FTF 2016
TECHNOLOGY FORUM

LS1012A, LS1024A, LS1043A CUSTOMER TRAINING

QorIQ LS1043A APPLICATION SOLUTION KIT (IoT, NAS, RGW)

RICHIE PEARN
Senior Principal Engineer
FTF-DES-N1853
May 19, 2016

PUBLIC USE

ABSTRACT

- This session provides a demonstration led by the instructor that showcases how to develop end products using the Application Solution Kit (ASK) for QorIQ LS1043A processors.

AGENDA

- ASK Description
- Designing with LS1012A, LS1024A, or LS1043A ASK
- Using the ASK to Make RDB Binaries
- Adding and Installing a Package
- Adapting the ASK for Your Hardware
- ASK Documentation

ASK DESCRIPTION

Software Products and Services

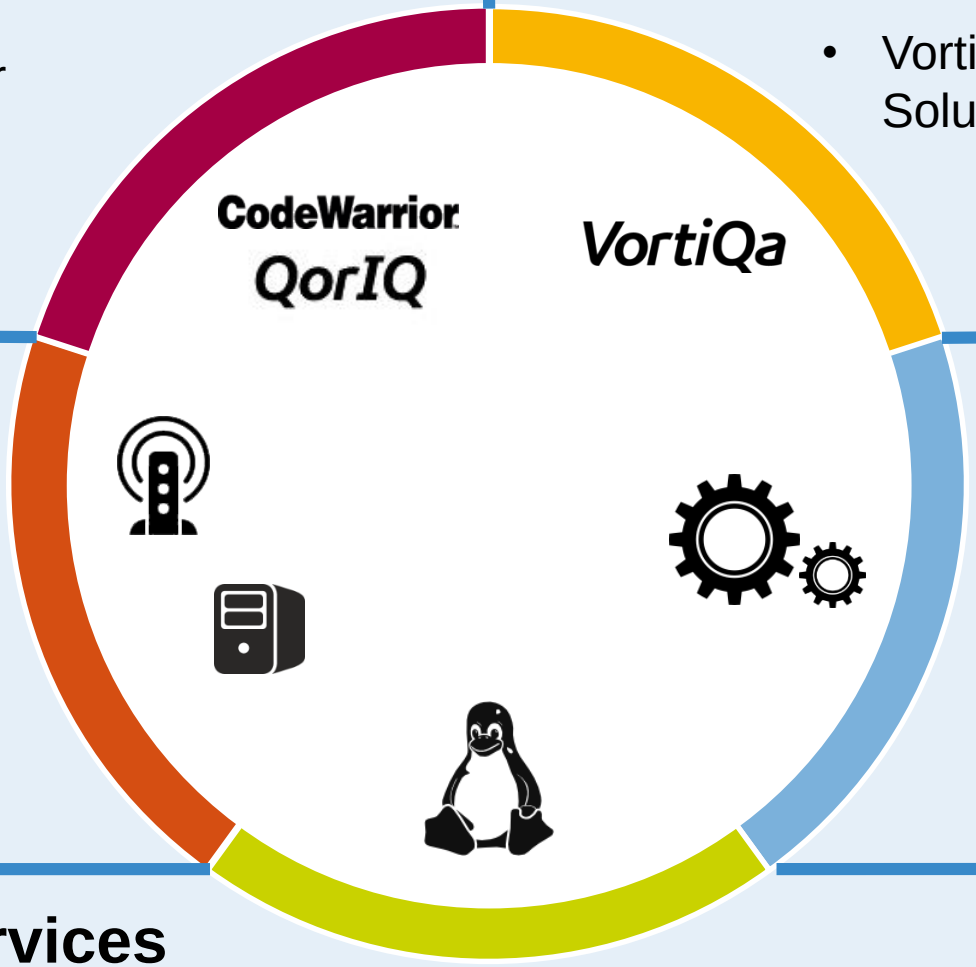
Visit us in the Tech Lab – #247

Development Tools

- CodeWarrior

Runtime Products

- VortiQa Software Solutions



Solutions Reference

- IOT Gateway
- OpenWRT+

Integration Services

- Security Consulting
- Hardened Linux

Linux® Services

- Commercial Support
- Performance Tuning

Accelerate **Customer Time-to-Market**

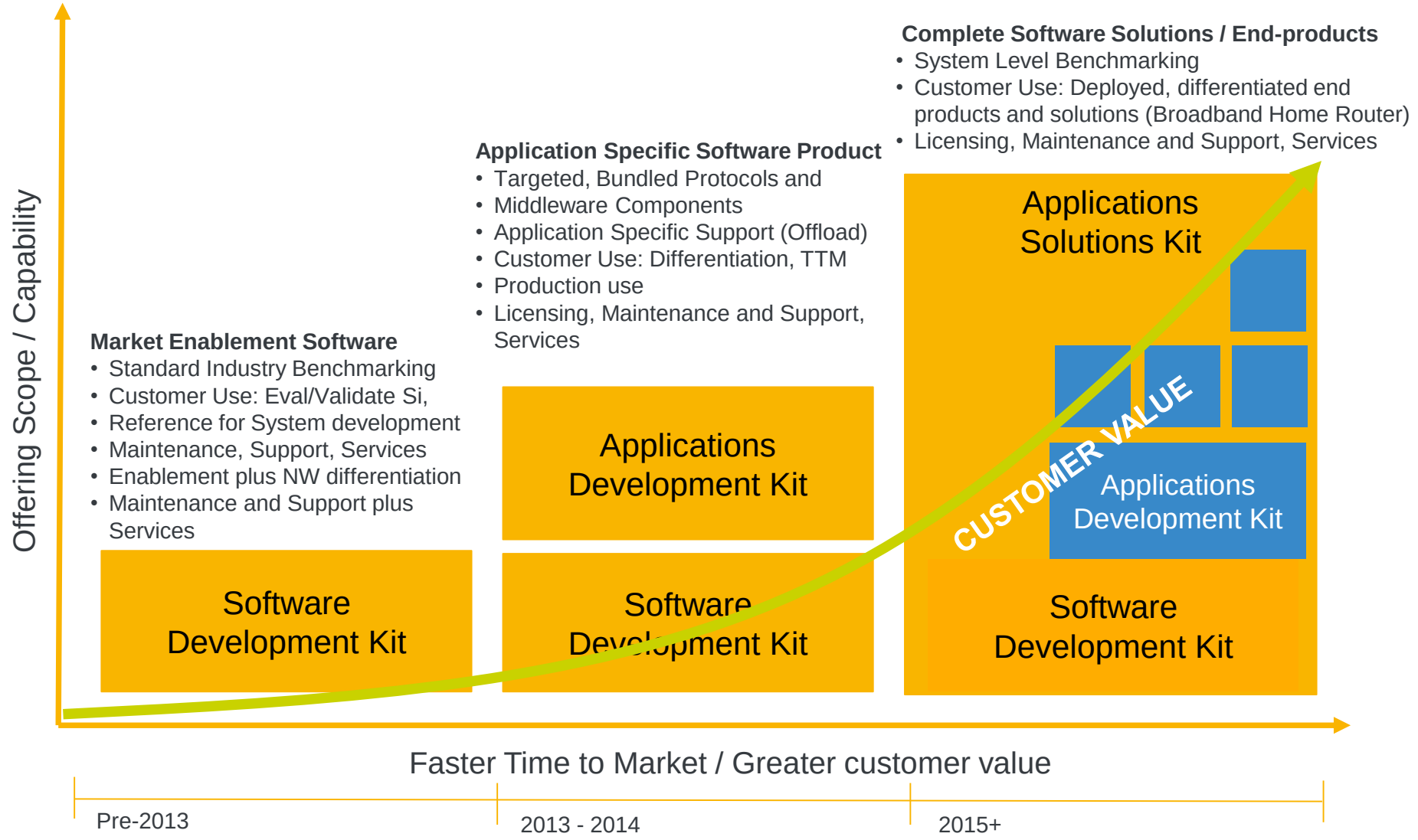
Deliver **Commercial Software, Support, Services and Solutions**

