

# Enterprise Database – Forgotten, or not?

Jürgen Bittner

SQL Projekt AG, Franklinstraße 25A, 01069 Dresden, Deutschland  
Juergen.bittner@sql-ag.de

**Abstract.** With the appearance of the Database Technology the target of an enterprise wide application was joined, respective the integrated management of all relevant data of the enterprise within one database, because on this way a maximal usefulness of the new technology seemed achievable. Several preconditions have been avoided this target's achievement. The current self-evident utilization of the database technology is referring predominantly to individual areas or departments of the organizations. But the requirements to support processes and analytics exceeding isolated systems were advancing permanently. Beside the further development of the database technology this led to integration solutions of different levels. They are developed and operated often with large time and effort. The Data Warehouse approach oriented to data consolidation for the purpose of reporting and analytics, but without a possibility to improve the underlying operational processes. The integration technology supports database crossing processes, can speed up these and reduce their cost, but has only limited influence on data quality. The objectives of data quality in connection with the integration technology as they are considered in the Master Data Management get the data consolidation in the focus again, but now, if the realization is consequent, with the requirement to provide the operational systems with improved data. Could this become the Enterprise Database? For it the software industry would need a technology conversion and above all a mindset change.

**Keywords:** Enterprise Database, Data Warehouse, Data Quality, Master Data Management.

## 1 Die ursprüngliche Vision

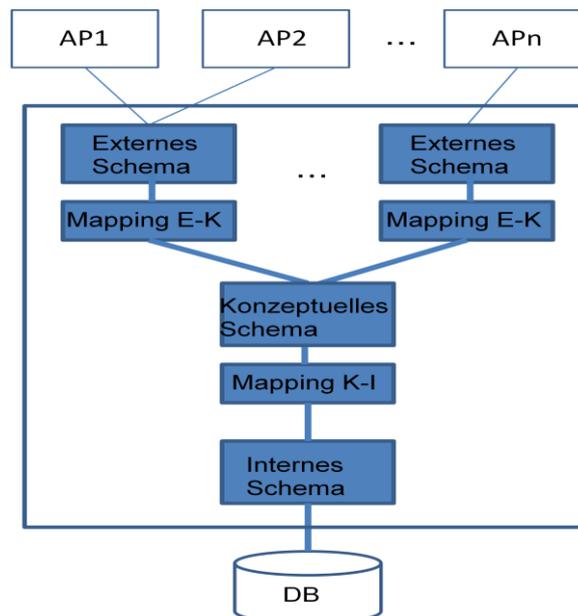
Das grundlegenden Ziele der Verarbeitung von Daten, alle Sachverhalte einmalig zu erfassen, die Daten möglichst einmalig zu speichern mit all ihren relevanten Beziehungen und die vielfache Verwendung diese Daten zu ermöglichen führten Ende der 60er Jahre zur Technologie der Datenbank. Korrektheit, Aktualität und Konsistenz der Daten sollten helfen, die Wirksamkeit gesellschaftlicher Prozesse, vor allem in der Betriebswirtschaft und Produktion zu verbessern. Die technologischen Komponenten zum Speichern von Daten und Beziehungen in einigen der ersten DBMS-Produkte waren bereits weitgehend geeignet, die vernetzte Datenstruktur eines Unternehmens abzubilden. Die Ziele der Datenbank-Technologie wurden deshalb von ihren Befürwortern fast von Anfang an auf eine unternehmensweite Wirkung projiziert. Die Enterprise-Datenbank war das anerkannte Ziel.

Das in den 70er Jahren bevorzugte Streben nach Integrierter EDV hat in zahlreichen Beiträgen und Konzepten für IMIS (Integrated Management Information Systems) die Datenbank-Technologie allerdings nur wenig berücksichtigt.

Die stark gewachsene und weiter wachsende Anwendung von Datenbanksystemen nahm auch einen anderen Verlauf. Zahlreiche Softwaresysteme für Fachgebiete oder Abteilungen besitzen Datenbanken, die Daten der gleichen Objekte enthalten, jedoch in unterschiedlicher syntaktischer und semantischer Ausprägung und mit unterschiedlichen Teilmengen des jeweiligen Objekttyps. Sie werden entwickelt für verschiedene Anforderungen und Benutzergruppen, von verschiedenen Entwicklern, zu verschiedenen Zeiten und für unterschiedliche Standorte, so dass ein einheitliches Verständnis und eine einheitliche Darstellung der Daten, von wenigen Standards abgesehen, nicht erreicht wurde. Mehrfacherfassung, Inkonsistenzen und ungenügende Aktualität mindern die Effizienz der Prozesse, teils deutlich erkennbar, häufig auch verdeckt.

