



Qt for MCUs

Code Once. Deploy Everywhere.

www.qt.io

What is Qt?

- › C++ based cross-platform application framework
- › Provides a suite of libraries (over 70) and design and developer tools - <https://www.qt.io/features>
- › Popular as GUI framework for embedded Linux systems
 - › Available for various popular MPUs from NXP, Renesas, ST, Nvidia, Intel
- › Now also available for MCUs

DEVELOPEX

Source: DevelopEX Blog

DevelopEX Blog Why Qt for UI Development?

Qt is de-facto the most suitable framework for the commercial application of a cross-platform GUI library available for C++, Python, Go, Haskell and some other languages.

[Read full article »](#)

[DevelopEX's Competitors](#) | [DevelopEX's News](#) | [DevelopEX's Financials](#)

The Qt Company around the world

2 M +

Developers

70 +

Industries

3 500 +

Commercial Customers



The Qt Group offices are located in Finland, China, Germany, Japan, South Korea, Norway, the USA, France and India. The HQ is in Espoo, Finland.

BUILT WITH Qt



2019 CUSTOMER SURVEY

95%

ROI expectations exceeded

70%

find Qt easy to use

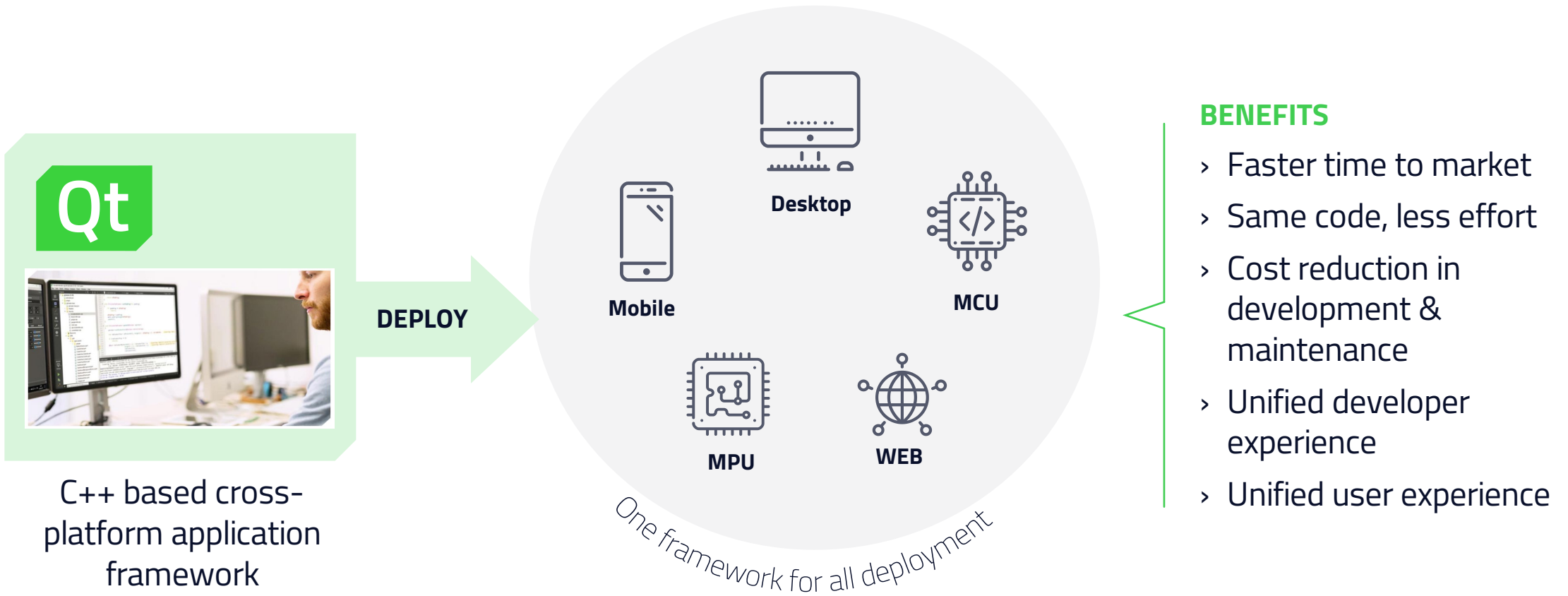
80%+

are more productive with Qt



Code once, deploy everywhere with Qt

Unified developer experience across all production line

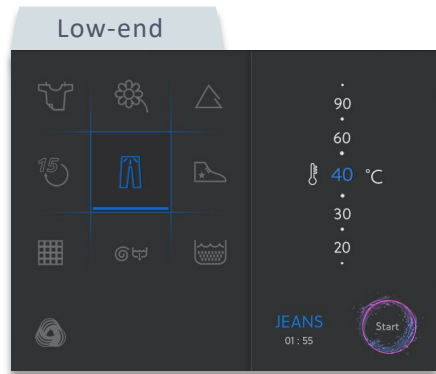


C++ based cross-platform application framework

SINGLE CODEBASE

Cross product-line development

Retain a unified look & feel for your pixel-perfect UIs across an entire range of products, using the same core technology, and without increasing TTM or TCO.



