IoT Big Data Common Enablers for Analytics Services

Version 1.0

22 December 2017

This is a Non-binding Permanent Reference Document of the GSMA

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is confidential to the Association and is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Disclaimer

The GSM Association ("Association") makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Antitrust Notice

The information contain herein is in full compliance with the GSM Association's antitrust compliance policy.

Table of Contents

1 Introduction

1.1 Overview

"Analytics Services" are increasingly being offered by Mobile Network Operators to accompany traditional services such as voice, mobile messaging and connectivity. A good example of such a service would be the processing of mobile network data to produce a population statistics report which is used for transportation planning.

While there are formal technical specifications defined for core mobile network services this is not the case for "Analytics Services". There is a wide variety of types of such services, which are competitive in some cases, and therefore it is not practical to constrain or standardise all aspects of these services.

However, having a common approach to the delivery of analytics is helpful as it helps operators understand the best practices for delivery of analytics results and expectations of potential customers who will be receiving the services. And it is helpful to have to have a common approach to the insights themselves in some cases. For example, some analytics outputs such as geographic areas, weather patterns and population movements often create the foundation for higher level or aggregated “Analytics Services”. Having a common approach in these areas means that analytics insights from multiple operators could be aggregated or insights from different markets compared or combined more easily by a customer.

The inspiration and development of this document has been through collective input primarily within the GSMA IoT Big Data Ecosystem project, where operators including China Mobile, China Unicom, China Telecom, FarEasTone, KT, Orange, Telefonica and Turkcell have contributed requirements and inputs which extend the operator offer to include “Analytics Services” related to IoT and Big Data. In addition, this document has also been developed out of work in the GSMA Big Data for Social Good project where a larger group of operators is working towards supporting the social good ecosystem with a set of 'common insights'.

Examples of "common" analytics services

Mobile networks can support the delivery of a range of common analytics services which have wide application across commercial and governmental domains.

The following use-cases and proposed services have been distilled from the requirements from operators in the IoT Big Data Ecosystem and the Big Data for Social Good projects.

The example use-cases addressed in this document include:

- Providing services to predict short term air quality based on the application of Big Data/ Machine Learning and population/ traffic information derived from mobile networks;
- Providing analysis of historical data in domains such as weather, air quality, water quality and energy usage provided by IoT sensors;

- Using mobile network data to determine population statistics which can be used for the planning of resources including road infrastructure, hospitals, and retail parks;
- Using mobility data derived from the mobile network to determine unusual travel patterns of users such as during the development of an epidemic;
- Using mobility data derived from the mobile network to determine common transportation routes and methods for use in improving public and private transportation;
- Assessing certain risk factors that can be used to improve financial services e.g. banking fraud prevention or insurance based on mobile user location;
- Sending alerts to mobile users who may have been exposed to a risk e.g. disease outbreaks, landslides, flooding, earthquakes.

For these use-cases, it was identified that it would be useful to have a common approach to insights in the following areas:

1. Population density and movements
2. Context data including geography, weather and environment
3. Internet of Things related analytics

For many services, it is assumed reasonably that any mobile network operator can provide statistically significant results because the operator will typically provide service to millions of customers. The main exception to this is alert services, however, as an operator will normally only be able to provide these services to its own customers though it would generally be required to provide this service to the whole population.

1.2 Scope

This document identifies relevant best practices for common enablers in a number of specific areas. The aim is that operators embarking on delivery of "Analytics Services" can accelerate the delivery of such services by leveraging collective experience, and can also help end users of such services who would like to receive consistent results from multiple operators. Due to the variety of “Analytics Services” in practice, in some cases alternative best practice approaches are identified from which operators can select according to the particular service requirements.

This document is not intended to be a formal standard, or binding, acknowledging that the requirements and competitive nature of services are extremely varied.

This document identifies relevant best practices for common enablers in a number of specific areas. The aim is that operators embarking on delivery of "Analytics Services" can accelerate the delivery of such services by leveraging collective experience, and can also help end users of such services who would like to receive consistent results from multiple operators. Due to the variety of “Analytics Services” in practice, in some cases alternative best practice approaches are identified from which operators can select according to the particular service requirements.

The major sections of this document are as follows:

- Section 2 describes the logical architecture that can be used to deliver analytics services;

- Section 3 identifies common foundations for the delivery of analytics services;
- Section 4 proposes mechanisms to support the execution of analytics as background "batch mode" services;
- Annex A provides a set of definitions for parameters that appear in insight requests or responses;
- Annex B provides examples on geo coding using the GeoJSON standard;
- Annex C proposes a simple filtering mechanism that can be applied by applications to select relevant records of interest;
- Annex D defines a number of "de-facto" standard analytics services.

1.3 Excluded from the scope of this document

This document is not intended to cover:

a) Any specific requirements related to national security, policing and other forms of law enforcement, or legal intercept;

b) Use-cases within the normal operations of mobile networks e.g. billing, customer care or network optimisation.

1.4 Definitions

Term	Description
C	The C programming language
Hadoop	Hadoop is an open source project developed under the Apache Software Foundation, it supports distributed and scalable processing of data across clusters of computers and is the foundation of many Big Data analytics platforms. See also http://hadoop.apache.org Related to Hadoop are the complementary Apache Projects & capabilities Hadoop Distributed File System (HDFS) which provides high performance and resilient storage for data; Spark (http://spark.apache.org) – supporting large scale data processing with support also for streaming analytics; Flume (http://flume.apache.org) - a framework for collecting, combining (aggregating) and distributing large amounts of continuously generated data; HBase (https://hbase.apache.org) - a distributed and highly scalable Big Data store; Hive (https://hive.apache.org) - a data warehouse built on Hadoop; Mahout (http://mahout.apache.org) – an environment for creating high performance machine learning applications.
Hashing	A method of converting a data item to a representation of the bits of data such that it is difficult to reconstruct the original data item. In this document hashing is proposed for the encoding of personal data using an SHA2 based algorithm such as SHA-256 or higher. See also https://en.wikipedia.org/wiki/Secure_Hash_Algorithms
Java	The general purpose object oriented programming language created by Sun Microsystems (www.sun.com) as a portable "write once, run anywhere" language. Java is used extensively for web servers and enterprise applications as well as being embedded in many devices.
JavaScript	A programming language similar to C commonly used in web development, able to be run in client environments e.g. web browsers.

Term	Description
Python	Python is a general-purpose programming and scripting language that has become particularly popular for use in data science work. See also https://www.python.org
Salting	'Salting' is a method of adding random data to original data so that it is more difficult to decrypt or 'un-hash' by an attacker looking for common data. Salting is recommended when MSISDNs, IMSIs or related identifiers are hashed for storage by analytics platforms. The process should also be used for encryption of any other sensitive data. See also https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet
URL Encoding	The method of encoding 'special characters' for example spaces, Unicode characters or reserved characters such as & so that they can be used correctly within URLs. See also https://tools.ietf.org/html/rfc3986

1.5 Abbreviations

Term	Description
API	Application Programming Interface – in this context the means by which a third-party application may programmatically request analytics from an analytics server.
CDR	Call Detail Record contains data fields that describe a specific instance of a telecommunication transaction, but does not include the content of that transaction
CSV	Comma Separated Values – a file format that allows information to be exchanged between applications using a compact 'tabular' format where commas are used to separate the data for each column of data.
ESRI Shapefile	A commonly used de-facto standard for distributing Geographical Information System data describing geographical features. Developed by the Environmental Systems Research Institute https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf
FTP	File Transfer Protocol – a protocol used for transferring binary and text format files over a TCP/IP network such as the Internet. Defined in the IETF standard https://tools.ietf.org/html/rfc959 Note that the FTP protocol has poor native security and therefore it should be operated over a secure transmission layer such as an SSH tunnel.
GeoJSON	A variant of the JSON standard used for describing geographical points, lines, and areas. Defined in the IETF standard https://tools.ietf.org/html/rfc7946
GPU	Graphics Processing Unit is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device.
GUID	Globally Unique Identifier – a unique 128-bit identifier used on Windows platforms. See also UUID
GZIP	A file format for compression/ decompression. See also https://en.wikipedia.org/wiki/Gzip
HTTP	The Hypertext Transfer Protocol – the foundation of data communication over the Internet & World Wide Web and used generally for communication between devices and systems. See also https://tools.ietf.org/html/rfc7230 and related standards.

Term	Description
HTTPS	HTTP Secure (HTTPS) adapts HTTP for secure communications over networks including the Internet. Communications are encrypted using Transport Layer Security (TLS) See also https://tools.ietf.org/html/rfc2818
IoT	Internet of Things
IoTBD	Internet of Things Big Data
JSON	JavaScript Object Notation allows data objects to be defined in a portable way allowing simple interchange of data between disparate systems or devices. See also https://www.json.org
MIME	Multipurpose Internet Mail Extensions (MIME) is an internet standard used for exchanging different types of content, formerly over email but generalised to Internet content. This standard includes definition of content type associations with applications and is relevant to this document for content type delivery e.g. in the results of analytics. See also https://en.wikipedia.org/wiki/MIME
REST	Abbreviation for representational state transfer. See RESTful below
RESTful	Web services based on 'REST' (Representational state transfer) provide interoperability between systems and devices over the Internet. This is a common framework for publishing of web services using common HTTP methods including GET, POST, PUT and DELETE. See also https://www.w3.org/TR/ws-arch/
SFTP	A standard for 'Secure File Transfer Protocol' using SSH as a secure transport. See also https://tools.ietf.org/html/draft-ietf-secsh-filexfer-13
SHA	Secure Hash Algorithm, including the SHA-2 family of algorithms designed by the US's National Security Agency. See also https://www.owasp.org/index.php/Guide_to_Cryptography
SOAP	Simple Object Access Protocol is an XML based protocol designed to allow structured information to be transferred between systems such as over the Internet. This allows web services to be developed in a similar way to the use of JSON and RESTful services. SOAP is now more used within enterprise systems and less favoured for web service APIs which are more commonly delivered using JSON/ REST. See also https://www.w3schools.com/xml/xml_soap.asp
SSH	Secure Shell is a secure network protocol enabling services such as secure remote login or SFTP for secure file transfer. See also https://tools.ietf.org/html/rfc4251
TCP (also TCP/IP)	Transmission Control Protocol (/ Internet Protocol) – the lower levels of network protocol used across the Internet and other networks for end to end data communications. See also https://tools.ietf.org/html/rfc793
TLS	Transport Layer Security – crypto protocols that support secure communications over a computer network (including the Internet). See also https://tools.ietf.org/html/rfc5246
URI	Uniform Resource Indicator – a string of characters that identify a resource, often

Term	Description
	provided on a server connected to the Internet. The term URI encompasses both URLs and URNs. See also https://tools.ietf.org/html/rfc3986
URL	Uniform Resource Locator specifies the location of a resource e.g. on a server connected to the Internet, as well as the method of accessing it e.g. https:// See also https://www.w3.org/Addressing/URL/url-spec.txt
URN	Uniform Resource Name – a globally unique identifier for a resource. A URN identifies the resource but not necessarily the method to access it. See also https://tools.ietf.org/html/rfc8141
UUID	Universally Unique Identifier – an identifier which is 128 bits long and is guaranteed to be unique globally across systems and devices which might generate the UUID. See also https://tools.ietf.org/html/rfc4122.html
xDR	Event Data Record
XML	Extensible Markup Language – a syntax for structuring documents that allow devices and computer systems to transfer information in a portable way i.e. independent of operating system. See also https://www.w3.org/TR/xml/

2 Delivery architecture

There are many options supporting the delivery of analytics services. The following are seen as the principle building blocks that will generally be used to develop analytics:

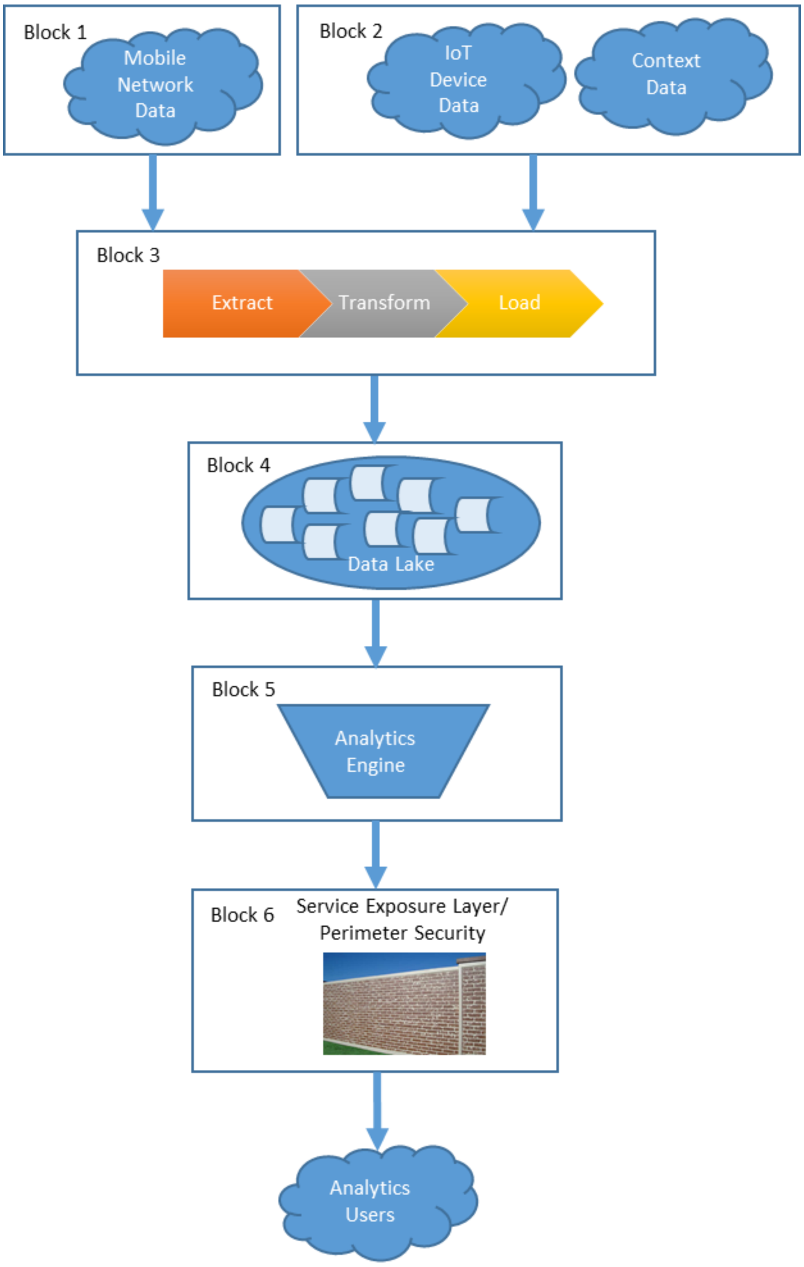


Figure 1 Delivery Architecture

2.1 Building block 1. Mobile network data acquisition

Where analytics are based on mobile network data the source information may be obtained in various ways

