# Qt for Mobile Applications

# Mobile Device Platforms

- Major mobile device platforms
  - Android™
  - iOS
  - Windows™

- Different platforms, different:
  - Skills, knowledge and experience
  - Development tools and ecosystems
  - Test and verification strategies
  - Maintenance cycles

- Features may drift over time between the device platforms

iOS

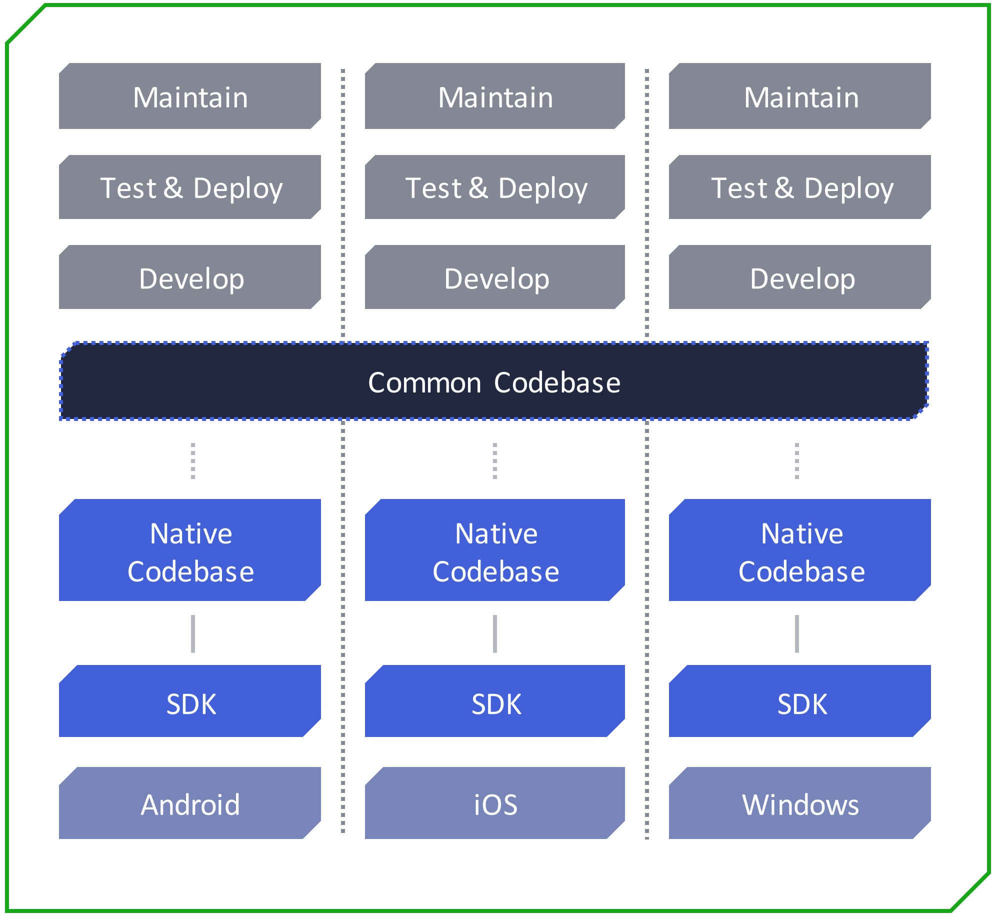
android

Windows

Android is a trademark of Google LLC. IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license. All other trademarks are the property of their respective owners.

