

Dual license? Open core? Subscription?...

Legal aspects of FOSS based business

malcolm bain

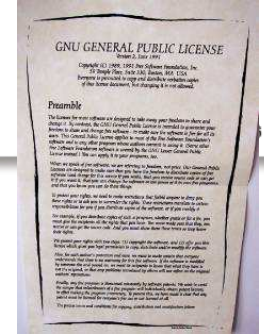
lawyer

www.id-lawpartners.com

www.brugueras.com

- Intellectual property rights (IPR)
- Software patents (really)
- Free software licenses
- Copyleft
- GPL or AGPL

And I am not going to make you read one of these...



GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

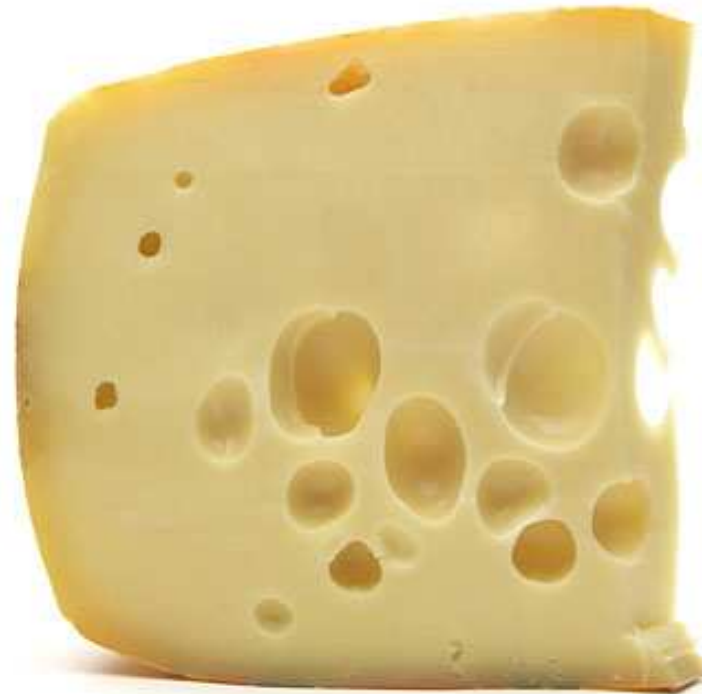
To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

AND 6 MORE PAGES

What I am going to talk about



And some of this



And this



And a bit of this



Courtesy of Ilchester Cheese Company Ltd



So....

Just a quick reminder....



- Inflexible
- Restrictive
- Pre-built



© Esteban



© JaulaDeArdilla



© JaulaDeArdilla

“Do what you want with it”



Usually comes with high level of service... or no service at all

Today's question: OPEN SOURCE **BUSINESS**

- Key words...

– ADDED VALUE

– COMPETITIVE ADVANTAGE

– VALUE PROPOSITION

¿EH? ¿QUE ES ESO?



¿QUE ES ESO?

ESO ES QUESO!



Gracias, Martin y Lara!



Today's question:

What does **open source** bring as **added value** and **competitive advantage** (i.e. what is the value proposition) for IT?

i.e. ..the **CHEESY** bits of technology.



© Stephen Hamilton

FOSS is

- Freaky
- Geeky
- Sneaky
- ...



- No money / returns
 - If it is free, it must be free
- Not safe
 - If it is open, it is not safe/secure
- Not stable
 - If it is run by a community, it can't be stable
- No quality control
 - Ditto: who controls the quality of the product?
- No support
 - Ditto: where is the support?
- Not fair (!!??)
 - Publically funded – intrusion in a competitive market
- No big users
 - All fringe stuff...
- Anti IPR
 - FOSS is “hostile” to IPR – you have to give up your IP
- Not sustainable
 - No long term plan and continuity (anyone can leave the project)

SO WHERE IS THE **VALUE PROPOSITION**???

Today's question:

What does open source **licensing** bring as added value and competitive advantage for FOSS based business?
(what is the **legal** value proposition)

i.e. the **LEGAL** cheesy bits...

