# Technology Scan and Feasibility Analysis

### Purpose:

The purpose of this document is to record the result of research into technologies that align with the requirements of the Commodity Blue Force Tracking project.

### Background:

The Commodity Blue Force Tracking project seeks to emulate the functionality of Blue Force Tracking systems for the purpose of science and technology experimental trials. The project requires two key technology-based deliverables:

* Android devices loaded with software capable of receiving and forwarding GPS location data
* A server capable of receiving and parsing location data, visualising data on a web-based display, streaming data to a UDP socket and exposing a REST endpoint for the request of logged data

These deliverables may be developed from the ground up, or may be based on free and open-source software, which will assist in reducing the development load. In order to assess the suitability of using existing software in the development approach, technologies were researched and their functionality was evaluated to determine if they supported the requirements of the project deliverables. Two Android applications and three server applications were investigated in-depth and the findings have been documented within this paper.

### Preliminary Technology Scan:

Prior to delving in-depth with the feasibility analysis, a preliminary investigation into existing technologies was conducted. Although the initial exploration found a multitude of Android applications capable of receiving and forwarding GPS data, and server software with the ability to receive, parse and visualise location data, it was decided that many of these applications did not meet the minimum requirements to bare further scrutiny. Many of the applications found, for example, contained the following shortcomings:

* Free versions of software that limited the number of user accounts or tracked devices
* Server software that displayed the last known location of devices, but not previous locations
* Web-based software that was served from an external host only and could not be further developed
* User interfaces which offered no configuration options and simply displayed track data

Had no other options been made available, many of these applications would have been suitable for further development, but would have drastically increased the workload. However, as there exists software that better aligns with the requirements of the project, it was decided the software with major shortcomings would be excluded from analysis.

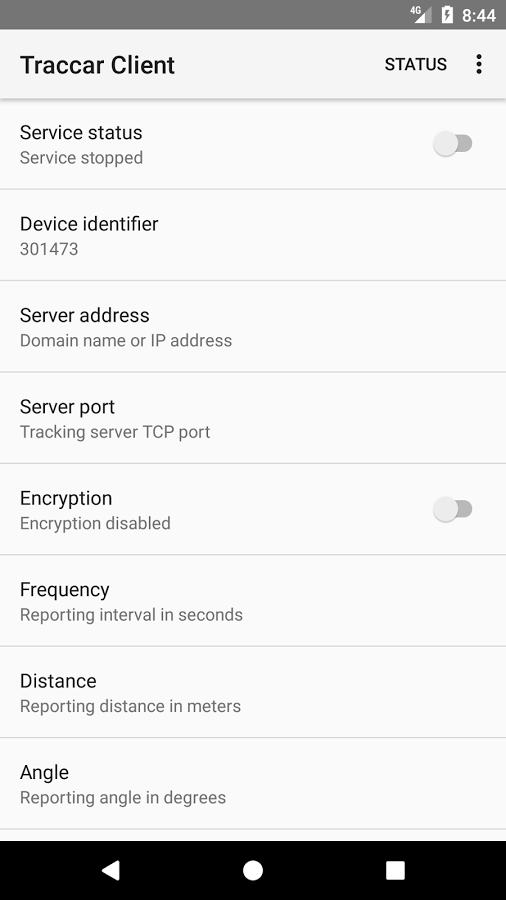
### Android Application:

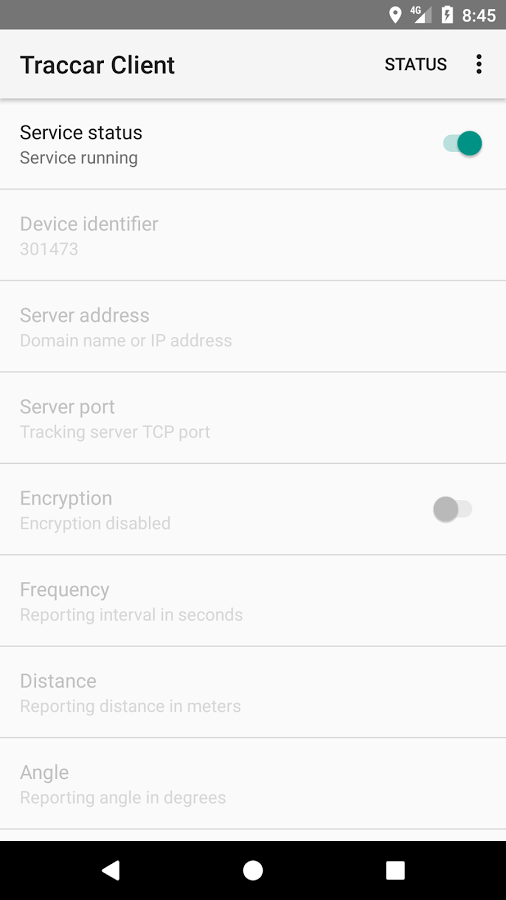
The two Android applications analysed were Traccar Client and CelltracGTS/Free. These applications were chosen as they provide out-of-the-box operation with the three servers reviewed. Many GPS tracking Android applications on the market are pre-configured to work with specific paired servers. There were applications which allowed the user to send location details to a server as a POST request with a JSON payload, which would provide generalisation – however, these applications had few installs and were not considered reputable enough to test.

|  |  |  |
| --- | --- | --- |
| ***Requirements*** | **Traccar Client** | **CelltracGTS/Free** |
| *Simple user interface* | Text based, 9 settings to configure, large buttons | Very busy interface – shows location data on main screen, 21 settings to configure |
| *Can run on low performance CPU* | Requires Android 1.5 and up. Requires permission to access location features on the device | Requires Android 2.3.3 and up. Requires permission to access location, phone, photos/media/files, storage, device ID and call information on the device |
| *Can run on low resolution displays* | Unknown | Unknown |
| *Ability to configure the server IP address and port to forward location data* | Yes | Yes |
| *Ability to turn GPS forwarding on and off* | Yes | Yes |
| *Configurable update rate, minimum duration of 1 second* | Yes | Pre-set configurations, varies depending on ‘event’ currently in progress |

**Traccar Client (Traccar Ltd 2017):**

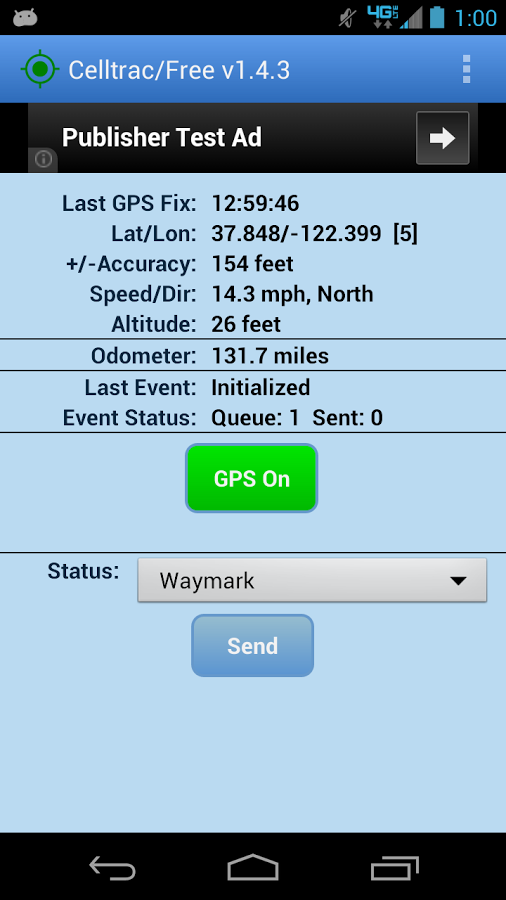
* An application that allows the user to use their mobile device as a GPS tracker
* Reports location to your own or hosted server. By default, it is configured to use a hosted Traccar server
* Is supported by other popular GPS tracking platforms
* Uses OsmAnd protocol to report GPS data in the following format: http://demo.traccar.org:5055/?id=123456&lat={0}&lon={1}&timestamp={2}&hdop={3}&altitude={4}&speed={5}
* Last updated May 2017

