

# SAP Business Information Warehouse Architecture

SAP entered the data warehouse market when it started maturing and has been able to take advantage of the experience available and avoid many mistakes made by early adopters. To those familiar with other data warehouse solutions and custom data warehouse development, as well as anyone following discussions about data warehousing, the high-level SAP business warehouse (BW) architecture will look familiar.

Building SAP BW on top of the SAP Web Application Server (formerly known as SAP Basis), SAP has been able to inherit not only the multi-tier architecture implemented there but also a complete software development environment, a large amount of systems management tools, and a lot of additional functionality (e.g., currency conversion or security) and tools available there. Because SAP BW is implemented on top of the SAP Web Application Server, it is often considered a part of or an add-on to SAP R/3. This is not correct: SAP BW, though still closely related to SAP R/3, is a completely separate software package that can be used in any—SAP or non-SAP—environment. At the end of this chapter, we'll have a closer look at the common architecture of SAP systems and how it serves SAP BW.

In Chapter 3 we continue the discussion from Chapter 2 on the SAP Business Intelligence strategy and how SAP BW is embedded into this. We provide an overview of the SAP BW architecture and its meta data concept, and map the SAP BW features to the corporate information factory (CIF) concept developed by Bill Inmon. This chapter concludes with the architecture of the SAP Web Application Server.

## 36 Chapter 3

# SAP BW Architectural Components

Figure 3.1 shows a high-level view of the SAP BW architecture, with six main building blocks. It is no surprise to see that SAP BW is completely based on an integrated meta data concept, with meta data being managed by *meta data services*. SAP BW is one of the few data warehouse products that offer an integrated, one-stop-shopping user interface for administering and monitoring SAP BW: the administration services available through the Administrator Workbench.

Looking at the center of Figure 3.1, we find the usual layered architecture of an end-to-end data warehouse accompanied by two administrative architectural components:

- Extraction, loading, and transformation (ETL) services layer
- Storage services layer, including services for storing and archiving information
- Analysis and access services layer, providing access to the information stored in SAP BW
- Presentation services layer, offering different options for presenting information to end users
- Administration services
- Meta data services

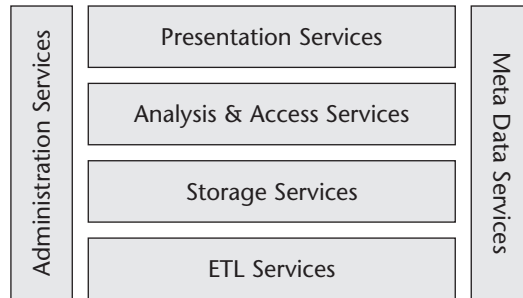
Each layer will be discussed in more detail in the sections that follow.

## Administration Services

The *administration services* include all services required to administer an SAP BW system. Administration services are available through the *Administrator Workbench (AWB)*, a single point of entry for data warehouse development, administration, and maintenance tasks in SAP BW. Figure 3.2 shows a screenshot of the AWB.

As the most prominent architectural component, the AWB includes a *meta data modeling* component, a *scheduler*, and a *monitor*, as shown in Figure 3.2. Other components of the AWB include the following:

- The *transport connector* supports the information modeling and development process by collecting objects that have to be transported from the development system to a test or production system and assigning those objects to transport requests. The details of the transport connector are covered later in this chapter.
- The *reporting agent* allows scheduling query execution for batch printing or raising exception alerts. The reporting agent is covered in more detail in Chapters 7 and 9.
- The *document interface* allows managing documents of any type assigned to SAP BW meta data objects.
- The *translation component* supports the implementation of multilingual systems.



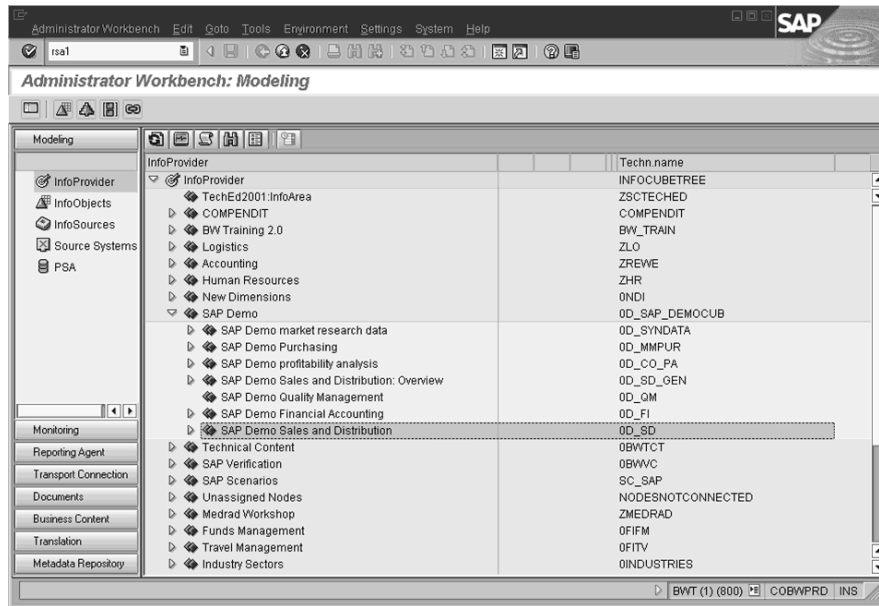
**Figure 3.1** SAP BW architecture.

### Meta Data Modeling

Like other data warehouse solutions, SAP BW is based on meta data, or data about data. Bill Inmon defines three classes of meta data:

- Technical meta data
- Business meta data
- Operational meta data

On the other hand, SAP BW distinguishes two basic classes of meta data: predefined meta data called Business Content and client-defined meta data. Both classes of meta data can be maintained using the same user interface. A detailed definition of the Business Content and a discussion of its role in SAP BW implementation projects can be found in Chapter 5.



**Figure 3.2** The Administrator Workbench.

Copyright © SAP AG

## 38 Chapter 3

Business and technical meta data is commonly referred to as meta data. Operational meta data refers to data about processes as opposed to data about data. SAP BW maintains all three types of meta data. However, the SAP BW meta data objects are used to model and maintain business and technical meta data, while operational meta data is generated by data warehouse processes and is available through scheduling and monitoring components.

The modeling functionality shown in Figure 3.2 is the most important part of the AWB, as it provides the main entry point for defining the core meta data objects used to support reporting and analysis. This includes everything from defining the extraction processes and implementing transformations to defining flat or multidimensional objects for information storage. The Business Content component allows you to browse through the predefined models available and activate them. Once activated, you can use these information models without further modification or extend them using the modeling component of the AWB.

The Meta Data Repository provides an online hypertext documentation of either activated meta data objects (the ones actually used in the BW system) and the meta data objects of the Business Content. You can export this hypertext documentation to a set of HTML files and publish it on a Web server, where it may also serve as an online and automatically updated project documentation.

An offline meta data modeling tool tentatively called *Repository Studio* is currently under development at SAP. The Repository Studio is designed to support offline meta data modeling for SAP BW meta data. SAP BW meta data is imported into the offline repository. There you can modify it using the modeling functionality of the Repository Studio and export it back into an SAP BW system. The Repository Studio is a completely Web-based, multi-user application that you can use in team environments without having to be connected to an SAP BW system. However, it still supports working offline (e.g., on laptops while traveling) by integrating a standalone Web server.

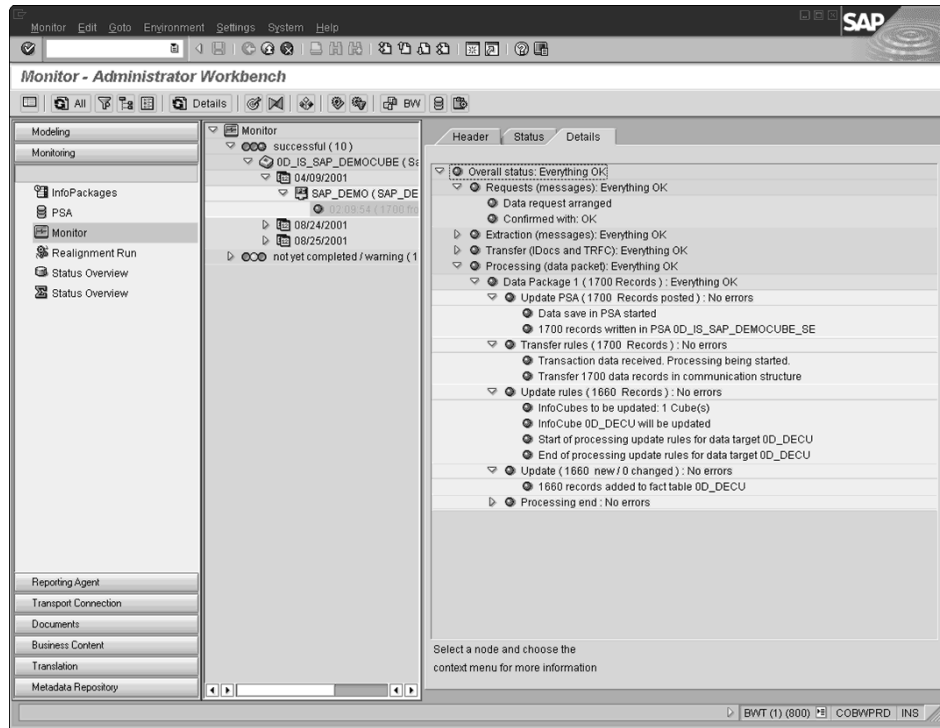
### **Scheduling**

Data warehousing requires batch processing for loading and transforming data, creating and maintaining aggregates, creating and maintaining database indexes, exporting information to other systems, and creating batch reports. These processes need to be planned to provide results in time, to avoid resource conflicts by running too many jobs at a time, and to take care of logical dependencies between different jobs.

SAP BW takes care of controlling these processes in the scheduler component by either scheduling single processes independently or defining *process chains* for complex networks of jobs required to update the information available in the SAP BW system. In addition, the scheduler supports Business APIs (BAPIs) used by third-party scheduling tools, such as IBM's Tivoli and Computer Associate's Unicenter Autosys. Both the scheduler and the monitor component are explained in detail in Chapter 9.

### **Monitoring**

Equally important as starting batch processes is monitoring and eventually troubleshooting them. This is what the SAP BW monitor is designed for. Figure 3.3 shows a screenshot of the Data Load Monitor.



**Figure 3.3** The Data Load Monitor.

Copyright © SAP AG

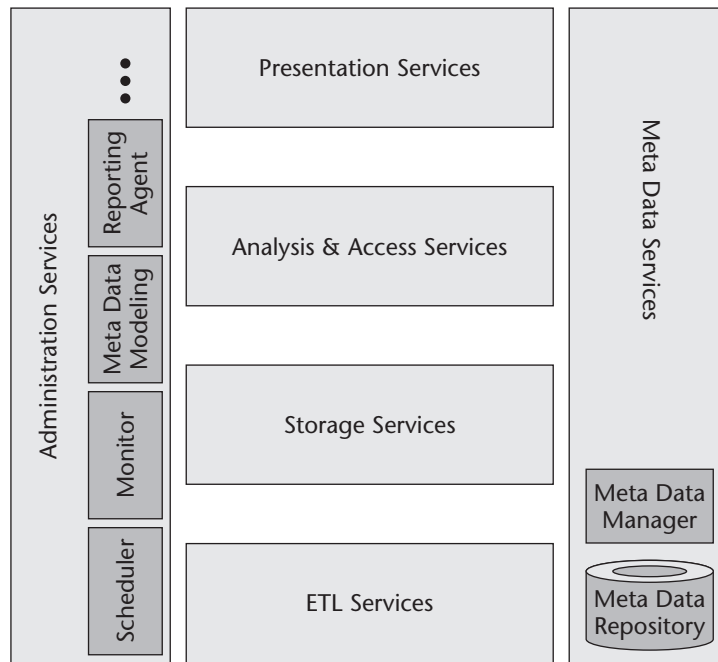
The Data Load Monitor supports troubleshooting by providing access to detailed protocols of all activities related to loading, transforming, and storing data in SAP BW—allowing you to access single data records and to simulate and debug user-defined transformations. Other processes monitored are ODS object activation, master data attribute activation, hierarchy activation, aggregate rollup, realignment and readjustment jobs, InfoCube compression jobs, database index maintenance, database statistics maintenance, and data exports.

### Reporting Agent

The *reporting agent* allows the execution of queries in batch mode. Batch mode query execution can be used to:

- Print reports.
- Automatically identify exception conditions and notify users responsible for taking appropriate action.
- Precompute query results for use in Web templates.
- Precompute value sets for use with value set variables (see the *Queries* section, later in the chapter, for a definition of variables).

## 40 Chapter 3



**Figure 3.4** Meta data services architecture.

### Meta Data Services

The SAP BW Meta Data Services components provide both an integrated *Meta Data Repository* where all meta data is stored and a *Meta Data Manager* that handles all requests for retrieving, adding, changing, or deleting meta data. The Meta Data Manager also allows the exchange of meta data with other systems compliant to the Common Warehouse Metamodel Initiative (CWMI) specified by the Object Management Group ([www.omg.org](http://www.omg.org)). Figure 3.4 shows the meta data services layer architecture.

Figure 3.5 shows the Meta Data Repository integrated into the Administrator Workbench, with a list of all meta data objects available there. A detailed discussion of the meta data available in SAP BW can be found in the SAP BW Meta Data Objects section later in this chapter.

### ETL Services

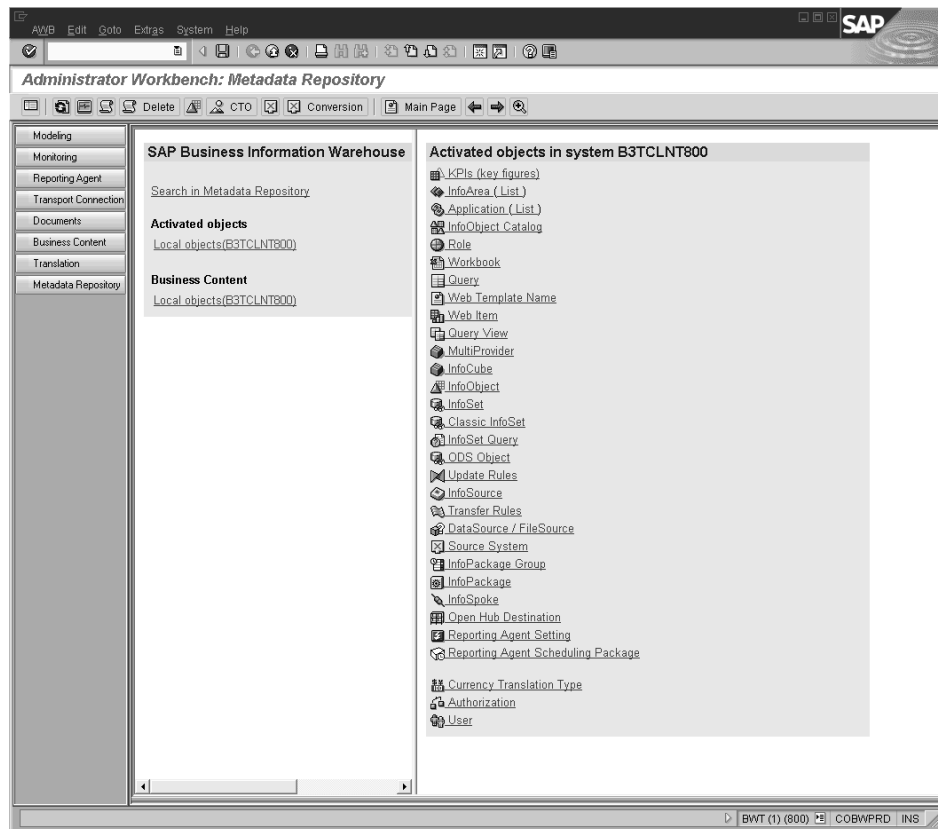
The extraction, transformation, and loading (ETL) services layer of the SAP BW architecture includes services for data extraction, data transformation, and loading of data and serves as a staging area for intermediate data storage for quality assurance purposes.