1. Bugs
2. Security holes
3. Support coverage
4. Product life-cycle
5. Over sell, under deliver
6. Interoperability (and legacy systems)
7. Usability
8. Standardisation
9. Privacy
10. Vendor Lock-in

1. Bugs
2. Security holes
3. Support coverage
4. Product life-cycle
5. Over sell, under deliver
6. Interoperability (and legacy systems)
7. Usability
8. Standardisation
9. Privacy
10. Vendor Lock-in

Oh... and...

1. Project delivery
2. License fees
3. Use restrictions
4. Vendor audits
5. Efficient providers
6. Updated documentation

NO PROBLEM...

Legal added value:

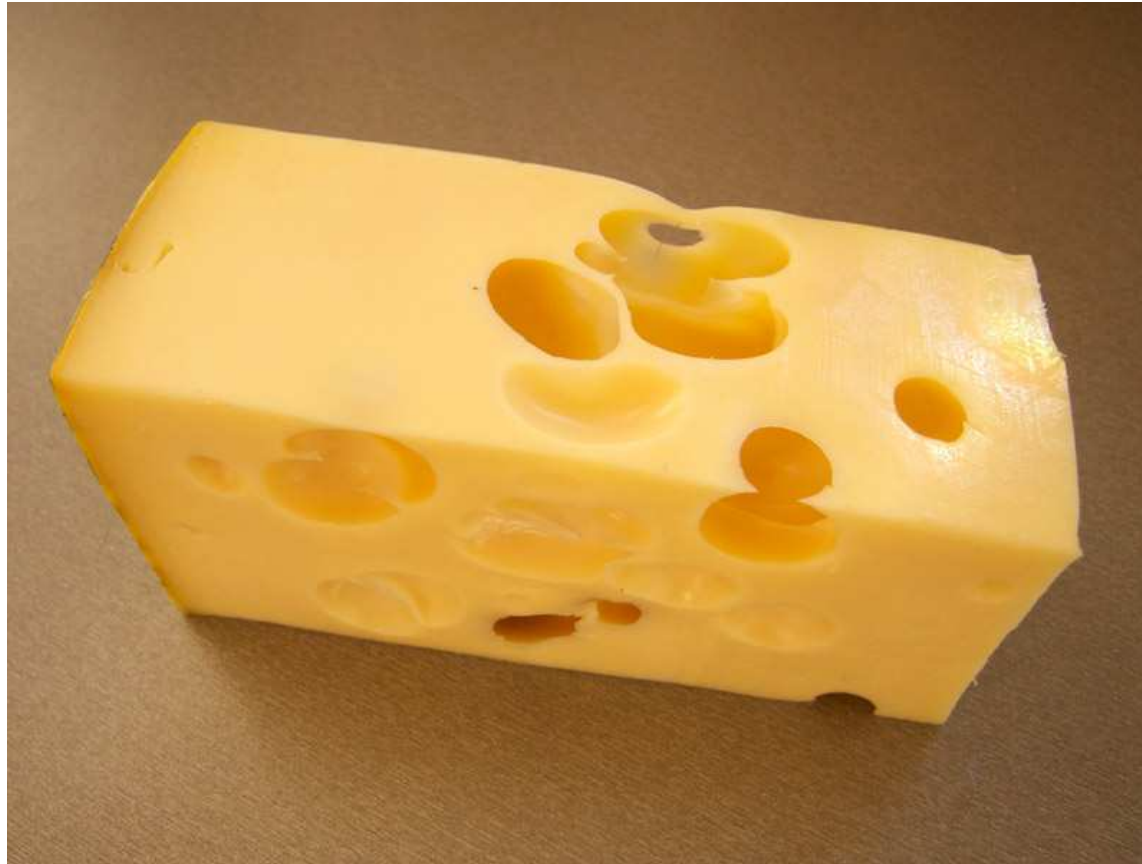
- Open source license grants you **access to the source code**
 - you can “physically” fix the bugs yourself
- Open source license gives you **rights to modify the code.**
 - You can “legally” fix the bugs yourself
- Open source license gives you **rights to distribute the code**
 - To share bug fixes across a community
- Open source license imposes **no confidentiality obligations**
 - You can report and ask the community to fix a bug