Simplify **Software Engagement with NXP**

Create **Success!**

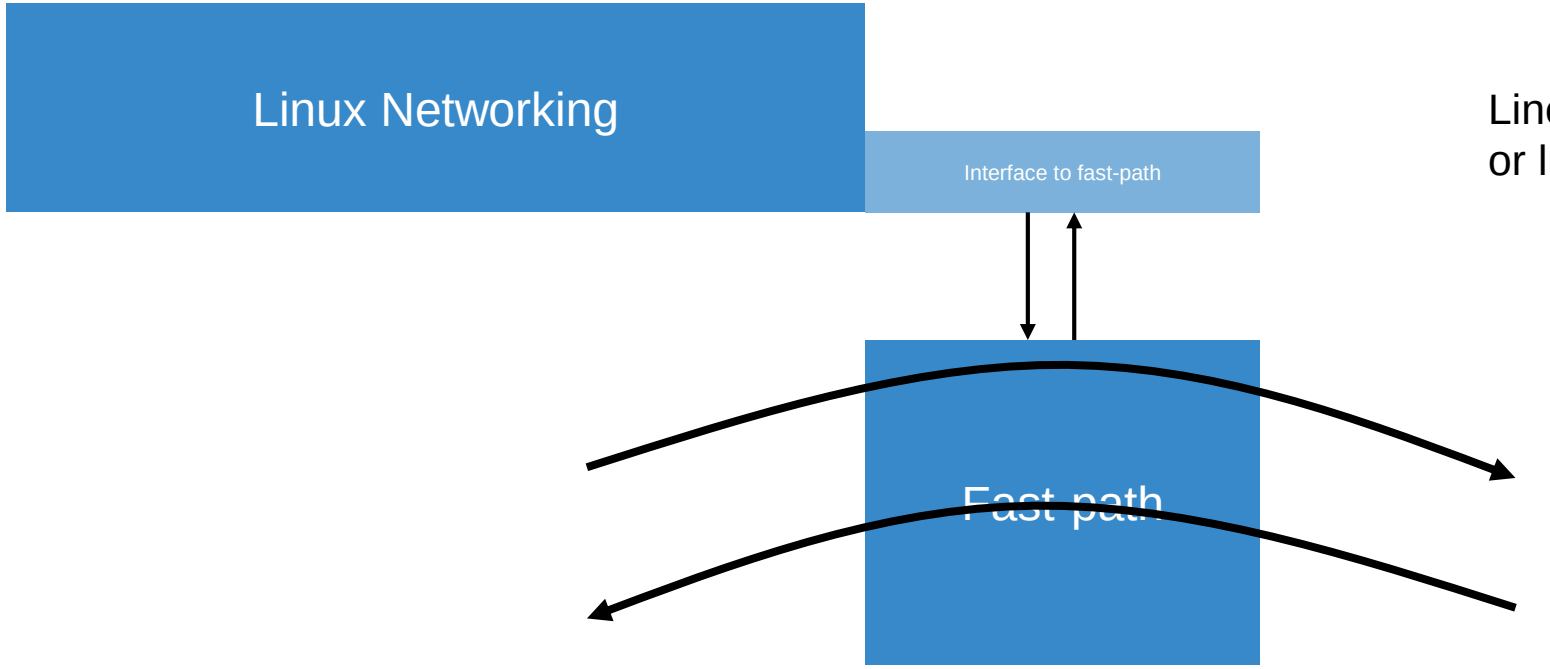
Find us online at www.nxp.com/networking-services

