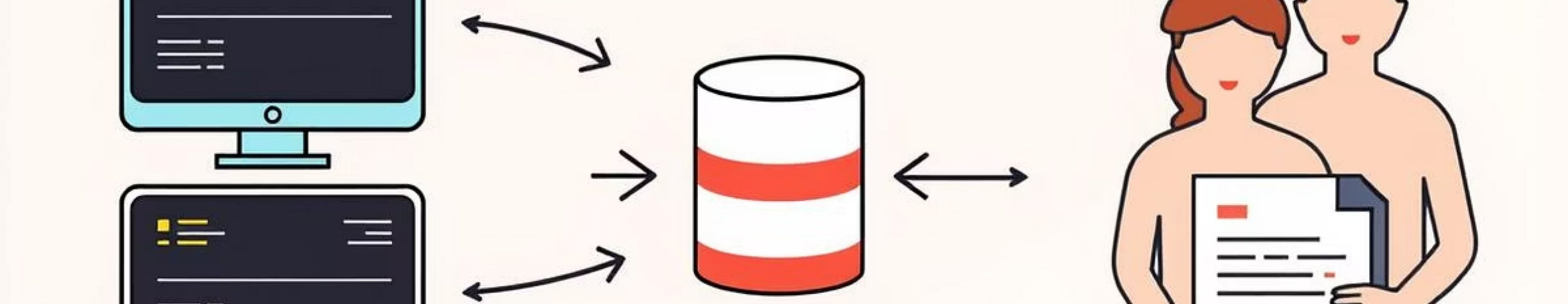
Pub/Sub Message Brokers for GenAI

Alaa Saleh, Susanna Pirttikangas, Lauri Lov´en

Introduction & Background

- What is Generative AI (GenAI)?
- Examples: ChatGPT, Image Generation, Autonomous Agents
- Why is GenAI data-hungry?
- The need for fast, reliable, scalable data communication



Role of Message Brokers in GenAI

- Brokers act as middlemen between data producers and consumers
- Crucial for GenAI apps that run across edge-cloud environments
- Used to manage queues, filter messages, balance loads

