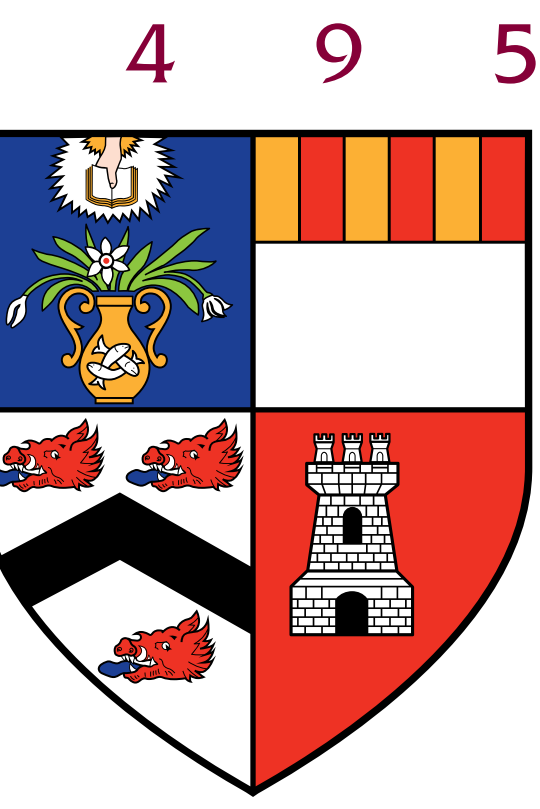


# RLEREWOLF – REINFORCEMENT LEARNING AGENT DEVELOPMENT FRAMEWORK FOR THE SOCIAL DEDUCTION GAME WEREWOLF

Georgi Ventsislavov Velikov – u01gvv16@abdn.ac.uk

University of Aberdeen



## Context

The game of Werewolf is a social deduction game where players are randomly assigned specific roles, at the start of a game, which fall under one of the general factions – *evil*, *neutral*, and *good*. The goal of the *good* faction is to kill off all *evil* members and vice-versa. The game is split into two times of the day – *day* and *night*. In the base version of Werewolf, there exist four roles:

- *Villager* – Part of the *good* faction and has no special actions; can vote during the *day* in order to execute a *Werewolf* suspect.
- *Guard* – A member of the *good* faction; can protect players during the *night* from being attacked, including himself.
- *Seer* – A member of the *good* faction; can find out the role of a player during the *night*.
- *Werewolf* – *Villagers* that turn into *Werewolves* during the *night*; member of the *evil* faction and have the ability to attack other players during the *night*.

