

Il software libero

Alberto Cammozzo

mmzz@stat.unipd.it

System manager, Università di Padova
coordinatore progetto ReFun, Pluto

presentazione per gli studenti del
corso di Basi Dati 2
Facoltà di Scienze Statistiche,
Università di Padova

8 Novembre 2004

Sommario

- Introduzione: libero/proprietario
- Storia e storie del software
- Software e Libertà
- Modelli di diffusione del software
- Modelli di produzione del software
- Modelli di licenze: GPL e le altre
- Aspetti socio-politici

Sorgente

Eseguibile

```
main()
{
    puts("hello world");
    exit();
}
```

```
457f 464c 0101 0001 0000 0000 0000 0000 0001 0003 0001 0000
0000 0000 0000 0000 0108 0000 0000 0000 0034 0000 0000 0028
000b 0008 0000 0000 0000 0000 0000 0000 8955 83e5 08ec c483
68f4 0000 0000 fce8 ffff 83ff 10c4 fce8 ffff 90ff 748d 0026
c3c9 b48d 0026 0000 8d00 27bc 0000 0000 0008 0000 0000 0000
0001 0000 3130 302e 0031 0000 6568 6c6c 206f 6f77 6c72 0a64
0000 4347 3a43 2820 4e47 2955 3220 392e 2e35 2034 3032 3130
3031 3230 2820 6544 6962 6e61 7020 6572 6572 656c 7361 2965
0000 732e 6d79 6174 0062 732e 7274 6174 0062 732e 7368 7274
6174 0062 722e 6c65 742e 7865 0074 642e 7461 0061 622e 7373
2e00 6f6e 6574 2e00 6f72 6164 6174 2e00 6f63 6d6d 6e65 0074
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 001f 0000 0001 0000
0006 0000 0000 0000 0040 0000 0030 0000 0000 0000 0000 0000
0010 0000 0000 0000 001b 0000 0009 0000 0000 0000 0000 0000
03a8 0000 0018 0000 0009 0000 0001 0000 0004 0000 0008 0000
0025 0000 0001 0000 0003 0000 0000 0000 0070 0000 0000 0000
0000 0000 0000 0000 0004 0000 0000 0000 002b 0000 0008 0000
0003 0000 0000 0000 0070 0000 0000 0000 0000 0000 0000 0000
0004 0000 0000 0000 0030 0000 0007 0000 0000 0000 0000 0000
0070 0000 0014 0000 0000 0000 0000 0000 0001 0000 0000 0000
0036 0000 0001 0000 0002 0000 0000 0000 0084 0000 000d 0000
0000 0000 0000 0000 0001 0000 0000 0000 003e 0000 0001 0000
0000 0000 0000 0000 0091 0000 0030 0000 0000 0000 0000 0000
0001 0000 0000 0000 0011 0000 0003 0000 0000 0000 0000 0000
00c1 0000 0047 0000 0000 0000 0000 0000 0001 0000 0000 0000
0001 0000 0002 0000 0000 0000 0000 0000 02c0 0000 00c0 0000
000a 0000 0009 0000 0004 0000 0010 0000 0009 0000 0003 0000
0000 0000 0000 0000 0380 0000 0027 0000 0000 0000 0000 0000
0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0001 0000 0000 0000 0000 0000 0004 fff1 0000 0000 0000 0000
0000 0000 0003 0001 0000 0000 0000 0000 0000 0000 0003 0003
0000 0000 0000 0000 0000 0000 0003 0004 0009 0000 0000 0000
0000 0000 0000 0001 0000 0000 0000 0000 0000 0000 0003 0006
0000 0000 0000 0000 0000 0000 0003 0005 0000 0000 0000 0000
0000 0000 0003 0007 0018 0000 0000 0000 0022 0000 0012 0001
001d 0000 0000 0000 0000 0000 0010 0000 0022 0000 0000 0000
0000 0000 0010 0000 6800 6c65 6f6c 632e 6700 6363 5f32 6f63
-----
```

Il codice sorgente
viene scritto dal
programmatore, poi
viene tradotto in
codice eseguibile.

Codice sorgente (*source*)



- Chiuso o proprietario:
 - Il programma è:
 - × eseguibile a certe condizioni
 - × sorgente non leggibile
 - × non modificabile
 - × non ridistribuibile
- Libero o *free* o *open*:
 - Il programma è:
 - ✓ eseguibile liberamente
 - ✓ sorgente leggibile
 - ✓ modificabile
 - ✓ ridistribuibile

C'era una volta un programma...

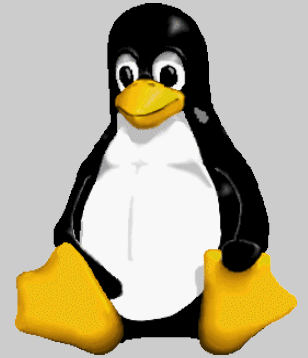
- Sui primi computer:
 - software era distribuito in sorgente,
 - condiviso tra chi lo scriveva,
 - in una comunità di utenti/sviluppatori.
- Con la diffusione dei minicomputer e dei PC
 - software era venduto a parte (UNIX, DOS, CP/M,...),
 - solo in forma eseguibile per evitare concorrenza,
 - a degli utenti isolati.



