Lecture 12

Advanced Topics:

Cheat Avoidance/Detection
Cloud Gaming

Cheat Detection / Avoidance

aimbot
maphack
wallhack
:

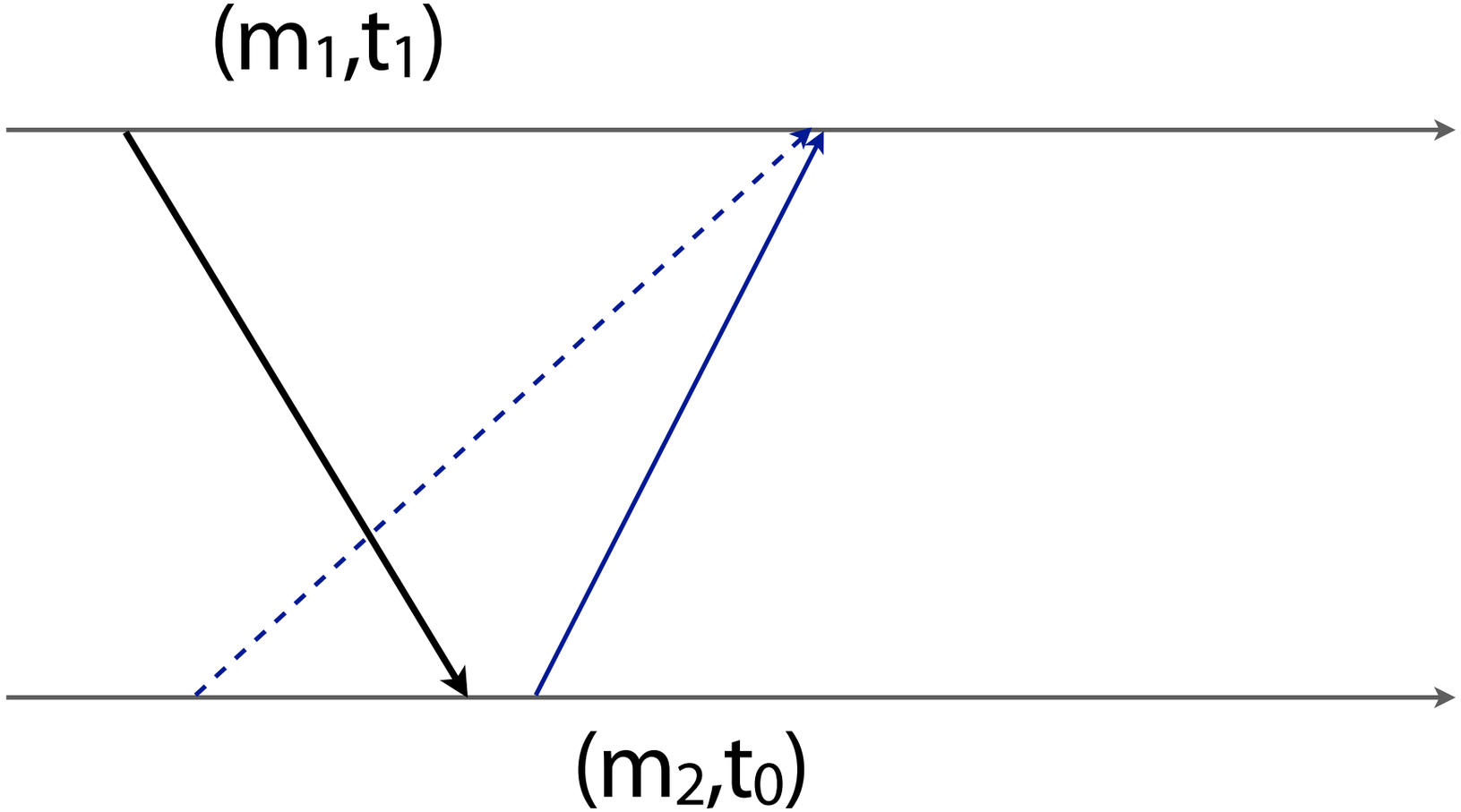
protocol-level cheats:

1. time-stamp cheat

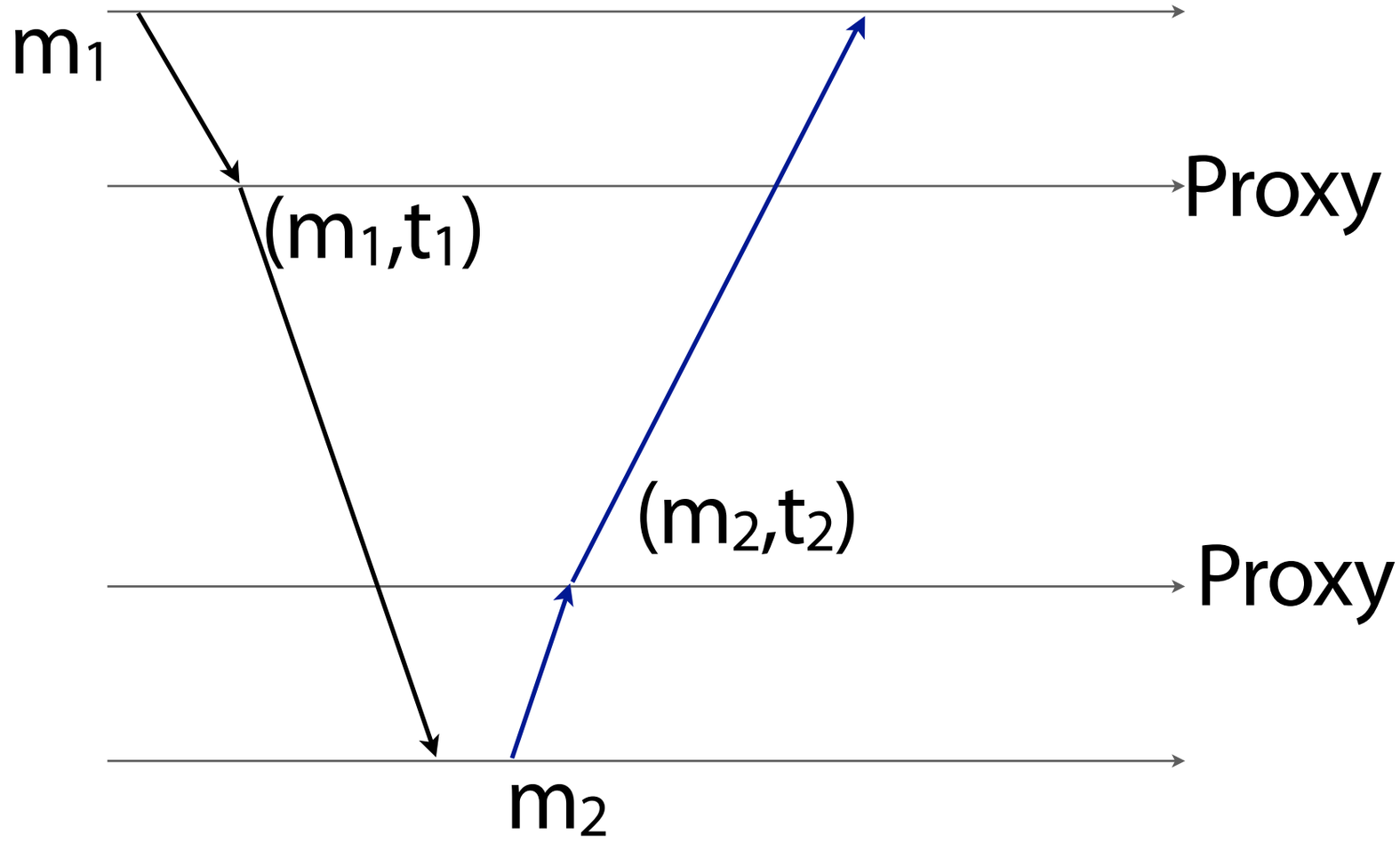
2. look-ahead chat

Timestamp Cheat

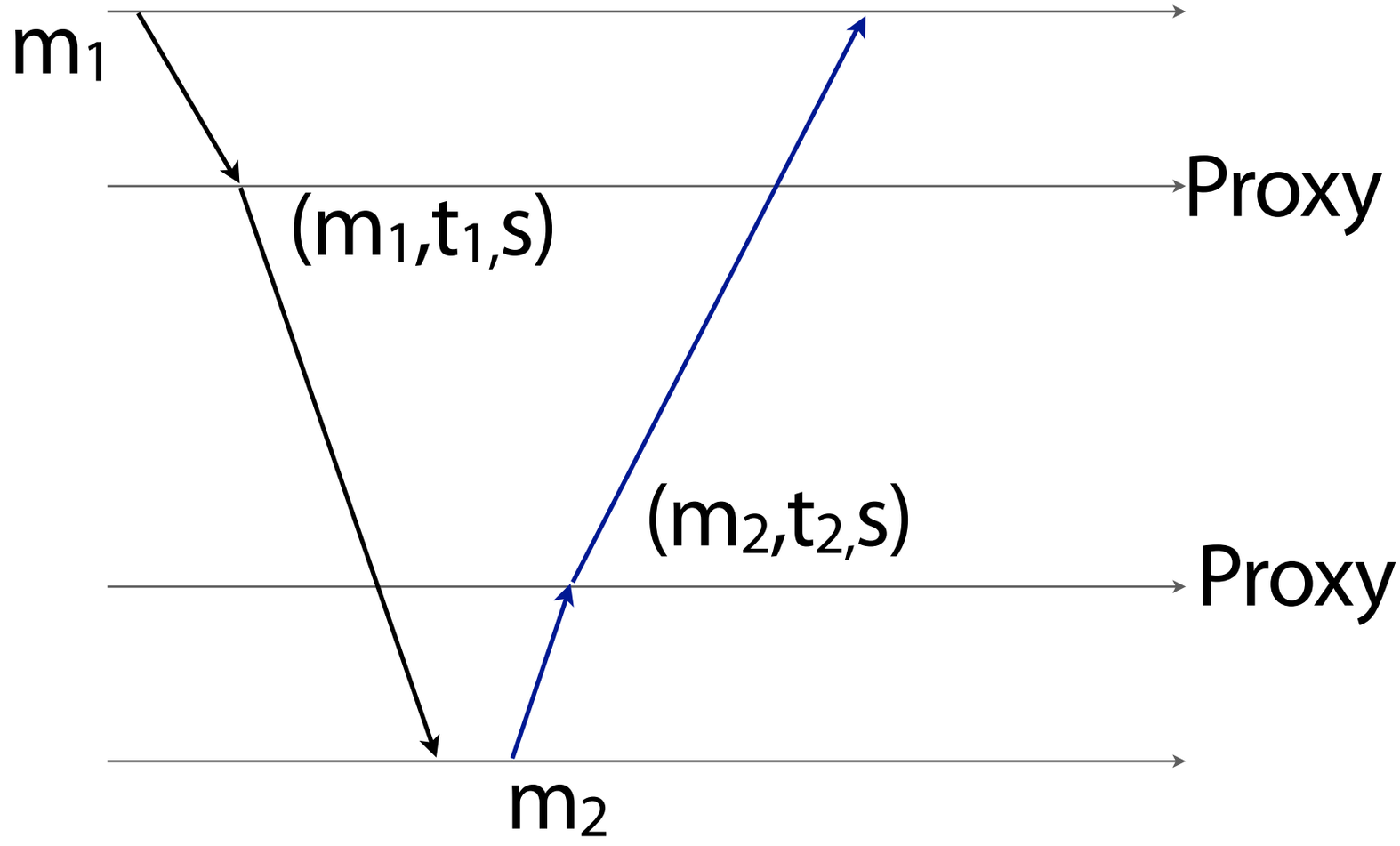
Player generates fake, earlier timestamp to gain advantage after knowing the opponent's move.



