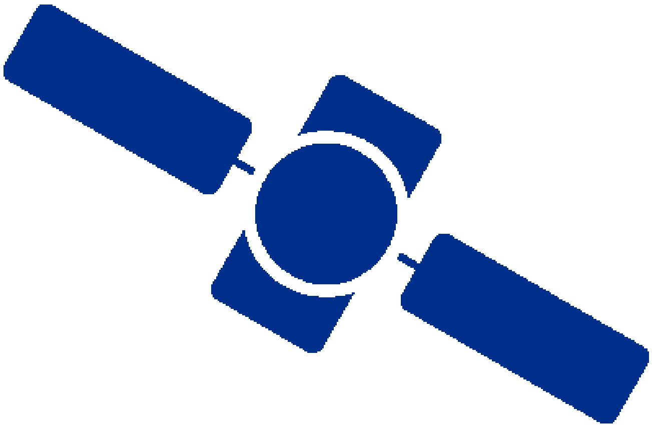
estec
European Space Research
and Technology Centre
Keplerlaan 1
2201 AZ Noordwijk
The Netherlands
T +31 (0)71 565 6565
F +31 (0)71 565 6040
www.esa.int



GNSS compare

Launching Galileo awareness

Prepared by **M. Burba, S. Ciuban, M. Krainski, D. Perz**
Date of Issue **13/11/2017**

European Space Agency
Agence spatiale européenne

1 INTRODUCTION

This is a proposal for the Galileo Smartphone App Challenge, introduced by the Directorate of Navigation (D/NAV), in collaboration with the Directorate of Technology Engineering and Quality (D/TEC) in October 2017. It describes a solution proposed by The Galfins team (Galileo Finders). Section 3 presents an overview of the problem and proposed solution. Section 4 presents a detailed design of the proposed application, specifying what will be implemented and how the User Interface might look like. Section 5 presents a proposed management scheme and development plan. It indicates what will be implemented when and presents risks and opportunities of the project. Section 5.3 describes the validation plan for checking if the application works as expected and the calculated results are correct. Section 6 presents team members of The Galfins (Galileo Finders), lists their strong points and describes how they will cooperate. Section 7 provides a short summary of the document.

1.1 Applicable Documents

Ref.	Title
AD1	Terms & Conditions, Galileo APP Competition
AD2	Prelimiary Technical Architecture

2 ACRONYMS AND ABBREVIATIONS

API Application Programming Interface

CEP Circular Error Probability

D/EOP Directorate of Earth Observation Programmes

D/HRE Directorate of Human Spaceflight and Robotic Exploration

D/NAV Directorate of Navigation

D/TEC Directorate of Technology Engineering and Quality

ECEF Earth Centered Earth Fixed

EKF Extended Kalman Filter

ESA European Space Agency

ESTEC European Space Research and Technology Center

GLONASS Globalnaya Navigazionnaya Sputnikovaya Sistema

GNSS Global Navigation Satellite System

GPS Global Positioning System

IMU Inertial Measurement Unit

PVT Position, Velocity and Time

RMSE Root Mean Square Error

RTK Real Time Kinematic

SLAM Simultaneous Localization and Mapping

SPAN Synchronous Position, Attitude and Navigation

TGVF Time and Geodetic Validation Facility

UI User Interface

WLS Weighted Least Squares

3 DESCRIPTION

This document describes The Galfins's proposal for the Galileo Smartphone App Challenge, introduced by the D/NAV, in collaboration with the D/TEC in October 2017. The Challenge is about creating a smartphone application, which will allow the user to choose which satellite constellation to use for Position, Velocity and Time (PVT) estimation. The aim is to increase awareness about ESA's Galileo program and allow users from common public to compare the performance of Galileo signals with the performance from other satellite constellations.

The main requirements for the application, listed in AD1, are:

- allow the user to select the constellations used for navigation, enabling at least Galileo only and Global Positioning System (GPS) only positioning, simultaneously,
- provide direct and real-time visualization on an open source world-wide map of the computed positions,
- allow the user to log their positions and the raw measurements used to compute such positions,
- demonstrate the quality of Galileo signals in the most modern and visual way.

In order to fulfill those requirements, we propose an Android smartphone application, which will use raw measurements from Global Navigation Satellite System (GNSS) satellites to calculate user's PVT. The user will be able to select parameters for two, calculated in parallel, processing schemes. The results of calculations will be displayed in a modern, graphical way, utilizing Material Design principles [1] and an open source map, e.g. OpenStreetMap [2]. The user will be able to decide which constellation of satellites should be used, what correction methods to apply and how the final PVT calculation should be performed. Thanks to this, the user will be able to compare the performance of selected GNSS constellations, learn how various correction methods improve the result and compare various PVT calculation schemes.

Each of the selectable components will include a description, for the user to read and learn about how it improves the position estimation and under what conditions it works best. This relates to both correction methods and PVT calculation methods.

During the design phase of the proposed application, it has become evident, that with little additional effort, the application can provide a functionality for advanced users. By advanced users, we understand organizations who are developing PVT or correction algorithms and plan to perform tests of their algorithms for GNSS positioning on a smartphone. For such users, the proposed application will allow storing raw measurements and the calculated positions in a chosen format. The application will also provide an Application Programming Interface (API) for research developers to easily add their own correction methods or PVT calculation algorithms to the processing scheme, allowing for comparative testing against standard, already implemented methods. With this, we propose a complete framework for testing and development of GNSS testing related applications.

4 DESIGN

This section describes the design of the application. It focuses mainly on two parts — the architectural design, describing how various components are going to interact with each other, and the User Interface (UI) preliminary design, describing how the user will be able to interact with the application.

GNSS Compare will be developed in Android Studio using JAVA Programming Language. It will be tested and verified using Samsung Galaxy S8+. The application will require Android 7.0 or higher and the possibility to access raw data from the GNSS chip. To retrieve this data `android.location` package will be used.

