Java

Client-server communication. WebSockets

Александр Помосов

Reminder

ТЕХНОАТОМ

Отметьтесь на портале

Reminder

ТЕХНОАТОМ

Обновите репозиторий

Agenda

TEXHOATOM

- IO/NIO
- Game server architecture
- Web sockets
- Client-server communication

Agenda

TEXHOATOM

- IO/NIO
- Game server architecture
- Web sockets
- Client-server communication

IO

TEXHOATOM

http://docs.oracle.com/javase/tutorial/essential/io/

API for input and output to

- files
- network streams
- internal memory buffers
- …

Blocking

IO. Byte streams

TEXHOATOM

- InputStream
- OutputStream

```
@see ru.atom.io.ByteStreams.java

look at System.out / System.err
```

IO. Character streams

TEXHOATOM

- Reader
- Writer
- Scanner

```
@see ru.atom.io.CharacterStreams.java
```

NIO

TEXHOATOM

Non-blocking IO

IO. File operations

TEXHOATOM

java.nio.file
Modern file API

```
@see ru.atom.nio.Files.java
```

Agenda

TEXHOATOM

- IO/NIO
- Game server architecture
- Web sockets
- Client-server communication

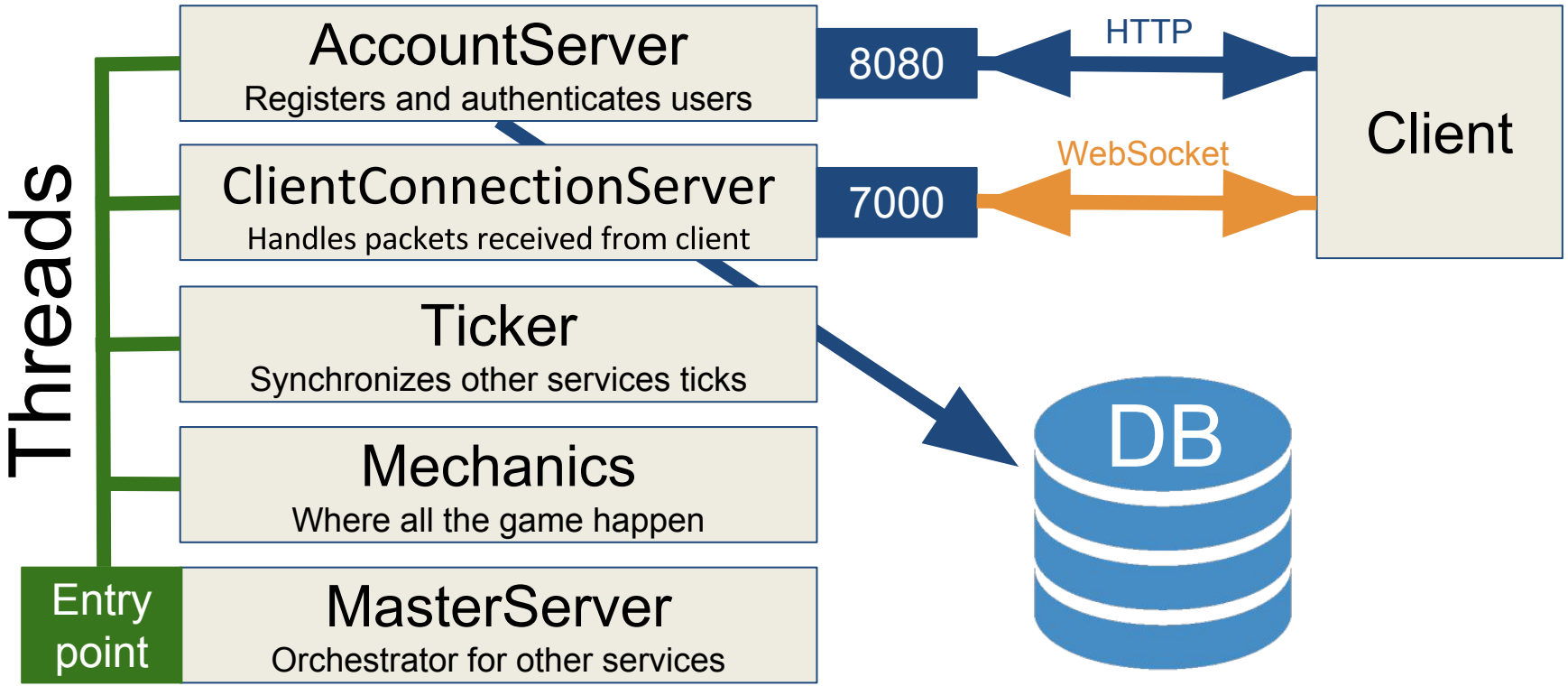
Game server architecture

TEXHOATOM

```
@see project server
```

Services overview

TEXHOATOM



ApplicationContext

TEXHOATOM

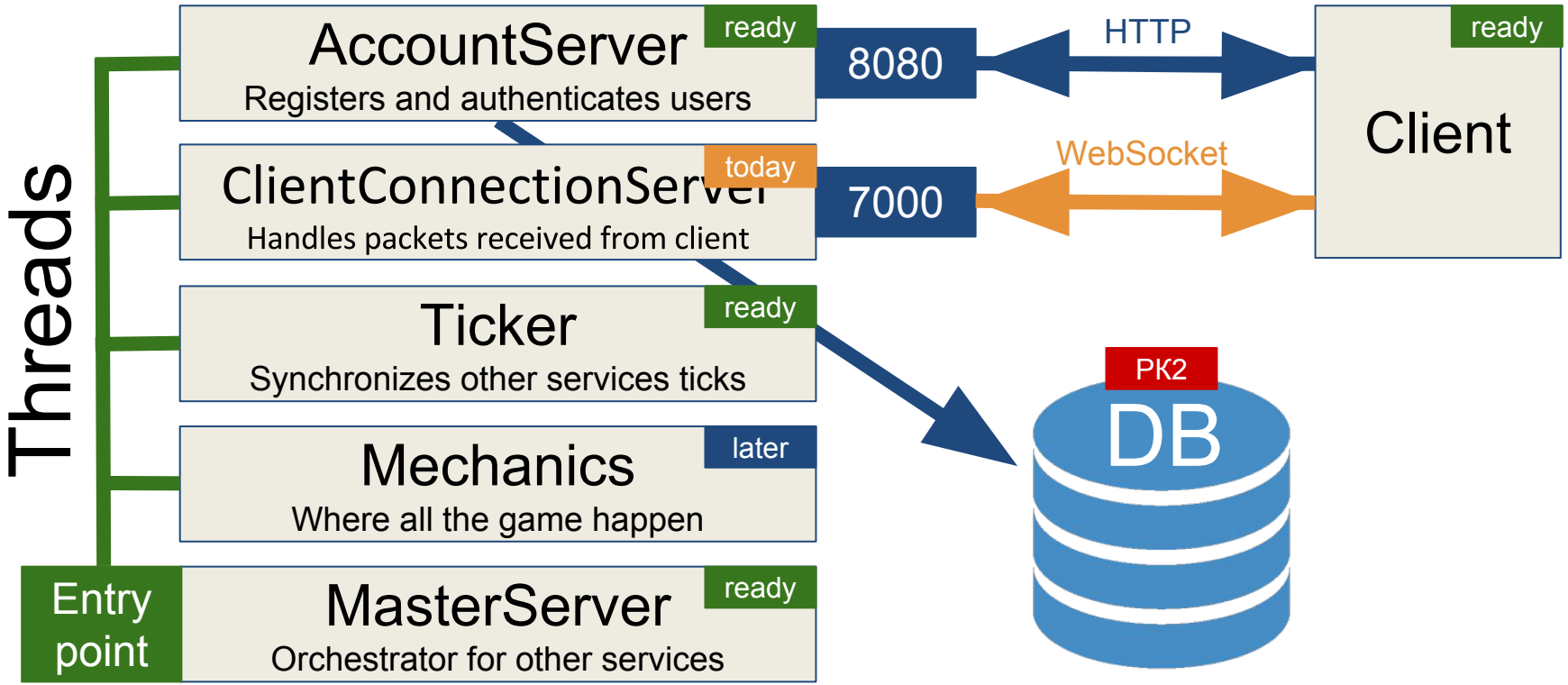
How do we manage dependencies in complex project?

- spaghetti-style
- public static fields and Singletons
- Dependency Injection
- contexts (instance containers)

```
@see ApplicationContext
```

What to do?