SAP BW has long been regarded as a proprietary solution, not allowing, or at least not very good at, loading data from non-SAP source systems. This is not true, and it

## SAP Business Information Warehouse Architecture 41

has not been true right from the early days of the 1.2 release. With the Staging BAPI, SAP has provided an open interface for exchanging meta data with SAP BW and uploading data to SAP BW. This interface has been widely adopted by ETL vendors like Ascential Software, ETI, and Informatica. While it has been limited to downloading meta data from SAP BW and uploading data to SAP BW, the Staging BAPI today supports two-way meta data transfers.

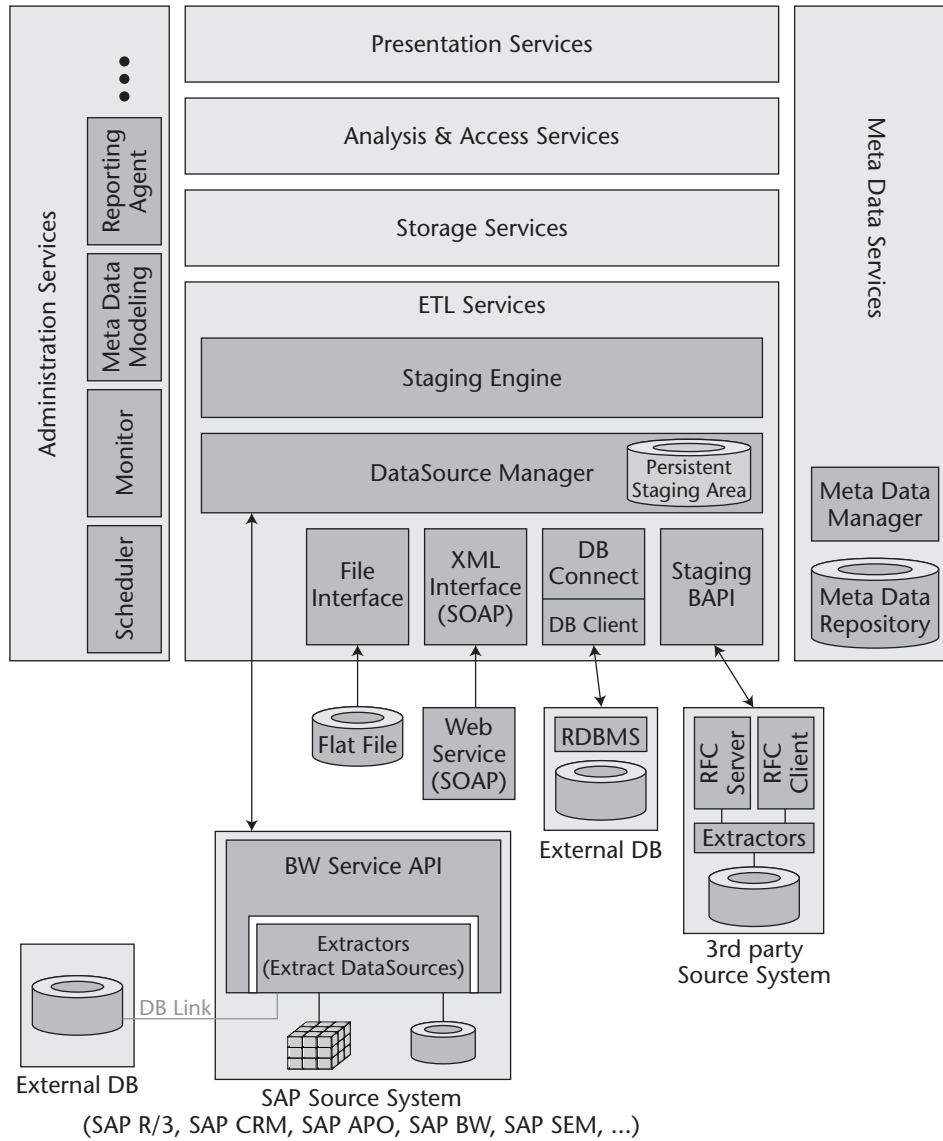
It is true, however, that the extraction technology provided as an integral part of SAP BW is restricted to database management systems supported by mySAP technology and that it does not allow extracting data from other database systems like IBM IMS and Sybase. It also does not support proprietary file formats such as dBase file formats, Microsoft Access file formats, Microsoft Excel file formats, and others. On the other hand, the ETL services layer of SAP BW provides all the functionality required to load data from non-SAP systems in exactly the same way as it does for data from SAP systems. SAP BW does not in fact distinguish between different types of source systems after data has arrived in the staging area. The ETL services layer provides open interfaces for loading non-SAP data. Figure 3.6 shows the architecture of the ETL service layer.



**Figure 3.5** SAP BW Meta Data Repository.

Copyright © SAP AG

**42 Chapter 3**



**Figure 3.6** ETL services architecture.

**Staging Engine**

The core part of the ETL services layer of SAP BW is the *Staging Engine*, which manages the staging process for all data received from several types of source systems. The Staging Engine generates and executes transformation programs, performing the



transfer and update rules defined in the AWB. It interfaces with the AWB scheduler and monitor for scheduling and monitoring data load processes. The Staging Engine does not care about the type of source system and applies the same staging process to non-SAP data as it does for SAP data.

However, the actual implementation of transformation rules will differ for different systems or different types of systems, simply because different systems may deliver data about the same business events (e.g., sales orders) using different record layouts, different data types, and different characteristics values for the same business semantics. In addition, different systems may provide different levels of data quality. A detailed discussion of the staging process can be found in Chapter 6.

### ***DataSource Manager***

The Staging Engine is supported by the *DataSource Manager*. The DataSource Manager manages the definitions of the different sources of data known to the SAP BW system and supports five different types of interfaces:

- BW Service API
- File interface
- XML interface
- DB Connect interface
- Staging BAPI

The DataSource Manager also allows capturing and intermediately storing uploaded data in the persistent staging area (PSA). Data stored in the PSA is used for several purposes:

**Data quality.** Complex check routines and correction routines can be implemented to make sure data in the PSA is consistent before it is integrated with other data sources or is uploaded to its final data target.

**Repeated delta updates.** Many extraction programs do not allow you to repeat uploads of *deltas*, which are sets of records in the data source that have been inserted or updated since the last upload. Repeated delta uploads are required in cases where the same delta data has to be updated into multiple data targets at different points of time.

**Short-term backup data source.** A short-term backup data source is required in cases where update processes fail for some technical reason (such as insufficient disk space or network availability) or where subtle errors in the transformations performed on the data warehouse side are only discovered at a later point in time. Once stored in the PSA, data may be read from the PSA and updated into the final data target at any point in time and as often as required.

**Supporting development.** Based on data in the PSA, SAP BW allows you to simulate transfer rules, and update rules, and to debug the implemented transformations.

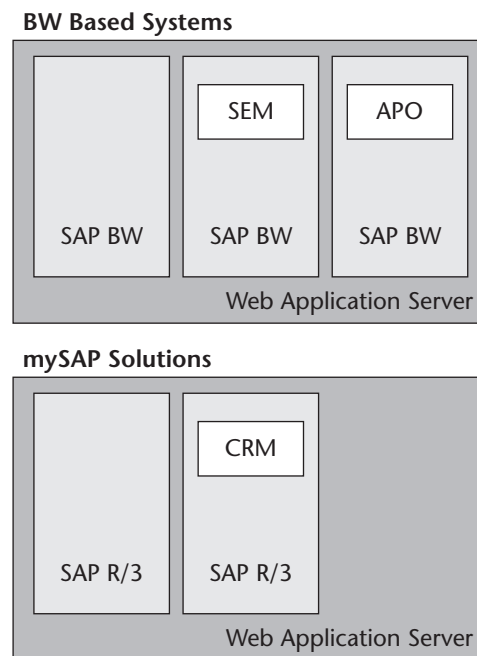
## 44 Chapter 3

### ***BW Service API***

The most important interface supported by the DataSource Manager in SAP environments is the *BW Service API*. The BW Service API is available for two basic types of SAP systems: SAP R/3-based systems, including SAP R/3 and SAP Customer Relationship Management (mySAP CRM™), and SAP BW-based systems, such as SAP BW itself; SAP Strategic Enterprise Management (mySAP SEM™); and SAP Advanced Planner and Optimizer (mySAP SCM™). SAP R/3-type systems usually provide operational data, while SAP BW-based systems allow the creation of complex information flow scenarios with cascading SAP BW instances (see Figure 3.7).

The BW Service API provides a framework for data replication from SAP systems, including generic data extraction, sophisticated delta handling, and online access to extraction programs via the remote InfoCube technology. It handles all communication between the source system and the requesting SAP BW system and makes a wide range of predefined extraction programs—encapsulating application know-how—available to SAP BW. It is included in most mySAP.com application components (such as SAP BW, SAP SEM, and SAP APO) and is available as part of the SAP R/3 plug-in, which also includes the actual extraction programs for SAP R/3.

Extraction programs either are part of the Business Content from where they may be enhanced according to client requirements or they are custom extraction programs defined by the client using the generic extraction technology. Generic extractors allow accessing any table or database view available in the SAP ABAP dictionary. Used in an SAP BW-based systems, the BW Service API provides access to the data stored in master data tables, ODS objects, and InfoCubes.



**Figure 3.7** Types of SAP systems.

To provide access to external, non-SAP databases, SAP has developed the *DB Link* tool, which allows access to data stored in an external database through the BW Service API. The basic idea behind the DB Link tool is to connect to a remote database, to make the required remote table or view visible to the SAP ABAP Workbench dictionary, and to define a generic extractor for the remote table or view. The DB Link tool is supported for Oracle, Microsoft SQL Server, and IBM DB/2 databases.

With the DB Connect Interface a part of SAP BW release 3.0, the DB Link tool will lose relevance for new developments. We strongly recommend using the DB Connect interface instead.

### ***DB Connect Interface***

The *DB Connect interface* is pursuing the same goal as the DB Link interface in that it connects to a remote database and makes remote database tables available to an SAP BW system. The technical implementation, however, is completely different. The DB Connect interface uses core parts of the SAP database interface layer and the database client software (which needs to be installed separately if the remote database system differs from the local database system) to connect to the remote database. The DB Connect interface can read the remote data dictionary, replicate table, and view meta data into the local SAP BW Meta Data Repository, and it allows extraction of data from those tables and views. The DB Connect interface supports all database systems supported by SAP BW.

### ***File Interface***

The *File interface* allows loading flat files of three different types into SAP BW:

**ASCII files.** The file interface reads ASCII files with fixed field lengths and variable record lengths, filling missing fields with blanks and ignoring extra fields at the end of data records.

**Comma-separated variables (CSV) files.** CSV files are text files using a variable field delimiter (usually “;” or “,”) and variable field and record length. They are commonly used to exchange data among different applications.

**Binary files.** The File interface can import binary files that comply with the physical data format used by ABAP programs writing data in binary format (documentation on the physical format used can be found at <http://help.sap.com>).

### ***XML Interface***

The *XML interface* introduced with the SAP BW 3.0 release accepts XML data streams compliant with the Simple Object Access Protocol (SOAP). While all other SAP BW interfaces follow the pull philosophy, meaning that SAP BW pulls data out of these systems by initiating data load requests, the XML interface follows a push philosophy where the actual data transfer is initiated by the source system.

Data loads through the XML interface are always triggered by an external Web service using SOAP to send XML format data to an SAP BW system, where the data is temporarily stored using the delta queue mechanism. SAP BW pulls data out of that delta

## 46 Chapter 3

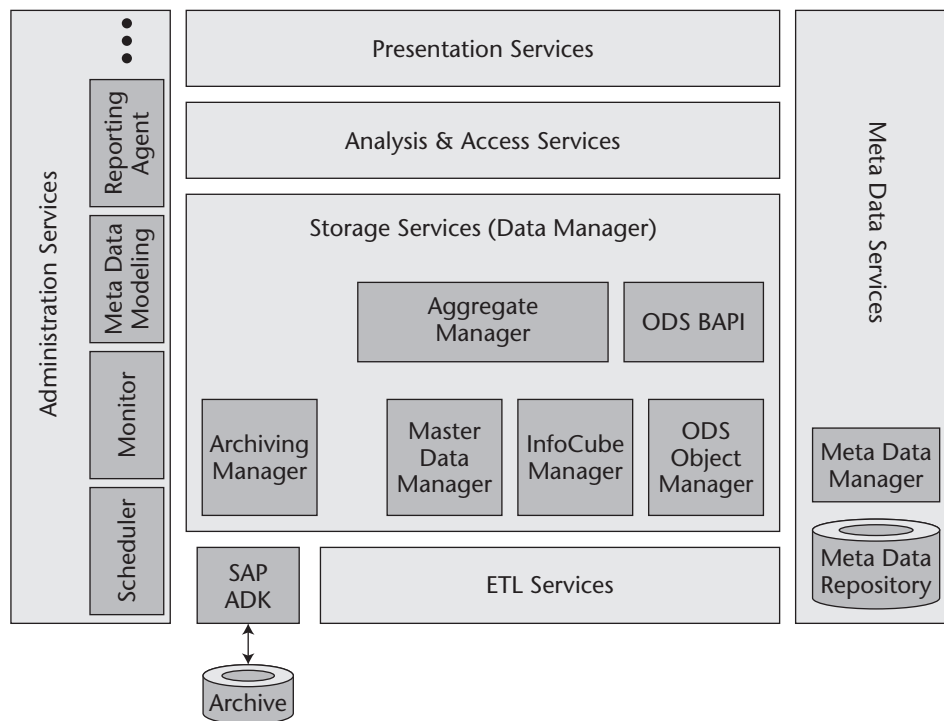
queue using the same scheduling mechanisms as for other interfaces. The XML interface and the push and pull philosophies are discussed in more detail in Chapter 6.

### Staging BAPI

The *Staging BAPI* is an open interface based on the BAPI technology. Available from the early days of SAP BW 1.2, the Staging BAPI allows third-party ETL tools as well as custom programs to connect to SAP BW, exchange meta data with SAP BW, and transfer data to SAP BW.

Systems using the Staging BAPI need to implement a simple RFC server program that waits for and schedules SAP BW data load requests, starts executing extraction programs accordingly, and sends the resulting data set back to SAP BW using Remote Function Call (RFC) client functionality. SAP has published detailed information about this open interface and has provided a sample extraction program implemented in Microsoft Visual Basic to showcase the use of the Staging BAPI.

As mentioned earlier in the chapter, the Staging BAPI has been widely adopted by third-party ETL tool vendors like Ascential Software, ETL, Informatica, and others. SAP has decided to strategically team up with Ascential Software to provide SAP BW clients with a low-cost, quality ETL solution for accessing arbitrary external database systems and file formats. A complete list of third-party ETL tools certified for use with SAP BW can be found on the SAP Web site ([www.sap.com](http://www.sap.com)).



**Figure 3.8** Storage services architecture.

## Storage Services

The *storage services* layer (also known as the SAP BW Data Manager) manages and provides access to the different data targets available in SAP BW, as well as aggregates stored in relational or multidimensional database management systems. The storage services connect to the SAP archiving module for archiving dormant data (data that is used infrequently or no longer used at all). Figure 3.8 provides an overview of the components of the storage services layer.

### **Master Data Manager**

The *Master Data Manager* generates the master data infrastructure consisting of master data tables as well as master data update and retrieval routines according to the definition stored in the Meta Data Repository. It maintains master data and provides access to master data for SAP BW reporting and analysis services. In Chapter 4 we'll take a closer look at the SAP BW master-data data model and discuss meta data that describes master data.

The task of maintaining master data includes:

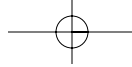
- Handling master data uploads
- Finding or generating surrogate keys
- Handling time windows for time-dependent master data
- Ensuring the technical correctness of master data hierarchies
- Providing a generic user interface for interactive master data maintenance
- Activating master data, a process that copies modified data in the master data tables from a *modified* version, which is not visible in reporting and analysis, to an *active* version, which is visible in reporting and analysis

From an output point of view, the Master Data Manager provides access to the master data for use by SAP BW reporting components (e.g., the BEx Analyzer), as well as for exporting to other data warehouse systems via the analysis and access services.

