

Copyleft and dual licensing for publicly funded software

Workshop on:
Implementing Open Source Software in Public Administrations:
Problems and Solutions
IDABC and Czech Ministry of Informatics
Prague, 11 April, 2006

Rishab Aiyer Ghosh (rishab@dxm.org)

UNU-MERIT
Maastricht, Netherlands

Definitions

- *Software*: when italicised, refers throughout this presentation to publicly funded software, either developed by a public agency (*funder*), or funded wholly or in part by a public agency but developed by a private entity (*contractor*).
- *Exploitation*: refers to commercial exploitation involving proprietary modification to *software*. Does not include commercial exploitation not based on such modification, e.g. services.

Definitions

- *Funder*: public agency providing funds, commissioning *software*, with influence over copyright and/or licensing terms
- *Contractor*: original creator of *software*, thanks to funds from *Funder*
- *Third party developer*: a third party, not funded by *funder*, who wishes to modify *software*
- *Community developer*: willing to modify *software* under a free software licence

Possible policy goals

1. *Software* should be available to the public, at no additional cost, for use and development
2. *Software* should be available to the public in perpetuity, including further developments to it
3. *Software* should be available for *exploitation* by the *contractor*
4. *Software* should be available for *exploitation* by *third party developers*.

Possible policy goals

1. *Software* should be available to the public, at no additional cost, for use and development
2. *Software* should be available to the public in perpetuity, including further developments to it
3. *Software* should be available for *exploitation* by the *contractor*
4. *Software* should be available for *exploitation* by *third party developers*.
5. (4), but without monitoring, further conditions or costs imposed by *funder* or *contractor*

Properties of licences

- Reciprocal (Copyleft) licences: e.g. GPL
- Permissive (non-copyleft) free software licences: e.g. BSD, Apache
- “Private” licences: e.g. for proprietary software, or designed to allow a single *third party developer* certain rights under certain conditions

Policy goals met by licences

Policy goal	Copyleft	Permissive	Private
1. public access	Strong	Yes	n.a.
2. (1) forever, for further developments	Strong	No	n.a.
3. <i>exploitation by contractor</i>	Yes	Weak	Yes
4. <i>exploitation by third party</i>	No	Strong	Yes
5. (4) without monitoring / control	No	Yes	No

The shaded area represents the policy goals achieved by dual licensing: Copyleft (GPL) + private licence

How does dual licensing work?

- *Software* is made available under a reciprocal licence (GPL)
- Access is available to the general public in perpetuity, including future modifications by...
- ...*Community developers* who can modify under the reciprocal licence
- *Contractor* or *funder* retains copyright over original *software*, and can *exploit*
- *Third party developer* can get a private licence from *funder* enabling *exploitation* without GPL reciprocity

Dual licensing dynamic effects

Time and version changes ----->		
Original work		
	Contribution from funded developer	
		Contribution from unfunded community developer
Key (each cell represents a component or modification of the software):		
	Copyright for this component of the software is held by the central agency, this component can be licensed to third parties for commercial exploitation	
	Copyright vests with unfunded community developer from the general public, this component cannot be licensed to third parties for commercial development but is still available for integrated use and further development by all community and funded developers.	