SDK vs. ADK vs. ASK



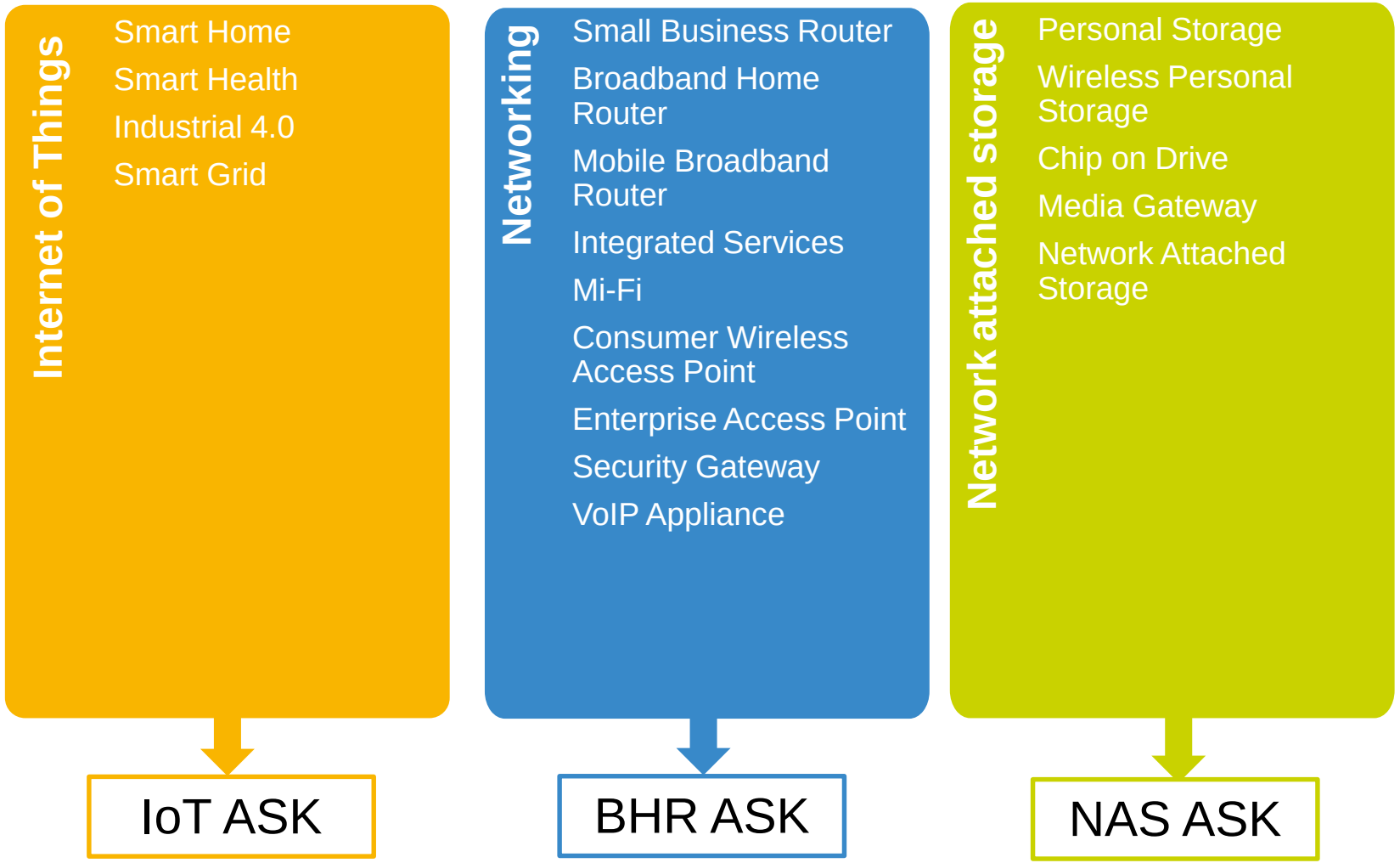
Why Use the ASK

- Production-tested targeted packages
  - Can give a real head-start on end-product
- Gives access to BHR and NAS Fast Path, VoIP binaries and supporting software