Call Detail Records (CDR) are generated by higher level entities in the mobile network e.g. the Mobile Switch generally in order to support customer billing or charging. These CDR records may also be ingested into the analytics platform as they may contain useful information about phone calls and text message activity which are useful to many analytics use-cases;

Event Data Records (xDR) are also generated by higher level entities in the mobile network, but include more advanced data than is in the basic CDR. These allow for example the user location to be known on an ongoing basis rather than at the time the user made a call or sent a text message;

"Network Probes" capture and process signalling information within the mobile network and can provide information useful to mobile network analytics similar to the information available from within xDRs;

Information from operator customer relationship management systems can be used to provide user context information e.g. subscriber age range, which can be used to profile groups of customers in analytics.

2.2 Building block 2. Internet of Things (IoT) device data and context data

Data from IoT devices such as temperature sensors and related context sources e.g. weather information may also be used as sources of information for analytics. Data may be available in a wide variety of sources and using a mix of open and proprietary formats and protocols depending on the source.

To make it easier to receive external data into platforms GSMA have adopted the de-facto standard NGSIv2 for the IoT Big Data project, allowing a common API to be used by analytics platforms when receiving the data from external sources. This API provides a generic publish/ subscribe model for any type of data that can be represented in JSON.

2.3 Building block 3. Data extract transform and load

The mobile network data, IoT data and context data will then usually be 'cleansed' and other forms of pre-processing applied – generically described as an 'extract, transform and load' phase. The exact process to be applied will depend on the source but will generally be developed using bespoke transformation code and/ or use off-the-shelf data transformation tools. The processes applied at this stage will include:

Data quality will be checked – for example to remove data which can be identified as duplicated or erroneous or where data is outside of the expected normal range;

Information which has no use to the analytics services may be removed to reduce data volumes;

Steps may be taken to support the pseudonymization and/or anonymization of any personal data;

Data will be transformed into a format that might be more easily used by analytics tools e.g. transformed to comma separated value format data files.

2.4 Building block 4. Bulk data store

The cleansed/ transformed data will then be stored in a 'data lake' which should provide significant storage capacity for both current and historical data. Data volumes are likely to be of the order of terabytes of data acquired each day and the data lake therefore needs to provide for online storage of the order of petabytes of data.

Common options for the bulk data store will include

Mass storage on distributed platforms such as Apache Hadoop and related Apache technologies such as the Hadoop Distributed File System (HDFS), Spark, Flume, HBase, Hive and Mahout;

Combining NoSQL data stores with traditional SQL databases e.g. MongoDB for unstructured data storage to a NoSQL database along with MySQL, PostgreSQL, Oracle or similar SQL databases for more structured data.

2.5 Building block 5. Analytics engine

Analysis of the data will be performed using various methods and tools

The data may be processed by 'off-the-shelf' analytics or visualisation platforms such as Tableau, SAS, Tibco or Power BI to produce dashboards, reports and data visualisations that can be shared with end customers;

Analytics can be built using various frameworks such as the widely used statistical package R, or statistical packages which can be used with common programming languages such as C, Java or Python such as the scikit-learn package for Python;

More advanced analytics utilising Machine Learning can also be built. For this, frameworks such as Berkeley Caffe, Google Tensorflow, Theano, Apache Mahout can be used – with the possibility of using high performance Graphics Processing Unit (GPU) acceleration for certain combinations of frameworks and algorithms;

Custom visualisations of analytics can be delivered using the likes of the D3.js JavaScript data visualisation library – this can be integrated into client accessible analytics portals;

Streaming analytics can also be provided. For this Apache Kafka is currently the best-known solution.

2.6 Building block 6. Service exposure layer/ perimeter security

For exposure of the analytics service to external clients there needs to be a service exposure layer

The analytics platform must be secured from attacks by deployment of firewalls with best-in-class attack prevention;

Any analytics request/ response must be delivered using secure protocols e.g. HTTPS, FTP over SSH, SFTP or VPNs;

External applications which require access to the analytics services must be required to present relevant access credentials, similarly analytics client users accessing analytics portals must also provide relevant user access credentials. An AAA server (accounting, authorization and authentication) should be available to support application and user access management.

3 Common enablers for delivery of analytics

The following are recommended as the recommended practices for use in the common enablers facilitating delivery of analytics services.

3.1 Methods of requesting/ obtaining analytics

Three main methods are identified for access to analytics depending on use-case. All of these are considered best-practices:

1. Exposing an API which allows a third-party application to request analytics 'on-demand';
2. Providing access to analytics via a web based portal, such as a self-service analytics portal;
3. 'Pushing' analytics to a third-party destination as these are generated.

Variations to the above methods are:

Make analytics available to third-party applications via a (secure) file server e.g. a secure FTP server;

Sending emails to service subscribers to notify them that analytics are available.

Note: That emailing the actual results of analytics or related data is not recommended as a best practice approach due to the following reasons:

Email can be subject to 'man-in-the-middle' eavesdropping and email contents divulged. Encryption of sensitive data should in this case be implemented but this tends to be logistically complicated and is not covered within this scope of this document;

Email can be forwarded[1] to individuals who are not supposed to receive the result and may even be outside of the organisation that the analytics is intended for.

3.1.1 Analytics API

The principle benefit of an analytics API is that this allows an application to request access to the analytics as needed. APIs typically enable new business models more easily through a structured mechanism by which a third-party application can directly request analytics, or subscribe/ unsubscribe to analytics.

The best practice recommendation for API services is to provide these using a 'JSON' based 'RESTful' service, this has become the preferred technology choice for contemporary API services as it is a highly portable mechanism that is easily consumed across the widest range of applications.

If API services are being delivered via legacy enterprise systems there may instead be delivery using SOAP/ XML or even XML over HTTP however these typically add complexity to the third-party application consuming the API.

With JSON over HTTP or SOAP/XML over HTTP essentially the same information can be conveyed but with a more compact representation when JSON is used.

[1] Either by the intended recipient or by the actions of a hacker

If the result of the analytics is a large set of tabular data it is preferable to return the result in a more compact representation as either a CSV (Comma Separated Variable) or as a Microsoft Excel format file[2].

3.1.2 Web based analytics portal

Provision of web based analytics via an analytics portal will typically require a larger investment than exposing an API, but enables users to access or interact with analytics directly rather than by their own development/ integration activities.

There are a range of analytics platforms available that can be used as a basis for a web based analytics portal platform e.g. Tableau, SAS, Tibco or Power BI to name a small selection of the most popular platforms. A fully bespoke analytics portal can also be built from the ground up using web based platforms such as Java/ .NET/ Node.js and visualisation tools such as D3.js.

Important considerations for a web analytics portal:

The platform must be interoperable across prevalent desktop browsers as well as mobile devices and tablets;

It must be possible to administer customer accounts to control access to only the information they need or are subscribed to;

It is desirable to have a large degree of 'self-service' support particularly over basics such as password recovery to minimise the support load as well as sorting & filtering of results;

The platform must be able to handle the large volumes of data that can be generated by and processed from the mobile network and/or IoT devices;

It should be easy to implement new analytics and visualisations on the platform;

There should be some form of data export capability so that customers can download relevant data/ charts for related uses e.g. further processing in Excel or incorporation in written reports.

3.1.3 "Push" analytics

For applications that use regularly updated analytics e.g. a transportation management service which uses aggregated mobile network user data, it may be desirable to have the analytics generated according to a fixed schedule and then pushed from the operator system to the third-party system.

Push analytics can be generated over different timescales:

Streaming analytics can be generated for "near real-time" events – where the analytics platform continuously monitors input sources and/or mobile network conditions and pushes results to external applications when conditions change;

Short term results e.g. average power storage levels recorded by IoT enabled solar panels and reported to electricity grid systems every five minutes for use in short term supply planning;

[2] Note there are certain limitations around numbers of columns and rows depending on the version of Excel used. In Excel 2013/2016 this limitation is 1,048,576 rows by 16,384 columns per worksheet.

Hourly or daily mobile network aggregate statistics e.g. average population density can be generated automatically and pushed to the third-party system immediately after they have been generated;

For push analytics there are several typical methods of delivery:

The widely-used open-source Apache Kafka streaming platform implements its own dedicated and efficient API for the production and consumption of streaming data/ analytics;

Data and analytics can be sent to the third-party system using an FTP (File Transfer Protocol) based interface, usually over a secure interface such as SFTP (Secure FTP) or using a VPN (Virtual Private Network). This is particularly useful when transferring large data files;

Data can be sent from the analytics platform to the third-party system using either JSON or SOAP encoding over an HTTP connection. This is useful for transferring data or analytics which are of a small to medium size i.e. up to a few megabytes;

The analytics platform can generate the analytics then send a notification to the third-party system that the result is available for collection. The notification can be sent using a JSON or SOAP or XML over HTTP based payload or even sent via email. Results can then be read from the analytics platform using for example FTP / Secure FTP or fetched as content using the HTTP protocol. A proposed notification mechanism is defined later in section 4.

3.2 Network protocols

The most common network protocol for obtaining analytics is expected to be through the use of the HTTP 1.1 protocol. More specifically it is expected that requests for analytics will use HTTP secured over a TLS 1.1 or later connection. The HTTP protocol is used as a basis for analytics which are delivered using a web based API (e.g. JSON, SOAP or XML over HTTP) or for delivery of web portal based analytics.

Streaming analytics may be delivered using lower level protocols e.g. Apache Kafka implements its own binary protocol over TCP.

An alternative to HTTP for data transfer is use of a file transfer protocol such as simple FTP or SFTP (FTP over SSH).

3.3 Service endpoints

It is assumed that each specific analytics type is available from a dedicated URL.

In case the analytics server publishes multiple types of analytics from one or more shared URLs the server should allow the requesting application to indicate the desired analytics through the named request parameter 'insight'.

3.4 HTTP Methods supported for analytics APIs

For any analytics service offered via APIs the following are recommended practices for use of the defined HTTP methods:

HTTP Method	Usage
GET	• Make a request for retrieving results of analytics 'by return', passing simple analytics configuration options via URL parameters. The result would be directly computed and fed-back in the HTTP response. • Make a request for retrieving analytics results which are generated according to a pre-defined schedule, passing simple result filter options via URL parameters. The pre-scheduled analytics may be of any type and may have been defined by either the user or the service provider. The filtered analytics result would be fed-back in the HTTP response. • Make a request for retrieving the status and/or some context information of any prior created analytic task, passing analytic task filter (e.g. job identifier) via URL path or URL parameters.
POST	• Alternative method to make a request for retrieving analytics results 'by return'. This is used where complex analytics configurations, filtering criteria, or external data are to be supplied for generation of the analytics, embedded in the HTTP request body . The result would be directly computed and fed-back in the HTTP response. • Create a user-defined analytic task, passing analytics configurations, result notification configurations and possible external data for analytics via the request body. This is applicable to either real-time streaming analytics or batch mode analytics. An assigned identifier of the analytics task and any associated task information would be fed-back in the HTTP response. • Also used in the case an external application requires the analytics platform to perform an action e.g. send a message to multiple users[3].
PUT/Patch	• Change the configuration, externally supplied data or scheduled plan of a previously created analytic task, passing analytic task identifier in the URL with other changes in the request body. The analytics task can be either batch-mode or real-time streaming mode.
DELETE	• Allow an application to request the cancellation and deletion of a previously created analytic task, passing the analytics task identifier in the URL via URL parameters.

3.5 URL encoding

[3] Users are expected to be identified by a pseudonymous identifier that does not disclose any personal data

For all analytics requested via a URL over HTTP all request parameters specified in the URL must be encoded by the application requesting the analytics to ensure that any reserved characters are properly coded to avoid misinterpretation by the analytics server.

