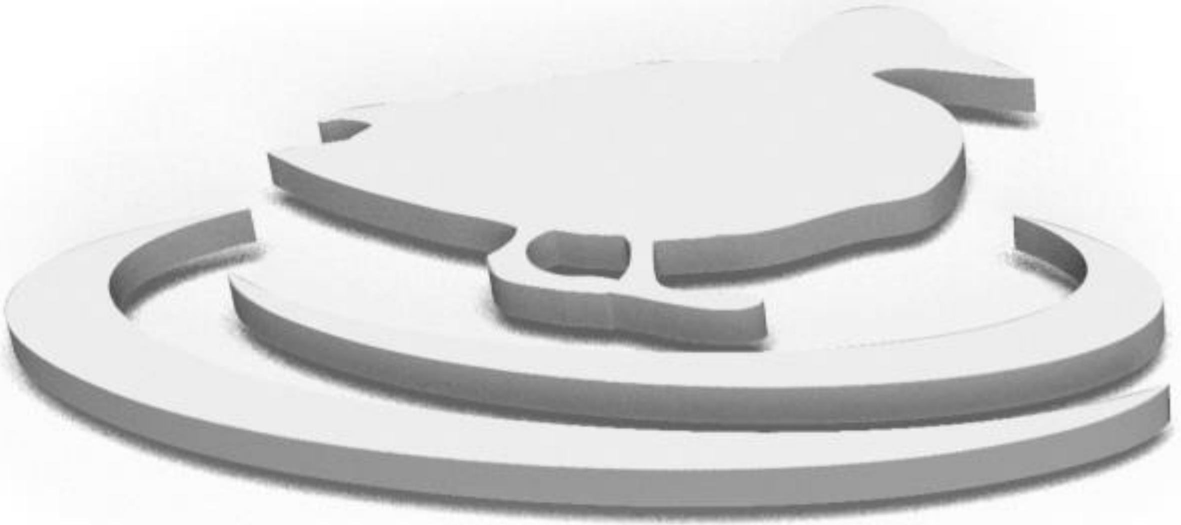
Managing the Android Supply Chain

Peter Vescuso

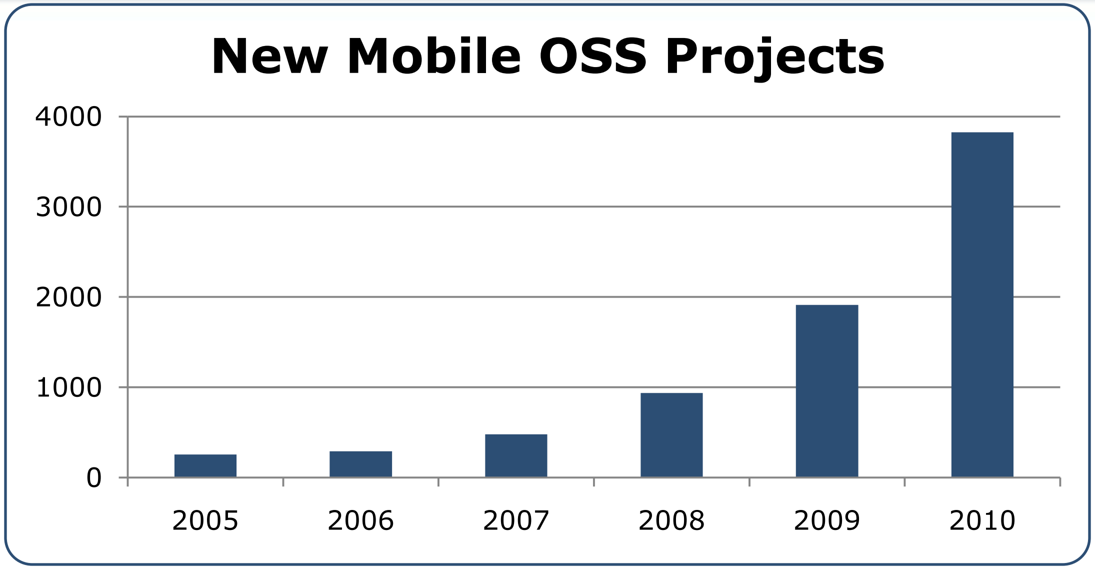
Black Duck Software

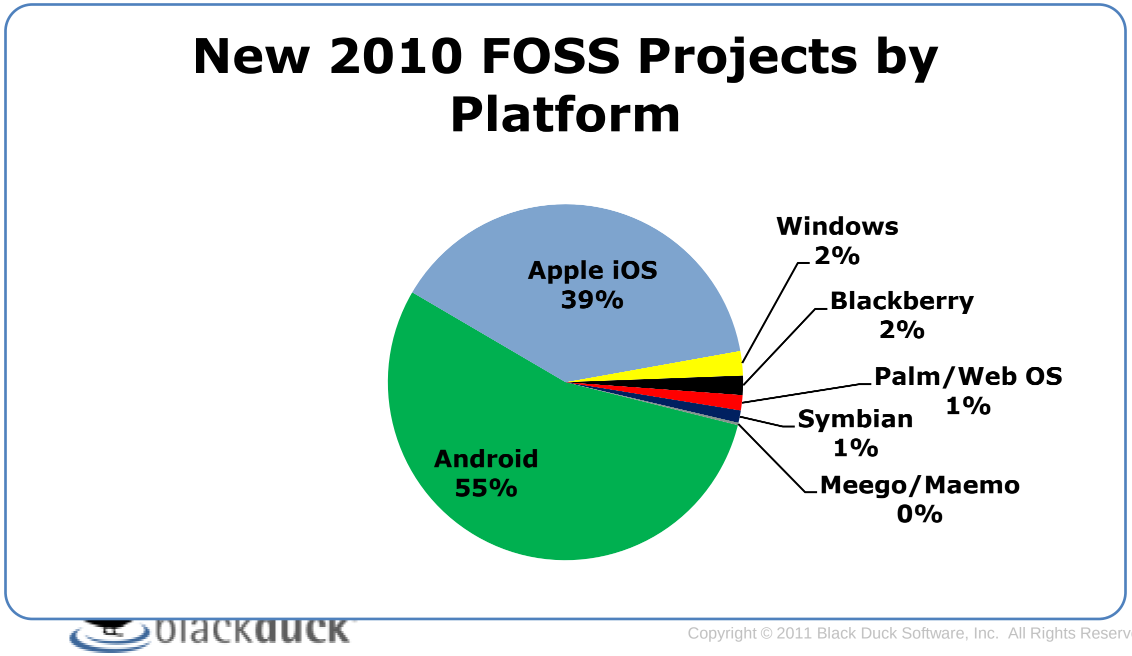
Agenda

- FOSS in Mobile Trends
- Device Manufacturers
- Application Developers
- Supply Chain Management
- Summary



Open Source Drives Mobile Innovation