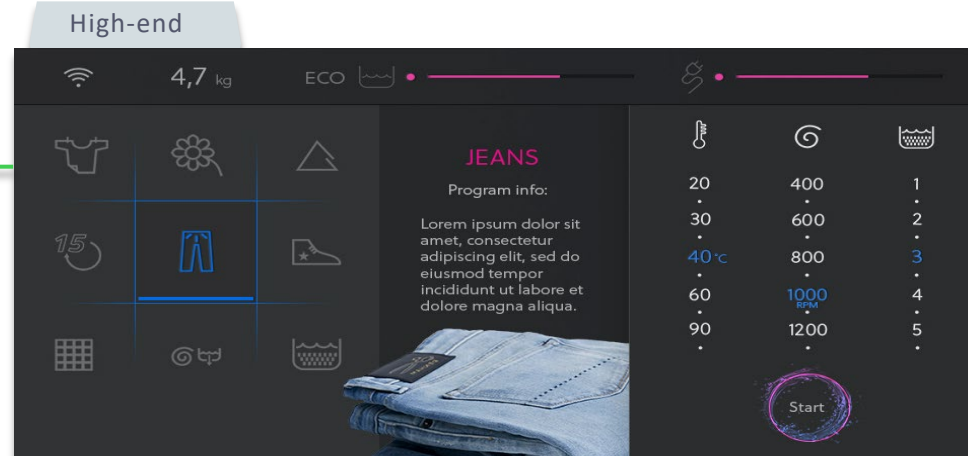
Low-end MCU – example resolution: 320x240

- ✓ Qt for MCUs
- ✓ Smartphone-like UX
- ✓ Basic animations
- ✓ Bare metal or freeRTOS



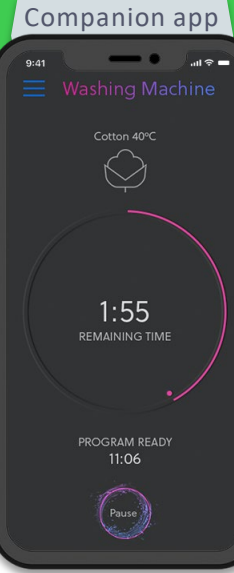
High-end MCU or low-end MPU – example resolution: 840x480

- ✓ Higher resolution
- ✓ 2.5D Graphics
- ✓ Full Qt Framework
- ✓ Advanced animations
- ✓ Linux or RTOS



High-end MPU – example resolution: 1920x720

- ✓ Highest resolution
- ✓ Dual screen support
- ✓ 2D/3D Graphics
- ✓ Full Qt Framework
- ✓ Linux or RTOS
- ✓ Video



- ✓ Complex/simple apps
- ✓ Win, Mac, Linux, Android, iOS
- ✓ WEBASM



Remote UIs



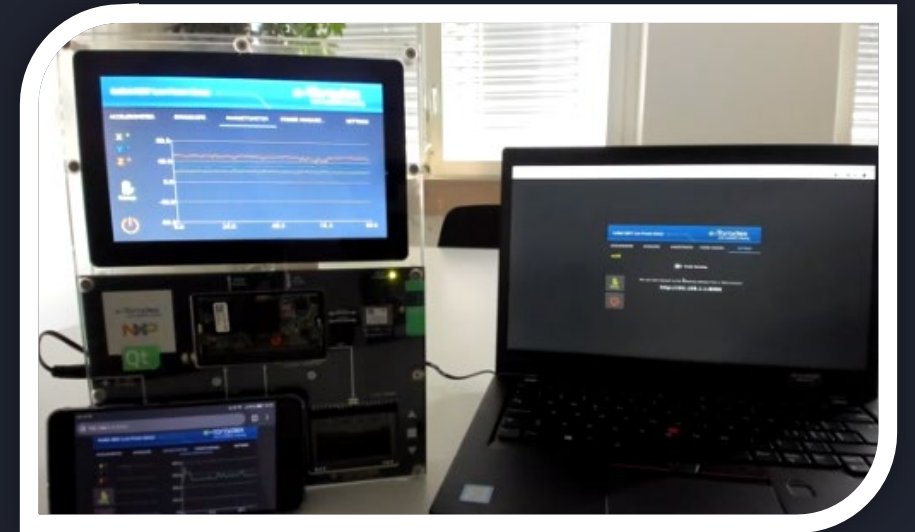
Qt Quick WebGL

- › Run the UI in a local or remote display
 - › No re-compilation needed, just a plugin change
- › Stream the Qt Quick UI over the network
- › Show the UI in WebGL-enabled browser in a remote device



