BAM! Pong

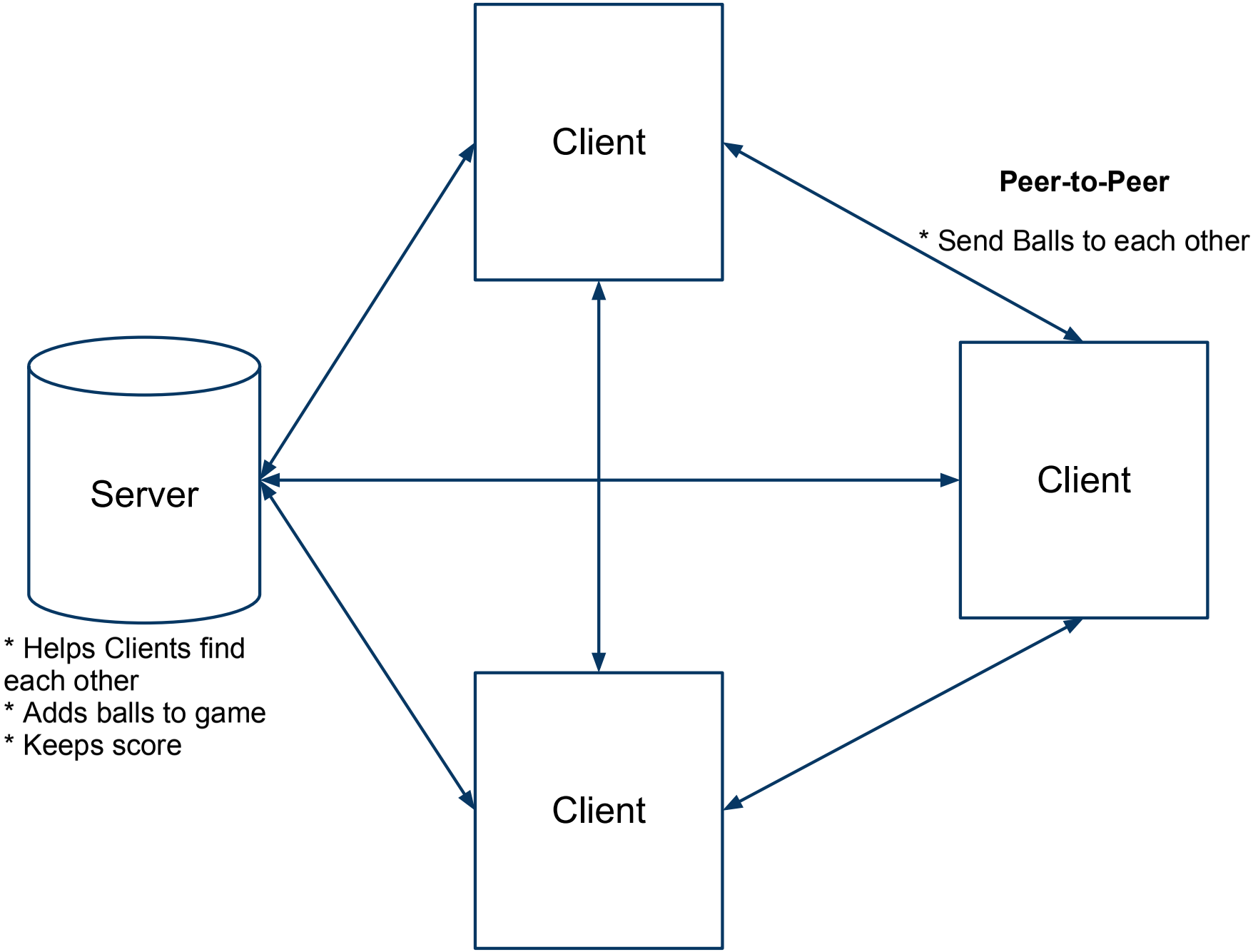
Team

Brian Gernhardt
Aniket Sharma
Max Bogue

Project Motivations

- Investigate Peer-to-Peer and Client-Server methodologies.
- Provide message passing across different platforms: PC, Blackberry, and Android.
- Create a system that was both educational and entertaining.

