3 Youtube: Proof of Concept

Die Insta360 Kamera bietet von Haus aus eine Funktionalität, um direkt an YouTube einen Stream zu senden.

Da diese Funktionalität bereits komplett implementiert ist, war unser erster Schritt, diese zu testen und zu prüfen, ob unser Vorhaben grundsätzlich möglich ist und ob sogar YouTube eine mögliche Lösung für die gegebene Aufgabenstellung ist.

3.1 Anleitung zum Starten eines Livestreams mit YouTube

Die Erzeugung eines Livestreams über Youtube ist recht einfach und erfordert nur wenige Schritte.

Voraussetzung, um diese Funktionalität nutzen zu können ist allerdings ein registriertes Konto bei YouTube (beziehungsweise Google, da YouTube zum Konzern Google gehört.)

- 1. iPhone mit Kamera koppeln.
- 2. App Insta360 ONE öffnen.
- 3. Option YouTube wählen, Konto auswählen und bestätigen.
- 4. Ein Link, um Zugang zum Stream zu erhalten, wird in die Zwischenablage kopiert.



Figure 1: Starten eines Livestreams mit YouTube

Weitere Einstellung können im YouTube Konto unter dem Reiter **Events** (siehe Figure 2) getroffen werden. Darunter fällt zum Beispiel die Sichtbarkeit des Streams, also ob der Stream über die Suchfunktion in YouTube gefunden werden kann.

Außerdem werden hier gewisse Informationen wie die Laufzeit und der Streamstutus angezeigt.

Jetzt streamen	_	Streaming b	eenden LIVE 00:00:45		Hilfe
Kamera	NEU	Mit dem Vorscha	J-Player synchronisieren 🕜		
COMMUNITY	~		26. November 2018 um 08:13 (GMT-8)	DURCHSCHNITTLICHE WIEDERGABEDAUER VON	00:00:00
KANAL	~	GUT	Der Status ist einwandfrei.	GESAMTWIEDERGABEZEIT (STUNDEN)	0
ANALYTICS	~	1440P-STREAM		GLEICHZEITIGE VERBINDUNGEN ZU SPITZENZEITEN	0
ÜBERSETZUNGEN & UNTERTITEL	~	Verwalten	Analytics		Feedback gebe
VIDEO-TOOLS	~				
		VORSCHAU		BROADCAST-WARNUN	G
				Keine	•
				Aktivieren	
lilfe & Feedback				Broadcast-Warnungen funk HTML5-Playern und auf ma	tionieren nur in Flash- und nchen Geräten
Hilfe & Feedback				möglicherweise überhaupt i	licht.

Figure 2: Konfiguration des Streams im YouTube Konto

3.2 Auswertung

Der Livestream funktioniert und ist auch sehr einfach zu realisieren.

Ein positiver Aspekt des YouTube Players ist es, dass man den Stream jederzeit pausieren kann. Sobald der Livestream endet, wird er als Video in der Bibliothek gespeichert und kann jederzeit erneut angesehen werden.

Besonders negativ fallen aber zwei Punkte auf:

Die hohe Latenz (etwa 30 Sekunden) und dass fremde Server genutzt werden. Ideal wäre eine Latenzzeit von wenigen Sekunden. Grundsätzlich ist das Nutzen von Servern dritter zwar möglich, gerade im Bereich der SmartFactory ist aber natürlich auch eine gewisse Sicherheit gewünscht und der Einsatz eines eigenen Servers ist sinnvoll.

Außerdem gibt es ein Browserplugin für den Edge Browser von Microsoft [1], das es ermöglicht, den 360° Stream auf der "Mixed Reality Brille" von Acer, die uns zur Verfügung steht, zu anzusehen.

Letztlich kommen wir zu dem Schluss, dass YouTube die nötige Technik bereit stellt, aber für unsere Zwecke noch nicht zufriedenstellend ist.

Das Proof of Concept ist aber erbracht und damit ist klar, dass unser Ziel auch umsetzbar ist.