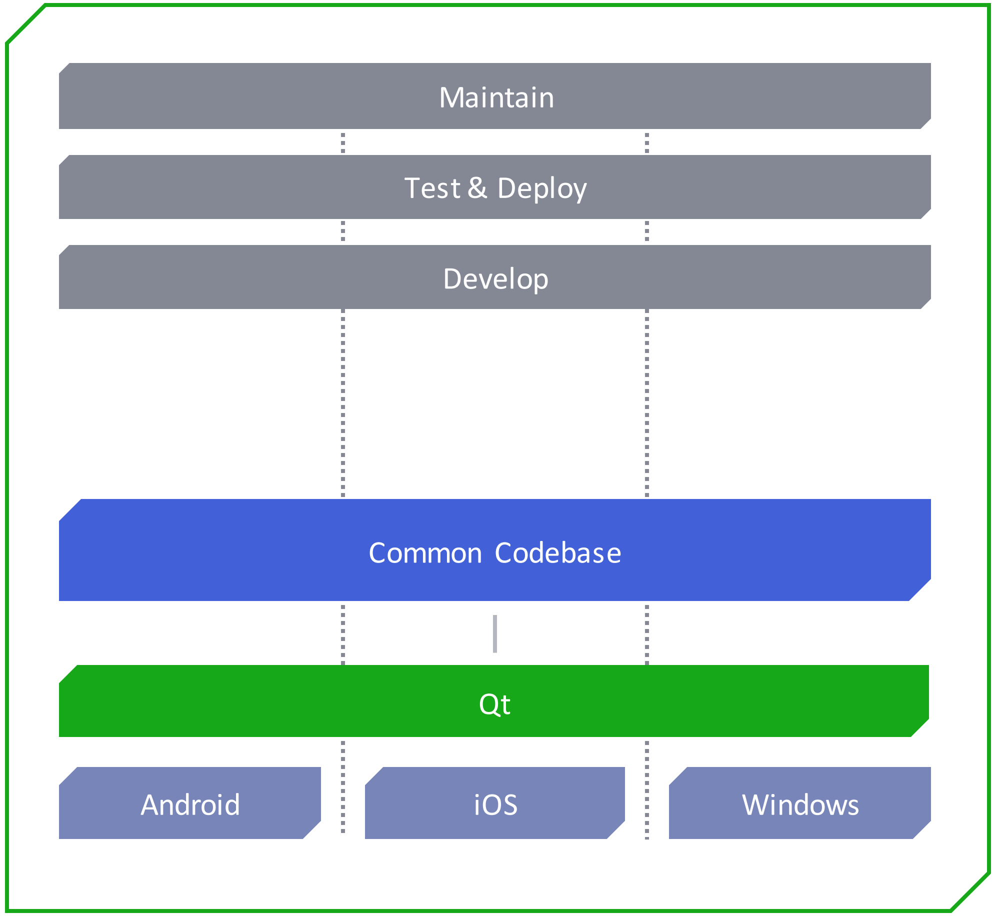
# Native Development

- Native development
  - Perfect native look and feel
  - Perfect integration with the device platform
  - Easy to conform to the platform’s human interface guidelines
- But there is a cost
  - Development
  - Testing and verification
  - Maintenance
- Total cost increases with the number of supported platforms



# The Qt Solution

- Reach all platforms
  - One common codebase
  - The same toolset and deployment process
- Simplify maintenance
- Get access to additional platforms
  - App can also be deployed to desktops, and even embedded hosts
- Develop for, and deploy to, all platforms
  - Use the same source code and the same tools
  - A single team can service all platforms



# Qt for Mobile – Value-Adding Features

UI Creation

Sensors

Location and Positioning

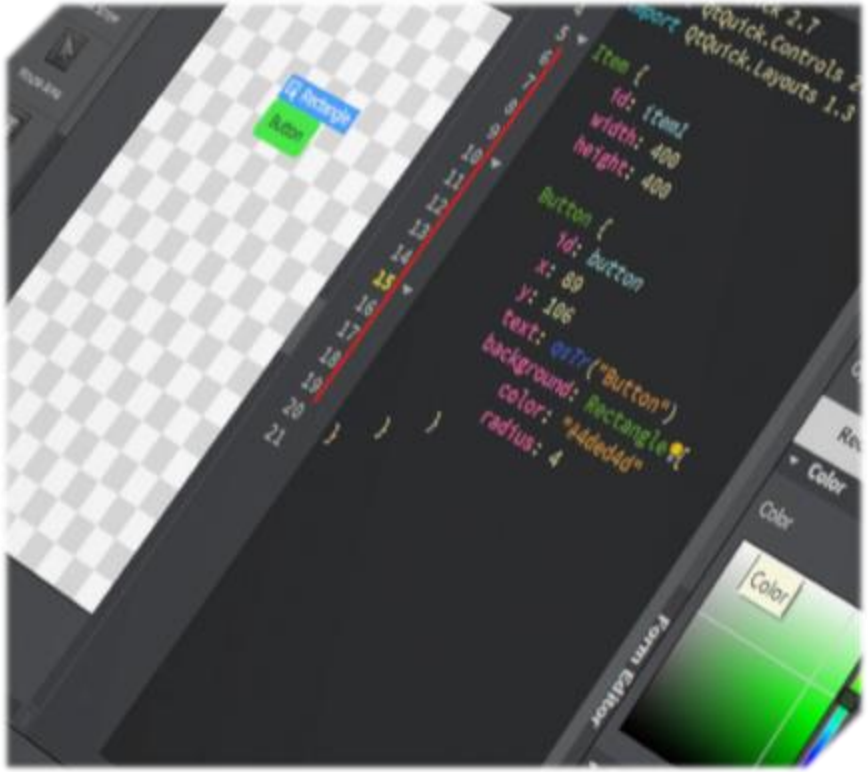
In-App Purchasing

Connectivity

Web Connectivity

# UI Creation

- Qt Quick Controls
  - Predefined controls
  - Can be styled and customized
  - Several predefined styles
- QML
  - Powerful declarative GUI design language
- Touch and gesture inputs
- Native performance

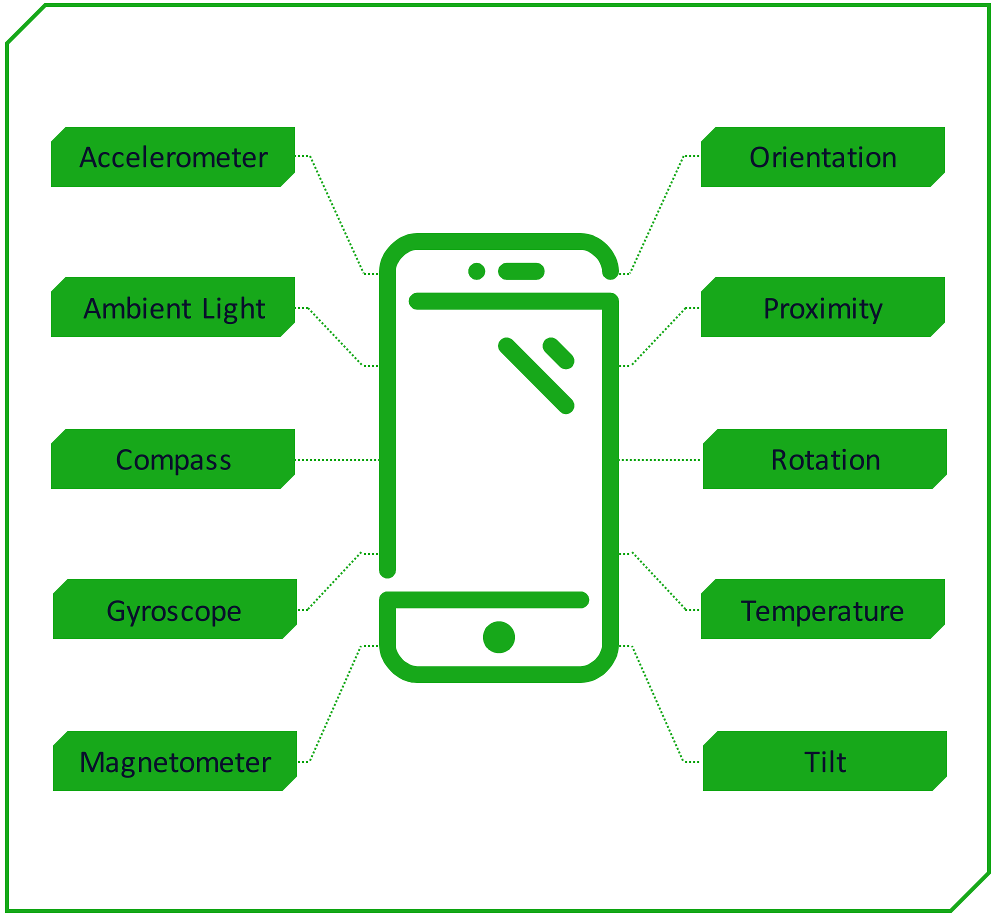


```
ck 2.7
port QtQuick.Controls 2
     QtQuick.Layouts 1.3

Item {
    id: item1
    width: 400
    height: 400

    Button {
        id: button
        x: 89
        y: 106
        text: qsTr("Button")
        background: Rectangle {
            color: "#4ded4d"
            radius: 4
        }
    }
}
```

