

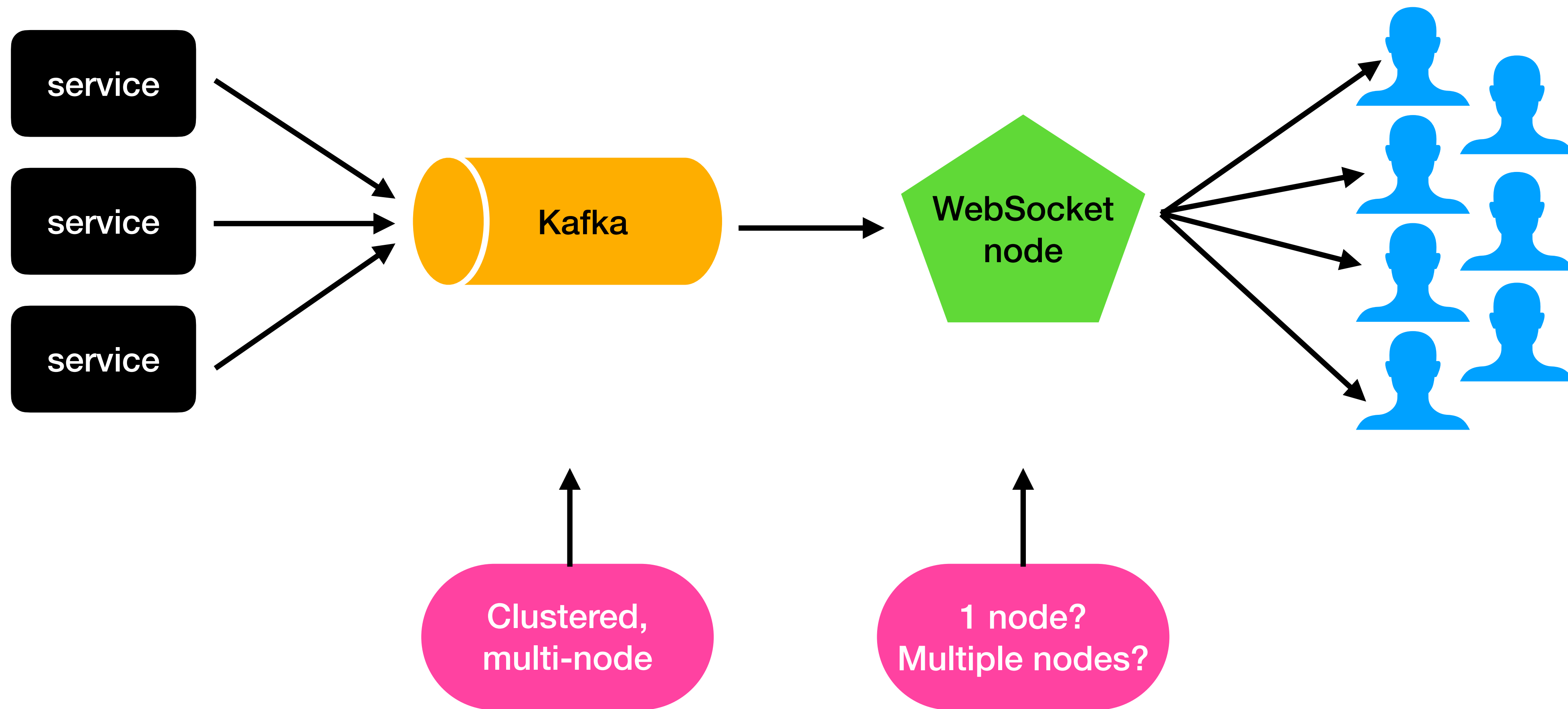
# Delivering: from Kafka to WebSockets

@adamwarski, Bolt Warsaw Tech Exchange, April 2022