Community added value

- In fact... the community may fix the bug of itself



2. Security holes



It's made to have holes in it
AND
It's MANDATED to have holes in it

THREAT LEVEL | miscellaneous

FBI Admits It Controlled Tor Servers Behind Mass Malware Attack

BY KEVIN POULSEN 09.13.13 4:17 PM

Share



Photo: Andrew Hart/Flickr

It wasn't ever seriously in doubt, but the FBI yesterday acknowledged that it secretly took control of Freedom Hosting last July, days before the servers of the largest provider of ultra-anonymous hosting were found to be serving custom malware designed to identify visitors.

FOLLOW THREAT LEVEL



MOST RECENT WIRED POSTS



Lightening An Airplane's Load



Long Exiled, Tokyo's Indie Game Makers See Signs of Breakthrough



Keep Those Societies Secret



When Twitter Makes You Cry: Writer Eulogizes Mother in Front of 40K Followers



Make Your Embarrassing Old Facebook Posts Unsearchable With This Quick Tweak



This Amazing Magnetic Pen Will Take Hours From Your Life

Your open source alternative





LOOK: NO HOLES!

Well, not big ones, at least

3. Support coverage

NO PROBLEM...

Legal added value:

- Open source license enables you **access to source code**
 - To support yourself
 - To get a local IT supplier to support you
- Open source license gives you **rights to modify the code**
 - To support yourself
 - To get a local IT supplier to support you
- Open source license gives you **rights to distribute the code**
 - To share support across a community



4. Product life-cycle

NO PROBLEM...

Legal added value:

- Open source license provides **unlimited license term**
 - To keep on using, updating and fixing the code so long as it works
- Open source license enables you **access to source code**
 - To keep on using, updating and fixing the code so long as it works
- Open source license gives you **rights to modify the code**
 - To keep on using, updating and fixing the code so long as it works

**NICE OLD STINKY LINUX 2.0 →
Vintage 1996
(the older the better!)**



5. Over sell, under deliver

NO PROBLEM...



Legal added value:

- Open source license provides **unlimited use right**
 - To test and evaluate the code before deploying (and buying professional support)
- Open source license enables you **access to source code**
 - To analyse why the code doesn't do what the sales guys said it does
- Open source license gives you **rights to modify the code**
 - To make the code actually do what the sales guys said it does

Tasty (testy) bites! →



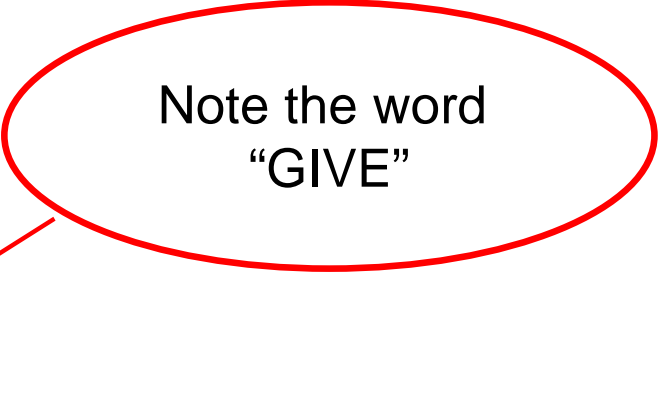
NO PROBLEM...

Legal added value:

- Open source license allows you to **modify the code**
 - To adapt it to user needs



- Standardisation
- Privacy
- Vendor Lock-in
- Limited use rights....



Note the word
“GIVE”

NO PROBLEM

- Open source license gives you unlimited license term
- Open source license gives you unlimited use right
- Open source license gives you access to source code
- Open source license gives you rights to modify the code
- Open source license gives you rights to distribute the code

NO MORE CHEESE PICTURES PLEASE!

- No bugs at all
 - but you can see them and get them fixed
- Efficient project teams
 - but you can get them trained or train them yourself
- Updated documentation
 - but you can write some yourself
- Projects delivered on time
 - Oops, sorry, no remedy for this one! (well, you could get a lawyer...)

To summarise...



