Service Oriented Database Architectuur met SQL Server 2005

# SQL Server Service Broker

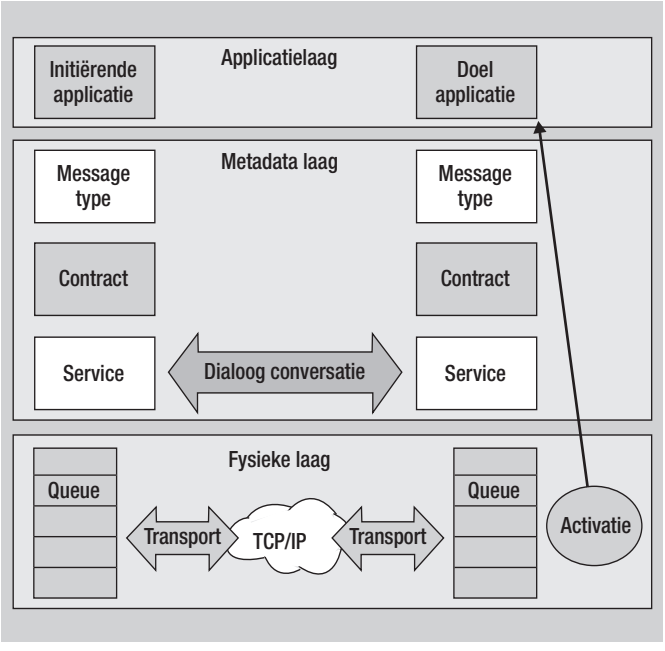
Astrid Hackenberg en Anko Duizer

**In het artikel in DB/M 2 is geschreven over de Service Oriented Database Architectuur (SODA). In een SODA smelten de service concepten samen met de databasewereld. Microsoft SQL Server 2005 beschikt over veel functionaliteit die het mogelijk maakt om een SODA te realiseren.**

Een belangrijk onderdeel om een SODA te realiseren met Microsoft SQL Server 2005 is de SQL Service Broker (SSB). De SSB zorgt ervoor dat de database een *queue* wordt die via services te benaderen is. In dit tweede artikel worden de concepten van de SSB behandeld. Daarnaast komen de belangrijkste T-SQL uitbreidingen aan de orde die een SSB implementatie ook daadwerkelijk mogelijk maken.

## Wat is de SQL Server Service Broker?

De SQL Service Broker (SSB) realiseert de combinatie van queues met services in de database. Voor de meeste programmeurs is het eenvoudiger om te programmeren met services dan met berichten en queues. Daarom zorgt SSB bovenop het queue-model voor een abstractielaag in de vorm van services. Dit past uitstekend in de SODA gedachtewereld.




**Afbeelding 1:** SSB architectuuroverzicht.

| Laag | Omschrijving |
|---|---|
| Applicatie | Deze laag moet zelf worden gebouwd. Hierin draaien programma's (meestal stored procedures) waarin gebruik wordt gemaakt van de services. Dit is een functionele laag. |
| Metadata | Deze laag beschrijft de specifieke SSB implementatie. Hierin worden de services gedefinieerd die gebruikt kunnen worden door de applicaties. |
| Fysiek | De fysieke laag zorgt voor de daadwerkelijke distributie en persistentie van de berichten die tussen applicaties worden verzonden. |

**Tabel 1:** Overzicht van lagen in de SSB architectuur.

SSB zorgt voor betrouwbare conversaties die tussen services worden opgezet. De berichten worden gegarandeerd slechts eenmaal verzonden. Dit kan onderdeel zijn van een SQL Server transactie. Achter een service kan een programma worden aangeroepen (automatisch of handmatig). Een service-programma wordt geïmplementeerd met een stored procedure. Een dergelijke stored procedure kan eventueel draaien in de CLR. Hier komt de integratie met .NET om de hoek kijken. Conversaties kunnen zowel lokaal als 'remote' zijn.

Om een SSB oplossing te realiseren biedt Microsoft SQL Server 2005 een uitbreiding op T-SQL. Er zijn bijvoorbeeld commando's om een queue te creëren of om de queue te lezen. De basis T-SQL commando's komen in dit artikel aan bod.

