Timesys University

Track Two

Building an Internet Radio with the TI Sitara AM3517 using LinuxLink

Session 2

How to build a modern User Interface to launch Internet Radio playback using Qt Embedded for Linux

Audio streaming is available for this event.
Turn on your speakers to listen.

Tools You Can Use

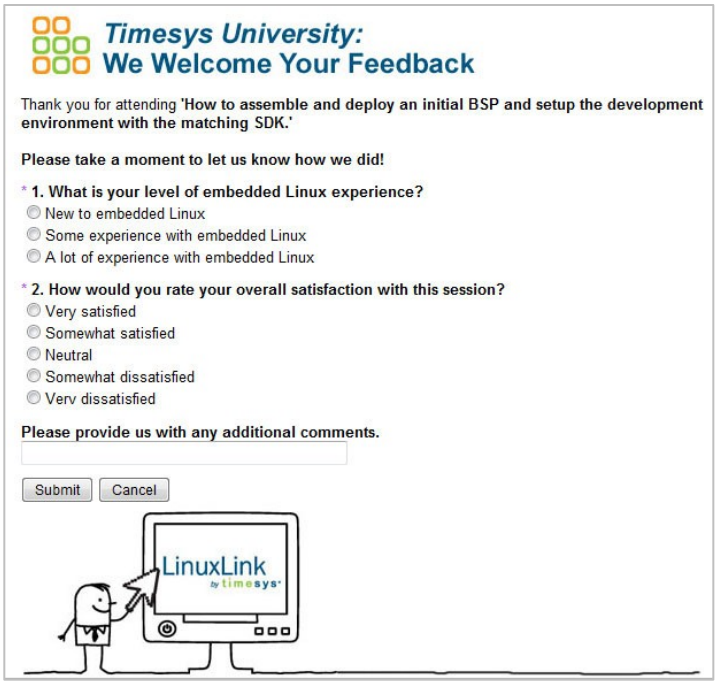
Q&A

- Click on Q&A panel (?) or chat icon in the bottom right corner
- Type in your question in the space provided
- Click on “Submit”

Tools You Can Use

Polling

- The poll will appear on your screen
- Select your answer for each question
- Click on “Submit”



Session Information

- **You can download the slides for today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/ti_am3517
- **You can view a recording of today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/ti_am3517
- **Today's speaker:**

Maciej Halasz
Director, Product Management
Timesys

Building an Internet Radio with the TI Sitara AM3517

- **Session 1 – Recording Available**

  How to assemble and deploy an initial BSP and setup development environment with the matching SDK

  http://www.timesys.com/embedded-linux/training/timesys-university/ti_am3517

- **Session 2 – Today**

  How to build a modern User Interface to launch Internet Radio playback using Qt Embedded for Linux

- **Session 3 – June 13 @ 1pm EDT**

  How to decode a media stream and integrate Bluetooth functionality for a remote speaker

- **Session 4 – June 27 @ 1pm EDT**

  How to optimize, test and integrate the solution for fast boot and quick deployment

Today's Agenda

- **Recap of what we have done so far**
- **Modify the underlying Linux image to support Qt and audio streaming**
- **Qt development tools installation/setup**
- **Internet Radio – GUI development**
  - Create a QMainWindow application
  - Use a number of Qt widgets including
    - QTabWidget
    - QPushButton
    - QWebKit
    - Layouts and more
  - Add code for managing internet radio streams
  - Add code for launching media player
  - Test your application locally
  - Cross-compile for the AM3517 target
- **Summary**