[Platform Policies](#)[Supplemental Materials](#)**Open Source** >[Documentation & Code L...](#)

Open Source

Introduction

Facebook has been developed from the ground up using open source software. Developers building with Platform scale their own applications using many of the same infrastructure technologies that power Facebook.

Want to learn more? Check out [Engineering at Facebook](#).

Platform

Our Platform engineering team has released and maintains open source SDKs for **Android**, **iOS**, **JavaScript** and **PHP**.

Infrastructure

Apache Cassandra is a distributed storage system for managing structured data that is designed to scale to a very large size across many commodity servers, with no single point of failure.

Apache Hive is data warehouse infrastructure built on top of Hadoop that provides tools to enable easy data summarization, adhoc querying and analysis of large datasets.

FlashCache is a general purpose writeback block cache for Linux. It was developed as a loadable Linux kernel module, using the Device Mapper and sits below the filesystem.

HipHop for PHP transforms PHP source code into highly optimized C++. HipHop offers large performance gains and was developed over the past two years.

Open Compute Project an open hardware project aims to accelerate data center and server innovation while increasing computing efficiency through collaboration on relevant best practices and technical specifications.

Scribe is a scalable service for aggregating log data streamed in real time from a large number of servers.

Thrift provides a framework for scalable cross-language services development in C++, Java, Python, PHP, and Ruby.

Tornado is a relatively simple, non-blocking web server framework written in Python. It is designed to handle thousands of simultaneous connections, making it ideal for real-time Web services.

Our engineers contribute to

Apache Hadoop provides reliable, scalable, distributed computing infrastructure which we use for data analysis.

Apache HBase is a distributed, versioned, column-oriented data store built on top of the Hadoop Distributed Filesystem.

Cfengine is a rule-based configuration system that is used to automate the config and maintenance of servers. Facebook uses Cfengine to maintain host configs and to automate many janitorial operations on our production tiers.

jemalloc is a memory allocator which is fast, consistant, and supports heap profiling. Facebook engineers added heap profiling and made many optimizations. [Learn more](#).

memcached is a distributed memory object caching system. Memcached was not originally developed at Facebook, but we have become the largest user of the technology.

MySQL is the backbone of our database infrastructure. You can find our patches on [Launchpad](#) and learn more about how we use it on the [MySQL@Facebook page](#).

PHP is an incredibly popular scripting language which makes up the majority of our code-base. Its simple syntax lets us move fast and iterate on products.

Varnish serves billions of requests every day to Facebook users around the world. Whenever you load photos and profile pictures of your friends, there's a very good chance that Varnish is involved.



[Home](#)

Open Source

Introduction

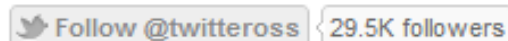
Twitter is built on open source software, from the back-end to the front-end. Twitter engineers [use](#), contribute to and [release](#) a lot of open source software. We of the Twitter Open Source Program Office support a variety of open source [organizations](#) and are grateful to the open source community for their contributions, and want to maintain our healthy, reciprocal relationship.

If you have questions or problems, please tweet us at [@twitteross](#). For media inquiries, please go [here](#).

Development

If you're interested in the projects we have released, check out our official [organization](#) on GitHub.

If you want to stay up to date, please follow our official twitter account.



Like what we do? Then consider [finding a job](#) you're passionate about and come [@JoinTheFlock!](#)

Code and Documentation Licensing

The majority of open source software exclusively developed by Twitter is licensed under the liberal terms of the [Apache License, Version 2.0](#). The documentation is generally available under the [Creative Commons Attribution 3.0 Unported License](#). In the end, you are free to use, modify and distribute any documentation, source code or examples within our open source projects as long as you adhere to the licensing conditions present within the projects.

And another ...

The Google logo is centered on the page. It features the word "Google" in its signature multi-colored font: blue for 'G', red for 'o', yellow for 'o', blue for 'g', green for 'l', and red for 'e'. A small trademark symbol (TM) is located at the top right of the letter 'e'. The logo has a slight drop shadow, giving it a three-dimensional appearance.

“What are the most important open-source projects you ingest?”

... Obviously the Linux kernel is incredibly important. Every time you use Google, you're using a machine running the Linux kernel.