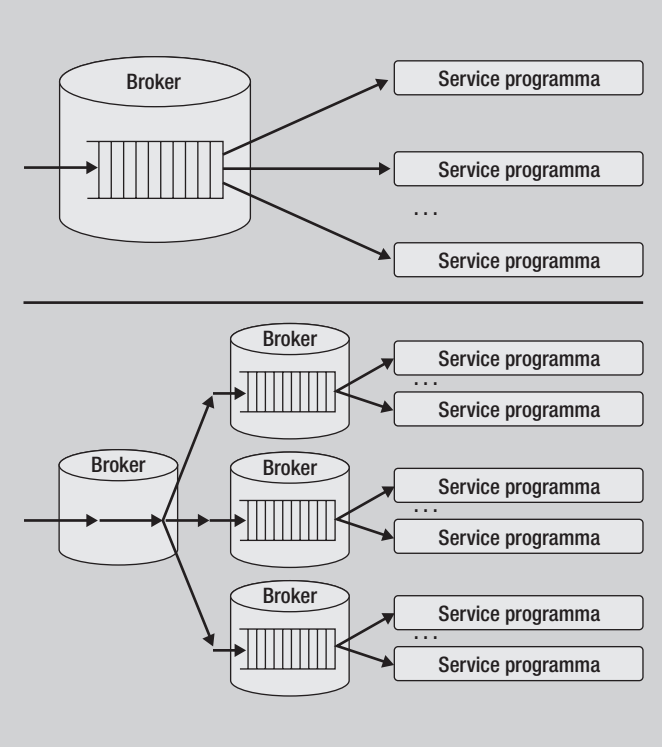
## Een overzicht van de SSB architectuur

De SSB infrastructuur levert de basisvereisten van een service-oplossing. De SQL Service Broker is een volledige Microsoft SQL Server 2005 oplossing. De gehele SSB implementatie is in de database gerealiseerd. Afbeelding 1 geeft een schematisch overzicht van de gelaagdheid zoals deze wordt geïmplementeerd door SSB. In tabel 1 zijn drie lagen te onderkennen.

De service interface is belangrijk, zeker binnen een SODA. De services zorgen voor abstractie doordat de applicaties die gebruik maken van de SSB alleen de services kennen. De implementatie

daarachter kan wijzigen, zonder dat de applicaties hiervan 'last' hebben. Zowel de metadata-laag als de fysieke laag worden altijd geïmplementeerd door Microsoft SQL Server 2005. De applicatie-laag kan in de database draaien, maar kan bijvoorbeeld ook een .NET applicatie zijn buiten de database.
De SSB architectuur biedt mogelijkheden voor scale-out, zoals weergegeven in afbeelding 2. Lokaal kunnen meerdere service-programma's worden gestart. Daarnaast kan een SSB implementatie remote communiceren om de werkdruk te verdelen. Dit kan over meerdere fysieke instanties van Microsoft SQL Server 2005 of over fysieke machines. De SSB infrastructuur draagt zorg voor een betrouwbare communicatie.

De volledige SSB communicatie vindt plaats via 'messages'. De ene applicatie verstuurt een verzoek naar een andere applicatie door het zenden van een bericht naar een service. De service ontvangt het bericht. De SSB infrastructuur draagt er vervolgens zorg voor dat het bericht de andere applicatie bereikt. Dit kan via een pull- of push-mechanisme. De ontvangende applicatie kan er voor kiezen om zelf service-verzoeken uit te voeren om op die manier te vragen of er nog berichten zijn om te verwerken. De SSB infrastructuur biedt ook de mogelijkheid om een 'push'-model te implementeren. Zodra een bericht binnenkomt wordt het gekoppelde service-programma geactiveerd.




**Afbeelding 2:** Scale-out met SSB.

> Voor het maken en lezen van queues zijn specifieke T-SQL statements aanwezig

In afbeelding 3 worden alle objecten schematisch weergegeven die samen het SSB framework vormen. De verschillende onderdelen van dit framework worden verderop in dit artikel in meer detail behandeld. Alle SSB objecten moeten een unieke naam hebben in de database. Dit is vrije tekst, maar meestal wordt een URN naamgeving gehanteerd. De voorbeeldcode in dit artikel maakt ook gebruik van de URN naamgeving.

