



Simone Bonfante

Software Architecture, Alma Mater Studiorum - Università di Bologna

a.a. 2018 / 2019

# Introduzione

- Piattaforma leader di video streaming
- Gaming e IRL
- Focus in performance e scalabilità
- Software proprietario

**140**

MILIONI

gamers ogni  
mese

**15**

MILIONI

visitatori unici  
ogni mese

**20**

MILA

gamers  
guadagnano

**900**

MILIONI

di ore a guardare  
Twitch nel 2018

# Storia di Twitch

- 2007 - Lancio di Justin.tv
- 2011 - Nascita di Twitch.tv
- 2013 - 20 milioni di visitatori unici al mese
- 2014 - Acquisizione da parte di Amazon per \$970 milioni
- 2019 - 500~ milioni di visitatori unici al mese

# Features di Twitch

- Streaming live video
- Video On Demand
- Chat real-time
- Abbonamenti e donazioni
- Bot
- APIs & Estensioni
- Data analysis

I Bot permettono di interagire a livello di codice con la chat utilizzando IRC:

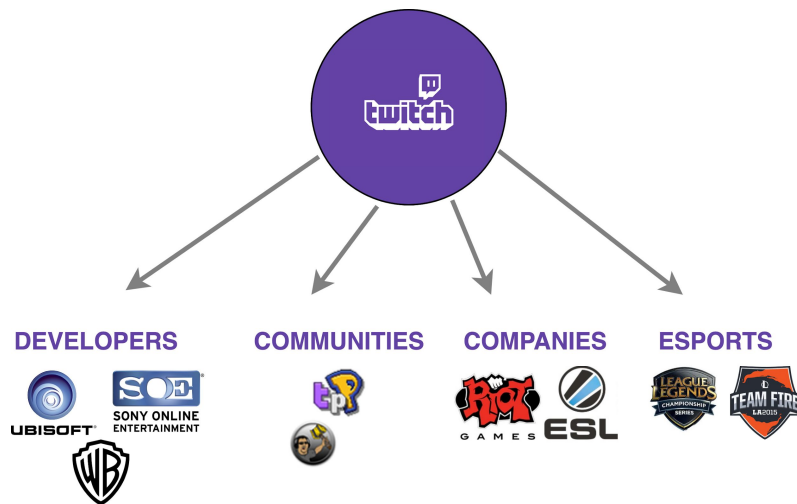
- ❖ Moderazioni
- ❖ Donazioni
- ❖ Analisi

