



life.augmented

Mobile apps & Internet of Things



Agenda

1 IoT Components

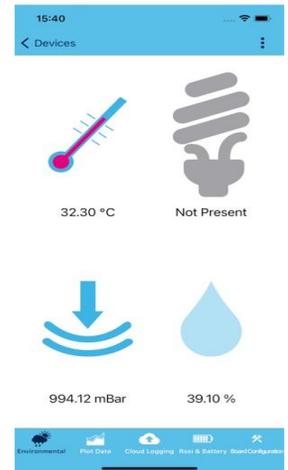
2 Mobile Platform

3 IoT App Architecture

4 Demos : Mobile App and
Web Application

IoT Components

- Hardware comprises of computation, sensing and connectivity at board level
- Mobile App on iOS / Android Platform for small sets of data / alarm
 - The purpose is consumption of data
- Web Application to monitor large sets of data and do some deep / complex computation of data
 - The purpose is both consumption and computation of data



Mobile applications



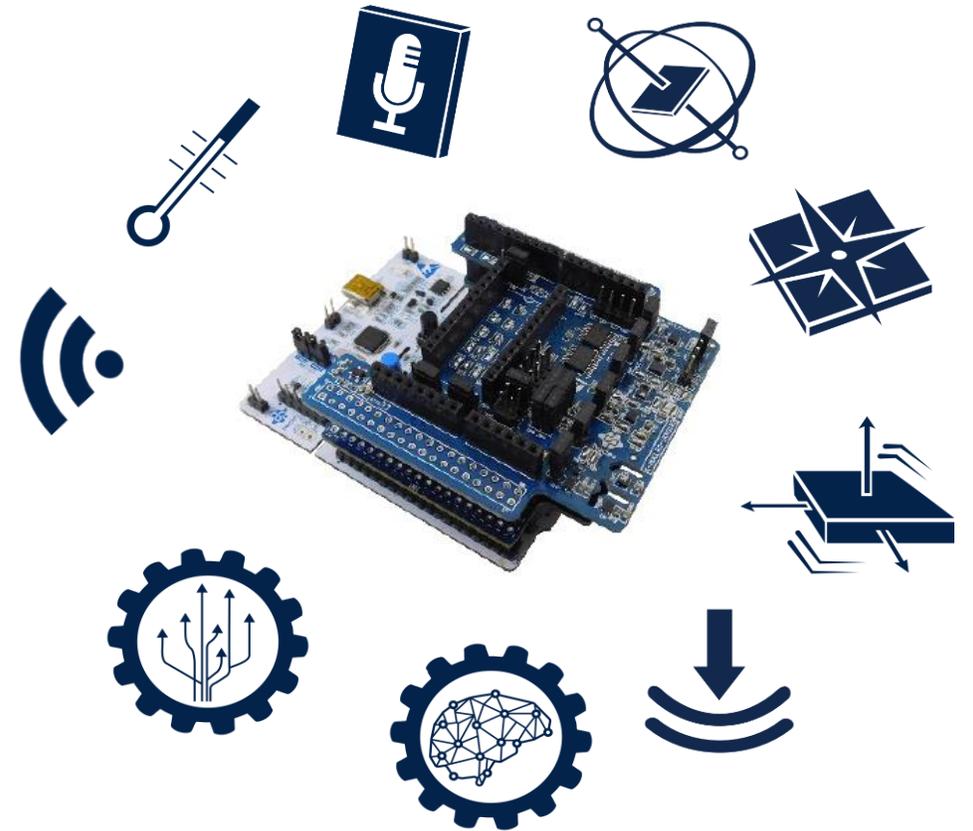
Wi-Fi

Bluetooth

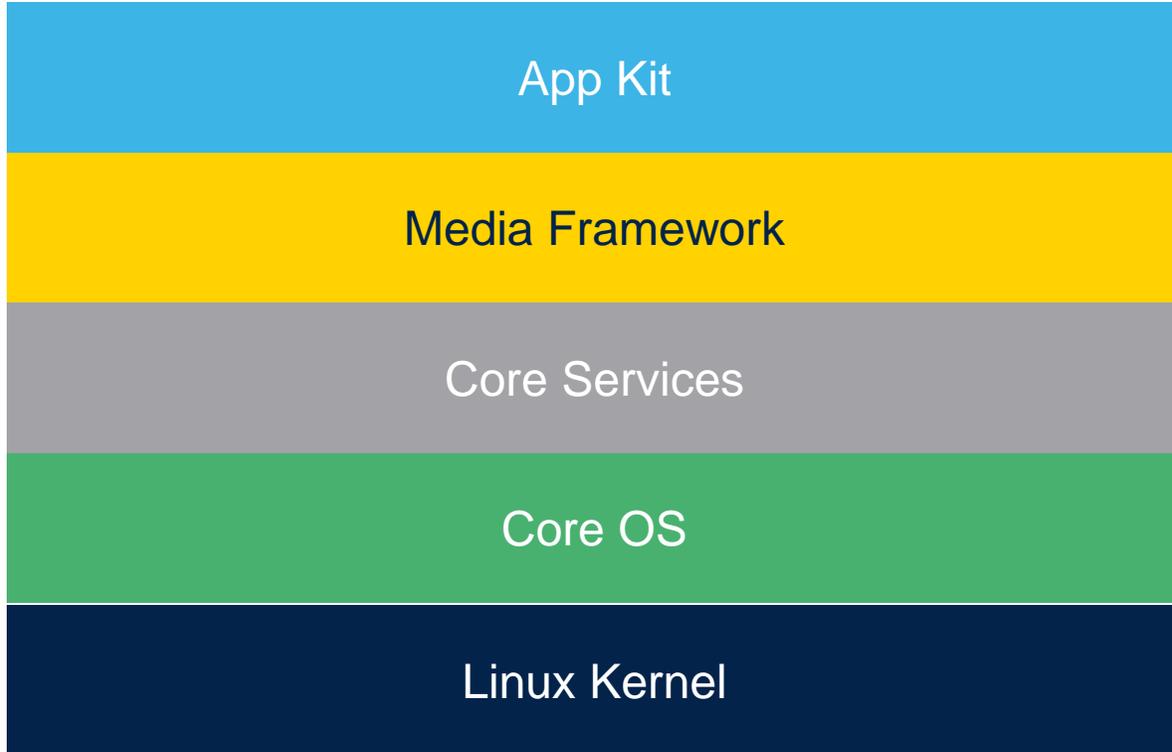
NFC

GPS

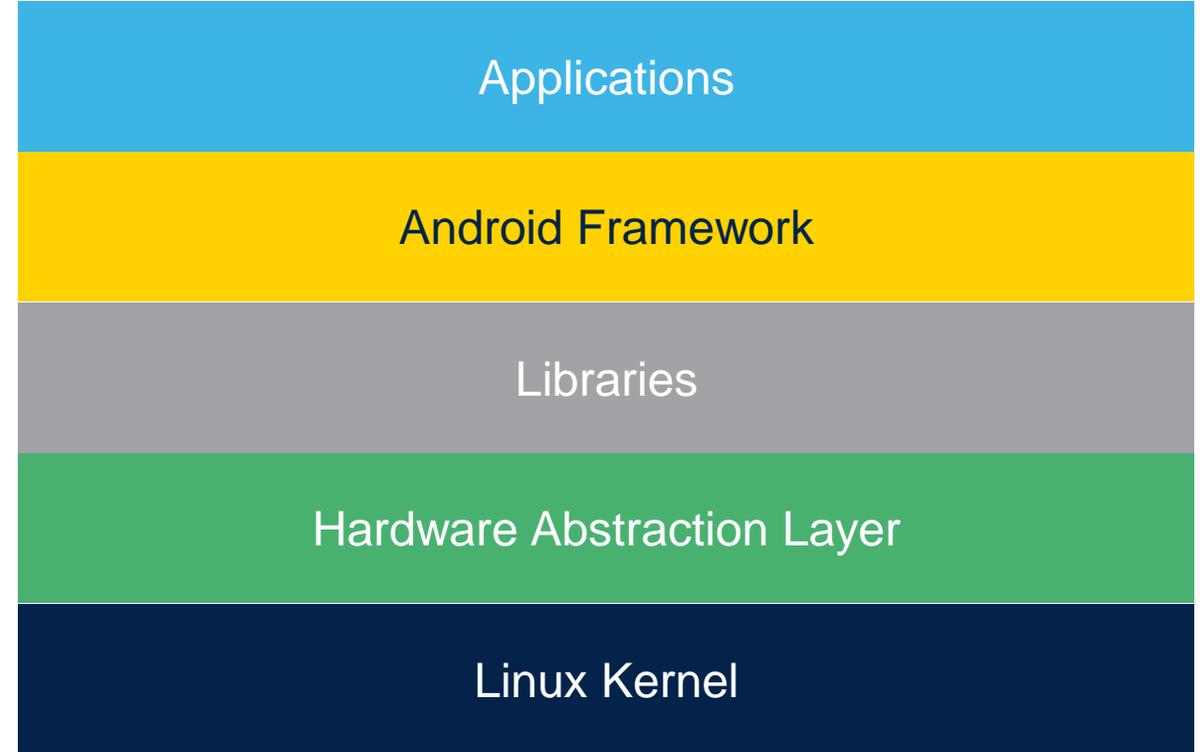
Sensors



Mobile OS architecture



iOS



Android

Hardware support in Mobile Phones

- Near Field Communication
- Bluetooth
- Motion Sensors
- Ambient Light Sensors
- Touchscreen
- GPS
- Fingerprint detection
-



Activity Recognition

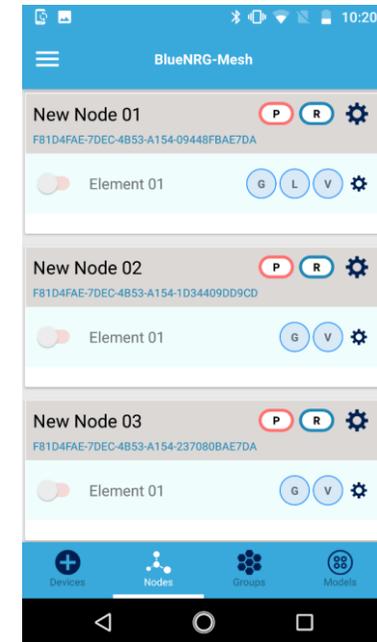


Vertical context detection
Stair count / Man fall down
Pose & Fitness monitoring

Vertical Detection



Altitude monitoring
Indoor V position for E911
Ambient pressure monitor



developers  Platform Android Studio Google Play Jetpack Kotlin Docs Games

DOCUMENTATION

Overview Guides Reference Samples Design & Quality

On-device search

Games

- Tools
- Customize or port game engines
- Use prebuilt or turnkey game engines
- Game best practices
- Optimize
- Distribute

Best practices

- Testing
- Performance
- Accessibility
- Privacy
- Security

Android Developers > Docs > Guides Rate and review

Near field communication overview

