

# Open Source: A New Business Model baed on Licenses

Xiao Zhang

Law School/Intellectual Property School  
Peking University

# Outline

- Monetizing Software: An Evolutionary History
- Modern Expression of Old Business Model
- Licenses: the skeletons of opensource monetization
- Legal Risks in moneitizing opensource software
- Q&A

# Monetizing Software: An Evolutionary History

- 1970's: software was not recognized as a “Property”
  - Value was mixed with hardware
  - The labor force used in software development was counted as a cost to the tangible product
  - Software Developer: “Labor Provider” , not “Property Creator”
  - Law: mostly a kind of “know-how”
- Late 1970's~Early 1980's: legislations changed the story
  - Many countries began to recognize it's copyrightability
  - e.g, CONTU report
  - Legally, it became a “Property”

# Monetizing Software: An Evolutionary History

- 1980's: software was recognized as a “Property”
  - Value separated from hardware
  - An independent business
  - Software company: a new sort of “Property Lease Company”
    - What: Software
    - How: Granting Licenses
    - Specialty: multiple leasees to same property
- Open source movement kick-off
  - 1983, GNU program established
  - 1985, Free Software Foundation Established
  - 1989, GPL v1 Published

# Monetizing Software: An Evolutionary History

- 1990's: software industry boomed
  - Proprietary software licensing became most profitable business in history
  - Open Source movement continued to grow
    - 1991, GPL v2 published; Linux kernel published;
    - 1995, Apache HTTP server launched
    - 1997, Eric Raymond, The Cathedral and the Bazaar
    - 1998, OSI established
  - Opensource software companies emerged
  - Proprietary software company: look opensource as an enemy
    - “Linux is a cancer”--Steve Ballmer, 2001

# Monetizing Software: An Evolutionary History

- After 2000's: “We love open source!”
  - Proprietary software companies embraced open source
    - 2006, Microsoft opened Microsoft Linux and open source labs
  - Internet giants use a lot of open source software
    - On the servers
    - Linux based mobile operating systems
- “We love open source!”--Steve Ballmer, 2010
  - Why not!
  - As long as you understand how to monetize open source software.
  - All in all, WHO DOESN'T LOVE MONEY?

# Outline

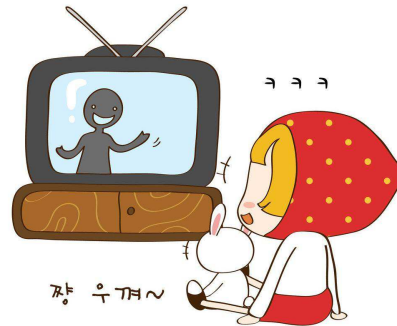
- Monetizing Software: An Evolutionary History
- Modern Expression of Old Business Model
- Licenses: the skeletons of opensource monetization
- Legal Risks in moneitizing opensource software
- Q&A

# Modern Expression of Old Business Model

- 1920's



- 1940's



- 2000's



- Business Model:
  - Two-sided Market
  - Providing one thing for free, but charging from another thing
- Open Source:
  - Technical service
  - Additional warranty
  - Open Souce license+commercial license
  - Traditional software license(allowed by most open source licenses)
  - Charge “copy distribution” fee(under GPL)



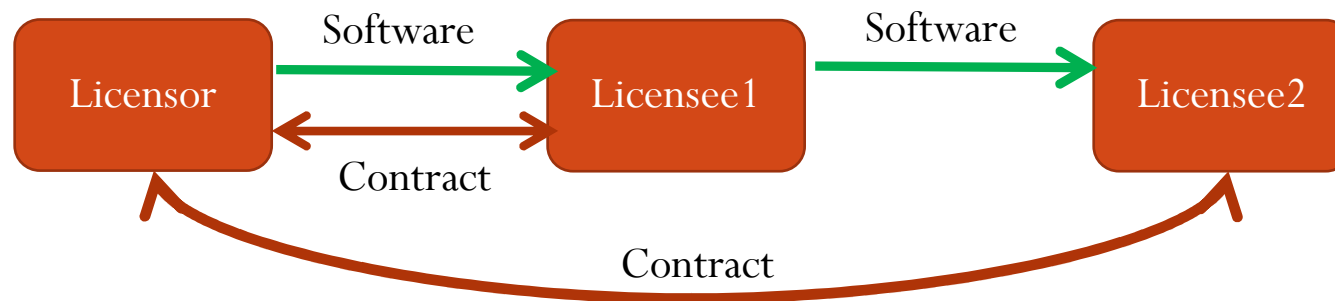
# Outline

- Monetizing Software: An Evolutionary History
- Modern Expression of Old Business Model
- Licenses: the skeletons of opensource monetization
- Legal Risks in moneitizing opensource software
- Q&A