# Sensors

- Unified access to sensors
  - Same code for all platforms

- Handle the actual sensors on the actual device
  - Not all devices have all sensors
  - Qt will emulate missing sensors where possible



Accelerometer

Ambient Light

Compass

Gyroscope

Magnetometer

Orientation

Proximity

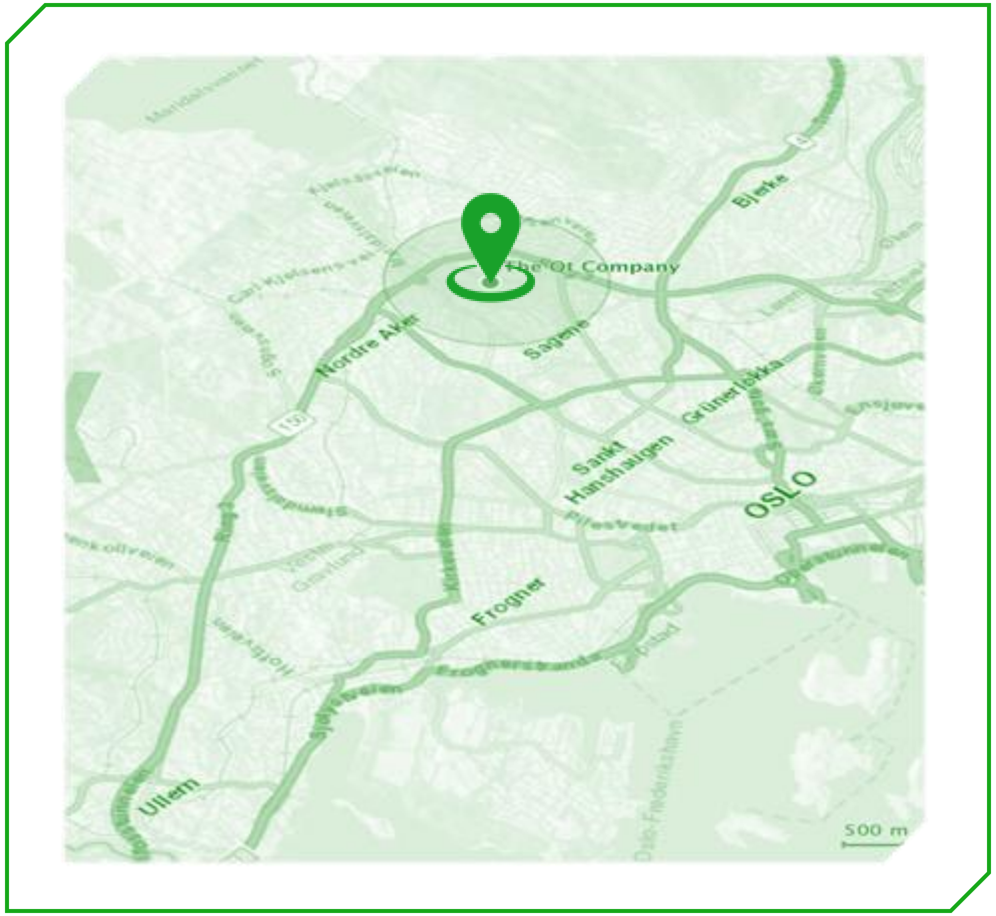
Rotation

Temperature

Tilt

# Location & Positioning

- Device-platform agnostic access to device’s:
  - Longitude – latitude – altitude (location)
  - Velocity – bearing – timestamp of reported location
- Location determined using:
  - GPS – cellular network – wifi – other as supported by device
- Maps using plugins:
  - Esri – Open Street Maps – HERE – Mapbox
- Location and route queries

