Implementing Security on the Embedded Internet of Things using SE for Android

Andreas BURGHART
Solution Sales Engineer
Andreas.Burghart@digi.com

Agenda

- Digi Introduction
- Digi Freescale-based Embedded Solutions
- Android Operating System Overview
- Security Enhancements (SE) for Android – Details
- Digi Development Kit Offering

Digi: Strength In Numbers

285
PATENTS ISSUED AND PENDING

100M
THINGS CONNECTED

25K
CUSTOMERS

DGII NASDAQ	1985 Year Founded	600 Employees Worldwide	12 Consecutive Years of Profitability	200 Million In Revenue	100 Million In Cash

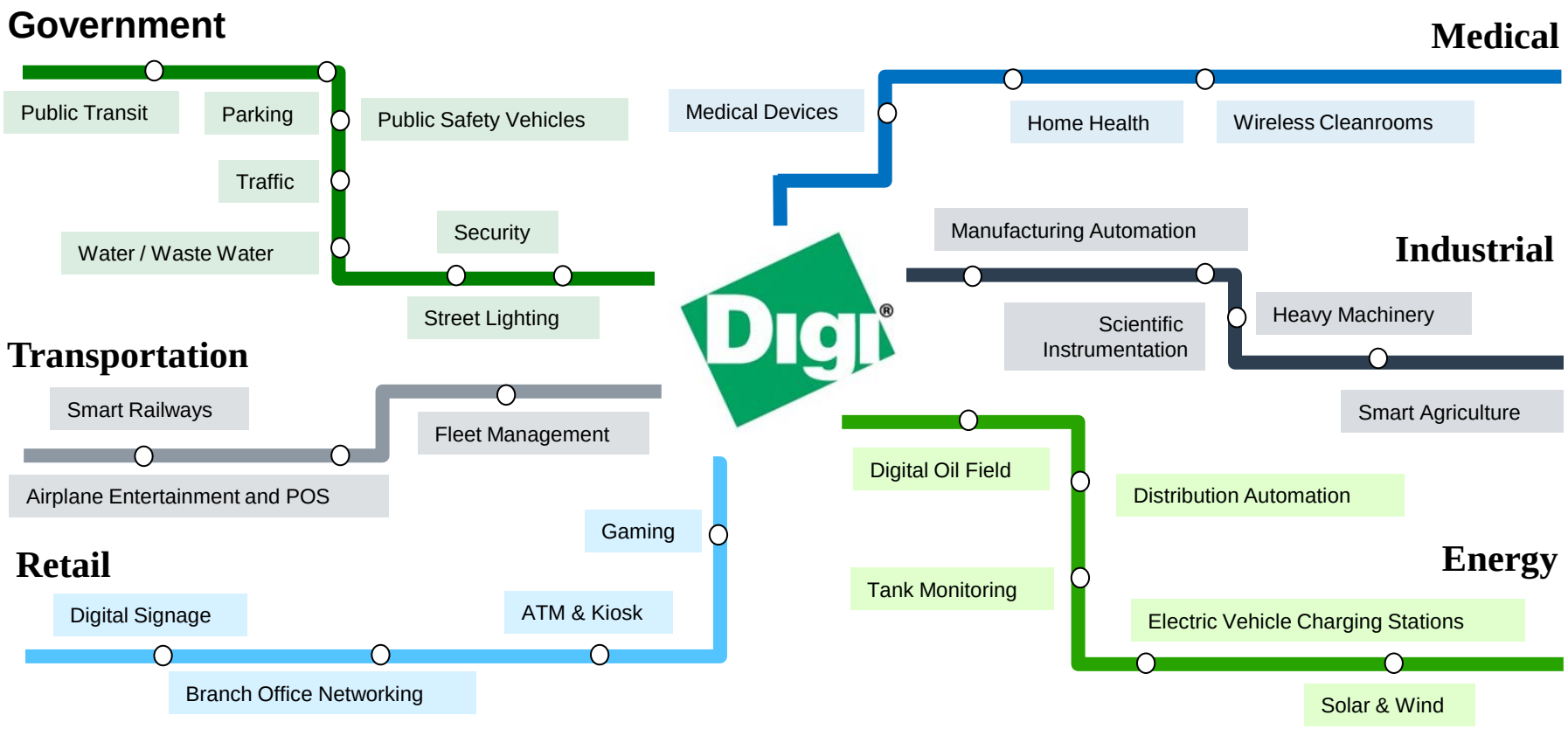
Extensive Global Reach

HQ Minnetonka, MN, USA	16 Regional Offices	200+ Digi Technical Resources	250+ Channel Partners	800+ Channel Technical Resources

