

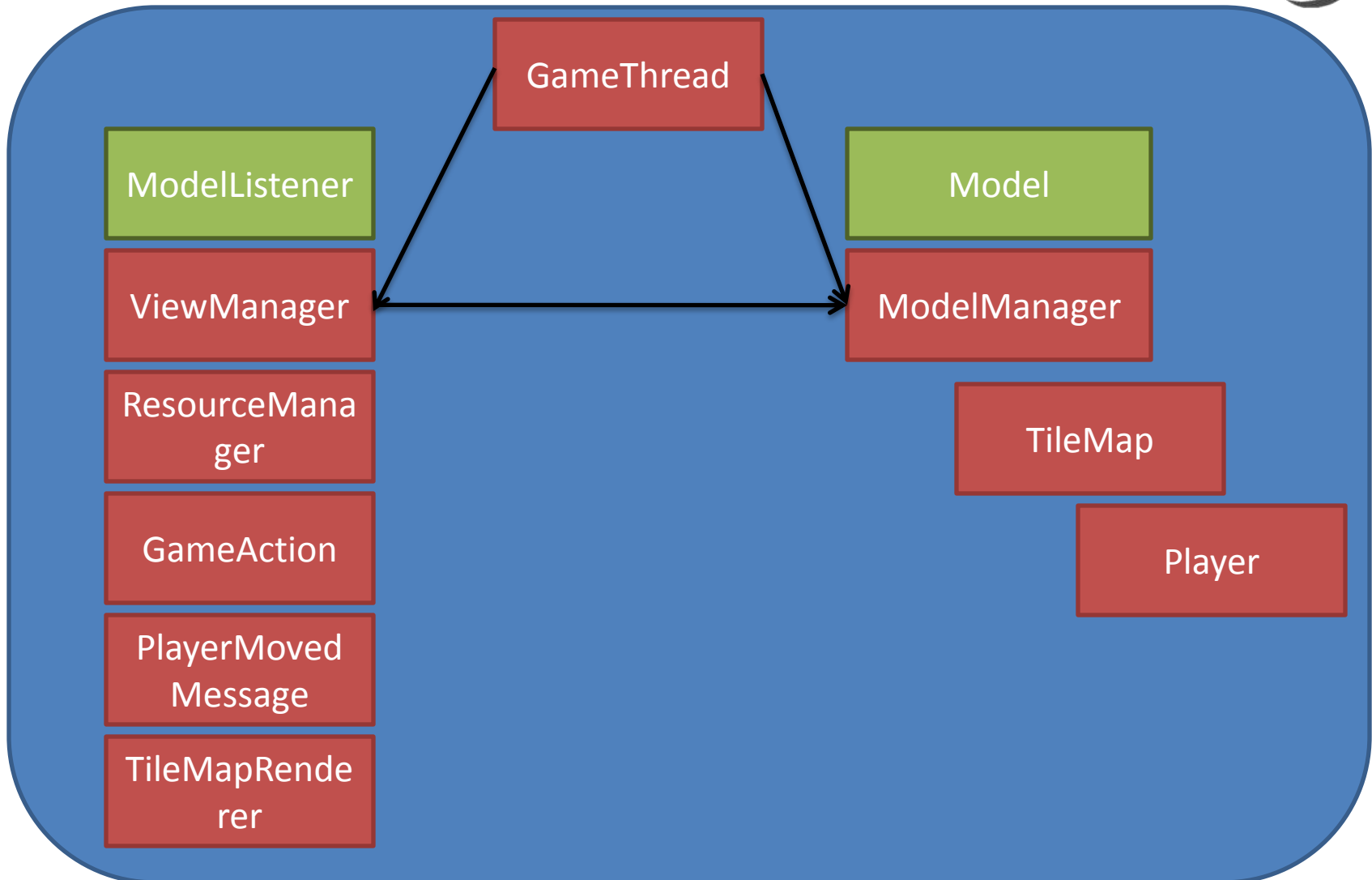
# IDS – RMI Project

Boris Fritscher & Ulysse Rosselet



