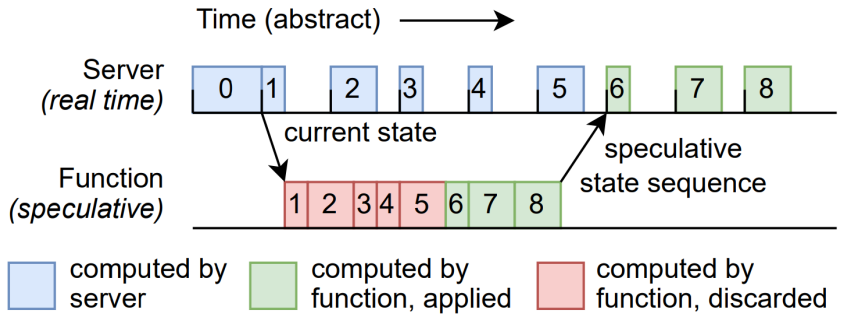
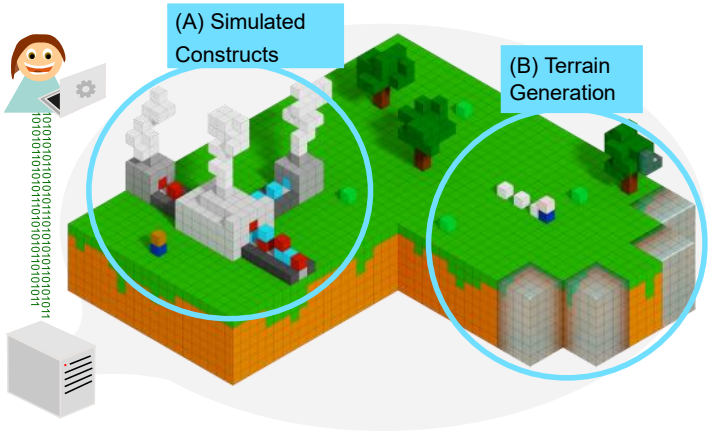
Playing together with 4 Billion People: Creating Large-Scale Modifiable Virtual Worlds for Digital Societies



MINECRAFT



My publications

Jesse Donkervliet

j.j.r.donkervliet@vu.nl

https://www.jdonkervliet.com

Our Society Benefits from Games

Take Minecraft

- Over **125 million people** play Minecraft every month
- **40,000+ mods**
- **100+ games "like M'craft."**

Generally Beneficial Features

- **Entertainment**
- Education
- Activism
- Social Interaction

YouTube

Twitch

Minecraft: Connecting More Players Than Ever Before

by Helen Chiang, Studio Head, Mojang Studios • May 18, 2020 @ 6:00am

MINECRAFT

HISTORY BLOCKS

8-10 yrs old | 11-13 yrs old | 14-18 yrs old | 18+ yrs old | Geography | Reading and Writing | Service Learning & Social Good

The purpose of this activity is to guide teachers through an activity in which students reconstruct Unesco world heritage sites Minecraft.

How Greenpeace Used Minecraft to Stop Illegal Logging in Europe's Last Lowland Primeval Forest

Good game, everybody

By Angela Natividad | January 22, 2018

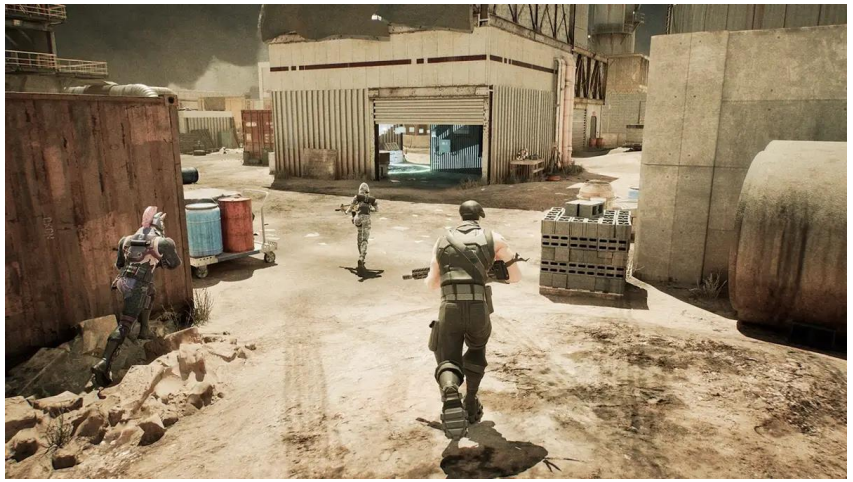
Pussy Riot, Idles to play Minecraft virtual festival 'Block by Blockwest'

It starts this Saturday

By Makena Kelly | @kellymakena | Apr 22, 2020, 1:33pm EDT

Games are Massively Popular; MVEs are the Most Popular Games

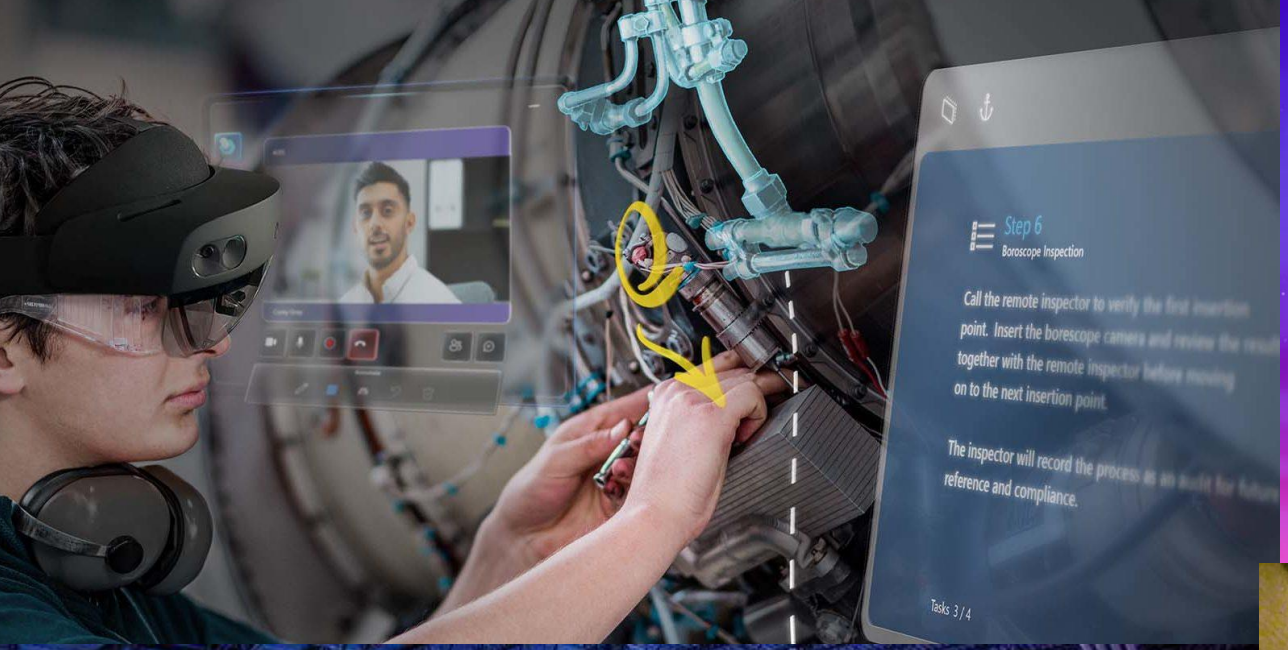
- 3.3B players in 2023, 3.8B forecasted for 2027
- $188B annual gaming market revenue



Top Monthly Active Users (MAU)

1.		Minecraft
2.		ROBLOX
3.		Fortnite

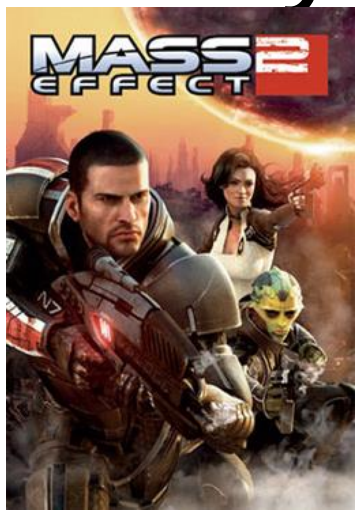
Source: Newzoo https://newzoo.com/resources/rankings/top-20-pc-games

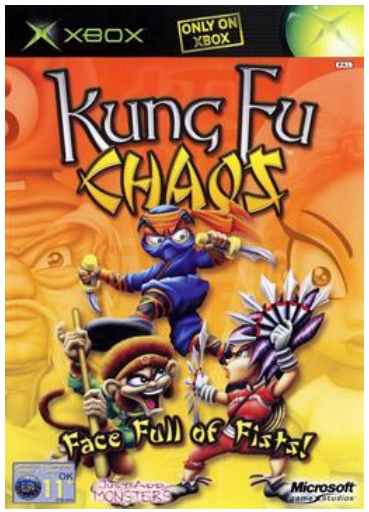


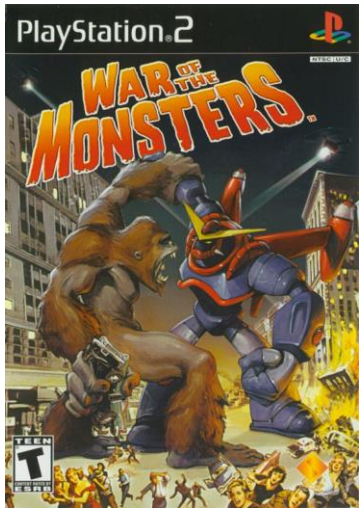
@Large Research
Massivizing Computer Systems

VU VRIJE UNIVERSITEIT AMSTERDAM

Me as a Gamer:
Exploring Worlds and Playing Together







PHENOMENON: PERFORMANCE DROPS IN VIRTUAL WORLDS

SECOND LIFE

ROBLOX

MINECRAFT

Polygon

Source: http://bit.ly/EveOnline21Crash

NEWS

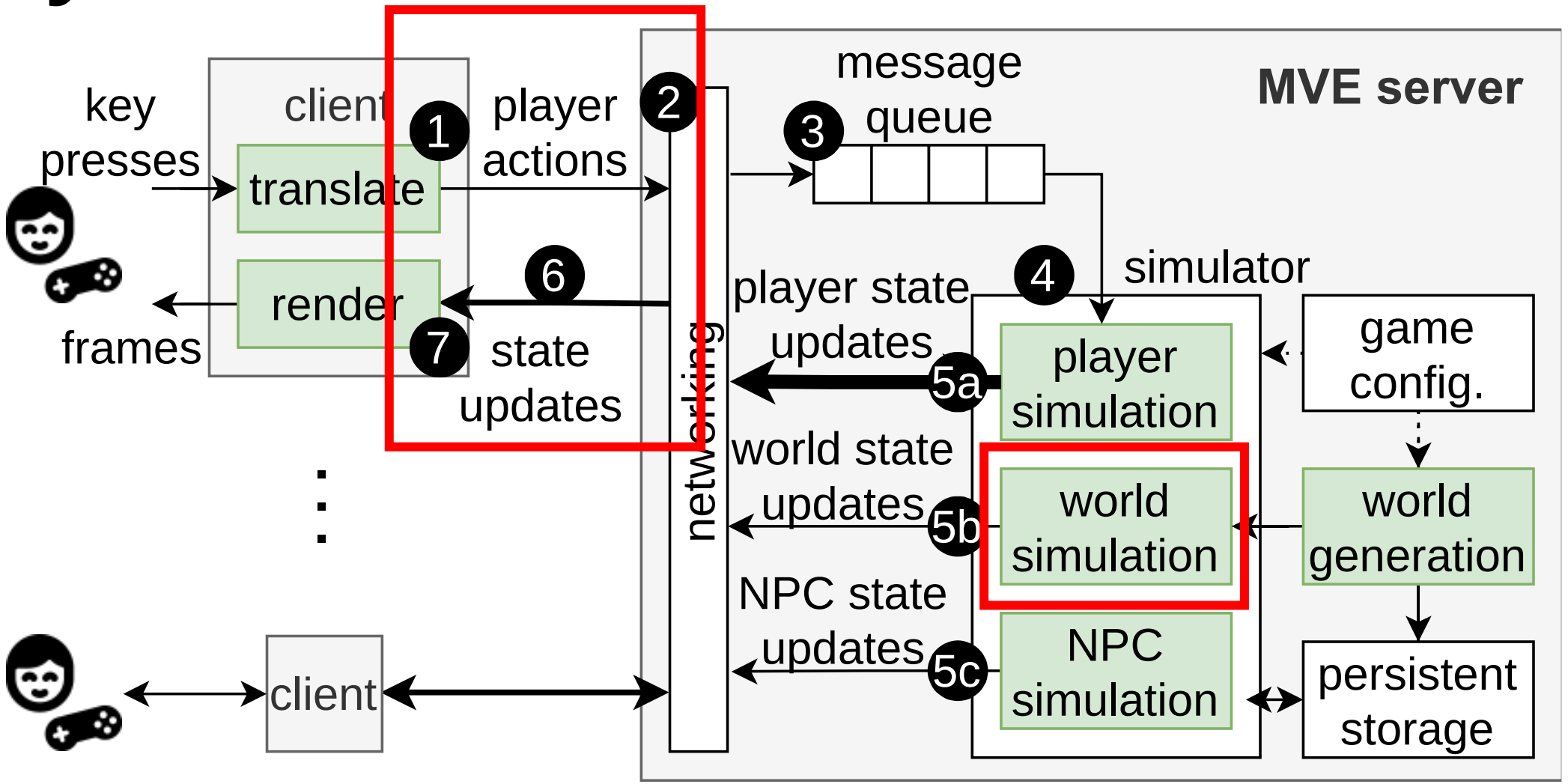
Players in Eve Online broke a world record — and then the game itself

Developers said they're not 'able to predict the server performance in these kinds of situations'

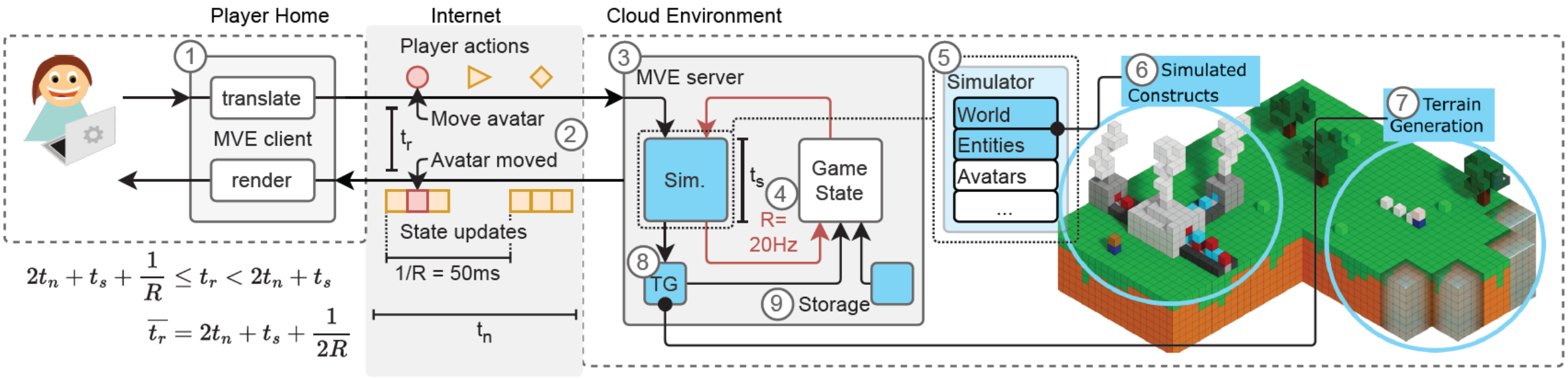
By Charlie Hall | @Charlie_L_Hall | Jan 5, 2021, 2:54pm EST

Source: Razorien/CCP Games

Why Are MVEs Difficult to Scale?



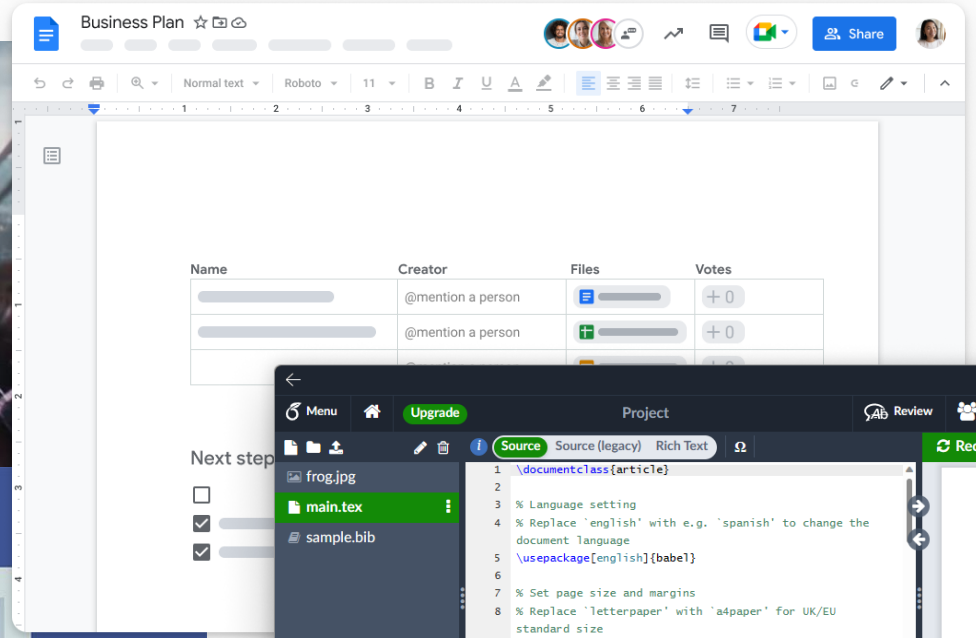
Why Are MVEs Difficult to Scale?

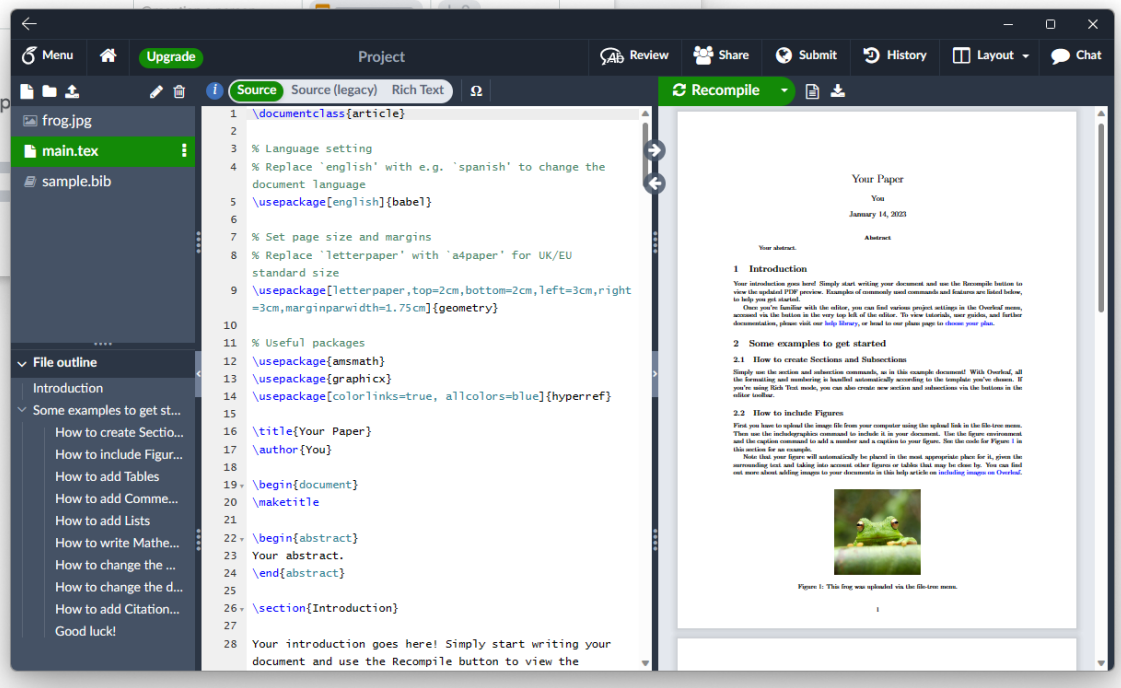


$$2t_n + t_s + \frac{1}{R} \leq t_r < 2t_n + t_s$$

$$\overline{t_r} = 2t_n + t_s + \frac{1}{2R}$$

Why Are MVEs Difficult to Scale?





In this talk

1. Benchmarking Modifiable Virtual Worlds (Yardstick)
2. Dynamic Consistency Management (Dyconits)
3. Serverless offloading for MVEs (Servo)
4. Performance Analysis of Virtual Reality Hardware (Dizi)

Meterstick: Benchmarking Performance Variability in Cloud and Self-hosted Minecraft-like Games

Jerrit Eickhoff

M.Sc. @ TU Delft, AtLarge Research

jerrit.eickhoff@gmail.com

https://atlarge-research.com/opencraft/

Ir. Jesse Donkervliet

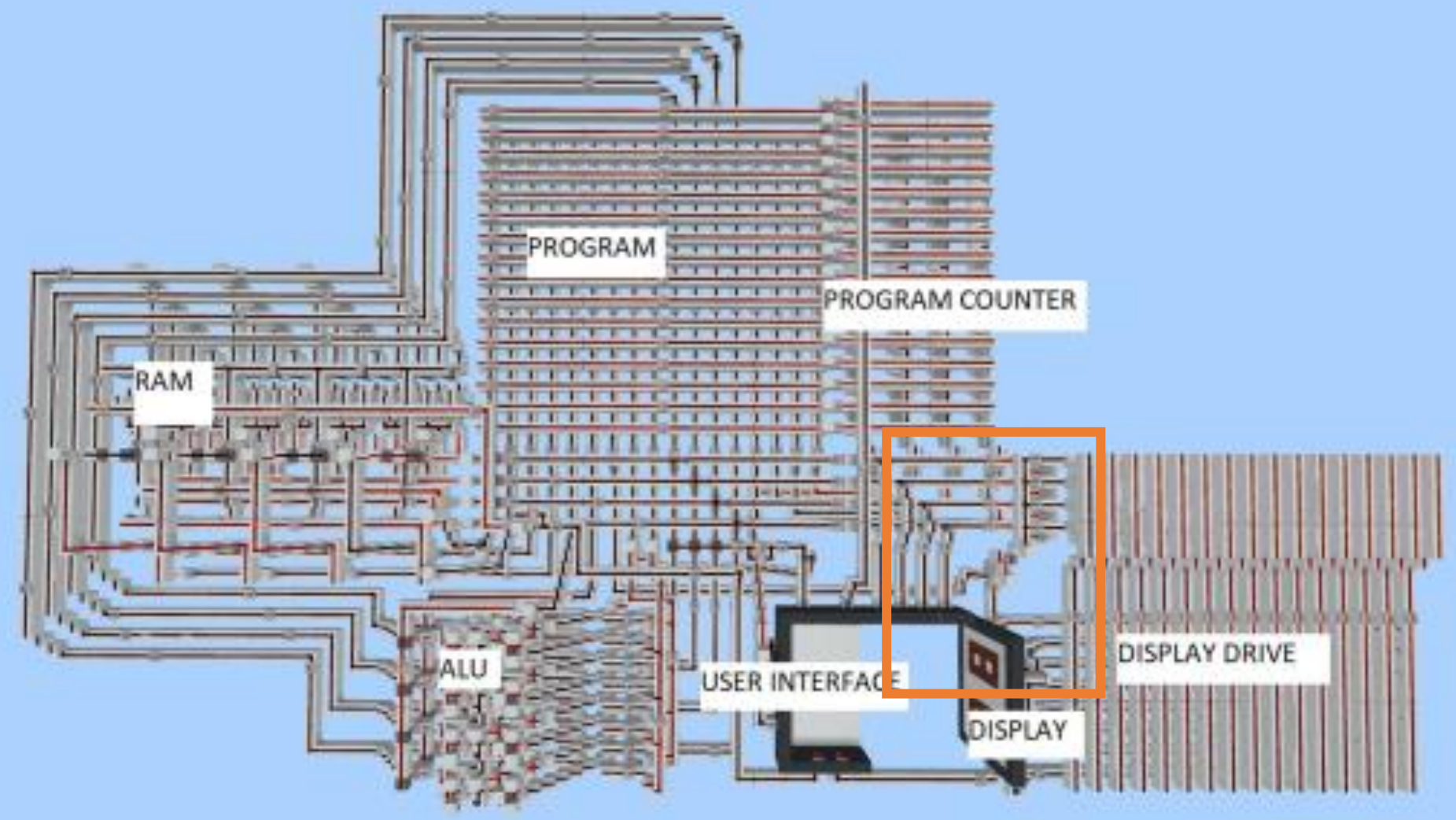
Prof. dr. ir. Alexandru Iosup

@Large Research
Massivizing Computer Systems

Source and data available!
Meterstick: https://github.com/atlarge-research/Meterstick
Data: https://zenodo.org/record/7657838

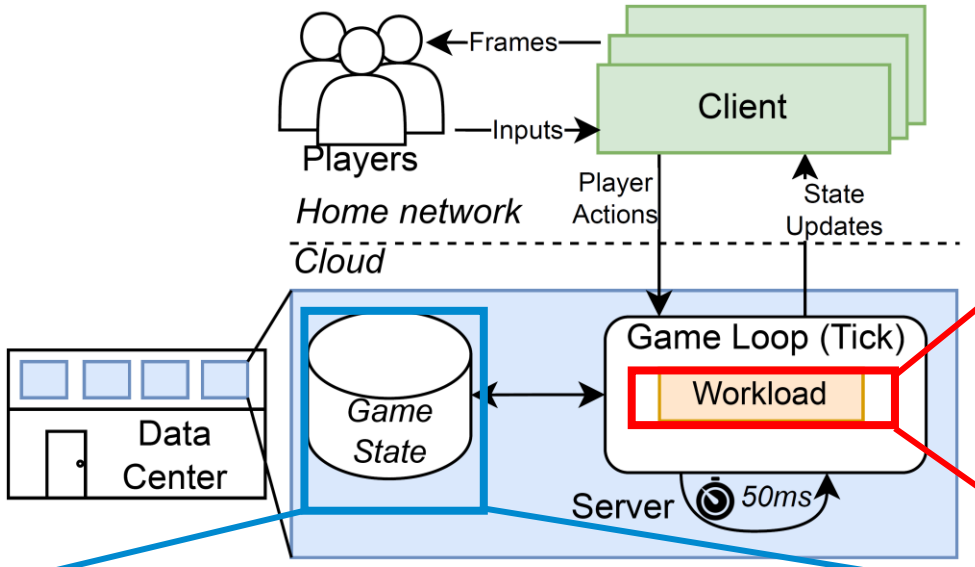
ICPE 2023
Coimbra, Portugal

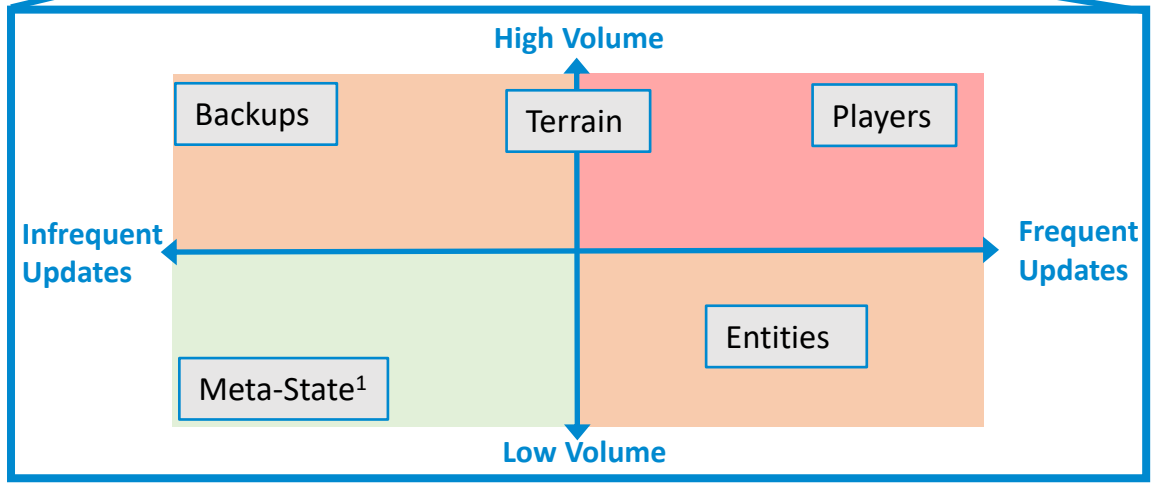
TU Delft VU Vrije Universiteit Amsterdam


PROGRAM
PROGRAM COUNTER
RAM
ALU
USER INTERFACE
DISPLAY DRIVE
DISPLAY

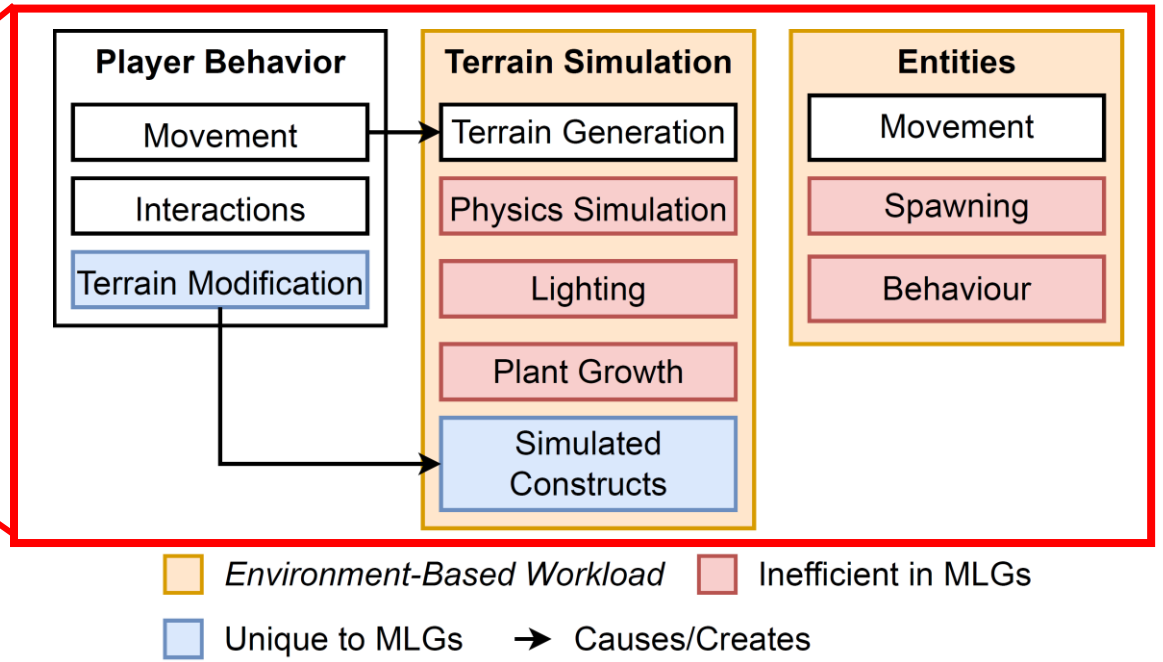
A single player crashing the game!? How can this be?

