

# 06. Сетевые приложения

Лекции по информатике  
для студентов второго курса Высшей школы ИТИС КФУ  
2020

**Ференец Александр Андреевич**

старший преподаватель кафедры программной инженерии

С использованием материалов  
к. т. н., доцента кафедры программной инженерии Абрамского М.М.

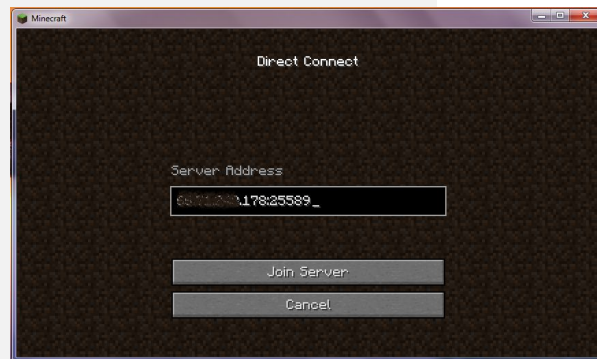
[aferenets@it.kfu.ru](mailto:aferenets@it.kfu.ru)

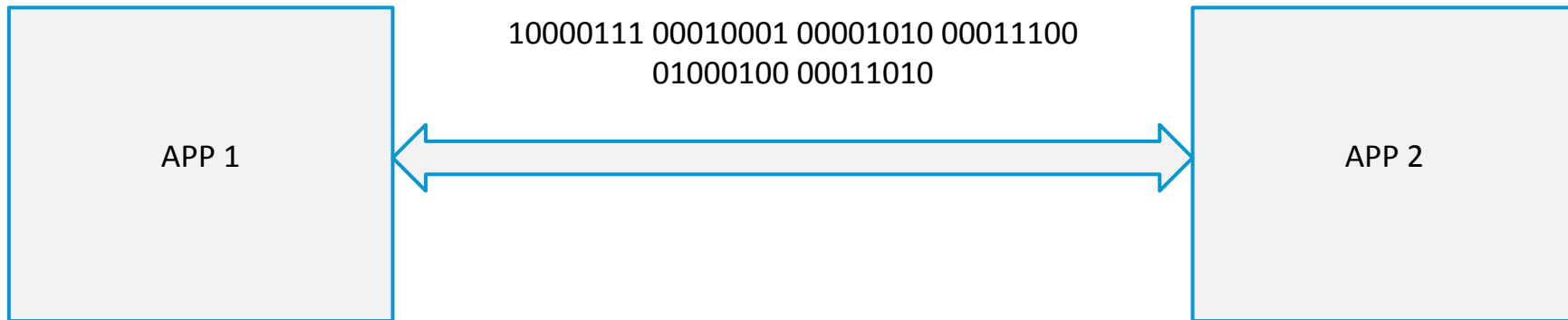


- Многие мобильные приложения
  - Instagram
  - Приложения вызова такси
  - Карты

- Сетевая игра
  - ...

- Сайт
  - Rich Web Application (Gmail, Google Drive, игры)
  - Сайт с анимациями, активными элементами
  - Просто сайт с ссылками и формами





Нужен некий набор правил взаимодействия.  
Стандарт.

**Сетевой протокол** – набор правил, действий, форматов,  
регламентирующий обмен данными между узлами сети

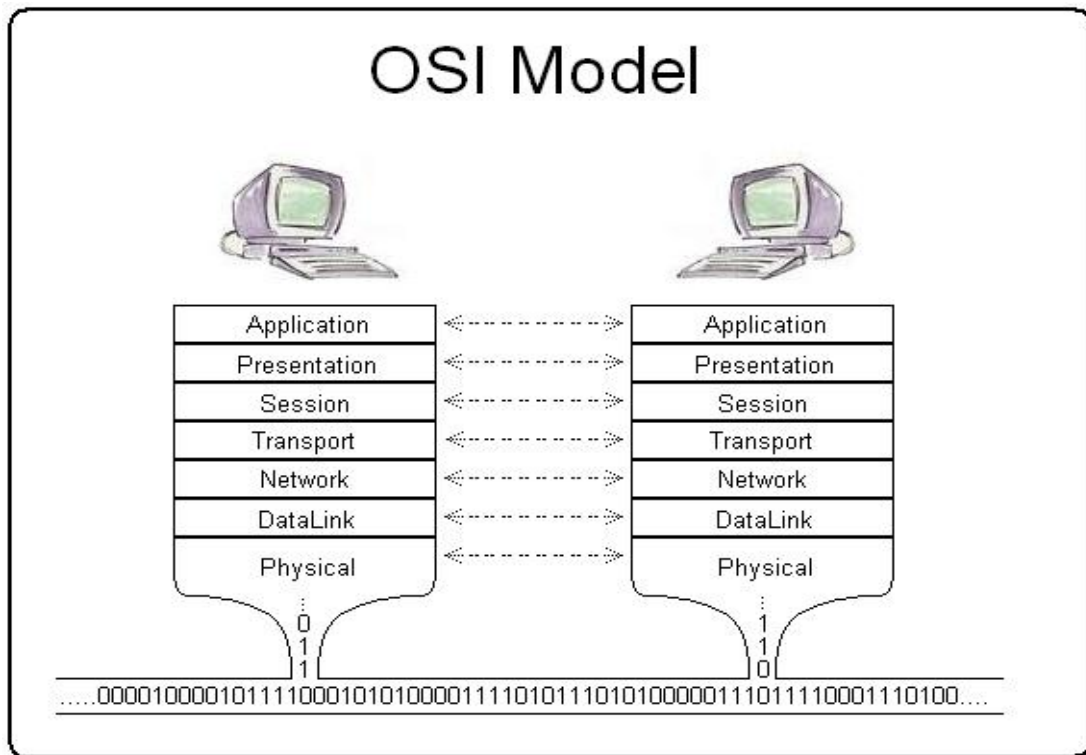
## ПРОТОКОЛ. С чего начать создание?