Qt for WebAssembly

- › Build your applications for WebAssembly
- › Run in the browser
- › Zero installation
- › Easy to distribute to any device with a browser



Design



Interaction Designers

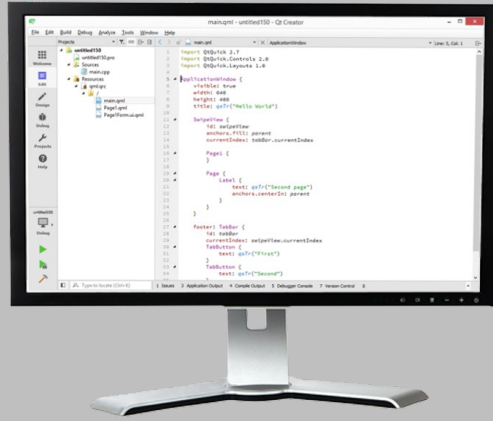
UI flow & navigation
Wireframes



Sensorial Designers

Visual assets
Motion designs
Audio assets

Develop



Client Developers

Custom UI components
Data connections
Back-end logic

Deploy



Testers & Management

Rapid iterations
Visuals review
Performance / UX testing

OUR VISION

Collaborative product development



The best innovations are crafted through team collaboration and rapid iterations.

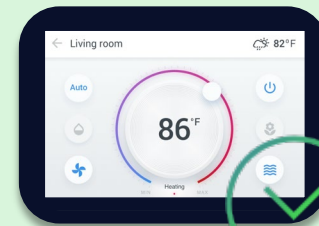


The designs must be testable on real hardware in all phases in order to validate the feasibility of the designs.