### **ODS Object Manager**

ODS objects are flat data structures used to support reporting, analysis, and data integration in SAP BW. The *ODS Object Manager* generates the ODS object infrastructure, which consists of an active data table, a change log, and an activation queue, as well as update and retrieval routines according to the definition stored in the Meta Data Repository. It maintains ODS object data, creates a change log for every update applied to the ODS object data as part of the activation process, and provides access to ODS object data for SAP BW reporting and analysis functionality.

The ODS Object Manager allows real-time updates to transactional ODS objects through the ODS API. Closely related to the ODS Object Manager, the ODS BAPI provides open read access to ODS objects. The details on the ODS object data model are



## 48 Chapter 3

---

discussed in Chapter 4, while more information about the ODS object meta data definition can be found later in this chapter.

**NOTE** While BAPIs are documented and supported from release to release, an API is not necessarily documented, or guaranteed to remain unchanged from release to release.

### *InfoCube Manager*

The main structures used for multidimensional analysis in SAP BW are called InfoCubes. The *InfoCube Manager* generates the InfoCube infrastructure consisting of fact and dimension tables, as well as the update and retrieval routines according to the definition stored in the Meta Data Repository. It maintains InfoCube data, interfaces with the Aggregate Manager (discussed in the next section) and provides access to InfoCube data for SAP BW reporting and analysis services.

More details on the InfoCube data model can be found in Chapter 4; a discussion of the InfoCube meta data definition can be found later in this chapter.

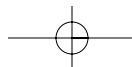
### *Aggregate Manager*

*Aggregates* are multidimensional data structures similar to InfoCubes containing an aggregated subset of information available through InfoCubes. Aggregates are used for optimizing reporting performance. The *Aggregate Manager* generates the aggregate infrastructure consisting of fact and dimension tables, along with the update and retrieval routines according to the definition stored in the Meta Data Repository. Maintenance of aggregates implies keeping track of updates applied to the underlying InfoCube and of updates to master data used in these aggregates, as well as applying those changes to the data stored in the aggregate.

Since SAP BW 3.0, aggregates can not only be stored in relational but also in multidimensional database systems, providing the best of both worlds to SAP BW users. Although SAP BW 3.0 was initially developed to support the multidimensional technology of Microsoft SQL Server, there are discussions about supporting other multidimensional database systems in future releases.

### *Archiving Manager*

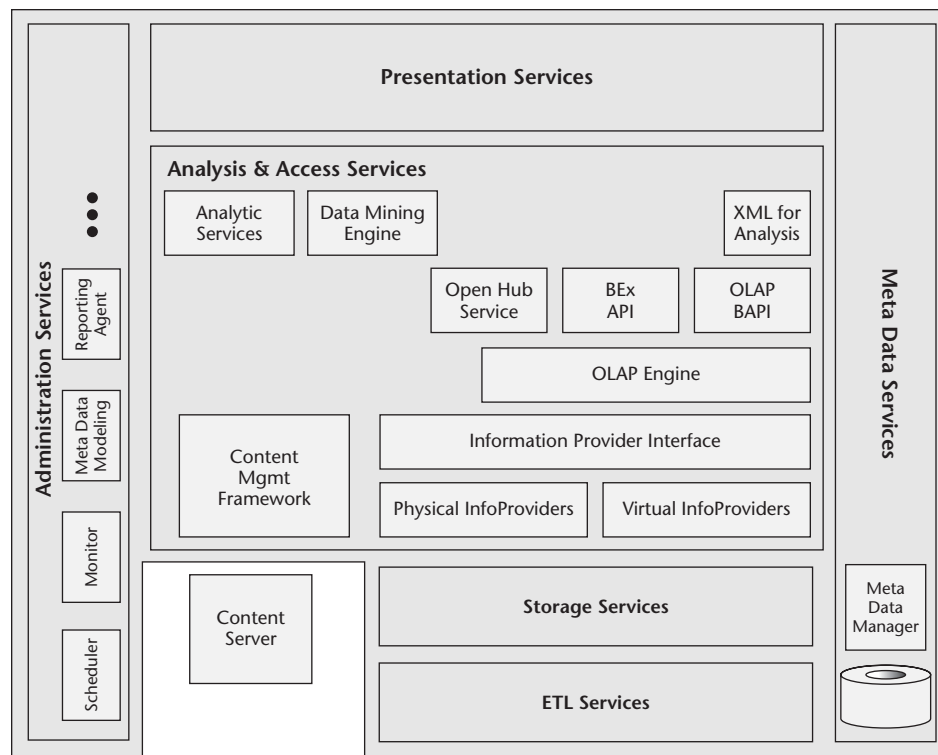
The *Archiving Manager* connects SAP BW to the *Archive Development Kit* (ADK). The Archiving Manager allows archiving unused, dormant data in a safe place, where it is still available if required. The Archiving Manager does not only store raw data, it also keeps track of relevant meta data—such as the layout of InfoCubes and ODS objects—which may change over time. Information archived using the ADK has to be restored in order to make it available for reporting again.



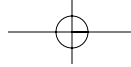
Another option available with the Archiving Manager in cooperation with FileTek's StorHouse solution allows transparent access to information archived from ODS objects without the need for explicitly restoring that information. For more information about this solution visit the FileTek Web site: [www.filetek.com](http://www.filetek.com).

## Analysis and Access Services

The *analysis and access services layer* provides access to analysis services and structured and unstructured information stored in the SAP Business Information Warehouse. Structured information is retrieved through so-named InfoProviders; unstructured information resides on a content server, which is accessed using the content management framework. Figure 3.9 provides an overview of the components of the analysis and access services layer.



**Figure 3.9** Analysis and access services architecture.



## 50 Chapter 3

---

### **Information Provider Interface**

With SAP BW 3.0 the *Information Provider interface* has been introduced to generalize access to data available in SAP BW. The Information Provider interface allows access to physical and virtual InfoProviders. Physical InfoProviders include basic InfoCubes, ODS objects, master data tables, and InfoSets physically available on the same system. Access to physical InfoProviders is handled by the storage services layer. Virtual InfoProviders include MultiProviders and remote InfoCubes. Access to virtual InfoProviders requires analyzing the request and routing the actual access to a remote system (in case of remote InfoCubes) or accessing several physical objects through the storage services layer (in case of MultiProviders).

### **OLAP Engine**

All analysis and navigational functions—like filtering, runtime calculations, currency conversions, and authorization checks—are provided by the *OLAP engine*. The OLAP engine retrieves query definitions from the Meta Data Repository, eventually generates or updates query execution programs, and finally executes the queries by running the generated program.

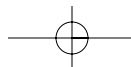
### **OLAP BAPI**

The *OLAP BAPI* provides an open interface for accessing any kind of information available through the OLAP engine. The OLAP BAPI specification is based on Microsoft's OLE DB for OLAP interface specification, utilizing the MDX language definition and adopting the basic API layout (functions and interfaces available). The OLAP BAPI is used by both third-party front-end tool vendors and SAP clients to provide specialized front-end functionality for the SAP BW end user.

The OLE DB for OLAP Interface (ODBO Interface) is an industry-standard interface proposed by Microsoft Corporation for accessing multidimensional data. The OLE DB for OLAP (or ODBO) Interface allows third-party front-end and analysis tools to connect to SAP BW and provide display, navigation, and specialized analysis functionality to end users. Although not designed for this purpose, the ODBO interface would also allow extracting small amounts of information from an SAP BW system for use in other custom or third-party software systems. For detailed information about OLE DB for OLAP, refer to [www.microsoft.com/data/oledb/olap](http://www.microsoft.com/data/oledb/olap).

### **XML for Analysis**

The OLAP BAPI serves as a basis for the SAP implementation of *XML for Analysis*. XML for Analysis is an XML API based on SOAP designed for standardized access to an analytical data provider (OLAP and data mining) over the Web.





### ***Business Explorer API***

The *Business Explorer API* connects the Business Explorer (BEx)—the SAP BW reporting and analysis front-end solution—to the OLAP engine, allowing access to all available queries. While the BEx API provides the most comprehensive set of functionality, it is not an officially published interface available for use by other applications.

### ***Open Hub Service***

The *Open Hub Service* allows controlled distribution of consistent data from any SAP BW InfoProvider to flat files, database tables, and other applications with full support for delta management, selections (filtering records), projections (selecting columns), and aggregation. All operations of the Open Hub Service are fully integrated into the scheduler and monitor.

### ***Analytic Services and Data Mining Engine***

As part of the Business Content and analytical application development, SAP has incorporated a number of analytical services, including a data mining engine, into SAP BW. While these services are integrated into the analytical applications (e.g., the data mining engine has originally been developed as part of the CRM analytics analytical application), they can still be used in custom applications.

### ***Content Management Framework***

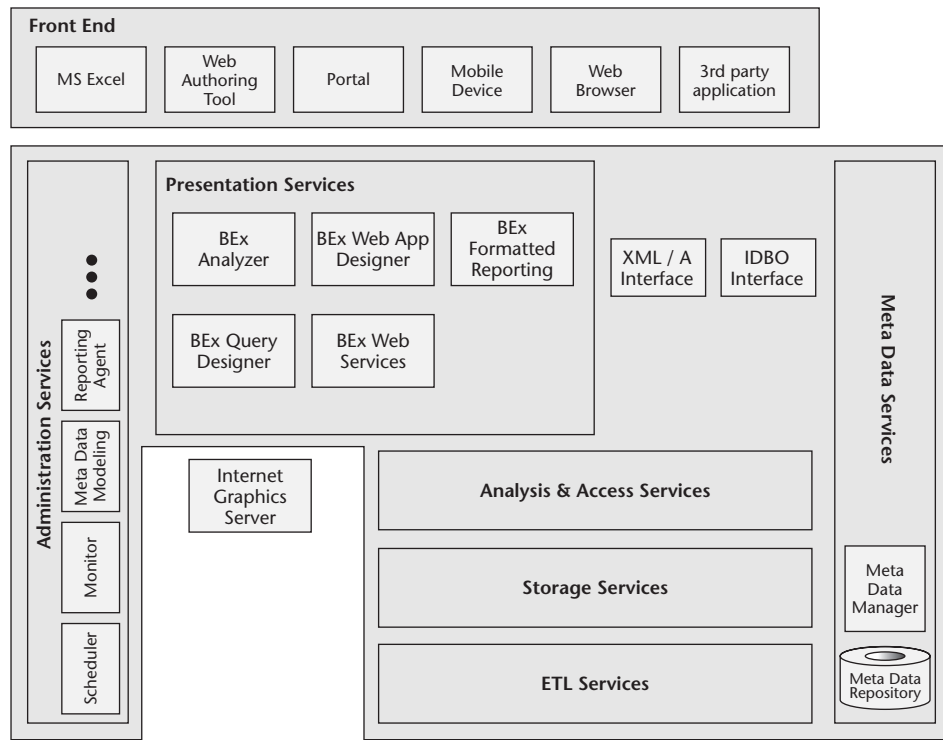
The *content management framework* (CMF) allows you to link documents stored in the SAP Web Content Management Server or any other content server available through the HTTP-based content server interface to SAP BW meta data objects, such as InfoObjects, InfoCubes, and queries, to dynamic query result sets and even single cells of query result sets. This enables you to add additional comments, descriptions, and documentation to these objects. You can access these documents from the Administrator Workbench, the Business Explorer, and from the Web.

The SAP Web Content Management Server stores unstructured information and allows you to find and use this information efficiently. Integration with the SAP BW content management framework provides an integrated view on structured and unstructured information to the end user.

### ***Presentation Services***

The SAP BW *presentation services* layer includes all components required to present information available on the SAP BW server in the traditional Microsoft Excel-based Business Explorer Analyzer (BEx Analyzer), in the BEx Web environment, or in third-party applications. Figure 3.10 provides an overview of the components of the presentation services layer.

## 52 Chapter 3



**Figure 3.10** Presentation services architecture.

### ***BEx Analyzer***

The traditional SAP BW tool for actually invoking multidimensional reporting and analysis in SAP BW is the *BEx Analyzer*. The *BEx Analyzer* is implemented as an add-on to Microsoft Excel, combining the power of SAP BW OLAP analysis with all the features (e.g., charting) and the Visual Basic for Applications (VBA) development environment of Microsoft Excel. Storing query results in Microsoft Excel workbooks, for example, allows you to use information in offline mode, send offline information to other users, or implement complex VBA applications.

**NOTE** You may note that the **BEx Browser** is missing in **Figure 3.10**. While the **BEx Browser** still is a part of the SAP BW offering and still is supported by SAP, many clients have chosen to either start with a Web-based approach or replace the **BEx Browser** by an intranet solution, making the **BEx Browser** obsolete.

### ***BEx Query Designer***

All multidimensional reporting and analysis performed in SAP BW is based on query definitions stored in the Meta Data Repository. Queries provide access to multidimensional information providers (InfoCubes), as well as flat information providers (InfoSets, ODS objects, master data). The *BEx Query Designer* provides easy-to-use yet comprehensive functionality for defining queries in an interactive standalone application.

### ***BEx Web Application Designer***

The *BEx Web Application Designer* is one of the most important additions to SAP BW functionality in the 3.0 release. It allows you to quickly design complex Web pages, including not only the traditional query elements (such as query results and navigation blocks, business charts, and maps) but also interactive components like push buttons and drop-down boxes by simply dragging and dropping the required objects into the layout window, adding some additional text and graphics, adjusting the object properties, and publishing the new page to the integrated Web server. If required, users can also directly manipulate the generated HTML code. Web pages designed with the *BEx Web Application Designer* provide all functionality available in the traditional *BEx Analyzer*.

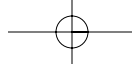
### ***BEx Web Services***

The *BEx Web Services* handle query navigation requests by converting URLs and parameters specified in these URLs into OLAP engine requests and by rendering the data sets returned into query result tables, business charts, maps, or controls supported by the Web application designer toolbox in a device-dependent way. SAP BW application developers no longer have to care about different display properties on different types of devices, such as computer screens, mobile phones, and handheld computers.

Formerly being implemented on top of the *SAP Internet Transaction Server (ITS)*, the *BEx Web Services* have been enhanced significantly and are now integrated into the *SAP Web Application Server*, which is a core part of the SAP BW software.

### ***BEx Formatted Reporting***

Although much of the formatting functionality required can now be provided by the *BEx Web Application Designer*, there still are many applications where reports have to follow specific formatting rules—for instance, for legal reporting in many countries. SAP BW integrates with *Crystal Reports by Crystal Decisions* to provide comprehensive pixel-level formatted reporting functionality on a cell-by-cell basis. Details on formatted reporting can be found in Chapter 7.



## 54 Chapter 3

---

### **Internet Graphics Server**

The *Internet Graphics Server* (IGS) takes care of dynamically rendering graphics to a device-dependent format. The IGS is used to generate interactive business charts and maps based on dynamic SAP BW query for display by the Web services and the Microsoft Excel-based Business Explorer Analyzer.

### **Front-End Tools**

SAP BW allows different types of OLAP front ends to be used. Microsoft Excel can be used in conjunction with the traditional BEx Analyzer discussed previously, while mobile devices and HTML-compliant Web browsers utilize the Web functionality of SAP BW. Web authoring tools can be used to further enhance the look and feel of Web applications—possibly in conjunction with optional Java applets, Java Server Pages, VBScripts, and other technologies supported by modern Web browsers.

Third-party applications either use the ODBO, OLAP BAPI, or XML for Analysis features. Examples of third-party tools optimized for use with SAP BW include Business Objects, Cognos PowerPlay, dynaSight by Arcplan, and others. A complete list of third-party ODBO consumers certified for use with SAP BW can be found on the SAP Service Marketplace (<http://service.sap.com/bw>).

Finally, SAP BW queries may be integrated into any kind of portal implementation, including, of course, the SAP Enterprise Portal offering.