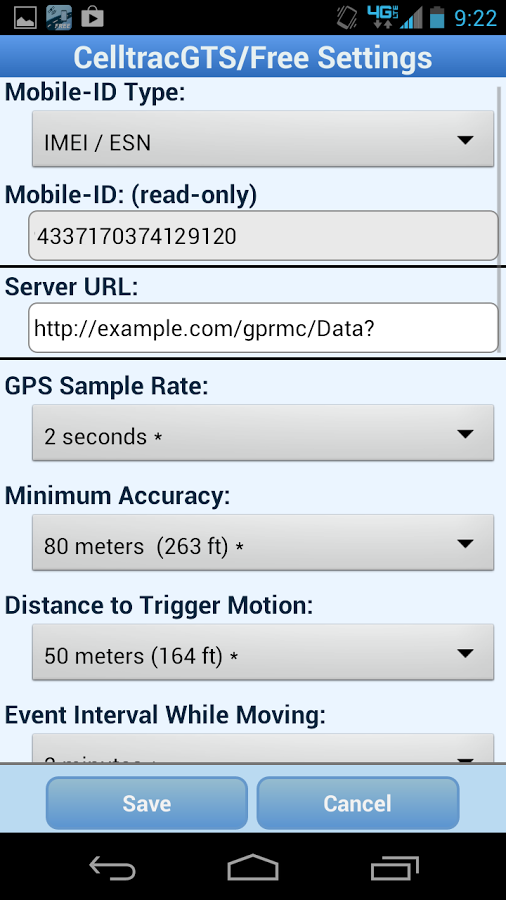


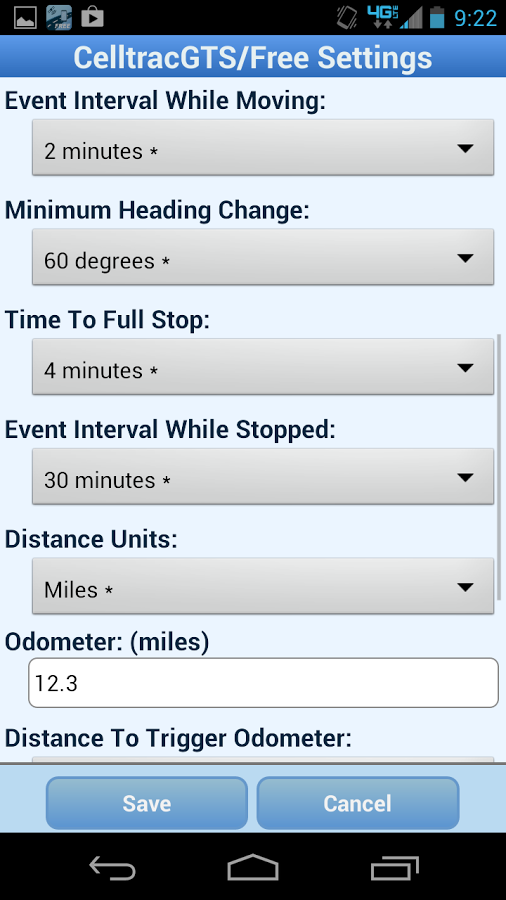


**CelltracGTS/Free (GEOTelematic Solutions Inc 2015):**

* The official Android phone GPS tracking application for OpenGTS server
* Automatic event generation based on current GPS state analysis. Generates a "Start" event when phone moves a selected distance from its original location, followed by periodic "In-Motion" events while en-route, or "Heading-Change" events if the direction of travel changes. When the phone has come to rest for a selected period of time a "Stop" event is generated, followed by periodic "Dormant" events when the phone is still at rest.
* You can configure the application to send location data every 30 seconds while stopped
* To trigger motion, the minimum distance moved is 30 metres. While moving, events are generated every 30 seconds at minimum
* Has advertisements
* Last updated June 2014







**Conclusion:**

Of the two applications, Traccar Client far surpasses CelltracGTS/Free in terms of both functional requirements and usability. While both applications provide the ability to configure the server IP address and port, as well as the ability to turn GPS forwarding on and off, CelltracGTS/Free is lacking a crucial feature – the ability to configure the reporting frequency of location to a minimum of one second. Instead, CelltracGTS/Free offers wider reporting frequencies, which are based on the current event status of the device. The event status of a device is not a requirement of the project – all that is currently required is the location data. The ability to configure the reporting frequency based on event status means that the user interface on the CelltracGTS/Free application is much more convoluted – in total there are 21 configurable options. Traccar Client, on the other hand, can be configured to report the device location in one second intervals. In addition, Traccar Client has a much simpler interface, with just 9 configurable settings. Traccar is also compatible with older versions of the Android operating system, as well as requiring fewer device permissions than CelltracGTS/Free, meaning it is likely more suitable for cheaper, less powerful, or older devices. The one drawback of Traccar Client is the fact that it uses the OsmAnd protocol to send data. OsmAnd is an open source GPS navigation and map application for smartphone and tablet devices that utilises OpenStreetMap data. The benefit of using OsmAnd over the NMEA protocol is that an ID can be sent along with location data. However, as the project requires a specific three-line format for the stream sent to the data diode socket, which includes a GPRMC NMEA sentence, the OsmAnd data will need to be converted appropriately. Fortunately, there exists free software for GPS data conversion, such as GPSBabel (Lipe 2017). Another option would be to modify the source code, which is available on GitHub (Tananaev 2017), so that location data is sent in a desired format. In conclusion, for a simple, configurable application that sends location data to a given server, Traccar Client appears to be an ideal solution.