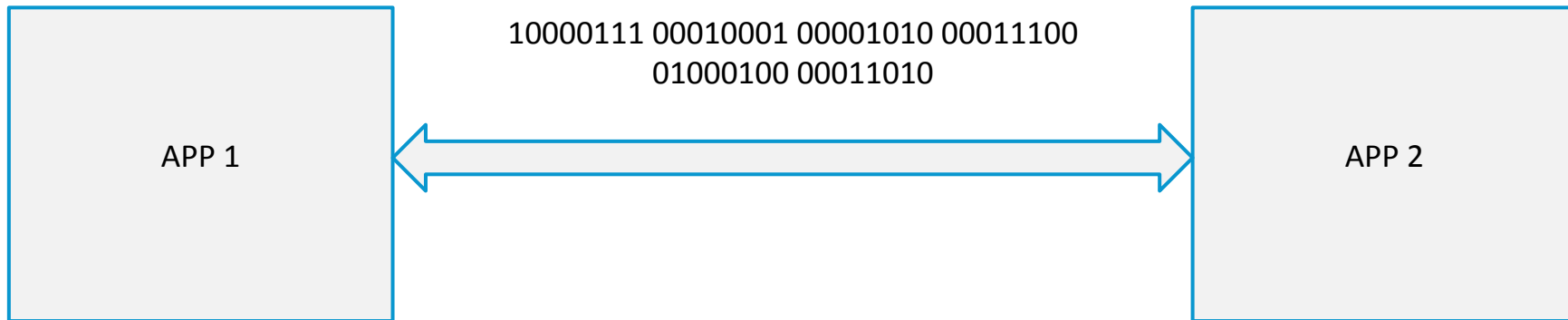
1. Кто начинает “разговор”?
2. Как понять, что “разговор” окончен?
3. Когда перестать “слушать” и можно начать “говорить”?
4. Как компактнее уместить информацию?
5. Как лучше “объяснить”, о чём информация?

1. Инициализация сеанса/сессии
2. Ping/Timeout и объявление окончания сеанса/сессии
3. Указание конца пакета, длины сообщения
4. Дробление на пакеты, архивирование
5. Выделение технической части пакета

## ПРОТОКОЛ. С чего начать создание?

На каком уровне будет работать создаваемый протокол <--> какие его основные задачи?





Хочется манипулировать битами (байтами)?

```
Client::send(byte[] data);
```

Хочется абстрагироваться!

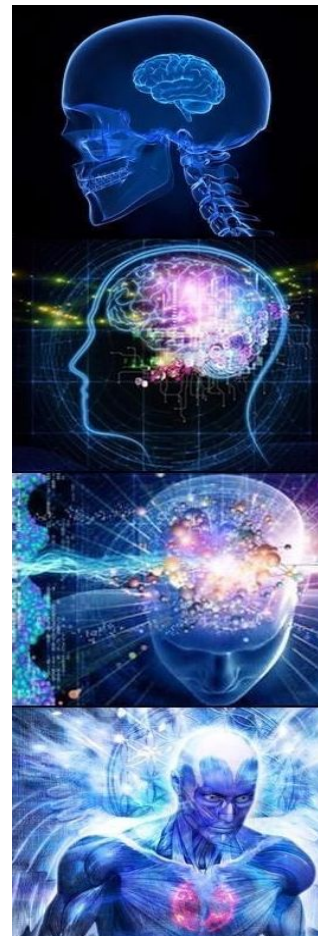
```
Client::send(Message data);
```

```
Network::send(byte[] data)
```

```
Network::send(Message data)
```

```
GameNet::send(Movement move)
```

```
Game::move(int x, int y)
```





```
Network::send(byte[] data)
```

```
Network::send(Message data)
```

```
GameNet::send(Movement move)
```

```
Game::move(int x, int y)
```

```
class Message<T>{  
    private T      data;  
    private MessageType type;  
    private Instant created;  
    private byte   priority;  
    ...  
}
```

```
class Movement{  
    private Point destination;  
    private Transport transp;  
    ...  
}
```



```
Game::move(int x, int y)
```

