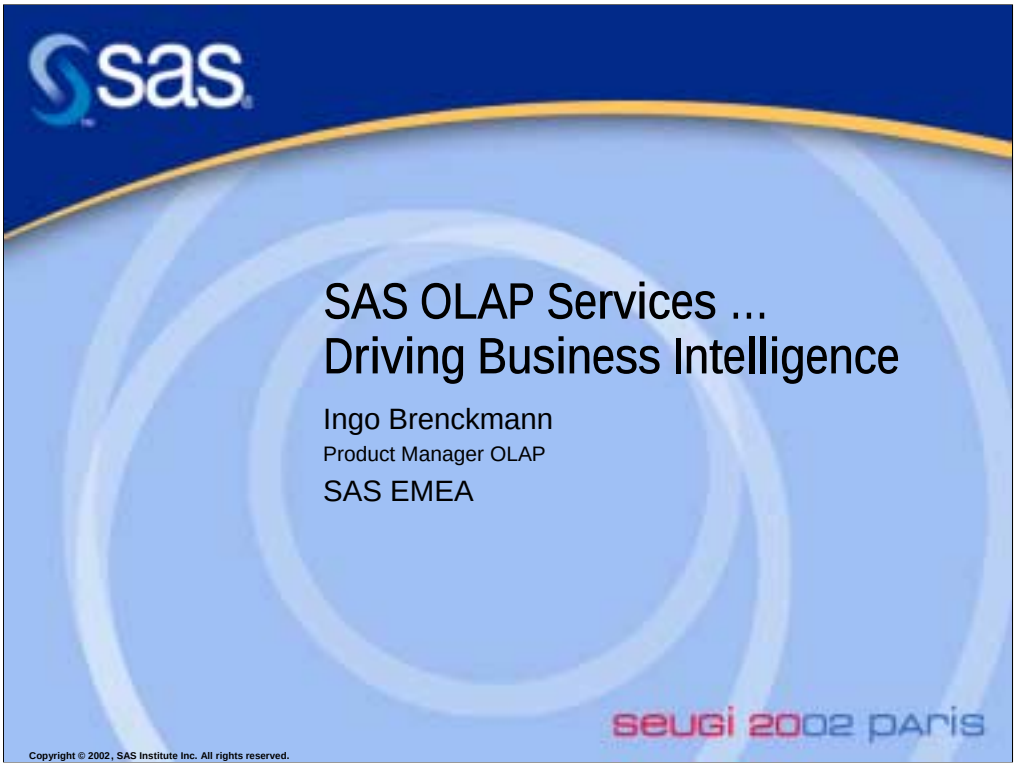
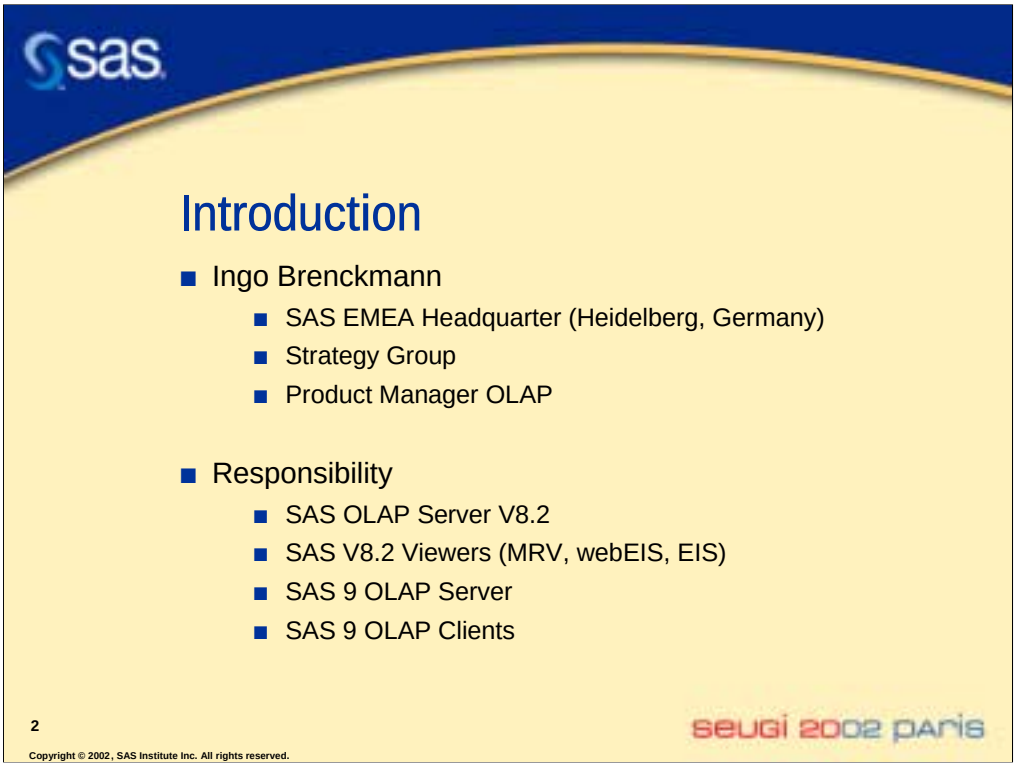


17.06.2002 13:34



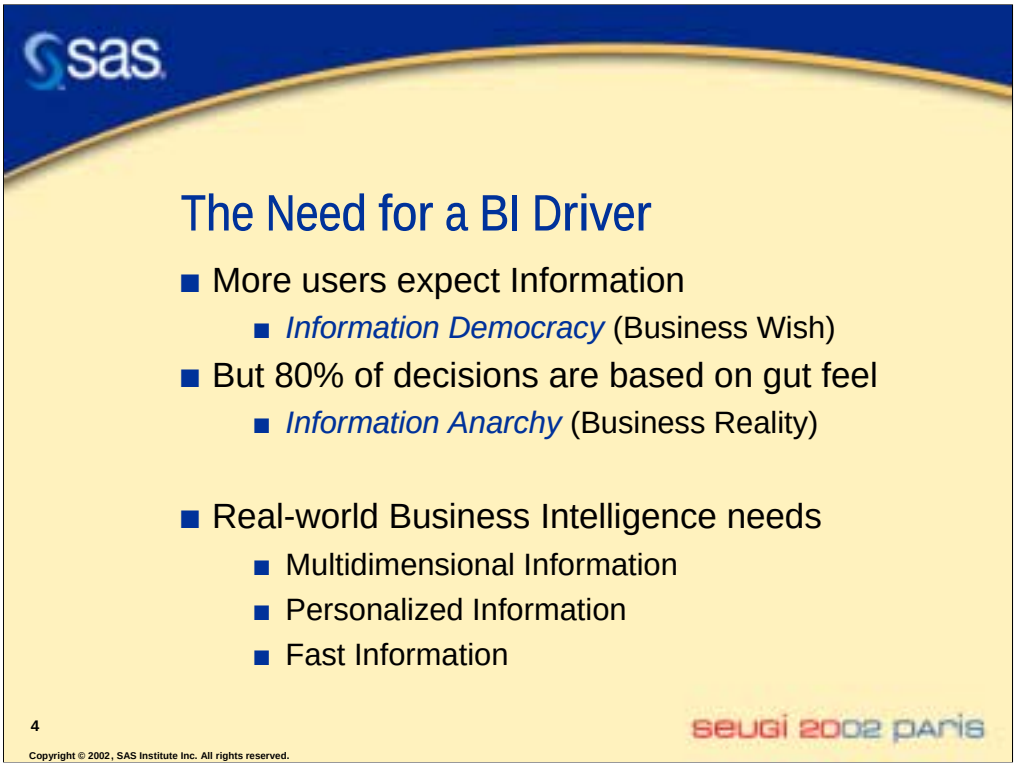
Due to a growing awareness for the need to analyze business data, companies are generating more and more reports that put an ever-growing burden on back end database servers, while still lacking intuitive interactivity, required to get a deeper understanding of hidden dependencies in data.

