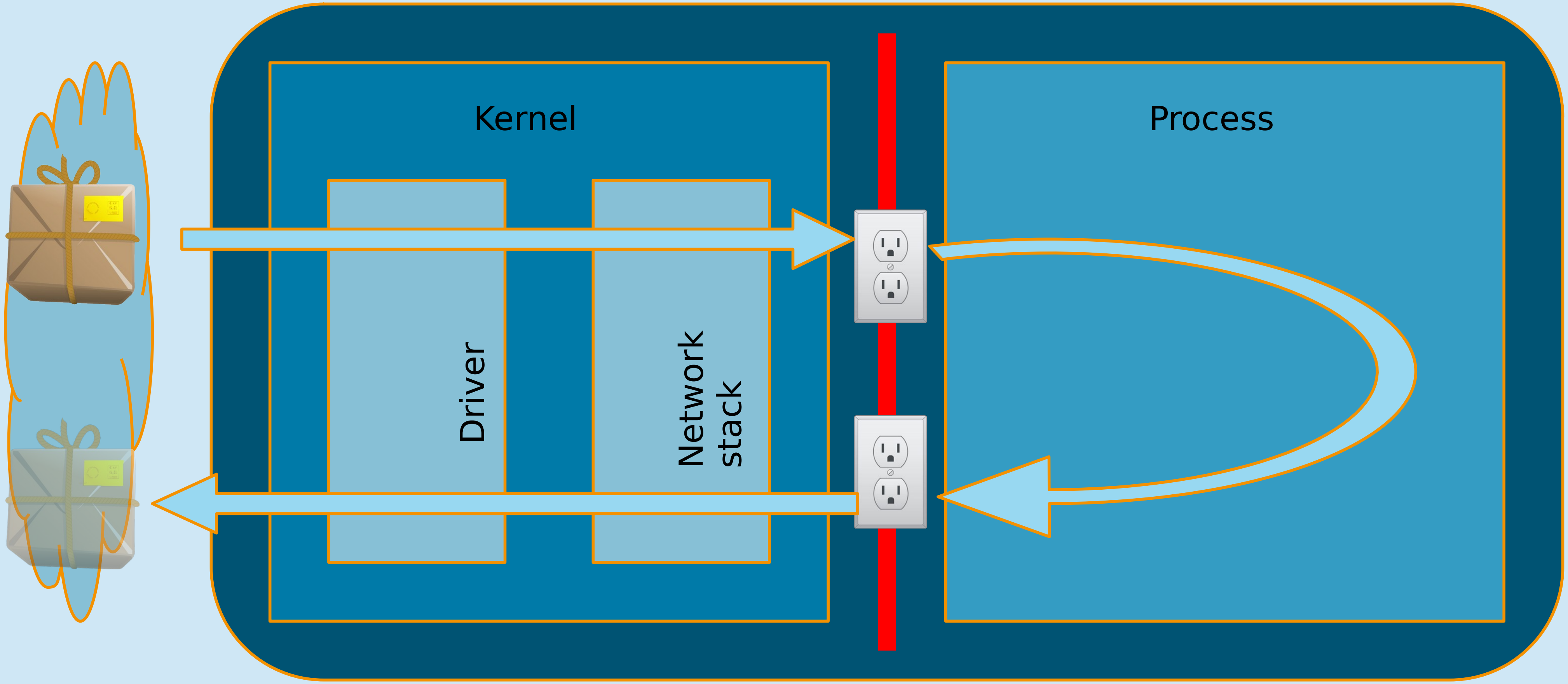
Fast kernel-level SFU with Sipwise RTPengine

Richard Fuchs
rfuchs@sipwise.com

Origins

- Need for a media proxy to solve NAT problems
- Existing solutions (10+ years ago):
  - Only basic RTP forwarding
  - Only basic SDP manipulation
  - Low performance!
    - Few 100s of calls max
    - Why?

Performance

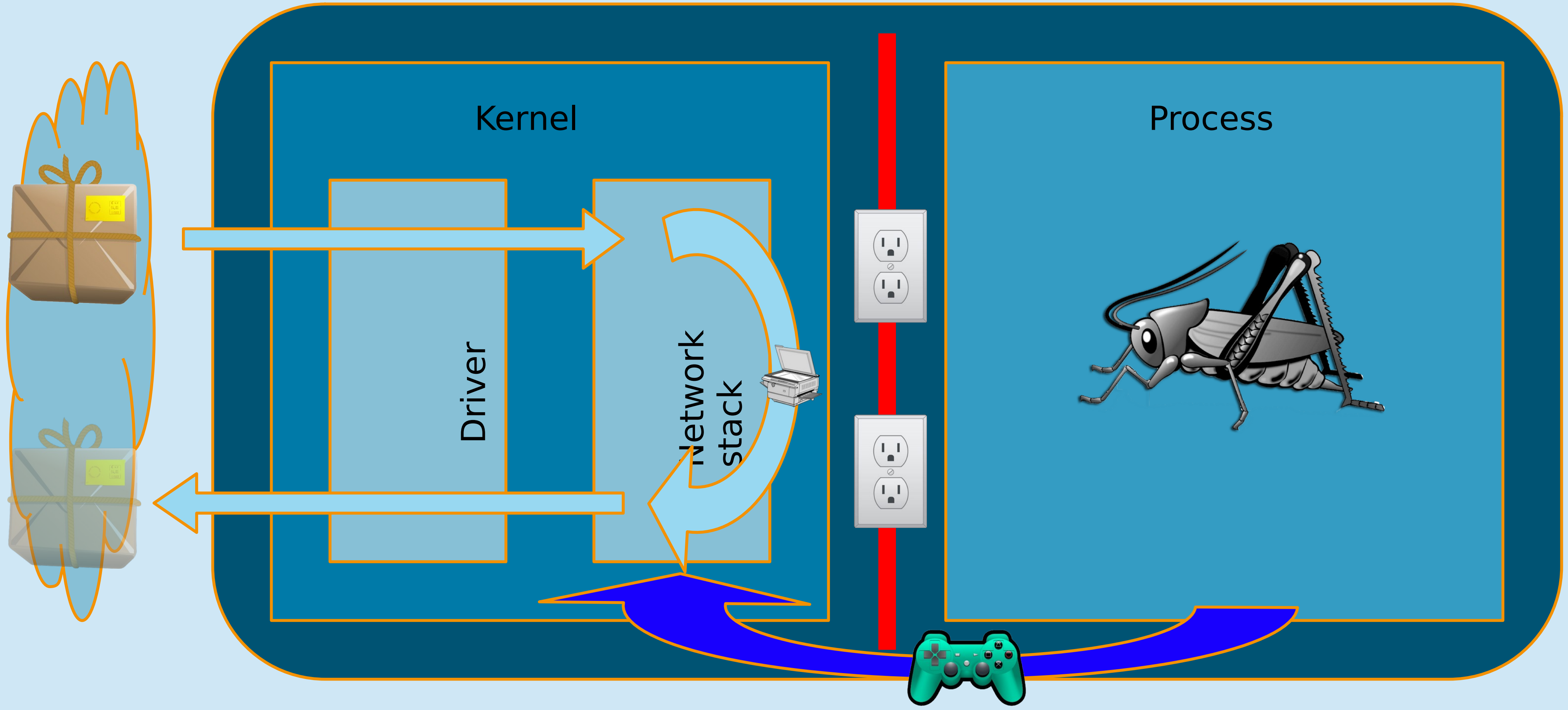


	IRQ	Kernel	Process
1	Driver RX, network stack		
2	Assign to socket queue		
3	Wake up sleeping process		
4			Resume from poll()
5			recv() from socket
6		Fetch from queue	
7		Copy contents	
8			Return from recv()
9			Determine destination
10			send() to socket
11		Copy contents	
12		Add to send queue	
13			Return from send()
14			Perhaps sleep w/ poll()
15	Driver TX		

Performance

- *At least* 4 context switches per packet
- 400+ context switches per call per second!
- Current situation: better but worse
    - Better CPUs, more cores
    - Dedicated SYSCALL instruction, vDSO
    - Meltdown, Spectre, …
- No sendfile() or splice() for datagrams
- System calls are expensive, but unavoidable
    - … or are they?

