Stand Up to the Lawyers

Open Source Licensing and Intellectual Property Law 101 for Developers

Donald Smith

Eclipse Foundation

http://eclipse-ecosystem.blogspot.com/

Agenda

- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

Disclaimers

IANAL

- I
- Am
- Not
- A
- Lawyer

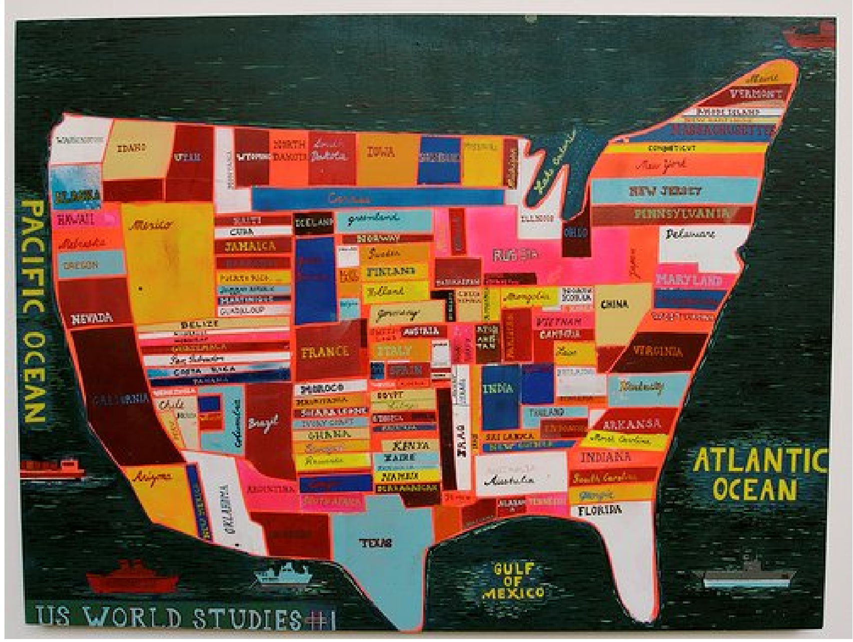
IANAL

- This information is for modeling and debate purposes only
- This information is not legal advice as I am not a lawyer
- Even if I were a lawyer, and able to give professional legal advice, I am still not ***your*** lawyer
- This information is in no way to be considered advice from my Employer, even when I refer to my Community’s license (EPL)
- Many lawyers may not agree with anything presented here, but that’s not the point of this presentation…

This information is an anthology of experiences from one developer to another

A Note on Geography…

- Sorry, but we are going to stick primarily with the American system here
- Most concepts are quite portable, but there will be lots of exceptions in different countries



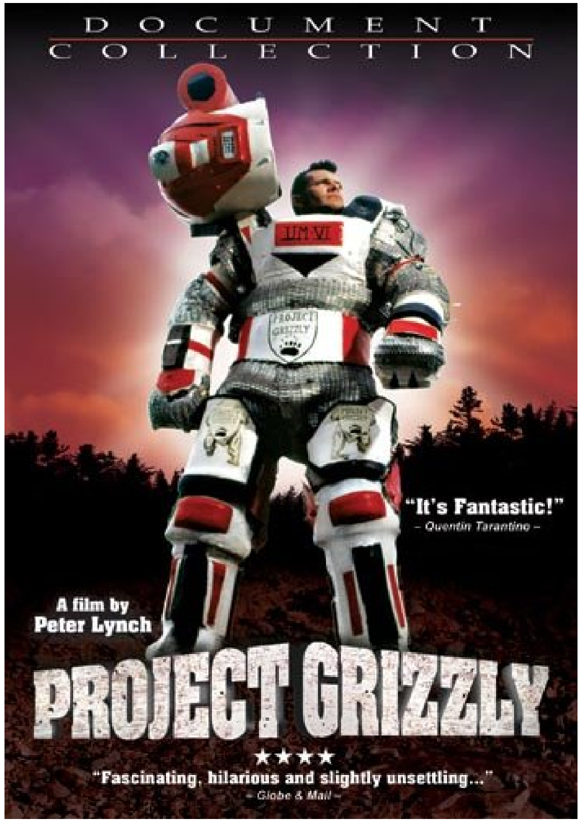
http://www.flickr.com/photos/nss/144588132/