Prima storia: GNU (*Gnu is Not Unix*)

- Richard Marshall Stallman (rms), MIT, 1982
- Nella chiusura del codice vede una aggressione alla libertà: fonda la Free Software Foundation.
- Sceglie di scrivere da zero un S.O. Unix-like, ma completamente libero: GNU project.
 - *Tools*: Emacs('84), gcc('85), tar, ld, make, gawk, ...
 - *Kernel*: GNU Hurd: è più difficile del previsto e blocca il progetto.
- Per mantenere libero il software: licenza GPL.
- “*Make the world a better place*”.

Seconda storia: Linux



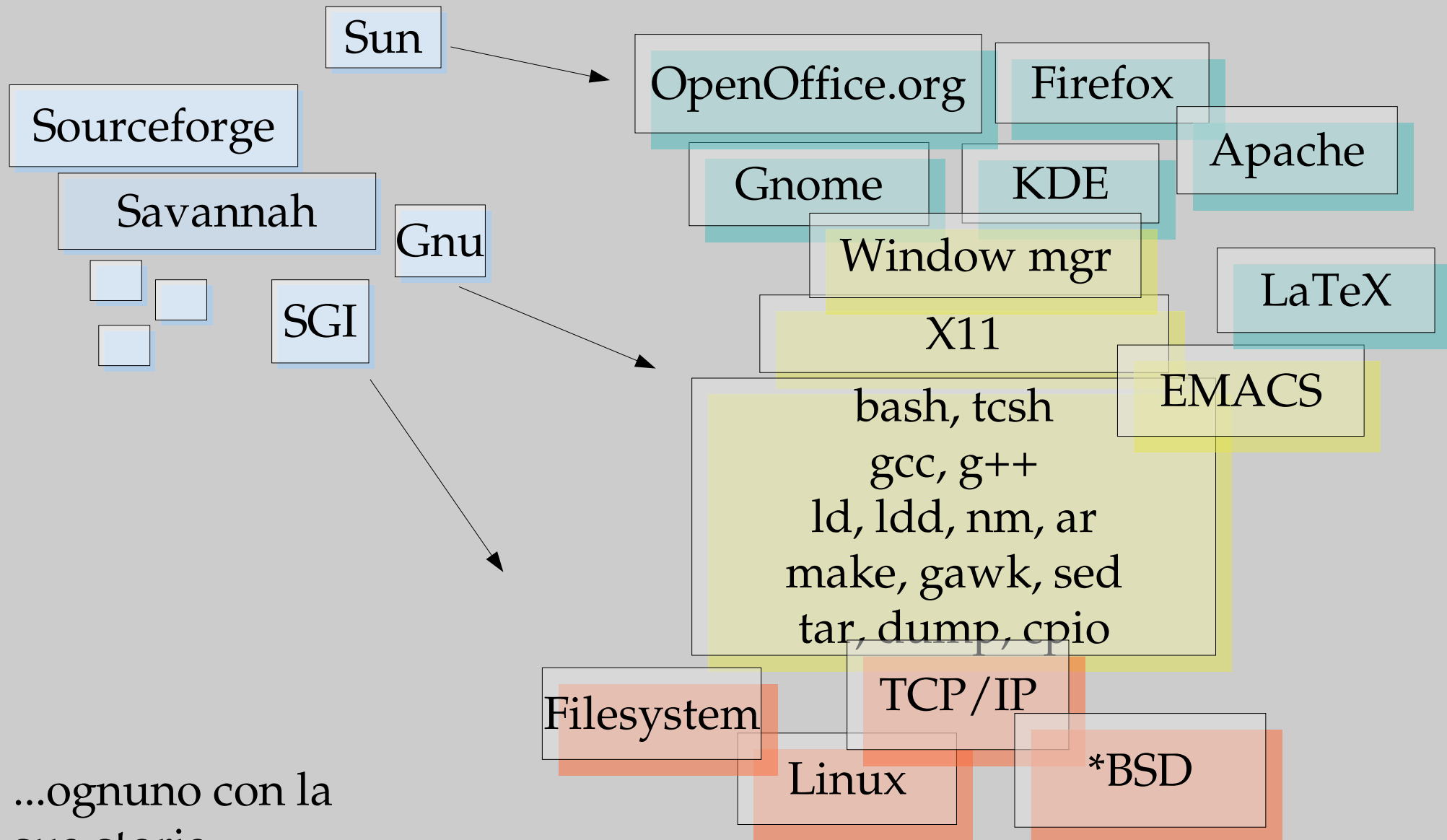
- Linus Torvalds, Helsinki University, 1991
- Pochi soldi, non può comprarsi uno Unix: ispirato da Minix decide di farselo da solo.
- Trova tutti i *tool* GNU già pronti e li usa per sviluppare l'embrione del *kernel*.
- Trova in Internet una comunità di programmatori pronti a ampliare, estendere e correggere il *kernel*.
- GNU/Linux: completamento del progetto GNU.
- 1994-... Altri programmi ne ampliano il successo.

Terza storia: Open Source



- Il successo di Linux e degli altri programmi liberi attira (e spaventa) il mercato in pieno boom *' .com '*.
- *Free* (che suona *gratis*) è un termine tabù per il business: *Open Source* è il suo *marketing term*.
- Pragmatismo, meno idealismo: fare software migliore qualitativamente.
- Imprese affermate liberano i loro prodotti: OpenOffice.org (Sun), Mozilla (Netscape).
- Occasione di concorrenza a Microsoft.
- Ammette la coesistenza proprietario/libero

Mappa di un sistema libero



...ognuno con la sua storia.

