Die Versorgung eines Data Warehouse (DWH) mit frischen Daten kann zuweilen eine große Herausforderung sein. Es sind in der Regel nicht nur ein DWH-System, sondern zumeist mehrere Systeme wie die Entwicklungs-, Test-, Integrations-, Wartungs- und Produktions-Umgebungen zu bedienen.

Global Staging Area: Implementierung einer zentralen Daten-Drehscheibe

Sven Bosinger, its-people GmbH

Jede der zu versorgenden Umgebungen hat spezielle Anforderungen. Gleichzeitig soll auf der Datenlieferanten-Seite die Anzahl der Schnittstellen, über die Daten abgegeben werden, überschaubar bleiben. Zusätzlich spielen regulatorische und rechtliche Vorgaben eine Rolle. Entwickler dürfen immer häufiger keinen Zugang mehr zu personalisierten Daten erhalten, sondern müssen auf maskierten und verfremdeten Daten entwickeln.

In der hier dargelegten Lösung geht es um die Implementierung einer zentralen Datendrehscheibe, die sogenannte „Global Staging Area" (GSA), die aus verschiedensten Quellsystemen mit Daten bestückt wird. Sie gibt wiederum die gepufferten Daten an die diversen DWH-Systeme gezielt weiter. Dadurch wird in jedem Quellsystem nur noch eine Schnittstelle benötigt, die damit Datenlieferant für alle nachgelagerten DWH-Systeme ist. In der GSA wird nach einem vorgegebenen Regelwerk entschieden, wann, wie und in welcher Form die Daten an die DWH-Systeme weitergegeben werden. So lässt sich ein permanenter Datenstrom mit allen Echtdaten an die Produktionsumgebung einrichten, wohingegen die Entwicklungsumgebung mit einem reduzierten und verfremdeten Datenbestand versorgt wird. Neue Quellsysteme können einfach über Oracle-Standard-Technologien (Streams, CDC, AQ, Trigger etc.) an die GSA angebunden werden.

Ausgangslage

Viele Anwender einer DWH-Lösung haben sich für einen klassischen Aufbau ihres DWH entschieden (siehe Abbildung 1). Dabei werden die Daten der Quellsysteme in einer Staging Area gesammelt, durch einen Batch-Lauf in ein zentrales Enterprise-Modell integriert und abschließend Business-Area-spezifische Data Marts aufgebaut.

In der Regel betreibt ein Anwender aber nicht nur eine DWH-Instanz, sondern mehrere. Je nach Vorgehen werden neben der Produktion noch weitere Instanzen für Entwicklung, Test, Abnahme und Wartung benötigt. Dies bedeutet, dass nicht nur eine Instanz permanent mit Daten aus den Quellsystemen versorgt werden muss, sondern viele. Ausgehend davon kommt man zu folgenden Datenanforderungen:

- *Produktions-Instanz*
  Regelmäßige Belieferung mit vollumfänglichem, unverfälschtem Datenbestand
- *Wartungs-Instanz*
  Regelmäßige Belieferung mit vollumfänglichem, unverfälschtem Datenbestand, um Fehler in der Produktion nachstellen zu können
- *Abnahme-Instanz*
  Regelmäßige Belieferung mit gegebenenfalls eingeschränktem und maskiertem Datenbestand, um Abnahmetests durchzuführen
- *Test-Instanz*
  Bedarfsgesteuerte Belieferung mit eingeschränktem und maskiertem (Test-) Datenbestand, um Tests durchzuführen
- *Entwicklungs-Instanz*
  Bedarfsgesteuerte Belieferung mit eingeschränktem und maskiertem (Test-) Datenbestand, um zu entwickeln

Bei einer DWH-Entwicklung besteht zudem die Besonderheit, dass eine erfolgreiche Entwicklung nur auf produktionsnahen Echtdaten und nicht auf

