# BUILDING MOBILE & WEARABLE HEALTH TECHNOLOGY IN FLUTTER

The CARP Team @DTU

# Today's Program

**OVERVIEW OF THE CARP PROJECT**

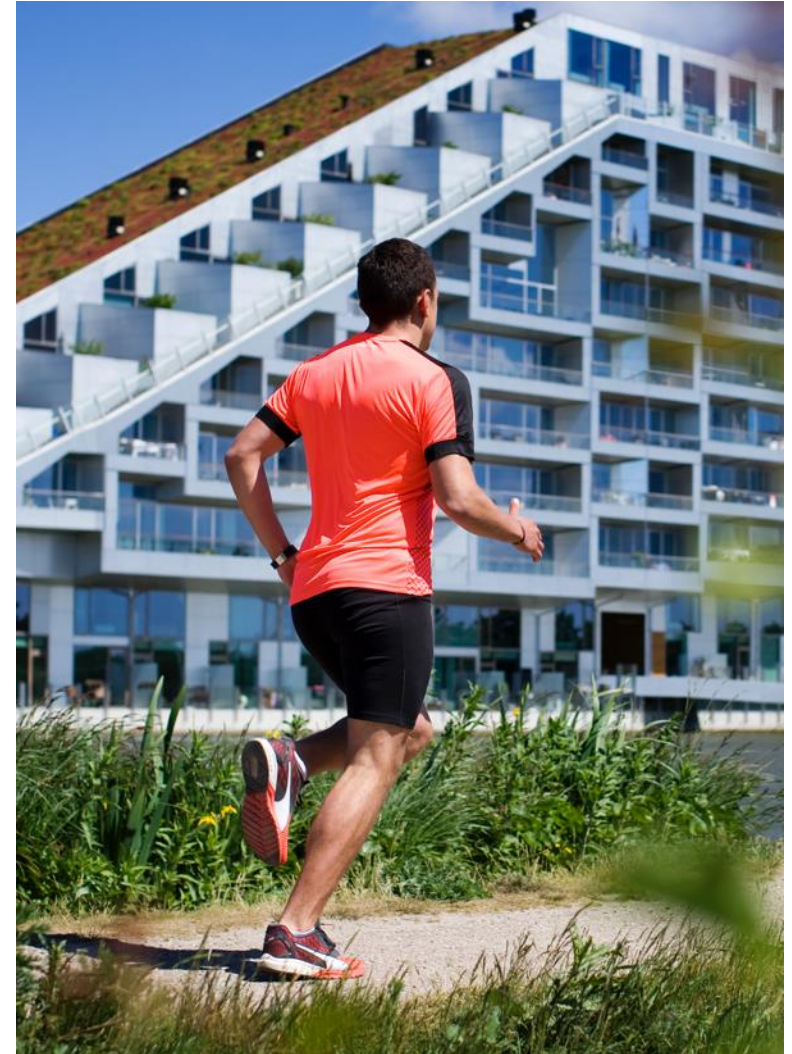- Mobile and Wearable Technology for Health
- Architecture & Packages

**EXAMPLES**

- Neurology (Neuropathy Tracker)
- `health` and other CARP plugins
- Cardiology and ECG recordings
- Digital Phenotyping (Studies App)

**DEMOS & BREAK**

**CLOSING**

- Documentation & Resources
- Looking ahead

# THE COPENHAGEN RESEARCH PLATFORM (CARP)

# Copenhagen Research Platform – CARP

**Large-scale data science platform for digital phenotyping and mobile health technology**

- **Open-source programming framework**
  - components & frameworks for the design of mHealth applications
  - used to create disease-specific solutions
  - developed and shared with research & industry partners (open source)

- "Out-of-the-Box" Study Hosting
  - CARP instance hosted @DTU Computerome (HPC)
  - GDPR compliant for Danish researchers
  - configurable study setup
  - large-scale analysis of data across studies
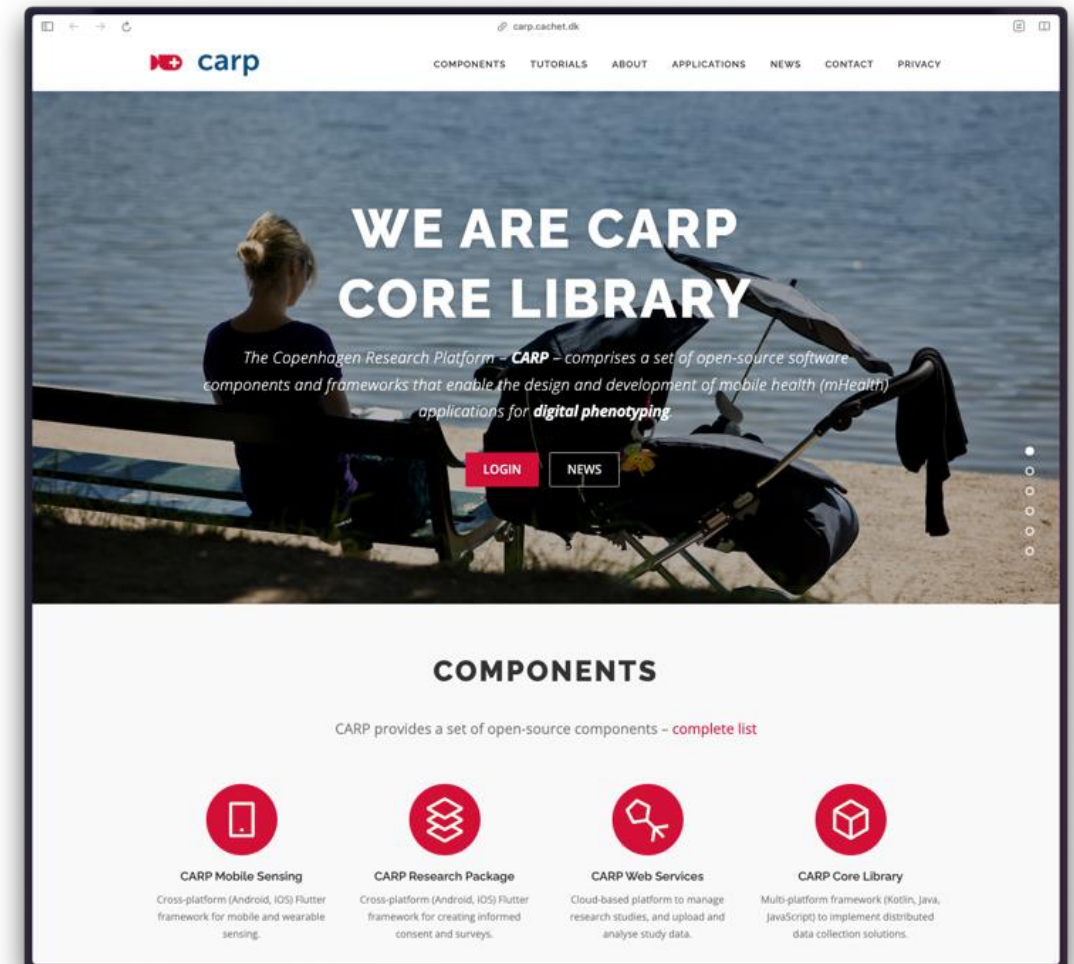


