Business and Sustainability Models Around Free and Open Source Software

OUCS, 12 January 2009

JISC

What we will cover:

- Enforcement, Exclusions and Risks
- Software Patents and FOSS
- Sustainability and Business Models
- Some Project Examples
- Conclusion

JISC

Enforcement, Exclusions and Risks (1)

- Many within the FOSS community reject the idea that a FOSS licence is a contract
- This is mainly for practical reasons, as contract law varies widely between countries, and is relatively expensive to litigate.
- In comparison IP law and specifically copyright law is more uniform, being the subject of numerous international treaties
- They argue that there is thus no need for explicit acceptance – the licensor either abides by the conditions of the licence or has no licence
- Thus enforcement of licence conditions is traditionally undertaken by asserting copyright infringement (no licence) rather than contractual breach

JISC

Enforcement, Exclusions and Risks (2)

• Exclusions of warranty and limitations of liability in all current FOSS licences are drafted to be effective under US law. *Note: The GNU GPL v3 allows the inclusion of additional, regionalised exclusions and limitations.*

• Occasionally the legal basis for the enforceability of FOSS licences is challenged, for example **Wallace v. FSF**, where it was argued that the GPL was a form of price-fixing and **Jacobsen v. Katzer**, where it was argued that a FOSS licensor must rely on contract law to enforce their conditions (both failed)

• Many FOSS licences do not specify a jurisdiction in which disputes should be resolved (and some specify inappropriate jurisdictions)

JISC

Software Patents and FOSS

• Traditionally staff charged with exploiting software IP generated in UK Higher Education have considered the obtaining of software patents.

• Care needs to be taken when assessing FOSS exploitation and patent exploitation in proximity. In general, FOSS licensing will undermine most exploitable value in a software patent held on processes embodied in that software (although see below).

• This is because many FOSS licences explicitly license all patent rights necessary to use and distribute the software to everyone in the world, free of charge. Even those licences which do not do this explicitly can be strongly argued to do so implicitly under the legal doctrines of regions such as the US and UK.

JISC

Business and Sustainability Models

- These are mostly not mutually exclusive, and will most often be used in combination as appropriate – more accurately they are elements of business models
- This is still an emerging area of business practice
- Some of the current success of FOSS software exploitation techniques may be attributable to dissatisfaction with more traditional proprietary techniques and their associated big-name vendors, rather than any innate superiority
- It remains to be seen whether the current global financial difficulties will help FOSS business or hinder it. Analysts are currently predicting both.

JISC

First - what you cannot / should not do

- Charge for licences for specific uses of your code, for example commercial use (Open Source Definition point 6)
- Charge for licences in general (Possible but subject to low/zero-cost competition from all recipients)
- Tweak an existing FOSS licence for your purposes and still call your software Free Software or Open Source Software (Strong community rejection of these practices)
- Silently incorporate FOSS software in your proprietary offering without abiding by the licence conditions (detection is likely, and although legal damages are unlikely, damage to reputation is certain)

JISC

Academic Community Development

• FOSS licensing permits a varied group of contributors to work on software that addresses a particular problem domain.

• Institutions and their academics can gain public profile by contributing to such projects and becoming associated with respected tools in specific areas of research. It can also help ensure the continued existence of useful solutions.

• Examples include **BioImage Suite** (biological image analysis software) **YARP** (experimental robotics software) and **The Versioning Machine** (software for aligning differing versions of xml-encoded texts).

• Recognition for work on academic tools is still, however, some way behind more traditional forms of academic recognition for publication etc

JISC

Establishing a separate legal entity

- Adds to sustainability by isolating risks (IP infringement, event organisation, damages from failure) from the parent institution
- Facilitates donation of money and simplifies tax issues
- Most research institutions are already well-practised in setting up spin-out companies. In the case of sustaining FOSS projects some kind of not-for-profit entity may be just as or even more appropriate
- Such an entity can still have an affiliated commercial entity engaged in exploiting the software and the brands that it stewards

JISC

Moving into an external foundation

• The benefits of foundation status have led to the establishment of umbrella foundations holding multiple FOSS projects.

