

Rapid Application Development with Qt Quick

Henrik Hartz

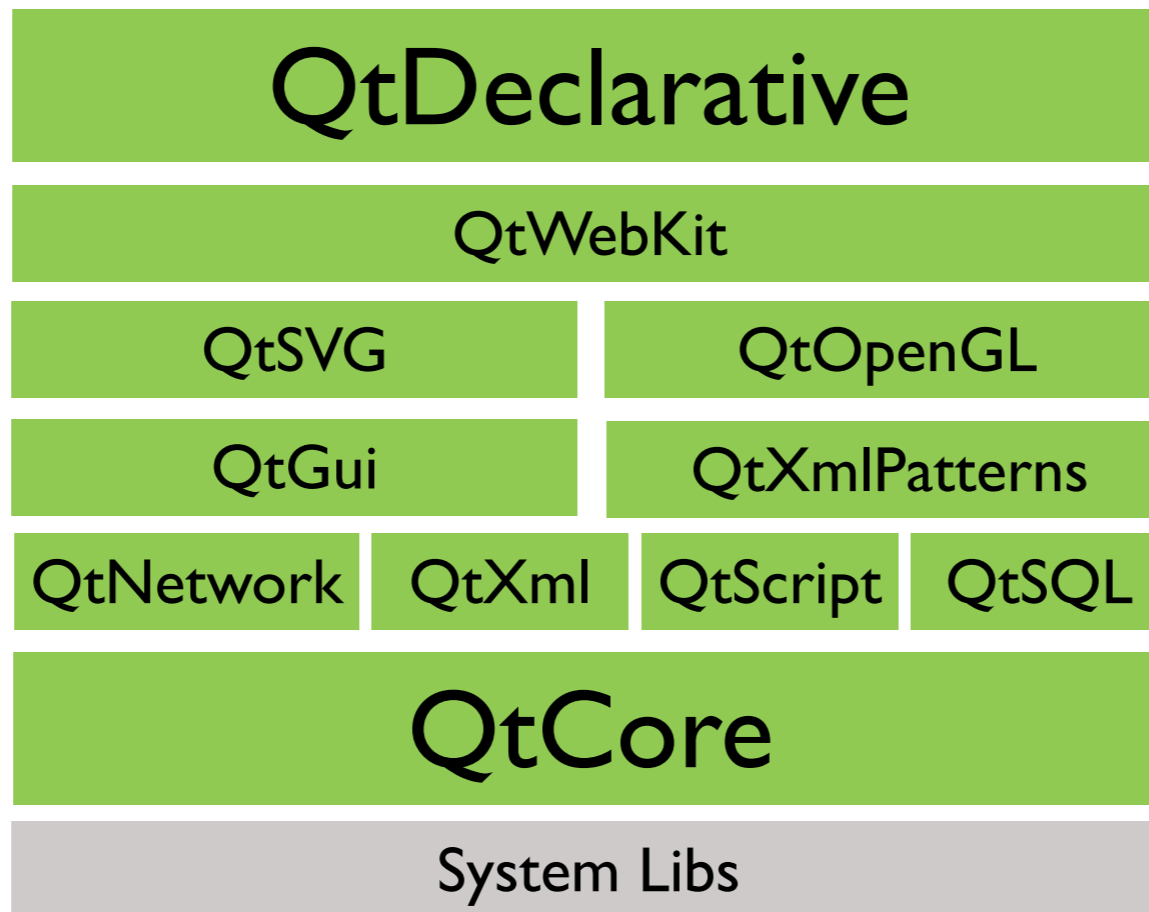
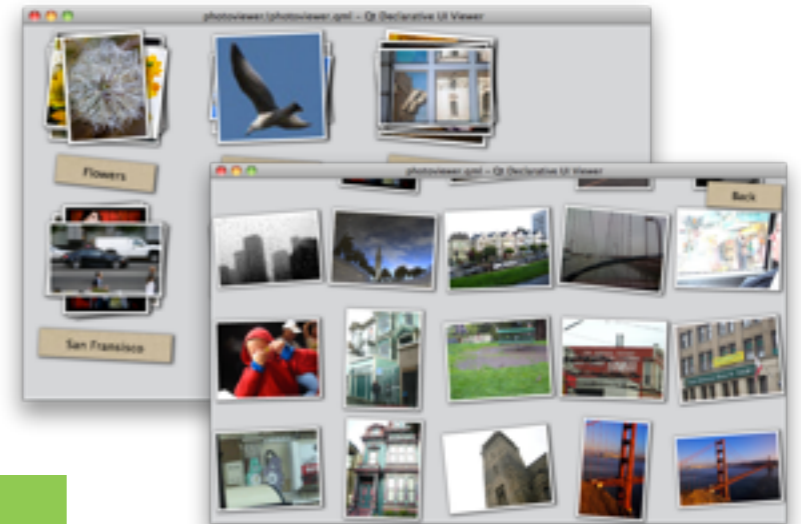
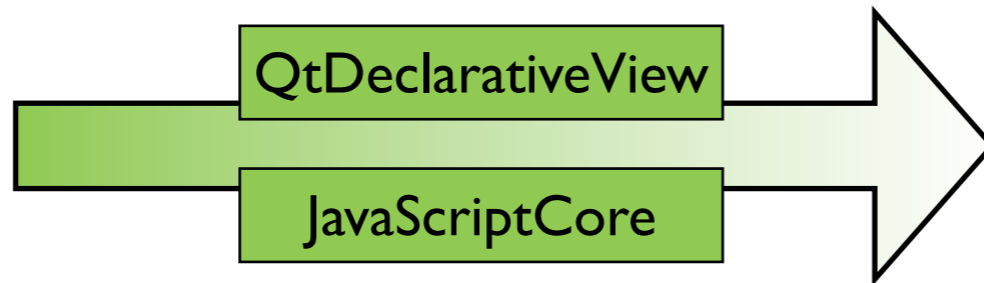
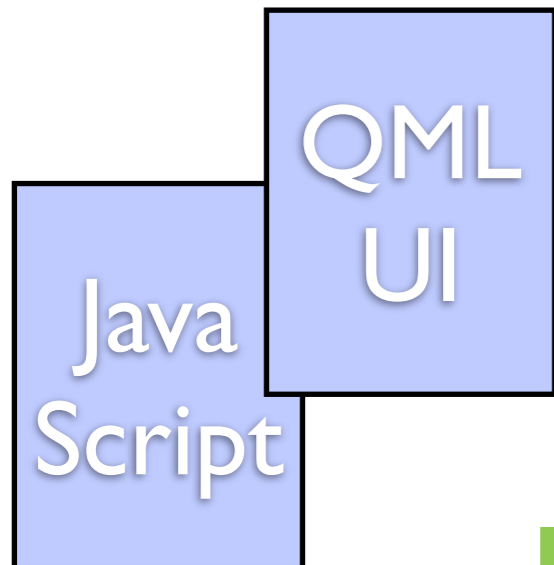
Agenda

