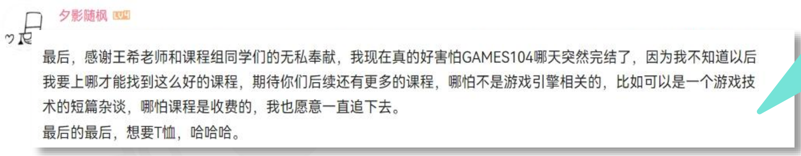
Voices from Community

- Thank you all for supporting the course team and Piccolo Community for the last 5 month !
- Beside the lectures, we are thinking about how to share more about game engine to the community
- Please post your ideas on Blibili Lecture 19, of **what else the course team could do to help you learning** (i.e. documents, videos on Piccolo codes or whatever you think will be meaningful)
- We will select 5 comments @GAMES-WEBINAR Bilibili Course 19, to send course T-shirts, deadline: 00:00 2022/08/22

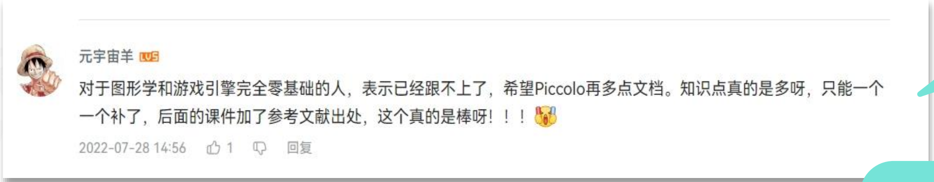


夕影随枫 LV4

最后，感谢王希老师和课程组同学们的无私奉献，我现在真的好害怕GAMES104哪天突然完结了，因为我不知道以后我要上哪才能找到这么好的课程，期待你们后续还有更多的课程，哪怕不是游戏引擎相关的，比如可以是一个游戏技术的短篇杂谈，哪怕课程是收费的，我也愿意一直追下去。

最后的最后，想要T恤，哈哈哈。

好害怕104哪天突然完结了，期待后续有更多的内容

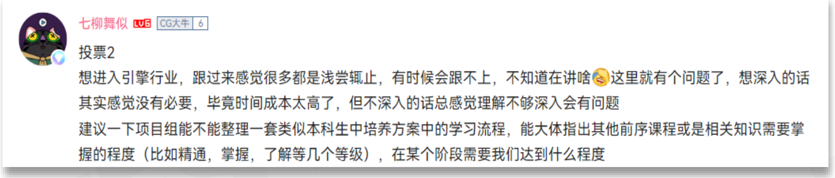


元宇宙羊 LV5

对于图形学和游戏引擎完全零基础的人，表示已经跟不上了，希望Piccolo再多点文档。知识点真的是多呀，只能一个一个补了，后面的课件加了参考文献出处，这个真的是棒呀！！！

2022-07-28 14:56 1 回复

希望Piccolo再多一点文档,知识点真是多啊



七柳舞似 LV6 CG大牛 6

投票2

想进入引擎行业，跟过来感觉很多都是浅尝辄止，有时候会跟不上，不知道在讲啥这里就有个问题了，想深入的话其实感觉没有必要，毕竟时间成本太高了，但不深入的话总感觉理解不够深入会有问题

建议一下项目组能不能整理一套类似本科生中培养方案中的学习流程，能大体指出其他前序课程或是相关知识需要掌握的程度（比如精通，掌握，了解等几个等级），在某个阶段需要我们达到什么程度

课程组能否整理一套类似本科生中培养方案的学习流程，大体指出其他前序课程或者是需要掌握的知识程度

Rewarding list for 精选留言

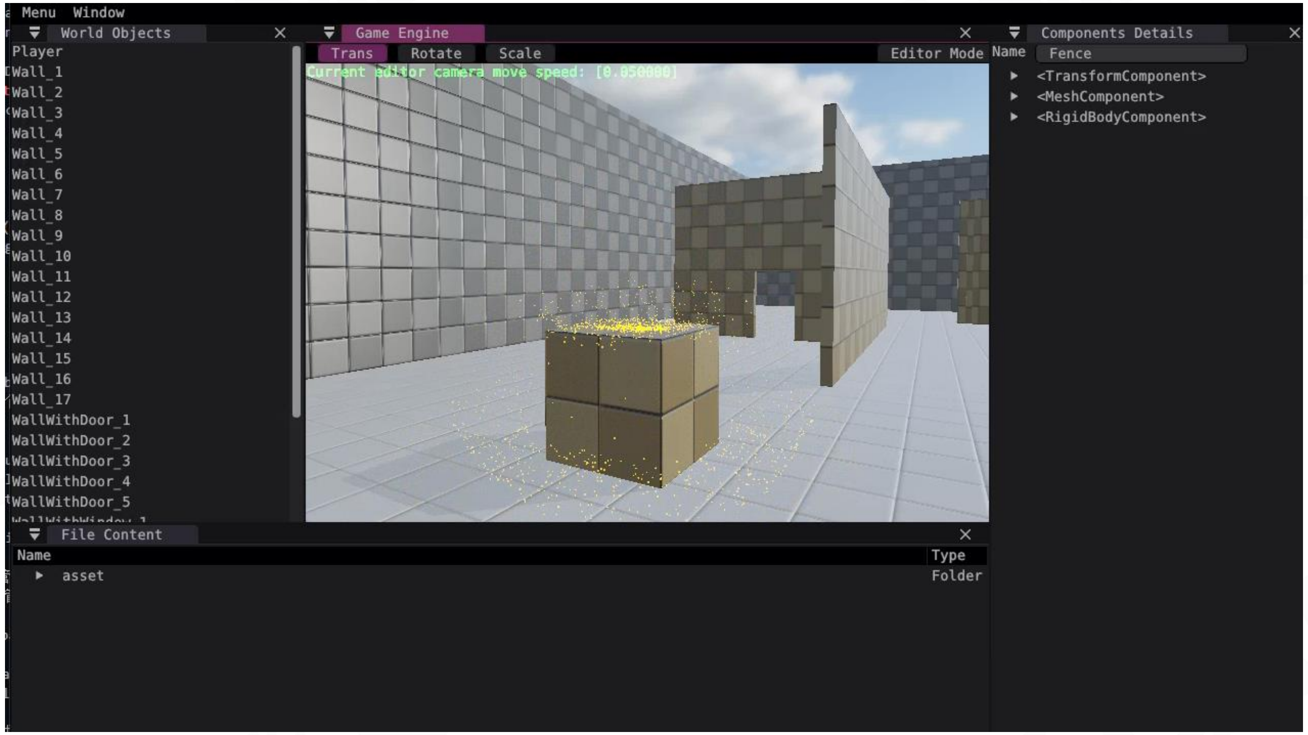
（请联系小秘书-阿曼达或“GAMES-WEBINAR” B站后台**发送联系方式；）**

@夕影随风 @七柳舞四 @多佛郎mingle @liangyush@我不是小杰

@剑锋不快 @Welann @暖风游戏厅 @云上男孩 @ANAΓKH

Piccolo Engine Following Updates (1/2)

- **RHI Optimization**
  - Better encapsulation of the RHI layer to prepare for the multi-graphics API (DX12, Metal) supports
- **GPU Particle System**
  - Emitters and Particles
  - GPU based particle simulation



Player
Wall_1
Wall_2
Wall_3
Wall_4
Wall_5
Wall_6
Wall_7
Wall_8
Wall_9
Wall_10
Wall_11
Wall_12
Wall_13
Wall_14
Wall_15
Wall_16
Wall_17
WallWithDoor_1
WallWithDoor_2
WallWithDoor_3
WallWithDoor_4
WallWithDoor_5

Current editor camera move speed: [0.050000]

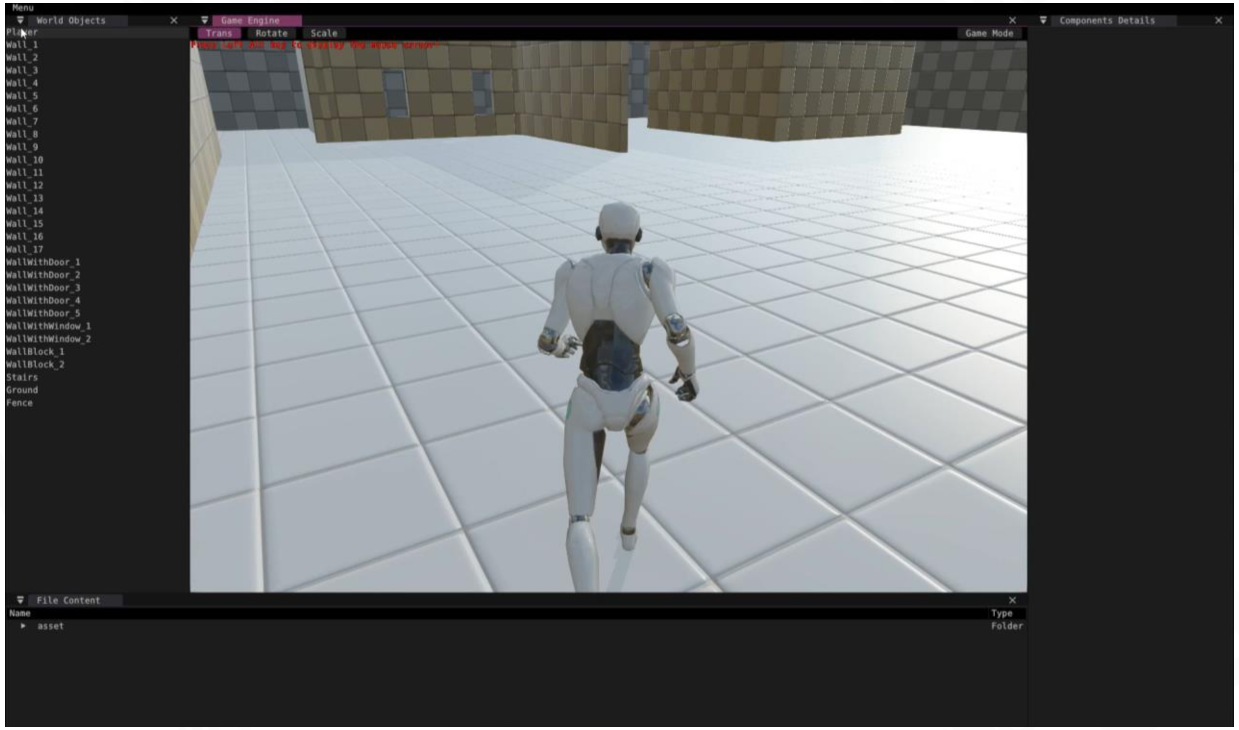
<TransformComponent>
<MeshComponent>
<RigidBodyComponent>

Name	Type
asset	Folder

Piccolo Engine Following Updates (2/2)

PICCOLO
Game engine

- **DebugDraw System**
  - Improve the debuggability of engine systems
  - Support drawing a variety of geometries: point, segment, box, sphere, cube, capsule, cylinder, text and triangle mesh



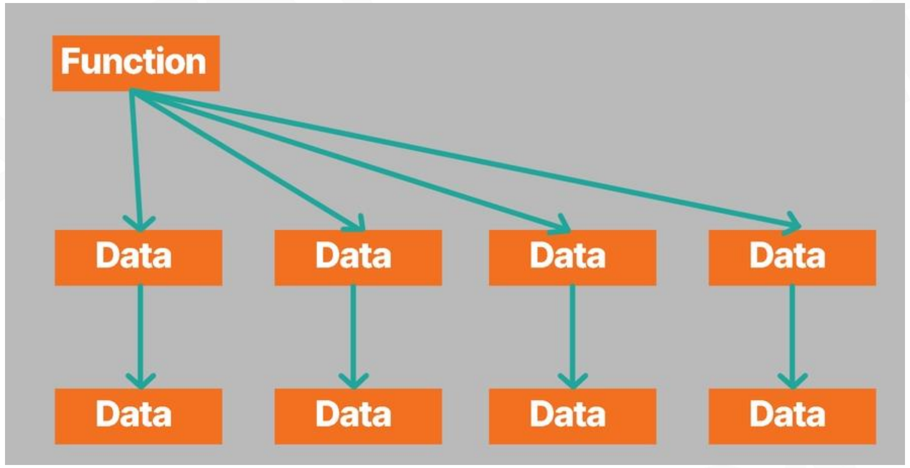
Press Left Alt key to display the mouse cursor

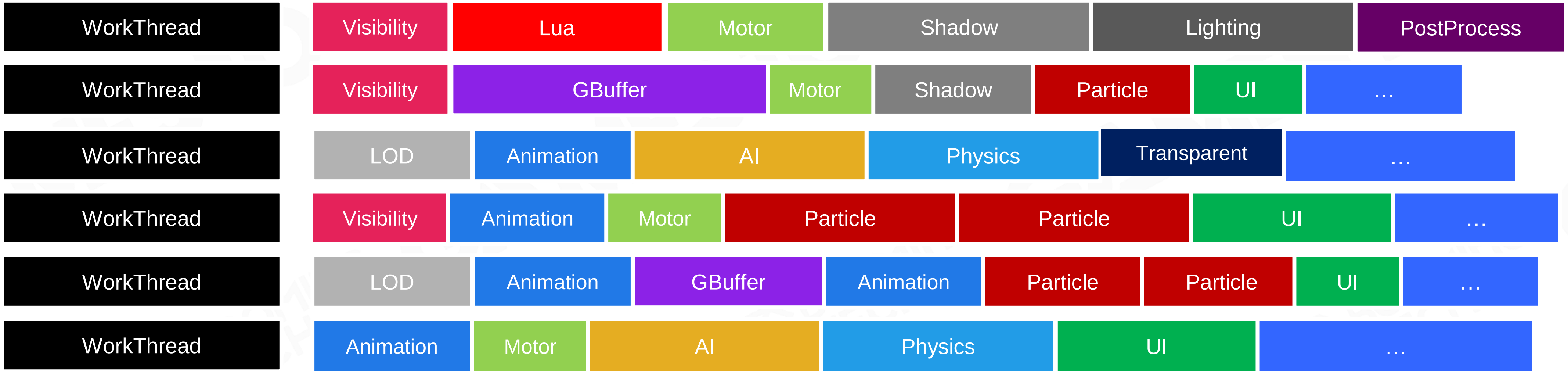
Q&A

- Q1:Can we mix up lockstep synchronization with state synchronization?
- Q2:If native cloud-based games came to real life one day, would network synchronization still be necessary?
- Q3:Can players cheat in state synchronization?

Advanced Topics (1/3) – DOP & Job System

- Data Oriented Programming (DOP)
- Job System





WorkThread | Visibility | Lua | Motor | Shadow | Lighting | PostProcess

WorkThread | Visibility | GBuffer | Motor | Shadow | Particle | UI | …

WorkThread | LOD | Animation | AI | Physics | Transparent | …

WorkThread | Visibility | Animation | Motor | Particle | Particle | UI | …

WorkThread | LOD | Animation | GBuffer | Animation | Particle | Particle | UI | …

WorkThread | Animation | Motor | AI | Physics | UI | …

Advanced Topics (2/3) – Nanite & Lumen

Nanite

Lumen

Advanced Topics (3/3) – Motion Matching & PGC

- Motion Matching
- Procedurally Generated Content (PGC)

Lecture 19

Online Gaming Architecture

Advanced Topics

WANG XI

GAMES 104

2022

Outline

01.

Basics

- Network Protocols
  - TCP, UDP and Reliable UDP
- Clock Synchronization
- Remote Procedure Call (RPC)
- Network Topology
- Game Synchronization
  - Snapshot Sync.
  - Lockstep Sync.
  - State Sync.

02.

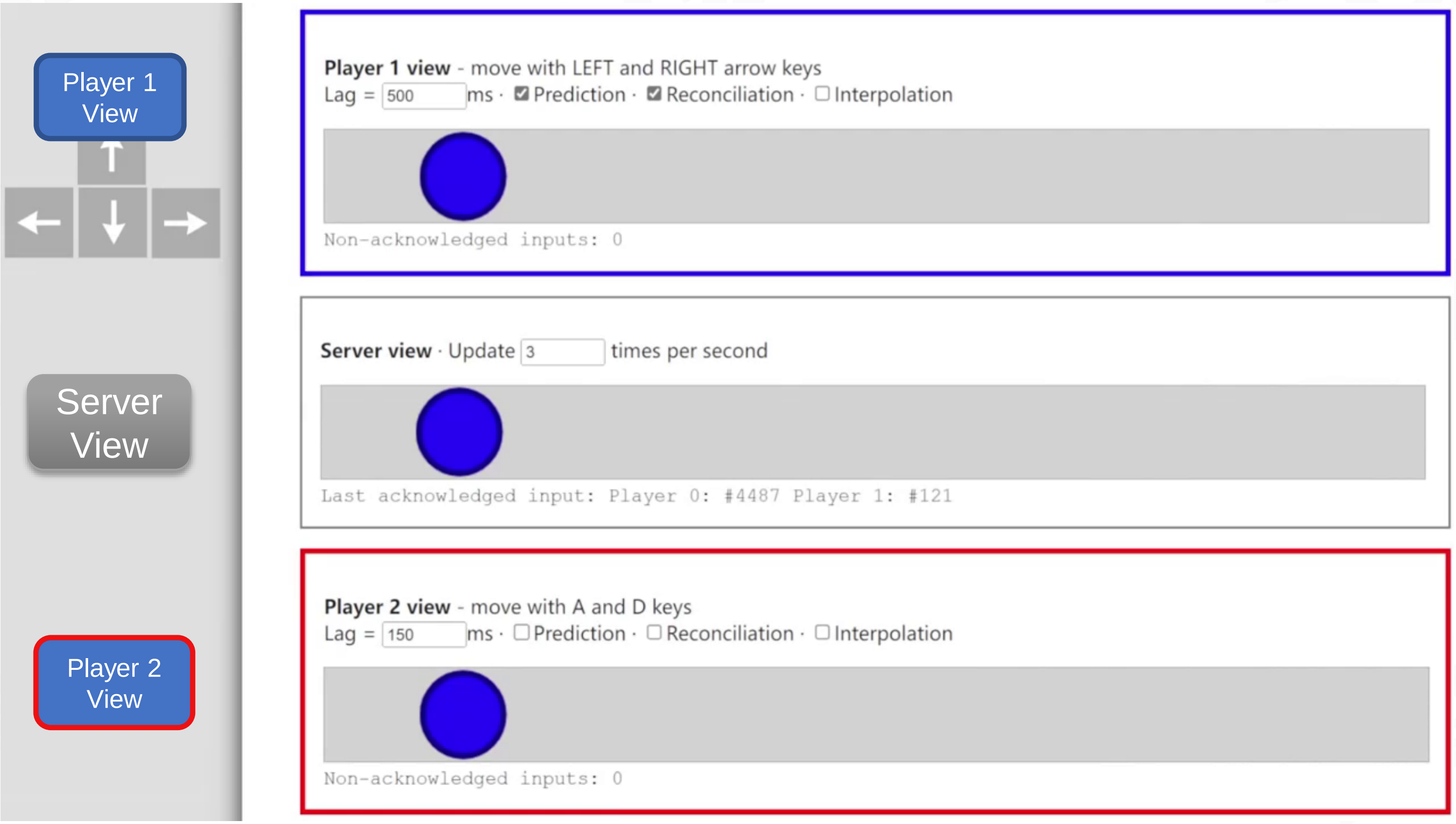
Advanced

- Character Movement Replication
- Hit Registration
- MMOG Network Architecture
- Bandwidth Optimization
- Anti-Cheat
- Build a Scalable World

Character Movement Replication

Character Movement Replication

From player 2's point of view, player 1's movement is very choppy and lags behind player 1's actual position



Player 1 view - move with LEFT and RIGHT arrow keys
Lag = 500 ms · ☑ Prediction · ☑ Reconciliation · □ Interpolation

```
Non-acknowledged inputs: 0
```

Server view · Update 3 times per second

```
Last acknowledged input: Player 0: #4487 Player 1: #121
```

Player 2 view - move with A and D keys
Lag = 150 ms · □ Prediction · □ Reconciliation · □ Interpolation

```
Non-acknowledged inputs: 0
```

Interpolation & Extrapolation

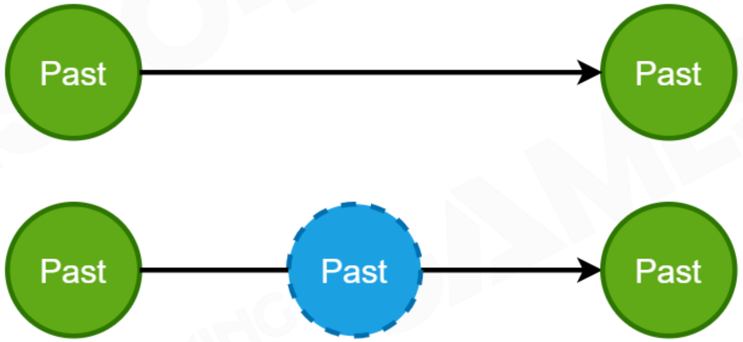
Purpose: **Smooth movement** of player's characters on screen

Interpolation

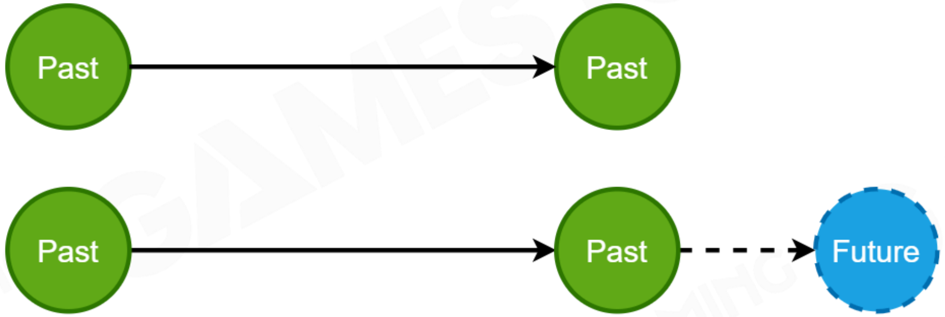
- Calculate the state between old but known states

Extrapolation

- Predict where entity is going from old states



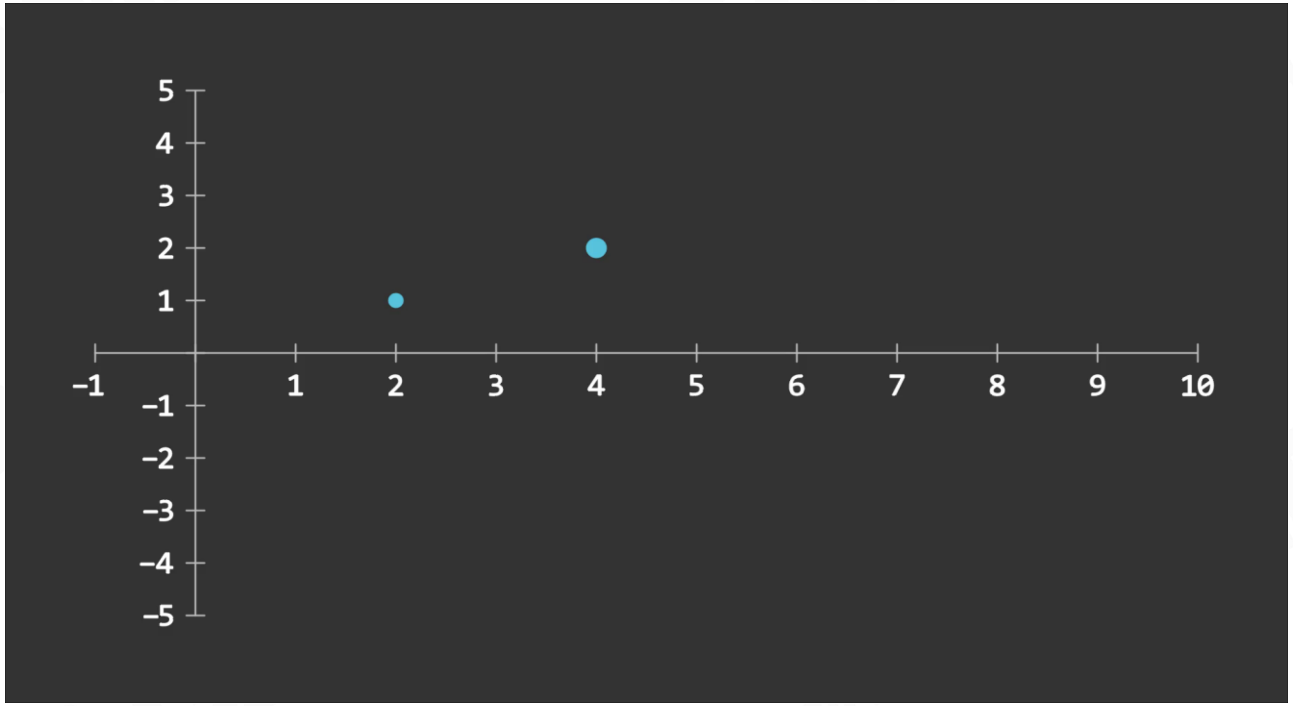
Interpolation



Extrapolation

Smooth States by Interpolations

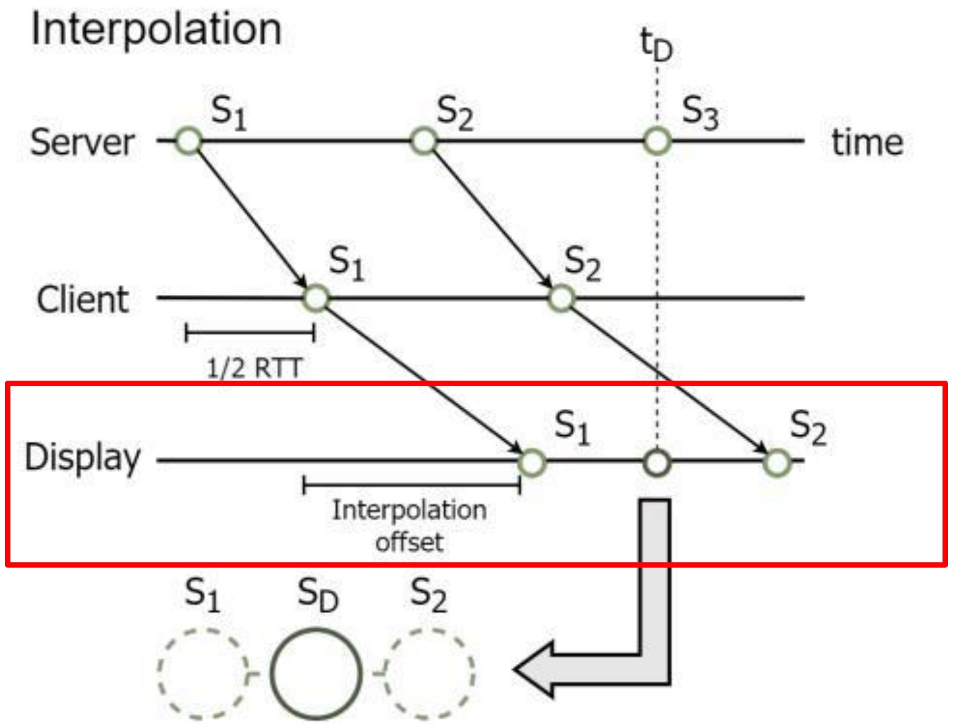
- Position and Orientation can be interpolated between two recently received data



Known state ●

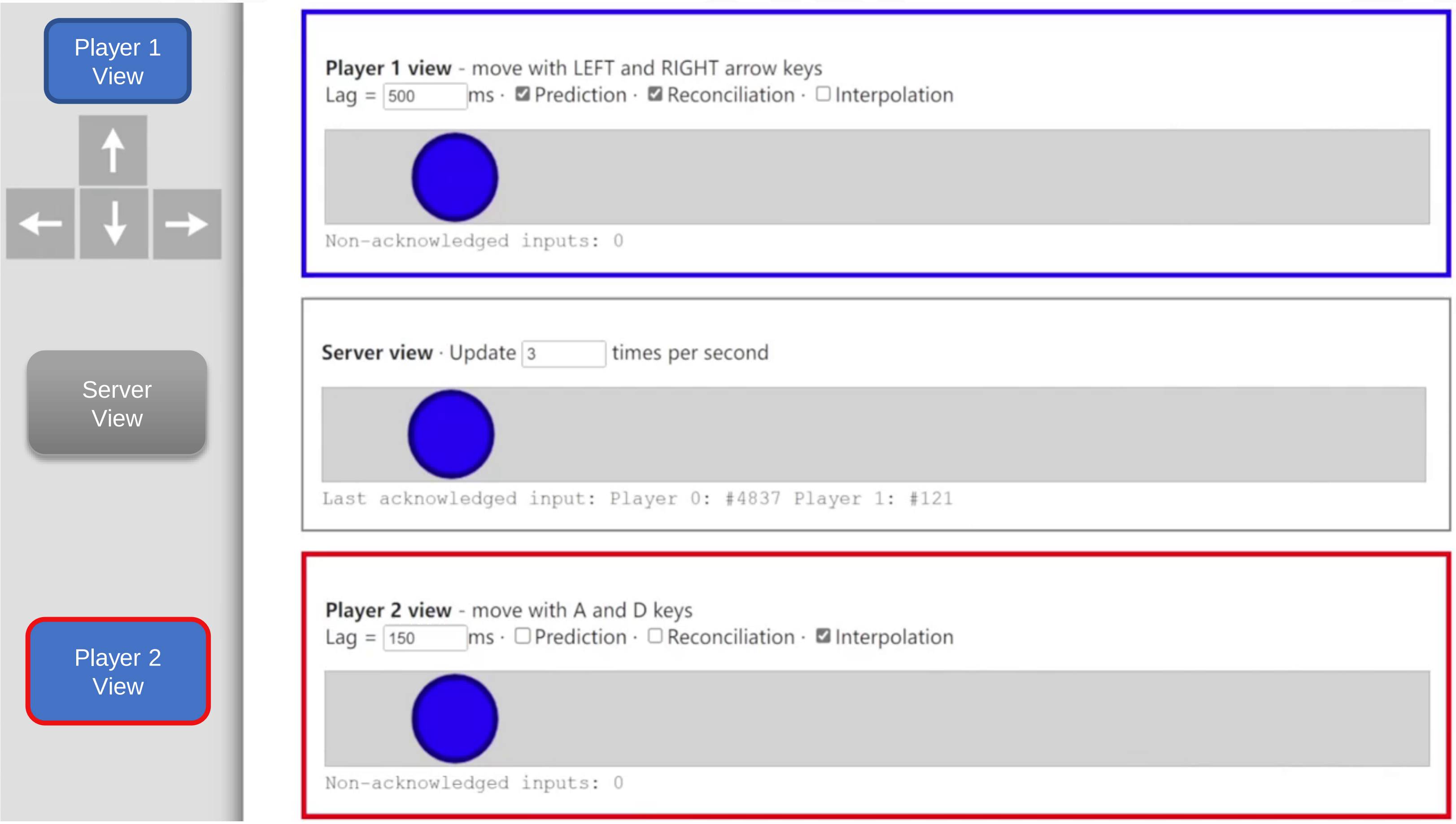
Buffer States and Deferred Render

- Data packet will not be rendered immediately when received
- Put into memory and wait for a new data packet
- After waiting for a time offset, start to render first received data packet
- Create an artificial delay of interpolation offset



Character Movement Replication by Interpolation

Result after interpolation was implemented



Player 1 view - move with LEFT and RIGHT arrow keys
Lag = 500 ms · ☑ Prediction · ☑ Reconciliation · □ Interpolation

Non-acknowledged inputs: 0

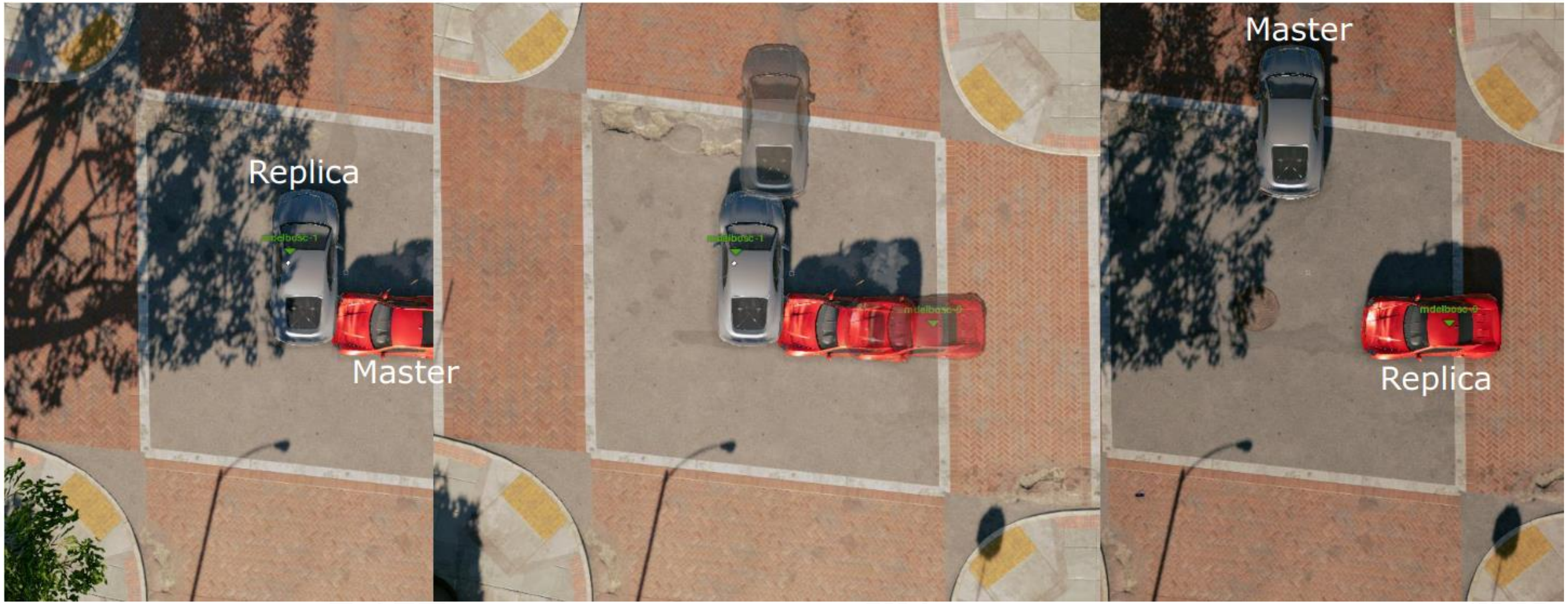
Server view · Update 3 times per second

Last acknowledged input: Player 0: #4837 Player 1: #121

Player 2 view - move with A and D keys
Lag = 150 ms · □ Prediction · □ Reconciliation · ☑ Interpolation

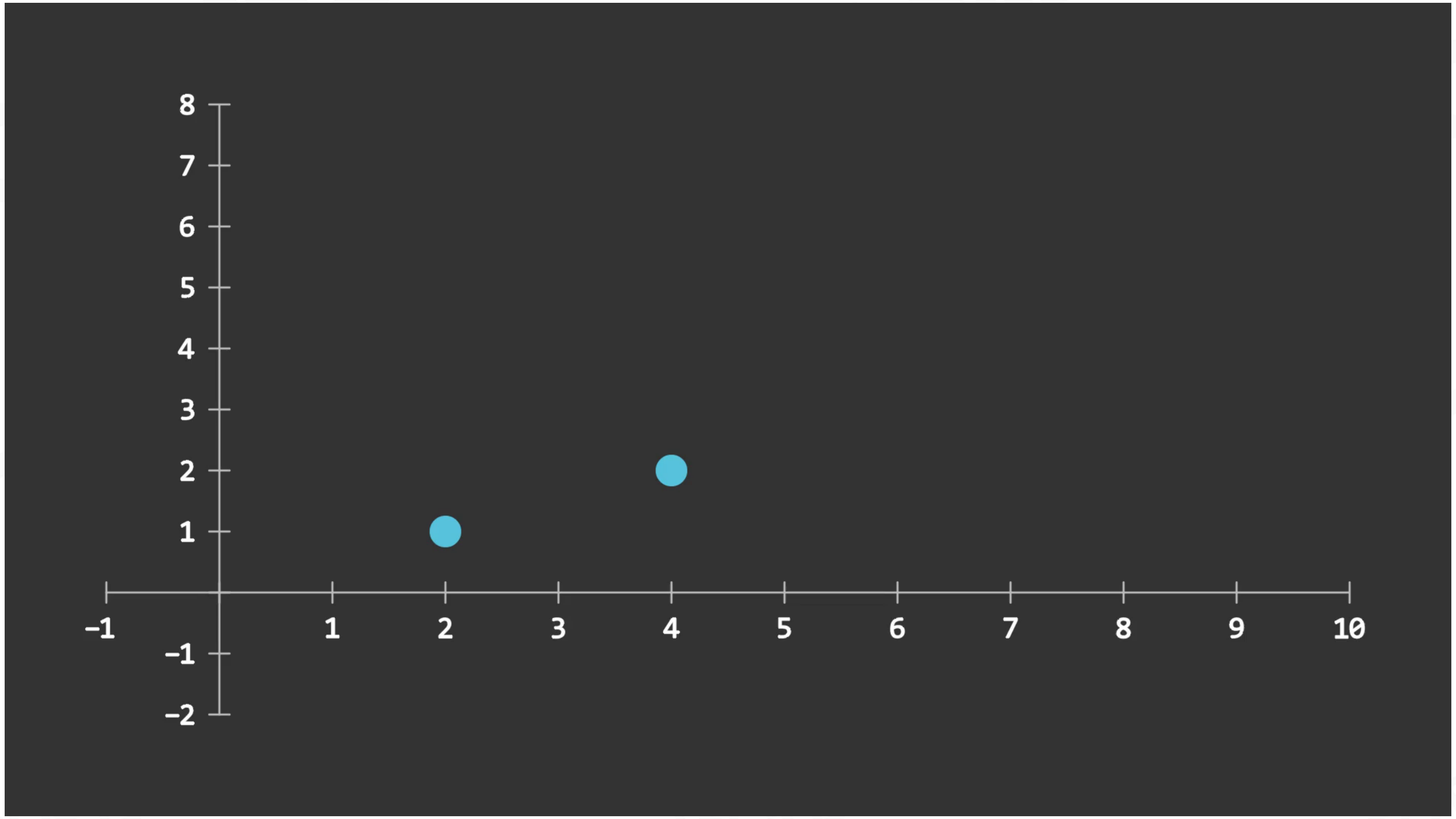
Non-acknowledged inputs: 0

Interpolation Challenges of Vehicle Movement Replication



Estimate Current State by Extrapolation

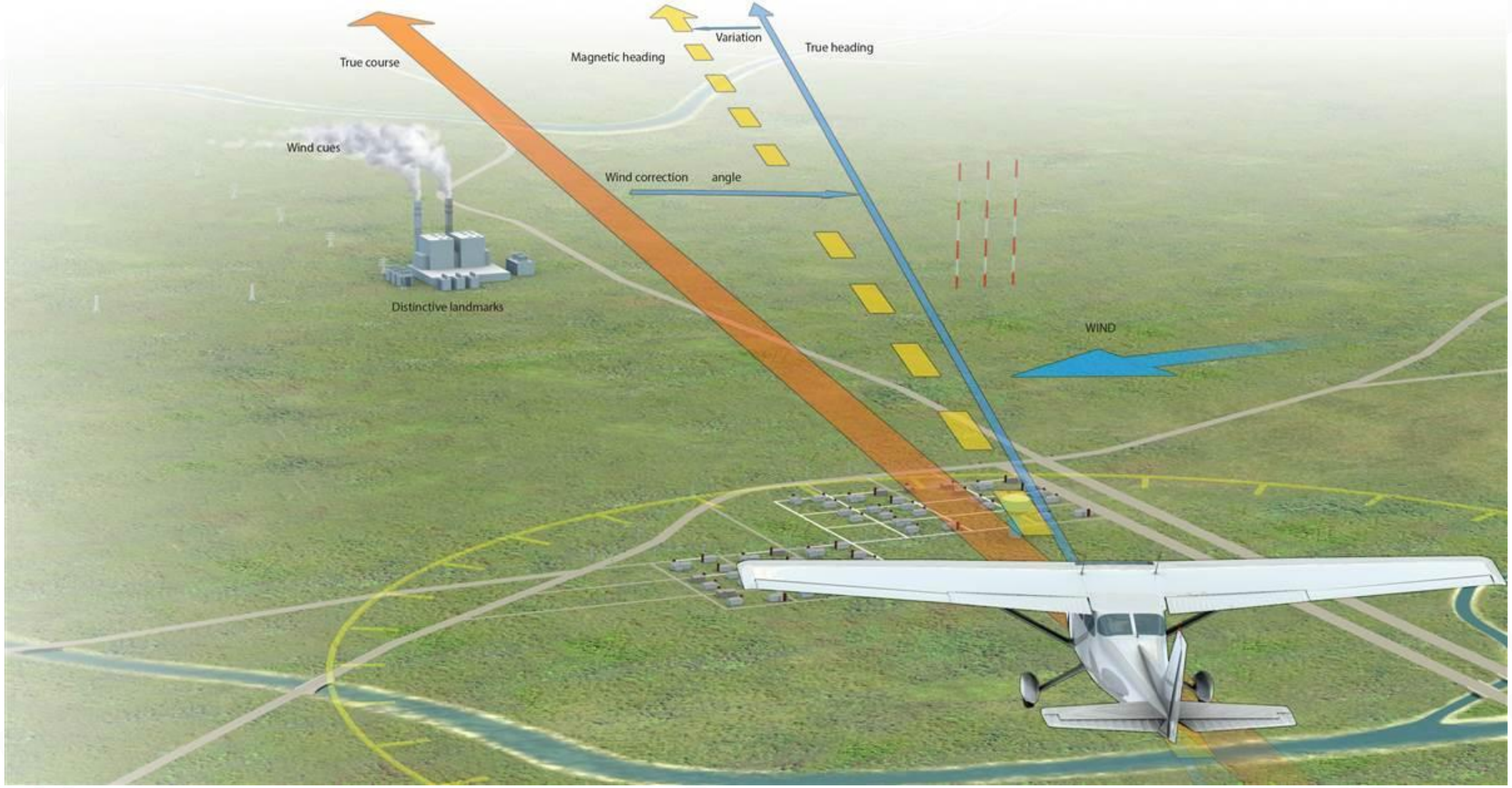
- Use **past** states to estimate **current** state to compensate net lag