URL encoding must not be used within payload data unless the associated request or response content type is 'application/x-www-form-urlencoded'

3.6 HTTP response codes

For any analytics service implemented over HTTP the following HTTP response codes should be used:

HTTP response code	Usage
200 (OK)	• Request for the analytics has been accepted and the result is returned directly in the HTTP response • Batch mode analytics status check call has been processed and the requested analytics generation has been completed. The HTTP response will include additional status information including the address (URL) of the generated analytics result or results
201 (Created)	• The request for the analytics has been accepted but is either only available as a batch output or will require more significant time to generate. The HTTP response will include a URL to a 'status check' service which can be used by the third-party application to poll for completion of the analytics generation • The request to send one or more messages to mobile users has been accepted and is being processed
202 (Processing)	• This response is generated in response to an analytics status check request, it confirms that the analytics generation is still in progress. Additional status information is optionally returned in the HTTP response including the estimated completion ratio and time of completion.
204 (No Content)	• Returned in the case that the third-party application has cancelled an analytics generation request being queued or processed in batch mode
400 (Bad request)	• A request was made for analytics which does not exist at the server, has omitted any required parameters, or specified invalid values for one or more parameters used in generating the analytics. The response body should include error information as shown below.
401 (Unauthorised)	• The application requesting access to the analytics has not provided valid authentication/ authorisation details
404 (Not found)	• The resource requested (which may include the analytics endpoint) does not exist at the server • The requested analytics generation batch mode check service resource could not be found (for cancellation or status check)
405 (Method not supported)	• The application requested access to a service but using an HTTP method not supported for that service e.g. the application requested analytics using an HTTP PUT method but the service is only available when using an HTTP POST method

503 (Server busy or service unavailable)	• There is a temporary problem processing the request due to either server load or error. Retry the request at a later time

For more details on these see https://www.ietf.org/rfc/rfc2616.txt

For any error response (i.e. status codes in the 400 and 500 range) it is recommended to include an error message that can be interpreted by the application or user. For example, if the response is generated as the MIME type application/json the response body would be returned as follows:

```
{"error" : "Missing Required Parameters"}
```

3.7 HTTP Accept & Content-Type headers

Applications must use the HTTP 'Accept' header to indicate the preferred format or formats for analytics outputs. The use of these is defined in RFC 2616[4]

If the application supports multiple output formats it should specify the list of MIME types in order of preference using commas to separate the list. Quality values (also defined in RFC 2616) can also be used to indicate the desired priority;

An application can omit the HTTP Accept header, or specify a MIME type of */* to indicate it will accept any type of output;

Note: That the server generating the analytics may not respond with the desired format, particularly:

If the analytics cannot be generated in the required format but the server is able to generate the analytics in a more suitable format;

If the application requested analytics in 'interactive' mode but the analytics will instead be generated in 'batch' mode.

The analytics server will respond with the HTTP "Content-Type" header to indicate the MIME type associated with any content in the HTTP response.

Example MIME types

MIME Type	**Usage**
text/csv	Content is being transferred using the 'Comma Separated Values' format
text/html	Response content is an HTML page that is expected to be displayed to the end user
text/plain	Content is being transferred as simple inline text
text/tab-separated-values	Content is being delivered using the 'Tab Separated Values' format
application/json	Content is being transferred using JSON (JavaScript Object Notation) format
application/pdf	Content is being delivered as an Adobe Portable

[4] https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

	Document Format file
application/octet-stream	Content is being transferred in a binary encoded format. Normally there will be an additional HTTP header (Content-Disposition) that will indicate the intended filename of the output – allowing the receiving application to correctly handle the response data.
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	Microsoft Excel (Open XML) format
application/vnd.openxmlformats-officedocument.wordprocessingml.document	Microsoft Word (Open XML) format

3.8 Response compression

Analytics platforms should enable the use of HTTP response compression using gzip, where practical. The application requesting analytics should indicate support for gzip decompression by setting the HTTP 'Transfer-Encoding' header set in analytics generation request.

This is particularly helpful for text based results such as comma separated values or JSON as gzip compression can substantially reduce the size/ bandwidth required for transmission of the content.

3.9 Authentication

All requests for analytics must be authenticated to confirm the request is being received from an application or user authorised to access the particular service.

3.9.1 API Authentication

Analytics API requests must be authenticated. Recommended best practice approaches are

Use of the HTTP Basic Authentication standard, with each application identified by a unique pairing of application key and secret used in the HTTP authorization header:

The application key and secret will be generated by an operator authentication platform connected with the analytics service, the key and secret will be shared with the developer of the third-party application consuming requesting the analytics. The business/ technical process for this is outside the scope of this specification[5];

Requests for analytics must only be sent over a secure channel such as using HTTPS (i.e. HTTP over TLS) to maintain protection against Man-In-The-Middle (MITM) attacks.

Use of the OAuth 2 authentication framework (including OpenID Connect/ Mobile Connect) where an HTTP Bearer token is supplied by the application in the HTTP authorization header:

[5] Typical options for sharing application key & secret would include through using a developer portal or sharing the details via email.

The application will obtain an access token to use in the authorization header through a process based on authentication of the application and an optional user consent process – the complete process for this is outside the scope of this document and may vary to some degree between providers;

The application specifies the access token for authentication purposes when invoking the analytics APIs;

If the lifetime (time-to-live) of the access token is exceeded the provider may implement a 'refresh' process to obtain a new access token.

For legacy platforms there may be alternative authentication mechanisms used e.g. supplying a username/ password as URL parameters, or in the request body.

3.9.2 User authentication for analytics portals

Where analytics are provided by a web portal each user should be required to be identified and authenticate. Recommended practices are:

Each user should have their own individual user identifier – which may be their email address;

Users should be required to set a strong password or pass-phrase;

Two factor authentication should be enabled for access to more sensitive analytics – for example use of Mobile Connect as the second authentication factor;

All access to the analytics portal must be over a secure channel e.g. HTTPS over TLS;

Best industry practices for account security should be implemented e.g. see the OWASP (the Open Web Application Security Practice) Authentication Cheat Sheet https://www.owasp.org/index.php/Authentication_Cheat_Sheet .

3.10 Modes of usage for analytics APIs

For analytics service delivered via an API it is expected there will normally need to be two distinct modes of usage:

"On-Demand" – the analytics generation will be processed synchronously i.e. the application requiring the analytics will submit an HTTP request over TLS and once processed will receive an HTTP response which either includes the result data either directly within the HTTP response or via a reference to a URL[6] which can then be accessed by the third-party application to retrieve the analytics response. 'On-demand' analytics may be characterised by the fact they can be delivered quickly i.e. within a matter of seconds, or have a relatively simple result which can be represented or referenced within a small to medium sized response;

Batch – more complex analytics may involve processing significant amounts of data, have a heavy demand on system resources and take significant time to process, in practice this will mean they are generated via 'job queue' processing. The application requesting the analytics will submit its request, and will typically be advised later when the analytics has been generated. Batch processing will typically be required for

[6] This URL should only be accessible in a secure manner.

analytics which have a complex output format e.g. animated video or take minutes or longer to produce.

The requirement for batch mode in the analytics API comes from the fact that the HTTP protocol is synchronous and may be subject to timeouts, server limitations on file lengths, and server resource constraints which could cause issues if analytics require a significant amount of time to generate. If an On-Demand request takes too long to process there may be a disconnection of either the client or the server, and system resources at both ends can be tied up whilst the request is processed leading to potential scaling issues. Therefore, support for a batch processing mode allows application requests to be processed in an orderly and efficient manner.

It is expected that analytics platforms should implement both API usage modes depending on the nature of the analytics services being offered. As a rule of thumb:

If an analytics service completes execution within a minute or less and returns a result which is of the order of megabytes of data or less it is appropriate to deliver as an interactive or "On-Demand" service;

If an analytics service takes minutes or longer to execute or produces a result which is of the order of 10's of megabytes of data or more it is better to be delivered as a batch service.

Analytics platforms are recommended to implement a consistent mechanism for batch/ interactive mode support. See section 4 for recommendations.

3.11 Methods for notifying applications that analytics batch processing is complete

When analytics are being generated via a batch process there is a need to confirm to the requesting application when the generation is complete.

The following is recommended as a best practice for analytics servers which implement batch generation processes:

Each request for batch analytics should cause an analytics job to be queued by the analytics server. Each 'queued job' should be identified using a unique job identifier which is stored by the analytics platform and provided to the application requesting the relevant analytics;

The analytics server should expose 'batch job' status check service allowing the application which requested the analytics to check if the generation is complete. This can be invoked via a polling method. See section 4 for the proposed API semantics for this check service.

The analytics server should also provide a 'batch job' cancellation service which allows the application which requested the analytics to request the job is cancelled. See section 4 for the proposed API semantics for this cancellation service.

The analytics platform should also implement a 'callback' mechanism where the application requesting the analytics will receive a server notification once the analytics generation is complete. See section 4 for the for the proposed API semantics for the callback mechanism.

3.12 Geo coding of areas

A number of the analytics generated from mobile networks are expected to be based on processing the location of IoT devices and/or mobile devices. For example, in an IoT context the analytics may provide electricity usage information broken down by geographical districts. Another example is the use of aggregate user movements in a city to help improve the routing of public transportation.

The proposed service definitions (Annex D) list different methods by which applications can specify custom areas with geographical bounds of interest, however, this flexibility may not be supported by all analytics servers or for all analytics. In addition, there may need to be certain restrictions on the granularity of such geographical areas to protect user privacy or commercial confidentiality.

Best practice approaches recommended for implementation in analytics servers comprise the following

For servers which implement 'pre-defined' areas:

The application can request results which are limited to a list of target areas or target area pairs relevant to the specific analytics being generated;

The application can request results at different geographical levels e.g. Administrative Levels, Post or Zip Code areas etc. that provide aggregation levels;

The application can query the geographic bounds of one, multiple, or all target areas. Geographic bounds can be returned in a variety of common formats including GeoJSON, Well Known Text ("WKT") or ESRI Shapefiles;

For servers which permit 'custom' areas to be specified by the requesting application:

The application can provide details of a single area, a list of target areas or a list of target area pairs relevant to the specific analytics being generated;

Areas can be defined using GeoJSON or Well Known Text representations with each area associated with a name identifier that will be used in the generated analytics;

The analytics server can reject (by ignoring or returning null values for numeric response fields) any areas which do not meet minimum requirements either because the area is too small, discloses populations or movements involving a sensitive area e.g. military or security forces, or if the number of devices/ population statistics for that area do not meet minimum sample sizes (see the later section 3.14 for information on coarsening approaches).

3.13 Pseudonymisation

To maintain highest levels of individual privacy it is recommended that any storage of data that could be considered personal information is pseudonymised at the earliest opportunity during data storage and processing. Personal information typically includes information such as the real name of the customer, email address, mobile phone number (MSISDN), registered address, registered post code etc. In addition, certain other identifiers e.g. IMSI, IMEI and device IP address can become personal data if associated with other information.

Pseudonymisation is generally achieved by the substitution of an alternative, system generated, identifier to replace one or more sensitive user fields. If the pseudonymised identifier is shared with external parties they are unable to identify personal information about the user.

Note: That many analytics work on aggregated user data so in these cases there is no need to share pseudonymised identifiers with the client receiving the analytics.

Best practice recommendations for Pseudonymisation include the use of one or more of the following:

Use of mechanisms such as a long, random UUID's to substitute for one or more pieces of personal data. 128-bit Type 4 random UUIDs are recommended e.g. 'cb3eb75f-cbbc-414c-a146-624518ed7537' (hex coded);

Use of strong hashing (SHA-256 or better) with 'salting' to one-way encode personal data in a way that cannot practically be reversed;

Generating different pseudonymous identifiers for personal data for each separate user of the analytics;

Generating different short-lived pseudonymous identifiers for personal data between defined periods e.g. a new identifier is generated each day for any given MSISDN and any given application so that the application cannot track the long-term actions of users[7].

In the case that a service may need for example to send a message to a user identified using a pseudonymised identifier the analytics platform should adopt the following recommendations:

There should be a separate data store which provides a reverse mapping from pseudonymised identifier back to the original personal data;

This separate data store must be highly secure, have the most limited access possible to both operatives and connected systems, and employ strong encryption for any stored data;

This data store should not be delivered outside of the operator systems boundary;

Data should be purged from this store at the earliest possible opportunity according to intended uses.

3.14 "Coarsening" of results

If analytics servers provide excess support for drilling down into result data there is a risk that an external party could obtain information about operator systems, commercial facts or mobile users that is not intended as part of providing the analytics.

The following methods are proposed as best practices to protect against such unintended risks.

Note: That the recommendation is to implement as many of these techniques as are appropriate to the use-case and organisation requesting the analytics:

3.14.1 Coarsening reporting periods

It is possible that reporting periods may potentially be varied by the requesting application to obtain levels of detail in analytics which are commercially sensitive e.g. movement patterns

[7] Even though the identify of the user is not known.

between 6pm and 7pm may be used to determine numbers of business customers whereas between 3pm and 4pm may disclose school students.

To mitigate against such disclosures analytics platforms may implement constraints or procedures related to reporting periods such as one or more of the following:

1. Ensure the minimum reporting period is relevant to supported use-cases and operator policies e.g. hourly reporting may not be supported but daily reporting is supported, or hourly reporting is supported but no shorter interval;
2. Ensure the historical period over which reporting can be run is relevant to the use-case e.g. a use-case which aids the deployment of taxis may only need access to current user densities rather than needing access to historical data whereas a transport planning application would use historical data;
3. Rather than provide historical data to allow the application to 'correct' for public holidays (Christmas, Eid, Diwali) instead perform such corrections in the analytics platform;
4. Trusted agencies e.g. government departments may be allowed access to more granular reporting than less trusted customers e.g. delivery companies.