4.1 Architecture design

4.1.1 Application core

The general schematic of the application's architecture is shown in the Figure 1. At this level only a preliminary design of the architecture and API can be presented. During development it may become clear that more, less or different information should be exchanged between each pair of blocks. In that case the architecture and API will be modified to best fit the application objective.

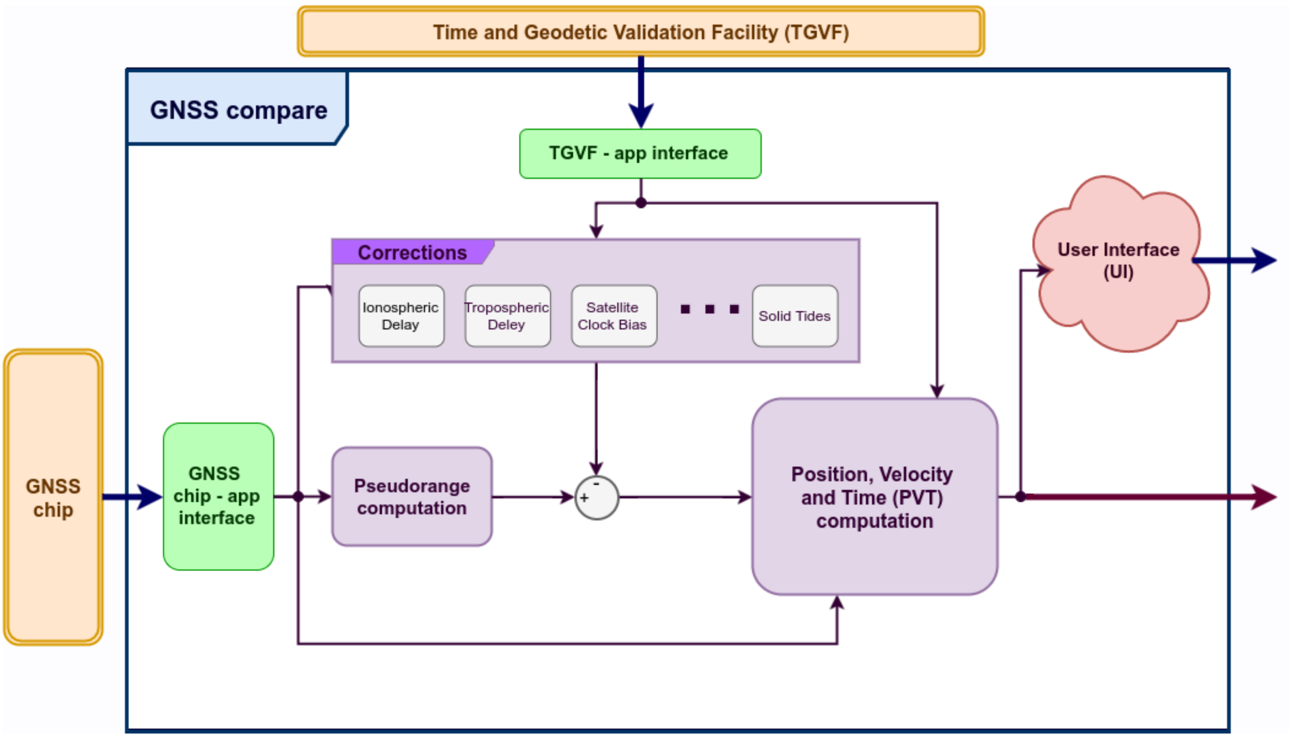


Figure 1: Architecture design

Several types of blocks can be distinguished on the schematic above:

- **External blocks** (*GNSS chip, Time and Geodetic Validation Facility (TGVF)*) - these are components from which *GNSS Compare* will retrieve essential data.

- **Internal interface blocks** (*GNSS chip - app and TGVF - app interface*) - these blocks link *GNSS Compare* with sources of data like external blocks mentioned before.
- **Expert blocks** (*correction blocks, pseudorange computation, PVT computation*) - these blocks include expert navigation knowledge like implementation of the correction computation (see section 4.1.2) or PVT algorithms (see section 4.1.3).
- **User Interface block** - this block includes all the visual aspect of the *GNSS Compare* and describes possible interaction with the user (see section 4.2).

Structure and interfaces of the external blocks are predetermined by Android libraries. For the internal blocks the following preliminary API is proposed:

- `[position of the satellites, exact time of the measurement, environment data]` **`gnssChipInterface`**`(<input compatible with the GNSS chip interface>)`
  This block will be implemented as a `broadcast receiver` Android component.
- `[list of data from single satellite (id, constellation, position of the satellites), estimated position of the receiver]` **`tgvfInterface`**`(<input compatible with the TGVF interface>)`
  This block will be implemented as a `broadcast receiver`.
- `[calculated correction]` **`correctionModule`**`(environment data, position of the satellites)`
  This block will be implemented as a `service` Android component.
- `[computed pseudoranges]` **`pseudorangeCompute`**`(list of data from single satellite)`
  This block will be implemented as a `service`.
- `[calculated position, time and velocity]` **`pvtAlgorithm`**`(corrected pseudoranges, position of the satellites)`
  This block will be implemented as a `service`.
- All UI blocks will be implemented as an `activity` Android components and depending on them being a settings or display screens, they will set or get stored data accordingly.

As seen above, the architecture of the application is based on different blocks. This modularity simplifies the process of adding additional corrections modules or PVT algorithms. By making API public we can allow developers and researchers to implement and test their modules with the *GNSS Compare* application.

Data stored and used within the application can be accessed either by the UI or by logging raw GNSS measurements into an external file. An interesting additional way of sharing the calculated position would be to implement a `content provider` - an Android component that allows data sharing across different applications. One of the use cases for this feature would be to run e.g. a navigation application with position obtained from *GNSS Compare*.