**carp.dk**

# CARP-based mHealth Applications

## mCardia
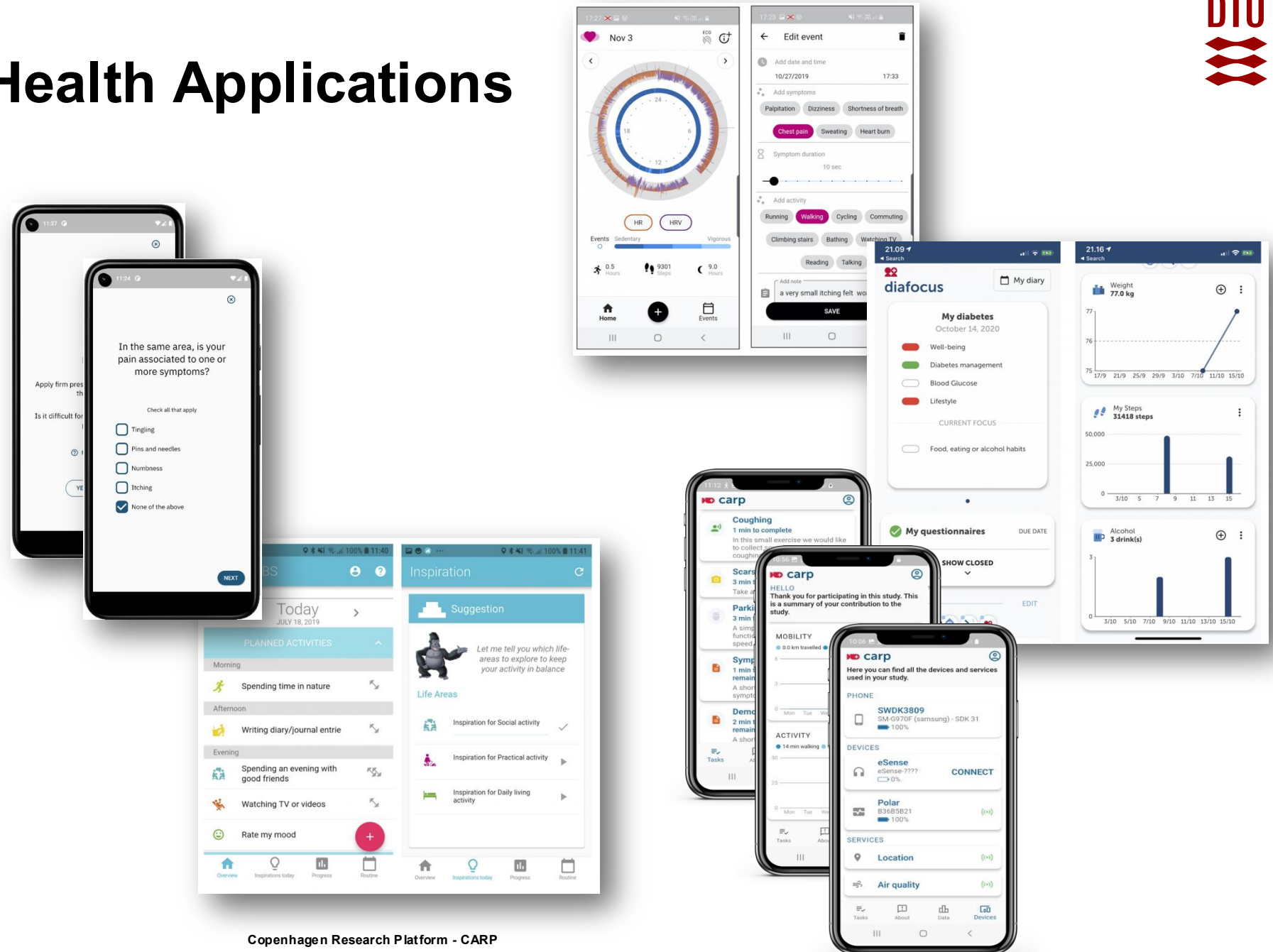- Cardiology

## DiaFocus
- Type 2 Diabetes

## MUBS
- Mental Health

## Neuropathy Tracker
- Neurology

## CARP Studies
- Digital Phenotyping

# CARP Flutter Components

**CARP Core**

- domain model and standards

**CARP Mobile Sensing**

- mobile sensing framework
- wearable devices

**CARP Research Package**

- informed consent
- surveys

**CARP Cognition Package**

- cognitive test framework
- 14 pre-made tests

# Data Collection in CARP Mobile Sensing

**Physiological**
- weight, height, ...
- ECG, HR, HRV, blood pressure...
- Blood glucose

**Behavioral**
- physical activity (steps, movement, ...)
- social activity (communication, calendar, messaging, ...)
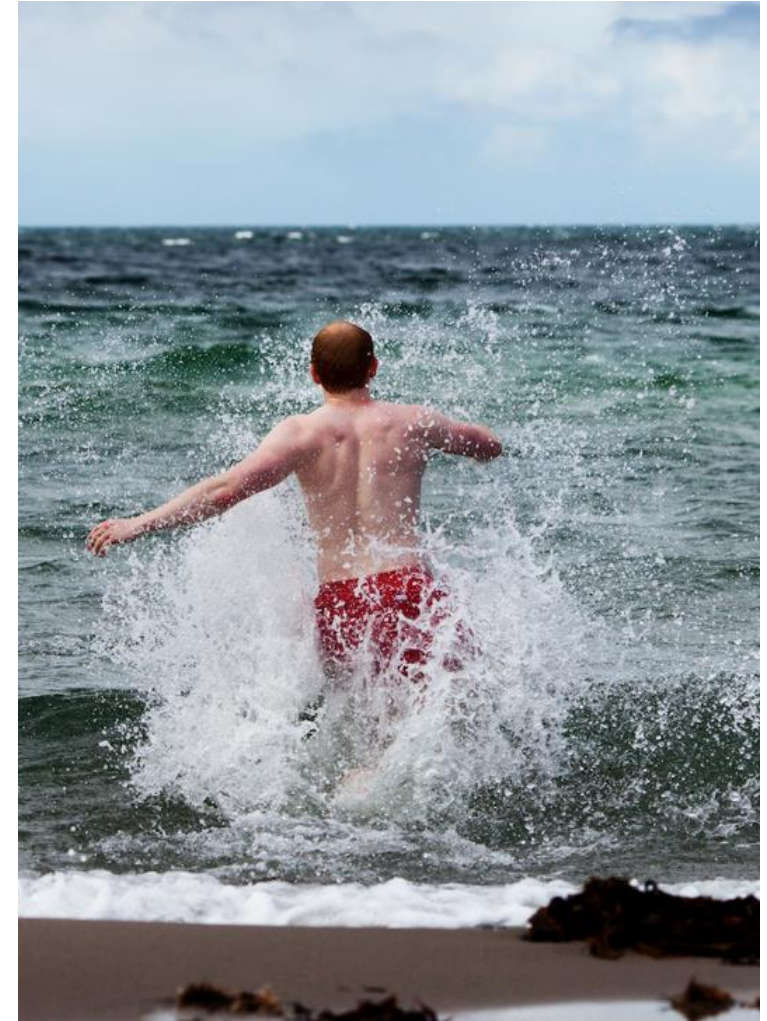- phone usage (screen, connectivity, ...)