Natürlich haben auch Hardware-Ressourcen und Verfügbarkeit die Chancen für große Enterprise-Datenbanken beschränkt. Es bleibt aber auch festzustellen, dass besonders während der ersten beiden DB-Jahrzehnte die Unterstützung des Entwerfens von Datenbanken mit der Technologieentwicklung der DBMS nicht Schritt gehalten hat. Das betrifft einerseits die methodischen Grundlagen, z.B. auch den Aspekt kooperierenden Entwerfens als auch besonders die Verfügbarkeit von Werkzeugen. Aus dieser Sicht war die Zielstellung einer Enterprise-Datenbank nicht realistisch.

Ein grundlegendes Konzept, dass die Zielstellung einer Enterprise-Datenbank wenigstens begünstigt hätte, ist die 3-Ebenen-Architektur nach ANSI/X3/SPARC [1].



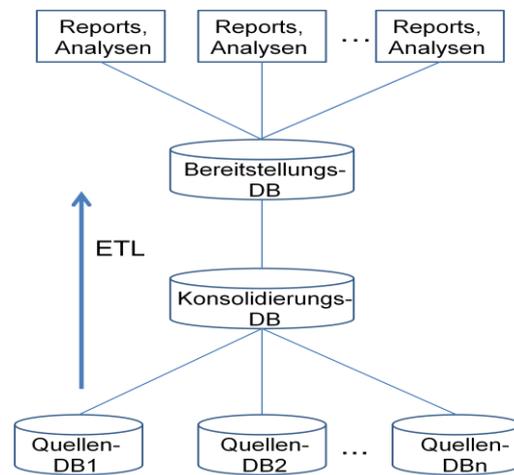
**Bild 1.** ANSI/X3/SPARC 1975: 3-Ebenen-Architektur des Datenbanksystems

Die primären Zielaspekte dieser Architektur sind Datenunabhängigkeit und Unterstützung des Datenbank-Entwurfs durch Trennung der Inhaltsaspekte von den Realisierungsstrukturen einerseits und den Verwendungsstrukturen andererseits. Damit

ergibt sich gleichzeitig ein Rahmen für die Vereinbarkeit dezentraler Sichten mit dem Ziel der Enterprise-Datenbank [2][3], was möglicherweise für künftige Anforderungen wieder ins Blickfeld kommen könnte.

## 2 Enterprise Information erfordert Konsolidierung

Der Bedarf nach Reports und Analysen, die über die Inhalte einzelner Datenbanken hinausgehen, führte in den 90er Jahren zur Data-Warehouse-Konzeption und auch entsprechenden Produkten.



**Bild 2.** Data-Warehouse-Architektur

Innerhalb des ETL-Prozesses ist hierbei die syntaktische und semantische Konsolidierung der verschiedenen Quellen-Datenbanken erforderlich (vgl. z.B. [4]). Auf der Schema-Ebene ist das ein Entwurfsprozess, der ein normalisiertes Schema für die Konsolidierungs-Datenbank liefern muss sowie die Transformationsregeln für die Überführung der Daten aus den Quellen in die Konsolidierungs-Datenbank. Bei entsprechendem Umfang der einbezogenen Quellen ist damit strukturell eine Enterprise-Datenbank gegeben.

Das Entwerfen einer solchen als „sekundär“ zu bezeichnenden Enterprise-Datenbank profitiert von der Tatsache, dass mit den Quellsystemen bereits umfangreiches Wissen gesammelt wurde. Andererseits wirkt erschwerend, dass Divergenzen und Inkompatibilitäten in der Menge der unabhängig voneinander entstandenen Quellen erkannt und durch Transformationen bereinigt werden müssen.

