

Pikko Server

Scalability when using Erlang on the server side
for massive multiplayer game servers.

Agenda

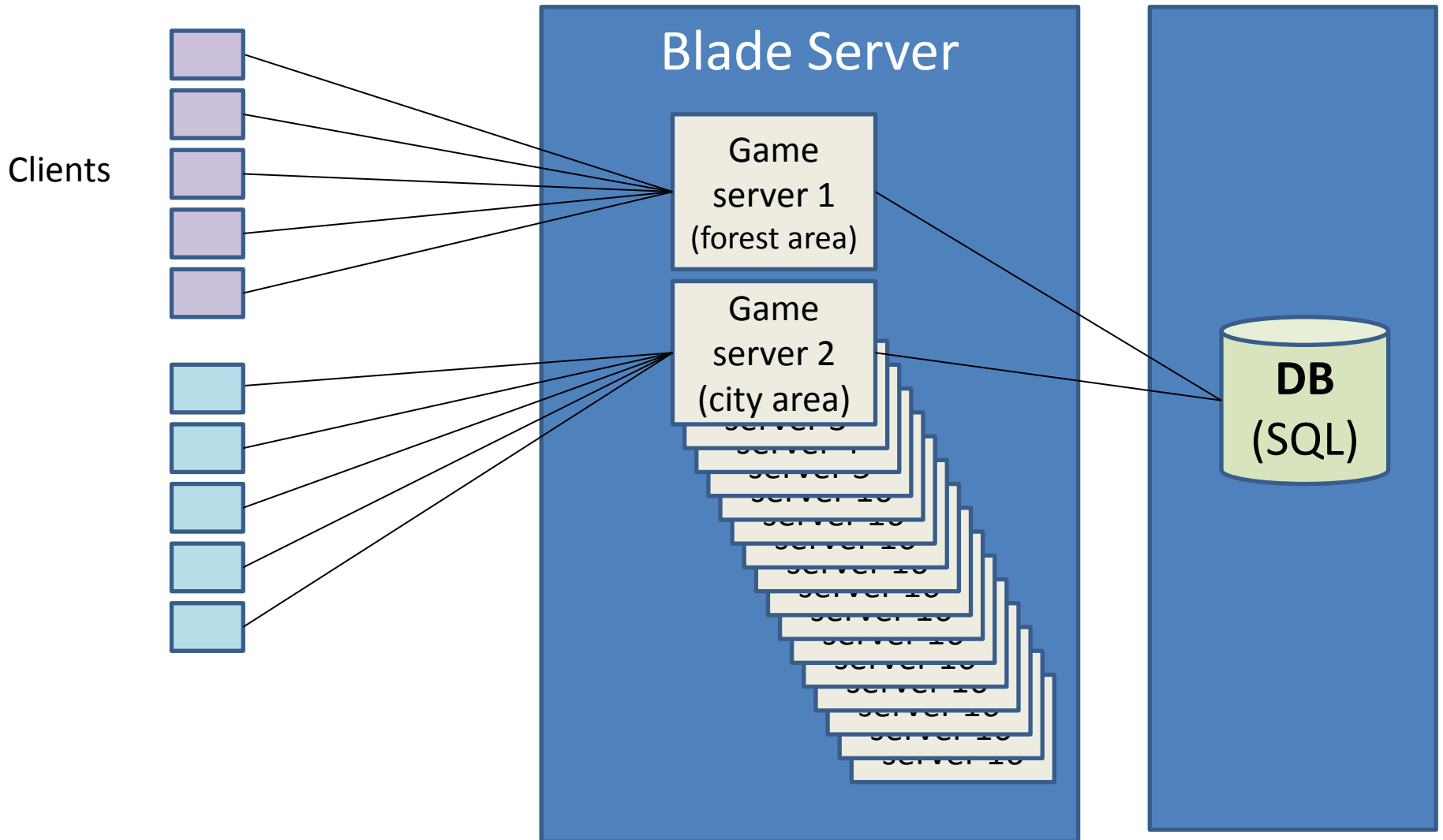
1. David & PikkoTekk
2. Normal game servers
3. The scalability problem
4. A more scalable solution – Pikko Server
5. World record attempt for game servers
6. Questions & Answers

David Almroth & PikkoTekk

- Did 10 years of Enterprise Java (Banking & Finance)
- Now Erlang architect
- Started PikkoTekk 2009 together with Reep, a group of talented game programmers in Uppsala
- PikkoTekk is a company producing scalable software for multiplayer computer games

