



Лаборатория системного программирования и  
информационных технологий СПбГУ

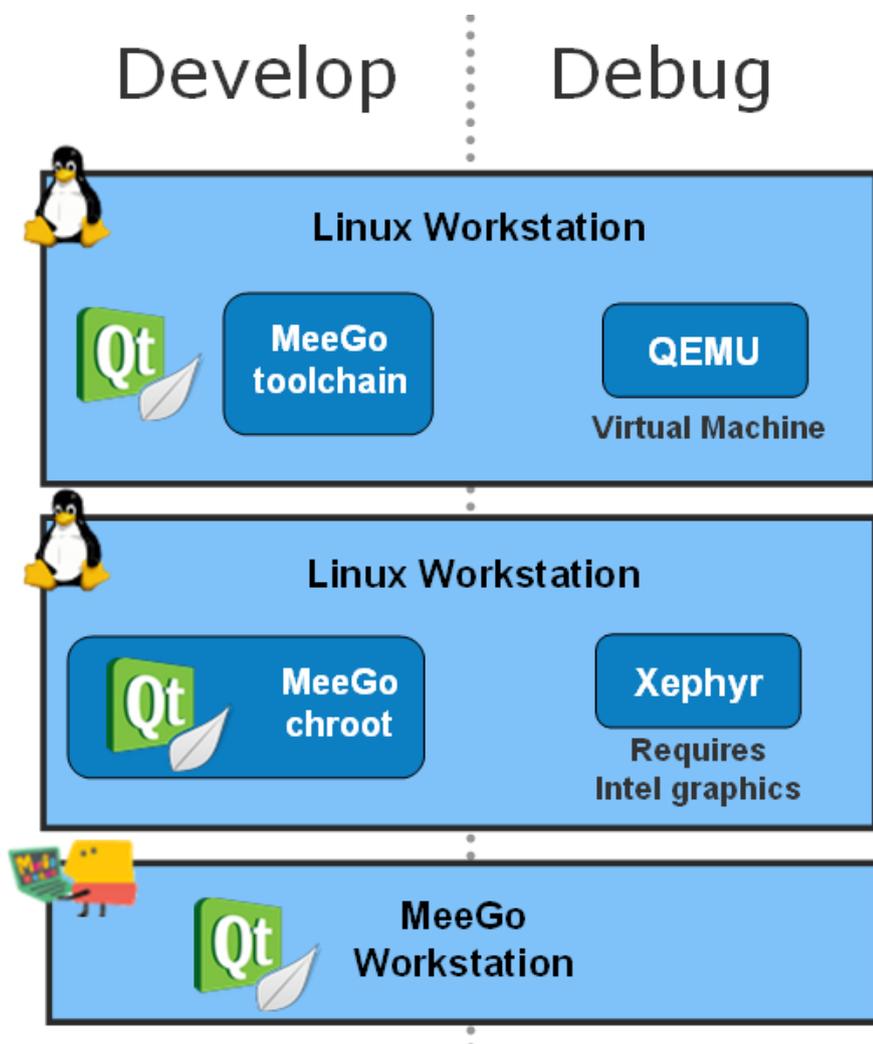
---

# MeeGo SDK. Обзор технологии Qt

*Сергей Леви*

- ▶ MeeGo SDK
  - ▶ технические требования
  - ▶ варианты развёртки
- ▶ Технология Qt
  - ▶ общий обзор и история развития
  - ▶ обзор библиотек
  - ▶ инструменты разработки
  - ▶ система сборки qmake
  - ▶ механизм сигналов и слотов

# MeeGo SDK



- 1) Разработка ведётся сразу на целевом устройстве под MeeGo
- 2) Виртуализация: MeeGo выполняется на эмуляторе QEMU
- 3) chroot: MeeGo выполняется в нативной среде, но в изолированной песочнице
- 4) Разработка на любой машине под Linux. Удалённая отладка на целевом устройстве

## Разработка в MeeGo

- ▶ Native-установка MeeGo на нетбук
- ▶ В MeeGo могут быть установлены все необходимые инструменты для разработки
- ▶ Нет сложностей с эмуляцией, виртуализацией, chroot – всё работает “из коробки”
- ▶ Пользовательский интерфейс MeeGo не ориентирован на разработчика
- ▶ Пользовательский интерфейс смартфонов *совсем* не ориентирован на разработчика

- ▶ QEMU – эмулятор процессора (можно запустить как ia32, так и arm-версию)
- ▶ Используется особая сборка QEMU с поддержкой GL для отрисовки GUI
- ▶ Для интеграции используется MADDE (*MeeGo Application Development and Debugging Environment*)
- ▶ **Обязательные требования:**
  - ▶ процессор с поддержкой виртуализации
  - ▶ графический ускоритель



## МееGo в chroot-среде

- ▶ chroot – замена корня файловой системы для конкретного процесса (процесс оказывается в песочнице)
- ▶ выполняется на той же машине, на которой ведётся разработка
- ▶ инструменты разработки также запускаются под chroot
- ▶ Для работы графического UI МееGo в окно на хостовой системе используется X-сервер Xephyr
- ▶ На данный момент требует GPU от Intel



