Advanced Software Engineering FOSS

Mark Struberg

Kontakt: teaching@inso.tuwien.ac.at

About me

- Mark Struberg
- 25 years in the industry
- TU-Wien / INSO researcher
- Apache Software Foundation member
- struberg [at] apache.org
- Committer / PMC for Apache OpenWebBeans, MyFaces, TomEE, Maven, OpenJPA, BVal, Isis, DeltaSpike, JBoss Arquillian, ...
- Java JCP Expert Group member
- MicroProfile Spec Author
- Twitter: @struberg

Agenda

- General Considerations
- Legal basics
- OSS Licenses
- The ASF in numbers
- How the ASF works
- How to Contribute?

Safe Harbor Statement

- **I am NOT your lawyer!**
- **Anything explained in this talk might be totally wrong or only valid on a small island in the pacific...**
- **Always ask your own lawyer if something is unclear.**
- **Intention is to highlight potential problems.**

What interests me in FOSS?

- **FOSS: Free and Open Source Software**
- **Looking behind the scenes!**
  - Open Source is an easy way to look at how things *really* work
  - It's important to know the fundamental mechanism used in a tool to be able to judge side effects and solve problems the right way.
- **E.g.: How do virtual methods work?**
  - Method overriding via Virtual Function Pointer Table

General Considerations

The Weapon of Choice

- **"If you have a hammer, every problem seems to be a nail"**
- **"Use the right tool for the right job"**
- **Every design decision has pros and cons!**
  - There is no solution which perfcectly fits all your problems
  - Example: centralised vs de-centralised systems,
    App evolution in waves: HOST -> server/client PCs -> HTML webapps -> AJAX -> native phone apps -> microservices ->?

FOSS vs Closed Source

- **Pros and cons of commercial software tools**
- **Pros and cons of Open Source Software tools**
- **Big benefit of OSS if you hit a bug in a FOSS framework**
  - You don't need workarounds in your own code
  - Instead you can fix bugs directly in the OSS libraries you use
- **"Standard Software"**
- **Custom Development**
- **Project Development vs Product Development**
  - Developing a generic Product costs much more

"Copyright"

Copyright

- **The right to do whatever I want with a certain piece of work.**
- **Copyright actually consists of many rights.**
- **Difference between 'Urheberrecht' (non-dispositive, ius cogens) and 'Verwertungsrechte'**
- **in EU -> implicit**
- **in US -> rather explicit**

Threshold of Originality

- **ger: Schöpfungshöhe**
- **You only create originary IP if you**
  - wrote stuff yourself
  - it is not a trivial change
- **Obvious bug fixes do NOT constitute own IP**
- **Reformatting does NOT constitute own IP**
- **Nemo plus iuris transferre potest quam ipse habet!**

The Authorship

- **The author owns all the rights,**
- **except he is employed**
  - and does this in his payed time
  - or uses resources or company funded know-how
- **In those cases IP belongs to the employer!**
- **Different in the US!**
  - depending on the country even spare-time stuff might belong to the employer!

CLA

- **Contributor License Agreement**
- **Two different reasons**
  - Make the Contributor aware of the legal impact
  - Grant additional rights beyond the License
- **iCLA vs cCLA**

Who owns the code at all?

- **Especially important to clarify in a customer relationship**
- **Werkvertrag vs Arbeitsvertrag**

Code Provenance

- **Where does the code come from?**
- **This is utter important for big companies!**
  - e.g. if they get sued or to show prior art
- **Reason why ASF doesn't accept unverified pull requests from github for big changes**
- **ASF requires signed iCLA for non-trivial contributions**

Open Source Licenses

The License Situation

- **Commercial Licenses**
- **MIT / BSD**
- **GPL, LGPL**
- **ALv2**
- **Others**

What is a License?

