

Technological Innovation in Software Industry
—
Open Source Software

Ph.D. Stud. Kasper Edwards
Institut for Produktion og Ledelse
Danmarks Tekniske Universitet

Hvorfor bliver open source software udviklet?
-
Incitament og licenser i økonomisk perspektiv

Ph.D. Stud. Kasper Edwards
Institut for Produktion og Ledelse
Danmarks Tekniske Universitet

Planen

- **Hvad er open source software**
- **Det forskningsmæssige problem**
- **En model**
- **Relation til andre områder**
- **Andre perspektiver**
- **Konklusion**

Hvad er open source software?

- **Software som foræres væk**
 - Tilgængeligt på Internettet
- **Software som alt anden software**
 - Programmer til computere
 - Operativsystemer, browsere, etc.
- **Særlig licens**
 - Licens = En begrænset overdragelse af rettigheder
 - Ophavsret er central
 - Fri kopiering og distribution
 - Ret til at modificere kildekoden

```
int notifier_chain_register(struct notifier_block //
**list, struct notifier_block *n)
{
    write_lock(&notifier_lock);
    while(*list)
    {
        if(n->priority > (*list)->priority)
            break;
        list= &((*list)->next);
    }
    n->next = *list;
    *list=n;
    write_unlock(&notifier_lock);
    return 0;
}
```

Hvad er open source software?

□ Ikke noget nyt fænomen

- 1970 -1976 Tidlig Unix, AT&T og deling af software
- 1977 -1983 Første frie Unix, begyndende fragmentation
- 1984 -1990 The Free Software Foundation, Fri software licens
- 1991 -1997 Tidlig Linux, fri software spredes
- 1998 - Nu Open Source Software Era

□ Organisation

- Et program/software pakke – Et projekt
- En eller flere personer tegner projektet
- Altid en form for adgangskontrol til kildekoden
- Maintainer eller andet organ udvælger hvad der inkluderes

Hvad er open source software?

□ Udviklingsmetode

1. Maintainer frigiver software og kildekode
2. Bruger downloader software og kildekode
3. Bruger identificerer problemer eller manglende features
4. Bruger implementerer ændringerne
5. Bidrager returnerer ændringerne til maintainer
6. Ændringerne diskuteres
7. Maintainer inkluderer ændringerne og frigiver en ny version

Det forskningsmæssige problem

□ Observation

- Firmaer og private bidrager til udviklingen af open source software
- Udvikling af OSS tager tid, Tid som er har en offeromkostning
- OSS muliggøre ikke umiddelbart at der profiteres
- Hvorfor udvikles open source software?

□ Mulige forklaringer

- Sjøv
- Ego og gaveøkonomi
- Rygteeffekter som resulterede i betalende job
- En græsrodsbevægelse mod Microsoft

Det forskningsmæssige problem

□ Ønskes: En forklaringsmodel

- Baseret på økonomi
- Forholder sig til udviklingsprocessen
- Forholder sig eksplicit til licensen

□ Forklaringsmodel

- Adfærd (incitament og mekanismer) = Egenskaber ved software + Typen af agent
- Primær interesse: Udvikling i det enkelte projekt

□ Teoretisk basis

- Økonomiske goder
- Eksternaliteter
- Konkurrerende teknologier

Det forskningsmæssige problem

□ Forskningsspørgsmål

- Hvorfor bliver open source software udviklet?
 - » Hvordan bliver OSS brugt, udvekslet og markedsført?
 - » Hvad karakteriserer OSS udviklingsprocessen?
 - » Hvordan påvirker egenskaber ved software det økonomiske gode?
 - » Hvad er eksternaliteterne i OSS? Og hvordan påvirker disse adfærd?