Какая-то логика верхних уровней абстракции. Например, взаимодействие с UI (вспоминаем сервлеты). *Presenter?*

```
GameNet::send(Movement move)
```


Сервис бизнес-логики

```
Network::send(Message data)
```

Java не знает про такие штуки, но нам удобно манипулировать на низком уровне не массивом байт, а объектом-сообщением. Придётся его сериализовать, а это уже часть придумываемого протокола.

```
OutputStream::write(byte[] data)
```

Всё равно необходимо опираться на понятия ЯП. Например, передача данных в Java → Input/Output Stream. Про Channel потом



```
Game::move(int x, int y)
```

```
GameNet::EventReact(Event e)
```

```
EventDispatcher::dispatch(NetworkEvent e)
```

```
Network::process(Message data)
```

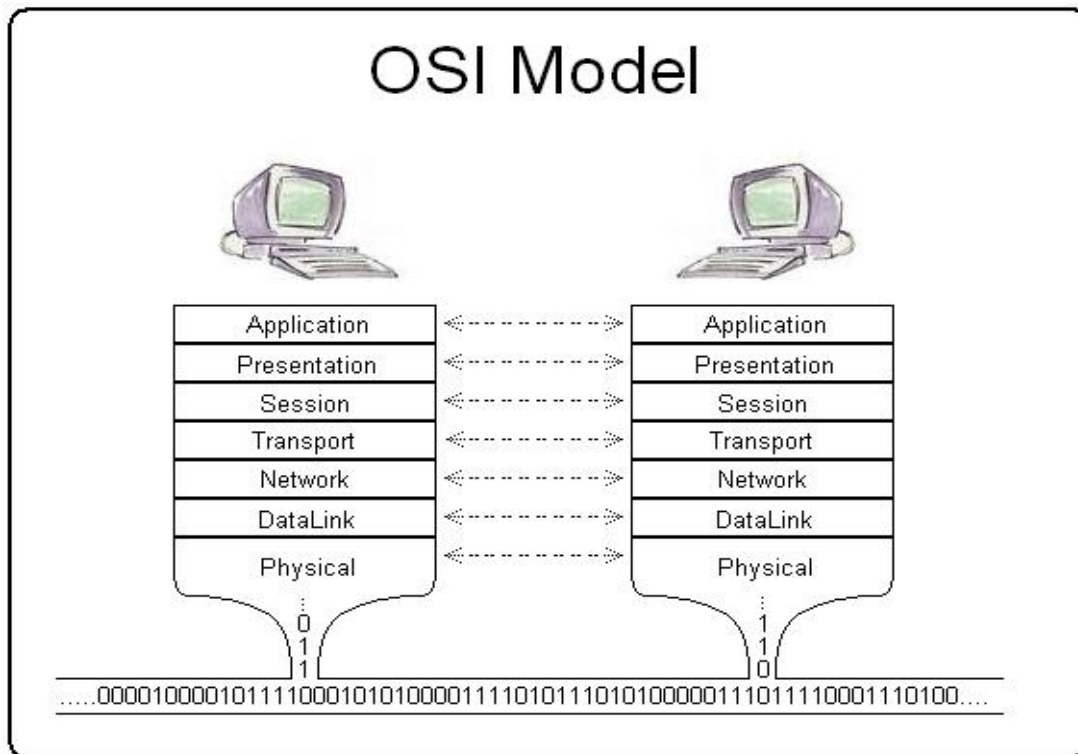
```
InputStream::read(byte[] data)
```

1. **UI и реакция на события из него** – не сетевой модуль. В других лекциях.
2. **Бизнес-логика** – не сетевой модуль. Это ваши идеи.
3. **Listeners + Events + EventDispatcher**. Это либо ваши идеи, связанные с бизнес-логикой, либо модель данных, связанная с бизнес-логикой, либо вопрос реализации шаблона Observer/Listener.
4. **Создание Output/InputStream**, логика последовательности чтения из них, сериализация/десериализация Message в/из byte[]. Это и есть создание протокола.
  - a. Типы передаваемых данных
  - b. Оптимизация
  - c. Последовательность чтения, записи – обязанности клиента-сервера
  - d. ....

