IP Issues in Open Source

Eclipse Banking Day

Janet Campbell
Eclipse Foundation Inc.
Legal Counsel & Manager, IP

Jeffrey D. Neuburger
PROSKAUER ROSE LLP
Partner

Key Areas of Focus

Open Source

- What is Open Source?
- Open Source Licenses Explained

Doing your Homework

- Areas of Focus
- The Eclipse Example
- Distributors – Planning Ahead
- The Importance of Active Management

Open Source Software

- Software that is distributed with its source code (or an offer for it) under a license agreement that allows for its use and modification.
    1. “Permissive” or “Attribution” Open Source License Agreements

       E.g, BSD License
    2. “Copyleft” Open Source License Agreements

       E.g., EPL, MPL
- Distribution is not a requirement; licensees can use internally without obligations.

Open Source Initiative

Determined by the License Characteristics.

Defined by the Open Source Initiative

http://www.opensource.org/docs/osd

1. Free Redistribution
   - No fees or royalties
2. Source Code
   - Included and Redistributable
3. Derived Works
   - Allowed and redistributable under same terms.
4. Integrity of The Author's Source Code
   - The license must permit distribution of software built from modified source code.
5. No Discrimination Against Persons or Groups
   - The license must not discriminate against any person or group of persons.

Open Source Initiative Cont…

Determined by the License Characteristics.

Defined by the Open Source Initiative

http://www.opensource.org/docs/osd

6. No Discrimination Against Fields of Endeavor
   - Can’t restrict commercial use for example.
7. Distribution of License
   - Must be self standing and not require a non-disclosure or other agreement
8. License Must Not Be Specific to a Product
   - The rights attached to the program must not depend on the program's being part of a particular software distribution.
9. License Must Not Restrict Other Software
   - The license must not place restrictions on other software that is distributed along with the licensed software.
10. License Must Be Technology-Neutral
    - Cannot restrict use to certain platforms

Grounded in Copyright Law

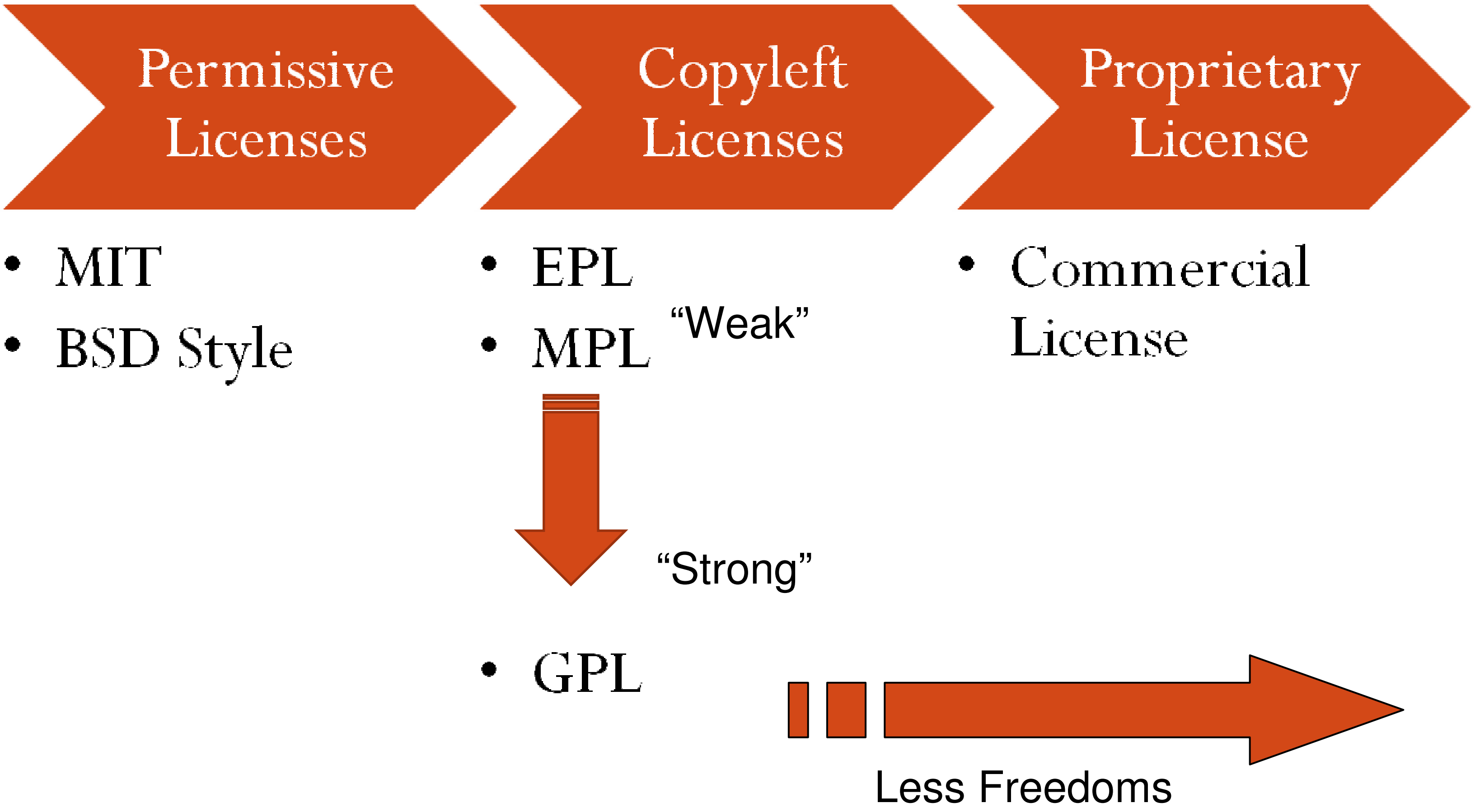
- Form of protection defaulted by law that protects “original works of authorship” (original, minimally creative, tangible)
- Actions covered under US Law (although many concepts are portable):
  - Reproduction
  - Public Display
  - Publicly Perform
  - Prepare Derivative Works
  - Distribution
- Essentially, except in limited scope (e.g. fair use), you can’t do any of the above actions on any original work of authorship without permission