## Wat is een conversatie?

De verschillende SSB objecten zorgen ervoor dat er een conversatie tussen applicaties kan plaatsvinden. Het basisconcept van de SSB is de conversatie. De conversatie zorgt voor een berichtuitwisseling in een vaste volgorde. Er zijn twee vormen van conversaties:

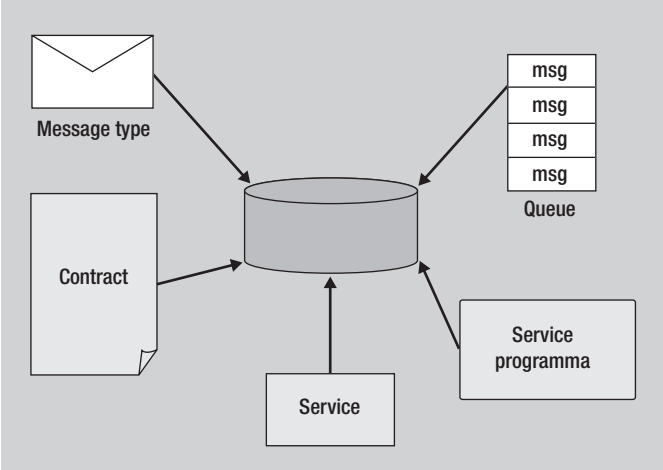
- Dialoog, een conversatie tussen twee applicaties waarbij de berichten beide kanten op verstuurd kunnen worden. De applicatie die de conversatie start door het zenden van het eerste bericht wordt de 'initiator' genoemd. De andere partij wordt 'target' genoemd. Het eerste bericht gaat altijd van 'initiator' naar 'target'. Hiermee is de conversatie tot stand gebracht. Nu kunnen berichten in willekeurige richting over en weer worden verstuurd;
- Monoloog, dit is eenrichtingsverkeer van een 'initiator' naar één of meerdere 'targets'. Dit komt overeen met een 'publish-subscribe' concept. Deze manier van converseren is nog niet geïmplementeerd in Microsoft SQL Server 2005. In de toekomst zal dit wel geïmplementeerd worden door Microsoft.

Een dialoog wordt in afbeelding 4 schematisch weergegeven. In T-SQL zijn statements beschikbaar om een conversatie te voeren. Voorbeeldcode 1 toont hoe een dialoog gestart wordt.

```
BEGIN DIALOG CONVERSATION @dialog_handle
  FROM SERVICE [//class-a.nl/Training/
                                InschrijvingService]
    TO SERVICE '//class-a.nl/Training/
                          ProcessInschrijvingService'
```

**Voorbeeldcode 1:** Starten van een dialoog in T-SQL.

Een unieke eigenschap die de conversatie met zich meebrengt is het gegeven dat berichten gegarandeerd in de juiste volgorde aankomen. De meeste messaging-systemen garanderen wel dat een bericht aankomt, maar niet dat de oorspronkelijke volgorde waarin de berichten verzonden zijn gehandhaafd blijft. SSB garandeert de volgorde zelfs over verschillende transacties heen.




**Afbeelding 3:** Overzicht van SSB onderdelen.

## Messages

Applicaties wisselen berichten met elkaar uit. Deze berichten zijn altijd van het SQL Server datatype varbinary(max). Dit is een 'emmer' van maximaal 2 GB aan informatie. De applicatie kan hierin van alles verzenden. Om te bepalen hoe het bericht eruit moet zien en waar de inhoud van het bericht aan moet voldoen, wordt een message type gedefinieerd. Een message type wordt geïdentificeerd door een zelf gekozen logische naam. In voorbeeldcode 2 wordt in T-SQL een message type gedefinieerd.

```
CREATE MESSAGE TYPE
  [//class-a.nl/Training/Inschrijving]
  VALIDATION = VALID_XML
  WITH SCHEMA COLLECTION classaschemas
```

**Voorbeeldcode 2:** Het creëren van een message type.

Er wordt een bericht gedefinieerd voor een training-inschrijving. Naast de logische naam wordt een tweetal zaken gedefinieerd die informatie geven over het type bericht. Ten eerste wordt bepaald dat het bericht van dit type altijd 'valid xml' moeten zijn.
Ten tweede wordt een XML schema, genaamd classaschema gekoppeld aan het bericht. Wanneer een conversatie gebruik wil maken van dit message type zijn er dus strenge eisen waaraan het bericht moet voldoen.
Een bericht hoeft geen XML bestand te zijn. Het blijft een varbinary(max) waarin alles opgslagen kan worden. Het is wel gebruikelijk om XML berichten te gebruiken in een SSB oplossing. Een XML bericht hoeft vervolgens weer niet per se aan een XML schema te zijn gekoppeld. Wanneer een XML schema aan een message type wordt gekoppeld, dan dient dit schema wel geregistreerd te zijn in Microsoft SOL Server. Er zijn T-SQL statements in Microsoft SQL Server om een XML schema te registreren en op te slaan als metadata. Deze statements vallen buiten de scope van dit artikel maar komen in een volgend artikel van deze reeks aan de orde.