□ Empiri

- Uendeligt meget – kommunikation foregår via postlister
- Dog er det indforstået snak
- Åbne interview

□ Metode

- Idealet: Hypotetisk deduktiv
- Realiteten: Eksplorativ koblet med Hypotetisk deduktiv
- Åbne interview -> lav model -> analyser interview udfra model

En model

□ Forklaringsmodel

- Adfærd (incitament og mekanismer) = Egenskaber ved software + Typen af agent

□ Egenskaber ved Software

- Tekniske egenskaber
- Licensegenskaber

□ Tekniske egenskaber

- Software kan reproduceres uden tab af kvalitet
- Kopieringsomkostningerne er negligerbare
- Distributionsomkostningerne er negligerbare

□ Licensegenskaber

- Licensen udstikker rammerne for adfærd
- Rammerne er analyseret i forhold til et økonomisk perspektiv

Typen af agent

- **To typer agenter**
 - Virksomheder
 - Individer

- **Virksomheder**
 - Råder over ressourcer
 - Skal betale løn
 - Profit maksimerende

- **Individer**
 - Råder kun over egen tid
 - Nyttmaksimerende



Licensegenskaber

□ Tre Ekstremer

- Microsoft EULA
- GNU GPL
- BSD

□ Microsoft End User License Agreement

- Man må *ikke* kopierer, distribuerer og modificerer
- Programmet må bruges på den intenderede måde
- Programmet må sælges en gang
- Der er ingen garanti for funktion og tab af data
- - - - En monopolists ønskedrøm



Licensegenskaber

□ GNU General Public License (GPL)

- Man må kopierer, distribuerer og modificerer
- Kildekoden til distribuerede modifikationer skal være tilgængelige
- Distribuerede kopier skal have samme licens
- Viruseffekten
- - - - Beskytte brugerne



□ BSD Licensen

- Man må kopierer, distribuerer og modificerer
- Modifikationer og afledte arbejder må distribueres under andre licenser
- - - - Det liberale alternative



Roller, brugsprodukter og økonomi

□ To roller

- Maintainer – har taget ansvar for et program
- User-developers – bruger og udvikler programmet

