



**baseflow**

TEAM | TECH | INNOVATION

# An introduction into Flutter

Questions are always allowed



**But first a bit about us** 



# Nice to meet you

## My past

Studied in Eindhoven – Innovation Sciences

Worked with Xamarin at MedApp

Created my first open source (Flutter) project in December 2017: *CachedNetworkImage*

## My present

Work since Februari as software developer at Baseflow

Worked on 1 project with

- Dart (Flutter)
- Swift (iOS BLE logic)
- Kotlin (Android BLE logic)
- C# (.Net Core backend)
- A little bit javascript
- A little bit objective C

Work on the Baseflow open source Flutter packages



**Rene Floor**

Software developer





# baseflow

TEAM | TECH | INNOVATION

[hello@baseflow.com](mailto:hello@baseflow.com)

# Asito - DSP

## Client's challenge

Disruptions cannot be resolved easily & efficiently, because the location of employees is unknown.

## Goal

Creating a bluetooth enabled driven application on ipad and mobile devices to connect employees with situations that need immediate attention.

## Key results

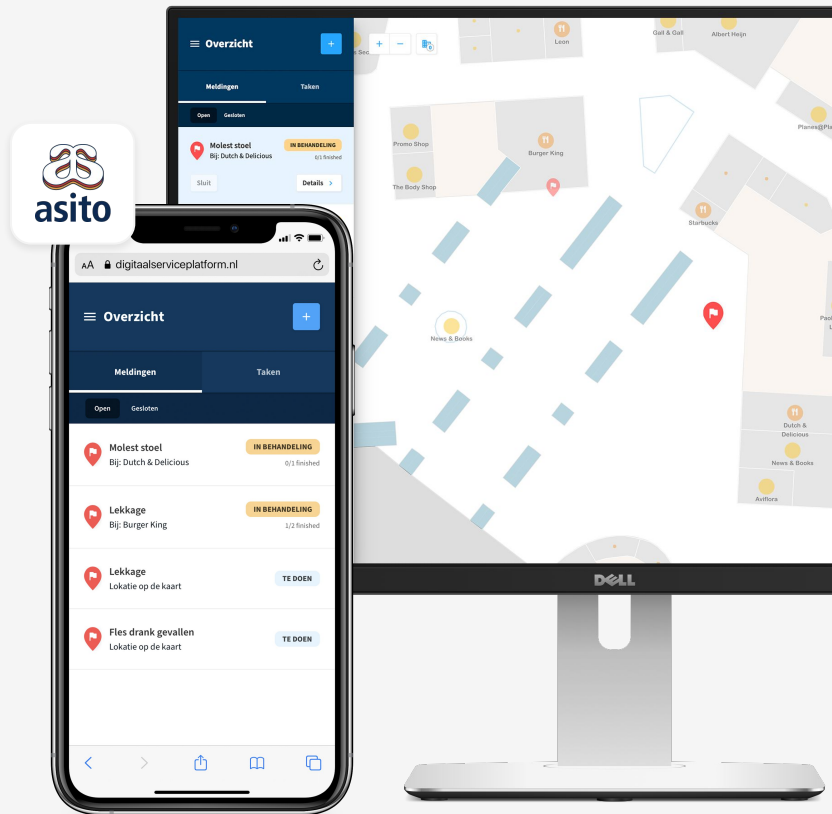
- Productivity increase of the staff with timely responses to hazardous incidents
- Fewer accidents reported within the environment, the areas are safer and clean

## Added value

Improvements to staffing solutions after analysing tracking data from the app meant the company could save money while providing excellent service.



*Wouter Arkink (Manager Business Development, Asito): "With Baseflow we work as one team. Together we determine where we're going"*



# WeGo Carsharing

## Client's challenge

To create a shift in the way cars are used as a means of mobility by increasing car sharing between different users.

## Goal

A technology solution that; integrated well with an existing Bluetooth technology base, was easily adaptable and human-friendly and was able to be rolled out in under three months.

## Client's challenge

A beautiful app made with Flutter with logical user experience. The result was a native-like App for both iOS and Android devices. An impeccable Geolocation device that helps you show your location and where your car is located. Never get lost or lose your car again!

## Key results

- Fewer cars on the road
- The estimated reduction in emissions from cars
- Amount of money companies can save by using this
- Public space is freed up because fewer parking spaces are needed at ground level,
- Car mobility costs are reduced for companies and individuals.



# An introduction into Flutter

What is Flutter

Everything is a Widget

Making an app reactive with Riverpod

Testing





# What is Flutter

**A SDK** that makes building high-performing, modern and beautiful apps easy

**Works** for Android, iOS, macOS, Windows, Linux, the web and other platforms \*

**An** open-source toolkit, developed by Google

**Many** contributions from the open source community

\* Desktop and web are not production ready yet.



# Who is Flutter for?

**Designers** converge on a brand-driven experience on Android and iOS

**Prototypers** enjoy a high-fidelity and fast way to build working prototypes.

**Developers** benefit from fantastic developer tools, an easy-to-use language, a rich set of widgets and great IDE support. Flutter frees up valuable time for working on features and delightful experiences.