- Background
- Structure of a Quick App
- Hello Elements
- Integrating with Native code
- Hello Device
- Adding New Semantics
- Hello Widgets

Background

QWidget	QGraphicsView	Components
Controls come from desktop	Low-level API	Convenience of prebuilt components
Controls are boxed	Close to hardware	C++
Controls are for Mouse	Cumbersome to implement fluid UI	Difficult to customize
Controls are heavyweight	Limited API to push hardware capabilities	UX-specific
	Easy to shoot yourself in the foot	

Qt Quick App



Qt Meta-Object Language

```
import Qt 4.7
import "content"

Rectangle {
    color: "#545454"
    width: 300; height: 300

    // Dial with a slider to adjust it
    Dial { id: dial; anchors.centerIn: parent; value: slider.x *100 / (container.width - 34) }

    Rectangle {
        id: container
        anchors.bottom: parent.bottom; anchors.bottomMargin: 10
        anchors.left: parent.left; anchors.leftMargin: 20
        anchors.right: parent.right; anchors.rightMargin: 20; height: 16
        gradient: Gradient {
            GradientStop { position: 0.0; color: "gray" }
            GradientStop { position: 1.0; color: "white" }
        }
        radius: 8; opacity: 0.7; smooth: true
        Rectangle {
            id: slider
            x: 1; y: 1; width: 30; height: 14
            radius: 6; smooth: true
            gradient: Gradient {
                GradientStop { position: 0.0; color: "#424242" }
                GradientStop { position: 1.0; color: "black" }
            }
            MouseArea {
                anchors.fill: parent
                drag.target: parent; drag.axis: "XAxis"; drag.minimumX: 2; drag.maximumX: container.width - 32
            }
        }
    }
}
```