4.1.2 Correction modules

In order to provide accurate positioning results, corrections for various physical phenomena, ranging from effects in the atmosphere and time-space relativity to the influence of gravitation of other celestial bodies on Earth's crust are required. We plan to implement the following correction modules:

1. *Ionospheric Delay Correction*: corrects for the ionospheric delay of the signal due to

varying free electron content resulting in a varying propagation speed in the Ionosphere. Galileo messages provide coefficients which enable the computation of the ionospheric delay. The ionospheric delay can be computed for GPS and Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONASS) from the Galileo coefficients as long as the signal's frequency is known. [3, 4]

2. *Tropospheric Delay Correction*: corrects for the variation of the refractive index, which is a function of temperature, pressure and water vapor concentration. The latter varies on short time scales, whereas temperature and humidity can be considered climatologically. [5, 6, 7]
3. *Satellite Clock Bias Correction*: is needed due to synchronization errors and can be corrected using coefficients given in the broadcast message. These coefficients can be used to create a polynomial, which approximates the behavior of the satellite clock bias. [8]
4. *Relativistic Clock Correction of the Satellite Clock*: is needed due to the difference in gravitation potential between the satellite clock and the receiver clock and their relative speed to one another. The correction has a constant part depending on the orbit semi-major axis and a periodically changing part due to orbit eccentricity. Galileo includes the variable part in the broadcast message. [9]
5. *Sagnac Correction*: corrects for the movement of the receiver during signal propagation. The receiver is moving relative to the satellite due to the rotation of the Earth. [10]
6. *Relativistic Path Range Correction*: corrects for the path prolongation resulting from the curvature of space-time by Earth's gravitation. A term depending on the distance of the satellite and the receiver to Earth's center of gravity, respectively, as well as on their relative distance to one another has to be added to the Euclidean distance between satellite and receiver. A neglection of the relativistic path prolongation is only relevant for precisions on centimeter scale. [8]
7. *Solid Tides Correction*: is to correct for the gravitational impact of external bodies on Earth's crust. The Earth's solid matter moves as function of the position of the Moon and the Sun relative to Earth. [11, 8]

4.1.3 PVT algorithms

For the estimation of the PVT we consider using the available code-based pseudoranges and pseudoranges rates measurements on L1 frequency (1575.42 MHz). Before discussing about the proposals in terms of estimation techniques, we recall the linearized pseudorange measurement model from [8](page 140) and the pseudorange rate equation from [12] (page 2). Assuming knowledge about an approximate receiver position (x_0, y_0, z_0), the position of a certain satellite (*j*), both in Earth Centered Earth Fixed (ECEF) frame and the noise term of the equation neglected, the linear model for the pseudorange measurement is:

$$PR^j - \rho^j - Corr^j = \mathbf{u}^j \cdot \delta\mathbf{r} + \delta t_R \tag{1}$$

where

- PR^j is the measured pseudorange between the receiver and the j^{th} satellite,

- ρ^j is the geometric distance between the approximate position of the receiver and the j^{th} satellite,
- $Corr^j$ represents the results of the applied correction models for the pseudoranges,
- $\mathbf{u}^j = \left[\frac{x_0 - x^j}{\rho_0^j}, \frac{y_0 - y^j}{\rho_0^j}, \frac{z_0 - z^j}{\rho_0^j}\right]$ is the line of sight unit vector between the approximate position of the receiver and the j^{th} satellite,
- $\delta\mathbf{r} = \left[\delta x, \delta y, \delta z\right]$ is the vector containing the receiver unknown position variations,
- δt_r is the receiver clock bias expressed in units of length.

And the pseudorange rate equation is expressed as

$$PRR^j = \mathbf{u}^j \cdot \mathbf{v}_R + \delta\dot{t}_R \tag{2}$$

where

- PRR^j is the measured pseudorange rate,
- $\mathbf{v}_R = \left[v_x, v_y, v_z\right]$ is the unknown receiver velocity vector,
- $\delta\dot{t}_R$ is the receiver clock bias drift expressed in units of length per second.

If we consider that we can access the pseudoranges and the pseudoranges rate measurements from n satellites then the following system of equations, based on equations (1) and (2) can be formed:

$$\begin{aligned}
\begin{pmatrix} PR^1 - \rho^1 - Corr^1 \\ PR^2 - \rho^2 - Corr^2 \\ \vdots \\ PR^n - \rho^n - Corr^n \end{pmatrix} &= \begin{pmatrix} \mathbf{u}^1 & 1 \\ \mathbf{u}^2 & 1 \\ \vdots & \vdots \\ \mathbf{u}^n & 1 \end{pmatrix} \underbrace{\begin{pmatrix} \delta\mathbf{r}^{\mathrm{T}} \\ \delta t_R \end{pmatrix}}_{unknown\ vector}, \\
\begin{pmatrix} PRR^1 \\ PRR^2 \\ \vdots \\ PRR^n \end{pmatrix} &= \begin{pmatrix} \mathbf{u}^1 & 1 \\ \mathbf{u}^2 & 1 \\ \vdots & \vdots \\ \mathbf{u}^n & 1 \end{pmatrix} \underbrace{\begin{pmatrix} \mathbf{v}_R^{\mathrm{T}} \\ \delta t_R \end{pmatrix}}_{unknown\ vector}.
\end{aligned} \tag{3}$$

In order to estimate the unknown vectors from the above observation models, the following estimation techniques are considered for the users to choose from:

1. **Weighted Least Squares (WLS)**
   To estimate the unknowns of the systems of equations from (3), WLS requires at least 4 satellites in view (n ≥ 4), associated pseudoranges, pseudoranges rates and a weighting strategy for the measurements. This should take into account that the signals coming from low elevation satellites are more likely to have lower power, causing the measurements to be less precise, compared to the signals coming from higher elevation satellites. However, the case in which less than 4 satellites are observed when doing single constellation positioning may arise. In this context a standard WLS will not be able to

produce estimates of the unknowns as the systems of equations from (3) will be undetermined. To be able to obtain the PVT with only 3 satellites in view the algorithm presented in [13] and [14] by John W. Betz called *Undetermined WLS solution via Altitude Hold* is proposed . The approach is based on two assumptions that fit the case of the pedestrian and/or dynamic user of our smartphone application:

- a fully determined solution was made before the receiver observes only 3 satellites,
- the user's altitude does not have a significant variation.