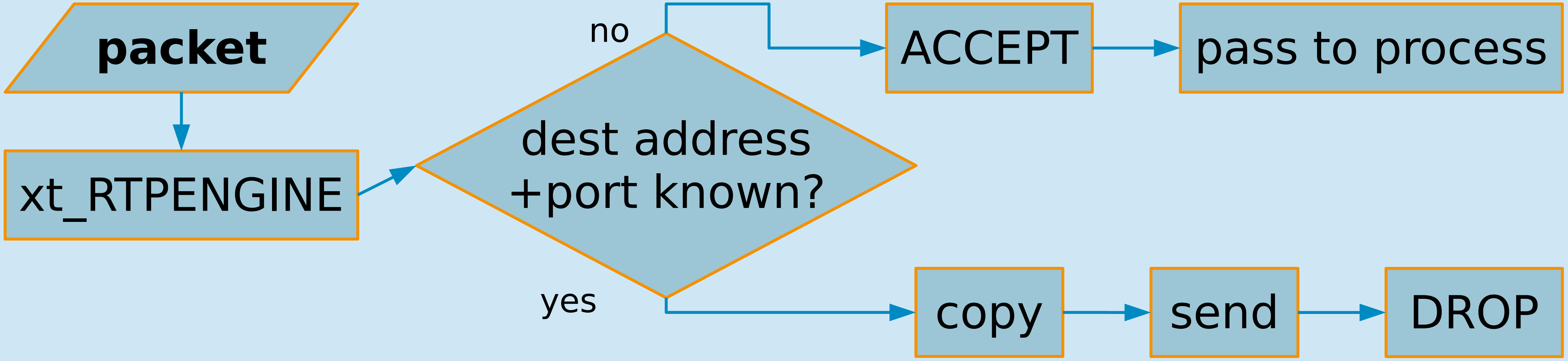
Kernel-based forwarding



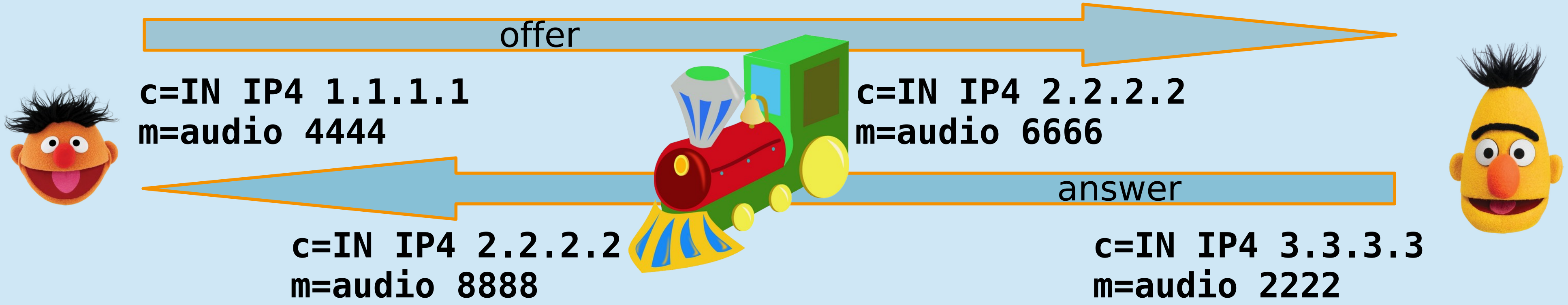
Kernel module (x_tables)

```
moose:~$ sudo lsmod | grep RTP
xt_RTPENGINE           53248  1
x_tables               61440  6 xt_conntrack,nft_compat,xt_addrtype,xt_RTPENGINE,ip_tables,xt_MASQUERADE
```

```
moose:~$ sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   83 13451 RTPENGINE  udp  --  *      *       0.0.0.0/0            0.0.0.0/0            RTPENGINE id:0
```

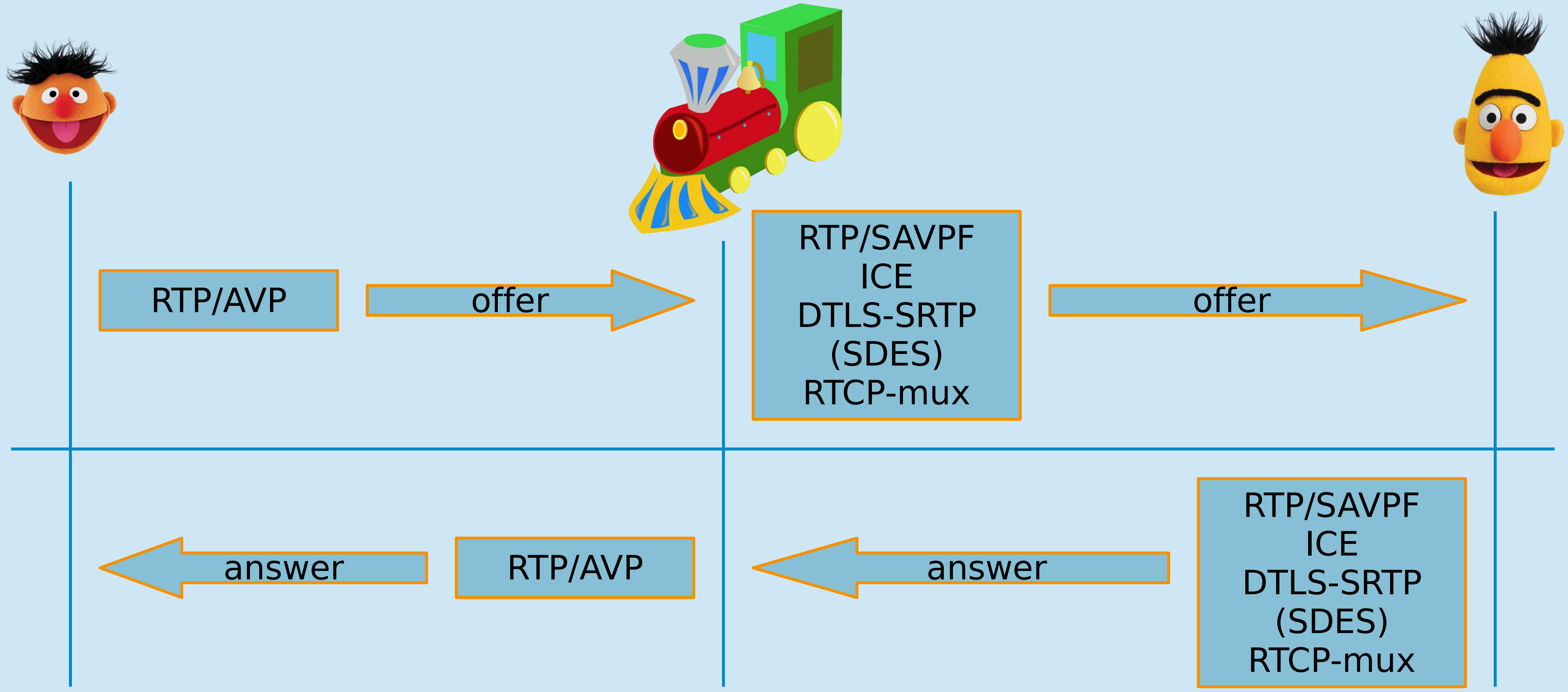


Signalling: Offer/answer

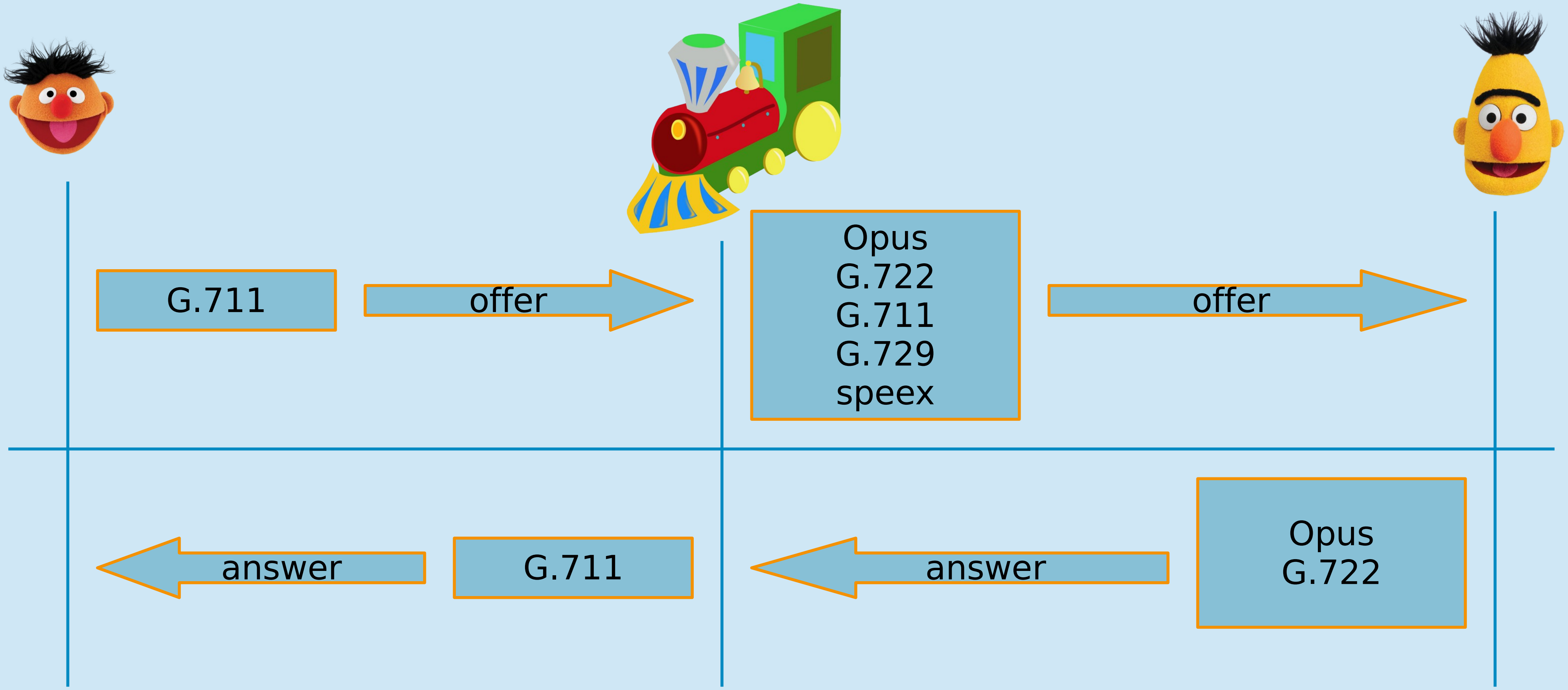


- SIP-agnostic
- Operates on entire SDP
- Multiple control protocols
  - UDP, TCP, HTTP, HTTPS, WS, WSS
  - Binary format or JSON

