

Bringing Replays to World of Tanks: Mercenaries

Andrew Glover

Lead Software Engineer

Wargaming Sydney



Infrastructure



Game Integration



In Development



Future Challenges





Enjoying the
game you just
dominated

Sharing
games with
friends

Learning from
your matches

Creating
custom videos
for YouTube

Watching your
favourite E-
sports team

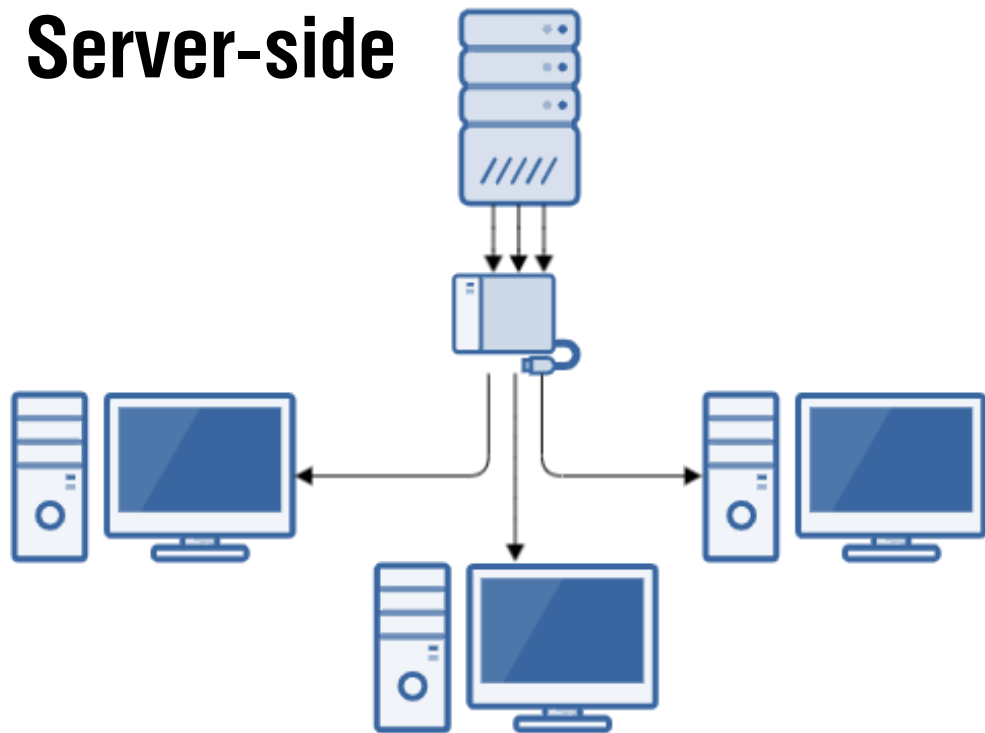
Reliable
Debugging

Automated
Testing

Machine
Learning