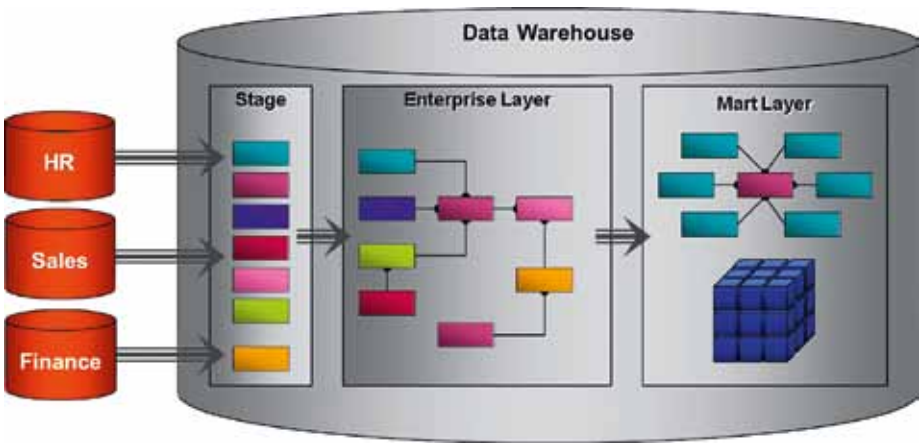

Abbildung 1: Das klassische DWH

Testdaten möglich ist. Jegliche DWH-Entwicklung ist eine Daten-getriebene Entwicklung. Fragen der Performance, statistische Auswertungen und Variationsvielfalt sind durch eingeschränkte Testdaten in der Regel nicht zu beantworten. Es ist also häufig notwendig, schon in den Entwicklungs-Instanzen mit Produktionsdaten zu arbeiten. Spätestens in der Abnahmeumgebung muss auf Produktionsdaten gearbeitet werden, um abnahmefähige Testfälle generieren zu können. Daher ergibt sich die Problematik, dass die produktiven Quellsysteme nicht nur mit dem Produktions-DWH über Schnittstellen verbunden werden müssen, sondern auch mit den übrigen DWH-Systemen. Dies führt zu einem überproportionalen Anwachsen der Anzahl der Schnittstellen (siehe Abbildung 2).

Aufgrund von Datensicherheitsaspekten dürfen häufig sensible Produktionsdaten, wie Kreditkarten-Informationen oder Bankdaten, nicht in eine ungeschützte Entwicklungsumgebung gelangen. Vor allem nicht, wenn bei der Entwicklung Near- oder Offshore-Kräfte eingesetzt werden sollen. Hier müssen Daten gegebenenfalls verfremdet oder ausgeblendet sein. Darüber hinaus bestehen branchenabhängige rechtliche Vorgaben, die eine Modifikation der Daten zwingend erforderlich machen. Diese Anforderungen müssen bei einem klassischen DWH-Ansatz mit lokalen Staging Areas in der jeweiligen Schnittstelle realisiert werden.