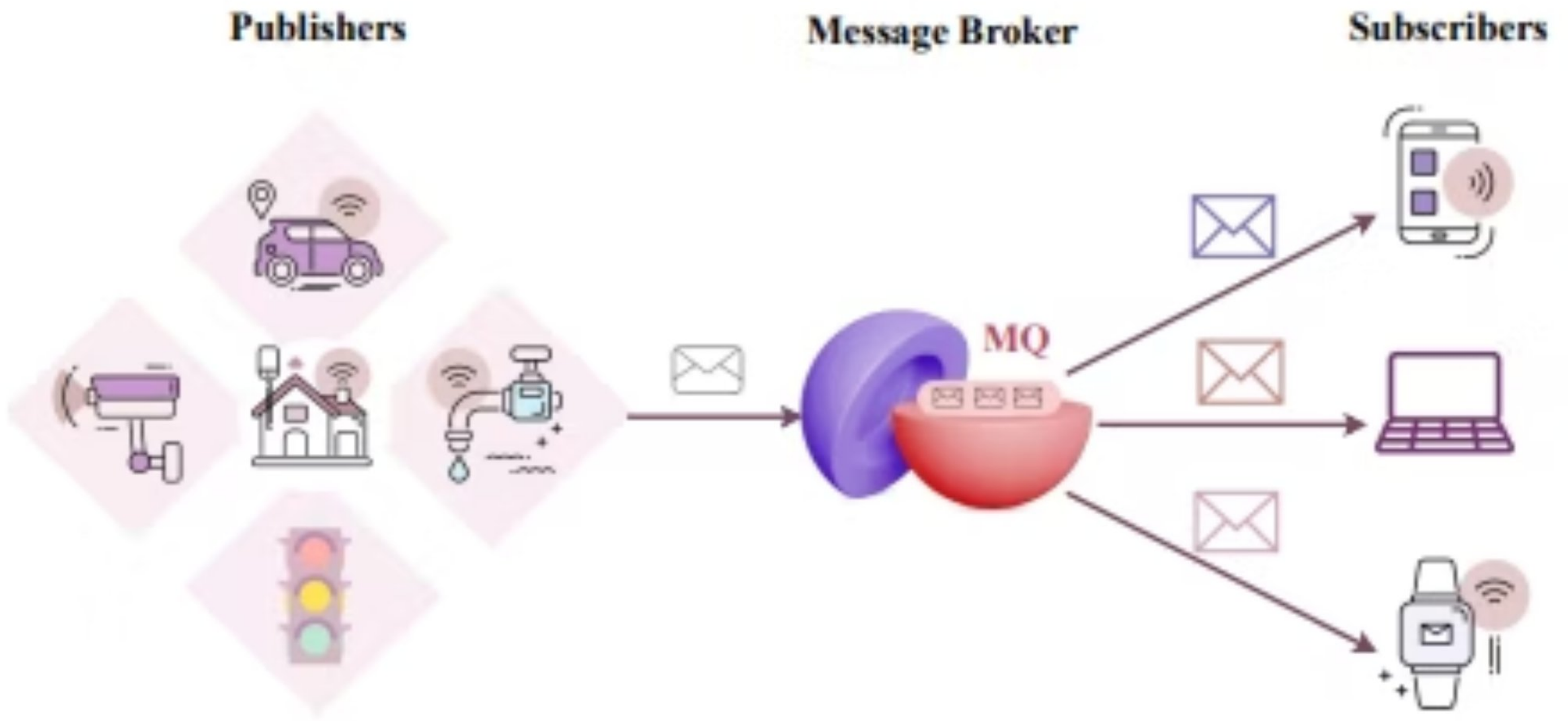


Figure 1: The Publish/Subscribe paradigm.

Publish/Subscribe Messaging Paradigm

1

Publisher Sends Messages