# Licenses: the skeletons of opensource monetization

## ----Basics

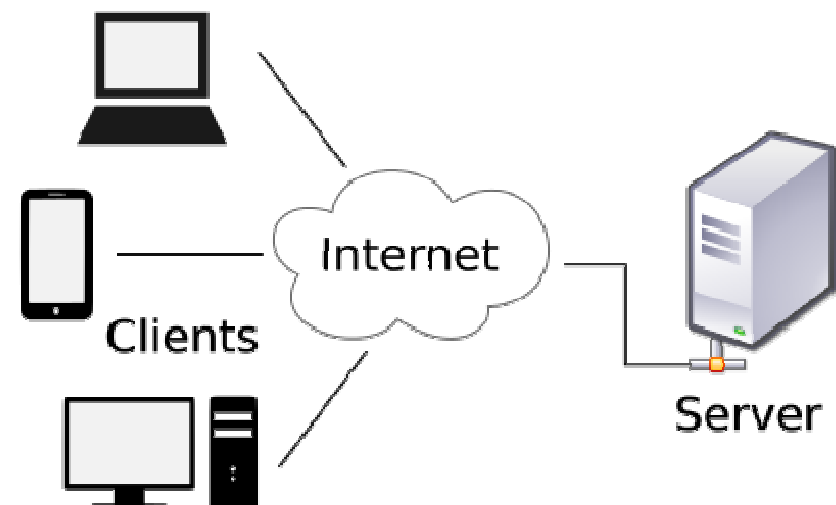
- the Nature of Open Source:
  - An intellectual property based revolution in licensing
  - Copyright based copyleft
  - What if violating licenses? --Infeasible without intellectual property protection
- Licenses Definition
  - A contract btw Licensors and Licensees
  - “Conductive” contract



# Licenses: the skeletons of opensource monetization

## ----Basics

- Open Source Licenses classification:
  - Contagious: requires licensees to use the license to modifications at redistribution, GPL-style
  - Non-contagious: Not required, Apache, BSD-style
- Impact on open source monetization of different licenses
  - Internal copying and use-- No impact(except for AGPL)
  - Redistribution--Huge impact



# Licenses: the skeletons of opensource monetization

## ----Basics

- What if violation of the license?
  - Intellectual property licenses terminates IMMEDIATELY & AUTOMATICALLY
  - Any use afterwards constitutes infringement
  - Plaintiff: developer or his agent
    - FSF, Oracle etc.
- “Redistribution and use in source and binary forms, with or without modification, are permitted **provided that** the following conditions are met:”

# Licenses: the skeletons of opensource monetization

## ---Licenses Introduction: GPL2.0

- Some Important Concept:
  - **Program**: the codes obtained
  - **work based on the Program** (modified work)

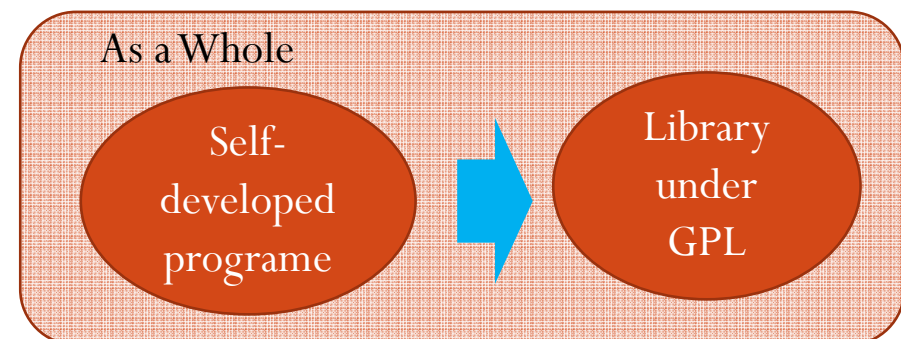
### Typical Situations

- **Modifying Program**: Adding some codes, deleting some codes, optimizing some algorithm
- Taking some codes from the **Program**, and incorporating into the self-developed codes.
- Contagious: When Redistribution
  - Source codes should be available
  - Should be redistributed under GPL
  - Burdensome requirements for redistributors