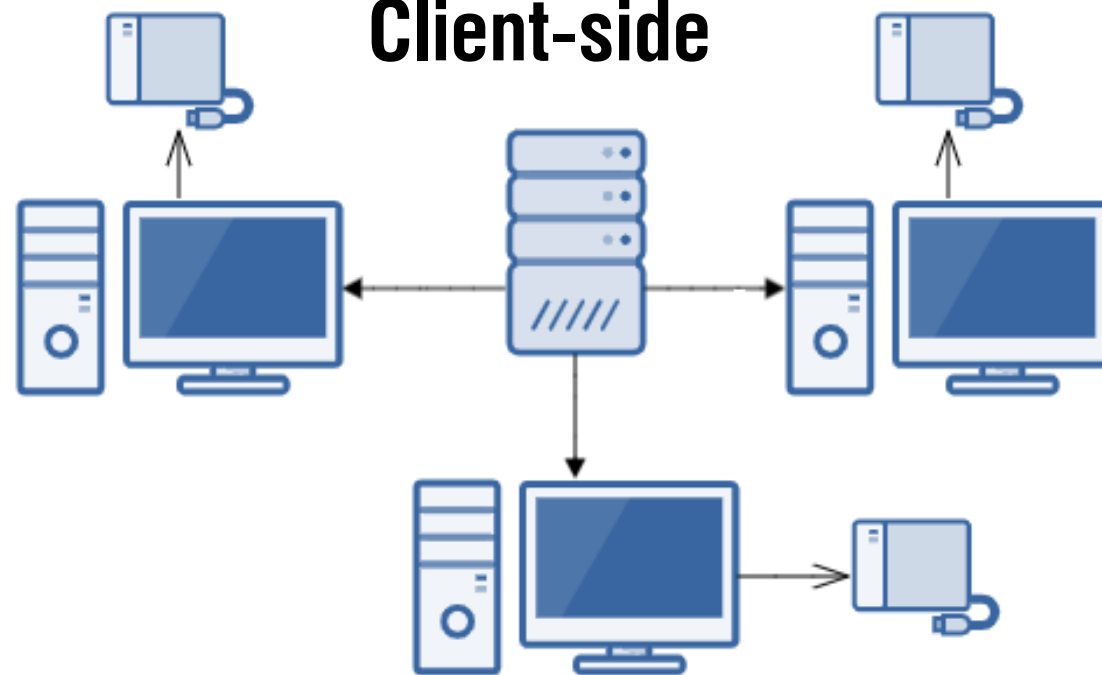
Infrastructure

Server-side

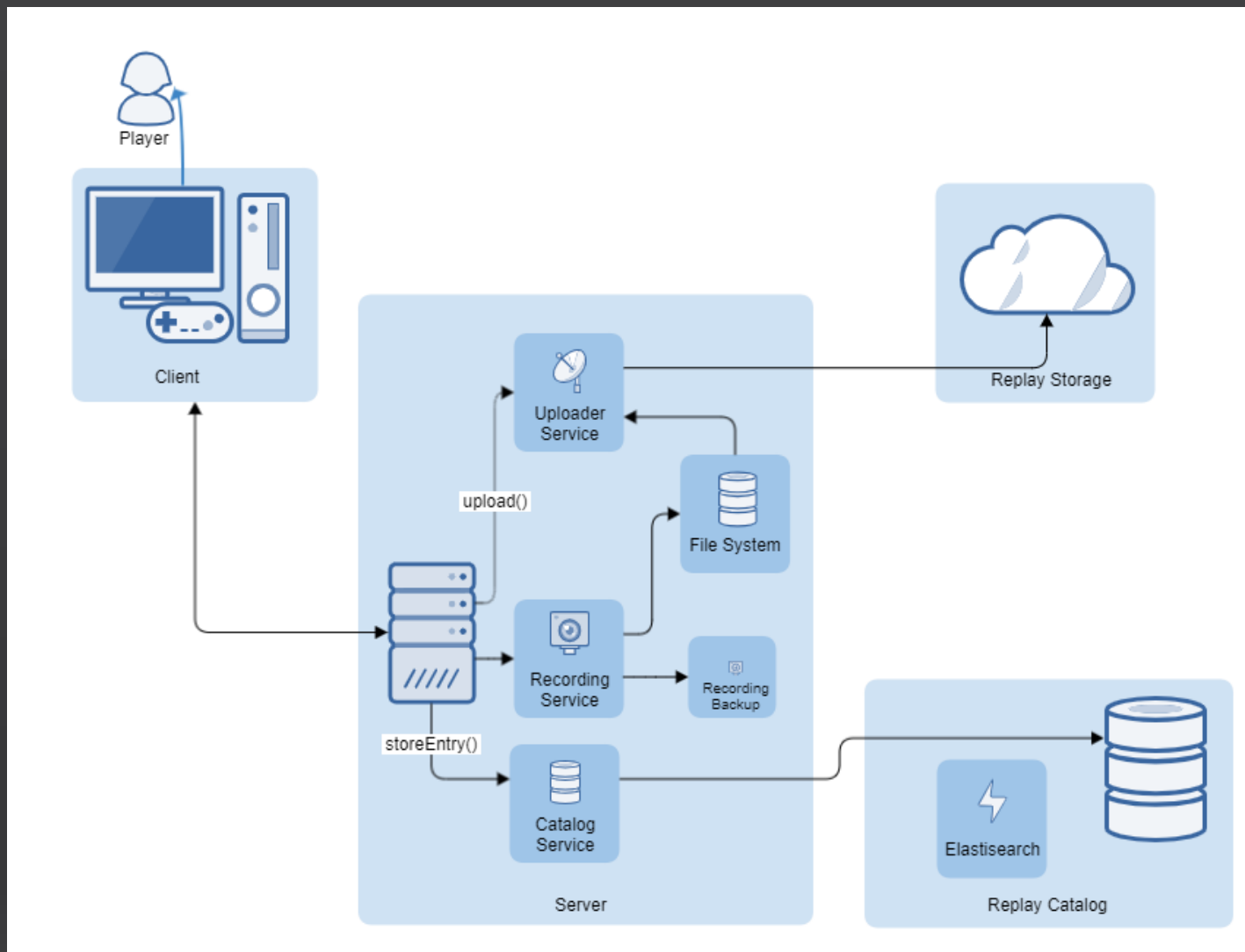


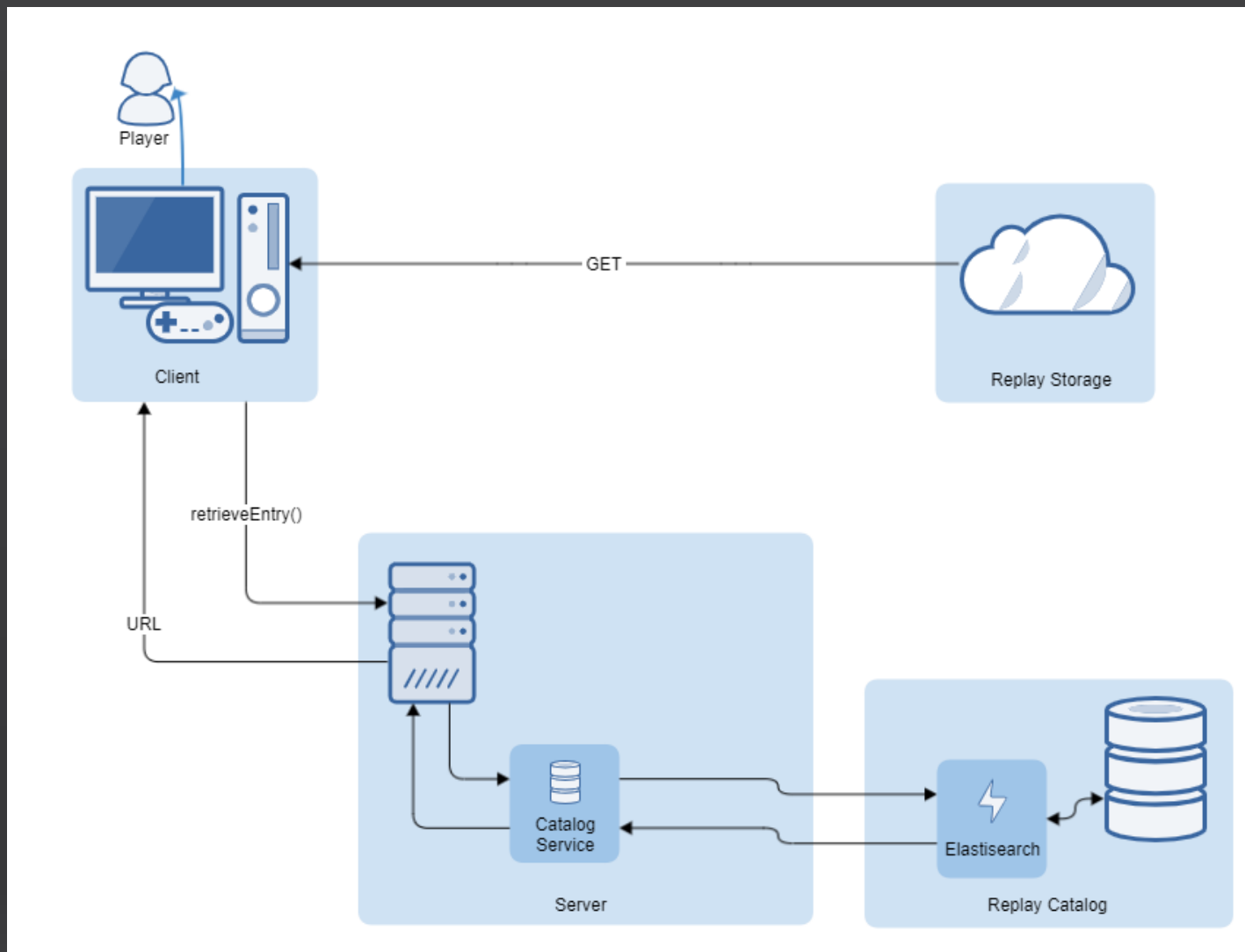
- Record everything in the match
- A single recording per match

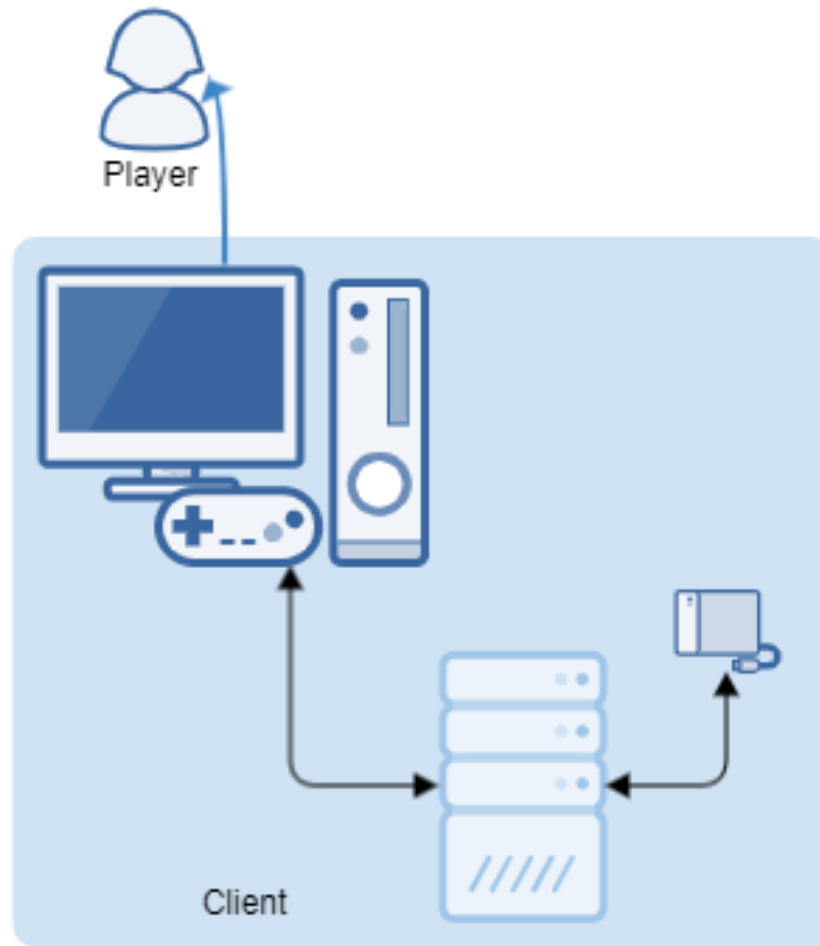
Client-side



- Exact replica of player viewpoint
- Replay file is always local







Game Integration

Browsing
Replays

Pause, Speed
up, Slow down

Jump
Forwards and
Backwards

Free, Attached
and Sniper
Cameras

Player Vision
(AoI)

