



Using Message Queues (**NSQ**) with Microservice

Hell o:)

Hengki Sihombing
Co Founder & CTO

UH **URBANHIRE**

Community

Jakarta JS

Jakarta JavaScript User Group

<https://www.meetup.com/jakartajs/>

jakartajs-join.herokuapp.com



NSQ at Urbanhire

What is NSQ

A realtime distributed messaging platform

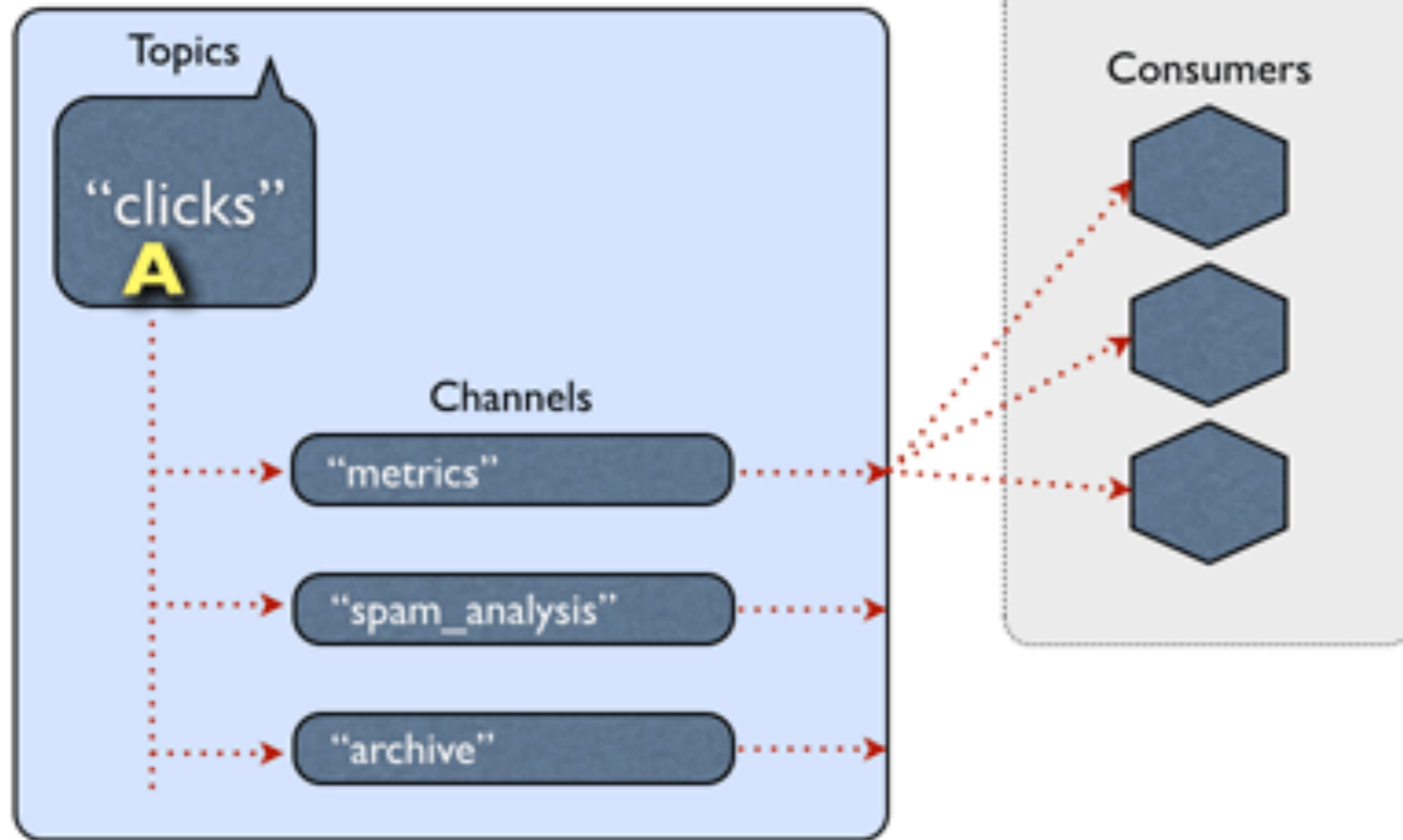
- **Open-sourced by Bitly**
- **Written in Go**

