



ABAP Part I

Lesson 01: R/3 Architecture

Lesson Objectives



After completing this lesson, participants will be able to -

- Know the meaning of ERP and SAP
- Understand the R/3 system
- Understand the Basics of SAP
- Log on to SAP and do the Basic Navigations



What is ERP?

- E – Enterprise R – Resource P – Planning

Definition:

- An integrated information system that serves all departments within an enterprise.
- ERP is a way to integrate the data and processes of an Organization into one single system.
- Software solution that addresses the enterprise needs taking the process
- View of an organizational goals tightly integrating all functions of an enterprise

Introduction to ERP



SAP	Oracle	PeopleSoft	JDEdwards
SD	Marketing, Sales	Supply chain	Order management
MM	Procurement	Supplier relationship	Inventory, procurement
PP	Manufacturing		Manufacturing mgmt
QM		Enterprise perform	Technical foundation
PM	Service	Enterprise service	
HR	Human Resources	Human capital mgmt	Workforce management
FI	Financials	Financial mgmt sol.	Financial management
CO			Time & Expense mgmt
AM	Asset Management		Enterprise asset mgmt
PS	Projects		Project management
WF	Order Management		
	Contracts		Subcontract, real estate

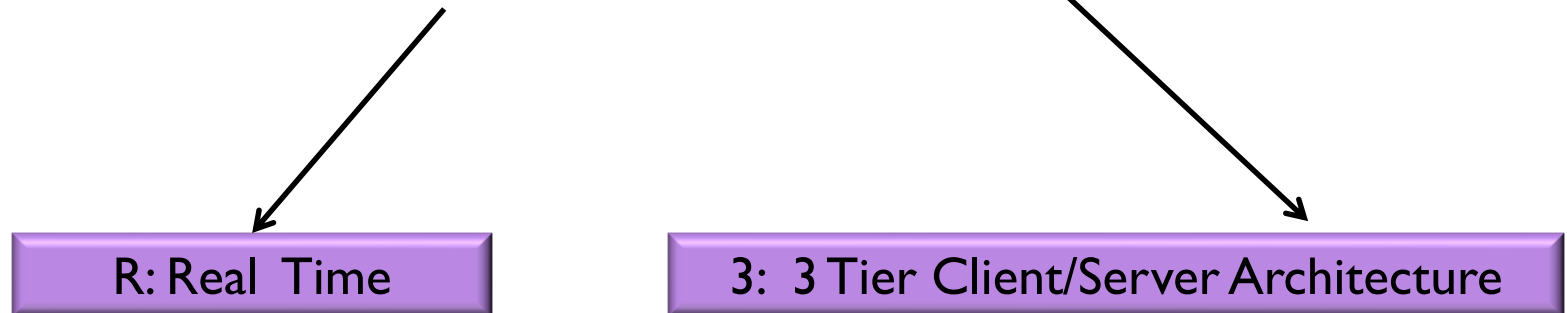
Introduction to ERP



What benefit	How
Reliable information access	Common DBMS, consistent and accurate data, improved reports.
Avoid data and operations redundancy	Modules access same data from the central database, avoids multiple data input and update operations.
Delivery and cycle time reduction	Minimizes retrieving and reporting delays.
Cost reduction	Time savings, improved control by enterprise-wide analysis of organizational decisions.
Easy adaptability	Changes in business processes easy to adapt and restructure.
Improved scalability	Structured and modular design with “add-ons.”
Improved maintenance	Vendor-supported long-term contract as part of the system procurement.
Global outreach	Extended modules such as CRM and SCM.
E-Commerce, e-business	Internet commerce, collaborative culture.



SYSTEMS APPPLICATIONS and PRODUCTS in Data Processing





SAP – An Introduction

ERP (Enterprise Resource Planning) Product

Name of the Company and its Product

German Based

ERP Market Leader

Industry Specific Best Practices



What is SAP?

- S – Systems A – Applications P – Products in Data Processing

SAP was started in 1972 by five former IBM employees in Mannheim, Germany.

SAP have a very high level of integration among its individual applications which guarantee consistency of data throughout the system .