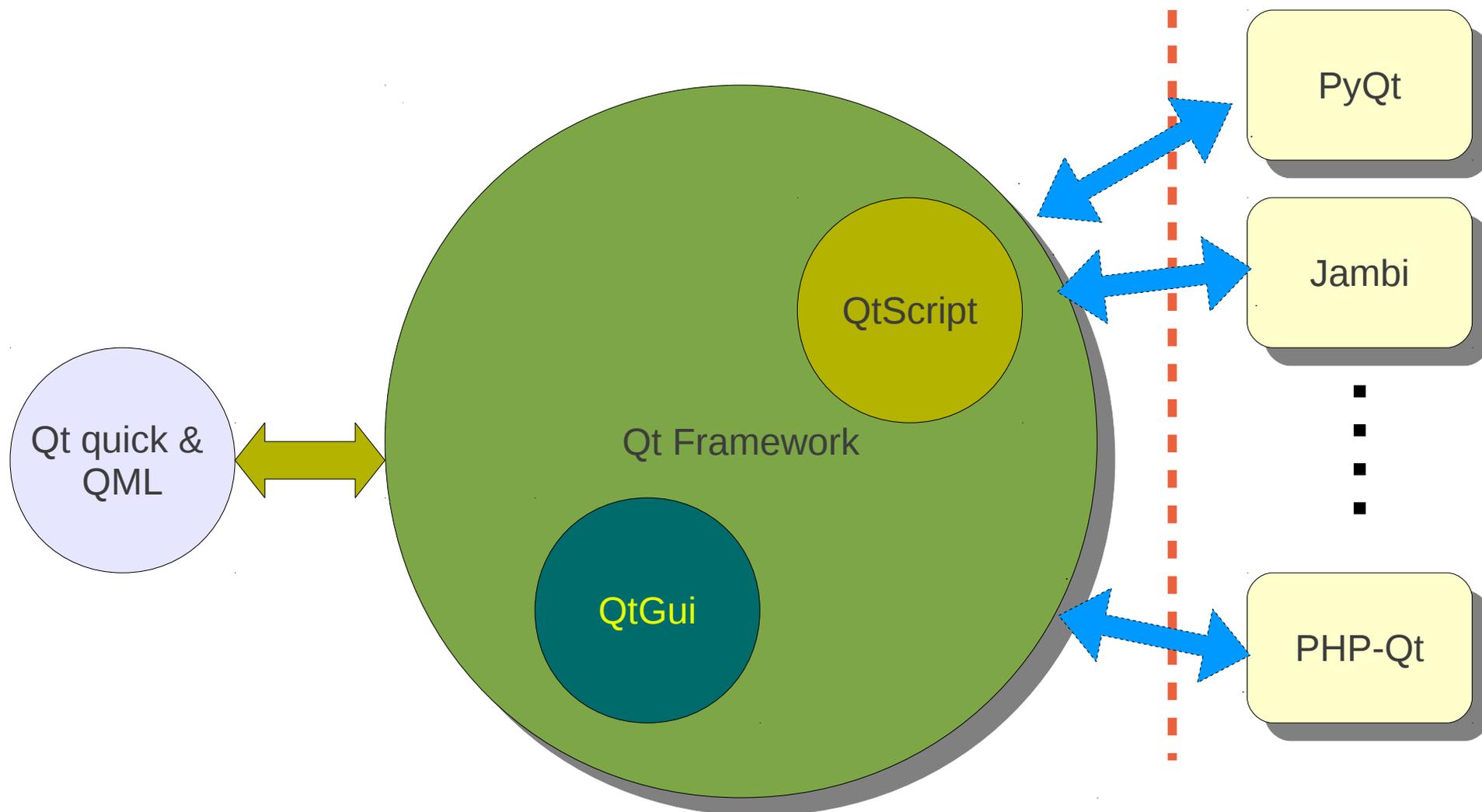
# Кросс-компиляция и удалённая отладка на целевом устройстве

- ▶ Разработка ведётся на любой машине под Linux.
- ▶ Кросс-компиляция при помощи GCC
- ▶ Удалённая отладка на целевом устройстве при помощи gdb
- ▶ Отладка “по живому”
- ▶ handset-вариант MeeGo на сегодняшний день запускается только на Nokia N900 и на Aava

# Технология Qt



# Что такое Qt?





# Мотивация использования фреймворка

- ▶ Обёртка с простым интерфейсом для порой очень сложных API
- ▶ Единый интерфейс для разных платформ
  - ▶ Linux / Unix (включая MeeGo)
  - ▶ Mac OS X
  - ▶ Windows
  - ▶ Symbian
  - ▶ Windows CE / Mobile
  - ▶ ...
- ▶ Отказ от API ОС в пользу API Qt обеспечивает переносимость кода.
- ▶ Приложение использует мощност native-интерфейсов целевой платформы.

