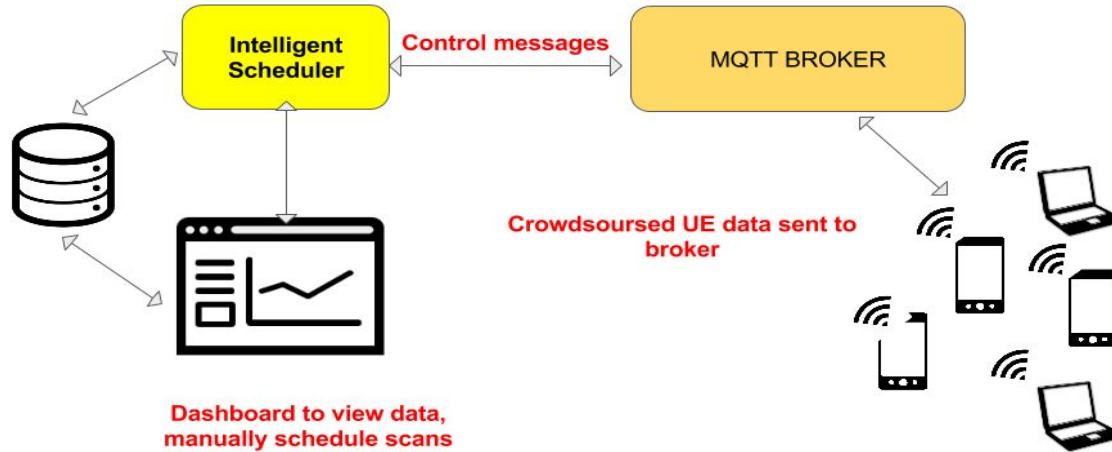

Designing a scalable crowdsourcing platform

— Using Kafka, HiveMq, HaProxy —

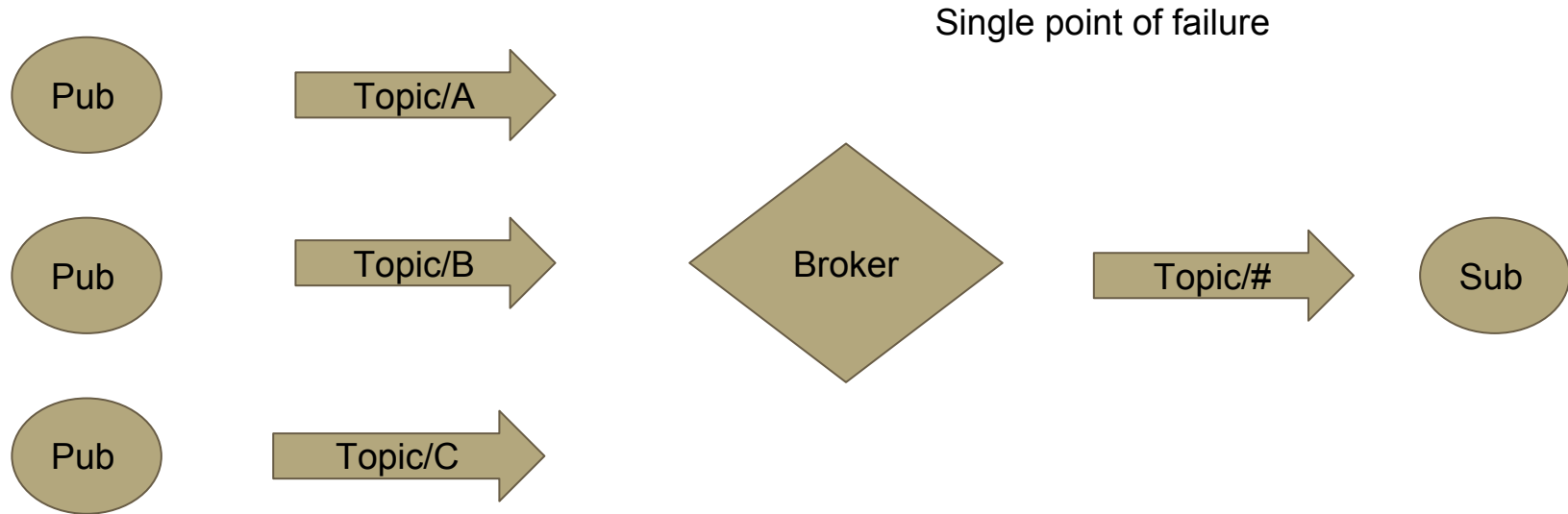
Typical Crowdsourcing Architecture



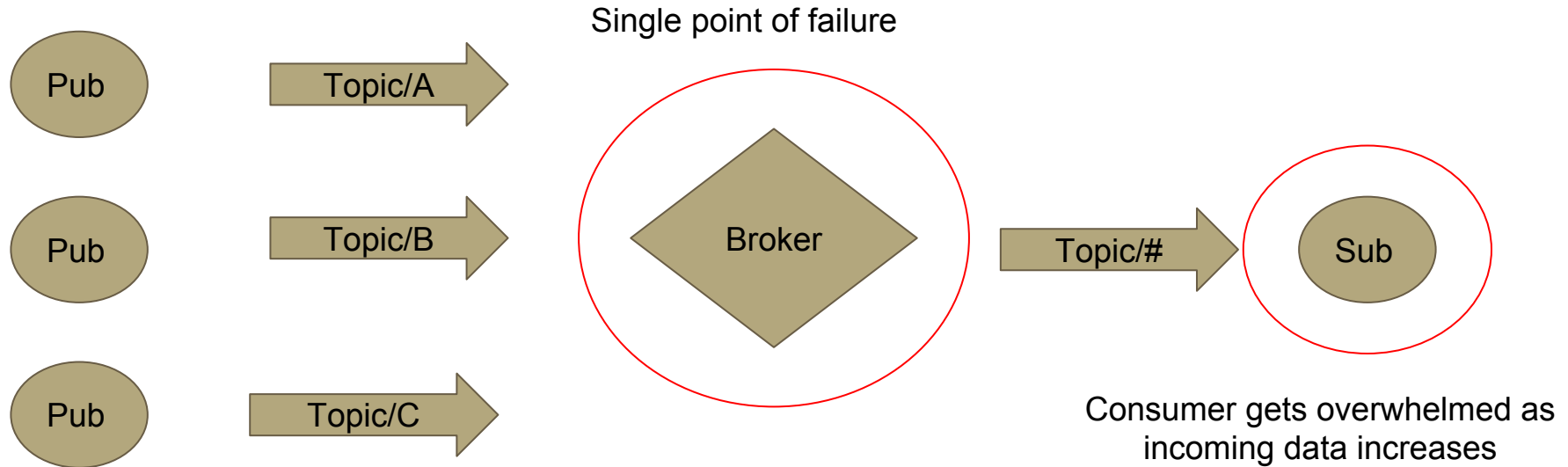
Problem - *to build a data pipeline that will:*

- 1) Build redundancy in system for achieving failovers.
- 2) Make design scalable - add more resources to handle more data.
- 3) Ensure the consumers of the messages are not overwhelmed by the data volume

