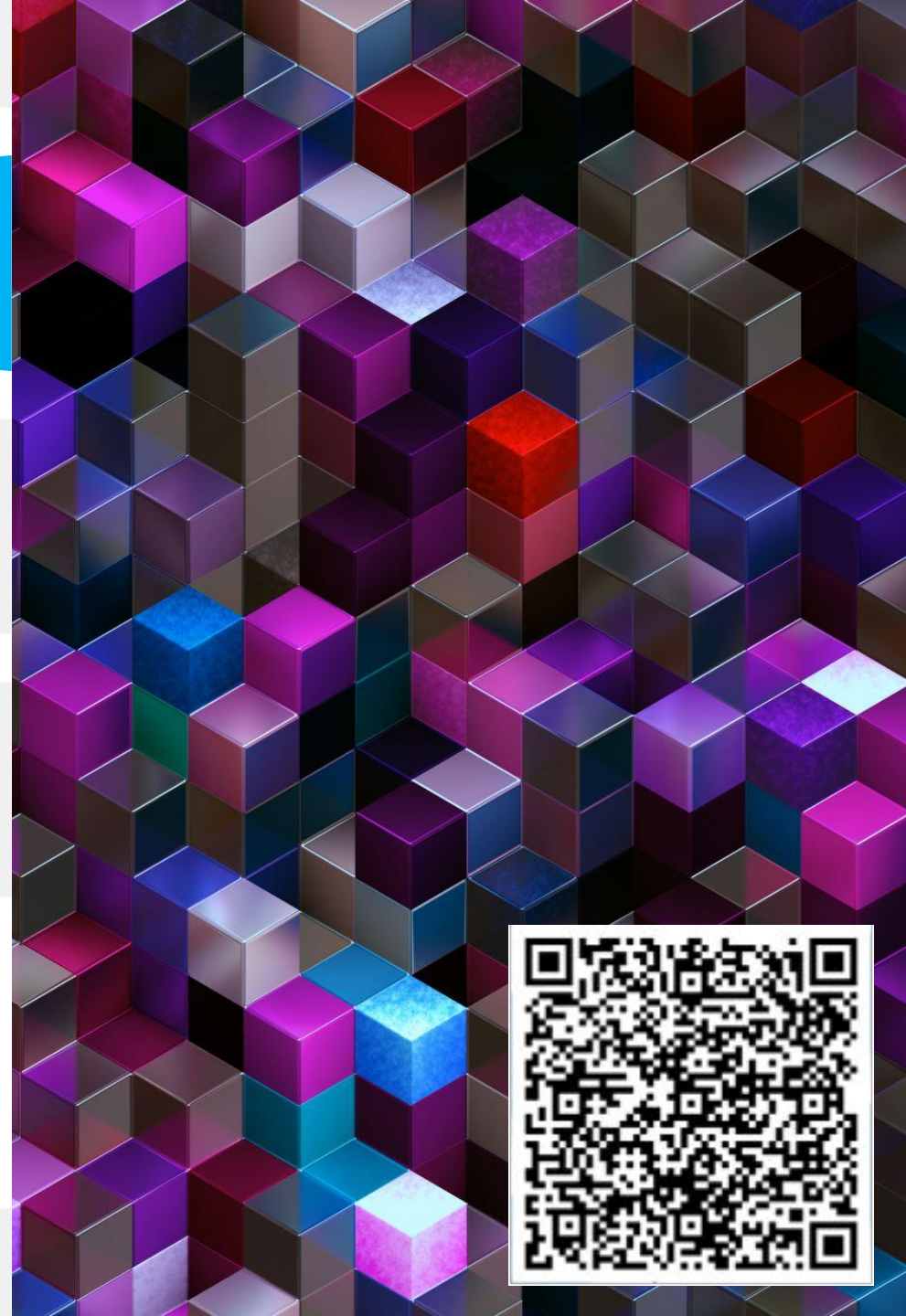
Designing a Protocol-Agnostic Benchmark for MVEs