Digi EMEA

- **HQ in Paris:**
  - European Management
  - Marketing & Sales
- **Sales Offices:**
  - Belgium, Denmark, France, Germany, Russia & Eastern Europe, Spain, The Netherlands, UK
- **Admin & Support Center in Dortmund:**
  - Finance
  - Product Specialists
  - Technical Support
- **R&D locations:**
  - Logroño, Spain
- **Distribution Channel:**
  - Strong network of distribution partner and Integrators

Success Across Six Industries



Award-Winning Products & Services

Create

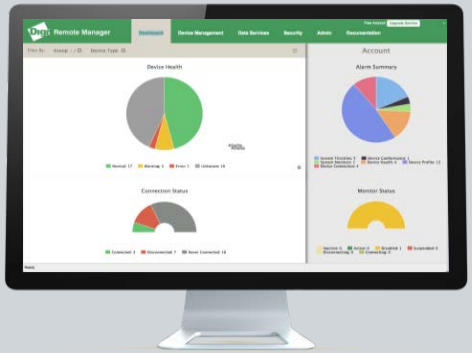
RF Modules
Embedded Modules & SBCs
Wireless Design Services

Deploy

Wireless Routers and Gateways
Device Networking Solutions

Manage

Digi Remote Manager
Digi Device Cloud
Etherios Cloud Services

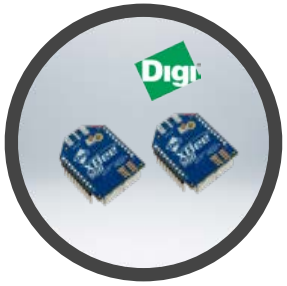


Complete end-to-end IoT Solution

Equipment

Infrastructure

Application



Sensor/Nodes

Aggregator/ Gateway

ConnectCore for Freescale i.MX

ConnectCard for i.MX28

- Freescale i.MX28
- ARM926EJ-S running at up to 454 MHz (1.2 DMIPS/MHz)
- 802.11a/b/g/n + Bluetooth 4.0, single/dual 10/100 Mbit/s Ethernet
- LCD, UART, USB, CAN, SPI, I2C, I2S, ADC, GPIO
- PCI Express Mini Card form factor (51 mm x 35 mm x 3 mm)
- Up to -40 to 85°C operating temperature

LINUX

i.MX28

ConnectCore for i.MX53

- Freescale i.MX53
- Cortex-A8 running up to 1 GHz (2.0 DMIPS/MHz)
- 802.11a/b/g/n, single/dual 10/100 Mbit/s Ethernet
- Dual-display, 2D/3D GPU, 720p/1080p VPU, dual camera
- USB, UART, SPI, I2C, I2S, ADC, SD/MMC, CAN, SATA, GPIO
- Compact 82 mm x 50 mm x 8 mm footprint
- Industrial operating temperature -40 to 85°C

LINUX

i.MX5x

ConnectCore 6

- Freescale i.MX6 (Solo/Dual/Quad) Multichip Module
- Cortex-A9 running at up to 1.2 GHz (2.4 DMPIS/MHz)
- 802.11a/b/g/n + Bluetooth 4.0, Gigabit Ethernet
- Kinetis KL2/K20 microcontroller assist option
- Up to 4 displays, 2D/3D GPU, 1080p VPU, dual camera
- CAN, USB, UART, SPI, I2C, I2S, SD/MMC, SATA, PCIe, GPIO
- Low-profile 50 mm x 50 mm x 5 mm footprint (SMT)
- Industrial operating temperature -40 to 85°C

LINUX

i.MX6

Freescale i.MX6

DIGI CONNECTCORE 6

ConnectCore 6

ConnectCore for i.MX6

- Freescale i.MX6, up to 64 GB eMMC, 2 GB DDR3
- Up to four Cortex-A9 cores up to 1.2 GHz (2.5 DMIPS/MHz)
- On-module Dialog PMIC with high efficiency
- **Ultra low-power Freescale Kinetis KL2 / K20 (Cortex-M0+/M4) micro for unique power management and customer specific implementations**
- **802.11a/b/g/n + Bluetooth 4.0**, single Gigabit Ethernet (MII) w/IEEE1588
- Up to 4 displays, 3D GPU with up to 4 shaders, up to two 2D GPUs, 1080p VPU
- UART, USB, CAN, MIPI DSI/CSI, CSI, I2C, I2S, crypto/security, MMC/SDXC, PCI Express (x1)
- **SMT module, LGA-400, 50 mm x 50 mm max**
- **Industrial operating temperature -40 to 85°C**

LINUX

i.MX6

- Innovative, cost-efficient and reliable wireless multichip module solution
  - LGA module, BGA optional, 50x50 mm, 400 pads, allowing automated placement
  - Connector-less mounting for reduced system cost and reliability
  - Pre-certified 802.11abgn + Bluetooth 4.0 connectivity options
- Truly scalable embedded platform solution
  - Performance scalability through footprint-compatible single, dual, and quad core variants
  - Unique design-for-cost scalability extending from low to high volume applications
- Quick time-to-market through embedded services offering
  - Design services support for customization, antenna design, cellular integration, regulatory/carrier compliance, etc.