Line-rate throughput with little or limited additional CPU load

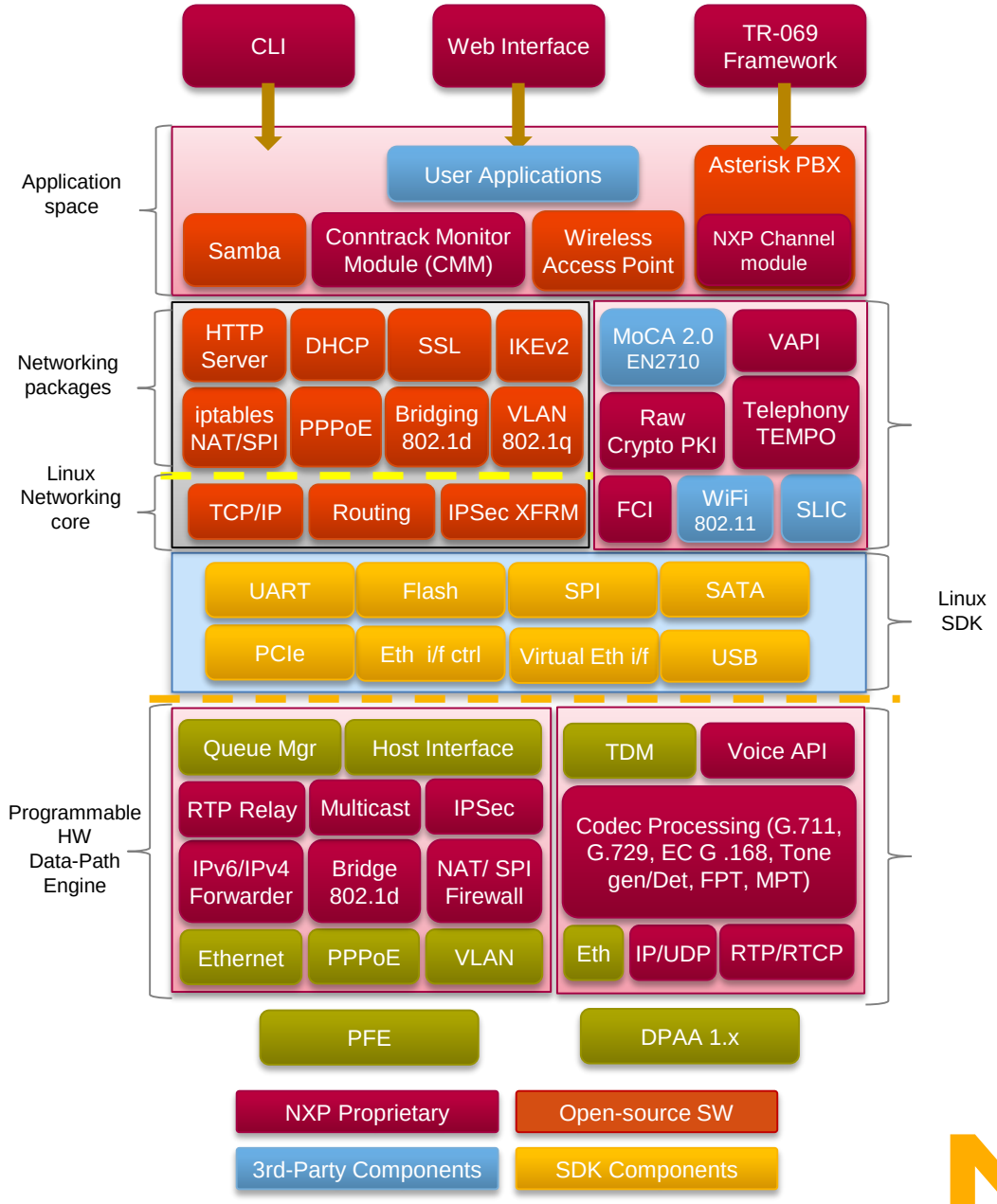
Application Solution Kits (ASK) – LS1



BHR Application Solution Kit

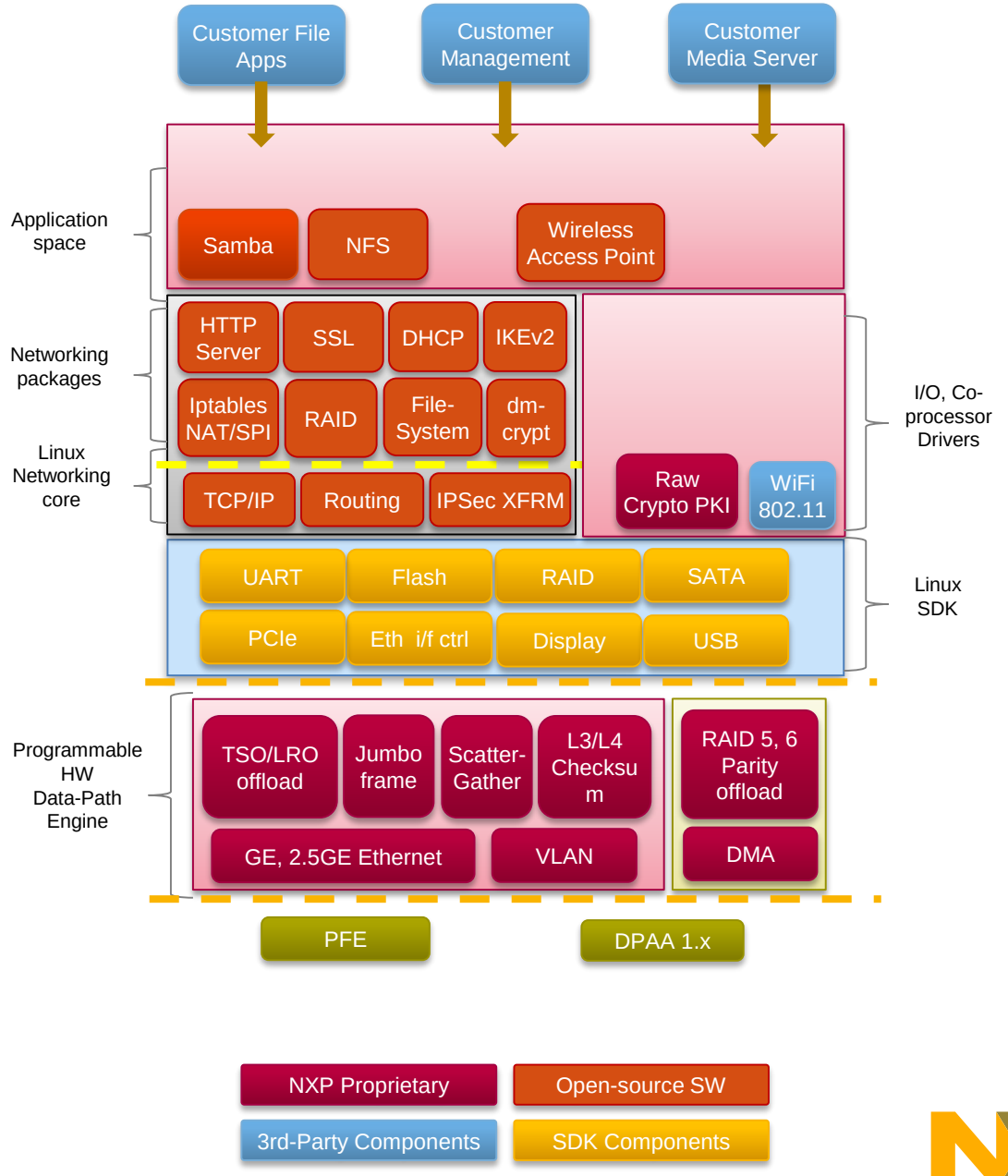
- Turnkey Market Specific Solutions
  - Application, Middleware and SDK integration
  - Full-featured and optimized
  - Deployable directly or via ODM partners
  - Systems Integration and Customization

- Target Markets
  - Multi-Service Gateways
  - Enterprise/Access Gateways
  - WLAN Access-points
  - Consumer/Prosumer NAS
  - Intelligent NIC adapters
  - IoT Gateways



