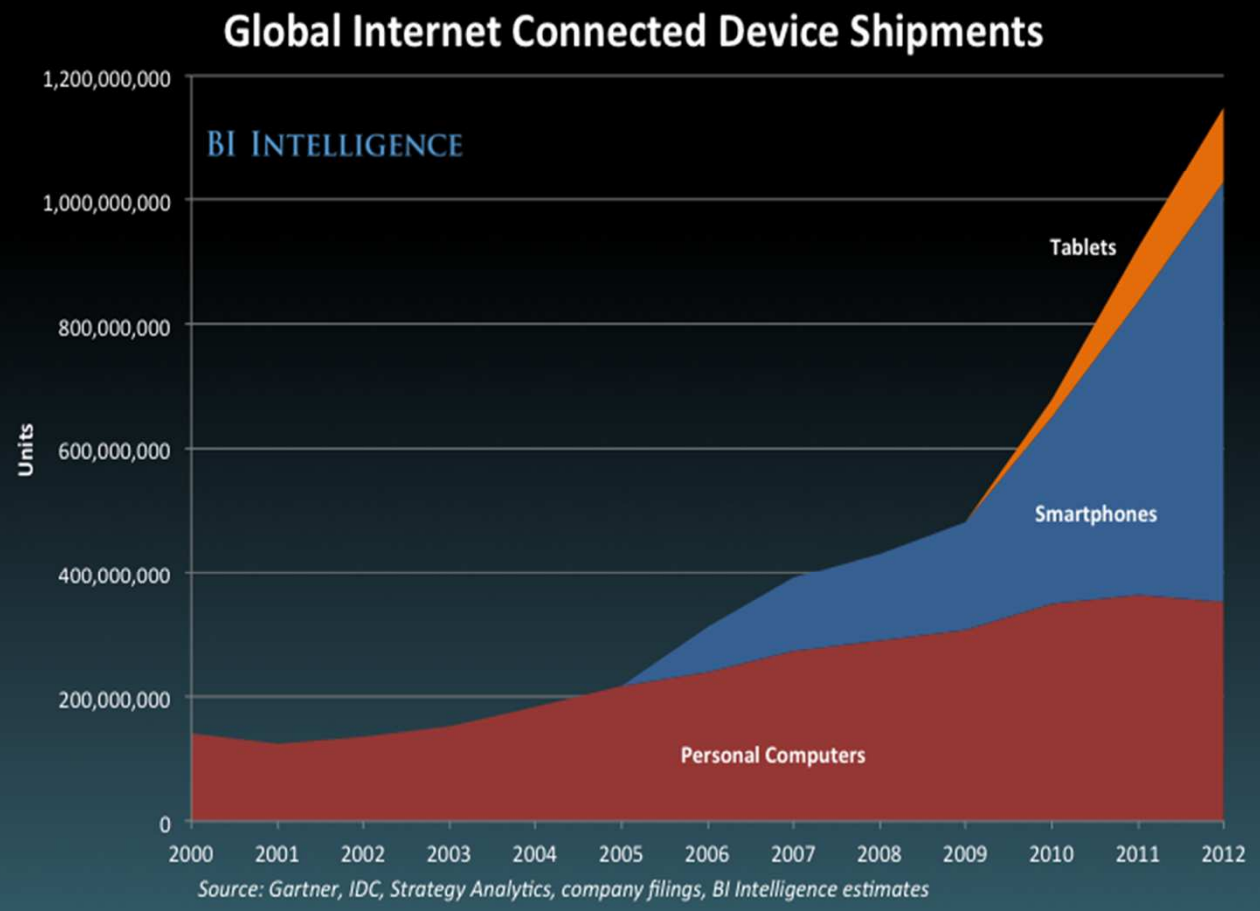
Multi-Platform Applications with Qt

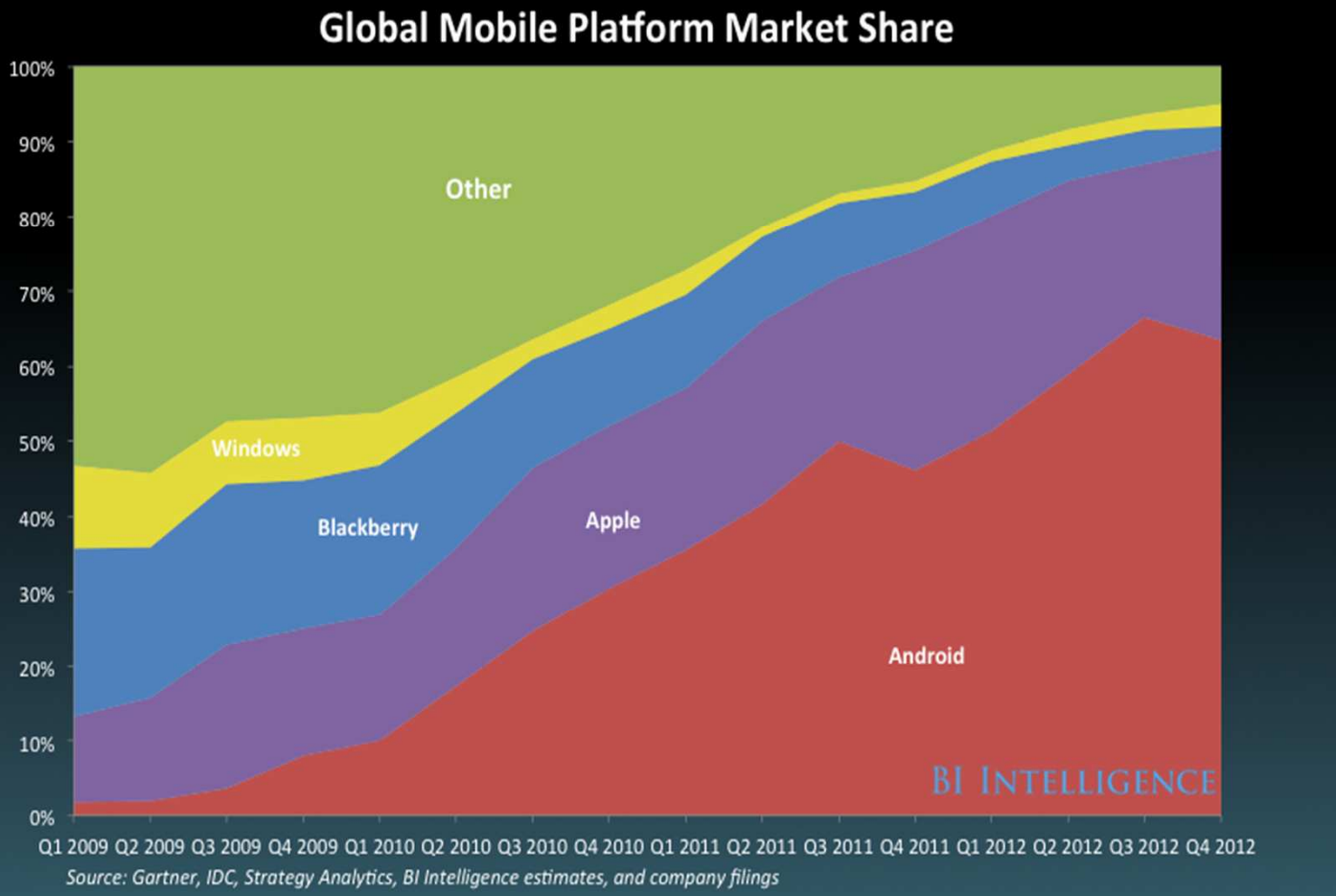
Tuukka Ahoniemi

tuukka.ahoniemi@digia.com
@tuukka_a
Santa Clara, CA
(soon back in Tampere, Finland)

Globally, mobile device sales are 2x PC sales



They work across all the mobile operating systems



BUSINESS INSIDER

So…

- Desktop and mobile OS landscape fragmented
  - Android, BlackBerry, iOS, Windows Phone
  - Mac OS X market share on desktop is growing
    - 14% market share in US, >30% in Norway
  - Tablets replacing laptops
- Data often stored in cloud, access from everywhere
- Embedded Device, control from mobile, tablet or PC
  - Central heating, Sound system, Set top box
  - Factory lines