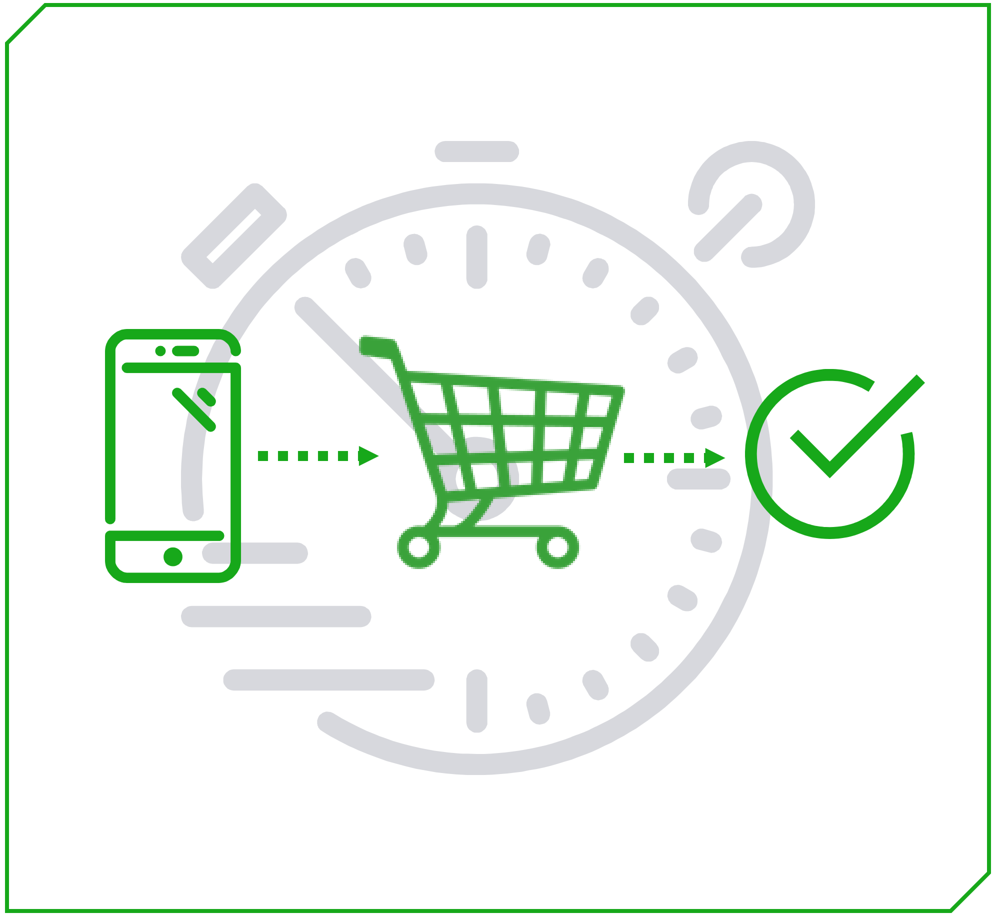


# In-app Purchases

- Device-agnostic access to:
  - Google Play
  - Apple App Store

- In-app purchases made easy
  - Same Qt API for stores

- Utilizes device platform’s APIs
  - Purchase process familiar to device platform
  - Access to stored purchasing information

