Service Broker

David Hlaváček
MCP, MCSE, MCSA, MCTS, MCITP, MCT
david.hlavacek@alza.cz
david@hlavacek.pro
alza.cz

Obsah

Obsah

- Service Broker
  - Komunikace, asynchronní zpracování, škálování
- Zpráva
  - Nositelem instrukce, informace a dat
- Asynchronní aplikace jsou často složitější na programování (debugging)
  - Životní cyklus zprávy od jejího odeslání probíhá bez přímé interakce uživatele
  - Řešení chyb a problémů může být problematické
  - Chyby a PRINT operace aktivace se logují do ERRORLOGu
- **Možnosti využití ve Vašem řešení jsou limitovány pouze vaší fantazií**

alza.cz

Synchronní a asynchronní zpracování

Synchronní zpracování

- Zcela běžně používaný typ plnění zadaných úkolů v programových objektech SQL Serveru
  - Uložená procedura obsahuje několik T-SQL příkazů, kdy se každý další vykoná po dokončení předchozích
  - Job ve službě SQL Agent obsahuje několik kroků, které se vykonají ve specifikované posloupnosti

Asynchronní zpracování a komunikace

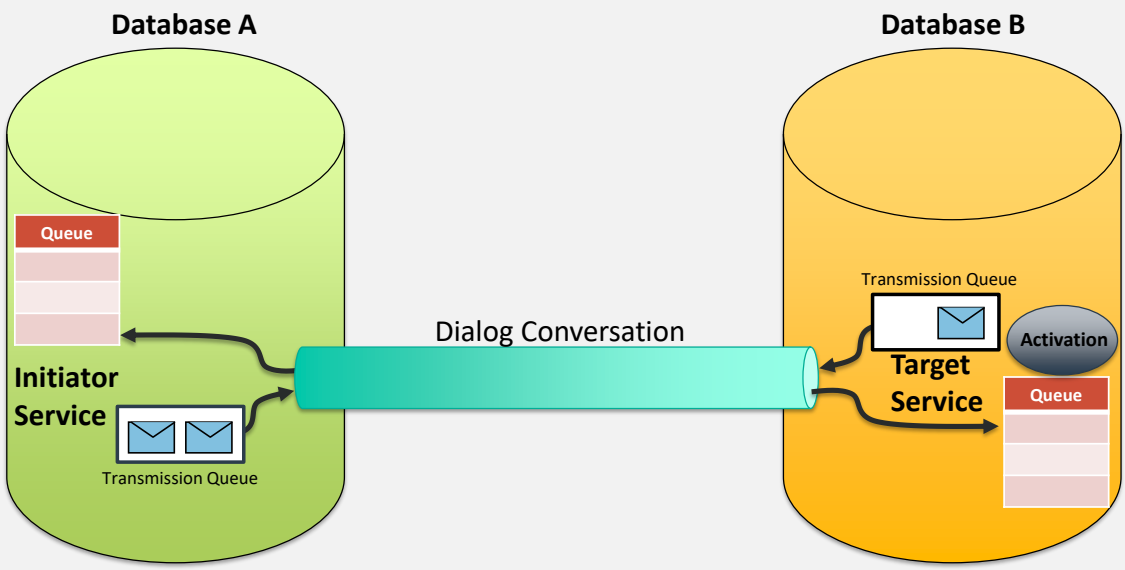
- Řešení postavené na zasílání zpráv
- Odesílatel zprávy může pokračovat v práci zatímco se čeká na příjemce, až zprávu zpracuje a případně na ni odpoví
  - Přímá analogie s odesíláním emailů
- Iniciátor komunikace pouze umístí zprávu do fronty a další její zpracování probíhá bez jeho přímé interakce
- Efektivní škálování

alza.cz

Service Broker

- Interní služba SQL Serveru pro asynchronní zasílání zpráv
  - (VELMI zjednodušeně lze přirovnat k funkci mailu)
- Asynchronní zasílání a zpracování zpráv
  - Odesílatel pokračuje v práci zatímco zpráva může být na pozadí doručena a zpracována příjemcem (eventuálně na ni příjemce odpovídá)
- Minimální GUI podpora
- Specifické T-SQL příkazy a objekty
  - SEND, RECIEVE,BEGIN DIALOG…
  - QUEUE, SERVICE, MESSAGE TYPE, CONTRACT ,ROUTE…
- Dostupný ve všech edicích SQL Serveru
  - Express edice nemohou komunikovat napřímo

alza.cz

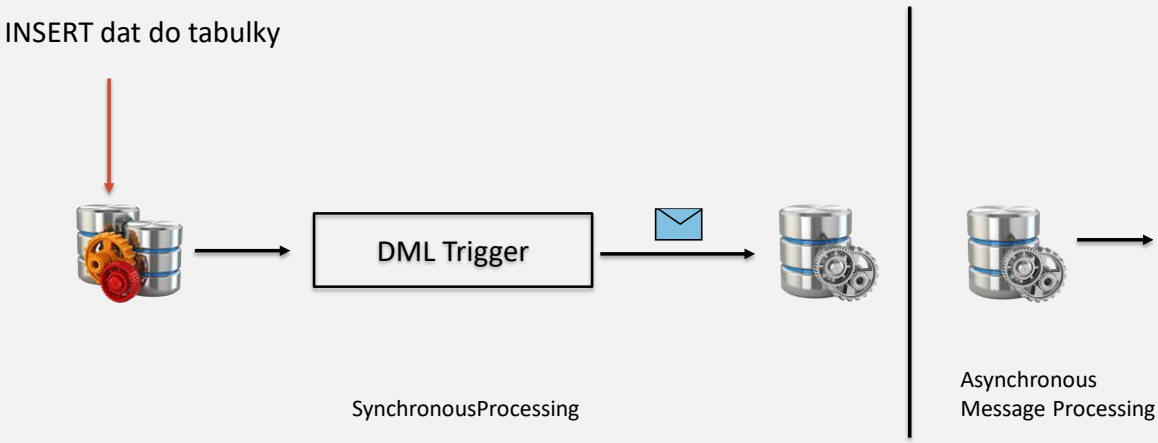


Service Broker

- Příklady reálného využití
  - SQL Server maintenance
  - Asynchronní triggery
  - Publish – subscribe
  - Asynchronní zpracování s prioritizací
  - Dávkové zpracování
  - Sběr dat z více zdrojů
  - Škálování výkonu a operací