Server-Client Architecture





Minecraft-like Game Workload Model

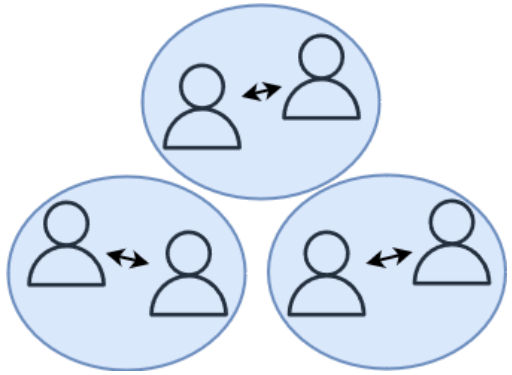


Assumed Game State Volume and Update Frequency

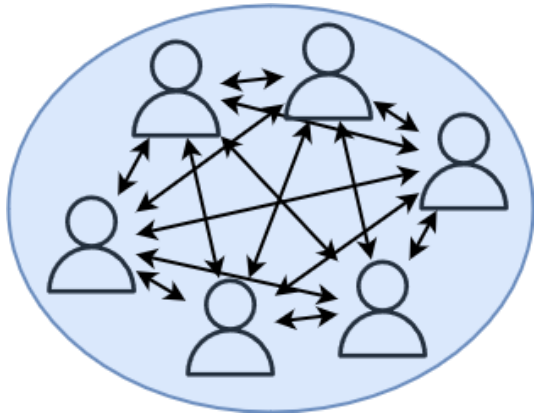
[1]: State concerning functional operation of the game server rather than game features, such as administrative logs or user authentication tokens.

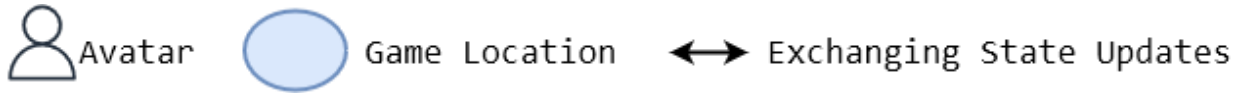
Player Workload

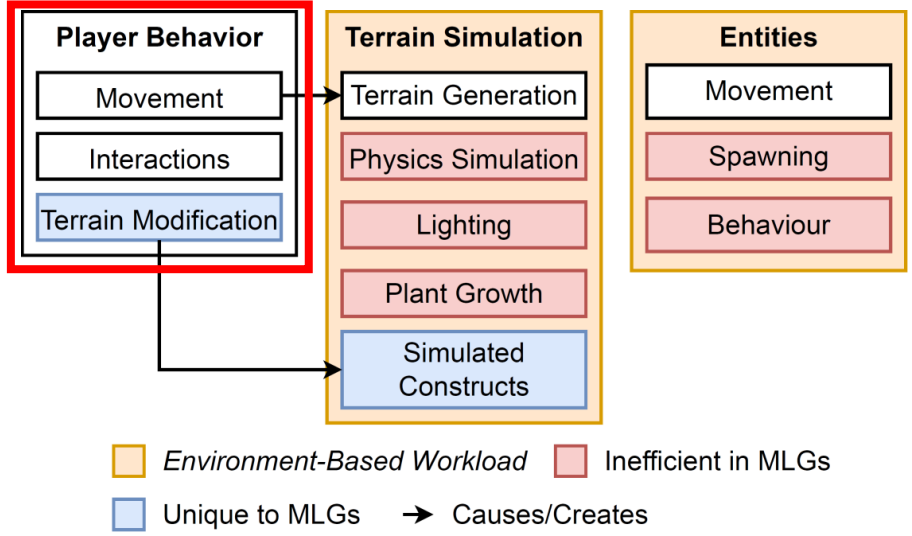
Player Avatars Sparse



Players Avatars Dense







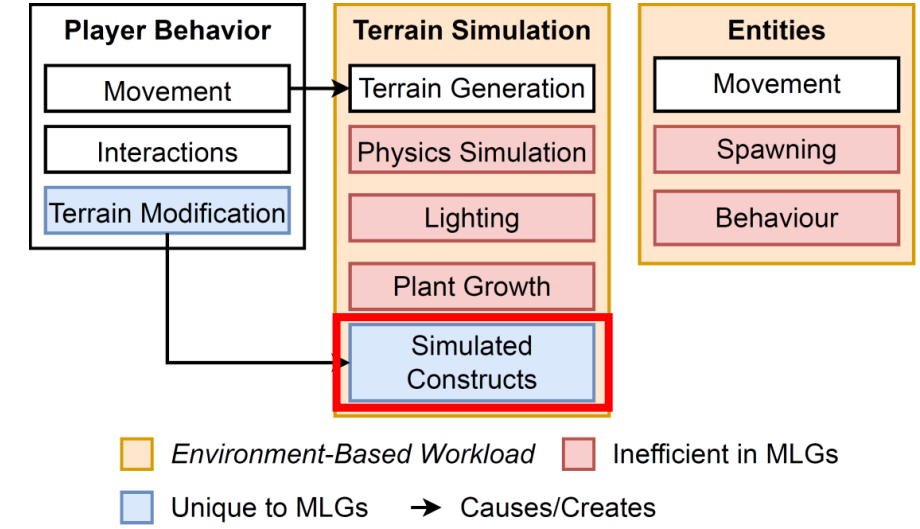
Eve Online 13,700 player battle causes performance disruptions

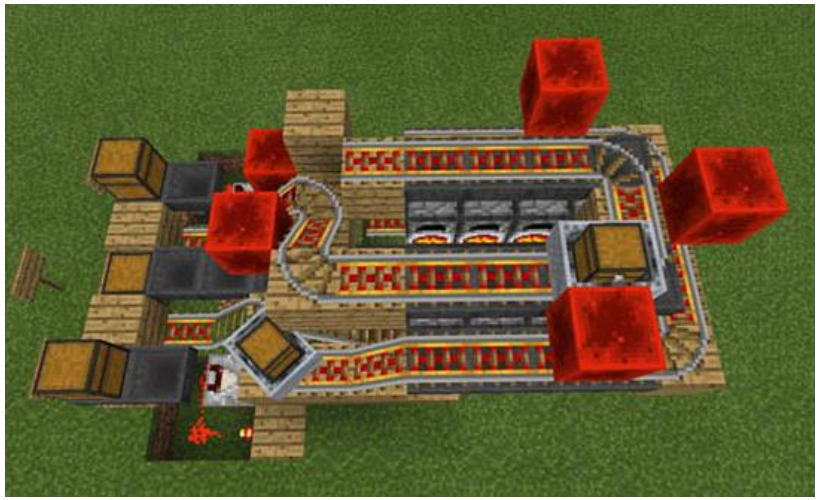
Sources: Polygon

Environment-Based Workloads

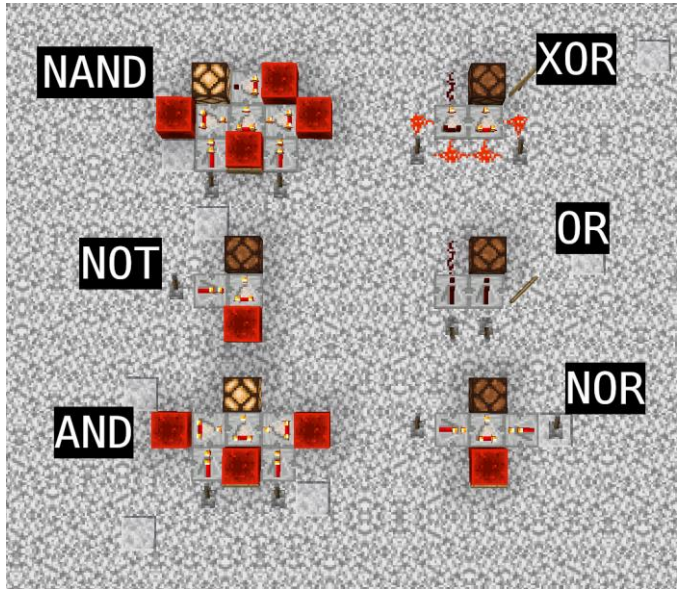
Simulated Constructs

- **Player-constructed** structures consisting of dynamic elements
- "Programmed" to **automatically perform** some in-game task

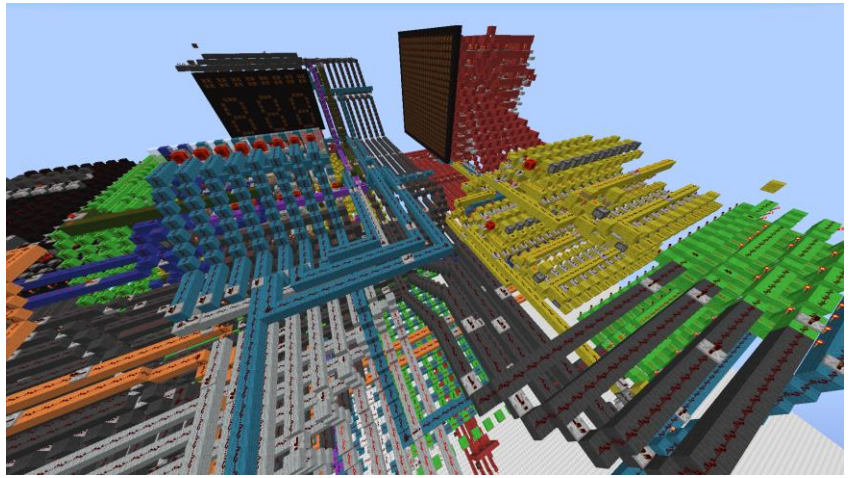




Automatic resource processing



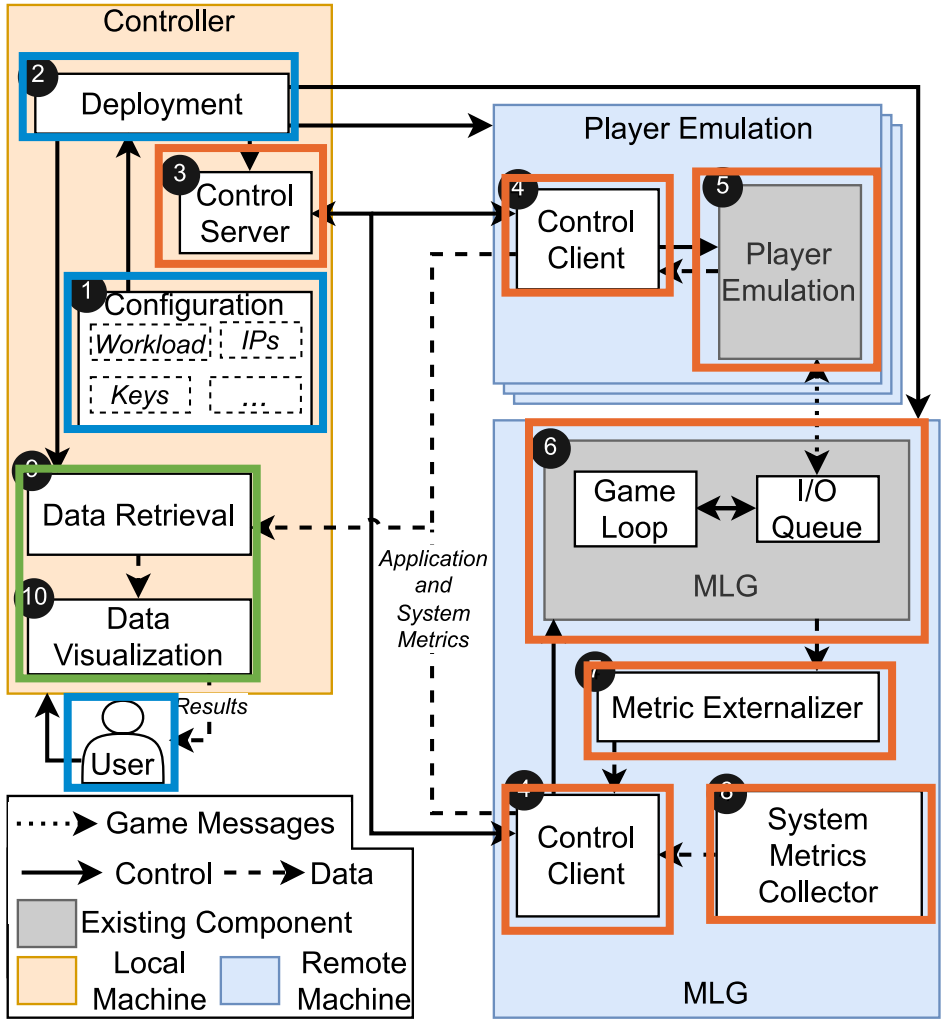
Logic Gates



Operational 16-bit, 1Hz computer

Sources: Reddit, Reddit

Meterstick Benchmark: Design

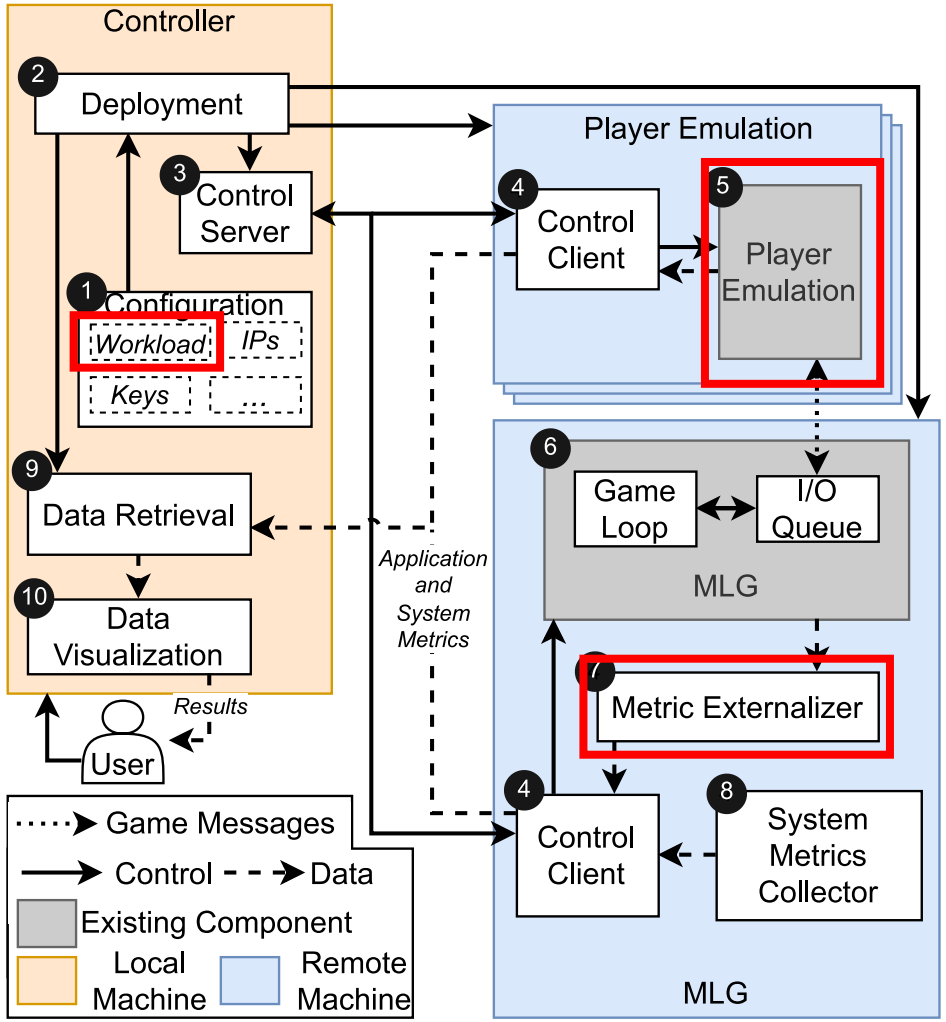


- Supports **environment-based workloads**
- Uses **player-emulation** for player contribution to workload
- Deploys Minecraft-like Games experiments on **commercial clouds**
- Collects relevant **application and system metrics**

Steps:

1. Deployment
2. Experiments
3. Data retrieval

Meterstick Benchmark: Design



- Workloads, Player Emulation, and Metric Externalization tied, directly or indirectly, to application protocol
- Currently supports Minecraft-like games utilizing the **Minecraft protocol**

Minecraft server implementations

Popular[1] mod packs

Same server technology for different games!

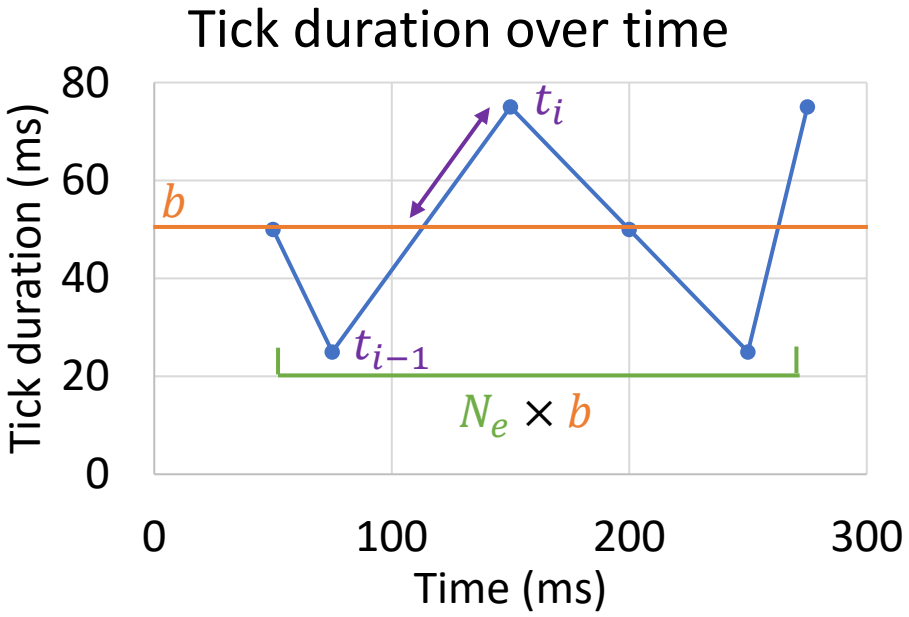
[1]: Ranges from 150 thousand to 2 million downloads, with some individual mods reaching 223 million downloads. See TechicPack and CurseForge

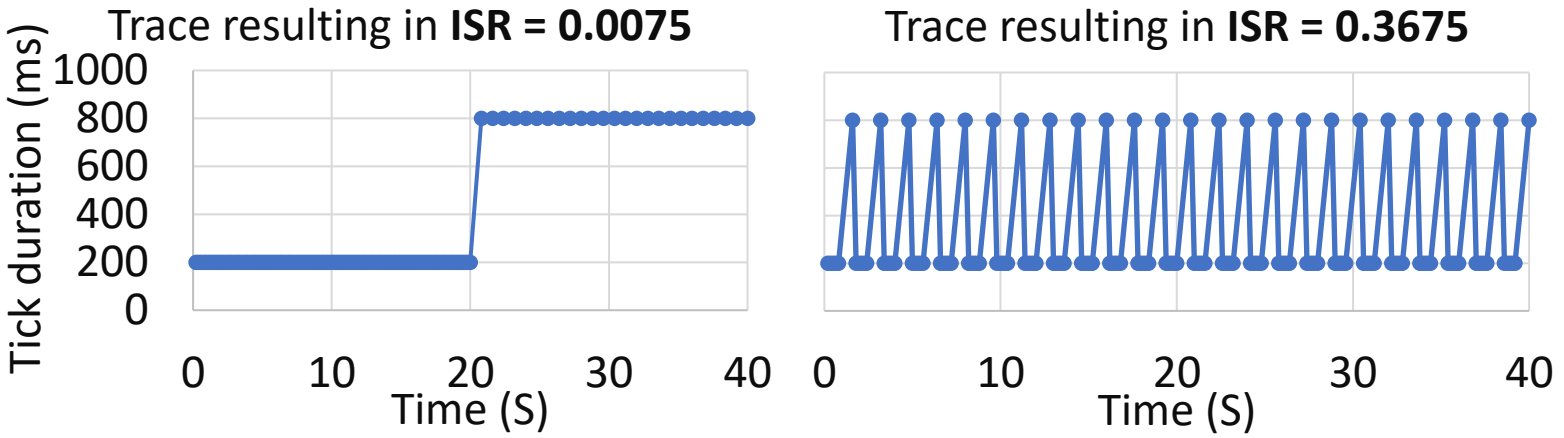
Instability Ratio (ISR)

$$ISR = \frac{\sum_{i=1}^{N_a} |\max(b, t_i) - \max(b, t_{i-1})|}{N_e \times 2b}$$

b = minimum delay between ticks
t_i = duration of i^{th} tick
N_a = actual number of ticks
N_e = expected number of ticks

- **Stability > lowest latency** for online gaming [1-3]
- **Normalized** measure of instability given a trace of tick durations, based on cycle-to-cycle jitter.
- **Order dependent**





Same tick durations, different order

Sources:
1: How sensitive are online gamers to network quality? Chen et al. Commun. ACM 49, 11 (2006)
2: Player Perception of Delays and Jitter in Character Responsiveness, Normoyle et al. SAP2014
3: Empirical study of subjective quality for Massive Multiplayer Games, Ries & Rupp, IEEE (2008)

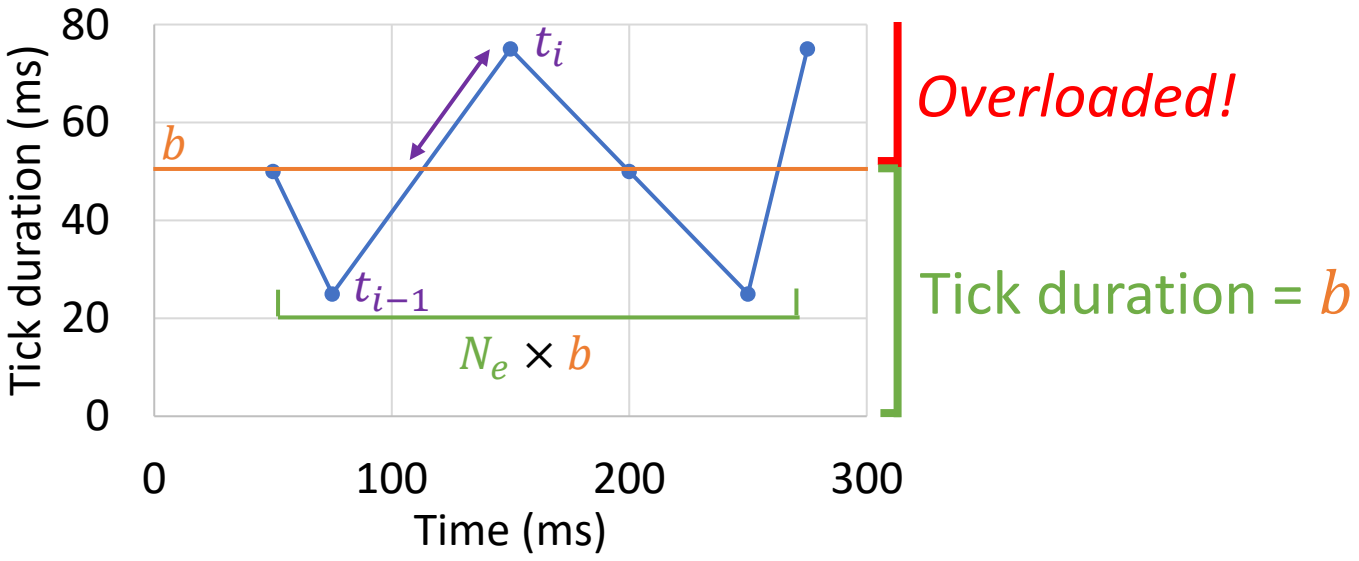
Instability Ratio (ISR)

$$\boldsymbol{ISR} = \frac{\sum_{i=1}^{N_a} |\max(b, t_i) - \max(b, t_{i-1})|}{N_e \times 2b}$$

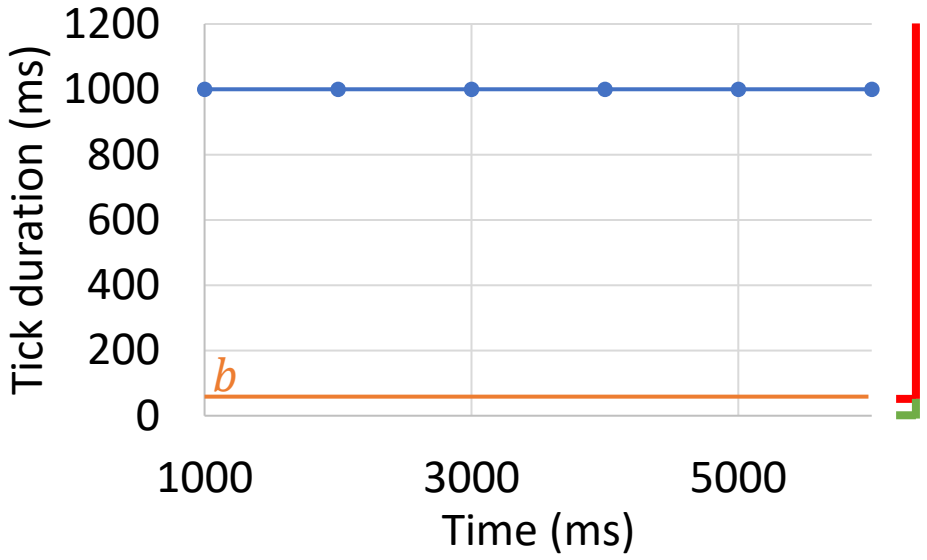
- b = minimum delay between ticks
- t_i = duration of i^{th} tick
- N_a = actual number of ticks
- N_e = expected number of ticks

- $\boldsymbol{ISR}$ = 0 if all ticks below b!
- $\boldsymbol{ISR}$ = 0 if all ticks are **the same!**
- **Not meant to be used as standalone performance metric!**

Tick duration over time



Trace resulting in **ISR = 0**



Experiment - Setup

Minecraft-like Games

Minecraft
Default

Forge
Mods

PaperMC
Performance

Environments

Amazon Web Services

Azure

DAS5 Cluster

2vCPU: AWS: *T3.Large,* Azure*: Standard_D2_v3*

Workloads:

Workload Name	Description
Control*	Freshly generated world
TNT*	Fast entity actions, terrain updates
Farm*	Many simulated constructs
Lag*	Simulated construct stress test
Players	25 moving players in small area