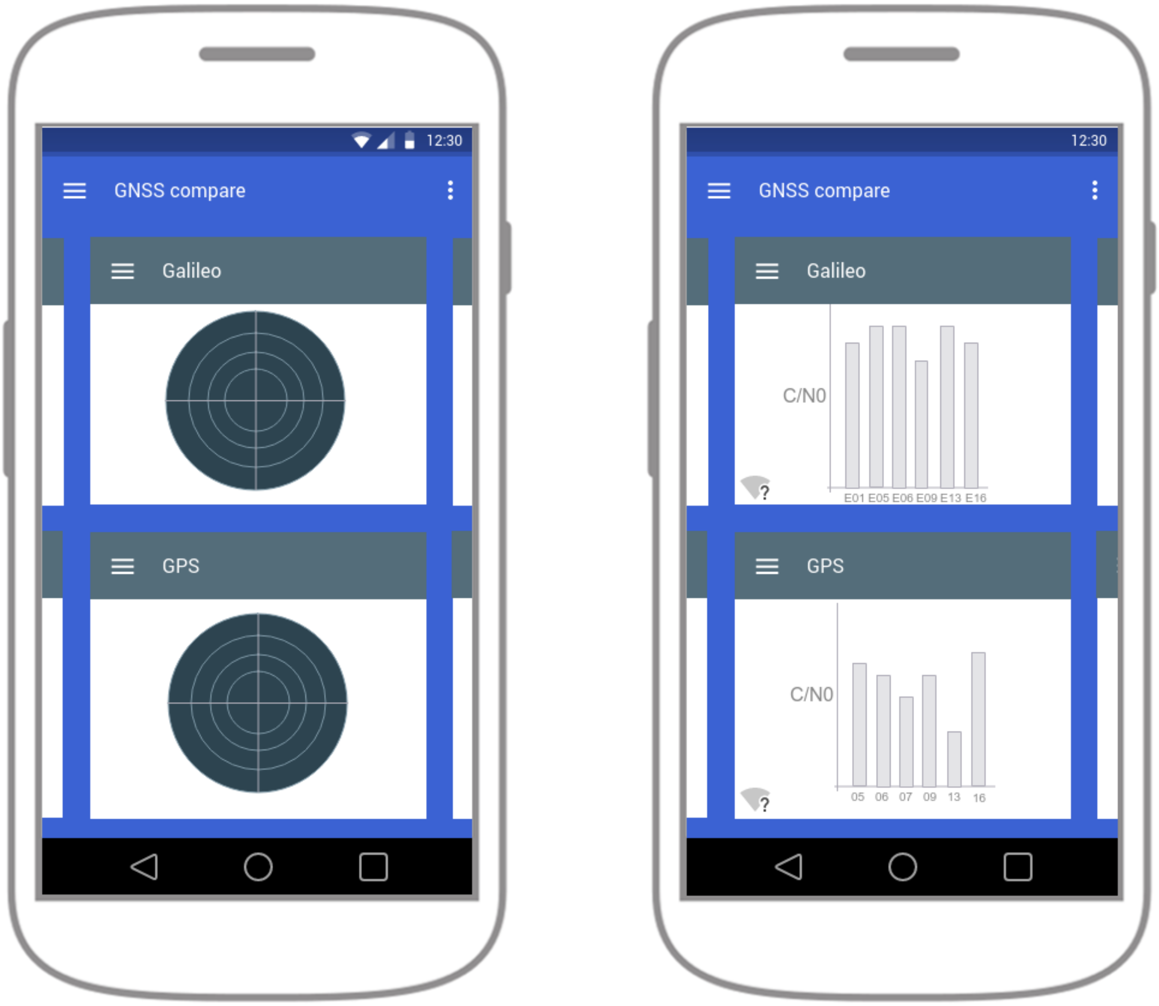
Regarding the implementation of these techniques, the WLS algorithms available on open-source platforms such as *The GPS Easy Suite* [15] will be implemented and enhanced.

2. **Extended Kalman Filter (EKF)**
   The second estimation technique proposed, as an option for the PVT computation, is the EKF. As the EKF is able to combine the knowledge of measurements made upon a system (3) with the mathematical description of the system's physical behavior to infer its current state (e.g., position, velocity), the EKF is more suitable for dynamic cases. Moreover, as mentioned by Mark Petovello in [16], the EKF has some advantages compared to WLS. If we consider the case when less than 4 satellites are observed, the EKF is able to deliver the estimates of the parameters of interest. In that situation, the EKF will rely more on the quality of the implemented dynamic model. The technique is very attractive, especially because it is able to perform in urban canyons and in reduced satellite visibility conditions. Also in this situation, the usage and adaptation of existing open-source algorithms like [15] and Real Time Kinematic (RTK) lib [17] are considered.
3. **Sensor fusion with the Inertial Measurement Unit (IMU)**
   The objective of this method is to fuse GNSS position measurements with measurements provided by the IMU of the smartphone. Doing so, the PVT solution will have an increased robustness against reduced satellite visibility, short total outages (total loss of lock on the satellites) and can bring improvements in terms of accuracy. The fusion is done with the help of an EKF that also takes into account the inertial navigation equations [18](Chapter 5) beside the GNSS measurement modelling.

4.2 User Interface preliminary design

The main screen of the application will display two active viewers. Each viewer will independently be able to display calculated parameters for selected GNSS constellation, corrections and the PVT method. These parameters will include the current positioning accuracy, the signal strength from the satellites, current position on a map, sky plot of the satellites, horizontal domain error etc. Draft design of the viewers has been shown on Figure 2.

