

RabbitMQ

Tomáš Jecha (MVP)

✉ tomas@jecha.net

t jechtom

e www.jecha.net



Školení vývojářů
www.dotNETcollege.cz

Články a diskuzní fóra
www.VbNet.cz

Program

- Úvod, instalace, konfigurace
- Fronty, publikace a čtení zpráv
- Pokročilé routování zpráv
- Demo – implementace do stávající aplikace
- Pluginy
- Další postupy

ÚVOD, INSTALACE, KONFIGURACE

RabbitMQ

- Open source message broker
- Pod hlavičkou VMware
- Multiplatformní
 - Windows, Linux / Unix, Mac OS X
- Napsané v jazyce Erlang
- Podpora řady vývojářských platforem

Komunikace se serverem

- Klienti (oficiálně):
 - .NET, Java, Erlang
- Klienti (neoficiálně)
 - Ruby, Python, PHP, Perl, C/C++, Lisp, Haskell, Ocaml, Go, Amazon EC2...
- Protokoly:
 - AMQP+ (0.9.1)
 - STOMP (1.0, 1.1, 1.2)
 - Management REST HTTP API

Advanced Message Queuing Protocol

- Standardizovaný, otevřený (www.amqp.org)
 - Microsoft, VMware, Cisco, Red Hat
- Binární protokol pro bezpečný a spolehlivý přenos zpráv
- Implementován například:
 - Windows Azure Service Bus, ApacheMQ, StormMQ
- RabbitMQ – verze 0.9.1 + extenze

Instalace

- „Erlang Windows Binary File“
 - Erlang runtime (Ericsson)
- „RabbitMQ Server“ pro Windows
 - Serverová windows služba
 - Základní sada pluginů
 - Konzolové příkazy na správu

Konfigurace a správa

- Příkazy:
 - rabbitmqctl.bat
 - Hlavní konfigurace (uživatelé, zabezpečení, cluster...)
 - rabbitmq-plugins.bat
 - Povolování / zakazování pluginů
 - rabbitmq-server.bat
 - Standalone server
 - rabbitmq-service.bat
 - Správa serveru jako Windows služby
- Environment variables (RABBITMQ_BASE)
- Konfigurační soubory
- Web management plugin (<http://localhost:15672/>)

