Webinar Series

Software Licensing Simplified

Barry Datlof
U.S. Army
Medical Research and Materiel Command

Kathleen McDonald
DOE
Los Alamos National Laboratory

Aaron Sauers, CLP
DOE
Fermi National Accelerator Laboratory

Today's Road Map

- Is it better to protect software through patents, copyrights, or "black box"?
- How can things you're not traditionally allowed to patent give you value?
- What can you do at a GOCO that you can't at a GOGO, and vice versa?
- Software distribution models, their benefits, and their concerns
  - Commercial (proprietary)
  - Non-commercial
  - Open source
  - SaaS (Software as a Service)

There Are Many Ways to Make Money

- Patent
- Copyright
- Trademark

} Classic Technology Transfer

- Code and code updates
- Data/content

} Classic Software Licensing

- Black box website
- App store
- Web method/API access
- Community of Practice

} Modern Software Licensing

- Help desk, training
- Consulting and customization

} Commercial Partner and/or Lab

What Are the Ways and How Should You Best Protect Software?

- Governed in large part by what you are allowed to do AND by inertia
- What are the differences between each?
  - Patent
  - Copyright
  - Trademark
  - Black box

What is a Copyright?

- Copyright protection subsists….in original works of authorship fixed in any tangible medium of expression …..otherwise communicated, either directly or with the aid of a machine or device. (*17 U.S.C. §102*)

- Applicable for
  - Software
  - Schematics/drawings/images
  - Firmware
  - Publications (technical/journal articles, etc.)

What's *Not* a Copyright?

In no case does copyright protection…extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

17 U.S.C. § 102(b)

Unlike a patent, a copyright protects the **form of expression** rather than the subject matter of the work. Thus, for software, an author is s/he who actually keys the code.

Example: patented operating system vs. copyrighted "look and feel"

You Have Many Rights to License Under Copyrights

- **Reproduction** - Only the copyright owner may make any reproductions or copies of the work
- **Derivative works** - The right to make a work based upon one or more preexisting works
  - A second version of a software program is generally considered a derivative work based upon the earlier version
- **Distribution** - The right to make a work available to the public by sale, rental, lease or lending, and to prevent the distribution of unauthorized copies
- **Performance** - The right to control the public performance of certain copyrighted works
- **Perform or Display Publicly**
- **The right to assign or license** the copyrighted work

It Is Getting Harder to Get Software Patents

Most countries do not provide meaningful software patent protection

A computer program may be patentable if it produces a tangible and useful result

- ***Bilski v. Kappos***, 130 S. Ct. 3218 (2010) – highly anticipated ruling, but had little impact on tech transfer processes
- Solidified understanding of scope of patent subject matter available for business method patents
  - (1) be tied to a particular machine or apparatus or
  - (2) transform a particular article into a different state or thing
- Court also confirmed previous cases holding that fundamental principles of mathematics or science are not patentable
- ***Alice Corp. v. CLS Bank International*** 573 U.S. __, 134 S. Ct. 2347 (2014) – invalidated patent claims drawn to abstract ideas

Black Boxing Uses Software as a Service (SaaS) for Wider Distribution While Hiding the Underlying Technology

- Licensing/delivery model for software
- Keeps source code in a “black box”
  - Know-how/trade secrets
  - FOIA implications
  - Real-time algorithm adjustments and learning systems value (improve with age/date)

Zip Code

BioMedical Software Store
Established 2013
Developers
Commercial
Users
ORTA
Free Apps
Paid Apps
Proprieter
TEDCO
Desk Top
Open source
Web SaaS + API's
Barry's Bikes
Original image by SuperSnappz

Proposed Software Store Model

Govt. ORTA
License
Negotiation

Partnership Intermediary
License Application
& Marketing

Industry

Consumer

Researchers & Other Govt. Labs

Life-cycle Services

Pre-licensing
- Coding and beta testing
- Usability testing
- Security testing
- **Platform adaptation** (SaaS, etc.)