Пункты 3 и 4 можно взять готовые: есть протоколы и библиотеки, работающие с ними. Можно сделать приложение на основе HTTP. А...  
*Ну, да. Сайт на сервлетах уже делали.*

# СОЗДАНИЕ ПРОТОКОЛА. Какой уровень?

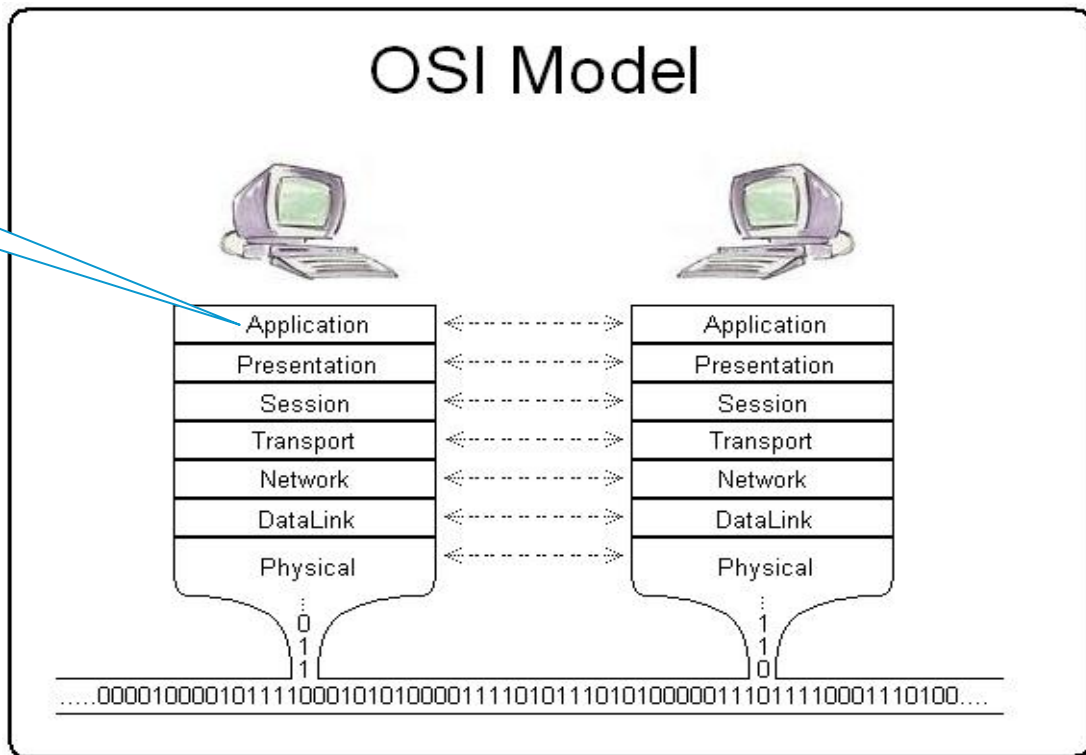
На каком уровне будет работать создаваемый протокол <--> какие его основные задачи?



## СОЗДАНИЕ ПРОТОКОЛА. Какой уровень?

На каком уровне будет работать создаваемый протокол  $\leftrightarrow$  какие его основные задачи?

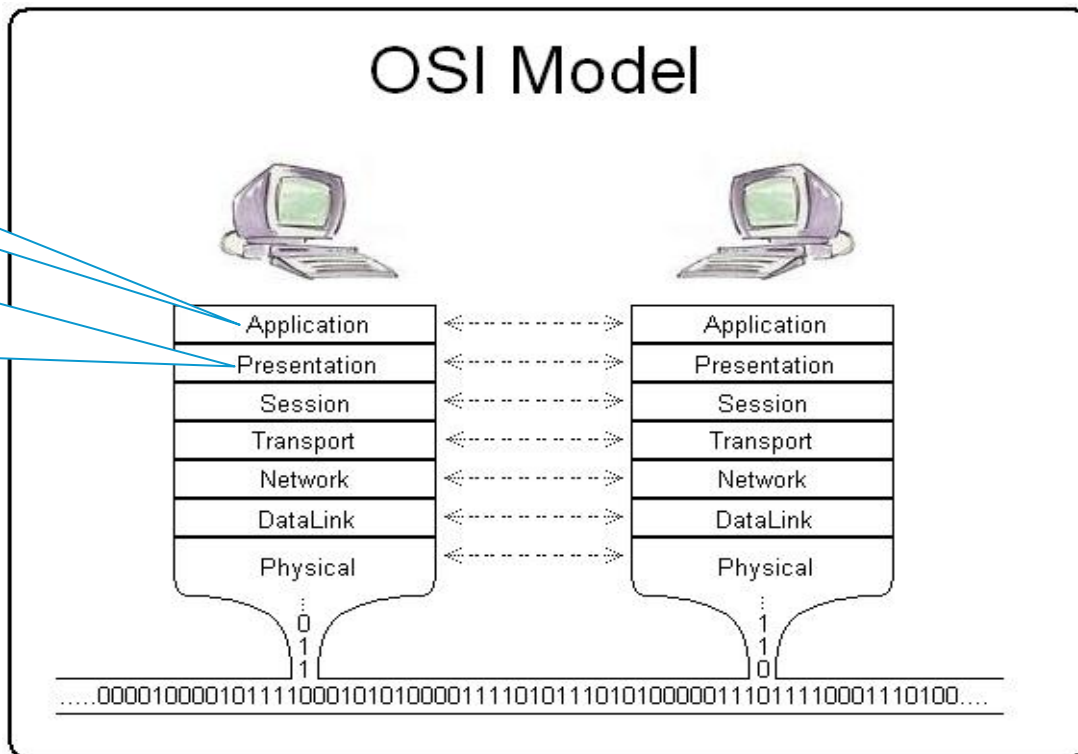
Своя модель и  
бизнес-логика.