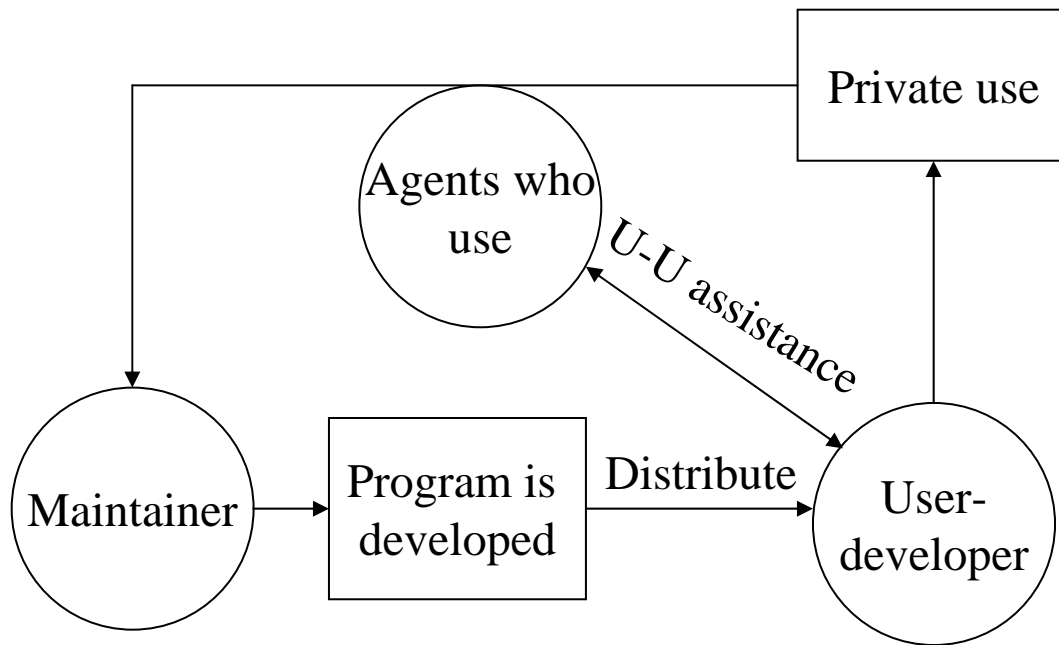
□ Brugsprodukt

- En agents personlige brug af features i et eller flere programmer

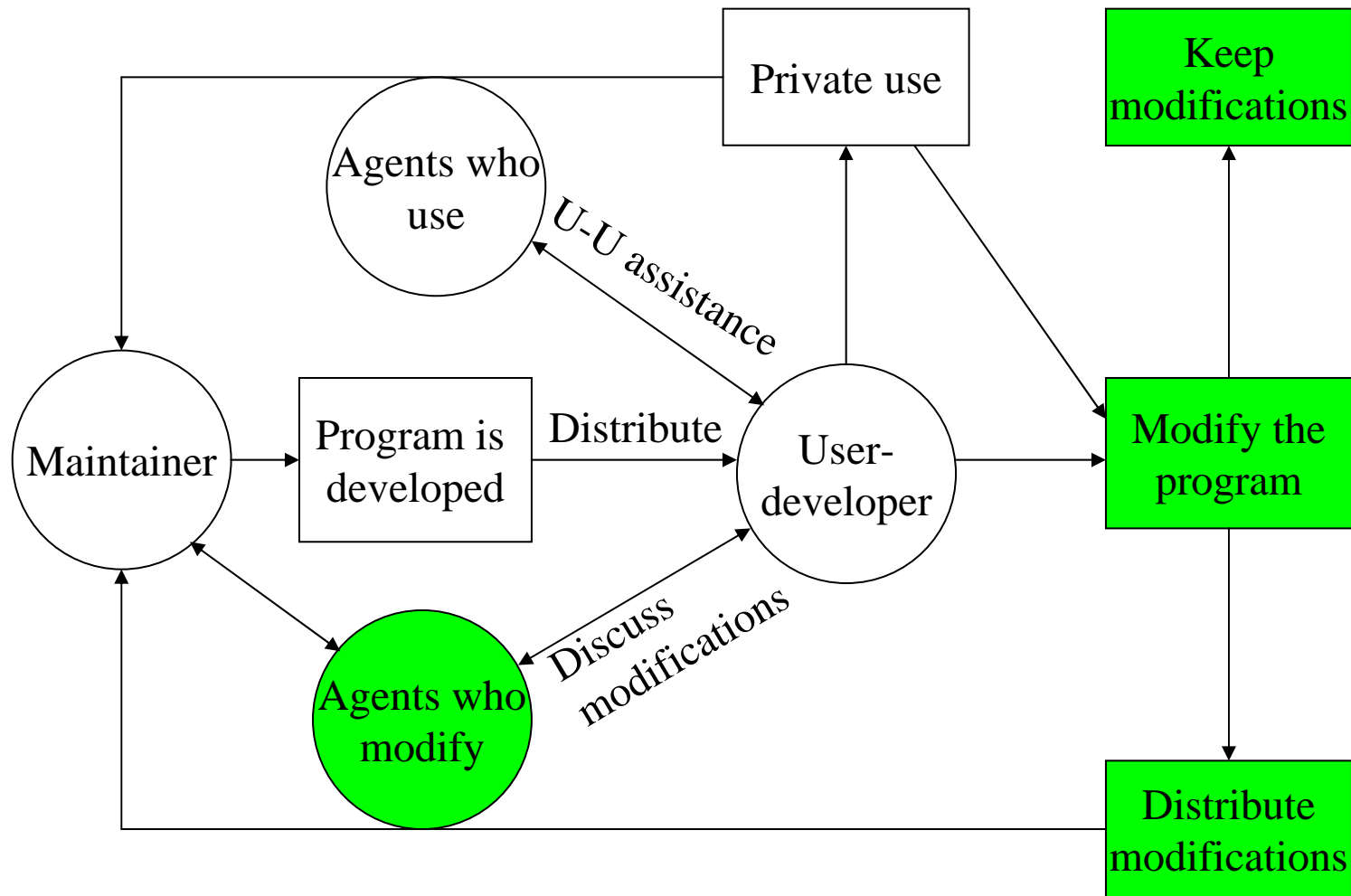
□ Økonomisk adfærd

- Agenter handler rationelt
- Nytte- og/eller profitmaksimerende
- Aktiviteter har en omkostning
- Et brugsprodukt har en anvendelsesværdi
- Agenter bruger og/eller deltager kun hvis giver nytte