• Examples include the **Apache Software Foundation**, which supports Apache HTTP Server, Cocoon, Lucene, **Software in the Public Interest,** which supports the Debian Linux distribution and PostgreSQL, and the **Software Freedom Conservancy**, home to Samba, Busybox and Wine

• Entering an umbrella foundation can radically reduce running costs for projects that receive financial donations, as the foundation will handle the necessary book-keeping, as well as providing the risk management benefits that come with separating legal responsibility for a project from your host institution

JISC

'Community Source' Foundations

- Where a number of separate institutions see a benefit in jointly developing a piece of FOSS, they can adopt a model which has come to be known, somewhat confusingly, as 'Community Source'
- Each institution contributes resources to developing the code, the ownership of which rests in an external foundation
- In the initial phases the code may be unavailable outside the foundation, although it will eventually be released under a FOSS licence
- Contributing resources to the foundation buys institutions early code access and influence on the governance of the project and its functionality
- Mellon-funded projects **Sakai** and **Kuali** both began using this model

JISC

Consultancy

• Consultancy is another traditional technique for educational institutions looking to financially exploit their resources

• A more traditional model might be to sell licences to a piece of research-derived software and sell consultancy services and/or bespoke development services alongside it

• Potentially a FOSS release of the software can improve uptake, given its low cost of acquisition, and drive the market for associated consultancy and development services more successfully than the traditional model

JISC

Internal Cost Reduction

• Institutions may be happy to sustain an internally-developed FOSS project themselves if the project can demonstrate that it drives down the running costs of that institution or solves an institutional problem

• Projects that reduce costs in one institution may have good potential, when mature, to be deployed in others. This provides opportunities for paid consultancy and/or provision of the software as a service (see below)

JISC

Provision of Paid Support / Documentation

- Just because your code is freely available, it does not mean that the documentation or your help needs to be (as with the consultancy and bespoke development model)
- Support can be provided in time- or incident-limited bundles
- Support can be in the form of guaranteed performance on specific hardware
- Documentation can take the form of paid access to a knowledge base of previously resolved issues
- HOWEVER, in this case one is in competition with the software's user base/ community, who may be willing to provide peer support for free

JISC

Integration / Managed Upgrades

- Managing the integration of various FOSS technologies, with their varying dependencies and release cycles, is a service that people are prepared to pay for
- Similarly managing the deployment of upgrade patches can be a paid service
- Bundles of tested, integrated FOSS software can be sold along with, potentially, support agreements
- HOWEVER, close integration may trigger responsibilities in particularly copyleft licences that could prevent integrated distribution – read the licences

JISC

Competitor Disruption

• Sometimes a FOSS alternative to a competitor's product can disrupt their business model and provide competitive advantage (although this is almost never the sole motivation behind the release or distribution)

• Examples (arguably) include **Sun**'s OpenOffice.org, **Google**'s bundling and distribution of Microsoft-competing software such as OpenOffice.org, Firefox and Chrome (the 'Google Pack'), **Netscape Corporation**'s FOSS release of Netscape Navigator

JISC

Software as a Service

- Increasingly consumers are becoming comfortable with so-called 'cloud'-based software offerings – software that is accessed and used over the internet, and which stores data remotely from the user
- SaaS can be a useful solution to the problem of institutionally developed software that relies integrally on copyleft-licensed code
- Provision of service using copyleft software does not count as distribution, and thus does not trigger copyleft's reciprocal licensing responsibilities
- HOWEVER – this is a known 'bug' in copyleft licensing, and licences such as the GNU Affero GPL v3 are already in existence to 'fix' it.

JISC

Advertising / Referral

• Your software or accompanying web site may be able to direct network traffic to an entity that is willing to pay for hits (although of course this functionality can always be engineered out by technically apt users)

• This is **Mozilla Foundation**'s main source of income

• **Firefox**'s built-in search box directs queries to Google

• In 2007 'the vast majority' of **Mozilla Foundation**'s $75m revenue came from **Google** under this deal. They are now being investigate by the US IRS

• **Wordpress**, the FOSS blogging software and hosting platform raised $29.5 million in its last round of investment and is expected to move to this business model in the future

