

**ØMQ**

**Сокеты на стероидах**

# Who is Who





# Требование первое: Скорость





## Требование второе: Надежность







# Расплата

```
;~ Get file size
mov     ebx, szFile
mov     ecx, stat
mov     eax, sys_newstat
int     80H

;~ Get end of bss section
xor     ebx, ebx
mov     eax, sys_brk
int     80H
mov     [Org_Break], eax
mov     [TempBuf], eax
push   eax

; extend it by file size
pop     ebx
add     ebx, dword [stat + STAT.st_size]
mov     eax, sys_brk
int     80H

; open file
mov     ebx, szFile
mov     ecx, 0_RDONLY
xor     edx, edx
mov     eax, sys_open
int     80H
xchg   eax, esi

; read in file to buffer
mov     ebx, esi
mov     ecx, [TempBuf]
mov     edx, dword [stat + STAT.st_size]
mov     eax, sys_read
int     80H

; display to terminal
mov     ebx, stdout
mov     ecx, [TempBuf]
mov     edx, eax
mov     eax, sys_write
int     80H

; close file
mov     ebx, esi
mov     eax, sys_close
int     80H

; "free" memory
mov     ebx, [Org_Break]
```

I

**2004**

JP Morgan инициирует создание AMQP



IMatix создает спецификацию протокола AMQP  
IMatix разрабатывает брокер на C

**2010**

Начало потери интереса

**2011**

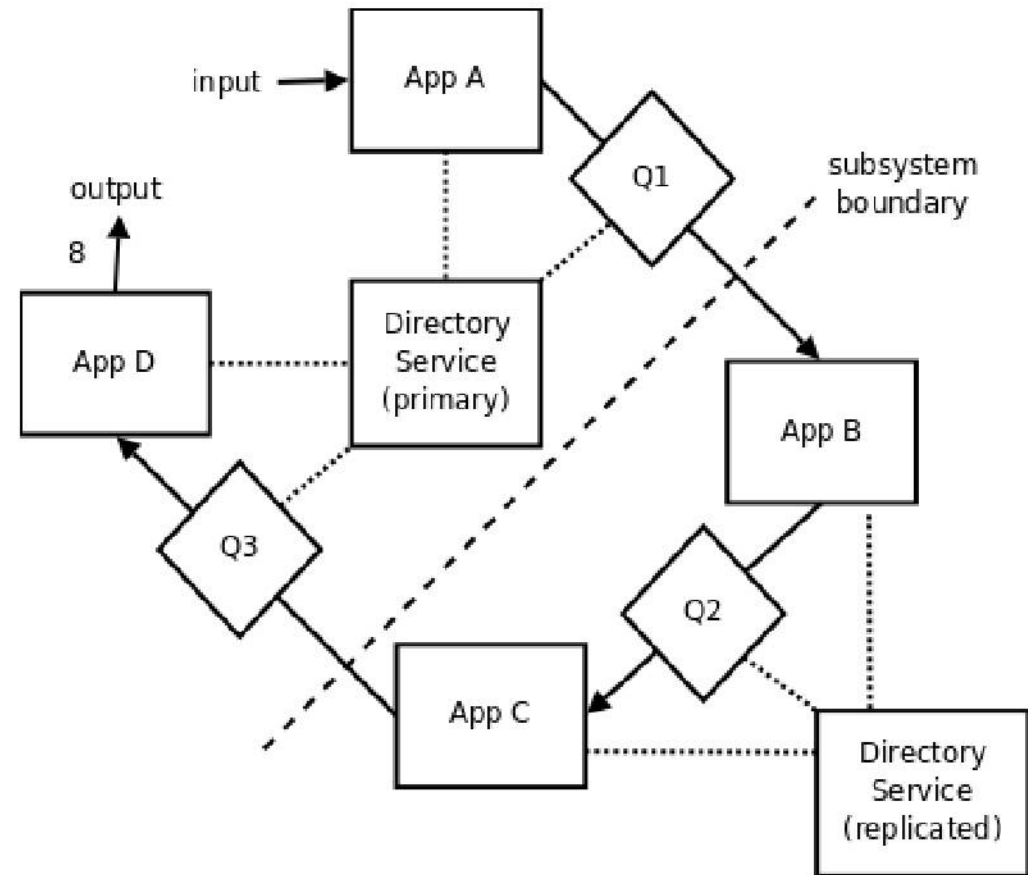
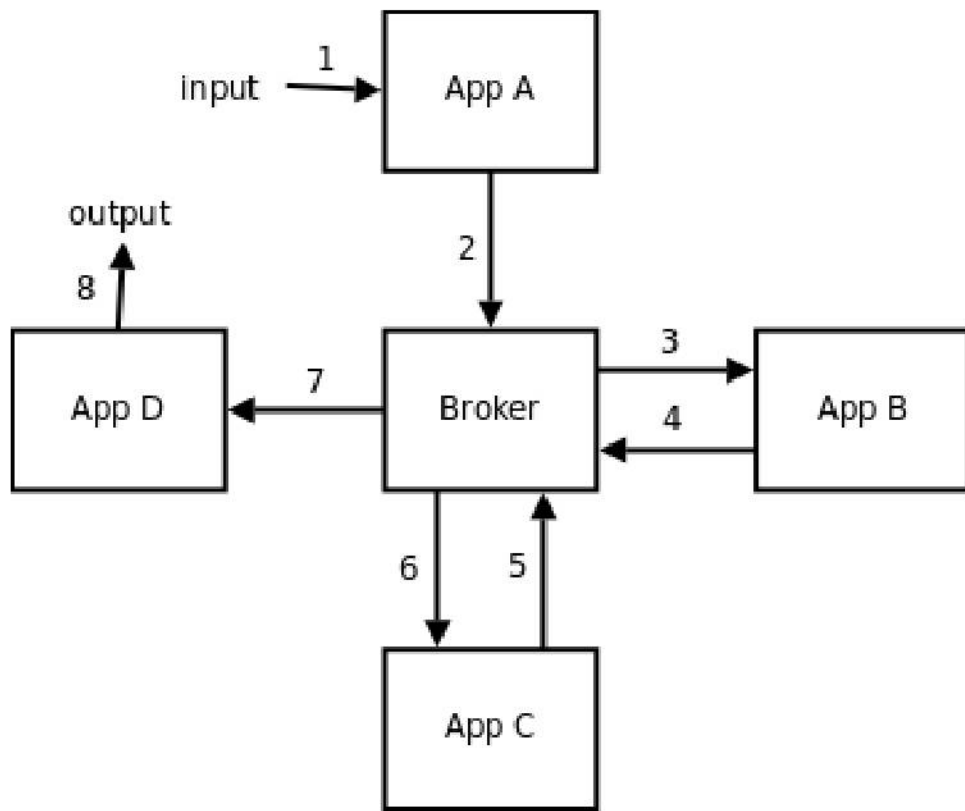
IMatix выходит из рабочей группы AMQP