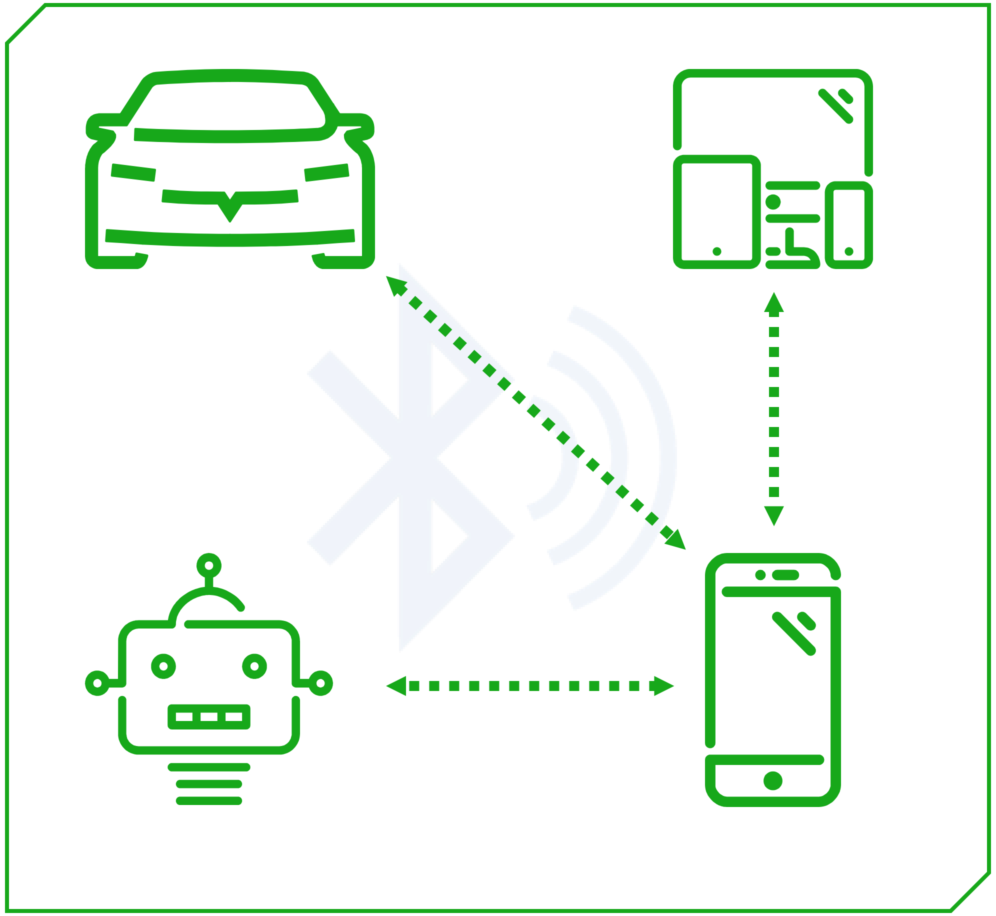


# Connectivity

- Classic Bluetooth

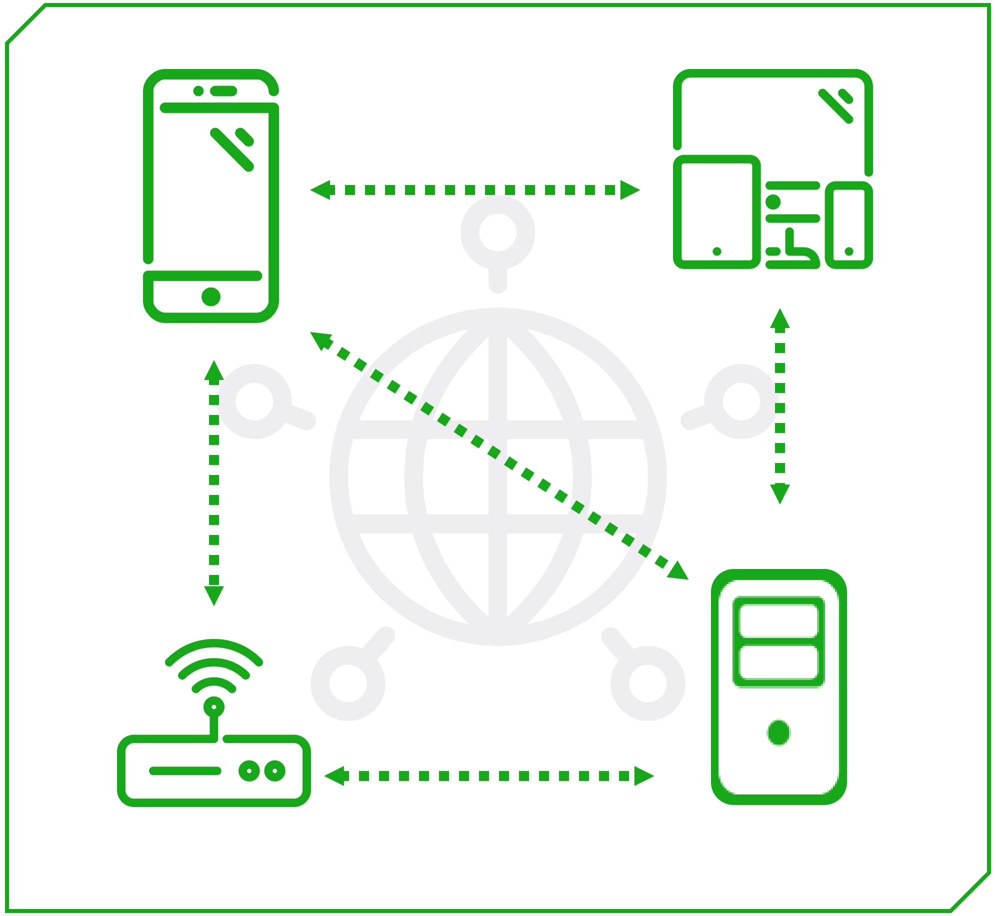
- Bluetooth Low Energy
    - Central device
    - Peripheral device

