

---

# Message Brokers pre distribuované systémy

Róbert Novotný  
UINF/KOPR,  
zima 2020

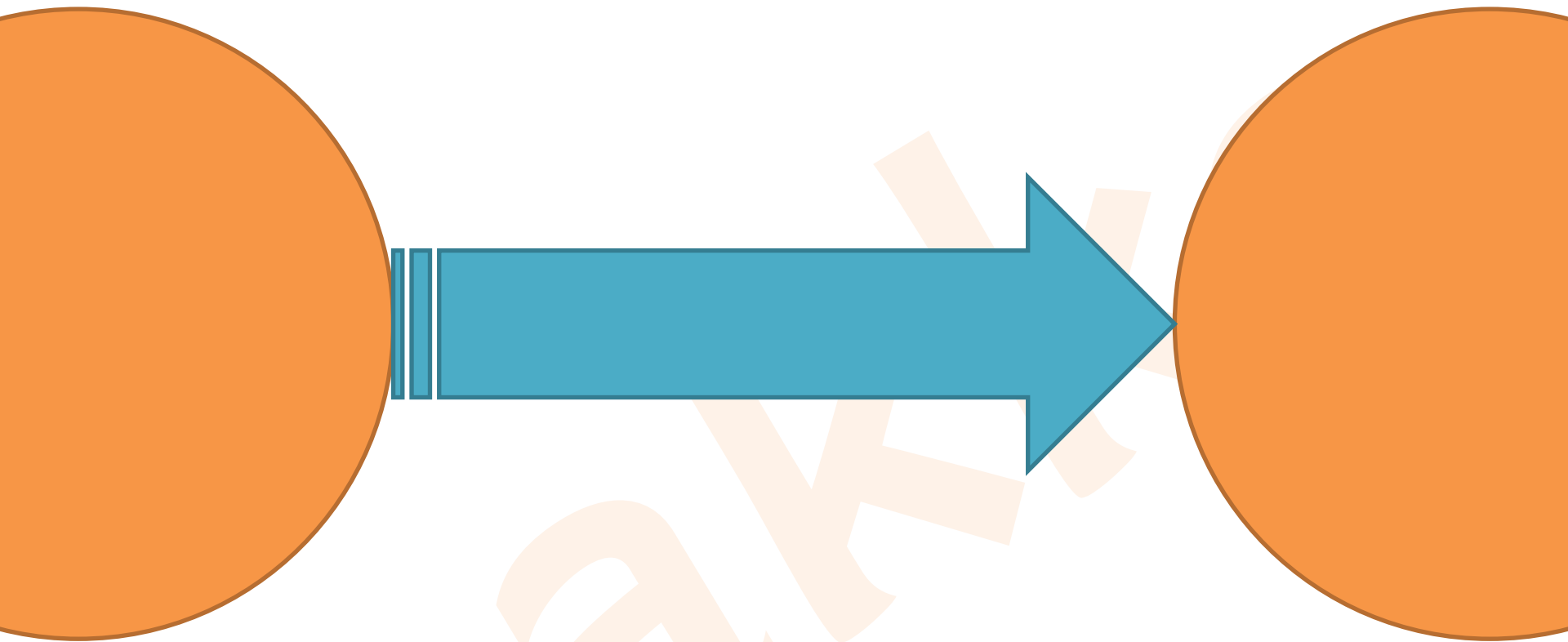
---

Ako komunikovať medzi aplikáciami?

---

„Distribúovaný systém je taký, v ktorom váš počítač prestane fungovať, pretože zlyhal iný počítač, o ktorom ste doteraz ani netušili.“

-- Leslie Lamport (1987)



# Distribúované systémy

---

- nezávislé uzly

počítače, jadrá, procesy

- komunikácia po sieti

kábel, WiFi, northbridge

# Výzvy

---

- Uzol môže zlyhať
  - vypnutie, spomalenie, preťaženie
- Sieť môže zlyhať
  - preseknutý kábel, znížený prietok

akva

# Výzvy (Tanenbaum, 2007)

---

- **architektúra**: ako navrhnúť?
- **komunikácia**: a protokoly
- **pomenovanie**: a vyhľadávanie uzlov
- **koordinácia**: zladenie práce
- **konzistencia/replikácia**: čo je pravda, kde ju nájsť?
- **tolerancia k zlyháním**: kto krachol čo s tým?
- **bezpečnosť** a prístupy

# Message Broker = Poštový úrad

---

- **smerovanie** — routing
- **topológia** — architektúra doručovania
- **transformácie správ** — medzi formátmi
- **perzistencia** — udržiavanie správ, kým ich adresát neprevezme
- **agregácia** — z viacerých zdrojov
- **synchronizácia** — čakanie na adresáta



# Populárne technológie

---

- AMQP / RabbitMQ
  - univerzálne riešenia pre klasické topológie
- Kafka
  - výkonná platforma pre streaming dát
- MQTT
  - špeciálne pre IoT
- JMS (Java Message Service / Jakarta Messaging)
  - klasická pre Java svet
- légie rozličných protokolov

# Spoločné vlastnosti

---

- uzly si posielajú **správy**
- správy sú **asynchrónne**
  - nečaká sa na odpoveď!
- odosielateľ **netuší** o adresátoch
- broker spravuje front (**queue**)
  - producent zapisuje do jedného konca
  - konzument(i) čítajú z opačného konca
- rieši (ne)**spoláhlivosť** doručenia
- **perzistencia** nedoručených správ

# Architektúry: point to point

---

- viacero producentov zapisuje do **queue**
- správu vie zjesť presne jeden konzument
  - môže byť ich viac, vyhráva prvý
  - zjedená správa zmizne z **queue**
- konzument môže **acknúť** správu
  - potvrdená správa = mizne z queue
- producent a konzument nemusia byť pripojení naraz

# Architektúry: publish-subscribe

---

nástenka

- producenti vytvoria **topic** a píšú doňho
- konzumenti sa prihlásia na odber správ z topicu
- každý konzument dostane kópiu doručenej správy
  - správa mizne
- variant: **durable** konzument, ktorý sa reštartne, nezmešká správy

# Architektúry: komplexné

---

- **queue** je základ každého brokera
- brokeri poskytujú vlastné varianty
  - **RabbitMQ**: pouličné schránky -> úrady -> bytové schránky
  - **Kafka**: topic -> partícia -> črieda konzumentov

# Formáty správ

---

- binárny protokol?
  - vlastný / Avro / Protobuf / Java serializácia
- textový protokol
  - XML / JSON / STOMP

akvka

# Spol'ahlivost' doručenia

---

- at most once
  - správy sa môžu strácať
- at least once
  - klient musí potvrdzovať spracované správy
  - ak klient váha, opakujme doručenie, kým nepotvrdí
  - kľúčové slovo: idempotencia klienta
- exactly once
  - príliš ťažké ;-)

# Spoločnosť doručenia

---

spoločnosť je niečo za niečo!