Qt in Education

An Introduction to Qt and MeeGo

NOKIA

Introduction

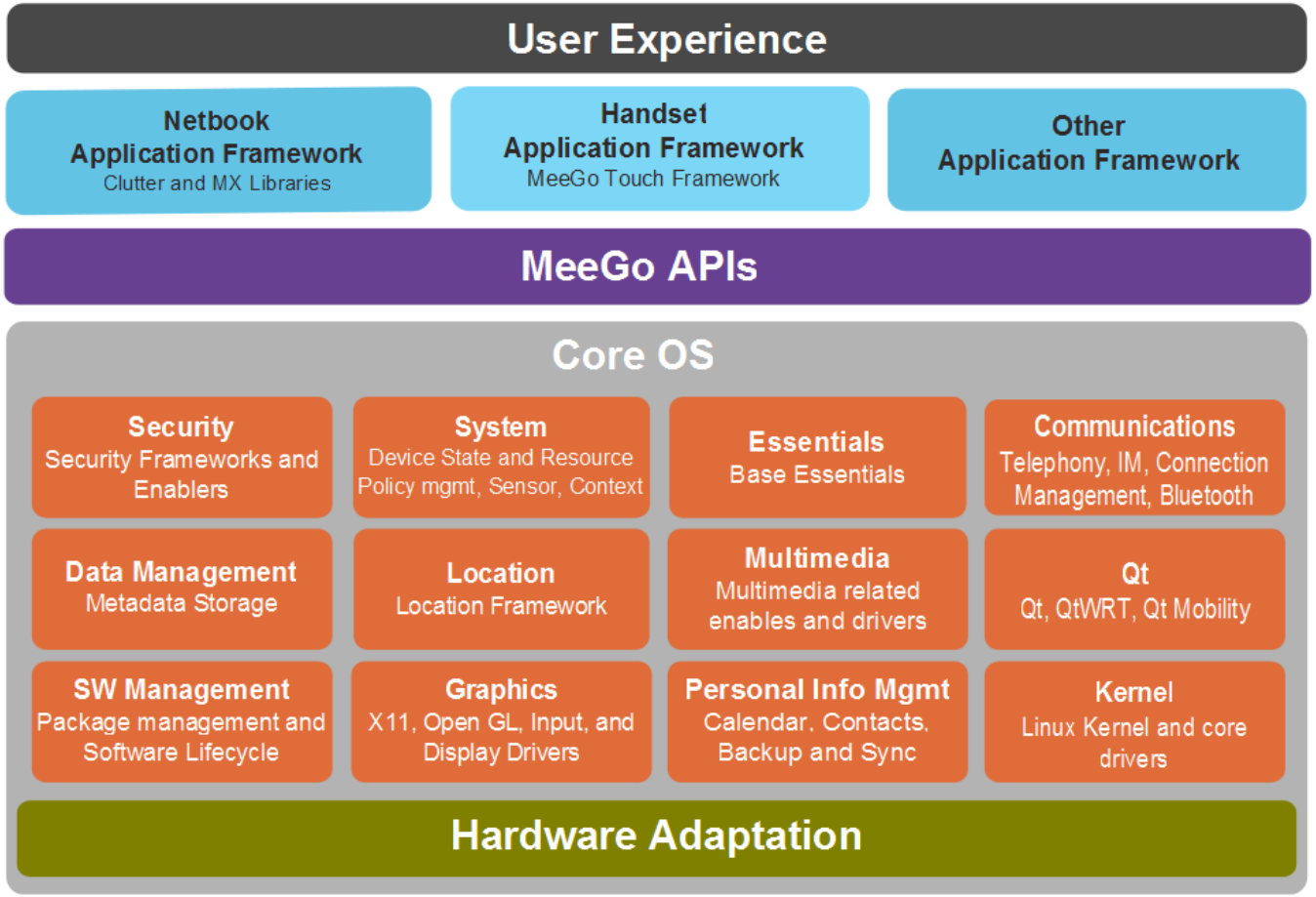
MeeGo 

MeeGo

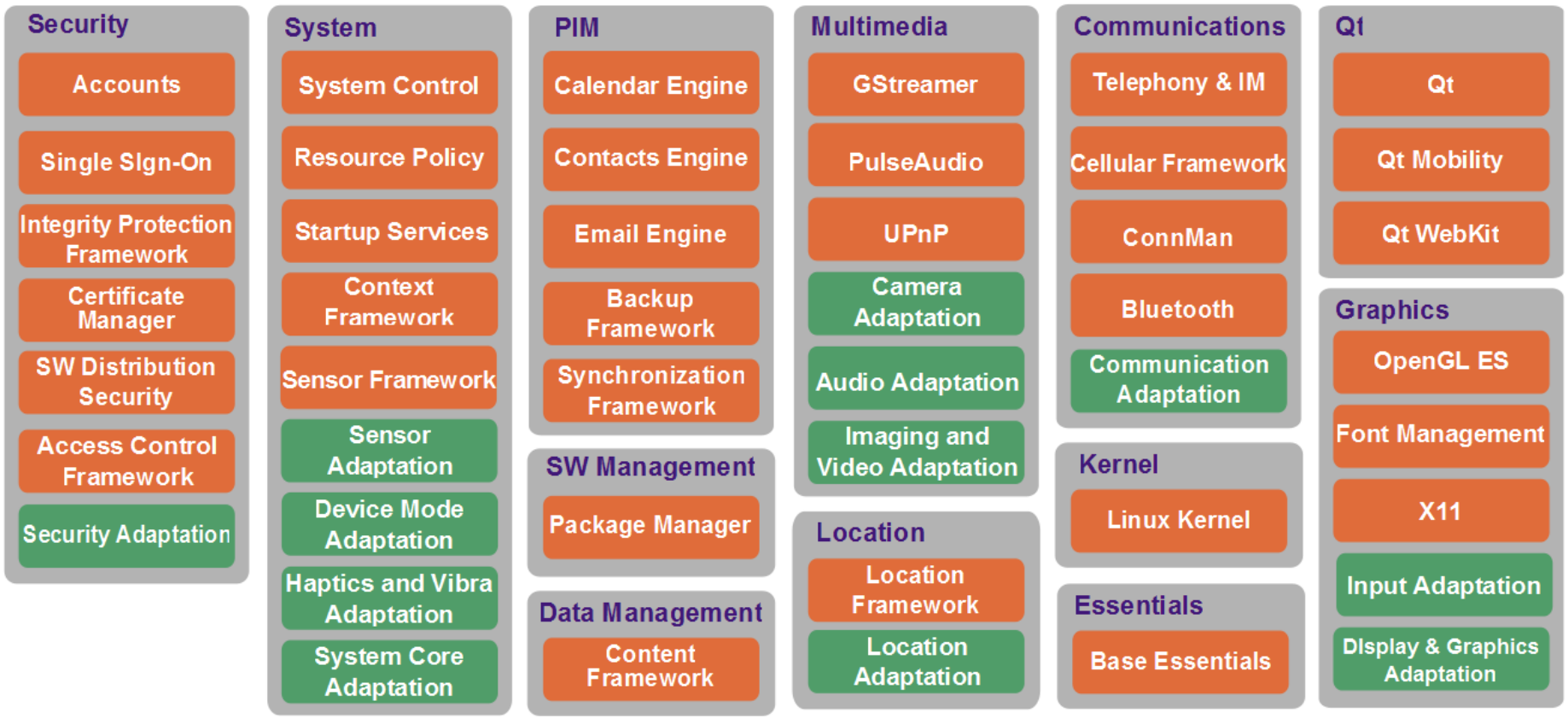
- MeeGo is a Linux distribution
  - Founded by Nokia and Intel

  - Hosted by the Linux Foundation

MeeGo


User Experience
Netbook
Application Framework
Clutter and MX Libraries
Handset
Application Framework
MeeGo Touch Framework
Other
Application Framework
MeeGo APIs
Core OS
Security
Security Frameworks and Enablers
System
Device State and Resource Policy mgmt, Sensor, Context
Essentials
Base Essentials
Communications
Telephony, IM, Connection Management, Bluetooth
Data Management
Metadata Storage
Location
Location Framework
Multimedia
Multimedia related enables and drivers
Qt
Qt, QtWRT, Qt Mobility
SW Management
Package management and Software Lifecycle
Graphics
X11, Open GL, Input, and Display Drivers
Personal Info Mgmt
Calendar, Contacts, Backup and Sync
Kernel
Linux Kernel and core drivers
Hardware Adaptation

MeeGo


Security
Accounts
Single SIgn-On
Integrity Protection Framework
Certificate Manager
SW Distribution Security
Access Control Framework
Security Adaptation
System
System Control
Resource Policy
Startup Services
Context Framework
Sensor Framework
Sensor Adaptation
Device Mode Adaptation
Haptics and Vibra Adaptation
System Core Adaptation
PIM
Calendar Engine
Contacts Engine
Email Engine
Backup Framework
Synchronization Framework
SW Management
Package Manager
Data Management
Content Framework
Multimedia
GStreamer
PulseAudio
UPnP
Camera Adaptation
Audio Adaptation
Imaging and Video Adaptation
Location
Location Framework
Location Adaptation
Communications
Telephony & IM
Cellular Framework
ConnMan
Bluetooth
Communication Adaptation
Kernel
Linux Kernel
Essentials
Base Essentials
Qt
Qt
Qt Mobility
Qt WebKit
Graphics
OpenGL ES
Font Management
X11
Input Adaptation
DIsplay & Graphics Adaptation

User Experiences

- The user interface used on top of the MeeGo core is commonly referred to as a User Experience – UX
  - A UX is more than a user interface
  - It defines not only the look of an application, but the entire interaction experience
    - Integration of applications
    - Common interface methods
    - Integration with the physical device
    - etc

