

IRC Project

CS
IS



What is IRC?

- IRC (Internet Relay Chat) is a protocol for realtime Internet text-messaging (chatting).
- Mainly used for group communication in channels (like forums)
- Allows one-to-one communication (private messages)
- Allows sending files too.

Wait, what's a protocol?

- We heard of a political protocol, and a diplomatic protocol
- In networking, a protocol is a system of **digital message formats** and **rules** for exchanging those messages in or between computing systems

Popular protocols

- **SMTP** (sends emails), **IMAP** (receives emails), **FTP** (for sending and receiving files)
- A little something called **HTTP**
- IRC is like HTTP, they're both in the application layer

Seems like old news

- IRC is actually old
- And we can chat together using GTalk, MSN, Yahoo and Facebook.
- So what's new?

IRC is a **protocol**

- MSN is owned by Microsoft, Yahoo is mostly owned by Microsoft, and Facebook is partially owned by Microsoft :)
- They are all **proprietary** networks: they belong to a certain company
- We don't and can't know how they work

Seriously, IRC is a protocol

- It even has an RFC (Request for Comments) document.
- The IETF (Internet Engineering Task Force) uses RFCs as **Internet Standards**.

What's it like?

- There are channels (like forums) about a certain topic
- They look like this: #sockets, #networks
- Everything you say in a channel can be read by everyone else in that channel
- You can send private messages to users

Reminds me of...



Twitter

- Actually Twitter is based a lot on IRC
- Some analysts have called twitter back when it first launched as “IRC with a new shade of lipstick”.

DEMO

- IRC Commands
- IRC Topology
- Overview of the Phases
- Phase I: Standalone Server
- Template Code
- Delivery Plan
- Policies and Honor Code

IRC Commands

Client to Server

- A client like mIRC talks to the IRC server in a language
- This language is defined in the protocol
- It is formed of a set of commands, and a set of responses

How many clients talk to the server

Command Format

- COMMAND <arguments>
- Arguments are separated by space
- The last argument may be a sentence, and so it is preceded by a “:”
- Example:
 - USER amohamed :Ahmed Mohamed

Response Format

- In case of success:
 - OK
- In case of error:
 - ERRORCODE <other info>
 - Example:
 - 462 :You are already registered

Definitions

- Username: Your username.
- Nickname: Your nickname on the network. It is how you are identified to others.
- Hostname: The name of your machine
- Server name: The name of the IRC server you are connecting to.

Step One: Registration

- Registration to the network happens every time you log in. It is done in two steps:
 - Send a USER Command
 - Send a NICK Command
- The order doesn't matter. NICK can be before USER, they just have to BOTH exist.

USER Command

- USER username hostname servername realname
- Example:
 - USER ahmed ahmed-pc irc-server ahmed
 - USER ahmed ahmed-pc irc-server :Ahmed Aly

USER Command Errors

- ERR_NEEDMOREPARAMS = 461
- Example:
 - 461 USER :Need more params
- ERR_ALREADYREGISTERED = 462
- Example:
 - 462 :You are already registered

NICK Command

- NICK nickname
- Example:
 - NICK eagle

NICK Command Errors

- `ERR_NONICKNAMEGIVEN = 431`
 - Example:
 - `431 :No nickname given`
- `ERR_NICKNAMEINUSE = 433`
 - Example:
 - `433 eagle :Nickname already in use`

PRIVMSG Command

- Send messages to users. The target is one or more nicknames.
- PRIVMSG <target>{,<target>} :<text to be sent>
- Example:
 - PRIVMSG Angel :I'm sending a message!

Receiving PRIVMSG

- When you receive a private message, this is what you get:
- :Adam PRIVMSG :I'm sending a message!

