

**Embedded
Online
Conference**



www.embeddedonlineconference.com

Embedded Development with Qt: Exploring Use Cases Beyond MCUs and MPUs

Exploring Use Cases Beyond MCUs and MPUs

Veli-Pekka Heinonen

THE SPEAKER

Veli-Pekka Heinonen



➔ Senior Product Manager, The Qt Group

Focus: Embedded software development

Veli-Pekka Heinonen is currently Senior Product Manager at The Qt Company, responsible for embedded platforms, WebAssembly, multimedia and related Cloud projects. He has over 20 years of experience in the IT and hardware industries, and he has been working in various product management and product marketing roles. Before joining Qt, his last position was at Cisco, working with customer projects, support, and security for a communications client.

Veli-Pekka holds a master's degree in software engineering from Helsinki University of Technology (now known as Aalto University).

AGENDA

1 Traditional Qt Use Cases

2 Qt for WebAssembly

3 Qt in the Cloud

4 Qt Angle to Digital Twins

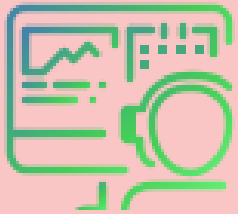
5 Key Takeaways



Traditional Qt Use Case

Leverage full Qt offering to your benefit – design, develop, test, deploy, also for a portfolio of products across MCU and MPU.

Challenges in Creating Digital Experiences



Getting great software developers is hard

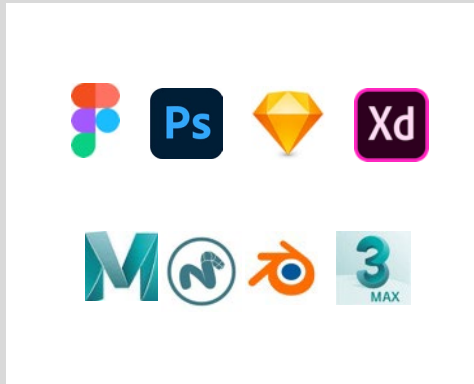


Building a portfolio of devices is expensive



UI designers, developers and QA work in silos

UI Design

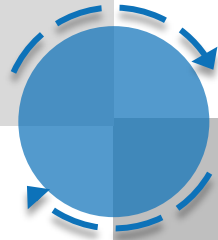


2D / 3D UI Design Tools

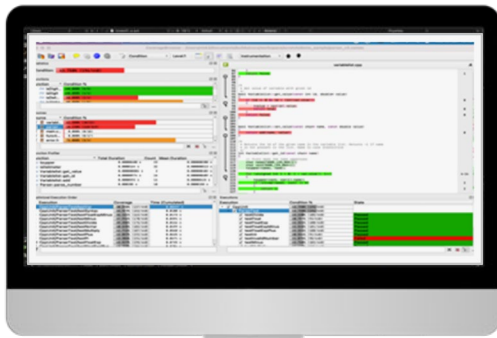
UI Composition



Qt Design Studio

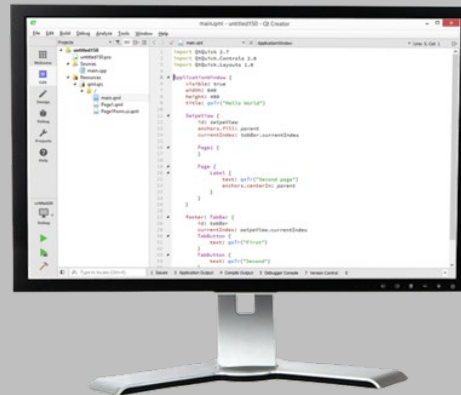


Test



Qt Squish, Qt Coco

Develop



Qt Creator IDE

Agile Software Development

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.



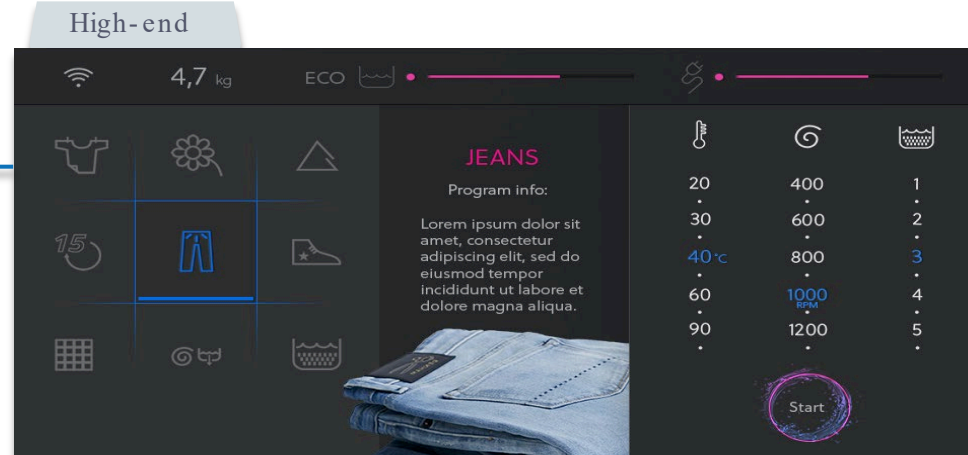
Cortex-M4 MCU (<10 EUR BOM) – 640x480

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



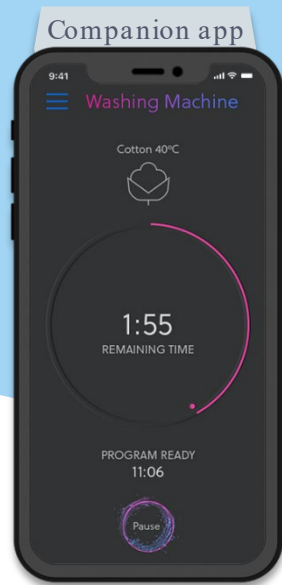
ARMv7A 32bit low end MPU (<30 EUR BOM) – 854x480