**Contextual**
- location (geo-position, geofence, ...)
- weather, air quality

**Patient-Reported**
- surveys
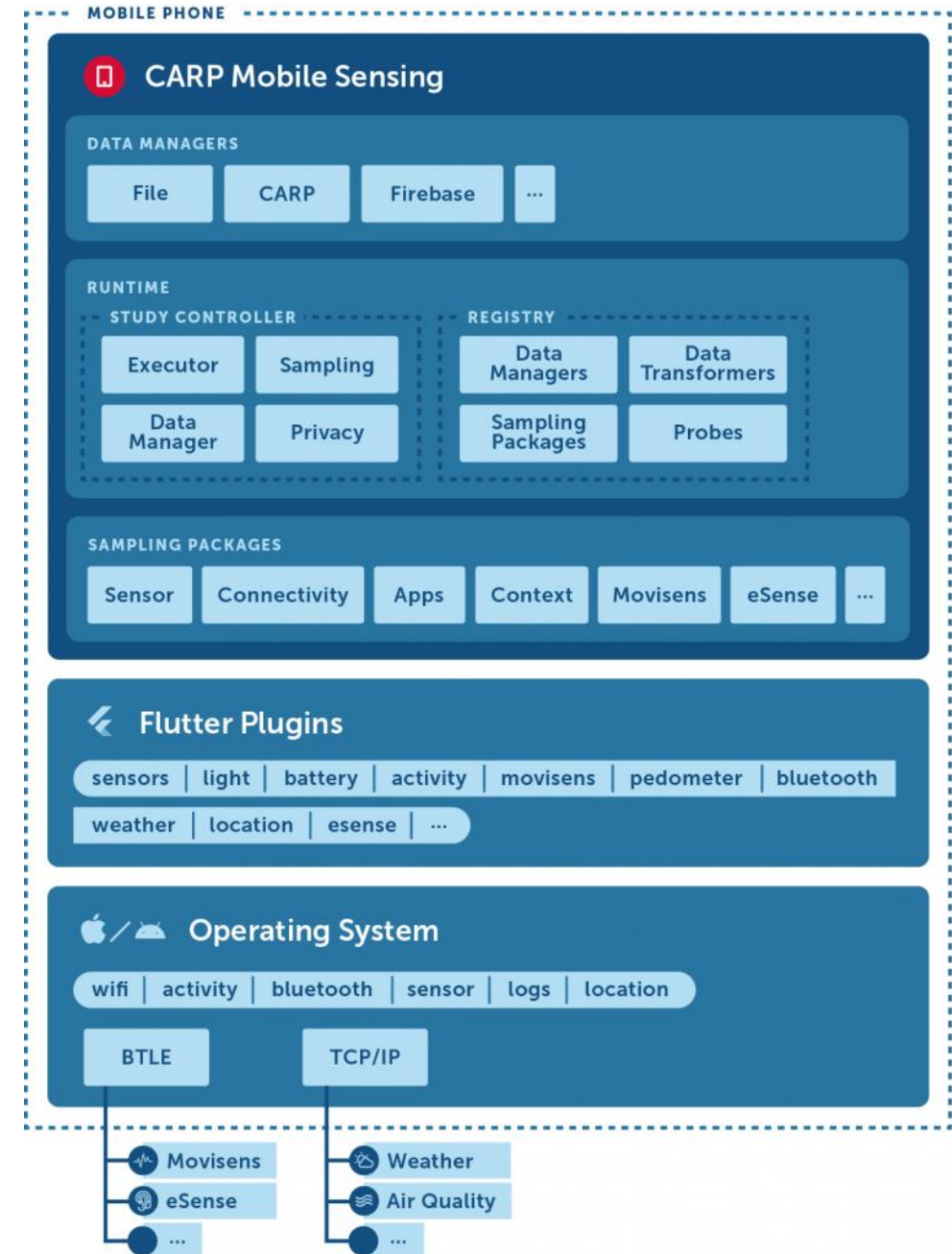- ecological momentary assessments (EMA)
- audio & video

**Cognition**
- 8 Neurocognitive domains
- 14 validated gold-standard cognitive tests

# Software Architecture

- Layered
  - CARP Mobile Sensing
  - Sampling Packages
  - Flutter Plugins
  - OS / Web Services

- Extensible
  - Sampling Package
  - Plug'n Play
  - Configuration | Data Model | Probe | Device Mgmt

# Cross-platform Mobile Sensing

| Type | Android | iOS | Package | Description |
|------|---------|-----|---------|-------------|
| accelerometer | + | + | sensors | Accelerometer data from the built-in phone sensor |
| gyroscope | + | + | sensors | Gyroscope data from the built-in phone sensor |
| pedometer | + | + | sensors | Step counts from the device on-board sensor |
| light | + | - | sensors | Ambient light from the phone's front light sensor |
| device | + | + | device | Basic device information |
| battery | + | + | device | Battery charging status and battery level |
| screen | + | - | device | Screen event (on/off/unlock) |
| memory | + | - | device | Free memory |
| connectivity | + | + | connectivity | Connectivity status |
| bluetooth | + | + | connectivity | Scanning nearby bluetooth devices |
| wifi | + | + | connectivity | SSID and BSSID from connected wifi networks |
| location | + | + | context | Request the location of the phone. |
| geolocation | + | + | context | Listens to location changes. |
| activity | + | + | context | Activity as recognized by OS |
| weather | + | + | context | Current weather and weather forecasting |
| air_quality | + | + | context | Local air quality from land-based air pollution stations |
| geofence | + | + | context | Entry/dwell/exit events in circular geofences |
| audio | + | + | audio | Records audio from the device microphone |
| noise | + | + | audio | Detects ambient noise from the device microphone. |
| phone_log | + | - | communication | Log of phone calls in/out |
| text_message_log | + | - | communication | Log of text messages (sms) in/out |
| text_message | + | - | communication | Text message (sms) events when received |
| calendar | + | + | communication | All calendar events from all calendars on the phone |
| apps | + | - | apps | List of installed apps |
| app_usage | + | - | apps | App usage over time |
| survey | + | + | survey | User surveys via the Flutter `research_package` |
| movisens | + | - | movisens | ECG-related data from the Movisens EcgMove4 device. |
| esense | + | + | esense | Sensor and button events from eSense devices. |
| health | + | + | health | Wearable device data from Apple Health / Google Fit. |

- Bardram, J. E. (2020). The CARP Mobile Sensing Framework--A Cross-platform, Reactive, Programming Framework and Runtime Environment for Digital Phenotyping. *arXiv preprint arXiv:2006.11904*
- Bardram, J. E. (2022). Software Architecture Patterns for Extending Sensing Capabilities and Data Formatting in Mobile Sensing. *Sensors, 22*(7), 2813

# Wearable Devices

- Movisens Move4 (activity)
- Movisens EcgMove4 (activity, ECG)
- Nokia Bell Labs eSense (noise, movement)
- Polar Sense & H10 (HR, ECG, PPG, movement)
- Empatica E4 (HR, GSR, activity)
- Dexcom (CGM)

- Apple Health
- Google Health Connect

- Garmin (activity, sleep, HR, ...)
- Fitbit (activity, sleep, HR, BP, ECG, weight,
- Withings (activity, sleep, HR, BP, ECG, weight, ...)

# CARP Flutter Plugins

- **health**

- **noise_meter**

- **weather**

- **air_quality**

- **mobility_features**

- **pedometer**

## CARP Flutter plugins

This repo contains the source code for Flutter first-party plugins developed by developers at the Copenhagen Research Platform (CARP) at The Technical University of Denmark. Check the `packages` directory for all plugins.

Flutter plugins enable access to platform-specific APIs using a platform channel. For more information about plugins and how to use them, see https://flutter.io/platform-plugins/.

### Plugins

These are the available plugins in this repository.

| Plugin | Description | Android | iOS | http://pub.dev/ |
|---|---|:---:|:---:|---|
| screen_state | Track screen state changes | ✓ | ✓ | pub v4.1.1 |
| light | Track light sensor readings | ✓ | ✗ | pub v4.1.0 |
| pedometer | Track step count | ✓ | ✓ | pub v4.1.1 |
| noise_meter | Read noise level in Decibel | ✓ | ✓ | pub v5.0.2 |
| app_usage | Track usage of all applications on phone. | ✓ | ✗ | pub v4.0.1 |
| weather | Get current weather, as well as forecasting using the OpenWeatherMap API. | ✓ | ✓ | pub v3.2.1 |
| air_quality | Get the air quality index using the WAQI API. | ✓ | ✓ | pub v4.1.1 |
| notifications | Track device notifications. | ✓ | ✗ | pub v3.1.0 |
| movisens_flutter | Movisens sensor communication. | ✓ | ✓ | pub v3.2.0 |
| esense_flutter | eSense ear sensor plugin. | ✓ | ✓ | pub v1.2.0 |
| health | Apple HealthKit and Google Fit interface plugin. | ✓ | ✓ | pub v12.2.0 |
| activity_recognition | Activity Recognition | ✓ | ✓ | pub v5.0.0 |
| audio_streamer | Stream audio as PCM from mic | ✓ | ✓ | pub v4.2.0 |
| mobility_features | Compute daily mobility features from location data | ✓ | ✓ | pub v6.0.0 |
| carp_background_location | Track location, even when app is in the background | ✓ | ✓ | pub v4.0.0 |
| flutter_foreground_service | Foreground service for Android | ✓ | ✗ | pub v0.4.1 |