### Server Application:

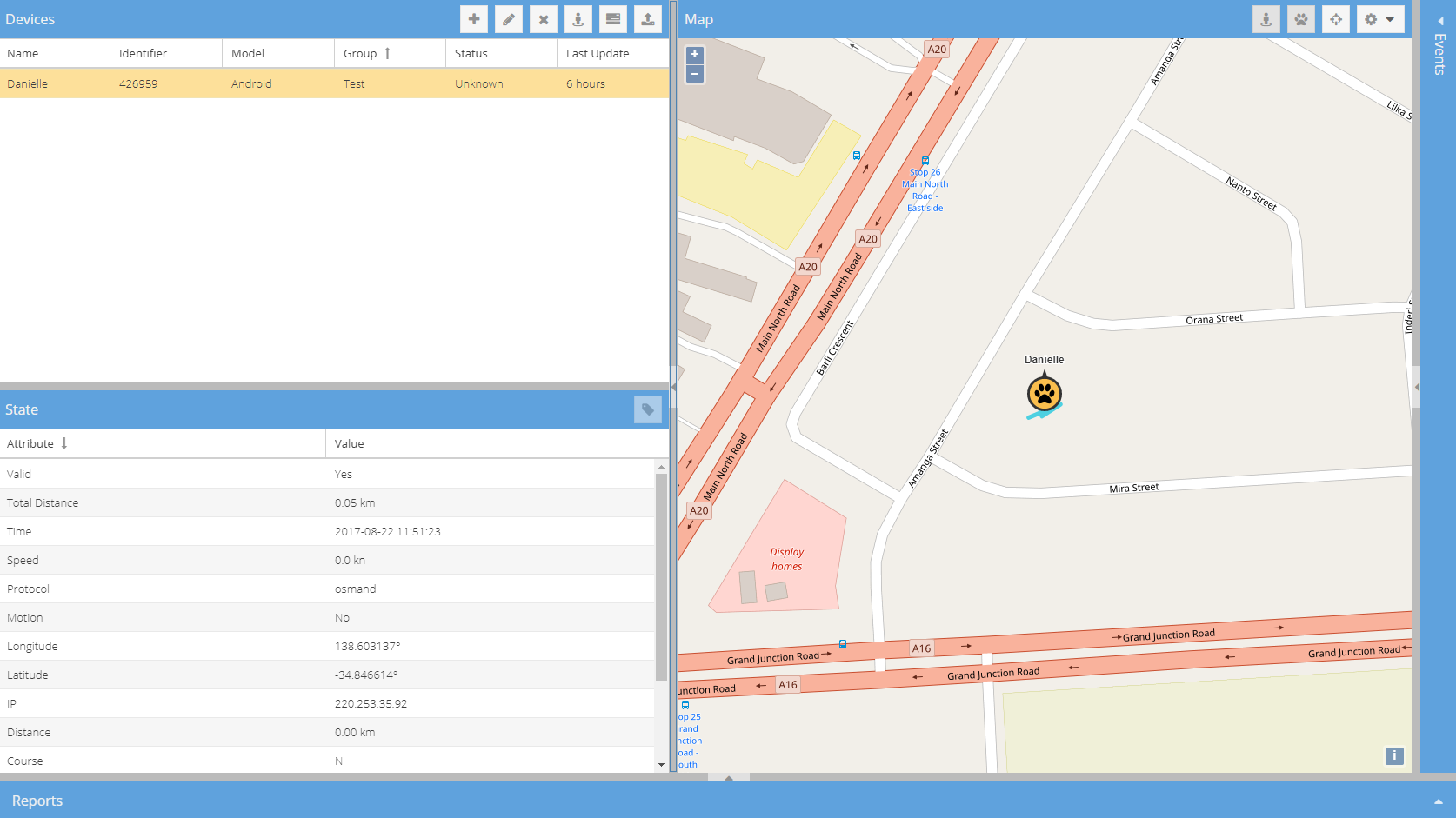
The three server applications analysed were Traccar Server, OpenGTS and Traccar Web UI, a modified version of the original Traccar Server project. As Traccar Web UI is forked from the original Traccar Server, it retains many of the same backend functionality and capabilities – therefore, I have not documented behaviour that is identical between the two applications.

|  |  |  |  |
| --- | --- | --- | --- |
| **Backend Server** | | | |
| ***Requirements*** | **Traccar Server** | **OpenGTS** | **Traccar Web UI** |
| *Ability to configure the IP address and port to forward UDP stream to* | Unknown | Unknown |  |
| *Able to log received messages from Android devices* | Data is stored in a default H2 DBMS, although a MySQL DB can be configured | Default datastore is MySQL |  |
| *Implement a lookup table for friendly/hostile forces* | Unknown | Unknown |  |
| *Open standards for serving information* | Unknown | Unknown |  |
| *The ability to configure a REST endpoint to retrieve logged messages* | Supports a Swagger JSON-based REST API. There is a request that allows the user to pass a device or group ID, in addition to a start and end time and receive a list of positions | Supports a XML-based RESTful interface that allows other programs and tools to query and retrieve data from the system | Simplified JSON-based version of RESTful API which exposes all methods from DataService interface class. There is a request that allows the user to pass a device and start and end time and receive a list of positions |
| *Able to package software to allow for the installation on standalone networks* | Available on Windows, Linux and other platforms. Server can be hosted on-premises or in the cloud | Written in Java, using Apache Tomcat for web service deployment. Will run on any system which supports these technologies. Recommended server has 4GB RAM, 100GB disk space and requires root access |  |
| *Outgoing UDP stream should occur whenever device feed is coming in, or whenever a user is replaying recorded data* | Unknown | Unknown |  |
| *Support for numerous protocols and devices* | Supports over 140 GPS protocols and over 1,000 GPS tracking devices. If using the Traccar Client application, the port is always 5055 | Devices from different manufacturers can be tracked simultaneously. Out-of-the-box support for a number of devices, with the ability to integrate other devices with custom coding |  |

