Distributed Systems

Practicum introduction

MOSAIC
Modeling Of Systems And Internet Communication
University of Antwerp

Practical considerations

Coordinates

- Tom De Schepper – M.G.215
- tom.deschepper@uantwerpen.be

Practicum

- 1 project assignment
- Groups of 2
- 1 final deadline before the start of the exams

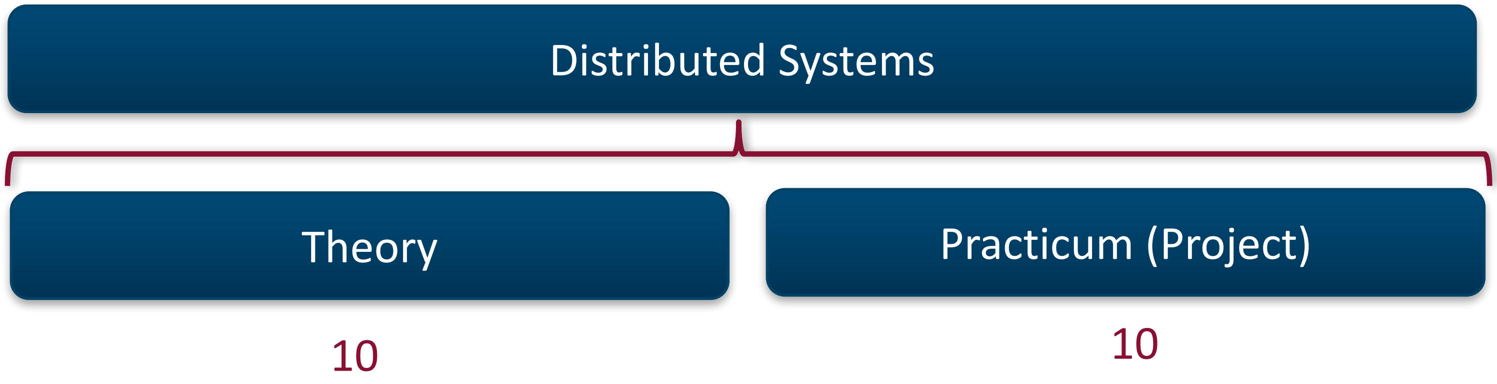
Schedule

Date		Topic
13/10	09u00 – 12u00	Project assignment
20/10	09u00 – 12u00	Q&A project
17/11	09u00 – 12u00	Q&A project
8/12	09u00 – 12u00	Q&A project
22/12	09u00 – 12u00	Q&A project
8/01	23u59	Deadline: Project

Practical considerations

Grades

- 10 points (out of 20)
- Need 50 % for both theory and project



Programming

Tools

- Java as programming language
- Eclipse
- Avro library
- Other libraries are allowed

Programming

Java vs C++

- Compiliation:
  - C++: to system-level bytecode, runs natively
  - Java: to JVM bytecode, runs on Java Virtual Machine
- Error behaviour java is defined
  - E.g., out-of-bound-index, ….
- Similar language synstax
- Object Oriented
- Memory management
  - auto garbage collection, no explicit pointers, …