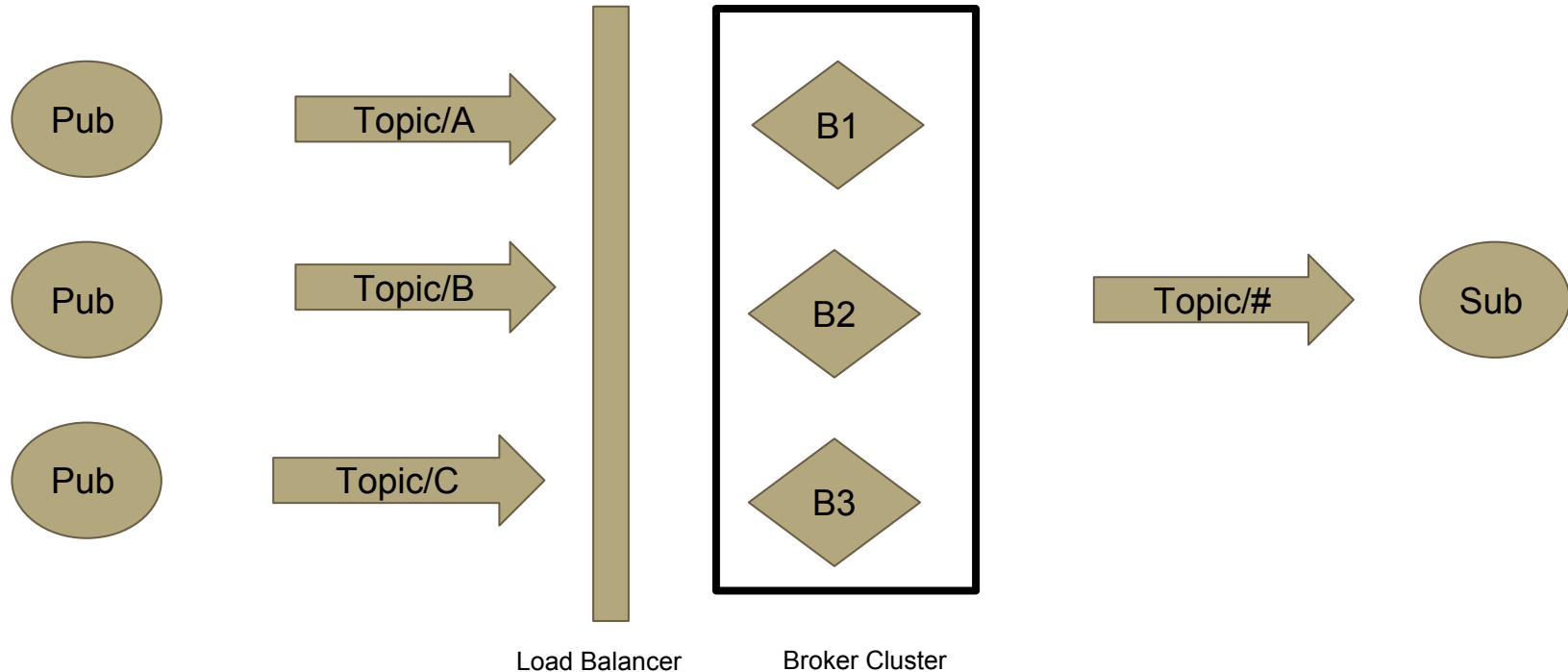
Single Broker Single Subscriber - A1



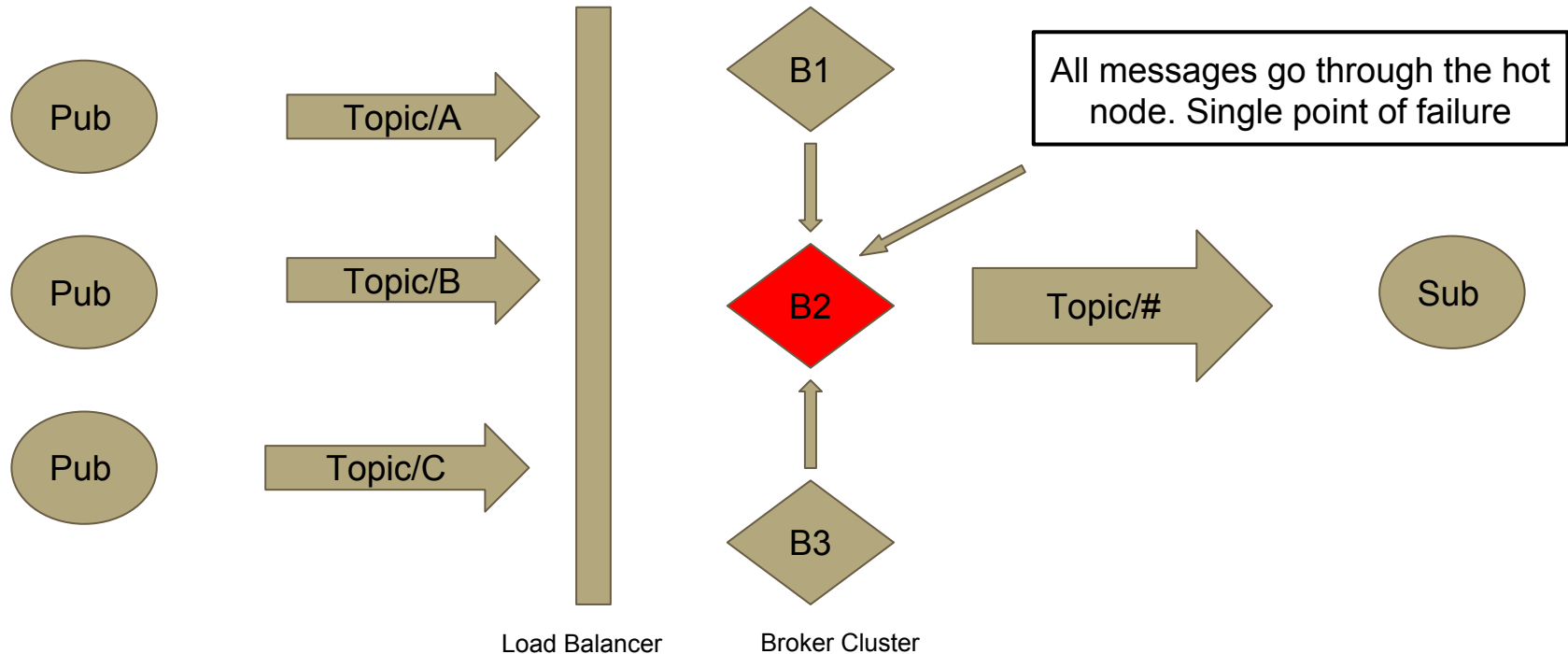