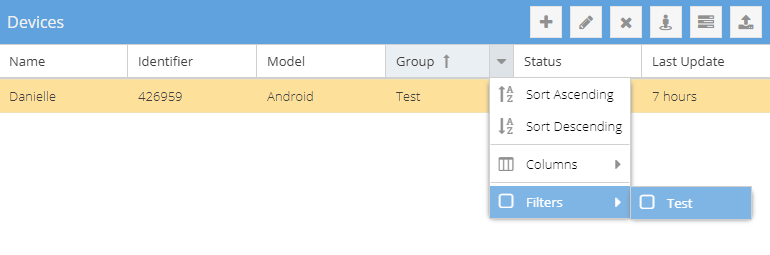
|  |  |  |  |
| --- | --- | --- | --- |
| **User Interface** | | | |
| ***Requirements*** | **Traccar Server** | **OpenGTS** | **Traccar Web UI** |
| *Ability to zoom in and out and pan the map* | Yes | Yes, also has the ability to measure straight-line distance between any two points |  |
| *Able to add or remove tracked devices* | Yes – but removed devices can no longer generate reports. There is the possibility to ‘unlink’ devices from a user and have only the admin ‘remove’ devices |  |  |
| *View positions of devices on map* | Observe GPS devices in real-time | Location information produced from devices is transmitted to the server. The server saves this information in the database and places it on a map when needed. When a user opens a map page, all location information is read from the database and sent to the page as a static document. The page can be refreshed as often as desired to give the feeling of real-time updates. You can configure the private.xml file to define a refresh rate and have an auto-refresh button in the map. You should be able to gain near-real-time display by configuring a map refresh period equal or less than the device update rate |  |
| *Toggle the display of individual devices on and off* | Devices can be assigned attributes, such as name, model, contact or group. There is then an option to filter by certain attributes, and have only matching attributes display on the map. However, you cannot simply select or deselect devices by checkbox | There are two types of maps – ‘Vehicle’ and ‘Group’. Within the ‘Vehicle’ map, you select the device you want to view from a dropdown box | Each device has a checkbox which can be selected or deselected to toggle the display of that device |
| *Toggle the display of friendly/hostile devices on and off* | It is possible to assign a device to a group and then filter by group so only devices assigned to a particular group display on the map | It is possible to assign a device to a group, and then display the ‘Group Map’ for that cluster of devices. However, the ‘Group Map’ only shows the last known location of devices |  |
| *Browser based* | Web interface with desktop and mobile-friendly layouts, including native mobile applications for Android and iOS platforms | Desktop friendly web interface | Web interface with desktop and mobile-friendly layouts |
| *Open standards for visualising information* | Unknown | Unknown | Unknown |
| *Utilise internet-connected foundation imagery services* | Various map options, including road maps and satellite imagery. Default tile server is OSM, but Bing Maps, Google Road, Google Satellite, ArcGis Topo, ArcGis Imagery and Wikimedia are available | Default tile server is OSM, but supports Google Maps, Microsoft Virtual Earth and Mapstraction. Other mapping service providers can be integrated if needed |  |
| *Allow the user to assign devices as friendly/hostile* | Categories can be assigned to individual devices. However, there is no current feature to filter by category. The colour of the icon is currently dependent on whether the device is online, offline, or unknown. You can also assign a device to a group and filter by groups instead | Pushpins can be assigned based on parameters in the database. Devices can also be assigned to groups | Icons can be assigned to individual devices. However, there is no current feature to filter by icon. The colour of the icon is currently dependent on whether the device is default, selected, or offline. You can also assign a device to a group and filter by groups instead |
| *Able to handle at least 20 devices, but extensible to handle more* | Can handle as many devices as your server hardware can handle. Has been tested with 100,000 devices. Traccar is Java based, so it is sensitive to amount of RAM. At least 1GB is recommended for about 100 devices | Has been used to track thousands of devices on a single server, however this is dependent on the event reporting interval, network speed and computer resources. The minimum recommended RAM is 4GB |  |
| *Able to support at least 5 logon accounts, but extensible to handle more* | Yes. There are three main user roles – admin, manager and user. Admin has unlimited access to the entire server, manager is a user with extended capabilities allowing them to manage their subset of users and register new ones and user is an ordinary user that can manipulate any of his own devices and add new ones | Each account can support multiple users and each user has its own logon credentials and controlled access to sections within their account. You can assign device groups to a user, and control their permissions for editing vehicles, producing reports, etc |  |
| *Select date and time to replay recorded data (replay should display on UI)* | Supports reports such as location history and trips. Reports are viewable in the web or mobile application, or can be downloaded as an Excel file. History can be projected on the map, as long as the device has not been removed. The history does not ‘replay’. However, data for devices may persist on the database | Historical data can be reported on using an internal XML-based reporting engine. For a given timeframe, a single device can be selected to replay the path the device has taken. The replay does not occur at the speed of the original recording – it is sped up |  |
| *Pause all incoming traffic to allow for replay of recorded data* |  | All incoming traffic will continue to be fed into the database as a track is replayed. A replay is simply grabbing data from the database and plotting it to a map in a controlled manner |  |
| *Only visualise what is available in GPRMC message (time, date, latitude, longitude, speed in knots, true course, variation) and ID and device model* | Server can handle a wide variety of additional information from GPS units |  |  |

**Traccar Server (Traccar Ltd 2017):**

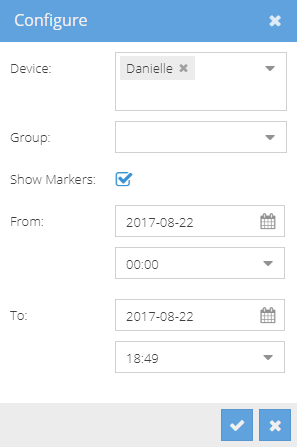
* An open source server for various GPS tracking devices
* Written in Java and works on most platforms with installed Java Runtime Environment
* Last release 9 July 2017



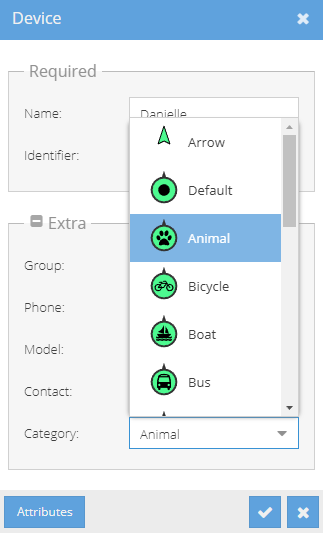
Device list, device state and device position on map



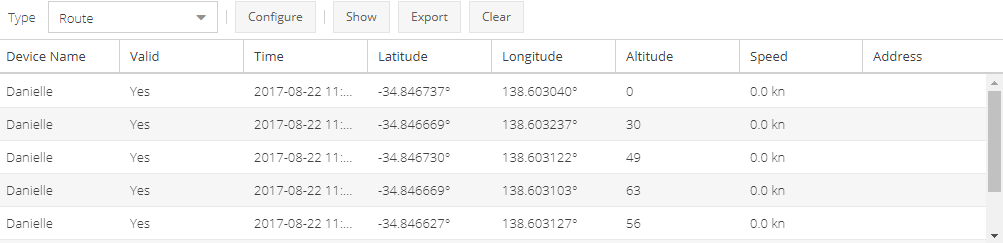
Filtering by group name – unselected values are not displayed on the map



Configure a time period and devices or groups to retrieve archived data



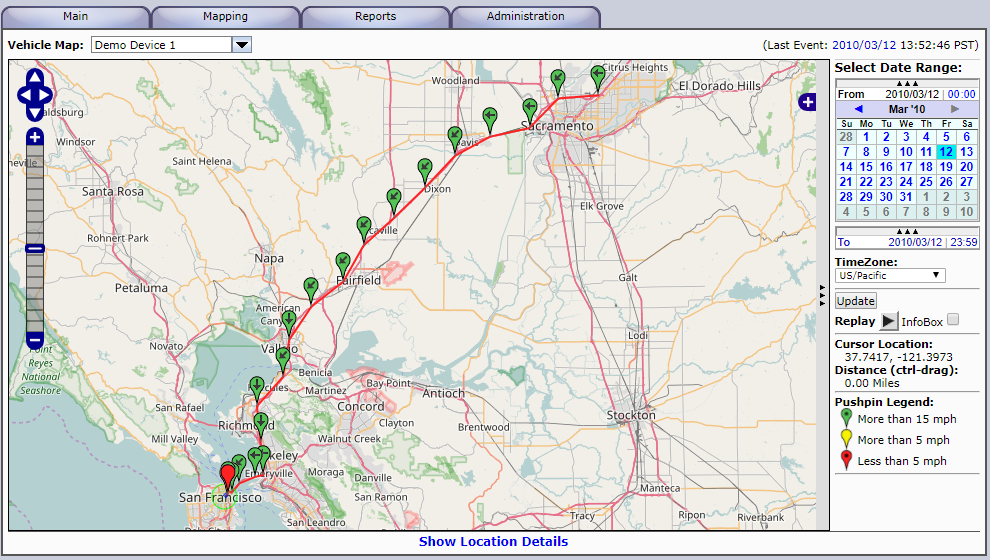
The device category can be changed, which is represented as an icon for the device



