

Il software libero

Alberto Cammozzo

mmzz@stat.unipd.it

Dipartimento di Scienze Statistiche, Università di Padova
progetto ReFun, PLUTO
FSFE fellow

Servizio Formazione
Università' di Padova
Ottobre 2005

versione 1
revisione 08/11/05

© Alberto Cammozzo, 2005; La copia letterale e integrale e la distribuzione di questo documento sono permesse con qualsiasi mezzo, a condizione che questa nota sia riprodotta.

Seconda parte: f/oss

Storia e storie di free software

Il mercato del software

Modelli: licenze, aspetti di novita'

Aspetti sociali e politici

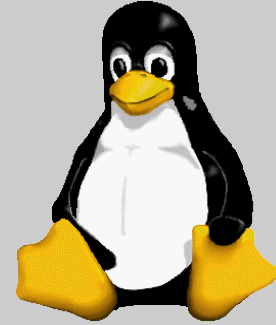
C'era una volta un programma...

- 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.



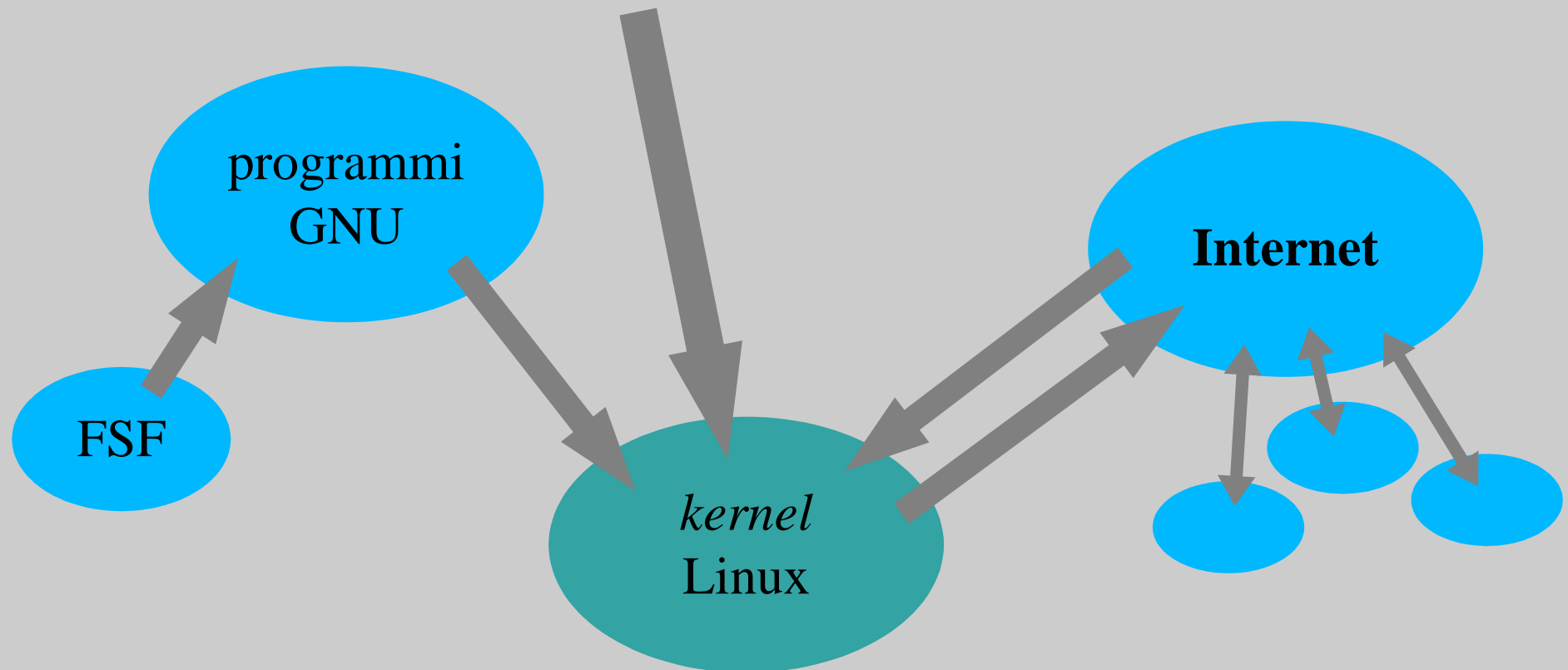
Prima storia: GNU (*Gnu is Not Unix*)

- Richard Marshall Stallman, MIT, USA.
 - “*ci fu impedito di fare cose utili*”
 - aggressione alla libertà ↗ Free Software Foundation.
- 1983: *GNU project*: come Unix, ma completamente libero:
 - *Tools*: editor (Emacs), compilatore (gcc), '84 ⇒ '91
 - *Kernel*, il nucleo: GNU Hurd: è molto avanzato e più difficile del previsto: blocca il progetto.
- Per mantenere libero il software: licenza GPL.
- “*Make the world a better place*”.



