

Open Source Software Project Development

Dr. T.Y. Wong

Week 1

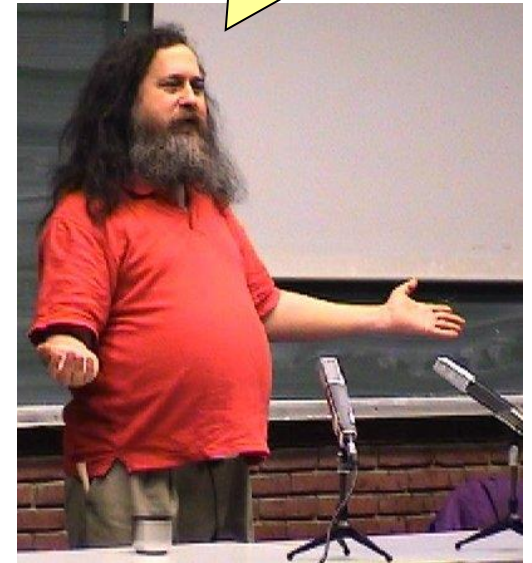
Introduction to Open-Source Development

Remember, it is not the same as “opening” your source...

What is a free software

- Free software == free-of-charge?
- Not exactly...
 - Every software should bear a license.
 - A free software bears a free software license.

"Free software is
as in free speech,
not in free beer."



**Richard Stallman -
the iconic spokesman of free software movement,
the father of GNU software suite.**

Why do we need a license?

- The fundamental purpose of a license, e.g., license on a book, is to:
 - protect the originality of the work;
 - protect the inventor's right;
 - the deny anybody the right to exclusively exploit the work.
- Free software license and open-source license carry different properties.

What is a free software?

Stallman defines what “free” is!



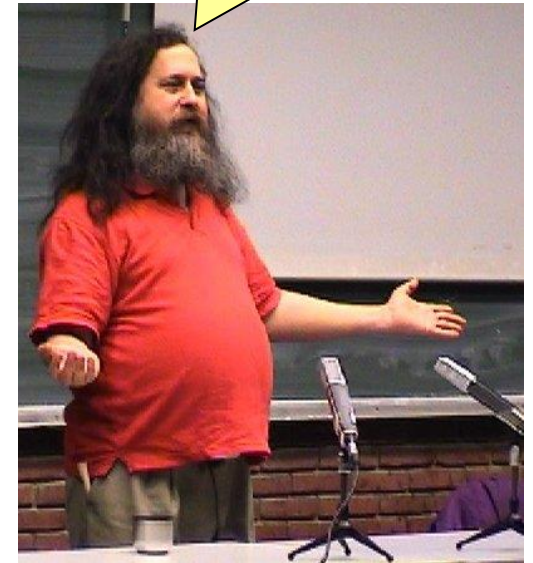
Speech itself does not carry any monetary value. But, the action may take value.

E.g., Clinton’s speech takes several millions HKD. But, the speech itself may be ...



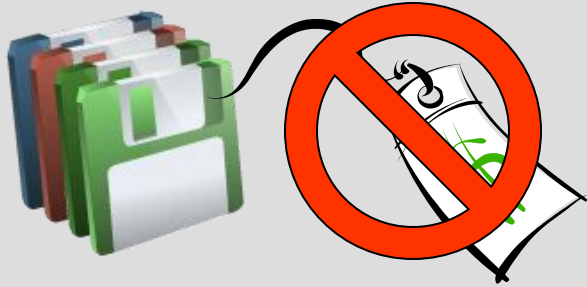
Beer itself has monetary value. But, you may be able to get it for free.

“Free software is as in free speech, not in free beer.”



What is a free software?

Stallman's Utopia

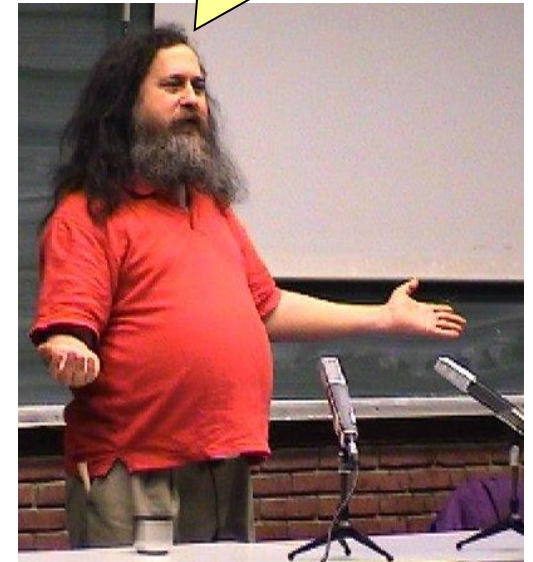


The price tag on the software should be removed. Programmers should give it out free-of-charge.



I make a living by giving out free software. **In return, I receive donations.** It is great. Why not joining us?