In Table 2 sind noch einmal die positiven und negativen Aspekte eines Livestream mit

YouTube kurz und übersichtlich dargestellt.

YouTube			
Vorteile	Nachteile		
Kostenlos	Hohe Latenz (etwa 30 Sekunden)		
Einfache Inbetriebnahme	Fremde Server		
Übersichtlich			
Direkte Bearbeitung möglich			
Kostenlose Browser Plugins für			
VR-Brille verfügbar			

Table 1: Auswertung der Livestreaming Kapazitäten durch YouTube

4 Streaming mit nginx - RTMP Server

Als nächsten Schritt, haben wir uns mit dem Aufbau eines eigenen Streaming Servers auseinander gesetzt, so benötigen wir keinen Streamingdienst wie z.Bsp. YouTube oder Twitch.

Die Insta360 Kamera liefert die Möglichkeit direkt an einen RTMP Server zu streamen. Nach einiger Recherche beschloßen wir, dass es am sinnvollsten ist die kostenlose Webserver Software **nginx** zu verwenden, um einen Server aufzubauen.

nginx erfreut sich großer Beliebtheit, da es modular aufgebaut ist und viele Techniken zur Verfügung stellt.

Um diesen Server aufzusetzen benötigen wir einen Rechner mit Linux Betriebssystem.

Der Stream ist nur innerhalb eines Netzwerkes verfügbar und kann nicht von außerhalb geöffnet werden.

In Figure 3 ist die Netzwerkstruktur erkenntlich: Die Kamera streamt direkt an unseren RTMP Server, die Clienten können sich dann mittels der IP-Adresse des RTMP-Servers den Stream anzeigen lassen.



Livestream

Figure 3: RTMP Server im internen Netzwerk

4.1 Aufsetzen des Streaming Servers

Im Folgenden werden die einzelnen Schritte, um den RTMP Server aufzusetzen dargestellt [3]:

- 1. Die nötigen Dateien herunterladen um den Build von nginx durchführen zu können: sudo apt-get install build-essential libpcre3 libpcre3-dev libssl-dev
- 2. Nun wird der NGINX Source Code heruntergeladen: wget http://nginx.org/download/nginx-1.15.1.tar.gz
- 3. Zusätzlich muss das RTMP Modul [2] für nginx heruntergeladen werden: wget https://github.com/sergey-dryabzhinsky/nginx-rtmp-module/archive/dev.zip
- 4. Anschließend wird das Directory erstellt und die Datein werden dorthin entpackt: tar -zxvf nginx-1.14.2.tar.gz unzip dev.zip cd nginx-1.14.2
- 5. Als nächstes wird der Build-Prozess durchgeführt und nginx installiert: ./configure -with-http_ssl_module-add-module=../nginx-rtmp-module-dev -without-http_gzip_module make sudo make install
- Jetzt wird nginx gestartet und die Konfiguration geöffnet: sudo /usr/local/nginx/sbin/nginx sudo nano /usr/local/nginx/conf/nginx.conf
- 7. Nun wird eine Basis Konfiguration erstellt, die es ermöglicht, dass jeder Client, den Livestream weitergeleitet bekommt:

Code:

```
rtmp {
    server {
        listen 1935;
        chunk_size 4096;
            application live{
                live on;
                record of;
        }
     }
    }
}
```

8. Jetzt wird nginx einmal neugestartet, um den Server korrekt konfiguriert zu starten: sudo /usr/local/nginx/sbin/nginx -s stop sudo /usr/local/nginx/sbin/nginx

4.2 Livestream mit RTMP Server und Insta360

Der Server ist jetzt bereit, um einen Livestream innerhalb eines Netzwerkes zu ermöglichen.

Zunächst wird der Server über die Konsole gestartet, wie in Figure 4 zu sehen ist.



