

Scalability Layer
hits

The Internet Stack

The Internet

connecting boxes

The Web

connecting documents

What about
services?

Fast.

Cheap.

Out of control.

Request/Reply

```
import zmq
```

```
ctx = zmq.Context()
```

```
s = ctx.socket(zmq.REP)
```

```
s.bind("tcp://*:5566")
```

```
while True:
```

```
    msg = s.recv()
```

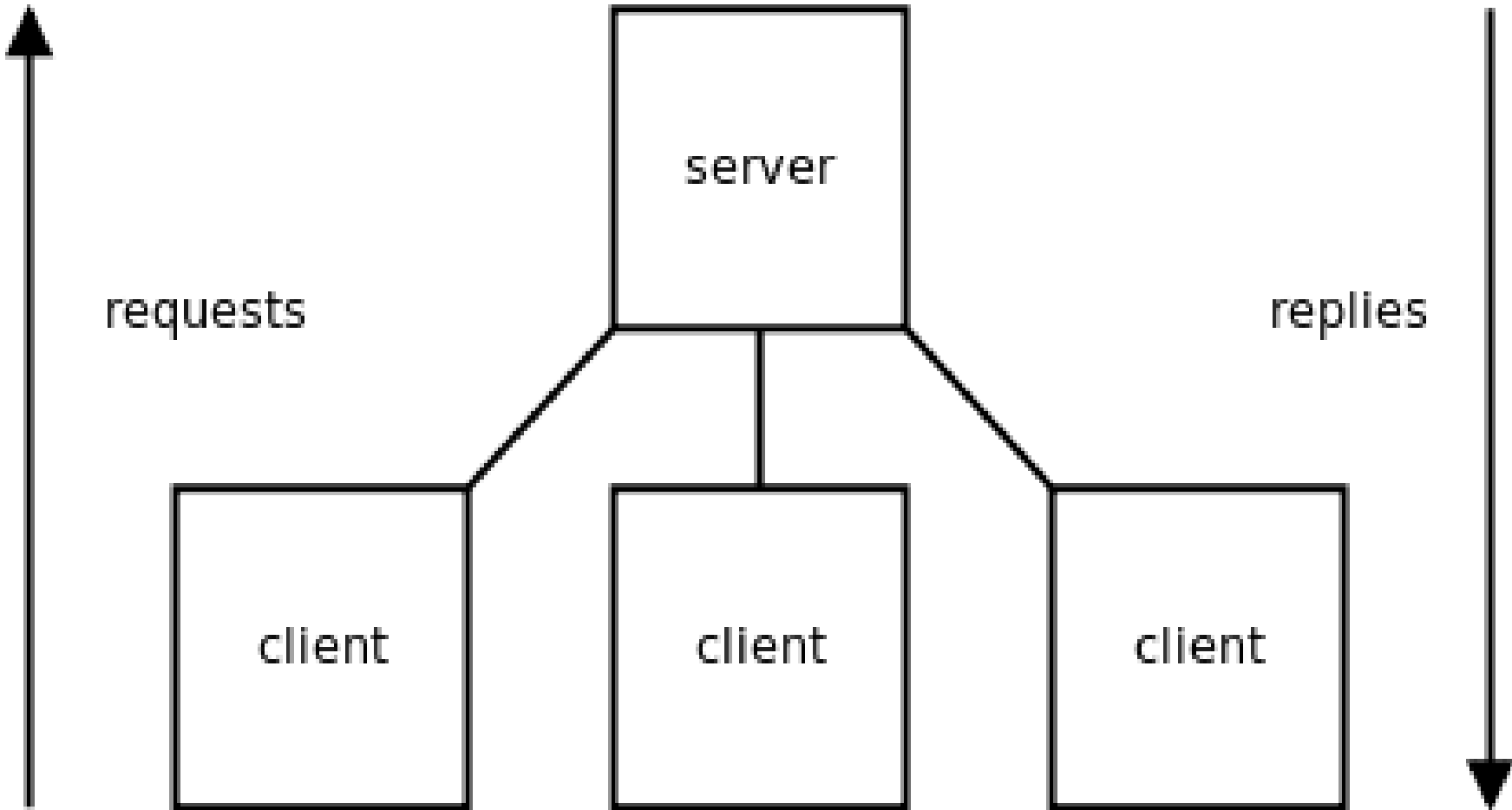
```
    print msg
```

```
    s.send("World")
```

```
import zmq

ctx = zmq.Context()
s = ctx.socket(zmq.REQ)
s.connect("tcp://localhost:5566")

s.send("Hello")
msg = socket.recv()
print msg
```

Publish/Subscribe

```
import zmq
```

```
ctx = zmq.Context()
```

```
s = context.socket(zmq.PUB)
```

```
s.bind("tcp://*:5566")
```