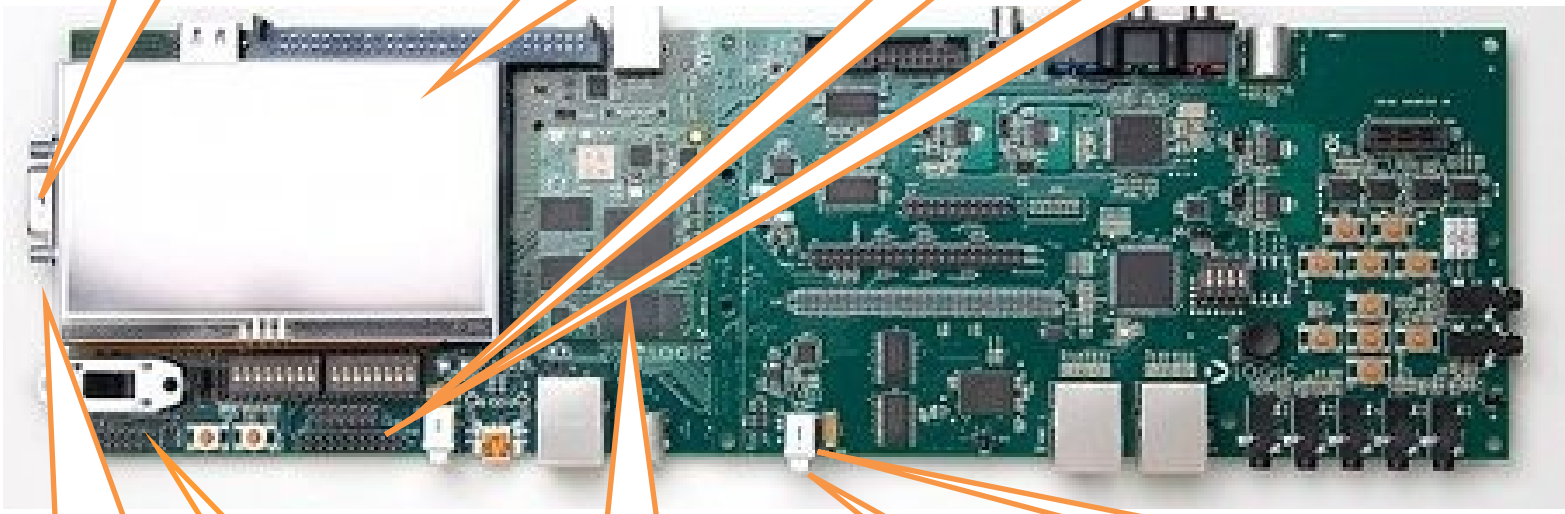
Session 1 Recap

What We Have Accomplished So Far

- **Learned about TI AM3517 LinuxLink – needed for all exercises**
- **Reflected product requirements in Linux BSP and SDK**
- **Built a custom BSP with LinuxLink Web Edition**
  - Experiment on day one with a pre-built starting point
- **Setup a development environment**
  - System level development and optimizations
  - Development of a value-add software (applications)
- **Deployed the system on the target via NFS for future development**
  - Transferred images
  - Configured bootloader

