Cairo University

Faculty of Computers and Information

CS252

Software Engineering II

"DRDC"

# Decentralized Replicated Data Store

## Yusuf Fawzy Elnady

March – 2019

• Description:

This program is about enabling **the peer to peer communication**. By letting the **DRDC (Decentralized Replicated Data Store)** to add as many users as you want. Then they can send to each other the messages or fake data by using the port numbers of each. Our program also supports the **Network Discovery** and the **data store consistency**.

At the end **each peer** connected to the network will have in **its data store** all the **transactions** or **messages** have been **sent** and **received from the start of the program.**
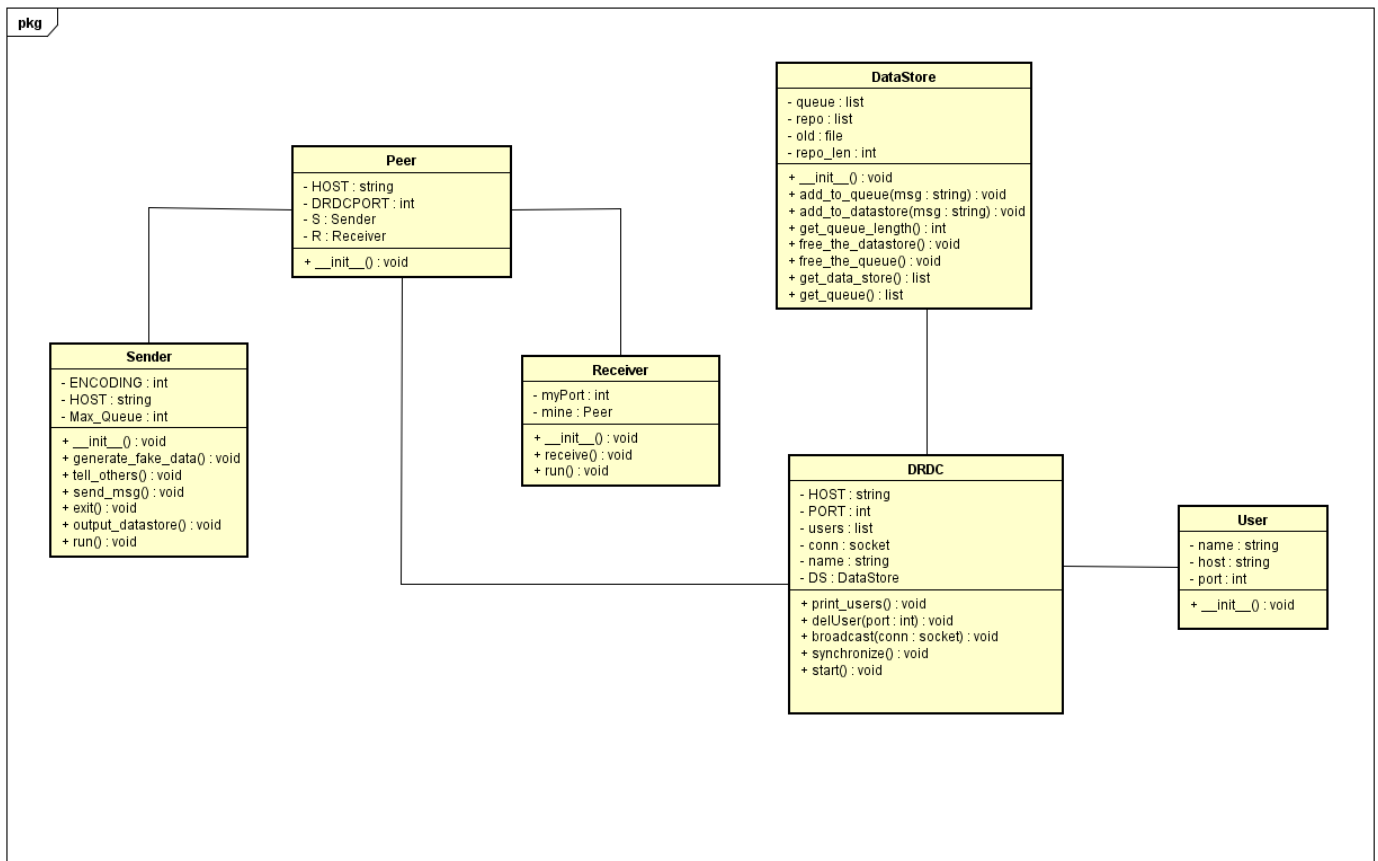
• Implemented use case:
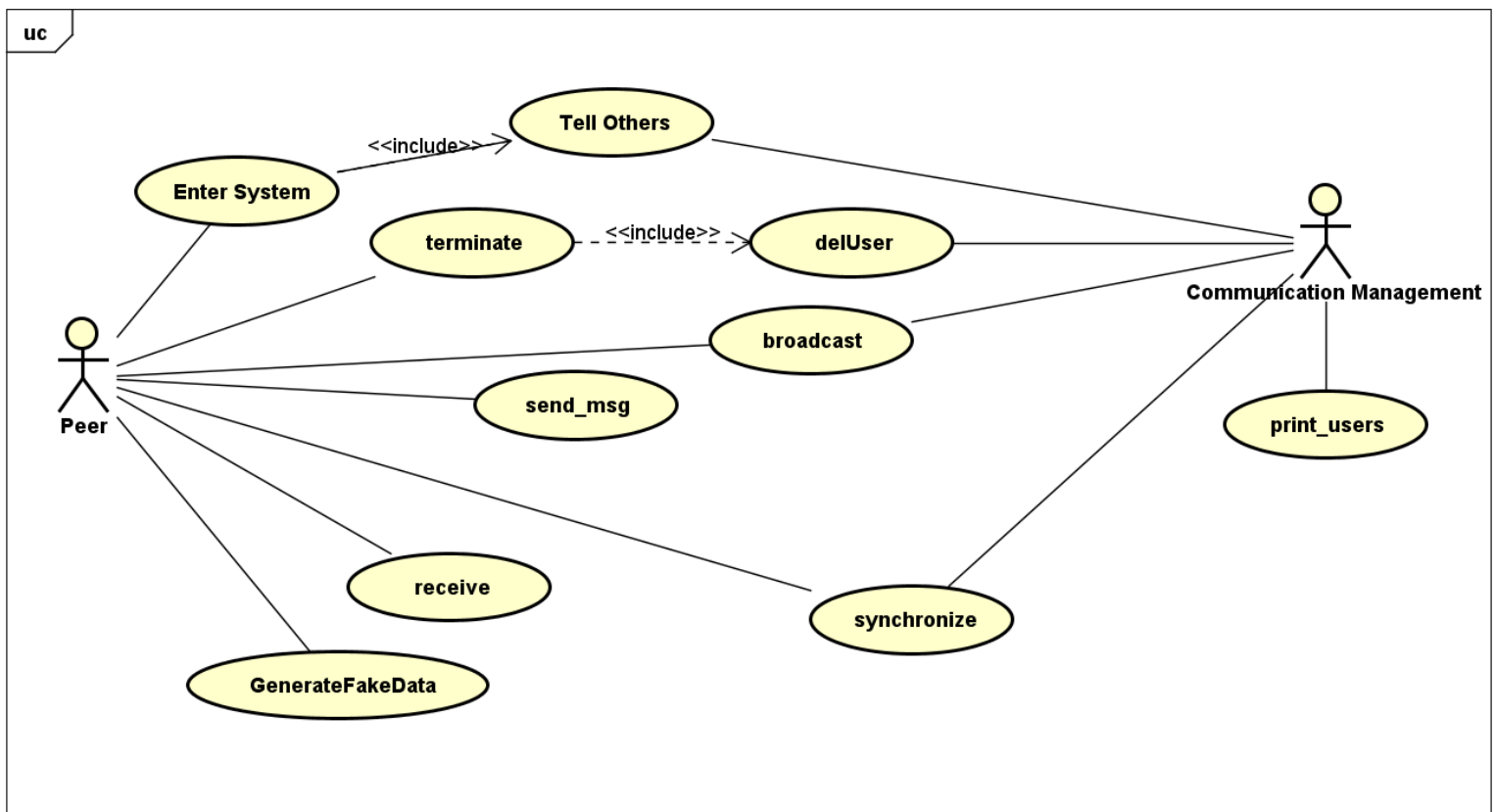
• System Architecture:
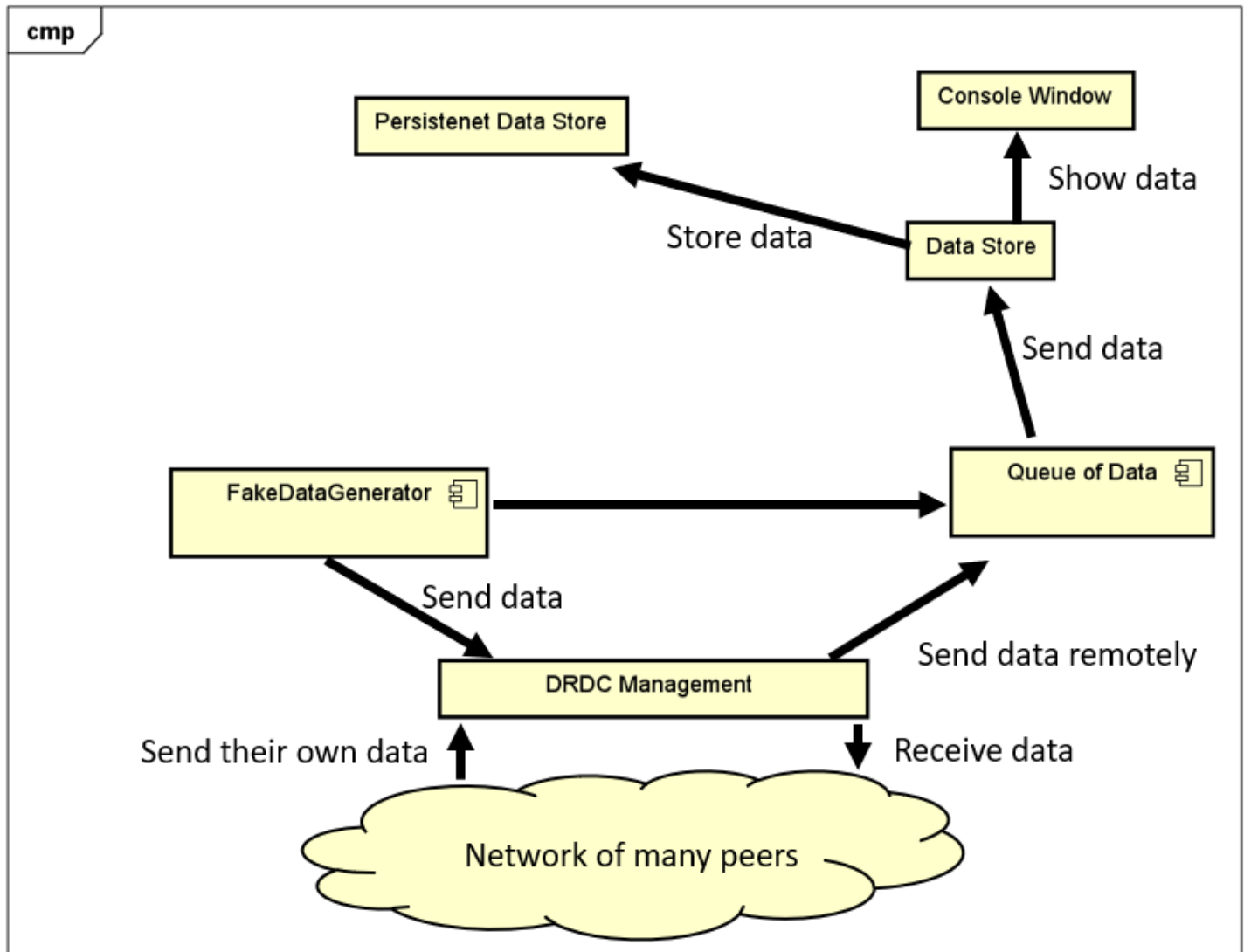
- ## System Design:

