

BSMART Technology Limited

Integration Design Document for SUMATRA VTS - Web API for GPS Vendor Data Integration

Revision 1.14

Table of Contents

1 INTRODUCTION3

1.1 PURPOSE 3

2 INTEGRATION OVERVIEW 3

2.1 INTEGRATION METHOD 3

2.2 REAL-TIME DATA VIA WEB SERVICE 4

3 DATA STRUCTURE 5

3.1.1 Vehicle Positions 5

3.1.2 Example requests 10

4 REVISION HISTORY 14

1 Introduction

1.1 Purpose

This document provides design of the integration of vehicle GPS positioning information between External Agency and SUMATRA.

2 Integration Overview

2.1 Integration Method

A Web service is a method of communication between two electronic devices over the Web (Internet).

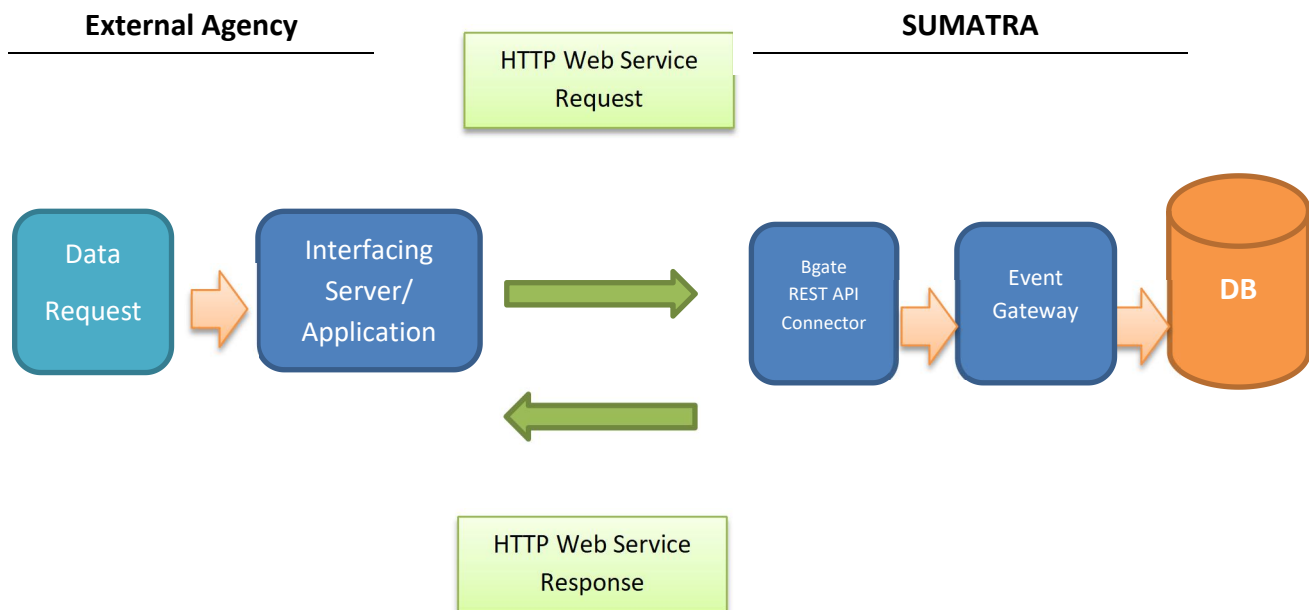
The W3C defines a "Web service" as "a software system designed to support interoperable machine-to-machine interaction over a network". It has an interface described in a machine-processable format (specifically Web Services Description Language, known by the acronym WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."

Web Services offer many benefits over other types of distributed computing architectures.

- **Interoperability** - This is the most important benefit of Web Services. Web Services typically work outside of private networks, offering developers a non-proprietary route to their solutions. Services developed are likely, therefore, to have a longer life-span, offering better return on investment of the developed service. Web Services also let developers use their preferred programming languages. In addition, thanks to the use of standards-based communications methods, Web Services are virtually platform-independent.
- **Usability** - Web Services allow the business logic of many different systems to be exposed over the Web. This gives your applications the freedom to choose the Web Services that they need. Instead of re-inventing the wheel for each client, you need only include additional application-specific business logic on the client-side. This allows you to develop services and/or client-side code using the languages and tools that you want.
- **Reusability** - Web Services provide not a component-based model of application development, but the closest thing possible to zero-coding deployment of such services. This makes it easy to reuse Web Service components as appropriate in other services. It also makes it easy to deploy legacy code as a Web Service.
- **Deplorability** - Web Services are deployed over standard Internet technologies. This makes it possible to deploy Web Services even over the fire wall to servers running on the Internet on the other side of the globe. Also thanks to the use of proven community standards, underlying security (such as SSL) is already built-in.

