how to protect your intellectual property when using open source

Shawn Kwon <ksoonson@samsung.com>

Samsung Electronics

Disclaimer

- I am not a lawyer, opinions in this session are all personal and does not represent my employer

Intellectual Property

Protect my copyright / patent

Do not infringe other's copyright / patent

- Copyright
  - Expression : source code
- Patent Right
  - Method: idea and algorithm
- Trademark
- Trade Secret
- IP issue should be handled two ways: copyright and patent

Open Source Overview

Increase Efficiency

- Shorter Development Time
- Royalty Free
- High-Quality Component
- Reduce maintenance cost

Expand Platform

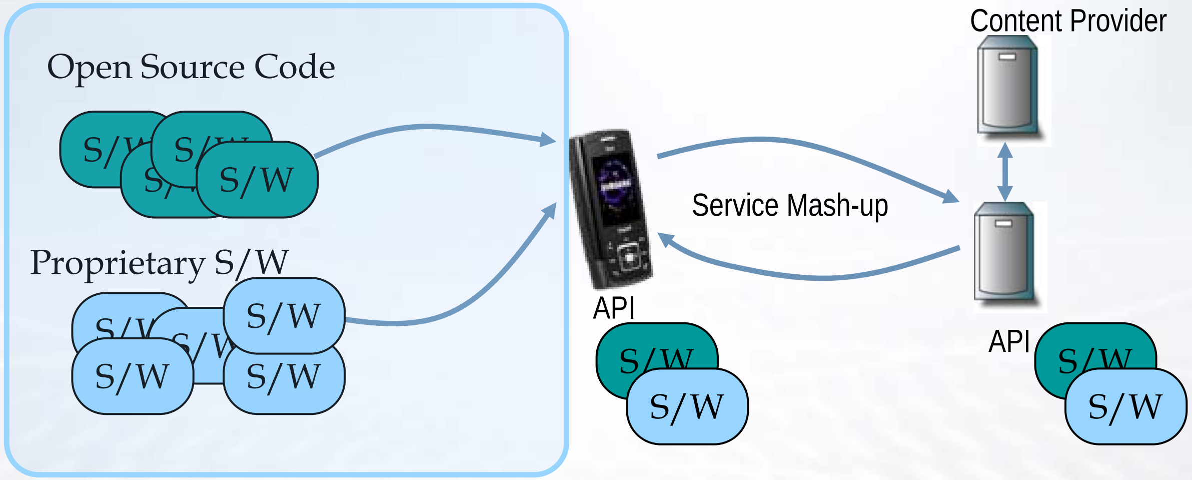
- Encourage SW/Service
- Encourage mobile device purchase (Smartphone, PMP, PDA)
- Tech/Platform Marketing

Remove Obstacle
(License Issues, Recommend)

Avoid Vendor Lock-In

Expand Open Source
(Industry Assoc.,
Community sponsorship)

Best Practice
(Platform dev., participate community)



Open Source Deployment

We are bound to the license term of the SW including Open Source SW

- Avoid infringing someone else' IP
  - Follow the license term
    - Each Open Source license has different terms
    - If you do not follow correctly, then you may be in trouble

- Protect my IP
  - If IP is implemented in Open Source code, then that IP can be virtually Royalty-Free
    - Not every open source does not protect IP
    - For example, extra care should be taken when using GPL

Open Source License Violation

Brand Image

Stop sales

Much effort is needed

- Once it goes public, it will seriously damage you
- Even if it is not public, it still takes much longer time / effort to resolve: it is best to work on it before shipping
- FSF compliance lab, gpl-violaions.org, Freedom Task Force…

Open Source License Obligations

- Various Open Source Licenses
  - Over 60 (http://www.opensource.org/licenses/)
  - Each License has different terms

- Case-by-case
  - Release source code of the open source SW
    - GPL/LGPL/MPL…
  - Provide the license text to the end user
    - GPL/LGPL/OpenSSL
  - Patent issue
    - GPL/LGPL/Apache2.0/MPL/Apache1.0: if you implement patented algorithm into open source code…