JISC

Training and Accreditation

- As well as support and consultancy, generalised training documents, courses and qualifications may be viable products
- Control of an associated trademark enables the provision of *'X-Certified Professional'* style programmes
- Actual training and examination are readily out-sourced

JISC

Trademark Licensing / Merchandising

• Just because your code is available under a FOSS licence, you do not have to permit universal use of your project's name and associated symbols

• Unlike copyright, trademarks are a registered form of IP, meaning that you have to apply to relevant government agencies for ownership. However, compared to patent application, trademark registration is relatively inexpensive

• Owning your trademark facilitates the sale of associated merchandise and accreditation and marks like “Powered by X” and “Using X technology”

• Can be a deterrent to forking if the brand is strong enough – the motivation to increase personal reputation by providing functionality outside project “*X*” is partially undermined by the inability to call the new project *“Improved X”*

JISC

IP Indemnification

• As the *SCO v. IBM* case showed, software licensees can sometimes be subject to unexpected threats of IP infringement action from third parties

• In traditional proprietary software licences the licensor will deal somehow with the issue of what happens when a third party claims that the licensed software contains some of their property (perhaps indemnifying the licensee against any resulting financial costs, or perhaps specifically declining to do so)

• Indemnification against third party claims can be a product sold alongside FOSS distributions

• Many corporate customers look for this kind of security in IT solutions

JISC

Proprietary Versions and Components

- Sometimes referred to disparagingly as the 'Bait and Switch' model
- A FOSS edition of software is offered which lacks some of the functionality of a paid edition, either throughout its code or in the form of missing proprietary components
- While the existence of better-supported or hardware-accredited forms of FOSS offerings is generally accepted by the FOSS community, proprietary components and versions are less well-liked (although there is perhaps growing acceptance as the community matures)
- HOWEVER, this is another example of competing with the community. The FOSS model means that anyone can produce freely available versions of your paid functionality, given enough time and expertise

JISC

Dual Licensing

- Provided that you have the necessary ownership or sub-licensing rights over your project's code, you can provide it under differing licences
- In the classic case, these would be a copyleft licence and a paid proprietary licence
- Customers who wish to build software product incorporating your code and who do not wish to use the copyleft licence must pay for the proprietary licence
- This is therefore most suitable for code which is readily susceptible to inclusion within commercial software products, for example database backends

JISC

'Patentleft'

- A variation on dual licensing
- Obtain patent on a software invention embodied in your software
- Release your code under a copyleft licence
- (Optional) Covenant not to assert patent rights against other FOSS software implementations, (perhaps with some exceptions – undistributed, distributed with hardware)
- Sell patent licences to interested parties who are neither protected by the covenant or prepared to accept the responsibilities of your copyleft licence.
- Used by **International Characters**, a spin-out company of Simon Fraser University in British Columbia

JISC

Examples: Red Hat

• **Red Hat** provides an enterprise Linux distribution, upgrades and many services in exchange for a periodic subscription fee. Red Hat also runs the Fedora project, an open development Linux version with many of the features of Red Hat Enterprise Linux.

• Provision of paid support

• IP Indemnification

• Training and Accreditation

• Managed Upgrades

JISC

Examples: MySQL

• **MySQL** provides a SQL database system under commercial or copyleft licences. **Sun** bought MySQL in 2008 for $1bn

- Provision of paid support
- Proprietary components (proposed)
- Consultancy
- Dual Licensing
- IP Indemnification (on commercial licence)
- Training and Accreditation
- Managed Upgrades

JISC

Examples: Exim

• **Exim**, the popular message transfer agent, began as an internal project within the computing services at the University of Cambridge in 1996, and has been in part supported by donation of staff resources by Cambridge since then.

• Internal Cost Reduction

• Training

JISC

Examples: Xensource (pre-2007)

• **Xen** is a Linux-based virtualisation solution developed at the University of Cambridge. After receiving VC funding the project sold tested distributions in combination with hardware vendors and incident-limited bundles of technical support. In 2007 Xensource was acquired by Citrix for $500m

• Provision of paid support

JISC

Conclusion

• 'FOSS' and 'commercial' are not antagonistic concepts - FOSS code is increasingly accepted as a necessary component of commercial software (IT consulting firm **Gartner** predicts FOSS will form some part of 80% of commercial software offerings by 2012*)

• Thus it is becoming increasingly important for educational software projects to understand FOSS licensing and exploitation if they are to reach sustainability

* http://gartner.com/it/page.jsp?id=593207

JISC

Questions?