Per essere *free* deve essere libero, inoltre può essere gratis (e spesso lo è)

Richard Stallman (FSF) insiste sul termine *free*:

- “*proprietary software divides and subjugates the users*”
- “*a free OS will give users to have freedom while they use their computers*”
- “*The freedom to cooperate with other people, to have a community, is important for quality of life, [...] for having a good society that we can leave in, and that is [...] even more important than having powerful and reliable software.*” [dal film *Revolution OS*]

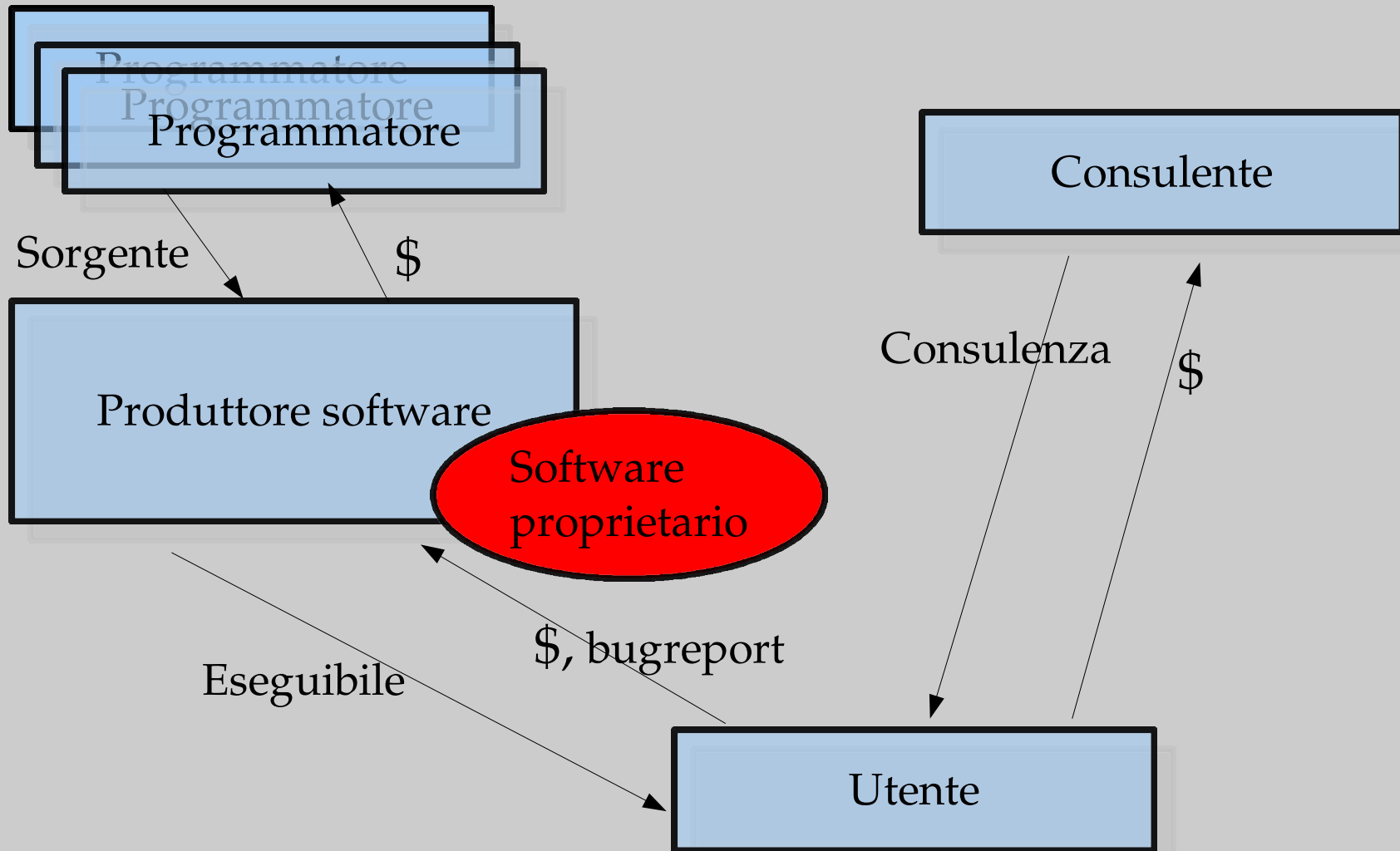
Le 4 libertà

- Richard M. Stallman, Cambridge MA, 1984
- 0: Libertà di **eseguire** il programma, *per qualsiasi scopo*.
- 1: Libertà di **studiare come funziona** il programma, e **adattarlo** *alle proprie necessità*.
- 2: Libertà di **ridistribuire** le copie *in modo da aiutare il prossimo*.
- 3: Libertà di **migliorare** il programma, e **distribuirne pubblicamente i miglioramenti**, *in modo tale che tutta la comunità ne tragga beneficio*.