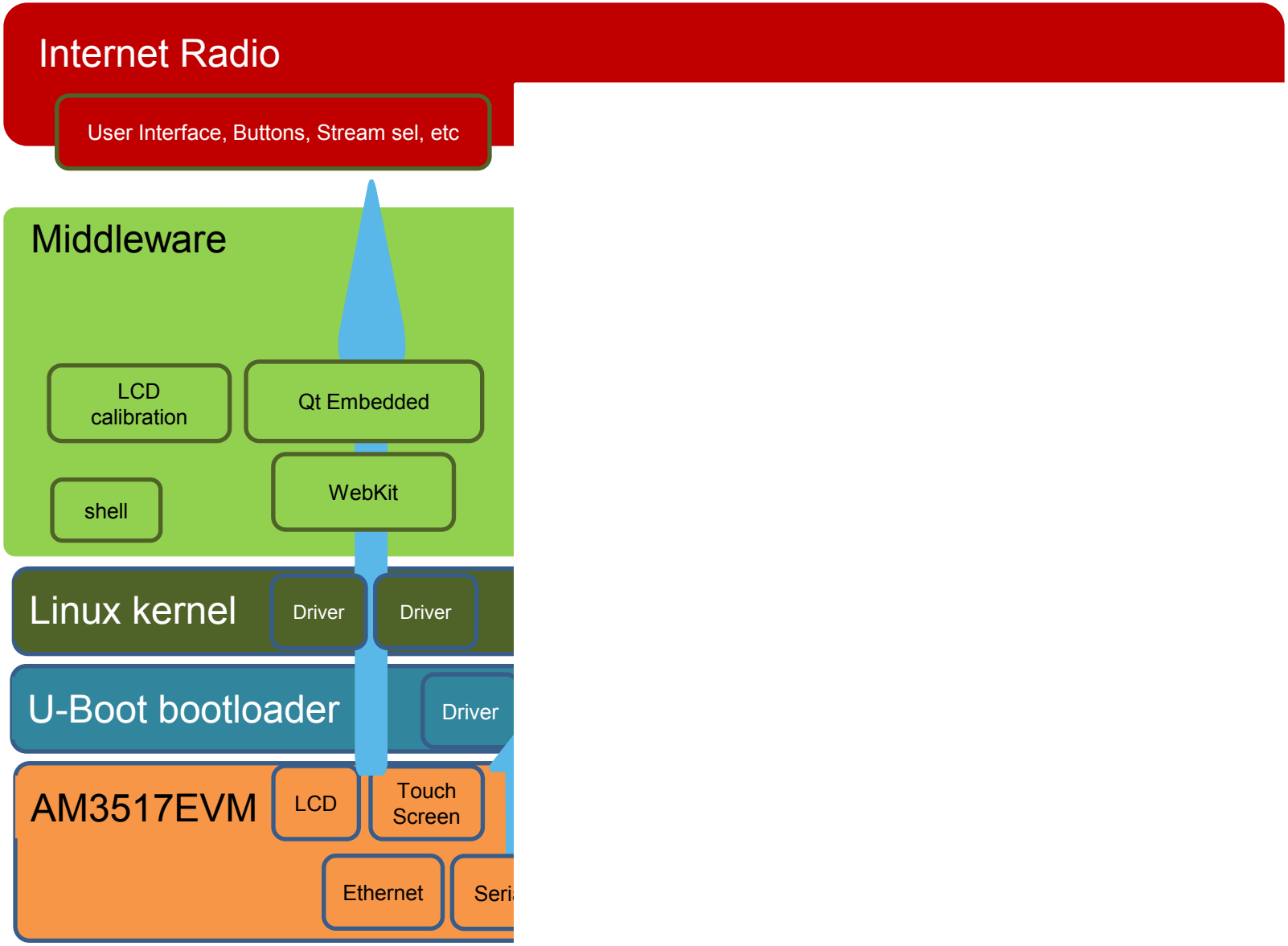
Project Requirements (TI AM3517EVM)

- **Serial port communication**
- **Graphics**
- **Touchscreen**
- **Applications**
  - Screen calibration
  - Playback control UI
- **USB**
  - Storage (USB stick)
  - Extensions
- **Ethernet**
  - Secure Connection
  - Transfer (FTP/SCP)
  - Console (Telnet/SSH)



- **Audio**
  - Alsa Mixer
  - Sound playback
- **SD/MMC Card**
  - Filesystem
  - Booting Linux
- **NAND Flash**
  - Boot from
  - Additional storage
- **WiFi (optional)**
  - Streaming audio
- **Bluetooth**
  - Sensor connections

Internet Radio (Blueprint)



Last Week's Homework – Share Your Experience

1. **Outline your own requirements**
2. **Assemble a BSP that matches your needs**
3. **Setup your development environment**
4. **Run your custom Linux images on your hardware**