This paper provides a vision how to keep business intelligence clients away from accessing a central data warehouse by providing OLAP data sources that spin off from the data warehouse. The vision shows, how OLAP is capable of driving business intelligence applications at the speed of thinking and with functionality that supports the user's train of thought, providing a real added value to Business Intelligence.

The first section discusses the reasons why there is a need for OLAP and how it is capable of driving Business Intelligence.

Section two details how SAS OLAP server can be used in an enterprise level deployment that uses OLAP as a service for Business Intelligence.

The last section provides a first look on the new SAS 9 ngOLAP Server to be released next spring. The information provided is based on the experimental release contained in SAS Version 9.

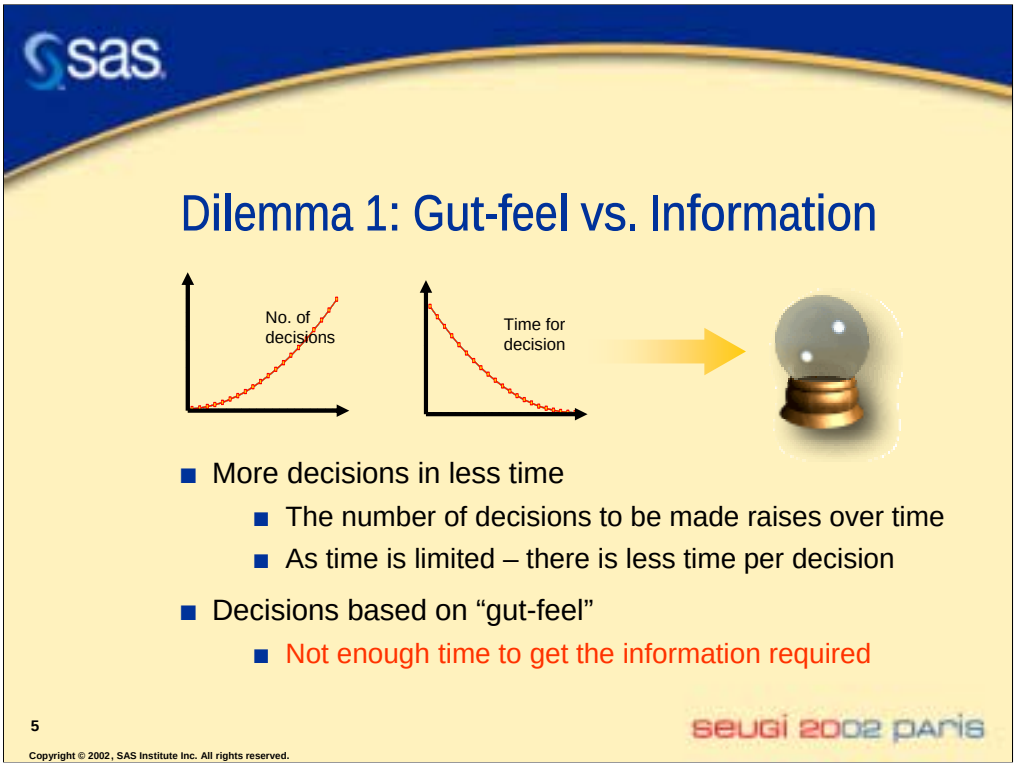


There is a business rule that says, that a decision should be taken by somebody that is expected to be close to the people that will have to deal with the results of the decision. These decisions, being taken somewhere in a hierarchy require information – information to base decisions on.

A perfect world model provides “Information Democracy” where every single employee has access to information that he/she needs to take the right decisions at the right time – no more and not less information. This is typically what business would like to get.

In the real world, 80% of the decisions being taken are based on gut-feel. Decision takes to not know how to get the appropriate information or it is simply not available within the timeframe a decision has to be taken.

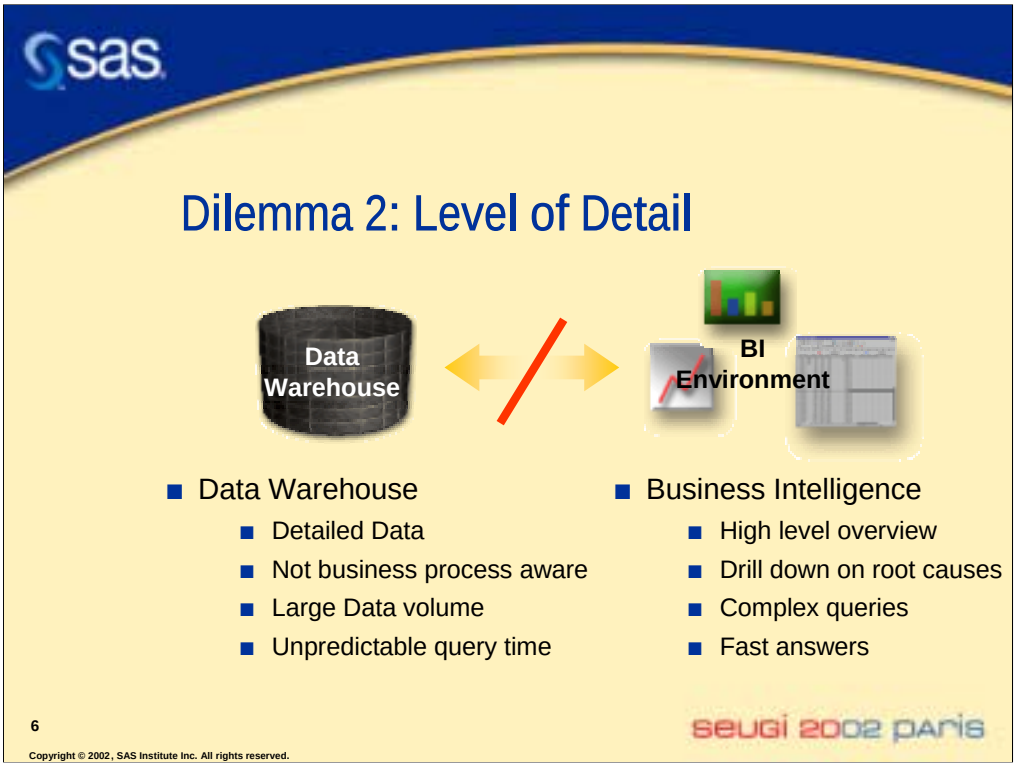
As a result, business needs a piece of software that delivers information to a common accessible place. This information needs to be delivered in time to take decision so it needs to be delivered fast. As business users are not willing to search for hours in order to find relevant data, the information provided needs to be personalized. Different business users have different views on the company and on business. All these views need to be supported by the underlying software, providing a multidimensional view on business.



There are two major dilemmas for business users:

First of all, they do not have enough time to get required information so they base their decisions on gut-feel.

A reason for this situation can be found in the race for efficiency setup by most large companies. Basically companies reduce their workforce to cut costs and the remaining staff has to decide upon the same amount of business issues. The single person has to make much more decision but does not get any extra time to make theses decisions. As a result the time he or she can spend on a single decision is decreasing which leaves no time for finding the relevant information but rather requires "good gut-feel".



The second major dilemma can be seen in the level of detail a business user requires to decide on something.

Most companies do have data warehouses implemented that collect all data available to the company. This data is very detailed as it contains every event that occurred in history. The business user does not need that level of detail to make a decision. For him/her it is important to have an overview of the current situation so he/she needs a high level overview on a topic.

The data warehouse is not aware about the business process that puts all the events stored in it into a business context. How are the facts stored related to the business events that occur in an enterprise and how do they affect key performance indicators? This information is not available from the data warehouse but it is required by the business user in order to be able to drill down from a high level overview to details that affected this overview.