Qt Meta-Object Language

Built-in Elements

```
import Qt 4.7
import "content"

Rectangle {
    color: "#545454"
    width: 300; height: 300

    // Dial with a slider to adjust it
    Dial { id: dial; anchors.centerIn: parent; value: slider.x *100 / (container.width - 34) }

    Rectangle {
        id: container
        anchors.bottom: parent.bottom; anchors.bottomMargin: 10
        anchors.left: parent.left; anchors.leftMargin: 20
        anchors.right: parent.right; anchors.rightMargin: 20; height: 16
        gradient: Gradient {
            GradientStop { position: 0.0; color: "gray" }
            GradientStop { position: 1.0; color: "white" }
        }
        radius: 8; opacity: 0.7; smooth: true
        Rectangle {
            id: slider
            x: 1; y: 1; width: 30; height: 14
            radius: 6; smooth: true
            gradient: Gradient {
                GradientStop { position: 0.0; color: "#424242" }
                GradientStop { position: 1.0; color: "black" }
            }
            MouseArea {
                anchors.fill: parent
                drag.target: parent; drag.axis: "XAxis"; drag.minimumX: 2; drag.maximumX: container.width - 32
            }
        }
    }
}
}
```

Qt Meta-Object Language

Local Elements

```
import Qt 4.7
import "content"

Rectangle {
    color: "#545454"
    width: 300; height: 300

    // Dial with a slider to adjust it
    Dial { id: dial; anchors.centerIn: parent; value: slider.x *100 / (container.width - 34) }

    Rectangle {
        id: container
        anchors.bottom: parent.bottom; anchors.bottomMargin: 10
        anchors.left: parent.left; anchors.leftMargin: 20
        anchors.right: parent.right; anchors.rightMargin: 20; height: 16
        gradient: Gradient {
            GradientStop { position: 0.0; color: "gray" }
            GradientStop { position: 1.0; color: "white" }
        }
        radius: 8; opacity: 0.7; smooth: true
        Rectangle {
            id: slider
            x: 1; y: 1; width: 30; height: 14
            radius: 6; smooth: true
            gradient: Gradient {
                GradientStop { position: 0.0; color: "#424242" }
                GradientStop { position: 1.0; color: "black" }
            }
            MouseArea {
                anchors.fill: parent
                drag.target: parent; drag.axis: "XAxis"; drag.minimumX: 2; drag.maximumX: container.width - 32
            }
        }
    }
}
```

Qt Meta-Object Language

```
import Qt 4.7
import "content"
```

Toplevel
Window

```
Rectangle {
    color: "#545454"
    width: 300; height: 300

    // Dial with a slider to adjust it
    Dial { id: dial; anchors.centerIn: parent; value: slider.x *100 / (container.width - 34) }

    Rectangle {
        id: container
        anchors.bottom: parent.bottom; anchors.bottomMargin: 10
        anchors.left: parent.left; anchors.leftMargin: 20
        anchors.right: parent.right; anchors.rightMargin: 20; height: 16
        gradient: Gradient {
            GradientStop { position: 0.0; color: "gray" }
            GradientStop { position: 1.0; color: "white" }
        }
        radius: 8; opacity: 0.7; smooth: true
        Rectangle {
            id: slider
            x: 1; y: 1; width: 30; height: 14
            radius: 6; smooth: true
            gradient: Gradient {
                GradientStop { position: 0.0; color: "#424242" }
                GradientStop { position: 1.0; color: "black" }
            }
            MouseArea {
                anchors.fill: parent
                drag.target: parent; drag.axis: "XAxis"; drag.minimumX: 2; drag.maximumX: container.width - 32
            }
        }
    }
}
```


