Ch 17 Application Issues

Prof. Magda El Zarki
Dept. of CS
Univ. of CA, Irvine
Email: elzarki@uci.edu
http://www.ics.uci.edu/~magda

Overview

- Security and Cheating
- Geometry and Animation
- Tiered Systems
- Thin Clients

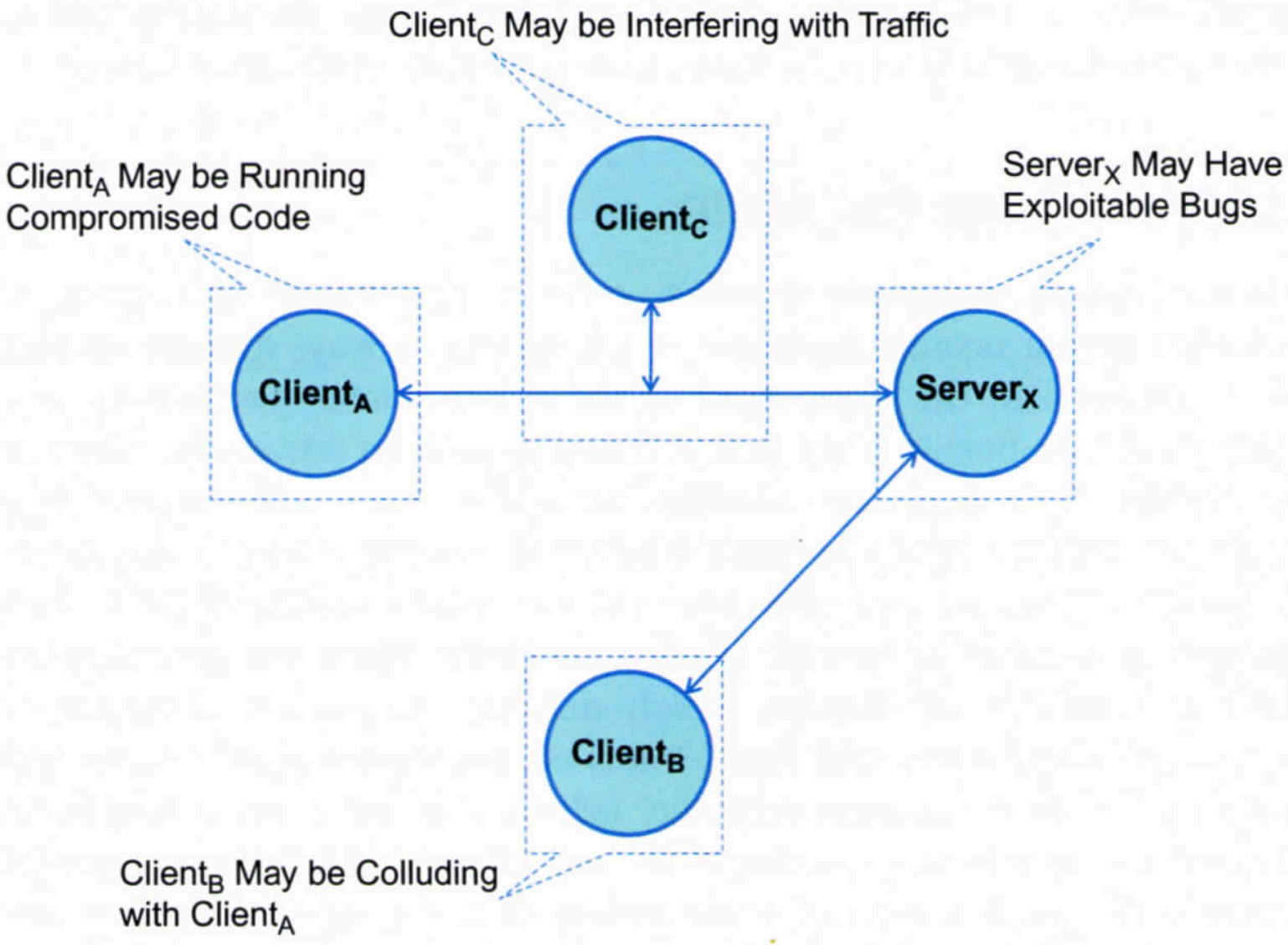
Security and Cheating

- Security is support within a system to protect against faults and attacks
  - Prevents unauthorized use of the system
  - Prevents users from disrupting service
- Cheating is the exploitation of some aspect of the system to gain an advantage in an environment
  - Cheating might exploit security problems – getting unauthorized access to a system
  - Cheating might be legal from a system's perspective but illegal from a social angle – virtual money?!?!?

