

# **Sistem Terdistribusi (Distributed Systems)**

## **Messaging for Distributed Sytems**

# Introduction

- Aplikasi perangkat lunak modern sangat jarang berjalan terisolasi, umumnya mengandalkan layanan atau mengkonsumsi informasi yang disediakan oleh entitas jarak jauh lainnya.
- **Perpesanan (*Messaging*)** , selama lebih dari satu dekade, adalah solusi untuk mengatasi tantangan-tantangan dari sistem terdistribusi, seperti:
  - jaringan yang tidak dapat diandalkan
  - Kopling (*couple*) yang kuat antara produsen dan konsumen,
  - dan heterogenitas aplikasi

# Introduction

- Sistem terdistribusi modern dapat terdiri dari ratusan, atau ribuan aplikasi yang beroperasi di berbagai tingkatan dan menyediakan layanan dan fungsi yang berbeda satu sama lain.
- Arsitektur terdistribusi, ada banyak tantangan seperti jaringan tidak dapat diandalkan, kopling kuat antara produsen dan konsumen dan heterogenitas aplikasi yang ditujukan untuk membangun sistem yang solid dan andal.

# Connection-oriented communication

- Komunikasi berorientasi koneksi adalah solusi sederhana untuk bertukar informasi di antara entitas jarak jauh.
- Komunikasi berorientasi koneksi akan menjadi cara yang cepat dan mudah untuk bertukar informasi

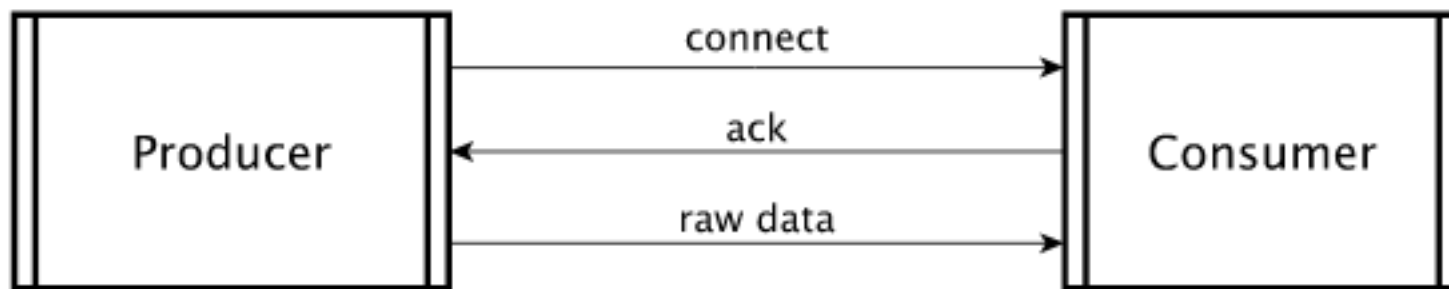


Figure 1: Tightly coupled communication.

# Connection-oriented communication

- Tetapi pada saat yang sama komunikasi itu akan didasarkan pada sejumlah asumsi yang perlu dipenuhi untuk mengatur komunikasi yang akan terjadi:
  - **Ketergantungan sementara (*Temporal dependency*):** semua komponen harus tersedia pada saat yang bersamaan.
  - **Lokasi (*Location*):** setiap komponen harus saling mengetahui alamat lainnya.
  - **Struktur dan representasi data (*Data structure and representation*):** dalam implementasinya yang paling sederhana, semua komponen harus menyetujui format dan representasi data

# Messaging for loosely coupled communication.

- Kopling dapat diukur sebagai sejumlah asumsi yang dibuat oleh pihak-pihak yang berkomunikasi tentang satu sama lain ketika mereka berkomunikasi.
- **Messaging** adalah contoh dari solusi komunikasi yang mana sambungan antara produsen dan konsumen digabungkan secara longgar di mana **message (pesan)** adalah blok pembangun informasi, yang bertujuan untuk meminimalkan asumsi-asumsi tersebut.
- Daripada mengirim informasi langsung ke alamat tertentu, pesan dapat dikirim ke saluran yang dapat dialamatkan, untuk menyelesaikan masalah ketergantungan lokasi.

# **Messaging for loosely coupled communication.**

- Untuk menghapus ketergantungan temporal, saluran tersebut dapat ditingkatkan untuk mengantri informasi hingga entitas jarak jauh siap menerimanya.
- Dengan cara ini, produsen sekarang dapat mengirim permintaan ke saluran dan melanjutkan pemrosesan tanpa harus khawatir tentang pengiriman

# Messaging scenario

- Kasus penggunaan perpesanan (messaging) yang umum adalah:
  - Penerbitan Informasi (**Information Publishing**): entitas menerbitkan informasi yang volatile (berubah-ubah) tanpa pengetahuan tentang siapa yang tertarik (mis. sensor);
  - Penyimpanan Informasi (**Information Storing**): suatu entitas mengumpulkan informasi dari berbagai sumber (mis. Pengumpul log);
  - Panggilan Prosedur Jarak Jauh (**Remote Procedure Call**): suatu entitas mengirim permintaan ke satu atau lebih entitas jarak jauh dan mengharapkan balasan.