Client UI

Highlights and
Highlight Only
Mode

Feature parity
on PS4, Xbox
One and 360

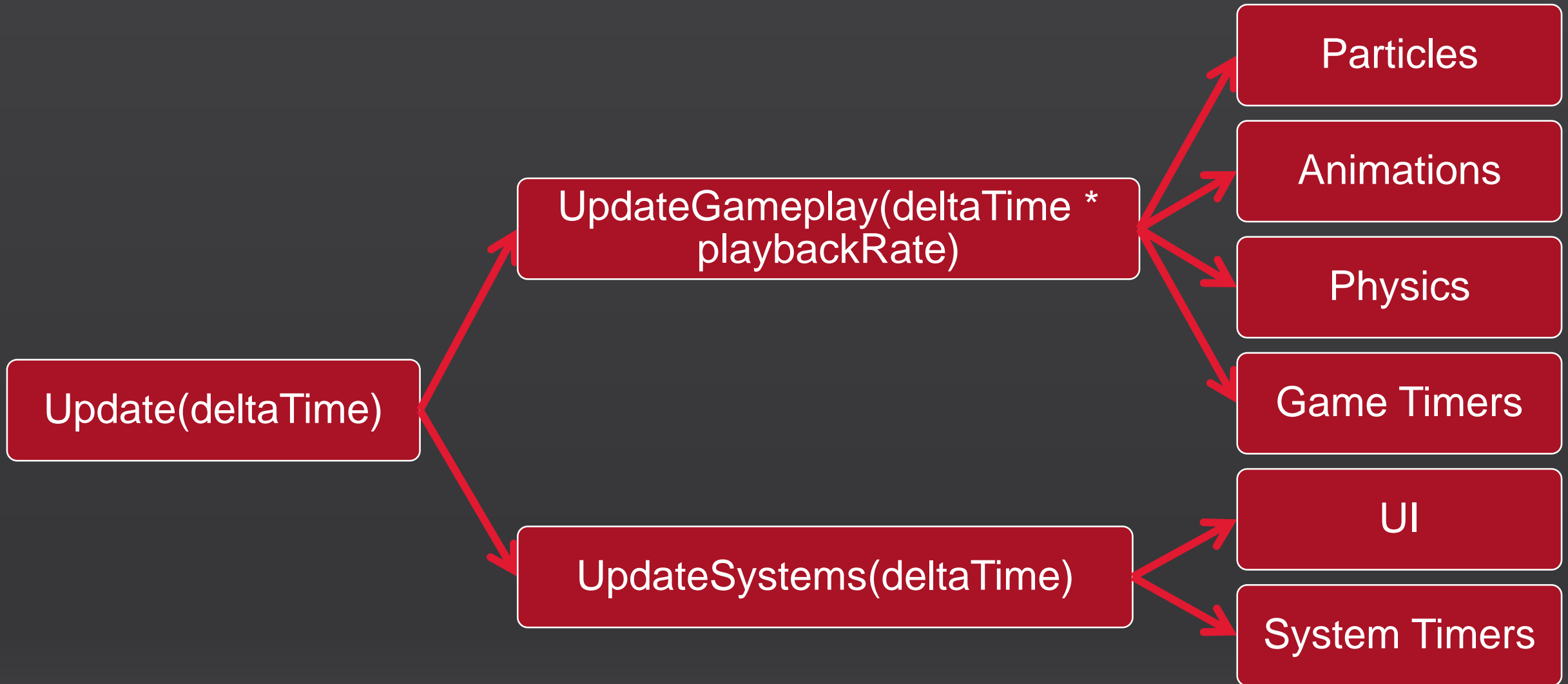




Problem: Everything pauses



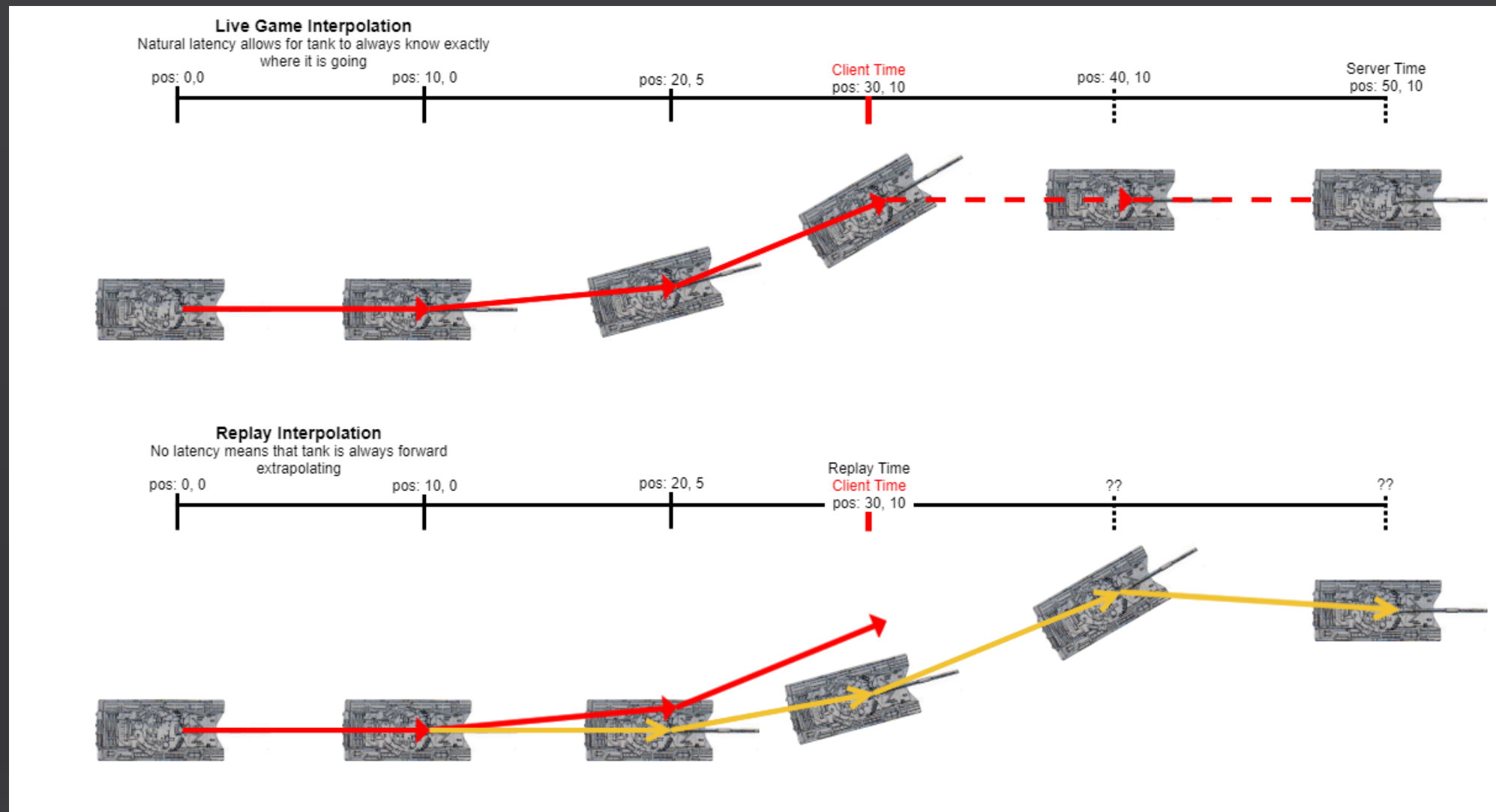
Solution: Split update paths



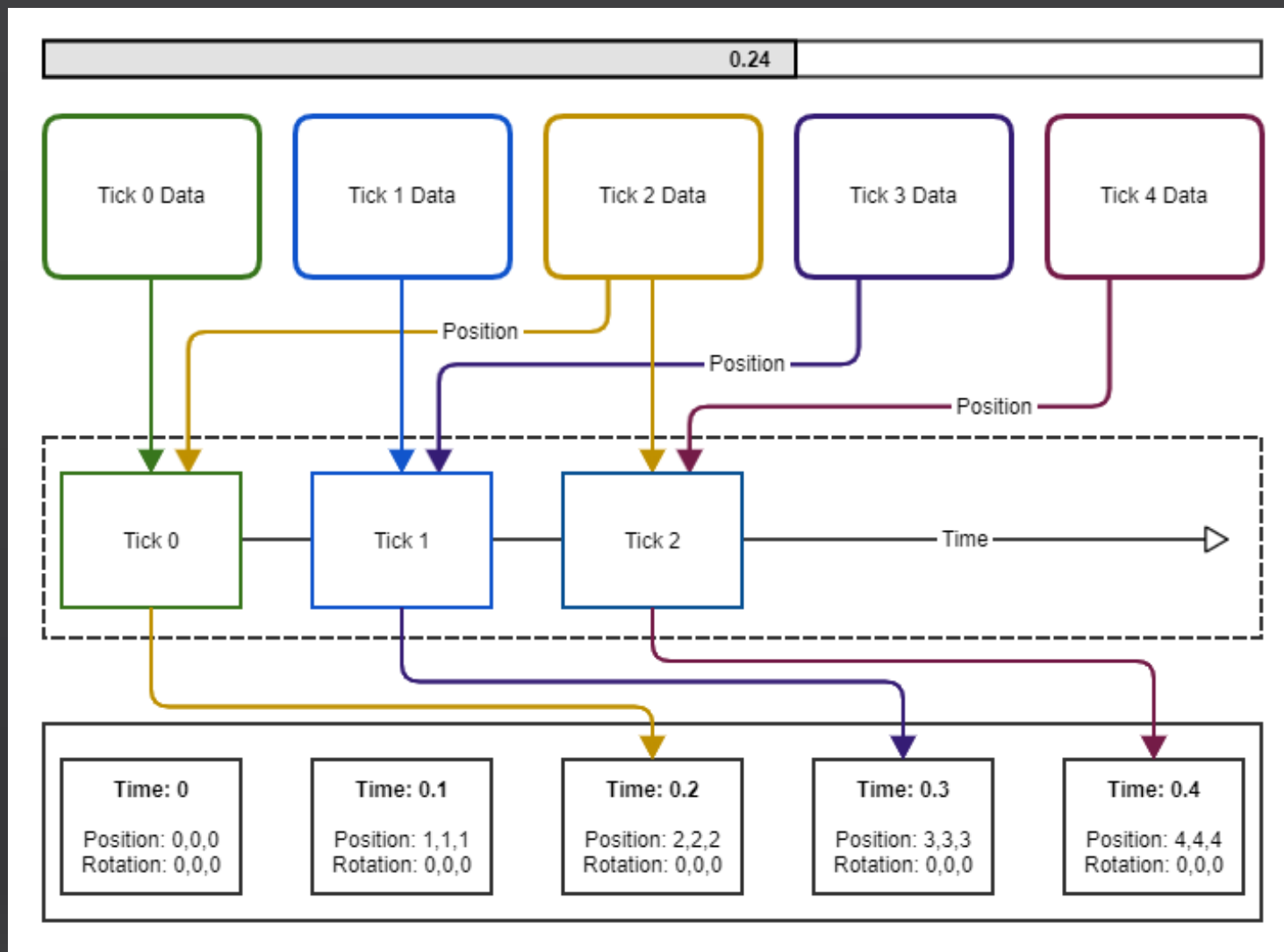
Problem: Sliding entities



Problem: Sliding entities



Solution: Movement interpolation seeding



Problem: Jumping Backwards



Jump Backwards

Reset

Jump
Forwards

Unload World

Playback to 0

Loop through
skipped frames

Keep Asset Resources Loaded

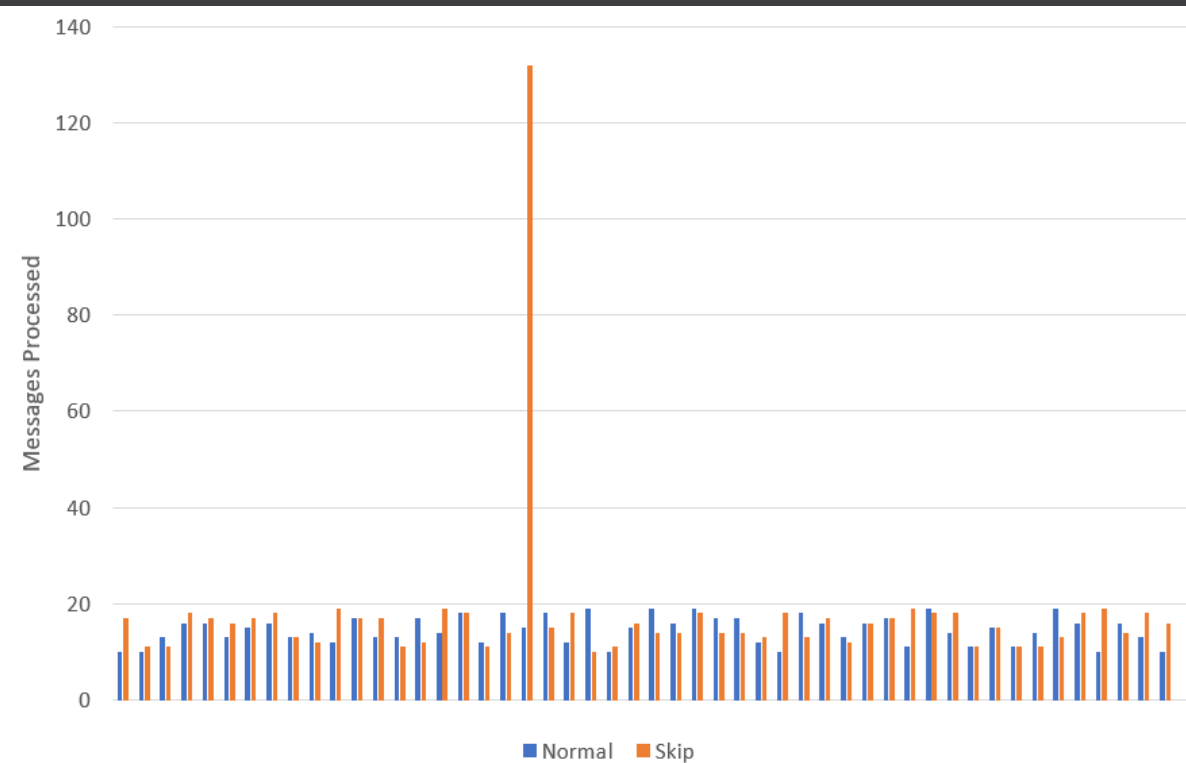
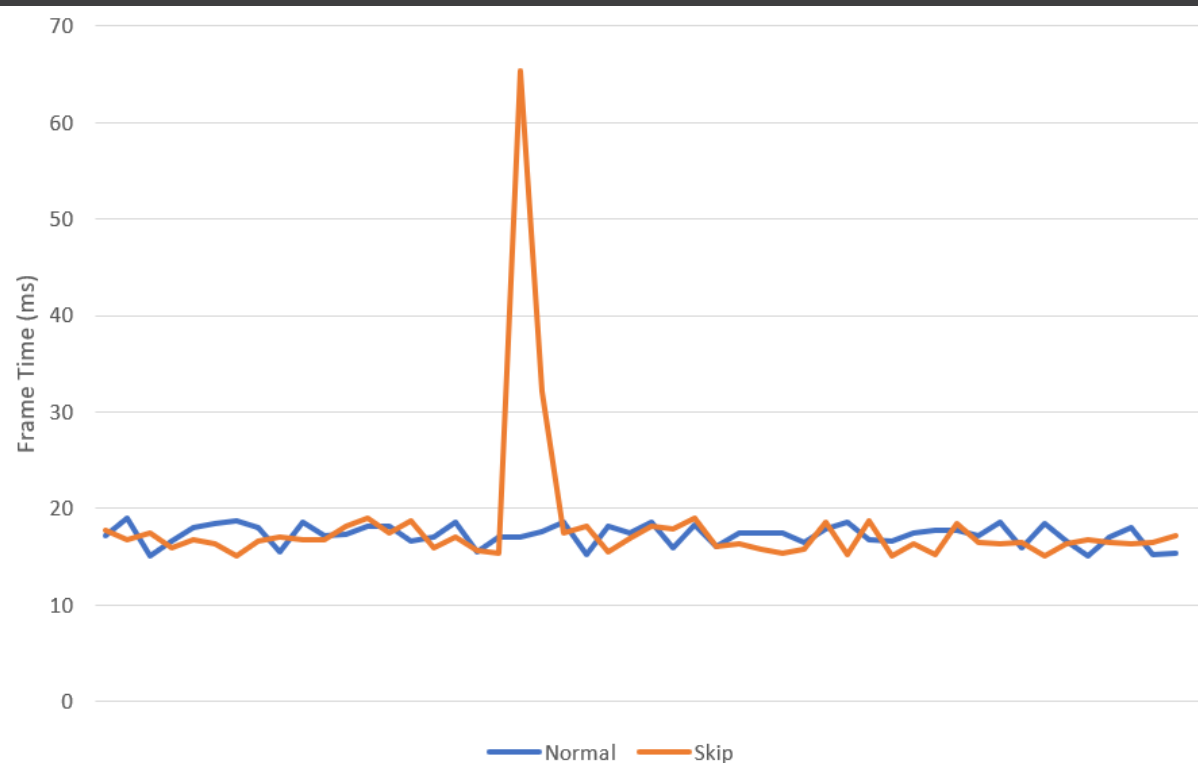
World Instancing

- Keep a fresh copy of your world in memory
- Copy over the existing one on reset

Clear Persistent Effects

- Decals
- Particles
- Music
- Animations

Problem: Jump forwards spike



Drop All

- Anything that doesn't impact the game beyond a momentary visual
- Shot Tracers

Keep Last

- Anything that is its own discreet update
- Properties like health and speed

Always

- Anything that has to play as future event rely on it
- Persistent Effects like decals

Original

[03:05.20]	(Entity1)	Position = (200, 10, 300)
[03:05.20]	(Entity1)	ShotFired()
[03:05.20]	(Entity2)	Position = (200, 10, 150)
[03:05.30]	(Entity2)	Health = 30
[03:05.30]	(Entity1)	AddScore(5)
[03:05.30]	(Entity1)	Position = (200, 10, 310)
[03:05.30]	(Entity2)	Position = (210, 10, 145)
[03:05.40]	(Entity1)	Position = (200, 10, 320)
[03:05.40]	(Entity2)	Position = (215, 10, 155)
[03:05.50]	(Entity1)	ShotFired()
[03:05.60]	(Entity2)	Health = 20
[03:05.60]	(Entity2)	SetDamaged(ENGINE)
[03:05.60]	(Entity1)	AddScore(5)
[03:05.70]	(Entity1)	Position = (205, 10, 330)
[03:05.70]	(Entity2)	Position = (220, 10, 160)
[03:05.70]	(Entity1)	ShotFired()