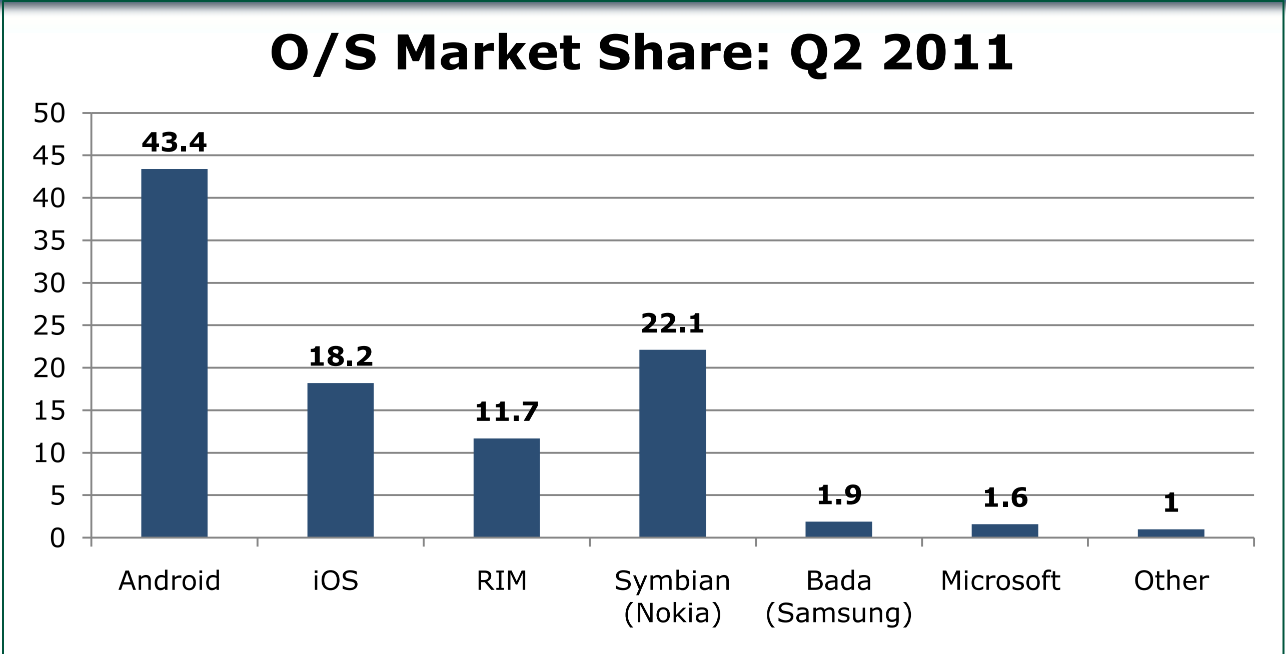


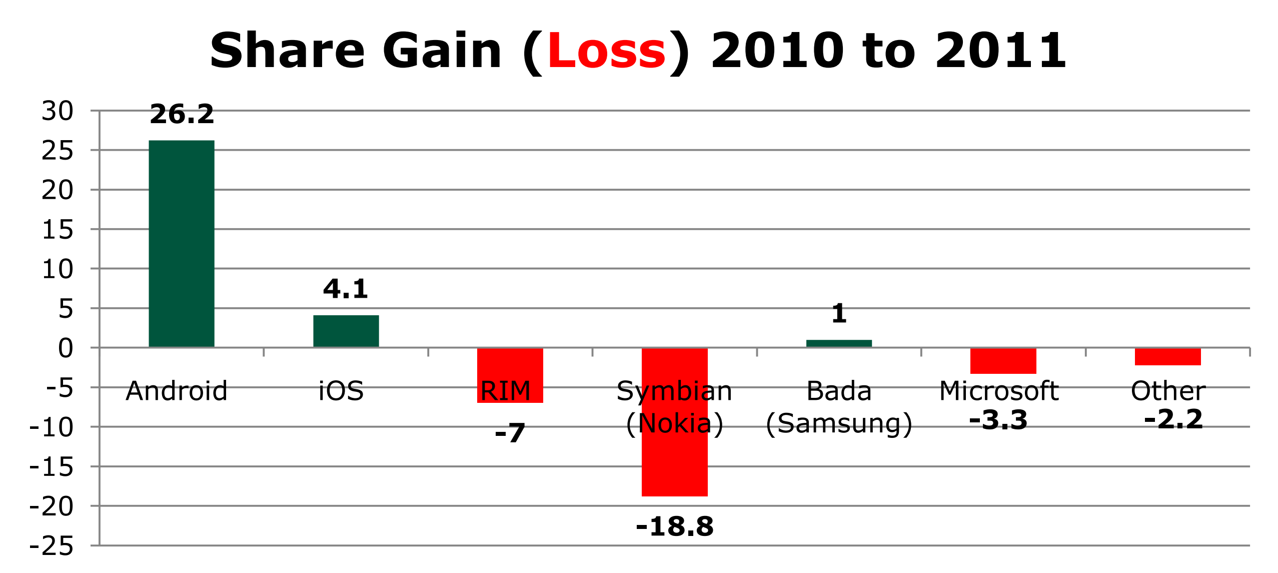
- Over 3,800 new OSS projects in 2010, doubling each of the last 3 years
- 94% of new projects that specify a platform are targeting Android and Apple/iOS
- Open source has redefined the mobile industry and is spreading far beyond

blackduck

Know Your Code.