N-Screen Problem

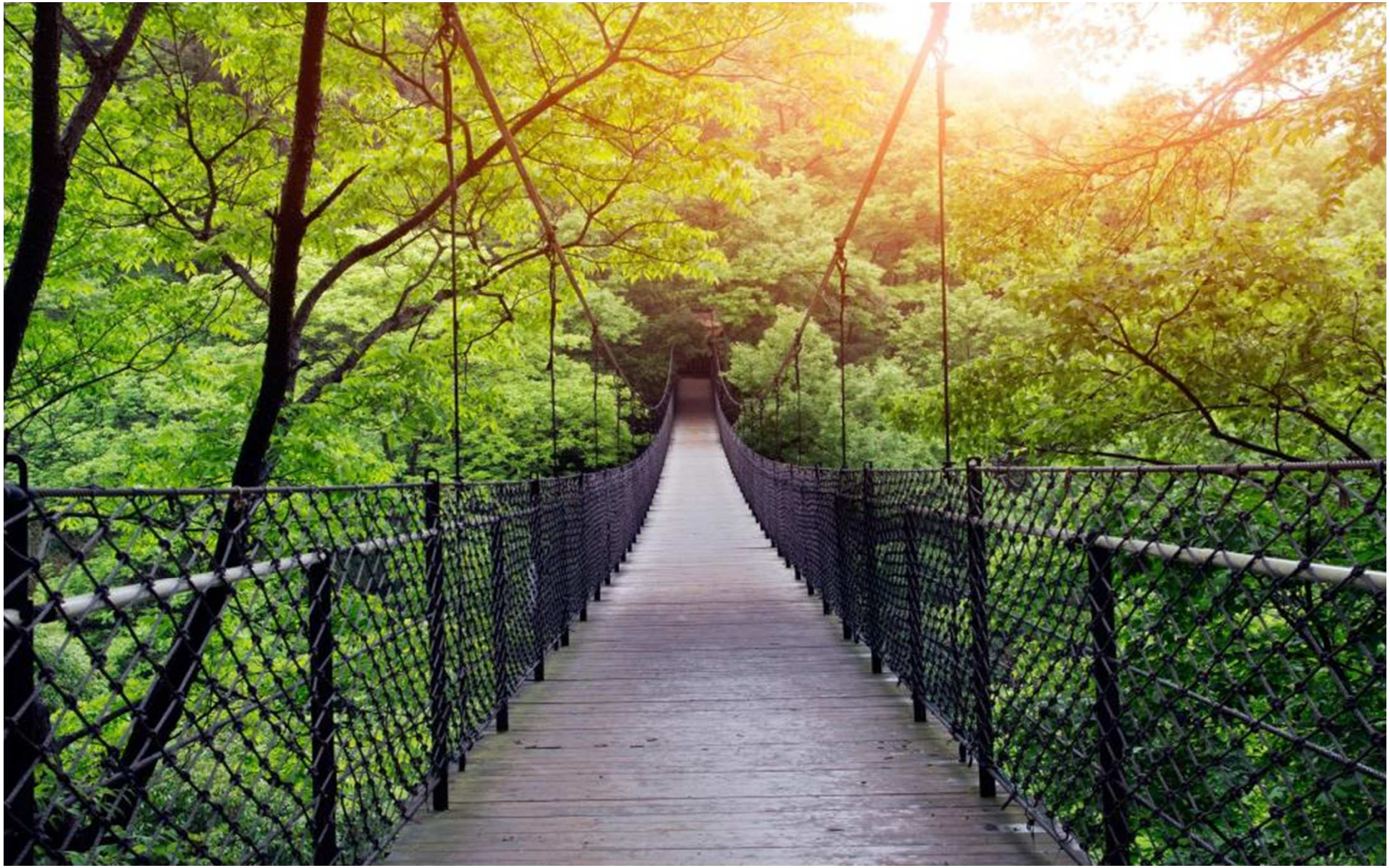
It’s Not Too Late to Change Your Career!

…Contact Your Local Military Draft Person Today!

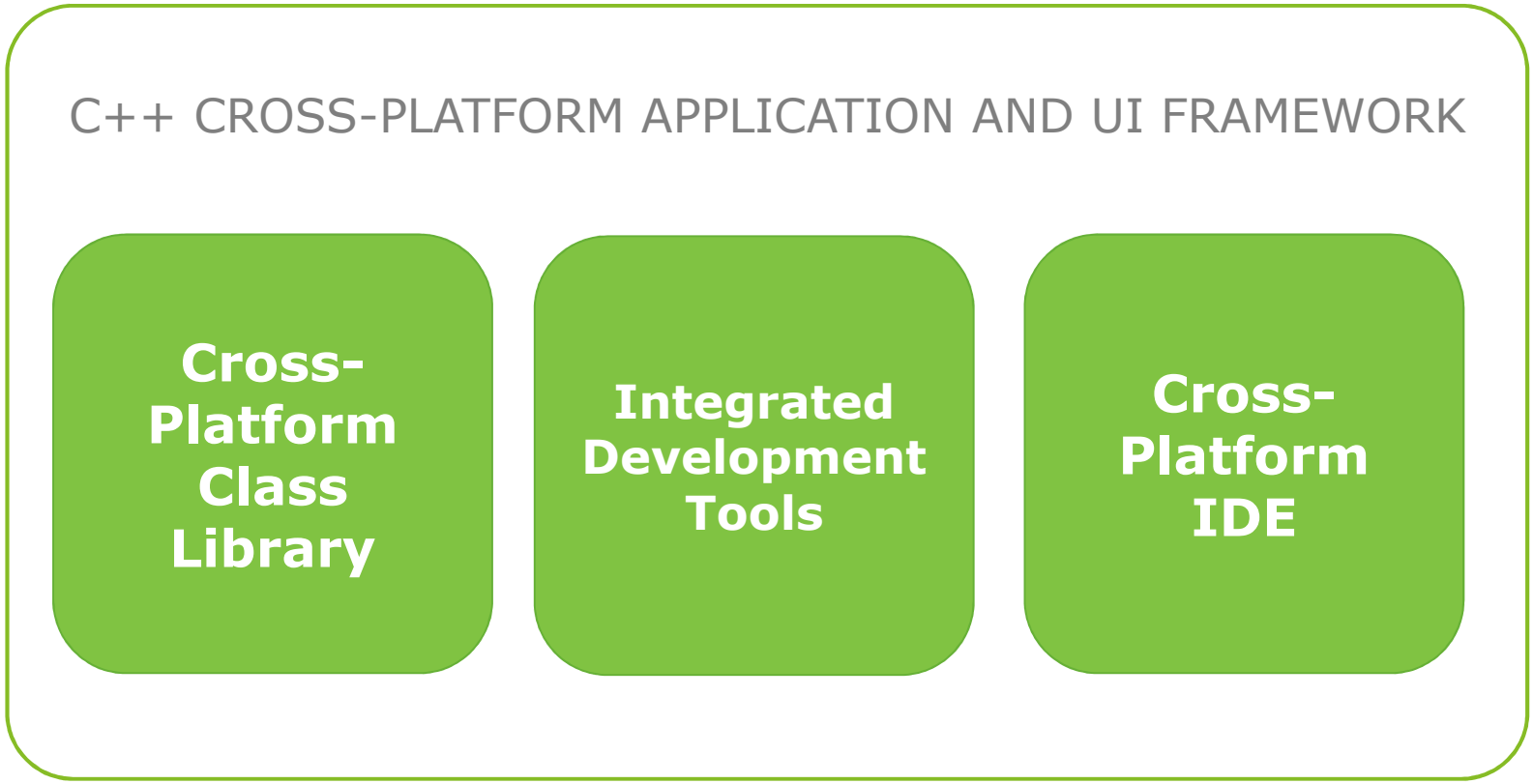
Let’s Talk About Qt

Contents

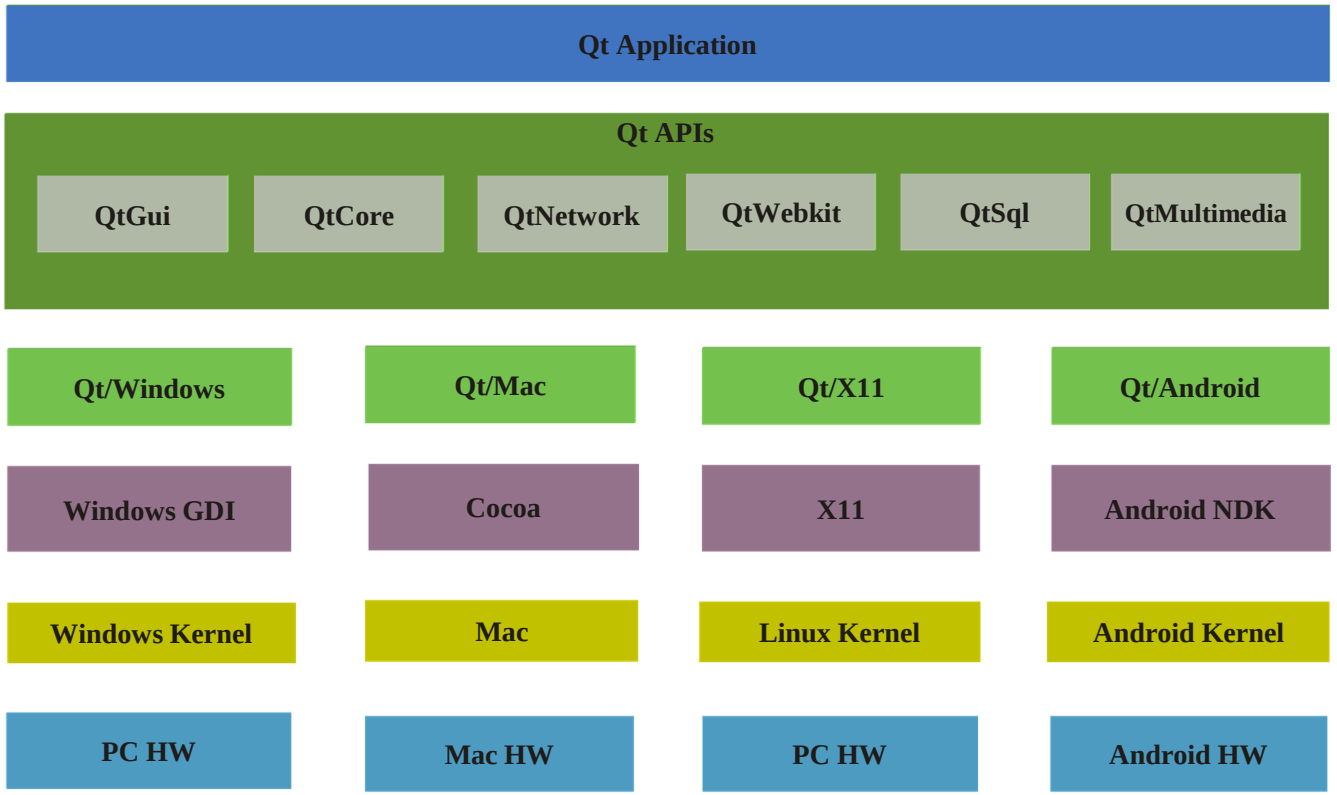
- Brief Introduction to Qt
  - Traditional Qt/C++ with QWidgets
  - Qt Quick and QML
- Qt 5 and Qt Roadmap
- Multi-Screen Approaches with Qt Quick



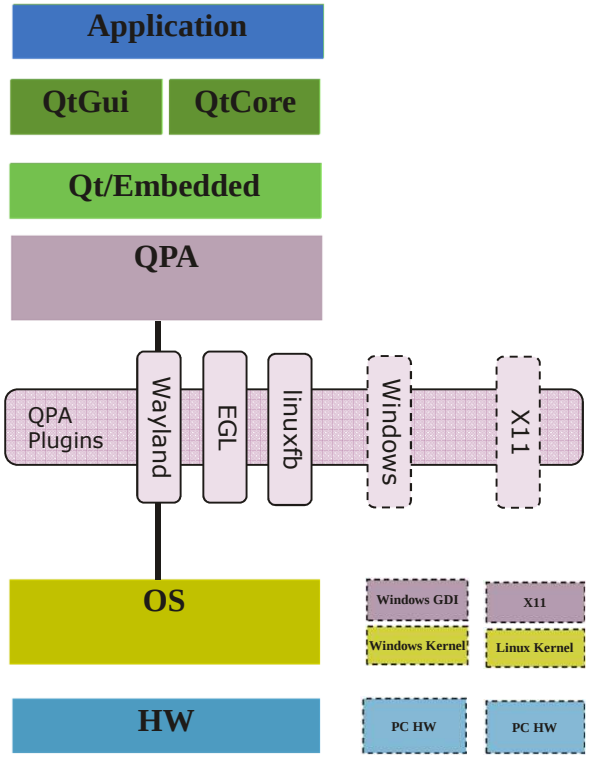
What is Qt?



