Ubiquitous Computing
Assignment

Peer-to-Peer Model and Client Server Model
Which one is better?

DT230 Shuo Liang D05109889  DT202 Joshue O Connor D07118219
No server, no client

In terms of Ubiquitous computing we compare the two kinds of architecture and examine the use of both Peer-to-Peer (P2P) and Client Server models (CSM) within the domain of Ubiquitous Computing and its applications. Which is better in the world of Ubiquitous Competing?
Ubiquitous computing will bring the Internet into our daily lives.

To find out we examine the various strengths and weaknesses of both within the ubiquitous computing domain, and give an overview of where the models work and are suitable (such as P2P active nodes in virtual environments or CSM in secure data transfer and validation in the mobile space).
With no effort...

- Instead of keeping lists of pertinent URLs or 'favourite places' on our browsers, the devices that need the information can find it themselves.

- Ubiquitous computing is the new “brave new world” of computing. [This includes including mobile communication Technology, tiny computing devices, new operating system technology and software models that can deal with intermittent connectivity and thin clients.]
Actually, ubiquitous computing is a natural extension of network computing which enables not only personal computers, but also to other small intelligent embedded devices that can be connected to the network.
But what about real-world applications of Ubiquitous Computing today?

One of the main ideals of Ubiquitous computing is to bring about an age of transparent and invisible computing.

In order for Ubiquitous computing to work, computing devices will become smaller and smaller, and easy for people anywhere to use and even wear.
Properties of Ubi Comp

- Some of the main properties of Ubiquitous computing is that the nodes are:
  - Everywhere.
  - Are very small.
  - Are aware.
Marc Weiser sez..

“Dwelling with computers means that they have their place, and we ours, and we co-exist comfortably. Unfortunately, our existing metaphors for computers are inadequate to describe the 'dwelling' relationship. Over the next twenty years computers will inhabit the most trivial things: clothes labels (to track washing), coffee cups (to alert cleaning staff to clean cups), light switches (to save energy if no one is in the room), and pencils (to digitise everything we draw). In such a world, we must dwell with computers, not just interact with them."
So what does that mean?

- Ubiquitous computing is a technology that will be an “aware aid”

- It is predicated that there are different stages to the adoption of Ubiquitous computing. This will probably be kicked off by a stage where people are more aware of the computers and technology that is all around them. Then the devices themselves will gradually get smaller and smaller, until they become invisible, embedded in the very fabric of the environment.
Key areas..

- **Low Power Devices**: Ubiquitous devices must be inexpensive, compatible low-power devices.

- **Low Cost**: The hardware necessary to create suitable Ubiquitous computing devices could herald an age of small, disposable computers;

- **Mobility**: A Ubiquitous computing network will have to support mobile applications and devices, as well as having some kind of future proof capabilities for other platforms not yet envisaged;

- **Mobile IP**: A mobile aware IP protocol that can suit a Ubiquitous computing environment.

- **Effective User Interfaces**: Ubiquitous computing promises an age of invisible user interfaces. They will also have to be able to be used by users, who are blind, vision impaired, have limited physical mobility and cognitive issues. These are areas of real challenges for Ubiquitous computing technologies.
**Implications of Ubiquitous computing**

- So what are some of the implications of Ubiquitous computing? Will society want its activities logged, with the potential more invasive monitoring that that implies?

- Society is already coming to terms with the impact of the web on its very fabric.

- Is the age of Ubiquitous computing a natural progression?
Privacy and Ethical issues in Ubiquitous computing

- How will the data in a Ubiquitous computing environment be stored? Who will own it? The owners of the network that was temporarily used by a user or the user themselves?

- What about when the Ubiquitous computing environment is used to break the law?
Client/Server Model Overview

- The client-server model of computing is a form of distributed processing used for accessing shared server side capabilities.

- Typically, a client will make a request for a certain service etc and the server will provide this request.
There are generally two types of client, both "Fat" and "Thin". A Fat client is where there are fewer calls to the server and there is much processing done on the client side. A thin client is where most of the processing etc is done on the server side, meaning that the client is particularly light (in terms of processing power, resources and data retention).
This client-server model is used where both parties are well known to each other and therefore trust one another. Trust is a huge part of success in pervasive computing and it at the core of successful client-server model interactions.
Client/Server Model Overview

- Some examples of current clients are:

  1) Web browsers such as Opera, Firefox, and Safari etc.

  2) E-mail clients such as Thunderbird, Mail etc. The client allows you to receive the mail.

  3) When using torrent-based services, then your computer is a client.
Client/Server Model Overview

1) File servers are computers on a network which store programs and data shared by the users of the network.

2) A print server is device that connects to one or more printers and accepts print jobs from clients on the network.

3) A mail server is used to transfer mails from one computer to another. It receives incoming mail, and manages outgoing mail.

4) Web servers are programs which accept HTTP requests from clients, and provide them with a HTML document, images or other types of files. All the internet traffic will pass through the web server.
Client/Server Model Overview

- These client server interactions are often performed sequentially, so they can be an unresponsive model prone to delays, dependent on the network capacity etc. There is therefore often a need for a mechanism to protect server based computing resources.