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



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

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

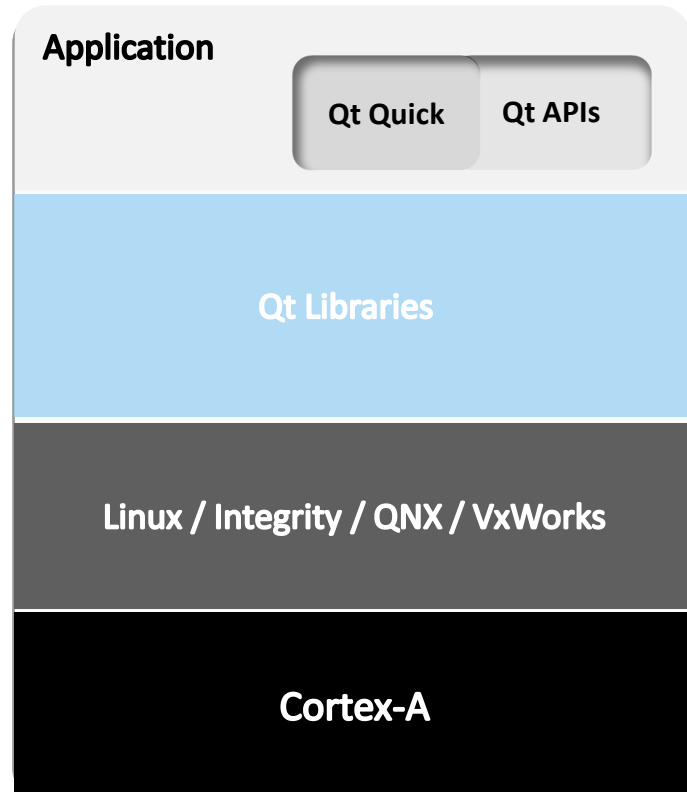


- ✓ Complex/simple apps
- ✓ Win, Mac, Linux, Android, iOS
- ✓ WEBASM

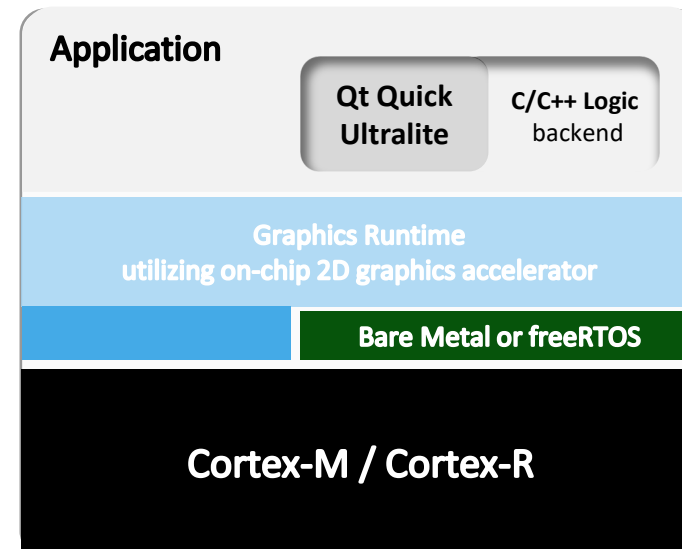


A New Endeavor to Address Scalability

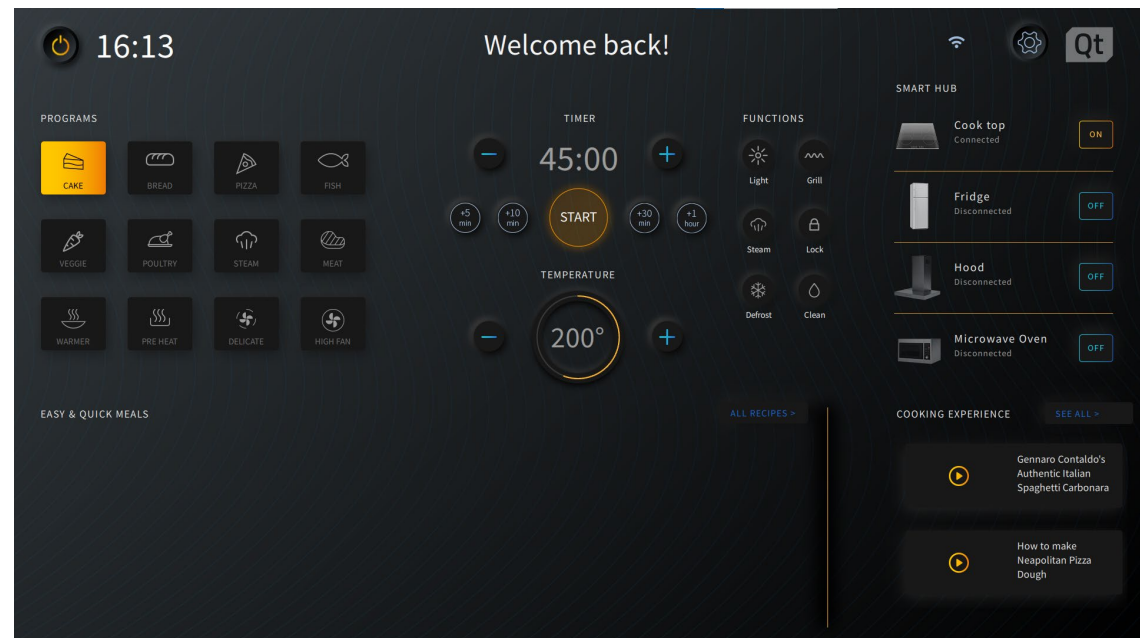
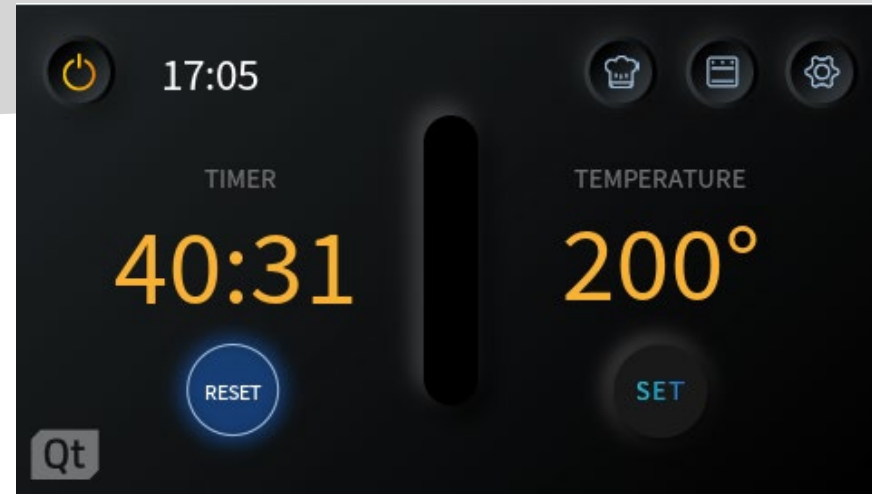
Qt for Device Creation on MPU