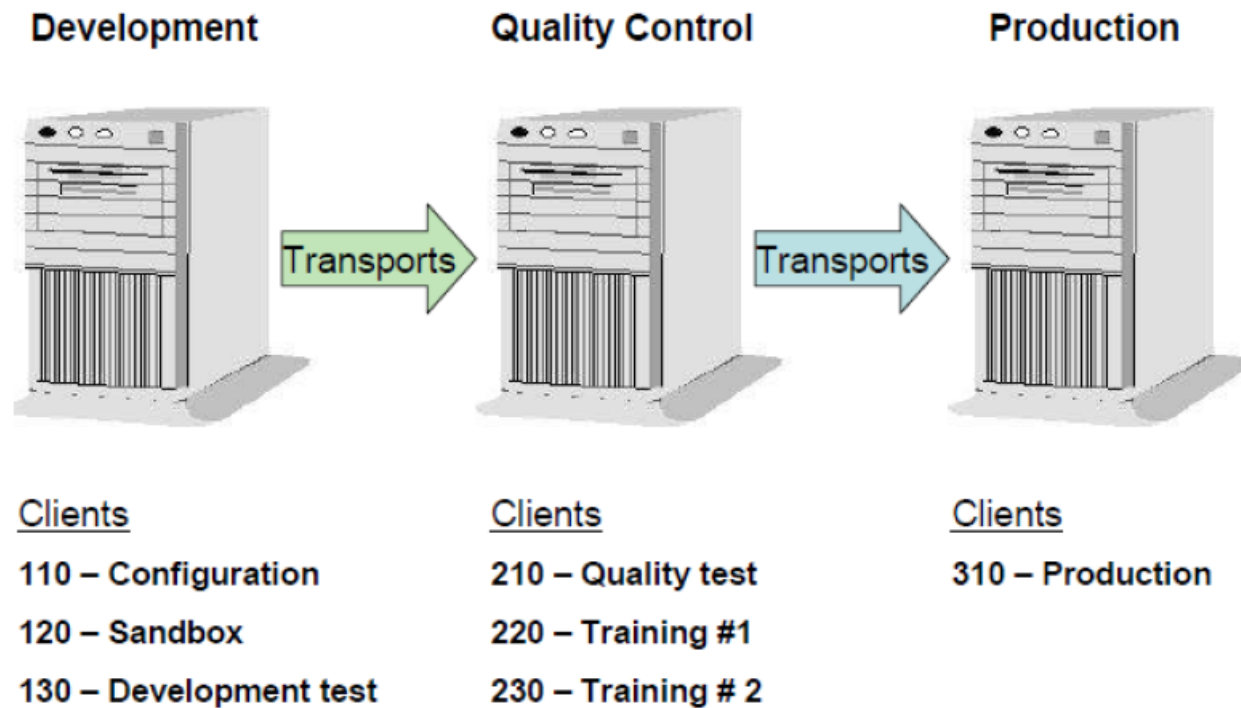
Functional Modules

- FICO – Finance & Control
- PP – Production Planning
- MM – Material Management
- SD – Sales & Distribution
- HR – Human Resources

Technical Modules

- ABAP – Advanced business applications programming
- XI – Exchange Infrastructure
- Basis –
- BIW – Business Information Warehousing

SAP Landscape



Introduction to NetWeaver



NetWeaver is SAP's platform for composition and integration of loosely coupled applications following Service Oriented Architecture The Application Platform is the runtime environment for SAP NetWeaver.

Supports two languages (ABAP and Java) at the same time

ABAP – Advanced Business Application Programming



The main purpose of an R/3 system is to provide a suite of tightly integrated, large-scale business applications. The standard set of applications delivered with each R/3 system are the following:

- PP (Production Planning)
- MM (Materials Management)
- SD (Sales and Distribution)
- FI (Financial Accounting)
- CO (Controlling)
- AM (Fixed Assets Management)
- PS (Project System)
- WF (Workflow)
- IS (Industry Solutions)
- HR (Human Resources)
- PM (Plant Maintenance)
- QM (Quality Management)

Defining R/3



R/3 means Real-time 3-tier Architecture

R/3 software supports all of a company's business transactions and links them together using real-time integration

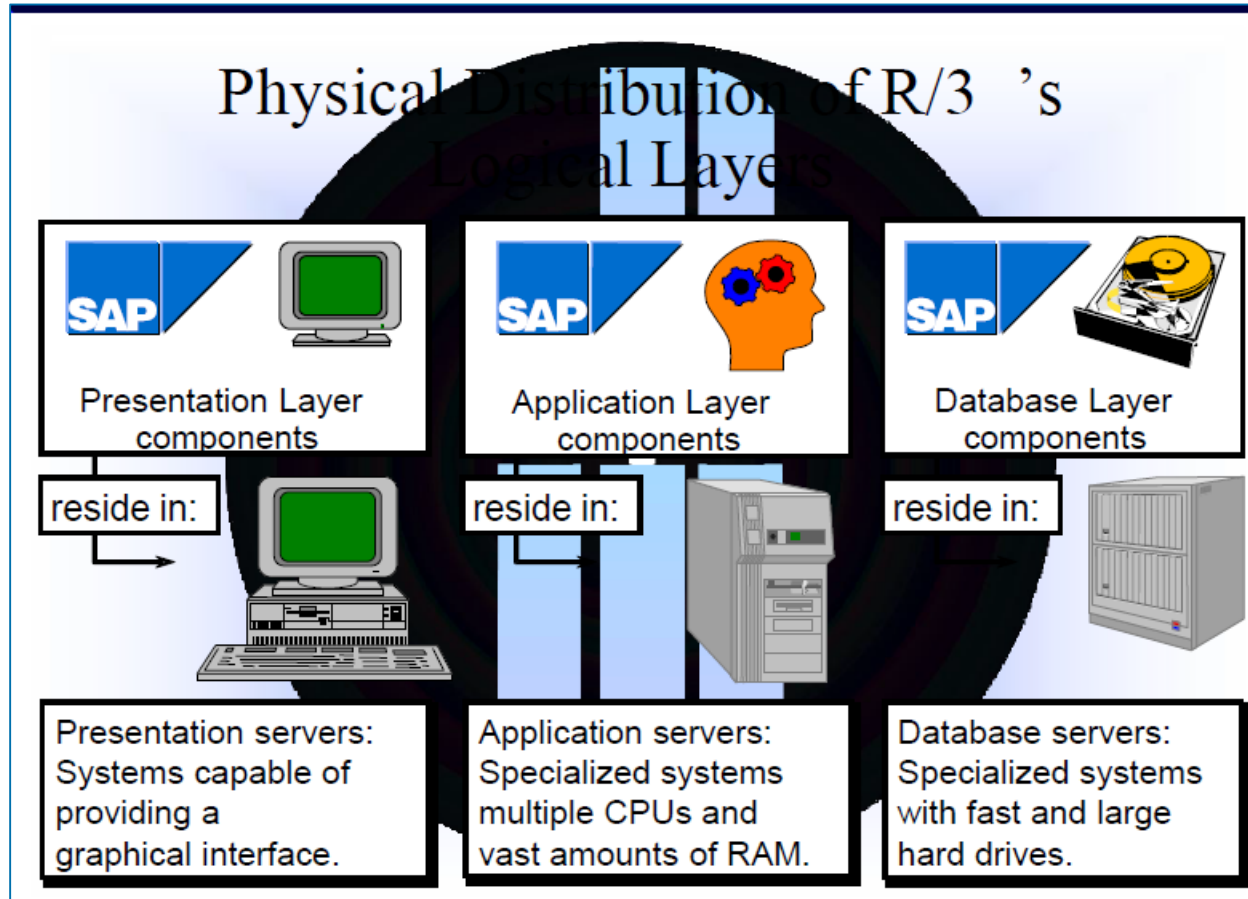
Real-time integration means that each change or update in one application causes the automatic change or update of the data in the other applications involved.

R/3 also represents 3-tiered Client-Server Architecture.