Qt Applications Are Native Applications



The Qt/Embedded Stack



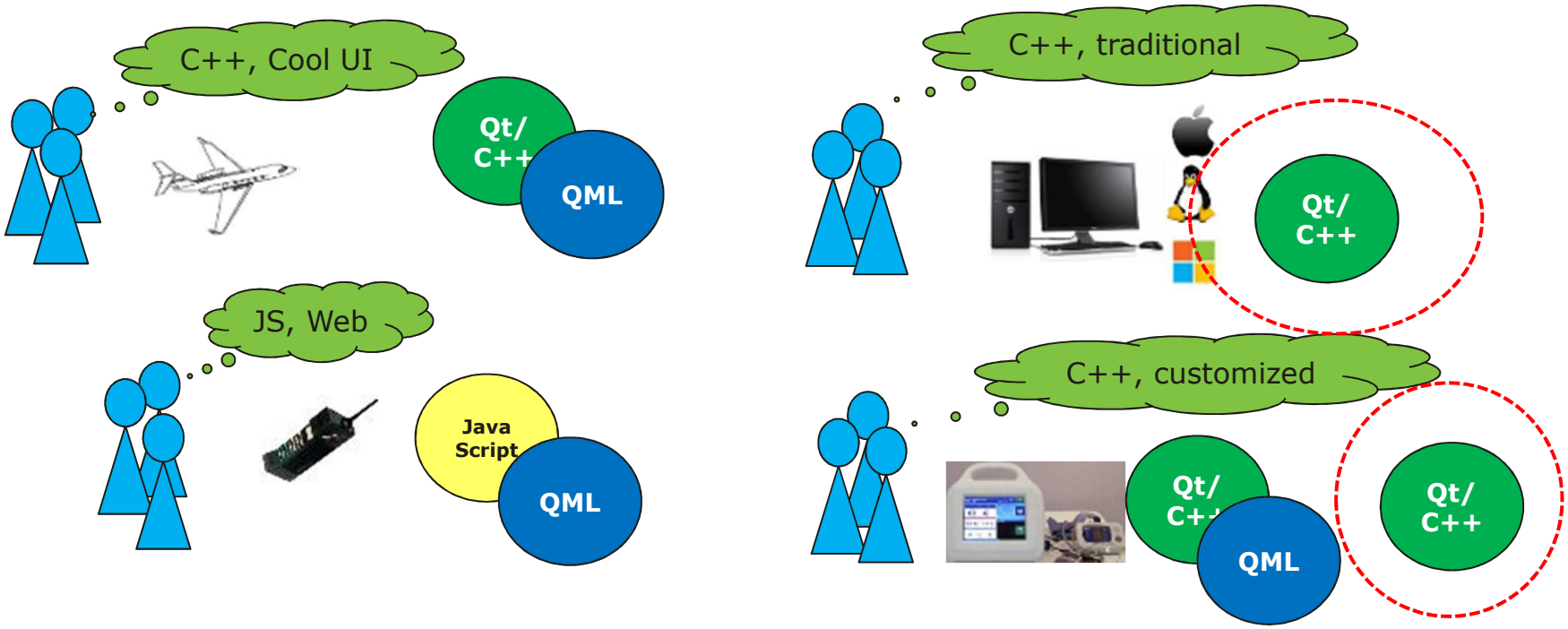
Supported Platforms

Desktop	Embedded	Mobile
Windows	Windows Embedded (Standard/Compact 7)	Android (5.2, beta 5.1)
Linux	Embedded Linux	iOS (5.2, alpha 5.1)
Mac OS X	INTEGRITY	Win8 on ARM (WinRT) (5.2?)
Solaris	QNX	BlackBerry 10
Enterprise UNIX	VxWorks	Jolla Sailfish

Programming with Qt/C++

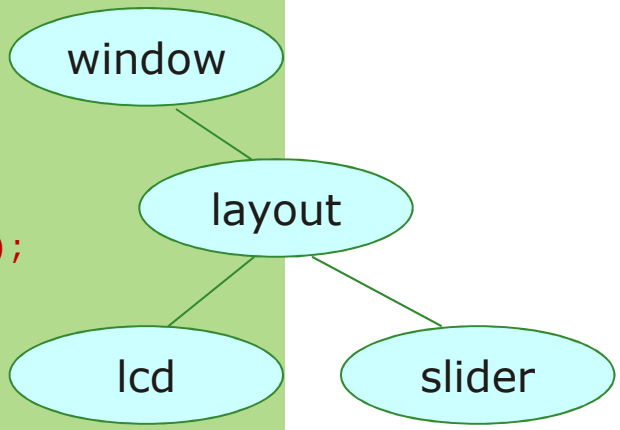
Qt UI Offering - Match the Purpose

Qt provides a set of UI tools and technologies for developers to choose from



Small GUI Example with QWidgets 1/2


```
int main( int argc, char** argv ) {
   QApplication app( argc, argv );

   QWidget window;
   QVBoxLayout* layout = new QVBoxLayout( &window );
   QLCDNumber* lcd = new QLCDNumber( &window );
   QSlider* slider = new QSlider( Qt::Horizontal, &window );
   layout->addWidget( lcd );
   layout->addWidget( slider );

   window.show();
   return app.exec();
}
```



Small GUI Example with QWidgets 2/2

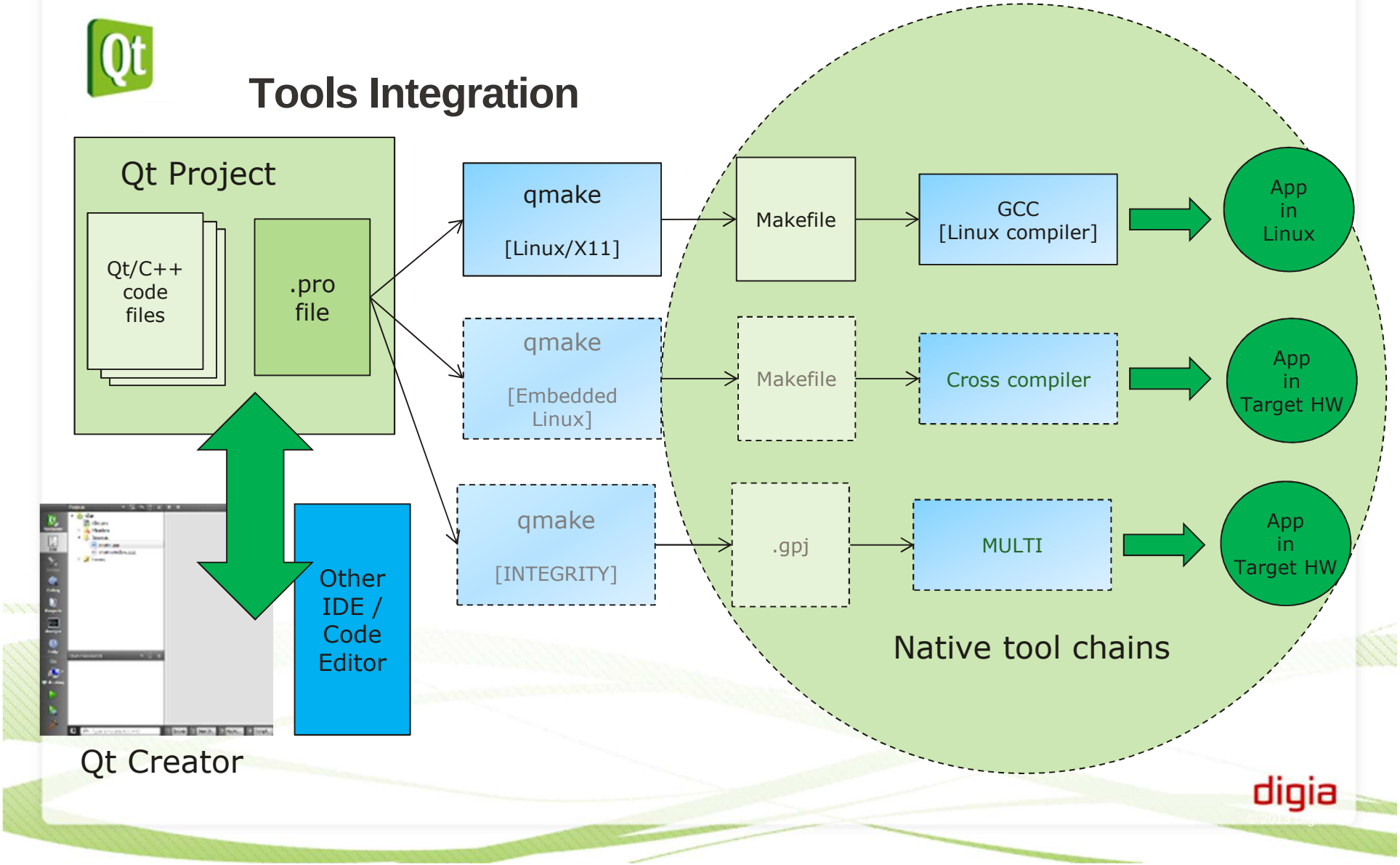
```
int main( int argc, char** argv ) {
   QApplication app( argc, argv );

   QWidget window;
   QVBoxLayout* layout = new QVBoxLayout( &window );
   QLCDNumber* lcd = new QLCDNumber( &window );
   QSlider* slider = new QSlider( Qt::Horizontal, &window );
   layout->addWidget( lcd );
   layout->addWidget( slider );

   QObject::connect( slider, SIGNAL(valueChanged(int)),
                     lcd, SLOT(display(int)));

   window.show();
   return app.exec();
}
```