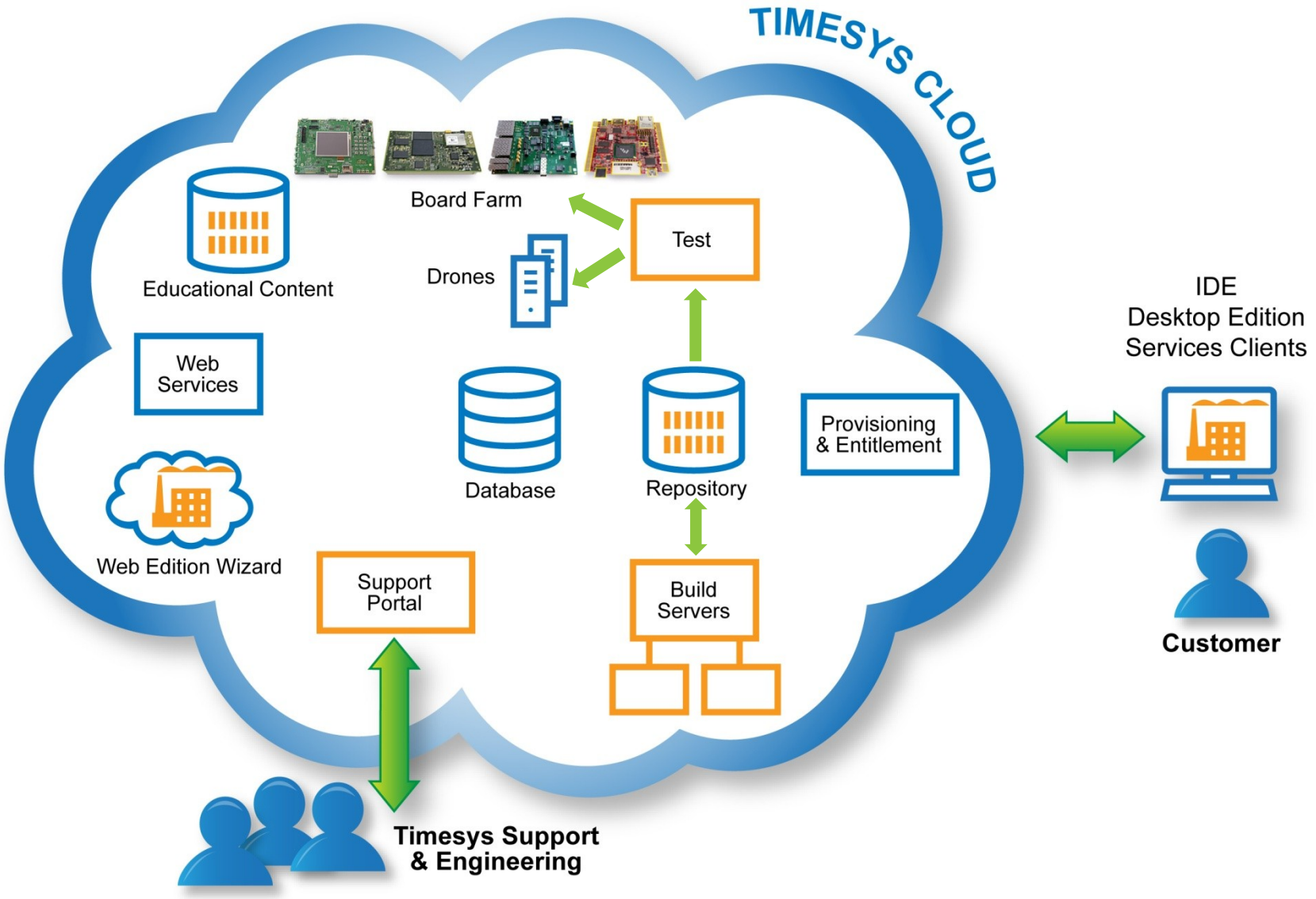
- Did you have any problems outlining your requirements and then translating them into a Linux blueprint?
- How did the assembly of your custom BSP go?
- Were you successful in getting your custom Linux images up and running on your boards?

Giveaway!!!

