How to make business with open source

St. Andrews 2012

Michael Widenius
MySQL & MariaDB founder
monty@askmonty.org
Eben Moglen, Software Freedom Law Center

"The license is the constitution for the community"
This talk is for

- Companies considering to use Open Source
- Companies considering to develop / go Open Source
- People wanting to create a company working with Open Source
- Officials from countries & business that want's to understand and benefit from Open Source
Key questions addressed

- What to consider before creating a company doing business with open source.
- What is an Open Source License?
- Why go Open Source?
- Why is it in your advantage to use Open Source?
- How to choose a License?
- What are the main popular Licenses in use, and what are their main differences?
- How to develop high quality open source products
- What kind of community creates high quality products?
Questions to consider when creating a company

- Do you plan for a virtual company (“no offices”)?
- Are you creating a company of equals? Should the company be owned by employees (Hacking business model)?
- Do you want to concentrate on services or development?
  > If development, what license to choose?
- Do you plan to have a big community or work with a few big companies?
- Do you plan to take in investors?
  > If yes, then you need an 'exit' plan.
What business model to choose?

- Service company
  - Man powered services (support, training, consulting)
  - Valuation 2 x revenue
- Software company
  - Licensing, Software as a service (SAS), Subscription
  - Valuation 10 x revenue + X * number of users
When to take in investors

Answer: **AS LATE AS POSSIBLE!!**

**Wait until** (if possible):

- You have a business model that works
  - You have a proven strategy to make money
- You have paying customers and a method to turn some community users to customers
- You have a large user/developer base
- You are profitable and don't critically need them!

The longer you can wait, the better deal you get!

Don't take more money than you need to prove yourself!
What is the final goal with the company

- Sold on the market
  > Unpredictable future for product/employees
  > Quick, often high, profit for owners
- Go public on stock market
  > Original owners can still be part of steering the company
  > Somewhat unsure profit (as owners can typically only sell after 6 months)
- Owned by founders, employees (and investors)
  > Stable predictable future
  > Owners gets dividends, employees gets bonus
- Create an open source foundation
Key features for a software company that is likely to succeed

- **Unique idea** that solves a common problem
- **Good team** to implement idea and drive business
- **Big user / developer base**
- **Right software license** (That gets at least some part of the users to pay)
- A proven method to **convert some users** to paying customers
What is an Open Source License?

...Is a Copyright License for Computer Software
...that makes the source code available
...for modification and redistribution
...free (gratis) of charge

However:

- It does NOT mean you can do what you want with the code
- But you can use it freely as long as you follow the *Terms of the License*
- “Libre” but not “Gratis”
  > You should give something back (support or donate)!
Why go Open Source?

- Spread the product more quickly *(more users)*
- Get some part of the development done elsewhere *(lower cost)*
- Get things more tested and more bug reports *(higher quality)*
- Possible to get development done in 'not business critical' directions *(more useful product)*
- Easier to find good developers, partners and customers

The above means more market recognition, feedback, leads, business, partners and sell opportunities and a strong trademark.

In general open source projects gets *more feedback* and *better bug reports* than closed source project.
Reasons for users to trust a open source vendor

- Open source vendors are more trustworthy as they depend on trust to survive.
- **No vendor lock in.** Your investment in using the product is safe even if:
  - Vendor goes out of business
  - Vendor would surprisingly change business terms radically
  - Vendor would stop supporting the version of the product you are using
  
  If this would happen to a popular product, someone would go away with the code and start maintain it themselves.
- Little risk for **hidden trap doors** as one can examine the product code.
Benefits for developers using open source vendor

- Easy to get access, look at and use the code
- Freedom to examine and change any part of the code to satisfy your business reasons, fix bugs or port to other systems.
- Freedom to find anyone to do the above
- Freedom to use (read, build and change) the code and redistribute the code in an open source environment.
Benefits for big business / countries using open source

- You can develop your own infrastructure to tune the software for your own means (language, unique requirements)
  > Facebook, Google
  > Brazil, Iceland
- You get internal knowhow you can utilize for business
- Not depending on an outside vendor (for minor things).
- No license costs; Low to very low cost of ownership!
- It's in your interest to collaborate with the original community for long term sustainability
  > A full fork is expensive in the long run!
When go Open Source?

IF you can create a sustainable business model around Open Source, proprietary vendors will have hard to compete with you

It's nowadays increasingly harder to sell closed source solutions in a lot of business sectors.

Nor surprisingly, it's hard to choose the right license for your software as it significantly impacts your strategy, i.e.  

- How you can co-operate with your user community
- How you can build your business
How to choose an OS license?

• Key Questions:

• What is your business idea around the open source product?
  > Services, subscriptions and/or licensing?