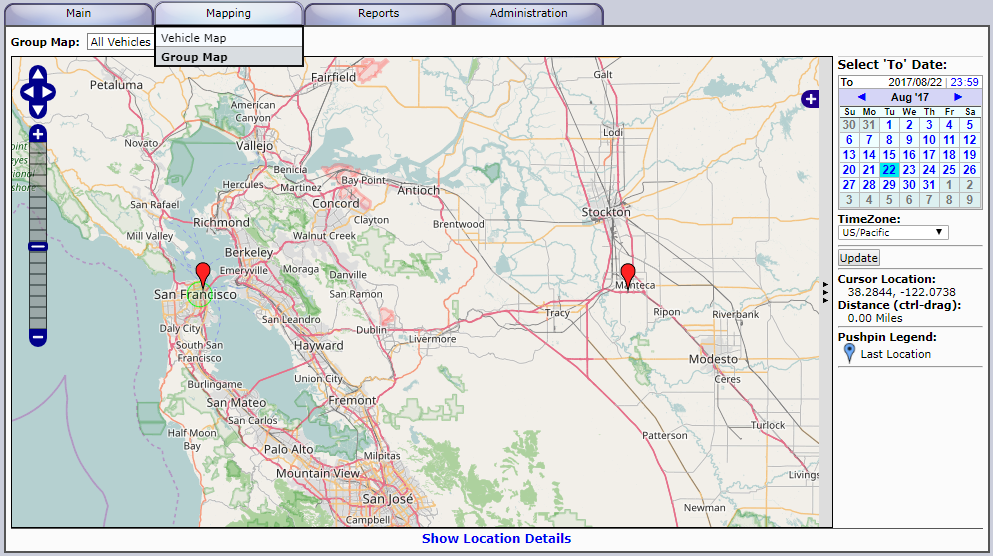
Archived data is displayed in a list and is also visualised on the map

**OpenGTS (GEOTelematic Solutions Inc 2015):**

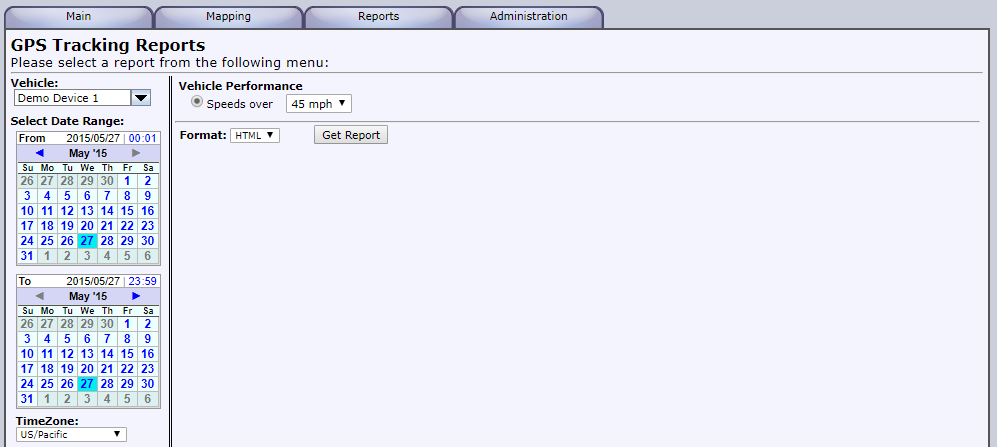
* An open source project designed specifically to provide web-based GPS tracking services for a "fleet" of vehicles
* The replay function within the ‘Vehicle Map’ shows the events of a single device at it travels down the road. Each pushpin is related to the next in that it represents a chronological ordering of locations where the vehicle was located at that particular time. When showing all points of the selected device on a map, the replay allows viewing the order in which these events occurred. The replay is handled within the client browser, using Javascript to make each point visible on the map as the replay interval progresses - all points are already cached in the browser from the previous "update"
* The ‘Group Map’ shows all devices within the selected group where they were located as of the selected "To" date (since only a single pushpin is shown for each device, the "From" date is not needed). Each pushpin represents a single device, independent of all other pushpins on the map which represent other devices, with the only relationship between points being that they belong to the same device group. A 'replay' on the group map would not be suitable, since each pushpin/vehicle is independent of the other pushpins/vehicles.
* Much more suitable as fleet management software
* Has a vast library of documentation and guides
* Last release 14 April 2017



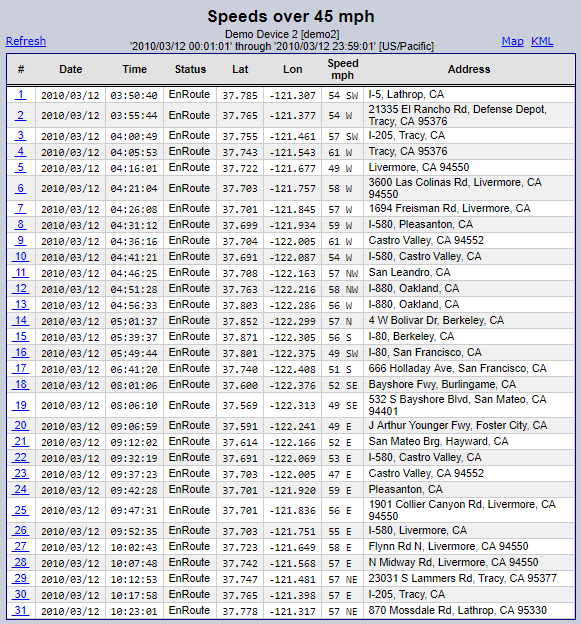
Individual devices are selected from the dropdown box ‘Vehicle Map’. The map displays the route of a device for a selected time period



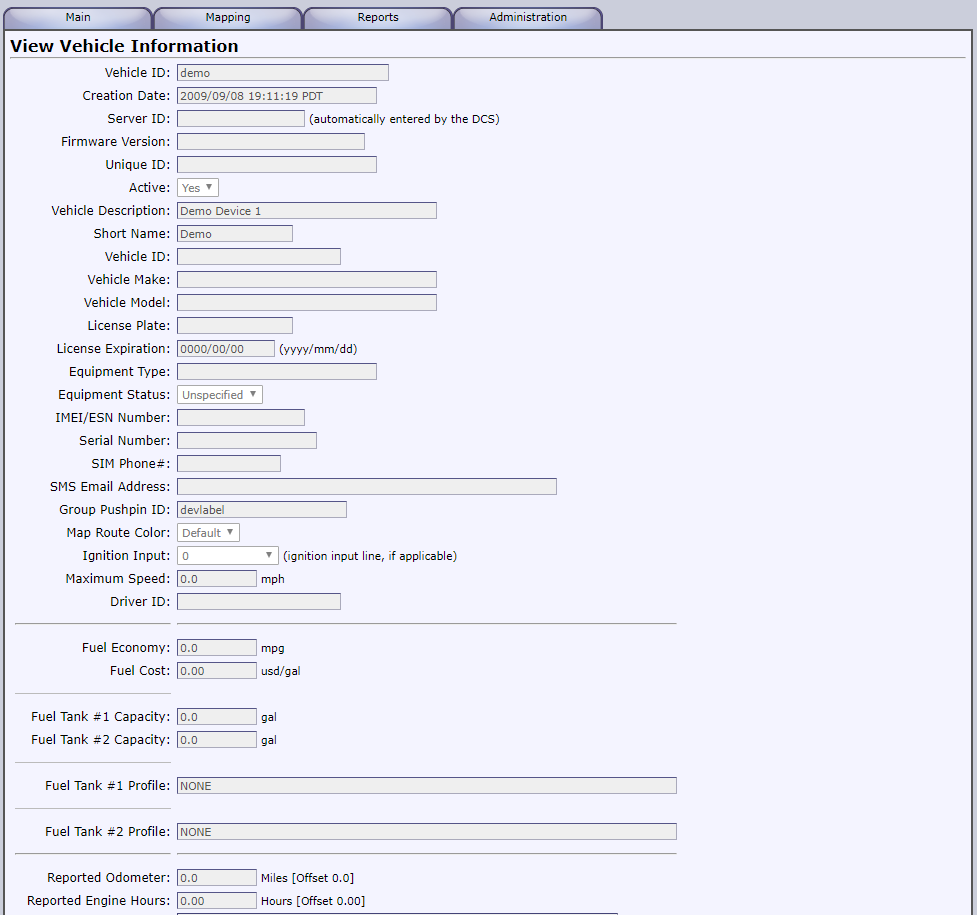
To display groups, the ‘Group Map’ must be selected. Only last known locations of devices within a group are displayed on the map