The data warehouse contains large amounts of data and business user have complex queries to be answered. This imposes a big impact on a large database as each single query can become so complex that it takes hours to search through the entire data store. Queries can become unpredictable as it might as well take a few days to answer a query. This is not suitable for decision takers as they need to act fast.

So there needs to be a way how to satisfy these requirements. Let's first list these requirements.

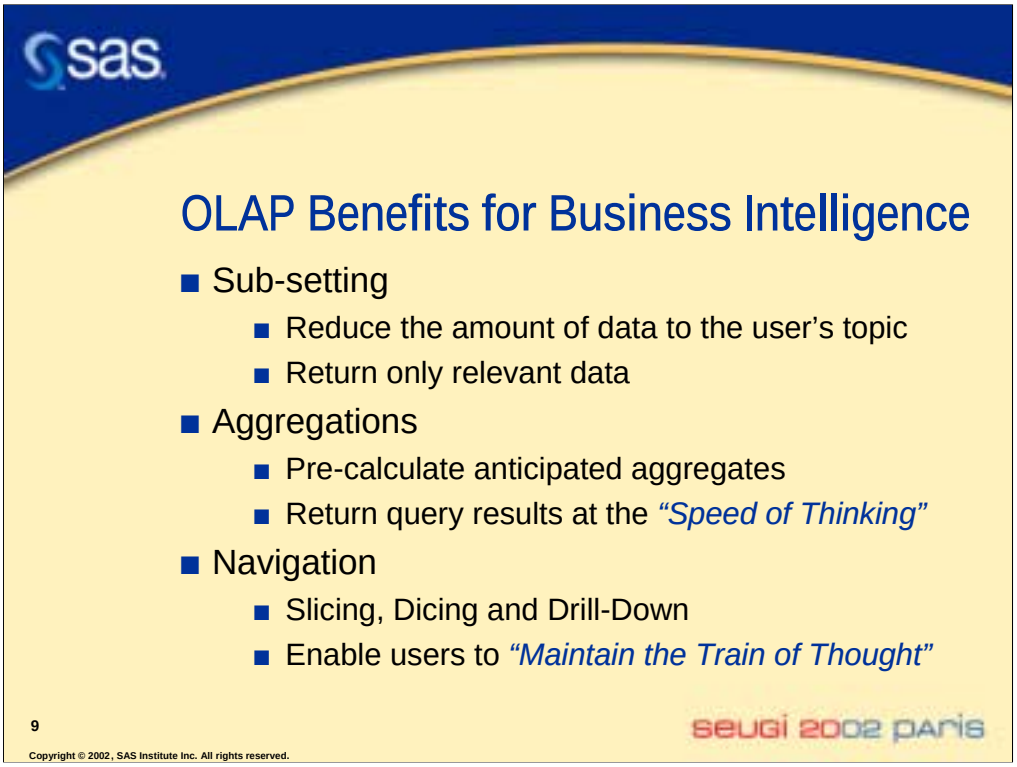
Here is a list of requirements that is setup by business requirements:

The major two requirements are fast access to information and the restriction to exactly these pieces of information required by a business user.

In order to provide fast access to information, the behavior of a user needs to be anticipated. If there is a high probability that a certain information will be accessed by user, this information should be pre-calculated and stored for fast access. The amount of data driving such business intelligence environment should not be too big as large amounts of data require a longer time to be searched. The amount of data can be reduced by splitting up the entire set of data into smaller chunks that deal with specific topics, e.g. sales, marketing or human resources.

In order to provide personalized data, the business intelligence environment needs to be aware of the identity and role of a user and needs to reduce the data according to that information. The user can be protected from an information “overkill” by not providing him with detailed records of data but rather a high level overview with the possibility to drill down on the details.

Let's take a look at SAS OLAP Server and how it can satisfy the needs defined in the first chapter.



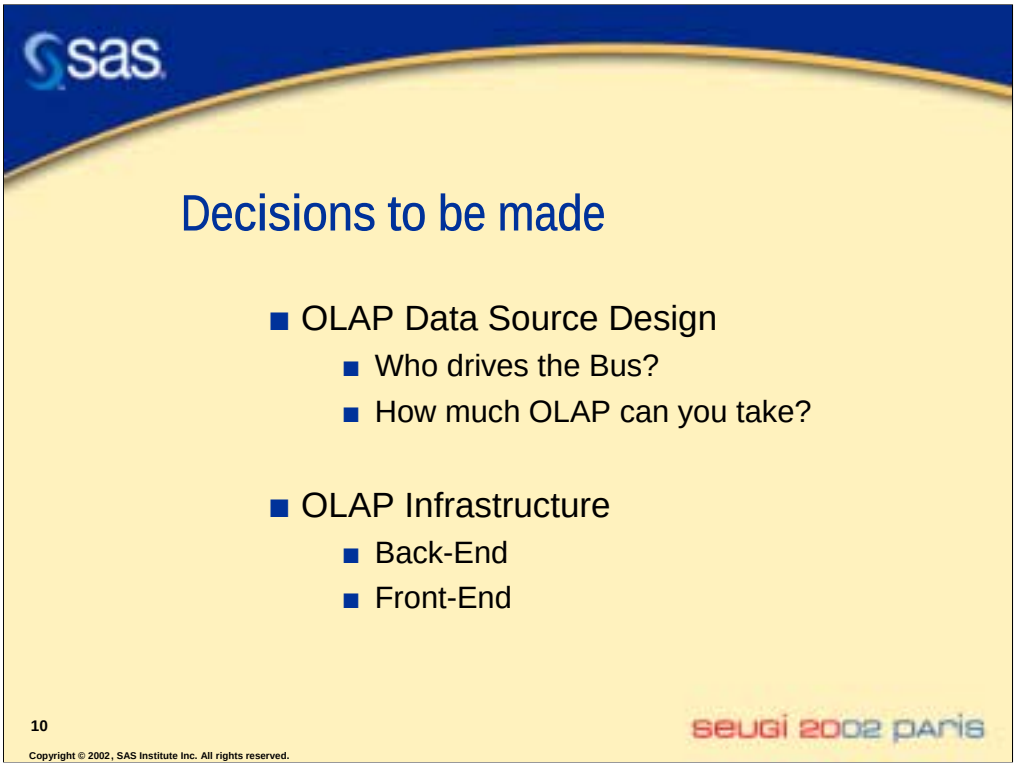
Sub-setting, aggregations and navigation – these are the major benefits that OLAP provides to Business Intelligence environments.

Sub setting reduces the amount of data provided by a data warehouse to the amount of data associated with the topic of a special OLAP data store. Access Control Lists can be applied to these OLAP data stores that further reduce the information to those pieces that are related to the user accessing the OLAP data store.

OLAP data stores contain aggregated data for different levels of detail. The aggregations are generated when an OLAP data store is built. By precalculating aggregates, queries issued by business users can be answered at the speed of thinking.

OLAP provides exploration patterns that provide a deeper understanding of data contain in an OLAP store. Slicing enables the user to reduce the number of dimensions to two (a slice has only two dimensions). The slice can be moved through the OLAP data store by varying the other dimensions, e.g. sales for products and countries while the user is capable to move through years. Dicing enables the user to further reduce the amount of information to a smaller cube. The drill-down pattern provides the capability to query for details about an aggregated value, e.g. the current report provides a sales figure for 2001 – by drilling down on 2001 the user gets a report that provides values for the four quarters that make up the year 2001.

Using these navigation-patterns the user can maintain his/her train of thought while browsing through the information.