Cheating

- Client side: attacker compromises client software
- Server side: attacker subverts server software
- Network level: attacker accesses the network traffic
- Social: players collude or use multiple clients

Types of Cheating



Client Side Attacks

- Player changes client side software to effect some advantage for the player.
  - Most common attack is to change the software to make walls/ objects invisible so that the client can see the enemy. So even though the server filters the data so that it doesn't send information about objects the client isnt supposed to see, with invisible walls, the client can still see them.
- Another attack is "lookahead" cheat, where a player's response is purposefully delayed, thus lying about its latency and so it receives updates from other players in plenty of time to respond (server tries to compensate for the client's latency).
  - There are some mechanisms that can detect cheating in dead-reckoning protocols.
- Another type of hack involves the graphics cards on a players system. The hacks change the way the image is rendered. The graphics command stream can be intercepted and changed.

Client Side Cheating – contd.

- Another attack it so augment a player's skill. It does this by changing the control input that the player is sending. E.g., correct the player's aim or fire when it is the right time. We call this an "aimbot."
- The are other bots that actually will play for you and complete actions and tasks, e.g., after you have left the persistent world. They are called bots. (see next slide)
- Detecting this type of attacks is difficult.

An example of a bot – Glider – Taken form Wikipedia

- Blizzard Entertainment created and operates a popular online world game known as World of Warcraft (WoW). WoW is a massively multiplayer online role-playing game, in which players control characters and complete a variety of tasks, such as exploring the landscape and performing quests. As players continue to play and succeed in their tasks, their characters gain various talents and skills.

- Michael Donnelly, the founder of MDY Industries, LLC, created a software bot called Glider to play WoW for its users. Thus, Glider users were able to advance their WoW characters unattended.

- In its ruling on Blizzard's contributory copyright infringement claims, the district court first considered whether purchasers of WoW were legal "owners" of the client software. According to 17 U.S.C. § 117, owners of computer programs are allowed to create copies or adaptations of the computer program if it is an essential step towards utilization of the program.[2]

- The Court agreed with Blizzard's arguments that WoW purchasers were not legal owners of the game software but instead licensees. As licensees, players are required to make use of the software within the scope of the End User License Agreement. In the terms of that agreement, Blizzard specifically prohibited "the use of bots or third-party software to modify the WoW experience."[1] Thus, the Court found that players who use the Glider program violated the TOU and were not licensed to use WoW.

From a Player's perspective (Consalvo 2009):

- Developed by Michael Donnelly and MDY Industries, the glider is a small program or mod for WoW that lets the user program one of her avatars to travel along a preset path, killing whatever is found, skinning, looting, and gaining experience points from the looped activity. The makers of the program/ mod stress on their Web site that the mod is designed to eliminate the tedious aspects associated with leveling a character (the grind), especially for players who may have already done so with several other characters—in other words, this is for alts or very experienced players to "fast-forward" through undesirable parts of the game.

- Fast-forwarding is a common reason people will cheat in a game (Consalvo, 2007), although in multiplayer games such as WoW, the developers usually consider such activities as in violation of the game's terms of service (ToS). It is thus illegal,and the creators of the glider are currently being sued by Blizzard for their creation (Markee, 2007).

Contd.

