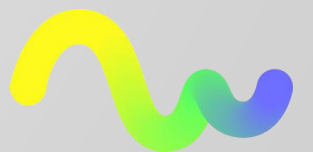


TV Anytime in nuuday

Peter Wille –Architect
petw@nuuday.dk

TV Anytime in nuuday

- Current setup
- Challenges
- Status on development





TV Anytime Current setup

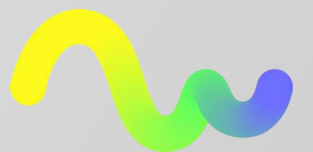
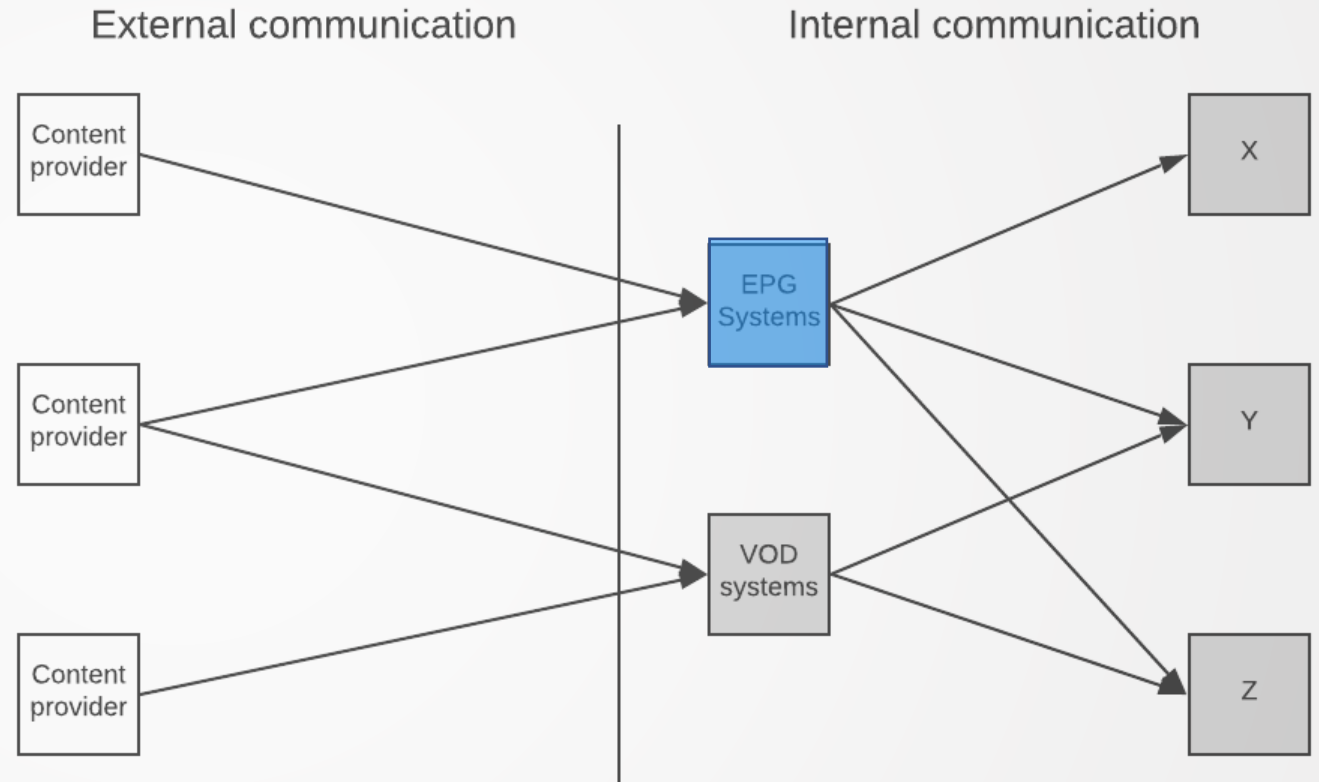
Nuuday data flow from content providers

EPG systems Overview

- Ingests Flow Channel related metadata from content providers

EPG systems Details

- Separate data model from VOD
- Enriching data
- Manuel workflows
- Automatic detection of start and end times