By: Alexandr Costei

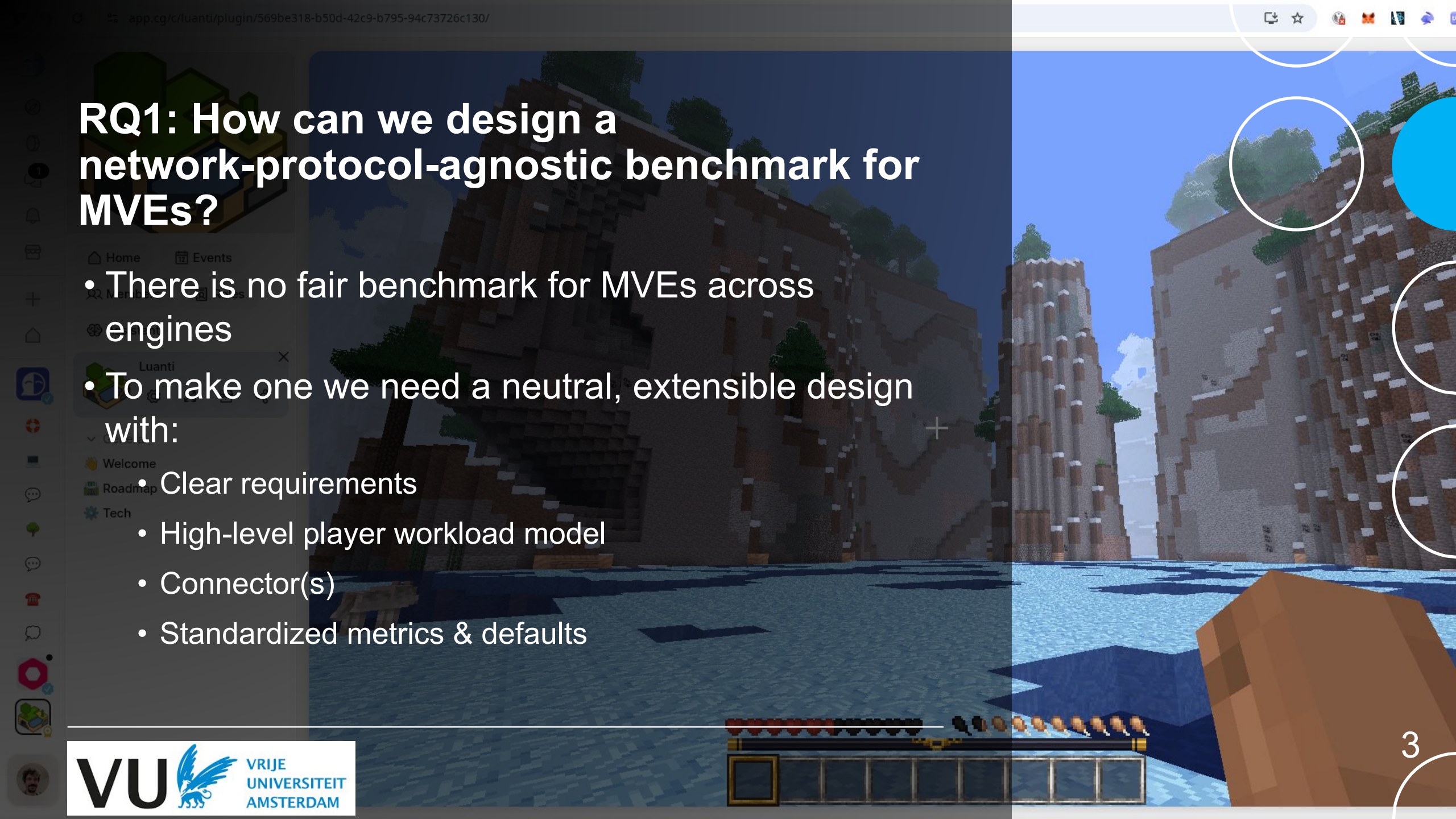
Supervisor: Jesse Donkervliet



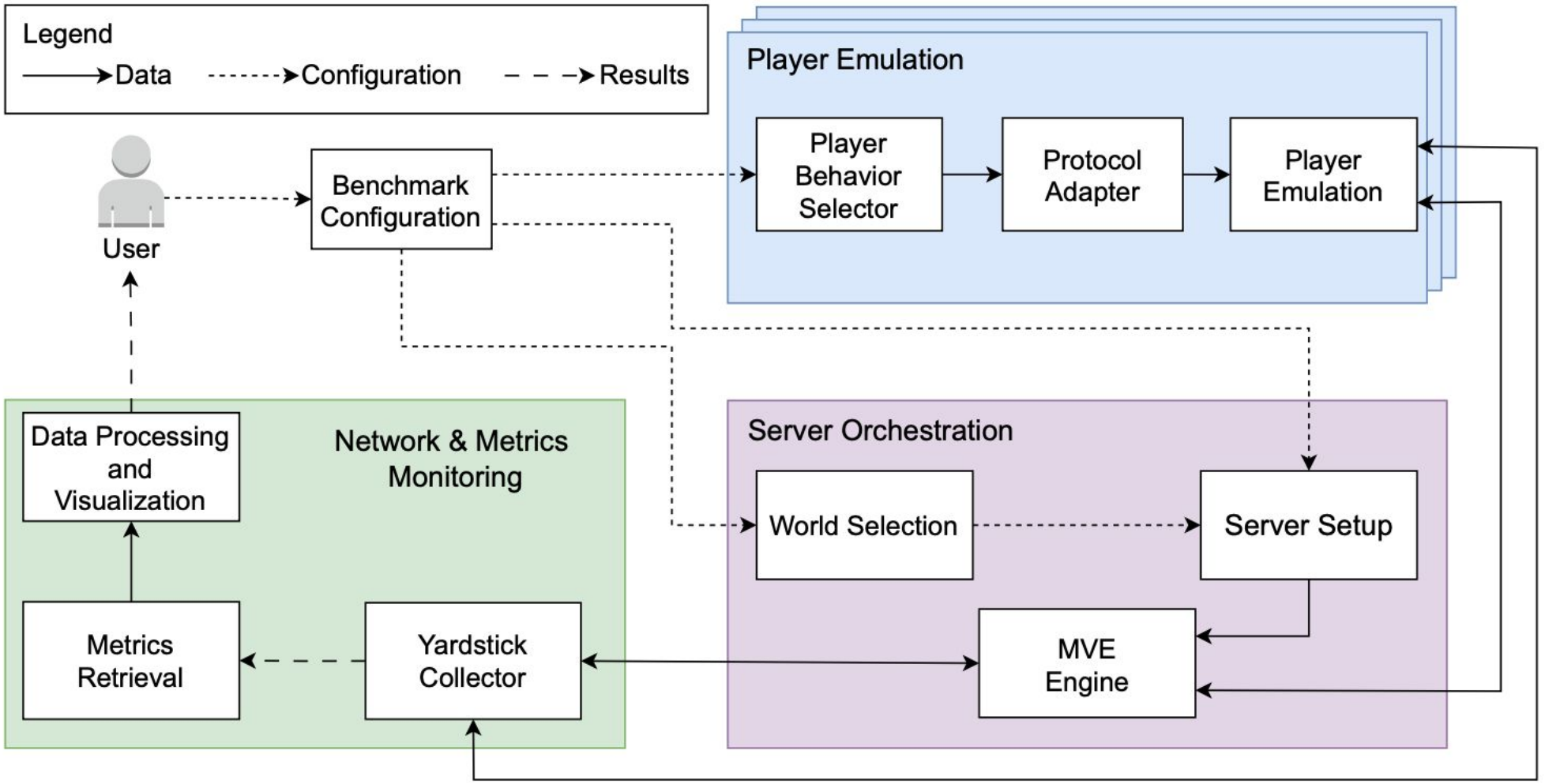
VU VRIJE UNIVERSITEIT AMSTERDAM

RQ1: How can we design a network-protocol-agnostic benchmark for MVEs?

- There is no fair benchmark for MVEs across engines
- To make one we need a neutral, extensible design with:
  - Clear requirements
  - High-level player workload model
  - Connector(s)
  - Standardized metrics & defaults



VU VRIJE UNIVERSITEIT AMSTERDAM



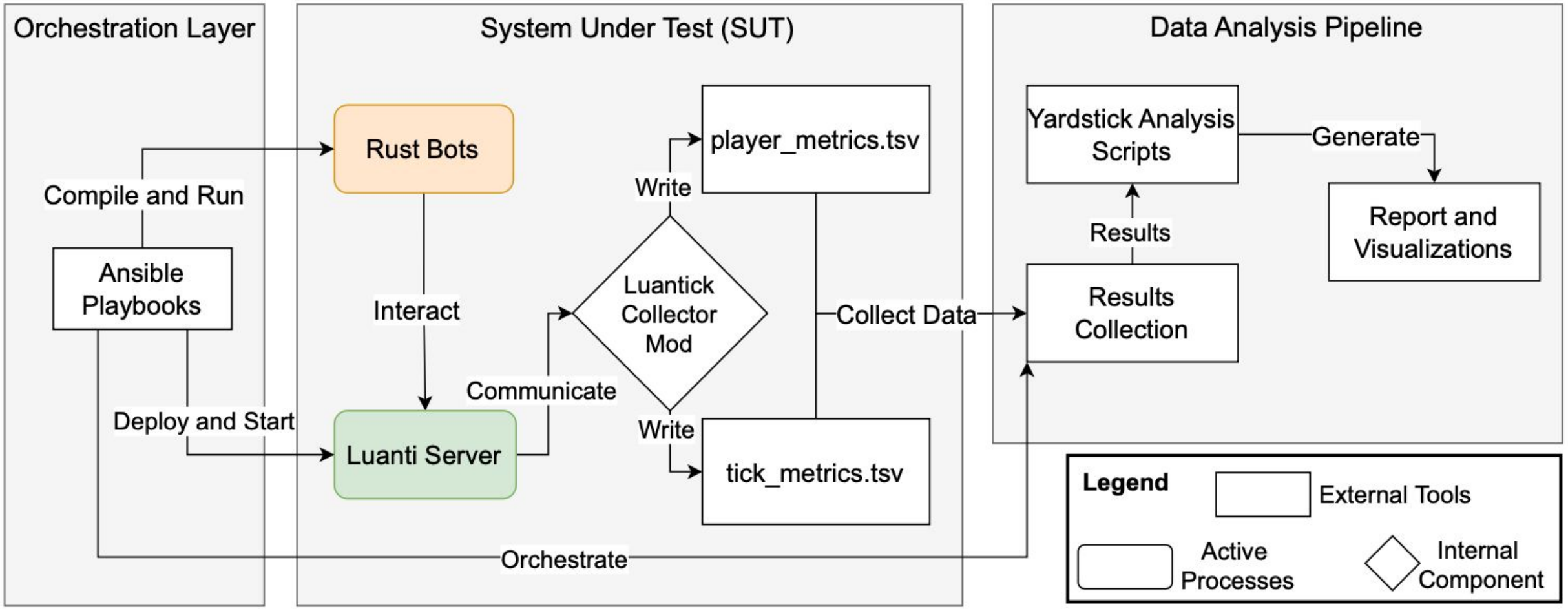
Designing a network-protocol-agnostic benchmark for MVEs

RQ2: How can we implement this benchmark in practice?