Qt Meta-Object Language

```
import Qt 4.7
import "content"

Rectangle {
    color: "#545454"
    width: 300; height: 300

    // Dial with a slider to adjust it
    Dial { id: dial; anchors.centerIn: parent; value: slider.x *100 / (container.width - 34) }

    Rectangle {
        id: container
        anchors.bottom: parent.bottom; anchors.bottomMargin: 10
        anchors.left: parent.left; anchors.leftMargin: 20
        anchors.right: parent.right; anchors.rightMargin: 20; height: 16
        gradient: Gradient {
            GradientStop { position: 0.0; color: "gray" }
            GradientStop { position: 1.0; color: "white" }
        }
        radius: 8; opacity: 0.7; smooth: true
        Rectangle {
            id: slider
            x: 1; y: 1; width: 30; height: 14
            radius: 6; smooth: true
            gradient: Gradient {
                GradientStop { position: 0.0; color: "#424242" }
                GradientStop { position: 1.0; color: "black" }
            }
            MouseArea {
                anchors.fill: parent
                drag.target: parent; drag.axis: "XAxis"; drag.minimumX: 2; drag.maximumX: container.width - 32
            }
        }
    }
}
}
```

Basic
properties

Qt Me

Language

```
import Qt 4.7
import "content"

Rectangle {
    color: "#545454";
    width: 300; height: 300;

    // Dial with
    Dial { id: dial; }

    Rectangle {
        id: container;
        anchors.fill: parent;
        anchors.margins: 10;
        anchors.top: dial;
        gradient: Gradient {
            GradientStop { position: 0; color: "#545454"; }
            GradientStop { position: 1; color: "#545454"; }
        }
        radius: 8;
        Rectangle {
            id: shadow;
            x: 10; y: 10;
            radius: 8;
            gradient: Gradient {
                GradientStop { position: 0; color: "#545454"; }
                GradientStop { position: 1; color: "#545454"; }
            }
        }
        MouseArea {
            anchors.fill: parent;
            onClicked: {
                root.value = root.value + 1;
            }
        }
    }
}
}
```

Custom Type →

```
import Qt 4.7

Item {
    id: root
    property real value : 0

    width: 210; height: 210

    Image { source: "background.png" }

    Image {
        x: 93
        y: 35
        source: "needle_shadow.png"
        transform: Rotation {
            origin.x: 11; origin.y: 67
            angle: needleRotation.angle
        }
    }
    Image {
        id: needle
        x: 95; y: 33
        smooth: true
        source: "needle.png"
        transform: Rotation {
            id: needleRotation
            origin.x: 7; origin.y: 65
            angle: -130
            SpringFollow on angle {
                spring: 1.4
                damping: .15
                to: Math.min(Math.max(-130, root.value*2.6 - 130), 133)
            }
        }
    }
    Image { x: 21; y: 18; source: "overlay.png" }
}
```

34) }

container.width - 32

Qt Meta-Object Language

```
import Qt 4.7
import "content"

Rectangle {
    color: "#545454"
    width: 300; height: 300

    // Dial with a slider to adjust it
    Dial { id: dial; anchors.centerIn: parent; value: slider.x *100 / (container.width - 34) }

    Rectangle {
        id: container
        anchors.bottom: parent.bottom; anchors.bottomMargin: 10
        anchors.left: parent.left; anchors.leftMargin: 20
        anchors.right: parent.right; anchors.rightMargin: 20; height: 16
        gradient: Gradient {
            GradientStop { position: 0.0; color: "gray" }
            GradientStop { position: 1.0; color: "white" }
        }
        radius: 8; opacity: 0.7; smooth: true
        Rectangle {
            id: slider
            x: 1; y: 1; width: 30; height: 14
            radius: 6; smooth: true
            gradient: Gradient {
                GradientStop { position: 0.0; color: "#424242" }
                GradientStop { position: 1.0; color: "black" }
            }
            MouseArea {
                anchors.fill: parent
                drag.target: parent; drag.axis: "XAxis"; drag.minimumX: 2; drag.maximumX: container.width - 32
            }
        }
    }
}
}
```

