# Lecture 1

## Introduction to 5COSC022C Client-Server Architecture

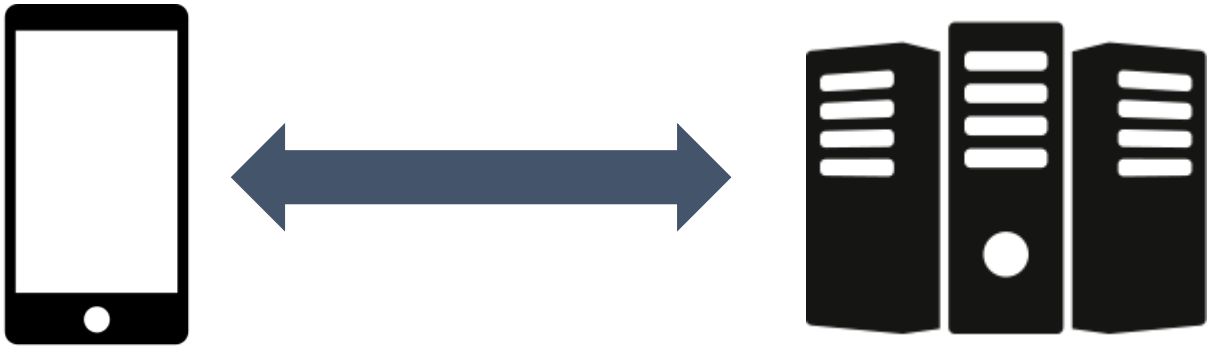
## Concepts

# Intended Learning Outcomes of Today

- Compare Local and Remote Programming
- Compare Traditional vs Client Server Programming
- Describe Client-Server Paradigm

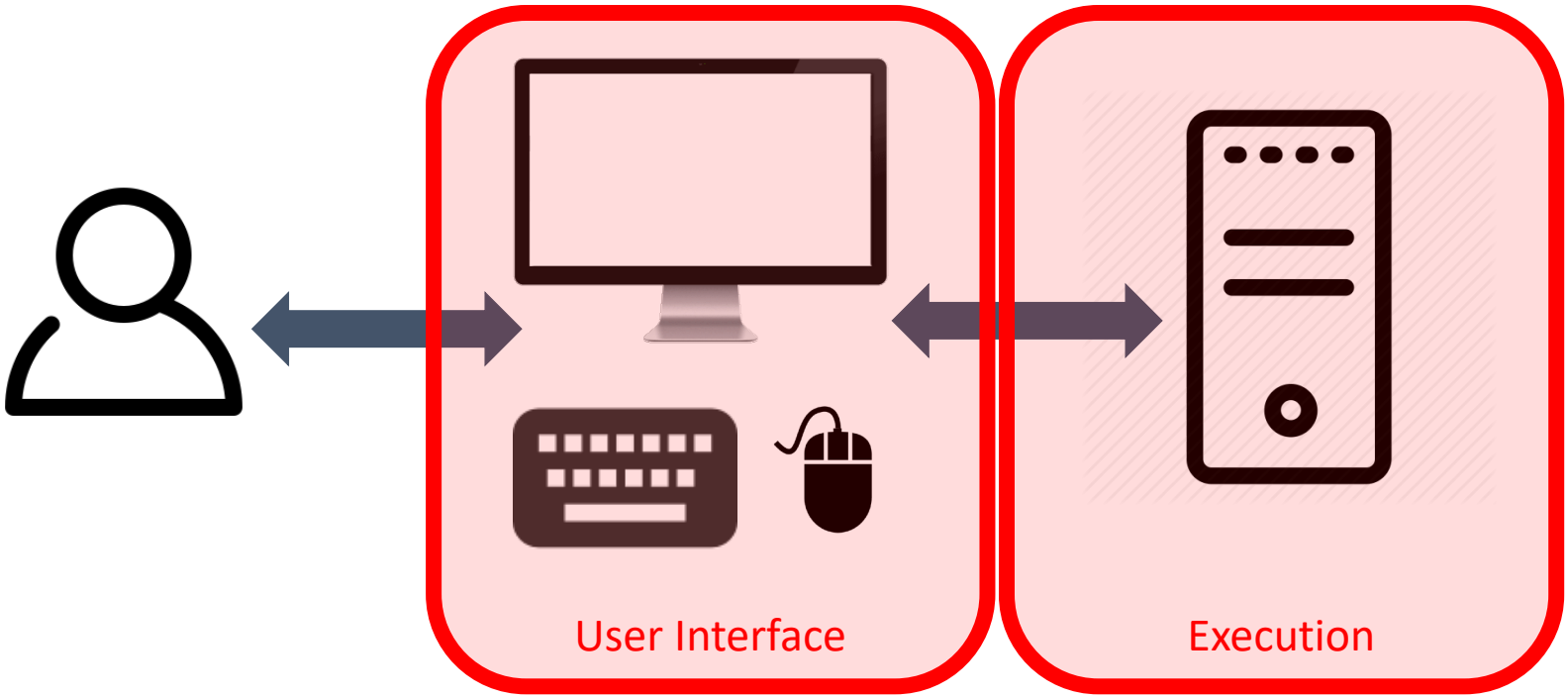
# Local and Remote Programming

- Up to know, you have only studied programs that run in your local machine but most of the programs you use in your daily life (google, Facebook, email) does not run on your local phone/laptop.



