# Details of Research – John Gilmore – P2P MMOG project – NRF application

# Research Rationale and Motivation

With the advent of broadband Internet, Massively Multiplayer Online Games (MMOGs) have gained significant popularity over the course of the past few years [1]. MMOGs are characterised by expansive worlds, where a large number of players interact online with each other and the virtual environment to achieve certain goals through collaboration and teamwork.  
  
MMOGs hold great commercial as well as academic value. From a commercial perspective, the growing number of active subscriptions [2], translates to a growing MMOG market. The DFC Intelligence figure in [2], shows the online games market forecast by DFC Intelligence, a company specialising in game market forecasts for various sectors. It forecasts an MMOG market value in 2010 of approximately $5 billion.   
  
From an academic perspective, MMOGs also hold great value. An MMOG is a complex networked application, with clients requiring reliable real-time feedback on actions taken. The design of an MMOG requires in-depth knowledge of server architectures and network design. The design of a server architecture determines how many players the game will be able to host and what the user experience will be in terms of quality of service. The server architecture must be able to handle thousands of requests, store large amounts of data, update the game state and send responses back to all clients in real time.  
  
MMOGs, therefore, hold great academic as well as commercial value. Recently, significant research has been done in order to develop Peer-to-Peer (P2P) MMOGs [2], [3], [4], [5], [6].   
These P2P MMOGs hold many advantages over current systems. Advantages include:

1: greater robustness, because of no centralised control,

2: lower cost of scalability, because many lower spec machines are cheaper than a single mainframe,

3: lower cost to the operator, since no server bandwidth or hardware is required

4: lower latencies, since P2P connections can be set up to achieve direct connections and remove the need to go through the server,

5: improved handling of peak transient loads, since players entering the game host themselves, whereas in Client/Server architectures, backup resources have to be provisioned. These backup resources are rarely used, but are required in order to ensure responsiveness under instantaneous heavy load changes.  
  
These P2P MMOGs, however, still have many issues that need to be resolved before they can become commercially viable. Issues include:

1: how to determine which updates are valid for which players in a distributed fashion (interest management),

2: how to send valid updates to players (event dissemination),

3: cheating mitigation,

4: how to ensure that players contribute resources to the system (incentive mechanisms) and

5: how and where to reliability store game state in a network where players are constantly leaving and joining (state persistency).  
  
[1] B. S. Woodcock. (2008) Total MMOG active subscriptions. mmogchart.com. [Online]. Available: http://www.mmogchart.com/Chart4.html  
[2] L. Fan, “Solving key design issues for massively multiplayer online games on peer-to-peer architectures,” Ph.D. dissertation, School of Mathematical and Computer Sciences – Heriot-Watt University, May 2009.  
[3] B. Knutsson, H. Lu, W. Xu, and B. Hopkins, “Peer-to-peer support for massively multiplayer games,” in INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1, 7-11 2004, p. 107.  
[4] T. Hampel, T. Bopp, and R. Hinn, “A peer-to-peer architecture for massive multiplayer online games,” in NetGames ’06: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games. New York, NY, USA: ACM, 2006, p. 48.  
[5] A. R. Bharambe, S. Rao, and S. Seshan, “Mercury: a scalable publish-subscribe system for internet games,” in NetGames ’02: Proceedings of the 1st workshop on Network and system support for games. New York, NY, USA: ACM, 2002, pp. 3–9.  
[6] L. Fan, H. Taylor, and P. Trinder, “Mediator: a design framework for P2P MMOGs,” in NetGames ’07: Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games. New York, NY, USA: ACM, 2007, pp. 43–48.

# Problem Identification

Of the five issues mentioned earlier: interest management, event dissemination, cheating mitigation, incentive mechanisms and state persistency [1], state persistency was identified as the field that still requires the most work. State persistency entails storing the game state reliability, while allowing state to be retrieved with low latency.  
  