Near Field Communication (NFC) is a set of short-range wireless technologies, typically requiring a distance of less to initiate a connection. NFC allows you to share small payloads of data between an NFC tag and an Android-powered device, or between two Android-powered devices.

Tags can range in complexity. Simple tags offer just read and write semantics, sometimes with one-time-programmable areas to make the card read-only. More complex tags offer math operations, and have cryptographic hardware to authenticate access to a sector. The most sophisticated tags contain operating environments, allowing complex interactions with code executing on the tag. The data stored in the tag can also be written in a variety of formats. Many of the Android framework APIs are based around a [NFC Forum](#) standard called NDEF (NFC Data Exchange Format).

Android-powered devices with NFC simultaneously support three main modes of operation:

1. **Reader/writer mode**, allowing the NFC device to read and/or write passive NFC tags and stickers.

DOCUMENTATION

Overview Guides Reference Samples Design & Quality

App Actions

Slices

On-device search

Games

- Tools
- Customize or port game engines
- Use prebuilt or turnkey game engines
- Game best practices
- Optimize
- Distribute

Best practices

- Testing

Android Developers > Docs > Guides Rate and review 

Bluetooth overview

The Android platform includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The app framework provides access to the Bluetooth functionality through Bluetooth APIs. These APIs let apps connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features.