Seconda storia: Linux

- Linus Torvalds,
Helsinki University, Finlandia, 1991



Terza storia: Open Source



- Prima ditta **assistenza** sul free software: 1989 (Cygnus Software, Michael Tyman)
- Il successo di Linux e degli altri programmi liberi attira il mercato.
- *Free* (che suona *gratis*) è un termine tabù per il business: *Open Source* è il suo *marketing term*.
- 1998: Netscape, sotto la pressione della concorrenza Microsoft, decide di rilasciare il sorgente del browser Mozilla.
- Sun: OpenOffice.org
- Anni 2000: IBM, Novell supportano e diffondono Linux

Il successo di GNU/Linux

Programmi chiave:
- webserver apache
- openoffice.org

...

Comunita' di
programmatori
e utenti

Kernel Linux

Programmi GNU

Interesse
dei produttori
di software
proprietario
e hardware

Interesse
delle PA e
dei governi

Aziende
produzione:
rilascio
di sorgenti

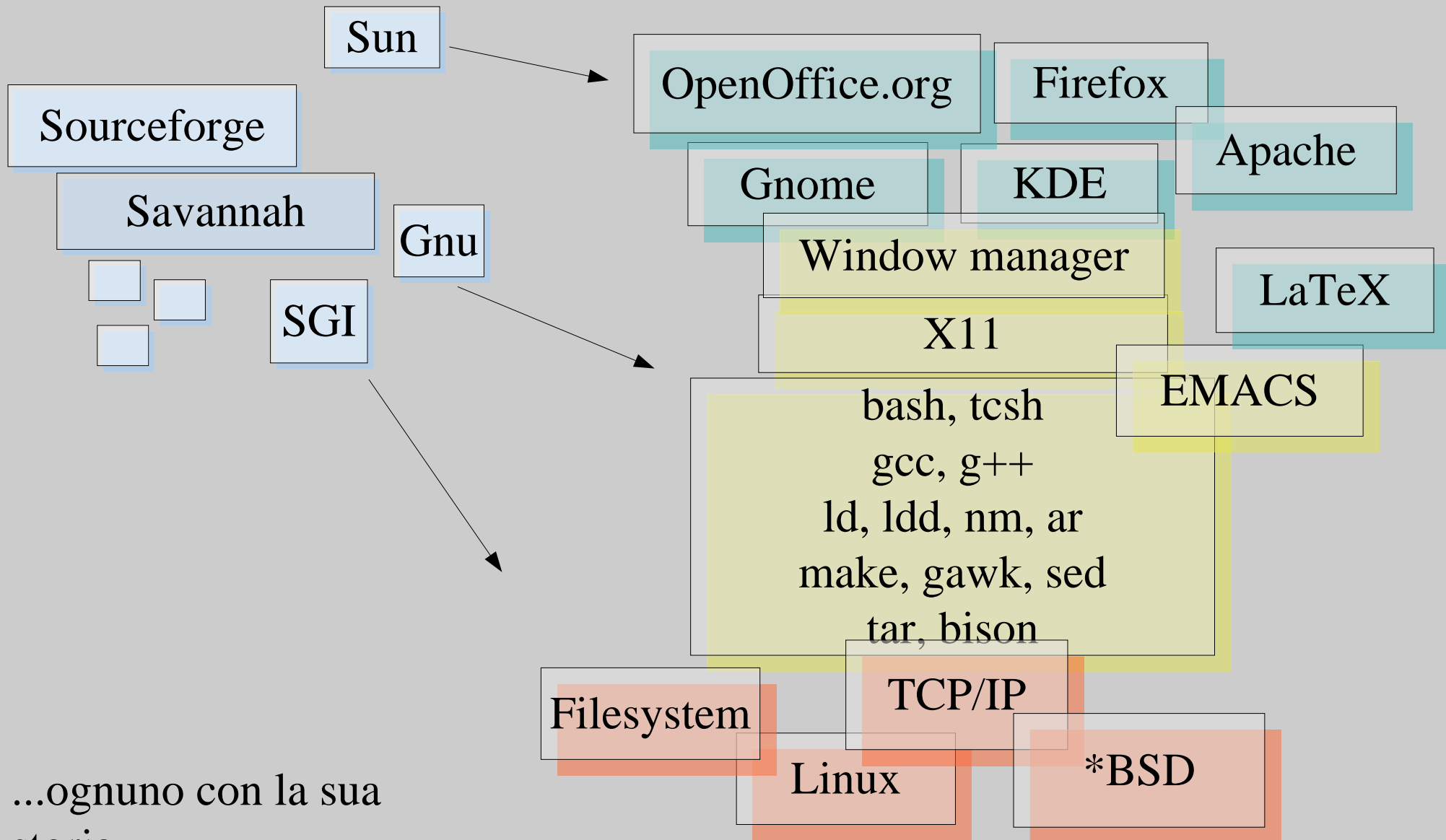
new economy
.com

Idea
Open Source

Aziende
che prestano
assistenza,

Idea
Free Software

Mappa di un sistema libero

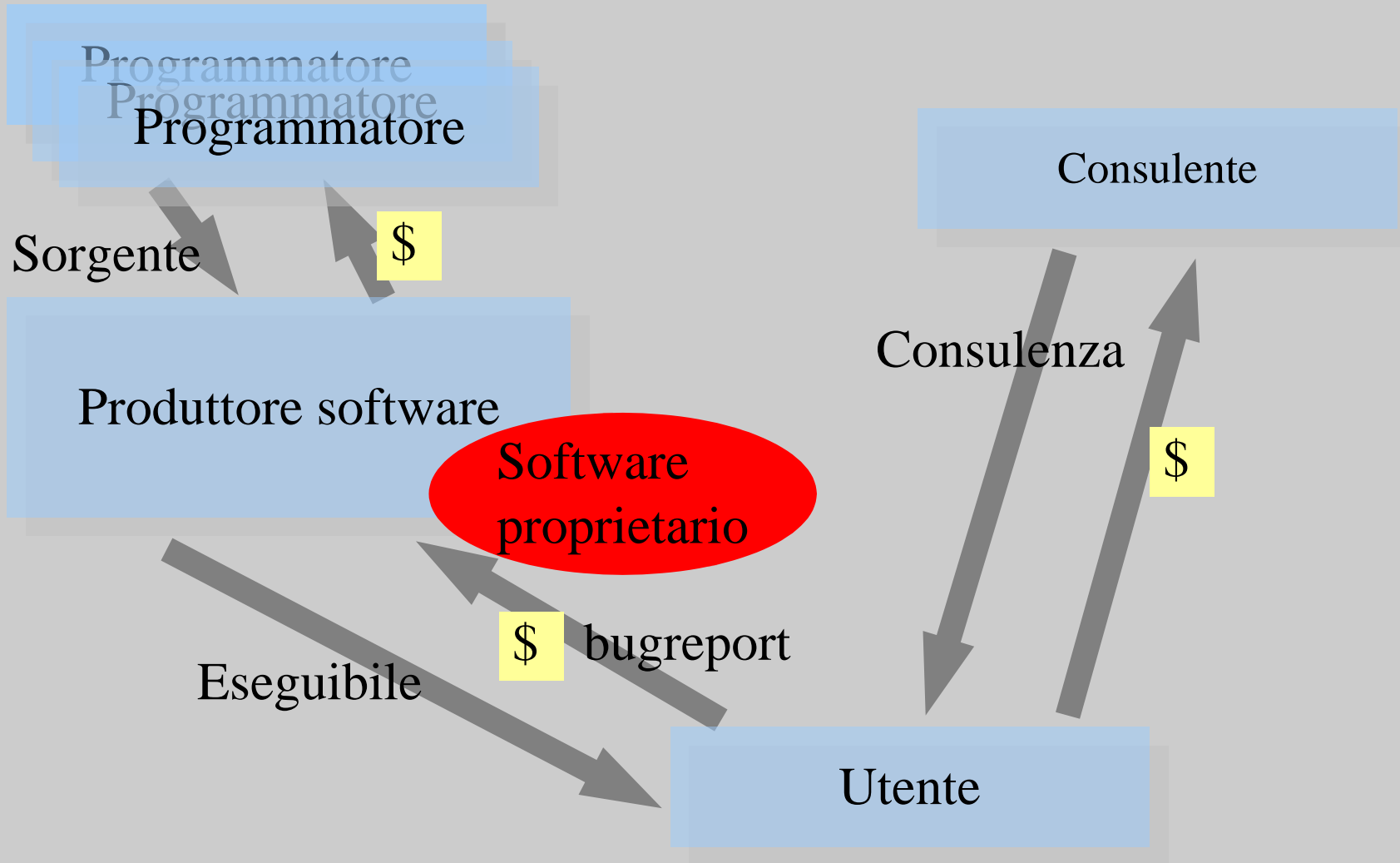


...ognuno con la sua storia.

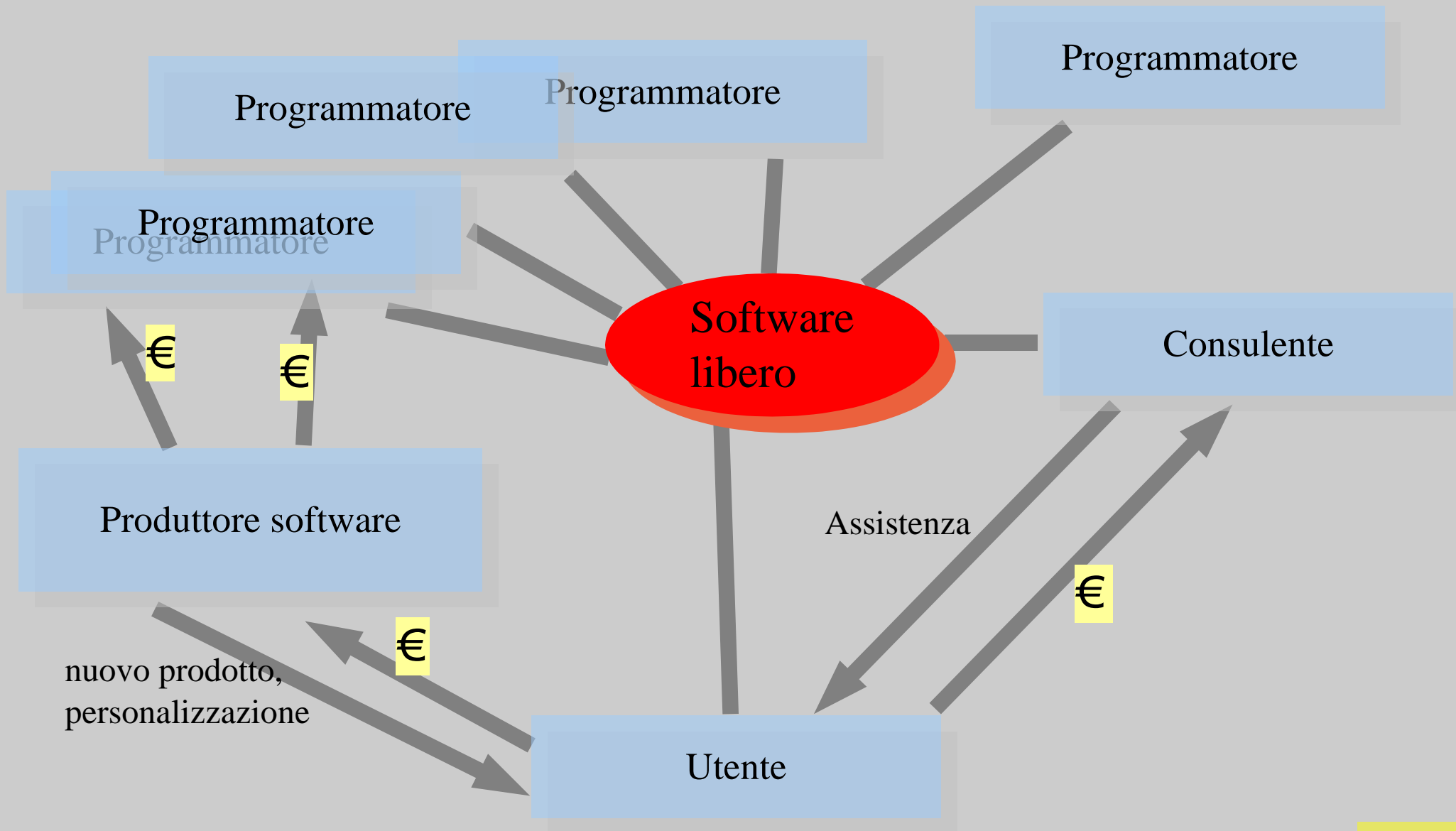
Ma come funziona il mercato del software?