But such cheat is impossible if a trusted proxy timestamp the messages instead of the players.

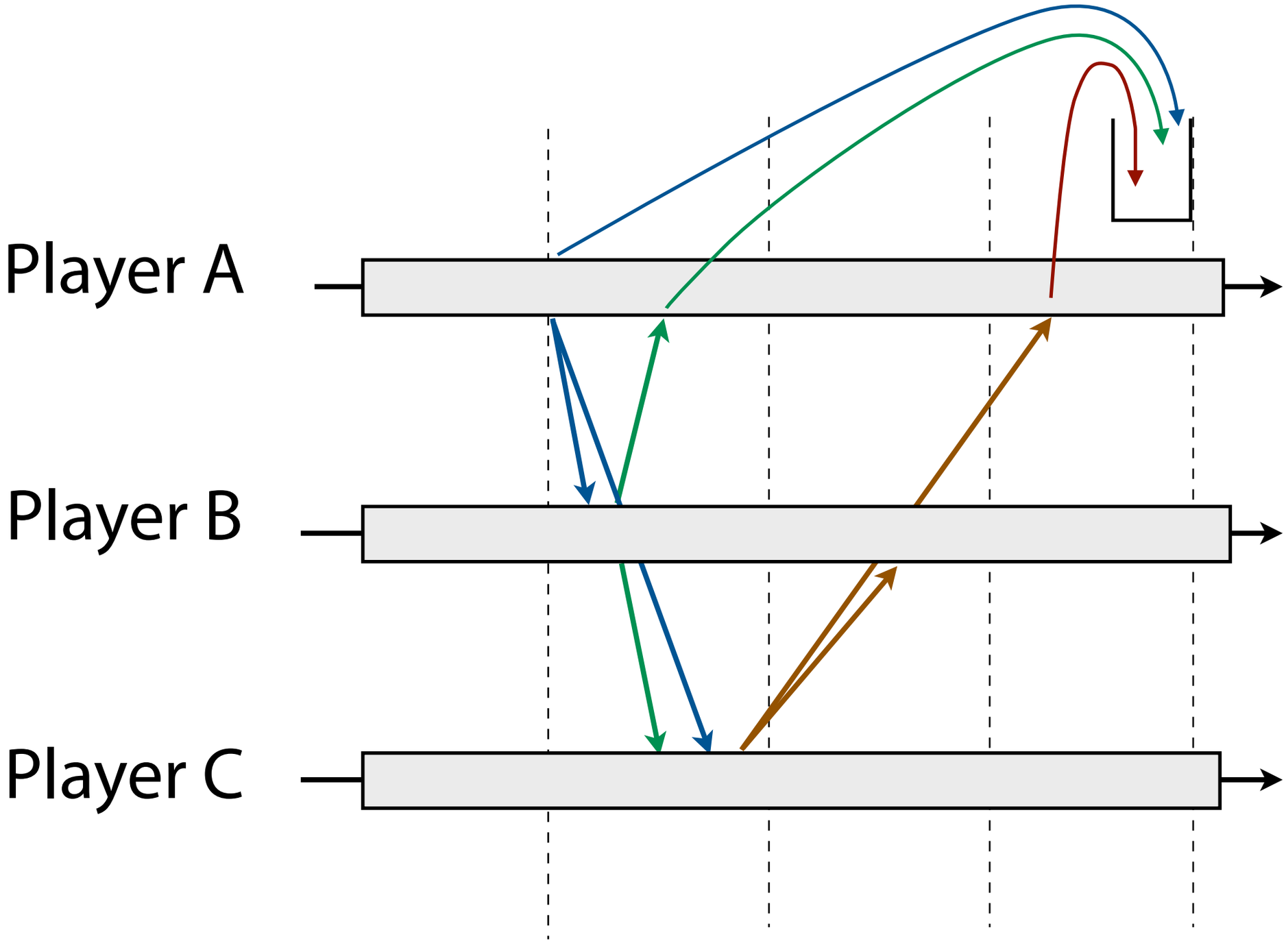


The proxy can digitally signed the message to proof that he is the one that added the timestamp.



Look Ahead Cheat

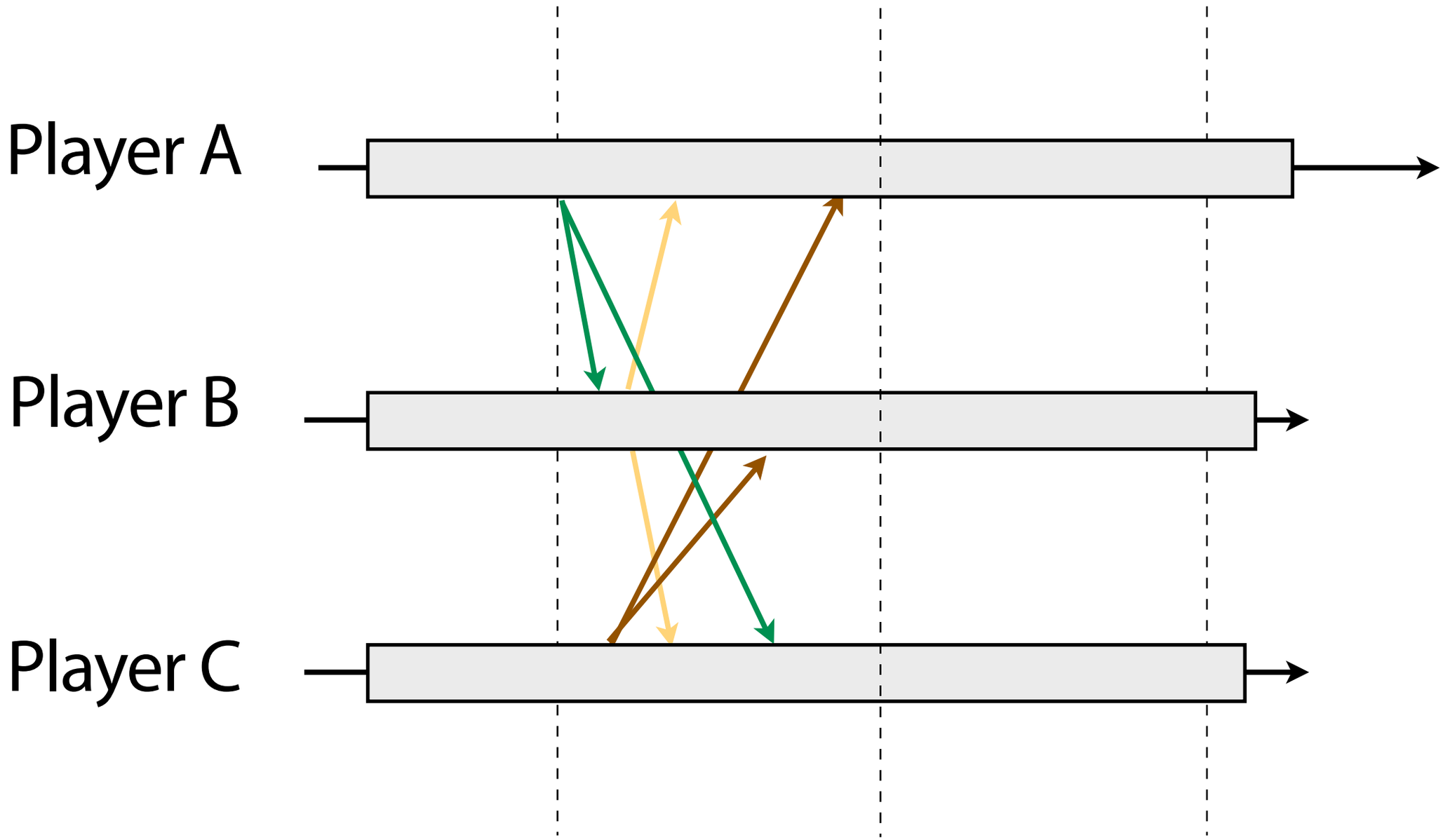
A player waits until the opponent have made its move before making its own.



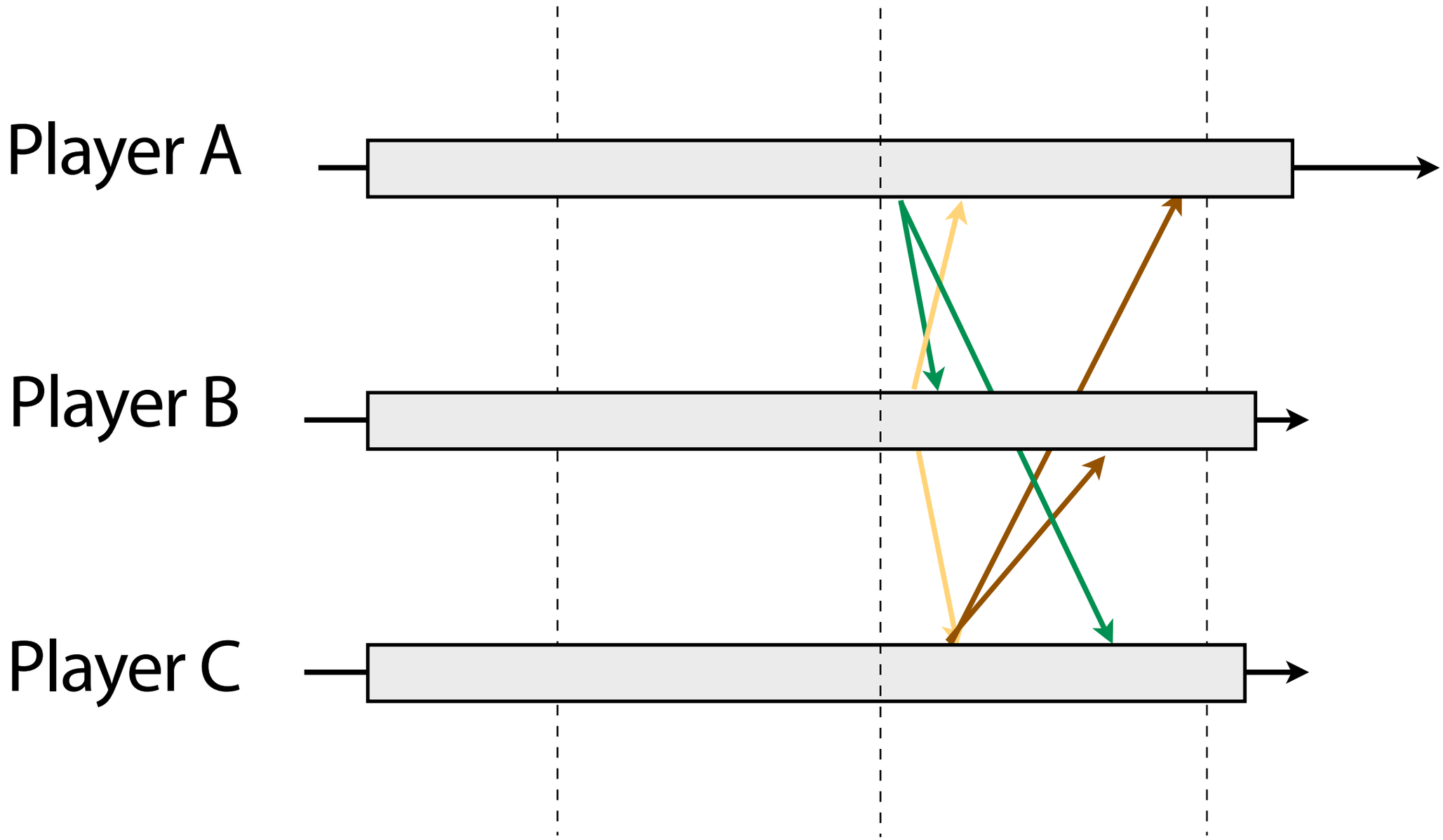
Lockstep Protocol

One-Way Function f: Given x, we can compute f(x) easily. Given f(x) it's hard to find out x if x is random.

