

Asynchronní integrace aplikací

RabbitMQ

Tomáš Sedláček

Obsah

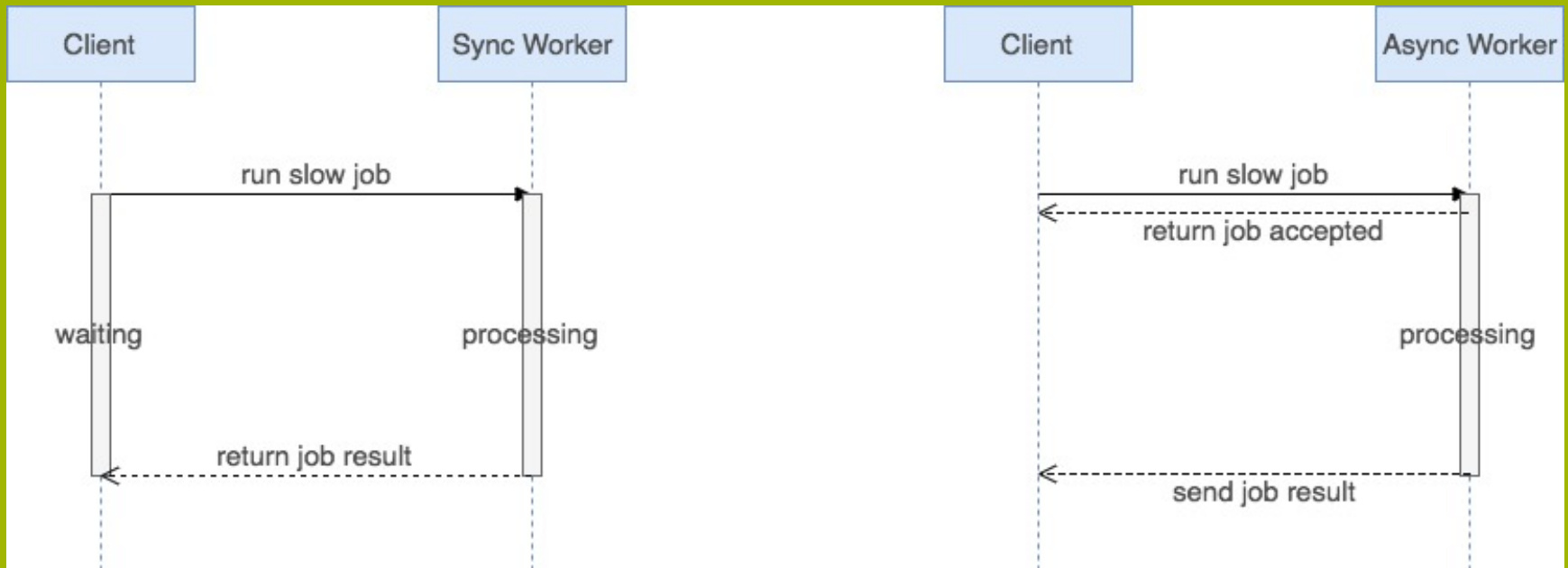
- Asynchronnost
- RabbitMQ principy
- RabbitMQ tipy a doporučení
- Demo publisher-consumer

