Open Source, Cloud Computing and IPRs

Noam Shemtov
n.shemtov@qmul.ac.uk

Centre for Commercial Law Studies
Queen Mary

Free & Open Source Software

What is Free/Open Source?

- A philosophy
- A methodology
- A set of licences
- A business model

Intro to software development

- Source code

```
//========================================================================
//                the list of all the possible states for the current FSM
//========================================================================
enum STATE{ START, INT, FLOAT, SCIENTIFIC, EXPONENT, S1, S2, INVALID } state;

STATE Transition( char *str );
void PrintState( STATE state );

int main() {
  // declaring buffer variable
  char buffer[32] = {0};
  // getting input from the user
  cout << "\nPlease enter a number: ";
  cin.getline( buffer, 32 );
  // compute final state
  STATE FINAL_STATE = Transition(buffer);
  // prints the final state
  PrintState(FINAL_STATE);
  return 0;
}
```

- Object code
  - 00101001011101010010100100010101011111

Intro to software development

- Modern software rarely written from scratch
- Usually an assemblage of modules, with code gluing them together
- Methods of combining code/modules
  - Cutting and pasting
  - Linking (static/dynamic)
  - Plug-ins

Modern Software Development

- Code from many different sources is likely to mean one codebase contains many different copyright owners.
- Licensing structure can be complex
  - Licence/sub-licence
  - Parallel licences
  - Co-ownership

Proprietary licences typically...

- Limit use to specified (number of) computers
- Restrict number of users
- Restrict types of use (e.g. home/student)
- Restrict jurisdiction
- Restrict assignment/transfer
- (Attempt to) restrict ability to reverse engineer
- Require payment of fees

Free Software Foundation

- Four Freedoms
  - The freedom to run the program, for any purpose (freedom 0).
  - The freedom to study how the program works, and change it to make it do what you wish (freedom 1). Access to the source code is a precondition for this.
  - The freedom to redistribute copies so you can help your neighbor (freedom 2).
  - The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

http://www.gnu.org/philosophy/free-sw.html

Open Source Initiative – OSI Definition

1. **Free Redistribution**
2. **Source Code**
3. **Derived Works**
4. **Integrity of The Author's Source Code**
5. **No Discrimination Against Persons or Groups**
6. **No Discrimination Against Fields of Endeavor**
7. **Distribution of License**
8. **License Must Not Be Specific to a Product**
9. **License Must Not Restrict Other Software**
10. **License Must Be Technology-Neutral**

http://www.opensource.org/docs/osd

Examples in the Real World

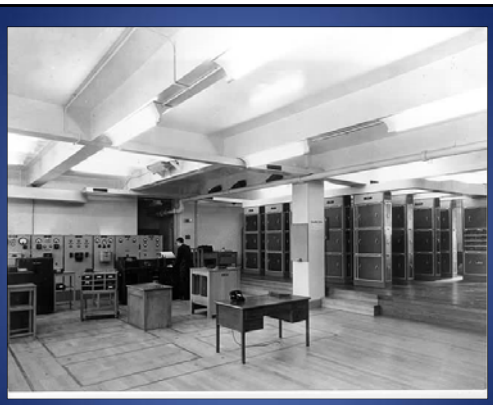
- Google
  - "Every time you're using Google, you're using Linux"
- Amazon
  - Marketplace infrastructure and EC2
- Apache
- GNU/Linux (Ubuntu, Red Hat)

Some Business Statistics

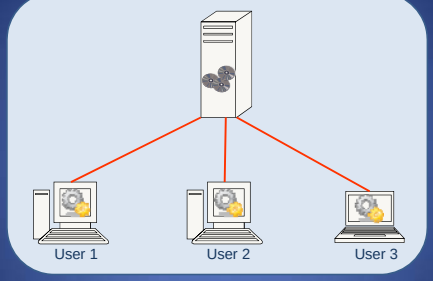
- MySQL/Sun - $1Bn acquisition
- Oracle/Sun - $7Bn acquisition
- Red Hat turnover > $500m

Why is it big business?