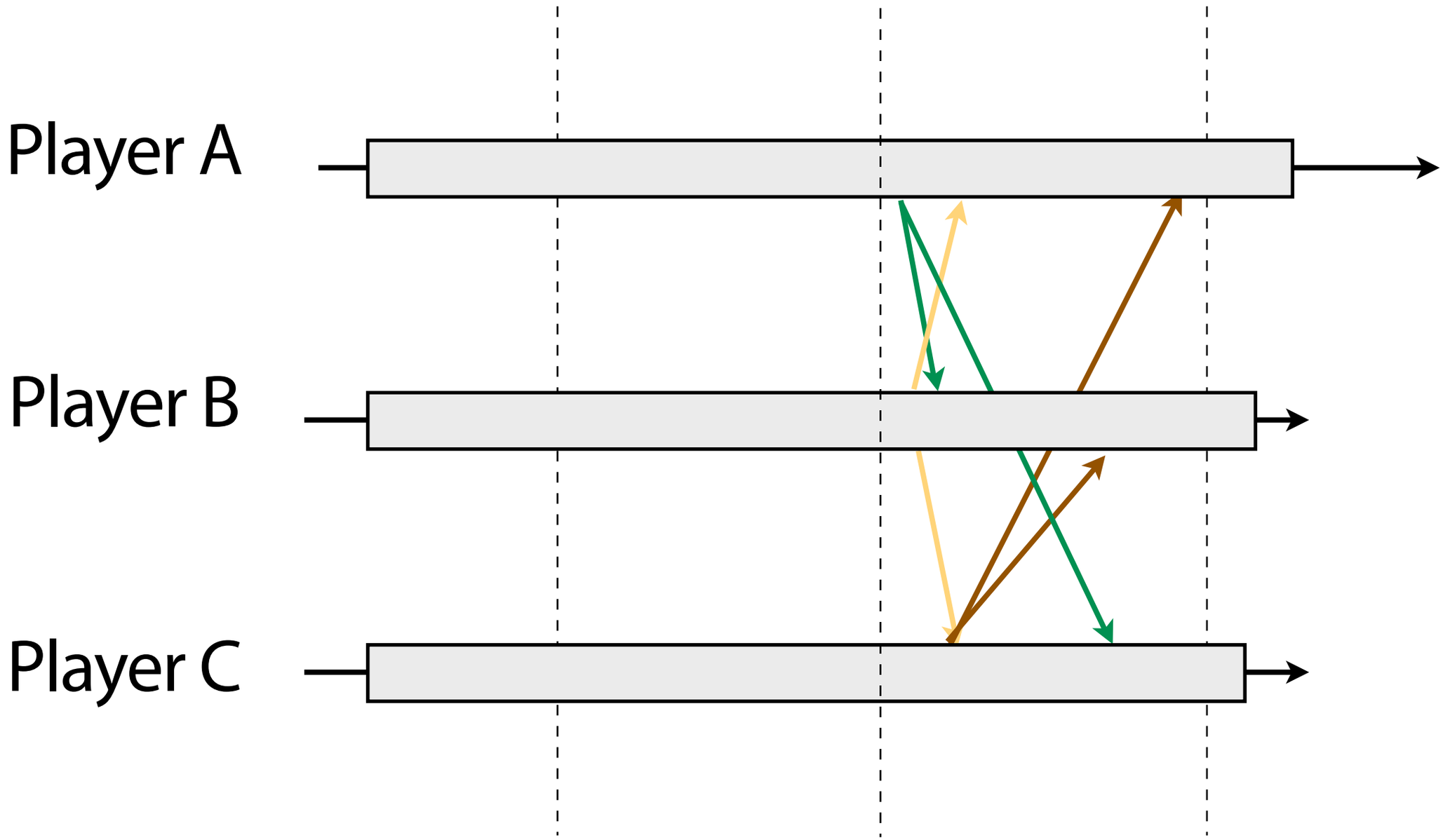
Stage **1** (Commit): Everyone decides on its move x, and sends f(x) to each other.



Stage **2** (Reveal): After f(x)s from all other players are received, sends x to each other.



Stage **3** (Verify): Players check if f(x) is consistent.



f(x) is known as commitment to x.

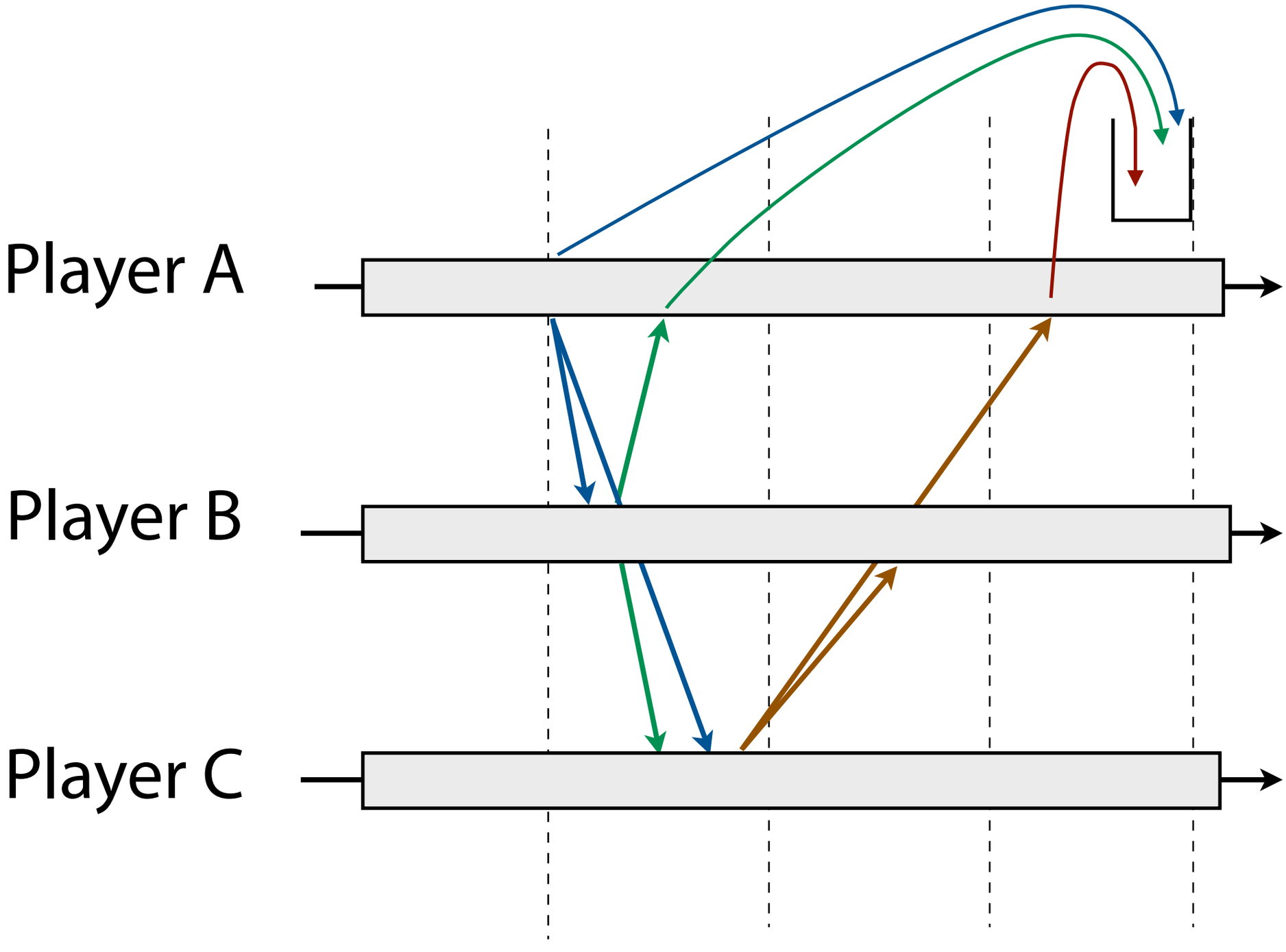
A player, once committed to its move, can’t change it.

Problem:

Lock-step protocol is slow.

Cheat detection is easier than prevention

Mmm.. player C always sends its update later than the rest.



Player Behavior Analysis

can also be used to detect aimbot

Game Log Analysis

Detect Aimbot/Triggerbot in FPS

http://www.theinquirer.net/inquirer/news/1023279/more-on-intel-anti-cheat-technology-revealed

Idea from Intel Research

Chipset in input devices collect traces
Game server does the same

Compare

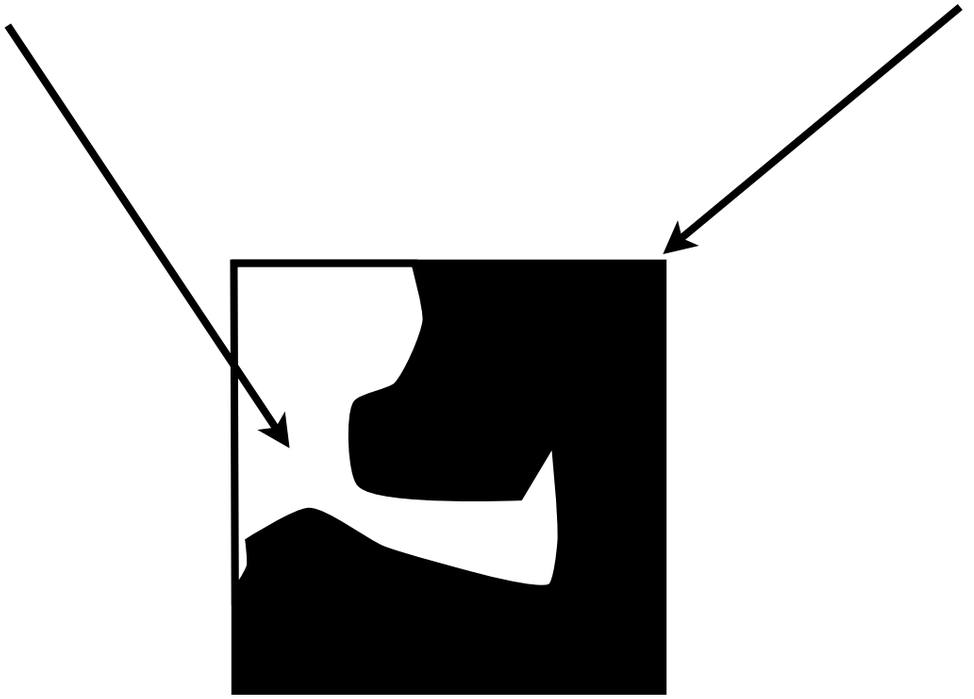
http://www.theinquirer.net/inquirer/news/1023279/more-on-intel-anti-cheat-technology-revealed

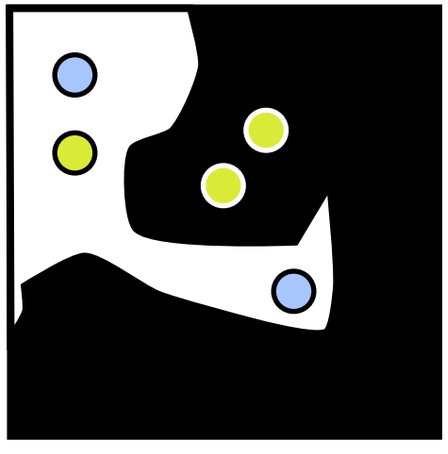
Game Log Analysis

Detect Maphack in P2P RTS

Area this player can see

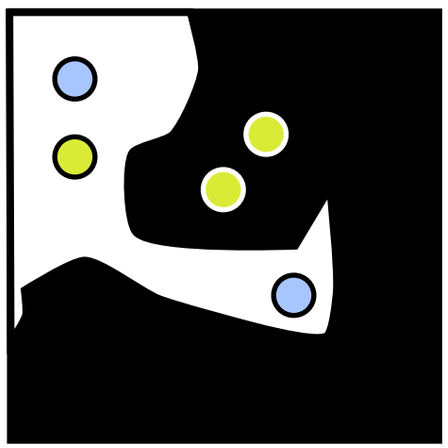
Fog of war





idea: Send every movement and viewing area to someone during gameplay.