- **Conditions under which a Licensee can get the rights to use the code**
- **Consensual**
- **Juristically not a contract, but close**
  - same intent, different handling
    (contract law vs copyright law in some countries)
- **If you don't follow all the terms which are defined in the license then you don't get any of the rights!**

Commercial Licenses

- **Hard to grok**
- **Usually bloated with exits and safety valves for the vendor**

MIT License

- X11 License
- Provided "as is", leave me alone...
- rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies

BSD License

- **Allows to copy, change, distribute**
  - in source
  - and binary form
- **Requires to keep the original Copyright Headers of the original files**
  - For binaries, the original Copyright Header must get listed/printed somewhere, e.g. on the 'About' page.
- **Requires of Berkley attribution:**
  - “This product includes software developed by the University of California, Berkeley and its contributors.”

GPLv2

- **GNU Public License Version 2**
  - https://www.gnu.org/licenses/gpl-2.0.html
- **strong copy-left: all changes to GPL software are also under GPL**
  - applies in case of static AND dynamic linking
  - does not apply for just 'using'
- **What is 'dynamic linking' exactly?**
  - unclear even amongst lawyers!
- **Distributing the results requires to also distribute the modified sources**

LGPL

- Library or Lesser GNU Public License
- GPL but allowed to use in dynamic linking
- Careful when shading in!

The Apache License v 2.0 (ALv2)

- **http://www.apache.org/licenses/LICENSE-2.0**
- **Liberal open source software license**
- **Business friendly**
- **Requires redistributing NOTICE file**
- **Includes Patent Grant**
- **Can be sub-licensed**
  - not to confuse with re-licensing, which is not allowed!
  - sub-licensing: allows adding your own code to the existing one in any license you like
  - re-licensing: would allow to change the license of all existing code

Not really OSS

- **do-no-evil-license**
- **beer-license**

The Github dilemma

- **What happens if no license is defined?**
- **Always add some License! (re github)**
  - If you don't add any license to your published sources then no one can use them because they have no rights.

Facebook BSD + FB Patents License

- **React**
- **RockDB**
- **Not OSI approved!**
- **ASF does not allow it in Apache projects**
  - for a good reason

Patents

Patents

- **Some licenses contain a 'patent grant'**
  - Apache License v 2.0 (ALv2)
  - GNU Public License v3.0 (GPLv3)
  - Mozilla Public License v1.1 (MPL)
- **Software Patents are allowed in the US**
  - at least it is handled that way right now
- **Software Patents are not allowed in the EU**
  - but they give a damn about that...
- **Patents often registered as defence weapon**

Trademarks

Trademarks

- **Choose your mark wisely!**
- **Name must be unique (in your field)**
  - google
  - trademarkia.org
  - EU trademark registration office

Trademarks

- **Actively defend your marks**
  - Marks vanish if they frequently get used by others without proper attribution
  - require attribution
    "Apache Foo is a trademark of the apache Software Foundation"
  - http://www.apache.org/foundation/marks/
- **Establish a mark without restricting the economy around your project**
  - e.g. the Apache Hadoop ecosystem
  - allow 'powered by...', '... for Apache Foo'

Ways to monetise OSS

- **Adding commercial available value on top of a base OSS offering**
  - RedHat
  - JBoss AS, IBM WebSphere
- **Professional Training**
- **Embedding OSS into hardware**
  - most routers use Linux, iptables, etc
  - Android
- **Service Contracts**
  - 'insurance' for your project
- **Sharing the costs - saving money!**
- **Project Consulting**
  - Help other companies to save money by using OSS.