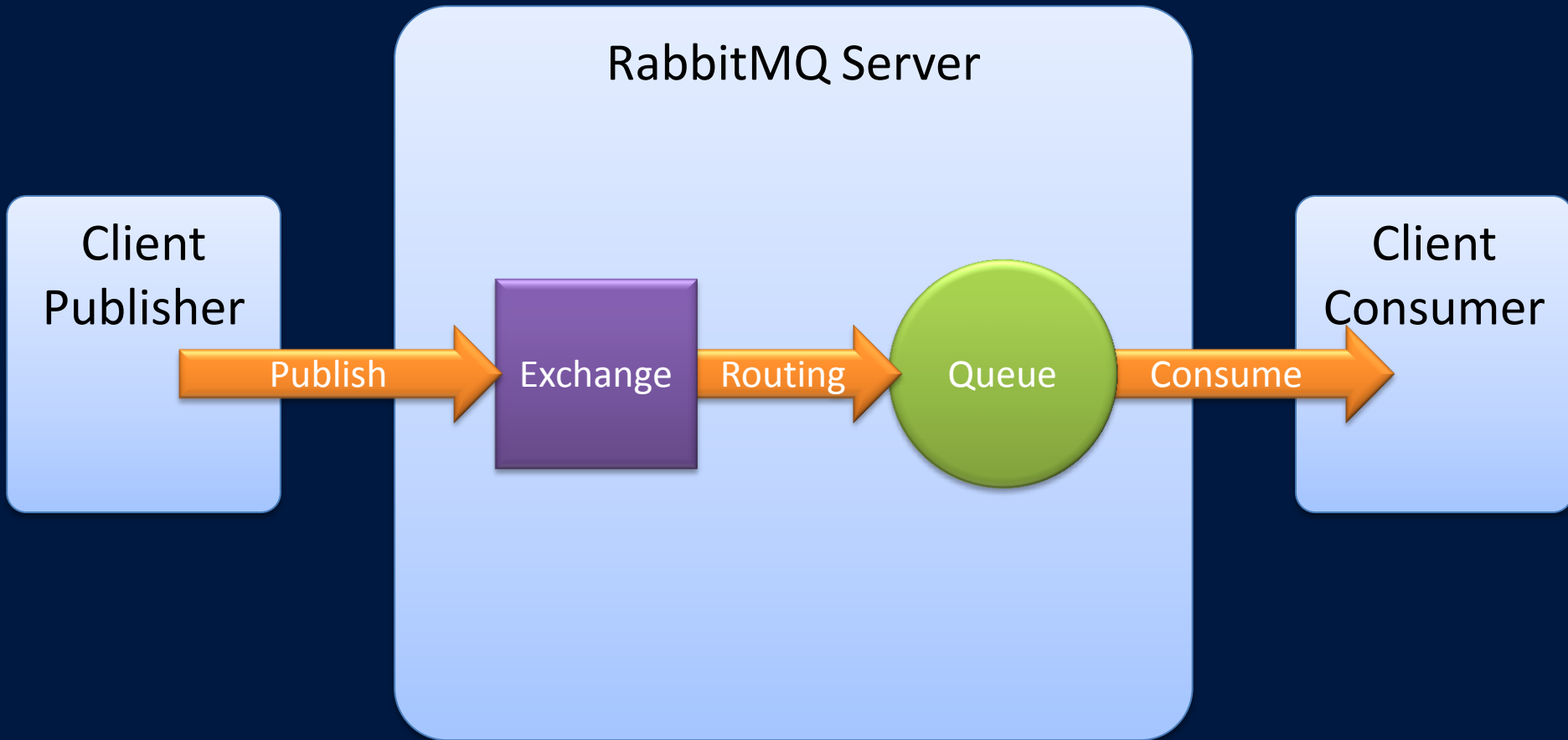
demo

Instalace a konfigurace

Web management plugin

Konfigurace permissions

Architektura RabbitMQ



Exchanges

- Fronta pro předávání zpráv
- Přijímá zprávy od:
 - Klientů (publisher)
 - Jiných exchanges
- Zpráva je vyřazena při:
 - Předání všem odpovídajícím cílům
- Vlastnosti:
 - Binding – kam zprávy předávat?
 - Type (fanout, direct, topic, headers)
 - Durable/Transient
 - Internal, Alternate Exchange
 - Custom properties

Queues

- Fronta zpráv
- Přijímá zprávy od Exchanges
- Zpráva je vyřazena při:
 - Potvrzení/odmítnutí konzumentem
 - Vypršení TTL
- Vlastnosti:
 - Durable/Transient
 - TTL, Dead Letter Exchange, Auto Delete
 - Custom properties

Zpráva

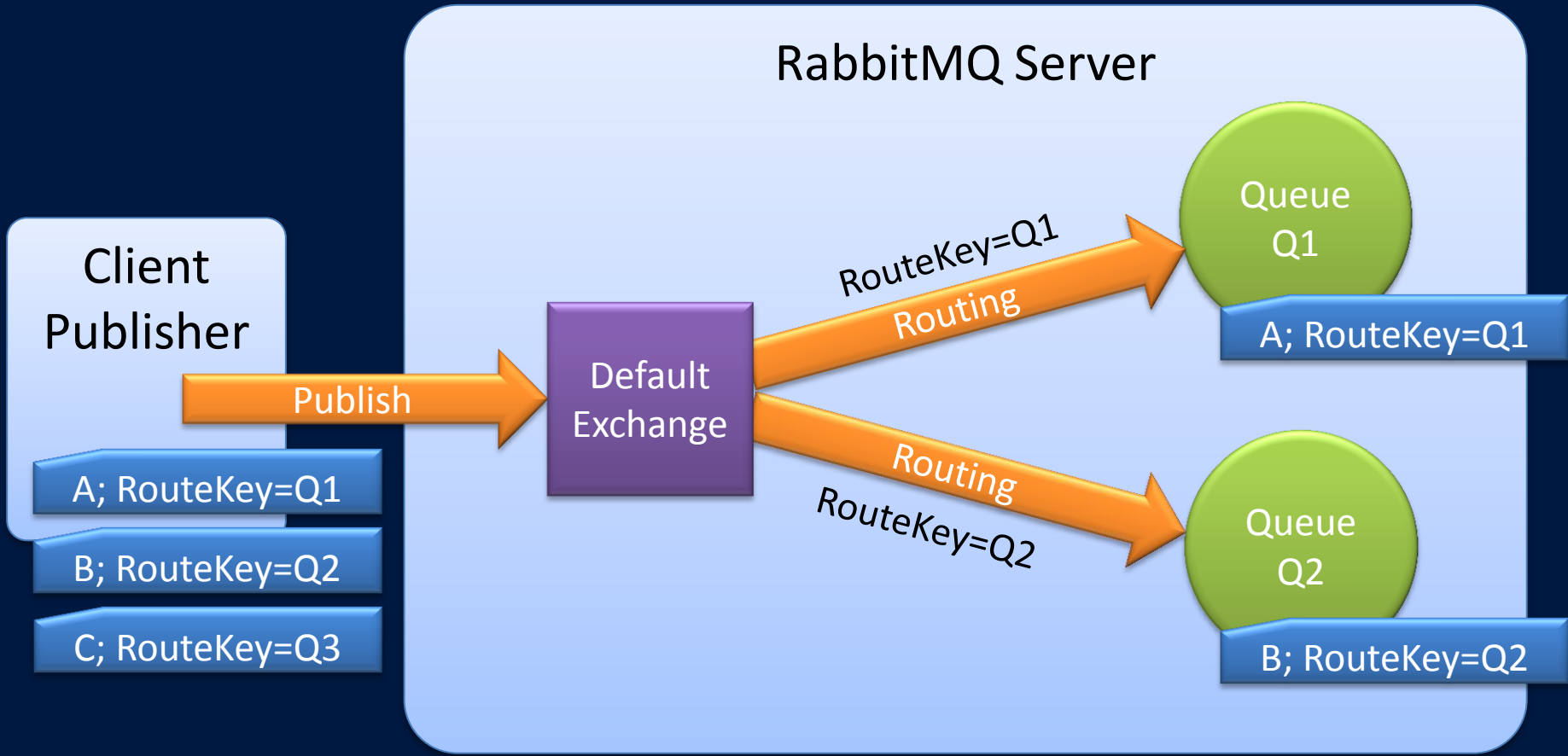
- Směrování
 - Exchange
 - RouteKey
- Properties
 - System properties
(DeliveryMode, CorrelationId, ApplicationId...)
 - Custom properties
- Obsah

