

# OpenHPS: A Modular Framework to Facilitate the Development of FAIR Positioning Systems

Maxim Van de Wynckel<sup>1</sup> and Beat Signer<sup>1</sup>

<sup>1</sup> Vrije Universiteit Brussel, Web & Information Systems Engineering Lab, Brussels, Belgium

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Open Journals](#)

## Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Abstract

Positioning systems determine the location of people and objects using various technologies and algorithms. While GPS dominates outdoor positioning, indoor and smaller-scale systems often require alternative technologies for improved latency, accuracy, or efficiency. These systems are often developed as single-use prototypes with no standard data format, hindering reusability and expansion. OpenHPS addresses these challenges by providing a modular, graph-based framework for creating versatile positioning systems on multiple platforms. It supports a wide range of algorithms and enables extensibility through custom nodes for sensor fusion and algorithm integration.

## Keywords

Hybrid positioning systems, interoperable positioning systems, indoor positioning, stream processing, linked data, RDF

## Statement of Need

Existing positioning systems or frameworks (Georgiou et al., 2015; Pustka et al., 2011; Scholl, 2020) often focus on specific use cases, platforms, or a set of algorithms. While this facilitates the deployment of a system, it limits the flexibility to design or reuse custom algorithms and systems. In systems developed for academic purposes (García et al., 2015), proprietary systems are often difficult to replicate, extend, or deploy in a production-ready environment due to the use of software such as MatLab. Furthermore, the data produced by such systems is often unstandardised. OpenHPS was created to address these issues by providing a modular framework that allows developers and researchers to create positioning systems for a wide range of use cases. The framework and data created with our framework is designed to be interoperable, allowing developers to share their algorithms and data with others. Using additional modules that add support for Solid (Van de Wynckel & Signer, 2022a) and DHTs, we have enabled the development of findable, accessible, interoperable and reusable (FAIR) positioning systems.

## Framework Overview

OpenHPS is an open-source hybrid positioning system framework written in TypeScript. It runs on the server (Node.js), in the browser, or within hybrid mobile applications. The general design of a positioning system created using OpenHPS consists of a graph with a set of nodes that process data. Background services can be added to the graph to persist data or to communicate with other systems. Our decision to use TypeScript compared to using a low-level language was based on the goal of achieving cross-platform reusability.

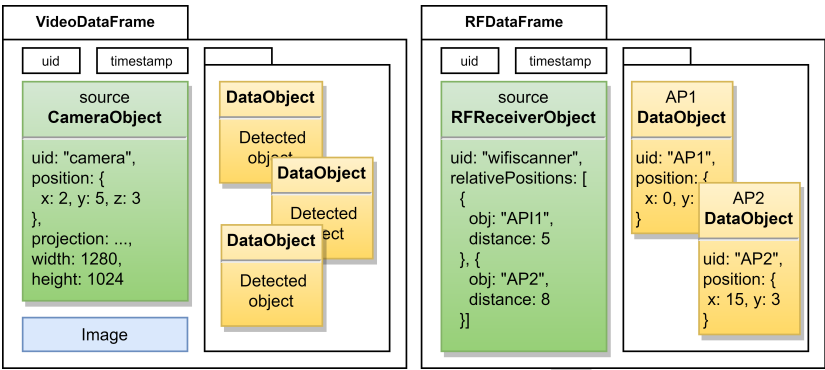


Figure 1: DataFrames and DataObjects

38 All concepts, ranging from positions to sensor values, can be expressed in various ways with  
39 varying units, enabling OpenHPS to support both small-scale use cases such as tracking a pen  
40 on a paper to larger use cases such as tracking airplanes across the globe. Our framework uses  
41 stream-based processing of DataFrames which contain all time-sensitive information. These  
42 frames contain one or more DataObjects as illustrated in Figure 1, which indicate the spatial  
43 objects relevant for the information within the data frame. This enables the tracking of multiple  
44 actors, unlike frameworks such as ROS (Quigley et al., 2009).