Normal Zoned MMO Game Servers

(For example: World of Warcraft, Eve Online)



MMO = Massive Multiplayer Online

Scalability Problems for Normal Game Server

- **Can not utilize the power of modern multicore processors**
- Game servers usually have one single thread = not scalable

The result is:

- **No big crowds** can be handled (not without challenges)
- **No big battlefields** can be handled
- **Database gets overloaded** is another problem
(not the focus today)

Normal Solutions to the Server Overload Problem

1. Limit the number of players per server (World of Warcraft) → they make instances
2. Allow many players, but server will get slower and slower (Eve Online)
3. Turn off fighting and other realtime activities in crowded areas. (Cities in WoW)

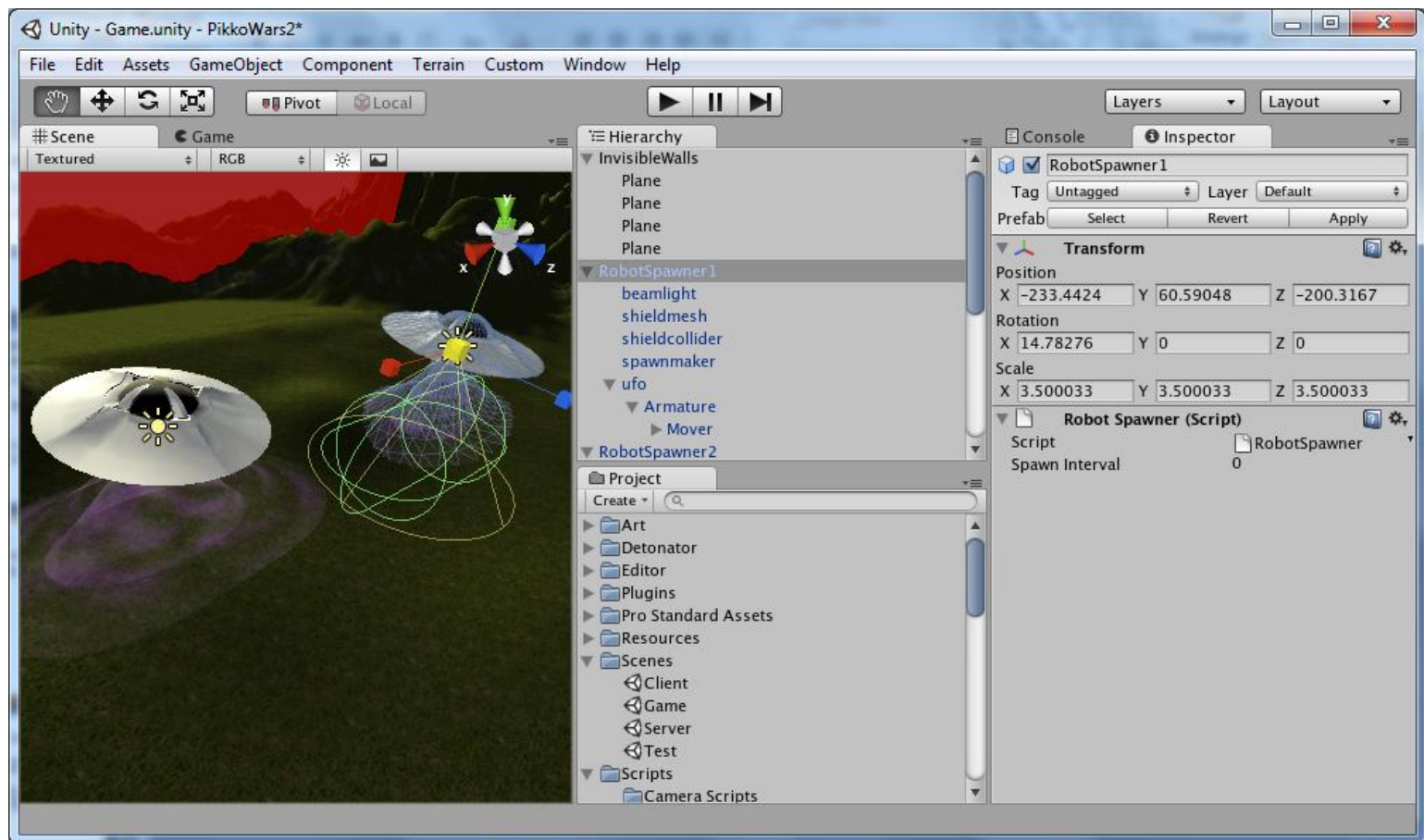
THIS IS BORING!

