IT Dienstleistungen GmbH

Oracle Advanced Queuing AQ

13.09.2012

Referenten:

Claus Cullmann

Andreas Steinel

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

eXirius
IT Dienstleistungen GmbH

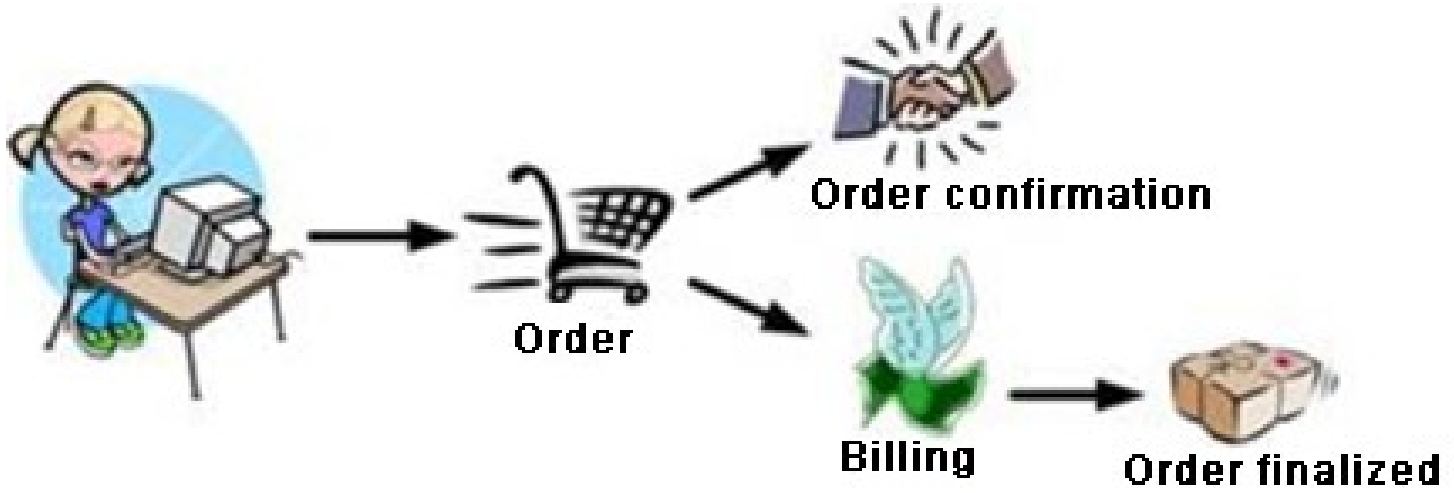
Inhalt

- Motivation
- Message Systeme
- Eigenschaften, Beispiele Oracle AQ
- Terminologie AQ
- Beispiel „pure SQL“
- Beispiel Java-Anwendung
- Beispiel Ruby-Anwendung
- Beispiel Multiconsumer mit Transformation

eXirius IT Dienstleistungen GmbH

Motivation AQ

Hier im Beispiel wird über einen Onlineshop eine Bestellung generiert. Der Kunde erhält sofort eine Bestätigung, während der komplexe Prozess der Bestellabwicklung parallel und asynchron abgewickelt wird.



Quelle http://www.theserverside.de/jms-mit-oracle-advanced-queueing/

eXirius IT Dienstleistungen GmbH

Message-Systeme

Software-Lösungen, die sich mit Mitteilungen beschäftigen, nennt man Message Oriented Middleware (MOM)

MOM bezeichnet eine Klasse von Softwareprodukten, die als Kommunikationsinfrastruktur in Anwendungslandschaften eingesetzt wird. MOM-Produkte dienen der sicheren, robusten und skalierbaren Übertragung von Nachrichten zwischen verteilten Diensten.

Vertreter sind z.B.