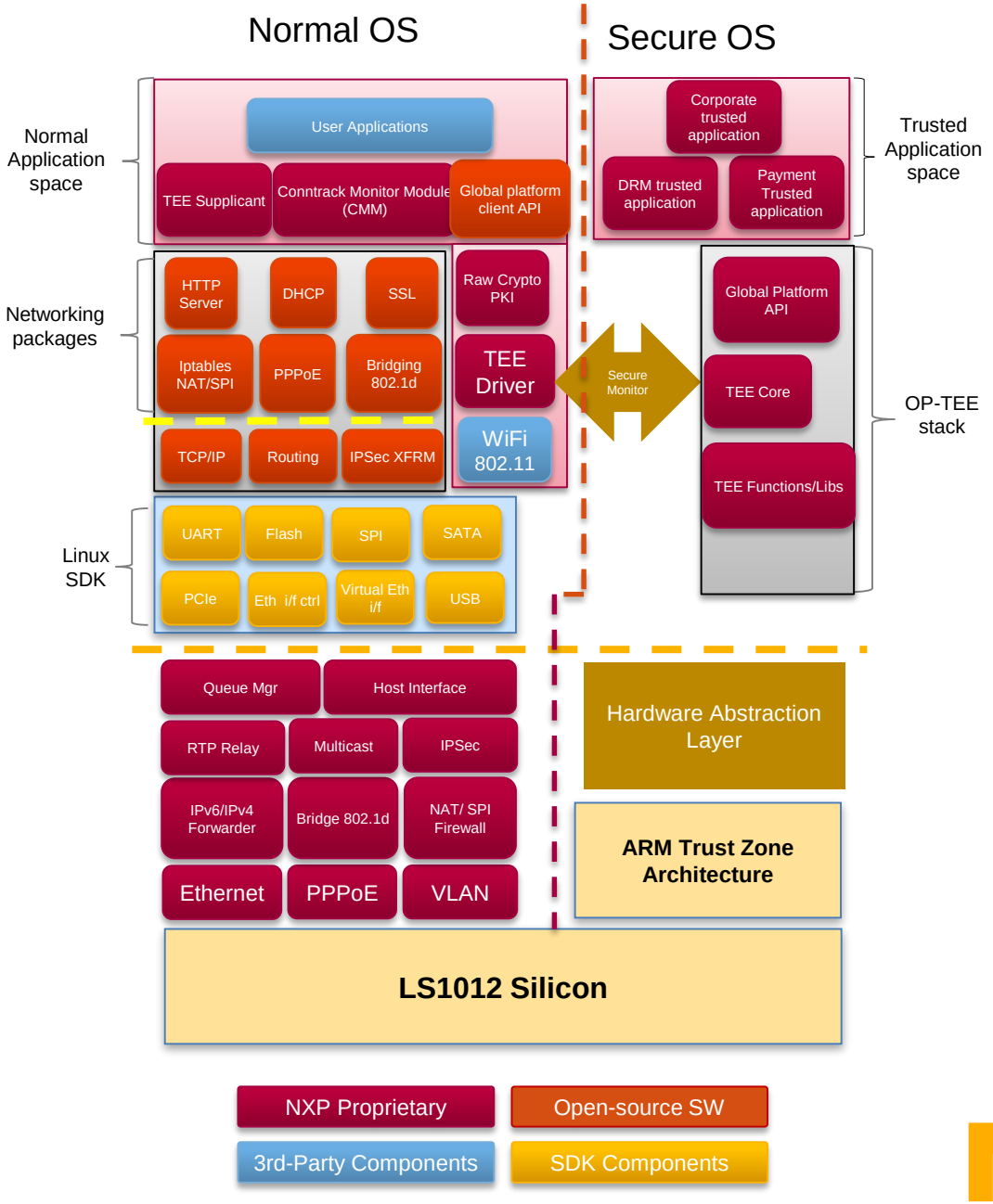
Consumer NAS ASK

- Key Highlights
  - HW offload delivers best-in-class NAS performance
  - Rich set of RAID & file-system support
  - Secure storage and secure access of data
  - Optional VPU(ZN200) integration for hardware transcoding – delivers up to 4 simultaneous HD channels
  - Offloads
    - LRO/TSO, Jumbo-frame, checksum
    - RAID parity calculation, DMA
  - Highly optimized storage stack
    - Samba, NFS, HTTP, SSL
    - RAID 0, 1, 5, 6, JBOD support



IoT ASK (Preliminary Info)

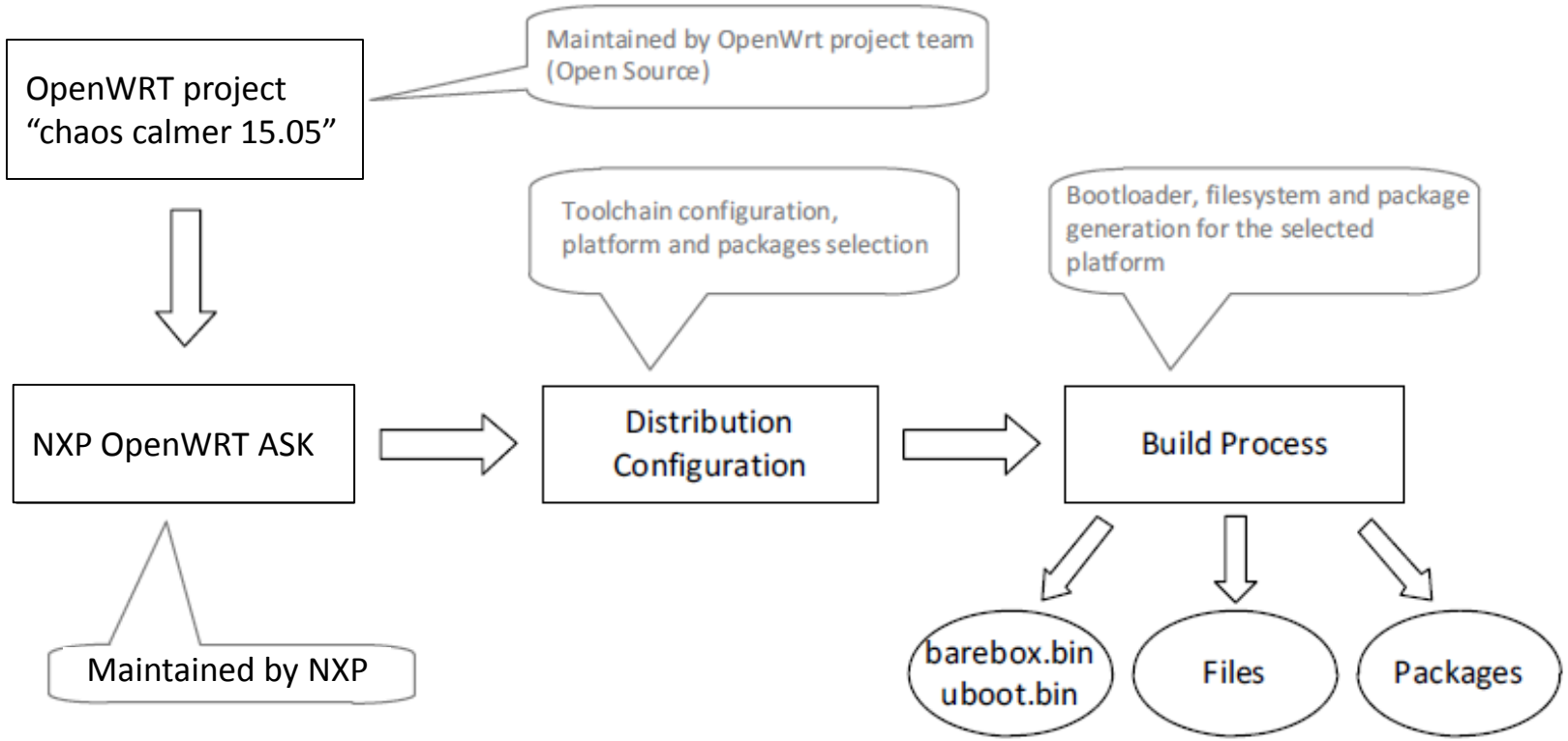
- ARM Trustzone® execution environment
- SEC engine with symmetric and asymmetric crypto support
- Full-featured Op TEE stack
  - Secure OS
  - Secure key management
  - Global platform API support
- Networking and upper application layers
  - Feature rich and optimized networking stack
  - Optional Pre integrated Partner OSGi and JVM layers for quick Time to Market
- Available later 2016



