



## The CAMS eSense Framework

Enabling Earable Computing for mHealth Apps and Digital Phenotyping

**Bardram, Jakob Eyvind**

*Published in:*

Proceedings of the 1st International Workshop on Earable Computing

*Link to article, DOI:*

[10.1145/3345615.3361137](https://doi.org/10.1145/3345615.3361137)

*Publication date:*

2019

*Document Version*

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*

Bardram, J. E. (2019). The CAMS eSense Framework: Enabling Earable Computing for mHealth Apps and Digital Phenotyping. In *Proceedings of the 1st International Workshop on Earable Computing* (pp. 3-7). Association for Computing Machinery. Proceedings of the 1st International Workshop on Earable Computing <https://doi.org/10.1145/3345615.3361137>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# The CAMS eSense Framework: Enabling Earable Computing for mHealth Apps and Digital Phenotyping

Jakob E. Bardram

[jakba@dtu.dk](mailto:jakba@dtu.dk)

Copenhagen Center for Health Technology  
Department of Health Technology, Technical University of Denmark  
Copenhagen, Denmark

## ABSTRACT

Earable computing devices can be an important platform for mobile health (mHealth) applications and digital phenotyping, since they allow for collection of detailed sensory data while also providing a platform for contextual delivery of interventions. In this paper we describe how the eSense earable computing platform has been integrated with a programming framework and runtime platform for the design of mHealth applications. The paper details how this programming framework can be used in the design of custom mHealth technologies. It also provides data and insight from an initial study in which this framework was used to collect real-life contextual data, including sensory data from the eSense device.

## CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**; • **Software and its engineering** → **Development frameworks and environments**; • **Applied computing** → Health informatics.

## KEYWORDS

digital phenotyping, mobile health, mobile sensing, eSense, earable computing

### ACM Reference Format:

Jakob E. Bardram. 2019. The CAMS eSense Framework: Enabling Earable Computing for mHealth Apps and Digital Phenotyping. In *Proceedings of 1st International Workshop on Earable Computing (EarComp'19)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3345615.3361137>

## 1 INTRODUCTION

A significant body of research has been applying mobile sensing to health and wellness applications [3] including, for example, the EmotionSense [11], BeWell [10], and StudentLife [15] systems, which classify physical activity, sleep, and social interaction based on sensor data. Similarly, studies in mental health have demonstrated correlations and predictive power between phone-based features on physical activity, mobility, social activity, phone usage,

and voice data on the one side, and mental health symptoms in e.g., depression [14], bipolar disorder [5, 7], and schizophrenia [4] on the other side.

More generally, it has been argued that obtaining a more precise understanding of a disease can happen in multiple dimensions, and one new dimension is the use of mobile devices to measure people's activity and other factors more continuously and accurately. As such, mobile sensing has been defined as central to the 'Precision Medicine' initiative; genotypic information can become more powerful if phenotypic information is also available [1]. The use of everyday mobile and wearable technology for collection of behavioral, psychological, and health data has been termed 'digital phenotyping' [8, 13], which can be defined as;

*continuous and unobtrusive measurement and inference of health, behavior, and other parameters from wearable and mobile technology.*

Earable computing devices provide a novel and significant technological platform for the design of mobile health (mHealth) technologies and digital phenotyping. First of all because earable computing enables new sensor modalities and the collection of a new type of data, including head movement (from accelerometers and gyroscopes), sound and noise levels as experienced by the user (and not by the phone which might be in a pocket), as well as more health and well-being features related to e.g. cardio-vascular activity (pulse, heart rate (HR), and heart rate variability (HRV)), sleep detection, etc. Secondly because earable computing might be a platform for delivering Just-in-Time Adaptive Interventions (JITAI) [12], by coupling contextual sensing with the delivery of a personalized and private intervention using the headset speakers. Hence, an mHealth intervention no longer needs to rely on notifications on the user's phone screen (with all the problems of notification fatigue associated with this), but instead can be delivered as small audio messages targeted for the specific person only. Thirdly, if the earable device has some input modalities – like a push button – simple Ecological Momentary Assessment (EMA) sampling can be done.