Programming

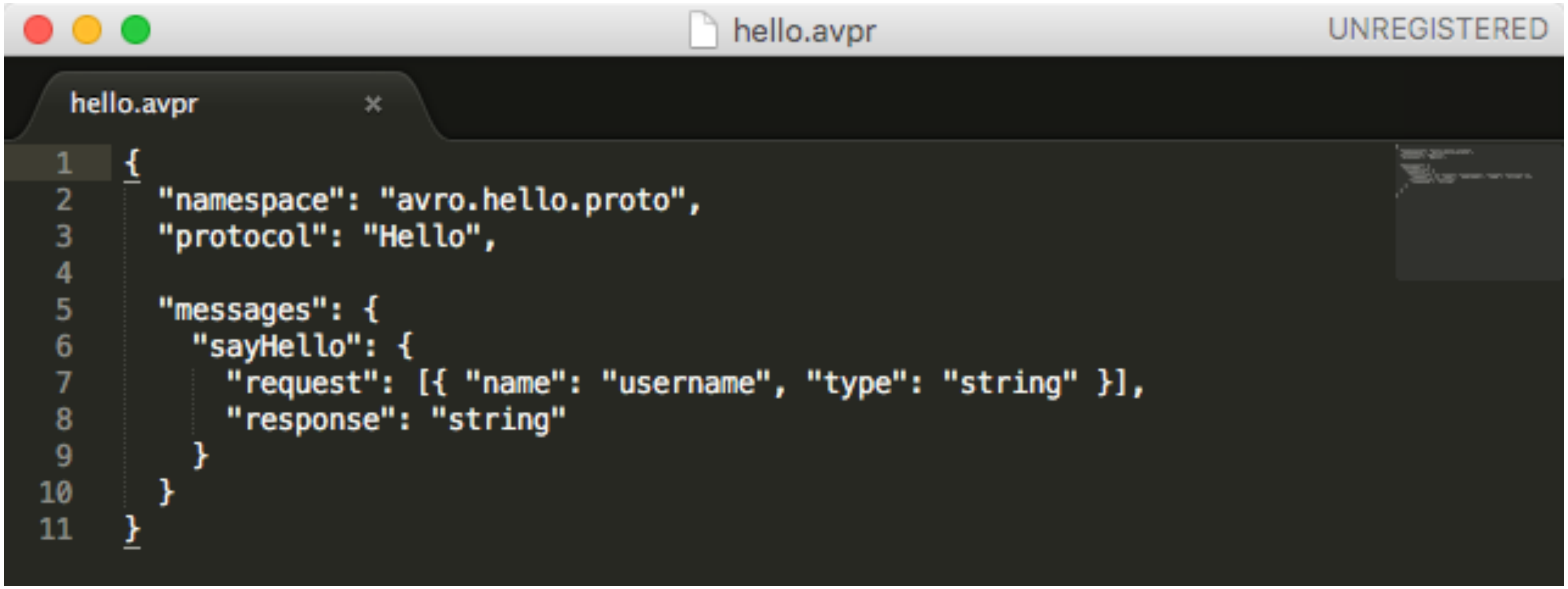
Java vs C++

- Java is always passed-by-value
  - Primitive types and object references
- Other
  - No operation overloading
  - Final instead of const
  - I/O with streams
  - Interface
  - Java serialization
  - …