Very little work has been done on state persistency for P2P MMOGs. Generally, four techniques were identified, of how state is currently stored in P2P MMOGs. These techniques are super peer storage [2], overlay storage [3], [4], [5], [6], hybrid storage [7], [8] and distance-based storage [9], [10], [11].  
  
None of the storage techniques currently used to implement state persistency completely satisfies al the criteria that were identified to be of importance to the P2P MMMOG space. The identified criteria are reliability, responsiveness, security, fairness and consistency. Some techniques, for example overlay storage, is very reliable but not responsive, while super peers storage is responsive but not very reliable.  
  
[1] L. Fan, P. Trinder, and H. Taylor, “Design issues for peer-to-peer massively multiplayer online games,” Int. J. Adv. Media Commun., vol. 4, no. 2, pp. 108–125, 2010.  
[2] B. Knutsson, H. Lu, W. Xu, and B. Hopkins, “Peer-to-peer support for massively multiplayer games,” in INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1, 7-11 2004, p. 107.  
[3] L. Fan, “Solving key design issues for massively multiplayer online games on peer-to-peer architectures,” Ph.D. dissertation, School of Mathematical and Computer Sciences – Heriot-Watt University, May 2009.  
[4] T. Hampel, T. Bopp, and R. Hinn, “A peer-to-peer architecture for massive multiplayer online games,” in NetGames ’06: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games. New York, NY, USA: ACM, 2006, p. 48.  
[5] S. Douglas, E. Tanin, A. Harwood, and S. Karunasekera, “Enabling massively multi-player online gaming applications on a P2P architecture,” in In  
Proceedings of the IEEE International Conference on Information and Automation. IEEE, 2005, pp. 7–12.  
[6] M. Merabti and A. El Rhalibi, “Peer-to-peer architecture and protocol for a massively multiplayer online game,” in Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE, 29 2004, pp. 519 – 528.  
[7] T. Iimura, H. Hazeyama, and Y. Kadobayashi, “Zoned federation of game servers: a peer-to-peer approach to scalable multi-player online games,” in NetGames ’04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games. New York, NY, USA: ACM, 2004, pp.  
116–120.  
[8] C. Gauthier Dickey, D. Zappala, and V. Lo, “A fully distributed architecture for massively multiplayer online games,” in NetGames ’04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games. New York, NY, USA: ACM, 2004, pp. 171–171.  
[9] S.-Y. Hu, S.-C. Chang, and J.-R. Jiang, “Voronoi state management for peer-to-peer massively multiplayer online games,” in Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, 10-12 2008, pp. 1134 –1138.  
[10] E. Buyukkaya and M. Abdallah, “Data management in Voronoi-based P2P gaming,” in Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, 10-12 2008, pp. 1050 –1053.  
[11] A. Bharambe, J. Pang, and S. Seshan, “Colyseus: A distributed architecture for online multiplayer games,” in Proc. of NSDI: 3rd Symposium on Networked Systems Design & Implementation. USENIX, 2006.

# Approach/Methods/Techniques

A. Model design  
The persistency model is based on groups of players, rather than regions. An architecture is proposed that uses a hybrid model of storage, similar to the Zoned Federation model proposed in [1]. A distinction will be made between ephemeral and permanent data. Ephemeral data are data that is only valid for a short time, while permanent data are data that remains constant for a significant amount of time. An example of ephemeral data is player movement, which changes frequently and, therefore requires long-term storage. An example if permanent data would be a player’s attributes or inventory contents.  
  
All players will form part of a specific group. The hierarchical state persistency model on one level operates within groups and on the higher level, amongst groups. The proposal is that ephemeral data will only exist within groups, with a high level of responsiveness. Unicast connections will be used amongst groups that would enable fast retrieval of ephemeral data such as player position updates. It is also proposed that a fully connected P2P network model is maintained within groups of players. This would allow for high responsiveness. Ephemeral player data might be stored on a second level overlay storage, which only exist amongst the members of every group. Group data can thus be distributed amongst players to achieve maximum fairness. It is also proposed that no central peer be used to apply game logic to peer events as is the case with super peers.  
  