## **SAP BW Meta Data Objects**

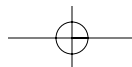
---

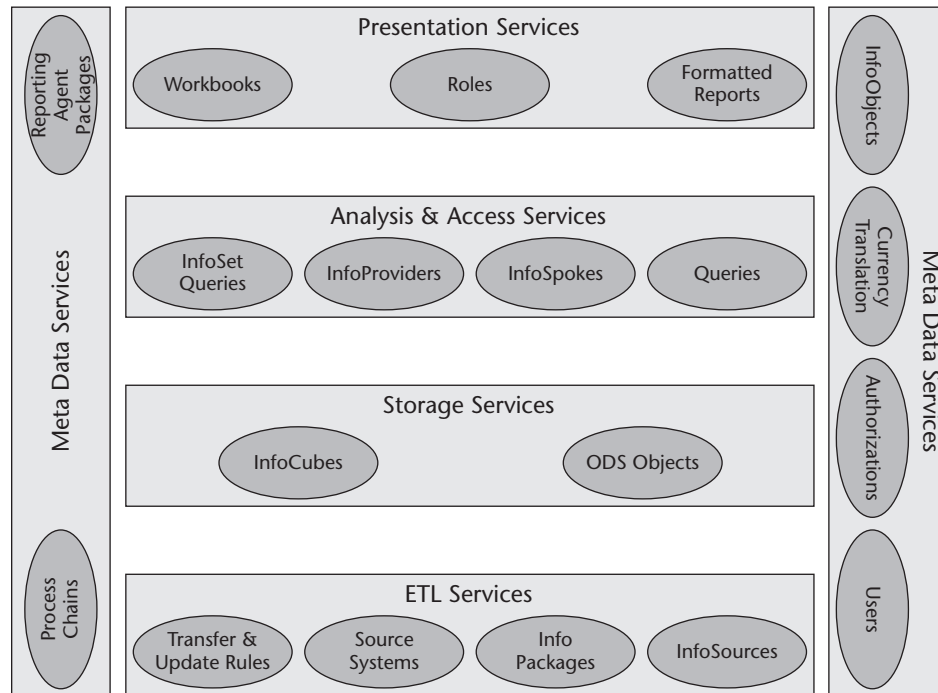
This section provides a definition and a more detailed discussion of all relevant meta data objects available, including InfoObjects, InfoSources, InfoCubes, and queries. Figure 3.11 shows SAP BW meta data objects in the context of the SAP BW architecture.

Besides fundamental meta data needed by data extraction, staging, and analysis processes stored in SAP BW itself, any kind of documentation maintained in the content management framework—such as word processor documents, spreadsheets, and presentations—may be linked to relevant meta data objects (e.g., InfoObjects, InfoSources, InfoCubes, queries) and even dynamic query result sets.

### **InfoObjects**

InfoObjects are the core building blocks for all other data warehouse-related meta data objects in SAP BW, for example, sources of data, analysis structures, and queries. InfoObjects implemented in SAP BW provide a powerful basis for setting up complex information models supporting multiple languages, multiple currencies with automated translations based on the same sophisticated currency conversion rules as in SAP R/3, multiple units of measure, multiple hierarchies, multiple versions of hierarchies of any type, and time-dependent master data.





**Figure 3.11** Meta data objects in context.

An *InfoObject* is the SAP BW representation of the lowest-level business object used to describe business processes and information requirements. There are four types of InfoObjects available in SAP BW: key figures, characteristics, unit characteristics, and time characteristics.

*Key figures* are used to describe any kind of numeric information from the business process level. Low-level numbers such as sales quantities or sales revenues and high-level key performance indicators such as customer lifetime value are all modeled using SAP BW key figures. SAP BW distinguishes six different types of key figures: amount, quantity, number, integer, date, and time key figures:

**Amount.** Key figures of type *amount* are numeric values with an associated fixed or variable currency. SAP BW enforces a consistent representation consisting of both the key figure and the currency through the whole staging and reporting/analysis process. Variable currencies are specified by unit characteristics (see later in this section), whereas fixed currencies are specified by currency codes stored in the InfoObject description.

**Quantity.** Key figures of type *quantity* are numeric values with an associated fixed or variable unit of measure. As with amount key figures, SAP BW enforces a consistent representation of the key figure and the unit of measure. Variable units of measure are specified by unit characteristics (see later in this section), and fixed currencies are specified by codes for units of measure stored in the InfoObject description.

## 56 Chapter 3

**Number.** Key figures of type *number* are used for storing numbers in a floating-point or fixed-point format with no dimensions (currencies or units of weight) associated.

**Integer.** Key figures of type *integer* are used for storing numbers in an integer format with no dimensions (currencies or units of weight) associated.

**Date.** Key figures of type *date* are used for storing date information. In contrast to time characteristics, date key figures can be used for date computations (e.g., actual date - planned date = delay).

**Time.** Key figures of type *time* are used for storing time information. In contrast to time characteristics, time key figures can be used for time computations (e.g., start time - end time = duration).

The properties of a specific key figure stored in the Meta Data Repository include a technical description of the key figure (e.g., the data type) and a business description, such as the unit of measure, currency, aggregation behavior, and display properties used in the Business Explorer.

*Characteristics* are used to describe the objects dealt with in business processes. These can be anything from core business objects like customers, products, and accounts to simple attributes like color, zip code, and status. While key figures from a database point of view simply describe a single field in a database base table, characteristics are more complex. The description of a characteristic includes a field description as it does for key figures, but it may also include the description of a complete set of master data tables storing attributes, texts, and hierarchies associated to that field. An InfoObject definition includes:

- Technical field descriptions such as data type, length, and conversion exits.
- Display properties such as display keys/texts, value help, relevance, and properties for geographical representations.
- Transfer routines that are executed whenever data records referencing this InfoObject are uploaded.
- Master data descriptions such as a list of attributes (which themselves are InfoObjects of any type), time dependency, and navigational properties of attributes, text properties (short, medium, long texts, time and language dependency), properties of hierarchies associated with the InfoObject (time and version dependency, among others), and finally a list of other characteristics used in a compound key for this InfoObject.

A more detailed description of the data model used for storing master data can be found in Chapter 4.

*Unit characteristics* are used to store either currencies or units of measure in conjunction with key figures of type *amount* and *quantity*. Unit characteristics have a reduced set of properties compared with regular characteristics.

*Time characteristics* are used in the obligatory time dimension of InfoCubes to express the time reference of business events. As time characteristics in SAP BW are internally treated in a special way, there is currently no way to create client-specific time characteristics. Time characteristics provided by SAP are shown in Table 3.1.

**Table 3.1** Time Characteristics in SAP BW

TIME CHARACTERISTIC	DESCRIPTION
OCALDAY	Full date in YYYYMMDD format
OCALMONTH	Month in YYYYMM format
OCALMONTH2	Month in MM format
OCALQUART1	Quarter in Q format
OCALQUARTER	Quarter in YYYYQ format
OCALWEEK	Week in YYYYWW format
OCALYEAR	Year in YYYY format
OFISCPER	Fiscal period including fiscal year variant in YYYYMMM format
OFISCPER3	Fiscal period with fiscal year in YYYYMMM format
OFISCVARNT	Fiscal year variant in VV format
OFISCYEAR	Fiscal year in YYYY format
OHALFYEAR1	Half yearQuarter in H format
OWEEKDAY1	Day of week in D format

### **InfoObject Catalogs**

An *InfoObject catalog* is a directory of InfoObjects used in the same business context. Separate types of InfoObject catalogs are used for key figures and characteristics. In addition, InfoObjects can be assigned to several InfoObject catalogs simultaneously.

InfoObject catalogs are very useful in organizing project work in large SAP BW implementations, as there are hundreds of different InfoObjects mostly used in several business contexts (e.g., an InfoObject for products would be used in production, sales, and marketing). There should be two InfoObject catalogs (one for key figures and one for characteristics assigned) defined for every business context, and every InfoObject used in this business context should be assigned to these InfoObject catalogs.

### **InfoCubes**

An *InfoCube* is a multidimensional data container used as a basis for analysis and reporting processes in SAP BW. InfoCubes consist of key figures and characteristics, the latter being organized in dimensions. SAP BW supports two classes of InfoCubes: physical InfoCubes called *basic InfoCubes* and virtual InfoCubes called *remote InfoCubes*. While basic InfoCubes are physically stored in the same SAP BW system as their meta data description, remote InfoCube contents are physically stored on a remote SAP BW, SAP R/3, or third-party/custom system supporting the remote InfoCube BAPI.



## 58 Chapter 3

Basic InfoCubes come in two flavors: standard and transactional. *Standard InfoCubes* are optimized for read access, allowing for scheduled uploads initiated by SAP BW. *Transactional InfoCubes* have been developed for use by applications that need to directly write data into the InfoCube, for example, planning applications such as SAP APO.

Three different types of remote InfoCubes are available in SAP BW as of today:

**SAP remote InfoCubes.** SAP remote InfoCubes refer to sources of data available in SAP R/3 systems through the BW Service API discussed in *the ETL Services* section at the beginning of this chapter.

**General remote InfoCubes.** General remote InfoCubes refer to data stored on a remote system available through the remote InfoCube BAPI. This BAPI is used for third-party and custom data providers.

**Remote InfoCubes with services.** Remote InfoCubes with services refer to data stored on a remote system available through a user-defined function module. This type of remote InfoCube allows flexible user-defined online access to data stored on an arbitrary remote system.

Regardless of which class they belong to, InfoCubes always consist of key figures and characteristics. SAP BW organizes characteristics used in the InfoCube in up to 16 dimensions. Three of these dimensions are predefined by SAP: the time dimension, the unit dimension, and the data packet dimension. You can customize the *time dimension* by assigning time characteristics. The unit characteristics associated to key figures included in the InfoCube definition are automatically added to the *unit dimension*. The *data packet dimension* uniquely identifies data packages loaded into the InfoCube, supporting the data quality efforts of SAP BW administrators.

The terminology SAP uses to describe InfoCubes has caused some confusion in the data warehouse community. In that community, *dimension* is commonly used for what SAP calls a *characteristic* and *dimension* is used by SAP to refer to a collection of characteristics. This explains why a maximum of 13 dimensions in SAP BW is not actually a serious restriction; one single dimension in SAP BW may be composed of more than 250 different characteristics.

InfoCubes can also include *navigational attributes*. Navigational attributes are not physically stored in the InfoCube; instead, they are available through characteristics used in the InfoCube definition. From an end user's perspective, characteristics and navigational attributes are used in exactly the same manner. However, navigational attributes differ from characteristics in two important ways: First, the use of navigational attributes results in slightly more expensive data access paths at query execution time, and second, characteristics and navigational have different semantics in reporting and analysis. For a more detailed discussion of both topics, refer to Chapter 4.

**NOTE** While you cannot define custom characteristics that are treated as time characteristics, you can define characteristics of an appropriate data type and use those to store time references of various kinds. These characteristics cannot be assigned to the standard time dimension but need to be assigned to a custom dimension. See Chapter 4 for a more detailed discussion of characteristics and dimensions.



## Aggregates

Most of the result sets of reporting and analysis processes consist of aggregated data. An *aggregate* is a redundantly stored, usually aggregated view on a specific InfoCube. Without aggregates, the OLAP engine would have to read all relevant records at the lowest level stored in the InfoCube—which obviously takes some time for large InfoCubes. Aggregates allow you to physically store frequently used aggregated result sets in relational or multidimensional databases. Aggregates stored in relational databases essentially use the same data model as used for storing InfoCubes. Aggregates stored in multidimensional databases (Microsoft SQL Server 2000) have been introduced with SAP BW 3.0.

Aggregates are still the most powerful means SAP BW provides to optimize the performance of reporting and analysis processes. Not only can SAP BW automatically take care of updating aggregates whenever necessary (upload of master or transaction data), it also automatically determines the most efficient aggregate available at query execution time. Refer to Chapter 10 for a more detailed discussion of aggregates.

## ODS Objects

An *ODS object* is a flat data container used for reporting and data cleansing/quality assurance purposes. An ODS object consists of key figures and characteristics being organized into key and data fields, where key figures cannot be used as key fields. As with InfoCubes, there are two flavors of ODS objects: standard ODS objects and transactional ODS objects, the latter again allowing for direct updates. Transactional ODS objects are used by planning applications such as SAP APO that need to directly write back forecasts and planning result data.

It is important not to confuse ODS objects with the operational data store (ODS) as defined by Bill Inmon. ODS objects are building blocks for the operational data store—they may be *objects in the ODS*. ODS objects play an important role in designing a data warehouse layer, and from an end-user point of view, ODS objects made available for reporting purposes behave just like ordinary InfoCubes.

A more detailed discussion of the role of ODS objects in the context of the corporate information factory and the differences between ODS objects and InfoCubes can be found later in this chapter. Modeling aspects of ODS objects are discussed in Chapter 4.

## Data Target

A *data target* is a physical data container available in an SAP BW system. Data target is a generic term subsuming basic InfoCubes, ODS objects, and master data tables.

## InfoProviders

An *InfoProvider* is a physical or virtual data object that is available in an SAP BW system and that provides information. InfoProvider is a generic term subsuming all data targets (InfoCubes, ODS objects, and master data tables), in addition to InfoSets, remote InfoCubes, and MultiProviders. InfoProviders are generally available for reporting and analysis purposes.

## 60 Chapter 3

### **MultiProviders**

A *MultiProvider* is a union of at least two physical or virtual InfoProviders available in an SAP BW system. A MultiProvider itself is a virtual InfoProvider.

MultiProviders actually succeed the MultiCube concept of the 2.0 release of SAP BW, which was restricted to defining a union of InfoCubes instead of a union of general InfoProviders. MultiProviders allow combining information from different subject areas on the fly at reporting/analysis execution time.

### **InfoAreas**

An *InfoArea* is a directory of InfoProviders and InfoObject catalogs used in the same business context. Every InfoProvider or InfoObject catalog belongs to exactly one single InfoArea.

InfoAreas in the same way as InfoObject catalogs help organize project work in large SAP BW implementations.

### **Source Systems**

A *source system* is a definition of a physical or logical system providing data to an SAP BW system. Six types of source systems are available:

- SAP R/3-based mySAP.com application components (e.g., SAP R/3, SAP CRM) equipped with the SAP BW extraction program add-n.
- SAP BW-based mySAP.com application components (e.g., SAP BW, SAP APO, SAP SEM) source systems, allowing the user to extract data from other SAP BW-based systems or to extract data from itself.
- Flat-file source systems, used for uploading flat files in ASCII, CSV (comma-separated variables), or binary format.
- DB Connect source systems providing access to external database systems.
- Third-party systems using the Staging BAPI interface; these can either be standard ETL tools supporting the Staging BAPI interface (like Ascential, ETI, or Informatica) or custom programs.
- XML source systems accepting XML data streams.

All types of source systems except the flat-file source system include references to some physical source system or service. The description of a flat-file source system just consists of a name and a short verbal description of the source system; requests for data loads are executed by the SAP BW server itself in this case.