![Diagram of Client-Server processing](image)

*Figure 2: Client – Server Processing*
Client/Server Advantages

- An application server running when the data load is light - when some operation needs to be carried out, the client automatically finds a suitable server program and processes the client request in accordance with predetermined rules for response. The server then returns a result to the client. All the processing is done on the server side; therefore the load on the client is light.
Client/Server Advantages

- **Data storage management functions are more transparent** – where procedures are "transparent", and they do not interfere (usually cannot interfere with) the underlying process, we can complete our tasks. In the client-server architecture applications, the foreground process is not very "thin", so troublesome things can happen to the server and network. In the Client / Server system, the database cannot really become public such as in specialised warehouses, which are independent and need constant management.
Current use of established corporate computer networks - a central server handles all security and file information, allows functions and responsibilities of a computing system to be distributed, nodes are known to each other only via this network this means greater ease of maintenance and improved security. Data storage is therefore centralised, so updates to that data are far easier to administer than what would be possible under a P2P model.
Client/Server Disadvantages

- High maintenance costs
- Large initial investment to set it up.
- The traditional Client / Server structure of the software may require different operating systems for the different versions of the software or to insure interoperability.
- Server overload potential.
Peer-to-Peer Model

- P2P is a technology that enables a computing environment where peers (users’ end-systems) connect to each other without central connection points (such as servers in the client-server model).

- Interaction directly between personal computers sharing resources and services

- Each node is both a client and a server. (or a client- to client model)
P2P (in terms of modern file-sharing) can be defined as a technology enabling users to share communications, processing power, and data files with other users. P2P, if used properly, can prove beneficial to the end users and bypasses the traditional central server model. Your computer becomes a “node” in the P2P network, so it becomes the server. Each node in the network is a server where user can share content.
Peer-to-Peer Model

Napster Protocol

Napster Client

Napster Client

Napster Central Index Server

Napster Client

Your Computer

query: “debaser.mp3”

“I’ve got it!”

© 2005 HowStuffWorks
Peer-to-Peer Model

Gnutella

query: "Baby Go Home.mp3"

The file-transfer load is distributed between the computers exchanging files, but file searches and transfers from your computer to others can cause bottlenecks.

8,000 - 10,000 computers

"I've got it!"
Peer-to-Peer Model

Bit Torrent

BitTorrent tracker identifies the swarm and helps the client software trade pieces of the file you want with other computers.

Computer with BitTorrent client software receives and sends multiple pieces of the file simultaneously.
What's wrong with P2P?

- Traditionally P2P models are viewed with certain distrust.
- Needs a coherent adaptive trust model for measuring and comparing the trustworthiness of peers based on a transaction-based feedback system.
The politics of trust

There are three basic trust parameters:

1) The Feedback a peer receives through its transactions with other peers, are incorporated into the total number of transactions.

2) The credibility of the feedback sources into the model for evaluating the trustworthiness of peers.

3) A valid self regulation model.
Advantages

- The P2P model is “Decentralised”. Resources and services of P2P network are distributed all over the nodes.

- Transmission between the nodes without intermediate links and servers, so no bottlenecks.

- Scalability

- High performance, each node in the P2P system brings with it certain resources (Processing ability, bandwidth or storage space).
Advantages

- **Cost sharing / reduction.** – particularly set up costs.

- **Reliability** – hack resistant, can adjust to node failure.

- **Dynamic** - resources, services and so on are constantly interacting. In the P2P networks, resources entering and leaving each computing node do not need to go via a central server, which will save a lot of time and processes. For an efficient dynamic network, P2P is very suitable.
Disadvantages

- **Manageability issues** - In the future the sheer scale of a large P2P network, may result in it becoming unmanageable, unless the nodes become more self aware.

- **Security issues.** In the P2P network, the server is no longer the centre of the network, as each node should in itself be a server but the work of the whole network should still be co-ordinate somehow.
Applications Within Ubiquitous Computing
Peer to Peer handheld devices are a new platform for mobile computing. The specific applications, interfaces, design features, and infrastructure that will best support the development and delivery of a pervasive mobile computing environment.

- **Chord Protocol.** A distributed lookup protocol that addresses this problem. The Chord protocol provides a given key, it then maps the key onto every node within a network.
Difficult problems during developing P2P System or Application. (Using Chord)

- **Load balance**
- **Decentralisation** - Chord is fully distributed
- **Scalability** - The cost of a Chord lookup grows as the log of the number of nodes
- **Availability**
- **Cooperative Mirroring** - multiple providers
- **Time-shared storage** - for nodes with intermittent connectivity
- **Distributed indexes** - to support Gnutella- or Napster-like keyword search.
- **Large-Scale combinatorial search** - such as code breaking
Client Side Computing in Mobile Environments

- **Mobile computing** is where users carrying small compact devices have access to data and information services regardless of location.

- Three Main research areas in a mobile client-server system.
  - **Mobile-aware adaptation** - A mobile system must constantly adapt to changes in the environment and to the allocated resources available.
  - **Extended client-server model** - a facilitator of the mobile client server and thereby provides way designed explicitly to enable mobile computing.
  - **Mobile data access**
Different Frameworks

- **D-Stampede Framework** - Support different system in a Ubiquitous Computing environment.
  - Key features: 1) Indexing data streams temporally. 2) Correlating different data streams temporally. 3) Performing automatic distributed garbage collection of unnecessary stream data, 4) Supporting high performance

- **DIVE**: Distributed Interactive Virtual Environment. Designed to scale with a large number of simultaneous participants, while ensuring maximum interaction at each sit.
Ubiquitous Computing Application

- Trust in a peer-to-peer information system.
- Context sensitive middleware
- Scalable peer-to-peer network virtual environment
- Super-peer model for resource discovery service
Conclusion

- P2P has the upper hand in terms of dominance in the Ubiquitous Computing environment.

- Client server model is redundant within the Ubiquitous Computing space.

- In the future it may be that networks need to adopt either model as needed, or as the resources and demands of the users require.