Apache Avro

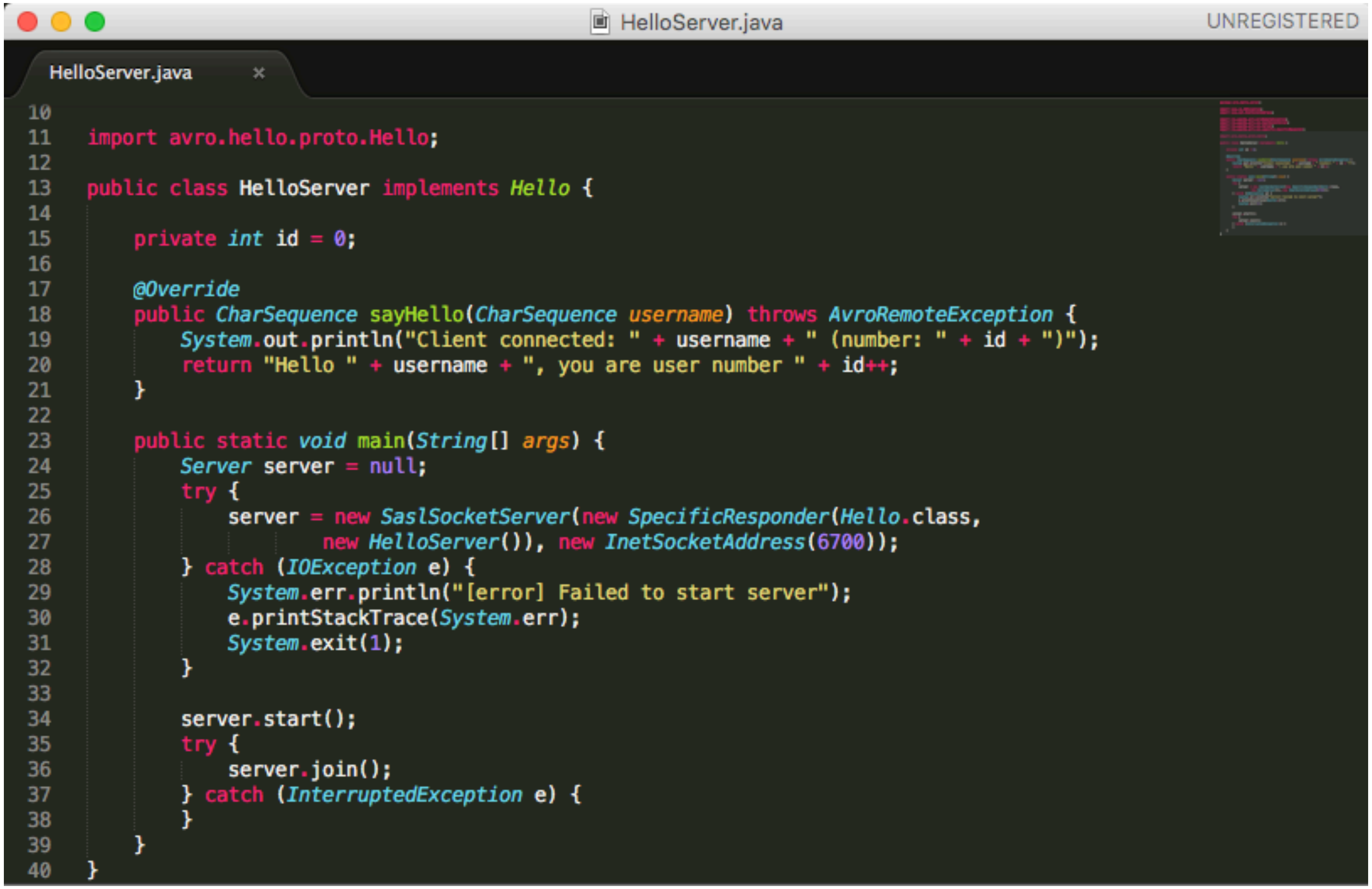
- Remote procedure call and data serialization framework
- Newer than Java RMI
- Schema based (JSON)