- Implement servers and clients

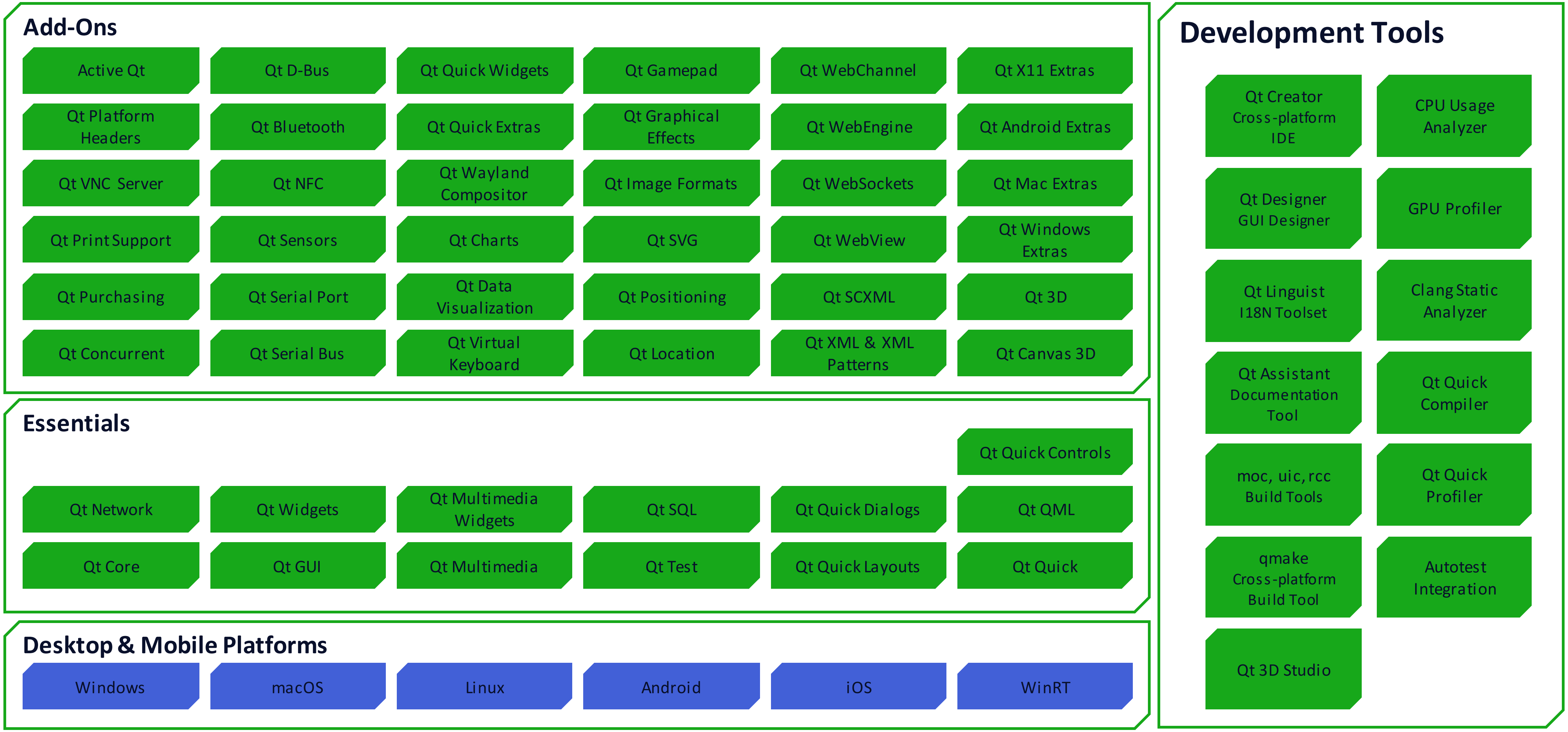


# Web Connectivity

- Qt WebSockets
  - Cross-platform socket support
  - Connect to remote services (REST, SOAP, etc.)
- Qt WebChannel
  - Cross-platform peer-to-peer support
  - Implement server/client communication
- Security
  - OAuth and OAuth2
  - Support authentication and security



# Qt – The Complete Picture



## Add-Ons

| | | | | | |
|---|---|---|---|---|---|
| Active Qt | Qt D-Bus | Qt Quick Widgets | Qt Gamepad | Qt WebChannel | Qt X11 Extras |
| Qt Platform Headers | Qt Bluetooth | Qt Quick Extras | Qt Graphical Effects | Qt WebEngine | Qt Android Extras |
| Qt VNC Server | Qt NFC | Qt Wayland Compositor | Qt Image Formats | Qt WebSockets | Qt Mac Extras |
| Qt Print Support | Qt Sensors | Qt Charts | Qt SVG | Qt WebView | Qt Windows Extras |
| Qt Purchasing | Qt Serial Port | Qt Data Visualization | Qt Positioning | Qt SCXML | Qt 3D |
| Qt Concurrent | Qt Serial Bus | Qt Virtual Keyboard | Qt Location | Qt XML & XML Patterns | Qt Canvas 3D |

## Essentials

| | | | | | |
|---|---|---|---|---|---|
| | | | | | Qt Quick Controls |
| Qt Network | Qt Widgets | Qt Multimedia Widgets | Qt SQL | Qt Quick Dialogs | Qt QML |
| Qt Core | Qt GUI | Qt Multimedia | Qt Test | Qt Quick Layouts | Qt Quick |

## Desktop & Mobile Platforms

| | | | | | |
|---|---|---|---|---|---|
| Windows | macOS | Linux | Android | iOS | WinRT |

## Development Tools

| | |
|---|---|
| Qt Creator<br>Cross-platform<br>IDE | CPU Usage Analyzer |
| Qt Designer<br>GUI Designer | GPU Profiler |
| Qt Linguist<br>I18N Toolset | Clang Static Analyzer |
| Qt Assistant<br>Documentation Tool | Qt Quick Compiler |
| moc, uic, rcc<br>Build Tools | Qt Quick Profiler |
| qmake<br>Cross-platform<br>Build Tool | Autotest Integration |
| Qt 3D Studio | |

# Qt Tools

## UI Development

- **Qt Quick Designer**
  GUI Designer
- **Qt Designer**
  GUI Designer
- **Qt 3D Studio**

## Localization

- **Qt Linguist**
  **I18N Toolset**

## Development

- **Qt Creator**
  Cross-platform IDE
- **Qt Assistant**
  Documentation Tool
- **moc, uic, rcc**
  Build Tools
- **qmake**
  Cross-platform Build Tool
- **Qt Quick Compiler**
- **Autotest Integration**

## Performance Optimization

- **CPU Usage Analyzer**
- **GPU Profiler**
- **Clang Static Analyzer**
- **Qt Quick Profiler**

# A Few of the Success Stories

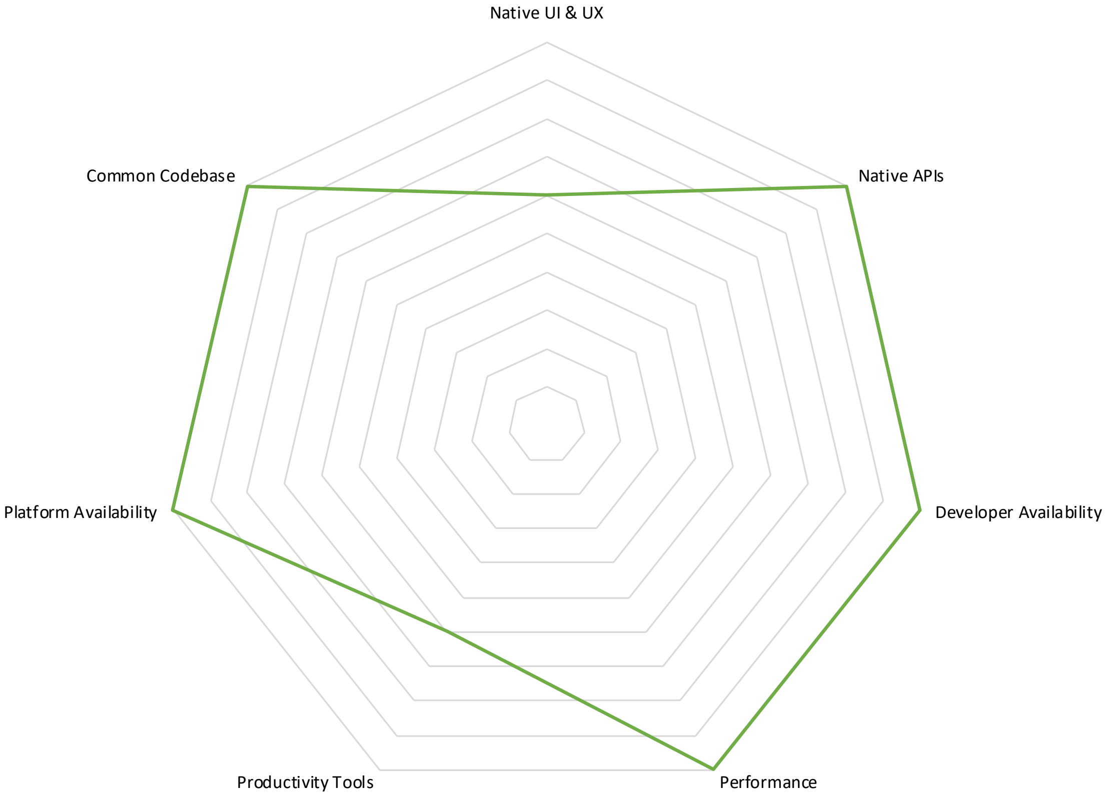
- MuseScore – Music Notation Software
- Imaginando – DRC synthesizer
- eyeMaps – 3D augmented reality maps
- Devinco – Admin without paperwork
- Canonical – Ubuntu Linux