Text based reports can be generated for an individual device or group of devices for a selected time period



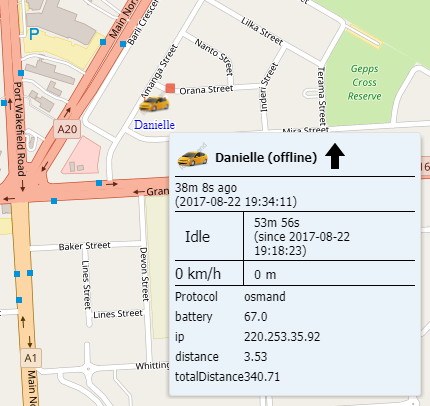
Reports can then be viewed on a map, or a KML file can be produced in the browser



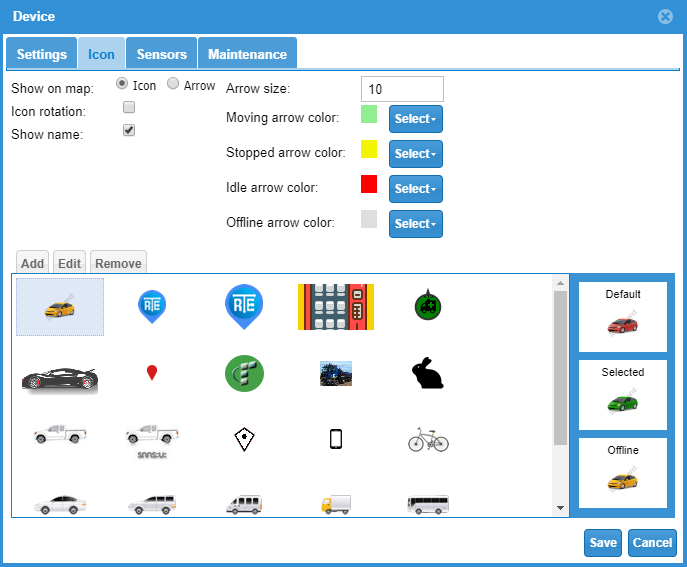
Vehicles (devices) can be thoroughly configured with pre-set attributes

**Traccar Web UI (Litvak 2014):**

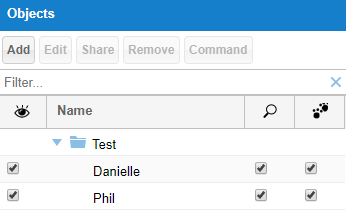
* A modified version of the original Traccar Server, with additional features
* Last release 17 Dec 2016



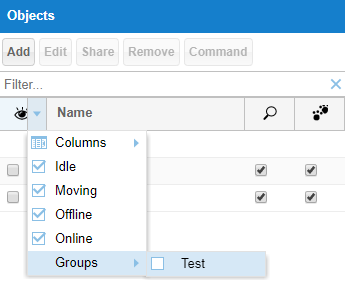
Hovering over a device icon on the map displays summary information regarding that device



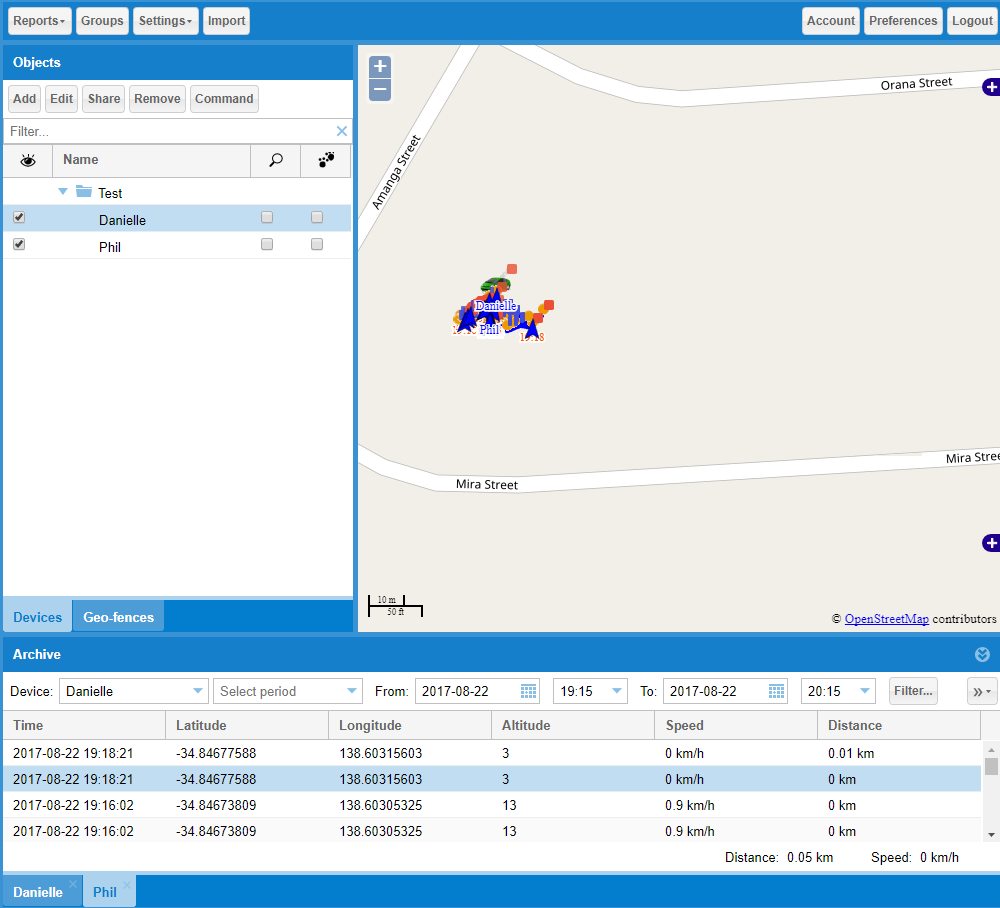
There are a range of icons available to assign to devices