HAVING AN IMPACTFUL AND EFFECTIVE DESIGN REQUIRES WORKING TOOLCHAIN

HOW? EXTEND THE YARDSTICK “LUANTICK” AND ADD PER-GAME TRANSLATION LIBRARIES FOR INPUTS

THE OUTCOME SHOULD BE A TRUSTED DEPLOYABLE BENCHMARK



Implementation

RQ3: How can we evaluate MVEs that rely on different network protocols?

- Fair and repeatable evidence
- Establish a procedure for collecting, analyzing, and comparing results across different platforms

VU VRIJE UNIVERSITEIT AMSTERDAM

Experimental Setup

Enviornment

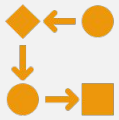
DAS-5 cluster
Isolated test nodes
Reproducible configuration

Variables

Bot count scaling (10 □ 200+)
Workload types (movement, building, mixed)
Duration per run (3–5 min)

Measurements

Tick duration / TPS stability
System-level metrics (CPU, memory, network)



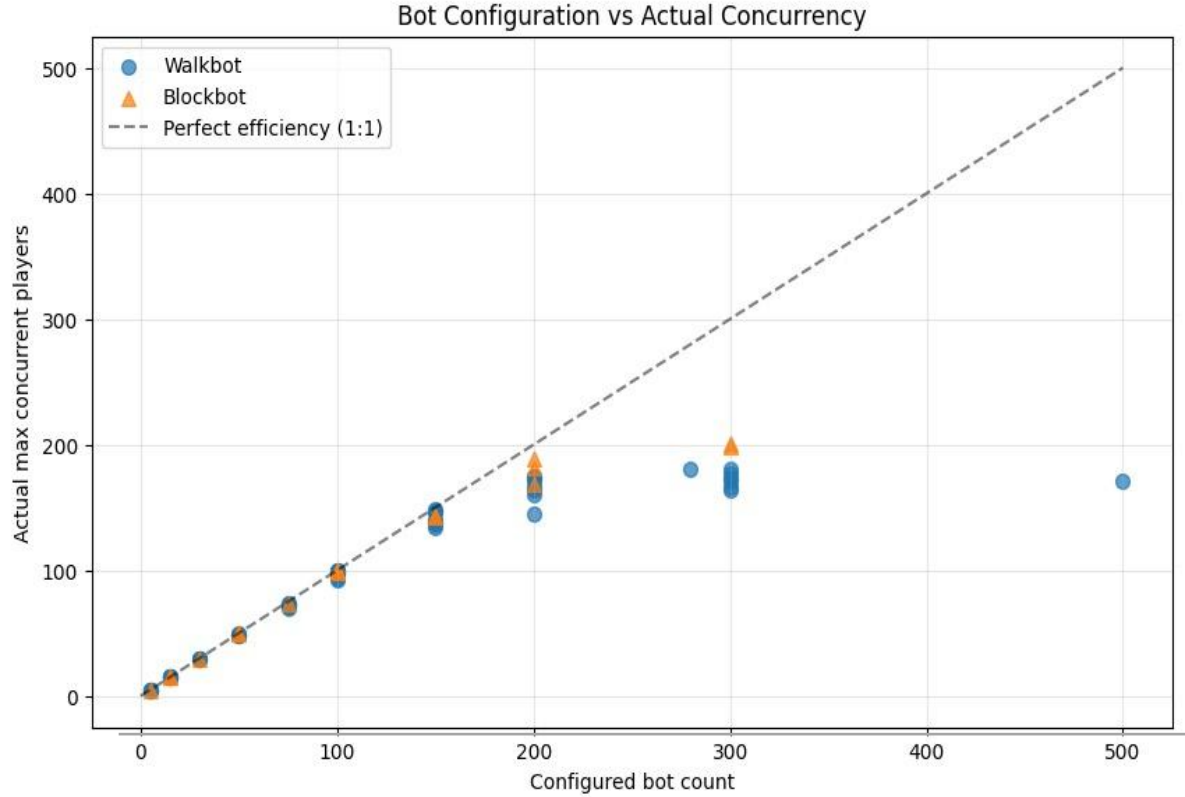
Procedure

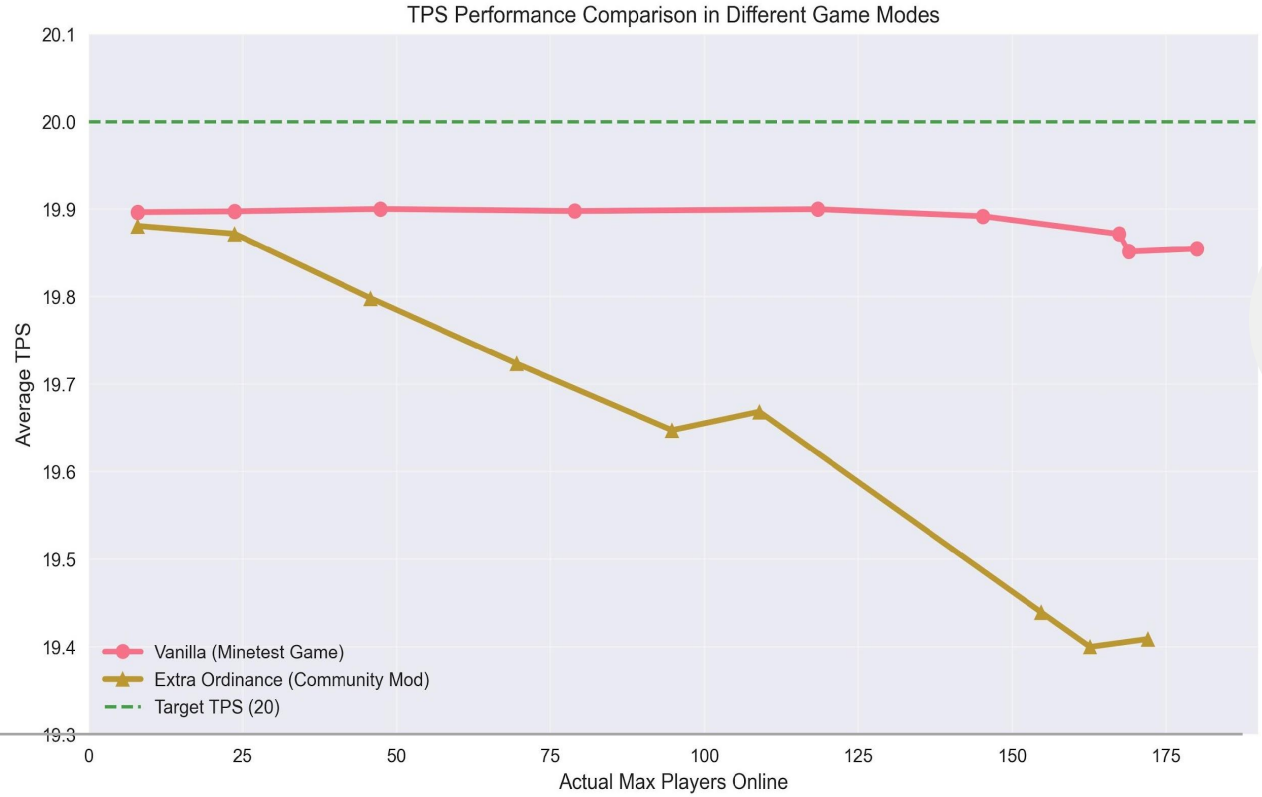
Incremental scaling until overload
Repeated runs for statistical confidence

Results

Main Findings 1 & 2

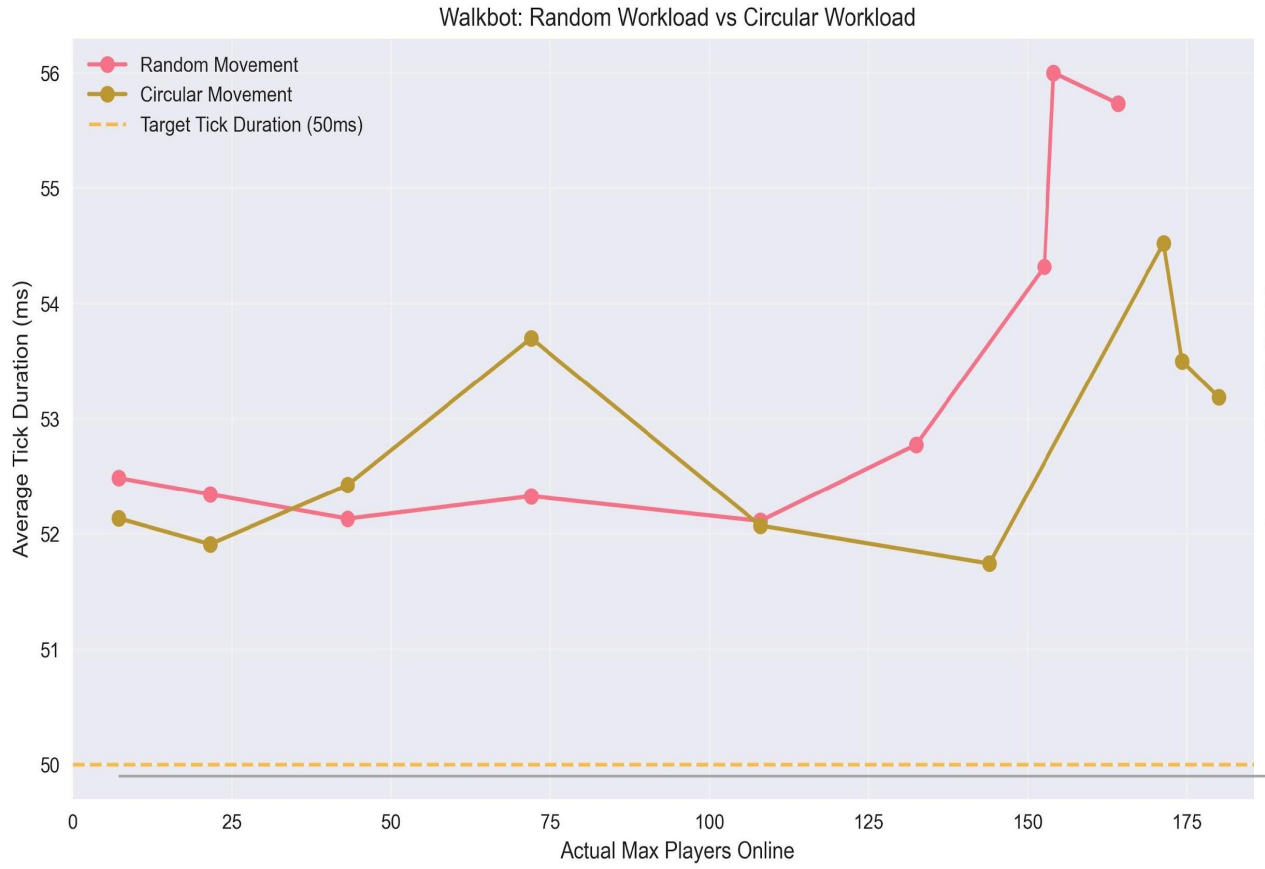
- Luanti demonstrates protocol-specific scaling with UDP, leading to efficient packet handling at low player counts
- But scalability ceiling reached earlier than expected (~180–190 bots)