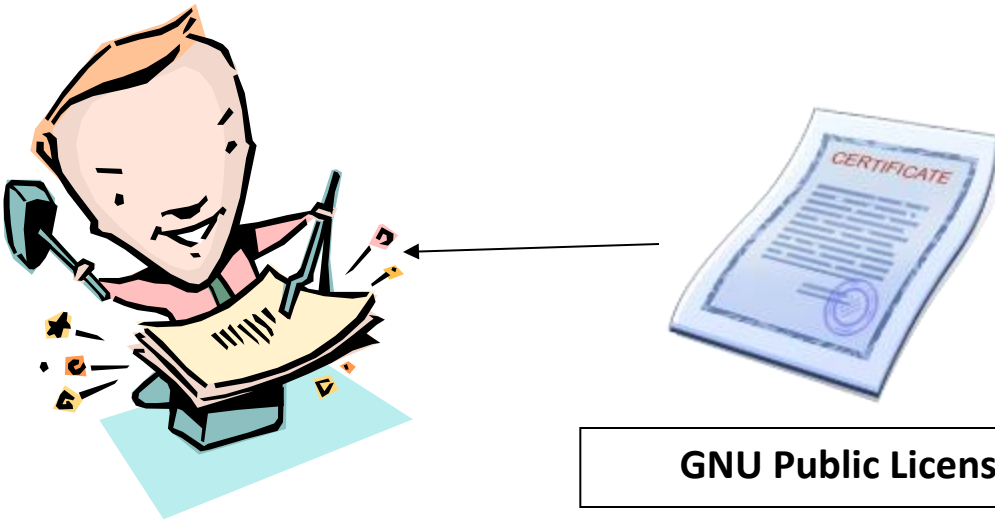
"Free software is as in free speech, not in free beer."



Source: <http://www.gnu.org/software/global/model.html>

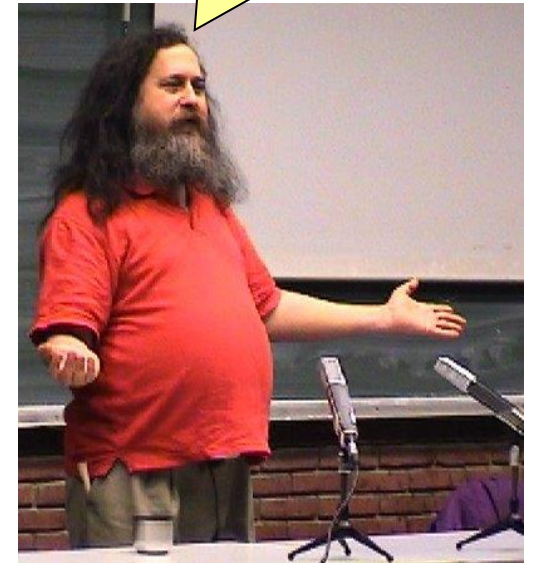
What is a free software?

- The free software license says:
any users can free to...
 - use, study, copy, modify, and redistribute the computer software.



GNU Public License - GPL

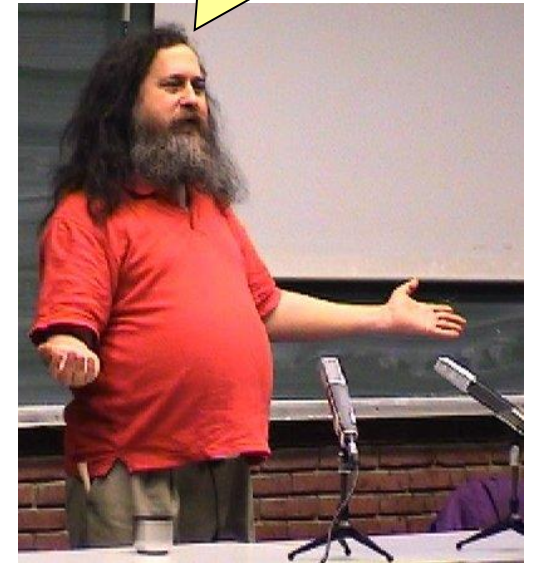
"Free software is
as in free speech,
not in free beer."



What is a free software?

- The free software license also says:
 - A user is free to modify the source code.
 - When the user distributes **the modified software**, **the new software is automatically carries GPL**.
 - This property is called “**copyleft**”, as it is opposing “*copyright*”.

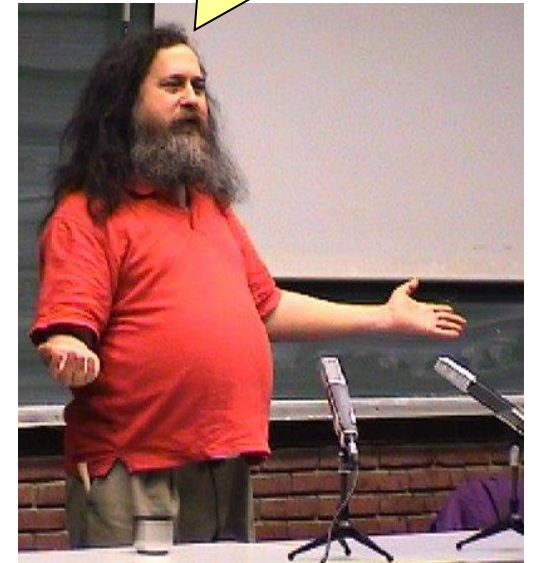
“Free, but with one and only one restriction...”