FRONTY, PUBLIKACE A ČTENÍ
ZPRÁV

Defaultní Exchange

- Zaslání zpráv přímo do front
 - Automaticky vyhledá front podle **RouteKey** zprávy
- Vhodné pro jednodušší scénáře
 - Není potřeba Exchange vytvářet nebo konfigurovat, stačí založit frontu
- Jméno: „“ (empty string)

Default Exchange



demo

Nová fronta

Publikace zpráv – C#

Další vlastnosti zpráv

- Delivery mode
 - 1 = Non-persistent; 2 = Persistent
- AppId
 - Identifikátor zdroje
- CorrelationId
 - Párování zpráv (například RPC volání)
- Reply To
 - Jméno exchange pro odpověď (RPC)
- Content Encoding & Content Type
 - Informace o obsah zprávy

Konzumace zpráv

- Způsob doručení:
 - Pull – příkaz „Get“, získá 1 zprávu
 - Push – příkaz „Consume“, přihlášení k odběru
- Potvrzování zpráv
 - Bez potvrzení (noAck=true)
 - Zpráva je po doručení považována za zpracovanou
 - S potvrzením (noAck=false)
 - Zpráva je po doručení ve stavu „nepotvrzena“ a čeká se na „Ack“ / „Nack“ (Reject)
- Pokud zprávu zamítneme, lze je:
 - „Zabít“ (requeue=false)
 - Nebo vrátit do fronty (requeue=true)

demo

Konzumace zpráv

Použití čtení Pull vs Push

Potvrzování / odmítání zpráv

POKROČILÉ ROUTOVÁNÍ ZPRÁV

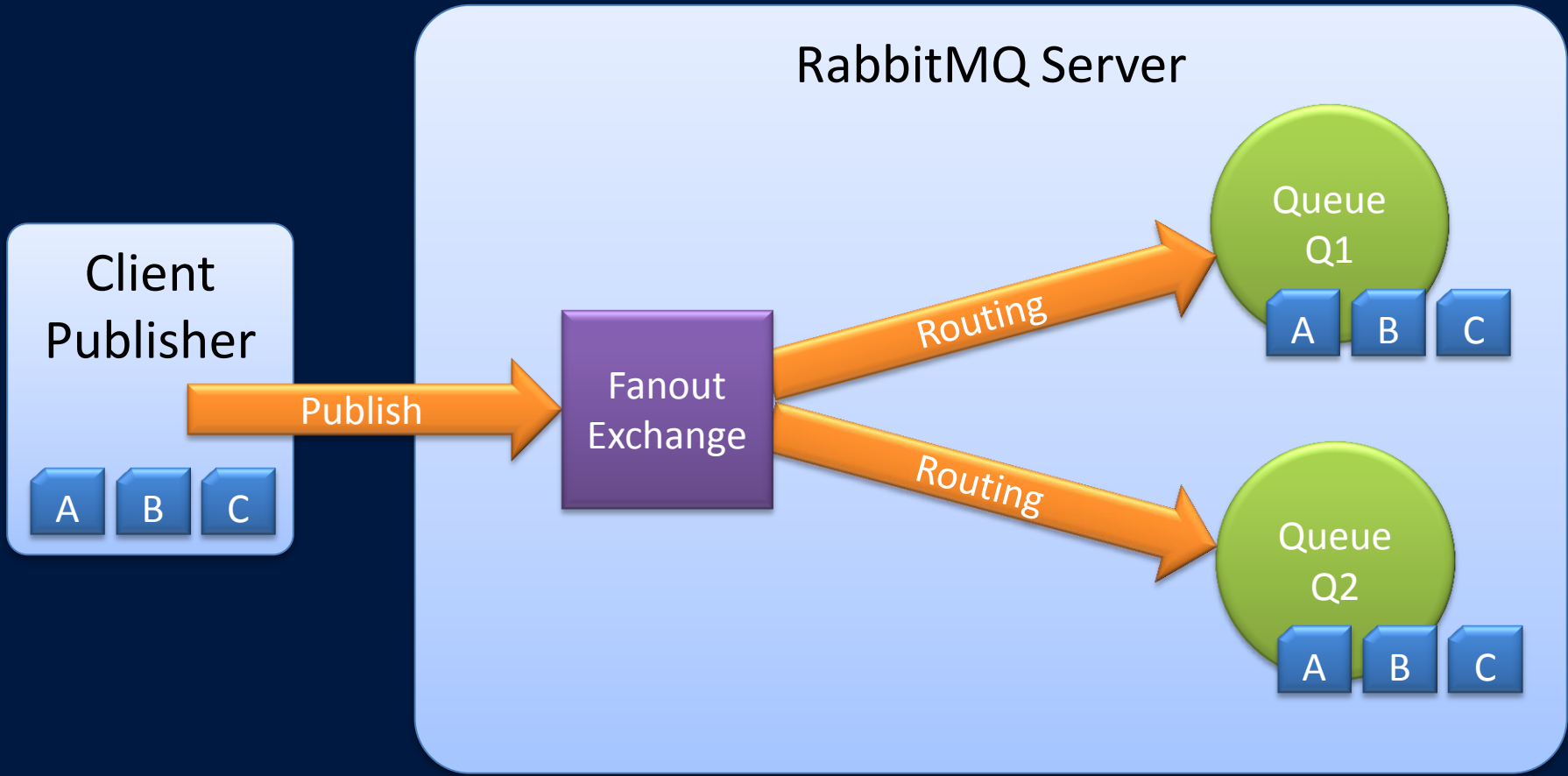
Typy Exchanges

- Fanout
- Direct
- Topics
- Headers

Typy Exchanges

- **Fanout**
 - Distribuce všem bindovaným odběratelům
- Direct
- Topics
- Headers