ConnectCore 6

ConnectCore for i.MX6

- Freescale i.MX6, up to 64 GB eMMC, 2 GB DDR3
- Up to four Cortex-A9 cores up to 1.2 GHz (2.5 DMIPS/MHz)
- On-module Dialog PMIC with high efficiency
- **Ultra low-power Freescale Kinetis KL2 / K20 (Cortex-M0+/M4) micro for unique power management and customer specific implementations**
- **802.11a/b/g/n + Bluetooth 4.0**, single Gigabit Ethernet (MII) w/IEEE1588
- Up to 4 displays, 3D GPU with up to 4 shaders, up to two 2D GPUs, 1080p VPU
- UART, USB, CAN, MIPI DSI/CSI, CSI, I2C, I2S, crypto/security, MMC/SDXC, PCI Express (x1)
- **SMT module, LGA-400, 50 mm x 50 mm max**
- **Industrial operating temperature -40 to 85°C**

LINUX

i.MX6

i.MX 6Solo

- Single ARM Cortex-A9 at 1.0GHz
- **512KB** L2 cache, Neon, VFPvd16, Trustzone
- **3D graphics** with 1 shader
- 2D graphics
- 32-bit DDR3 at 400MHz
- Integrated EPD controller

i.MX 6DualLite

- **Dual** ARM Cortex-A9 at 1.0GHz
- 512KB L2 cache, Neon, VFPvd16, Trustzone
- 3D graphics with 1 shader
- 2D graphics
- **64-bit** DDR3 at 400MHz
- Integrated EPD controller

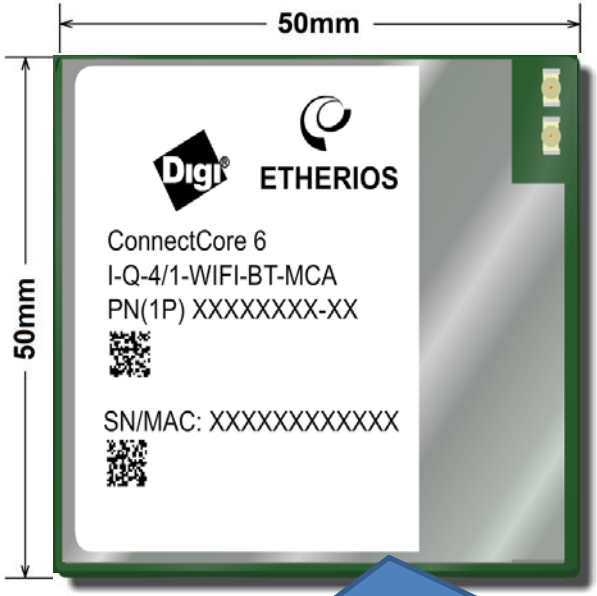
i.MX 6Dual

- **Dual** ARM Cortex-A9 at 1/**1.2GHz**
- **1 MB** L2 cache, Neon, VFPvd16, Trustzone
- 3D graphics with **4 shaders**
- **Two** 2D graphics engines
- 64-bit DDR3 at **533MHz**
- Integrated **SATA-II**

i.MX 6Quad

- **Quad** ARM Cortex-A9 at 1.2GHz
- 1 MB L2 cache, Neon, VFPvd16, Trustzone
- 3D graphics with 4 shaders
- Two 2D graphics engines
- 64-bit DDR3 at 533MHz
- Integrated SATA-II

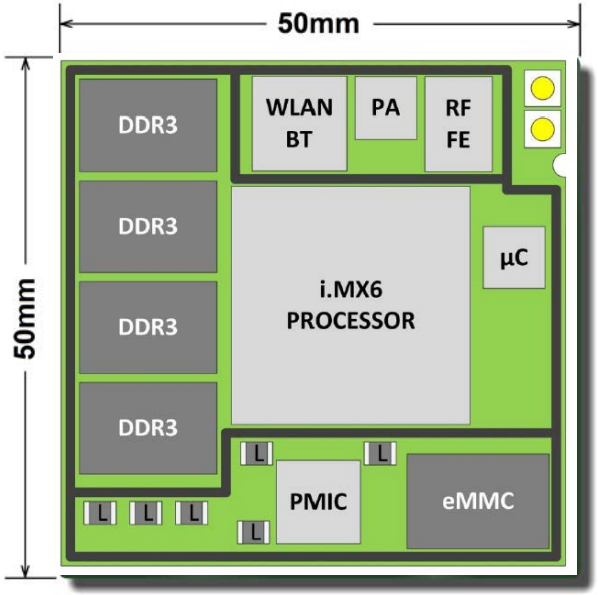
ConnectCore 6: Solution Benefits 1/2



- **Low-profile, ultra-compact form factor**
  - Enabling customers to build compact products, including mobile and semi-mobile applications
- **Common footprint w/scalable performance**
  - Embedded product platform suitable for wide range of applications, both performance and cost
- **Industry's first surface mount (SMT) module**
  - Mounting without connectors for superior reliability, reduced system cost, low profile
- **Completely shielded single component**
  - Low emissions (FCC Class B) and surface for simplified thermal management
- **Integrated Gigabit Ethernet and secure wireless connectivity options**
  - Complete wired and wireless connectivity