3.14.2 Minimum relevant population size restrictions

If analytics can be generated that can identify individual or close groups of users by the precise nature of their activities there is a potential risk to user privacy. For this reason, analytic outputs should only include results aggregated over a minimum selected subject population size.

For example, if an analytics report could be generated which looks for travel patterns between two specific geographical areas and within a very specific time period there is the potential to resolve to an individual independently known to move in one of those areas e.g. an individual commuter might be identifiable.

As such operators should implement a minimum set size for aggregated results within analytic outputs discard sets of results which fail to meet this minimum. Whilst the minimum size is dependent on use-case and the local legal and regulatory situation the recommendation is to make the minimum reporting quantity for any result row between 10 and 25 devices or users.

Again, for trusted partners such as government agencies the minimum reporting quantity may be set at a lower limit.

3.14.3 Coarsening geographical resolution

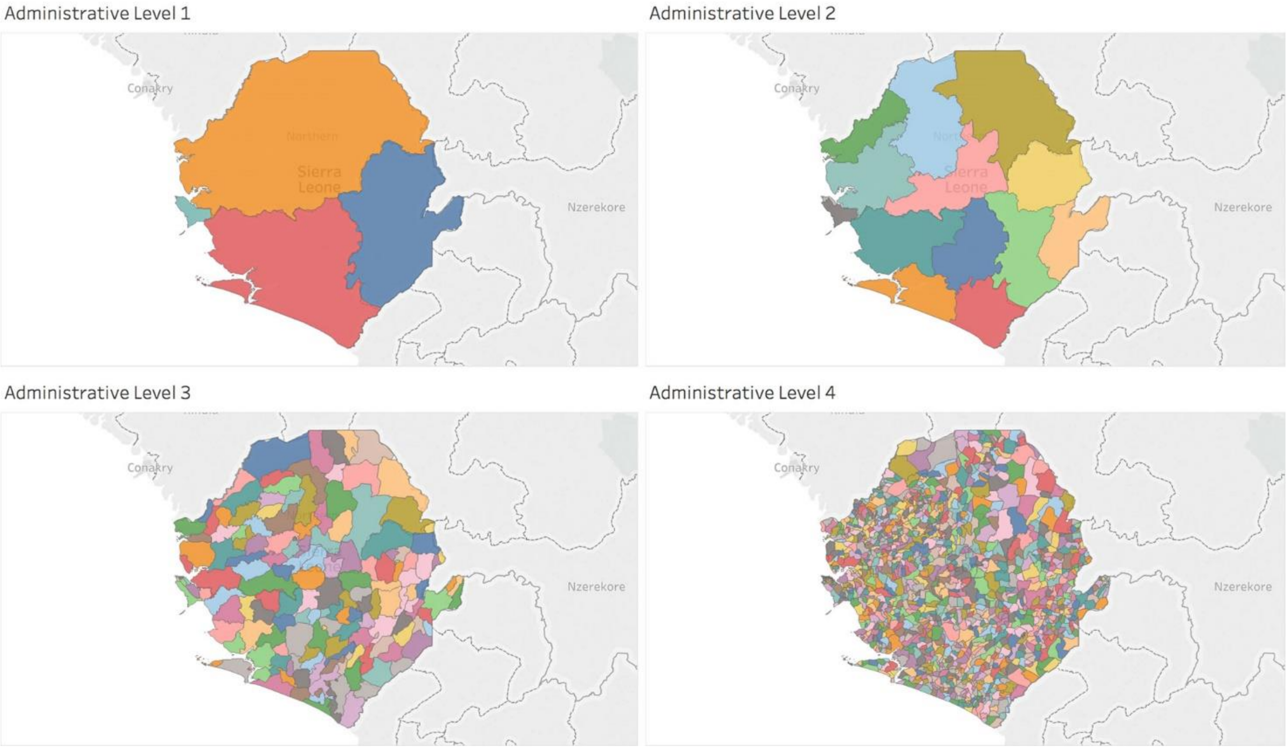
There is the potential to compromise user privacy and/or commercial confidentiality if precise geographic areas are included in the results generated by the analytics platform.

Whilst some use-cases, e.g. locating potentially trapped users in the case of an earthquake, benefit from high precision user location it is recommended that generally the analytics interface should not allow selection or precise reporting of individual user location for privacy reasons.

Additionally, information about the number of users in a particular network cell is information that is likely to be commercially confidential to the operator and it is expected there will be an interest in protecting this information particularly across the whole operator network.

The following measures can therefore be employed:

1. Reports are aggregated geographically to a larger area that encompasses many network cells. For example, the diagram below shows four 'Administrative Levels' for Sierra Leone which can be used to study migration patterns. Analytics platforms can implement their own geographical aggregations relevant to the use-case;
2. Analytics platforms change the reporting point for a cell from the actual location of the network equipment to a representative centre and bounded area describing the cell coverage – this helps protect the location of the cell equipment and is likely to be more useful;
3. Precision is limited according to use-case and end customer, for applications such as retail footfall estimation the analytics platform may provide numbers for only those users in the vicinity of existing or planned retail stores rather than high precision counts across a whole city or country.



Administrative Level 1

Administrative Level 2

Administrative Level 3

Administrative Level 4

3.15 Protecting commercial confidentiality of devices/ user numbers within results

As mentioned briefly above the numbers of devices/ users in a given geographical area is information that is generally commercially confidential. Also, for many use-cases the analytics customer is ultimately interested in the general population rather than the actual number of devices/ users that are served by a specific operator.

There are various 'corrections' to raw device numbers that make sense to apply when reporting user counts, these have the dual benefit of protecting operator commercial information as well as providing more useful information to analytics customers. The recommended correction factors include:

Scale the number of users according to the estimated operator market share in any given area, taking into account also sections of the population who do not own mobiles i.e. younger children, the elderly or very poor;

- Correct for the proportion of devices per user – eliminating ancillary devices such as USB MODEMs, MiFi devices and 'data only' tablet devices;
- Exclude where possible 'shared' devices – which have usage patterns outside of general norms.

For some use-cases it may also be possible to report a relative proportion of people rather than an absolute number of people – therefore for these the analytics output can be reasonable to provide in terms of a percentage of the whole, or percentage change. For example:

- For all the people who spend their day working in the City of London what is the relative breakdown by voting ward across London?
- What is the increase in people visiting London's Oxford & Regent Streets in December compared with November?

Therefore, the best-practice recommendations are

- Ensure the metric output by the analytics server is appropriate for the use-case i.e.
  - Where possible prefer indication of percentages rather than absolute numbers;
  - If percentages are not useful consider instead whether 'delta's are suited to the selected use-case. The baseline can further be set according to a meaningful measure e.g. a seasonal norm/ adjusted for day of week;
- Correct source data to exclude devices which aren't indicative of users – where this is really what the analytics customer wants;
- Consider discarding results that are within normal bounds of variation, e.g. if a usual mobility pattern is 20,000 users/ devices move in or out of a specific city on average each working day +/- 2000 users then an alert or notification would only trigger once the delta exceeds +/- 2000;
- For most use-cases it is best to apply 'market share' correction factors to convert from the number of mobile network users to figures representing the population. For example, if the operator identifies a delta compared to the norm of 200 devices/ users in an area where the operator evaluates it has 40% market share but its national market share is 35% the operator would opaquely scale this result to 175 [200*35%/40%].

3.16 User consent

In the case that services are provided to individuals there may be a need to obtain or manage user consent, for example,

1. If the use-case requires the release of any personal information to a third-party analytics provider or other end organisation. An example of this might include the end user agreeing to share analytics on their historical location positioning with an insurance company as part of a risk assessment during policy quotation;
2. If the use-case might involve the sending of targeted alerts to an end-user based on for example an outbreak of flu in their home or work areas so that they can receive vaccinations;
3. The user might opt-in to receive alerts of special promotions from designated retail stores, sports venues, restaurants and some of these might be location enabled. The partner company would not be informed generally about the position of the customer, but when the user is within a specific range of a venue might receive a notification

including the hashed mobile number of the user along with any relevant market segmentation information to allow the provider to send an appropriate response to the user (refer to section 3.13 which explains the use of hashing when pseudonymising user data).

Note: That in none of the above cases is it expected the detailed mobile user's movements will be identified to the end organisation, this information is securely maintained within the analytics platforms and only the results of the analysis are shared.

In cases 1 and 3, above it is recommended that Mobile Connect is used to provide such a 'one-time' consent confirming that the specific mobile user has granted permission to share information about their 'risk'.

In case 2, it is expected the operator sending alerts (typically via a text message) will allow users to opt-out of receiving further alerts if they reply with the message 'STOP'. In this case users are normally opted-in but are able to opt-out at any time. Operators may be subject to local regulations which govern requirements on consent for text based services.

4 Support for 'batch mode' generation of insights

When a request is accepted by the analytics server which results in the creation of a 'batch job' to generate the insights the server should return details of an insight report generation check service and/or a report generation cancel service e.g. as follows:

HTTP Status Code : 201 (Created)

```
{
"checkService":
      "https://{example.com}/{pathToCheckService}?{requestId}",
      "cancelService":
"https://{example.com}/{pathToDeleteService}?{requestId}"
}
```

In this

- {example.com} is replaced by the fully qualified domain name for the server providing management of batch insight generation services;
- {pathToCheckService} is replaced by the URL path to the service providing a status check for the insight generation job;
- {pathToDeleteService} is replaced by the URL path to the service providing the ability to cancel a queued or in progress insight generation job;
- `{requestId}` would be replaced by a unique job reference to the request.

The analytics server may also support a notification mechanism whereby the requesting application is notified when the batch insight generation process has been completed. Further details of this are provided below to the application

4.1 Check the status of an insight generation request

For any insights which are being generated through a batch processing job the following is recommended as a way of the application requesting the insight being able to check for completion of the insight generation.

Request URL Format

URI: `https://{example.com}/{pathToCheckService}?{requestId}`

HTTP Method:GET

Operation: Query the current status of an insight generation batch job

Request Parameters:

Parameter	Usage	Example
requestId	Specifies a unique request identifier allocated by the analytics server for the specific insight generation request. It is suggested that a Type 4 Random UUID is allocated to the insight generation request.	19c9fc9c-e943-11e6-877b-f3378de460ba

Response Parameters:

Parameter	Usage	Example
status	Indicates the current status of the insight generation request with the following values • "Processing" - the request is queued for processing, or being processed • "Completed" – the results of the request are available • "Cancelled" – the request has been cancelled either by the requesting application or the analytics system	Completed
estimatedCompletionRatio	Numeric value representing percentage completion of the insight generation. Ranges from 0 to 1	0.35
estimatedCompletionTime	The date/time represented in ISO8601 including the timezone when the analytics server expects the insight will be generated. It is recommended that UTC is	2017-09-11T07:54:28Z

	used for the estimated completion time.	
resultURI	Specifies the URL where the application can obtain the results of the insight generation request. Authentication must be used to request the result.	http://example.com/report/AAB123.csv
content-type	Specifies the MIME type of the result data to avoid possible ambiguity.	text/csv
cancellationSource	In the case that the request has been cancelled indicates the source of the cancellation • Application – the request to cancel was made by the requesting application • System – the system cancelled the insight generation request	Application

'Processing' Response

In the case that the insight is still in the process of being generated the report server should respond as follows:

HTTP Status Code : 202 (Accepted)

```
{
    "status": "Processing",
    "estimatedCompletionTime": "2017-09-11T07:54:28Z",
    "estimatedCompletionRatio": 0.35
}
```

'Cancelled' Response

In the case that the insight generation has been cancelled (either by the application or the reporting system) the report server should respond as follows:

HTTP Status Code : 200 (OK)

```
{
    "status": "Cancelled",
    "cancellationSource": "Application"
}
```

Source should be set to "Application" in the case the application requested cancellation of the report, or "Server" in the case the report server cancelled the report for any reason.

'Completed' Response

In the case that the insight generation has been completed, and the results are available the report server should respond as follows:

HTTP Status Code : 200 (OK)

```
{
    "status": "Completed",
    "resultURI": "http://example.com/report/AAB123.csv",
    "content-type": "text/csv"
}
```

In the completed response `resultURI` will indicate the Internet accessible location of the generated result. Access to this should be subject to HTTP authorization.

The `content-type` field indicates the MIME type of the result. If this is set to "`text/html`" it indicates the result is to be accessed using a web browser, otherwise the result will be a downloadable file e.g. "`text/csv`" refers to a comma separated values file that can be imported to an application such as Microsoft Excel.

4.2 Cancel an insight generation request

In the case an application wishes to cancel a previously submitted insight generation batch job (e.g. the user has cancelled a request) the analytics server should support the cancellation through the submission of an HTTP DELETE request to the `'cancelURL'`.

Request URL Format

URI: `https://{example.com}/{pathToDeleteService}?{requestId}`

HTTP Method:DELETE

Operation: Cancel an insight generation request

Request Parameters:

Parameter	**Usage**	**Example**
requestId	Specifies a unique request identifier allocated by the analytics server for the specific insight generation request. It is suggested that a Type 4 Random UUID is allocated to the insight generation request.	19c9fc9c-e943-11e6-877b-f3378de460ba

'Successful Cancellation' Response

In the case that the insight generation could be cancelled the report server should respond as follows:

HTTP Status Code : 204 (No Content)

'Request Not Found' Response

In the case that the insight generation request could not be found the report server should respond as follows:

HTTP Status Code : 404 (Not Found)

```
{
    "message":
   "The insight generation request specified does not exist"
}
```

4.3 Batch job callback / notification mechanism

As described above there is a proposal for a notification mechanism used to indicate insight generation is complete. This is modelled on the OneAPI notification mechanism as used for messaging services.