- MMQ Microsoft Message Queuing
- JMS Java Message Service
  Diese bedienen sich den Queue-Diensten. Solche Server nennt man dann JMS-Provider.

eXirius IT Dienstleistungen GmbH

JMS-Middleware

Name	Firma	Lizenz
ActiveMQ	Apache	Open Source (Apache 2)
ApolloMQ	Apache	Open Source (Apache 2)
FioranoMQ	Fiorano	kommerziell
iBus//MessageServer	Softwired	kommerziell
HornetQ (ehemals "JBoss Messaging")	Red Hat	Open Source (LGPL)
JORAM	OW2	Open Source (LGPL)
MantaRay	Coridan	Open Source (Mozilla Public License)
Mom4j	Mom4j development team	Open Source (LGPL)
MRG Messaging	Red Hat	kommerziell
MuleMQ	MuleSoft Inc.	kommerziell
OpenJMS		Open Source
Open Message Queue	Sun Microsystems	Open Source
Oracle Advanced Queuing (OAQ)	Oracle	kommerziell
Qpid	Apache	Open Source (Apache 2)
RabbitMQ	VMware, Inc.	Open Source (Mozilla Public License)
SAP JMS	SAP	kommerziell
SonicMQ	Progress Software	kommerziell
Sun GlassFish Message Queue	Sun Microsystems	Open Source
SwiftMQ Router	IIT Software	kommerziell
TIBCO Enterprise Message Service	TIBCO	kommerziell
webMethods Broker	Software AG	kommerziell
Websphere MQ	IBM	kommerziell
ØMQ oder auch ZeroMQ oder auch 0MQ	iMatix Corporation	Open Source (LGPL)

Quelle: http://de.wikipedia.org/wiki/Java_Message_Service

Oracle AQ

Historie

- Ab 8.1 AQ verfügbar
- Ab 9.2. AQ in Oracle Standard Edition
- Ab 10.1. AQ integriert in Oracle Streams „Oracle Streams AQ“
- Ab 11.2. AQ dient als Basis für den synchronen Capture-Prozess in der Tabellen-Replikation der Standard Edition

eXirius IT Dienstleistungen GmbH

Beispiele Oracle AQ

Middleware:

- Oracle AQ wird genutzt im JMS Java Message Service im Oracle Enterprise Service Bus

Datenbank: Oracle AQ wird genutzt in der Prozess-Kommunikation

- Oracle Data-Pump
- Oracle Streams
- Oracle Scheduler
- Oracle Cloud Control (Agents und Management Server)
- Oracle Statistik-Generierung
- Oracle Workflow Manager
- Weitere...

eXirius IT Dienstleistungen GmbH

Oracle AQ

Keyfeatures

- Point-to-point
- publish/subscribe-Technologie
- Queue to Queue Abbildung

samt Monitoring, Security Features, Transformation

Message Schnittstellen

AQ kann nahtlos in existierende Anwendungen eingebunden werden, da es Schnittstellen in verschiedenen Programmier-Sprachen gibt:

Es gibt Schnittstellen zu

- PL/SQL
- C/C++
- Java
- Visual Basic (Oracle Objects for OLE)
- Java Message Service (oracle.jms Java package)
- HTTP, HTTPS, und SMTP

eXirius
IT Dienstleistungen GmbH

Terminologie AQ

Message

Messages sind die Mitteilungen der Anwendungen. Sie beinhalten den Datenstrom in Form eines Objekt-Typs. Messages haben zusätzlich Metadaten wie Adressat, Priorität, Zeitfenster usw .

Es gibt *persistent* und *buffered* Messages.

eXirius
IT Dienstleistungen GmbH

Terminologie AQ

Queue (Nachrichtenwarteschlange):

- Messages (Mitteilungen) werden in Queues abgelegt
- Queues sind so etwas wie Mailboxen
- Verschiedene Anwendungen können in der Queue nachschauen, ob „Post“ für sie da ist.

Queue Tables :

- Queues werden in Queue Tables gespeichert

eXirius IT Dienstleistungen GmbH

Terminologie AQ

Agents :

Ein *Agent* ist ein Queue User. Es gibt 2 Arten von Agenten:

- *Producer*:
  Der Producer erzeugt die Message und stellt sie in die Queue ein.
  Diesen Vorgang nennt man "Enqueue".

- *Consumer*:
  Der Consumer holt die Message aus der Queue ab.
  Diesen Vorgang nennt man "Dequeue".