# Licenses: the skeletons of opensource monetization

## --Licenses Introduction: GPL2.0

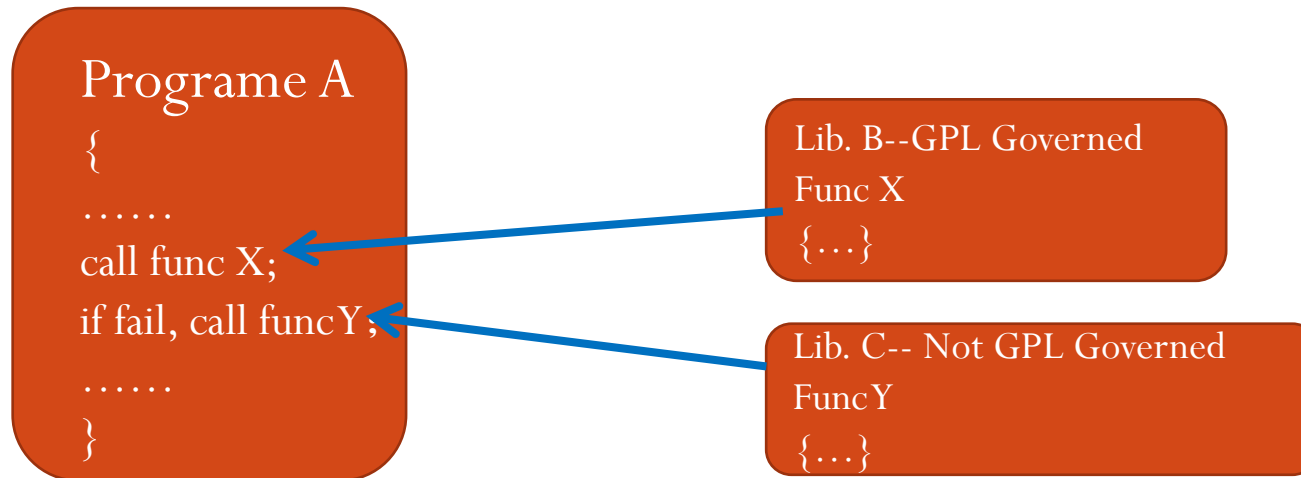
- A special clause for **modified work**:
  - **modified work** as a whole is governed by GPL
  - Exception: “If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves”.
  - Exception to exception: “But when you distribute the same sections as part of a whole which is a work based on the Program”
- Conclusion:
  - Premise: Dynamic Call
  - Key:Indep. Operation
  - Affecting: Redist. Method.



# Licenses: the skeletons of opensource monetization

## --Licenses Introduction: GPL2.0

- An example:



- Safer redistribution method:
  - Programme A+Lib C together, Lib B independently
  - Programme A, Lib C and Lib B separately and independently

# Licenses: the skeletons of opensource monetization


## --Licenses Introduction: GPL2.0

- Fee Model
  - No difference between **Programe** and **modified work**
  - Distribution Fee-- Yes ; Royalty--No



# Licenses: the skeletons of opensource monetization

## ---Licenses Introduction: LGPL2.1

- Focusing on Lib documents to solve the operation between a programme and lib.
- Key concepts:
  - **Library**
  - **Work based on the Library** 
    - Same with GPL in redistribution
    - If modified, independent of application programmes
  - **Work that uses the Library**
    - The calling programme
    - It links with Library to create a special “**work based on the Library**” (“**Linked Work**”)
  - **Combined Library**: combine the facilities in the Library with other facilities. It's a special “**work based on the Library**”

# Licenses: the skeletons of opensource monetization

## --Licenses Introduction: LGPL2.1

- Distributing **Work that uses the Library**
  - If it complies with *header files*, **a work based on the Library might**(object codes) be created--depending on the copyrightability of header files
  - If the object codes only includes following of the **Library**:
    - numerical parameters
    - data structure layouts and accessors
    - small macros
    - small inline functions (ten lines or less in length)
  - Then exempted from “**work based on the Library**”, regardless of how the law defines it

# Licenses: the skeletons of opensource monetization

## ---Licenses Introduction: LGPL2.1

- Distributing **Linked Work**
  - Freedom:
    - use your own license
    - bottom line: allow the user to modify and do reverse engineering
  - Other requirements:
    - Inform the user the **Library** is used and provide LGPL
    - Codes:
      - Provide source codes of Library and object codes of the **Work that uses the Library**
      - Dynamic Call v. Static Call: does not matter