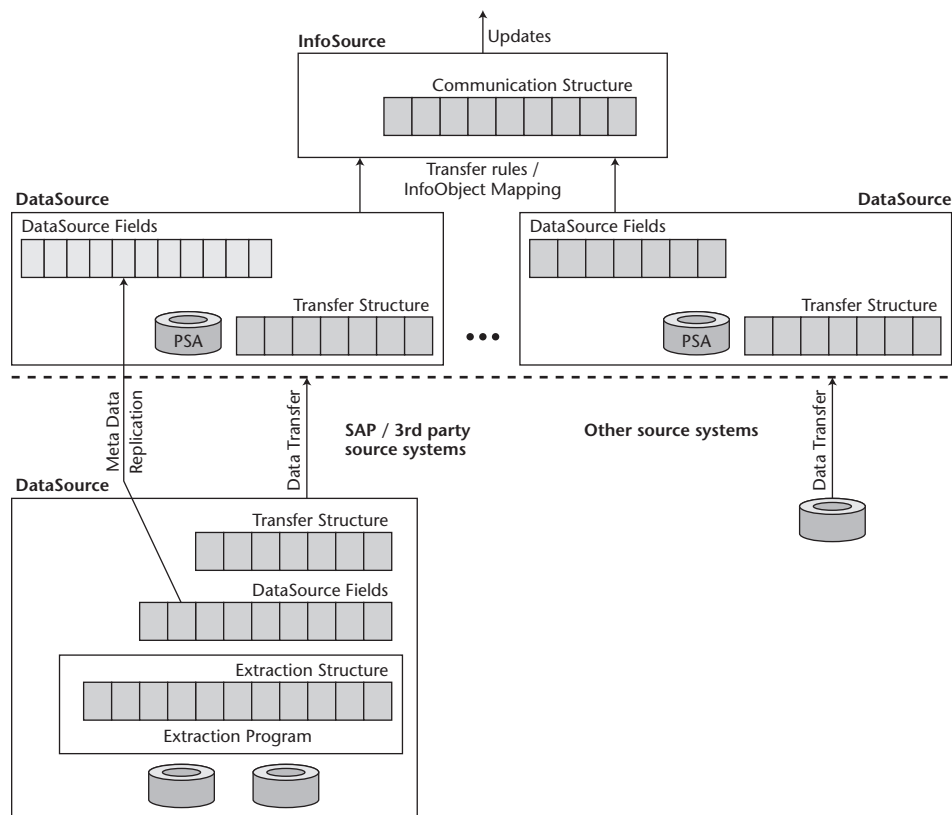
The description of physical source systems includes network or service contact information (such as the RFC destination) to allow SAP BW to automatically connect to the source system and retrieve meta data or request data extractions.

## InfoSources

An *InfoSource* describes a source of business information (business events or business object description) available in one or multiple source systems. The core part of an InfoSource definition is the *communication structure* that is composed of a set of InfoObjects.

An InfoSource is not used to store data. Instead, it is an intermediary between the technical details of the data transfer process and the specific business requirements modeled into the InfoCubes, ODS objects, and master data. Figure 3.12 shows an InfoSource and its communication structure.

SAP BW used to distinguish between transaction and master data InfoSources. Transaction data InfoSources were used for updating ODS objects and InfoCubes, whereas master data InfoSources were used for updating master data tables (attribute tables, texts, and hierarchies). In release 3.0 SAP replaced transaction data InfoSources with a more flexible type of InfoSource capable of staging transaction and master data to all kinds of data targets. For compatibility reasons, master data updates are still supported.



**Figure 3.12** InfoSources and DataSources.

## 62 Chapter 3

### ***Application Hierarchy***

The *application hierarchy* is used to group InfoSources available in SAP BW according to the applications they represent (e.g., SAP R/3 Sales and Distribution).

Just as InfoObject catalogs are useful for organizing InfoObjects, the application hierarchy helps to organize InfoSources. InfoSources cannot be assigned to more than one node in the application hierarchy.

### ***DataSources***

A *DataSource* describes a specific source of data on a specific source system from a technical point of view. The DataSource description includes information about the extraction process and the data transfer process, and it provides the option to store data transferred to SAP BW in the persistent staging area. SAP BW distinguishes between DataSources for transaction data, master data attributes, texts, and hierarchies.

DataSource descriptions are source-system-specific, as different source systems may provide the same data in different specifications, technical formats, or with a different level of detail. Source systems may provide a list of fields available for the DataSource, which may be replicated to the SAP BW Meta Data Repository, as shown on the lower left-hand side of Figure 3.12. Or DataSources may have to be maintained manually, as for DataSources for flat-file source systems (lower right-hand side of Figure 3.12).

Note that regardless of the type of source system, the DataSource definition itself is always controlled by SAP BW, while the extraction process and the technical specifications of the extraction program are defined by the source system.

### ***Transfer Rules***

*Transfer rules* are a set of transformations defining the mapping of fields available in a specific DataSource to the fields used in the InfoSource definition. You create transfer rules by assigning a DataSource to an InfoSource and assigning InfoObjects to the fields of the extract structure (InfoObject mapping). The main purpose of transfer rules is converting the source-system-specific representation of data into an SAP BW-specific view and eliminating technical or semantic differences between multiple source systems providing the same data. Typical transformations used for this purpose include data type conversions, key harmonization, and addition of missing data. Transfer rules allow you to check the data loaded for referential integrity—enforcing that all characteristics values sent by the source system are already available in the corresponding master data tables. In conjunction with the persistent staging area (more on the PSA coming up), you can also use transfer rules to check and ensure data integrity.

SAP BW offers several ways to actually define a specific transformation:

- Simple field assignments, where a field of the transfer structure is assigned to a field of the InfoSource
- Constant value assignment, where a constant value is assigned to a field of the InfoSource
- Formulas, where predefined transformation functions can be used to fill a field of the InfoSource

- Routines, which allow you to implement custom ABAP code for complex transformations

A *transfer structure* is data structure used to describe the technical data format used to transfer data from a source system to an SAP BW system. The transfer structure can be regarded as a contract or an agreement between the SAP BW system and its source system on how to transfer data and what data to transfer. Transfer structures effectively are a projection view upon the fields of the DataSource, as they usually are made up of a subset of those fields.

Multiple DataSources can be assigned to a single InfoSource, allowing you to extract the same kind of data from different source systems (e.g., sales orders from different operational systems used in different regions) or to extract different flavors of the same kind of data from one single source system (e.g., standard material master data and material classification data from an SAP R/3 system). A DataSource can only be assigned to one single InfoSource; assigning a DataSource implicitly assigns a source system to that InfoSource.

The *persistent staging area (PSA)* is a set of database tables for storing data uploaded to an SAP BW system prior to applying transformation rules. The main purpose of the persistent staging area is to store uploaded data for data quality and consistency maintenance purposes. Once stored in the PSA, data is available for multiple updates into multiple data targets at different points of time, avoiding multiple extraction runs for the same set of data.

The PSA can be accessed using a published API and supports error handling and simulation of data updates. A complete error-handling scenario based on the PSA includes identifying and tagging invalid records as part of the upload process, manually or automatically correcting the tagged records utilizing the PSA API, and restarting the upload for the corrected records. The simulation feature includes debugging options and has proved to be helpful in developing transfer and update rules.

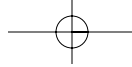
## **Update Rules**

*Update rules* connect an InfoSource to a data target (InfoCube, ODS object, or master data table), allowing it to specify additional transformations from a business point of view.

Update rules establish a many-to-many relationship between InfoSources and data targets. An InfoSource can be used to update multiple data targets, and a data target can be updated from multiple InfoSources. While transfer rules are used to eliminate technical differences, update rules are used to perform transformations required from a business point of view. For example:

- Perform additional data validity and integrity checks
- Perform data enrichment (e.g., adding fields read from master data tables)
- Skip unnecessary data records
- Aggregate data
- Dissociate data provided in a single data record into several records in the InfoCube (e.g., dissociate plan and actual data delivered in one record)
- Convert currency and unit of measure

Update rules support the same types of transformations as transfer rules, plus an automated lookup of master data attributes that, for example, allows you to assign a



## 64 Chapter 3

---

material group value read from the material master data table to the material group characteristic of an InfoCube. Update rules automatically take care of mapping the logical data flow to the physical implementation of the data target, including generation of surrogate keys. For more information, see Chapter 4.

### **InfoPackages**

All scheduling and monitoring functions for data load processes in SAP BW are based on InfoPackages. InfoPackages are defined per DataSource. The following paragraphs present an overview of the properties of an InfoPackage:

**Selection criteria.** Selection criteria are similar to the standard ABAP select options. Fields available in the InfoSource and tagged as selection fields can be used to restrict the set of data extracted from the source system, provided that the source system supports field selections. Selection parameters can be specified as fixed or variable values. Hierarchies do not support selections based on field values; instead, the selection screen for hierarchies allows you to select one of the hierarchies available in the source system for the current InfoSource for upload.

**External filename, location, and format.** These options are available only for uploads from a file source system and specify the details about the file to be uploaded.

**Third-party parameters.** Third-party parameters are those required by the third-party extraction program (ETL tool or custom program). These parameters heavily depend on the actual source system and typically include usernames and passwords.

**Processing options.** Processing options depend on the definition of the transfer rules. If the transfer rules are PSA-enabled, the processing options allow you to specify if and how the PSA should be used during the upload process.

**Data target selection.** Data target selection allows you to select which of the data targets available for the InfoSource should be updated by the upload process and how to handle existing data in the data target (keep data, delete based on selection criteria, or delete all data).

**Update parameters.** Update parameters are used to request full or delta loads and for defining basic error handling parameters.

**Scheduling.** Scheduling parameters allow you to specify exactly when and at what frequency a specific data upload is supposed to be executed. Options for specifying the time of an upload include immediate upload, upload at a specific point of time, upload after completion of a specific job, and upload at a certain event.

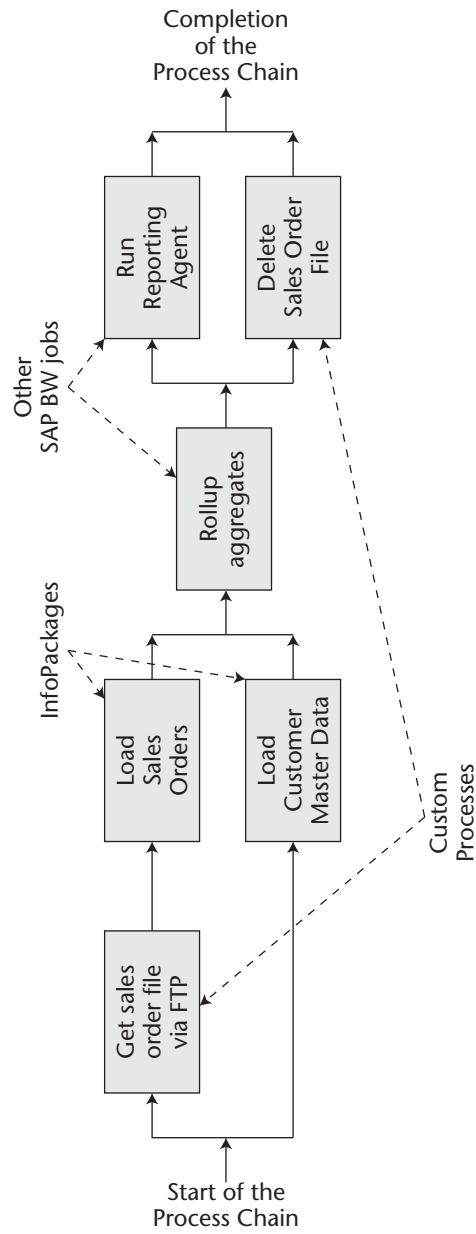


Figure 3.13 Sample process chain.

## 66 Chapter 3

InfoPackages are fully integrated into the SAP BW job control functionality around process chains, discussed in the next section.

### Process Chains

A *process chain* is a defined sequence of interdependent processes required to perform a complex task in an SAP BW environment. Data maintenance tasks in SAP BW are not restricted to uploading data. Aggregate rollups, index maintenance, master data and ODS activation, and a variety of other jobs are required to update data, guarantee best-possible performance, and maintain data integrity. Typical SAP BW implementations have complex interdependent networks of jobs in place that run every night, week, or month. Figure 3.13 shows a simple example of a typical SAP BW job network, including external custom processes, InfoPackages, and other SAP BW tasks. Please note that process chains may also include jobs exporting data using the Open Hub Service.

Previous releases of SAP BW did not provide an integrated solution for scheduling and monitoring those kinds of job networks. This has changed with the introduction of process chains in release 3.0. Process chains allow you to define complex job networks consisting of standard SAP BW jobs, as well as custom jobs; they support visualizing the job network and centrally controlling and monitoring the processes.

While SAP BW still supports the use of the old meta data objects for modeling process meta data (InfoPackage groups and event chains), it includes a tool for migrating those meta data objects to the process chain technology. All new development work should be done using process chains.

### Queries

A *query* is a specification of a certain dynamic view on an InfoProvider used for multi-dimensional navigation. Queries are the basis for all kinds of analysis and reporting functionality available in SAP BW.

Queries are based on exactly one InfoProvider. All characteristics, navigational attributes, and key figures available through that InfoProvider are available for use in query definitions. Because queries are multidimensional objects, they effectively define sub-cubes called *query cubes* on top of the InfoProvider. Query cubes define the degree of freedom available for query navigation in the presentation layer (see Figure 3.14).

A query basically consists of query elements arranged in rows, columns, and free characteristics. While query elements assigned to rows and columns are displayed in the initial query view, free characteristics are not displayed but are available for navigation. Each individual navigational step (drill down, drill across, add or remove filters) in the analysis process provides a different query view. Following are all available query elements:

- A *reusable structure* is a particular commonly used collection of key figures or characteristics stored in the Meta Data Repository for reuse in multiple queries (e.g., a plan/actual variance).
- A *calculated key figure* is a formula consisting of basic, restricted, or other calculated key figures available in the InfoProvider. Calculated key figures are stored in the Meta Data Repository for reuse in multiple queries (e.g., an average discount rate).
- A *restricted key figure* is a key figure with an associated filter on certain characteristic values stored in the Meta Data Repository for reuse in multiple queries (e.g., year-to-date sales of previous year).



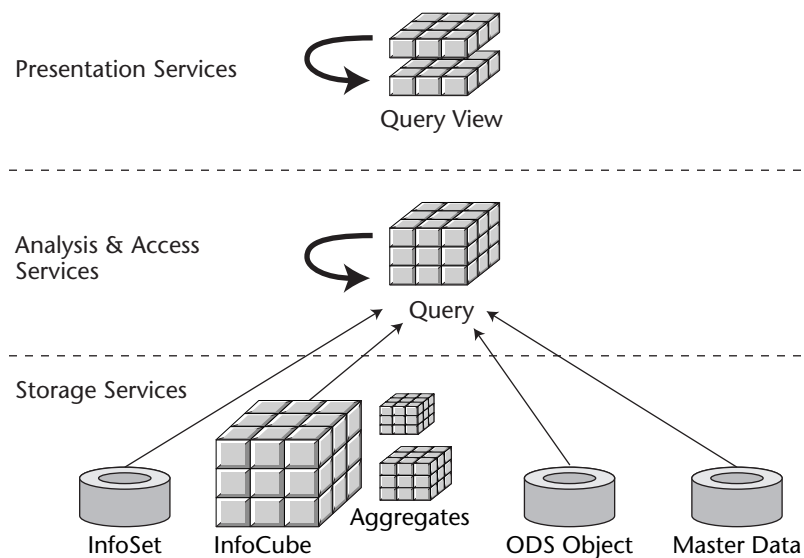
- A *variable* is a parameter of a query. Usually SAP BW determines values of variables at query execution time by running a user exit or requesting user input, but you may also choose to specify constant values as part of the variable definition. Variables are available for characteristic values, hierarchies, hierarchy nodes, texts, and formulas.
- A *condition* is a filter on key figure values with respect to a certain combination of characteristic values.
- An *exception* assigns an alert level from 1 to 9 (1 meaning lowest, 9 meaning highest) to a range of key figure values with respect to a certain combination of characteristic values. Alerts can be visualized in queries or in the alert monitor and can be used to automatically trigger a workflow (e.g., by sending an email).

As Figure 3.14 shows, queries are not device- or presentation-tool-dependent. The same query definition may be used by the BEx Analyzer, in a Web environment, on a mobile device, for batch and exception reporting in the reporting agent, in formatted reporting, and in a third-party presentation tool.

Query definitions are created and maintained in the graphical Query Designer by simply dragging the available query elements into the rows, columns, free characteristics, or filter area and eventually defining additional properties. The Query Designer also integrates all functionality required to define the query elements in the preceding list.

### Query Workbooks

A *query workbook* is a standard Microsoft Excel workbook with embedded references to query views and optional application elements built using Microsoft Excel functionality (e.g., business charts or graphical elements such as push buttons and list boxes) and Visual Basic for Applications (VBA) code.



**Figure 3.14** Queries and navigation.

## 68 Chapter 3

---

Using Microsoft Excel as one of the query execution options in SAP BW allows you to combine the functionality of multidimensional analysis on top of a data warehouse solution with the functionality of the Microsoft Excel. In addition to the application development functionality mentioned in the preceding definition, workbooks allow for using query results (and the applications built on top of that) embedded in a query workbook offline or for distributing the query workbooks to a bigger audience via email or other file distribution mechanisms.