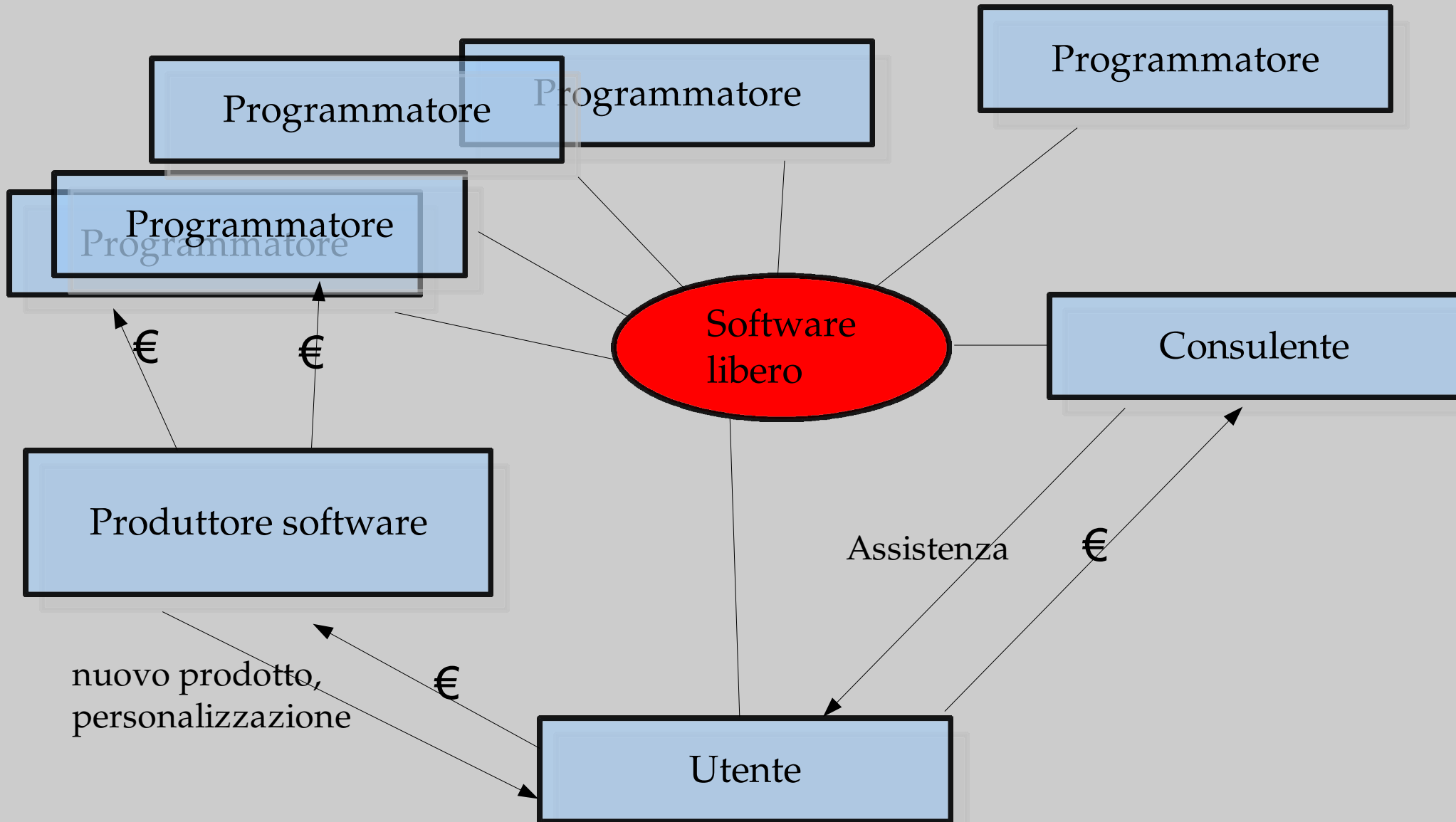
Come classificare il software

- **Per modello di diffusione:**
 - *come viene distribuito all'utente*
- **Per modelli di sviluppo:**
 - *come viene prodotto e da chi, come viene coordinato*
- **Per prezzo:**
 - *a quali condizioni economiche viene diffuso:*
(Gratis, Freeware, Shareware, Commerciale)
- **Per licenza:**
 - *a quali condizioni legali viene diffuso*
(GPL, BSD, Public Domain, EULA...)

Modello di diffusione tradizionale



Il modello Free/Open software



Modelli di sviluppo

(“la cattedrale e il bazaar” Raymond 1998)

- **Tradizionale/Cattedrale**
- Sviluppo centralizzato
 - gruppo ristretto di sviluppatori, selezionati dal management
 - *feedback* utente con *bugreport*
- Project management forte
 - standard di sviluppo rigorosi
 - gerarchia più o meno pronunciata
 - disaccordo: abbandono progetto
- Diffusione solo di prodotti “finiti”
 - debugging a carico di chi sviluppa
- **Internet-centrico/Bazaar**
- Sviluppo distribuito, delocalizzato
 - gruppo di sviluppo ampio, autoselezionato
 - *feedback* diretto con gli utenti
- Project management lasco
 - *peer review*
 - coordinatore progetto = filtro
 - in caso di disaccordo: `fork()`
- Diffusione precoce di prodotti grezzi
 - “*more users find more bugs*” (Brooke)
 - il debugging è parallelizzabile