- **If you attend at least 3 out of 4 sessions in this Timesys University track, we will automatically enter you into a drawing for a chance to win a Logic PD Zoom AM3517 EVM Development Kit**

Adding APIs/Libraries Using the LinuxLink Desktop Interface

LinuxLink Architecture



Requirements Helper

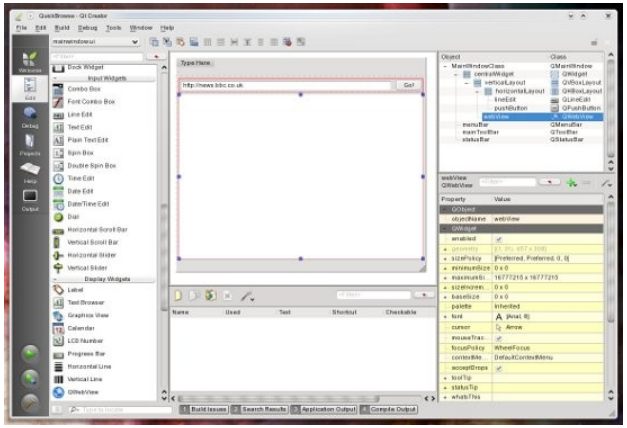
Feature	What I need	Where to select
Linux kernel	Latest, 2.6.3x kernel	Linux kernel
Cross-Toolchain	GCC, Binutils, GDB	Toolchains
C Library	glibc	Toolchains
Touch screen	tslib	Packages
Qt	qt-embedded-linux	Packages
Bluetooth	bluez (will select dbus)	Packages
Media playback	Mplayer/GStreamer	Packages
WiFi	wireless_tools wpa_supplicant chipset dependent	Packages
Sound	alsa	Packages
Streaming	live555	Packages
System initialization	busybox	Packages