DESIGNING WITH LS1012A, LS1024A, AND LS1043A ASK

OpenWRT ASK

- Complete build and development environment
- Includes sources from the level of microloader, right up to user applications, e.g. Media server, Web GUI
- New features and applications can be added
  - Often ready-made package sources are already on OpenWRT repository



Example Packages in OpenWRT ASK

- Includes:
  - DHCP server, DNS Resolver, ebtables, ethtool, Firewall (IPv4 and IPv6) and NAT, FTP Server, Httpd, IGMP Proxy, ip (Routing Control utility), ipsec-tools, IPv6 DNS Server Discovery Daemon, IPv6 MLD Proxy, IPv6 Multicasting Routing Daemon, IPv6 Routing Advertisement Daemon, net-tools-rarp, Network Interfaces, NTP Client, PPP (PPP Daemon), rp-pppoe relay
- More complete list in software documentation
- New packages are easy to add via OpenWRT package framework

Release Archive Overview

- There are 2 archives that are provided in the ASK
  - `src-openwrt-ls1043a_0.7-rc1.tar.bz2`
  - `dl-openwrt-ls1043a_0.7-rc1.tar.bz2`

- `src-openwrt-ls1043a_0.7-rc1.tar.bz2:`
  - Default board configuration files
  - Makefiles, configuration files, patch files and scripts to build the toolchain, tools, kernel, packages, and generate the final images

- `dl-openwrt-ls1043a_0.7-rc1.tar.bz2:`
  - Default downloads from OpenWRT and other repositories
  - Build system uses this directory of compressed images first before going to the web
  - Makes additional downloads unnecessary

The Linux Kernel

- Central part of the software bundle
- Takes care of most low-level hardware functions, interrupt handling, task scheduling, low-level interface to fast-forward engine
- Where most of the network packet processing is done
- Code running inside the kernel can be compiled within the kernel image, or as a separate module loaded when necessary:
  - In the latter case, the kernel module sources can also be kept and compiled separately from the whole Linux source tree
- Modified by NXP to:
  - Add support for booting on supported NXP devices
  - Support the various RDB H/W interfaces (Ethernet, SPI, PCIe, SATA, Flash memory, UART, GPIOs…), and external devices (e.g. L2 switch, WiFi, SLICs)
  - Improve performance in certain areas (mostly network processing)
- Kernel version 3.19.3 currently used in latest LS1043A ASK

Preparing the Development Environment

- Run "`make menuconfig`" to check pre-requisites on your machine
  - Install missing packages e.g. `sudo apt-get install <package_name>`
  - git, automake 1.11.1, autoconf 2.65, auto-tools-dev 20050803.1, cpp, cvs, ctags, flex , g++, gcc, kernel 3.2.x, libtool 2.2.6b, libncurses5-dev, libncurses5, make 3.81, uuencode, zlib1g, zlib1g-dev, subversion , gawk, perl & python, lzop
- Approximate required disk space for compiling the OpenWrt stable source is 8 GB

- Build the system without super user privileges:
  - Running as "root" user not recommended to compile and install the software

- Suggested to use Ubuntu 12.04 or later versions for development environment
  - There is no limitation, other Linux distro's can be used
  - Ubuntu14.04 LTS is fine

Directory Tree Overview

- On the Ubuntu system, as non-root, uncompress the code release files

```
$ tar xfj dl-openwrt-ls1043a_0.7-rc1.tar.bz2
$ tar xfj src-openwrt-ls1043a_0.7-rc1.tar.bz2
```

  - This creates and populates two directories: `dl` and `src-openwrt-ls1043a_0.7-rc1`
- Make a symbolic link from the src build directory (aka <buildroot>) to the dl directory

```
$ cd src-openwrt-ls1043a_0.7-rc1
$ ln –s ../dl .
```

  - If you see downloads happening during make of the default config, check the link is there

- Three main directories are created as the src tarball is uncompressed
  - `toolchain`
    - Contains the Makefiles and associated files for all software related to the cross-compilation tool chain
  - `target`
    - Contains the items for a specific embedded platform, e.g. kernel patches
  - `package`
    - Contains all the Makefiles and scripts for the packages available within the distribution.

- Note
  - Both the target and package steps use `build_dir` as a temporary directory for compiling
  - Anything downloaded by the toolchain, target, or package steps will be placed in the "dl" directory

Directory Tree Overview

- `toolchain`
  - At build time, the following two new directories are created
    - `toolchain` which is a temporary directory used for building the tool chain for a specific architecture
    - `staging_dir` where the resulting toolchain **“binaries”** are installed
  - There is no need to do anything with the `toolchain` directory unless a new version of one of these components is added
- `target`
  - `target/linux/<device_name>` is platform-specific and contains the kernel .config file and kernel patches for the device being used
  - `target/linux/imagebuilder` describes how to package a firmware for a specific platform
- `package`
  - Most of the firmware is packaged as .ipk modules (e.g. applications, drivers, libraries) which can be installed on a running system – more on this later !
  - This can provide new features or remove features to save space

