MZUMBE UNIVERSITY

FACULTY OF SCIENCE AND TECHNOLOGY

COURSE NAME:	**DISTRIBUTIVE SYSTEMS**
SUBJECT CODE:	**ICT 311**
TASK:	**GROUP ASSIGNMENT**
DATE OF SUBMISSION:	**14/12/2022**
GROUP:	**Techies 5**

NAME	REGISTRATION NUMBER
AGRIPA JIMSON SANGA	14320022/T.20
SAADIA A SAIDI	13304055/T.19
SOPHIA FREDRICK	14322022/T.20
GODLOVE S. NKYA	13304023/T.19
ASMA A ALLY	13304055/T.19
TAWAJUDI B. SUED	14322050/T.20
TWAHA B. JUMANNE	14322023/T.20

QESTIONS

1. Identify types of communication in Distributed Systems

2. For each type, explain its strength and weakness

Group Assignment: Explain the concept of microservice with an example. Using the same example, determine types of interprocess communication that can be used for successful service provision

According to articles:

There are several types of communication in distributed systems. Some of the most common ones include:

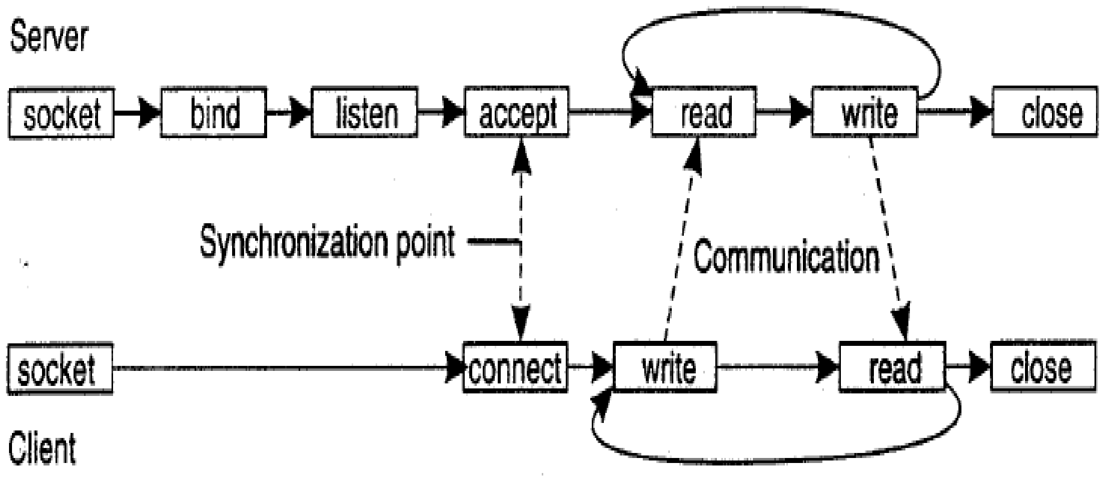
i. **Shared Memory Communication:** In this type of communication, the different components of a distributed system share a common memory space. This allows them to communicate with each other by reading and writing to the shared memory.

ii. **Message passing Communication**: In this type of communication, the components of a distributed system communicate with each other by exchanging messages. Each component has its own memory space, and communication is achieved by sending messages over a network.

Remote procedure calls and remote object invocations contribute to hiding communication in distributed systems, that is, they enhance access transparency.

Messages are usually sent and received asynchronously.

Example email system.



iii. **Remote Procedure Call (RPC) Communication**: In this type of communication, a component of a distributed system can invoke a remote procedure on another component. This is similar to calling a local function, but the procedure is executed on a different machine.

According to Birrell and Nelson suggested that the programs to call procedures located on other machines. When a process on machine *A* calls' a procedure on machine *B,* the calling process on *A* is suspended, and execution of the called procedure takes place on *B.*

- Client and Server Stubs

The idea behind RPC is to make a remote procedure call look as much as possible like a local one. In other words, we want RPC to be transparent-the calling procedure should not be aware that the called procedure is executing on a different machine or vice versa

To summarize, a remote procedure call occurs in the following steps:

1. The client procedure calls the client stub in the normal way.
2. The client stub builds a message and calls the local operating system.
3. The client's as sends the message to the remote as.
4. The remote as gives the message to the server stub.
5. The server stub unpacks the parameters and calls the server.
6. The server does the work and returns the result to the stub.
7. The server stub packs it in a message and calls its local as.
8. The server's as sends the message to the client's as.
9. The client's as gives the message to the client stub.
10. The stub unpacks the result and returns to the client.

For each, explain its strength and weakness

i. **Shared Memory Communication:**

Strength: Shared memory communication is typically fast and efficient, since the components can directly access the shared memory without the overhead of message passing or remote procedure calls.