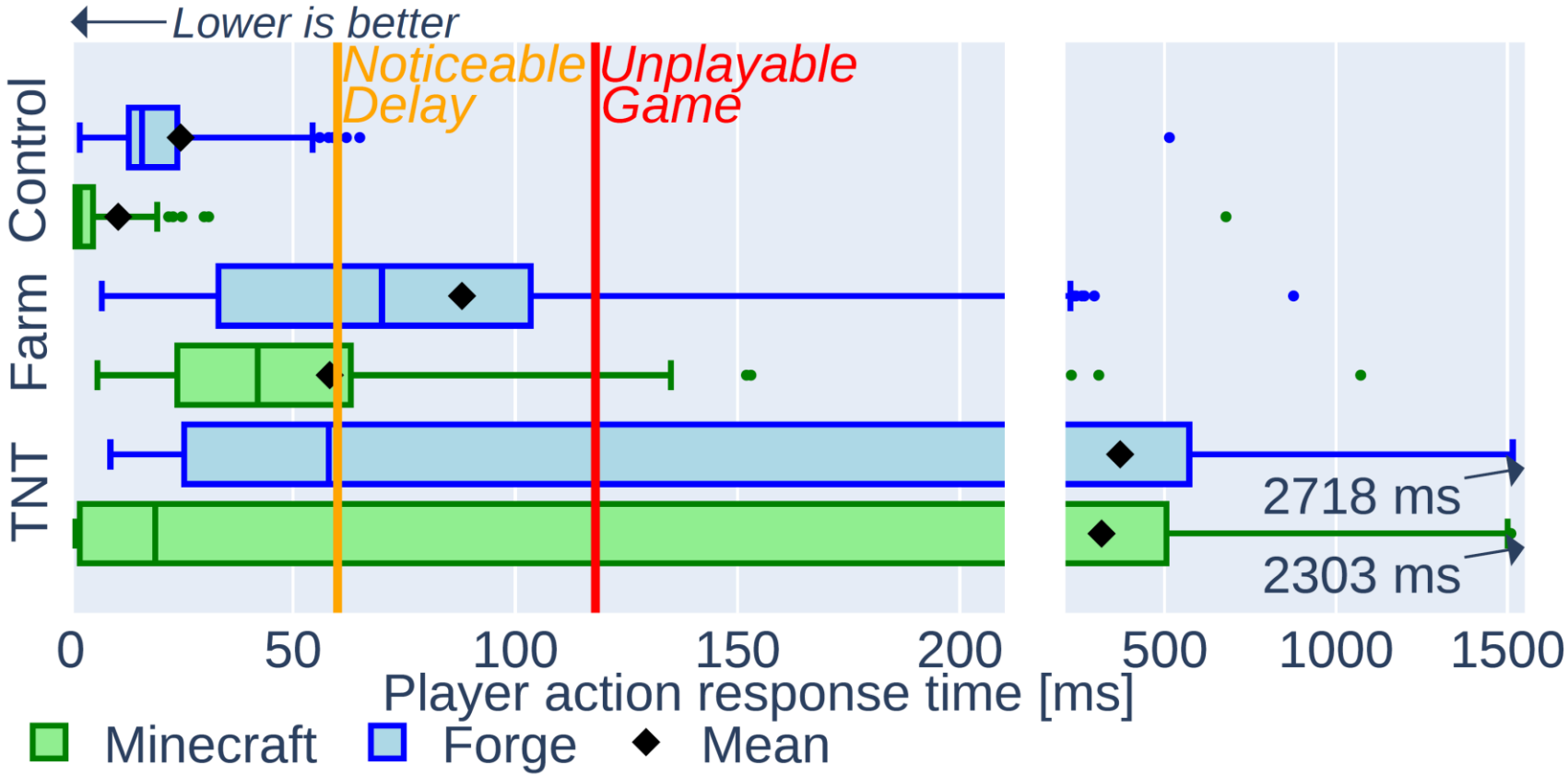
**Only one player, stationary*

Hardware Guidelines:

Service	vCPU[#]	CPU Speed [GHz]
Server.pro	2	2.4
Skynode	2	3.6
Hostinger	3	NP
Ferox Hosting	Not reported	Not reported
MelonCube	Not reported	3.4
Azure	2	Variable
AWS	1	Variable

Full list of cloud service Minecraft-like game hosting recommendations, and community simulated constructs, available in technical report: https://arxiv.org/abs/2112.06963

Environment-based workloads cause significant performance instability



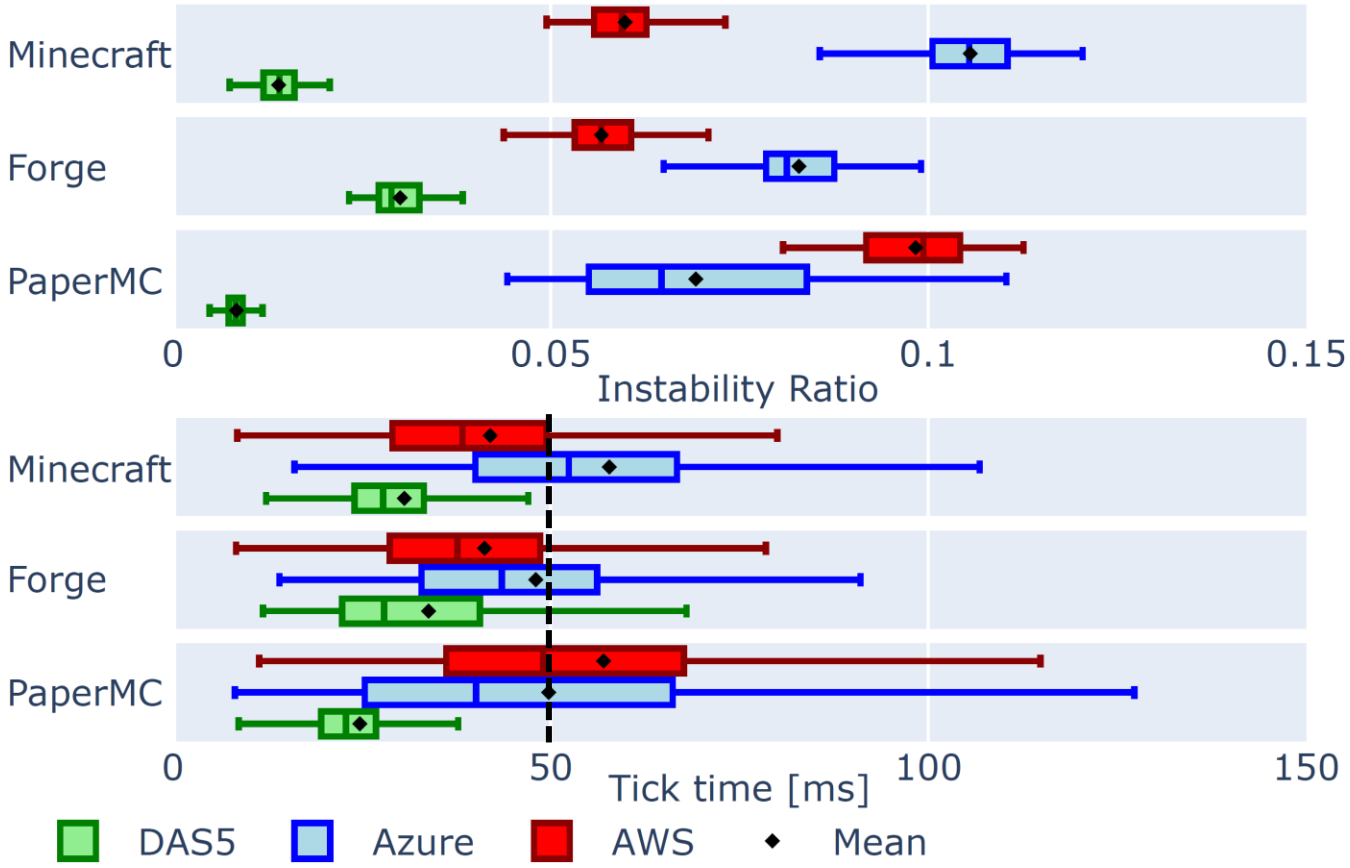
Lag workload missing, crashed!

Player action response time on AWS

Whiskers to 5th, 95th percentiles

Sources for Noticeable, Unplayable thresholds:
1: Analysis of factors affecting players' performance and perception in multiplayer games, Dick et al. Netgames 2005
2: Are 100 ms Fast Enough? Characterizing Latency Perception Thresholds in Mouse-Based Interaction, Forch et al. EPCE 2017

Cloud environments cause significant performance variability



Variation of **Instability Ratio** and **Tick time** over 50 iterations of Players workload

Whiskers to 1.5 x IQR

Dyconits: Scaling Minecraft-like Services through Dynamically Managed Inconsistency

Jesse Donkervliet, Jim Cuijpers, Alexandru Iosup

jesse.donkervliet@vu.nl
@jdonkervliet
https://atlarge-research.com/opencraft/

Scalability Challenge

Minecraft supports 126 million active monthly players, but only by using **isolated instances** that do not scale beyond **a few hundred players**.[1]

Minecraft music festival Block By Blockwest postponed after servers crash

Over 100,000 people logged on to catch virtual performances by Massive Attack and more

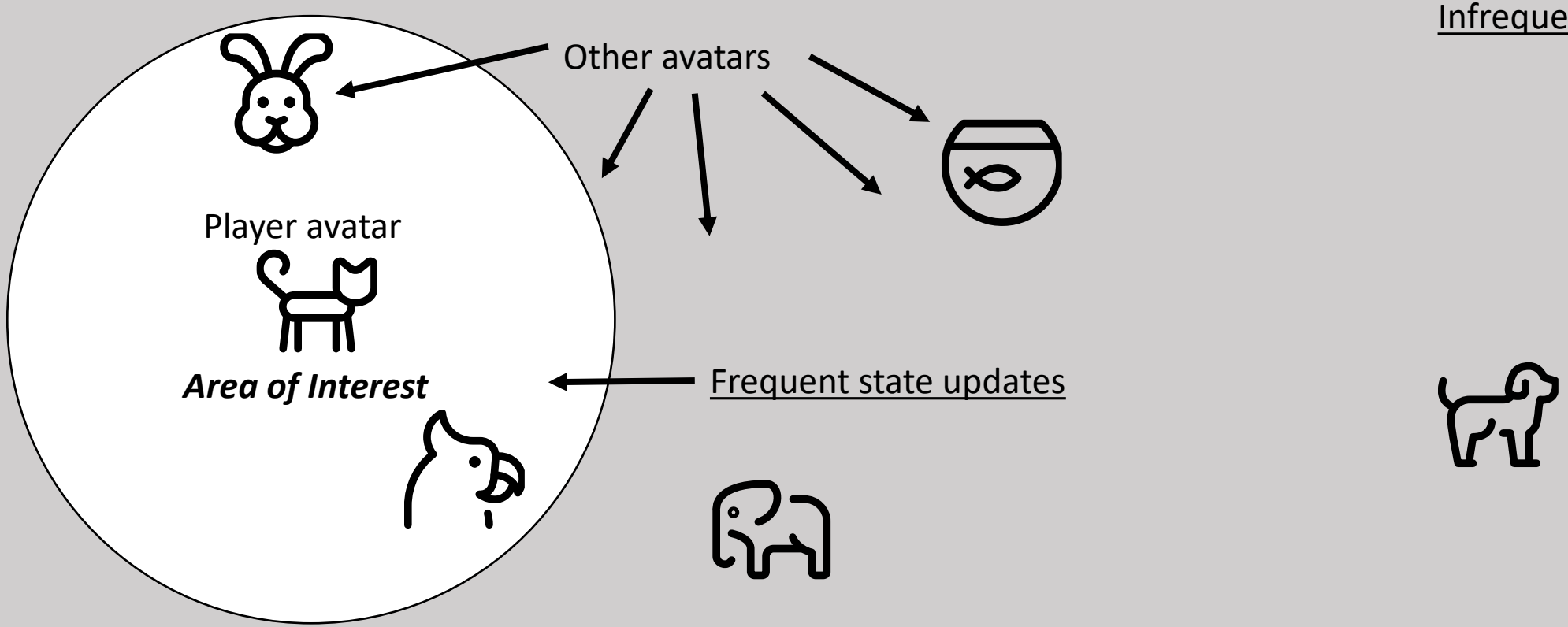
By **Patrick Clarke** | 26th April 2020

Sources:
1: Yardstick: A benchmark for minecraft-like services, Jerom van der Sar et al. ICPE2019.

https://www.nme.com/news/music/minecraft-music-festival-block-by-blockwest-postponed-after-servers-crash-2653948

Interest Management

Intuition: only update state players are interested in



Limitations of Interest Management

Existing approaches consider only *staleness*, or are not dynamic

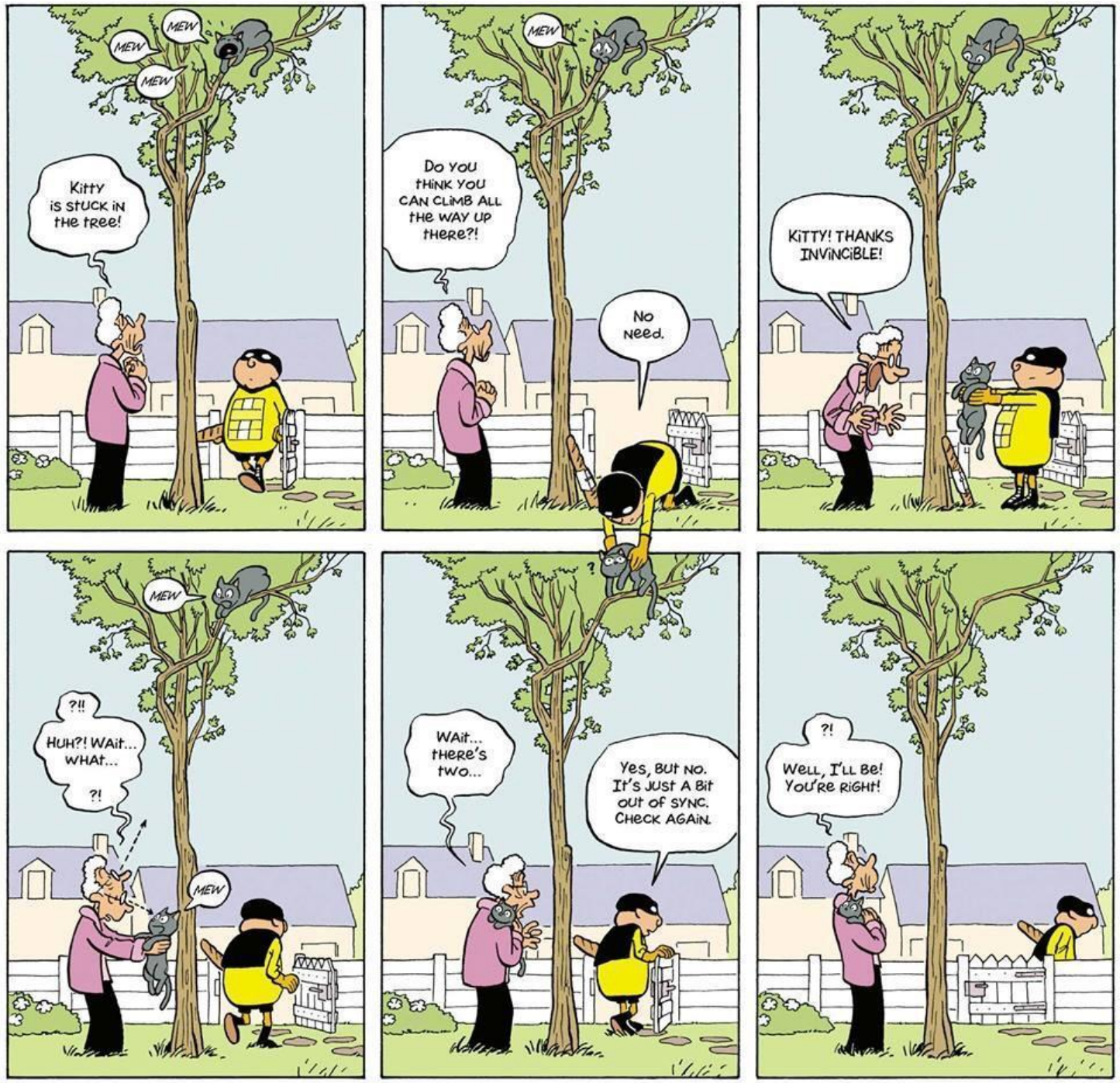
1. Staleness
   (how old is this update?)
2. Numerical error
   (how large is the impact of this update?)

Does the system require inconsistency to support the current workload?

Inconsistency

In Online Games, state is replicated across multiple machines

Changing state takes time to propagate to all replicas



Imbattable: Justice et légumes frais
by Pascal Jousselin

Inconsistency in Online Games

Can lead to bad player experiences:

- Rubber banding
- “shot around the corner”
- Complex failures →

Image: *Razorien/CCP Games*

“Both during and after the fight, players experienced things that don’t happen under normal circumstances,” CCP said in its blog post. “Things like ships disappearing, ships reappearing, ships not appearing in the right systems — even after going through the jump tunnel.”

--- from Polygon.com

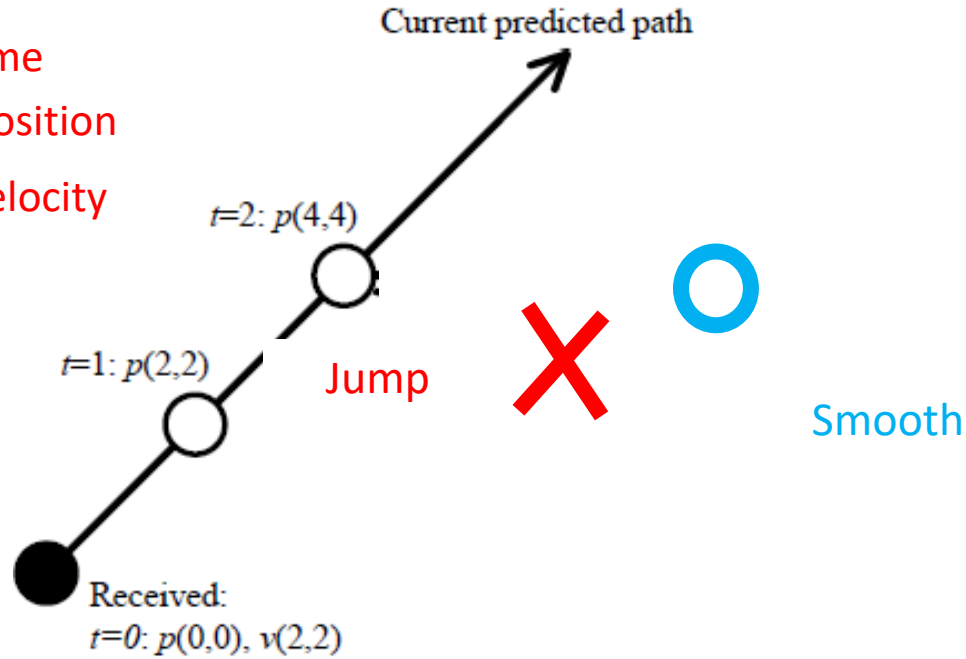
Hiding Inconsistency

Games can hide inconsistencies in entity locations using *dead reckoning*

t = time

p = position

v = velocity

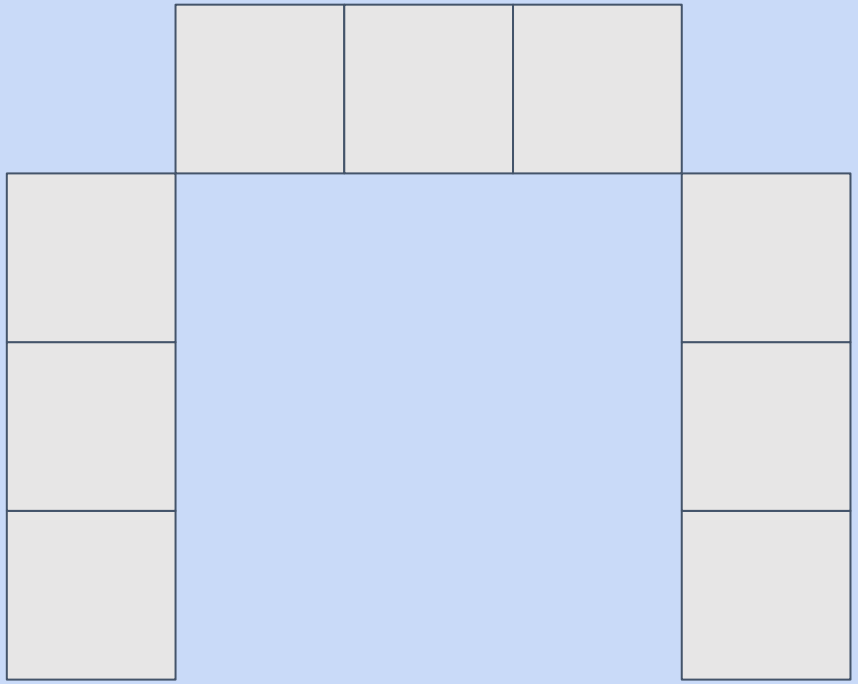


Smed et al., A Review of Networking and Multiplayer Computer Games. TR454, 2002.
http://staff.cs.utu.fi/~jounsmed/papers/TR454.pdf

