Ministry of Education of Republic of Moldova
Technical University of Moldova
CIM Faculty
Anglophone Department

Report

on PAD

Laboratory Work Nr. 1

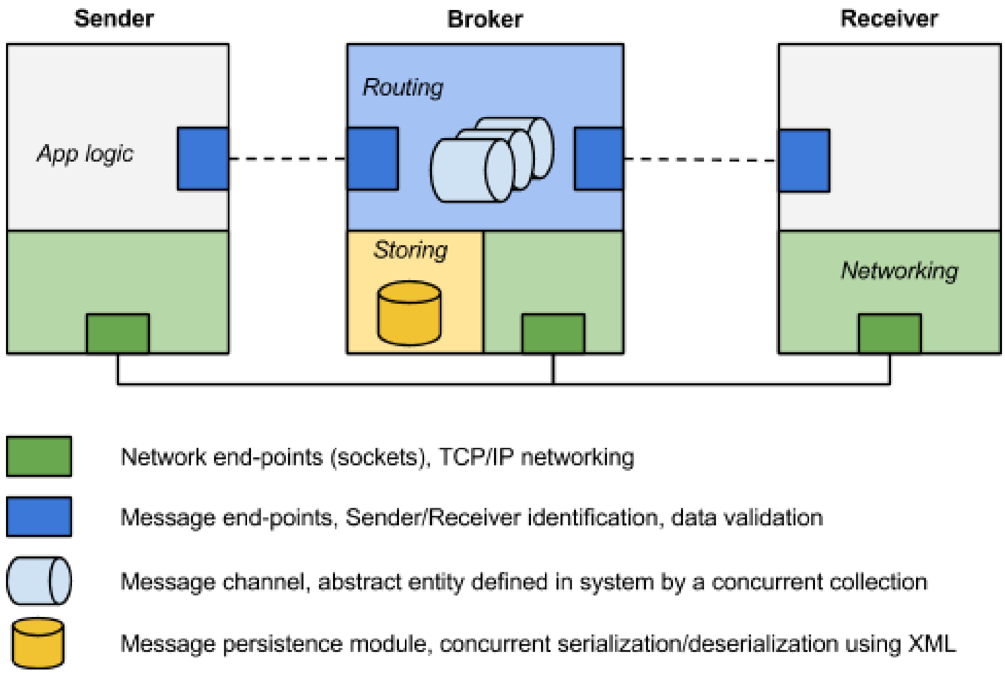
Performed by	st. gr. FAF-131 **Boldescu A.**
Verified by	lect.univ. **Ciorbă D.**

Chișinău 2016

Objective

Integration based on message brokers which allows asynchronous communication between distributed components of a system

Specific purposes


Sender
Broker
Receiver
App logic
Routing
Storing
Networking
Network end-points (sockets), TCP/IP networking
Message end-points, Sender/Receiver identification, data validation
Message channel, abstract entity defined in system by a concurrent collection
Message persistence module, concurrent serialization/deserialization using XML

1. Defining message broker protocol:
   - Format of the messages (recommended XML)
   - Number of channels (variable/fixed) depending on message type
   - Communication structure provided by agent (one- to-one ,one –two- many).
2. Develop abstract communication layer(network layer) for receiving seding messages by receiver/sender entities.
3. Coursing appropriate transport protocol
4. Handling concurrent requests.
5. Elaborating elements which will ensure keeping messages in memory:
   - Transient method: messages will be stored in data collections specific for used language.

- Persistent method: messages will be serialized / desterilized using processing methods of concurrent or asynchronous processing.

General notion

Message broker - is a physical component that manages communication between the components of a distributed application. The advantage of this technique is the decoupling sender from the receiver of messages. Therefore an application participating agent only sends messages indicating a logical name of the receiver.

The broker can expose various application interfaces together and can transfer messages between them, without requiring a common interface to all the participants in order to ensure interaction. Responsibilities and collaborations essential to a message broker are shown in the table below:

- Transient method: messages will be stored in data collections specific for used language.

Table 1. Responsibilities/Collaborations

Responsibilities	Collaborations
Receiving messages	Apps that send messages to agent
Finding destination and routing	Apps which receive message from broker(receivers)
Treating differences between interfaces	
Sending messages	

Decision of using broker for interrogation of application balances between flexibility received by decupling the participants and the effort to maintain the broker.

Advantages:

- **Reduce coupling** - only senders communicate with the broker, so a group of several potential recipients under a common logical name may become clear sender;
- **Larger inerrability** - applications that communicate with the broker must have the same interface, the broker can become a bridge between applications with different levels of security and quality of service QoS;
- **Better evolution** - the broker protects the components from individual modifications of integrated applications, often giving the dynamic configuration capabilities;

Disadvantages

Increases complexity - broker communicating with all participants must implement multiple interfaces (protocols) and better use multithreading from performance perspective;

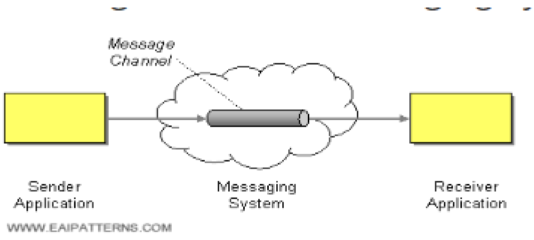
Increase the effort for maintenance - all participants must be registered by the broker and ask for a mechanism to identify them;

Reduced availability - one component that mediates communication is the single point of failure (single point of failure - eng.), Failure involves blocking the activity of the whole system; This problem is remedied by doubling the broker and synchronization primary and secondary agent states;

Reduces performance - messaging agent additionally adds a step that involves overhead .

Problem description:

Message Channel (Messaging Systems)

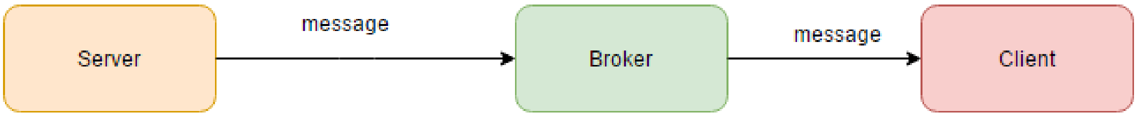


Message Channel is a logical element used to interconnect applications. An application writes messages in the channel and another reads. Therefore this method of intermediation is one primary and message queue is implementing its shape.

Protocol description:

- Format of the messages (plain text)
- Number of channels –one channel
- Communication structure provided by agent -one- to-one
- Transport protocol - TCP
- Transient method for storing messages (lists).
- Sender/broker/receiver code separated from one another in different files.

Composition Diagram



Server sends message to broker it receives it and send it to respective client.

It was used a BSD sockets communication approach based on simple client/ server app in python.

What are Sockets?

Sockets are the endpoints of a bidirectional communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents.

Python code reference:

Conclusion

In this laboratory work I studied the basics of python networking principles and learned how to build a simple message broker application based on BSD sockets and how to work with sockets.