Available UXes

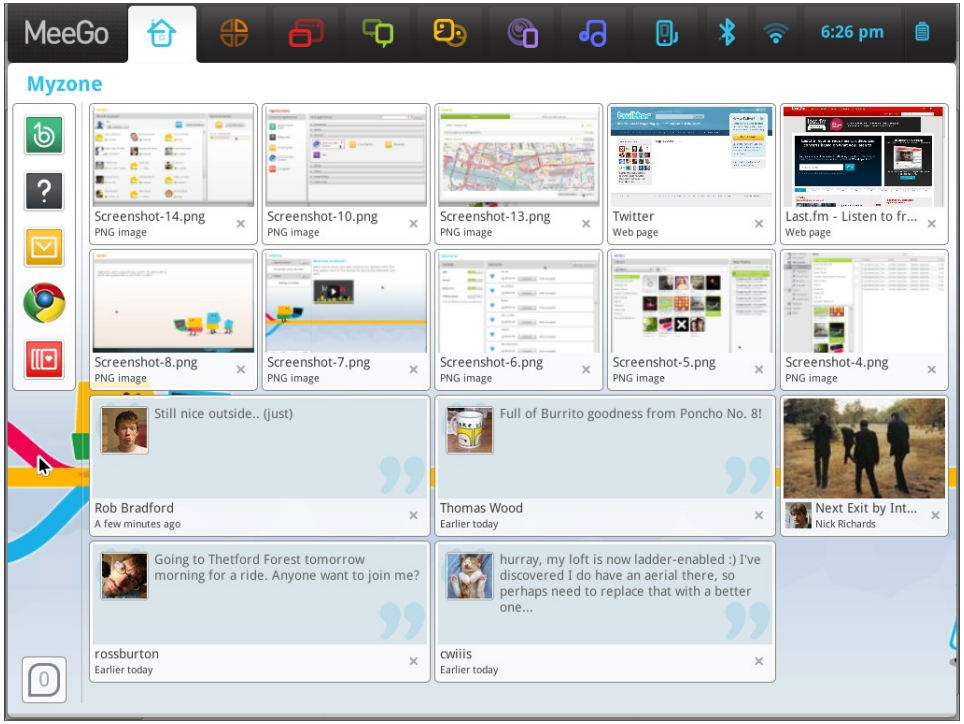
- Netbook
- Tablet
- Handset
- In-vehicle Infotainment
- Smart TV
- Media Phone





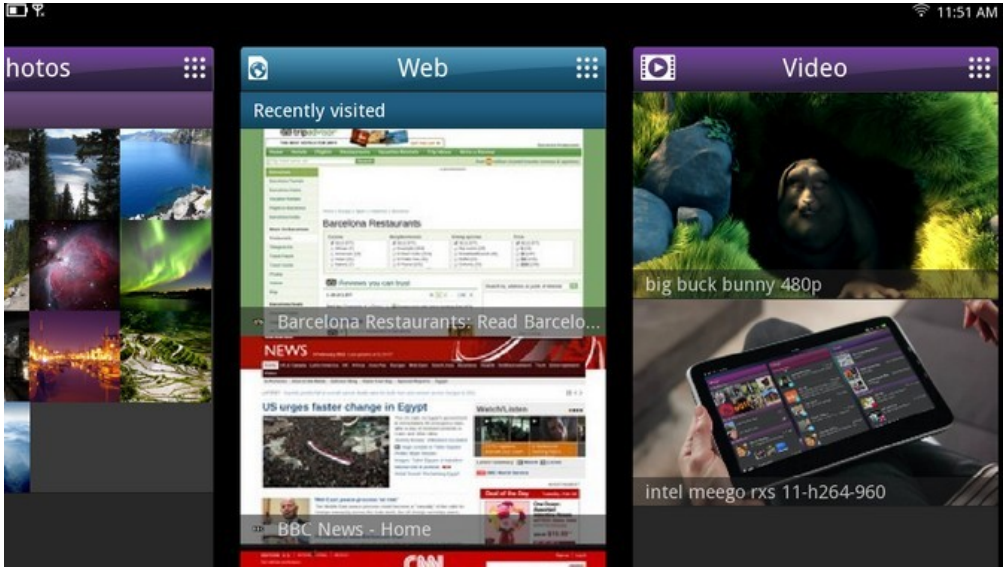

UX: Netbook

- Windows are maximized or full screen
- Devices have keyboards
- Devices have pointer devices
- Devices might have touch screens