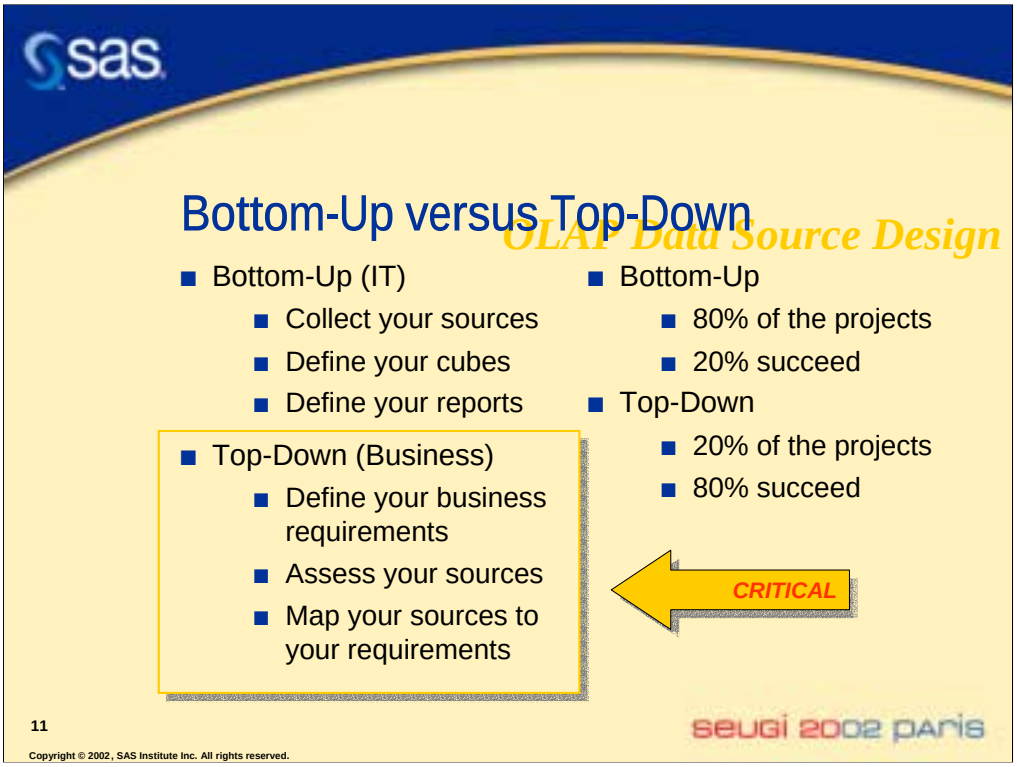


Before implementing an OLAP server two major decisions have to made. One about the OLAP data source design and one about the Infrastructure.

The decision on the OLAP data source design has to take into consideration whether the OLAP data sources should be designed bottom-up or top-down and how many different source should be built.

Deciding on the OLAP infrastructure includes the decision for a certain back-end infrastructure and a certain front-end infrastructure.

The following slides provide aspects of these decisions.

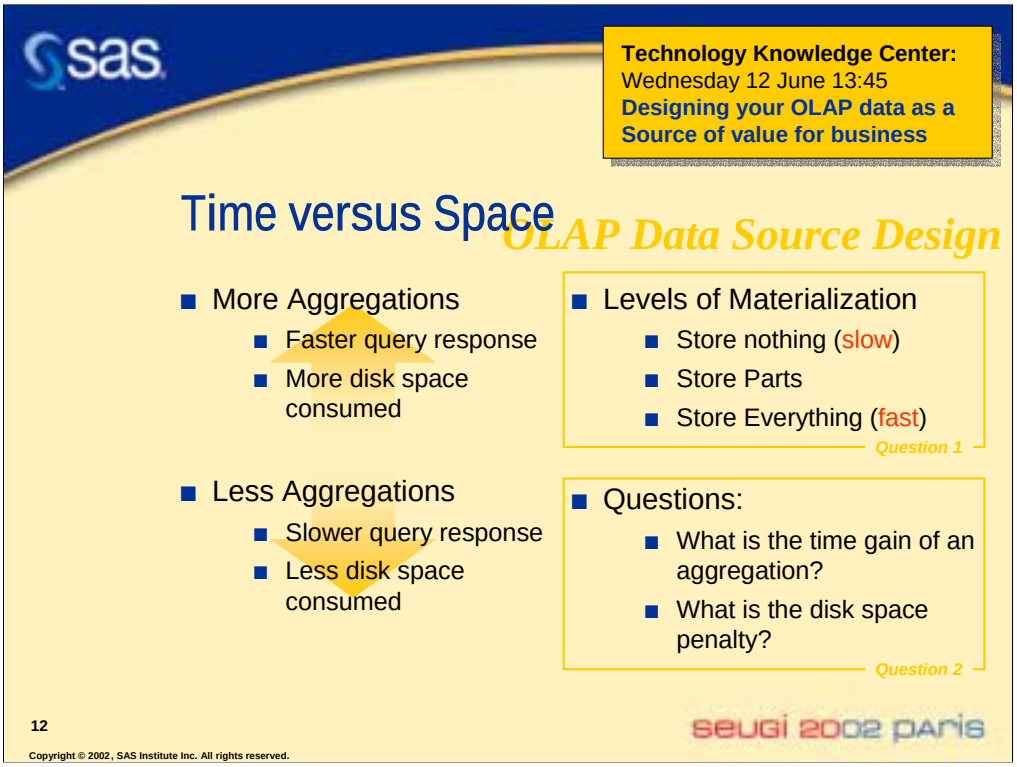


When designing OLAP data sources you can first collect all data sources that exist in your company, e.g. SAP/R3 data on DB2, some CRM data on ORACLE. When you know about your data source, you design OLAP data stores on top of these data sources and connect reports to these OLAP data stores. This approach is called bottom-up and is the typical way how IT proceeds.

Another approach, the top-down approach, focuses on the business requirements. In a first step, business requirements are gathered throughout the participating divisions of an enterprise. In a second step all data sources available in the company will be assessed. The third and last step maps the sources to the requirements and builds up OLAP data stores that satisfy the business needs.

80% of all BI projects start using the bottom-up approach but only 20% of these projects succeed. On the other hand 20% of the projects use the top-down approach and 80% of these succeed.

It is critical to the success of your project that you follow the top-down approach when designing your OLAP data sources as OLAP is a service for Business Intelligence and business defines the requirements for Business Intelligence.



Another decision you have to take when designing OLAP data sources is the decision pro time or pro space.

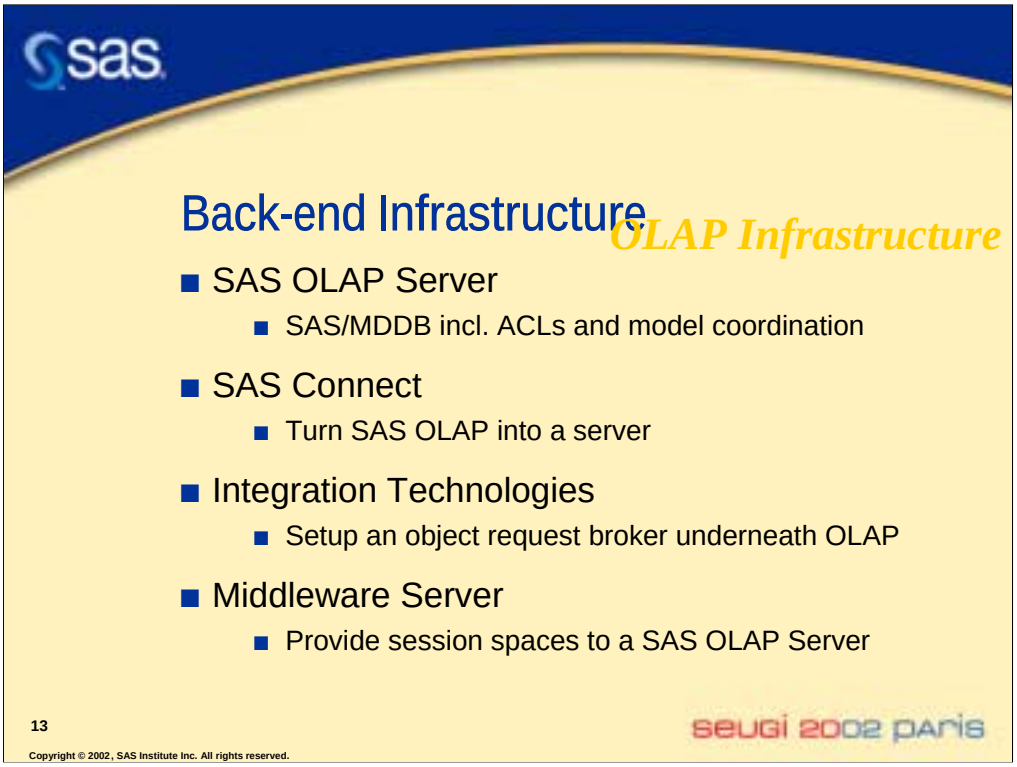
If you add more aggregations to your OLAP data sources, you anticipate more possibilities to navigate through the information. Query times will be reduced to the minimum but each aggregation consumes space.