Publisher sends messages to the broker.

2

Broker Routes Messages

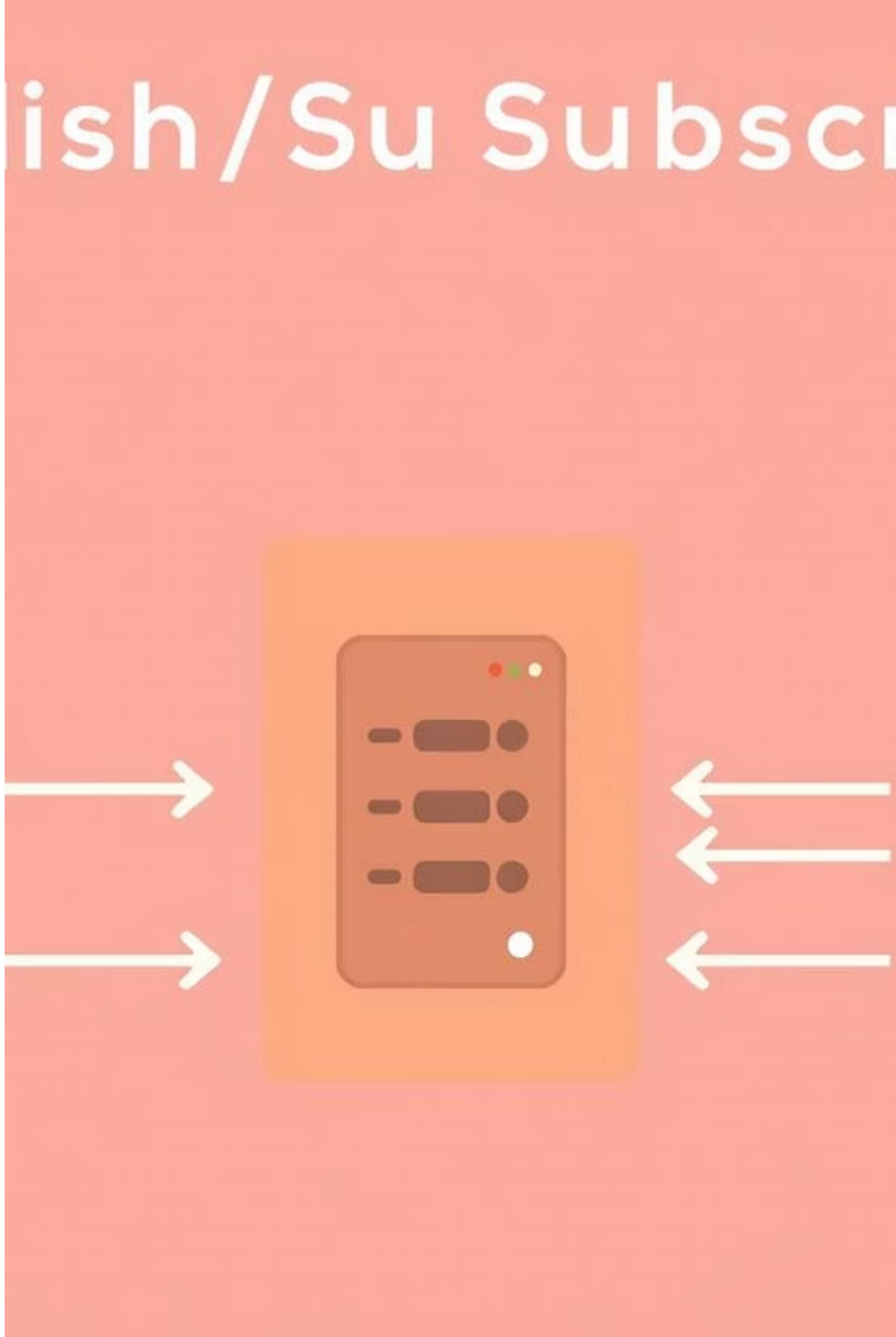
Broker receives and routes messages.