It should be explored whether an event-based or update-based consistency model should be used within groups. If group sizes could be bound, the scalability issues may be minimised. This would allow for all peers to apply game logic and for validity checking to be done on updates sent from peers. This would increase the security of the system and would prevent any malicious super peer from hijacking the game. How a limited group size will affect the game will be investigated to determine whether it is feasible.  
  
It is then proposed that the overlay acts as a slow, but reliable storage medium, with data being constantly backed up to this medium. If data are lost before it could be backed up, the game should still be able to return to a previously stable state for the players that logged out, after logging back in.  
  
The main purpose of the proposed consistency model is to develop a low latency, highly reliable, consistent, fair and secure storage network. It is proposed that the use of super peers be avoided as much as possible, in order to obtain all benefits of a P2P model.  
  
B. Testing and evaluation  
1) Evaluation Criteria: In order to evaluate any consistency model, metrics have to be defined to measure the key consistency challenges. To evaluate consistency is to evaluate how well the game state remains consistent between peers. For the game state to be consistent, all game objects should be consistent. What will be measured is the differences between root and replica objects. Root objects are those objects that represent part of the global game state and replica objects are local objects used to display the game state at the client side. When an update is sent to a root object, the changes to the root object should be reflected in all replica objects. The appearance of a root object should also be the same for all peers. Any peer querying a root object should receive the same object. This can be tested by querying root objects and updating these objects as they are queried. What should be tested is how long it takes for an object’s appearance to update, after an update has been submitted by some  
node.  
  
To measure fairness, the distribution of game state should be measured. This can be measured at a file level, i.e. what is the variance of the number of files contained on each node, or on byte level, i.e. what is the variance of the number of bytes stored on each node. A lower variance will point to a fairer data persistency scheme.  
  
Reliability encompasses both robustness and availability. Robustness means that the data should be resilient to nodes leaving the network and availability means that data should be available to any node in the network, with the correct permissions. To measure robustness, nodes have to leave the network at different rates and the loss of storage for different rates of churn has to be measured. It can then be determined what the churn threshold is for the network to maintain all data at a certain rate of churn. This will be a function of the number of redundant objects in the storage system.  
  
Responsiveness can be measured by the time it takes for an object to be available for reading, anywhere in the network, after having been written. How long it takes to read or write data to the storage network can also be measured.  
  
Security is the combination of a number of objectives. These are: Authentication, Authorisation, Data Integrity, Confidentiality, Availability, Trust, Privacy and Identity Management [2]. Initially, a focus will be placed on data integrity as a failure here will result in the game data being directly compromised. Other aspects of security are also important, but these can be added on later, when the system is more mature. Data integrity can be tested by making sure the storage system can withstand the dropping, delay or, possibly, the modification of data packets by malicious nodes in the network. Nodes can be programmed to randomly drop packets and the availability of the data under these circumstances can be measured by the time it takes to rebuild a file under different rates of packet drops or delays.  
  
2) Testing methodology: These are the metrics by which the proposed consistency model will be evaluated. Initially an evaluation by simulation approach will be followed for the proposed consistency model. For this evaluation there are three possible options which should be investigated. The first option is to use a generic Discrete Event Simulator (DES), such as DESMO-J, the second option is to use a network simulator, such as OMNeT++ or ns-2, and the third option is to use a higher level simulator running on an existing network simulator, such as OverSim [3].  
  
If the consistency model is not too complex for simulation, a DES can be used to simulate the model from scratch. DESMOJ, for example, is a Java based simulator that provides the tester with basic queues, probability distributions and allows her to create objects. DESMO-J also tracks the sizes of all queues over time and is capable of producing graphs of queue lengths over time as well as a detailed report. Java objects are created in DESMO-J, each with a particular behaviour. This would entail sampling from a probability distribution to simulate processing time, adding itself or other objects onto queues or producing other objects. Objects can also easily communicate using the simulation environment.  
  