# Licenses: the skeletons of opensource monetization

## ---Licenses Introduction: LGPL2.1

- Distributing Combined Library
  - Accompany the Combined Library with a copy of the facilities in the Library separately
  - Notify the user that the Combined Library includes facilities in the Library and how to find them

# Licenses: the skeletons of opensource monetization

## --Licenses Introduction: Apache 2.0

- Distributing **Work**(unmodified programme)
  - under Apache 2.0
  - may distribute object codes only
- Distributing **Derivative Work**(modified programme)
  - Notify that modifications are made
  - Retain original statement of rights
  - may distribute under other terms
    - A4 last paragraph: You may ...provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions
    - A4.1: You must give any other recipients of the ...Derivative Works a copy of this License

# Licenses: the skeletons of opensource monetization

## --Licenses Introduction: BSD 2.0

- Distributing work/modified work: Either in source form or in binary form
  - retain copyright notice
  - retain “condition list”
  - retain disclaimer
- What is “condition list”?



# Licenses: the skeletons of opensource monetization

## —Case Introduction: Andorid



# Licenses: the skeletons of opensource monetization

## --Case Introduction: Android

- Linux Kernel

NOTE! This copyright does *\*not\** cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does *\*not\** fall under the heading of "derived work". Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the linux kernel) is copyrighted by me and others who actually wrote it.

Linus Torvalds

-----

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
675 Mass Ave, Cambridge, MA 02139, USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

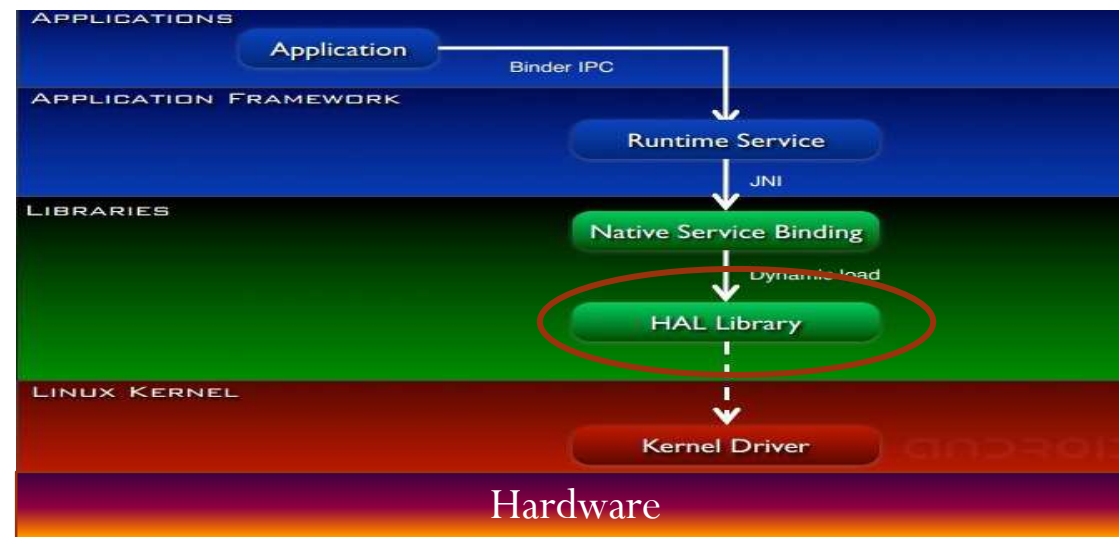
The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public



# Licenses: the skeletons of opensource monetization

## --Android

- Glibc: replace Bionic libc(BSD) for glibc ( GPL )
- Protection of Hardware driver
  - Drivers were separated from Linux Kernel
  - The drivers in Kernel are just used to transfer data and commands