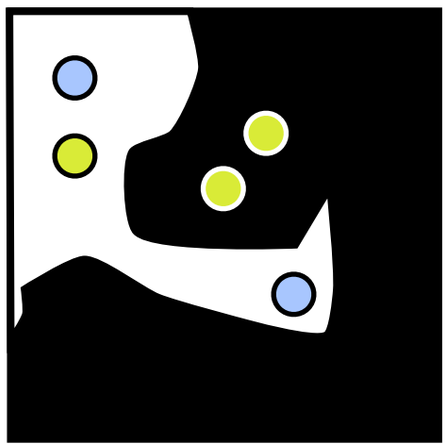
That someone replays the game to verify that the viewing area is correct.



no server -- other players have to check

idea: Send every movement and viewing area to another player during gameplay.

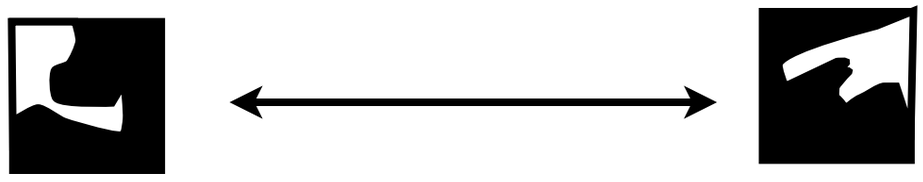
The other player replays the game to verify that the viewing area is correct.

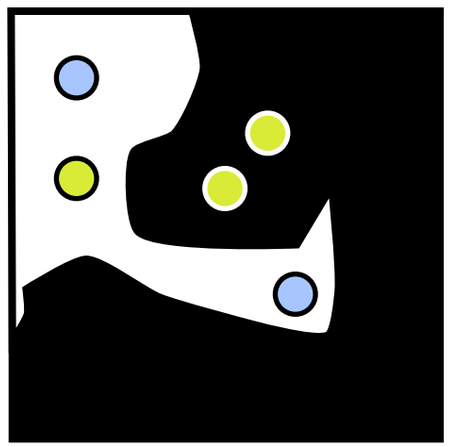


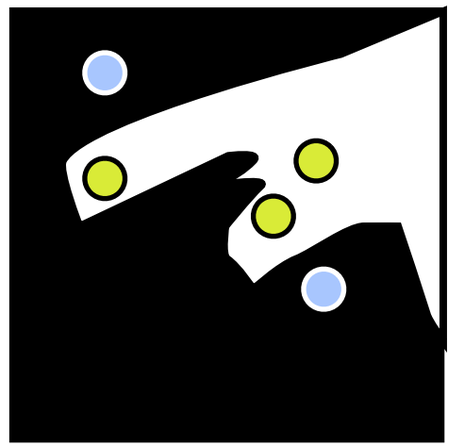
but, should not send movement in plain text

