Capgemini

SAP BASIS Introductory Training Program

Day 6: Agenda

- User Management in JAVA
- Break
- Internet Communication: ICM, ITS &ICF
- Lunch Break
- SAP Web Dispatcher
- Break
- SAP NetWeaver AS JAVA Monitoring Tools
- Exercise & Break Out Session

User Management in Java

User Management in Java

Users – Groups – Roles

- In the UME environment, the term Principle designates the following, central objects:

Principles in UME environment

Principle	Meaning
User	General properties of a user (such as name, e-mail, telephone number etc.)
User account	Logon-related properties of a user (such as password, validity, lock indicator etc.)
Group	Set of user and/or groups
Role	Set of (Java) authorizations

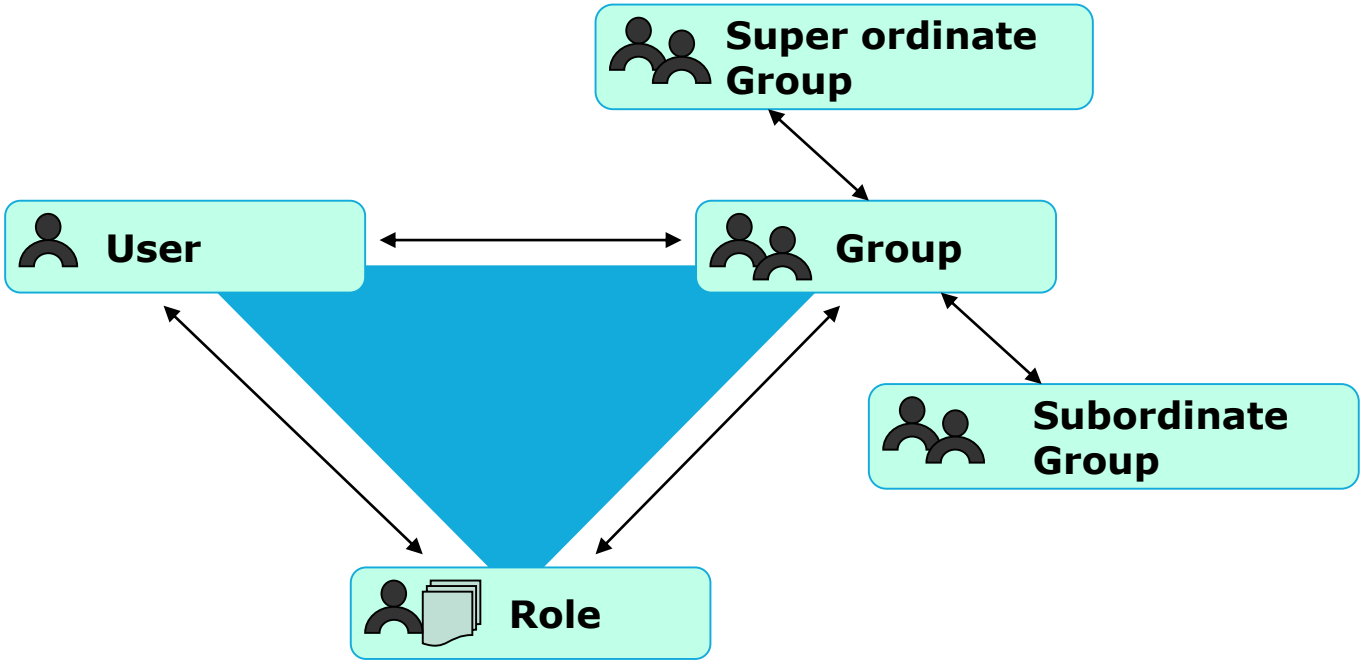
Assigning Principles

Users are usually assigned to groups to which roles are then assigned. However, it is also possible to assign roles to users directly. The Principle group supports hierarchies of groups. A group may also possess super ordinate and subordinate groups. Users actually possess the roles which

are directly assigned to them

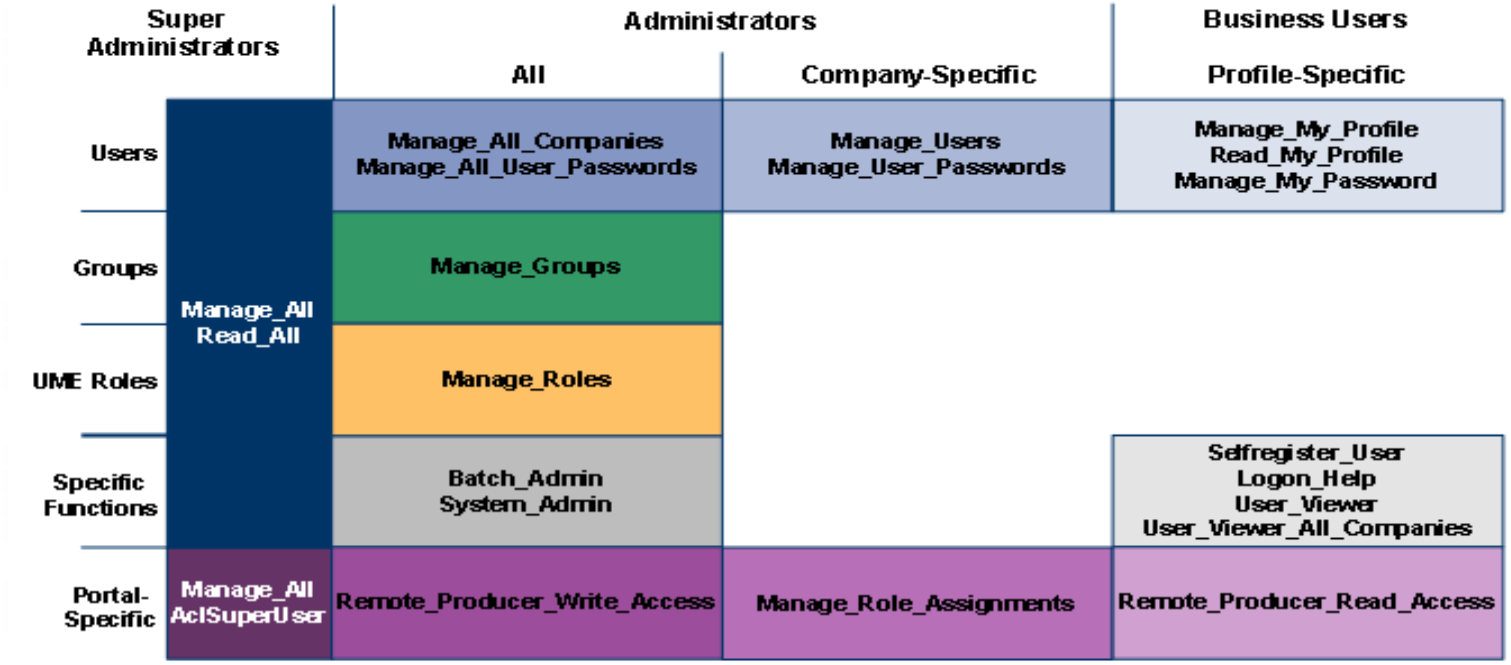
are assigned to the groups to which they belong

are assigned to the super ordinate group of the groups to which they belong



Managing Users, Groups and Roles

To manage users, groups, or roles, you must be assigned a role that includes the relevant actions or combination of actions. For example, to assign roles to users, your role assignments must include UME actions that enable you to change both principals, roles and users, such as UME.Manage_Roles and UME.Manage_Users. The figure below summarizes the UME actions available by default in the SAP NetWeaver Application Server (AS).



UME Actions According to Principal and Role

Special Features of the ABAP System Data Source

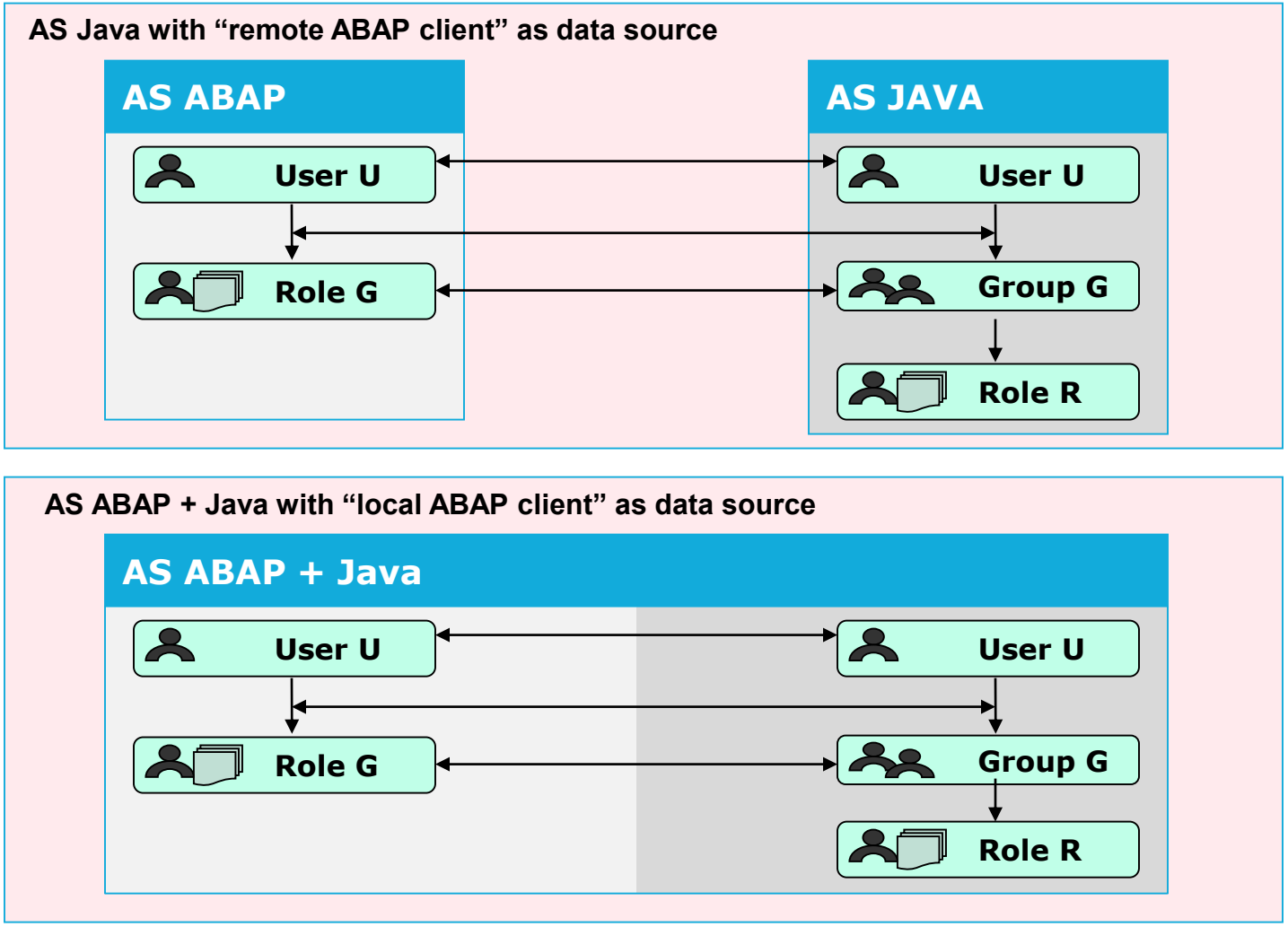
If you use a client of an ABAP system (and consequently the configuration file dataSourceConfiguration_abap.xml) as the data source then UME behaves as follows:

- The ABAP users are visible in AS Java and can log onto AS Java with their ABAP passwords.
- The ABAP roles are depicted in AS Java as UME groups of the same name.
- In AS Java, the assignment of ABAP users to ABAP (composite) roles appears as the assignment of UME users to UME groups.

The reason for this group administration concept is the shared authorization administration for applications that have both ABAP and Java components. Applications such as PI, for example, are made of both ABAP and Java components. The ABAP authorizations are mapped with PFCG roles. The J2EE authorizations are realized using UME roles. A user should be assigned a PFCG role in the ABAP system and a UME role on the Java side for the user to have both ABAP and Java authorizations.

The connection between the UME in an AS Java and user management in an AS ABAP is established via the Java Connector (JCo). A communication user existing in ABAP is stored as a UME parameter (this usually has SAPJSF in its name). This communication user's ABAP authorization determines whether it is possible to modify ABAP user master records using UME resources.

Special Features of the ABAP System Data Source



Administration Tools

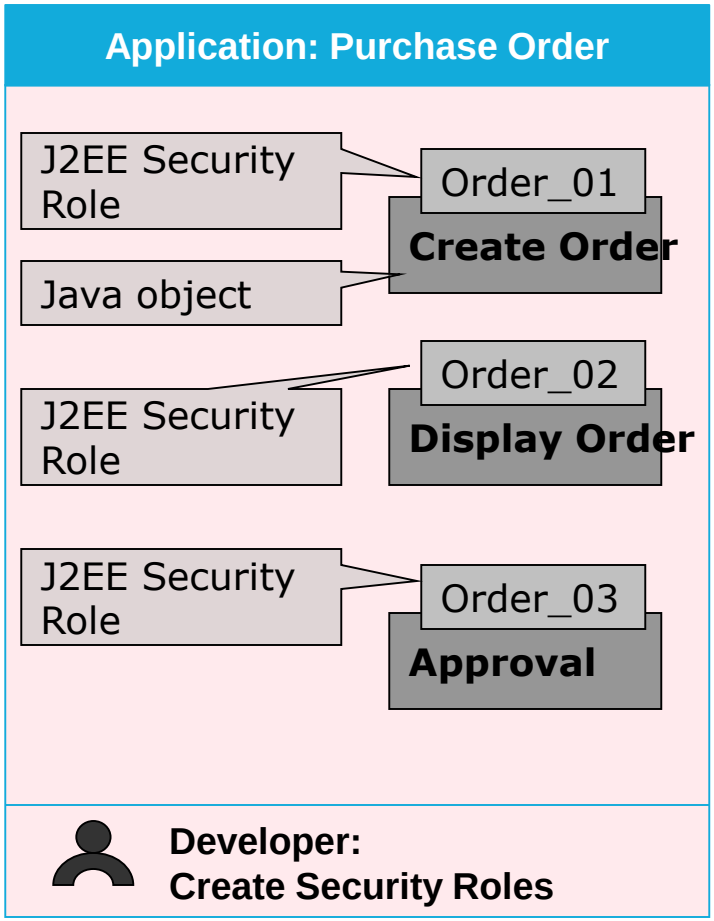
The most important tool for a user administrator in an AS Java system is the UME Administration Console. This functions independently of the configured data source and is implemented as an application running in a Web browser (based on Web Dynpro Java). To get started the Administration Console...