Single Broker Single Subscriber - A1



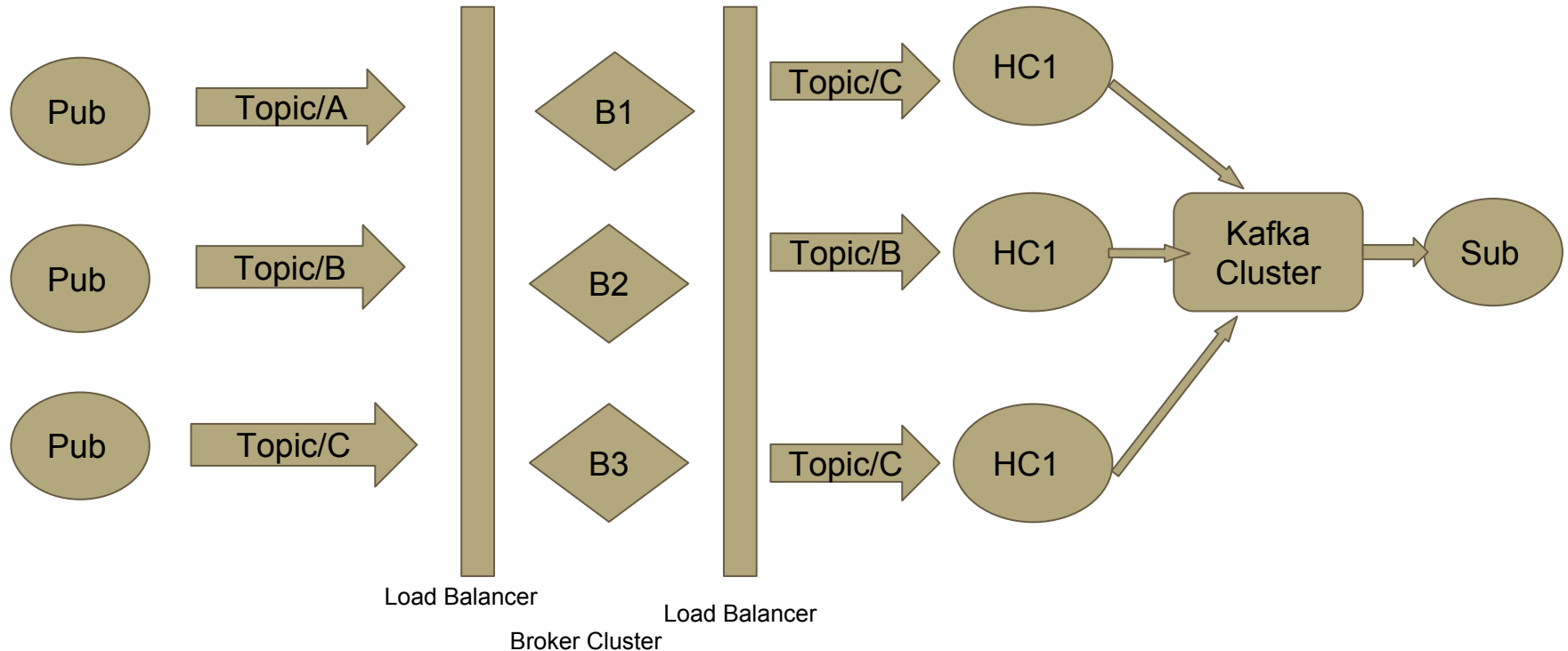
Single Subscriber with Load Balancer - A2