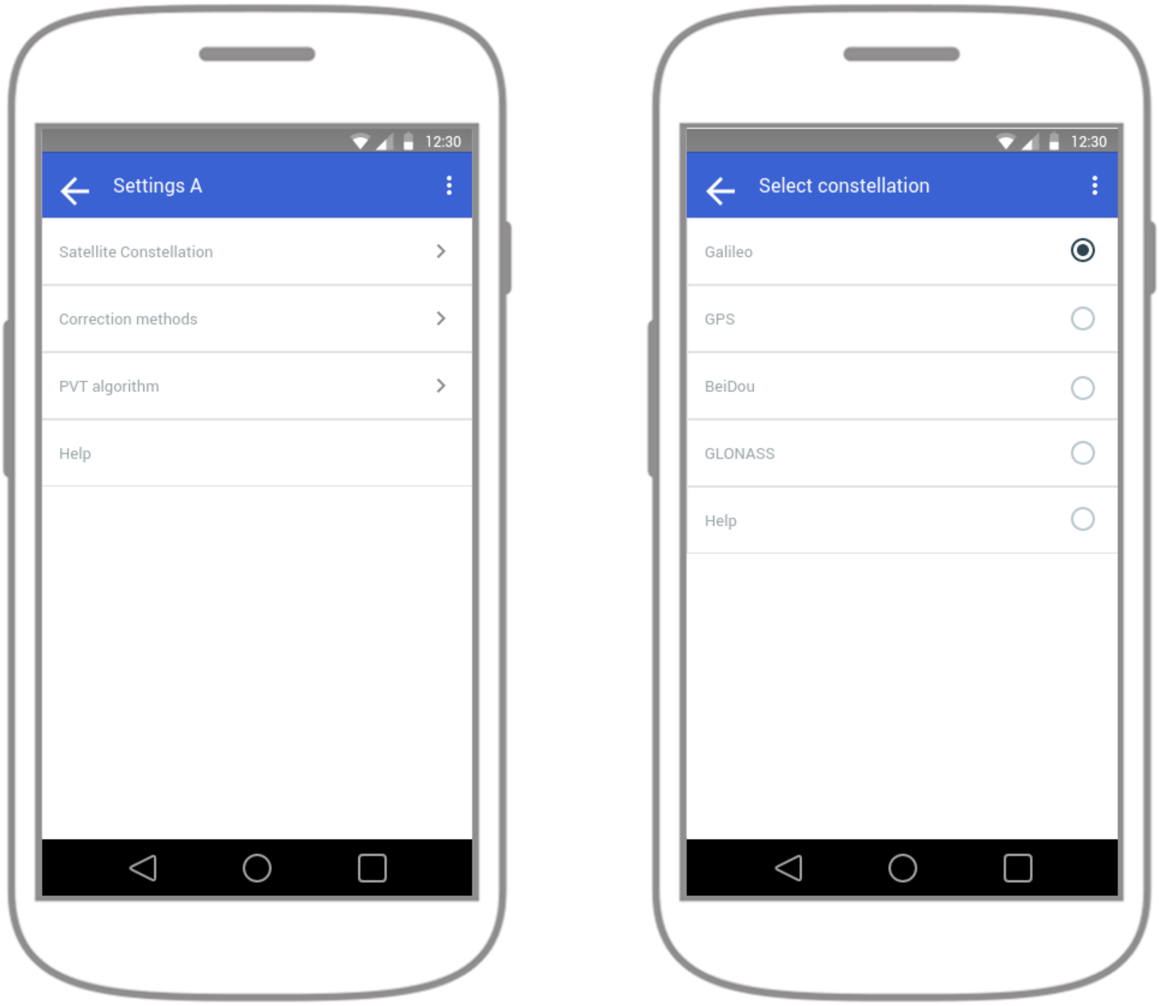
The user will be able to change the displayed parameters, swiping the viewer to the right or left.



(a) Horizontal domain error (b) Signal strength

Figure 2: Draft design of the main screen of the UI.

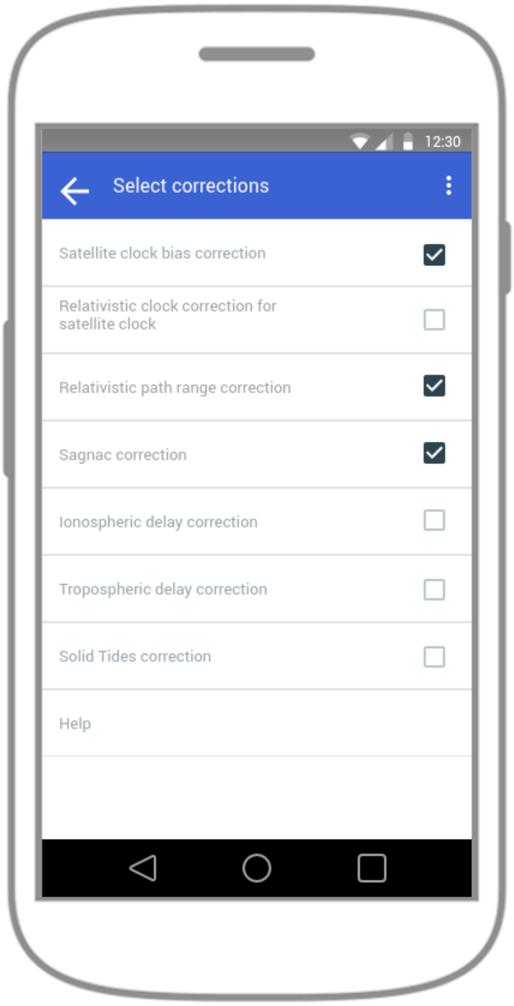
In order to select possible options, the user will enter the settings menu for each viewer independently (Figure 3a), then select used constellation (Figure 3b), then select applied correction modules (Figure 4a), then select used PVT calculation algorithm (Figure 4b). On each step, the user will have a help option available to aid him in the general theories or specific information about the currently selected item (e.g. how neglecting the relativistic clock correction impacts the position estimation).