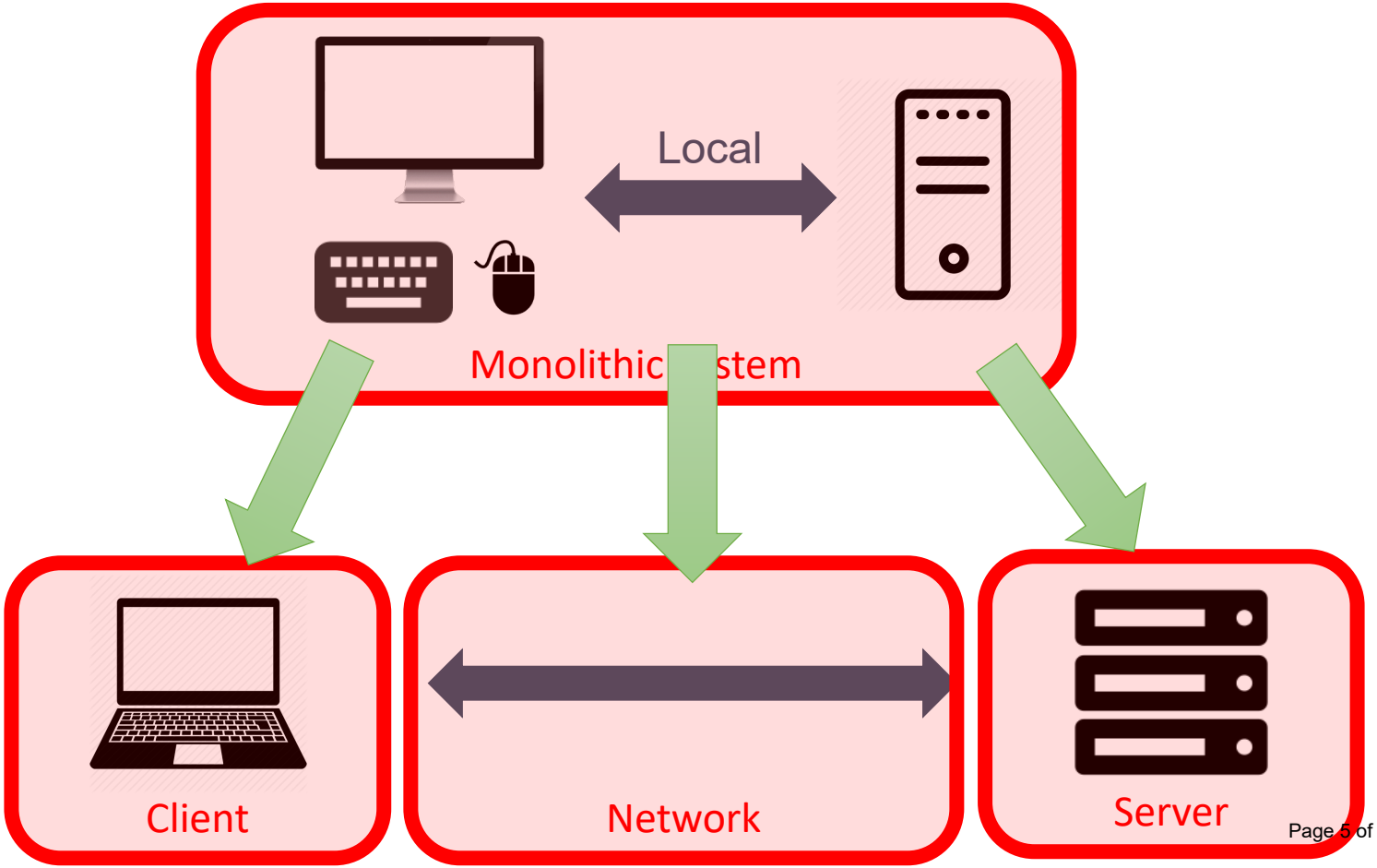
# Traditional Programming

The interactions between the user and the system and the execution of the program all happen in the same system. That is easy !



# Client-Server Programming (1)

Let's separate the user interface and the main execution of the program, i.e. two basic components of the system.