Known state ● Predicted state ●

Dead Reckoning

- Estimate future state based on states that have been received

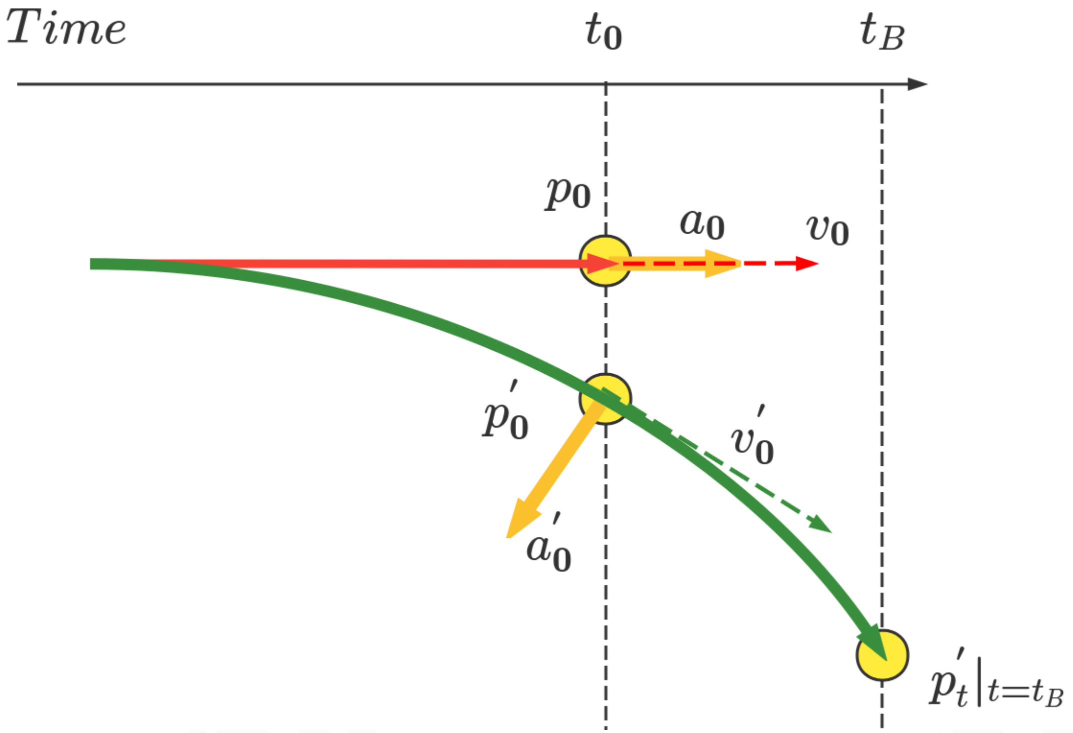


Projective Velocity Blending (1/2)

- At t_0, the replicated character is at p_0 with velocity v_0 and acceleration a_0, and receive the synced states with position p'_0, velocity v'_0, acceleration a'_0
- We can predict position p'_t after a time duration t based the synced states

$$p'_t = p'_0 + v'_0 t + \frac{1}{2} a'_0 t^2$$

- Our goal is to reach $p'_t|_{t=t_B}$ smoothly after a fixed blending time duration: $t_B - t_0$



Projective Velocity Blending (2/2)

At any time t, we can get the blending velocity v_t

$$\lambda = \frac{t - t_0}{t_B - t_0}$$

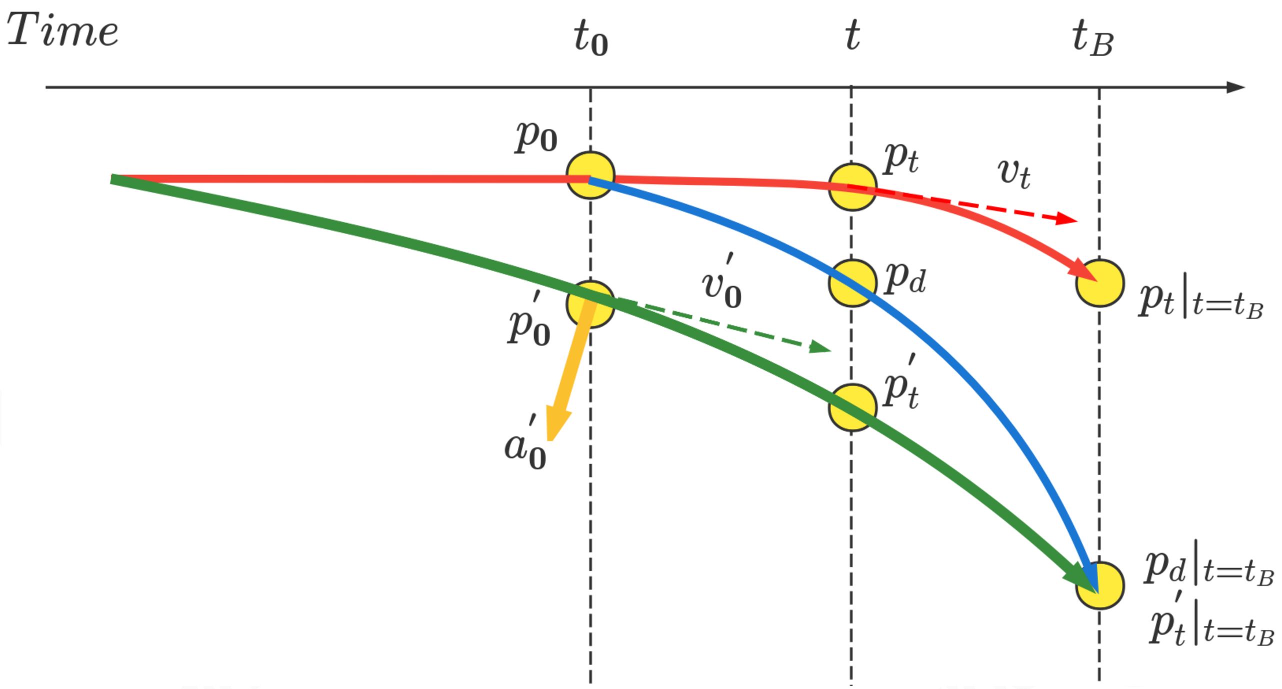
$$v_t = v_0 + \lambda(v_0' - v_0)$$

And projecting the position p_t from p_0

$$p_t = p_0 + v_t t + \frac{1}{2} a_0' t^2$$

Then get the dead reckoned position p_d by combining p_t and p_t'

$$p_d = p_t + \lambda(p_t' - p_t)$$



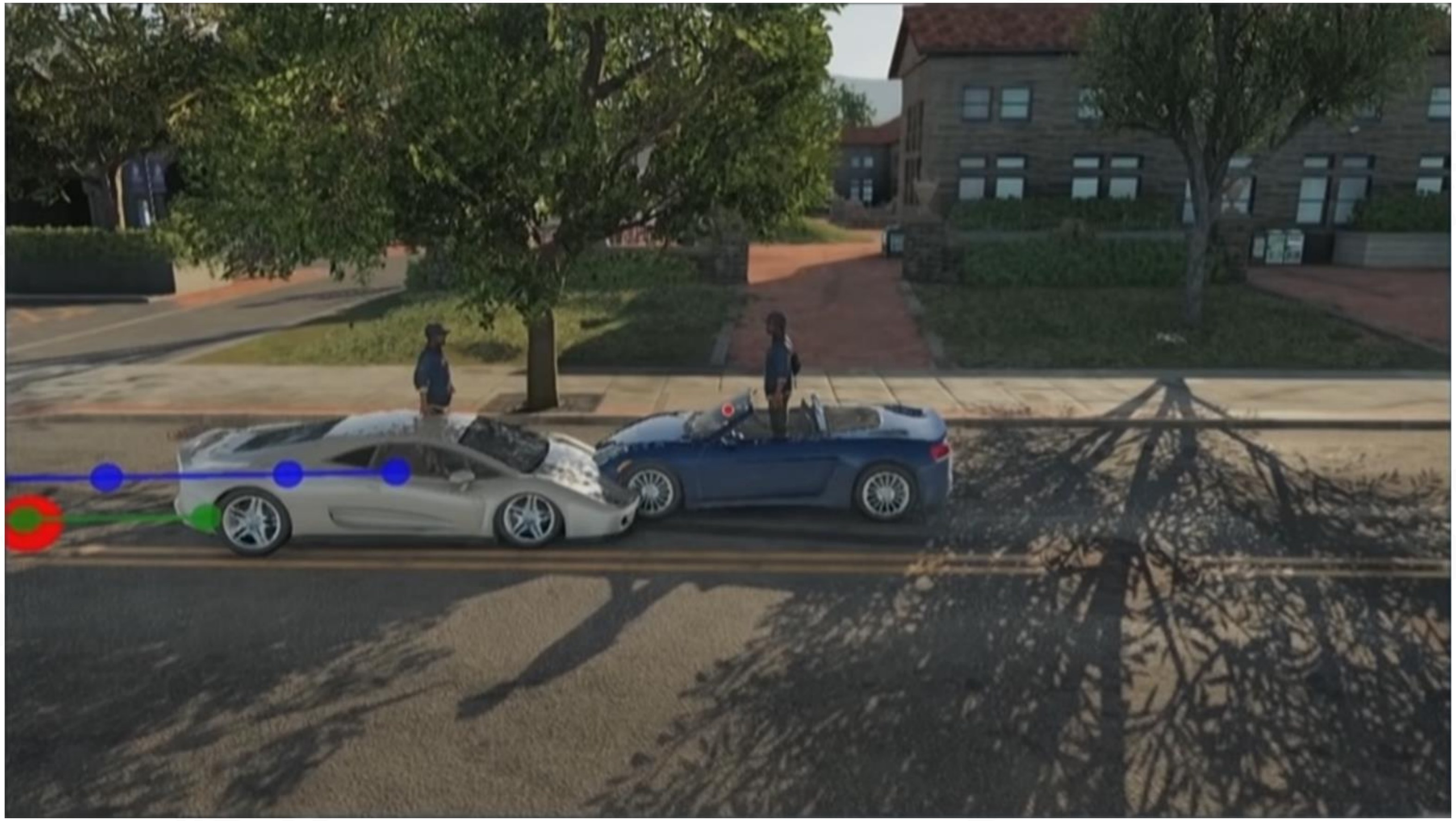
Collision Issues (1/4)

Dead reconking Collision trajection looks weird

Red: Snapshot
Blue: Simulated track
Green: Ground truth

Collision Issues (2/4)

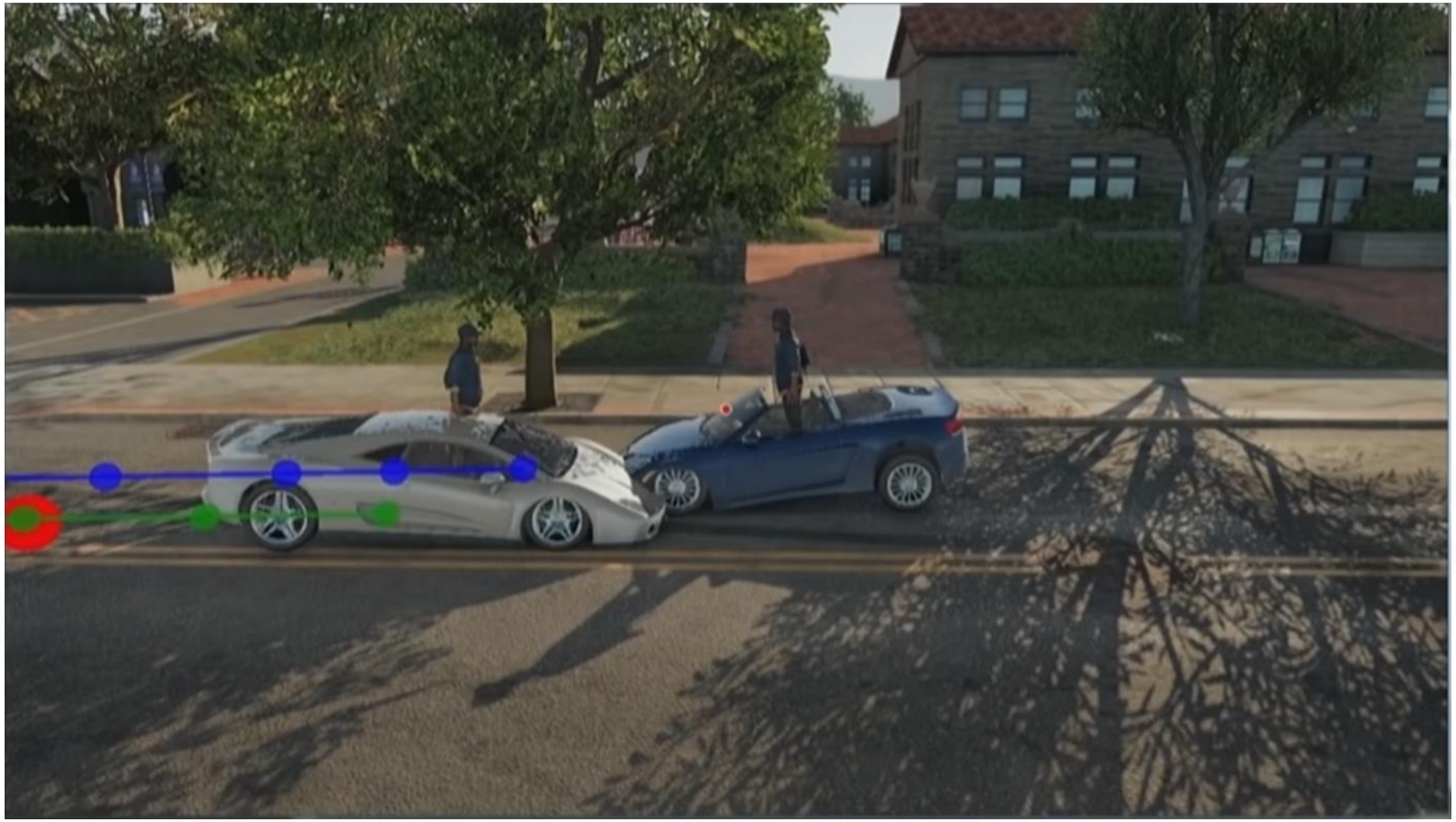
Phase 1: Collision starts



Red: Snapshot
Blue: Simulated track
Green: Ground truth

Collision Issues (3/4)

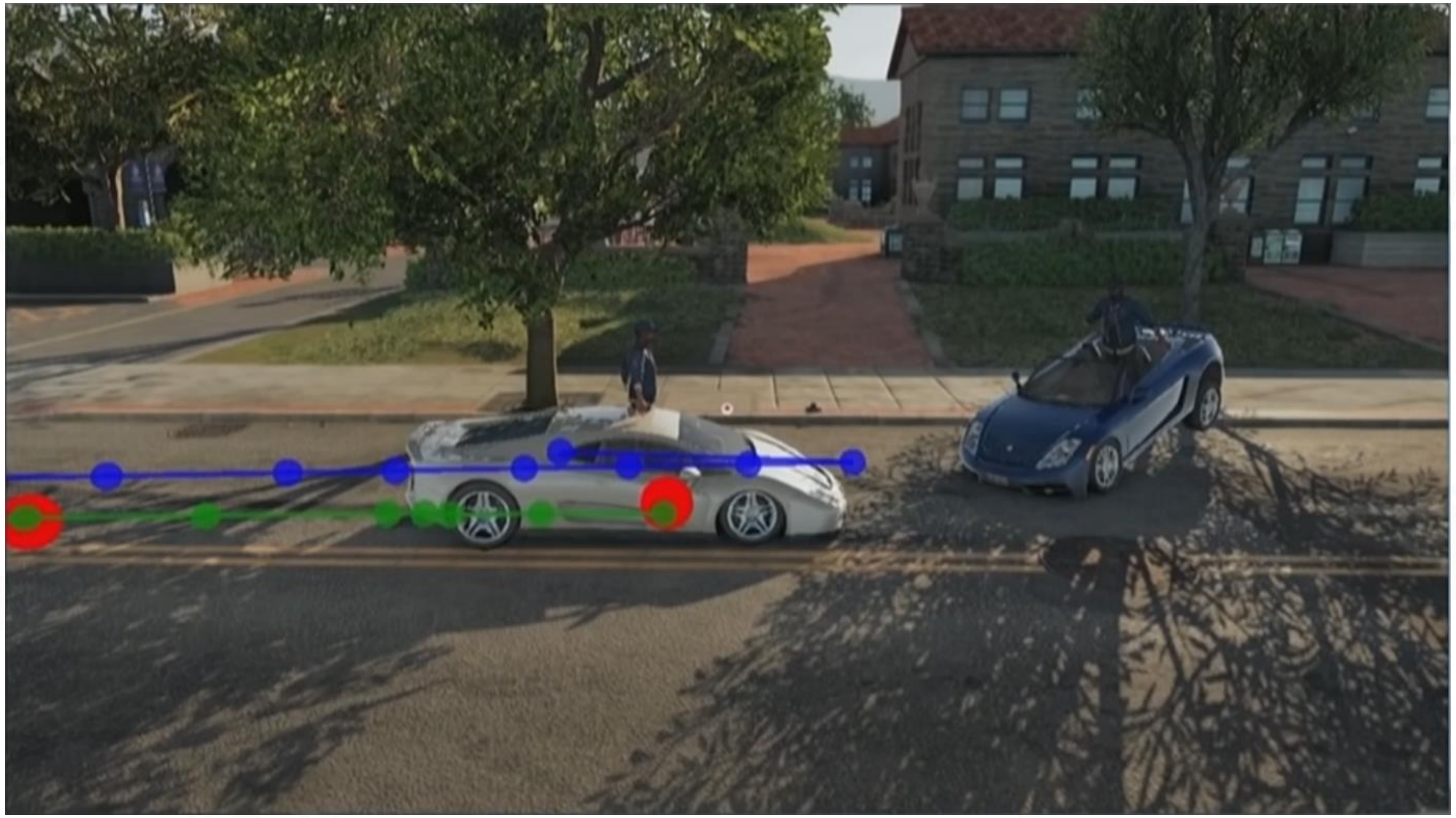
Phase 2: The replica keeps going, since the extrapolation is based on the last snaphsot



Red: Snapshot
Blue: Simulated track
Green: Ground truth

Collision Issues (4/4)

Phase 3: Finally we receive a snapshot to stop the replica, but replica gives master’s rigidbody a huge velocity to pushing master away

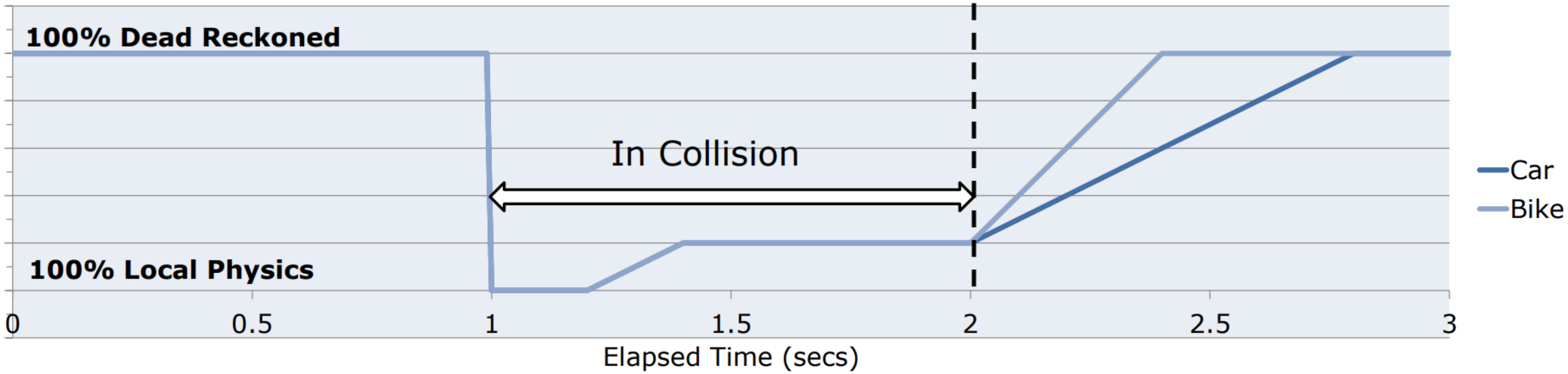


Red: Snapshot
Blue: Simulated track
Green: Ground truth

Physics Simualtion Blending During Collision

Tunable between two states

- State calculated by the client physics simulation
- State that tries to reach the dead reckoned positions



Tuned blending factors from Watch Dogs 2, Ubisoft Toronto. Bikes recover faster than cars

Usage Scenario of Interpolation

Scenario for Using Interpolation

- Characters' movement are very non-deterministic with high acceleration
- Gameplay suffers from the “wrap” when extrapolation errors occur

Typical examples

- FPS
- MOBA

Apex Legends

Usage Scenario of Extrapolation

Scenario for Using Extrapolation

- Player movement uses a realistic physical model
- Gameplay suffers from latency due to network transmission

Typical examples

- Racing game. Vehicle systems (Tanks, Ships, etc.)

World Of Warships

Blend Scenario of Interpolation and Extrapolation

Sometimes we need to apply both interpolation and extrapolation for the game to work properly

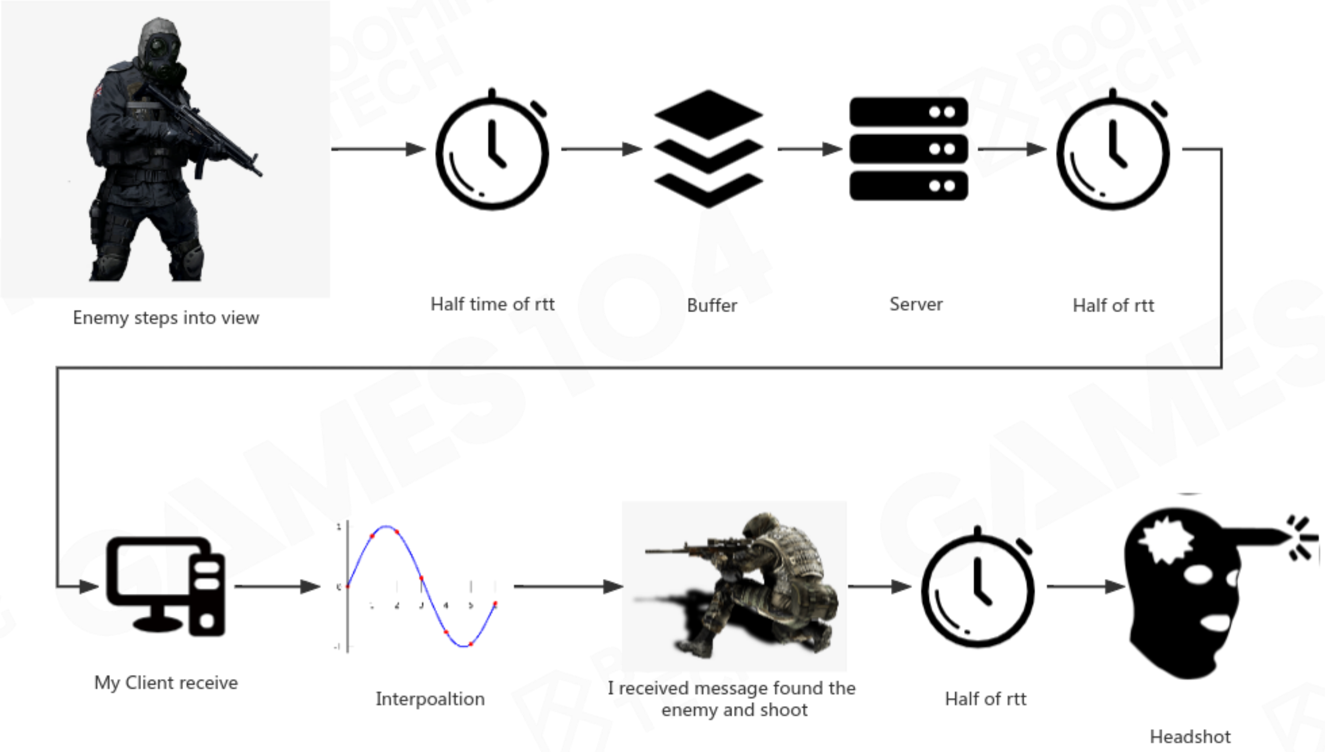
- Apply Extrapolation on vehicles
- Apply Interpolation for characters
- Do extrapolation if not enough data received

Battlefield1

Hit Registration

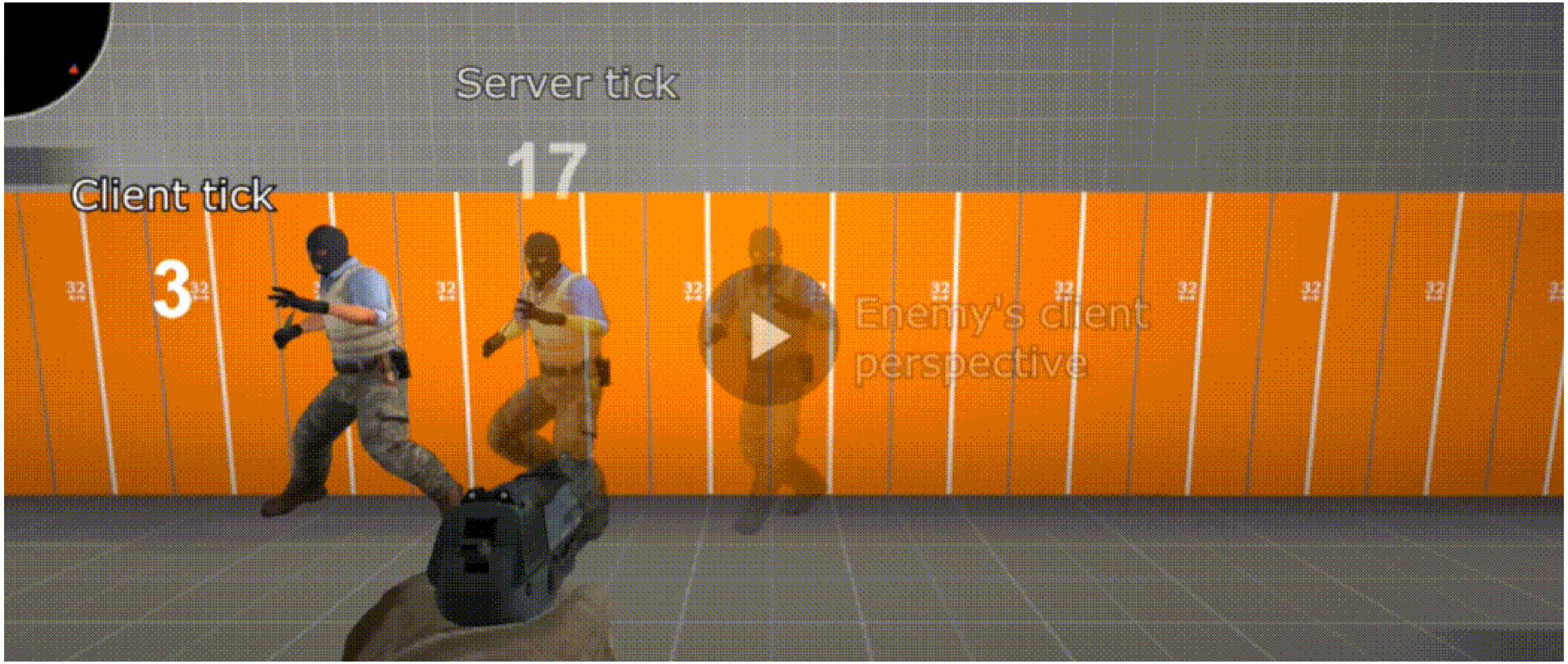
How to Make a Headshot in Online Game

Net messages to travel from client to server, and interpolation causes you to see the enemy way lag behind

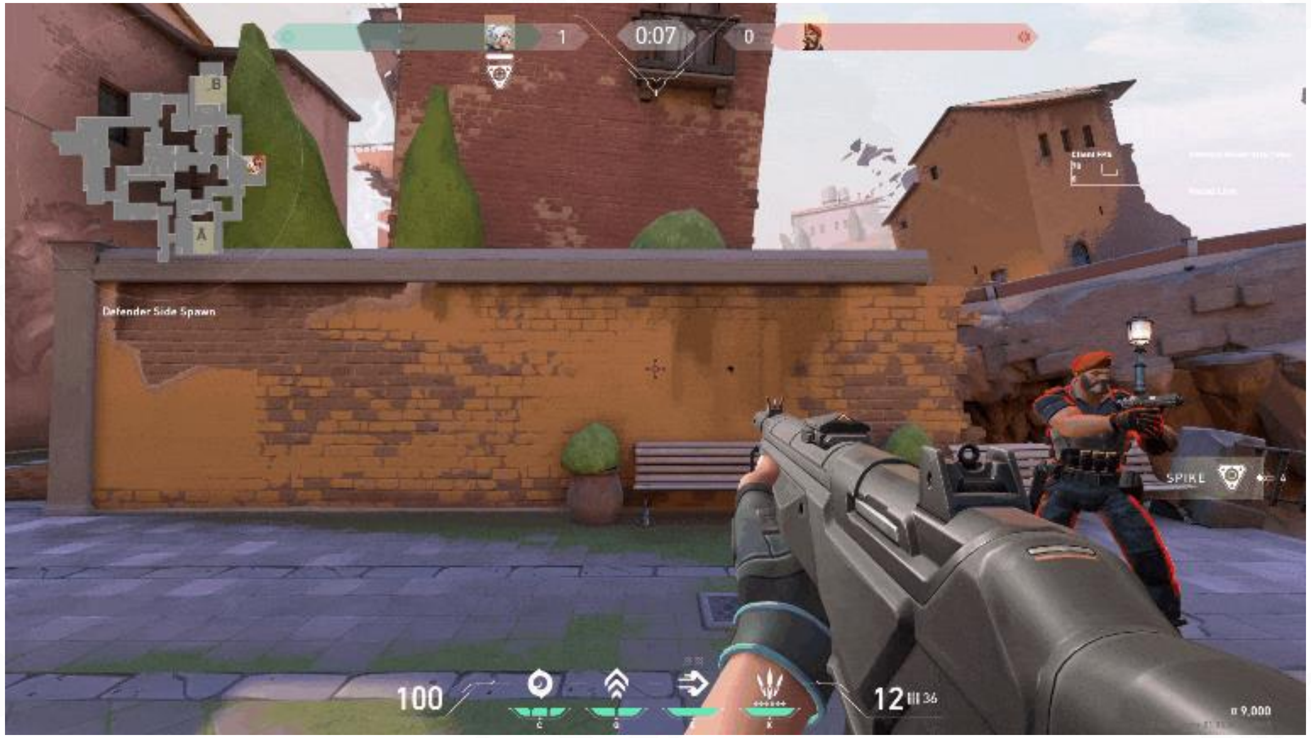


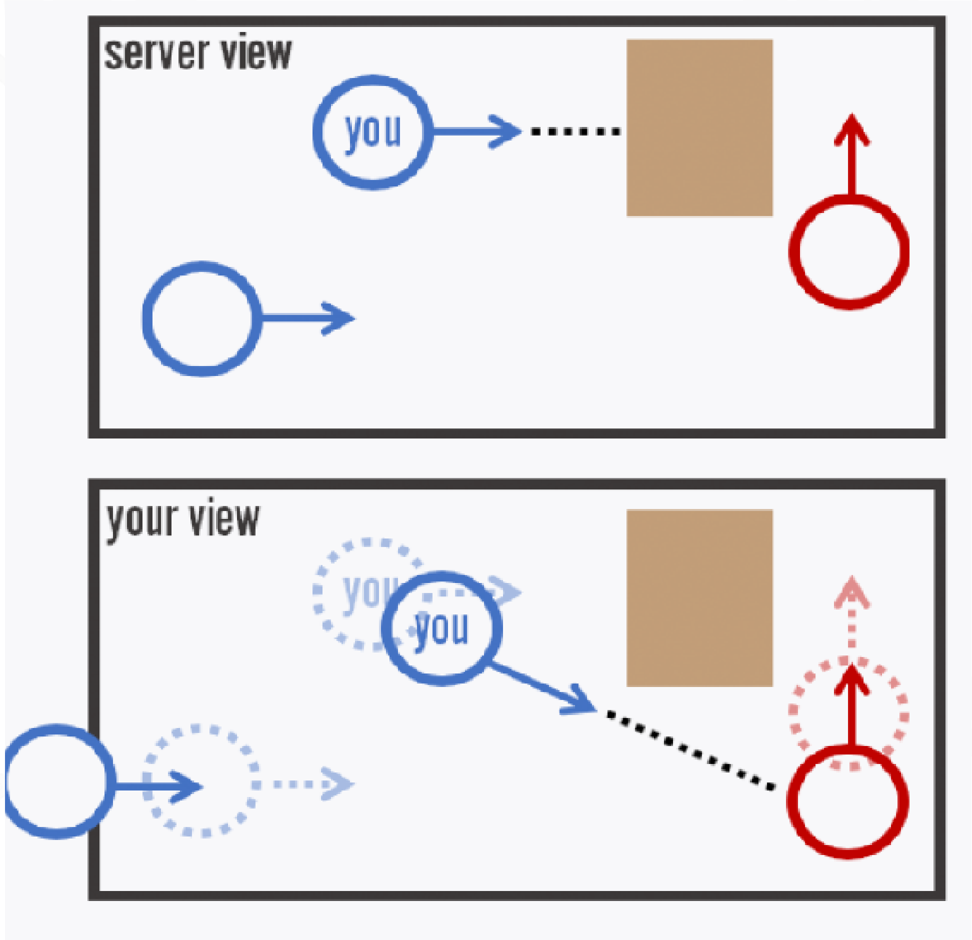
Where is the Enemy?

Due to latency, interpolation offset and time delay, you'll see other players slightly behind their current server positions. Where should I shot?



Where Should I Shot?





Hit Registration

Hit registration is making a **consensus** of all players that whether you've actually hit your enemy

Battlefield 3: **Client-side hit detection**

CSGO: **Server-side hit-registration**

Hit Registration

- Detecting hit event on client-side with replicated character positions
- Send hit events to the server
- Server run simple verification

The large map and lots of players

PUBG

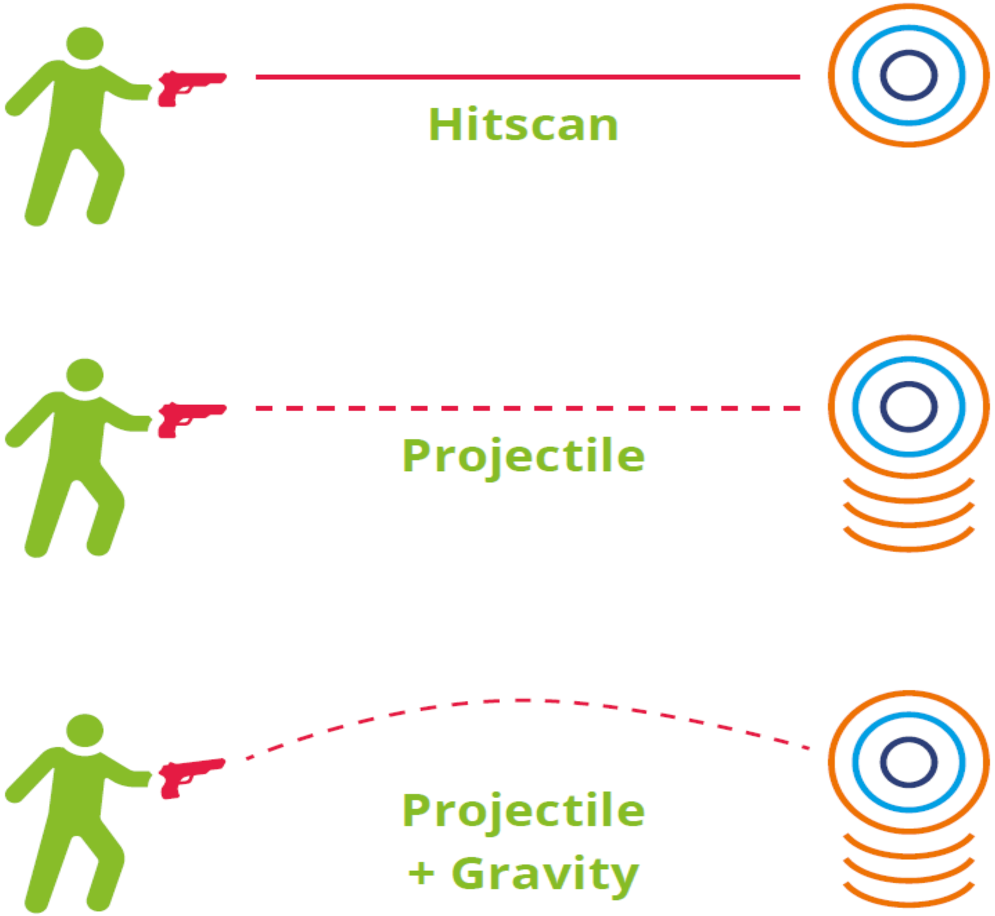
Destruction and Vehicles

Battlefield 3

A Comparison of Hitscan Weapons versus Projectile Weapons

Unlike hitscan weapons, projectile weapons can also simulate the effect of gravity

The scenery in Battlefield is built from several hitboxes, so destruction can take away the walls, the floors, etc.