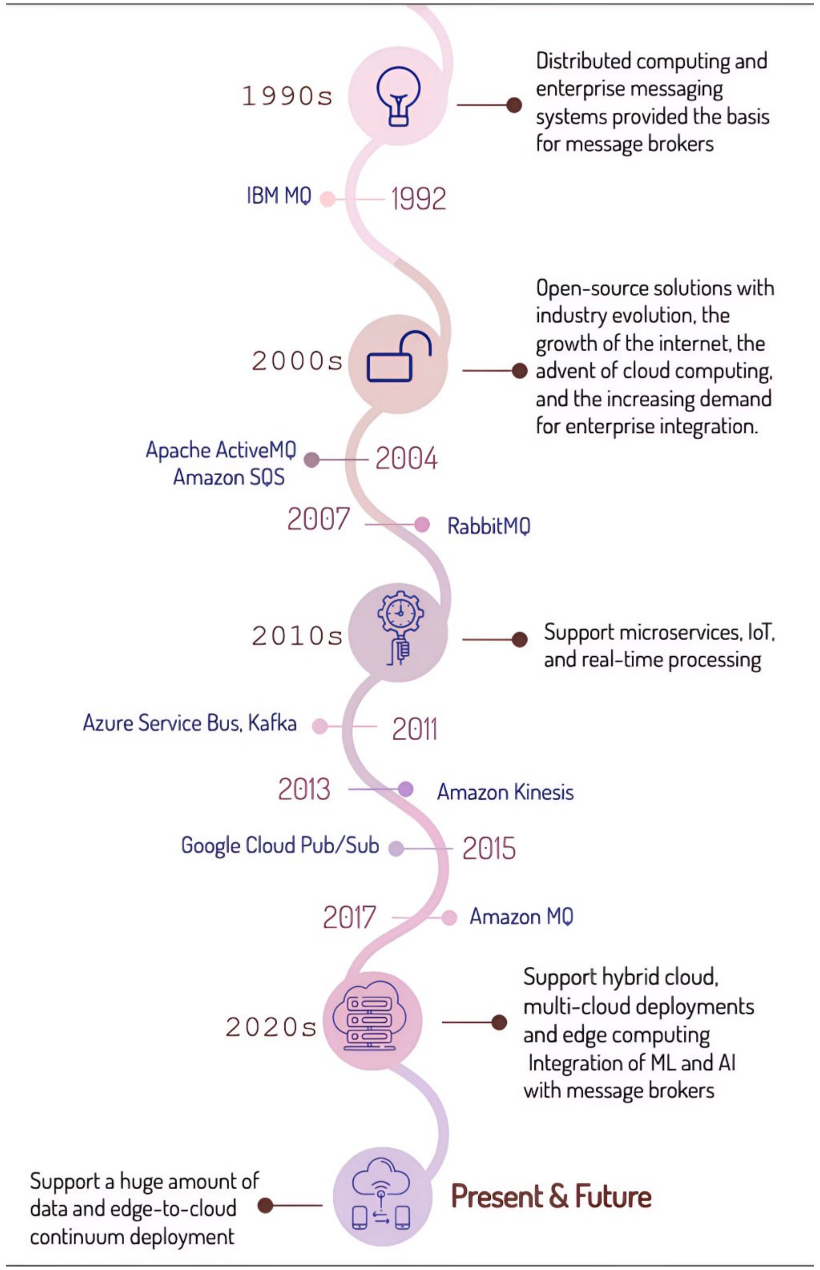
3

Subscriber Receives Messages

Subscriber receives messages from the broker.

- Allows decoupling: components evolve independently
- Broker adds: routing, storage, filtering, retries





1990s

Distributed computing and enterprise messaging systems provided the basis for message brokers

IBM MQ — 1992

2000s

Open-source solutions with industry evolution, the growth of the internet, the advent of cloud computing, and the increasing demand for enterprise integration.

Apache ActiveMQ
Amazon SQS — 2004

2007 — RabbitMQ

2010s

Support microservices, IoT, and real-time processing

Azure Service Bus, Kafka — 2011

2013 — Amazon Kinesis

Google Cloud Pub/Sub — 2015

2017 — Amazon MQ

2020s

Support hybrid cloud, multi-cloud deployments and edge computing
Integration of ML and AI with message brokers

Present & Future

Support a huge amount of data and edge-to-cloud continuum deployment

The timeline of message broker evolution from 1990 to present

Open Source Brokers - Feature Comparison

Kafka, Pulsar, Redis, HiveMQ, Celery, RabbitMQ

Feature Table: Clustering, monitoring, QoS, auth, scalability

Clustering	Yes	Yes	Yes	Yes	No	Yes
Monitoring	Yes	Yes	Yes	Yes	Yes	Yes
QoS	No	Yes	No	Yes	No	Yes
Auth	Yes	Yes	Yes	Yes	No	Yes
Scalability	High	High	Medium	High	Low	Medium

