

# IDS – P2P Project

Boris Fritscher & Ulysse Rosselet



Survive the parasite apocalypse.

# Features



- All features of RMI version except ability to join a game in progress
- Option to choose a player name when connecting to pre-game server
- CPU through leader (master)
- Infection management through leader (master)
- Converting disconnected human player to cpu
- Leader election if leaders is disconnect



tile\_A.png



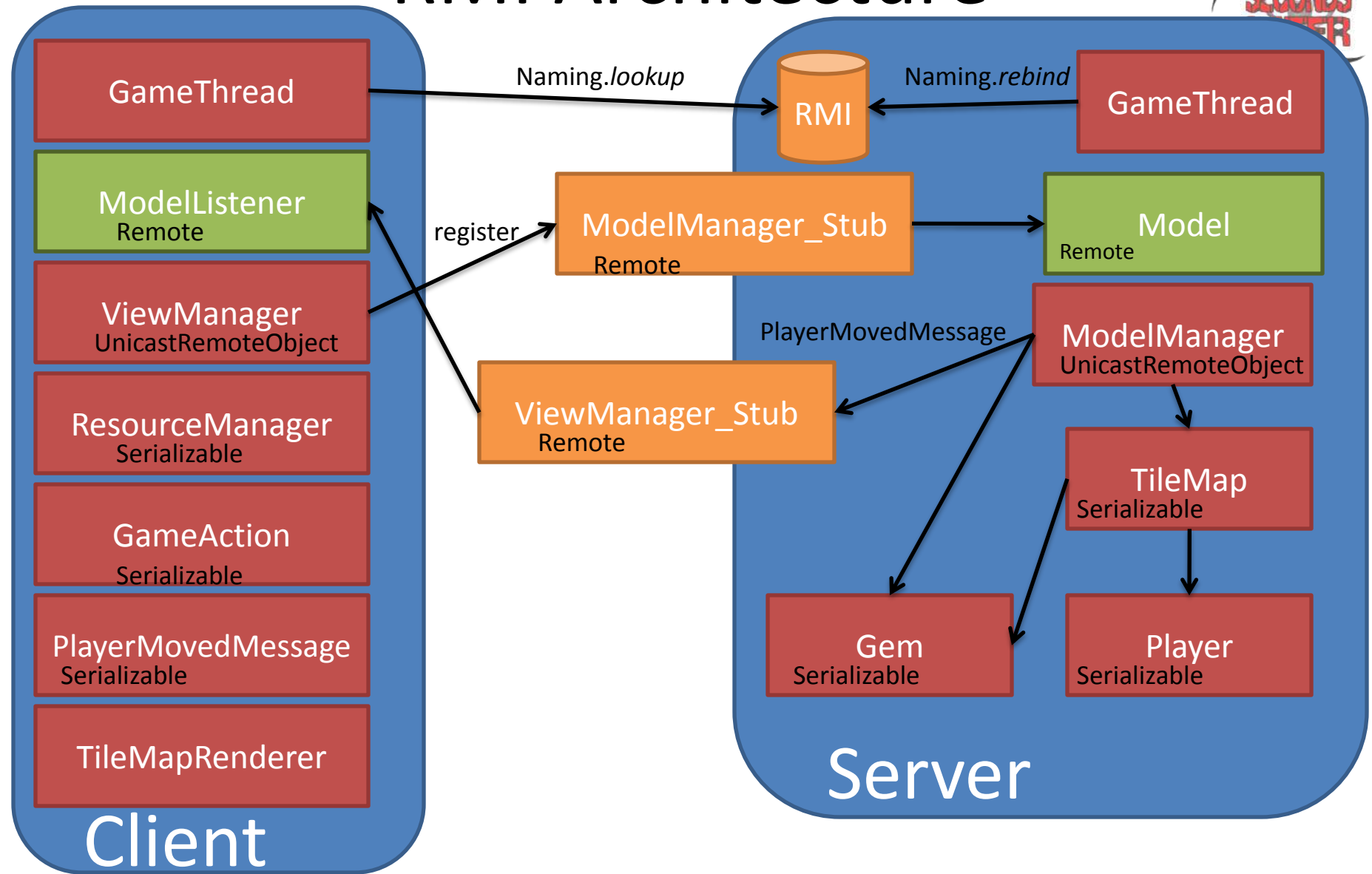
tile\_B.png



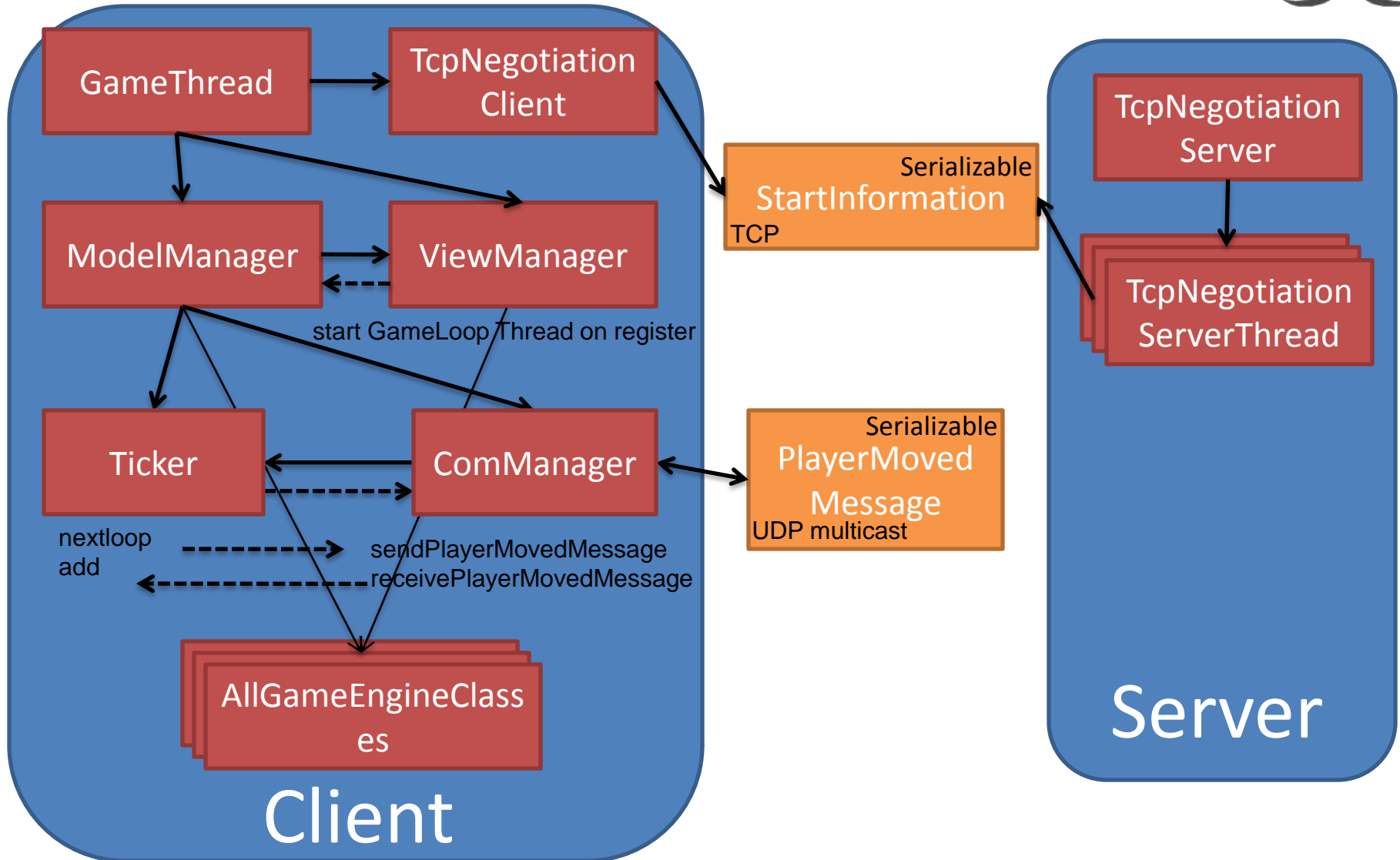
tile\_C.png



# RMI Architecture



# New Architecture Com. Only



# TCP: StartInformation



- String status
  - UPDATE, FULL, GO
- String multicastGroupIP
  - ip of the multicastgroup onto which to broadcast
- int playerId
  - client player number
- String[] players
  - Names of the players
- boolean[] cpu

# UDP: PlayerMovedMessage



- int playerNumber
  - Id of the player sending the message
- int direction
  - Int = Config.Direction
- int round
  - Round to which message applies
- int health
  - Health of the player;
- int infectedPlayerNb
  - Nb only set by the master
  - else -1
- int masterNb
  - Nb of the master if master
  - else -1
- no need for CPU flag because
  - If (masterNb <> -1 && masterNb <> playerNumber) = CPU

# ModelManager: gameLoop



- if(master): send PlayerMovedMessages for CPUs
- send myPlayerMovedMessage
- get nextloop(round) from Ticker (blocking)  
=(PMM for every player)
- if(master & game running): check that there is one infected, else infect someone randomly
- if(!master): update my CPU list
- update Players with health, direction, infection
- round++

# Ticker: nextloop(round)



- Have we received all messages for the round?
  - Yes: return them
  - No: wait
- Check if we missed a message for too long? Yes
  - We are the master -> TakeAction
  - We are not the master
    - Did we received a message from a master?
      - Yes: do nothing master will resolve the problem