Android is a Large, Growing Opportunity





- 428.7 million units
- 16.5% growth form Q2 ’10

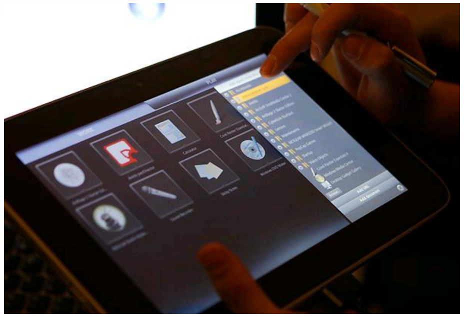
Source: Gartner , August 2011

Know Your Code.

Android Devices: Phones, Tablets, eReaders, Autos, more.....

Automobile: Android powered SaaB

Barnes & Noble Nook



Lenovo LePad

Droid by Motorola

Samsung Galaxy

Dell Streak

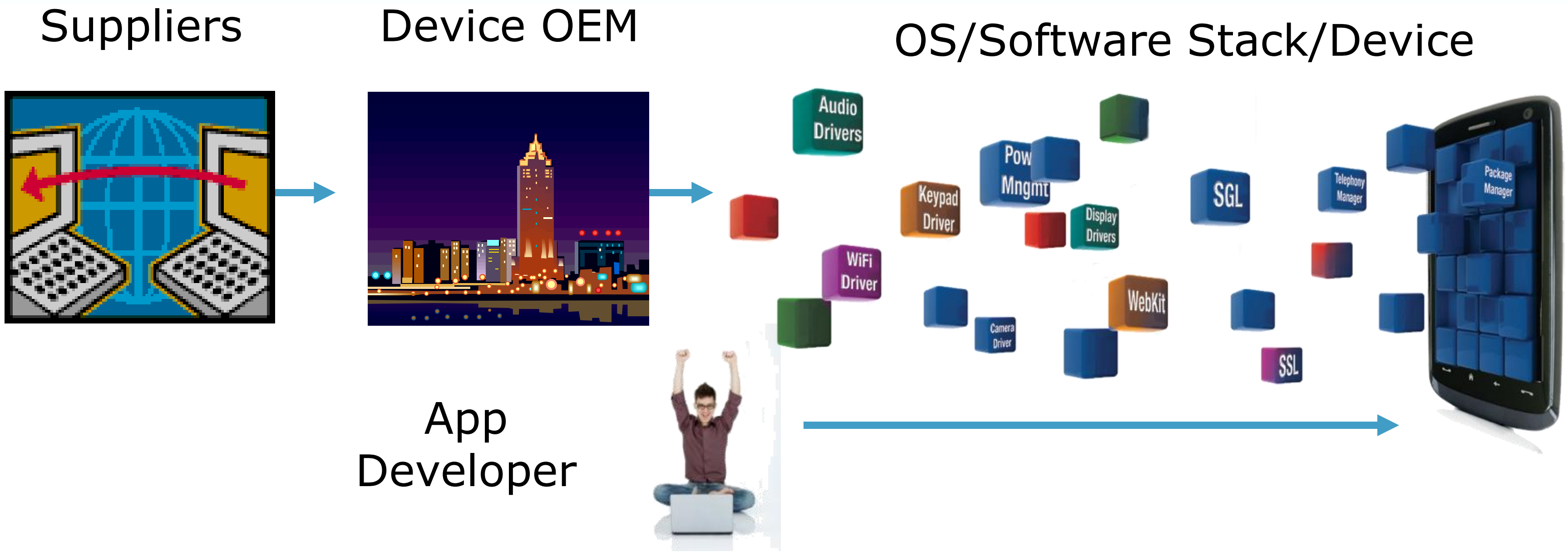
HP Touchpad

HTC Evo Shift

Sony Internet TV

Motorola Xoom

Managing FOSS in the Android Ecosystem and Software Supply Chain



Suppliers

Device OEM

OS/Software Stack/Device

App Developer

Typical Smartphone has over <u>300</u> components

- *Corporate-Owned IP*
- *Proprietary/Licensed IP*
- *FOSS*
- *Outsourced development*
- *Multi-level supply chains*

- *Security*
- *Networking*
- *Email*
- *Graphics*
- *Database*
- *Web Services*
- *Many more…*

blackduck

Know Your Code.

