Your Wonderful Project Title

Topic outline

Author #1 Name (auth#1 NetID) - *auth#1 major*
Author #2 Name (auth#2 NetID) - *auth#2 major*

Supervision

- Supervisor #1 (sup#1-email-address)
- Supervisor #2 (sup#2-email-address)

Context

A very brief overview of the issues, and a short selection of related work. It's acceptable if the latter doesn't fit the topic perfectly; they are references that help circumscribe the perimeter of the problem you want to tackle.

For example, let's say you want to work on a secure online chat service. Your service aims to allow users to exchange messages without worrying about their privacy. You'll have to consider issues such as trust and malicious behaviours, both from users and from inconsiderate outsiders; [1] provides a service that contends with similar challenges. One of the hardest attacks to fend off, and that you'll probably have to consider is known as the *Sybil attack* [2], where users take on multiple identities within the service to raise their influence on the network. Opting for an architecture that has no central server offers two advantages: it removes the concern of a single point of attack, and it mitigates the power of nuisance of the service provider. But a decentralized architecture raises another tough challenge for messaging. In [3], Guerraoui discusses the complexity of delivering a sequence of messages in the same order to remote recipients.

Objectives

Start with a rough summary (one or two paragraphs) of the final result, in other words what you expect to achieve at the end of the semester. Let's go back to the secure online chat example. The summary should describe what features will be available to users; which of the issues mentioned in the context you'll focus your work on; if any, which off-the-shelf services and software you'll base your implementation on.

Finish with your work plan, with a set of steps you've laid out in order to reach your final result. For instance, in the case of the chat service example:

1. Design the user interface of the service, and use it to discuss the offered features and how they'll interact.
2. Come up with an architecture for an entirely serverless chat service.
3. Implement the service as a single local client application.
4. Deploy the service and test its operationality (loop back to the previous step if necessary).
5. Carry out a benchmark of attacks on the implementation to assess the security of the service.
6. Use the same benchmark on another service for comparison.

References

[1] V. Pathak and L. Iftode, "Byzantine fault tolerant public key authentication in peer-to-peer systems," *Computer Networks*, vol. 50, no. 4, pp. 579–596, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2005.07.007

[2] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK: Springer, Berlin, 7-8 March 2002, pp. 251–260.

[3] R. Guerraoui, "Genuine atomic multicast in asynchronous distributed systems," *Theoretical Computer Science*, vol. 254, pp. 297–316, 2001.