Agenda

- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

The Risk Contradiction

- How much risk is acceptable in your life?
- How much risk is acceptable to your business?
- Why will some people drive a motorcycle without a helmet, yet obsess over obscure and unlikely interpretations of copyright law?
- Use of any license, commercial or otherwise brings with it some legal risk.
- Don't obsess unnecessarily over risk, recognize when your organization *is.*



The Risk Contradiction

- Regard company lawyers as *Insurance Agents*
- Legal approval ~ “Insurance Policy”
- Insurance Policies require:
  - Evidence of very low (no) risk
  - Evidence of a disproportionate benefit (to the agent) for the risk being taken

This is why getting legal approval to do anything is so damn hard…

Agenda

- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

Software Licensing 101

- Free Open Source Software is protected by Copyright © law
  - Just like music, poetry and books
- All software written after 1978 is automatically protected by very restrictive and protective (for the author) rights
- THEREFORE, a Free Open Source software license is actually a **grant of rights** to the end user **not otherwise granted by law**

Wait a second, what's copyright law?

Isn't this related to patents?

What about Commercial Software?

Agenda

- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

Ten ~~Thirty~~ minutes of Intellectual Property Law

- Copyright ©
- Patents
- Copyright VS Patents
- Derivative works
- Trademarks™
- Trade Secrets

Ten minutes of Intellectual Property Law

- Copyright ©
- Patents
- Copyright VS Patents
- Derivative works
- Trademarks™
- Trade Secrets

Copyright ©

- Form of protection defaulted by law that protects “original works of authorship” (original, minimally creative, tangible)
- Actions covered:
  - Reproduction
  - Public Display
  - Publicly Perform
  - Prepare Derivative Works
  - Distribution
- Essentially, except in limited scope*, you simply can’t do any of the above actions on any original work of authorship without permission

* Generally known as “fair use”. 4 criteria. 1. how copyright is used (commercial or non-profit educational), 2. nature of the work. 3. % of material used, 4. effect on comercialability based on the use.

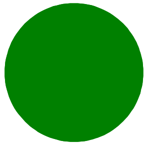
Copyright ©

- Neither registration nor notice is required
- You AUTOMATICALLY inherit Copyright
- Term is your life + 70 years, or 95 years if the creation was "for hire"
- Even if you don't want Copyright you get it
- You do not need to declare Copyright ©, you STILL GET IT
- Declaring Copyright is beneficial for the Copyright holder though:
  - Establishes start date of Copyright so there can be no disputes for your heirs
  - Expands liability for violators

Key Requirements to Inherit Copyright

- Original expression
- Some *minimal* amount of creativity
- Fixed in a tangible medium (thoughts are not Copyrightable)

Does this get Copyright ©?



Does this get Copyright ©?

© 2009

Key Requirements to Inherit Copyright

- Original expression
- Some *minimal* amount of creativity
- Fixed in a tangible medium

Does this get Copyright?

```
for ( i=1; i<k; i++ ) {
    System.out.println(i);
}
```

Does this get Copyright?

```
// Copyright 2009 By John Doe Inc
public void verifyText(VerifyEvent e) {
  Text text = ((Text)e.widget);
  StringBuffer fulltext = new
      StringBuffer(text.getText());
  fulltext.replace(e.start, e.end, e.text);
  String sign=""; //$NON-NLS-1$
  if(fSigned)
   sign="-?"; //$NON-NLS-1$
  e.doit = fulltext.toString()
   .matches(sign+
   "(0?|[1-9][0-9]*|0x[0-9a-fA-F]*
    |0[0-7]+)"); //$NON-NLS-1$
}
```

Who owns the Copyright ©

- Most programmers do not own the right to **any** code they write, their Employer does!
- Most Employment Contracts contain something like the following:

You hereby agree to assign to the Corporation all right, title and interest in and to any and all Inventions whether or not patentable or registrable under copyright or similar statutes, made or conceived or reduced to practice or learned by you, **either alone or jointly with others**, during your employment, which (a) relate to methods, apparatus, designs, products, processes or devices sold, leased, used or under construction or development by the Corporation, or otherwise relate to or pertain to the actual or anticipated business, functions, operations, research or development of the Corporation, (b) utilize any physical or intellectual property owned by the Corporation, or (c) are based on any information or knowledge gained by you through your employment with the Corporation.