Fanout Exchange



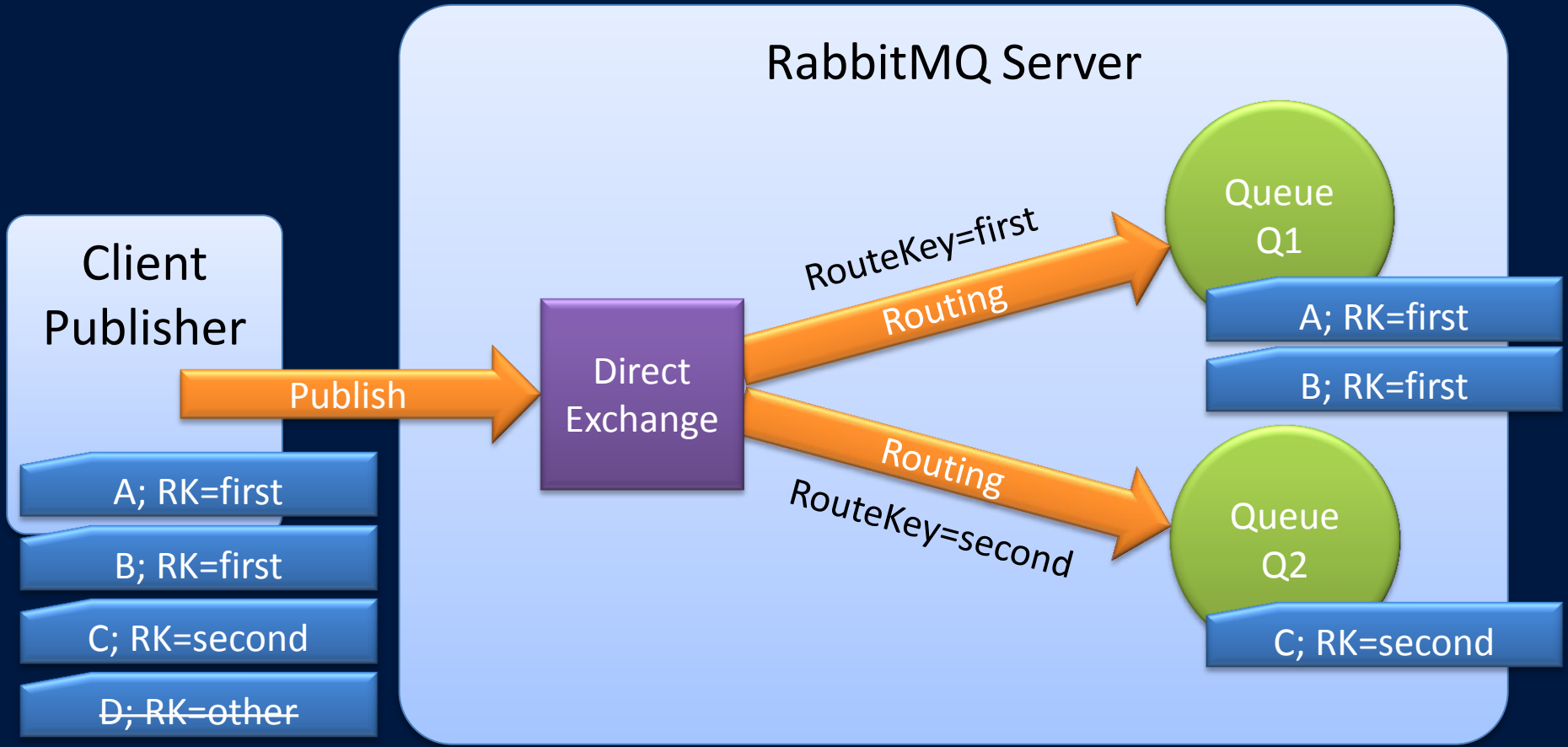


Fanout Exchange

Typy Exchanges

- Fanout
- **Direct**
 - Rozhoduje parametr zprávy **RouteKey**
 - Bude odeslána routováním, kterém má nastavený shodný **RouteKey**
- Topics
- Headers