The eSense device from Nokia Bell Labs provide such an earable computing platform [9]. In combination with a more general data sampling platform, this device can provide important additional sensing modalities for digital phenotyping as well as in the design of mHealth applications. In this paper we describe how the eSense technology has been integrated into a larger runtime platform and programming framework for digital phenotyping and mHealth application development, which then allows researchers and mHealth application designers to achieve the visions for earable computing, as outlined above. The paper also reports from a small study in

---

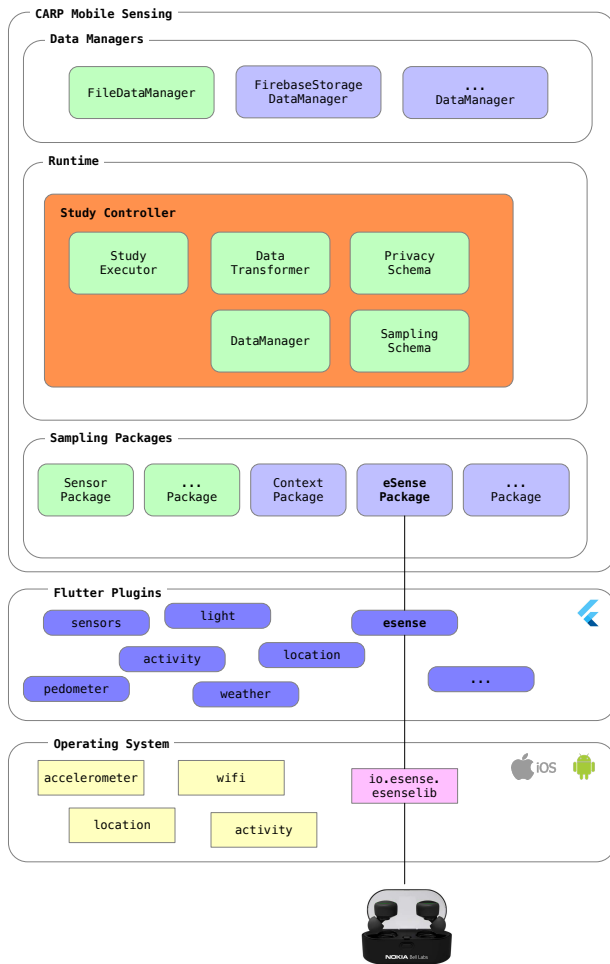
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*EarComp'19, September 9, 2019, London, United Kingdom*

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6902-2/19/09...\$15.00

<https://doi.org/10.1145/3345615.3361137>



**Figure 1: A simplified view on the CAMS architecture highlighting how the eSense device is integrated into the framework using both a Flutter plugin and a CAMS sampling package.**

which the eSense device was used in data collection and where earable sensor data is combined with ‘traditional’ mobile sensing modalities, like location, connectivity, communication patterns, activity recognition, noise, etc. The earable computing programming framework presented in this paper is open source and we hope that others in the earable community can benefit from using this.

## 2 CAMS ESENSE FRAMEWORK

The eSense technology has been integrated into the CARP Mobile Sensing (CAMS) framework [2]<sup>1</sup>. CAMS have been described elsewhere and there are plenty of online resources in terms of application programming interface (API) documentation and tutorial available – see Appendix A.2. CAMS is a cross-platform (iOS & Android) programming framework for building mHealth technology that incorporates mobile and wearable sensing. It is designed

<sup>1</sup>CARP is an abbreviation of the CACHET Research Platform.

to be highly extensible allowing for; (i) adding new data sampling modalities (such as wearable devices like the eSense device), (ii) implementing different kinds of data transformation (e.g. transforming sensor data to standardized formats or on-phone pre-processing before upload), (iii) using data sampling actively in app design, and (iii) supporting different data off-loading strategies (such as local file storage or cloud-based upload of data). CAMS is implemented in Flutter, which is Google’s cross-platform portable toolkit for building natively-compiled applications for mobile, web, and desktop from a single codebase [6]. Flutter rely on Dart, which is a modern object-oriented, reactive programming language optimized for non-blocking user-interface programming with a mature and complete async-await event-driven code style, paired with isolate-based concurrency.