Using the Bluetooth APIs, an app can perform the following:

- Scan for other Bluetooth devices.
- Query the local Bluetooth adapter for paired Bluetooth devices.
- Establish RFCOMM channels.

Software APIs support

Property List Key

Near Field Communication Tag Reader Session Formats Entitlement

The Near Field Communication data formats an app can read.

Availability
iOS 11.0+

Details

Key
com.apple.developer.nfc.readersession.formats

Framework
Core NFC

On This Page
[Details](#)
[PossibleValues](#)
[Discussion](#)
[See Also](#)

Type
Array of strings

Possible Values

Framework

Core Bluetooth

Communicate with Bluetooth low energy and BR/EDR ("Classic") Devices.

Availability
iOS 5.0+
macOS 10.10+
Mac Catalyst 13.0+
tvOS 9.0+
watchOS 4.0+

Overview

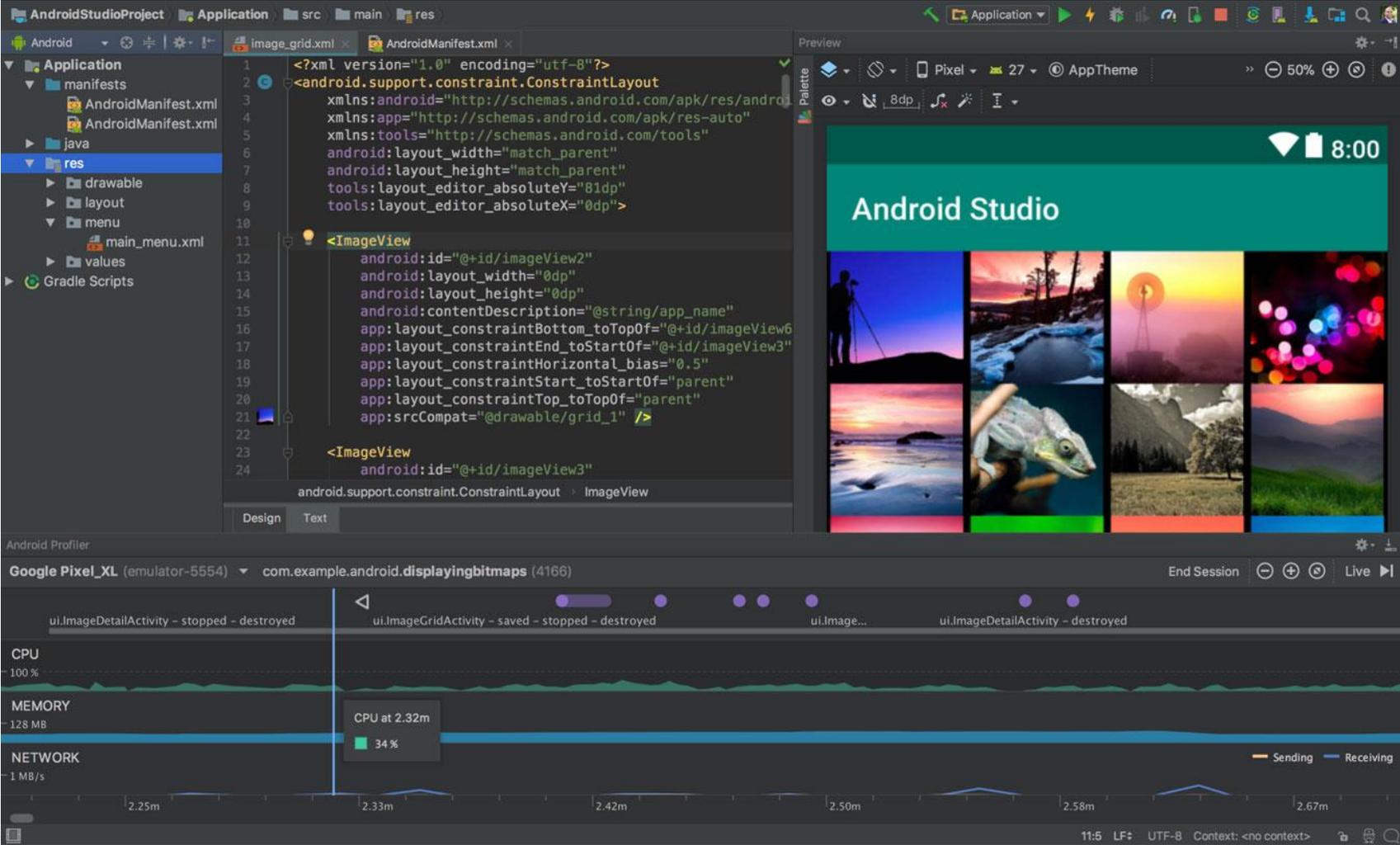
The Core Bluetooth framework provides the classes needed for your apps to communicate with Bluetooth-equipped low energy (LE) and Basic Rate / Enhanced Data Rate (BR/EDR) wireless technology.

Don't subclass any of the classes of the Core Bluetooth framework. Overriding these classes isn't supported and results in undefined behavior.

Core Bluetooth background execution modes aren't supported in iPad apps running on macOS.