alza.cz

Asynchronní trigger



alza.cz

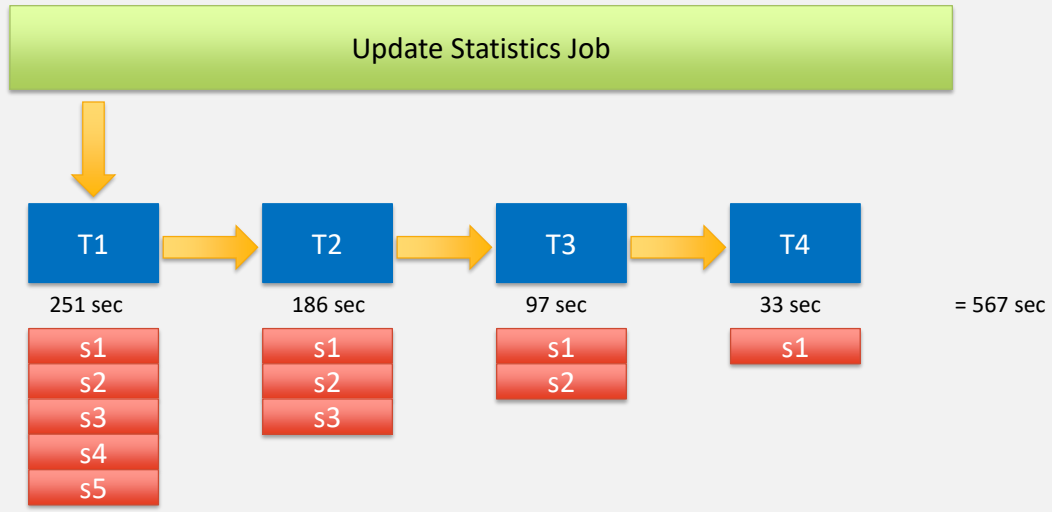
Asynchronní trigger

```
CREATE TRIGGER TRSalesDetailAudit
ON Sales.SalesOrderDetail
AFTER INSERT
AS

INSERT dbo.SalesDetailAudit
SELECT i.SalesOrderID, i.SalesOrderDetailID,
GETDATE(), SUSER_SNAME()
FROM inserted i

DECLARE @OrderTurnover decimal(10,4) = 0;

SELECT @OrderTurnover += i.LineTotal
FROM inserted i

UPDATE Sales.DailyTurnover
SET TotalDailyTurnover += @OrderTurnover
WHERE TurnoverDay = CAST(GETDATE() AS date)

GO
```

```
CREATE TRIGGER SalesDetailAudit
ON Sales.SalesOrderDetail
AFTER INSERT
AS

DECLARE @SalesDetail XML;
SET @SalesDetail =
(SELECT i.SalesOrderID, i.SalesOrderDetailID,
GETDATE() AS Datum, SUSER_SNAME() AS LoginName
FROM inserted i
FOR XML AUTO)

EXEC dbo.SendAuditMessage @SalesDetail

DECLARE @OrderTurnover decimal(10,4) = 0;

SELECT @OrderTurnover += i.LineTotal
FROM inserted i

EXEC dbo.SendTurnoverInsert @OrderTurnover
GO
```

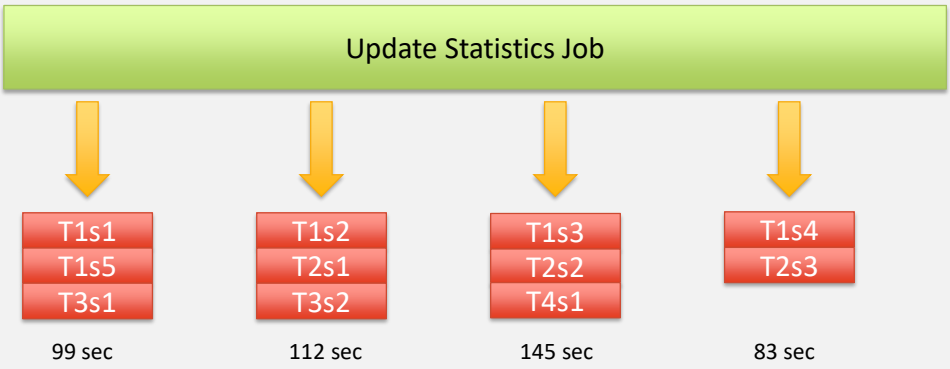
alza.cz

SQL Server maintenance



alza.cz

SQL Server maintenance



- Trace flag 7471
  - Dostupný od SQL 2014 SP1 CU6
  - Mění způsob uzamykání UPDATE STATISTICS z X na U
- SQLSkills.com – Jonatanhan Kehayias – Parallel Maintenance Tasks

alza.cz

Synchronní vs. Asynchronní – Database maintenance

- Mezi běžné úlohy údržby databáze patří
  - ALTER INDEX … REBUILD/REORGANIZE
  - UPDATE STATISTICS

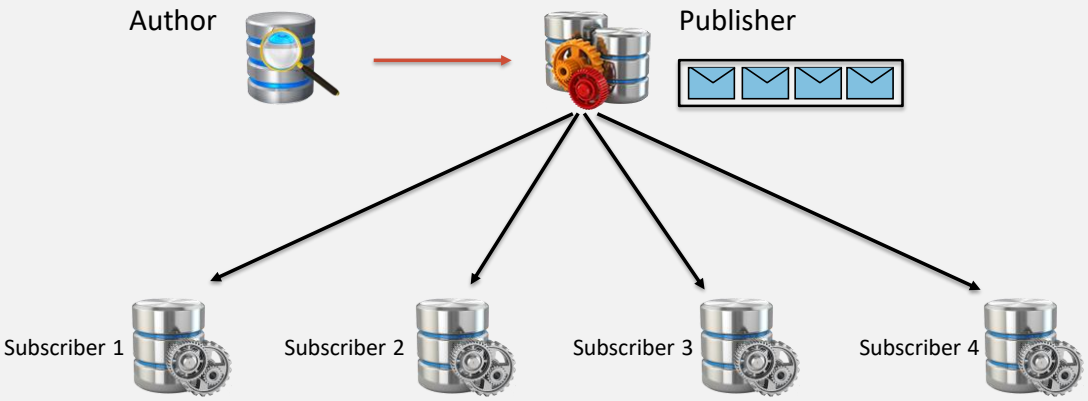
```
UPDATE STATISTICS dbo.Products
WITH FULLSCAN
-- 1 hour 28 mins

UPDATE STATISTICS dbo.SalesOrderHeader
WITH FULLSCAN
-- 13 hours 15 mins

UPDATE STATISTICS dbo.SalesOrderItem
WITH FULLSCAN
-- 4 hours 34 mins
-- …
-- Total - 33 hours 41 mins
```

```
SEND ON CONVERSATION @Dailog_stats1
MESSAGE TYPE [UPDATE_STATISTICS]
('<Table>dbo.Products</Table>')

SEND ON CONVERSATION @Dailog_stats2
MESSAGE TYPE [UPDATE_STATISTICS]
('<Table>dbo.SalesOrderHeader</Table>')

SEND ON CONVERSATION @Dailog_stats3
MESSAGE TYPE [UPDATE_STATISTICS]
('<Table>dbo.SalesOrderItem</Table>')
```

alza.cz

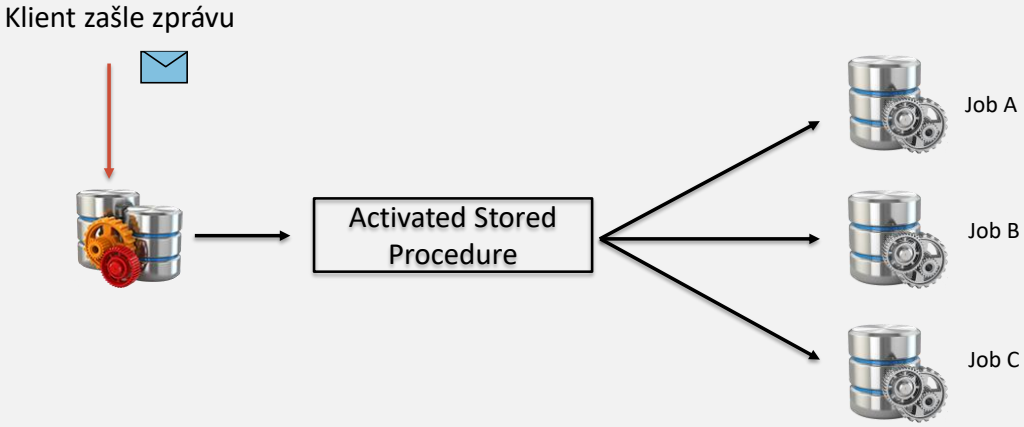
Publish-Subscribe

- Autor vytvoří zprávu a odešle ji vydavateli, který provede distribuci každému odběrateli
- Pokud odběratel není dostupný nebo nemůže zprávu přijmout, zůstává zpráva v Transmission queue do doby, než je úspěšně doručena



alza.cz

Batch Framework



alza.cz

Teoretické základy SB

alza.cz

Konverzace

Definice
- Spolehlivá a řazená výměna zpráv mezi dvěma službami Service Brokeru

Pro snazší představu
- Logický komunikační kanál, na základě kterého probíhá zasílání zpráv mezi službami SB
- Tento komunikační kanál poskytuje záruky kvality komunikace

