HomeWork 3

Names

Hanan Egbaria
ID : 305297178
hananeg@post.bgu.ac.il

Majdy Aburia
ID : 302534334
majdy@post.bgu.ac.il

Project architecture

- Implemented Actors:

Client side:
1. client actor

Server side:
1. manager actor
2. channel actor
3. userServer actor (has 4 behaviors) :
    behaviors:
        1.regular user
        2.voiced user
        3.channel uperator
        4.channel Owner

- Implemented Message types:

Client Messages:
ClientPrintMessageMsg { String Text;}
ClientUserServerActorMsg { ActorRef ActRef;}
ShutDownMsg {}

Manager Messages:
CreateUserServerMsg {String UserName;}

channel messages:
GetChUsersListMsg {}
UserAsk2JoinChannelMsg{String userName; ActorRef userRef;}
LeaveUserMsg { String UserName;}
SetChannelTitleMsg { String chTitle;}
BanUserMsg { String bannedUser;}
KickUserMsg { String kickedUser;}
AddVoicedUserMsg { String UserName;}
RemoveVoicedUserMsg { String UserName;}
AddChannelOperatorMsg { String UserName;}
RemoveChannelOperatorMsg { String UserName;}
ResponsingJoinRequest { Boolean CanJoin;}
WhatIsMyModeMsg { String UserName;}
InitChannelMsg { String channelName; String channelOwner; }
GetChannelTitleMsg { }
DeleteChannelMsg { }

User messages:
TalkMsg { String Text;}
TakeChUsersListMsg {Map<String,ActorRef> UsersList;}
ClientAsk2JoinChannelMsg {String ChannelName;}
LeaveChannelMsg { String chName;}
ClientBanUserMsg { String bannedUser; String chName;}
ClientKickUserMsg { String kickedUser; String chName;}
ChangeChannelTitleMsg { String chTitle; String chName;}
ClientAddsVoicedUserMsg { String UserName; String chName;}
ClientRemovesVoicedUserMsg { String UserName; String chName;}

ClientAddsChannelOperatorMsg { String UserName; String chName;}

ClientRemovesChannelOperatorMsg { String UserName; String chName;}

DisbandMsg { String chName;}

TalkWithAnotherClientMsg {String OtherUserName; String Text;}

PrintMessageMsg { String Text;}

InitUserMsg { String nickName; Pair<String,ActorRef> DefaultCH; ActorRef myClient; ActorRef userServerRef;}

KickedMsg { String chName;}

MyModeMsg { UserServer.Mode MyMode;}

ShowAllChannelsMsg {}

ShowThisChannelUsersMsg {}

Jump2ChMsg {String chName;}

UpdateYourModeMsg {UserServer.Mode newMode;}

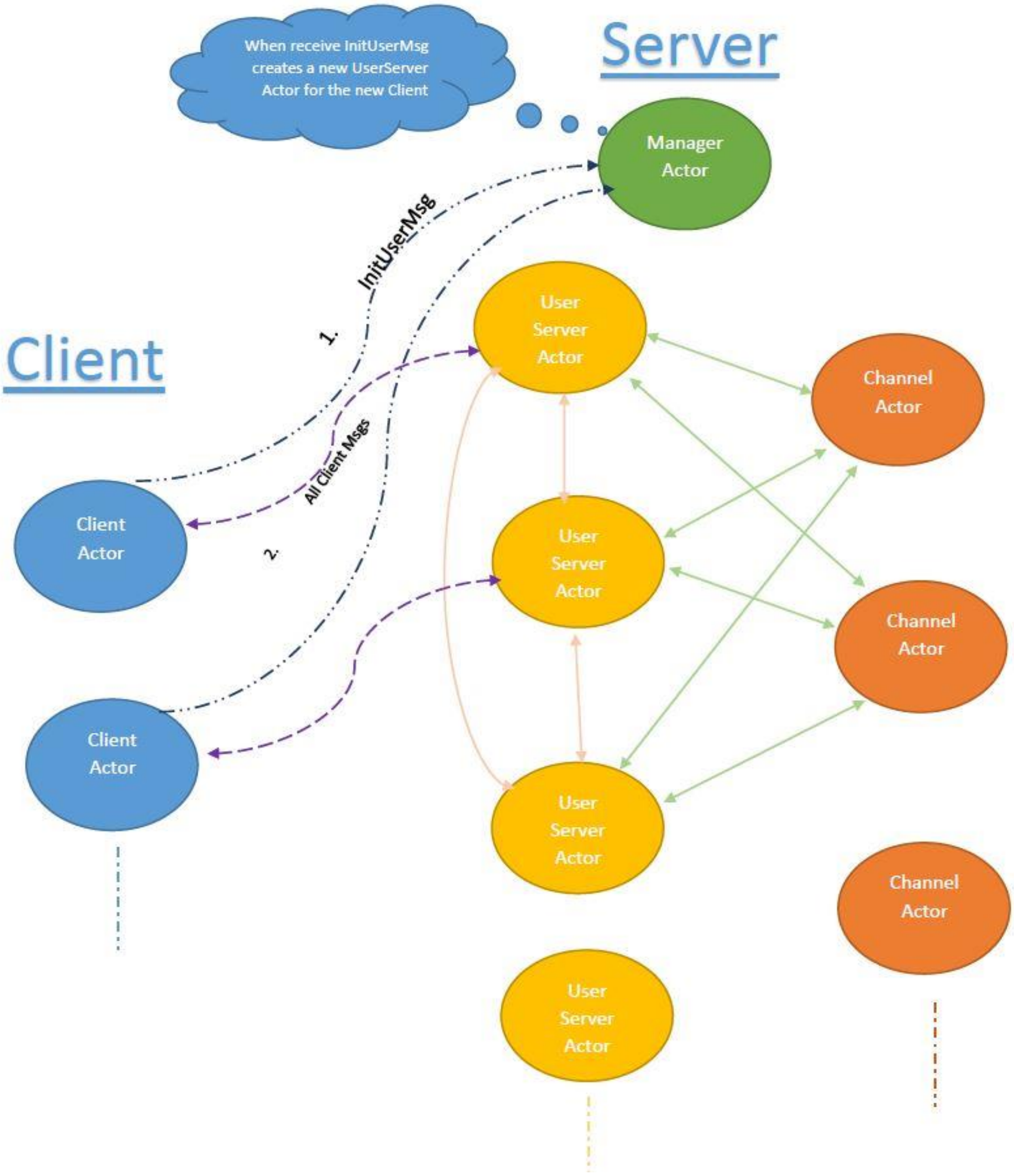
ChannelTitleMsg { String ChTitle;}

- Communication between Actors:

When ever a Client opens a client console, an local actor is created to enable communication with server side,
then this local actor sends the server (Manager Actor) InitUserMsg, which in his turn , creates a new UserServer Actor that will take care of all the communication with the client from now on.

All Actors in Server can communicate with each other , UserServer with other UserServer, UserServer with Channel and channel with UserServer.

Diagram on the next page
→


When receive InitUserMsg creates a new UserServer Actor for the new Client
Server
Manager Actor
InitUserMsg
1.
Client
User Server Actor
Channel Actor
All Client Msgs
Client Actor
2.
User Server Actor
Channel Actor
Client Actor
User Server Actor
Channel Actor
User Server Actor