On This Page
[Overview](#)
[Topics](#)
[See Also](#)





Android Studio (2/2)

The screenshot displays the Android Studio interface for a project named 'Malavidaapp'. The main editor shows the XML code for 'activity_fullscreen.xml' with the following content:

```
<?xml version="1.0" encoding="utf-8" android:namespace="http://schemas.android.com/apk/res/android"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#0099cc"
    tools:context=".Software">

    <!-- The primary full-screen view. This can be replaced with any view that
         is needed to present your content, e.g. ViewPager, etc. -->
    <TextView android:id="@+id/fullscreen_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:keepScreenOn="true"
        android:textColor="#33b5e5"
        android:textStyle="bold"
        android:textSize="50sp"
        android:gravity="center"
        android:text="@string/dummy_content" />

    <!-- This FrameLayout insets its children based on the platform specific default insets
         and fills the remaining space -->
```

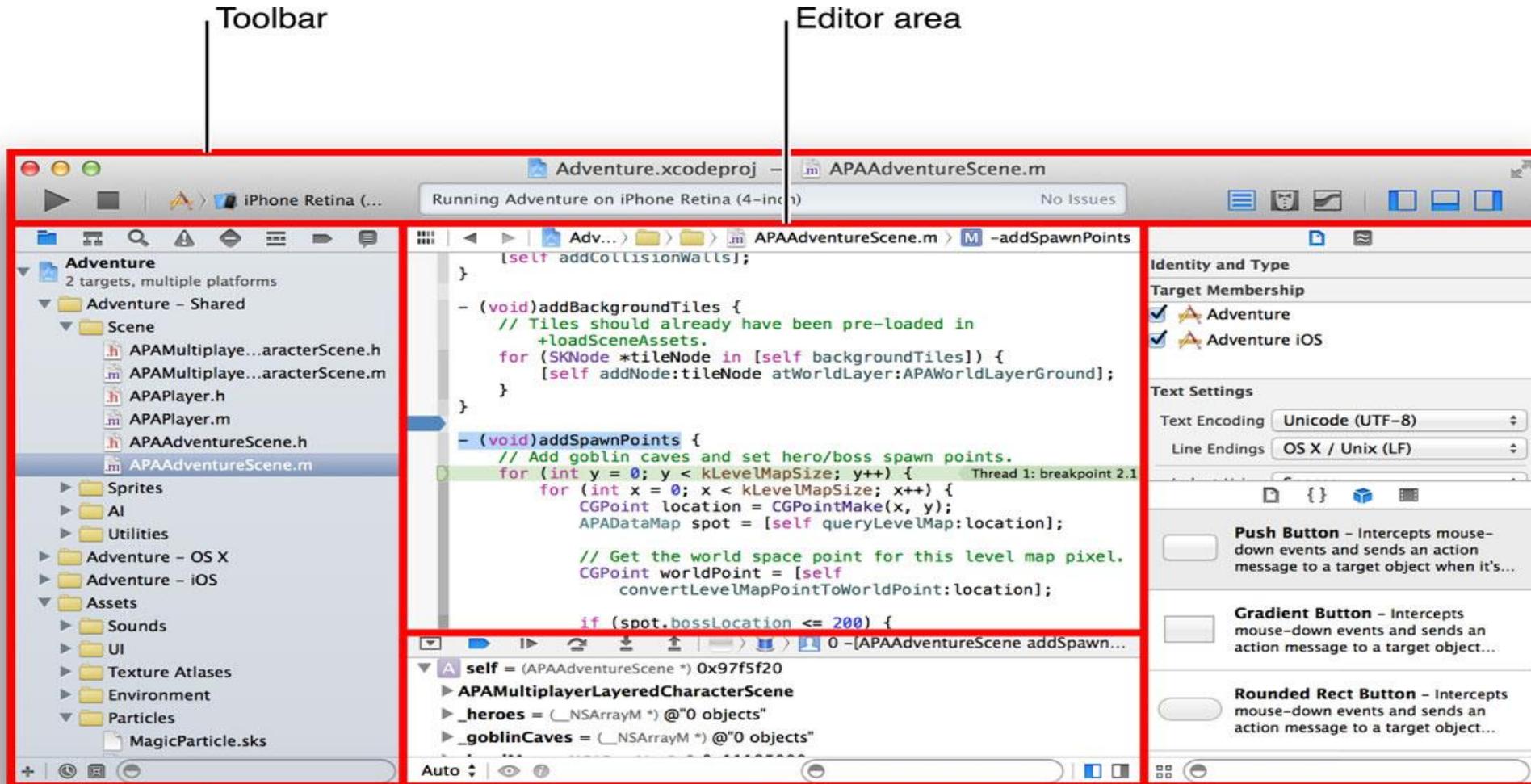
The preview window on the right shows a Nexus 4 device with the app running. The screen displays 'Malavida app' at the top, a blue background with 'DUMMY CONTENT' in the center, and a 'Dummy Button' at the bottom.

The Messages window at the bottom shows the following error messages:

- Information: Compilation completed with 3 errors and 0 warnings in 17 sec
- Information: 3 errors
- Information: 0 warnings
- Error: Gradle: Execution failed for task ':Malavidaapp:processDebugResources'.
> Could not call IncrementalTask.taskAction() on task ':Malavidaapp:processDebugResources'
- C:\Users\Ontecnia\AndroidStudioProjects\MalavidaappProject\Malavidaapp\src\main\res\values\attrs.xml
- Error: Gradle: Attribute "buttonBarStyle" has already been defined
- Error: Gradle: Attribute "buttonBarButtonStyle" has already been defined

The status bar at the bottom indicates 'Compilation completed with 3 errors and 0 warnings in 17 sec (moments ago)'. The bottom right corner shows '1:1 LF UTF-8'.

Xcode IDE (2/2)



Toolbar

Editor area

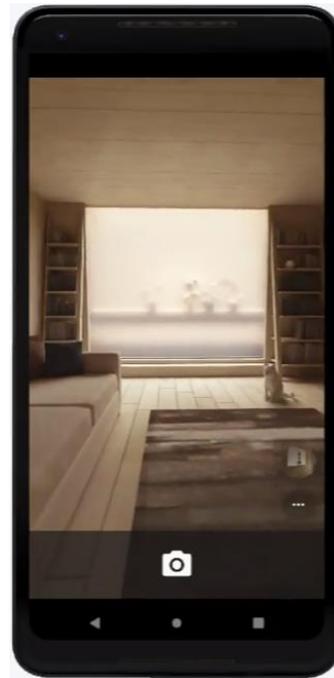
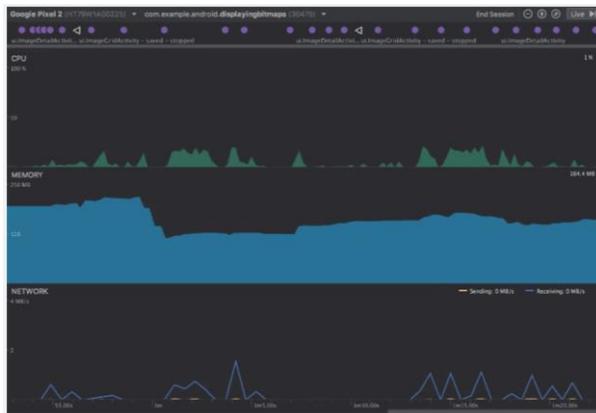
Navigator area