What is a free software?

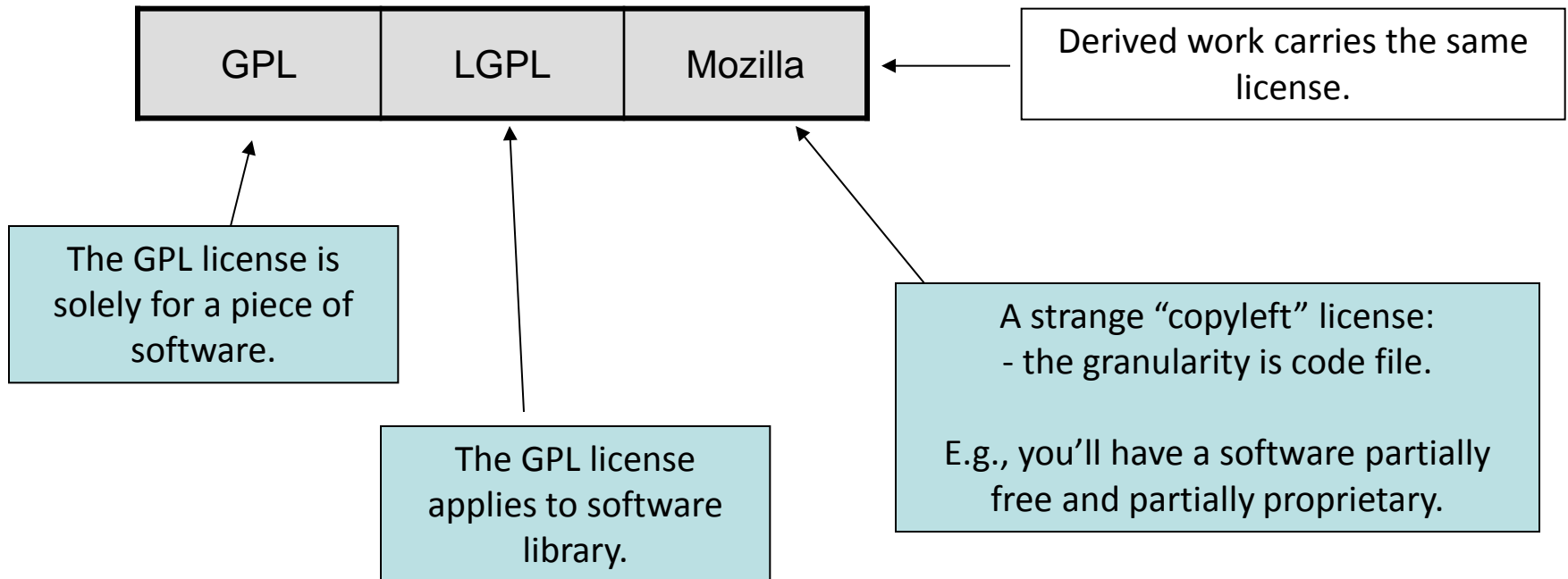
- Stallman's action was to initially oppose UNIX... because
 - there is a price tag on UNIX.
- He was **so sad** about commercial software: you own the physical copy of the software, but you are bound...
 - not to copy the work;
 - not to make derivative works based on the work; and
 - not to authorize anyone else to do either of the above two things.
- Have you ever heard of the **EULA, end-user license agreement?**

"Software should be free!! "



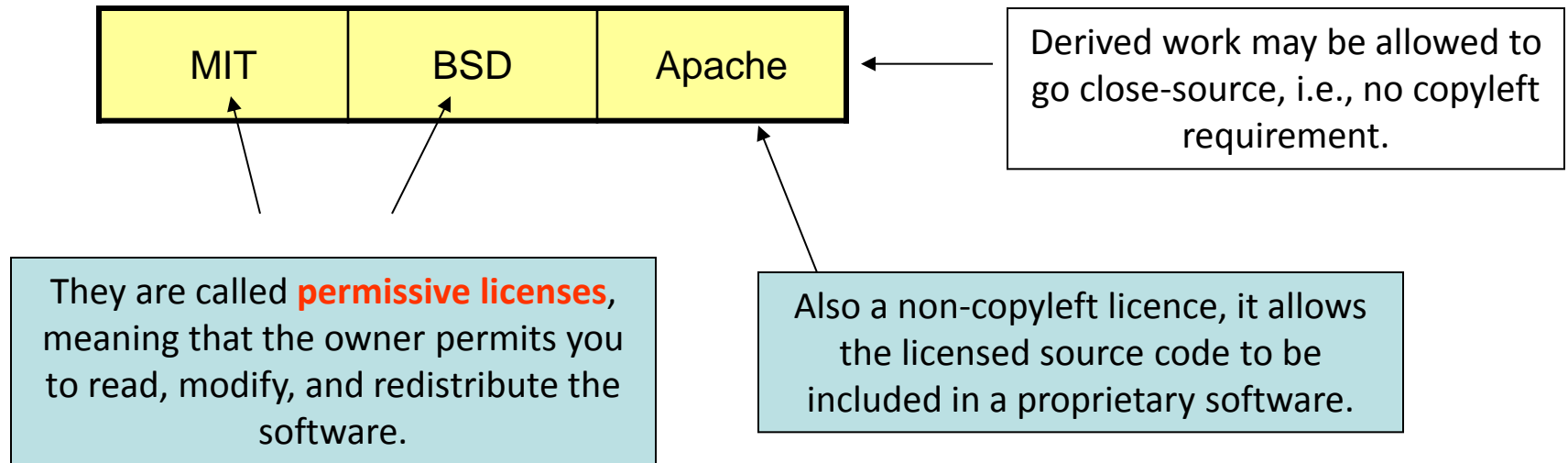
What is an open-source software?