On the other hand you can save lots of space by not storing any aggregation but the user will have to wait for answers as all aggregations will have to be calculated at runtime.

A compromise has to be found in-between storing nothing and storing every possible aggregation. There are two questions you should answer for each aggregation before storing it:

1. What is the time gain of the aggregation?
2. What is the disk space penalty for storing this aggregation?

Only if the combination of both makes sense you should go for the aggregation.



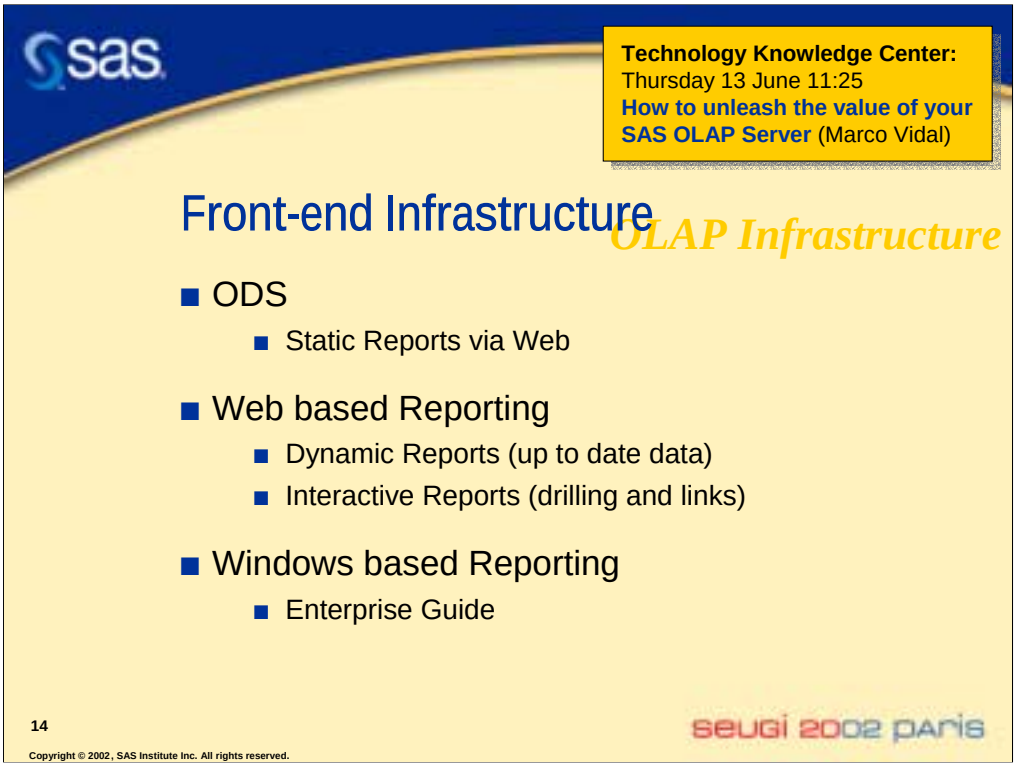
SAS offers a whole range of products that allow you to setup an OLAP infrastructure tailored to your specific needs. The products relevant to this topic are:

The SAS OLAP server which includes the SAS/MDDB, support for access control lists and a model coordinator.

If you would like to run the SAS OLAP server in a multi user environment, SAS/Connect provides the required connectivity to the OLAP server.

For larger deployments it can become necessary to have multiple instances of the OLAP server running on one machine in parallel on several processors. SAS Integration Technologies provides the object platform that lets you run several instances of SAS.

If you expect a large audience of users to hit the OLAP server, the middleware server will help you. It provides session spaces to users which only connect to SAS sessions when there is work to do. So you can have a large number of user sessions connecting to a much smaller number of SAS sessions that do the work.



Looking at the front-end there are once more many options that allow you to adopt a solution to your specific needs.

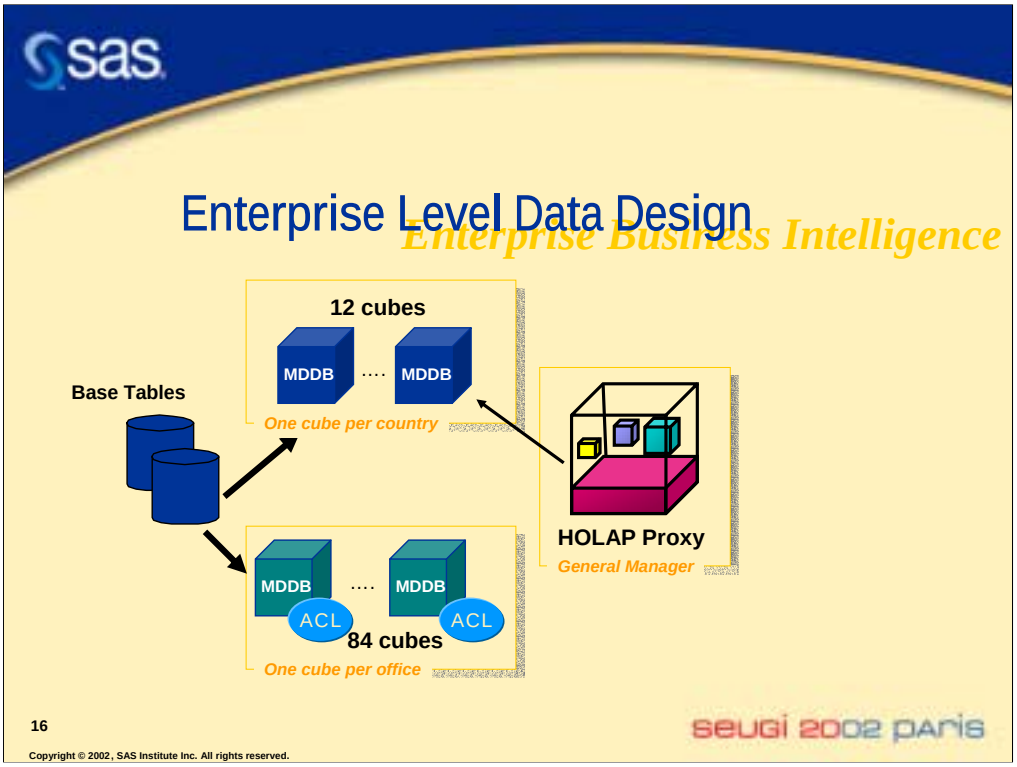
The Output Delivery System (ODS) can generate personalized static reports based on OLAP data in an overnight batch process. Reports are sent to a web-server and can be accessed from the SAS Information Delivery Portal (SAS IDP)

If users require interactivity in their reports, either dynamic reports that refresh their data when the underlying data changes can be implemented or even interactive reports. Interactive reports can be changed by the user at any point in time. Users can drill down on details or they can change dimensions and subset results. The SAS Application Development Studio helps you create such environments. Various technologies like applets and Java server pages are available via webEIS while CGI based solutions can be setup using the SAS Multidimensional Report Viewer (MRV).

