Gear up to deliver world class UI/UX on Electric Vehicles & Medical Devices with Qt for MCUs

Presenters

Thilak Ramanna

Head of India, Australia and New Zealand

Sumitabh Ghosh

Presales & Solution Architect, India, Australia and New Zealand

Increasing Expectations on Hardware

The number of smart connected devices is predicted to grow significantly, and many of those devices are expected to have requirements below.

Real-time Processing

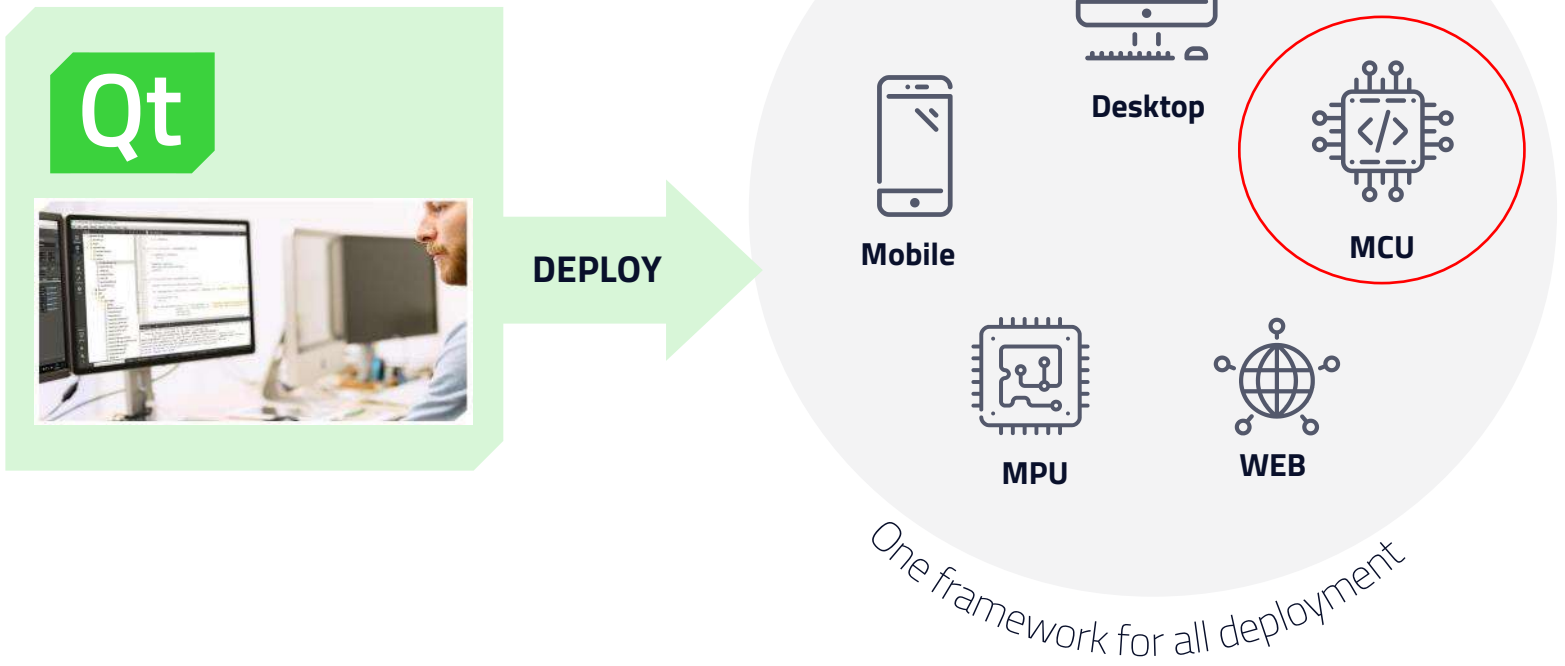
Low power consumption

Instant Boot time

Low BOM cost

Code once, deploy everywhere with Qt

Unified developer experience across all production line



BENEFITS

- Faster time to market
- Same code, less effort
- Cost reduction in development & maintenance
- Unified developer experience
- Unified user experience