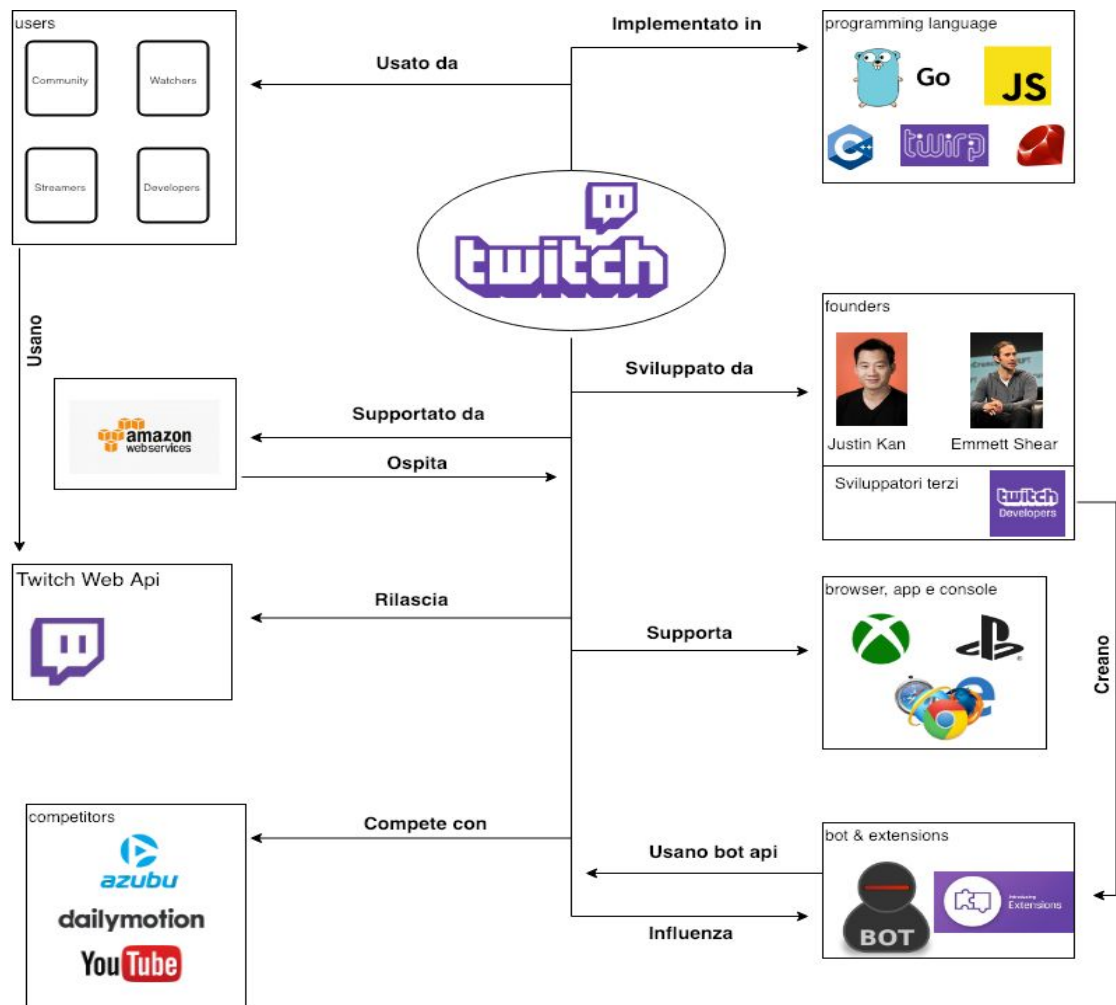
# Casi d'uso

- Trasmettere/Guardare video live
- Chatting nei canali
- Implementare/Utilizzare i Bot

Streamer	Watcher
Intrattenimento	Svago
Guadagno	Apprendimento

Twitch interagisce con:





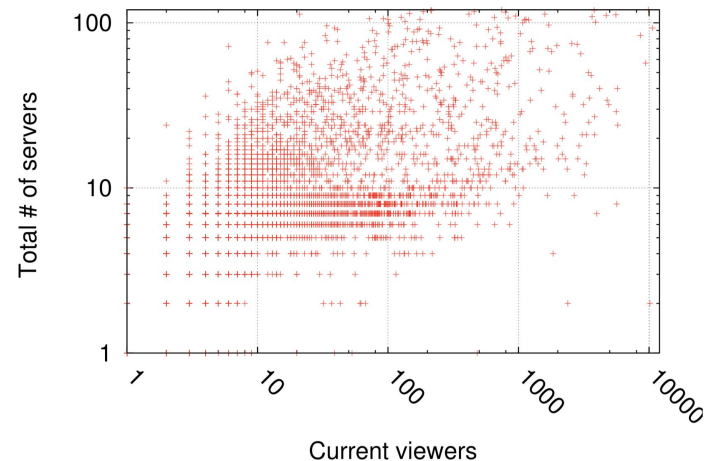
# Requisiti non funzionali: Scalabilità

806 proxy su 50 stati diversi (Europa (154), Asia (372), Africa (24), Australia (4), Nord America (138) e Sud America (114))

Fraction(%)	NA cluster	EU cluster	AS cluster
North America	99.4	0.6	0
South America	96	4	0.01
Europe	17	82	1
Africa	21.8	78.2	0
Asia	34.4	20	45.6

Scalabilità basata su determinate metriche:

- Numero di spettatori sul canale **per regione**
- Topologia e peering
- Distanza geografica



# Requisiti non funzionali: Performance 1/2

Slow broadcasting bandwidth comparata ad altri sistemi.

## Parametri

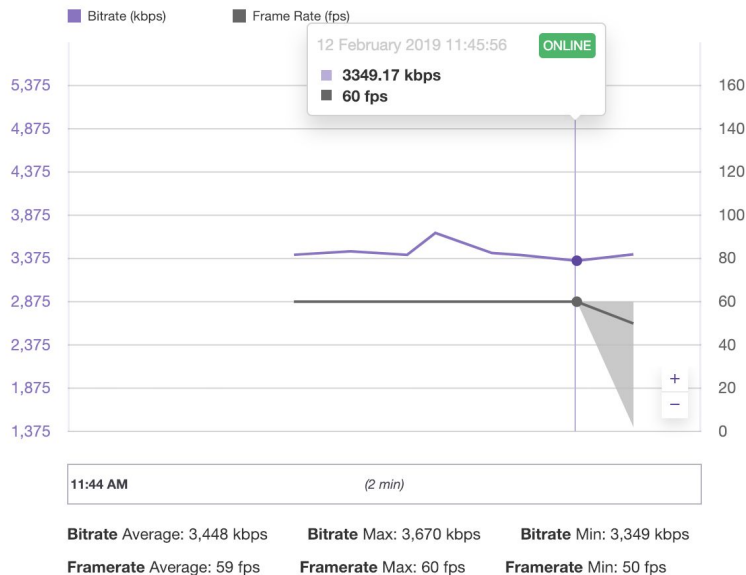
- Bandwidth: 3500 - 6000 kbps
- Risoluzione: 1920 x 1080
- Frame: 60 fps