There is also a windows based SAS client that provides a front end to all the power SAS can provide. The Enterprise Guide can connect to OLAP data sources and data mine this information or simply create reports using the data.

The following example builds up a reporting environment for a large company with different types of users that expect different levels of information.

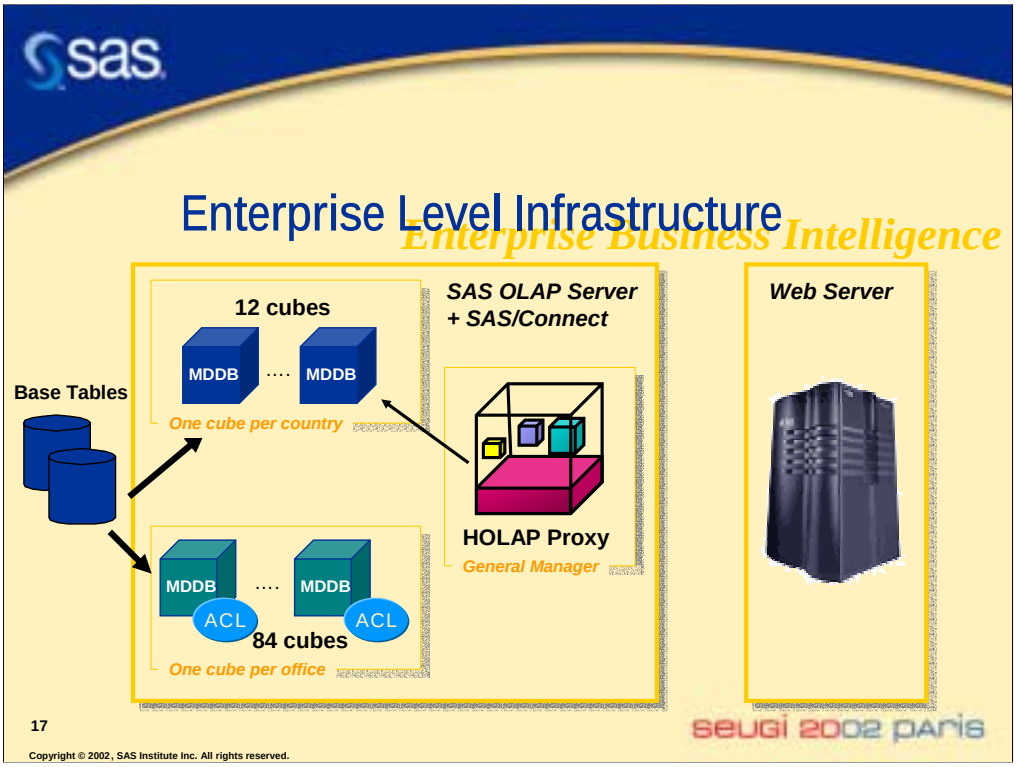
The company is a global player with several offices in different countries. There is a general manager that oversees the entire operation. There are country managers responsible for one country and there are office managers responsible for one office. In one office there can be several agents working.



In order to reduce the amount of data accessed by different users, each country and each office has its own cube. A country manager has access to the OLAP data source of his county and an office manager has access to the OLAP data source of his office.

A HOLAP (hybrid OLAP) data store combines all the cubes from the different countries to form a virtual cube across all operations. The general manager uses this cube to analyze the company as a whole.

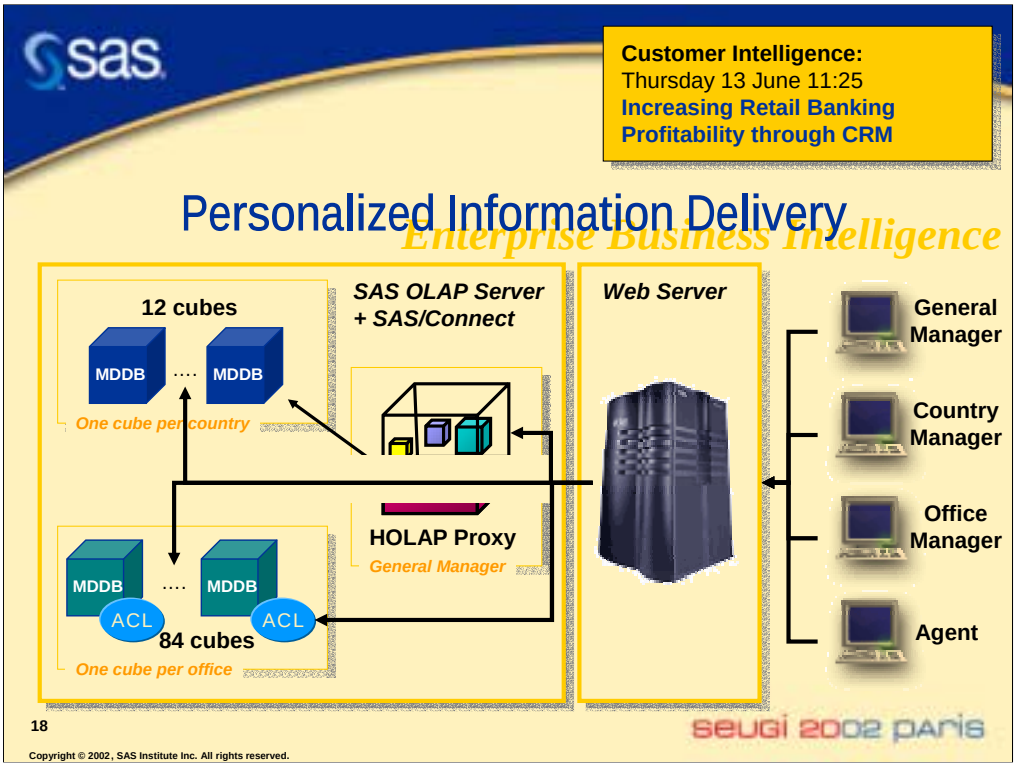
Access Control Lists have been applied to the office cubes in order to reduce the view of an agent to data that he/she is associated with.



All the cubes are located on one OLAP server. The OLAP server uses SAS/Connect to provide an access point for interactive reports accessing data over the internet.

All cubes are loaded in an overnight ETL process. As non of the cubes are based on another cube, all of them can be generated in parallel. The entire built time is equal to the built time of the biggest cube.

Reports are hosted on a web-server. If the reports need interactivity, they can connect from the web server to the OLAP server via SAS/connect.



This slide includes animations that display the four different users:

General Manager

If the general manager logs into the SAS Information Delivery Portal, he/she gets access to his/her reports. The reports access the HOLAP proxy that makes use of all cubes from the countries.

Country Manager

If a country manager logs into the portal he/she gets access to his reports which are based on the one single cube containing data about his/her country.

