

# Raspberry Pi 3 + HLS + Nginx でつくる Live中継環境(IQ低め)

株式会社 新潟通信サービス  
櫻井 佑樹

<http://streaming.ncom.ad.jp>

2017年4月1日

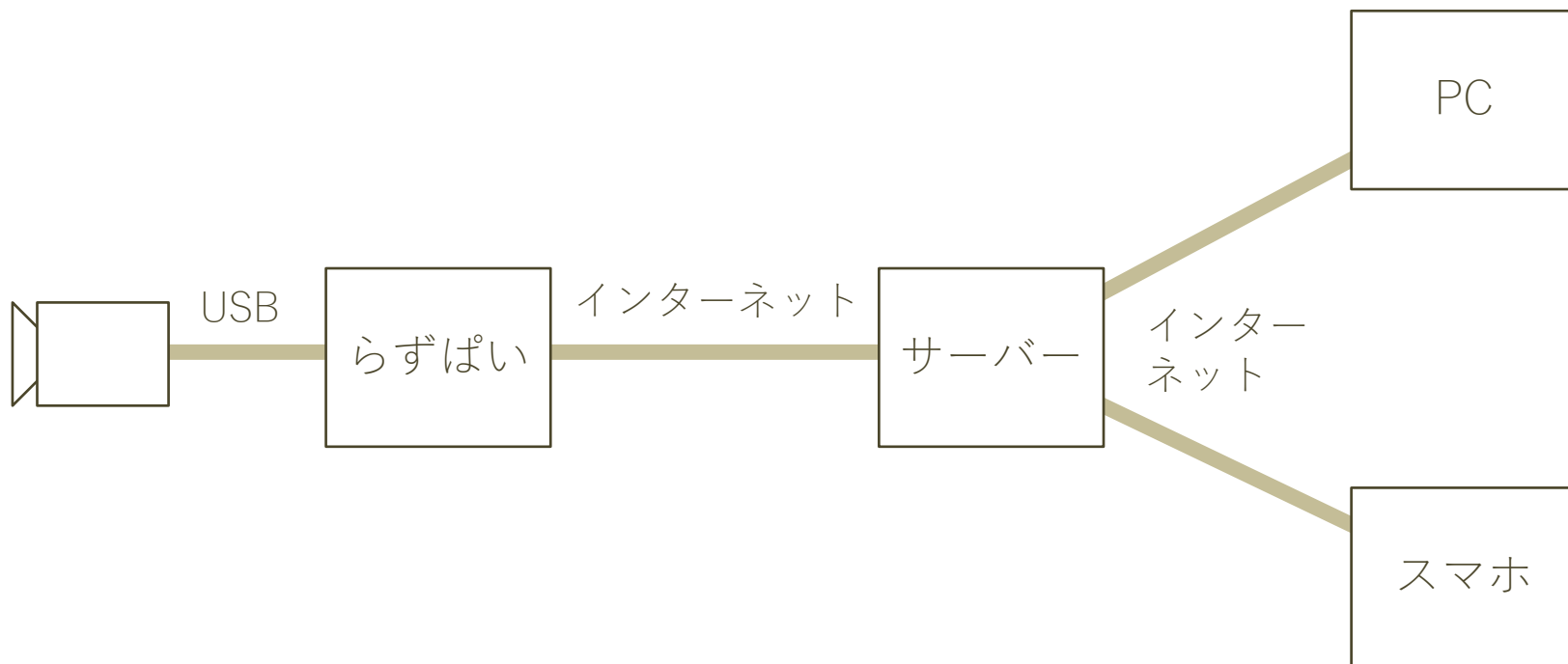
Ustream 「ほなな」

IBM Cloud Video 「よろしくニキー！」

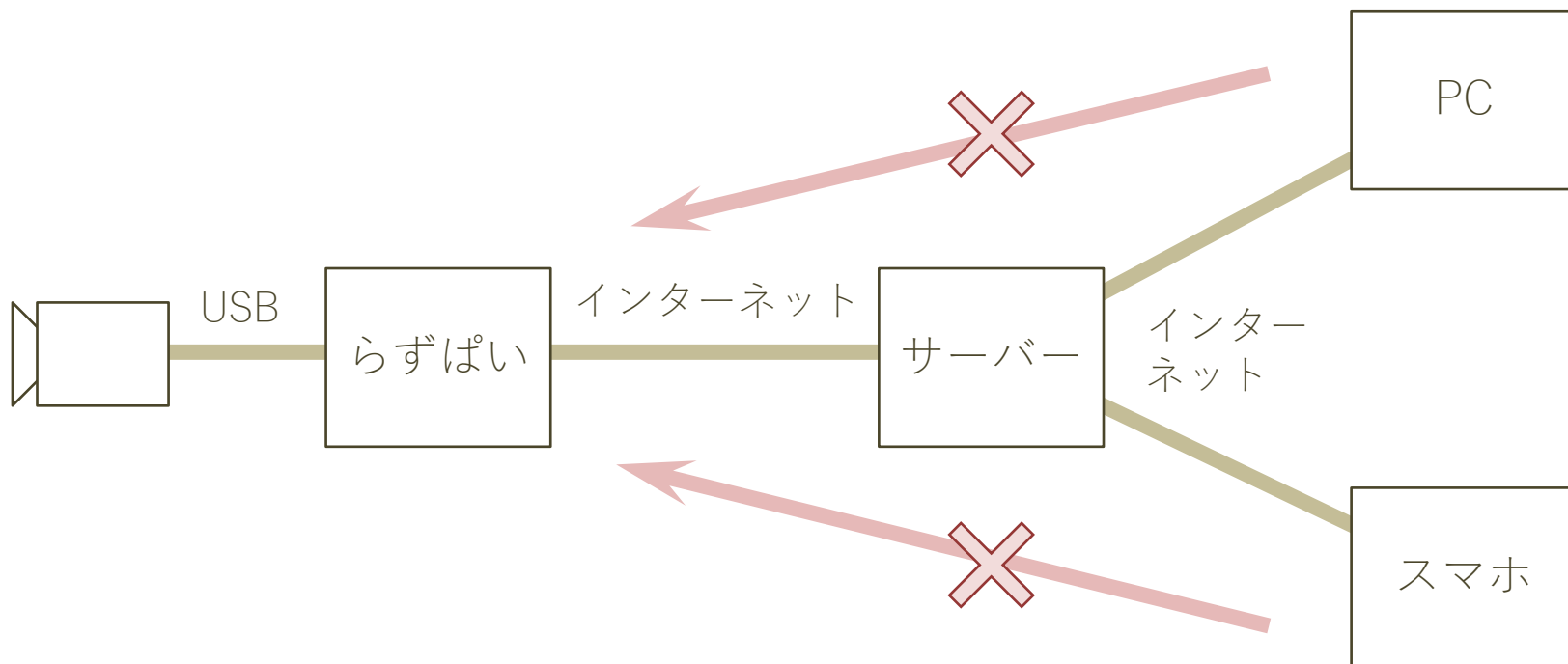
ぼく「Live配信サイトつくらなきゃ！（使命感）」

ぼく「うまくいったらENOGのネタにしたろ」

こうそう



こうそう



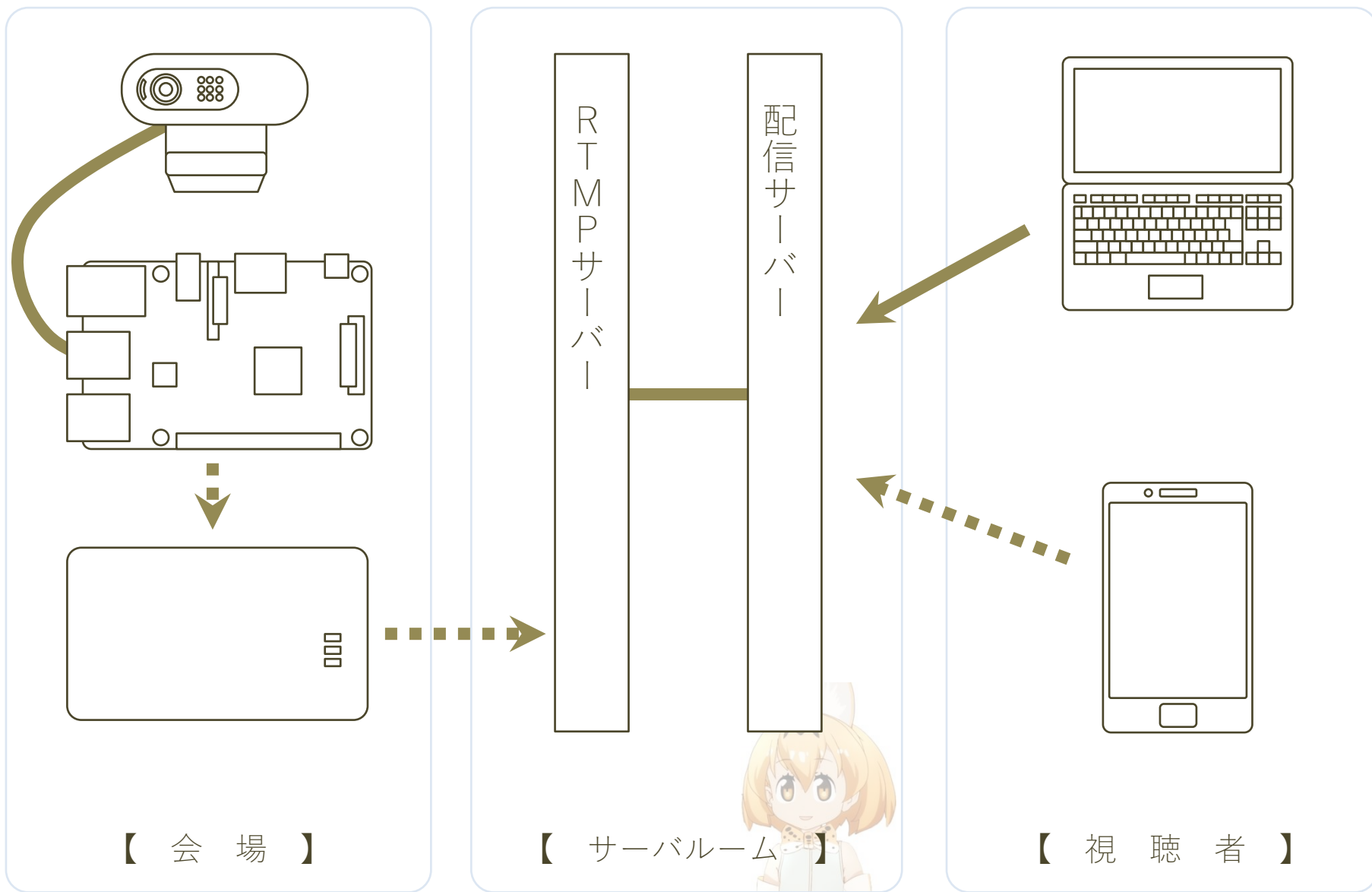
直接はRaspberry Piにアクセスさせたくないな