# Research Package

- A Flutter implementation of Apple Research Kit

- Informed Consent Flow

- Patient-Reported Data
  - Surveys
  - Questionnaires
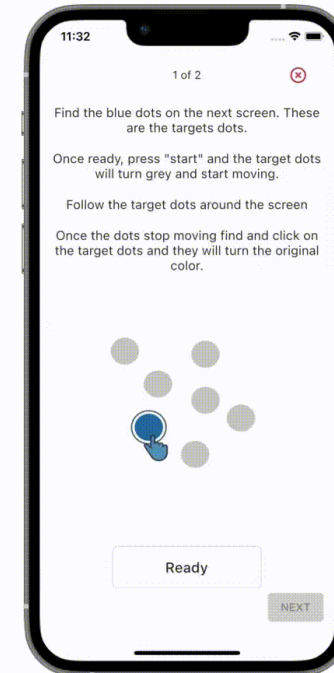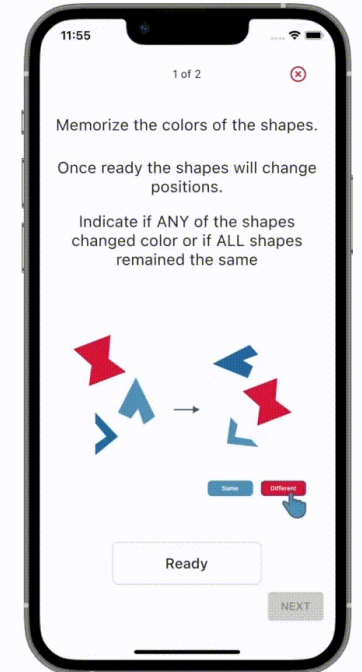  - Ecological Momentary Assessment (EMA)
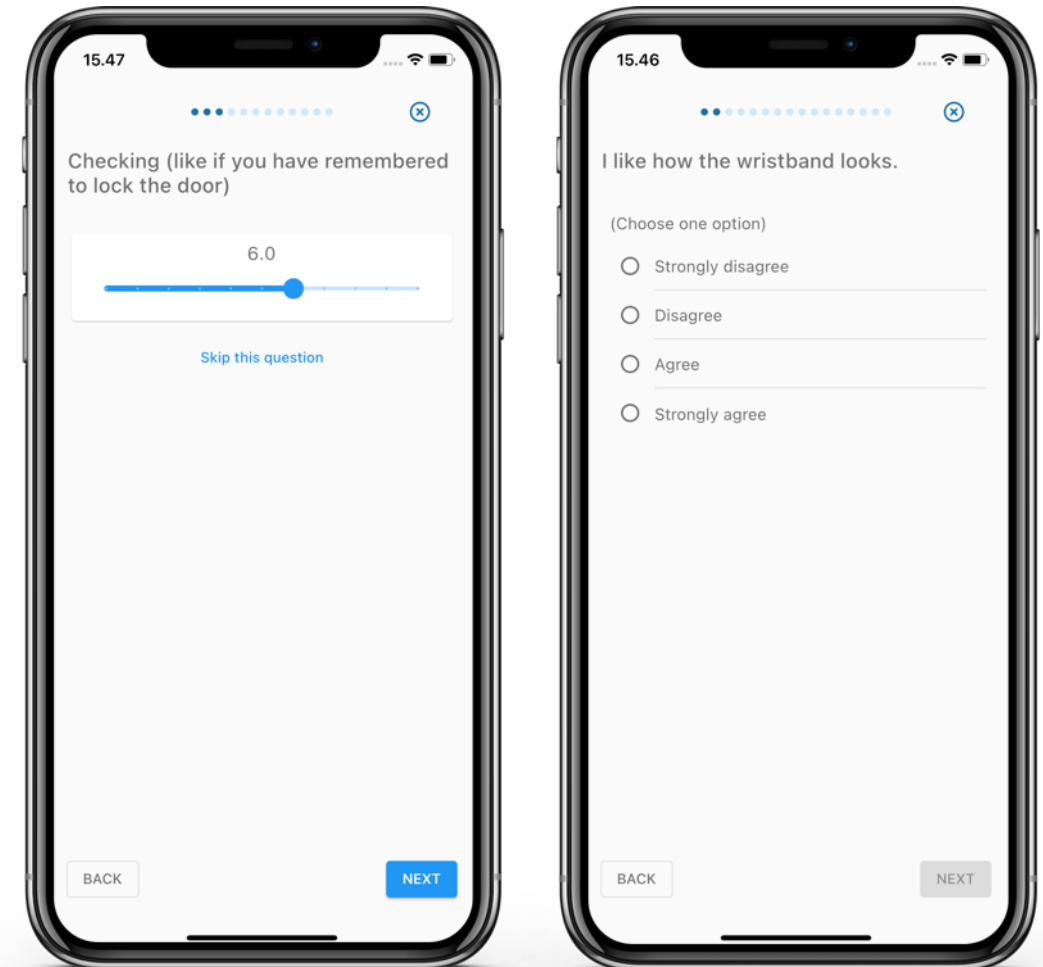
# Cognition Package

- 14 validated gold-standard cognitive tests

- 8 Neurocognitive domains
  - Sensation
  - Perception
  - Motor skills and construction
  - Attention and concentration
  - Memory
  - Executive functioning
  - Processing speed
  - Language and verbal skills
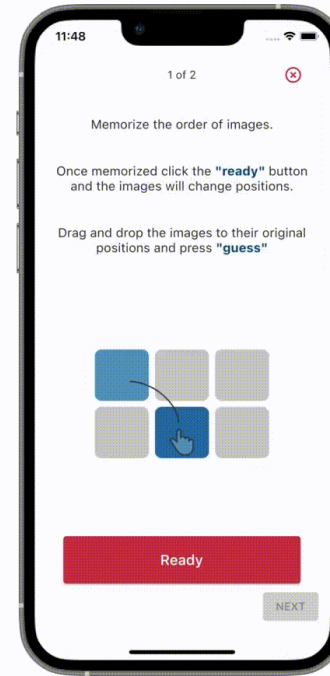


Picture Sequence Memory



Multiple Object Tracking



Visual Array Change

# health 12.2.0

Published 20 days ago • ✓ cachet.dk ( Dart 3 compatible )

SDK | FLUTTER    PLATFORM | ANDROID | IOS                            👍 597

Readme    Changelog    Example    Installing    Versions    Scores    Admin    Activity log

# Health

Enables reading and writing health data from/to Apple Health and Google Health Connect.

> NOTE: Google has deprecated the Google Fit API. According to the documentation, as of **May 1st 2024** developers cannot sign up for using the API. As such, this package has removed support for Google Fit as of version 11.0.0 and users are urged to upgrade as soon as possible.

The plugin supports:

- handling permissions to access health data using the `hasPermissions`, `requestAuthorization`, `revokePermissions` methods.
- reading health data using the `getHealthDataFromTypes` method.
- writing health data using the `writeHealthData` method.
- writing workouts using the `writeWorkout` method.
- writing meals on iOS (Apple Health) & Android using the `writeMeal` method.
- writing audiograms on iOS using the `writeAudiogram` method.
- writing blood pressure data using the `writeBloodPressure` method.
- accessing total step counts using the `getTotalStepsInInterval` method.
- cleaning up duplicate data points via the `removeDuplicates` method.
- removing data of a given type in a selected period of time using the `delete` method.

Note that for Android, the target phone **needs** to have the Health Connect app installed (which is currently in beta) and have access to the internet.

See the tables below for supported health and workout data types.

**Publisher**

✓ cachet.dk

**Weekly Downloads**

2024.06.23 — 2025.05.18

**Metadata**

Wrapper for Apple's HealthKit on iOS and Google's Health Connect on Android.