Debug area

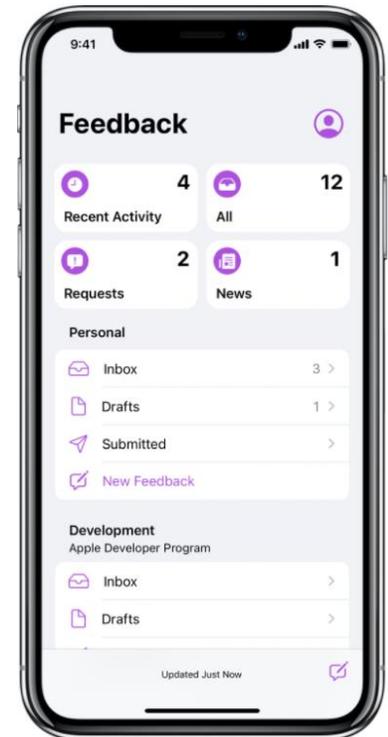
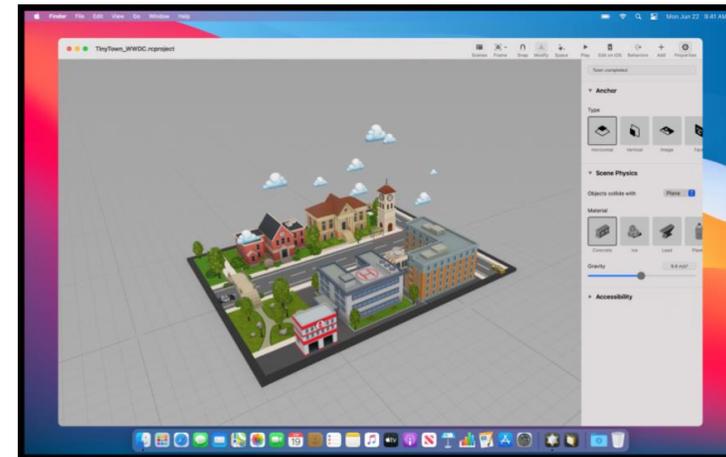
Utility area

IDE features

- Android Studio
 - APK Analyzer
 - Fast Emulator
 - Intelligent Code Builder
 - Flexible Build Systems
 - Visual Layout Editor
 - Realtime Profilers



- iOS
 - CloudKit Console
 - Command Line Tools
 - Dashboard
 - Reality Composer
 - Reality Builder
 - Swift UI



Documentation for each platform

The screenshot shows a web browser at the URL <https://developer.android.com/studio/intro>. The page title is "Meet Android Studio" and it is part of the "Android Studio" section of the "Platform" navigation. The main content area features a breadcrumb trail "Android Developers > Android Studio > User guide" and a large heading "Meet Android Studio". Below the heading, it states: "Android Studio is the official Integrated Development Environment (IDE) for Android. On top of IntelliJ's powerful code editor and debugger, it enhances your productivity when building Android apps." A list of features follows:

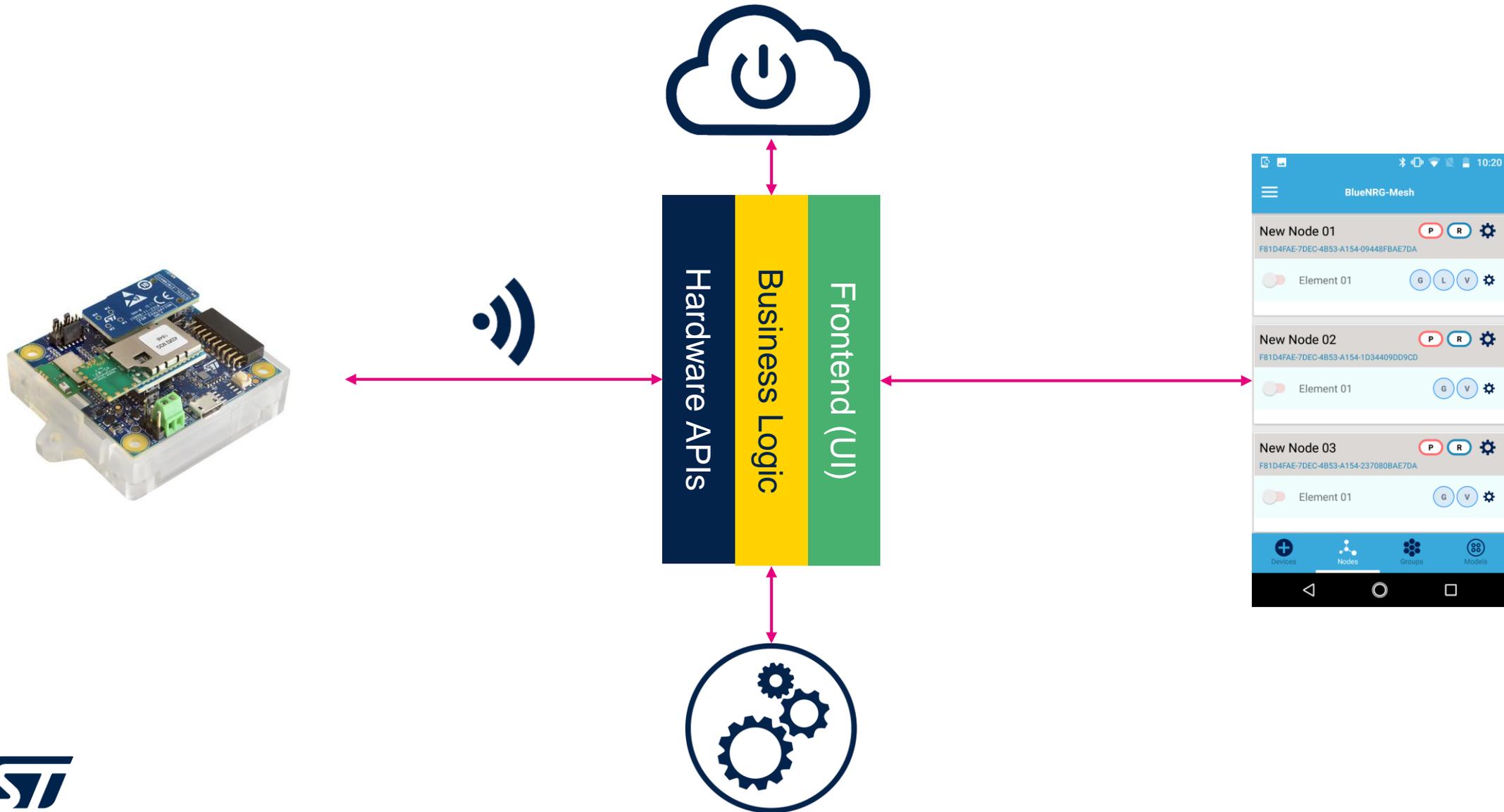
- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for Android, iOS, and Wear OS
- Apply Changes to push code and resource changes to a running app
- Code templates and GitHub integration to help you get started
- Extensive testing tools and frameworks

