Capgemini

SAP BASIS Introductory Training Program

Day 4 : Agenda

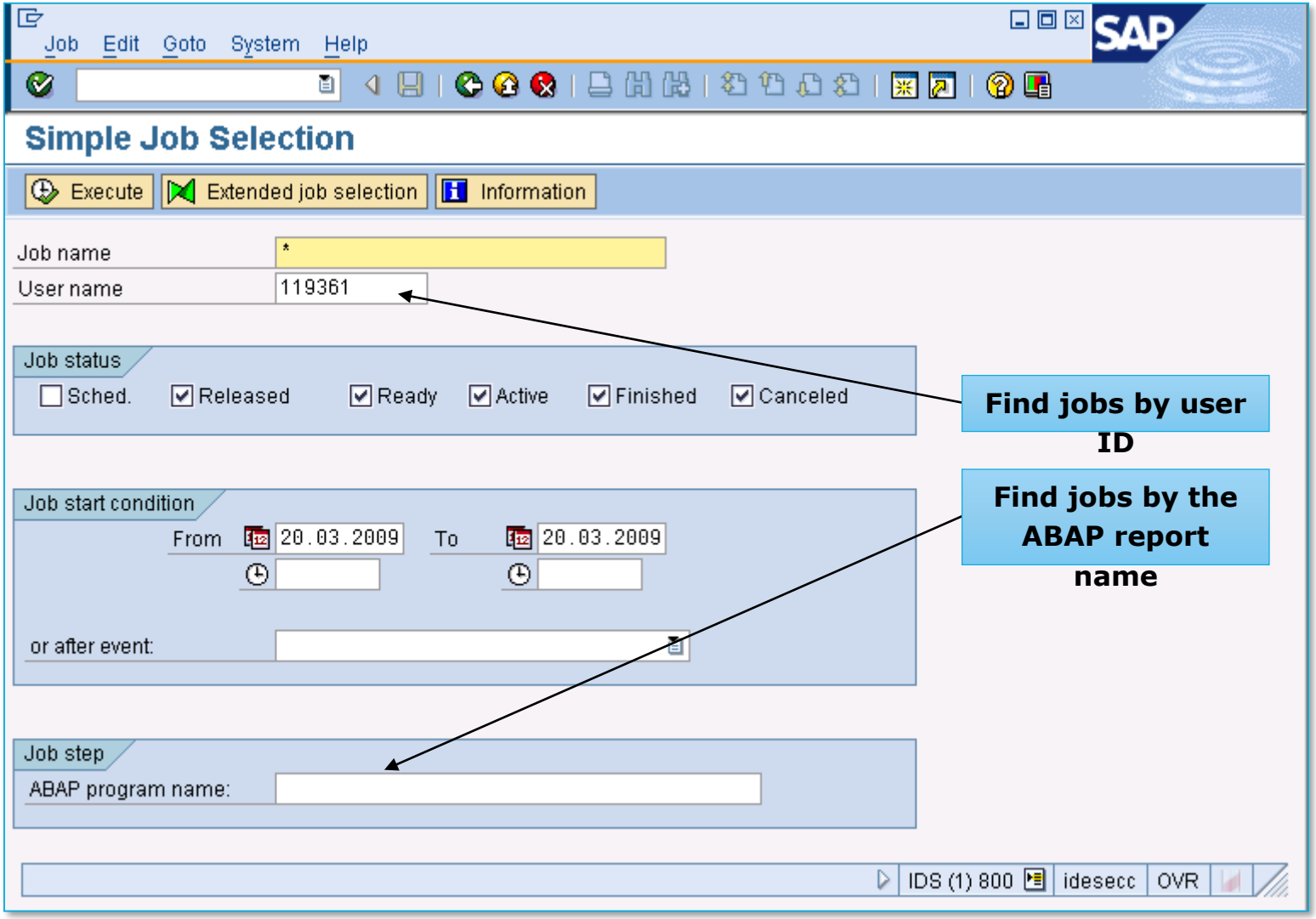
- System Monitoring and troubleshooting
- Break
- Web Enablement and Communication in SAP
- Lunch Break
- AS JAVA Architecture and Processes
- Break
- AS JAVA Architecture and Processes
- Exercise & Break Out Session

System Monitoring & Troubleshooting

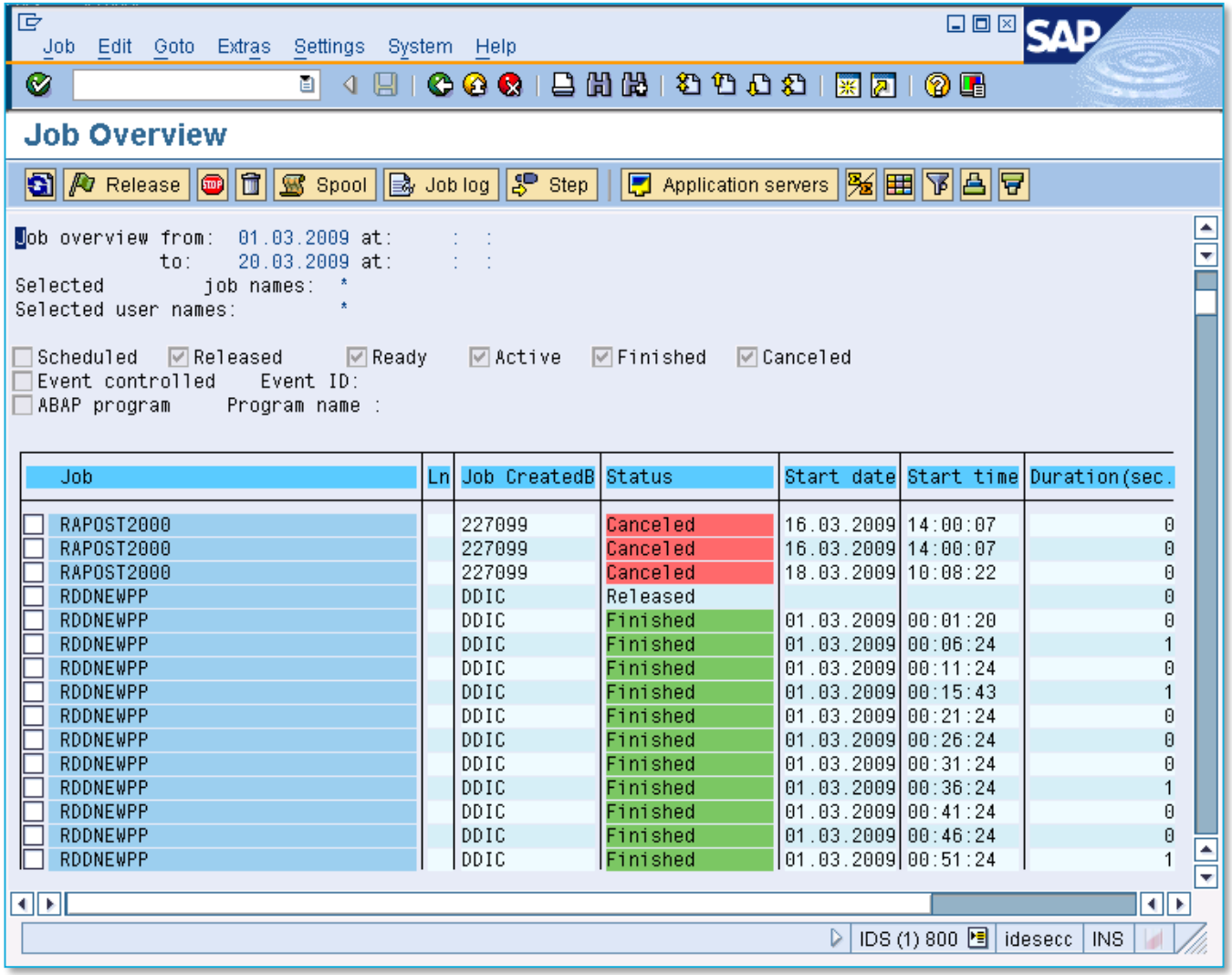
Monitoring Background Jobs – SM37

- Monitoring background jobs is an essential part of performance monitoring
- The transaction SM37 can be used to check the status of jobs running in the system
- Background jobs are essentially programs that are run by background work processes
  ("B" of DVEBMGS)
- Since the programs may process large volumes of data, background work processes are allocated a higher memory limit for usage of heap and shared memory
- Background work processes take up a higher percentage of CPU utilization, so care must be taken to run high volume background jobs during a quiet window, that is, a duration where dialog users are not active in the system
- Typically in customer environments, such jobs are run during late evening, night and in early part of the morning

Using SM37

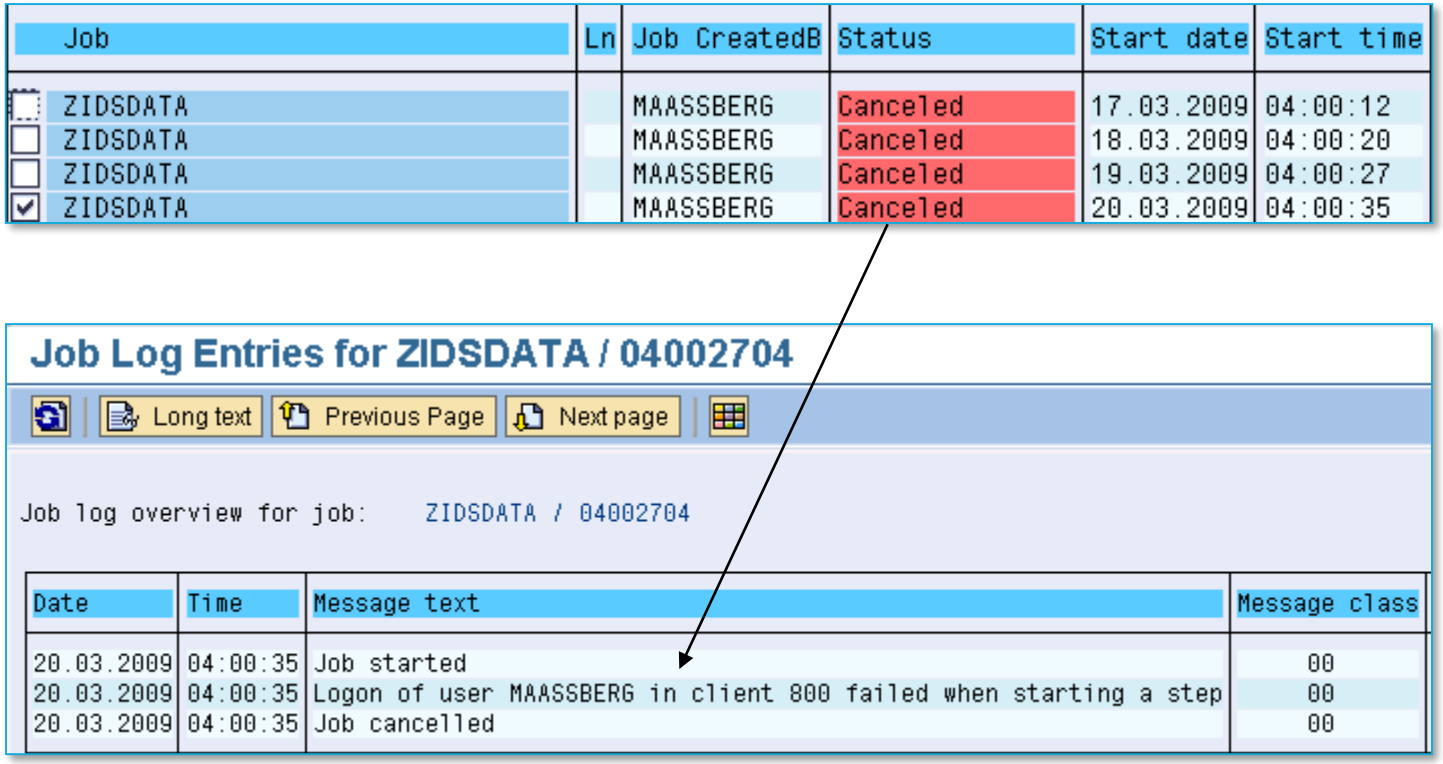

Job Edit Goto System Help
SAP
Simple Job Selection
Execute
Extended job selection
Information
Job name
*
User name
119361
Job status
Sched.
Released
Ready
Active
Finished
Canceled
Job start condition
From
20.03.2009
To
20.03.2009
or after event:
Job step
ABAP program name:
IDS (1) 800
idesecc
OVR
Find jobs by user ID
Find jobs by the ABAP report name

Features of SM37



Features of SM37

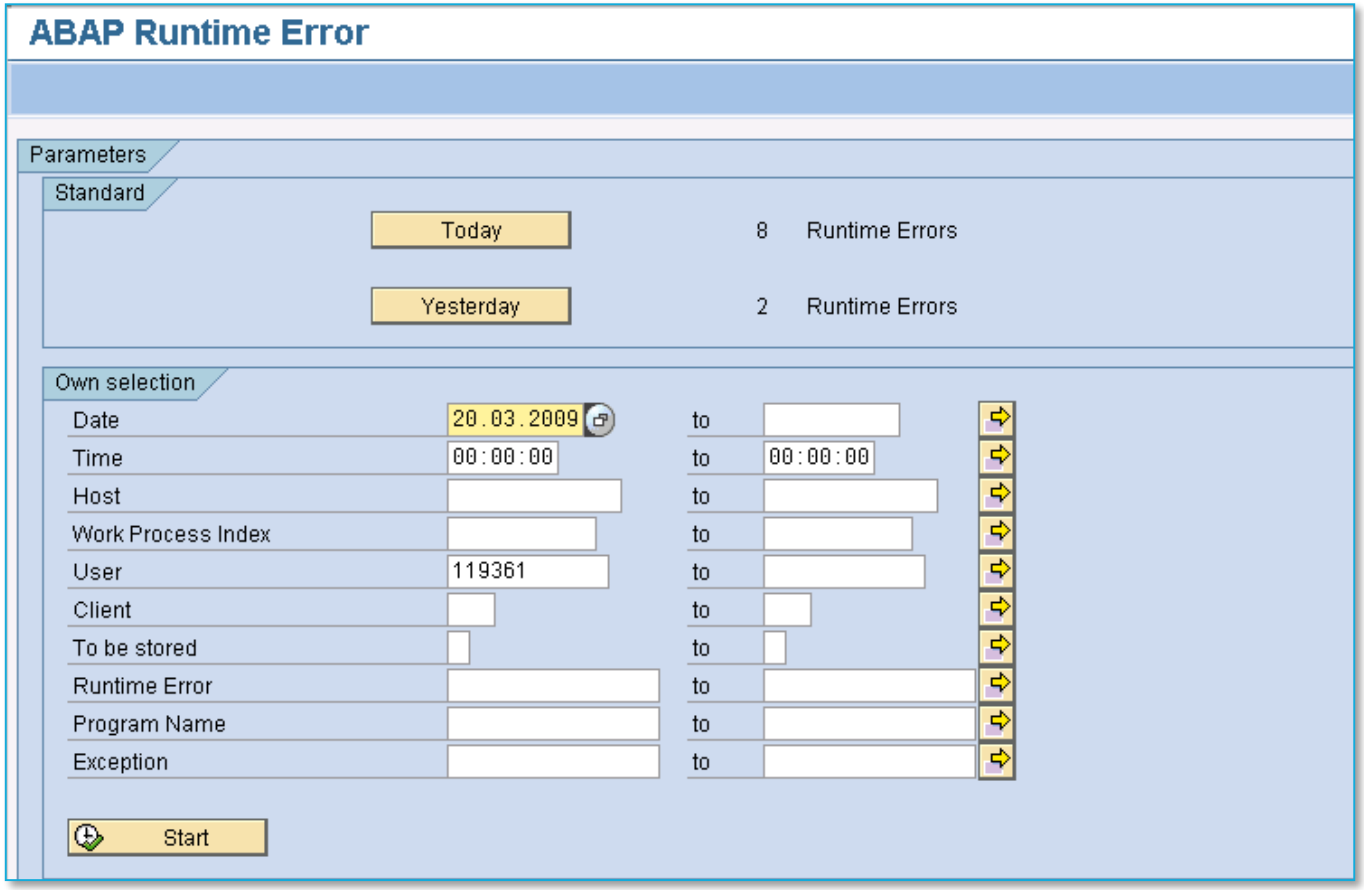
- The job log provides critical information from a troubleshooting perspective. In the case of cancelled jobs, the ABAP dump name is provided.