PRIVMSG Errors

- ERR_NORECIPIENT
- ERR_NOTEXTTOSEND
- ERR_NOSUCHNICK

More Info

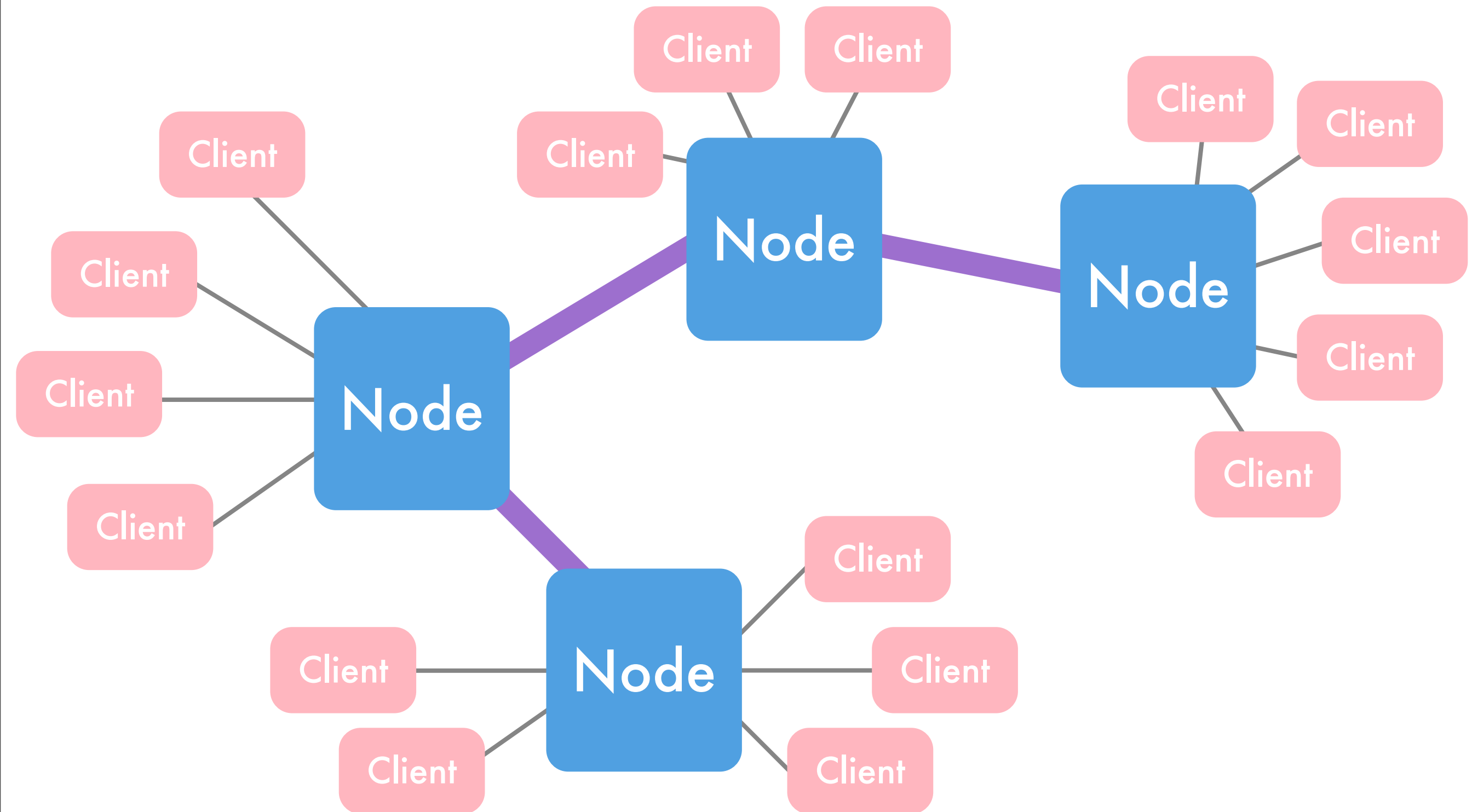
- All the commands are fully documented in the project description document