A left-hand navigation menu lists various topics: Meet Android Studio, Workflow basics, Manage your project, Write your app, Build and run your app, Configure your build, Debug your app, Test your app, Profile your app, Publish your app, Command line tools, Troubleshoot, Known issues, and Report a bug.

This block contains a grid of links for documentation. The links are organized into two columns: "Develop" and "Distribute".

Develop	Distribute
Xcode	Developer Program
Swift	App Store
Swift Playgrounds	App Review
TestFlight	Mac Software
Documentation	Apps for Business
Videos	Safari Extensions
Downloads	Marketing Resources
	Trademark Licensing

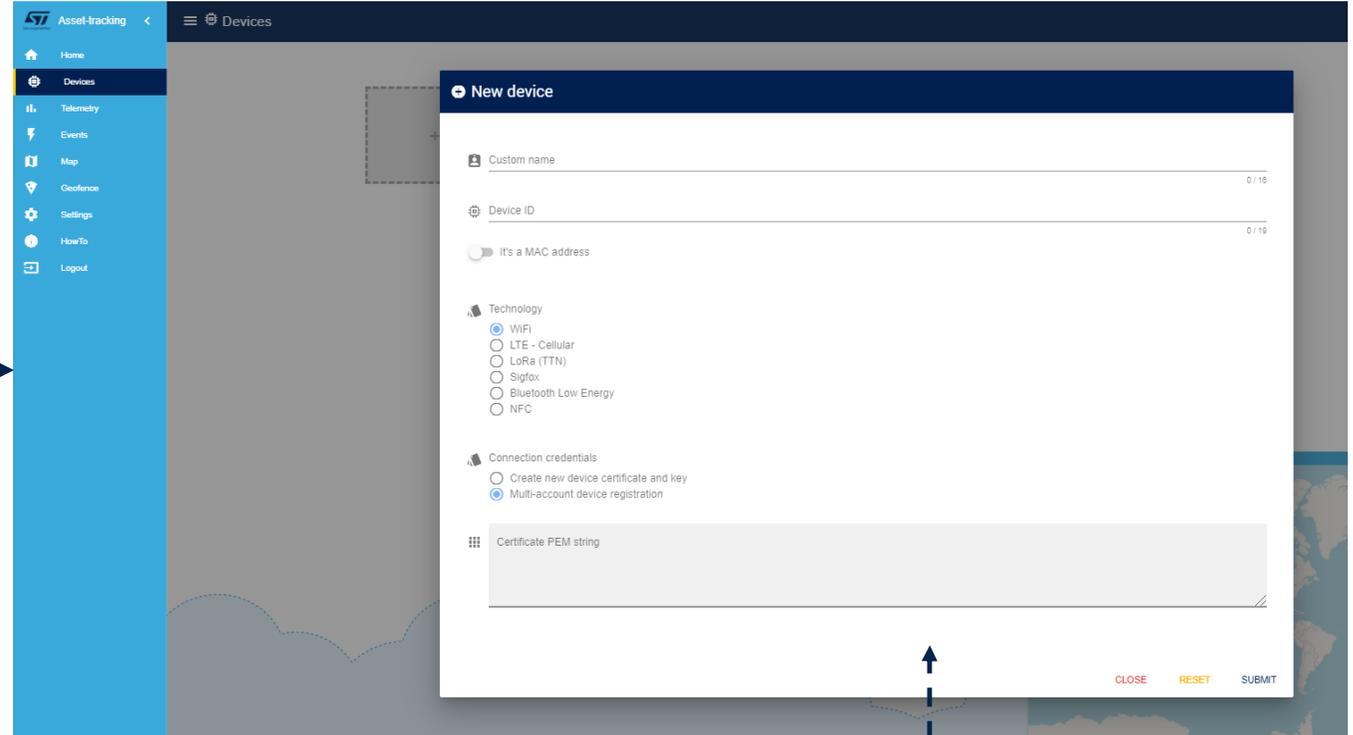
Typical mobile app architecture



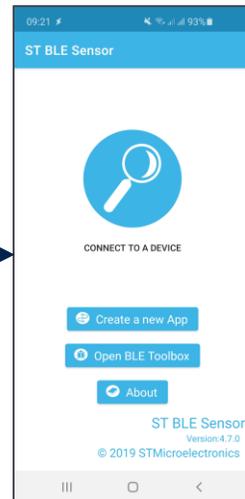
IoT data to cloud



Wi-Fi / Internet



BLE



Mobile app to Cloud

ST asset tracking dashboard

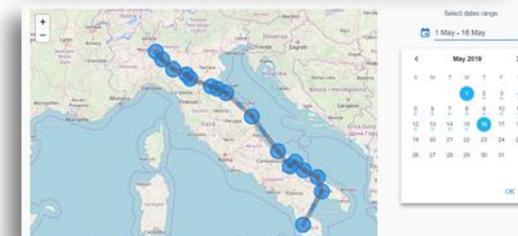
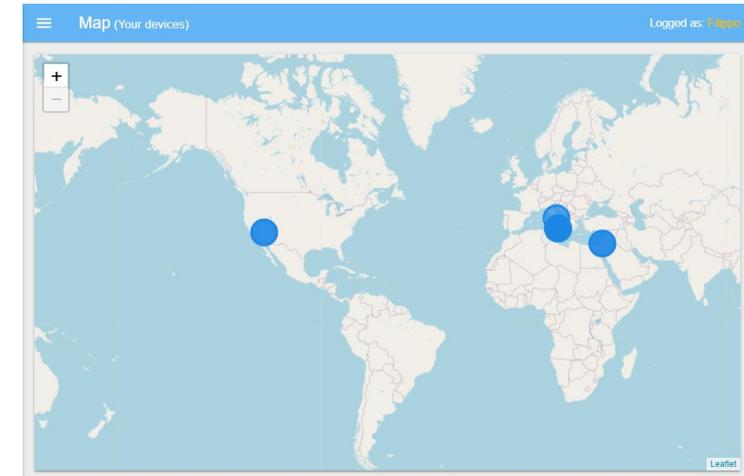
The screenshot displays the 'ASSET-TRACKING PLATFORM' dashboard with the subtitle 'Dashboard and Device Management'. The interface features a dark blue header with a 'Home' button and the ST logo. The main content area is set against a background of a satellite map showing a port with stacked shipping containers. Five interactive cards are arranged in a grid, each with a distinct icon and a 'GO' button. The cards are: 1. Register and configure your devices (green bar, gear icon); 2. Look at your devices telemetry (orange bar, bar chart icon); 3. Analyze the events detected by your devices (red bar, lightning bolt icon); 4. Monitor your device on their geo localization (cyan bar, location pin icon); 5. Set and detect geofencing events (purple bar, geofence icon).

ASSET-TRACKING PLATFORM
Dashboard and Device Management

- 1. Register and configure your devices**
Follow this [guide](#) to register a new device based on its connectivity capability.