## Risultati

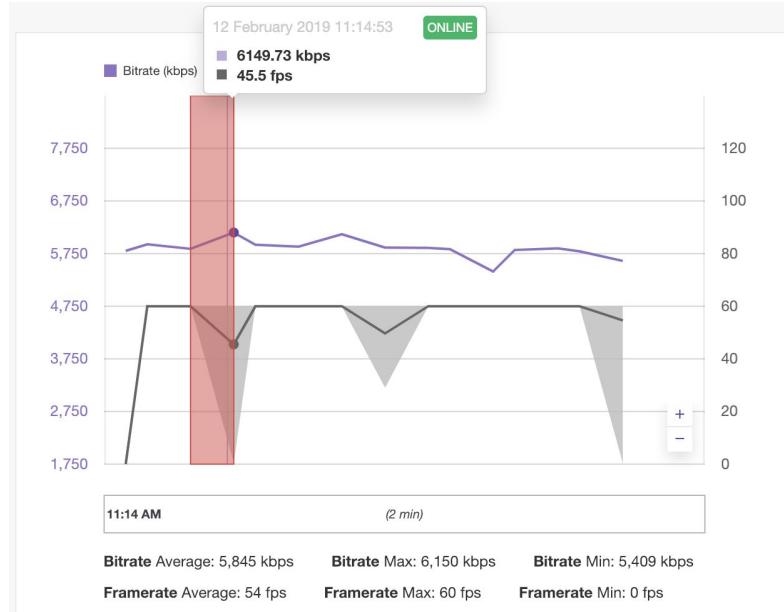
- Buffering: Limitato
- Latenza video: 6 - 8 s
- Qualità: medio - bassa



# Requisiti non funzionali: Performance 2/2



Bitrate: 3500 kbps, 60fps, 1280x720 px.  
Stabile



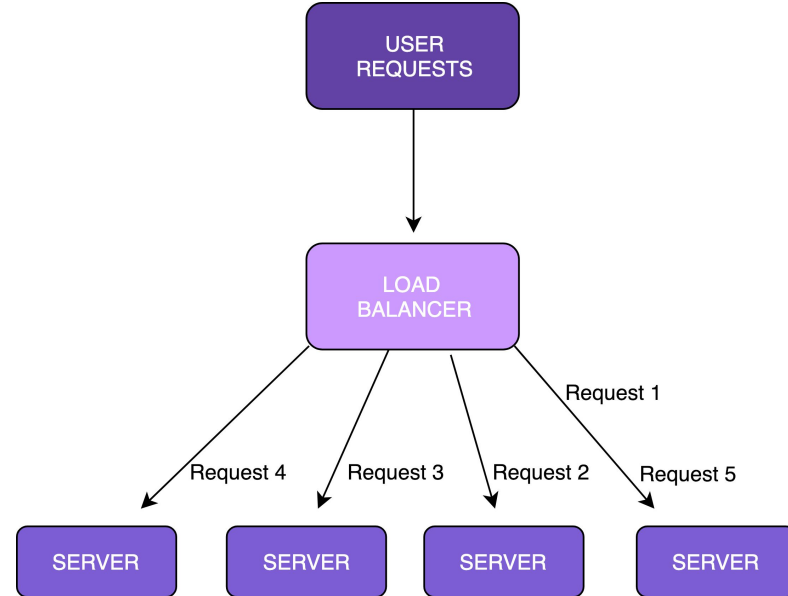
Bitrate: 6000 kbps, 60fps, 1920x1080 px.  
Non del tutto stabile

# Requisiti non funzionali: Disponibilità

**Problema:** interruzione dello streaming dovuto a un guasto di un server

**Soluzione:**

- Ridondanza dei servizi in server diversi
- Copie continue dello status del broadcast
- Tecnica di bilanciamento del carico e di fault-tolerance → **Round-Robin DNS.**