As illustrated in Figure 1, CAMS has three main layers; (i) a runtime layer (in the middle), (ii) a set of data managers (top layer), and (iii) a set of sampling packages (bottom layer). CAMS is a very flexible and extensible ‘plug-and-play’ architecture, in which most of the components show in Figure 1 can be tailored, extended, or replaced, and customized components can be added. For example, a new sampling package can be added, which supports sampling of data from a new source – both from on-board phone sources (such as a phone sensor or log) or off-board wearable sensors which can be accessed e.g. via Bluetooth Low Energy (BTLE). The integration of eSense into CAMS is an example of the latter, where an eSense sampling package have been implemented, which then can be linked and used in app development. The implementation of a sampling package in CAMS rely on access to one or more Flutter plugins for data access. These Flutter plugins are strictly speaking not a part of CAMS, but are a generic way to access the phone’s operating system (OS) in a cross-platform manner. A Flutter plugin is often implemented using the ‘Platform Channel’ technology in Dart/Flutter, which allow Flutter to access the native OS API on both Android and iOS<sup>2</sup>.

Hence, in order to support the eSense device in CAMS, we have implemented two components; (i) a Flutter plugin which uses a platform channel to access the eSense Java API, and (ii) an eSense sampling package which integrates support for eSense into CAMS.

### 2.1 The eSense Flutter Plugin

The eSense Flutter plugin is designed to be used directly in a Flutter app and is hence an independent library. It has been designed to resemble the Android eSense API almost 1:1 and the eSense Android programmer will be able to recognize the names of the different classes, methods, and class variables. For example, the methods on the ESenseManager class is mapped 1:1. However, one major design change has been done; the eSense Flutter plugin follows the Dart/Flutter reactive programming architecture using *streams*. Hence, you do not ‘add listeners’ to an eSense device (as you do in Java) – rather, you obtain a Dart stream and listen to this stream, and utilize all the other very nice stream operations that are available in Dart – including creating very beautiful reactive user interfaces (UIs). Listing 1 shows the basic Dart code on how to use the eSense plugin. As can be seen, it quite straight-forward and

<sup>2</sup>More information on how to write platform-specific code in Flutter is available at <https://flutter.dev/docs/development/platform-integration/platform-channels>.

it only requires a few lines of code to use the plugin. The reader familiar with the eSense Java API will recognize the way to use the API and its names.

```
1 import 'package:esense_flutter/esense.dart';
2
3 // listen to connection events before connecting
4 ESenseManager.connectionEvents.listen((event)
5   => print('CONNECTION event: $event'));
6
7 // try to connect to the eSense device with a given name
8 success = await ESenseManager.connect('eSense-0332');
9
10 // listen to sensor events and print them
11 ESenseManager.sensorEvents.listen((event)
12   => print('SENSOR event: $event'));
```

**Listing 1: Using the eSense Flutter plugin.**

Note that playing and recording audio are performed via the Bluetooth Classic interface and are not supported by the eSense plugin. However, as we shall present below, CAMS supports sampling of audio and noise, which is done via the eSense device microphone, once connected.

The eSense Flutter plugin has been released to the Flutter package sharing site<sup>3</sup> using the name `esense_flutter` including documentation on the API and how to use the plugin<sup>4</sup>. Links to online resources are provided in Appendix A.1. The eSense Flutter plugin has implemented support for both Android – using the Nokia Bell Labs implementation – as well as for iOS – using the eSense iOS Library from the University of Tokyo. This means that all eSense apps implemented using Flutter and CAMS will run on both Android and iOS without any platform-specific development needed. This is the true strength of using Flutter.