The [Linux] kernel, compilers--GCC, the Python interpreter. Python is very important to us. Google App Engine--it's a Python hosting system, basically. Java is very important to us, and that's become open-source now.Once you get past those three projects--the compilers, the languages, the kernel--then you go to the libraries. For us that's OpenSSL, zlib, PCRE. MySQL is hugely important to us. Past that, it starts tapering off pretty quick

Chris DiBona, Google Open Source Program Manager in “Q&A: Google's open-source balancing act”, CNET, May 28, 2008

אנדרואיד

open source project

Home

Source

Compatibility

Tech Info

Community

About

Welcome to Android

Here you can find the information and source code you need to build an Android-compatible device.

Android is an open-source software stack for mobile devices, and a corresponding open-source project led by Google. We created Android in response to our own experiences launching mobile apps. We wanted to make sure that there was no central point of failure, so that no industry player can restrict or control the innovations of any other. That's why we created Android, and made its source code open.

[Learn more »](#)



News

Source Code Available for Android 4.0

The source code for the Android 4.0 platform and software stack has been released! This release allows OEMs to begin preparing Android 4.0 for installation on new and existing devices, and allows hobbyists, enthusiasts, and researchers to develop custom builds. For information on how to obtain the software, visit our [Getting the Source](#) page.

Compatibility Definition for Android 2.3

The Compatibility Definition Document for Android 2.3.3 has been published. Android 2.3 allows device manufacturers to use the Android source code to ship a significantly wider variety of devices, including devices with extra-large screens, such as tablets. Android 2.3.3 adds enhanced Near-Field Communications support to the Android APIs. For more information, visit the [Compatibility](#) page.

Source



If you're interested in contributing to the Android source code or helping out with the open-source project, our Source pages have the information you need.

[Get Involved »](#)

Compatibility



If you're an organization building an Android device, you'll want to check out our Compatibility pages to find out how to take advantage of the benefits of compatibility.

[Get Compatible »](#)



[Project Home](#) [Wiki](#) [Issues](#) [Source](#)

Search Current pages for

[Whats New?](#)
[Getting Started](#)
[Join a project](#)
[Project Hosting FAQ](#)
[Making hosting better](#)

Wiki
 Issues
 Downloads
 Source
 Administer

Getting Started

A quick guide to getting started with project hosting on Google Code.
[Featured](#), [Restrict-AddWikiComment-Commit](#)

Updated May 16, 2012 by augie@google.com

Getting Started

Project Hosting on Google Code provides a free collaborative development environment for open source projects. Each project comes with its own member controls, Subversion/Mercurial/Git repository, issue tracker, wiki pages, and downloads section. Our project hosting service is simple, fast, reliable, and scalable, so that you can focus on your own open source development.

This guide provides information on:

- [Getting Started](#)
 - [Contributing Open Source Code](#)
 - [Working with your Project](#)
 - [Customizing your Project](#)
 - [Working with your Source Repository](#)
 - [Documenting your Project](#)
 - [Tracking Project Tasks](#)
 - [Sharing Releases](#)
 - [Finding Further Help](#)