Qt for Device Creation on MCU

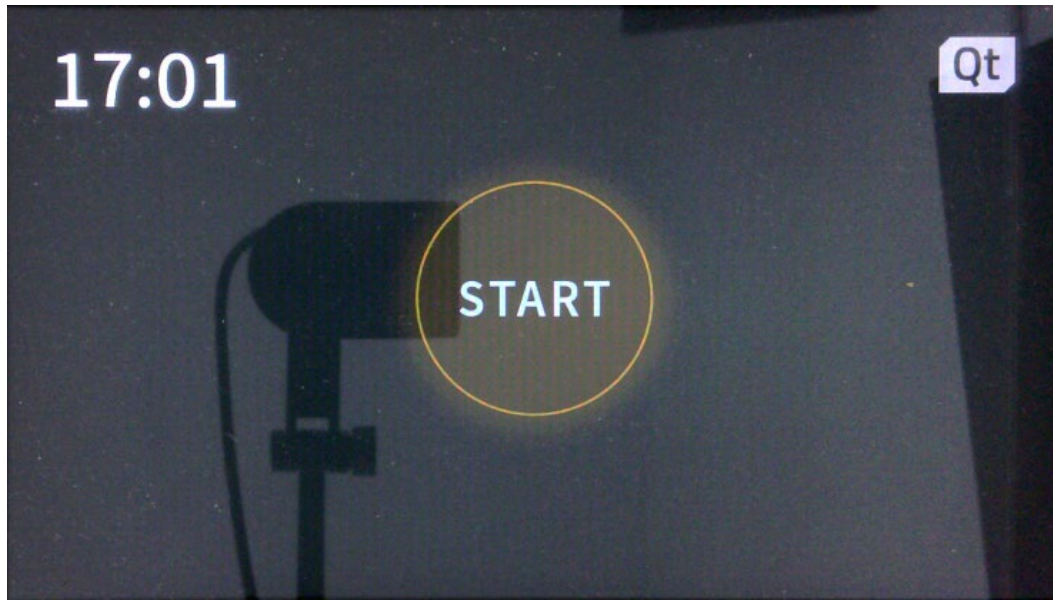


- Oven interface
- Modern look and feel
- Product variants MCU, MPU
- Controls
- Multimedia
- Internet content access
- 3D animations



Qt Toolchain From Design to Deploy – MCU targets

NXP i.MX RT1050



Deploy



Figma, Sketch or Photoshop

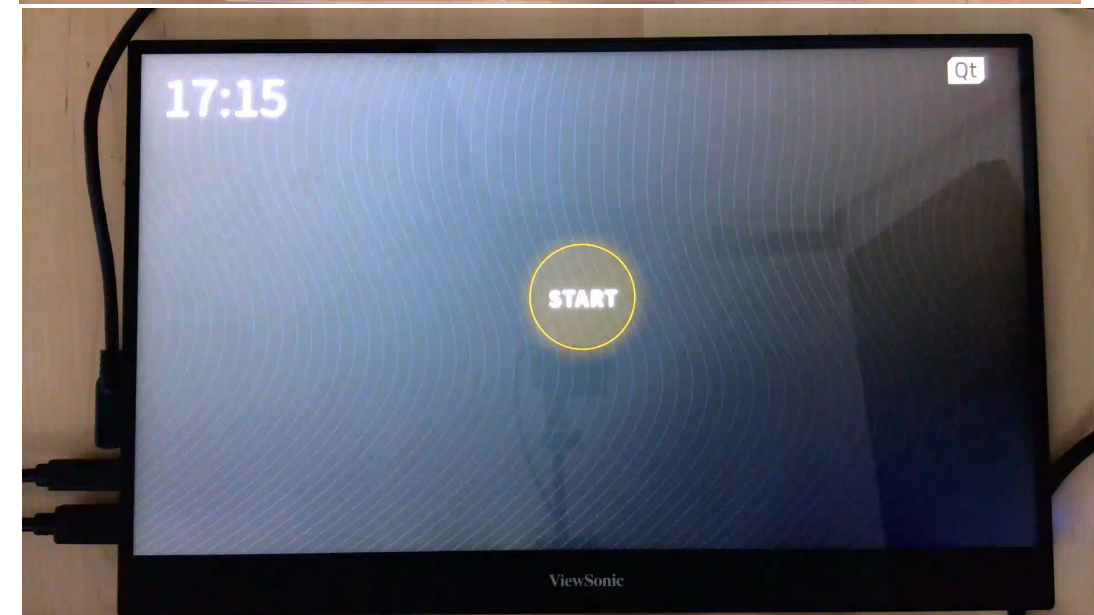


Qt Design Studio – Design + UI/UX



Qt Creator – Backend C/C++ connection & Deploy

i.MX8 QM





2

Qt for WebAssembly

Key use cases of web -enabling your existing C/ C++ application and getting a web story using same team, same tools, same programming language

- Standard browser capability based on W3C standard in phones, tablets and computers – Big industry bet on **zero install** platform – recompile C++ code to run in browser's Javascript sandbox in binary format
- Some platform aspects under evolution
 - See Qt page: <https://doc.qt.io/qt-6/wasm.html>
- Comes with both client and server support – Qt supports the client side

W3C COMMUNITY & BUSINESS GROUPS

Home / WebAssembly Community Group

WEBASSEMBLY COMMUNITY GROUP

The mission of this group is to promote early-stage cross-browser collaboration on a new, portable, size- and load-time-efficient format suitable for compilation to the web.

[webassembly/spec](#)

Group's public email, repo and wiki activity over time

Year	Activity
2015	Low activity
2016	Low activity
2017	Low activity
2018	Low activity
2019	Low activity
2020	Low activity
2021	Low activity
2022	High activity (JFMAMJJASOND)

Note: Community Groups are proposed and run by the community. Although W3C hosts these conversations, the groups do not necessarily represent the views of the W3C Membership or staff.