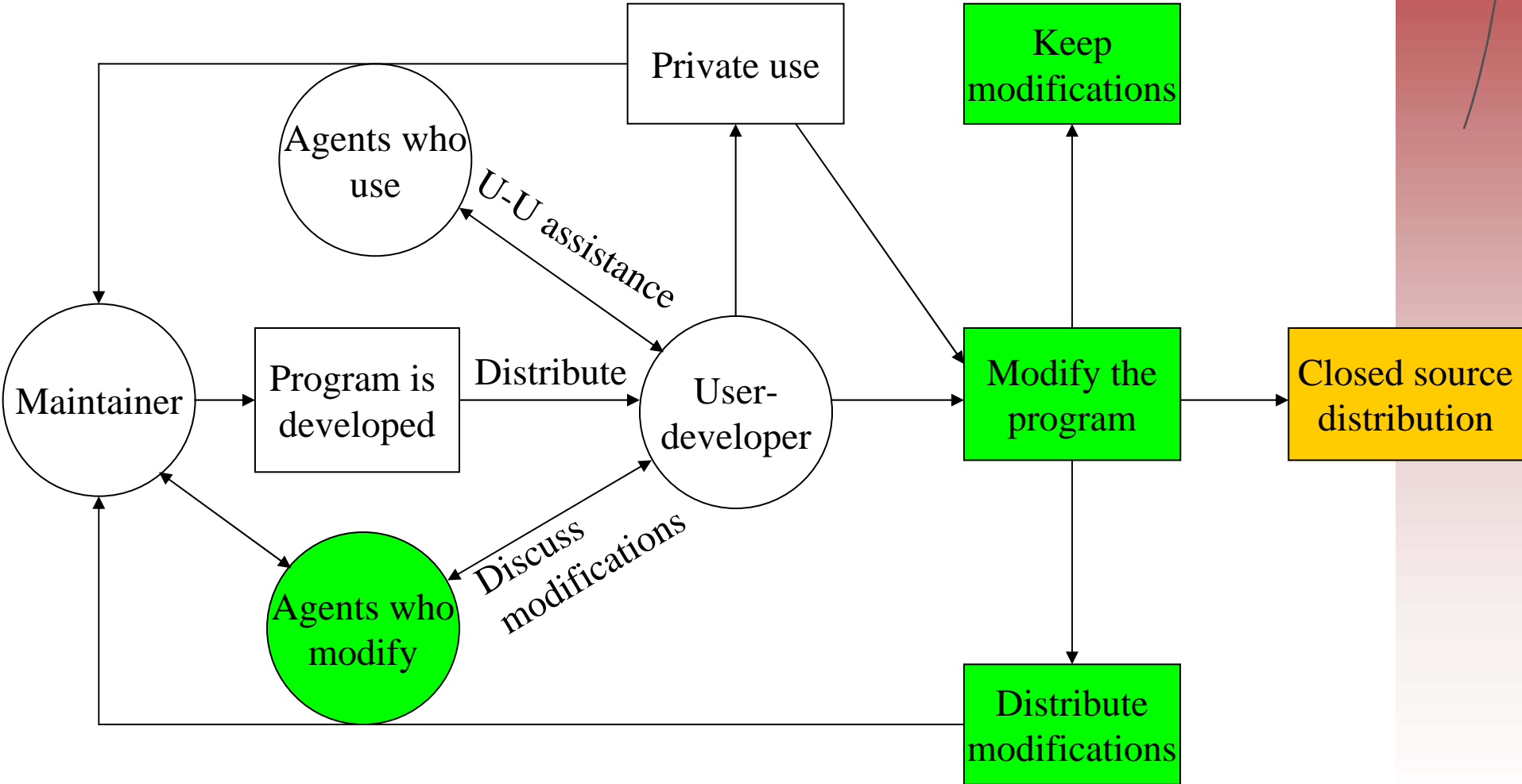
GPL licensen



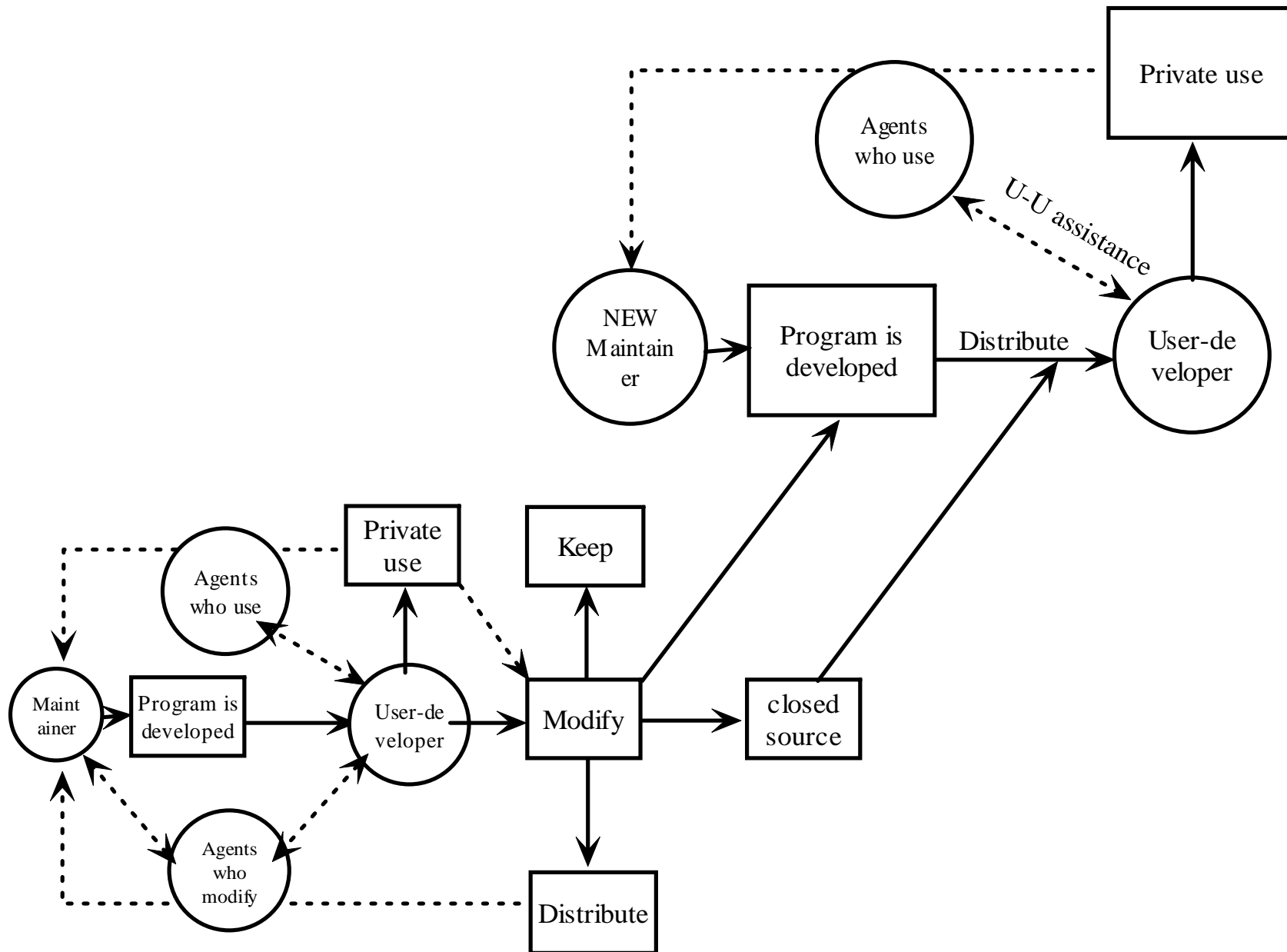
GPL licensen



BSD licensen



A Model – BSD License



Relation til andre områder

- **En (fæl) sammenblanding**

- **Økonomi**
 - Kvalitativ forståelse
 - Mangler formalisme

- **Jura**
 - Økonomiske konsekvenser af licenser

- **Softwareudvikling**
 - Forståelse af incitament

Andre perspektiver

□ Ledelse

- Dybere indsigt i maintainer
- Vanskeligt at forstå ledelsesopgaven

□ Epistemiske samfund og læring

- Mennesker med samme overbevisning finder sammen
- Forklarer klikedannelse
- Læring belyser vigtigheden af at socialisere nye ind i et projekt

Konklusion

- **En forklaring på hvorfor OSS bliver udviklet**
 - Tilbyder en rationel forklaring på fænomenet

- **Licensen sætter rammerne for adfærd**
 - De tre licenser giver forskellige incitamenter

The End

DTU



To eksempler på OSS udvikling

□ Henrik – Mangler brugsprodukt

- Programmør og and Linux entusiast
- Interesseret i kerneudvikling