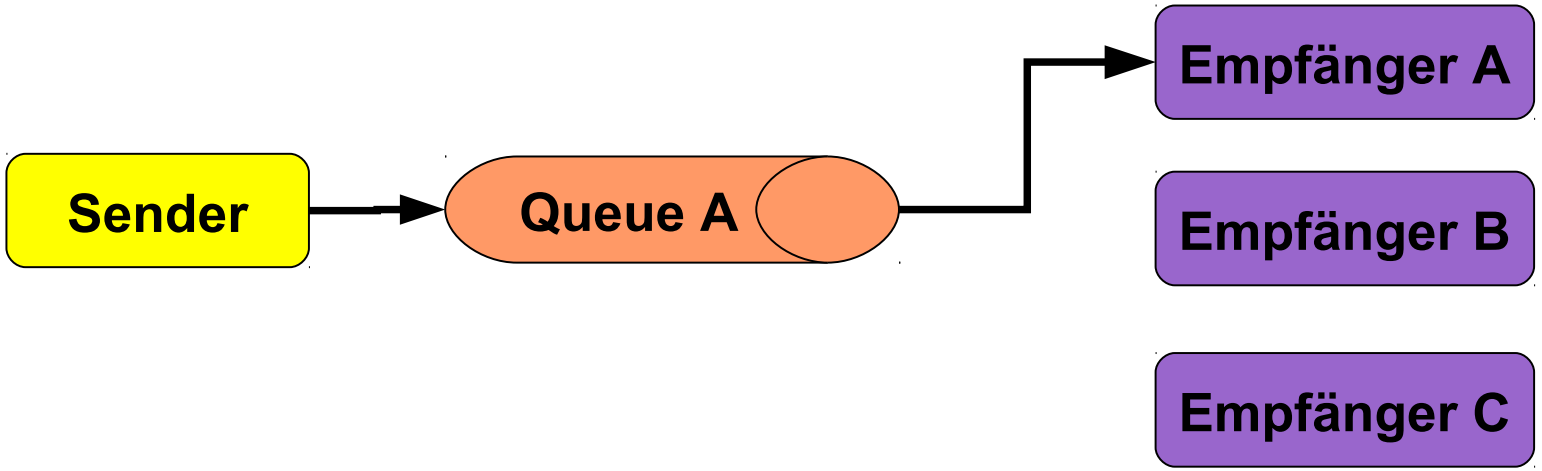
Time Manager :

Der Time Manager ist ein Hintergrund-Prozess, der die Queues überwacht. Er stellt Mechanismen bereit, um eine Message zu "veralten", wiederherzustellen oder zu löschen. Der Prozess (Queue Monitor Coordinator) wird mittels Init-Parameter *aq_tm_processes* gestartet.

Queue Modelle

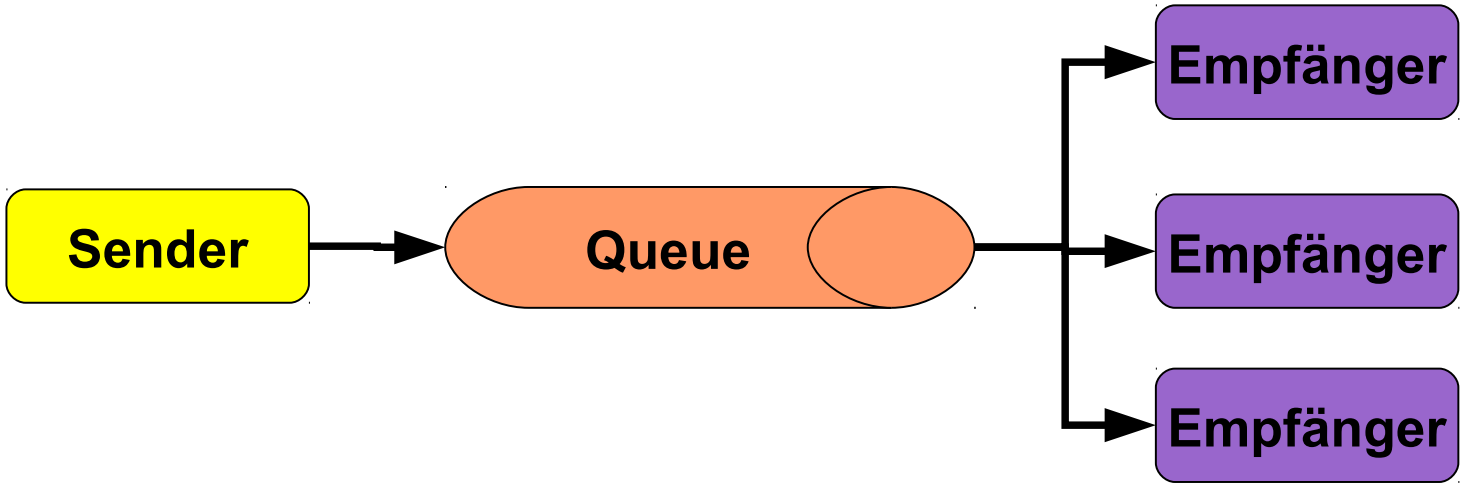
Es gibt 2 Arten von Queue Modellen.