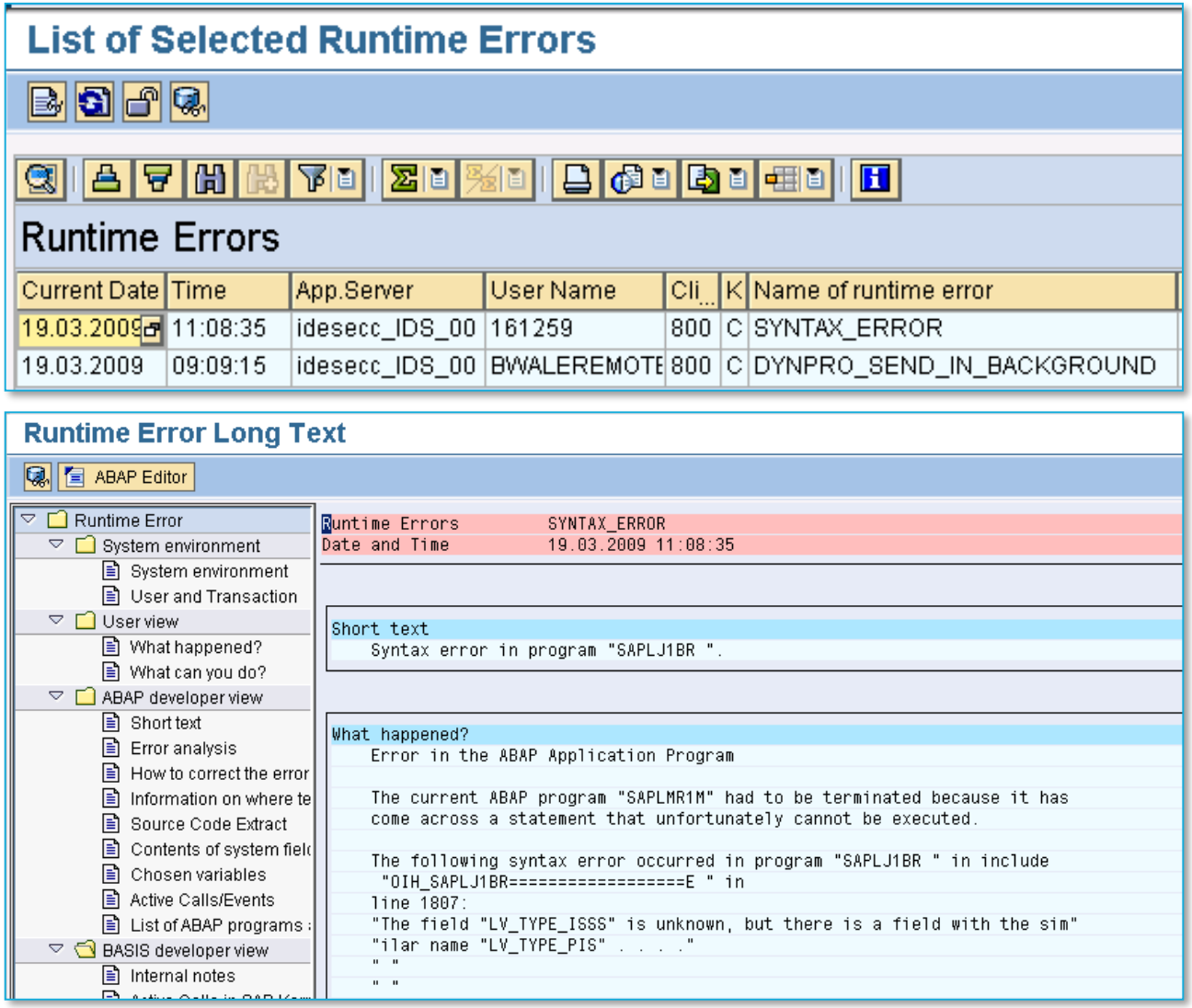
Job	Ln	Job CreatedB	Status	Start date	Start time
ZIDSDATA		MAASSBERG	Canceled	17.03.2009	04:00:12
ZIDSDATA		MAASSBERG	Canceled	18.03.2009	04:00:20
ZIDSDATA		MAASSBERG	Canceled	19.03.2009	04:00:27
ZIDSDATA		MAASSBERG	Canceled	20.03.2009	04:00:35

ABAP Dumps Analysis using ST22

- ABAP Dumps provide detailed information on the nature of the program failure



ABAP Dump Details



Troubleshooting ABAP errors using OSS Notes

What is OSS?

- OSS is an abbreviation for Online Support Services

What are OSS Notes ?

- SAP OSS Notes are technical solution briefs that provides information on the cause of an error, its solution and a recommendation on what technical steps are required to be taken to fix the problem

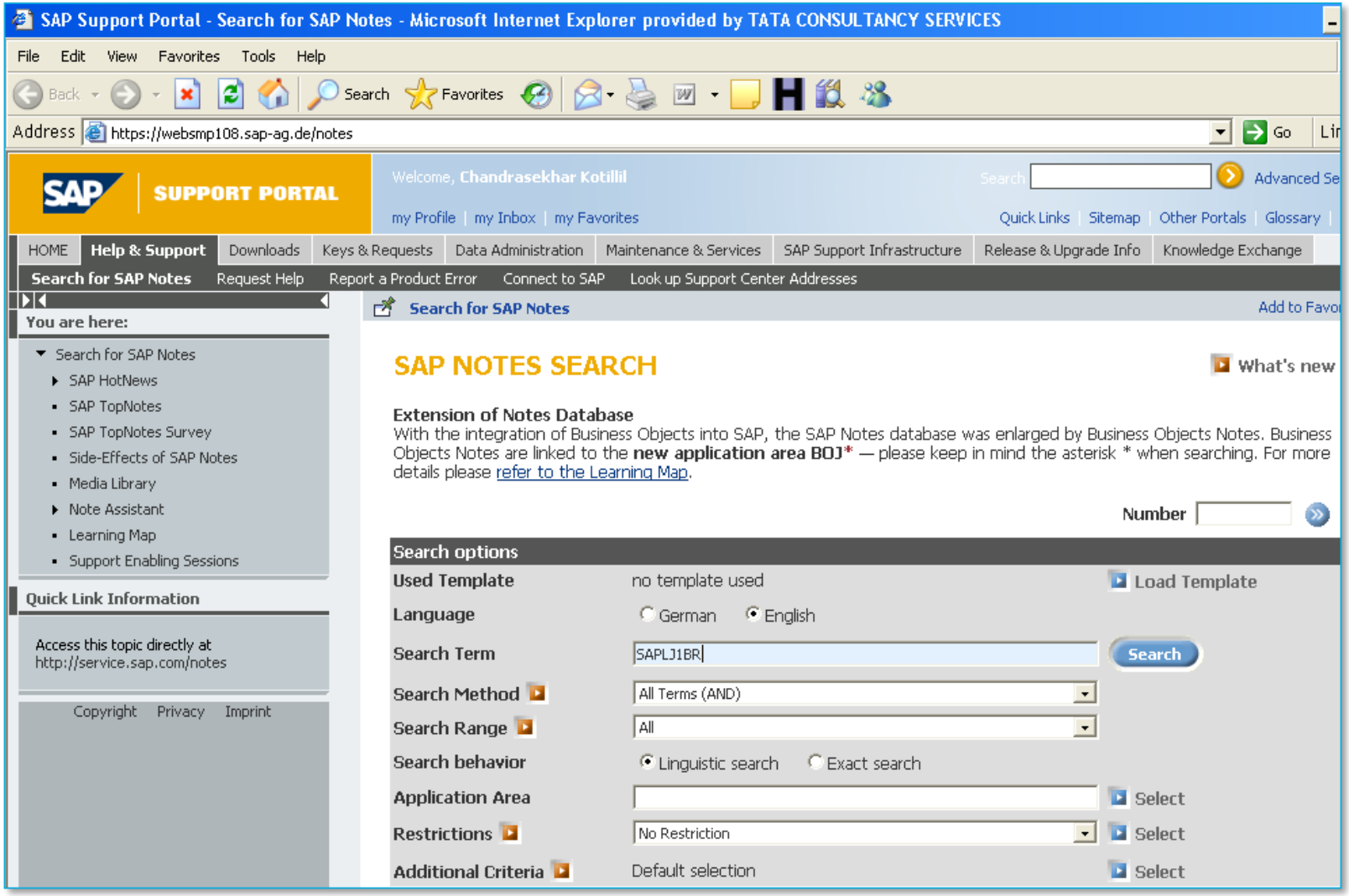
Where can I access OSS Notes ?

- OSS Notes are accessible via SAP Service Market Place Portal with the URL: https://service.sap.com/notes

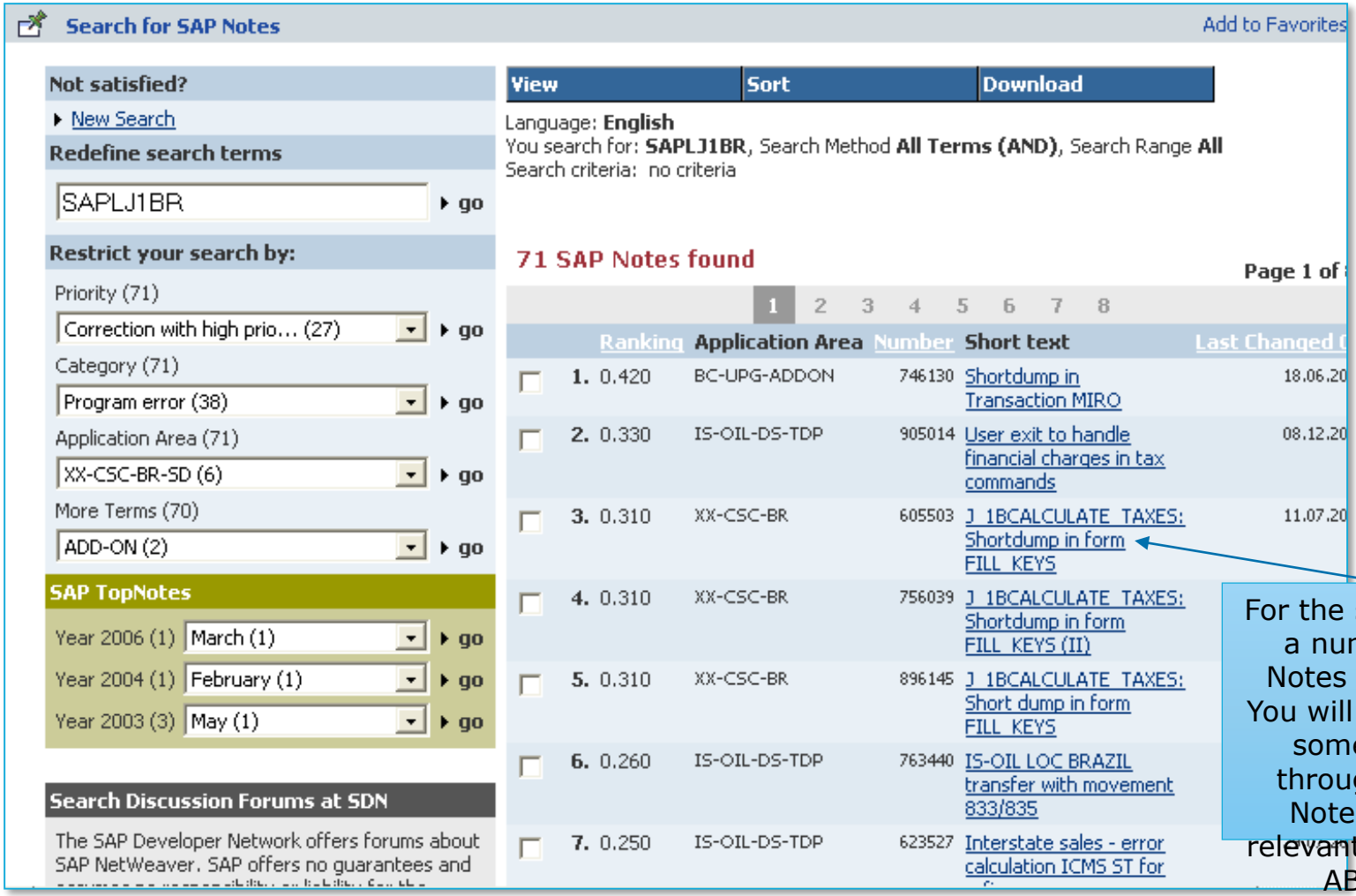
How to I gain access to the SAP Service Market Place ?

- You can request the SAP Super Administrator of the SAP environment for a “S” user ID and password with the authorization for viewing OSS Notes

Sample Usage of OSS Notes

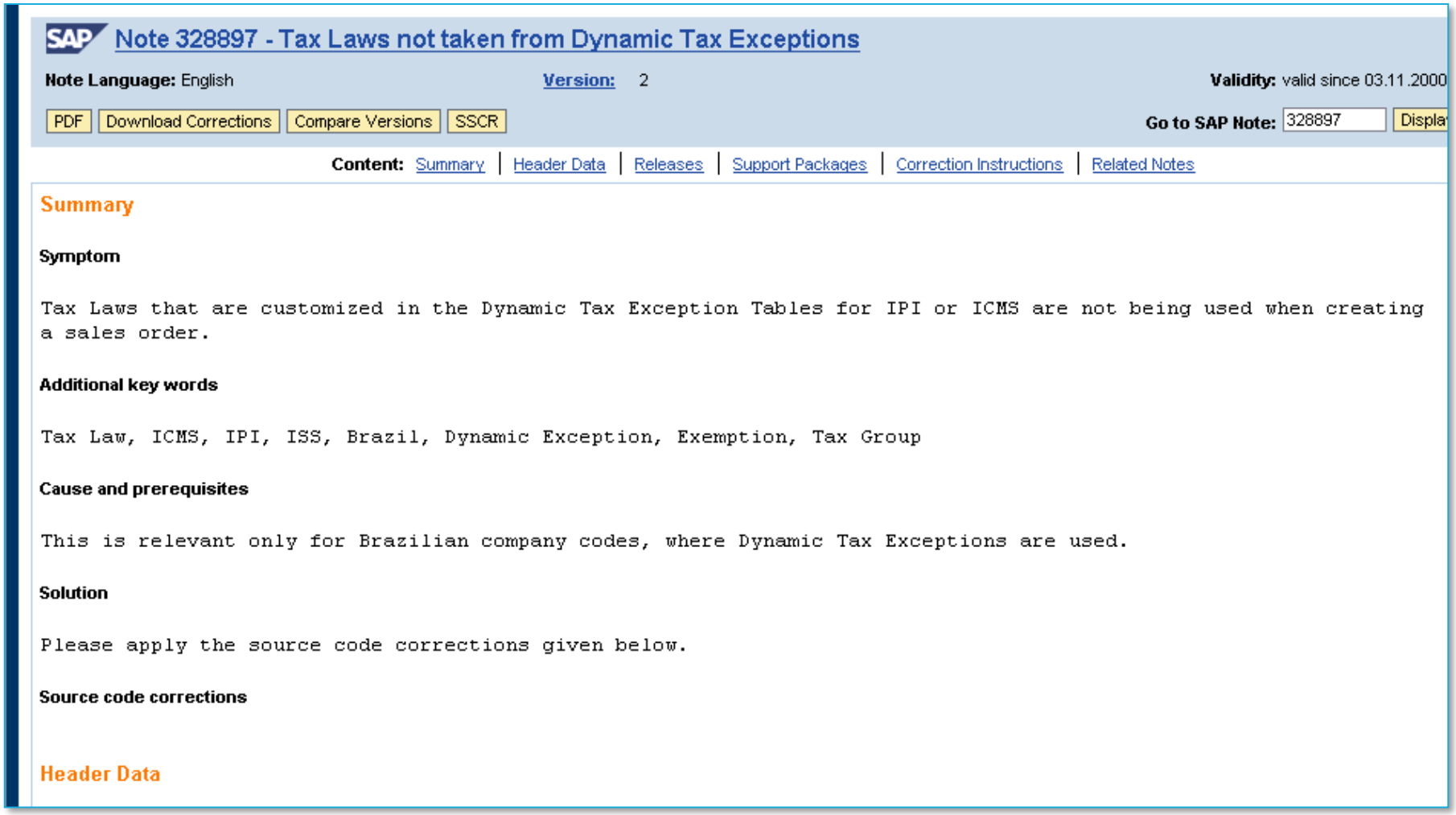


Browse through OSS Notes



For the search criteria, a number of OSS Notes are provided. You will need to spend sometime going through each OSS Note to see if its relevant to solving the ABAP dump

Sample OSS Note



Note 328897 - Tax Laws not taken from Dynamic Tax Exceptions

Note Language: English **Version:** 2 **Validity:** valid since 03.11.2000

PDF | Download Corrections | Compare Versions | SSCR **Go to SAP Note:** 328897 Displa

Content: Summary | Header Data | Releases | Support Packages | Correction Instructions | Related Notes

Summary

Symptom

Tax Laws that are customized in the Dynamic Tax Exception Tables for IPI or ICMS are not being used when creating a sales order.

Additional key words

Tax Law, ICMS, IPI, ISS, Brazil, Dynamic Exception, Exemption, Tax Group

Cause and prerequisites

This is relevant only for Brazilian company codes, where Dynamic Tax Exceptions are used.