**Gamers want higher
player density !**

Normal Multiplayer Game Development

- The game developer usually works in a single editor (called a "Game Engine")
- Client and server is often programmed with the same language
- The developer uses a lot of middleware that demands C++ (currently the game industry standard).
- Their focus is fun games, not scalable servers

The Unity Game Engine (GUI screenshot)



How should they build more scalable game servers that can use multicore?

- Use multicore power with a bunch of **C++** threads? - Scary!
- Multicore with **Java**? - Scary too!
- Make a complete game server in **Erlang**?

Why not make a complete game server in **Erlang**?



- Can not use C++ middleware on the server
- Have to use different platforms for client and server
- How do you make game logic scripting easy?
- Erlang is a new programming paradigm for game developers
- It can be done, yes, but there are big challenges !

A more scalable solution: Pikko Server

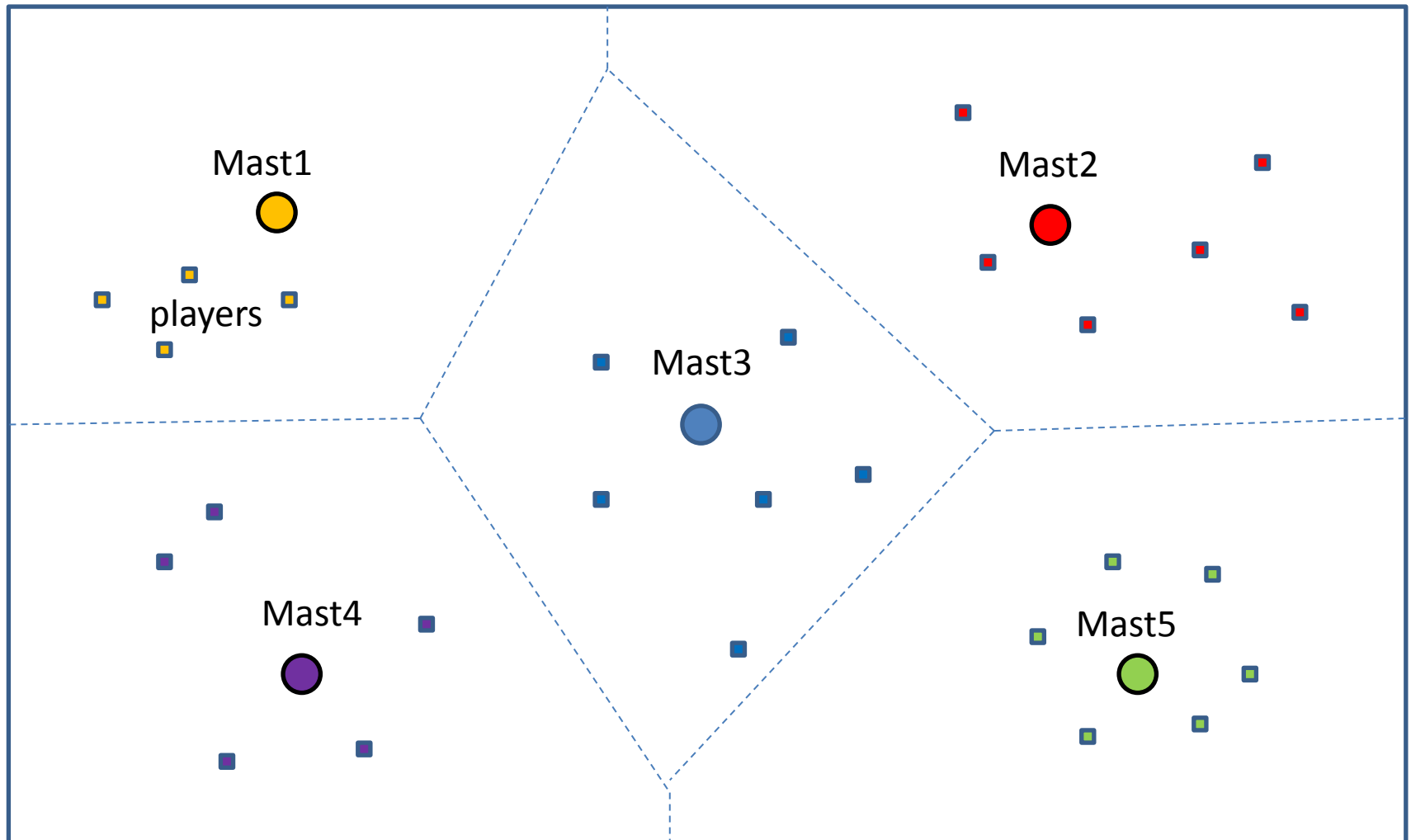
- Keep the game servers the way they are and limit the number of players per game server to around 50-100.
- Put an intelligent router = an Application Level Gateway (ALG) based on Erlang in front of them.
- Move players dynamically between game servers to avoid overload.