1. Point-to-Point:
Eine *Point-to-Point*- oder *Single Consumer*-Queue hat ein spezifisches Ziel: Producer und Consumer haben eine einheitliche Queue. Eine Message kann nur einmal dequeued werden.



Queue Modelle

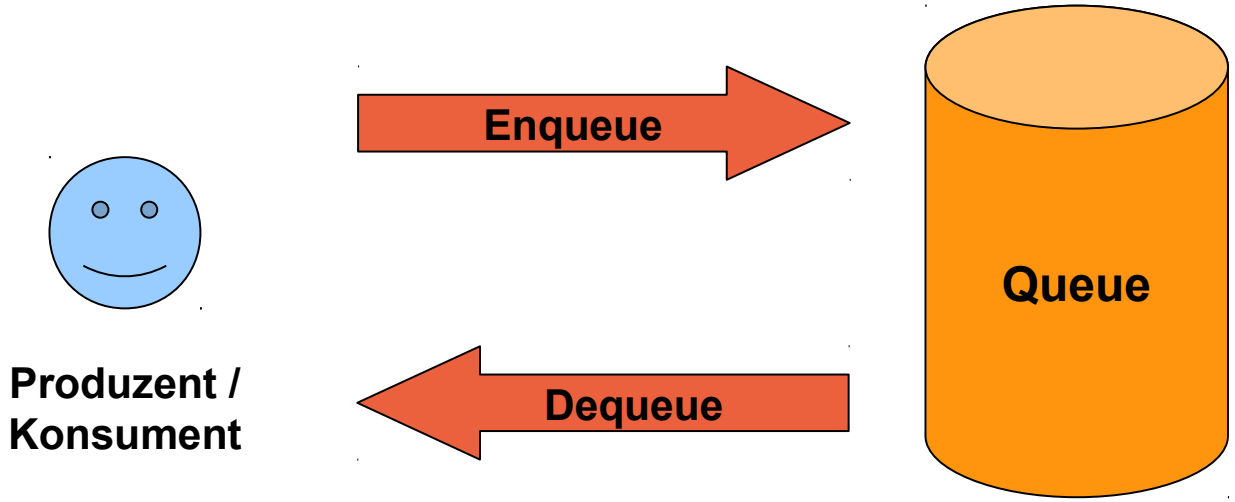
2. *Publish/Subscribe*: (subscribe: abonnieren)
Diese Queue kann zur Ausstrahlung von Mitteilungen an viele Empfänger dienen. Es gibt unbekannt viele Consumer. Diese können via OCI auch über neue Mitteilungen informiert werden.



In diesem Modell bleibt eine Mitteilung solange aktiv, bis entweder das eingestellte Timeout erreicht wird oder alle registrierten Empfänger die Nachricht abgefragt haben.

Beispiel simples PL-SQL

Hier in diesem Beispiel ist der Produzent gleich dem Konsument und es wird eine Nachricht eingestellt und abgerufen.