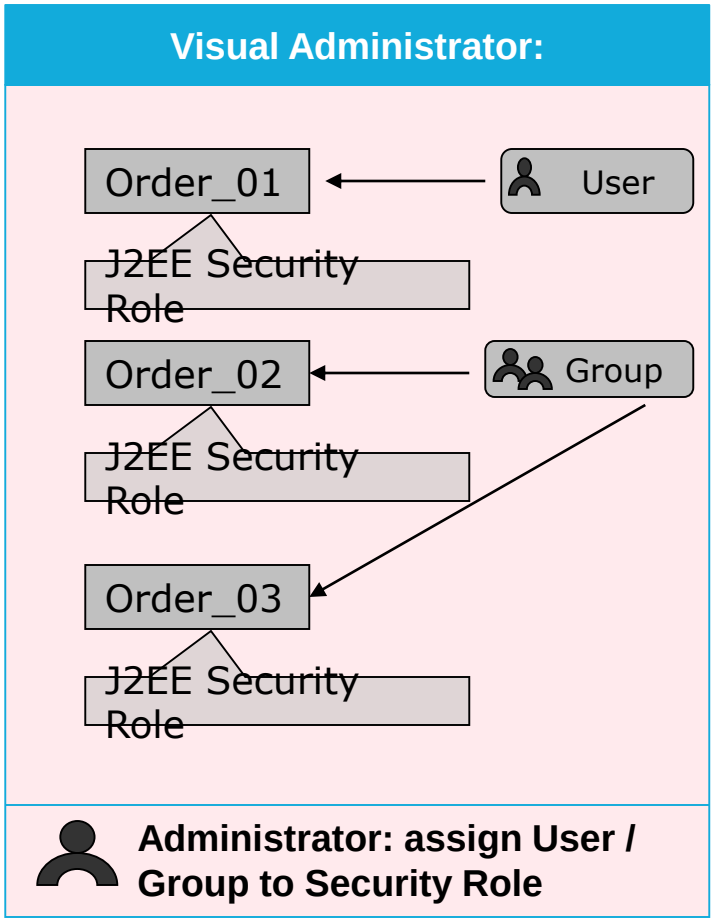
- via the URL http(s)://<hostname>.<domain>:<http(s) port>/useradmin
- via the SAP NetWeaver Administrator (URL .../nwa) via the path System Management → Administration → Identity Management
- via portal URL http(s)://<Hostname>.<domain>:http(s) port/irj via the path System Administration → System Configuration → UME Configuration

UME User Types

User Type	Logon to AS Java	Password Rules	Mapped ABAP user Types (with ABAP System as data source)
Standard	Possible	Applies	Dialog
Technical Users	Possible	Does not apply	System
Internal Service user	Not possible	Applies	___
Unknown	Possible	Applies	Communication, Service and Reference

Structure of J2EE Security Roles

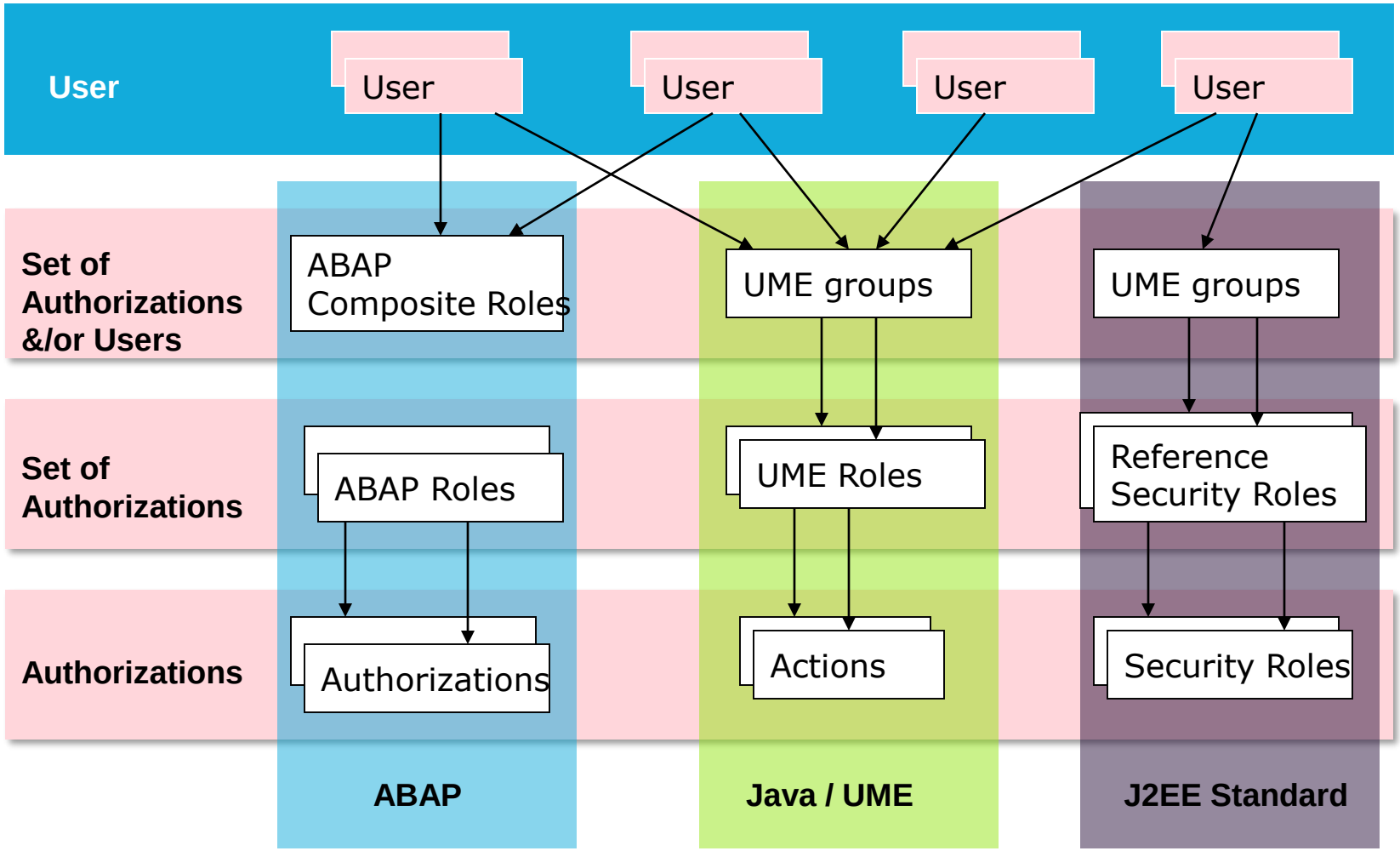




Structure of J2EE Security Roles

The previous figure shows the Order application as an example. For this application, a developer creates objects such as Create order, Approve order, and so on. If you are using J2EE security roles, a security role must be created for each object. The role is defined in the deployment descriptor (XML file) of a specific application. If the application is made available on the J2EE server, the administrator must add user names or user groups to each of these security roles for the users that are to use this application. The administrator must assign each single authorization/J2EE security role individually to a user or a group.

Comparison of Authorization Concepts



UME Datasources

The data repositories or persistence layers from which the user management engine (UME) retrieves user management data are referred to as data sources. With the UME, you can leverage existing user data repositories in your system infrastructure by connecting to them using configurable persistence adapters. You can read data from and write data to multiple data sources in parallel.

A persistence manager is responsible for reading the data from or writing the data to the correct data source. The data source to which the persistence manager writes is transparent to applications using UME.

The configuration of the UME for the different data source types is defined by the data source configuration file. The data source configuration file is an XML file that defines a configuration for standard scenarios, such as storing standard user data in a corporate LDAP directory (directory service) and application-specific data in the AS Java database. For more information about the data source configuration files available, see the sections about the individual data source types.

The UME can use the following types of data sources:

- Database of the AS JAVA
- Directory Service
- User Management of the AS ABAP

UME Datasources

Database Only as Data Source

All user, user account, role, and group data is stored in the database of the SAP NetWeaver Application Server (AS) Java.

Configuration file: dataSourceConfiguration_database_only.xml

LDAP Directory as Data Source

The user management engine (UME) can use an LDAP directory as its data source for user management data. You can connect the LDAP directory as a read-only data source or as a writeable data source.

1. User management data is stored in a combination of an LDAP server and a database

You have a mixed system landscape including both SAP and non-SAP systems, or you have an existing corporate LDAP directory in your system landscape. You want to store standard user data such as name, address, email, and so on in the directory while you want to store application-specific data in the database .

Configuration file: If the LDAP directory has a flat hierarchy:

dataSourceConfiguration_<LDAP_directory_vendor>_not_readonly_db.xml

If the LDAP directory has a deep hierarchy:

dataSourceConfiguration_<LDAP_directory_vendor>_deep_not_readonly_db.xml

UME Datasources

2. User management data is stored in a combination of a read-only LDAP server and a database

You have an existing corporate LDAP directory in your system landscape and have existing processes for administering user data on this directory. You are using the UME with SAP NetWeaver Portal and want all users that register themselves in the portal to be stored separately from the user data on the corporate directory.

Configuration file: If the LDAP directory has a flat hierarchy:

dataSourceConfiguration_<LDAP_directory_vendor>_readonly_db.xml

If the LDAP directory has a deep hierarchy:

dataSourceConfiguration_<LDAP_directory_vendor>_deep_readonly_db.xml

The user management engine (UME) can use an SAP NetWeaver Application Server (AS) ABAP as its data source for user management data. This enables you to take advantage of the following:

Users of the ABAP system are visible as users in the UME and can log on with their passwords from the ABAP system.

Roles of the ABAP system appear as groups in the UME.

The AS Java depicts the hierarchy between collective roles and single roles as nested group structures. When you create new groups created with the AS Java, the AS Java stores them in its database

UME Datasources

User Management of Application Server ABAP as Data Source

The user management engine (UME) can use an SAP NetWeaver Application Server (AS) ABAP as its data source for user management data. This enables you to take advantage of the following:

Users of the ABAP system are visible as users in the UME and can log on with their passwords from the ABAP system.

Roles of the ABAP system appear as groups in the UME.

The AS Java depicts the hierarchy between collective roles and single roles as nested group structures. When you create new groups created with the AS Java, the AS Java stores them in its database.

User and role assignments in the ABAP system appear as user and group assignments in the UME. You can use the ABAP roles for authorization management in the UME, by adding the groups representing the ABAP roles to the UME roles.

Configuration file is dataSourceConfiguration_abap.xml

BREAK

Internet Communication:ITS,ICM and ICF

Internet Communications – SAP ITS

SAP delivered the first version of the SAP Internet Transaction Server (SAP ITS) with SAP R/3 3.1G in 1996. It is a software component that acts as a gateway between a Web server and an SAP system. SAP ITS switches between Internet protocols and formats (such as HTTP, HTTPS, and HTML) and those of the SAP system (such as DIAG, RFC, and screens).

User Access

- SAP GUI
- Web browser or mobile devices through SAP ITS

User Interface

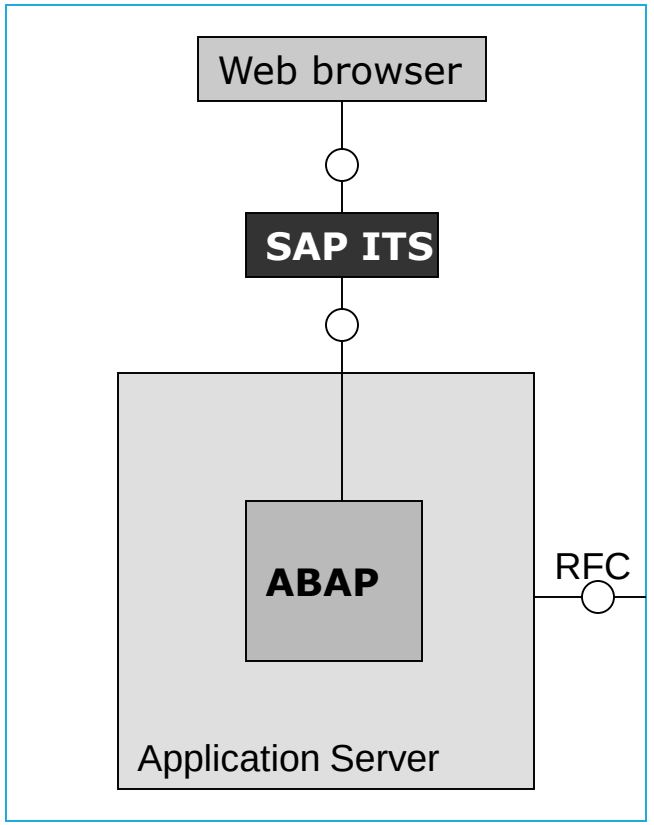
- Screen

Programming Language

- ABAP

Communication Interface

- RFC
- 3rd party products through connectors & gateways



SAP ICM

Based on the highly-scalable infrastructure, new technologies are used as of SAP Web AS 6.10 to process HTTP requests (and other protocols) directly from the Internet or to send HTTP client requests to the Internet. To achieve this, the SAP Kernel has been extended with the Internet Communication Manager (ICM) process.