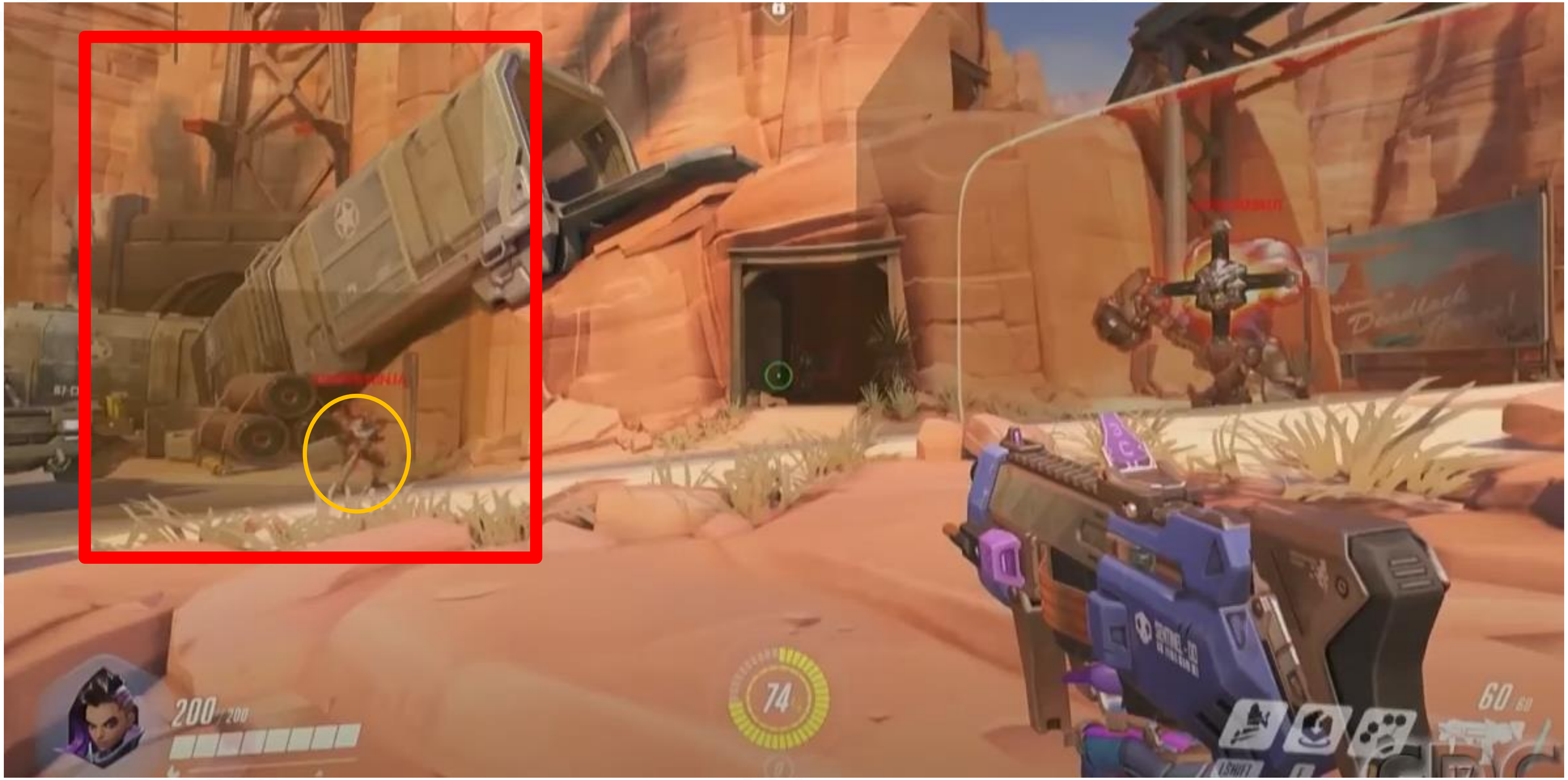


A Very Simple Server Verification of Hit Event

- Client send hit event with complete ray information to server
  - StartPoint, HitPoint and HitObject of the Raycast
- Validate StartPoint whether is really close enough to shooter
- Validate the HitPoint whether is really belong to HitOject
- Ensure nothing is blocking along the path by casting a ray from the StartPoint and HitPoint

In real game, the server verification is VERY TRICKY AND COMPLICATED

Server Verification Has to Guess



Problem of Client-Side Hit Detection

Efficient and Precise

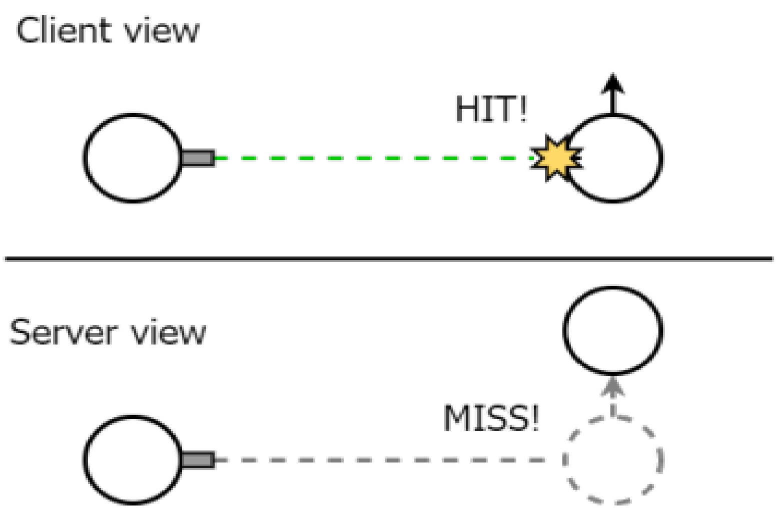
- Very efficient for hit detection without huge server workload
- Best shooting experience with pixel precision

Unsafe for cheating

- Fake hit event message
- Lag switches
- Infinite ammo …

Detecting Hit on Server-Side?

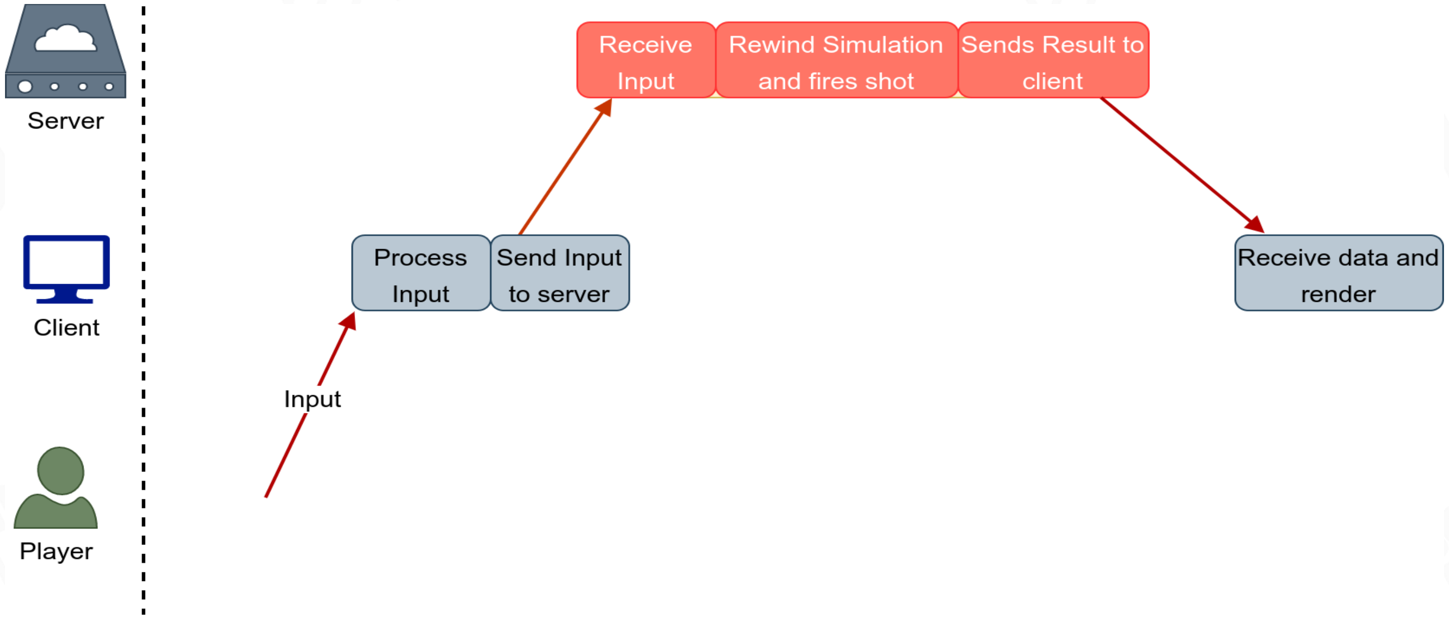
Client doesn’t know the target current location on server



Lag Compensation

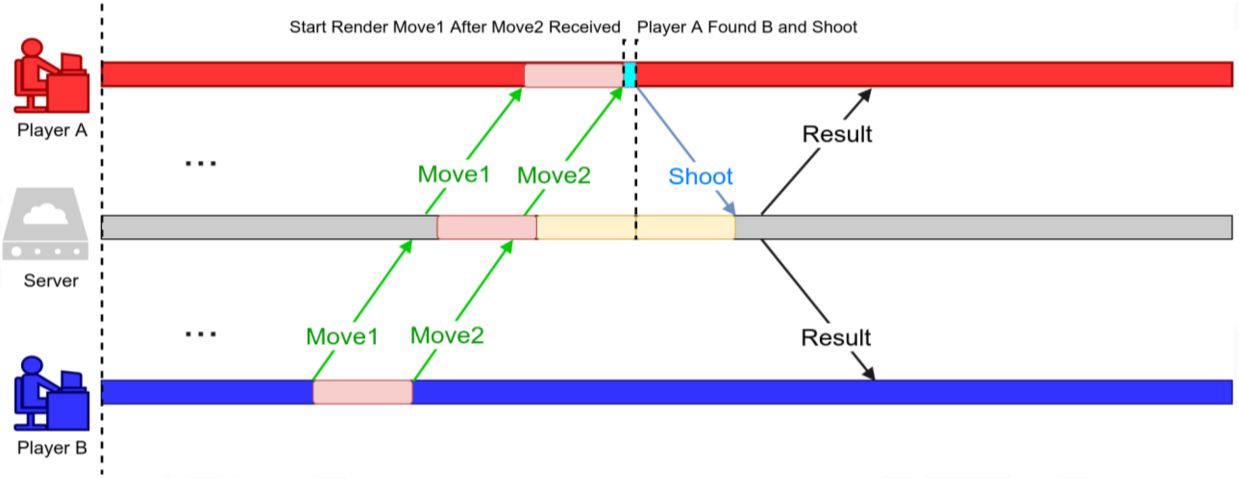
Server-side state rewinding to compensate network lags when player’s commands are executed

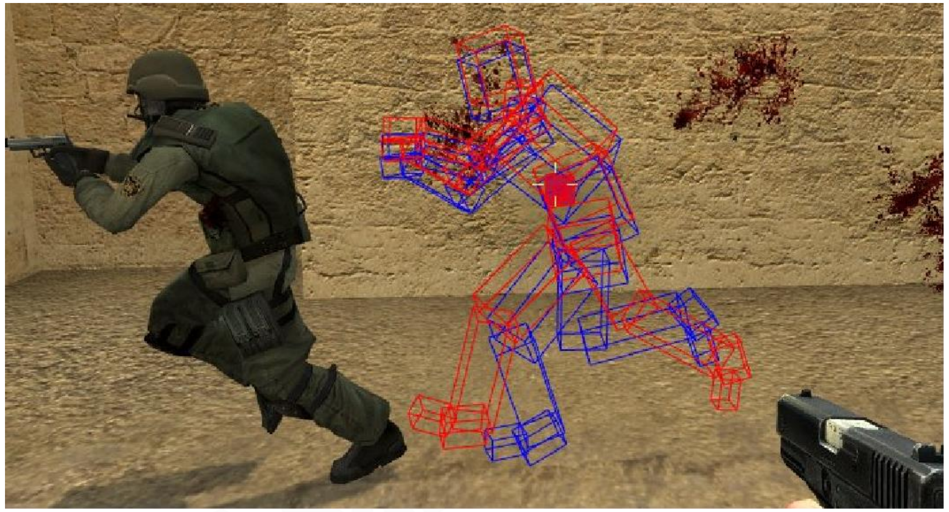
- Get information from clients
- Rewind game state in cached state snapshots that matches the client's action time
- Run client operation in rewind game state



Compensate all Possible Lags

- **RewindTime = Current Server Time - Packet Latency - Client View Interpolation Offset**


Start Render Move1 After Move2 Received
Player A Found B and Shoot
Player A
Server
Player B
...
Move1
Move2
Shoot
Result
...
Move1
Move2
Result

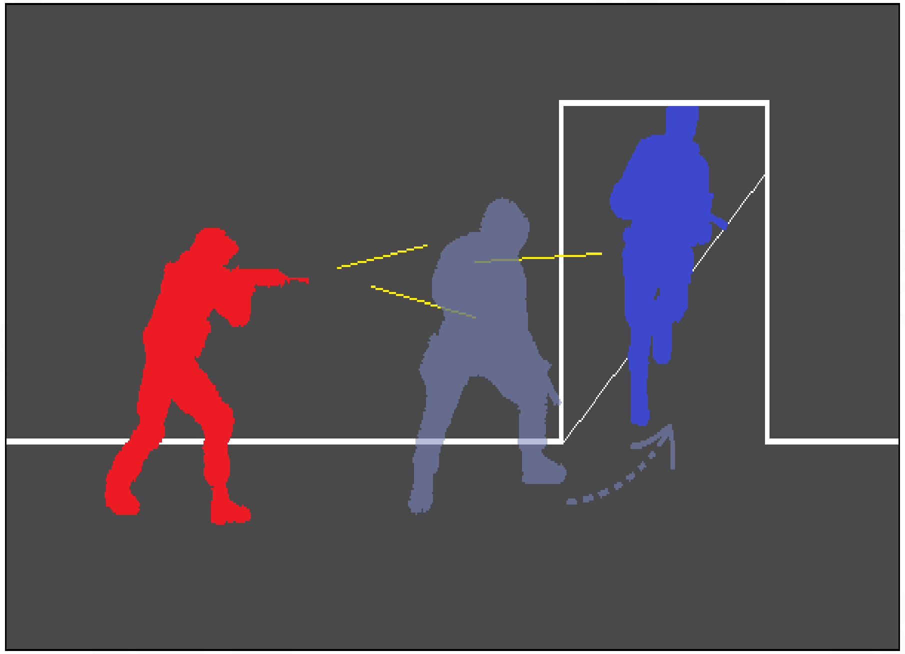


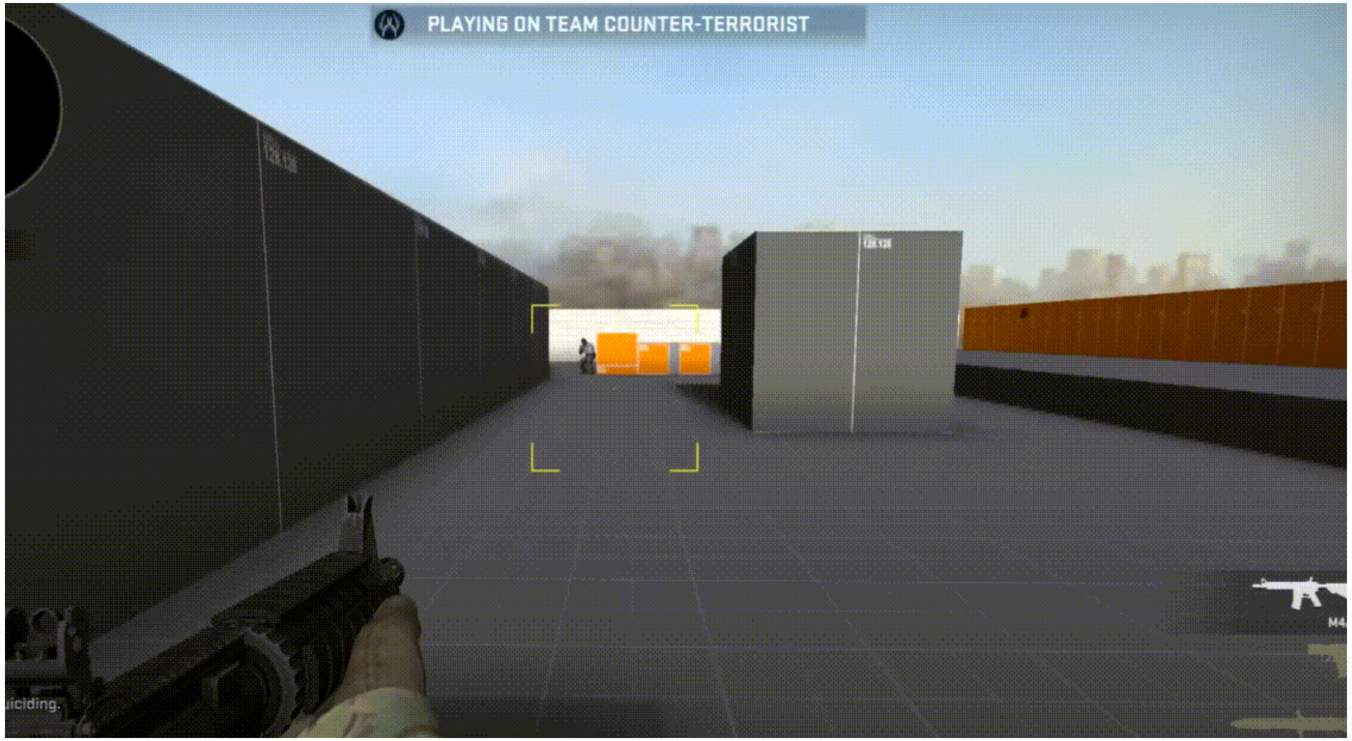
Actor: Enemy's client state

Red collision box: Enemy in the player's view

Blue collision box: Rewinded server state

Cover Problems – Running into Cover

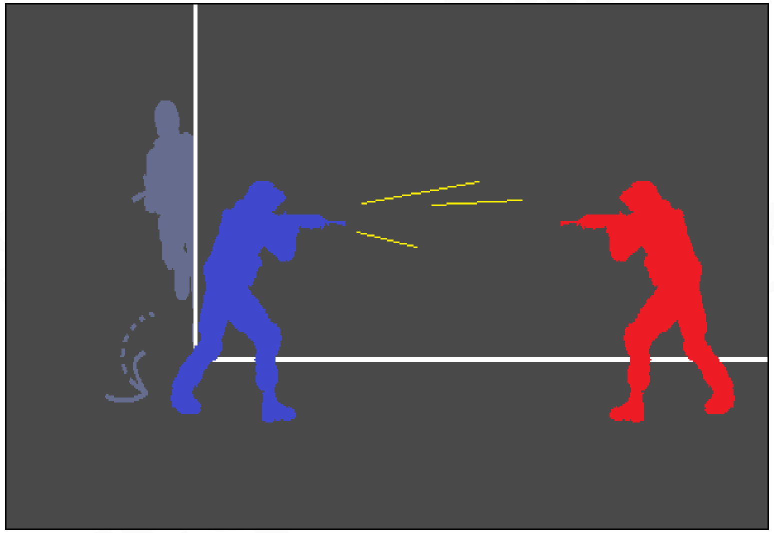


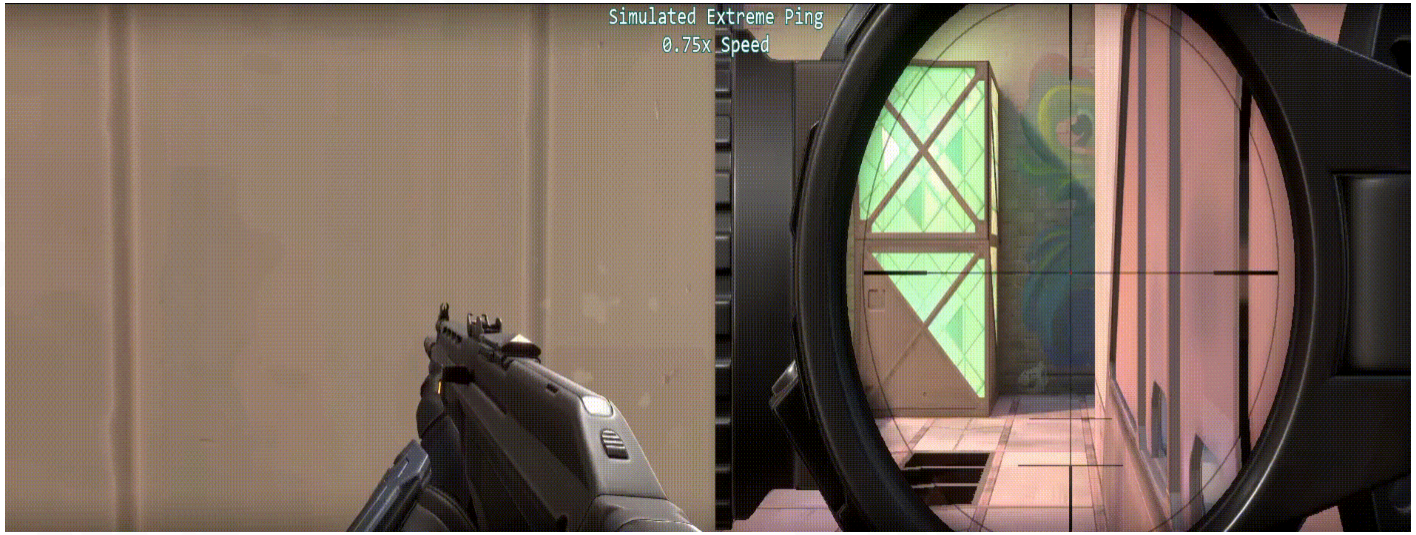


Shooter’s advantage

Cover Problems – Coming out from Cover

Peeker’s advantage



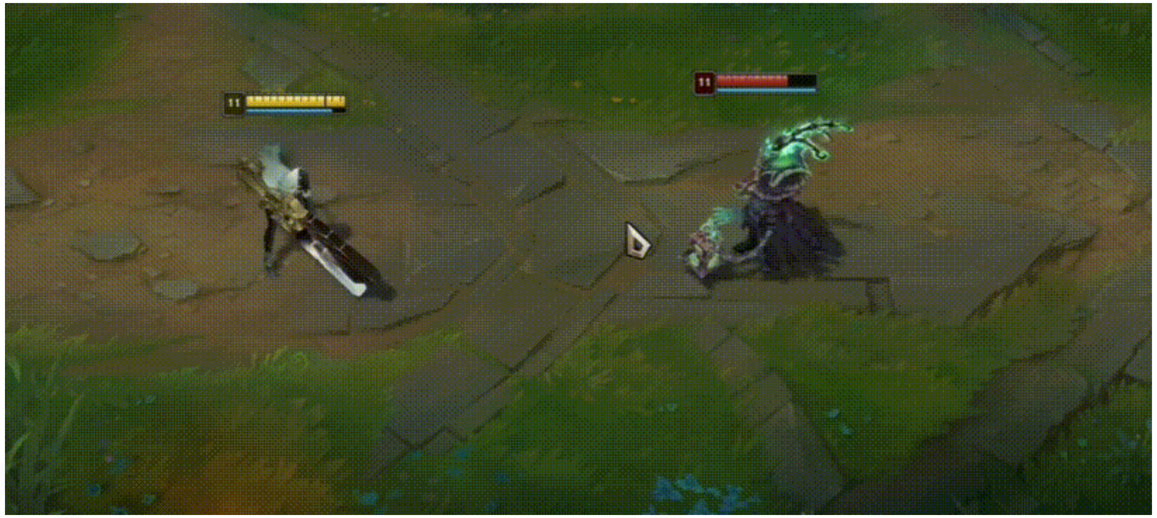


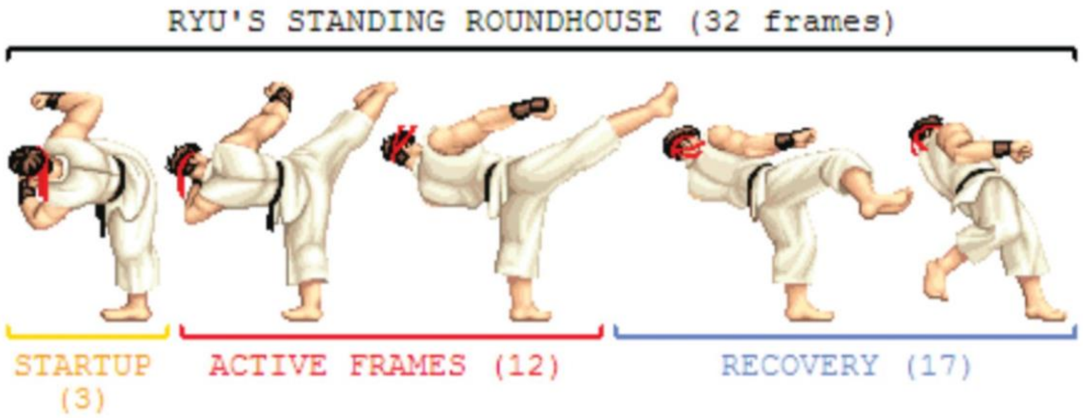
Peeker

Holder

Startup Frames to Ease Latency Feeling

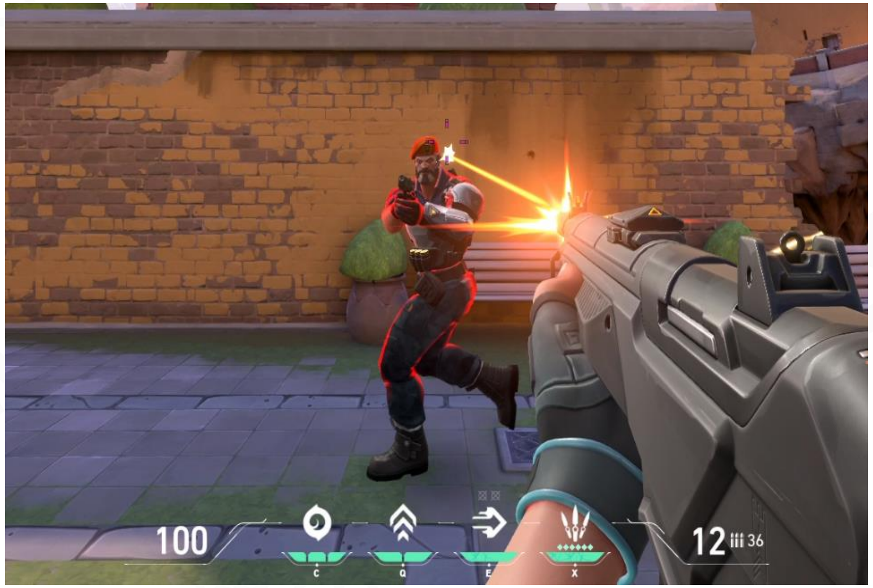
- A fixed animation before attack or move can also eliminate the effect of lag from network transmission
- Players will keep their attention on animations and ignore the state delay

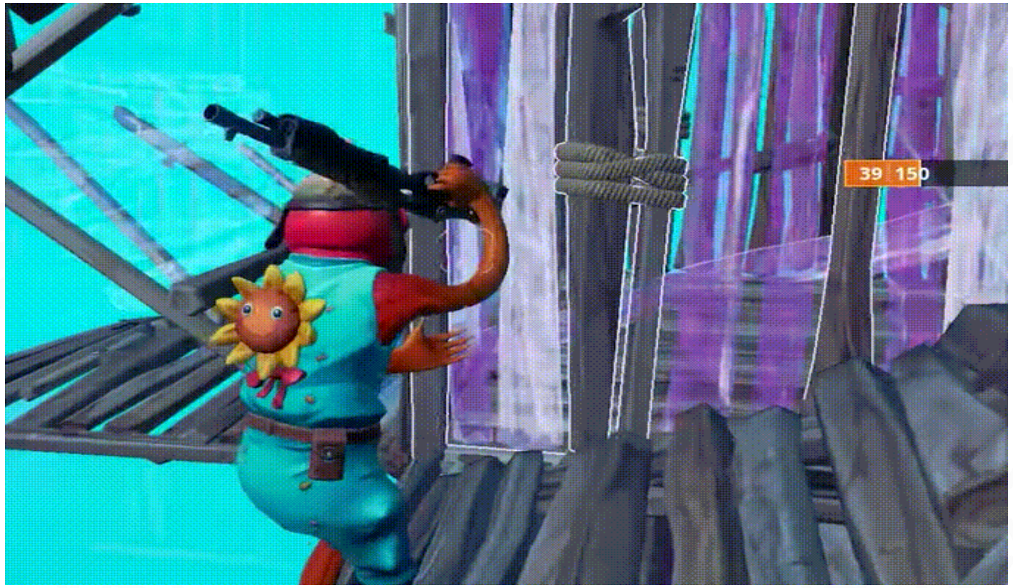




Local Forecast VFX Impacts

- Clients can perform local hit tests in order to give the player some instant feedback, such as displaying a blood splatter visual effect
- However, any permanent effects of the hits, such as reducing the hit points of a player, are only applied after receiving confirmation from the server





MMOG Network Architecture

What is MMOG?

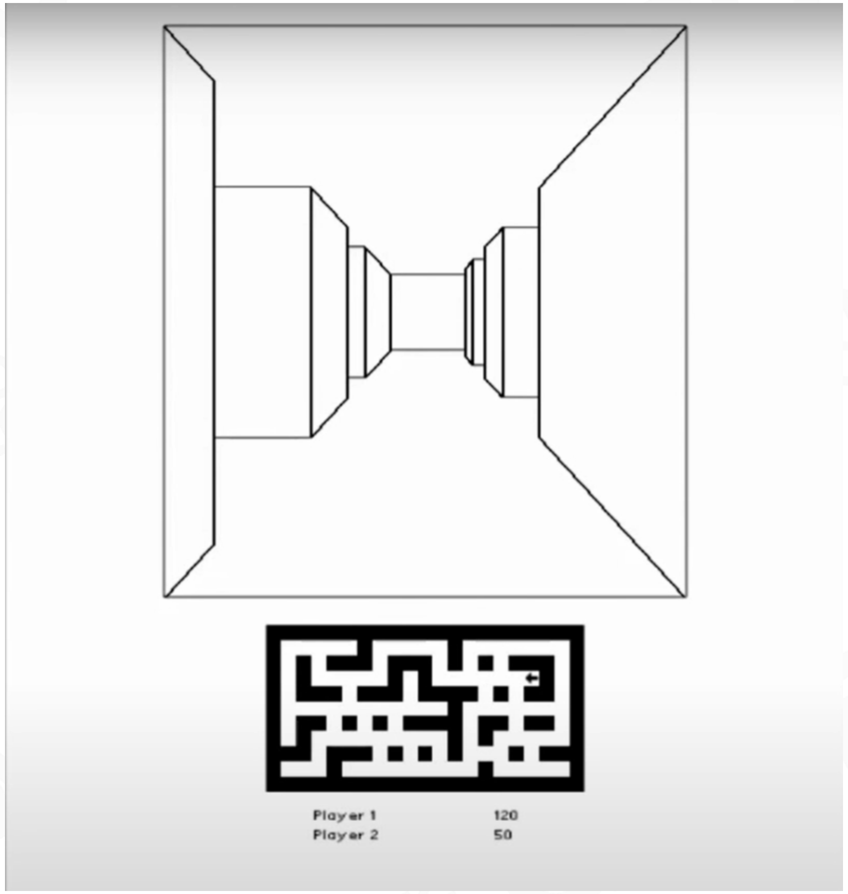
MMOG: Massively Multiplayer Online Game, or more commonly **MMO**

MMOs with a large numbers of players, often hundreds or thousands, on the same server can enable players to cooperate and compete with each other on a large scale, and include a variety of gameplay types (MMORPG, MMORTS, MMOFPS, etc.)

Final Fantasy XIV - MMORPG

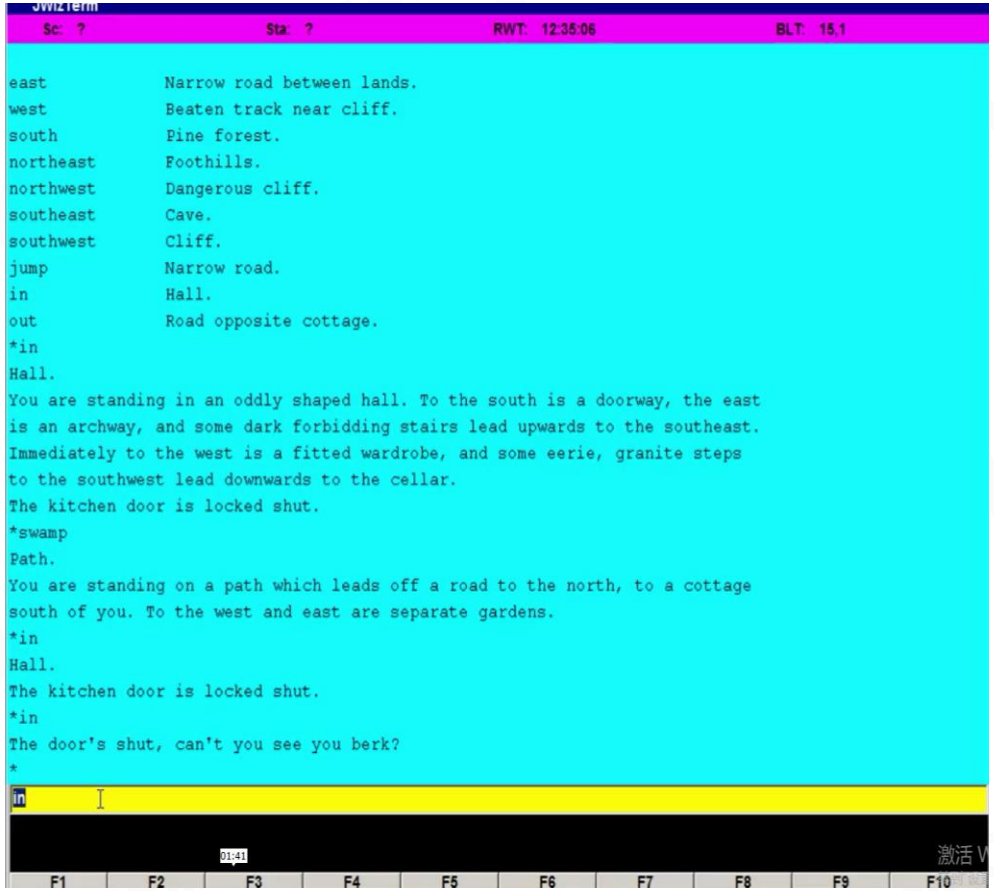
PlanetSide 2 - MMOFPS

The first Network Game



Mazewar in 1974

The first Role Playing Game



```
Sc:  ?                 Sta:  ?                 RWT:  12:35:06          BLT:  15.1

east            Narrow road between lands.
west            Beaten track near cliff.
south           Pine forest.
northeast       Foothills.
northwest       Dangerous cliff.
southeast       Cave.
southwest       Cliff.
jump            Narrow road.
in              Hall.
out             Road opposite cottage.
*in
Hall.
You are standing in an oddly shaped hall. To the south is a doorway, the east
is an archway, and some dark forbidding stairs lead upwards to the southeast.
Immediately to the west is a fitted wardrobe, and some eerie, granite steps
to the southwest lead downwards to the cellar.
The kitchen door is locked shut.
*swamp
Path.
You are standing on a path which leads off a road to the north, to a cottage
south of you. To the west and east are separate gardens.
*in
Hall.
The kitchen door is locked shut.
*in
The door's shut, can't you see you berk?
*
in
```

Multi-User Dimension in 1978

Diversities of Modern MMO

Version 4.3.4 (15595) (Release x64)
Apr 10 2012

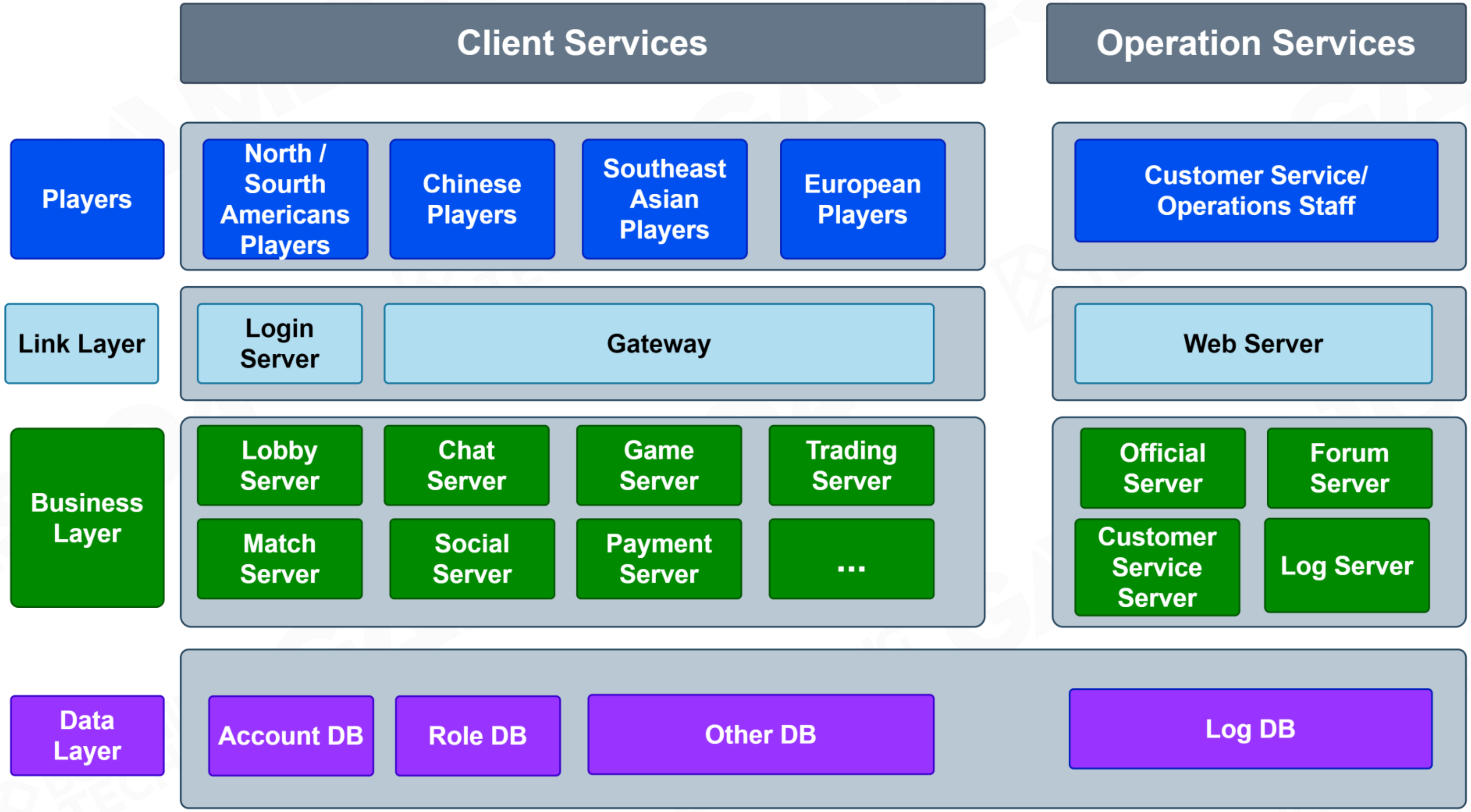
Copyright 2004-2012 Blizzard Entertainment. All Rights Reserved.