- **Dramatically reduced design risk, effort, and accelerated Time to Market**
- **Designed for product reliability and longevity**
- **Guaranteed long-term availability for embedded designs**

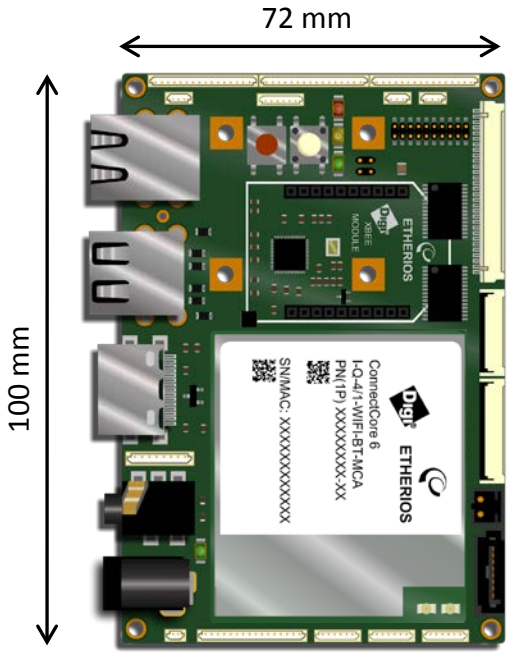
ConnectCore 6: Solution Benefits 2/2



TIM = Thermal Interface Material

- **Superior Thermal Behavior**
  - Thermally modeled design
  - Internal thermal compound (Tputty) to make conduction path to the shield / heatsink
- **True Industrial Design**
  - Industrial rated SOC used on industrial CC6
  - Enabling 24/7 applications >10 year lifetime!
  - Dialog PMIC for superior power management
  - HALT and Vibration tested

ConnectCore 6: SBC Approach



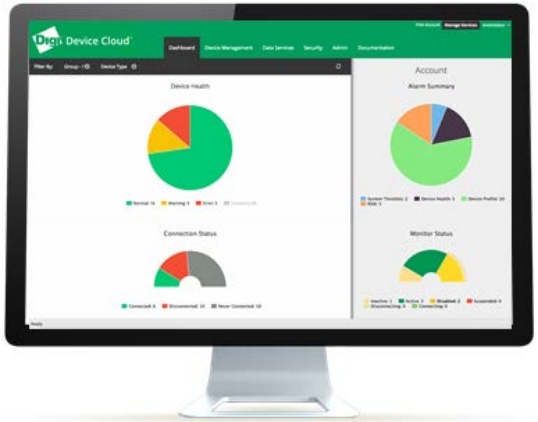
- **Included in ConnectCore 6 dev kits**
  - Carrier board with ConnectCore 6 module
  - Complete design files available (Altium 14) to customers online for reference/customization
- **Also sold to customers as a Digi product**
  - Selected variants will be available
  - Industrial temperature options
  - With product reliability testing
  - Reference enclosure design files posted online
- **Extends reach into new opportunities**
  - Prototyping, proof-of-concept for module
  - New SBC-only opportunities (100 to 1,000 units)
- **Fully connected platform design capabilities**
  - Wi-Fi, Bluetooth, XBee, Cellular, Ethernet

Embedded Operating System Offering

	yocto PROJECT	android	Windows Embedded Compact
Kernel	Linux 3.10	Linux 3.10	Windows Embedded
Positioning	Maximum flexibility – build your own custom Linux distribution QT / GTK support for graphical development	Easy Java application development including graphical user interface programming	Fully componentized and complete offering of high-level Windows components, including GUI , multimedia, and IPv4/v6 networking
Development Tools	Command line / Eclipse plugin C/C++	Digi ESP Java	MS Visual Studio C#,VB,C++

Digi Device Cloud

Digi Device Cloud℠



Enterprise Solution

- Cloud service for Device connectivity, management and integration
- Secure platform for application development
- PCI-DSS validated with “Report on Compliance”

Simplifies Complexities of:

- Taking applications to market
- Integrating new things
- Managing infrastructure
- Managing growth
- Security requirements

Commercial-Grade Reliability:

- Change-Control Management
- Server Management
- Access Management
- Systems & Performance Monitoring
- Logging
- Disaster Recovery

Target Users:

- ✓ Application Developer
- ✓ Solution Provider
- ✓ Value Add Reseller
- ✓ Integrators