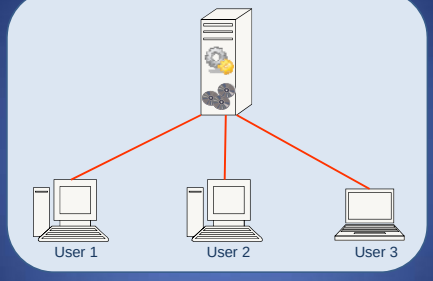
- Big software projects are mainly (maybe 80%) services, and not licence fees
- Use of FOSS reduces R&D costs, through not reinventing the wheel
- Bruce Perens: it's commodity code that tends to become OS, people can charge for the frills
- Also can be regarded as collaborative R&D
- Efficient and inexpensive



Cloud Computing



Where applications reside on the client, with the server hosting shared data, we have a **thick client** system.



Where both applications and data reside on the server, we have a **thin client** system.

Thick Client Networks

- In thick client networks, the computing power is at the 'edge' of the network, close to the user.
- Advantages:
  - The user has full use of the computing power of the client.
  - The user can process data locally without needing a continuous connection.
- Disadvantages and Issues:
  - Multiple access to the same data can cause database consistency problems.
  - Each local client requires a distinct copy of (and licence for) each application.
  - Application updates must be applied to each client.

Thin Client Networks

- In thin client networks, the computing power is concentrated in the core server. Data is processed in the server and only results are sent over the network.
- Advantages:
  - Only one copy of an application (although with a multiple-user licence) is required.
  - Application updates can be applied centrally.
  - It can be easier to ensure data consistency.
- Disadvantages and Issues:
  - Requires a continuous network connection to allow the user to process data.
  - The user has to share the computing power of the server.

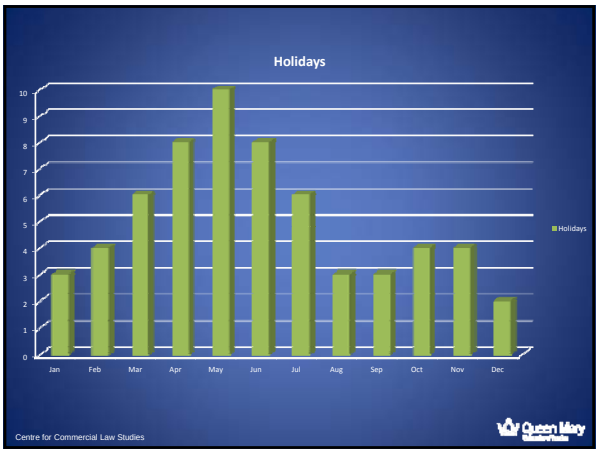
Historical Trends

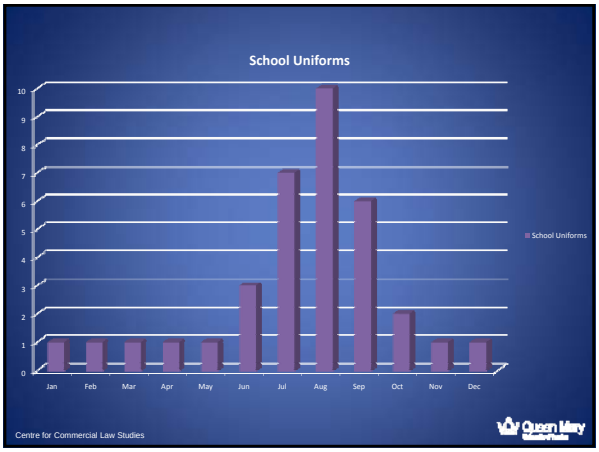
- The history of networked computing tends to show a cycle between thick and thin client networks.
- The availability of powerful clients encourages use of thick client solutions.
- The availability of fast and resilient network connections encourages the use of thin client solutions.
- At different times, one trend has overtaken the other, resulting in shifts from thick to thin clients and *vice versa*.