IRC Topology

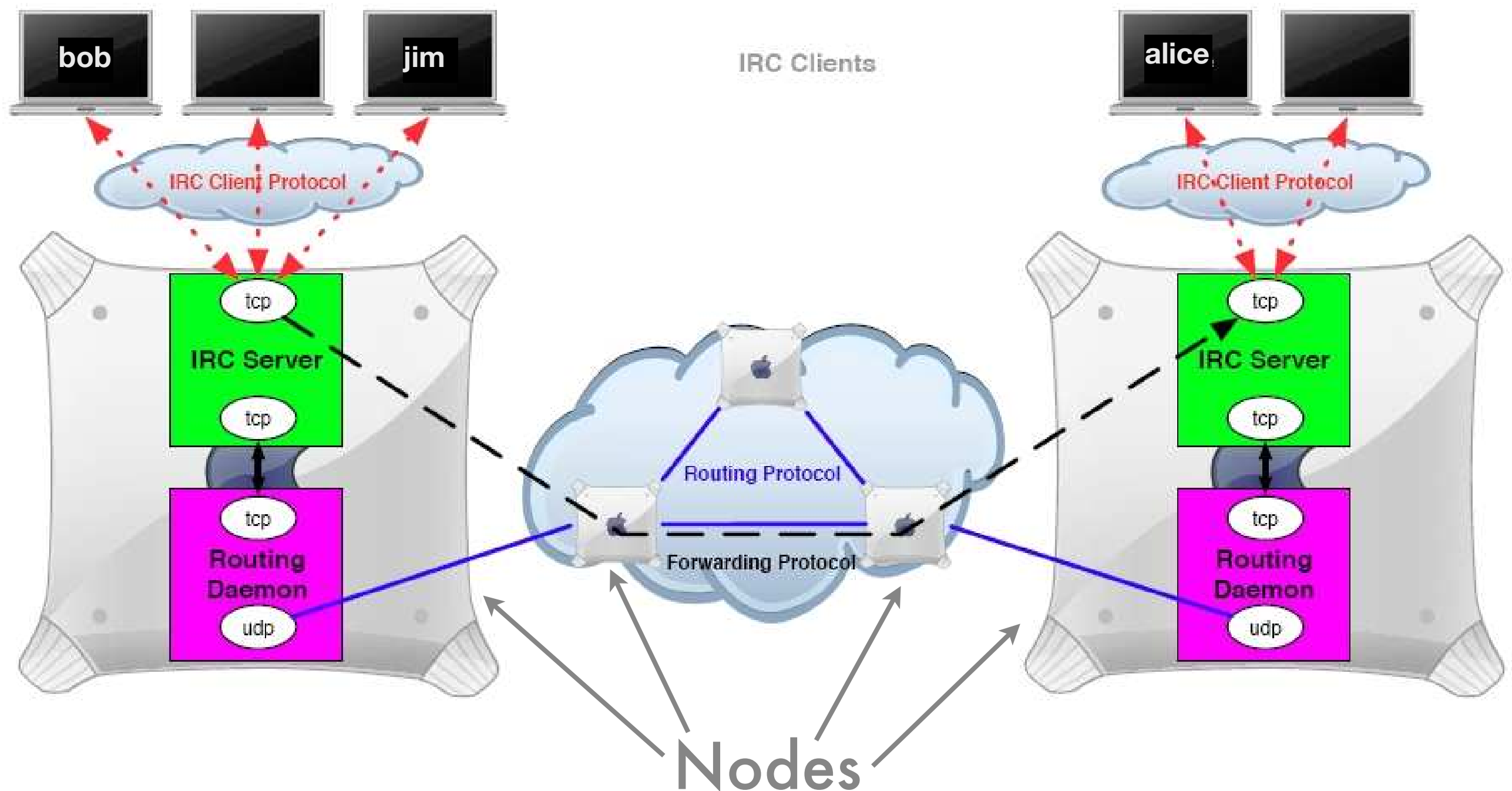
Definitions

- A client is a user like you and me
- A server is the server we are connecting to. We may connect to different servers, one in Egypt and another in the US.