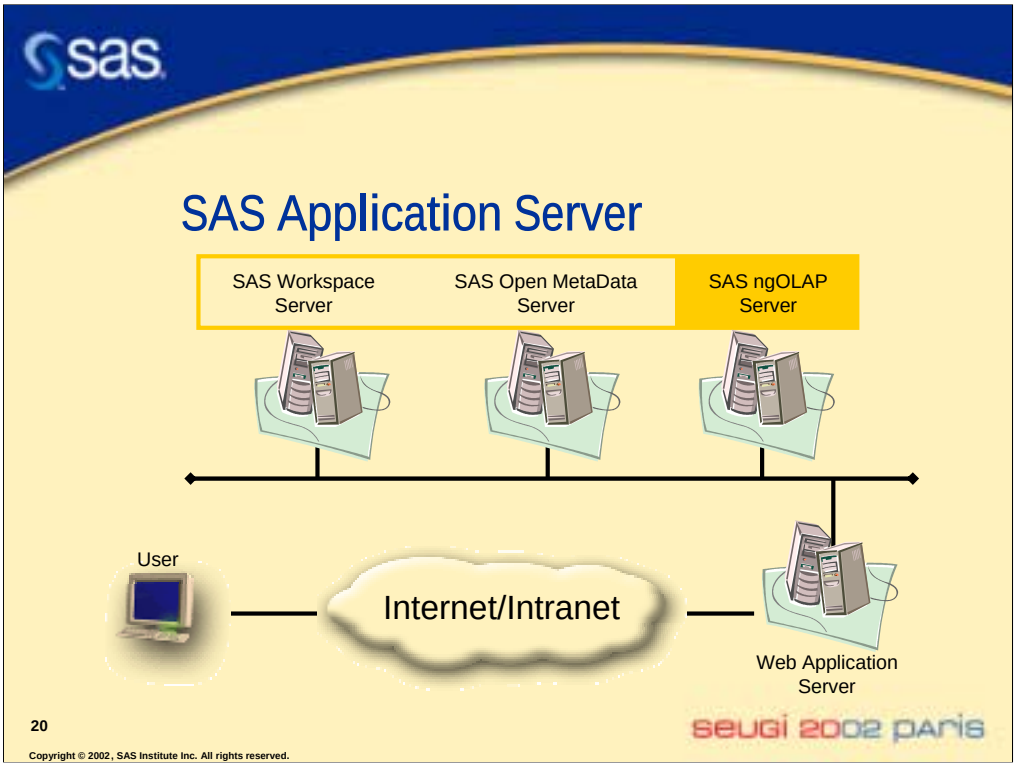
Office Manager

If an office manager logs into the portal he/she will access reports based on the one cube containing data on the appropriate office.

Agent

Now the agent logs in. He/she works for a certain office and has got access to his/her data. The access to the entire office data is restricted via Access Control Lists.

Looking at the future of OLAP within SAS, the focus on seeing OLAP as a service for Business Intelligence is getting becomes obvious.



The new SAS V9 infrastructure knows four kinds of application servers:

SAS Workspace Server

The SAS Workspace Server provides sessions that execute SAS code and provide access to SAS libraries.

SAS Open Metadata Server

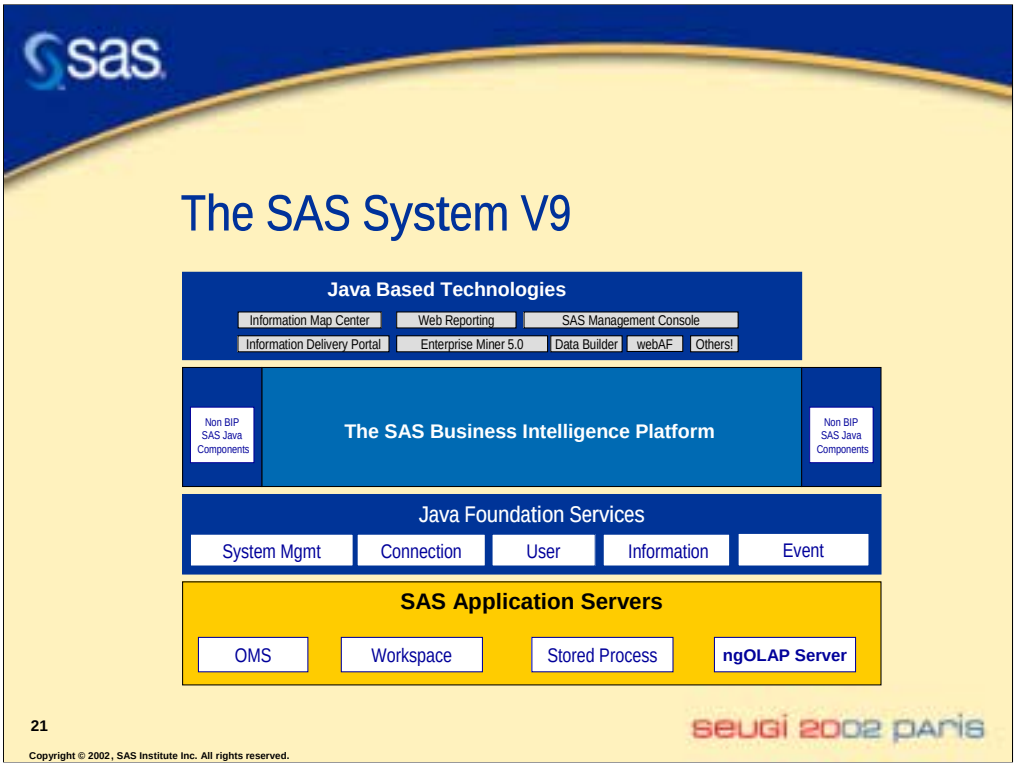
The SAS Open Metadata Server provides metadata to all applications based on the SAS environment. The Open Metadata Server makes use of the common warehouse metadata model (CWM) as defined by OMG. You can register any kind of resource in this server, e.g. other servers, libraries, applications, users, roles.

SAS Stored Process Server (not on the diagram)

The SAS Stored Process Server provides common process that can triggered by any consumer in the network. Processes can contain any sort of SAS code.

SAS ngOLAP Server

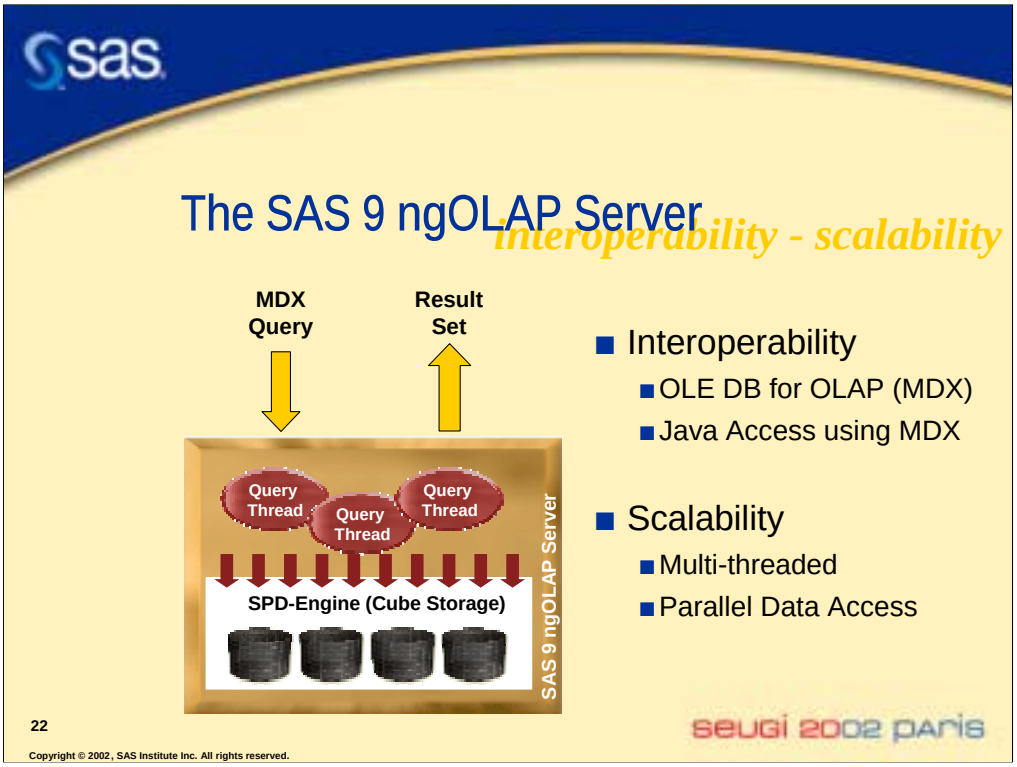
The new SAS OLAP Server with a multithreaded kernel and parallel access to data.