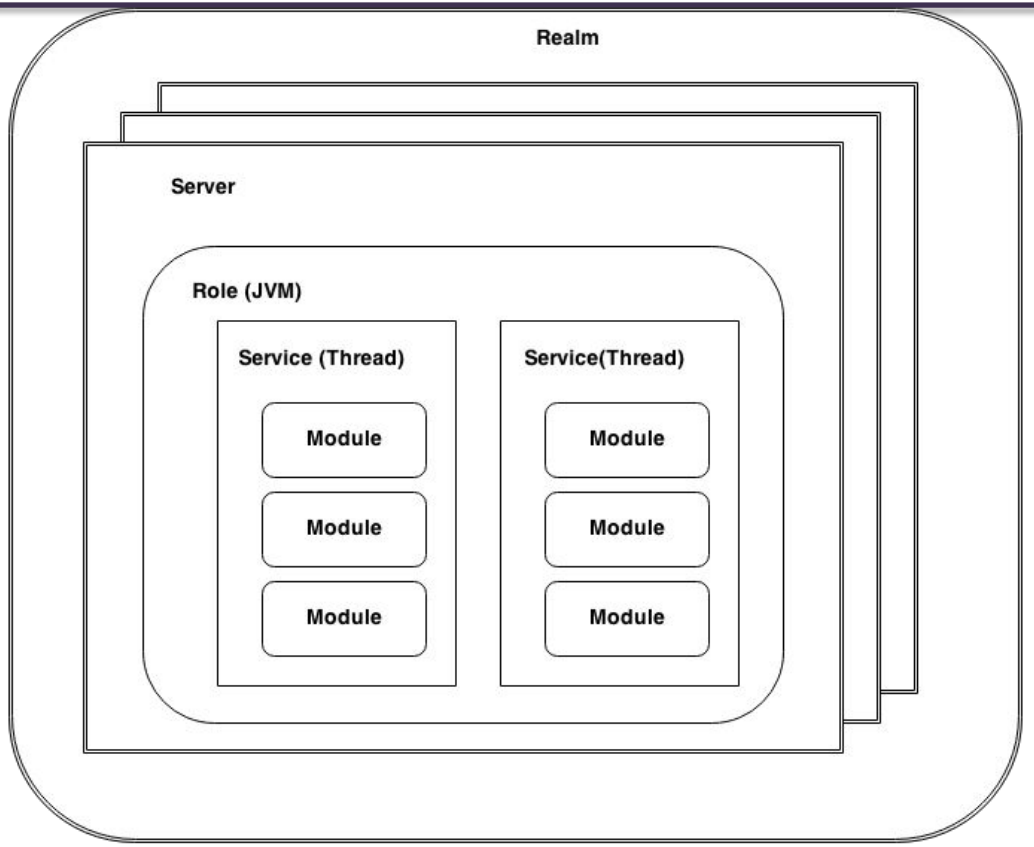
TEXHOATOM



Real Game

TEXHOATOM

Does monolitic mean **bad**?



https://habrahabr.ru/company/mailru/blog/220359/

Agenda

TEXHOATOM

- IO/NIO
- Game server architecture
- Web sockets
- Client-server communication

WebSocket

TEXHOATOM

- client-server
- application-layer
- over single TCP connection
- full-duplex
- protocol (ws:// wss://)

Standardized in 2011
https://tools.ietf.org/html/rfc6455
Supported by most modern browsers
but can be used for any extensive client-server communication