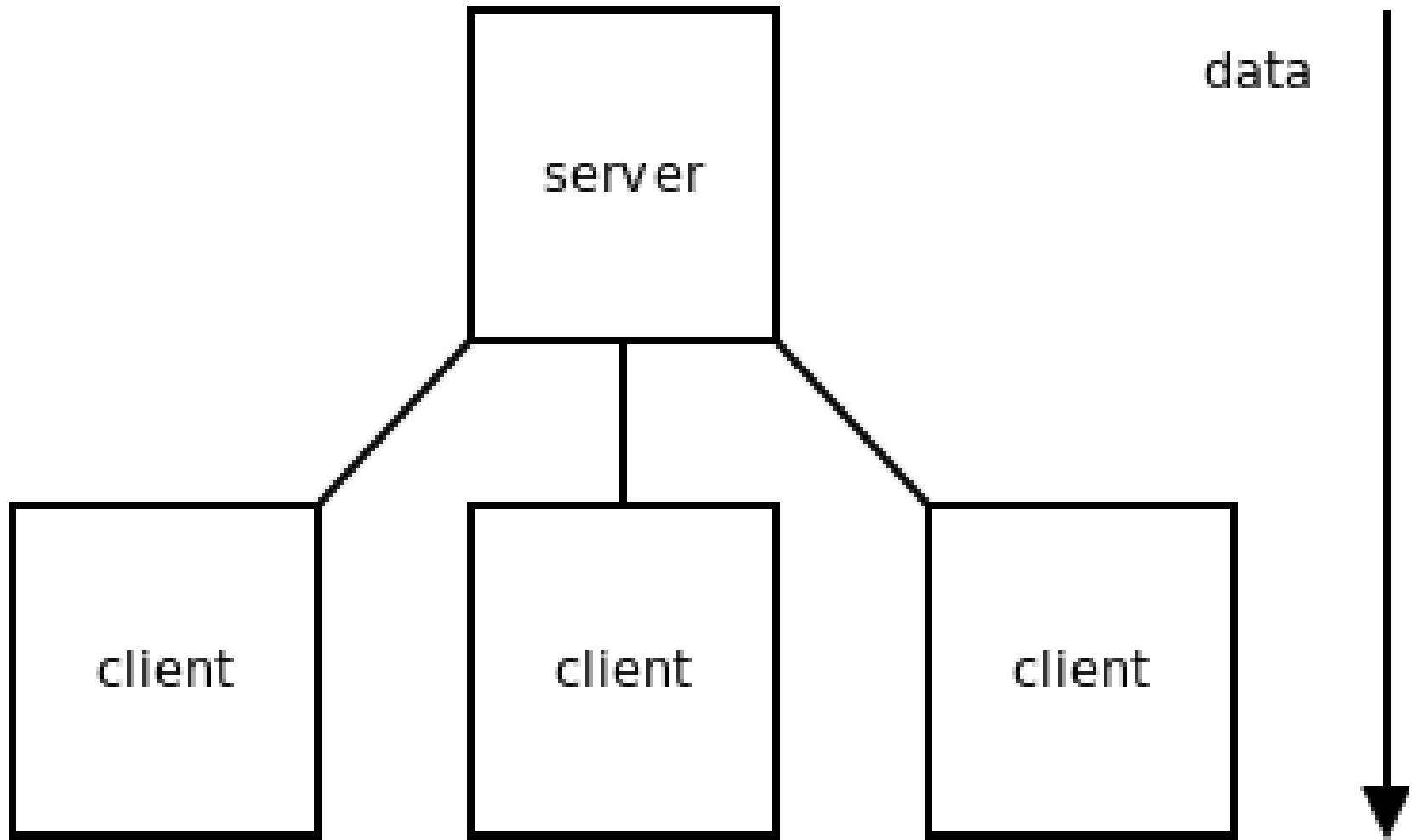
```
while True:
```

```
    s.send("HELLO?")
```