How can Pikko Server dynamically route the traffic to the correct game server?

Pikko Server divides the virtual world like a cell phone network !

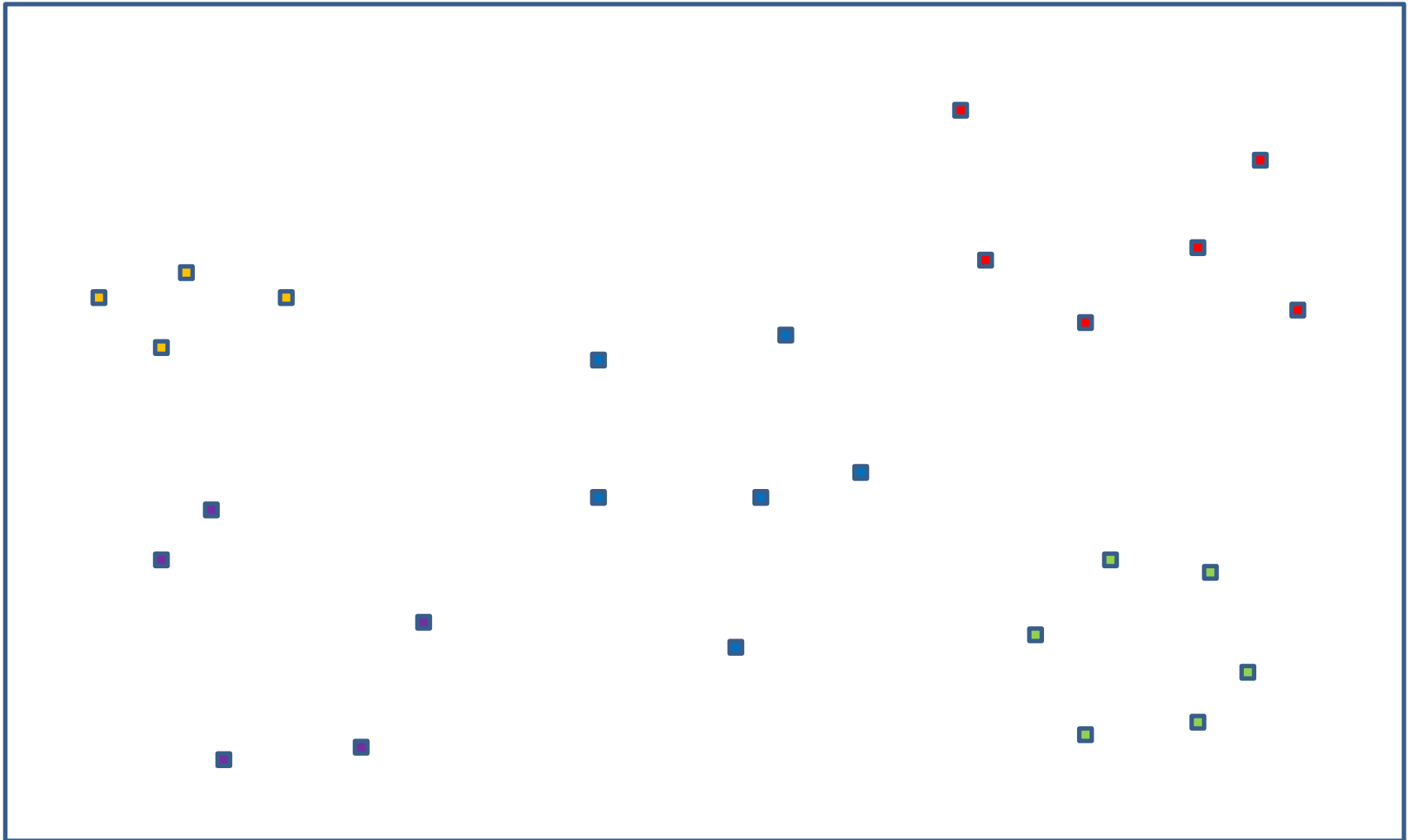
- Every player is a cell phone. 
- Every game server is a mobile network antenna (we call them "masts") 
- Cell phones (players) can be handed over between masts (game servers)
- The masts can move (!)