- Open-source software == free software?
- Not exactly.
 - Depends on which license the software bears.



What is an open-source software?

- Open-source software == free software?
- Not exactly.
 - Depends on which license the software bears.



Android and Apache License: <http://www.thepowerbase.com/2012/11/android-community-demands-miui-rom-comply-with-foss-licenses/>

Traditional Software Development Cycle

- Traditional software development cycle:
 - Programmers work in the office.
 - They have regular meetings to brainstorm ideas, scolded by bosses, etc.
 - They have the manager to check on their progress.

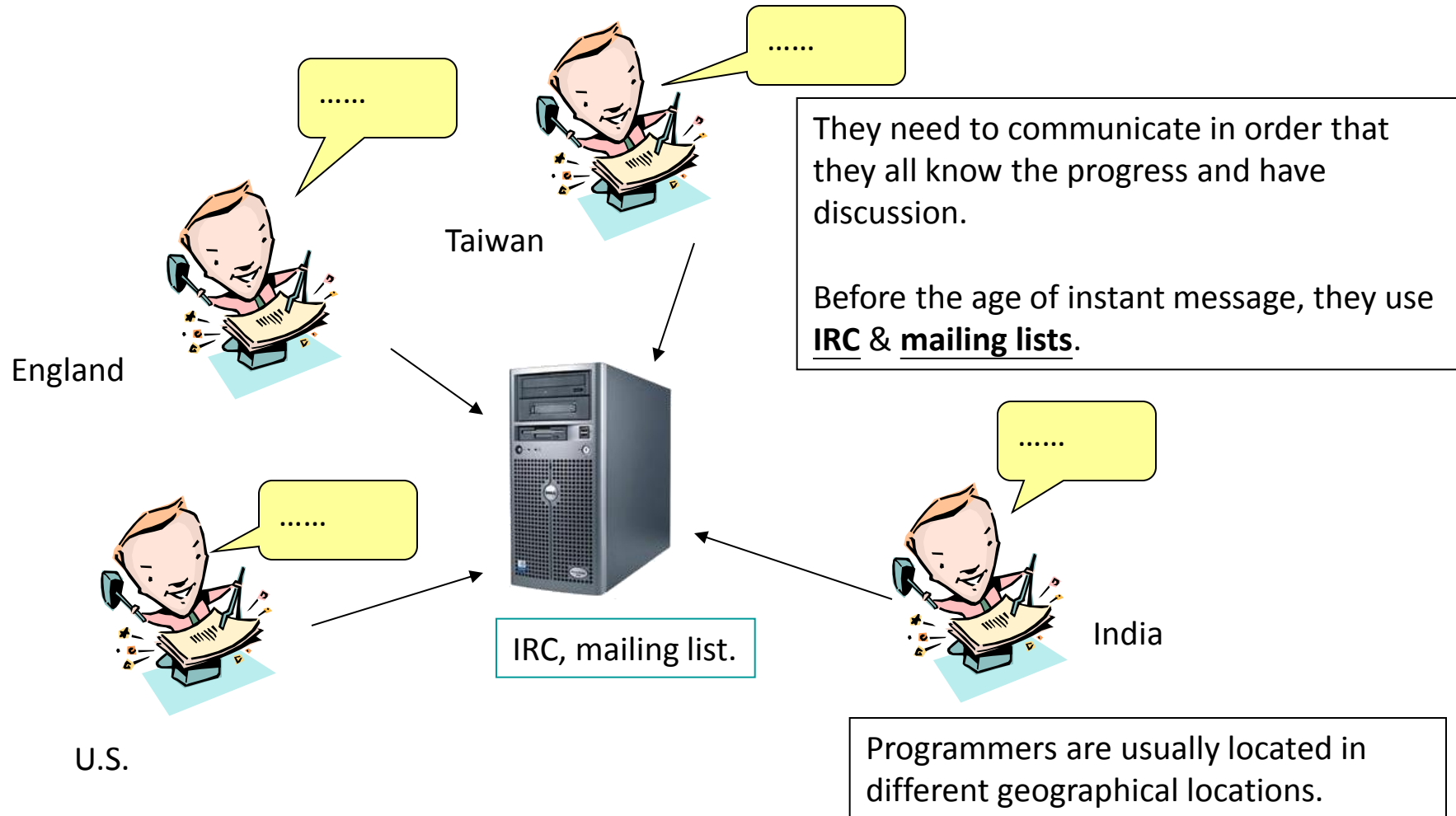


Open-Source Software Development Cycle