JISC

OSS Watch

OSS Watch - http://www.oss-watch.ac.uk/
Free Software Foundation - http://www.fsf.org/
Open Source Initiative - http://www.opensource.org/
Bill Gates' letter to hobbyists - http://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists
The Cathedral and the Bazaar - http://www.catb.org/~esr/writings/cathedral-bazaar/
GNU GPLv3 - http://gplv3.fsf.org/
Wallace v. FSF - http://www.fsf.org/news/wallace-vs-fsf
Jacobsen v. Katzer - http://www.groklaw.net/articlebasic.php?story=20060514233436196
BioImage Suite - http://www.bioimagesuite.org/
YARP - http://eris.liralab.it/yarp/
The Versioning Machine - http://v-machine.org/
Apache Software Foundation - http://www.apache.org/
Software in the Public Interest - http://www.spi-inc.org/
Software Freedom Conservancy - http://conservancy.softwarefreedom.org/
Sakai - http://sakaiproject.org/portal
Kuali - http://kuali.org/
OpenOffice.org - http://www.openoffice.org/
Google Pack - http://pack.google.com/intl/en-gb/pack_installer.html
GNU Affero GPLv3 - http://www.fsf.org/licensing/licenses/agpl-3.0.html
Wordpress - http://wordpress.org/
SCO v. IBM - http://en.wikipedia.org/wiki/SCO_v._IBM
International Characters - http://www.international-characters.com/
Red Hat - http://www.redhat.com/
MySQL - http://www.mysql.com/
Exim - http://www.exim.org/
Xen - http://citrix.com/English/ps2/products/product.asp?contentID=683148

JISC

Business and Sustainability Models Around Free and Open Source Software

OUCS, 12 January 2009

JISC

SS Watch

What we will cover:

- Enforcement, Exclusions and Risks
- Software Patents and FOSS
- Sustainability and Business Models
- Some Project Examples
- Conclusion

JISC

Enforcement, Exclusions and Risks (1)

- Many within the FOSS community reject the idea that a FOSS licence is a contract
- This is mainly for practical reasons, as contract law varies widely between countries, and is relatively expensive to litigate.
- In comparison IP law and specifically copyright law is more uniform, being the subject of numerous international treaties
- They argue that there is thus no need for explicit acceptance – the licensor either abides by the conditions of the licence or has no licence
- Thus enforcement of licence conditions is traditionally undertaken by asserting copyright infringement (no licence) rather than contractual breach

JISC

OSS Watch

Enforcement, Exclusions and Risks (2)

• Exclusions of warranty and limitations of liability in all current FOSS licences are drafted to be effective under US law. *Note: The GNU GPL v3 allows the inclusion of additional, regionalised exclusions and limitations.*

• Occasionally the legal basis for the enforceability of FOSS licences is challenged, for example **Wallace v. FSF**, where it was argued that the GPL was a form of price-fixing and **Jacobsen v. Katzer**, where it was argued that a FOSS licensor must rely on contract law to enforce their conditions (both failed)

• Many FOSS licences do not specify a jurisdiction in which disputes should be resolved (and some specify inappropriate jurisdictions)

JISC

OSS Watch

Software Patents and FOSS

• Traditionally staff charged with exploiting software IP generated in UK Higher Education have considered the obtaining of software patents.

• Care needs to be taken when assessing FOSS exploitation and patent exploitation in proximity. In general, FOSS licensing will undermine most exploitable value in a software patent held on processes embodied in that software (although see below).

• This is because many FOSS licences explicitly license all patent rights necessary to use and distribute the software to everyone in the world, free of charge. Even those licences which do not do this explicitly can be strongly argued to do so implicitly under the legal doctrines of regions such as the US and UK.