[03:05.80]	(Entity2)	Health = 10
[03:05.80]	(Entity1)	Position = (210, 10, 330)
[03:05.80]	(Entity2)	Position = (220, 10, 180)

Drop All

[03:05.20]	(Entity1)	Position = (200, 10, 300)
[03:05.20]	(Entity1)	ShotFired()
[03:05.20]	(Entity2)	Position = (200, 10, 150)
[03:05.30]	(Entity2)	Health = 30
[03:05.30]	(Entity1)	AddScore(5)
[03:05.30]	(Entity1)	Position = (200, 10, 310)
[03:05.30]	(Entity2)	Position = (210, 10, 145)
[03:05.40]	(Entity1)	Position = (200, 10, 320)
[03:05.40]	(Entity2)	Position = (215, 10, 155)
[03:05.50]	(Entity1)	ShotFired()
[03:05.60]	(Entity2)	Health = 20
[03:05.60]	(Entity2)	SetDamaged(ENGINE)
[03:05.60]	(Entity1)	AddScore(5)
[03:05.70]	(Entity1)	Position = (205, 10, 330)
[03:05.70]	(Entity2)	Position = (220, 10, 160)
[03:05.70]	(Entity1)	ShotFired()
[03:05.80]	(Entity2)	Health = 10
[03:05.80]	(Entity1)	Position = (210, 10, 330)
[03:05.80]	(Entity2)	Position = (220, 10, 180)

Keep Last

[03:05.20]	(Entity1)	Position = (200, 10, 300)
[03:05.20]	(Entity2)	Position = (200, 10, 150)
[03:05.30]	(Entity2)	Health = 30
[03:05.30]	(Entity1)	AddScore(5)
[03:05.30]	(Entity1)	Position = (200, 10, 310)
[03:05.30]	(Entity2)	Position = (210, 10, 145)
[03:05.40]	(Entity1)	Position = (200, 10, 320)
[03:05.40]	(Entity2)	Position = (215, 10, 155)
[03:05.60]	(Entity2)	Health = 20
[03:05.60]	(Entity2)	SetDamaged(ENGINE)
[03:05.60]	(Entity1)	AddScore(5)
[03:05.70]	(Entity1)	Position = (205, 10, 330)
[03:05.70]	(Entity2)	Position = (220, 10, 160)
[03:05.80]	(Entity2)	Health = 10
[03:05.80]	(Entity1)	Position = (210, 10, 330)
[03:05.80]	(Entity2)	Position = (220, 10, 180)

Always

[03:05.30] (Entity1) AddScore(5)

[03:05.60] (Entity2) SetDamaged(ENGINE)

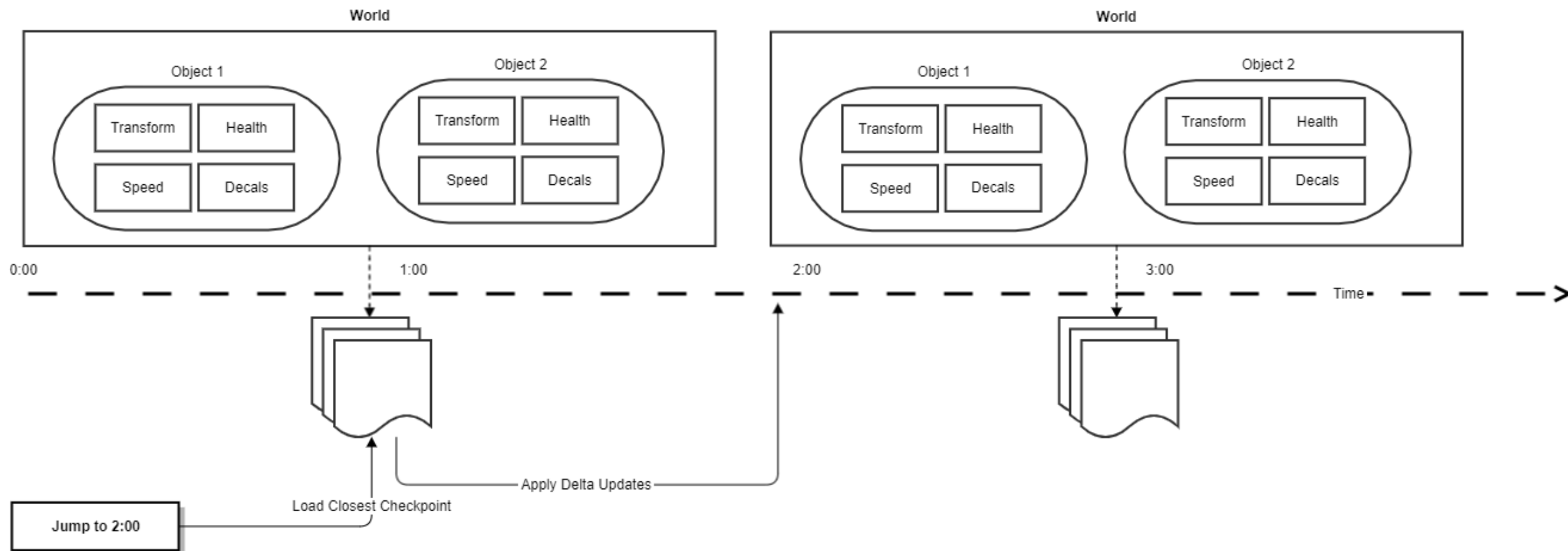
[03:05.60] (Entity1) AddScore(5)