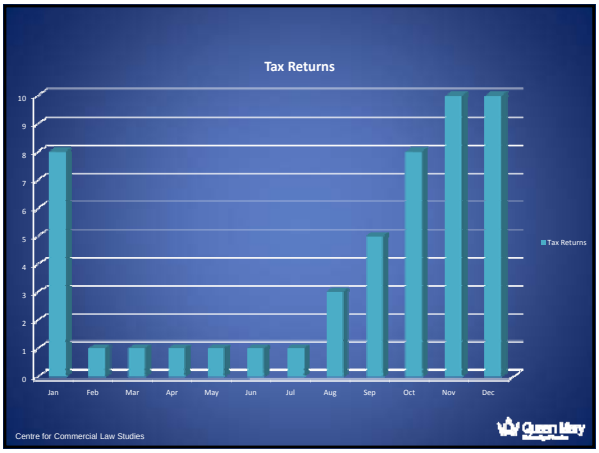
Historical Trends

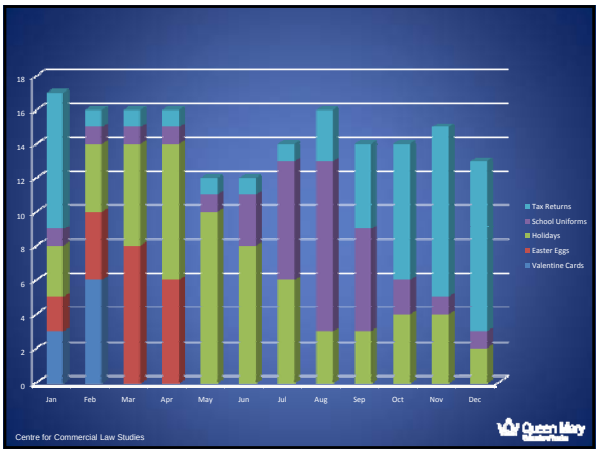
- Mainframe Era:
  - Applications concentrated in dedicated mainframes.
  - Remote access, if it existed, via 'dumb' terminal.
- Microcomputer Era:
  - Applications concentrated in desktop machines.
  - Remote access via LAN or dial-up to remote databases.
- Network Era:
  - Growth of data servers and then data centres.
  - Increasing overhead of desktop administration.

Centre for Commercial Law Studies


Holidays
10
9
8
7
6
5
4
3
2
1
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Holidays
Centre for Commercial Law Studies

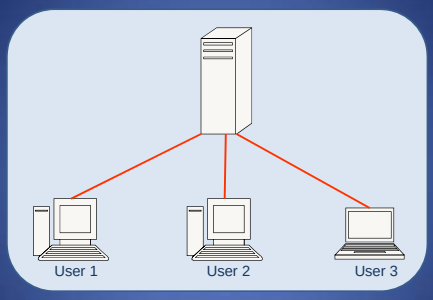

School Uniforms
10
9
8
7
6
5
4
3
2
1
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
School Uniforms
Centre for Commercial Law Studies


Tax Returns
10
9
8
7
6
5
4
3
2
1
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Tax Returns
Centre for Commercial Law Studies