Coordinamento del progetto

- Management capace di coordinare:
 - Modello *single guru*: Don Knuth (TeX)
 - Modello *master-disciples*, o *benevolent dictator*: Linux
 - Modello *project team*: Apache, X11
- Successo di un progetto:
 - programmatori bravi
 - *timing* giusto: sviluppo, *versioning* (beta, stabile, ...)
 - ampia base di utenti esperti e collaborativi
 - debugging efficace e veloce

Licenze e Diritto d'autore

- Il software è opera d'intelletto: come un libro o un progetto.
- È protetto dal “diritto d'autore” (copyright ©)
- L'autore può cedere il diritto sullo **sfruttamento** dei diritti (ma resta l'autore)
- La **licenza** regola la cessione di alcune *facoltà*:
 - copia
 - modifica
 - ulteriore cessione a terzi

Licenze

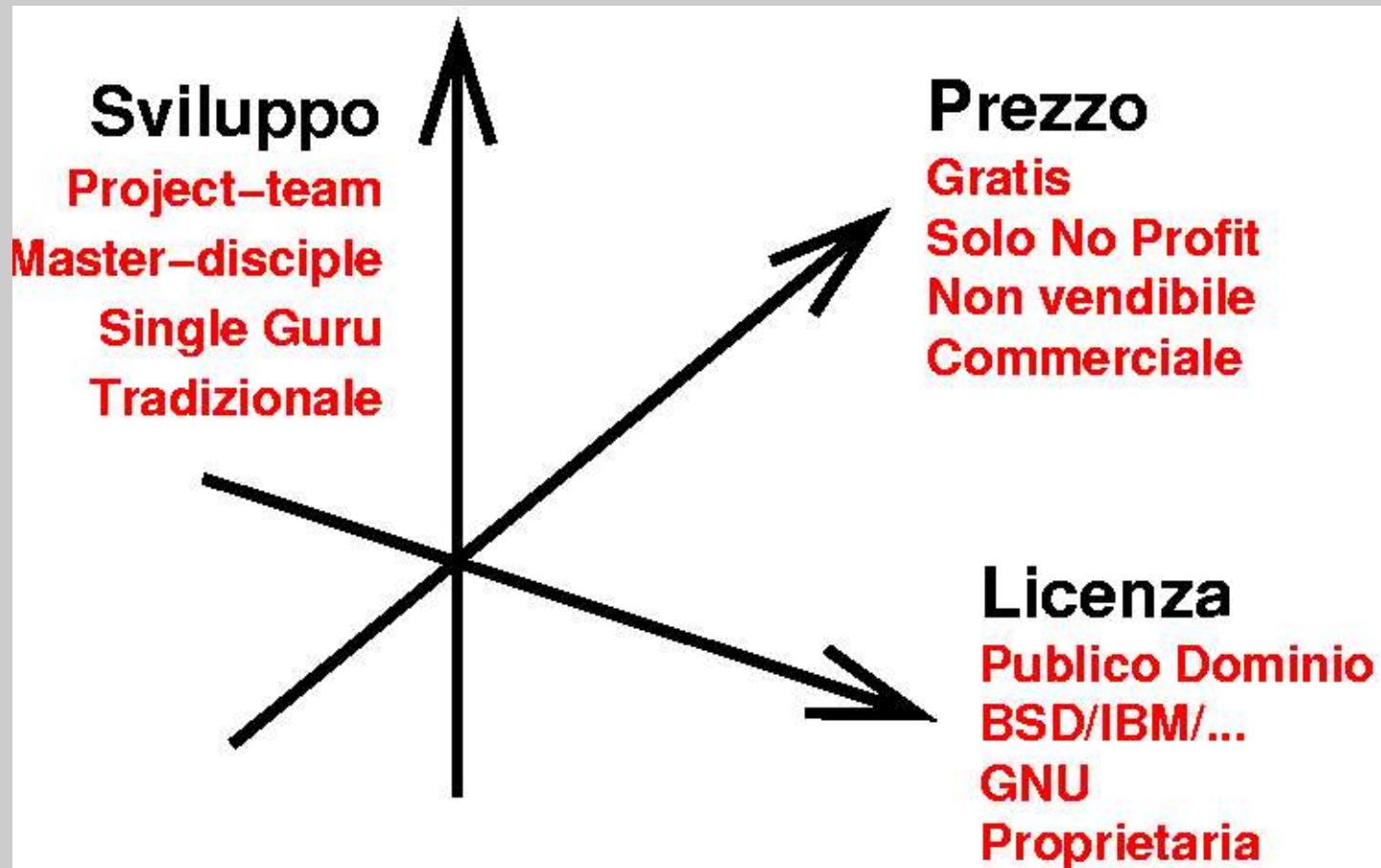
- Esempi di licenze per software con sorgente aperto:
 - **Public Domain**: viene ceduto tutto, anche il ©.
 - **BSD** (Berkeley Standard Distribution): dono liberale del codice:
 - può essere reso proprietario, pur restando il © dell'autore.
 - **GPL** (Gnu Public License): collaborativo;
 - insieme al sorgente devono essere ceduti tutti i diritti: “*copyleft*” o *permesso d' autore*.

GNU GPL: General Public License



- Ideato dalla FSF per garantire le 4 libertà:
 - E' **irreversibile**: nessuno può negare i diritti concessi, ovvero chiudere il sorgente.
 - Anche le **opere derivate** devono essere GPL .
- Sembra essere piuttosto robusta.