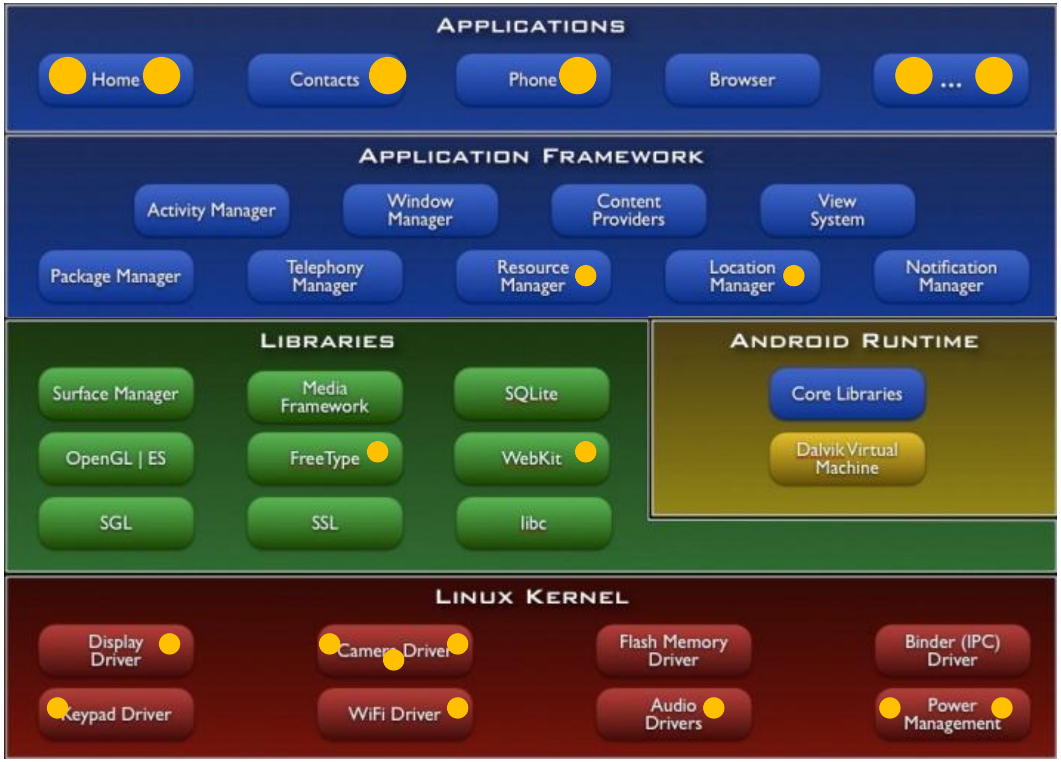
Agenda

- FOSS in Mobile Trends
- Device Manufacturers
- Application Developers
- Supply Chain Management
- Summary

Complexity for Device Manufacturers

- Components and code from many suppliers
- Need to control and manage building software on a rapidly changing O/S
  - Multiple releases per year
- Customize Android for:
  - The type of device (phone, tablet, TV, etc.)
    - Device drivers, power consumption, etc.
  - User experience
- Do it all while ensuring compliance

Android & Vendor Innovation



Developers

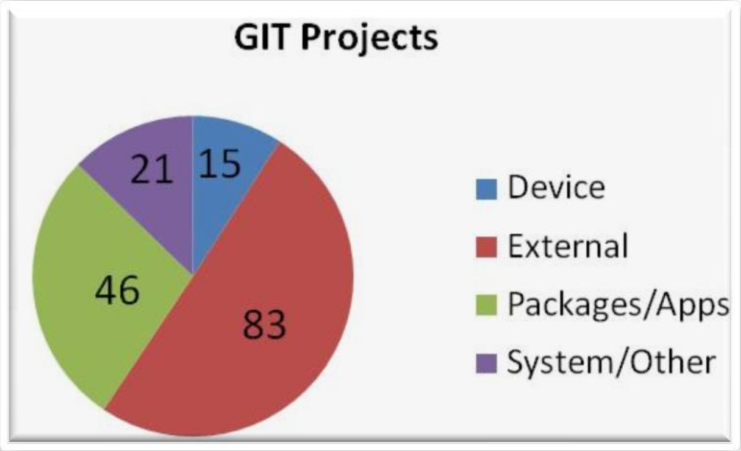
● Typical areas of vendor/developer innovation

Source: Google - //source.android.com/

What’s Inside Android?

Android

- 165 Projects
  - 83 are “External”
  - Does not include Kernel Mirror
- Total Size
  - Over 80,000 Files
  - Over 2GB total size
  - Does not include Kernel Mirror



A Look Inside Two Android Components: Bionic & Webkit

License types in: Bionic

BSD 2.0*
CMU License
Cryptix License
Free clause
FreeBSD
Historical free
INRIA OSL
Intel OSL
Internet Software Consortium
MIT
Public Domain
Python InfoSeek
X.Net License