UX: Tablet

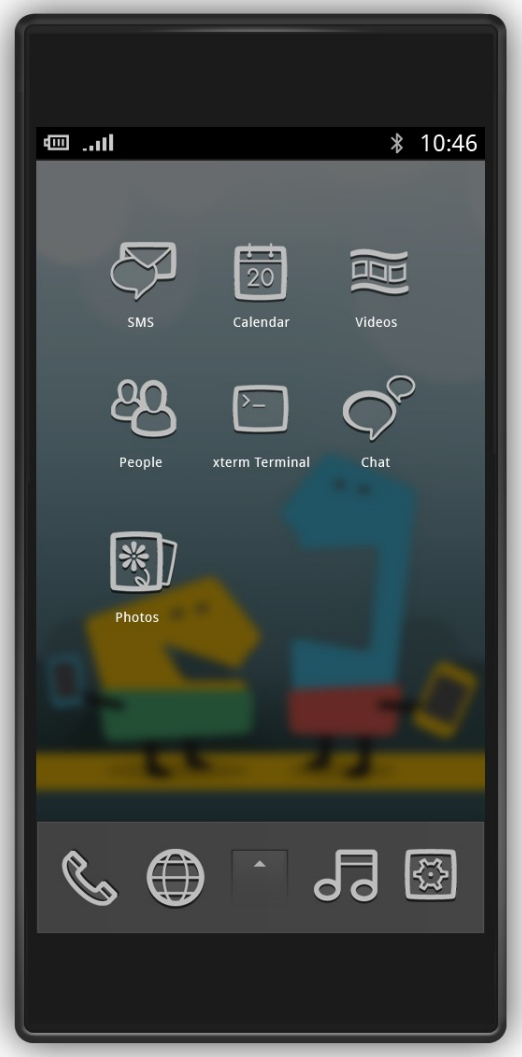
- Windows are maximized or full screen
- The screen orientation can change



- Devices might not have keyboards
- Devices have (multipoint) touch screens

UX: Handset

- Windows are maximized or full screen
- The screen orientation can change
- Devices are likely not to have keyboards, or to have limited keyboards
- Devices have touch screens



UX: In-Vehicle Infotainment

- Windows are maximized or full screen
- Interaction can be limited to a very limited set of buttons

UX: Smart TV / Media Phone

- The Smart TV and Media Phone UXes are work in progress
- Targets media streaming, video conferencing, entertainment, etc

UX Comparison Chart

	Netbook	Tablet	Handset	IVI
Screen resolution	medium	high	medium	medium
Screen orientation	landscape	landscape / portrait	landscape / portrait	landscape
Keyboard	yes	no	limited / no	limited / no
Pointer device	yes	no	no	no
Touch	optional	yes	yes	optional

Why MeeGo

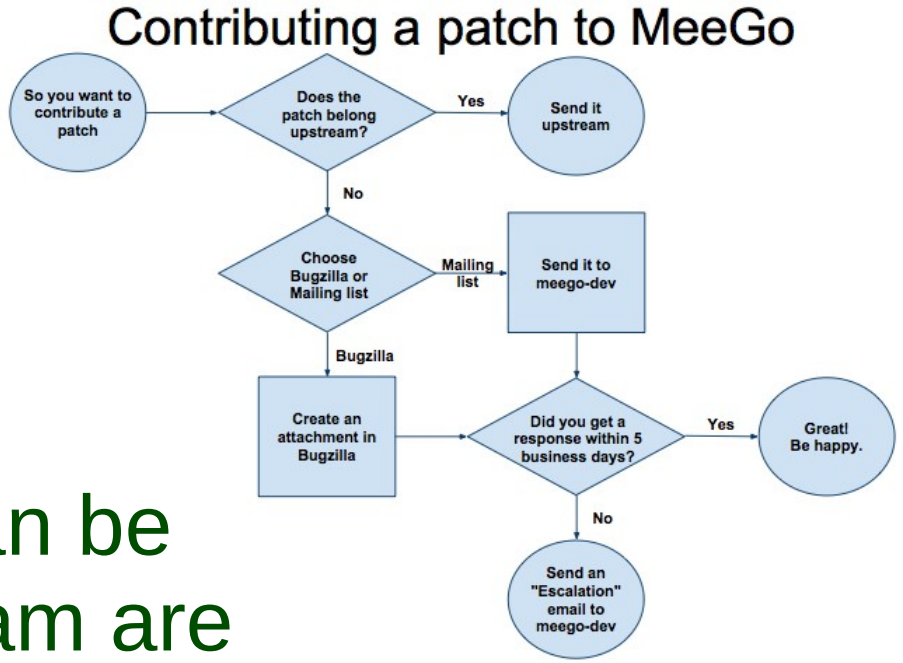
- Users
  - The same application across multiple verticals
  - Modern user experience
- OEMs
  - Time to market
  - Based on a stable, shared ground
- Developers
  - One target environment for multiple markets
  - Great development tools

Finding MeeGo

- Maemo is a predecessor to MeeGo
  - Used on N7x0, N8x0 and N900 devices
  - Supported by Qt
- MeeGo for netbooks
  - Runs on Intel-based netbooks
- MeeGo for handsets
  - Runs on N900 and Aava handsets