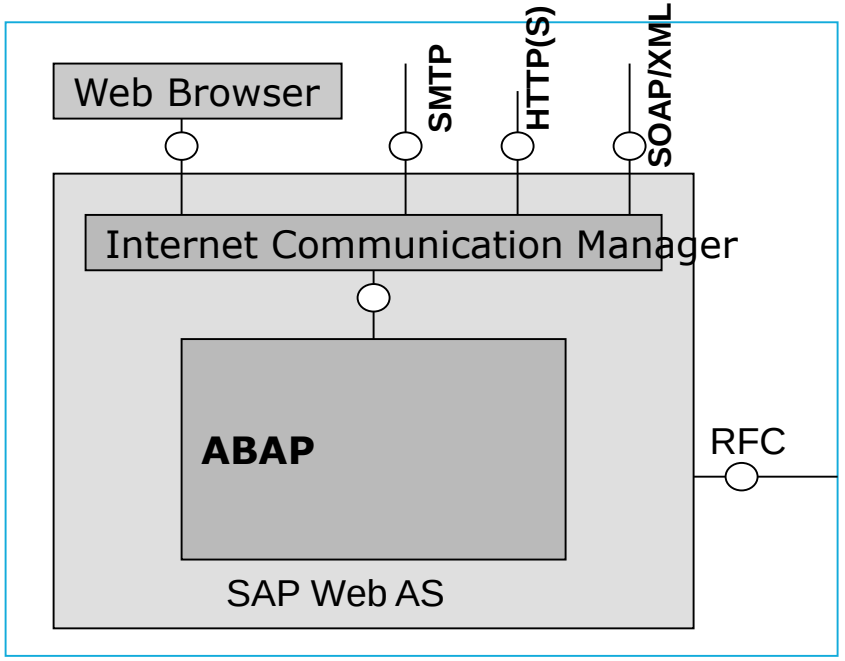
The ICM process forwards requests to the Internet Communication Framework (ICF), which supports numerous programming models. This is how the SAP CRM, SAP BW, and SAP XI software components use this infrastructure. A programming model for such applications are the Business Server Pages (BSPs).

User Access
- SAP GUI
- Web browser or mobile devices

User Interface
- Screen
- BSP

Programming Language
- ABAP

Communication Interface
- RFC
- SMTP
- HTTP(S)
- SOAP/XML



SAP ICM Components

Thread Control: This thread accepts the incoming TCP/IP requests and creates (or raises) a worker thread from the thread pool to process the request.

Worker Thread: This thread handles requests and responses for a connection. A worker thread contains an I/O handler for the network input and output, and various plug-ins for the different supported protocols.

Watchdog: A worker thread usually waits for a response (whether it is client or server); if a timeout occurs, the watchdog takes over the task of waiting for the response. The worker thread can then be used for other requests.

Signal Handler: Processes signals that are sent from the operating system or another process (such as the ABAP dispatcher).

Connection Info: Table with information for each existing network connection.

Memory Pipes: These memory-based communication objects allow data transfer between the ICM and the ABAP work processes.

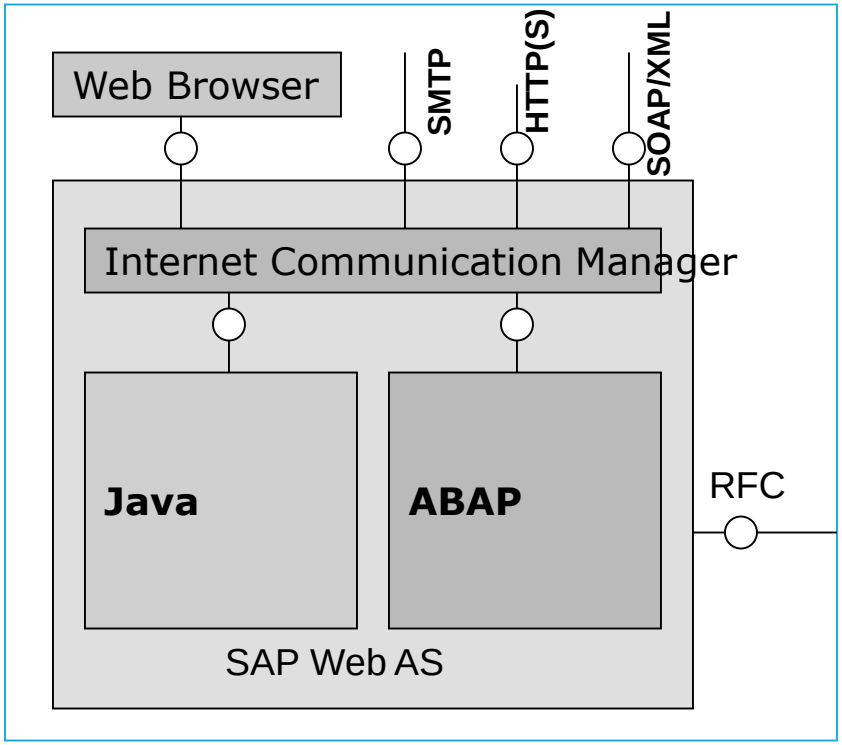
SAP AS Java

User Access
- SAP GUI
- Web browser or mobile devices

User Interface
- Screen
- BSP
- JSP

Programming Language
- ABAP
- Java

Communication Interface
- RFC
- SMTP
- HTTP(S)
- SOAP/XML



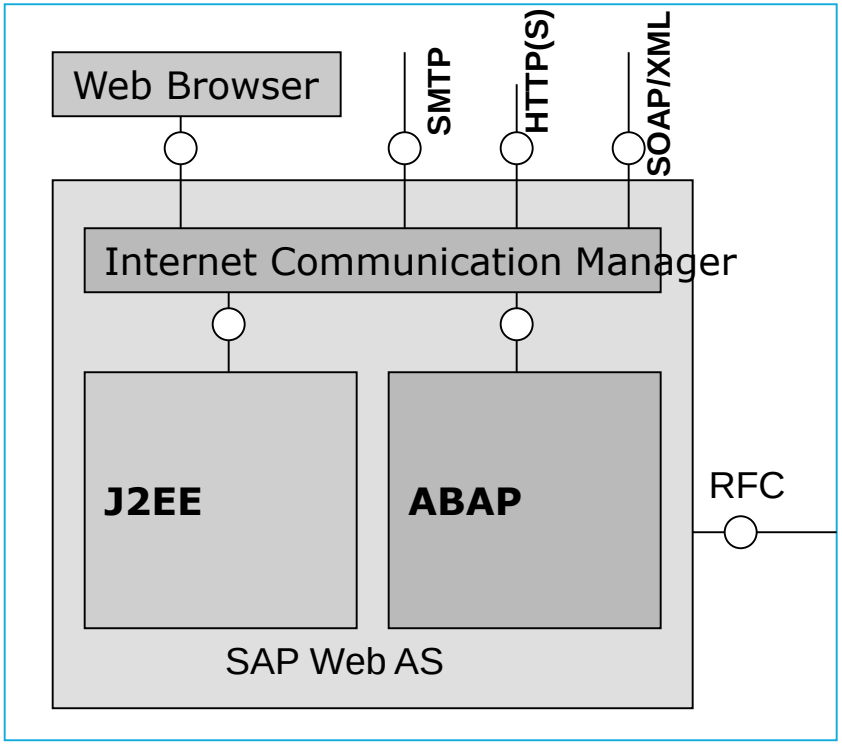
Web Dynpro – Java

User Access
- SAP GUI
- Web browser or mobile devices

User Interface
- Screen
- Web Dynpro for Java
- BSP
- JSP

Programming Language
- ABAP
- Java

Communication Interface
- RFC
- SMTP
- HTTP(S)
- SOAP/XML



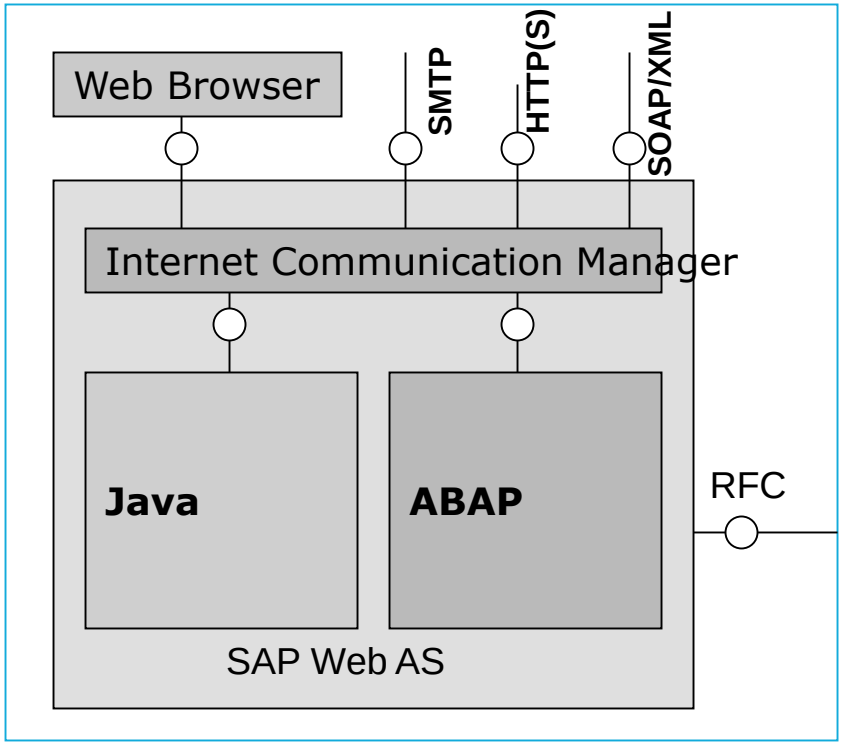
Web Dynpro – ABAP

User Access
- SAP GUI
- Web browser or mobile devices

User Interface
- Screen
- Web Dynpro for Java
- Web Dynpro for ABAP
- BSP
- JSP

Programming Language
- ABAP
- Java

Communication Interface
- RFC
- SMTP
- HTTP(S)
- SOAP/XML

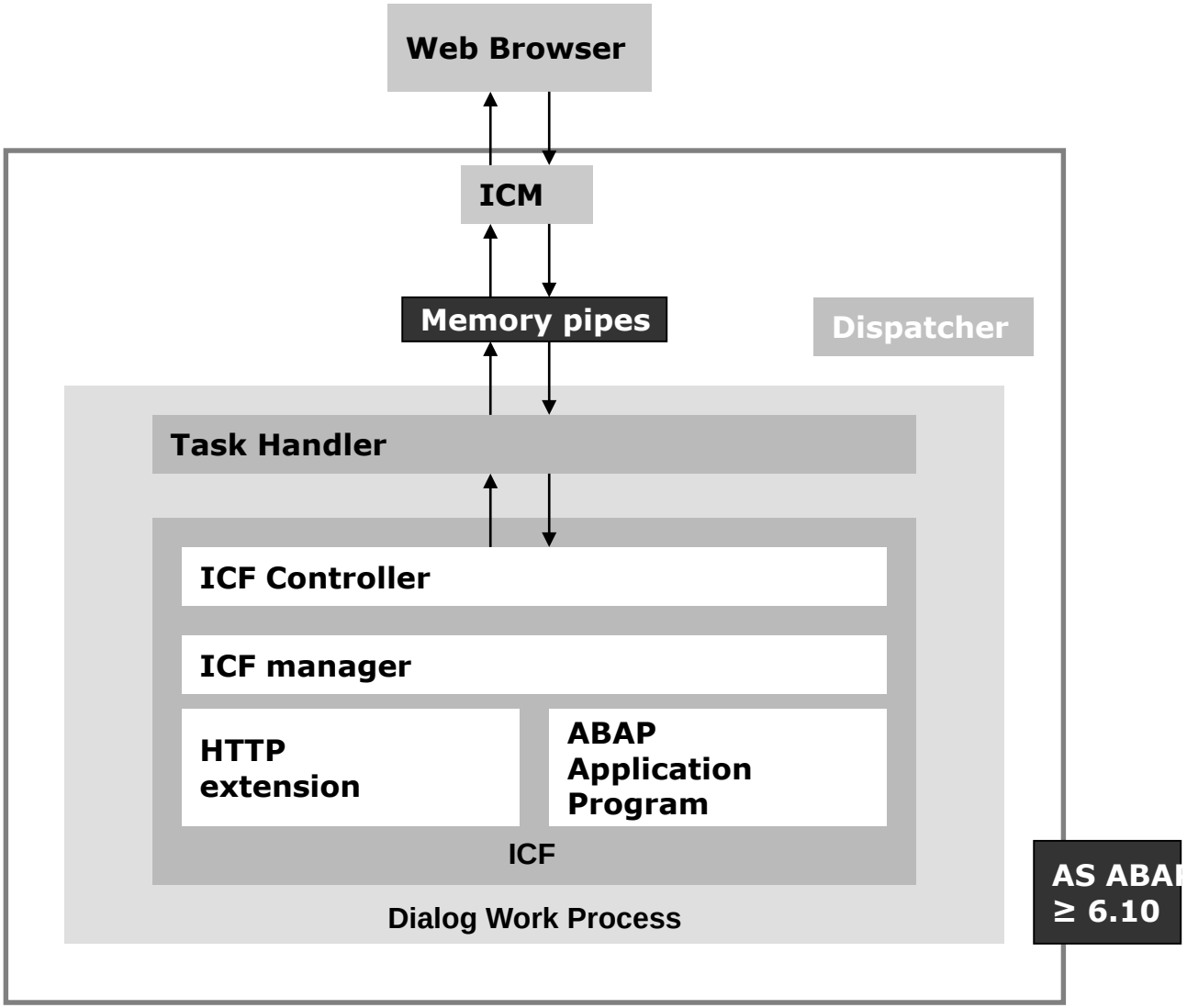


Internet Communication Framework – ICF

- The Internet Communication Framework (ICF) provides an environment for handling Web requests in the ABAP work process of an SAP system. 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.
- 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.
- ICF services can be active or inactive, which is indicated by different colors in transaction SICF