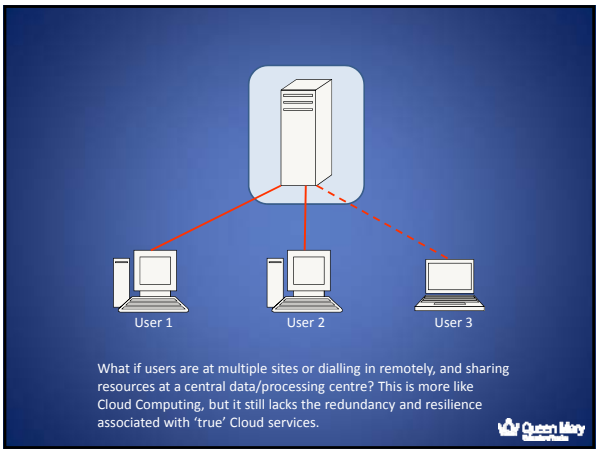


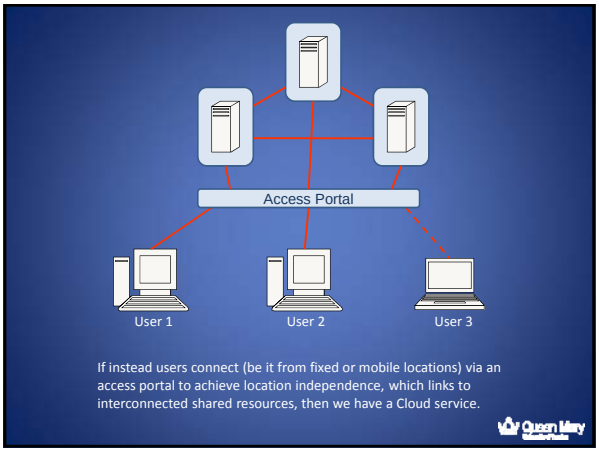
STRUCTURE

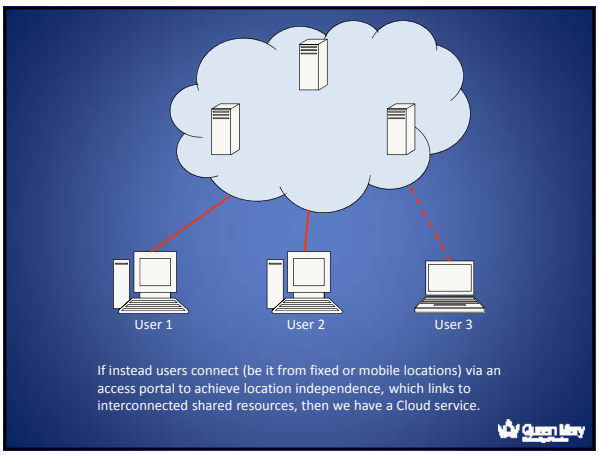
Centre for Commercial Law Studies

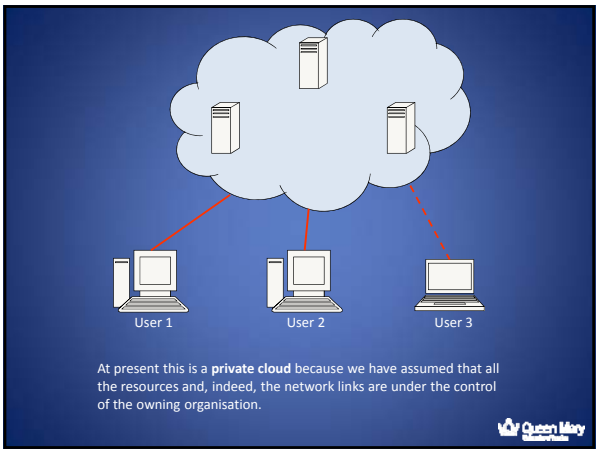


If all users and computing resources are collocated then this is not Cloud Computing – it's just a traditional IT infrastructure.

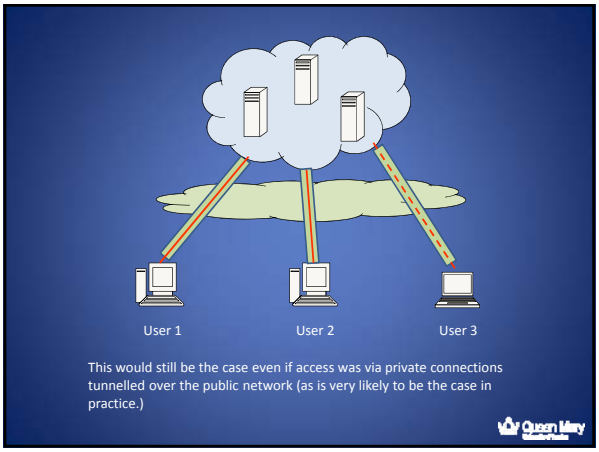

User 1
User 2
User 3
What if users are at multiple sites or dialling in remotely, and sharing resources at a central data/processing centre? This is more like Cloud Computing, but it still lacks the redundancy and resilience associated with 'true' Cloud services.


Access Portal
User 1
User 2
User 3
If instead users connect (be it from fixed or mobile locations) via an access portal to achieve location independence, which links to interconnected shared resources, then we have a Cloud service.

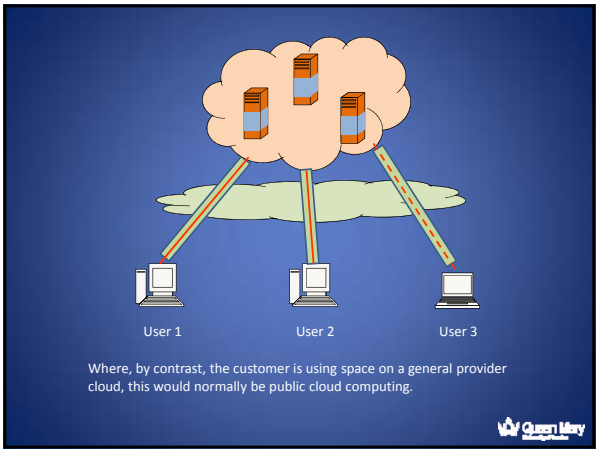

User 1
User 2
User 3
If instead users connect (be it from fixed or mobile locations) via an access portal to achieve location independence, which links to interconnected shared resources, then we have a Cloud service.