Licensing
- **Web storefront**: payment processing plus code and documentation provision
- Open-source distributions

Post-licensing
- **Code updates**
- IP pooling
- Custom coding
- Platform adaptation (SaaS, etc.)
- Language translation
- FDA/FCC regulatory assistance
- **Community of Practice (COP)**

FLC

How Can Things You're Not Traditionally Allowed to Patent Give You Value?

- Copyright protection for work made for hire is **95-120 years**!
- Copyright rights are different and **complementary** to patent rights
- **Smaller cost** of protection of copyright versus patent; thus, multiple nonexclusive agreements often make good business sense
- The smaller capital outlay required for software makes arms-length transactional management of codes a viable commercialization option, whereas patent decisions are almost always strategic and better for exclusive licensing
- Software and designs have been appropriated from the Lab, resulting in **loss of credit** to the creators and the Lab

Trademarks Can Also Provide Value in Software Licenses

- Statutory authorities differ

- Setting of royalty rate
  - What market can bear
  - What licensee can bear when combined with patent license royalty
- Changing of rate
  - As a trademark ages, it becomes more valuable
  - Should we increase the trademark royalty over time?
  - Should we increase the trademark royalty over time if we are also decreasing patent royalty over time?

Build User Communities

- Build user groups to relieve your PIs of the burden of troubleshooting
- Open-Source Release— leverage the “power of the crowd” to improve the tools
- Build the reputation of the developers and the lab

What Is a CoP and How Can We Use It?

- A group of **active practitioners** across multiple organizations who share knowledge, contacts, data, tips, best practices, questions, and support for each other
- Membership is restricted to those actually performing in the field of the CoP
- CoPs share
  - Knowledge objects (e.g., white papers, PowerPoints, templates, etc.)
  - Knowledge embedded within individuals (e.g., personal approach to a problem)
  - Knowledge embedded in a community (e.g., analyses of members' best) practices)

How a CoP Helps Technology Development and Commercialization

- Disseminates inventions to a **broader audience**
- Enhances CoP member's **role in the community**
- **Supports** licensee's commercialization efforts
- Increases **royalty revenue** back to lab from increased sales by licensee
- Provides ability for lab to **compare** its practices and results with outside organizations
- Provides **forum** for technology transfer licensing personnel to seek input, leads, etc.

CoP Object Model Needs to Be Created & Reused

Databased Content

- Contacts
- Links
- Content
- Events

File Content

- Documents
- Video
- Podcasts

Examples of the Value of Lab Recognition

- Argonne's MPICH implementation of the Message Passing Interface (MPI) is the most popular version of MPI
- LANL's MCNP® is a widely known Monte Carlo code with more than 13,000 licenses worldwide

Stranded Assets— Beware!

What are they?

- Created under subcontracts (with universities or commercial entities when doing work for the U.S. Government)
- Take the form of copyrightable materials (software, drawings, etc.)
- Subcontract fine print (**data rights**) dictates ownership and rights (retained U.S. Government right but not ownership)

Why care?

- Impact on IP portfolios that are otherwise licensed or en route to commercialization
- Lack of clear title (due to failure of subcontractor electing title creates "stranded asset")

Stranded Assets – What Can I Do?

Approaches to overcome unclaimed copyrightable assets created under subcontracts

- **Modify data rights provisions** in the subcontracts
  - Not always possible or welcomed by legal or subcontracts staff

- Work with the subcontractor (university/company) to obtain rights/title greater than U.S. Government rights
  - **Assignment of title**
  - **Royalty-sharing agreement**

- *Challenges*
  - Obtaining disclosure of IP from subcontractor to act upon – closeout procedures
  - Time to negotiate for title or commercial rights

What Can You Do at a GOCO That You Can't at a GOGO & Vice Versa?

- U.S. Government (GOGO) **does not assert copyright** in works created under official duties of employees
- Contractors (GOCO) **can assert copyright** – particularly relevant to software
- Contractors can **assign copyright** to another entity (i.e., a publisher) or the government
- DOD – Defense Authorization Act – awaiting implementation guidance, but some license software under multiple authorities

Why Do We Need to Change the Way We Commercialize Software?