Contributing Open Source Code

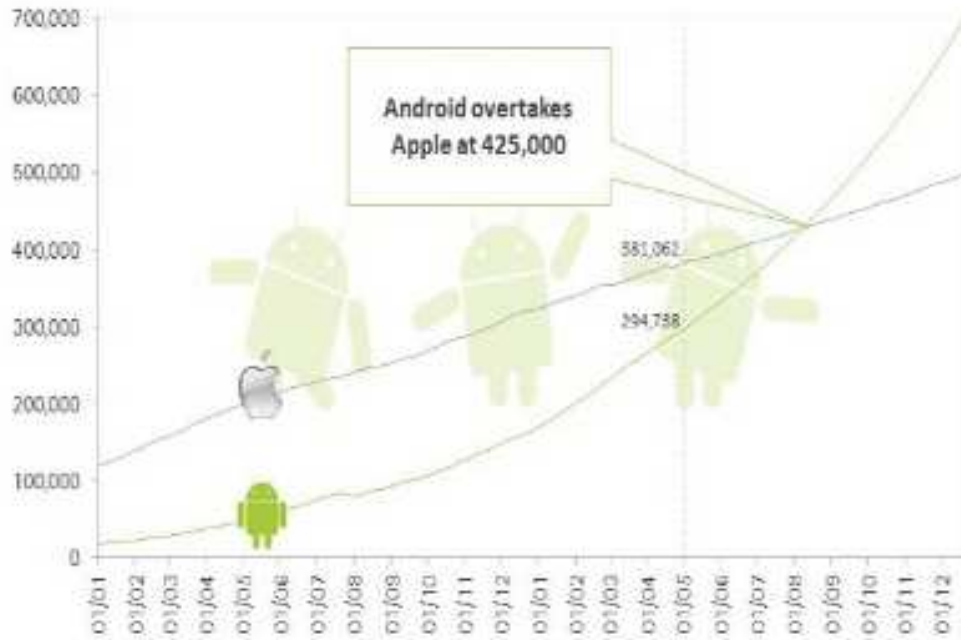
Before you create a new project, please [search](#) for existing projects on this site and elsewhere on the web. It is better to help out with an existing project than to start one from scratch.

- Go to the [Project Hosting main page](#) and search for existing projects ([How to join a project](#))
- Go to the [Create Project](#) page and fill out the form. Remember, pick good labels (see [Project Hosting main page](#) for examples).
 - [How to choose a license](#).
 - [How to choose a version control system](#).

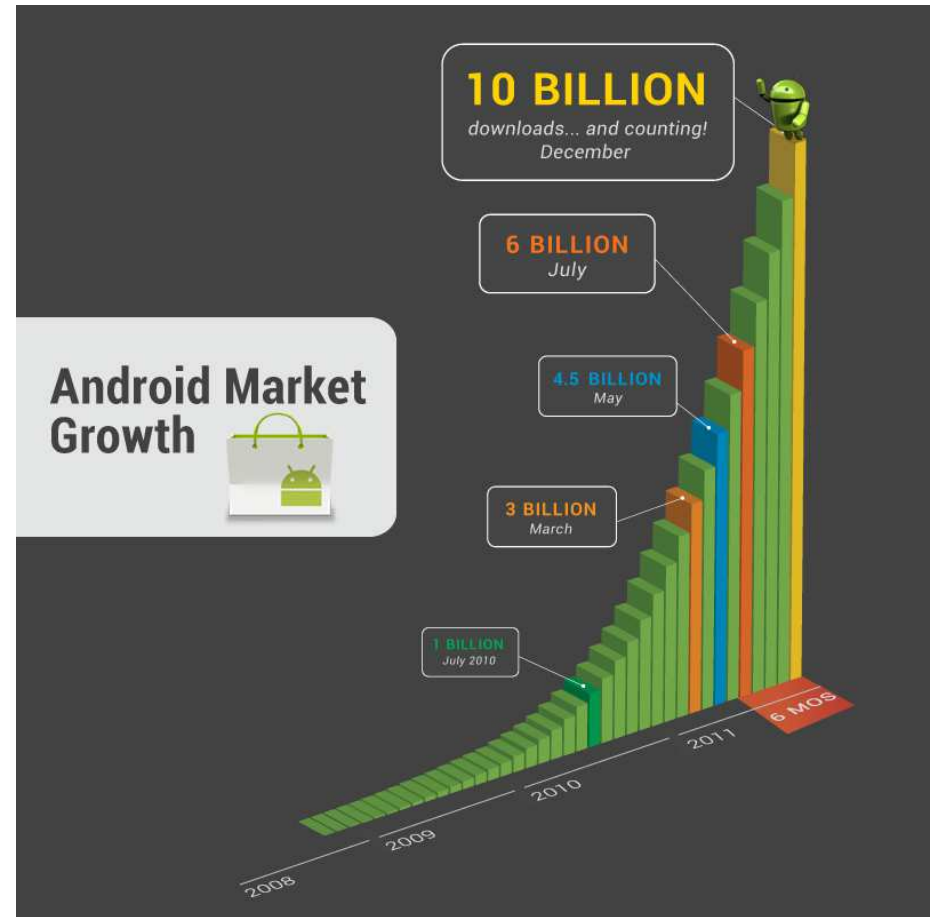
Working with your Project

Smartphones / Apps

Number of apps in Apple App Store and Android Market (01/2010 – 12/2011E)



research2guidance
the mobile research specialists



So how do you do it, then?