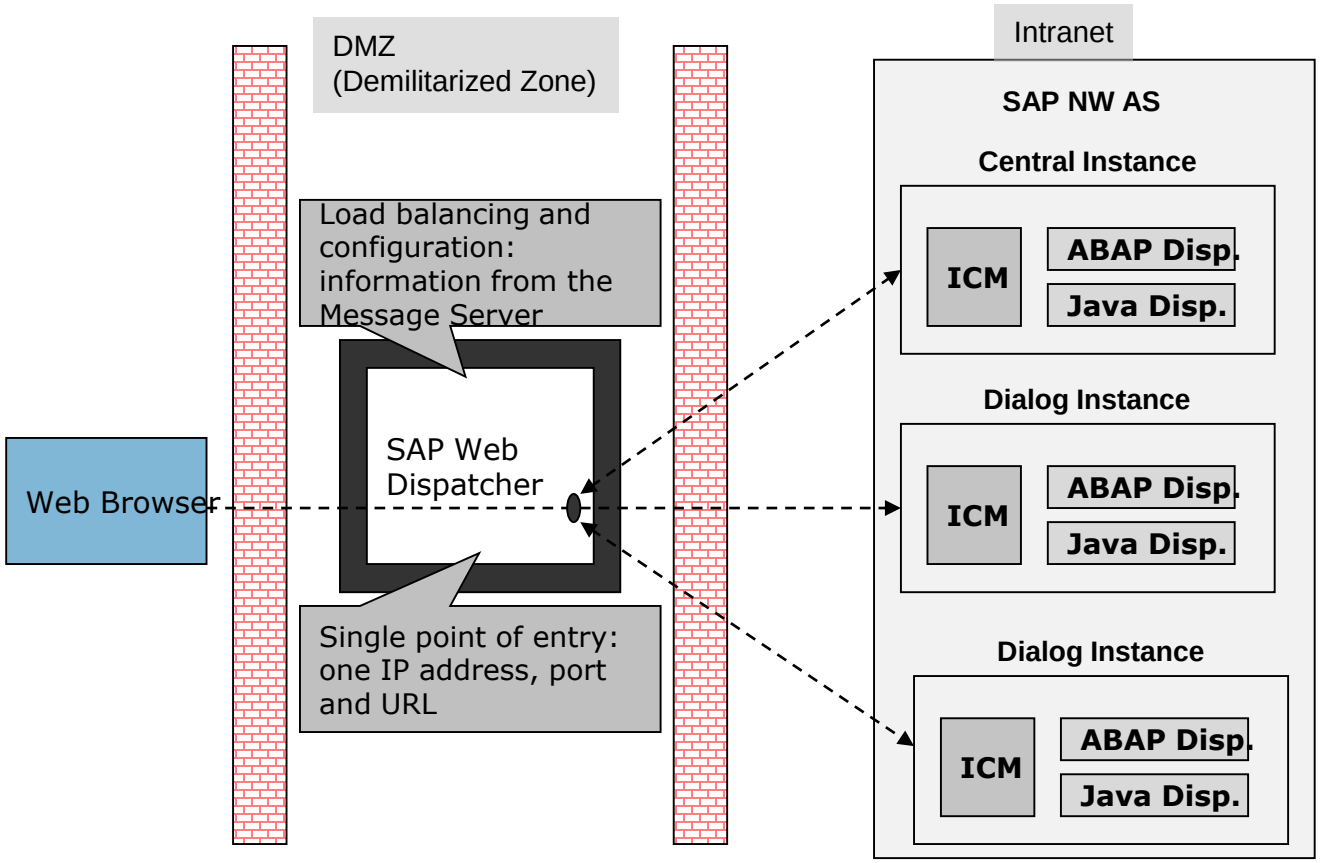
Status of ICF Services		
Status	**Color in SICF**	**Meaning**
Active	**Black**	Service can be called
Inactive	Gray	Service explicitly deactivated
Inactive	**Blue**	Service implicitly deactivated

ICM – Interaction Model



SAP Web Dispatcher

The SAP Web Dispatcher, delivered as of SAP Web AS 6.20, acts like a software Web switch. It is a stand-alone program that you can run on a separate host without any additional software. In this way, the SAP Web Dispatcher implements a central entry point for HTTP(S) requests to an SAP system, including load distribution across multiple instances



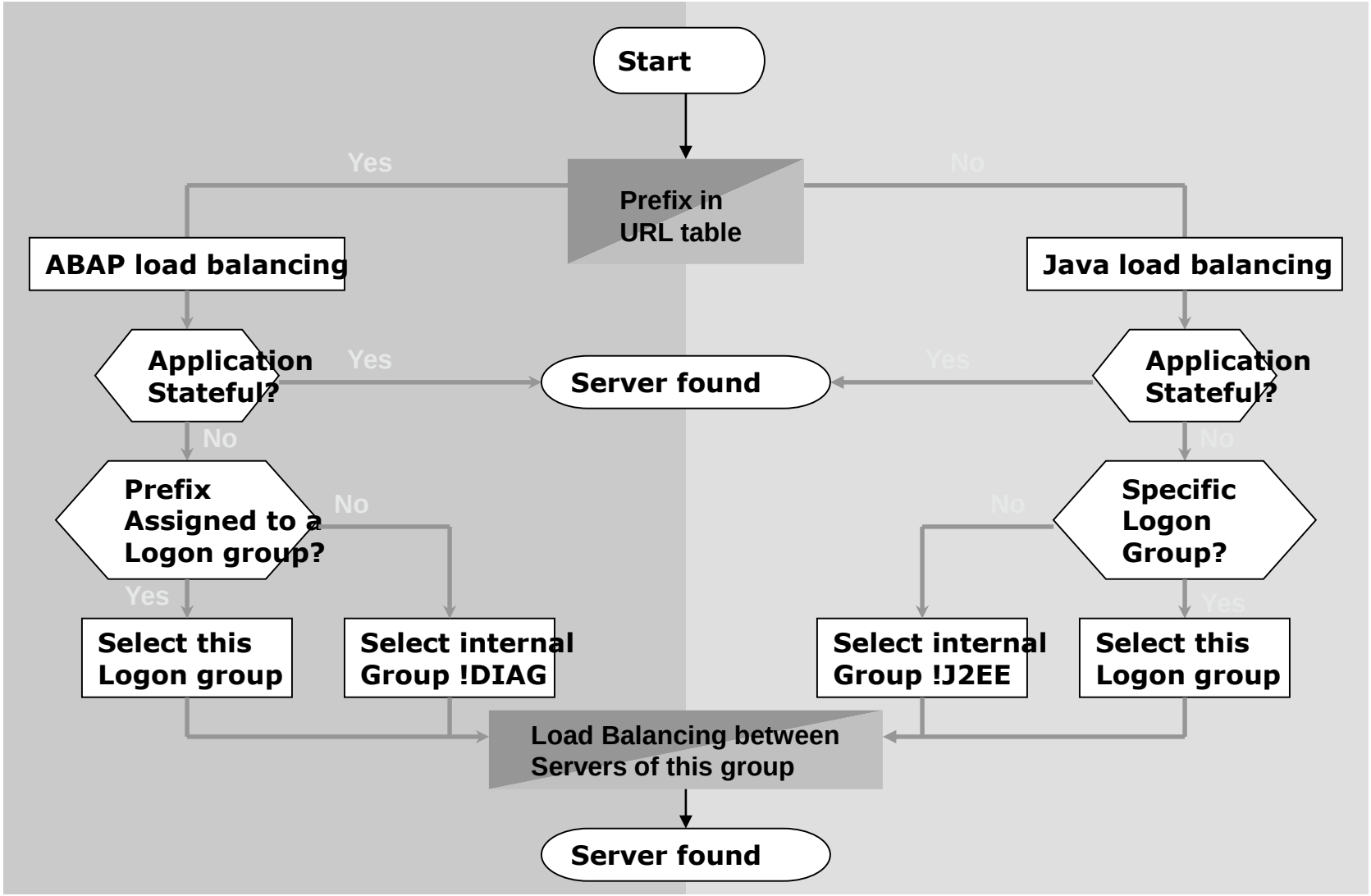
SAP Web Dispatcher – Function

The SAP Web Dispatcher ultimately forwards an HTTP(S) request to a specific application server. This section outlines the criteria by which this is performed. An HTTP request (or unpacked HTTPS request) is assigned to a server in two stages:

1. First, the SAP Web Dispatcher determines whether the incoming HTTP request is to be forwarded to an ABAP or Java server. It then finds a group of servers in the SAP system that can execute the request.
2. Load balancing is then carried out within this group. After the SAP Web Dispatcher has identified a server, it forwards the request to the ICM of the relevant application server.

A SAP Web Dispatcher can distribute requests for only one SAP system. If multiple SAP systems are required, you have to set up and start separate SAP Web Dispatcher processes for each of the respective systems (which can run together on one computer).

From the HTTP(S) Request to the Application Server



SAP Web Dispatcher – Operation

As of SAP Web AS 6.40, you can also start the SAP Web Dispatcher without a profile file. For this bootstrap option (started with command sapwebdisp -bootstrap), the following steps are carried out:

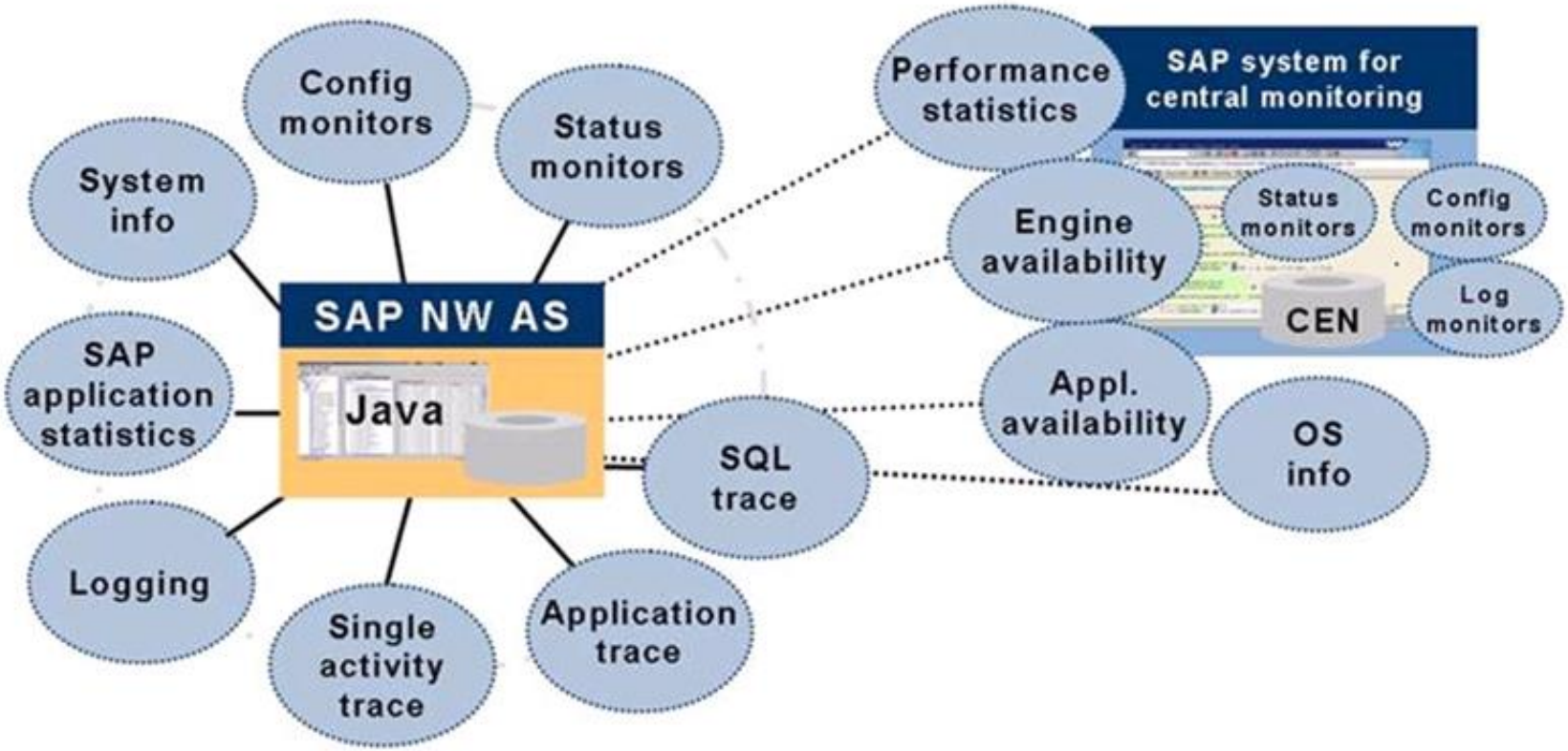
1. If the profile file sapwebdisp.pfl does not exist already, it is created based on interactive entries.
2. If the authorization file icmauth.txt does not exist, it is created and a user is entered for Web administration (see below).
3. The SAP Web Dispatcher is started with the profile file created.

As of SAP Web AS 6.40, a Web-based interface is available for SAP Web Dispatcher administration and monitoring.