なんやかんやありまして

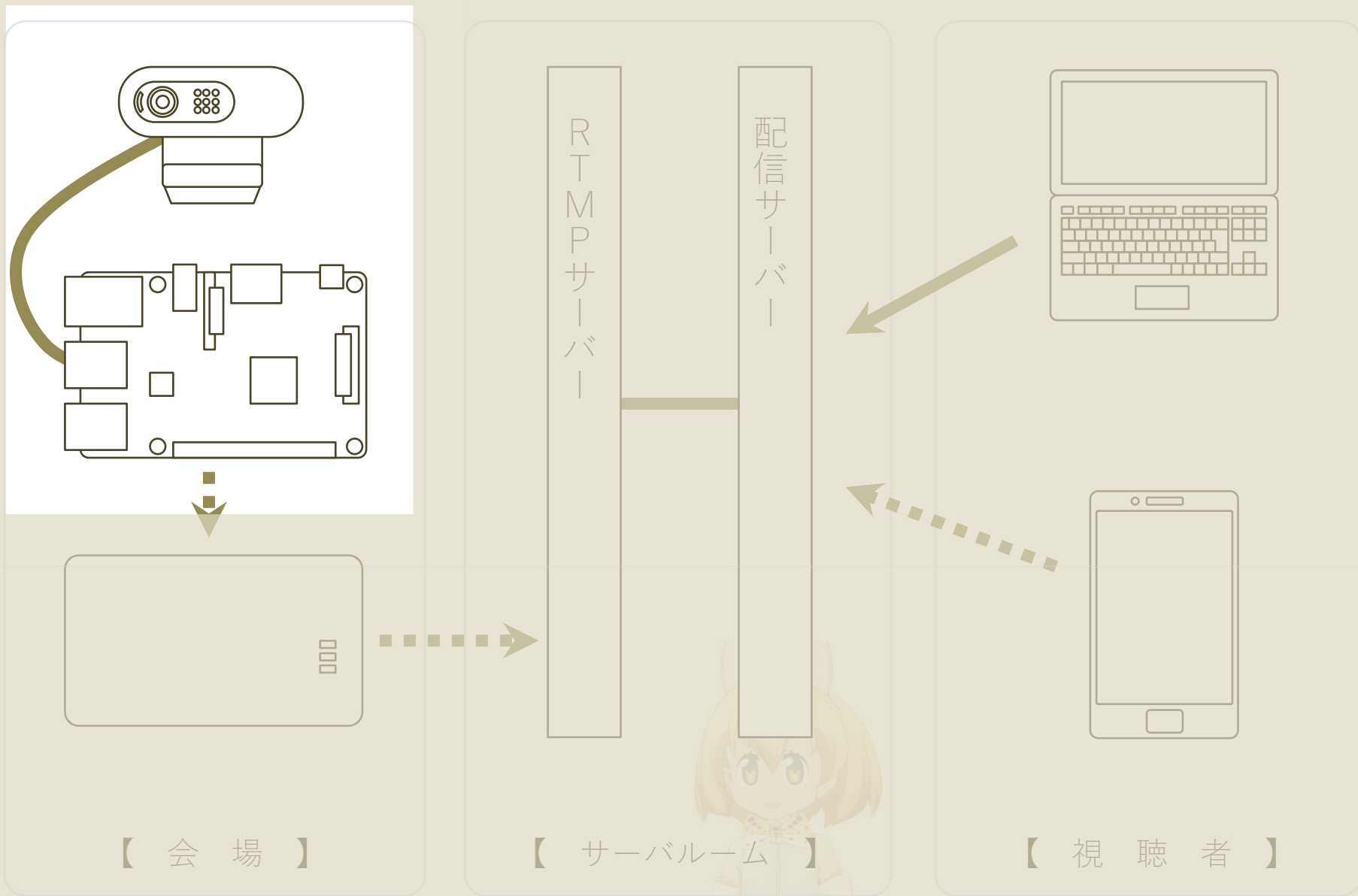




# こうせい



# こうせい



# Raspberry Pi 3 準備

## Raspberry Pi 3のお仕事

- ・ USBカメラの映像と音声をFFmpegを使ってRTMPサーバーに投げる

# Raspberry Pi 3 準備

## USBカメラを接続

```
root@raspberrypi:/home/pi# lsusb  
Bus 001 Device 004: ID 046d:0825 Logitech, Inc. Webcam C270  
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter  
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

# Raspberry Pi 3 準備

## FFmpeg をインストール

- ALSA (Advanced Linux Sound Architecture)に対応させる
- Raspberry Pi 3 に内蔵のH264ハードウェアエンコーダ「h264\_omx」(H264 OpenMAX)に対応させる

# Raspberry Pi 3 準備

## FFmpeg をインストール

```
cd /usr/local/src  
git://source.ffmpeg.org/ffmpeg.git
```

```
wget ftp://ftp.alsa-project.org/pub/lib/alsa-lib-1.1.3.tar.bz2  
tar xvjf alsa-lib-1.1.3.tar.bz2  
cd alsa-lib-1.1.3  
./configure --prefix=/usr/local/src/ffmpeg  
make -j4  
make install
```

```
cd /usr/local/src/ffmpeg  
./configure --enable-gpl --enable-nonfree --enable-mmial --enable-omx-rpi --enable-omx ¥  
--extra-cflags=-I/usr/local/src/ffmpeg/include --extra-ldflags=-L/usr/local/src/ffmpeg/lib --extra-libs=-ldl  
make -j4  
make install
```

# Raspberry Pi 3 準備