Tools for this group

- Mailing List
- IRC
- Github repository
- RSS
- Contact This Group

Resources

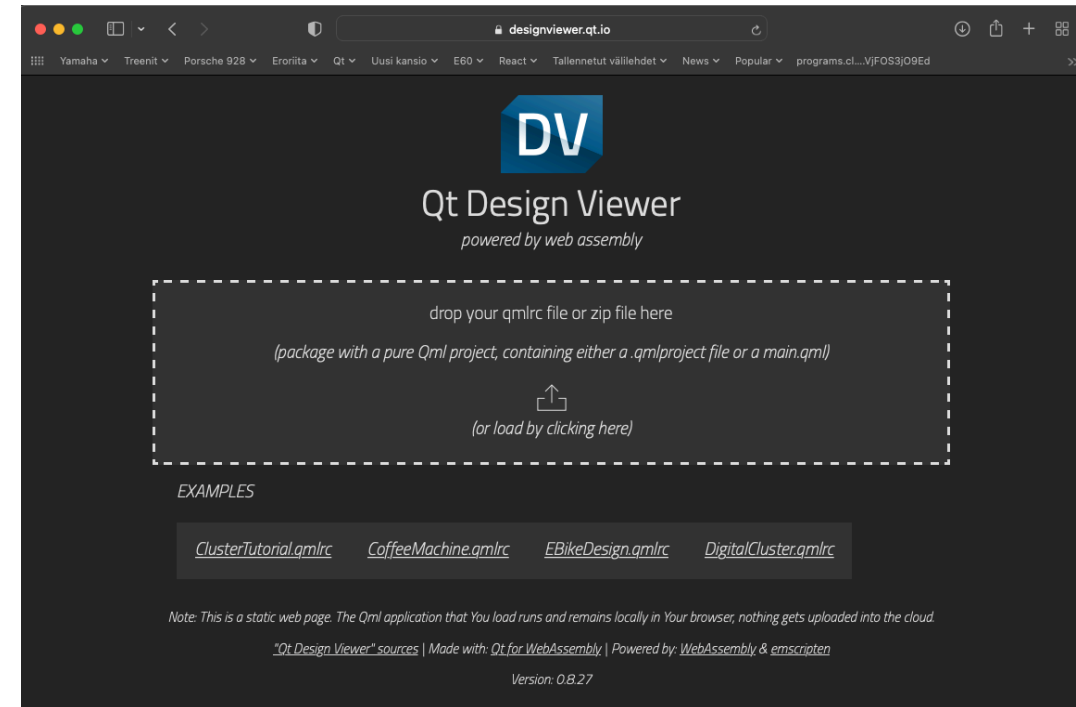
- Charter
- Code of Conduct

Get involved

Qt for WebAssembly

High Level Use Cases

- › Benefit
 - Take your Qt application to the web using standard W3C browser capability
- › Use-Cases
 - Taking native apps to the web – often requires working around platform limitations
 - Deploying apps without app stores – zero install
 - Sharing e.g. embedded designs with stakeholders
 - Embedded devices with no or limited display
 - Remote control and monitoring of devices in the field
- › Content / Deliverables
 - Full support for WebAssembly in Qt 6.4+
 - Documentation and support



Qt Design Studio 3.8

Qt for WebAssembly Product Strategies

- In general WebAssembly has had higher performance compared to javascript, making it especially suited for more complex web apps.
- Please see Qt documentation for WebAssembly limitations <https://doc.qt.io/qt-6/wasm.html>
- Different architectures have been utilised as per webassembly.org
 - Entire code base in WebAssembly
 - Main frame in WebAssembly, but the UI in JavaScript / HTML
 - Re-use existing code by targeting WebAssembly, embedded in a larger JavaScript / HTML application



Areas for Special Consideration with Qt for WebAssembly

- Multithreading – needs to be enabled separately and is still experimental in WebAssembly, additionally multithreading requires modifying the server configuration
- SIMD performance enhancement – needs to be enabled separately on a need basis, may not help all apps.
- Networking – websockets are the basic mechanism although TCP/UDP sockets can be used as well with limitations. HTTP requests with limitations.
- Local file access – QFileDialog will display the virtual filesystem instead of the user's real filesystem.
- Clipboard access – some differences due to the web sandbox
- Debugging and profiling - Wasm debugging is done on browser javascript console, debugging applications on Wasm directly within Qt Creator is not possible.
- Fonts – only a few fonts supported by default, but this can be enhanced by the app.



<https://doc.qt.io/qt-6/wasm.html>

WebAssembly Use Case – Device Without Display

Code reuse

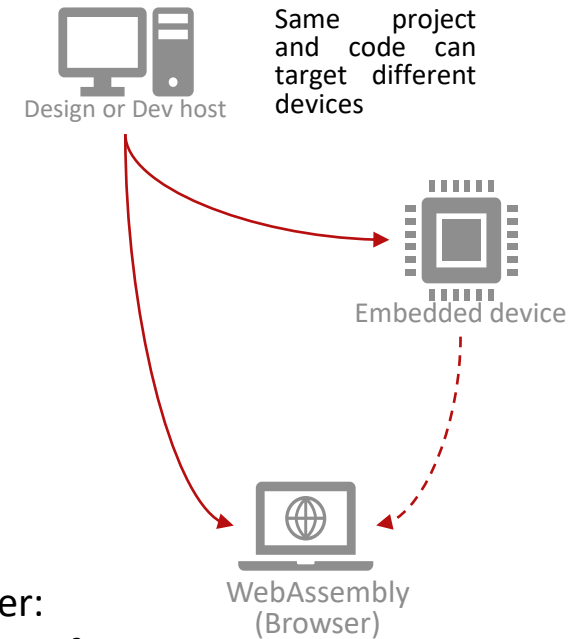
Allow embedded developers to have a web distribution story also:

- Qt HTTP server for hosting wasm binaries in embedded device
- Utilise ready interface technology between app front end (browser) and backend (embedded device)
 - All code does not have to run in browser
 - Wasm can be combined with HTML/ CSS ->faster startup

Solution for products without UI

Decide when to offer Qt mobile vs Qt WebAssembly

- Zero installation key for some projects



Qt can offer:

- 3D wasm performance
- Data channel (gRPC, REST, etc)
- Low-end performance over HTML5
- Reuse of old code
- WebGPU (roadmap)



2

Qt in the Cloud – concept in validation phase

How you can use Qt in a cloud -based development environment to your benefit.

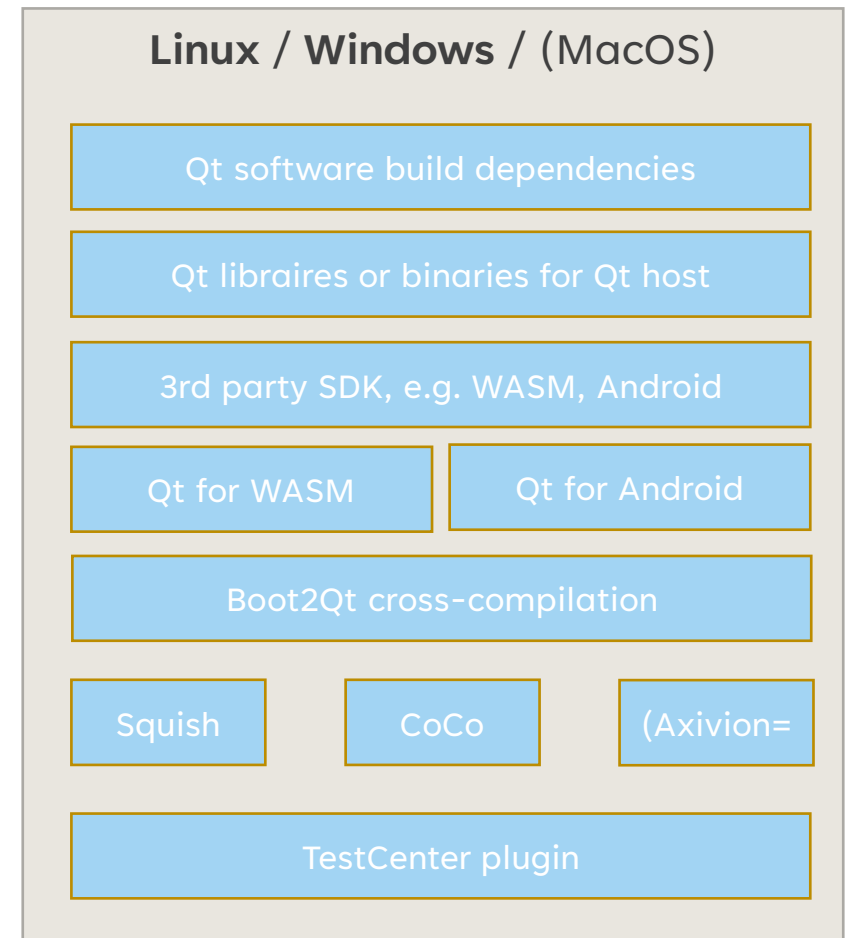
For piloting contact: qt_ci_cloud@qt.io

Qt CI/QA Cloud - Piloting

⑩ Qt container images which contain

- Prebuilt Qt libraries
- Pre-packaged Qt dependencies
- Pre-packaged QA tools

⑩ You can choose a specific container image with modules that are relevant to you