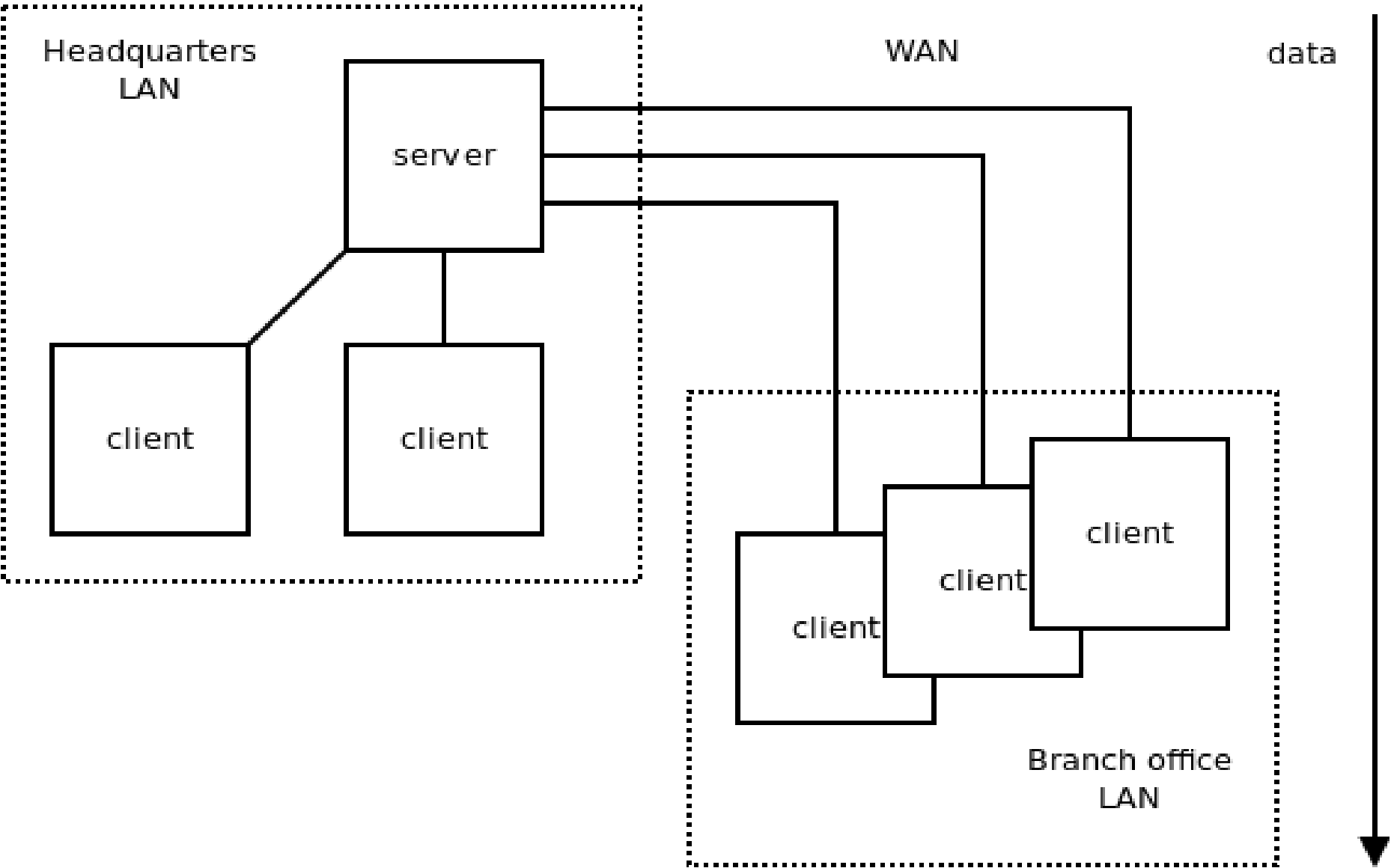
```
import zmq

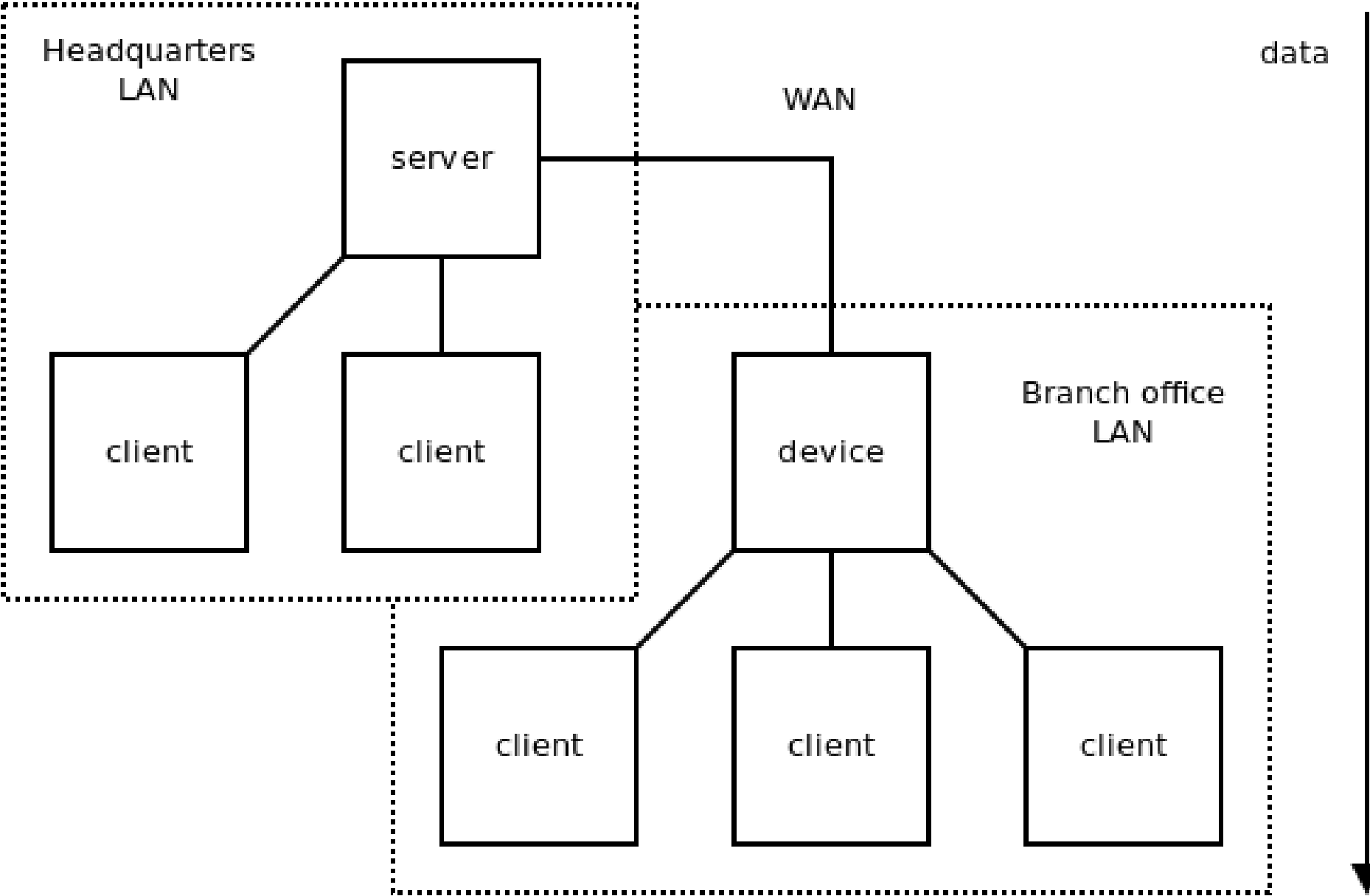
ctx = zmq.Context()
s = ctx.socket(zmq.SUB)