## 2.2 The eSense Sampling Package

A CAMS sampling package basically consists of three components; (i) a Measure which defines *what data to collect*, (ii) a Probe that implements *how data is collected*, and (iii) a Datum object which specify the *data format* of the collected data. Two types of data can be collected from the eSense device; (a) button pressed / released events and (b) sensor events from the device’s inertial measurement unit (IMU) (accelerometer and gyroscope). CAMS supports both these type of measures, using two different probes, and stores it in two different datum objects. Hence, the two types of measures are independent and the app developer can choose to use one and/or the other.

Listing 2 shows how the eSense measures are configured to be part of a CAMS study. A Study object is created with a name and a file storage as the data endpoint (line 1–3), and then a list of measures are added to a task and a trigger, which basically just starts the sampling immediately and runs forever (line 4–15). The study is configured to sample eSense button events, eSense sensor events, noise, location, activities, local weather information, and scans for Bluetooth devices in the phone’s proximity. The exact configuration of these measures is defined in a so-called `SamplingSchema`, where the common schema is used in this case.

```
1 study = Study('1234', 'user@dtu.dk')
2 ..name = 'CARP Mobile Sensing - eSense sampling demo'
3 ..dataEndPoint = getDataEndpoint(DataEndPointTypes.FILE)
4 ..addTriggerTask(
5   ImmediateTrigger(),
6   Task()
7   ..measures = SamplingSchema.common().getMeasureList([
8     ESenseSamplingPackage.ESENSE_BUTTON,
9     ESenseSamplingPackage.ESENSE_SENSOR,
10    AudioSamplingPackage.NOISE,
11    ContextSamplingPackage.LOCATION,
12    ContextSamplingPackage.ACTIVITY,
13    ContextSamplingPackage.WEATHER,
14    ConnectivitySamplingPackage.BLUETOOTH,
15  ]));
```

**Listing 2: Using the eSense measure types as part of a CAMS study.**

Once the study is defined, it can be handed over to the CAMS `StudyController` as shown in Listing 3. This paper do not allow for going into the details of CAMS, but the code examples hopefully illustrates that a sampling study can be configured and executed quite easily.

```
1 // Create a controller for this study, initialize it, and start it
2 controller = StudyController(study);
3 await controller.initialize();
4 controller.start();
5
6 // listening on all data events from the study and print it
7 controller.events.forEach(print);
```

**Listing 3: Starting a study.**

## 3 ESENSE STUDY

In order to evaluate the eSense Flutter plugin and the eSense CAMS sampling package, we created the study listed in Listing 2 and deployed it in the CAMS client app. This study configuration invokes a set of corresponding probes, which samples the specified data types. The CAMS client app with this list of probes is shown in Figure 2. The following measures were configured:

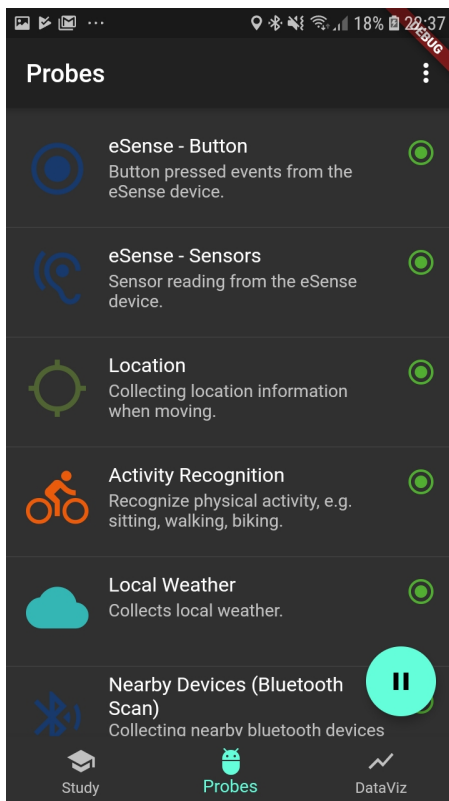
- eSense button events
- eSense sensor events with a sampling rate of 10 Hz
- Ambient noise, sampled over a 5 seconds window every 45 seconds.
- Location triggered by the phone on movement.
- Activity as recognized by the activity recognition API on the phone.
- Local weather as collected from the WeatherAPI service.
- Scanning of nearby Bluetooth devices every 60 seconds.