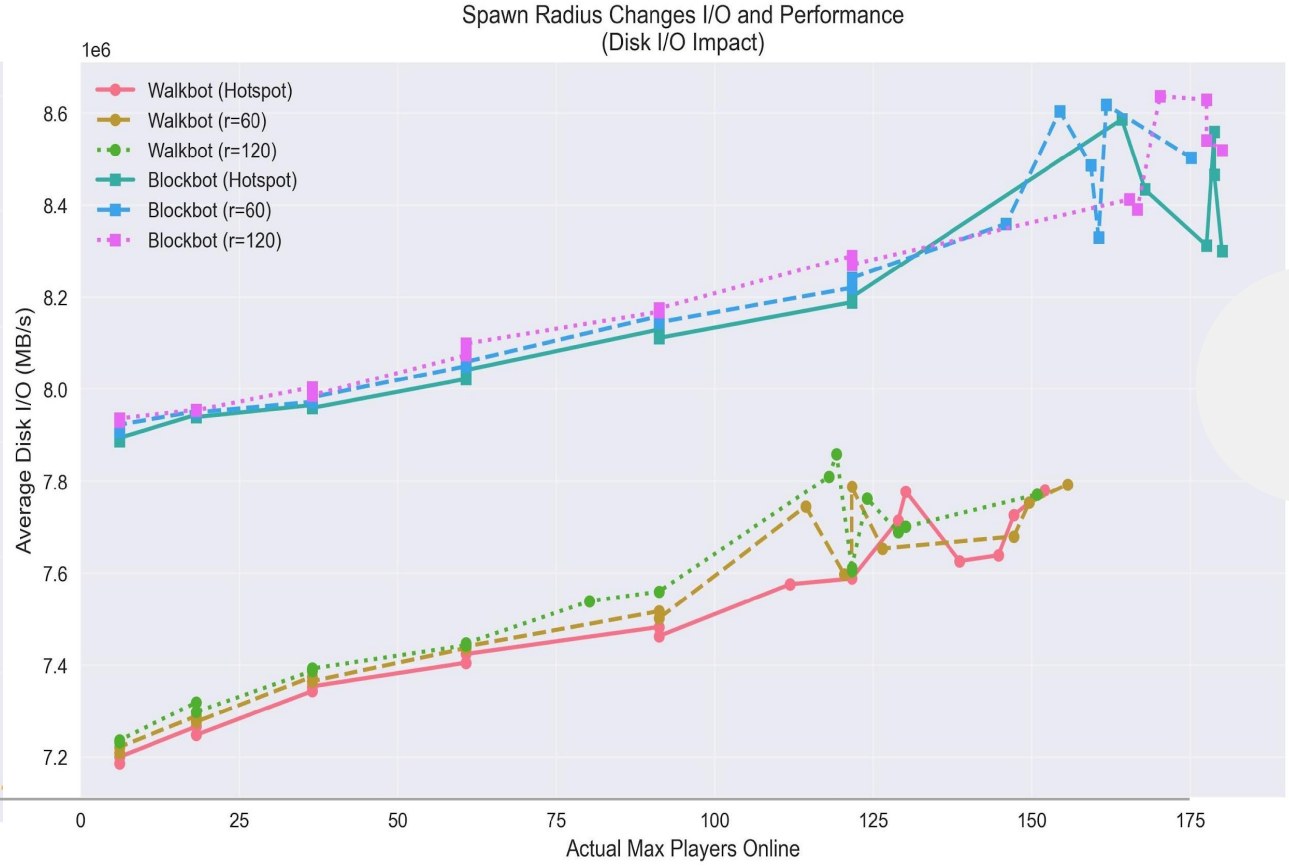


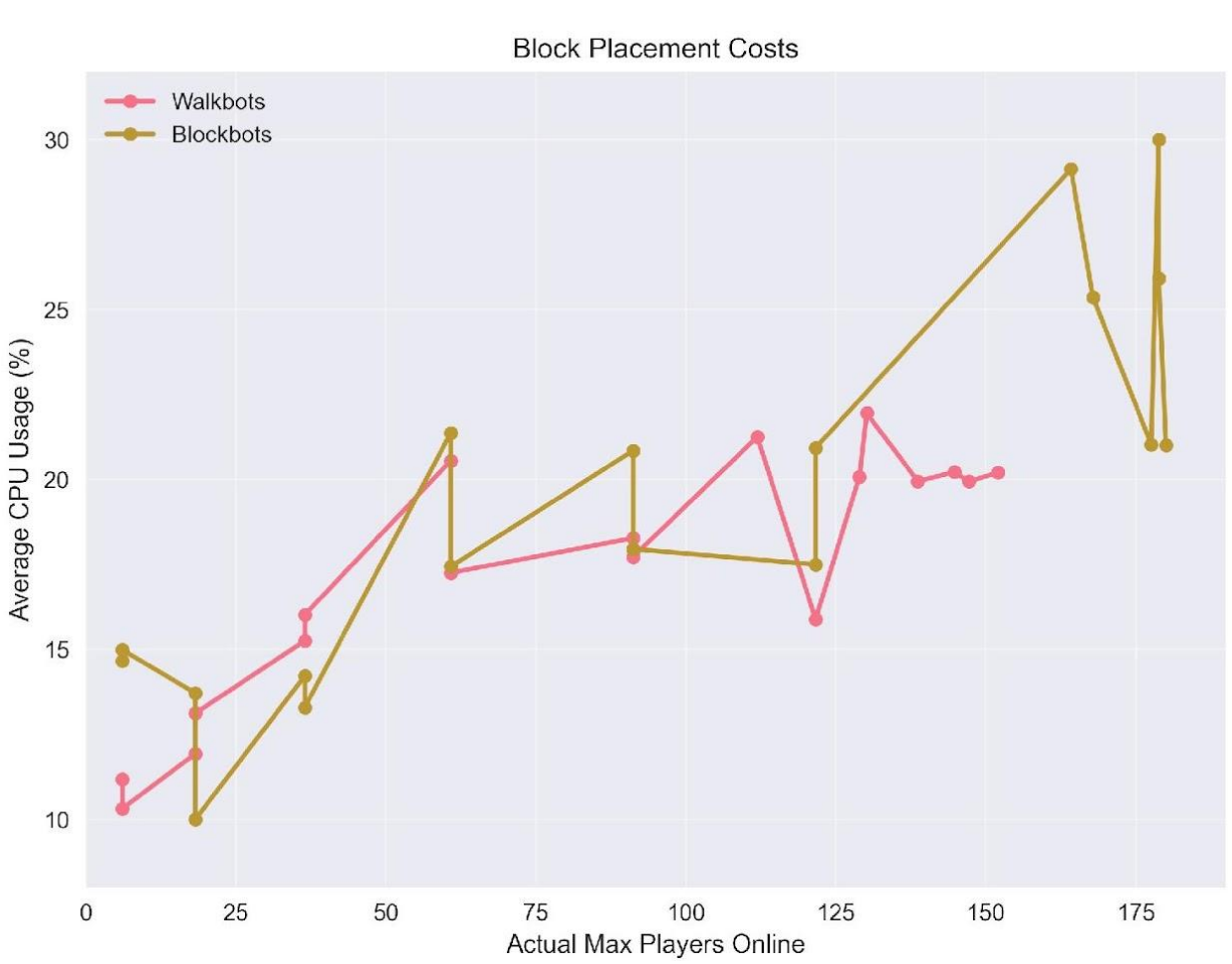


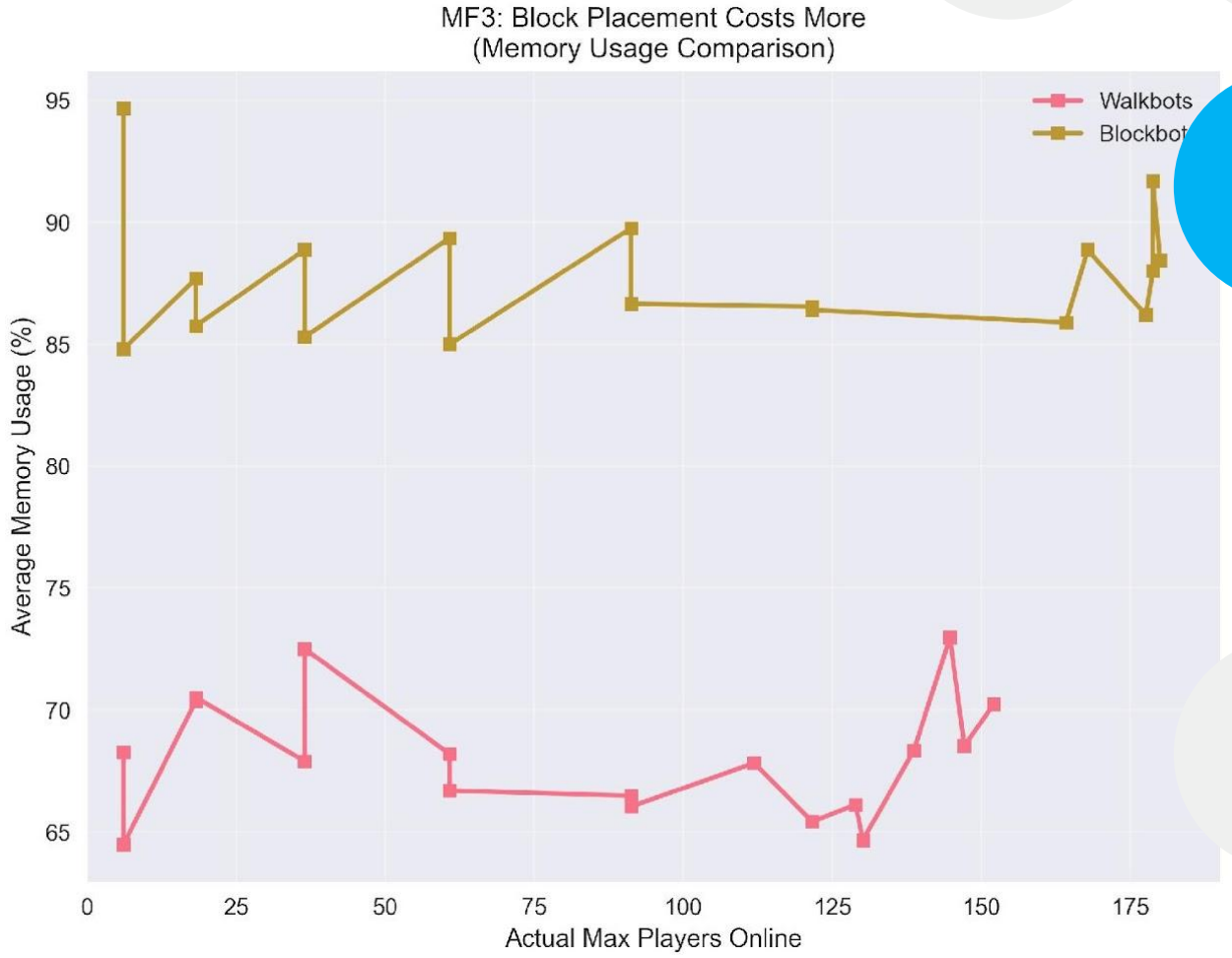
Main Findings 3 & 4

- Tick duration variability grows under mixed workloads