What to avoid

- **Don't try to sell the same product you give away for free**
- **Respect the freedom!**
  - Don't force/restrict others to do something only because *you* need it
  - Respect the community
- **Don't rely (only) on a payroll**
  - don't push a project into one corner just because a customer likes it that way
- **OSS project planing is different than company projects**
- **Spread the influence across different companies**

Commercial impact of OSS

- **Most entertainment products nowadays use OSS inside**
  - Almost all modern TV
  - Almost all Blueray players
- **Android: ALv2**
- **Apple OSX/IOS (FreeBSD)**
- **Most Servers**
- **OpenJDK**
- **more than 50% of all software is OSS**

About the Apache Software Foundation

History of the ASF

- **1994 - bunch of hackers started The Apache WebServer**
- **1998 - first 'commercial' companies kicked in**
- **1999 - 'ASF' officially formed as foundation**
- **1999 - Apache Jakarta got started**
  - in corporation with SUN microsystems
  - first Java projects at ASF
  - JSP and Servlets RI
  - tomcat, ant, maven, struts, etc
- **2002 - Apache Incubator got started**
- **2004 - Apache License v 2.0**
- **2016 - 176 TLP, 5921 committers, ...**

Founding Principles

- **"The Foundation exists to serve the development community"** - Bill Stoddard
- **"Let developers focus on what they do best: Code. The foundation exists to do the rest"** - Justin Ehrenkrantz
- **Not-for-profit Organisation**
- **Volunteer Organisation**
- **Not affiliated with nor controlled by any government or commercial entity**

Mission Statement

- **Provide legal and technical infrastructure for open source software development and to perform appropriate oversight of such software.**
- **http://www.apache.org**
- **Apache projects are defined by**
  - collaborative consensus based processes,
  - an open, pragmatic software license and
  - a desire to create high quality software that leads the way in its field.
- **"Community over Code"**
- **"Community for Code"**

The Apache License v 2.0 (ALv2)

- **http://www.apache.org/licenses/LICENSE-2.0**
- **Liberal open source software license**
- **Business friendly**
- **Requires attribution**
- **Includes Patent Grant**
- **Can be sub-licensed**
  - not to confuse with re-licensing, which is not allowed!
  - sub-licensing: allows adding your own code to the existing one in any license you like
  - re-licensing: would allow to change the license of all existing code

ASF Structure

- Contributors
- Committers
- PMC Members
- ASF Members
- Board of Directors
- Public Relations, Infra, Legal advisors

Contributor

- Report bugs
- Share ideas
- Get feedback on the mailing lists
- Provide patches via JIRA
- Provide Code and new modules via patches
- ASF requires a signed iCLA for non-trivial changes

Committer

- **Merit must be earned!**
- **Public recognition for your existing contributions**
- **Write-access to your project SCM**
- **Own yourid@apache.org email address**
- **http://people.apache.org/~yourid/ web space**

Project Management Comittee Member

- **The PMC manages a TLP and provides the direction for the project**
- **Binding count in release votes and fundamental decisions**
- **Must provide legal oversight of the project**
- **Access to private project mailing list**

Apache Software Foundation Member

- Legal member of the non-for-profit organisation
- Right to elect the board and official positions
- Access to almost everything within the ASF
- Must get voted in by existing members

Board of Directors

- 9 members elected annually
- Official representation of the ASF
- Oversight and circuit breakers for PMCs

The Incubator

- **Own TLP which 'shepherds' new candidate projects**
- **Goal is to make those projects (code and people) 'mature' and TLPs on their own.**
- **Ensure all donations meet ASF technical standards**
- **Ensure all donations meet ASF legal standards**
  - provenance checks and IP clearance
- **Ensure that the community is diverse enough**
- **Incubation is a good chance to build up a broad community**
- **"Becoming an Apache project is a process, not just a decision" - Bertrand Delacrétaz**
- **http://incubator.apache.org**

How to contribute

- **Pick a project**
- **Become familiar with the topic**
- **Start reading the mailing list**
  - you can also read the archives via markmail, nabble, etc
  - http://lists.apache.org
- **Check out the Source Code**
- **Use the project!**
- **Start reporting bugs...**
- **... and ship patches.**

The 'Apache Way'

- **Goals**
  - Reduce barriers to project participation
  - Improve quality
  - Achieve consensus and resolve conflicts
  - Balance needs of corporate interest with needs of individual contributors

Questions?