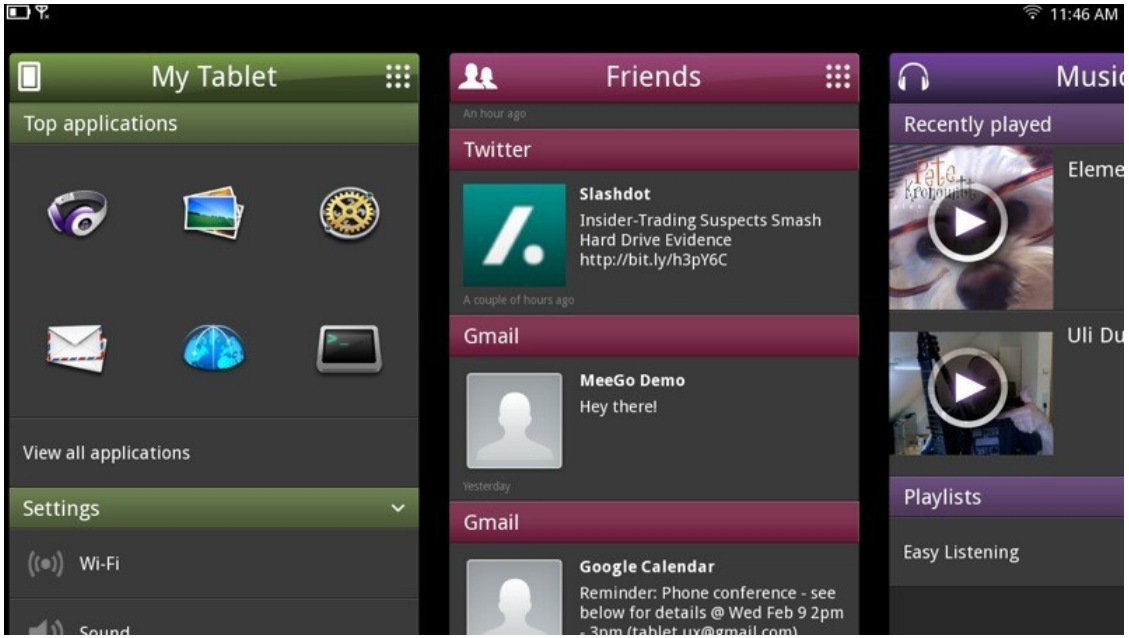
How MeeGo Works

- MeeGo wants to integrate well with the open source community
- All patches that can be submitted up-stream are required to be submitted there
- In an ideal world, MeeGo would not depend on any patches



Qt and MeeGo

- Qt is the preferred way of building user interfaces across all MeeGo UXes
- Qt Quick is used extensively for tablets



- The MeeGo SDK builds on Qt Creator

Break

Example Hardware

- Nokia N9
  - Arm-based
  - Multi-point touch
  - Slide-out keyboard

Example Hardware

- Intel based Netbook
  - Intel Atom or Intel Core 2 CPU *(requires support for SSSE3)*
  - Not GMA-500, Nvidia, or ATI

Example Hardware

- Aava Handset
  - Intel Atom CPU
  - Multi-touch

Example Hardware

- BeagleBoard
  - TI OMAP3

- PandaBoard
  - TI OMAP4

- Arm-based evaluation boards with multimedia focus

Developing for MeeGo

- The MeeGo SDK contains most components
  - Uses MADDE for building

    (MeeGo Application Development and Debug Environment)
  - QtSimulator for quick development
  - Remote debugger for hardware target development
- Virtual machine for target development
  - VirtualBox

Downloading MeeGo SDK

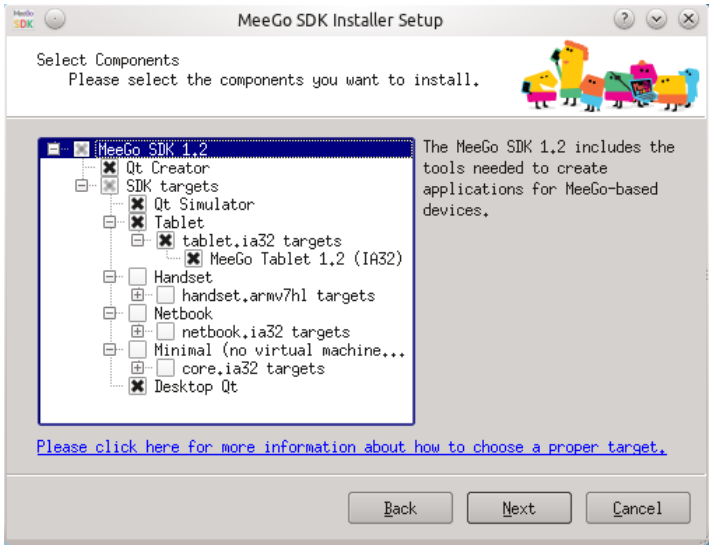
- From http://meego.com/downloads
- Follow the links through the page until you locate th installer binary
- Download the binary