Containerized Development for Embedded Cross-Compilation

Benefit

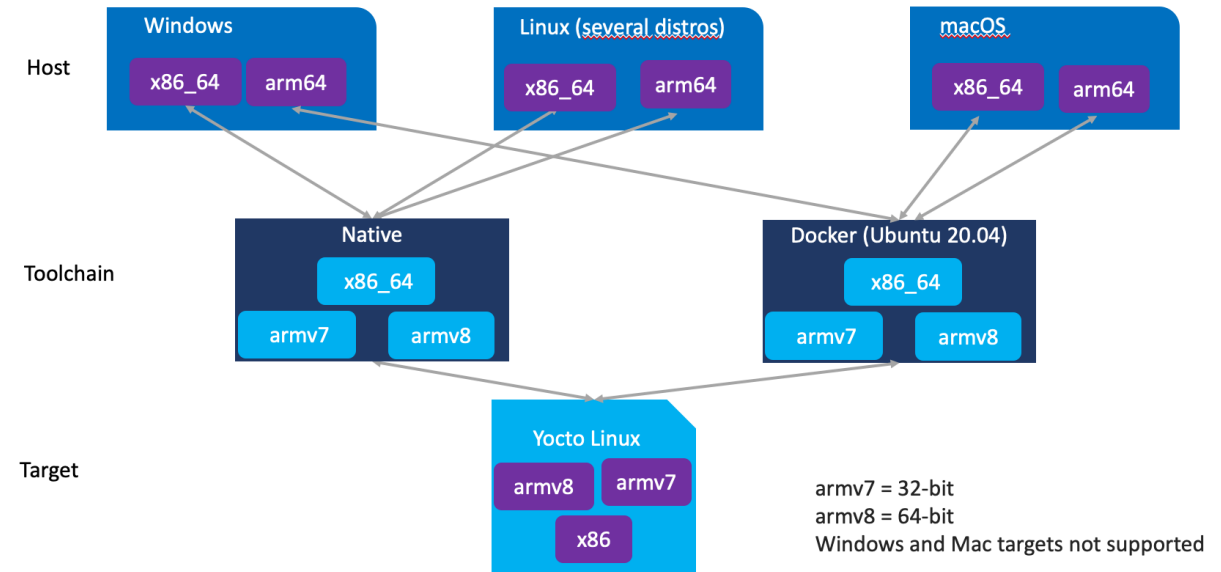
- Broader embedded developer reach with new host platforms
- More unified development environment

Use Cases

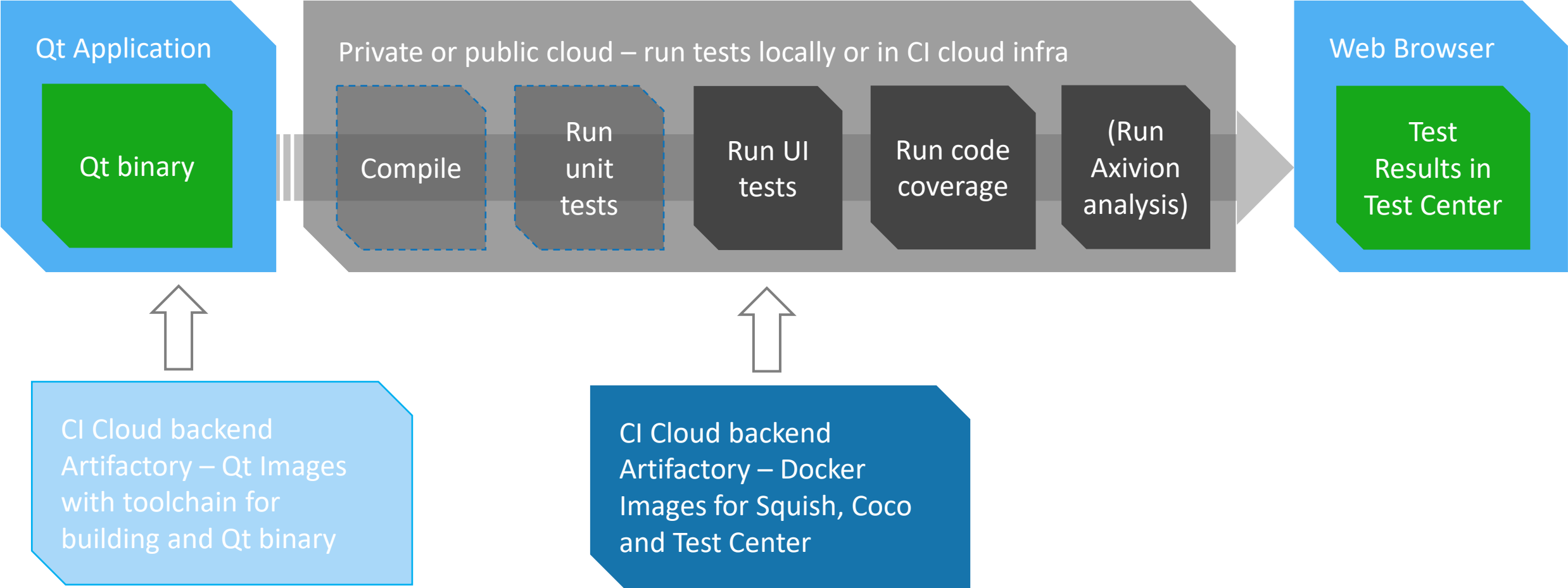
- Use macOS (Intel or ARM) for Boot2Qt development
- Use Windows on ARM for Boot2Qt development

Content / Deliverables

- Docker image containing the Boot2Qt toolchain
- Documentation



Qt CI/QA Cloud – Development Flow



Qt CI/QA Cloud

Demo Video for Squish UI
Testing

Easy setup using containers
Headless operation
Test cases in Squish IDE



The screenshot shows the Squish Test Center web interface. The browser address bar displays the URL: `test-center.qt.io/Testcenter/history?page=1&project0=QtCI Tests&batch0=2022-12-22&testsuite0=pytest&group=Tests%2CStats&start=-10b&end=0b`. The interface includes a navigation bar with 'Squish Test Center', 'QtCI Tests', 'History', 'Explore', and 'Manual Testing'. A user profile 'Admin' is visible in the top right. The main content area shows a 'Stats' view for a test batch from 2022-12-22. A table titled 'Statistics over 3 batches from Mon Dec 19 2022 to Wed Dec 21 2022' displays various performance metrics for different test cases.