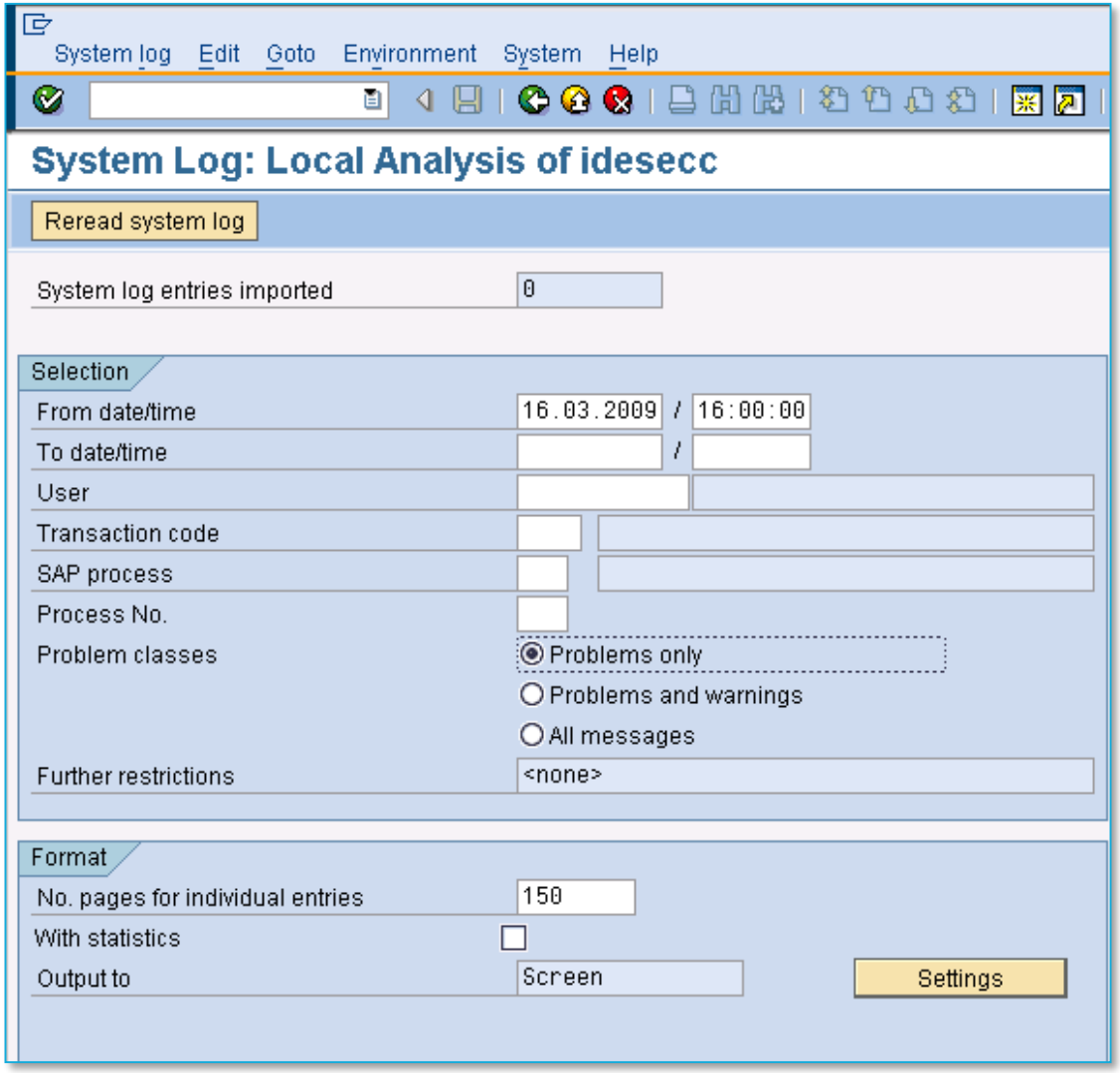
Solution

Please apply the source code corrections given below.

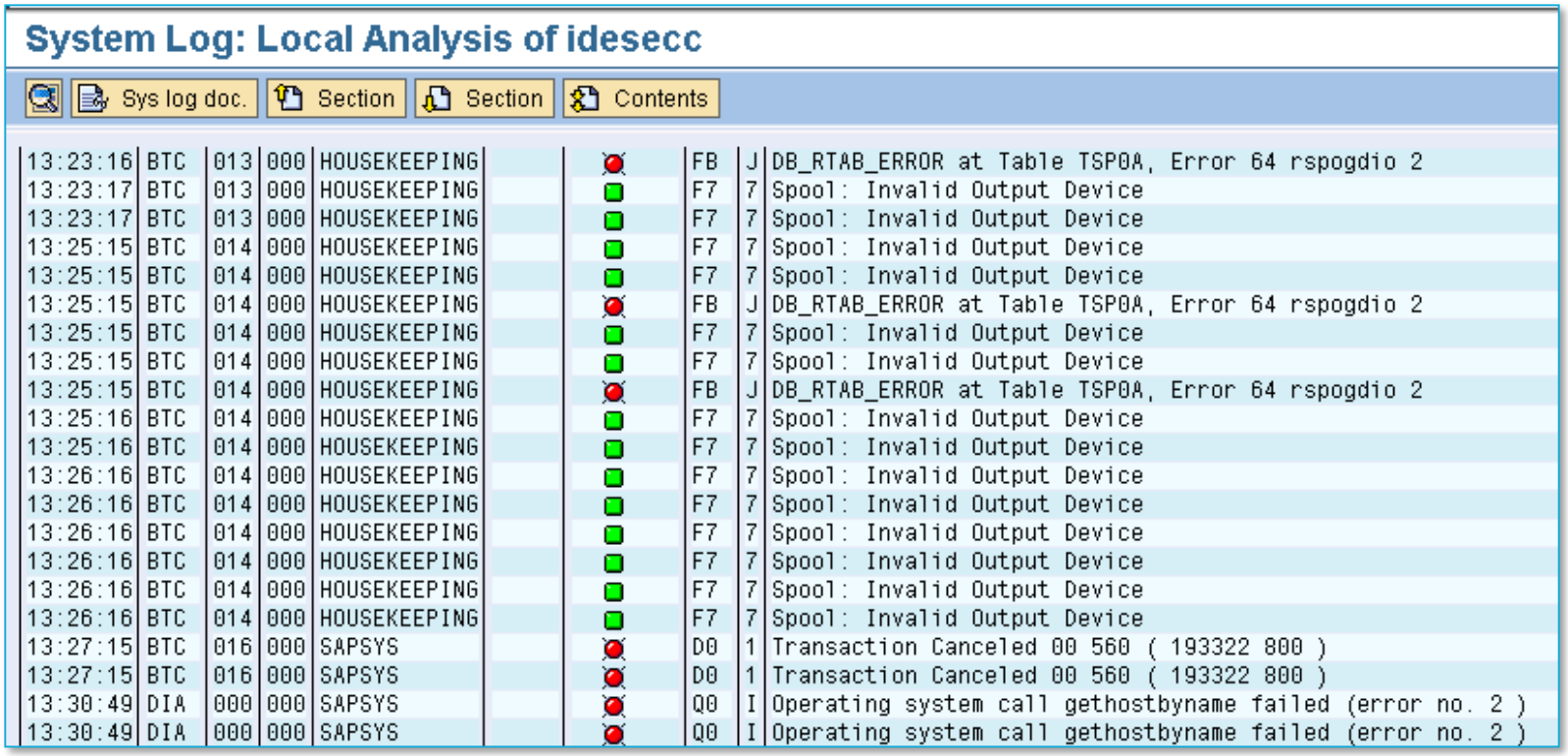
Source code corrections

Header Data

System Logs Analysis using SM21



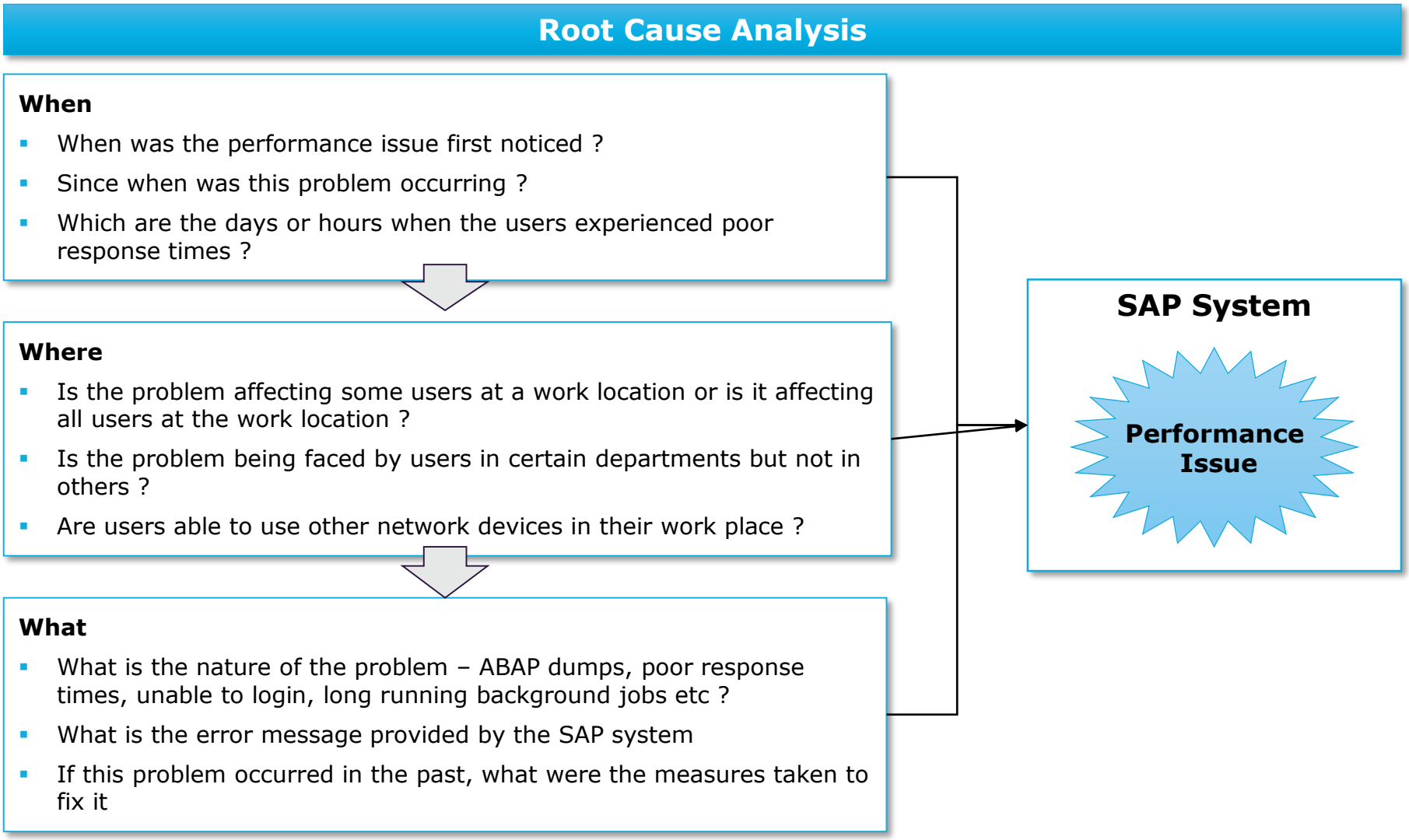
Details in SM21



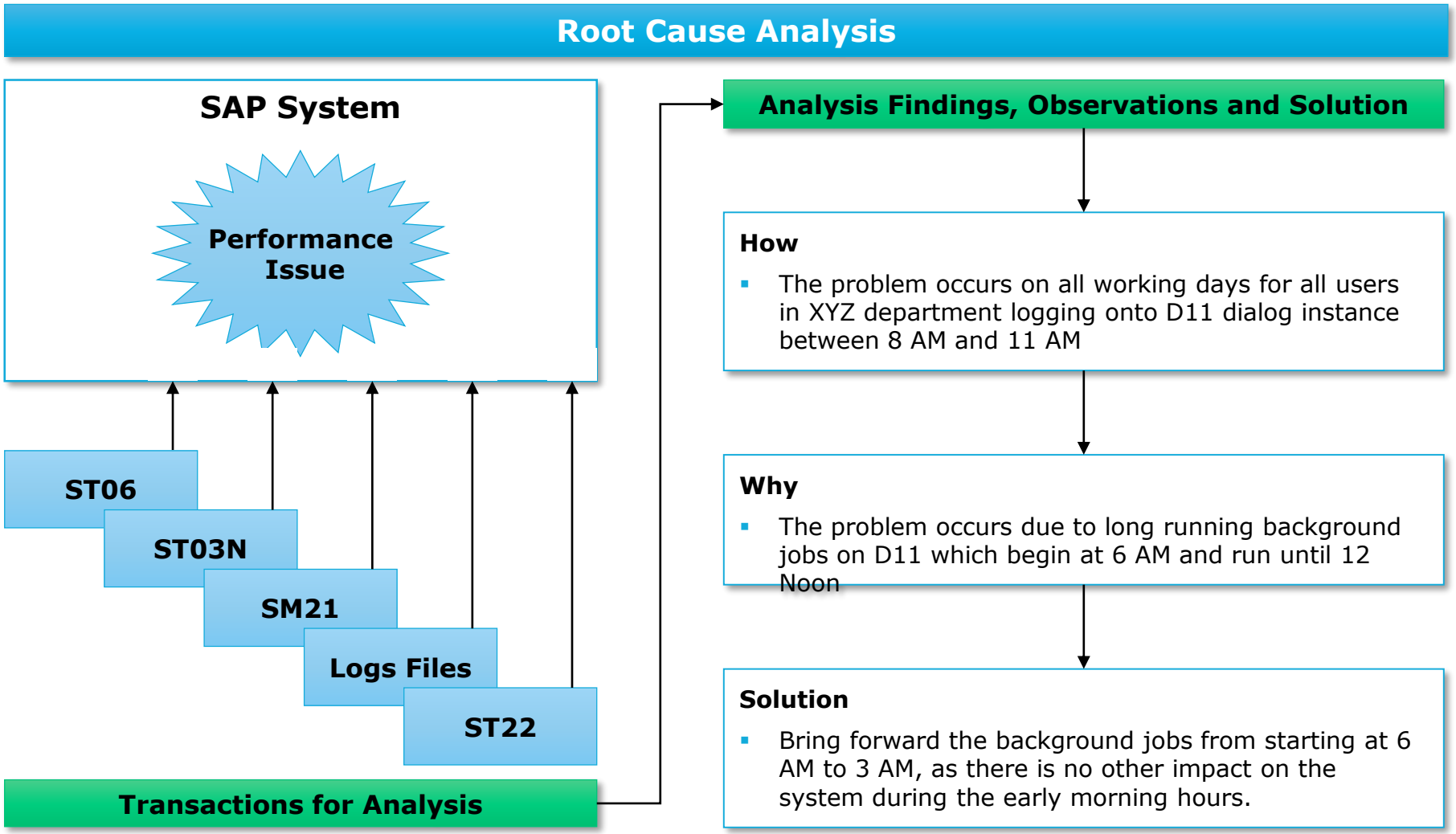
The information is read from the dev_w<x>, current work process log file

- This symbol signifies an error of “high priority”
- This symbol signifies a “warning”, which is of low priority