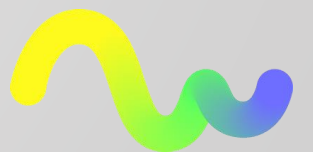
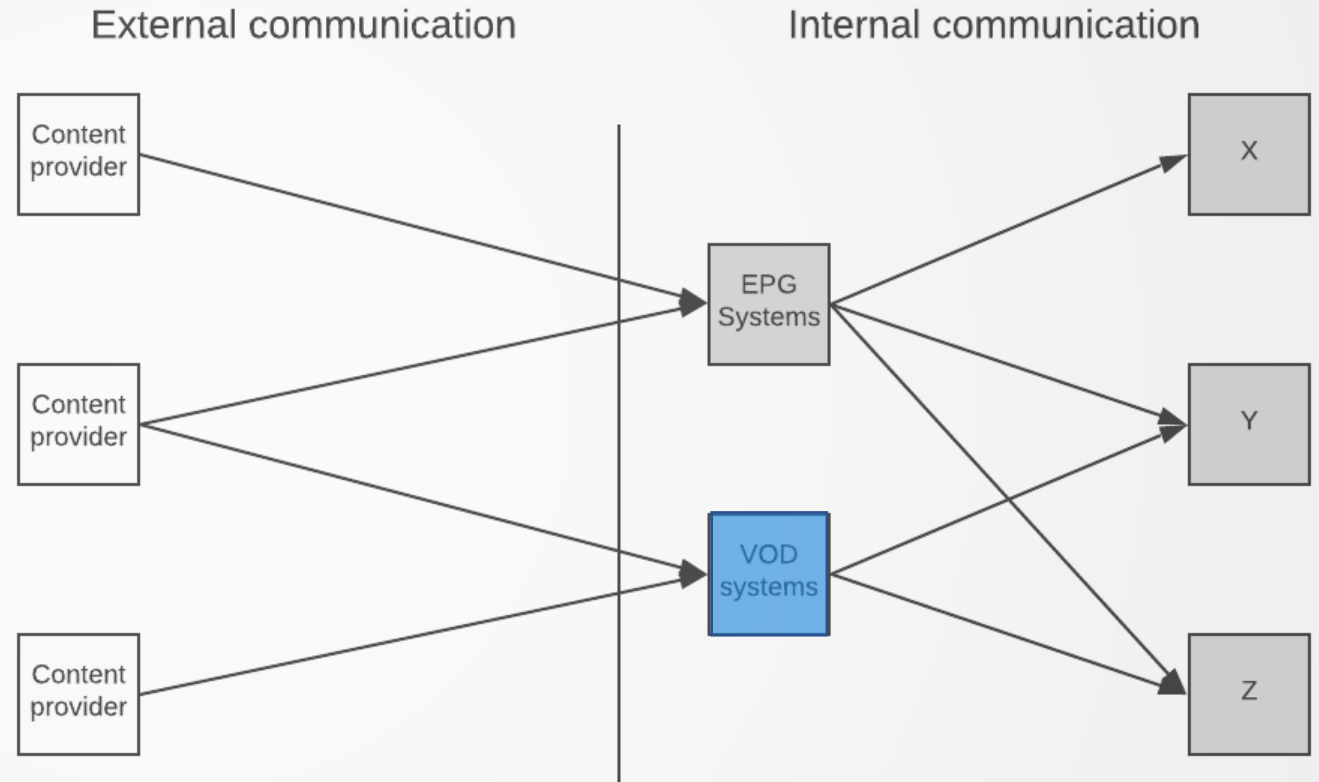
Nuuday data flow from content providers

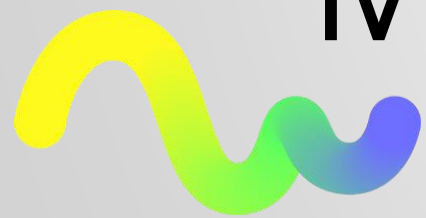
VOD systems Overview

- Ingests on demand related metadata and asset from content providers

VOD systems Details

- Separate data model from EPG
- Transcoding

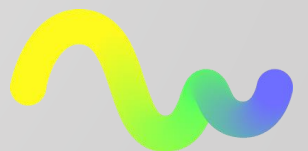




TV Anytime Challenges

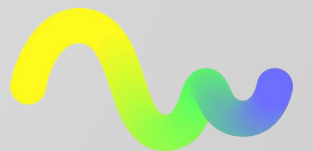
Challenges of being an aggregator

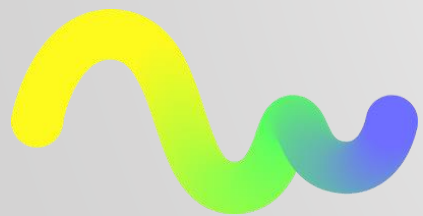
- 200+ flow channels across 36 content providers and 16 SVOD and TVOD content providers
 - Lots of different ways of getting data in very different levels of quality
 - A few content providers that both have on demand and flow channels
 - Identifying the same content even from the same content providers across flow and on demand is a challenge
 - Getting Real time and accurate schedules from providers for Catchup OTT and now and next