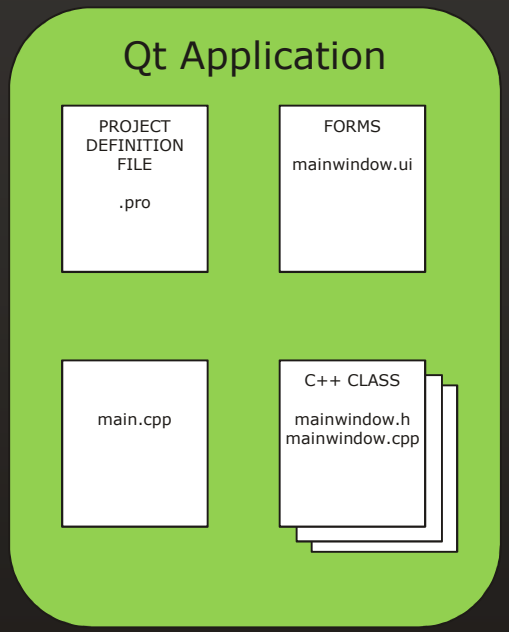
1. Slider is moved (to value 21)

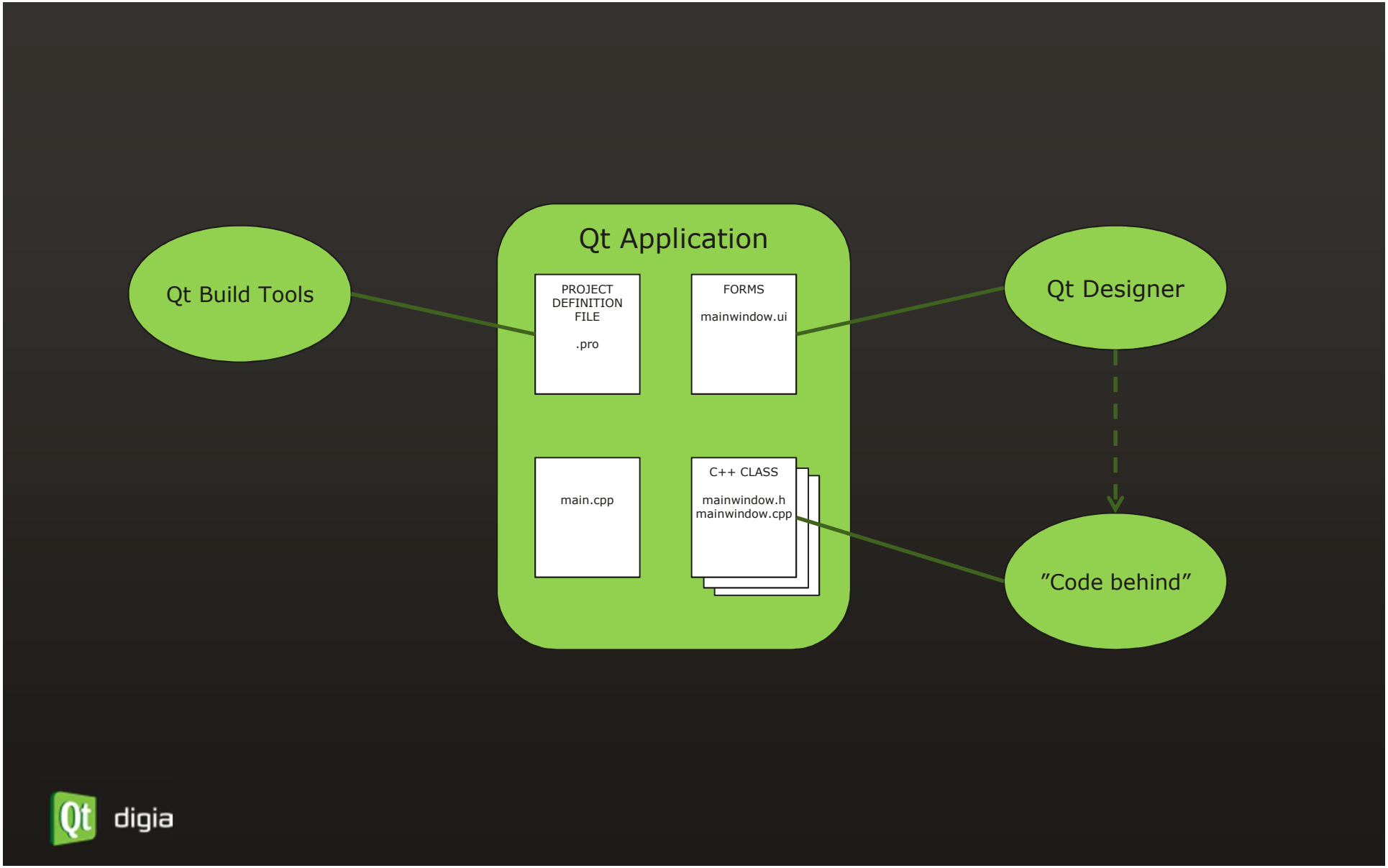
2. `emit valueChanged(21)`

3. `display(21)`

Tools Integration




Qt Application
PROJECT DEFINITION FILE
.pro
FORMS
mainwindow.ui
main.cpp
C++ CLASS
mainwindow.h
mainwindow.cpp


Qt Application
Qt Build Tools
PROJECT DEFINITION FILE
.pro
FORMS
mainwindow.ui
Qt Designer
main.cpp
C++ CLASS
mainwindow.h
mainwindow.cpp
”Code behind”
Qt
digia

Non-GUI Qt/C++ Support

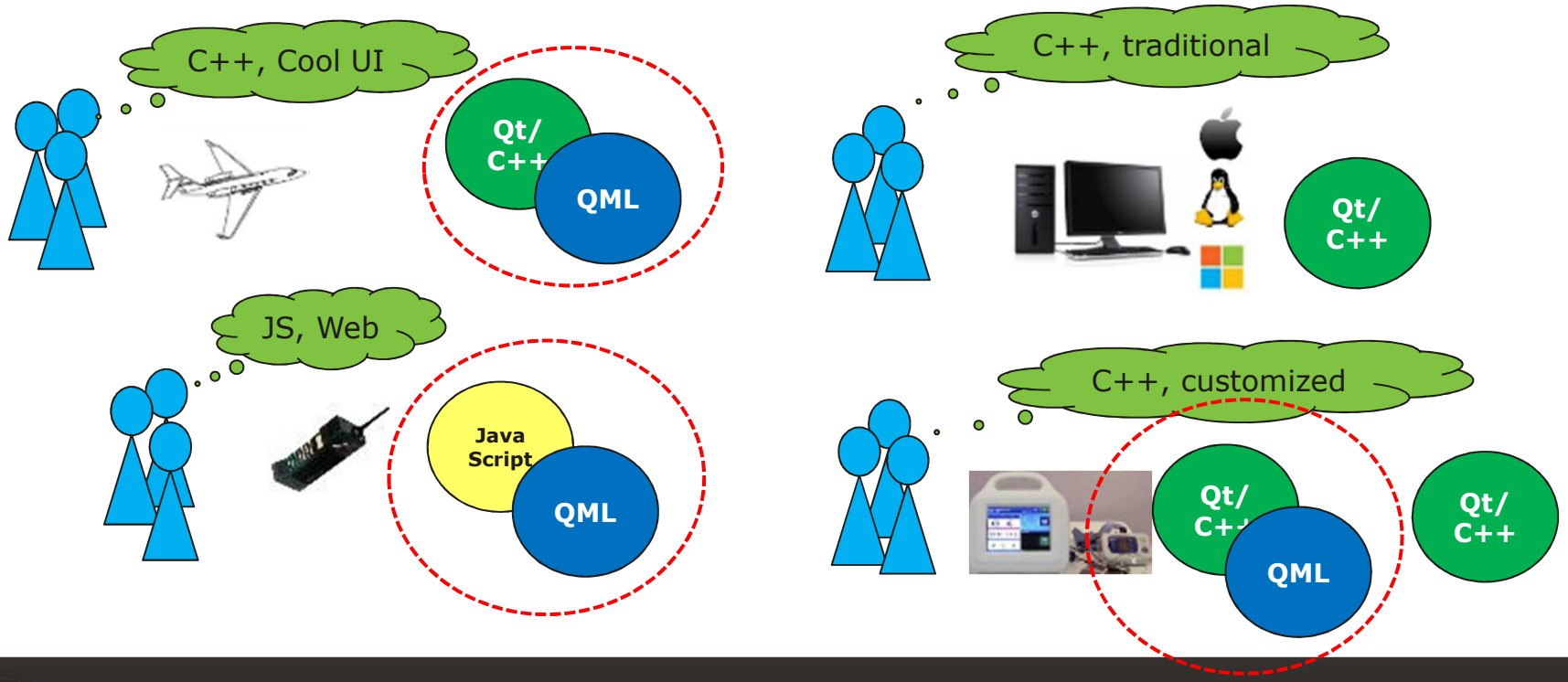
- QtCore
  - Data types, containers
  - Threads, Processes, IPC
  - File I/O
  - String handling
- QtNetwork
  - TCP/UDP, HTTP, FTP, SSL
- QtSql
- QtWebkit
- Qt Serial Port (new in 5.1)
- Etc.