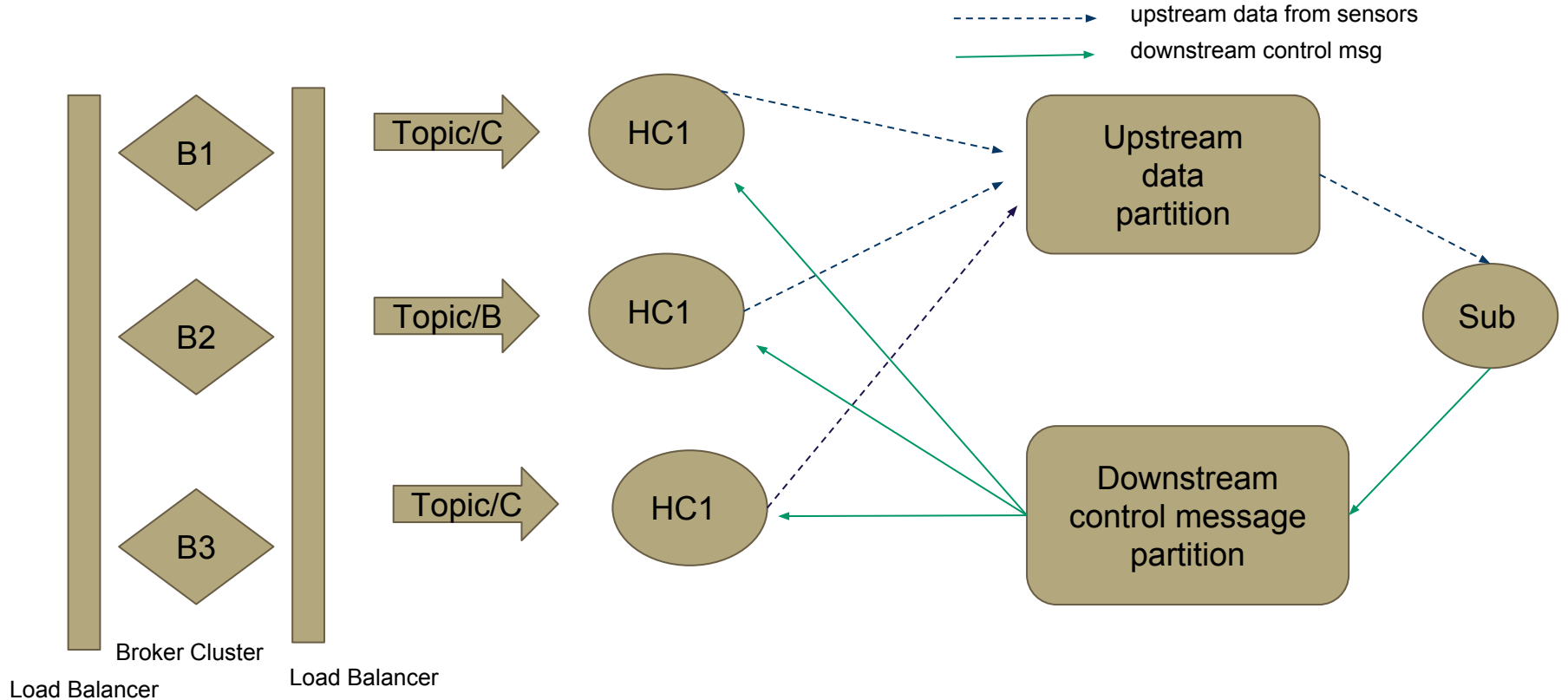
Single Subscriber with Load Balancer - A2