The PikkoServer view of a virtual world



Client view

Players can see all opponents (in their line of sight)



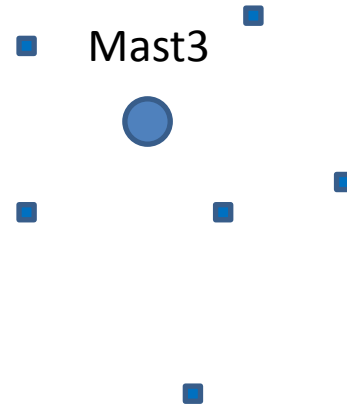
Live Demo Game



Servers hosted by **Game-Hosting GH AB** in Uppsala.

Game server view

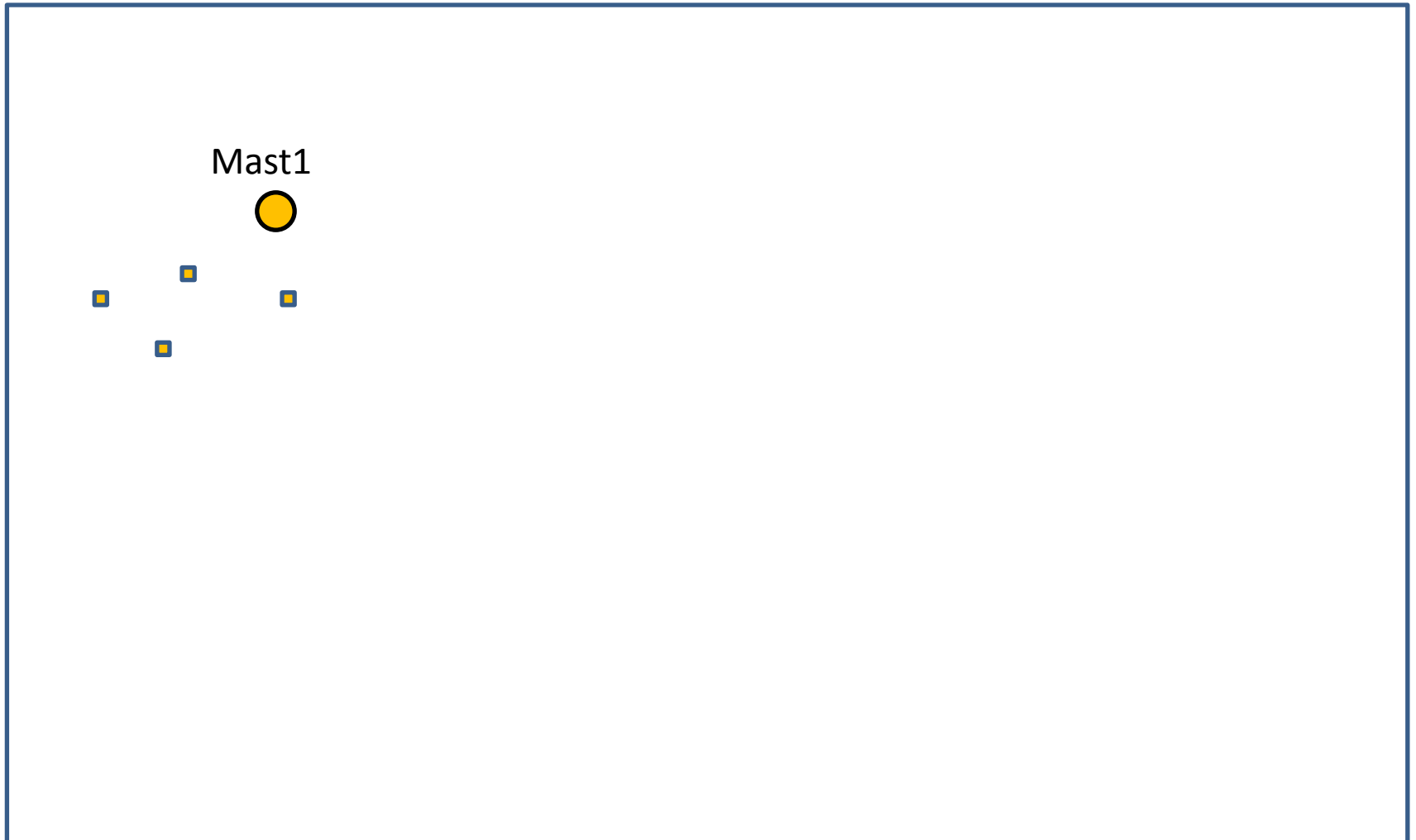
- The view of one game server



Note: It is not overloaded !