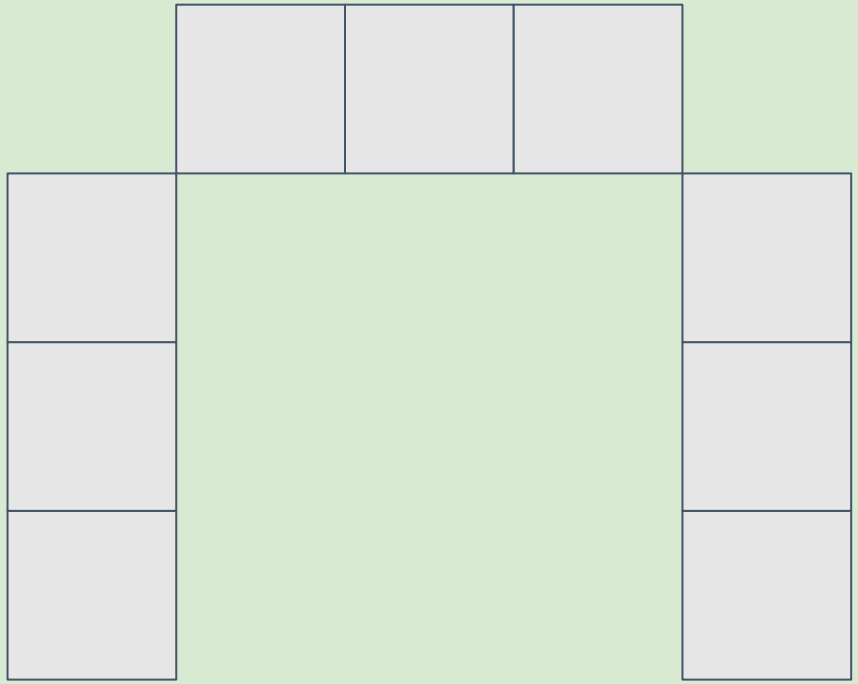
Virtual World without Dyconits

Tick: 1

Server

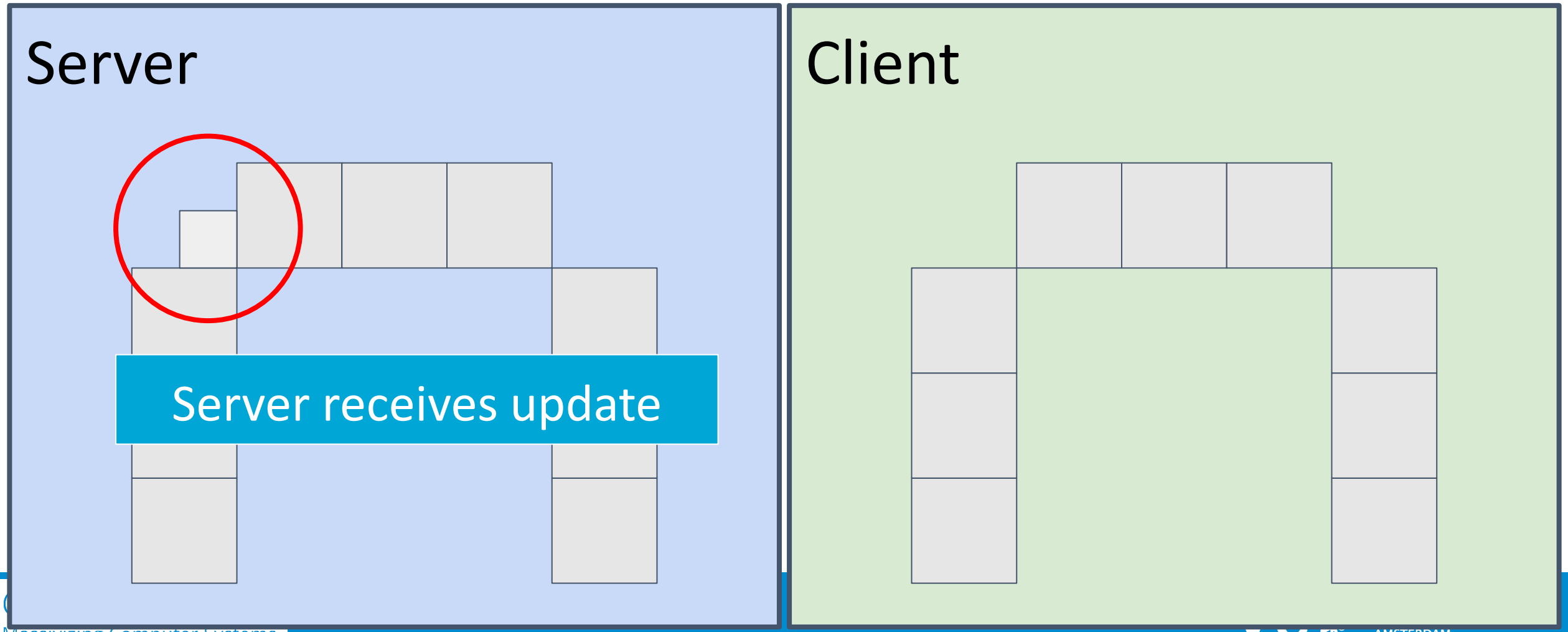


Client



Virtual World without Dyconits

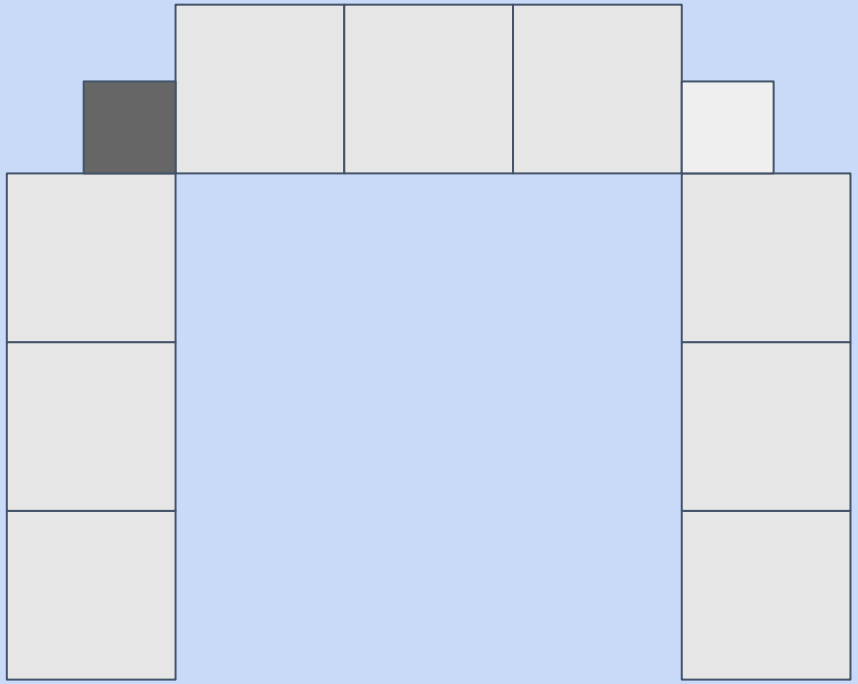
Tick: 2



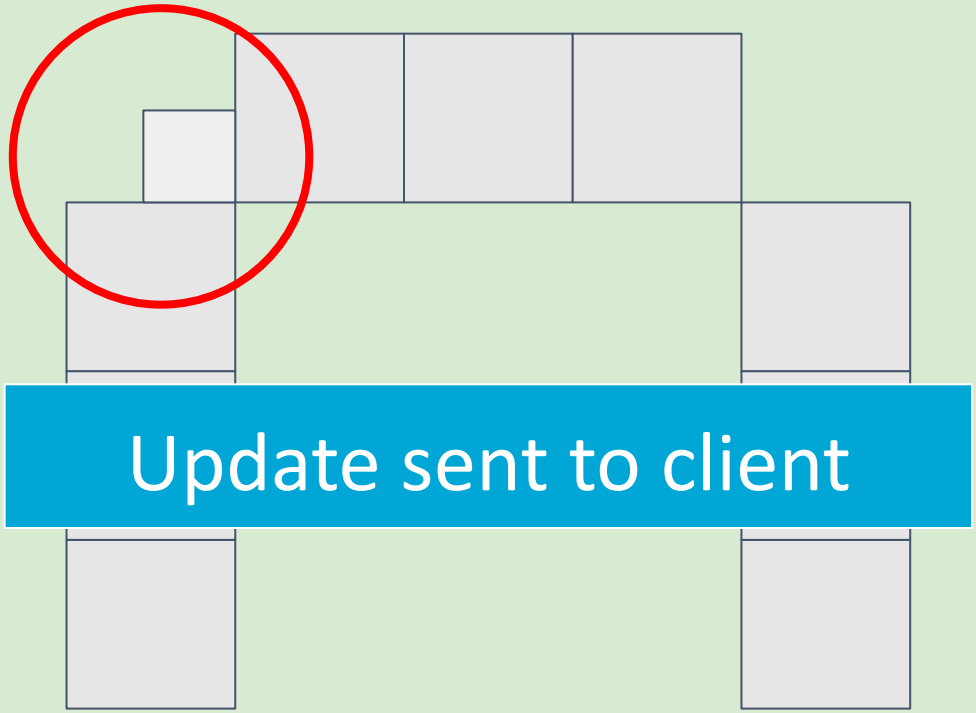
Virtual World without Dyconits

Tick: 3

Server



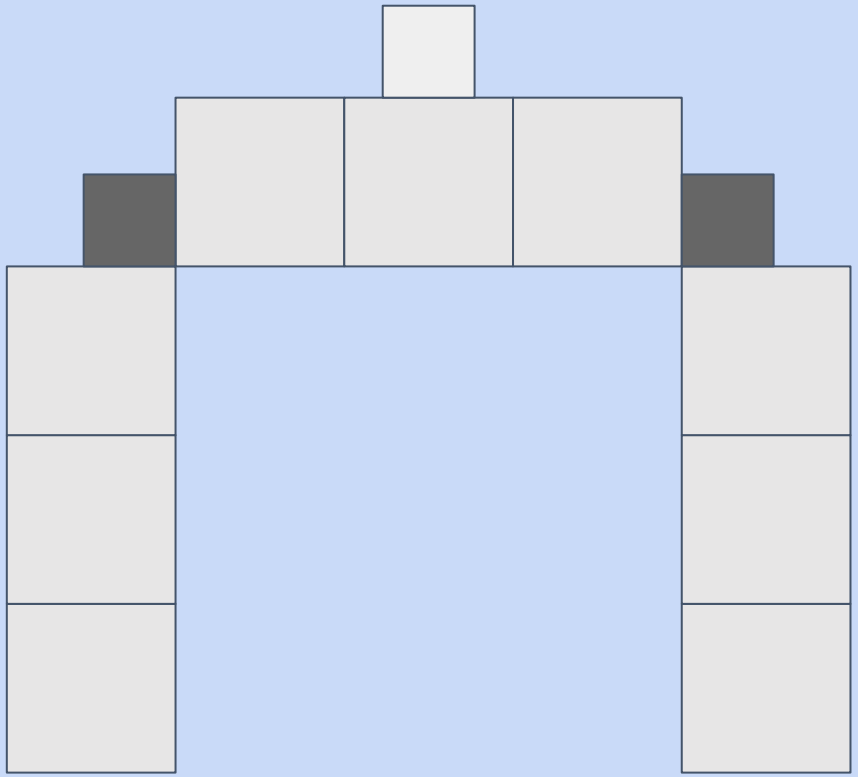
Client



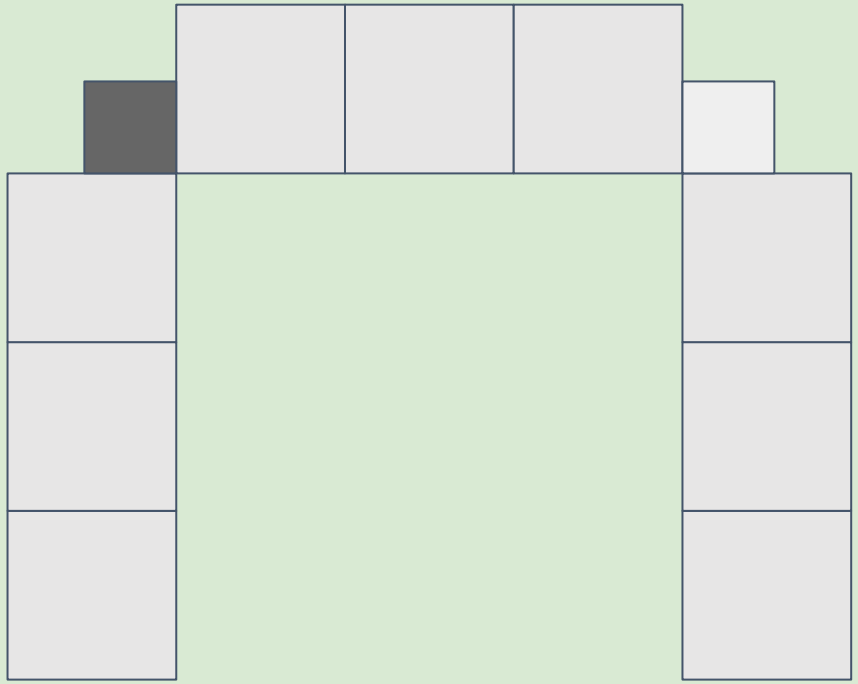
Virtual World without Dyconits

Tick: 4

Server



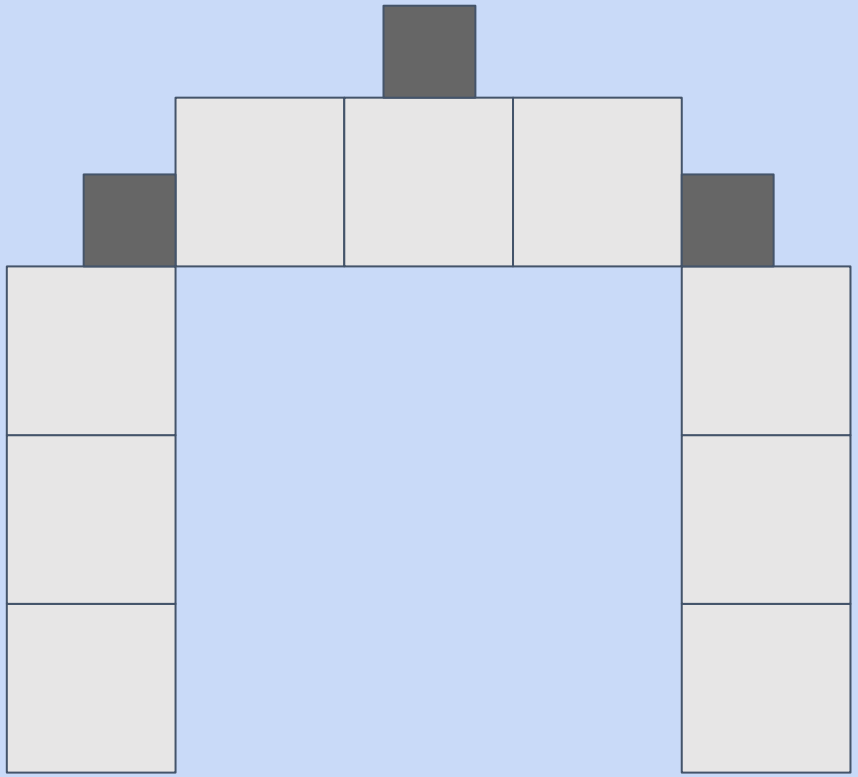
Client



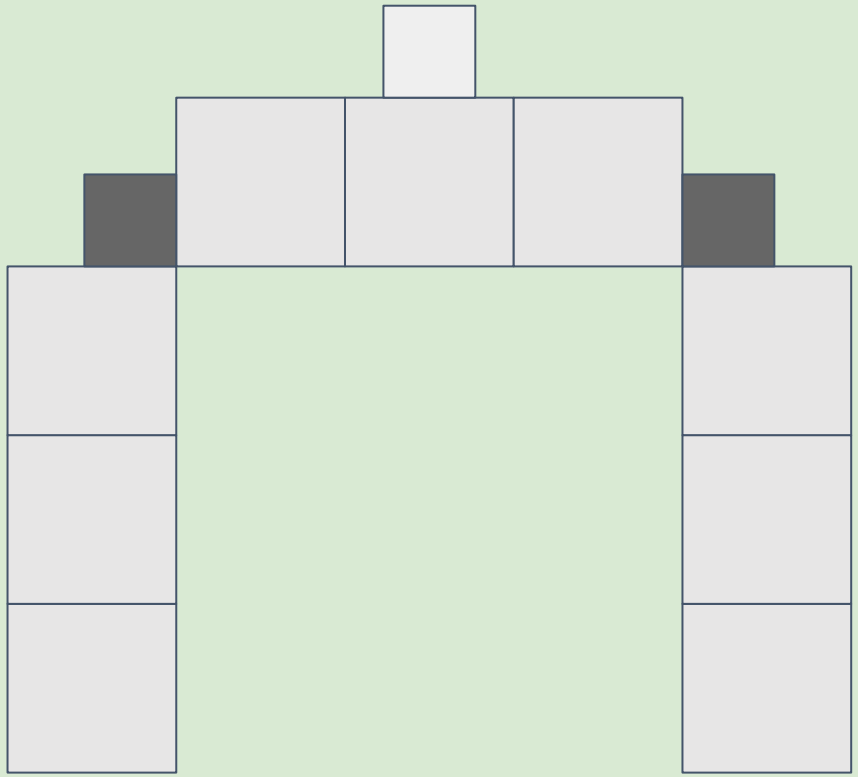
Virtual World without Dyconits

Tick: 5

Server



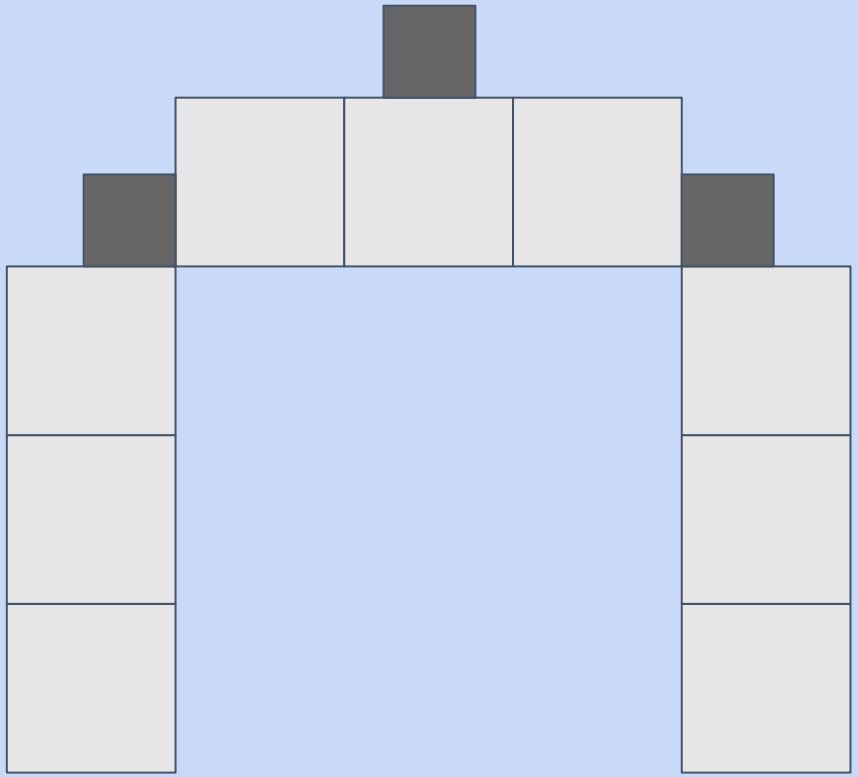
Client



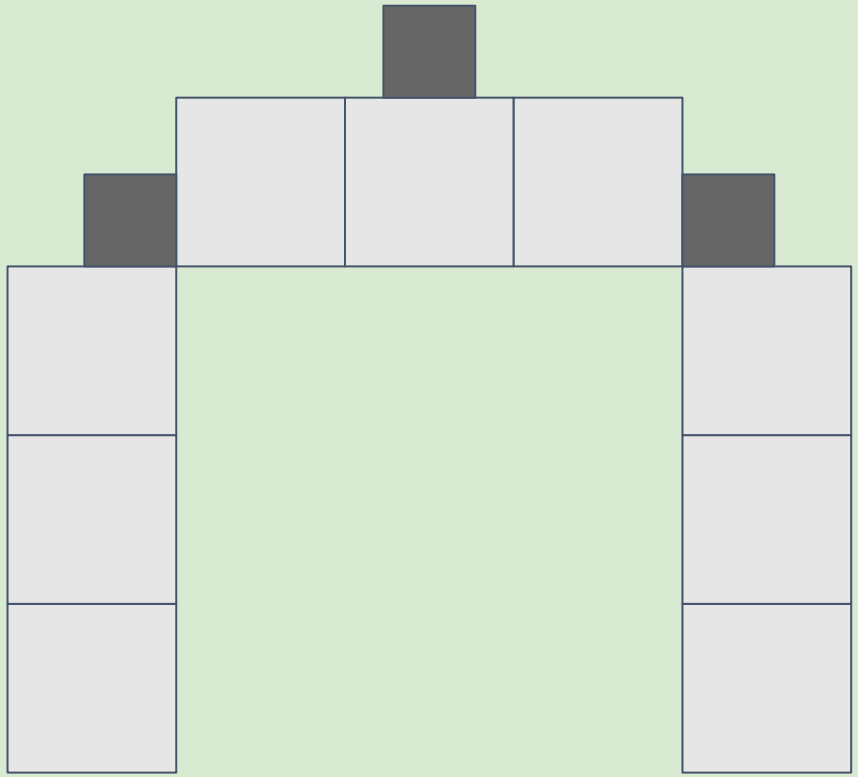
Virtual World without Dyconits

Tick: 5

Server



Client

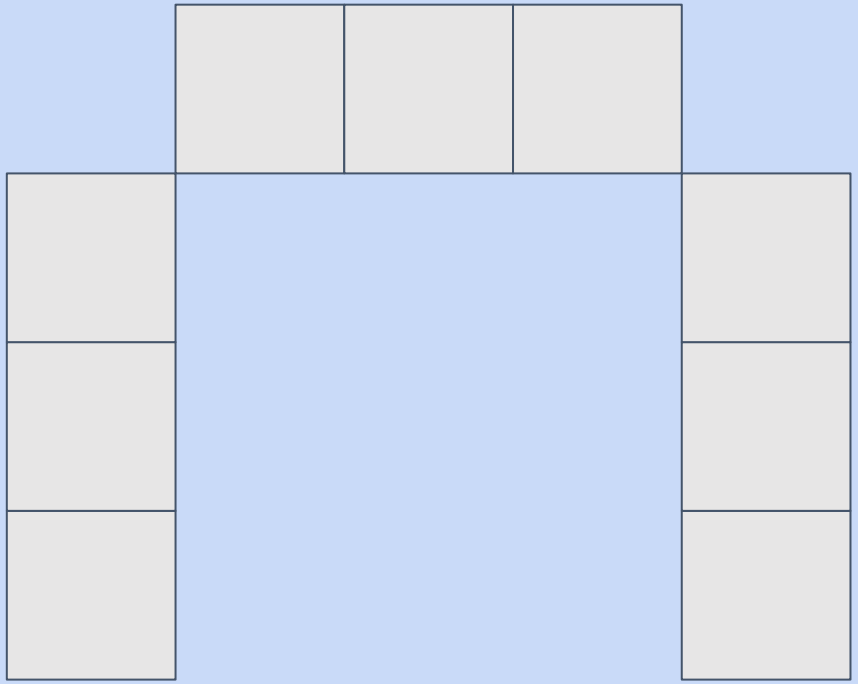


Virtual World with Dyconits

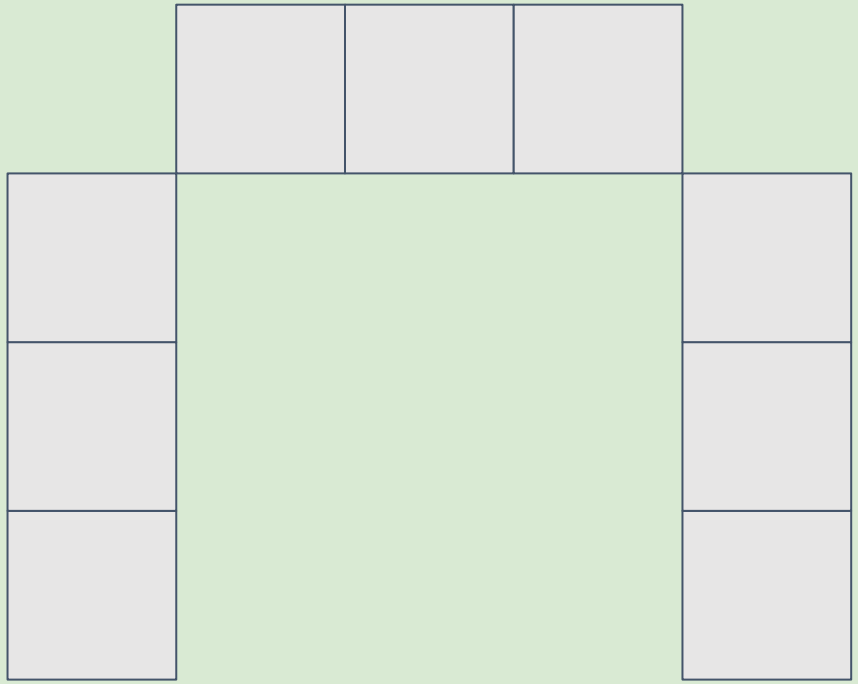
Tick: 1

Numerical Error Bound: 3

Server



Client

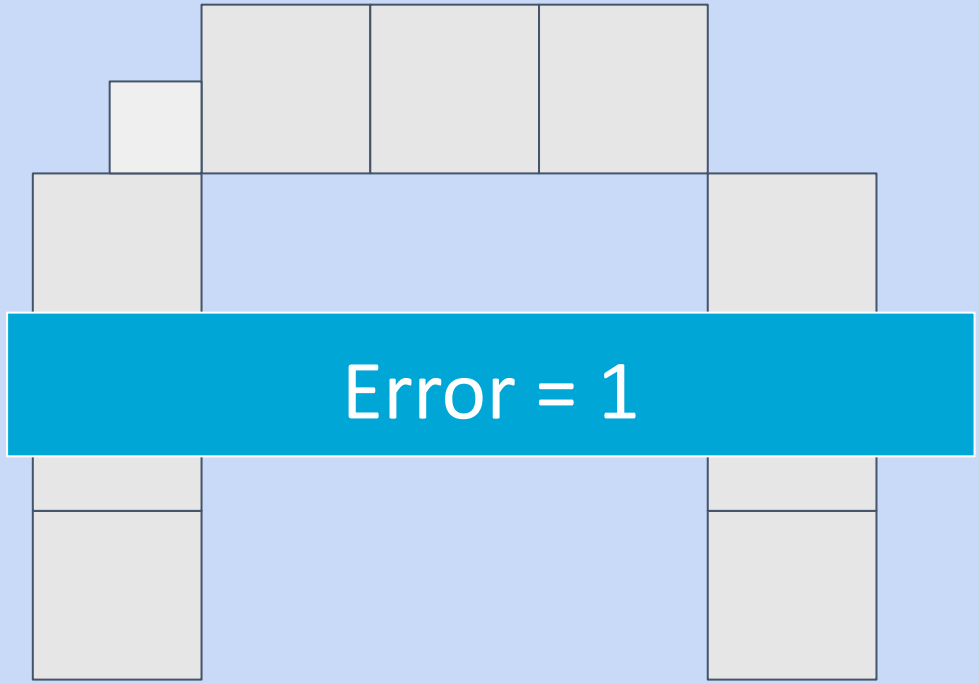


Virtual World with Dyconits

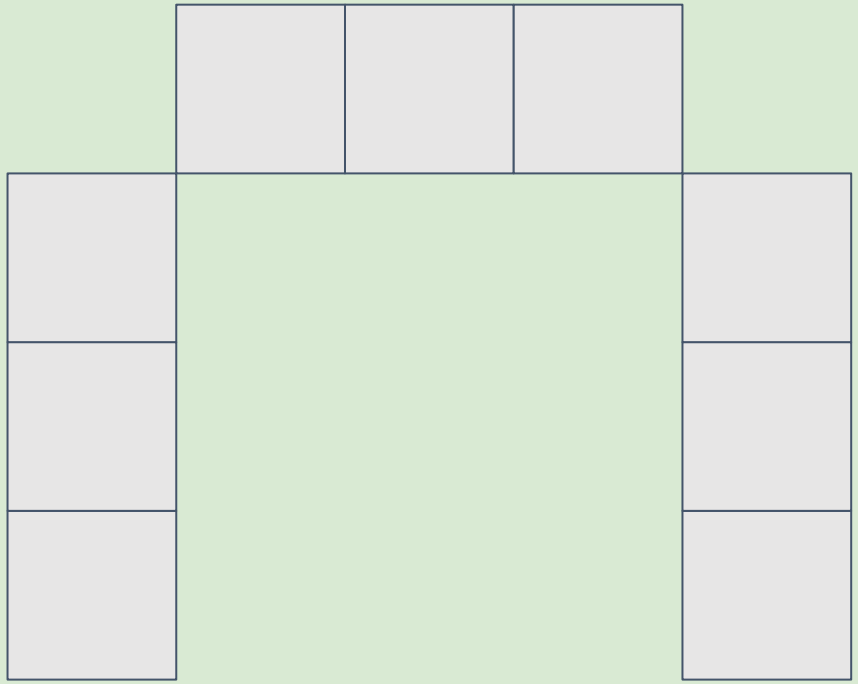
Tick: 1

Numerical Error Bound: 3

Server



Client

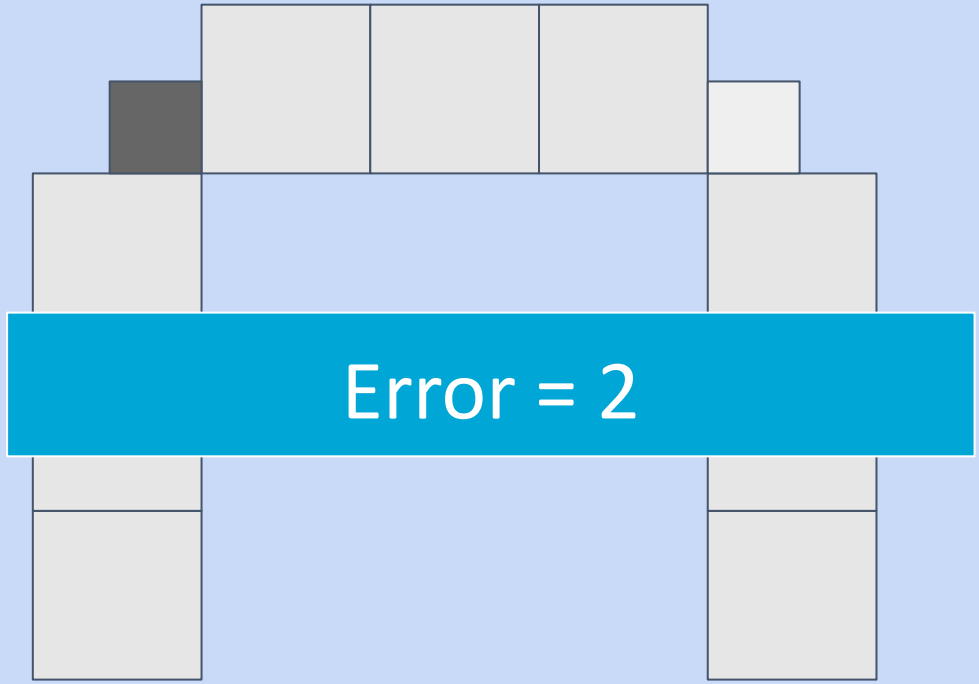


Virtual World with Dyconits

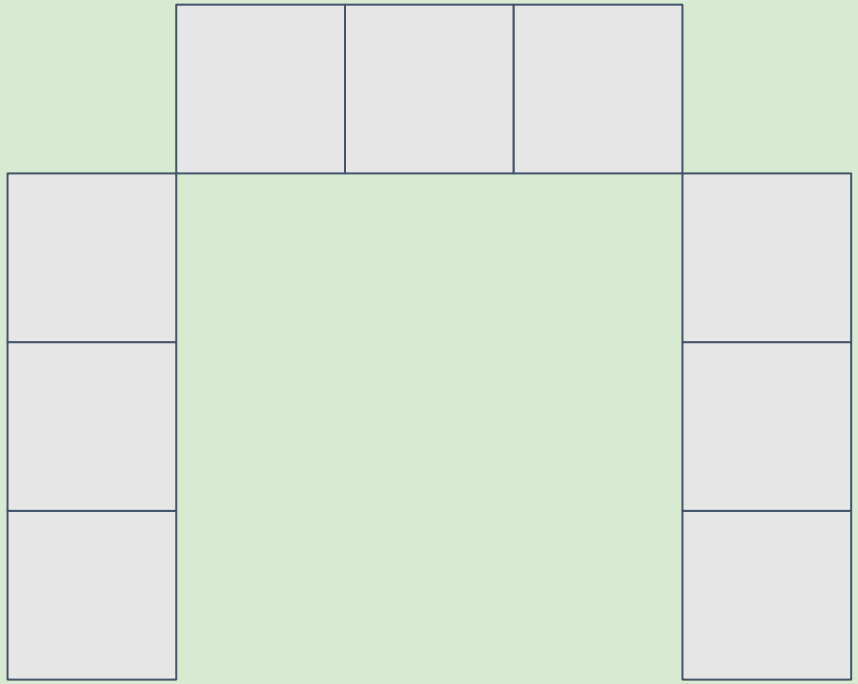
Tick: 2

Numerical Error Bound: 3

Server



Client

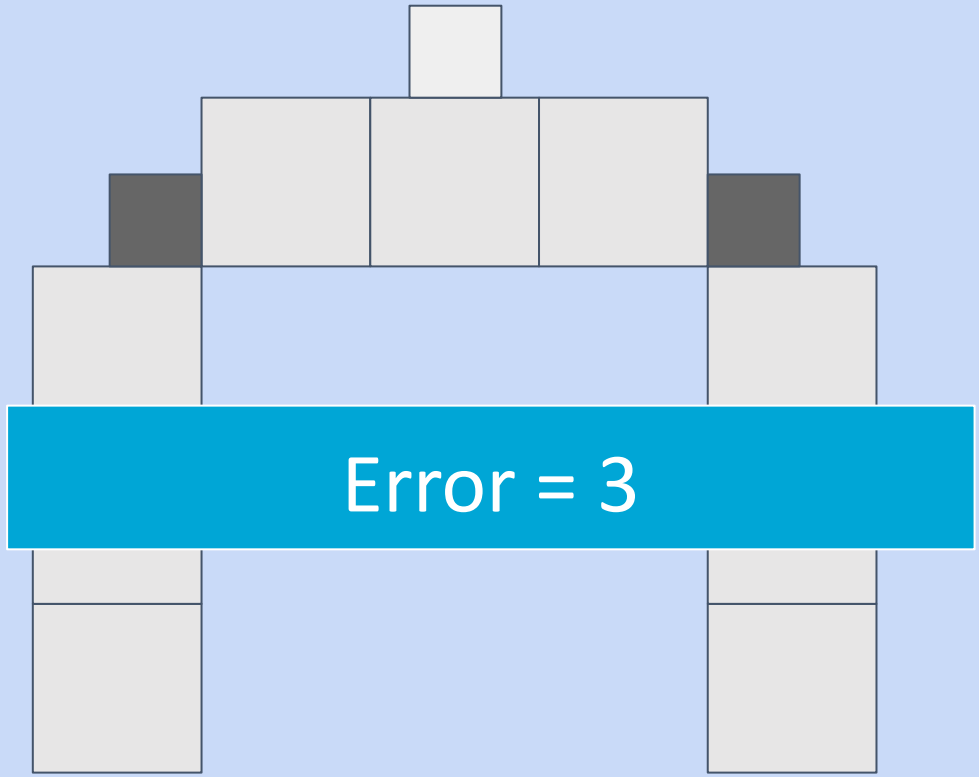


Virtual World with Dyconits

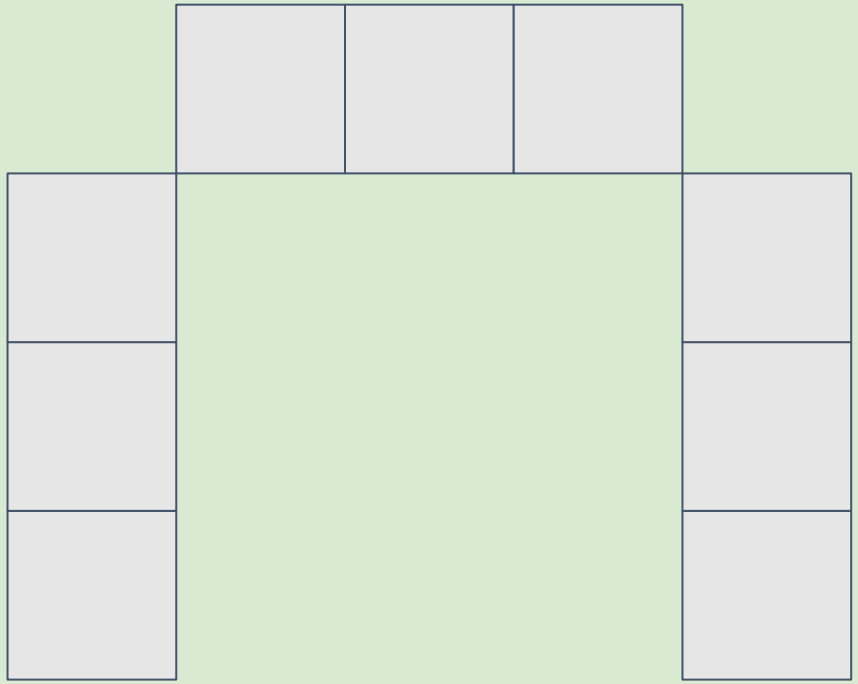
Tick: 3

Numerical Error Bound: 3

Server



Client

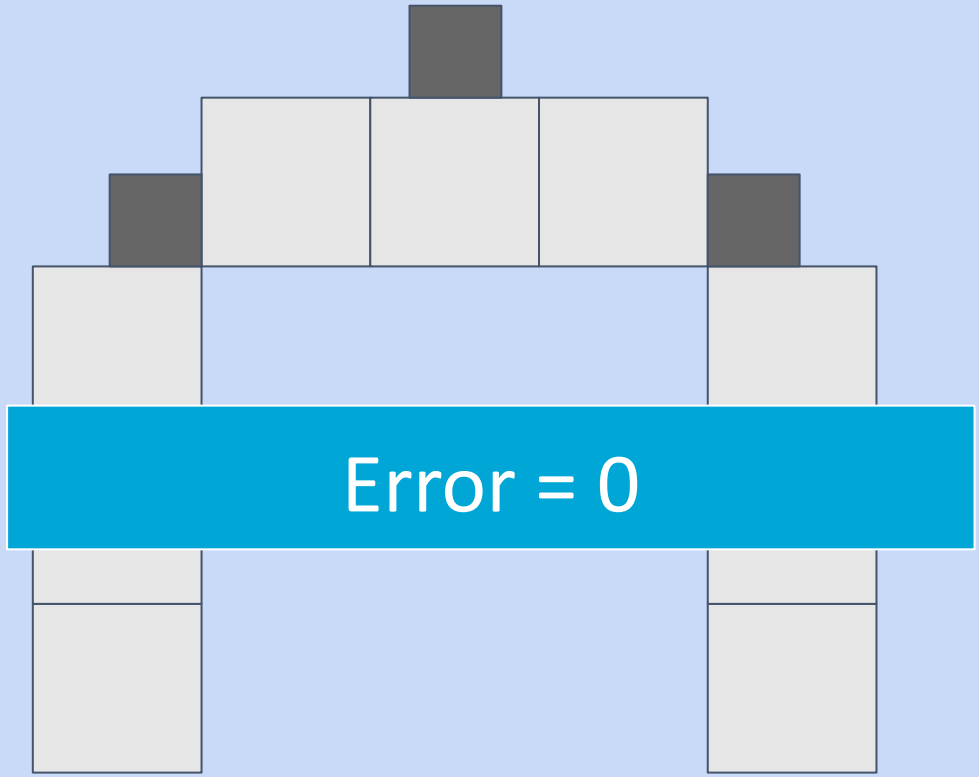


Virtual World with Dyconits

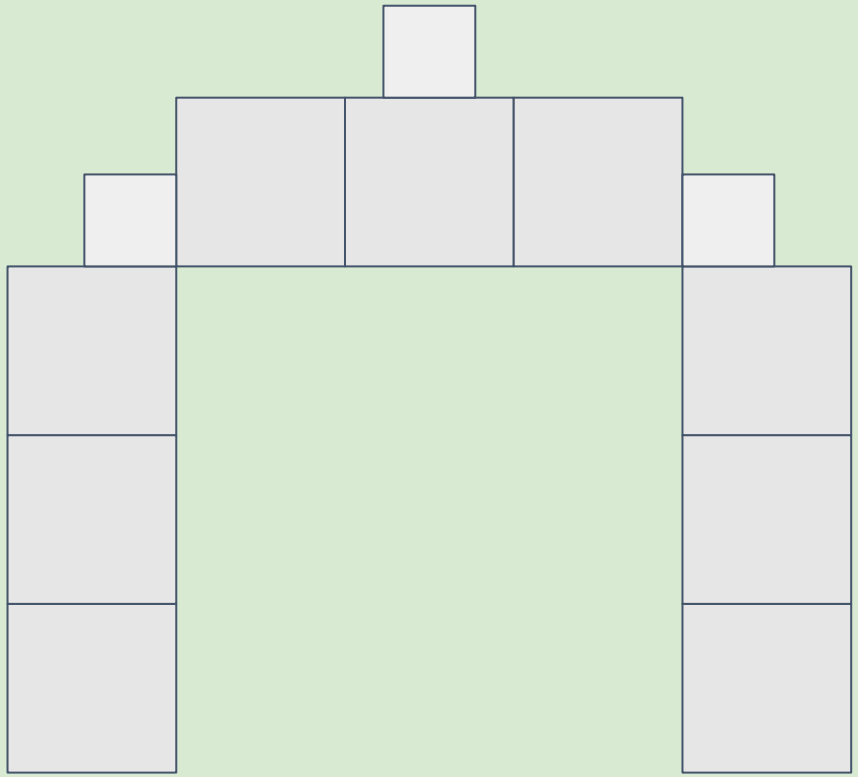
Tick: 4

Numerical Error Bound: 3

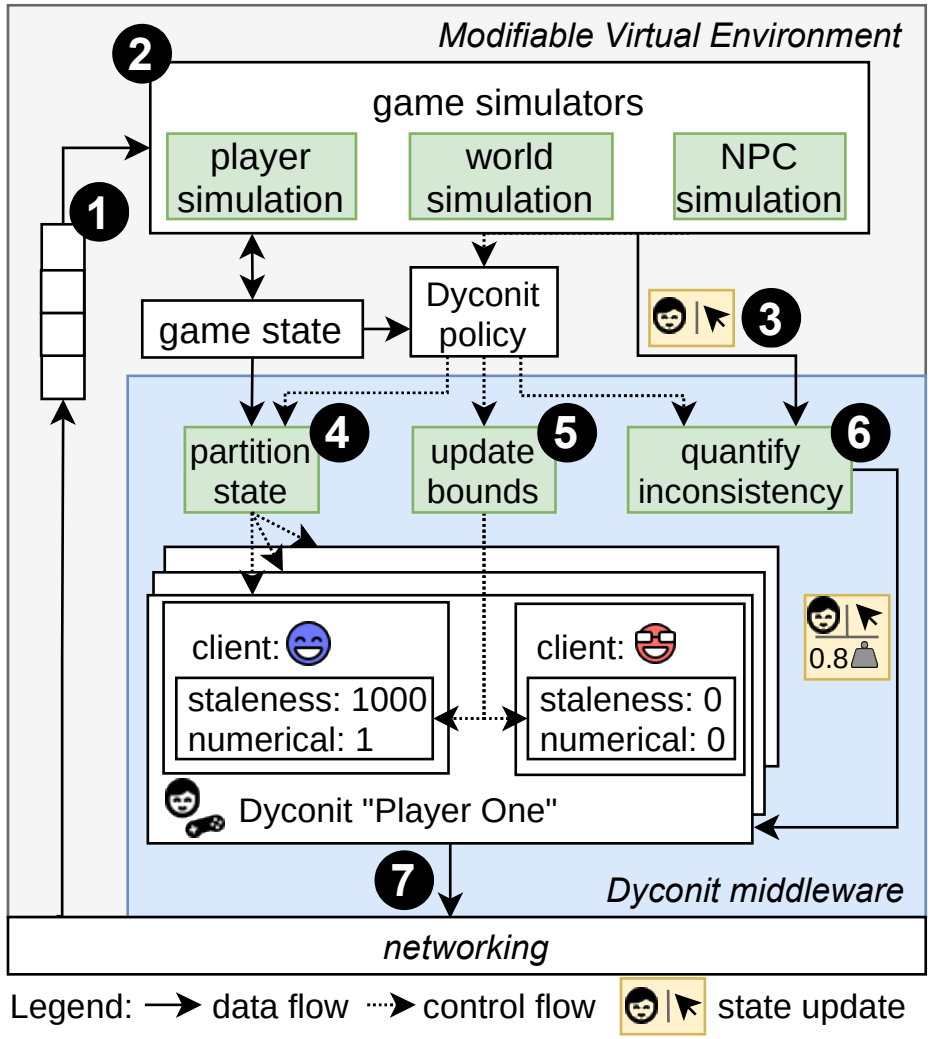
Server



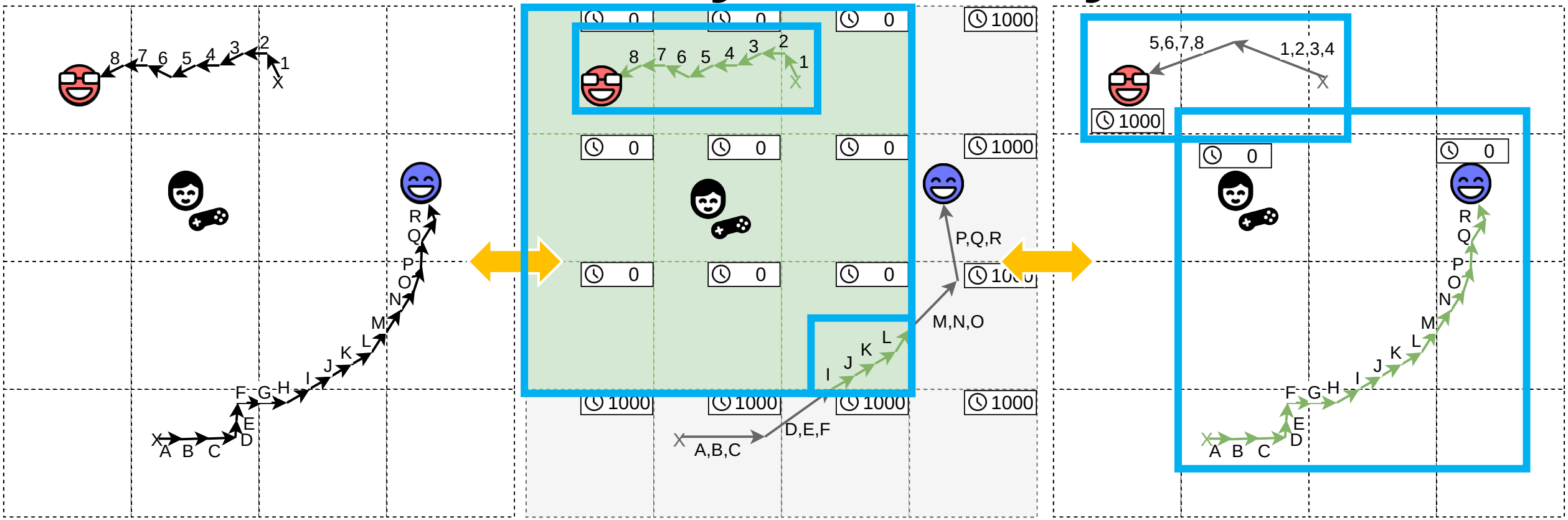
Client



Design of Dyconit Middleware