Figure 4: Starten von NGINX und dem RTMP Server

Nun muss die IP-Adresse des Servers ermittelt werden, dazu wird der Linux Befehl *ifconfig* genutzt, um die verschiedenen Interfaces anzuzeigen. In Figure 5 finden wir das Network Interface **wls2** mit der IP-Adresse **10.0.1.51**, dabei handelt es sich um unseren RTMP Server.

pojekt@ProjektPC:~\$ ifconfig
eno1: flags=4099 <up,broadcast,multicast> mtu 1500</up,broadcast,multicast>
ether 4c:cc:6a:32:a9:21 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets θ bytes θ (θ , θ , B)
TY errors 0 dropped 0 overrups 0 carrier 0 collisions 0
device intersuot 16 memory avdf100000 df120000
Los floor 72 UD LOOPRACK DUNNING
to: rtags=75cup,Loupback,Running> into 65550
thet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10 <host></host>
loop txqueuelen 1000 (Lokale Schleife)
RX packets 447 bytes 37372 (37.3 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 447 bytes 37372 (37.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
wls2: flags=4163 <up.broadcast.running.multicast> mtu 1500</up.broadcast.running.multicast>
inet 10.0.1.51 netmask 255.255.255.0 broadcast 10.0.1.255
inet6 fe80::e06d:8dcf:2652:907f prefixlen 64 scopeid 0x20 <link/>
ether ac:2h:6e:27:a8:a3 txqueuelen 1000 (Ethernet)
PX parkets 183665 bytes 262696489 (262 6 MB)
PV across @ dropped @ oversups @ frame @
Ty enclose anopped over units of thate o
TX packets 89278 Dytes 8089017 (8.0 MB)
IX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Figure 5: Ermitteln der IP-Adresse

Jetzt überprüfen wir, durch Eingabe der IP-Adresse in die Browserzeile, ob der Server auch korrekt läuft. nginz zeigt dazu einen Willkommenstext an, wie in Figure 6 zu sehen ist.

Welcome to riginal × +			
(←) → C* @ (0 10.0.1.51		··· 🕲 🕁	Ir\ © ≡
	Welcome to nginx!		
	If you see this page, the nginx web server is successfully installed and working. Further configuration is required.		
	For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.		
	Thank you for using nginx.		

Figure 6: Serverstatus: online

Da der Server jetzt online ist, können wir jetzt in der Insta360 App den Livestream an einen RTMP Server starten.Dazu wird der Link angegeben, der sich aus der IP-Adresse und dem Streamnamen zusammensetzt. Figure 7 zeigt das entsprechende Menü in der App.



Figure 7: Stream an RTMP Server in der Insta360 App

Jetzt kann der Livestream über den VLC Player geöffnet werden. Dazu wird der Link einfach im Menü "Medien öffnen"- "Netzwerk" eingegeben und mit dem "Wiedergabe"-Button gestartet, siehe Figure 8

	Medien öffnen 🧧					
Datei	⊛ <u>M</u> edium	¥ <u>N</u> etzwerk	Aufnahmegerät öffnen			
Netzwerkp	rotokoll					
Bitte geb	en Sie eine Net	zwerkadresse eir	ו:			
rtmp://1	0.0.1.51/live			-		
http://@ mms:// rtsp://s http://v	ionen anzeigen	:om/stream.asx rg:8080/test.sdp m/watch?v=gg64x	:			
	5		Wiedergabe 👻	<u>A</u> bbrechen		

Figure 8: Stream im VLC Player öffnen

In Figure 9 ist ein Screenshot des aktiven Livestreams zu sehen.



Figure 9: Beispielbild aus dem Livestream (VLC Player)

4.3 Auswertung des eigenen Streamingservers

Der Stream funktioniert einwandfrei und läuft sehr stabil - wir konnten keine Einbrüche feststellen. Die Latenz liegt bei circa 5 Sekunden und ist damit im Vergleich zu unseren Test mit YouTube deutlich geringer. Der Stream läuft nur Netzwerk intern und ist somit von Dritten nicht auffindbar, ein weiterer Vorteil gegenüber YouTube, da dort der Stream immer über Google Server läuft.

