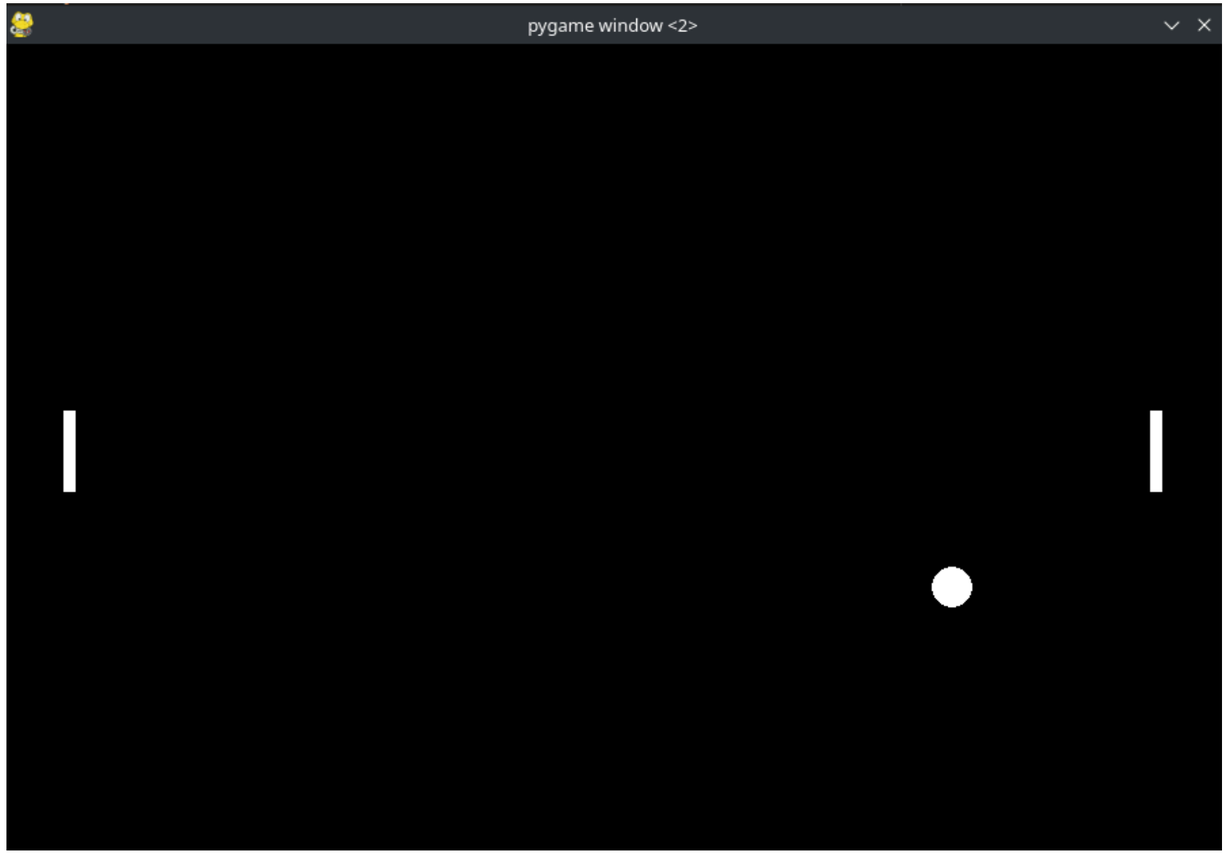


DPONGPY: Adding Communication Protocols and Etcd State Management

Exploring Distributed Systems Through Interactive Gameplay

A.Y. 2022/2023

Lorenzo Massone
lorenzo.massone@studio.unibo.it

Objectives

Goal: Extend communication protocols in DPONGPY beyond UDP.