Other Terms Sometimes Included

- Express grant of applicable patent rights
- Disclaimer of warranties and liability
- Indemnification of copyright holders in the case of commercial distribution

- Terms affecting the redistribution of modified source code (Also referred to as “copyleft” attributes)

The License Spectrum



Permissive Licenses

- MIT
- BSD Style

Copyleft Licenses

- EPL
- MPL “Weak”

“Strong”

- GPL

Proprietary License

- Commercial License

Less Freedoms

“Permissive” Licenses

- “Permissive” or “Non-Copyleft” Free software comes from the author with permission to redistribute and modify, and add additional restrictions to the license terms.
  - A subsequent party can modify the non-copyleft free program and distribute the modified program as a proprietary software product, without making the source code available to others on the same terms.
  - E.g., BSD License

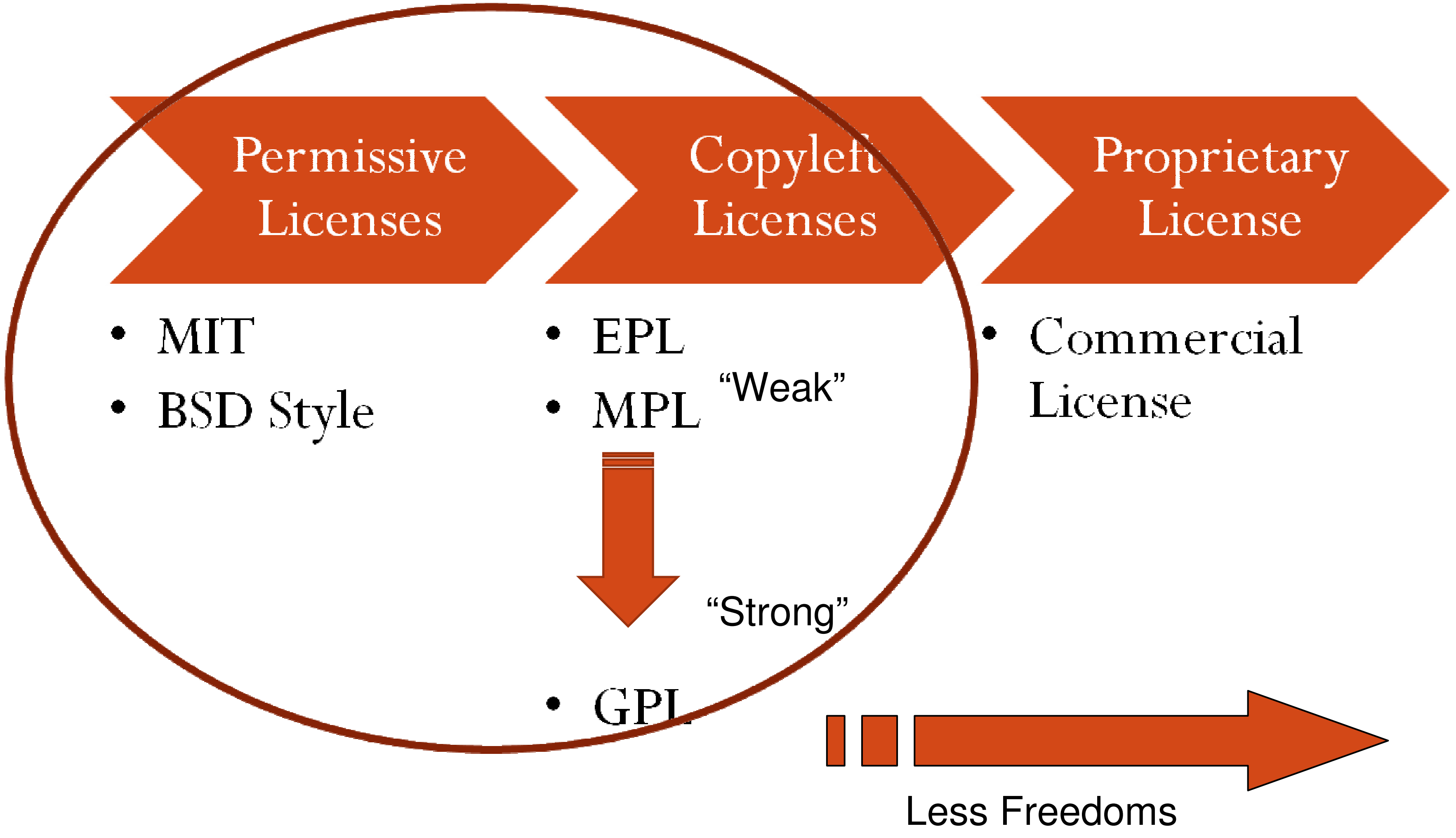
“Copyleft” Licenses

- *Copyleft* requires all modified versions of the program to be provided under the same license as the original software was obtained.
- The impact of copyleft varies from license to license:
  - Under the Eclipse Public License (EPL), the copyleft requirement only applies to that which is in the same module as the EPL code or that which is otherwise a “derivative work” of the EPL code as defined by copyright law
  - Under the GPL 2.1, merely “linking” GPL code with other code may require that the other code (and the combination) be licensed under the GPL.
- Any copyleft license needs to be reviewed carefully within the context of what you wish to do.