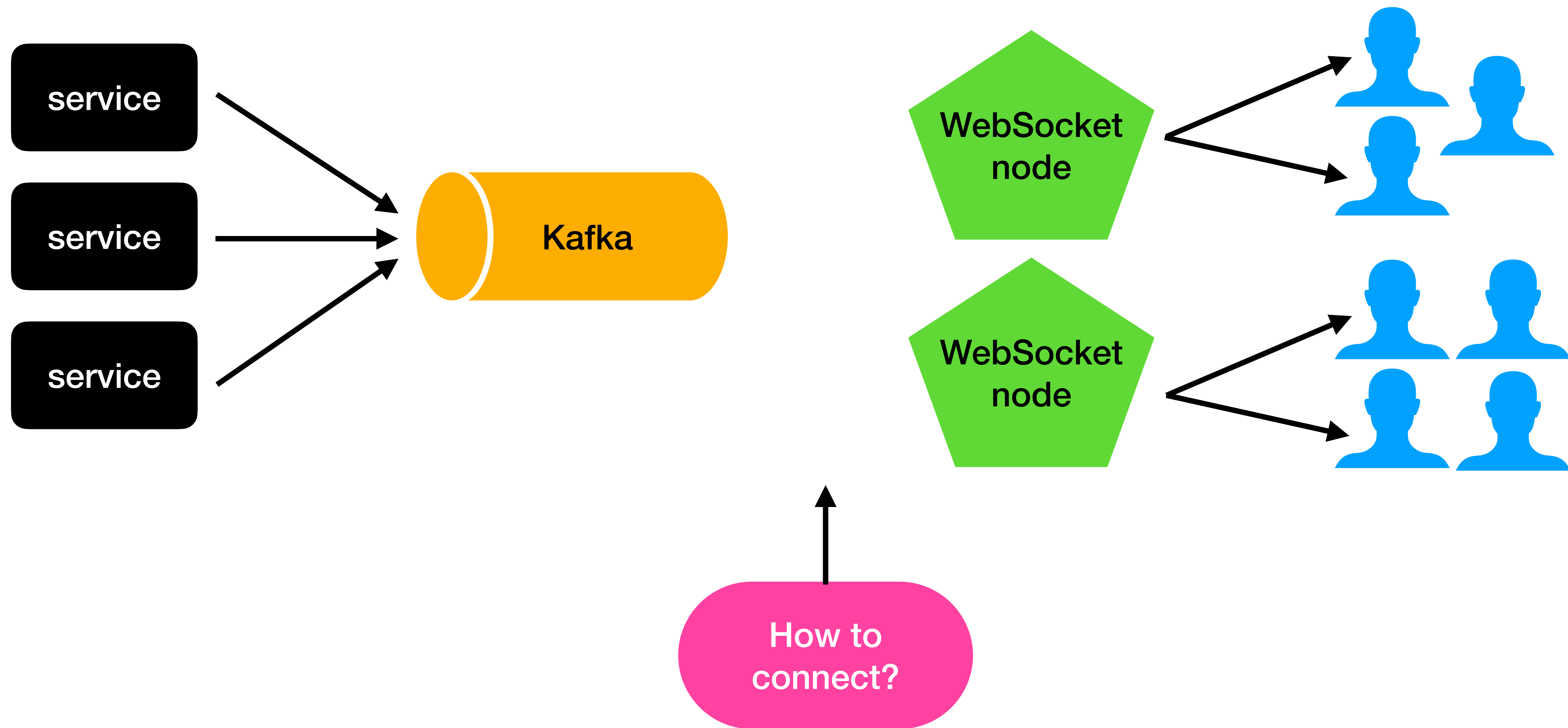


# Problem statement

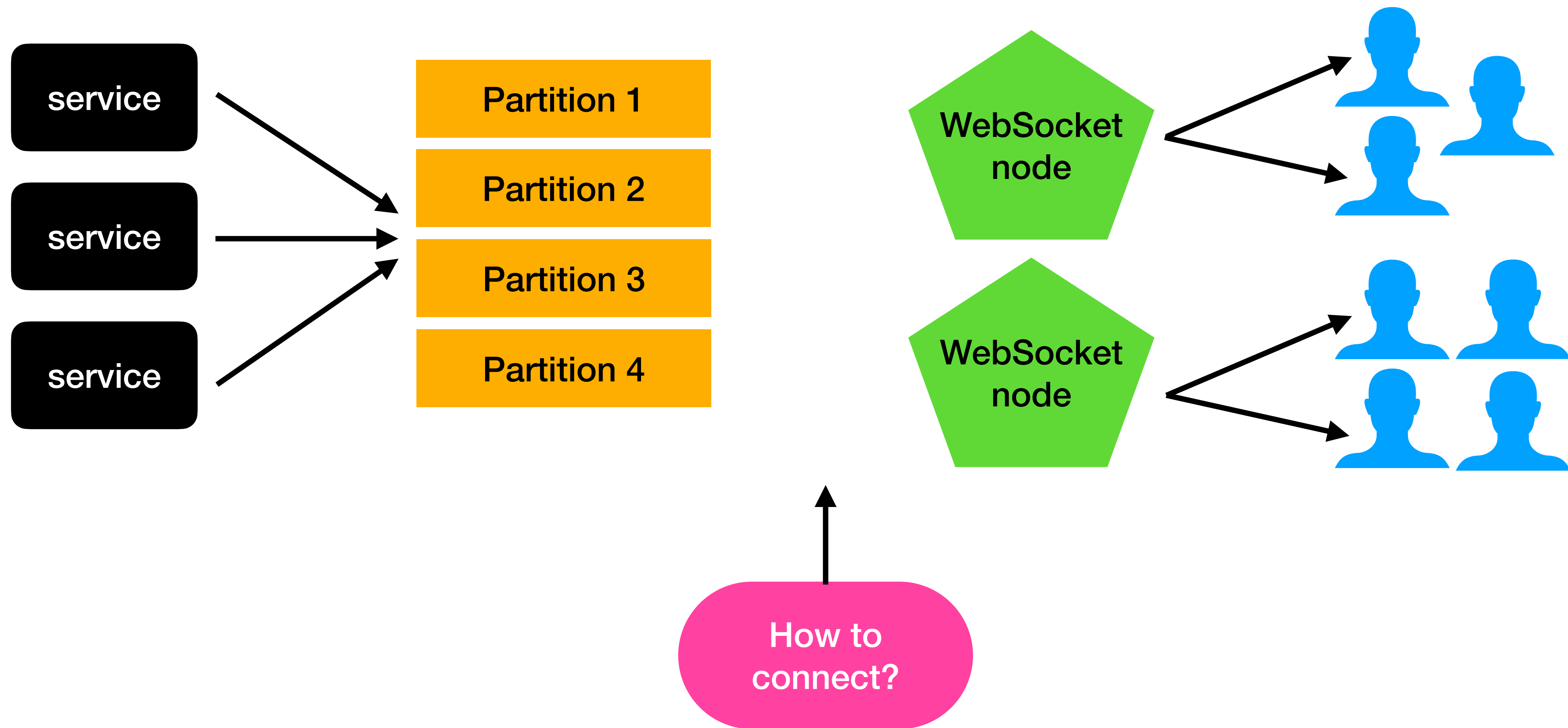
Send messages from a Kafka topic to WebSockets