In minutes

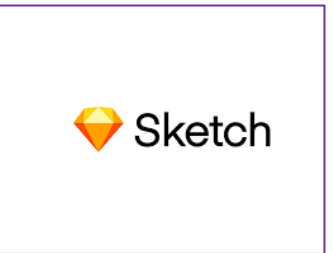
Maintain



Cross Platform Maintenance

Desktop / Embedded / Mobile
MCU up to High Spec HW

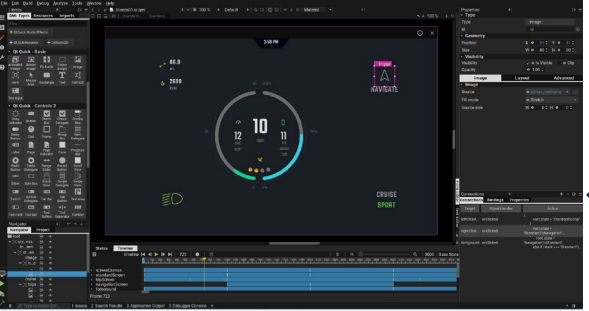
Qt Toolchain From Design to Deploy



UI Asset
↔

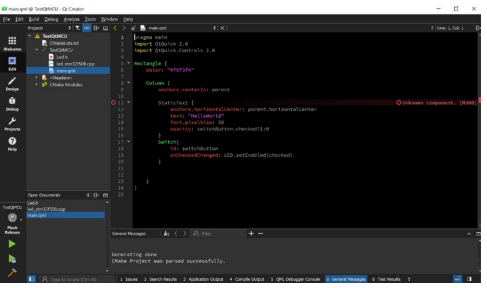
DS Qt Design Studio

Qt Template

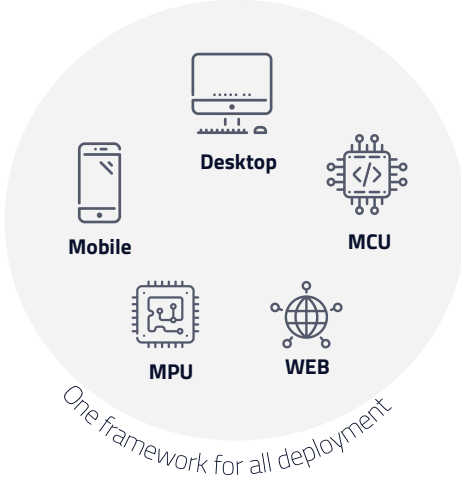


QML
↔

C Qt Creator



Deploy →

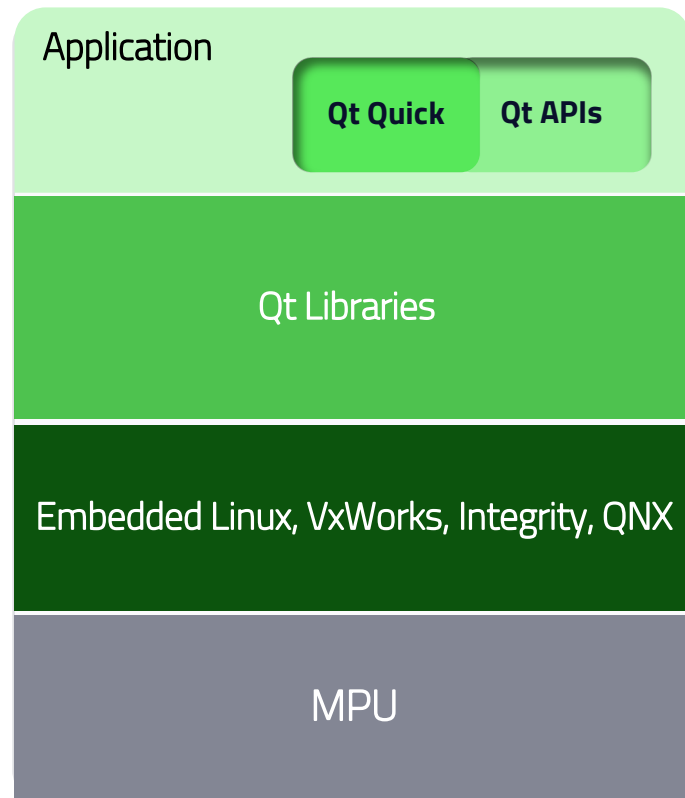


Live Preview



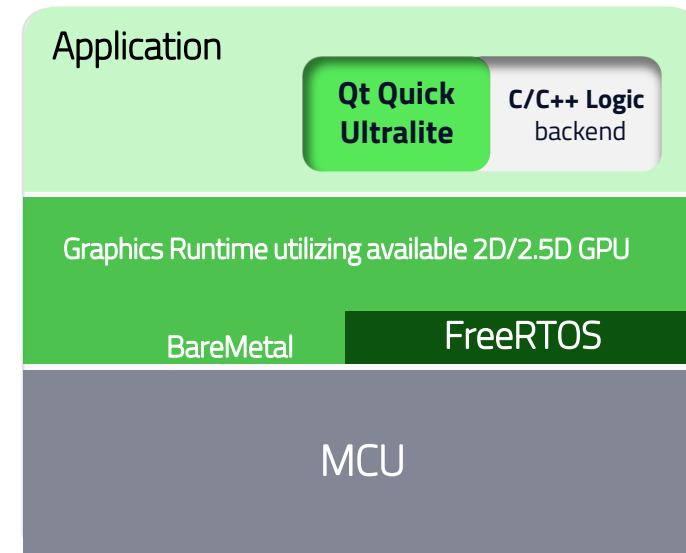
Qt on MPUs / MCUs

Qt for Device Creation



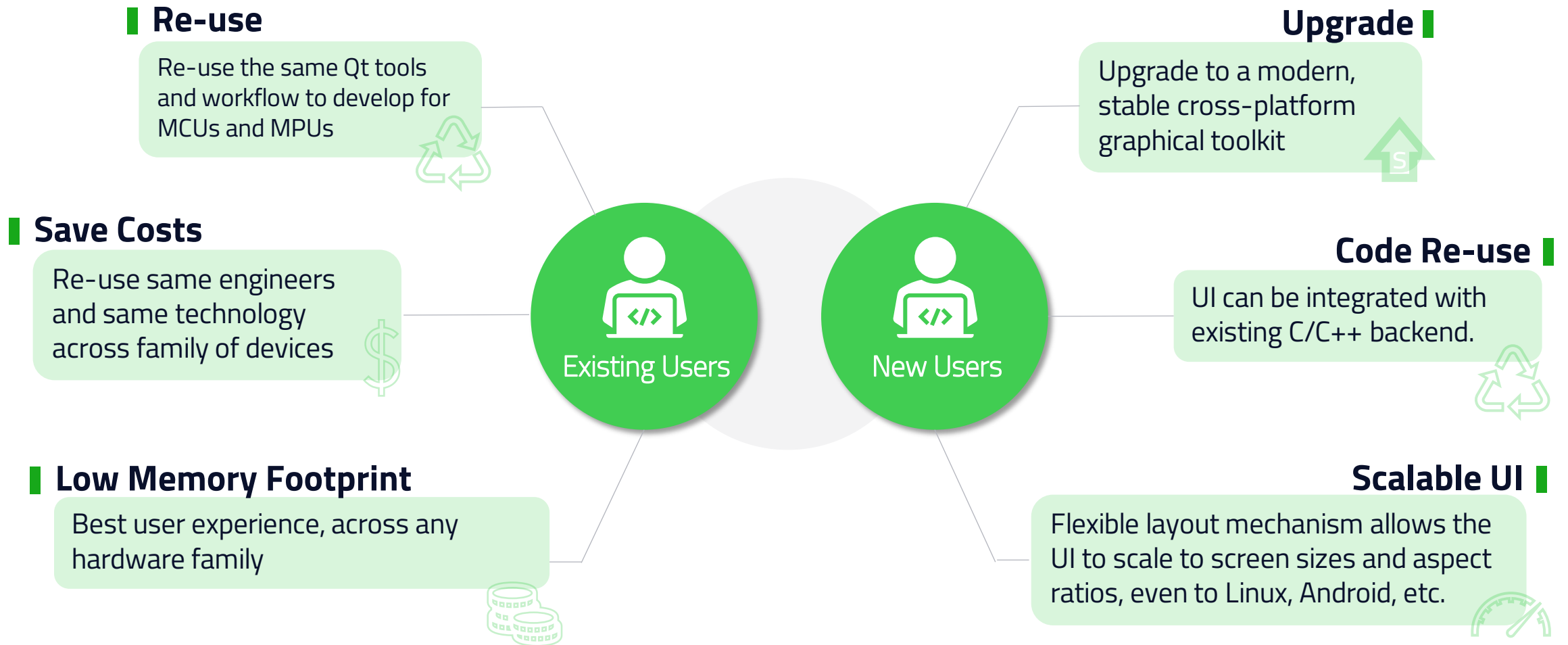
Qt for Device Creation is a commercial offering that provides the Qt development framework for multiple embedded platforms: [embedded Linux](#), [QNX](#), [INTEGRITY](#), and [VxWorks](#).