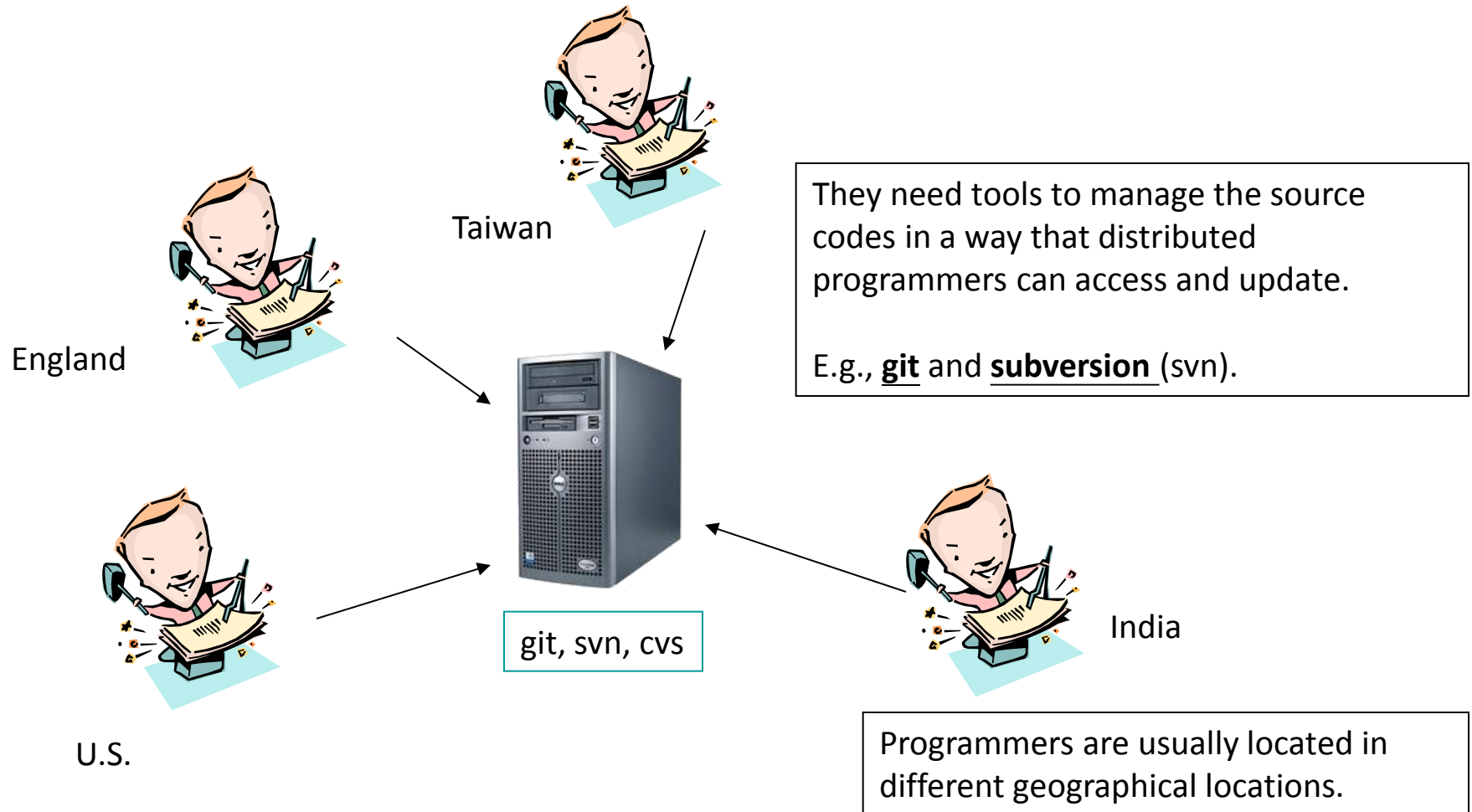
- Open-source software development cycle:
 - Office? Yes, programmers usually have a regular job. However, they work on the open-source software at home.
 - Meeting? Yes, but on the Internet.
 - Manager? Yes, instead, he/she is called the lead programmer(s) or the project founder(s).



Open-Source Software Development Cycle



Open-Source Software Development Cycle



Pros and Cons of Open-Source

- Pros #1: innovation.
 - There are great people who:
 - Don't ask for any reward;
 - Just want to contribute to the world;
 - Make the (computer) world a better place;
 - E.g., Linus Torvalds wrote his Intel-based Minix-like kernel just because he wanted to contribute to the community.