Repository (GitHub)

**Documentation**

API reference

**License**

⚖ MIT (license)

**Dependencies**

carp_serializable,
device_info_plus, flutter,

# Today's Program

**OVERVIEW OF THE CARP PROJECT**

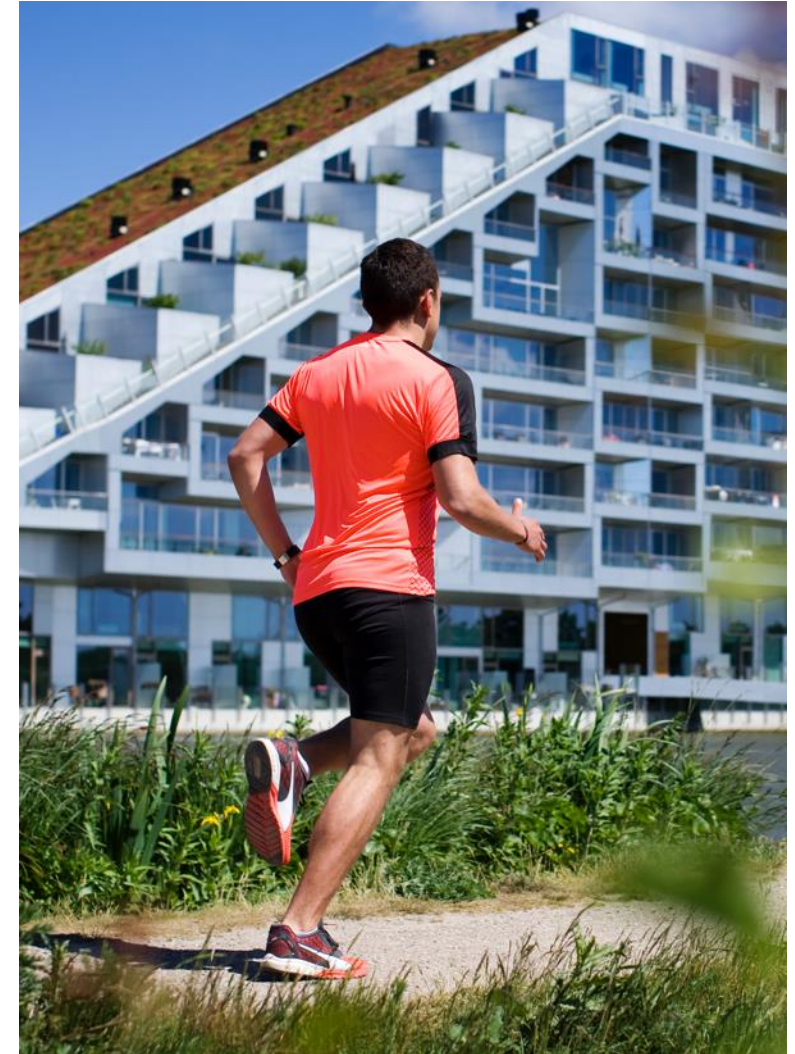- Mobile and Wearable Technology for Health
- Architecture & Packages

**EXAMPLES**

- Neurology (Neuropathy Tracker) [Mads]
- `health` and other CARP plugins [Alireza]
- Cardiology and ECG recordings [Henrik]
- Digital Phenotyping (Studies App) [Panos]

**DEMOS & BREAK**

**CLOSING**

- Documentation & Resources
- Looking ahead

**EXAMPLES**

# CARP in the Neuropathy Tracker

**Surveys containing steps like**

- Informed consent
- Signatures
- Instructions
- Single/Multipe-choice
- Date and Time
- Slider
- Many others...
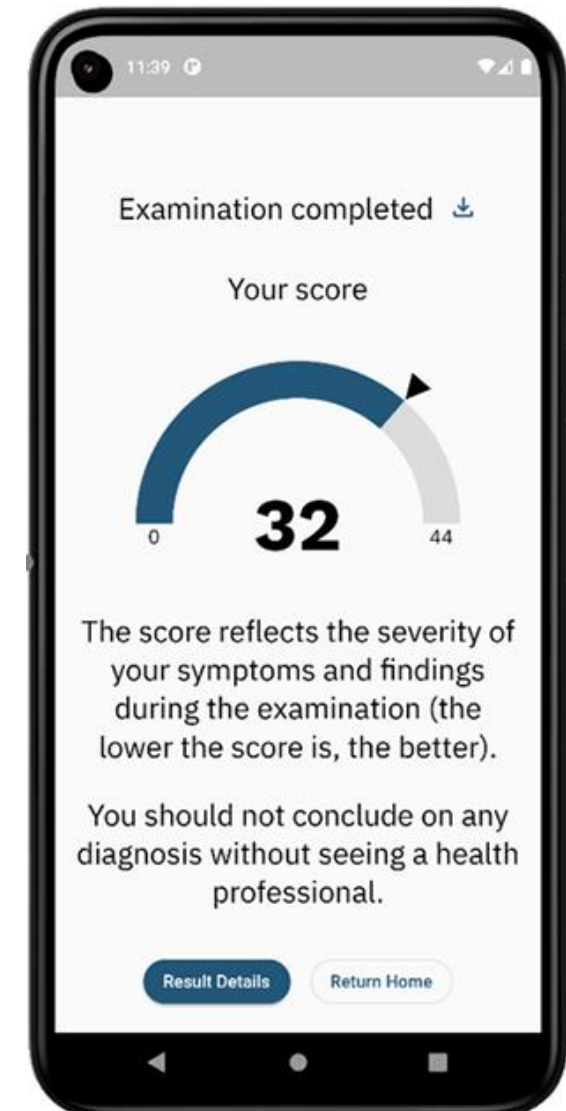
# Localization

One JSON per language

- `da.json`

```json
{
    "yes": "Ja",
    "no": "Nej"
}
```

- `en.json`
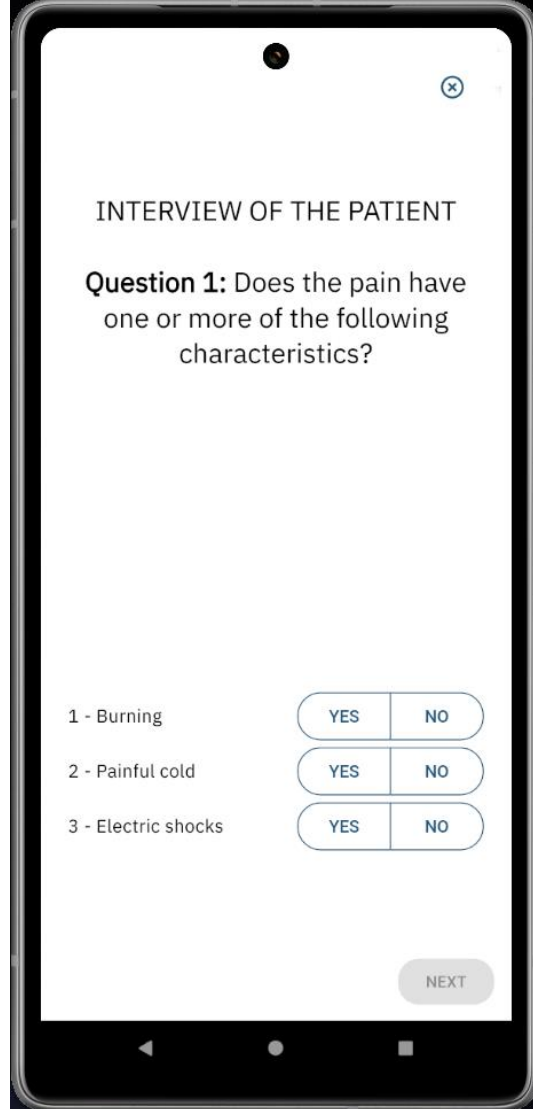
```json
{
    "yes": "Yes",
    "no": "No"
}
```

- `de.json`

```json
{
    "yes": "Ja",
    "no": "Nein"
}
```



INTERVIEW MED PATIENTEN

**Spørgsmål 1:** Har smerten et eller flere af følgende kendetegn?

| | | |
|---|---|---|
| 1 - Brændende | JA | NEJ |
| 2 - Smertefuld kold | JA | NEJ |
| 3 - Elektriske stød | JA | NEJ |