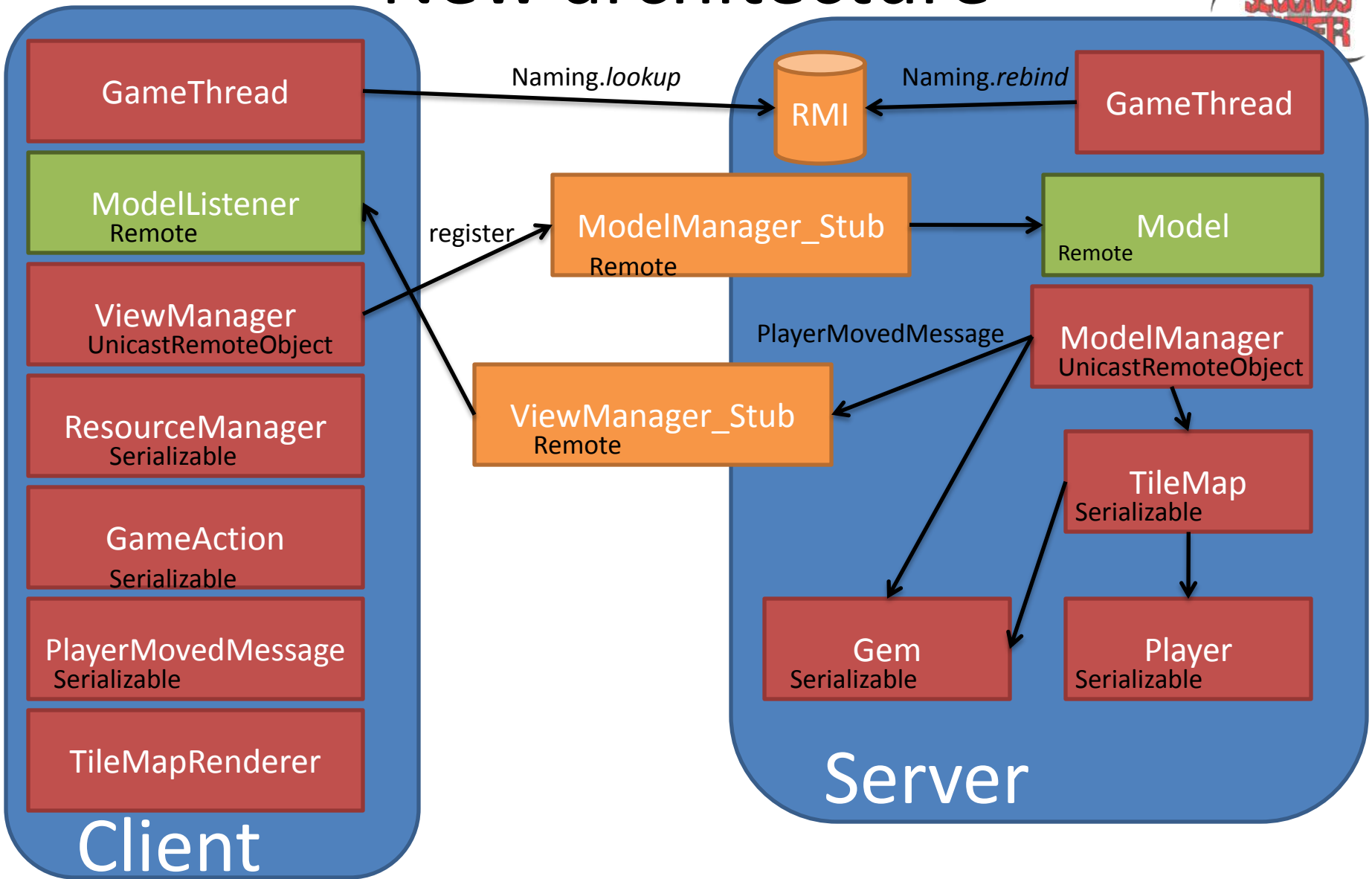
Survive the parasite apocalypse.