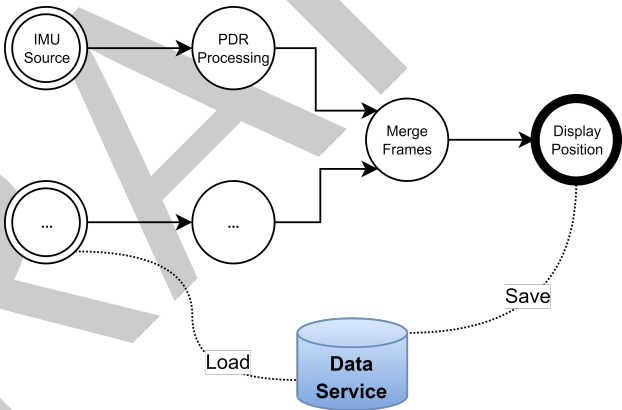


Figure 2: OpenHPS graph of a positioning system

45 Any data pushed through the graph can be serialised to JSON or RDF data using the POSO  
46 ontology (Van de Wynckel & Signer, 2022b), enabling interoperability between systems. Each  
47 node in the graph represents a step in the processing of data frames from source to sink  
48 as shown in Figure 2. A SourceNode generates information, a ProcessingNode processes  
49 information and a SinkNode consumes information.

50 OpenHPS is modular by design, mainly due to the ability to extend data frames and objects.  
51 These extensions allow for different data objects, including different sensors like cameras, IMU  
52 sensors or spatial landmarks such as Bluetooth beacons. Each node in a graph can be extended  
53 as well, allowing the creation of custom algorithms that can be added or removed from a  
54 positioning system. Researchers can focus on the prototyping of new algorithms without having  
55 to worry about the integration of these algorithms into a larger system.

56 **Interoperability**

57 Our framework is designed to offer FAIR positioning systems, leveraging its open architecture  
58 and the data it produces. Modules such as @openhps/rdf and @openhps/solid, we have

59 enabled developers to serialise their data to RDF and store it in Solid Pods. This facilitates the  
60 creation of positioning systems that are transparent and privacy-preserving, while also ensuring  
61 that other positioning systems or consumer applications can access the data, regardless of  
62 whether they were created with OpenHPS. With various extensions such as semantic beacons  
63 (Van de Wynckel & Signer, 2023), we enable the discovery of these systems and the data they  
64 produce.

## 65 Performance

66 OpenHPS uses JavaScript at runtime. To overcome the challenges associated with creating  
67 real-time dataflows, our graph can be executed using Web Workers (Van de Wynckel &  
68 Signer, 2020). All data transmitted through our graph is serialisable, eliminating the need  
69 for developers to handle this serialisation or communication between workers. In addition to  
70 the ability to run graphs on multiple workers, communication nodes, such as MQTT, enable  
71 developers to offload the processing of complex tasks to other servers or dedicated processors.  
72 For more high-demanding algorithms, modules like @openhps/opencv and @openhps/openvslam  
73 provide C++ bindings to low-level libraries. While these bindings may not achieve the same  
74 performance, the ability to reuse other libraries is part of the main requirements.

## 75 Examples of Research Work

76 OpenHPS has been a building block for various research that has been used in indoor positioning  
77 systems (Van de Wynckel & Signer, 2021a), and its ability to serialise location data to RDF  
78 has been demonstrated in an application that aims to preserve privacy and transparency using  
79 Solid Pods (Van de Wynckel & Signer, 2022a). Further, OpenHPS was used in the SemBeacon  
80 demonstrator application (Van de Wynckel & Signer, 2023) that is written in CapacitorJS  
81 and uses the framework to deserialise positioning data and positioning systems. Modules have  
82 been created for other academic projects, such as the FidMark ontology (Van de Wynckel  
83 et al., 2024), which provides fiducial marker classification within the framework. OpenHPS's  
84 modular node design and domain focus facilitate the sharing of algorithms and findings, as  
85 well as the rapid creation of prototypes and demonstrators that make use of location data.  
86 Finally, OpenHPS and the related modules have been used to collect several datasets that can  
87 be used by researchers to evaluate their algorithms (Van de Wynckel, 2025; Van de Wynckel  
88 & Signer, 2021b, 2025).