NÆSTE



INTERVIEW OF THE PATIENT

**Question 1:** Does the pain have one or more of the following characteristics?

| | | |
|---|---|---|
| 1 - Burning | YES | NO |
| 2 - Painful cold | YES | NO |
| 3 - Electric shocks | YES | NO |

NEXT

# Extending RP steps
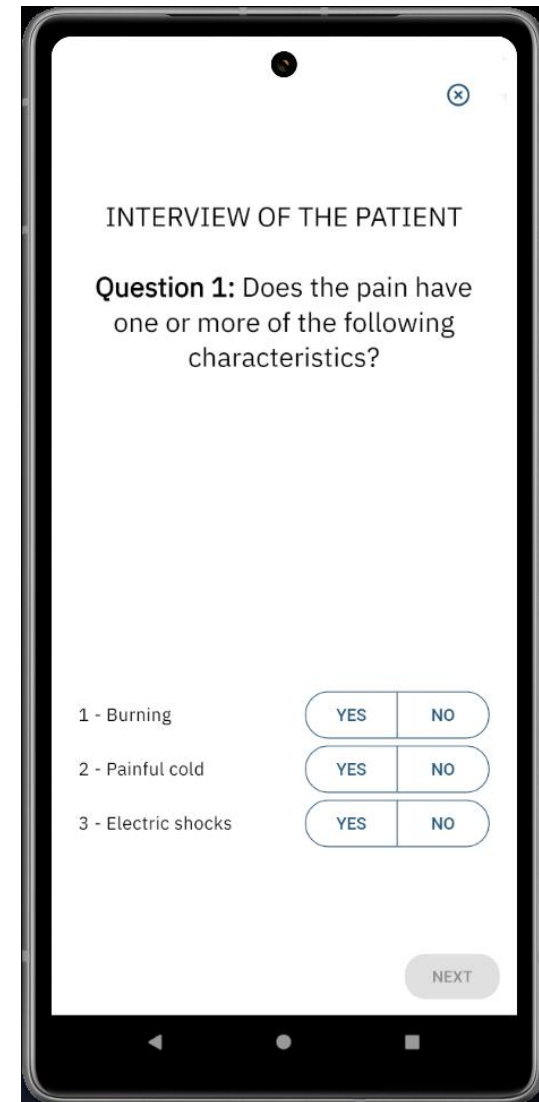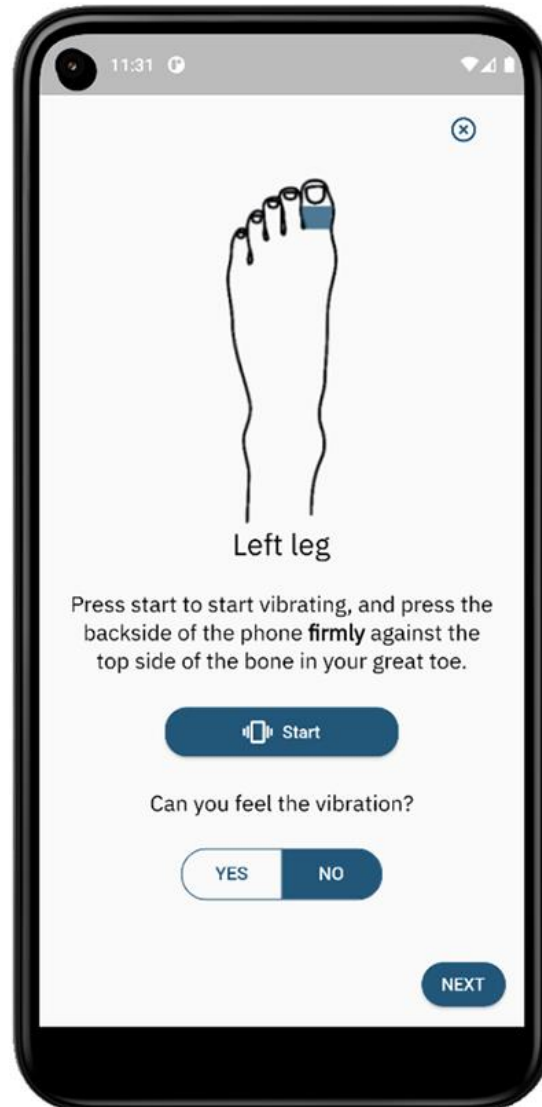
# Creating a survey with RP

```
List<RPChoice> yesNo = [
  RPChoice(text: "yes", value: 1),
  RPChoice(text: "no", value: 0),
];

RPChoiceAnswerFormat yesNoAnswerFormat = RPChoiceAnswerFormat(
  answerStyle: RPChoiceAnswerStyle.SingleChoice,
  choices: yesNo,
);

RPQuestionStep smokingQuestionStep = RPQuestionStep(
  identifier: "booleanQuestionStepID",
  title: "doYouSmoke",
  optional: false, // This parameter is optional
  answerFormat: yesNoAnswerFormat,
);
```

```
RPStepJumpRule noSmokingRule =
    RPStepJumpRule(answerMap: {0: alcoholQuestionStep.identifier});

RPNavigableOrderedTask examinationTask = RPNavigableOrderedTask(
    identifier: 'ExaminationTaskID',
    steps: [
        introductionStep,
        smokingQuestionStep,
        cigarettesADayQuestionStep,
        alcoholQuestionStep,
    ])
  ..setNavigationRuleForTriggerStepIdentifier(
      noSmokingRule, smokingQuestionStep.identifier);
```

```
@override
Widget build(BuildContext context) {
  return RPUITask(
    task: examinationTask,
    onSubmit: (result) {
      resultCallback(result);
    },
  );
}
```
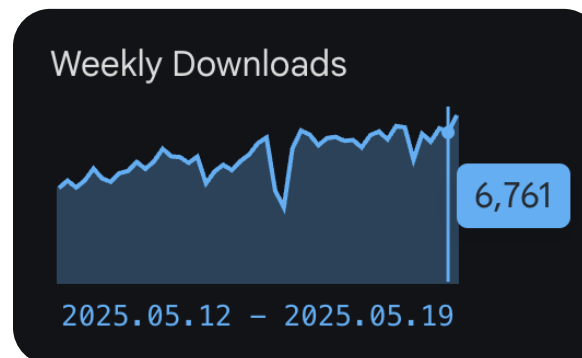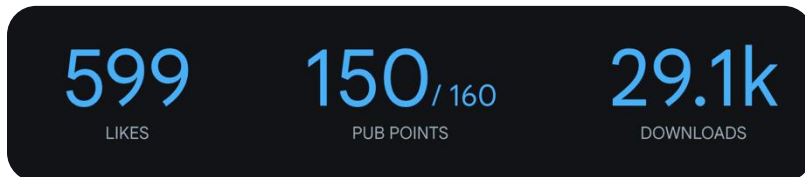
# Flutter Health Plugin

### Android Health Connect

### Apple HealthKit

# Flutter Health Plugin

- Handling permission access

- CURD data for both iOS and Android

  (Workouts, Data Types, Specific Tasks, etc.)

