Openness in the supply chain

10 May 2017

Andrew Katz
Partner, Moorcrofts LLP, Steering Group Member, LIM-IT Project,
Visiting Researcher, University of Skövde
andrew.katz@moorcrofts.com
www.moorcrofts.com
+44 1628 470003

Open Source – Why?

- Low-overhead collaboration mechanism
  - No formal collaboration agreement
  - No need for intra-group arrangements
  - No formal commitment
  - Drop in/drop out
  - Minimise competition law issues
  - No reinventing the wheel

Open Source – Why?

- Attractive to developers – staff retention
- May generate informal standards
- The power of upstreaming
  - Outsource support
  - Keep consistency with main project
- Concentrate on perfecting your differentiator

The Genivi Experience

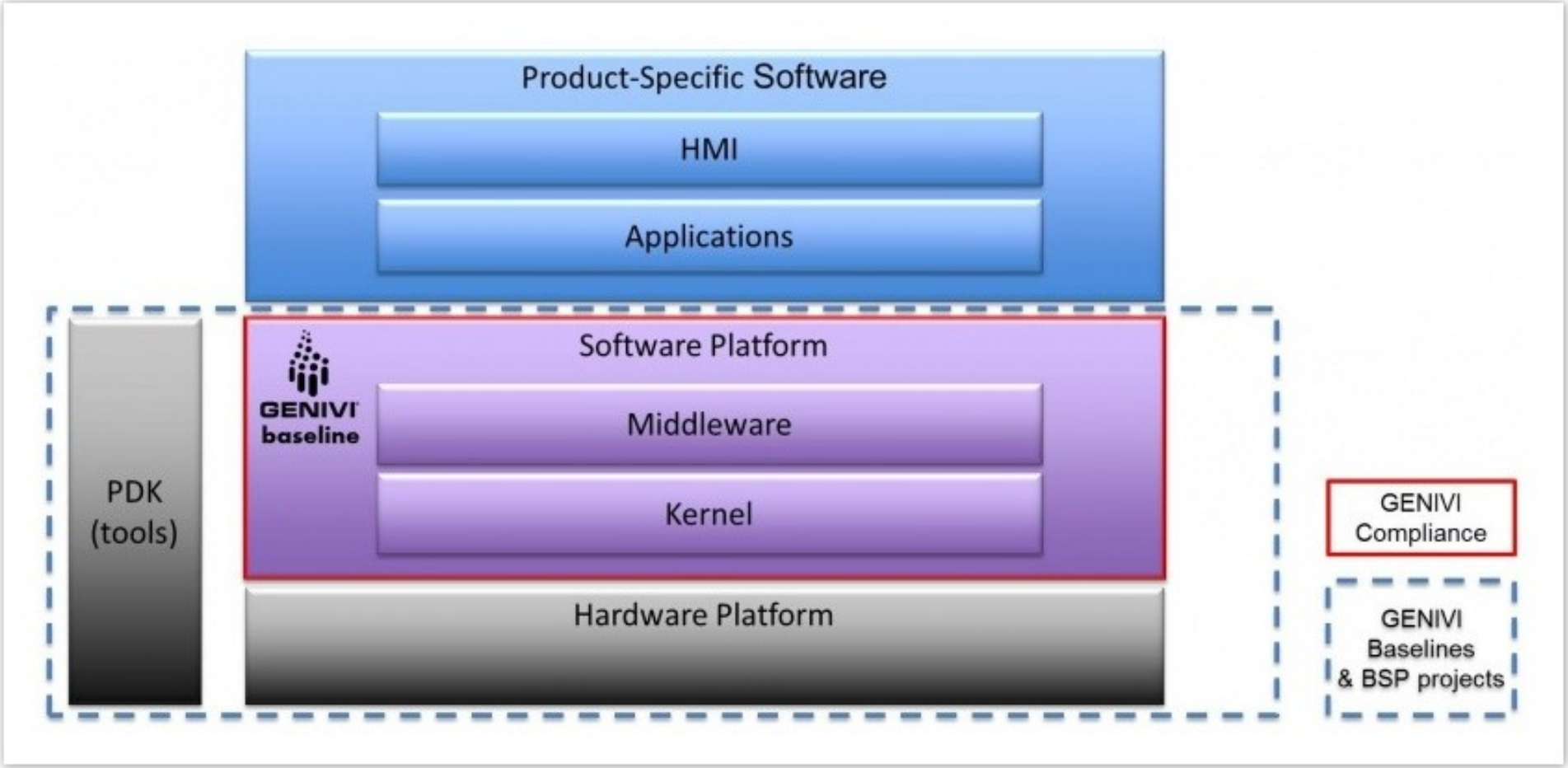
Genivi

- In-Vehicle-Infotainment:
  - Music
  - News
  - Multimedia
  - Navigation
  - Location (“where did I park my car?”)
  - Telephony
  - Internet Services
  - Etc.