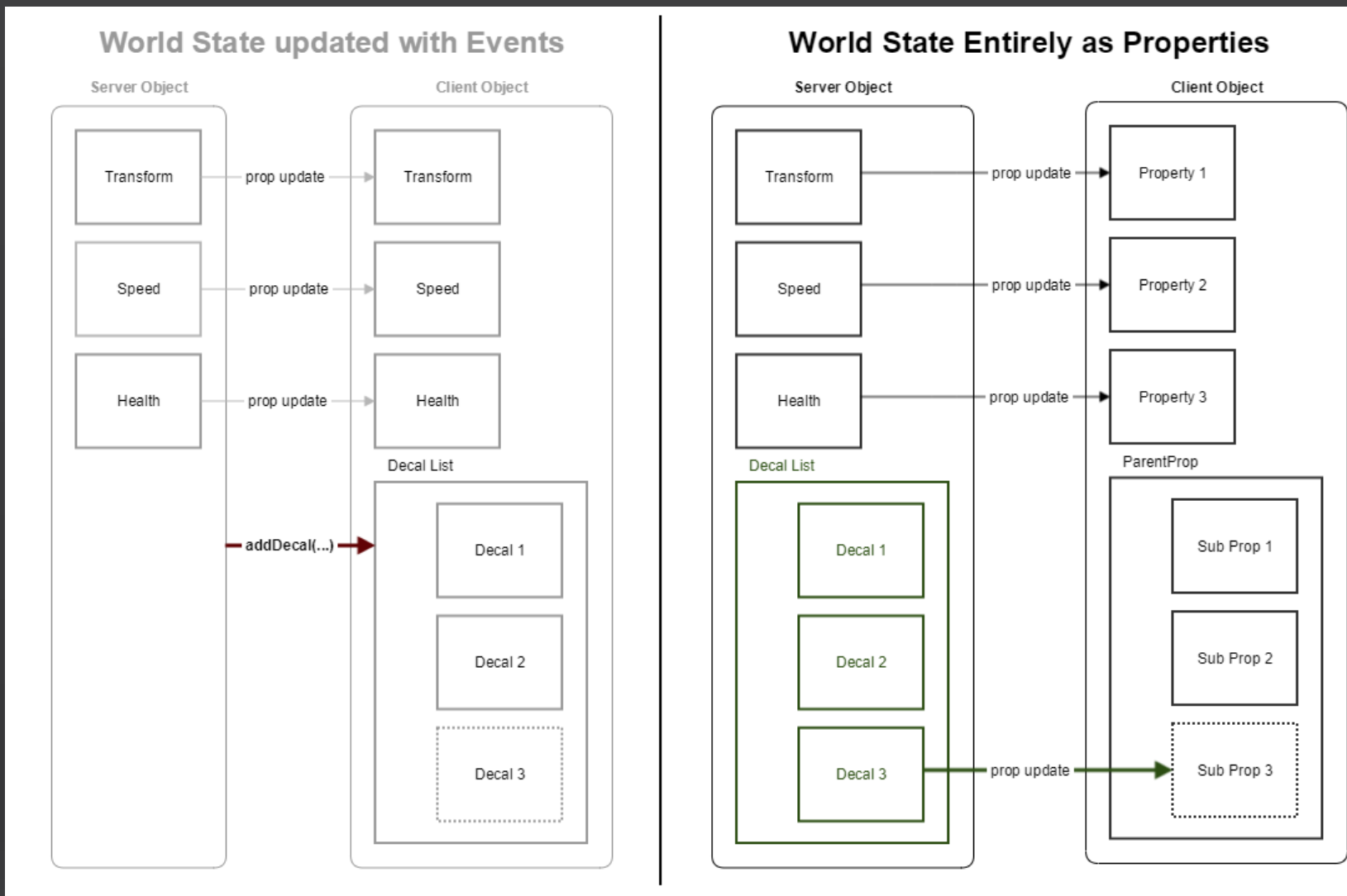
[03:05.80] (Entity2) Health = 10

[03:05.80] (Entity1) Position = (210, 10, 330)

[03:05.80] (Entity2) Position = (220, 10, 180)

Ideal Solution: Checkpointing





Problem: Everything skipped happens at once



Animation

Effects

- Shots
- Hits
- Power-ups

HUD

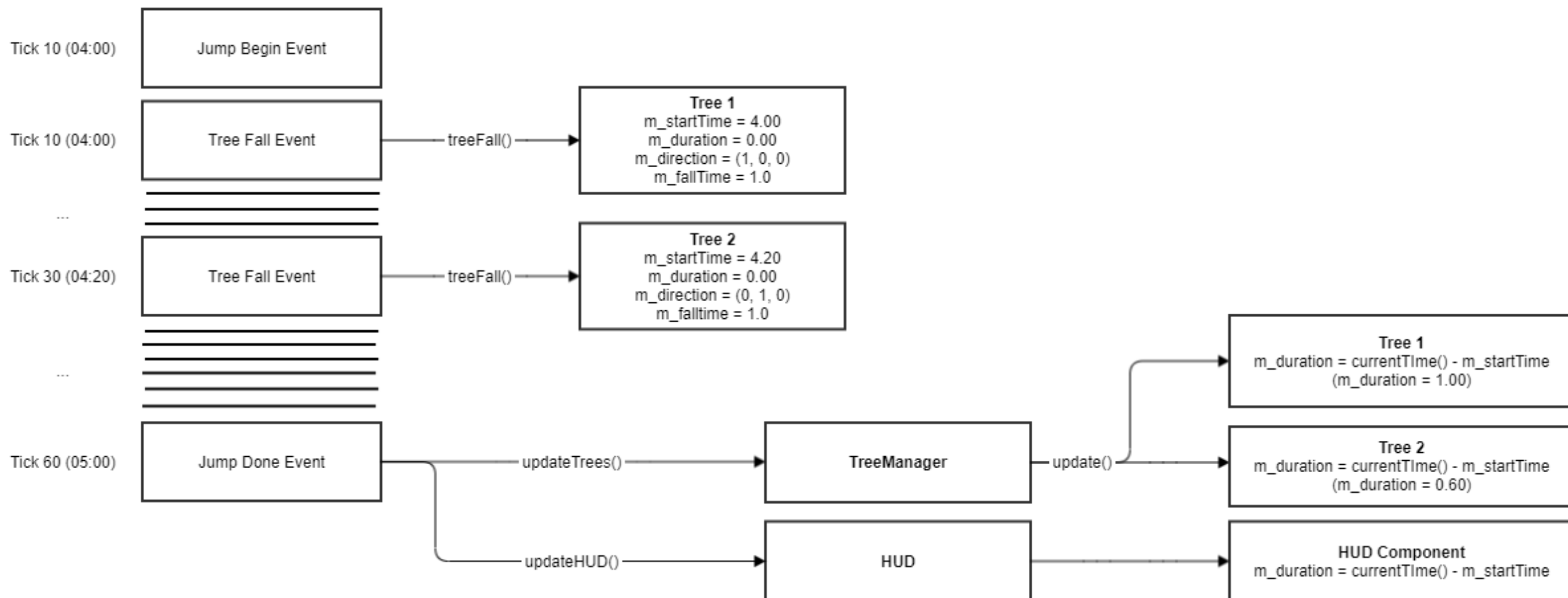
- Damage Indicators
- Event Logs
- Minimap