- We can consider the potential activities associated with the WoW glider in several ways. Some individuals might use the glider to level avatars that they intend to sell to other players for (real) currency and thus profit off the fast-forwarding mod. Some players may wish to level their second, third, or fourth avatar through either some or all of the grind in the game, to achieve higher levels, but keep those avatars as part of their account to play with in the future. Some players (admittedly few) may find the WoW glider before even beginning the game and use it to level their avatar to get to the content they assume is most valuable—such as end game raiding. Although the developers have deemed all those activities as cheating and violations of the ToS, they obviously have different meanings for the players involved. Likewise, they have different meanings and outcomes for players who do not use the WoW glider but who are nonetheless affected by its presence as a mod.

- Nonglider using players may consider the opportunity to purchase such a leveled avatar as a bargain, rather than leveling an avatar on their own. Nonglider-using players may feel that avatars running automatically on preset paths in certain areas are unfairly hogging resources in game, which they may need and feel more legitimate claim to, being in actual control of their avatar. Finally, nonglider using players may feel fiscal ramifications of glider-using players, if glider users also gather large amounts of consumable resources to sell via auction houses and either flood markets (driving prices down) or control the sale of certain items (driving prices up).

Controlling Client Side Cheating

- Two approaches:
  - Client Verification
    - Examine client memory and disk to verify the code and data. Not a very popular solution as it infringes on privacy.
  - Behaviour Tracking
    - Looks at characteristic patterns in user and compare it to new patterns such as generated by bots. Bots tend to do repetitive, monotonously times actions.
    - Client server systems can track this behavior. Some of it in real time – such as zero response time that a bot would have. Others might need offline analysis. Some work has shown that 200 secs of game play can be sufficient to detect foul play.
    - Some bots are intelligent and try to simulate human behavior but turns and movement rate anomalies and frequency and patterns of actions can be detected if long enough traces are collected.

Server Side Attacks

- Exploitation of bugs
  - A bug that lets people enter a privileged location or obtain an object (e.g., money or armor)
  - A bug that allows duplication of an object, e.g., duplicating wealth (e.g., Sony's experience with Everquest II)
- Brute force attacks - DoS
  - SYN flooding – opens many TCP connections and leaves them hanging.
  - ICMP flooding using a modified ping
  - Sending messages that cause buffer overflow at receiver

Network Level

- Eavesdropping on packets
  - Change the content of the packets – the angle of aim or the timing of an action, or even the target of the aim, etc.
  - Can be tackled by using secure network messages:
    - IPSec – VPNs and gateways
    - Secure sockets (SSL) – both sides have to agree
  - Application or middleware level security
    - Disadvantage consumes CPU cycles

Social

- Collusion – where players agree to a strategy that allows one player to gain advantage and then split the spoils
- Player behavior and ethics – the whole bot usage and the Glider example
- What is ethical and what isnt???