```
root@raspberrypi/home/pi# ffmpeg -formats
ffmpeg version git-2017-04-04-1229007 Copyright (c) 2000-2017 the FFmpeg developers
  built with gcc 4.9.2 (Raspbian 4.9.2-10)
  configuration: --enable-gpl --enable-nonfree --enable-mmal --enable-omx-rpi --enable-omx
  --extra-cflags=-I/usr/local/src/ffmpeg/include --extra-ldflags=-L/usr/local/src/ffmpeg/lib --extra-libs=-ldl
```

-- 前略 --

DE alsa ALSA audio output

-- 後略 --

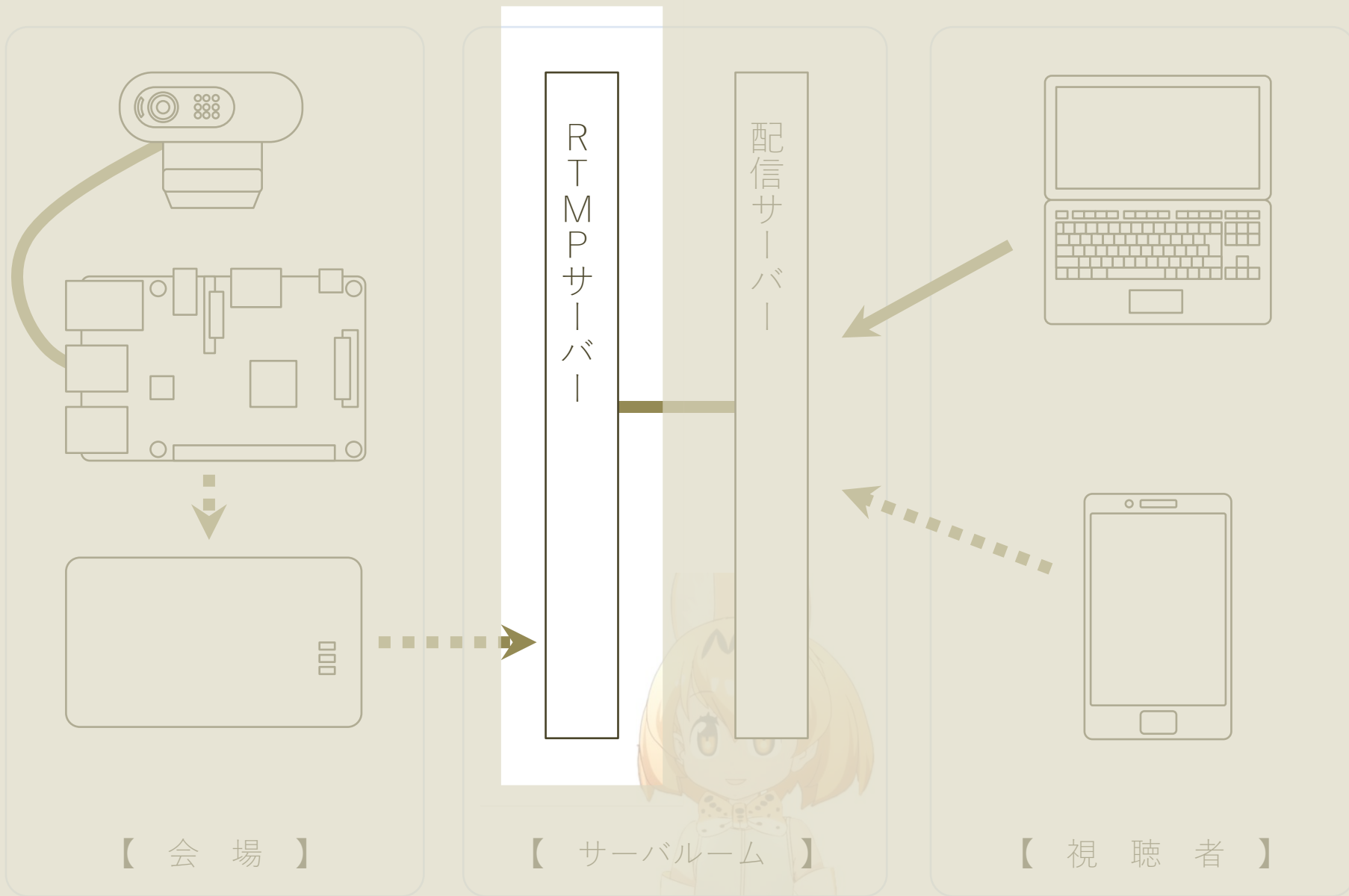
```
root@raspberrypi:/home/pi# ffmpeg -codecs
ffmpeg version git-2017-04-04-1229007 Copyright (c) 2000-2017 the FFmpeg developers
  built with gcc 4.9.2 (Raspbian 4.9.2-10)
  configuration: --enable-gpl --enable-nonfree --enable-mmal --enable-omx-rpi --enable-omx
  --extra-cflags=-I/usr/local/src/ffmpeg/include --extra-ldflags=-L/usr/local/src/ffmpeg/lib --extra-libs=-ldl
```

-- 前略 --

DEV.LS h264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 (decoders: h264 h264\_mmal ) (encoders:  
h264\_omx )

-- 後略 --

# こうせい





# RTMPサーバーの準備

RTMP(Real Time Messaging Protocol)

- ・ リアルタイムにコミュニケーションを行うためのプロトコル
- ・ flashで動画や音声をやり取りするためにAdobeが開発
- ・ TCP上でうごく
- ・ httpを介さないなので低レイテンシ
- ・ RTMP用のサーバーが必要

# RTMPサーバーの準備

## RTMPサーバーのお仕事

- Raspberry Pi 3がストリーミングデータを流せるようにポートを開けてバッチコイする
- 配信サーバがストリーミングデータを取りに来れるようにポートを開けてバッチコイする

# RTMPサーバーの準備

## RTMP対応Nginxのインストール