The main goal of this sampling schema is to sample data in a scenario where a user is physically active during a day, performing different activities (e.g. biking, walking, sitting), at different locations, with different weather, with different noise levels, and with different people (the Bluetooth scan). The eSense sensor data is collected to see if this can be correlated or used in classification of activities. The user is instructed to press the eSense button when starting a new activity and/or changing context.

We ran the study for one day which included activities of walking, driving, sitting at a desk, and biking. During the study we collected more than 150,000 data points (61 MB data), of which the vast majority was the detailed sensor data from the eSense IMU sensor. All of this data was stored locally on the phone (hence,

<sup>3</sup><http://pub.dev>

<sup>4</sup>[https://pub.dev/packages/esense\\_flutter](https://pub.dev/packages/esense_flutter)



**Figure 2: The study configured in Listing 2 shown in the list of probes in the CAMS mobile sensing app.**

offloading to e.g. Firebase was not used in this study). The app and the eSense device ran continuously during the entire day (8 hours) without any significant breakdowns and problems. However, valuable experience in handling the eSense device was obtained. For example, there might be inference between the eSense Bluetooth connectivity while doing the Bluetooth scan in CAMS and the BTLE connection to the eSense IMU could break, if the earplug was used for streaming music. Hence, this study gave some input in how to make the data sampling package and the eSense probes more robust in handling disconnection and re-connection scenarios. At the time of writing the collected data has not been analysed. But the study demonstrated the feasibility of using CAMS with the eSense sampling package in such digital phenotyping studies. The data is available for download – see Appendix A.3 for details.

## 4 CONCLUSION

This paper has presented the integration of the eSense earable computing platform from Nokia Bell Labs into the CARP Mobile Sensing (CAMS) framework. This integration allows software developers of mHealth apps to include mobile and wearable sensing to their app design, which now also include the eSense device. Moreover, the framework allows for using eSense in digital phenotyping, as demonstrated by a small study. Now that this infrastructure is in place, we plan to set up more studies where the eSense earable

computing technology can be used in combination with all the other sampling measures available in CAMS.

## ACKNOWLEDGMENTS

This work has been funded by the Copenhagen Center for Health Technology (CACHET) [www.cachet.dk].