## Research Question

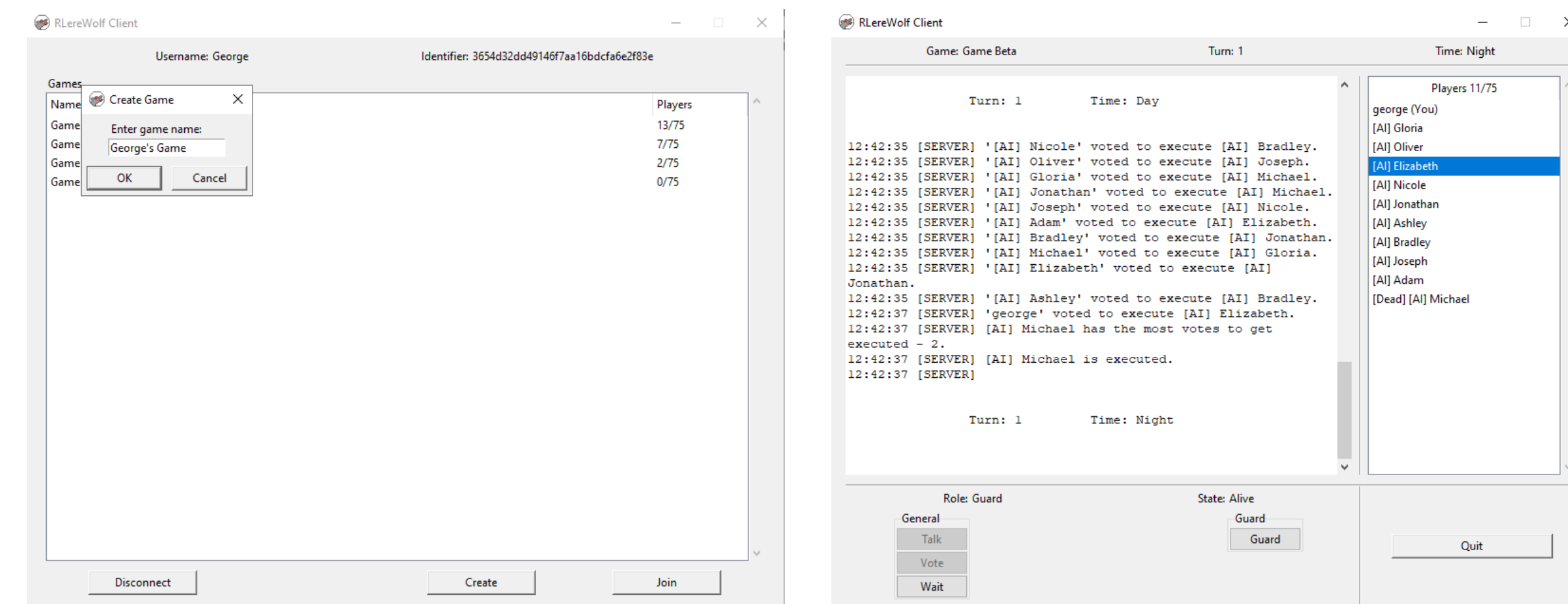
The framework proposes a new framework which aims to provide analytic tools and built-in *Agents* which can play all supported by the game roles. Moreover, the improvements to the communication protocol and the addition of *trust* and *honesty* factors should perform better than a stochastic player approach and worse or equal to the optimised *Q-learning* approach.

## Conclusion

The current built-in *Agents* experiment results, unfortunately, do not reach the performance of *AiWolf's Agents*. This is largely attributed to the unfinished state in which the built-in *Agents* are in. However, the framework, RLeReWolf, has shown its potential for the development of *Agents* whose performance is not entirely bound on the *Game* implementation. Furthermore, the metrics and analysis tools the framework provides are not currently provided by any existing framework.

## Contribution 1: Client

An access point for users to play the game Werewolf with other humans or *Agents* on the targeted RLeReWolf *Server*. The *Client* has a an easily expandable Tk based GUI which is rendered on run-time by the built-in Pygubu builders and renderers.