- Dva typy konverzace
  - Monolog
    - Jednosměrná konverzace mezi publikační službou a několika službami odběratelů
    - Není podporováno
  - Dialog
    - Obousměrná konverzace mezi vždy dvěma službami
    - Iniciátor konverzace, cíl konverzace

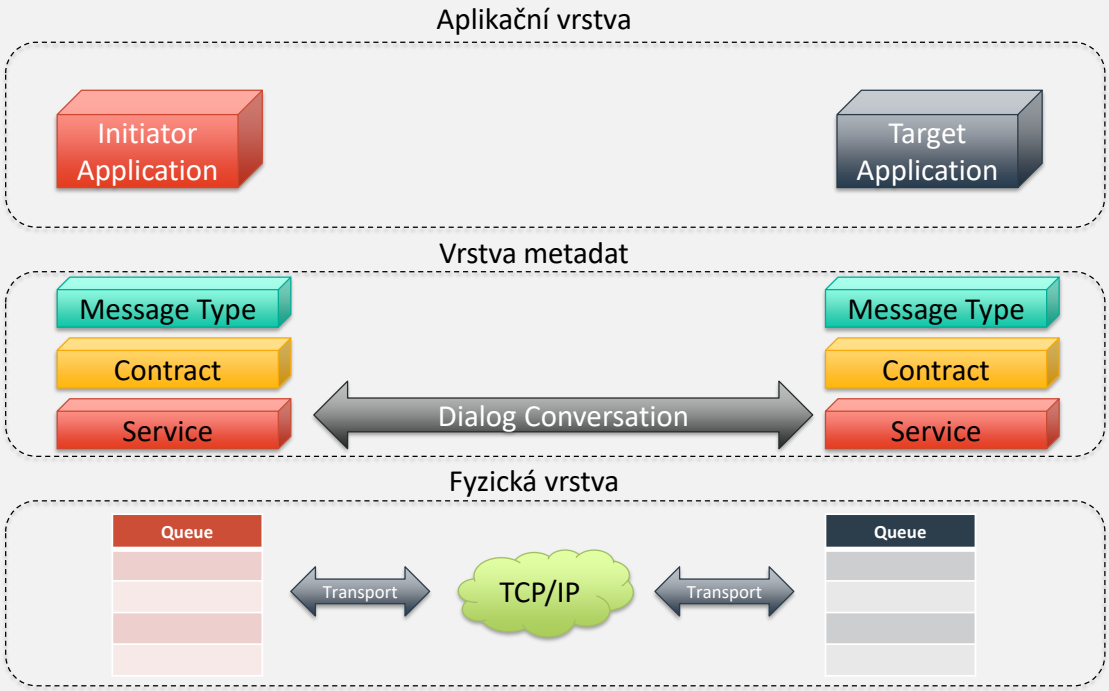
alza.cz

Dialog Conversation

- Obousměrná konverzace mezi dvěma službami Service Brokeru

- Vlastnosti dialogové konverzace :
  - Garantované doručení zprávy
    - Odesílatel zprávy má jistotu o doručení zprávy i když je příjemce aktuálně offline
  - Délka trvání konverzace
    - Od jednotek sekund po roky
  - Zpráva je doručena v jediném exempláři
    - Kontrolní mechanizmy zajistí, že i v případě problémů v komunikaci bude zpráva na cíl doručena pouze v jediném exempláři
  - Řazené doručení
    - Zprávy jsou doručeny vždy ve stejném pořadí, ve kterém byly odeslány
  - Trvalost dialogové konverzace
    - Dialog přežije restart databáze i celé databázové služby tzn.

alza.cz



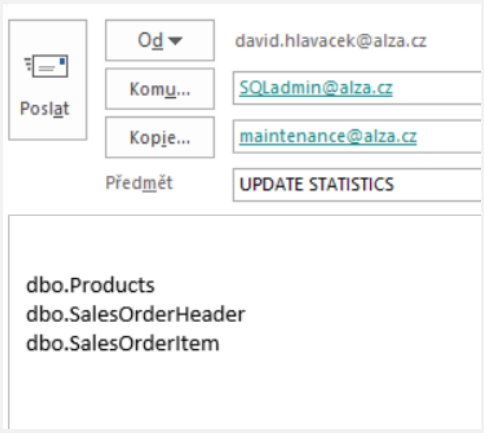
Service Broker – Konfigurace databáze

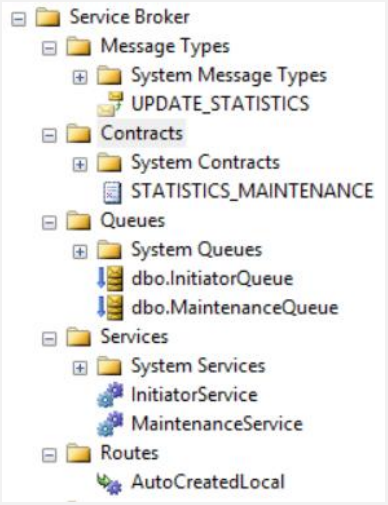
- V nově vytvořené databázi je SB deaktivován
- SB je deaktivován po RESTORE nebo ATTACH operaci databáze

```
ALTER DATABASE AdventureWorks
SET
```

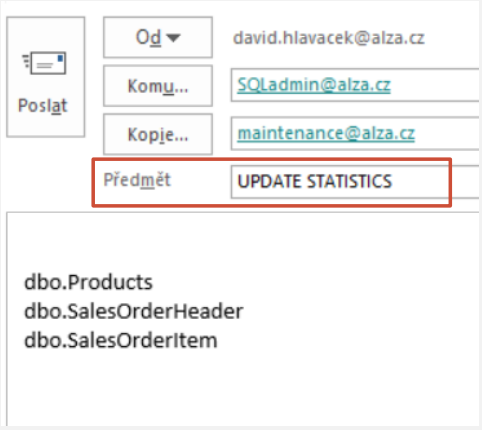
- ENABLE_BROKER
  - Aktivuje funkčnost SB v dané databázi, vyžaduje X lock na databázi
  - Ponechá stávající SB identifikátor
- DISABLE_BROKER
  - Deaktivuje funkčnost SB a ponechá stávající SB identifikátor
- NEW_BROKER
  - Vytvoří nový SB identifikátor pro databázi a ukončí všechny konverzace bez END DIALOG
  - Nutno opětovně vytvořit všechny ROUTE, které na danou DB směrovaly
- ERROR_BROKER_CONVERSATIONS
  - Aktivuje SB, ukončí všechny konverzace s chybou (čištění)
- HONOR_BROKER_PRIORITY

Service Broker





Service Broker – Message Type



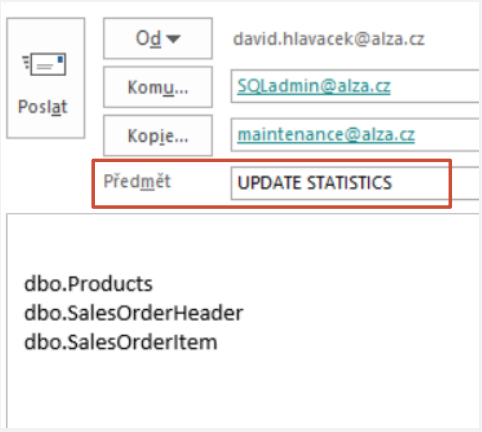
- Zpráva
  - Nositelem dat, instrukcí, informace
- Message Type
  - Jaké zprávy bude SB odesílat a přijímat ?

```
CREATE MESSAGE TYPE [UPDATE_STATISTICS]
VALIDATION = WELL_FORMED_XML
```