Challenge

Government **software** innovations are difficult to commercialize because of…

- Historical preference for public domain or open source distribution
- Constraints imposed by patent and copyright law in relation to government-created software
- Uncertainty how software innovation can be monetized and who shares in proceeds
- Need for diverse software skills to bring software to multiple platforms

Software Commercialization Themes We Seek to Leverage

- **Disintermediation** allows users/licensees to directly purchase, use, and interact with other users/licensees
- **Standardization** lowers development cost, eases learning for users across different products, and enables specialized developers to reuse code and concepts
- **Virtual** and **distributed** development provides access to greater talent pool and lower costs
- **Multiplatform** services user needs for access from multiple devices and platforms

Commercial Licensing Using EULA

- End-User License Agreements
  - Allow multiple nonexclusive agreements
  - Allow multiple price points for academia, industry, foreign, domestic, etc. (leverage domain)
  - Suitable for high-demand/volume licensing
  - May only provide executable version of code

Commercial Licensing Using Distributors

- License to a software distributor or software store
  - License will convey sublicensing rights
  - License may convey source code
  - Relieves the burden of executing numerous license agreements in exchange for a periodic royalty

App Store Royalties to Love?

Non-commercial (But Still Controlled!)

- Research, QA/QC, other internal uses that may entail a license fee
- Export control may be an issue
- You may not want everyone to have your software, so no sublicensing
- You may want greater visibility in terms of who is interested in your software

Open Source Software (OSS) Distribution

- There are *many* open-source license styles, ranging from permissive to strongly protective
- "**Hereditary**," or "**Viral,**" licenses contain reach-through requirements affecting derivative works
- Common OSS licenses
  - BSD and MIT are **permissive**
  - GPL is strongly **protective**
  - There are many others, including your own terms