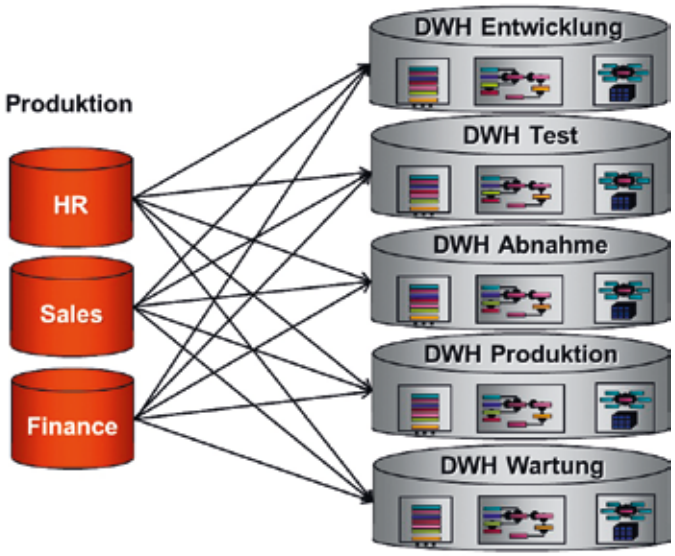

Abbildung 2: Schnittstellen-Explosion

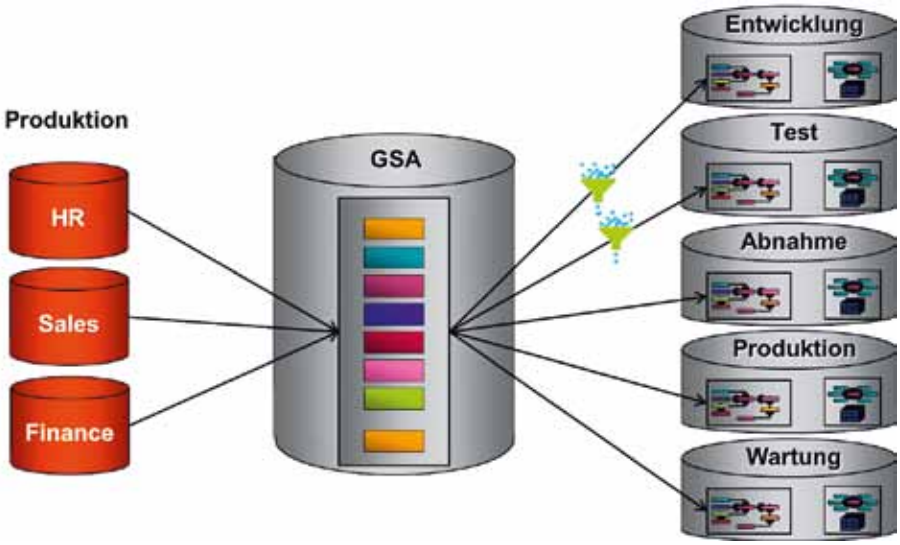

Abbildung 3: Global Staging Area

Global Staging Area

Das Verfahren der Global Staging Area (GSA) ersetzt alle lokalen Staging Areas in den einzelnen DWH-Instanzen (siehe Abbildung 3). Alle DWH-Instanzen verarbeiten die Stage-Daten weiterhin im Rahmen eines klassischen ETL-Prozesses. Insofern wird auf die GSA zugegriffen, als ob es sich um eine klassische, lokale Staging Area handeln würde. Die Quellsysteme werden ausschließlich über Schnittstellen an die GSA angebunden (siehe Abbildung 4). Daher muss für jedes Quellsystem nur noch eine Schnittstelle definiert werden, egal, wie viele DWH-Instanzen bedient werden müssen. Dabei kommt ausnahmslos ein Push-Verfahren zum

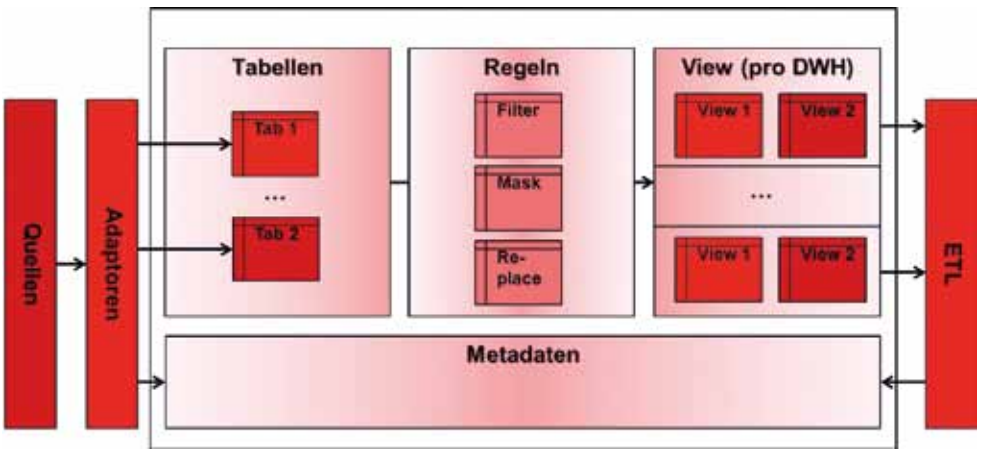

Abbildung 4: Interner Aufbau der GSA

Einsatz, die Quellsysteme stellen also die Datenlieferungen zusammen und übertragen diese direkt in die GSA. Dabei werden die technischen Verfahren „Change Data Capture“, „Advanced Replication“, „Advanced Queuing“, „Streams“ und „Trigger“ (via Database Link) unterstützt.

Die Quellsysteme liefern die Produktionsdaten umfänglich, die Daten werden also so geliefert, wie sie im Produktions-DWH erforderlich sind. Etwaige Filterungen, Daten-Maskierungen oder Verfremdungen erfolgen innerhalb der GSA. Diese Aktivitäten sind DWH-Instanz-spezifisch, die Produktionsumgebung erhält somit die Daten ungefiltert, wohingegen die Daten für die Test-Umgebung gefiltert und Konto-Informationen maskiert werden können. Die Bereitstellung der Daten für die nachgelagerten ETL-Prozesse ist von der GSA durch entsprechende Instanz-spezifische Sichten auf die Stage-Daten gewährleistet. Während des Aufbaus dieser Sichten werden dabei metadatengesteuert die entsprechenden Filterregeln, Maskierungen und Verfremdungen angewandt. Jede DWH-Instanz erhält nur Zugriff auf seine Sichten.