Game Sub-Systems

MMOs have a variety of gameplay and are supported by many sub-systems

- User management
- Matchmaking
- Trading system
- Social system
- Data storage
- ...

MMO Architecture



Client Services

Operation Services

Players

North / Sourth Americans Players

Chinese Players

Southeast Asian Players

European Players

Customer Service/ Operations Staff

Link Layer

Login Server

Gateway

Web Server

Business Layer

Lobby Server

Chat Server

Game Server

Trading Server

Match Server

Social Server

Payment Server

...

Official Server

Forum Server

Customer Service Server

Log Server

Data Layer

Account DB

Role DB

Other DB

Log DB

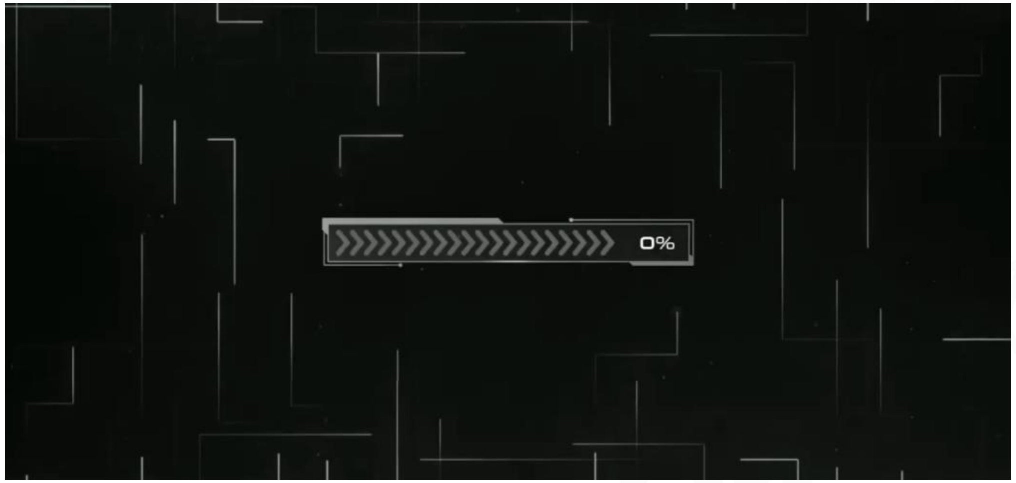
Services of Link Layer

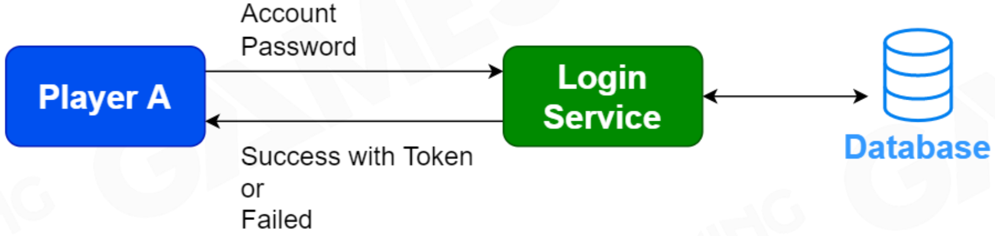
Login Server

- Verification of client connection

Gateway

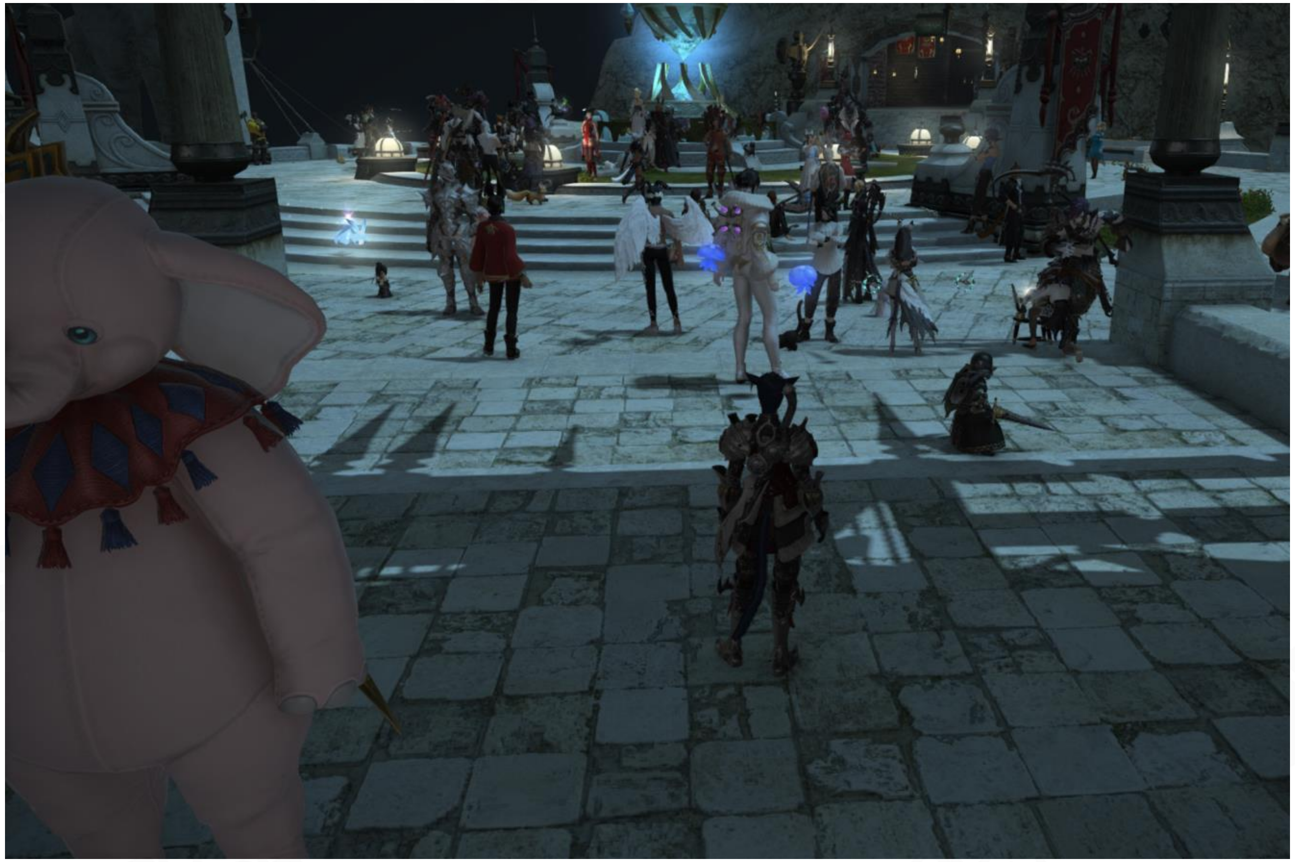
- Very important layer to separate inside/outside networks





Lobby

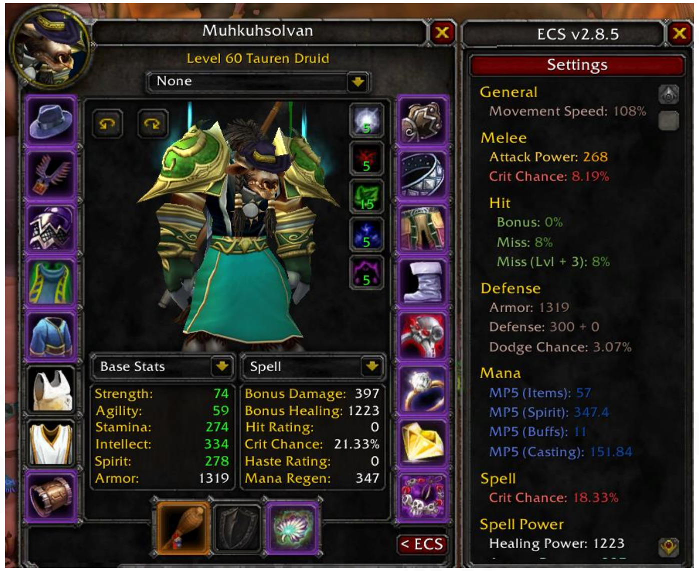
- Players can gather in the lobby, see and interact with other players
- When the number of players continues to increase, it is a challenge to the performance of the server and the client



Final Fantasy XIV

Character Server

All player data is managed in one system. Such as account info, character info, backpack info, mail info, etc.



Muhkuhsolvan

Level 60 Tauren Druid

None

Base Stats

Strength: 74
Agility: 59
Stamina: 274
Intellect: 334
Spirit: 278
Armor: 1319

Spell

Bonus Damage: 397
Bonus Healing: 1223
Hit Rating: 0
Crit Chance: 21.33%
Haste Rating: 0
Mana Regen: 347

< ECS

ECS v2.8.5

Settings

General
Movement Speed: 108%

Melee
Attack Power: 268
Crit Chance: 8.19%

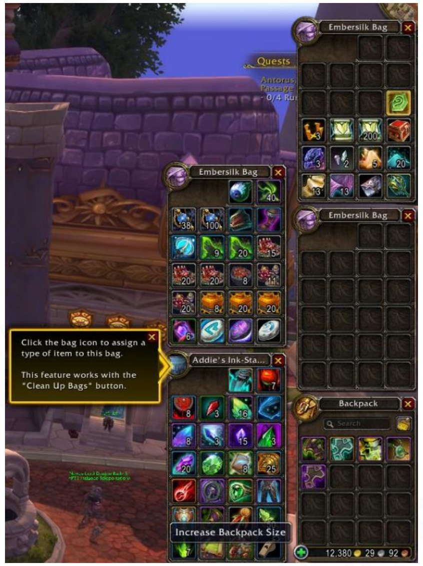
Hit
Bonus: 0%
Miss: 8%
Miss (Lvl + 3): 8%

Defense
Armor: 1319
Defense: 300 + 0
Dodge Chance: 3.07%

Mana
MP5 (Items): 57
MP5 (Spirit): 347.4
MP5 (Buffs): 11
MP5 (Casting): 151.84

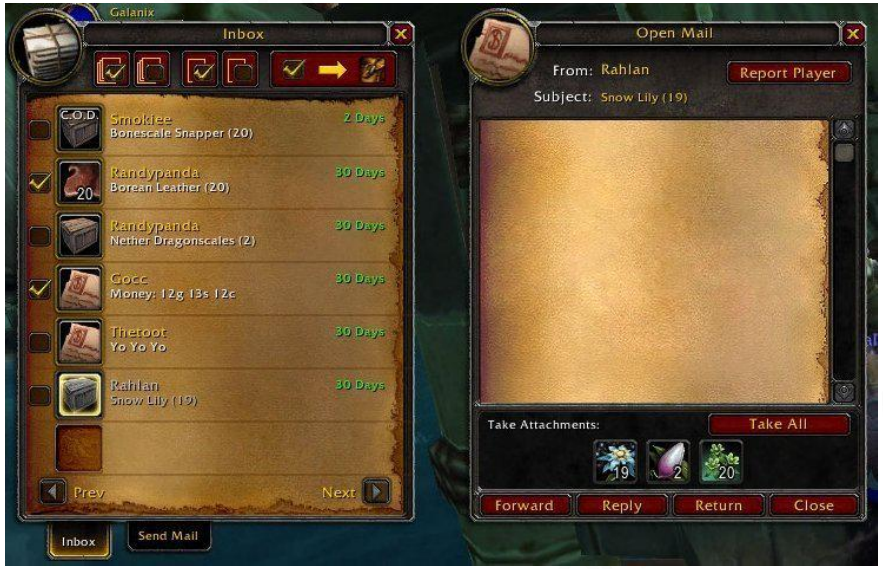
Spell
Crit Chance: 18.33%

Spell Power
Healing Power: 1223



Click the bag icon to assign a type of item to this bag.

This feature works with the "Clean Up Bags" button.



Galanix

Inbox

C.O.D. Smokiee	Bonescale Snapper (20)	2 Days
Randypanda	Borean Leather (20)	30 Days
Randypanda	Nether Dragonscales (2)	30 Days
Gocc	Money: 12g 13s 12c	30 Days
Thetoot	Yo Yo Yo	30 Days
Rahlan	Snow Lily (19)	30 Days

Open Mail

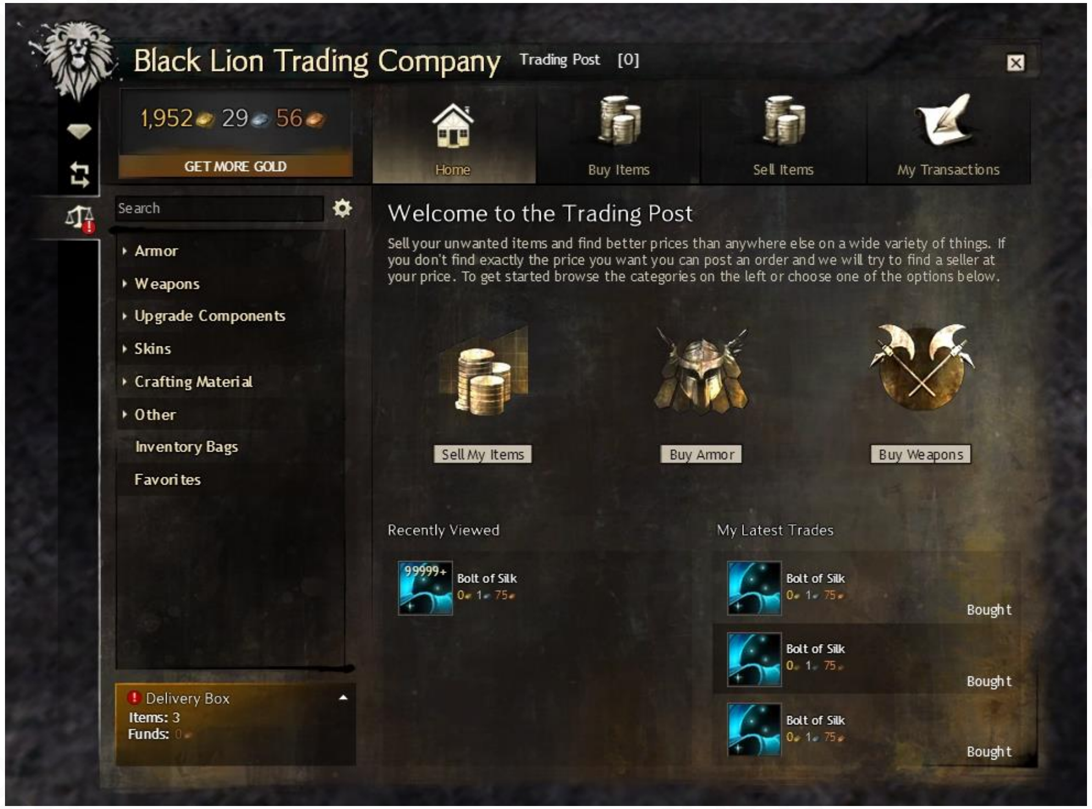
From: Rahlan

Subject: Snow Lily (19)

Take Attachments:

Trading System

- Buying and selling items on the marketplace
- Sending items or coins to other players through the in-game Mail
- Game designers need to keep an eye on market prices to prevent imbalances
- For a persistent world to maintain a stable economy, a balance must be struck between currency sources and sinks
- Players can use real-world money to buy a specific in-game item



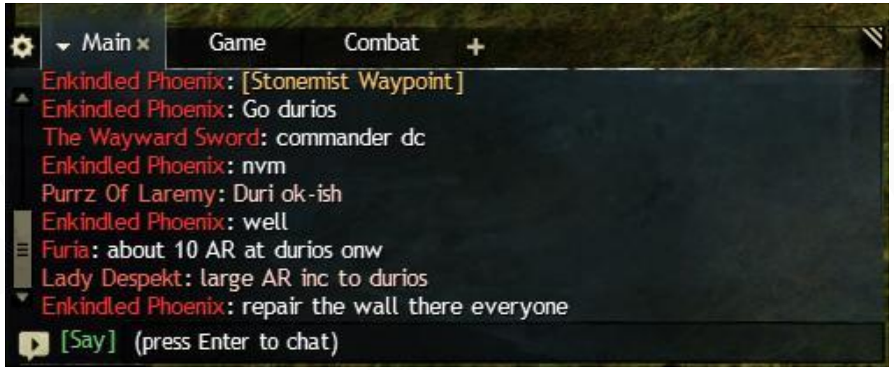
Welcome to the Trading Post

Sell your unwanted items and find better prices than anywhere else on a wide variety of things. If you don't find exactly the price you want you can post an order and we will try to find a seller at your price. To get started browse the categories on the left or choose one of the options below.

Trading System in *Guild Wars 2*

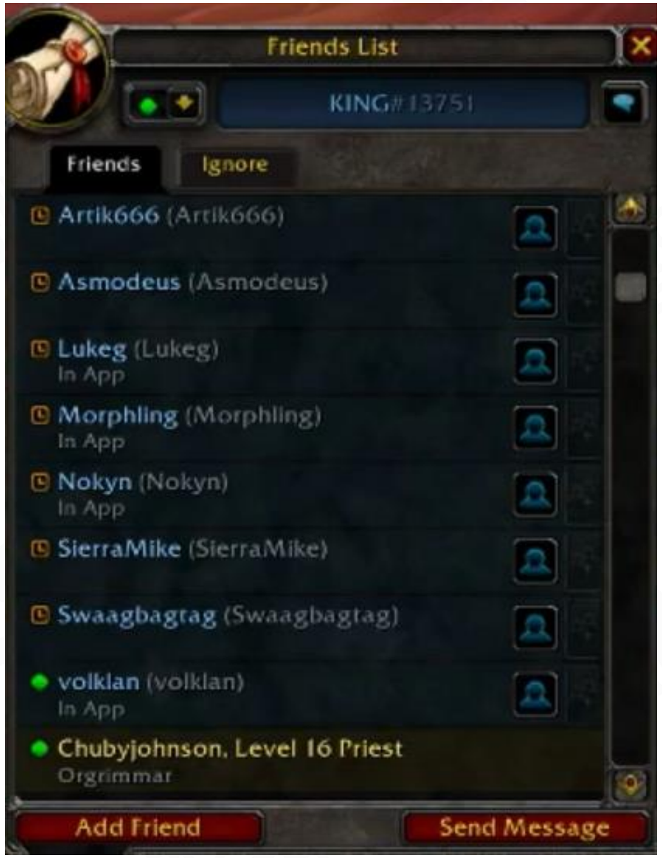
Social System

- Player-to-player interplay and communication
- Foster stronger social cohesion in-game



Enkindled Phoenix: [Stonemist Waypoint]
Enkindled Phoenix: Go durios
The Wayward Sword: commander dc
Enkindled Phoenix: nvm
Purrz Of Laremy: Duri ok-ish
Enkindled Phoenix: well
Furia: about 10 AR at durios onw
Lady Despekt: large AR inc to durios
Enkindled Phoenix: repair the wall there everyone
[Say] (press Enter to chat)

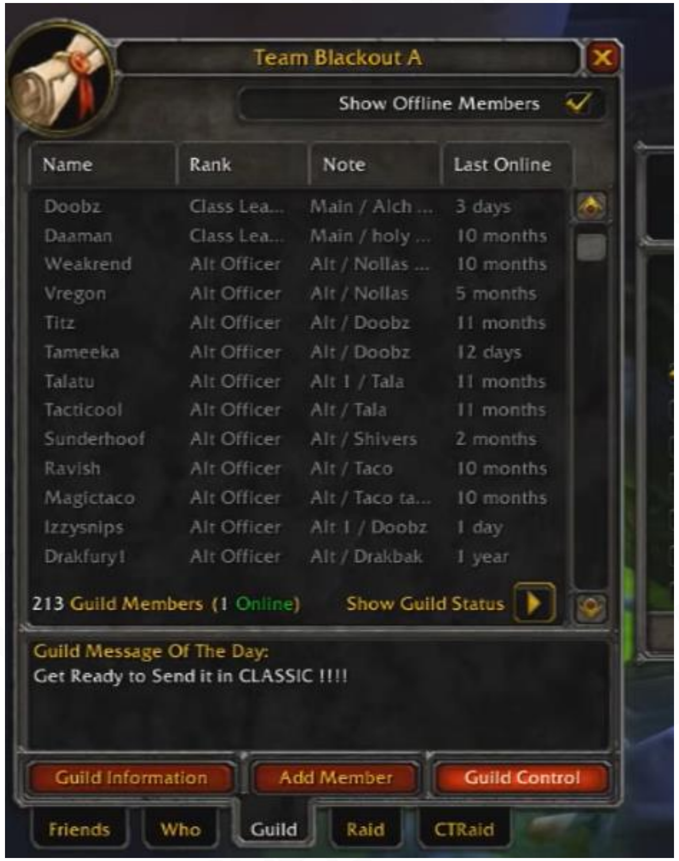
Multiple Chat Channels



KING#13751

Artik666 (Artik666)
Asmodeus (Asmodeus)
Lukeg (Lukeg)
In App
Morphling (Morphling)
In App
Nokyn (Nokyn)
In App
SierraMike (SierraMike)
Swaagbagtag (Swaagbagtag)
volklan (volklan)
In App
Chubyjohnson, Level 16 Priest
Orgrimmar

Friends List



Name	Rank	Note	Last Online
Doobz	Class Lea...	Main / Alch ...	3 days
Daaman	Class Lea...	Main / holy ...	10 months
Weakrend	Alt Officer	Alt / Nollas ...	10 months
Vregon	Alt Officer	Alt / Nollas	5 months
Titz	Alt Officer	Alt / Doobz	11 months
Tameeka	Alt Officer	Alt / Doobz	12 days
Talatu	Alt Officer	Alt 1 / Tala	11 months
Tacticool	Alt Officer	Alt / Tala	11 months
Sunderhoof	Alt Officer	Alt / Shivers	2 months
Ravish	Alt Officer	Alt / Taco	10 months
Magictaco	Alt Officer	Alt / Taco ta...	10 months
Izzysnips	Alt Officer	Alt 1 / Doobz	1 day
Drakfury1	Alt Officer	Alt / Drakbak	1 year

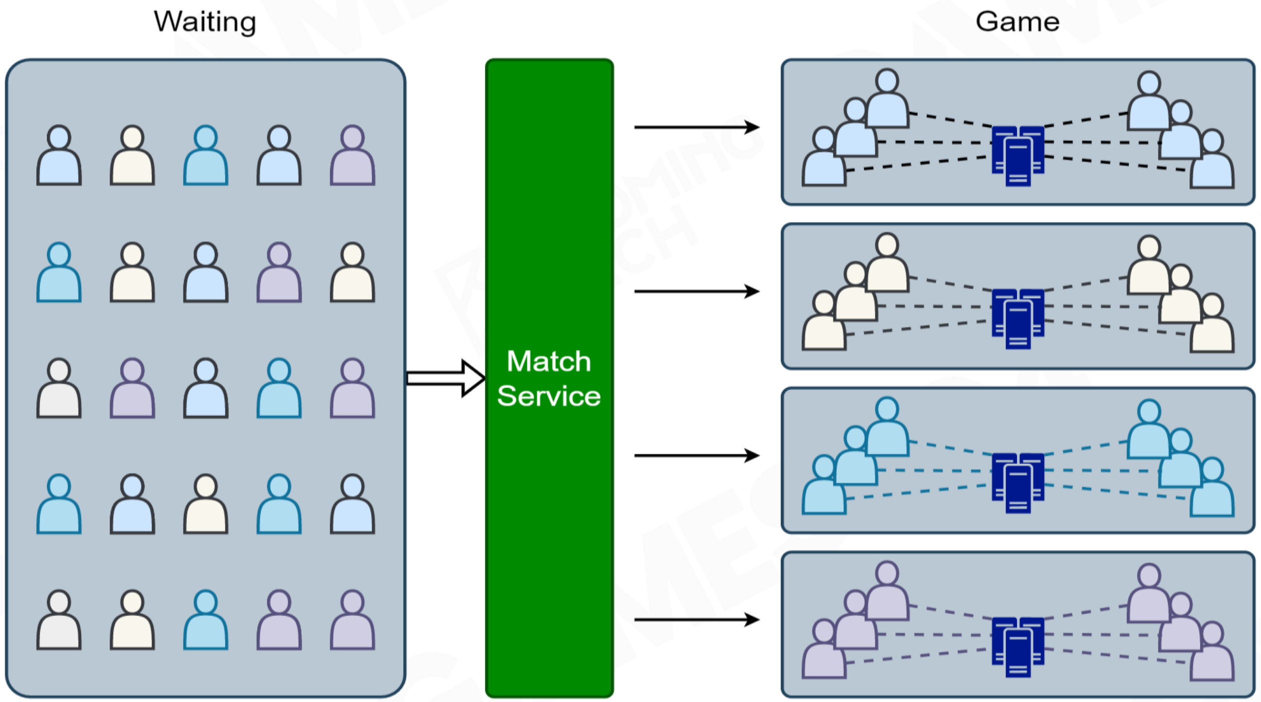
213 Guild Members (1 Online)

Guild Message Of The Day:
Get Ready to Send it in CLASSIC !!!!

Guild

Matchmaking

- You have to consider attributes like skills, level, latency, wait time...
- In general, making a good matchmaking service is core for a game design
- Running this on a global scale for your player population presents a whole different set of challenges



Data Storage

The game data is very complex and diverse

- Player data (guilds, dungeons, warehouse, etc.)
- Monitoring data
- Mining data
- ...
- Data needs to be securely persisted and efficiently organized for retrieved and analysis etc.

Choices of Database

Relational Data Storage

- Requires Structure to be Predetermined
- Flexible Queries
- Always Consistent

Game Development Examples

- Player Data
- Game Data
- Inventory
- Item Shops/Trading

Non-Relational Data Storage

- Structure Can Change For Each Entry
- Queries Have Higher Specificity
- May Not Always Be Consistent

Game Development Examples

- Player/Item Stats/Profile Game Data
- Enchantments and Upgrades
- Game States
- Quest Data

mongoDB

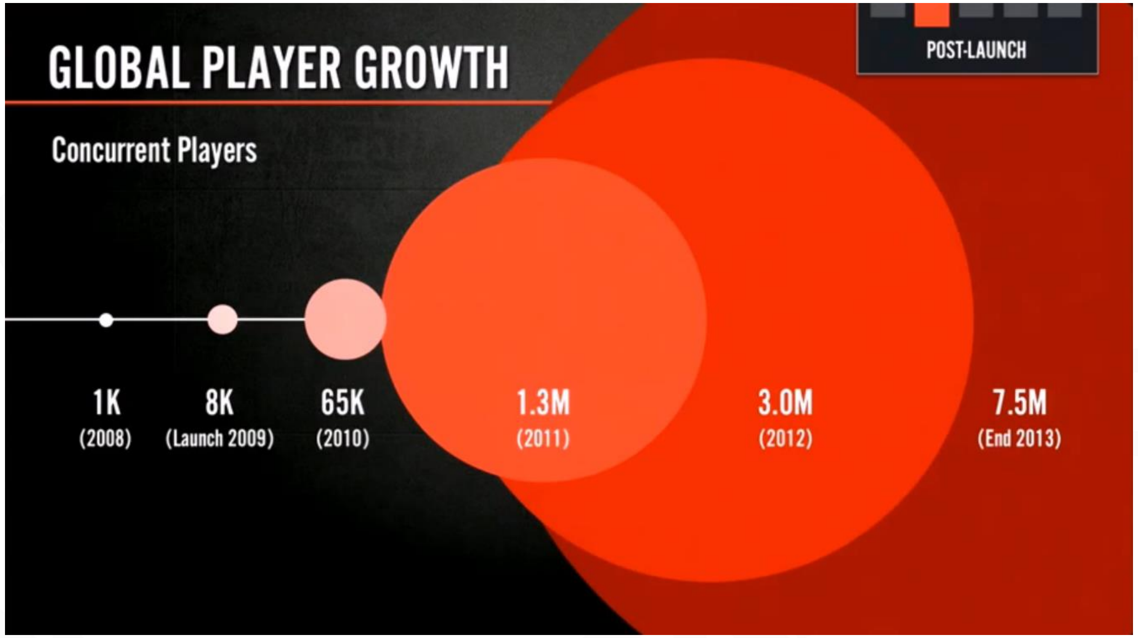
In-Memory Data Storage

- Extremely Fast (Memory versus Hard Disk)
- Key-Value
- Fast Sorted/Ranged Searches
- Persistence among servers

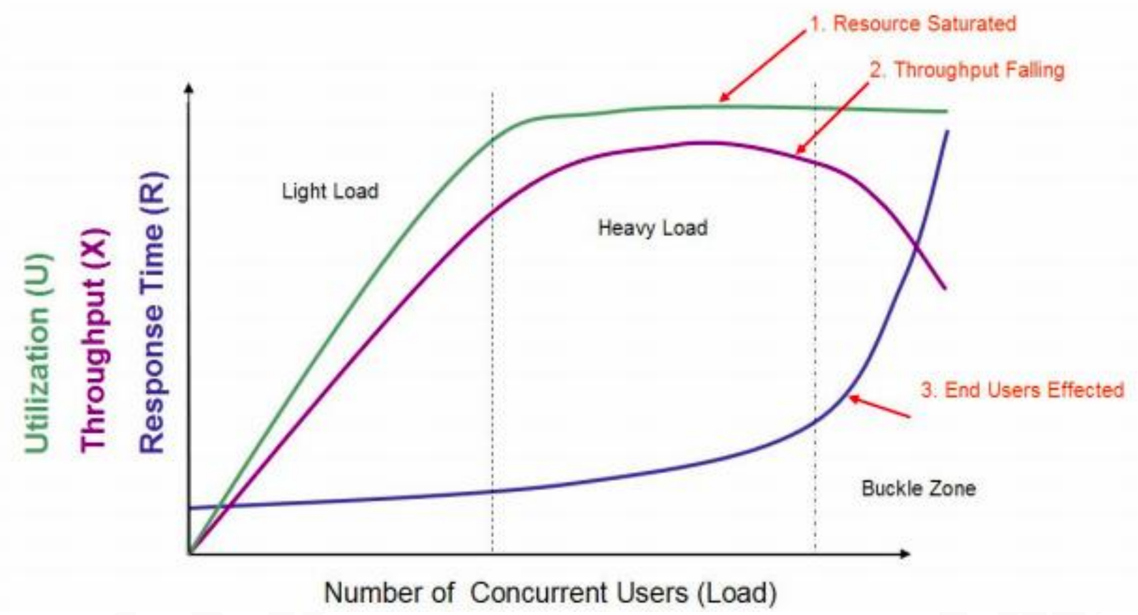
Game Development Examples

- Matchmaking
- Leaderboards
- Session Management
- Boost Performance For Other Databases

Player Number Growth



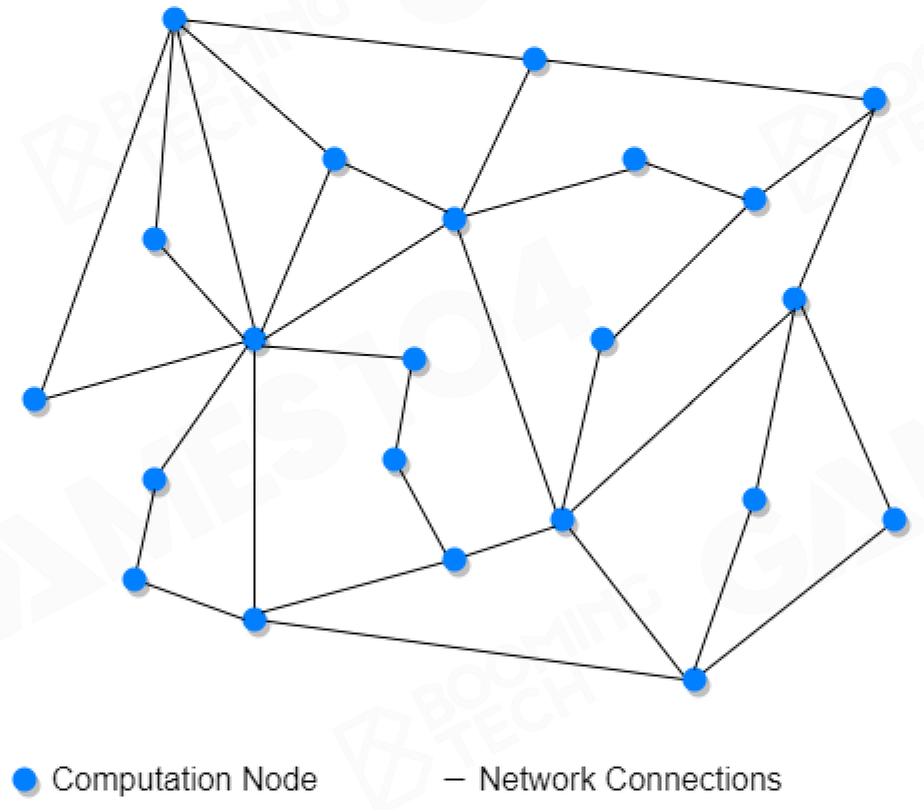
Global player growth in *LOL*



The relationship between user load, service request response time, and resource utilization

Distributed System

A distributed system is a computing environment in which various components are spread across multiple computers (or other computing devices) on a network



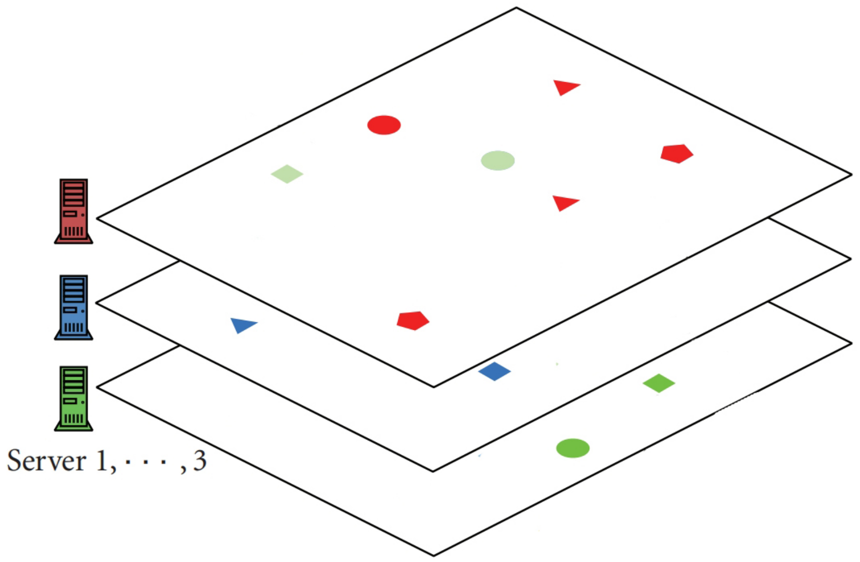
Challenges with Distributed systems

- Data access mutual exclusion
- Idempotence
- Failure and partial failure
- Unreliable network
- Distributed bugs spread epidemically
- Consistency and consensus
- Distributed transaction

Load Balancing

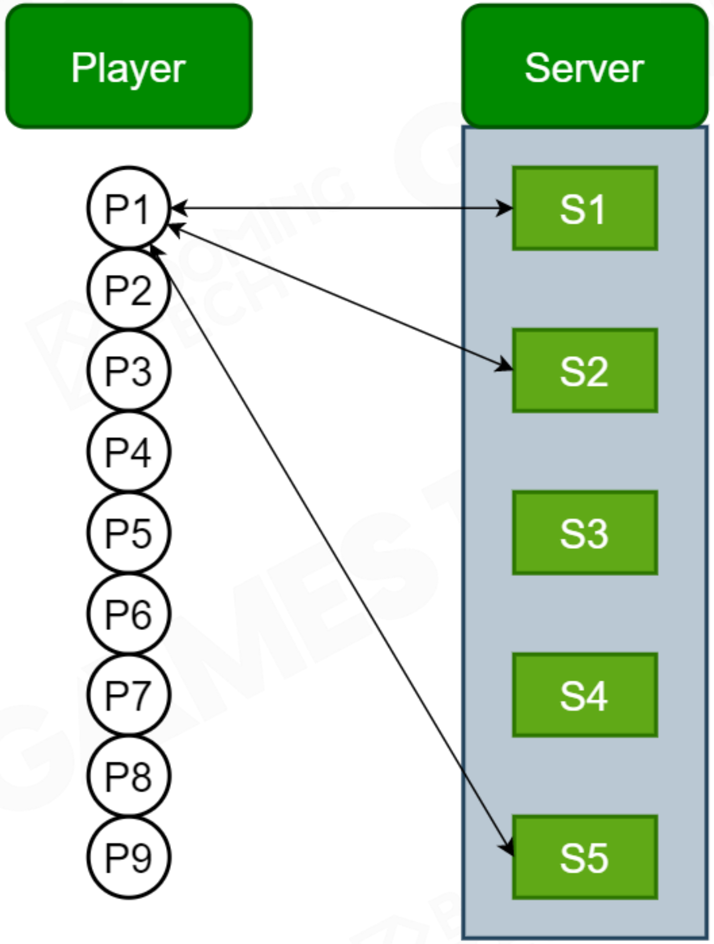
Refers to the process of distributing a set of tasks over a set of resources (computing units), with the aim of making their overall processing more efficient

- Optimize the response time
- Avoid unevenly overloading some compute nodes while other compute nodes are left idle
- All players are evenly divided on multiple servers