This is a really big issue for Open Source projects……

Ten minutes of Intellectual Property Law

- Copyright ©
- Patents
- Copyright VS Patents
- Derivative works
- Trademarks™
- Trade Secrets

Patents

- Exclusive right for a fixed period of time, given by government, to directly or indirectly make, use, sell or import an invention
- An invention is a device, method, process or compound
- An invention must be novel, non-obvious, have utility and be describable in detail such that it is reproducible

BY: http://www.flickr.com/photos/automatt/55799165/

Ten minutes of Intellectual Property Law

- Copyright ©
- Commercial Software Licenses
- Patents
- Copyright VS Patents
- Derivative works
- Trademarks™
- Trade Secrets

Copyrights Versus Patents

- Copyrights protect a specific *expression* of an idea
- Patents protect an *idea* itself
- How evil could this be:
  - As a Copyright holder you grant people the right to copy and create derivative works of your code
  - But do NOT give them the rights to your Patent for the idea the code represents!

In Software, Patents are like Broad Copyright

This might be Copyright ©

```
public void verifyText(VerifyEvent e) {
  Text text = ((Text)e.widget);
  StringBuffer fulltext = new
      StringBuffer(text.getText());
  fulltext.replace(e.start, e.end, e.text);
  String sign=""; //$NON-NLS-1$
  if(fSigned)
   sign="-?"; //$NON-NLS-1$
  e.doit = fulltext.toString()
   .matches(sign+
   "(0?|[1-9][0-9]*|0x[0-9a-fA-F]*
    |0[0-7]+)"); //$NON-NLS-1$
}
```

…but someone could independently produce this:

```
public void verifyText(VerifyEvent ve) {
  String legal=“0123456789ABCDEFabcdef”;
  Text t = ((Text)ve.widget);
  String s =  t.getText();
  s = s.substr(0,ve.start)
   + ve.text + s.substr(ve.end, s.length());
int start=0;
if(fSigned)
   {start=1;
    if s[1]!=“-”; {
      ve.doit=false; return;}
while(start++<s.length())
   if(legal.indexOf(s.charAt(start))=-1)
     {ve.doit=false; return;}
ve.doit=true;
}
```

Please bear with me if the code is not perfect or correct. The point is that Copyright only applies to a particular expression and therefore with Software, it's quite possible, easy and even likely to have many implementations of the same idea. This is OK, unless…..

Patent To Protect the Idea, not the Expression

Patent: 5555121212

Date Granted: February 30th, 2001

Title: “A method for verifying selected text to be a numeric representation as denoted under ANSI numeric representation standard”

Authors: Joe Schmoe

Claims: 1. All methods for verifying text, 2. All methods for verifying that a text is a number, 3. All methods for verifying that text is an ANSI defined number, 4. All methods for verifying text that is an ANSI defined number by brute force iteration.

Pseudo Implementation:

```
Some Selection Event is Raised, E {
  Extract the Text, T, to be verified from E
  Validate if the first Character T is a sign (+/-), if relevant
  Iterate over the text T, character by character C
    Validate that each C belongs to the ANSI set of valid Hex characters
  If everything is Valid, set the event execution flag to True, otherwise false
}
```

*** Please note this is not a real patent, it is an example of patenting the idea instead of the expression. I am not implying this idea is or ever was patentable or patented, or that software patents are a good or bad idea, but if this idea were patentable, this might be what the patent abstract looked like. Apologies to all Joe Schmoe’s.

Ten minutes of Intellectual Property Law

- Copyright ©
- Commercial Software Licenses
- Patents
- Copyright VS Patents
- Derivative works
- Trademarks™

Derivative Works

"…**a work based on one or more preexisting works**, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgement, condensation, or **any other form in which a work may be recast, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications,** which, as a whole, represents an original work of authorship, is a 'derivative work'"

Section 101 of U.S. Copyright Act

- Derivative works are not considered original creations
- They are considered *copies* of the original and you therefore need permission to display, distribute, etc, the derivative work

Derivative Works

Starting with this….

```
public void verifyText(VerifyEvent e) {
  Text text = ((Text)e.widget);
  StringBuffer fulltext = new
      StringBuffer(text.getText());
  fulltext.replace(e.start, e.end, e.text);
  String sign=""; //$NON-NLS-1$
  if(fSigned)
   sign="-?"; //$NON-NLS-1$
  e.doit = fulltext.toString()
   .matches(sign+
   "(0?|[1-9][0-9]*|0x[0-9a-fA-F]*
    |0[0-7]+)"); //$NON-NLS-1$
}
```