- CPU and memory bottlenecks appear before network saturation



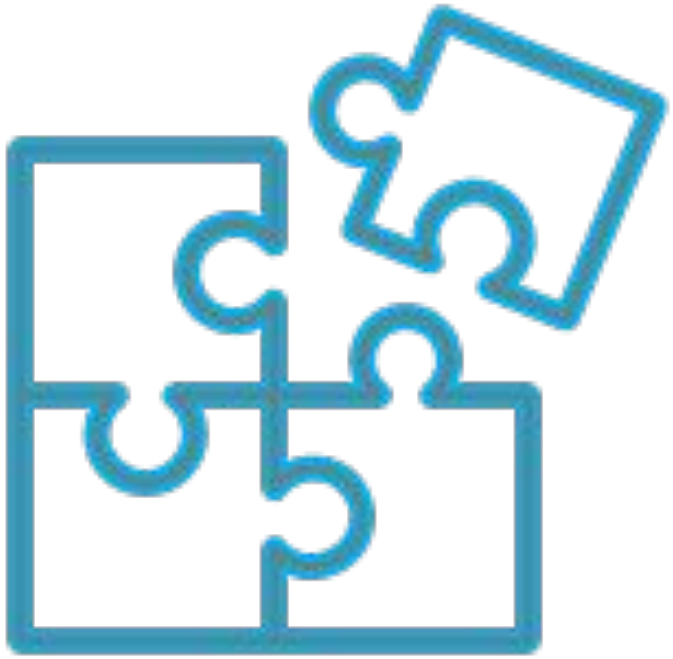






Main Finding 5

- Reproducibility confirmed across runs, but system variability remains



A protocol-agnostic benchmark makes fair comparison of MVEs possible, creating a foundation for reproducible research and future engine innovation