Internal challenges

- Current data model is schedule oriented not content oriented
- Hard coupled legacy systems
- Multiple layers of polling-based update functions
- Too few development resources



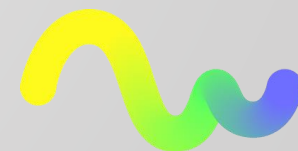
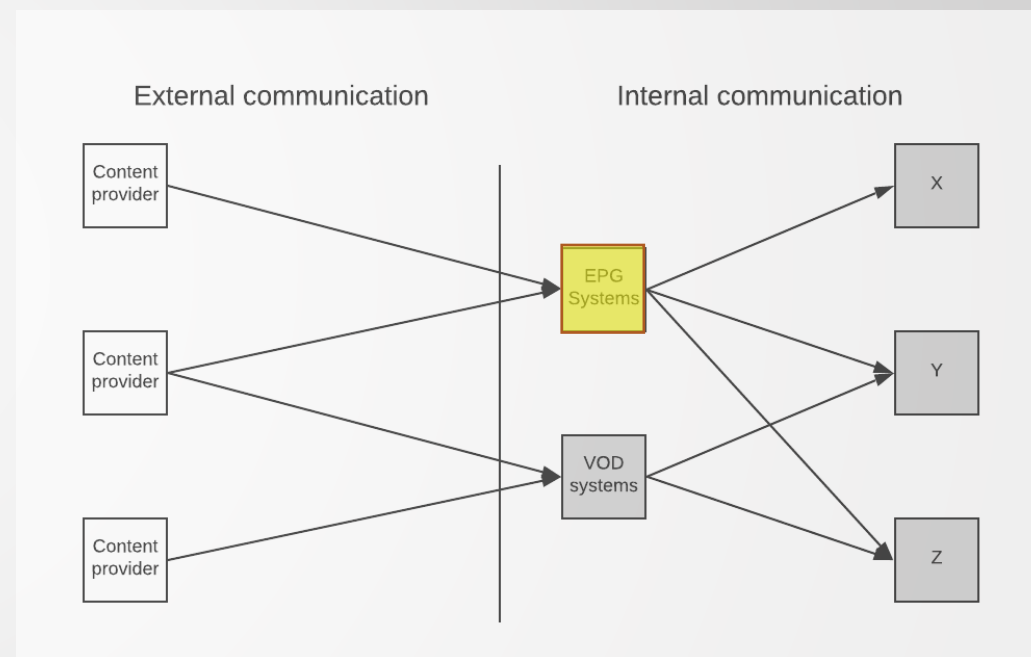


TV Anytime Status on development

TV Anytime-alike for internal usage

● Status on development

- Building on the existing plan of using the best part of TV Anytime
 - Reusing the basis entity structure of TV Anytime
 - Schedule event, on demand and so on
- Extending the model to solve internal challenges
 - Event based communication instead of polling based
 - Why: The new architecture and using event will reduce end to end time from 1-2 hours to seconds.
 - Json instead of pure XML