Breaking offer/answer

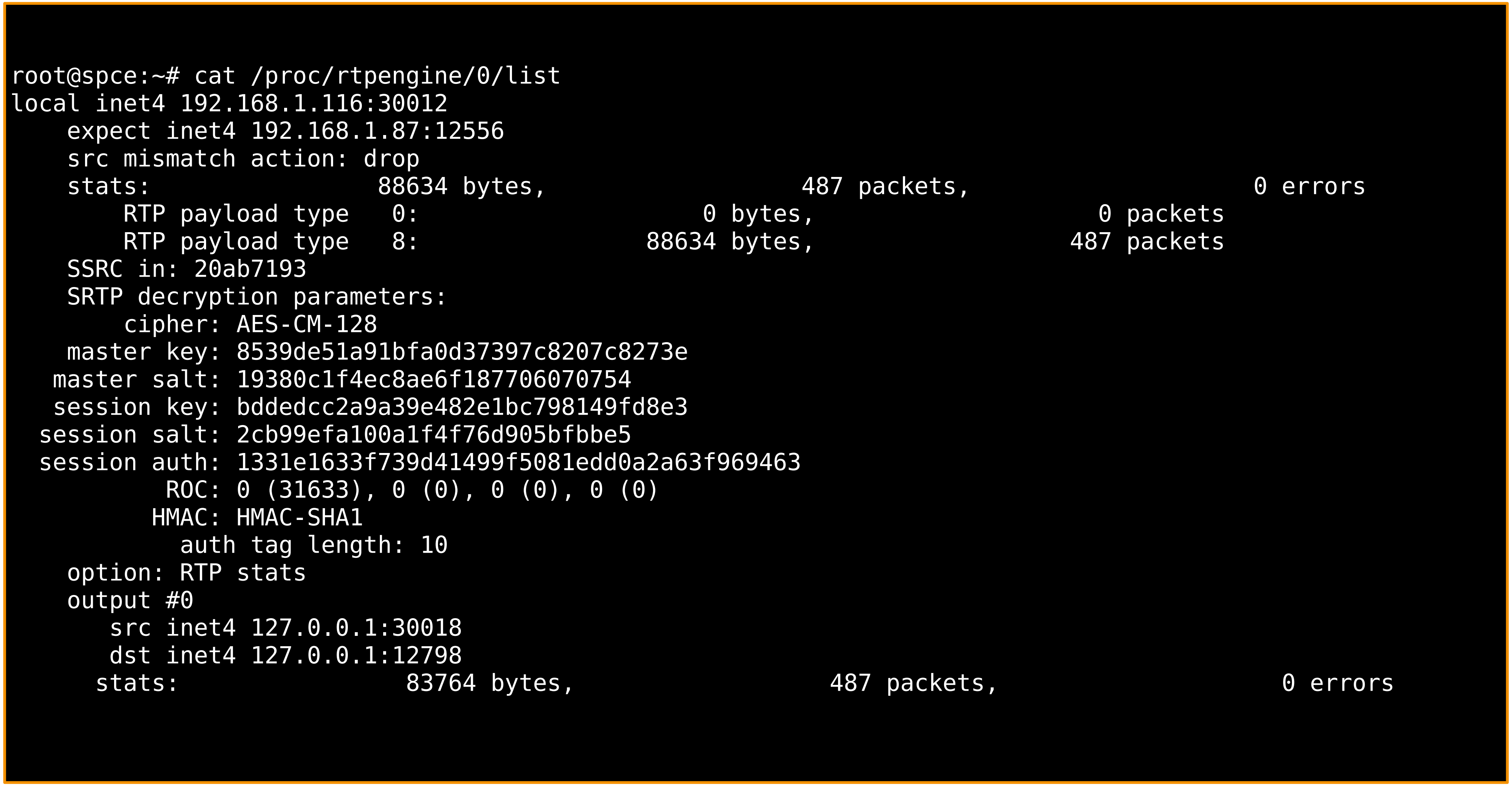


Breaking offer/answer



Kernel support

- RTP manipulation supported by kernel module:
  - SRTP   RTP, SRTP   SRTP (DTLS or SDES)
    - AES-CM
    - AES-GCM (AEAD)
    - AES-F8
    - 128 – 256 bits
  - Media blocking
  - Very limited media silencing (specific codecs only)
- NOT supported by kernel module:
  - Any sort of transcoding

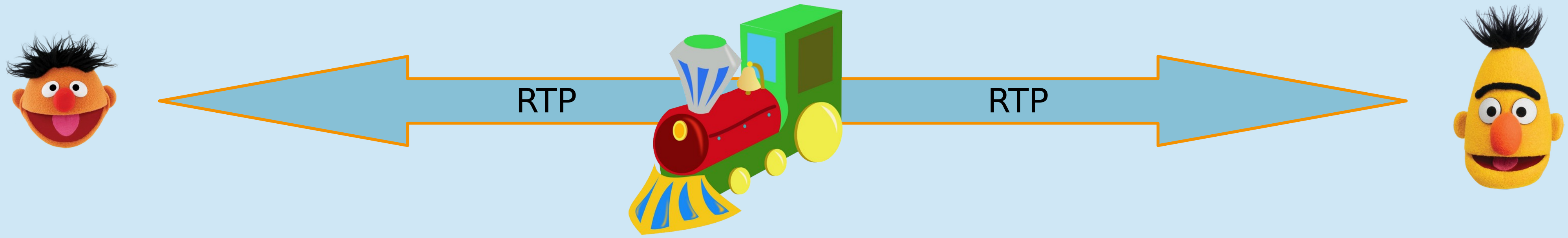


```
root@spce:~# cat /proc/rtpengine/0/list
local inet4 192.168.1.116:30012
    expect inet4 192.168.1.87:12556
    src mismatch action: drop
    stats:               88634 bytes,                   487 packets,                   0 errors
        RTP payload type   0:                0 bytes,                 0 packets
        RTP payload type   8:            88634 bytes,               487 packets
    SSRC in: 20ab7193
    SRTP decryption parameters:
        cipher: AES-CM-128
    master key: 8539de51a91bfa0d37397c8207c8273e
   master salt: 19380c1f4ec8ae6f187706070754
   session key: bddedcc2a9a39e482e1bc798149fd8e3
  session salt: 2cb99efa100a1f4f76d905bfbbe5
  session auth: 1331e1633f739d41499f5081edd0a2a63f969463
           ROC: 0 (31633), 0 (0), 0 (0), 0 (0)
          HMAC: HMAC-SHA1
            auth tag length: 10
    option: RTP stats
    output #0
       src inet4 127.0.0.1:30018
       dst inet4 127.0.0.1:12798
      stats:               83764 bytes,                   487 packets,                   0 errors
```