Directory Tree Overview

- `bin`: contains the final binary images created during the build process
- `build_dir`: non toolchain source code and compiled images
- `configs`: configuration files for reference and development boards
- `dl`: all of the OpenWrt source code packages
- `docs`: OpenWrt documentation
- `include`: Default core makefiles (*.mk) for OpenWrt (e.g. kernel, packages, filesystem)
- `packages`: OpenWrt and NXP generated makefiles and source code patches
- `scripts`: scripts supporting the build makefiles
- `staging_dir`: compiled toolchain, including library includes, used to compile the rest of the distribution
- `tools`: necessary tools to build the image

USING THE ASK TO MAKE RDB BINARIES

Openwrt Configuration File

- In the NXP ASK OpenWrt distribution, there are pre-configured `.config` files for the development and reference platforms
- These files contain the hardware and software configuration parameters to operate the board
- Developers may use these configuration files as a template in order to generate the OpenWrt image
- Current platforms supported
  - LS1043A RDB: `config-ls1043a-rdb, config-ls1043a-qds` (NXP internal hardware)

- Pick the appropriate config file for your board, and copy into buildroot as `.config`:

```
$ cd src-openwrt-openwrt_ls1043a_0.7-rc1
$ cp configs/config-ls1043a-rdb .config
```

Customizing Openwrt Options with `make menuconfig`

- Menuconfig is a menu-driven configuration tool (using ncurses) to select all features
- The developer can select the targeted platform, which versions of the toolchain to use, and what packages to install into the firmware image
  - Toolchain (e)glibc 2.19 is supported by default
- Similar to the Linux kernel configuration
- Menu options for configuring an OpenWrt build
  - Long list of features, sorted into different sections
  - Selecting items from the menu by pressing the y, m, or n keys
    - **<y> - will be compiled and included in the firmware image**
    - **<m> - will be compiled but not included in the image, may be used for a later, runtime install**
    - **<n> - will not be compiled or included**

- Running it:

```
$ cd sdk-openwrt-openwrt_ls1043a_0.7-rc1
$ make menuconfig
```

- Note that running `make menuconfig` first runs a pre-requisite check on your build machine.
  - You have to ensure this check passes before the build will continue
- When in Menuconfig, use '/' to search for a particular config item

- To generate the RDB binaries, don't change the config, but save on quitting Menuconfig

Building Everything

- Launch the build:

  ```
  $ make
  ```

- This will take a while; around 1 hour the first time, depending on build machine
- If some package sources are needed, they will be downloaded from the OpenWRT repository or the Internet, but all defaults are already in `dl`
  - If Internet access is through a proxy, set the 'http_proxy' environment variable
    - Package download is done with wget
- Locations of the resulting binaries for bootloader, kernel, rootfs, and packages:
  - `bin/ls1043a-glibc/`
- The file system tree generated by OpenWrt includes:
  - all packages marked as <*> in menuconfig
  - Linux kernel image
  - voip firmware where applicable (/`lib/firmware/voip.axf`)
  - Fast-path binaries where applicable

Some Specific Make Commands

- Select the number of compilation machine CPU cores to be used for make
  `$ make -j n`
- Verbose make output
  `$ make V=99`
- Recompile the kernel
  `$ make V=99 target/linux/{clean,compile}`
  `$ make V=99 target/linux/install`
- Compile a specific package (see later)
  `$ make V=99 <package_Makefile_location>/compile`
  `e.g. make V=99 package/openwrt/net/tcpdump`
- Clean up a specific package
  `$ make V=99 package/openwrt/net/tcpdump/clean`
- Clean up compilation
  `$ make clean`
  - Deletes contents of the directories /bin and /build_dir
  - Does not remove toolchain, nor does it clean architectures/targets other than those selected in .config
- Start again from scratch...use with care !
  `$ make distclean`
  - nukes everything you have compiled or configured and also deletes all downloaded feeds contents and package sources.
  - **CAUTION**: In addition to all else, this will erase build configuration (<buildroot_dir>/.config), toolchain, and all other sources.
  - Atfer using this, you’re back to the point where you’ve just uncompressed the supplied src and dl tarballs.

Build Procedure Summary

- Step 1
  - Decompress the ASK tarball files

```
$ tar xfj dl-openwrt-ls1043a_0.7-rc1.tar.bz2
$ tar xfj src-openwrt-ls1043a_0.7-rc1.tar.bz2
```

- Step 2
  - Create link to the dl directory

```
$ cd src-openwrt-ls1043a_0.7-rc1.tar.bz2
$ ln –s ../dl .
```

- Step 3
  - Copy and rename the configuration file at the top level

```
$ cp configs/config-ls1043ardb .config
```