Direct Exchange



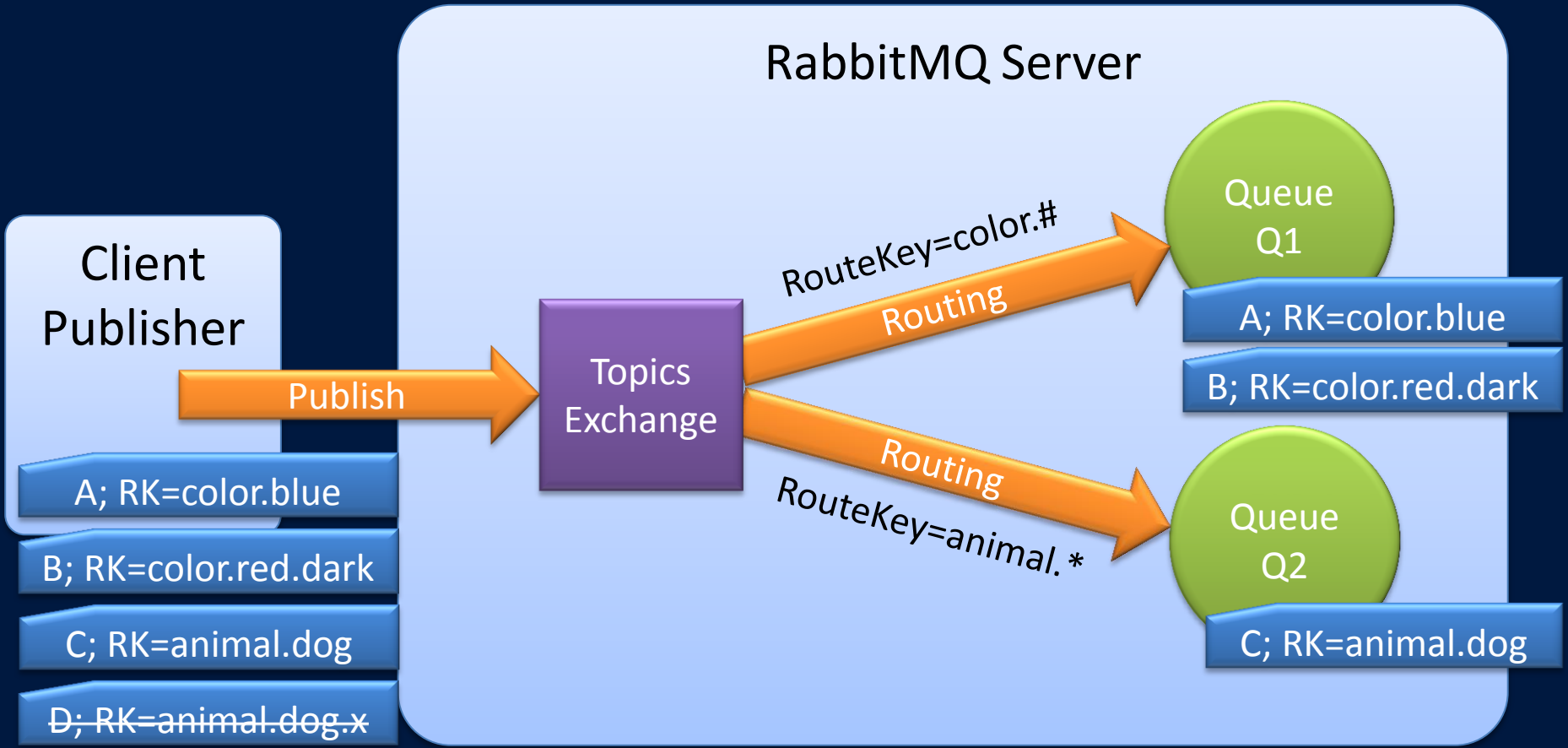


Direct Exchange

Typy Exchanges

- Fanout
- Direct
- **Topics**
 - Podobně jako u „Direct“ rozhoduje parametr zprávy **RouteKey**
 - Zástupné znaky:
 - * = libovolné jedno slovo (oddělená tečkama)
 - # = žádné nebo více libovolných slov
- Headers

Topics Exchange





Topics Exchange

Typy Exchanges

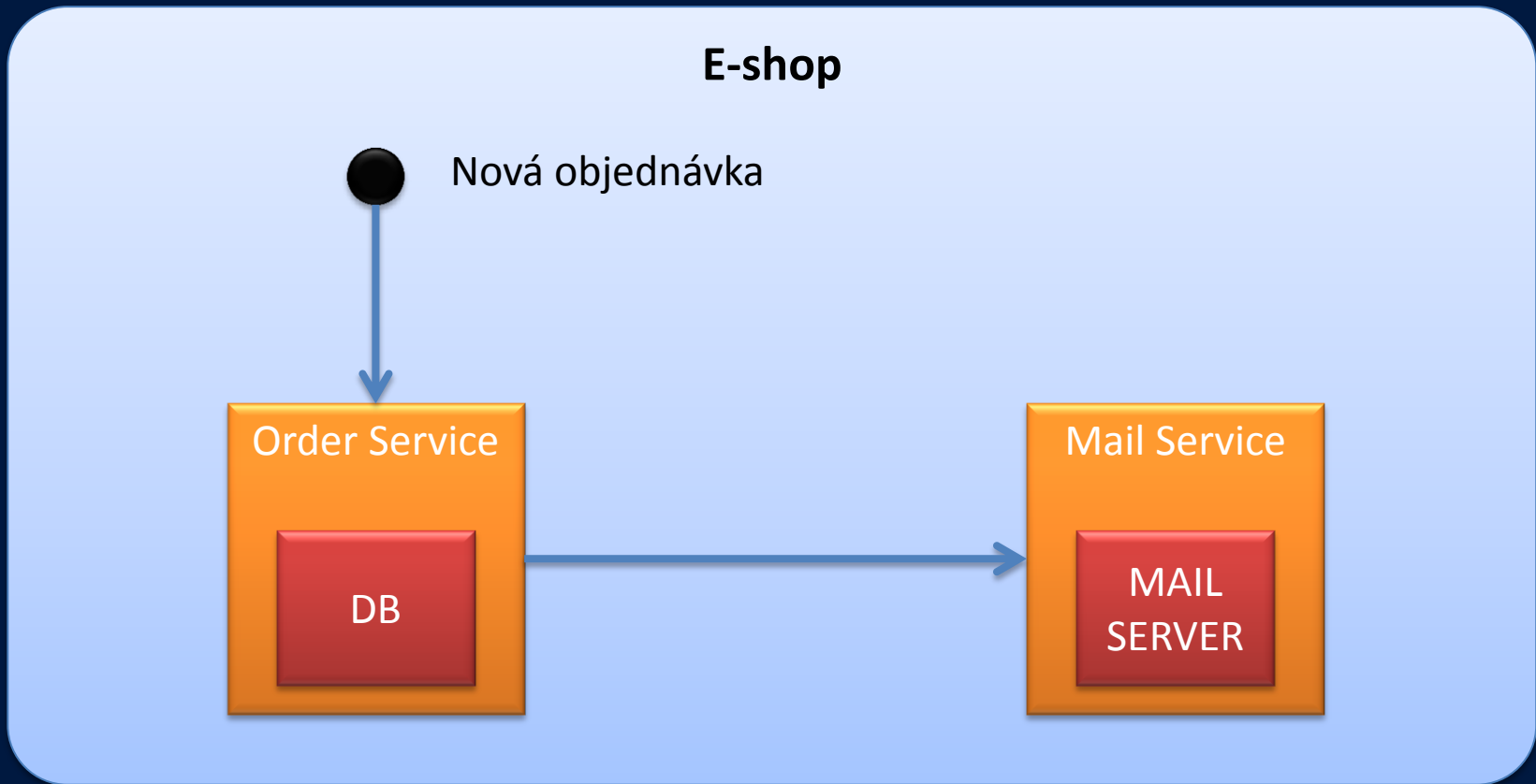
- Fanout
- Direct
- Topics
- **Headers**
 - Pokročilejší routování podle libovolných vlastností zprávy (ne jen podle **RouteKey**)