Replicated broker Cluster with Kafka - A3

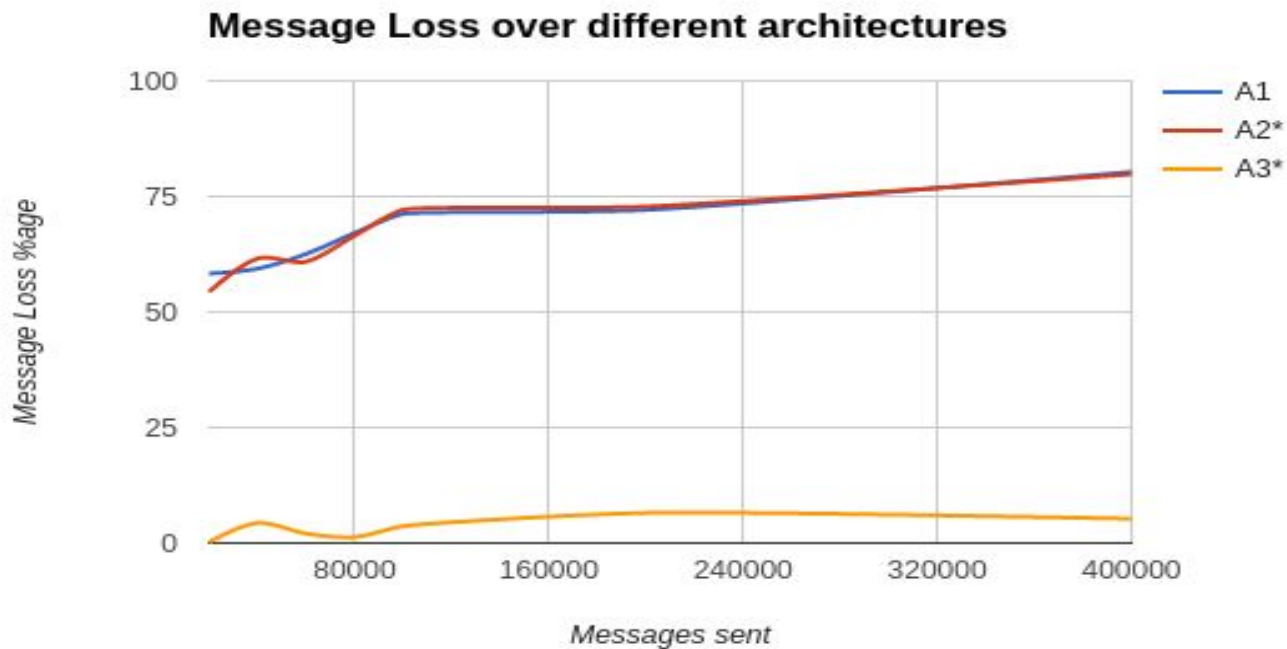


A3 - Message Flows

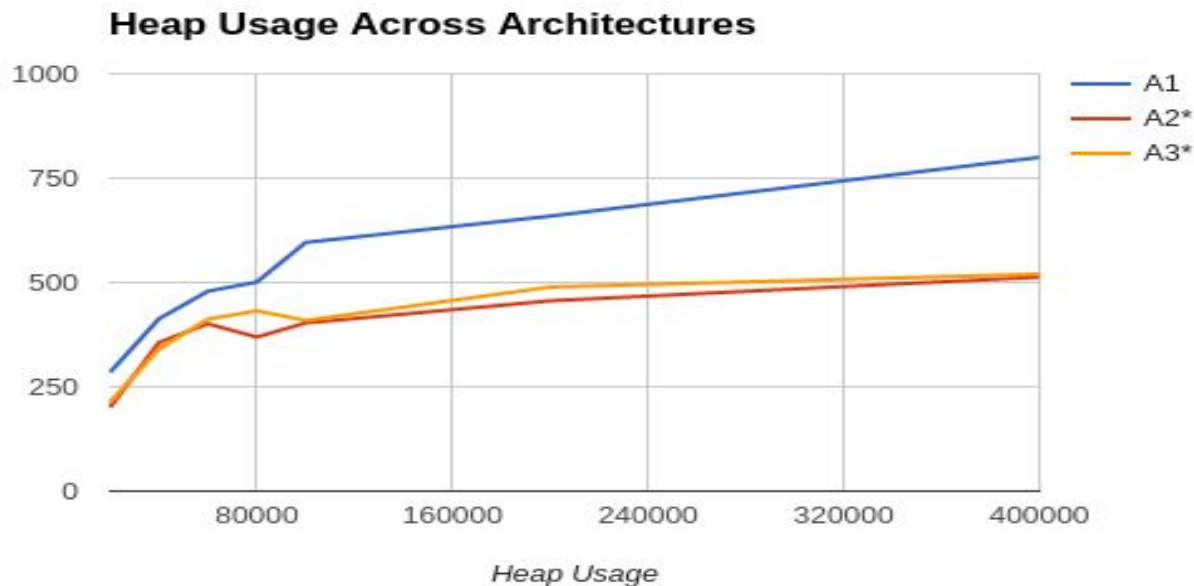


Results

