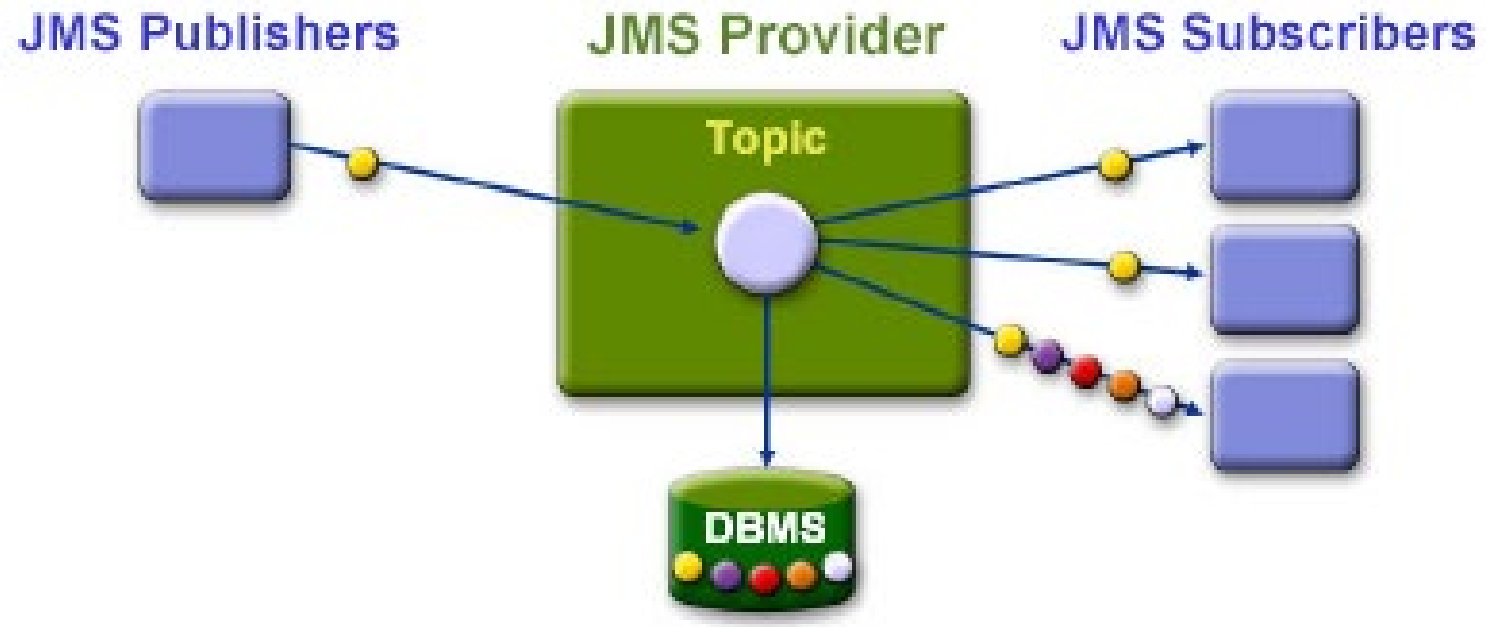
Instant Messaging service

Juan Luis Gorricho

Publisher/Subscriber Pattern



- Messages are classified by topics
- Publishers send messages by topic
- Subscribers register to specific topics

First approach

- This is a first approach implementing an instant messaging service.
- This approach focuses on defining interfaces and their implementations to manage the usual data of an instant messaging service: TopicManager, Publisher and Subscriber.
- The approach uses Swing clients instead of Internet browsers, and it doesn't consider any persistence at the server or the client.

Interfaces

```
public interface Publisher {

   void        incPublishers();
   int         decPublishers();
   void        attachSubscriber(Subscriber subscriber);
   boolean     detachSubscriber(Subscriber subscriber);
   void        detachAllSubscribers();

   void        publish(Message message);
}

public interface Subscriber {

    void onMessage(Message message);
    void onClose(Subscription_close subs_close);
}
```