When an insight has been generated the server shall generate a notification to the callbackURL defined by the application in the insight generation request. This notification will be sent in the form of an HTTP POST request with a 'Content Type' header of 'application/json'. The payload shall also include any callbackData provided by the application when it submitted the insight generation request.

The notification will include the following data

```
{"insightGenerationNotification": {
    "callbackReference": {
        "callbackData": "{string}",
        "callbackURL":
             "https://appserver.example.com/pathToReceiver"
    },
    "status": "Completed",
    "resultURI": "http://example.com/report/AAB123.csv",
    "content-type": "text/csv"
    }
}
```

The HTTP response from the application server will be ignored irrespective of the status code or error information. It is not expected that the application will resend the notification if there is an error.

Annex A Common data dictionary

The following are the names/ usage of a set of insight request parameters that are expected to be used across multiple types of insight reports. See the relevant insights for the applicability of these parameters to that insight

Parameter name	Usage
accept (HTTP header)	Indicates the preferred format for delivery of the

	insight. Common output formats would be CSV (text/csv), JSON (application/json) or Excel (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) for results in a manipulatable format. Web viewable content would be requested using the MIME type text/html. PDF content would be requested using the MIME type application/pdf.
insight	Specifies the name or other 'id' for the insight being requested, this will be based on a name or 'id' agreed between the operator and the consuming application. This is required only if multiple insights are published from the same URL.
mode	Specifies the preference of the application regarding insight generation process. Expected values include • "immediate" – the application would prefer to receive the insight output as part of the HTTP response • "batch" – the application would prefer to submit the request for generation of the insight to a batch processing queue and receive notification once complete
callbackURL	Optionally specified to allow the application to receive notification that batch mode insight generation has been completed.
callbackData	Optionally specified to allow the application to specify context information related to the current request. This context information will be included in the notification sent to the callbackURL.
point	Optionally allows the application to specify the request is for a report covering a single area centred at the specified point. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead. The point defines the WGS 84 (GPS) Latitude and Longitude. This will be used with the optional radius to constrain reporting to devices reckoned to be in/ near the requested area. Note that for privacy or commercial reasons operators may choose not to support arbitrary geographical points but instead may offer reports with pre-defined geographic zones based on administrative levels (such as province,

	municipality, district…).
radius	Optionally used alongside 'point' to specify the radius of the location search. Expected to be specified in units of kilometres (km). If omitted a default value of 1 kilometre is assumed. Note that for privacy or commercial reasons operators may choose not to support arbitrary geographical points but instead may offer reports with pre-defined geographic zones based on administrative levels (such as province, municipality, district…). In addition, operators may limit the minimum or maximum radius that can be used in reporting.
box	Optionally allows the application to specify the request is for a report covering a single area designated by a bounding box. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
customAreas	Optionally allows the application to specify the request is for a report covering multiple custom areas provided by the application. The areas are defined using GeoJSON or WKT (Well Known Text) as named areas. As it is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customAreas.
polygon	Optionally allows the application to specify the request is for a report covering a single area as a closed polygon defined by the application. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
selectedAreas	Optionally allows the application to specify the request is for a report covering selected named areas which are already pre-defined by the analytics server. Area names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request.
baselineStartDateTime	Specifies the starting date/time used to set a baseline period for a given insight e.g. in the population snapshot report this is used to establish 'normally resident' for the population. This value is expected as an ISO 8601 format combined date/time value based on the relevant timezone for the country e.g. '2017-01-25T11:59:03Z'.
baselineEndDateTime	Specifies the ending date/time used to set a

	baseline period for a given insight e.g. in the population snapshot report this is used to establish 'normally resident' for the population. This value is expected as an ISO 8601 format combined date/time value based on the relevant timezone for the country e.g. '2017-01-25T11:59:03Z'.
reportingStartDateTime	Specifies the starting date/time used to set a reporting period for a given insight e.g. in the population snapshot report this is used to establish the range of dates/ times to be included in the output report. This value is expected as an ISO 8601 format combined date/time value based on the relevant timezone for the country e.g. '2017-01-25T11:59:03Z'.
reportingEndDateTime	Specifies the ending date/time used to set a reporting period for a given insight e.g. in the population snapshot report this is used to establish the range of dates/ times to be included in the output report. This value is expected as an ISO 8601 format combined date/time value based on the relevant timezone for the country e.g. '2017-01-25T11:59:03Z'.
reportingInterval	Requests that the insight be generated for successive intervals between reportingStartDateTime and reportingEndDateTime. This can take the following value formats 'd' means report by day. An optional numeric prefix indicates the interval should be for the specified whole number of days. e.g. '4d' means report at intervals of 4 days. 'h' means report by hour. An optional numeric prefix indicates the interval should be for the specified whole number of hours. e.g. '4h' means report at 4 hourly intervals. 'w' means report by week. An optional numeric prefix indicates the interval should be for the specified whole number of weeks. e.g. '4w' means report at 4 weekly intervals. 'm' means report by month. An optional numeric prefix indicates the interval should be for the specified whole number of months. e.g. '4m' means report at 4 monthly intervals. 'y' means report by year. An optional numeric prefix indicates the interval should be for the specified whole number of years. e.g. '4y' means report at 4 yearly intervals.

maxRecords	Indicates the maximum number of records that the application would like to receive in any generated insight report. Note this may not be applicable to all insight reports.
filter	A generic filter specification that can be applied to any result columns of the insight report to constrain the output. This parameter must be sent as part of the JSON formatted HTTP request body using the POST method. The filter syntax is defined in Annex C. Note this may not be applicable to all insight reports and report services may not support all or part of the filter syntax.
order	Specifies the sorting order applied to the insight report. This should contain a list of comma separated column headings which are applied in turn, with the optional prefix character of '+' meaning the column should be sorted in ascending order and '-' meaning the column should be sorted in descending order.

Annex B GeoJSON examples

The GeoJSON format is defined by IETF RFC7946 (https://tools.ietf.org/html/rfc7946)

Within this document the following geometries are adopted:

- 'Point' – which can be used to refer to either a single position:
  - This may refer to the centre of a circular area (contained within a defined radius) used in querying which devices are contained within that area;
  - It may refer to the approximate position of a mobile device – this may have had an error applied.
- 'MultiPoint' – refers to a sequence of positions:
  - This can be used to report a sequence of positions of a mobile device over time. The positions may be approximate/ have had an error applied.
- 'Polygon' – refers to a closed polygon of three or more distinct nodes. The first and last positions must be the same. 'Holes' in the area can also be defined;
- 'MultiPolygon' – contains multiple areas of type 'Polygon';
- 'FeatureCollection' – allows for the definition of a number of geometrically defined points/ areas. This would be used for example to describe the cities/ districts of a country.

Point

```
{
    "id": "Talamau",
    "type": "Point",
    "coordinates": [100.0, 0.0]
}
```

MultiPoint

```
{
    "id": "Talamau to NW",
    "type": "MultiPoint",
    "coordinates": [
        [100.0, 0.0],
             [101.0, 1.0]
    ]
}
```

Polygon

This first example defines a 'filled' polygon i.e. there are no 'holes' within the area described by the co-ordinates:

```
    {
        "id": "Riau",
        "type": "Polygon",
        "coordinates": [
            [
                [100.0, 0.0],
                 [101.0, 0.0],
                [101.0, 1.0],
                [100.0, 1.0],
                [100.0, 0.0]
            ]
        ]
    }
```

This second example defines an outer 'polygon' area which has an internal polygon representing a 'hole'. This could be used for example to represent an outer city area where the central area is excluded from the geometry.

```
    {
        "id": "Riau suburbs",
        "type": "Polygon",
        "coordinates": [
            [
                [100.0, 0.0],
                [101.0, 0.0],
                [101.0, 1.0],
                [100.0, 1.0],
                [100.0, 0.0]
            ],
            [
                [100.8, 0.8],
                [100.8, 0.2],
                [100.2, 0.2],
                [100.2, 0.8],
                [100.8, 0.8]
            ]
        ]
    }
```

MultiPolygon

A MultiPolygon can contain an array of Polygon's, each of which can be either 'solid' or can contain 'holes' as described above.

```
{
    "id": "Riau and environs",
    "type": "MultiPolygon",
    "coordinates": [
        [
            [
                [102.0, 2.0],
                [103.0, 2.0],
                [103.0, 3.0],
                [102.0, 3.0],
                [102.0, 2.0]
            ]
        ],
        [
            [
                [100.0, 0.0],
                [101.0, 0.0],
                [101.0, 1.0],
                [100.0, 1.0],
                [100.0, 0.0]
            ],
            [
                [100.2, 0.2],
                [100.2, 0.8],
                [100.8, 0.8],
                [100.8, 0.2],
                [100.2, 0.2]
            ]
        ]
    ]
}
```

FeatureCollection

```
{
       "type": "FeatureCollection",
       "features": [{
           "id": "Kota Pelalawan",
           "type": "Feature",
           "geometry": {
               "type": "Point",
               "coordinates": [102.0, 0.5]
           },
           "properties": {
               "prop0": "value0"
           }
       }, {
           "id": "Riau",
           "type": "Feature",
           "geometry": {
               "type": "Polygon",
               "coordinates": [
                   [
                       [100.0, 0.0],
                       [101.0, 0.0],
                       [101.0, 1.0],
                       [100.0, 1.0],
                       [100.0, 0.0]
                   ]
               ]
           }
       }, {
           "id": "Riau SE",
           "type": "Feature",
           "geometry": {
               "type": "Polygon",
               "coordinates": [
                   [
                       [102.0, 0.0],
                       [103.0, 0.0],
                       [103.0, 1.0],
                       [102.0, 1.0],
                       [102.0, 0.0]
                   ]
               ]
           }
       }]
   }
```

Annex C Filter syntax

Analytics servers may optionally support filtering of report outputs. The following is suggested as a general-purpose filtering mechanism. This syntax is broadly based on the MongoDB JSON based query mechanism[8]

Directive	Usage	Examples
'field':'value' 'field':value	Matches fields which are equal to string, numeric or Boolean value	'zoneIdentifier':'SW1'

'field':{'$eq':'value'}
'field':{'$eq':value}
	Alternate test for equality to string, numeric or Boolean value	'zoneIdentifier':{'$eq':'SW1'}

'field':{'$gt':'value'}
'field':{'$gt':value}
	Test for 'greater than' string, numeric or Boolean value i.e. field>value	'percentageDevicesDisplaced':{'$gt':0.5}

'field':{'$gte':'value'}
'field':{'$get':value}
	Test for 'greater than or equal to' string, numeric or Boolean value i.e. field>=value	'percentageDevicesDisplaced':{'$gte':0.5}

'field':{'$lt':'value'}
'field':{'$lt':value}
	Test for 'less than' string, numeric or Boolean value i.e. field<value	'percentageDevicesRemaining':{'$lt':0.5}

'field':{'$lte':'value'}
'field':{'$lte':value}
	Test for 'less than or equal to' string, numeric or Boolean value i.e. field<=value	'percentageDevicesRemaining':{'$lte':0.5}

'field':{'$ne':'value'}
'field':{'$ne':value}
	Test for 'not equal to' string, numeric or Boolean value i.e. field!=value	'zoneIdentifier':{'$ne':'SW1'}

'$and':[expression1, expression2, …]	Performs a Boolean 'and' on two or more expression values within the supplied array	'$and':['percentageDevicesDisplaced':{'$ge':0.5}, 'percentageDevicesRemaining':{'$le':0.5}]
'$or':[expression1,	Performs a Boolean	'$or':['zoneIdentifier':'SW1', 'zoneIdentifier':'NE1']

[8] This is a subset of the functions provided by MongoDB

expression2, …]	'or' on two or more expression values within the supplied array	
'$not': {expression}	Performs a Boolean 'not' on an expression (typically a compound Boolean expression)	'$not':{'$or':['zoneIdentifier':'SW1', 'zoneIdentifier':'NE1']}
'field':{'$in':[value1, value2, …]}	Matches any of the values (string, number or Boolean) in the supplied array	'zoneIdentifier':{'$in':['SW1', 'SW2', 'SW3']}
'field':{'$nin':[value1, value2, …]}	Matches provided the value (string, number of Boolean) is not in the supplied array	'zoneIdentifier':{'$nin':['NE1', 'NE2', 'NE3']}

Annex D Analytics Service Exemplars

D.1 Population insights

The following insights support use-cases which need to know some details about the user population, as derived principally from mobile network data.

D.1.1 Population snapshot

Provides a nominal daily or hourly snapshot report over a specified range of time indicating the estimated population make-up and the split between

- Users which are 'normally resident' in a designated area and are currently in the same area
- Users which are 'normally resident' in a designated area but are currently outside that area
- Users which are 'normally resident' outside the designated area but are now within it

The insight will be generated for one or more geographic zones in the country:

- Analytics servers may implement their own set of fixed zones types e.g. administrative levels or postcode/ zip-code areas. The results can be generated in advance of the application requesting this insight;
- Servers may also allow applications to specify custom zones for which they require the analysis:
  - Custom zones should be provided by the application in the form of GeoJSON or WKT (Well Known Text) defined polygons or multi-polygons;
  - Depending on the number of zones and requested timeframe this may mean the insight needs to be generated via a batch job due to the length of time required to generate the insight;
  - The server may implement constraints on the custom zones e.g. minimum area, exclusion of zones which intersect sensitive locations

Note: It is expected that operators will generally scale results to the population as a whole rather than reporting the actually identified numbers of users. Refer to section 3.15 of this document for recommendations.