- Attori:
 - **Clienti** (aziende, PA, privati, ...): acquistano, segnalano errori.
 - **Software house**: producono, adattano.
 - **Programmatori**: scrivono il codice.
 - **Consulenti, rivenditori**: raccomandano i prodotti e ove possibile li adattano.
- Prodotti:
 - Pacchetti: confezionati per tutti.
 - Software *custom*: su misura.

Modello tradizionale



Il modello Free/Open software



Differenze: utente/cliente

- possesso sorgenti:
 - garanzia di supporto continuo
 - sicurezza informatica
- minore o spesso nessun costo
- ampia scelta di soluzioni
 - personalizzabili
 - provare prima di usare.
- prodotti meno *rifiniti*

Differenze: programmatore

- pubblica paternita' codice
 - pubblicità, riconoscimento
 - interazione piu diretta col cliente, con altri programmatori
 - meno tempo speso in correzione errori
- meno €€(o forse anche di più)
- *libreria* di tutto il software libero

Differenze: Pubblica Amministrazione

- Oltre agli altri vantaggi:
 - garanzia di accessibilita' dei dati nel tempo .
 - Dati realmente pubblici.
 - Interfaccia trasparente e equa nei confronti del cittadino.
 - Stimolo all'innovazione locale.

Differenze: consulente

- codice sorgente disponibile
 - può adattare il prodotto al cliente
 - ampia scelta di soluzioni da proporre
 - interazione diretta col programmatore
 - può correggere direttamente un errore.
- rivenditori e intermediari: forse meno spazio

Differenze: produttore

- E ancora presto per sapere
 - minor spesa per sviluppo e debugging
 - sceglie programmatori che conoscono già il codice
 - può sfruttare nuovi modello di sviluppo
 - meno spese legali?
- Spostarsi dal prodotto ai servizi.
- Tutti amano il codice aperto *degli altri*.
- Sun, IBM, Novell... Microsoft?

Come classificare il software

- 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...).
- Per modelli di **sviluppo**:
 - *come viene prodotto e da chi, come viene coordinato.*
- Per modello di **diffusione**:
 - *come viene distribuito all'utente* (Internet, negozio, ...).

La libertà non ha prezzo?

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

- Essenziale perchè sia libero: poter ispezionare, modificare e redistribuire il *codice sorgente*.

Prezzo

- **Freeware:** gratis, sorgente non accessibile (proprietario)
- **ShareWare:** non è gratis, può essere copiato e provato, prima di essere pagato (di solito poco)
- **A pagamento**