© The Qt Company

# Comparable Technologies

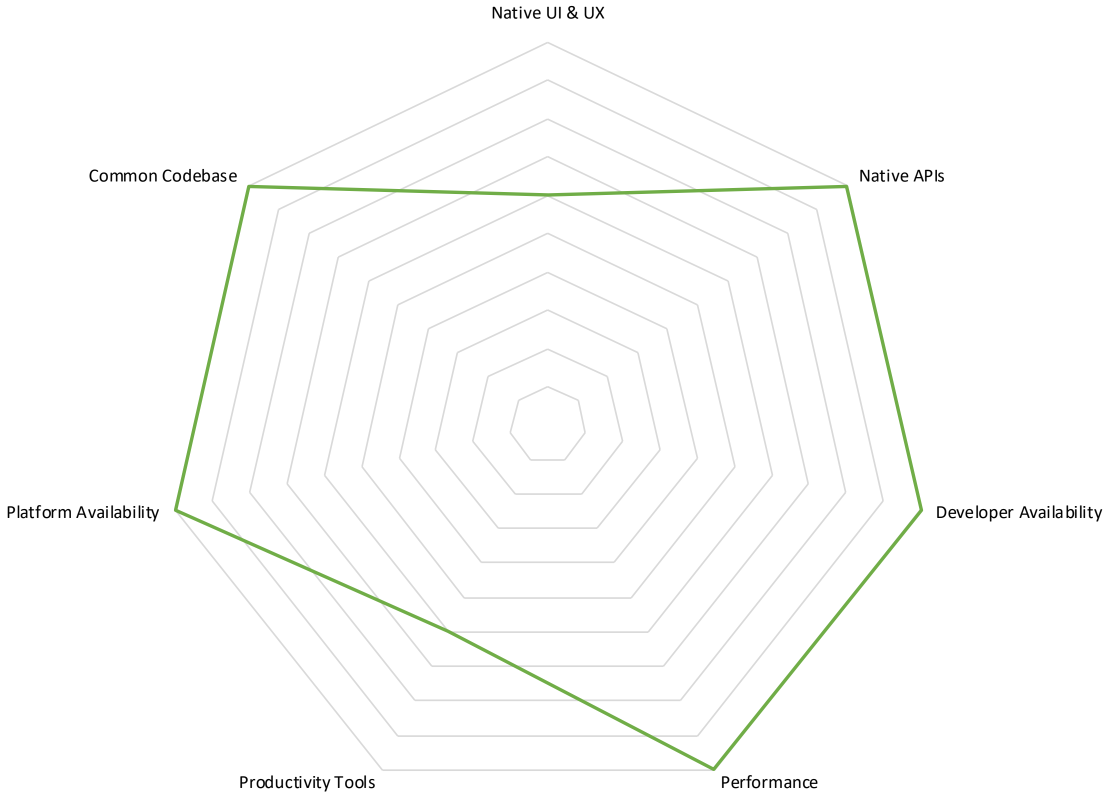
# React Native

| + | ÷ |
|---|---|
| Native UI and UX | Lower performance |
| Declarative UI | Only Android and iOS |
| Javascript | |
| Great developer tools | |