## 89 References

- 90 García, E., Poudereux, P., Hernández, Á., Ureña, J., & Gualda, D. (2015). A Robust UWB  
91 Indoor Positioning System for Highly Complex Environments. *2015 IEEE International  
92 Conference on Industrial Technology (ICIT)*, 3386–3391. [https://doi.org/10.1109/ICIT.  
93 2015.7125601](https://doi.org/10.1109/ICIT.2015.7125601)
- 94 Georgiou, K., Constambeys, T., Laoudias, C., Petrou, L., Chatzimilioudis, G., & Zeinalipour-  
95 Yazti, D. (2015). Anyplace: A Crowdsourced Indoor Information Service. *Proceedings  
96 of the International Conference on Mobile Data Management: Systems, Services and  
97 Middleware (MDM 2015)*, 291–294. <https://doi.org/10.1109/MDM.2015.80>
- 98 Pustka, D., Huber, M., Waechter, C., Echtler, F., Keitler, P., Newman, J., Schmalstieg, D., &  
99 Klinker, G. (2011). Automatic Configuration of Pervasive Sensor Networks for Augmented  
100 Reality. *IEEE Pervasive Computing*, 10(3), 68–79. <https://doi.org/10.1109/MPRV.2010.50>
- 101 Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng,  
102 A. Y. (2009, May). ROS: An Open-Source Robot Operating System. *Proceedings of  
103 the International Workshop on Open Source Software*. [https://api.semanticscholar.org/  
104 CorpusID:6324125](https://api.semanticscholar.org/CorpusID:6324125)

- 105 Scholl, Z. (2020). *The Framework for Internal Navigation and Discovery (FIND)*. <https://github.com/schollz/find3>
- 106
- 107 Van de Wynckel, M. (2025). *Garage Positioning Dataset*. Kaggle. [https://doi.org/10.34740/](https://doi.org/10.34740/KAGGLE/DS/6654647)
- 108 [KAGGLE/DS/6654647](https://doi.org/10.34740/KAGGLE/DS/6654647)
- 109 Van de Wynckel, M., & Signer, B. (2022a). A Solid-based Architecture for Decentralised
- 110 Interoperable Location Data. *Proceedings of the 12th International Conference on Indoor*
- 111 *Positioning and Indoor Navigation (IPIN 2022)*, CEUR Workshop Proceedings, 3248.
- 112 <https://ceur-ws.org/Vol-3248/paper11.pdf>
- 113 Van de Wynckel, M., & Signer, B. (2021a). Indoor Positioning Using the OpenHPS Frame-
- 114 work. *Proceedings of the 11th International Conference on Indoor Positioning and Indoor*
- 115 *Navigation (IPIN 2021)*, 1–8. <https://doi.org/10.1109/IPIN51156.2021.9662569>
- 116 Van de Wynckel, M., & Signer, B. (2020). *OpenHPS: An Open Source Hybrid Positioning*
- 117 *System*. Vrije Universiteit Brussel. <https://doi.org/10.48550/ARXIV.2101.05198>
- 118 Van de Wynckel, M., & Signer, B. (2021b). *OpenHPS: Single Floor Fingerprinting and*
- 119 *Trajectory Dataset*. <https://doi.org/10.5281/zenodo.4744380>
- 120 Van de Wynckel, M., & Signer, B. (2022b). POSO: A Generic Positioning System Ontology.
- 121 *Proceedings of the 21st International Semantic Web Conference (ISWC 2022)*. [https://doi.org/10.1007/978-3-031-19433-7\\_14](https://doi.org/10.1007/978-3-031-19433-7_14)
- 122
- 123 Van de Wynckel, M., & Signer, B. (2023). SemBeacon: A Semantic Proximity Beacon Solution
- 124 for Discovering and Detecting the Position of Physical Things. *Proceedings of the 13th*
- 125 *International Conference on the Internet of Things (IoT 2023)*. [https://doi.org/10.1145/](https://doi.org/10.1145/3627050.3627060)
- 126 [3627050.3627060](https://doi.org/10.1145/3627050.3627060)
- 127 Van de Wynckel, M., & Signer, B. (2025). *Sphero Dead Reckoning and CV Tracking Dataset*.
- 128 Kaggle. <https://doi.org/10.34740/KAGGLE/DS/6760212>
- 129 Van de Wynckel, M., Valadez, I., & Signer, B. (2024). FidMark: A Fiducial Marker Ontology
- 130 for Semantically Describing Visual Markers. *Proceedings of the Semantic Web (ESWC*
- 131 *2024)*, 235–250. [https://doi.org/10.1007/978-3-031-60635-9\\_14](https://doi.org/10.1007/978-3-031-60635-9_14)