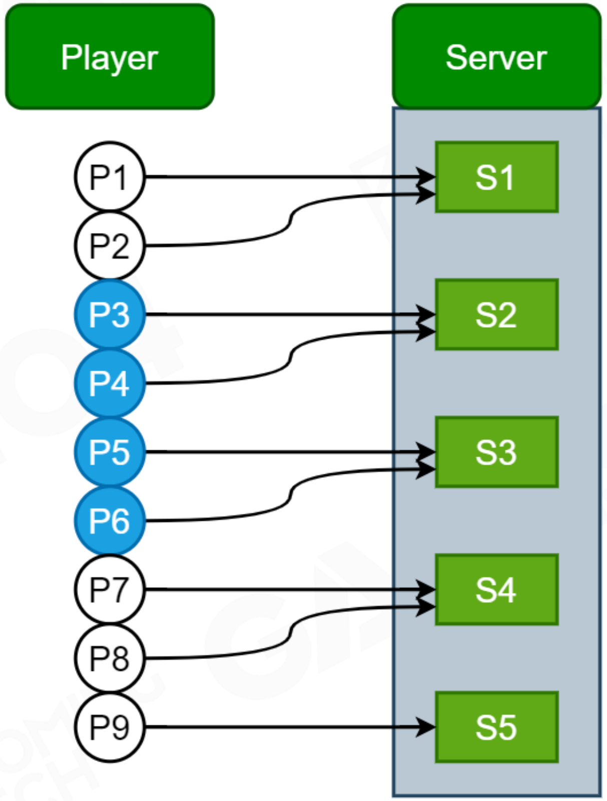


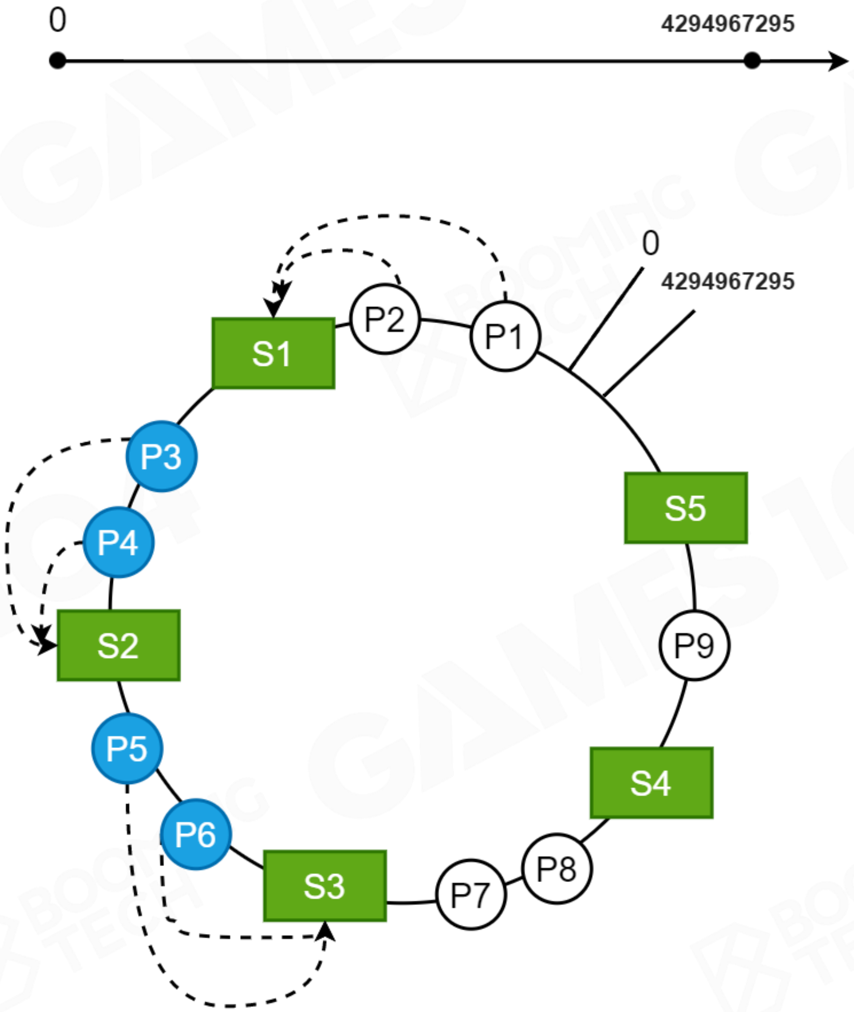
Consistent Hashing (1/3)

It was designed to avoid the problem of having to reassign every player when a server is added or removed throughout the cluster

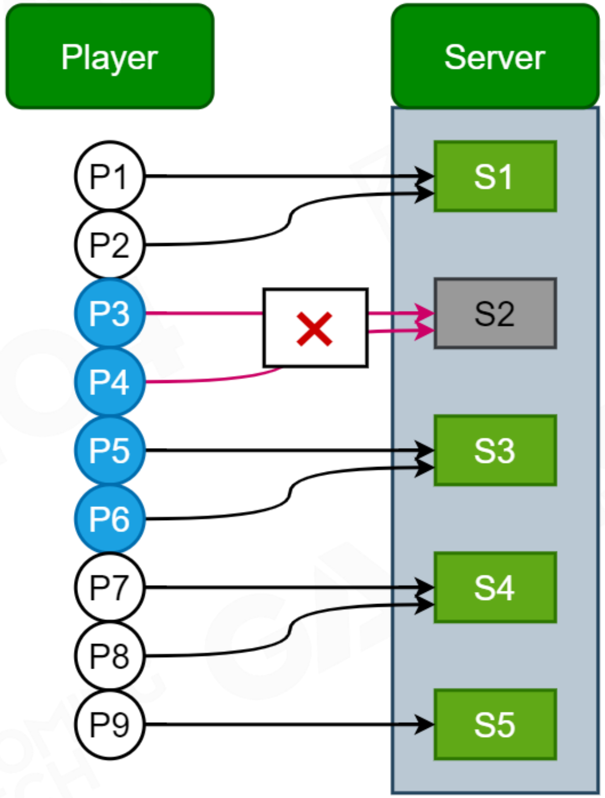


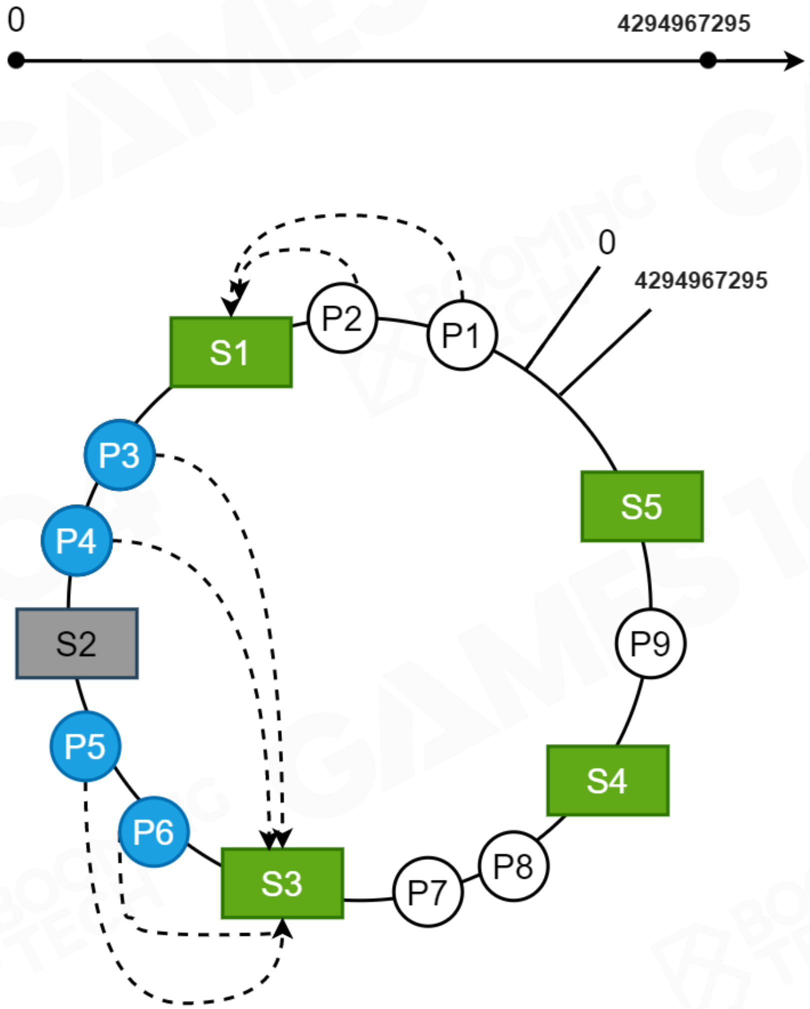
Consistent Hashing (2/3)



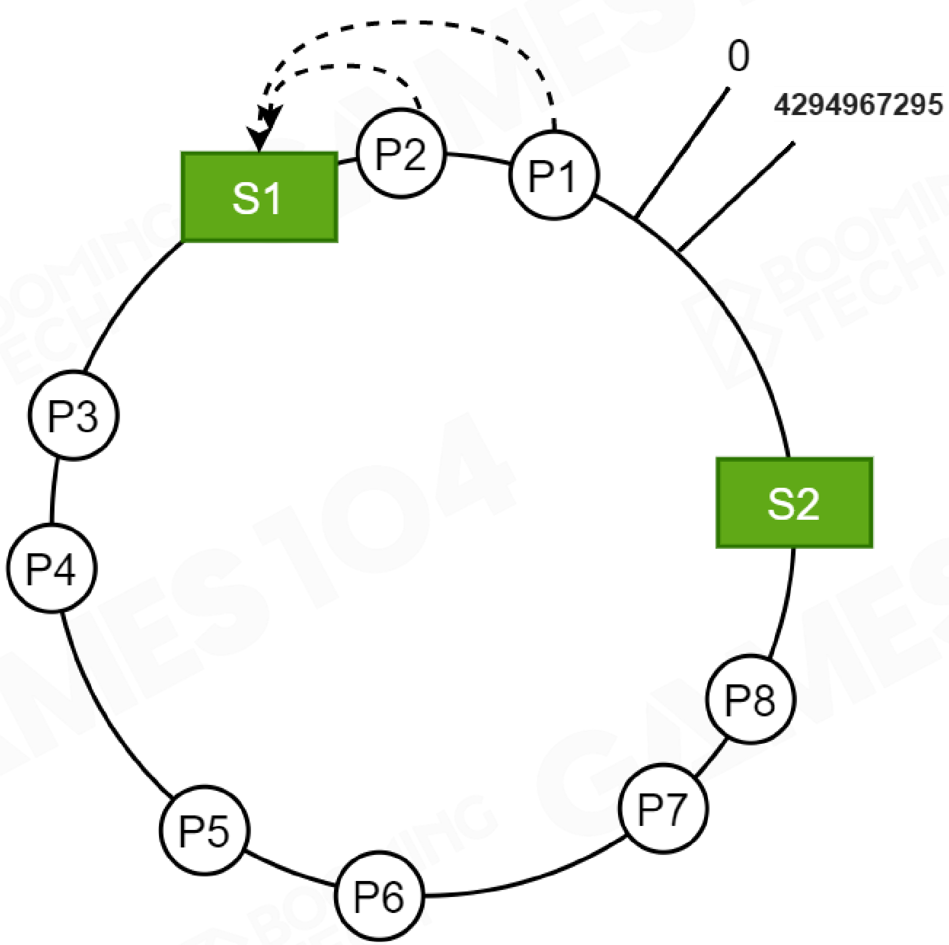


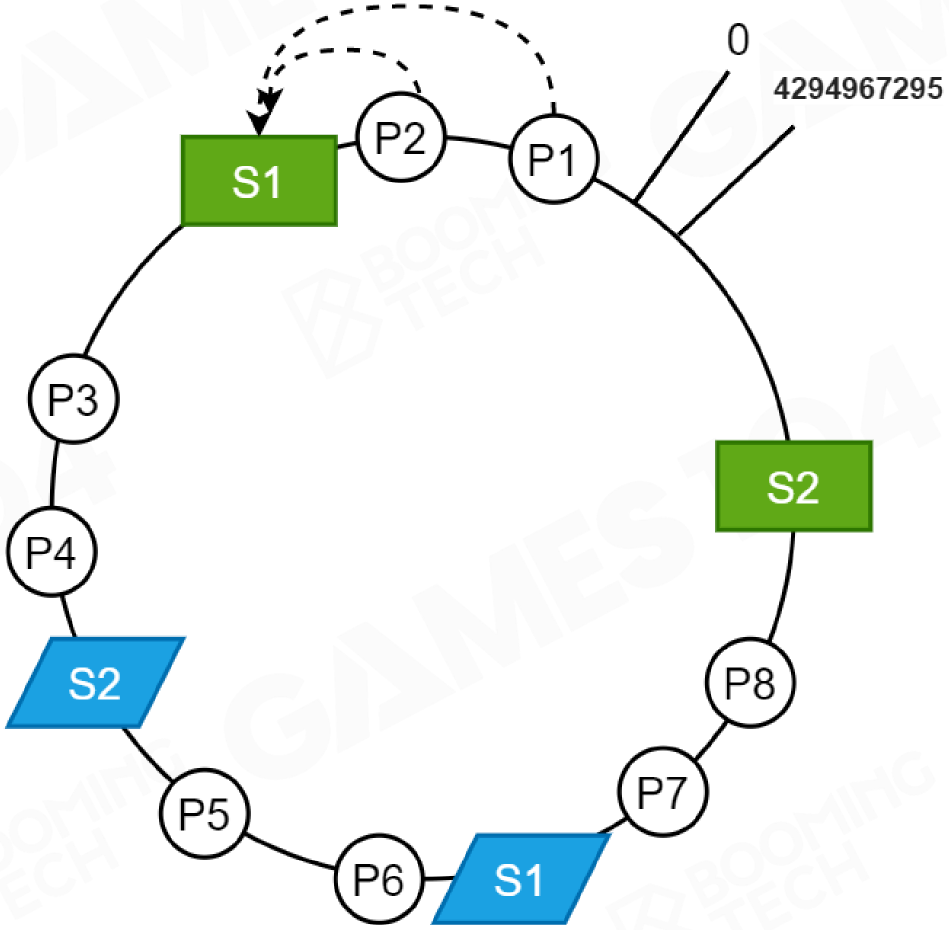
Consistent Hashing (3/3)





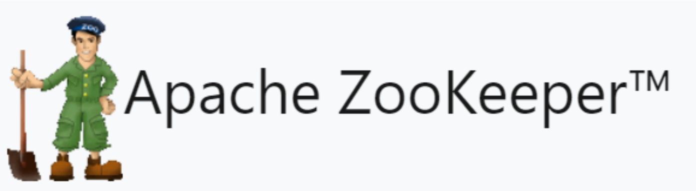
Virtual Server Node in Consistent Hashing

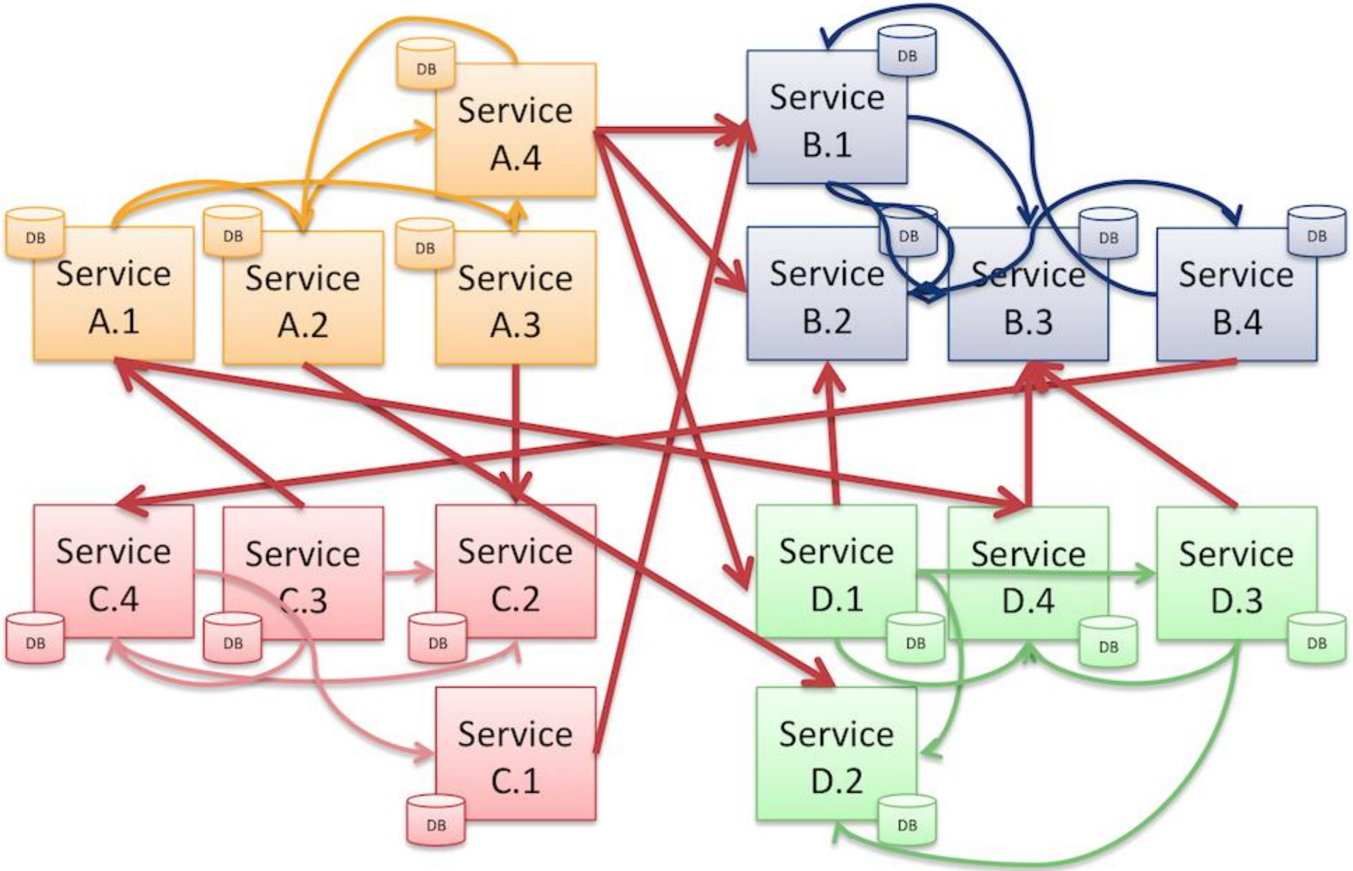




Servers Management

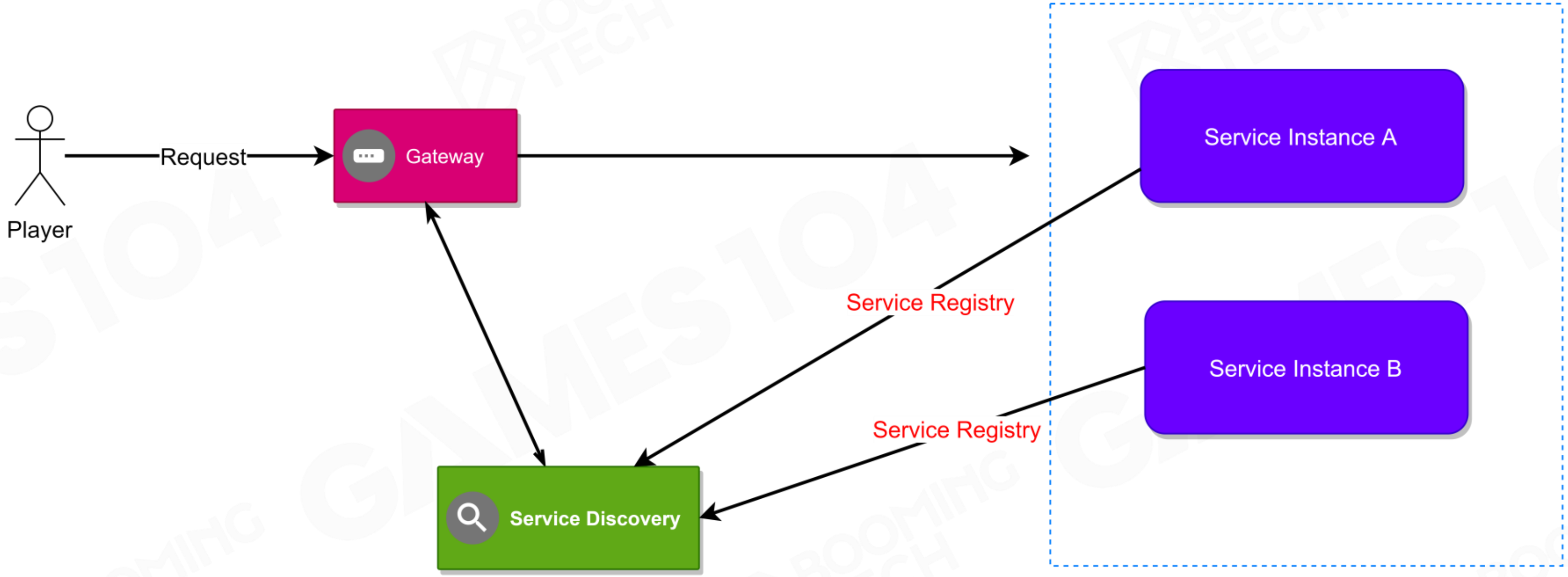
- The number of services increases
- Difficult to manage
- Lacks the flexibility to change the IP port at a later point in time





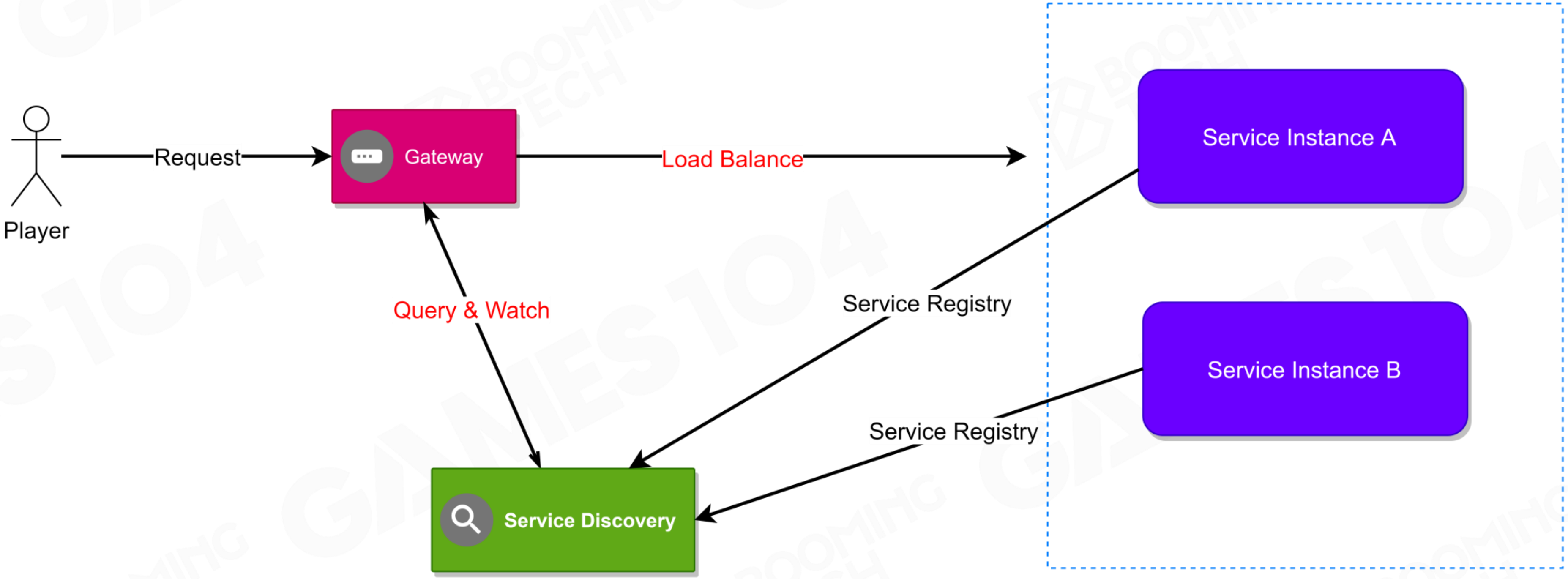
Service Discovery - Registry

- Registers itself with the service registry when it enters the system
- An example of Register value
  - server type/server_name@server_ip:port



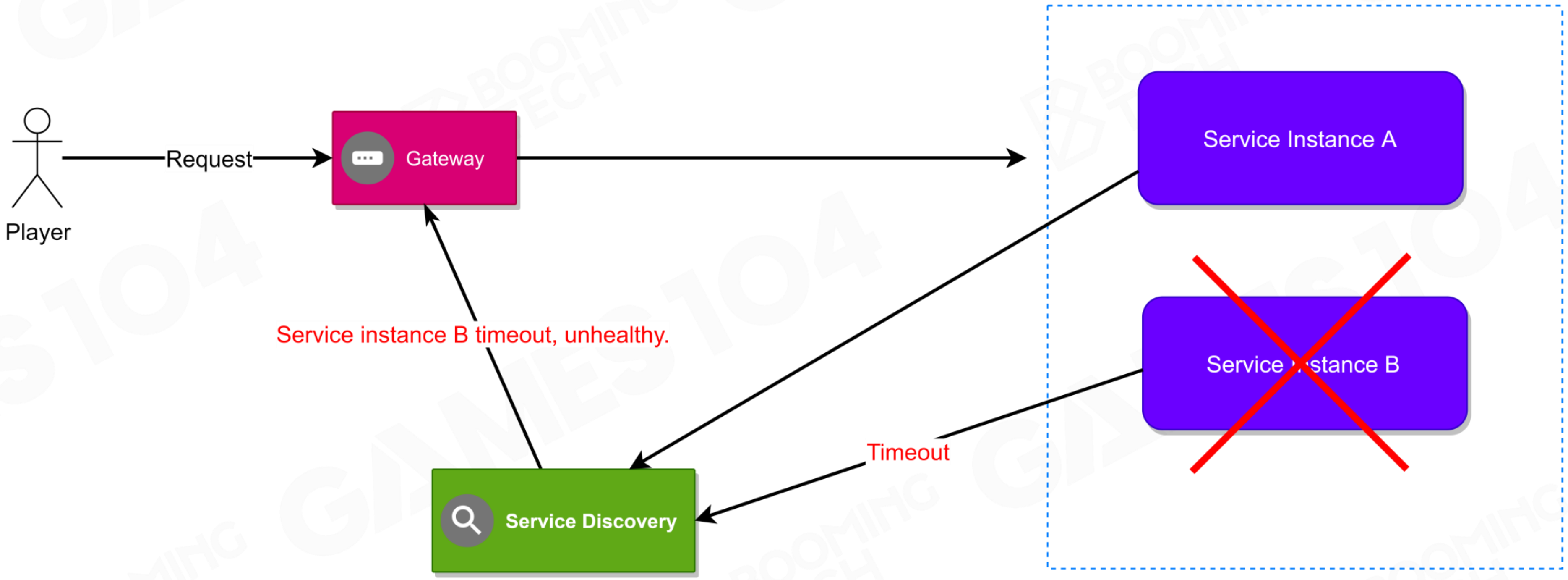
Service Discovery - Query and Watch

- Request service discovery service to query all values through service type and watch it



Service Discovery - Health Check

- Notice Gateway Server B Failure when Server Instance B Heartbeat timeout



Bandwidth Optimization

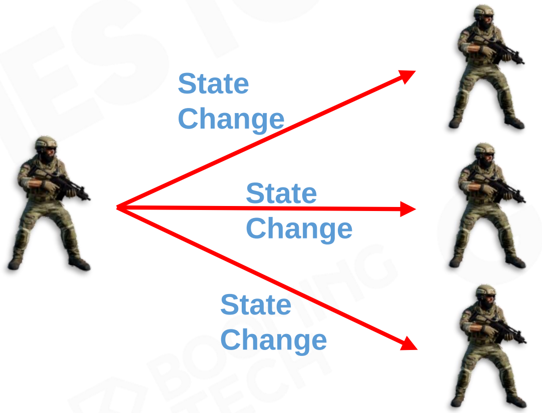
Why Bandwidth Matters

- Usage-based billing: e.g. mobile, cloud service
- Latency increased by bandwidth: packet splitting/drop
- Connection drops due to message overflow

Calculate Bandwidth

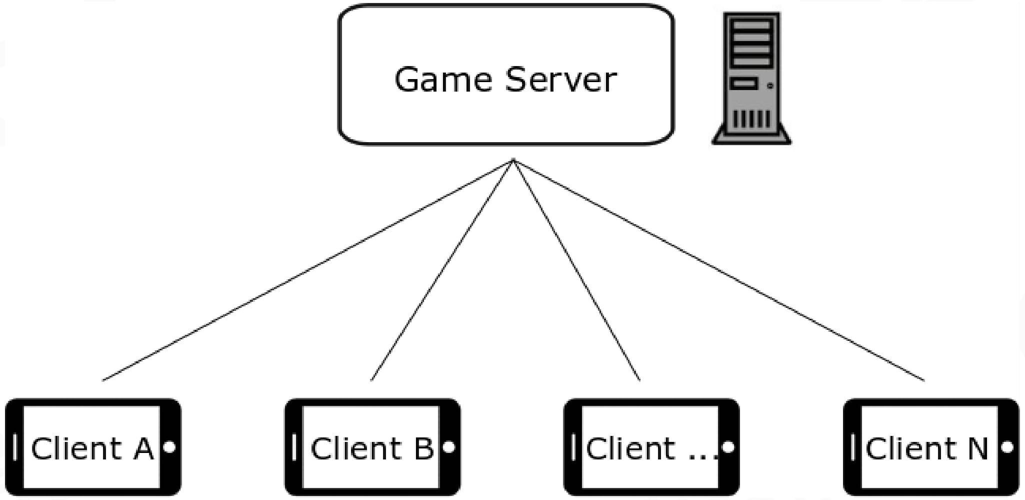
Affecting factors

- ***n*** = player numbers
- ***f*** = update frequency
- ***s*** = size of game state



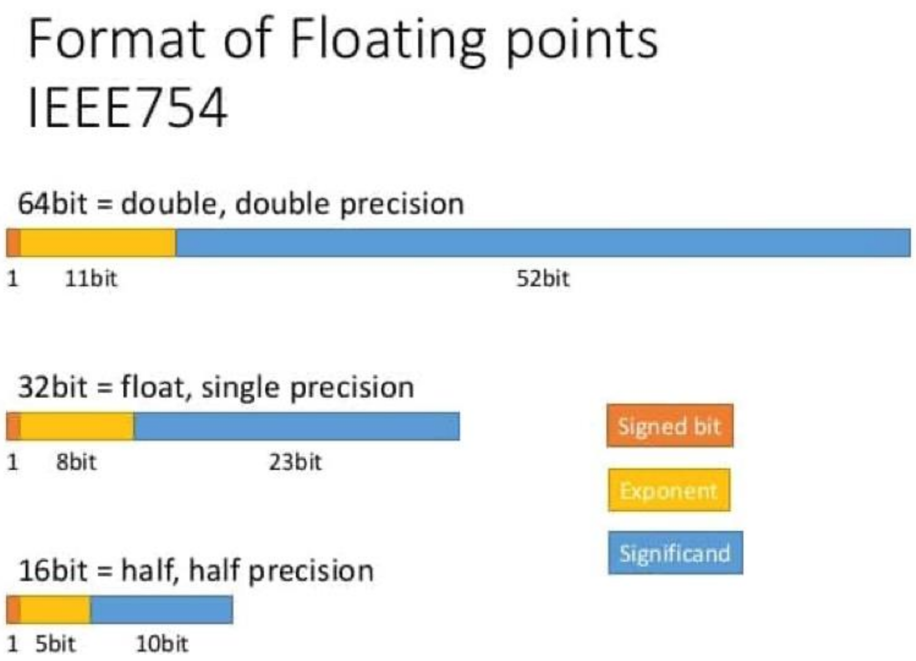
Data transfer per second

- Server: $O(n \cdot s \cdot f)$
- Client (downstream): $O(s \cdot f)$
- Client (upstream): $O(f)$



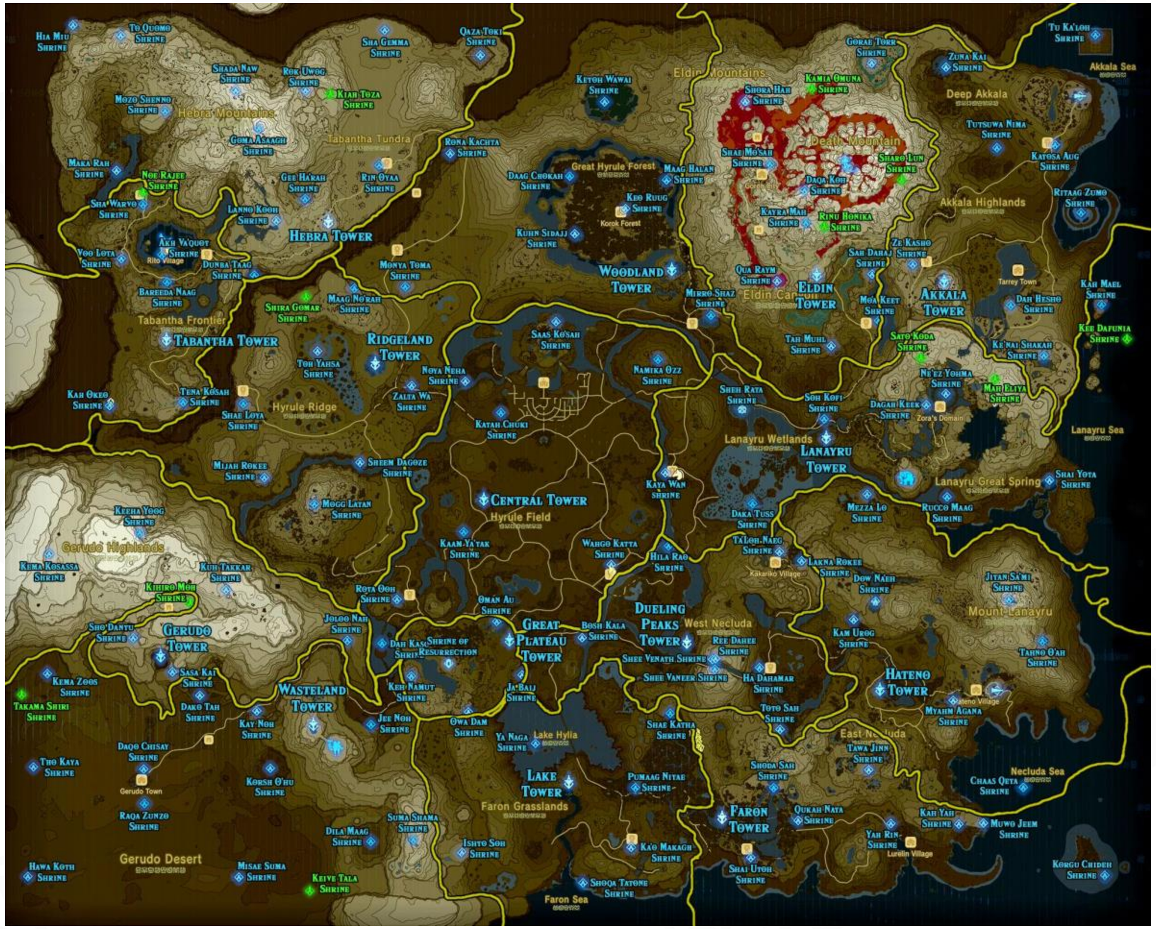
Data Compression (1/2)

- There are a lot of floating point numbers in the game synchronization data, such as position, rotation, speed, etc.
- Choosing the right floating-point precision can save a lot of bandwidth
  - e.g When representing human running speed, only half precision is required


Format of Floating points
IEEE754
64bit = double, double precision
1
11bit
52bit
32bit = float, single precision
1
8bit
23bit
16bit = half, half precision
1
5bit
10bit
Signed bit
Exponent
Significand

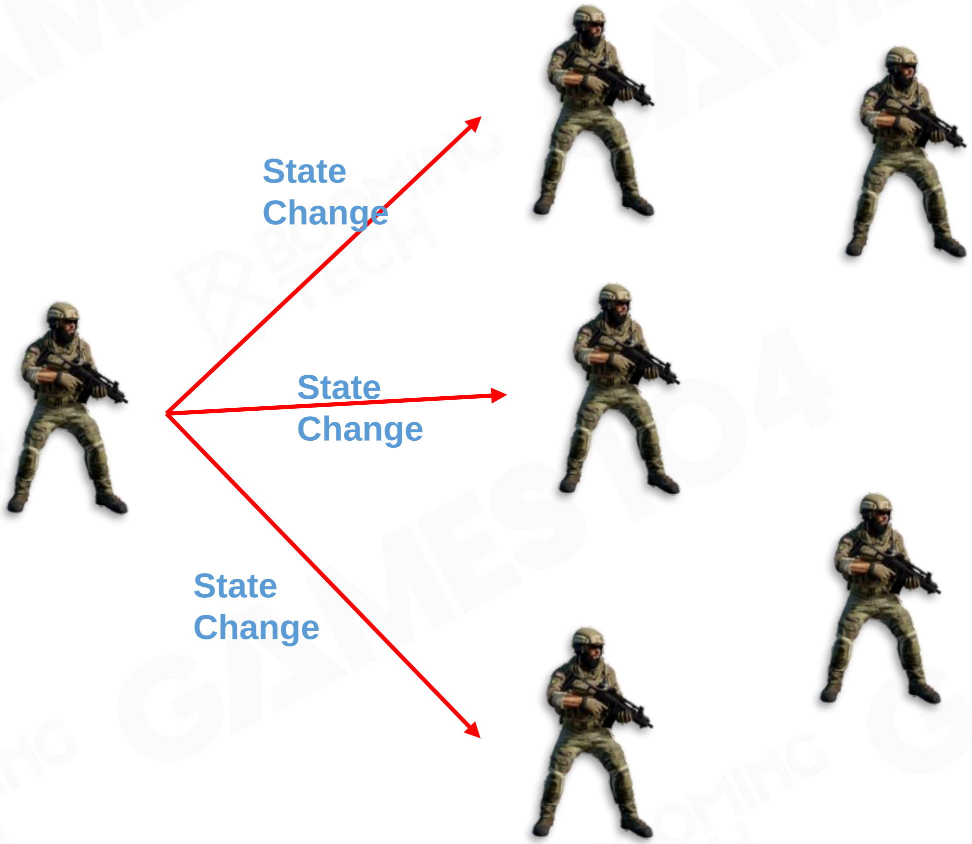
Data Compression (2/2)

- When representing player position, the player will only move within a certain range due to player speed limitations
- We can divide the map into different small pieces and use the relative position to represent the player's position, which can reduce the precision of the floating point number of the synchronization position