JISC

OSS Watch

Business and Sustainability Models

• These are mostly not mutually exclusive, and will most often be used in combination as appropriate – more accurately they are elements of business models

• This is still an emerging area of business practice

• Some of the current success of FOSS software exploitation techniques may be attributable to dissatisfaction with more traditional proprietary techniques and their associated big-name vendors, rather than any innate superiority

• It remains to be seen whether the current global financial difficulties will help FOSS business or hinder it. Analysts are currently predicting both.

JISC

OSS Watch

First - what you cannot / should not do

- Charge for licences for specific uses of your code, for example commercial use (Open Source Definition point 6)
- Charge for licences in general (Possible but subject to low/zero-cost competition from all recipients)
- Tweak an existing FOSS licence for your purposes and still call your software Free Software or Open Source Software (Strong community rejection of these practices)
- Silently incorporate FOSS software in your proprietary offering without abiding by the licence conditions (detection is likely, and although legal damages are unlikely, damage to reputation is certain)

JISC

OSS Watch

Academic Community Development

• FOSS licensing permits a varied group of contributors to work on software that addresses a particular problem domain.

• Institutions and their academics can gain public profile by contributing to such projects and becoming associated with respected tools in specific areas of research. It can also help ensure the continued existence of useful solutions.

• Examples include **BioImage Suite** (biological image analysis software) **YARP** (experimental robotics software) and **The Versioning Machine** (software for aligning differing versions of xml-encoded texts).

• Recognition for work on academic tools is still, however, some way behind more traditional forms of academic recognition for publication etc

JISC

OSS Watch

Establishing a separate legal entity

- Adds to sustainability by isolating risks (IP infringement, event organisation, damages from failure) from the parent institution
- Facilitates donation of money and simplifies tax issues
- Most research institutions are already well-practised in setting up spin-out companies. In the case of sustaining FOSS projects some kind of not-for-profit entity may be just as or even more appropriate
- Such an entity can still have an affiliated commercial entity engaged in exploiting the software and the brands that it stewards

JISC

OSS Watch

Moving into an external foundation

• The benefits of foundation status have led to the establishment of umbrella foundations holding multiple FOSS projects.

• Examples include the **Apache Software Foundation**, which supports Apache HTTP Server, Cocoon, Lucene, **Software in the Public Interest,** which supports the Debian Linux distribution and PostgreSQL, and the **Software Freedom Conservancy**, home to Samba, Busybox and Wine

• Entering an umbrella foundation can radically reduce running costs for projects that receive financial donations, as the foundation will handle the necessary book-keeping, as well as providing the risk management benefits that come with separating legal responsibility for a project from your host institution

JISC

OSS Watch

'Community Source' Foundations

- Where a number of separate institutions see a benefit in jointly developing a piece of FOSS, they can adopt a model which has come to be known, somewhat confusingly, as 'Community Source'
- Each institution contributes resources to developing the code, the ownership of which rests in an external foundation
- In the initial phases the code may be unavailable outside the foundation, although it will eventually be released under a FOSS licence
- Contributing resources to the foundation buys institutions early code access and influence on the governance of the project and its functionality
- Mellon-funded projects **Sakai** and **Kuali** both began using this model

JISC

OSS Watch

Consultancy

• Consultancy is another traditional technique for educational institutions looking to financially exploit their resources

• A more traditional model might be to sell licences to a piece of research-derived software and sell consultancy services and/or bespoke development services alongside it

• Potentially a FOSS release of the software can improve uptake, given its low cost of acquisition, and drive the market for associated consultancy and development services more successfully than the traditional model

JISC

OSS Watch

Internal Cost Reduction

• Institutions may be happy to sustain an internally-developed FOSS project themselves if the project can demonstrate that it drives down the running costs of that institution or solves an institutional problem

• Projects that reduce costs in one institution may have good potential, when mature, to be deployed in others. This provides opportunities for paid consultancy and/or provision of the software as a service (see below)

JISC

OSS Watch

Provision of Paid Support / Documentation

- Just because your code is freely available, it does not mean that the documentation or your help needs to be (as with the consultancy and bespoke development model)
- Support can be provided in time- or incident-limited bundles
- Support can be in the form of guaranteed performance on specific hardware
- Documentation can take the form of paid access to a knowledge base of previously resolved issues
- HOWEVER, in this case one is in competition with the software's user base/ community, who may be willing to provide peer support for free