Dyconits: Switch Policies Dynamically



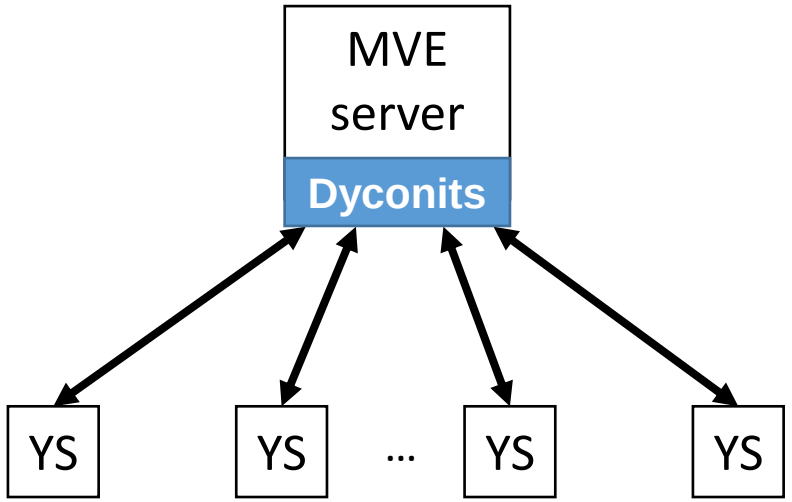
(a) Baseline policy (ZERO)

(b) Area Of Interest policy

(b) Interest Set policy (IS)

Experiment Setup

Real-world experiments on DAS-5 super computer
One node for MVE server, one node per 50 emulated clients

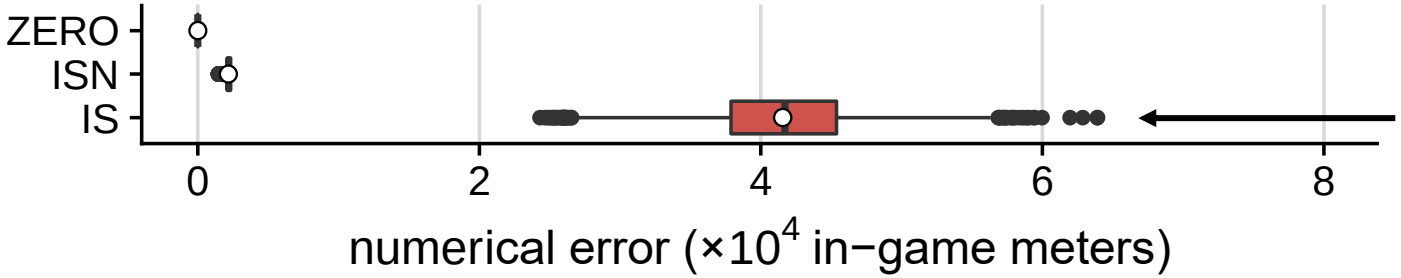


= 1 physical node YS = 50 emulated players

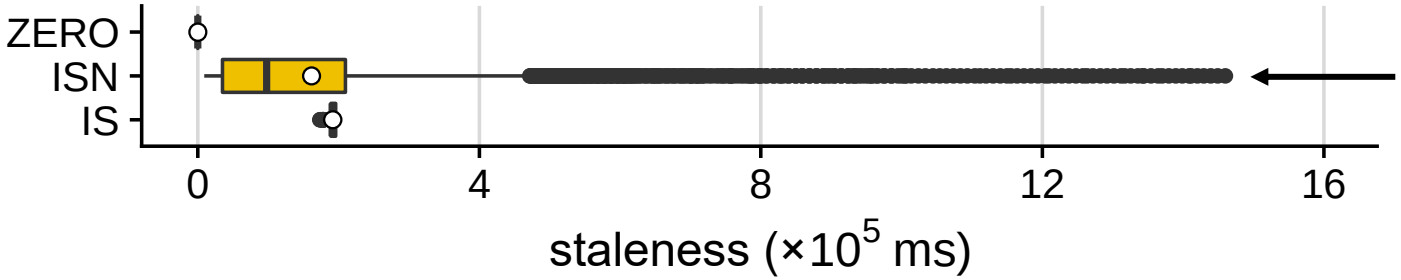
Each node equipped with:

- dual 8-core 2.4 GHz CPU
- 64 GiB of memory
- InfiniBand network with max. throughput of 48 Gbps

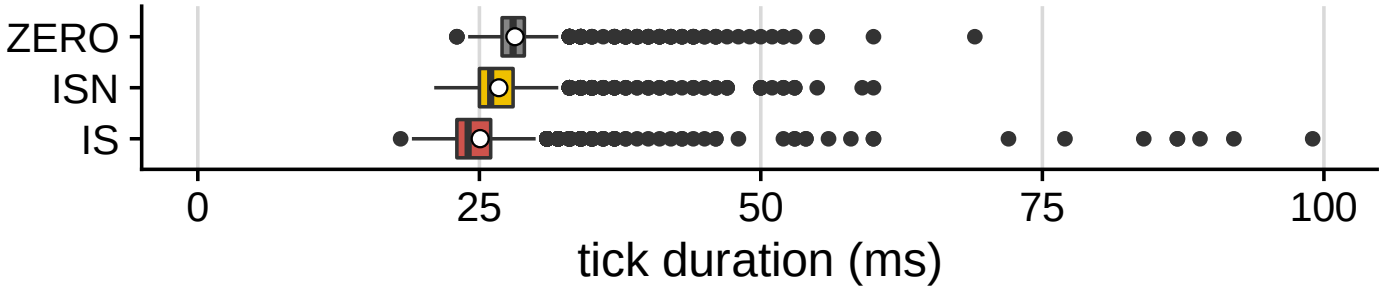
Dyconits Bound Inconsistency



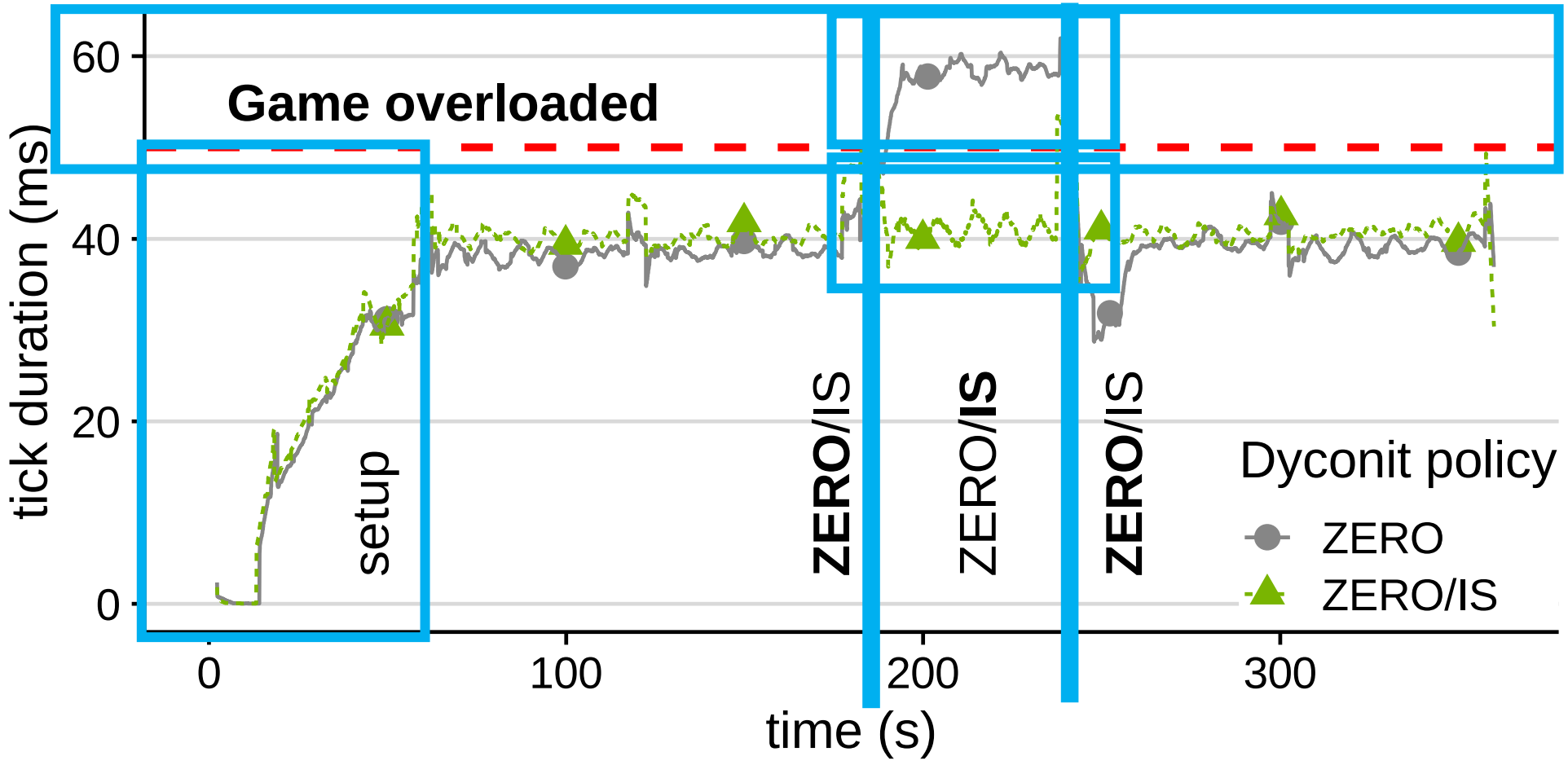
≈0.55 to 1.5 meters error per player per avatar



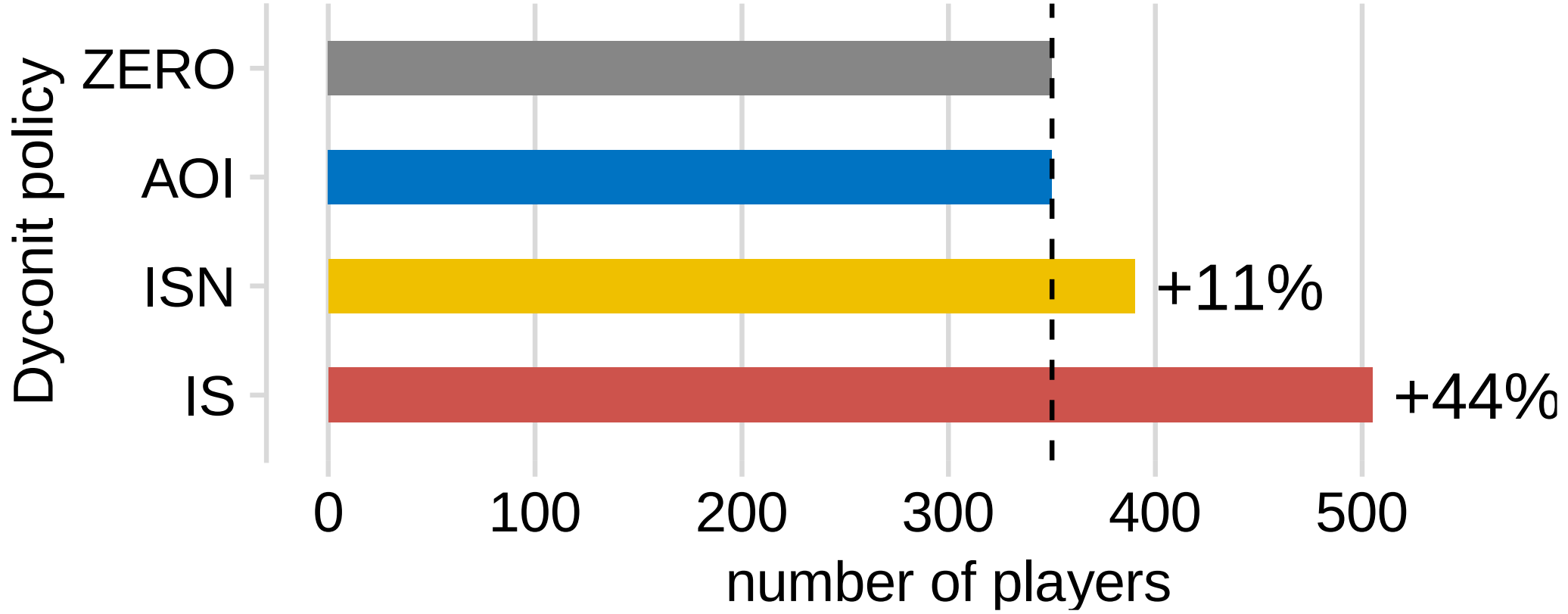
Up to 7.5 seconds error per player



Dyconits Can Dynamically Trade off Consistency for Performance

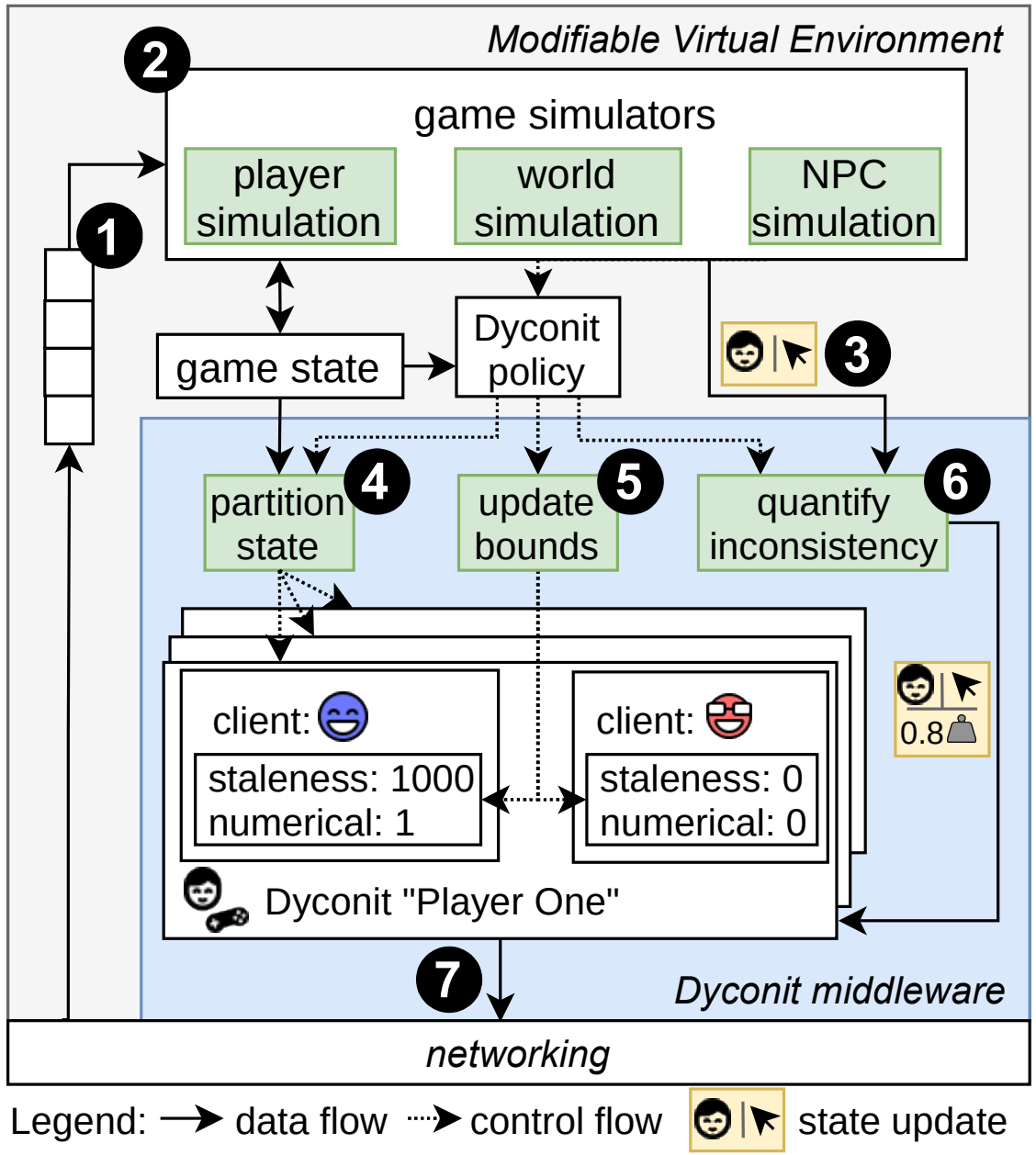


Dyconits Improve Scalability



Main Contributions

1. Design of *Dyconits* to address scalability issues
2. Prototype of a Minecraft-like game using Dyconits; Game and Dyconits code publicly available
3. Real-world experiments to evaluate scalability

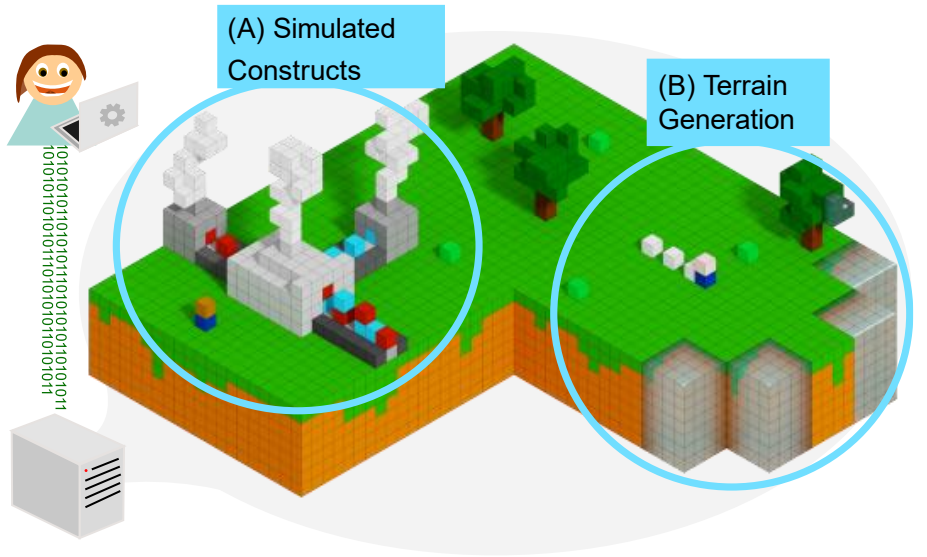


Opencraft: https://github.com/atlarge-research/opencraft
Dyconits: https://github.com/atlarge-research/opencraft-dyconits

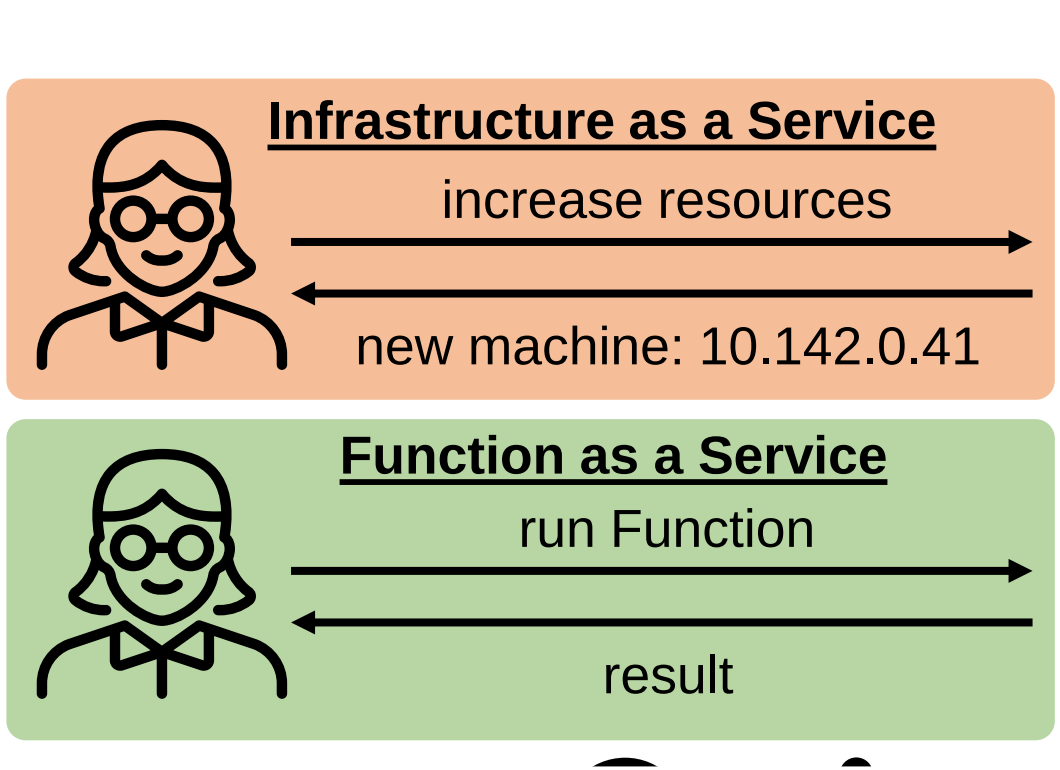
Servo: Increasing the Scalability of Modifiable Virtual Environments Using Serverless Computing

Jesse Donkervliet, Javier Ron, Junyan Li, Tiberiu Iancu, Cristina L. Abad, Alexandru Iosup

j.j.r.donkervliet@vu.nl
@jdonkervliet
https://www.jdonkervliet.com

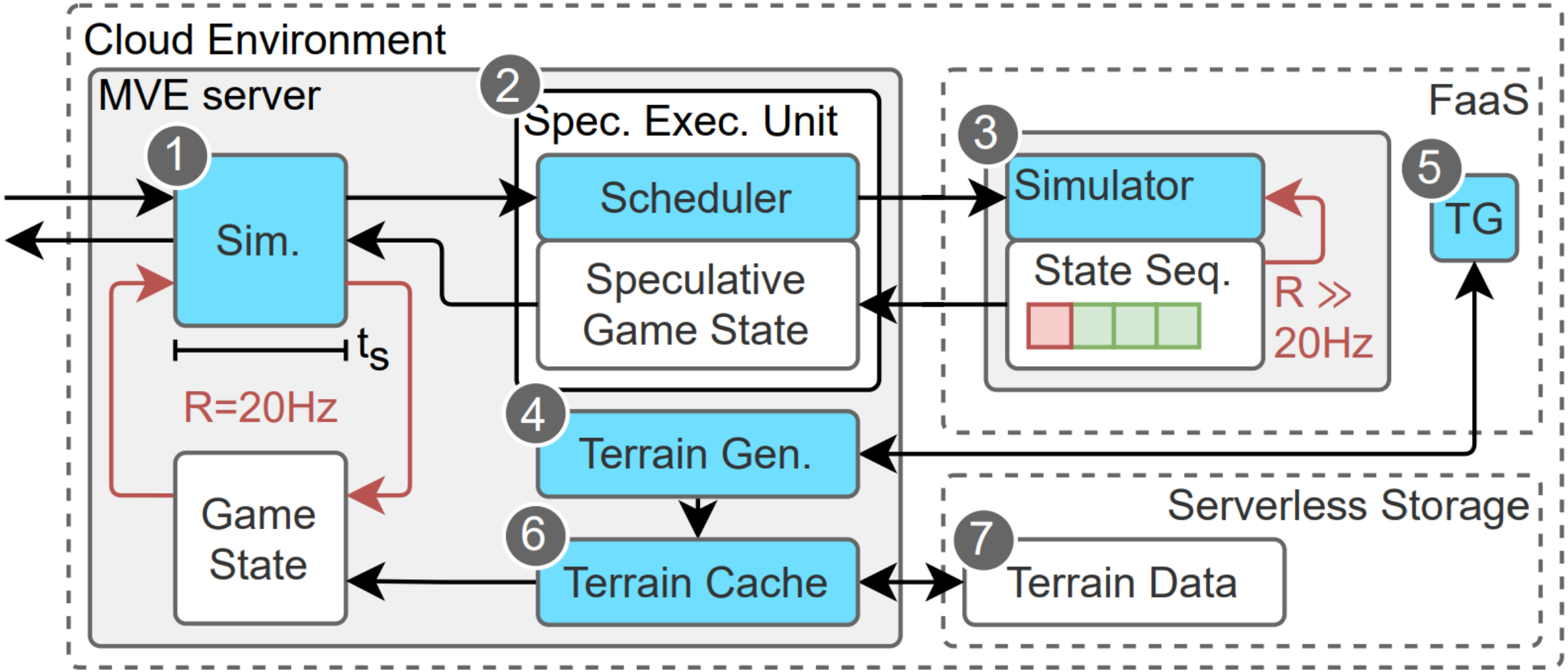


What is Function as a Service (FaaS)?