…makes this a derivative work

```
public void verifyText(VerifyEvent e) {
  Text text = ((Text)e.widget);
  // Better way to get String Buffer
  StringBuffer fulltext = new
      text.getStringBuffer();
  fulltext.replace(e.start, e.end, e.text);
  //Sign not important to me
  e.doit = fulltext.toString()
   .matches(
   "(0?|[1-9][0-9]*|0x[0-9a-fA-F]*
    |0[0-7]+)"); //$NON-NLS-1$
}
```

Please bear with me if the code is not perfect or correct. The point is that even making major changes and improvements to someone else’s code, even if it’s way, way, way cooler and better, is still a derivative work.

Ten minutes of Intellectual Property Law

- Copyright ©
- Patents
- Copyright VS Patents
- Derivative works
- Trademarks™
- Trade Secrets

Trademarks™

- Identifies the origin of product or service
- Distinctive symbols, pictures, or words
- Registration® is not required, but helps
  (similar to copyright – expands your ability to sue, and more)
- For example, it is never proper to write "Java". You really should always denote it as Java™
- Companies sometimes look evil defending their Trademarks, but this is not necessarily because they are evil…
  - If you establish something as a Trademark, you are **compelled by law** to defend any perceived infringement on the trademark or you seriously jeopardize your rights
  - For example, when Starbucks sued "Sambucks", they were not necessarily being evil, but acting as compelled by law

Few open source licenses grant any trademarks rights

Java™ is a trademark of Sun Microsystems®

Ten minutes of Intellectual Property Law

- Copyright ©
- Patents
- Copyright VS Patents
- Derivative works
- Trademarks™
- Trade Secrets

Trade Secrets

(BY:) flickr.com/photos/59953599@N00/2852302091/

- Aka "Confidential Information"
- Non-compete clauses
- Non-disclosure contracts
- "Reverse Engineering" Restrictions
- Not a typical software developer problem – Cryptography, compression
- More of an issue for hardware touch points – networking, mobile, etc.

Intellectual Property Law Summary *As it applies to Software Licensing

- Software is protected by Intellectual Property law
  - Even if you're the author of said code and don't want it to be
- Patents protect and restrict the use if *ideas* whereas Copyright protects and restricts *expressions* of an idea
- Derivative Works, even if substantially modified, are **not** considered original themselves and therefore you need permission from original author
- Trademarks identify products or services and are protected by law, but **must** be rigorously defended to be valid
- Trade Secrets not generally an issue, but beware of certain domains and when you're close to the metal

An OS license is a *grant* of rights not given by IPL

Agenda

- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

Commercial Software Licenses

- Commercial Software often imposes extra restrictions on users that are not covered by Intellectual Property law
  - Agreement not to disassemble or reverse Engineer
  - Agreement to use on only one computer
  - Agreement not to transfer or resell your license to another entity
  - Agreement to allow software to report usability metrics periodically
  - Agreement not to rent or lease the computer with the software
  - Agreement to notify all staff and users of computer of all rules of the software license
  - Agreement to wear purple shirts when using the software

The Subtle Difference

- Commercial Software Licenses (aka “EULA”) are *restrictive*
  - To enforce a license, owner must be able to prove end user acceptance
    - Signed contract, click-through, shrink-wrap*
  - Disputes may fall into Contract law
- Free Open Source Licenses *grant* rights, not take them away
  - There is no need to prove acceptance
  - No one can claim they didn’t know what the conditions were since most actions would be violation of Copyright © law and therefore the license need to be consulted
  - Disputes likely to be Intellectual Property law

But as a *user*, you need to trust the Copyright © owner granting the license actually is the copyright owner…

Open Source EULA’s

- Open Source Software may still have a “EULA” or User Agreement
  - Typical in cases where an OS Project is distributing 3rd party OS Code
  - Requires end users to be aware of the 3rd party licenses that may apply
  - Clarifies *what* software is covered – for example include CVS code (or update manager in Eclipse)
- Example OS EULA’s
  - http://www.eclipse.org/legal/epl/notice.php
  - http://www.mozilla.com/legal/eula/firefox-en.html

Agenda

- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

Key Characteristics of Software License

- Definitions
- Grant of Copyright License Rights
- Warranty and Liability
- Jurisdiction and Duration
- Sublicensing
- Reciprocity
- Patent Rights
- Patent Retaliation
- Use of Trademarks

Fairly standard
legal mumbo jumbo

Differentiating
Factors

Wait! What makes a Software License an *Open Source* Software License?