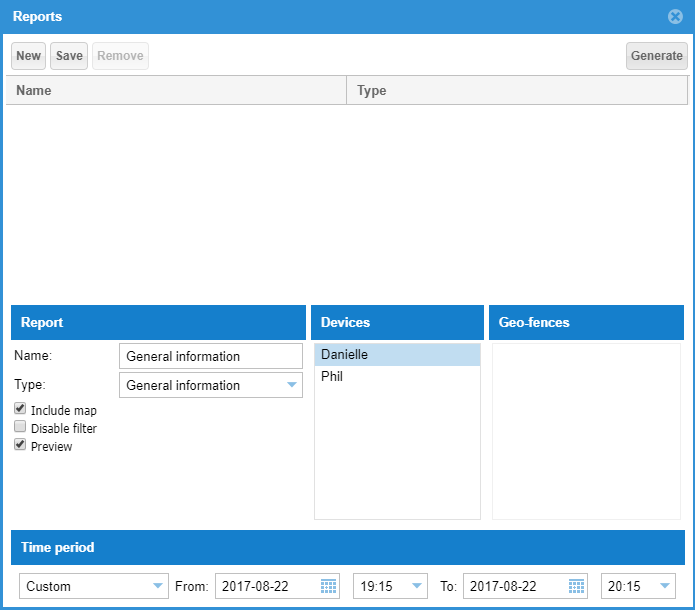
Check boxes allow the user to select individual devices to view on the map



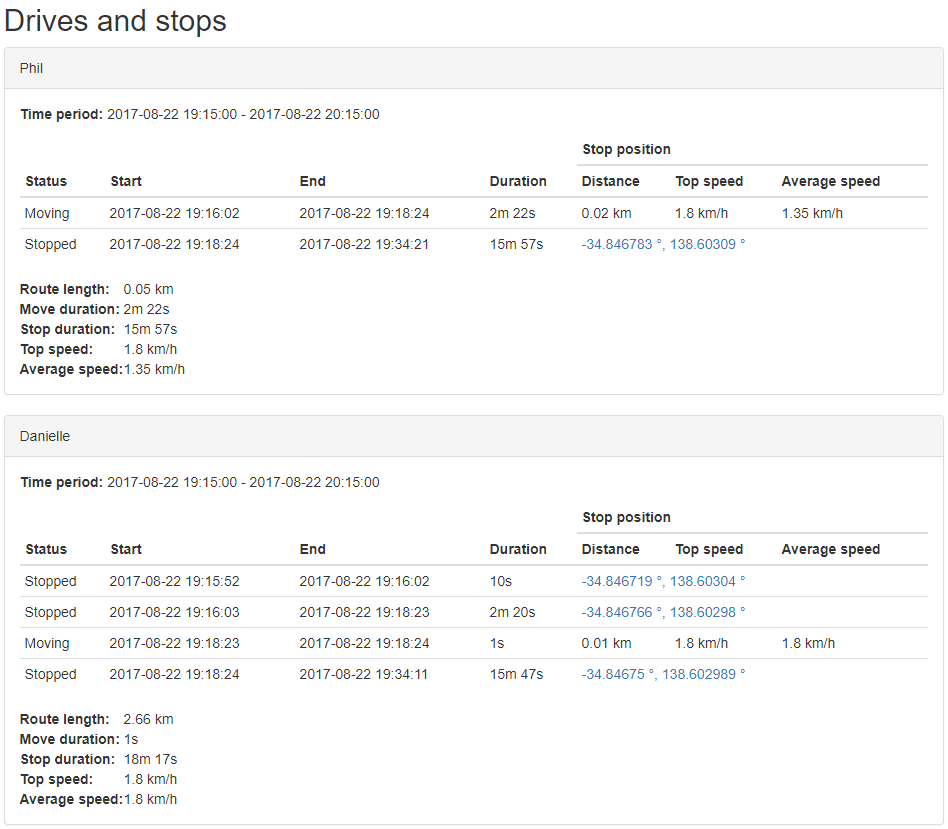
Filtering by group name or status – unselected values are not displayed on the map



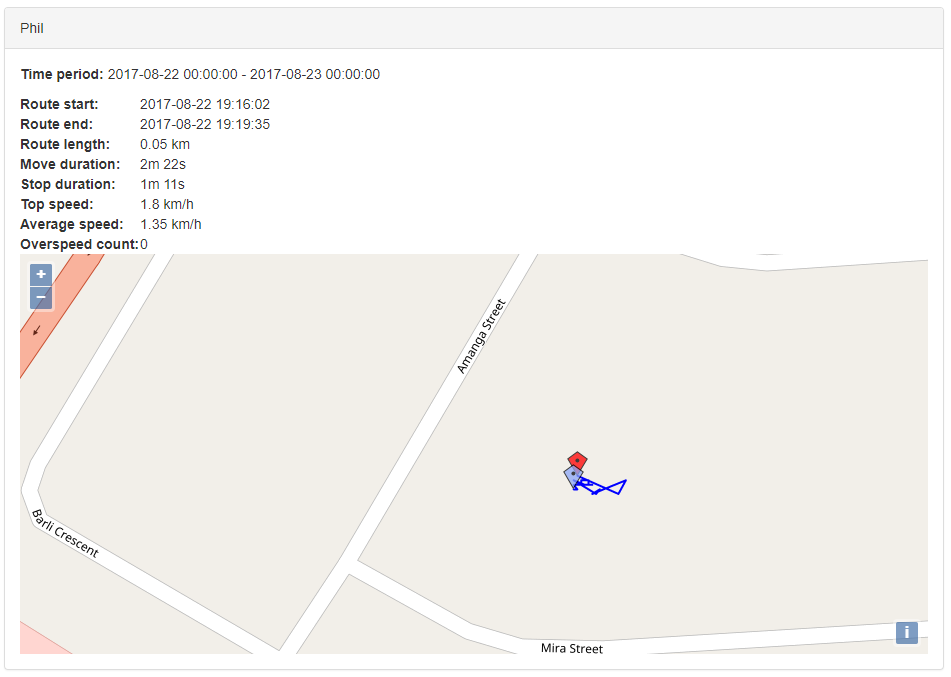
Archives of multiple devices can be loaded simultaneously, and appear as tabs at the bottom of the Archive panel. Archives can be exported as a .GPX file



Individual or multiple devices can be selected to generate a report. The current release does not allow groups to be loaded, however there has been a recent commit that does allow this functionality



After reports have been generated, the report for each device is shown one after another



Relevant reports can be displayed with a map of route data, however, multiple reports do not concatenate routes on to one map – each device displays its own map

**Conclusion:**

Of the three applications explored, Traccar Server and Traccar Web UI appear to be better suited to the scope of the project. In terms of usability, the two Traccar-based applications are much simpler to operate as they make use of panels for the major components of the user interface – devices, state, map, archives and events. OpenGTS, on the other hand, uses tabs and drop down boxes to switch between components, which provides for a clunky experience. The fact that individual and groups of devices cannot be displayed on the same map means one of the key requirements of the project cannot be met – the ability to distinguish the location of many devices at the same time to improve situational awareness. The map should be able to visualise more than one individual device at a time, or more than one group of devices at a time, or any combination of these two scenarios. Although OpenGTS does have the ability to replay track data, which both Traccar applications lack, the functionality is minimalistic. Replay can only occur for one device at a time, and is not configurable – each point in a route is simply made visible after certain period of time. OpenGTS appears to be much better suited for vehicular fleet management, which is made apparent through the ‘Vehicle Information’ screen, which allows the user to configure preset attributes of the device, such as licence plate, fuel economy and odometer.