DEMO – IMPLEMENTACE DO STÁVAJÍCÍ APLIKACE

Proč používat fronty?

- Aplikace spolu nekomunikují přímo
- Technologická nezávislost
- Výpadek aplikace nemusí způsobit výpadek celé služby
- Bezpečnost
- Možnost opakovat události

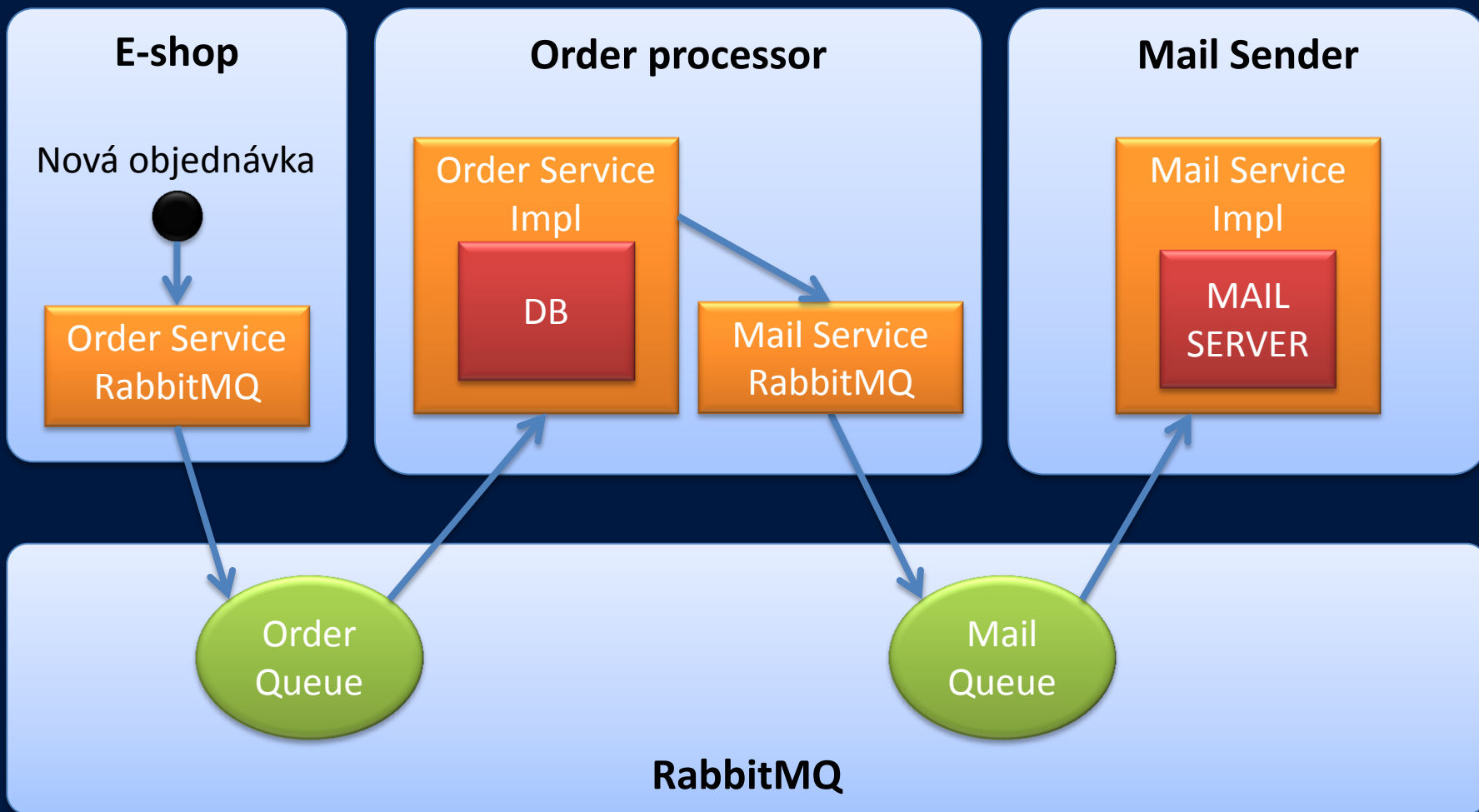
Modelová aplikace





Stávající implementace

Modelová aplikace





Implementace