IRC Network



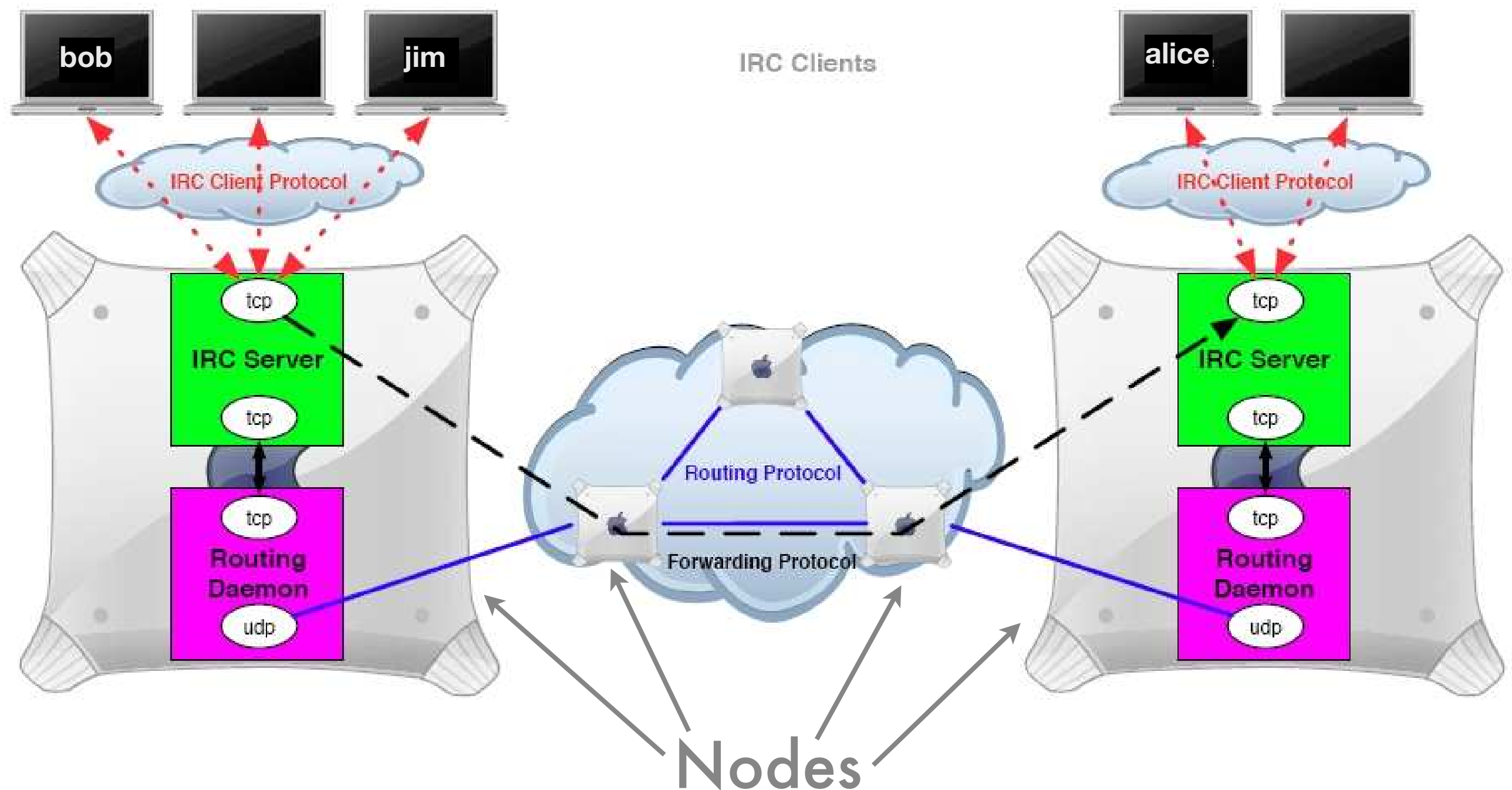
IRC Topology



Definitions

- **Node:** A single entity in the IRC network, consists of an IRC Server and a Routing Daemon
- **IRC Server:** responsible for communicating with clients and executing commands
- **Routing Daemon:** responsible for finding a user that the server couldn't find locally

PRIVMSG from bob to alice

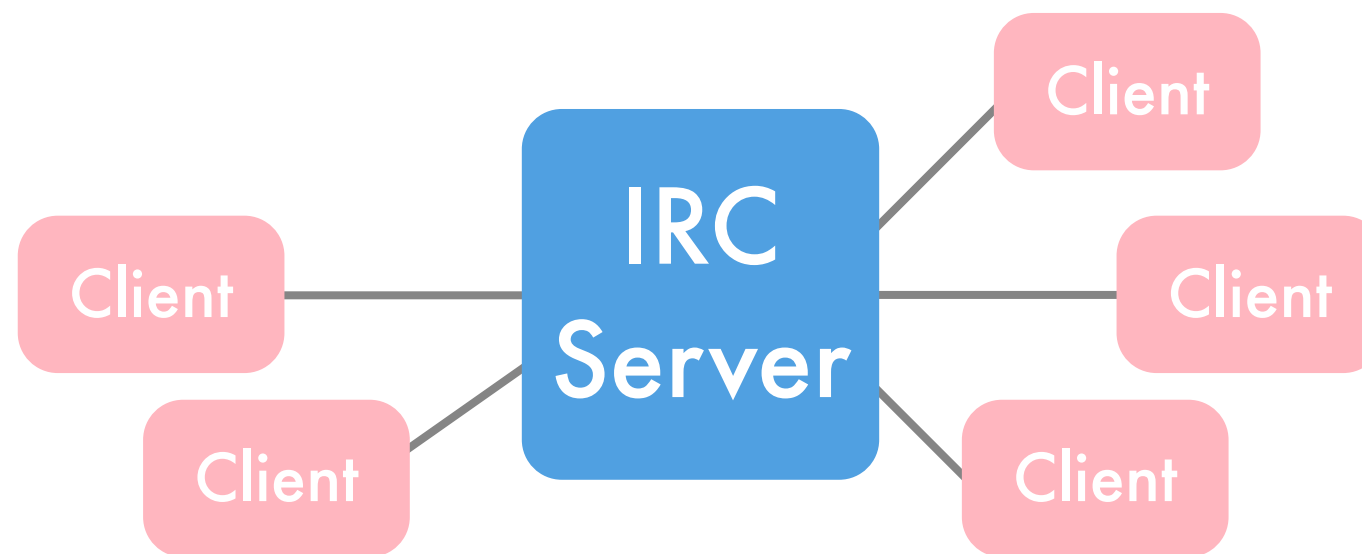


So what's the project about?