«...normal TCP socket, inject it with a mix of radioactive isotopes stolen from a secret Soviet atomic research project, bombard it with 1950-era cosmic rays, and put it into the hands of a drug-addled comic book author with a badly-disguised fetish for bulging muscles clad in spandex»

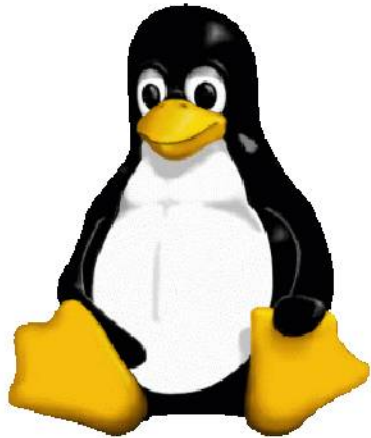
Pieter Hintjens



# 0MQ: Broker centric VS Brokerless











## Клиентик

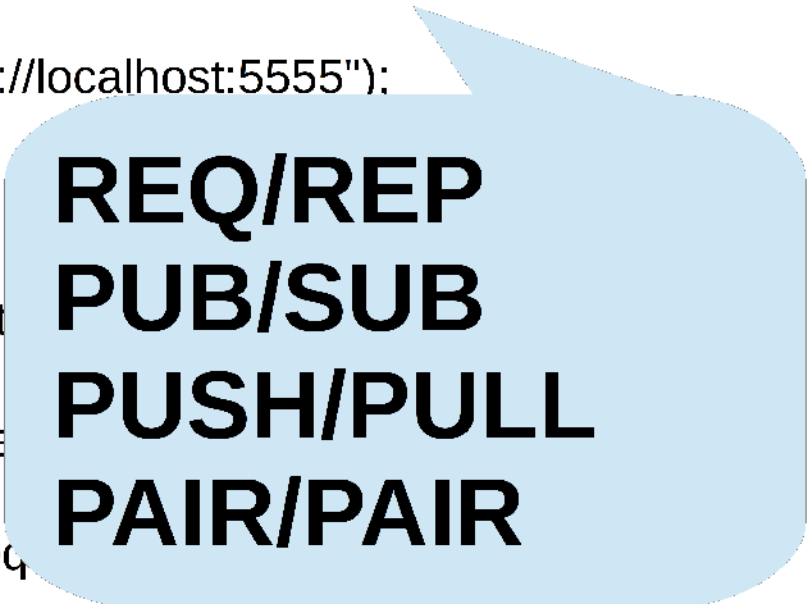
- 1 `void *context = zmq_ctx_new ();`
- 2 `void *requester = zmq_socket (context, ZMQ_REQ);`
- 3 `zmq_connect (requester, "tcp://localhost:5555");`
- 4 `zmq_msg_t request;`
- 5 `zmq_msg_init_size (&request, 5);`
- 6 `memcpy (zmq_msg_data (&request), "Hello", 5);`
- 7 `zmq_msg_send (&request, requester, 0);`
- 8 `zmq_msg_close (&request);`

## Серверчик

- 1 `void *context = zmq_ctx_new ();`
- 2 `void *responder = zmq_socket (context, ZMQ_REP);`
- 3 `zmq_bind (responder, "tcp://*:5555");`
- 4 `zmq_msg_t request;`
- 5 `zmq_msg_init (&request);`
- 6 `zmq_msg_rcv (&request, responder, 0);`
- 7 `zmq_msg_close (&request);`



- 1 `void *context = zmq_ctx_new ();`
- 2 `void *requester = zmq_socket (context, ZMQ_REQ);`
- 3 `zmq_connect (requester, "tcp://localhost:5555");`
- 4 `zmq_msg_t request;`
- 5 `zmq_msg_init_size (&request, ...);`
- 6 `memcpy (zmq_msg_data (&request), ...);`
- 7 `zmq_msg_send (&request, requester, 0);`
- 8 `zmq_msg_close (&request);`



**REQ/REP**  
**PUB/SUB**  
**PUSH/PULL**  
**PAIR/PAIR**





# 0MQ: PUSH/PULL









The  
Looney  
Tunes  
Show™



## TCP

```
zmq_bind(socket, "tcp://*:5555");  
zmq_bind(socket, "tcp://127.0.0.1:5555");  
zmq_bind(socket, "tcp://eth0:5555");  
  
zmq_connect(socket, "tcp://server1:5555");
```

## PGM/EPGM

```
zmq_bind(socket, "epgm://eth0;228.1.2.3:5555");  
  
zmq_connect(socket, "epgm://eth0;239.192.1.1:5555");  
zmq_connect(socket, "pgm://192.168.1.1;239.192.1.1:5555");
```



## IPC

```
zmq_bind(socket, "ipc:///tmp/feeds/0");
```

```
zmq_connect(socket, "ipc:///tmp/feeds/0");
```



## INPROC

```
zmq_bind(socket, "inproc://my-endpoint");
```

```
zmq_connect(socket, "inproc://my-endpoint");
```

## 0MQ: Poll

```
1 zmq_bind(socket_1, «tcp://*:5555»);  
2 zmq_bind(socket_2, «ipc:///tmp/local»);  
3 zmq_bind (socket_3, "inproc://contact_point");  
4 zmq_pollitem_t items[] = {  
5     {socket_1, 0, ZMQ_POLLIN, 0},  
6     {socket_2, 0, ZMQ_POLLIN, 0},  
7     {socket_3, 0, ZMQ_POLLIN, 0}  
8 };
```



## 0MQ: Poll(продолжение)

9

```
while(working) {
```

10

```
    zmq_poll(items, 3, -1);
```

11

```
    if(items[0].revents & ZMQ_POLLIN) {
```

12

```
        //Что-то есть для первого сокета
```

13

```
    } else if (items[1].revents & ZMQ_POLLIN) {
```

14

```
        //Второй сокет так же получил сообщение
```

15

```
    } else if (items[2].revents & ZMQ_POLLIN) {
```

16

```
        //Третий не скучает
```

17

```
    }
```

18

```
}
```

- **High Water Mark**
- **SWAP**
- **SUBSCRIBE**
- **RECONNECT**
- **Еще порядка 7-ми параметров**





# Вопросы