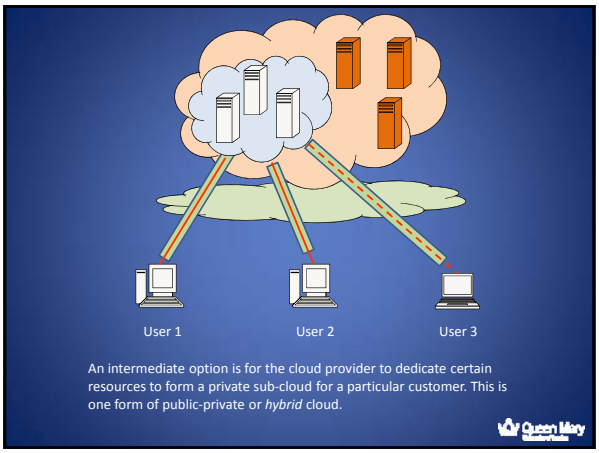
At present this is a **private cloud** because we have assumed that all the resources and, indeed, the network links are under the control of the owning organisation.



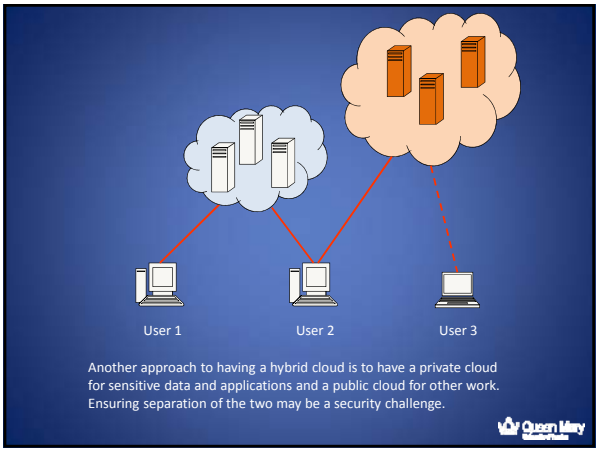
This would still be the case even if access was via private connections tunnelled over the public network (as is very likely to be the case in practice.)



Where, by contrast, the customer is using space on a general provider cloud, this would normally be public cloud computing.



An intermediate option is for the cloud provider to dedicate certain resources to form a private sub-cloud for a particular customer. This is one form of public-private or *hybrid* cloud.



Another approach to having a hybrid cloud is to have a private cloud for sensitive data and applications and a public cloud for other work. Ensuring separation of the two may be a security challenge.

"X as a Service"

Software as a Service
Platform as a Service
Infrastructure as a Service

also...
Storage as a Service
Applications as a Service
Data as a Service

and so on...

Open Source + Cloud Computing?

Using the Advantages of OS for the provision of Cloud solutions

- An example – OpenStack
  - a collection of open source technology products delivering a scalable, secure, standards-based cloud computing software solution
  - All of the code for OpenStack is freely available under the Apache 2.0 license. Anyone can run it, build on it, or submit changes back to the project. The basic premise is that an open development model is the best way to foster badly-needed cloud standards, remove the fear of proprietary lock-in for cloud customers, and create a large ecosystem that spans cloud providers.
  - Who uses: Corporations, service providers, VARS, SMBs, researchers, and global data centers looking to deploy large-scale cloud deployments for private or public clouds leveraging the support and resulting technology of a global open source community.

Using the Advantages of OS for the provision of Cloud solutions

- Open source software provides the foundations for many cloud implementations.
- AGPL v.3 released by FSF with the view to plug in a loophole in the ordinary GPL according to which the copyleft provision apply only when the software is distributed, but not when used. AGPL contains an additional provision triggering the copyleft provision in case of use of the software over computer network.

Presentation Concludes

Thank you for your time

n.shemtov@qmul.ac.uk