Vergleichend mit der Idealvorstellung einer „echten“ oder „primären“ Enterprise-Datenbank bleibt festzustellen:

Die Konsolidierungs-Datenbank ist kontinuierlich oder periodisch mit den Änderungen der Quellsysteme zu synchronisieren. Das kann erhebliche Kosten für Prozessorlast und Speichervolumen bewirken.

Ein virtuelles Data-Warehouse, das auf die Materialisierung der Konsolidierungs-Datenbank und/oder der Bereitstellungs-Datenbank verzichtet, verlagert die Kosten in die Query-Laufzeit und wird tendenziell zur Sicherung der Performance noch höhere Kosten für Prozessorlast verursachen. Der Entwurfsaufwand für die Konsolidierung ist natürlich unverzichtbar.

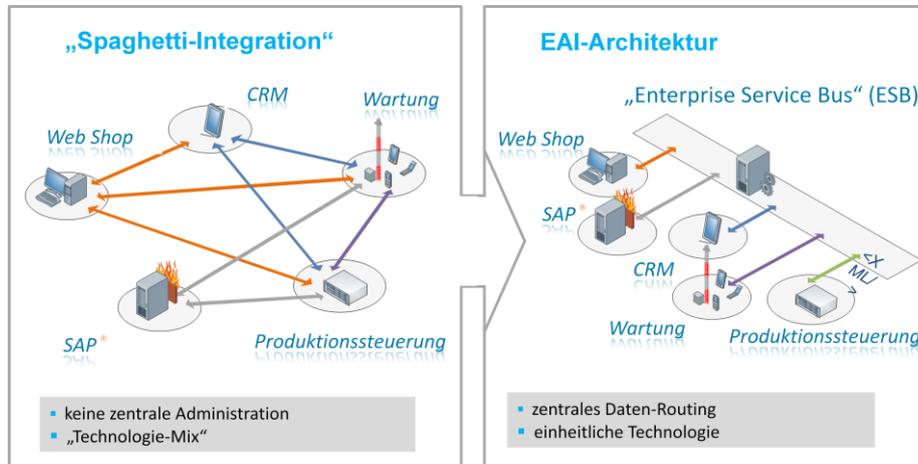
Eine Konsolidierungs-Datenbank oder das vollständige Data-Warehouse auf in-memory-Basis kann die beste Performance liefern, die damit erhöhten Kosten könnten aber die Frage nach Kostendämpfung durch einen primären Enterprise-Ansatz motivieren.

Unabhängig von den verschiedenen Technologien zur Realisierung der Enterprise-Sicht gibt es in der Data-Warehouse-Konzeption naturgemäß keine Rückwirkung der Bereinigung und Homogenisierung der Daten auf ihre Quellen. Somit bleiben alle Mängel und Ineffizienzen der unternehmensweiten Geschäftsprozesse, die sich aus der unkoordinierten Entwicklung von Softwaresystemen mit eigenen Datenbanken ergeben, weiter bestehen.

### **3 Prozessqualität erfordert Integration**

Die Realität der heterogenen Anwendung von Softwaresystemen, die nicht oder wenig koordiniert entwickelt wurden, hat sich immer stärker ausgeprägt und dauert zu einem großen Teil bis heute an.

Steigende Anforderungen an die Unternehmensprozesse hinsichtlich Geschwindigkeit, Korrektheit und Effizienz erforderten es, den Datenaustausch und die Synchronisation der Systeme in Echtzeit oder zeitgesteuert zu unterstützen. Replikation und 2-Phase-Commit können hierbei nur in Spezialfällen helfen. Neben einem sehr großen Anteil anwendereigener Lösungen kam es zur Produktentwicklung mit der Bezeichnung Enterprise Application Integration (EAI) oder auch Enterprise Service Bus (ESB). Diese bieten die Möglichkeit anstelle der in der Praxis häufig vorkommenden ungeordneten peer-to-peer Integrationsnetze eine geordnete Architektur zu realisieren, die eine gute Administration, Weiterentwicklung und ein Monitoring erlaubt (Bild 3).



**Bild 3.** Architektur der Integration

Ein entsprechender Integrationsserver verfügt im allgemeinen über eine Menge geeigneter Adaptoren, die die Datenstrukturen der Softwaresysteme senden und empfangen können sowie über eine leistungsfähige Mapping- und Routing-Funktionalität. Damit ergibt sich die Möglichkeit, aktuelle Daten der führenden Systeme so schnell wie nötig allen anderen zu übergeben, die diese Daten benötigen. Bestimmte Definitionen ermöglichen auch die Steuerung von Geschäftsprozessen.