			Flakiness ↕	Success Rate ↕	Times broken ↕	Retry Count ↕	Skip rate ↕	Avg duration ↕	Avg time to fix after failure ↕	Avg runs between fixes ↕
OS	pytest Test Suite	test_qtci Test Case	0.20	80.00%	1	0	0.00	4.6s	-	-
	pytest Test Suite	test_all_variables Test Case	0.00	100.00%	0	0	0.00	1ms	-	-
	pytest Test Suite	test_client_setup Test Case	0.00	0.00%	5	0	0.00	4.8s	-	-
	pytest Test Suite	test_token_and_verify Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_status_config Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_get_remotes Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_branches Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_commits Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_get_all_pipelines Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_get_pipeline_with_id Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_get_file Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_post_stage Test Case	0.00	0.00%	0	0	5.00	< 1ms	-	-
	pytest Test Suite	test_post_commit Test Case	0.00	0.00%	5	0	0.00	1ms	-	-
	pytest Test Suite	test_post_pipeline Test Case	0.00	0.00%	0	0	5.00	1ms	-	-
	pytest Test Suite	test_post_push Test Case	0.00	0.00%	0	0	5.00	1ms	-	-

ANALYZE RESULTS IN TEST CENTER

The Test Center can be running locally, on-site or in cloud and can be configured as desired.

We even provide a easy install image for deploying a new Test Center

Qt HW Cloud Pilot

Use existing HW boards by Qt Group over the cloud for faster development and testing

- Web-based reservation of hardware
- SSH-access
- Supports running Squish UI-tests on selected HW
- HW selection
 - i.MX8 QuadMax
 - i.MX8 Nano
 - i.MX8 Mini
 - i.MX8M Quad
 - Toradex Apalis i.MX6
 - Raspberry Pi 4
 - Intel NUC





4

Qt Angle to Digital Twins – concept in validation phase

Qt-based API simulation for speeding up your
development project before you get the real hardware
– piloting available soon

Qt Angle to Digital Twins

Background

Focus on the **needs of a UI application developer**

- To enable UI application development without hardware

Provide **simulation layers**

- For HW & OS related Qt APIs

Today

- Get HW and start developing SW

With Qt Digital Twin

- Test your designs without HW
- Develop without HW
- Code ready before HW available



"Test your designs without HW "
"Develop without HW "
"Code ready before HW available"

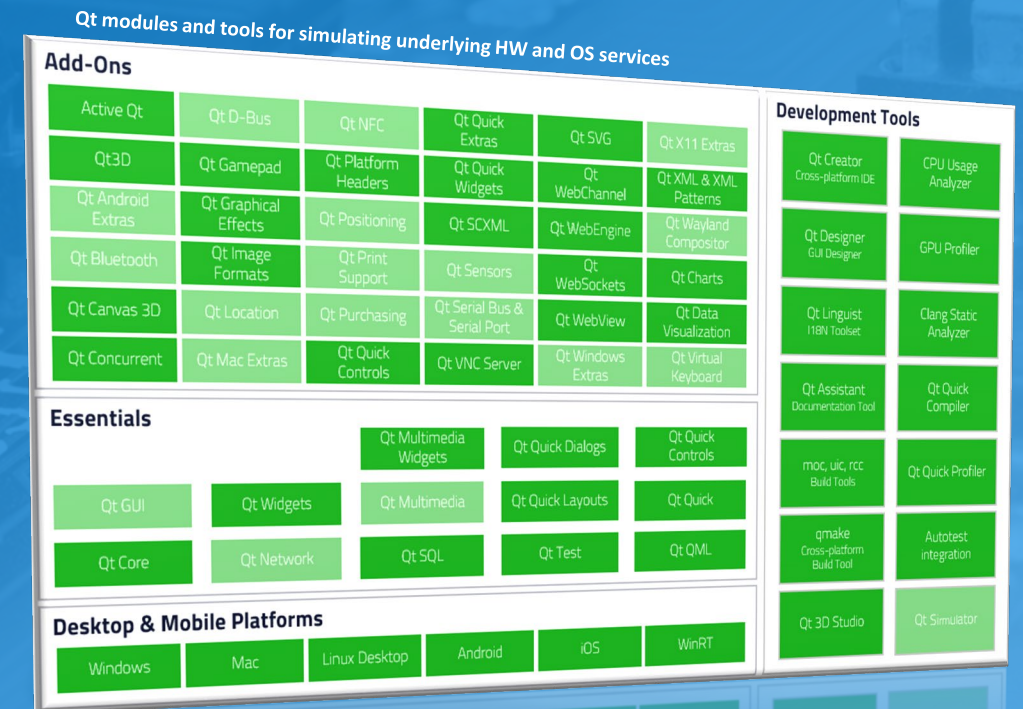
Why Qt Digital Twin?

Benefit

Qt Framework provides modules that abstract the underlying HW and OS services

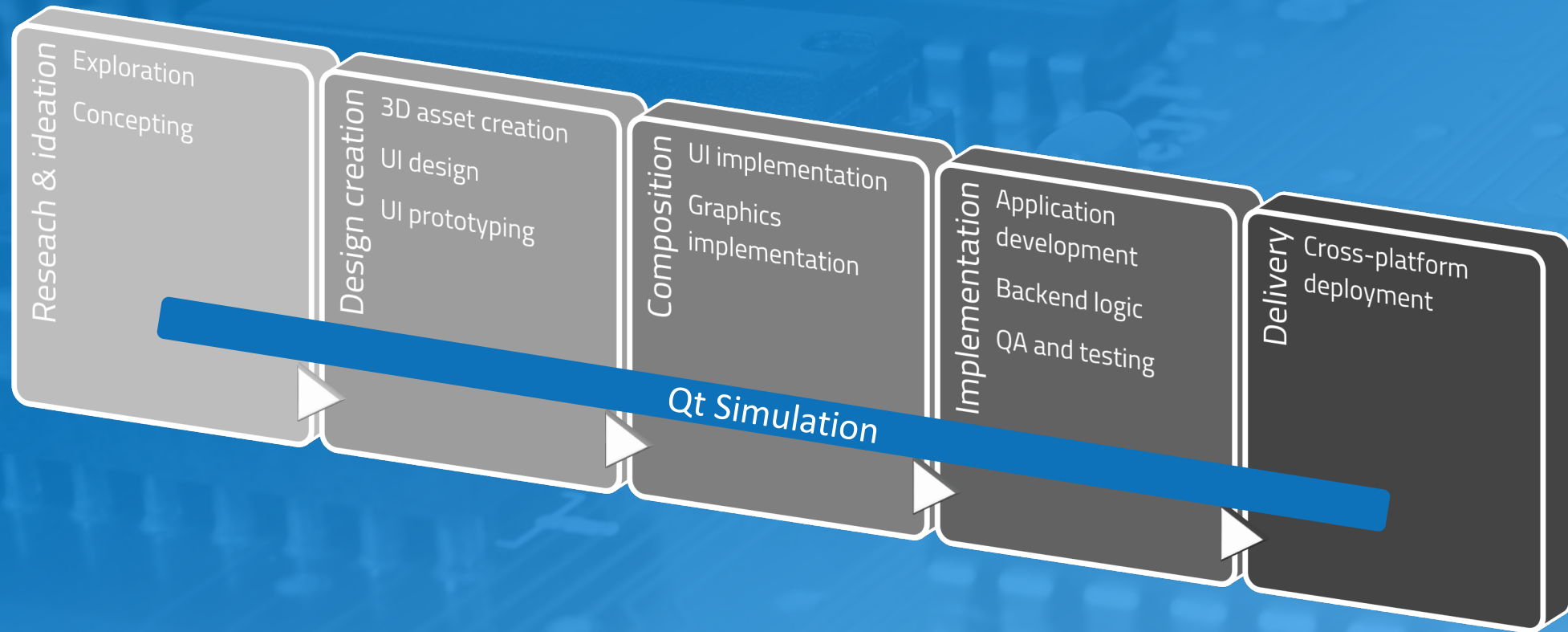
This allows "*Code once, apply everywhere*"