idea: use commitment protocol

1. send viewing area to each other



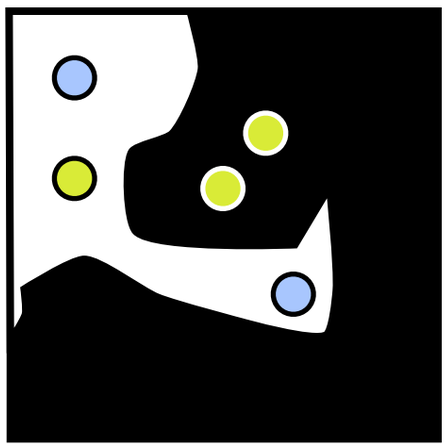


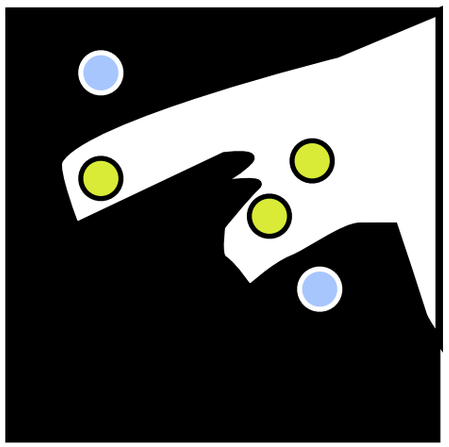


2. **for** each unit u,

if u is in opponent's view,

send in plain text

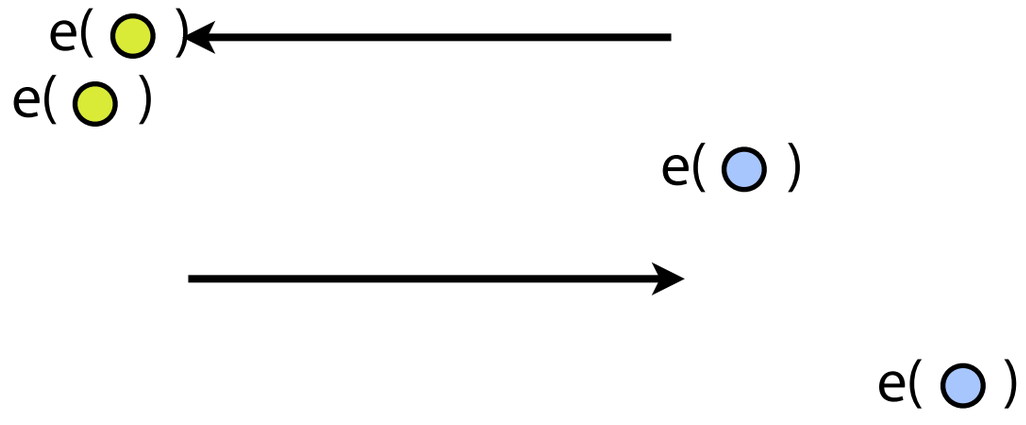


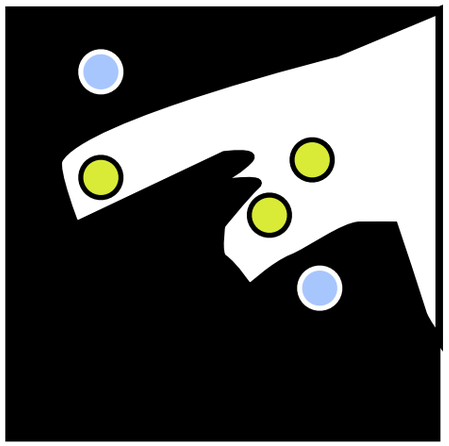


3. **for** each unit u

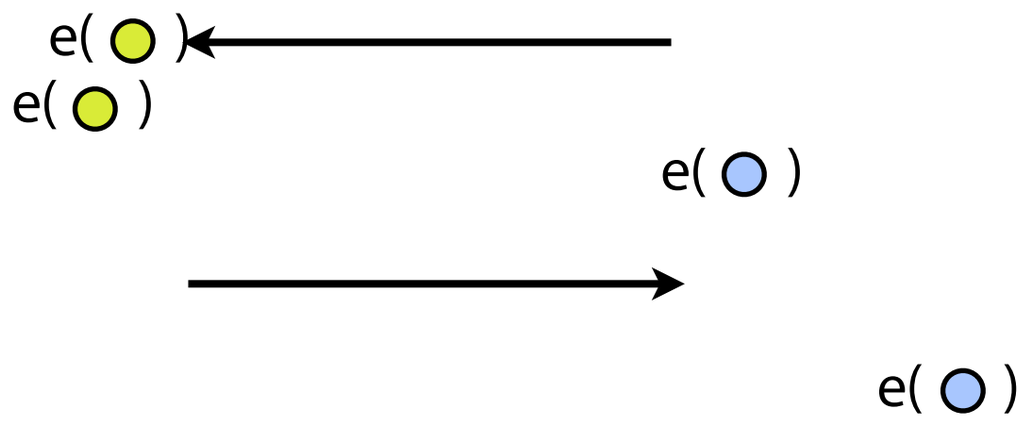
if u is hidden from opponent

send encrypted u





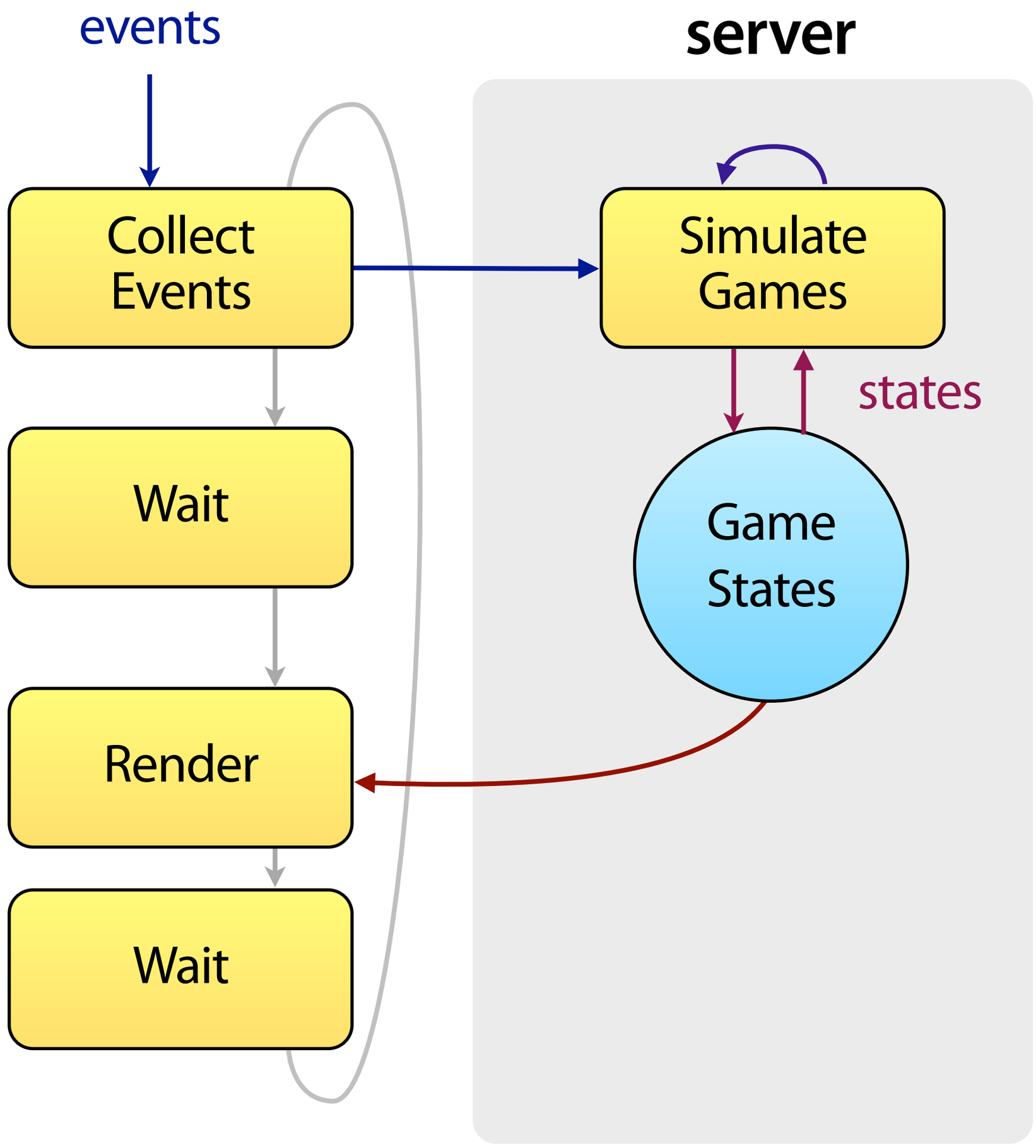
4. after the game, send decryption key

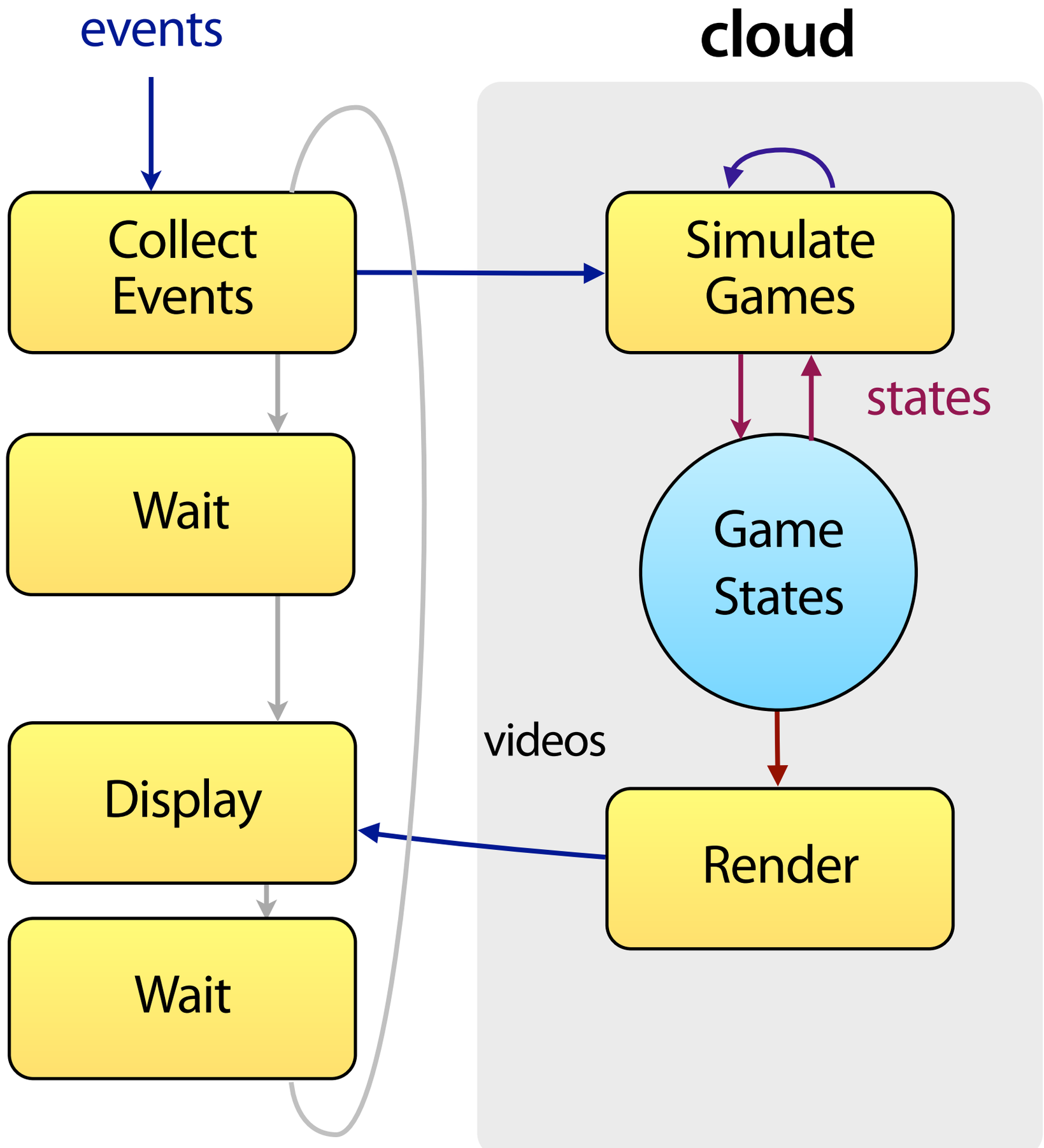


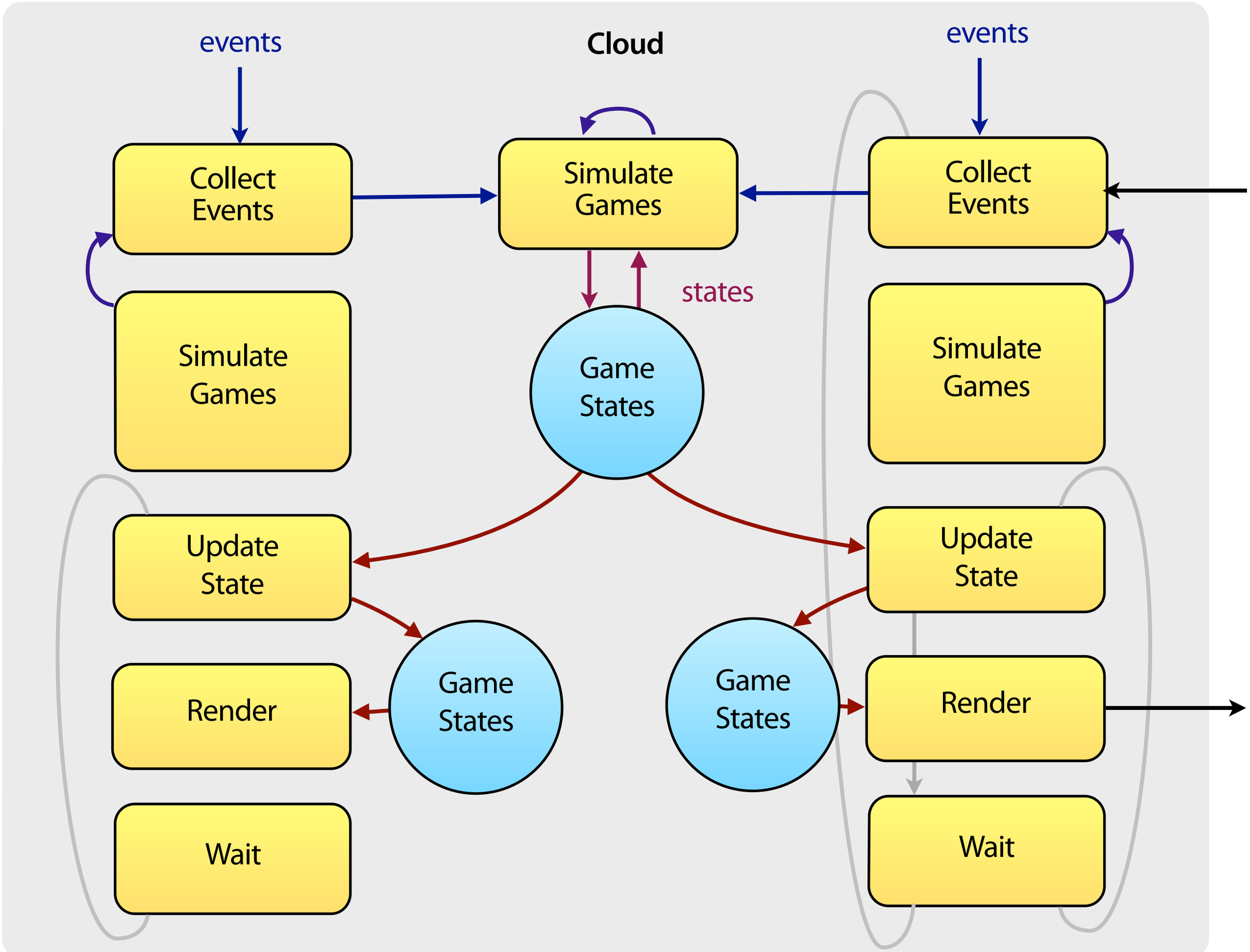
The challenge

how to prevent and reliably detect cheats in games?

Cloud Gaming


events
server
Collect Events
Simulate Games
states
Wait
Game States
Render
Wait


events
cloud
Collect Events
Simulate Games
states
Wait
Game States
videos
Display
Render
Wait


events
Cloud
events
Collect Events
Simulate Games
Collect Events
states
Simulate Games
Game States
Simulate Games
Update State
Update State
Render
Game States
Game States
Render
Wait
Wait

Why Cloud Gaming?

1. Can play on any client / anywhere

2. Reduce development cost

3. Prevent game piracy

Agawi

Any game. Anywhere. Instantly.

About Press Platform AGPP

Join the Agawi Game Partner Program (AGP Gold) for Windows 8 today!

onLIVE®

ABOUT GAMES PLAYPACK STORE

onLIVE®

Free your games. Free yourself.

OnLive delivers top-tier video games on demand to your TV, PC, Mac® or tablet —whatever you have on hand. Sign in. Play. It's that simple.

Sign up FREE!

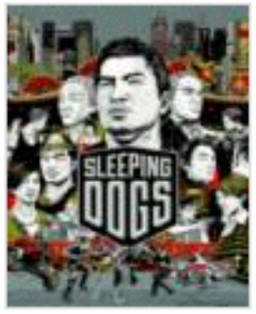
NEW! Desktop

Cloud Gaming

New Releases

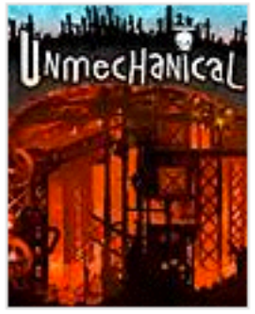
View All Games

Play now!



Learn More

Play now!



Play now!

Play now!

News

onLIVE PLAYPACK

All you can play. Anytime. Anywhere.

Over 200 games and growing! Unlimited play for only $9.99/month. Learn More

RECOMMENDED SYSTEM REQUIREMENTS

- Internet Connection: 5 Mbps, wired connection or Wi-Fi connection
- Operating System: Windows® 7 or Vista (32 or 64-bit) or XP SP3 (32-bit), M
- Computer: Dual-core PCs, all Intel-based Macs
- Screen Resolution: 1280x720
- Graphics card with Pixel Shader 2.0 or better (PC)
- CPU must support the SSE2 instruction set (PC)

Almost all OnLive users have had good results if their system meets the above r

Systems Requirements for OnLive

threatened legal action, contending that the license agreement did not permit the use of Windows 7 as a hosted client, nor for Office to be provided as a service on Windows 7 since this would be only allowed using Windows Server and Terminal Services.[50] On the 7 April 2012 it was discovered that the OnLive Desktop Service had changed and had now begun to use Windows Server 2008, presumably to settle this dispute.[51]

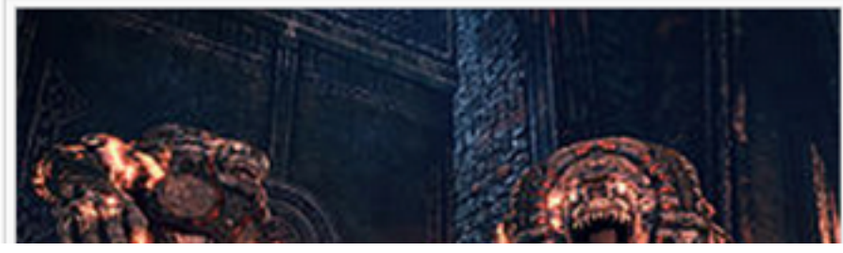
Architecture [edit]