“Commercial” Licenses

- Commercial Software often imposes extra restrictions on users:
  - 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

The License Spectrum



Permissive Licenses

- MIT
- BSD Style

Copyleft Licenses

- EPL
- MPL

"Weak"

"Strong"

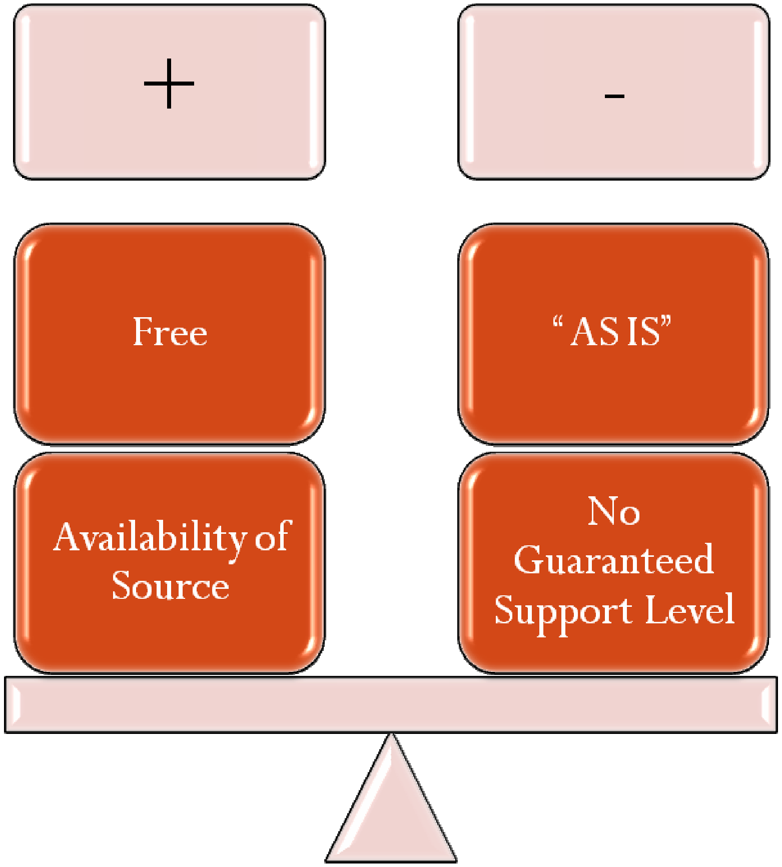
- GPL

Proprietary License

- Commercial License

Less Freedoms

OSS License Characteristics



+

-

Free

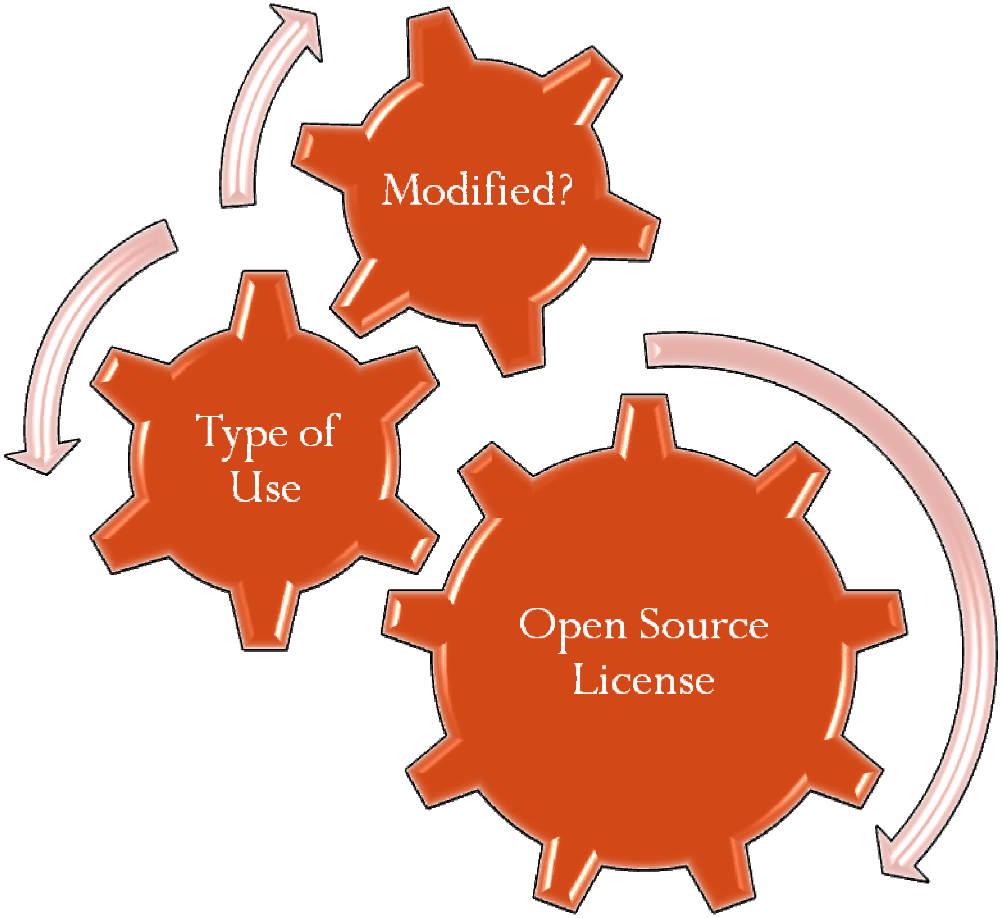
Availability of Source

“ AS IS”

No Guaranteed Support Level

What are my (+)s and (-)s and how does it balance out for me?