Qt for MCUs



Qt for MCUs is a software toolkit to create high performance graphical user interfaces with low memory consumption on MCUs running either Bare Metal or RTOS such as FreeRTOS

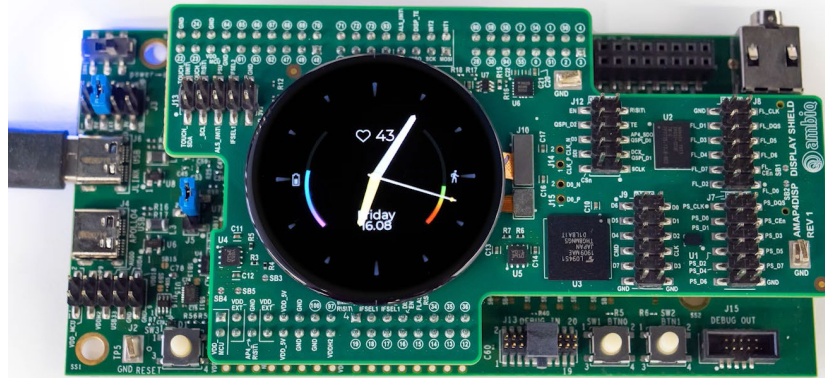
Why use Qt across MCU and MPU?



Qt for MCUs on Ambiq Apollo4

› Low Power device on Qt for MCUs 2.0

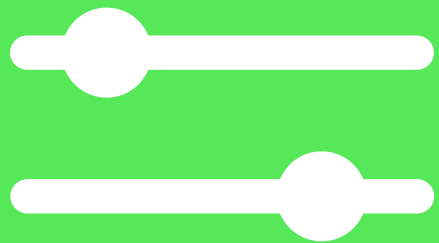
: Showcases wearable application, 2.5D GPU usage



Qt for MCUs on Apollo4 EVB
Rev.3 with Display Shield

Screen Resolution: 454x454
Color: 16 bit (RGB565)
Application Code Size: 356KB
Frame Buffer: 455KB
RAM: 86KB
RAM with Assets: 372KB

Key Metrics		Ambiq Apollo4
Display	Resolution	390x390px
	Pixel Depth	16-bit color
RAM Usage	Qt runtime	163 kB
	Framebuffer (single buffering)	304 kB
	Total	467 kB
Flash Usage	Application	420 kB
	Assets	389 kB
	Total	809 kB
Frame rate	Average	40