- Step 4
  - Customize or inspect the build options

```
$ make menuconfig
```

- Step 5
  - Run the make command to build the code

```
$ make
```

ADDING AND INSTALLING A PACKAGE

Enable new feature via Menuconfig

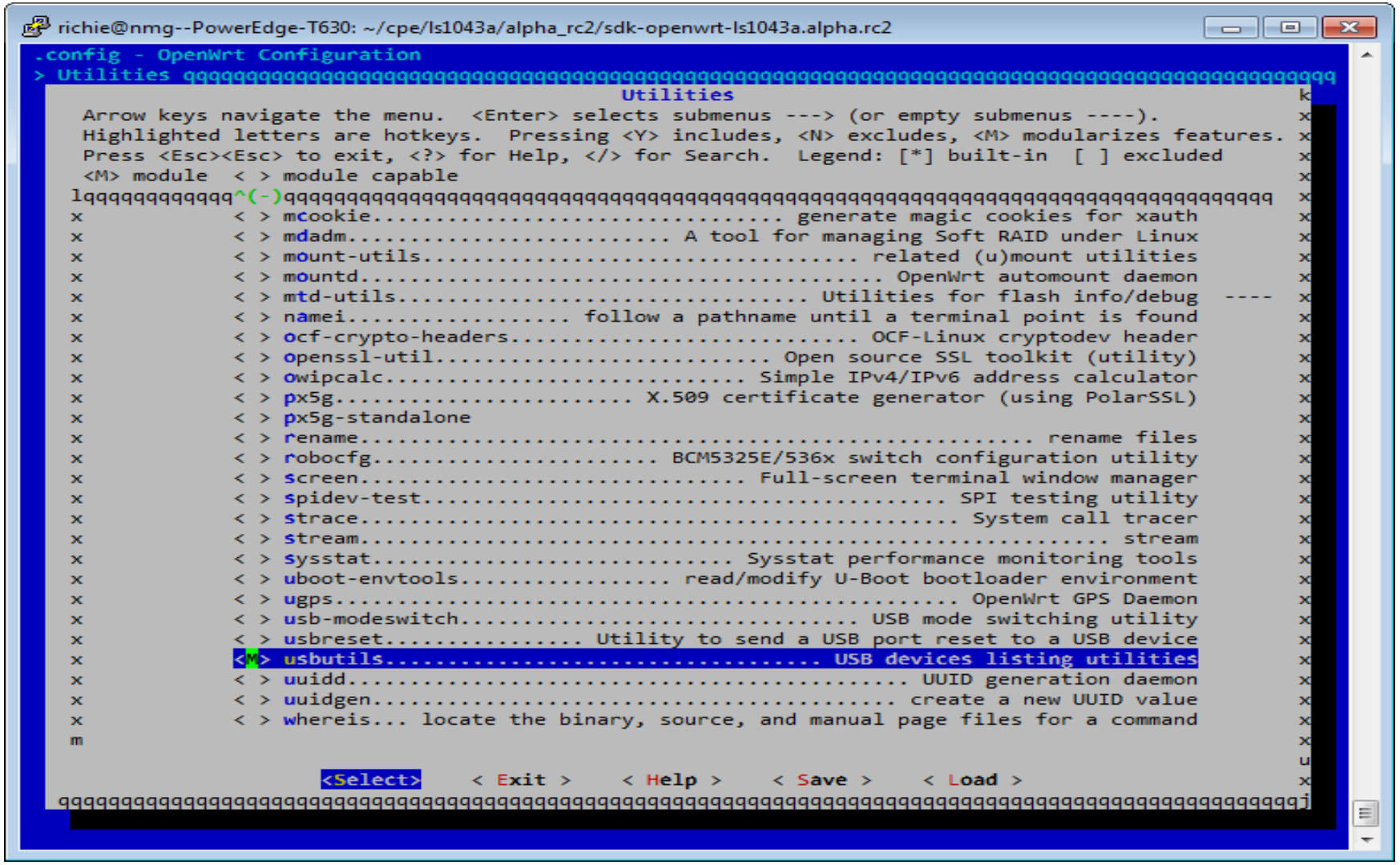
- Not all available items are enabled in a default ASK build
- Plenty of Menuconfig items can simply be enabled and built
- Make process downloads new package from online repo, confirms MD5 and builds it
- Example of USB-utils, which includes ‘lsusb’ command

```
$ make menuconfig
```

- Under ‘Utilities’ Menu, scroll down to ‘usbutils’, hit <spacebar> until ‘<M>’ is seen.
- Hit <ESC> and select <Yes> to save config.

```
$ make V=99 package/utils/usbutils/compile
```

Menuconfig showing usbutils selected



New package and any needed libs are built

- The resulting package and library installables are placed in .../bin/packages directory

```
richie@nmg--PowerEdge-T630:~/cpe/ls1043a/0.7/src-openwrt-ls1043a_0.7-rc1$ ls -lht bin/ls1043a-glibc/packages/base/
total 4.7M
-rw-r--r-- 1 richie richie 203K Apr  8 10:49 usbutils_007-1_ls1043a.ipk
-rw-r--r-- 1 richie richie  26K Apr  8 10:49 libusb-1.0_1.0.19-1_ls1043a.ipk
```

- Take the new .ipk's, transfer to board (e.g. scp, USB), and install

```
opkg --install <package_name>
```

- It will tell you if you need to install a pre-requisite package (e.g. libusb in this case) first

Example of USB-utils package Makefile

- The key item needed is the Makefile

usb_utils_ls1043_Makefile

Get Package from OpenWRT Packages Repository

- https://downloads.openwrt.org/sources/
  - Shows packages already available for OpenWRT
  - Only Makefile, ‘files’ directory, and ‘patches’ directory & contents needed

openwrt_transmission_Makefile

```
$ make V=99 packages/<new_package>/compile
```

- .ipk(s) are placed under .../bin

- Example of ‘transmission’ bittorrent client
  - https://github.com/openwrt/packages/tree/master/net/transmission
  - Put these in .../packages/openwrt/network directory

- Easiest, create a local clone of the openwrt git repos:

```
$ git clone git://git.openwrt.org/12.09/openwrt.git
$ git clone git://git.openwrt.org/12.09/packages.git
```

  - and copy transmission package files from here into buildroot

Use Existing Makefile as Template

- If there isn’t an existing OpenWRT Makefile, use an existing one from .../packages/ as a template.
- Don’t forget to enable it in menuconfig
- To make things easier, download the source tarball into .../dl first, and check the MD5 manually

ADAPTING THE ASK FOR YOUR HARDWARE

Device Tree

- LS1043A ASK OpenWRT is based on source code from the LS1043A yocto-SDK
  - Does not use Yocto distribution
  - Modify for specific hardware per Yocto documentation
    - Primarily making sure the kernel device tree matches the hardware

Board File (LS1024A Example)

```
/* ------------------------------------------------------------------
 *  NOR device
 * ------------------------------------------------------------------ */
#if defined(CONFIG_MTD_COMCERTO_NOR)

static struct resource comcerto_nor_resources[] = {
        {
                .start  = NORFLASH_MEMORY_PHY1,
                .end    = NORFLASH_MEMORY_PHY1 + SZ_64M - 1,
                .flags  = IORESOURCE_MEM,
        },
};
```

- Modify board file to match desired hardware config

ASK DOCUMENTATION

ASK Documentation

- Software Documentation
  - Programmer's Guide
  - Architecture Manual
  - VAPI User Guide (html)
  - MSP API
  - Debug Guide

- OpenWRT has a great Wiki...

SECURE CONNECTIONS
FOR A SMARTER WORLD

ATTRIBUTION STATEMENT

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, CoolFlux, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE Classic, MIFARE DESFire, MIFARE Plus, MIFARE FleX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TrenchMOS, UCODE, Freescale, the Freescale logo, AltiVec, C 5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and µVision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. © 2015–2016 NXP B.V.