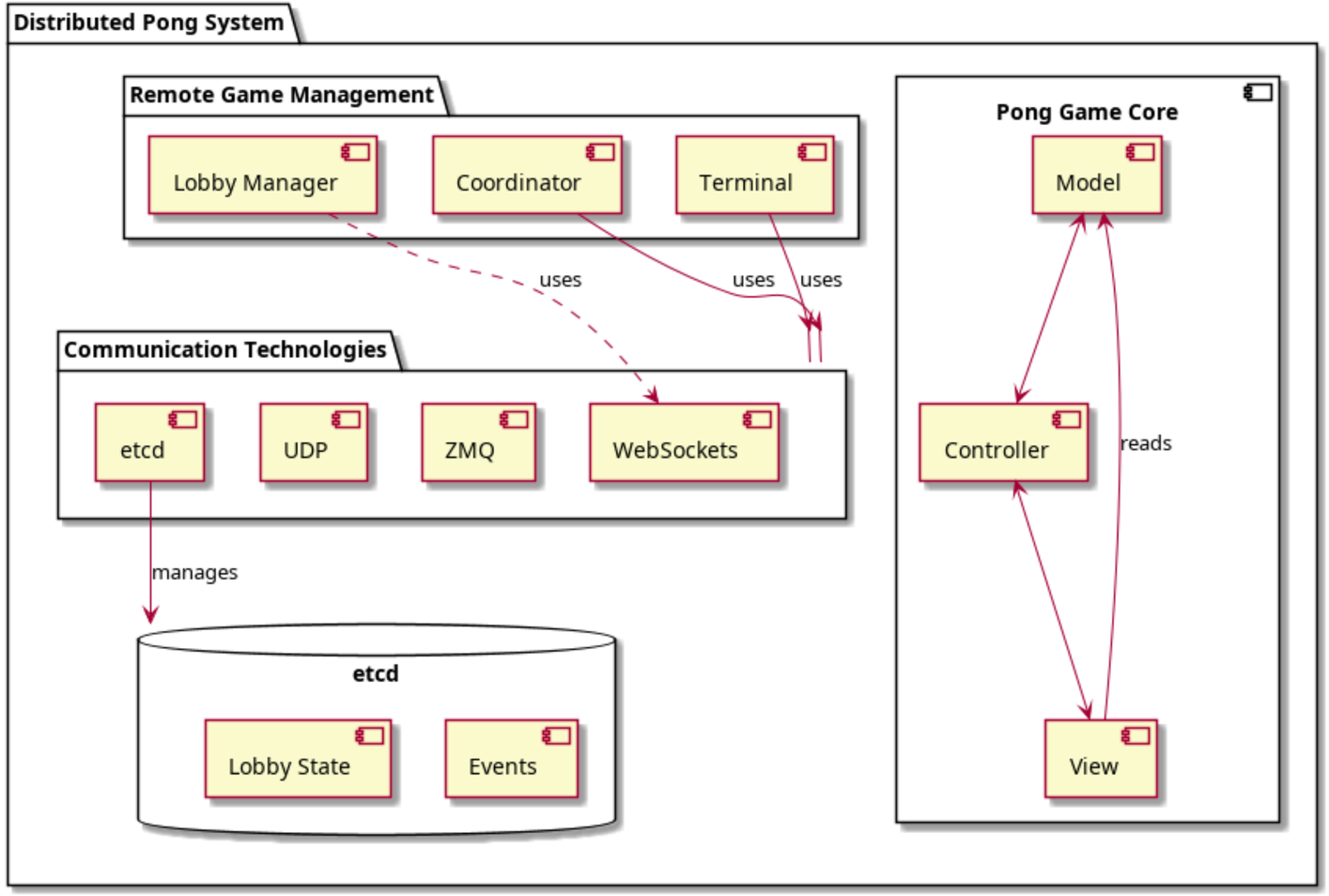
- Integrations:

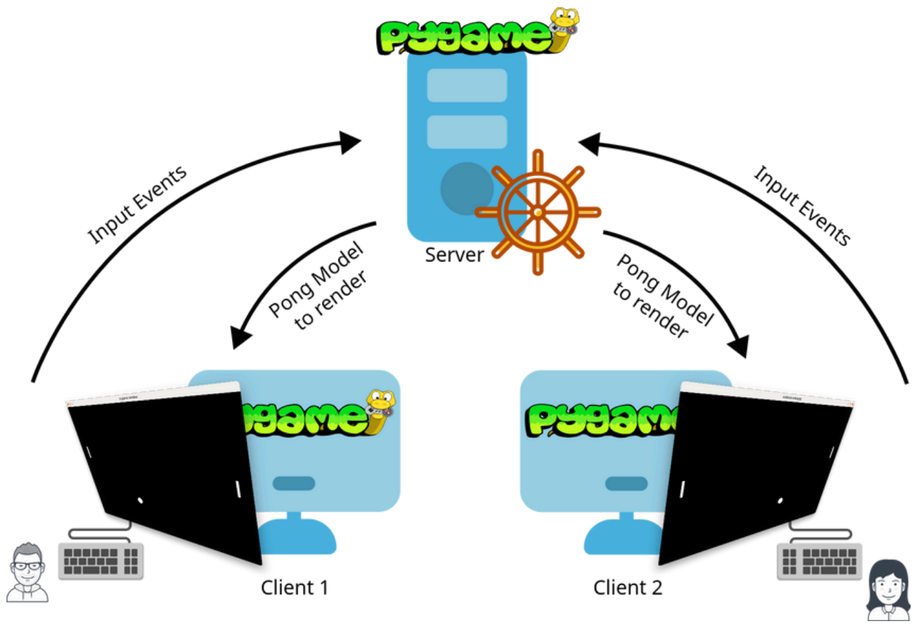
1. ZeroMQ: Alternate protocol for coordinator-terminal communication.
2. WebSockets: RESTful API integration to start games once all players connect and WebSockets to manage connection during the game
3. etcd: Distributed state management without a central coordinator.

- Focus: showcase trade-offs between different communication protocols

Architecture Overview

The architecture is MVC based, with other packages introduced
to implement the new communication protocols





Centralised Architecture: Coordinator-Terminal Pattern

Usable with UDP, WS and ZMQ protocols

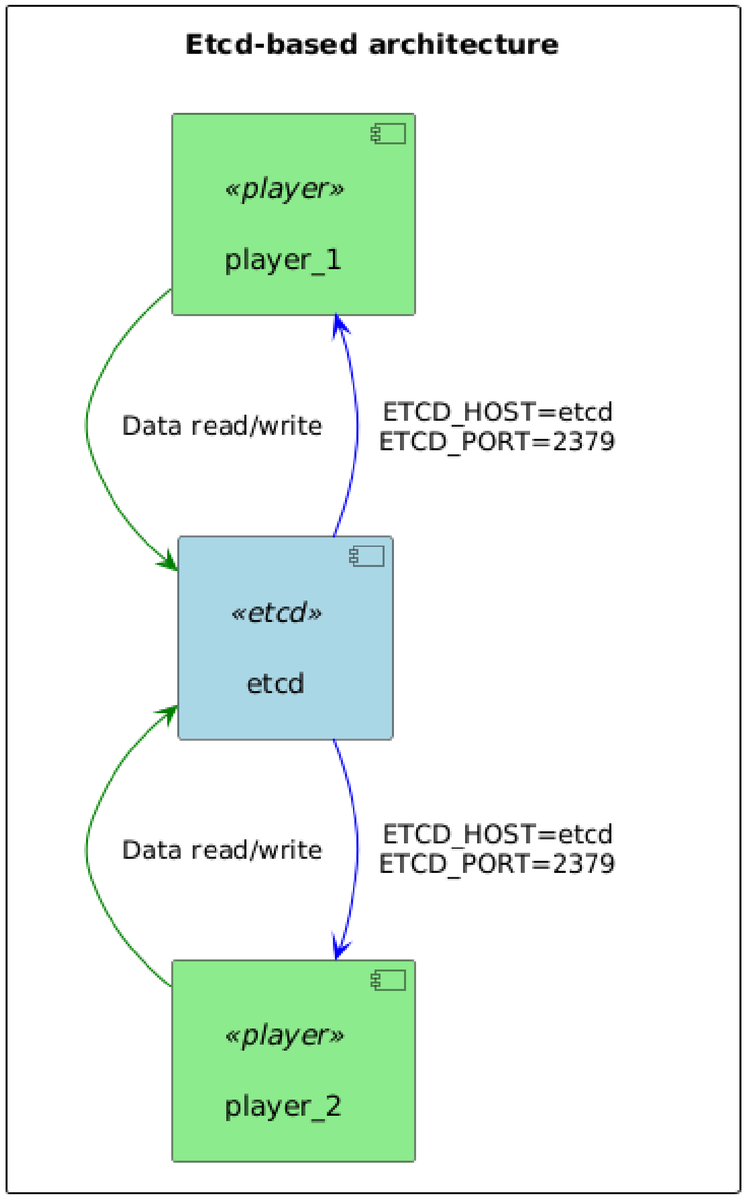
- Coordinator: Central server managing game state, synchronizing terminals, and handling game loop.
- Terminal: Client nodes handling player input, receiving game state updates, and rendering the game.