JISC

OSS Watch

Integration / Managed Upgrades

- Managing the integration of various FOSS technologies, with their varying dependencies and release cycles, is a service that people are prepared to pay for
- Similarly managing the deployment of upgrade patches can be a paid service
- Bundles of tested, integrated FOSS software can be sold along with, potentially, support agreements
- HOWEVER, close integration may trigger responsibilities in particularly copyleft licences that could prevent integrated distribution – read the licences

JISC

Competitor Disruption

• Sometimes a FOSS alternative to a competitor's product can disrupt their business model and provide competitive advantage (although this is almost never the sole motivation behind the release or distribution)

• Examples (arguably) include **Sun**'s OpenOffice.org, **Google**'s bundling and distribution of Microsoft-competing software such as OpenOffice.org, Firefox and Chrome (the 'Google Pack'), **Netscape Corporation**'s FOSS release of Netscape Navigator

JISC

Software as a Service

• Increasingly consumers are becoming comfortable with so-called 'cloud'-based software offerings – software that is accessed and used over the internet, and which stores data remotely from the user

• SaaS can be a useful solution to the problem of institutionally developed software that relies integrally on copyleft-licensed code

• Provision of service using copyleft software does not count as distribution, and thus does not trigger copyleft's reciprocal licensing responsibilities

• HOWEVER – this is a known 'bug' in copyleft licensing, and licences such as the GNU Affero GPL v3 are already in existence to 'fix' it.

JISC

OSS Watch

Advertising / Referral

• Your software or accompanying web site may be able to direct network traffic to an entity that is willing to pay for hits (although of course this functionality can always be engineered out by technically apt users)

• This is **Mozilla Foundation**'s main source of income

• **Firefox**'s built-in search box directs queries to Google

• In 2007 'the vast majority' of **Mozilla Foundation**'s $75m revenue came from **Google** under this deal. They are now being investigate by the US IRS

• **Wordpress**, the FOSS blogging software and hosting platform raised $29.5 million in its last round of investment and is expected to move to this business model in the future

JISC

Training and Accreditation

- As well as support and consultancy, generalised training documents, courses and qualifications may be viable products
- Control of an associated trademark enables the provision of *'X-Certified Professional'* style programmes
- Actual training and examination are readily out-sourced

JISC

OSS Watch

Trademark Licensing / Merchandising

• Just because your code is available under a FOSS licence, you do not have to permit universal use of your project's name and associated symbols

• Unlike copyright, trademarks are a registered form of IP, meaning that you have to apply to relevant government agencies for ownership. However, compared to patent application, trademark registration is relatively inexpensive

• Owning your trademark facilitates the sale of associated merchandise and accreditation and marks like “Powered by X” and “Using X technology”

• Can be a deterrent to forking if the brand is strong enough – the motivation to increase personal reputation by providing functionality outside project *“X”* is partially undermined by the inability to call the new project *“Improved X”*

JISC

OSS Watch

IP Indemnification

• As the *SCO v. IBM* case showed, software licensees can sometimes be subject to unexpected threats of IP infringement action from third parties

• In traditional proprietary software licences the licensor will deal somehow with the issue of what happens when a third party claims that the licensed software contains some of their property (perhaps indemnifying the licensee against any resulting financial costs, or perhaps specifically declining to do so)

• Indemnification against third party claims can be a product sold alongside FOSS distributions

• Many corporate customers look for this kind of security in IT solutions

JISC

OSS Watch

Proprietary Versions and Components

- Sometimes referred to disparagingly as the 'Bait and Switch' model
- A FOSS edition of software is offered which lacks some of the functionality of a paid edition, either throughout its code or in the form of missing proprietary components
- While the existence of better-supported or hardware-accredited forms of FOSS offerings is generally accepted by the FOSS community, proprietary components and versions are less well-liked (although there is perhaps growing acceptance as the community matures)
- HOWEVER, this is another example of competing with the community. The FOSS model means that anyone can produce freely available versions of your paid functionality, given enough time and expertise