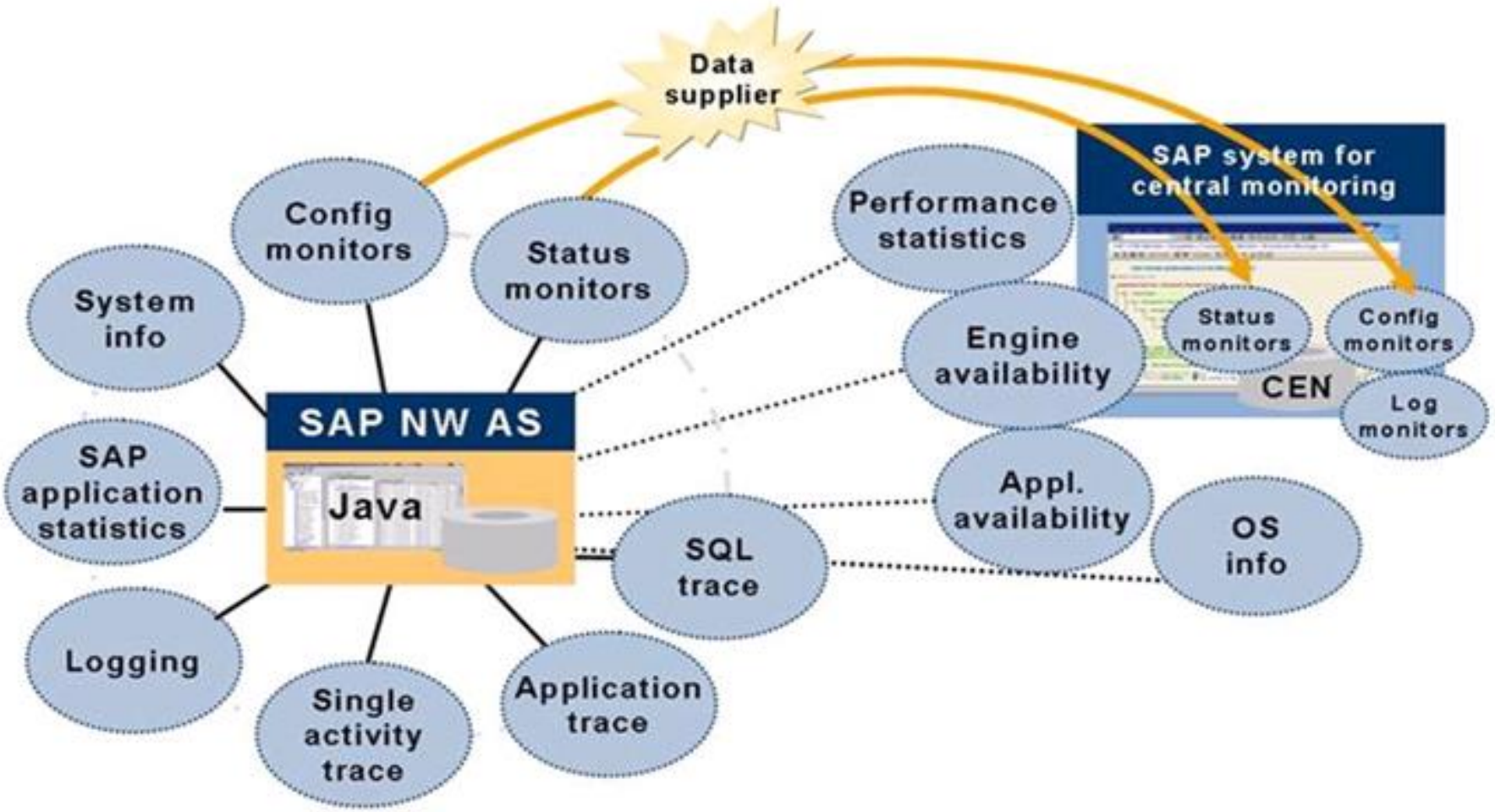
Overview of Operations in Java and Patching

- Concept of SAP Net Weaver AS Java monitoring tools
- Concept of Monitoring Infrastructure
- Configuration for Central Monitoring System
- Operate the integrated and the central Log Viewer
- Configuration of Log Configuration service
- GRMG
- Patching Java Stack

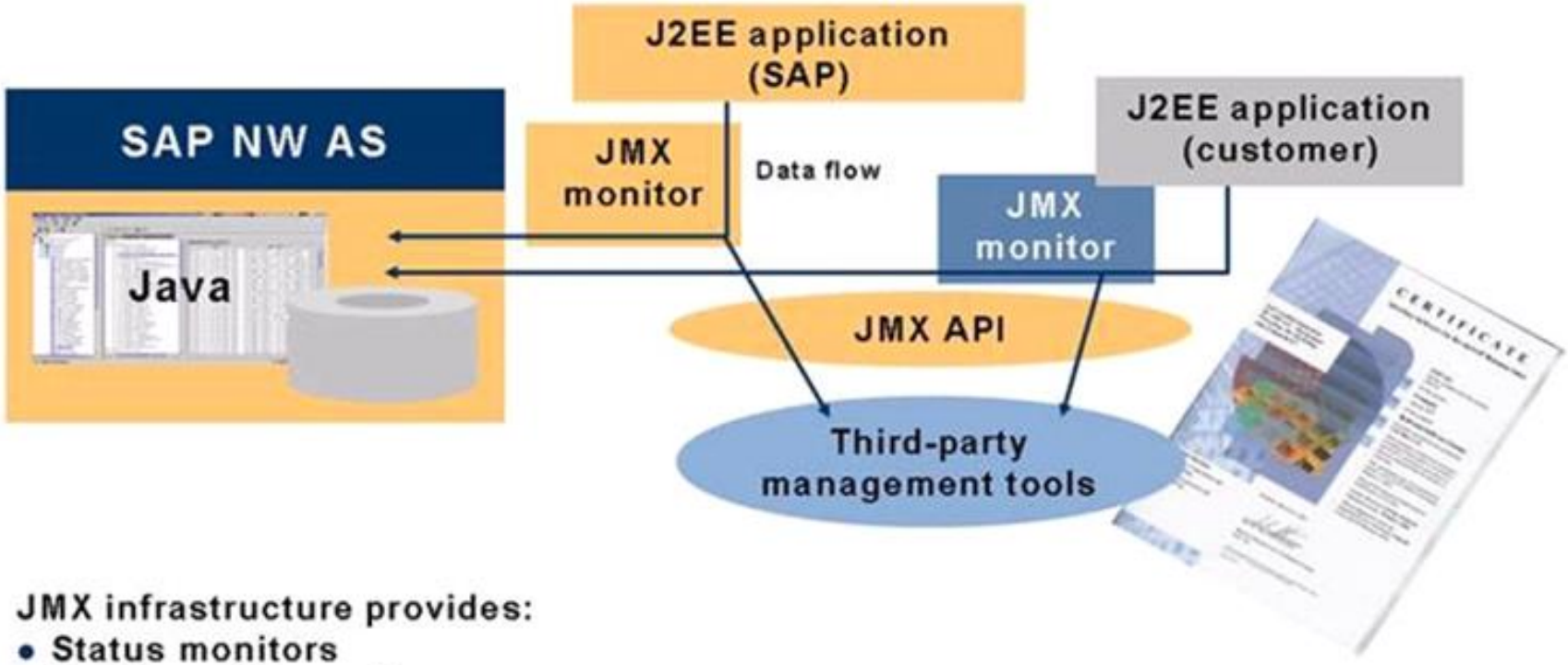
Java Monitoring



Central Monitoring of SAP Netweaver AS Java



Monitoring Infrastructure (JMX)

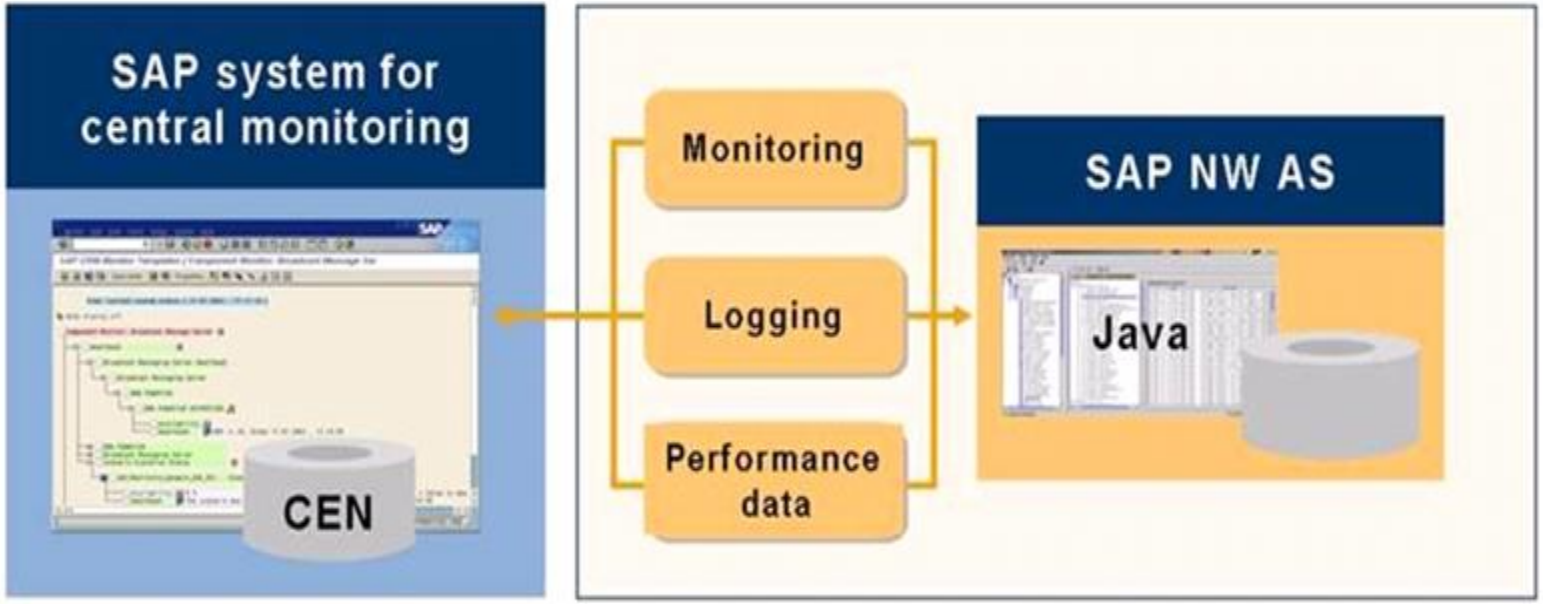


JMX infrastructure provides:
- Status monitors
- Configurable monitors

SAP provides customers with:
- SAP templates for JMX monitors to integrate your own J2EE applications

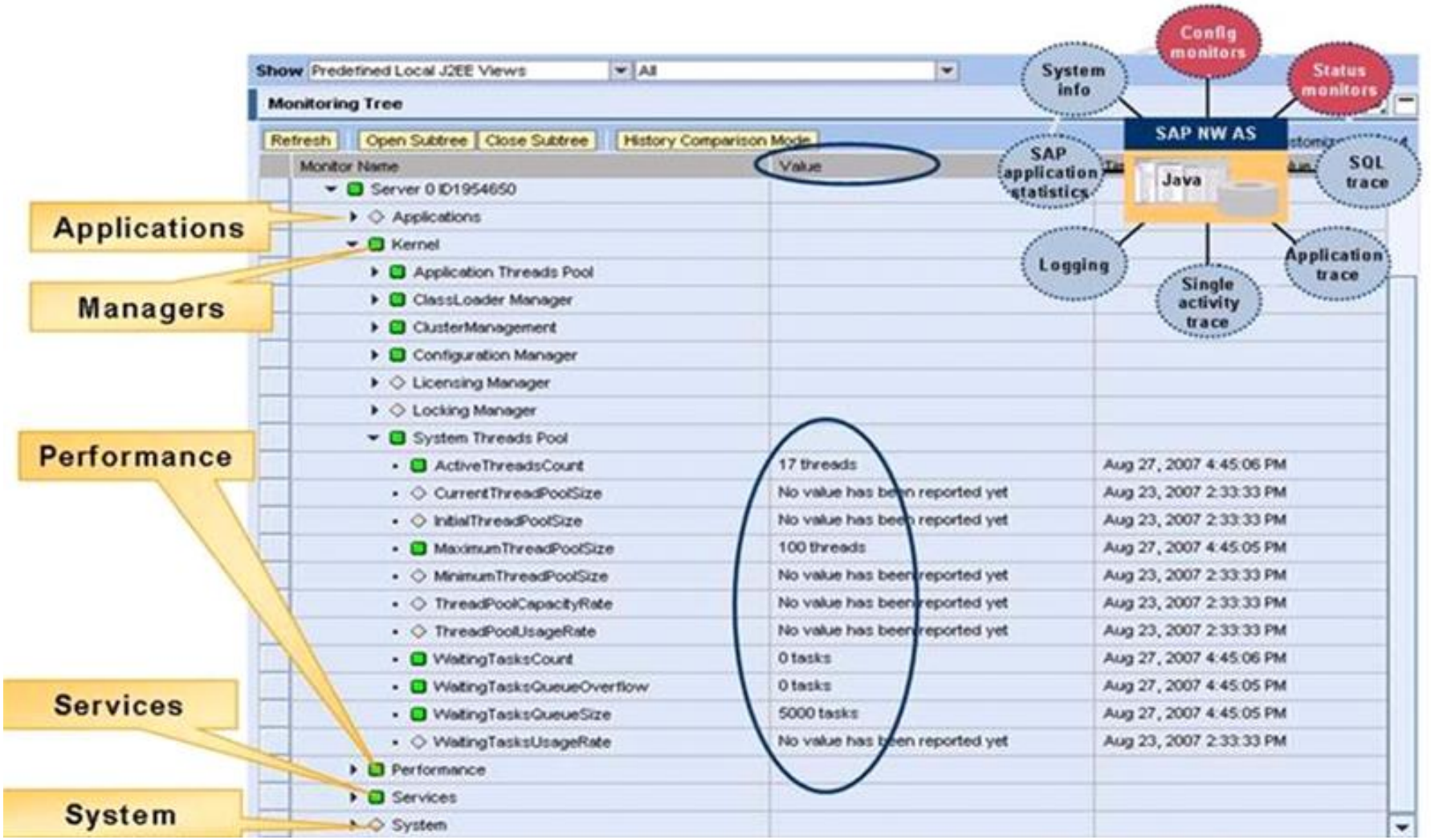
Connection to third-party management tools:
- Display all current values
- Adjust group configurations
- Create/delete groups and install/uninstall monitor nodes