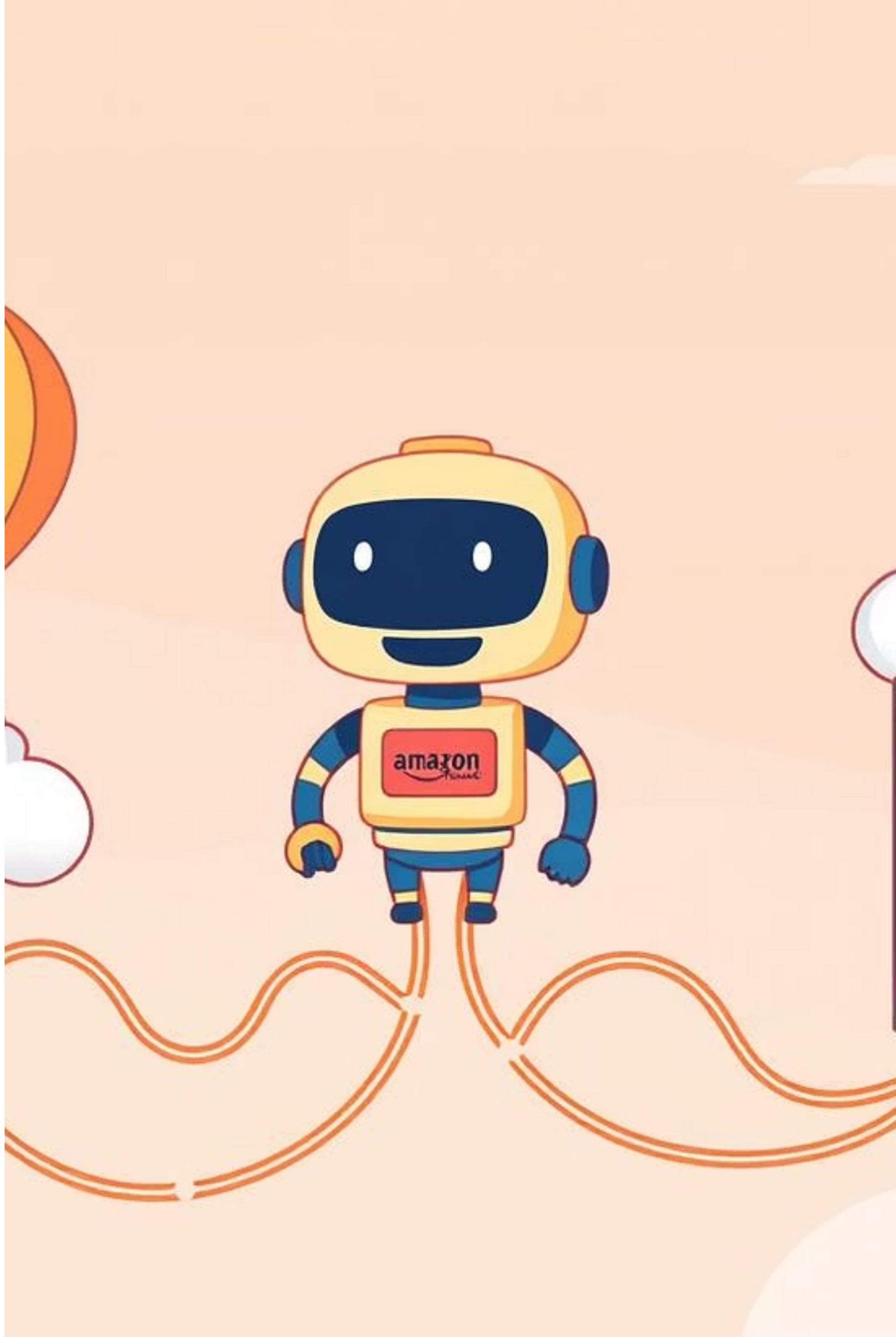
- Highlight: Kafka = high throughput, lacks priority queuing