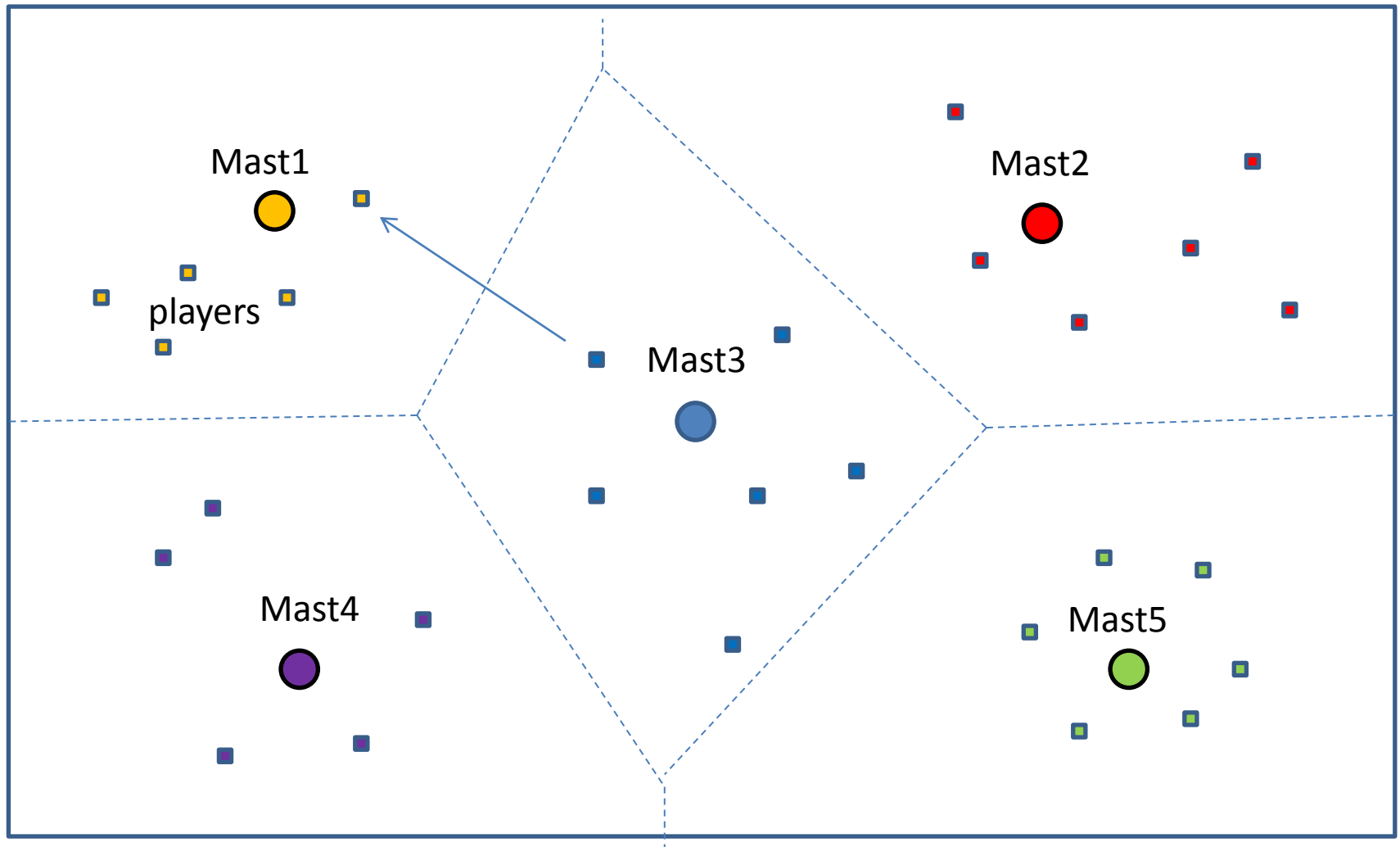
Game server view

- The view of another game server



Every game server (mast) has loaded the