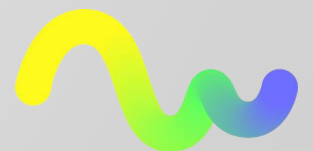
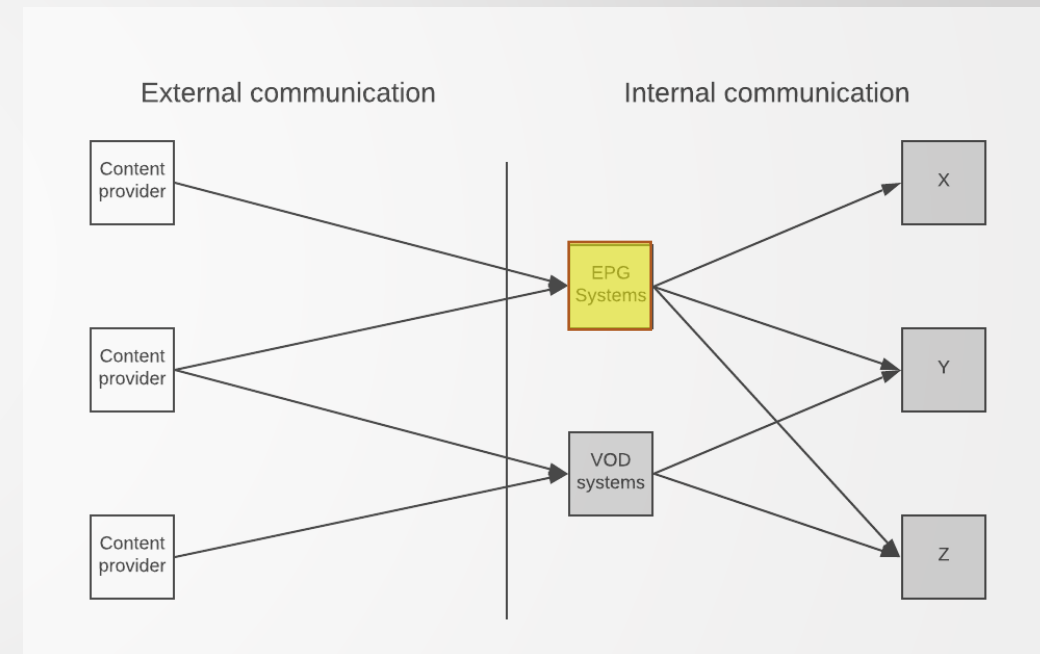


TV Anytime-alike for internal usage

● Status on development

- Building on the existing plan of using the best part of TV Anytime
 - Reusing the basis entity structure of TV Anytime
 - Schedule event, on demand and so on
- Extending the model to solve internal challenges
 - Event based communication instead of polling based
 - Why: The new architecture and using event will reduce end to end time from 1-2 hours to seconds.
 - Json instead of pure XML

Estimated completed in November 2020

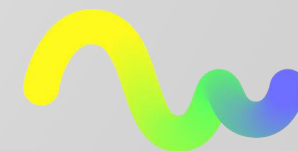
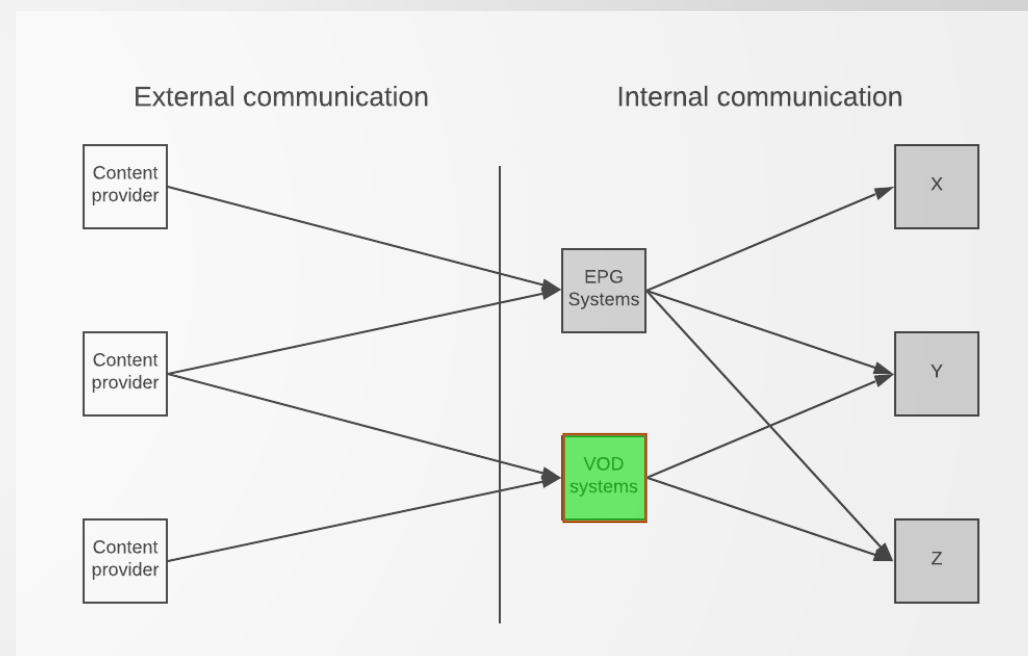


TV Anytime-alike for internal usage

● Status on development

- Building on the existing plan of using the best part of TV Anytime
 - Reusing the basis entity structure of TV Anytime
 - Schedule event, on demand and so on
- Extending the model to solve internal challenges
 - Event based communication instead of polling based
 - Why: The new architecture and using event will reduce end to end time from 1-2 hours to seconds.
 - Json instead of pure XML

Was completed in September 2020





Any questions?