- Validation
  - EMPTY, NONE, WELL_FORMED_XML, VALID_XML
    - Performance VALID_XML ?

alza.cz

Service Broker – Contract

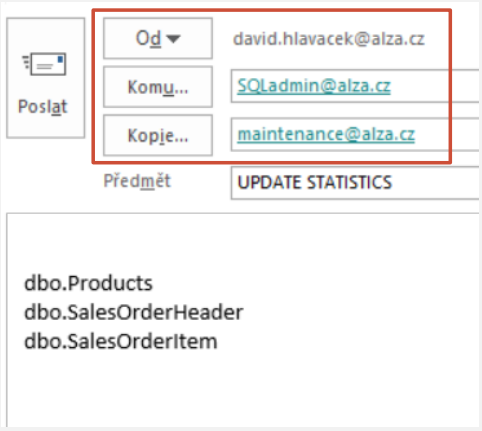


- Contract
  - Definuje typy zpráv, které mohou být při komunikaci zasílány mezi službami Service Brokeru
  - SEND BY INITIATOR
  - SEND BY TARGET
  - SEND BY ANY
  - Pokud se jedná o jednosměrnou komunikaci, není nutné v kontraktu odesílatele uvádět typ zprávy

```
CREATE CONTRACT [STATISTICS_MAINTENANCE]
( [UPDATE_STATISTICS] SENT BY INITIATOR )
```

alza.cz

Service Broker - Queue

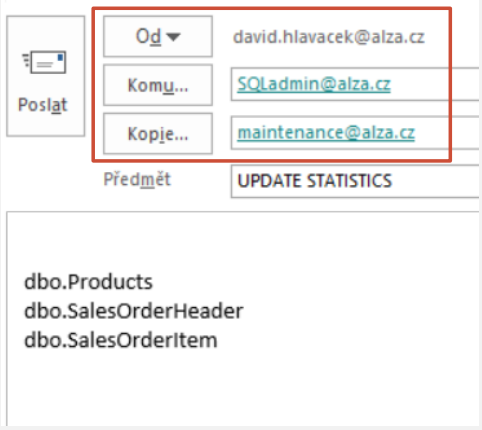


- Queue
  - Tabulka, do které jsou uloženy příchozí zprávy
  - Statická nebo aktivační

```
CREATE QUEUE InitiatorQueue
    WITH STATUS=ON

CREATE QUEUE MaintenanceQueue
WITH STATUS=ON,
ACTIVATION
(
 PROCEDURE_NAME = dbo.MaintenanceProcedure,
 MAX_QUEUE_READERS = 5, EXECUTE AS 'dbo'
) ;
```

alza.cz

Service Broker - Service

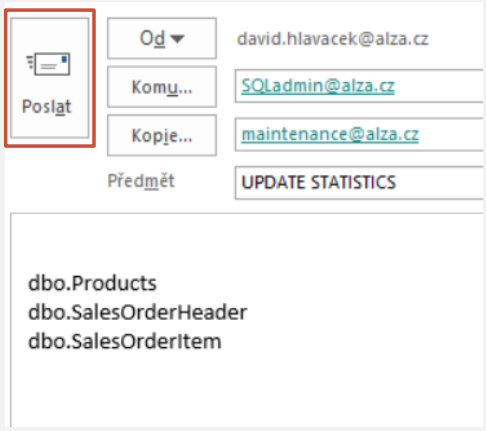


- Service
  - SB používá Service pro směrování a doručení zprávy do příslušné Queue v databázi
  - Pokud není definován kontrakt, může Service přijímat pouze zprávy komunikace, kterou iniciovala

```
CREATE SERVICE InitiatorService
    ON QUEUE [dbo].[InitiatorQueue]


CREATE SERVICE MaintenanceService
    ON QUEUE [dbo].[MaintenanceQueue]
    ([STATISTICS_MAINTENANCE]) ;
```

alza.cz

Service Broker - Dialog



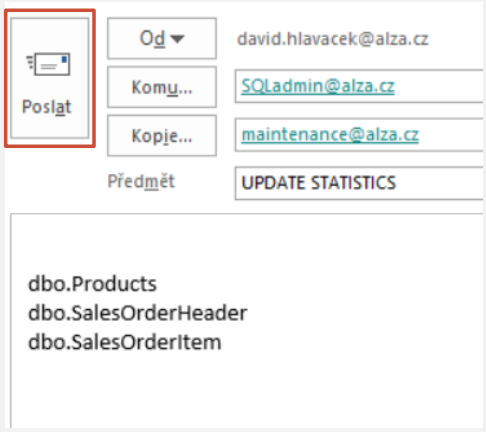
- Dialog
  - Logický komunikační kanál mezi dvěma službami
  - Slouží pro odesílání zpráv
  - Záznam v databázi na obou stranách komunikace

  - Garance doručení
  - Zpráva je doručena v jediném exempláři
  - Doručení v konkrétním pořadí
  - Bezpečnost

```
DECLARE @Dailog_stats1 uniqueidentifier;

BEGIN DIALOG CONVERSATION @Dailog_stats1
FROM SERVICE [InitiatorService]
TO SERVICE 'MaintenanceService'
ON CONTRACT [STATISTICS_MAINTENANCE]
WITH ENCRYPTION = OFF;
```

alza.cz

Service Broker - Send



- Send
  - Odešle zprávu za použití jedné nebo několika konverzací
  - Zpráva je umístěna do cílové Queue, pokud se příjemce nachází ve stejné SQL instanci a Queue je aktivní
  - Pokud se cíl konverzace nachází mimo instanci, je zpráva umístěna do sys.transmission_queue

```
SEND ON CONVERSATION @Dailog1
MESSAGE TYPE [UPDATE_STATISTICS]
('<Table>dbo.Products</Table>')
```

alza.cz

RECEIVE – výběr zpráv z fronty

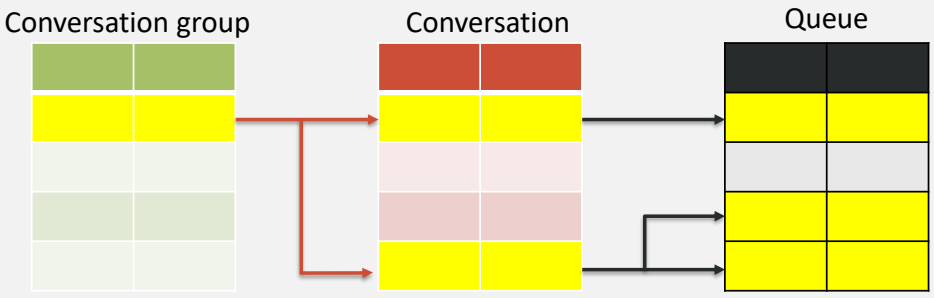
- Získá jednu nebo více zpráv z fronty
  - Pokud není uvedeno jinak, výstupem je resultset obsahující 0-n řádků zpráv
  - Výstup lze směrovat do tabulkové nebo skalárních proměnných
  - Po úspěšném vyzvednutí zprávy je zpráva z fronty odstraněna
    - Nutné použít adekvátní error handling a transakce
  - Při výběru zpráv je použit specifický způsob uzamykání

```
WAITFOR (
    RECEIVE TOP(1) conversation_handle, message_type_name,
    message_body FROM [dbo].[MaintenanceQueue]
    INTO @MaintenanceTab),
TIMEOUT 60000;
```