- Easy setup and usage ;)

**599** LIKES
**150** / 160 PUB POINTS
**29.1k** DOWNLOADS

Weekly Downloads

6,761

2025.05.12 – 2025.05.19

```dart
final health = Health();
await health.configure();

// define the types to get
var types = [
    HealthDataType.STEPS,
    HealthDataType.BLOOD_GLUCOSE,
];

bool requested = await
health.requestAuthorization(types)
;

var now = DateTime.now();

List<HealthDataPoint> healthData
= await
health.getHealthDataFromTypes(
    now.subtract(Duration(days:
1)), now, types);
```
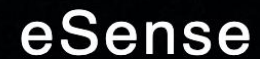
# CARP Sampling Package

**Polar H10 Sensor**
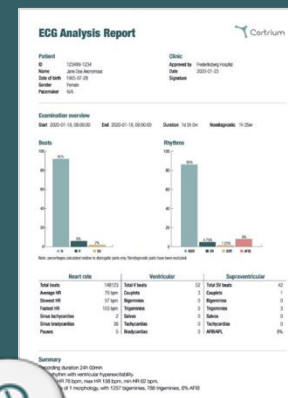
**C3+ ECG Sensor**

**HR & MD Sensors**

**Earbuds**

- Unified way to access data from different sensors

- Triger action for different sensor with one code

- Unified output from all types of sensors

- Storage and device management

# Cortrium ECG Made Easy

**Refactor into Flutter codebase**

Henrik Bo Hansen
Chief Product Officer

# Cortrium Product Overview

- Cortrium Apex + C3$^+$ Holter Monitor

- End-to-End ECG Solution

- C3$^+$ device: 3-channel, cable-free, 14-day recording
- Apex software: cloud portal for setup, upload & reporting

- Extended monitoring for improved arrhythmia detection
- Ultra-light, patient-friendly form factor
- Real-time signal preview & one-click data transfer
- Human-verified

# Cortrium C3 Ambulatory ECG Device

- **3-lead Holter monitor: continuous ECG at 256 Hz, 24-bit**
- **14-day battery: 3.7 V, 680 mAh Li-Polymer**
- **Compact: 85 × 80 × 15 mm, 36 g**
- **Rugged: IP67 dust-/water-proof**
- **Connections:**
  - **3 electrode leads**
  - **USB-C for charging & data offload (BLE for App)**
- **Standards: IEC 60601-1/2-47; ISO 10993-5/-10**

# Cross-Platform Refactor Project

**Issues !!**

- **Two native codebases (Android & iOS) → duplicated effort**
- **Separate pipelines → higher maintenance & slower releases**
- **Inconsistent UX → harder to build user habits / No SDK**

**Solution** ✅

- **Rebuild in Flutter → single codebase for both platforms with SDK**
- **Unified CI/CD & shared libraries → faster time-to-market**
- **Consistent UI & UX + rich ecosystem → easier maintenance and better user experience**

# Cortrium C3 Flutter SDK Overview

**Plugin Structure**
- **DeviceManager singleton wrapping native BLE drivers**

**Requirements**
- **Flutter ≥ 1.17.0 · Dart ≥ 3.4.1 < 4.0.0 · iOS 13+ · Android SDK 21+ · Java 17 · AGP ≥ 8.3 · Gradle ≥ 8.4**

**Permissions & Setup**
- `AndroidManifest` **– BLE feature + Android 12+ scan/connect perms + legacy BT/location**
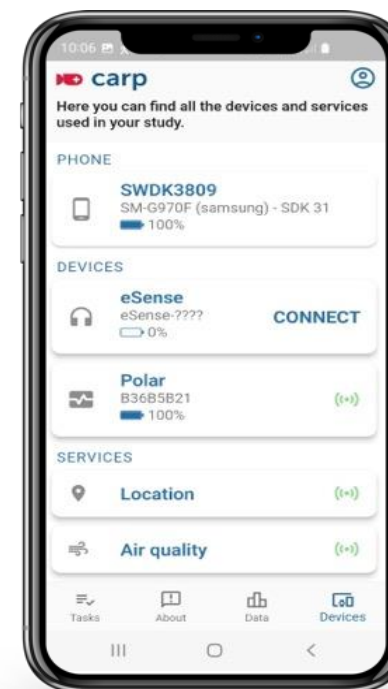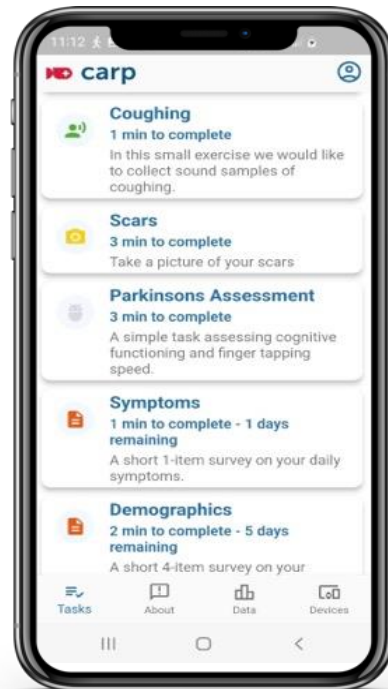- `Info.plist` **– NSBluetoothAlwaysUsageDescription**

**Event Streams**
- `connectionEvents` **→ discovery/state/scanning**
- `dataEvents` **→ ECG samples**
- `accEvents` **→ accelerometer**
- `batteryEvents` **→ battery level**
- `buttonEvents` **→ button presses**
- `historicDataEvents` **+ getHistoric → last 30 s ECG**

**Workflow**
- `startScanning → connect → subscribe → disconnect → cleanup`
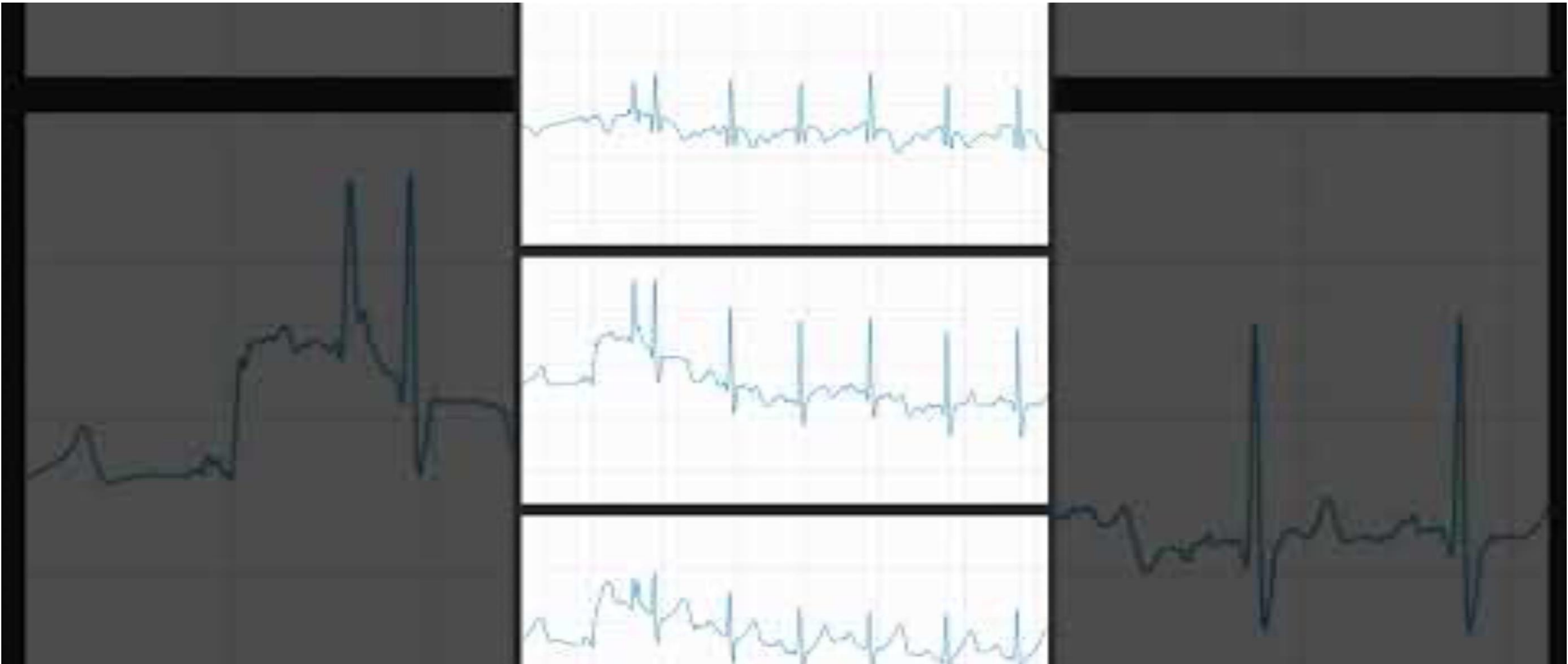
# CARP Studies App