<https://github.com/kedlas/>





Synchronní vs. Asynchronní

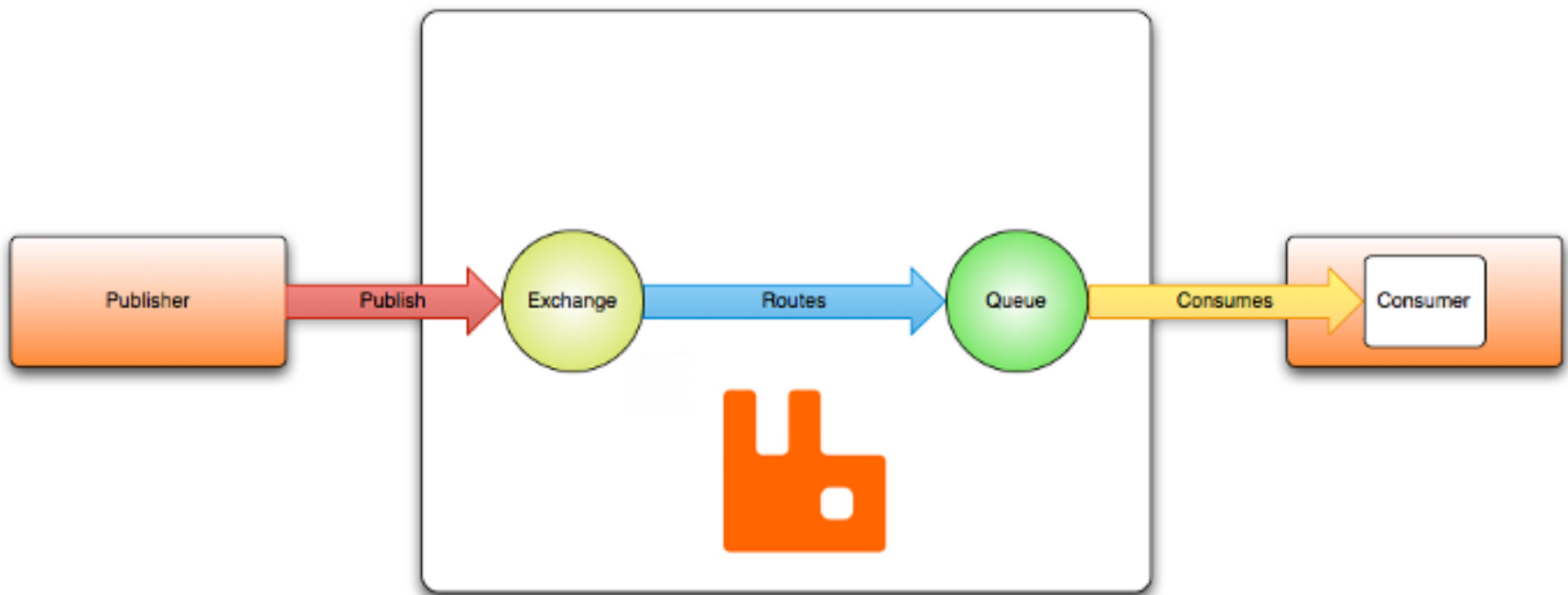


Na co se asynchronnost hodí?

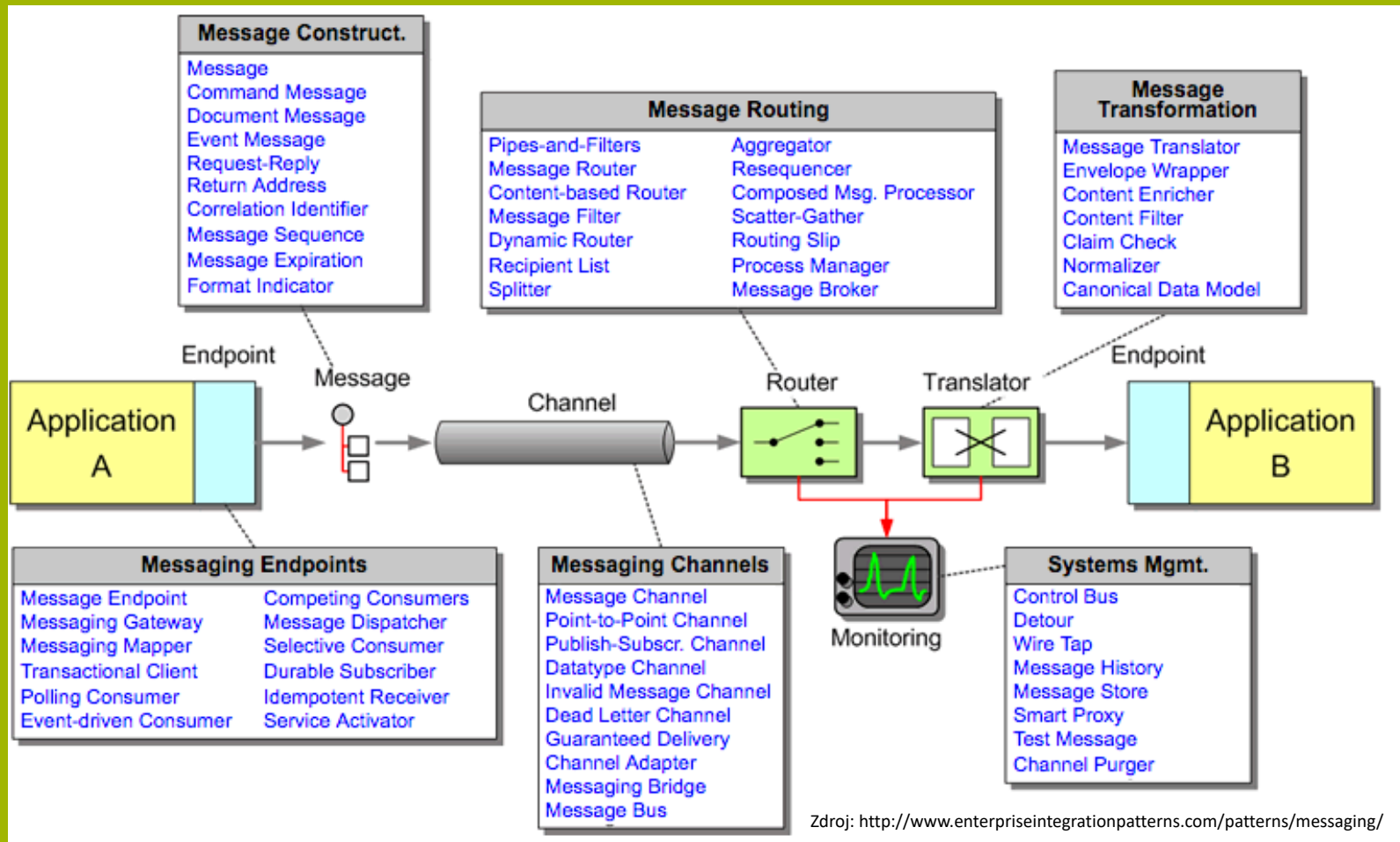
- dlouhotrvající procesy
- paralelní procesy
- non-business-core procesy (emaily, notifikace)