Prozesse

Der Betrieb der GSA erfolgt prozessgesteuert. In der GSA werden hierzu drei (optional vier) Prozesse betrieben:

- *Push in die GSA*
  Die Quellsysteme liefern die Daten in Real/Near Time in die GSA. Dort werden sie ungeprüft in die entsprechenden Stage-Tabellen eingefügt. Jeder Datensatz wird mit einer systemweiten, eindeutigen DWH-ID und einem Liefer-Datum versehen. Zusätzlich wird im Metadaten-Katalog der Status jedes einzelnen Datensatzes DWH-Instanz-bezogen festgehalten. Das Einfügen der Daten in die GSA ist transaktionsgesichert (siehe Abbildung 5).
- *Pull aus den DWHs*
  Die Verarbeitung der Daten aus der GSA in der jeweiligen DWH-Instanz erfolgt mithilfe eines klassischen ETL-Prozesses. Die zu verarbeitenden Sätze werden aus Instanz-spezifischen Views gelesen. Ebenso wie die erfolgreiche Verarbeitung werden Fehler im Metadatenkatalog

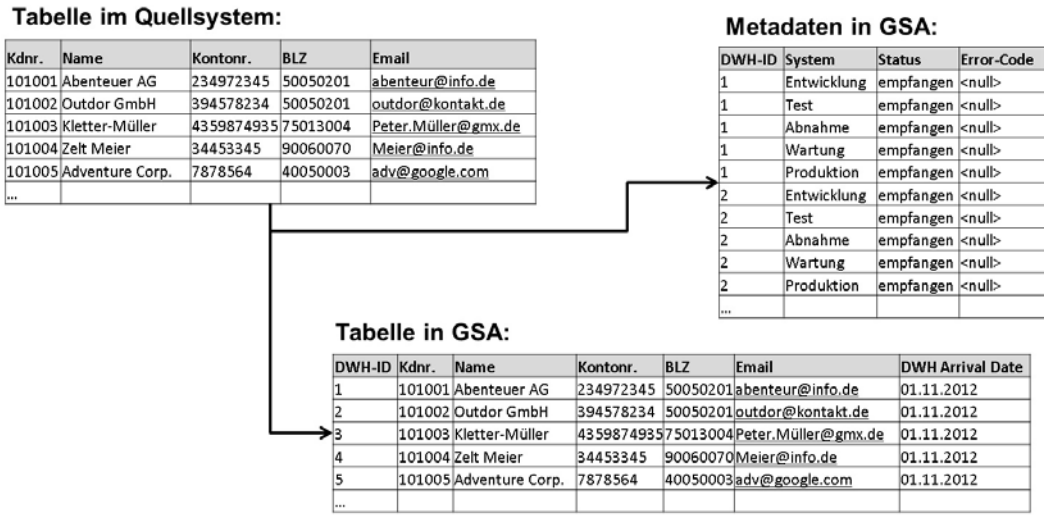

Abbildung 5: Push in die GSA

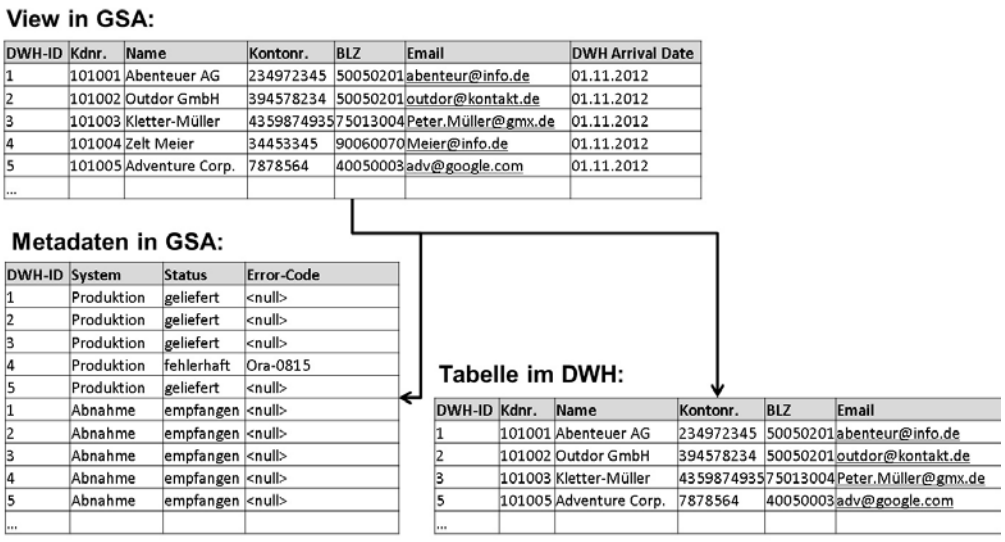

Abbildung 6: Pull aus dem DWH

protokolliert. Die Views werden pro Instanz unter Berücksichtigung der Instanz-spezifischen Filterregeln, Maskierungen und Verfremdungen erzeugt. Der Pull pro Instanz kann völlig unabhängig von allen anderen Instanzen betrieben werden. Unterschiedliche Ladezeitpunkte und Frequenzen sind problemlos realisierbar (siehe Abbildung 6).

- *CleanUp der SGA*
  Alle erfolgreich in die DWH-Instanzen geladenen Daten, erkenntlich über den Status im Metadatenkatalog, werden regelmäßig asynchron gelöscht. Dazu kann eine Vorhaltezeit definiert werden, sodass die Daten zu Nachverfolgungszwecken einen definierten Zeitraum in der GSA verbleiben (siehe Abbildung 7).
- *Real-/Near-Time-Auswertungen (optional)*
  Solange die Daten in der GSA stehen, können sie zusätzlich noch zu Auswertungszwecken genutzt werden. Sobald sie aus der GSA gelöscht wurden, stehen sie im DWH bereit. Real-/Near-Time-Auswertungen können somit auf der SGA einfach realisiert werden.

Performance