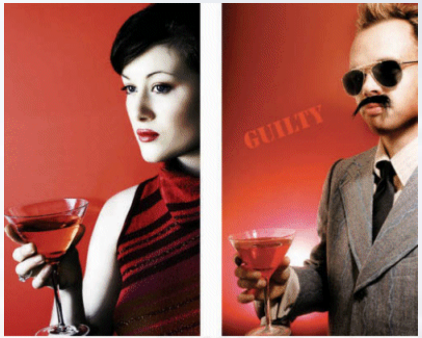
- Common
  - Trademark
    - All: cannot use the name of existing open source SW
  - Copyright Assignment
    - All: never delete any copyright-related comment at the source code

Protecting Copyright (1/2)

Your code should not be the derivative work

- Definition of derivative work
  - Work based on that program
    - If you modify or copy from original work, then you are creating derivative work

- No way to determine code copying

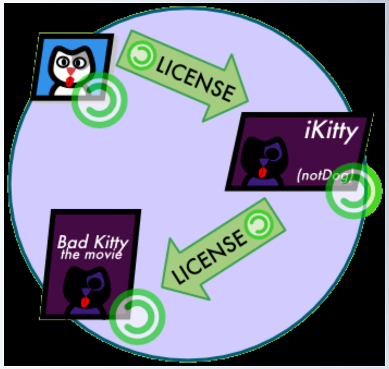


This example is likely a derivative

Protecting Copyright (2/2)

You should check before because the definition of derivative work is all different (especially GPL)

- Reciprocal license: derivative work should be open source
  - Non-Recprocal license: BSD License…etc
  - GPL, MPL, EPL…etc
- Derivative work is defined differently among licenses
  - GPL, LGPL: how you link the programs
  - MPL, EPL: adding more code or modifying original code creates derivative works
- Author (Contributer)
  - Example: Even among open source developers, there are much debate on GPLv3…etc



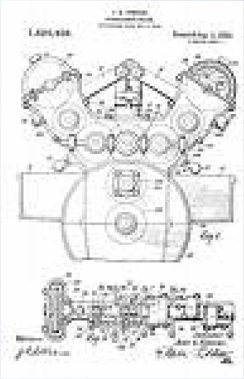
Protecting Idea / Algorithms

Check before when applying patent,
or implementing patented technology into open source

- Most Open Source license does not talk about patent right clearly except a few recent ones like MPL

- If you implement your patent into open source, you should check the license before

- Example
  - GPL: Implicit. Cannot distribute GPL'd SW if you do not have full patent right
  - Apache: If you sue the developer for patent infringement which is implemented in that SW, you cannot use that SW
  - MPL, APSL, NOKOS: If you sue the developer for any kind of patent infringement, you cannot use that SW