Troubleshooting & Root Cause Analysis



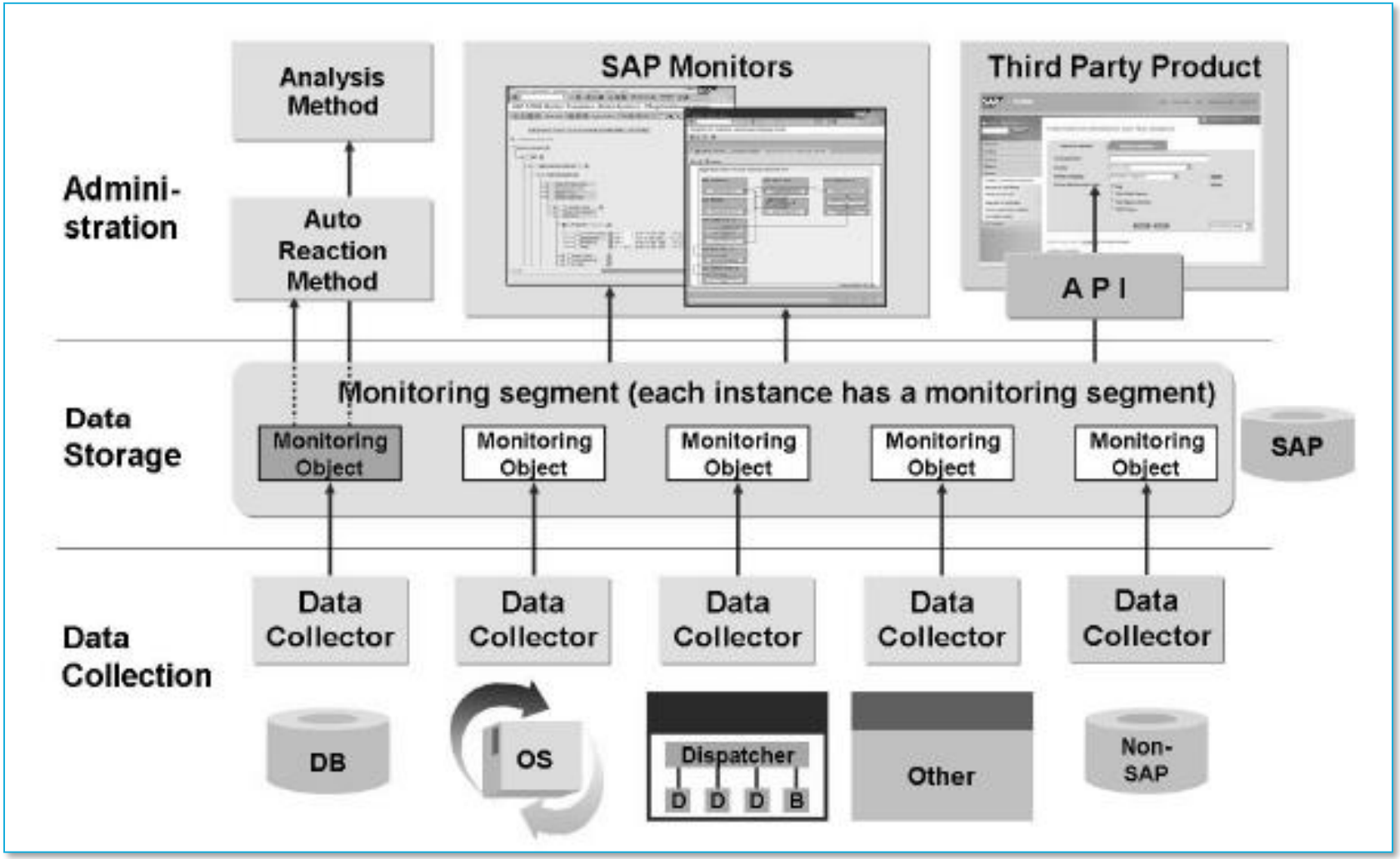
Troubleshooting & Root Cause Analysis



CCMS Monitoring Overview

- CCMS stands for Computing Center Management System
- CCMS is used for central monitoring of the entire SAP landscape
- CCMS brings under a single transaction all the necessary performance data of SAP systems
- CCMS is started from transaction RZ20
- The system which does the central monitoring is called a “CEN”, i.e. the central system
- CCMS can also monitor the performance of SAP interfaces, third party systems connected to SAP as well as non-SAP applications with which data is exchanged

CCMS Monitoring Architecture



Details of the architecture

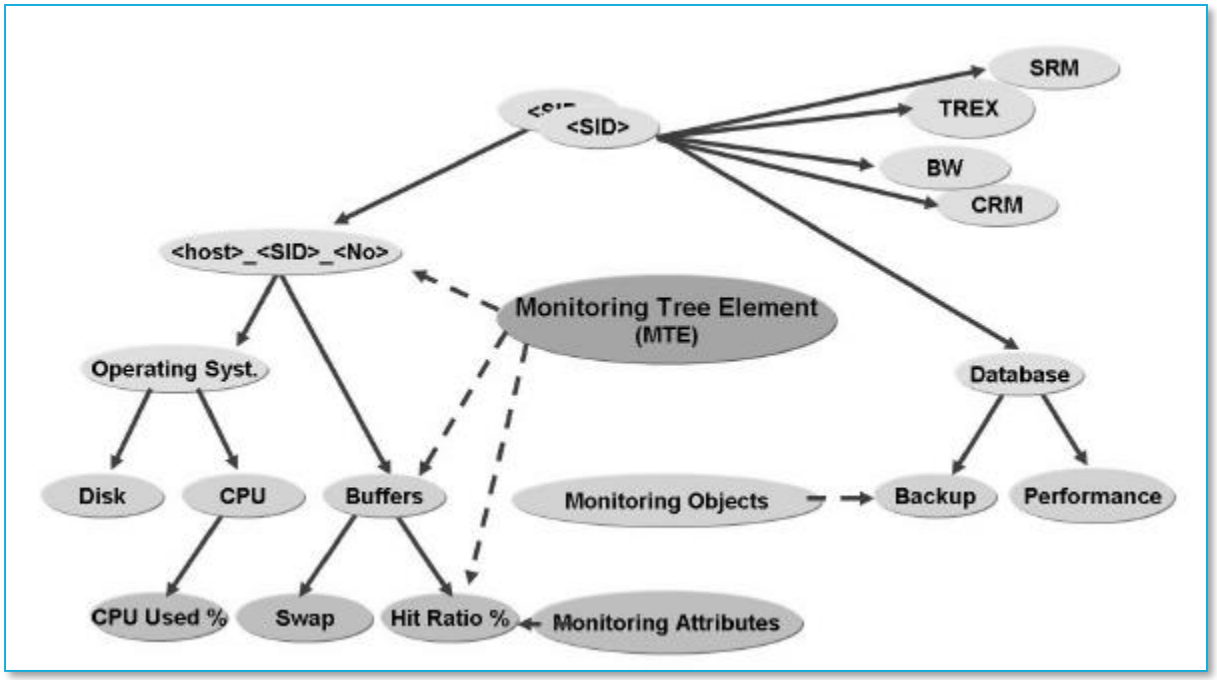
At the **data collection** level, small subareas of an SAP system are monitored by special programs called data collectors. Data collectors can be ABAP, C, or Java programs. There are several hundred data collectors in ABAP alone. Each data collector checks its subcomponent at regular intervals and stores the collected monitoring data in the main memory of its host.

At the **data storage** level, the area of the main memory that contains the monitoring data from the data collector is called the monitoring segment. As the main memory data is always overwritten, it can be permanently copied to database tables. You can then analyze the data later. The data collection and storage elements must be present on every component that is to be centrally monitored.

Caution: Note that every instances of an SAP system has its own monitoring segment in shared memory. This means that for an SAP system with eight instances, there are eight different monitoring segments. The number of monitoring segments is determined by the number of instances. Whether or not several instances run on the same hardware, for instance, is of no significance here.

The **administration level** allows the data from the monitoring segment to be displayed and evaluated. SAP provides an expert tool, the CCMS Alert Monitor (transaction RZ20) as a display transaction. The SAP Solution Manager can show the data in a business process-oriented context. If the system identifies a problem, it can execute a prepared automatic reaction, such as informing the responsible person. The analysis method then helps you to investigate the problem. The CCMS Alert Monitoring Infrastructure can be extended. You can integrate your own components using data collectors that you have written yourself. Third-party vendors and partners can export the monitoring data from the monitoring segment using various interfaces.

CCMS Monitor Structure



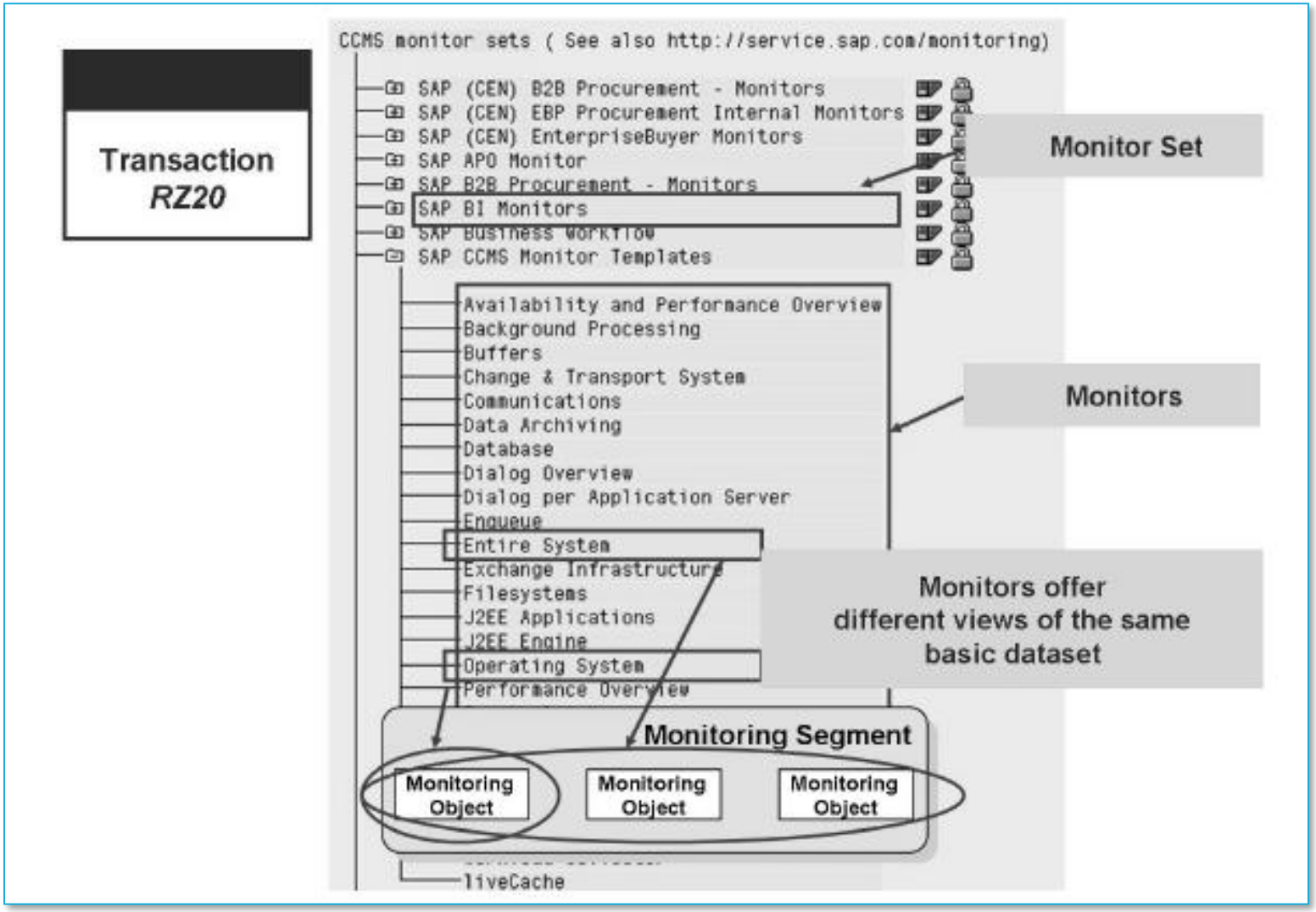
The CCMS Alert Monitor (transaction RZ20) displays the monitoring data from the monitoring segment in a tree structure. The tree structure allows a clear display when you are displaying a large number of measured values.

Any node in the tree is called a Monitoring Tree Element (MTE)

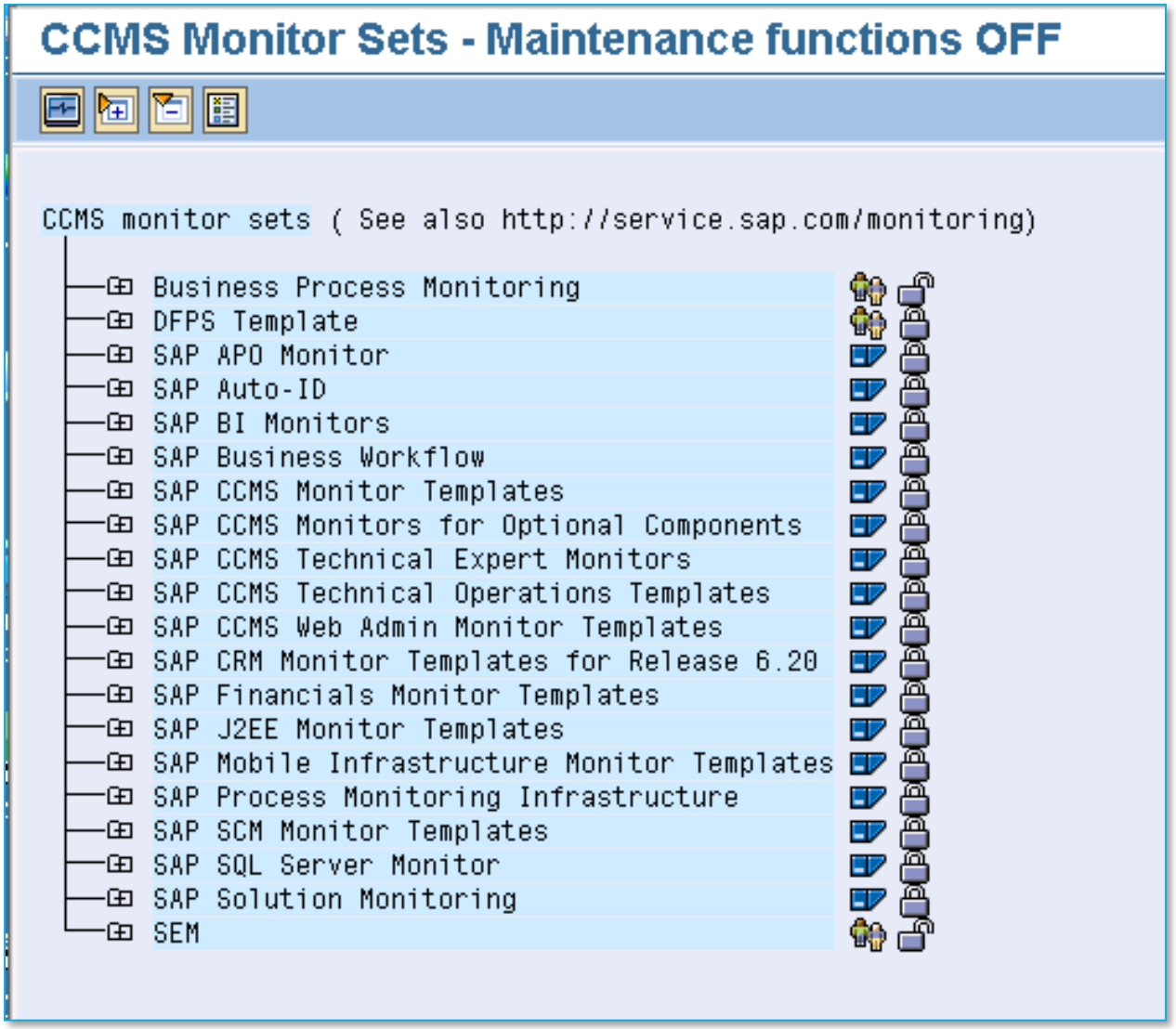
The measured values that are collected by the data collectors are displayed at the lowest level in the leaves of the tree. The leaves are known as monitoring attributes

Threshold values can be stored for a monitoring attribute. SAP delivers default threshold values. However, in order to customize the monitor as well as possible for your system environment, you should check these threshold values, and adjust them if required