Qt Quick

Bridging the gap between designers and developers

Qt UI Offering - Match the Purpose

Qt provides a set of UI tools and technologies for developers to choose from




NOKIA
Flow'd
Exit
Recent Flow
What's on your mind?
Shout
Antti Vitanen
Jani Manninen
Tommi Laitinen
Ilkka Pärssinen
Hessu Järvinen
Flow
Places
Profile

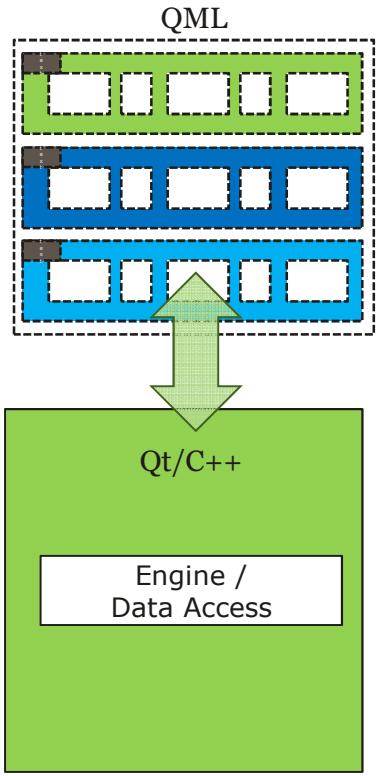

NOKIA
Flow'd
Back
Digia Lappeenranta
Flow
Places
Profile


Ellie Jones
mobile : +356400629926
work mobile : +35650123456


My Books
BOOKS
MAGAZINES
RECENTLY OPENED


ShopWizer
OFFERS
POPULAR
MY ITEMS
BH-905
Nokia
DT-600
Nokia
79€
Nokia
379€
Products
Map
Account
Search

What is Qt Quick?



Declarative UI Design

Imperative Logic

QML at a Glance

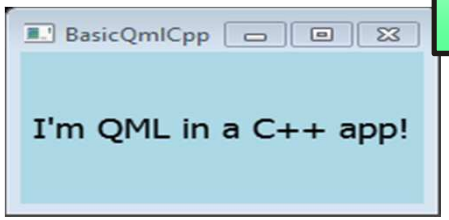
```
import QtQuick 2.1

Rectangle {
   width: 200; height: 200

   Text {
            id: helloText
            anchors.horizontalCenter: parent.horizontalCenter
            font.pixelSize: parent.height / 10
            font.bold: true
            text: “MEET QML!"
   }
   Image {
            id: helloImage
            anchors.top: helloText.bottom
            anchors.horizontalCenter: parent.horizontalCenter
            source: "icons/qt_logo.png"
   }
   MouseArea {
            anchors.fill: parent
            onClicked: {
               helloImage.visible = false;
               helloText.text = "Bye-bye picture!";
     }
   }
}
```

MEET QML!

It’s simply all about elements, properties and their values!

C++ Integration – Minimal Example

```
Rectangle {
   width: 200; height: 100
   color: "lightblue"
   Text {
      anchors.centerIn: parent
      font.pixelSize: 18
      text: "I'm QML in a C++ app!"
   }
}
```



1: `main.qml`
Write your QML UI

```
QT += quick

TARGET = minimalapp
TEMPLATE = app

SOURCES += main.cpp
```

2: `minimalapp.pro`
Link against the QtQuick module

```
#include <QtGui/QGuiApplication>
#include <QtQuick/QQuickView>

int main(int argc, char *argv[]) {
   QGuiApplication a(argc, argv);
   QQuickView v(QUrl::fromLocalFile("main.qml"));
   v.show();
   return a.exec();
}
```

3: `main.cpp`
Use the widget `QQuickView` to display your QML UI

A Bit More QML

Let’s look at some demos!

Qt 5 and Qt Roadmap

Re-Factored Internal Architecture

Same APIs, Re-designed internals
Qt Platform Abstraction
New Platforms

Tailored Parts for Embedded and Mobile

Compelling and performant graphics
New modularization
Better touch support

Modern Graphics Offering

”Velvet-like” Animations
OpenGL based Qt Quick 2
Improved Multimedia
Graphics Effects
OpenGL Shaders

Highlights

Ease-of-Use

Qt 4 compatibility
Qt Creator 2.7
Device Deployment

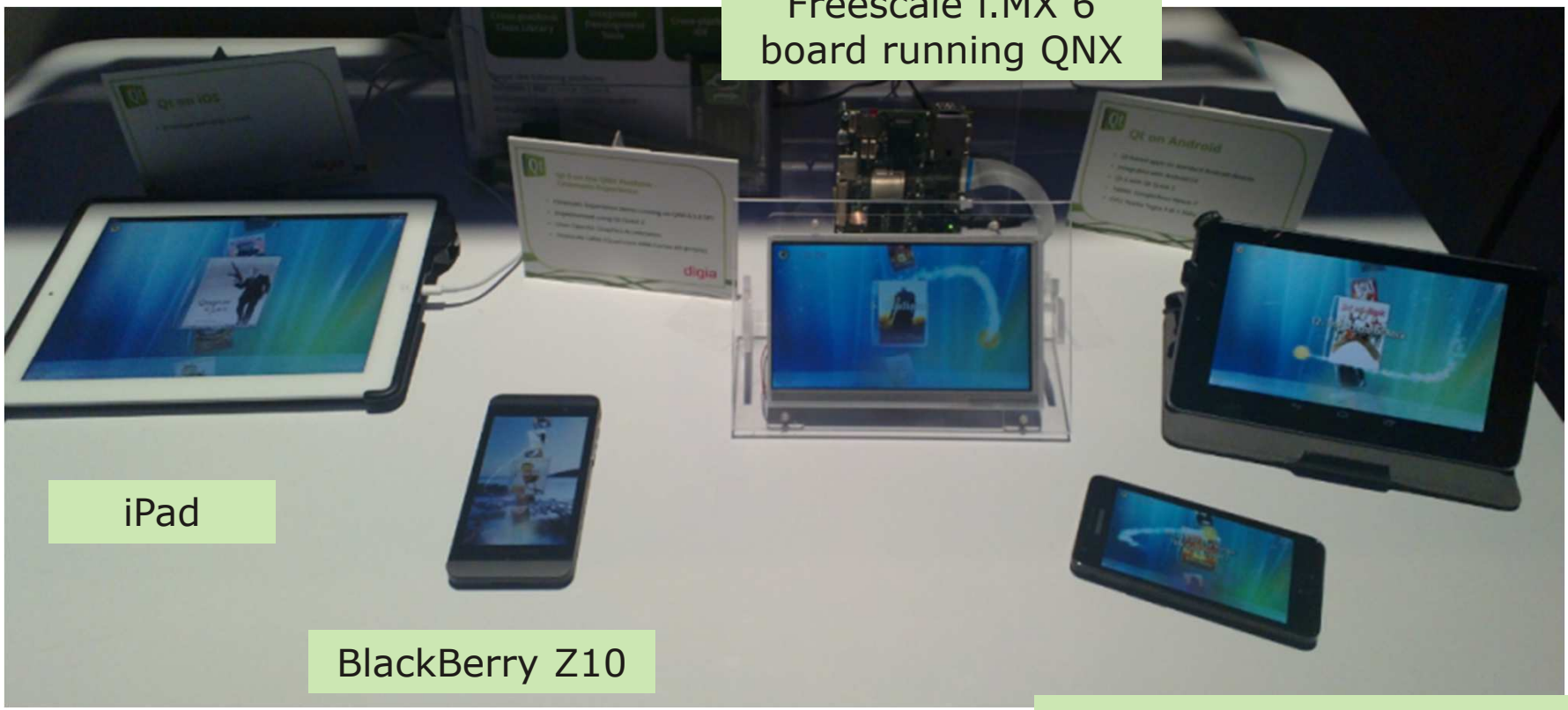
Power of Web Connectivity

Qt Webkit 2
Native JSON support

Qt 5.1 Highlights

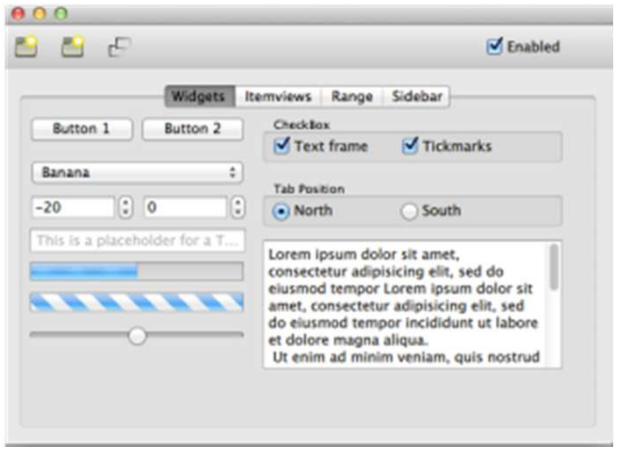
- Performance and Stability
  - 3000+ bug fixes done since 5.0
- Qt Quick Controls
  - Qt Quick just got serious on desktop as well
- Android and iOS as technology preview
  - Let the projects begin!
- New modules
  - Qt Sensors, Qt Serial Port, X11 Extras, Wayland

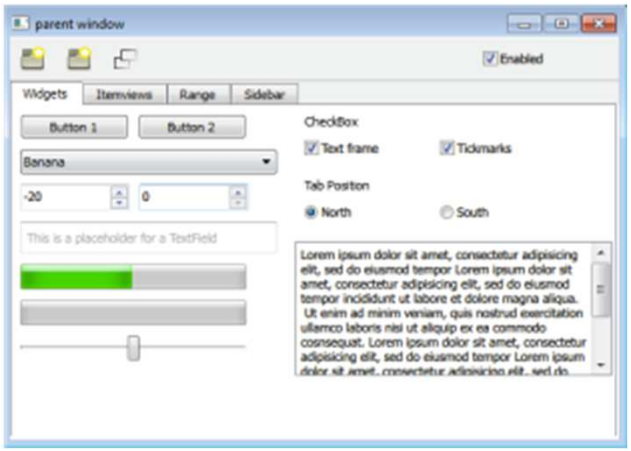
This Is What It's All About!