Electric Vehicle Use Case

Instrument Cluster / Digital Cockpit

Charging Station

Mobile App

Medical Device Use Case

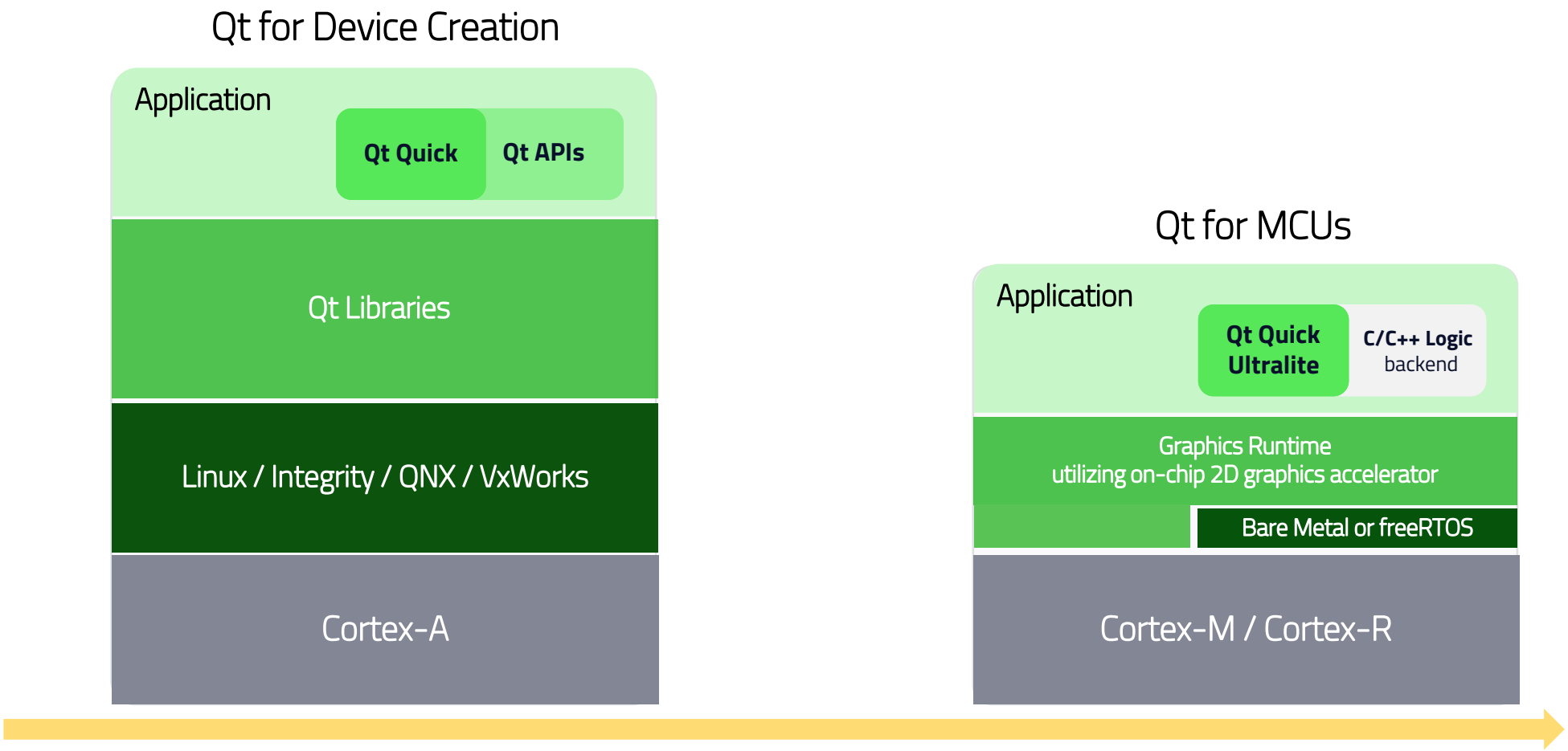
Anaesthesia Care

Portable Ultrasound

Skin Cancer Detector

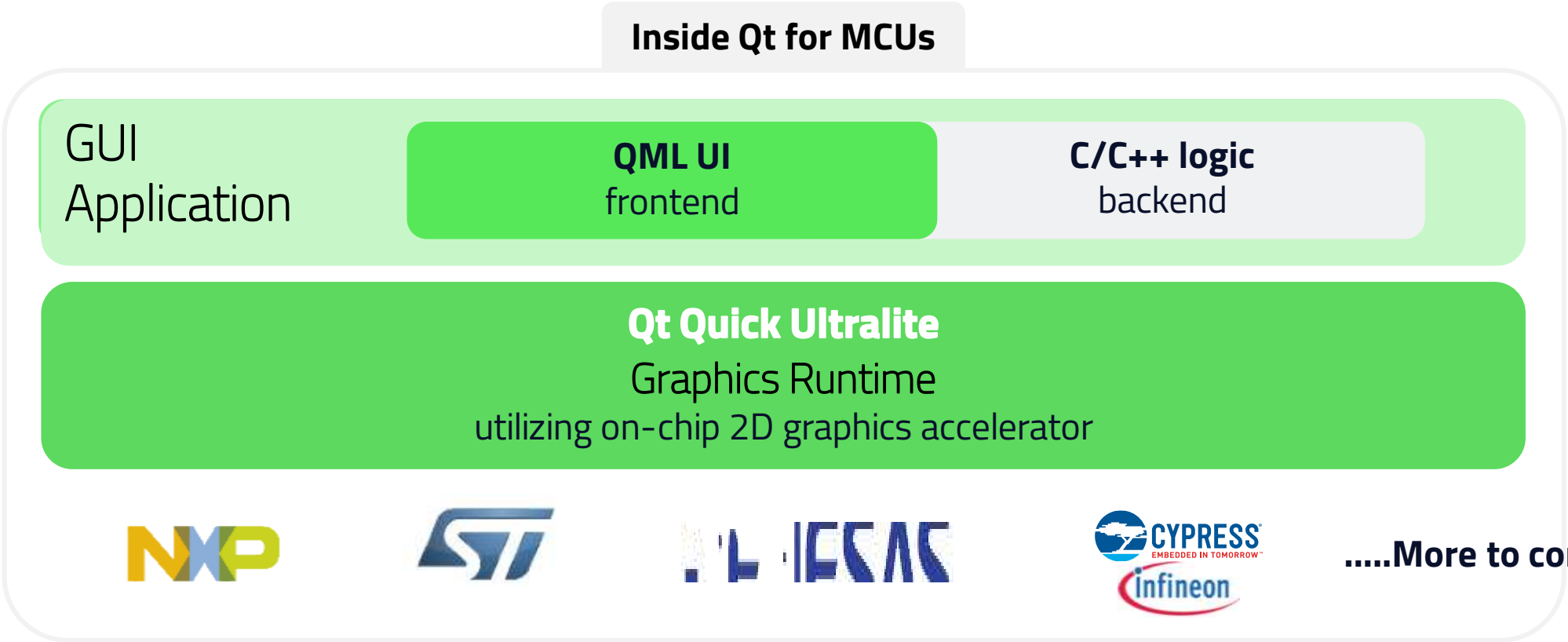
A NEW Endeavor in to address Scalability

