

Online Communities and Client-Server Architectures

Rui Guerra (pt/nl)

V2, Institute for the Unstable Media
Software architect
rui@v2.nl

Since early 2000, the World Wide Web has become a common place for online communities to collaborate and exchange information. Most of that exchange occurs under the auspices of equal participation on platforms that in fact are privately owned and controlled. In his article 'The social web and its social contracts' Michel Bauwens states such conflict clearly: "The social web facilitates an unprecedented level of social sharing, but it does so mostly through the vehicle of proprietary platforms." (Bauwens 2008)

The centralization of control existing in communities based on the Web is not specific to some platforms but it is inherent to the Web architecture. The Web, as we know it today, is based on a client-server architecture using the Hypertext Transfer Protocol (HTTP) as a communication standard. The power relations existing on the web have been extensively analyzed by Alexander Galloway on his book 'Protocol – How control exists after decentralization'. For the sake of brevity, this text focuses on the less discussed aspect of server-client architectures. The server is typically a computer running an application that responds to several client requests. The word 'client' can be seen as an euphemism for slave, as the 'client' is always limited by the server defined rules. The exchange of information, that participants of the so called social web experience, is in fact an illusion of direct communication since any interaction between participants is mediated by servers. Therefore, the participation model that the social web advocates is much closer to manipulation than to equal participation.

In order to demonstrate the manipulative power of the client-server architecture, the project *www_hack* has been developed, which encompasses a script that can be installed on a website by simply including the following code on the HTML page header:

The script alters the normal behavior of the page by exposing the mouse movements of all visitors. For example, once the webpage is simultaneously visited by five different users, it is possible to see five mouse cursors moving in 'real-time'. When there is only one visitor the page operates as expected and there is no immediate evidence of the script tracking mouse movements. The script creates an immediate and sometimes sinister evidence of the presence of the other since the server does not provide any possibility of contact other than moving a computer mouse. Users are therefore trapped in

a forced participation controlled by a server, at least as long as they visit the website and do not turn off the javascript engine.

In order to change the power relations existing on online communities it is fundamental to revise the ownership of their foundations. In his 'Antisocial Applications: Notes in support of antisocial networking', Geoff Cox writes:

"[T]erms like social networking hold the potential to transform server-client relations into peer-to-peer relations but only if held within the public realm, outside of private ownership and as part of the commons." (Cox 2008)

Networks based on peer-to-peer architectures are an alternative to privately owned infrastructures. In such architectures there is no differentiation between nodes, instead all nodes are considered peers and communication is achieved by exchanging information among peers leading to a network that is owned by all participants. However, it is important to notice that peer-to-peer and client-server architectures are not incompatibly and are often used simultaneously. The use of server-client software on top of peer-to-peer networks immediately compromises its distributed aspect as the software becomes a central point where control can be deployed.

The project *Laptalk*, demonstrates the client-server hegemony on distributed networks. *Laptalk* is a presentation system that can be used to deliver slide presentations directly to the spectators laptops instead of the more traditional setup where slides are presented using large projections. Before the presentation starts, the spectators are encouraged to join a mesh network initiated by the speaker. Once several spectators have joined the network the speaker's laptop is temporarily turned off to demonstrate that the network is not centralized on the speaker's laptop. The speaker invites the spectators to point their browsers to the IP address of its own laptop where a web server is running the *Laptalk* software. Such software is based on the same code used to implement *www_hack* and allows the speaker to control in real-time the slides that are deliver to the spectators. When the speaker moves the presentation to the next slide, the webpage that the spectators are visiting is automatically updated to the next slide. Although the computers are interconnected using a distributed network, the presentation is controlled only by the speaker.

Online communities that wish to control the rules of their exchange must take into account that neither the network nor the software they operate is based on server-client architectures. Such issues affect not only online communities but all people that exchange information online. For example, most email software used today is based on an server-client architecture. In fact, it is intriguing to notice that most of the software we use today is based on a server-client architecture.

References

- Bauwens, Michel (2008): The social web and its social contracts: Some notes on social antagonism in netarchical capitalism [online] Retrieved from: <http://www.re-public.gr/en/?p=261> [Accessed 6 June 2010].
- Cox, Geoff (2008): Antisocial Applications: Notes in support of antisocial networking [online] Bristol: Arnolfini. Retrieved from: <http://project.arnolfini.org.uk/projects/2008/antisocial/notes.php> [Accessed 6 June 2010].