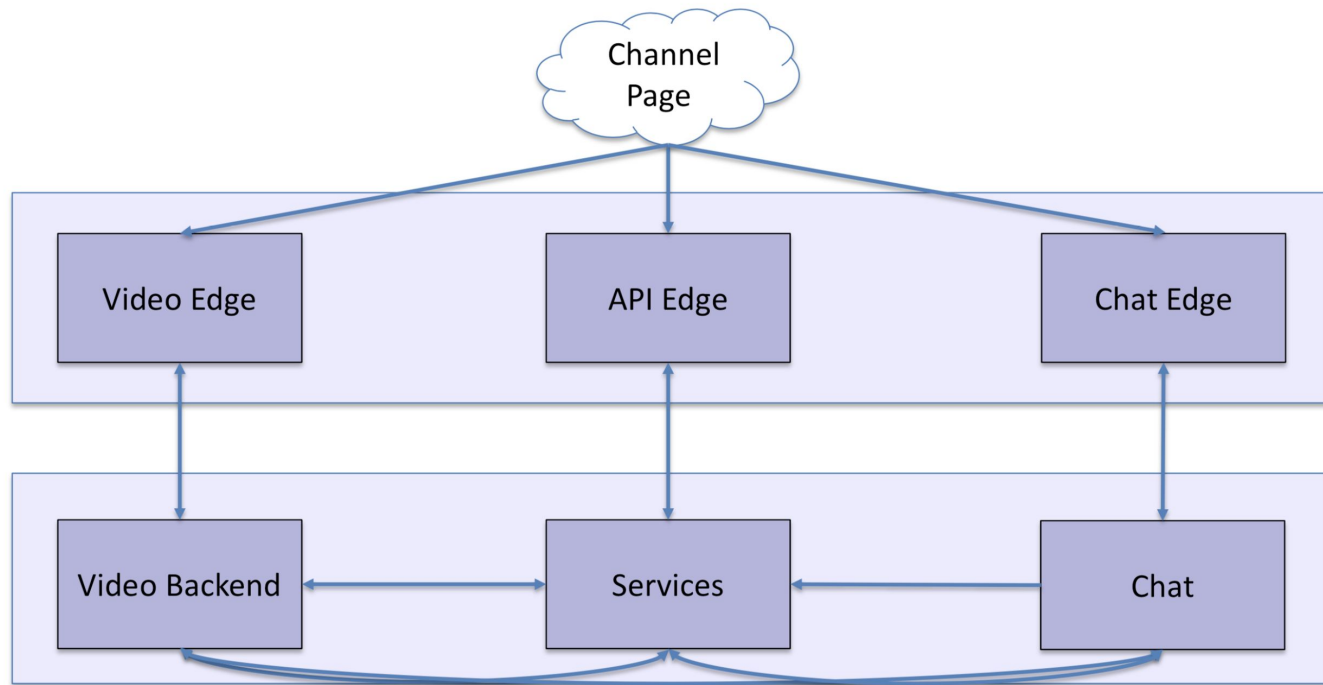


# Requisiti non funzionali: Sicurezza

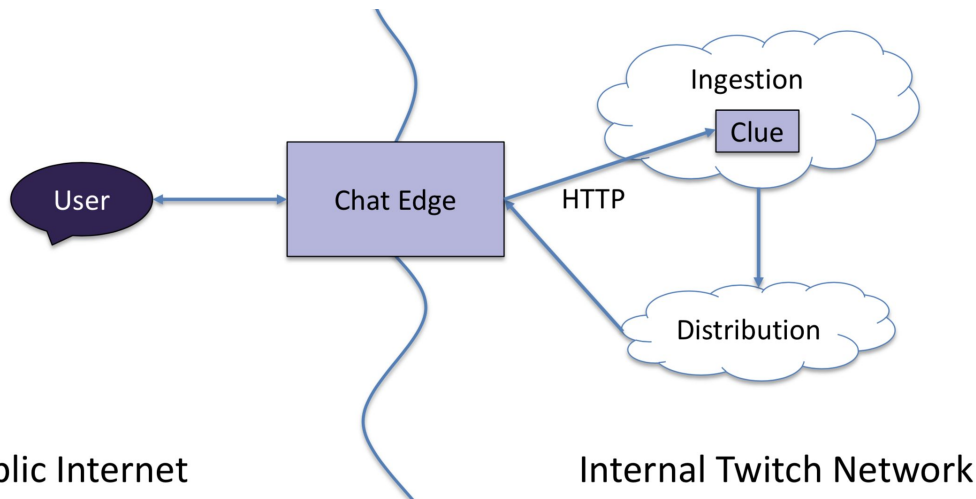
- Autenticazione OAuth 2.0
- Chiave segreta per lo streaming
- Ecosistema Amazon Web Service
- Consigliato l'utilizzo di una VPN