Qt Quick Ultralite can be reused for your Qt for Device Creation application code



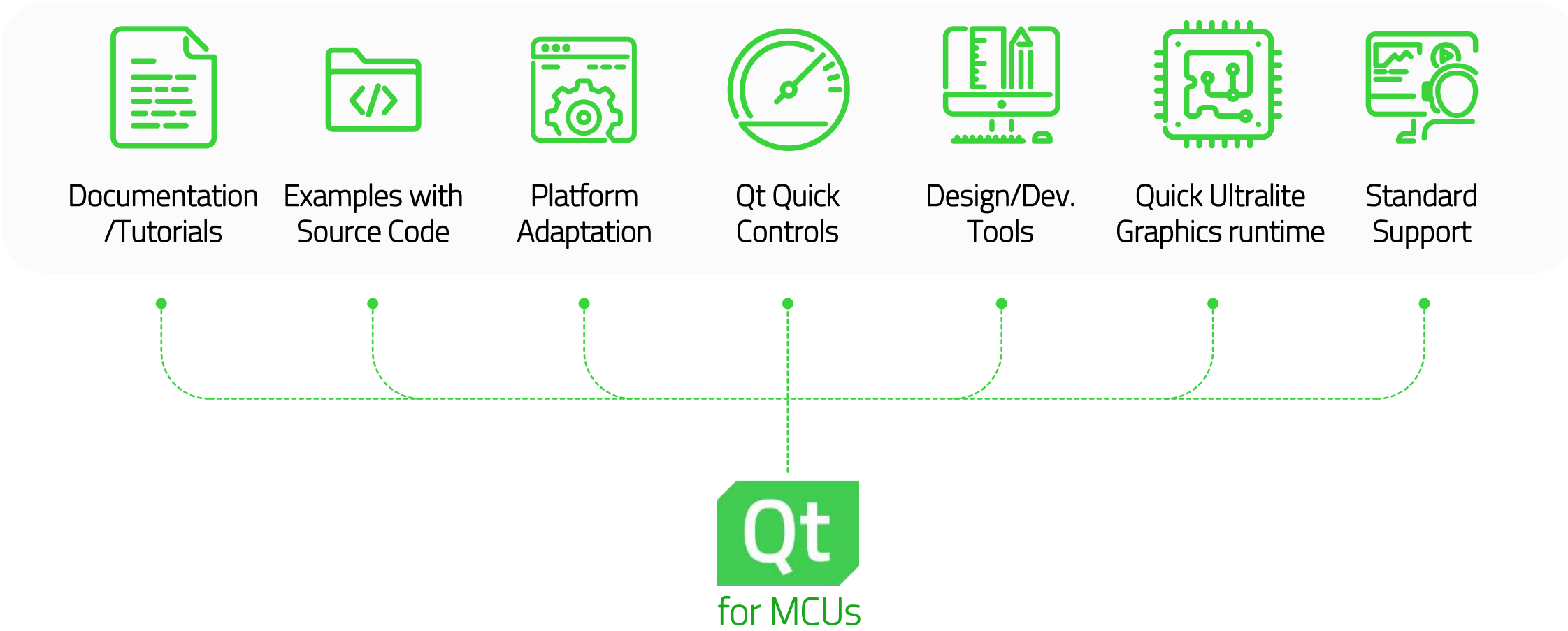
Qt for MCUs – *Ultimate performance, Tiny footprint*

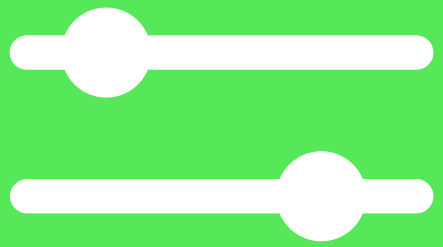
Qt for MCUs uses a new graphic runtime, Qt Quick Ultralite, that delivers high performance with low memory consumption, which is achieved by a new translation of QML to C++.



What is in the package?

Qt for MCUs provides:



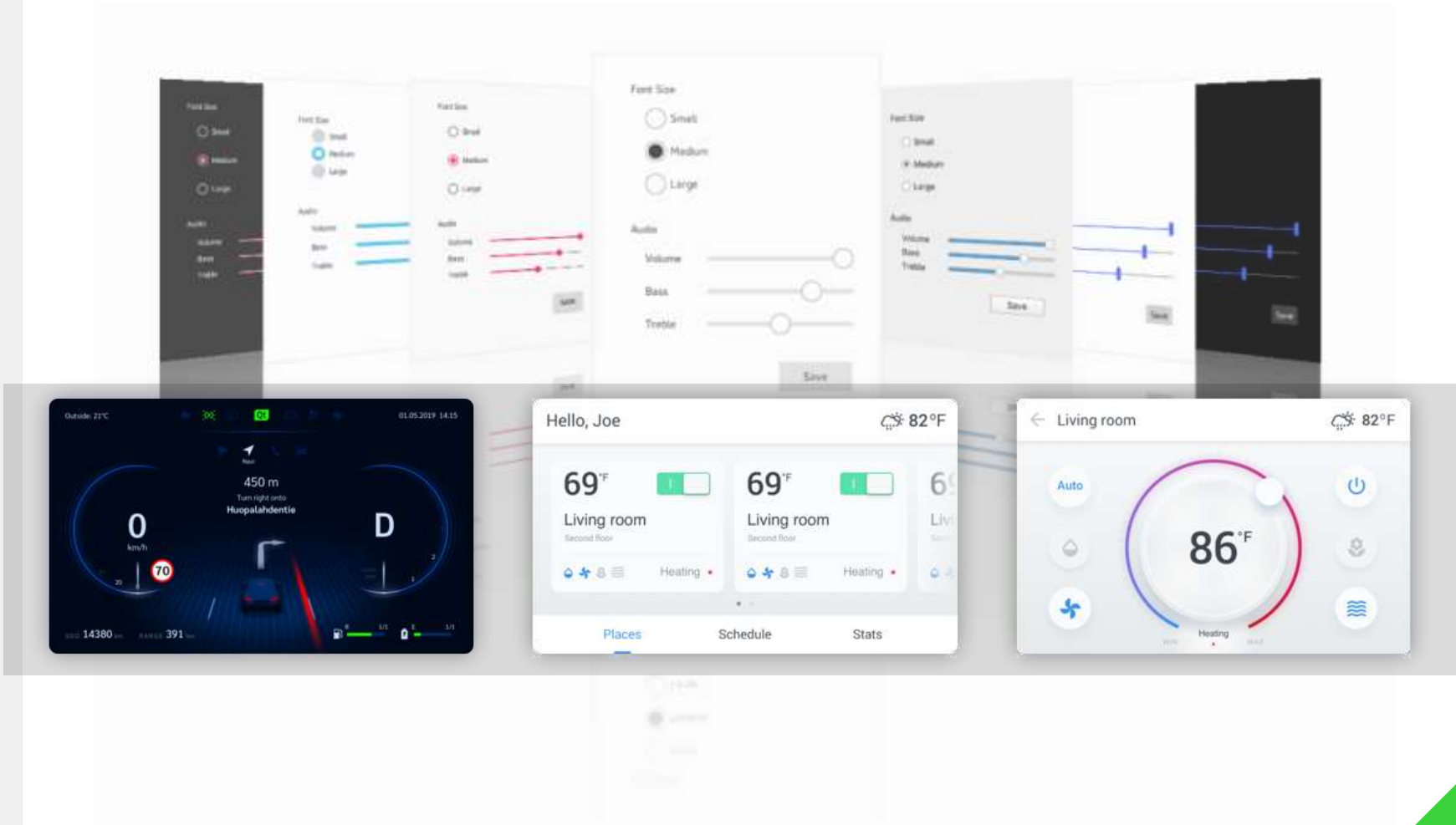


UX

Provide a smartphone-like user experience with Qt Quick Control

Qt Quick Control 2.0 Provides Mobile-like User Experience

Build and deploy complete interface with rich library of UI controls

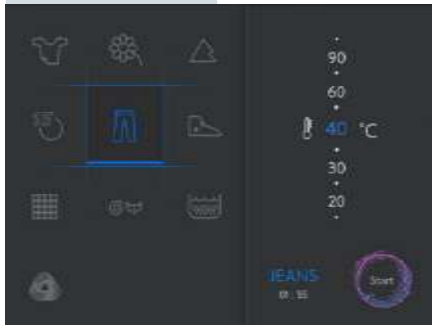


SINGLE CODEBASE

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



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

Cortex-M4 MCU (<10 EUR BOM) – 640x480

Companion app

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

Mid-range

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

ARMv7A 32bitlow end MPU (<30 EUR BOM) – 854x480

High-end

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

ARM-v8A 64bit Quad Core high end MPU (<100 EUR BOM) – 960x480

Reuse

Reuse source code across ARM architectures and Mobile applications

Graphic Reuse on Powerful Platforms

Code Once, Deploy Everywhere

GUI Application	Qt UI Frontend	C++ Backend	GUI Application	Qt UI Frontend	Qt C++ Backend	GUI Application	Qt UI Frontend	Qt C++ Backend
Graphics Runtime utilizing on-chip 2D graphics accelerator			Qt Framework utilizing OpenGL-ES/Vulcan			Qt Framework Android / any OS		
ARM Cortex-M			ARM Cortex-A			ARM Cortex-A		

Reuse the UI frontend
Extend backend logic with Qt C++ APIs

Speed Up

Fast, effective development with QML and Qt Tools

Boost your process with QML and Qt Tools

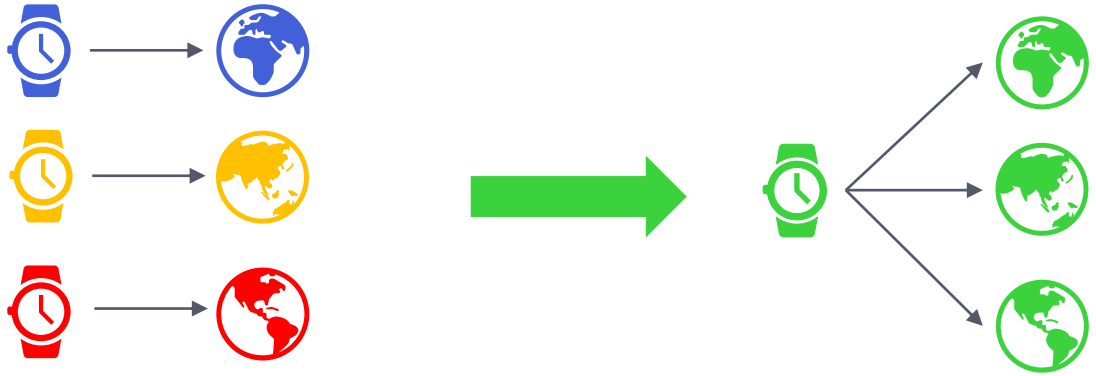
Easy, intuitive QML for UI frontend, C++ logic backend and straightforward tools simplify development process.