Destruction

Initialization

- Spawning
- Resupply

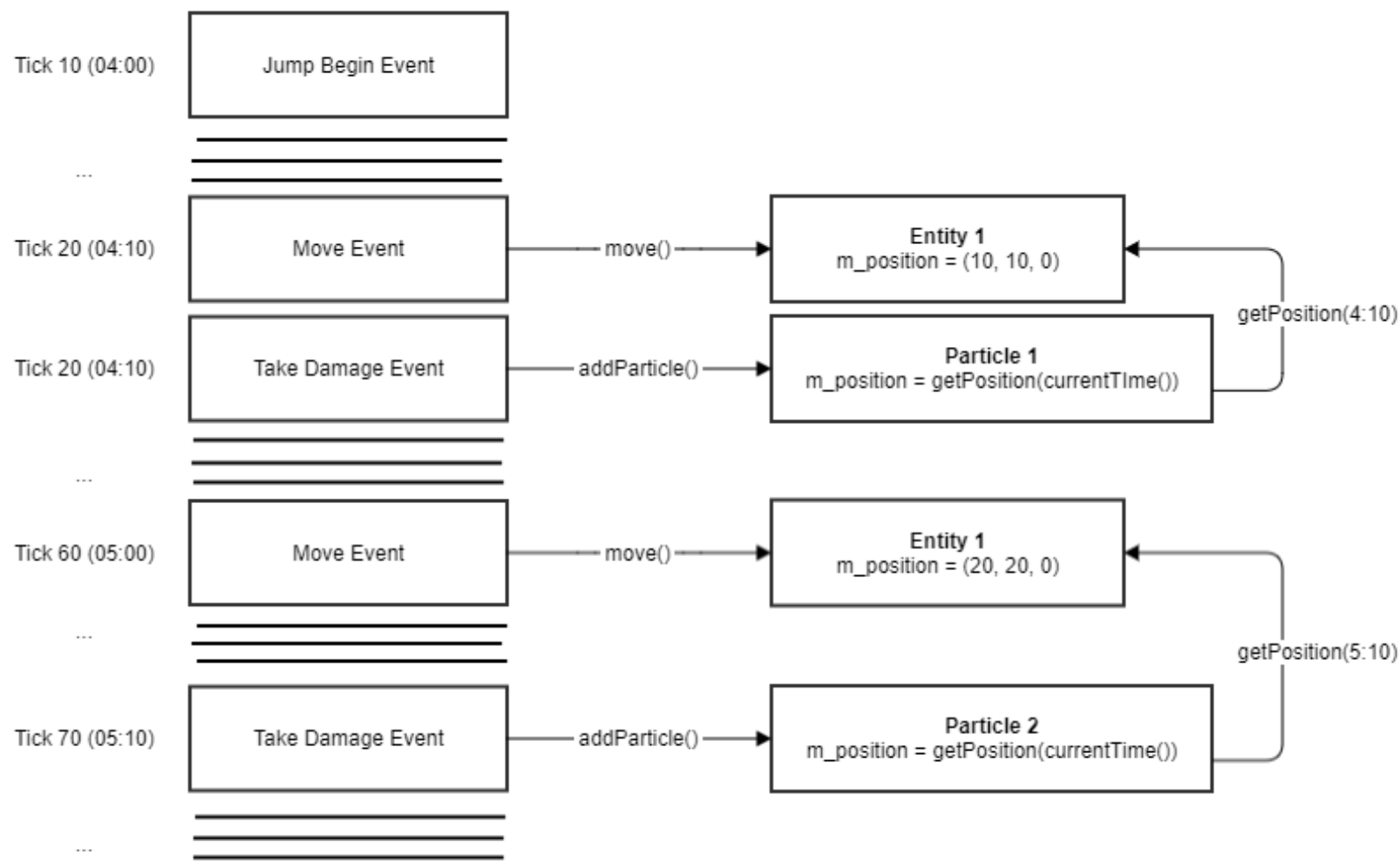
Solution: Store start times for events



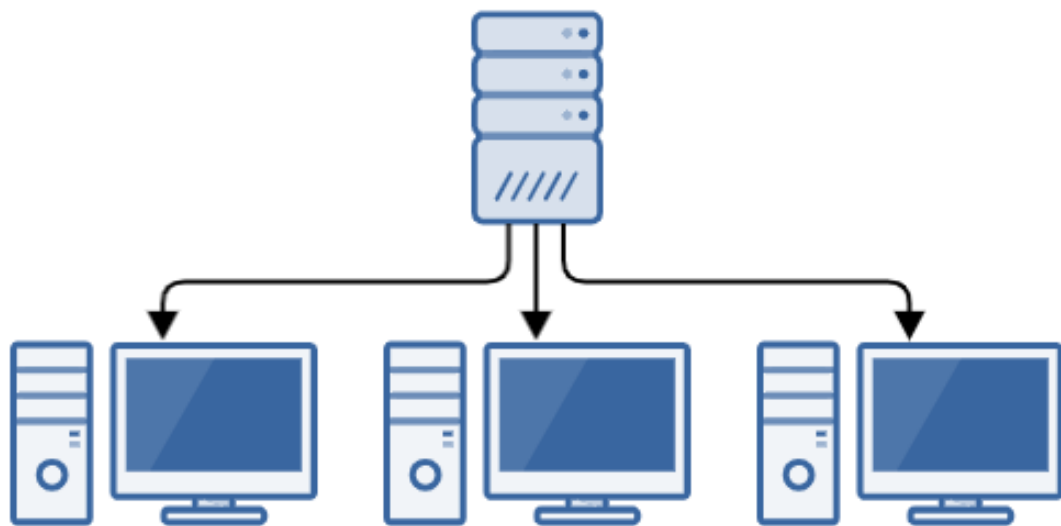
Problem: Events occur at wrong positions



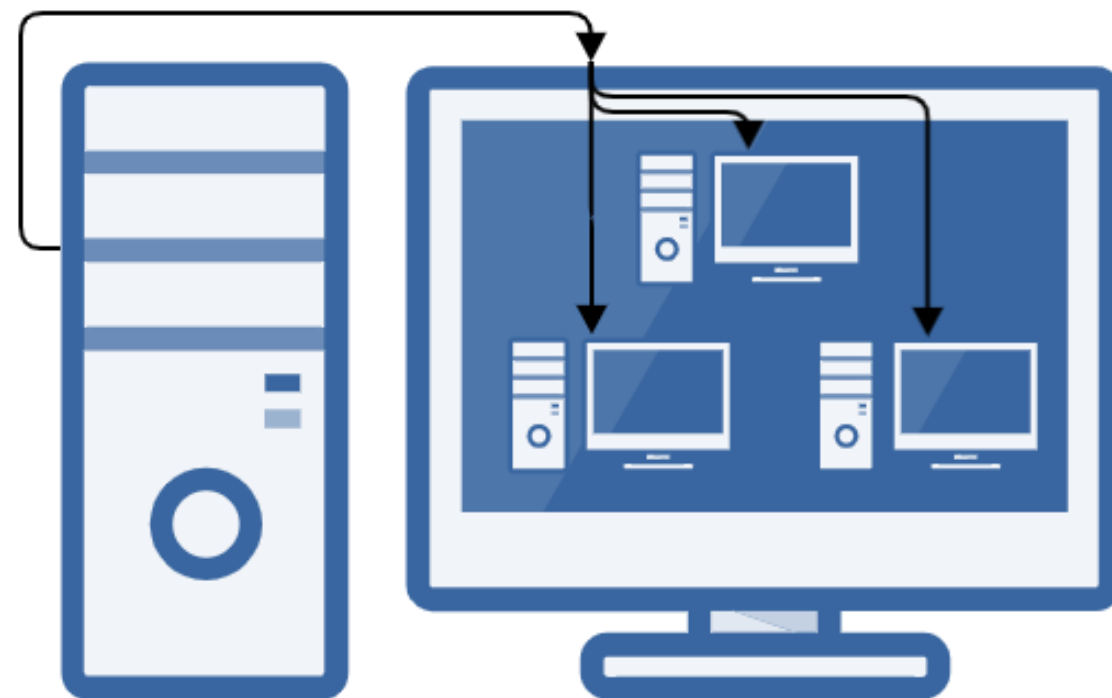
Solution: Store current position during jump