S/W Linking and GPL/LGPL

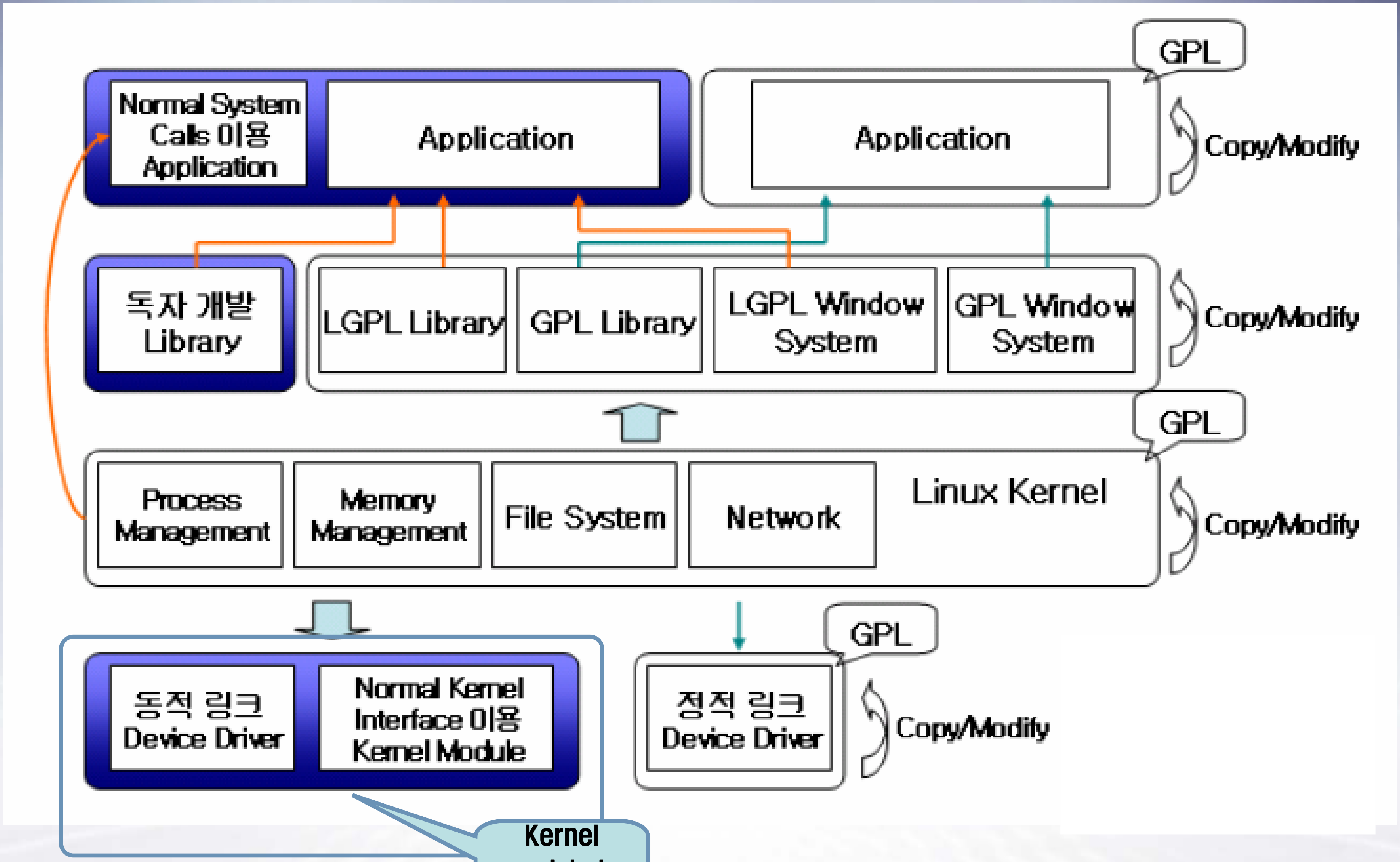
- GPL Library vs Application Linking
  - The linked Application should be GPL for Static or Dynamic linking both

  → Should release source code

- LGPL Library vs Application Linking
  - The linked Application need not be provided source code whether it is Static or Dynamic linking
  - For static linking, object code of the application should be provided

=> Recommend LGPL Library

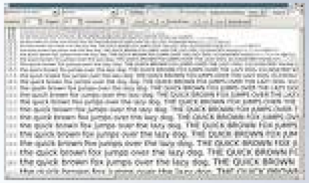
GPL/LGPL Summary



Kernel module is GPL?.

Example: Userspace Apps

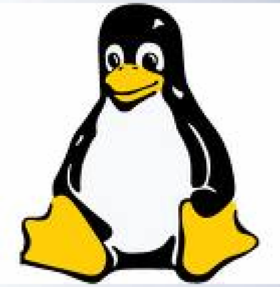
- SQLite
  - Public domain, can do virtually anything
- FreeType
  - Dual License(Freetype License or GPL): FreeType License is preferred: Attribution Notice
  - Patent Issue
    - Bytecode Interpreter Enable: pay royalty to Apple first



- Glib
  - LGPL
  - You should release the source code of Glib regardless of modifying it or not
- DPKG
  - GPL
  - Same above about code release