      - No: wait
    - longer delay passed perhaps master is dead in between -> TakeAction
- TakeAction
  - Am I the lowest playerId alive?
    - Yes: become Master, generate CPU messages for missing players



# Bumps in the road



- Javadoc
  - Raise `NullPointerException` vs return null
- Strange Object behaviour in TCP
  - without `connection.reset()`
- mutable array (`static final String[] players`)
  - `clone()`!
- `setTimeToLive(0)`
  - no comment...



# Changes



- Removed RMI/serializable - `gui.ResourceManager`, `gui.input.GameAction`, `model.components.*`, `game.*`
- `config.Config` – added methods to set a custom player name
- `game.GameThread` – change to start `TcpNegotiationClient` and handle response to start game
- `game.Model` – added `getNbPlayers()`
- `game.ModelManager` – updated gameloop logic, `ComManager`, `Ticker`, player name integration
- `game.PlayerMovedMessage` – added new fields (`round`, `health`, `infectedPlayerNb`, `masterNb`)
- `game.ViewManger` – removed rmi disconnect handling, added round rendering counter
- `gui.TileMapRenderer` – added Custom name support
- `gui.graphics.IntroJMenu` – change menu for starting tcp server / and joining a game
- `gui.graphics.IntroPane` – added field for custom name
- `gui.graphics.ScreenManager` – changed startup form `GameThread` to `TcpNegotiationServer`
- `model.TileMap` – support for custom name
- `model.components.Player` – support for custom name, generate directions for cpu

# Changes



- network – all new
  - ComManager – send and receive UDP multicast messages
  - StartInformation – Message Object used for TCP communication
  - TcpNegotiationClient – get start information from server
  - TcpNegotiationServer – Server starts a TcpNegotiationServerThread for each client connection
  - TcpNegotiationServerThread – sending information on each state change until start is received
  - Ticker – Manages messages order and fallback if not received

# Questions ?