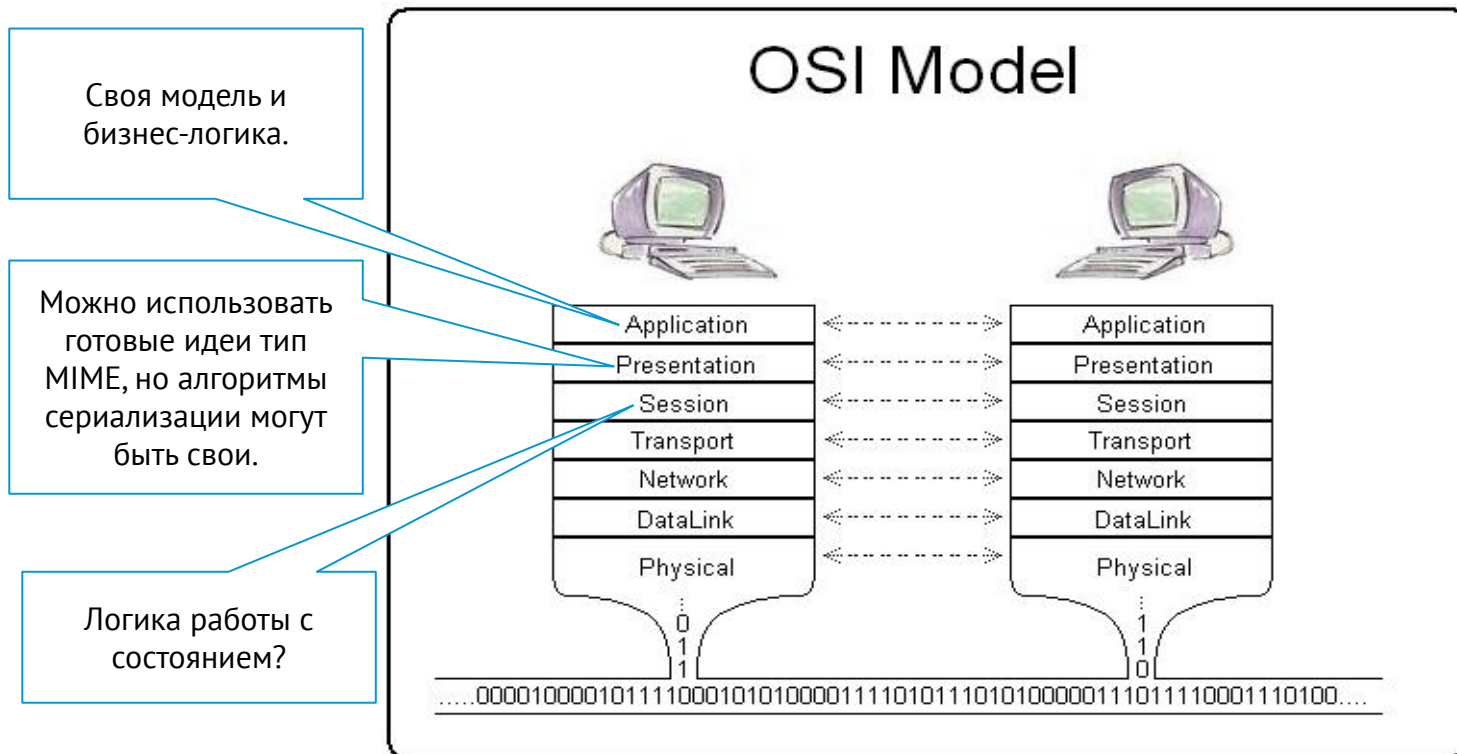
На каком уровне будет работать создаваемый протокол  $\leftrightarrow$  какие его основные задачи?

Своя модель и  
бизнес-логика.

Можно использовать  
готовые идеи тип  
MIME, но алгоритмы  
сериализации могут  
быть свои.