Pros and Cons of Open-Source

- Pros #2: reliability.
 - There are large number of people who:
 - are interested in existing open-source projects; and
 - are willing to devote time to make the projects better;
 - Such a number is too big comparing to commercial software companies...
 - because every (enabled) user can help to fix bugs since the software is open-source.
 - Close-source software cannot entertain from such a benefit.

Pros and Cons of Open-Source

- Pros #3: longevity.
 - A commercial software may be abandoned by its publisher.
 - E.g., Microsoft Corp. wants to discontinue the Windows XP in order to boost the adoption rate of Windows Vista (and failed).
 - Not the case for open-source. Community plays an influential role.
 - Enthusiastic programmers will continue previous work.
 - E.g., “**micq**” project is continued even though its founder died in a car accident.
 - E.g., “**ReiserFS**” has stopped because the author killed his wife!

Pros and Cons of Open-Source

- Cons #1: lack of support
 - Suppose you bought the MS Windows Server 2008 and, of course, its genuine license.
 - When you encounter any problems...a phone call to Microsoft can definitely help you.
 - Suppose you downloaded the Ubuntu 11.04 and you encounter configuration problems...then...
 - “RT..M”, “Ask the doc”, “Ask the so-called expert”, or “Ask google” are the only ways out.
 - [very important] don’t ask Dr. Wong!

Pros and Cons of Open-Source

- Cons #2: Licensing problems
 - Open-source software with GPL is problematic to commercial companies.
 - E.g., I think that **software A, which is under GPL**, is great, but it is not great enough.
 - Then, **I modify it** and it is perfect now.
 - Can I sell the new software A' and make a profit?
 - Yes...but, you have to **open the source**...
 - Then, everyone knows how you implement it.
 - Then, why does someone need to pay?

Brief open-source software history

- Pre-dotcom era...

Start of free
software
movement.

GNU tool chain:
gcc, make, etc.



Linux kernel
using GPL



(close-source)

Dark age:
proprietary software
ruled the world

1983

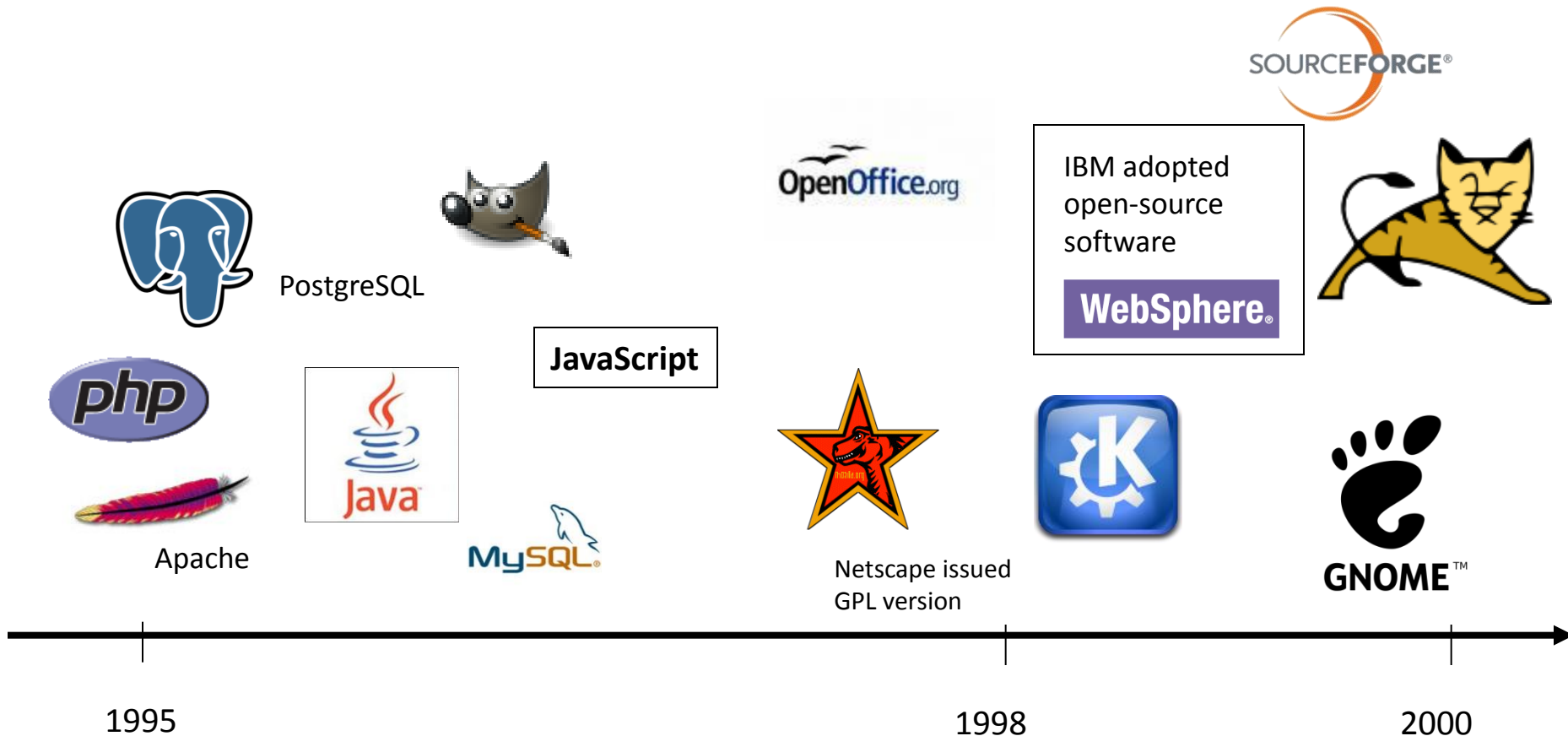
1991

1993

1994

(Not drawn to scale)