alza.cz

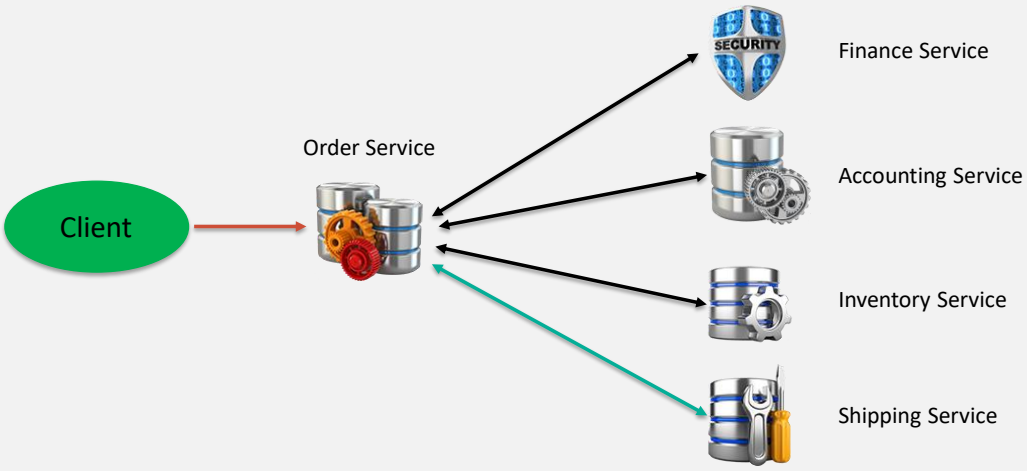
RECEIVE – způsob uzamykání při výběru zpráv

- Zpráva je doručena přes konkrétní konverzaci
- Konverzace jsou seskupovány do skupin – Conversation group
  - Slouží pro seskupování více konverzací za účelem identifikace jediného úkolu
  - Pokud nenajdete využití, není nutné používat a ve výchozím platí, že každá konverzace je seskupena jedinou conversation group



alza.cz

Škálování výkonu a operací



- Možnost využití jediné conversation group pro realizaci business tasku

alza.cz

Typy systémových zpráv Service Brokeru

- http://schemas.microsoft.com/SQL/ServiceBroker/EndDialog
  - Pokud iniciátor nebo cíl ukončí konverzaci, dojde k automatickému odeslání zprávy

```
IF @message_type = N'http://schemas.microsoft.com/SQL/ServiceBroker/EndDialog'
BEGIN
     END CONVERSATION @handle;
END
```

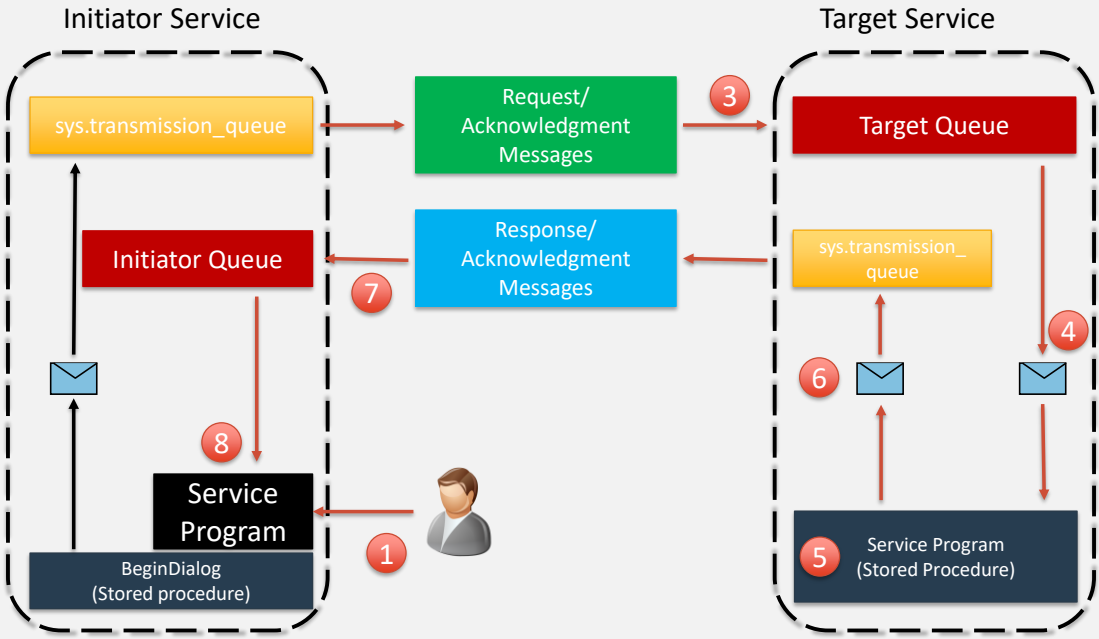
- http://schemas.microsoft.com/SQL/ServiceBroker/Error
  - Zpráva je odeslána v případě výskytu závažné chyby nebo ukončení konverzace chybou

```
END CONVERSATION @dialog_handle
WITH ERROR = 101 DESCRIPTION = N'Chyba
konverzace' ;
```

- http://schemas.microsoft.com/SQL/ServiceBroker/DialogTimer
  - Zpráva je umístěna do lokální fronty specifikované konverzace

```
BEGIN CONVERSATION TIMER (@dialog_handle)
TIMEOUT = 120 ;
```

alza.cz



Queue Activation

- Aktivace umožní automatické spuštění kódu, který zpracuje příchozí zprávy

- Interní
  - Nejběžněji používané řešení
  - SQL Server Queue monitor automaticky spouští uloženou proceduru v případě potřeby
  - Aktivační uložená procedura může použít T-SQL nebo CLR
  - Lze konfigurovat maximální počet paralelně spuštěných procedur
  - Exekuční kontextu
  - Používá zdroje SQL Serveru

- Externí
  - Obvykle se jedná o kombinaci interní aktivace a externího programu
  - Na přijetí zprávy do fronty reaguje Event Notification která odesílá zprávu do cílové služby
  - Cílová služba aktivuje proceduru, která upozorní externí program na přijetí zprávy
  - Používá se pro komplexní zpracování mimo SQL Server (mimo zdroje SQL Serveru)

alza.cz

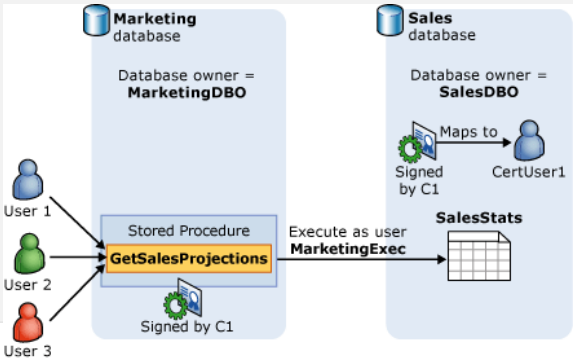
Security context aktivační procedury

- V aktivační proceduře může dojít k chybě, debugging se řeší jejím ručním spuštění
  - EXEC dbo.MaintenanceProcedure
  - Výsledek ? Žádná chyba… Aktivace v rámci Queue však stále nedělá co má…
- Byť je procedura volána pod dbo, stále se jedná o databázového uživatele
  - Nemá práva vystoupit mimo kontext své databáze a používat serverové objekty nebo přistupovat k jiným databázím

```
CREATE QUEUE MaintenanceQueue
WITH STATUS=ON, ACTIVATION
(PROCEDURE_NAME = dbo.MaintenanceProcedure,
 MAX_QUEUE_READERS = 5, EXECUTE AS 'dbo') ;
```