Does the License suit your use?

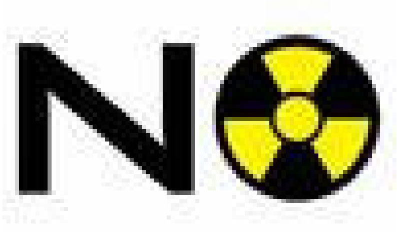


Type of Use

Internal Use

- Most open source licenses are well suited to internal use.

- Watch for Modifications to Popular Open Source Licenses



Type of Use

Distributed Use & Permissive License

- Generally do not pose any difficulties

Distributed Use & Copyleft License

- More careful review is required
  - How is the code structured?
  - Is the code modified?
  - Are there multiple license relationships

Doing Your Homework

What about Risk?

- Risk is associated with the use of any software – open source or proprietary.
- While the concern about litigation exists, there has been very little litigation concerning open source to date.
- None of the enforcement proceedings so far interpret the tough issues (e.g. license compatibility, the scope of the GPL’s copyleft provisions).
- Open source continues to flourish.
- There are benefits to using it.
- Risk can be mitigated by doing your homework.

How Eclipse Mitigates Risk

Code Originates from Three Sources:

1. Contributions from Eclipse Committers
2. Contributions from Contributors
3. Contributions from third party sources (e.g. another open source project)

Securing the Necessary Rights

Committer Contributions

- Legal Agreements are entered into to secure the necessary rights to have the code included in Eclipse.
  - Member Committer Agreement
  - Individual Committer Agreement
  - If the Individual Committer is Employed – An Employer Consent Form
- Through these Agreements, the Committer agrees that the Eclipse Public License (EPL) governs the code submitted by the Committer.

Securing the Necessary Rights

Contributor Contributions

- All content must be submitted through any of the channels existing on the Eclipse Foundation website such as, the Bugzilla bug reporting system.
- This material is licensed to others under the terms of the Eclipse Foundation Terms of Use.
- The Eclipse Foundation Terms of Use define the license terms that apply to any intellectual property submitted to the Eclipse Foundation website.
  - Modifications to EPL code are governed by the EPL
  - Modifications to code governed by another license are governed by that other license and the EPL.

Securing the Necessary Rights

Contributor Contributions continued…

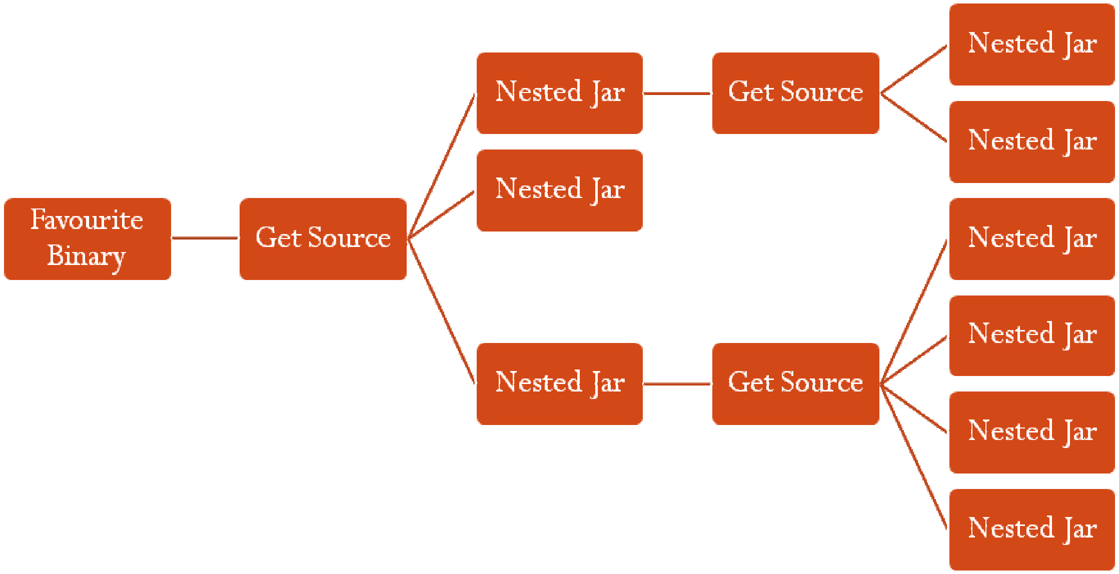
- For all other contributions…
  - “you grant (or warrant that the owner of such rights has expressly granted) the Eclipse Foundation, the Members and the users of this Web-site a worldwide, unrestricted, royalty free, fully paid up, irrevocable, perpetual, non-exclusive license to use, make, reproduce, prepare derivative works of, publicly display, publicly perform, transmit, sell, distribute, sublicense or otherwise transfer such Materials, and/or derivative works thereof, and authorize third parties to do any, some or all of the foregoing including, but not limited to, sublicensing others to do any some or all of the foregoing indefinitely.”

Third Party Contributions

- Third party contributions such as code originating from another open source project (e.g. www.apache.org) are licensed under the license terms that apply to that project.
- Eclipse completes due diligence on each of these packages.