Weakness: Shared memory communication requires that the components have a common memory space, which can be difficult to implement in some distributed systems. Additionally, shared memory communication can be vulnerable to race conditions and other concurrency issues.

ii. **Message passing Communication:**

Strength: Message passing communication allows the components of a distributed system to communicate with each other without having a common memory space. This makes it more flexible and scalable than shared memory communication.

Weakness: Message passing communication can be slower and less efficient than shared memory communication, since messages must be sent over a network and may incur additional overhead. Additionally, message passing communication can be more difficult to implement and debug.

iii. **Remote Procedure Call (RPC) Communication:**

Strength: RPC communication allows a component of a distributed system to invoke a remote procedure in a similar way to calling a local function. This can make the communication more intuitive and easier to implement.

Weakness: RPC communication can be slower and less efficient than other forms of communication, since it involves additional overhead for marshalling and unmarshalling data.

Because the calling and called procedures run on different machines, they execute in different address spaces, which causes complications.
Both the client and the server need to be online at the time of the communication.

Additionally, RPC communication can be more complex to implement and debug, especially when dealing with failure modes such as timeouts and retire

- ❖ Explain the concept of microservice with an example. Using the same example, determine types of interprocess communication that can be used for successful service provision.

Microservice is a style of developing software as a collection of independent services. Each service is running on its own process that is independent from other processes and can be deployed separately from other services **Example** e - commerce application. When a customer searches for a specific product to purchase, the product's availability needs to be confirmed in the inventory by making a request to product convenience service. Because customer should know about what the current availability of the product to place the order.

In this example, we use synchronous form of communication to know about the product's real-time availability in inventory and price information

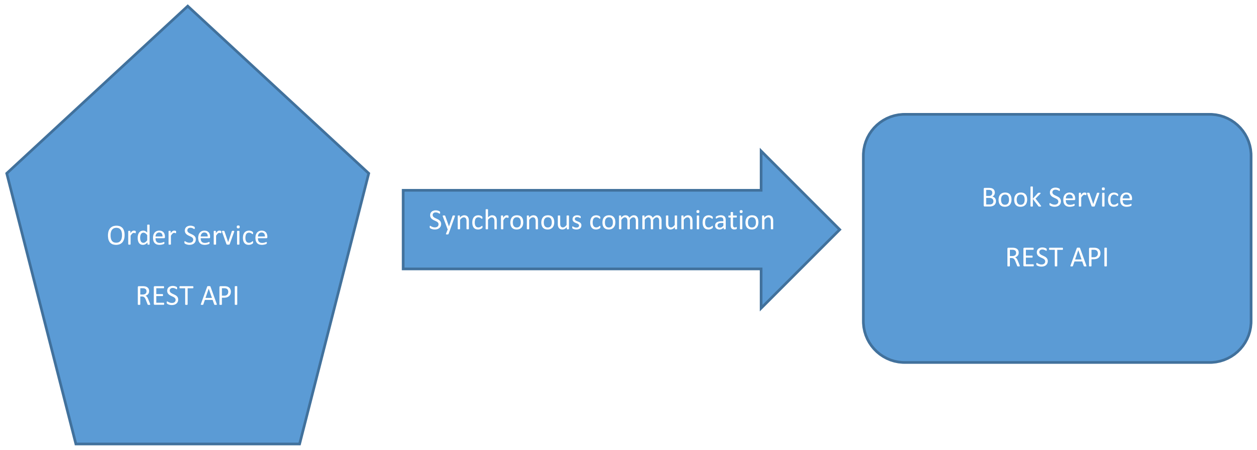


Figure: synchronous communication

Types of interprocess communication

1.Synchronous Communication

Synchronous communication is often regarded as request/response interaction style. One microservice makes a request to another service and waits for the services to process the result and send a response back. In this style, it is common that the requester blocks its operation while waiting for a response from the remote server.

2. Asynchronous Communication

The asynchronous form of communication can be implemented in microservices when services exchange messages with each other through a message broker. In this form of interaction, the message broker acts as an intermediary between services to coordinate the request and responses.

REFERENCE

J. Thönes, Microservices, IEEE software 32 (2015) 116–116.

D. Namiot, M. Sneps-Sneppe, *On micro-services architecture, International Journal of Open Information Technologies* 2 (2014) 24–27.

L. L. Peterson, N. C. Buchholz, R. D. *Schlichting, Preserving and using context information in interprocess communication*, ACM Trans. Comput. Syst. 7 (1989) 217–246. doi:10.1145/65000.65001.

S. Haselböck, R. Weinreich, G. Buchgeher, *Decision guidance models for microservices: service discovery and fault tolerance, in: Proceedings of the Fifth European Conference on the Engineering of Computer-Based System*s, 2017, pp. 1–10.