# Licenses: the skeletons of opensource monetization

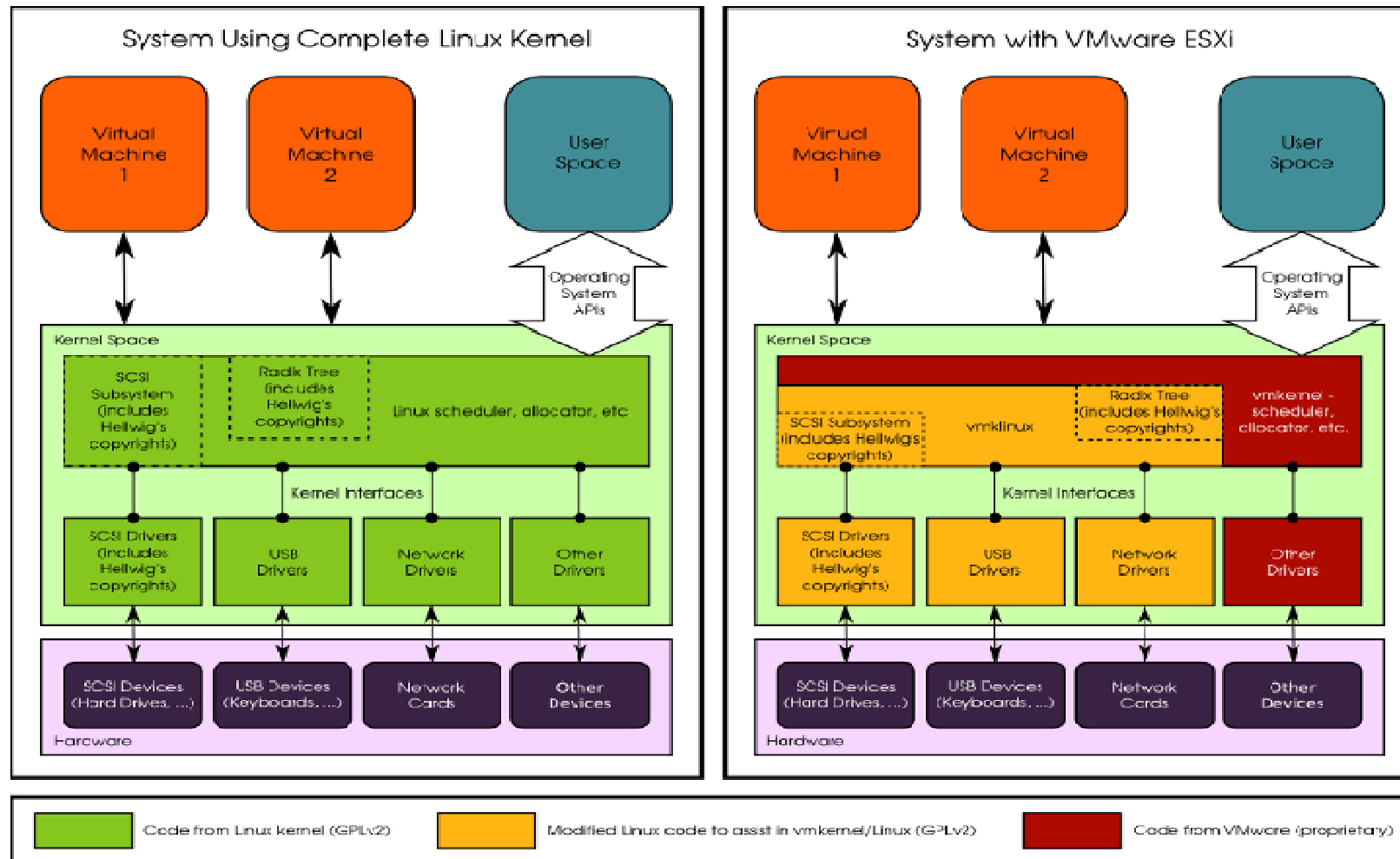
## --Case Introduction: JAVA

- Java API--Java programmes developed in advance
  - Declaring code: “public static int max(int x, int y)”
  - Implementing code: programmes executing the function
  - License Model:
    - General Public License
    - Specification License
    - Commercial License
- Oracle vs. Google
  - 37 API involved
  - “structure, sequence, and organization”
  - 2012: Federal District Court-- not copyrightable(by jury)
  - 2014:CAFC-remand
  - 2016:Federal District Court--not copyrightable(by jury)
  - Oracle will appeal again

# Licenses: the skeletons of opensource monetization

## ---Case Introduction: VMWARE

- Violation of GPL



# Outline

- Monetizing Software: An Evolutionary History
- Modern Expression of Old Business Model
- Licenses: the skeletons of opensource monetization
- Legal Risks in moneitizing opensource software
- Q&A

## Legal Risks in monetizing open source software

- Risk1: Violation of licenses-- controllable by lawyers
- Risk2: Defect in software--controllable by lawyers and technicians
- Risk3: Intellectual Property Infringement
  - Copyright/Patent infringement
  - Almost Uncontrollable--the codes obtained from internet may have been “polluted”--the contribution itself is infringing

# Outline

- Monetizing Software: An Evolutionary History
- Modern Expression of Old Business Model
- Licenses: the skeletons of opensource monetization
- Legal Risks in moneitizing opensource software
- Q&A

## Q&A

- Thank you!