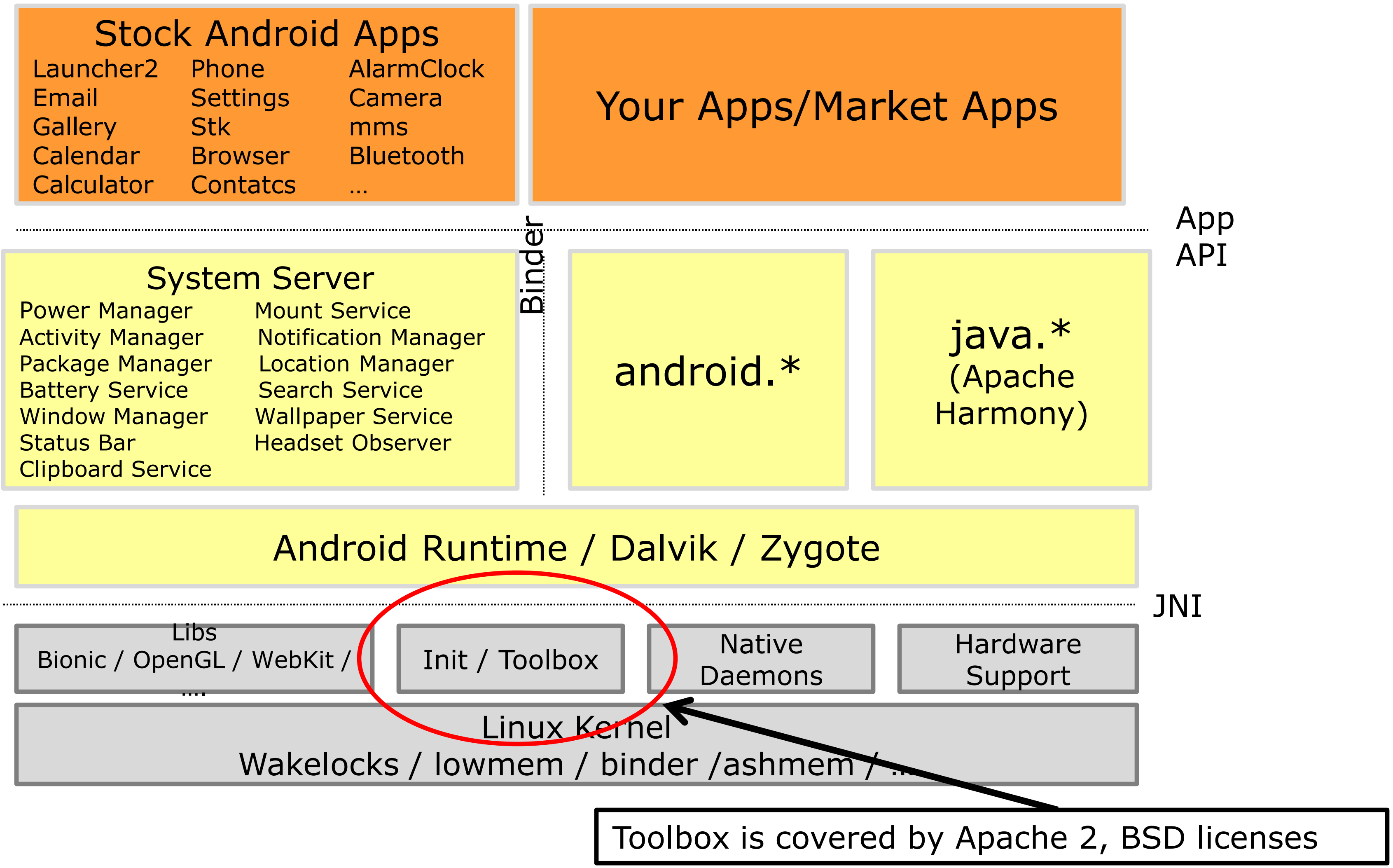
License types in: Webkit

BSD 2.0
David M. Gay License
GPL 2.0
ICU License
LGPL 2.1*
MIT License V2
MIT v2 with Ad Clause License
Mozilla Public License 1.1
PCRE License
Public Domain
SWIG License
The wxWindows Library License
zlib/libpng License

***Declared license**

Overall Architecture - Android

Architecture diagram from Karim Yaghmour of Opersys, Linux Foundation Android Builder's Summit, April 2011



Stock Android Apps

Launcher2, Phone, AlarmClock, Email, Settings, Camera, Gallery, Stk, mms, Calendar, Browser, Bluetooth, Calculator, Contatcs, …

Your Apps/Market Apps

App API

Binder

System Server

Power Manager, Mount Service, Activity Manager, Notification Manager, Package Manager, Location Manager, Battery Service, Search Service, Window Manager, Wallpaper Service, Status Bar, Headset Observer, Clipboard Service

android.*

java.* (Apache Harmony)

Android Runtime / Dalvik / Zygote

JNI

Libs Bionic / OpenGL / WebKit / ….

Init / Toolbox

Native Daemons

Hardware Support

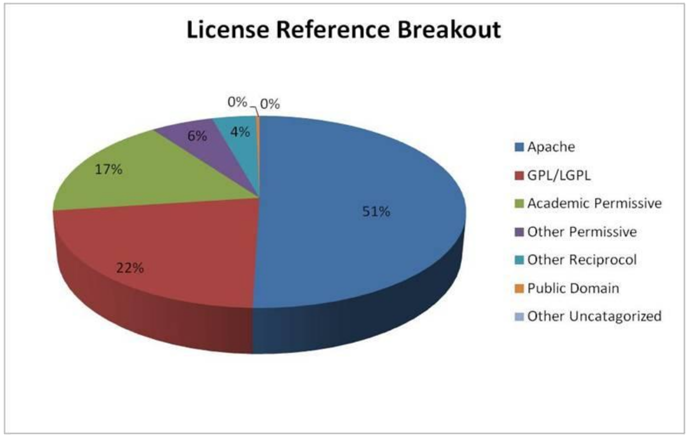
Linux Kernel

Wakelocks / lowmem / binder /ashmem / …

Toolbox is covered by Apache 2, BSD licenses

Android’s Composition

- Licenses
  - Declared license: Apache 2.0
  - Components reference 19 different licenses
  - External components
    - Linux, Webkit use reciprocal licenses (GPLv2, LGPL)
  - Other components: more than 30 of them use reciprocal licenses (GPL, LGPL, CPL, etc.)
    - e.g. dbus, grub, emma, e2fsprogs, bluez, Bison
  - Non-OSI approved licenses are used, including OpenSSL and Bzip2