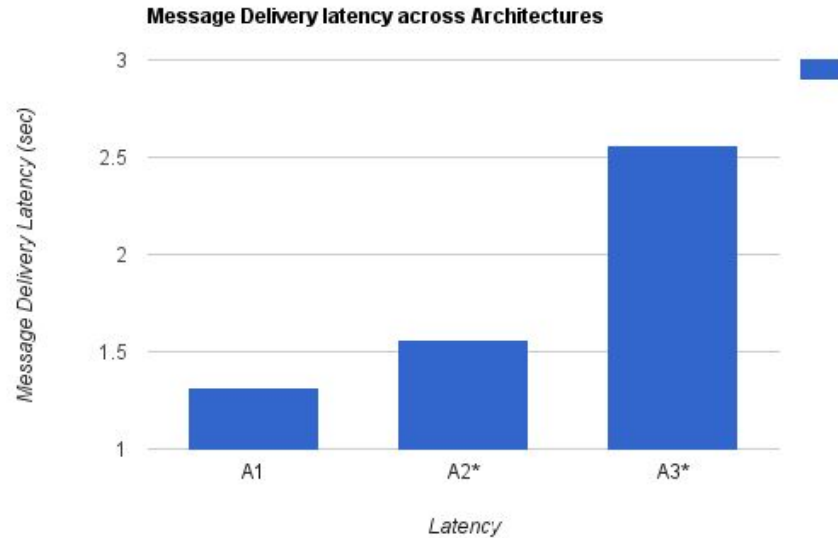
Message Loss Across Architectures



Heap Usage in Mqtt Brokers



Message Latency Across Architectures



Thank You