ANDROID

Android Adoption

- 1 Billion+ Android device activations since 2008
- Activation rate in July 2013 was 1.5 million/day
- Android becomes increasingly more attractive to embedded developers due to UX integration, Java and a vibrant community

Source: Google, 2014

Consumer Appliance Example

Smart Fridge-freezer

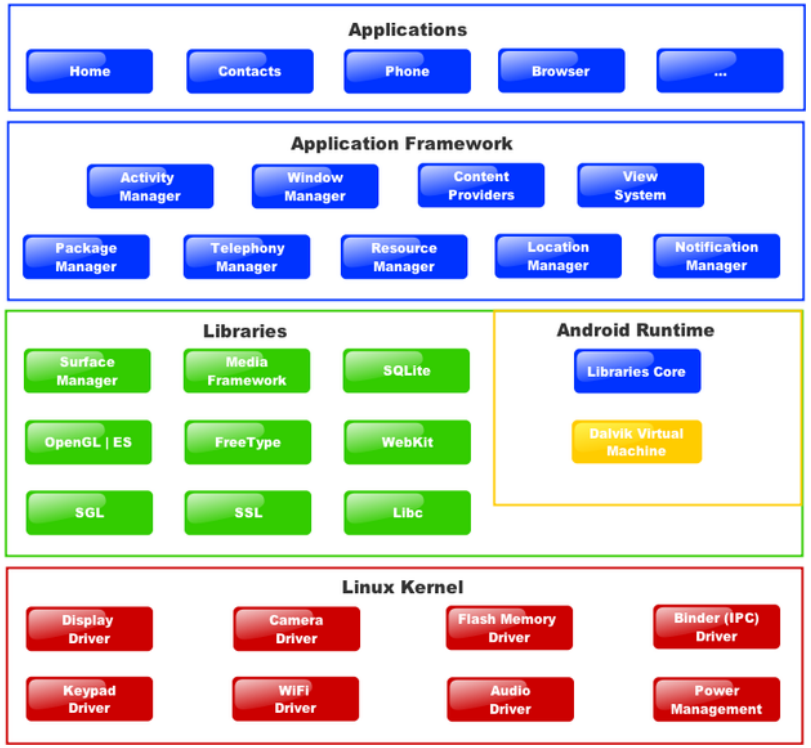
- Samsung RF4289HARS
- WLAN-enabled LCD
- Closed Android system
- Apps included
  - Memos
  - Picasa
  - Epicurious
  - Calendar
  - Weatherbug
  - AP News
  - Pandora
  - Twitter
  - Control Settings

Android Overview

- Complete Software Stack for Mobile Devices
  - Operating system based on Linux kernel
  - Middleware
  - Key Applications
- Royalty free
- Open Source (published by Google)
- Spec defined by Open Handset Alliance
- Goal: Fast & Easy Application Development
  - Applications written in high-level Java
  - Core apps and user apps use same APIs
  - Users can integrate, extend and replace components

Android Details

- Software stack consists of
  - Java applications running on OO application framework
  - Java core libraries / Dalvik or ART JVM / JIT compiling
- Features
  - Open GL ES 2.0
  - SQLite
  - Wi-Fi, Cellular, BT
  - Web Browser
  - Java (and C++) Support
  - (Streaming) Media Support
  - Touch / Camera / GPS / Acceler.
  - Google Play Store for Apps (250k+ apps available)
  - Security Enhancements

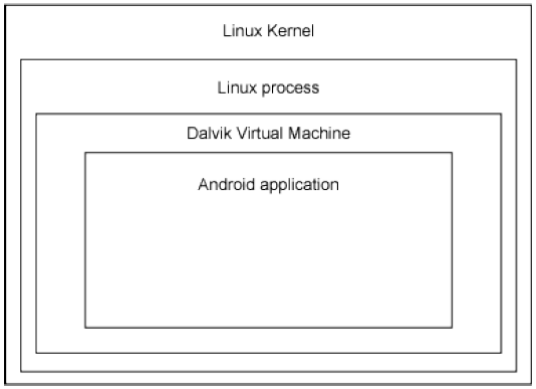


Why Linux as Base System?

- Open source
- Mature, robust and stable
- Secure
- Android is utilizing
  - Linux Memory and Performance Management
  - Linux Network stack
  - Linux Driver Model
  - Linux Security