Etcd-based architecture

The coordinator is unnecessary. The terminals will perform the following actions:

- Elect a leader using etcd.
- Store the game state in etcd through a shared key.
- Synchronize the state by monitoring updates in etcd.

Ensure consistency via etcd's consensus mechanism.



Protocols & Trade-offs

Protocol	Consistency	Latency	Use Case
UDP	❌ Low (No ACKs)	⚡ Very Low	Real-time games with tolerance for packet loss
ZeroMQ	✅ Medium (Ordered delivery)	🕒 Medium	Systems requiring reliable messaging
WebSockets	✅ High (TCP-based ordering)	🕒 Medium	Real-time apps with session coordination
etcd	🔒 Very High (Raft consensus)	⏳ High	Decentralized state sync with strong consistency

Technologies

Poetry

Poetry was used for managing the project's Python dependencies and virtual environment.
It made the development process easier by maintaining a clean, consistent setup for running the project

Technologies

ZeroMQ

ZeroMQ, was used in the project for high-performance messaging between distributed components. Its lightweight design and support for various messaging patterns enabled scalability across the system.

Technologies

FastAPI

FastAPI was instrumental in enabling the API interactions required for the lobby management system in the WebSocket-oriented implementation

Uvicorn

Uvicorn was used as the ASGI server for running the project's FastAPI application.

Technologies

etcd

etcd was used for the decentralised architecture as the key-value store for managing the state and events.
It made it easy to maintain consistency and reliability across the terminals.

Deployment instructions

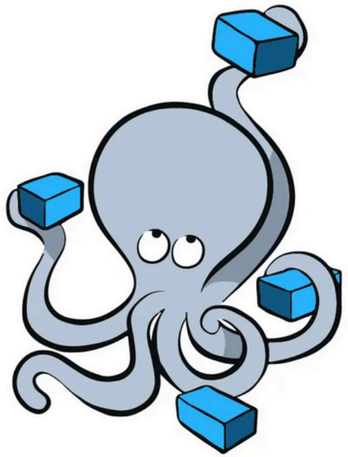
Centralised architecture

```
poetry install
poetry shell
# Coordinator
python -m dpongpy --mode centralised --role coordinator --host localhost --port 5000 --comm-type COMM
PROTOCOL
# Terminals
python -m dpongpy --mode centralised --role terminal --host localhost
  --port 5000 --side right --keys arrows --comm-type COMM PROTOCOL

python -m dpongpy --mode centralised --role terminal --host localhost
  --port 5000 --side left --keys arrows --comm-type COMM PROTOCOL
```

Deployment instructions

Etcd-based architecture



docker Compose

First is essential to allow X Server connections from the container to the host machine:

```
xhost '+local:*'
```

```
docker compose up
```

Deployment instructions

Etcd-based architecture

```
services:
  etcd:
    image: bitnami/etcd:latest
    environment:
      - ALLOW_NONE_AUTHENTICATION=yes
      - ETCD_ADVERTISE_CLIENT_URLS=http://etcd:2379
    ports:
      - "2379:2379"
    networks:
      - dpong_network

  player_1:
    build: .
    environment:
      - PLAYER_SIDE=left
      - DISPLAY=${DISPLAY}
      - ETCD_HOST=etcd
      - ETCD_PORT=2379
    volumes:
      - /tmp/.X11-unix:/tmp/.X11-unix
    depends_on:
      - etcd
    networks:
      - dpong_network

  player_2:
    build: .
    environment:
      - PLAYER_SIDE=right
      - DISPLAY=${DISPLAY}
      - ETCD_HOST=etcd
      - ETCD_PORT=2379
    volumes:
      - /tmp/.X11-unix:/tmp/.X11-unix
    depends_on:
      - etcd
    networks:
      - dpong_network

networks:
  dpong_network:
    driver: bridge
```

Grazie dell’attenzione!

Demo