Topics and Channels

combine pubsub, distribution, and queueing

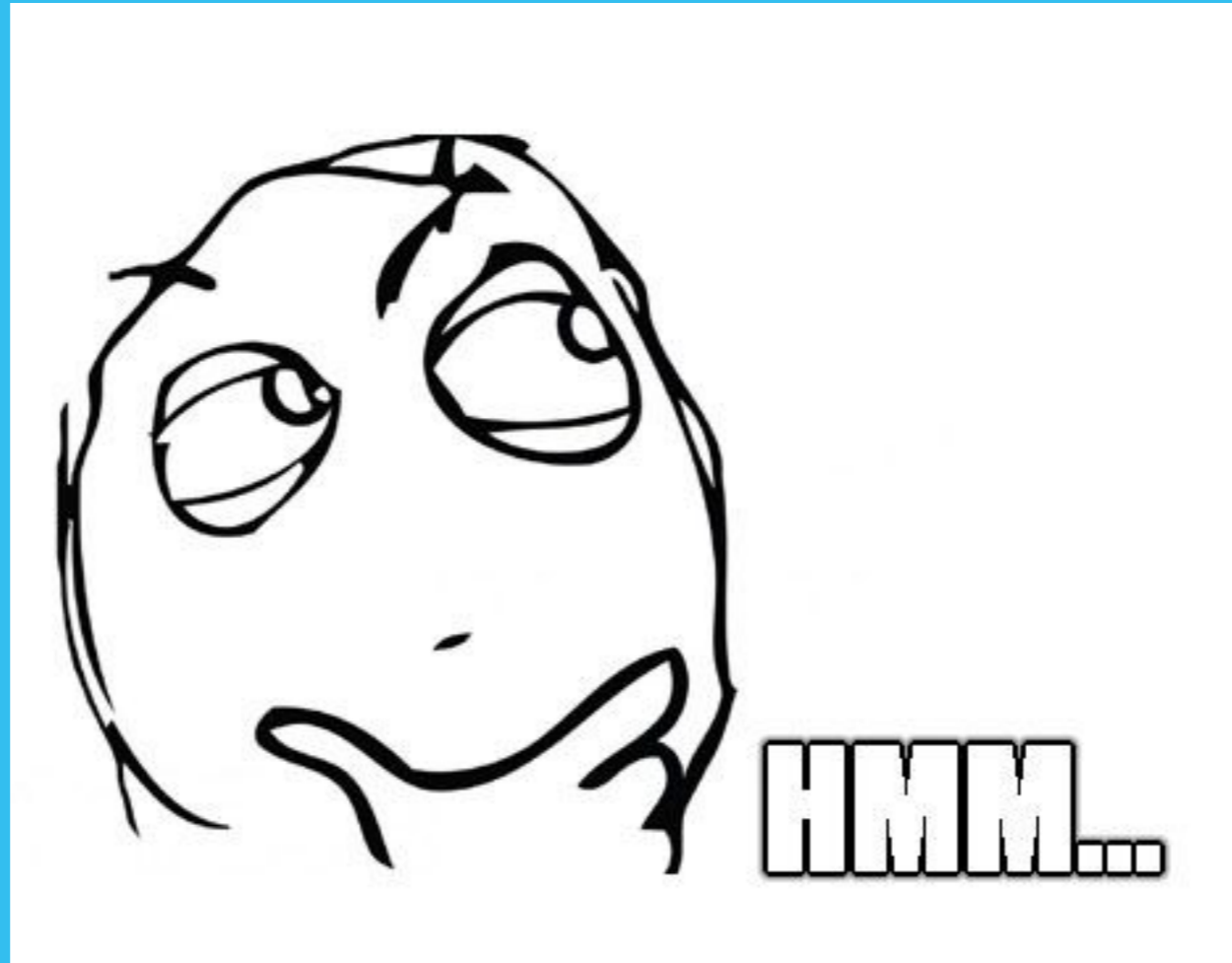
- **a topic is a distinct stream of messages**
- **a topic has one or more channels**
- **topics and channels are created at runtime**
- **messages are pushed to consumers**

nsqd



<http://nsq.io/overview/design.html>

Why We Use NSQ



Why we don't use these Libraries



Some Reason

- **Simplicity** was one of the primary reasons we decided to adopt NSQ in the first place
- We need retry with a delay mechanism
- Have an admin dashboard for monitoring and pausing queues.
- They have official library for **Node.JS**

Where we used it

- **Distribute our Job Ads to our Job Portal Partners**
- **Send email notification for interview schedule**
- **Sync jobseeker profile to Elasticsearch every time jobseeker update their profile**
- **Our Aggregator Engine**

Thank You

We're Hiring

email me: hengki@urbanhire.com

jobs.urbanhire.com/urbanhire