UX

Provide a smartphone-like user experience with Qt Quick Control



Reuse

Reuse source code across ARM architectures



Speed Up

Fast, effective development with QML and Qt Tools

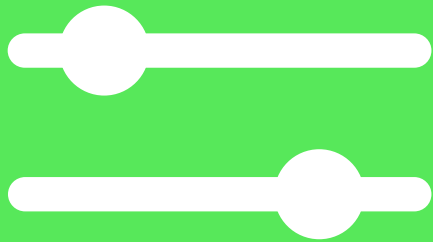


Community

Benefit from a large developer community

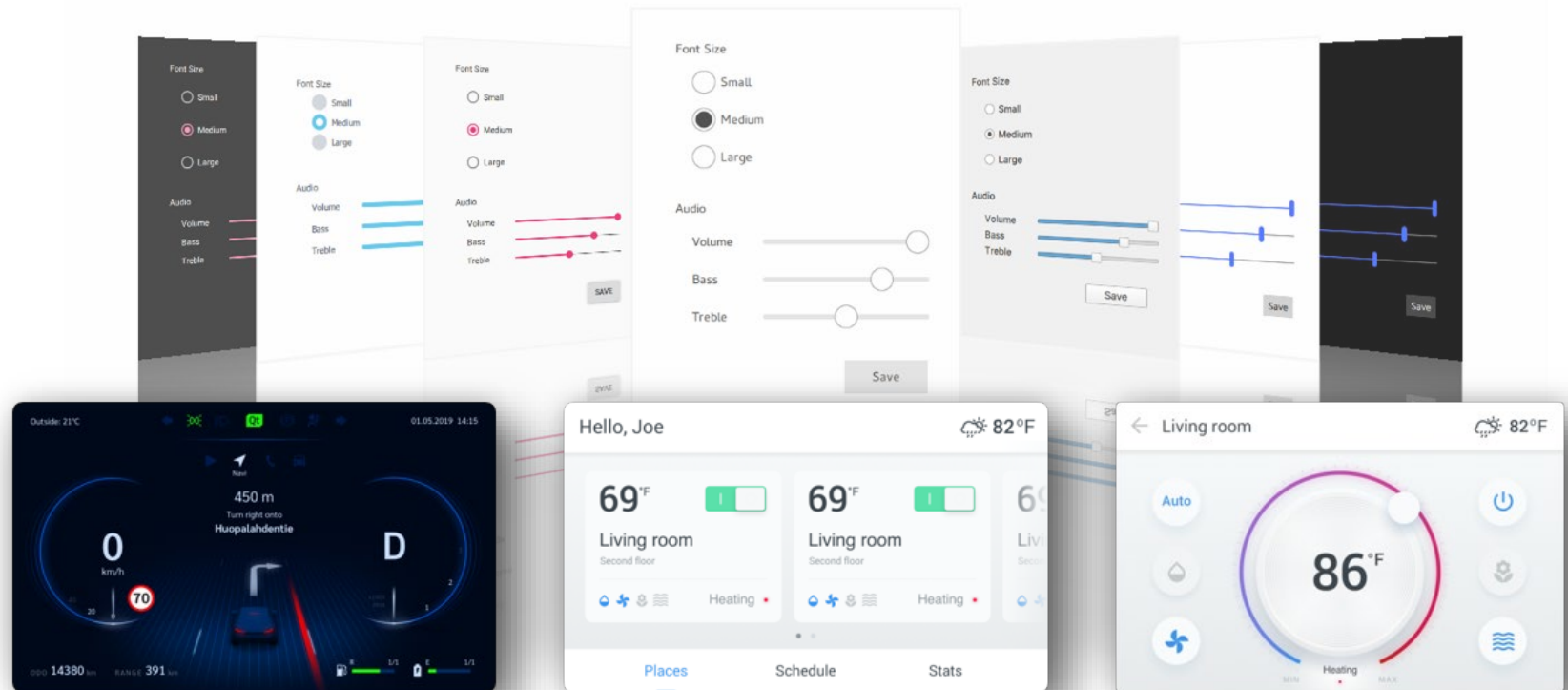
Qt Quick Control Provides Mobile-like User Experience