WebSocket

TEXHOATOM

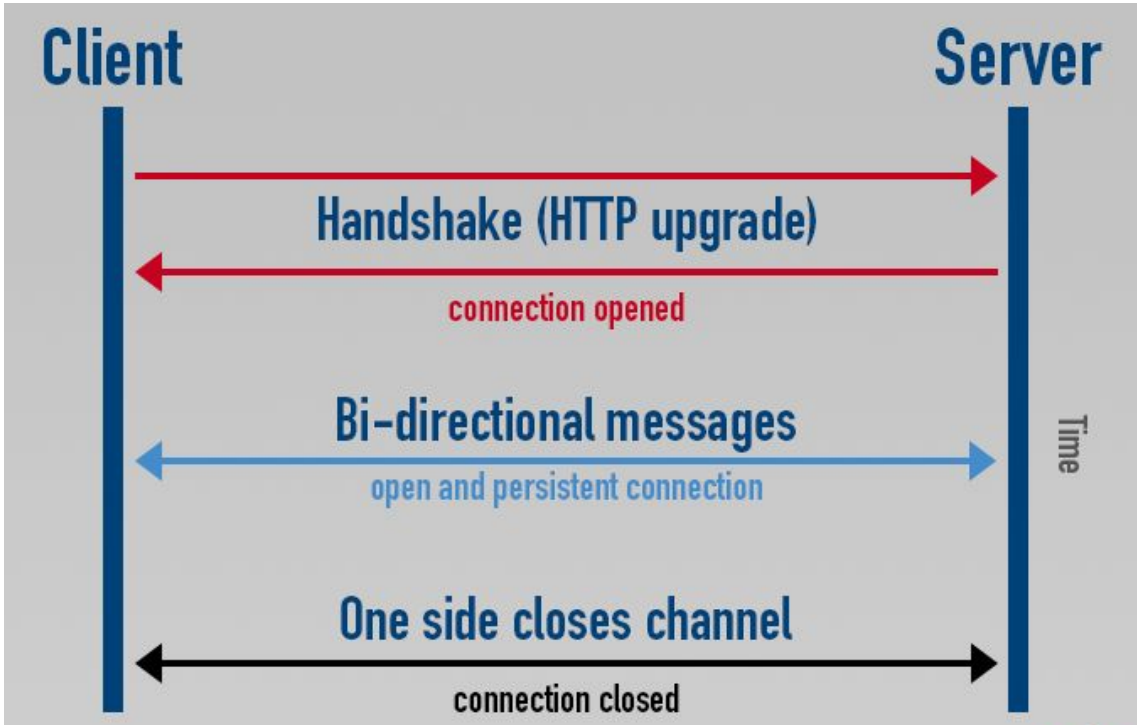
Client-server

client is session initiator

Full-duplex

exchange data
in both directions

Good for games!