Investment

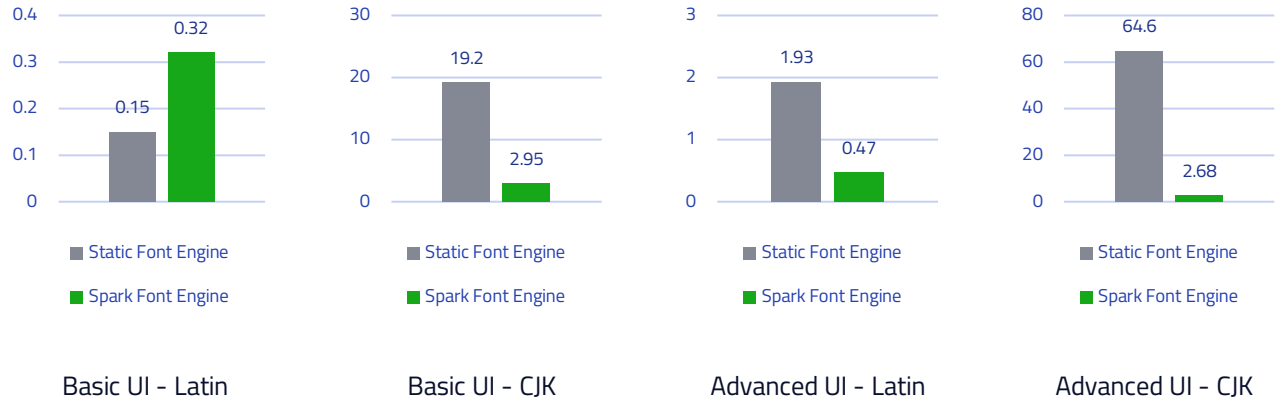
Minimizing footprint and logistic effort for product line targeting global market

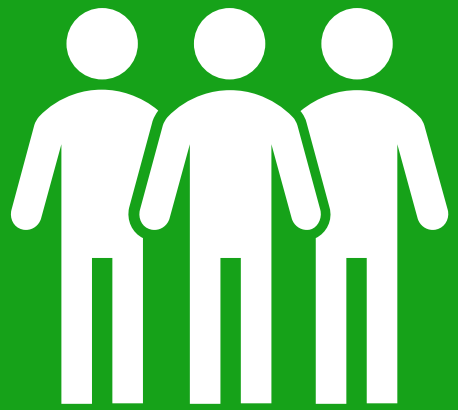
Optimized font engine to support product for a worldwide market

Reducing the memory footprint with dedicated font engine and disrupting the complexity in managing different regions and languages



› Binary sizes in MB (lower is better)





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."

Why Qt for MCUs?– *User perspective*

Re-use

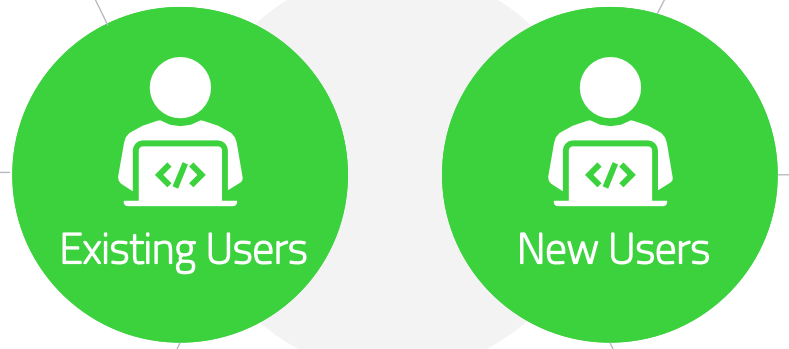
Re-use the same Qt tools and workflow to develop for MCUs and MPUs

Upgrade

Upgrade to a modern, stable cross-platform graphical toolkit

Save Costs

Re-use same engineers and same technology across family of devices



Code Re-use

UI can be integrated with existing C/C++ backend.

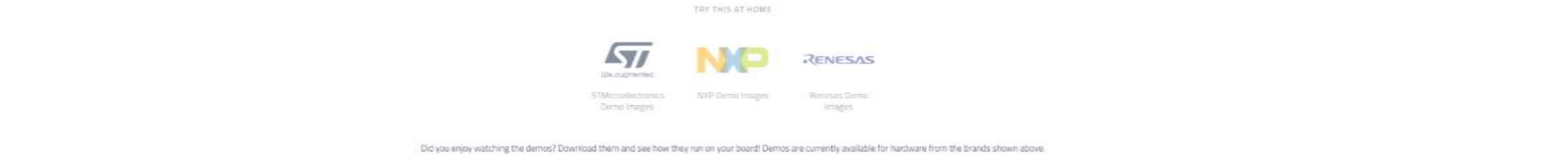
Low Memory Footprint

Reduce BOM costs by moving to MCU without compromising on user experience.

Scalable UI

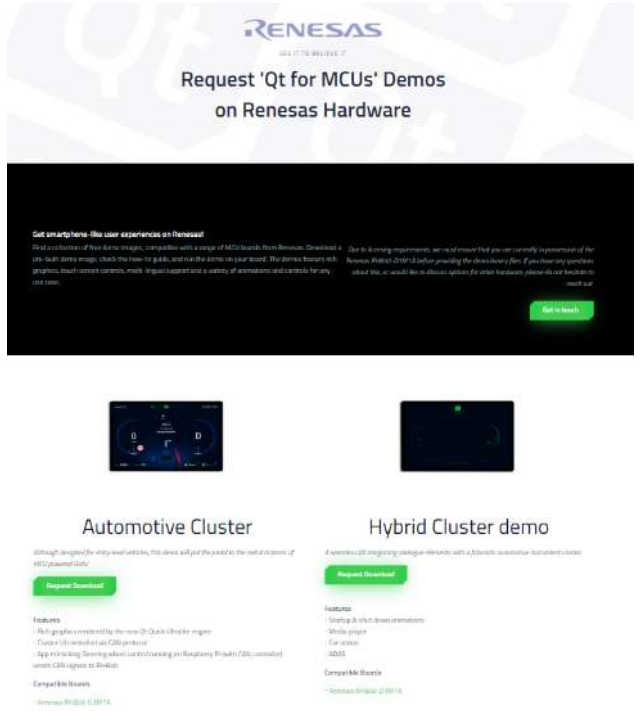
Flexible layout mechanism allows the UI to scale to screen sizes and aspect ratios, even to Linux, Android, etc.