# Adding more nodes



# A Kafka topic has many partitions



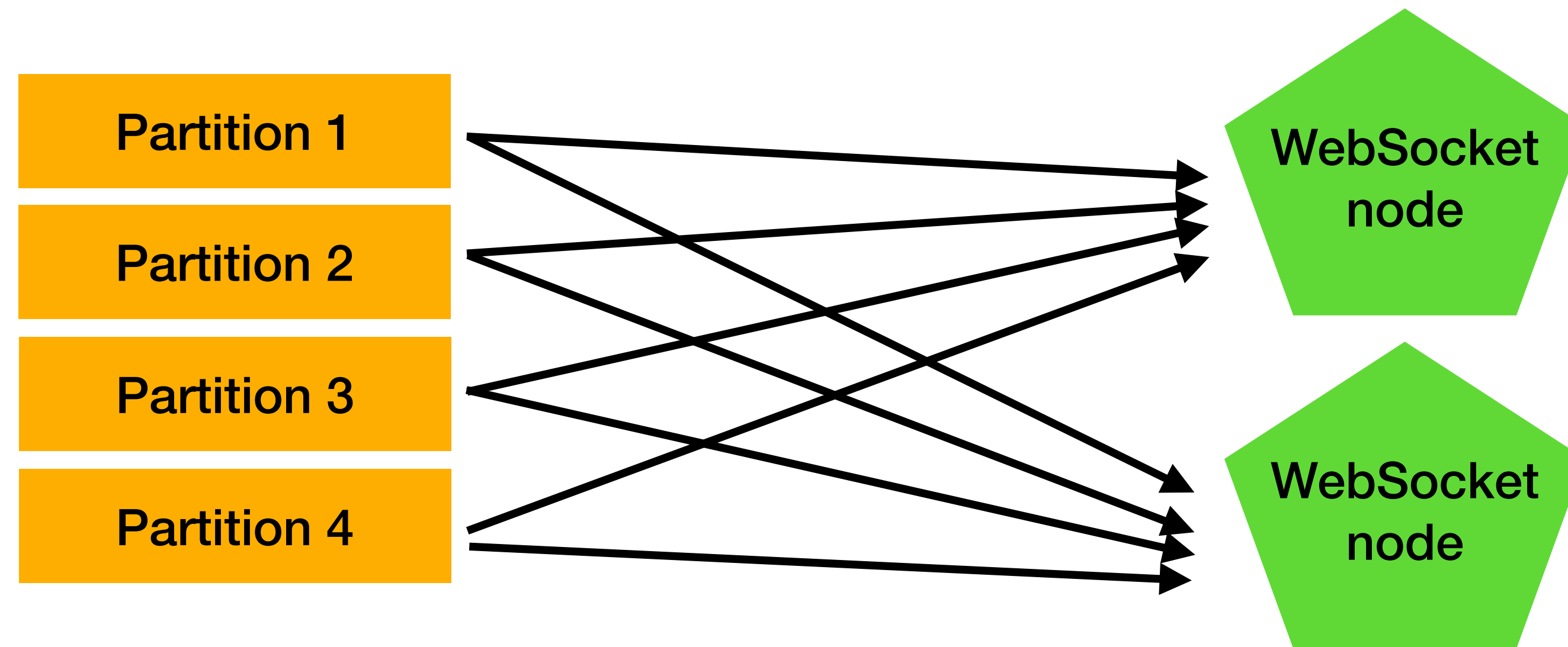
# About



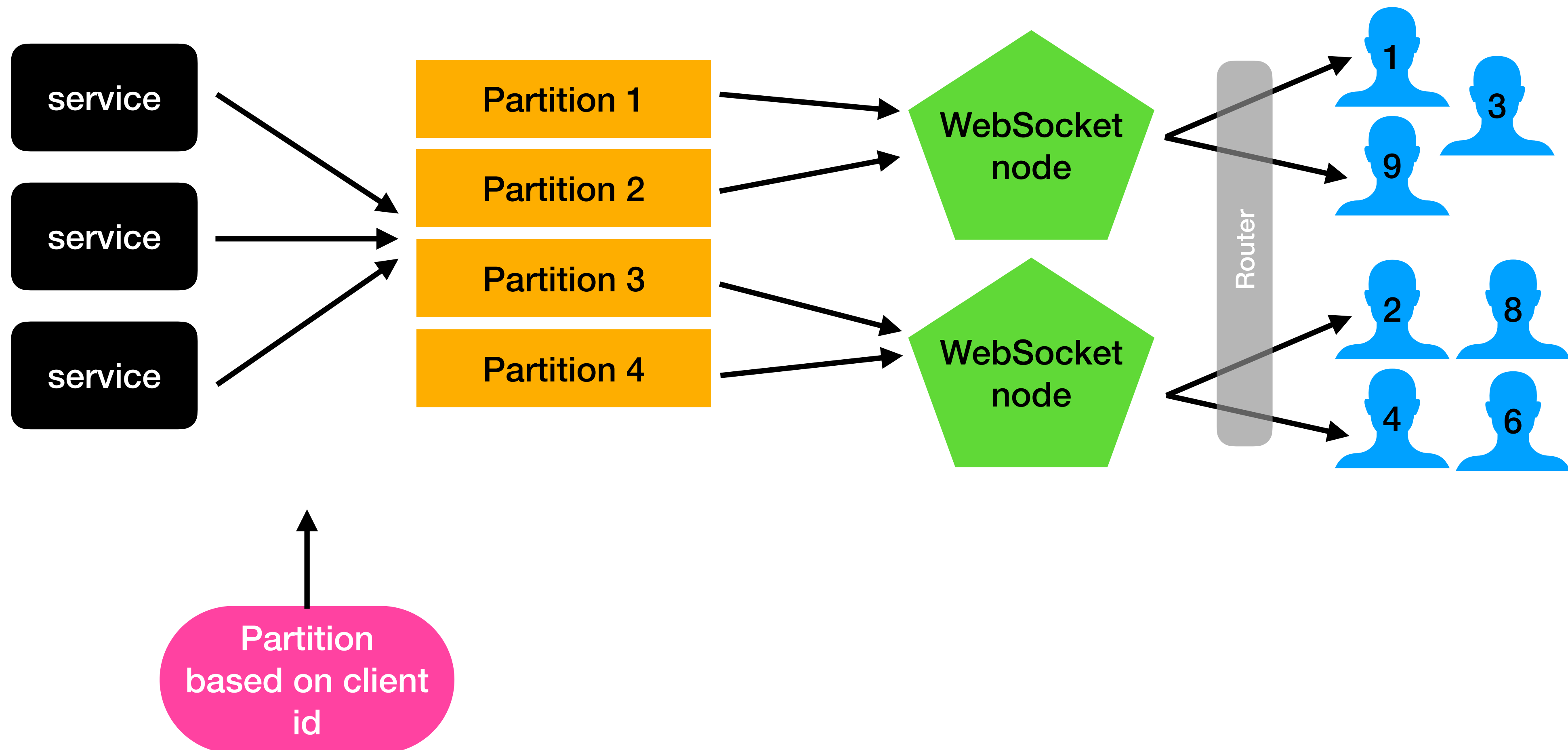
- 15 years of backend engineering
- mainly using Scala, but quite a lot of Kafka as well
- coder & CTO @ SoftwareMill
  - custom software: distributed systems, Big&Fast Data, ML/AI, Blockchain, Messaging, ...
  - Scala/Kafka/Cassandra consulting
- Blog: <http://www.softwaremill.com/blog>
- Twitter: <https://twitter.com/adamwarski>