# Today's Program

**OVERVIEW OF THE CARP PROJECT**

- Mobile and Wearable Technology for Health
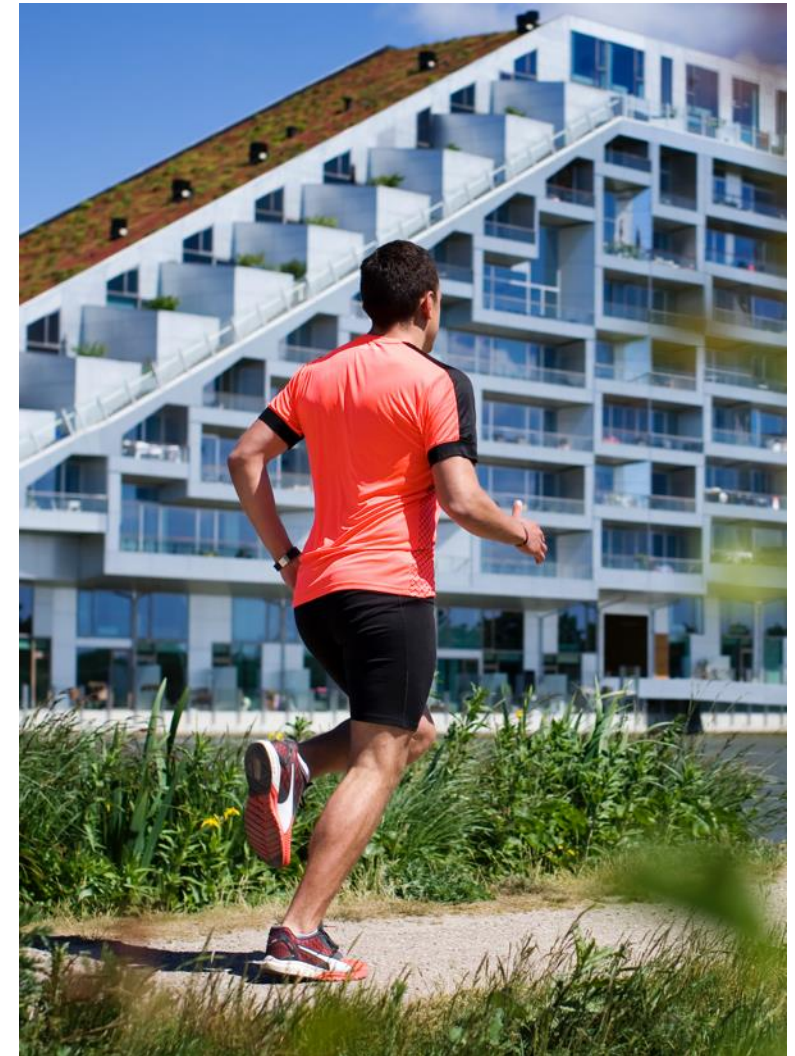- Architecture & Packages

**EXAMPLES**

- Neurology (Neuropathy Tracker)
- `health` and other CARP plugins
- Cardiology and ECG recordings
- Digital Phenotyping (Studies App)

**DEMOS & BREAK**

**CLOSING**

- Documentation & Resources
- Looking ahead

# Documentation

- **Website** – www.carp.dk
  - Overview
  - Tutorials
  - News
- **Github** – github.com/cph-cachet
  - Components
  - Open source projects
  - Study Configurations
  - Documentation – Wiki
  - Issue tracking & Release management
- **Scientific papers**
  - ... mostly for academics who want to reference our work
  - Reports on "research", i.e., studies done using CARP

# Looking Ahead...

- **Maintenance (!%&#€)**
  - continue to support the Flutter Plugin community, especially the `health` plugin
  - CARP Mobile Sensing
  - support for new devices (Sampling Packages)

- **CARP Data Analytics** (the "Analysis" sub-system)
  - real-time data processing / ML
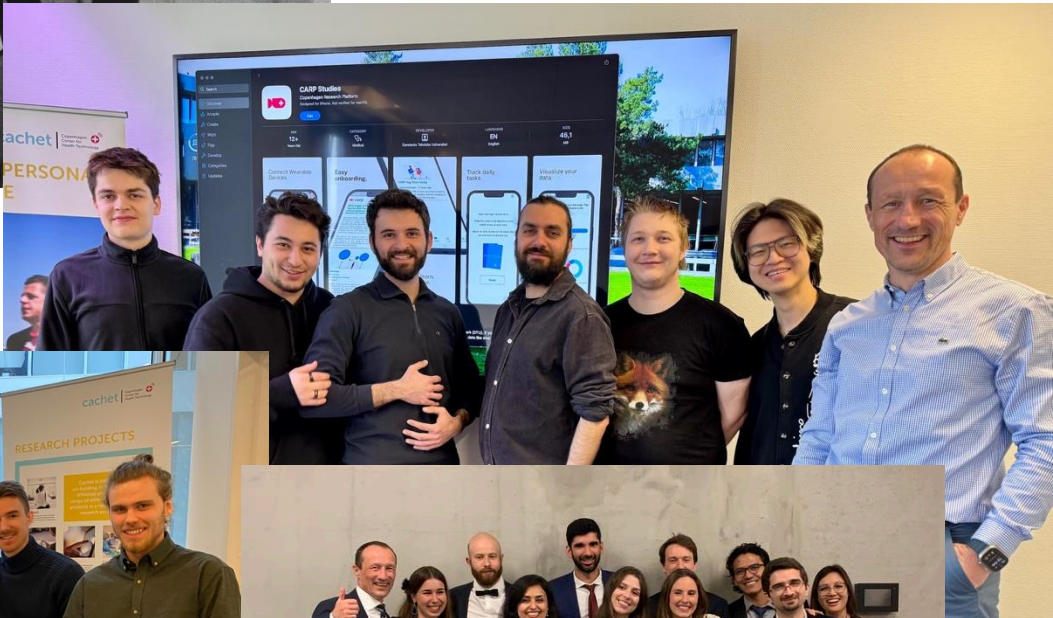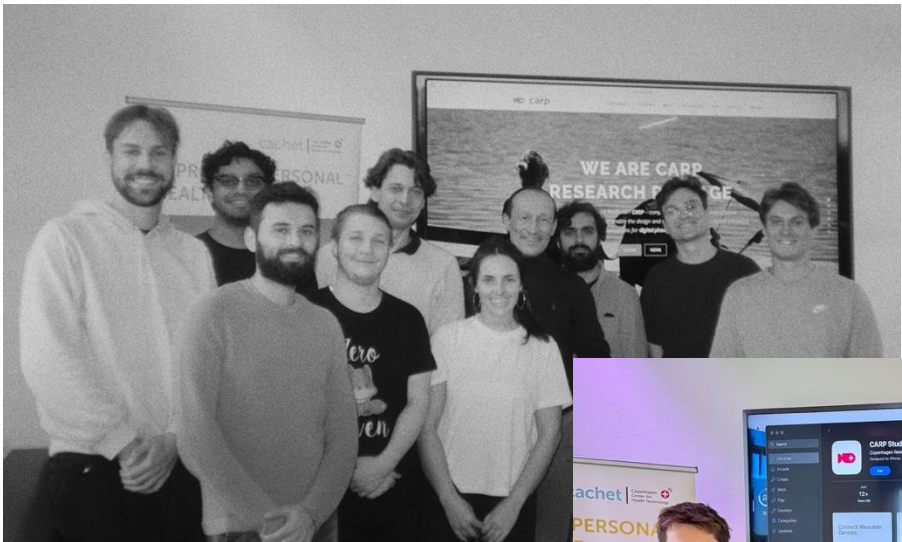  - server-side
  - edge computing

- **Your ideas?**
  - ...

# FUNDING

# CARP TEAM

People

View all

Contributors 13