WebSocket

TEXHOATOM

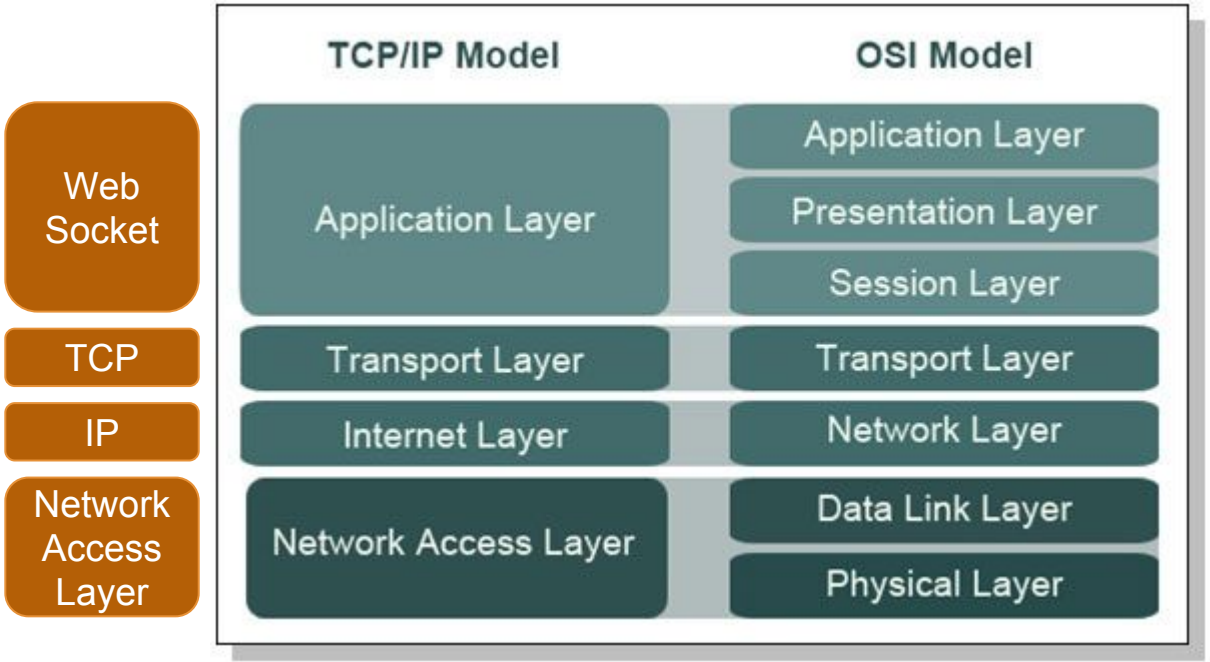
Application layer

handshake like HTTP

Over single tcp connection

but then no handshakes or headers required

only raw data



WebSockets dependencies

TEXHOATOM

We use jetty implementation of WebSocket

```
<dependency>
    <groupId>org.eclipse.jetty.websocket</groupId>
    <artifactId>websocket-api</artifactId>
    <version>9.3.7.v20160115</version>
</dependency>
<!-- To run websockets in embedded server -->
<dependency>
    <groupId>org.eclipse.jetty.websocket</groupId>
    <artifactId>websocket-server</artifactId>
    <version>9.3.7.v20160115</version>
</dependency>
```

tcpflow

ТЕХНОАТОМ

tcp traffic sniffer in human-readable format

https://github.com/simsong/tcpflow

слушать loopback интерфейс (-i) на портах 8080 и 7000 и выводить в консоль (-c)

```
sudo tcpflow -c -i lo port 8080 or port 7000
```