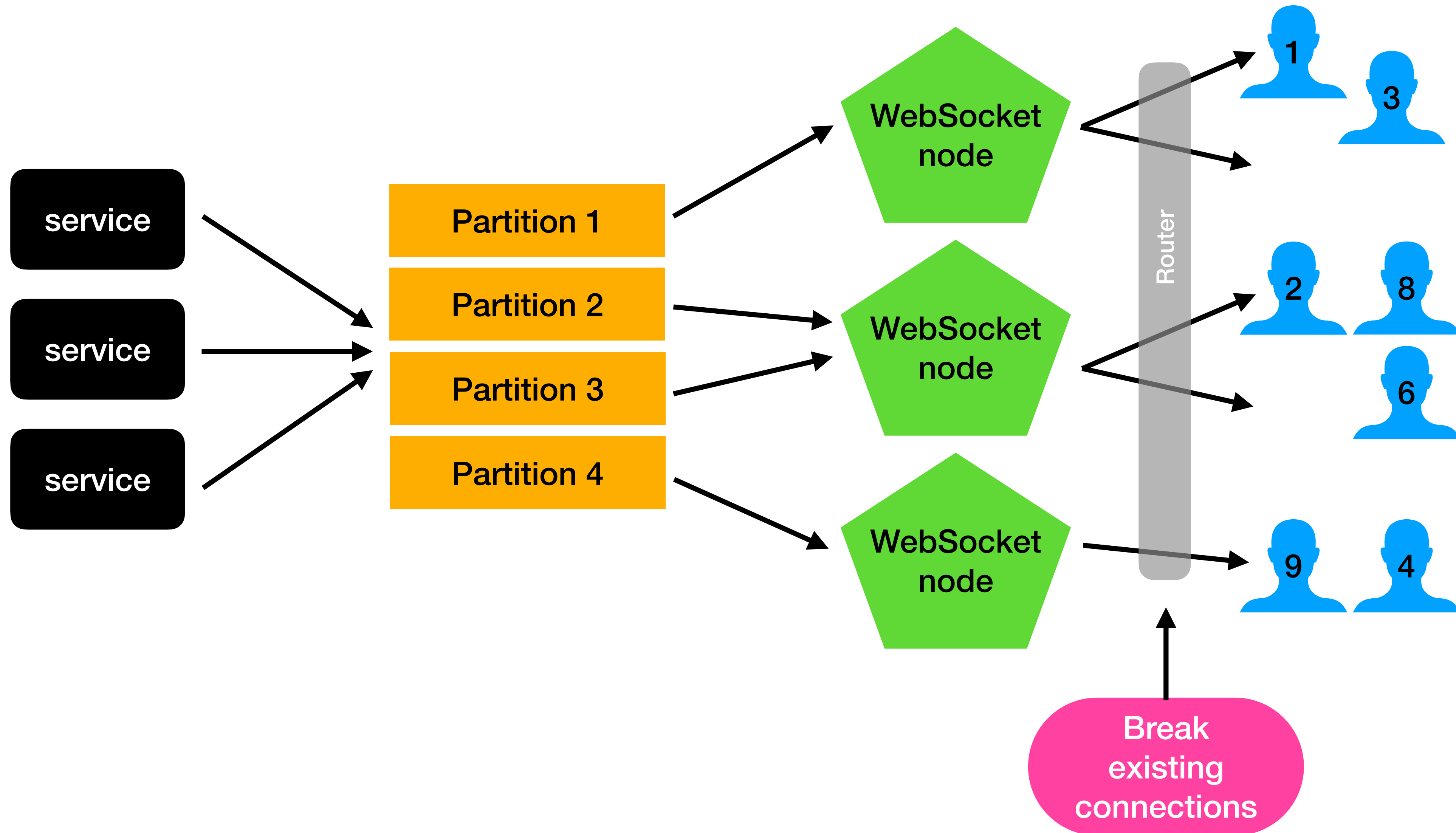
# #1: everybody reads everything



# #2: intelligent router