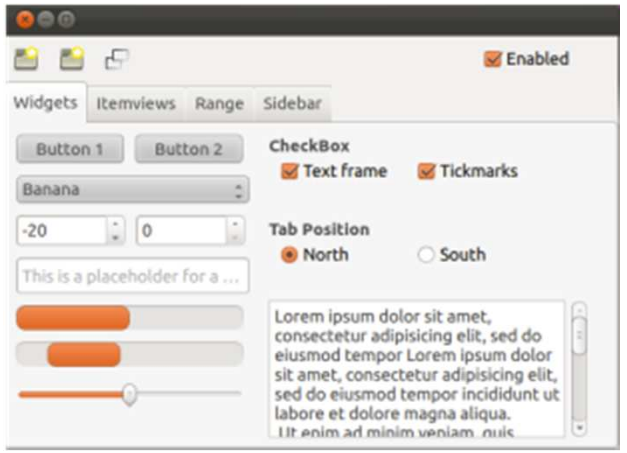


Qt Quick Controls and Layouts

- Finally, a UI control library for QML language (Qt Quick)
- Toolbox of premade, re-usable UI components.
  - 5.1 has desktop controls with native look-and-feel
  - 5.2 will have more, like touch controls, industry specific controls







Qt on New Mobile Platforms – Tech Previews

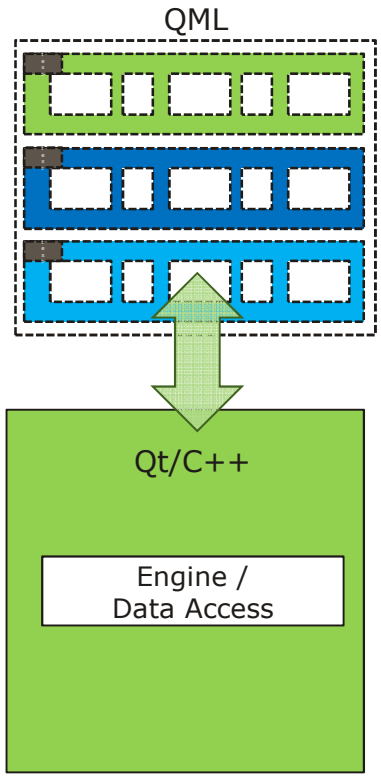
- Qt/Android *(”beta”)*
  - Regular Qt applications running on regular Android OS (phones, tablets)
  - Most of Qt functionality already there
  - No Android functionality (native look-and-feel, back-button, in-app purchase)
  - Qt Creator integration in good shape
  - Deployment (to Google Play) not recommended even though might work

- Qt/iOS *(”alpha”)*
  - Qt applications on iPhone or iPad
  - Some Qt modules already there (Widgets, Qt Quick 1). **No Qt Quick 2 or Webkit**
  - No Qt Creator integration, works in XCode
  - Major internal changes taking place for 5.2 before deployment can be done

Multi-Screen Approaches with Qt

Where Are We Going with All This?