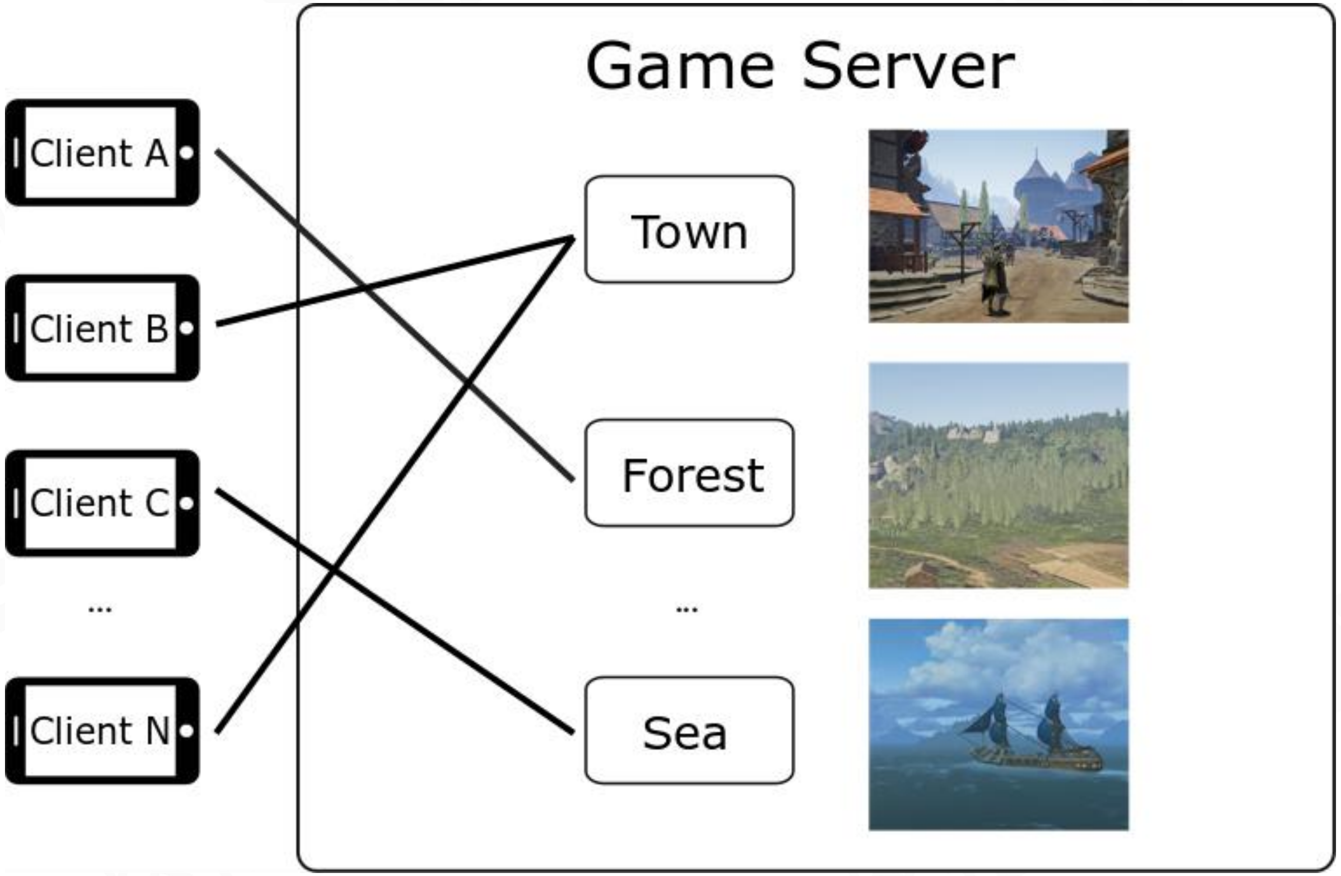
Object Relevance

- Objects in relevance
  - The player will be informed of state updates
  - Usually, the ones player can see & interact
- Easiest implementation: all objects relevant to all clients (for small player num). $O(n^2)$
- Limiting factor for max concurrent players



Relevance - Static Zones

- Distribute players into different zones
- Players are relevant in the same zone
- Reduce bandwidth waste



Affecting factors

- *n = player numbers*
- *f = update frequency*
- *s = size of game state*

Relevance - Area of Interest (AOI)

- The area within which objects are relevant to Player/NPC
- Only see & interact with objects within range
- Remove unnecessary network data

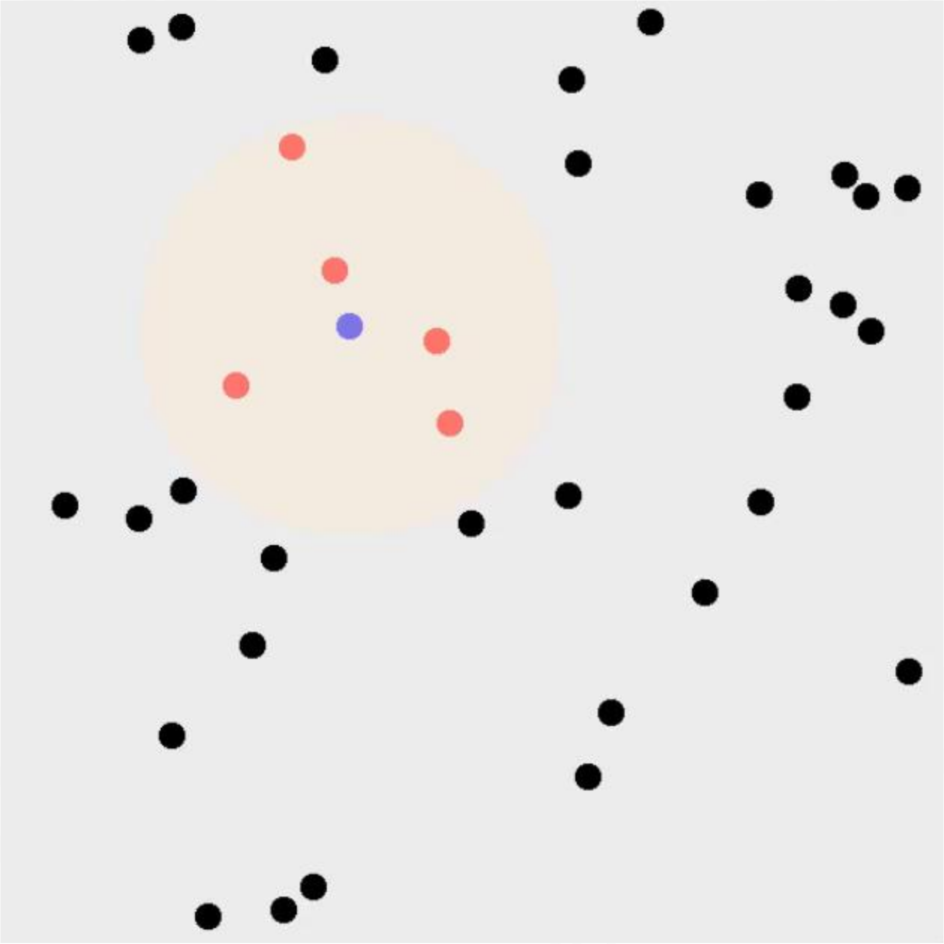
Affecting factors

- *n = player numbers*
- *f = update frequency*
- *s = size of game state*

AOI - Direct Range-Query

- $\sqrt{(x_{player} - x_i)^2 - (y_{player} - y_i)^2} <= r_{aoi}$
- Simple to implement
- Time complexity: $O(n^2)$
- Not suitable for MMOG, e.g. 1000 players in one zone, 20 ticks/s

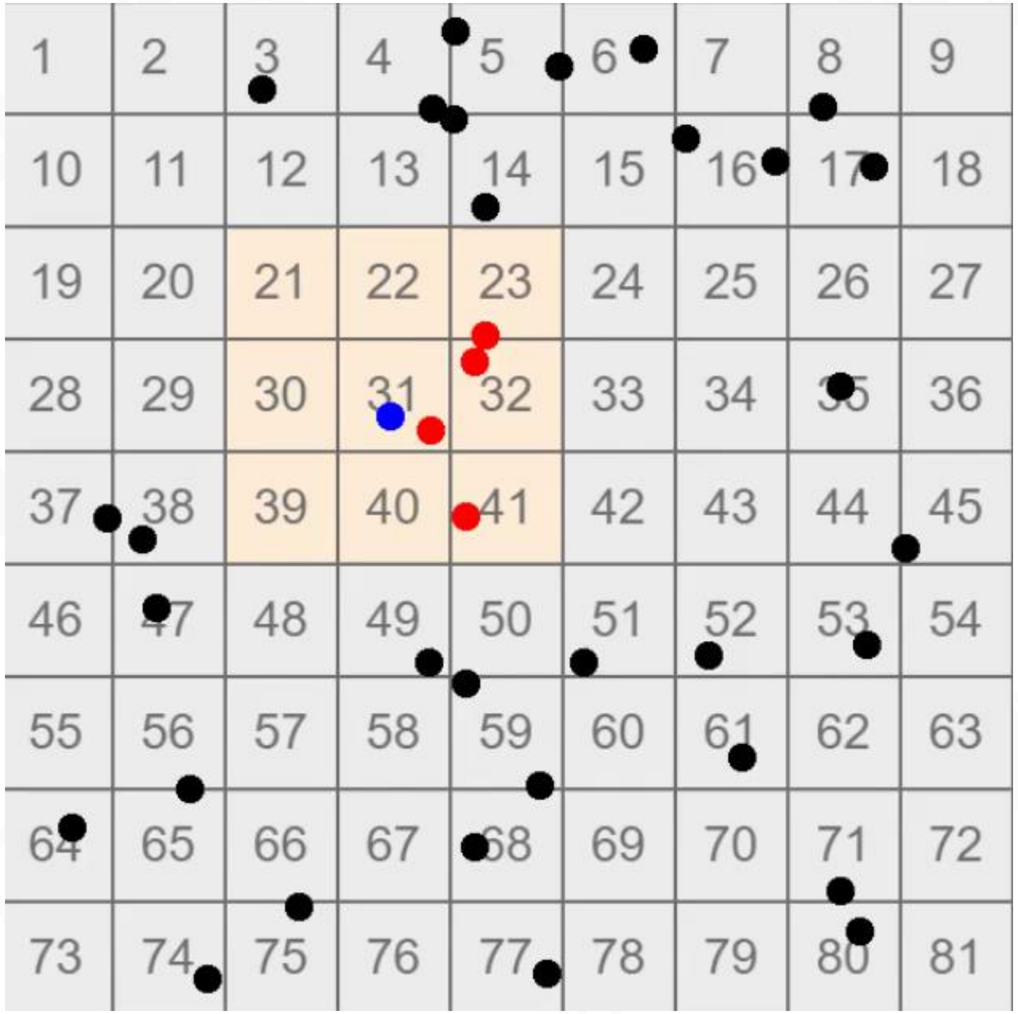
⇨ 1000x1000x20 = 20,000,000 distance computations per second



AOI - Spatial-Grid (1/3)

Mapping Entities

- Map entity (x, y) ⟹ grid N
- Relevant entities in the grids around current player's grid
- Player's AOI list can be cached



1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

AOI - Spatial-Grid (2/3)

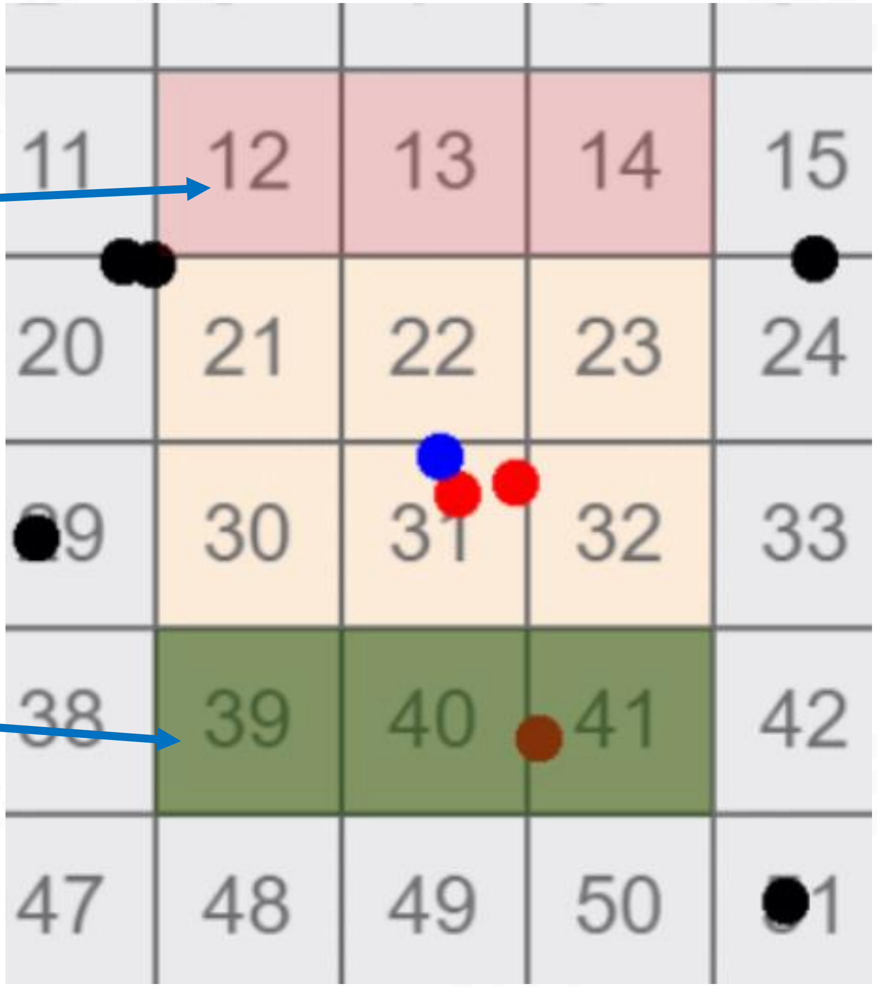
Events

Enter

- Add entities from observation (observed) list

Leave

- Remove entities from observation (observed) list



AOI - Spatial-Grid (3/3)

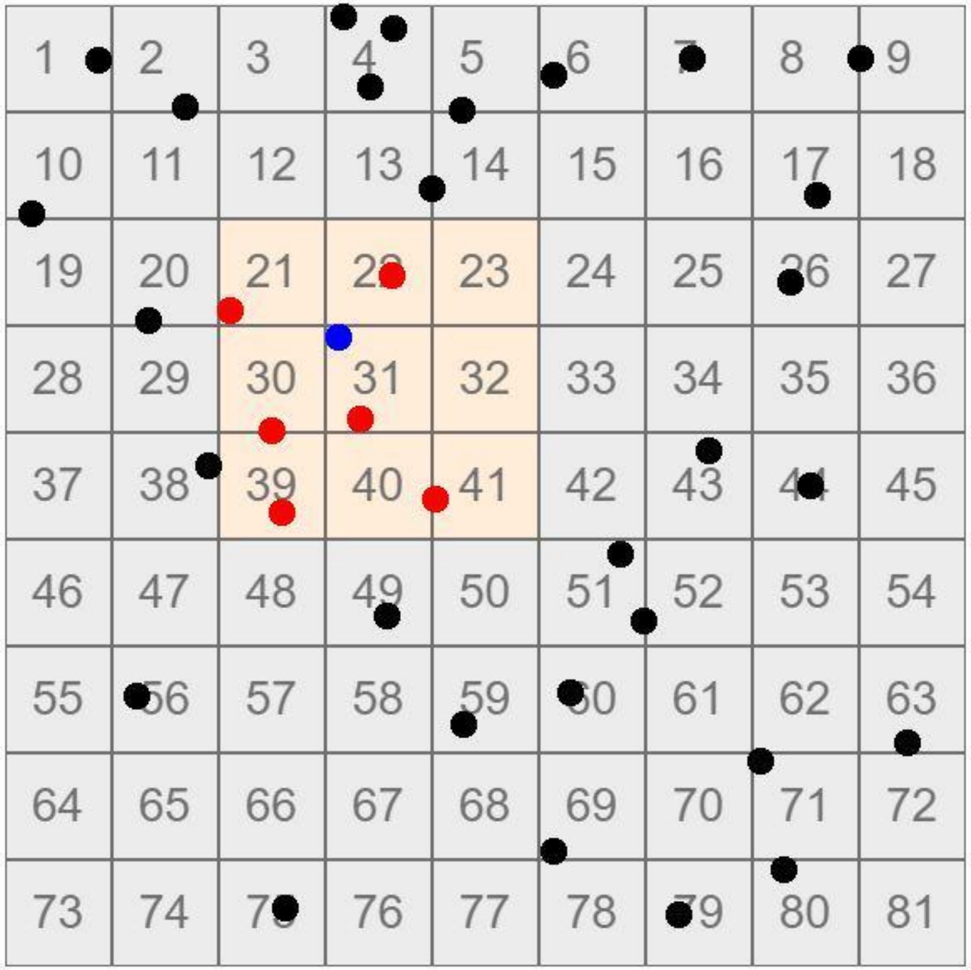
Pros and Cons

Pros

- Fast query time **O(1)**

Cons

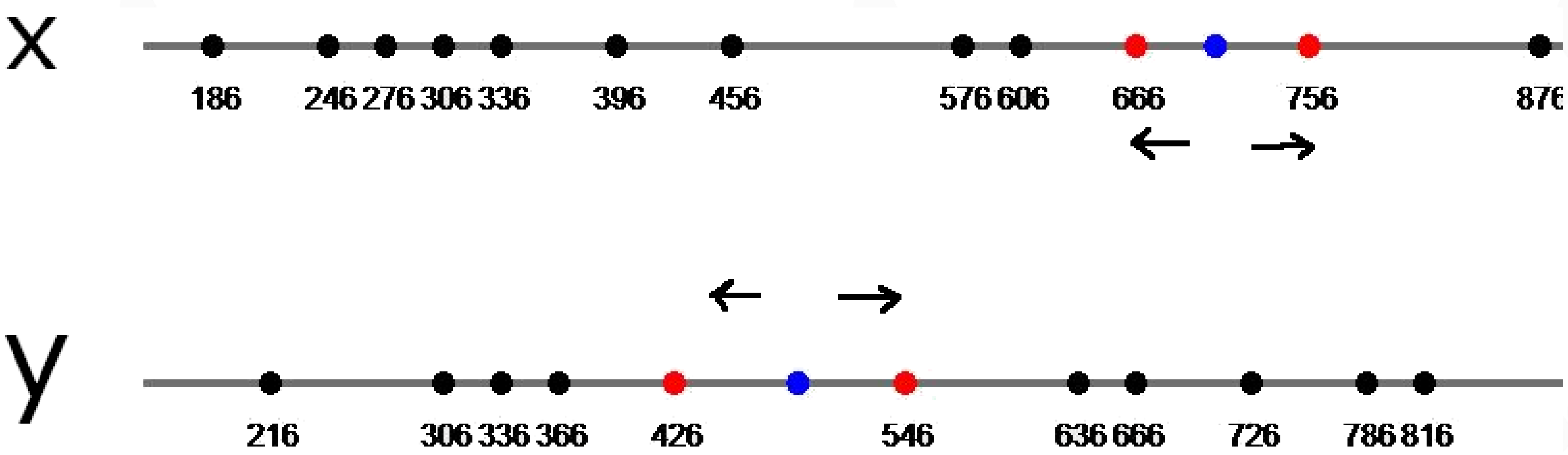
- Small grid: high memory cost
- Large grid: high CPU cost
- Object with varying AOI radius?



1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

AOI - Orthogonal Linked-list (1/4)

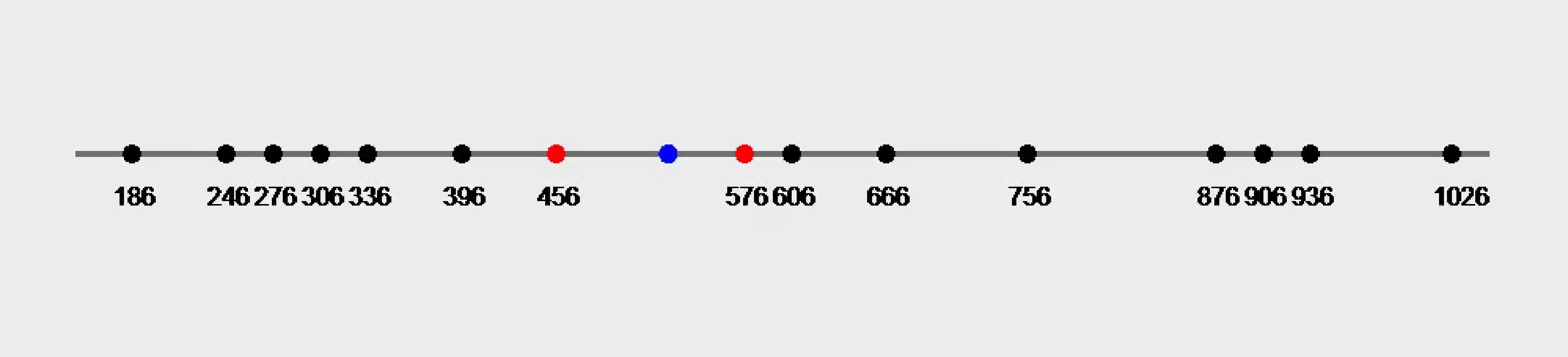
- Game entities in double linked-list
  - xlist, ylist
  - ascending order
- Less Objects to traverse

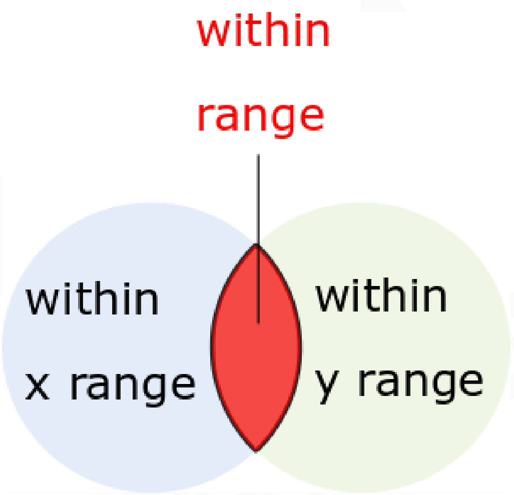


AOI - Orthogonal Linked-List (2/4)

Traverse entities

- Within aoi radius
- Left/right direction
- For both x/y lists

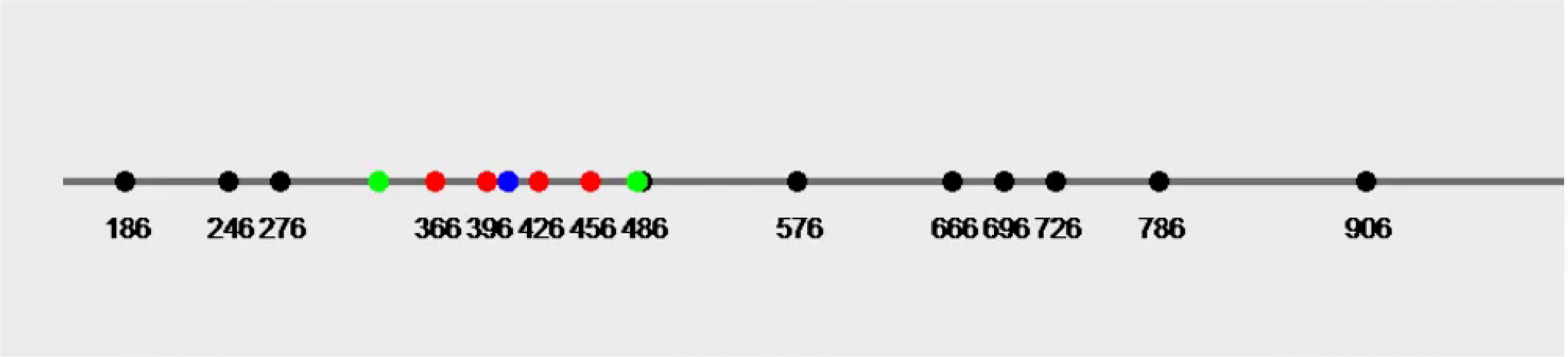




AOI - Orthogonal Linked-List (3/4)

Better Approach - Range Trigger

- Entity move ⇨ trigger move
- Compare with trigger
- Event driven



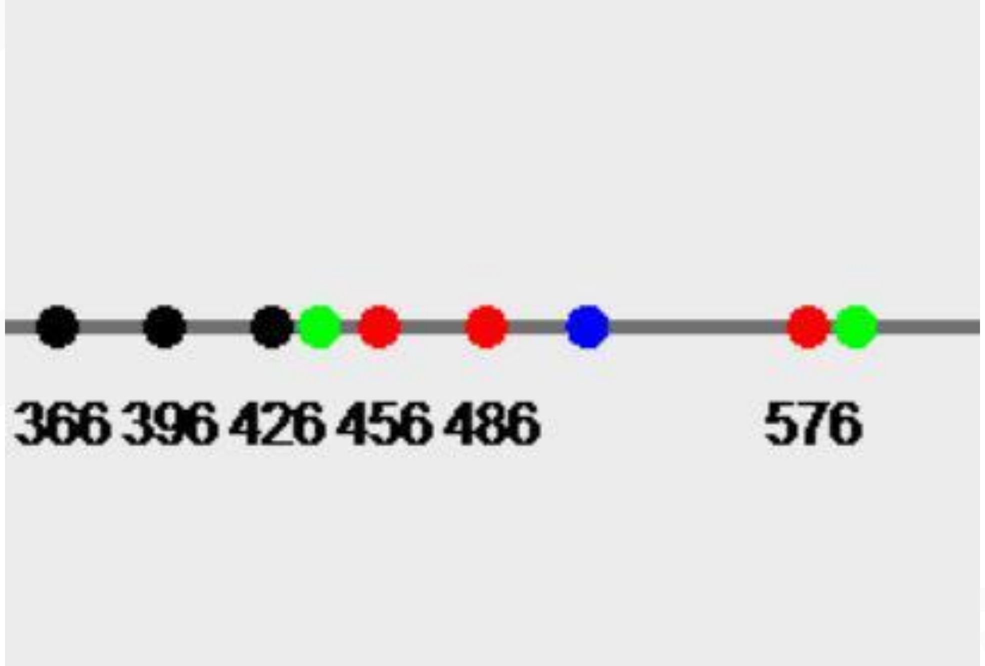
AOI - Orthogonal Linked-List Approach (4/4)

Pros

- Memory efficient
- Varying AOI radius

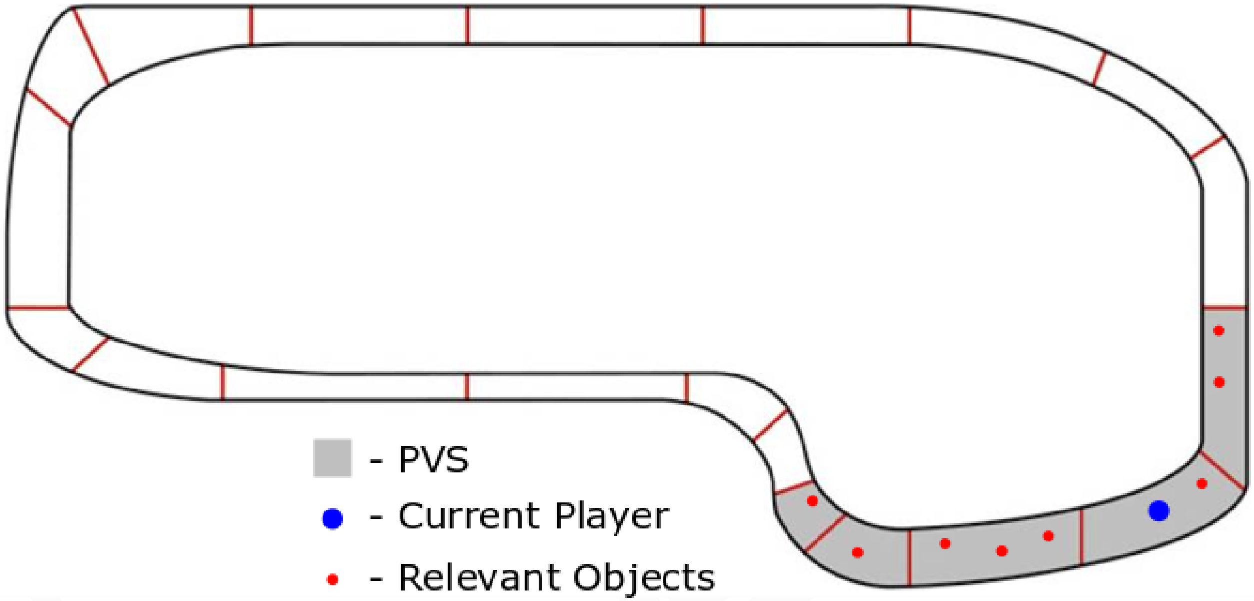
Cons

- New object insertion cost **O(n)**
- Not Suitable when entities move large distance frequently



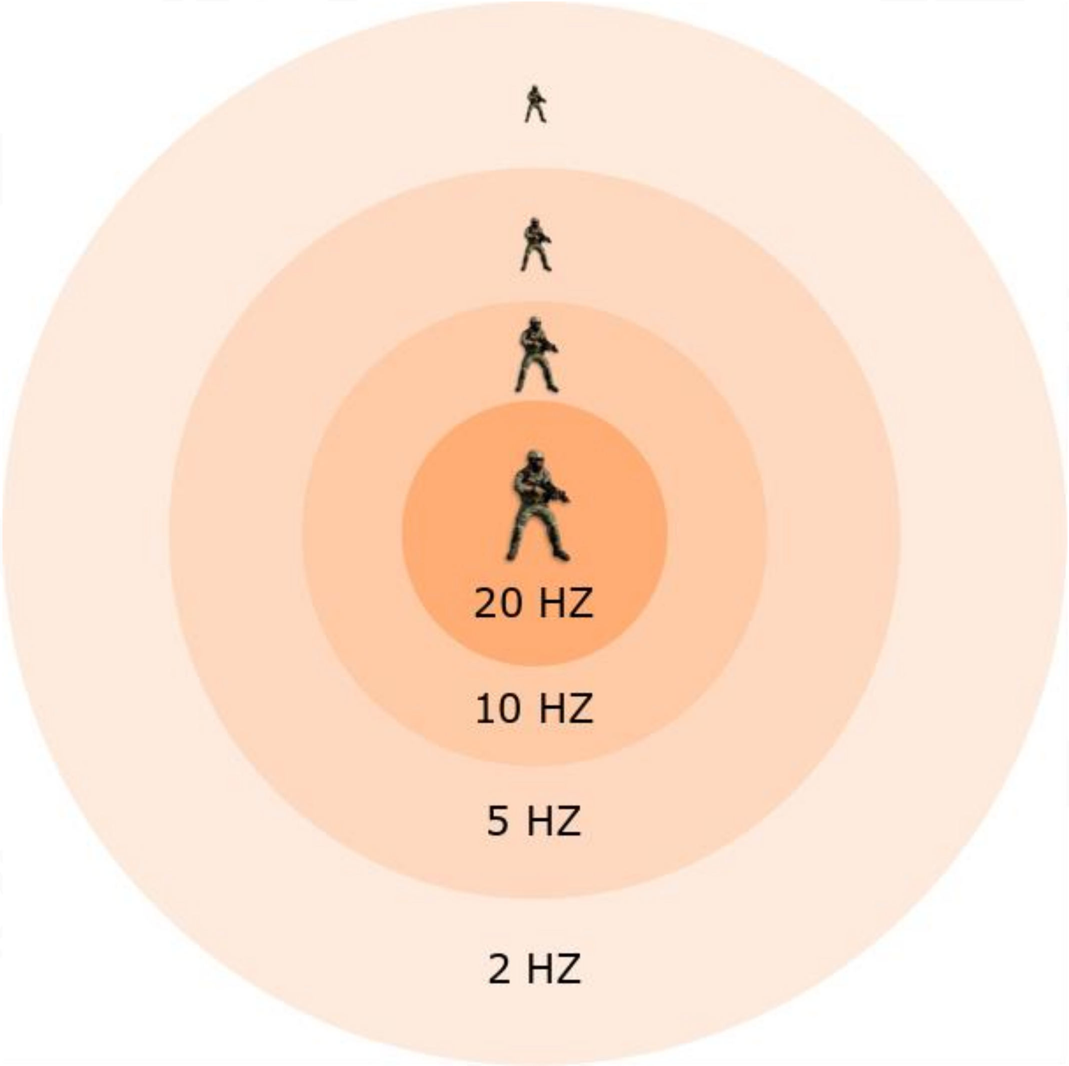
AOI - Potentially Visible Set (PVS)

- Set of potentially visible areas
- Can be calculated offline
- Determine relevant objects from PVS
- e.g. Racing game: fast-moving car



Varying Update Frequency by Player Position

- Distance-based update frequency
- Only closer objects are interactable
- Distance ↑ ⇨ *f* ↓ ⇨ bandwidth ↓

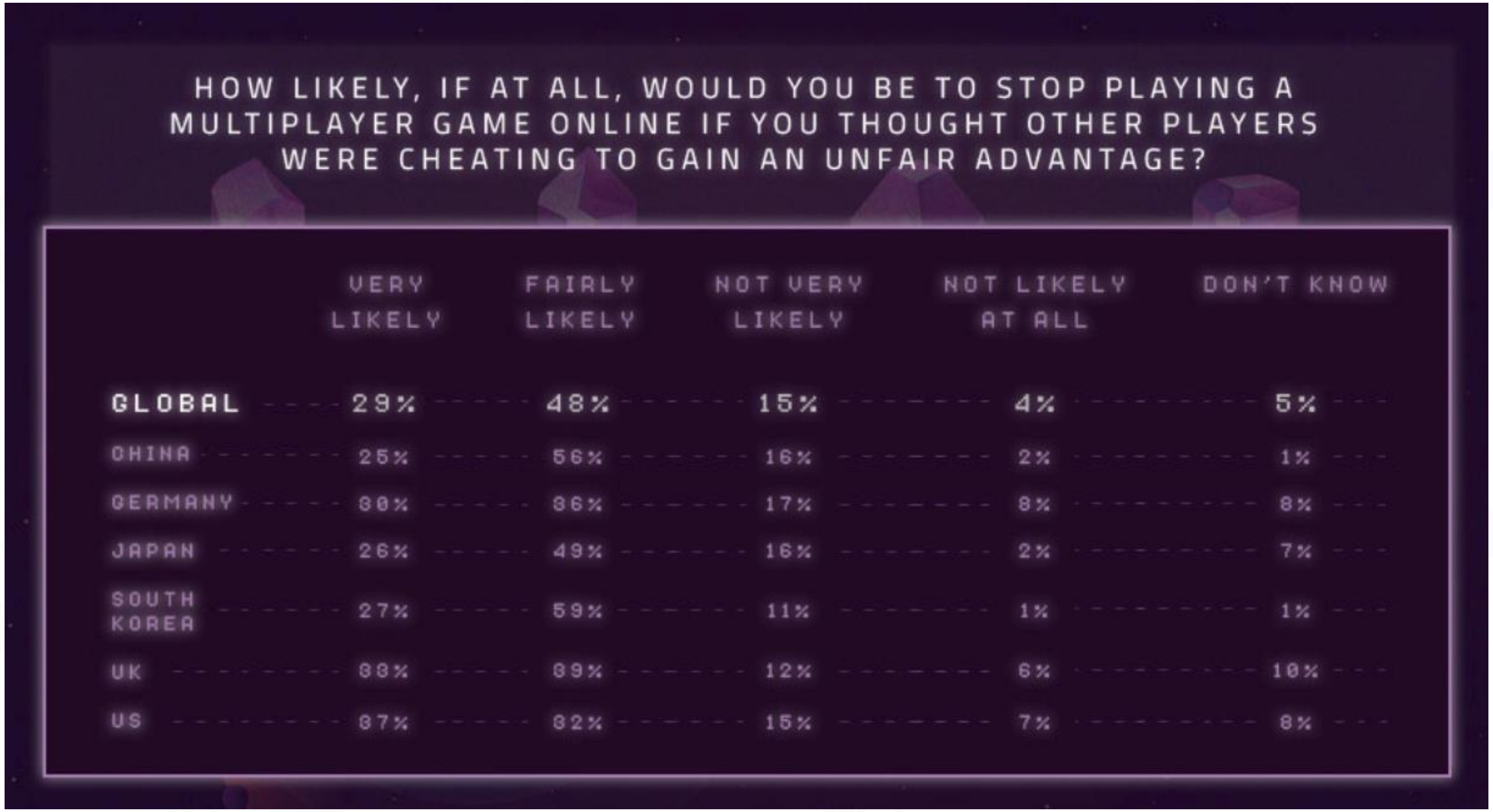


Affecting factors

- *n = player numbers*
- *f = update frequency*
- *s = size of game state*

Anti-Cheat

Cheating Kill Online Games



HOW LIKELY, IF AT ALL, WOULD YOU BE TO STOP PLAYING A MULTIPLAYER GAME ONLINE IF YOU THOUGHT OTHER PLAYERS WERE CHEATING TO GAIN AN UNFAIR ADVANTAGE?

	VERY LIKELY	FAIRLY LIKELY	NOT VERY LIKELY	NOT LIKELY AT ALL	DON'T KNOW
GLOBAL	29%	48%	15%	4%	5%
CHINA	25%	56%	16%	2%	1%
GERMANY	30%	36%	17%	8%	8%
JAPAN	26%	49%	16%	2%	7%
SOUTH KOREA	27%	59%	11%	1%	1%
UK	33%	39%	12%	6%	10%
US	37%	32%	15%	7%	8%

77% of players will likely stop playing online games when other players are cheating, according to the survey of Irdeto.

Millions Ways of Cheating

Game code modifications

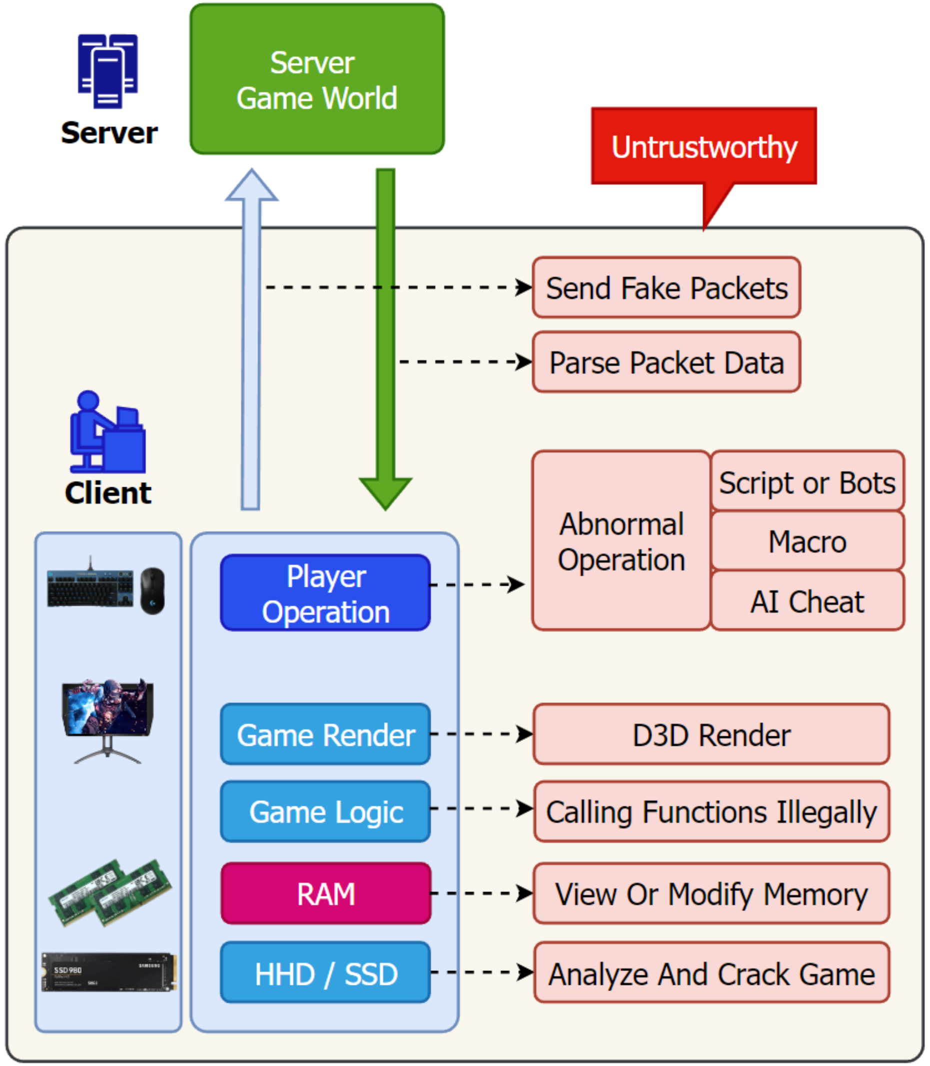
- Modify or read memory data
- Crack client

System software invoke

- D3D Render Hook
- Simulate mouse and keyboard operations
- ...