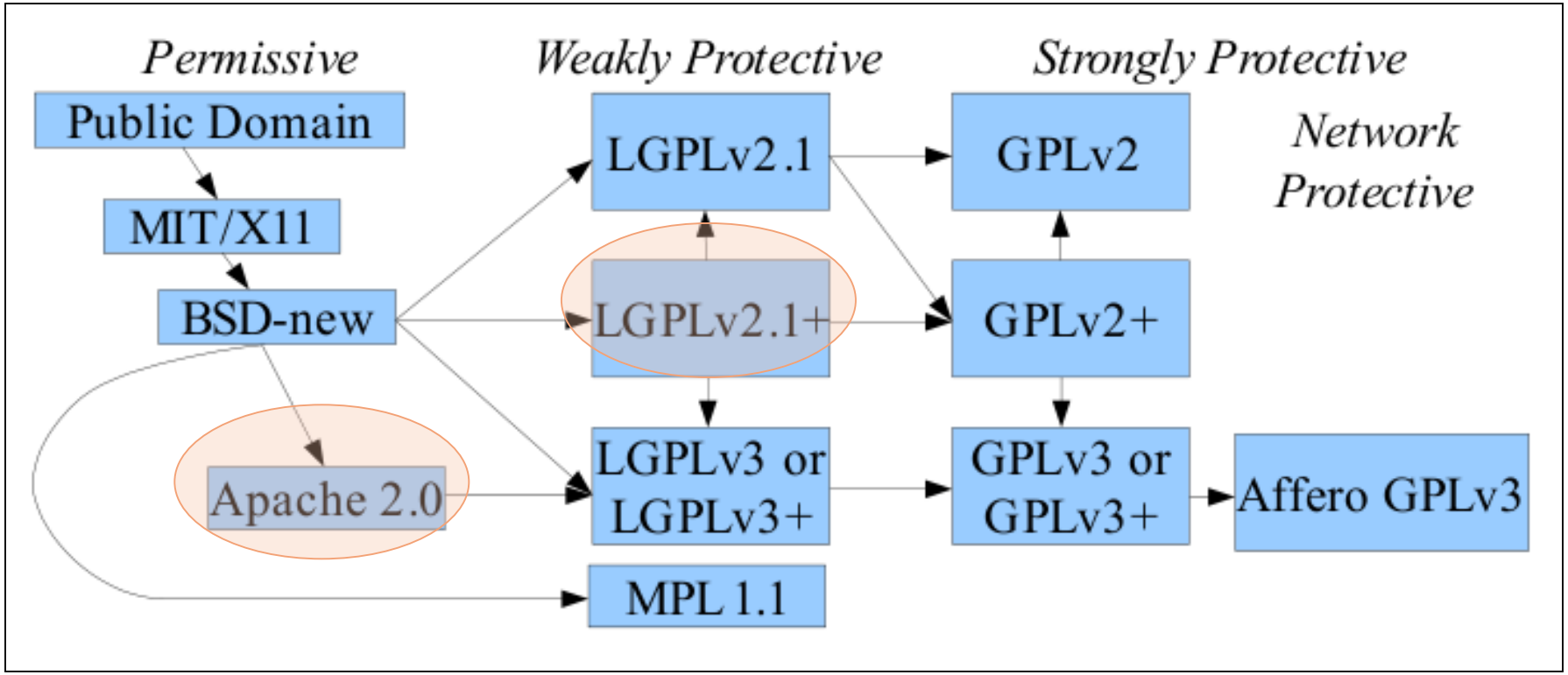
Distribution Decision Criteria

- ❑ Are there preexisting license requirements?
- ❑ Open-source software? (GPL, LGPL, BSD, etc.)
- ❑ Third party contributions?

- ❑ Are there reach-through license requirements stemming from the compiler?
- ❑ Multi-lab collaboration incorporating GOGOs?
- ❑ Is there an IP management agreement?

- ❑ Partnership intermediary?
- ❑ Are there related patents?

Diligence: License Compatibility



http://www.dwheeler.com/essays/floss-license-slide.html

An entire webinar could be devoted to OSS license compatibility

http://dodcio.defense.gov/OpenSourceSoftwareFAQ.aspx

Examples of Incompatible Licenses

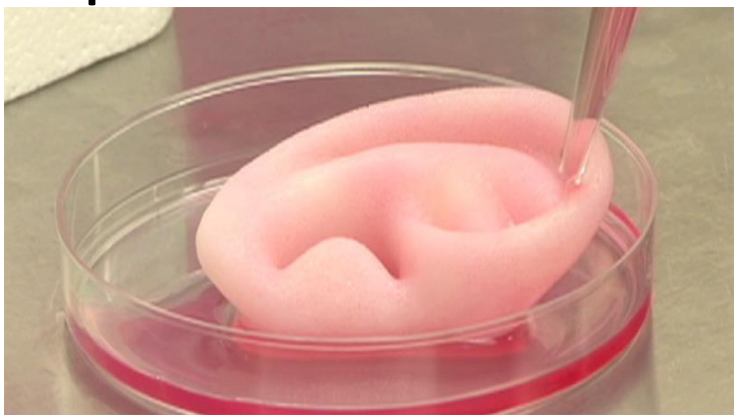
- GNU GPL v.X and NASA Open Source Agreement
  - NASA requires that changes be your “original creation”
- GNU GPL v.X and Yahoo! Public License 1.1
  - Yahoo! has a choice of law clause in section 7
- GNU GPL v.3 and Apache License v2.0
  - If Apache project software becomes a derivative work of GPLv3 software, it must be distributed under GPLv3— but all Apache software must be distributed under the Apache License 2.0

Can SaaS Help in Cases Where the Software Contains Incompatible Licenses?

?

How Will We Handle 3D Inventions?

- E.g., 3D printed livers or ears
- Patent, copyright, know-how, or data?
- Improvements and modularity challenges



Source: http://www.standardmedicalsystems.com

+

Source: http://www.enetmd.com/

- Digital rights management in our 3D store

From Exclusivity to Ubiquity

FLC

NATIONAL MEETING

2016 • CHICAGO

meeting.federallabs.org