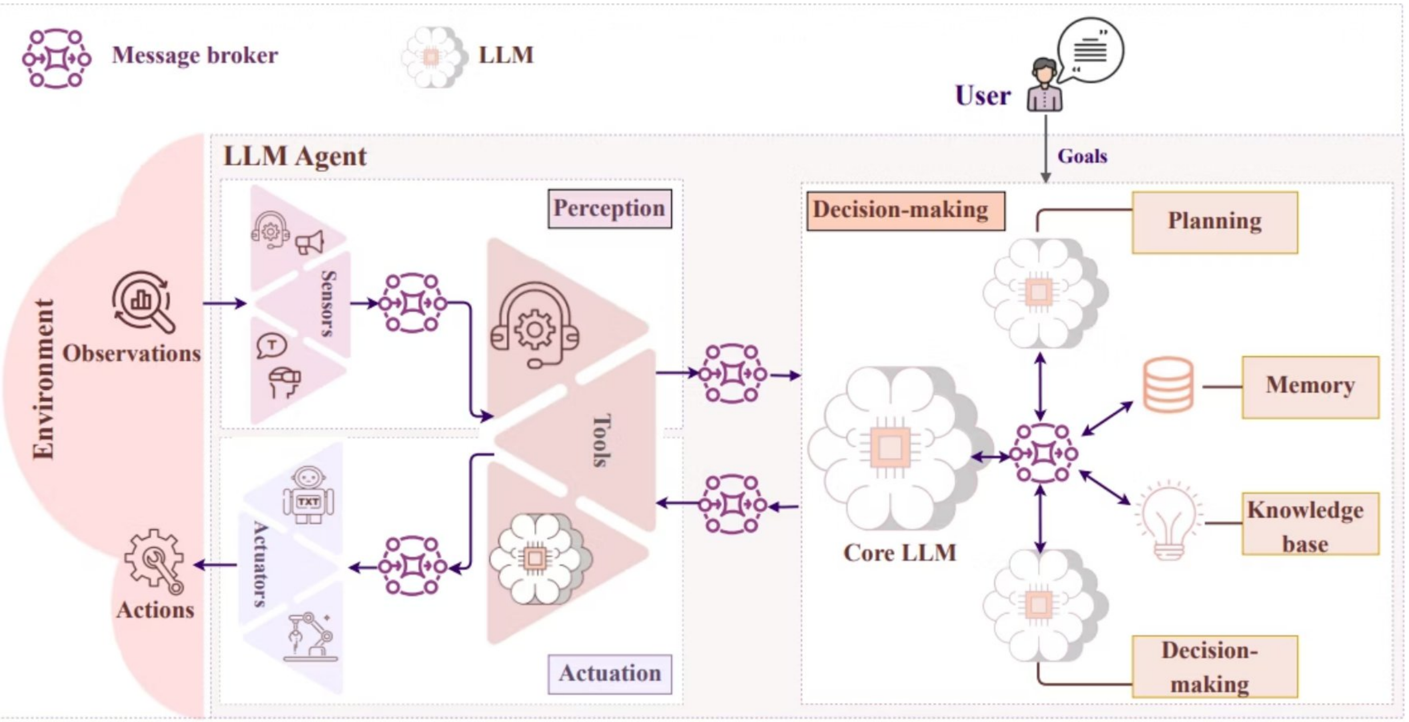
Proprietary Brokers - Feature Comparison