Genivi

- A reusable, open source IVI platform will
  - Speed time to market
  - Accelerate prototyping to deliver better consumer functionality
  - Dramatically reduce development costs
  - Allow for redeployment on a variety of hardware platforms
  - Provide transparency and mitigate vendor lock-in

How?


Product-Specific Software
HMI
Applications
Software Platform
GENIVI baseline
Middleware
Kernel
Hardware Platform
PDK (tools)
GENIVI Compliance
GENIVI Baselines & BSP projects

But NOT the User Interface (HCI)

Image Credit Flickr – Jaguar MENA

Typical Key Differentiators

- Now more valuable than later
  - Bloomberg
- We have better expertise to build on the content than others
  - Red Hat
- We have goodwill in X
  - BMW’s engines as opposed to Mini’s
- We’ve got better icing
  - Apple
- Our system is the de facto standard
  - Microsoft
- We can complete the whole
  - Amazon

Do our competitors use open source?

- 44% of all code created in the world is OSS and increasing
- 80% of newly deployed code is open source
- 31% of OSX is (was!) OSS, 75% of Android.
- Stats demonstrate OSS more innovative than proprietary
- 36% lower defects in OSS than comparable proprietary code
- 80% of software is non-differentiating
- Revenue/employee in FLOSS firms: 221% of non-FLOSS
- http://transfersummit.com/sites/default/files/materials/rgardler/ts11daffara-notes.pdf - thanks Carlo!
- http://www.openforumacademy.org/library/ofa-fellows-reference-library/ofe-fellows-reference-library/Hosted%20Files/first-conference-proceedingsA4.pdf

Key questions (1)

- How are you deploying?
  - Internal
  - Internal/external (SaaS)
  - External (software distribution)
  - External (embedded appliance/distribution)
- Is some/all of the product a differentiator?
- Can you build a community around it (and do you want to?)

Key Questions (2)

- What are your goals?
  - Maximise use?
  - Maximise freedom?
  - Sell exceptions?
  - Monetise surrounding services?
  - Create a community?
  - Sell associated hardware?

Key Questions (3)

- Maximise use?
  - Apache
- Maximise freedom?
  - GPL/AGPL
- Sell exceptions?
  - GPL/AGPL
- Monetise surrounding services?
  - GPL?
- Create a community?
  - Depends…
- Sell associated hardware
  - GPLv3

What are the risks?

- Giving away differentiators
- Reputational risk
- Legal risks
  - Copyright/database right infringement
  - Patent infringement
  - Failure of performance/professional negligence

Content/Software Distributors

- If distributing, must consider licence compatibility
- Governance
  - Document source of code
  - Have a compliance policy
    - Which licences?
    - Which sources of code?
    - How to document?
    - How to deal with exceptions?
    - How to test compliance?

Copyright infringement

- Trusted sources
- Licence compatibility
- [You can’t easily blend components which are under different copyleft licences, and distribute them]
  - Easy for SaaS provision
  - More difficult for distribution

Open Chain

- Hosted by Linux Foundation
- Version 1.1 Launched on 27 April 2017
- Self certification for organisations
- Initial adopters
  - Siemens
  - Qualcomm
  - Pelagicore
  - Wind River

Open Chain

- Aims to be an ‘ISO for FOSS process transparency’
- Covers organisations, not software
- Not prescriptive. Covers practice and procedure for:
  - Documentation
  - Licence selection
  - Training and awareness
- Simplifies and encourages trust in the supply chain.

Open Chain (3)

- Specification:
  - https://www.openchainproject.org/spec
- Online Self-Certification:
- https://www.openchainproject.org/conformance
- Curriculum:
  - https://www.openchainproject.org/curriculum
- FAQ:
- https://www.openchainproject.org/faq

Engaging with Communities

- Why?
  - Upstreaming
  - Access to expertise/support
  - Develop deep understanding of software
  - Influence future development roadmap
  - Developer kudos
  - Recruitment source

Establishing your own community

- Developer community or user community?
- Balance of power – ceding control to third parties (possibly competitors)
- E.g Eclipse
  - Balances corporates with developers
  - Acknowledges need for sponsorship AND developer involvement

Different types of community

- Apache-style
  - More relaxed about licensing
- GPL style
  - May be suspicious of corporate involvement
- Learn the terminology and methods of engagement
- CLAs

Compliance with Standards

- Open standards
- Do not embrace and extend
- Long term maintenance and support
  - Do not assume customers will always upgrade
  - Adherence to standards facilitates migration between versions
  - Avoids lock-in
- Public sector may be subject to procurement rules (e.g. EIFv1)

Key Points

- Always think ‘key differentiators’
- SaaS is always easier to manage than on-premise
- Understand communities and the benefits of engagement
- Upstreaming, standards and long-term benefits.