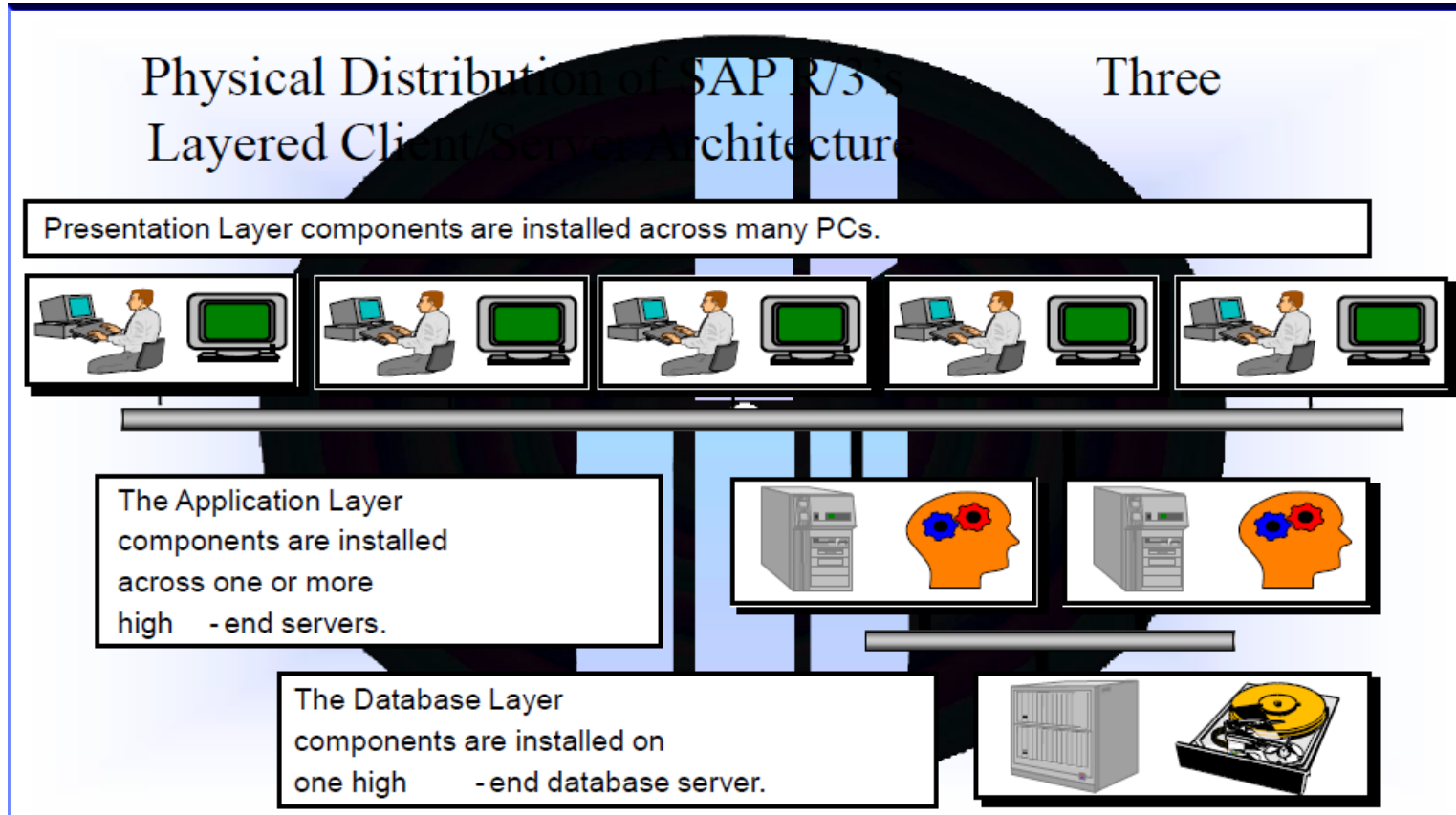
The three Logical Layers of this R/3 Architecture are...

- The Presentation Layer: Collects user input and creates process request.
- The Application Layer: Uses the Application logic of Program to collect and process request.
- The Database Layer: Stores and Retrieves all Data.

SAP R/3 Architecture



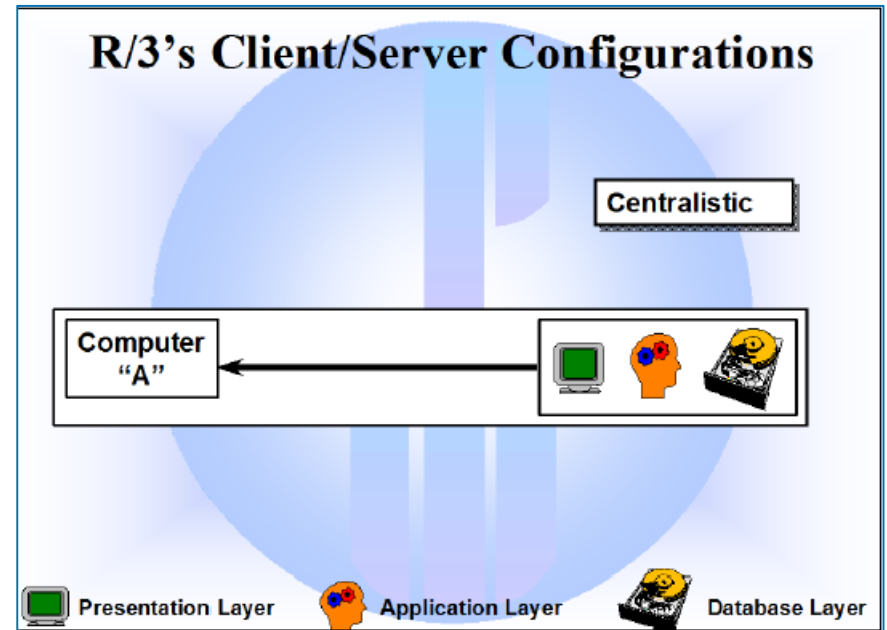
SAP R/3 Architecture





Centralistic:

- All SAP R/3 layers reside on the same physical computer. One important distinguishing characteristic between the various R/3 client/server configurations is "Scalability".
- If a configuration is scalable, adding additional computers to the system will increase the overall performance of the system.
- Centralistic client/server configurations are not scalable at all.
- Consequently, this configuration is never used in a production environment.
- SAP has actually installed an entire SAP R/3 system on a notebook computer for use by SAP's sales representatives

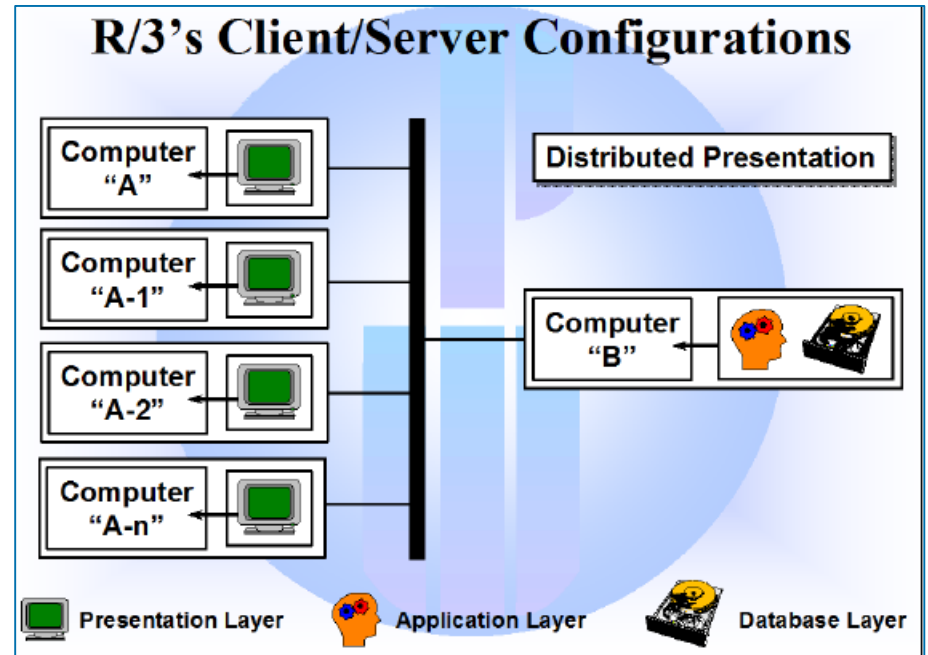


SAP R/3 Architecture



Distributed presentation:

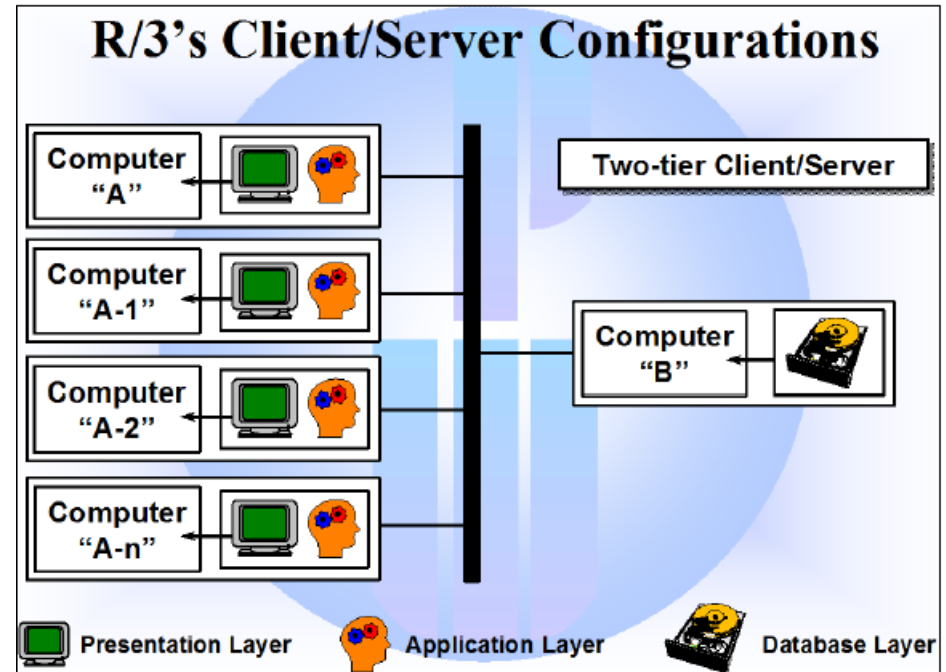
- The presentation layer is “rolled out” to desktop PCs.
- The application layer and the database layer are installed on the same computer.
- In terms of increased performance, the Distributed presentation configuration is no more scalable than the centralistic configuration. This configuration is very “mainframe”-“ish”





Two-tier client/server :

- The presentation and application layers are installed on the same computer. The database layer is installed on a separate computer.
- Two-tier client/server configurations offer increased scalability.
- However, two-tier client/server configurations create other problems.
- The communication across the network between the front-end (presentation and application layers) and the backend (database layer) becomes a bottle-neck very quickly





Three-tier client/server:

- Presentation, application, and database layers run on separate computers.
- Currently, three-tier client/server offers the best solution for most businesses.
- It is highly scalable, and offers better distribution of process requests received from the users.
- The computers in the application layer are often capable of satisfying the users process requests without accessing the database, which in turn boosts performance.

Application Server Architecture



The components of an application server are shown in the figure below. It consists of a dispatcher and multiple work processes.