complete map of the virtual world

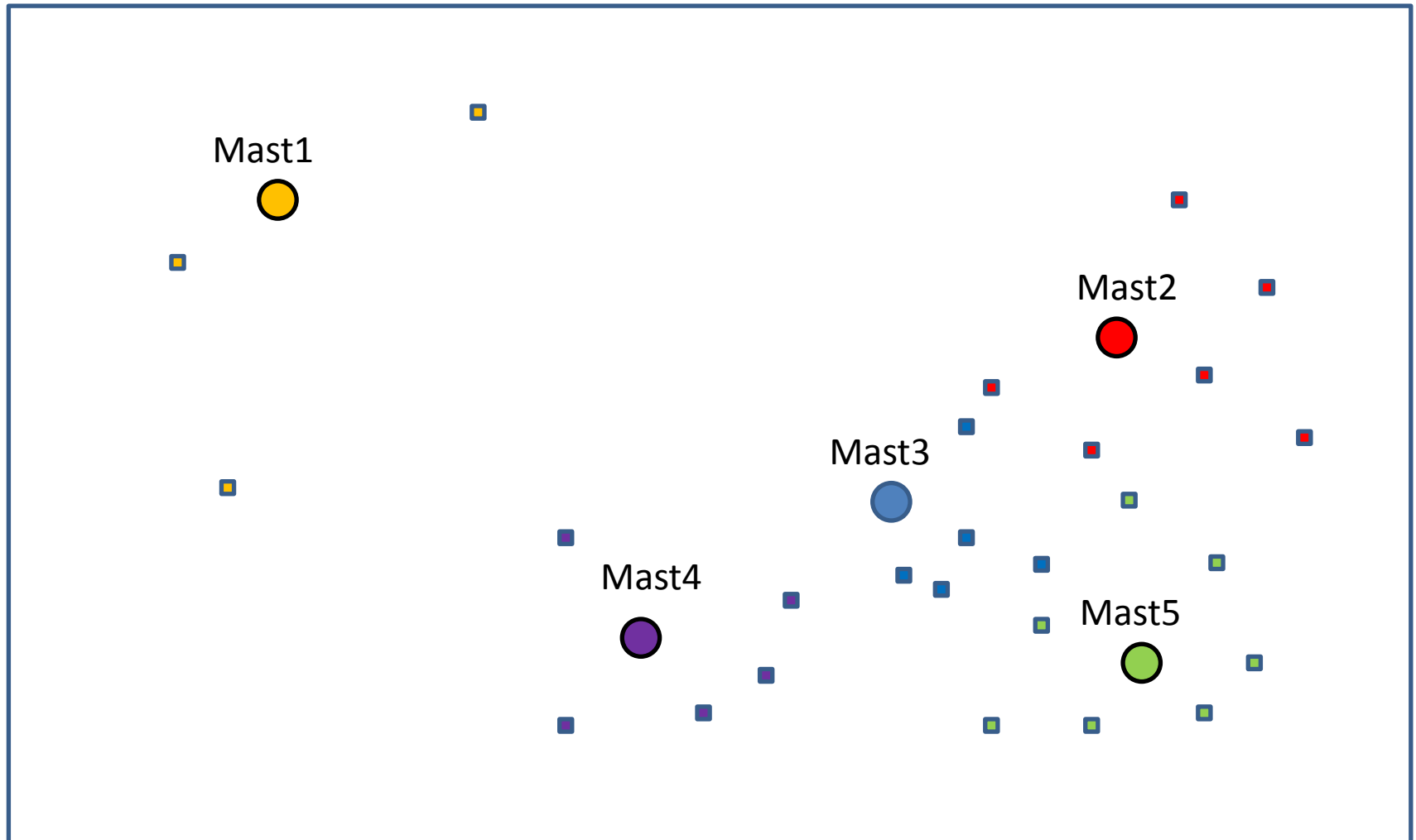
One Player Moves: This movement will trigger a "handover"