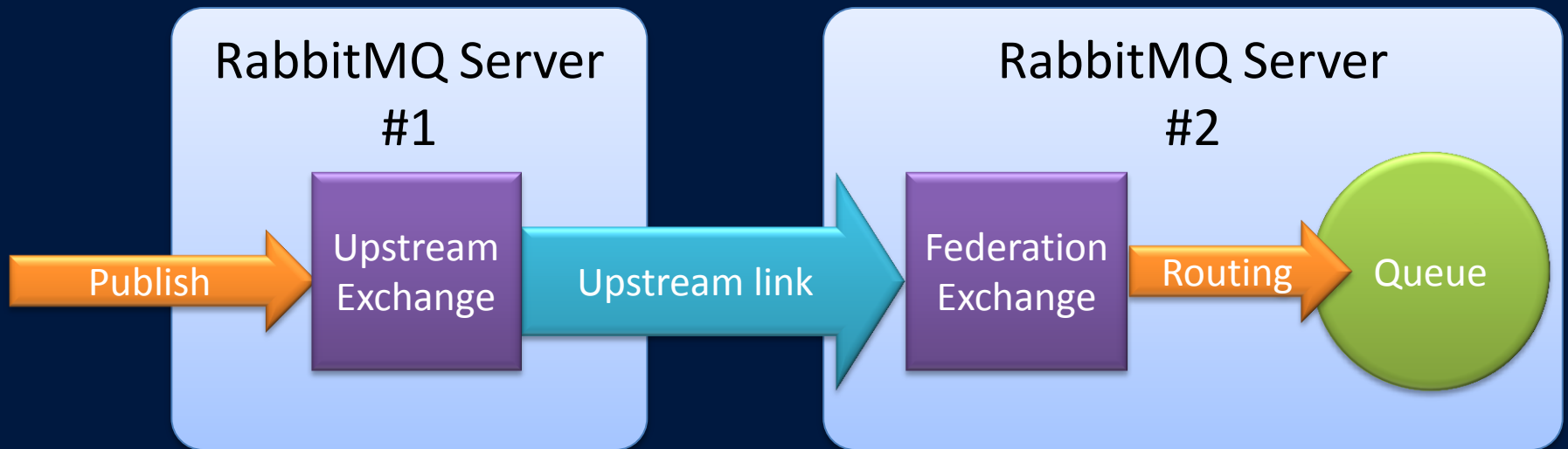
PLUGINY

Pluginy

- Serverové:
 - Management Plugin – Webový interface
 - STOMP – Simple Text Messaging Protocol
 - Autentifikace pomocí LDAP
 - SSL komunikace

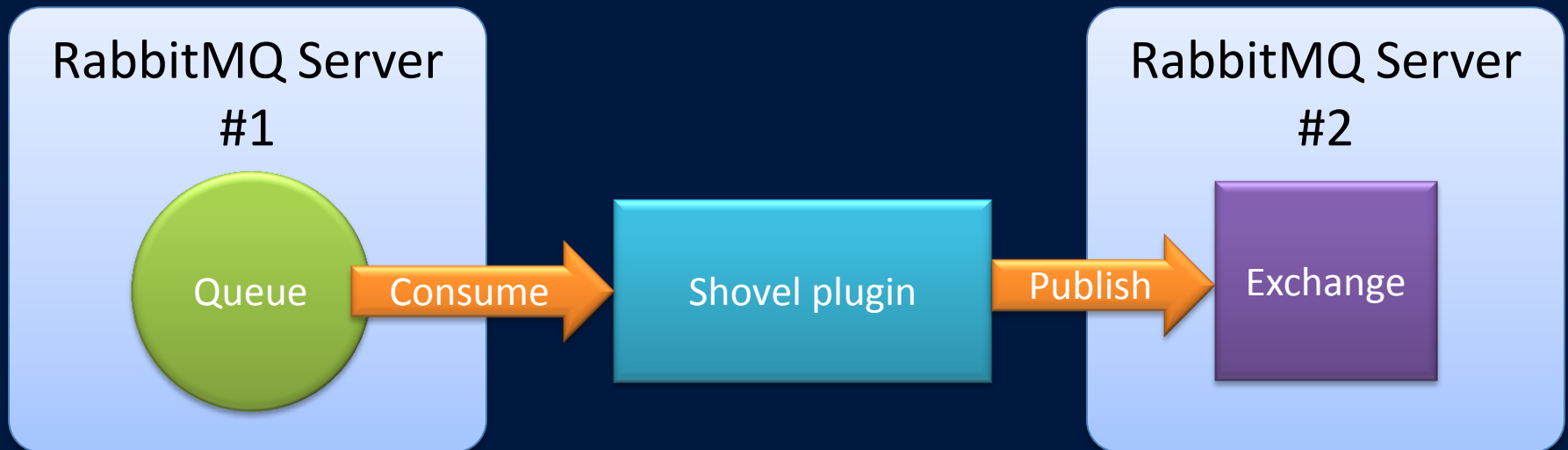
Federation Plugin

- Nový typ exchange „Federation“ konzumuje zprávy vložené do vzdálených „Upstream“ exchanges