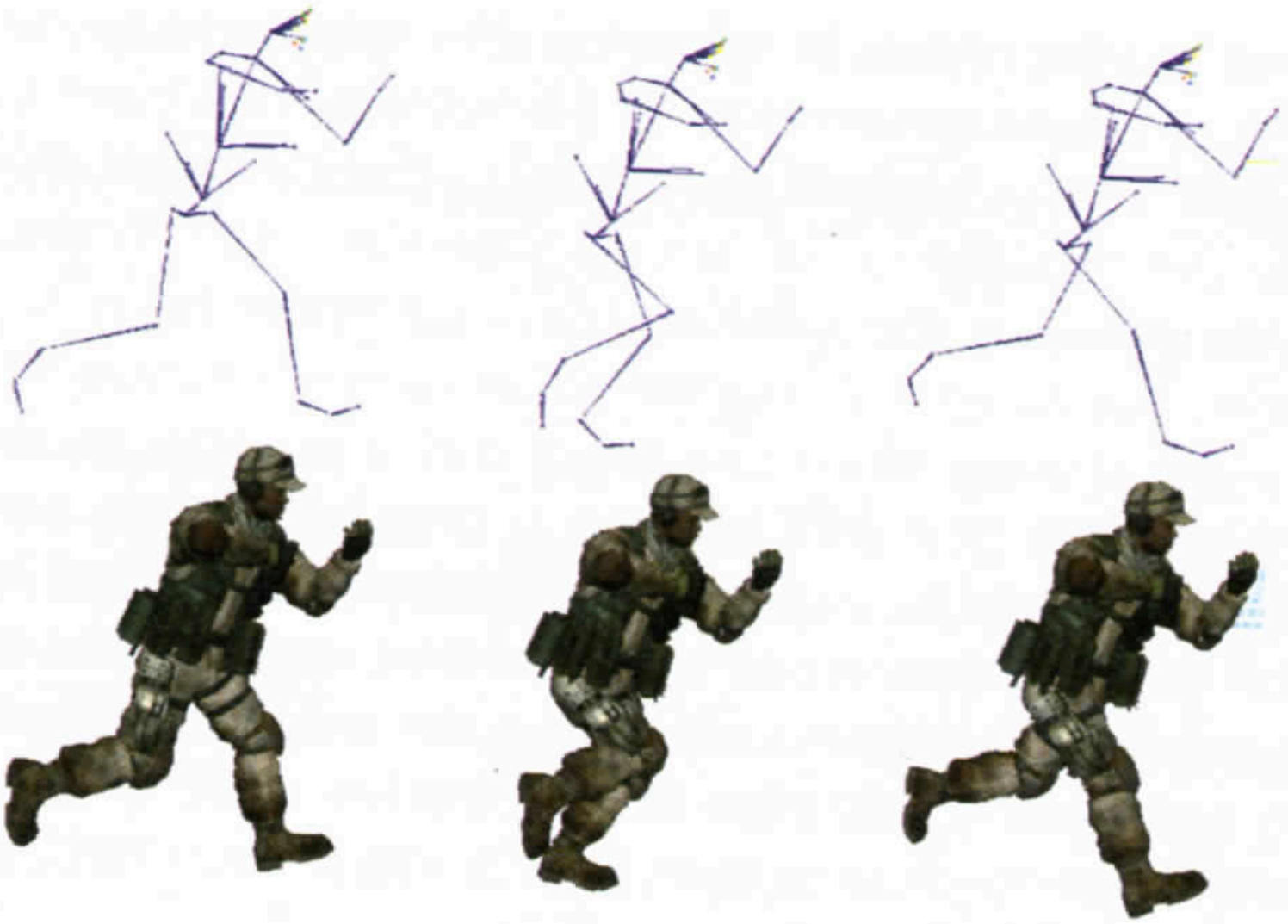
Repercussions

- Identify cheaters – beware of false positives
- How to police communities
- Banning cheaters – ineffective in environments where you can create a new identity
- On consoles – identities are tied to the console and so can be tracked. Also harder to hack software on a console – altogether a much safer environment.

Geometry and Animation

- Complex animations: character animation
  - Use keyframe animation – only the keyframe identifier and the time needs to be sent
    - An action is defined by a number of keyframes describing the action over a period of time.
    - Interpolation is used to give smooth action. If frames exist for every .1s the,n e.g., angles are interpolated for the in between motion
    - If bandwidth is low, keyframes can be skipped, picking which ones is an art to maintain smooth action/motion
    - Keyframes can also be cached on a client.