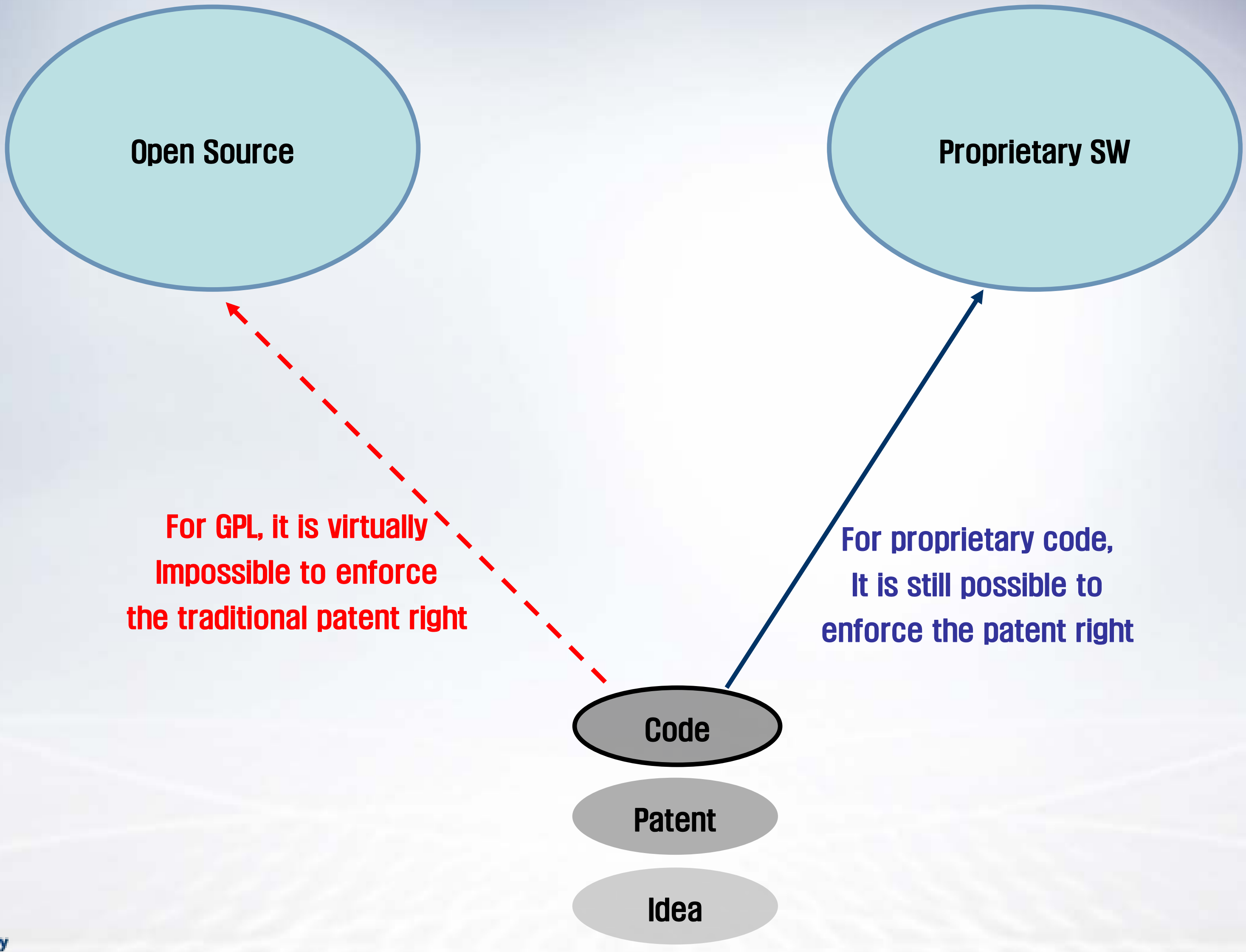
Example 2: Linux Kernel Module

- Kernel Module is GPL?
  - It is debatable
  - Should not modify the kernel interface
    - Ex: EXPORT_SYMBOL() → EXPORT_SYMBOL_GPL()
    - That module should work with kernel interface that is downloaded from http://kernel.org
  - Who determines the kernel licensing issue?
    - Copyright holders: Linus Torvalds and ETC
    - FSF, Richard Stallman??



Example 3: Open Source and Patent

- You should check the license term if you implement patent



Open Source

Proprietary SW

For GPL, it is virtually
Impossible to enforce
the traditional patent right

For proprietary code,
It is still possible to
enforce the patent right

Code

Patent

Idea