2.2 Real-time data via Web Service

This web service integration approach for the integrating the real-time data from External Agency to SUMATRA are as below. The data structure for communication shall be in JSON data format.



The SUMATRA interfacing application shall query the last known GPS data from database and respond the result in JSON format.

3 Data Structure

3.1.1 Vehicle Positions

The data dictionary for vehicle positions data integration between External Agency and BSMART are as below. (200 max).

The location data point is included in the request as JSON array, and in 1 request there shall be 3*N data points due to SUMATRA VTS BGate requirement.

Request from Bus Operator to SUMATRA VTS

Data Type	Vehicle Position
Format	JSON
HTTP Method	POST
Authorization	Basic xxxxxxxxxx (exact value provided separately)
URL	http://IPADDRESS:8090/data-integration/integration/gps
Username	tobeGiven
Password	tobeGiven

Field Name	Description	Required	
vehicle_reg_no	Vehicle Registration number	Yes	
type	The type of this data points, currently 'poi' is supported	Yes	poi
IMEI	International Mobile Equipment Identity	Yes	
Items Array			
latitude	Latitude (in decimal degrees)	Yes	decimal degrees
longitude	longitude (in decimal degrees)	Yes	decimal degrees
altitude	Altitude	Yes	Integer
timestamp	GPS Time stamp (YYYY-MM-DD HH:MM:SS)	Yes	**refer to note
speed	Speed (km/h)	Yes	decimal degrees
bearing	Direction (degree)	Yes	decimal degrees
odometer	Odometer (KM)	Optional	*please use 0 if no value
satellite_count	Receivable Number of satellites	Yes	Integer
HDOP	Horizontal Dilution Of Precision (Quality of GPS signal)	Yes	decimal degrees
d2d3	Satellite mode 2D or 3D	Yes	2 or 3
RSSI	Received Signal Strength Indication	Yes	integer
LAC	Local Area Code	Yes	integer
Cell_ID	Cell ID	Yes	integer
MCC	Mobile Country Code	Yes	integer

MGS_ID	Unique data running number (64bits)	Yes	integer
activity_id	<p>Activity ID:</p> <ol style="list-style-type: none"> 1. Movement/Logging (Default) 2. Engine ON 3. Engine OFF 4. Speeding 5. Hash Braking 6. Hash Turning 7. Hash Acceleration 8. Panic Button (Driver) 9. Internal Battery Low 10. External Power Disconnected 11. Excessive Idle 12. Accident 13. Panic Button (Passenger) 14. Device Tempering (For example: GPS Antenna Disconnected) 15. Black Box Data Logging 16. Fuel data report 17. Invalid Scan 18. Engine Start 19. Engine Stop 20. Enter Boundary 21. Leave Boundary 22. Enter Checkpoint 23. Leave Checkpoint 24. Ibutton Scan (Regular) 	Yes	integer
addon_info	<p>Add-on information for related activity in JSON format depends on the activity ID</p> <p>Applicable data for Trip End activity (activity 3):</p> <ul style="list-style-type: none"> • distance travelled in km format. • trip duration in minutes. • Average speed (KM/h) • Max speed (KM/h) <p>Example:</p> <pre>{ "distance_travelled": "505.2", "trip_duration": "400", "avgSpeed": "60", "maxSpeed": "85" }</pre> <p>Applicable data for Trip Start activity (activity 2):</p> <ul style="list-style-type: none"> • idleTime (in seconds) • v_driver_identification_no (16 digit hex string represent driver number) <p>Example:</p>	Optional (put " " if not match any activity)	addon_info