# Great tooling

Powerful, yet easy to use **CLI**

Full featured **extensions** for IntelliJ IDEA, Android Studio and Visual Studio Code

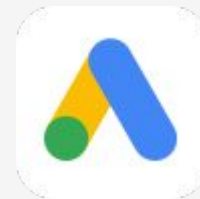
**Hot reload**

Advanced **debugging**

Build-in **package manager**



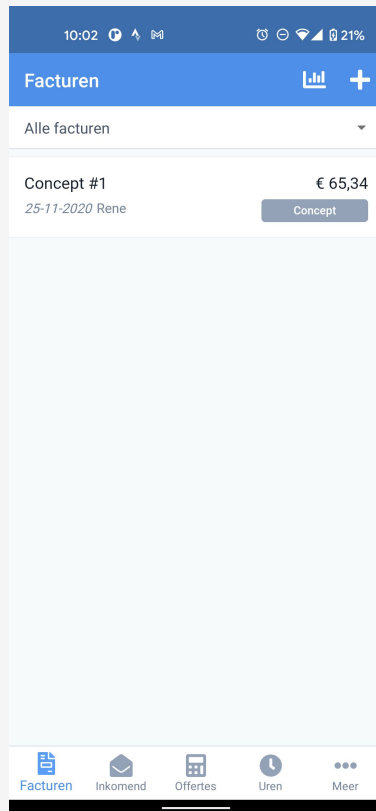
# Cases



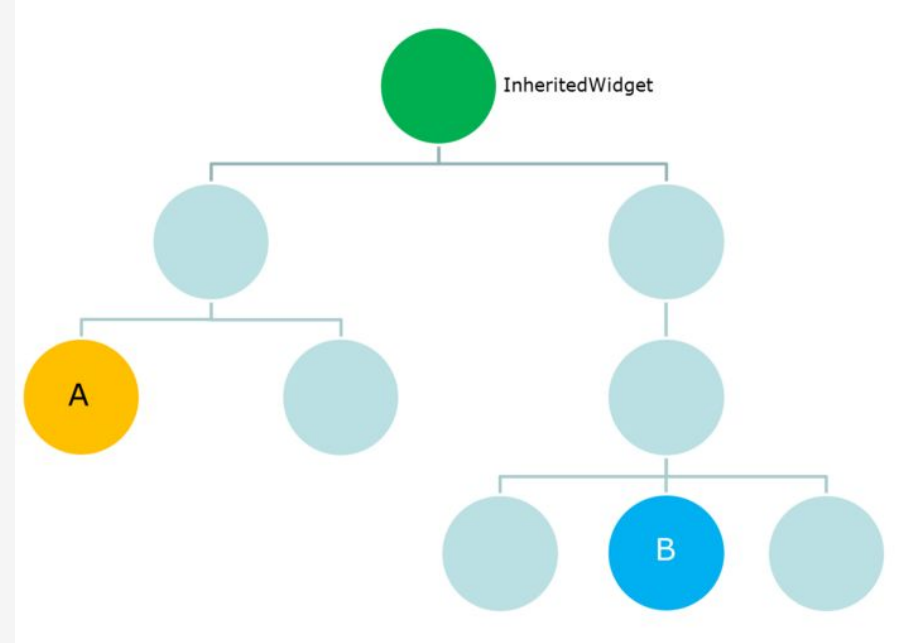
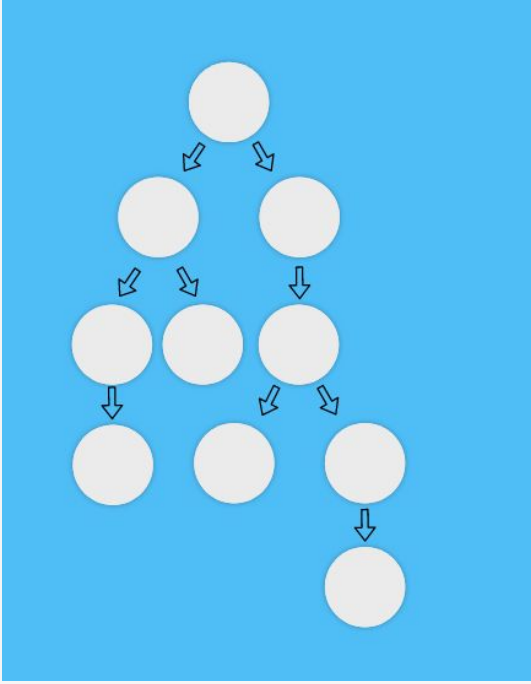
# Everything is a Widget



# Everything is a Widget



# Inherited Widget



# Inherited Widget: Theme

```
Container(  
  color: Theme.of(context).accentColor,  
  child: Text('Text with a background color',  
    style: Theme.of(context).textTheme.headline6,  
  ),  
);
```

“Inherited widgets, when referenced in this way, will cause the consumer to rebuild when the inherited widget itself changes state.”





# Inherited Widget: Riverpod



Provider, but different

Get Started

## Create a provider

```
final counterProvider = StateNotifierProvider((ref) {  
  return Counter();  
});  
  
class Counter extends StateNotifier<int> {  
  Counter() : super(0);  
  
  void increment() => state++;  
}
```

## Consume the provider

```
class Home extends ConsumerWidget {  
  @override  
  Widget build(BuildContext context, ScopedReader watch) {  
    final count = watch(counterProvider.state);  
  
    return Text('$count');  
  }  
}
```

# Widget testing



# More info

[baseflow.com](https://baseflow.com)

[github.com/baseflow/moneybird-webinar](https://github.com/baseflow/moneybird-webinar)

[twitter.com/FloorRene](https://twitter.com/FloorRene)





# baseflow

TEAM | TECH | INNOVATION

[hello@baseflow.com](mailto:hello@baseflow.com)