Insoweit könnte man annehmen, dass die Integrationstechnologie die Zielstellung und Notwendigkeit einer Enterprise-Datenbank weitgehend ersetzen kann, ohne dass ein Enterprise-Datenbank-Schema explizit existiert. Dazu sei aber die Einschätzung erlaubt, dass die heutige Anwendung der Integrationstechnologie in der Praxis oft noch nicht das Niveau einer systematischen Enterprise Integration erreicht hat, stattdessen werden schrittweise Schwerpunktaufgaben behandelt, die nur sehr offensichtliche Prozess-Verbesserungen sichern.

Zu bedenken sei aber auch die seit mehr als 10 Jahren von Gartner und anderen publizierte Feststellung, dass der Aufwandsanteil für Integration in der IT weit mehr als 30% beträgt und dass Integration mit steigender Tendenz ein kritischer Teil der Informationsversorgung ist [5]. Auch wenn die Anwendung professioneller produktbasierter Integrationslösungen den spezifischen Aufwand reduziert, die Integrationsdichte der Softwaresysteme eines Unternehmens aber absolut bedeutend zunehmen wird, kann man wohl annehmen, dass dieser Anteil nicht kleiner wird.

In dieser Hinsicht könnte auch die Ressourcenfrage bedeutsam werden.

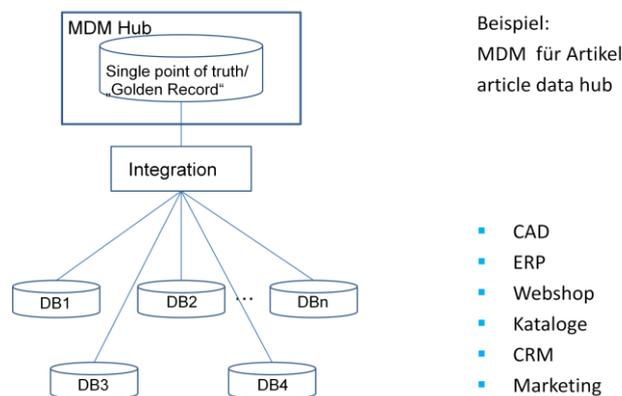
#### 4 Datenqualität erfordert Master Data Management

Typischerweise sind Objekte wie Geschäftspartner oder Artikel jeweils in mehreren Datenbanken eines Unternehmens enthalten. Ein Integrationssystem kann neue Daten vom jeweiligen Ursprungssystem gemäß geltender Regeln zu den Systemen transfe-

rieren, die diese Daten benötigen. Ein hohes Niveau der Integration der Daten eines Unternehmens durch Vermeidung der Mehrfacherfassung und schnelles Synchronisieren sichert aber nicht automatisch eine hohe Datenqualität.

Hierzu haben sich Anforderungen ausgeprägt wie z.B. Validierung, Standardisierung, Vervollständigung, Klassifizierung und Identifikation mit Doublettenerkennung und –behandlung [6].

Die Realisierung dieser Anforderungen erfordert eine Architektur, die seit einigen Jahren grundlegend zum Master Data Management (MDM) gehört.