Anchor Lines
w/Margins →

Qt Meta-Object Language

```
import Qt 4.7
import "content"

Rectangle {
    color: "#545454"
    width: 300; height: 300

    // Dial with a slider to adjust it
    Dial { id: dial; anchors.centerIn: parent; value: slider.x *100 / (container.width - 34) }

    Rectangle {
        id: container
        anchors.bottom: parent.bottom; anchors.bottomMargin: 10
        anchors.left: parent.left; anchors.leftMargin: 20
        anchors.right: parent.right; anchors.rightMargin: 20; height: 16
        gradient: Gradient {
            GradientStop { position: 0.0; color: "gray" }
            GradientStop { position: 1.0; color: "white" }
        }
        radius: 8; opacity: 0.7; smooth: true
        Rectangle {
            id: slider
            x: 1; y: 1; width: 30; height: 14
            radius: 6; smooth: true
            gradient: Gradient {
                GradientStop { position: 0.0; color: "#424242" }
                GradientStop { position: 1.0; color: "black" }
            }
            MouseArea {
                anchors.fill: parent
                drag.target: parent; drag.axis: "XAxis"; drag.minimumX: 2; drag.maximumX: container.width - 32
            }
        }
    }
}
}
```

Mouse
Interaction area

Qt Meta-Object Language

```
import Qt 4.7
import "content"

Rectangle {
    color: "#545454"
    width: 300; height: 300

    // Dial with a slider to adjust it
    Dial { id: dial; anchors.centerIn: parent; value: slider.x *100 / (container.width - 34) }

    Rectangle {
        id: container
        anchors.bottom: parent.bottom; anchors.bottomMargin: 10
        anchors.left: parent.left; anchors.leftMargin: 20
        anchors.right: parent.right; anchors.rightMargin: 20; height: 16
        gradient: Gradient {
            GradientStop { position: 0.0; color: "gray" }
            GradientStop { position: 1.0; color: "white" }
        }
        radius: 8; opacity: 0.7; smooth: true
        Rectangle {
            id: slider
            x: 1; y: 1; width: 30; height: 14
            radius: 6; smooth: true
            gradient: Gradient {
                GradientStop { position: 0.0; color: "#424242" }
                GradientStop { position: 1.0; color: "black" }
            }
            MouseArea {
                anchors.fill: parent
                drag.target: parent; drag.axis: "XAxis"; drag.minimumX: 2; drag.maximumX: container.width - 32
            }
        }
    }
}
```

Code Walkthrough

- (MeeGo?) RSS viewer
 - Written in-flight; ~4 hrs
 - Single window, 2 lists
 - Graphics borrowed

Hello Elements

- Item, Rectangle
- MouseArea, LineEdit
- Model, View
- BorderImage, Gradient, Rotation

What's Quick about it?

- Simplicity of language and intuitive structure
 - Created with Designers in mind
 - Easy to pick up
 - Good separation between code and presentation
- Designers and developers in same environment
 - Common understanding of semantics
 - Design around same constraints
- Hardware accelerated
 - OpenGL (ES), OpenVG

States and Transitions

- State represents a configuration of properties for some items
 - PropertyChanges, ParentChange, AnchorChanges
- Transitions define interpolation between two states
 - PropertyAnimation, SequentialAnimation/ParallelAnimation, PauseAnimation

Integrating with Native Code

- QtDeclarative == Qt, native C++
 - QGraphicsView runtime environment populated by QDeclarativeEngine::load()
 - All QMetaObject information exposed
 - Signals, Slots, Properties
- ➔ Easy to bind to native business logic

Hello Device

- QML UI 'runtime' identical on Desktop and Device
- Device-specific features can be guarded and added in executable
- Device-specific packaging apply
- Lets do it..