Security context aktivační procedury

- Database TRUSTWORTHY ON
  - Moduly s EXECUTE AS mohou použít server security context
  - Potencionální bezpečnostní riziko
- Použití certifikátů
  - Preferovaná, ale pracnější varianta
  - Na zdroji certifikátem podepíšeme proceduru a vytvoříme zálohu public key
  - V cíli vytvoříme certifikát ze zálohy a vytvoříme z něj uživatele, který má práva na cílovou tabulku



Poison message

```
CREATE QUEUE MaintenanceQueue
WITH STATUS=ON,ACTIVATION
( PROCEDURE_NAME = dbo.MaintenanceProcedure,
 MAX_QUEUE_READERS = 0, EXECUTE AS 'dbo'),
 POISON_MESSAGE_HANDLING (STATUS = ON);
```

- Vypne frontu v případě 5 po sobě jdoucích neúspěšných pokusech o RECEIVE z fronty
- Defaultně ON, lze deaktivovat od MS SQL 2012
  - Nutno implementovat vlastní ošetření chyb

- Protože Service Broker garantuje řazený příjem a zpracování zpráv, nemůže v případě chyby automaticky zprávu přeskočit a zpracovat další

alza.cz

Komunikace mezi různými SQL instancemi

alza.cz

Komunikace SB mezi různými instancemi SQL Serveru

- Při vývoji SB je lhostejné, kde budou fyzicky jednotlivé služby umístěny
  - Výměna práv mezi službami v jediné databázi
  - Výměna zpráv mezi službami v různých databázích stejné instance
  - Výměna zpráv mezi službami různých databází různých instancí

- Jaké objekty / konfigurace je třeba vyřešit :
  - Route
  - Endpoint
  - Security

Service Broker Endpoint

- Service Broker používá TCP/IP a umožňuje vícero dialogům použít jediné TCP/IP spojení
- Endpoint definuje port, na kterém SB naslouchá pro navázání komunikace
- V instanci SQL Serveru je možné definovat pouze jediný SB Endpoint

- Další možnosti :
  - ENCRYPTION
    - Default REQUIRED a od SQL 2016 AES
  - MESSAGE_FORWARDING
    - Umožňuje přeposílání zpráv službám, které se nenachází v dané instanci

```
CREATE ENDPOINT BrokerEndpoint
    STATE = STARTED
    AS TCP ( LISTENER_PORT = 4038 )
    FOR SERVICE_BROKER ( AUTHENTICATION = WINDOWS NEGOTIATE ) ;
```

Service Broker – Security

- Transport Security
  - Zajišťuje autentizaci mezi instancemi databázových služeb
  - Umožňuje šifrování zpráv
  - Windows authentication
    - NTLM nebo Kerberos, instance musí být v doméně s trustem (pozor na SPN)
    - Pozor na čekání autentizace ovlivňující celkový výkon
  - Certificate authentication
    - Autentizace probíhá na základě certifikátů (TLS)
    - Certifikát musí být vždy chráněn master klíčem a nikoliv heslem

- Dialog Security
  - Zajišťuje autentizaci mezi SB službou iniciátora a cíle
  - Umožňuje šifrování zpráv
    - Zprávy jsou šifrovány a dešifrovány mezi konkrétními dvěma službami

Transport Security

- Windows
  - Vytvořit Login pro Service account vzdálené instance

```
CREATE LOGIN [ALZA\OrderLogin] FROM WINDOWS
```

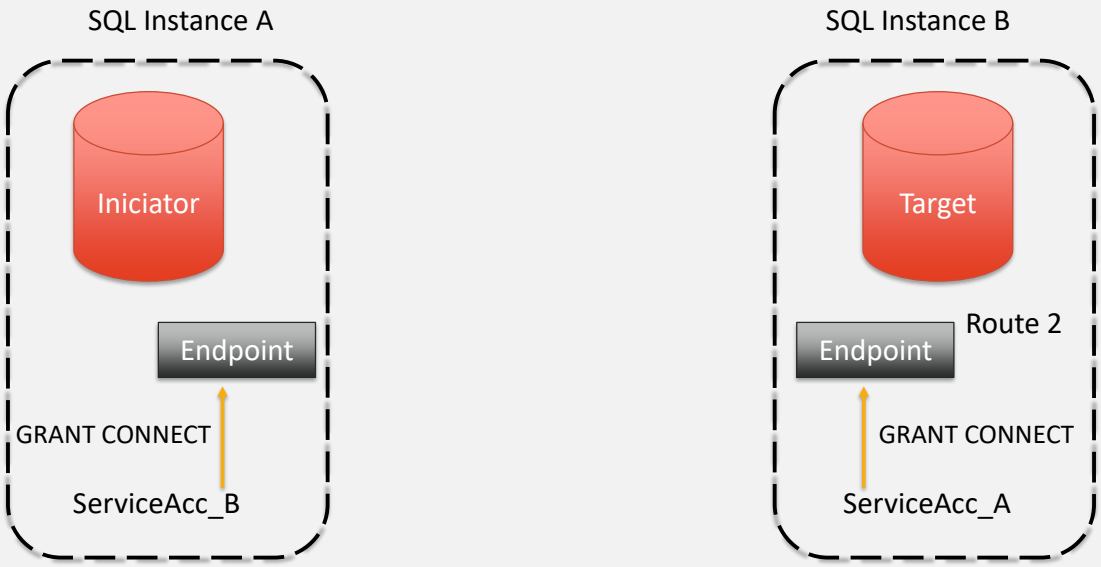
  - Přidělení oprávnění CONNECT k SB Endpointu instance

```
GRANT CONNECT ON ENDPOINT::[BrokerEndpoint] TO [Alza\OrderService]
```

  - V případě, kdy nechceme řešit Dialog Security

```
GRANT SEND ON SERVICE::[AccountingService] TO PUBLIC
```

Transport Security - Windows



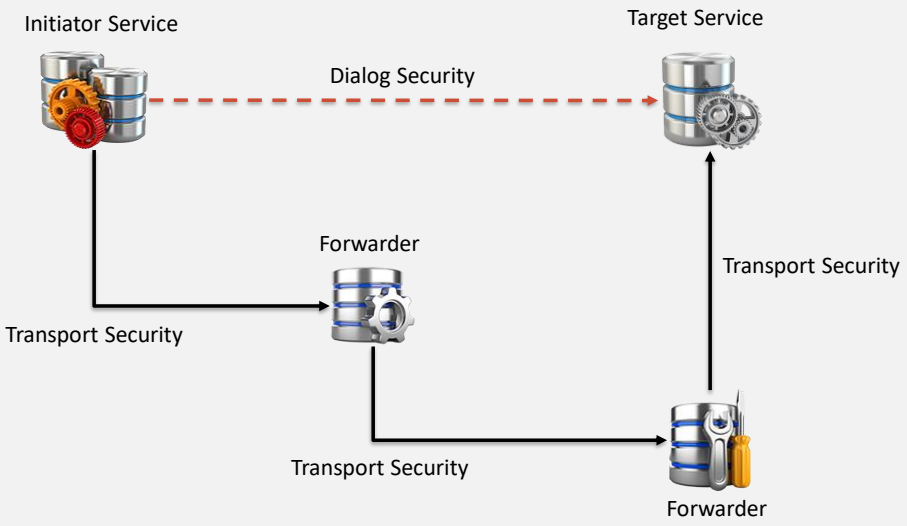
alza.cz

Transport Security

- Certificate
  - Iniciátor i cíl vytvoří v databázi master certifikát
  - Iniciátor i cíl nakonfigurují endpoint s vlastností AUTHENTICATION = CERTIFICATE
  - Iniciátor i cíl provedou backup public klíče svého certifikátu
  - Iniciátor i cíl vytvoří SQL Login s heslem a v databázi master na něj namapují uživatele
  - Iniciátor i cíl přiřadí oprávnění CONNECT na Endpoit vytvořenému Loginu
  - Iniciátor i cíl vytvoří certifikát ze zálohy svého protějšku v databázi master a jeho vlastnictví nastaví vytvořenému uživateli

