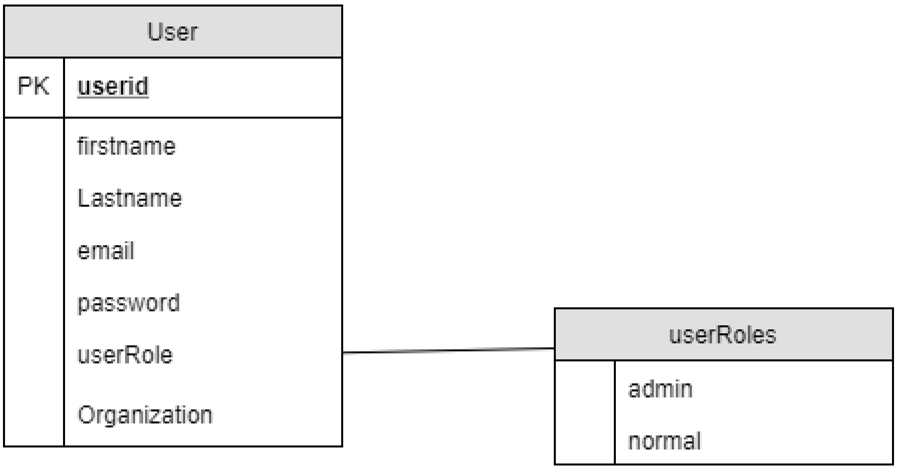
```
request

https://<url>/user/logout



body

{

“userName”:”mobiliyauser1”

}





response

"success"
```

6. Database Design

Following diagram shows database design for Security and surveillance solution



Below diagram shows user management database design E-R diagram.


User
PK
userid
firstname
Lastname
email
password
userRole
Organization
userRoles
admin
normal