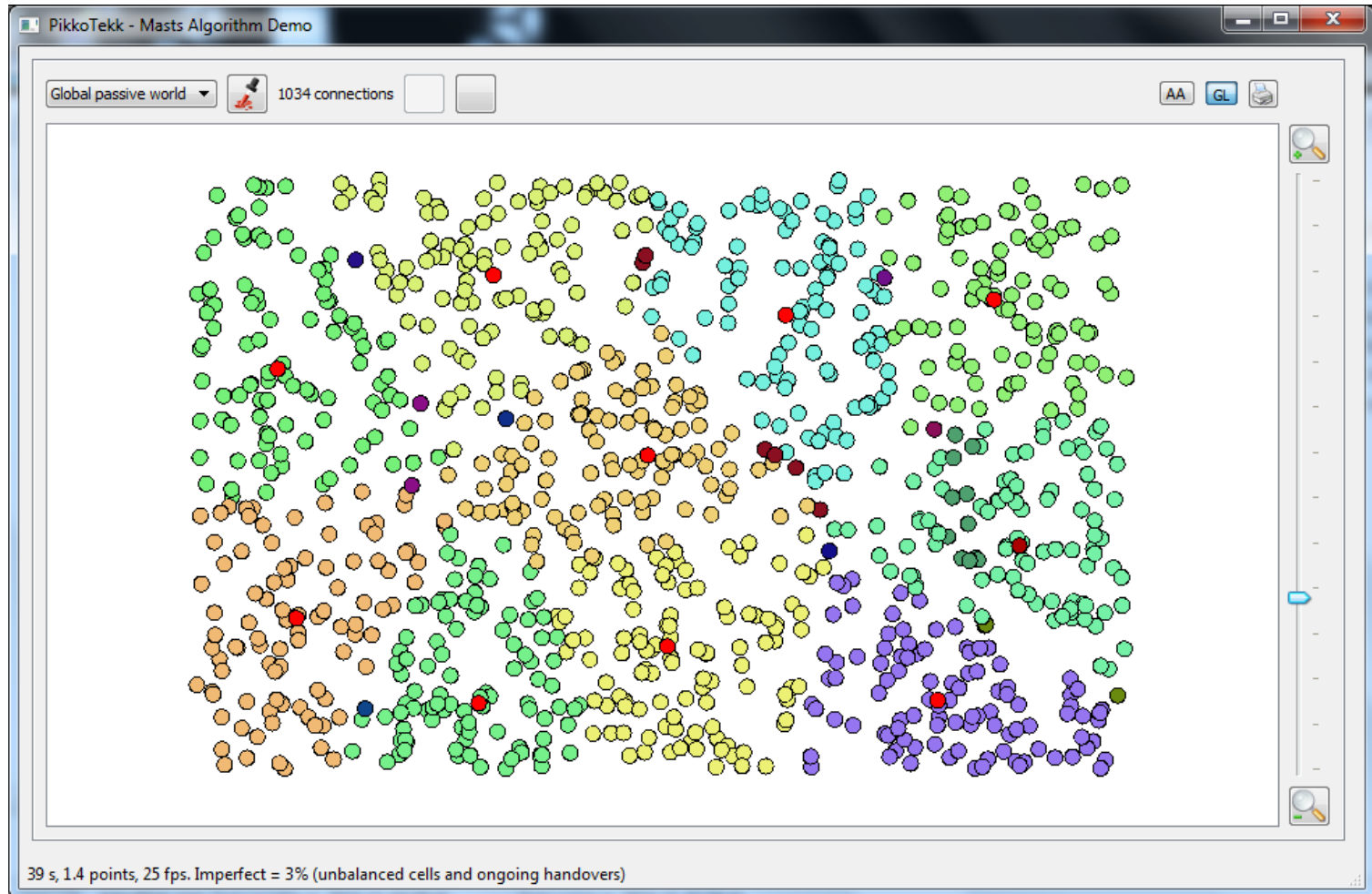
Handovers

- The "Mast algorithm" in **Pikko Server decides** when a player object needs to be transferred from one game server to another.
- Player state is serialized and transferred from one game servers to another (via Pikko Server).
- The game server has to implement our API to be able to participate in handovers.

Massive movements affect positions of masts



Mast Algorithm - Demo



Pikko Server – Now Available

- We have a solution for scaling game servers based on Elrang
- Game developers do not have to learn Erlang, instead they can use Pikko Server to get **scalable many-core power** now.
- Pikko Server can be used in any game engine.
(Our current demo game is based on the Unity Game Engine)
- Marketing and sales starts today.

Current Performance

- We can handle at least 1000 players when game servers send 15 updates (for all moving objects) every second.
- We can multicast 500,000 "game actions" per second per core.
- Close to linear scalability up to 8 cores. We have not tested more yet.
- 1000 active UDP sockets sending 1 Megabit/s each = 1 Gb/s total server bandwidth downstream from Pikko Server.

Test lab hosted by **Game-Hosting GH AB** in Uppsala.

World record attempt soon

- PikkoTekk will soon make a world record attempt with 1000 real players with a fast paced action demo game.
- Please send me an email now if you want to be part of this event.

david.almroth@pikkotekk.com

Subject: World record

Questions & Answers

David Almroth

www.pikkotekk.com