s.connect("tcp://localhost:5566")
s.setsockopt(zmq.SUBSCRIBE, "")

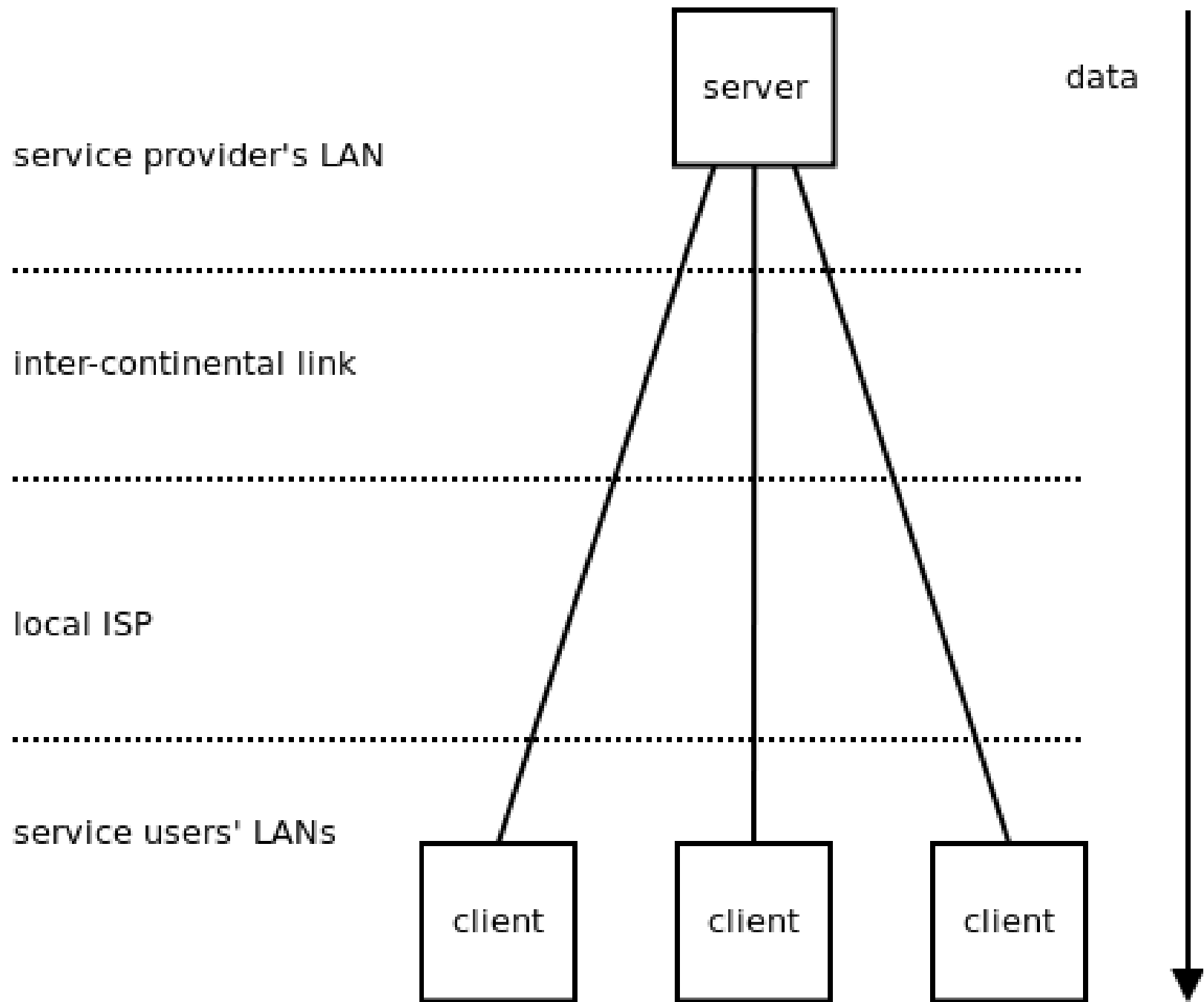
while True:
    msg = s.recv()
    print msg
```