Ein großer Nachteil ist aber, dass der Stream nur im 16:9 Format angezeigt wird (siehe Figure 9 und nicht als 360° Stream. Der VLC Player unterstützt zwar 360° Videos (offline), jedoch keine Livestreams in diesem Format.

Bis zu dem heutigen Tag steht keine Lösung bereit, um den Livestream im 360° Format anzeigen zu können.

NGINX vs. YouTube				
NGINX	YOUTUBE			
Eigener Server	Google Server			
Aufwändige Inbetriebnahme	Inbetriebnahme einfach und			
	selbsterklärend			
Niedrige Latenz zwischen 4-6	Hohe Latenz >25 Sekunden			
Sekunden				
Nur 16:9 Format	360°			
Keine Plugins für VR-Brille	Kostenlose Plugins für VR-Brille			

Table 2: Vergleich NGINX und Youtube

5 Streaming mit dem Raspberry Pi

Mit einem Raspberry Pi 3B+ wollen wir nun eine mobile Lösung realisieren. Ziel ist es dabei, dass wir keinen PC und Bildschirm mehr benötigen.

Damit der Streaming Server direkt beim Starten des R-Pi gestartet wird, wurde ein Script installiert.

5.1 Installation des Scripts

Es wurden folgende Schritte durchgeführt, um das automatische Starten Servers zu implementieren:

- Service-File *rc-local* erstellen : sudo nano /etc/systemd/system/rc-local.service
- Code einfügen:

Code:

```
[Unit]
Description=/etc/rc.local Compatibility
ConditionPathExists=/etc/rc.local
```

```
[Service]
Type=forking
ExecStart=/etc/rc.local start
TimeoutSec=0
StandardOutput=tty
RemainAfterExit=yes
SysVStartPriority=99
```

```
[Install]
WantedBy=multi-user.target
```

- Prüfen, ob *rc-local* funktioniert und System Boot erlauben: sudo chmod +x /etc/rc.local sudo systemctl enable rc-local
- Den Service starten und den Status überprüfen: sudo systemctl start rc-local.service sudo systemctl status rc-local.service

Zur Veranschaulichung noch die folgenden beiden Figures.

```
In Figure 10 wird der Service Status überprüft und es ist ersichtlich, dass der Service aktiv ist. Das Skript funktioniert also einwandfrei.
```

pi@raspberrypi: ~	-		×
Datei Bearbeiten Reiter Hilfe			
 rc-local.service - /etc/rc.local Compatibility Loaded: loaded (/etc/systemd/system/rc-local.service; enabled; vendor preset: enabled) Drop-In: /lib/systemd/system/rc-local.service.d			^
Active: active (exited) since Thu 2018-12-20 11:53:19 CET; 12min ago Process: 303 ExecStart=/etc/rc.local start (code=exited, status=0/SUCCESS) Main PID: 374 (code=exited, status=0/SUCCESS)			
Dez 20 11:53:18 raspberrypi systemd[1]: Starting /etc/rc.local Compatibility Dez 20 11:53:19 raspberrypi sudo[308]: root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/local/ng Dez 20 11:53:19 raspberrypi sudo[308]: pam_unix(sudo:session): session opened for user root by (uid=0) Dez 20 11:53:19 raspberrypi sudo[308]: pam_unix(sudo:session): session closed for user root Dez 20 11:53:19 raspberrypi systemd[1]: Started /etc/rc.local Compatibility.	jinx	/sl	bi

Figure 10: Status des Service wird überprüft

In Figure 11 ist zur besseren Übersicht noch einmal die entsprechende Ordner-Struktur und der Code von rc-local ersichtlich. Entscheidend ist hier der Aufruf vor dem $exit \ 0$: sudo /usr/local/nginx/sbin/nginx



Figure 11: Script *rc-local* zum automatischen Starten von nginx