	<pre>{ "idleTime": "1231", "v_driver_identification_no": "D70000127F886501" }</pre> <p>Applicable data for power status activity (activity 9/10):</p> <ul style="list-style-type: none"> External Power Voltage Device Internal Battery Voltage <p>Example:</p> <pre>{ "ext_power_voltage": "24.22", "int_battery_voltage": "3.89" }</pre> <p>Applicable data for Invalid Scan (activity 17) and Regular Ibutton Scan (activity 24):</p> <ul style="list-style-type: none"> v_driver_identification_no (16 digit hex string represent driver number, FFFFFFFFFFFFFFFF when no valid identification number detected) <p>Example:</p> <pre>{ "v_driver_identification_no": "D70000127F886501" }</pre>		
fuel_info	<p>Fuel data information object when activity id is 16 (activity = 16)</p> <p>Applicable data for this objects include:</p> <ul style="list-style-type: none"> Data valid flag (0 for valid, other value to indicate non-valid fuel data) Received Signal Sensitivity of the fuel sensor (0-99) Software status code (0 for normal) Hardware fault code (0 for normal, see table 2 for details) Fuel level (smoothed) in mm (Integer) Real time fuel level in mm (Integer) Tank temperature in Celsius * (integer value with original temperaturex10, example: For 23.5 Celsius, 235 should be input in API) Fuel tank compartment (integer value with default should be 1) <p>Example:</p> <pre>{ "validFlag": "0", "signalLevel": "75", "softStatus": "0",</pre>		fuel_info

	<pre>"hardFault": "0", "fuelLevel": "2100", "rtFuelLevel": "2134", "tankTemp": "312", "channel": "1" }</pre>		
--	--	--	--

Table 1 Data structure for API

Note 1: About Hardware fault code

Item	Description
0	No Fault
1	Probe Cable Failed (connection fail)
2	Probe Cable Intermediate (connection lose)
3	Reserved
4	Sensor Coupling Layer abnormal
5	Detected fuel level zero
6	Reserved
7	Reserved
8	Reserved
9	Reserved

Table 2 Hardware fault code

Note 2: About timestamp

In order to avoid the confusion of different ISO timezone format, we use the epoch millisecond (the number of milliseconds since January 1, 1970, 00:00:00 GMT) to represent the time for GPS.

In general, many program languages have built-in method to convert the Date format to epoch timestamp. For instance in java the `getTime()` method can be used to convert the date to timestamp, example as below:

```
String myDate = "2014/10/29 18:10:45";

SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

Date date = sdf.parse(myDate);

long millis = date.getTime();
```

3.1.2 Example requests

Request JSON Sample for movement (activity 1):

```
{
  "vehicle_reg_no": "DUMMY123",
  "type": "poi",
  "imei": "100000000004000",
  "items": [
    {
      "latitude": "-6.79124",
      "longitude": "39.1",
      "altitude": "21",
      "timestamp": "1541603095967",
      "horizontal_speed": "80",
      "vertical_speed": "0",
      "bearing": "150",
      "satellite_count": "9",
      "HDOP": "1",
      "d2d3": "3",
      "RSSI": "0",
      "LAC": "123",
      "Cell_ID": "12345",
      "MGS_ID": "12345",
      "MCC": "635",
      "activity_id": "1"
    }
  ]
}
```

Request JSON Sample for Engine On (activity 2):

```
{
  "vehicle_reg_no": "DUMMY123",
  "type": "poi",
  "imei": "100000000004000",
  "items": [
    {
      "latitude": "-6.79124",
      "longitude": "39.1",
      "altitude": "21",
      "timestamp": "1541603095967",
      "horizontal_speed": "80",
      "vertical_speed": "0",
      "bearing": "150",
      "satellite_count": "9",
      "HDOP": "1",
      "d2d3": "3",
      "RSSI": "0",
      "LAC": "123",
      "Cell_ID": "12345",
      "MGS_ID": "12345",
      "MCC": "635",
      "activity_id": "2",
      "addon_info": {
        "idleTime": "1231",
        "v_driver_identification_no": "D70000127F886501"
      }
    }
  ]
}
```

Request JSON Sample for Engine Off (activity 3):

```
{
  "vehicle_reg_no": "DUMMY123",
  "type": "poi",
  "imei": "100000000004000",
  "items": [
    {
      "latitude": "-6.79124",
      "longitude": "39.1",
      "altitude": "21",
      "timestamp": "1541603095967",
      "horizontal_speed": "80",
      "vertical_speed": "0",
      "bearing": "150",
      "satellite_count": "9",
      "HDOP": "1",
      "d2d3": "3",
      "RSSI": "0",
      "LAC": "123",
      "Cell_ID": "12345",
      "MGS_ID": "12345",
      "MCC": "635",
      "activity_id": "3",
      "addon_info": {
        "distance_travelled": "505.2",
        "trip_duration": "400",
        "avgSpeed": "60",
        "maxSpeed": "85"
      }
    }
  ]
}
```