blackduck

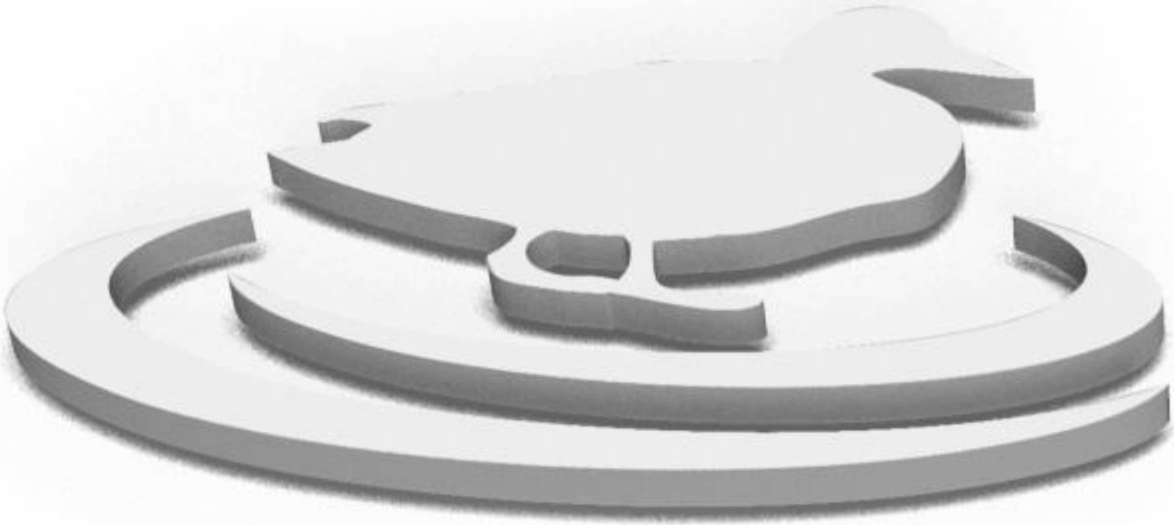
Know Your Code.

Obligations and Misperceptions

- No “small device” exceptions
- Must provide source for the specific device
- Compliance is required by every vendor that ships the platform
- There is no “downstream defense for upstream” violations

Agenda

- FOSS in Mobile Trends
- Device Manufacturers
- Application Developers
- Supply Chain Management
- Summary

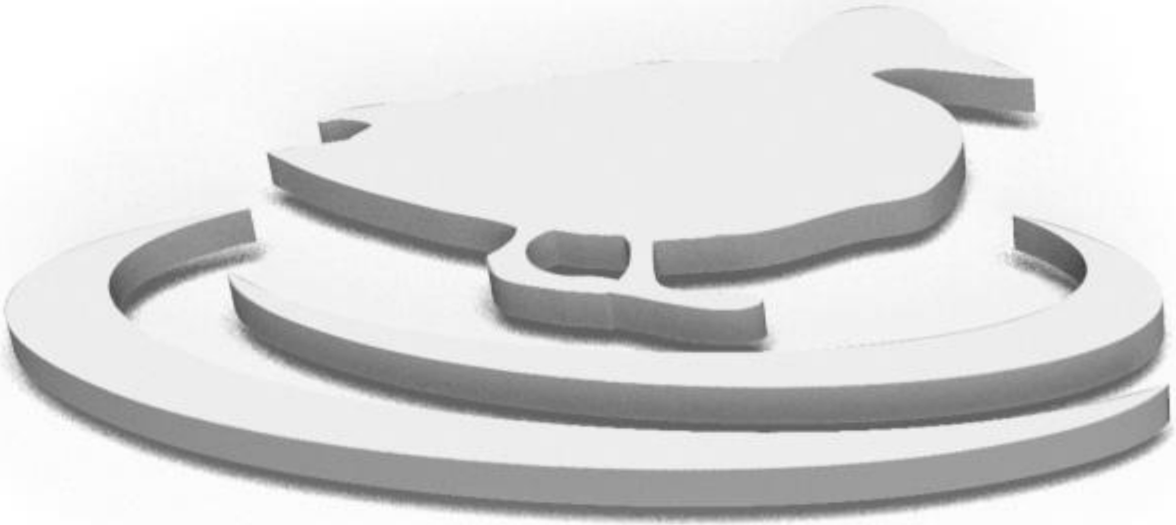


App Stores and FOSS Licenses

- GPL licensed app’s can not be distributed through the Apple iTunes Store (or any store that imposes restrictions)
  - Apple ToS (terms of service) require that all software be licensed for use on a single device only
  - “Copylefted software can’t be *un-freely* relicensed, so it can’t be transacted for under Apple’s current ToS” Eben Moglen, SFLC
  - Just like GPLv2, GPLv3 prohibits distributors from placing additional restrictions on the software through legal documents or similar means” Brett Smith, Free Software Foundation
- Android stores
  - “So far as we know…the Google Android market… do not place any limitation on how a market participant’s application is licensed that would inhibit distributing Android applications in the market under copyleft licensing.” Eben Moglen, SFLC
- Permissive licenses (e.g., Apache, MIT, BSD) appear to be compatible with app store ToS