```
apt-get install gcc make openssl libssl-dev zip libpcre3 libpcre-dev
```

```
wget https://github.com/arut/nginx-rtmp-module/archive/master.zip  
unzip master.zip
```

```
wget https://nginx.org/download/nginx-1.12.0.tar.gz  
tar zxvf nginx-1.12.0.tar.gz  
cd nginx-1.12.0
```

```
./configure --user=www-data --group=www-data --with-http_ssl_module --with-http_realip_module ¥  
--add-module=../nginx-rtmp-module-master  
make  
make install
```

# RTMPサーバーの準備

## RTMPサーバーとしてNginxを起動

```
vi /usr/local/nginx/conf/nginx.conf
```

最下部に追記

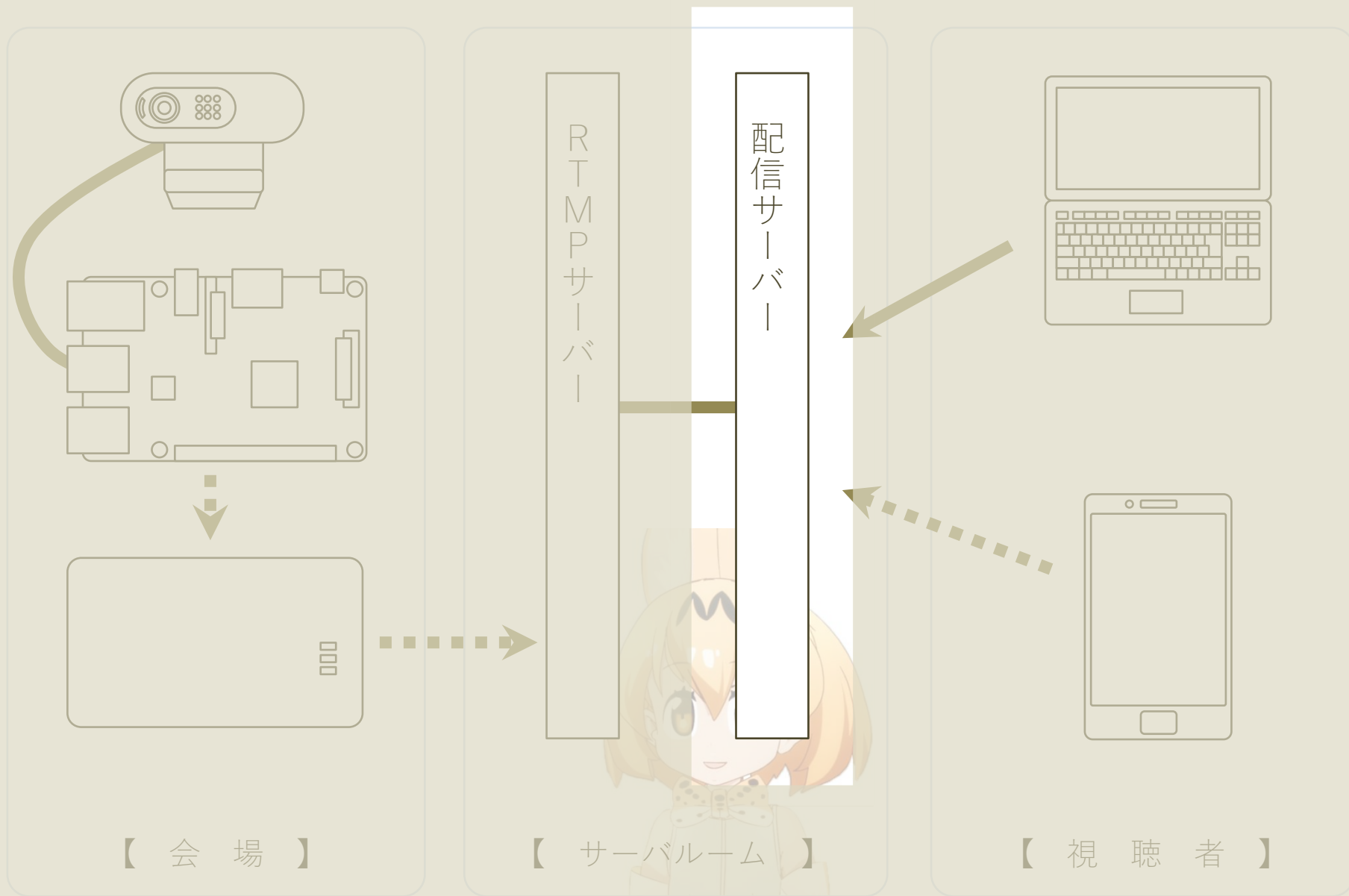
```
rtmp {  
    server {  
        listen 1935;  
  