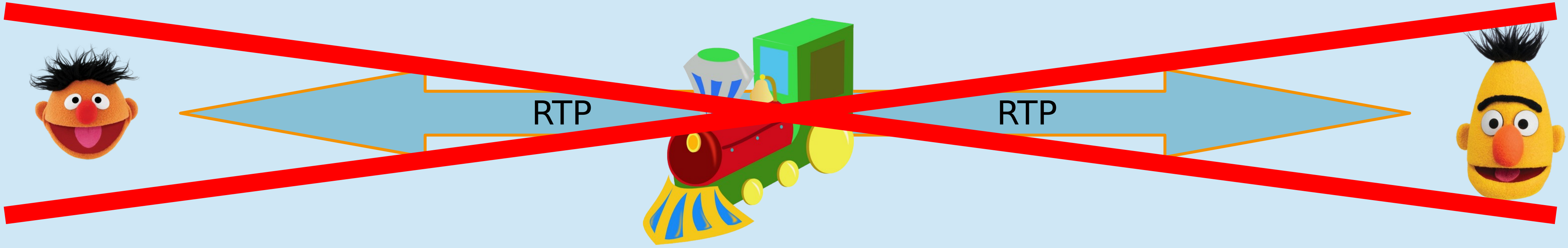
SIPREC & Conferencing

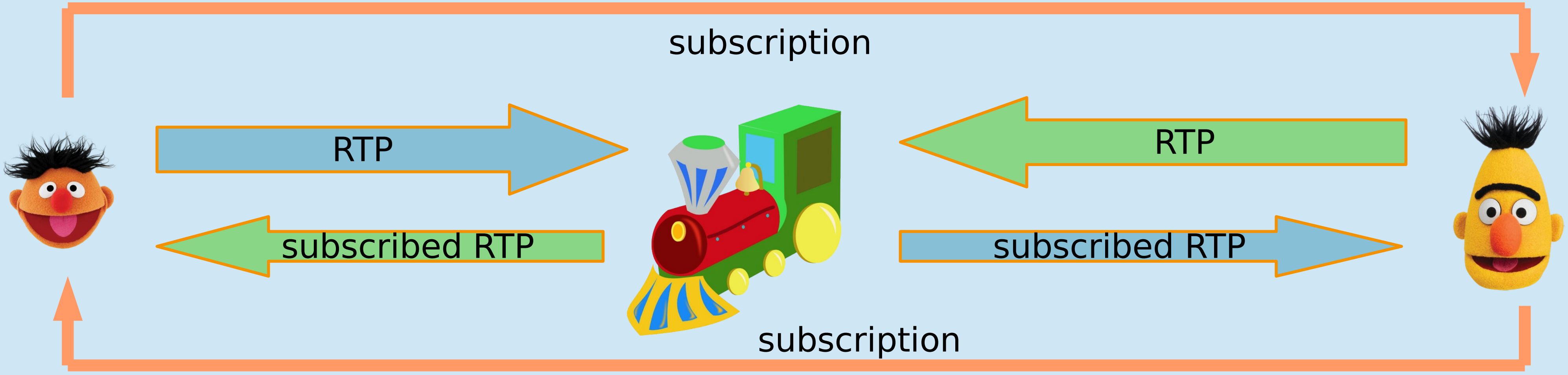
- SIPREC:
  - Fork media flow to external server (SRS)
  - Offer/answer based signalling to SRS
  - Signalling using SDP
- Conferencing:
  - SFU
  - Media forking to multiple destinations
  - Signalling using SDP
- Existing solutions?

Breaking offer/answer (again)



Breaking offer/answer (again)

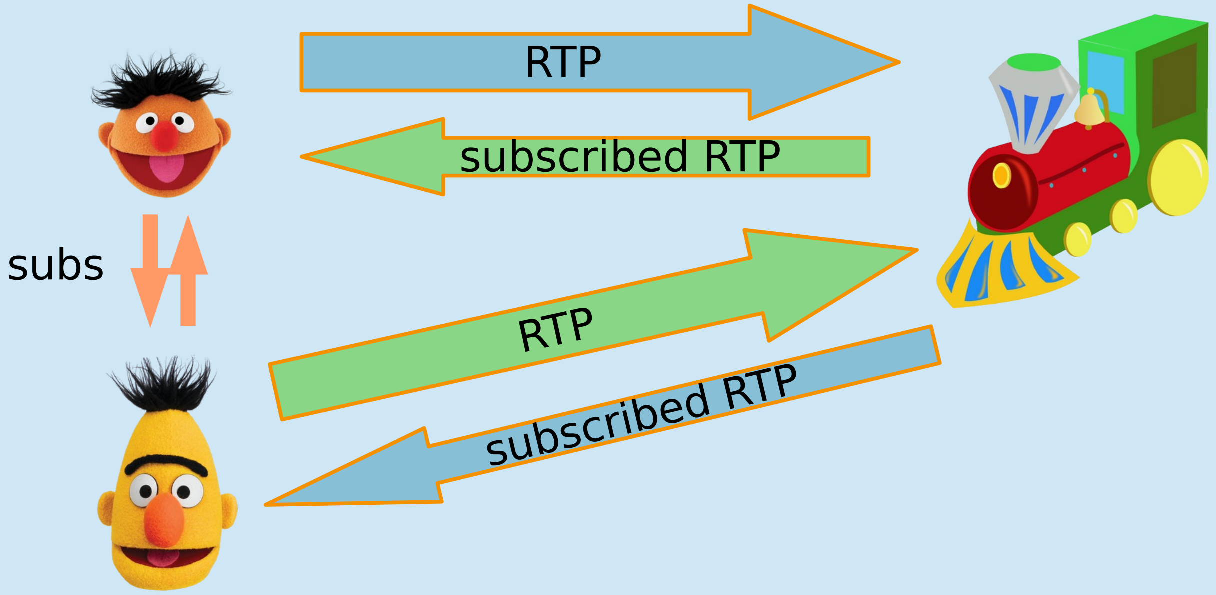




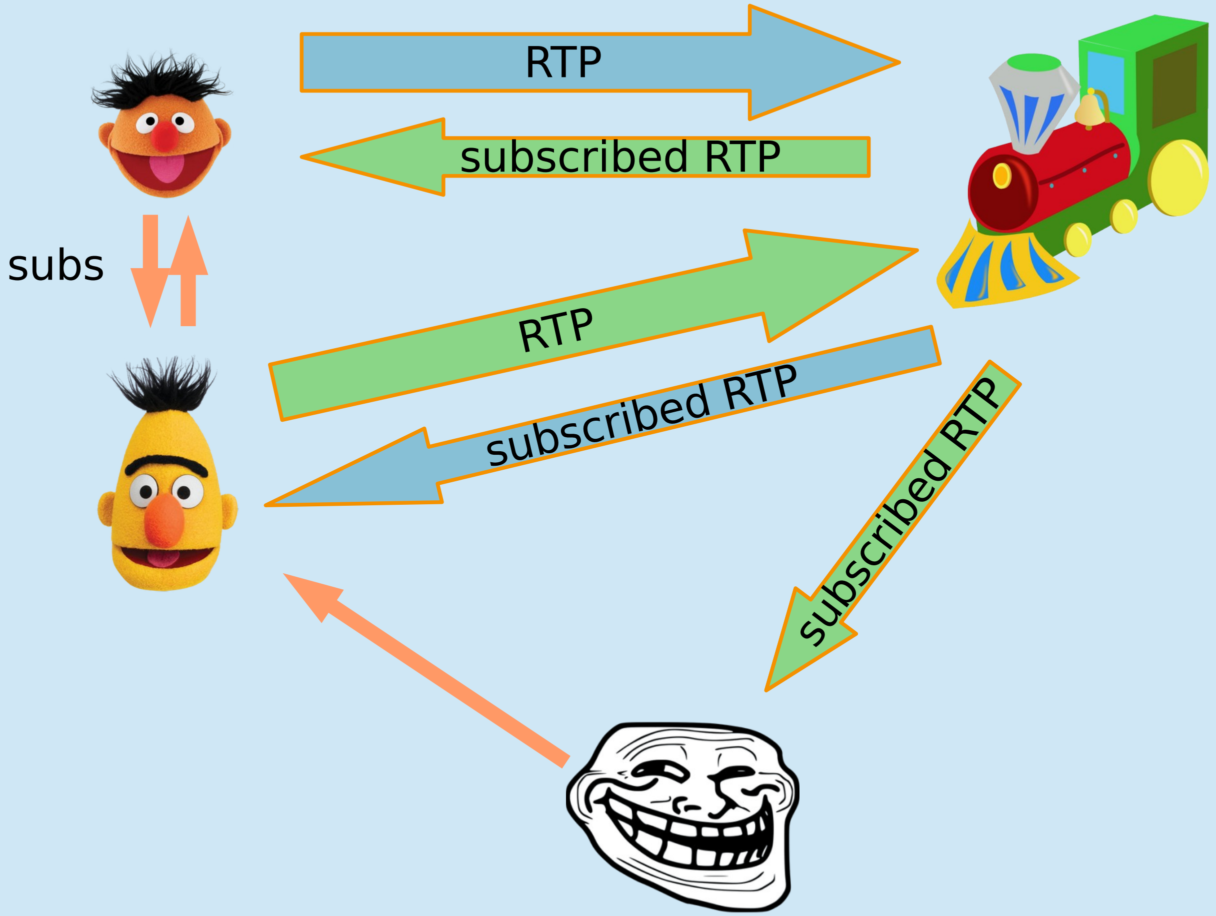
1-to-n RTP forwarding

- Eliminate 1-to-1 media mapping
- Implement 1-to-n media mapping
- Changes internal only
    - Retain offer/answer model
    - n=1 for offer/answer sessions
    - Retain all existing features (SRTP, transcoding, etc)
    - Retain kernel module capabilities
    - Additional methods for n>1