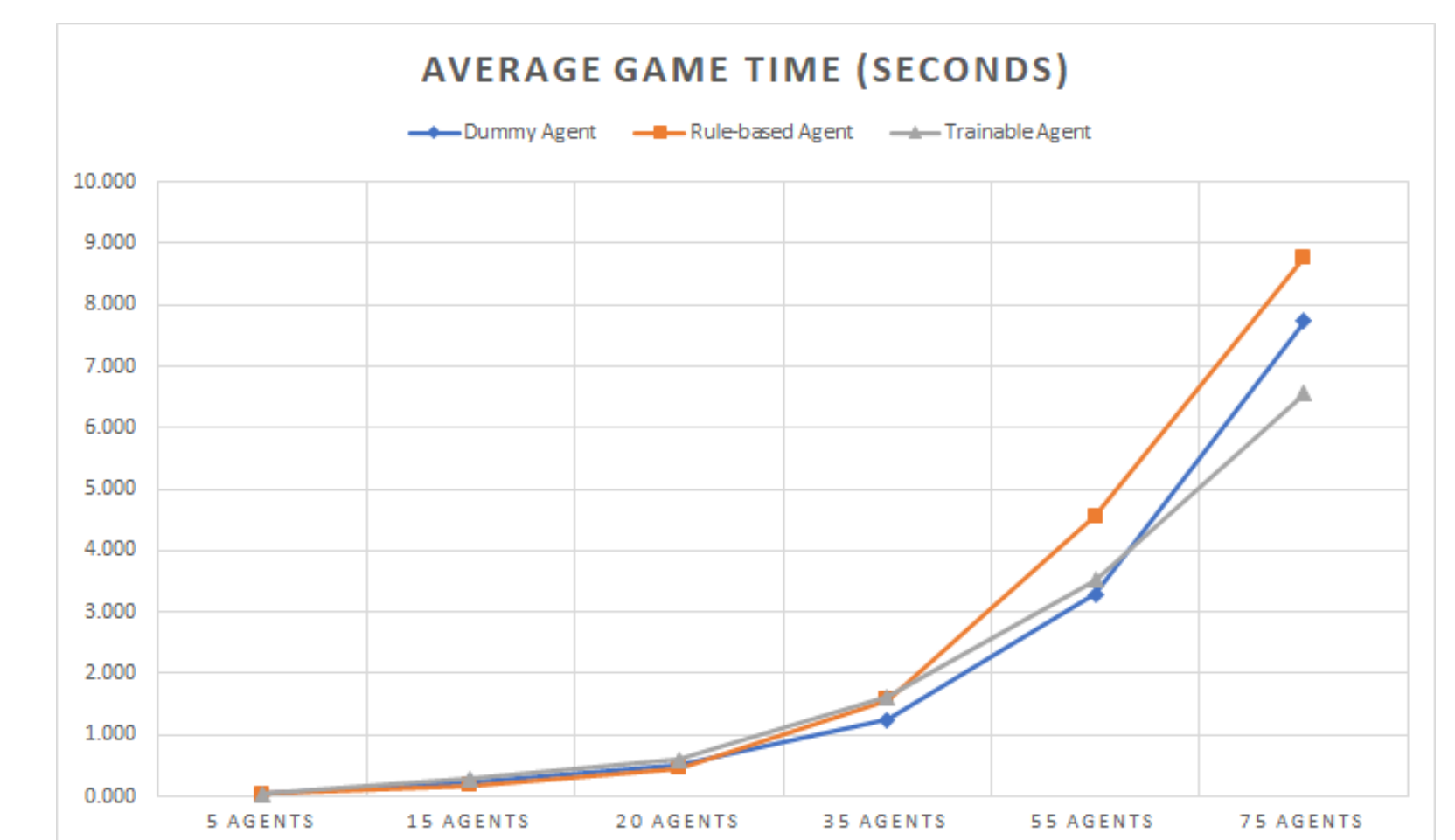
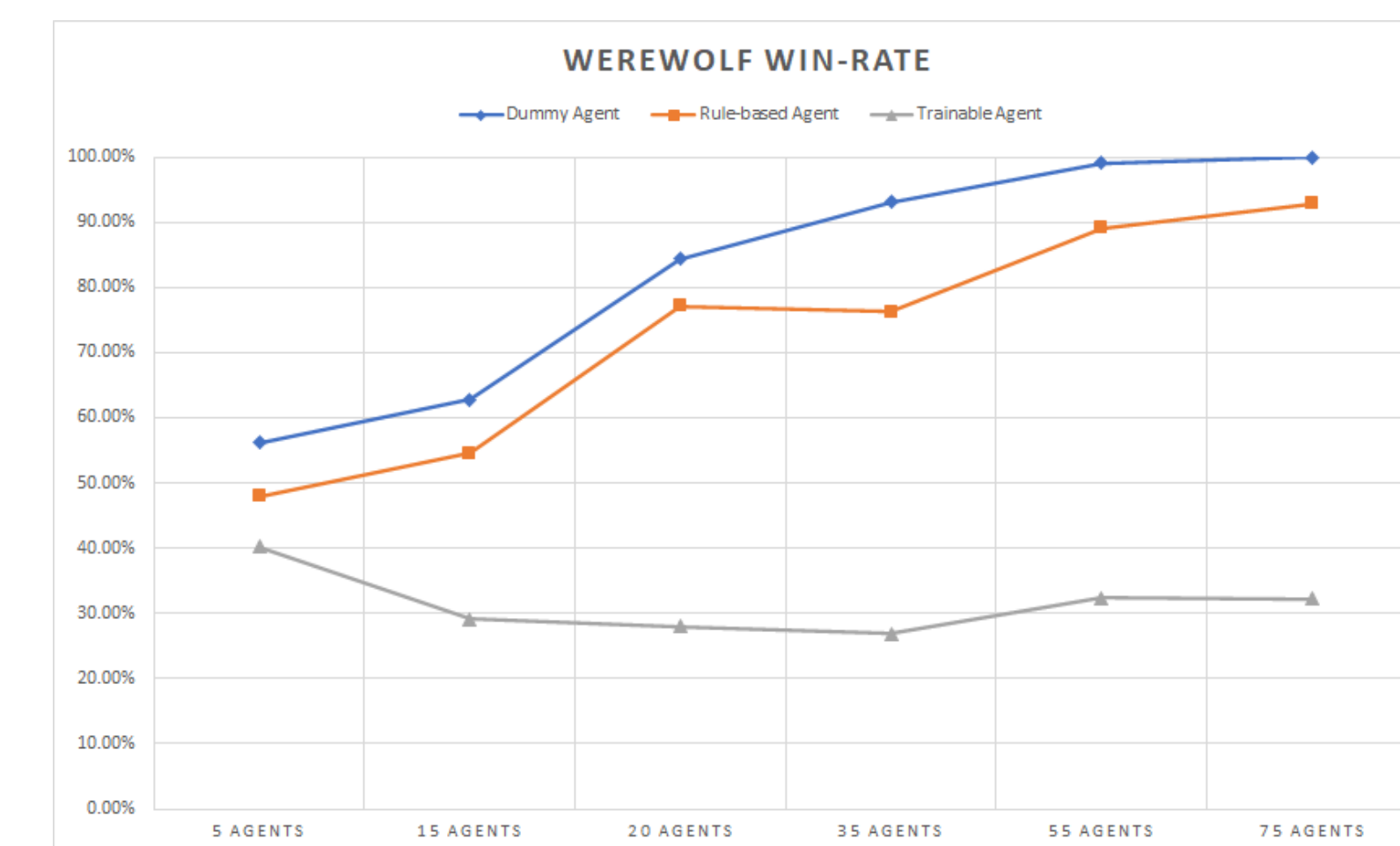
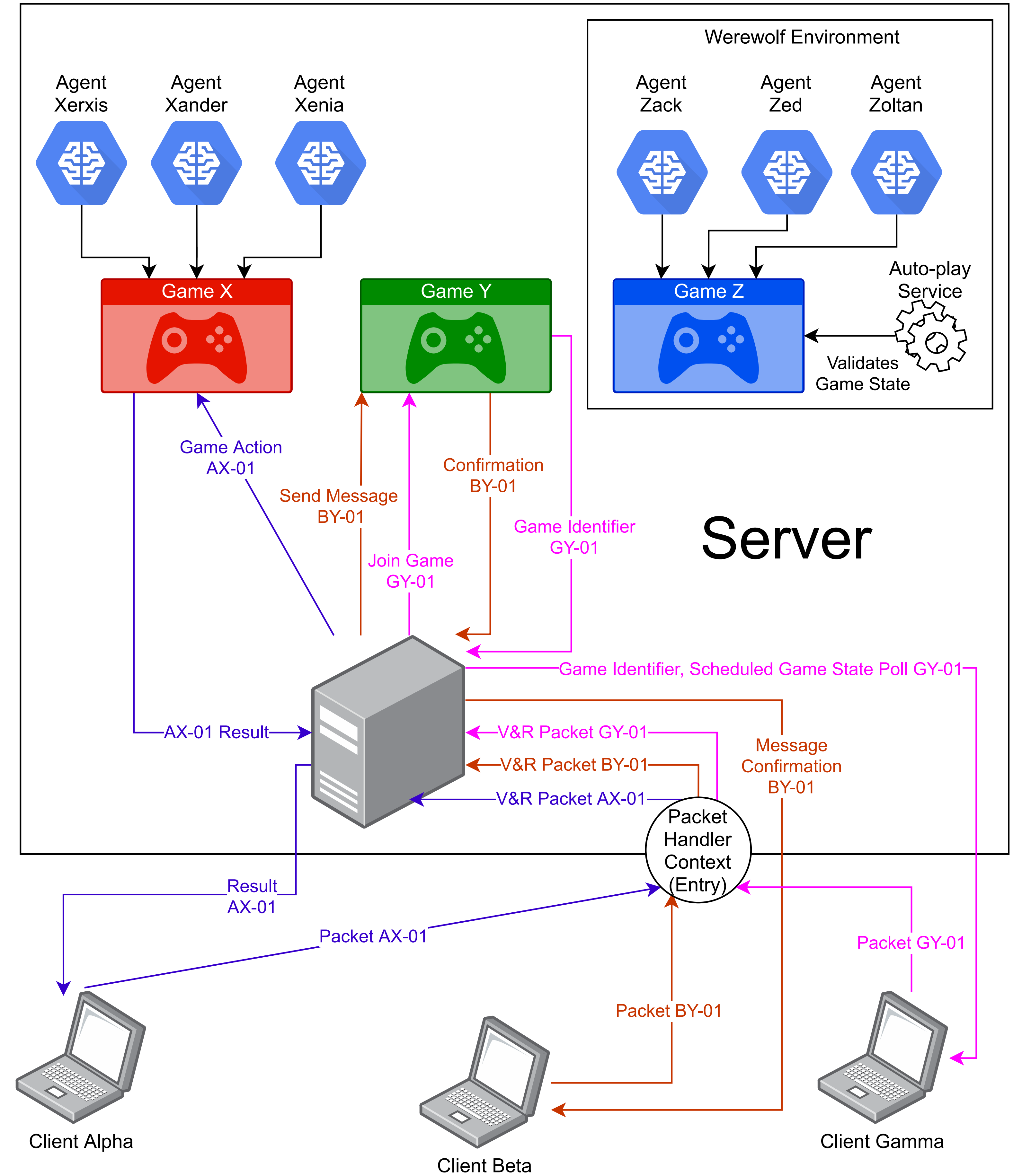
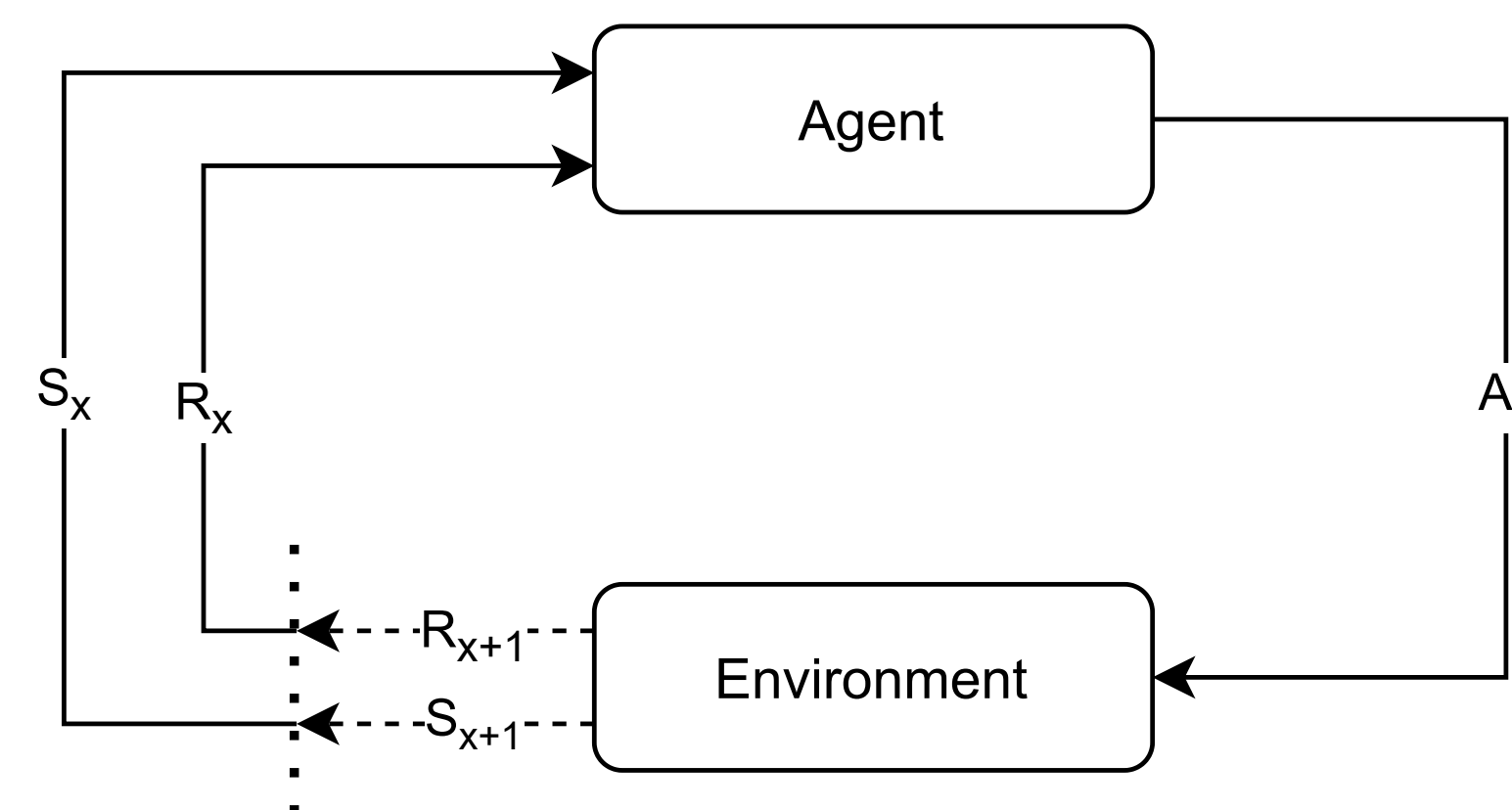
## Contribution 2: Development Framework

The development framework consists of the code base for the Werewolf game implementation, the analytic utilities provided by the built-in training *Environment*, comprehensive server & game activity logging, and the modular implementation of the four subsystems: *Client*, *Server*, *Game*, and *Environment*.

## Contribution 3: Built-in Agents

Three built-in *Agents* which allow the framework's users to use them as a foundation for future *Agents*. The three *Agents* are:

- *Dummy Agent* – A stochastic *Agent* that does random *valid* actions.
- *Rule-based Agent* – An *Agent* with an *honesty* factor who votes for the least *trustworthy*, according to them, *Player*.
- *Trainable Agent* – An *Agent* that can learn from playing multiple games of the current Werewolf *Game* implementation. Has no pre-existing knowledge of the game and needs to *train* in order to learn the game's rules and how to optimally play it.



- ✂ Tool  
tinyurl.com/RLeReWolfFramework
- 📄 Dissertation  
tinyurl.com/RLeReWolfPaper
- 🌐 LinkedIn  
linkedin.com/in/georgi-velikov/