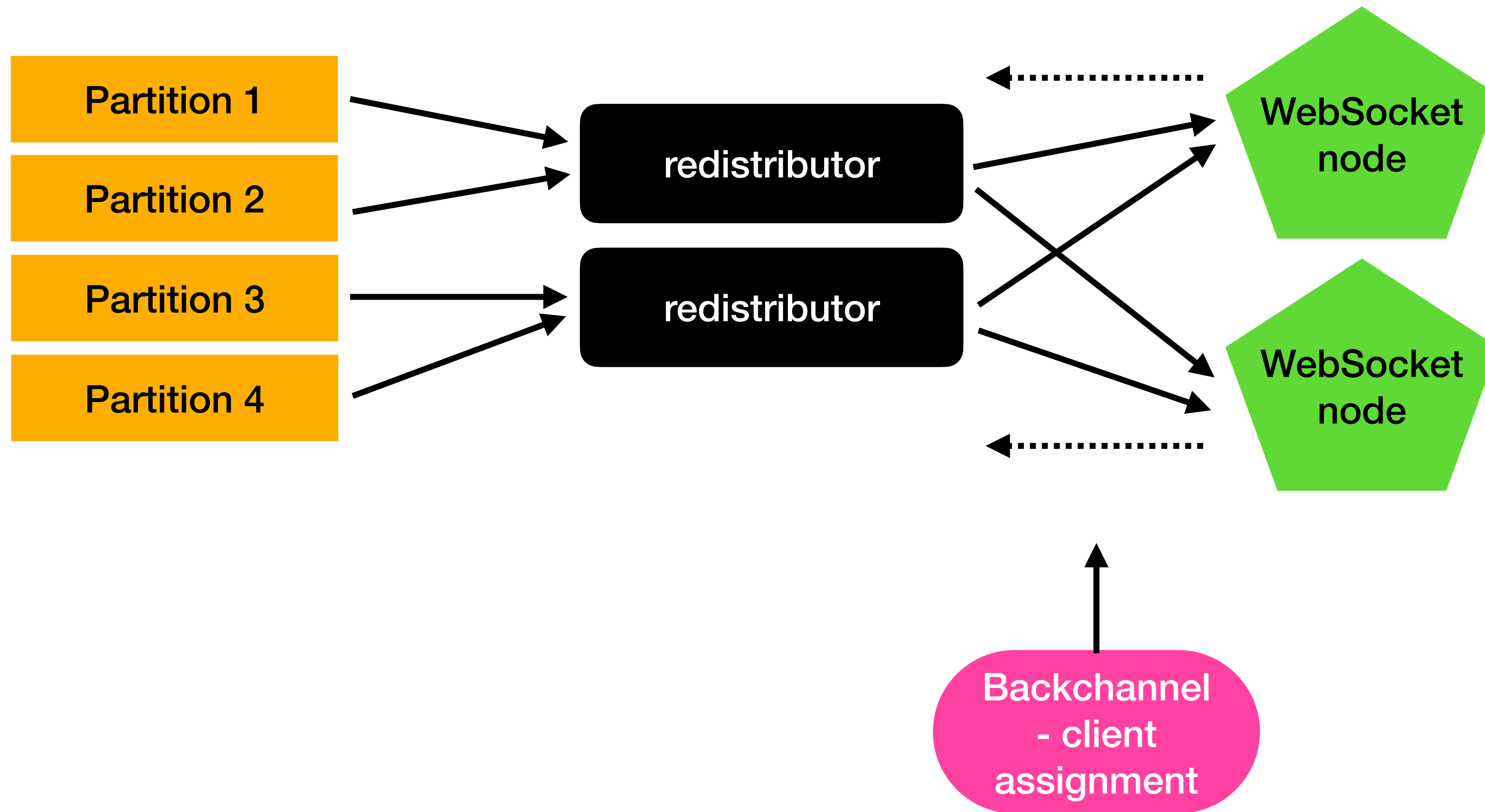


# #2: intelligent router

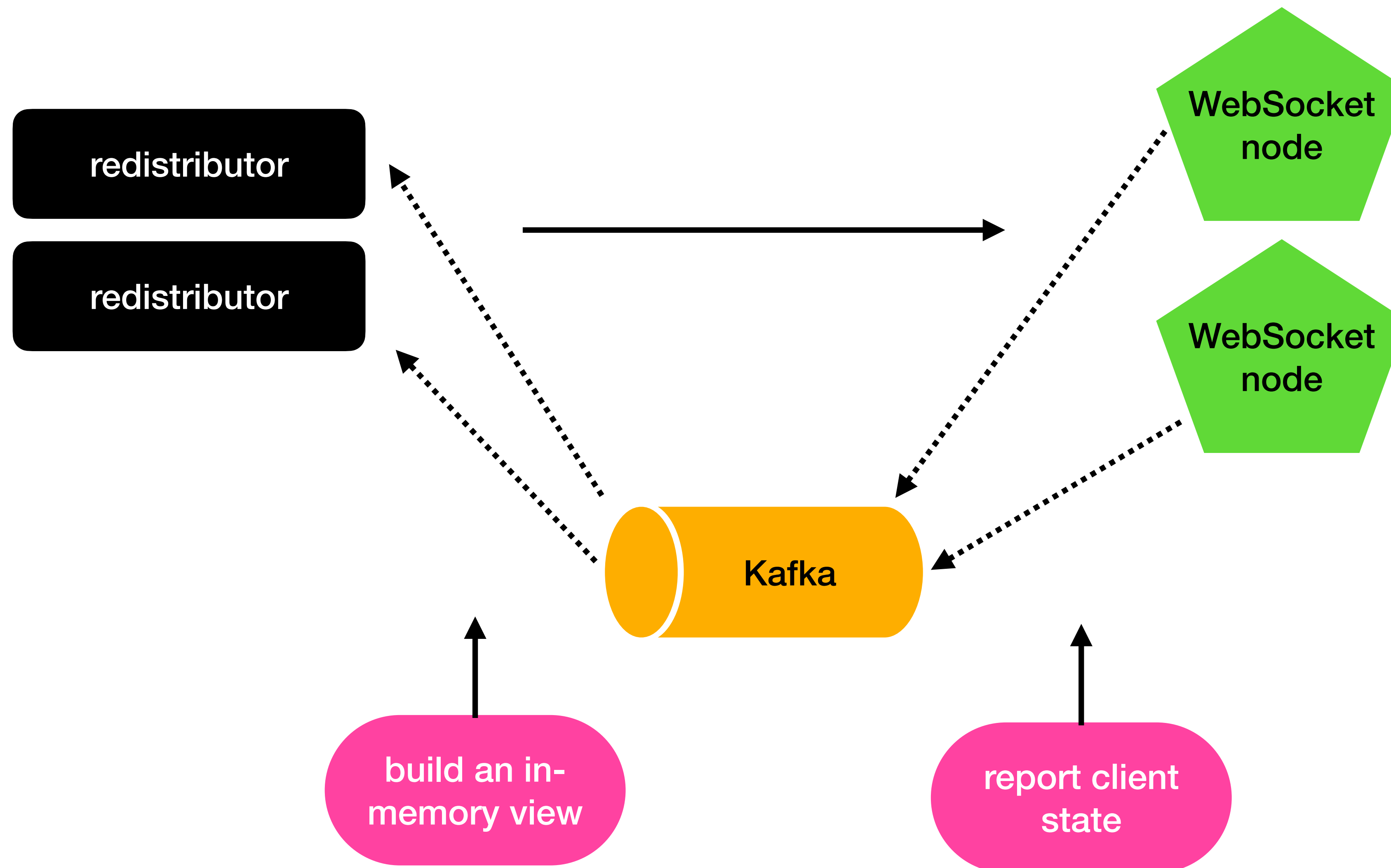




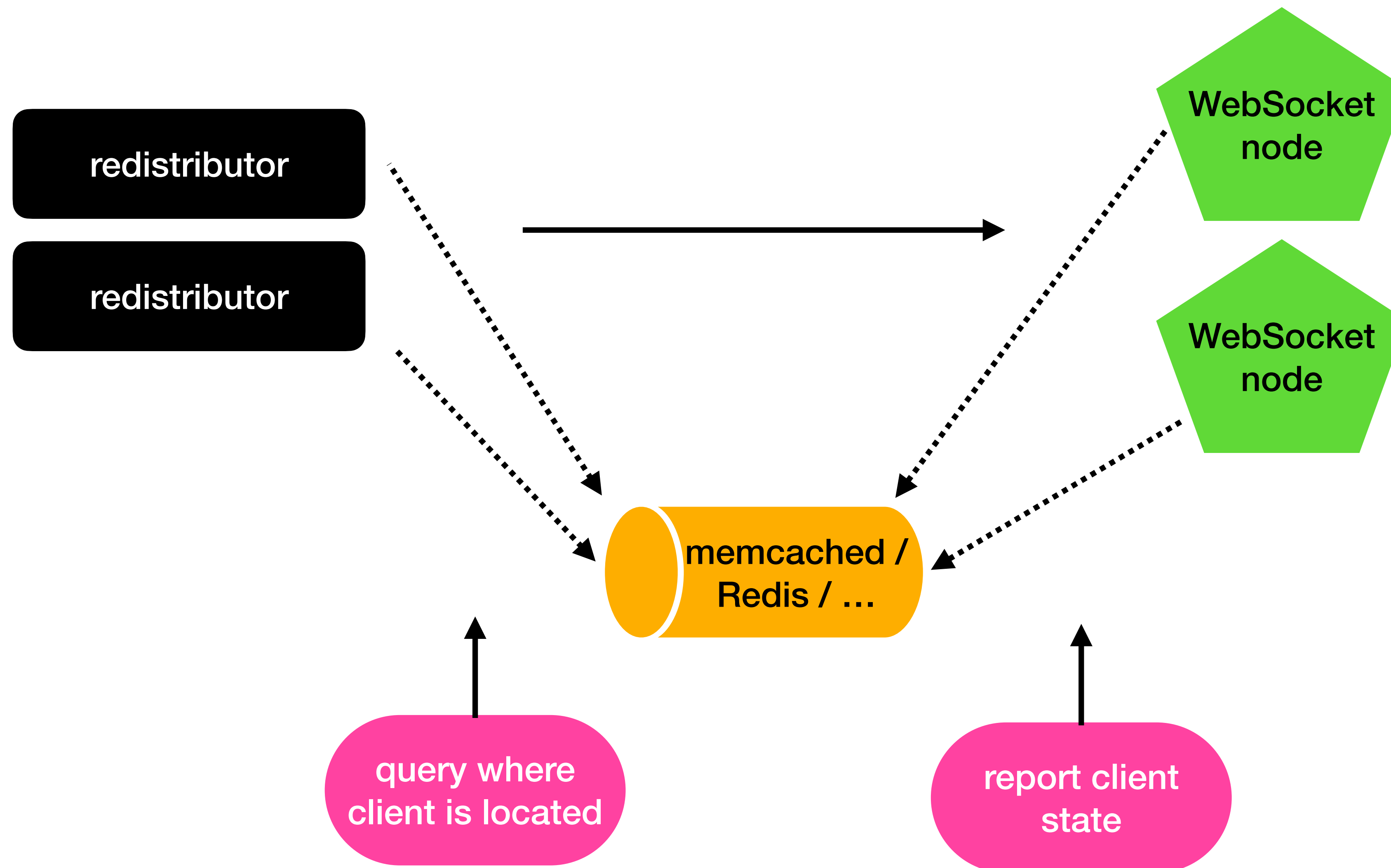