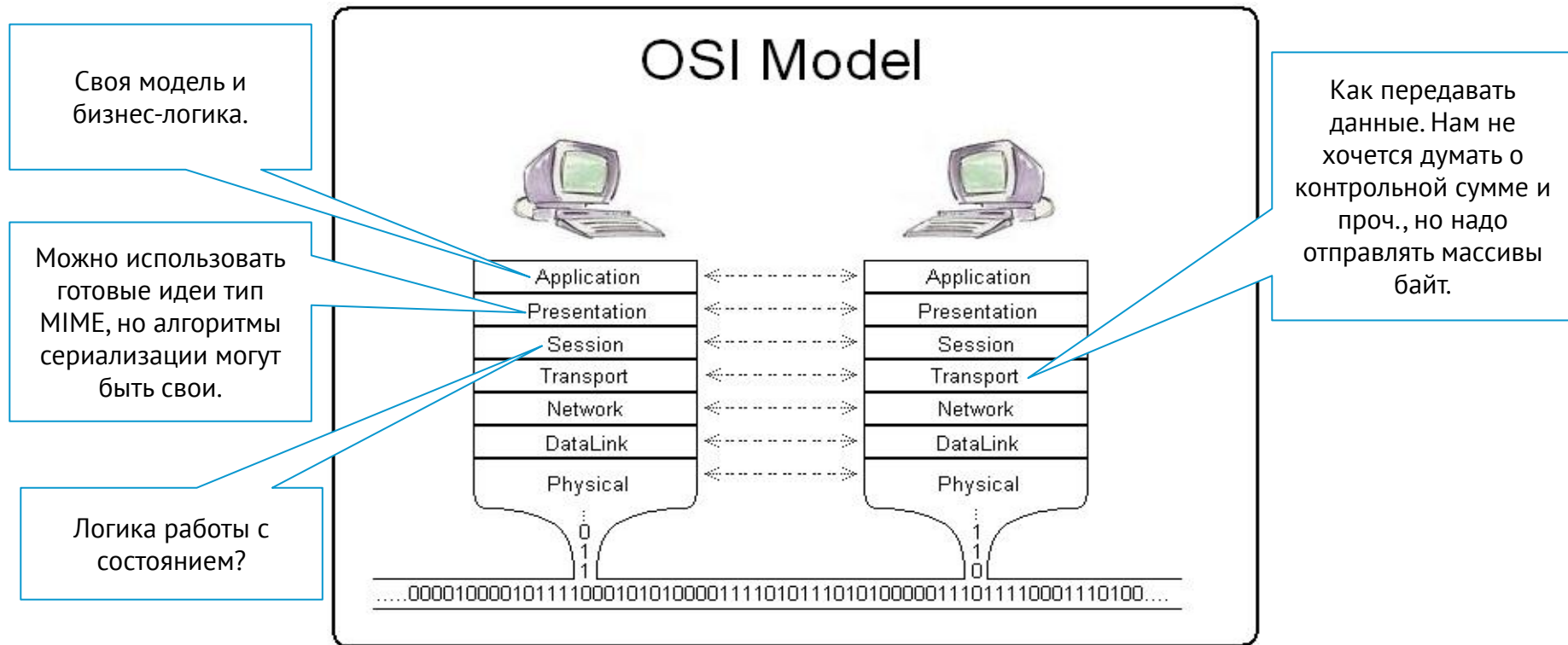


На каком уровне будет работать создаваемый протокол  $\leftrightarrow$  какие его основные задачи?