# Architecture Overview



# Struttura: chat server organization



Public Internet

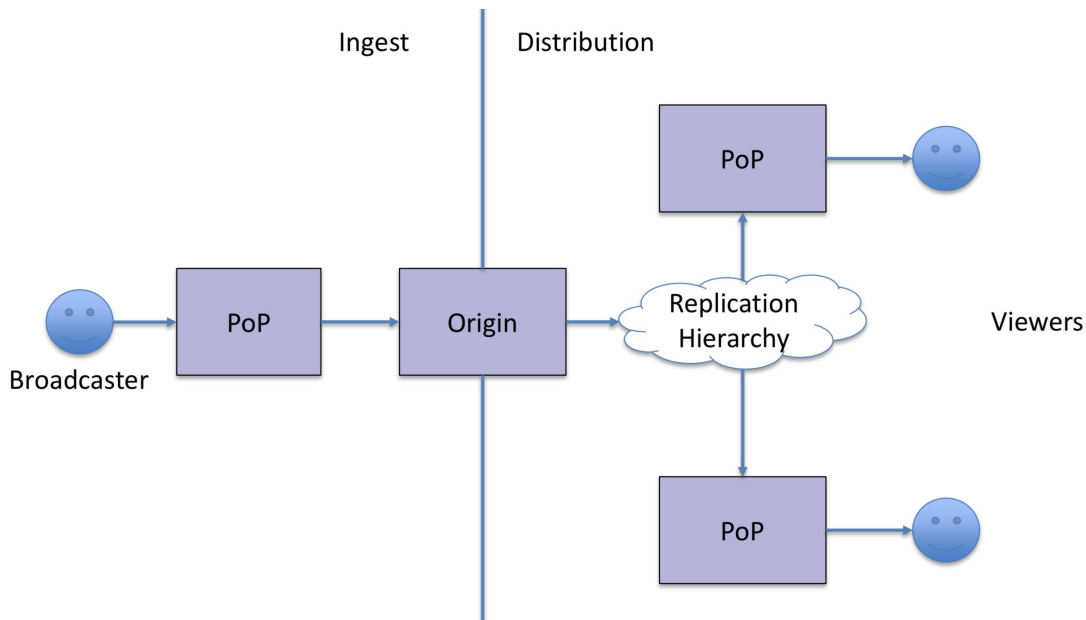
Internal Twitch Network

- **Ingestion:** riceve tutti i messaggi e un suo componente, Clue, ha il compito di eseguire analisi sulle azioni degli utenti
- **Distribution:** prende decisioni su come e quando reindirizzare i messaggi provenienti da Clue

3 tipi di cluster:

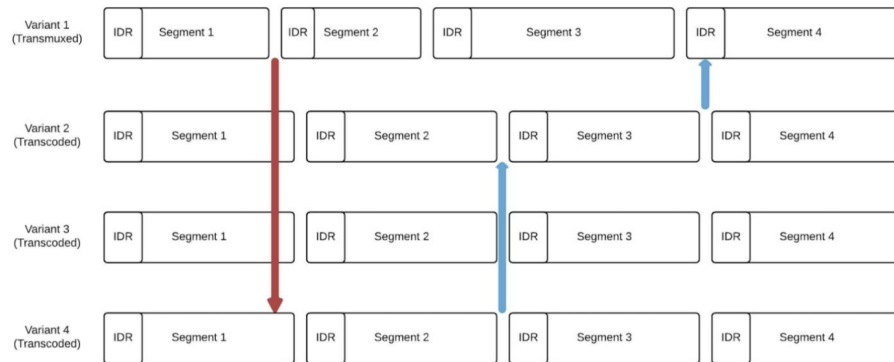
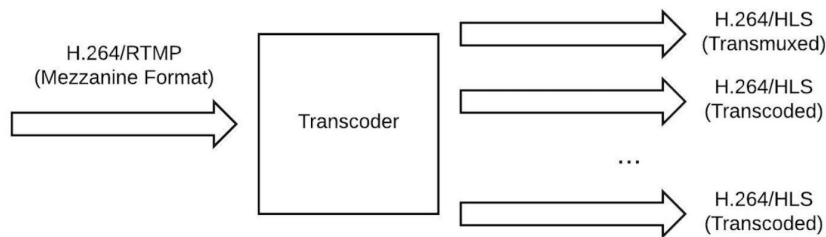
- ❖ Main: 10k-12k di viewers
- ❖ Event: eSport, LoG ecc.. 100k di viewers.
- ❖ Test: testing del nuovo codice. 0,2% del traffico