This environment allows for the creation of objects with specific behaviour. Objects can be made to model anything in the virtual world. It is easy to learn the environment and one can quickly start to produce results. It does however lack the depth and extensive library support of the network simulators. If no network specific protocols need to be simulated for the consistency model, DESMO-J might be the best choice.  
  
Network simulators possess large libraries that allow a tester to easily simulate any existing Internet protocol. A complete protocol stack can be simulated, using this simulator. It would then be possible to implement the consistency model in the application layer and simulate the functionality of the system from the ground up. This will allow for all network parameters  
to be taken into account for the simulation. The question is whether it is necessary to take into account all these parameters.  
  
A strength of the more well known open network simulators is that users have implemented their own simulators on top of these network simulators and made these simulators available to the public. OverSim, which is based on OMNeT, allows for the simulation of a P2P overlay network in the network simulator. Is already has a working simulation of an overlay network and allows users to add nodes with certain behaviours into this overlay simulation. This would allow for the simulation of the consistency model on an overlay and could greatly decrease the time required to set up the simulation. The viability of this solution will depend on the availability of features to support the simulation of the consistency models.  
  
Grouping algorithms first have to be investigated on which the consistency model and network architecture will be built. A consistency model will then be developed using the grouping algorithm. This model will then be tested using the metrics identified in Section 1. Other consistency models will also be simulated in order to compare their performances with the proposed model. Player movement data is required for this simulation. If possible, real game traces will be obtained and used to test the consistency model performance with actual player movements. If this is not possible, player movement would have to be simulated, which could become complex as the concept of player flocking should also then be simulated.  
  
MIH owns part of Astrium Online, which in turn operates the game called "Allods Online". It would be possible to obtain real-time anonymised player traces from Astrium in order to create a real world like simulation of player movements.  
  
After the implementation of the state persistency module, it should be integrated into a compete P2P MMOG architecture. Only this will make it possible to objectively test the performance of the storage systems with a working P2P MMOG game.  
  
Initially, this will be done by incorporating the state persistency model with an existing P2P MMOG architecture such as the Badumna network suite. Badumna implements interest management, event dissemination as well as object consistency. It, however, uses a centralised storage mechanism, where all root objects are stored on a single server. The state persistency architecture will be designed to intelligently relocate Badumna objects to implement a distributed state persistency scheme with low latency.  
  
After the state persistency scheme has been tested on Badumna, it will be incorporated into a completely novel P2P MMOG architecture, that will be based on the group based architecture.  
  
[1] T. Iimura, H. Hazeyama, and Y. Kadobayashi, “Zoned federation of game servers: a peer-to-peer approach to scalable multi-player online games,” in NetGames ’04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games. New York, NY, USA: ACM, 2004, pp.  
116–120.  
[2] A. Belapurkar, A. Chakrabarti, H. Ponnapalli, N. Varadarajan, S. Padmanabhuni, and S. Sundarrajan, Distributed Systems Security: Issues, Processes and Solutions. Wiley, 2009.  
[3] I. Baumgart, B. Heep, and S. Krause, “OverSim: A Flexible Overlay Network Simulation Framework,” in Proceedings of 10th IEEE Global Internet Symposium (GI ’07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA, May 2007, pp. 79–84. [Online]. Available:  
http://doc.tm.uka.de/2007/OverSim 2007.pdf

# Research Feasibility, Impact and Potential Outputs

The research is completely feasible. To determine feasibility, two factors have to be taken into account. Firstly: Is the necessary infrastructure in place in order to pursue the research and secondly: is there a sufficient academic base to support the research.  
  
The first question was answered earlier, but will be repeated here. MIH Medialab, along with Naspers and MIH provide broadband Internet access, access to the Amazon cloud, powerful lab computers and all the resources of the companies owned by MIH. It is also from here that player traces might be obtained to support the testing of the state persistency model.  
  