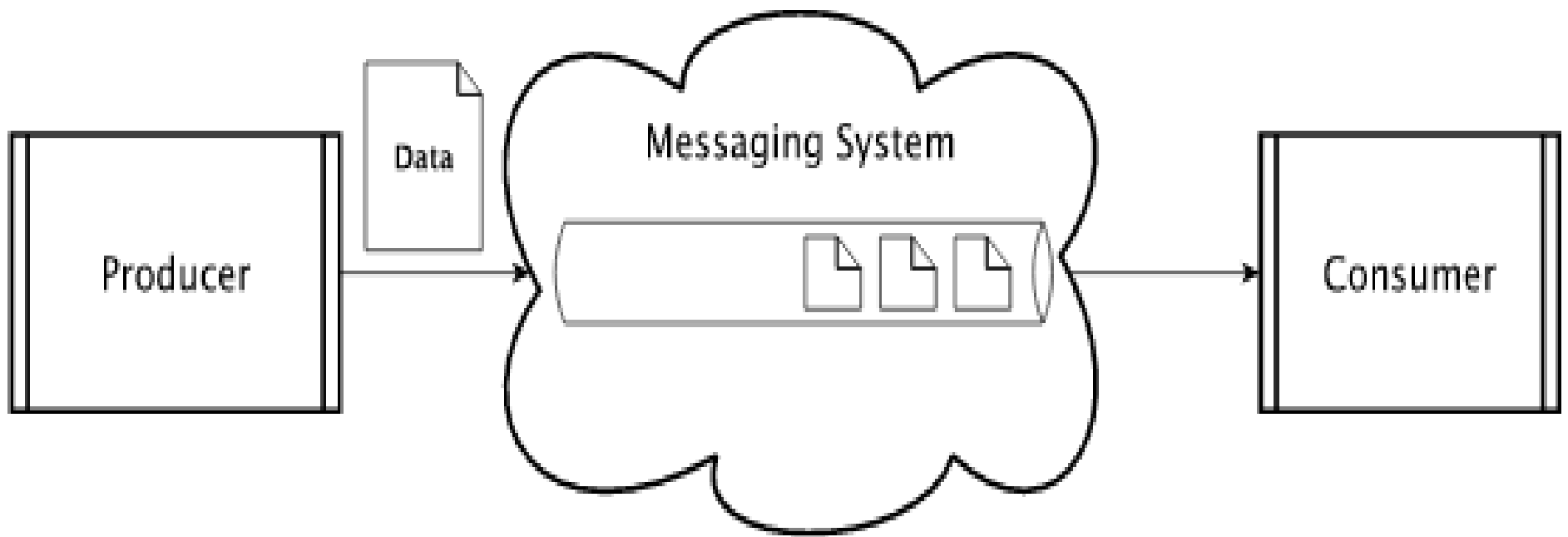
# Messaging middleware

- Perpesanan (***messaging***) adalah solusi komunikasi yang digabungkan secara longgar yang meminimalkan ketergantungan produsen dan konsumen.
- Menghapus dependensi ini membuat arsitektur keseluruhan lebih fleksibel dan toleran terhadap perubahan, tetapi disertai dengan kompleksitas tambahan.
- Oleh karena itu, **messaging middleware** khusus telah dikembangkan selama bertahun-tahun untuk menyediakan fungsionalitas pengiriman pesan tanpa harus berurusan dengan kompleksitas.

# Messaging systems

- Sistem perpesanan (Messaging systems) , seperti yang ditunjukkan pada Gambar, bertindak sebagai lapisan “tipuan” (*indirection layer*) di antara entitas yang ingin berkomunikasi. Biasanya disebut sebagai “pialang” (*broker*) pesan.
- **Message broker** bertanggung jawab untuk mentransfer data, sebagai pesan dari satu aplikasi ke aplikasi lain, sehingga produsen dan konsumen dapat fokus pada apa yang akan dibagikan daripada pada bagaimana cara membagikannya.

# Messaging systems



# Messaging systems

- **Message:** adalah blok pembangun informasi. Ini disusun oleh sebuah **body**, yang tidak dapat diubah dan berisi data terstruktur (mis. JSON, XML, protokol serialisasi) objek komunikasi dan oleh satu set **header**, biasanya pasangan nilai kunci yang dapat diproses oleh broker dan digunakan untuk routing

# Messaging systems

- **Communication models:** Sistem perpesanan mendukung model komunikasi yang berbeda, masing-masing mendefinisikan bagaimana informasi dipertukarkan antara produsen dan konsumen.
- Model komunikasi yang paling umum adalah **antrian (queue)** dan **topik (topic)**. Antrian digunakan untuk mengimplementasikan komunikasi point-to-point, di mana, jika tidak ada konsumen saat informasi diproduksi, pesan disimpan di saluran untuk pengiriman nanti, sementara jika ada banyak konsumen pesan dikirim hanya sekali.
- Topik adalah untuk skenario publish/subscribe klasik, di mana jika tidak ada konsumen, pesan akan dibuang dan jika ada banyak konsumen, sistem pesan mengirimkannya ke masing-masing pelanggan.

# Messaging systems

- **Protocols:**
  - AMQP(Advanced Message Queuing Protocol)
  - STOMP(Streaming Text Orientated Messaging Protocol)
  - MQTT(Message Queue Telemetry Transport)

# Messaging technology

- Middleware berorientasi pesan (***Message-Oriented Middleware***) adalah ekosistem layanan dan pustaka yang kaya dan solid.
- Teknologi MOM:
  - Message broker: RabbitMQ, ActiveMQ
  - Kafka (stateless broker)
  - ZeroMQ (lightweight messaging)

# Studi kasus **ZeroMQ**