Third Party Packages

- It is common for more material to be included in the distribution than is needed.
- Can we narrow the scope?



Example – Apache Muse 2.0

- Committer wanted to use Muse 2.0
- Grabs the binary – one file – set to go….
- Reads up on Muse 2.0 and finds that some of the functionality is dependent on Axis 2 Version 1.1.
- Committer grabs another binary.
- And now we have two – this is going to be easy

Muse 2.0 – 1st Level of Nesting

```
muse-2.0.0-bin\modules\axis2\muse-platform-axis2-2.0.0.jar
muse-2.0.0-bin\modules\core\muse-core-2.0.0.jar
muse-2.0.0-bin\modules\core\muse-util-2.0.0.jar
muse-2.0.0-bin\modules\core\muse-util-qname-2.0.0.jar
muse-2.0.0-bin\modules\core\muse-util-xml-2.0.0.jar
muse-2.0.0-bin\modules\core\muse-util-xstream-2.0.0.jar
muse-2.0.0-bin\modules\core\muse-wsa-soap-2.0.0.jar
muse-2.0.0-bin\modules\osgi\core\muse-core-2.0.0.jar
muse-2.0.0-bin\modules\osgi\core\muse-osgi-core-2.0.0.jar
muse-2.0.0-bin\modules\osgi\core\muse-osgi-soa-axis2-2.0.0.jar
muse-2.0.0-bin\modules\osgi\core\muse-osgi-soa-core-2.0.0.jar
muse-2.0.0-bin\modules\osgi\core\muse-util-all-2.0.0.jar
muse-2.0.0-bin\modules\osgi\core\muse-wsa-soap-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-api\muse-wsdm-muws-adv-api-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-api\muse-wsdm-muws-api-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-api\muse-wsdm-wef-api-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-api\muse-wsn-api-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-api\muse-wsrf-api-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-api\muse-wsx-api-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-impl\muse-wsdm-muws-adv-impl-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-impl\muse-wsdm-muws-impl-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-impl\muse-wsdm-wef-impl-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-impl\muse-wsn-impl-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-impl\muse-wsrf-impl-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-impl\muse-wsrf-rmd-2.0.0.jar
muse-2.0.0-bin\modules\osgi\ws-fx-impl\muse-wsx-impl-2.0.0.jar
muse-2.0.0-bin\modules\tools\muse-tools-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-api\muse-wsdm-muws-adv-api-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-api\muse-wsdm-muws-api-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-api\muse-wsdm-wef-api-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-api\muse-wsn-api-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-api\muse-wsrf-api-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-api\muse-wsx-api-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-impl\muse-wsdm-muws-adv-impl-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-impl\muse-wsdm-muws-impl-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-impl\muse-wsdm-wef-impl-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-impl\muse-wsn-impl-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-impl\muse-wsrf-impl-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-impl\muse-wsrf-rmd-2.0.0.jar
muse-2.0.0-bin\modules\ws-fx-impl\muse-wsx-impl-2.0.0.jar
```

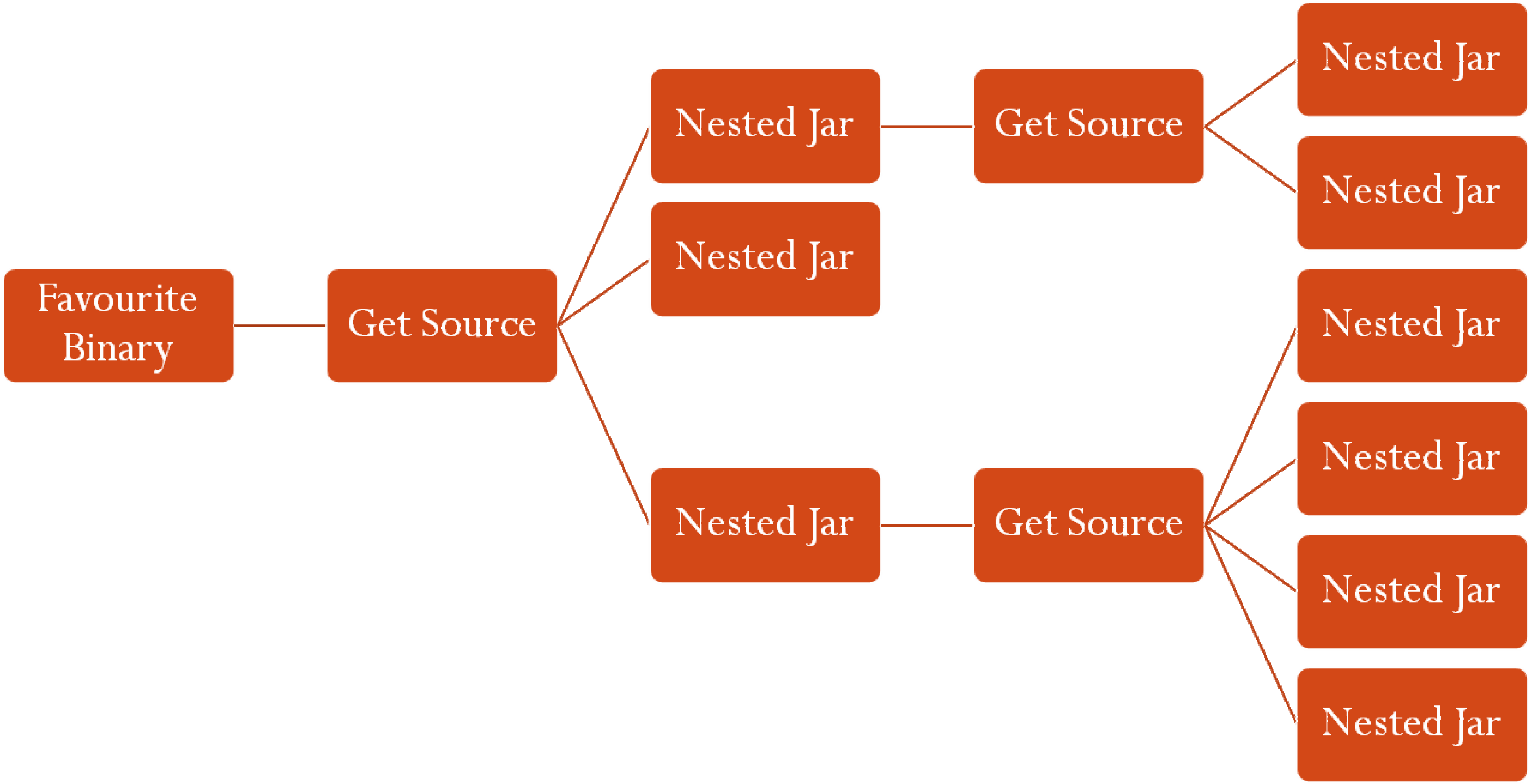
http://www.apache.org/dist/ws/muse/2.0.0/bin