Monitoring - Tools

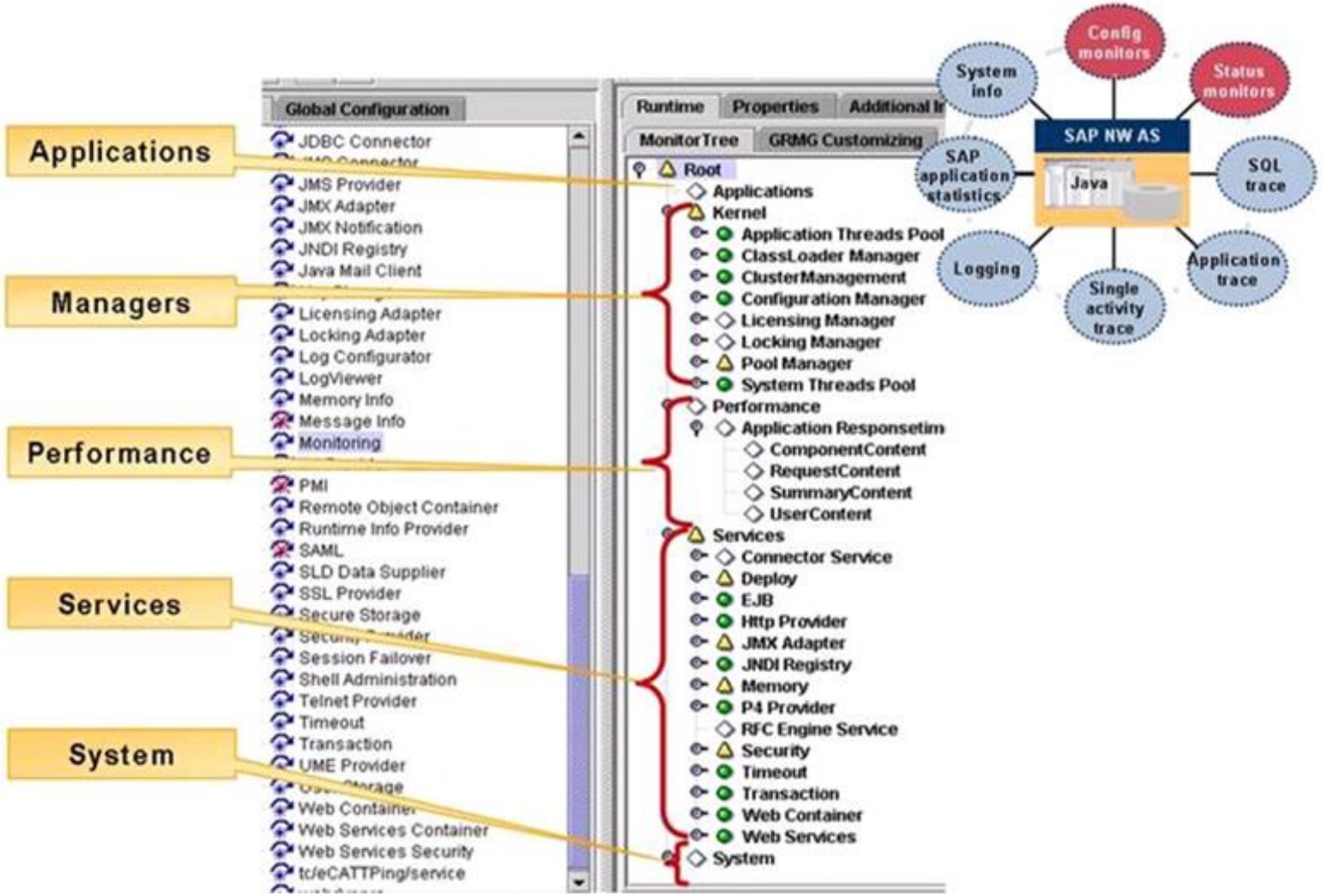


Monitoring Browser in the NWA

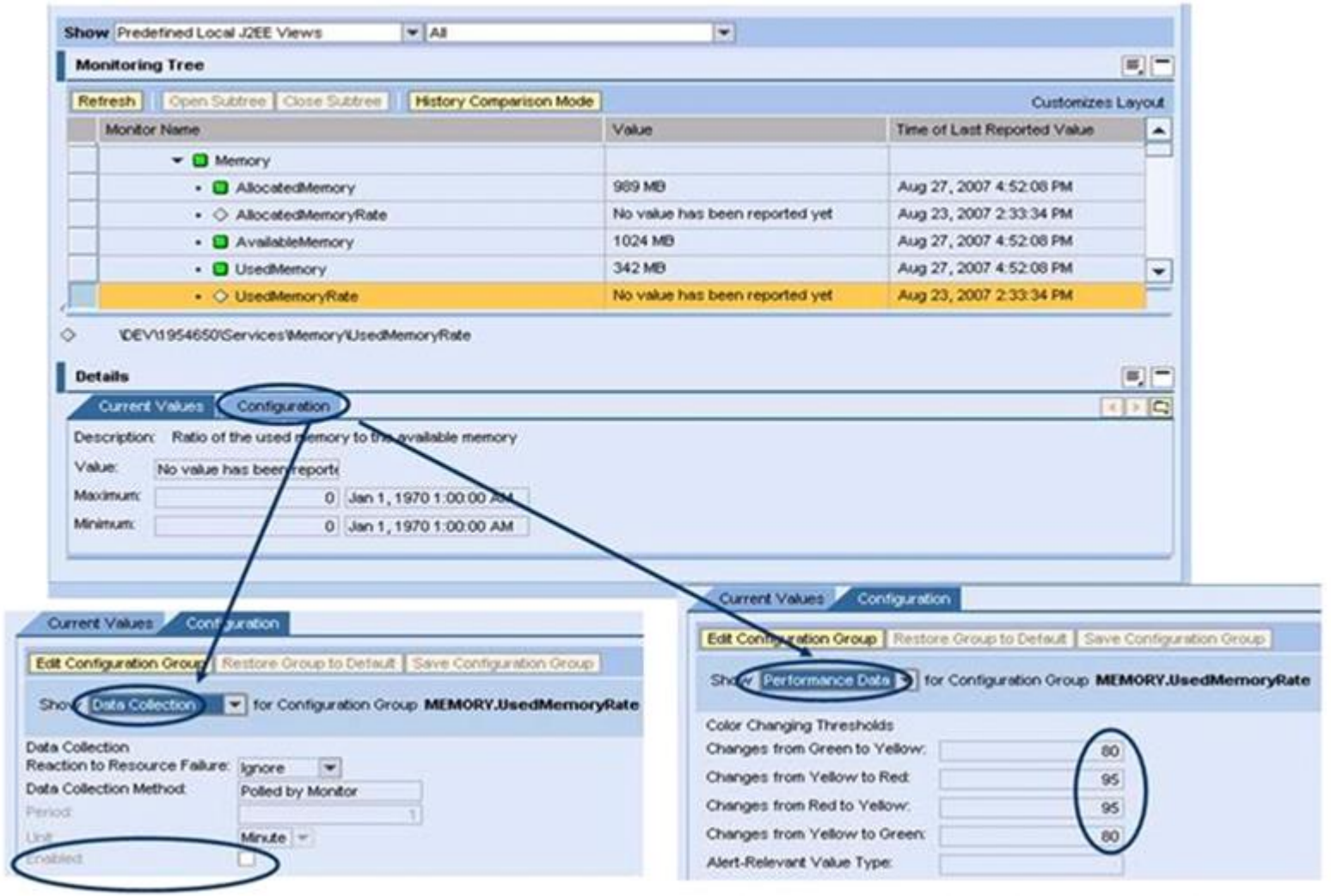
Monitoring Tree in the NWA



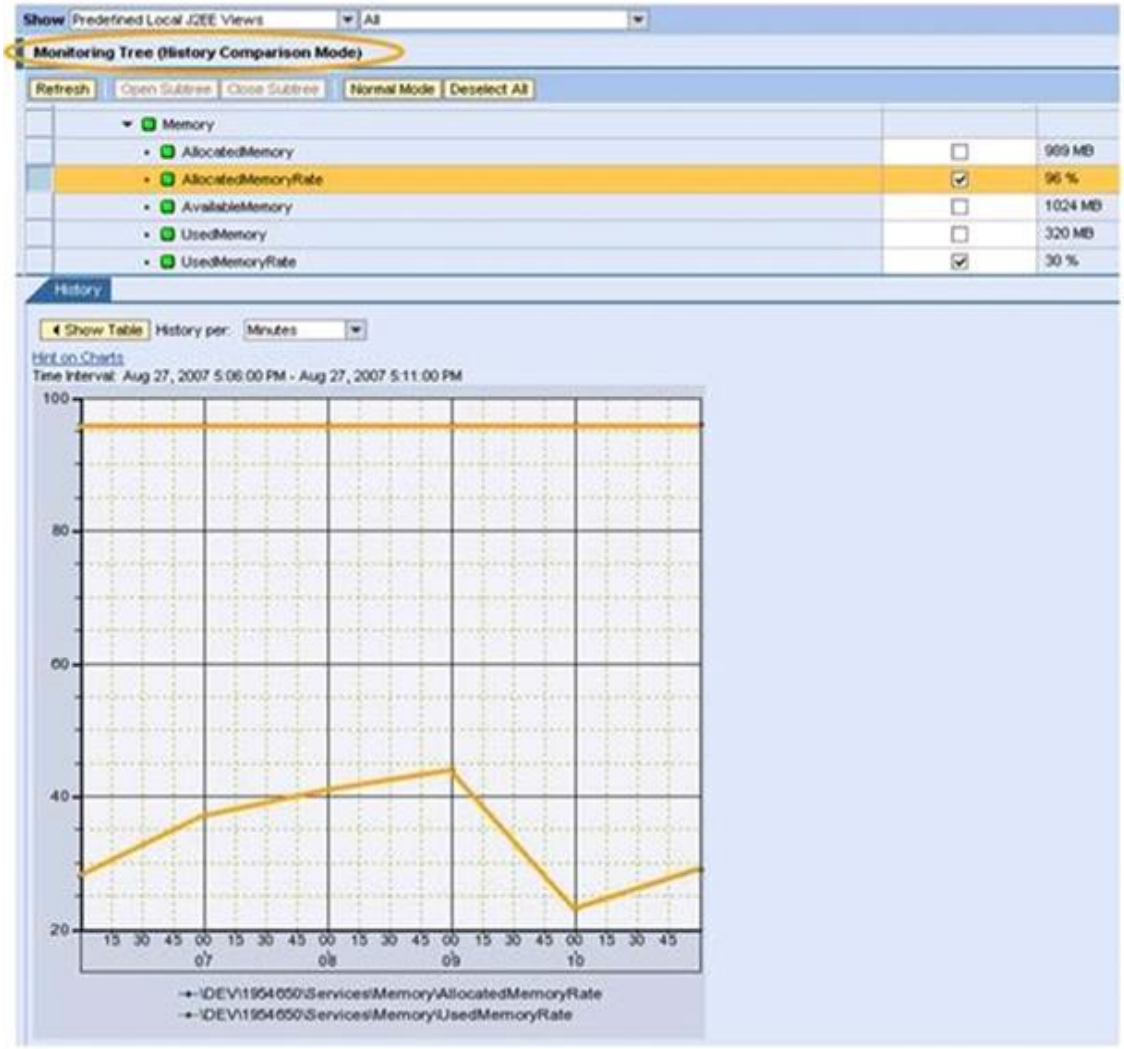
Visual Administrator – Monitoring Service