- **AMQP** (Advanced Message Queueing Protocol)
 - protokol určený ke komunikaci mezi systémy
- **AMQ broker** – systém uchovávající zprávy a zaručující jejich doručení od odesílatele k adresátům
 - RabbitMQ
 - Apache ActiveMQ
 - Kafka
 - ...

Message flow



Enterprise Integration Patterns



RabbitMQ spojení

- **Connection** - TCP/IP spojení. Je drahé. Ve většině případů by jedné instanci aplikace mělo stačit jedno otevřené spojení.
- **Channel** - tenká vrstva nad connection určená k multiplexování spojení. Vytváření channelů je a jejich údržba je levná. Pro každý proces/vlákno je vhodné mít otevřený samostatný channel.
- **Heartbeat** - datové pakety k ověření funkčnosti spojení. Hodnota v sekundách. Pozor na falešné výsledky při malých hodnotách heartbeat, které mohou být způsobeny např. jen zdržením paketu při cestě po síti. Při nastavení 60s odešle každých 30s datový paket, pokud neobdrží 2 za sebou jdoucí odpovědi, považuje spojení za nefunkční.

Vlastnosti zpráv

content-type	MIME content type.
content-encoding	MIME content encoding.
delivery-mode	Non-persistent (1) or persistent (2).
priority	Message priority, 0 to 9.
correlation-id	Application correlation identifier.
reply-to	Address to reply to.
expiration	Message expiration specification.
message-id	Application message identifier.
timestamp	Message timestamp.
type	Message type name.
user-id	Creating user id.
app-id	Creating application id.
headers	Custom headers fields table
body	payload

Pošli zprávu a zapomeň

- Publisher se po úspěšném odeslání zprávy do brokera už o nic nestará.
- Za doručení zprávy je zodpovědný broker.
- Broker může vrátit chybový stav při přijímání zprávy, např. kvůli plnému write bufferu.

Potvrzování zpráv

- **Acknowledgment**
 - consumer přebírá zodpovědnost za přijatou zprávu.
 - Ack by měl proběhnout až po dokončení zpracování zprávy, aby se při pádu consumera o zprávu nepřišlo.
 - zprávy lze konzumovat s flagem **noack**. V tomto případě ack proběhne automaticky v momentě kdy consumer zprávu přijme.
- **Nack**
 - negativní ack.
- **Reject**
 - Consumer vrátí aktuální zprávu zpět brokerovi => Requeuing.

Fronta umrlych zpráv

Dead Letter Queue

- Do DLQ spadnou všechny zprávy u kterých:
 - proběhl **reject/nack** a requeueing je vypnutý
 - vypršelo jejich TTL
 - přetekla maximální **délka fronty**
- Zprávy v DLQ obsahují dodatečné informace:
 - reason (rejected, expired, maxlen)
 - exchange (na jakou exchange byla zpráva poslána)
 - queue (název poslední fronty ve které zpráva byla než spadla do DLQ)
 - Time
 - routing keys
 - ...