Axis 2 v. 1.1 – 1st Level of Nesting

```
\axis2-1.1\lib\activation-1.1.jar
\axis2-1.1\lib\axiom-api-1.2.jar
\axis2-1.1\lib\axiom-impl-1.2.jar
\axis2-1.1\lib\axis2-adb-codegen-1.1.jar
\axis2-1.1\lib\axis2-java2wsdl-1.1.jar
\axis2-1.1\lib\axis2-kernel-1.1.jar
\axis2-1.1\lib\axis2-soapmonitor-1.1.jar
\axis2-1.1\lib\axis2-tools-1.1.jar
\axis2-1.1\lib\backport-util-concurrent-2.2.jar
\axis2-1.1\lib\commons-fileupload-1.1.1.jar
\axis2-1.1\lib\commons-io-1.2.jar
\axis2-1.1\lib\geronimo-spec-jms-1.1-rc4.jar
\axis2-1.1\lib\jaxen-1.1-beta-10.jar
\axis2-1.1\lib\jibx-run-1.1.2.jar
\axis2-1.1\lib\neethi-2.0.jar
\axis2-1.1\lib\stax-api-1.0.1.jar
\axis2-1.1\lib\wsdl4j-1.6.1.jar
\axis2-1.1\lib\xalan-2.7.0.jar
\axis2-1.1\lib\xercesImpl-2.8.1.jar
\axis2-1.1\lib\XmlSchema-1.2.jar
\axis2-1.1\lib\annogen-0.1.0.jar
\axis2-1.1\lib\axiom-dom-1.2.jar
\axis2-1.1\lib\axis2-adb-1.1.jar
\axis2-1.1\lib\axis2-codegen-1.1.jar
\axis2-1.1\lib\axis2-jibx-1.1.jar
\axis2-1.1\lib\axis2-saaj-1.1.jar
\axis2-1.1\lib\axis2-spring-1.1.jar
\axis2-1.1\lib\axis2-xmlbeans-1.1.jar
\axis2-1.1\lib\commons-codec-1.3.jar
\axis2-1.1\lib\commons-httpclient-3.0.1.jar
\axis2-1.1\lib\commons-logging-1.1.jar
\axis2-1.1\lib\jakarta-httpcore-4.0-alpha2.jar
\axis2-1.1\lib\jibx-bind-1.1.2.jar
\axis2-1.1\lib\mail-1.4.jar
\axis2-1.1\lib\servletapi-2.3.jar
\axis2-1.1\lib\woden-1.0.0M6.jar
\axis2-1.1\lib\wstx-asl-3.0.1.jar
\axis2-1.1\lib\xbean-2.2.0.jar
\axis2-1.1\lib\xml-apis-1.3.03.jar
```