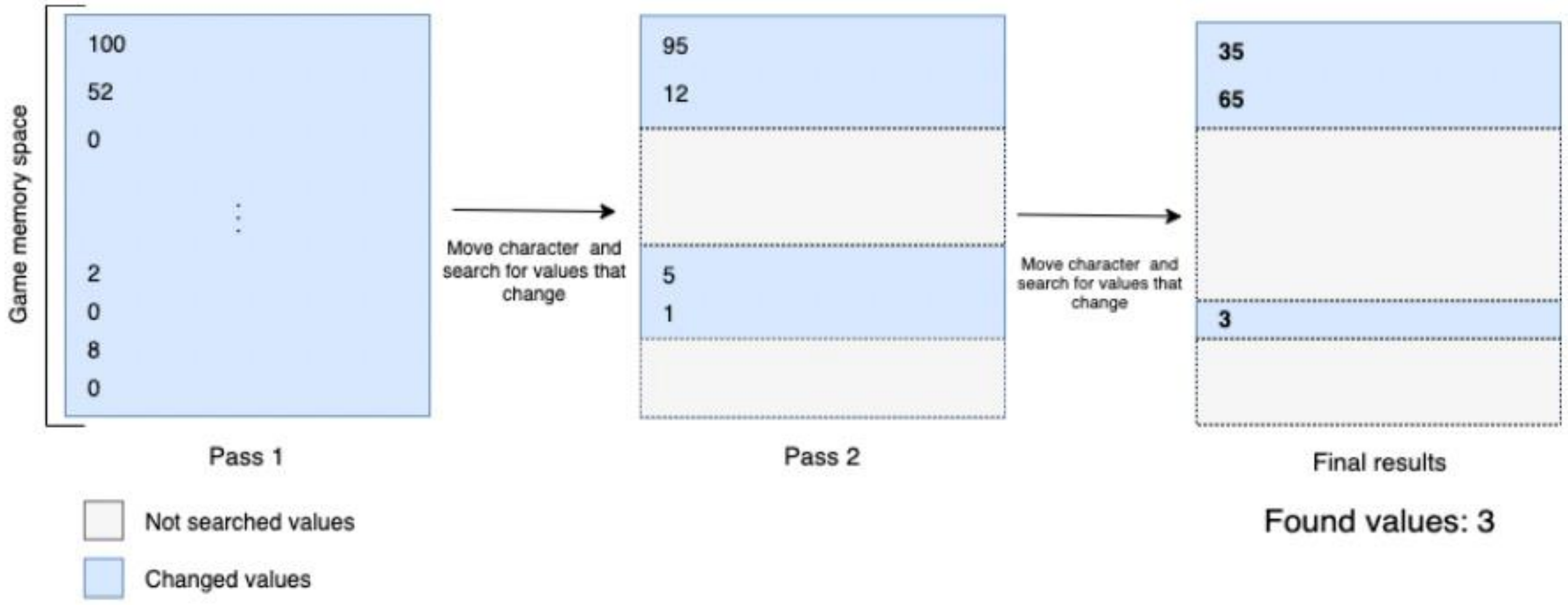
Net Packet interception

- Send fake packets
- Modify packet data



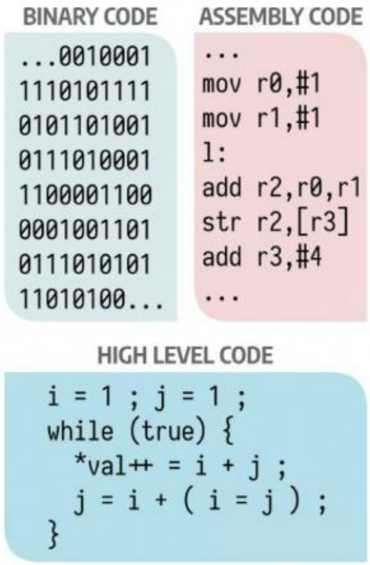
Obfuscating Memory

- A cheater might be able to get the location of the player coordinates in the memory and move the character ignoring the game rules, such as passing the wall
- Furthermore, the cheater can utilize the location of these values to map out even larger data structures in the memory, such as the player object itself



Executable Packers (1/2)

- Game core logic can be restored by reverse engineering
- Players can crack the game by analyzing the code, finding game loopholes, making plug-ins, etc..



BINARY CODE

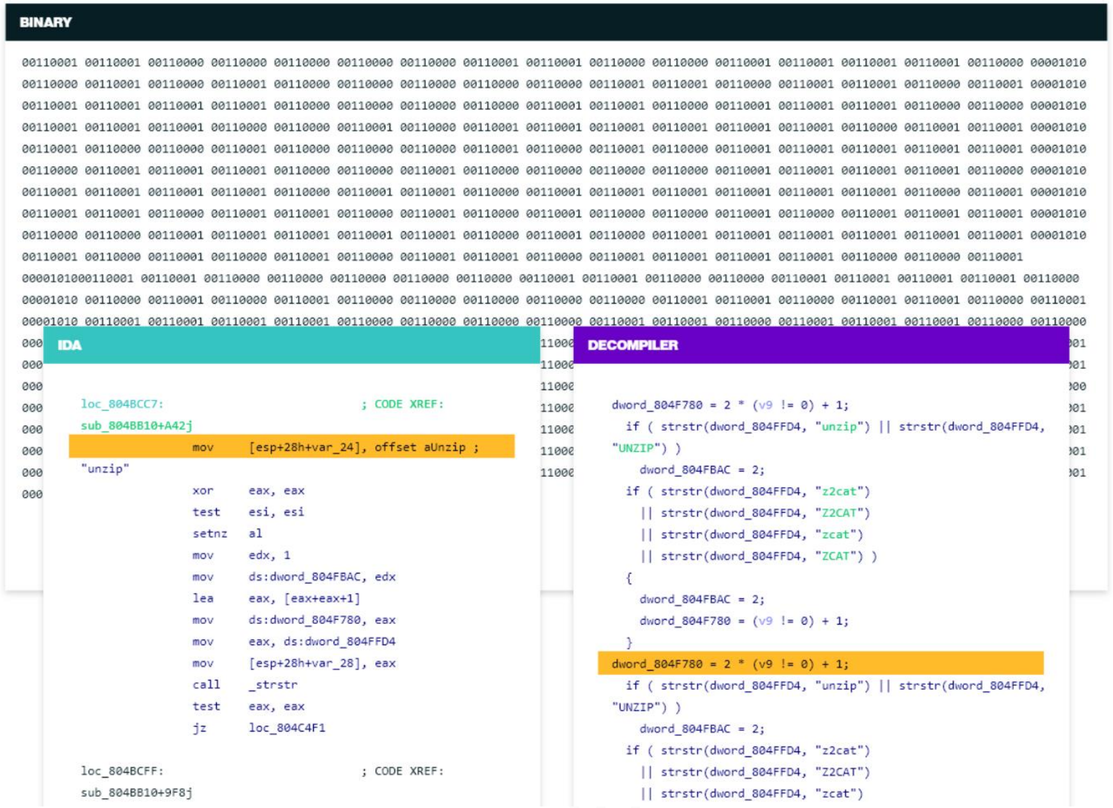
```
...0010001
1110101111
0101101001
0111010001
1100001100
0001001101
0111010101
11010100...
```

ASSEMBLY CODE

```
...
mov r0,#1
mov r1,#1
l:
add r2,r0,r1
str r2,[r3]
add r3,#4
...
```

HIGH LEVEL CODE

```
i = 1 ; j = 1 ;
while (true) {
  *val++ = i + j ;
  j = i + ( i = j ) ;
}
```



BINARY

```
00110001 00110001 00110000 00110000 00110000 00110000 00110000 00110001 00110001 00110000 00110000 00110001 00110001 00110001 00110001 00110000 00001010
00110000 00110001 00110000 00110001 00110000 00110000 00110000 00110000 00110000 00110001 00110001 00110000 00110001 00110001 00110000 00110001 00001010
00110001 00110001 00110001 00110001 00110000 00110000 00110000 00110000 00110001 00110001 00110000 00110001 00110001 00110001 00110000 00110000 00001010
00110001 00110001 00110001 00110000 00110000 00110001 00110000 00110001 00110001 00110001 00110001 00110001 00110001 00110000 00110001 00110001 00001010
00110001 00110000 00110000 00110001 00110000 00110000 00110000 00110001 00110000 00110001 00110000 00110001 00110001 00110001 00110001 00110001 00001010
00110000 00110001 00110001 00110001 00110000 00110000 00110001 00110000 00110000 00110000 00110000 00110001 00110001 00110001 00110001 00110000 00001010
00110001 00110001 00110001 00110001 00110000 00110001 00110001 00110000 00110001 00110001 00110001 00110001 00110001 00110001 00110000 00110001 00001010
00110001 00110001 00110000 00110001 00110001 00110000 00110001 00110000 00110001 00110000 00110000 00110001 00110000 00110001 00110001 00110001 00001010
00110000 00110000 00110001 00110001 00110001 00110001 00110001 00110000 00110001 00110000 00110001 00110001 00110001 00110001 00110001 00110001 00001010
00110001 00110000 00110001 00110000 00110001 00110000 00110001 00110001 00110000 00110001 00110001 00110001 00110001 00110000 00110000 00110001
0000101000110001 00110001 00110000 00110000 00110000 00110000 00110000 00110001 00110001 00110000 00110000 00110001 00110001 00110001 00110001 00110000
00001010 00110000 00110001 00110000 00110001 00110000 00110000 00110000 00110000 00110000 00110001 00110001 00110000 00110001 00110001 00110000 00110001
00001010 00110001 00110001 00110001 00110001 00110000 00110000 00110000 00110000 00110001 00110001 00110000 00110001 00110001 00110001 00110000 00110000
```

IDA

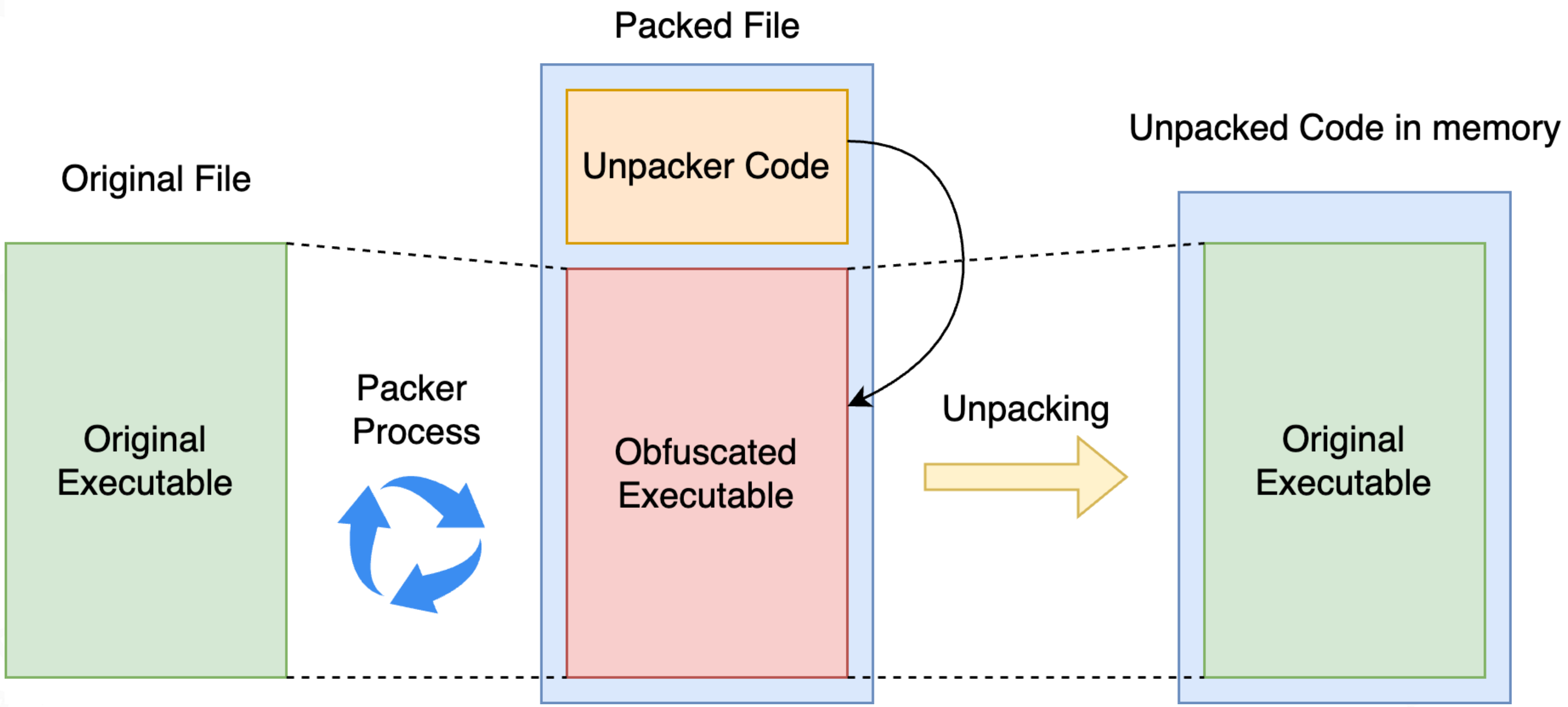
```
loc_804BCC7:                          ; CODE XREF:
sub_804BB10+A42j
                 mov     [esp+28h+var_24], offset aUnzip ;
"unzip"
                 xor     eax, eax
                 test    esi, esi
                 setnz   al
                 mov     edx, 1
                 mov     ds:dword_804FBAC, edx
                 lea     eax, [eax+eax+1]
                 mov     ds:dword_804F780, eax
                 mov     eax, ds:dword_804FFD4
                 mov     [esp+28h+var_28], eax
                 call    _strstr
                 test    eax, eax
                 jz      loc_804C4F1

loc_804BCFF:                          ; CODE XREF:
sub_804BB10+9F8j
```

DECOMPILER

```
dword_804F780 = 2 * (v9 != 0) + 1;
  if ( strstr(dword_804FFD4, "unzip") || strstr(dword_804FFD4,
"UNZIP") )
    dword_804FBAC = 2;
  if ( strstr(dword_804FFD4, "z2cat")
    || strstr(dword_804FFD4, "Z2CAT")
    || strstr(dword_804FFD4, "zcat")
    || strstr(dword_804FFD4, "ZCAT") )
  {
    dword_804FBAC = 2;
    dword_804F780 = (v9 != 0) + 1;
  }
dword_804F780 = 2 * (v9 != 0) + 1;
  if ( strstr(dword_804FFD4, "unzip") || strstr(dword_804FFD4,
"UNZIP") )
    dword_804FBAC = 2;
  if ( strstr(dword_804FFD4, "z2cat")
    || strstr(dword_804FFD4, "Z2CAT")
    || strstr(dword_804FFD4, "zcat")
```

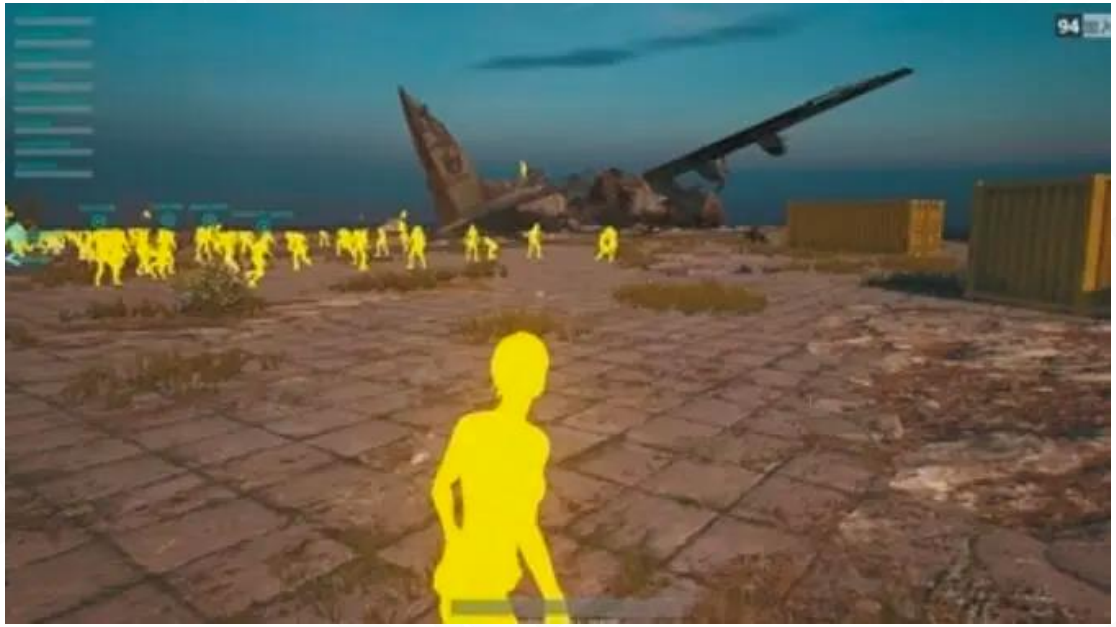
Executable Packers (2/2)

- The packager obfuscates the source program and adds decompression code
- The decompression code will execute first, and the source program is decrypted in memory



Verifying Local Files by Hashing

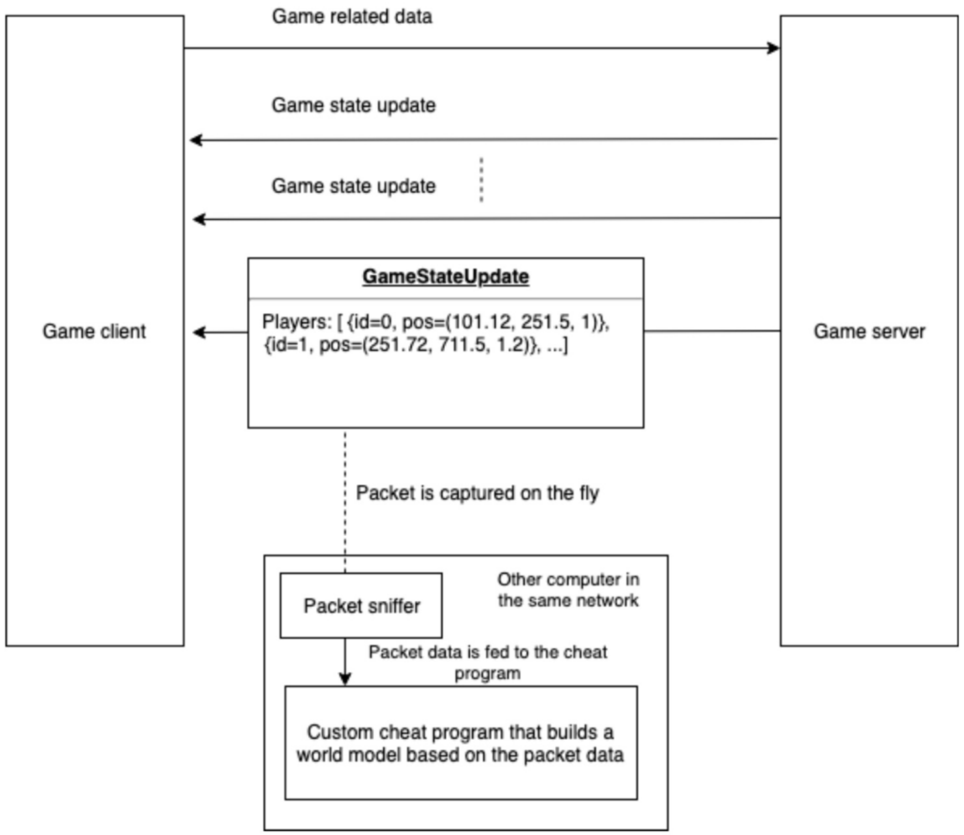
- Ensure that the game files have not been modified
- For example, the cheater could modify the wall textures to be transparent so all enemies could be seen through the walls
- The cheater could also adjust the lightning to make it easier to see enemies



Packet Interception and Manipulation

- When the data is not encrypted or hacked, the player can build game logic based on packet data even without starting the game

- Such cheat programs often become money-making tools, which seriously reduce game's the overall profit

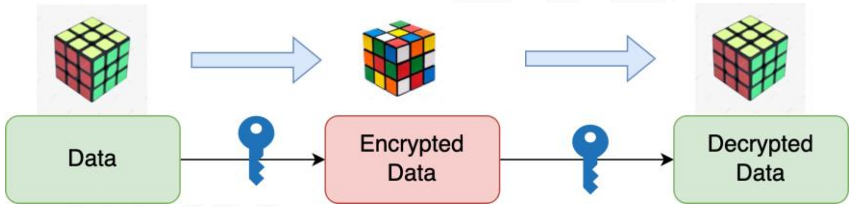

Game related data
Game state update
Game state update
GameStateUpdate
Players: [{id=0, pos=(101.12, 251.5, 1)}, {id=1, pos=(251.72, 711.5, 1.2)}, ...]
Game client
Game server
Packet is captured on the fly
Packet sniffer
Other computer in the same network
Packet data is fed to the cheat program
Custom cheat program that builds a world model based on the packet data

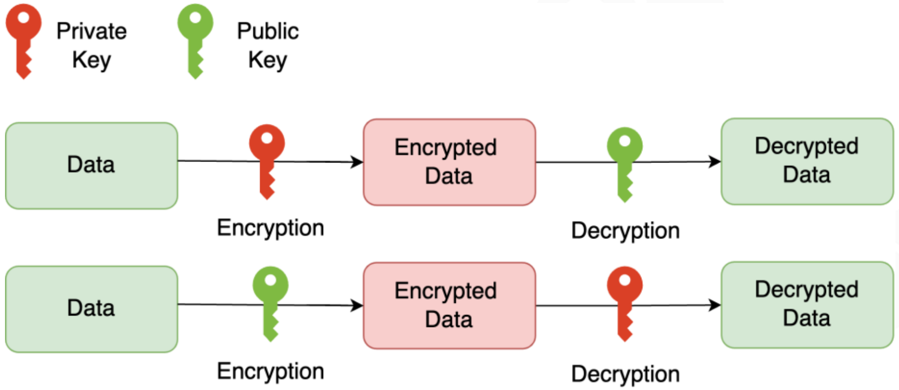
Encrypt the Network Traffic (1/2)

Two kinds of algorithms

- **Symmetric-key algorithm**
  - Obfuscate and restore data according to the same key
  - Fast and efficient

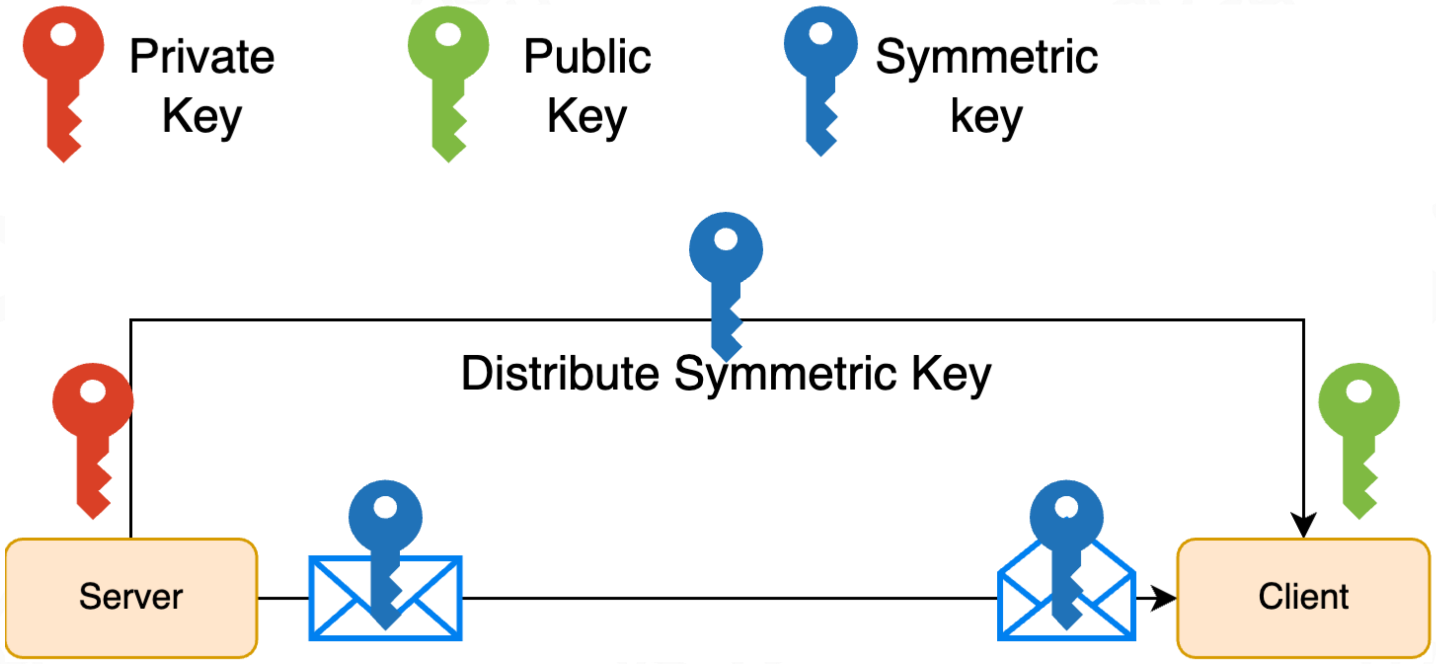
- **Asymmetric encryption**
  - Encryption and decryption use different keys
  - Slow, only used for encrypting critical data





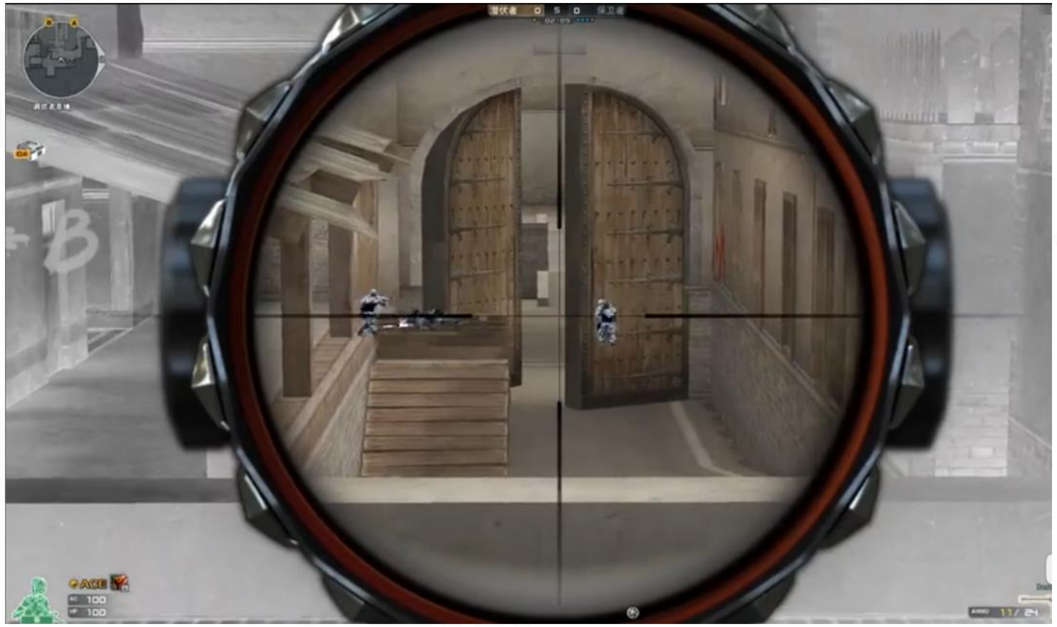
Encrypt the Network Traffic (2/2)

- Distribute symmetric key securely using asymmetric encryption
- Transfer data using symmetric encryption key



System Software Invoke

- Modify the DirectX kernel and change the execution flow of the rendering function
- Can force the rendering engine to modify the occlusion relationship
- See the movement of the enemy behind the wall



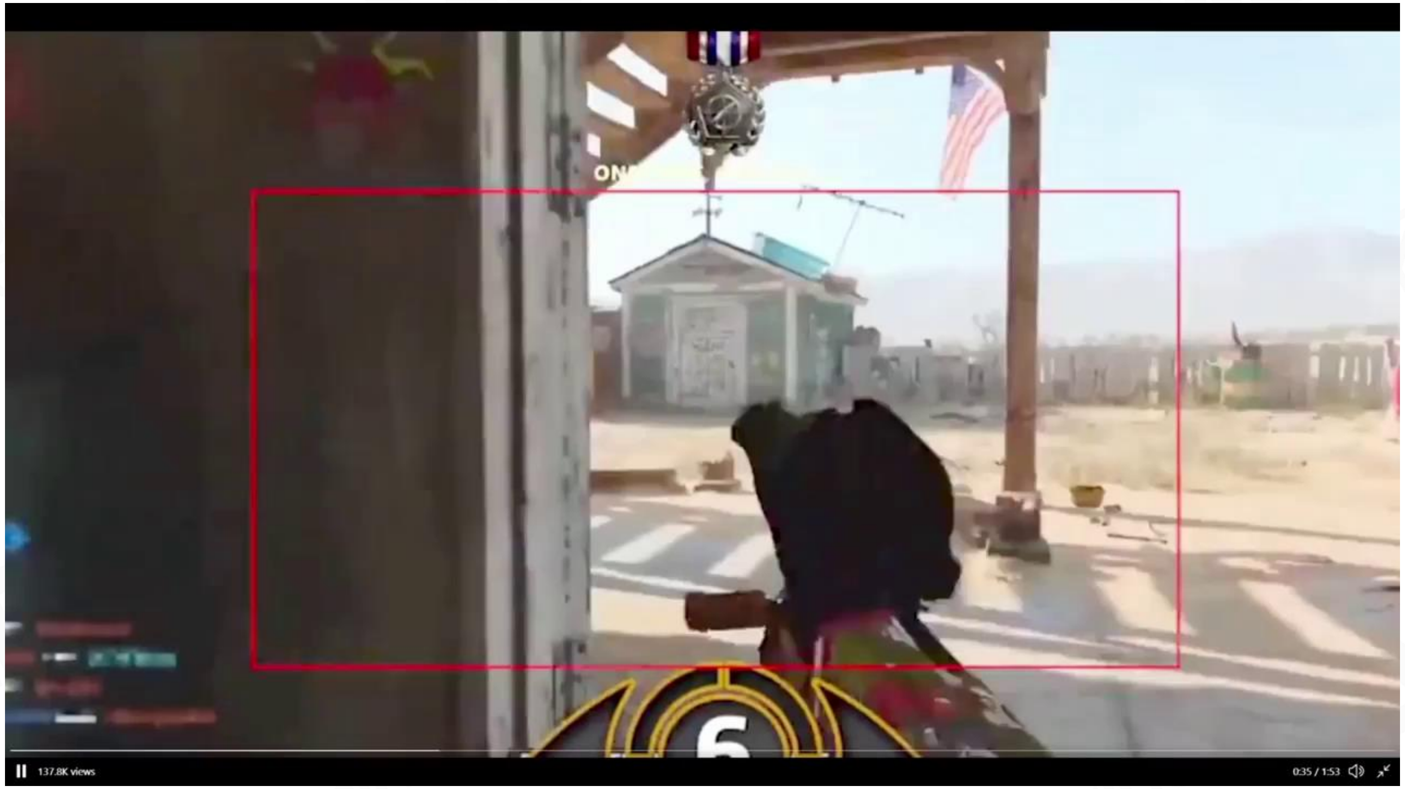
Valve Anti-Cheat and Easy Anti-Cheat

- Detects malicious behavior caused by any file conflicts while interacting with the game
- Stops the player from playing the game at all
- Prevents any illegal modifications and configuration changes that enable the use of exploits in a game

AI Cheat

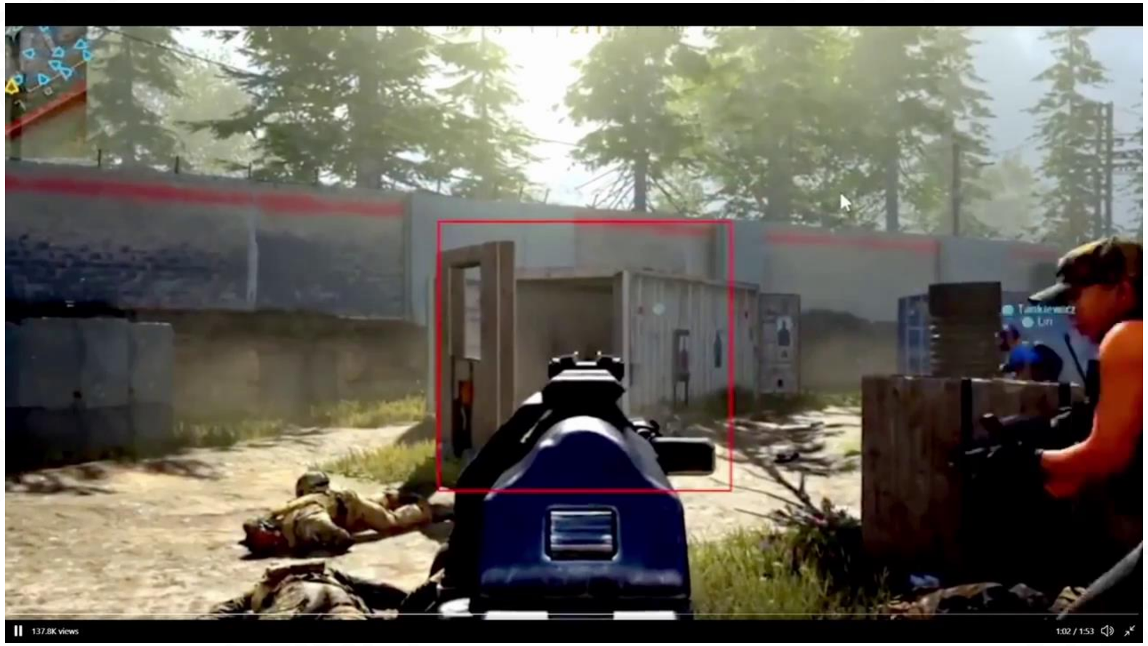
- All platforms
- No code modification required
- Independent from the game

- Game screen
- Target detection
- Move cursor
- Fire



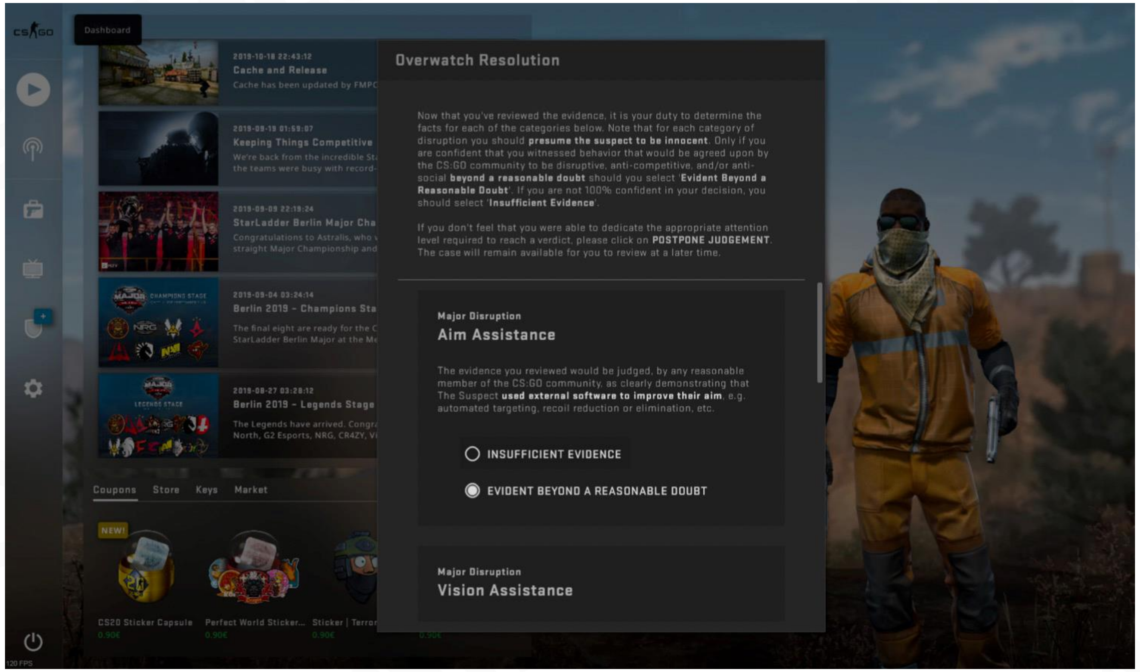
Rich AI Middlewares

- Real-Time Object Detection. YOLO V5, V7 ...
- Skeleton based Action recognition



Counter-Strike: Overwatch

- The system is based on other players reviewing footage from players that are suspected of cheating
- Many reviewers are looking at the same cases and the majority decide whether the suspect was cheating or not


CS:GO
Dashboard
Cache and Release
Keeping Things Competitive
StarLadder Berlin Major Cha
Berlin 2019 - Champions Sta
Berlin 2019 - Legends Stage
Coupons
Store
Keys
Market
NEW!
CS20 Sticker Capsule
Perfect World Sticker...

Overwatch Resolution

Now that you've reviewed the evidence, it is your duty to determine the facts for each of the categories below. Note that for each category of disruption you should **presume the suspect to be innocent**. Only if you are confident that you witnessed behavior that would be agreed upon by the CS:GO community to be disruptive, anti-competitive, and/or anti-social **beyond a reasonable doubt** should you select '**Evident Beyond a Reasonable Doubt**'. If you are not 100% confident in your decision, you should select '**Insufficient Evidence**'.

If you don't feel that you were able to dedicate the appropriate attention level required to reach a verdict, please click on **POSTPONE JUDGEMENT**. The case will remain available for you to review at a later time.

Major Disruption

Aim Assistance

The evidence you reviewed would be judged, by any reasonable member of the CS:GO community, as clearly demonstrating that The Suspect **used external software to improve their aim**, e.g. automated targeting, recoil reduction or elimination, etc.

