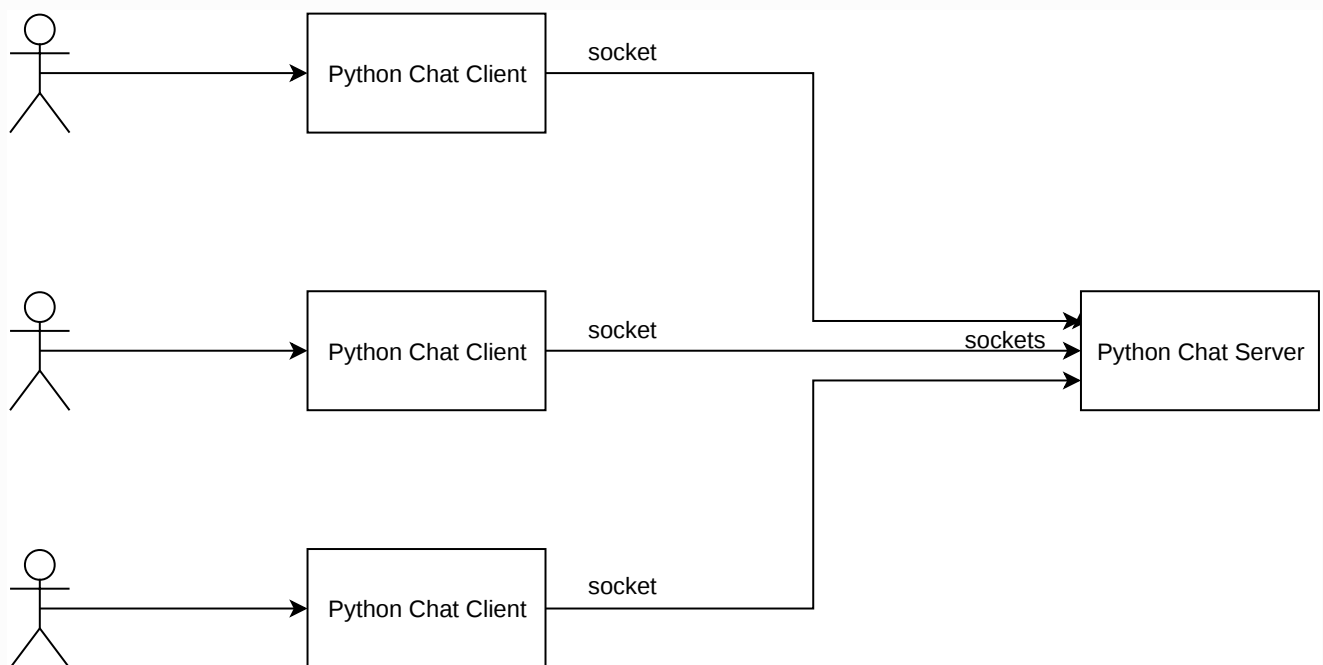


# Individual Project 1

15 NOV. 2016

**ALI ABDULMADZHIDOV**

## 1. Architectural diagram



## 2. Design decisions

I decided to use client-server architecture because it was recommended by task and it seemed simpler to implement. Python was chosen as programming language for it's fast and user friendly development process. Client-server architecture is based on TCP socket connections. For asynchronios input/output service uses select loop.

## 3. How to use

### 3.1 Server

Server works on 35535 port (it is free according IANA site), so wee need to configure docker to open and listen port on host machine. That's all.

```
sudo docker run -p 35535:35535 st.os3.su:5000/ali_abdulmadzhidov
```

## 3.2 Client

```
sudo docker run -a stdin -a stdout -i st.os3.su:5000/ali_abdulmadzhidov_client
```

After launch client would ask host ip address to connect. You can use external ip address of machine or ip address of docker container (172.17.0.1).

Then server generates username for client and connects him to chat.

There is 4 command in chat that is available to user:

- exit - exits from chat;
- listall - prints out list of all users, that are in chat, and shows who has administrative rights;
- password type - user just types password and gets administrator rights;
- kick - kicks out user with username.