http://ws.apache.org/axis2/download/1_1/download.cgi

Requirements are Identified



Eclipse Due Diligence

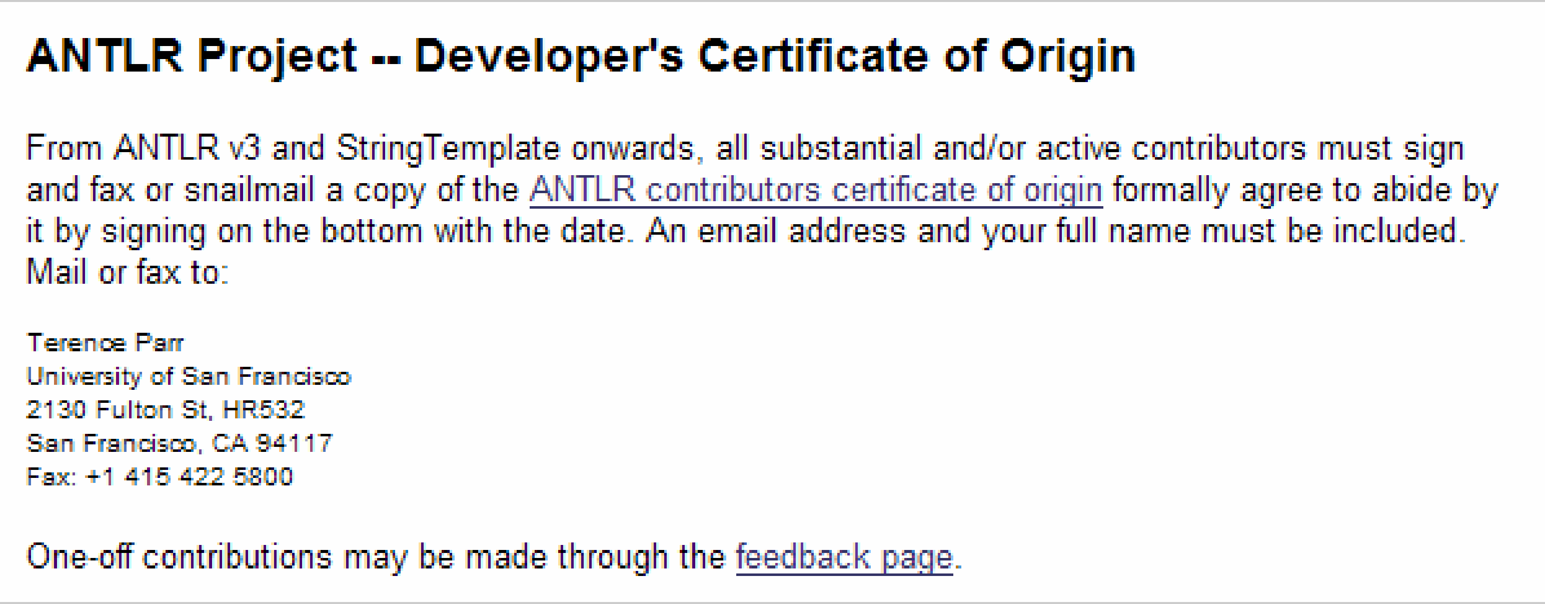
- The components that are identified as needed are submitted for review.
- Each component is examined from the standpoint of:
  1. Provenance
  2. License Compatibility
- We use tools to help us

How is Provenance Managed

Who wrote this stuff and how did they agree to the license?

ANTLR 3.0

Developers who are involved in ongoing development of ANTLR or contribute significant code, must sign and return a “Certificate of Origin” document (www.ANTLR.org).



ANTLR Project -- Developer's Certificate of Origin

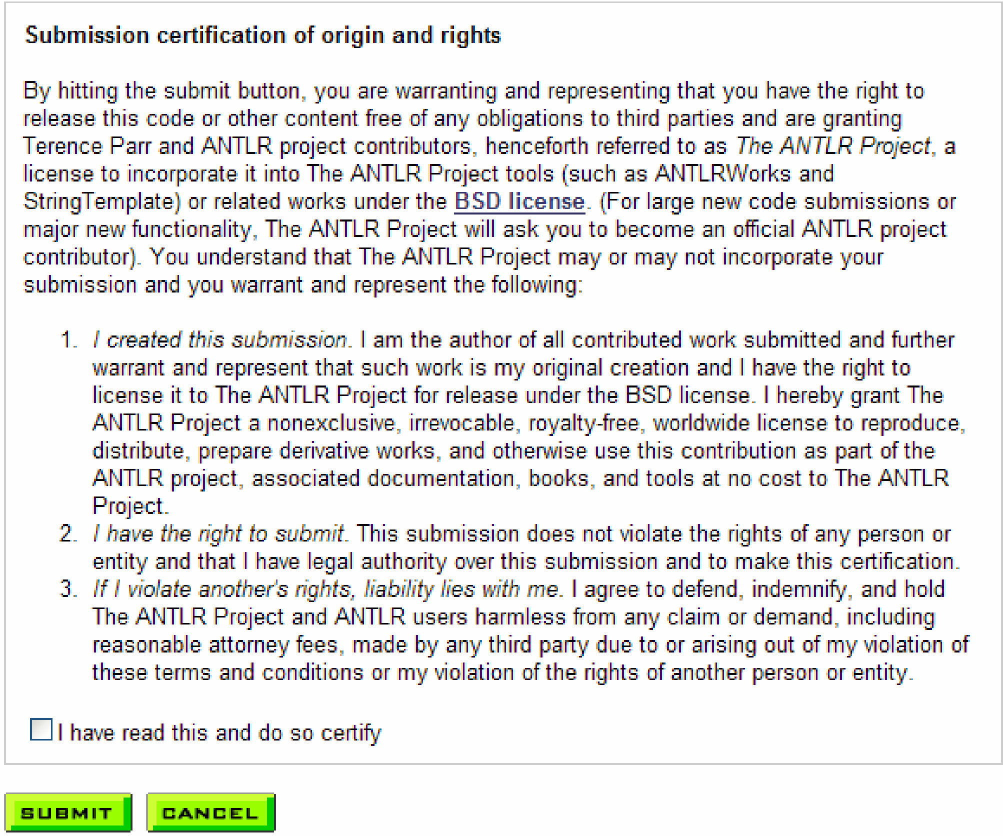
From ANTLR v3 and StringTemplate onwards, all substantial and/or active contributors must sign and fax or snailmail a copy of the ANTLR contributors certificate of origin formally agree to abide by it by signing on the bottom with the date. An email address and your full name must be included. Mail or fax to:

Terence Parr
University of San Francisco
2130 Fulton St, HR532
San Francisco, CA 94117
Fax: +1 415 422 5800

One-off contributions may be made through the feedback page.