Schema



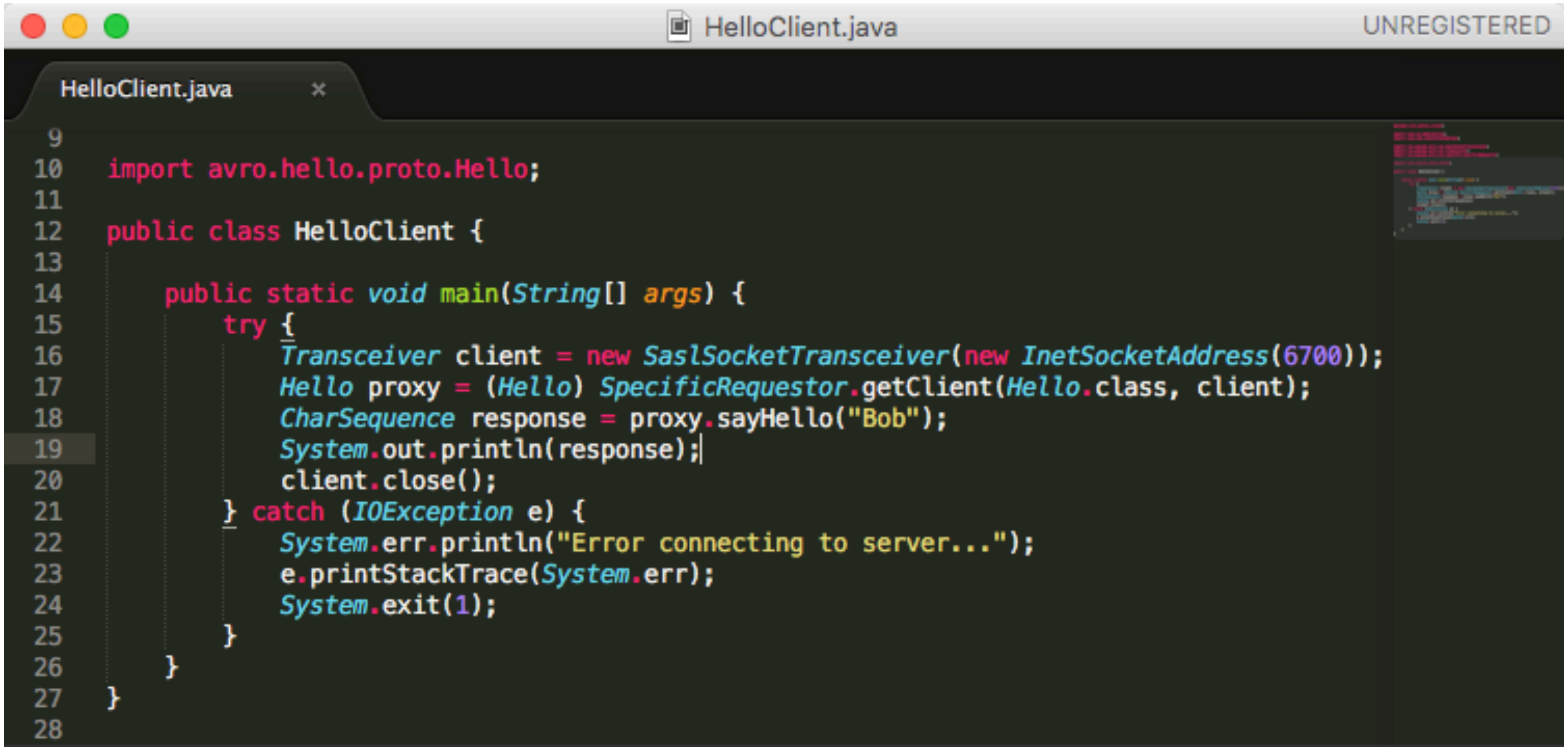
```
{
  "namespace": "avro.hello.proto",
  "protocol": "Hello",

  "messages": {
    "sayHello": {
      "request": [{ "name": "username", "type": "string" }],
      "response": "string"
    }
  }
}
```

Server



```java

import avro.hello.proto.Hello;

public class HelloServer implements Hello {

    private int id = 0;

    @Override
    public CharSequence sayHello(CharSequence username) throws AvroRemoteException {
        System.out.println("Client connected: " + username + " (number: " + id + ")");
        return "Hello " + username + ", you are user number " + id++;
    }

    public static void main(String[] args) {
        Server server = null;
        try {
            server = new SaslSocketServer(new SpecificResponder(Hello.class,
                    new HelloServer()), new InetSocketAddress(6700));
        } catch (IOException e) {
            System.err.println("[error] Failed to start server");
            e.printStackTrace(System.err);
            System.exit(1);
        }

        server.start();
        try {
            server.join();
        } catch (InterruptedException e) {
        }
    }
}
```

Client



HelloClient.java UNREGISTERED

HelloClient.java ×

```java

import avro.hello.proto.Hello;

public class HelloClient {

    public static void main(String[] args) {
        try {
            Transceiver client = new SaslSocketTransceiver(new InetSocketAddress(6700));
            Hello proxy = (Hello) SpecificRequestor.getClient(Hello.class, client);
            CharSequence response = proxy.sayHello("Bob");
            System.out.println(response);
            client.close();
        } catch (IOException e) {
            System.err.println("Error connecting to server...");
            e.printStackTrace(System.err);
            System.exit(1);
        }
    }
}

```