**Bild 3.** Grob- Architektur des Master Data Management

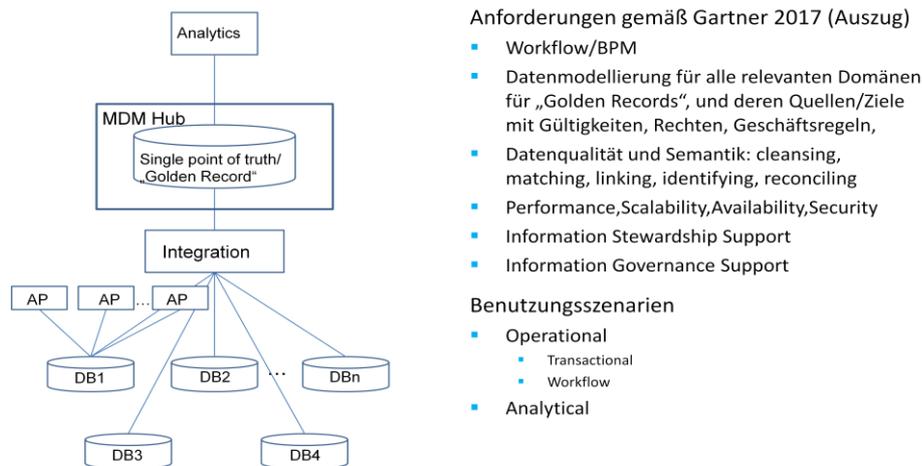
“**Master data management (MDM)** is the practice of defining and maintaining consistent definitions of business entities (e.g., customer or product) and data about them across multiple IT systems and possibly beyond the enterprise to partnering businesses. MDM gets its name from the master and/or reference data through which consensus-driven entity definitions are usually expressed. An MDM solution provides shared and governed access to the uniquely identified entities of master data assets, so those enterprise assets can be applied broadly and consistently across an organization.” [7]

In der MDM-Architektur gibt es eine spezielle Datenbank, üblicherweise als MDM-Hub bezeichnet, die eine konsolidierende Darstellung der relevanten Objekte des Unternehmens verwaltet, also nicht nur wie ursprünglich als „article data hub“ oder „customer data hub“ sondern als multi-domain-hub. Von den verschiedenen Softwaresystemen erfasste Daten werden mit Hilfe des Integrationssystems dem MDM-hub übergeben. Dort werden die definierten Regeln zur Sicherung der Datenqualität ausgeführt, im positiven Fall entsteht ein „golden record“, der mit Hilfe des Integrationssystems gemäß Bedarf an andere Softwaresysteme in der jeweils erforderlichen Darstellung publiziert wird. Der MDM Hub gilt als „single point of truth“. Meist ist im insert-Falle eine tiefgründige Identitätsprüfung erforderlich, weil die Vermeidung von Doubletten wirtschaftlich relevant ist. Dazu ist eine anspruchsvolle unscharfe Suche zur Erkennung gleicher oder ähnlicher Objekte erforderlich. Der hohe Anspruch des MDM an Datenqualität führt zwangsläufig auch dazu, dass nicht

alle Datenänderungen transactional ausgeführt werden können, sondern Entscheidungs- und Freigabeprozesse einschließen, die Workflows benötigen.

Die MDM-Nutzung an sich und speziell auch die Workflow-Notwendigkeit bewirken, dass die beteiligten Softwaresysteme Anpassungen erfordern.

Die in den letzten Jahren interessant gewachsene Menge von Anforderungen an MDM und dementsprechende Funktionsangebote der MDM-ProduktHersteller findet man im aktuellen Gartner-Report [8]. Bild 4 enthält eine vereinfachte Liste dieser Anforderungen.



**Bild 4.** Wichtige Anforderungen an Master Data Management

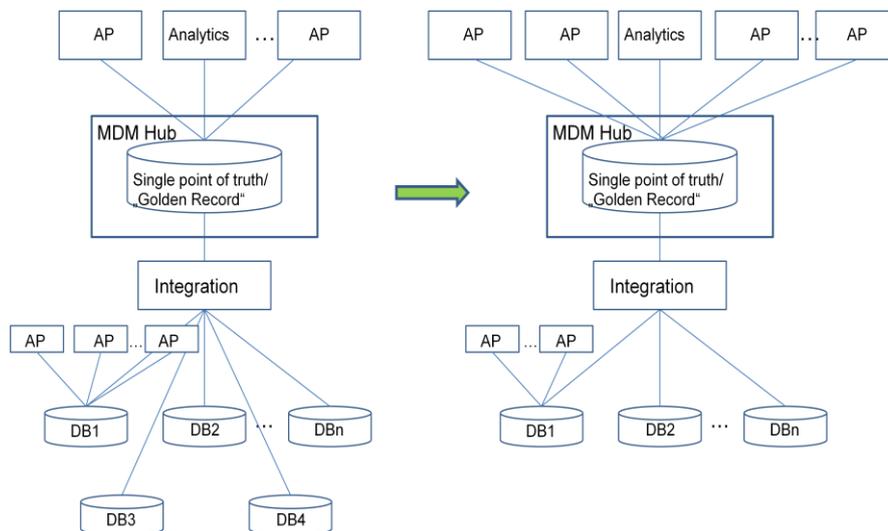
Auffällig ist dabei auch der Trend zu einer wesentlich verbesserten Darstellung der Semantik, was die Einbeziehung auch der Daten, die nicht zu den Master Data gezählt werden in den MDM Hub nahelegt. Insoweit erfüllt der MDM Hub die Rolle einer Konsolidierungsdatenbank im Data-Warehouse. Als Konsequenz wird dem MDM neben den operationalen Benutzungsszenarien Transactional und Workflow nun auch Analytics zugeordnet. In Bild 4 sind die Anwendungsprogramme mit operationaler Funktionalität wie üblich in Verbindung mit ihren spezifischen Datenbanken dargestellt.