Definition of an Open Source License

- Who was better, the 1999 New York Yankees or the 1973 Oakland Athletics?
- Did Chris Moneymaker just get incredibly lucky when he won the 2003 WSOP™ or was he skillful and just been unlucky since?
- Which is the better spin off, CSI™ Miami, or CSI™ New York?

The point is, we could debate this all day and it's subjective. Many people and organizations (including myself) agree that the "Open Source Initiative" (OSI) defines the industry standard of what is an open source license.

OSI Definition of an Open Source License

1. Free Redistribution
   - No Fees or Royalties
2. Source Code
   - Included and Redistributable
3. Derived Works
   - Allowed and redistributable under same terms or “better”
4. Integrity of Authors Source Code
   - May require derived works to carry different name or version
5. No discrimination against person or group of persons
   - Can warn of legal constraints such as trade embargos, but not explicitly forbid

OSI Definition of an Open Source License

6. No discrimination against fields of endeavor
   - Closes loophole that might restrict commercial use
7. Distribution of license
   - Must be self standing and not require a non-disclosure or other agreement
8. License Must be Not Be Specific to a Product
   - "Distribution" may be better word – must allow selective use of functionality
   - Closes another loophole
9. Must not restrict other software
   - In the distribution, so GPL is OK
10. Technology Neutral
    - Cannot restrict use to certain platforms (Windows™ for example)

Key Characteristics of Software License

- Definitions
- Grant of Copyright License Rights
- Warranty and Liability
- Jurisdiction and Duration
- Sublicensing
- Reciprocity
- Patent Rights
- Patent Retaliation
- Use of Trademarks

Fairly standard
legal mumbo jumbo

Differentiating
Factors

Key Characteristics of Software License

- Definitions
- Grant of Copyright License Rights
- Warranty and Liability
- Jurisdiction and Duration
- Sublicensing
- Reciprocity
- Patent Rights
- Patent Retaliation
- Use of Trademarks

Fairly standard
legal mumbo jumbo

Differentiating
Factors

Definitions

- Defines important terms used within the license
- May seem trivial, but the definitions are often "telling" of what is important in the license
- May define terms that are used in applicable laws
  - Many licenses clarify the term "Derivative Works", which is defined in law, but expanded for clarification specific to software
- May define terms that are interesting to the license
  - Community based licenses like Apache and Eclipse define "Contribution" and "Contributor"

Grant of Copyright Rights

- Sometimes called “Grant of Copyright License” or “Grant of Rights”
- Grants you rights you do not have under Copyright law such as the right to copy, create derivative works, to redistribute, etc.
- There may be conditions, such as “Reciprocity” (more later)
- This is often short and sweet

From Apache 2.0 license:

Grant of Copyright License: Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

Warranty and Liability

- YOU MAY NOTICE SECTIONS OF LICENSES THAT SEEM LIKE THEY ARE YELLING
- THEY BASICALLY SAY, “HEY THERE, YOU DIDN’T PAY FOR THIS SO THERE IS NO WARRANTY, AND YOU CAN’T SUE US IF IT BREAKS, OK? AND ALSO WE ARE NOT LIABLE FOR ANYTHING BAD THAT HAPPENS, EVEN IF WE INTENTIONALLY DID SOMETHING WRONG TO THE EXTENT AFFORDED BY LAW.”
- NOTE – THE YELLING IS NOT BEING RUDE, OR LAWYERS BEING LAWERS – IT’S ACTUALLY A COURT PRECEDENT THAT IT BE ALL IN UPPER CASE

Confirmed in:

UCC 2-316

UCC 1-201(10)

Amendments to rules of court, Virginia Supreme court Nov 1, 2002

Jurisdiction and Duration

- Some licenses specify a legal jurisdiction and terms
- Generally a simple way to further minimize any possible legal risks
  - Location
  - Timeline
  - Type of trial
- For example, the EPL states:

"This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation."