# Struttura: video service



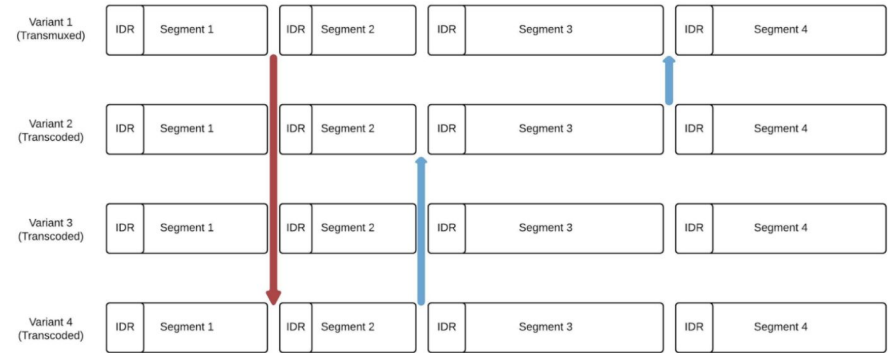
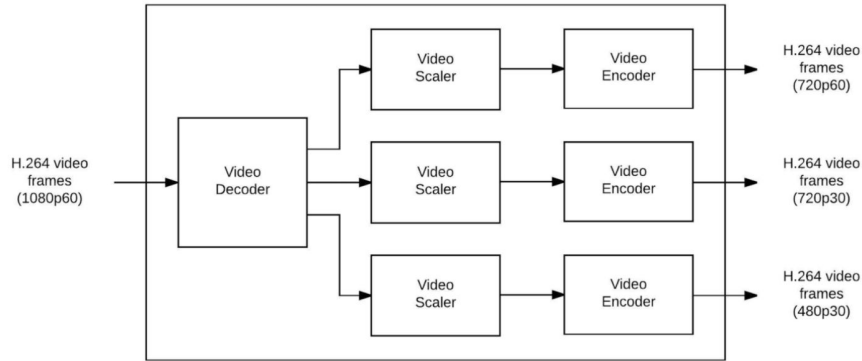
- **Video Ingest:** flussi RTMP raccolti e inviati al sistema di transcodifica
- **Transcodifica:** RTMP in arrivo transcodificato in più flussi HLS.
- **Distribuzione:** distribuisce i flussi HLS ai POP geograficamente disparati
- **VOD:** tutti i flussi video in arrivo vengono archiviati.

# Struttura: video service - Transcoder FFmpeg



- Transcoder è incaricato di convertire un flusso RTMP in entrata nel formato HLS con più varianti (ad es. 1080p, 720p, ecc.)
- FFmpeg eseguito su N istanze indipendenti
- FFmpeg non garantisce allineamento dei segmenti

# Struttura: video service - TwitchTranscoder 1/2

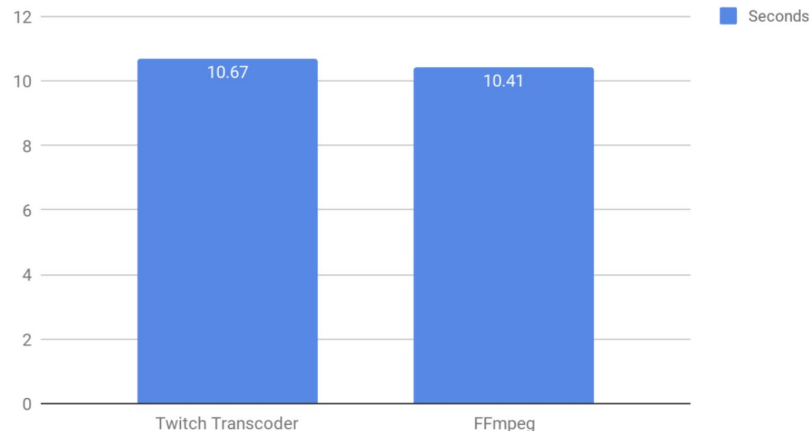


- N transcodificatori 1-in-1-out indipendenti ciascuno dei quali genera 1 flusso di output
- Condivisione di *Decoder*, *Scaler* e *Encoder*
- Architettura multi-core