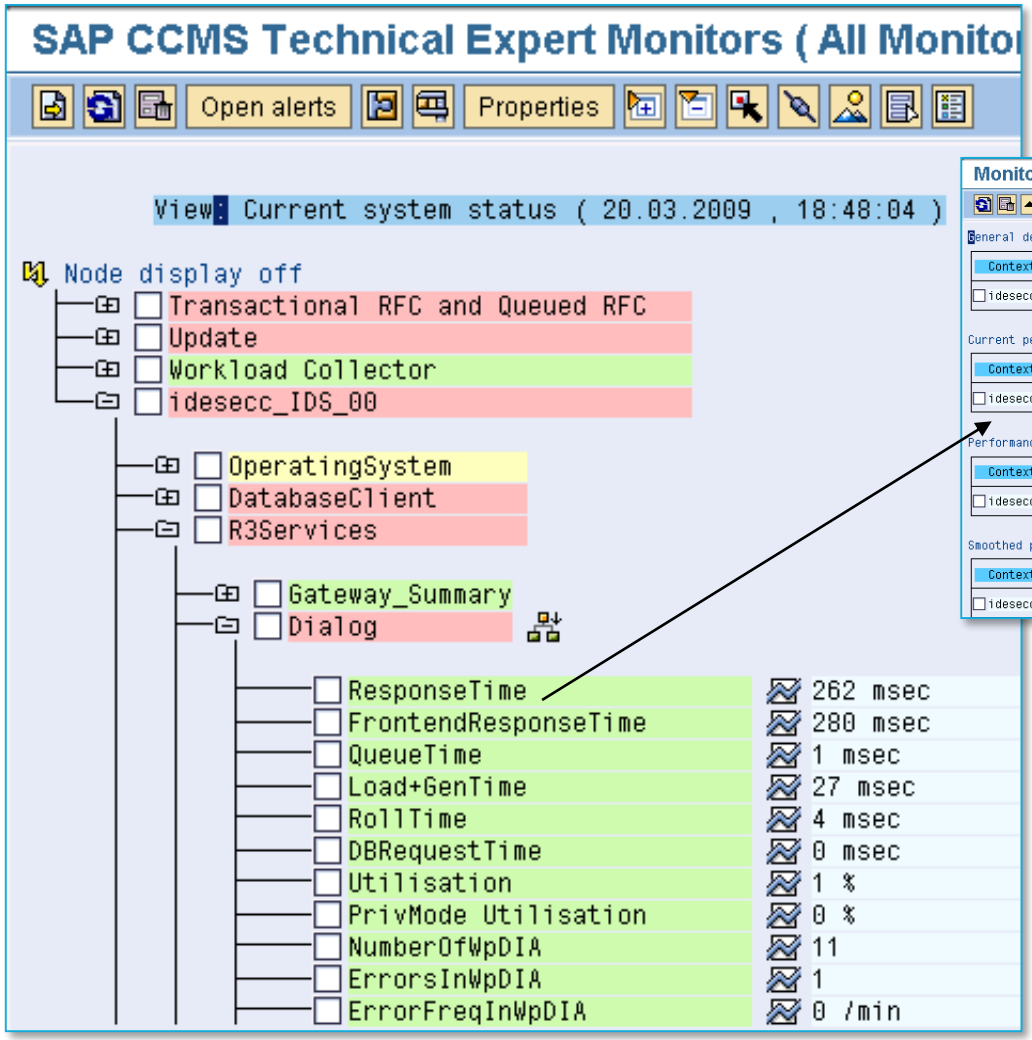
Alert Monitor – RZ20



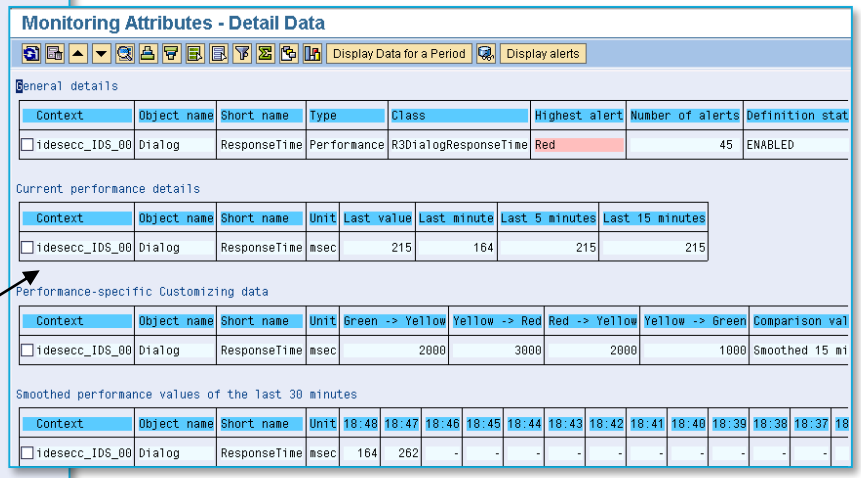
Using the Alert Monitor – RZ20



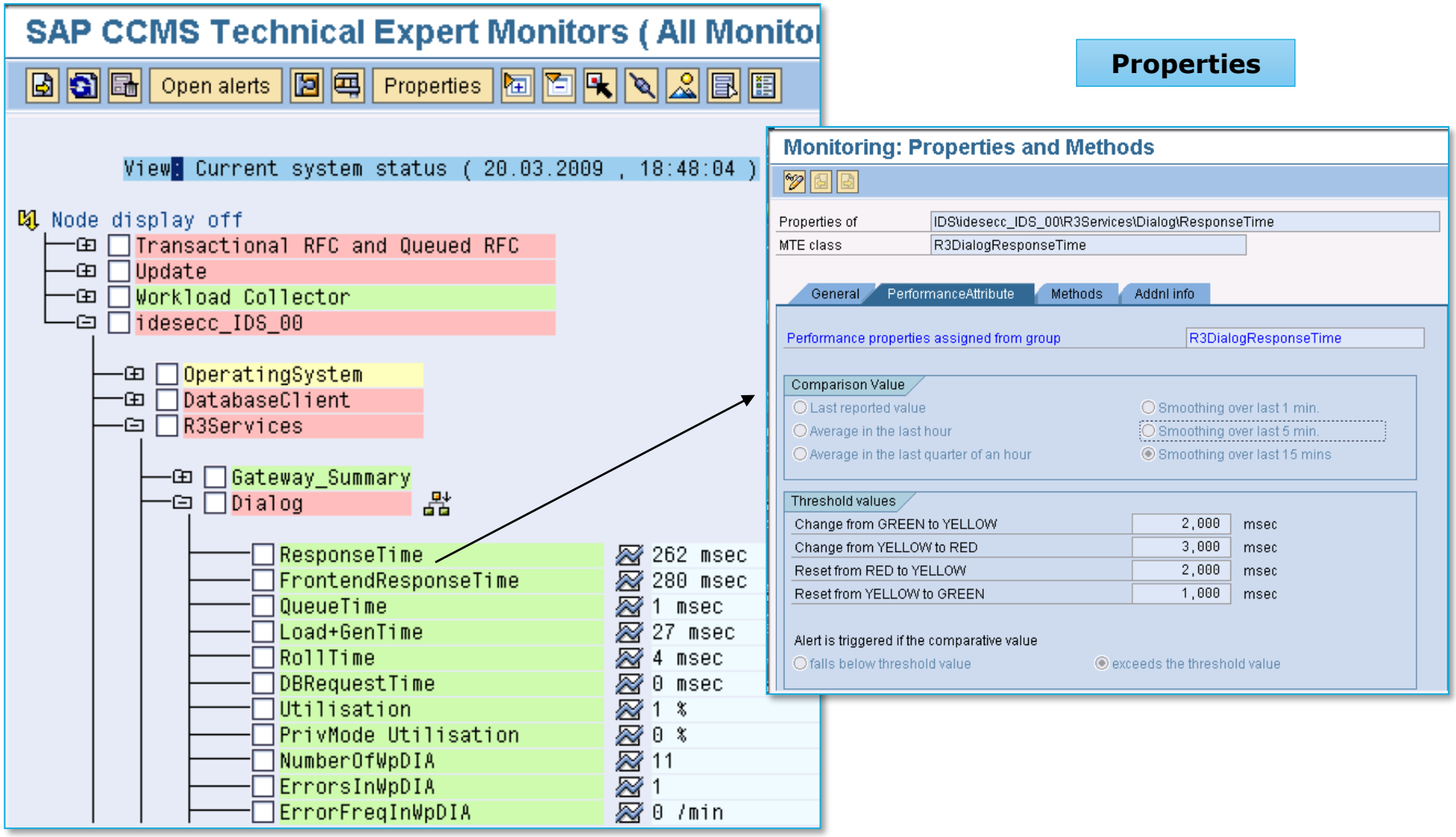
Sample View of RZ20 MTE Attributes



Attributes



Sample View of RZ20 MTE Properties



Break

Web Enablement & Communication

SAP Web Server Concepts

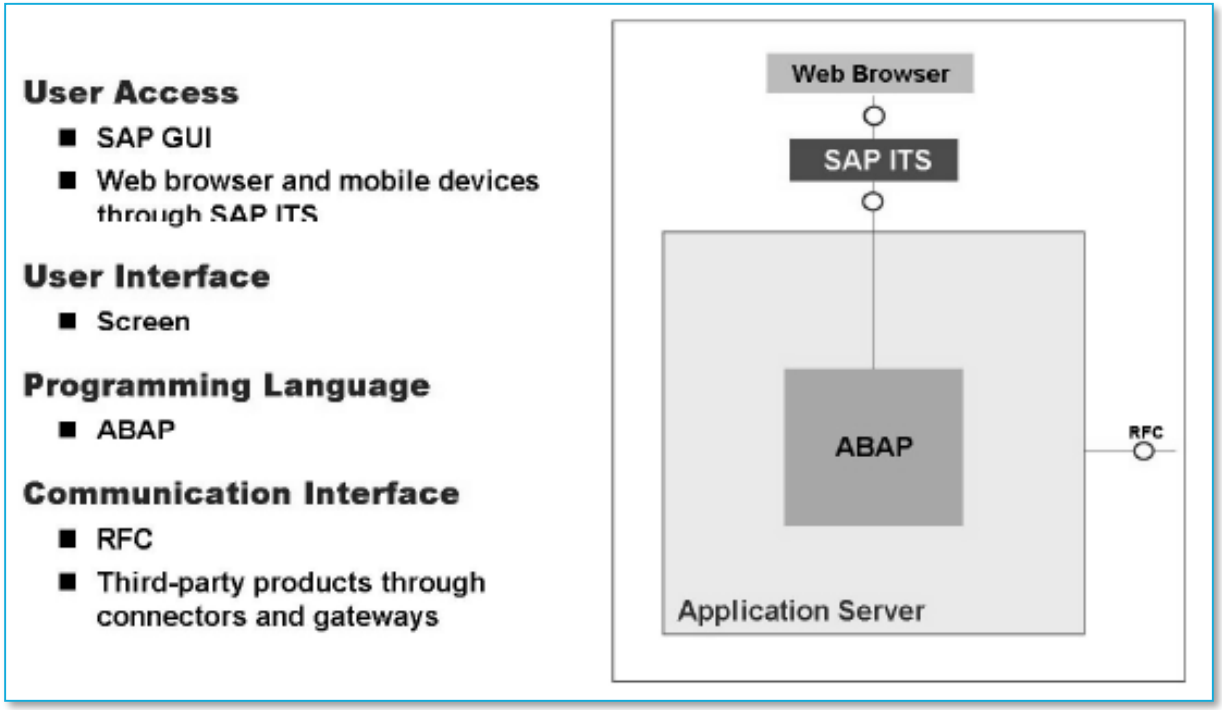
The SAP Internet Transaction Server (ITS) is used with Web applications (IACs) and with SAP GUI for HTML. Depending on the system release and scenario in question, the functions of the SAP ITS can be implemented by means of a standalone ITS or using the ITS integrated in the AS ABAP

The Internet Communication Manager (ICM) is the process that turns the conventional ABAP application server into a Web server or Web Client

The Internet Communication Framework (ICF) provides an environment for handling HTTP(S) requests in the ABAP work process using Web applications such as BSPs

The SAP Web dispatcher distributes HTTP(S) requests to a suitable application server (instance)

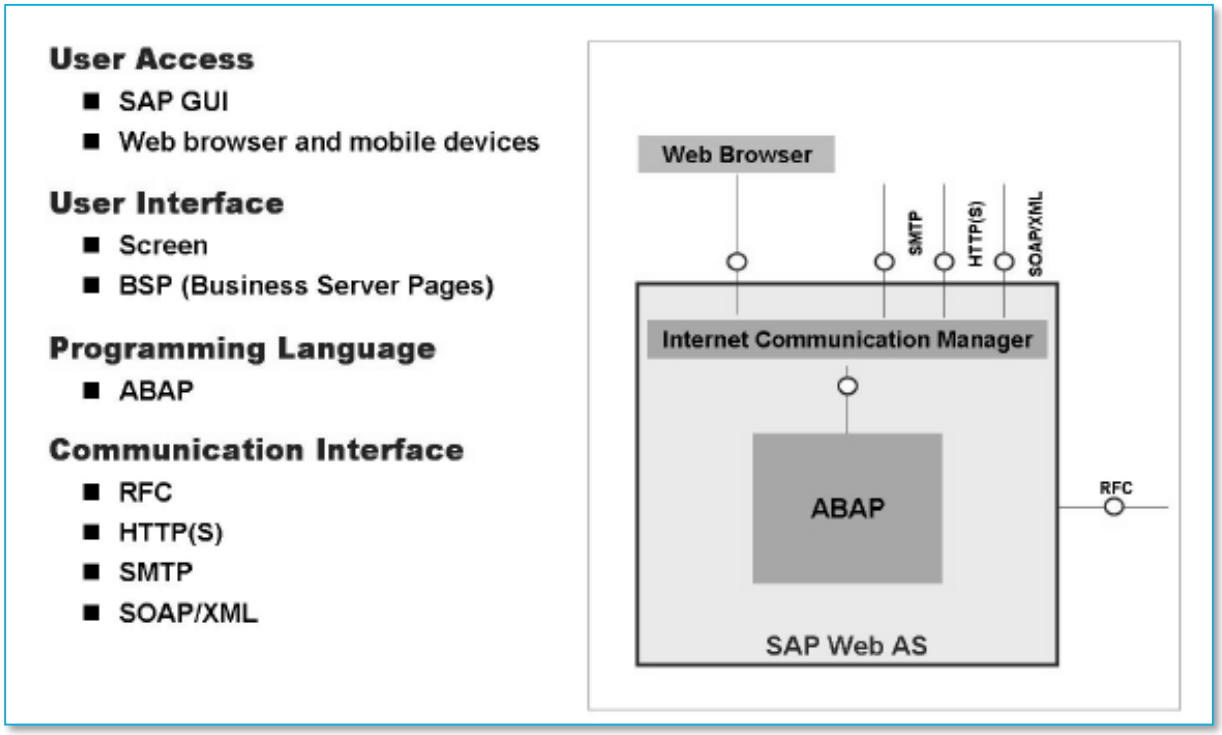
SAP Internet Transaction Server (ITS)



Web applications that were developed specifically for SAP ITS are called Internet Application Components (IACs). These include Employee Self Services (ESS) that are based on SAP R/3 and SAP R/3 Enterprise or the SAP Online Store. The SAP GUI for HTML also uses the SAP ITS.

SAP ITS is therefore required for existing Web applications (in IAC technology) and the SAP GUI for HTML, regardless of the basis release of the corresponding SAP system.