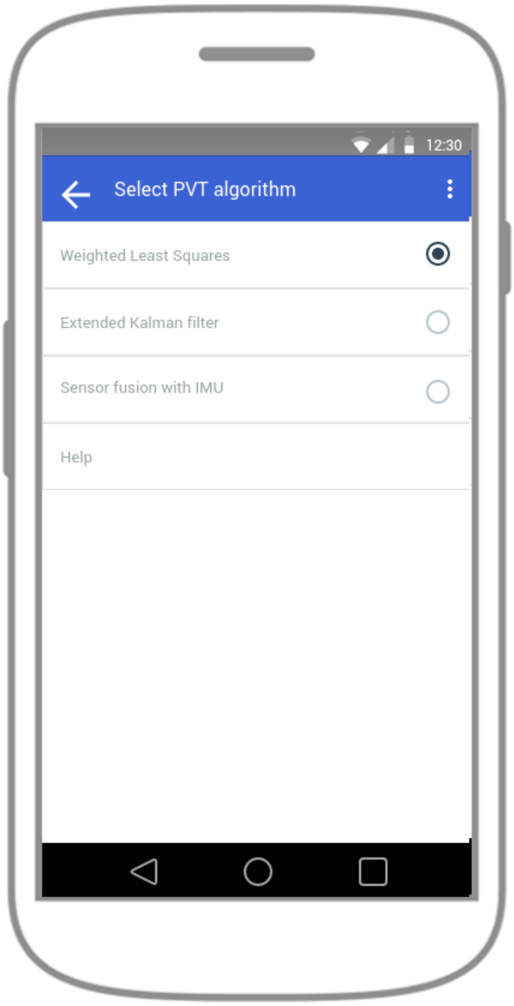
(a) Viewer settings (b) Constellations

Figure 3: Draft design of the viewer settings of the UI.

The help section will focus on presenting the ideas in a simple and understandable way, rather than providing mathematical expressions used in the algorithms. Its focus will be on helping the user to understand how various factors are taken into account and improve his or her knowledge about GNSS systems in general.



(a) Correction modules



(b) PVT algorithms

Figure 4: Draft design of the viewer settings of the UI.

5 DEVELOPMENT PLAN

For this project we propose a slightly modified agile methodology project management scheme [19]. In a standard agile software development scheme, the team starts with a list of features (called a backlog), which are prioritized and implemented one by one, starting from the most important. After a certain period of time, usually after two weeks, a new version of the developed software is released. This new version, extends the previous one, implementing some of the features from the backlog. These short periods are called sprints.

The key idea of agile is to work closely with the end-users or customers and include their feedback during development. Thanks to this iterative process, the customer can test the product in early stages of development, asking for modifications. This approach enforces modularity and keeping the software in good state during whole development process. Before the end of each sprint, the code has to be refactored and documented – because after the sprint ends, the software is released as a working product. Here, the term *released* means e.g. given for use to other team members or to the Galileo Services committee if needed; not necessarily given to the public.

We propose a following list of features, which are to be implemented (based on section 4.1):

1. Pseudorange computation module,
2. TGVF connector,
3. TGVF data parser,
4. Correction modules:
   (a) generic API for correction modules,
   (b) Ionospheric delay correction,
   (c) Tropospheric delay correction,
   (d) Satellite clock bias correction,
   (e) Relativistic clock correction for satellite clock,
   (f) Relativistic path range correction,
   (g) Sagnac correction,
   (h) Solid Tides correction,
5. PVT calculation modules:
   (a) generic API for PVT algorithms,
   (b) WLS + Undetermined WLS method via Altitude Hold,
   (c) EKF,
   (d) Fusion with the IMU,
6. UI:
   (a) menu and general application structure,
   (b) logging to file,
   (c) positioning accuracy viewer,
   (d) sky plot of the satellites viewer,
   (e) satellite signal strength viewer,
   (f) horizontal domain error viewer,
   (g) current position on map viewer,
7. Support for GNSS satellite constellations:
   (a) Galileo,

(b) GPS,
(c) GLONASS,
(d) BeiDou.

Worth noting is that this list is not in any particular way prioritized. We propose for each sprint, in our case, to last a month. A following list of features (or milestones) is predicted to be completed after each month:

sprint	feature description	feature IDs
1	app core running, interfaces and APIs defined	1, 4a, 5a, 6a
2	first measurements recorded, implementations of one PVT and correction module	2, 6b, 7a, 7b, 5b, 4b
3	measurements calculated and displayed by app, further implementation of correction modules	3, 4c, 4d, 6c, 6d, 6e
4	Further implementation of PVT and correction modules, finalization of the UI	5c, 4e, 4f, 6f, 6g
5	Final implementations of PVT and correction modules	4g, 4h
6	Implementations of support for other constellations	7c, 7d

The above list is just a proposition. It might, and probably will, change during development as we understand better user's requirements and the implementation issues.

Tests and validation activities will be performed when deemed necessary, after crucial development milestones.

5.1 Risks

Threats and opportunities in this project are mostly related to time. Main advantage of the design proposed in 4.1 is its modularity. Even if, due to time constraints, we do not manage to implement all of the features we have intended, the first product of our work is a framework, which can be used by researchers to easily test their algorithms, theories or to log experiment data from selected constellations. Also a non-scientific user will be able to use the application to test Galileo reception and signal quality.

During development it might become evident that some features are more complex to implement than initially predicted. In such cases, the feature list will be reevaluated and priorities might change. This is also one of the benefits of using agile methodology. The team remains flexible and can easily adapt to new findings, developments and feedback from experts. Workforce can be easily changed and adjusted, so that most effort is put to solve the most serious issues.