Architecture



Design

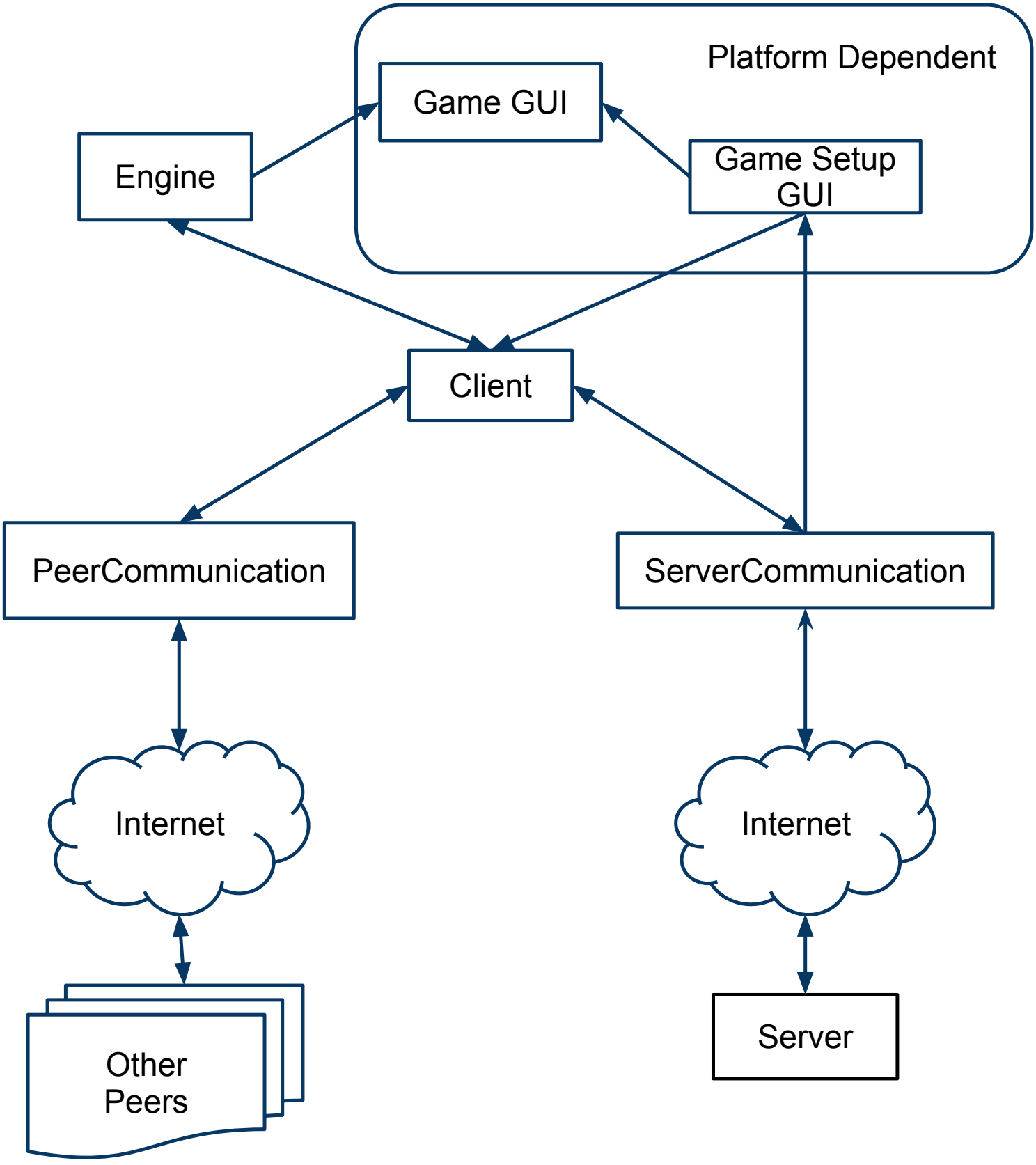
- **Game Start:** Clients connect to server, using simple messages to join games. Server sends message to all clients when requested to start game and adds initial balls.

- **Ball Passing:** Clients pass the ball via peer-to-peer messages, using Paxos to ensure that all clients agree and are informed of the current state.

- **Client Failure:** When a client loses connectivity to a peer, it initiates a round of voting to decide if the peer should be dropped. If a majority of clients can not see the peer, it is dropped.

- **Server Failure:** Similar to a client drop, a vote is initiated and if clients agree the server has failed, a backup server is contacted by all clients.

Implementation



Lessons Learned

- **Expect New Things to Take Longer:** Our team did not have much experience with any major section of the project: Android, networking and game physics. It took much longer than we had planned to get basic implementations

- **Integrate Early and Often:** We each worked on a major section of code, learning the techniques required and developing in pieces. We left integrating the sections until late in the project.

- **Even Simple Distributed Algorithms are Complicated:** “Fast Paxos” (Lamport, 2005) describes the Paxos algorithm in 4 steps over three pages. “Paxos for System Builders” (Kirsh and Amir, 2008) gives complete psuedocode for the same algorithm and takes 10 pages.

Current Status

- **Blackberry Platform Abandoned:** While Blackberry does have a Java SDK via Eclipse, it requires a custom install of Eclipse and does not implement standard Java packages like java.net.*

- **Desktop Expanded:** Originally, the desktop version was a simple test platform but was expanded to become a full game client.

- **Networking:** Paxos turned out to be a more complicated algorithm than originally expected. When the implementation had problems, we went back to a more simple (but less failure resistant) message passing method.

- **Android:** The Android platform is mostly functional, but due to the disconnected method of development and indirect nature of its APIs, it has proven difficult to debug.

Future Work

- **Server Failure:** Backup servers were never implemented.

- **Consensus Algorithms:** Paxos seemed to be far more complicated than actually required for this project. Other forms of voting and data propagation could be investigated.

- **Additional Platforms:** While Blackberry was never developed due to lack of time, the modular nature of the code should allow the differences to be abstracted away.