Character Animation



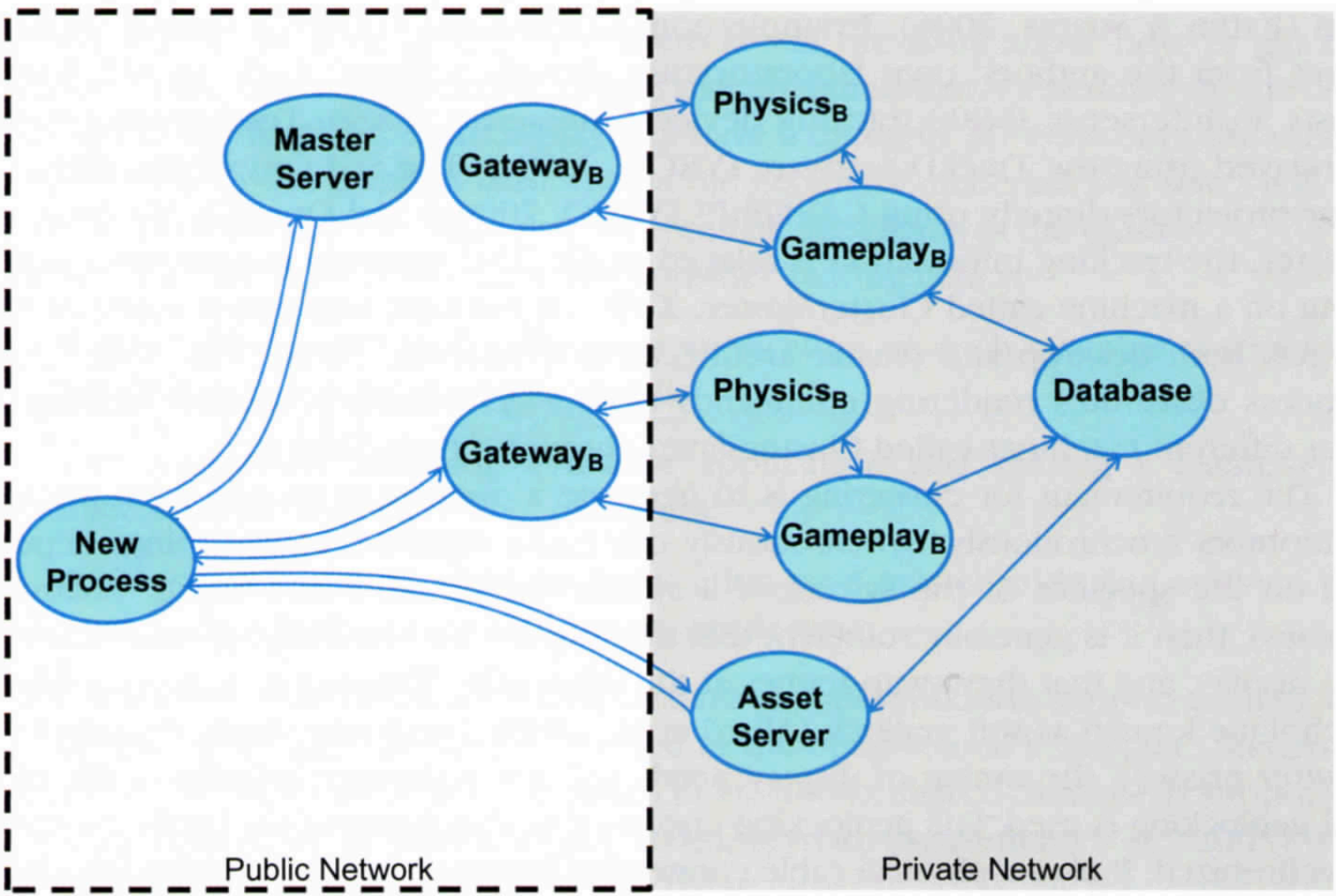
Large complex systems

- The models are stored on the server
- Downloaded incrementally depending on player interest
  - Geometry selection phase
  - Geometry transmission phase
- Environments stored in such a way that it is easy to find what next needs to be sent.
  - Client can decide – pull system
  - Server decides – push system
- Compression of meshes to reduce BW
- Progressive meshes (similar to video compression and layers)
- Using images of an object instead of a mesh - impostors

New Arhcitectures

- Instead of augmenting number of servers
- Instead of clustering clients
- Instead of using peer to peer
- Use a different tiered system that is split along functions

A possible architecture



Thin Clients

- Not a new idea
- Has existed for awhile
- Proposed for online games because of the increased complexity of the virtual environments
- Mostly to keep costs down for clients – graphics cards and high end CPUs not cheap
- Drawback – bandwidth requirements – streaming of audio and video – high resolution > 5Mbps

What is next???

- Now its your turn!
- Presentation Guidelines
  - Maximum 10-15 slides
  - State goal
  - Explain problem(s)
  - Describe/illustrate solution(s)
  - Examples great idea
  - Pictures/video worth a thousand words
  - Conclude
  - Give references
  - Leave time for 2-3 questions