In order to avoid the situation in which the developed application does not fulfill the requirements or expectations of the organizational board, a monthly meeting scheme is proposed.

During such, rather short, monthly meetings, the progress could be presented, issues could be discussed, and a new version of the application could be released for testing and evaluation. Based on feedback from these meetings, the priorities might change and workflow might be adjusted.

5.2 Opportunities

If the motivation level is kept high and work is progressing very well, during the whole development time of the project, development might finish earlier than predicted. This will open new opportunities, such as a possibility to implement more PVT algorithms or correction methods than initially assumed. The plan presented in section 5 is just an outline of the work. During development new ideas might arise and prove to be better than originally proposed. In such situation the possibilities and risks will be reevaluated and, if needed, the feature list will be changed. For example the Galileo fusion with the IMU is representing an attractive feature for *GNSS compare*. The choice of the fusion technique (tight or loose fusion) will be made at the time when there is a clear decision regarding the implementation of this feature.

Given experience of two team members in robotics, we might drain inspiration from that field as well. One example is a method called monocular Simultaneous Localization and Mapping (SLAM) [20], which uses images from a camera to calculate motion and displacements. Although computationally heavy, it might be very useful when high accuracies are needed.

If the Galileo Services committee would require such a solution, we might consider adding a networking component. So that the application, after user's permission, could be transmitting raw measurements to a server. Such data could be later used by European Space Agency (ESA) experts, to analyze Galileo signal quality all around Europe, or further away, wherever the users will be located.

5.3 Validation Plan

The validation plan for the application's features and positioning performance is consisting of a *preliminary* and *final* phase. In each of the phases we will perform tests and analyses for static and dynamic use cases, utilizing the equipment from D/TEC's Radio-Navigation Laboratory.

5.3.1 Preliminary phase

This phase is planned to begin during the fourth month of development (Section 5). For this phase, the anechoic chamber and the SPIRENT GNSS signal simulator are planned to be used. Having the smartphone placed inside the chamber, test scenarios will be created (static and dynamic use cases), with the help of SPIRENT's software tool (SimGEN) with which different GNSS constellation signals can be simulated. The following objectives have been identified for such setup:

- Test the application's features by controlling the smartphone from a computer through a dedicated software tool (e.g., Vysor) while the simulated signals are received and processed.

- Analyze the behavior and the performance of the implemented PVT algorithms in simulated favorable conditions (e.g., unobstructed environment) and in unfavorable ones (e.g., urban canyons).
- Quantify the PVT algorithm performance for the different scenarios with statistical measures such as Root Mean Square Error (RMSE), Circular Error Probability (CEP) and the 95 percentile (R95).
- Design scenarios that will simulate firstly the current status of Galileo, GPS constellations and redo the performance analysis. Also, depending on the current status of the application's algorithms we consider defining scenarios that also include the other GNSS constellations.

5.3.2 Final phase

We predict the final phase of our validation plan to begin after the evaluation of the simulation-based results. In this phase we will move towards testing and assessing our application's features and performance in real scenarios:

- For the static use case we plan to quantify the PVT algorithms performance with respect to a highly accurate reference point across European Space Research and Technology Center (ESTEC) campus whose coordinates have been determined using one of the available professional GNSS receivers and high-precision positioning techniques (e.g., RTK). At this stage, we will ask for ESA's navigation experts support to fix such reference point if there is none available in the premises of ESTEC.
- To assess the application's performances in the dynamic use case we consider keeping close contact with ESA's navigation experts. Along with their advises we plan to define a reference trajectory with the help of the mobile test van as it is equipped with the highly-accurate NovAtel's Synchronous Position, Attitude and Navigation (SPAN) system.

During the entire validation plan, continuous adjustments and improvements of the application's features are expected to be carried on.

6 OUR TEAM

Our team

High diversity and diverse experience in a team is crucial to success when it comes to creative ideas and development. In our team we have experts from four different directorates of ESA — Mateusz is from the Directorate of Human Spaceflight and Robotic Exploration (D/HRE), Dominika is from the D/TEC, Sebastian is from the D/NAV and Mareike is from the Directorate of Earth Observation Programmes (D/EOP).

6.1 Team structure

Four main project roles have been identified:

1. Algorithm designer: responsible for implementing the PVT and correction algorithms,
2. Android developer: responsible for implementing the core framework on Android,
3. User experience designer: responsible for end user's interactions with the application, including design of the UI,
4. Product owner: responsible for defining priorities and keeping the team on correct track, as well as contacting key partners, scheduling of laboratory tests and other managerial tasks.

Out of those, the Product owner role is most subtle and algorithm designer is the most time-consuming. With his broad knowledge of the GNSS world, Sebastian will take the main role as the algorithm designer, and he will be supported by Mareike. Given her experience with software development and Android platform, Dominika will take the main role as the Android developer, supported by Mareike. Mateusz will take on main role as the User experience designer, supported by Dominika. Also, given his experience with multiple student projects, and management of projects led voluntarily, after hours, Mateusz will serve as the Product owner, supported by Sebastian. A table below presents a short summary of the roles and responsibilities in the project. It shows that all of the fields and people responsible for them overlap. We believe that this way we will achieve synergy, empower our strengths and diminish our weaknesses.

role \ person	Sebastian	Mareike	Dominika	Mateusz
Algorithm design	x	x		
Android developer		x	x	
User experience designer			x	x
Product owner	x			x

6.2 Team members

Below we present additional information about each of the team members.