## Contract

Een Service Broker contract is het volgende object dat gedefinieerd moet worden op weg naar een SSB oplossing. Een contract legt vast welke boodschappen binnen een conversatie heen en weer gezonden kunnen worden. Een contract bepaalt de interface van de service en zorgt daarmee voor een vereenvoudiging voor de applicatie-ontwikkelaar. Doordat de mogelijke berichten bekend zijn, zijn dit ook de enige berichten waarmee rekening hoeft te worden gehouden bij het gebruik van de service.
Tijdens de definitie van een contract wordt vastgelegd welk type bericht verzonden mag worden door de 'initiator', de 'target' of door beide. In voorbeeldcode 3 wordt een eenvoudig contract gedefinieerd met T-SQL statements. In de voorbeeldcode wordt duidelijk dat een contract een naam moet hebben. Vervolgens worden de berichttypen gedefinieerd die onderdeel zijn van het contract.

```
CREATE CONTRACT
[//class-a.nl/Training/InschrijvingContract]
([//class-a.nl/Training/Inschrijving]
       SENT BY INITIATOR,
 [//class-a.nl/Training/InschrijvingResponse]
       SENT BY TARGET
```

**Voorbeeldcode 3:** Het creëren van een contract in T-SQL.

## Queues

De queues zijn het hart van de SSB architectuur. De queues zorgen voor de asynchroniteit. Bij asynchrone verwerking is het mogelijk dat een verzoek (bericht) op een later moment wordt verwerkt door de ontvangende partij. Het bericht wordt tijdelijk bewaard. De plaats waar het bericht wordt bewaard wordt een queue genoemd.

> Een bericht blijft een varbinary(max) waarin alles opgslagen kan worden

Het is heel natuurlijk om een queue in een database te plaatsen. Het gaat uiteindelijk om het (tijdelijk) persistent bewaren van data. Hiervoor zijn databases bedoeld. Microsoft SQL Server 2005 implementeert een queue dan ook als een *hidden* tabel.
Deze tabellen zijn echter niet rechtstreeks benaderbaar met een DML statement. Het is dus niet mogelijk om een bericht in de tabel via een INSERT statement toe te voegen. Het is daarentegen wel mogelijk om via een SELECT statement de data te lezen die in een queue staan.

Voor het maken en lezen van queues zijn ook specifieke T-SQL statements aanwezig. In voorbeeldcode 4 wordt een queue aangemaakt. Ook de queue moet een unieke naam hebben in de database. Na creatie is de queue beschikbaar voor gebruik.

```
CREATE QUEUE InschrijvingQueue
```

**Voorbeeldcode 4:** Het creëren van een queue in T-SQL.

Berichten worden niet rechtstreeks in een queue geplaatst, dit gebeurt via een service. Het lezen van de queue om de berichten te verwerken gebeurt wel rechtstreeks. Hiervoor is een T-SQL statement beschikbaar zoals in voorbeeldcode 5. In dit voorbeeld wordt één bericht uit de queue gelezen. Het lezen van de berichten gebeurt altijd binnen de context van een conversatie. Wanneer de RECEIVE binnen een transactie wordt uitgevoerd, blijft het bericht in de queue tijdens het lezen en de verwerking. Zodra de transactie 'commit' is het bericht verwijderd uit de queue. Binnen de transactie is het bericht 'gelocked' en kan dus niet door anderen worden gelezen.

```
RECEIVE TOP(1)@conversation=conversation_handle,
              @msgType= message_type_name,
              @msg= message_body
FROM InschrijvingQueue
```