The SAS Application Server provide the basis for the new SAS System V9.

On top of the servers there are Java Foundation Services that provide the basic services required to access the application servers, e.g. connection and session management, event handling etc.

The Business Intelligence Platform uses the Java Foundation Services to expose the SAS Application Servers to the Java Based Clients.



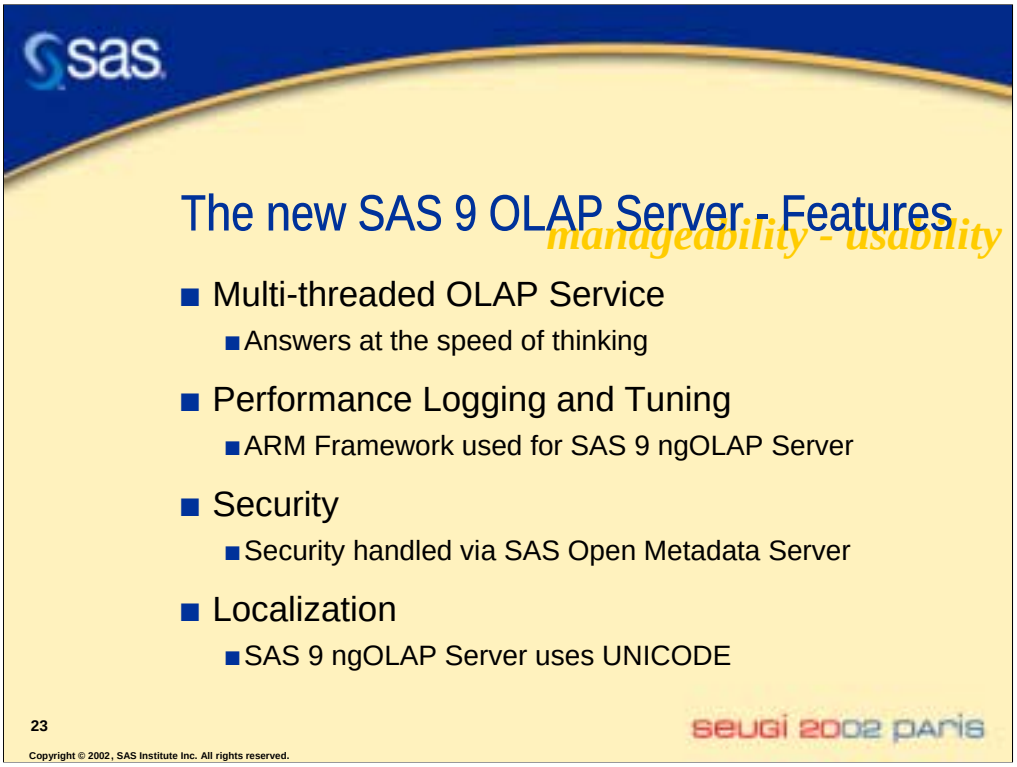
The new SAS ngOLAP server stores multidimensional data and provides access to it via MDX (multidimensional expressions) MDX is a standard language for querying multidimensional data source, define by Microsoft in the OLE DB for OLAP specification. You can use OLE DB for OLAP to query the new SAS ngOLAP Server or you can use SAS Java Reusable Components that send MDX statements from Java applications, applets or servlets to the server.

Using MDX the SAS ngOLAP server is **interoperable** with all major tools that use OLE DB for OLAP:

The SAS ngOLAP Server uses the scalable performance data engine (SPD-E) to store its data. SPD-E partitions data and stores it together with hybrid indexes on several hard discs connected to several controllers.

Each user sending a query to the OLAP server is handled by an individual thread. The thread analyses the query and tries to split it up into several smaller queries that can be executed in parallel. Each of these sub queries is assigned to a thread and all of the threads can access the data in parallel using SPD-E.

Using SPD-E the SAS ngOLAP Server can **scale** with users and data sizes.



The SAS ngOLAP Server can be administered using the SAS Management Console. OLAP data sources are designed using the SAS OLAP Administrator which can also plug into the ngWarehouse Administrator.

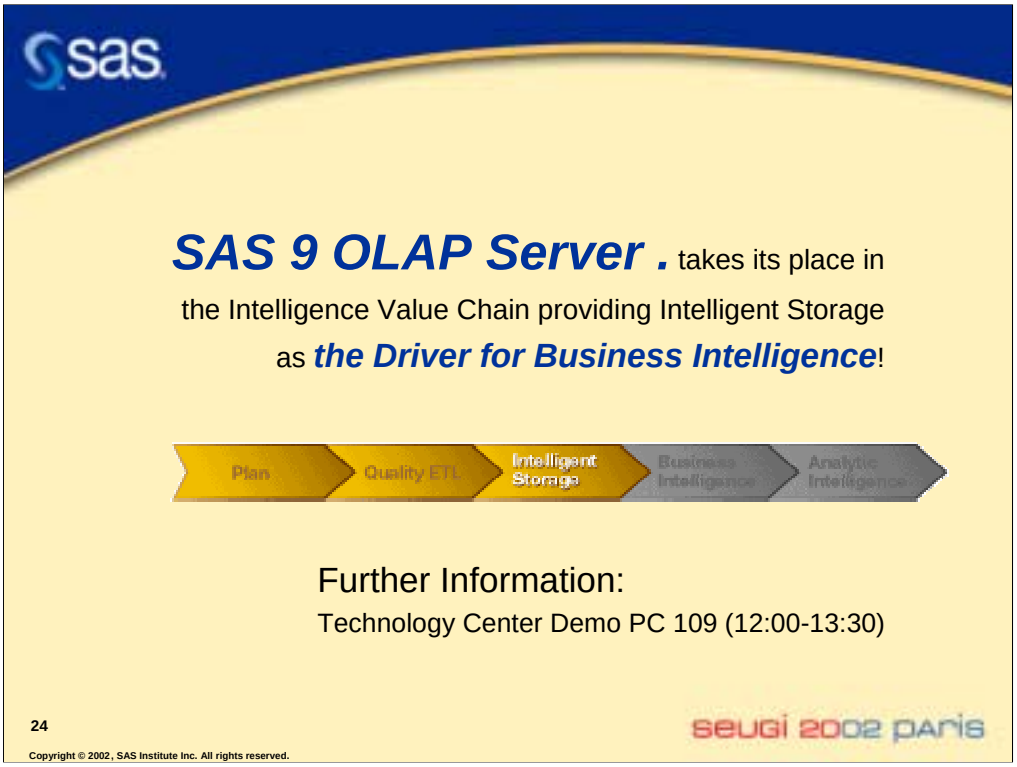
These Java Administration and Design tools enhance the **manageability** of the entire system.

The multithreaded OLAP server provides answers to complex business questions at the speed of thinking. Administrators can monitor the performance of the OLAP server from inside the server as the SAS ngOLAP Server was build using the Application Response Measurement Framework (ARM).

Security is not an issue as it is integrated into the security system of the entire SAS system stored in the Open Metadata Server.

The SAS ngOLAP Server uses UNICODE when accessing its SPD-E based data stores.

All these features make SAS ngOLAP Server very **easy to use**.



SAS 9 ngOLAP Server

Takes its place in the Intelligence Value Chain

Providing Intelligent Storage

As the Driver for Business Intelligence.

Ingo.Brenckmann@eur.sas.com
Ingo Brenckmann
Product Manager OLAP
SAS EMEA Strategy