# Xamarin

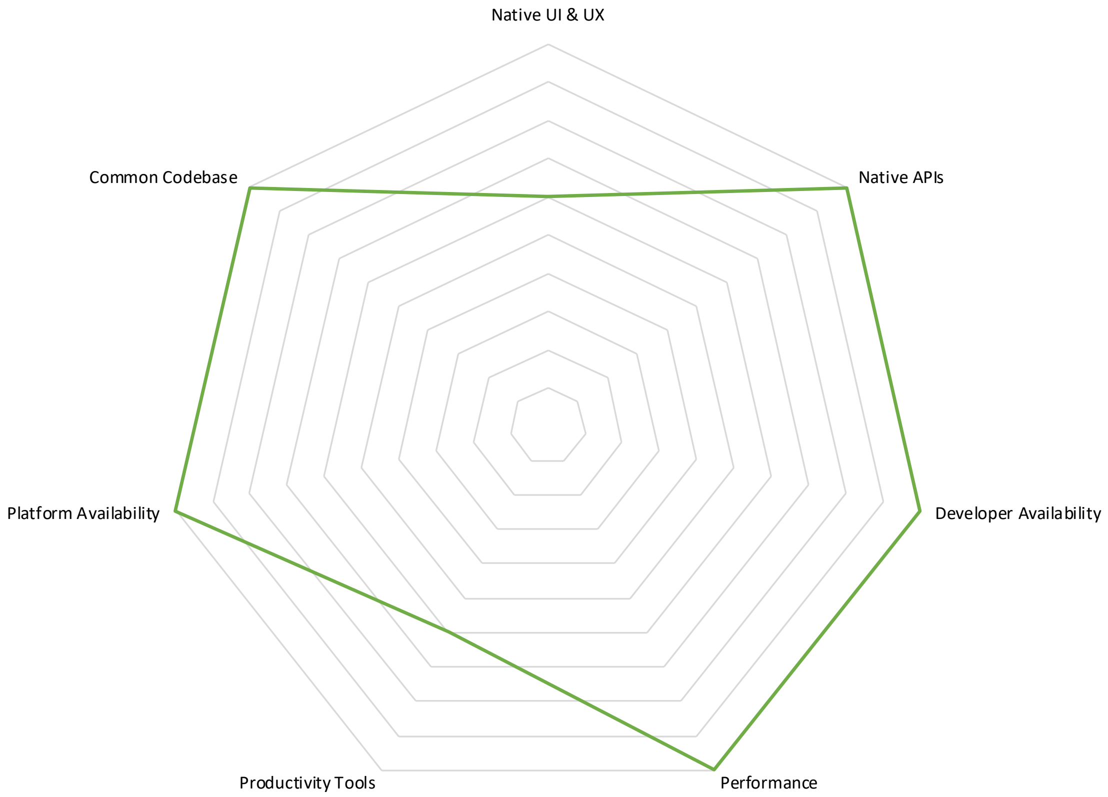
| ✚ | ÷ |
|---|---|
| Native performance | Not 100% shared codebase |
| Native UI and UX | |
| Many third-party libraries | |
| Power of Visual Studio | |
| C# - many developers | |



© The Qt Company

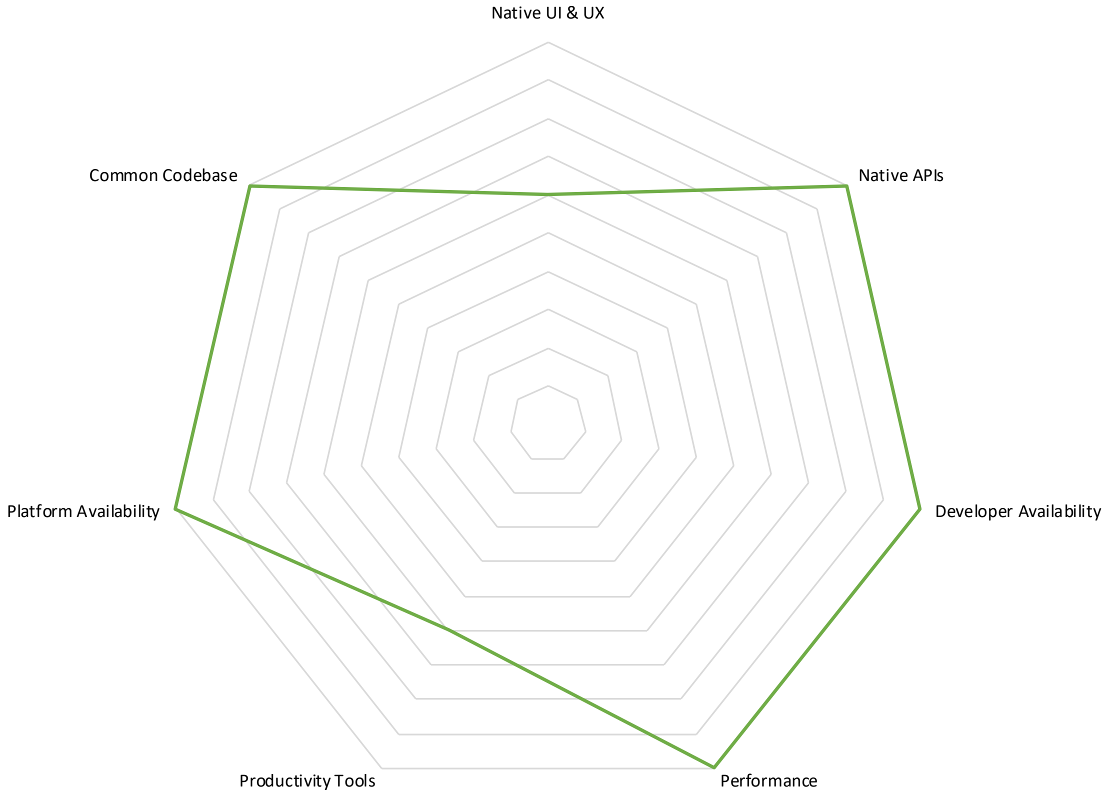
# Flutter

| + | ÷ |
|---|---|
| Native performance | Dart – fairly new language |
| Great developer tools | Only Android and iOS |



# Ionic2

| + | ÷ |
|---|---|
| Rapid prototyping | Lower performance |
| Desktop, mobile and web | Poor code binding capabilities |
| Angular JS, HTML & CSS | |
| Native UI and UX | |
| Good developer tools | |



© The Qt Company

# Wrap-up

- Qt offers:
  - True cross-platform framework
    - Also desktop environments
  - Powerful development tools
  - Improved efficiency and reduced time-to-market

- Qt is perfect if you:
  - Have a smaller development team
  - Deploy to more than one mobile device platform
  - Require high performance and stability

https://www.qt.io/

https://www.qt.io/mobile-app-development/