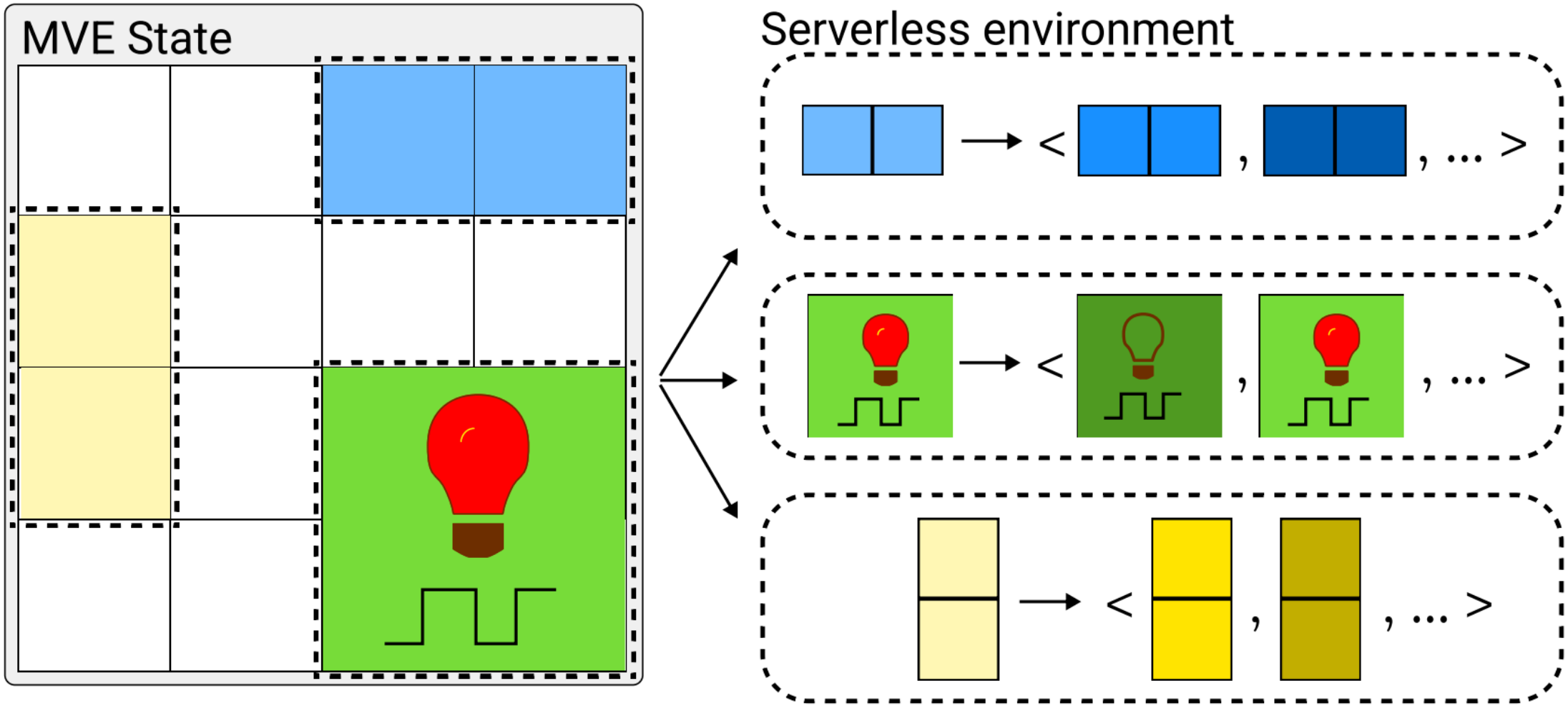


How to leverage serverless for online games?

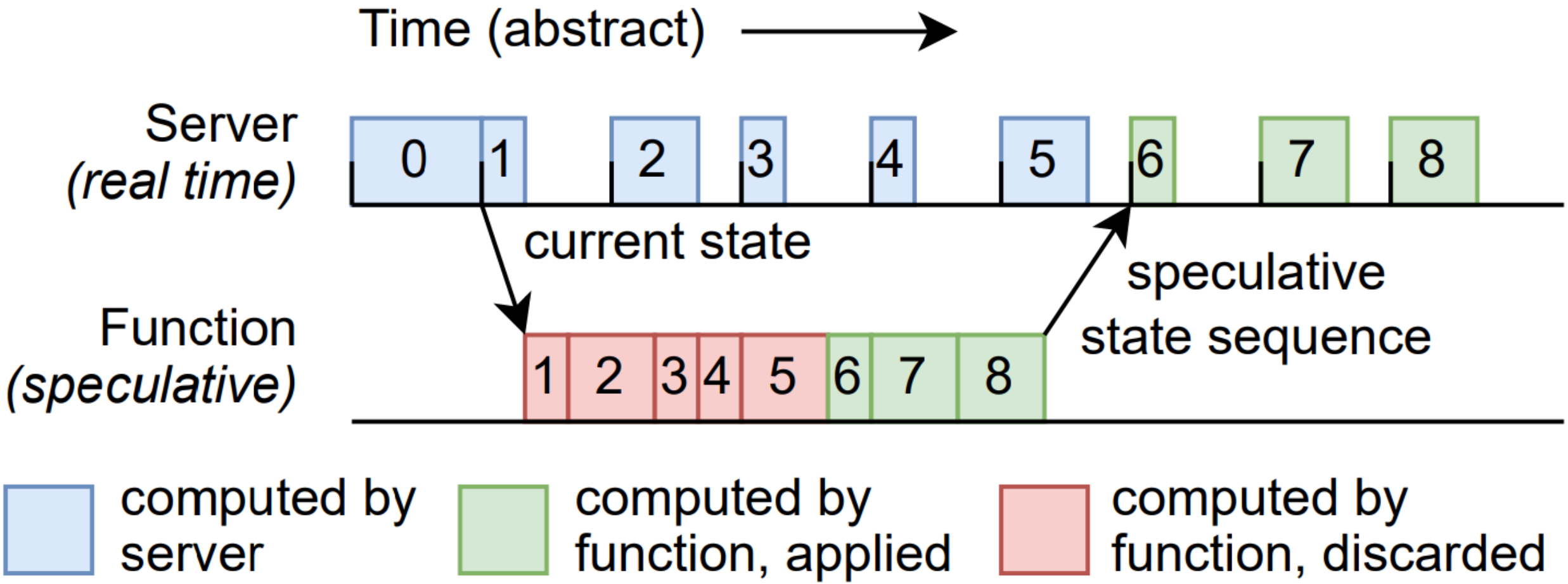
Servo System Overview



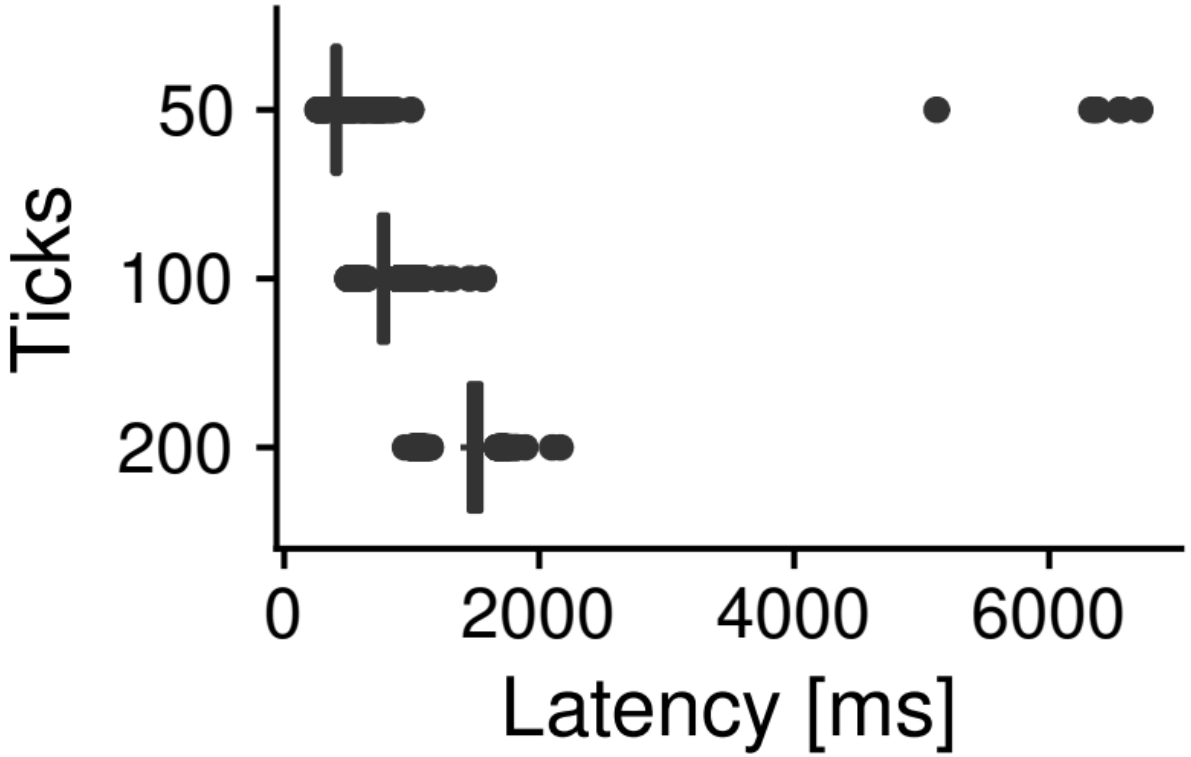
Fine-Grained Comp. Offloading



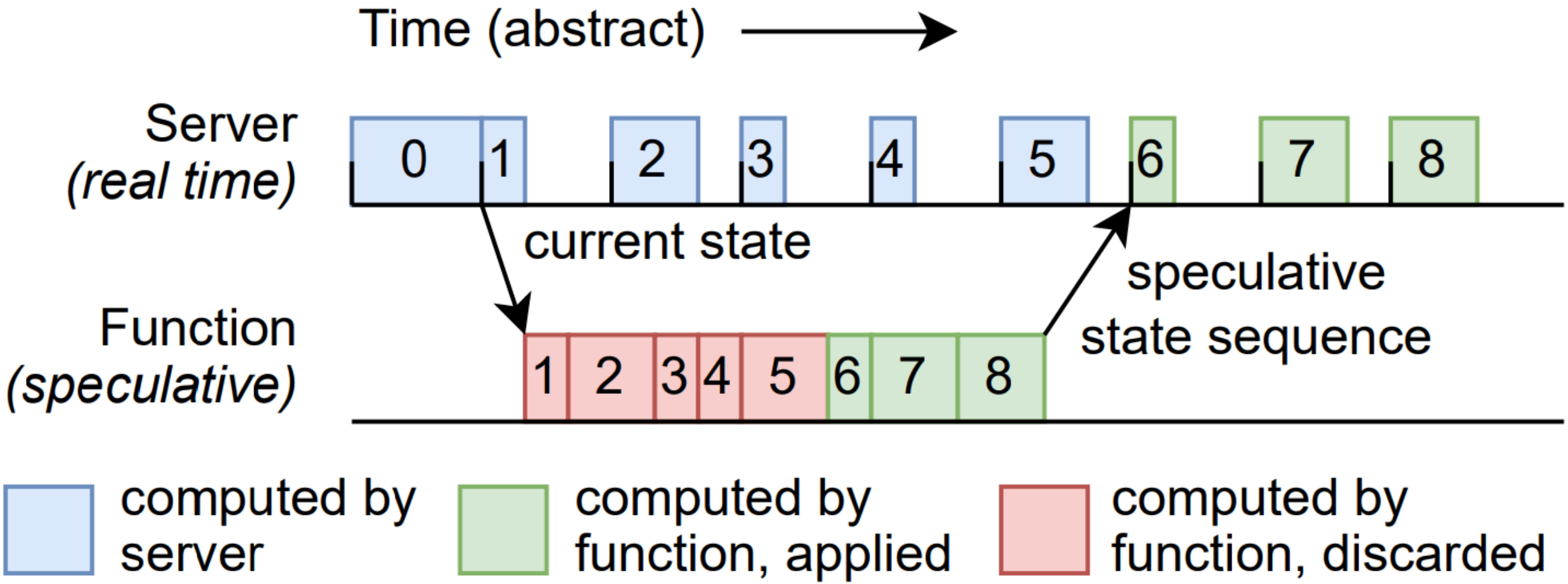
Speculative Offloading



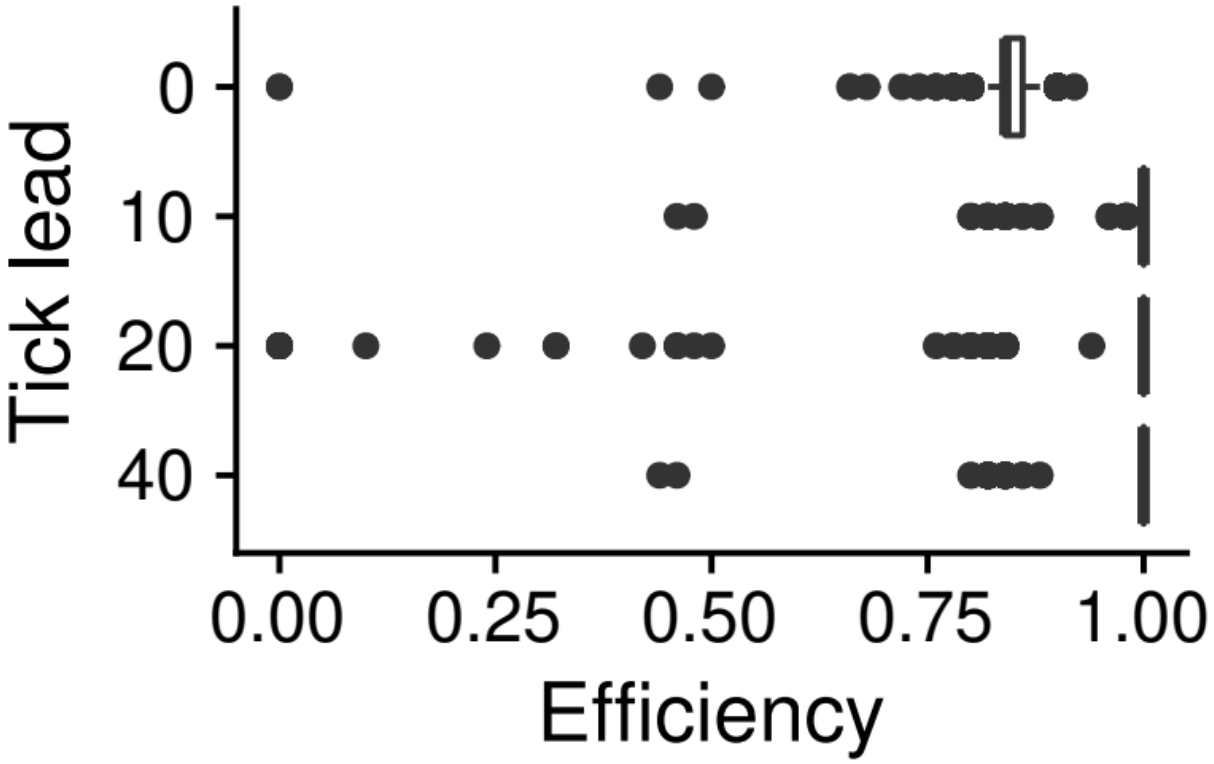
Trade-off between Latency and Number of Invocations



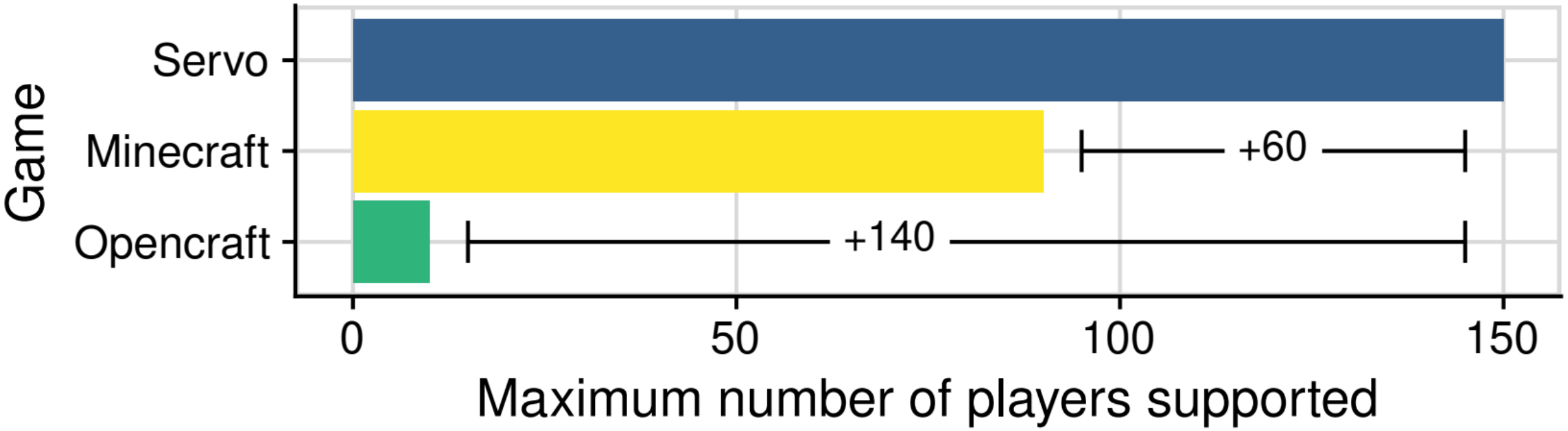
Speculative Offloading



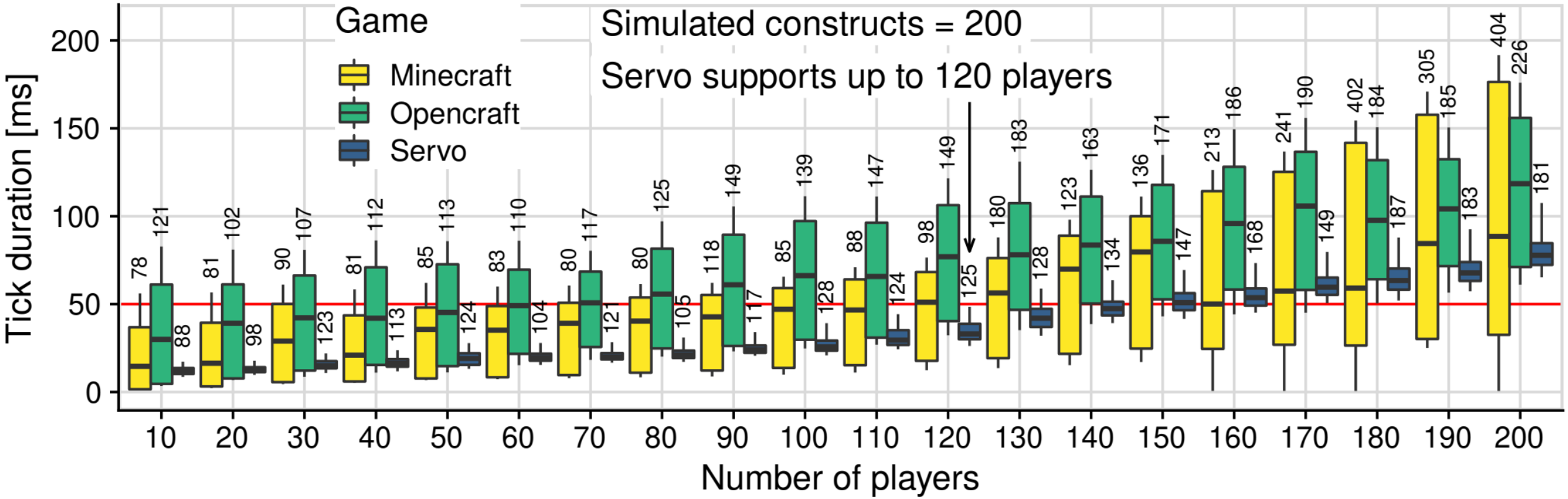
Invoke Early to Improve Efficiency



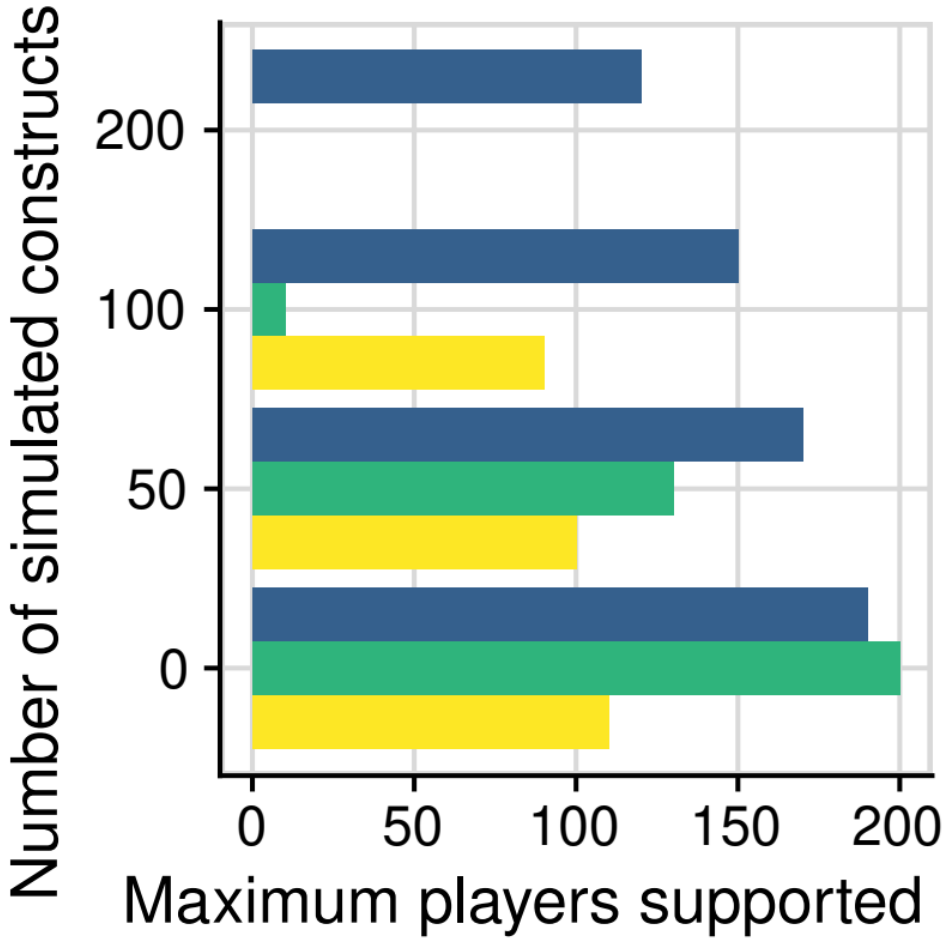
Servo Can Increase Virtual World Scalability



Servo Can Increase Virtual World Complexity



Servo can Scale Computationally Intensive Virtual Environments



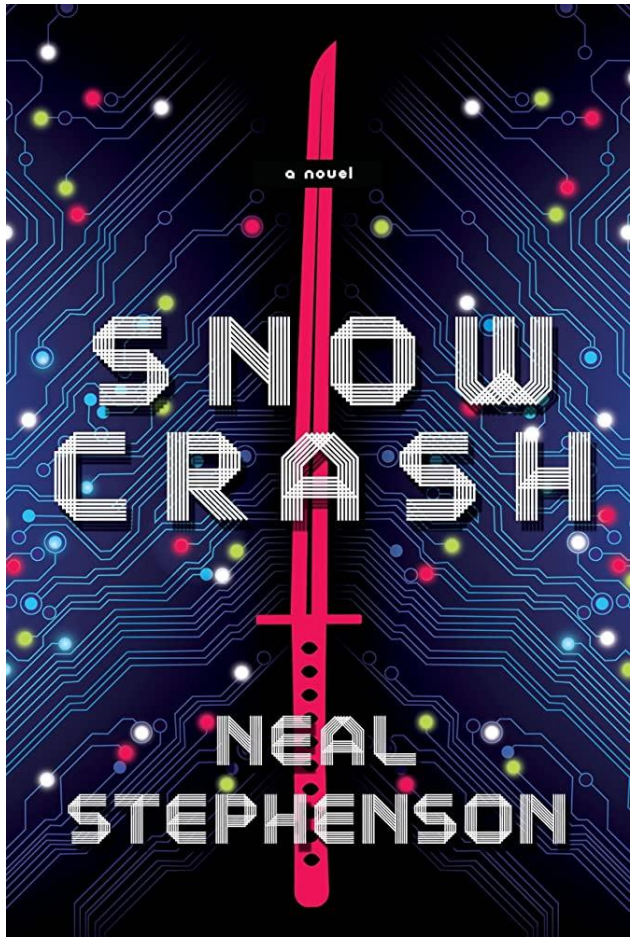
Can My WiFi Handle the Metaverse? A Performance Evaluation Of Meta's Flagship Virtual Reality Hardware

Jesse Donkervliet,* Matthijs Jansen,*
Animesh Trivedi, Alexandru Iosup

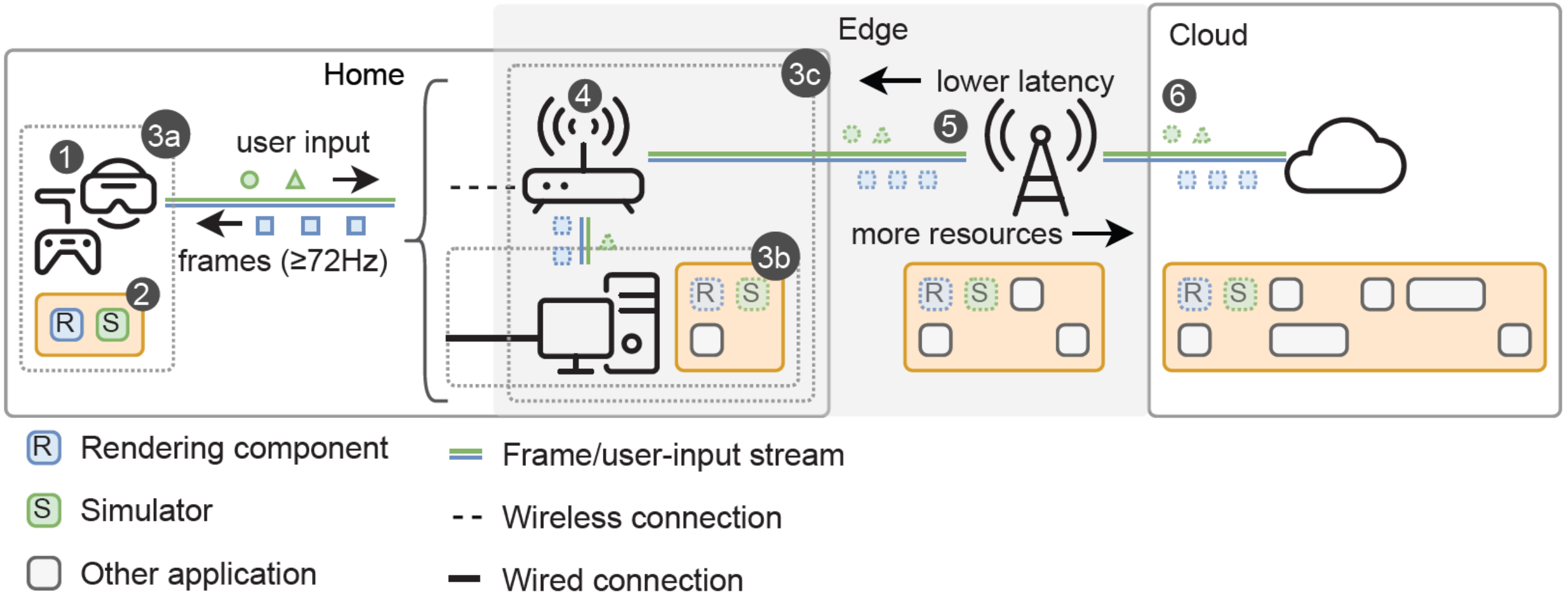
jesse.donkervliet@vu.nl
@jdonkervliet
https://www.jdonkervliet.com

* Both authors contributed equally to this work

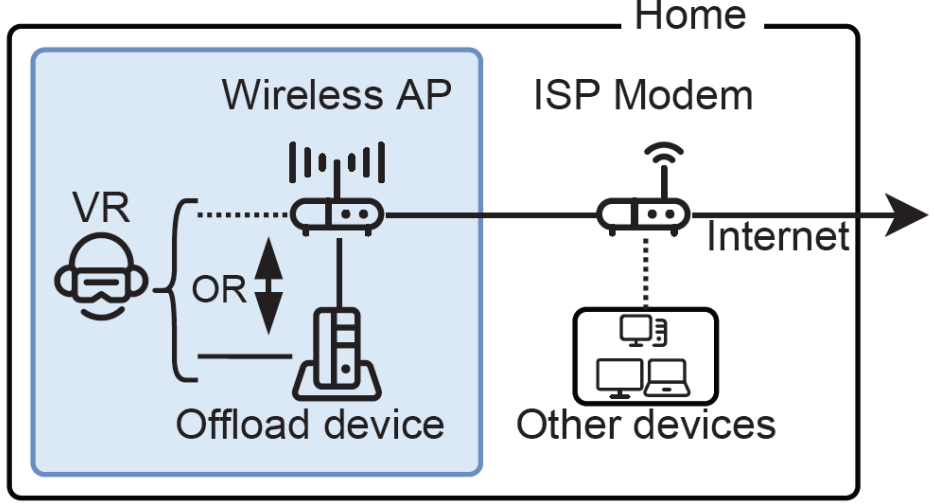

NEUROMANCER
WILLIAM GIBSON


a novel
SNOW CRASH
NEAL STEPHENSON

How to Deploy Metaverse Infrastructure?