alza.cz

Transport vs. Dialog Security



alza.cz

ROUTE

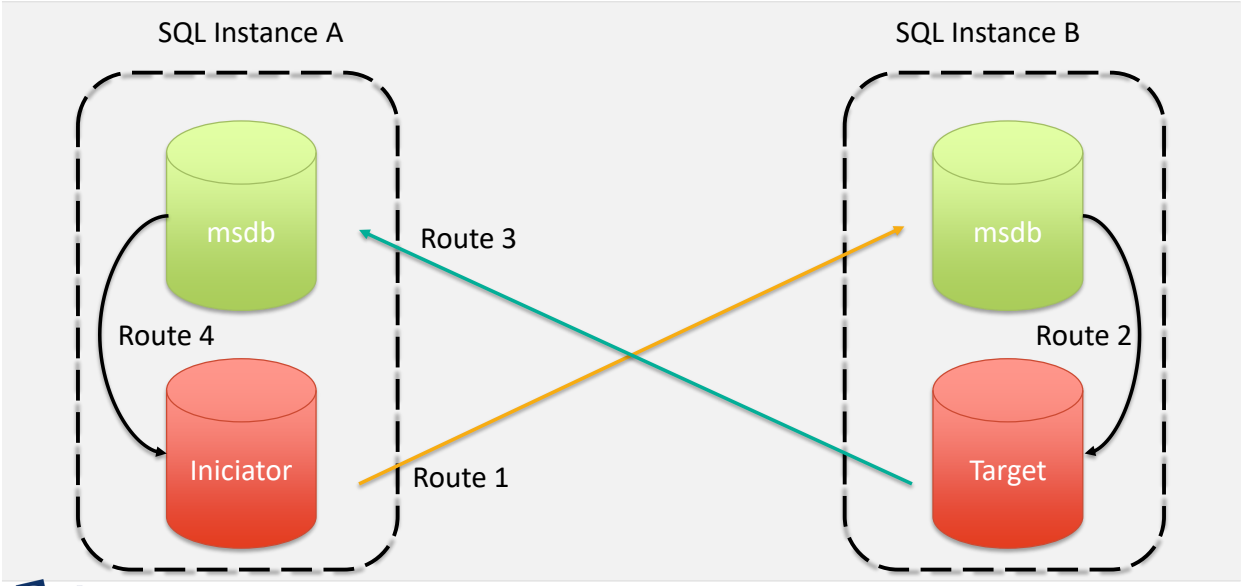
- Databázový objekt sloužící po směrování zpráv
- Mapuje název služby (případně i Broker Instance) na síťovou adresu

- Při odesílání zprávy dochází k její klasifikaci
  - SQL Server používá routing algoritmus pro nalezení cíle zprávy
  - Pokud se v instanci nachází více služeb se shodným názvem, SB může toto brát jako strategii škálování výkonu a směrovat zprávy do služeb náhodně

```
CREATE ROUTE OrderRoute
WITH
    SERVICE_NAME = 'OrderService',
    BROKER_INSTANCE = 'F76D0420-3048-49E0-ABC9-6433156CC101',
    ADDRESS = 'TCP://SQLInstance1.alza.cz:5001'
```

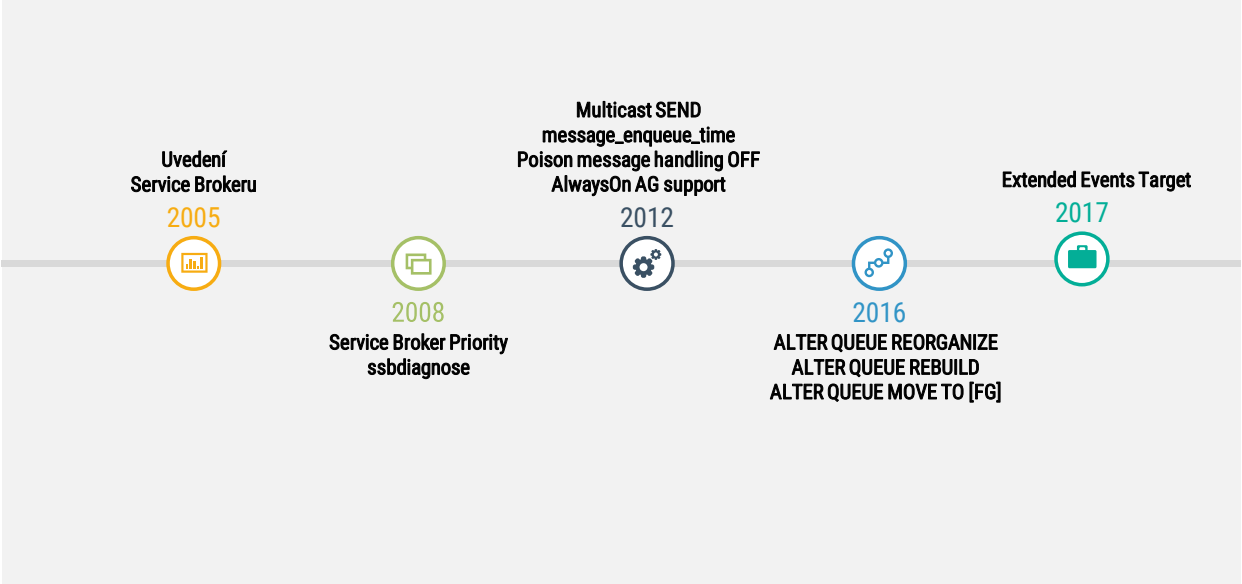
alza.cz

ROUTE



alza.cz

Historie a rozšíření Service Brokeru



Strategie a techniky optimalizace SB Performance

alza.cz

Service Broker – „Fire and forget“ scénář

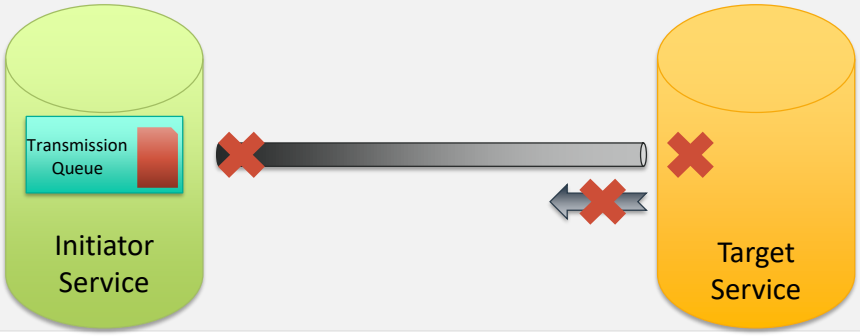
1. Initiator vytvoří dialog s Target
2. Initiator odešle zprávu příkazem SEND
3. Initiator ukončí dialog
4. Target vidí dialog ve stavu DISCONNECTED_OUTBOUND

alza.cz

Service Broker – „Fire and forget“ scénář

Problémy :

- Pokud je ukončen dialog, není možné přes něj přijmout zprávu
  - Pokud odeslaná zpráva nemůže být doručena (např. chyba na Target), protože má Initiator již dialog ukončen, nemůže přijmout informaci o chybě
    - Např. chyba security, XML validace, Target byl odstraněn nebo chyba contractu atd.