When Playing:
Each Client responsible for own Player



When Replaying:
Single Client controlling multiple Players



Recording

Create File

Add World Data

Add Fake Player

Add Entities

Playback

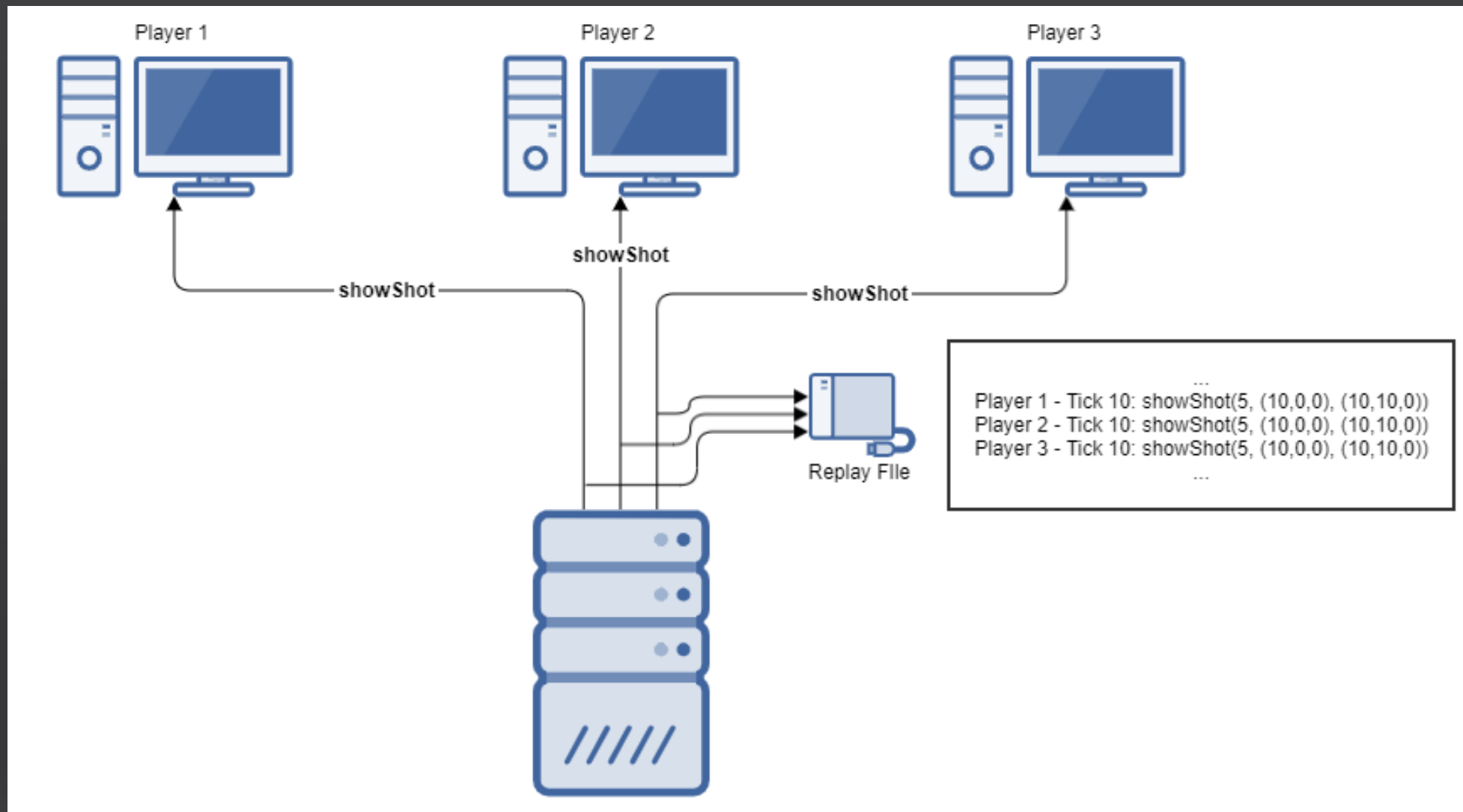
Open File

Load World

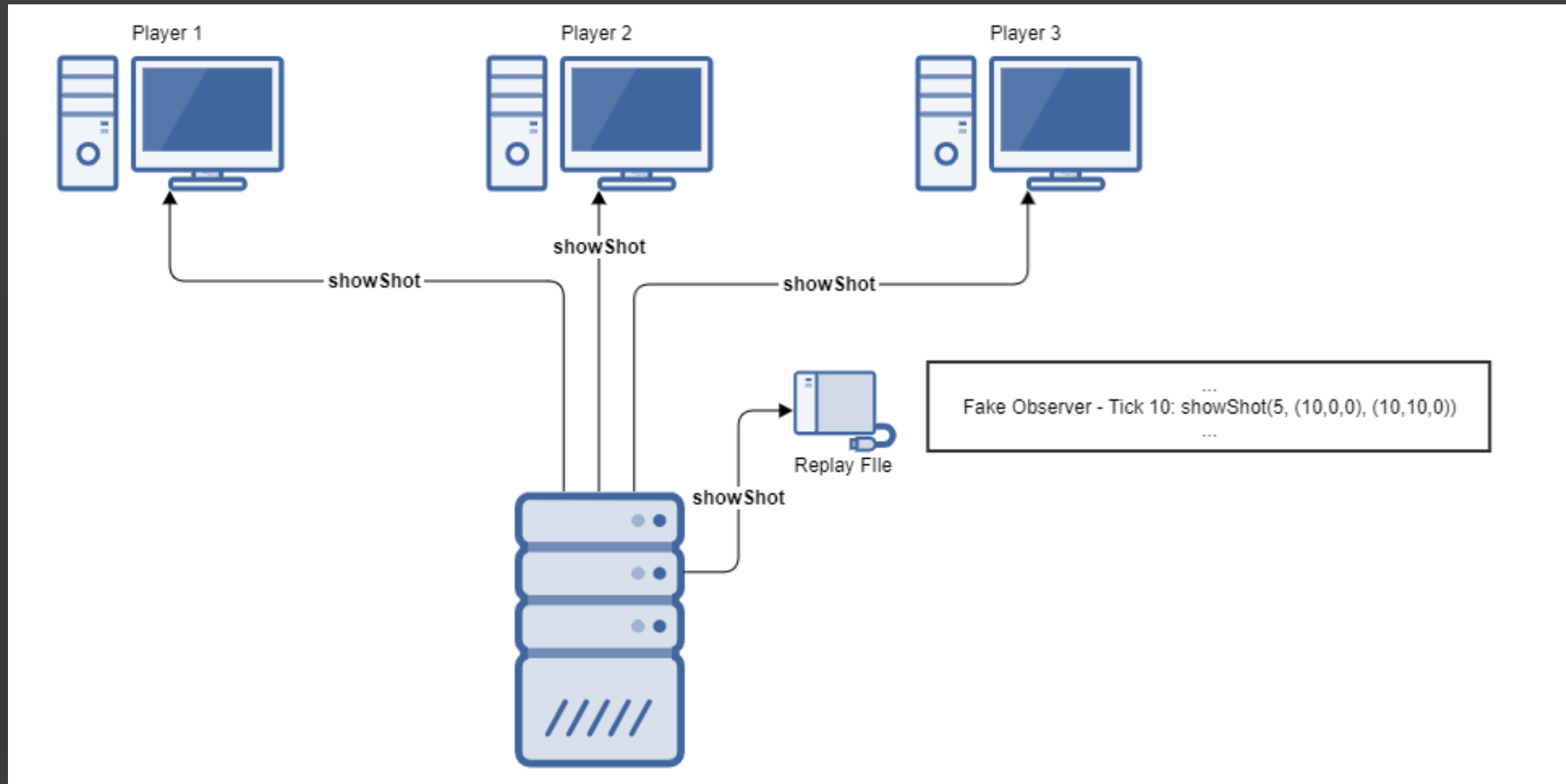
Create Player

Create Entities

Problem: Duplicate event playback

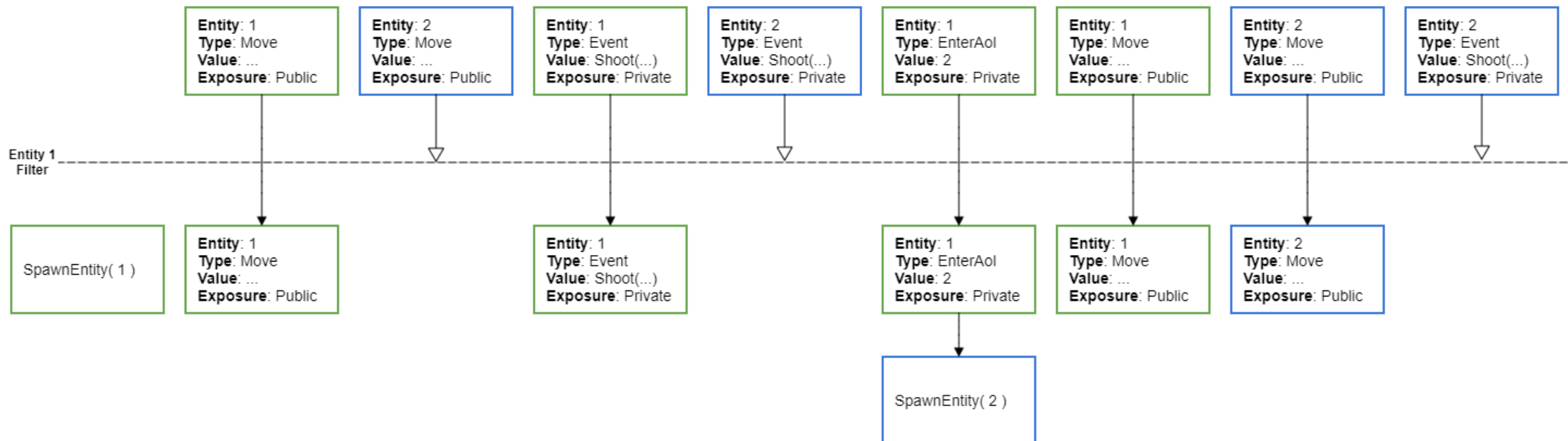


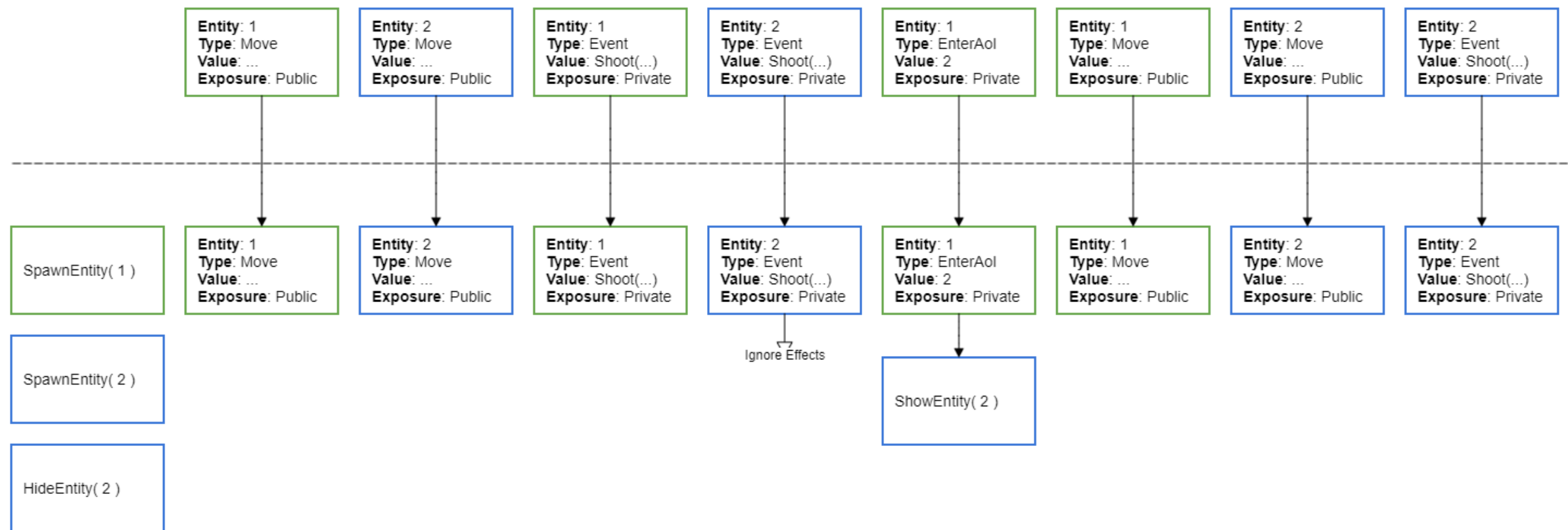
Solution: Special recording for multicast events



Problem: Can't see what the player saw







Problem: Watch an exact replica of the players experience



Solution: Please sir can I have some bandwidth



Solution: Fake it till you make it



Problem: Dynamic Metadata



Header

Initial Metadata

Tick Block 0

Tick Block 1

Metadata Block

Tick Block 3

...

Tick Block 12

Metadata Block

Tick Block 13



Memory

- No world instancing meant slower rewinds.
- Deactivated some resource caching.
- Put replay file on hard drive, which meant more certification requirements to adhere to.

Processing

- Free camera requires modifications to fog and loding.