## REFERENCES

- [1] Euan A. Ashley. 2015. The precision medicine initiative: A new national effort. *JAMA* 313, 21 (2015), 2119–2120.
- [2] Jakob E. Bardram. 2019. The CARP Mobile Sensing Framework: A Cross-platform, Reactive, Programming Framework and Micro-service Runtime Environment for Digital Phenotyping. *In Submission* (2019).
- [3] Jakob E Bardram and Mads Frost. 2016. The Personal Health Technology Design Space. *IEEE Pervasive Computing* 15, 2 (2016), 70–78.
- [4] Ian Barnett, John Torous, Patrick Staples, Luis Sandoval, Matcheri Keshavan, and Jukka-Pekka Onnela. 2018. Relapse prediction in schizophrenia through digital phenotyping: a pilot study. *Neuropsychopharmacology* (2018), 1.
- [5] Maria Faurholt-Jepsen, Jonas Busk, Mads Frost, Maj Vinberg, Ellen M Christensen, Ole Winther, Jakob E Bardram, and Lars V Kessing. 2016. Voice analysis as an objective state marker in bipolar disorder. *Translational psychiatry* 6, 7 (2016), e856.
- [6] Flutter 2019. Flutter – Google’s portable UI toolkit for building beautiful, natively-compiled applications for mobile, web, and desktop from a single codebase. Retrieved July 12, 2019 from <https://flutter.dev>
- [7] Agnes Grünerbl, Amir Muaremi, Venet Osmani, Gernot Bahle, Stefan Oehler, Gerhard Tröster, Oscar Mayora, Christian Haring, and Paul Lukowicz. 2014. Smartphone-based recognition of states and state changes in bipolar disorder patients. *IEEE Journal of Biomedical and Health Informatics* 19, 1 (2014), 140–148.
- [8] Sachin H Jain, Brian W Powers, Jared B Hawkins, and John S Brownstein. 2015. The digital phenotype. *Nat Biotech* 33, 5 (may 2015), 462–463.
- [9] Fahim Kawsar, Chulhong Min, Akhil Mathur, and Alessandro Montanari. 2018. Earables for Personal-Scale Behavior Analytics. *IEEE Pervasive Computing* 17, 3 (2018), 83–89.
- [10] Nicholas D Lane, Mashfiqui Mohammad, Mu Lin, Xiaochao Yang, Hong Lu, Shahid Ali, Afsaneh Doryab, Ethan Berke, Tanzeem Choudhury, and Andrew Campbell. 2011. Bewell: A smartphone application to monitor, model and promote wellbeing. *In 5th international ICST conference on pervasive computing technologies for healthcare*. 23–26.
- [11] Neal Lathia, Veljko Pejovic, Kiran K Rachuri, Cecilia Mascolo, Mirco Musolesi, and Peter J Rentfrow. 2013. Smartphones for Large-Scale Behavior Change Interventions. *IEEE Pervasive Computing* 12, 3 (2013), 66–73.
- [12] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. 2017. Just-in-Time Adaptive Interventions (JITAs) in Mobile Health: Key Components and Design Principles for Ongoing Health Behavior Support. *Annals of Behavioral Medicine* 52, 6 (12 2017), 446–462.
- [13] Jukka-Pekka Onnela and Scott L Rauch. 2016. Harnessing Smartphone-Based Digital Phenotyping to Enhance Behavioral and Mental Health. *Neuropsychopharmacology* 41, 7 (2016).
- [14] Sohrab Saeb, Emily G Lattie, Stephen M Schueller, Konrad P Kording, and David C Mohr. 2016. The relationship between mobile phone location sensor data and depressive symptom severity. *PeerJ* 4 (2016), e2537.
- [15] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. 2014. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. *In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 3–14.

## A ONLINE RESOURCES

This appendix provides link to different online resources relevant for the eSense Flutter plugin, CAMS, and the CAMS eSense sampling packages, as well as the data from the small study reported in this paper.

### A.1 eSense Flutter Plugin

- The esense Plugin at pub.dev – [https://pub.dev/packages/esense\\_flutter](https://pub.dev/packages/esense_flutter)
- The esense API documentation – [https://pub.dev/documentation/esense\\_flutter/latest/](https://pub.dev/documentation/esense_flutter/latest/)
- The esense Plugin GitHub – [https://github.com/cph-cachet/flutter-plugins/tree/master/packages/esense\\_flutter](https://github.com/cph-cachet/flutter-plugins/tree/master/packages/esense_flutter)

### A.2 CAMS Framework and Documentation

- The CARP Mobile Sensing (CAMS) core Flutter Plugin at pub.dev – [https://pub.dev/packages/carp\\_mobile\\_sensing](https://pub.dev/packages/carp_mobile_sensing)
- The CAMS tutorials and documentation – <https://github.com/cph-cachet/carp.sensing-flutter/wiki>
- The CAMS API documentation – [https://pub.dev/documentation/carp\\_mobile\\_sensing/latest/](https://pub.dev/documentation/carp_mobile_sensing/latest/)
- The CAMS eSense Sampling Package at pub.dev – [https://pub.dev/packages/carp\\_esense\\_package](https://pub.dev/packages/carp_esense_package)
- The CAMS GitHub – <https://github.com/cph-cachet/carp.sensing-flutter>

### A.3 Data from the eSense Study

The data and description from this small (N=1) study can be accessed from:

- <https://github.com/cph-cachet/data/tree/master/2019.08.28.eSense>