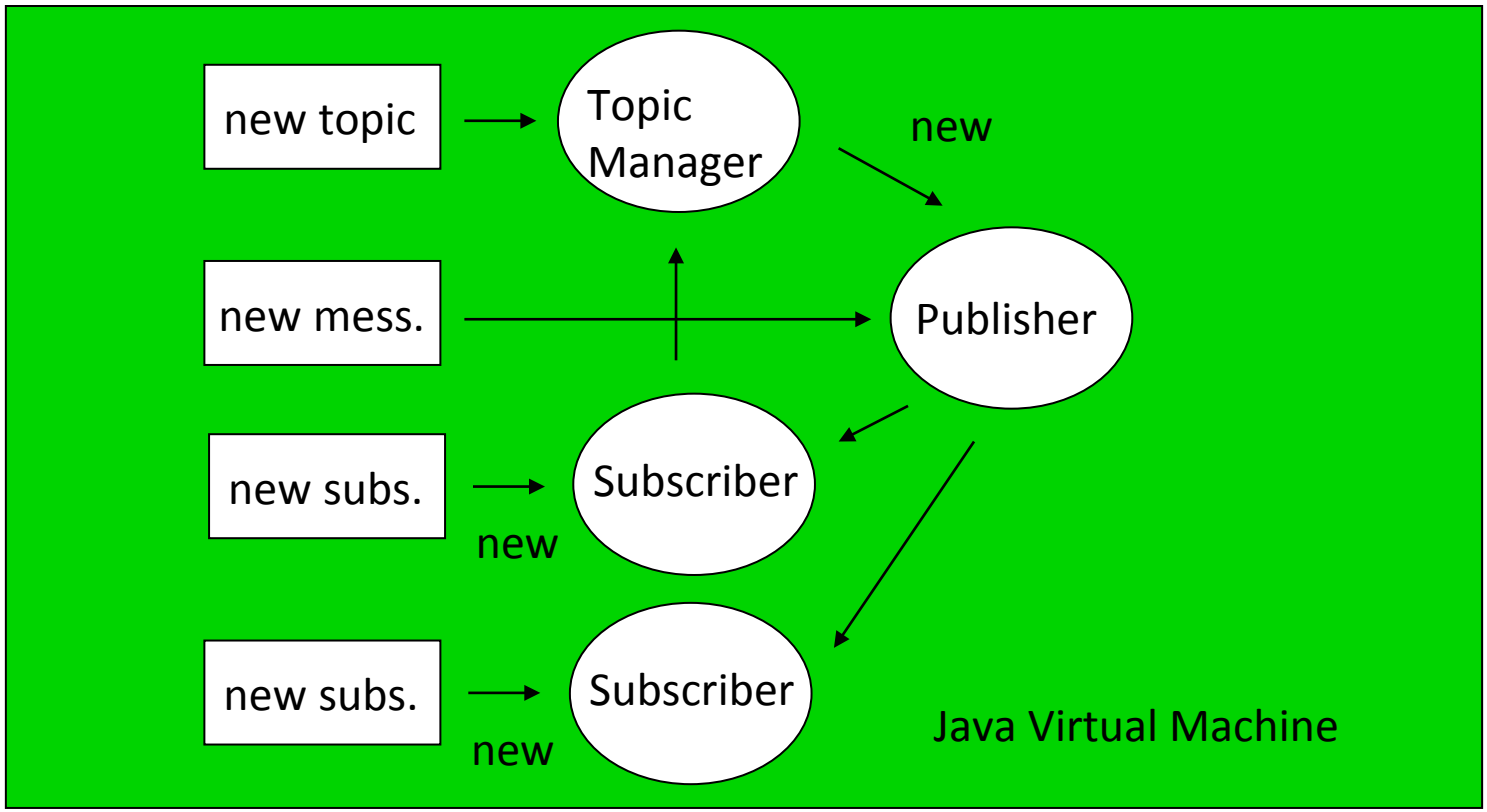
Interfaces

```
public interface TopicManager {

  Publisher     addPublisherToTopic(Topic topic);
  void          removePublisherFromTopic(Topic topic);
  Topic_check   isTopic(Topic topic);
  List<Topic>   topics();

  Subscription_check   subscribe(Topic t, Subscriber s);
  Subscription_check   unsubscribe(Topic t, Subscriber s);
}
```

Schema of Classes



Packets

- publisher:
  - Publisher
  - PublisherImpl
- subscriber:
  - Subscriber
  - SubscriberImpl
- topicmanager:
  - TopicManager
  - TopicManagerImpl

Packets

- entity:
  - Topic
  - Topic_check
  - Message
  - Subscription_check
  - Subscription_close
- main:
  - SwingClient
  - The_system

Exercise: local version

- Complete the code for the implementation of the classes: PublisherImpl and TopicManagerImpl.
- Review the coding of the SwingClient java class, try to understand the use of the swing library.
- Complete the code of all the ActionListeners for all buttons of the SwingClient.