Brief open-source software history



(Not drawn to scale)

Brief open-source software history

- Dot-com bubble bursted...

SOURCEfire®



targets on
enterprise users.



Xen™

MySQL®

Release both proprietary and
open-source versions.



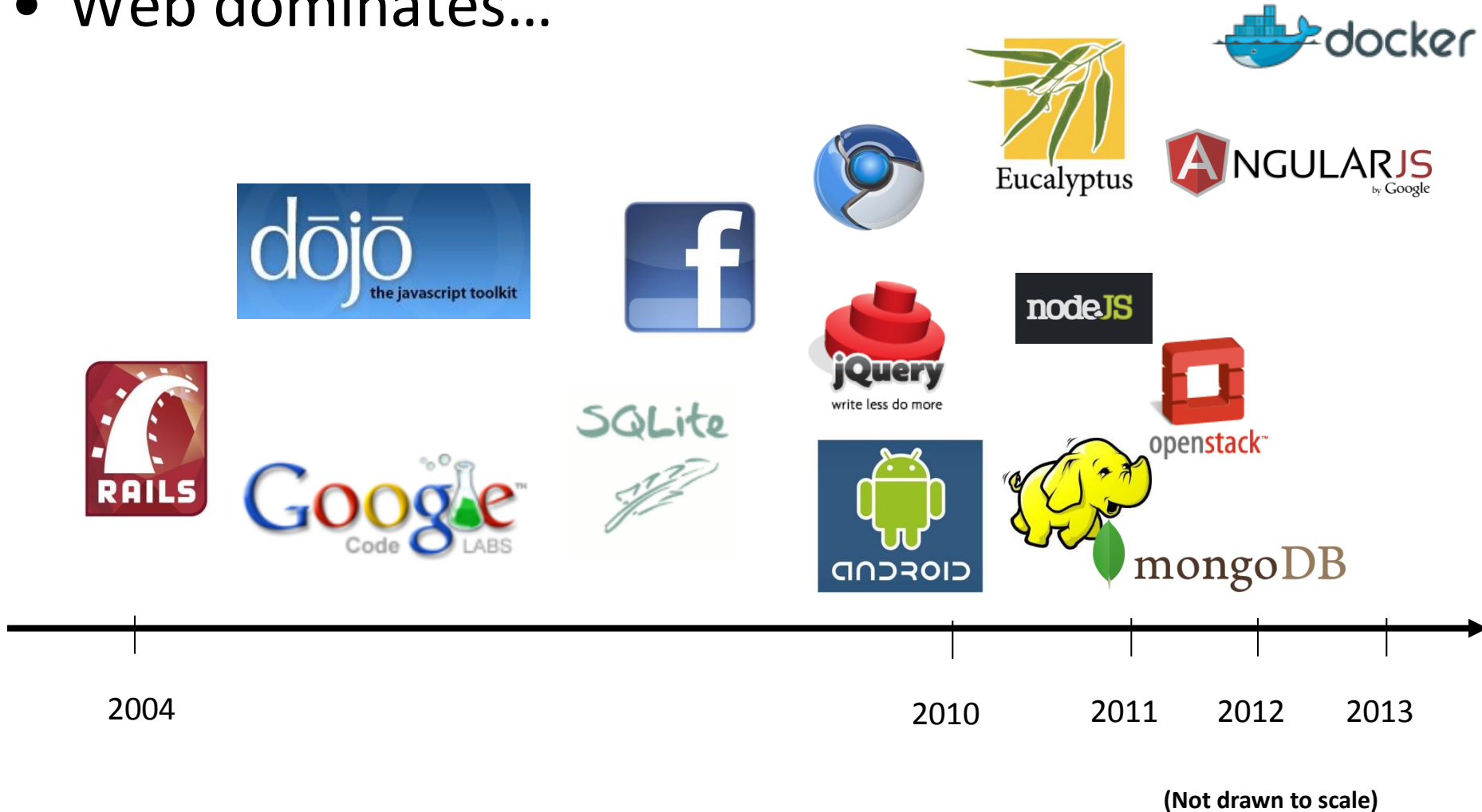
2000

2004

(Not drawn to scale)