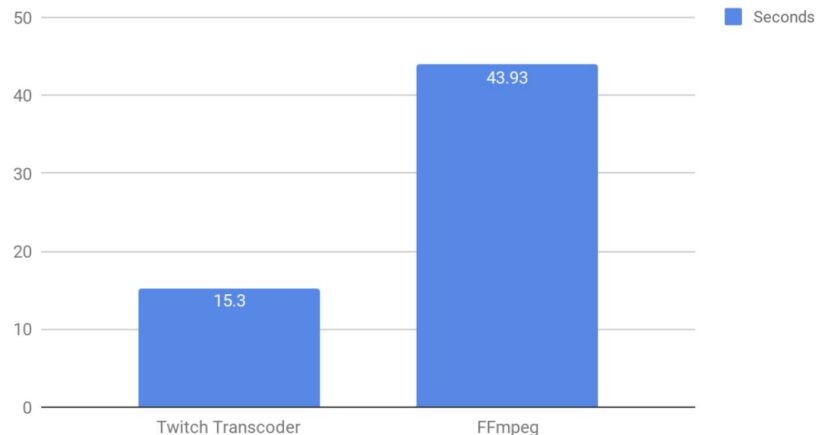


# Struttura: video service - TwitchTranscoder 2/2

Average Transcode Times (720p60)



Average Transcode Times (720p60, 720p30, 480p30, 360p30, 160p30)



- ❖ FFmpeg è leggermente più veloce per l'output a **singola variante**
- ❖ Con più varianti il modello **multithreading** di TwitchTranscoder ha un vantaggio maggiore
- ❖ **65% di tempo** di esecuzione risparmiato

# Struttura: Twirp

- Framework RPC Open Source
- Da codebase monolitica a microservizi
- Documento protobuf descrive l'API e genera un'interfaccia per descrivere il servizio
- Supporto HTTP 1.1
- Supporto di codifica JSON il che rende facile l'implementazione di client cross-language