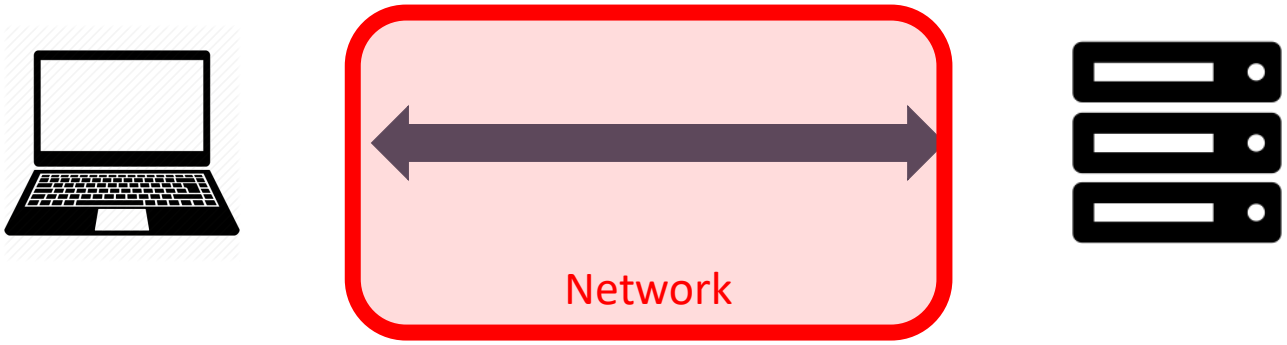


# Client-Server Programming (2)

Clients: how to program the client-side.

# Client-Server Programming (3)

Network: how to program the network-side

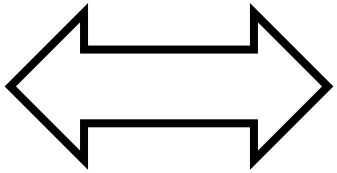


# Client-Server Programming (4)

Server: how to program the server-side.

# Client-Server Paradigm

# Client-Server Paradigm

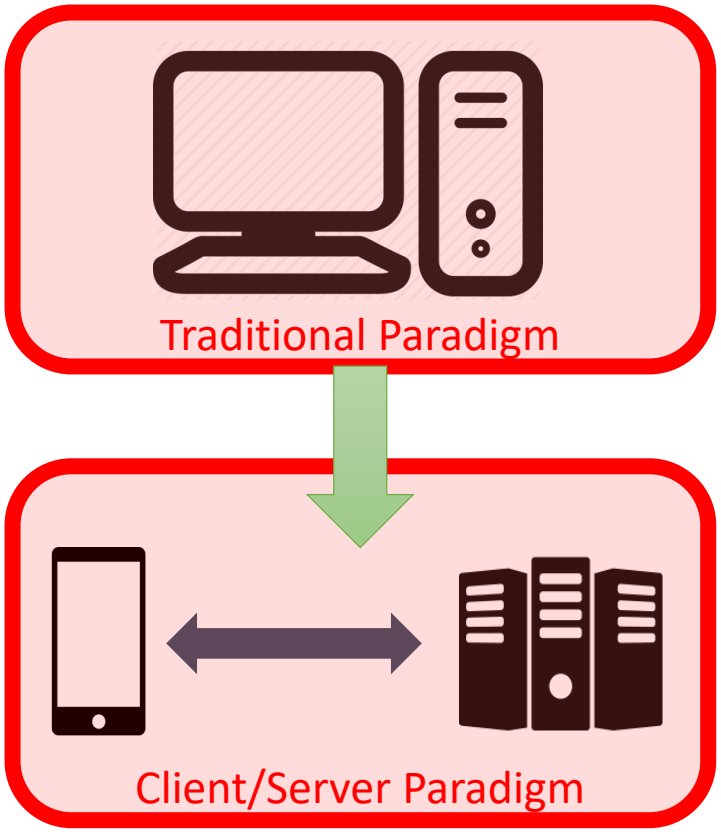


Client

Server(s)

# Client-Server Paradigm

Whenever you move a program (or part of a program) from the device it is accessed from (traditional architecture) to a client-server architecture, an entire new domain of programming opens up that covers many facets.



# Client-Server Paradigm (1)

## Definition (wiki)

- **Based on the distributed model having an architecture that partitions tasks or workloads between the providers of a resource or service (server) and service requesters (client) that communicate over computer network**
- **Server host runs one or more server programs which share their resources with clients**
- **Client does not share any of its resources, but requests a server's content or service function**
- **Server software accepts requests for a service from client software and returns the results to the client**
- **Client-server architecture separates a program from the device it is accessed from**

# Client-Server Paradigm (2)

## Components

**client:** **active initiators of transactions making requests for services**

- **user point of entry into a network**
- **normally a desktop computer, workstation, or laptop (or phone)**
- **user generally interacts directly only with the client portion software of an application**

**example: web browser**

**server:** **manages hardware and software resources to process requests**

- **servers are passive and react to client requests**

**example: web server**

**communication network:** **connects clients and servers through computer networks**

# Client-Server Paradigm (3)