GO
- 2. Look at your devices telemetry**
Select one or more device and the window time interested in, and then visualize telemetry data received by devices.
GO
- 3. Analyze the events detected by your devices**
Select one or more device and the window time interested in, and then visualize events received by devices.
GO
- 4. Monitor your device on their geo localization**
Select one or more device and the window time interested in, and then visualize geo-position data received by devices.
GO
- 5. Set and detect geofencing events**
Select one or more device and draw an area on map in order to enable tracking of geofence events raised by devices.
GO

ST asset tracking dashboard

The dashboard is titled "Dashboard (Device details)" and shows the user is logged as "Flippo". It features a header with a menu icon and the user name. Below the header, there is a section for device identification with the ID "C0634504002702E0" and a chip icon, and a section for the device name "sensortile1" with its MAC address "(F3:2F:C9:AB:85:5E)" and a Bluetooth icon. A green "Add new device" button is also present. The main content area is divided into several sections: "Device Status" with a trash icon and a red "NOT CONNECTED" button; "Dashboard settings" with a wrench icon and a "Show data of the last: 7 DAYS" button; "Temperature °C" with a bar chart icon and fields for MAX, MIN, and Latest; "Pressure hPa" with a bar chart icon and fields for MAX, MIN, and Latest; "Humidity %" with a bar chart icon and fields for MAX, MIN, and Latest; "GEO Tracking" with a location pin icon and fields for Latitude (40.91750000000) and Longitude (14.27570000000); and an "Events" section at the bottom with a list icon.

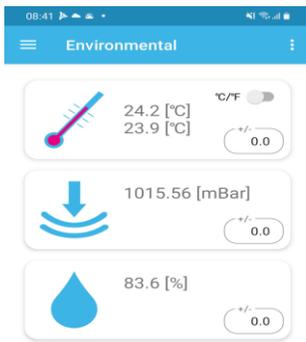
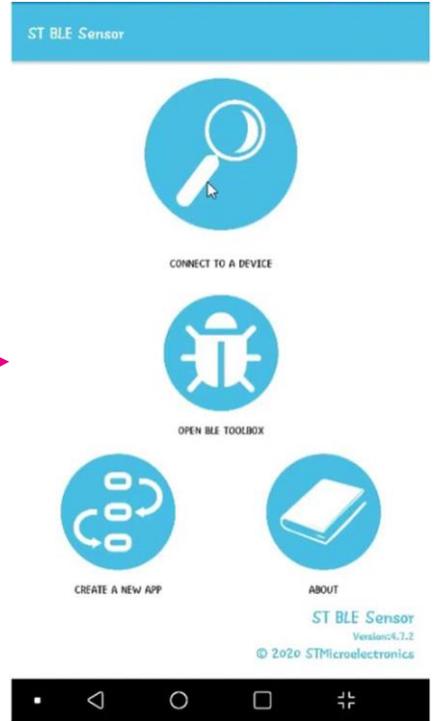
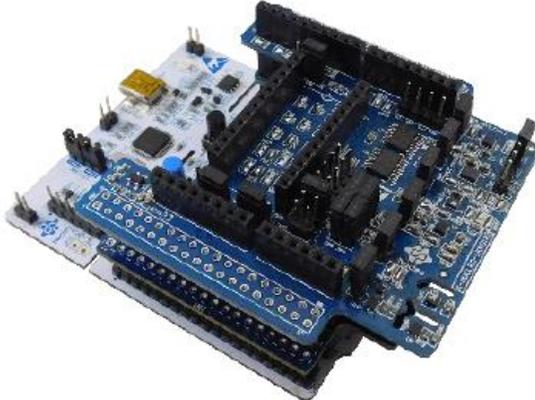