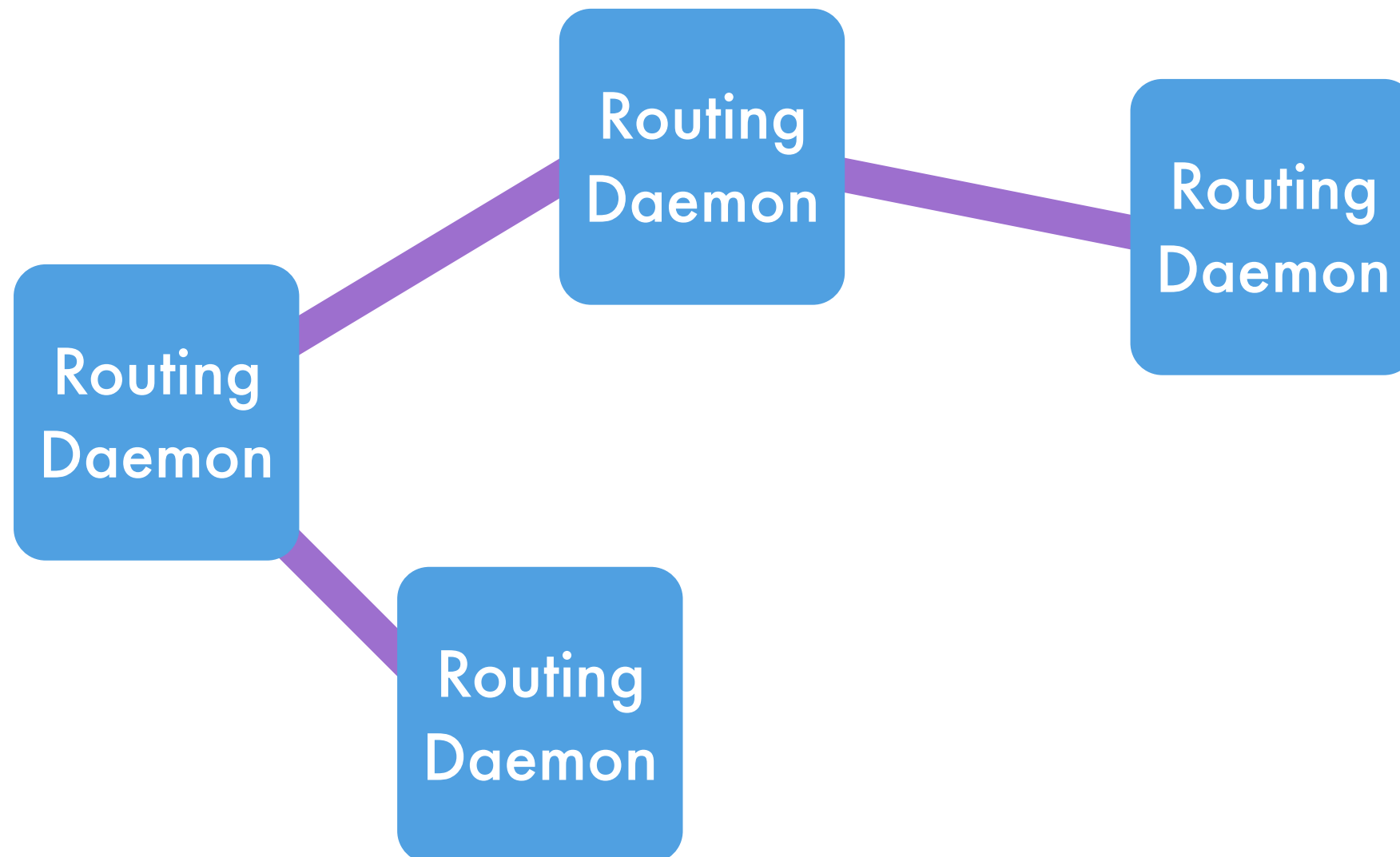
3 Phases

1..2..3

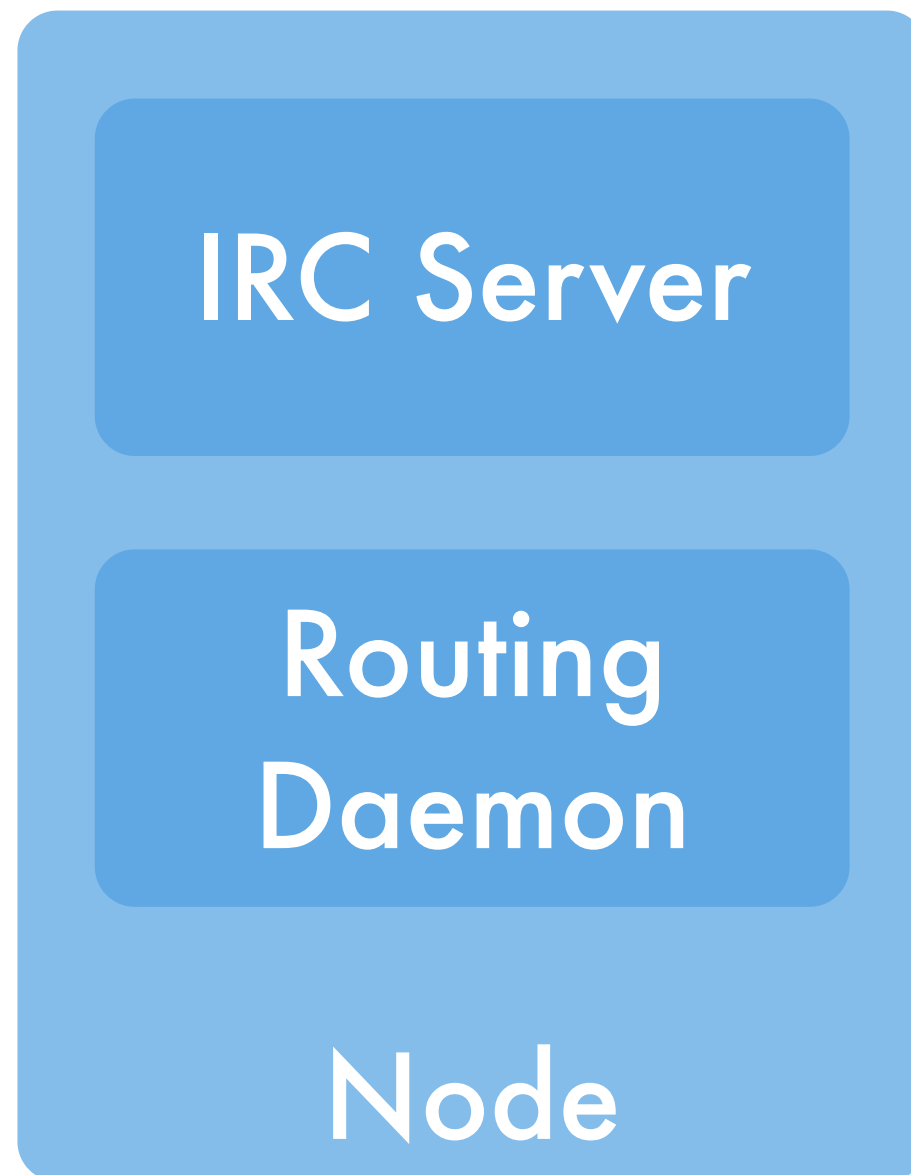
Phase 1: Standalone Server



Phase 2: Routing Daemon



Phase 3: Integration



Phase 1

Begins Today

Phase 1 Requirements

- IRC Standalone Server that accepts connection from multiple clients
- Implemented using Async sockets
- Responds to USER, NICK, QUIT, and PRIVMSG commands
- Handles errors correctly

Running the Server

- The server should be started from the command prompt as follows:
- `C:\>server.exe 9000`
- Where 9000 is the IRC port number that the server listens to.

Testing your Server

- Use the IRCClient we supply to test your server.

Template Code

Template Code

- We are providing template code to help you get a kickstart into implementing the project
- It utilizes a couple of design patterns suitable for the application
- The code is not obligatory to use

Design Patterns

Singleton Pattern

- The Singleton pattern restricts the instantiation of a class to one object. This is useful when exactly one object is needed to coordinate actions across the system.