Prezzo \perp Modello sviluppo \perp Licenza



Aspetti Sociali

- Promuove lo sviluppo sociale:
 - è **adattabile** al contesto locale, anche culturale,
 - comporta **trasferimento tecnologico**: il sorgente descrive il programma,
 - favorisce la **cooperazione** con una comunità internazionale di utenti e programmatori.
- Favorisce la crescita economica, motore di innovazione:
 - favorisce e stimola la modifica **autonoma** del codice,
 - favorisce una **industria ICT** locale che conosce le esigenze locali.
- E' più efficiente:
 - **riutilizzabilità**: il software diviene patrimonio comune,
 - minore **costo** hardware e software.

Aspetti politici

- Software nella Pubblica Amministrazione:
 - evitare il pericolo *lock-in*
 - strategico: è l'interfaccia col cittadino ⇒ il riferimento del mercato
- I governi devono *sostenere* il S.Libero?
 - devono usare solo quello?
- Risposta del mondo *closed-source*
 - rilascio sorgenti con *Non Disclosure Agreement*
 - sconti, lobbying ('Initiative for Software Choice')

Free/Open future?

- Hardware
 - www.open-hardware.org
 - www.opencores.org
- Open Documentation:
 - Documentazione, manuali consultabili , copiabili ed aggiornabili
- Open Content
- Open Education
- Digital Rights (*Restriction*) Management
- Brevetti sul software

Domande (Libere)?

Questa presentazione si trova qui:

<http://www.stat.unipd.it/~mmzz/Papers/foss-stat04.pdf>

Approfondimenti

- Vantaggi del F/OSS per i vari attori
- Approfondimenti sulle licenze
- Alcuni esempi
- Miti sul F/OSS
- Chi se ne occupa: puntatori
- Sigle
- Bibliografia

Quali vantaggi: programmatore

- Dipende (da cosa vuole)
 - più soddisfazione (pubblicità, proprietà del codice) e interazione con gente interessante
 - *peer review*, può concentrarsi sulla qualità
 - meno tempo speso in debugging
 - meno €€ (ma se è bravo forse anche di più)
 - può attingere alla 'libreria' di tutto il software libero

Quali vantaggi: utente / cliente

- Molti o moltissimi
 - affidabilità: garanzia di supporto continuo
 - sicurezza informatica
 - minore o spesso nessun costo
 - maggior scelta di soluzioni o personalizzazioni
 - *“fly before buy”*