¿Where is the “pasta”?



1. Dual licensing
 - same code, two licenses (copyleft/commercial)
 - MySQL, Funambul, Sleepycat
2. Open core
 - Two codes, two licenses: a basic FLOSS software and a commercial version / proprietary extensions
 - SugarCRM, Pentaho, Alfresco, Openbravo
3. Subscription: extensions, additional services
 - Selling a service, monthly or annual fees for access to continuous updates of an OSS product
 - Openbravo, Pentaho, Alfresco, etc
4. Product Specialist
 - Expert skills on single/several products
 - Red Hat/Fedora, Canonical/Ubuntu, Novell/Suse
5. Consulting and integration services
 - IBM, HP, Accenture, etc.
6. Cloud services
 - hosted SugarCRM/Openbravo/Zimbra
7. Platform providers
 - Rackspace, HP, Spikesource,

1. Dual Strategy

- Software company offers
 - free use of its software with some legal limitations, or
 - for a fee, commercial distribution rights (and a larger set of features)

- MySQL, Funambul, Sleepycat



2. “Opencore”



- A basic FLOSS software and a commercial version / proprietary extensions
- SugarCRM, Alfresco, Openbravo, Eucalyptus

3. Subscription Strategy

- Selling software as a service in charging the customer with monthly or annual fees for gaining access to continuous updates of an OSS product
- Openbravo, Pentaho, Sugar,
- Alfresco, Red Hat, etc



4. Product Specialist

- Revenues from services – both maintenance and consulting - “best code here” and “best knowledge here”
- Red Hat, Ubuntu, Suse,
- Boundless, Geosolutions



© Colorpix.be

5. Consulting / Services Strategy

- Integration consulting for open source software; pure service model, where the basic functionality costs nothing, and all the money is in customization
 - Consulting
 - Customisation
 - Support
 - Training
 - maintenance
- IBM, Novell, etc.
- Boundless, CamptoCamp, Alkante,
- Envitia, Mapgears...



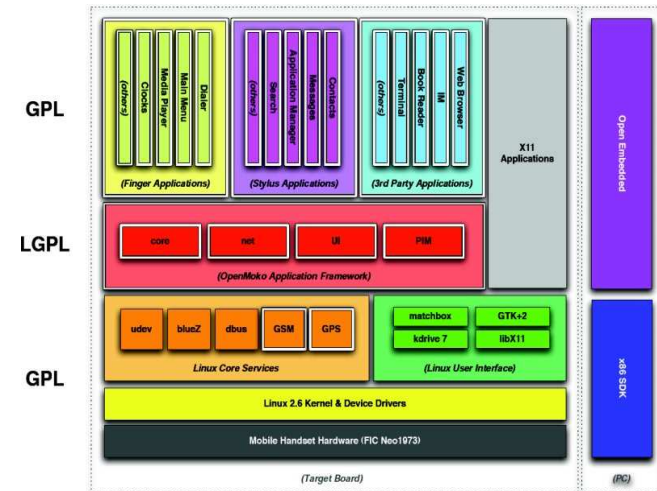
6. Hosted Strategy

- They don't sell their software, they let you use it or rent it
- Zimbra providers, Amazon, Google, OpenXchange
- CartoDB, Mapbox



7. Platform providers

- Selection, support, integration and services on a set of projects, collectively forming a tested and verified platform
- **Openstack, Spikesource, Redhat, Jboss**



9. Legal services



We live in a world that is designed, built and run by Open Source software.

In fact, we would find it hard to live today without it.

We are of course also mapping and viewing it with Open Source software 😊

And there are also a number of ways making a living out of it (somehow...)



Thank you for you attention

Malcolm Bain

malcolm.bain@id-lawpartners.com

mbain@brugueras.com