Request parameters

Parameter	**Notes**
accept (HTTP header)	Indicates the preferred format for delivery of the insight. The typical output formats would be CSV (text/csv), JSON (application/json) or Excel (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) Servers should default to providing a CSV output if this parameter is omitted.
insight	Required only if the analytics are available from a shared analytics URL. In this case should be set to 'PopulationSnaphot'
mode	Allows the application to indicate its preference for delivery of the report. Should be specified as 'batch' or 'immediate'. If omitted the analytics server can deliver the output according to it's own preference.
callbackURL	Optionally specified to allow the application to receive notification that batch mode insight generation has been completed.

callbackData	Optionally specified to allow the application to specify context information related to the current request. This context information will be included in the notification sent to the callbackURL.
point	Optionally allows the application to specify the request is for the population snapshot for a single area centred at the specified point. See note 1 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
radius	Optionally used if point is specified to govern the size of the circular area to measure. The analytics server can impose limitations on the radius e.g. a minimum radius size of for example 1km.
box	Optionally allows the application to specify the request is for the population snapshot for a single area designated by a bounding box. See note 1 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
customAreas	Optionally allows the application to specify the request is for the population snapshot for multiple custom areas provided by the application. As it is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customAreas. See note 1 below.
polygon	Optionally allows the application to specify the request is for the population snapshot for a single area designated by a defined polygon. See note 1 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
selectedAreas	Optionally allows the application to specify the request is for the population snapshot for selected named areas which are pre-defined by the analytics server. Area names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 1 below.
baselineStartDateTime	Optional – specifies the start of the period used to assess which users are 'Normally Resident' in each area. If omitted the period 1 year prior to the reportingStartDateTime is assumed. A user is considered normally resident in an area if they have spent the majority of their time in that area between baselineStartDateTime and baselineEndDateTime.
baselineEndDateTime	Optional – specifies the end of the period used to assess which users are 'Normally Resident' in each area. If omitted the period 1 year prior to the reportingEndDateTime is assumed.
reportingStartDateTime	Required – specifies the start of the period for which population statistics will be generated.

reportingEndDateTime	Required – specifies the end of the period for which population statistics will be generated.
reportingInterval	Optional. Used to specify the desired reporting interval between reportingStartDateTime and reportingEndDateTime. If omitted defaults to the whole period between reportingStartDateTime and reportingEndDateTime. The application may request other intervals e.g. hourly, daily, weekly, or monthly intervals over the range of reportingStartDateTime and reportingEndDateTime.
maxRecords	Optionally allows the application to specify the maximum number of records to receive in the output report. The analytics server may ignore this parameter.
filter	Optionally allows the application to specify constraints on the results of the output report e.g. to discard any results where there are fewer than 100 users. The analytics server may not support filtering or may ignore this parameter.
order	Optionally allows the application to specify a desired column sorting order on the output report e.g. sorting the report by population size, or area name. The analytics server may not support sorting or may ignore this parameter. It is recommended that the default sorting order should be areaName as the primary sort criteria followed by the intervalStartDateTime.

Note 1: Only one of the parameters point, box, polygon, selectedAreas or customAreas is allowed in a request. The report server should generate an HTTP 400 (bad request) error if more than one option is specified. If the request does not include any of these parameters it should default to reporting across a default set of pre-defined areas.

Response data

The response shall be a CSV/ Excel data file or JSON encoded result data with the following fields:

Parameter	**Notes**
intervalStartDateTime	Defines the starting date/time of the reporting interval
intervalEndDateTime	Defines the ending date/time of the reporting interval
areaName	Indicates the relevant area name where the report is generated for 'selectedAreas' or 'customAreas'.
geometry	Optionally confirms the geometry of the area reported. If the report is generated as a CSV or Excel file this should be encoded using WKT (Well Known Text) and if the report is generated as JSON data the area will be defined using GeoJSON encoding. GeoJSON is only used for JSON formatted output due to the complexity of representing polygons. For output in Excel/ CSV, Well Known Text (WKT) will be output.
populationCount	Total number of distinct users identified in the given area over

	the reporting interval. Note that because certain users might be moving around the country during the reporting interval it is likely that they will be reported against multiple areas during that interval.
activeResidentCount	Total number of users who were resident in the specified area during the baseline period and were also active in any area during the reporting interval.
remainResidentCount	Indicates the number of distinct users who are considered to be normally resident in the specified area and spent over 50% of their time in the same area during the reporting interval.
outflowResidentCount	Indicates the number of distinct users who were considered normally resident in the specified area during the baseline period but spent over 50% of their time outside the specified area during the reporting interval. Numerically equivalent to the difference between activeResidentCount and remainResidentCount
inboundResidentCount	Total number of users who were resident in any other area than the specified area during the baseline period and were active in the specified area during the reporting interval.
inboundTemporaryResidentCount	Total number of users who were resident in any other area than the specified area during the baseline period and spent over 50% of their time inside the specified area during the reporting interval.
otherUserCount	Total number of users who were not active (in the specified area or elsewhere) during the baseline period but were active in the specified area at any point during the reporting interval. This includes those users who joined the mobile network after the baseline period as well as users who might have had their device switched off, or had roamed outside of their home network during the baseline period.
otherTemporaryResidentCount	Total number of users who were not active (in the specified area or elsewhere) during the baseline period but have spent at least 50% of their time active in the specified area during the reporting interval. This considers those users who joined the mobile network after the baseline period as well as users who might have had their device switched off, or had roamed outside of their home network during the baseline period.

D.1.2 Population origin / destination matrix

This insight reports the movement of the population and is useful to a wide range of use cases in transportation, disease response, or marketing/ advertising. The main outcome of this is a matrix of movement data identified for each source and destination area pair. It is expected:

- This report will only include records which exceed a minimum sample size in order to avoid the possibility of identifying individuals;

- Operators will apply private market share estimates to calculated results to avoid disclosing confidential information as part of the insight sharing.

The report includes a nominal daily or hourly record over a specified range of time indicating the estimated population make-up and the split between

- The number of users who are considered 'normally resident' in each designated source area;
- The number of 'normally resident' users who have travelled to the specified destination area;
- The average proportion of the reporting period that the 'normally resident' user spent in the specified destination area;
- The average proportion of the reporting period that the 'normally resident' user spent in the designated source area;
- The average proportion of the reporting period that the 'normally resident' user spent in an area other than the designated source and destination areas.

The insight will be generated for one or more geographic zones in the country:

- Analytics servers may implement their own set of fixed zones types e.g. administrative levels or postcode/ zip-code areas. The results can be generated in advance of the application requesting this insight;
- Servers may also allow applications to specify custom zones for which they require the analysis:
    - Custom zones should be provided by the application in the form of GeoJSON or WKT (Well Known Text) defined polygons or multi-polygons;
    - Depending on the number of zones and requested timeframe this may mean the insight needs to be generated via a batch job due to the length of time required to generate the insight;
    - The server may implement constraints on the custom zones e.g. minimum area, exclusion of zones which intersect sensitive locations

Note: It is expected that operators will generally scale results to the population as a whole rather than reporting the actually identified numbers of users. Refer to section 3.15 of this document for recommendations.

Request parameters

Parameter	**Notes**
accept (HTTP header)	Indicates the preferred format for delivery of the insight. The typical output formats would be CSV (text/csv), JSON (application/json) or Excel (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) Servers should default to providing a CSV output if this parameter is omitted.
insight	Required only if the analytics are available from a shared analytics URL. In this case should be set to 'PopulationOriginDestinationMatrix'
mode	Allows the application to indicate its preference for delivery of the report. Should be specified as 'batch' or 'immediate'. If omitted the analytics server can deliver the output according to it's own

	preference.
callbackURL	Optionally specified to allow the application to receive notification that batch mode insight generation has been completed.
callbackData	Optionally specified to allow the application to specify context information related to the current request. This context information will be included in the notification sent to the callbackURL.
customAreas	Optionally allows the application to specify the geographic areas via GeoJSON/ Well Known Text (WKT). At least two area definitions are required. As it is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customAreas. See note 2 below.
selectedAreas	Optionally allows the application to specify the request is for the population origin/ destination matrix for selected named areas which are pre-defined by the analytics server. Area names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 2 below.
baselineStartDateTime	Optional – specifies the start of the period used to assess which users are 'Normally Resident' in each area. If omitted the period 1 year prior to the reportingStartDateTime is assumed. A user is considered normally resident in an area if they have spent the majority of their time in that area between baselineStartDateTime and baselineEndDateTime.
baselineEndDateTime	Optional – specifies the end of the period used to assess which users are 'Normally Resident' in each area. If omitted the period 1 year prior to the reportingEndDateTime is assumed.
reportingStartDateTime	Required – specifies the start of the period for which population statistics will be generated.
reportingEndDateTime	Required – specifies the end of the period for which population statistics will be generated.
reportingInterval	Optional. Used to specify the desired reporting interval between reportingStartDateTime and reportingEndDateTime. If omitted defaults to the whole period between reportingStartDateTime and reportingEndDateTime. The application may request other intervals e.g. hourly, daily, weekly, or monthly intervals over the range of reportingStartDateTime and reportingEndDateTime.

maxRecords	Optionally allows the application to specify the maximum number of records to receive in the output report. The analytics server may ignore this parameter.
filter	Optionally allows the application to specify constraints on the results of the output report e.g. to discard any results where there are fewer than 100 users. The analytics server may not support filtering or may ignore this parameter.
order	Optionally allows the application to specify a desired column sorting order on the output report e.g. sorting the report by population size, or area name. The analytics server may not support sorting or may ignore this parameter. The default sorting order should be comparisonStartDateTime (primary) then sourceAreaName (secondary) then destinationAreaName (tiertary).

Note 2: Either customAreas or selectedAreas must be specified, but not both. The report server should generate an HTTP 400 (bad request) error if neither option is specified or both are specified.

Response data

The response shall be a CSV/ Excel data file or JSON encoded result data with the following fields

Parameter	**Notes**
intervalStartDateTime	Defines the starting date/time of the reporting interval
intervalEndDateTime	Defines the ending date/time of the reporting interval
sourceAreaName	Indicates the relevant source area name as used in 'selectedAreas' or 'customAreas'.
sourceGeometry	Optionally confirms the geometry of the source area reported. If the report is generated as a CSV or Excel file this should be encoded using WKT (Well Known Text) and if the report is generated as JSON data the area will be defined using GeoJSON encoding. GeoJSON is only used for JSON formatted output due to the complexity of representing polygons. For output in Excel/ CSV, Well Known Text (WKT) will be output.
destinationAreaName	Indicates the relevant destination area name as used in 'selectedAreas' or 'customAreas'.
destinationGeometry	Optionally confirms the geometry of the destination area reported. If the report is generated as a CSV or Excel file this should be encoded using WKT (Well Known Text) and if the report is generated as JSON data the area will be defined using

	GeoJSON encoding. GeoJSON is only used for JSON formatted output due to the complexity of representing polygons. For output in Excel/ CSV, Well Known Text (WKT) will be output.
normallyResidentUserCount	Estimated number of people who are normally resident in the specified source area.
visitedUserCount	Estimated number of people who are normally resident in the specified source area and travelled to the specified destination area during the reporting interval.
residentMeanDwellRatio	Compared with the reporting interval the mean ratio of time that Normally Resident users dwelled in the source area. This field is expected to be a real number in the range 0 to 1 representing 0% to 100% respectively.
visitingMeanDwellRatio	Compared with the reporting interval the mean ratio of time that Normally Resident users dwelled in the destination area. This field is expected to be a real number in the range 0 to 1 representing 0% to 100% respectively.
otherMeanDwellRatio	Compared with the reporting interval the mean ratio of time that Normally Resident users dwelled in any other area than either the source or destination areas. This field is expected to be a real number in the range 0 to 1 representing 0% to 100% respectively.

D.1.3 Risk assessment service

This insight reports a risk assessment based on the location of the user across the mobile network. There are several applications for this

- Risk assessment for personal / vehicle insurance;
- Risk assessment for bank payments / credit card purchases;
- Verifying that vulnerable individuals are in a safe location.

Note: That as this service involves the processing of personal (location) data of the subject user there will need to be implementation of relevant user consent processes according to local regulations. These consent processes may use for example Mobile Connect to gain the subject user's approval.

Request parameters

Parameter	**Notes**
accept (HTTP header)	Indicates the preferred format for delivery of the insight. It is expected this service would normally provide a JSON (application/json) response.
insight	Required only if the analytics are available from a shared analytics URL. In this case should be set to 'UserRiskAssessmentService'
mode	Allows the application to indicate its preference

	for delivery of the report. Should be specified as 'batch' or 'immediate'. If omitted the analytics server can deliver the output according to it's own preference.
subjectIdentity	Provides the identifier for the user being assessed. There are several possibilities for this • MSISDN of the user; • Hashed MSISDN (default); • Pseudonymous Customer Reference (in the event the user is on a mobile network supporting Mobile Connect[9]).
subjectType	Recommended in case there is ambiguity regarding the subjectIdentity field. Should contain one of the strings • MSISDN • HashedMSISDN • PCR
callbackURL	Optionally specified to allow the application to receive notification that batch mode insight generation has been completed.
callbackData	Optionally specified to allow the application to specify context information related to the current request. This context information will be included in the notification sent to the callbackURL.
customSafeAreas	Optionally allows the application to specify a list of 'known safe' geographic areas via GeoJSON/ Well Known Text (WKT). These safe areas will be considered alongside safe areas already known by the analytics server. As it is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customSafeAreas. The analytics server can apply constraints to any provided safe area to prevent the disclosure of sensitive user information.
customUnsafeAreas	Optionally allows the application to specify a list of 'known unsafe' geographic areas via GeoJSON/ Well Known Text (WKT). These unsafe areas will be considered alongside unsafe areas already known by the analytics server. As it is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customSafeAreas.