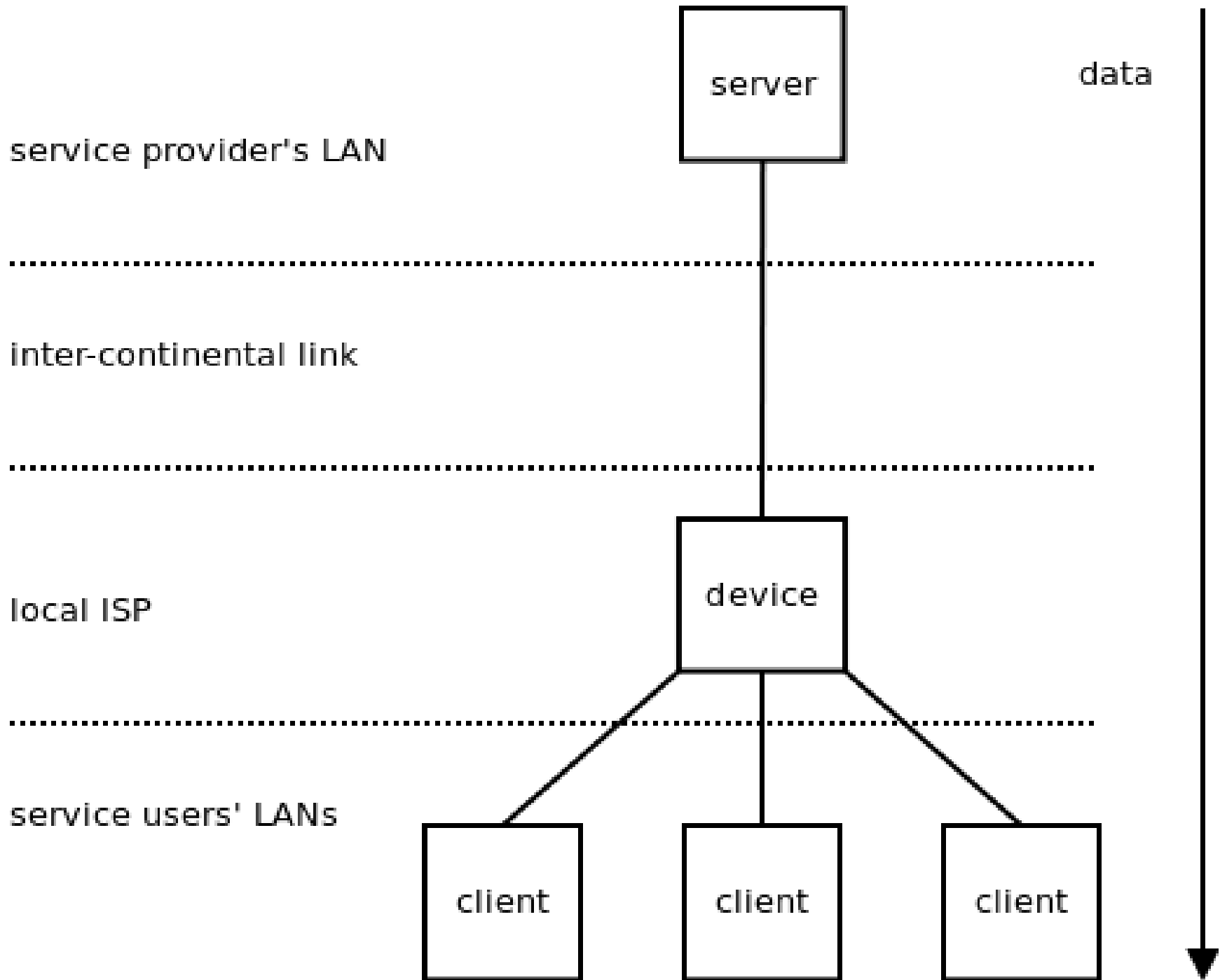


Scaling up Publish/Subscribe

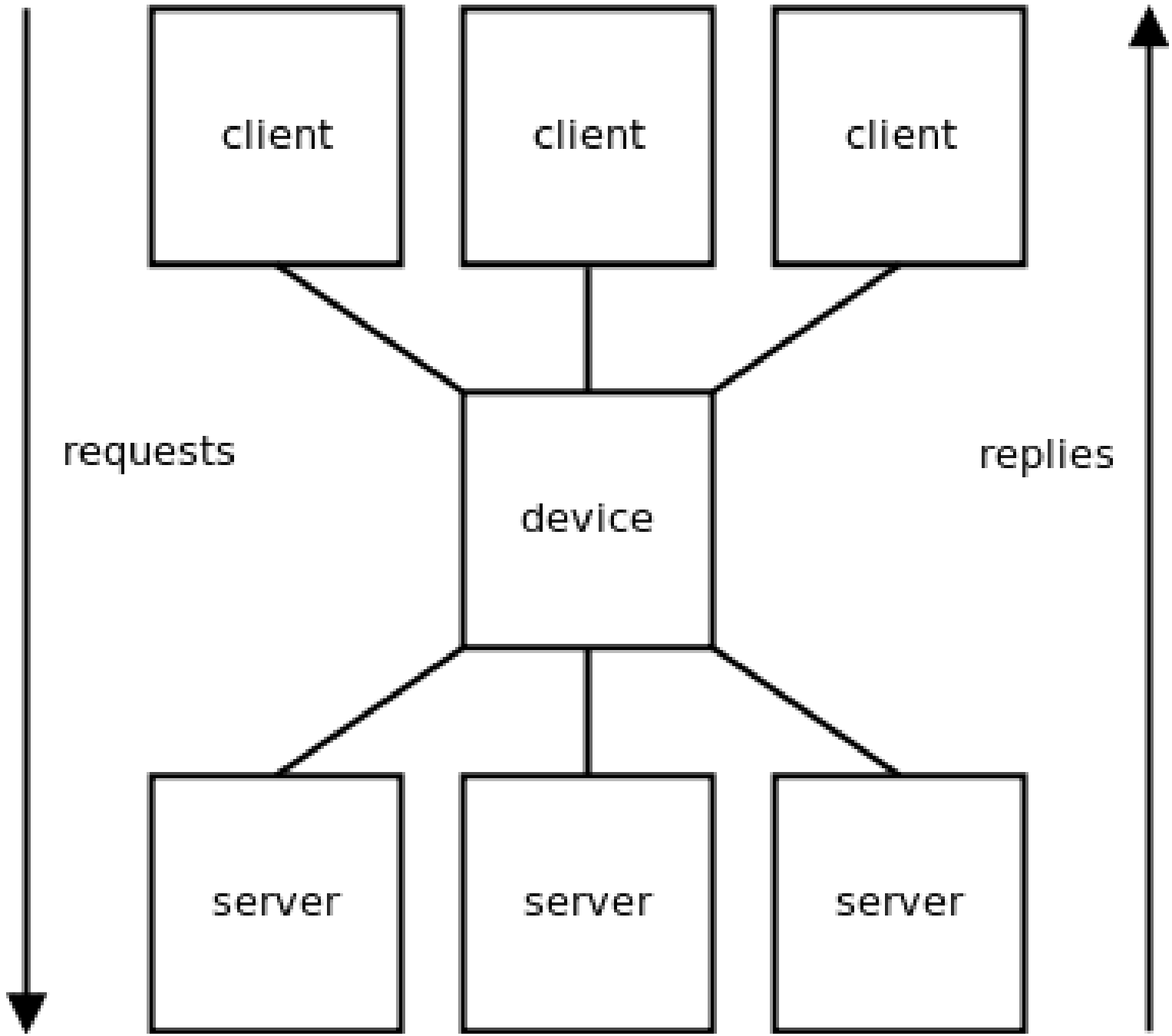


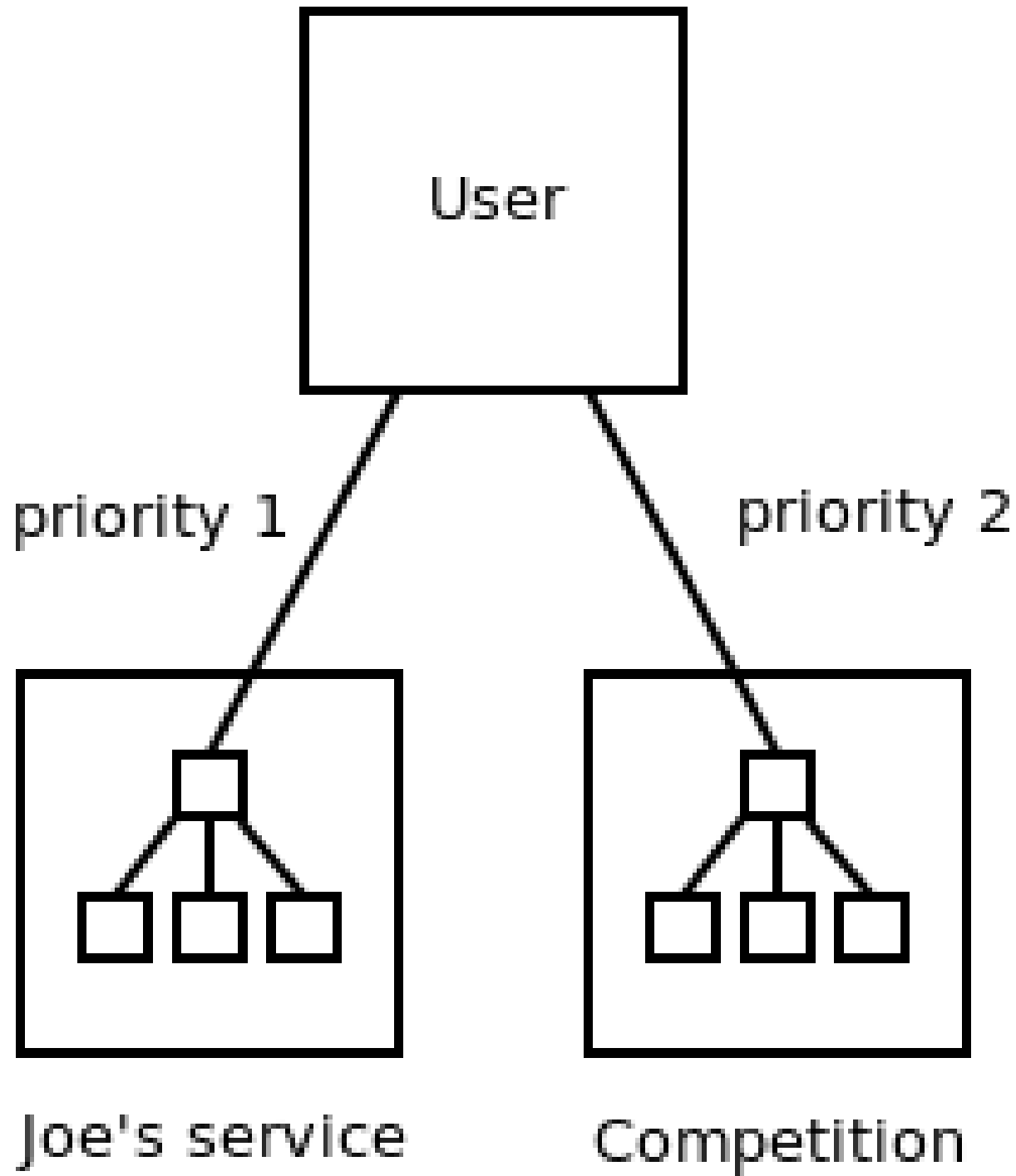


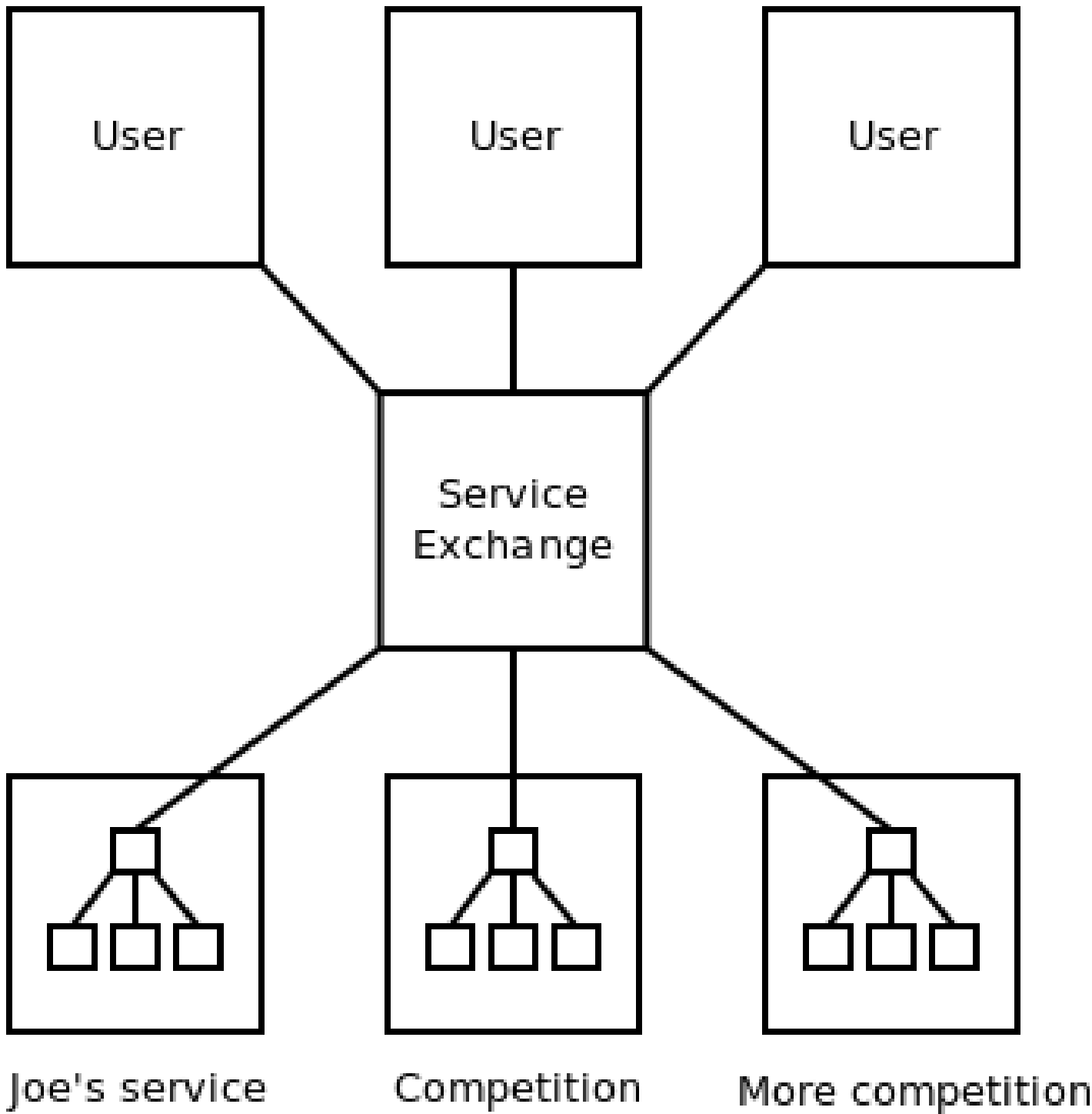


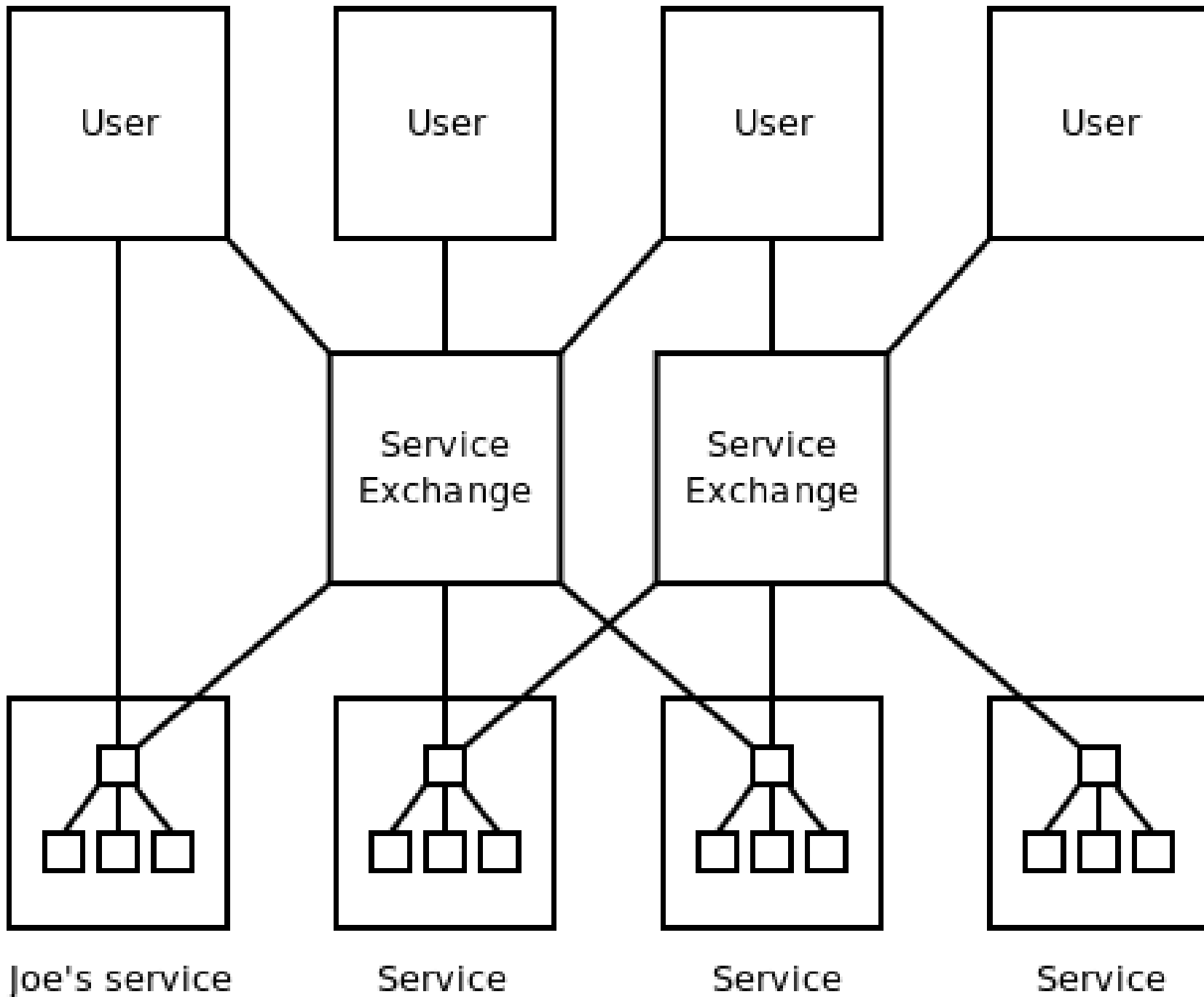


Scaling up Request/Reply









Extending the Internet Stack

SP

OSI stack

Internet stack

Application layer (7)	Application