- # Details:

  - o **Sender**: allow the peer class to perform **the sending functionality** in a **thread** so it can **send messages** to any other peer **using its port**. It also has some useful methods as sending messages or **broadcasting me** to others.

  - o **Receiver**: allowing the peer to **continuously check** if there's anyone to communicate with me. It also **runs in another thread** which is turned on by the peer instance.

- o **Peer**: This class is **the initiator** of any **new peer** wants to be added to our Decentralized Replicated Data Store. Its responsibility to take the **name** of the user and its **port**, then runs the sender and receiver objects in **two different threads.**
- o **User**: It's a small entity that holds only the **name**, **host** and the **peer** of the user.
- o **Datastore**(repository): This class have many useful functions such as **add_to_queue** which handles the **problem of caching** and **not to overwhelm** the datastore with many data. So, its responsibility to take data into queue and to free the queue and so on…
- o **DRDC**: This is called the **peer communication management**, it can **print** the current users, **broadcast** the msg which is the network discovery role, also continuously **synchronize the data** with the datastore of each node or each pair, so the **datastore will be consistent among all others**.

• Installation Guide:

- o To **start** using this application all you want is a **python terminal**.
- o First, write in the **directory textbox bar** in your **File Explorer**: "DRDC.py" → this will start your application.
- o Then to **start adding peers** to your program or to your network, type again in the **directory field**: "Peer.py" → this will **add a new peer** to your application.
- o **No need** to install any modules or servers to start using this application
- o You must assure that you have python 3.7 or above if any.
- o This program is built using **Python, PyCharm.**