MDM-Hersteller wie z.B. Semarchy Inc. [9] und auch der Gartner-Report[8] unterscheiden die mögliche Wirkungsweise des MDM in 4 Stufen. Sie werden als Implementation Styles bezeichnet. Tabelle 1 gibt einen Überblick zur Wirkung der Implementation Styles, wobei als „Daten“ hauptsächlich Master Data zu verstehen sind, jedoch kann es sinnvoll sein, auch andere Daten einzubeziehen, weil sich damit möglicherweise die Behandlung der Semantik im MDM und dessen Wirkung verbessern lässt. Außerdem könnte dadurch die Kooperation der Anwendungssoftware mit dem MDM vereinfacht werden.

**Tabelle 1.** Wirkungsweise des MDM

Implementation Style	Verfahren	Wirkung
Registry	Zentraler Index verweist auf alle Vorkommen eines Objekts	Daten bleiben unverändert
Consolidation	MDM liefert eine Konsolidierungsdatenbank	Operationale Daten unverändert, vergleichbar mit Data-Warehouse
Coexistence	MDM Hub verwaltet Teil der Daten, produziert golden records und publiziert diese an operationale Systeme	Zugriff auf diese Daten im MDM Hub oder in den operationalen Systemen möglich, weiterhin Mehrfachspeicherung
Centralized	MDM Hub verwaltet alle Daten zentralisiert	Zugriff auf die Daten (transactional oder workflow) im MDM Hub, keine Mehrfachspeicherung

Die Einführung eines MDM-Systems mit dem Implementation Style Coexistence könnte der pragmatisch sinnvolle Weg eines gleitenden Übergangs von einer heute üblichen schwach koordinierten Software-Architektur zu einer vollständig oder weitgehend zentralisierten MDM-Architektur sein. Anwendungsprogramme könnten zunehmend für Daten des Hubs entwickelt oder auf diese umgestellt werden (Bild 5). In der Konsequenz wäre das der Weg zu einer Enterprise-Datenbank. Daten, die nicht zu den Master Data zählen oder keine mehrfachen Repräsentationen benötigen, können mit zentralisiert werden oder in spezifischen Datenbanken verbleiben, wobei auch hierbei die Qualitätssicherung für eine MDM-gemäße Behandlung sprechen könnte.