# #3: redistributors



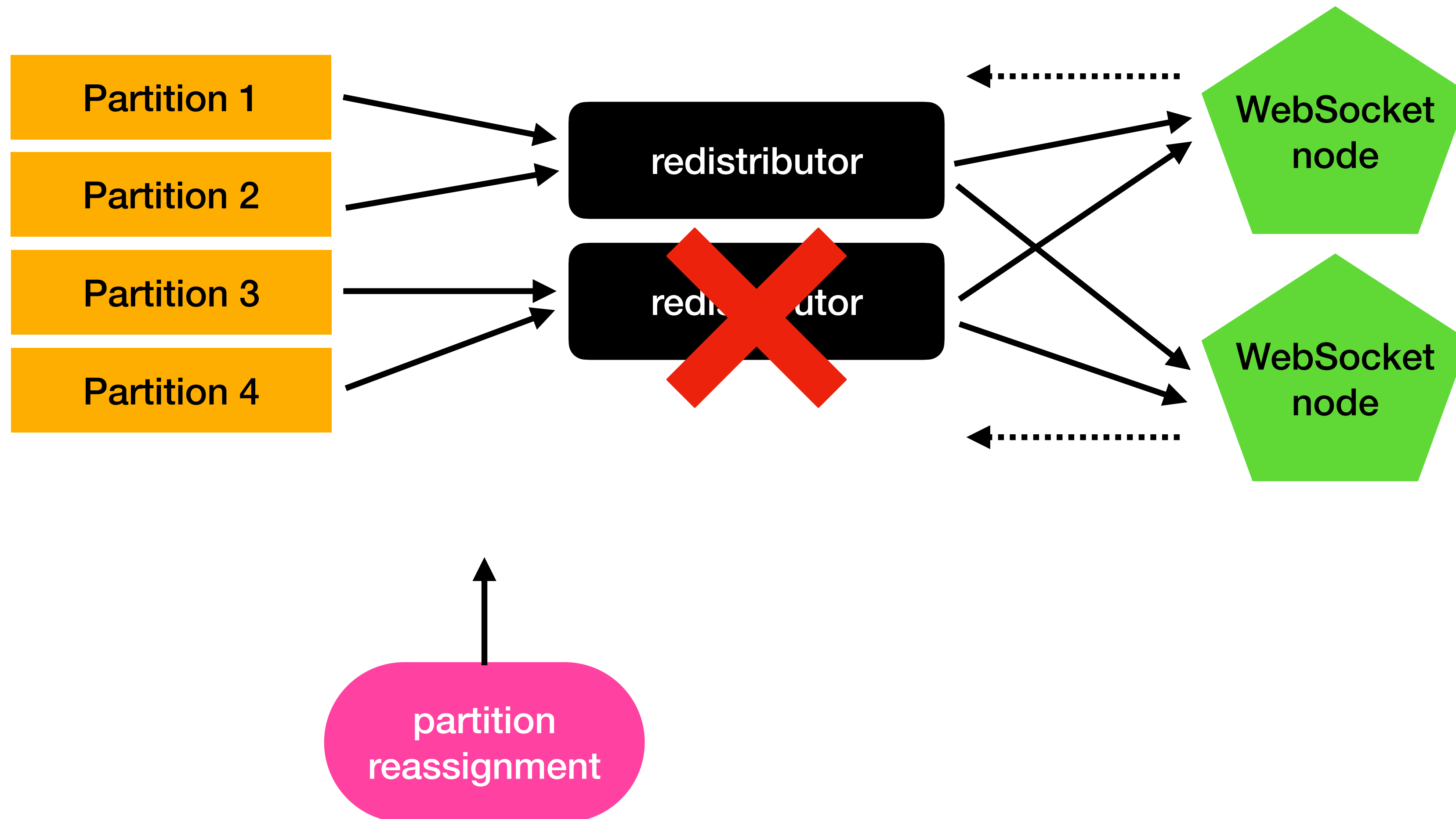
# #3.1: redistributors backchannel w/ Kafka