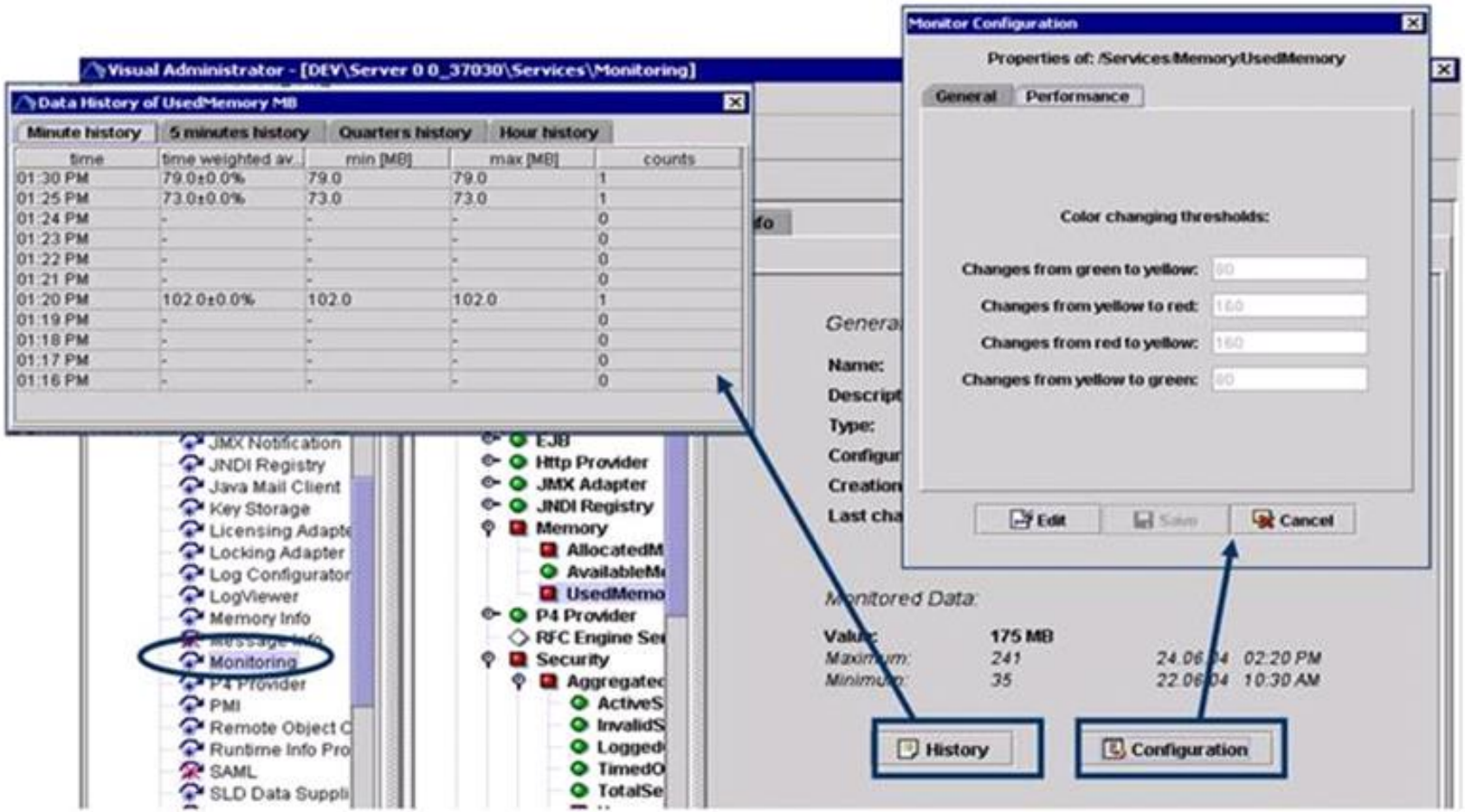
Configuration in the NWA



Comparative History in the NWA



History and Configuration



Lunch Break

Monitoring Service: Classification

Important Alert Monitors

Connections Manipulator Monitor
Application Threads Pool Monitor
System Threads Pool Monitor

Cluster Management Monitor

Memory Monitor

Table Buffer Monitor
Configuration Manager Monitor

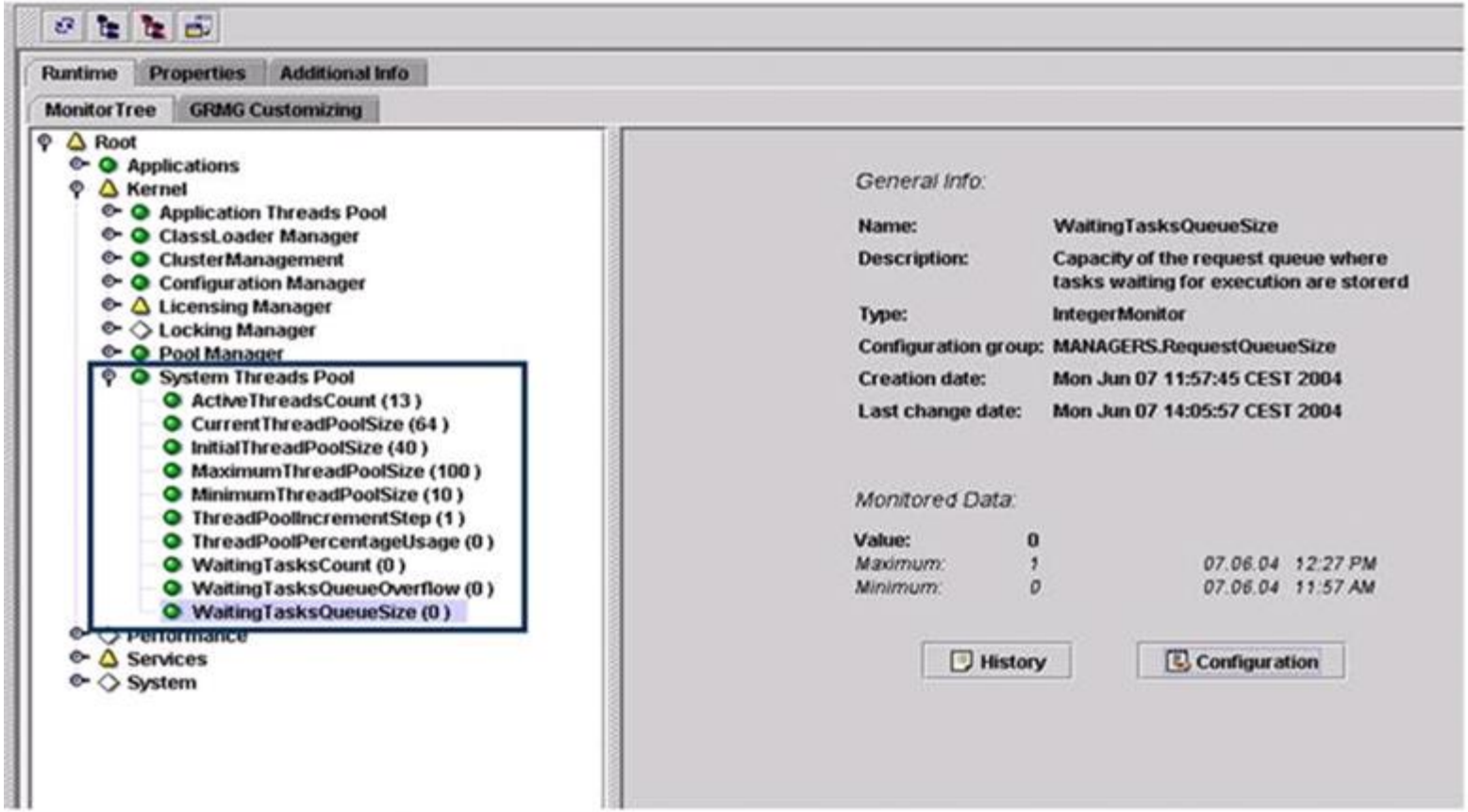
"Info" Monitors

ClassLoader Monitor
Locking Manager Monitor
Performance Monitor
Deploy Service Monitor
HTTP Provider Service Monitor
JNDI Registry Monitor
JMX Adapter Service Monitor
P4 Provider Monitor
Timeout Monitor
Web Container Monitor
Web Services Monitor

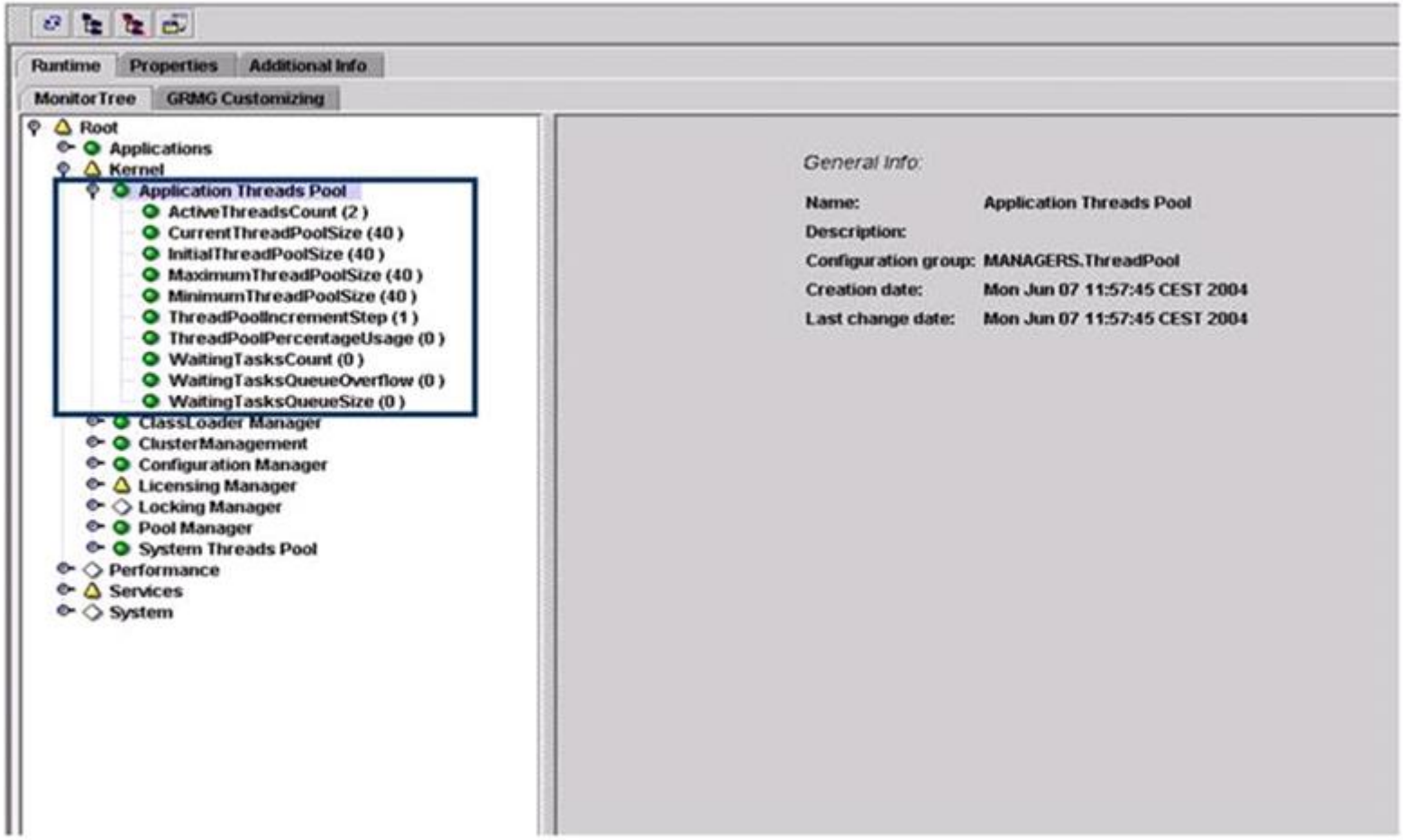
Other Useful Alert Monitors

Security Provider Monitor
Transaction Monitor
Log Configurator Monitor
Connector Service Monitor
Licensing Manager Monitor