Qt Development Tools Installation/Setup

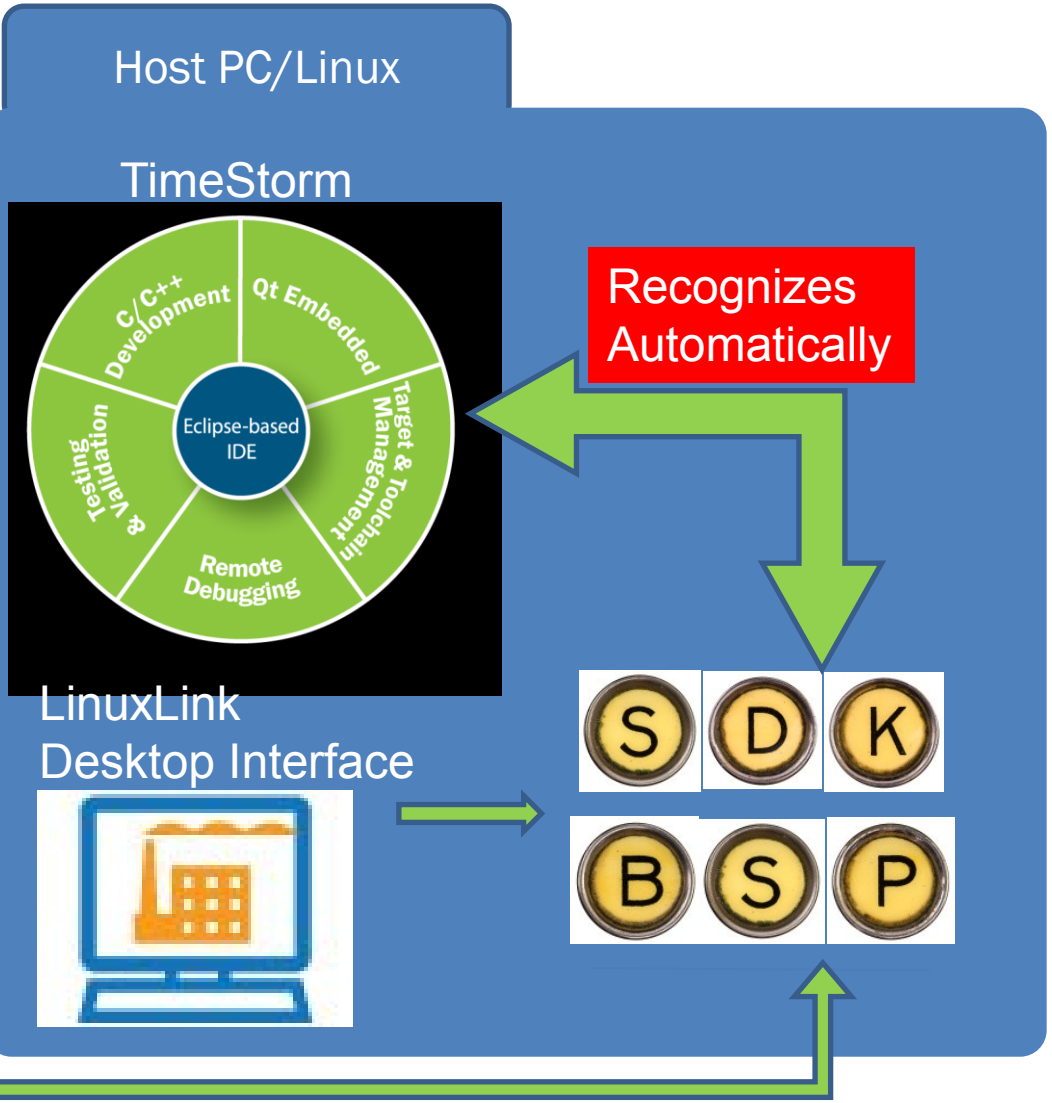
Qt Development Tools

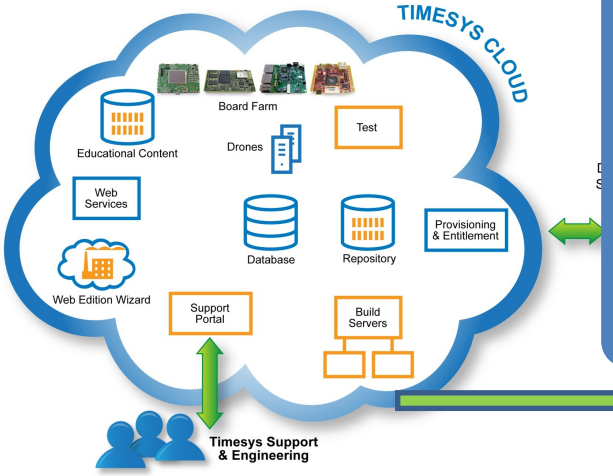
- **Options**
  - TimeStorm with Qt — linuxlink.timesys.com
  - QtCreator — qt.nokia.com
- **Installation**
  - Download from your LinuxLink account
  - Uncompress with

    ```
    tar xzvf timestorm.tar.gz
    ```
  - Generate a license file
- **Run**
  - Export your QMAKESPEC variable point it to linux-timesys-g++ in the SDK
  - Execute timestorm binary



IDE and cross-environment





timesys®

Internet Radio Application – GUI Development

Home Automation Application

- **Create a new Workspace**
- **Create a new Qt GUI Project**
  - UI Type - QMainWindow
  - Select additional Qt frameworks: XML and WebKit
- **Create the following components**
  - Remove menu bar and status bar
  - Use layouts
  - Add QTabWidget with 3 tabs:
    - Navigation
    - Saved
    - RSS feeds
  - Create buttons for music playback and stream management
  - Use Spacers
- **QDialog for adding a new radio station**
  - Connect windows together

Home Automation Application (Cont'd)

- **Define/Connect signals**
  - Capture the signal in a service class
  - Implement the service routine
  - Implement for all buttons
- **Change the look and feel of the design**
  - Create a resource file
  - Add a new JPGs to appropriate widgets

What We Have Accomplished

What We Have Accomplished

- **Downloaded desktop factory interface**
  - Added needed packages and rebuilt the runtime and SDK images
- **Installed the IDE development tools for Qt design**
- **Developed a GUI with the following:**
  - Layers, Several Widgets, Spacers and Labels
  - Implemented signals-handlers
- **Added code to our application to add/remove streams**
- **Cross-compiled for AM3517 target**

Next Session

- **June 13 @ 1pm EDT**

How to decode a media stream, and integrate Bluetooth functionality for a remote speaker

- Add libraries/packages to support media stream decoding
- Enable Bluetooth support in the Linux kernel
- Develop code to pair Bluetooth devices
- Configure A2DP HiFi Bluetooth profile for audio playback
- Add code in the application to access a stream and play it back via remote speaker
- More fun to come…

Homework

1. **Design your own GUI with Qt Embedded**
2. **Adapt/design your own UI with desired set of widgets**
3. **Create multiple windows. Connect buttons to handling routines via signals**

Let us know if you run into any challenges. We have setup a dedicated communication channel to share questions, comments and replies.