Traccar Server and Traccar Web UI appear to provide functionality better suited for utilisation within the project. They both support the real-time display of devices on a map, and allow for the display of all individual and groups of devices at the same time. The ability to monitor the position of many devices simultaneously is reminiscent of Blue Force Tracking systems, for which the purpose of this project is to emulate. Both applications allow for the filtering of devices and groups to display on the map, although Traccar Web UI has the added feature of checkboxes for selecting individual devices, which is user friendly. Traccar Web UI also displays the state of a device when hovered over, while Traccar Server displays this information in a panel. One of the nice features of Traccar Server is the ability to define attributes of a device, which could prove useful for assigning a friendly or hostile status to a device – however, Traccar Web UI offers the ability to upload and use custom icons to represent devices, which may be useful in visualising friendly and hostile forces in a glance. Traccar Server allows for archived location data for a group of devices to be displayed, while Traccar Web UI forces the user to select each device individually, which is a major shortcoming in terms of usability. One useful feature found within Traccar Web UI that is currently not supported by Traccar Server is the ability to export archived data as a .GPX file. Although the export function only works for individual devices, having access to a common GPS data format could be useful for replay functionality – GPS data can be uploaded to software such as MyGPSFiles, which in turn replays the track in real time.

In conclusion, both Traccar Server and Traccar Web UI are much more feasible options to use as the base for further development within this project than OpenGTS. However, choosing between the two Traccar-based applications is much more difficult – they both include noticeable benefits and drawbacks over one another.

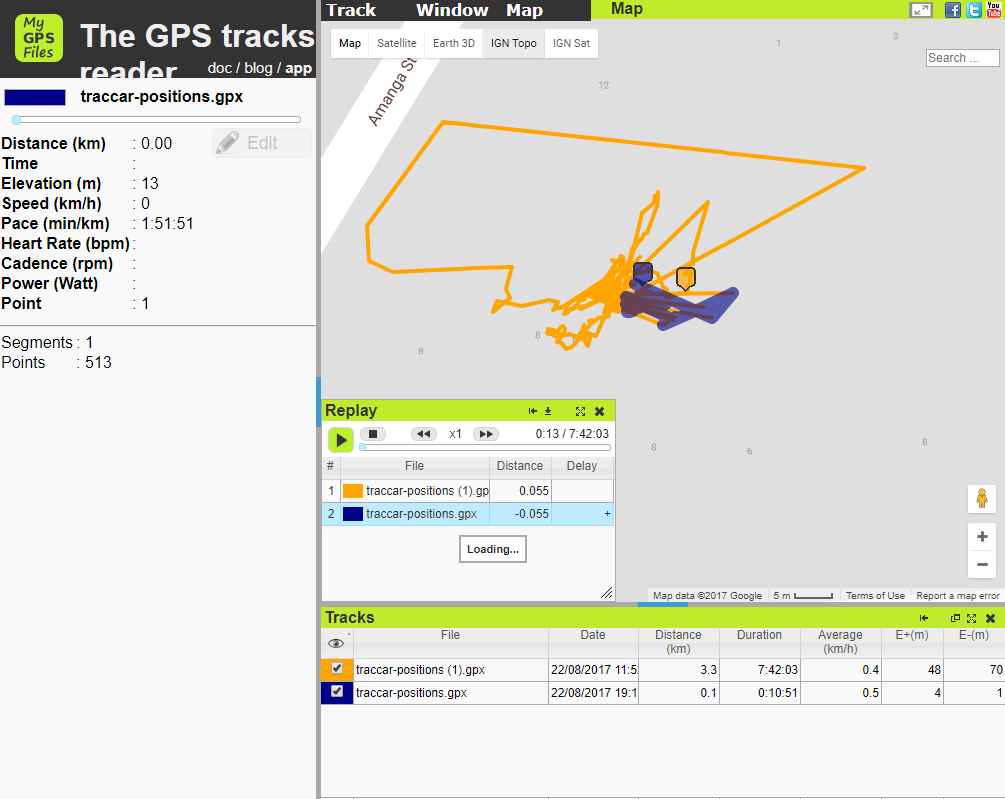
### Other Software:

During investigation, it became apparent that the method and functionality of replaying recorded messages required by the project was not something that any of the explored technologies possessed. Traccar and Traccar Web UI did not feature a replay functionality, and while OpenGTS did, it was limited to a single device at a time, and playback occurred at an increased speed, with no option to modify this within the user interface. A brief scan of existing software revealed MyGPSFiles, which may provide a standalone solution to this problem.

In addition to replay software, there may be a need to convert GPS data. GPSBabel may prove useful in this regard.

**MyGPSFiles (MyGPSFiles 2017):**

* A web browser application with the ability to read and display uploaded GPS tracks
* Supports gpx, tcx, crs and fit files
* Multiple tracks can be opened and displayed at a time
* Track data can be replayed, but replay is relative - at time = 0, all cursors are set at the beginning of each track. They then move from this point. Starting points are therefore important
* Windows, Linux and Mac compatibility



Two tracks captured through Traccar Web UI being replayed. The file name and track colour can be changed

**GPSBabel (Lipe 2017):**

* Supports the conversion of waypoints, track and routes between GPS data types
* Free to download, use and modify
* Runs on Windows, Mac and Linux

### Conclusion:

In terms of development effort, it appears to be a much more feasible option to utilise and further build upon existing software. This project will likely utilise Traccar Client and either Traccar Server or Traccar Web UI for its key deliverables, in addition to making use of GPSBabel and MyGPSFiles as required.

### References:

Tananaev, A 2017, *traccar-client-android,* GitHub, GitHub Inc, viewed 20 August 2017, <https://github.com/tananaev/traccar-client-android>.

GEOTelematic Solutions Inc 2015, *CelltracGTS/Free GPS Tracking Application for Android Phones,* GEOTelematic Solutions Inc, GEOTelematic Solutions Inc, viewed 20 August 2017, <http://www.geotelematic.com/CelltracGTS/Free.html>.

GEOTelematic Solutions Inc 2015, *OpenGTS – Open GPS Tracking System,* The OpenGTS Project, GEOTelematic Solutions Inc, viewed 20 August 2017, < http://www.opengts.org/>.

Lipe, R 2017, *GPSBabel,* GPSBabel.org, viewed 23 August 2017 <https://www.gpsbabel.org/index.html>.

Traccar Ltd 2017, *Traccar Client,* Traccar, Traccar Ltd, viewed 20 August 2017, < https://www.traccar.org/client/>.

Traccar Ltd 2017, *Traccar Server,* Traccar, Traccar Ltd, viewed 20 August 2017, < https://www.traccar.org/server/>.

Litvak, V 2014, *Traccar Web UI,* Vitaly Litvak, viewed 20 August 2017, <http://traccar.litvak.su/index.html>.

MyGPSFiles 2017, *The GPS tracks reader,* MyGPSFiles, MyGPSFiles, viewed 23 August 2017, <http://www.mygpsfiles.com/app/>.