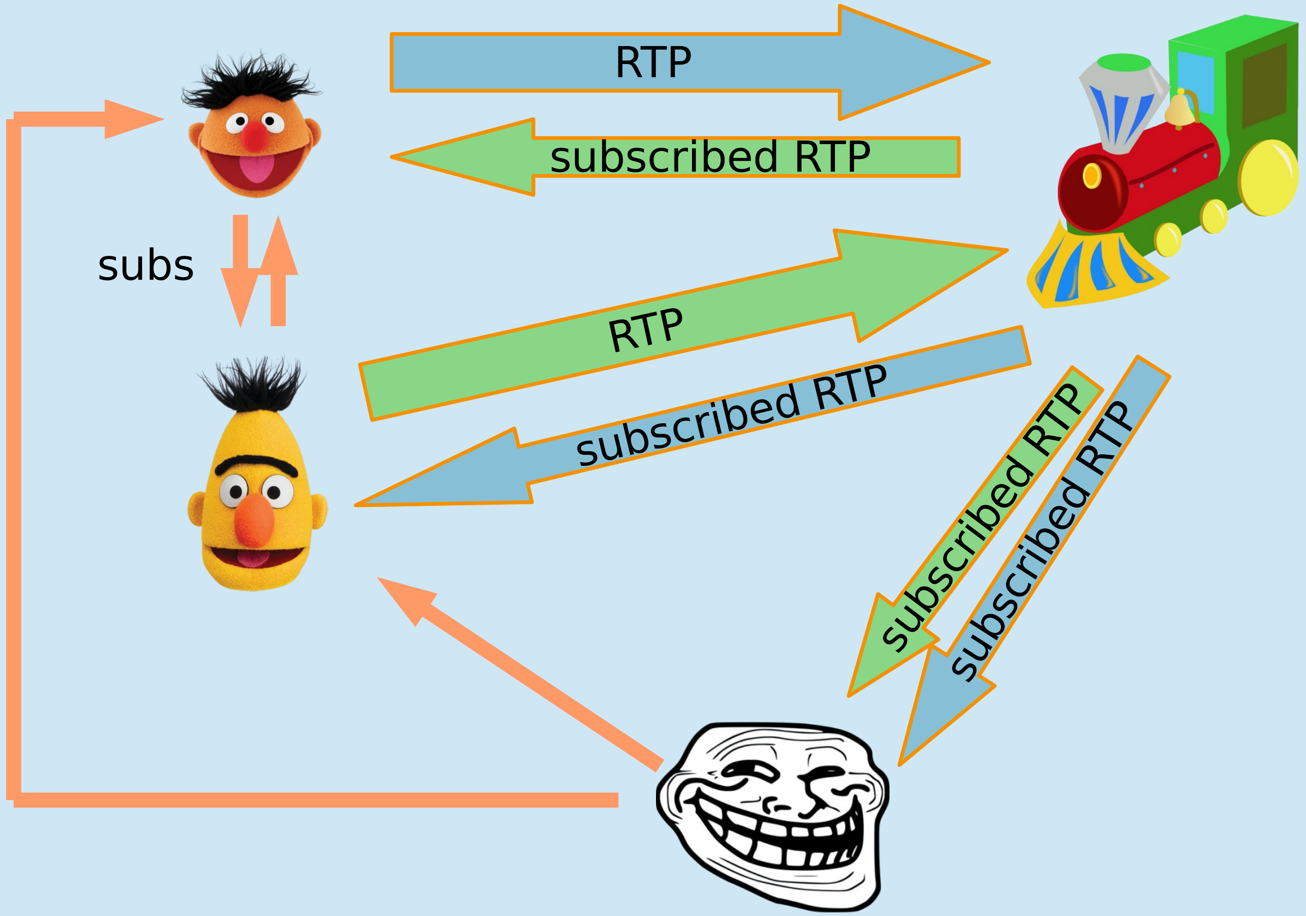
SIPREC (or LI, etc)



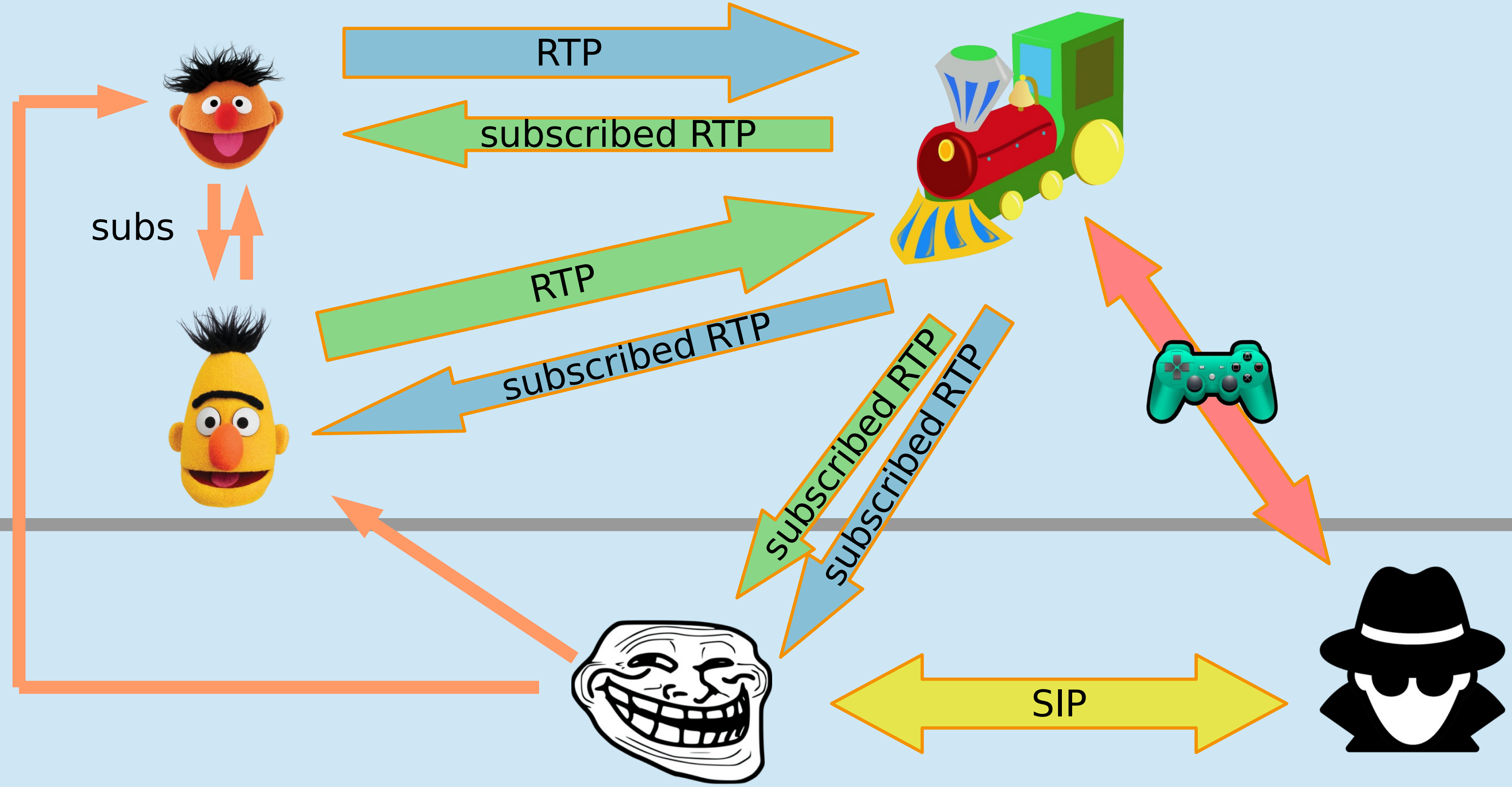
SIPREC (or LI, etc)



SIPREC (or LI, etc)