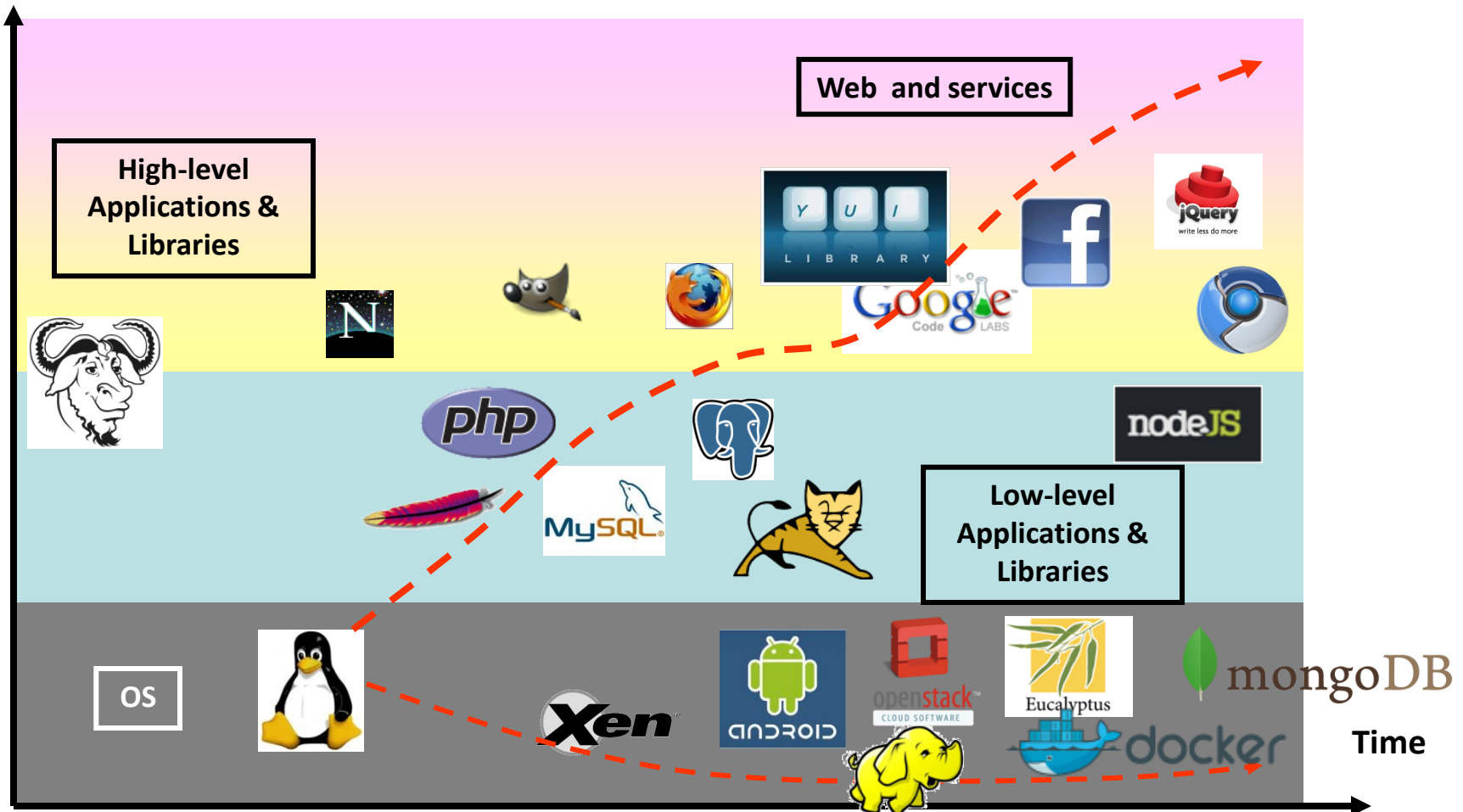
Brief open-source software history

- Web dominates...



Trends in Open-Source Software

Software



Trends in Open-Source Software

- Most of the open-source software is to **make your life easier**.

Category	Examples
System-level software	GNU toolchain, Linux kernel
Server software	Apache (Web), MySQL (database), Samba (Windows file sharing), etc.
Interface built to replace CLI, non-user-friendly software	GNOME vs Console, phpMyadmin vs mysql client program.
Proprietary counterparts	OpenOffice vs MS Office, CodeBlocks vs Visual Studio, GIMP vs Photoshop.

Trends in Open-Source Software

- Our course will focus on...
 - how to develop robust and efficient **software**.
 - how to use the **web browser** as an ubiquitous, versatile interface to ugly programs.
- Our future of OSS (instructor's opinion): **technology around the browser and the web!**

<u>JavaScript dominates.</u> <ul style="list-style-type: none">- more script libraries will come;- fancy layout with CSS3;- “MEAN” stack emerges;	<u>Browser, security, and OS merge!</u> (Not just Chrome OS, MS is also working on its own one.)
<u>Web Socket (in HTML5) replaces BSD socket.</u> <ul style="list-style-type: none">- socket.io, www.firebase.com, www.pusher.com	<u>Server-side software and multi-core programming merge!</u> <ul style="list-style-type: none">-boosting the degree of parallelisms.-Node.js

Your Role!

Act before the trend takes you.

Be the trend!