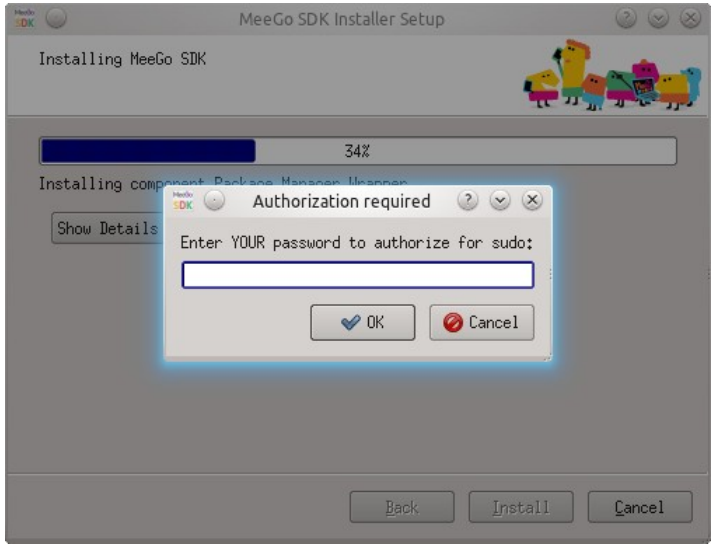
Running the Installer

- The installer is 26MB, but downloads GBs of data during Install
  - Make sure to have a good Internet connection while installing
- The installer must be made executable before running, e.g.
  - chmod +x meego-sdk-installer-online-*
  - ./meego-sdk-installer-online-*

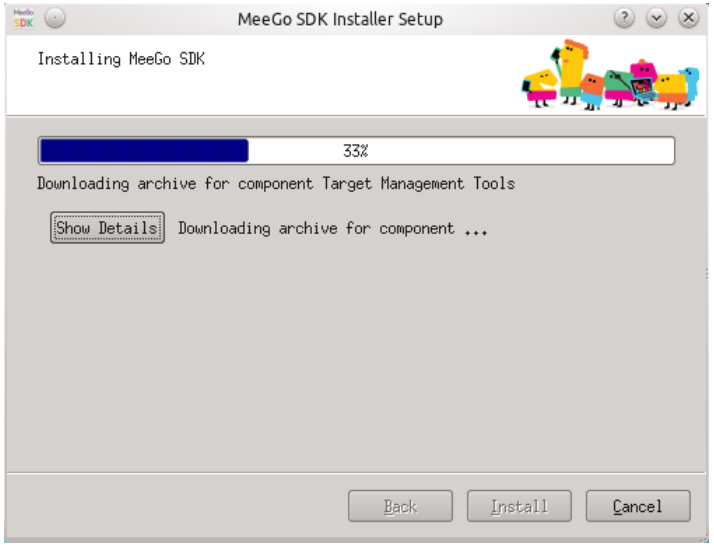
Running the Installer

- Simulator and Tablet





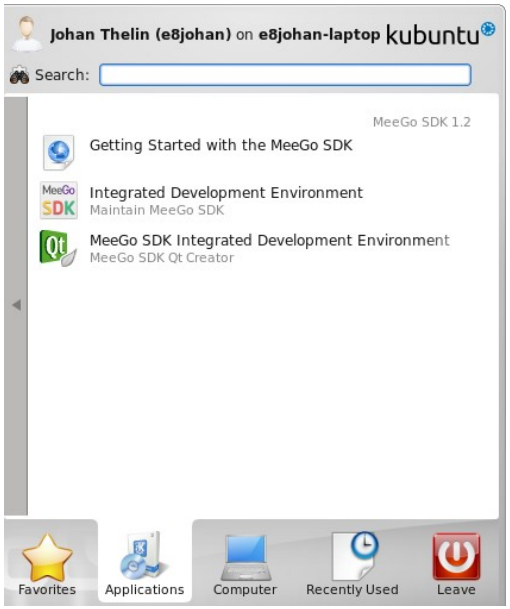
- Enter sudo credentials



- Enjoy the installation

What was Installed?