[9] See https://developer.mobileconnect.io/the-pcr

countryCheck	Allows the application to specify the ISO 3166-1 alpha-2 country code to check the user presence in that country. This particularly helps to avoid payment fraud.
reportingStartDateTime	Required – specifies the start of the period for which the user location will be assessed. Note that the analytics server can impose constraints on the reporting period or throttle requests to avoid applications using this service as a form of continuous user tracking.
reportingEndDateTime	Required – specifies the end of the period for which the user location will be assessed. Note that the analytics server can impose constraints on the reporting period or throttle requests to avoid applications using this service as a form of continuous user tracking.

Response data

The response shall provide JSON encoded result data with the following fields

Parameter	**Notes**
assessment	Defines the overall risk status of the user during the reporting period • Safe – the user was assessed to be in safe areas throughout the reporting period; • Unsafe – the user was assessed to be in one or more unsafe areas during the reporting period; • Unknown – the location of the user was not known during the reporting period
countryResult	Defines the result of the country check process • Present – the user was assessed to have remained in the specified country throughout the reporting period • Absent – the user was assessed to have been in one or more other countries during the reporting period • Unknown – the location of the user was not known during the reporting period

D.1.4 Location Broadcast Messaging Service

This service allows an external application to request the sending of a message to a set of users based on their presence in one or more defined geographical areas during one or more defined time periods. It is expected this can be used for various purposes including:

- Informing users about major traffic disruptions;
- Informing users about emerging health issues e.g. an outbreak of measles;

- Sending messages or promotions to users who are at sports or music events.

A key attribute of this service is that analytics are applied to select the users to send the message to, rather than the external application selecting users and then sending the message using bulk SMS which would have required the external application to have the mobile phone number for the user.

This service can only send the message to users of the mobile network receiving the request. Nationwide coverage would require the equivalent service to be available from all operators in that country.

Note: It is expected that any requests made with this service may be subject to moderation or throttling by the mobile network operator. Also, the mobile network operator will implement any relevant user consent including opt-in and opt-out as legally required in their country of operation.

Request parameters

Parameter	**Notes**
accept (HTTP header)	The response to this request is expected to be JSON, therefore it is expected the accept header is set to application/json
mode	Allows the application to indicate its preference for delivery of confirmation reports. Should be specified as 'batch' or 'immediate'. If omitted the analytics server can deliver the output according to it's own preference.
callbackURL	Optionally specified to allow the application to receive progress notifications at the following points The request has been accepted at the MNO system and is pending moderation Users are about to be sent the message All messages have been sent to the selected users
callbackData	Optionally specified to allow the application to specify context information related to the current request. This context information will be included in the notification sent to the callbackURL.
point	Optionally allows the application to request selection of users based on a geo point & radius. See note 3 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
radius	Optionally used if point is specified to govern the size of the circular area to measure when selecting users. The analytics server can impose limitations on the radius e.g. a minimum / maximum radius size of for example 1km.
box	Optionally allows the application to request selection of users based on a single area designated by a bounding box. See note 3 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
customAreas	Optionally allows the application to request selection of users based on multiple custom areas provided by the application. As it

	is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customAreas. See note 3 below.
polygon	Optionally allows the application to request selection of users based on a single area designated by a defined polygon. See note 3 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
selectedAreas	Optionally allows the application to to request selection of users based on their presence in selected named areas which are pre-defined by the analytics server. Area names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 3 below.

selectionIntervals	This required field contains an array of pairs of date/time values specifying the periods during which users are to be selected for receiving the message.
	This parameter should be sent in JSON format as a list of two element date/time values where each pair is also encoded as a JSON list. It is recommended HTTP POST is used to send this parameter with an HTTP Content-Type header of application/json.
	Date time values should be encoded as ISO 8601.
	`"selectionIntervals": [`
	`    ["2017-01-25T12:00:00Z", "2017-01-26T12:00:00Z"],`
	`    ["2017-01-29T00:00:00Z", "2017-01-29T23:59:59Z"]`
	`]`

scheduledStartDateTime	Optional – specifies the date/time that the requestor would like messages to be dispatched after. Should be specified as ISO 8601.
message	The text message to be sent to the selected users. This can either be • a simple text value in the case that all users will receive the same text message, • or (if the request type is application/json and HTTP POST is used to submit the request) a JSON dictionary of the form *<locale>:<localisedMessage>* allows the operator to implement policies to send the most appropriate message to users based on pre-existing knowledge of language preferences

Note 3: Only one of the parameters point, box, polygon, selectedAreas or customAreas is allowed in a request. The report server should generate an HTTP 400 (bad request) error if more than one option is specified. If the request does not include any of these parameters it should default to reporting across a default set of pre-defined areas.

Response data

The response to the request should be either

- HTTP Status Code 200 (OK) means the request was accepted at the operator server, the request body should include the information listed below (which is the same as would be sent in a notification callback) or
- Other HTTP status codes e.g. 4xx or 5xx indicate an error in the request

Parameter	Notes
status	Confirms the status for the request • 'Accepted' means the request is received at the MNO systems and ready for moderation • 'Approved' means the request has been approved by the moderator • 'Rejected' means the request has been rejected by the moderator • 'Starting' means the operator systems are now starting to send messages to users • 'Complete' means the operator systems have now finished sending messages to users
reason	In the case of rejection by a moderator this field will contain information from the moderator on why the request was rejected
selectedCount	Indicates the number of users that were identified for messaging based on the selected geography/ time window
sendCount	Indicates the number of users that were successfully sent a message (this is not an indication of the number of users that actually read the message)

Callback Notifications

As the broadcast request is processed callbacks may be made to the URL specified in the request. If no callbackURL was specified the operator systems will ignore sending callbacks.

The following data is recommended to be included in the callback notification:

Parameter	Notes
status	Confirms the updated status for the request • 'Accepted' means the request is received at the MNO systems and ready for moderation • 'Approved' means the request has been approved by the moderator • 'Rejected' means the request has been rejected by the moderator • 'Starting' means the operator systems are now starting to send messages to users • 'Complete' means the operator systems have now finished sending messages to users

reason	In the case of rejection by a moderator this field will contain information from the moderator on why the request was rejected
selectedCount	Indicates the number of users that were identified for messaging based on the selected geography/ time window
sendCount	Indicates the number of users that were successfully sent a message (this is not an indication of the number of users that actually read the message)
callbackData	Includes any callback data that was included in the original request.

D.2 Context data services

The following are data services that are useful as inputs to applications particularly when processing IoT data

D.2.1 Geo definitions

This service is used to support a common understanding between the analytics server and third party application for the geographical areas within a given country. It can be applied in any use case which aggregates results into distinct geographical areas.

This service may support alternate representations of geographic area definitions including

- GeoJSON
- Well Known Text (WKT)
- ESRI Shapefiles

It is expected that the results of this request are returned to the application immediately, either directly as usable mapping data or in the form of a redirect to the relevant content.

Request parameters

Parameter	**Notes**
accept (HTTP header)	Specifies the MIME type preferred for the output. For GeoJSON the type application/json should be specified. For Well Known Text the output formats would be CSV (text/csv) or Excel (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet). For ESRI shapefiles the output format should be application/octet-stream or application/zip Servers should default to providing a GeoJSON output if this parameter is omitted.
insight	Required only if this service is being published using a shared analytics URL. In this case should be set to 'GeoDefinitions'

level	Specifies the 'administrative level' of geographical areas that are requested in the output. The levels will vary by country according to land area, population and administration structures. [10] Examples include • "Admin1" – Administrative level 1 – the top level area definitions for the country. For example state level in the USA; • "Admin2" – Administrative level 2 – for example counties in the USA; • "Admin3" – Administrative level 3 – for example towns in the USA; • "Admin4" – Administrative level 4 – for example parishes in the UK; • "ElectoralDistrict" – an area returning one or more representative members in elections; • "PostalCode" – areas used for postal services, including zipcodes in the USA and postal codes elsewhere. Note that countries such as the UK which have fine granularity postal codes may offer postal code prefixes

Response data

The response data will vary according to the content type delivered.

GeoJSON response

The response will contain a list of named areas each of which has an associated GeoJSON polygon or multi-polygon

Parameter	**Notes**
name	Specifies the name of the defined area
geo:json	The GeoJSON definition for the area. See Annex B for examples of the coding of polygons & multi-polygons.

Well Known Text (WKT) response

Within the Comma Separated Values/ Excel output file the response will be provided as a table with the column names as below:

Parameter (column name)	**Notes**
name	Specifies the name of the defined area
geo	The Well Known Text (WKT) definition for the area as either a

[10] See for example https://en.wikipedia.org/wiki/List_of_administrative_divisions_by_country

	polygon or multi-polygon. [11]

ESRI Shapefile response

The analytics server shall return a 'zip' file which includes the relevant .shp, .shx, .dbf and .prj files according to the definition at https://doc.arcgis.com/en/arcgis-online/reference/shapefiles.htm .

HTTP redirect response

In the case that the analytics server is redirecting the application to the required content at another location, for example if provided by a content delivery network, the following is returned

- HTTP status code = 302 (moved temporarily)
- HTTP Location header = URL of the required content

D.2.2 Weather analysis

This service is used to provide an analysis over historical weather data acquired by the analytics server. It is designed to support applications in energy use, agriculture, water supply etc. The service will report analysis for one or more 'weather stations' either by name or in designated geographical areas.

The results of this service will be in a tabular format, suitable for consumption as JSON data, a Comma Separated Values file or an Excel spreadsheet.

Request parameters

Parameter	Notes
accept (HTTP header)	Indicates the preferred format for delivery of the insight. The typical output formats would be CSV (text/csv), JSON (application/json) or Excel (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) Servers should default to providing a CSV output if this parameter is omitted.
insight	Required only if this service is being published using a shared analytics URL. In this case should be set to 'WeatherAnalysis'
mode	Allows the application to indicate its preference for delivery of the report. Should be specified as 'batch' or 'immediate'. If omitted the analytics server can deliver the output according to it's own preference.
callbackURL	Optionally specified to allow the application to receive notification that batch mode insight generation has been completed.

[11] See for example https://en.wikipedia.org/wiki/Well-known_text

callbackData	Optionally specified to allow the application to specify context information related to the current request. This context information will be included in the notification sent to the callbackURL.
point	Optionally allows the application to specify the weather analysis is for observation stations within a single area centred at the specified point. See note 4 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
radius	Optionally used if point is specified to govern the size of the circular area which includes the weather stations of interest. The analytics server can impose limitations on the radius e.g. a maximum radius size of for example 25km.
box	Optionally allows the application to specify the weather analysis is for observation stations within a single area designated by a bounding box. See note 4 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
customAreas	Optionally allows the application to specify the weather analysis is for observation stations within multiple custom areas provided by the application. As it is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customAreas. See note 4 below.
polygon	Optionally allows the application to specify the weather analysis is for observation stations within a single area designated by a defined polygon. See note 4 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
selectedAreas	Optionally allows the application to specify the weather analysis is for observation stations within selected named areas which are pre-defined by the analytics server. Area names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 4 below.
selectedStations	Optionally allows the application to specify the weather analysis is for the provided named observation stations. Station names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 4 below.
reportingStartDateTime	Required – specifies the start of the period for which the weather analysis will be generated.
reportingEndDateTime	Required – specifies the end of the period for which the weather analysis will be generated.
reportingInterval	Optional. Used to specify the desired reporting interval between reportingStartDateTime and reportingEndDateTime.

	If omitted defaults to the whole period between reportingStartDateTime and reportingEndDateTime. The application may request other intervals e.g. hourly, daily, weekly, or monthly intervals over the range of reportingStartDateTime and reportingEndDateTime.
maxRecords	Optionally allows the application to specify the maximum number of records to receive in the output report. The analytics server may ignore this parameter.
filter	Optionally allows the application to specify constraints on the results of the output report e.g. to discard any results where the mean temperature was below 10 degrees Celsius. The analytics server may not support filtering or may ignore this parameter.
order	Optionally allows the application to specify a desired column sorting order on the output report e.g. sorting the report by date, or station name. The analytics server may not support sorting or may ignore this parameter. It is recommended that the default sorting order should be stationName as the primary sort criteria followed by the intervalStartDateTime.

Note 4: Only one of the parameters point, box, polygon, selectedAreas, selectedStations or customAreas is allowed in a request. The report server should generate an HTTP 400 (bad request) error if more than one option is specified. If the request does not include any of these parameters it should default to reporting across a default set of pre-defined areas.