- Var nød til at bruge Windows på arbejde
- Linux var ikke i stand til at læse Windows' filer
- Gordon havde en lap, som han vedligeholdte 😊
- Lappen var besværlig at fat i 😞
- I frustration beslutter Henrik at få lappen integreret
- Lappen integreres i Linux og nye versioner indeholder brugsproduktet

To eksempler på OSS udvikling

□ Poul – Betalte modifikationer

- FreeBSD core developer og konsulent
- Lille amerikansk ISP ønskede et nyt produkt
- Produktet: Individual hosting
- Hver kunde skulle opleve at have fuld kontrol
- Man brugte allerede FreeBSD, men brugsproduktet manglede
- Poul blev hyret til arbejdet

- To priser: 1) Private 2) Open source
- Gevinst: Generelt løft i FreeBSD's sikkerhed

The end

DTU



Does the Model Hold Up?

□ Henrik

- Individual, limited spare time
- High opportunity cost
- The modification was a one time investment

□ Paul

- Consultant, working for a firm
- Obvious incentive for keeping private
- Small firm, Paul charged extra for keeping private
- No in-house programmers – no wish for extra maintenance
- Number 1 to market the new product more important

Incentives and Costs

□ Incentives for individuals

- Ego – “LOOK What I have created!”
- Peer reputation
- Signaling effects – may spill into the real world
- A desired use-product can be obtained at little cost

□ Incentives for firms

- A desired use-product can be obtained at little cost
- Homemade modifications
- Possible to create services without maintaining the software

□ Costs

- Time (adoption, programming, integration, etc.)
- Firms: Wages
- Individuals: Opportunity cost

Dynamics

- **Uncertainty**
 - Difficult to predict development path => Lowered free riding
- **The “Keep Private” Penalty**
 - Keep private => High maintenance costs
- **Always a work in progress**
 - Low initial commercial value => license lock-in (GPL)
- **The cost of being too late**
 - Development becomes a sunk cost
 - Risk of the “Keep Private” penalty
- **Aggregated benefit**
 - One agents small contribution result in large aggregated benefits