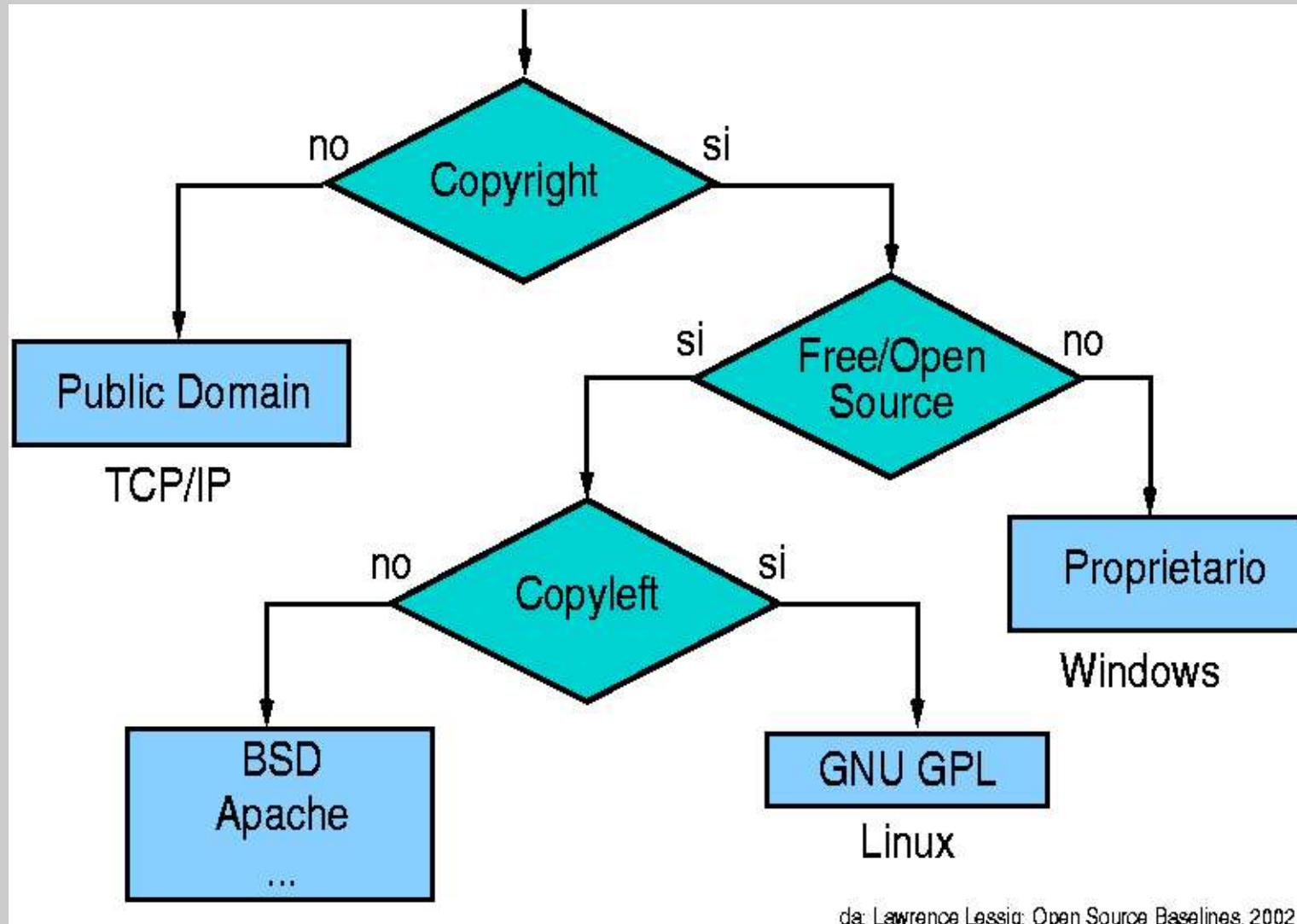
Quali vantaggi: consulente

- Molti
 - può adattare il prodotto al cliente
 - ampia scelta di soluzioni da proporre
 - fa il consulente e non il venditore
 - interazione diretta col programmatore
 - può correggere direttamente un baco.

Quali vantaggi: produttore

- Dipende (è ancora presto per sapere)
 - minor spesa per sviluppo e debugging
 - può sfruttare il nuovo modello di sviluppo
 - spese legali?
 - spostare i ricavi dal prodotto ai servizi, marchi...
- C'è molto movimento...
 - Vedere “*pandora's box for open source*” su news.com, 12 feb 2004

Tipi di licenze



Modelli di licenza

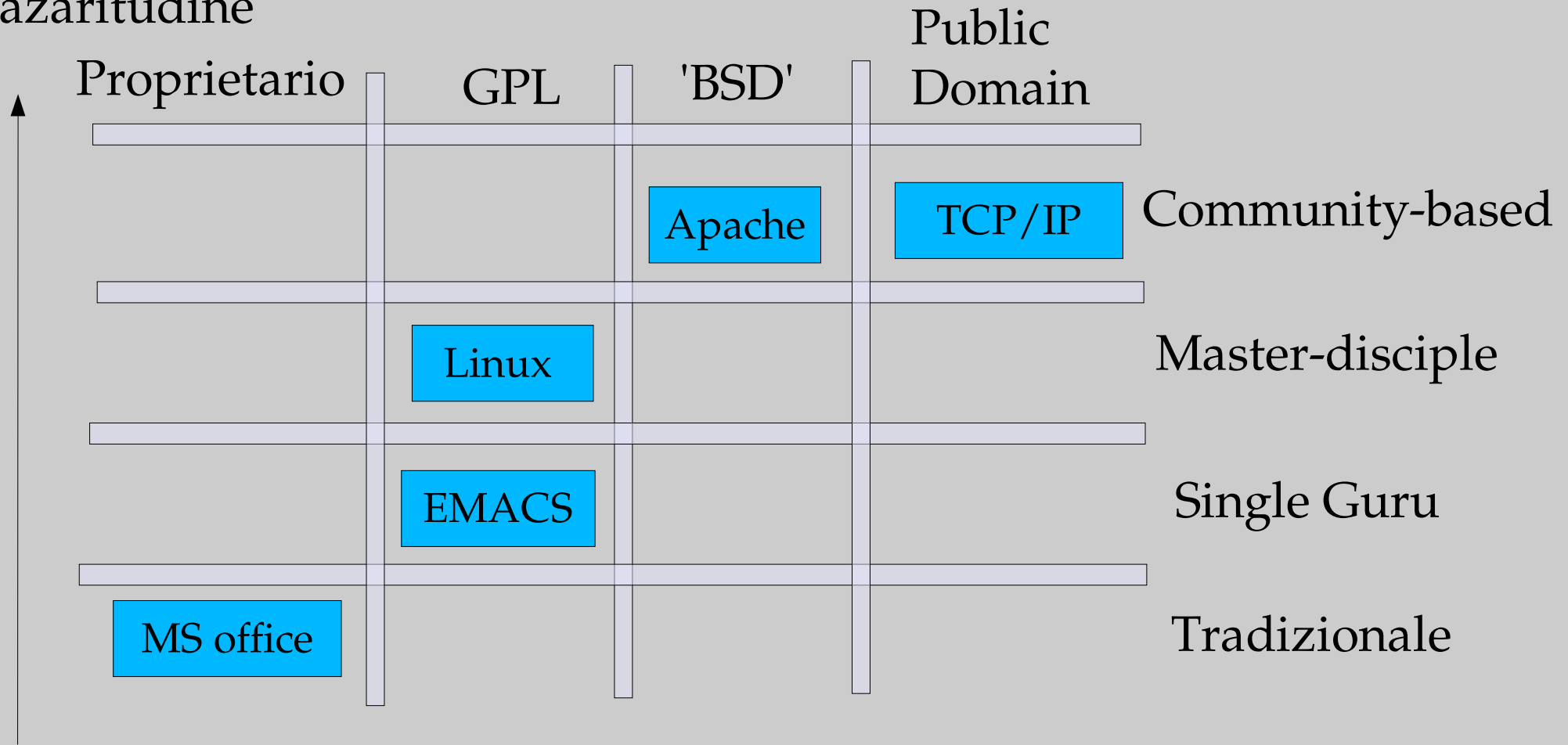
	Esecuz.	©	Letture/Copia	Modif.	Distribuz.
Public Domain	Si	Si	Si	Si	Si/\$i
BSD	Si	No	Si	Si	Si/\$i
IBM Public license	Si	No	Si	Si	Si/\$i
GNU GPL	Si	No	Si	Si/No	Si/\$i
Shared Source	\$i	No	\$i/No	\$i/No	No
Freeware	Si	No	No	No	Si
Shareware	\$i	No	No	No	Si/\$i
Commerciale	\$i	No	No	No	No