Technical Enablers



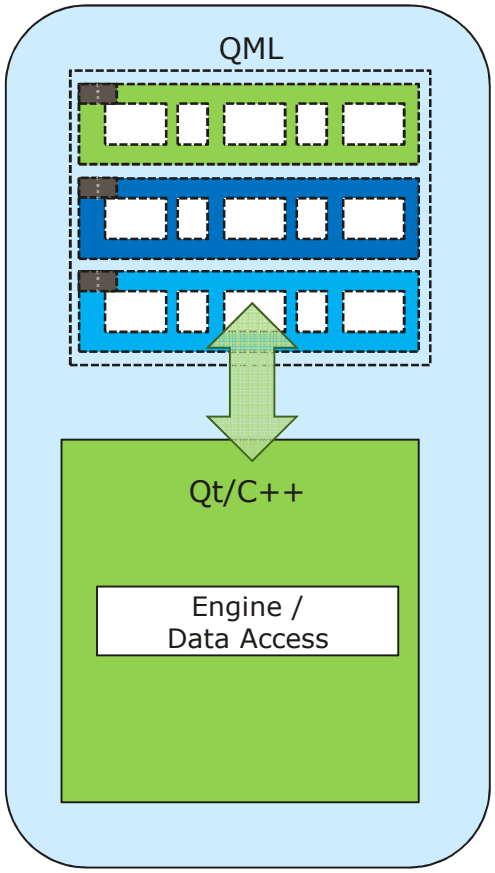
Declarative UI Design

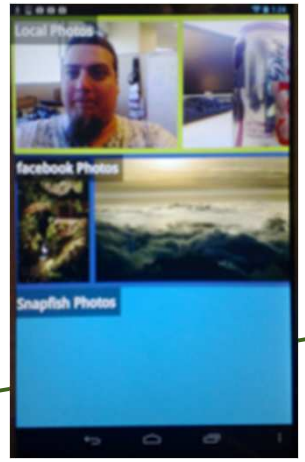
Imperative Logic

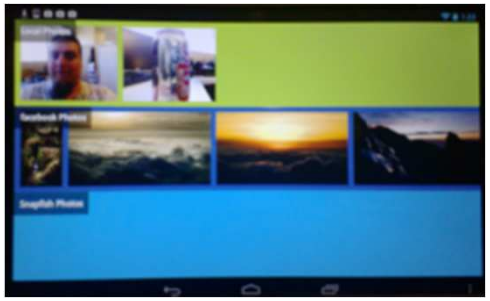
+

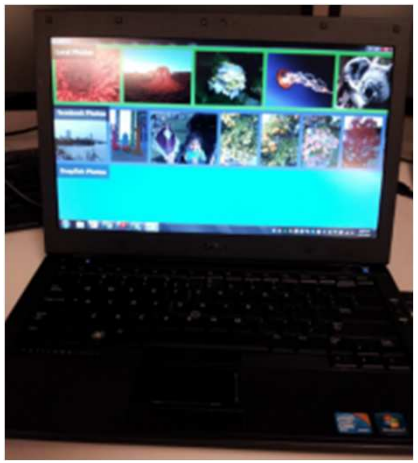
Wide platform support

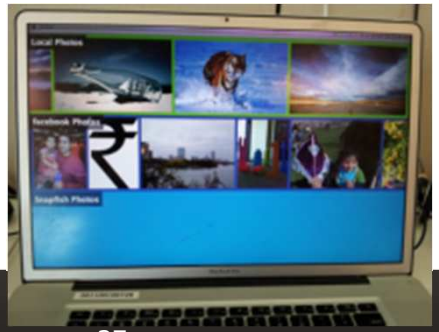
UIs Scale by Default



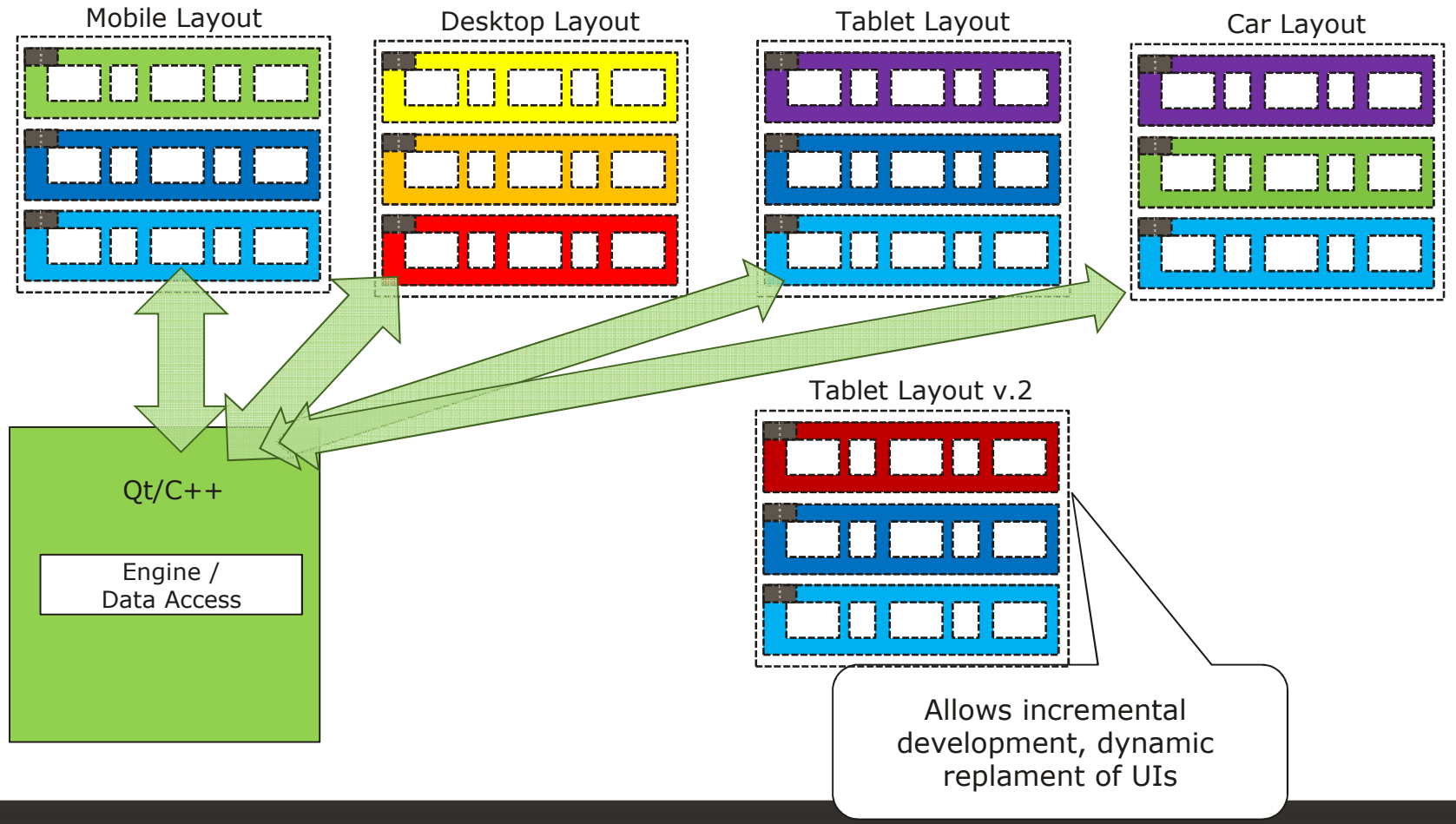




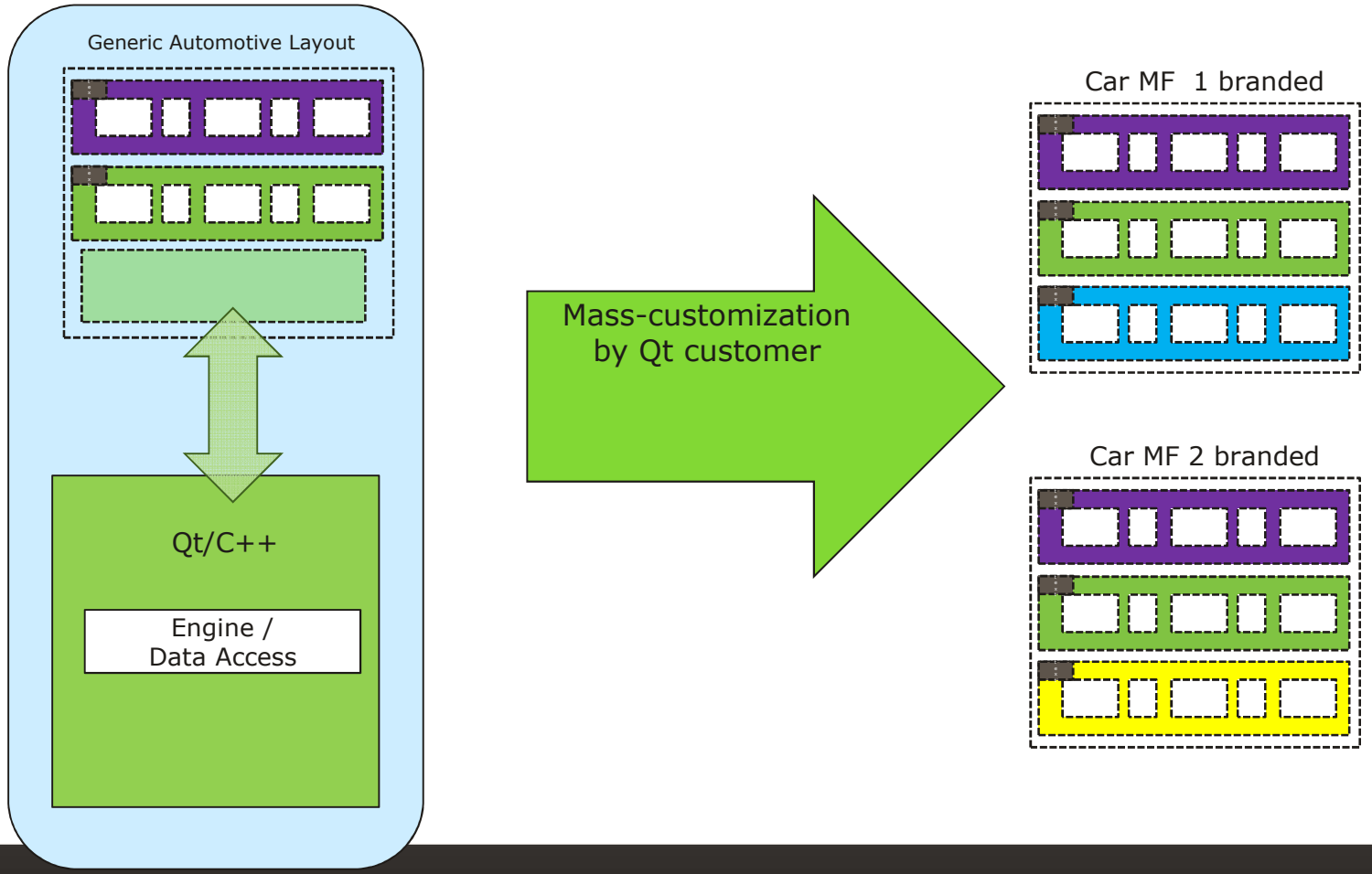




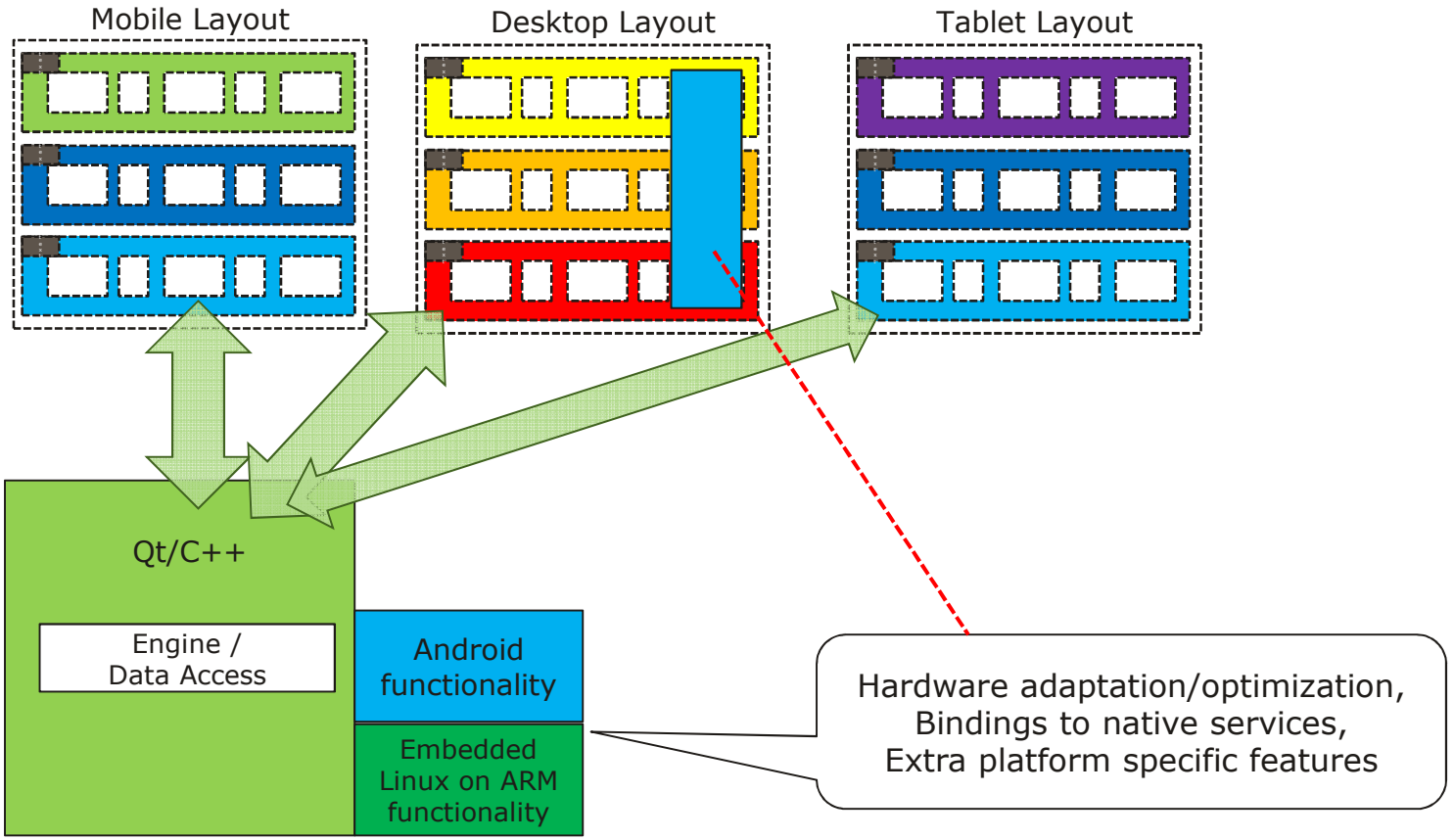
N-Screen Approaches – Different UI Designs