Android – Architecture

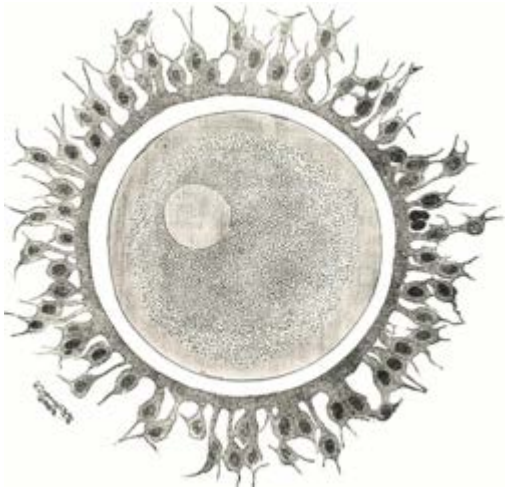
- Each application is started in its own Linux process and as a different Linux user / UID (security)
- Each application runs in its own instance of the Dalvik / ART VM (isolation)
- Each application, by default, has access only to the components that it requires to do its work and no more



- An application can request permission to access device data such as the user's contacts, SMS messages, the mountable storage (SD card), camera, etc.
- All application permissions must be granted by user at install time

Android – Zygote

- Zygote is a special core process started during OS boot process
- As with other Android processes, it runs in it's own instance of the Java VM
- "warmed-up" process that already has core libraries loaded
- Whenever app (new process) is started, it's forked from Zygote
  - There are now 2 VMs (Zygote and the new process)
  - Shared libraries are not copied -> performance gain for starting apps



Why Android? - Summary

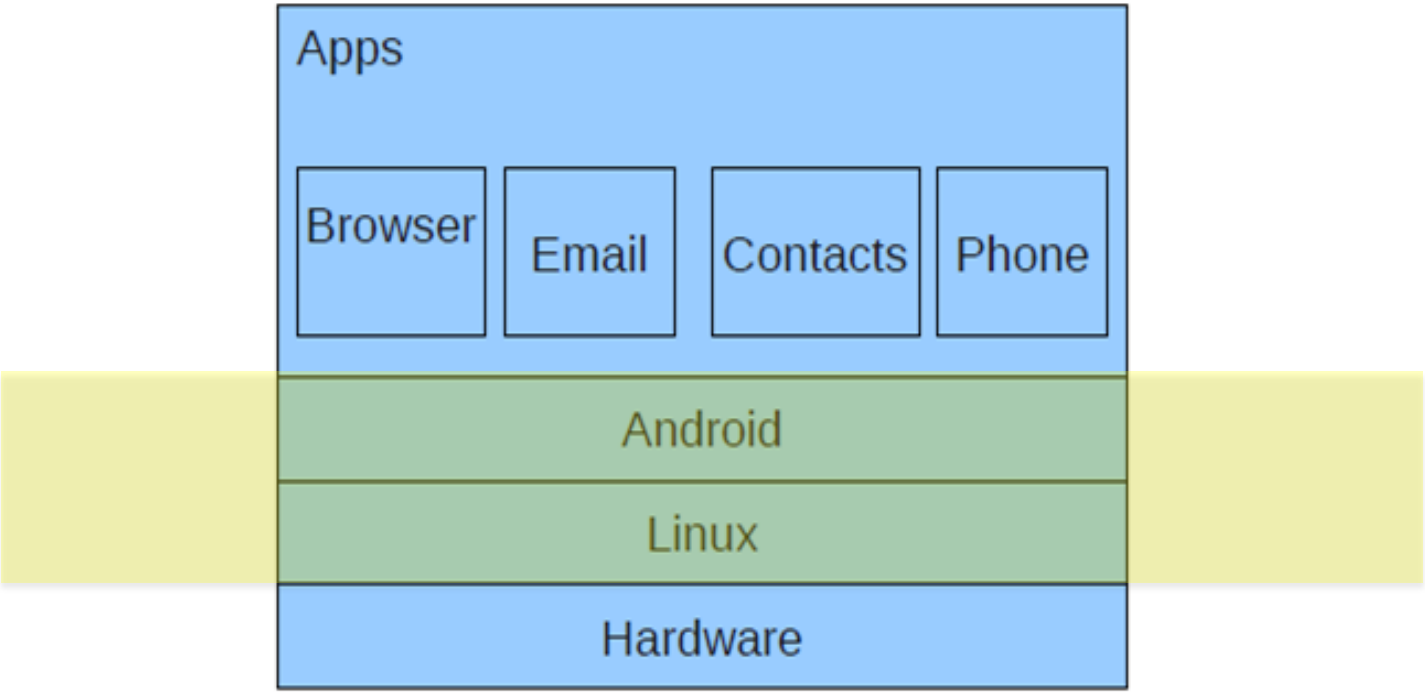
- Development
  - Fast & easy application development
  - Applications written using high-level Java API
  - Structured code / re-use existing components
  - Outstanding graphical (UI) design & implementation capabilities
- Software
  - Networking , Services & Applications
  - Key applications included
- Support
  - Community and professional
- Free
  - No development or run-time royalties
  - Proprietary Google apps licensing required