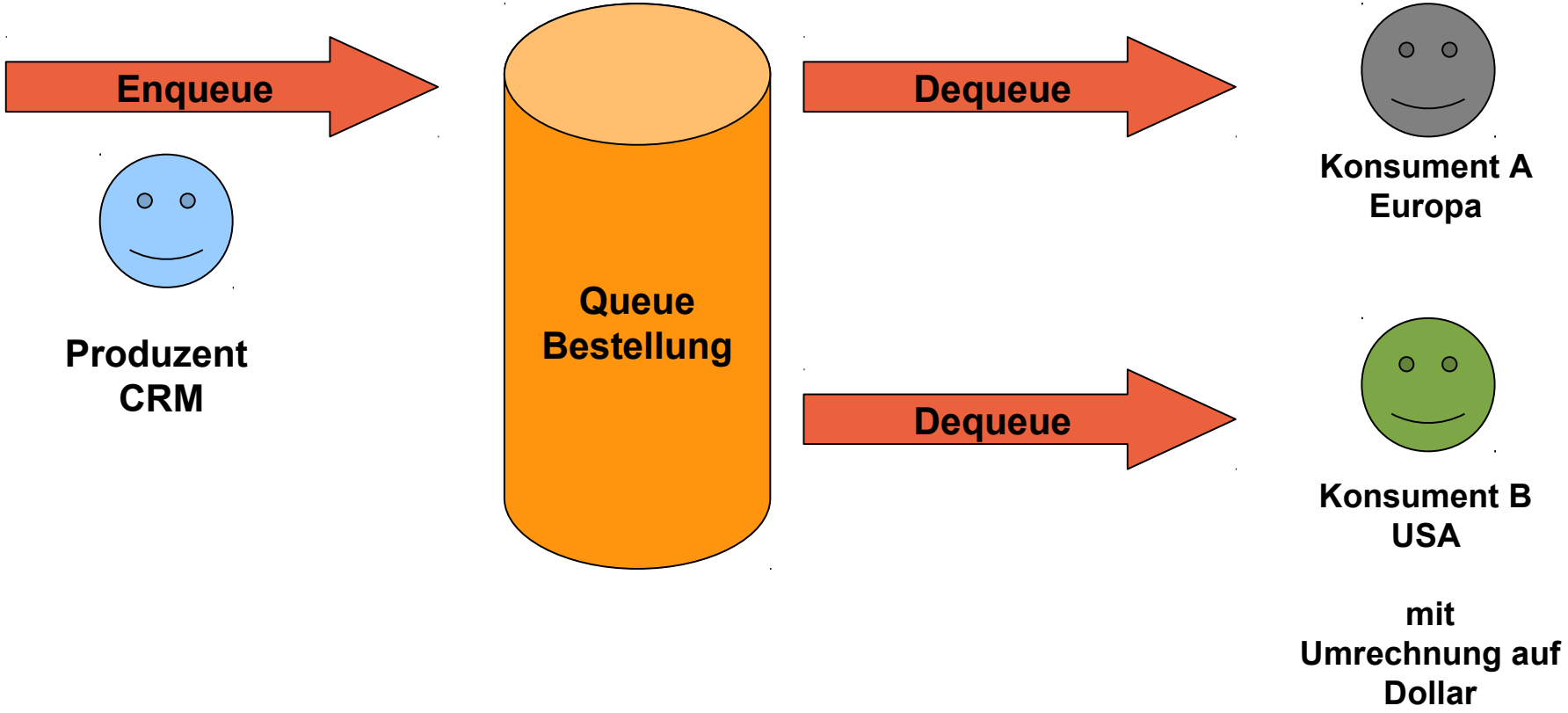


eXirius
IT Dienstleistungen GmbH

Live Beispiele mit Java und Ruby

Beispiel komplexeres PL-SQL

Hier sind 3 unterschiedliche Anwender beteiligt:
Es gibt 2 Subscriber und als Transformation wird bei der Abrechnung von Euro in Dollar umgerechnet.



eXirius IT Dienstleistungen GmbH

Monitoring AQ

Bei der Anlage einer Queue werden automatisch AQ-Views angelegt:

```
SQL> select view_name from user_views where view_name like 'AQ%';

VIEW_NAME
------------------------------
AQ$QT_BESTELLUNG
AQ$QT_BESTELLUNG_R
AQ$QT_BESTELLUNG_S
AQ$_QT_BESTELLUNG_F
```

AQ$<Queue Table Name>	Messages in Queue Table
AQ$<Queue Table Name>_S	Subscribers
AQ$<Queue Table Name>_R	Subscribers mit Regeln

Vielen Dank für Ihre Aufmerksamkeit