Problem: Replays violate assumptions



- Low-LoD zones
- Inside objects
- Clip Planes
- Fog
- Number of objects in view
- World traversal speed
- Server-trip time delays
- Timers
- Server->Client messages
- Entity spawning
- Resource loading
- Caching
- Singletons
- Interpolation
- HUD elements
- Menu options
- Controls
- Persistent particle effects

Time flow

Network interpolation logic

World state representation

Player controller authority

How entities handle time

Assumptions

In Development





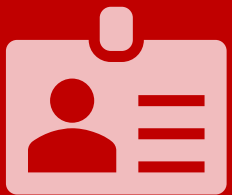
```
[11:00:00.000 (Tick: 0)] CREATE Vehicle[8332238]: {<Create Data>}
[11:00:00.000 (Tick: 0)] MOVE Vehicle[8332238] Teleport to (270.223, 20.8592, 154.908) and facing (1.72855e-05, -2.89399e-06, -1.48353)
[11:00:00.000 (Tick: 0)] CREATE AreaDestructibles[8012390]: {<Create Data>}
[11:00:00.000 (Tick: 0)] MOVE AreaDestructibles[8012390] Teleport to (51.3906, 0, -354.28) and facing (0, 0, 0)
[11:00:00.000 (Tick: 0)] CREATE AreaDestructibles[8012360]: {<Create Data>}
[11:00:00.000 (Tick: 0)] MOVE AreaDestructibles[8012360] Teleport to (251.391, 0, 254.28) and facing (0, 0, 0)
[11:00:00.000 (Tick: 0)] CREATE AreaDestructibles[8012349]: {<Create Data>}
[11:00:00.000 (Tick: 0)] MOVE AreaDestructibles[8012349] Teleport to (-345.72, 0, 348.609) and facing (0, 0, 0)
[11:00:00.000 (Tick: 0)] CREATE AreaDestructibles[8049626]: {<Create Data>}
[11:00:00.000 (Tick: 0)] MOVE AreaDestructibles[8049626] Teleport to (253.641, 0, 52.645) and facing (0, 0, 0)
[11:00:00.000 (Tick: 0)] CLIENT_CHANGED Avatar[8329357]
[11:00:00.000 (Tick: 0)] AOI_CHANGED witness: Avatar[8329357], entity: AreaDestructibles[8012344], isEnter: True
[11:00:00.000 (Tick: 0)] AOI_CHANGED witness: Avatar[8329357], entity: AreaDestructibles[8012345], isEnter: True
[11:00:00.000 (Tick: 0)] AOI_CHANGED witness: Avatar[8329357], entity: AreaDestructibles[8012347], isEnter: True
[11:00:00.000 (Tick: 0)] AOI_CHANGED witness: Avatar[8329357], entity: AreaDestructibles[8012348], isEnter: True
...
[11:00:03.100 (Tick: 31)] NEST_PROP Vehicle[8332237].['propertyName'] = [3, 9]
[11:00:03.100 (Tick: 31)] NEST_PROP Avatar[8329357].['otherProperty'] = 0.09633000195026398
[11:00:03.100 (Tick: 31)] NEST_PROP Avatar[8329357].['moreProperties'] = 26845157488478958
[11:00:03.100 (Tick: 31)] NEST_PROP Avatar[8329357].['yepStillMoreProps'] = 2
[11:00:03.100 (Tick: 31)] NEST_PROP Vehicle[8332237].['aNestedPropertyWithIndex', 0] = 35184
[11:00:03.100 (Tick: 31)] NEST_PROP Vehicle[8332237].['anArrayBasedProperty'] = "[35184]"
[11:00:03.100 (Tick: 31)] METHOD Avatar[8329357].updateSomethingOrOther( {"Arg0": 8332237, "Arg1": 0.0} )
[11:00:03.100 (Tick: 31)] MOVE Vehicle[8332237] Interpolate to (-67.0487, 16.3816, -380.984) and facing (0.184388, -0.00321671, 0.533024)
[11:00:03.200 (Tick: 32)] NEST_PROP Avatar[8329357].['moreProperties'] = 27411406111001338
[11:00:03.200 (Tick: 32)] METHOD Avatar[8329357].updateSomethingOrOther( {"Arg0": 8332237, "Arg1": 0.0} )
[11:00:03.200 (Tick: 32)] MOVE Vehicle[8332237] Interpolate to (-66.9459, 16.3796, -380.812) and facing (0.18526, -0.00884486, 0.534389)
[11:00:03.300 (Tick: 33)] NEST_PROP Vehicle[8332237].['engineMode'] = [3, 1]
[11:00:03.300 (Tick: 33)] NEST_PROP Avatar[8329357].['moreProperties'] = 1757357712889006838
[11:00:03.300 (Tick: 33)] METHOD Avatar[8329357].updateSomethingOrOther( {"Arg0": 8332237, "Arg1": 0.0} )
[11:00:03.300 (Tick: 33)] MOVE Vehicle[8332237] Interpolate to (-66.806, 16.3759, -380.578) and facing (0.187529, -0.0092779, 0.533327)
[11:00:03.400 (Tick: 34)] NEST_PROP Avatar[8329357].['moreProperties'] = 2838786772837288694
[11:00:03.400 (Tick: 34)] METHOD Avatar[8329357].updateSomethingOrOther( {"Arg0": 8332237, "Arg1": 0.0} )
[11:00:03.400 (Tick: 34)] MOVE Vehicle[8332237] Interpolate to (-66.6518, 16.3702, -380.32) and facing (0.190906, -0.00263722, 0.534449)
```



Reliable reproductions



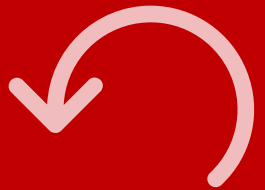
Share and access from anywhere



Metadata



Performance Testing



Regression Testing



Smoke Test



Reproducing error reports



Identifying cheats



Addressing player complaints



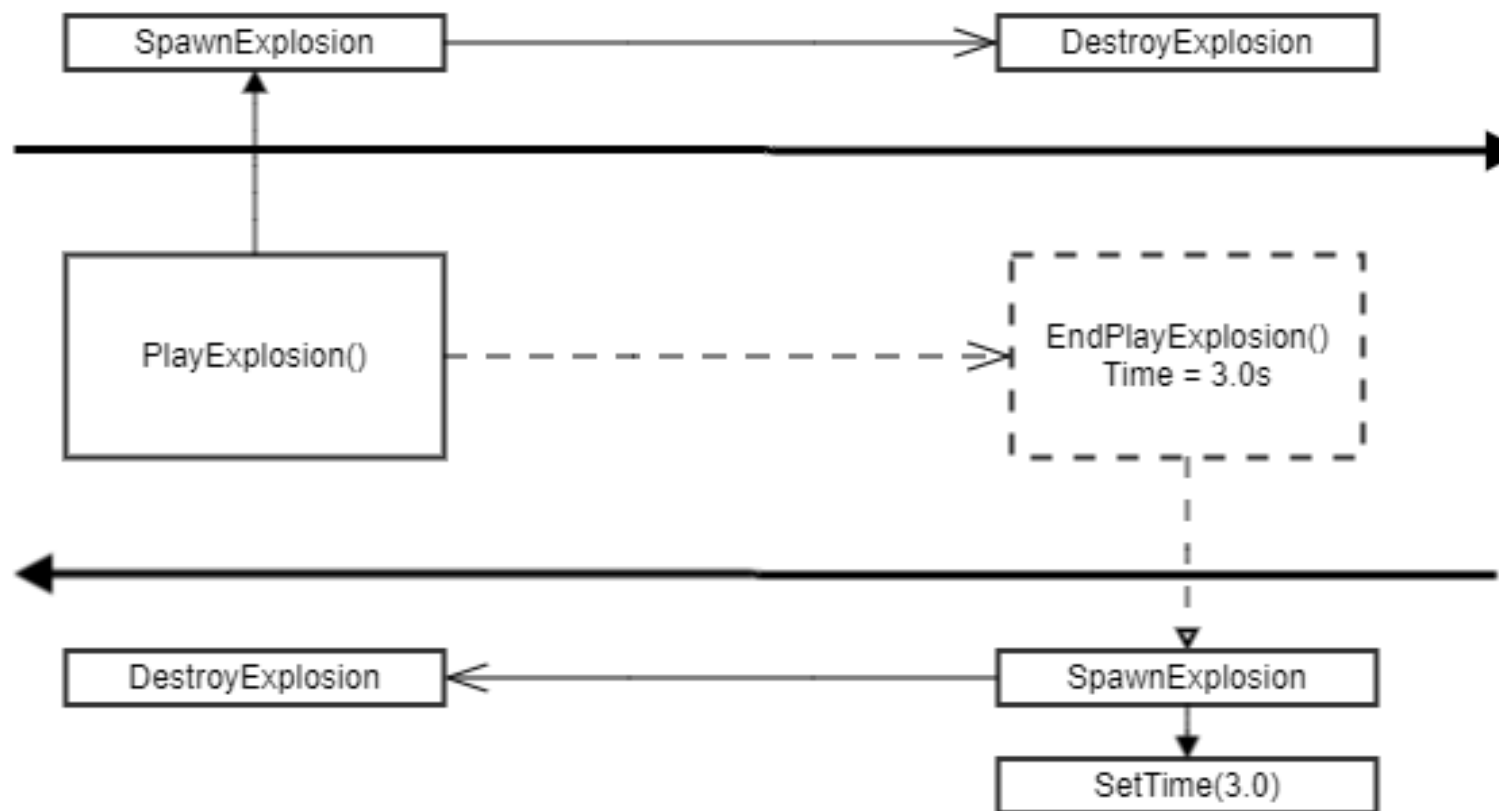
Post-Processed Analytics

Future Challenges

Problem: Playback in reverse



Potential Solution: Playback in reverse





Map Changes



Asset Changes



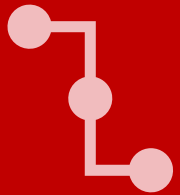
Entity Definition Changes



Code Changes



On-Demand Asset Download



Property Update Paths



Release Older Clients

- When designing/updating a system, consider:
 - Keep replays and live game code paths the same as much as possible
 - The flow of time and how well it handles abnormal rates
 - How time jumping around affects your output
 - How data is stored and updated
 - The importance of data members and events
- Why do I want to do all this extra thinking:
 - Valuable, engaging player feature
 - Powerful testing and debugging tool
 - More bulletproof systems design

THANK YOU! QUESTIONS?

Andrew Glover
Wargaming Sydney
a_glover@wargaming.net