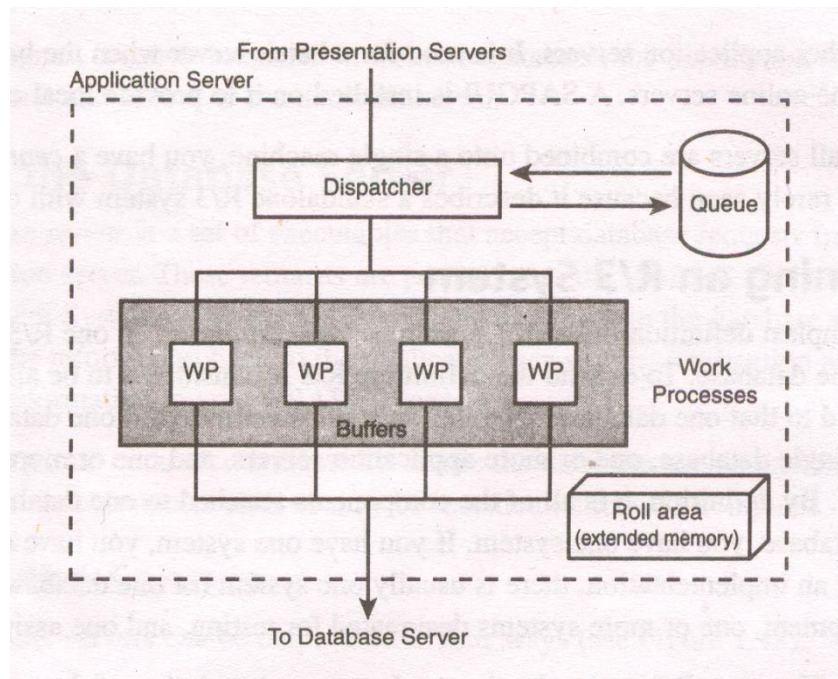


Fig: Application server architecture

Application Server Architecture

All requests that come in from presentation servers are directed first to dispatcher.

The dispatcher writes them first to the dispatcher queue.

The dispatcher pulls the requests from the queue on a first-in, first-out basis.

Each request is then allocated to the first available work process.

A work process handles one request at a time.

To perform any processing for a user's request, a work process needs to address two special memory areas: the *user context* and the *program roll area*.

The *user context* is a memory area that contains information about the user, and the *roll area* is a memory area that contains information about the programs execution

User Context

A *user context* is memory that is allocated to contain the characteristics of a user that is logged on the R/3 system.

It holds information needed by R/3 about the user, such as:

- The user's current settings
- The user's authorizations
- The names of the programs the user is currently running

When a user logs on, a user context is allocated for that logon.

When they log off, it is freed.

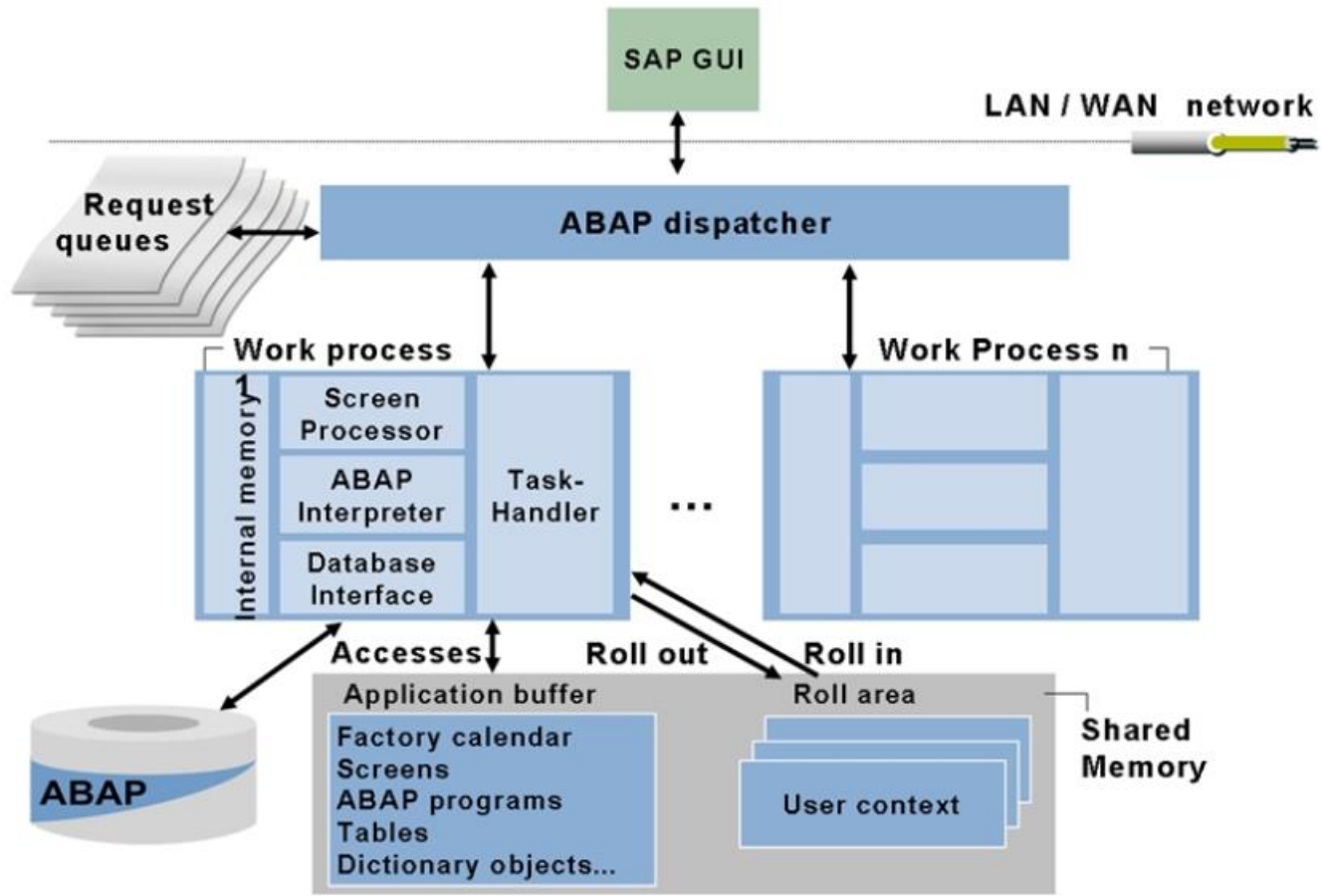
Roll Area

A roll area is memory that is allocated by a work process for an instance of a program.