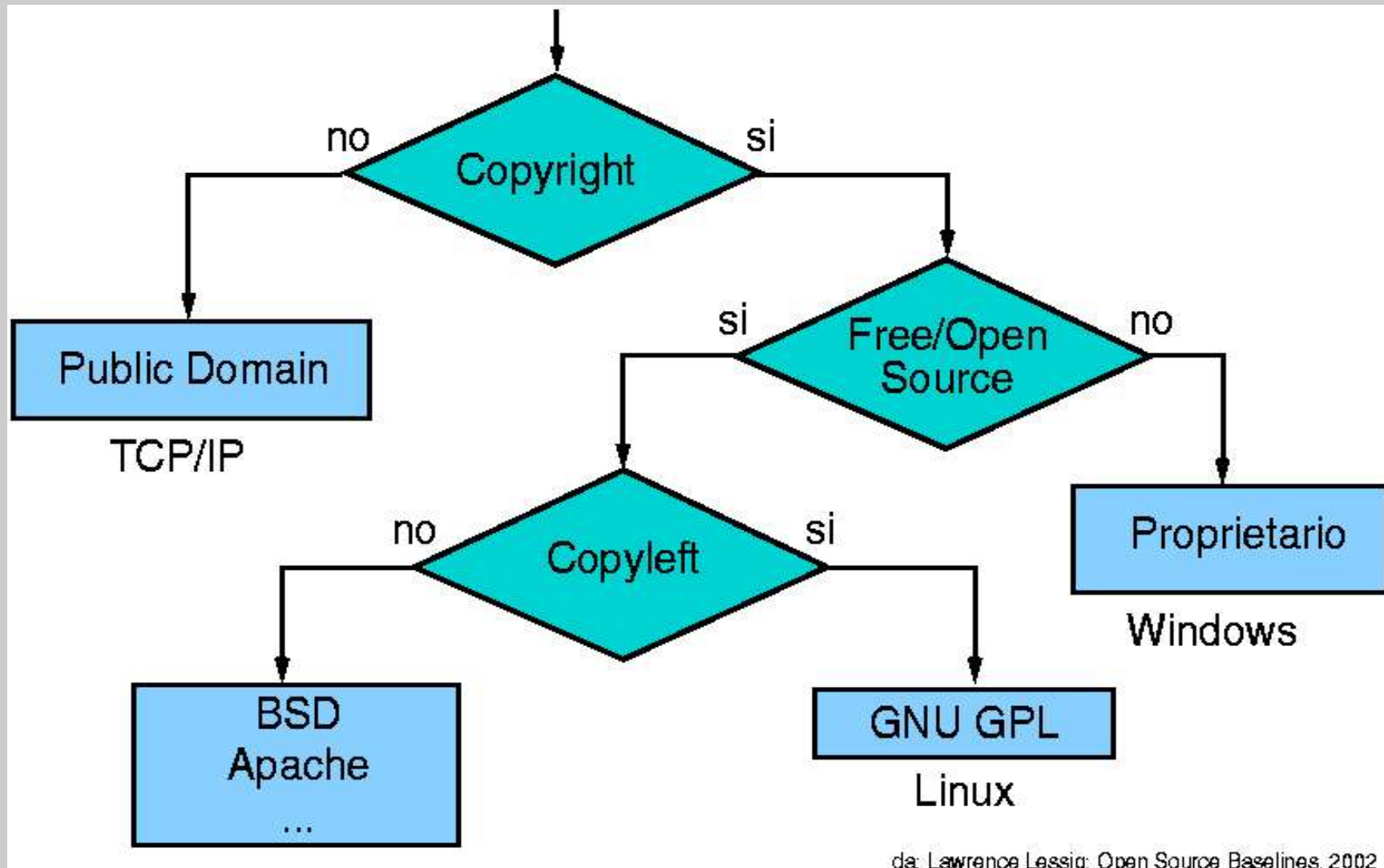
Licenze e Diritto d'autore

- Il software è opera d'intelletto: come un libro.
- 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
 - ...

Modelli di 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, deve rimanere il © dell'autore.
 - **GPL** (General Public License): collaborativo;
 - insieme al sorgente devono essere ceduti tutti i diritti: “*copyleft*” o *permesso d'autore*.

Tipi di licenze



da: Lawrence Lessig: Open Source Baselines, 2002

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 .
- › **Chi distribuisce** copie di un programma coperto da GPL, sia gratis sia in cambio di un compenso, deve concedere ai destinatari tutti i diritti che ha ricevuto.
- › Deve anche assicurarsi che i destinatari ricevano o possano ottenere il codice sorgente.
- › E deve mostrar loro le condizioni di licenza, in modo che essi conoscano i propri diritti.

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*.

Free Software Foundation

Richard Stallman 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*]

Modelli di licenza

	Esecuz.	©	Lettura/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).