Google Pub/Sub, Amazon SQS, IBM MQ

Tradeoff: cloud-native scale vs lack of customizability

Example: Amazon Kinesis for real-time stream ingestion



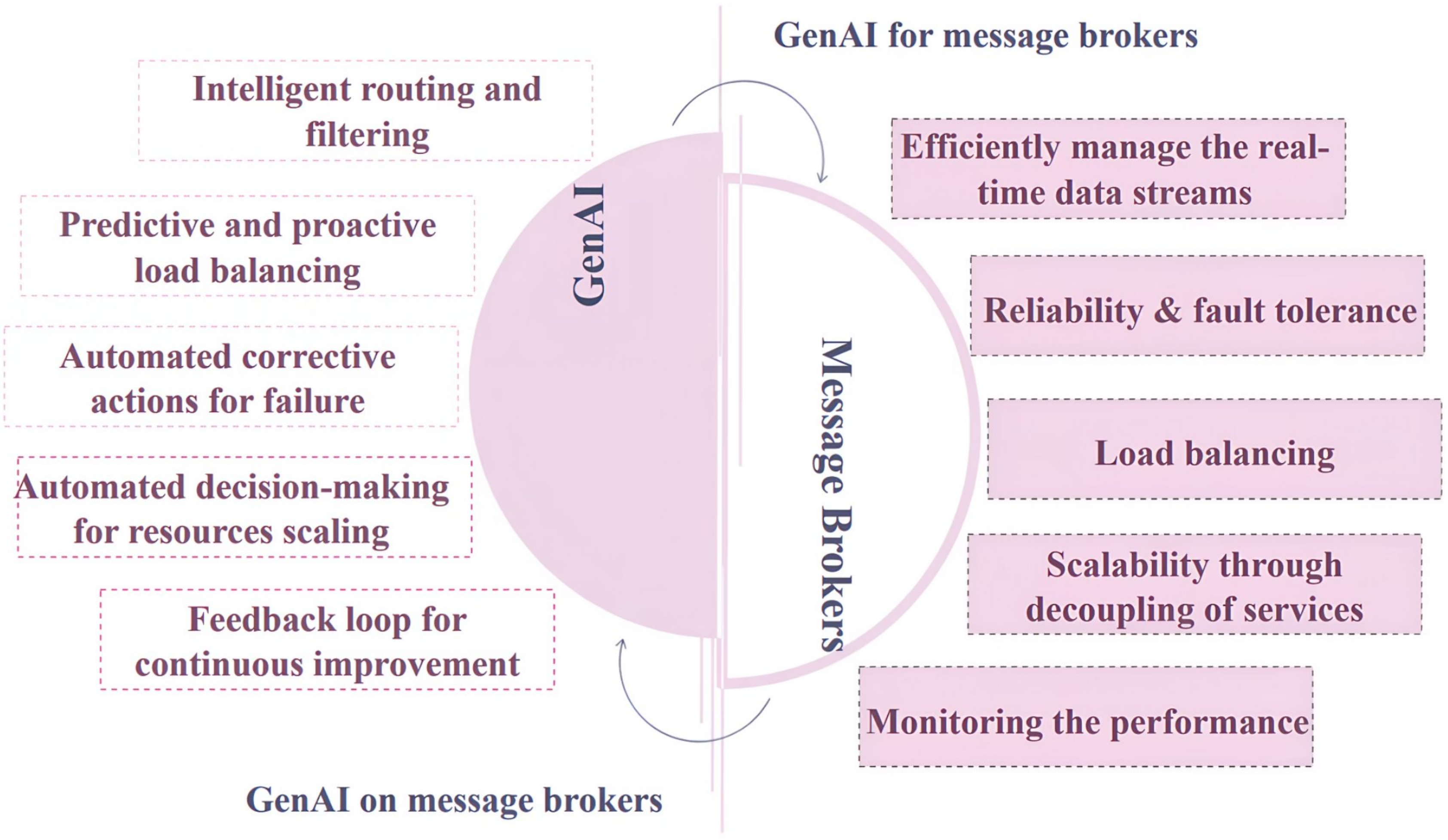


Pub/Sub + GenAI Agent Model

The overall architecture of a GenAI agent, with possible integration points with message brokers.

GenAI for Smarter Brokers

- How GenAI improves brokers:


GenAI for message brokers
Intelligent routing and filtering
Predictive and proactive load balancing
Automated corrective actions for failure
Automated decision-making for resources scaling
Feedback loop for continuous improvement
GenAI
Message Brokers
Efficiently manage the real-time data streams
Reliability & fault tolerance
Load balancing
Scalability through decoupling of services
Monitoring the performance
GenAI on message brokers

Brokers Empower GenAI

- Support massive message flows
- Enable real-time and asynchronous processing
- Distribute tasks across nodes
- Use case: Kafka feeds data to an LLM chatbot

Enhancing Brokers for GenAI

- Semantic communication (intelligent message content routing)
- Dynamic model loading + inference via brokers

🛠 Tools like Kafka Connect, Pulsar Functions

Monitoring & Security in GenAI Pipelines

- MLOps + Continuous Diagnostics (CDM)
- Real-time metrics, performance tuning
- Secure message passing (TLS, auth)

Scalability & Resource Management

- Broker support for parallelism, clustering, orchestration
- Broker + LLMs = edge/cloud resource balancing
- E.g., Celery & Kafka for distributing microtasks

The Future: Adaptive Brokers

- Need for GenAI-specific broker designs
- 5G/6G, quantum comms, real-time NLP agents
- Brokers will have embedded GenAI modules

Strengths of the Paper

- Very visual: many tables + diagrams
- Covers practical technologies (Kafka, Pulsar, Redis)
- Focused on a real need (GenAI workload management)

Limitations & Critique

- It is a survey — no experiments, models, or implementation
- Improvement ideas are conceptual
- Still, gives great foundation for innovation

Real-World Applications of Message Brokers

- Finance: fraud detection pipelines
- Healthcare: real-time monitoring and triage
- Retail: recommender systems with Kafka

Paper's Impact on Future Research

- Inspires hybrid broker + LLM orchestration
- Foundation for broker benchmarking under GenAI stress
- Calls for semantic and adaptive messaging systems

Reference

Paper: arXiv: https://arxiv.org/pdf/2312.14647v1

Thank you