But it also allows *simulation of HW and OS services* transparently without modifications to the application SW



Role fo Qt Simulation Offering

Process



Design Flow of Qt Simulation Offering

The screenshot shows the Qt Design Studio interface for a project named 'RobotSimu'. The main window displays a 2D view of a simulation environment with various components like 'WasmPart', 'Robot XYZ', and 'androidcontrolpart'. The 'Code' editor shows the QML code for the simulation, including imports for QtQuick, QtConnectivity, and QtSimulation, and the definition of the 'Robot Simulation Example' window. The 'Properties' panel shows the properties of the selected 'robotSimuProj' object. The 'Components' panel shows the available components for the simulation, including 'App', 'BT Tunneling', 'PhysicalComponent', and 'SimulationApplicationLaunch'.

The screenshot shows the 'RobotSimu' application window. The 'Object tree' panel displays the hierarchy of the simulation, including 'BT tunneling', 'RobotArmApp', 'robotSimuApp', 'AndroidPhoneSimu', 'BT device', 'assemblyLineView', 'FactoryScene', 'RoboArm', 'appAndroidController', 'BT device', and 'Robot XYZ (xyz1)'. The 'System view' panel shows the simulation environment with a 'BT Tunneling' component highlighted. The 'Properties' panel shows the properties of the selected 'xyz1' object, including 'Id: xyz1', 'Name: Robot XYZ', 'Model: RF-1234', 'Manufacturer: ABC Corp.', 'Connection Type: BT Low Power', and 'Address: 12:03:44:77:98:01'. A blue arrow labeled 'Run Ctrl+R' points to the 'Run' button in the bottom right corner.

API Support

First APIs

Qt Positioning API

- To enable a factory floor demo
- <https://doc-snapshots.qt.io/qt6-6.4/qtpositioning-index.html>

Qt Bluetooth Low Energy API

- To extend the existing BT support
- To enable the control of a physical robot
- <https://doc.qt.io/qt-6/qtbluetooth-le-overview.html>

Qt Sensors API

- Needed for a factory floor demo
- <https://doc.qt.io/qt-6/qtsensors-index.html>



Bluetooth Simulation + Tunneling

Pilot

Qt Bluetooth API is the 1st Qt API to get simulated
Simulation of BT protocol

- To provide virtual BT devices

Tunneling of BT signaling and data via WebSocket

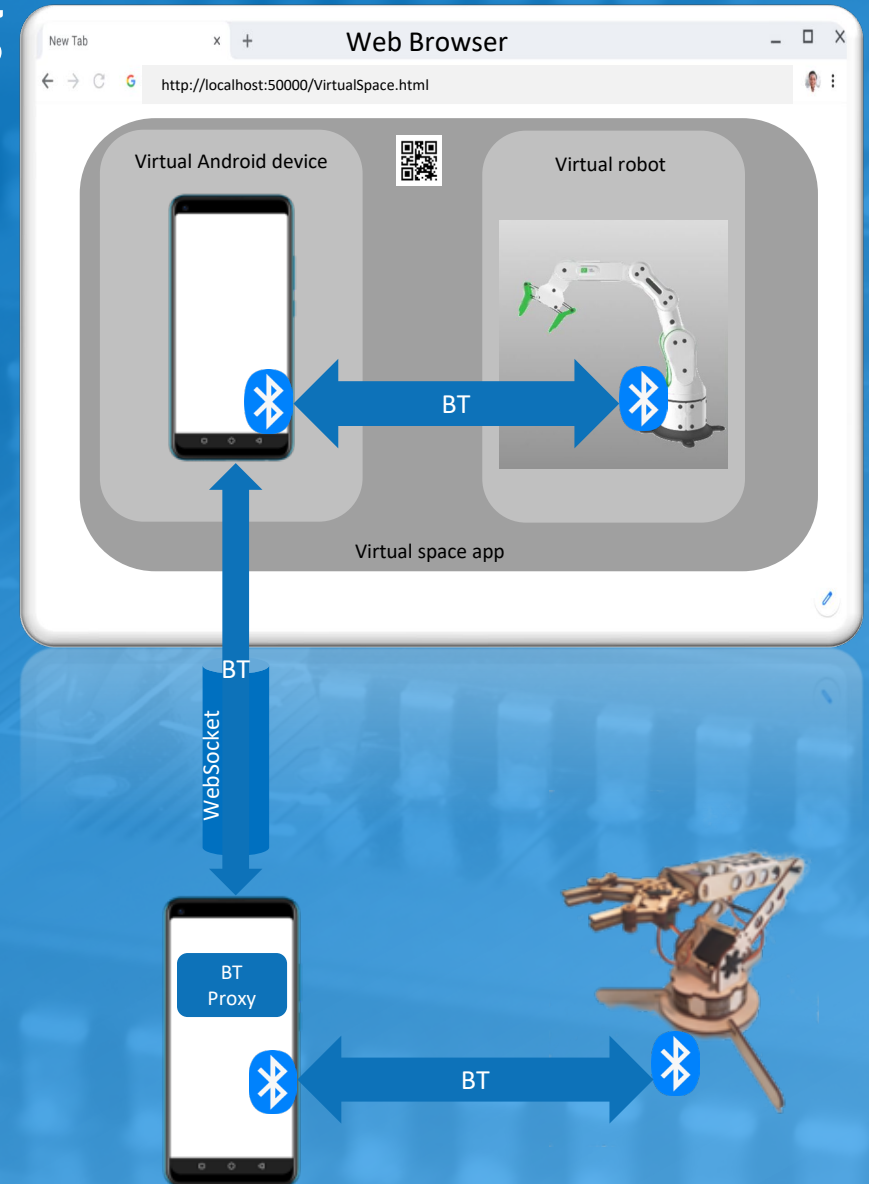
- To share the same BT network between (physical and virtual) BT devices