It holds information needed by R/3 about the program's execution, such as:

- The values of the variables
- The dynamic memory allocations
- The current program pointer
 - Each time a user starts a program, a roll area is created for that instance of the program.
 - If two users run the same program at the same time, two roll areas will exist-one for each user.
 - The roll area is freed when the program ends.
 - The roll area and the user context play an important part in dialog step processing

Process Flow for Requests

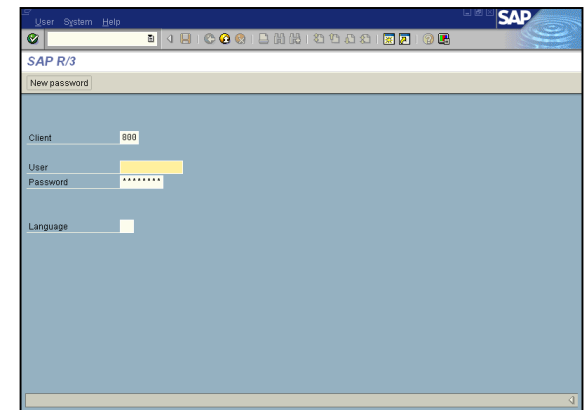


Logon Client

The term *logon client* has nothing to do the Client/Server-it is completely different.

The *logon client* refers to the number that the user types in the *Client* field on the logon screen.

The number entered here by the user corresponds to a set of rows within each client- dependent table within the database



Client-Dependent and Client-Independent Tables

There are two types of tables in the R/3 database: *client-dependent* and *client-independent*.

A table is client-dependent if the first field is of type CLNT.

The length will always be 3; and by convention, this field is named *mandt*.

If the first field is not of type CLNT, the table is client independent

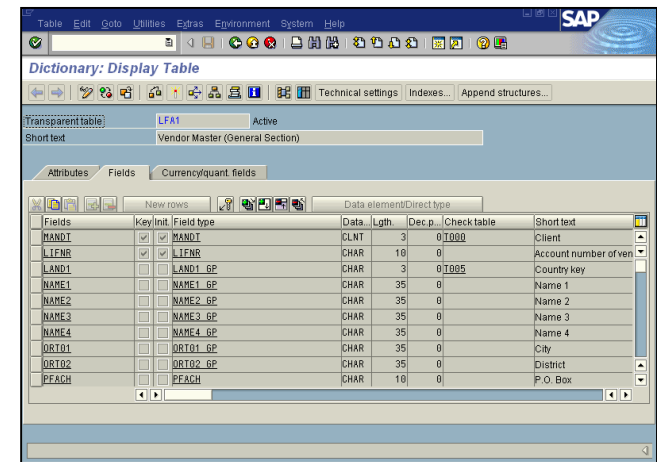


Table: Dictionary: Display Table

Transparent table: LFA1 Active

Short text: Vendor Master (General Section)

Attributes Fields Currency/quant. fields

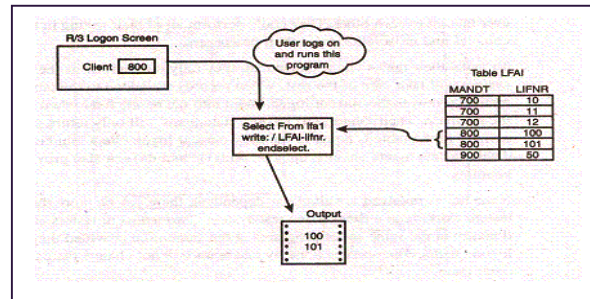
Fields	Key/Int.	Field type	Data	Length	Dec. p.	Check table	Short text
MANDT	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	01000	Client
LIFNR	<input checked="" type="checkbox"/>	LIFNR	CHAR	10	0		Account number of ven
LAND1	<input type="checkbox"/>	LAND1 GP	CHAR	3	0	01005	Country key
NAME1	<input type="checkbox"/>	NAME1 GP	CHAR	35	0		Name 1
NAME2	<input type="checkbox"/>	NAME2 GP	CHAR	35	0		Name 2
NAME3	<input type="checkbox"/>	NAME3 GP	CHAR	35	0		Name 3
NAME4	<input type="checkbox"/>	NAME4 GP	CHAR	35	0		Name 4
ORT01	<input type="checkbox"/>	ORT01 GP	CHAR	35	0		City
ORT02	<input type="checkbox"/>	ORT02 GP	CHAR	35	0		District
PFACH	<input type="checkbox"/>	PFACH	CHAR	10	0		P.O. Box

Client-Dependent and Client-Independent Tables

In the figure, the user logs on to client 800 and runs the program shown. This program selects rows from table lfa1 and writes out lfa1-lifnr. When this program is run, only two rows are selected: only those whose mandt equals 800. This happens automatically because the first field in the table is of the type CLNT.

There are five rows in the table, but the program writes out only those rows where mandt equals 800.

If the user were to log on to client 700 and run the same program, three rows of data would be found and written out. If the user were to log on to client 900, only one row of data would be found



Client-Dependent and Client-Independent Tables



The logon client mechanism divides the rows within a client-dependent table into distinct groups.

To access a different set of data, the user logs on and specifies a different client number.

The user master records (containing R/3 user IDs) are client-dependent.

Therefore, to gain access to a client, the system administrator must create a new user ID for you within that client

Developers and testers use the logon client mechanism to create and access multiple, independent set of data within a single table

Client-Dependent and Client-Independent Tables

The average R/3 installation has three systems: *development*, *test*, and *production*.

By default, each system comes with three clients installed: 000, 001, and 066.

It is common to have from three to six clients in the development and test systems, but rarely will you see more than one client in production

Using SAP's Open SQL

ABAP/4 code is portable between databases.

To access the database in an ABAP/4 program you will code SAP's *Open SQL*.

Open SQL is a subset and variation of ANSI SQL.

The ABAP/4 interpreter passes all Open SQL statements to the database interface part of the work process

There, they are converted to SQL that is native to the installed RDBMS.