Agenda

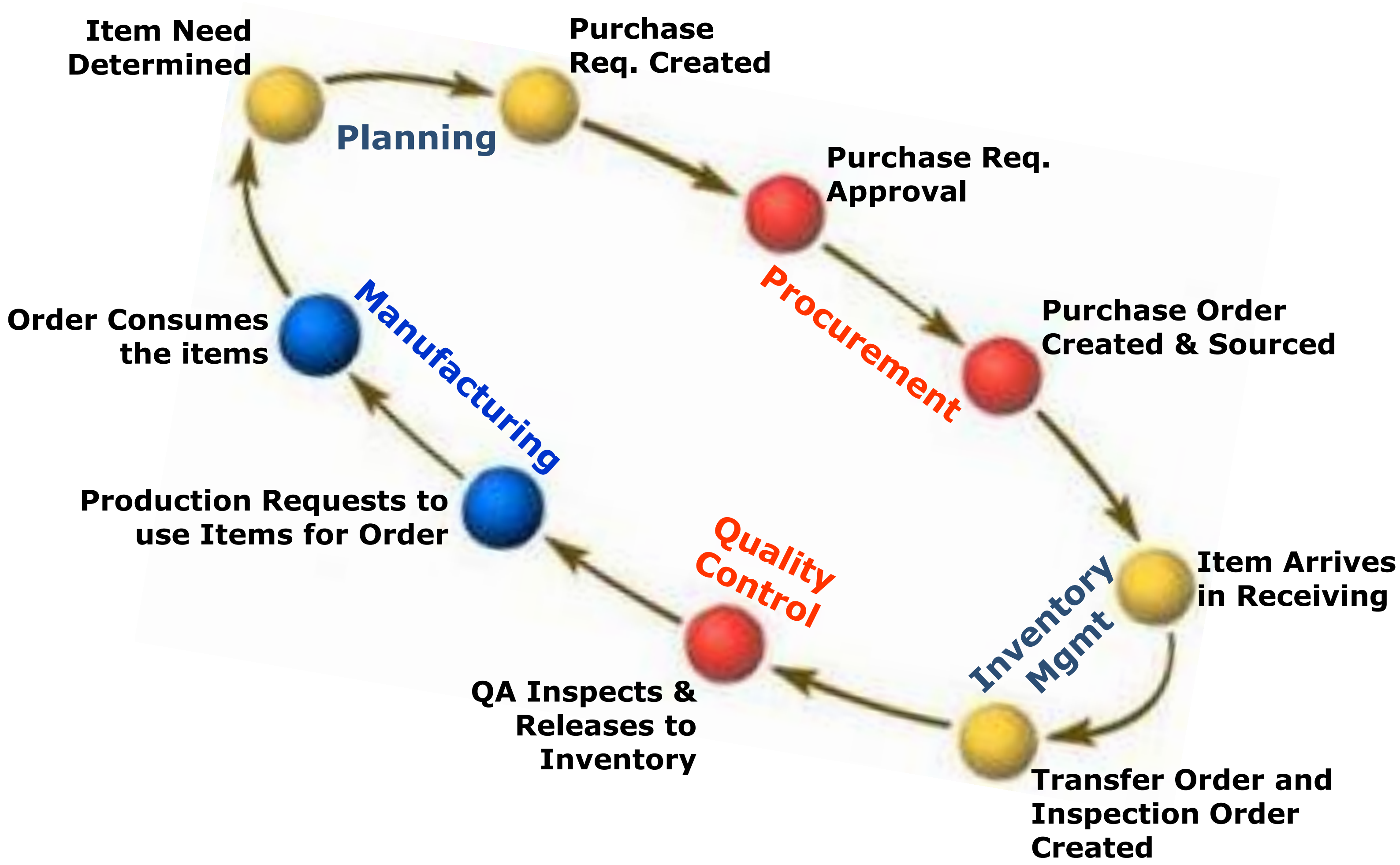
- FOSS in Mobile Trends
- Device Manufacturers
- Application Developers
- Supply Chain Management
- Summary



Software Supply Chain Management

- Open source is typically outside of normal commercial s/w procurement processes
- The Challenges
  - An increasingly diverse and distributed set of development resources
    - Internal teams
    - Commercial software vendors
    - Outsourcers
    - Open source communities
  - Little/no visibility into the origins of the software

Example Supply Chain Business Process


Item Need Determined
Purchase Req. Created
Planning
Purchase Req. Approval
Procurement
Purchase Order Created & Sourced
Item Arrives in Receiving
Inventory Mgmt
Transfer Order and Inspection Order Created
Quality Control
QA Inspects & Releases to Inventory
Production Requests to use Items for Order
Manufacturing
Order Consumes the items

blackduck

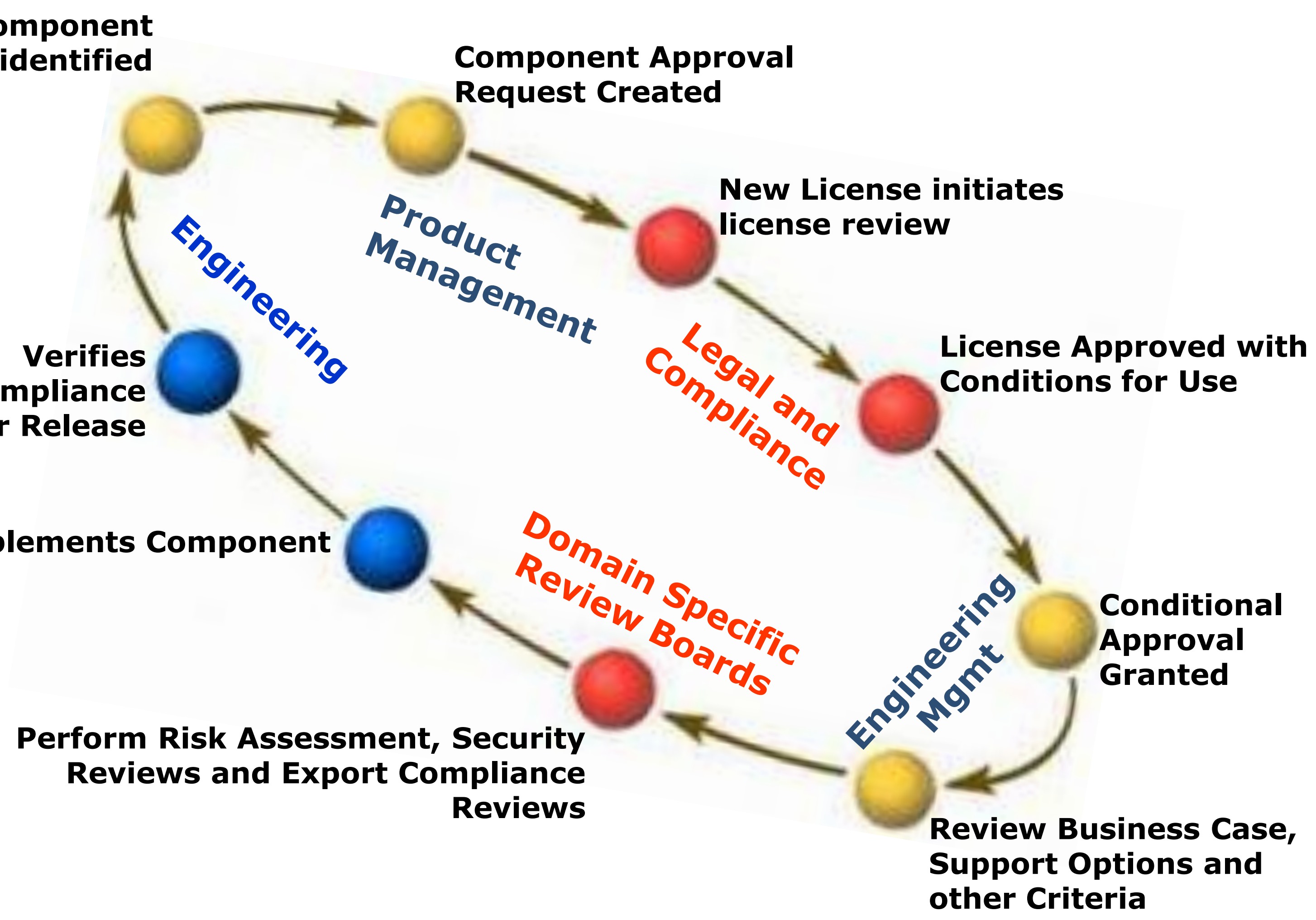
Know Your Code.