JISC

OSS Watch

Dual Licensing

- Provided that you have the necessary ownership or sub-licensing rights over your project's code, you can provide it under differing licences
- In the classic case, these would be a copyleft licence and a paid proprietary licence
- Customers who wish to build software product incorporating your code and who do not wish to use the copyleft licence must pay for the proprietary licence
- This is therefore most suitable for code which is readily susceptible to inclusion within commercial software products, for example database backends

JISC

OSS Watch

'Patentleft'

- A variation on dual licensing
- Obtain patent on a software invention embodied in your software
- Release your code under a copyleft licence
- (Optional) Covenant not to assert patent rights against other FOSS software implementations, (perhaps with some exceptions – undistributed, distributed with hardware)
- Sell patent licences to interested parties who are neither protected by the covenant or prepared to accept the responsibilities of your copyleft licence.
- Used by **International Characters**, a spin-out company of Simon Fraser University in British Columbia

JISC

Examples: Red Hat

• **Red Hat** provides an enterprise Linux distribution, upgrades and many services in exchange for a periodic subscription fee. Red Hat also runs the Fedora project, an open development Linux version with many of the features of Red Hat Enterprise Linux.

• Provision of paid support

• IP Indemnification

• Training and Accreditation

• Managed Upgrades

JISC

Examples: MySQL

• **MySQL** provides a SQL database system under commercial or copyleft licences. **Sun** bought MySQL in 2008 for $1bn

- Provision of paid support
- Proprietary components (proposed)
- Consultancy
- Dual Licensing
- IP Indemnification (on commercial licence)
- Training and Accreditation
- Managed Upgrades

JISC

Examples: Exim

• **Exim**, the popular message transfer agent, began as an internal project within the computing services at the University of Cambridge in 1996, and has been in part supported by donation of staff resources by Cambridge since then.

• Internal Cost Reduction

• Training

JISC

Examples: Xensource (pre-2007)

• **Xen** is a Linux-based virtualisation solution developed at the University of Cambridge. After receiving VC funding the project sold tested distributions in combination with hardware vendors and incident-limited bundles of technical support. In 2007 Xensource was acquired by Citrix for $500m

• Provision of paid support

JISC

Conclusion

• 'FOSS' and 'commercial' are not antagonistic concepts - FOSS code is increasingly accepted as a necessary component of commercial software (IT consulting firm **Gartner** predicts FOSS will form some part of 80% of commercial software offerings by 2012*)

• Thus it is becoming increasingly important for educational software projects to understand FOSS licensing and exploitation if they are to reach sustainability

* http://gartner.com/it/page.jsp?id=593207

JISC

OSS Watch

Questions?

JISC

OSS Watch

OSS Watch - http://www.oss-watch.ac.uk/
Free Software Foundation - http://www.fsf.org/
Open Source Initiative - http://www.opensource.org/
Bill Gates' letter to hobbyists - http://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists
The Cathedral and the Bazaar - http://www.catb.org/~esr/writings/cathedral-bazaar/
GNU GPLv3 - http://gplv3.fsf.org/
Wallace v. FSF - http://www.fsf.org/news/wallace-vs-fsf
Jacobsen v. Katzer - http://www.groklaw.net/articlebasic.php?story=20060514233436196
BioImage Suite - http://www.bioimagesuite.org/
YARP - http://eris.liralab.it/yarp/
The Versioning Machine - http://v-machine.org/
Apache Software Foundation - http://www.apache.org/
Software in the Public Interest - http://www.spi-inc.org/
Software Freedom Conservancy - http://conservancy.softwarefreedom.org/
Sakai - http://sakaiproject.org/portal
Kuali - http://kuali.org/
OpenOffice.org - http://www.openoffice.org/
Google Pack - http://pack.google.com/intl/en-gb/pack_installer.html
GNU Affero GPLv3 - http://www.fsf.org/licensing/licenses/agpl-3.0.html
Wordpress - http://wordpress.org/
SCO v. IBM - http://en.wikipedia.org/wiki/SCO_v._IBM
International Characters - http://www.international-characters.com/
Red Hat - http://www.redhat.com/
MySQL - http://www.mysql.com/
Exim - http://www.exim.org/
Xen - http://citrix.com/English/ps2/products/product.asp?contentID=683148

JISC