Adding New Semantics

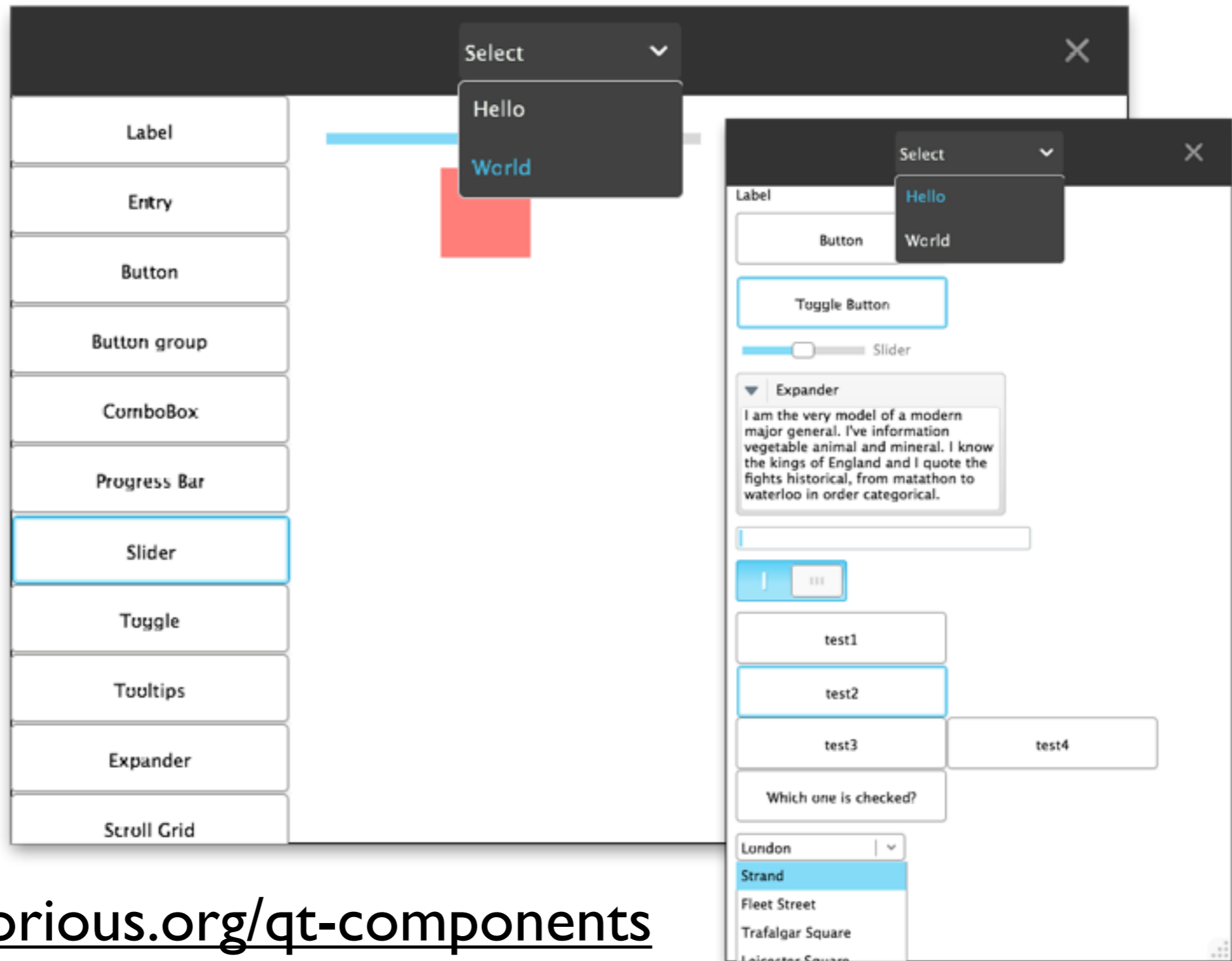
- Any QGraphicsItem can be exposed

```
#define QML_DECLARE_TYPE(T)
```

```
int qmlRegisterType ( const char * uri,  
                    int versionMajor,  
                    int versionMinor,  
                    const char * qmlName )
```

- Must subclass QObject, default-constructor
- Properties can be bound to if NOTIFY is present

Hello Widgets



<http://qt.gitorious.org/qt-components>

Roadmap

- ✓ Emulate MeeGo Netbook UX with Qt (QML)
- WIP components to enable easy app creation, starting with embedded/mobile space
 - MeeGo handset-optimized
 - ...but will work on others too
- Enable bindings of other UX (QGraphicsView-based) to QML
- Working on controller API for UI
 - i.e. models pluggable to multiple UX
 - <http://qt.gitorious.org/qt-components>

Where, What, How?

- <http://qt.nokia.com/developer/qt-roadmap>
- <http://qt.nokia.com/doc/4.7-snapshot/declarativeui.html>
- 4.7 branch of <git://gitorious.org/qt/qt.git>
- http://labs.trolltech.com/blogs/category/labs/graphics/kinetic/declarative_ui/
- <http://twitter.com/#search?q=QML>
- <http://chaos.troll.no/~hhartz/QtQuickAtLinuxCollaborationSummit2010.pdf>
- #qt-qml on irc.freenode.org