Response data

The response shall comprise a table of values containing the following

Parameter	**Notes**
name	Specifies the observation station name
intervalStartDateTime	Defines the starting date/time of the reporting interval
intervalEndDateTime	Defines the ending date/time of the reporting interval
location	Specifies the latitude/longitude of the weather observation station represented as GeoJSON if the output report is JSON or Well Known Text if the output report is Excel or Comma Separated Values
totalRainfall	The total amount of rain/ precipitation during the reporting interval standardized to millimetres of precipitation per square metre of ground
totalSunshine	The number of minutes of sunshine measured during the reporting interval
meanTemperature	The mean temperature during the reporting interval, reported in degrees Celsius
minTemperature	The minimum temperature during the reporting interval, reported in degrees Celsius
maxTemperature	The maximum temperature during the reporting interval,

	reported in degrees Celsius
meanWindSpeed	The mean wind speed during the reporting interval, reported in metres per second
maxWindSpeed	The maximum wind speed during the reporting interval, reported in metres per second
netWindSpeed	The prevailing wind speed over the reporting interval. See note below.
netWindDirection	The prevailing wind direction over the reporting interval reported in compass degrees where 0 = North, 90 = East, 180 = South and 270 = West. See note below.

Note: It is expected that netWindSpeed and netWindDirection are calculated by resolving all constituent readings at the observation station.[12]

D.2.3 Air quality analysis

This service is used to provide an analysis over historical air quality data acquired by the analytics server. It is designed to support applications in transportation management, energy production management, urban and industrial planning and monitoring, and planning for health services. The service will report analysis for one or more 'air quality monitoring stations' either by name or in designated geographical areas.

The results of this service will be in a tabular format, suitable for consumption as JSON data, a Comma Separated Values file or an Excel spreadsheet.

Request parameters

Parameter	**Notes**
accept (HTTP header)	Indicates the preferred format for delivery of the insight. The typical output formats would be CSV (text/csv), JSON (application/json) or Excel (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) Servers should default to providing a CSV output if this parameter is omitted.
insight	Required only if this service is being published using a shared analytics URL. In this case should be set to 'AirQualityAnalysis'
mode	Allows the application to indicate its preference for delivery of the report. Should be specified as 'batch' or 'immediate'. If omitted the analytics server can deliver the output according to it's own preference.
callbackURL	Optionally specified to allow the application to receive notification that batch mode insight generation has been completed.
callbackData	Optionally specified to allow the application to specify context information related to the current request. This context

[12] See http://www.ndbc.noaa.gov/wndav.shtml for method.

	information will be included in the notification sent to the callbackURL.
point	Optionally allows the application to specify the air quality analysis is for air quality monitoring stations within a single area centred at the specified point. See note 5 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
radius	Optionally used if point is specified to govern the size of the circular area which includes the air quality monitoring stations of interest. The analytics server can impose limitations on the radius e.g. a maximum radius size of for example 25km.
box	Optionally allows the application to specify the air quality analysis is for air quality monitoring stations within a single area designated by a bounding box. See note 5below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
customAreas	Optionally allows the application to specify the air quality analysis is for air quality monitoring stations within multiple custom areas provided by the application. As it is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customAreas. See note 5 below.
polygon	Optionally allows the application to specify the air quality analysis is for air quality monitoring stations within a single area designated by a defined polygon. See note5 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
selectedAreas	Optionally allows the application to specify the air quality analysis is for air quality monitoring stations within the selected named areas which are pre-defined by the analytics server. Area names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 5 below.
selectedStations	Optionally allows the application to specify the air quality analysis is for named air quality monitoring stations. Station names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 5 below.
reportingStartDateTime	Required – specifies the start of the period for which the air quality analysis will be generated.
reportingEndDateTime	Required – specifies the end of the period for which the air quality analysis will be generated.
reportingInterval	Optional. Used to specify the desired reporting interval between reportingStartDateTime and reportingEndDateTime. If omitted defaults to the whole period between reportingStartDateTime and reportingEndDateTime. The

	application may request other intervals e.g. hourly, daily, weekly, or monthly intervals over the range of reportingStartDateTime and reportingEndDateTime.
standard	Optional. Which reporting standard should be used for reporting of air quality measures. It is assumed the default is for World Health Organisation measurement standards to be reported (standard="WHO") and the output format is defined according to this. See below for the World Health Organisation standards
maxRecords	Optionally allows the application to specify the maximum number of records to receive in the output report. The analytics server may ignore this parameter.
filter	Optionally allows the application to specify constraints on the results of the output report e.g. to discard any results where the mean NO2 level was below 20 microgrammes per cubic metre. The analytics server may not support filtering or may ignore this parameter.
order	Optionally allows the application to specify a desired column sorting order on the output report e.g. sorting the report by date, or station name. The analytics server may not support sorting or may ignore this parameter. It is recommended that the default sorting order should be stationName as the primary sort criteria followed by the intervalStartDateTime.

Note 5: Only one of the parameters point, box, polygon, selectedAreas, selectedStations or customAreas is allowed in a request. The report server should generate an HTTP 400 (bad request) error if more than one option is specified. If the request does not include any of these parameters it should default to reporting across a default set of pre-defined areas.

World Health Organisation standards[13]

Pollutant	**Guideline values**
Fine Particulate Matter ($PM_{2.5}$)	10 µg/m^3 annual mean 25 µg/m^3 24-hour mean
Coarse Particulate Matter (PM_{10})	20 µg/m^3 annual mean 50 µg/m^3 24-hour mean
Ozone (O_3)	100 µg/m^3 8-hour mean
Nitrogen dioxide (NO_2)	40 µg/m^3 annual mean 200 µg/m^3 1-hour mean
Sulphur dioxide (SO_2)	20 µg/m^3 24-hour mean 500 µg/m^3 10-minute mean

[13] http://www.who.int/mediacentre/factsheets/fs313/en/

Response data

The response shall comprise a table of values containing the following

Parameter	**Notes**
name	Specifies the observation station name
intervalStartDateTime	Defines the starting date/time of the reporting interval
intervalEndDateTime	Defines the ending date/time of the reporting interval
location	Specifies the latitude/longitude of the air quality monitoring station represented as GeoJSON if the output report is JSON or Well Known Text if the output report is Excel or Comma Separated Values
averageNO2	The average level of nitrogen dioxide (NO_2) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
averageO3	The average level of ozone (O_3) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
averagePM25	The average level of fine particulate matter ($PM_{2.5}$) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
averagePM10	The average level of coarse particular matter (PM_{10}) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
averageSO2	The average level of sulphur dioxide (SO_2) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
maximumNO2	The maximum level of nitrogen dioxide (NO_2) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
maximumO3	The maximum level of ozone (O_3) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
maximumPM25	The maximum level of fine particulate matter ($PM_{2.5}$) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
maximumPM10	The maximum level of coarse particular matter (PM_{10}) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
maximumSO2	The maximum level of sulphur dioxide (SO_2) during the reporting interval. Measured in microgrammes per cubic metre ($\mu g/m^3$).
tExceedNO2AGV	The time (in minutes) during which levels of nitrogen dioxide (NO_2) exceeded the annual guideline value during the reporting interval.
tExceedNO2HGV	The time (in minutes) during which levels of nitrogen dioxide (NO_2) exceeded the hourly guideline value during the

	reporting interval.
tExceedO3H8GV	The time (in minutes) during which levels of ozone (O_3) exceeded the 8-hourly guideline value during the reporting interval.
tExceedPM25AGV	The time (in minutes) during which levels of fine particulate matter ($PM_{2.5}$) exceeded the annual guideline value during the reporting interval.
tExceedPM25DGV	The time (in minutes) during which levels of fine particulate matter ($PM_{2.5}$) exceeded the daily (24 hour) guideline value during the reporting interval.
tExceedPM10AGV	The time (in minutes) during which levels of coarse particular matter (PM_{10}) exceeded the annual guideline value during the reporting interval.
tExceedPM10DGV	The time (in minutes) during which levels of coarse particular matter (PM_{10}) exceeded the daily (24 hour) guideline value during the reporting interval.
tExceedSO2DGV	The time (in minutes) during which levels of sulphur dioxide (SO_2) exceeded the daily (24 hour) guideline value during the reporting interval.
tExceedSO2M10GV	The time (in minutes) during which levels of sulphur dioxide (SO_2) exceeded the 10 minute guideline value during the reporting interval.

D.2.4 Air quality prediction

This service is used to provide a prediction of air quality based on predictive analytics in the analytics server. It is designed to provide city administrators, health professionals and members of the public with an advanced prediction of poor air quality based on historical data and trends. The service will provide predictions for one or more 'air quality monitoring stations' either by name or in designated geographical areas.

The results of this service will be in a tabular format, suitable for consumption as JSON data, a Comma Separated Values file or an Excel spreadsheet.

Request parameters

Parameter	**Notes**
accept (HTTP header)	Indicates the preferred format for delivery of the insight. The typical output formats would be CSV (text/csv), JSON (application/json) or Excel (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) Servers should default to providing a CSV output if this parameter is omitted.
insight	Required only if this service is being published using a shared analytics URL. In this case should be set to 'AirQualityPrediction'
mode	Allows the application to indicate its preference for delivery of the report. Should be specified as 'batch' or 'immediate'. If omitted the analytics server can deliver the output according to it's own preference.

callbackURL	Optionally specified to allow the application to receive notification that batch mode insight generation has been completed.
callbackData	Optionally specified to allow the application to specify context information related to the current request. This context information will be included in the notification sent to the callbackURL.
point	Optionally allows the application to specify the air quality predictions are for air quality monitoring stations within a single area centred at the specified point. See note 6 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
radius	Optionally used if point is specified to govern the size of the circular area which includes the air quality monitoring stations of interest. The analytics server can impose limitations on the radius e.g. a maximum radius size of for example 25km.
box	Optionally allows the application to specify the air quality predictions are for air quality monitoring stations within a single area designated by a bounding box. See note 6 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
customAreas	Optionally allows the application to specify the air quality predictions are for air quality monitoring stations within multiple custom areas provided by the application. As it is likely the HTTP request body will be large the HTTP POST method should be used for requests involving customAreas. See note 6 below.
polygon	Optionally allows the application to specify the air quality predictions are for air quality monitoring stations within a single area designated by a defined polygon. See note 6 below. Nominally this parameter should be specified using the GeoJSON representation though WKT (Well Known Text) can be used instead.
selectedAreas	Optionally allows the application to specify the air quality predictions are for air quality monitoring stations within the selected named areas which are pre-defined by the analytics server. Area names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 6 below.
selectedStations	Optionally allows the application to specify the air quality predictions are for named air quality monitoring stations. Station names should be comma separated. If many area names are specified it may be necessary to use the HTTP POST method for the request. See note 6 below.
reportingStartDateTime	Required – specifies the start of the period for which the air quality predictions will be generated. Note that prediction reliability is considered likely to diminish as the prediction models look further into the future so the analytics

	server may override the provided reportingStartDateTime with date time ranges that there is a reasonable confidence for predicting.
reportingEndDateTime	Required – specifies the end of the period for which the air quality predictions will be generated. Note that prediction reliability is considered likely to diminish as the prediction models look further into the future so the analytics server may override the provided reportingEndDateTime with date time ranges that there is a reasonable confidence for predicting.
reportingInterval	Optional. Used to specify the desired precion intervals between reportingStartDateTime and reportingEndDateTime. If omitted defaults to the whole period between reportingStartDateTime and reportingEndDateTime. The application may request other intervals e.g. hourly or daily. Periods longer than daily are very likely to be of poor reliability so should not be supported.
maxRecords	Optionally allows the application to specify the maximum number of records to receive in the output report. The analytics server may ignore this parameter.
filter	Optionally allows the application to specify constraints on the results of the output report e.g. to discard any results where the predicted NO2 level is below 50 microgrammes per cubic metre. The analytics server may not support filtering or may ignore this parameter.
order	Optionally allows the application to specify a desired column sorting order on the output report e.g. sorting the report by date, or station name. The analytics server may not support sorting or may ignore this parameter. It is recommended that the default sorting order should be stationName as the primary sort criteria followed by the intervalStartDateTime.

Note 6: Only one of the parameters point, box, polygon, selectedAreas, selectedStations or customAreas is allowed in a request. The report server should generate an HTTP 400 (bad request) error if more than one option is specified. If the request does not include any of these parameters it should default to reporting across a default set of pre-defined areas.

Response data

The response shall comprise a table of values containing the following

Parameter	Notes
name	Specifies the observation station name
intervalStartDateTime	Defines the starting date/time of the reporting interval
intervalEndDateTime	Defines the ending date/time of the reporting interval
location	Specifies the latitude/longitude of the air quality monitoring

	station represented as GeoJSON if the output report is JSON or Well Known Text if the output report is Excel or Comma Separated Values
averageNO2	The predicted average level of nitrogen dioxide (NO_2) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
averageO3	The predicted average level of ozone (O_3) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
averagePM25	The predicted average level of fine particulate matter ($PM_{2.5}$) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
averagePM10	The predicted average level of coarse particular matter (PM_{10}) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
averageSO2	The predicted average level of sulphur dioxide (SO_2) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
maximumNO2	The predicted maximum level of nitrogen dioxide (NO_2) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
maximumO3	The predicted maximum level of ozone (O_3) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
maximumPM25	The predicted maximum level of fine particulate matter ($PM_{2.5}$) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
maximumPM10	The predicted maximum level of coarse particular matter (PM_{10}) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).
maximumSO2	The predicted maximum level of sulphur dioxide (SO_2) during the reporting interval. Measured in microgrammes per cubic metre (µg/m^3).

Annex E Document Management

E.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
0.1	22 Nov 2017	New PRD - draft	TG	Allan Bartlett / GSMA
1.0	21 Dec 2017	Approved first version	TG	Allan Bartlett / GSMA

E.2 Other Information

Type	Description
Document Owner	Internet of Things – IoT Big Data Project
Editor / Company	GSMA

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com

Your comments or suggestions & questions are always welcome.