Key Characteristics of Software License

- Definitions
- Grant of Copyright License Rights
- Warranty and Liability
- Jurisdiction and Duration
- Sublicensing
- Reciprocity
- Patent Rights
- Patent Retaliation
- Use of Trademarks

Fairly standard
legal mumbo jumbo

Differentiating
Factors

Grant of Sublicense Rights

- Sublicense – Licensee has right to license to 3rd party with same terms
- Relicense – Ability to distribute under different license
- Examples
  - Apache and Eclipse allow sublicense
  - Apache allows full relicense of your changes
    - Essentially you own the copyright of derivative works
  - Eclipse allows relicense of *object code* with disclaimers*
  - Each distribution of GPL code is technically a license from the Author, so technically not sublicensed

Eclipse “Relicense” Disclaimers:

Disclaim warranties and liability of contributors

States what’s different from the original work (if anything)

Must point to where original work source can be found

Reciprocity

- The exchange of comparable concessions
- No, File, Module, Derivative Works, Hosting, Container
- "Reciprocity Reach" – The extent to which your derivative works must be licensed under the same terms and conditions
  - Apache 2.0 – No Reciprocity
  - EPL 1.0 – No Reciprocity on modules that are not derivative works
  - GPL 3.0 – Reciprocity on "covered work"
    - "covered work" includes software "based on the Program"

Patent Rights and Retaliation

- Many licenses explicitly grant you rights to any patents the Copyright holders may have with respect to the code (Apache, EPL)
- Some licenses have implicit patent grants (GPL)
- Many licenses terminate the patent grants if you sue for patent infringement on any of the licensed code
  - EPL and Apache Patent rights explicitly terminate upon suing **any** entity related to the licensed code
  - GPL Patent rights implicitly terminate upon restricting royalty-free distribution of any GPL code

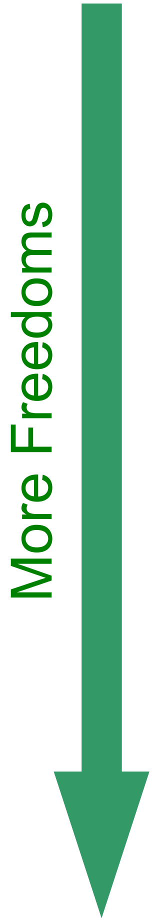
Use of Trademarks™

- Some licenses (Apache) explicitly clarify that the license does not permit the use of the trademark except as to document the origin of the source code as per the license requirements
- The PHP License explicitly defines how products that use PHP cannot be named
- Some licenses (EPL) do not mention trademarks in the license and therefore default Trademark law applies
    - This is intentional to allow Trademark flexibility outside of the scope of the use of the license

Agenda

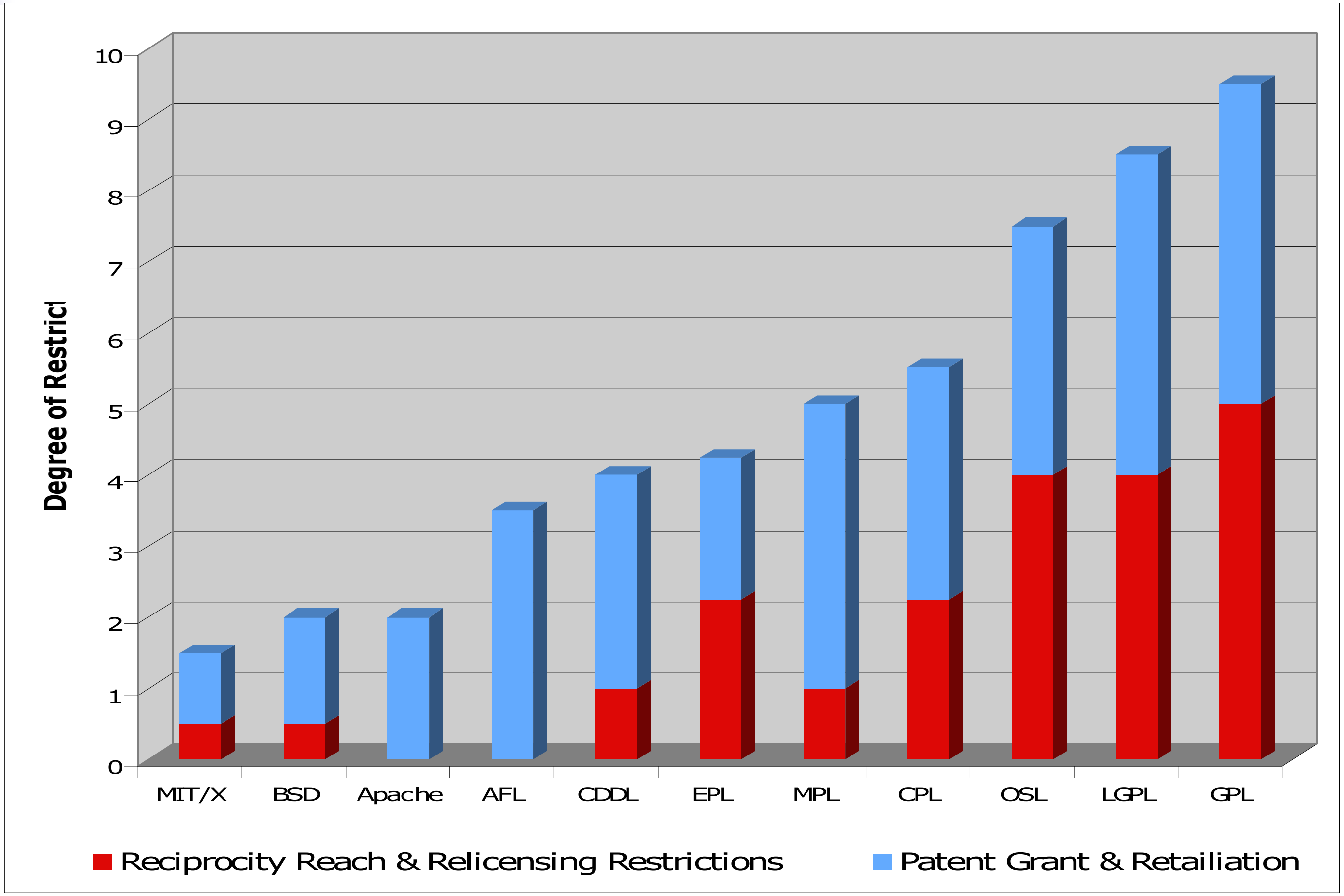
- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