### **Reporting Agent**

*Reporting agent settings* define the details of a particular activity performed by the reporting agent. Possible activities include printing query results in batch mode, identifying exception conditions and eventually triggering follow-up events, calculating Web templates, and calculating value sets for use with query variables. *Reporting agent scheduling packages* are used to schedule the execution of a specific reporting agent setting.

For batch printing of query results the reporting agent settings include selecting a query and designing the page layout (cover sheets, table and page headers, and footers) and query properties. Exception reporting requires selection of a query, an associated exception, and follow-up activities. Possible follow-up activities include sending email messages and adding entries to the alert monitor. Calculating Web templates requires specification of a Web template and a query. Calculating value sets requires specification of a characteristic and a query used to calculate the values for the value set.

### **InfoSets**

An *InfoSet* is a virtual InfoProvider implementing an additional abstraction layer on top of the SAP BW Meta Data Repository. InfoSets allow defining joins of multiple ODS objects and master data tables using the InfoSet Builder. An SAP BW InfoSet differs from classical InfoSets known from other mySAP.com application components in that they are specially designed to support SAP BW meta data objects.

While SAP BW 2.0 only supported defining BEx queries for ODS objects, release 3.0 provides a lot more flexibility because it generalizes the different data targets (InfoCubes, ODS objects, and master data tables), introducing the InfoProvider concept and extending this concept by adding InfoSets to the list of meta data objects available for reporting purposes.

Keep in mind that InfoSets neither replace MultiProviders nor are MultiProviders designed to replace InfoSets. MultiProviders implement a union of several InfoProviders of all types, while InfoSets provide joins of ODS objects and master data tables but do not support InfoCubes.

### **Open Hub Destination**

An *open hub destination* is a logical target system defining the technical details required to export data from SAP BW using the Open Hub Service. Open hub destinations are available for exporting data to flat files or database tables, or directly to an application.

The definition of an open hub destination includes a logical target system name and detail information about the data target, for example, name and format of the export or name of a database table.

### **InfoSpokes**

*InfoSpokes* are the core meta data objects of the Open Hub Service. An InfoSpoke definition is composed of a data source definition that refers to an InfoProvider, a set of selection and projection (selection of columns) criteria, and a set of simple mapping rules. InfoSpokes are the outbound counterpart of (inbound) InfoSources, mapping a business view available as an InfoProvider back to the technical specification of an outbound data interface. An InfoSpoke may have multiple open hub destinations assigned, allowing for different physical data targets. This again resembles the InfoSource concept to some extent.

InfoSpokes are fully integrated into the scheduling and monitoring functionality of SAP BW. InfoSpokes have become generally available with release 3.0B of SAP BW. Further development is expected in this particular area, so watch for updates on the accompanying Web site.

### **Users**

*Users* are individuals or automated processes that have a unique identifier allowing them to log on to and to use a specific SAP BW system. Automated processes in SAP BW are used to load data into an SAP BW system and to extract information from the SAP BW system for further processing.

### **Authorizations**

An *authorization* warrants a specific user the right to perform a specific action or retrieve a certain bit of information from an SAP BW system. SAP BW utilizes the technical infrastructure known from SAP R/3 for implementing its own specific authorization concept. These technical foundations are discussed later in this chapter. A more detailed description of SAP BW authorizations can be found in Chapter 9.

### **Roles**

As implemented in SAP BW, the *role* concept resembles the role or function individuals have in an organization. Role definitions in SAP BW are composed of a collection of menu items (referring to queries, transactions, and documents), authorizations, iViews, and a set of users assigned to this role.

Examples of such roles include a purchasing manager role, a sales representative role, and the CEO role. In the same way as in an organization, roles can be assigned to multiple individuals simultaneously (such as there may be multiple sales representatives), and the assignment of roles to individuals may change over time without affecting the definition of the role itself (a purchasing manager will always be expected to manage the purchasing process regardless of the individual filling that role).

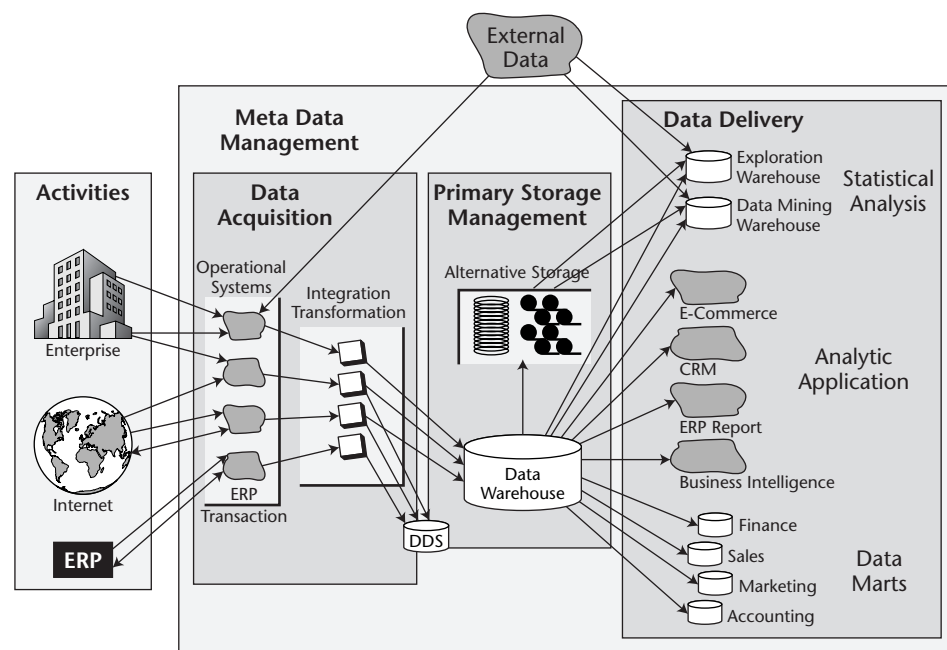
## 70 Chapter 3

### Currency Translation Types

Currency translation types are used to define how to convert currencies from a source to a target currency and which currency exchange rates to use for this conversion in the update rules or in reporting and analysis processes. Many OLAP tools and data warehouse solutions currently available only provide poor or actually no support for handling multiple currencies, although for most companies running a data warehouse, multiple currencies are everyday business in many business processes. SAP BW again utilizes existing SAP technology for the provision of currency translation mechanism and even allows synchronizing currency conversion types as well as conversion rates with existing SAP R/3 systems.

### Mapping the Corporate Information Factory to SAP BW components

Before we actually start laying out the options, methods, and tools available in SAP BW to implement the best-possible solution for analysis and reporting, let's first focus on the architectural layers of an SAP BW implementation along the lines of the CIF defined by Bill Inmon (see [www.billinmon.com](http://www.billinmon.com) or *Corporate Information Factory*, 2nd Edition, Wiley). (See Figure 3.15.)



**Figure 3.15** The corporate information factory.

Copyright © 2001 Billinmon.com LLC

Although SAP BW 3.0 provides meta data objects, tools, and methods allowing us to implement nearly all components of the CIF, the terminology used by SAP does not exactly match the terminology defined and used by Bill Inmon—especially in the primary storage management and data delivery layers.

The data acquisition layer is now completely covered by the SAP BW Staging Engine and by partnerships with ETL tool vendors like Ascential Software, Informatica, and others (the staging process is discussed in more detail in Chapter 6). Meta data management is handled by the Meta Data Repository, along with the Administrator Workbench, as a tool to view and modify SAP BW meta data.

One terminology mismatch is related to the ODS, which is defined by Bill Inmon as follows (see [www.billinmon.com](http://www.billinmon.com) or *Building the Operational Data Store*, 2nd Edition, Wiley):

*The Operational Data Store (ODS) is a hybrid structure that has characteristics of both the data warehouse and operational systems. Because the ODS is a hybrid structure, it is difficult to build and operate. The ODS allows the user to have OLTP response time (2-3 seconds), update capabilities, and decision support systems (DSS) capabilities.*

Bill Inmon distinguishes four types of operational data stores:

**Class I.** The time lag from execution in the operational environment until the moment that the update is reflected in the ODS is synchronous (i.e., less than a second)

**Class II.** The time lag from execution in the operational environment until the moment that the update is reflected in the ODS is in the 2- to 4-hour range (i.e., in a store-and-forward mode)

**Class III.** The time lag from execution in the operational environment until the moment that the update is reflected in the ODS is overnight (i.e., in a batch mode)

**Class IV.** Data is processed in the data warehouse and fed back to the ODS in an aggregated manner

SAP BW provides remote InfoCubes and ODS objects—nontransactional or transactional—to model the ODS layer of the CIF. Keep in mind that ODS objects are completely different concepts: ODS objects are meta data objects providing a certain functionality, whereas the ODS is an architectural layer in a data warehousing framework.

The data warehouse part of the corporate information factory is defined by Bill Inmon as follows (see [www.billinmon.com](http://www.billinmon.com) or *Building the Data Warehouse*, 3rd Edition, Wiley).

*The data warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data used to support the strategic decision-making process for the enterprise. It is the central point of data integration for business intelligence and is the source of data for the data marts, delivering a common view of enterprise data.*

The meta data object of choice for modeling the data warehouse layer in SAP BW is the ODS object, now equipped with complete archiving functionality.

Looking at the data delivery layer, we see there is an exploration warehouse, a data mining warehouse, analytical applications, and data marts. While exploration warehouses, data mining warehouses, and data marts can be built with SAP BW functionality using ODS objects, InfoCubes, and the data mining functionality/interfaces, analytical applications are usually built on top of the core functionality of SAP BW,

## 72 Chapter 3

utilizing open interfaces for data and meta data exchange and the integrated ABAP Workbench. To draw a line between the data delivery layer as defined in the CIF and the information modeling options in SAP BW that may be used to implement the data delivery layer, we use the term *InfoMart*.

An InfoMart has the following properties:

- It is dynamic and disposable.** InfoMarts may but do not have to be rebuilt dynamically or even disposed of, following adjustments driven by changing business environments. New or changed InfoMarts can be created very easily based on the data stored in the data warehouse layer.
- It is volatile.** Data in an InfoMart may or may not be updated depending on the analytical application. Pure reporting and analysis InfoMarts will be nonvolatile; InfoMarts used in other types of applications, such as planning (e.g., SAP SEM, SAP SCM) will be volatile.
- It is a subset of information for a specific audience.** InfoMarts focus on the reporting and analysis requirements of a specific, possibly cross-business-area audience inside or outside the organization and provide the subset of information required for this audience.
- It is persistent or virtual, multidimensional, or flat.** InfoMarts can be built using persistent (InfoCubes, ODS objects) or virtual InfoProviders (MultiProviders, remote InfoCubes) using multidimensional (InfoCubes) or flat (ODS objects) data models.
- It focuses on reporting and analysis.** InfoMarts are used primarily for reporting and analysis services, including analytical applications.

Table 3.2 summarizes the differences between InfoMarts and data marts.

**Table 3.2** Characteristics of InfoMarts and Data Marts

CHARACTERISTIC	INFOMART	DATA MART
Dynamic	Yes	No
Disposable	Yes	Yes
Volatile	Yes	No
Nonvolatile	Yes	Yes
Flat	Yes	No
Dimensional	Yes	Yes
Virtual	Yes	No
Specific audience	Yes	Yes
Focus on reporting and analysis	Yes	Yes
Exploration/mining	Limited	Limited

**Table 3.3** ODS Objects versus Remote Cubes for Building the ODS

CHARACTERISTIC ODS OBJECT	NONTRANSACTIONAL ODS OBJECT	TRANSACTIONAL CUBES	REMOTE
Redundancy	Yes	Yes	No
Class I ODS	No	Possible	Yes
Class II ODS	Yes	Possible	Possible
Class III ODS	Yes	Possible	Possible
Class IV ODS	Yes	Possible	Possible

## The Operational Data Store

SAP BW offers two options when you are modeling the ODS layer of the CIF: ODS objects and remote InfoCubes. While ODS objects are physically stored copies of data in the SAP BW system, remote InfoCubes are references to data records stored on a separate system. Queries against ODS objects are executed on the data warehouse system; queries against remote cubes are executed on remote systems.

ODS objects come in two flavors: nontransactional and transactional. Because ODS objects provide OLTP response times for OLTP-type queries, allow updates, and provide DSS functionality, they are ideal candidates to build the operational data store. Nontransactional ODS objects (the traditional ODS objects) are updated in batch mode based on a schedule defined by an administrator. Transactional ODS objects (new with release 3.0) may be updated in real time.

Nontransactional ODS objects can be used to build operational data stores of classes II, III, and IV. You can use transactional ODS objects to build class I (as well as II, III, and IV) operational data stores, provided the OLTP system has a real-time change data capture queue and allows you to automatically propagate updates to OLTP tables to this ODS object in real-time.

Another option to simulate rather than implement class I operational data stores is using a remote InfoCube referencing to the OLTP table. Remote InfoCubes fulfill the class I ODS requirements by mimicking multidimensional analysis on operational data, avoiding redundant storage in the data warehouse system through the remote InfoCube interface. Table 3.3 provides a summary of the preceding discussion.

## The Data Warehouse Layer

The main function of the data warehouse layer of the CIF is to provide an integrated history of data relevant for business decisions. The most important aspects of integration are:

**Selection.** Not all available data is relevant to information processing.

**Harmonization.** Data type conversions, calculations, and unification of technical values representing a property of an object, for example.



## 74 Chapter 3

**Data quality.** Add missing information (default values or derived values) and plausibility checks.

**Time.** Add timestamps.

**Aggregation.** Aggregation of data where the level of detail provided by a source system is too high.

Looking at these characteristics of integrated data, we see that modeling the data warehouse layer means defining the foundations for the corporate information infrastructure—what data at what level of granularity is available to fulfill today's and future reporting and analysis requirements.

The main SAP BW meta data object available for modeling the data warehouse layer is the ODS object already discussed in the previous section on modeling the operational data store. In releases prior to 3.0, the ODS object, not being enabled for master data, did not fully cover the requirements of a data warehouse.

### The InfoMart Layer

The final—and from an end user's point of view most important—layer of the CIF is the data delivery layer, which we refer to as the *InfoMart layer*. Most of the information available to end users is available on this level.

While InfoCubes still are the most important SAP BW meta data objects when it comes to delivering reporting and analysis functionality to end users, power users, and analytical applications, there are applications for ODS objects on this level, especially in multilevel staging scenarios where data from different applications areas need to be integrated into a single InfoProvider and for simple reporting applications. An overview of the differences between ODS objects and InfoCubes is shown in Table 3.4.