**Voorbeeldcode 5:** Berichten uit een queue lezen.

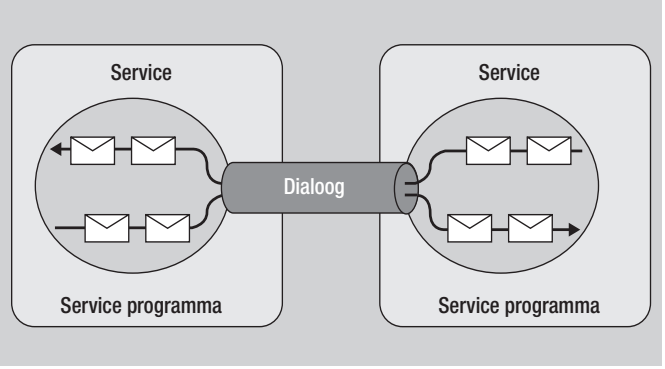
Omdat de queue een volwaardig database-object is, maakt de queue ook gebruik van de standaard database-functionaliteiten. Het is dus mogelijk om een queue onderdeel te laten zijn van een backup of restore. Dit is een belangrijk voordeel ten opzichte van andere messaging-oplossingen.

## Services

Applicaties spreken via services de SSB infrastructuur aan. Waarom wordt niet rechtstreeks met de queue gecommuniceerd? De reden hiervoor is abstractie, en de service is de SSB abstractie-laag boven op het queue-mechanisme. Een service kan bijvoorbeeld meerdere queues vertegenwoordigen via load balancing. Of een queue kan van machine verhuizen. Uiteindelijk heeft een applicatie hier geen last van, omdat er wordt gecommuniceerd via de services.

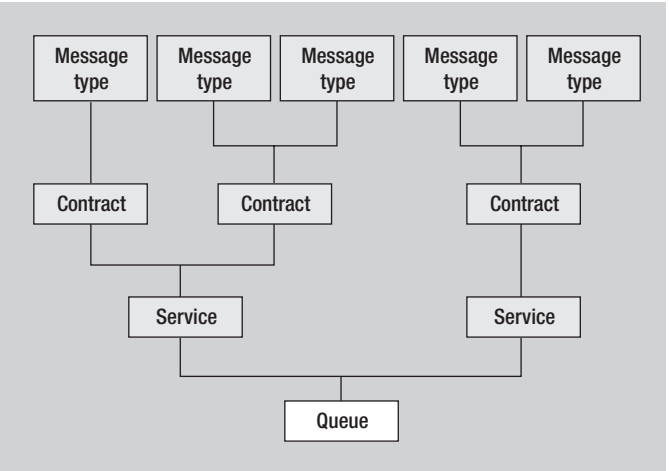
```
CREATE SERVICE [//class-a.nl/Training/
                              InschrijvingService]
  ON QUEUE InschrijvingQueue(
  [//class-a.nl/Training/ProcessInschrijving])
```

**Voorbeeldcode 6:** Het creëren van een service in T-SQL.




**Afbeelding 4:** Dialoog tussen twee applicaties.

Zoals weergegeven in voorbeeldcode 6 is er een T-SQL statement om een service te creëren. Bij het creëren van de service kunnen de contracten worden gedefinieerd die geldig zijn voor de specifieke service. Hiermee is vastgelegd welke berichten heen en weer gezonden kunnen worden via de service. De berichttypen zijn namelijk gedefinieerd als onderdeel van het contract.
De SSB infrastructuur zorgt voor referentiële integriteit. Het is daardoor niet mogelijk om een berichttype te verwijderen wanneer dit type onderdeel uitmaakt van een contract dat gebruikt wordt door een service.
De SSB infrastructuur biedt ook de mogelijkheid om een zogenaamde route te definiëren. Hiervoor zijn opnieuw T-SQL statements beschikbaar. Een route maakt het mogelijk om een service te definiëren op de ene machine, terwijl de verwerking plaatsvindt op een andere machine. De route bepaalt op welke machine de service de berichten ontvangt en waar de verwerking uiteindelijk plaats zal vinden. De SSB realiseert op deze manier distributie.
De relatie tussen de verschillende SSB meta-objecten is in afbeelding 5 weergegeven. Vanuit het oogpunt van de applicatie-programmeur wordt dus gecommuniceerd via services. Dit past uitstekend in de gedachtewereld van een SODA. Voorbeeldcode 7 laat een client programma zien in de vorm van een stored procedure die gebruikt maakt van SSB via services.