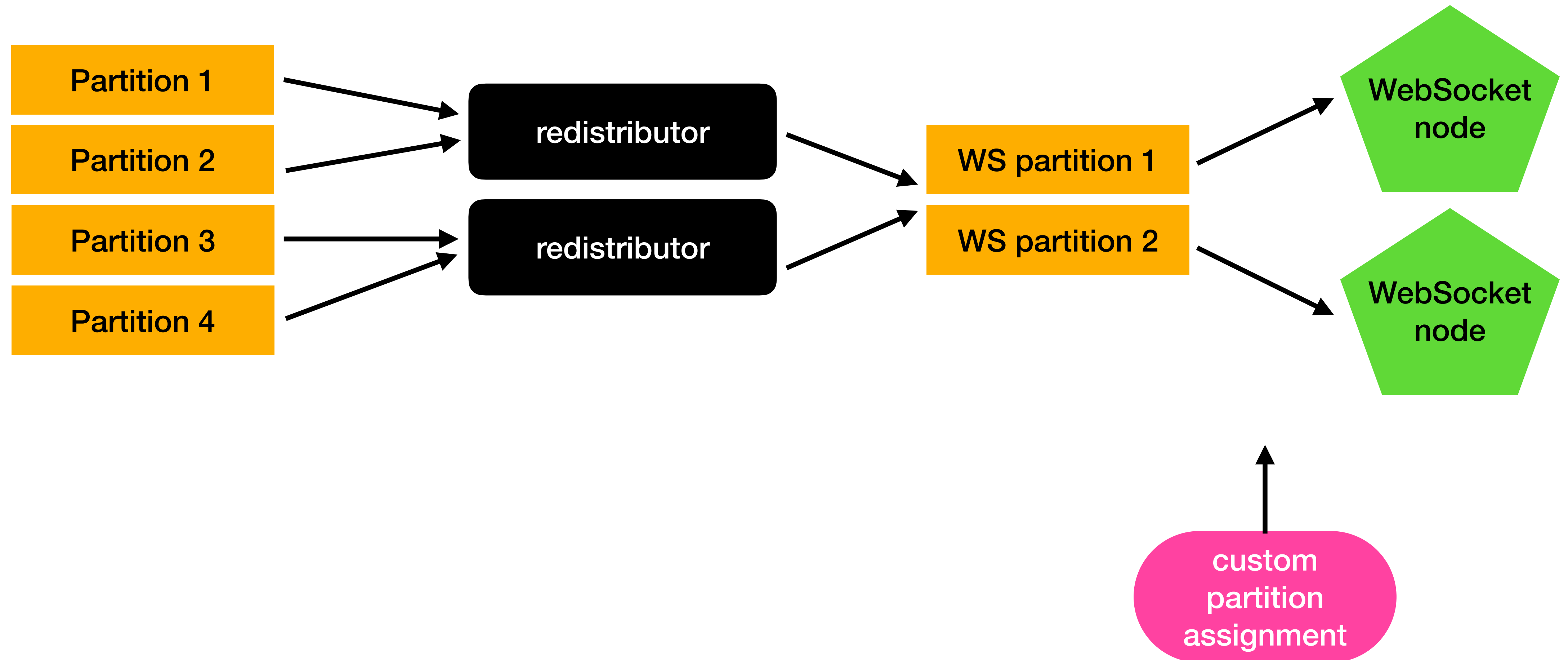
# #3.2: redistributors backchannel w/ cache