Supply Chain Comparison: HW vs SW

- HW Supply Chain Techniques
  - ERP systems brought together different users and processes
  - Workflow automates task creation
    - Notifications
    - Process Monitoring
  - Central repositories of data
  - Business Process Integration is the key
- Technology companies have software supply chains
- Software products have bill of materials (BOM’s)
- Similar roles and events: HW = SW

– Materials Planner	=	Product Management
– Purchase Req’s	=	Component Approval Request
– Warehouse	=	Source Code Management
– Quality Assurance	=	Numerous types of code analysis
– Procurement Approvals	=	Legal & Compliance Approvals
– Shop Floor Production	=	Engineering

Example Software Development Business Process



Need for a component is identified

Component Approval Request Created

New License initiates license review

License Approved with Conditions for Use

Conditional Approval Granted

Review Business Case, Support Options and other Criteria

Perform Risk Assessment, Security Reviews and Export Compliance Reviews

Implements Component

Verifies Compliance for Release

The Golden Rule for Proper Software Supply Chain Management

Treat the management of open source software as an integrated, cross functional **business process**, and not simply as a development process.

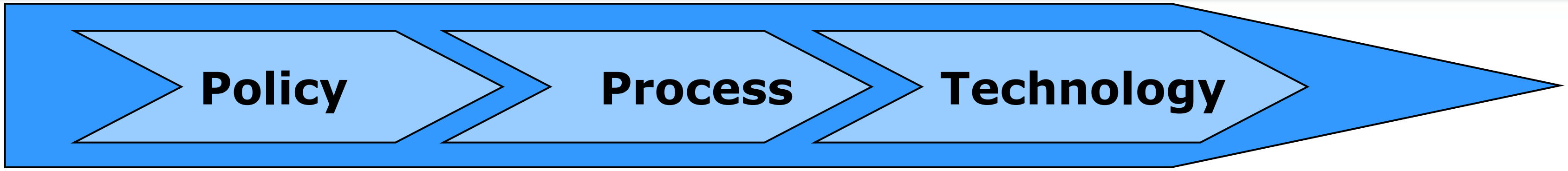
Supply Chain Program Elements

1. Published Policy
   - Created via Cross Functional Team
   - Organization is educated on the policy
2. Open Source Process Owner
   - Keeps the wheels running
   - Grant certain types of approvals
3. Approval Processes
   - Component Review & Approval
   - Sensitive to Use: internal/external/products
   - License Review & Approval
   - Release Plan Review & Approval
4. Monitoring & Tracking Process
   - Component Verification
   - Security Notifications
   - Component Upgrade Notifications
   - Application to contractors/outsource vendors
5. Obligation Verification Process
   - Ensure using approved components… and…
   - Meeting the license and business obligations
   - Current reporting for responsive due diligence request

Software Package Data Exchange™ (SPDX™)

- Working group of FOSSBazaar (governance best practices group under Linux Foundation)
- Charter:
  - ➢ Create data exchange standards to enable license and component information sharing (metadata)
- Participation from over 16 organizations including software, systems and tool vendors, consultants and foundations

Best Practices for Managing Android



- Adopt and enforce an open source and third-party code policy
- Identify and track all external code that is used
- Automate validation at the point of acquisition and development
- Automate monitoring and tracking of Android components
- Control the use of components and promote standardization
- Use automation tools to produce complete Bills of Material and reports for supply chain partners

Summary

- Android has revolutionized the mobile and device landscape
- Like many FOSS projects, Android has complexity inside
- Effective management and control requires training, tools, processes and *standards*
- “SPDX is a crucial building block in an industry-wide system of automated license compliance administration” Eben Moglen

Information Resources

- Webinar-based education:
  - //www.blackducksoftware.com/webinars/legal/
  - Introduction to Open Source Licenses
  - Understanding the Top 10 Open Source Licenses
  - Unraveling the Complexities of the GPL
- Black Duck Android white paper & webinar
  - //www.blackducksoftware.com/android
  - //www.blackducksoftware.com/webinars/legal/android.html

Thank You

Peter Vescuso

Black Duck Software

pvescuso@blackducksoftware.com