How we use it?

```
ServerBackend.Instance.AddUser(user);
```

- Instance is a static property
- You can't do this: `new ServerBackend()`

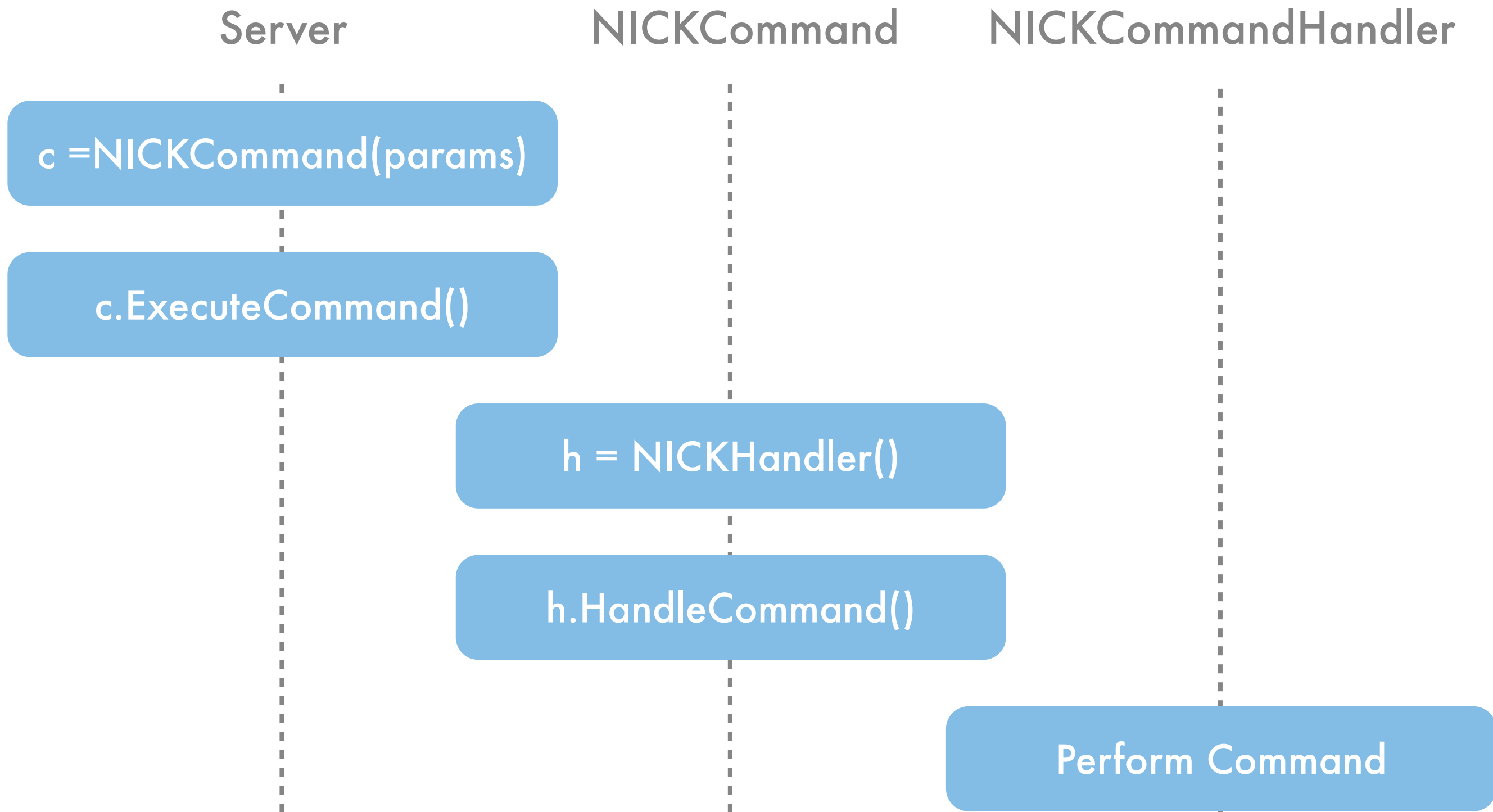
Command Pattern

- the command pattern is a design pattern in which an object is used to represent and encapsulate all the information needed to call a method at a later time.
- Suitable for networking applications

How we use it?