Demos



life.augmented

ST BLE Sensor App



Gateway Demo



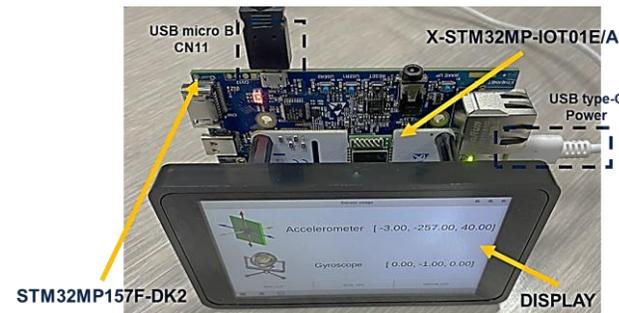
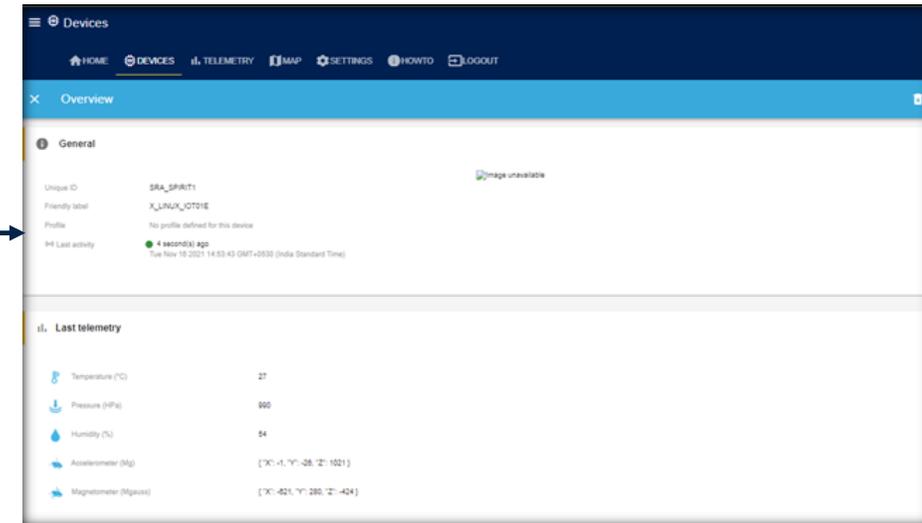
Sensor Node

Sub1-GHz



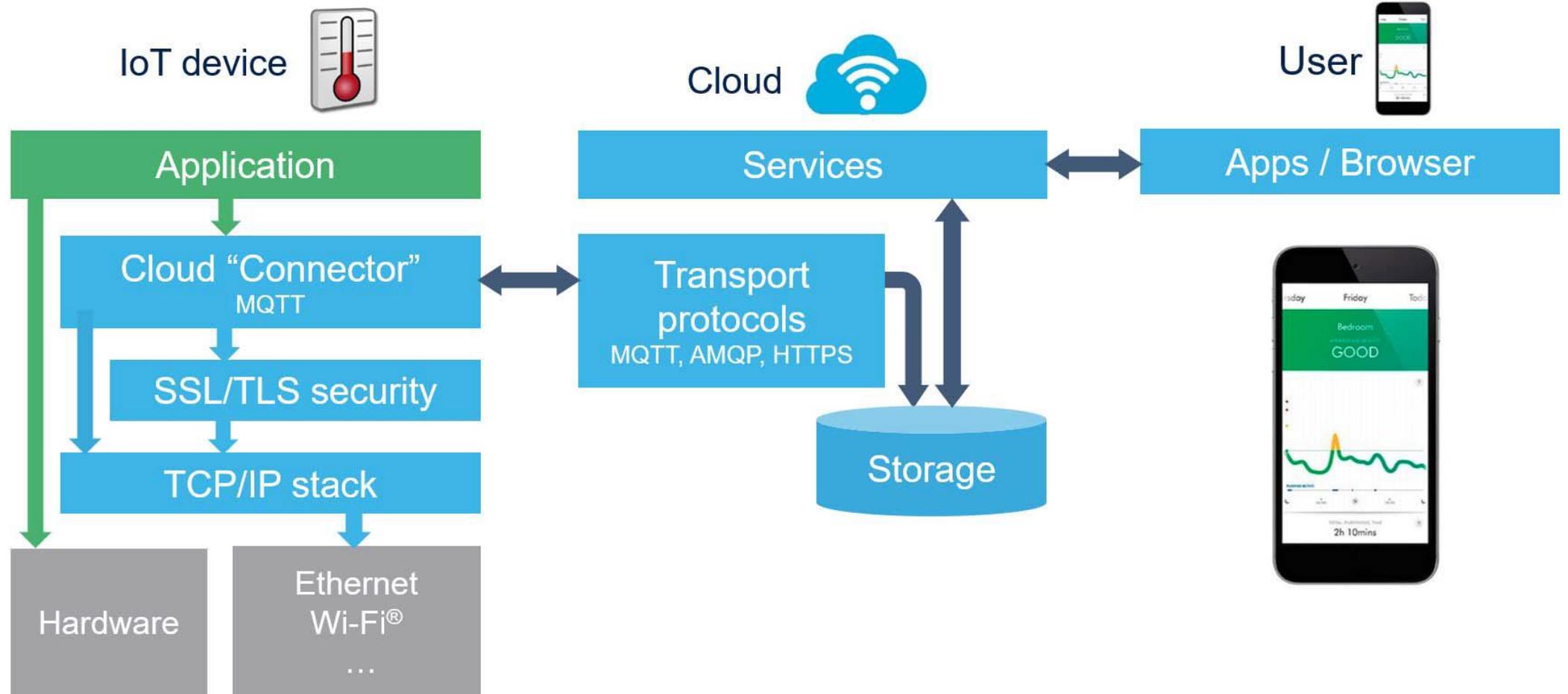
MPU based Gateway

Wi-Fi



Azure IoT Ecosystem

From device to remote user at any location



Our technology starts with You



Find out more at www.st.com

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.



life.augmented