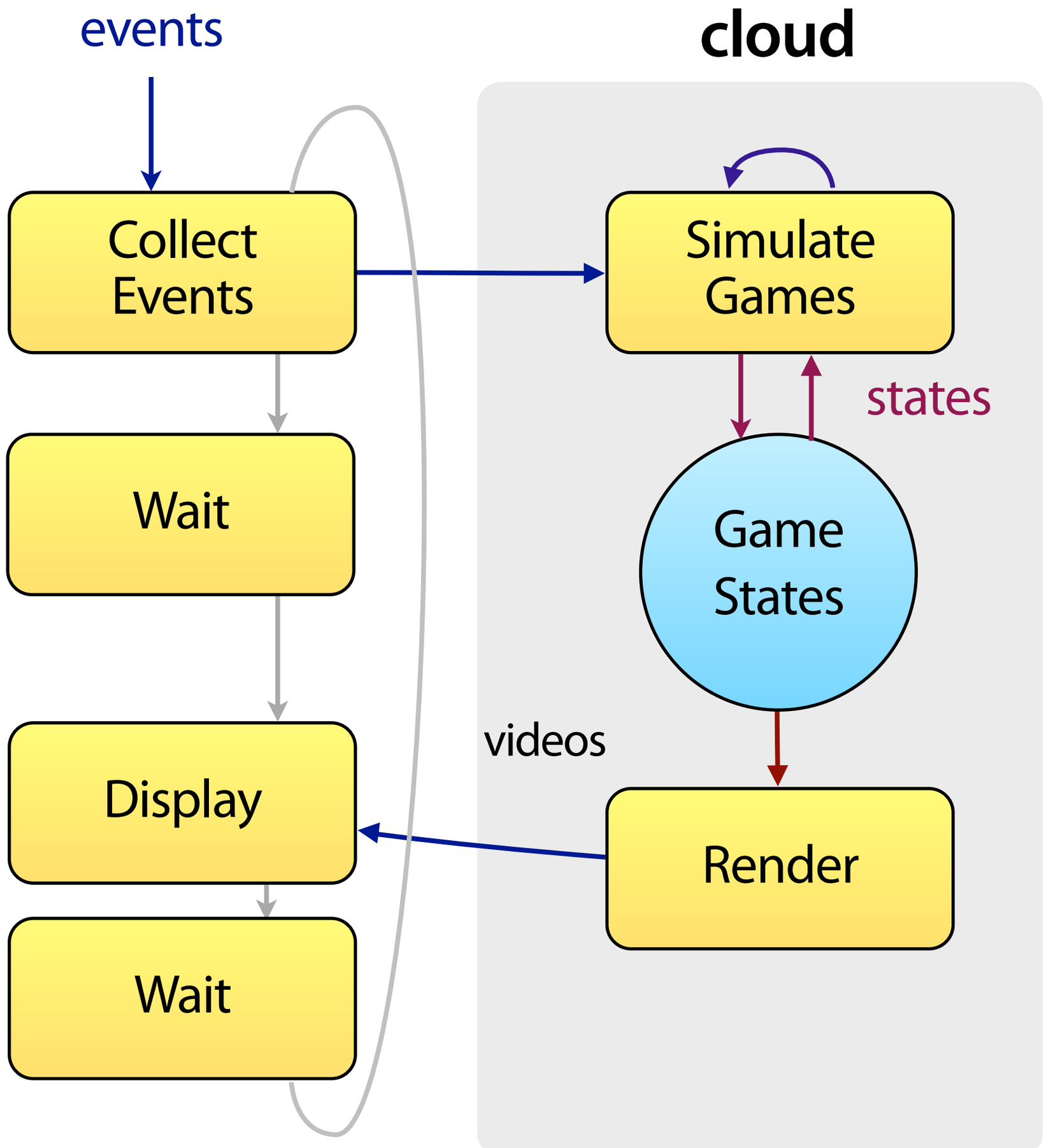
In the U.S., OnLive is hosted in five co-located North American data centers. There are facilities in Santa Clara, California and Virginia, with additional facilities in Dallas, Texas, as well as Illinois, and Georgia.[52][53] OnLive has stated that users must be located within 1,000 miles (1,600 km) of one of these to receive high-quality service.[54][55][56]

The hardware used is a custom set up consisting of OnLive's proprietary video compression chip as well as standard PC CPU and GPU chips. For older, or lower-performance, games such as *Lego Batman*, multiple instances can be played on each server using virtualization technology. However, high-end games such as *Assassin's Creed II* will require one GPU per game. Two video streams are created for each game. One (the live stream) is optimised for gameplay and real-world Internet conditions, while the other (the media stream) is a full HD stream that is server-side and used for spectators or for gamers to record videos of their gameplay.[42]

International availability [edit]

The service was launched in the United Kingdom on September 22, 2011[26] in partnership with British Telecom as a bundled service with their broadband packages.[57] The company plans to make its service available in the rest of Europe as well.[58]




events
cloud
Collect Events
Simulate Games
states
Wait
Game States
videos
Display
Render
Wait

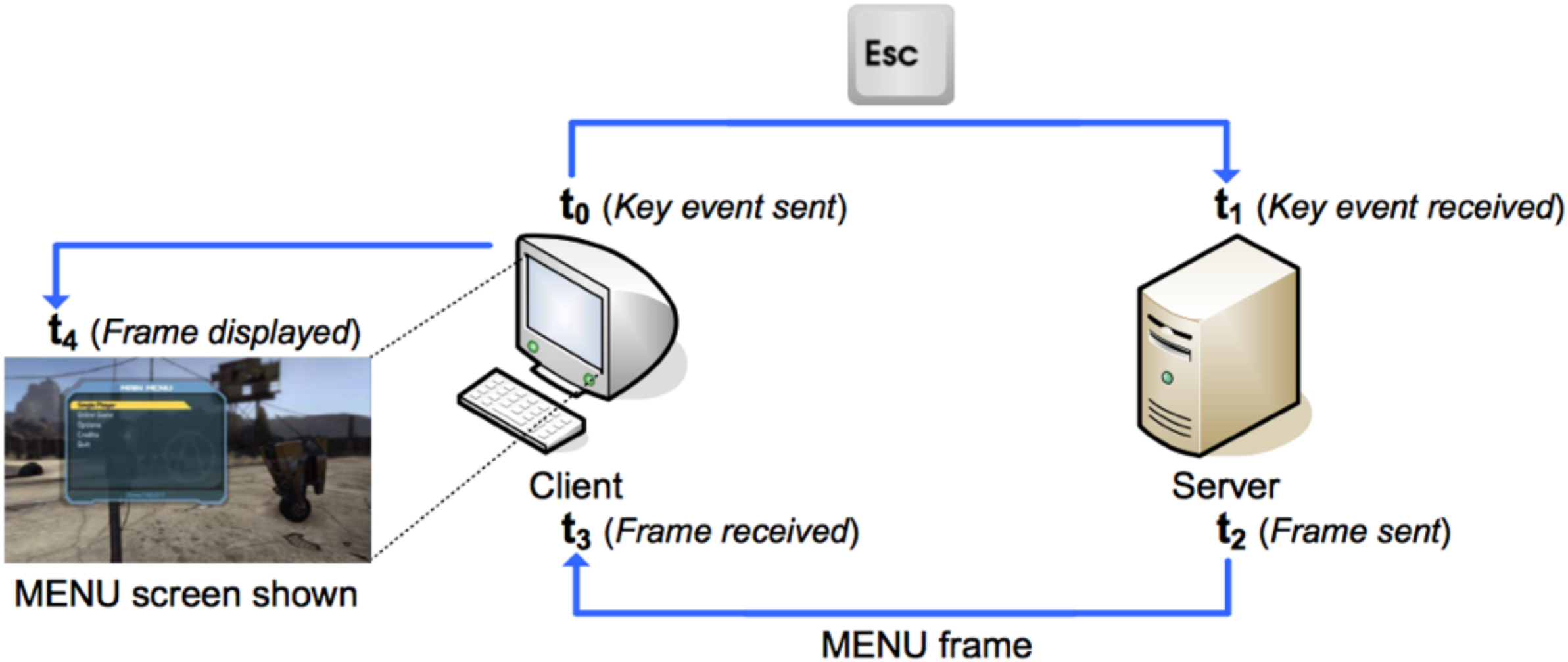

Esc
t0 (Key event sent)
t1 (Key event received)
t4 (Frame displayed)
Client
Server
t3 (Frame received)
t2 (Frame sent)
MENU screen shown
MENU frame

Figure 1

Probing t_3

- Send ESC at t_0
- t_{block} = rand(RD - 100, RD + 50) ms
- **if** menu appears before $t_0 + t_{block}$
  - $t_3 < t_0 + t_{block}$
- **else**
  - block inputs for 1 seconds
  - **if** menu appears after $t_0 + t_{block} + 1$
    - $t_3 > t_0 + t_{block}$

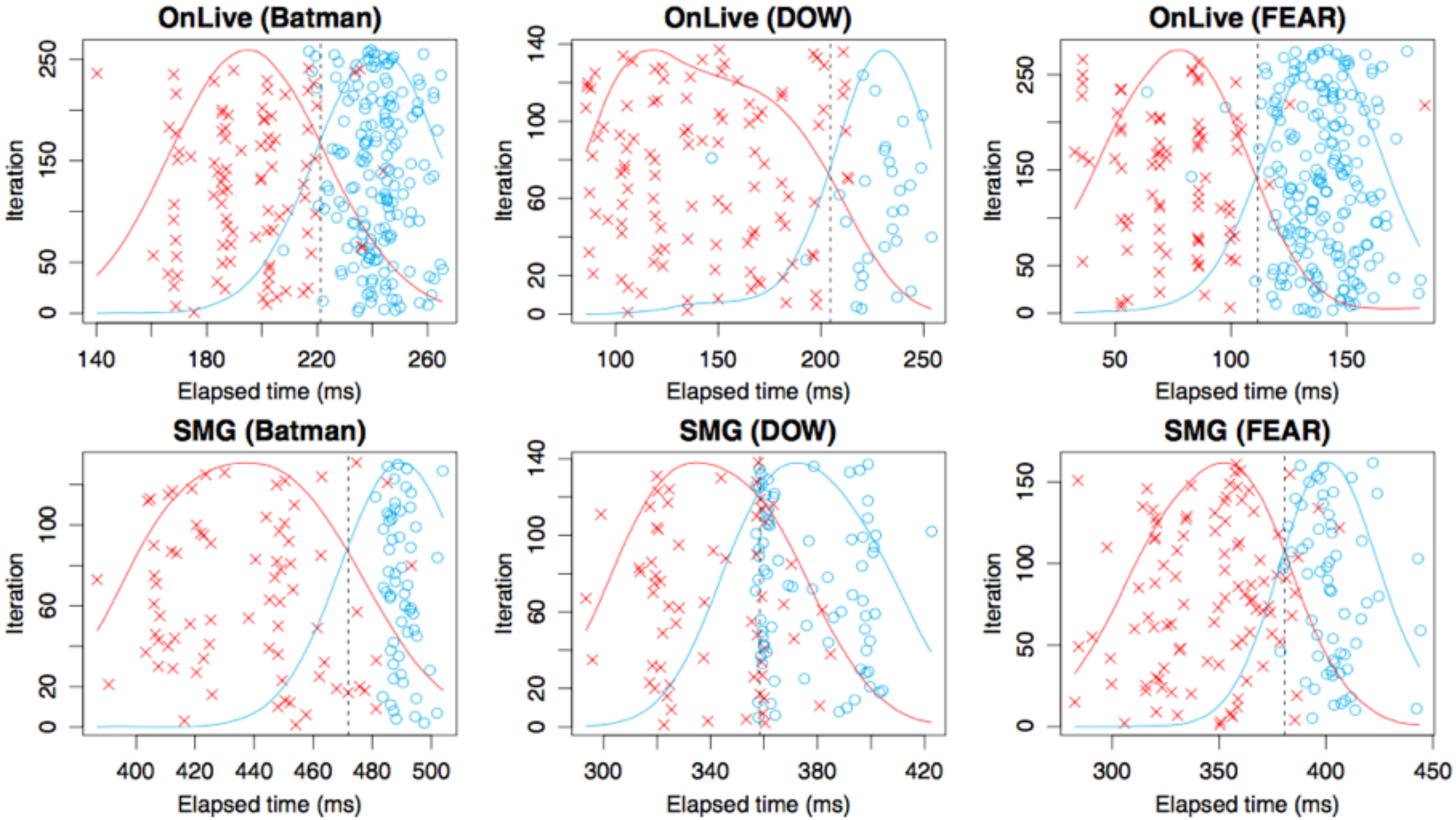

OnLive (Batman)
OnLive (DOW)
OnLive (FEAR)
SMG (Batman)
SMG (DOW)
SMG (FEAR)
Iteration
Elapsed time (ms)