        application live {  
            live on;  
            record off;  
        }  
    }  
}
```

```
/usr/local/nginx/sbin/nginx
```

```
netstat -antlp
```

| Proto | 受信-Q | 送信-Q | 内部アドレス       | 外部アドレス    | 状態     | PID/Program name |
|-------|------|------|--------------|-----------|--------|------------------|
| tcp   | 0    | 0    | 0.0.0.0:1935 | 0.0.0.0:* | LISTEN | 11254/nginx      |
| tcp   | 0    | 0    | 0.0.0.0:80   | 0.0.0.0:* | LISTEN | 11254/nginx      |

# こうせい



# 配信サーバーの準備

## HLS(HTTP Live Streaming)

- ・動画を数秒ごとの細切れにして(.ts)
- ・ファイル名や再生時間、順番を書いたリストを作って(.m3u8)
- ・再生側はリストを読んで細切れファイルをダウンロードする
  
- ・動画はH264 音声はAAC が一般的

# 配信サーバーの準備

## 配信サーバーのお仕事

- ・ RTMPサーバから流れてくるLive動画を  
HLS(HTTP Live Streaming)でストリーミング配信する
- ・ Windows PC 環境でも見れるようにしてあげる

# 配信サーバーの準備

FFmpeg をインストール

- ・ 略



# 配信サーバーの準備

Mac/iOS/Android : html5 <video>タグ

Win : オープンソースのHLS対応flashページを準備

- javascript : flowplayer-3.2.12.min.js
- flash : flowplayer.swf , HLSProviderFlowPlayer.swf

# 配信してみよう

Raspberry Pi 3 (送信)側

```
/usr/local/bin/ffmpeg ¥  
-re -y ¥  
-f v4l2 -thread_queue_size 16384 -ss 00:00:02 -i /dev/video0 ¥  
-f alsa -thread_queue_size 4096 -ac 1 -ar 44100 -i hw:1,0 ¥  
-c:v h264_omx -b:v 1000k -bufsize 1000k -s 640x480 ¥  
-c:a aac -b:a 96k -ar 44100 ¥  
-map 0:0 -map 1:0 ¥  
-f flv rtmp://219.101.65.200/live/1
```

# 配信してみよう

## Raspberry Pi 3 (送信)側

```
/usr/local/bin/ffmpeg ¥
```

```
-re -y ¥
```

```
-f v4l2 -thread_queue_size 16384 -ss 00:00:02 -i /dev/video0 ¥
```

```
-f alsa -thread_queue_size 4096 -ac 1 -ar 44100 -i hw:1,0 ¥
```

```
-c:v h264_omx -b:v 1000k -bufsize 1000k -s 640x480 ¥
```

```
-c:a aac -b:a 96k -ar 44100 ¥
```

```
-map 0:0 -map 1:0 ¥
```

```
-f flv rtmp://xxx.xxx.xxx.xxx/live/1
```

グローバルなオプション

動画の入力に関するオプション

音声の入力に関するオプション

動画のエンコードに関するオプション

音声のエンコードに関するオプション

出力に関するオプション

出力の指定

# 配信してみよう

配信サーバー (受信)側