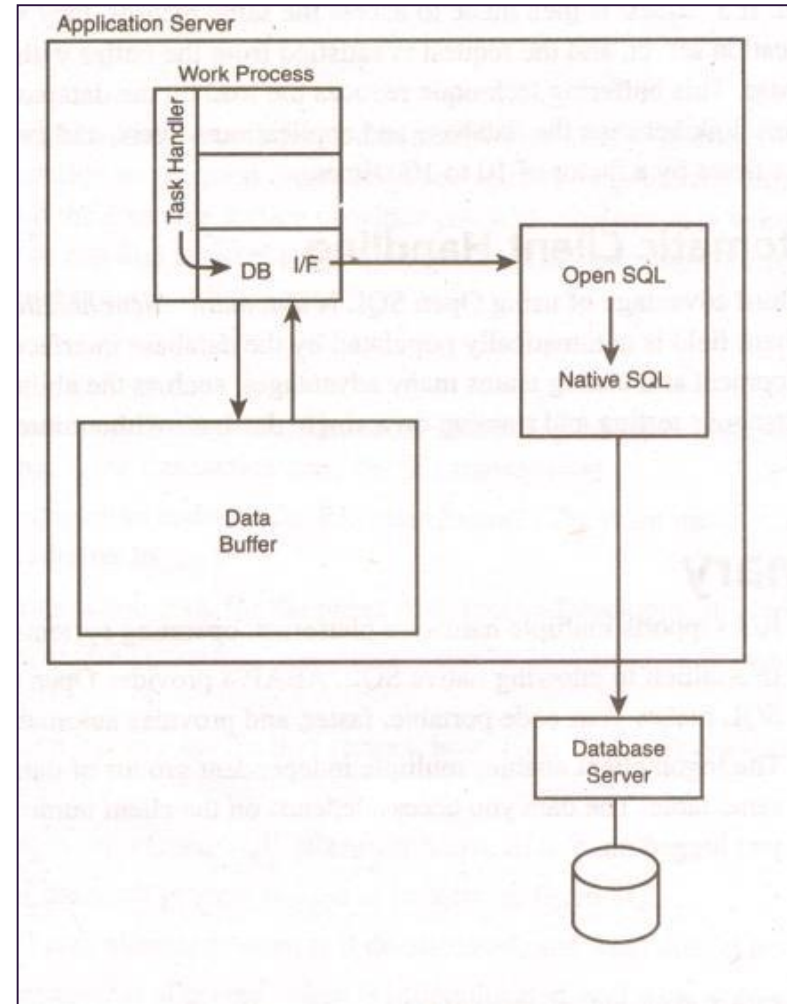
For example, if you were running an Oracle database, your ABAP/4 Open SQL would be converted by the database interface to Oracle SQL statements.

Using SAP's Open SQL

If you use Open SQL, your SQL statements will be passed to the database interface.

Using Open SQL has three main advantages.

All of these advantages are implemented via the database interface



Portability

The first advantage is the fact that your SQL statements will be portable between databases.

For example, if for some reason your company wanted to switch from an Oracle to an Informix database, it could change the database, and your ABAP/4 code would continue to run without modification

Buffering Data on the Application Server

Secondly, the database interface buffers information from the database on the application server.

When data is read from the database, it can be stored in the buffers on the application server.

If a request were then made to access the same records, they would already be on the application server, and the request is satisfied from the buffer without having to go to the database.

This buffering techniques reduces the load on the database sever and on the network link between the database and the application servers, and can speed up database access time by a factor of 10 to 100 times.