Qt for MCUs Demo Images by hardware partner



https://www.qt.io/microcontrollers-st

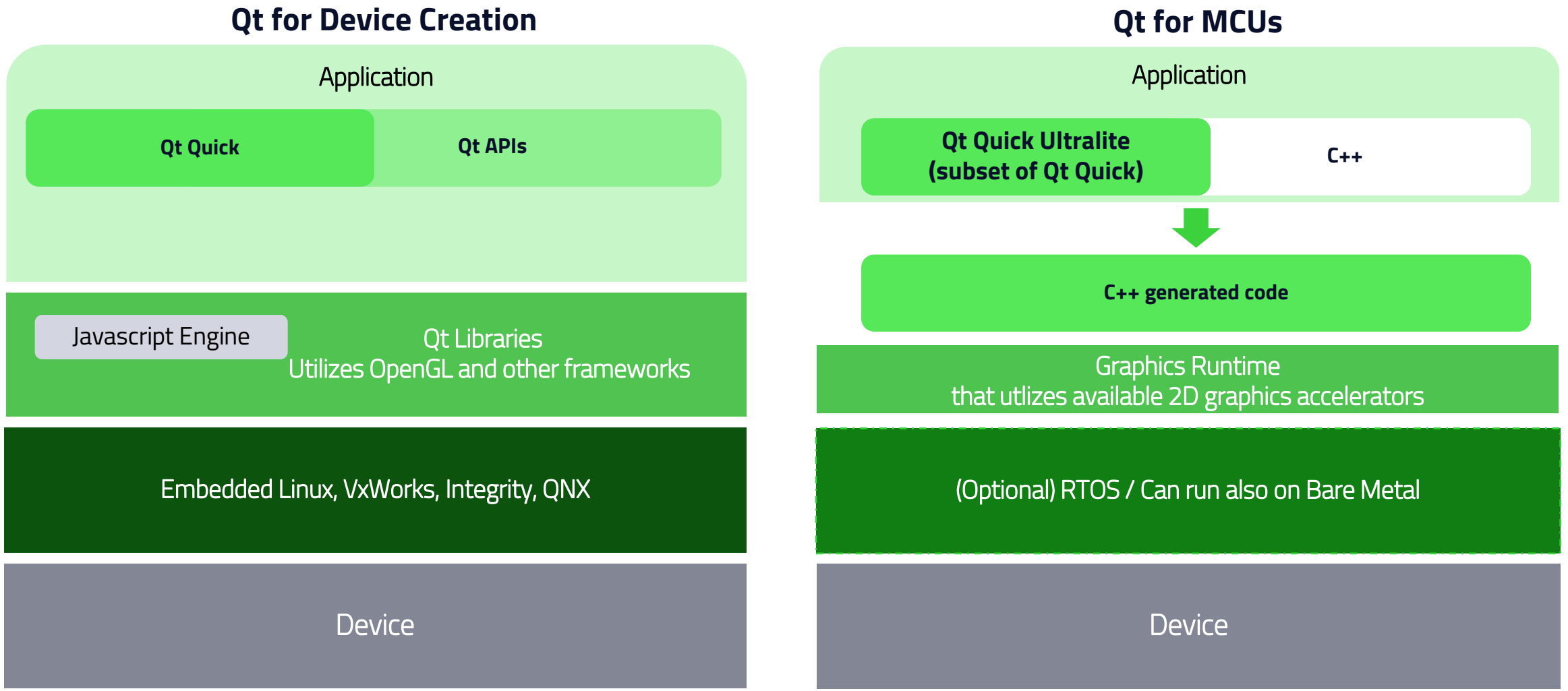
https://www.qt.io/microcontrollers-nxp



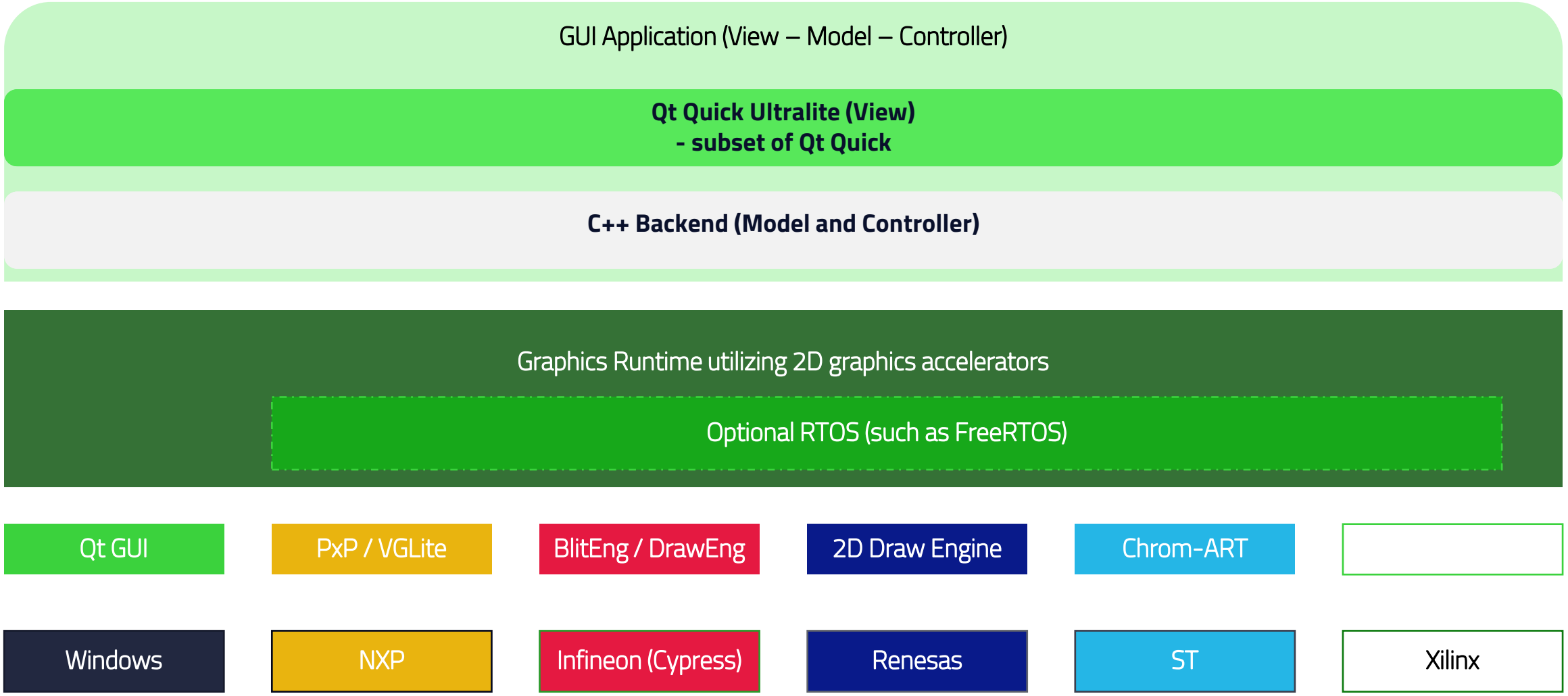
https://www.qt.io/microcontrollers-renesas

Qt for MCUs – Technical Deep Dive

Managing constrained devices scenario



Qt Quick Ultralite – Application Block Diagram



QT FOR MCUS

Supported Platforms

Vendor	MCU	OS	Compiler	Supported since	Industry Verticals		Developer License	
					Auto	Consumer / Industrial	Essential	Ultimate
Infineon	Traveo II	Bare Metal	GHS	*Pre-release version*				✓
ITE	IT986x	ITE provided	ITE custom GCC	*Pre-release version*				✓
Nordic	nRF5340	Zephyr OS	GCC 8	*Pre-release version*				✓
NXP	i.MX RT1050	Bare Metal , FreeRTOS	GCC 8, IAR 8.50	V1.0			✓	✓
	i.MX RT1060	Bare Metal	GCC 8, IAR 8.50	V1.3			✓	✓
	i.MX RT1064	Bare Metal , FreeRTOS	GCC 8, IAR 8.50	V1.1			✓	✓
	i.MX RT1170	FreeRTOS	GCC 8, IAR 8.50	V1.8				✓
	i.MX RT595	FreeRTOS	GCC 8	*Pre-release version*				✓

QT FOR MCUS

Supported Platforms

Vendor	MCU	OS	Compiler	Supported since	Industry Verticals		Developer License	
					Auto	Consumer / Industrial	Essential	Ultimate
Renesas	RA6M3G	Bare Metal	GCC 8, IAR 8.50	V1.8		✓	✓	✓
	RH850/D1Mx	Bare Metal	GHS 7.1.6	V1.0	✓			✓
	R-Car	FreeRTOS, SafeRTOS	GHS 7.1.6	*Pre-release version*	✓			✓
STM	32F769i-Disco	Bare Metal , FreeRTOS	GCC 8, IAR 8.50	V1.0		✓	✓	✓
	32F7508	Bare Metal , FreeRTOS	GCC 8, IAR 8.50	V1.0		✓	✓	✓
	32H750B	Bare Metal	GCC 8, IAR 8.50	V1.1		✓	✓	✓
	32F469i-Disco	Bare Metal	GCC 8, IAR 8.50	V1.1		✓	✓	✓
	32L4R9i-disco	Bare Metal	GCC 8, IAR 8.50	V1.1		✓	✓	✓
	32L4R9i-eval	Bare Metal	GCC 8, IAR 8.50	V1.1		✓	✓	✓