Presentation layer (6)	XDR
Session layer (5)	SP
Transport layer (4)	TCP
Network layer (3)	IP
Data Link layer (2)	Ethernet
Physical layer (1)	

Semantics

API

POSIX socket API

```
#include <zmq.h>

int main (int argc, char *argv[])
{
    void *ctx, *s;
    zmq_msg_t *msg;

    ctx = zmq_init (1);
    s = zmq_socket(ctx, ZMQ_REQ);
    zmq_bind (s, "tcp://*:5577");
    zmq_msg_init (msg, 6);
    strcpy (zmq_msg_data (msg), "Hello");
    zmq_send (s, msg, 0);
    zmq_msg_close (msg);
    zmq_close (s);
    zmq_term (ctx);
    return 0;
}
```

```
#include <sys/socket.h>
```

```
#include <netinet/sp.h>
```

```
int main (int argc, char *argv[])
```

```
{
```

```
    int s;
```

```
    struct sockaddr_in addr;
```

```
    s = socket(PF_SP, SOCK_REQ, IPPROTO_TCP);
```

```
    addr.sin_family = AF_INET;
```

```
    addr.sin_port = 5577;
```

```
    addr.sin_addr = INADDR_ANY;
```

```
    bind (s, addr, sizeof addr);
```

```
    send (s, "Hello", 6, 0);
```

```
    close (s);
```

```
    return 0;
```

```
}
```

Protocol

IETF

Kernel Implementation

Why?

Value-add?

Is it sane?

www.zeromq.com
Martin Sústrik
sustrik@250bpm.com