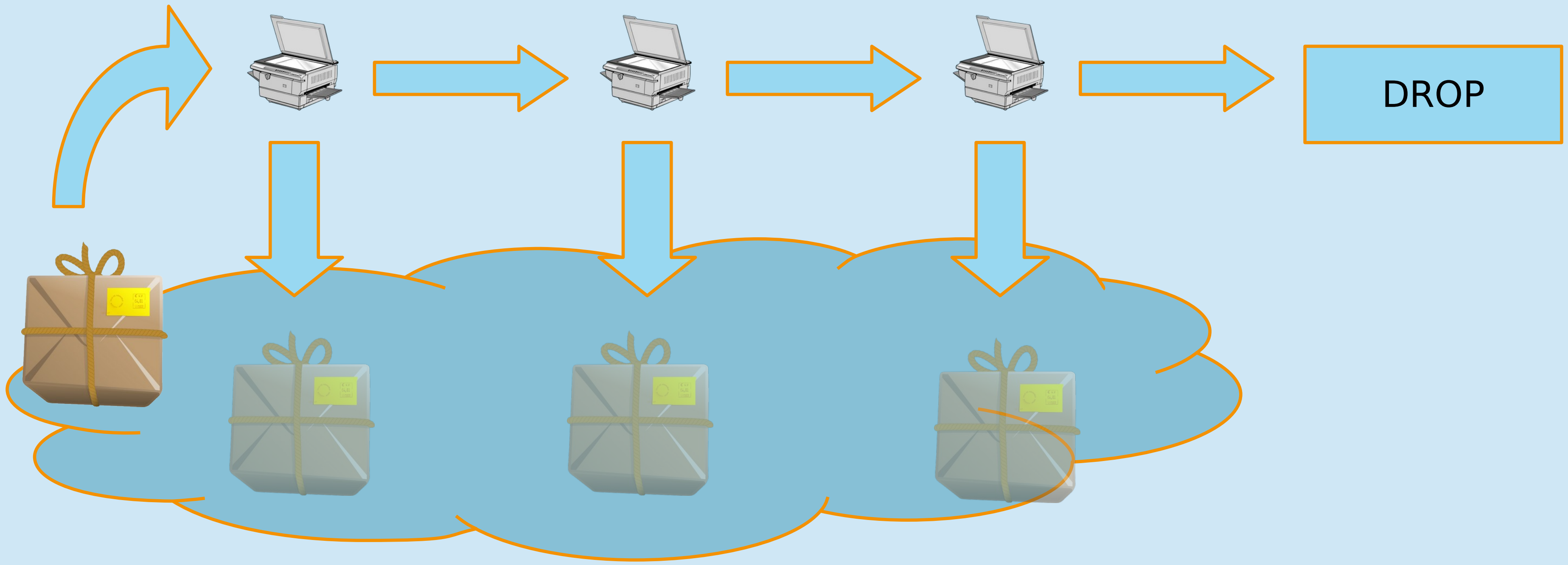
SIPREC


RTP
subscribed RTP
subs
RTP
subscribed RTP
subscribed RTP
subscribed RTP
SIP

New methods

- „subscribe request“
  - Produces „sendonly“ offer SDP from RTPengine
  - All SDP manipulations possible
  - Allows for multiple subscriptions
- „subscribe answer“
  - Signals answer SDP („recvonly“) back to RTPengine
  - Selects codec for transcoding if applicable
  - Establishes flow of forked media
- „unsubscribe“

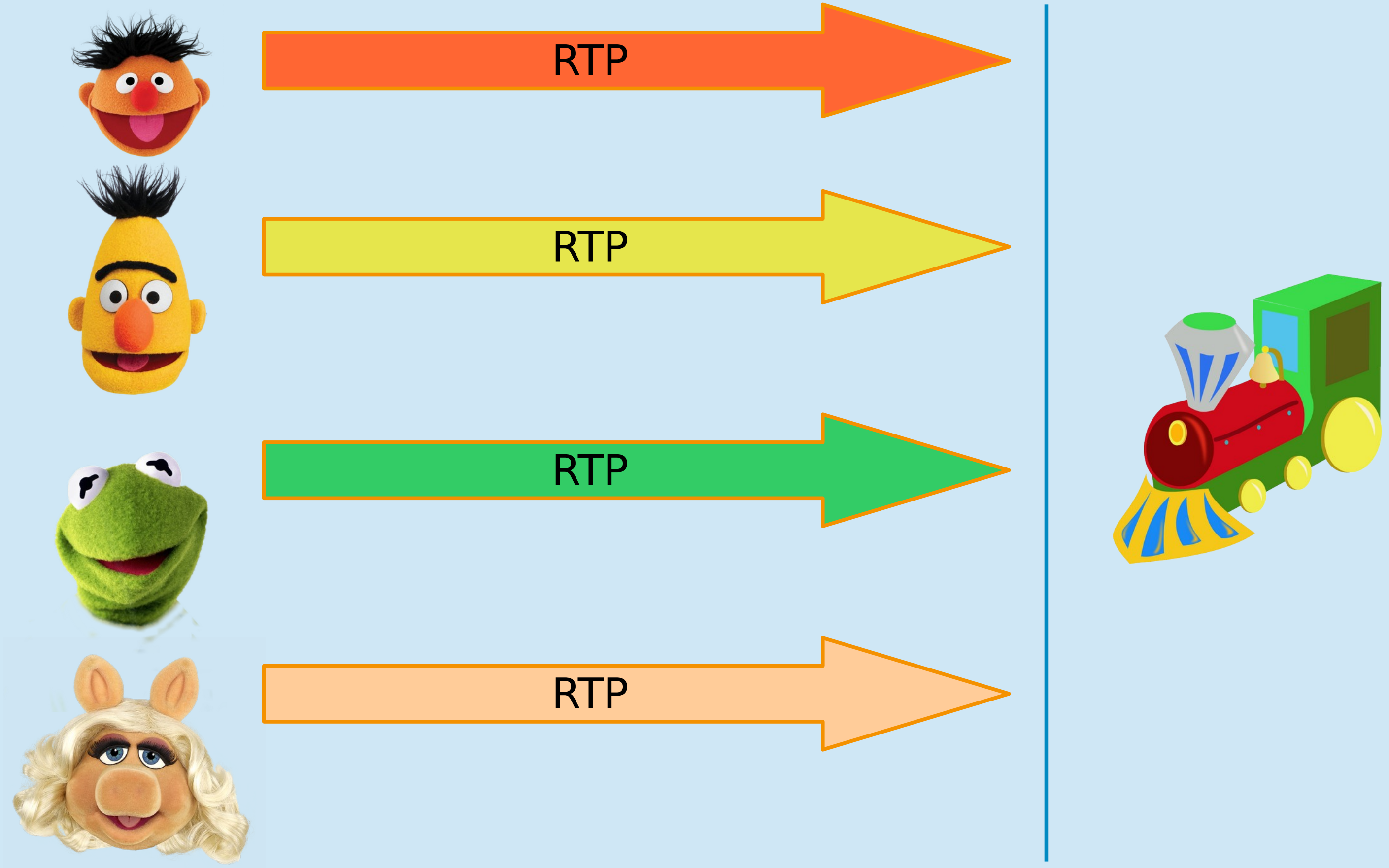
Kernel 1-to-n forwarding



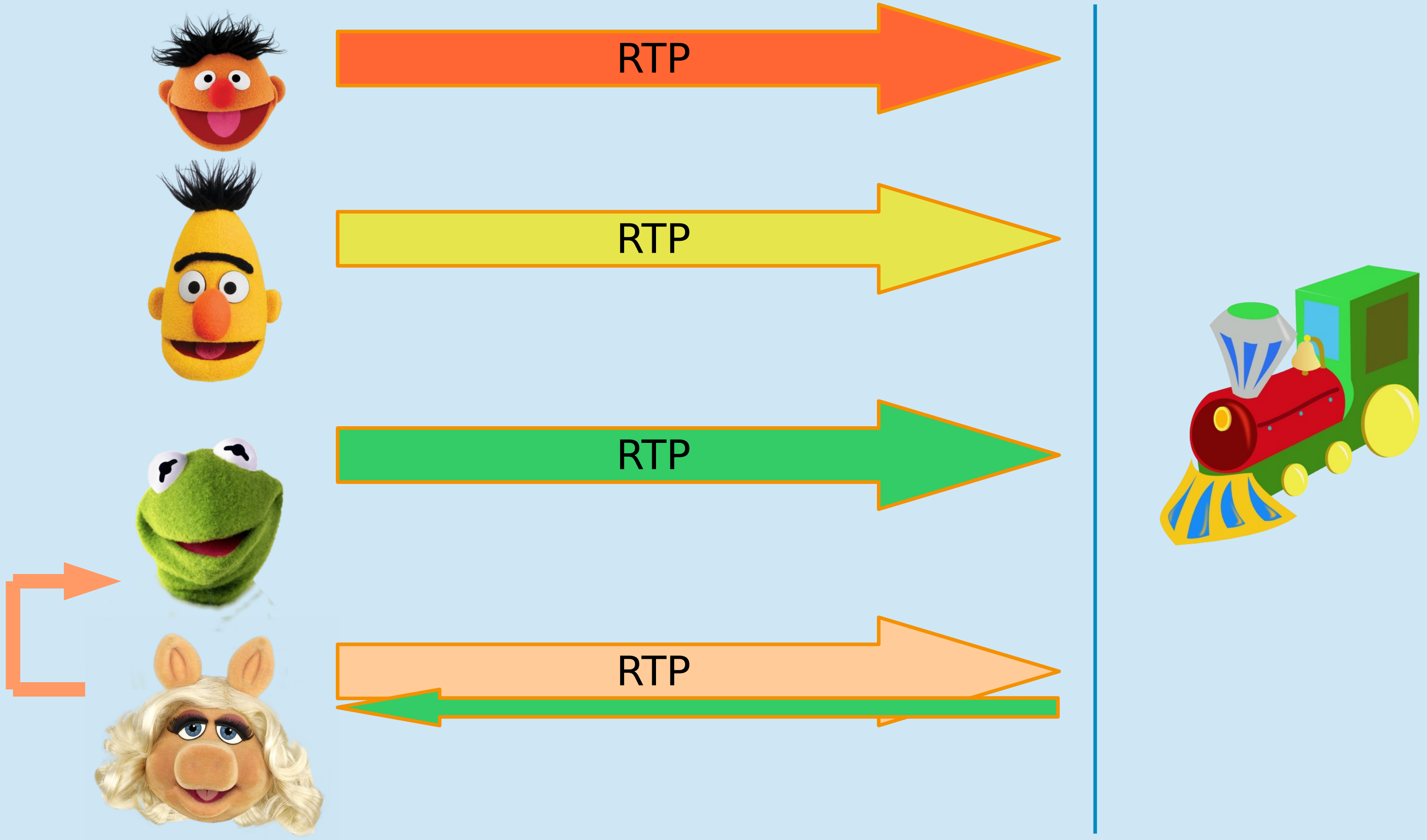
Conferencing (SFU)