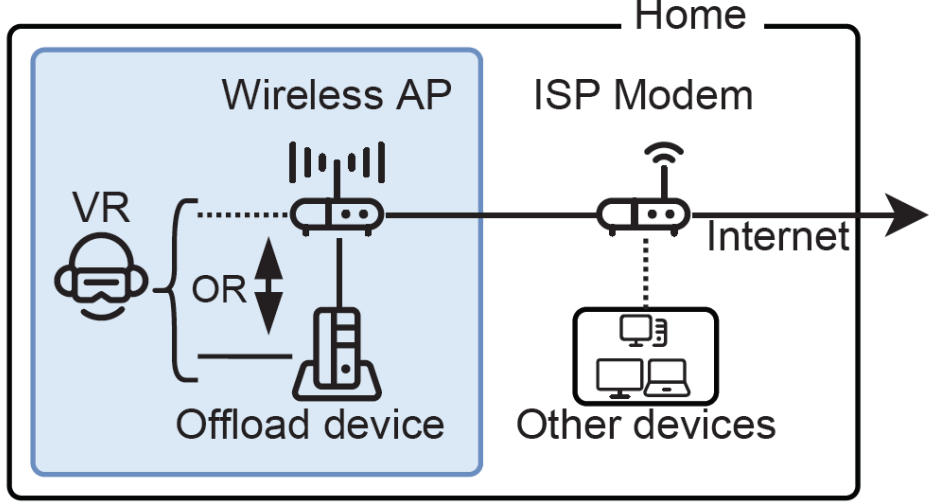


Experiment Setup

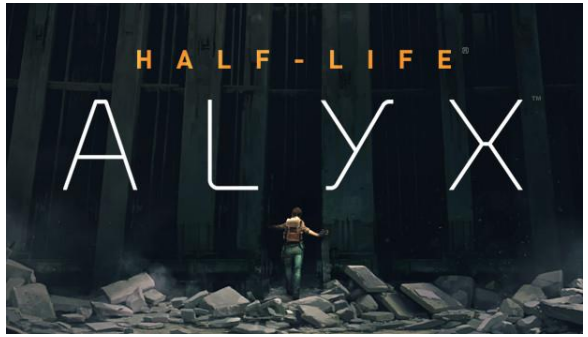


#1 Local

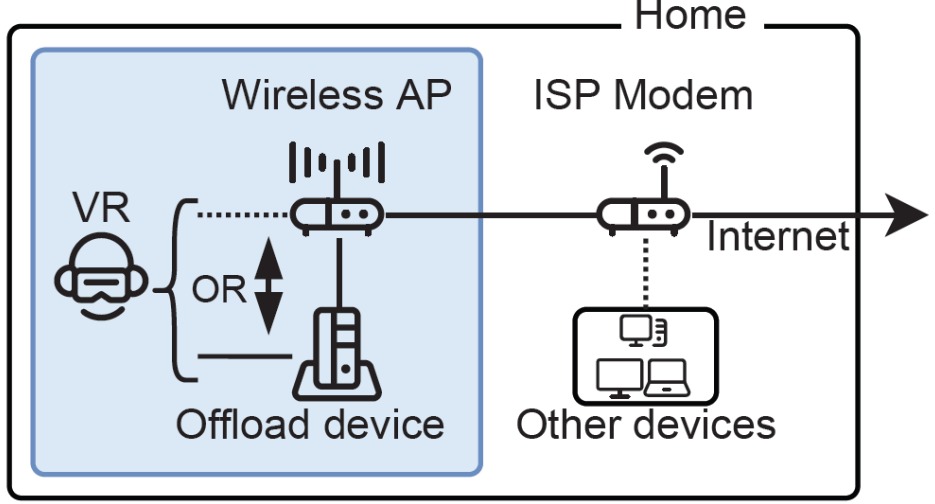
Experiment Setup



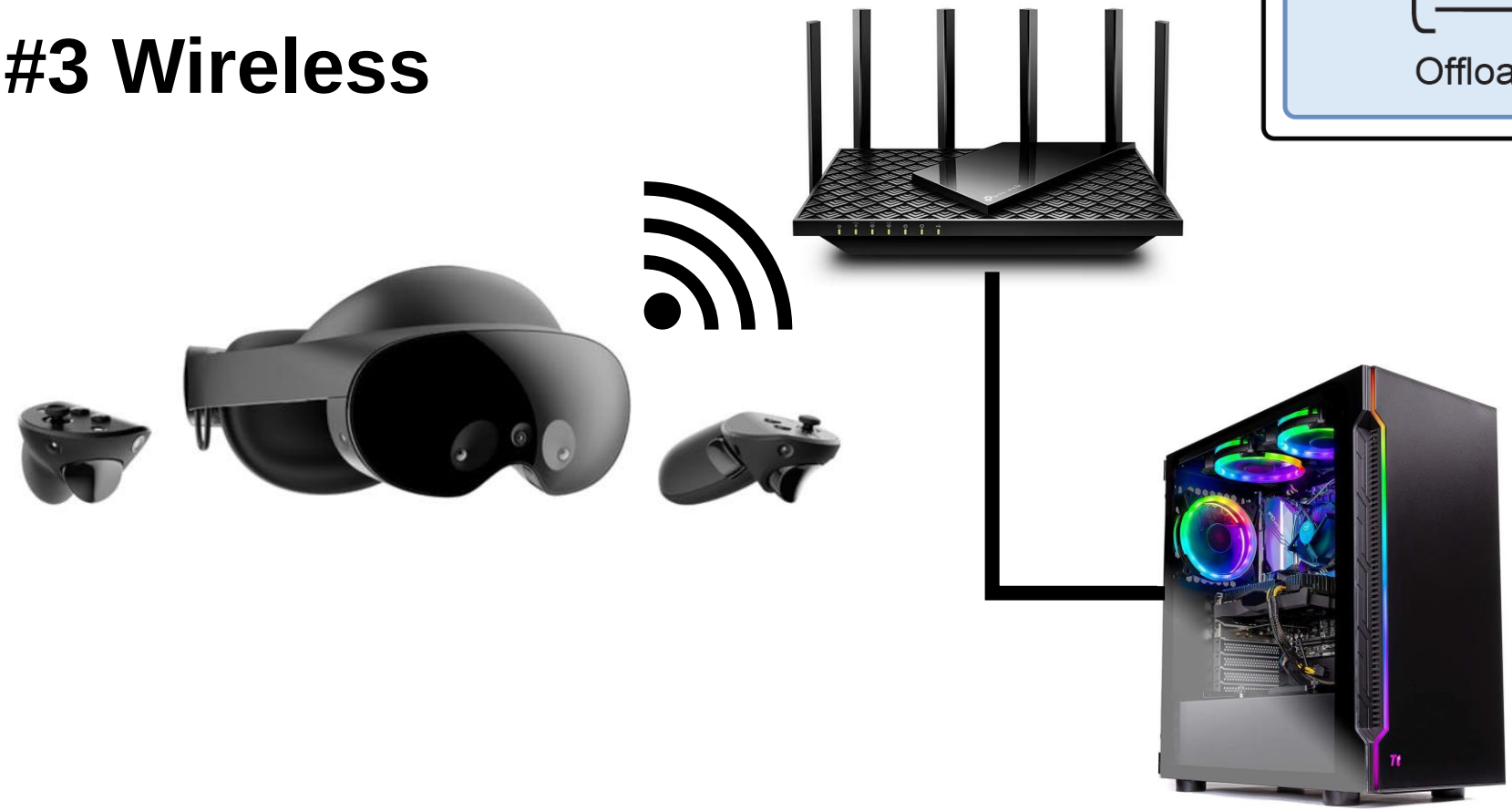
#2 Wired

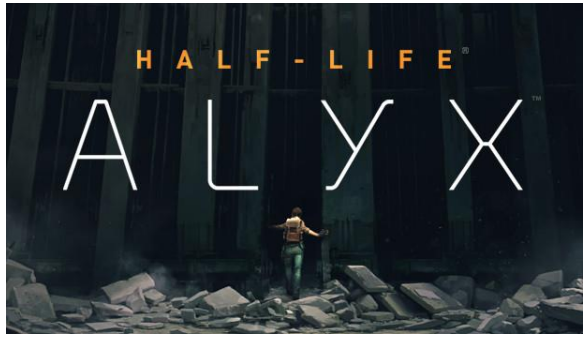


Experiment Setup



#3 Wireless





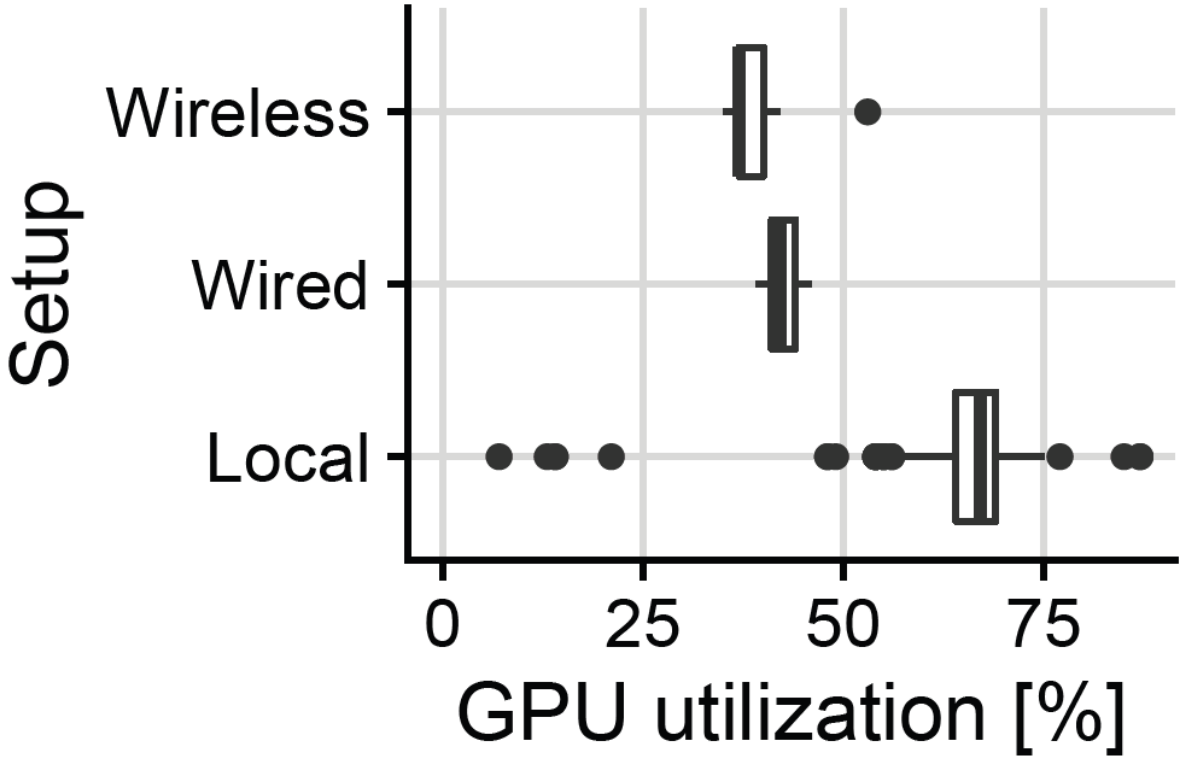
Experiment Design Goals

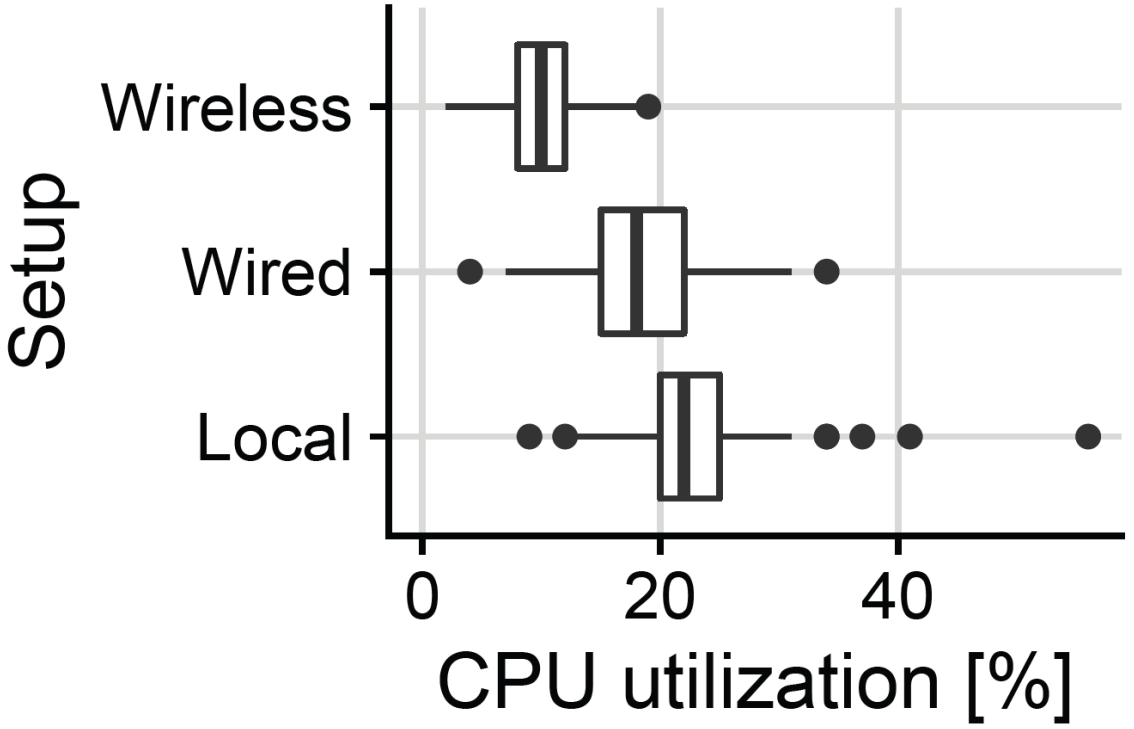
Q1 What is the **performance and resource usage** of VR applications on **state-of-the-art VR hardware**?

Q2 What are the **advantages and disadvantages of VR workload offloading** compared to native processing on VR headsets?

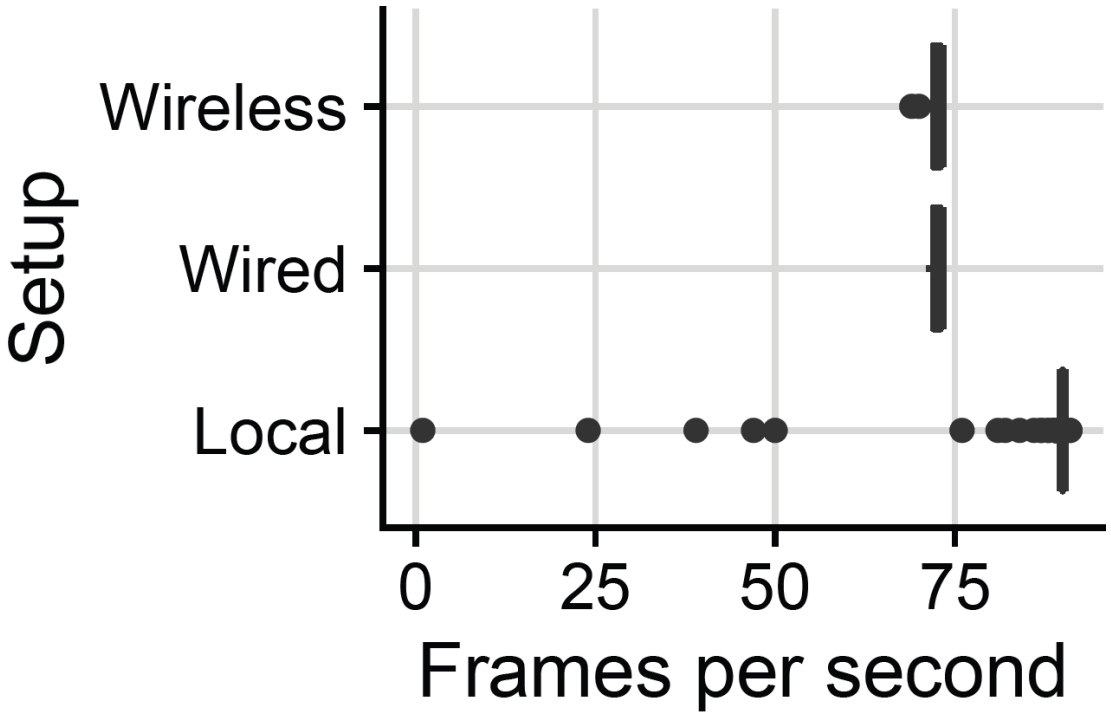
Q3 What are the **network requirements** to enable wireless compute offloading for VR?

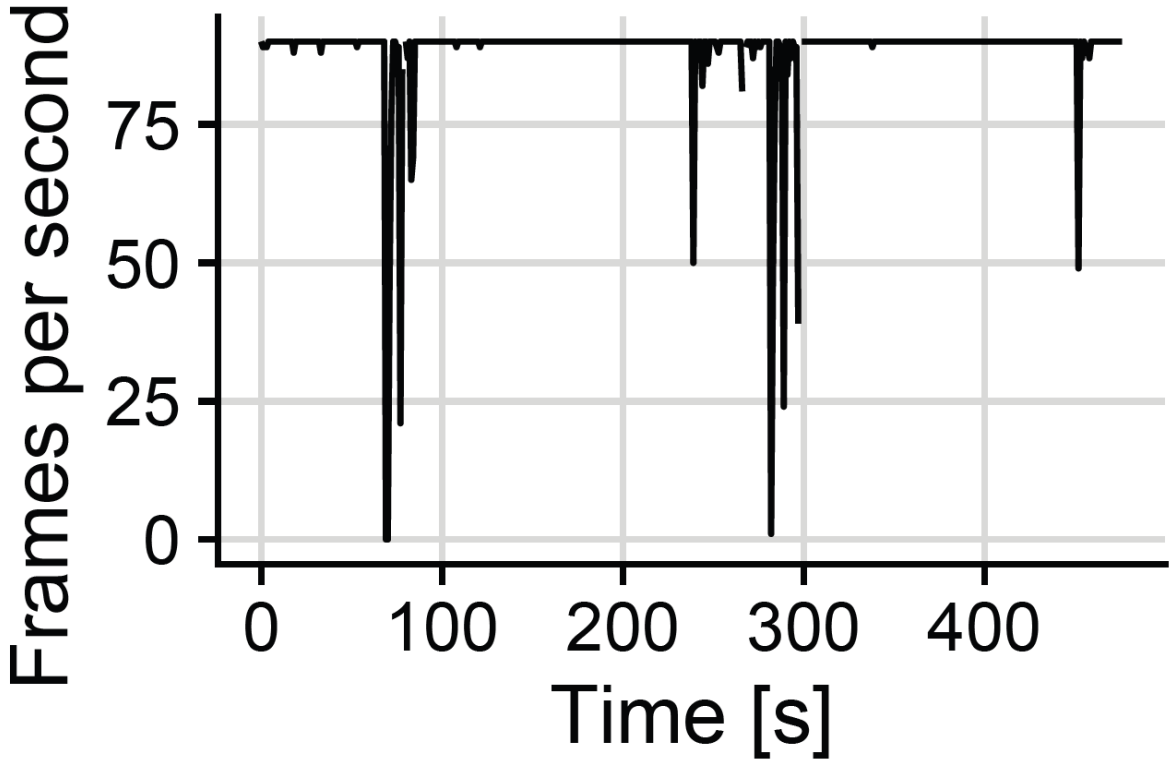
Resource Usage for All Tested Setups





Good Performance for All Tested Setups





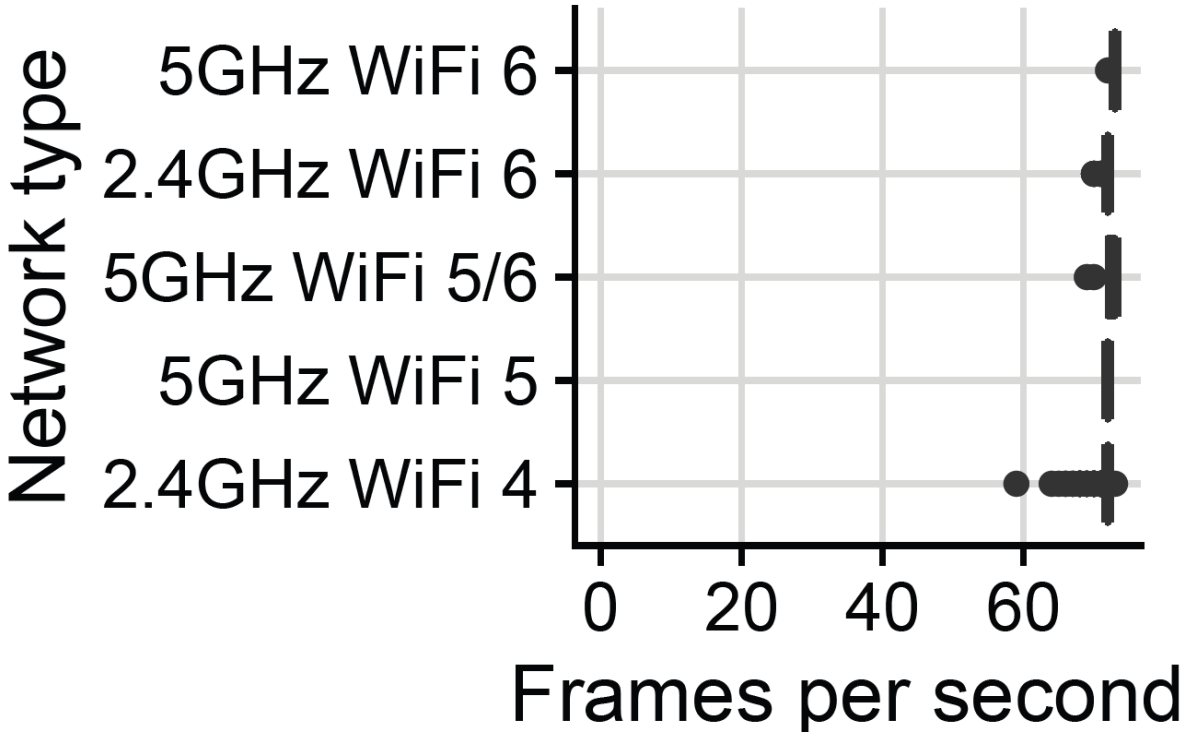
Experiment Design Goals

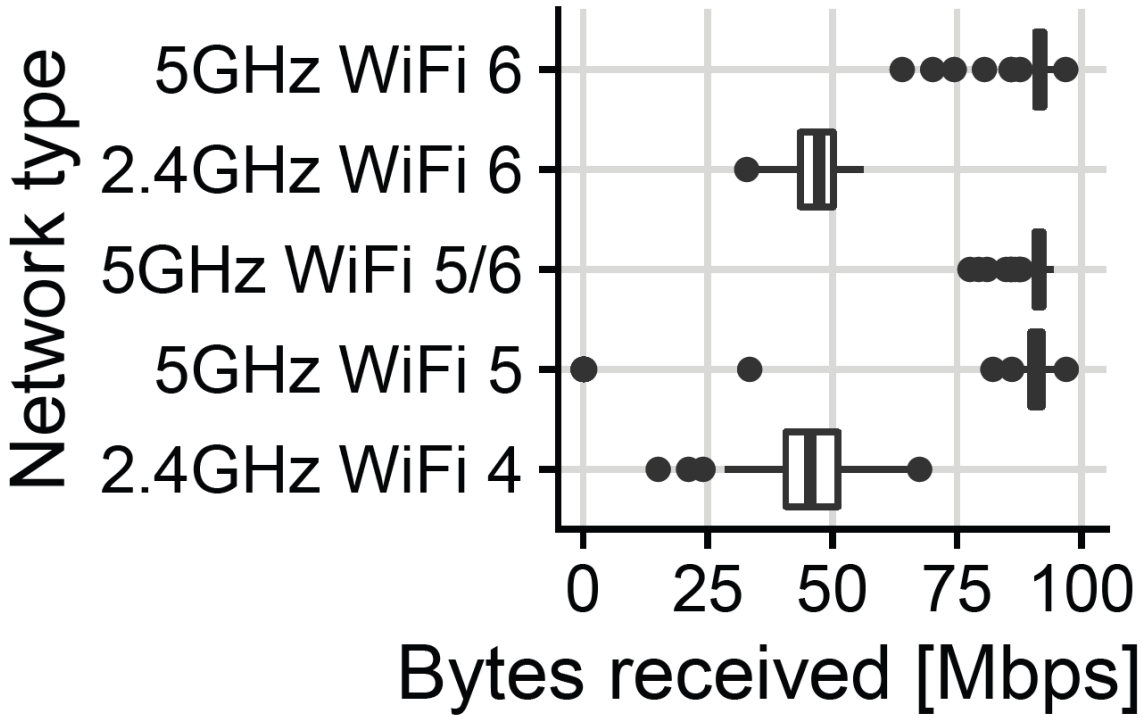
Q1 What is the **performance and resource usage** of VR applications on **state-of-the-art VR hardware**?

Q2 What are the **advantages and disadvantages of VR workload offloading** compared to native processing on VR headsets?