Build and deploy complete interface with rich library of UI controls



UX

Provide a smartphone-like user experience with Qt Quick Control



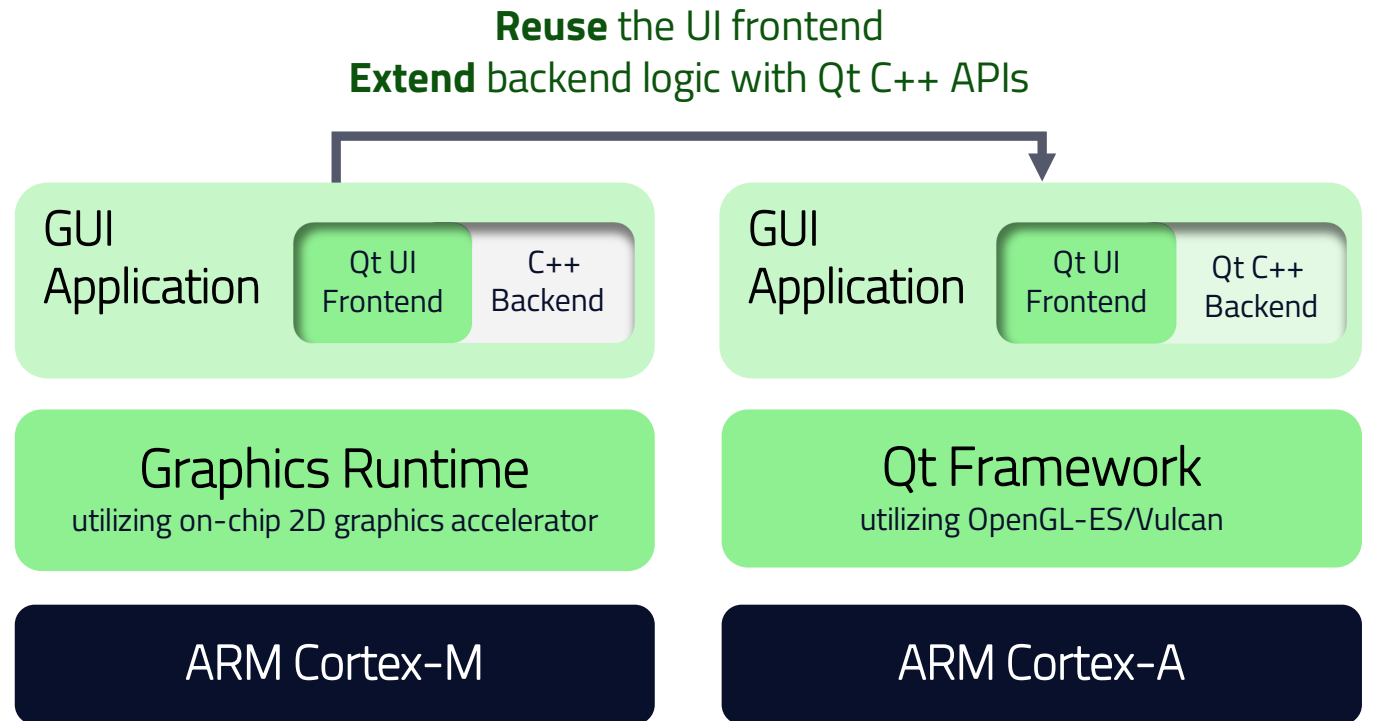


Reuse

Reuse source code
across ARM
architectures

Graphic Reuse on Powerful Platforms

Code Once, Deploy Everywhere





Speed Up

Fast, effective
development with
QML and Qt Tools

Boost your process by QML and Qt Tools

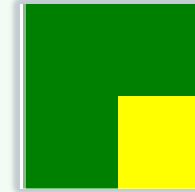
Easy, intuitive QML for UI frontend, C++ logic backend and straightforward tools simplify development process.

Boost by QML

QML is declarative language, objective code, compiled to a binary machine code

```
Rectangle {  
  x: 50  
  y: 50  
  width: 100  
  height: 100  
  color: "green"  
}
```

```
Rectangle {  
  x: 100  
  y: 100  
  width: 50  
  height: 50  
  color: "yellow"  
}
```



QML allows easy development process – JSON-like syntax

Boost by Tool

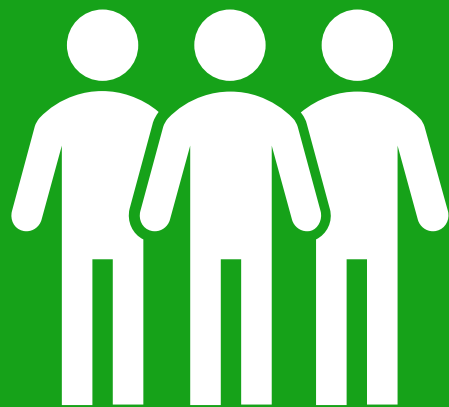
Designers produce QML based "UI Specification" directly usable by developers



Client Designers
UI flow & navigation
Wireframes
Visual assets



Client Developers
Custom UI components
Data bindings
Application logic



Community

Benefit from a large
developer community

Reuse your Qt Skilled Engineers

Use your Qt engineers again along with the developers in open source community.