○ INSUFFICIENT EVIDENCE

● EVIDENT BEYOND A REASONABLE DOUBT

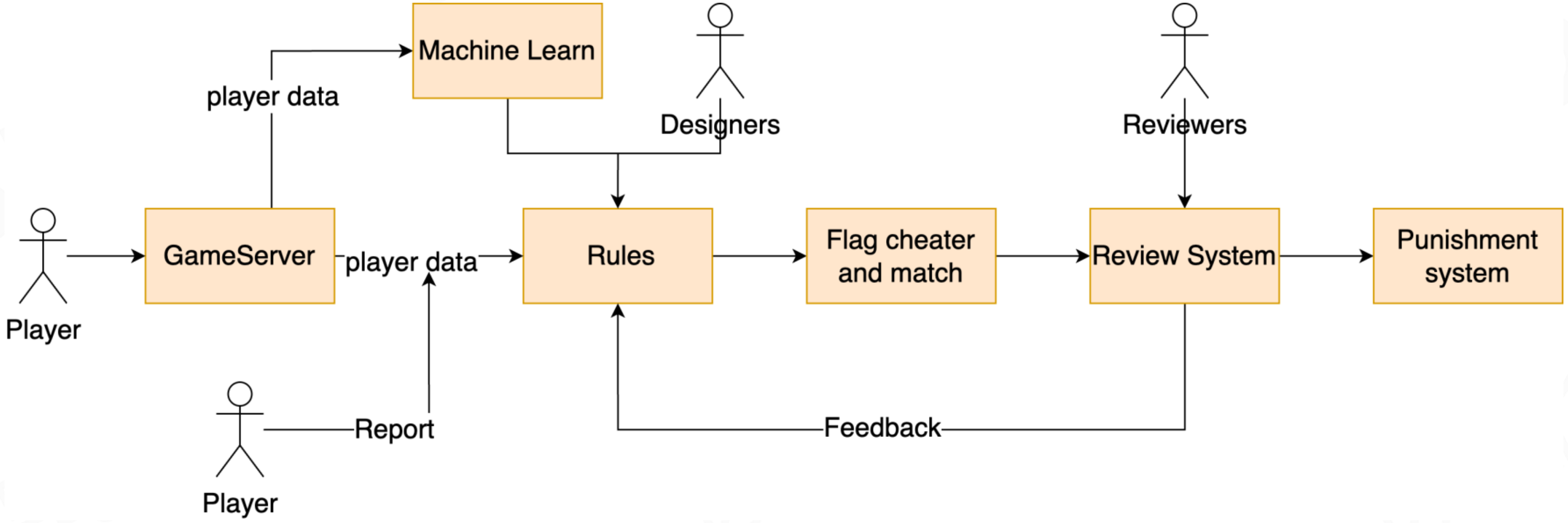
Major Disruption

Vision Assistance

Passing judgement after reviewing evidence in Counter Strike: Global Offensive's Overwatch system

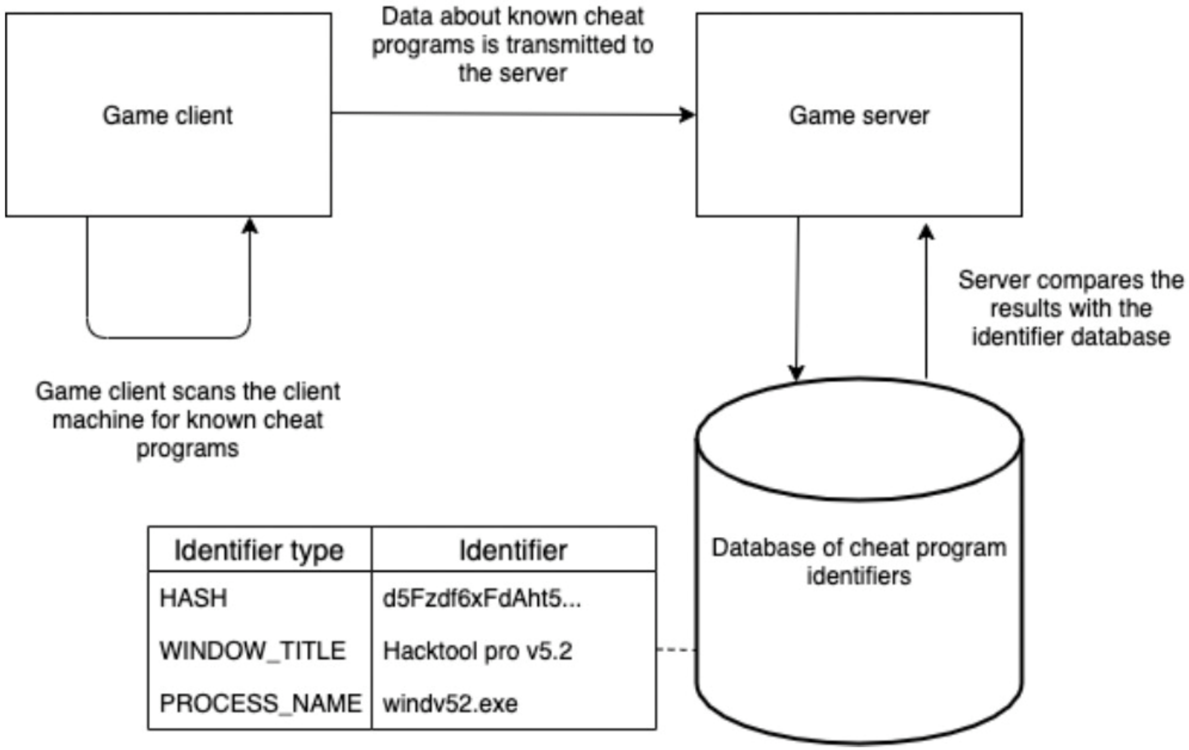
Statistic-based System

- Collect the user's game information, such as victory and critical hit rate
- Compare your own historical data and some thresholds rules or from other player's reports to mark players
- Check manually to confirm whether they cheat



Detecting Known Cheat Program

- A proper anti-cheat program should have a way to scan the user's computer for known cheating programs based on various signatures
- The simplest method can simply entail comparing hashes or process names



Game client

Data about known cheat programs is transmitted to the server

Game server

Game client scans the client machine for known cheat programs

Server compares the results with the identifier database

Database of cheat program identifiers

Identifier type	Identifier
HASH	d5Fzdf6xFdAht5...
WINDOW_TITLE	Hacktool pro v5.2
PROCESS_NAME	windv52.exe

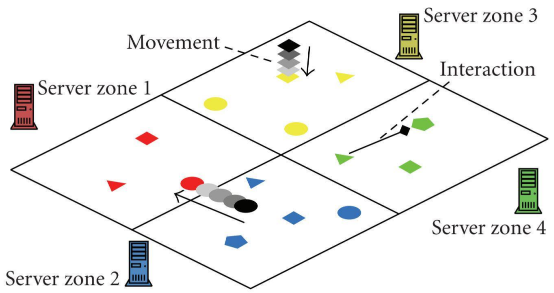
Example of identifier-based anti-cheat system

Build a Scalable World

Scalable Game Servers

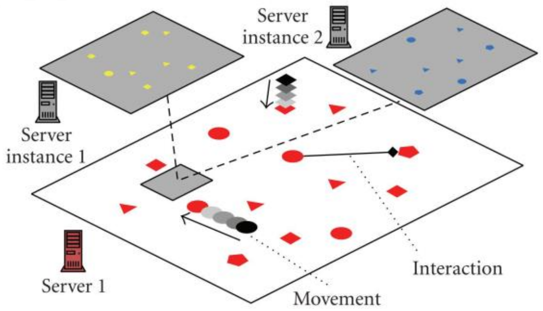
Zoning

- Distribute large player numbers in a large world
- Distribution might be uneven



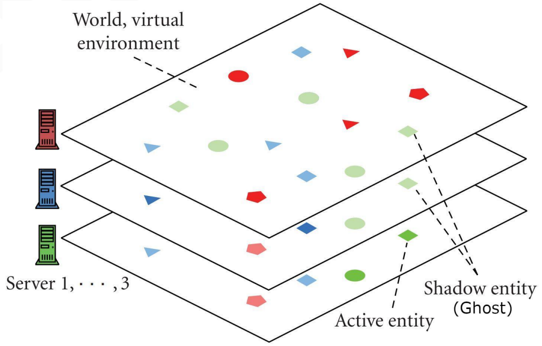
Instancing

- Run a large number of game areas independently in parallel
- Reduce congestion/competition



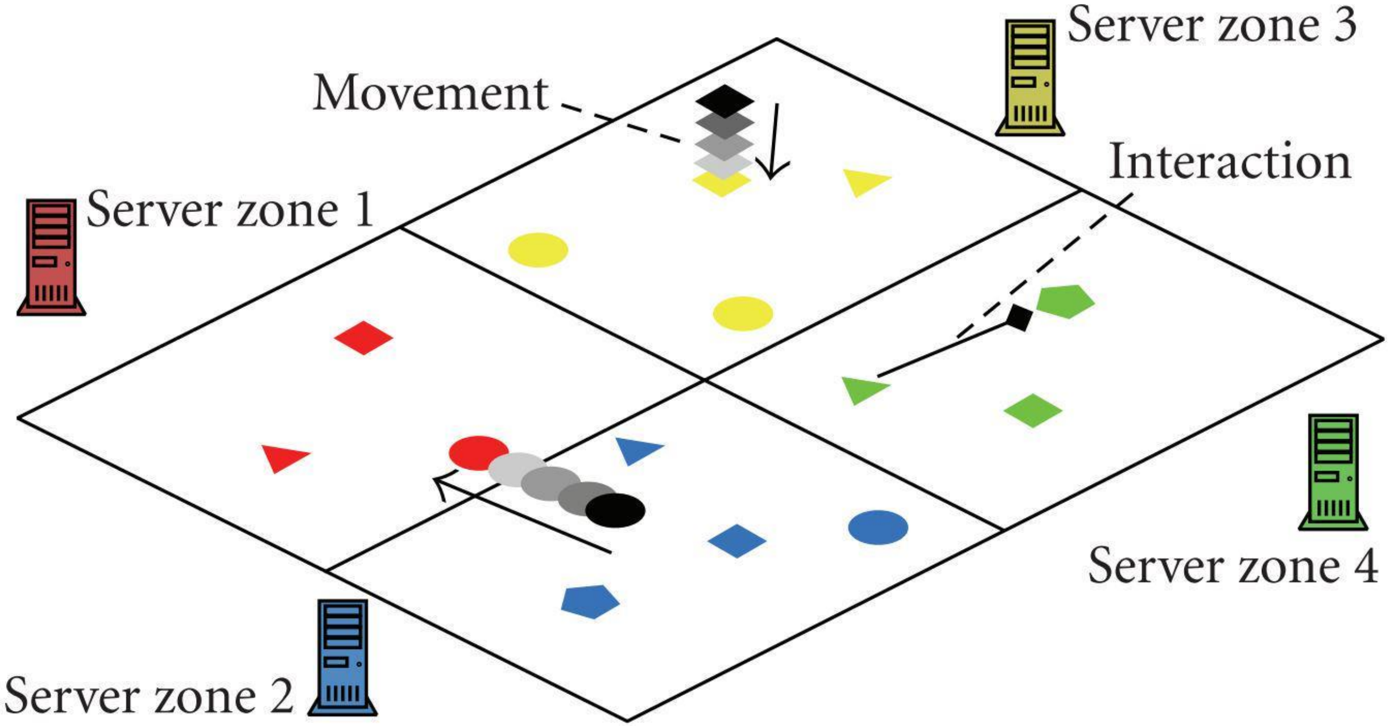
Replication

- Allows high user density
- E.g. high density PVP games



Zoning - Seamless Zones (1/4)

- Players are reasonably distributed in a large world
- The client only connects to one responsible server
- Cross border: auto transfer client to another server



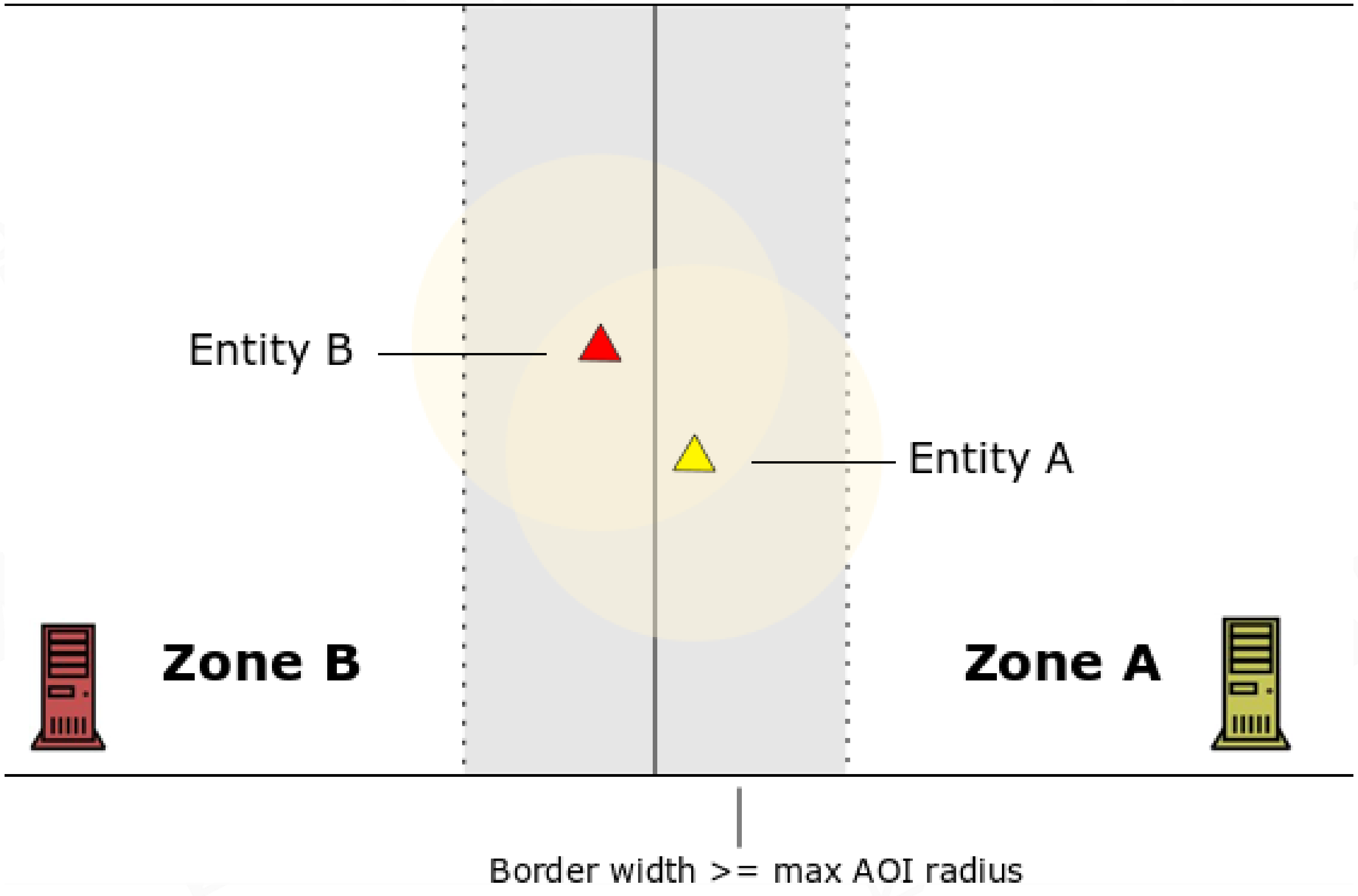
Zoning - Seamless Zones (2/4)

Zone Border

Smooth experience:

- Border width >= max AOI radius

But how to make them interact?



Zoning - Seamless Zones (3/4)

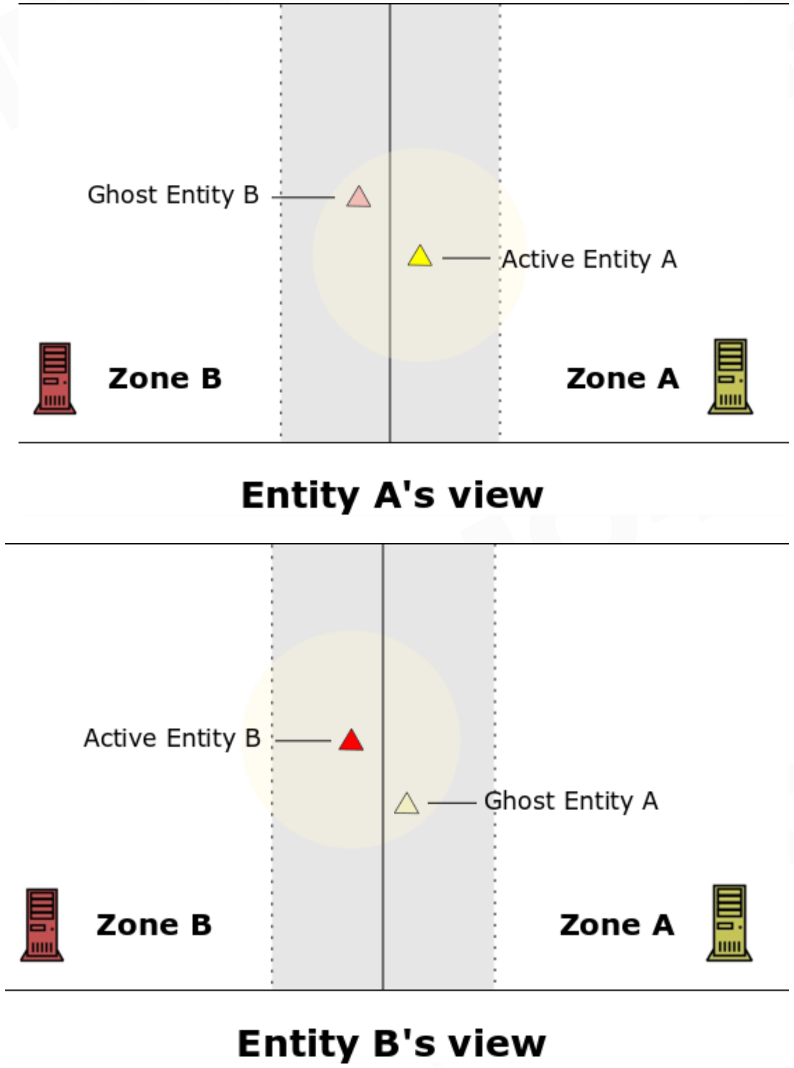
Zone Border - Entities

Active Entity

- Resides in connected zoned server (authority)
- Has a ghost agent in other zones
- Can see ghost entities in another zone

Ghost Entity

- Also called shadow entity
- Is an agent entity owned by another zone
- Receive updates from original entity



Zoning - Seamless Zones (4/4)

Cross Border: A -> B

① Before move

- An active entity in zone A

② Near boundary (A)

- Active in A; Ghost in B

③ At boundary

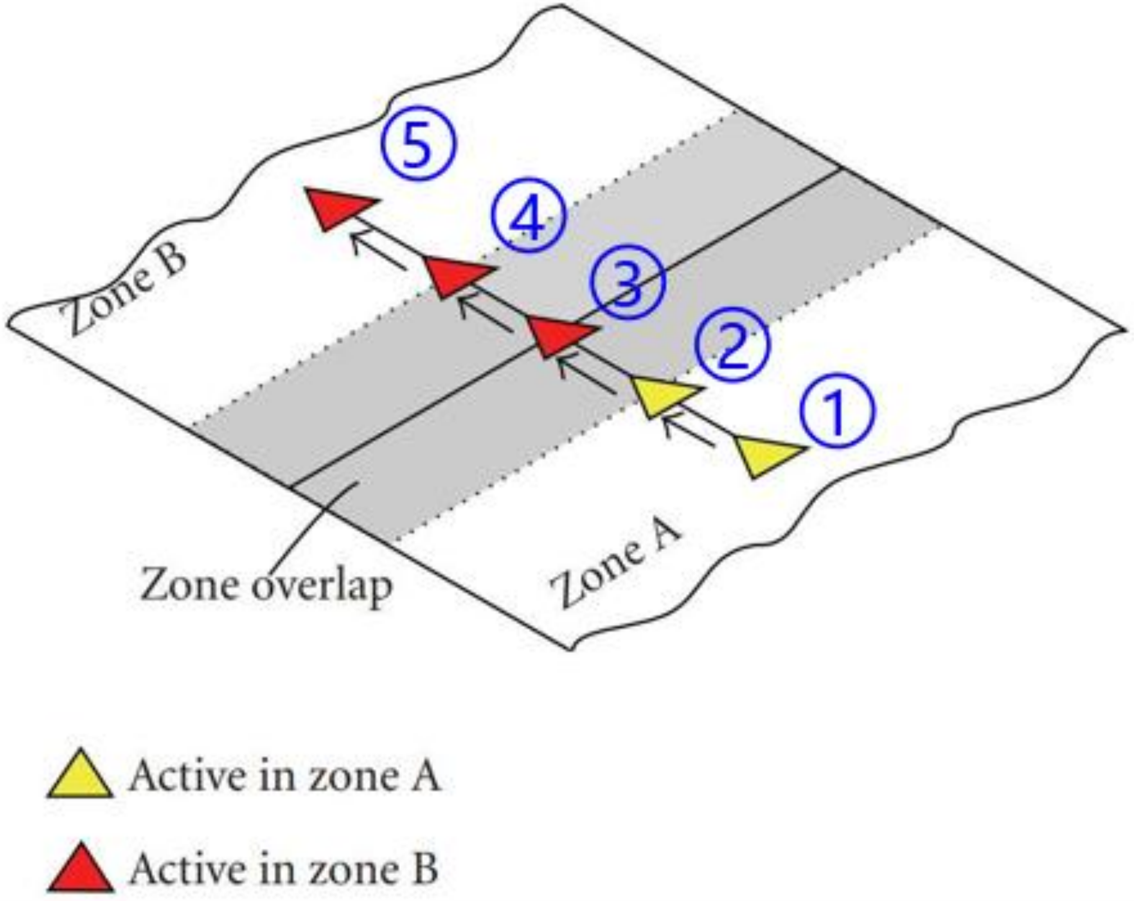
- The entity has been transferred to zone B

④ Near boundary (B)

- Active in B; Ghost in A

⑤ Beyond boundary (B)

- Removed from zone A

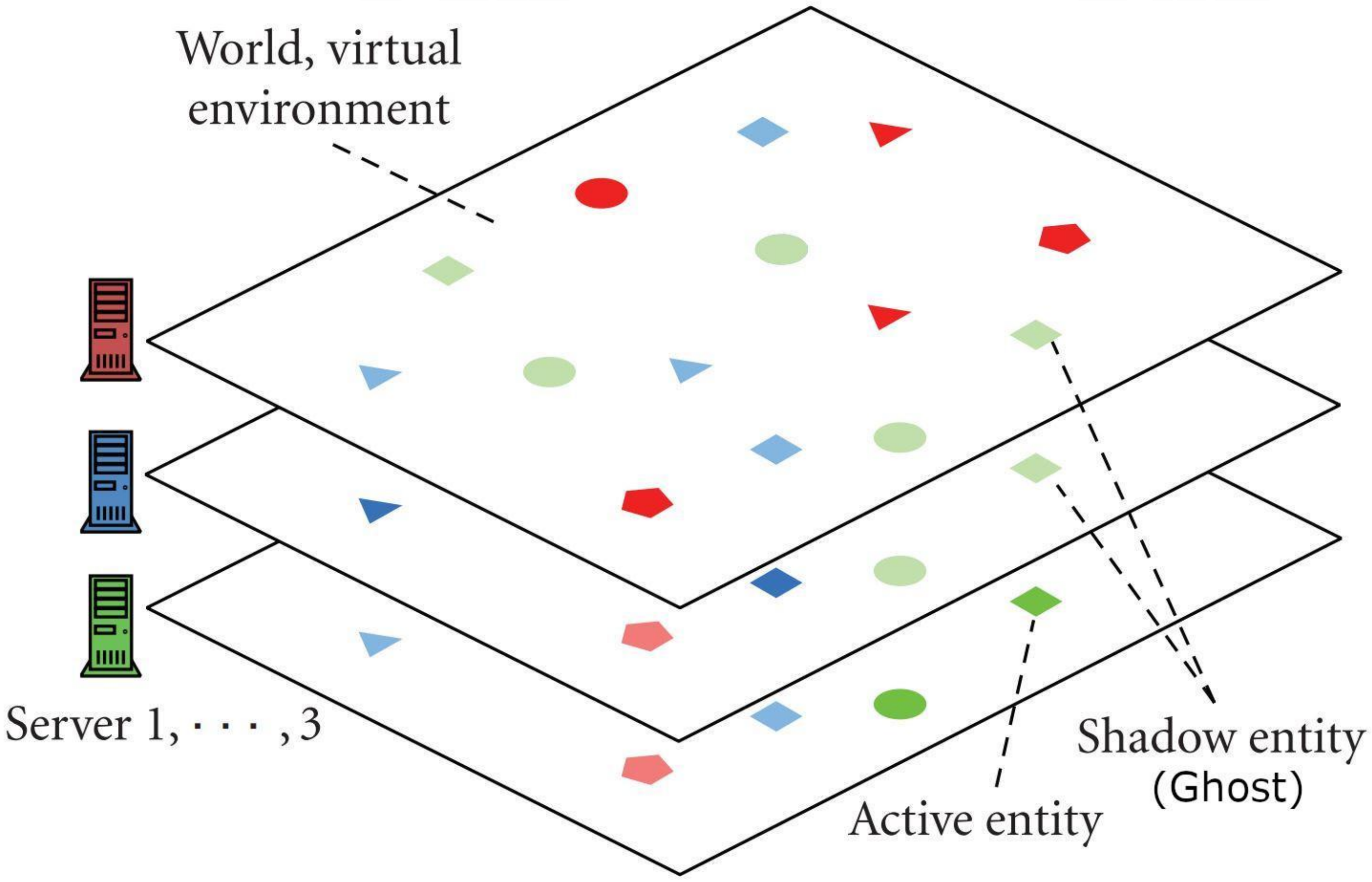


△ Active in zone A

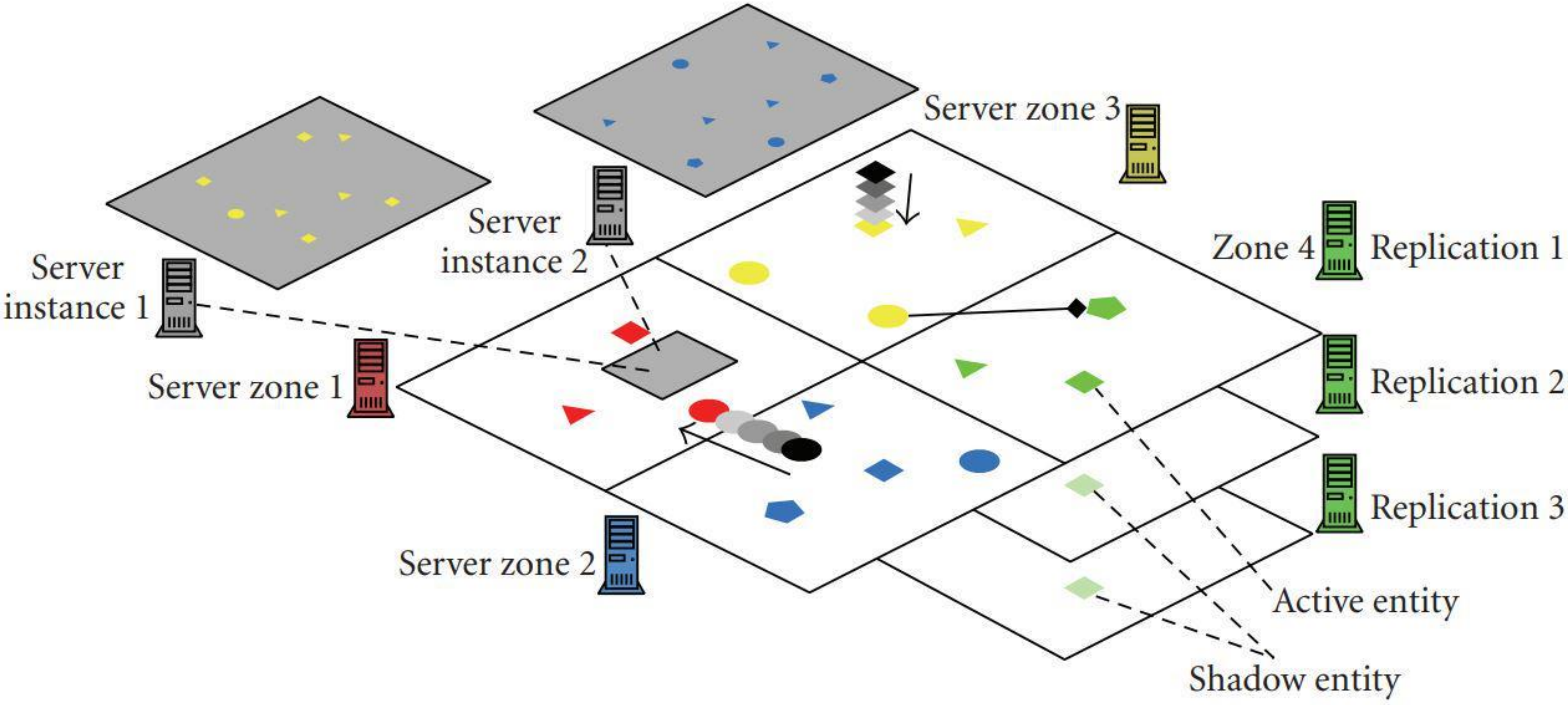
▲ Active in zone B

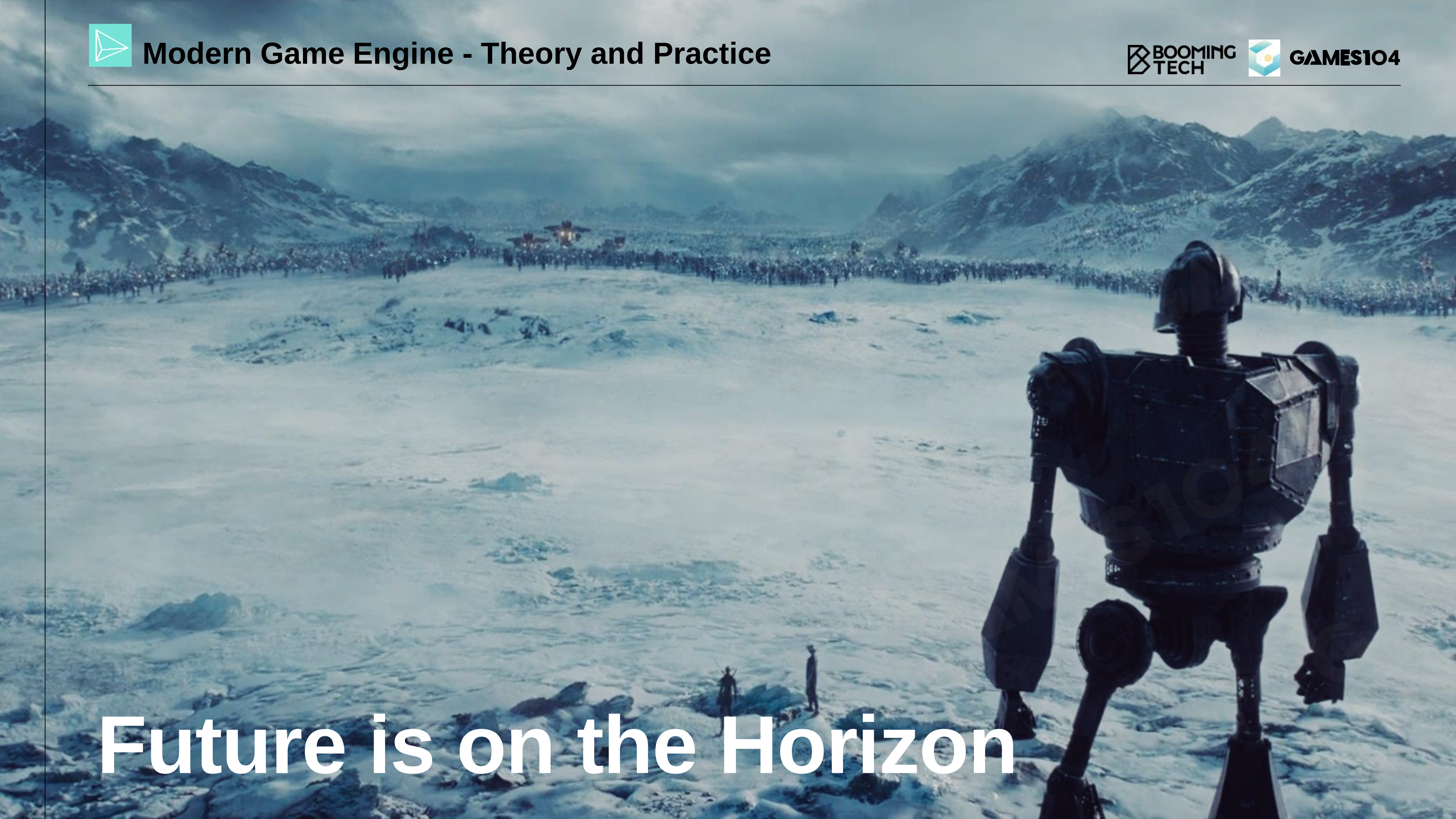
Replication

- Cooperatively process same world zone
- Entity updates are distributed among servers
- Each server creates its own active entities
- Updates to active entities will be auto replicated to all remaining servers (as Ghost)



Scalable Game Servers - Combination





Future is on the Horizon

Lecture 19 Contributor

- 德辉
- Peter
- Ximenes
- yf

- 鸭毛
- BOOK
- 伟哥
- Minjie

- 邓导
- 阿鹏
- 凯哥
- 喵小君

- 大喷
- 爵爷
- Jason

References

Replicate Character Movement

- Replicating Chaos: Vehicle Replication in 'Watch Dogs 2', Matt Delbosc, Ubisoft Toronto, GDC 2017: https://www.gdcvault.com/play/1024597/Replicating-Chaos-Vehicle-Replication-in
- "Believable dead reckoning for networked games.", Murphy, Curtiss, and E. Lengyel, Game Engine Gems 2 (2011) : 307-328. https://www.researchgate.net/publication/293809946_Believable_Dead_Reckoning_for_Networked_Games
- Client-side Interpolation: https://docs-multiplayer.unity3d.com/netcode/0.1.0/learn/clientside_interpolation/index.html
- Source Multiplayer Networking: https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking

Lag Mitigation

- Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization, Yahn W. Bernier: https://www.gamedevs.org/uploads/latency-compensation-in-client-server-protocols.pdf
- Valorant's netcode: https://technology.riotgames.com/news/peeking-valorants-netcode
- Source Multiplayer Networking, valve developer community: https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking
- Implementation and Evaluation of Hit Registration in Networked First Person Shooters, Jonathan Lundgren: https://liu.diva-portal.org/smash/get/diva2:1605200/FULLTEXT01.pdf
- How It Works: Lag compensation and Interp in CS:GO: https://www.youtube.com/watch?v=6EwaW2iz4iA&ab_channel=DevinDTV

MMOG Network Architecture (1/2)

- Intro to Databases in Games: How to Use Them in Games and Game Development - AWS Online Tech Talks: https://www.youtube.com/watch?v=7HppNxu_hdA
- Massively multiplayer online game, WIKI: https://en.wikipedia.org/wiki/Massively_multiplayer_online_game
- Consistent Hashing and Random Trees:
- Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web: https://dl.acm.org/doi/pdf/10.1145/258533.258660
- Consistent hashing: https://en.wikipedia.org/wiki/Consistent_hashing
- Fowler–Noll–Vo hash function: https://en.wikipedia.org/wiki/Fowler%E2%80%93Noll%E2%80%93Vo_hash_function

MMOG Network Architecture (2/2)

- Performance Testing Methodology:

  http://hosteddocs.ittoolbox.com/questnolg22106java.pdf
- Glinka, F. *et al.* (2008) 'High-Level Development of Multiserver Online Games', *International Journal of Computer Games Technology*, *2008*(327387). Available at:

  https://downloads.hindawi.com/journals/ijcgt/2008/327387.pdf
- Alibaba Cloud MMO Gaming Solution Architecture:

  https://www.alibabacloud.com/blog/593877

Bandwidth Optimization

- Replication in network games: Bandwidth (Part 4): https://0fps.net/2014/03/09/replication-in-network-games-bandwidth-part-4/
- Introducing Time Dilation: https://www.eveonline.com/news/view/introducing-time-dilation-tidi
- Glazer, J. and Madhav, S. (2016) *Multiplayer Game Programming: Architecting Networked Games.* New York: Addison-Wesley
- Managing and Mining Multiplayer Online Games for the Summer Semester 2017: https://www.dbs.ifi.lmu.de/Lehre/mmmo/sose17/slides/MMMO-2-Core_1.pdf
- Handling large amounts of players: https://mmo-blueprint.com/handling-large-amounts-of-players/?i=1
- Zinx应用-MMO游戏案例: https://github.com/aceld/zinx
- KBEngine: https://github.com/kbengine/kbengine

Anti-Cheat (1/2)

- Cheating in online games, WIKI: https://en.wikipedia.org/wiki/Cheating_in_online_games
- The Day FPS Games Died, 2021.07: https://www.youtube.com/watch?v=revk5r5vqxA
- 警惕AI外挂, LYi, 2021.08: https://www.bilibili.com/video/BV1Lq4y1M7E2
- YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao, 2022.07: https://arxiv.org/abs/2207.02696

Anti-Cheat (2/2)

- MarI/O - Machine Learning for Video Games, SethBling: https://www.youtube.com/watch?v=qv6UVOQ0F44
- Comparative Study of Anti-cheat Methods in Video Games, Samuli Lehtonen, 2020.3: https://helda.helsinki.fi/bitstream/handle/10138/313587/Anti_cheat_for_video_games_final_07_03_2020.pdf
- 游戏外挂攻防艺术, 徐胜, 2013
- Python Plays Grand Theft Auto 5 - Reboot, Harrison: https://github.com/Sentdex/pygta5

Q&A

Enjoy;)
Coding

Course Wechat

Please follow us for
further information

Citation (1/2)

- Maze War (1978) - First Person Shooter History - Ep01:

  https://www.youtube.com/watch?v=ZjY5s_05Qlw
- How It Works: Lag compensation and Interp in CS:GO:

  https://www.youtube.com/watch?v=6EwaW2iz4iA&ab_channel=DevinDTV
- World of Warcraft (2021) - Gameplay (PC UHD) [4K60FPS]:

  https://www.youtube.com/watch?v=d26Kl98_qo8
- Guild Wars 2’s greatest STRENGTH is also its WEAKNESS:

  https://www.youtube.com/watch?v=jGcQDhSPLHM
- FINAL FANTASY XIV: https://jp.finalfantasyxiv.com
- WHY PLAY EVE ONLINE in 2022? - And why this is one of my favorite games:

  https://www.youtube.com/watch?v=j2t_XPDOIFc

Citation (2/2)

- World of Warcraft: Cataclysm Login Screen: https://www.youtube.com/watch?v=isCsAEwpIOY
- 10 Best MMOs To Play In 2022: https://www.youtube.com/watch?v=W0rvIv6diVU
- Planetside 2: https://www.planetside2.com/home
- Planetside 2: Things Just Got Harder: https://www.youtube.com/watch?v=HTN-yc5x0pA
- 15 Years of WoW vs 1 Year of FFXIV: https://www.youtube.com/watch?v=5T-mcLYBnKc
- MUD1 (1978) - First MUD game: https://www.youtube.com/watch?v=9Gep3LwLKWk