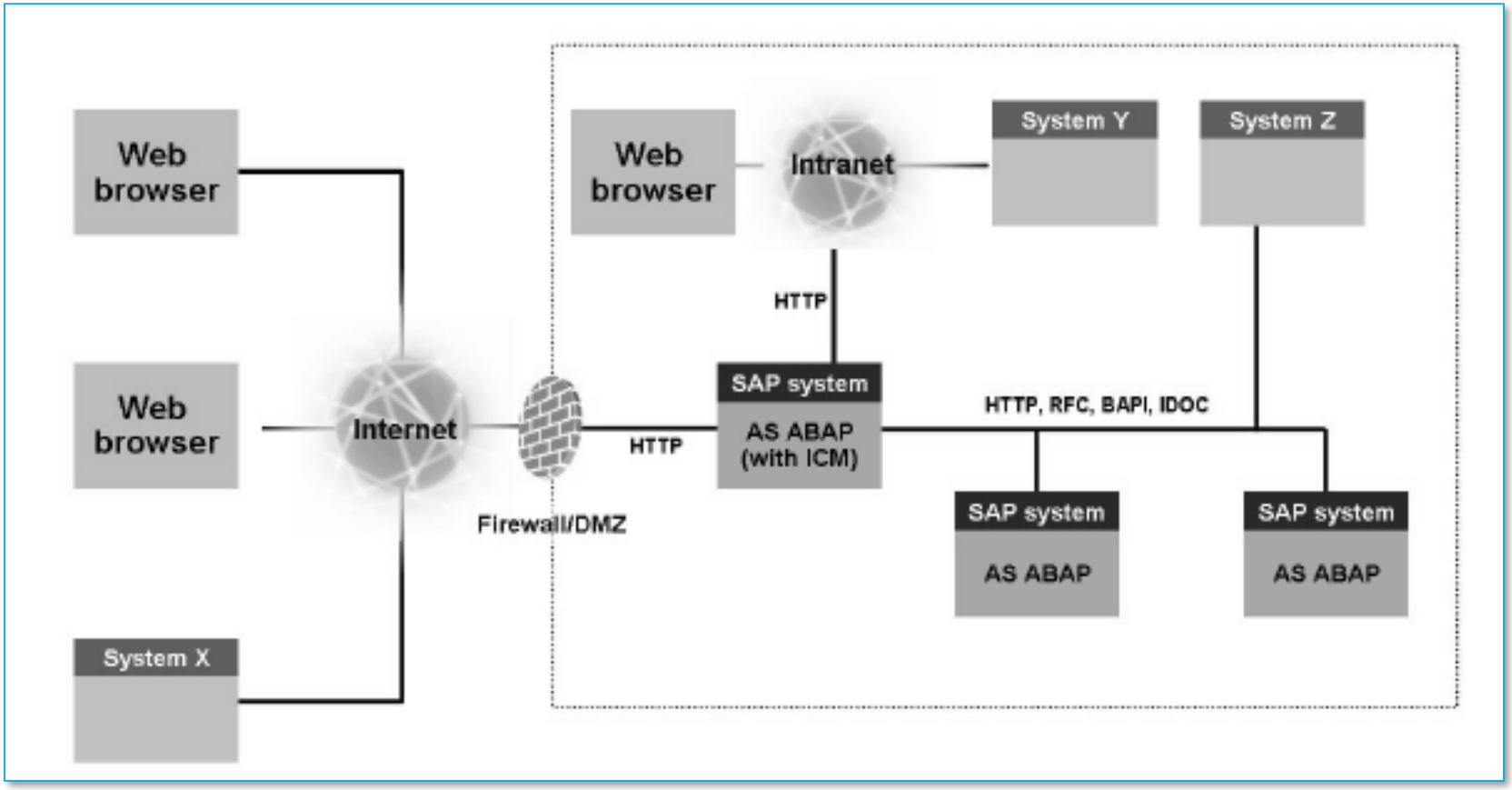
SAP ICM



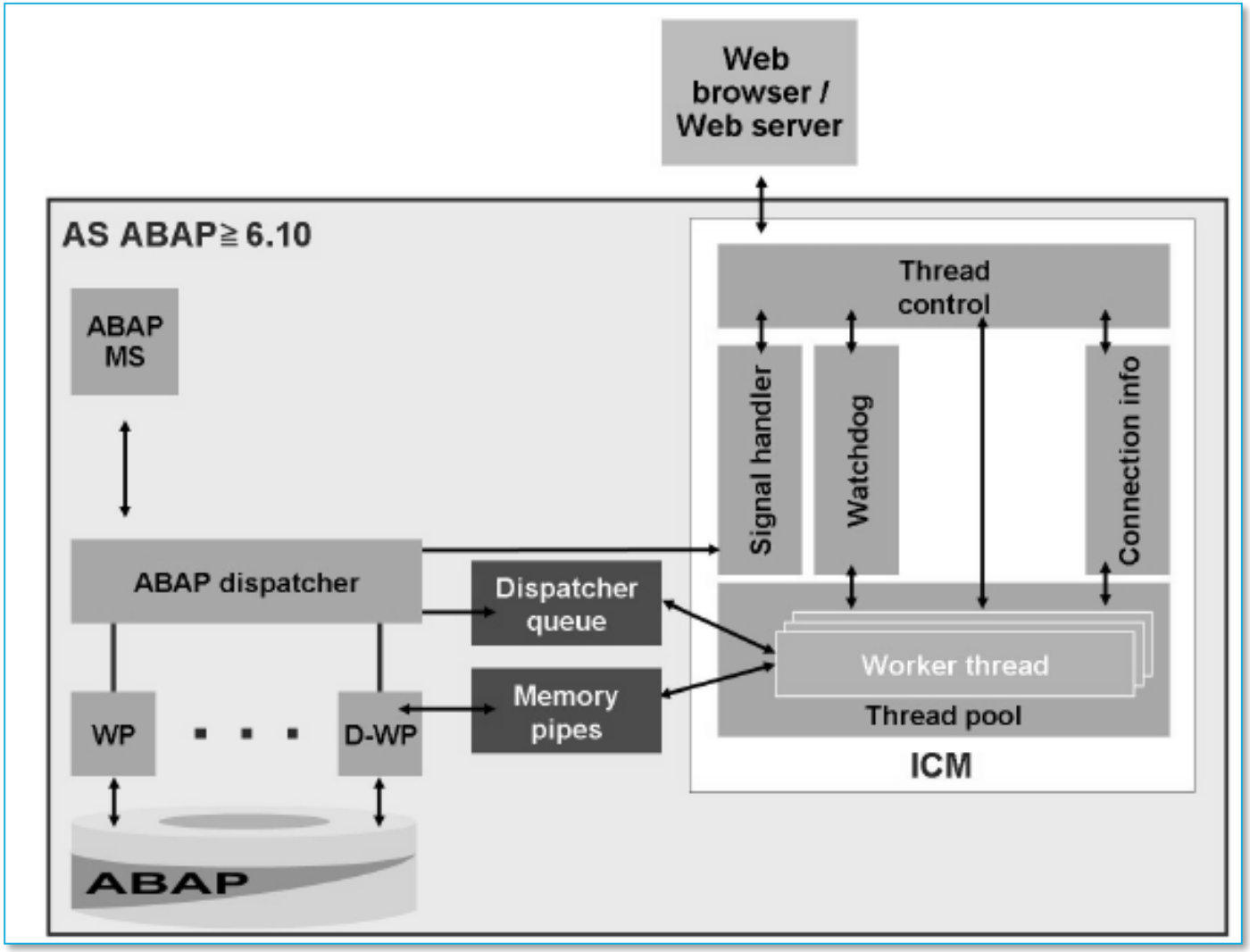
SAP Kernel has been extended with the Internet Communication Manager (ICM) process to process HTTP requests directly from the internet or to send HTTP client requests to the internet

The ICM process forwards requests to the Internet Communication Framework (ICF), which supports numerous programming models. This is how the SAP CRM, SAP BW, and SAP XI software components use this infrastructure. A programming model for such applications are the Business Server Pages (BSPs)

Architecture of the ICM Process



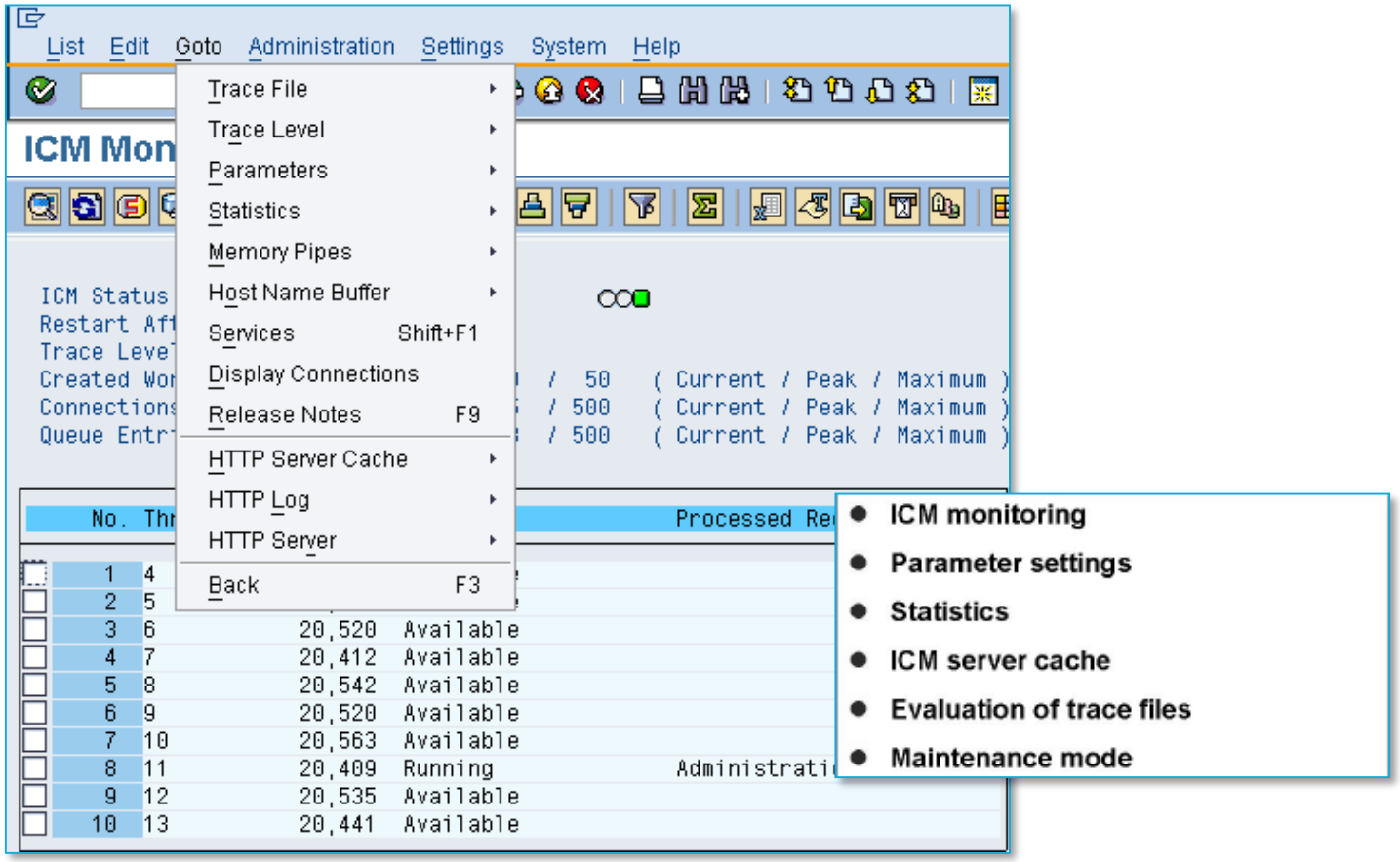
Internal Structure of the ICM Process



ICM Monitoring using SMICM Transaction

Transaction SMICM can be used to start and stop the ICM process

The http trace and logs can be viewed using SMICM



Internet Communication Framework

The Internet Communication Framework (ICF) provides a way for different systems to communicate with each other over the Internet using standard protocols (such as HTTP and SMTP). No additional programming libraries (for AS ABAP) are required from SAP. However, for the HTTPS protocol, the SAP Cryptographic Library (SAPCRYPTOLIB) must be installed and configured (see SAP Note 510007). Your system platform only must be configured to be Internet capable. This scenario allows for the most flexible setup of the overall communication requirements

The ICF allows a response to a request to be generated using an application. An HTTP request is sent from a client (such as a Web browser) to the server. It is then forwarded to an application by the ICF. Here, data is collected and sent back to the client as a response by the ICF. The response data is then displayed in the browser

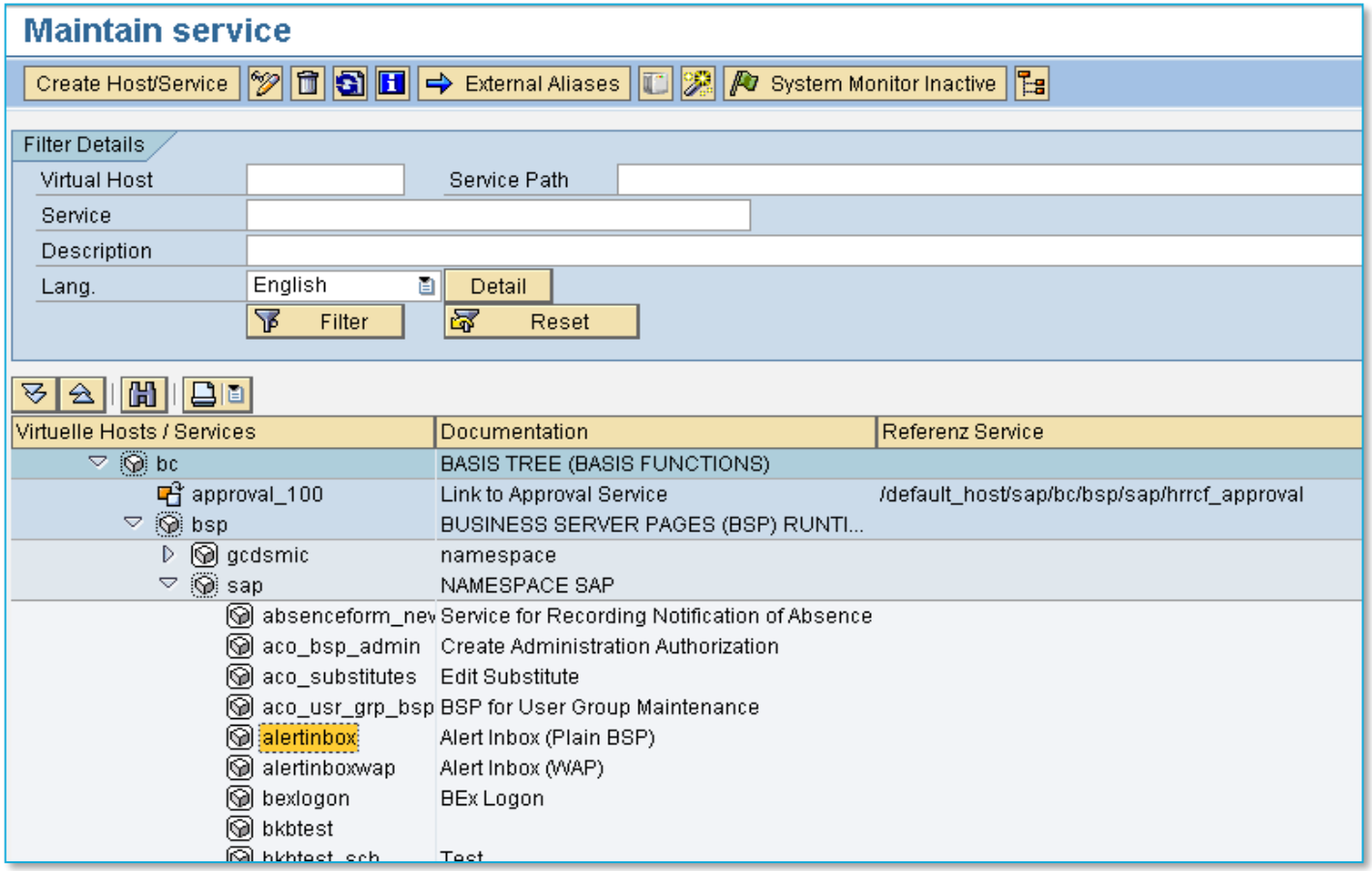
HTTPS Request Processing

An HTTP(S) request is processed in the following steps:

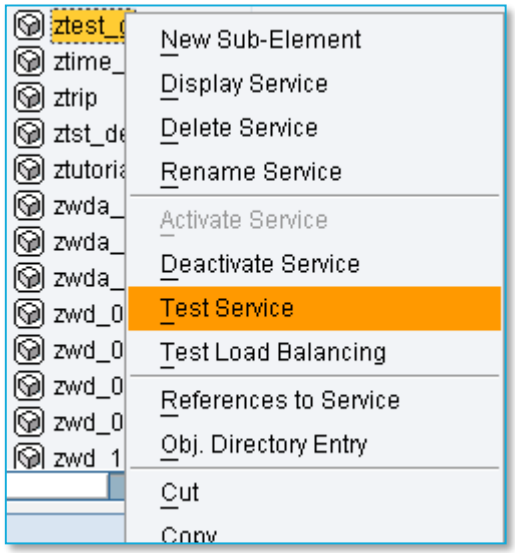
1. The request is sent from the user's Web browser to the ICM using the HTTP
2. protocol. The ICM uses the requested URL to determine whether the application called is implemented in the ABAP or Java stack of the SAP NetWeaver Application server. This example uses an ABAP application that must be processed by a dialog work process
3. The ICM stores the data received in a memory pipe (in the shared memory) and informs the ABAP dispatcher
4. The ABAP dispatcher adds the ICM request to the dispatcher queue, creates a new context (if there is no context that is processed statefully), and selects a work process for processing
5. The task handler in the work process reads the data from the memory pipe and transfers it to the ICF controller, which is implemented using function module HTTP_DISPATCH_REQUEST
6. The ICF controller transfers the request to the ICF manager, which is implemented by the ABAP class CL_HTTP_SERVER. The ICF controller creates a server control block and fills it with the HTTP request data that it requested from the ICM
7. The client is then authenticated, whereby several logon options are available
8. The HTTP request handler determined previously is called (this can process the request data, call further applications, access the response object, and so on). When the HTTP request handler is ready, it returns control to the ICF controller
9. The task handler writes the response back to the memory pipe (response serialization) and signals to the ICM that it has finished processing the request
10. The ICM returns the response to the Web browser

Activating Web Services using SICF

Specific web services that will be executed by the ICF can be activated and managed in SICF