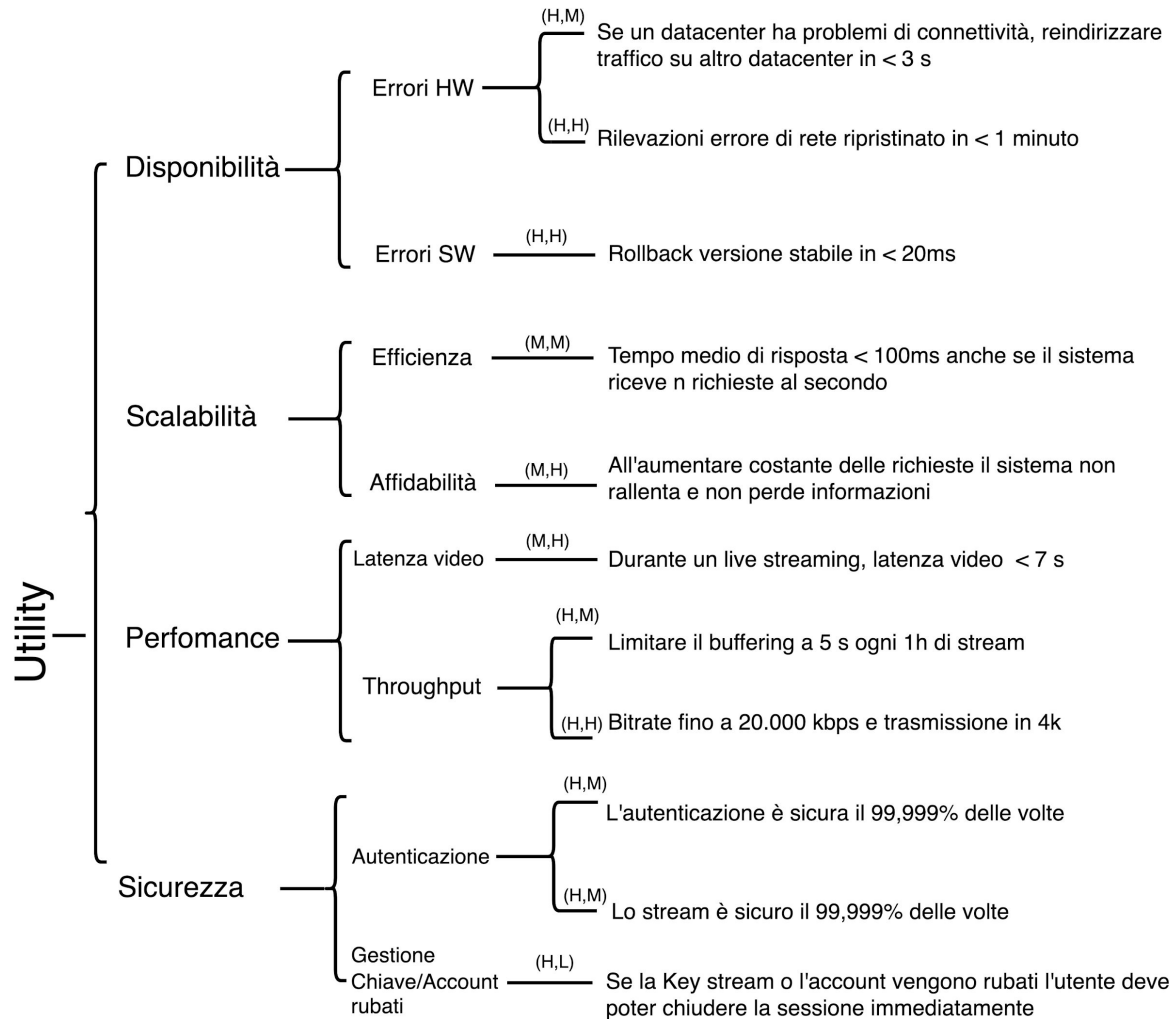


# Utility Tree

Vengono considerati solo i requisiti di maggior interesse analizzati.

Per ognuno di essi viene implementato uno scenario e assegnato un valore che indica **l'importanza** e un altro che indica il **costo/rischio**

- **Disponibilità**
- **Scalabilità**
- **Performance**
- **Sicurezza**



# Confronto: YouTube Live - Features

Features	Twitch	Youtube Live
Client	Mobile, Tablet, Web, Console, Chromecast	Mobile, Tablet, Web, Console, Chromecast
Revenue	✓	✗
Usability	✗	✓
Chat Service	✓ No Limits	✗ 3 msg in 30 s
Chat Bot	Moderatori, Donatori, Analisti	Moderatori
Content	Game	Game, IRL, etc..
DVR (pause, play, rewind)	✗	✓
Apis	✓	✗

# Confronto: YouTube Live



- Qualità:
  - 1920x1080 px
  - 60 fps
  - 6000 kbps
- Latenza: 7-8 s
- Low Latency: 2/3 s
- Buffering: Si



- Qualità:
  - 3840x2160 px
  - 60 fps
  - 51000 kbps
- Latenza: 3-4 s
- Normal, Low, Ultra Low Latency: 1 s
- Buffering: No

# Bibliografia I

- B. C. B. Churchill and W. Xu. The modern nation: A first study on twitch.tv social structure and player/game relationships, Oct 2016.
- M. Claypool, D. Farrington, and N. Muesch. Measurement-based analysis of the video characteristics of twitch.tv. In 2016 IEEE Games Entertainment Media Conference (GEM) , pages 1–4, Oct 2015.
- J. Deng, F. Cuadrado, G. Tyson, and S. Uhlig. Behind the game: Exploring the twitch streaming platform. In 2016 International Workshop on Network and Systems Support for Games (NetGames) , pages 1–6, Dec 2015.
- Jie Deng, Gareth Tyson, Félix Cuadrado, and Steve Uhlig. Internet scale user-generated live video streaming: The twitch case. In PAM, 2017.
- H. Pang, Z. Wang, C. Yan, Q. Ding, K. Yi, J. Liu, and L. Sun. Content harvest network: Optimizing first mile for crowdsourced live streaming, 2018
- Karine Pires and Gwendal Simon. Youtube live and twitch: a tour of user-generated live streaming systems. In MMSys, 2016.

# Bibliografia II

- Roman Poyane. Toxic communication during streams on twitch.tv. the case of dota 2. 2018.
- X. Wang, Y. Tian, R. Lan, W. Yang, and X. Zhang. Beyond the watching: Understanding viewer interactions in crowdsourced live video broadcasting services. IEEE Transactions on Circuits and Systems for Video Technology, 2018
- Cong Zhang and Jiangchuan Liu. On crowdsourced interactive live streaming: a twitch. tv-based measurement study. ACM, 2015
- D. Y. Zhang, J. Badilla, H. Tong, and D. Wang. An end-to-end scalable copyright detection system for online video sharing platforms, Aug 2018