- Introduce „publish“ method
- Offer/answer directly against RTPengine
  - „publish“ = offer SDP to RTPengine
  - Response = answer SDP from RTPengine
- Publish „sendonly“ streams
- Codec manipulation
  - Accepts one codec only
- Creates no subscriptions by itself
  - Use „subscribe request“ to receive media

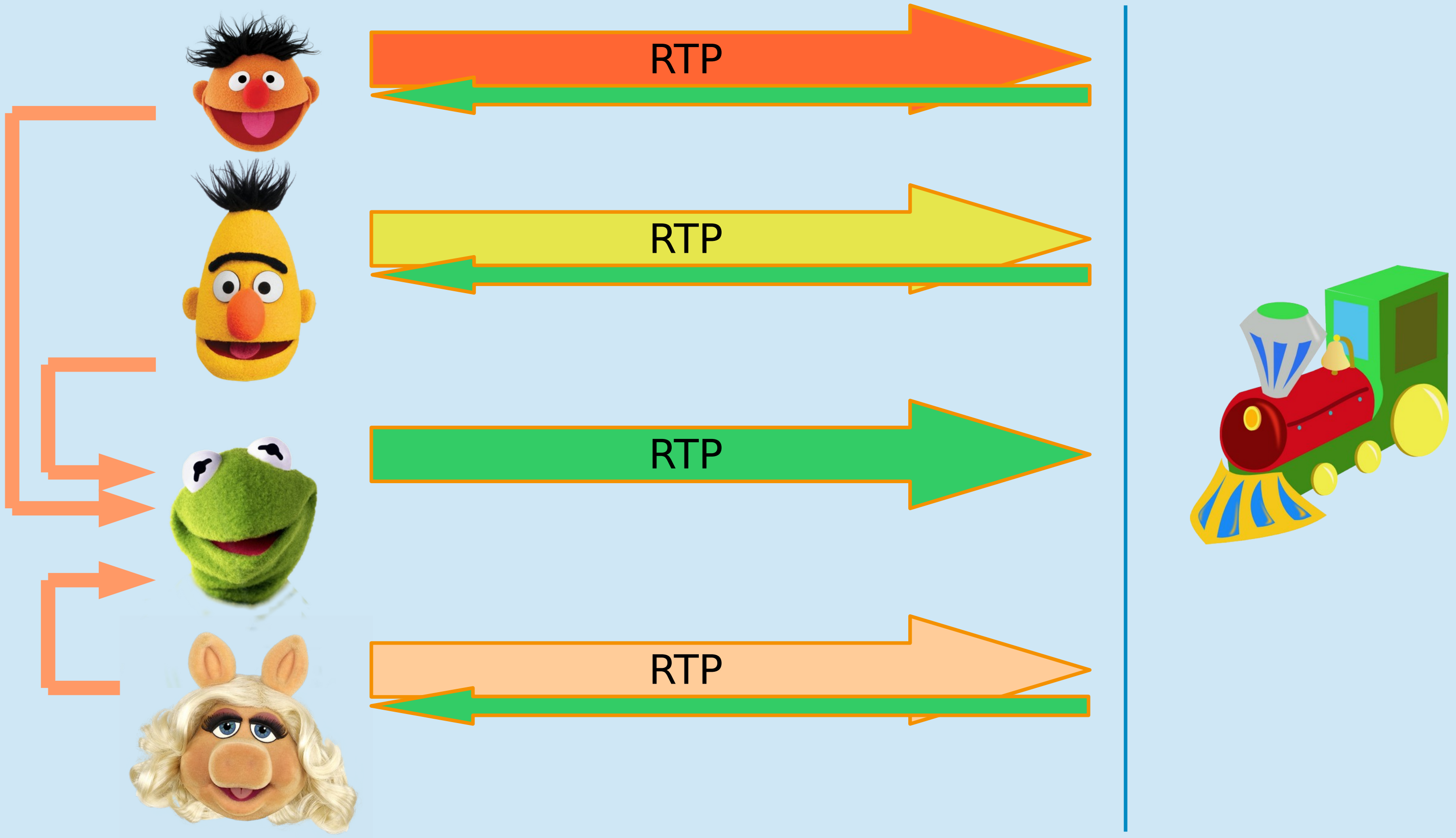
Conferencing (SFU)



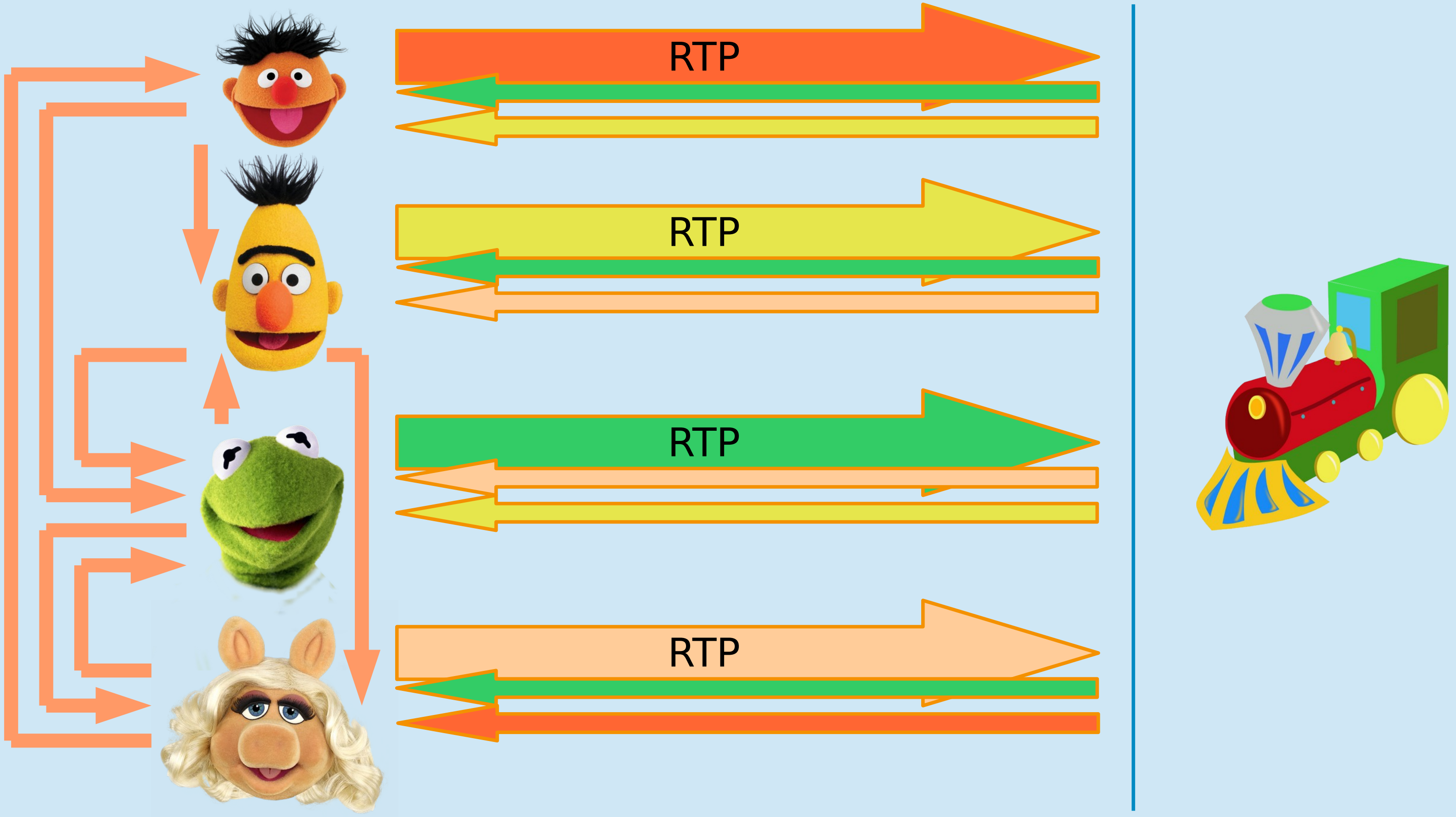
Conferencing (SFU)