**Table 3.4** Differences between ODS Objects and InfoCubes

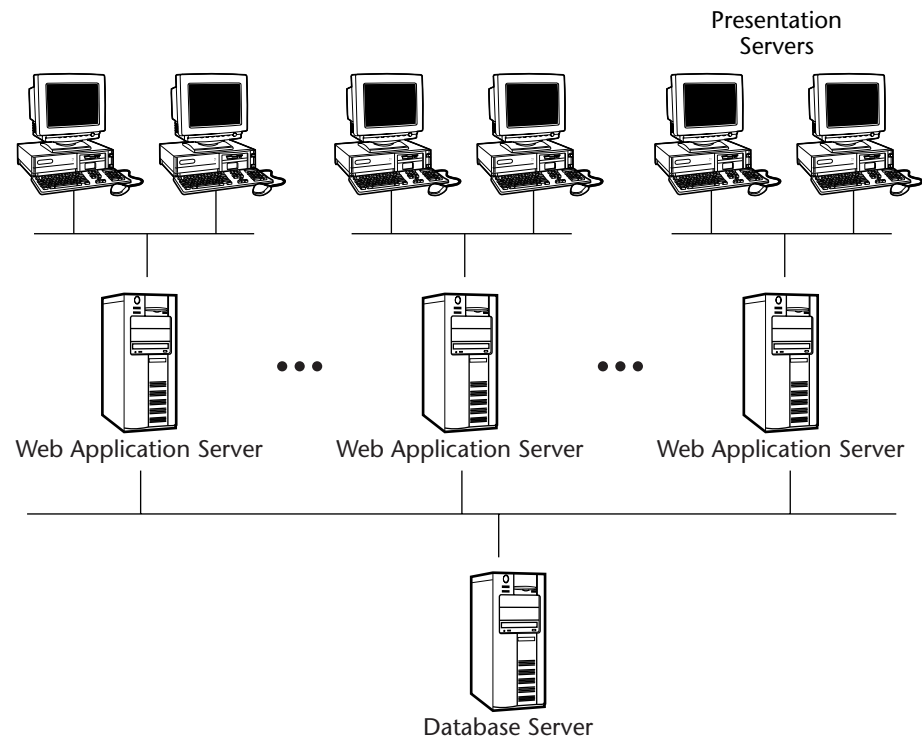
PROPERTY	ODS OBJECT	INFOCUBE
Architecture	Flat database tables	Extended Star Schema (ESS)
Support for granular data	Yes	Yes
Staging for transactional data	Yes	Possible
Staging for master data	Yes	No
Update of key values	Not possible	Not possible
Update of characteristics	Possible	Not possible, due to ESS
Update of key figures	Possible	Only for additive key figures
Change log	Tracks every change	Tracks new records only
Support for BEx queries	Yes (medium performance)	Yes (good performance)
Support for InfoSets	Yes	No



Given the user and application requirements, there still is no silver bullet for identifying what InfoCubes or ODS objects are required or how exactly to lay out the InfoMart. The best advice we can give without knowing about specific requirements is to go for rapid, iterative development cycles as described in Chapter 2 when planning for an SAP BW implementation project. Besides mistakes possibly made in the initial requirements collection phase or the design phase, requirements tend to increase and change rapidly after going live, as new information retrieved from the system often changes the focus of business analysis and raises curiosity to look at business processes from a different angle or by a changing business environment. Having a sound data warehouse in place ensures that you can adjust InfoMarts within a reasonable amount of time and effort and be able to meet future requirements.

## The Architectural Roots of SAP BW

SAP was among the first software development companies to fully adopt a multi-tier client/server model (see Figure 3.16) and move away from traditional host-based solutions for business software. This was the beginning of the SAP R/3 success story back in the early 1990s.



**Figure 3.16** SAP multi-tier architecture.

## 76 Chapter 3

To fully utilize the existing R/2 business applications—to a large extent implemented in the ABAP programming language—SAP had not only to implement a high-performance ABAP development and runtime environment but also a lot of tools that had been available on host computers at that time: transaction-based application server concept, job scheduling, and monitoring tools; a development environment; secure communication software; a business-driven authorization concept; a database abstraction layer; and so forth.

The result of this development process was the SAP Basis software, or Basis Component (BC). All SAP application modules—such as FI, CO, and SD—were originally developed using the BC functionality. Additional common functionality such as Application Link Enabling (ALE), Interchangeable Documents (IDocs), Business Application Programming Interfaces (BAPIs), handling of currencies and units of measure, documentation, and translation tools were developed separately and were distributed in a separate application component called Cross Application (CA).

There are some common misunderstandings about SAP BW being just another application module comparable to FI, CO, or SD, or about being able to install SAP BW as an add-on to SAP R/3 or SAP R/3-based systems like CRM. While SAP BW may be installed on the same physical server, it always has to be installed as a separate SAP instance and will always use its own separate database and its own separate application servers. SAP started developing the SAP Business Information Warehouse in late 1996 based on an SAP system with just the basis and cross-application components installed and with a completely separate architecture, optimized for reporting and analysis purposes, in mind.

Recent developments of the SAP Basis software toward Internet technologies finally resulted in changing the name from SAP Basis Component to *SAP Web Application Server*. This evolution is shown in Figure 3.17, focusing on the most relevant developments.

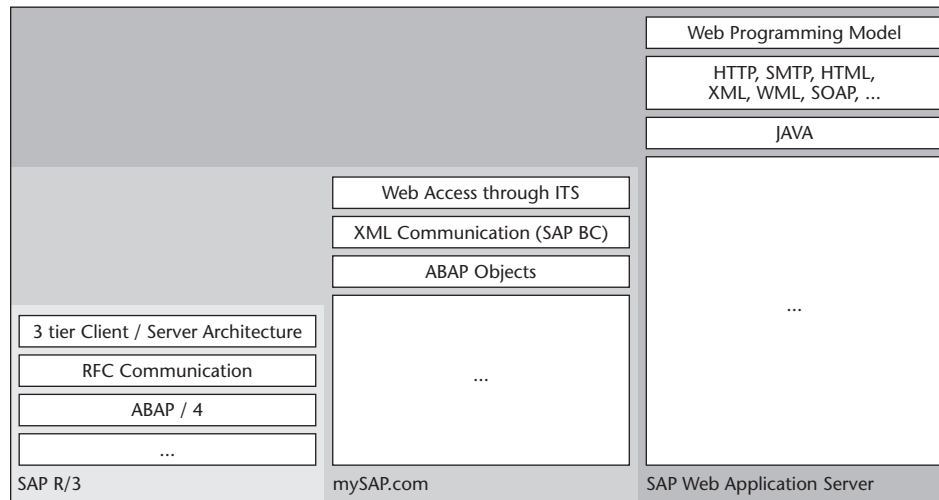
The traditional SAP R/3 Basis component allowed you to develop applications using the ABAP programming language and had been able to communicate with other systems using a proprietary protocol named *Remote Function Calls* (RFC). The mySAP.com platform included Web awareness through the Internet Transaction Server (ITS) for the first time, added XML support through the SAP Business Connector, and allowed for object-oriented software development using ABAP objects.

While the ITS and the SAP Business Connector have been separate software systems interfacing with core R/3 functionality, today the SAP Web Application Server provides an integrated server platform, fully aware of all relevant protocols and standards (such as HTTP, SMTP, XML, SOAP, and .NET) and allows application development using Java (including Enterprise JavaBeans; Java 2 Enterprise Edition, or J2EE; and Java Server Pages) and ABAP in all its flavors (ABAP, ABAP Objects, Business Server Pages).

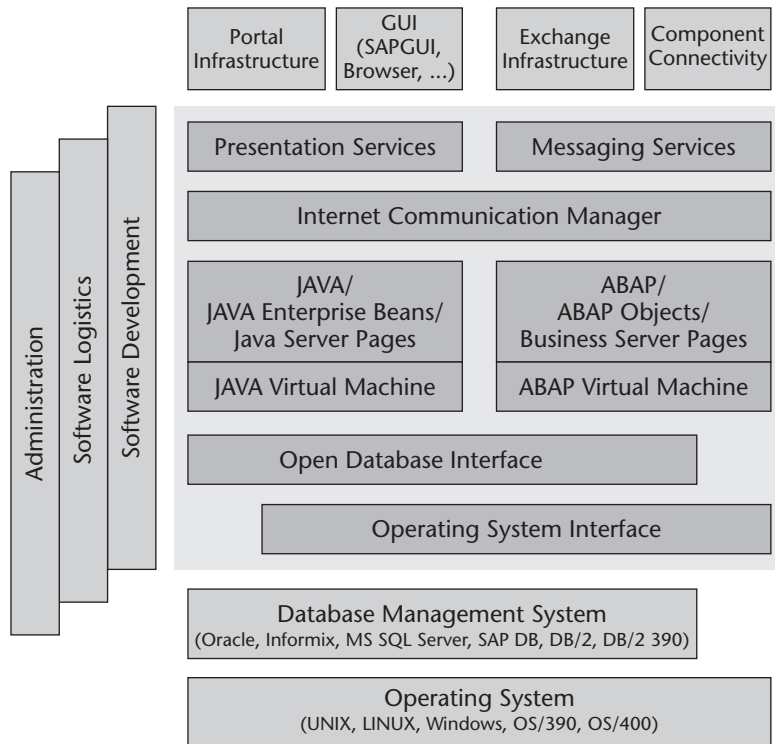
### SAP Web Application Server Architecture

The SAP Web Application Server is no longer just a platform for all SAP applications; it has now evolved into a serious multipurpose business application development and runtime environment with its own complex architecture, shown in Figure 3.18.

**SAP Business Information Warehouse Architecture 77**



**Figure 3.17** SAP application server evolution.



**Figure 3.18** SAP Web Application Server architecture.

## 78 Chapter 3

---

Going into the details of the SAP Web Application Server architecture is beyond the scope of this book. The following paragraphs provide an overview of the most important components from an SAP BW point of view.

### **Core Components**

The *operating system interface* allows the SAP Web Application Server to be installed on several different hardware and operating system platforms including UNIX/Linux, OS/390, OS/400, and Microsoft Windows. It hides the details of the different operating systems and provides an abstract layer for access to operating system functionality. The operating system interface provides shared process services, including dispatching of application requests, shared memory services, and synchronization services (enqueue/dequeue).

The *open database interface* enables SAP BW to utilize database functionality from different vendors, including Oracle, Informix, IBM, and Microsoft, as well as the SAP DB database offered by SAP. Besides hiding the implementation details of different database systems, the open database interface allows application-specific caching of database requests and provides buffer synchronization between multiple application servers using the same database server. The special multidimensional requirements of SAP BW provoked some significant enhancements of the open database interface and forced the SAP BW development team to develop its own additional database abstraction layer on top of the open database interface.

The *Java Virtual Machine* supports the execution of Java programs, Java Server Pages (JSPs), and JavaScripts and integrates with Enterprise JavaBeans. With the integration of the Java Virtual Machine into the SAP Web Application Server, SAP opened its platform to a whole new world of application developers.

The *ABAP Virtual Machine* has been in place right from the beginning of SAP R/3 development. Originally a functional programming language, ABAP has been extended by object-oriented software development features (ABAP objects) and Web-enabled objects (Business Server Pages, or BSPs) in the last couple of years. The ABAP Virtual Machine precompiles the ABAP code into a byte code, which is then stored in the database and used for execution. While most of the core business functionality of SAP R/3 and SAP BW is and will be implemented in ABAP, a lot of the more front-end-related development will be done in Java.

The *Internet Communication Manager* provides support for open Internet standards and protocols, including HTTP, SOAP, HTML, XML, and WML, as well as traditional SAP communication protocols such as RFC. The Internet Communication Manager has an integrated Web server, allowing external applications and front-end systems to use the HTTP protocol for communicating with the SAP Web Application Server.

The *presentation services* integrate with the SAP Enterprise Portal infrastructure, offering iViews and other portal infrastructures from third-party vendors. The presentation services support several types of front-end systems, including the traditional SAPGUI for Windows, the SAPGUI for Java, and the SAPGUI for HTML. The *messaging services* allow exchanging data with other applications using the SAP protocols and open protocols like the SMTP and SOAP.

## Software Development

One of the key success factors for SAP R/3 and SAP BW has been the integrated software development tools that allow customers to adapt the system to their specific requirements by implementing their own custom ABAP programs or by modifying programs delivered by SAP. The software development platform of SAP, called the ABAP Workbench, integrates various editors such as an ABAP editor, a screen designer, and a report designer with a data dictionary, allowing you to share common definitions of data types and table structures. Debugging functionality is available, allowing you to debug both custom and SAP code. Open interfaces—the BAPIs—allow access to SAP functionality by reading and writing data and meta data and by executing business processes.

SAP currently does not offer its own Java development environment; instead, it integrates with third-party development environments.

## Software Logistics

The SAP Web Application Server includes sophisticated software logistics support for software development objects, meta data, and customization data based on the *Transport Management System (TMS)*. The TMS performs the following major tasks around managing software development and distribution:

- It tracks all changes to development objects under its control, whether these are delivered by SAP or have been developed by the client. Objects under control of the TMS include programs, database tables, all BW meta data objects, and customizing data.
- The TMS provides sophisticated software distribution mechanisms to manage complex application development landscapes, including separate multistaged development, test, and production systems. More information about system development landscapes appears later in this chapter.
- It allows upgrading running systems and applying support packages; it automatically identifies modified objects and allows you to manually handle modifications during the upgrade or update process.

More information about the transport management system can be found at the end of this chapter.

## Security

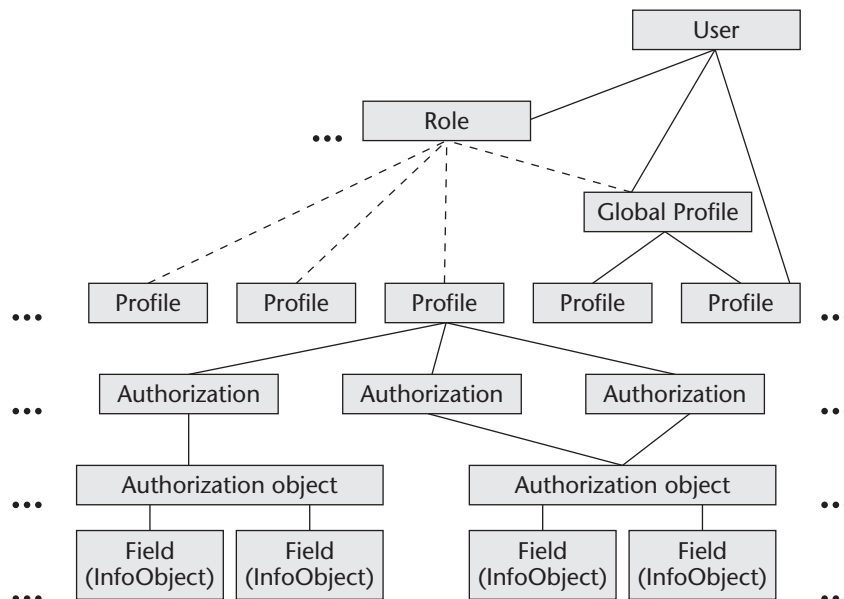
A key issue often neglected—or at least not implemented at a sufficient level of sophistication—in custom data warehouse solutions and sometimes even standard data warehousing tools is security. Protecting the information in the data warehouse is as important as protecting the data in your operational system against unauthorized access. SAP BW security relies on the SAP Web Application Server, which uses Secure Network Communications (SNC) to provide single sign-on and centrally managed LDAP-based user stores.

## 80 Chapter 3

The SAP authorization does not simply rely on authorization concepts provided by operating systems and database systems. Instead, it comes with its own application authorization concept, allowing for very detailed adjustments of authorizations in operational and data warehouse systems to the policies of any organization. Authorization checks are executed and defined by the SAP application using application server functionality.

To ease the task of administering a large amount of users on that detail level, the whole authorization concept is role-based (see the subsection *Roles* in the *SAP BW Meta Data Objects* section earlier in this chapter for a definition of *roles*). Roles can consist of several profiles, which basically are collections of authorizations required to perform a specific task. Roles may be assigned to multiple users, and users may have different roles assigned. Profiles may include other profiles and normally do include a couple of authorizations. Each authorization is an instance of an authorization object, describing exactly which operations are allowed for a certain object. Figure 3.19 provides an overview of authorization.

**NOTE** Roles are not only used to collect all authorizations required to execute on a specific business role; they are also used to provide easy access to menus or complex personalized applications in a portal environment.



**Figure 3.19** The SAP authorization concept.

Based on copyrighted material from SAP AG

While the authorization concept implemented in SAP BW allows you to define authorizations at a very granular level, we recommend keeping authorization models as simple as possible but as concise and restrictive as necessary. The more granular a model you choose, the more resources it will take to maintain the actual authorizations. Key to a simple authorization model is a proper set of naming conventions, as all objects that are effectively placed under control of the authorization concept have to be named. The more generic a name you can provide, the less effort you have to spend on ensuring the integrity of your authorization model.

The SAP authorization concept and strategies for implementing a customer authorization concept are discussed in more detail in Chapter 9.

### **Administration**

SAP administration covers the whole range of functionality required to run complex online applications and batch processes—including job control and job monitoring, user and authorization management, performance monitoring and performance optimization, output management, and archiving.

Most of the functions related to the more technical aspects of administering an SAP system are integrated in the *computing center management system* (CCMS); frequently used functionality such as printing and job scheduling and monitoring are available from nearly all transactions.