# Former architecture





# New architecture



Client

Server

# RMI Exception handling



- Connection
  - Server not found
  - Game full
- Soft disconnection
  - Set cpu to true
  - Allow reconnect
- Hard disconnection
  - Server side automatic client disconnection
- Start server
  - autodetect if we have to run rmi
  - Able to stop (unbind model)
- Kill server
  - Client auto disconnect timeout



# RMI Example



```
try {
//create rmi registry
java.rmi.registry.LocateRegistry.createRegistry(1099);
} catch (RemoteException e) {
    try{
        //rmi already started? try to connect
        java.rmi.registry.LocateRegistry.getRegistry();
    } catch (RemoteException ex) {window.displayErrorPane("Unable to contact/create RMI
registry");
return null;
}
}
//we have a registry ready test if server is already in registry
try {
    try{
        Naming.lookup("ModelManager");
    }catch (NotBoundException e){
        //registry has no server we can create it
        model = new ModelManager();
        Naming.rebind("ModelManager", model);
        return model;
    }
    window.displayErrorPane("Server already running!");
    return null;
} catch (RemoteException e) {
    window.displayErrorPane("Server unable to start/connect to rmi!");return null;
}
}
```

# Advanced Features



- Engine

- Refactoring in order to remove need of lapi.jar
- Map is generated from file (Multiple tiles possible A-Z)
- Ability to send from the server a message to be drawn on an specific individual client
- Server menu
- New sprites!



- Items

- Radar powerup shows all the living players on a mini map
- Life powerup gives 20 health points (maximum health is 100%)
- Skull, where a player died. Health decreases rapidly when standing near the corpse



# Advanced Features



- Infection
  - Randomly : at startup or if the infected player dies
  - Transferred through contact with another player
  - Infected player's health is decreasing (28s until death) but runs faster (has his own sprite)
  - When a player is dead : « ghost mode » he can move and see all other players but others do not see him
- Rounds
  - For a round to start, the game waits until all human players are ready (spacebar pressed)
  - The last living player (survivor) wins the round
  - The game ends after 5 rounds displaying the score board
  - Server ends after all human players disconnected



# Advanced features Example



```
//check if we have a winner
int remainingPlayers = livingPlayers.size();
if(remainingPlayers > 1){
    //infecting new player
    infectRandomPlayer();
}else{
    //we have a winner!
    try{
        Player winner = livingPlayers.get(0);
        int newscore = score.get(winner.getID()) + winner.getHealth()*10;
        winner.setScore(newscore);
        score.put(winner.getID(), newscore);
        for (ModelListener listener : listenerList) {
            try {
                listener.displayMessage( winner.getName() + " Won round "+round, 3000, 200);
            } catch (RemoteException e) {
                System.out.println("unable to send displaymessage");
            }
        }
        System.out.println(winner.getName() + " Won round " + round);
    }catch (IndexOutOfBoundsException e) {
        // catch if we made an update but nobody was alive
    }
}
```

Y offset

duration (ms)



# Important changes in java files



- Removed all engine.\*
- game.GameThread: RemoteException handling for connection errors
- game.Model: added unregister/stop
- game.ModelListener
  - added int state to PlayerMovedMessage (for Warmup defined in Config.STATE\_\*)
  - added displayMessage(String message, long duration) to display message on clients
- game.ModelManager: numerous changes for serving game logic
- game.ViewManager: numerous changes for displaying game logic (messages, scoreboard)
- gui.graphics.ScreenManager: new menu for server
- gui.ResourceManager: added loading of item(gem/settables) and tiles images
- Gui.TileMapRenderer: display logic (infected skin, radar, ...)
- Model.components.Player: added attributes, health, infected +PlayerDieTimerTask
- Serialized: GameAction, PlayerMovedMessage, ResourceManager, TileMap, Gem, Settable, Player, SerialTimer, PlayerDieTimerTask

28 Seconds Later

Warm up! (press space when ready)



	Shenon 100% human score 4
	Egon 100% cpu score 0
	Ray 100% cpu score 0
	Cindy 100% cpu score 0





28 Seconds Later

Ray has been infected !!



	Quentin 100% human score 0
	Egon 100% cpu score 0
	Ray 99% cpu score 0
	Cindy 98% cpu score 0





28 Seconds Later

**GAME HAS ENDED! you can disconnect!**

**Scores :**

**1 : Egon : 1200**

**2 : Ray : 1090**

**3 : Cindy : 170**

**4 : Quentin : 0**

	Quentin (EAG) human score: 0
	Egon (FV) cpu score: 1200
	Ray (FV) cpu score: 1090
	Cindy (EAG) cpu score: 170





# Questions ?