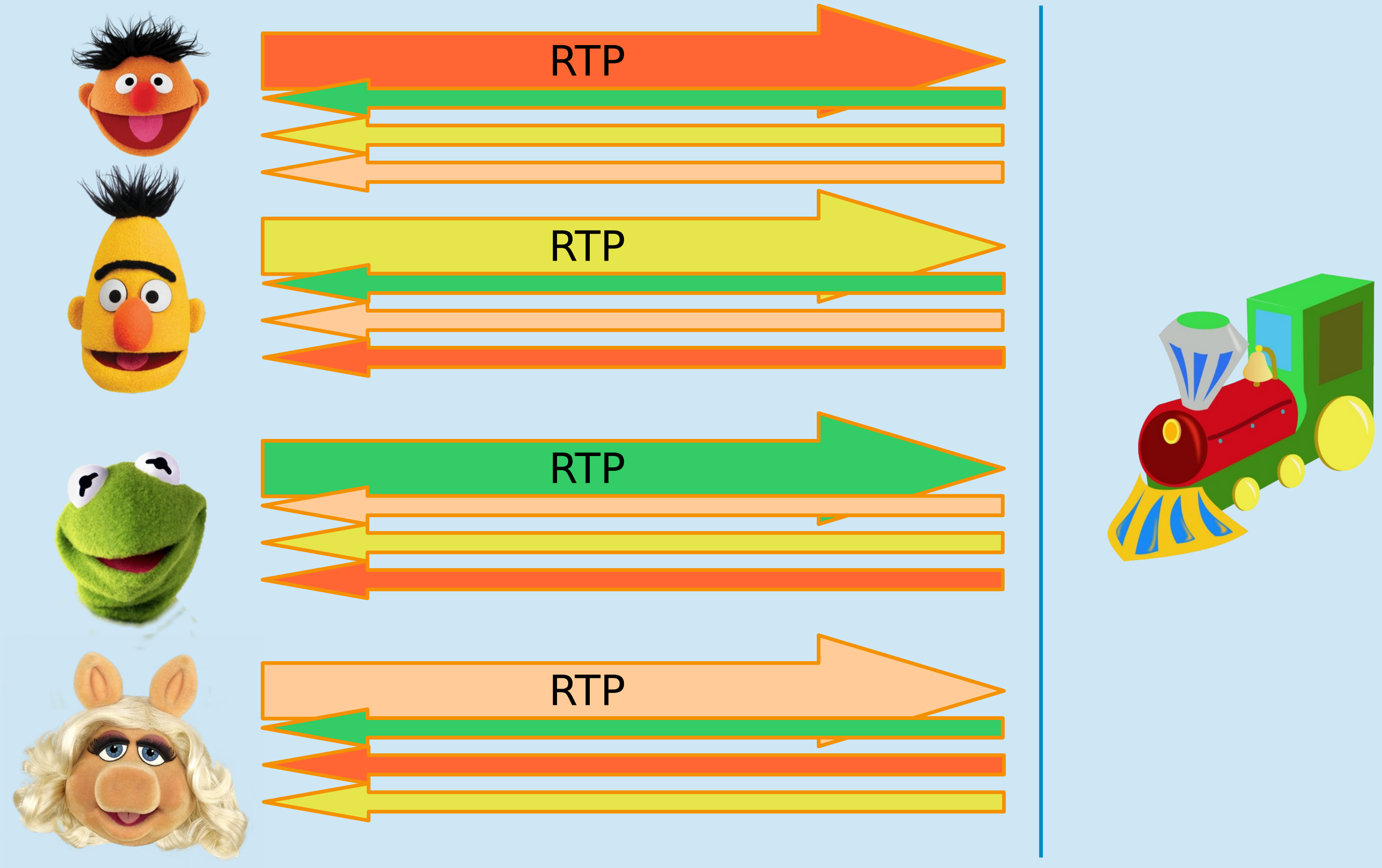
Conferencing (SFU)



Conferencing (SFU)



Conferencing (SFU)



Conferencing (SFU)

- All SDP manipulations supported
- All media manipulations supported
- Full support by the kernel module
  - Including mix-and-match RTP   SRTP
- Can be mixed with offer/answer

```
{
    "command": "publish",
    "call-id": "room 1",
    "from-tag": "Bert",
    "sdp": "v=0
            o=x ...
            s=-
            c=IN IP4 192.168.1.87
            t=0 0
            m=audio 49120 RTP/AVP 96 8 0
            a=sendonly
            a=rtpmap:96 opus/48000
            a=rtpmap:8 PCMA/8000
            a=rtpmap:0 PCMU/8000"
}
```

```
{
    "result": "ok",
    "sdp": "v=0
            o=- ...
            s=rtpengine-11-2-0-0-0
            t=0 0
            m=audio 30026 RTP/AVP 96
            c=IN IP4 192.168.1.116
            a=rtpmap:96 opus/48000
            a=recvonly
            a=rtcp:30027"
}
```

```
{
    "command": "publish",
    "call-id": "room 1",
    "from-tag": "Ernie",
    "codec": {
        "accept": [
            "opus",
            "speex",
            "PCMA",
            "PCMU"
        ]
    },
    "sdp": "v=0
            o=x ...
            s=-
            c=IN IP4 192.168.1.87
            t=0 0
            m=audio 49140 RTP/SAVP 8 0 96
            a=sendonly
            a=rtpmap:8 PCMA/8000
            a=rtpmap:0 PCMU/8000
            a=rtpmap:96 opus/48000
            a=crypto:1 AES_CM_128_HMAC_...
            a=crypto:2 F8_128_HMAC_..."
}
```

```
{
    "result": "ok",
    "sdp": "v=0
            o=- ...
            s=rtpengine-11-2-0-0-0
            t=0 0
            m=audio 30056 RTP/SAVP 96
            c=IN IP4 192.168.1.116
            a=rtpmap:96 opus/48000
            a=recvonly
            a=rtcp:30057
            a=crypto:1 AES_CM_128_HMAC_..."
}
```

```
{
    "command": "subscribe request",
    "call-id": "room 1",
    "from-tags": [
        "Ernie",
        "Bert"
    ],
    "transport protocol": "RTP/AVP"
}
```

```
{
    "result": "ok",
    "sdp": "v=0
            o=- ...
            s=rtpengine-11-2-0-0-0
            t=0 0
            m=audio 30160 RTP/AVP 96
            c=IN IP4 192.168.1.116
            a=rtpmap:96 opus/48000
            a=sendonly
            a=rtcp:30161
            m=audio 30178 RTP/AVP 96
            c=IN IP4 192.168.1.116
            a=rtpmap:96 opus/48000
            a=sendonly
            a=rtcp:30179",
    "from-tags": [
        "Ernie",
        "Bert"
    ],
    "to-tag": "8e1a2a7fc4d6..."
}
```

```
{
    "command": "subscribe answer",
    "call-id": "room 1",
    "to-tag": "8e1a2a7fc4d6...",
     "sdp": "v=0
         o=- 1234 1234 IN IP4 192.168.1.87
         s=-
         t=0 0
         m=audio 33344 RTP/AVP 96
         c=IN IP4 192.168.1.87
         a=rtpmap:96 opus/48000
         a=recvonly
         m=audio 33366 RTP/AVP 96
         c=IN IP4 192.168.1.87
         a=rtpmap:96 opus/48000
         a=recvonly"
}
```

```
{
    "command": "subscribe request",
    "call-id": "room 1",
    "from-tag": "Ernie",
    "transport protocol": "RTP/SAVPF",
    "ICE": "force",
    "SDES": [ "off" ],
    "rtcp-mux": [ "require" ],
    "codec": {
        "transcode": [
            "PCMA",
            "PCMU"
        ]
    }
}
```

```
{
    "result": "ok",
    "sdp": "v=0
            o=x ...
            s=-
            c=IN IP4 192.168.1.116
            t=0 0
            m=audio 30232 RTP/SAVPF 96 8 0
            a=rtpmap:96 opus/48000
            a=rtpmap:8 PCMA/8000
            a=rtpmap:0 PCMU/8000
            a=sendonly
            a=rtcp:30232
            a=rtcp-mux
            a=setup:actpass
            a=fingerprint:sha-256 05:85:...
            a=ice-ufrag:6JmwkYAC
            a=ice-pwd:FVsbtoIZrHigzph9pXKTqZY6s5
            a=candidate:u4f417Ek4y2Nk7kx 1 UDP ...",
    "from-tags": [
        "Ernie"
    ],
    "from-tag": "Ernie",
    "to-tag": "ef405123b35..."
}
```

Thank You

rfuchs@sipwise.com
https://rtpengine.com
https://sipwise.com