Beim Betrieb einer GSA wird mit Massendaten gearbeitet. Daher ist es notwendig, sich beim Design der GSA über die damit verbundenen Performance-Aspekte Gedanken zu machen. Nachfolgend ein paar Hinweise, ohne den Anspruch auf Vollständigkeit:

- *Push in die GSA*
  Der Upload der Daten soll aus den Quellsystemen initiiert werden. Dabei sollten nach Möglichkeit die Change-Data-Capture-Verfahren zum Einsatz kommen. Der Vorteil dieser Technologie besteht darin, dass es zu keiner zusätzlichen Belastung der Quellsysteme kommt. Die Redo Logs der Quellsysteme können hierbei zudem asynchron auf der GSA-Instanz ausgewertet werden.
- *Pull aus den DWHs*
  Die einzelnen DWH-Instanzen verarbeiten die GSA-Daten in einem Batch. Es wird also eine Mengen- und keine Einzelsatzverarbeitung durchgeführt. Das Protokollieren der verarbeiteten Sätze in den Metadaten-Tabellen der GSA erfolgt ebenfalls über Massen-Updates. Durch eine geeignete Partitionierung der zugrunde liegenden Tabellen können alle später zu löschenden Daten gezielt in die entsprechenden Partitionen gelegt werden.
- *CleanUp der SGA*
  Das Löschen von Daten ist in der Regel ein sehr aufwändiger Prozess. Daher sollte ein klassisches Löschen durch ein Delete-Statement vermieden werden. Effektiver ist es, alle zu löschenden Datensätze in Partitionen vorzuhalten und dann die gesamten Partitionen zu entfernen. Dies verhindert auch die Fragmentierung der Daten-Tabellen in der GSA.
- *Architektur*
  Es hat sich bewährt, die GSA in einer eigenen Datenbank-Instanz zu betreiben. Die Daten-Tabellen der GSA sollten partitioniert sein. Als Kriterium dafür kann das Arrival-Datum dienen. Es werden Tages-Partitionen gebildet. Der Vorteil ist, dass später nach der gewünschten Aufbewahrungsfrist die gesamte Tages-Partition komplett entfernt werden kann. Dazu müssen gegebenenfalls vorher die noch fehlerhaften Datensätze, die in der GSA verbleiben sollen, in eine nicht zu löschende Fehlerpartition ausgelagert werden.