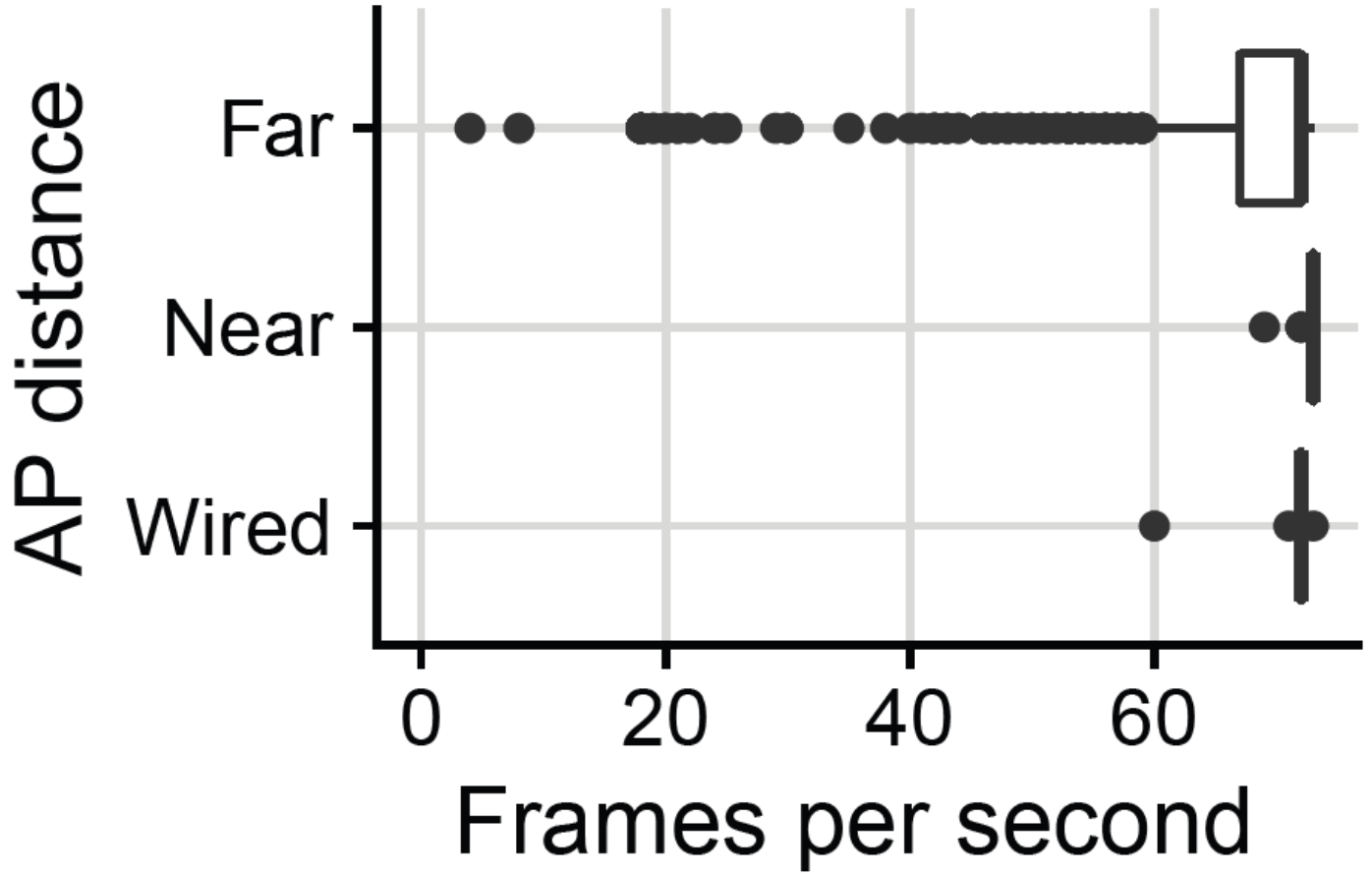
Q3 What are the **network requirements** to enable wireless compute offloading for VR?

Older WiFi Types Support VR Streaming

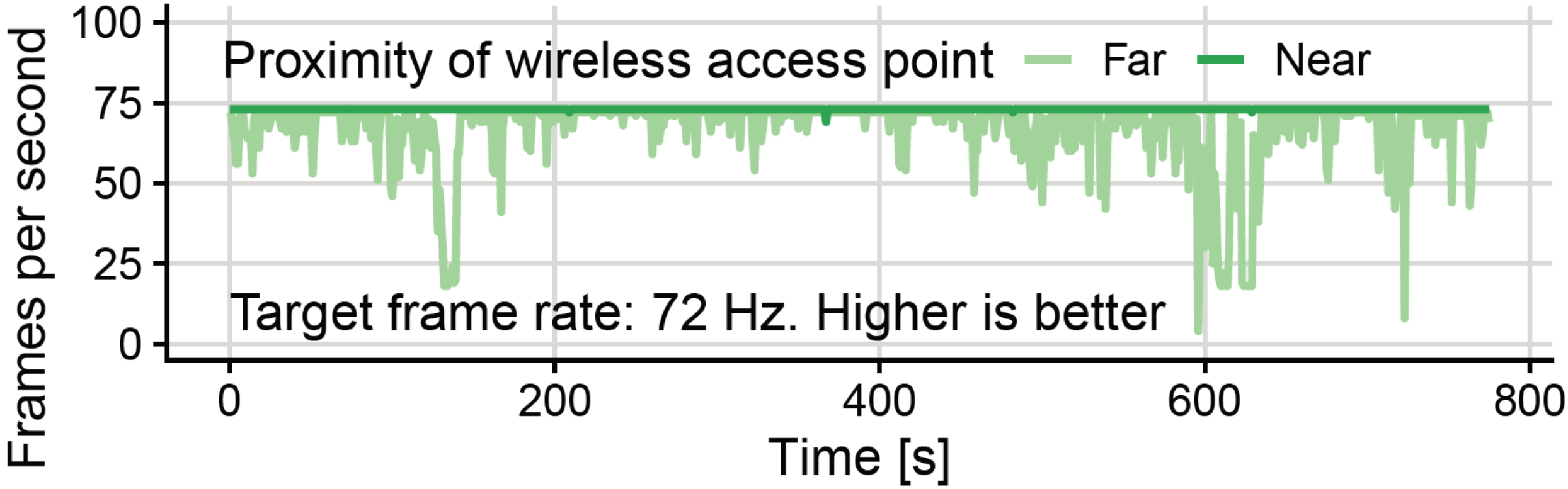




Performance Deteriorates Quickly When WiFi Signals Are Obstructed



Performance Deteriorates Quickly When WiFi Signals Are Obstructed



Towards a Workload Trace Archive for Metaverse Applications

Radu Apșan, Damla Ural, Paul Daniëlse, Vlad-Andrei Cursaru, Eames Trinh, **Jesse Donkervliet**, Alexandru Iosup

jesse.donkervliet@vu.nl
@jdonkervliet
https://www.jdonkervliet.com

How to Deploy Metaverse Infrastructure?

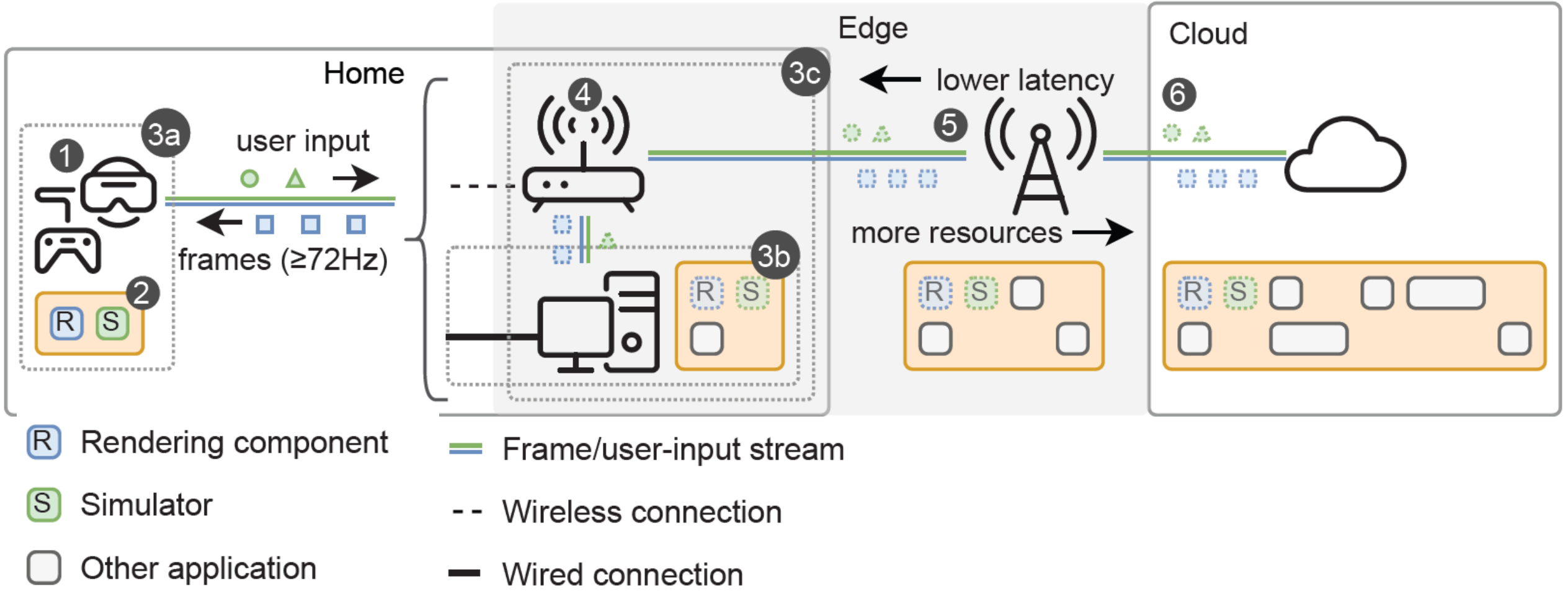


Figure source: *Can My WiFi Handle the Metaverse? A Performance Evaluation Of Meta's Flagship Virtual Reality Hardware*, Jesse Donkervliet, Matthijs Jansen, Animesh Trivedi, Alexandru Iosup (2023), ICPE HotCloudPerf 2023

How to Deploy Metaverse Infrastructure?

How to answer this question?

This talk

1. Performing real-world experiments with VR devices is labor intensive, devices are scarce and expensive
2. No publicly available datasets to explore
3. No simulators for the metaverse

Our Approach

1. We design a tracing system to simplify and partially automate performing real-world experiments with VR devices
2. Through real-world experiments, we create an initial **dataset for metaverse systems**
3. Future work: use datasets to create a simulator to explore metaverse system behavior for a fraction of the cost (time, money)

User Input Tracing with Record-n-Replay

1. Poll for input

App

1.5 RnR tracer

Input trace archive

≥72 Hz
(<14 ms per iteration)

2. Simulate and render frame

3. Send frame to display

Poll performance counters (1 Hz)

Perf. monitor

Node

Experiment Setup

1. Poll for input

App

1.5 RnR tracer

Input trace archive

≥72 Hz
(<14 ms per iteration)

2. Simulate and render frame

3. Send frame to display

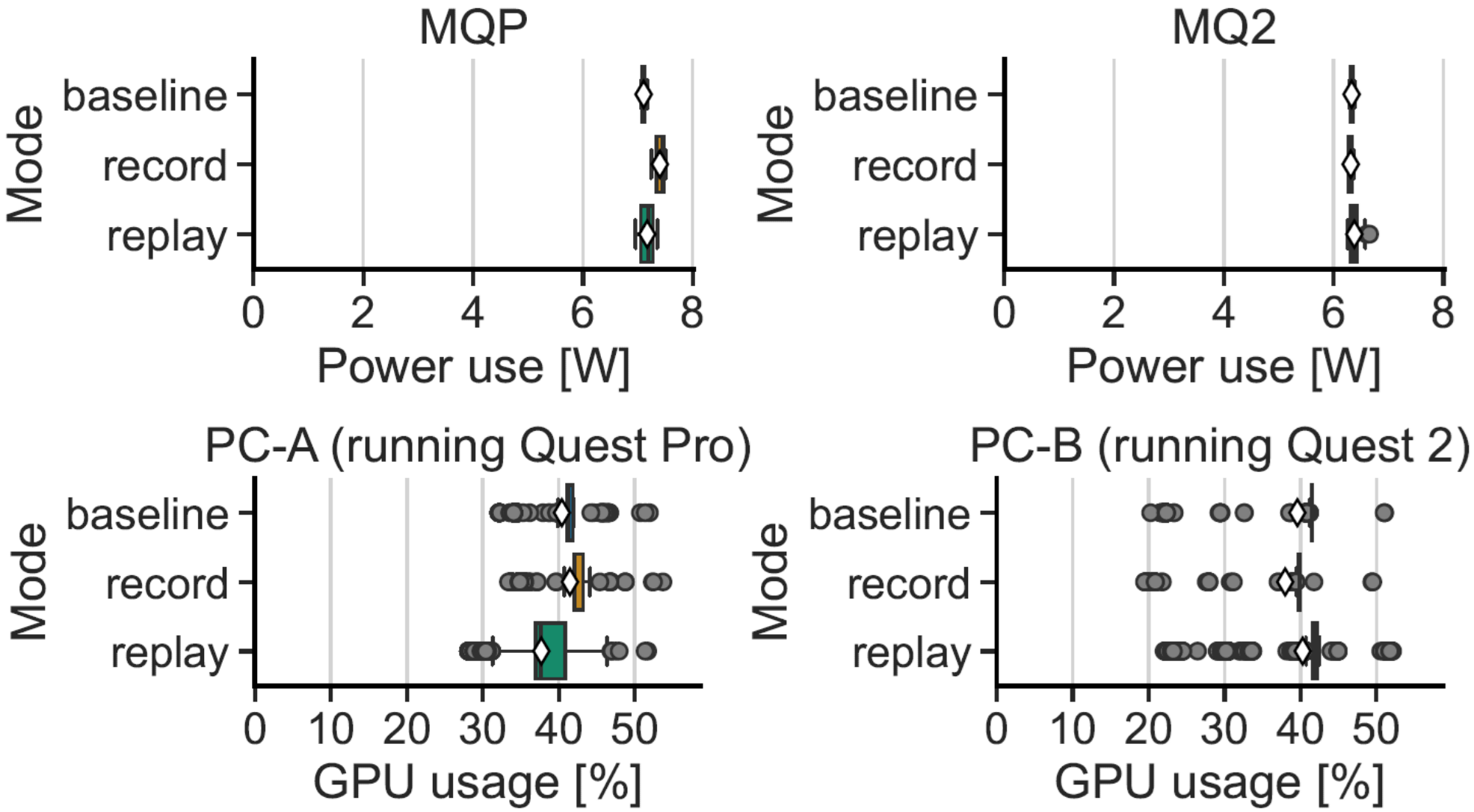
Poll performance counters (1 Hz)

Perf. monitor

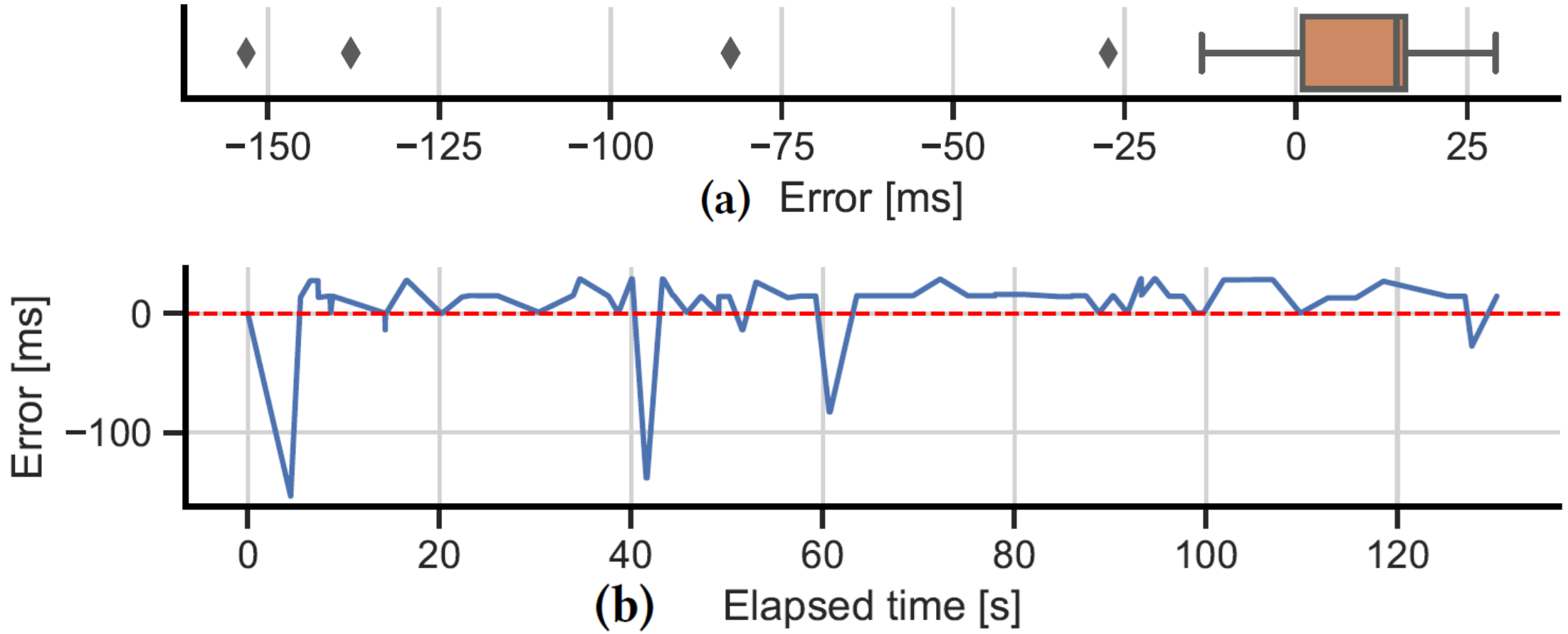
Network emulation

Node

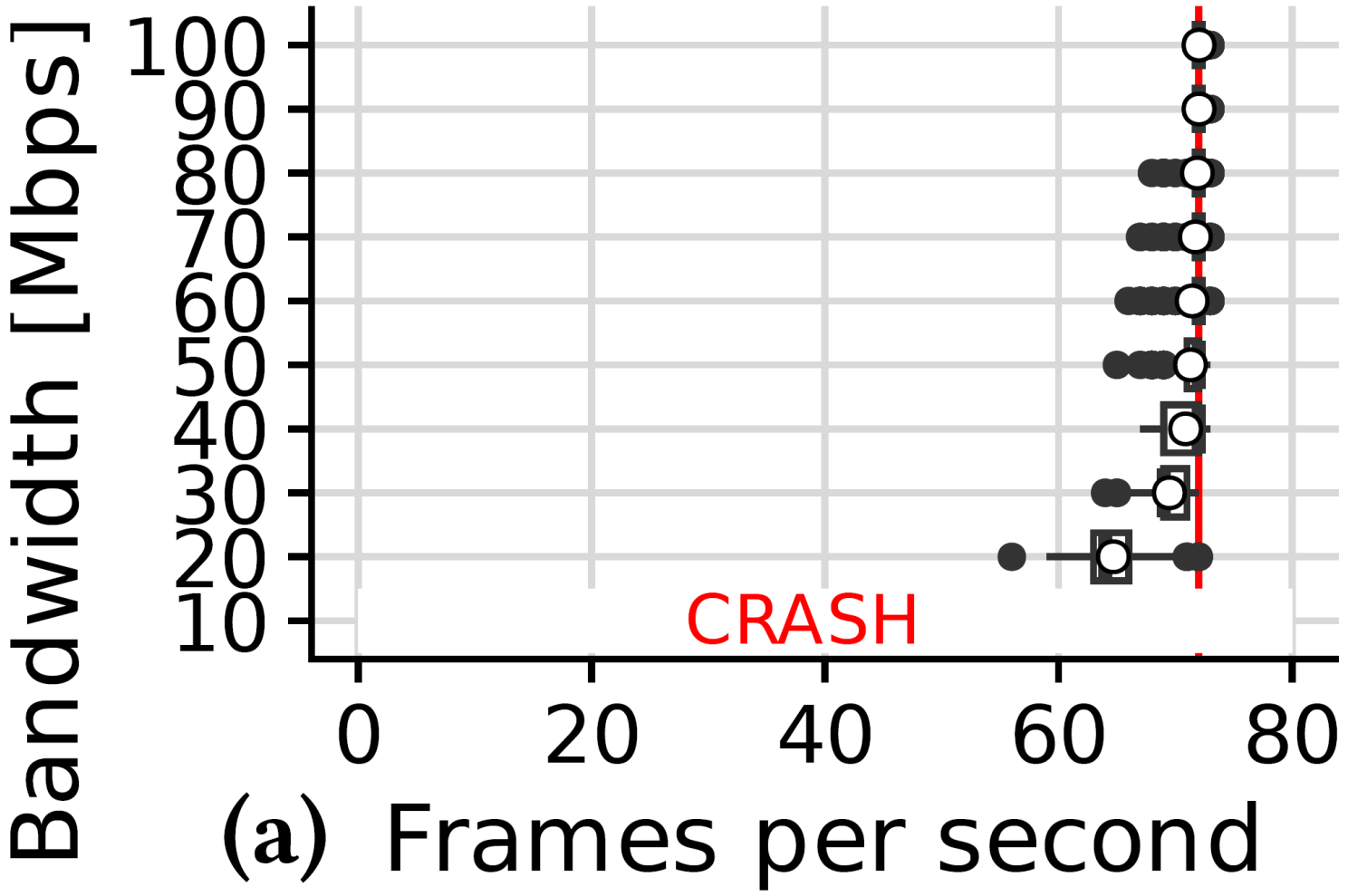
Record and Replay has low overhead



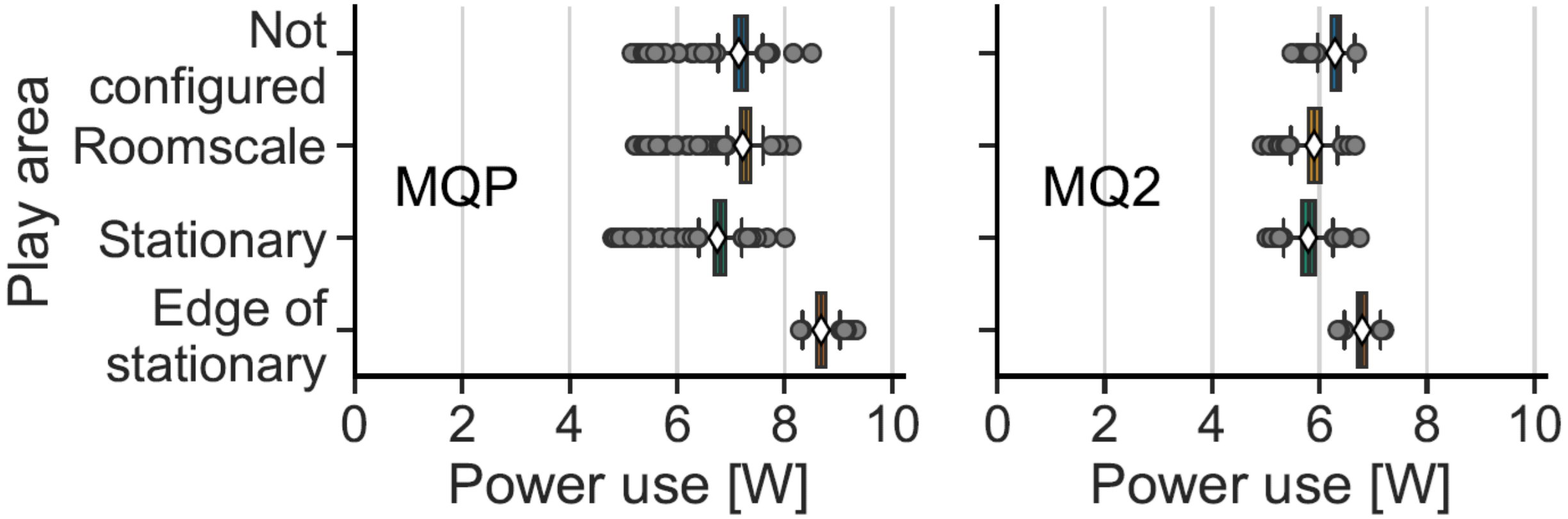
Record and Replay Input with High Timing Accuracy



VR streaming playable with (relatively) low bandwidth



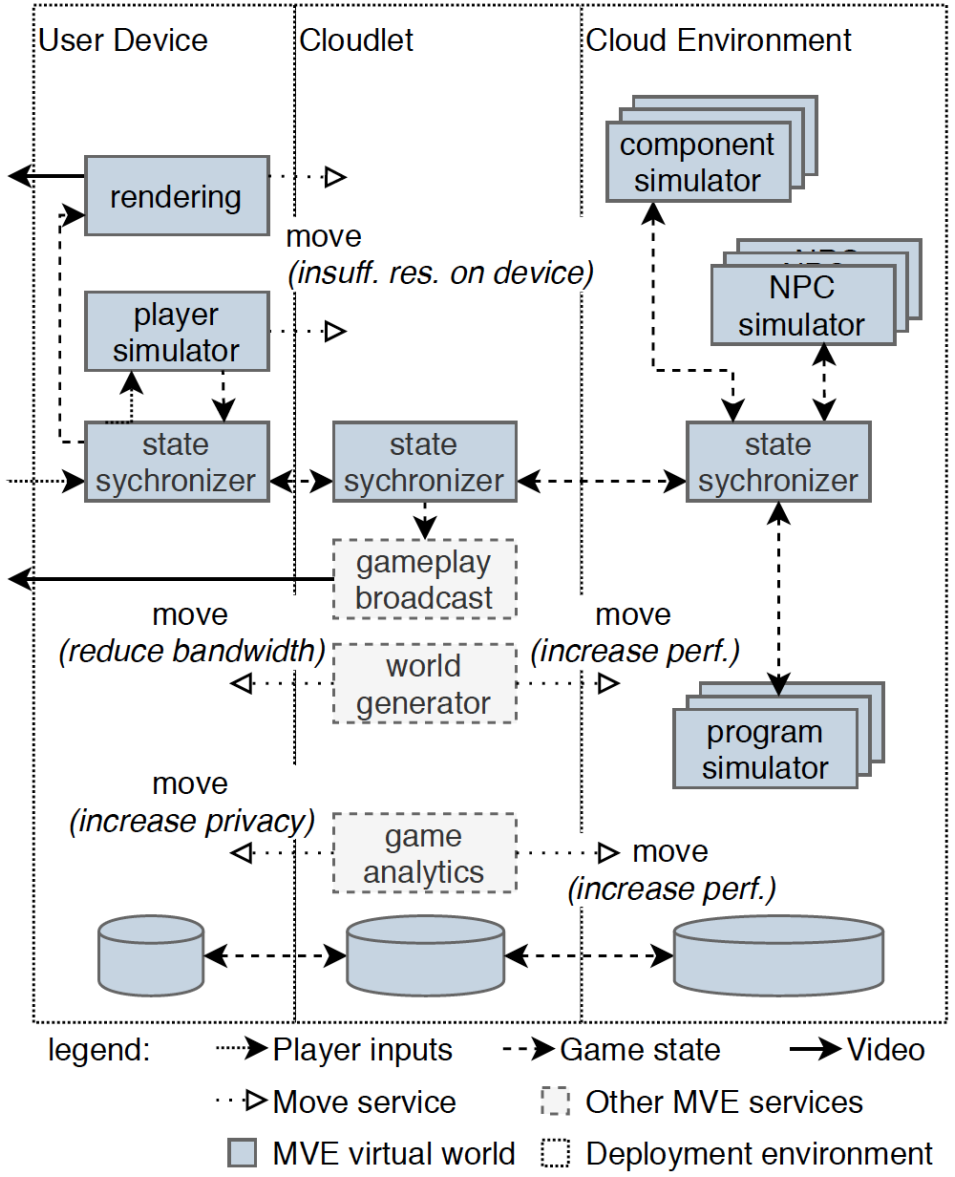
Blending Reality Increases Power Use



Blending Reality Increases Power Use

Ongoing Work

1. Vision for large-scale MVEs for digital societies
2. Workload trace archive and simulator for virtual reality
3. Large-scale modifiable virtual world as research platform



Jesse Donkervliet, Animesh Trivedi, Alexandru Iosup (2020) Towards Supporting Millions of Users in Modifiable Virtual Environments by Redesigning Minecraft-Like Games as Serverless Systems 12th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud'20)

(part of)

The Opencraft Research Team

Jesse Donkervliet
Tech Lead

Alexandru Iosup
Project Lead

Vlad Cursaru
MSc student

Elena Stroiu
BSc student

Evelina Niţoiu
Honours student

Gleb Mishchenko
Honours student

Ean-Dan Tjon-Joek-Tjien
MSc student

Javier Ron
MSc student

Jerom van der Sar
Honours student

Jerrit Eickhoff
MSc student

Tiberiu Iancu
Honours student

Jim Cuijpers
MSc student

Misha Rigot
MSc student

Victor Gavrilovici
BSc student

Damla Ural
MSc student

Eames Trinh
MSc student

Paul Daniëlse
MSc student

Radu Apşan
MSc student

Guivari Dzar Amri
BSc student

Chris Esterhuyse
MSc student

Ernst van der Hoeven
External

Further Reading

Can My WiFi Handle the Metaverse? A Performance Evaluation Of Meta's Flagship Virtual Reality Hardware, Jesse Donkervliet, Matthijs Jansen, Animesh Trivedi, Alexandru Iosup, ICPE HotCloudPerf 2023

Meterstick: Benchmarking Performance Variability in Cloud and Self-hosted Minecraft-like Games, Jerrit Eickhoff, Jesse Donkervliet, Alexandru Iosup, ICPE 2023

Servo: Increasing the Scalability of Modifiable Virtual Environments Using Serverless Computing, Jesse Donkervliet, Javier Ron, Junyan Li, Tiberiu Iancu, Cristina L. Abad, Alexandru Iosup, ICDCS 2023

Dyconits: Scaling Minecraft-like Services through Dynamically Managed Inconsistency, Jesse Donkervliet, Jim Cuijpers, Alexandru Iosup, ICDCS 2021

Towards Supporting Millions of Users in Modifiable Virtual Environments by Redesigning Minecraft-Like Games as Serverless Systems, Jesse Donkervliet, Animesh Trivedi, Alexandru Iosup, HotCloud 2020