The second question of whether there exists sufficient academic literature to base the research on is evident from the references provided throughout the detailing of the work. There exists a significant body of research into P2P MMOGs. The research includes partial implementation of architectures, handling of the core issues pertaining to P2P MMOGs as well as surveys discussing what the key issues going forward are. Based on the existing body of research, the proposed doctorate topic seems completely feasible.  
  
The impacts and potential outputs of the study are the state persistency model and the P2P MMOG architecture. But more than that, a significant part of the study will also be the objective comparison of the different architectures and models used in P2P MMOGs and testing the different architectures with real world data. None of these issues have been handled in past literature and such a novel architecture and objective comparison will be of great benefit to the field.

After the successful implementation of a complete P2P MMOG architecture, the research will have a large impact, both academically as well as commercially. MMOGs will be more robust, cheaper to host and maintain, be more responsive and be able to handle any amount of load and support any number of players.

The lower latencies and the removed need for a centralised server will also open up avenues for MMOG genres that have not been explored before. First Person Shooter games, which require very low latencies (10 ms) can now be implemented in the form of an MMOG. This could not have been done before, because of high server latencies. If it is possible to implement a First Person Shooters, racing game or sports games as an MMOG, a new market will open up in terms of MMOGs and gaming.

# Research Aims and Objectives

From the identifies challenges in the field of P2P MMOGs, the study has three expected outcomes:  
1: The main outcome will be a state persistency model that is provably better than currently used models for P2P MMOGs.  
2: The second part of the contribution is developing a P2P MMOG architecture, which uses the new state consistency model to achieve better performance than current architectures.  
3: The third contribution will be to compare different architectures and consistency models and determine which of the other proposed architectures in the literature work better with a P2P MMOG.

# Research Activities/Plan

1: Initial literature study with a milestone of a completed PhD proposal. The duration of this task was 72 days with 36 days set out for writing the proposal itself. Both tasks have already been completed.  
2: Write a survey article to be published in a scientific journal, based on the PhD proposal in 15 days. The article has been completed and should be submitted within the next few days.  
3: Deploy and test Badumna network suite locally in 20 days.  
4: Deploy and test Badumna network suite in the Cloud in 40 days.  
5: Develop a test framework to test the functionality of network suites in 40 days.  
6: Develop a grouping algorithm to be used with the state persistency model in 60 days.  
7: Test the grouping algorithm in 20 days.  
8: Develop the state persistency model using the grouping model in 52 days.  
9: Simulate and test the state persistency model in 40 days.  
10: Integrate the state persistency model with the Badumna network suite in 20 days.  
11: Test the state persistency in Badumna in 40 days.  
12: Write a paper on the developed state persistency technique in 25 days.  
13: Design a novel P2P MMOG architecture using the grouping algorithm as base and the state persistency model and adapt other components currently used in P2P MMOGs to work with the grouping algorithm to achieve a complete P2P MMOG architecture based on grouping in 20 days.  
14: Implement the designed architecture in 90 days.  
15: Test and compare other available architectures with the implemented architecture in 40 days.  
16: Write up the PhD in 90 days.  
  
Currently the plan is to distribute the implementation of components other than state persistency for the architecture as masters or final year projects. These components will then form part of the larger architecture, but will still be based on the grouping algorithm and implemented as projects of other students. The focus of the doctorate will be on state persistency and how to incorporate the state persistency into an fully designed P2P MMOG architecture.

Some modules that might be assigned as Masters or final year projects are: a group based interest management module, a group based cheating mitigation scheme, an incentive mechanism and the implementation of a game to visually test the architecture with.  
  
The required resources are broadband internet access, which is available in the MIH Medialab. Cloud availability is also required, which can be obtained directly from Amazon or through some partners in the MIH group at a reduced price and with higher availability. Simulation tools such as Matlab might also be required and is available in the Lab.  
  
For a Gantt chart of all research activities please refer to the appendix of the attached PhD proposal. The chart shows all timeframes as described and one can also see the expected start and completion dates of all activities.