Messages

(In nuudays context)



Having a pile of messages
to go through, to read one

=

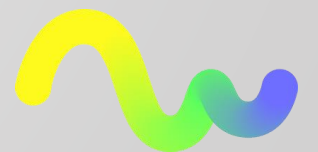
Time consuming



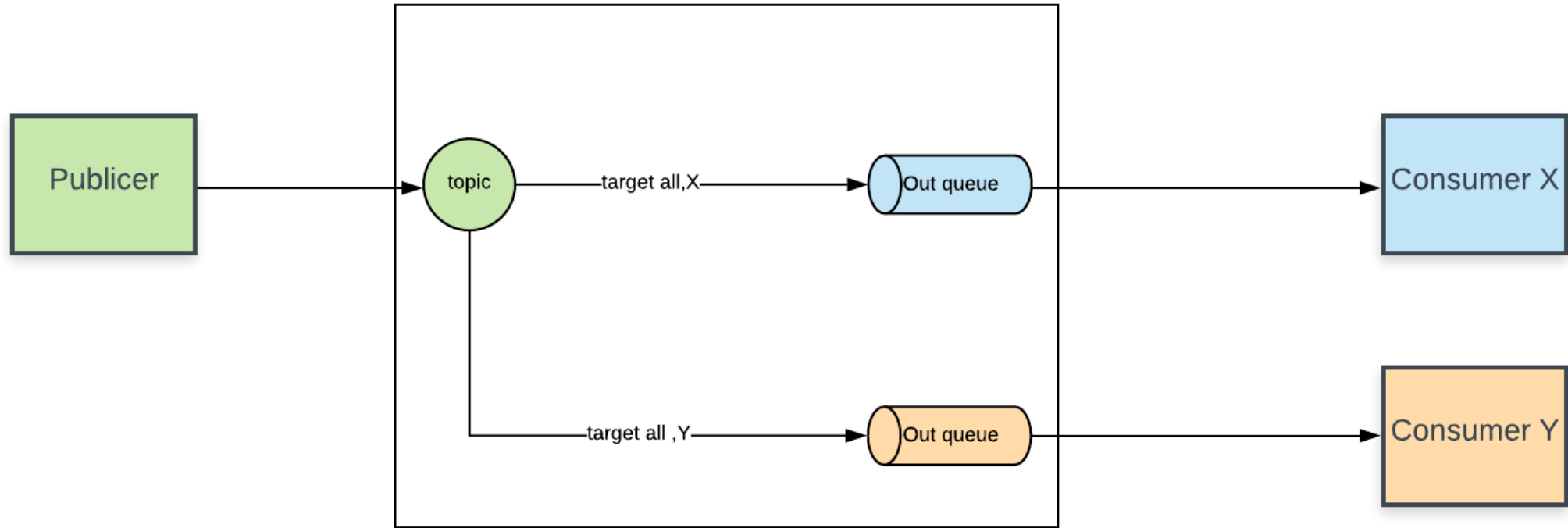
Being handed just one message
to read

=

Fast

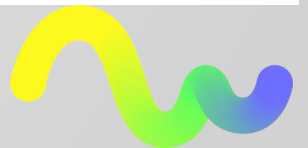


Pub - Sub using queues

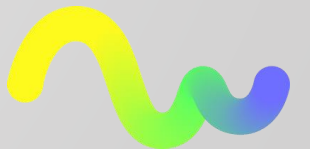
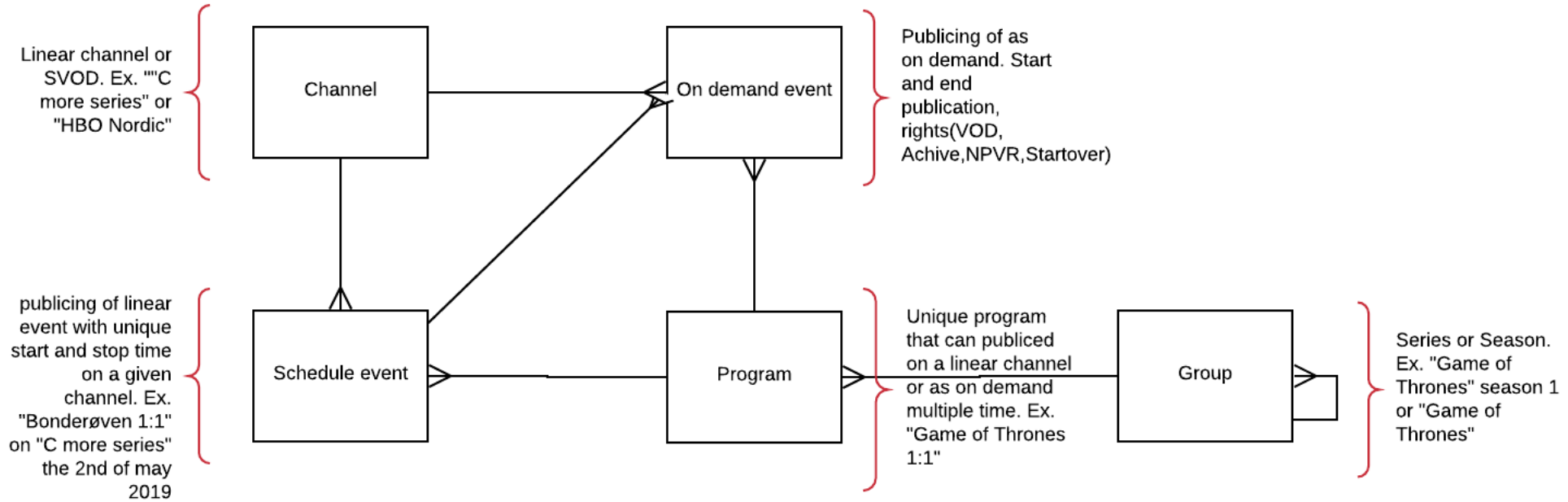


Topic name: [Entity]_Topic_[System]_[Version]
Queue name: [Entity]_[System]_[Version]