- Development Environment
- MADDE targets



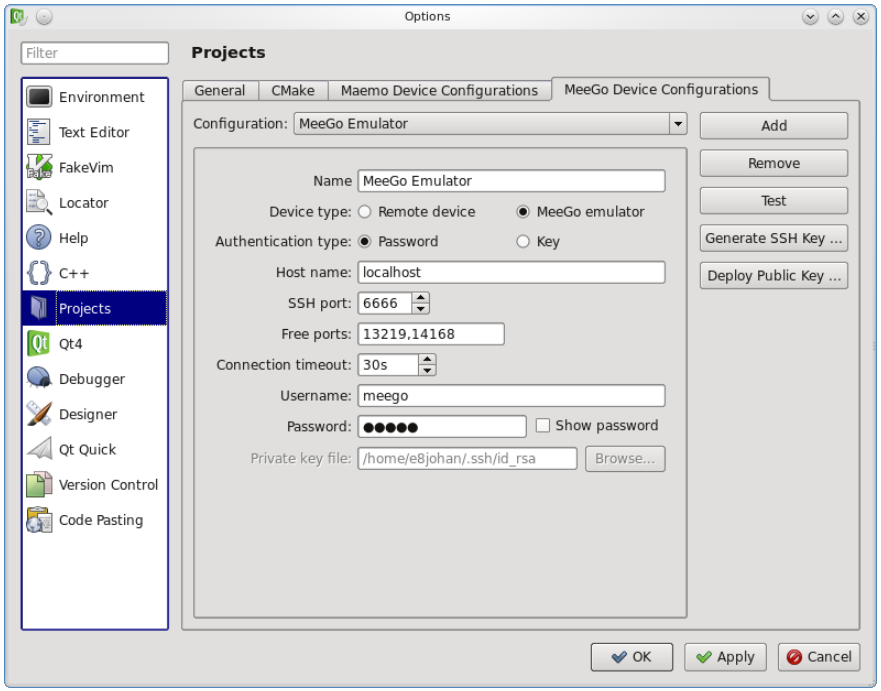
- `$ mad-admin list`

```
Targets:
meego-core-ia32-1.1                     (installable)
meego-core-ia32-1.1.2                   (installable)
meego-core-ia32-trunk                   (installable)
meego-handset-ia32-1.1.2                (installable)
meego-handset-ia32-1.1.2-4.7.1          (installable)
meego-handset-ia32-1.1                  (installable)
meego-handset-ia32-w32-1.1              (installable)
meego-handset-ia32-trunk                (installable)
meego-netbook-ia32-1.1.2                (installable)
meego-netbook-ia32-1.1.2-4.7.1          (installable)
meego-netbook-ia32-1.1                  (installable)
meego-netbook-ia32-w32-1.1              (installable)
meego-netbook-ia32-trunk                (installable)
meego-tablet-ia32-1.2.0.90.0.20110517.1 (installed)

Runtimes:
meego-handset-ia32-qemu-1.1.20110110.1026-runtime          (installable)
meego-handset-ia32-qemu-1.1.20110118.1010-runtime          (installable)
meego-handset-ia32-qemu-1.1.20101031.2201-sda-runtime      (installable)
meego-handset-ia32-w32-qemu-1.1.20101031.2201-sda-runtime  (installable)
meego-handset-ia32-qemu-1.1.99.0.20110329.5-runtime        (installable)
meego-netbook-ia32-qemu-1.1.20110110.1049-runtime          (installable)
meego-netbook-ia32-qemu-1.1.20110118.1020-runtime          (installable)
meego-netbook-ia32-qemu-1.1.20101031.2037-sda-runtime      (installable)
meego-netbook-ia32-w32-qemu-1.1.20101031.2037-sda-runtime  (installable)
meego-netbook-ia32-qemu-1.1.99.0.20110329.5-runtime        (installable)
meego-tablet-ia32-qemu-1.2.0.90.0.20110517.1-runtime       (installed)
```

Targets

- MADDE works with targets
  - A pre-configured build environment consisting of tool-chain and libraries
    - meego-core-armv7l
    - meego-netbook-ia32
    - etc
- Targets are added as Qt Versions in Qt Creator
- Simply pick the right target when building



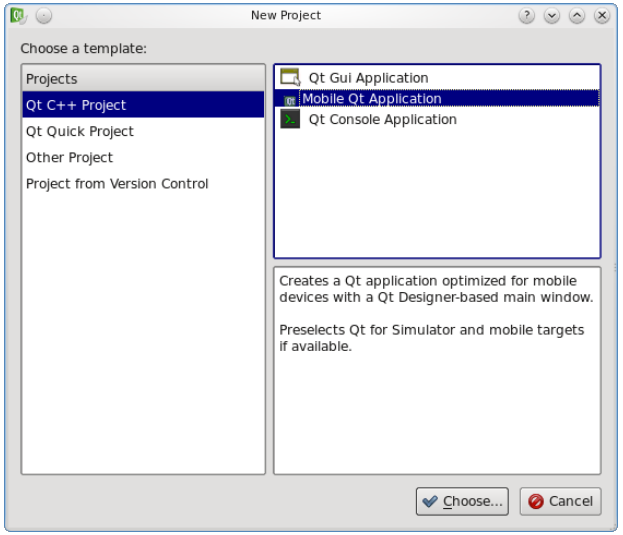
Typical Development

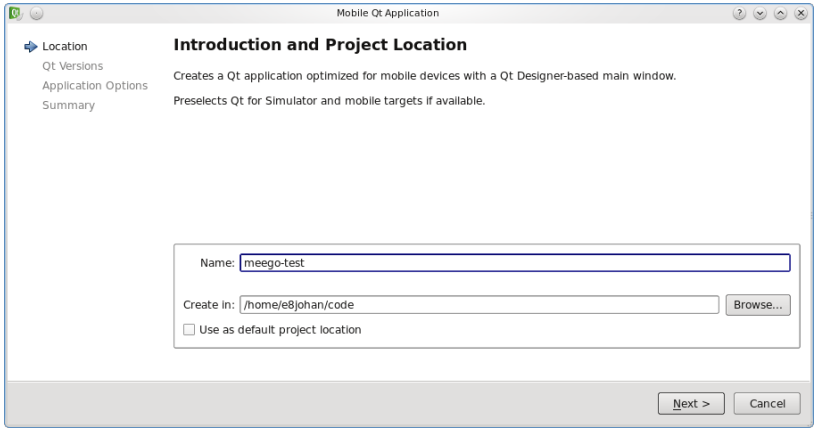
- Work-flow of a project
  - Setup a project
  - Develop using simulator
  - Switch to target and build an RPM
  - Deploy to virtual target or remote target
  - Perform final testing and polish