## СОЗДАНИЕ ПРОТОКОЛА. Какой уровень?

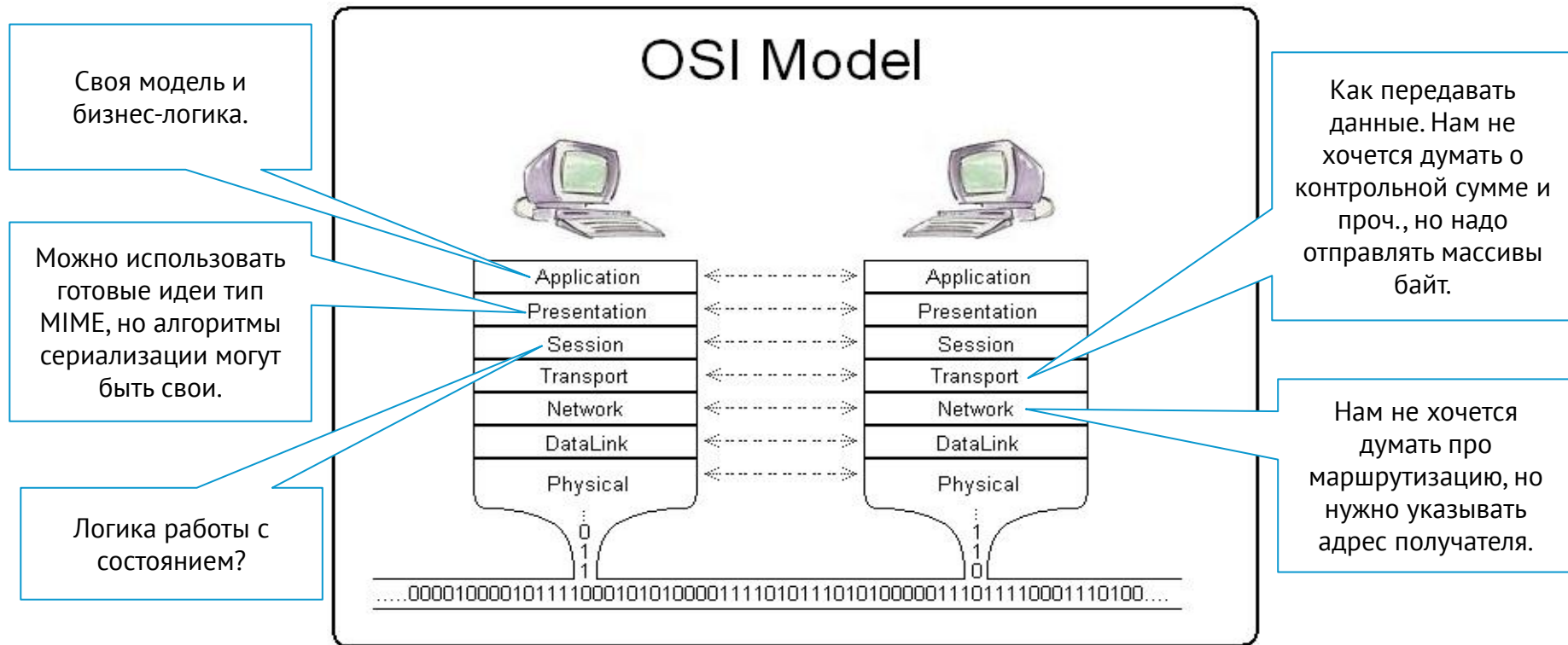
На каком уровне будет работать создаваемый протокол  $\leftrightarrow$  какие его основные задачи?





## СОЗДАНИЕ ПРОТОКОЛА. Какой уровень?

На каком уровне будет работать создаваемый протокол  $\leftrightarrow$  какие его основные задачи?





## СОЗДАНИЕ ПРОТОКОЛА. Какой уровень?

TCP vs UDP?  
Socket vs Datagram

Сокет



Сокет

**Сокет** – точка соединения сетевого приложения.

Не важно, кто сервер, а кто клиент. Какая логика взаимодействия.

*Не путать с протоколом WebSockets!*

```
10 ▶ public class Client {
11 ▶   public static void main(String[] args) throws Throwable {
12
13       System.out.println("Starting client...");
14       Socket s = new Socket(InetAddress.getLocalHost(), Protocol.PORT);
15       OutputStream out = s.getOutputStream();
16       System.out.println("Starting sending random bytes");
17       while(true){
18           Thread.sleep( millis: 1000);
19           byte random = (byte)(new Random()).nextInt( bound: 127);
20           System.out.println(">> " + random);
21           out.write(random);
22           out.flush();
23       }
24   }
25 }
```

```
9  ▶ public class Server {
10 ▶  public static void main(String[] args) throws Throwable {
11
12      System.out.println("Starting server...");
13      ServerSocket server = new ServerSocket(Protocol.PORT);
14      System.out.println("Starting to listen port " + Protocol.PORT);
15      Socket s = server.accept();
16      System.out.println("Got one connection.");
17      InputStream in = s.getInputStream();
18      int b;
19      while((b = in.read()) != -1){
20          System.out.println("<< " + b);
21      }
22  }
23 }
24
```

<CODE>

Клиент отправляет случайные байты серверу, а тот их выводит в консоль.

Конструктор класса Socket:

```
Socket(InetAddress address, int port)
```

```
Socket(String host, int port)
```

Не URI !!!!!11

**Хост** – это компьютер в сети.



IP. Internet Protocol. Сетевой уровень -> создание сети, выдача адресов.

- IPv4 (1981)
- IPv6 (1996)

127.0.0.01

0:0:0:0:0:0:0:1

```
InetAddress.getLocalHost()
```

```
InetAddress.getByName(String hostname)
```

Доменные имена легче запоминаются.