Dominika Perz

Polish Young Graduate Trainee in the Technology, Engineering and Quality Directorate at ESTEC, ESA. Currently working as a Project Manager for the Lunar Exploration Mission - internal project to determine a preliminary GNC (Guidance, Navigation and Control) design for the ascent, rendezvous and docking with the Deep Space Gateway station orbiting a Moon in the highly elliptical orbit. Her background is mainly in robotics and programming. She completed Control Engineering and Robotics master studies in Poland, during which she spent one semester in Madrid, Spain as an Erasmus exchange student. As a first international carrier experience, Dominika did a 6 weeks internship in R&D team at Venderlande (Eindhoven, Netherlands), where she worked on optimisation of the search algorithm. During holidays in 2016 she participated in the Aerospace Information Technology Summer School in Würzburg, Germany. Before coming to ESA, Dominika worked for a year at a software company GlobalLogic as a Junior Software Developer for embedded systems.

Sebastian Ciuban

Romanian Young Graduate Trainee in the Directorate of Navigation at the ESTEC. He holds a Master of Science degree in Aerospace Systems – Navigation and Telecommunications granted by the French Civil Aviation University (ENAC) from Toulouse, France. He developed his master thesis at the German Aerospace Center (DLR) in Oberpfaffenhofen. While being at DLR he was responsible with designing and implementing algorithms that fused the Precise Point Positioning (PPP) technique with Inertial Navigation Systems (INS) in a tight coupling setting. Moreover, he has also developed suitable integrity monitoring algorithms in order to measure the reliability of the designed fused systems. His research interests are related to precise positioning, sensor fusion, integrity monitoring and GNSS receiver signal processing.

Mareike Burba

German National Trainee with EOP-SMS, currently working on the atmospheric correction for the Fluorescence Explorer (FLEX). She holds a M.Sc. in Meteorology with a minor in Scientific Computing from the University of Hamburg. Her Master's thesis was about optimizing the numerical computation of atmospheric radiative transfer for the Infrared Atmospheric Sounding Interferometer (IASI) on MetOp. Mareike previously joined the World Climate Research Program Joint Planning Staff in the World Meteorological Organization for a 6 months internship in the framework of the Carlo Schmid program. Thanks to her studies and jobs as student research assistent, she speaks fluently Python, Matlab, Bash and Fortran, is communicative in C, Java and IDL, and has some experience with parallel computing.

Mateusz Krainski

Polish Young Graduate Trainee in the Directorate of Human Spaceflight and Robotic Exploration, at ESTEC, where he supports the European Robotic Arm (ERA) team. His main duties regard designing, developing and validating a robotic testbed for testing of ERA's on-board smart cameras. During studies, Mateusz was one of the key board members of a robotic student society, where he managed numerous projects ranging from small teams for quick projects (this includes a Space Startup Weekend, an Android app hackathon and few during-studies assignments), organizing robotic tournaments (with a team of over 15 people), up to technical projects counting over 30 people. Thanks to the Toastmasters International community, Mateusz has developed highly his public speaking skills. He not only helped start the first English speaking club in the area, but also received awards in presenting competitions on a semi-national level.

7 SUMMARY

In this document, a possible solution for the Galileo App Challenge objectives has been presented. The solution consists of two major parts – for the general public user, we propose an application to view simultaneously two different results, from two different processing schemes. For an advanced user, we propose a generalized framework, which will allow them to test their algorithms and compare results with general methods.

We have a strong and diverse team, consisting of both navigation and software development experts. We believe that with the help of ESA experts, we have all of the necessary skills to develop, test and validate the proposed framework.

References

[1] Google. Material design. https://material.io/guidelines/material-design/introduction.html, 2017.

[2] Openstreetmap - wiki. http://wiki.openstreetmap.org/wiki/Main_Page, 2017.

[3] J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares, R. Prieto-Cerdeira, and R. Orús Pérez. Nequick ionospheric model. http://www.navipedia.net/index.php/NeQuick_Ionospheric_Model, 2017.

[4] Ionospheric correction algorithm for galileo single frequency users. https://www.gsc-europa.eu/system/files/galileo_documents/Galileo_Ionospheric_Model.pdf, September 2016.

[5] J. Sanz Subirana, J.M. Juan Zornoza, and M. Hernández-Pajares. Tropospheric delay. http://www.navipedia.net/index.php/Tropospheric_Delay, 2011.

[6] *Review of tropospheric, ionospheric and multipath data and models for Global Navigation Satellite Systems*, Berlin, Germany, 2009. IEEE.

[7] Tropospheric delay and gnss signals: Tutorial. http://gnss.be/troposphere_tutorial.php, Feburary 2012.

[8] J.Sanz Subirana, J.M. Juan Zornoza and M.Hernandez-Pajares. *European Space Agency GNSS DATA PROCESSING, Volume I: Fundamentals and Algorithms*. 2013.

[9] J. Sanz Subirana, J.M. Juan Zornoza, and M. Hernández-Pajares. Relativistic clock correction. http://www.navipedia.net/index.php/Relativistic_Clock_Correction, 2011.

[10] Neil Ashby. Relativistic effects in the global positioning system. page 4, July 2006.

[11] Haitao Yin, Jie Li, Pifeng Ma, Shipeng Zhang, and Duan Xu. Effect of solid-earth-tide on gps time series. *Geodesy and Geodynamics*, 1(1):64 – 69, 2010.

[12] Salvatore Gaglione. *Inside GNSS: How does a GNSS receiver estimate velocity?* 2015.

[13] John W. Betz. *Course 541-1706: Advanced Receiver Processing of GNSS Signals held at ESTEC, ESA*. October 2017.

[14] John W. Betz. *Engineering Satellite-Based Navigation and Timing: Global Navigation Satellite Systems, Signals and Receivers*. 2016.

[15] Kai Borre. *The GPS Easy Suite*.

[16] Mark Petovello. *Inside GNSS: GNSS Filtering Options*. 2013.

[17] T. Takasu and other contributors. *RTKlib*.

[18] Paul D.Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. 2008.

[19] Agile methodology. http://agilemethodology.org/, 2017.

[20] Monocular slam. http://www.doc.ic.ac.uk/~ab9515/monoslam.html, 2017.