Fazit

Wie dargelegt, bietet der Einsatz einer GSA eine Reihe von Vorteilen:

- Die Anzahl der Schnittstellen in den Quellsystemen wird massiv reduziert. Pro Quellsystem ist lediglich eine Schnittstelle notwendig, um beliebig viele DWH-Instanzen mit Daten zu versorgen. Dadurch sinkt die Anzahl der Schnittstellen.
- In den Schnittstellen der einzelnen Liefersysteme wird keine zusätzliche

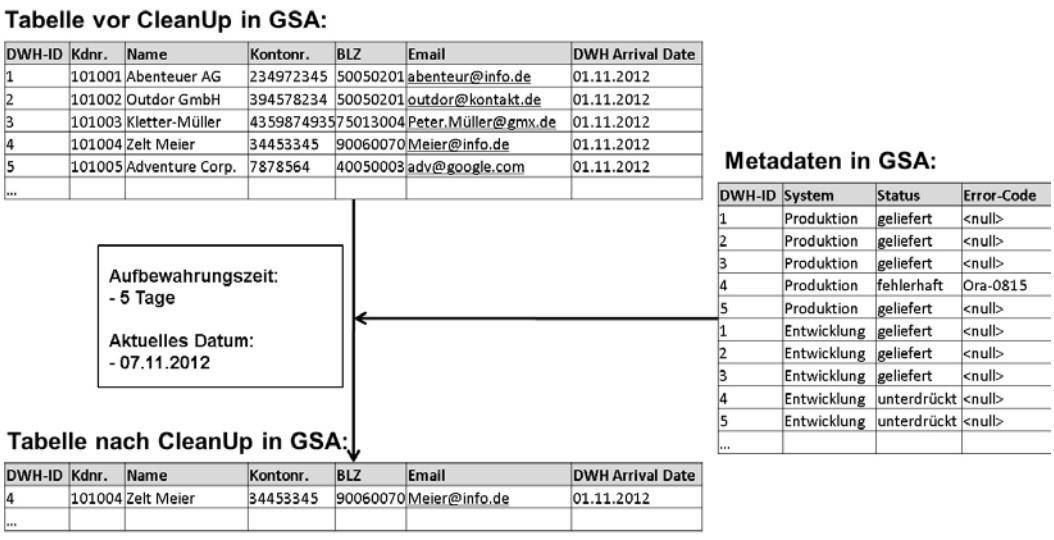

Abbildung 7: CleanUp der SGA

Logik benötigt; Verfremdung und Filterung der Daten wird bei Bedarf in der GSA durchgeführt. Dadurch wird die Komplexität der Schnittstellen reduziert.

- Alle DWH-Instanzen sind permanent über ein Push-Verfahren mit aktuellen Echtdaten versorgt. Dadurch kann bei der Entwicklung schon auf realistischen Datenmengen gearbeitet werden. Darüber hinaus sind alle vorkommenden Daten-Konstellationen berücksichtigt.
- Die Datenmenge in der GSA ist deutlich geringer als die Datenmenge aller lokalen Staging Areas zusammen. Jeder Datensatz wird nur einmal gespeichert und kann an beliebig viele DWH-Instanzen verteilt werden. Die Sichten sind lediglich logischer Natur und als Datenbank-Views aufgeteilt auf ein Schema pro DWH-Instanz.
- Die Datenmengen lassen sich bei Bedarf für einzelne Instanzen einfach durch konfigurierbare Filterregeln reduzieren.
- Sicherheitsrichtlinien werden durch Metadatenkonfiguration einfach und transparent umgesetzt, so können Maskierung und Verfremdung der Daten Instanz-abhängig eingestellt werden.
- Das komplette Verfahren ist transaktionsgesichert, es können damit keine Datensätze verloren gehen. Kommt es zu Abbrüchen oder Lieferausfällen, so werden keine unvollständigen Lieferungen gespeichert und damit auch nicht im ETL-Prozess verarbeitet.
- Durch das Push-Verfahren sind die Daten in Real/Near Time in der GSA gespeichert, was beispielsweise ein Real-/Near-Time-Reporting einfach macht.
- Zusätzliche DWH-Instanzen, die eventuell sogar nur temporär benötigt werden, sind einfach und schnell mit Daten zu versorgen. Lediglich die Instanz-bezogenen Views sind einmalig zu erzeugen.

Größere Nachteile waren beim Einsatz einer GSA bisher nicht feststellbar. Die Implementierung ist bei Vorliegen der beschriebenen Vorbedingungen (mehrere DWH-Instanzen greifen auf die gleichen Quellesysteme zu) uneingeschränkt zu empfehlen.

Sven Bosinger
sven.bosinger@
its-people.de