```
/usr/local/bin/ffmpeg ¥  
-y ¥  
-i rtmp://xxx.xxx.xxx.xxx/live/1 ¥  
-c copy ¥  
-f segment -segment_format mpegts -segment_time 5 ¥  
-segment_list /var/www/html/live/stream.m3u8 ¥  
/var/www/html/live/stream%04d.ts
```

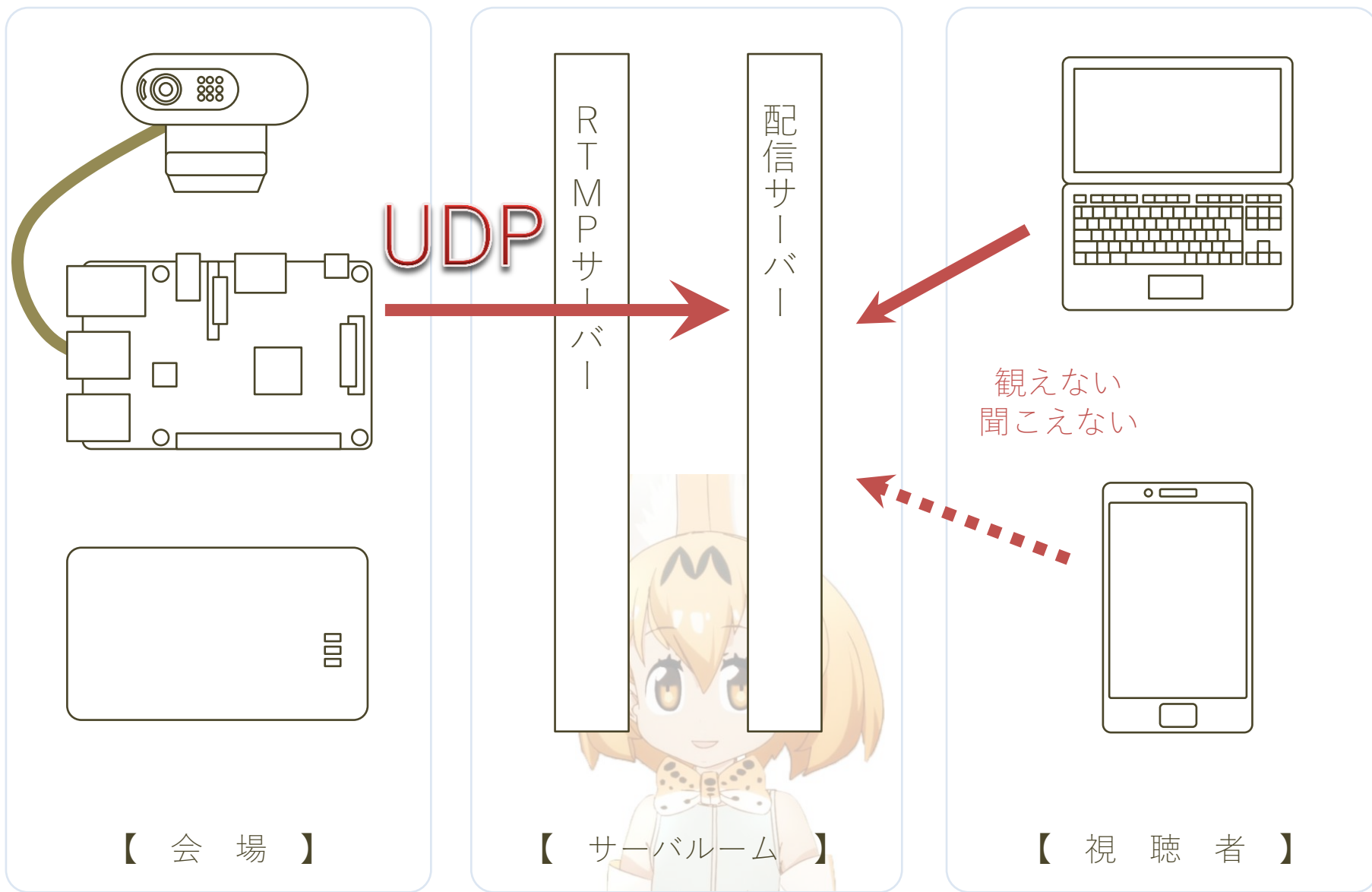
観れています？

こんなこともありました

# ffmpegむずかしい

- ・ まず動かない
- ・ 書く場所によって異なる意味を持つオプション
- ・ なんで1つのコマンドに -f が3つもでてくるの
- ・ なんで映像プチフリするの
- ・ なんでエラーが滝のように流れてるの
- ・ なんで君らは妨害してくるの

# UDPで送ればいいじゃん(いいじゃん)





# ちよつとずつノイジーになる音声

- ・視聴していると少しずつノイズが増えていき、突然ストンと通常の音声に戻る
- ・ていうか今も

# 今後の展望

- ・ 高品質化
  - 映像
  - 音声
  - 遅延
- ・ 移動撮影
  - リアルタイムで見ながら撮影
- ・ 視聴ページのSNS連携

ご清聴ありがとうございました