Android Security Model prior to SE

- **Application-level permissions model**
  - Controls access to app components
  - Controls access to system resources
  - Specified by app writers and seen by users (Manifest)
- **Kernel-level sandboxing and isolation**
  - Isolate apps from each other and from system
  - Prevent bypass of app permission model
  - Normally invisible to users and app writers
- **Enforced by Linux kernel**
  - Java VM is not a security boundary
  - Any app can run native code
  - Relies on default Linux security model
    - Discretionary Access Control (DAC)

Each Android process:
- Uniqe user ID
- Uniqe process ID
- Own VM instance

Android Security Concerns prior to SE



Example: Skype app

- Weaker separation
- Data / system resource access is entirely at the discretion of the app writer
- Prone to privilege escalation (e.g. root exploits)
- No organizational security goal enforcement

Example: vold daemon

Security Enhancements (SE) for Android

Overview

- **Project to identify and address critical security gaps in Android**
  - Open source project with integration into AOSP
  - Mainline adoption started in Android 4.1 (Jelly Bean)
  - Formerly known as “Security Enhanced Android”
  - Now “Security Enhancements for Android”

- **Scope of project not limited to secure OS aspects**
  - Future efforts may include virtualization and trusted computing

- **Derived from SE Linux**
  - Creators of SE Linux, Xen Security Modules, Linux Kernel Integrity Monitor
  - Driven by NSA’s Trusted Systems Research Group
    - Conducts and sponsors research to provide information assurance for security systems

SE for Android Benefits

- Policy driven access control
- Prevent privilege escalation by apps
- Prevent data leakage by apps
- Prevent bypass of security features
- Enforce legal restrictions on data
- Protect integrity of apps and data

DAC vs. MAC

- **Discretionary Access Control (DAC)**
  - Access control in Linux prior SE
  - Owner of the object specifies which subjects can access object
  - Model is called discretionary [illegible] discretion of owner
  - For example, user:group rwx[illegible]

- **Mandatory Access Control ([illegible]**
  - System policy specifies which subjec[illegible]cess specific objects
  - Uses security labels (metadata) att[illegible]ed to objects and subjects
  - For example, user:role:type[:level[:category]]
  - When subject (process) attempts to access object (file), the system (kernel) checks whether the policy allows the subject's context to access the object (Type Enforcement)

If also level (sensitivity) and category are specifed in a security label, this is called Multi Level Security (MLS)

SE for Android Details

- **Mandatory Access Control (MAC)**
  - Enforces a system-wide security policy
  - Over all processes, objects and operations
  - Based on security labels
- **Can confine flawed and malicious applications**
  - Even ones that run as root
- **Sandbox and isolate apps**
  - Stronger separation
  - Prevent privilege escalation by apps
- **Provide centralized, analyzable policy**
  - Small, fixed policy
  - No policy writing for app developers

Each app has its own MLS category

A unique domain is used for every priviliged service

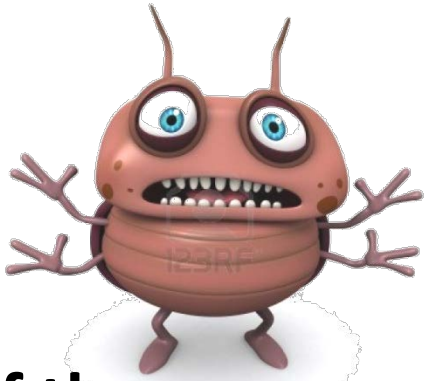
Effectiveness

Root Exploits

- Motochopper
- Mempodroid
- GingerBreak
- Exploid
- Zimperlich
- RageAgainstTheCage
- KillingInTheNameOf

Vulnerable Apps

- Skype
- Lookout Mobile
- Security
- Opera Mobile



SE for Android mitigates ALL of them.

SE for Android

Key Milestones

Jan 2012	Feb 2013	Apr 2013	July 2013	Oct 2013	Nov 2014
SE for Android release Google AOSP invite	Samsung announces KNOX incl. SE for Android	Galaxy S4 ships (>40M sold, Android 4.2.2 + SE)	Android 4.3 (JB) release First general SE for Android support	Android 4.4 (KitKat) release SE for Android in enforcing mode by default	Android 5 (Lollipop) release Fully merged into mainline

SE for Android

Summary

- **Enabling SE Android in enforced mode will significantly improve operational security of your embedded Android device**
  - Proven immunity from common exploits
  - Separation / sandboxing of apps and related data
- **Impact on memory footprint and performance negligible**
  - Kernel footprint increase ~100-150k
- **Policy changes / customization with granularity**
  - Proceed with care and test!
  - Default system policies are provided (across AOSP)
- **Implemented and enabled today**
  - Android 4.2+ , Samsung Knox, Android for Work
- **Project expected to grow beyond current SE for Android**
  - Virtualization, TrustZone, …