Request JSON Sample for activity 4/5/6/7/8/11/12/13/14/15:

```
{
  "vehicle_reg_no": "DUMMY123",
  "type": "poi",
  "imei": "100000000004000",
  "items": [
    {
      "latitude": "-6.79124",
      "longitude": "39.1",
      "altitude": "21",
      "timestamp": "1541603095967",
      "horizontal_speed": "80",
      "vertical_speed": "0",
      "bearing": "150",
      "satellite_count": "9",
      "HDOP": "1",
      "d2d3": "3",
      "RSSI": "0",
      "LAC": "123",
      "Cell_ID": "12345",
      "MGS_ID": "12345",
      "MCC": "635",
      "activity_id": "4" //Remove this comment in actual request and use
      corresponding activity id to represent the event
    }
  ]
}
```

Request JSON Sample for Power event (activity 9 and 10):

```
{
  "vehicle_reg_no": "DUMMY123",
  "type": "poi",
  "imei": "100000000004000",
  "items": [
    {
      "latitude": "-6.79124",
      "longitude": "39.1",
      "altitude": "21",
      "timestamp": "1541603095967",
      "horizontal_speed": "80",
      "vertical_speed": "0",
      "bearing": "150",
      "satellite_count": "9",
      "HDOP": "1",
      "d2d3": "3",
      "RSSI": "0",
      "LAC": "123",
      "Cell_ID": "12345",
      "MGS_ID": "12345",
      "MCC": "635",
      "activity_id": "9", //Remove this comment in actual request and use corresponding
activity id to represent the event
      "addon_info": {
        "ext_power_voltage": "24.22",
        "int_battery_voltage": "3.89"
      }
    }
  ]
}
```

Request JSON Sample for Fuel sensor data (activity 16):

```
{
  "vehicle_reg_no": "DUMMY123",
  "type": "poi",
  "imei": "100000000004000",
  "items": [
    {
      "latitude": "-6.79124",
      "longitude": "39.1",
      "altitude": "21",
      "timestamp": "1541387009000",
      "horizontal_speed": "80",
      "vertical_speed": "0",
      "bearing": "150",
      "satellite_count": "9",
      "HDOP": "1",
      "d2d3": "3",
      "RSSI": "0",
      "LAC": "123",
      "Cell_ID": "12345",
      "MGS_ID": "12345",
      "MCC": "635",
      "activity_id": "16",
      "fuel_info": {
        "validFlag": "0",
        "signalLevel": "75",
        "softStatus": "0",
        "hardFault": "0",
        "fuelLevel": "2100",
        "rtFuelLevel": "2134",
        "tankTemp": "311",
        "channel": "1"
      }
    }
  ]
}
```

```
{
  "status": "1",
  "timestamp": "1515131191442"
}
```

Response from SUMATRA VTS to 3rd Party Operator

Field Name	Description	Required
status	1=successful, 0=failed	Yes
timestamp	Service response time GPS Time stamp	Yes
Message	Message if error occurred.	Optional

Table 3 API response

Response JSON Sample:

```
{
  "status": "1",
  "timestamp": "1515131191442"
}
```

4 Revision History

Version	Date	Author	Remarks
1.0	2016-10-01	Halvin Hee	First release
1.1	2016-10-14	Halvin Hee	Amended activity name Added new activities
1.2	2017-04-07	Halvin Hee	Updated vehicle position data fields.
1.3	2017-04-11	Halvin Hee	Updated optional (additional info) data fields.
1.4	2017-05-04	Stephen TEANG	Updated removal of activity ID (trip info and geo-fencing info).
1.5	2017-05-24	Halvin Hee	Updated Battery Level information.
1.6	2017-05-31	Stephen TEANG	Updated Event Alert Setting
1.7	2018-03-23	Joseph Mukiibi	Updated to New API Info
1.8	2018-03-24	Joseph Mukiibi	Updated optional (additional info) data fields.
1.9			Format update
1.10	2018-11-03	Wenzheng Zhu	Update Fuel data information
1.11	2018-11-07	Wenzheng Zhu	Add more samples
1.12	2019-01-28	Stephen TEANG	Company Address Update
1.13	2019-04-25	Joseph Mukiibi	Update sample JSON (add IMEI)
1.14	2024-04-19	Joseph Mukiibi	Add regular Ibutton scan support

© BSMART 2024

All rights reserved. Reproduction in whole or in part is prohibited without the prior written consent from BSMART. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without any notice. BSMART will accept no liability for any consequence of its use. Publication thereof does not convey or imply any license under patent – or any other industrial or technical property rights. These products are not designed for use in life support applications, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. BSMART customers using these products for use in such applications will do so at their own risk and agree to fully indemnify BSMART for any damages resulting from such improper use.