Figure 3

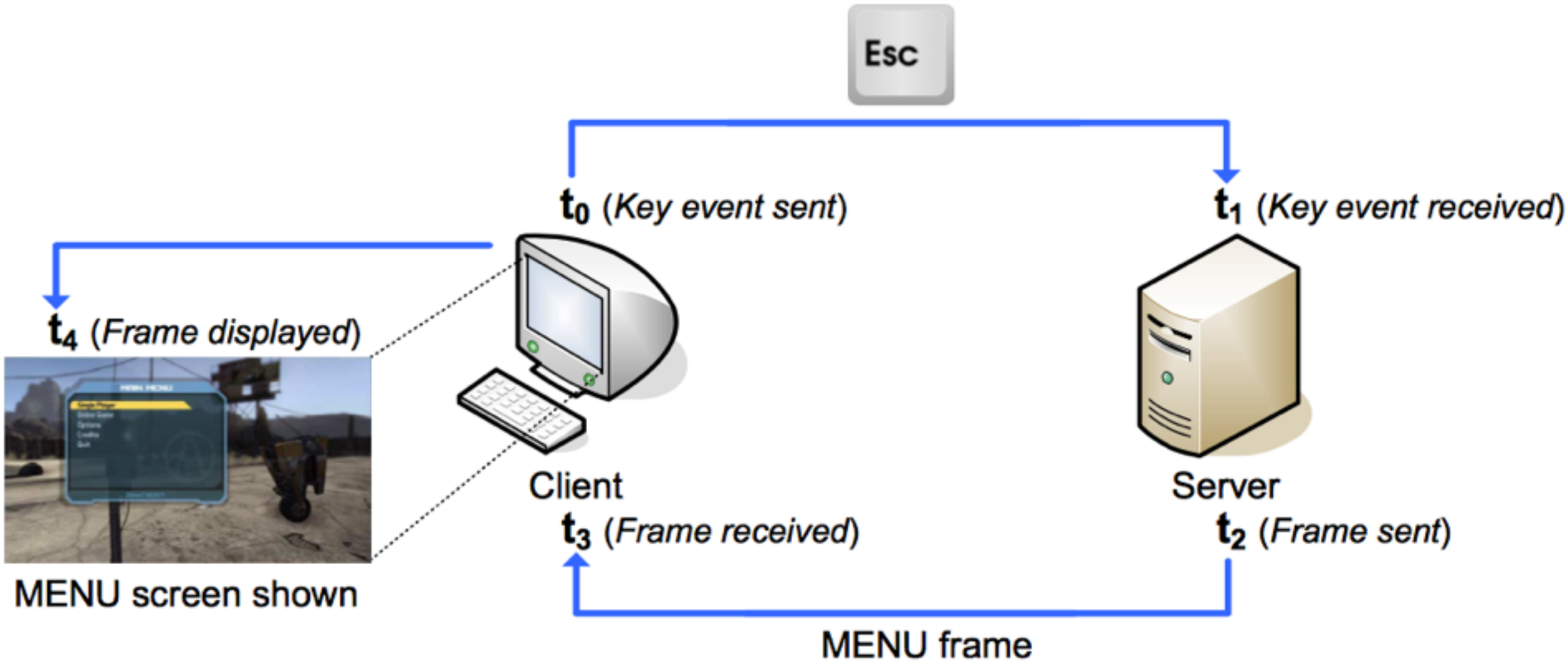

Esc
t0 (Key event sent)
t1 (Key event received)
t4 (Frame displayed)
Client
Server
t3 (Frame received)
t2 (Frame sent)
MENU screen shown
MENU frame

Figure 1

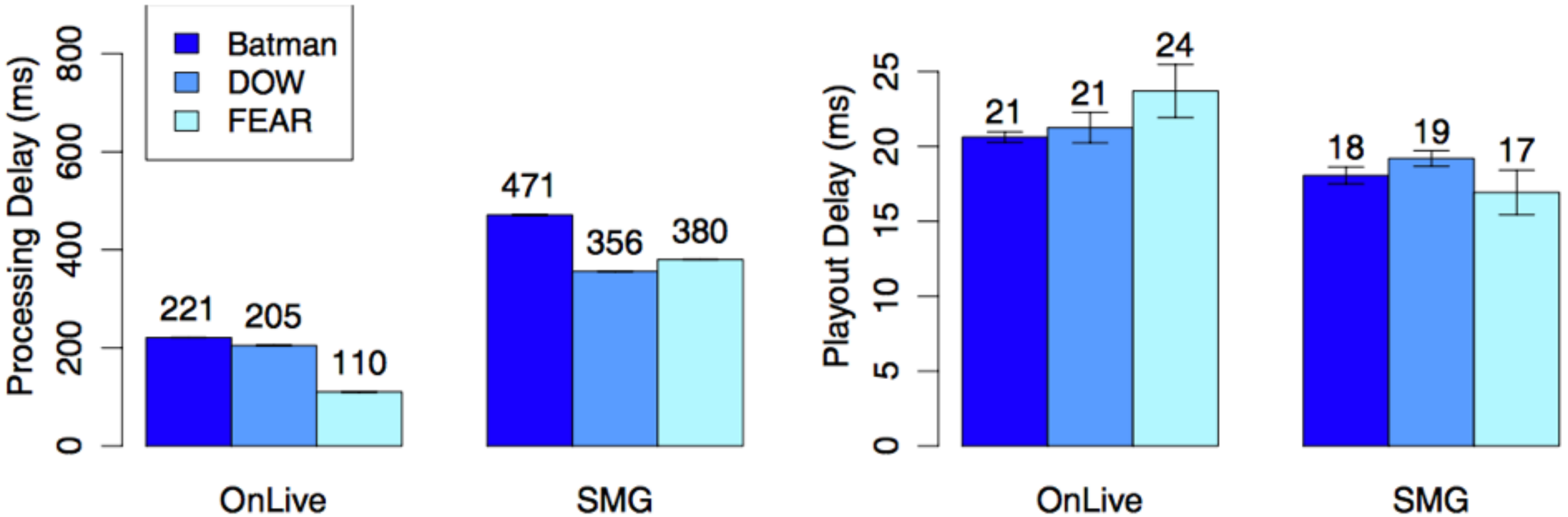
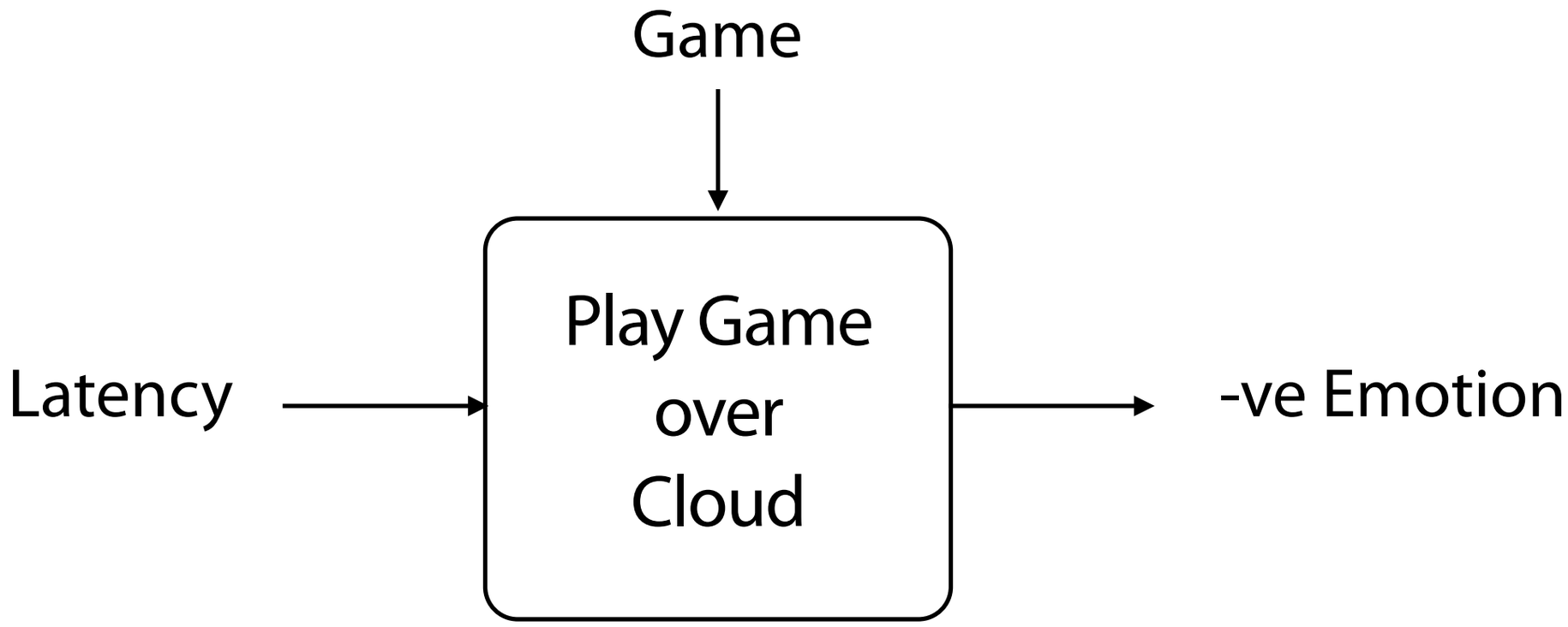

Batman
DOW
FEAR
Processing Delay (ms)
0
200
400
600
800
221
205
110
471
356
380
OnLive
SMG
Playout Delay (ms)
0
5
10
15
20
25
21
21
24
18
19
17
OnLive
SMG