>1M

Developers using Qt

Sufficient number of available engineers

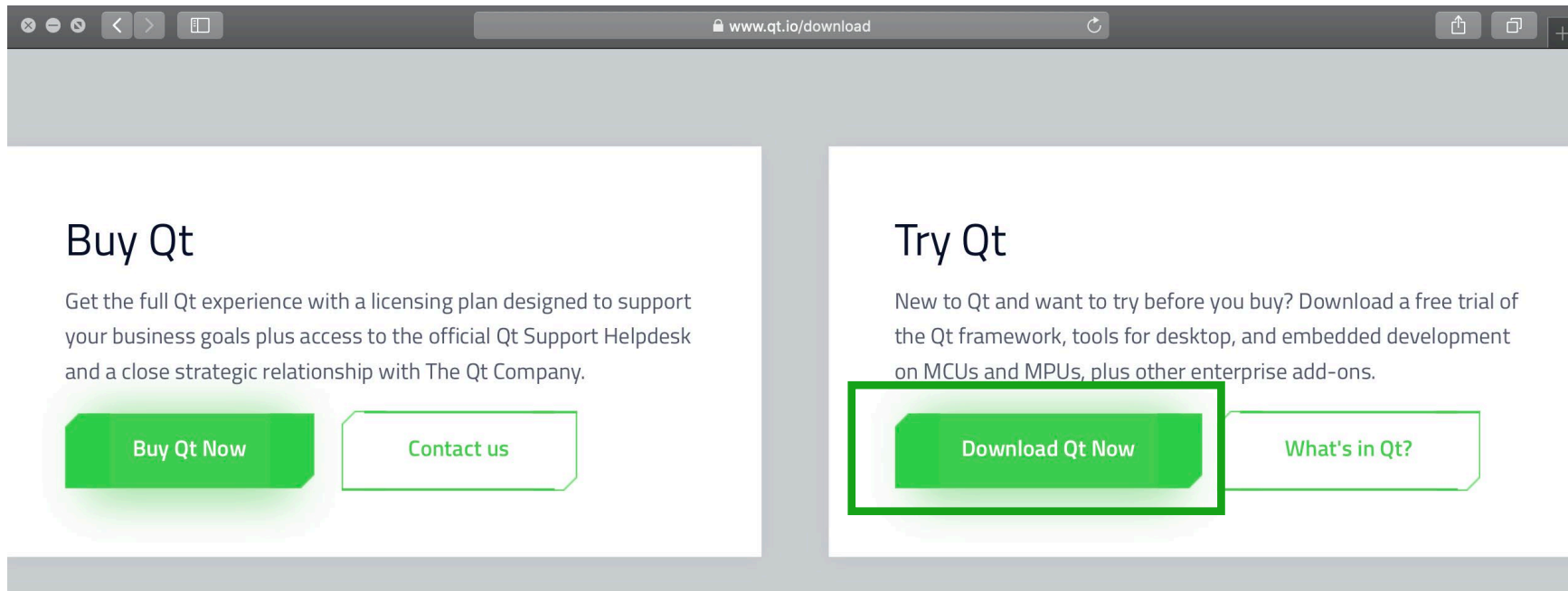
Well-cultivated open source community

Thanks to its roots in the Open-Source community, Qt constantly evolves through contributions from helpful developers around the world.

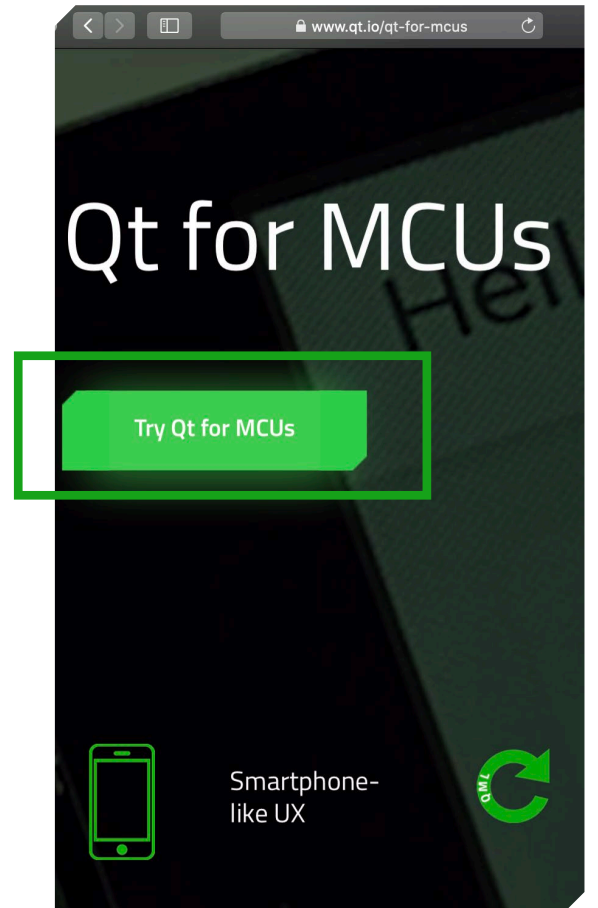
"What was amazing was that there was already a body of work done by the Qt Community. Had that open source community not been there we would have taken a much longer time to deliver."

Get Qt

www.qt.io/download



<http://www.qt.io/qt-for-mcus>



Qt Service Catalog

Qt Service Catalog

Adapt Qt Platform code to customer's setup/custom HW platform

Create Demo / PoC specific to customer's use case

Bug fixing / LTS on custom HW platform

Architecture advisory services

- covering Graphics Monitoring for Functional Safety
- covering optimal use of memory / CPU

Integration with external peripherals

Conduct Trainings / Workshops

Provide Turn-Key Solutions

Port to a niche RTOS



Thank you

The future is written with Qt

www.qt.io