Xilinx	UltraScale+ MPSoC	Bare Metal , FreeRTOS	GCC 8	*Request via PS*		✓		✓

What about other platforms and Custom Hardware ?

- Platform APIs are provided to implement hardware interaction
- Clear segregation between Application and Platform modules for easier porting
- Detailed Porting Guide is provided as part of documentation
- Qt Support team is there to help customers along
- Qt Professional Service can take up Platform enablement projects

Qt for MCUs 1.8

1. New Features
2. Other Changes
3. Coming Next

Why upgrade to 1.8

- **Reduce memory costs -> more ways to reduce RAM footprint**
  8-bit color graphics, font data in flash memory, vector graphics
- **Create advanced UI with custom shapes**
  Advanced 2D drawing with Qt Quick Shapes
- **Faster development cycle -> Easier integration with any build system**
  Build QUL applications as static libraries
- **More freedom -> additional platforms are supported**
  NXP i.MX RT1170 and Renesas RA6M3

Other Changes

- Font data runtime location
  - Optionally keep font data in flash memory
  - Slower text rendering but reduced RAM footprint
- Control QUL engine from custom main loops
  - Qul::Application::update()
- Improvements to Qt Creator and Qt Design Studio integration
- **10** bug fixes

RELEASE DATE: **JUNE 2021**

v1.9

Improved support for automotive platforms

- *New MCU: Infineon (Cypress) Traveo II*
- *Layer support for NXP i.MX RT1170*
- *Vector graphics support for Renesas RH850*

Complex scripts and bidirectional text

Display text in all most common languages with support for complex scripts such as Arabic, Indian scripts, Thai or Hebrew.