WebSocket. Examples

TEXHOATOM

```
@see websocket
```

Agenda

TEXHOATOM

- IO/NIO
- Game server architecture
- Web sockets
- Client-server communication

Many ways to IPC

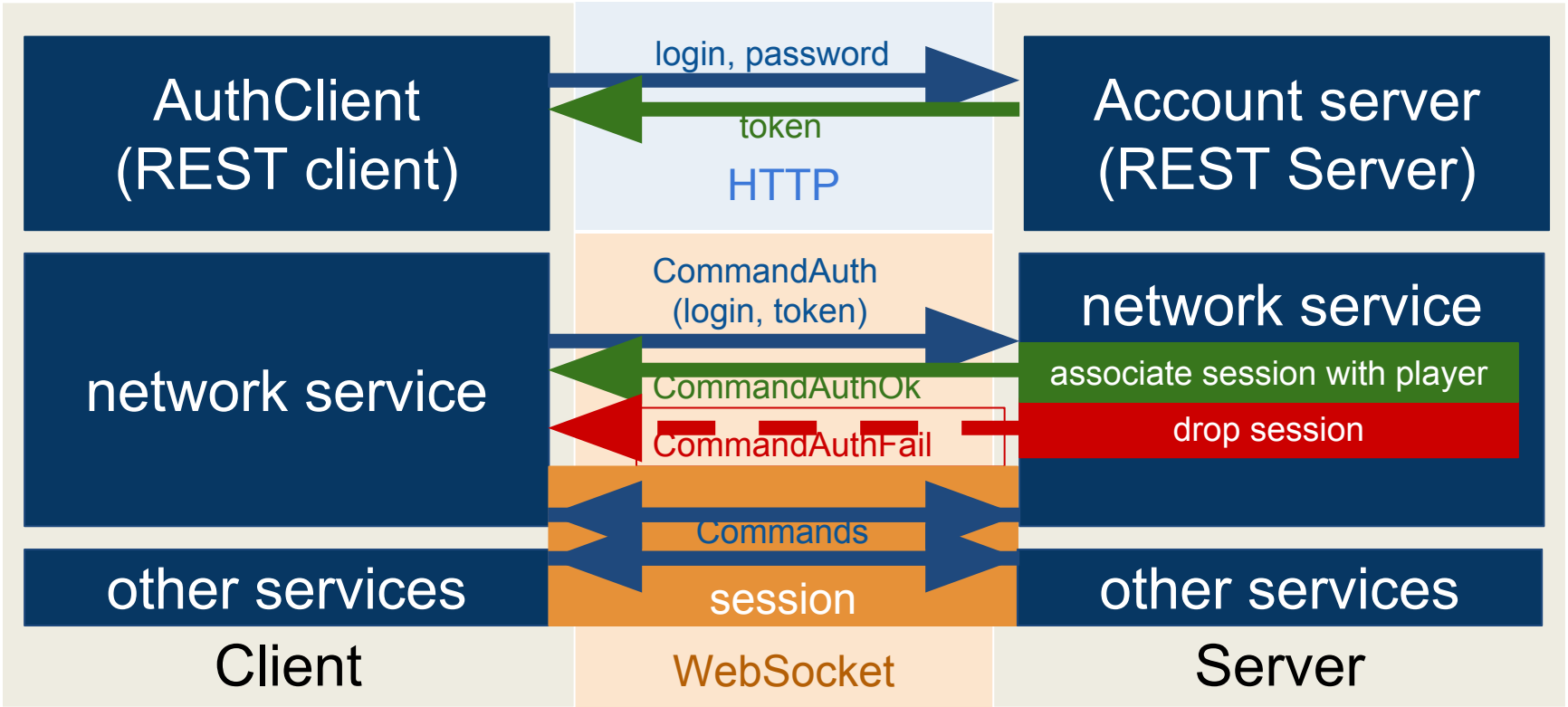
TEXHOATOM

real time game
client-server communication

How?

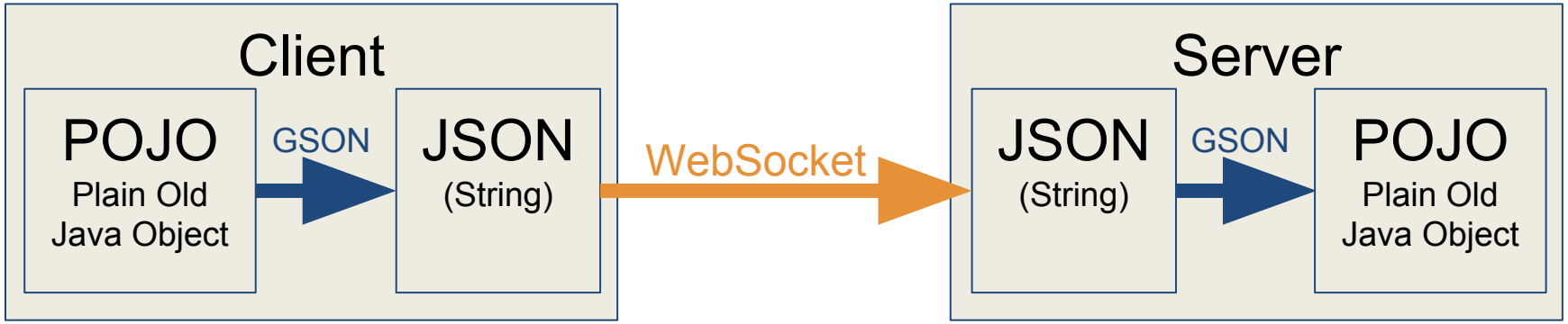
Full client authorization

TEXHOATOM



Client-server protocol

TEXHOATOM



So protocol POJO are **shared** between client and server

How can we achieve this?

Protocol project

TEXHOATOM

ZagarClientServerProtocol - project containing protocol Commands as POJO
It is included as dependency in both client and server (shared)

```
<dependency>
    <groupId>MIFIGame</groupId>
    <artifactId>ZagarClientServerProtocol</artifactId>
    <version>${clientServerProtocol.version}</version>
</dependency>
```

Protocol project contents

TEXHOATOM

Command - base class declaring that every command have a name (command field)

Client -> Server commands:

- CommandAuth
- CommandEjectMass
- CommandMove
- CommandSplit

Server -> Client commands:

- CommandAuthFail
- CommandAuthOk
- CommandLeaderBoard
- CommandReplicate

Protocol changing

TEXHOATOM

Changing protocol affects both client and server

If you change protocol you normally:

1. Change **<version>** in ZagarClientServerProtocol project
2. maven clean in ZagarClientServerProtocol
3. maven install in ZagarClientServerProtocol
   (put new version into local maven repository)

To update to new protocol change

1. clientServerProtocol.version in both server and client

Connection handling

TEXHOATOM

```
@see package network
```

network.packets contains all packets that can be sent from anywhere in server

network.handlers contains handlers for all accepted packets

ClientConnectionHandler decides which handler will service packet (based on command name)

Example - client auth

TEXHOATOM

```
@see ZagarClientServerProtocol.protocol.CommandAuth
@see client/zagar.network.packets.PacketAuth
@see server/zagar.network.handlers.PacketHandlerAuth
```

Practice - say thank you

TEXHOATOM

```
@see client/zagar.network.handlers.PacketHadlerAuthOk
Reply on CommandAuthOk with CommandThankYou(your name)
```

ТЕХНОАТОМ

Спасибо за внимание!

Александр Помосов

alpieex@gmail.com