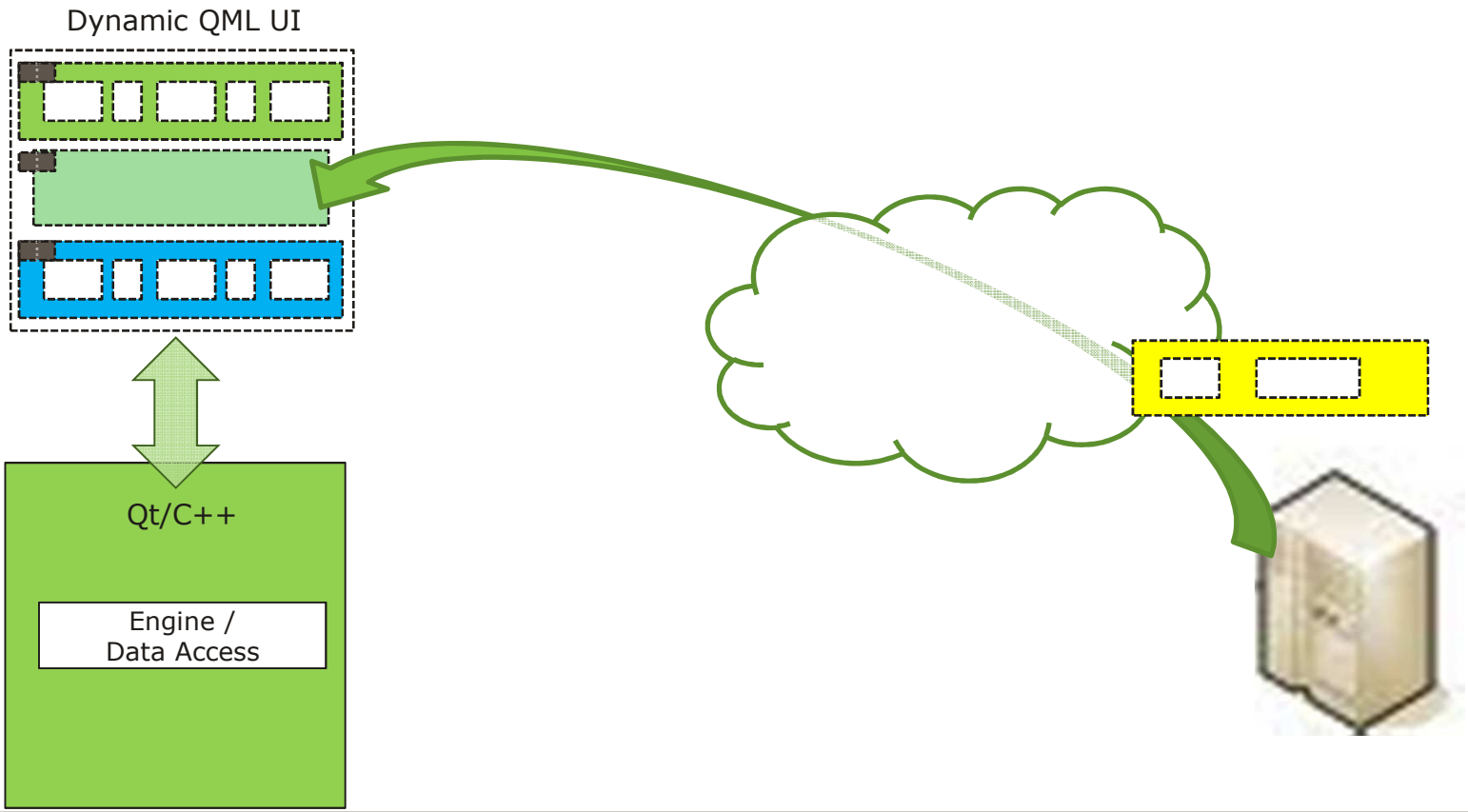
N-Screen Approaches – Mass-Customization



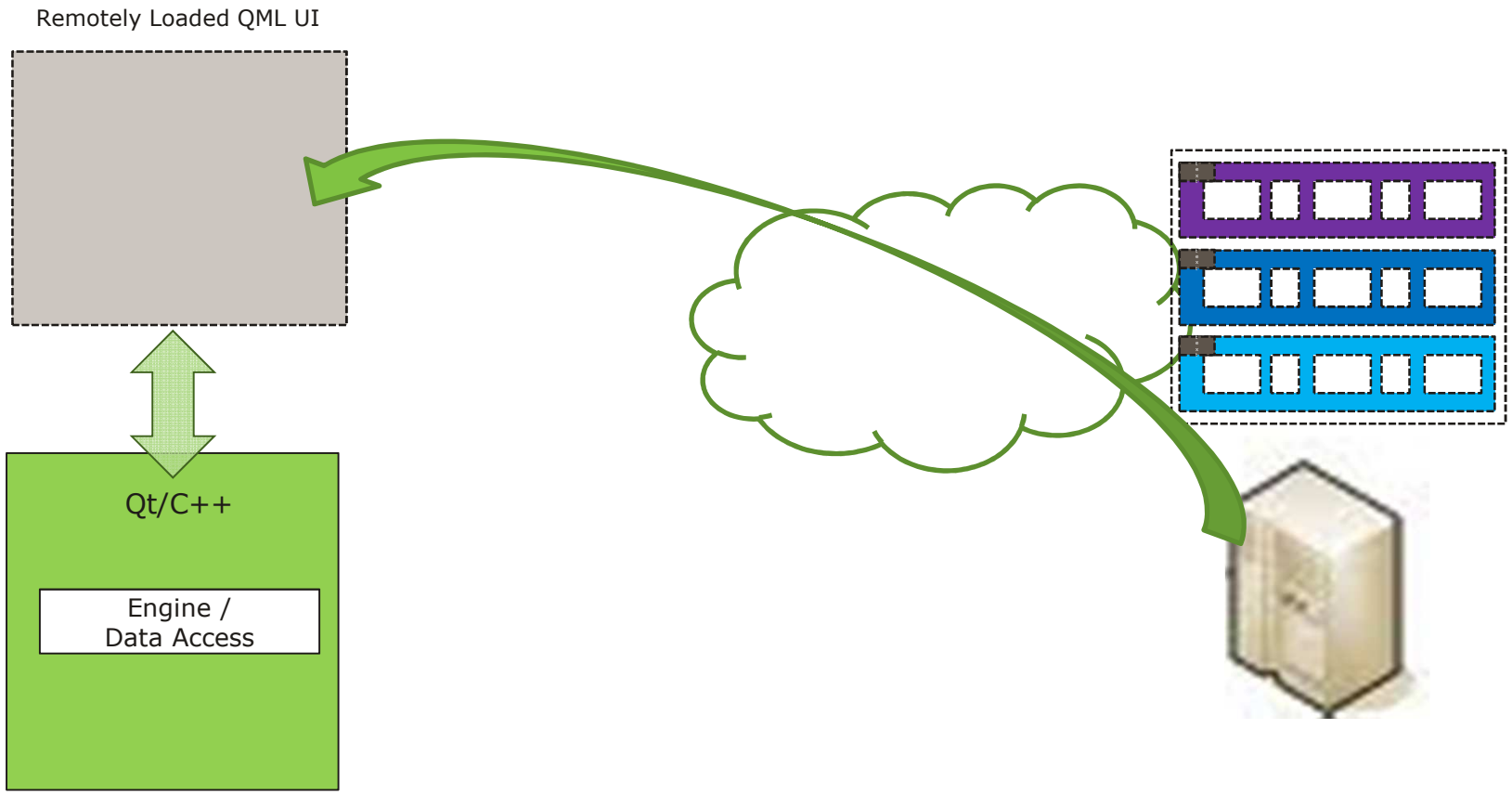
Platform Specific Functionality



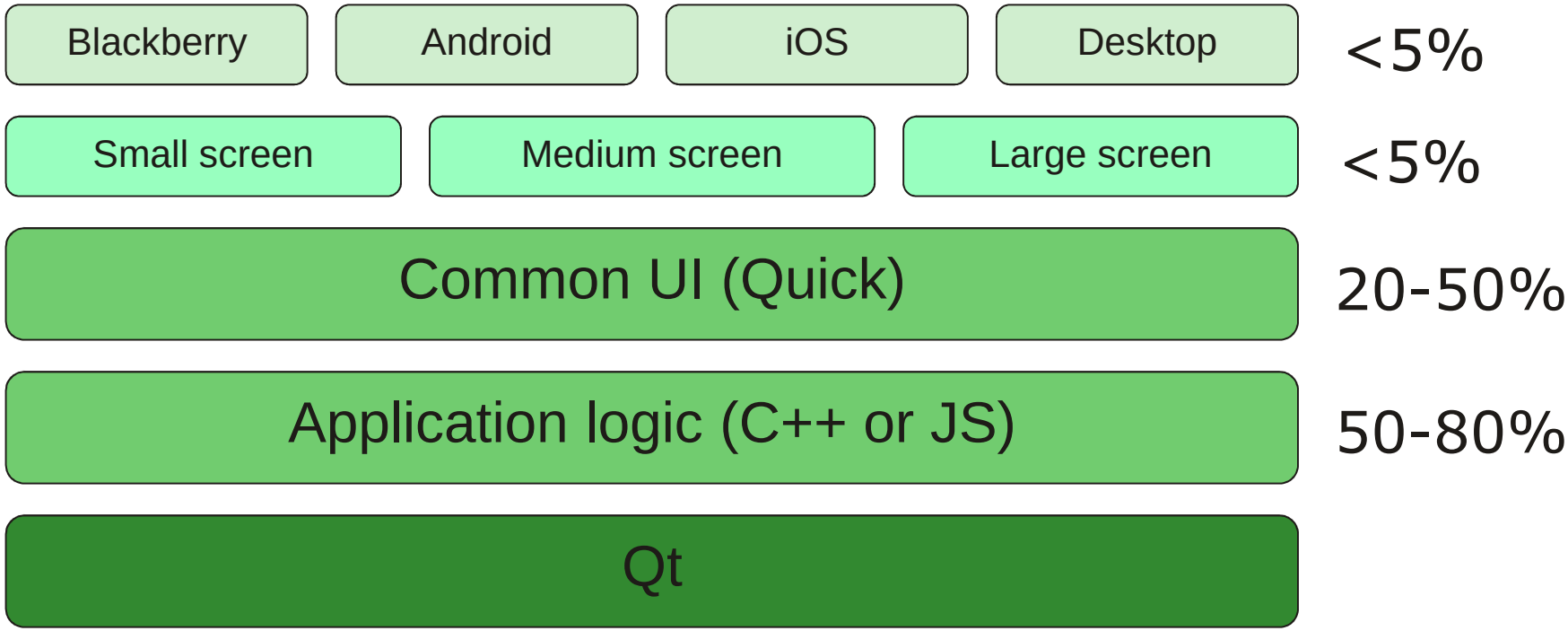
N-Screen Approaches – Partially Remote UI



N-Screen Approaches – Remote UI



Structure of an N-screen App



Summary – Developer Story of Qt in 2013

- Write Once, Deploy Everywhere
  - Desktop, Embedded, Phone
- Technical enablers in place for different multi-screen approaches
  - Maximize code re-use with Qt Quick
- In Native We Trust
  - The Power really lies in C++

Thank You!

tuukka.ahoniemi@digia.com

www.qt.digia.com