

Traffic Influence API Update

TI API Status

API Status (first release supporting LBO):

- **API Proposal** released
- **User Story** released
- **API specification** (YAML) released
- **API documentation** released

Notes on the implementation:

- **Users are identified** by (IPAddress, MSISDN, IMEI, MDN, GPSI) as for the other CAMARA Edge Cloud APIs
- To identify the **geographical area**, Zones and Regions are used as for the other CAMARA Edge Cloud APIs
- The **Application is identified** by a unique identifier, the same used by the other CAMARA Edge Cloud APIs
- **The API is asynchronous and provides notifications** with the status of the implementation of the request ('ordered', 'created', 'active', 'not active', 'error', 'deleted').

Security: the YAML file is **compliant with** the “[Authentication and Authorization Concept for Service APIs](#)” defined in Commonalities. The oAuth2 *clientCredentials* flow is adopted.

Inputs by GSMA OPAG on the TI API

The general **inputs from GSMA** for the **first release** of the TI API are:

- making the API **as simple as possible**. The goal is to provide to developer an API easy to use rather than supporting a great number of use cases.
- considering **in scope only EAS installed in the Telco Operator MEC platform**. Including also external Datacenters indeed requires a process that must be defined.
- having the API invoked for **one user at the time**. To be easily integrated with other CAMARA APIs.

YAML walkthrough

```
##### Paths #####
paths:
  /traffic-influences:
    get:
      tags:
        - Traffic Influence API GET Operation
      summary: Retries existing TrafficInfluence Resources
      description: Reads all of the active TrafficInfluence resources owned by the AF authenticated via OAuth2
      operationId: GetTrafficInfluences
      responses:
        '200':
          description: Returns the information about existing TrafficInfluence resources.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: "#/components/schemas/TrafficInfluence"
        '401':
          $ref: "#/components/responses/Generic401"
        '403':
          $ref: "#/components/responses/Generic403"
        '500':
          $ref: "#/components/responses/GenericError"
        '503':
          $ref: "#/components/responses/Generic503"
        '504':
          $ref: "#/components/responses/BackendConnTimeout"
    post:
      tags:
        - Traffic Influence API POST Operation
      summary: Creates a new TrafficInfluence resource
      description: Gets in input an object containing the intents from the AF and creates a TrafficInfluence resource accordingly. The trafficInfluenceID
        parameter, that is part of the object, must not be valorized when creating a new resource. It is automatically generated by the system and
        returned the response
```

Traffic Influence API GET Operation

- GET** /traffic-influences Retries existing TrafficInfluence Resources
- GET** /traffic-influences/{TIResource_id} read a specific TrafficInfluence resource identified by the TIResource_id value

Traffic Influence API POST Operation

- POST** /traffic-influences Creates a new TrafficInfluence resource

TrafficInfluence API PUT Operation

- PUT** /traffic-influences/{TIResource_id} updates a specific TrafficInfluence resource, identified by the TIResource_id value

TrafficInfluence API Delete Operation

- DELETE** /traffic-influences/{TIResource_id} Delete an existing TrafficInfluence resource

Schemas

Let's open the YAML file

Documentation walkthrough

OPAG-CAMARA Traffic Influence API

Overview

The reference scenario foresees a Service provided by many Application Server instances deployed in different geographical location on Telco Operator Edge sites (MEC sites). The Traffic Influence API provides the fastest routing from the user equipment (e.g. a Smartphone) to the optimal Edge Application Server (EAS) Instance in a specific geographical location. If a Service is offered by a Cloud Instance and by Edge Instances, the API can be used to route a selected set of users on the Edge Instance.

1. Introduction

The Traffic Influence API provides the capability to establish the best connectivity, in terms of latency, in a specific geographical area, between the User Equipment (UE), e.g. the user's smartphone, and the optimal Edge Application Server instance nearby. The Service architecture foresees one or more EASs and a component, the Application Function (AF), that is the API consumer. Invoking the TI API, the AF can create a "TrafficInfluence" resource specifying the desired request. The TI API provider implements the intent specified in the Traffic Influence resource. While the API is usually invoked to activate the fastest routing for any user, it can also be used to request the routing for a specific user. Invoking the API for each user (using the same "trafficInfluenceID"), the TrafficInfluence resource provides the requested routing for a set of users.

2. Quick Start

The usage of the Traffic Influence API is based on the management of a "TrafficInfluence" resource, an object containing the intent requested invoking the Traffic Influence API and that is implemented by the platform configuring the Mobile Network for the optimal routing toward the EAS Instance. The Traffic Influence resource can be created (providing the related parameters that specify the desired intent), queried, modified and deleted. The API is asynchronous, a notification is available providing information about the status of the requested resource.

Let's open the API Documentation file

CAMARA readiness checklist

API Readiness minimum criteria checklist

	No	Deliverables/Criteria	Mandatory	Reference template
✓	1	API Spec	Y	OAS3 (https://spec.openapis.org/oas/v3.0.3)
	2	API Implementation	N	
✓	3	API Documentation	Y	https://github.com/camaraproject/WorkingGroups/blob/main/Commonalities/documentation/DocumentationTemplate.md
✓	4	User Stories	Y	https://github.com/camaraproject/WorkingGroups/blob/main/Commonalities/documentation/template.md
	5	API test cases and documentation	Y	TBD
	6	Tested by atleast 2 operators	Y	
✓	7	Security review	Y	Spec contributions should include a security scheme section that complies with the AuthN&Au techniques agreed in Commonalities.

GSMA Openverse checklist

	Number	Title	Description
✓	1	Working Group Established	Form accepted by backlog WG and a WG is created/assigned for developing the API. Initial CAMARA pages created.
✓	2	User Stories Available	Business user stories are available to inform the design of the APIs
✓	3	API Scope Agreed	From User Stories a technical scope is agreed to understand what is included in the API and what will not work. Eg handset status is connected indicator only in first version.
✓	4	API Parameters Agreed	The parameters that we expect to be sent in the API from the developer, either as compulsory or optional are identified including type eg String, Integer, Boolean
✓	5	API Flow Agreed	How the API will flow from the developer into the API gateway and into network is understood and agreed
✓	6	YAML Created	Initial version of the YAML is created considering the above
✓	7	YAML Consolidated	YAML is agreed and stable. Working group considers YAML complete and agreed for present version and scope
✓	8	API Documentation Available	Documentation is finished for any developer to use the API and for a telecoms architect to start southbound integration into the network
✓	9	Security Review	Security have reviewed the API
? ✓	10	Draft Published	API is considered finished for the present version. It is stable and any operator could take it and deploy onto a test then live network