ANTLR 3.0 Cont…

Smaller contributors agree to the BSD electronically



Submission certification of origin and rights

By hitting the submit button, you are warranting and representing that you have the right to release this code or other content free of any obligations to third parties and are granting Terence Parr and ANTLR project contributors, henceforth referred to as *The ANTLR Project*, a license to incorporate it into The ANTLR Project tools (such as ANTLRWorks and StringTemplate) or related works under the BSD license. (For large new code submissions or major new functionality, The ANTLR Project will ask you to become an official ANTLR project contributor). You understand that The ANTLR Project may or may not incorporate your submission and you warrant and represent the following:

1. *I created this submission*. I am the author of all contributed work submitted and further warrant and represent that such work is my original creation and I have the right to license it to The ANTLR Project for release under the BSD license. I hereby grant The ANTLR Project a nonexclusive, irrevocable, royalty-free, worldwide license to reproduce, distribute, prepare derivative works, and otherwise use this contribution as part of the ANTLR project, associated documentation, books, and tools at no cost to The ANTLR Project.
2. *I have the right to submit*. This submission does not violate the rights of any person or entity and that I have legal authority over this submission and to make this certification.
3. *If I violate another's rights, liability lies with me*. I agree to defend, indemnify, and hold The ANTLR Project and ANTLR users harmless from any claim or demand, including reasonable attorney fees, made by any third party due to or arising out of my violation of these terms and conditions or my violation of the rights of another person or entity.

□ I have read this and do so certify

License Suitability

- Consistency with Intended Use
- Objectionable Terms
- Terms that Present Potential Difficulties for Downstream Consumers
- Legal Incompatibility

License Compatibility

- When more than one “copyleft” open source product is used in the same application, the applicable licenses may contradict one another.
  - One license may require that the application as a whole be licensed under its terms; while another may require that it be licensed under its terms.
  - As a result, it may not be possible to comply with both licenses at the same time. The licenses are “incompatible.”

We Use Tools to Help Us

- The Eclipse Foundation uses tools to assist with our review.
  - Keyword search tools
  - Code print matching tools

Two Small Words…

"I started designing internet applications in [year], when I joined [***Technology Company***]…. Very soon, when working on ***customer projects***, I introduced the concept of [concept]…. I began developing a small generic framework …. I improved the framework as I moved on, from customer to customer. "

"Until it was time to **open source** it! … As I spread the word about this framework within [Technology Company], several … [***other employees***]… began using it and making modifications… Another good reason [to open source it] was that I wanted to have feedback from the open source community and wanted to get help to improve it. Also, it was good to be able to provide to our customers a framework that would continue to evolve and be maintained even after we left the project." [*Emphasis added. Paraphrased from:*

http://jakarta.apache.org/cactus/participating/contributors.html]

What Possible Issues are Raised?

Does the Technology Company have an ownership interest in the software?

“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.”

What Possible Issues are Raised?

Do Technology Company’s customers have an ownership interest in the software?

“Consultant acknowledges that all right, title and interest in and to any of the deliverables developed as a result of the Services (including but not limited to all patents, copyrights, trademarks and any other intellectual property rights therein) provided hereunder are and shall remain the property of Company, and all rights, title and interest therein shall vest in Company and shall be deemed a “work made for hire” within the meaning of the U.S. Copyright Act, 17 U..S.C. Section 101 et. seq. To the extent any of the deliverables are not deemed to be a “work made for hire” Consultant hereby assigns to Company all rights, title and interest to the deliverables. At the expense and request of Company, Consultant agrees to execute all documents and do all other acts necessary in order to enable Company to protect its rights in such tangible or intangible property developed or arising directly as a result of the performance of the Services.”

What Possible Issues are Raised?

Are there other authors involved and did they consent to distribute the code under the license identified?

- Do their employers have an interest in the code?
- Do their customers have an interest in the code?

Changes to Project License Terms

- Comfort with license terms vary
- Open Source Projects may change their license terms after a period of time.

Additional Licenses

- Additional Licenses may be found at the file level

Copied Material

- Is it re-licensed material?
  - Did the original license allow the re-licensing?
  - Is the license compatible?

- Have the terms of the original license been complied with?

Benefits to our Community

- Risk of legal liability is reduced.
- Risk of having code that is subject to restrictive terms included in the Eclipse code base is reduced.
- Risk of having to later remove code and re-work the code stack is also reduced.

Leveraging Completed Reviews

- Code is submitted for review via a separate instance of Bugzilla which we call IPZilla.
- Requests to use another project's code, concerns identified during review, specialized Eclipse distributions, approvals and rejections are all documented in a searchable database.
- The projects and select members can leverage code that had already been reviewed and approved.

Distributors – Planning Ahead

- Warranties, Representations, Indemnities
  - Either negotiate for them from a vendor or do without them

- Support – determine your needs and negotiate with a vendor if necessary.

- Permitted Use – if your intended use isn't permitted, consider negotiating a license that permits your use.

Active Management is Key

- Whether you are talking about proprietary software or open source software, it is important to know:
  - What you have
  - Where you have it
  - How it is being Used

- Know what open source licenses are applicable to all code that is used by your company
  - Your open source-savvy customers will want to know
  - Your prospective acquirers and investors will want to know

Active Management

- Set up a process to review any new additions (whether open source or not) and any proposed changes to existing use.
- Consider multiple levels of review dependent on the nature of the open source involved.

Active Management

- For all use of open source software (internal and external, copyleft and non-copyleft)
  - Comply with license’s attribution, documentation and notification requirements
  - Perform periodic audits to assure that open source code is being used consistently with its license terms
  - Educate management organization and staff

Questions?

Thank You!