Shovel Plugin

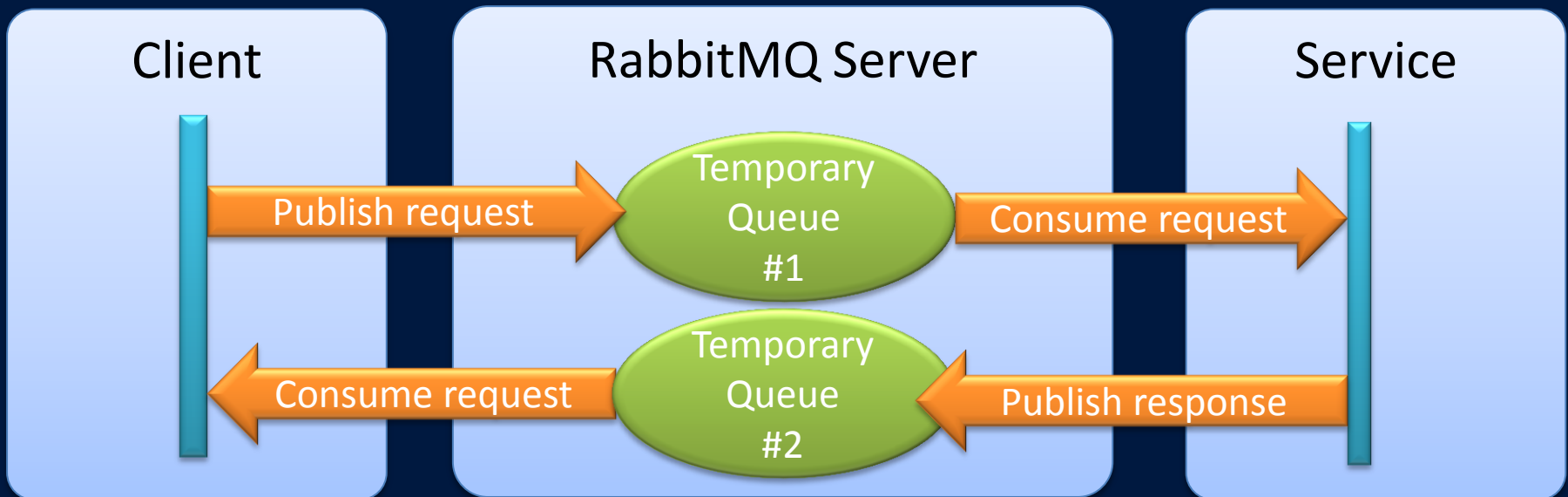
- „Přehazuje“ zprávy z jedné fronty do libovolného exchange na druhém brokeru
- Podobný jako Federation plugin, jen na nižší úrovni



DALŠÍ POSTUPY

RabbitMQ.ServiceModel

- WCF transportní vrstva pomocí front
- Lze použít pouze na synchronní volání (služba musí být v době volání online)
- Používá se **CorrelationId** a **Reply_To**



Dead Letter Exchange

- „Dead Letter“ je zpráva ve frontě, která by měla zaniknout z důvodu:
 - Expirace (TTL)
 - Nebo odmítnutí klientem (rejected)
- Lze nastavit vlastnosti:
 - x-dead-letter-exchange**
 - x-dead-letter-routing-key**
 - Událost se následně směřuje do vybrané exchange
 - Do vlastností se přidají informace



Dead Letter Exchange

Clustering

- RabbitMQ podporuje clusterování
- Konfigurace nodů je sjednocená
- Fronty v clusteru jsou mirrorované z jednoho „mastera“ na:
 - Všechny nody
 - Nebo konkrétní nody
 - Nebo určitý počet nodů
 - Nebo nejsou mirrorované vůbec
- Pouze do LAN - výpadky spojení nenesou dobře

Virtual Hosts

- Logicky rozděluje objekty v brokeru
- Každý virtual host může obsahovat vlastní:
 - Queues
 - Exchanges
 - Konfiguraci zabezpečení
- Identifikován jménem
 - Defaultně pouze 1 virtual host „/“

Kdo používá Erlang & RabbitMQ?

- Erlang:
 - Facebook chat
 - CouchDB
 - GitHub
 - Amazon SimpleDB
- RabbitMQ:
 - VMware – cloudové a virtualizační systémy
 - NASA - NEBULA Cloud Computing Platform
 - Google – Rocksteady projekt
 - Microsoft, AT&T, Mozilla, BBC, Digg...

Další zdroje

- RabbitMQ - www.rabbitmq.com
- AMQP (protocol) – www.amqp.org
- Erlang (language) – www.erlang.org
- Kurz C115 – dotNETcollege.cz

Tomáš Jecha (MVP)

✉ tomas@jecha.net

t [jechtom](https://twitter.com/jechtom)

e www.jecha.net



Školení vývojářů
www.dotNETcollege.cz

Články a diskuzní fóra
www.VbNet.cz