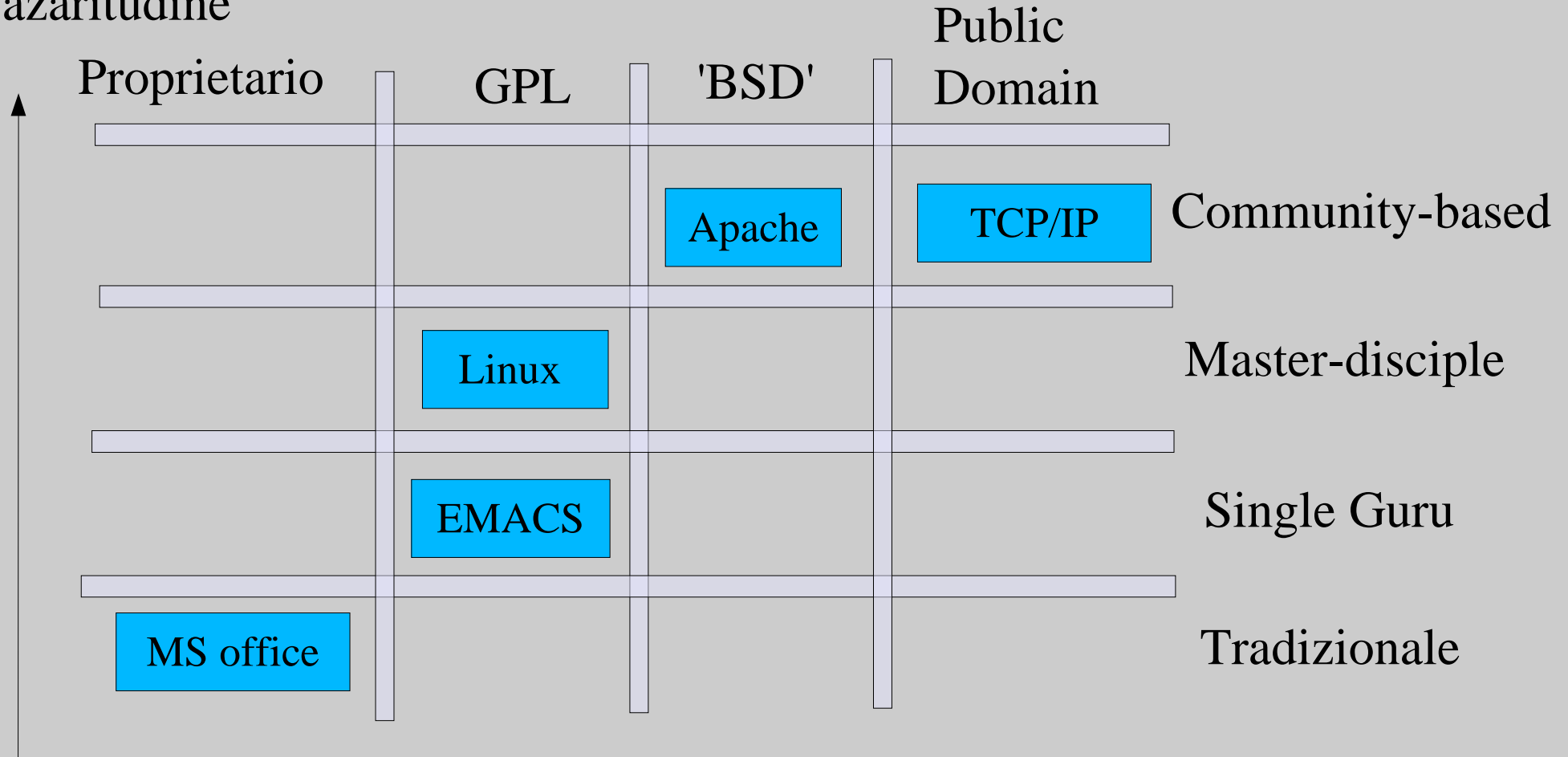
Modelli di sviluppo

(“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

Libero/Proprietario ⊥ Cattedrale/Bazaar

Bazaritudine



Cattedralità

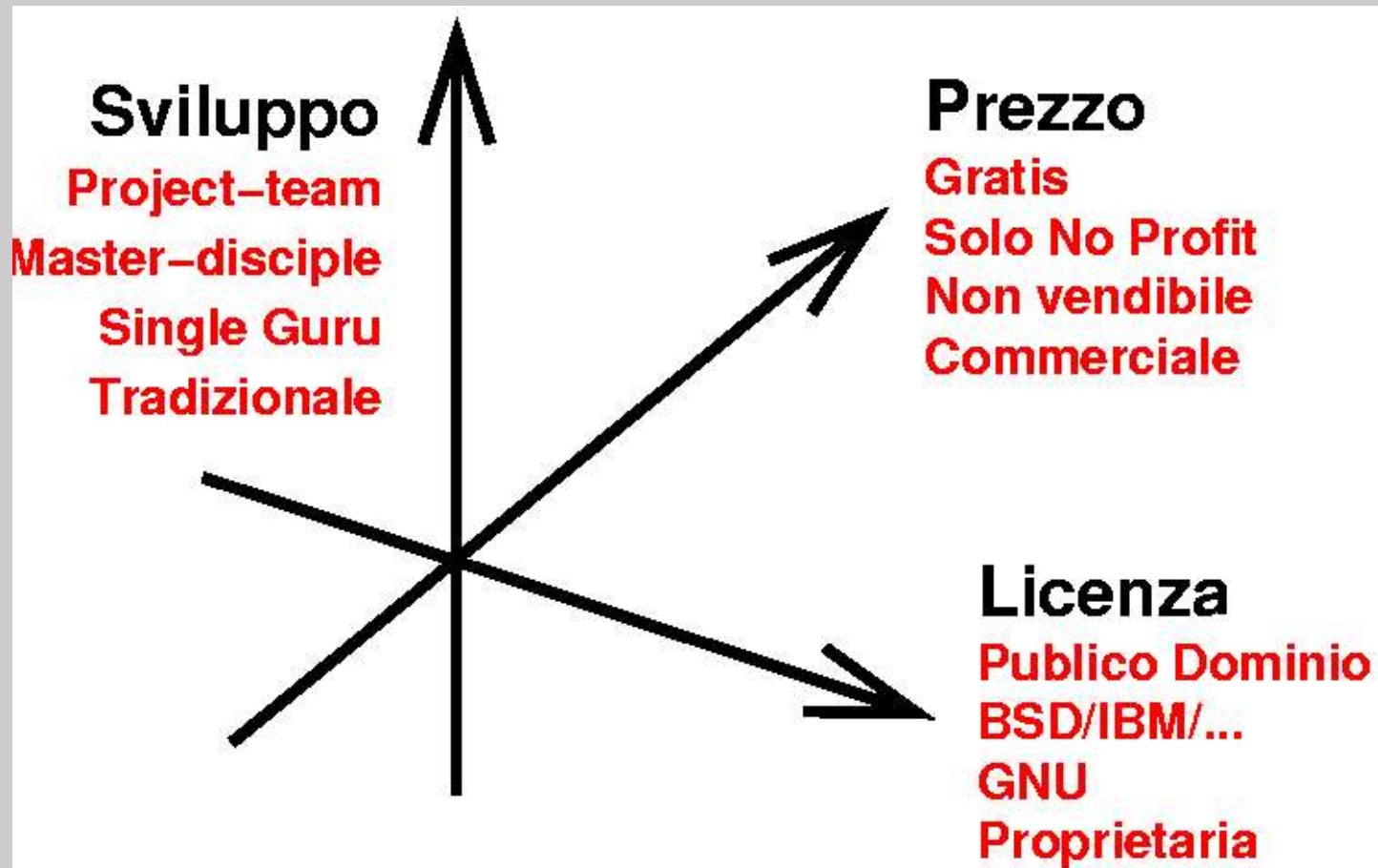
Chiusura

Apertura

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

Prezzo \perp Modello sviluppo \perp Licenza



Cosa c'e' di nuovo?

- FOSS e' una novita'
... dagli anni '70.
- Quale motivo per il nuovo impulso?
 - modello di business/prezzo? non sempre.
 - Licenze? Certamente!
 - *Distribuzione?* Sicuro!

Successo

- Licenze libere = Free software
- **Internet** + licenze libere = Free software *per tutti*
- Internet + licenze libere + **aziende** = Open Source

cosa c'e' di nuovo nel FOSS?

	esclusivo foss	nuovo	comune con non-foss	non esclusivo foss
sviluppo	nessun modello esclusivo	no	tradizionale, single-guru, master-disciple, core-team	
licensing	GPL, BSD, PD	no	no	EULA
prezzo	no	no	affitto servizio, affitto uso	shareware, freeware, non-commercial
business	no	no	servizi, ASP	
distribuzione	download completamente funzionale con codice sorgente	SI	alcuni mezzi in comune	limitato dal mezzo di distribuzione
uso di tecnologie innovative	no	no	tutte	non esclusivo foss
velocita' di rilascio	dinamico	no		limitato dal modello di distribuzione

Ruolo di Internet

- In che modo Internet ha cambiato le cose?
- Facile trovare i programmi che servono
 - motori di ricerca
 - repository (*sourceforge, savannah, ...*)
- Facile interagire con i programmatori (*email*)
- Facile provare il software
 - scarica, prova, butta
 - scarica, adatta, impiega