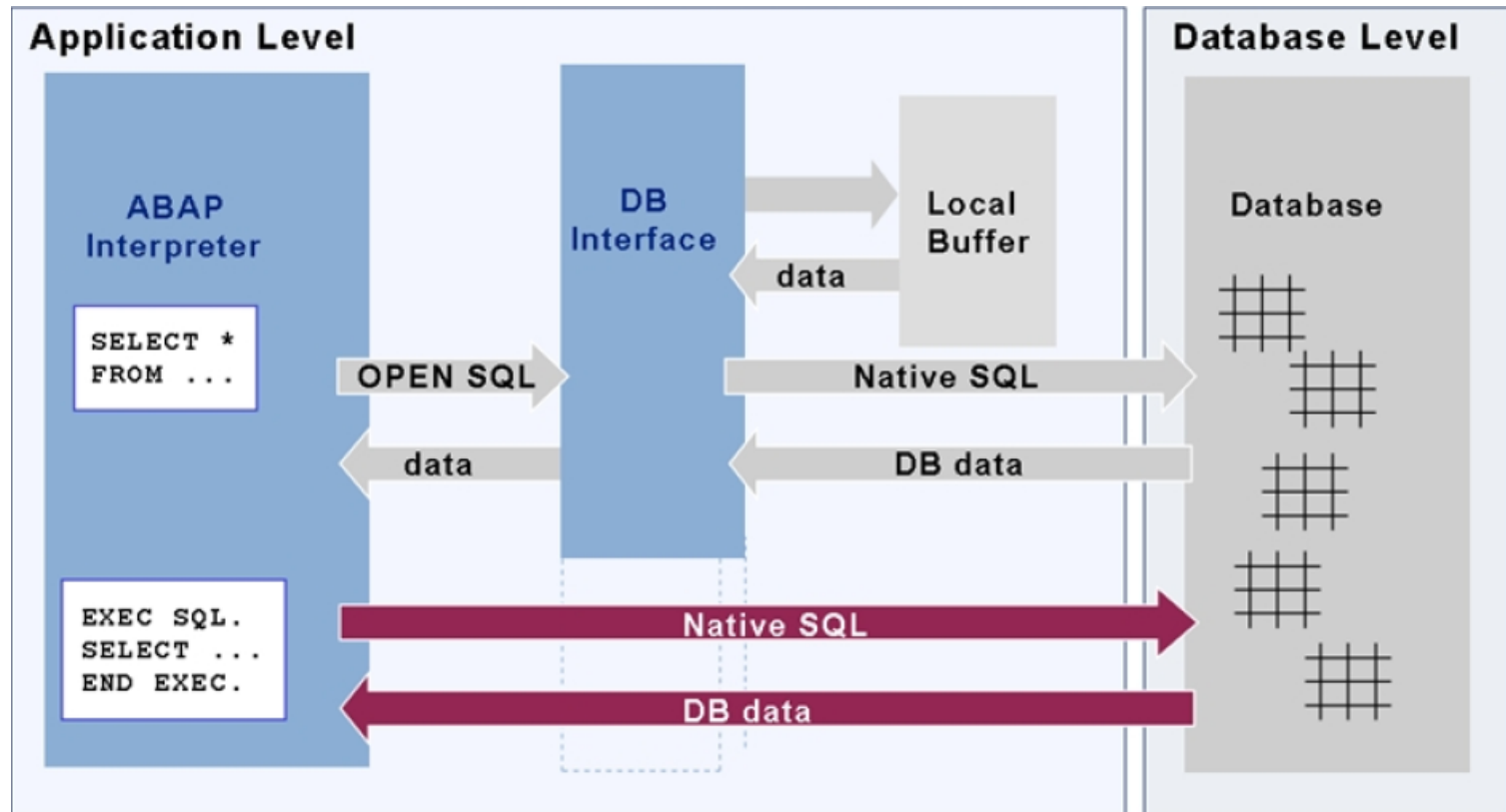
Automatic Client Checking

The third advantage of using Open SQL is *automatic client handling*.

With Open SQL, the client field is automatically populated by the database interface.

This gives your development and testing teams many advantages, such as ability to perform multiple simultaneous testing and training on a single database without interference from each other

Database Query Flow



SAP Logon

A utility to logon to SAP

- Choose an available SAP system
- Program connects to the message server of that system and obtains the address of a suitable Application Server
- Starts a SAP GUI (Graphical User Interface)
- SAP GUI Starts the logon Screen
- The user can open multiple sessions
- Applications are run within a session

The SAP GUI is based on Windows Style and is available for several Platforms, providing the same functions for each

Basic Navigation

Logging on to SAP

- Client
- Username
- Password
- Logon Language

Client	<input type="text" value="100"/>
User	<input type="text" value=""/>
Password	<input type="password" value="*****"/>
Logon Language	<input type="text" value="EN"/>

Demo

SAP Logon





Transaction Code

- Acronym to access menu path
- Sequence of Screens with Input and Output fields for Processing

Possible Command Field Entries

- /nxxxx – to call Transaction xxxx
- /n – cancel Transaction
- /oxxxx – to Call Transaction xxxx in a new Session
- /o – display an overview of Sessions



Several options are available to log off from system

- Menu Bar System Logoff
- Choose Yellow Arrow in the SAP EASY ACCESS Menu. If several sessions are open, it only closes the session
- Enter /nend in the command field
- Enter /nex in the command field.



The Commonly Used Transaction Codes are

- SE80 : Object Navigator (ABAP Development Workbench)
- SE38 : ABAP Editor
- SE37 : Function Builder
- SE11 : ABAP Dictionary
- SE21 : Package Builder
- SE91 : Message Maintenance

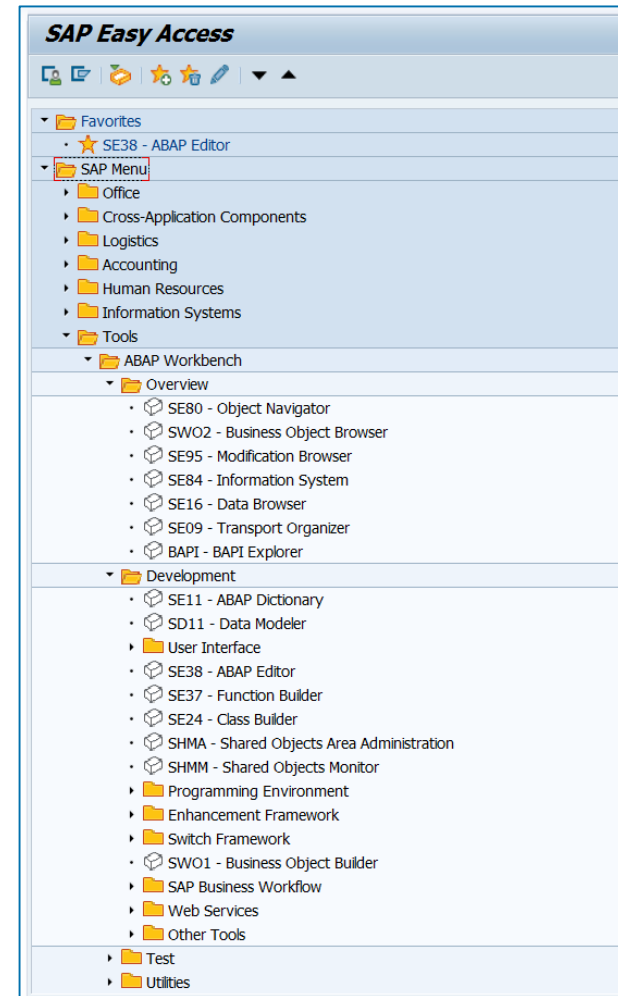
ABAP Workbench tools



The ABAP Workbench is a collection of tools you use to develop, test and run ABAP programs

Frequently Used Tools

- ABAP Editor
- ABAP Dictionary
- Screen Painter
- Menu Painter
- Function Builder



Review Question

1. _____ is like an operating system for R/3.
2. An _____ interprets the ABAP/4 programs and manage the input and output for them.
3. A _____ is memory that is allocated to contain the characteristics of a user that is logged on the R/3 system.
4. The _____ is used to cancel the transaction.
5. _____ is a collection of tools you use to develop, test and run ABAP programs



Summary

In this lesson, you have learnt:

- The R/3 System Architecture
- The Application Server Architecture
- The meaning of Logon Client
- Advantages of using SAP's Open SQL
- The Basics of SAP
- How to Log on to SAP and do the Basic Navigations