Please subscribe at https://lists.timesys.com/listinfo/timesys-university

While your questions will be answered by Timesys expert engineers, we encourage you to answer questions asked by others. Sharing experiences is always the best way to learn.

Glossary

LinuxLink (Web Edition) – Web-based version of LinuxLink

LinuxLink (Desktop Edition) – Local version with full customization and third-party tools integration

Workorder – Stores definition of your software – filenames, versions

Bootloader – Runs first, initializes necessary hardware, loads Linux

Linux kernel – Operating system that manages hardware access and other features for higher level software

Device Driver – Code that's part of a Linux kernel, defines how software accesses specific hardware

File System – All files (libraries/utilities/scripts/etc.) combined on a single storage, e.g. NAND flash

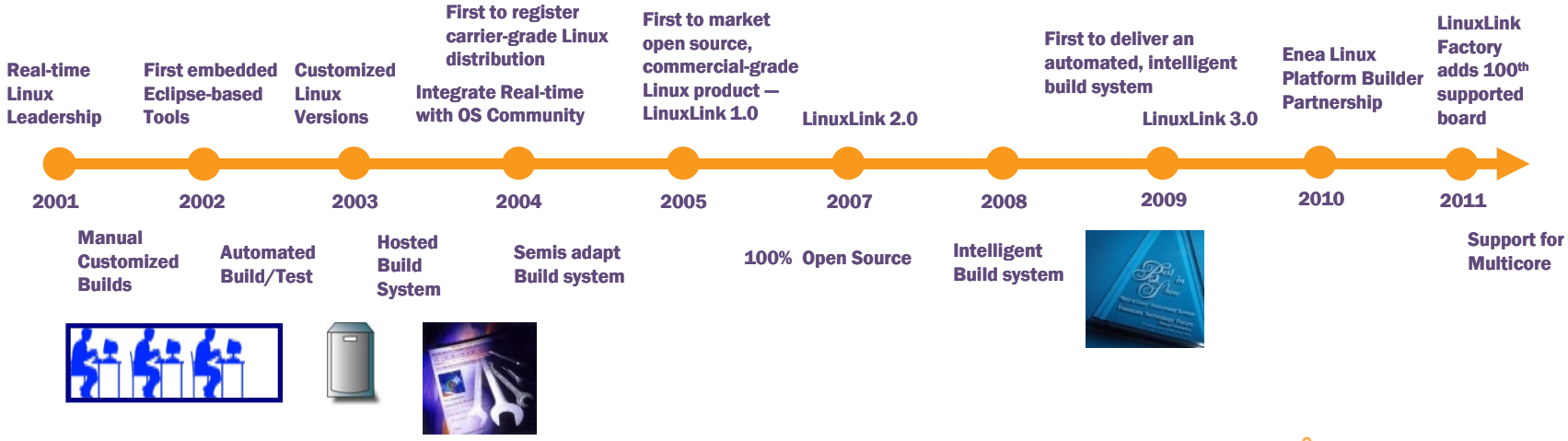
Middleware – Complete frameworks including APIs, utilities that provide specific functionality, e.g. Qt

API (library) – Used by applications, provide functionality, abstract hardware access

Toolchain (cross) – The most important part of the development environment. Used to compile source code into binaries.

About Timesys

- Carnegie Mellon University spin-off in 1996
- First real-time embedded Linux distribution
- First to register carrier grade Linux (CGL)
- First to market with an open source, commercial-grade embedded Linux development framework (LinuxLink)
- First to develop and deliver an award-winning, automated, intelligent, embedded Linux build system (LinuxLink 3.0)



More Info

- **You can download the slides for today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/ti_am3517

- **You can view a recording of today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/ti_am3517

Stay Online for Q&A!