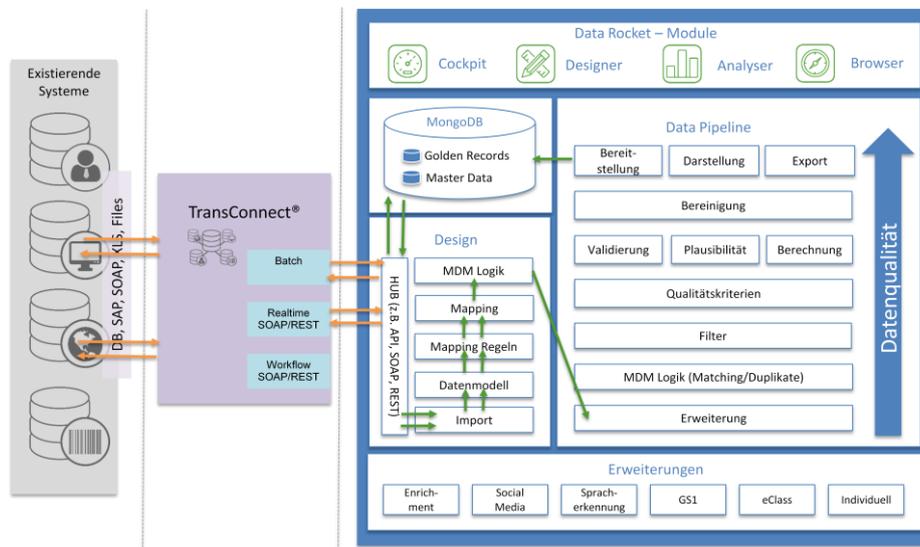


**Bild 5.** Coexistence-Architektur im Wandel

Falls sich der Bedarfstrend zum Master Data Management weiter entwickelt wie in den letzten Jahren, ist zu erwarten, dass die Anforderungen an einen unternehmensweiten Datenbank-Entwurf steigen, der dann vom MDM zu leisten ist. Konsolidierendes Entwerfen auf der Basis einer 3-Ebenen-Architektur müsste von den existierenden Datenbanken ausgehen, die als externe Sichten genutzt werden, möglicherweise auf Basis unterschiedlicher Datenmodelle, wobei auch unstrukturierte Daten zunehmend berücksichtigt werden sollten. Dabei ist die Darstellung und Behandlung der Semantik besonders zu unterstützen. Außerdem muss eine dynamische Weiterentwicklung des Entwurfs bzw. der Hubstruktur möglich sein.

In gleichem Maße wie MDM mit Coexistence Style oder Centralized Style benötigt wird, sind Anpassungen der Anwendungssoftware erforderlich. Die Softwarehersteller müssen zur Kooperation mit dem MDM bereit sein und dessen Rolle anerkennen. Das betrifft zunächst Interfaces und Workflows für Datenänderungen, die in den operationalen Systemen ausgelöst werden aber auch den Zugriff auf die vorhandenen Daten des MDM Hub. Hierin liegt die Hauptschwierigkeit, die einen schnellen Übergang zu einem Centralized MDM behindern dürfte. Jedoch könnte der schrittweise Ausbau eines Coexistence MDM bis zu einer „Full Coexistence“ erhebliche Prozessverbesserungen bewirken.

Wenn auch im MDM mit Coexistence Style die Mehrfachspeicherung der Daten im Prinzip noch nicht beseitigt, aber vielleicht reduziert werden kann, wäre es interessant, fallweise zu untersuchen, ob damit oder auch im MDM mit Centralized Style relevante Einsparungen an Ressourcen (Speicher und Integrationslast) erzielt werden können.



**Bild 6.** Beispiel einer Master Data Management-Architektur mit Implementation Style Coexistence auf der Basis von Data Rocket[10] und dem Integrationssystem TransConnect® [11]

## 5 Software-Entwicklung und Enterprise-Datenbank

Natürlich werden sich die Data-Warehouse-Technologie, die Integration und das Master Data Management noch lange weiterentwickeln. Es ist aber offensichtlich, dass alle 3 Technologien mehr oder weniger das Fehlen der Enterprise-Datenbank ausgleichen müssen. Besonders deutlich wird das bei einer konsequenten MDM-Zielstellung.

Sollte die ursprüngliche Vision einer Enterprise-Datenbank auf dem Wege des Master Data Managements oder direkt jemals eine Chance bekommen, müsste sich die datenbank-orientierte Software-Herstellung mit folgenden zwei Zielen auseinandersetzen, wobei die Datenbank im allgemeinen nicht mehr Bestandteil eines einzelnen Softwareproduktes sein dürfte, sondern in einem eigenständigen Entwurfsprozess entsteht und dynamisch weiterzuentwickeln ist:

**Standardisierung.** Die Standardisierung fach- und branchenspezifischer Datenstrukturen könnte zu einem Teil eine Voraussetzung sein, dass Software unterschiedlicher Hersteller mit der gleichen Datenbank arbeitsfähig wird. Die Realität ist von dieser Zielstellung weit entfernt. Auch wenn große Anstrengungen für diese Zielstellung geleistet würden, bliebe das Ergebnis begrenzt. Beispiele, bei denen verschiedene Hersteller die gleiche Datenbank benutzen, sind aber bekannt.