Sequence of Execution



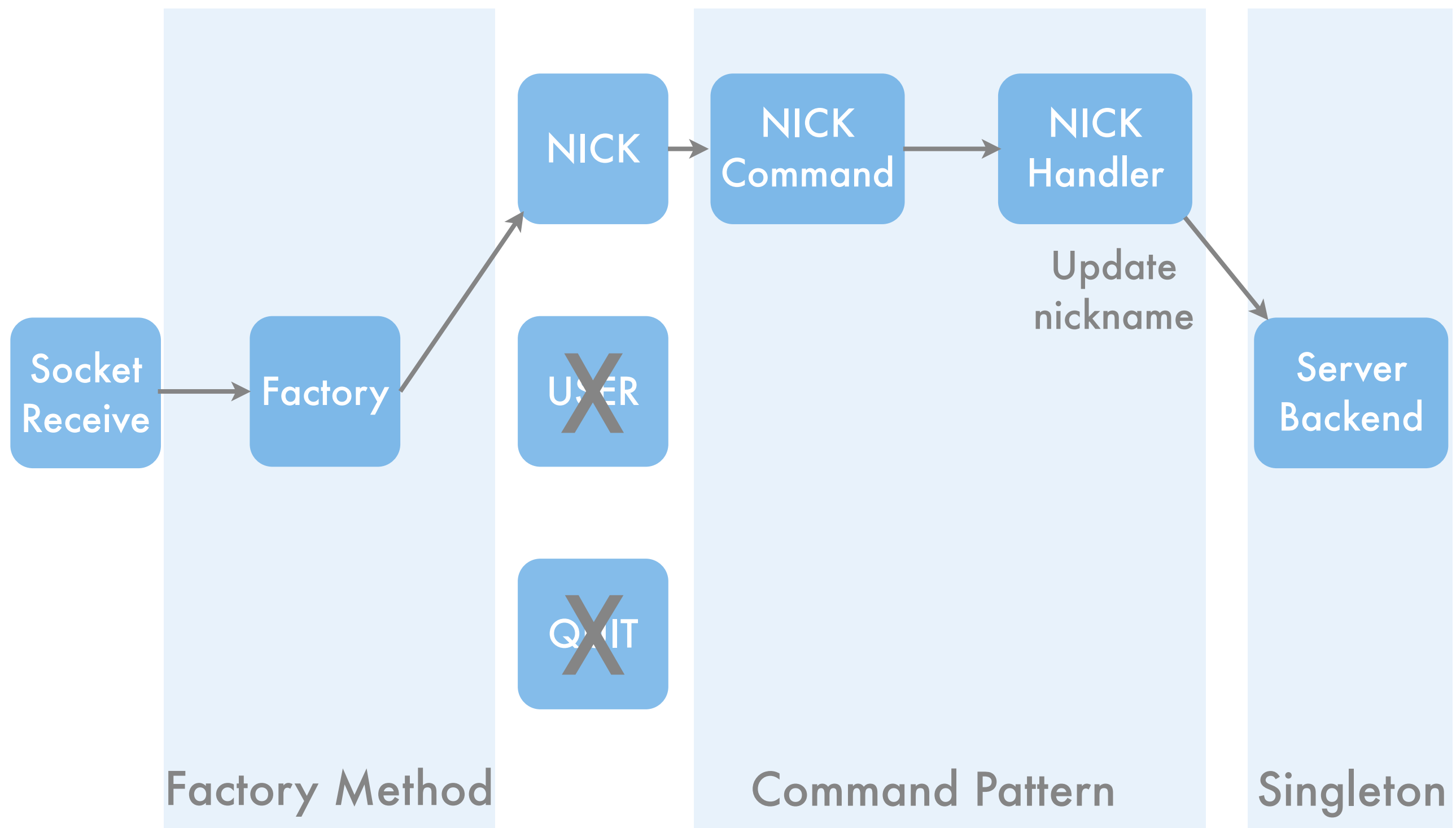
Factory Method Pattern

- The Factory Method pattern deals with the problem of creating objects (products) without specifying the exact class of object that will be created. The creation of an object often requires complex processes.

How we use it?

```
CommandBase GetCommand(message)
{
    type = GetType(message);
    if (type == NICKCommand)
        return new NICKCommand();
    ...
}
```

Example



Utilities & Helpers

- Session
- Command Parser
- Errors
- Logger (log4net)

DEMO

Grading

Phase 1	40 points
Phase 2	30 points
Phase 3	20 points
Code Style	10 points

Delivery Plan

Phase 1 Standalone Server	10 / 12 / 2011 11:59 pm
Phase 2 Routing Daemon	24 / 12 / 2011 11:59 pm
Phase 3 Integration	Practical Exam

Automatic Delivery

- Register your team and receive a secret key
- Download the test application
- Enter your secret key, the path of your server, and the port it listens to
- The application will run test cases
- It will upload the results and the code to our servers

Policies

- 5 Members **max** per team
- Every 12 hours of delay from the deadline will result in losing 10% of the grade
- You are not obliged to use the template code, yet you are required to submit well structured and readable code

Honor Code

Honor Code

- We will apply **strict** honor code policies to your work.
- Failing to adhere to the code will result in a **ZERO** in the practical grade.
- The project description document details all **prohibited** collaboration.

How will we enforce this?

- We will receive all your code (uploaded by the test tool)
- We will use a tool that compares C# code
- It catches common code, even if comments and variable names are changed.
- To “fool” it, it will be easier to right the code from scratch.

So what now?

To Do

- Register to this form before 27 / 11 (Late registration will lose you 5%)
- Read the project description document very carefully
- Download the template code

TOP PROJECT AWARD

Best Coverage - Style - Extra Mile

Finally...

- This project is something you should be proud of implementing
- CMU uses this project with its students, the difference between us and a world-class education is two things: staff + students
- We are dedicated to your world-class education, if you are dedicated to working hard
- Complain = Fail, Fight = Succeed

Thank you
Good luck