## История развития

- ▶ Разработка начата в 1991 Haavard Nord и Eirik Chambe-Eng, основавшими в последствии компанию *Quasar Technologies*, затем переименованную в *Trolltech*
- ▶ Первый релиз – 1995 год: библиотека графических компонент для X11/UNIX и Windows под двойной лицензией для UNIX-версии и проприетарной для Windows
- ▶ 2005 – в Qt 4 действие GPL распространено на версии для всех платформ
- ▶ 17 июня 2008 – *Trolltech* приобретена компанией *Nokia*
- ▶ 2009 – для версии 4.5 добавлена лицензия LGPL



## Ключевые библиотеки Qt

- ▶ QtCore – базовые примитивы, не имеющие отношения к GUI
- ▶ QtGui – примитивы GUI
- ▶ Phonon и QtMultimedia – библиотеки для работы с мультимедиа
- ▶ QtNetwork – работа с сетью
- ▶ QtOpenGL – поддержка OpenGL-графики
- ▶ QtXml – работа с XML
- ▶ QtSql – работа с SQL-базами данных
- ▶ QtScript – позволяет использовать скриптовый язык, аналогичный JavaScript в Qt-приложениях
- ▶ QtWebKit – позволяет работать с веб-движком WebKit

- ▶ Дополнительная библиотека (не часть дистрибутива)
- ▶ MeeGo 1.1 имеет поддержку Qt Mobility 1.0.2
- ▶ Предоставляет интерфейс для функциональностей, специфичных для мобильных устройств:
  - ▶ сервисы (GSM-связь, Bluetooth)
  - ▶ записная книжка
  - ▶ мгновенные сообщения
  - ▶ органайзер
  - ▶ устройства позиционирования
  - ▶ сенсоры (акселерометр, датчик освещённости)

# Ключевые инструменты разработки на Qt

- ▶ QtCreator – IDE для разработки на C++ и QML (язык построения интерфейсов на Qt)
- ▶ QtDesigner – наглядное создание графических интерфейсов
- ▶ QtLinguist – локализация интерфейса
- ▶ QtAssistant – система справки
- ▶ Qt Simulator – эмулятор мобильных устройств
- ▶ qmake – система сборки
- ▶ moc – метаобъектный компилятор
- ▶ uic – компилятор графических интерфейсов
- ▶ rcc – компилятор ресурсов



## Система сборки qmake

- ▶ Сборка Qt-приложений осуществляется make (а вы что думали?)
- ▶ qmake генерирует Makefile на основе файла проекта .pro (язык файлов проекта проще языка Makefile)
- ▶ Возможна автоматическая генерация файла проекта
  - ▶ `qmake -project`
- ▶ Процесс сборки контролируется установкой переменных qmake (выбор компилятора, директории сборки и т.д.)
- ▶ В проектных файлах можно использовать условные блоки и некоторые простые функции

## qmake - пример

---

```
CONFIG += qt debug
QT += core gui network
HEADERS += hello.h
SOURCES += hello.cpp main.cpp
LIBS += -lncurses

win32 {
    SOURCES += helloworld.cpp
}

unix {
    SOURCES += helloworld.cpp
}

!exists( main.cpp ) {
    error( "No main.cpp file found" )
}
```

## Signals and Slots

- ▶ Способ обмена сообщениями между объектами
- ▶ Каждый класс может объявлять сигналы, которые он может отправлять и слоты, которые можно ассоциировать с конкретными сигналами.
- ▶ Сигналы и слоты могут иметь аргументы
- ▶ Реализован как надстройка над синтаксисом C++ (исходный файл обрабатывается метакомпилятором moc, который генерирует вспомогательные файлы)
- ▶ Несовершенство метакомпилятора накладывает ограничения на допустимые конструкции C++

# Signals and Slots – пример

```
#include <QObject>

class Counter : public QObject
{
    Q_OBJECT

public:
    Counter() { m_value = 0; }

    int value() const {
        return m_value;
    }

public slots:
    void setValue(int value);

signals:
    void valueChanged(int newValue);

private:
    int m_value;
};
```

```
void Counter::setValue(int value)
{
    if (value != m_value) {
        m_value = value;
        emit valueChanged(value);
    }
}

...
Counter a, b;
QObject::connect(&a,
    SIGNAL(valueChanged(int)),
                &b, SLOT(setValue(int)));

a.setValue(12);    // a.value() == 12,
                  // b.value() == 12
b.setValue(48);   // a.value() == 12,
                  // b.value() == 48
```



## Для дополнительного чтения

---

### 1) MeeGo wiki

- ▶ [http://wiki.meego.com/MeeGo\\_SDK\\_Development\\_Options](http://wiki.meego.com/MeeGo_SDK_Development_Options)
- ▶ [http://wiki.meego.com/Getting\\_started\\_with\\_the\\_MeeGo\\_SDK\\_for\\_Linux](http://wiki.meego.com/Getting_started_with_the_MeeGo_SDK_for_Linux)

### 2) Qt Documentation

(<http://doc.qt.nokia.com/4.7/index.html>)

# Вопросы?

*sergeyle@gmail.com*



**Code less.  
Create more.  
Deploy everywhere.**