Proxying of BT signaling and data

- To allow any BT device (physical or virtual) to extend the shared BT network

Example use cases

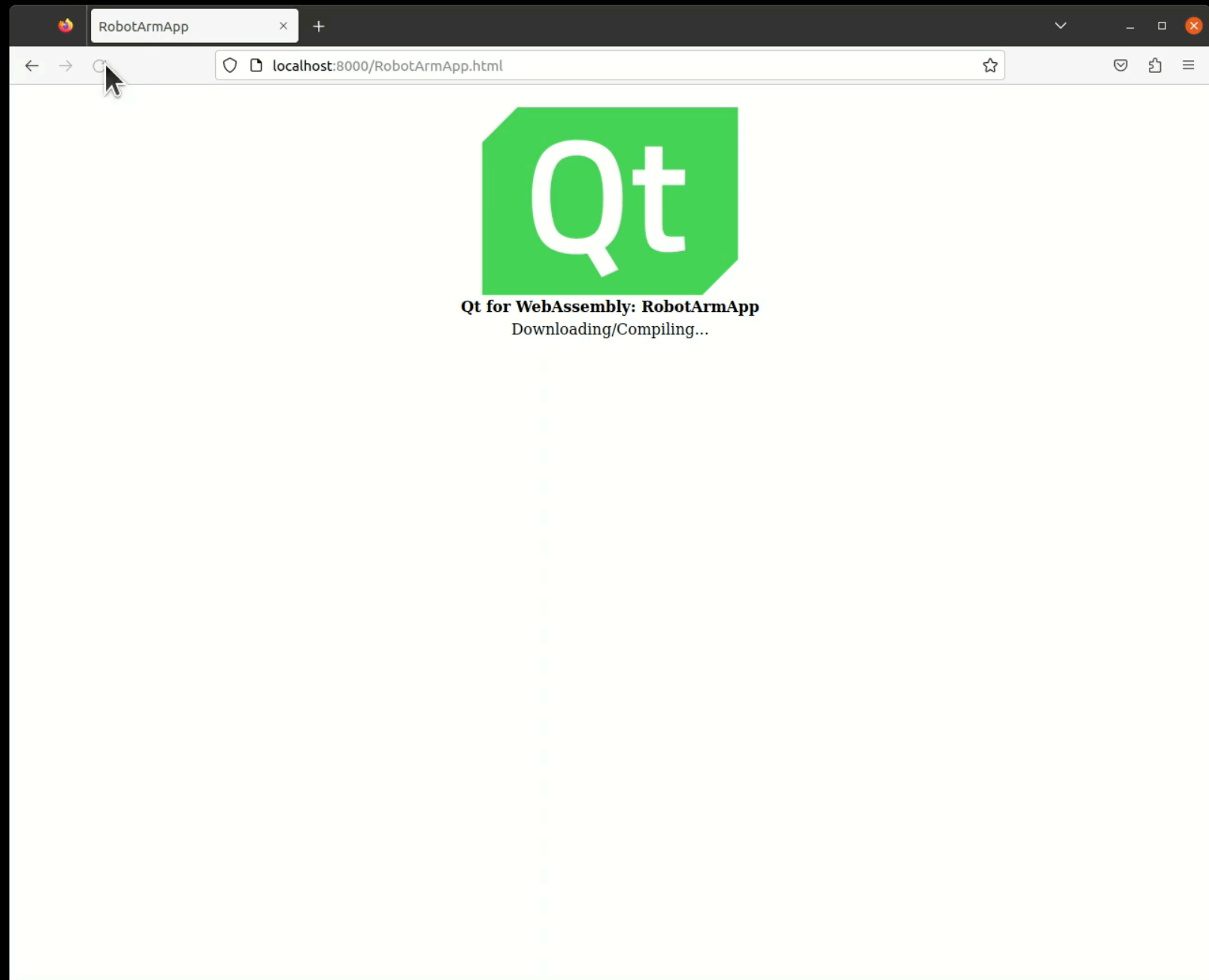
- To let a virtual Android device to control a physical robot
- To stream audio between a physical BT head set and a virtual Android device



Demo

Qt Bluetooth
and Sensor
APIs

Simulated environment





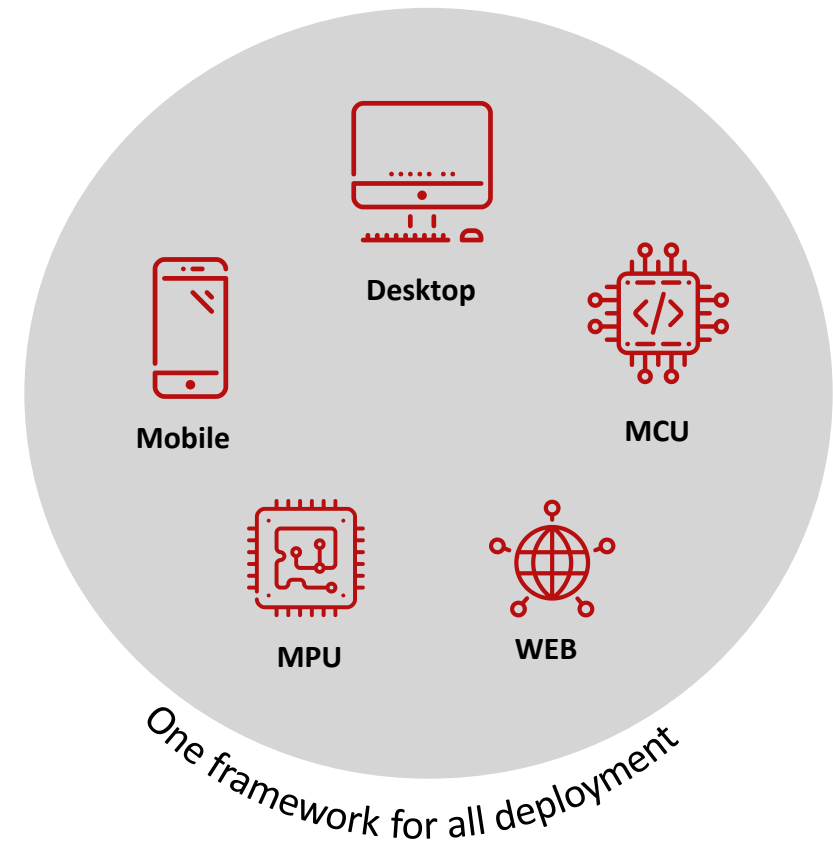
Summary

Key takeaways on extended Qt usage scenarios

Key Takeaways

Expanded Qt Use Cases – With Piloting Options in Concept Validation Phase

- In addition to the Qt framework Qt also provides a wide range of tools for design, development and test
- Qt can be used also with containers for easier deployment in development host or cloud
- WebAssembly offers a web story for C/ C++ apps using same tools and teams
- API simulation with Qt Digital Twin for faster development soon in pilots



THANK YOU

Embedded
Online
Conference

w w w . e m b e d d e d o n l i n e c o n f e r e n c e . c o m