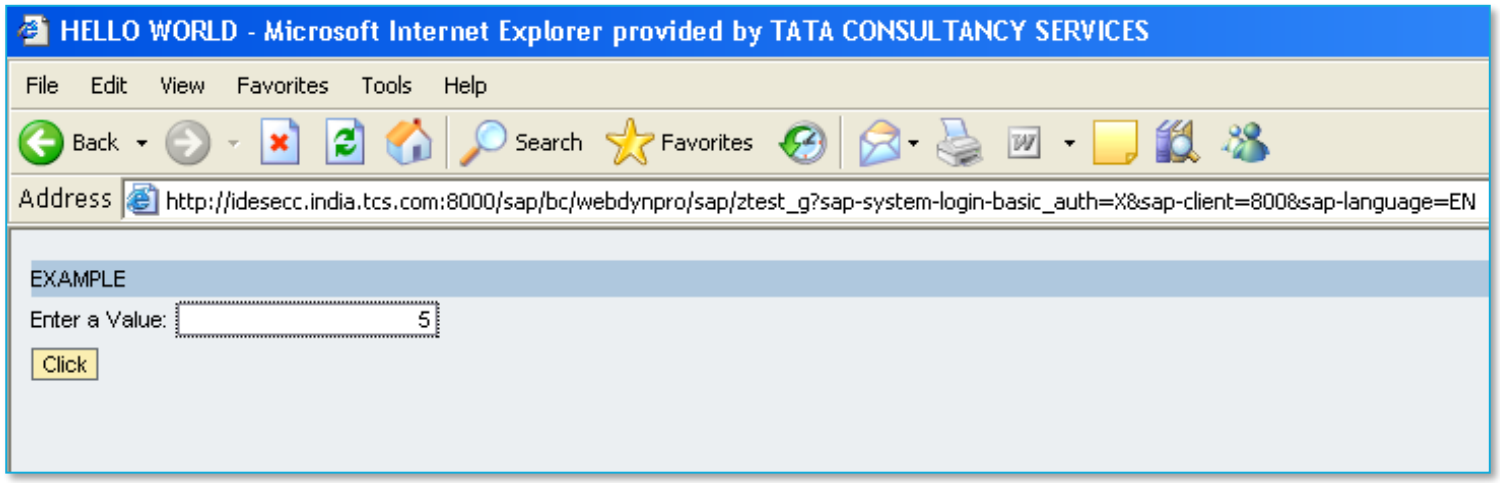


Calling Web Service using SICF



Please NOTE the structure of the URL. The port number for calling the ICM is always 80xx for an AS ABAP WAS, where xx is the instance number

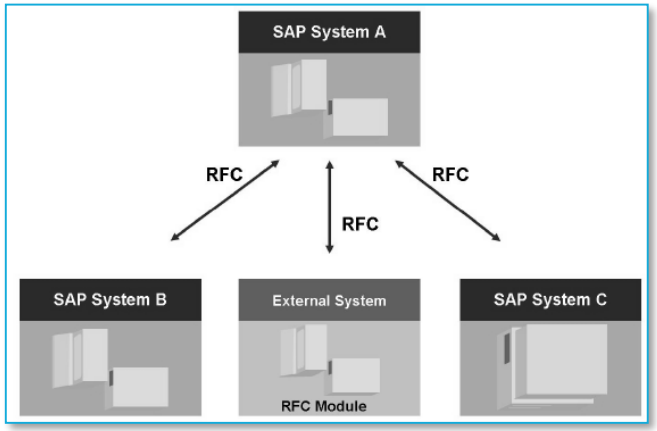
The fully qualified host name must be maintained in Instance profile parameters, i.e., icm/host_name_full



SAP Communication & Integration Technologies

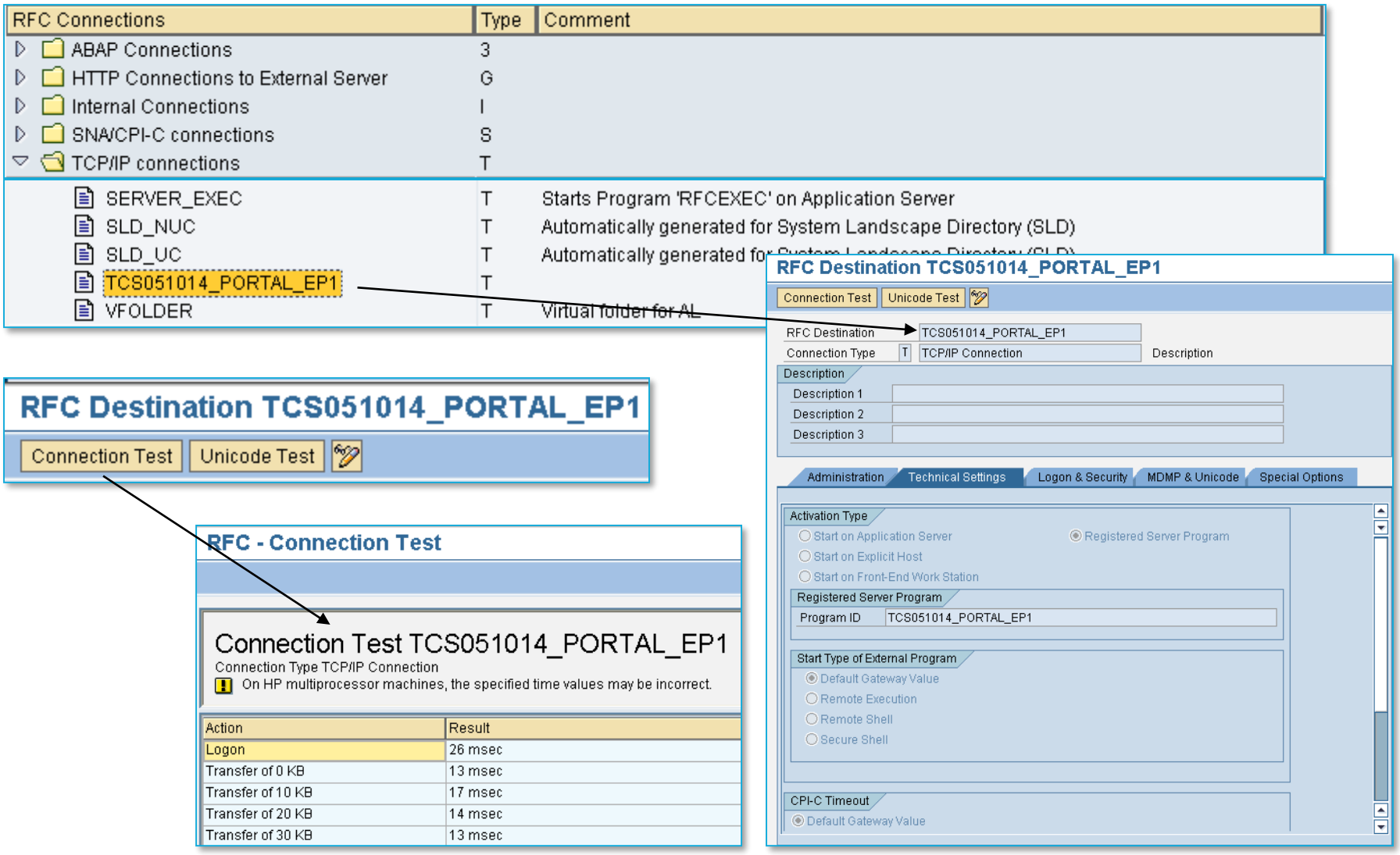
RFC Fundamentals

Remote Function Calls have been used for many years as the technical interface with which SAP and non-SAP systems are usually connected. It is irrelevant whether data exchange is synchronous or asynchronous, periodic or aperiodic, or transactional. Many conceivable variants are supported.



- A Remote Function Call(RFC) is the call of a function module that is running in a different system to the calling program. You can call a function module in the same system as an RFC too. However, RFCs are normally used when the calling and called function modules are running in different systems.
- In the SAP system, the RFC interface system provides this function. The RFC interface system allows function calls between two SAP systems or between an SAP system and an external (non-SAP) system.
- RFC is an SAP interface protocol that is based on the Common Programming Interface for Communication (CPI-C) and allows cross-host communication between programs. This enables external applications to call ABAP functions and SAP systems to contact (RFC-enabled) external applications.RFC means that ABAP programmers do not have to write their own communication routines. For an RFC call, the RFC interface converts all parameter data to the format required in the remote system calls the communication routines that are required to communicate with the remote system handles errors that occur during the communication.
- The RFC interface is easy for the ABAP programmer to use. The processing steps for calling external programs are integrated into the CALL FUNCTION statement.

Managing RFC Connections using SM59



RFC Usage Variants

Synchronous RFC (sRFC)

For communication between different systems and between SAP Web AS and SAP GUI

Asynchronous RFC (aRFC)

For communication between different systems and for parallel processing of selected tasks

Transactional RFC (tRFC)

A special form of asynchronous RFC. Transactional RFC ensures transaction-like processing of processing steps that were originally autonomous

Queue(d) RFC (qRFC)

Queued RFC is an extension of tRFC. It also ensures that individual steps are processed in sequence

RFC is a superordinate term for various implementation variants. sRFC is the synchronous call of function modules. This means that the client waits until the server has completed its processing

Within an SAP system, an RFC can also be executed asynchronously in another work process. This variant is called aRFC

Continued..

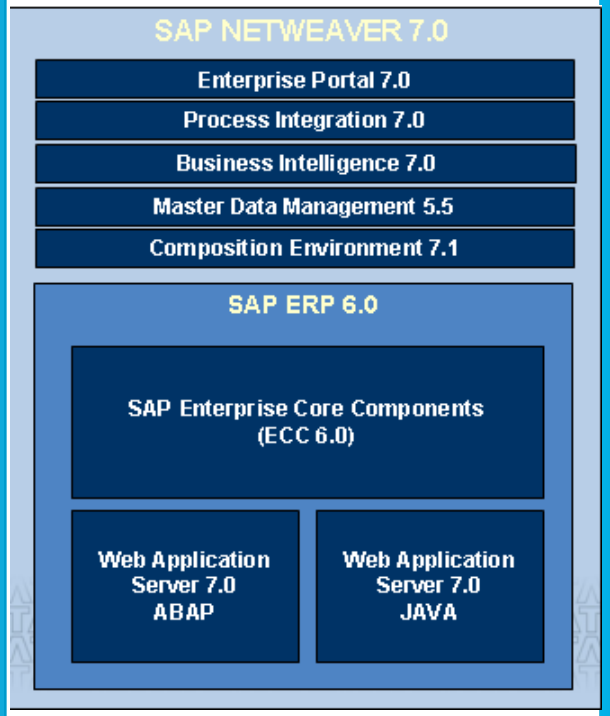
There is also tRFC, the transactional Remote Function Call. Transactional RFC is asynchronous and ensures that data that is sent more than once due to network problems can be recognized at the server side, by assigning a Transaction Identifier (TID). This allows you to prevent data being processed more than once, leading to erroneous information in the application. Due to the asynchronous processing, however, parameters can only be transferred from the client to the server in this case returning information or status information directly is not possible

qRFC with Send Queue is an extension of tRFC. It creates a layer between applications and the tRFC and only allows the tRFC to transfer a Logical Unit of Work (LUW) to the target server when its predecessors are no longer in the associated wait queues. After a qRFC LUW is executed, the qRFC manager automatically processes the next waiting qRFC LUW in accordance with the sequence in the wait queue

Lunch Break

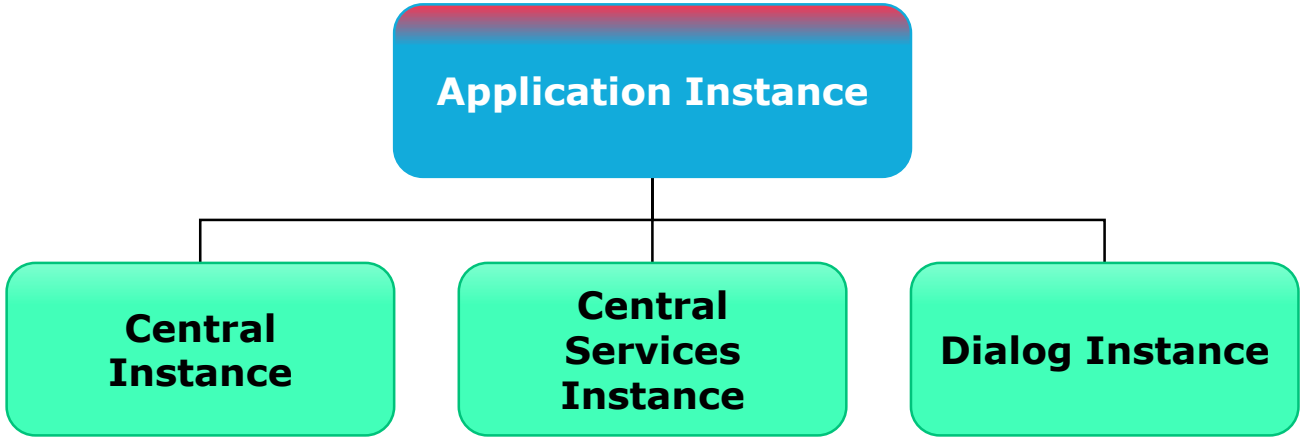
AS JAVA Architecture & Processes

Netweaver Architecture



- The Application server JAVA is a usage type of the SAP Netweaver platform
- The AS JAVA has a dedicated database schema
- AS JAVA can coexist with AS ABAP systems in what is known as Dual stack architecture
- In dual stack systems , there is only 1 database instance , but there are 2 schemas one each for ABAP and JAVA

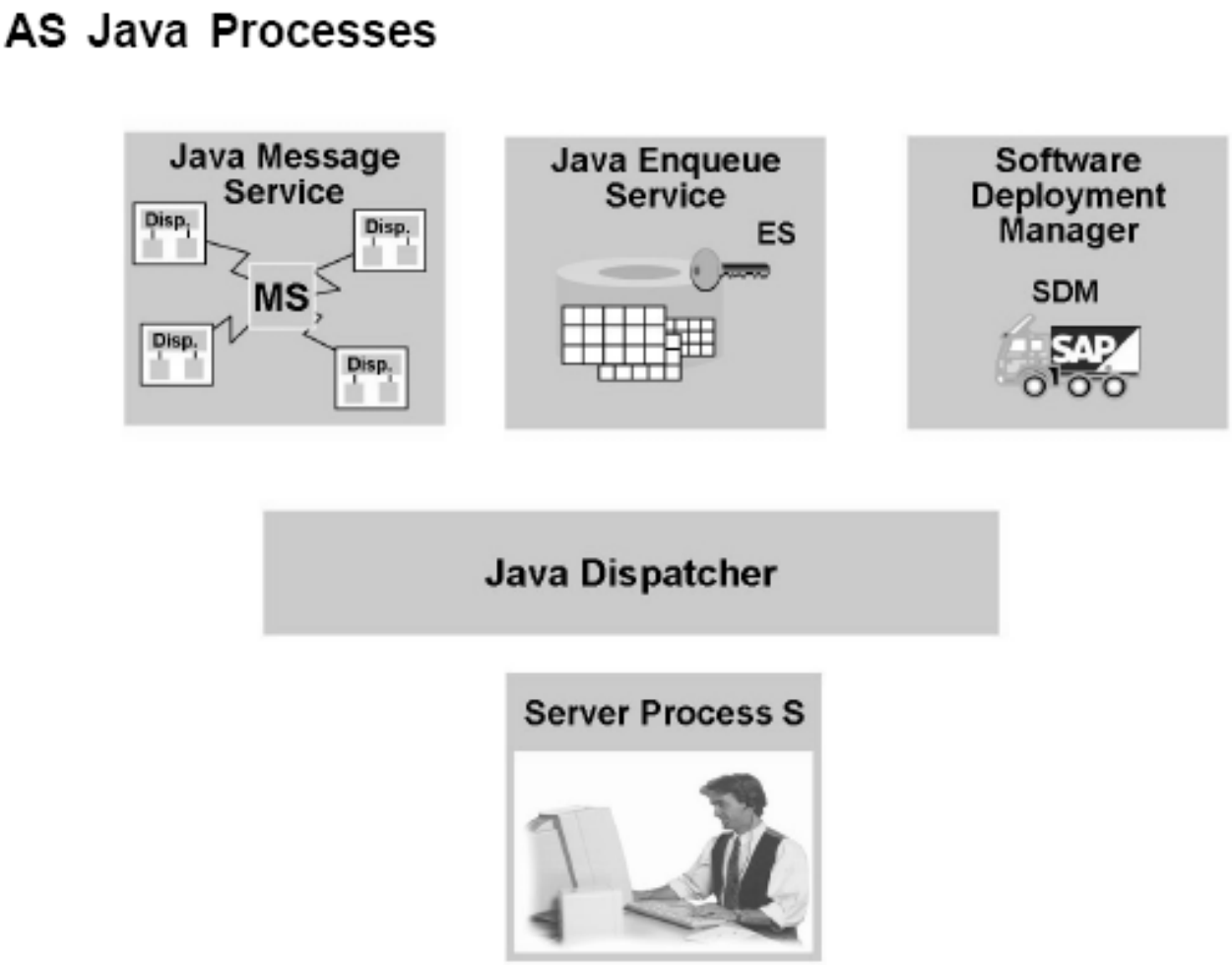
SAP AS JAVA Instance



Type of Application Server	Instance Name	Name of Processes
JAVA Application Server	Central Instance	Software Deployment Manager(SDM) , Dispatcher
	Central Services Instance	Message Server , Enqueue Server
	Dialog Instance	JAVA Server Processes

Each SAP system can have only 1 Central Instance , 1 Central Services Instance and only 1 Database instance. It can have any number of additional dialog instances