Chapter 9 covers general administration strategies in greater detail, and Chapter 10 includes a discussion of the CCMS functionality for performance monitoring and optimizing an SAP system (especially an SAP BW system).

### **Additional Functionality**

A lot of functionality for many different purposes has developed around the core of the SAP Web Application Server and is also available to SAP BW users. While there's not enough room in this book to describe all of these, we'll provide an overview of the most important functions.

*Content management* is generally available in SAP systems and has already been discussed briefly in the *SAP BW Architectural Components* section at the beginning of this chapter. SAP BW integrates access to the content management framework by linking meta data objects to documents available on the content server.

*Workflow management* functionality allows you to create and manage workflows in an SAP system. SAP BW currently uses workflow functionality in exception reporting to initialize workflows based on exceptional key figure values identified. It is also used in process monitoring where processes may kick off an alert workflow in case of failed processes or process chains to immediately notify administrators and keep track of the actions taken to recover from the problem.

## 82 Chapter 3

The *Computer-Aided Test Tool* (CATT) supports testing applications of any kind by allowing you to define and—as far as possible—to automate tests of newly developed software, of customizing activities, and of other development results.

Development of multilingual applications is supported by *documentation and translation tools*, by a general distinction between master data and text data (the latter being language dependent), by integrated support for the Unicode standard, and by support for separate language imports by the transport management system.

Sophisticated *currency conversion services* allow manual and automated maintenance of currency translation rate tables and provide different currency conversion procedures that take different currency conversion regulations (such as euro conversion) into account. Currency conversions can be performed with respect to specific points or periods of time, allowing you to convert currencies according to current exchange rates, historical exchange rates, average exchange rates, and statistical exchange rates. SAP BW offers integrated support for currency conversions in the update rules and in reporting and analysis.

SAP systems come with predefined and customized tables defining *units of measure* and conversions between different units of measure.

Integrated country- and client-specific *calendar functionality* is available for defining work days, which then can be used as a basis for scheduling jobs and calculating days of work in HR applications.

### System Landscape Considerations

It is good practice to keep separate instances of the system—especially for development, testing, and production purposes. SAP systems (including SAP BW) have always supported these activities with the Transport Management System introduced earlier this chapter. The TMS captures changes in many types of objects, including:

- All SAP BW meta data objects, including InfoObjects, InfoCubes, InfoSources, Data Sources, queries, Web applications, and process chains
- All programming objects, including ABAP programs, function groups, types, classes, includes, and messages
- All dictionary objects, including tables, views, data elements, and domains
- All customization tables, including currency translation types, application components, printer definitions, user profiles, authorizations, profiles, and calendars
- All meta data of other SAP systems such as SAP CRM, SAP SEM, and SAP APO

All development objects logically belonging together are assigned to *packages* (formerly known as development classes). These objects are stored in some table in the SAP database; the TMS now keeps track of changes by simply assigning the key values of such an object to a task, which itself is assigned to a transport request. Once all tasks are completed and released by the users assigned to the tasks, the request is released—effectively exporting the current versions (all table contents, not just the keys) of all objects tracked in that request into a flat file. Using the TMS, you can import this flat file into any SAP system (be careful not to import SAP R/3 requests into an SAP BW

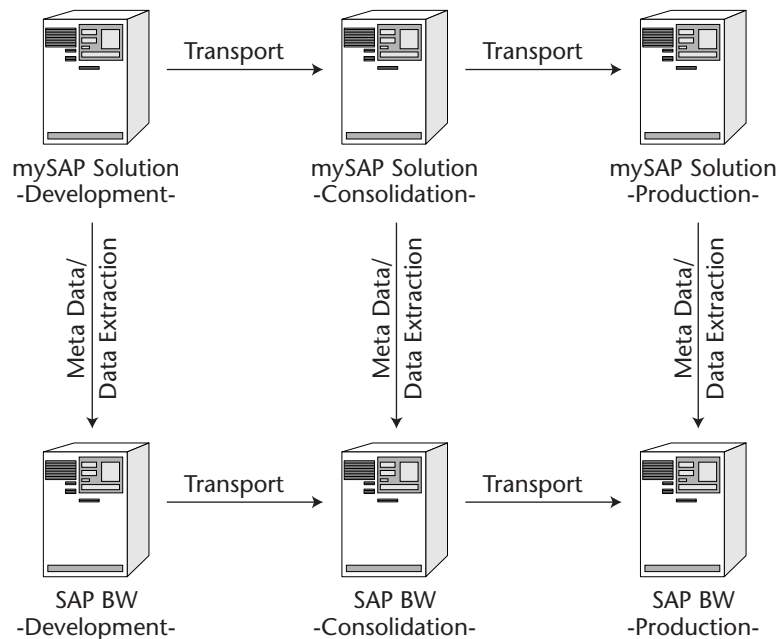


system and vice versa, unless you really know what you're doing), usually the test system, where special before- and after-import programs take care of additional actions after importing the objects. (Importing an InfoCube, for example, requires dynamically creating or modifying database tables and generating programs in the test system.) After testing is considered complete, you can import the same flat file into the production system for productive use.

Typical SAP BW transport requests are made up of a collection of different types of objects (e.g., an InfoCube consists of InfoObjects), making it difficult to manually ensure consistent transport requests. An SAP BW-specific transport connection tool allows you to select a specific object (e.g., an InfoCube) and collect all objects belonging to that object (e.g., InfoObjects from a definition perspective, InfoSources from a data sourcing perspective, or queries from an information delivery perspective).

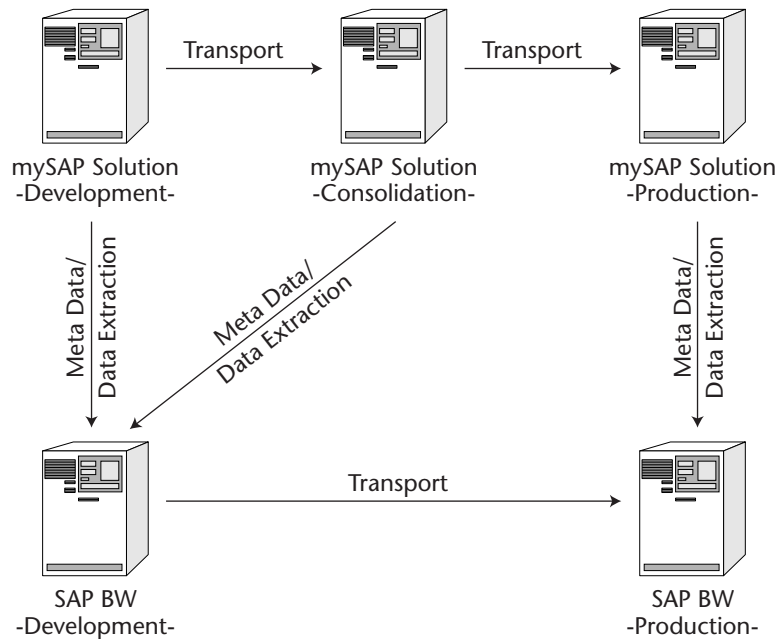
The TMS not only keeps track of changes in development systems; it also keeps track of changes in the test or production system—or depending on global system settings, prevents users from changing anything in these systems at all.

This type of scenario is well known to SAP users and larger organizations for complex custom software development and maintenance. The same paradigm and the same technology are now being utilized by SAP BW to ensure stable and highly available software systems. The TMS allows you to define and maintain complex system landscapes for application development—the most popular one for combined SAP R/3 and SAP BW, shown in Figure 3.20.



**Figure 3.20** Standard SAP BW system landscape.

## 84 Chapter 3



**Figure 3.21** Poor man's SAP BW system landscape.

There are two complete system landscapes now: one for SAP R/3 and one for SAP BW. And possibly there are even more than two, if there are additional SAP systems like SAP CRM and SAP APO used in an organization. These systems need to be in sync as well, for two major reasons:

- Development of an SAP BW application usually requires development and customization of existing or additional extraction programs, and sometimes, depending on the specific requirements, even changes to the customization of business processes in SAP R/3. Both development paths need to be kept in sync over the whole system landscape so that you can efficiently develop, test, and deploy the application.
- In typical development systems, testing is nearly impossible because of poor data quality. Reasons for poor data quality are that (1) development systems are normally sized according to development needs and do not allow mass testing or just even storing mass data and (2) data in development systems are often modified manually to provide test data for specific development test cases. A separate SAP BW test instance connected to the corresponding SAP R/3 test system enables high-quality testing.

However, some development activities in SAP BW are usually conducted in the test or even the production system. These include the definition of ad hoc queries, ad hoc Web applications, definition of data load procedures, and others. The TMS allows you to define exceptions for these types of objects—but that also implies that the TMS no longer tracks changes to these objects.

Note that talking about system landscapes does not necessarily mean talking about multiple physical systems. Today's high-performance parallel servers with many CPUs, large amounts of main memory, and access to storage networks allow you to install multiple instances of SAP R/3 or SAP BW systems on one single physical system. Even smaller physical systems today allow you to run a development system and an integration test system on one physical server in smaller organizations where development and testing frequently are consecutive processes so that there are little testing activities in intense development phases and vice versa.

Keeping that in mind, there's actually no reason to go for a poor man's system landscape like the one shown in Figure 3.21. While it may be used in early stages of SAP BW prototype development or with very small implementation teams, it will lead to more complicated and costly test preparation phases. In a combined SAP BW development and test system, all data will have to be checked and eventually reloaded each time a new test phase is started. Many data warehouse projects fail because of a lack of data quality and data credibility. Don't let yours!

A more complex scenario has proven useful in global rollouts, where a central development team works on a global template system, which is then localized and tested locally. The global rollout system landscape is depicted in Figure 3.22. The point here is that objects from the global development system are first transported into a global test system where the system may be tested prior to being rolled out to several local development systems. The local development systems are used to adapt the global template to local needs to some extent (language, local legal requirements, local business requirements). Keeping track of local changes to global objects, the TMS supports identification and synchronization of global and local objects. Once the localization of the application is completed, it may be transported to the local test and production systems.

**NOTE** SAP is actually using a similar scenario to roll out support packages and new releases of its new software to customers. And the same or similar mechanisms for identifying and synchronizing changes to global (in this case SAP-defined) objects are used to maintain the system's integrity. The SAP software itself can be considered a template for a local (in this case local means customer-specific) rollout, developed by SAP. Although the system landscape at SAP is more complex than the one shown in Figure 3.22, the basic principles remain the same.

Other complex application development projects might also require using a software integration system landscape (as shown in Figure 3.23), where objects from several different development systems meet in a central development landscape for integration work, final testing, and productive use.

The actual system landscape chosen in a particular situation largely depends on the complexity of the development work and the complexity of the rollout; investments in hardware and system setup pay off in the long run, through ease of development, integration, and testing. Experience has proven the return on investment through achieved increase in information accessibility and quality.

86 Chapter 3

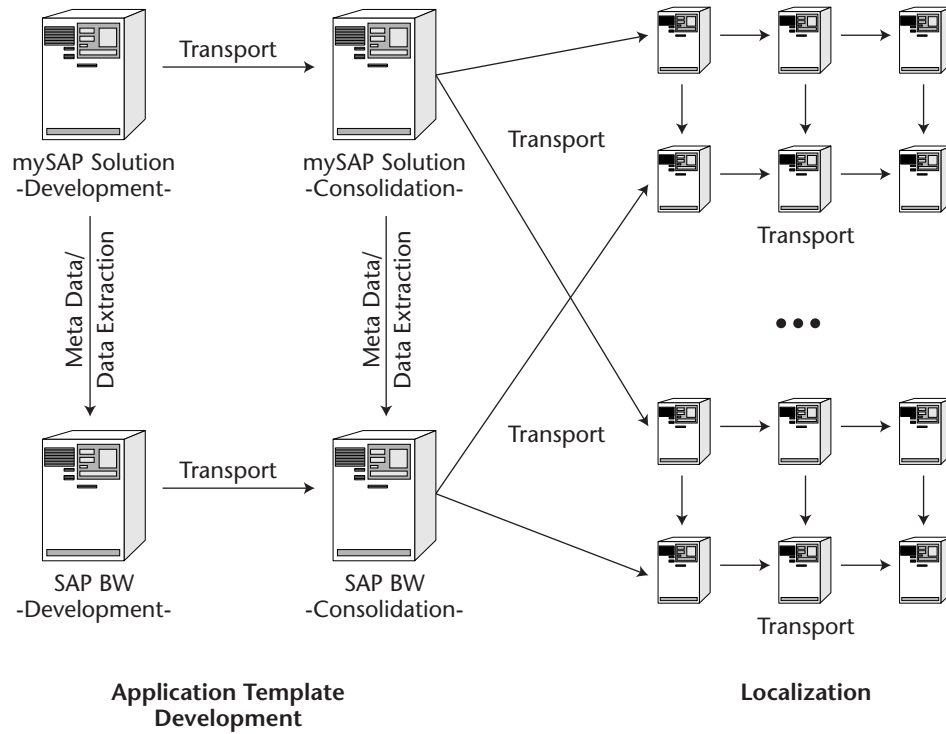


Figure 3.22 Global rollout system landscape.

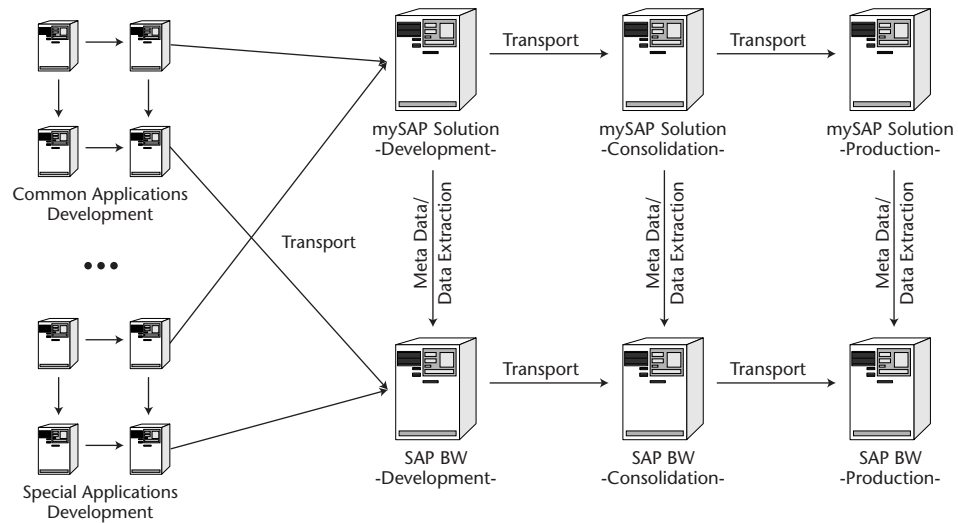


Figure 3.23 Software integration system landscape.

## Summary

---

SAP BW uses the usual layered architecture with an ETL layer, a storage layer, an analysis and access layer, and a presentation layer. SAP BW is completely based on meta data managed by the meta data services, and it is centrally controlled by the Administrator Workbench utilizing the administration services.

SAP-specific open interfaces like the Staging BAPI and the OLAP BAPI allow you to exchange data and meta data with other systems and tools optimized for SAP BW; industry-standard interfaces like OLE DB for OLAP, XML, and XML for Analysis are supported, allowing easy access to data and meta data maintained in SAP BW for virtually every tool supporting those industry standards.

SAP BW now provides close to complete functionality for building the corporate information factory. Operational data stores, data warehouses, and InfoMarts (basically defined as an extended-functionality data mart) can be built using the meta data objects available in SAP BW. Analytical applications are contained in the system built on top of the predefined Business Content.

The SAP Web Application Server has been used as a basis for developing the SAP BW software, effectively inheriting a broad range of tools, functionality, and code that originally proved to be helpful in speeding up the development of data warehouse software from scratch and now proves at least as helpful in cost-effectively maintaining that software.

With the transport management system, the SAP BW architecture includes a separate component for managing the overall development and deployment process for data warehouse applications supporting complex system landscapes and scenarios.