alza.cz

Service Broker – Použití dialogů

- Strategie „data push“
  - Jednosměrná komunikace a zaslání zprávy Initiator -> Target
  - Target nezasílá zprávy (vyjma systémových EndDialog apod.)
  - Velmi jednoduché na programování
    - Initiator vytvoří konverzaci, zašle zprávu
    - Target přijme zprávu, ukončí konverzaci
    - Initiator obdrží EndDialog a ukončí konverzaci

- Strategie „conversation reuse“
  - Vytvoření konverzace, odeslání jediné zprávy a ukončení konverzace je náročné na zdroje
  - Pokud je komunikace mezi dvěma službami častá, je vhodnější nechat konverzace otevřené a opětovně je používat

alza.cz

Optimalizační techniky

- Přijetí více zpráv
  - Použití jediné konverzace má dopad na SEND zprávy
    - Řádek konverzace v sys.sysdesend musí být aktualizován při odeslání
  - Vytvoření a použitím několika konverzací optimalizujeme odesílání zpráv

- Zasílání více zpráv přes konverzaci
  - Jednou z nejméně optimálních metod je tvorba konverzace, odeslání jediné zprávy a ukončení konverzace
    - Neustálý INSERT a UPDATE tabulky sys.sysdesend
  - Vytvořte konverzaci a používejte ji pro odesílání zpráv opakovaně
    - Ukončujte konverzaci na základě limitu počtu zpráv nebo času

alza.cz

Optimalizační techniky

- Použití více zpráv z konverzace
  - Každý aktivační task je schopen přijmout zprávy pouze z jediné konverzace
  - Pokud je konverzací zasláno několik zpráv, je možné provést RECIEVE několika
    - RECEIVE TOP(1) vs. RECAIVE TOP(N)
  - Za optimální způsob vyzvedávání zpráv se považuje :
    - RECEIVE TOP(N) (kde N > 1)
    - Uložení těchto zpráv do tabulkové proměnné
    - Procesování zpráv

  - Vyzvednutí zpráv příkazem RECEIVE by mělo být vždy součástí explicitní transakce
    - Nutnost zvážit transakci při zpracování nebo logování zpráv

- Počet aktivačních tasků by měl být maximálně roven počtu jader

alza.cz

Optimalizační techniky

- 150 trik
  - Pro odesílání zpráv jsou nutné konverzace
    - Jejich UPDATE v sys.sysdesend způsobuje contention na stránky této tabulky
  - Odesílání zpráv je náročnější než jejich příjem
    - 3 transakce
  - Pro maximalizaci výkonu při odesílání zpráv je vhodné použít každou 150to konverzaci
    - Na 8 kB stránku je možné umístit pouze 144 záznamů o konverzacích

Service Broker – Aktivační procedury a výkon

Metodika aktivačních procedur :

- Základní RECIEVE
  - V transakci vždy vybere jednu zprávu, zpracuje a potvrdí transakci
- Dávkové zpracování
  - Zahájí transakci, v cyklu vybere jednotlivé zprávy a zpracuje, potvrzuje až po zpracování definovaného počtu zpráv
- Kurzor
  - Zahájí transakci, vybere všechny zprávy konverzace, uloží je do @table a kurzorem v cyklu zpracovává jednu po druhé, ukončí transakci
- Množinové zpracování
  - Zahájí transakci, vybere všechny zprávy konverzace, uloží je do @table a jediným příkazem je všechny zpracuje, ukončí transakci

RECEIVE metody

Metoda	RECEIVE zpráv/sec	Čas pro 10k zpráv	Performance
Basic	772	13 sec	100%
Batch	853	12 sec	110%
Cursor	4629	2 sec	599%
Cursor Message Type	4967	2 sec	643%
Set based XML	5319	2 sec	688%
Set based Binary	6108	1 sec	791%

alza.cz

RECEIVE metody - Basic metoda

- Vyzvednutí a zpracování jediné zprávy v rámci jediné transakce

```
WHILE (1=1)
BEGIN
    BEGIN TRANSACTION
    WAITFOR(RECEIVE TOP(1)
            @h = conversation_handle,
            @messageTypeName = message_type_name,
            @payload = message_body
            FROM [Target]), TIMEOUT 1000;
IF (@@ROWCOUNT = 0)
BEGIN
    ROLLBACK;
    BREAK;
END;
```

alza.cz

RECEIVE metody - Batch metoda

- Vyzvednutí a zpracování jediné zprávy, COMMIT transakce po zpracování dávky

```
BEGIN TRANSACTION;
WHILE (1=1)
BEGIN
    WAITFOR(RECEIVE TOP(1)
            @h = conversation_handle,
            @messageTypeName = message_type_name,
            @payload = message_body
            FROM [Target]), TIMEOUT 1000;
    SELECT @batchCount = @batchCount + 1
    IF @batchCount >= 100
    BEGIN
        COMMIT;
        SELECT @batchCount = 0;
        BEGIN TRANSACTION;
    END
END
```

alza.cz

RECEIVE metody - Cursor metoda

- Vyzvednutí všech zpráv z fronty a uložení do tabulkové proměnné
  - RECEIVE bez TOP vyzvedne všechny zprávy ze stejné dialog group

- Kurzor s WHILE procesuje zprávy z tabulkové proměnné
- K potvrzení transakce dochází až po zpracování všech zpráv kurzorem

```
DECLARE @tableMessages TABLE
DECLARE cursorMessages CURSOR FORWARD_ONLY READ_ONLY
FOR SELECT ... FROM @tableMessages

WHILE (1=1)
BEGIN
BEGIN TRANSACTION;
WAITFOR(RECEIVE … FROM [Target]
INTO @tableMessages), TIMEOUT 1000;
```

alza.cz

RECEIVE metody - Cursor metoda

```
OPEN cursorMessages;
WHILE (1=1)
BEGIN
    FETCH NEXT FROM cursorMessages
    INTO @h, @messageTypeName, @payload;
...
END

CLOSE cursorMessages;
DELETE FROM @tableMessages;
COMMIT;
END
DEALLOCATE cursorMessages;
```


RECEIVE metody - Cursor Message Type metoda

- Zcela shodná z Cursor metodou
- Z Queue však nezískává message_type_name ale message_type_id
- Při testování typu zprávy tak nedochází ke konverzi int vs. string

```
IF @messageType = 2
    BEGIN
        END CONVERSATION @h;
    END

IF @messageType = 14
BEGIN
...
END
```

RECEIVE metody - Set based XML metoda

- Set based metody není možné použít plošně u všech typů aplikací
  - Omezené možnosti při zasílání různých typů zpráv

```
DECLARE @tableMessages TABLE

WHILE (1=1)
BEGIN
BEGIN TRANSACTION;

WAITFOR(RECEIVE queuing_order, conversation_handle,
message_type_name, CAST(message_body AS XML) AS payload
FROM [Target] INTO @tableMessages), TIMEOUT 1000;
```

alza.cz

RECEIVE metody - Set based XML metoda

```
;WITH XMLNAMESPACES (DEFAULT 'http://alza.cz/DHL/Datagram')
INSERT INTO [PayloadData] ([DateTime], [Payload], [User],
[Original])
SELECT payload.value(N'(/Datagram/@date-time)[1]', 'DATETIME'),
payload.value(N'(/Datagram/@payload)[1]', 'NVARCHAR(MAX)'),
payload.value(N'(/Datagram/@user)[1]', 'NVARCHAR(256)'),
payload
FROM @tableMessages
WHERE message_type_name = N'DEFAULT'
ORDER BY queuing_order;

COMMIT;
DELETE FROM @tableMessages;
END
```

alza.cz

Děkuji za pozornost

David Hlaváček
MCP, MCSE, MCSA, MCTS, MCITP, MCT
david.hlavacek@alza.cz
david@hlavacek.pro

alza.cz

alza.cz