• What rights do you keep to your code when used, modified and potentially redistributed?

• What kind of community do you want around the product?
How to choose an OS license?

Do you want to allow others to:

- Do anything with your code?
- Use your code freely in any manner?
- Change your code without giving you access to the changes?
- Include and redistribute your code together with commercial, closed source code?
- Use your code together with other open source code?
- Release their modifications to your code under a different license?
What are the main popular Licenses in use, and what are their main differences?

<table>
<thead>
<tr>
<th>License</th>
<th>Author</th>
<th>Link from code with a different license</th>
<th>Release changes under a different license</th>
</tr>
</thead>
<tbody>
<tr>
<td>APL (Apache License)</td>
<td>Apache Software Foundation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>BSD license</td>
<td>Regents of the Univ. of California</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GPL (GNU General Public License)</td>
<td>Free Software Foundation</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LGPL (GNU Lesser General Public License)</td>
<td>Free Software Foundation</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MPL (Mozilla Public License)</td>
<td>Mozilla Foundation</td>
<td>Yes</td>
<td>Limited</td>
</tr>
<tr>
<td>Public domain</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SPL (Sun Public License)</td>
<td>Sun Microsystems</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Full list under http://en.wikipedia.org/wiki/Comparison_of_free_software_licences
The main open source licenses (Very simple view)

- **Public domain**
  - Gives user freedom to do anything, including changing your copyright and claim they wrote it.

- **BSD/Apache**
  - Gives user freedom to full use, but needs to keep copyright in the source code.

- **LGPL**
  - Gives user freedom to use freely, but if they distribute it changed, they need to publish the changes under LGP

- **GPL**
  - Gives user freedom to use it freely, but if they distribute it, they need to publish changes and their code under GPL

- **AGPL** (Optional addition to GPL V3)
  - Free usage, but users needs to publish the code and the code connected to it even if not distributed (like web server).
Business models to use with open source

- **Open-Core Model** - have an open-source core and sell closed-source features on top of it (e.g. SugarCRM)
- **Dual Licensing Model** - one product/project that gets licensed with a viral, GPL-style license and a commercial closed-source license (e.g. MySQL)
  > Another option is “Business source” or delayed Open Source
- **Services Models** - where you get to download a productized version of an open-source project and pay a fee for the support you get on it or the features you sponsor. You can normally also pay for training, features etc
- **Subscriptions** (Usually a combination of support, extended product lifetime and guaranteed updates)
Open Core

- Probably most popular way nowadays for business trying to do open source first time.
- What Oracle is doing with MySQL and EnterpriseDB with PostgreSQL
- Is not an open source business model, because it uses closed components and most of the benefits open source developers expect from the product is gone:
  - You can't change, modify, port or redistribute the code
  - You are locked to one vendor
- You may be able to create a small developer community around the product but mainly by people that doesn't need the closed source extensions.
- For community developers, the “worst” possible offer is open core or closed source that used under subscription and stops working when subscription runs out
Business source

- Not an open source license, but gives the community similar advantages.
- Source code is available, anyone has the right to copy, modify & distribute but can't use it commercially under some conditions you define.
- After X years the code automatically converts to some open source/free license. The date has to be explicitly set in all source code files to avoid misunderstandings.
- Better than Open Core as this removes the one vendor problem and the code will eventually be free.
So, How to decide on licensing?

You need to provide the right balance of freedom:

- Being too restrictive will harm Community growth
  > Here you compete with companies like Microsoft
- Being too permissive will harm Business growth
  > Here you compete with projects like Wikipedia

Going “All-In” on either, can be tough in the competitive market of today if you are not unique

Try to create a business model, which enables you to grow a business, but is as free as possible to help you create a large Community
The importance of selling licensing

- MySQL would never have been possible without licenses.
- Very hard to get companies to pay for support or development
  - Companies expect to use open source for free
  - Exception is companies you work with to develop the open source project
- Subscriptions are good, but also hard to sell
- Licensing is “free money” for the project
  - You need one entity that holds the copyright to the whole project or use SAS (Software as a service)
  - You need to be able to dual license your project
  - Your project is an infrastructure project that is usually embedded into others.
Conclusions (why use Open Source software)

- No big change; Everyone is already using Open Source
- Safer to use than closed source software
  - You are in charge of your own destiny (More freedom)
- Easier to get access to, test, use and deploy
- Less bugs, no hidden traps, more resources
  - More users & developers -> More tested & secure
- Less expensive in the long run
  - No unexpected charges, no forced upgrades
  - Doesn't get old; Development happens elsewhere
- Better supported
  - Both commercial support and community support
  - Can get real support from original developers & many vendors
The end