Für den Austausch von Daten existieren dagegen verschiedene Branchenstandards, die von Branchenprodukten und Integrationssystemen genutzt werden. Bestimmte dieser Standards haben eine so gute Durchsetzung erreicht, teilweise durch Festlegung verantwortlicher Organisationen, dass Softwareprodukte, die diese Standards nicht unterstützen, ihre Marktfähigkeit verlieren. Ein Beispiel ist HL7 [10], ein Standard für den Datenaustausch klinischer Informationssysteme.

**Softwareprodukte mit virtueller Datenbank-Schnittstelle.** Wenn Softwareprodukte die benötigte Datenbank nicht selbst enthalten, müssten sie sich einer Schnittstelle bedienen, die die notwendigen Zugriffe ermöglicht. Die Anwendbarkeit einer solchen virtuellen Datenbank-Schnittstelle hängt davon ab, ob das entsprechende Schema auf das Schema der vorhandenen oder sich entwickelnden Enterprise-Datenbank abbildbar ist. Seit den 90er Jahren vorliegende Forschungsergebnisse zum Schema Matching und Schema Mapping (vgl. z.B. [11]) müssten dazu in Entwurfswerkzeugen verfügbar sein, mit denen diese Aufgabe bewältigt werden könnten.

Die zwangsläufig notwendigen Transformationen der Daten bewirken natürlich einen neuen Aufwand, der aber durch den Wegfall eines großen Teils der Transformationen des Datenaustauschs ausgeglichen wird.

Die Einführung neuer Software im Unternehmen würde dann entweder in das vorhandene Enterprise-Datenbank-Schema hineinpassen oder würde dieses erweitern. Semantische Widersprüche, die aus den Schemas verschiedener Software-Produkte resultieren, können möglicherweise auch zu Ergänzungen führen. Bei dieser Architektur ist im Enterprise Schema mit einer Dynamik zu rechnen, die in einem traditionellen DBMS schwer zu bewältigen ist. Mehr-Ebenen-Architektur ähnlich [1] müsste aus diesem, aber auch anderen Gründen ein Ziel der DBMS-Hersteller sein.

Die schrittweise Entstehung einer Enterprise-Datenbank ist heute mehr als in der Frühzeit der Datenbank-Technologie vorstellbar. Wesentliche Voraussetzungen haben

einen besseren Stand. Ob das zweifellos große und weiter wachsende Nutzenspotential ein ausreichender Antrieb ist, die Schwierigkeiten auf dem Wege dahin zu überwinden, wird die Zukunft zeigen.

## Verweise

1. ANSI/X3/SPARC: Interim Report, Study Group on Data Base Management System, (1975)
2. Bittner, J.: An External Schema for Application Programmers. In: Proceedings of the Sixth International Seminar on Database Management Systems, pp. 202-215. SZAMALK Computing Application and Service Co., Budapest (1983)
3. Bittner, J.: Design of the Conceptual Scheme on the Base of External Schemes. In: Proceedings of the Seventh International Seminar on Database Management Systems, pp. 47-55. Varna (1984).
4. Lehner, W.: Datenbanktechnologie für Data-Warehouse-Systeme. 1.Aufl. dpunkt-verlag, Heidelberg (2013).
5. Thoo, E., Friedmann, T., Beyer, M.: Magic Quadrant for Data Integration Tools, Gartner (2013).
6. Russom P.: Real Time Data Quality, p. 3. TDWI Checklist Report, (September 2011).
7. Russom P.: Next Generation Master Data Management, p. 5. TDWI Best Practices Report, (Second Quarter 2012).
8. O’Kane, B., Palanca, T., Moran, M.: Magic Quadrant for Master Data Management Solutions, Gartner (2017).
9. Semarchy Inc. Homepage, <https://www.semarchy.com/intelligent-mdm/>, last accessed 2017/06/15.
10. Innoscale AG, Homepage, <http://innoscale.de>, last accessed 2017/06/15.
11. SQL Projekt AG, Homepage, <http://www.sql-ag.de>, last accessed 2017/06/15.
12. Health Level Seven International (HL7) Homepage, [www.hl7.org](http://www.hl7.org), last accessed 2017/06/15.
13. Rahm, E., Bernstein, P.: A survey of approaches to automatic schema matching. In: The VLDB Journal 10 , pp. 334–350 (2001), Springer-Verlag( 2001).