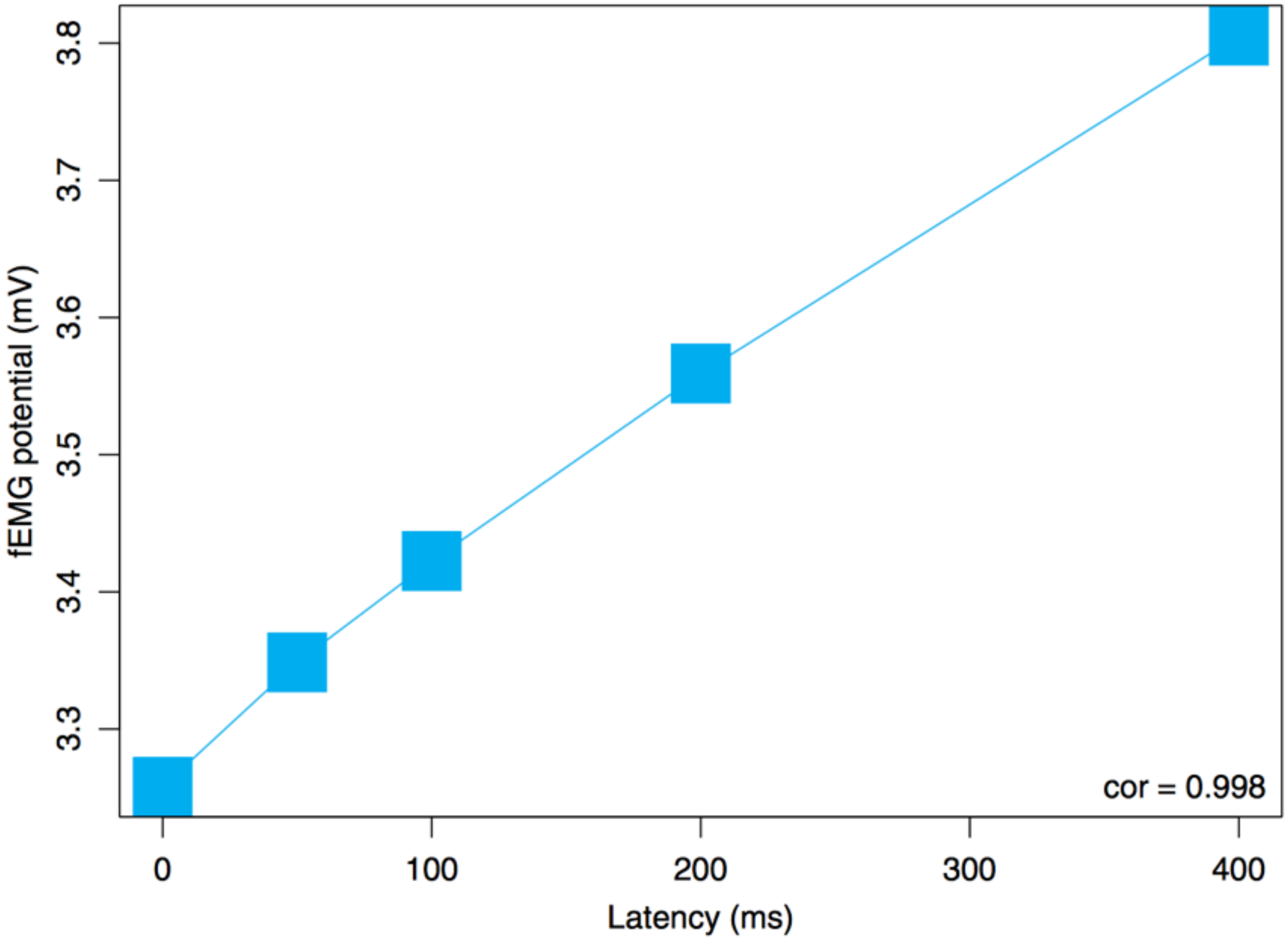
Figure 4

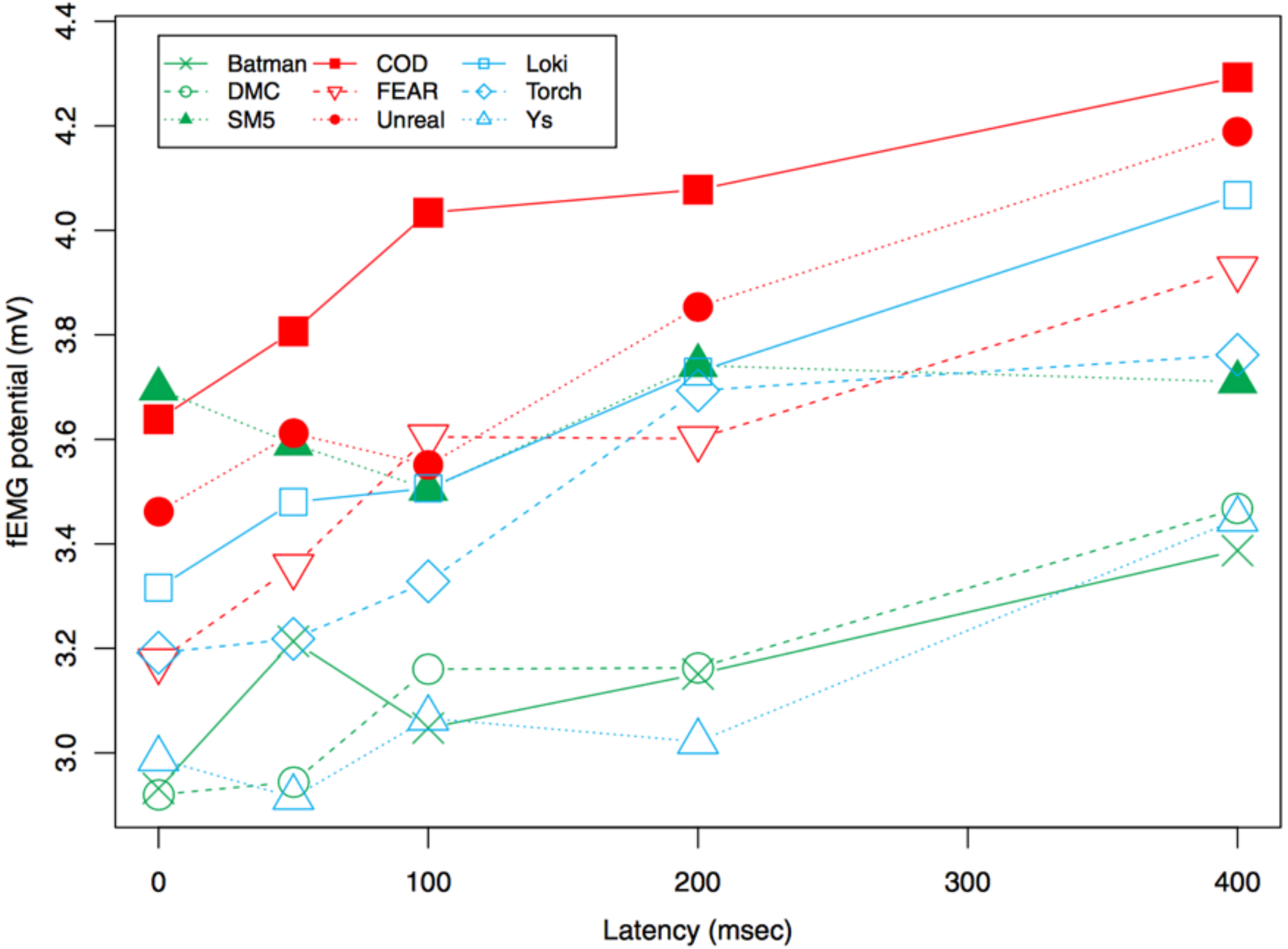
How cloud-gaming friendly is a game?


Game
Play Game over Cloud
Latency
-ve Emotion

Facial Electromyography

http://columbiabusinesstimes.com/17705/2013/05/13/read-my-mind/


fEMG potential (mV)
3.3
3.4
3.5
3.6
3.7
3.8
0
100
200
300
400
Latency (ms)
cor = 0.998


Batman
COD
Loki
DMC
FEAR
Torch
SM5
Unreal
Ys
fEMG potential (mV)
4.4
4.2
4.0
3.8
3.6
3.4
3.2
3.0
0
100
200
300
400
Latency (msec)

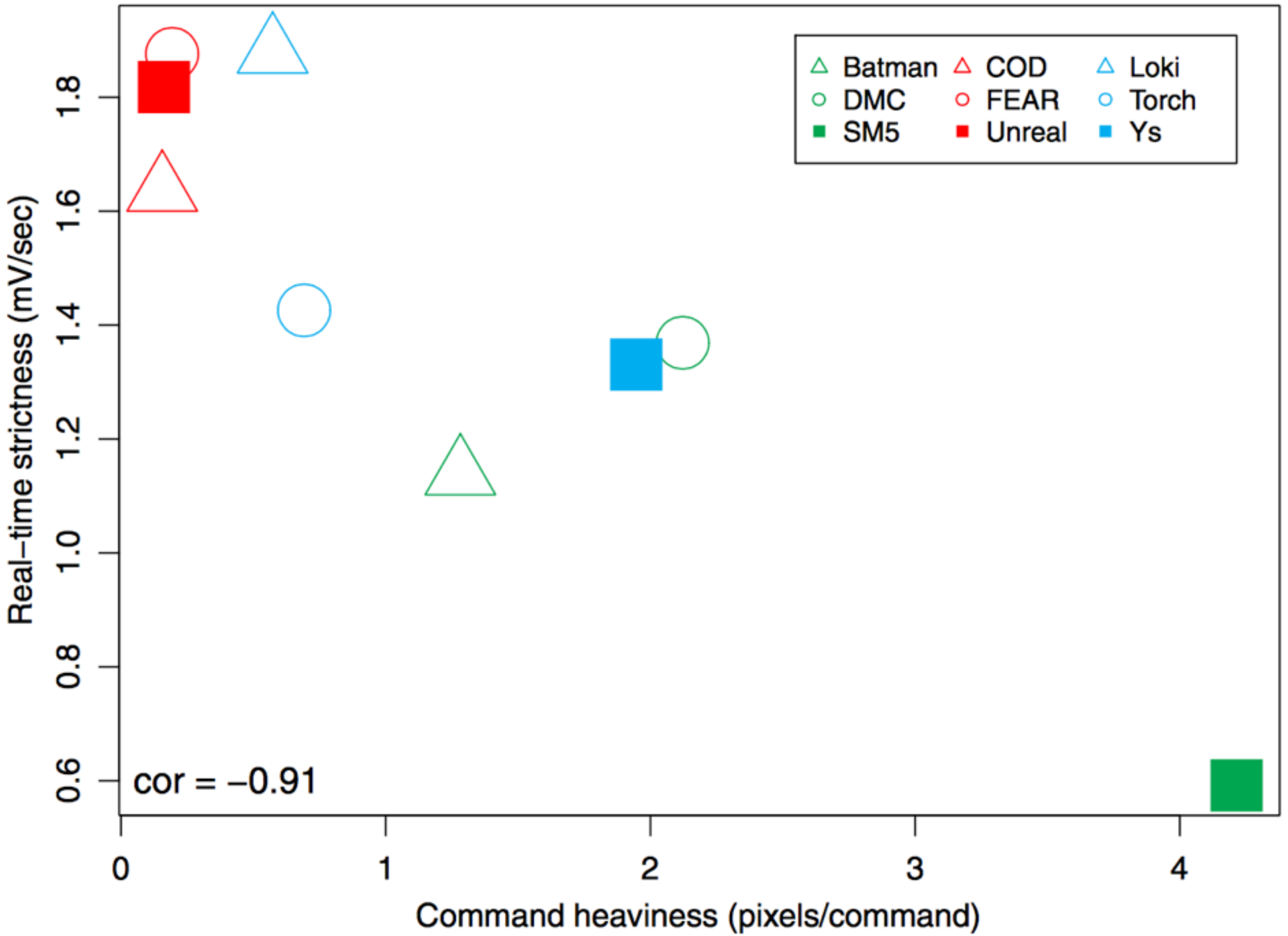
“Real-time Strictness”

Increase in Negative Emotion / Latency

“Command Heaviness”

If a single command leads to long and large amount of screen animation, then the game is said to be command-heavy.

Measured in pixels/command


Batman
COD
Loki
DMC
FEAR
Torch
SM5
Unreal
Ys
Real−time strictness (mV/sec)
0.6
0.8
1.0
1.2
1.4
1.6
1.8
cor = −0.91
0
1
2
3
4
Command heaviness (pixels/command)

Applications:

1. Allocate cloud server CPU/GPU

2. Server Location

Challenge 1:

Latency

Challenge 2:

Bandwidth

Challenge 3:

Hosting Cost

“the company had deployed thousands of servers that were sitting unused, and only ever had 1,600 concurrent users of the service worldwide.”

http://www.theverge.com/2012/8/28/3274739/onlive-report

The future?

virtualized GPU (e.g., Nvidia Grid)

better video codec

bigger network pipes

An Open Source Cloud Gaming Server