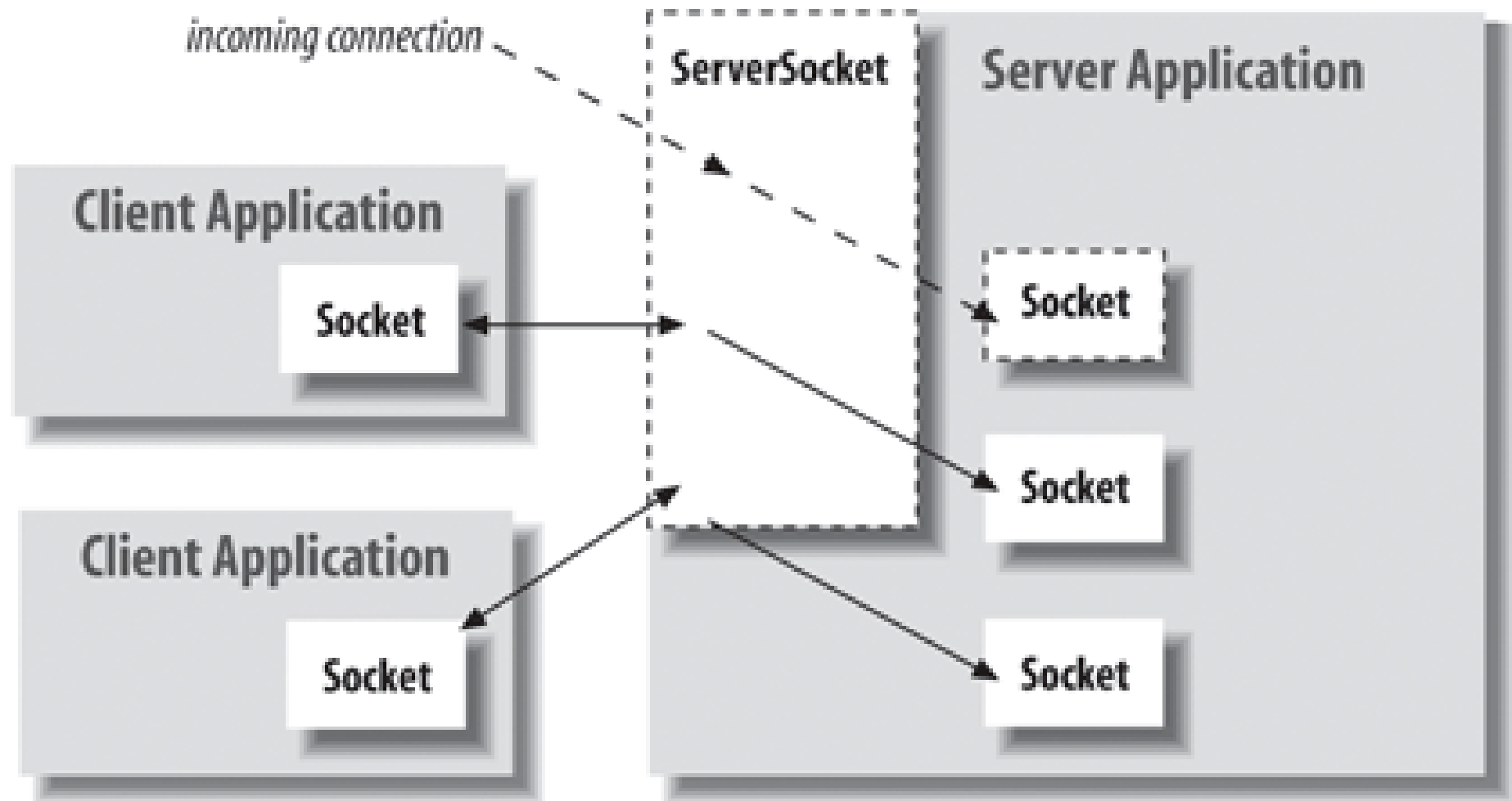


Сокети – от страна на клиент

Клиент и сървър



Основи операции на сокет

- Свързване към отдалечена машина
- Изпращане на данни
- Получаване на данни
- Затваряне на връзката
- Свързване към даден порт
- Слушане за входящи данни
- Приемане на връзки от отдалечени машини за даден порт

Жизнен цикъл на клиентско приложение

- Програмата създава нов сокет с конструктор.
- Сокетът се опитва да се свърже със зададения отделен хост.
- От свързания успешно сокет се взема входния поток и се прочита това, което сървърът изпраща.
- След приключване на работа сокетът се затваря.

java.net.Socket

- public Socket(String host, int port) throws UnknownHostException, IOException
 Socket toOReilly = new Socket("www.oreilly.com", 80);
- public Socket(InetAddress host, int port) throws IOException
 InetAddress oreilly = InetAddress.getByName("www.oreilly.com");
 Socket oreillySocket = new Socket(oreilly , 80);
- public Socket(InetAddress host, int port, InetAddress interface, int localPort) throws IOException
 InetAddress inward = InetAddress.getByName("router");
 InetAddress mail = InetAddress.getByName("mail");
 Socket socket = new Socket(mail, 25, inward, 0);
- protected Socket()

Прост TCP клиент

```
import java.io.*;
import java.net.Socket;
public class DateServerClient {
public static void main(String[] args) throws IOException
{
    Socket socket = new Socket("localhost", 2002);
    BufferedReader in = new BufferedReader(
        new InputStreamReader(
            socket.getInputStream() ) );
    System.out.println("The date on the server is: " +
        in.readLine());
    socket.close();
}
}
```

Информация от сокета

- `public InetAddress getInetAddress()`
`Socket theSocket = new Socket("java.sun.com", 80);`
- `InetAddress host = theSocket.getInetAddress();`
`public int getPort()`
- `int port = theSocket.getPort();`
- `public int getLocalPort()`
- `public InetAddress getLocalAddress()`
- `public void close() throws IOException`

Исключения

- **public class** SocketException **extends** IOException
- **public class** BindException **extends** SocketException
- **public class** ConnectException **extends** SocketException
- **public class** NoRouteToHostException **extends** SocketException

Примери

- Find out which of the first 1,024 ports seem to be hosting TCP servers on a specified host - [LowPortScanner.java](#)
- Find out which of the ports at or above 1,024 seem to be hosting TCP servers - [HighPortScanner.java](#)
- Get a socket's information – [SocketInfo.java](#)