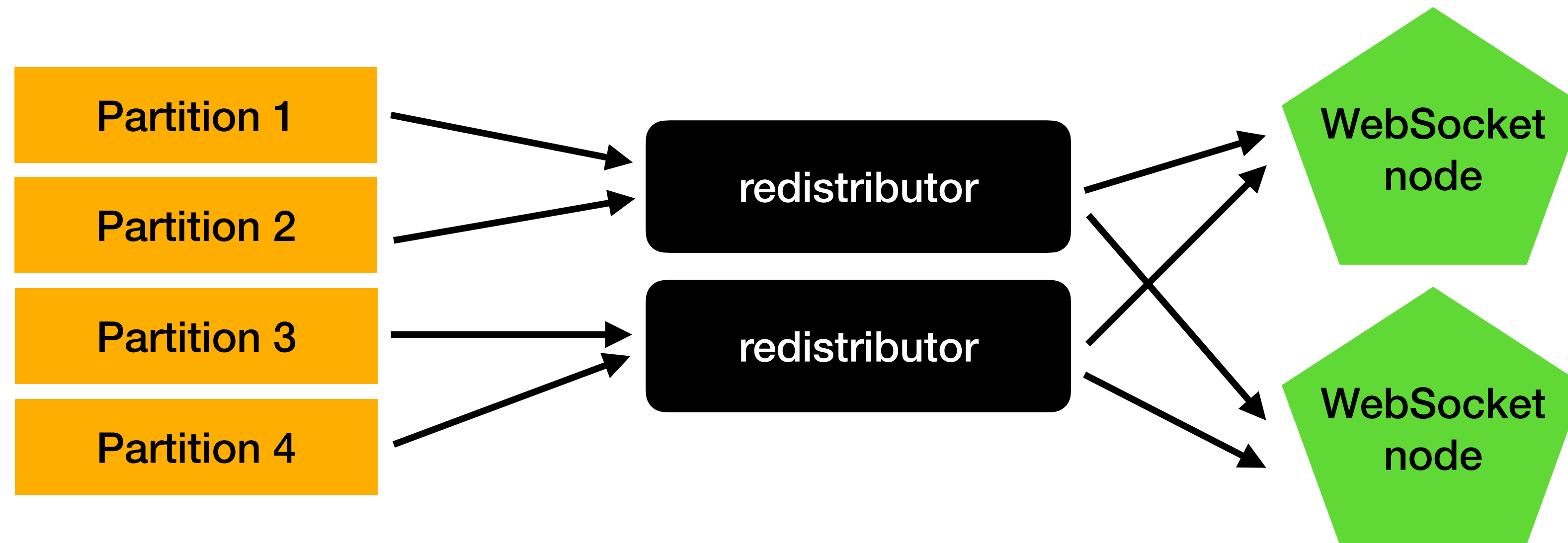
# #3: redistributors



# #3.A: forwarding using Kafka



# #3.B: forwarding using HTTP

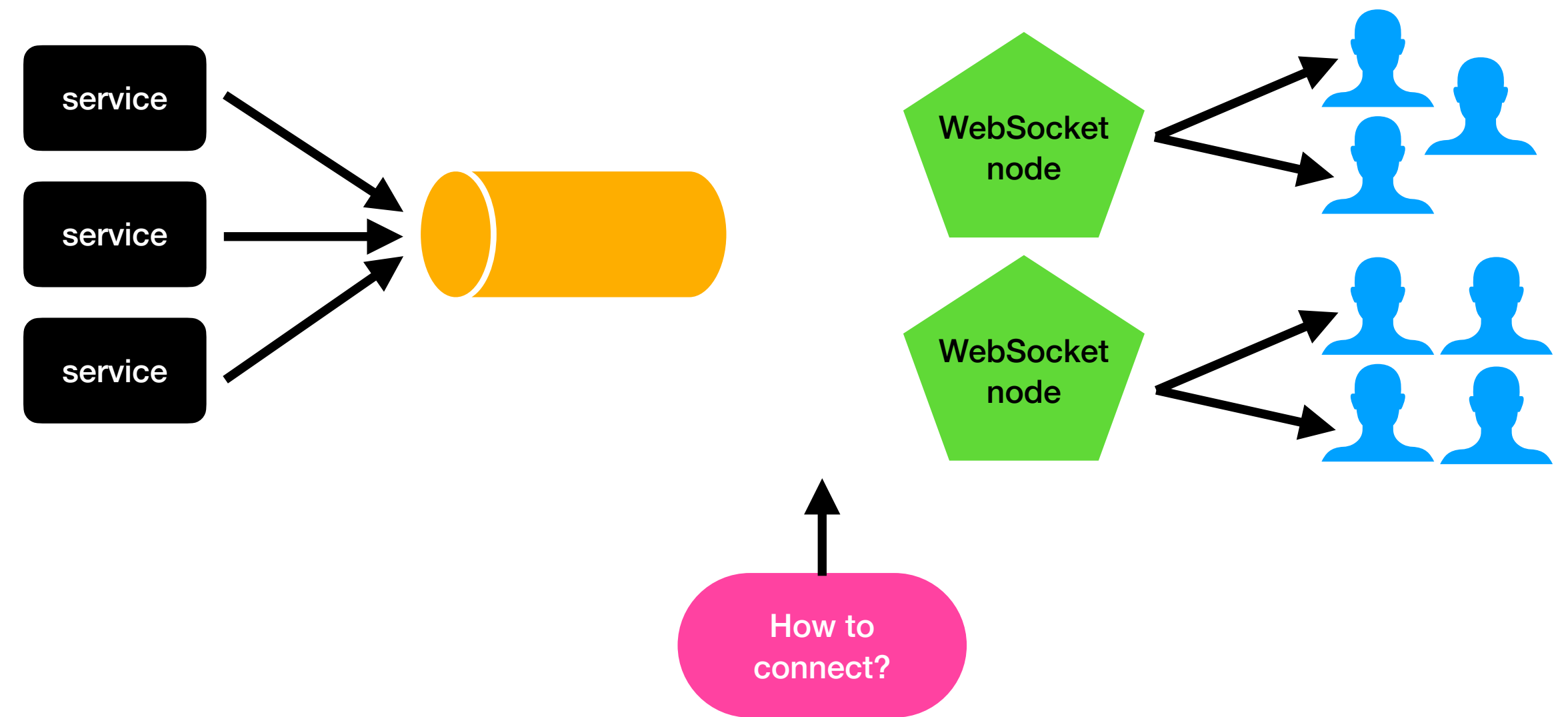


# How is Kafka influencing our design?

- Static & limited partitions
  - no partitions per client
  - for good reasons!

# Summing up

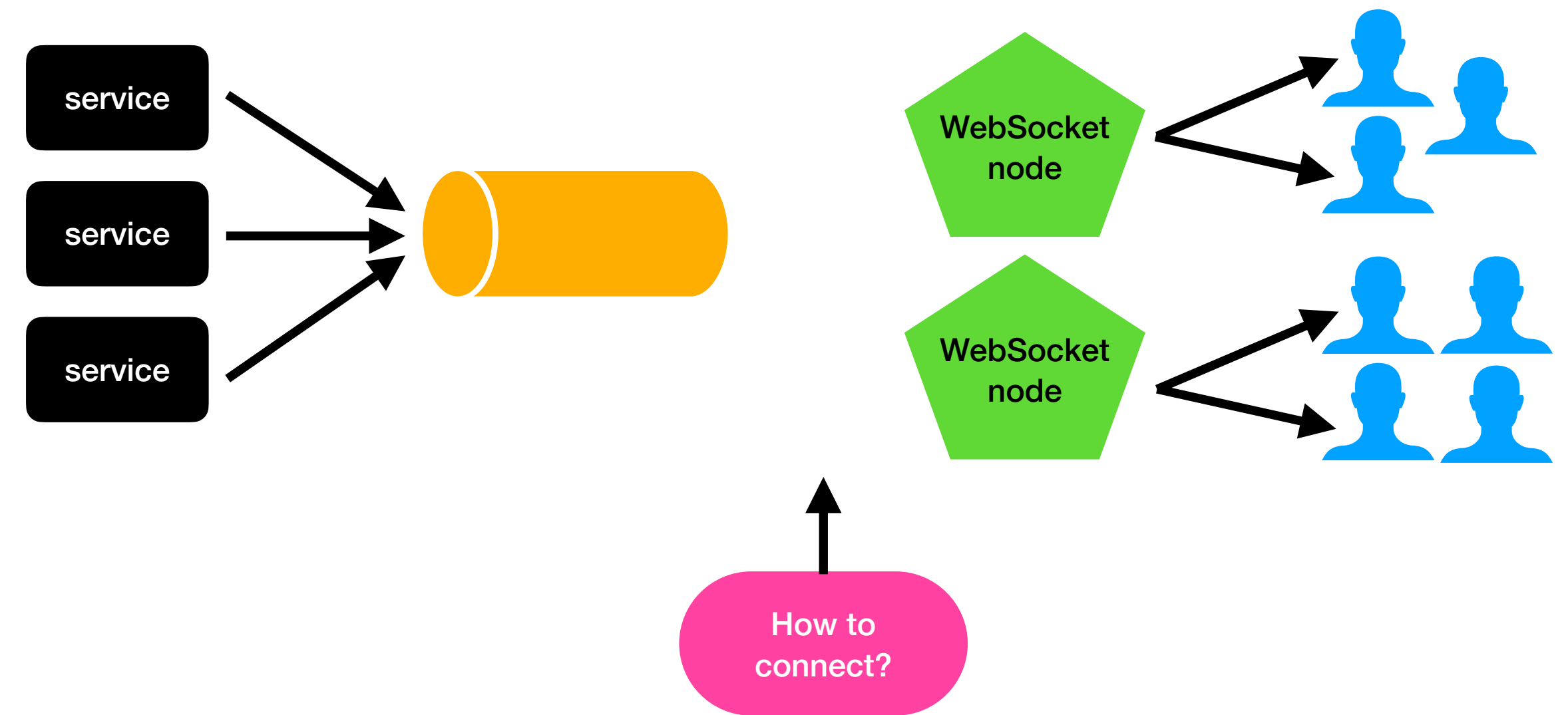
- Three possible solutions:
  - everybody reads everything
  - intelligent router
  - redistributor





# Summing up: redistributor

- Backchannel
  - using Kafka
  - using an in-memory cache
- Forwarding
  - using Kafka
  - using HTTP



# Links



- <https://kafka.apache.org>
- Real-time data pipelines and streaming ebook: <https://kafka.softwaremill.com>
- Evaluating persistent, replicated message queues: <https://softwaremill.com/mqperf/>
- Twitter: <https://twitter.com/adamwarski>



# Thank you!

@adamwarski