Custom visual items

Write complex GUI components in C++ and integrate them in QML scenes. The new PaintedItem API gives you direct access to screen pixels for any custom drawing.

Font quality control

Adjust the rasterization quality of glyphs with 1-bit per pixel and no anti-aliasing to lower the memory requirements.

Improved RTOS integration

Interrupt-safe and thread-safe event handling.

Development on Linux

Design, build and flash your Qt for MCUs application on Ubuntu 20.04.

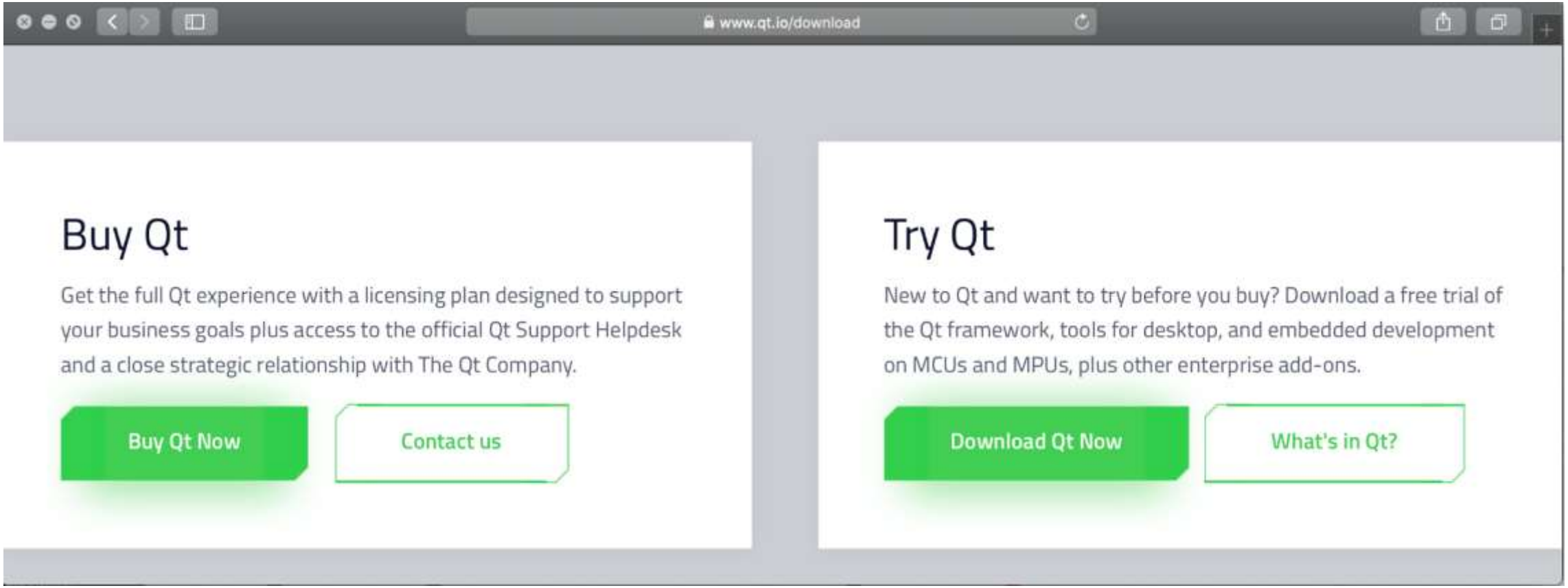
Camera example

Learn how to easily integrate camera frames to Qt for MCUs UIs using the camera module bundled with NXP i.MX RT1050/60 evaluation kits.

Qt

Get Qt

www.qt.io/download



Qt for MCUs
Resources / Videos/ Documentation etc

- All resources are under: https://resources.qt.io/qt-mcus
- Design Studio to MCU tutorial:
  - https://resources.qt.io/qt-mcus/creating-dynamic-uis-with-qt-design-studio-and-photoshop-on-mcus-on-demand-webinar
- Port existing Qt Quick applications to Microcontrollers
- Qt for MCUs Documentation

Q&A – Win a discount coupon or Qt branded goodies

1. Can developers reuse code written with Qt for MCUs on other products from Qt such as Qt for Device Creation ?
2. What is the programming language used to develop frontend and backend code in Qt for MCUs ?
3. Name any 3 hardware platforms supported by Qt for MCUs

*Refer to next slide for coupon & goodies details

Participation Details for Q&A coupons and goodies

Please e-mail your answers to india@qt.io before 31 May 2021 and win a discount coupon to one of our upcoming trainings on Qt for MCUs. 5 winners will be eligible for the discount coupon. Each of these trainings are worth USD 350.

- **May 25, Programming for Microcontrollers with Qt (STMicroelectronics):** https://www.qt.io/events/programming-for-microcontrollers-with-qt-stmicroelectronics-1614823030
- **June 8, Creating User Interfaces for Microcontrollers (NXP) :** https://www.qt.io/events/creating-user-interfaces-for-microcontrollers-1614803076
- **June 15, Design & Development for Microcontrollers (Renesas):** https://www.qt.io/events/hmi-ui-ux-design-development-for-microcontrollers-renesas-rh850-1614824699

For correct answers sent after 31 May 2021, we'll give away 10 Qt branded goodies.

Thank you!