Tabella delle *facoltà* o *diritti* concessi all' *utente*.

Vedere www.opensource.org per le licenze considerate Open dalla *Open Source Initiative* secondo la *Open Source Definition* (OSD).

Libero / Proprietario ⊥ Cattedrale / Bazaar

Bazaritudine



Cattedralità

Chiusura

Apertura

Caso MySQL AB



www.mysql.com

- Più diffuso database libero: 4.000.000 installazioni
- Doppia licenza: GPL/Commerciale
- Clienti: Motorola, NASA, Texas Instruments,...
- 100 dipendenti nel mondo. Migliaia di collaboratori
- Guadagni:
 - 1: supporto/assistenza online
 - 2: Licenze commerciali
 - 3: Sfruttamento marchio (certificazioni, corsi, ...)


Caso Ernie Ball



www.ernieball.com

- Visita BSA (Business Software Alliance) nel 2000:
 - licenze non in regola.
 - multa: \$65,000
 - spese legali: \$35,000
- “voglio Microsoft fuori dalla ditta in 6 mesi”.
- Risparmio: \$80,000 in 3 anni.
- *“The myth has been built so big that you can’ t survive without Microsoft”* Sterling Ball, 2003
- da www.news.com 20 agosto 2003

Miti

- Qualità: “se non pago, è sicuramente scadente.”
- Costo: “il TCO (*Total Cost of Ownership*) del software libero è più alto.”
- Assistenza: “non posso avere un contratto di assistenza.”
- Sicurezza: “il fatto che il sorgente sia leggibile lo rende vulnerabile; meglio la vecchia *security through obscurity* (e le chiavi NSA )”.

Chi se ne occupa

- Internazionale:
 - FSF Free Software Foundation : www.fsf.org
 - Open Source Initiative: www.opensource.org
 - GNU : www.gnu.org
 - UNESCO www.unesco.org/webworld/portal_freesoft
 - League for Programming Freedom: lpf.ai.mit.edu
 - Electronic Frontier Foundation: www.eff.org
 - Foundation for a Free Information Infrastructure: www.ffii.org
- Italia
 - AsSoLi (Associazione Software Libero): www.softwarelibero.it
 - ILS (Italian Linux Society) www.linux.it
 - PLUTO: www.pluto.it
- Padova
 - PLUTO Padova: www.plutopadova.org

SIGLE

- BSD : Berkeley Standard Distribution
- DRM : Digital Rights Management
- EULA: End User License Agreement
- FSF: Free Software Foundation
- GNU : GNU's Not Unix
- GPL : General Public License
- IPR : Intellectual Property Rights
- NSA : National Security Agency
- OS: *operating system*, sistema operativo
- TCO: Total Cost of Ownership
- TCPA : Trusted Platform Computer Architecture
- GNU Hurd: Hird of Unix-Replacing Daemons. Hird: Hurd of Interfaces Representing Depth.

Bibliografia

- Fuggetta, A.; *Open source software - an evaluation, 2003*, The Journal of Systems and Software 66, Elsevier
- Raymond, Eric S. ; *A Brief History of Hackerdom*, 2000, <http://www.catb.org/~esr>
- Stallman, Richard M.; *Free Software, Free Society*; Boston 2002, GNU Press
- Raymond, Eric S.; *The Cathedral & the Bazaar*, (2 ed.) O'Reilly, Sebastapol, CA, 2001.
- Bezroukov, Nikolai; *A Second Look at the Cathedral and Bazaar*, First Monday, volume 4, number 12 (December 1999); <http://firstmonday.org>, <http://www.softpanorama.org>
- DiBona, Chris; Ockman, Sam; Stone, Mark; editors; *Open Sources: Voices from the Open Source Revolution*. O'Reilly and Associates, Cambridge, Massachusetts, 1999
- van Reijswoud, Victor; Topi, Corrado; *Alternative Routes in the Digital World: Open Source Software in Africa*, 2003
- Torvalds, Linus; Diamond, David; *Just for Fun*, Texere, London, 2001. (*Rivoluzionario per caso*, Garzanti)
- Hahn, Robert W., editor; Bessen, James; Evans, David S.; Lessig, Lawrence; Smith, Bradford L.; *Government Policy toward Open Source Software*; AEI-Brookings, 2002
- Moore, J.T.S.; *Revolution OS*, Wonderview Productions, LLC, 2002 (film)

Fine

(happy hacking)

© Alberto Cammozzo, 2004; La copia letterale e integrale e la distribuzione di questo documento sono permesse con qualsiasi mezzo, a condizione che questa nota sia riprodotta.