**Afbeelding 5:** Relatie tussen SSB meta-objecten.

```
CREATE PROCEDURE Inschrijving
@firstName varchar(25),
@lastName varchar(30),
@training varchar(30)
AS
BEGIN

  DECLARE @message NVARCHAR(MAX)
  SET @message = NCHAR(0xFEFF)
              + '<Customer>'
              + '<CustomerName>' + @firstName +
              ' ' + @lastName + '</CustomerName>'
              + '<Training>' + @training +
                                  '</Training>'
              + '</Customer>'

  DECLARE @conversationHandle UNIQUEIDENTIFIER

  BEGIN DIALOG CONVERSATION @conversationHandle
  FROM  SERVICE [//class-a.nl/Training/
                            InschrijvingService]
    TO  SERVICE '//class-a.nl/Training/
                           EmailResponseService'
    ON  CONTRACT [//class-a.nl/Training/
                           InschrijvingContract]
  WITH  ENCRYPTION = OFF

  ;SEND ON CONVERSATION @conversationHandle
  MESSAGE TYPE [//class-a.nl/Training/
                 InschrijvingTraining] (@message)
END
```

**Voorbeeldcode 7:** Client stored procedure.

In voorbeeldcode 7 is eerst bepaald welke service en welk contract wordt gebruikt. Daarna is een boodschap opgebouwd, die vervolgens via het SEND statement als een bericht naar de queue verzonden wordt. Er wordt in de voorbeeldcode geconfigureerd dat er geen encryptie wordt gebruikt. Het is meestal wel noodzakelijk om de berichten versleuteld binnen een conversatie te verzenden.

## Activation

Door gebruik te maken van 'activation' op een queue wordt het push-model van de SSB geïmplementeerd. Een stored procedure, als service-programma gekoppeld aan een queue, kan automatisch worden geactiveerd door de SSB infrastructuur op het moment dat er een bericht binnenkomt. Dit gebeurt doordat er speciale logica wordt uitgevoerd in het 'commit' proces. Bij het wegschrijven van de data in de queue wordt gecontroleerd of er een instantie van het service-programma draait. Deze gaat het bericht dan verwerken. Als er geen service-programma draait wordt een instantie gestart.

De activatielogica beoordeelt bovendien of het huidige aantal service-programma instanties voldoende capaciteit biedt om de hoeveelheid binnenkomende berichten te verwerken. Is dit niet het geval dan wordt een nieuwe instantie van de stored procedure gestart. Zo gauw de queue leeg raakt worden de stored procedures gestopt. Het maximale aantal instanties configureer je bij activatie. In voorbeeldcode 8 is geconfigureerd dat maximaal vijf instanties van de stored procedure worden gestart.

```
CREATE QUEUE InschrijvingQueueWithActivation
WITH STATUS = ON,
ACTIVATION (PROCEDURE_NAME = ProcessInschrijving,
            MAX_QUEUE_READERS = 5,
            EXECUTE AS SELF)
```

**Voorbeeldcode 8:** Queue activation.

De service-programma's draaien onder een specifieke identiteit. Op basis van deze identiteit zijn de rechten bepaald. De identiteit wordt gezet door het EXECUTE AS statement. In voorbeeldcode 8 is de identiteit gezet op SELF, wat neerkomt op de ingelogde gebruiker. In de praktijk is het goed om een specifieke database-gebruiker te configureren waaronder de service-programma's draaien.

## Conclusies

De SQL Service Broker is een messaging-oplossing die volledig geïmplementeerd is in Microsoft SQL Server. De SSB zorgt voor de mogelijkheid van asynchrone acties in de database. Langdurende transacties kunnen hierdoor eventueel worden opgedeeld in meerdere kleine transacties. Doordat de queues database-objecten zijn, wordt er ook geprofiteerd van database. Zo is het mogelijk om van de queues gelijktijdig een backup te maken met de normale database-tabellen. Daarnaast kunnen de queues gebruik maken van het standaard transactiemechanisme. Een SSB infrastructuur wordt gecreëerd met behulp van T-SQL. Er zijn diverse nieuwe T-SQL statements in Microsoft SQL Server 2005, waarmee de SSB objecten worden gemaakt en de services worden gebruikt.
De SSB is de kern van de Service Oriented Database Architectuur met SQL Server 2005. De SSB brengt queues en service in de database. Is de database de queue? Of is een queue een database?

**Astrid Hackenberg** en **Anko Duizer** zijn beiden werkzaam bij Class-A als trainer/coach en medeoprichter. Class-A is een Microsoft kenniscentrum.

## Meer weten?

http://research.microsoft.com/~Gray/QueueIsDB.doc
www.pathelland.com
www.ankoduizer.nl
www.class-a.nl