Amazon SNS / SQS setup

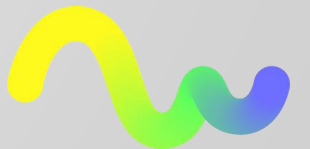
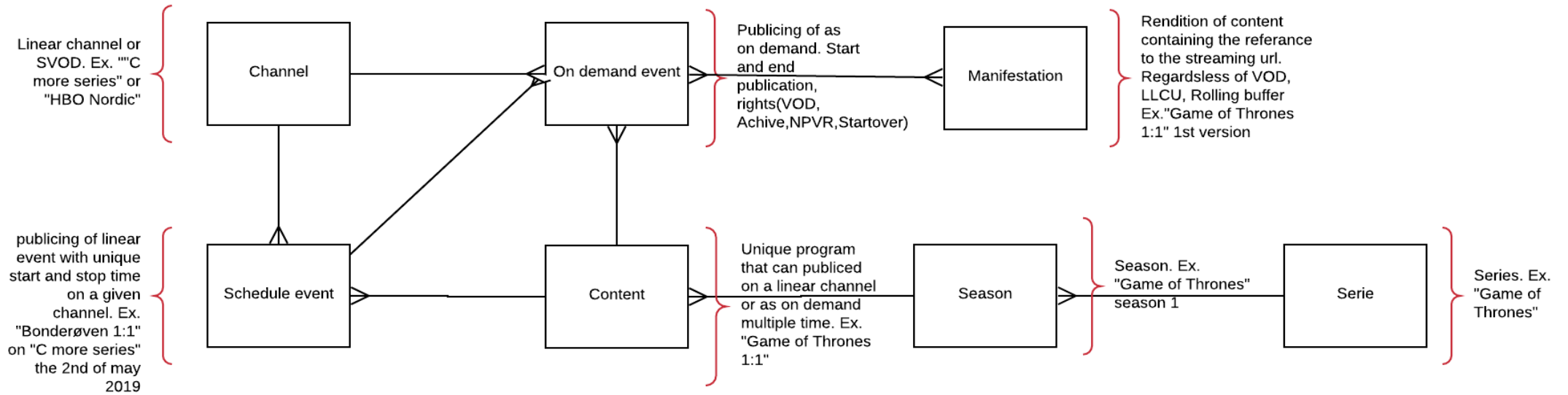


TV-Anytime



TV-Anytime

(In nuudays context)



Using an event driven model

TV Anytime document

- Document represents a schedule day or span of days

Proposed single entity message for internal use

- A message only contains a single schedule event and is only sent in case of a change

The model makes it easy quicker to communicate the relative few changes in a schedule as they happen