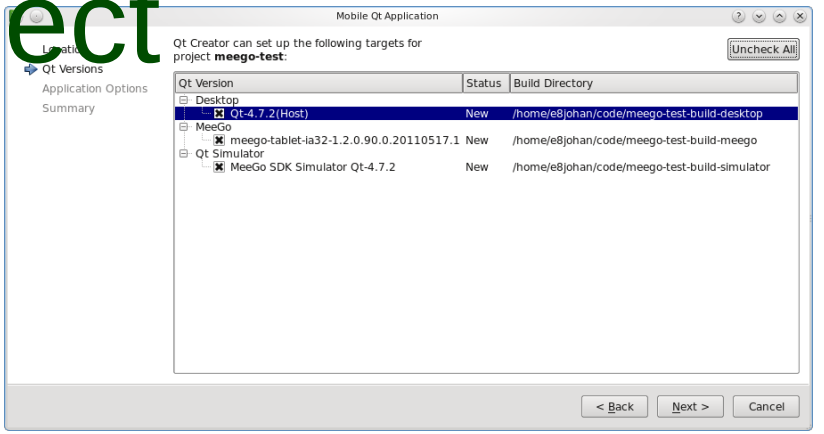
Qt Creator

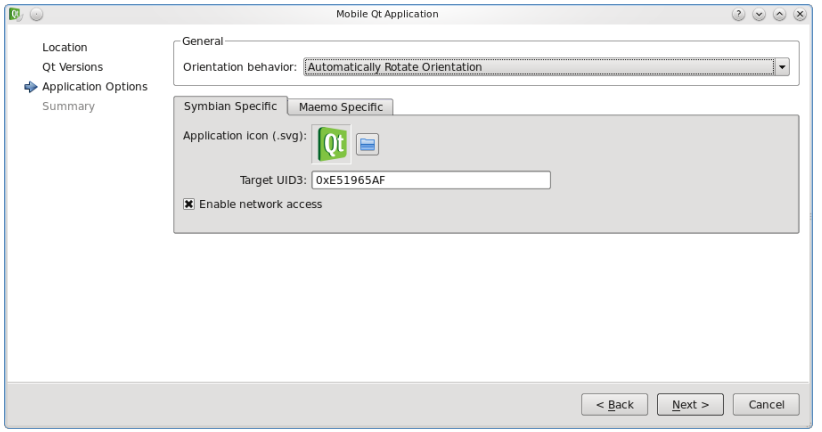

Qt Creator
File Edit Build Debug Tools Window Help
Welcome
Edit
Design
Debug
Projects
Help
Qt Creator
Getting Started
Develop
News & Support
Tutorials
The Qt Creator User Interface
Building and Running an Example
Creating a Qt C++ Application
Creating a Mobile Application
Creating a Qt Quick Application
Did You Know?
In the editor, F2 follows symbol definition, Shift+F2 toggles declaration and definition while F4 toggles header file and source file.
Featured
Tutorial
How to create a Qt Quick application
This tutorial describes how to use the Qt Creator to implement the states and transitions example application.
Take Tutorial
Examples
Explore Qt C++ examples:
Choose an Example...
Explore Qt Quick examples:
Choose an Example...
Open Project...
Create Project...
Feedback
Help us make Qt Creator even better
Type to locate
1 Build Issues
2 Search Results
3 Application Output
4 Compile Output

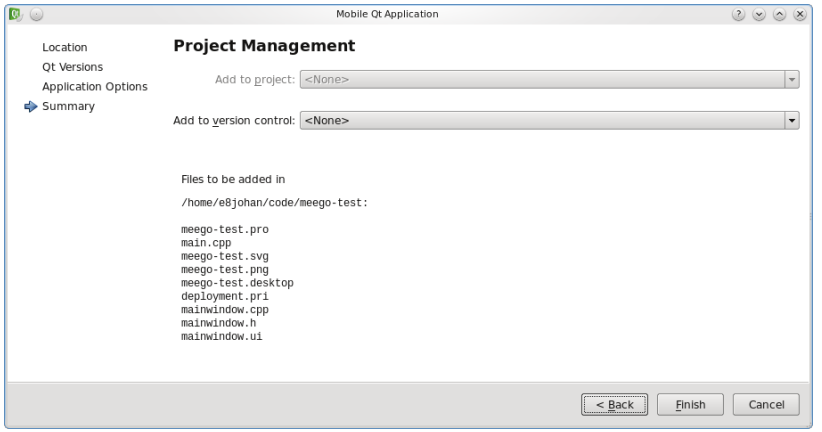
Create a Project











Targets

- Switching targets is just a matter of selecting it from from a menu