Alternativní exchange

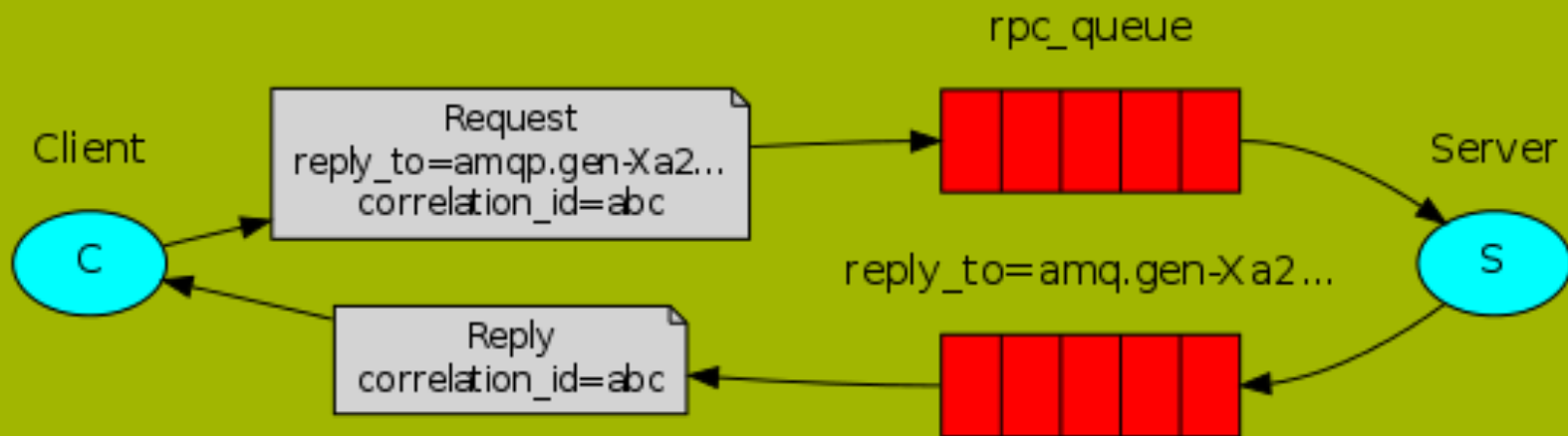
Hodí se v případech chybných/chybějících bindů.

Pokud exchange nemá nabindovánu žádnou frontu pro konkrétní zprávu, přepoše broker tuto zprávu na alternativní exchange.

RPC pattern

Klient i server jsou současně publisherem i consumerem.

Součástí zprávy od klienta jsou informace o tom kam poslat odpověď.



API

- <http://localhost:15672/api/>
- Umožňuje získávat informace o stavu běžící rabbitMQ instance pomocí jednoduchých dotazů
- Lze použít např. na:
 - monitoring délky front,
 - hlídání počtu otevřených spojení, channelů
 - atd.



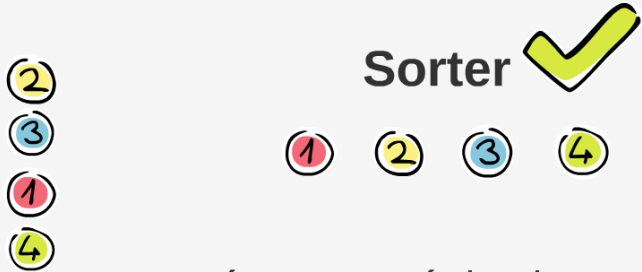
Předčítání

- Parametr prefetch udává, kolik může mít consumer unacknutých/rozpracovaných zpráv.
- Prefetch se nastavuje na channel.
 - `channel.setPrefetch($val, $global = true);`
 - každý consumer by měl pracovat nad vlastním kanálem.
- Pozor na **zamíchání pořadí** zpráv při **prefetch > 1**

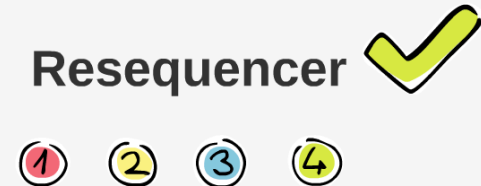
Předbíhání zpráv

- Doba zpracování jednotlivých zpráv se může lišit.
- Pokud vyžadujeme identické pořadí zpráv na výstupu jako na vstupu, musíme si sami správnost pořadí zpráv na výstupu hlídat.

Čas na demo



- zprávy *nemusí* obsahovat poř. číslo
- zprávy **musí** přijít seřazené



- zprávy **musí** obsahovat poř. číslo
- zprávy *nemusí* přijít seřazené

\$ git clone <https://github.com/kedlas/2017-rabbitmq-prednaska.git>



Dotazy?

Děkuji za pozornost