Digi Development Kit for Android

Digi Development Kit for Android

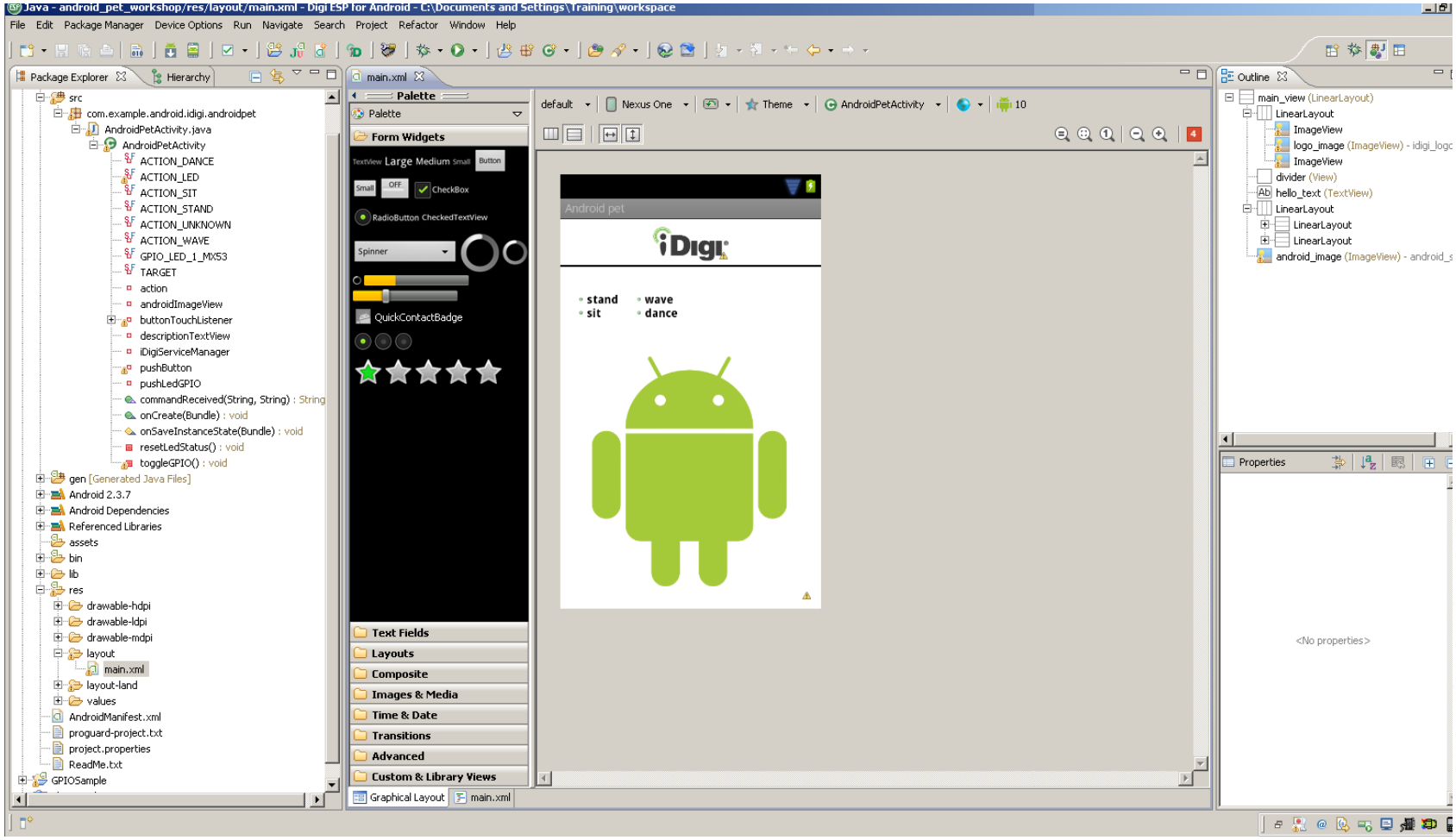
Features

- Ready to use Embedded Android development solution
- Embedded interfaces not common to handsets supported in API
  - e.g. Ethernet, CAN, UART, I2C, SPI, GPIO, ADC, …
- Support for headless operation
- Highly accelerated and efficient application development
  - Graphical UI builder to modify Android user interface
- No or minimal low-level system development effort
- Digi Device Cloud connector

Digi ESP Development Environment

- Based on Eclipse IDE and Android Development Tools (ADT)
- Providing complete tools to build Android applications
  - Project wizards (sample programs)
  - Workbench / Project Explorer
  - Graphical UI builder (Drag & drop, GUI layout XML code generation)
  - Build and powerful debugging tools
  - Terminal view (serial port monitoring)
  - File Explorer
  - LogCat (monitoring debug output from specific Android processes)
  - Android device view (remote device and task manager)
- Package Manager
- Quick Start Guides, Documentation and Help

Digi ESP for Android Development Environment



Digi Presence at Conference

- **Table-top exhibition**
- **Digi ConnectCore 6 SBC Dual-HDMI Android Demo**

Thank You