AS JAVA Process Overview

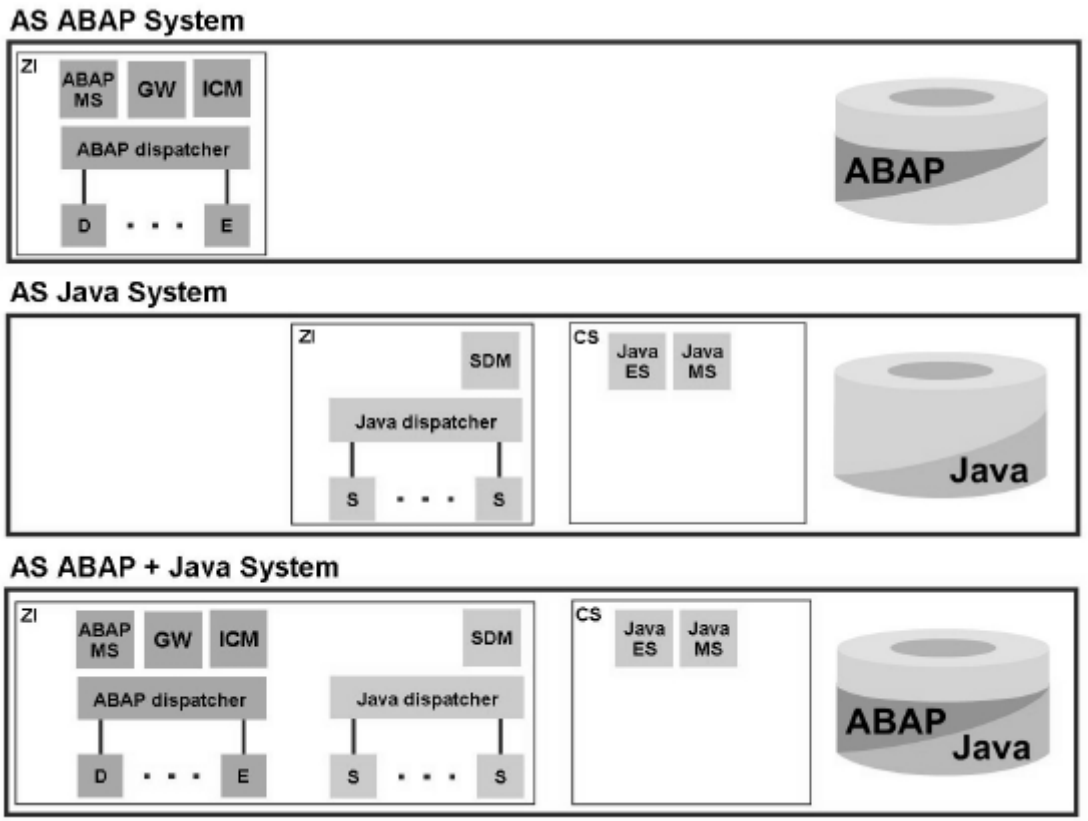


Summary of AS JAVA Processes

The following processes exist in AS Java:

1. The dispatcher distributes incoming requests to the server processes.
2. The server process executes the Java applications. Every server process is multi-threaded and can thus process a large number of requests in parallel (in contrast to the ABAP work processes).
3. For each dispatcher there is at least one server processes and there can be up to 16 server processes.
4. The Java message service manages a list of Java dispatchers and server processes. It is responsible for the communication within the Java runtime environment.
5. The Java enqueue service manages logical locks that are set by the executed Java application program in a server process.
6. The Software Deployment Manager (SDM) is the standard tool used to install Java software components on the SAP Web AS Java.

Types of Netweaver Server Configurations

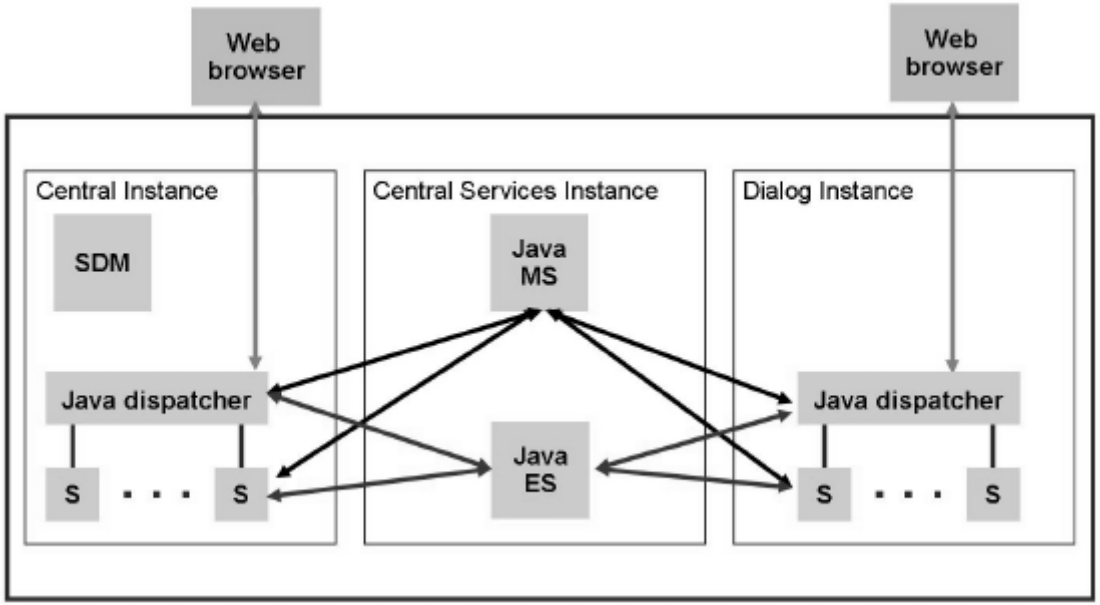


AS ABAP system: Complete infrastructure in which ABAP-based applications can be developed and used.

AS Java system: Complete infrastructure for developing and using J2EE-based applications.

AS ABAP+Java system: Complete infrastructure in which ABAP-based and J2EE-based applications can be developed and used. Such a system should only be installed if explicitly required by the application. For example, SAP NetWeaver PI 7.0 or SAP Solution Manager 4.0

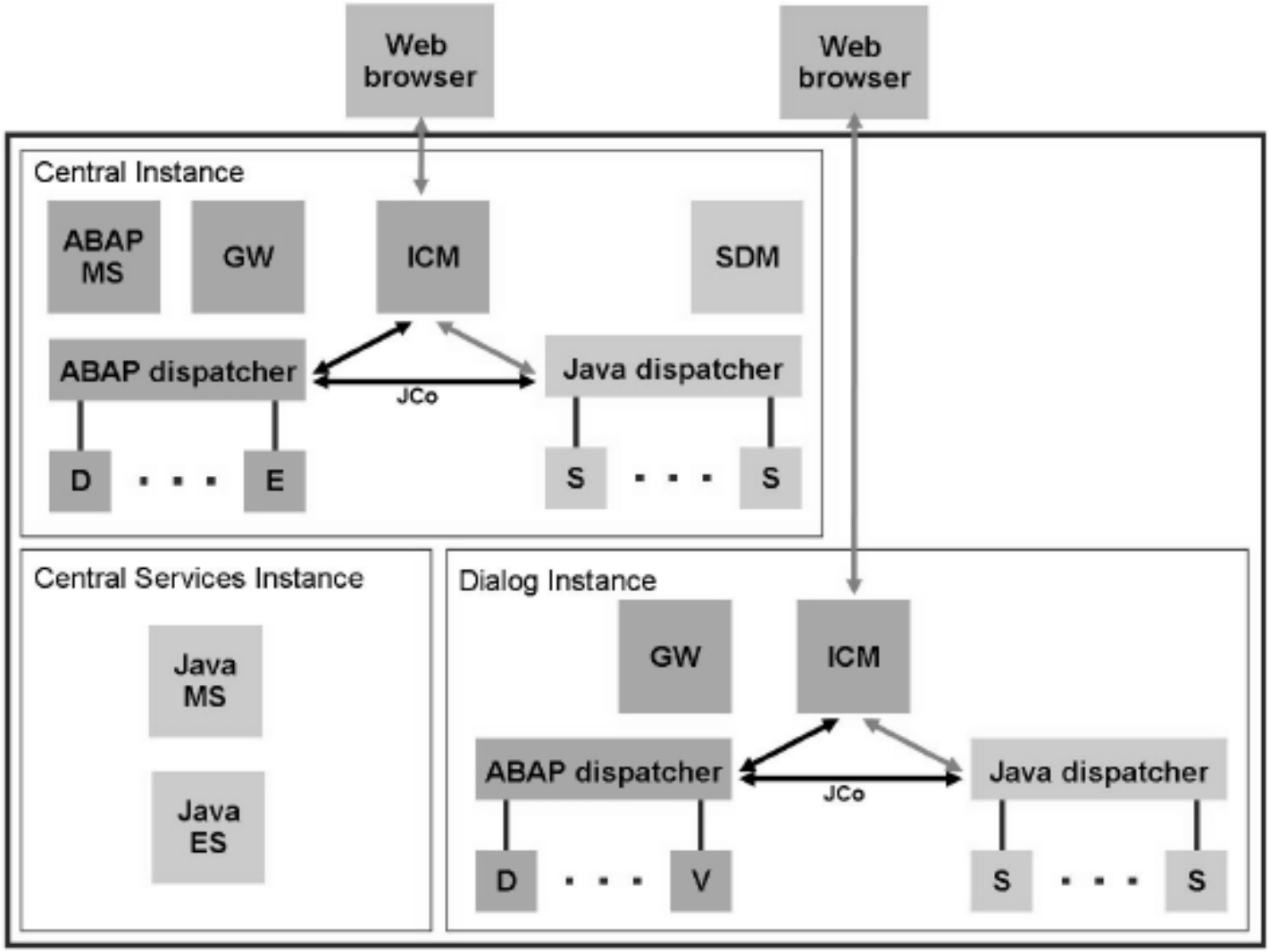
AS JAVA Architecture



In AS Java, the central instance is distinguished by the fact that the Software Deployment Manager (SDM) runs there. The central services Message Service (MS) and Enqueue Service (ES) run in the central services instance (CS instance). All other instances of the system are usually called dialog instances.

Note: The entirety of the Java environment (all processes and the database scheme) is also referred to a Java cluster, and the individual processes (dispatcher and server) as nodes of the Java cluster.

Dual Stack Architecture

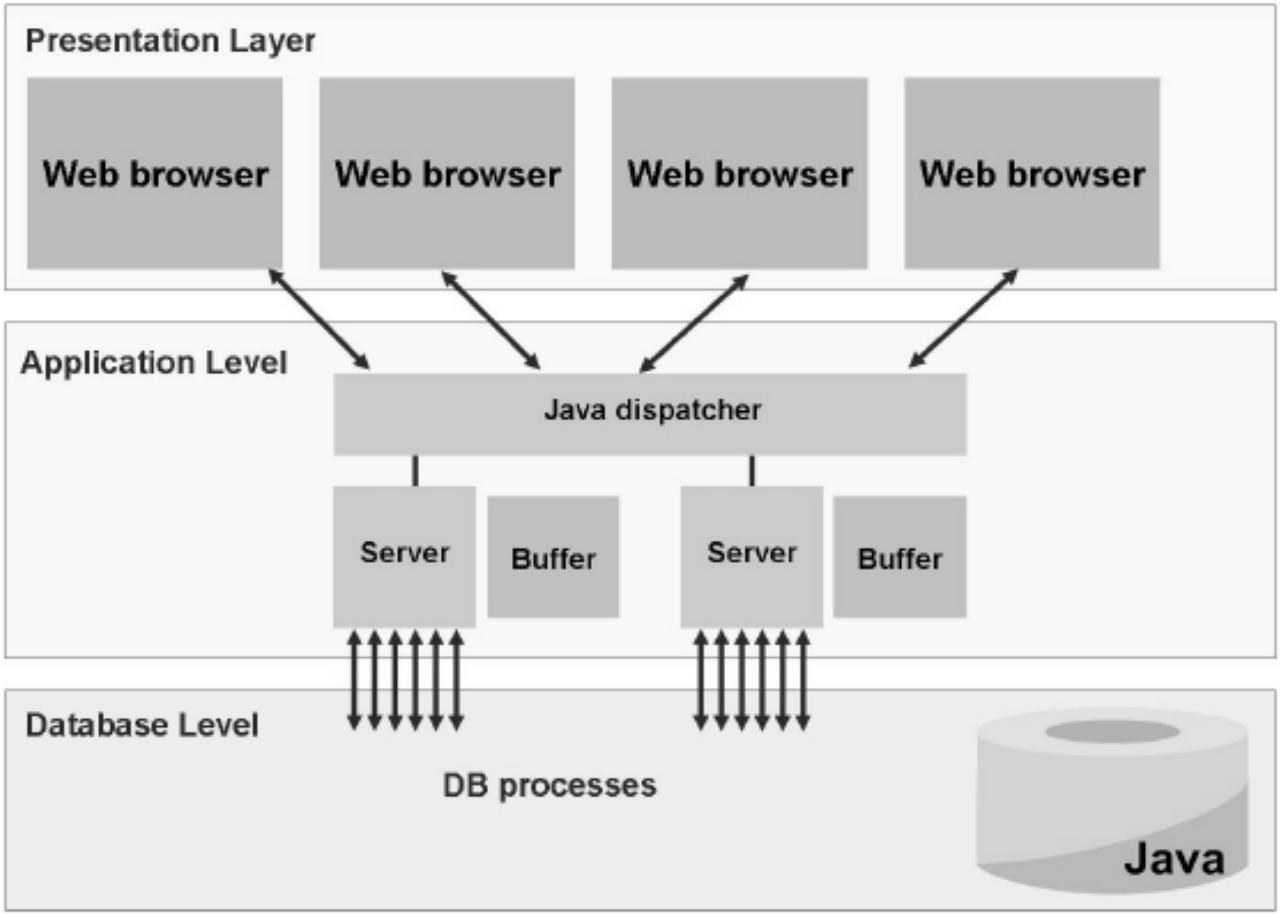


Key architecture principles of Dual stack systems

The central instance of an AS ABAP+Java system can be recognized by the following processes:

- ABAP-MS, enqueue work process and SDM. The central services of the Java runtime environment (Java-MS, Java-ES) are also provided in the Java central services instance here. All other instances are usually called dialog instances.
- Since both runtime environments are capable of answering requests via web protocols, the Internet Communication Manager must now decide whether the request is addressed to the ABAP or the Java runtime environment. It decides this by means of the URL of the request. In case of a request to the ABAP runtime environment, for example, the call of an ABAP Web Dynpro, the ICM forwards the request to the ABAP dispatcher and the work processes respond to the request. If the request is a request for the Java runtime environment, for example, the call of a Java Server Page (JSP), the ICM forwards the request to the Java dispatcher and one of the server processes responds to the request.
- In an AS ABAP+ Java system, data is also kept in separate database schemas (but in the same database installation). That is, work processes can only access ABAP data and server processes can only access Java data. In the data exchange, both runtime environments then communicate using the SAP Java Connector (JCo). This communication is necessary, for example, if billing data that is stored in the ABAP data schema is supposed to be displayed in a Java user interface.
- The SAP JCo is integrated into the AS Java and is also used when an AS Java system has to communicate with a remote AS ABAP system.

Processing User Requests in Java



A web browser is the standard user interface for AS Java. A user request for AS Java is usually an HTTP request that is received by the Java dispatcher. The dispatcher forwards the processing requests to one of the server processes of “its” instance.

Details on request processing

1. The actual processing takes place in the server process, whereby the user who sent the request is usually assigned the same server process again for the next request.
2. The dispatcher and server processes of AS Java are also called nodes. All processes of AS Java together with the database schema form the Java cluster. In contrast to the processes of AS ABAP (excluding the ICM), the cluster nodes of AS Java are multithreaded. This means that an AS Java process consists of many threads and one request can be processed in each thread. Hence, one server process always processes many user requests in parallel.
3. To process user requests it is often necessary to read data from the Java schema of the database or to write to it. To do so, each server process is connected multiple times to the Java schema of the database via a connection pool (DB pool).
4. Once processing is complete, the processing result from the server processes is returned to the web browser via the dispatcher.

Break

Exercise

Exercise

- Login to the server at the OS Level
- Identify the structure of the AS JAVA Instance , note the location of the profile files , log files and
  the kernel
- Start the browser and goto the URL : http://servername:5<instance Number>00/
- Navigate to system info link to get a clear understanding of the instance properties and the
  software components
- Now start the quicklink /nwa to start the Netweaver Administrator
- Using the logs & trace option under monitoring , open the latest files and note the nature of the
  logs created
- Start the Visual admin and walk through the dispatcher and server nodes. Note the manager & service names and their properties
- Start the config tool and navigate through the nodes & properties