Degree of Freedom (Generalized)



- Commercial License – “You can’t see our source”
  - Example – Pick one ☺
- Restricted Source – “You can see it, but not do much with it”
  - Sun Community Source License (SCSL)
  - Microsoft Shared Source (Ms-PL Ms-RL Ms-CL) (Permissive/Reference/Collaborative)
- Copyleft – “If you change it, you must publish the changes”
  - “Strong” Example – GPL
  - “Weak” Example – EPL, MPL
- Non-Copyleft – “Do what you want, just don’t claim it’s yours and don’t let us get blamed if your changes suck”
  - BSD, Apache

Degree of Restrictions (Generalized)



This graph is generalized and subjective in nature. This graph is Copyright© Cliff Schmidt and used under EPL.

Comparing Three Popular OS Licenses

	GPL v3	EPL v1	Apache v2
Reciprocity	[redacted]	[redacted]	[redacted]
Sublicense / Relicense	[redacted]	[redacted]	[redacted]
Patent Rights	[redacted]	[redacted]	[redacted]
Patent Retaliation	[redacted]	[redacted]	[redacted]
Trademark	[redacted]	[redacted]	[redacted]

This slide is not legal advice and is subjective. IANAL. Consult your trusted advisors before basing any decisions on this chart.

Agenda

- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

Choosing a License for your OS Project

- Strongly advocate picking a license from the Open Source Initiative (OSI) at opensource.org
- Avoid license proliferation, use a popular license like Apache, BSD, MIT, GPL, LGL, EPL, MPL.
- Choosing a license can be like choosing a religion
  - How strongly do you feel about the statement "**ALL** Software should be free?"
  - Would you be upset, or proud if someone else makes money from your copyright?
  - If your project is wildly successful, will you still be happy with your license choice?
    - Plan for success

Agenda

- The Risk Contradiction
- What is a software license?
- Thirty minutes of Intellectual Property Law
- Free vs Commercial Software Licenses
- Key Characteristics of a Software License
- Comparing and contrasting popular OS Licenses
- Choosing a License For Your OS Project
- Using OS in your Project

Using OS In Your Project

- Key question – What is your business model?
  - License revenue? Consulting? Support and services? Documentation?
- Considerations
  - GPL Considered "Viral" and limits adoption for use in commercially licensed products
  - Apache, BSD are extremely friendly to commercial use and redistribution
  - Eclipse and Mozilla licenses are also very friendly to commercial use and redistribution PLUS they give reciprocity
- Don't fall for the Risk Contradiction. Open Source licenses are not too good to be true!

Q&A