*Доменное имя != домен.*

RFC 1035, 2606, ...

maps.yandex.ru – доменное имя третьего уровня

ru – доменное имя первого уровня

. – доменное имя нулевого уровня

## ICANN (Internet Corporation for Assigned Names and Numbers)

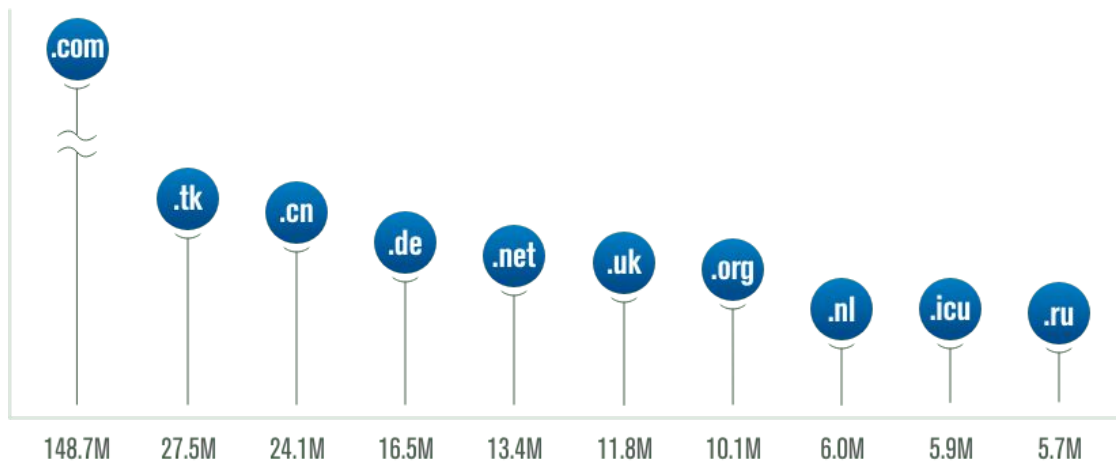
.eu	720 руб.	.com.ua	990 руб.	.biz	1680 руб.	.ooo	2810 руб.	.design	4760 руб.
.kiev.ua	760 руб.	.cc	1080 руб.	.бел	1780 руб.	.nyc	2920 руб.	.mn	4870 руб.
.odessa.ua	860 руб.	.kharkov.ua	1330 руб.	.lv	2180 руб.	.wedding	2920 руб.	.fm	11440 руб.
.in	880 руб.	.asia	1350 руб.	.ws	2710 руб.	.gmbh	3030 руб.		
.us	920 руб.	.by	1640 руб.	.bz	2810 руб.	.ua	4290 руб.		

цены Timeweb, 10.11.20

TLD	Register	Renew	Transfer
SALE .com *	\$8.88	\$12.98	<del>\$8.58</del> SPECIAL \$8.88
SALE .net *	<del>\$10.78</del> SPECIAL \$12.98	\$14.98	<del>\$10.98</del> SPECIAL \$11.88
SALE .org *	<del>\$9.18</del> SPECIAL \$12.98	\$14.98	\$11.48
SALE .io	<del>\$28.88</del> SPECIAL \$32.98	\$34.88	<del>\$28.88</del> BEST MARKET PRICE \$32.88
SALE .co	<del>\$7.98</del> 67% OFF \$23.98	\$25.98	<del>\$21.28</del> SPECIAL \$32.99
SALE .ai	<del>\$58.88</del> \$68.88	\$68.88	<del>\$158.98</del> SPECIAL \$168.98
SALE .co.uk	<del>\$6.88</del> SPECIAL \$7.58	\$9.58	\$0.00
.ca	\$11.98	\$13.98	\$11.98
.dev *	\$14.98	\$16.98	\$14.98
SALE .me	<del>\$5.88</del> SPECIAL \$18.99	\$20.98	\$15.99

цены Namecheap, 10.11.20

ICANN (Internet Corporation for Assigned Names and Numbers)



## Domain Name Service

Сервис сопоставления доменных имён и IP-адресов. Есть соот. Протоколы обмена информации.

Путём подмены информации можно заблокировать доступ, “подложив” свой адрес.

Есть публичные DNS-сервера. Google:

8.8.8.8

8.8.4.4

Часть записей можно внести в локальный файл сопоставлений:

/etc/hosts

G:\Windows\System32\drivers\etc\hosts

## DNS. Демонстрация использования hosts



Пример файла hosts и демонстрация работы браузера с настроенным для рабочей среды файлом.

```
class Connection implements Runnable {  
    private Socket socket;  
    private Thread thread;  
    private Server server;  
  
    public Connection(Server server, Socket socket) {  
        //...  
        this.socket = socket;  
        this.server = server;  
        //...  
        thread = new Thread(this);  
        thread.start();  
    }  
  
    public void run() {  
        // Взаимодействие между клиентом и сервером  
    }  
}
```

```
public class Server {
    private List<Connection> connections;

    public Server() throws IOException {
        connections = new ArrayList<>();
        init();
    }

    public void init() throws IOException {
        ServerSocket s1 = new ServerSocket(1234);
        while (true) {
            Socket client = s1.accept();
            connections.add(new Connection(this, client));
        }
    }
    // если вдруг у нас несколько серверов
    public static void main(String[] args) throws IOException {
        Server server = new Server();
    }
}
```