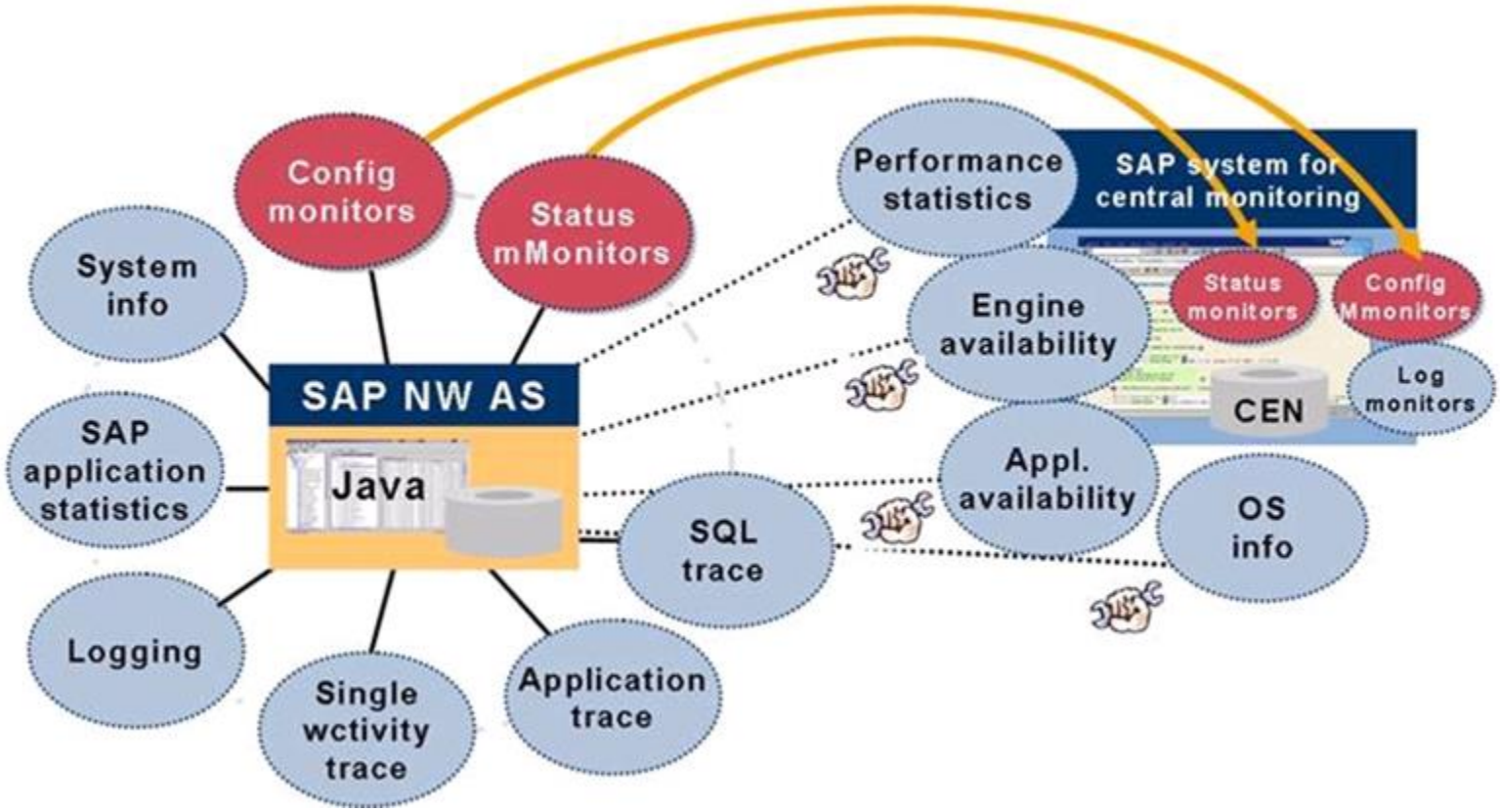
Monitor for the System Thread Pool



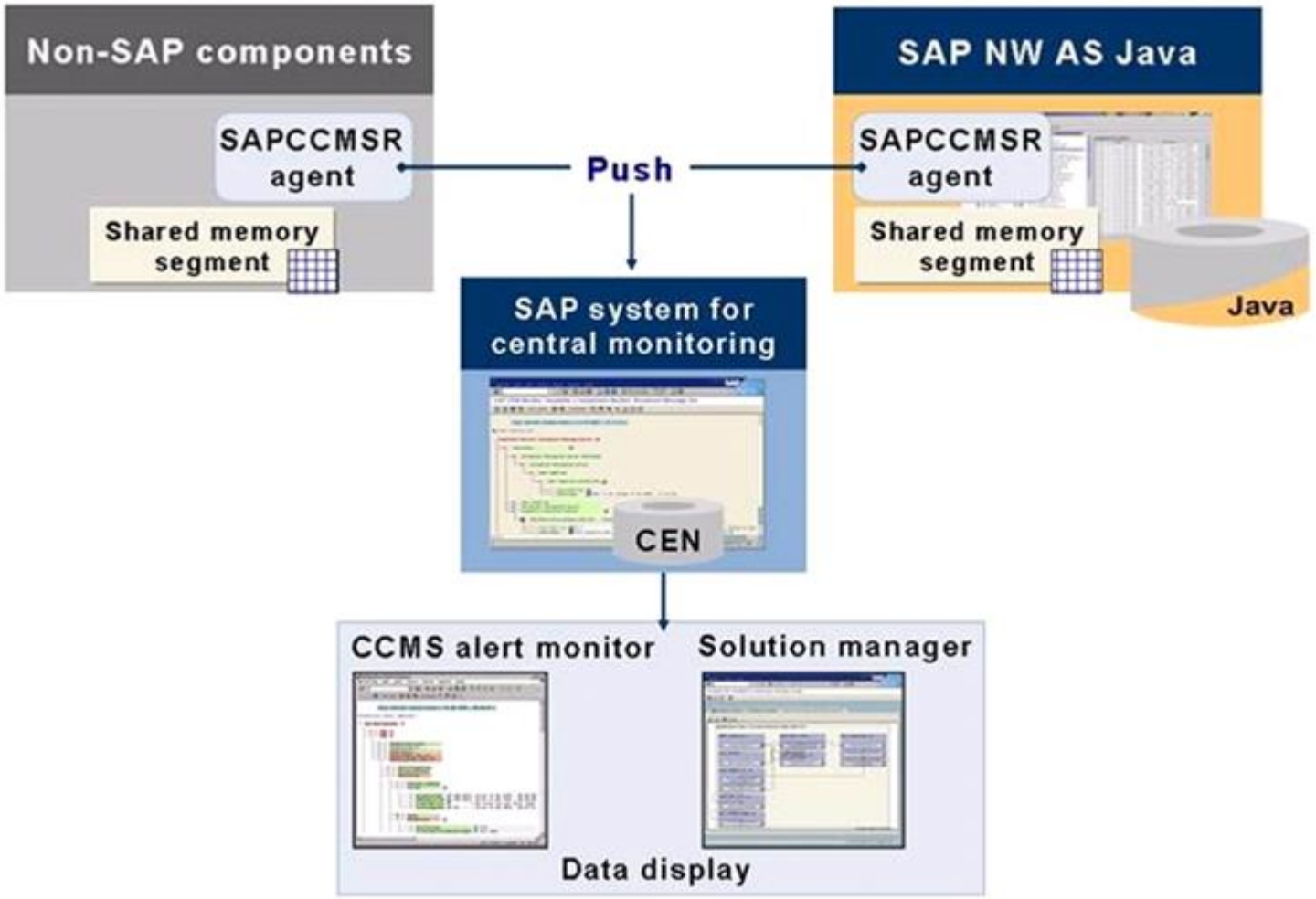
Monitor for Application Thread Pool



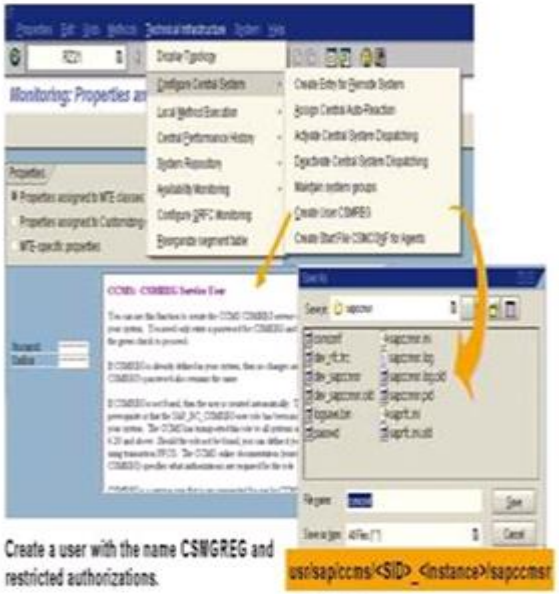
Connecting to a Central SAP ABAP Monitoring System



Data Transfer Using the SAPCCMSR Agent



Creating the CMSREG User and the CSMCONF File



Create a user with the name CSMGREG and restricted authorizations.

/usr/sap/ccms/<SID>_<instance>/sapccmsr

CSMCONF:

File containing all information about the central system that the agent requires for registration.

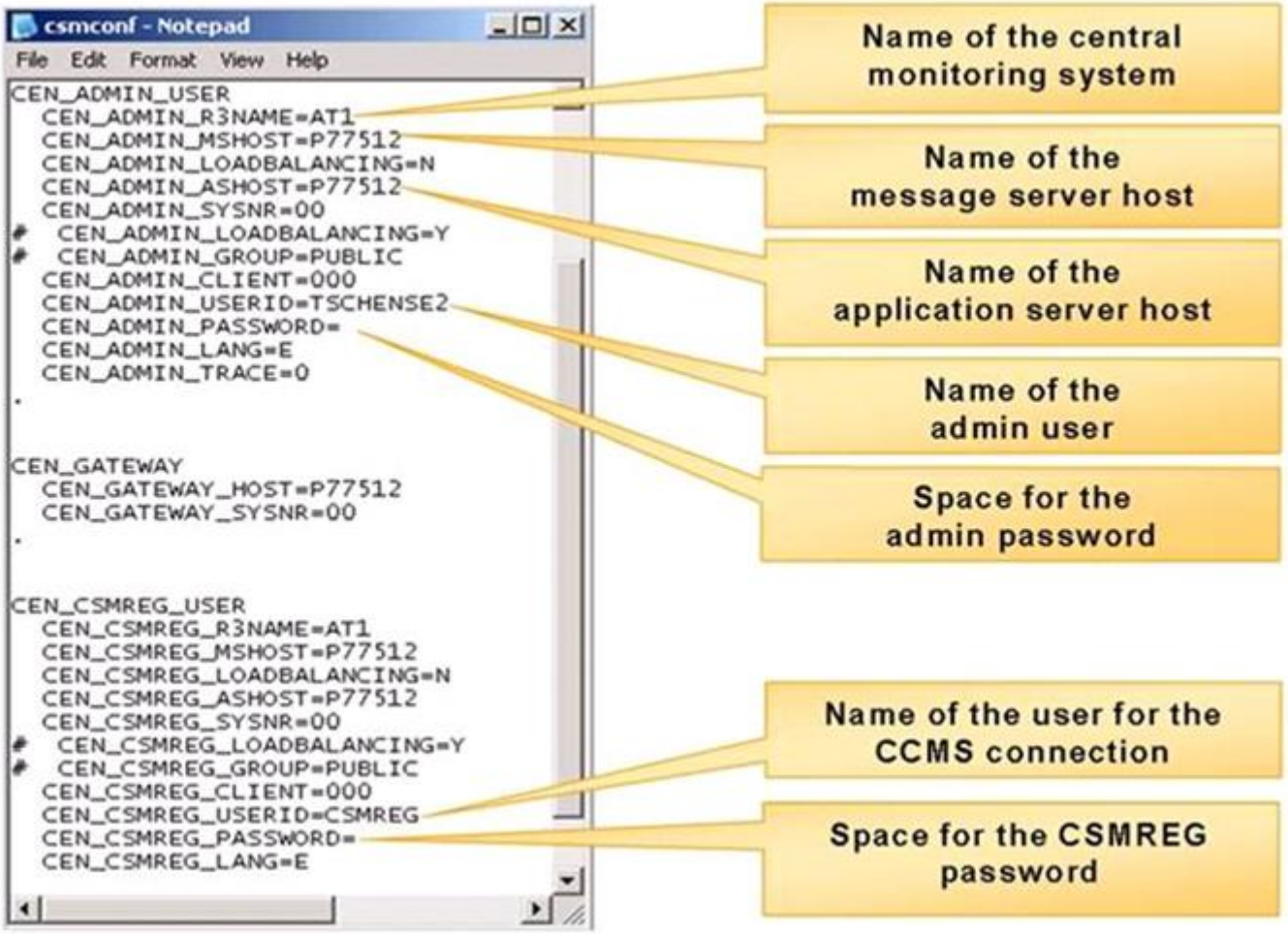
Installation Steps for the SAPCCMSR Agent:

Create the CSMREG user in the central monitoring system (transaction RZ21)

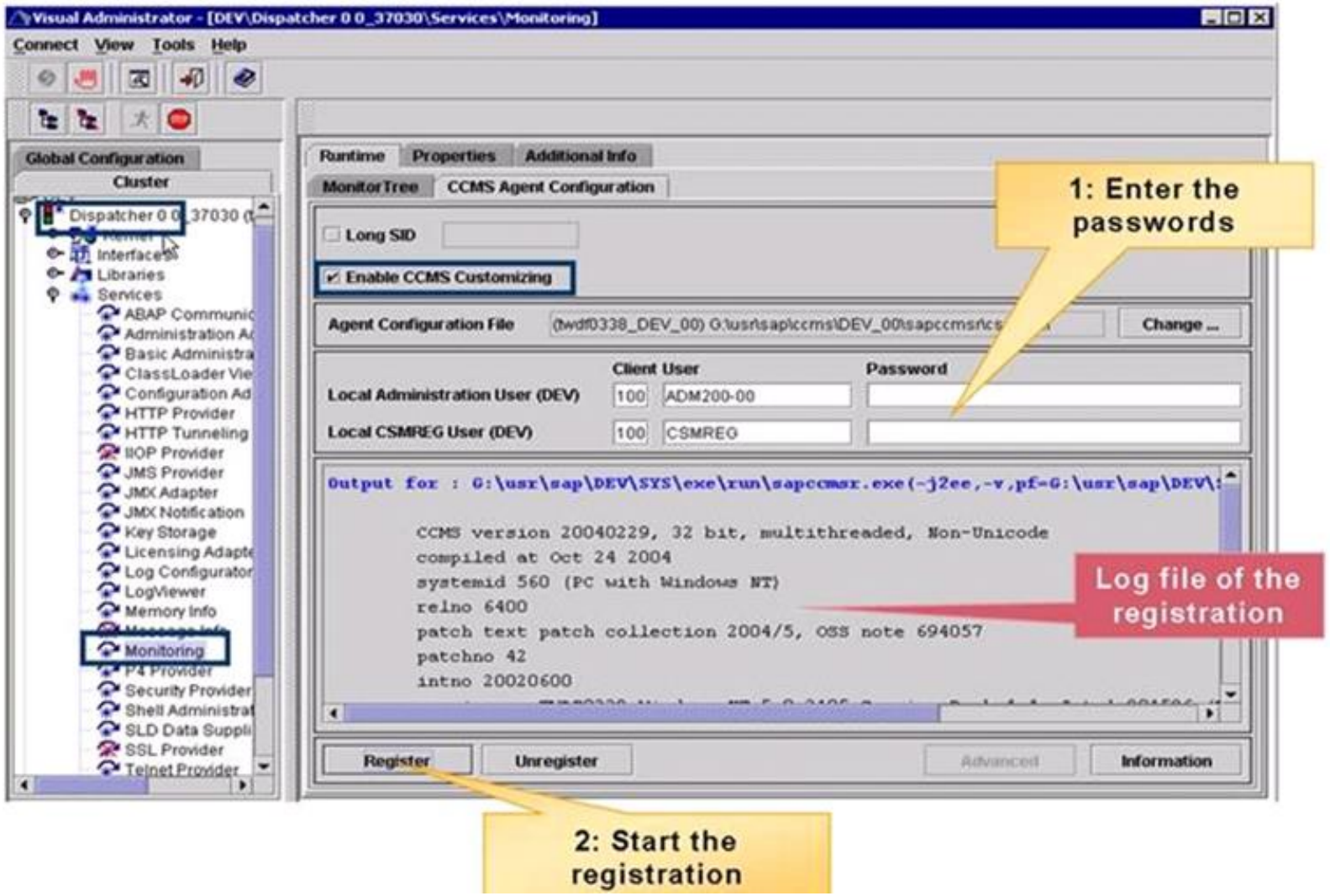
Create the CSMCONF file in the central monitoring system (transaction RZ21)

Register the agent in the Visual Administrator (*Dispatcher* → *Services* → *Monitoring*)

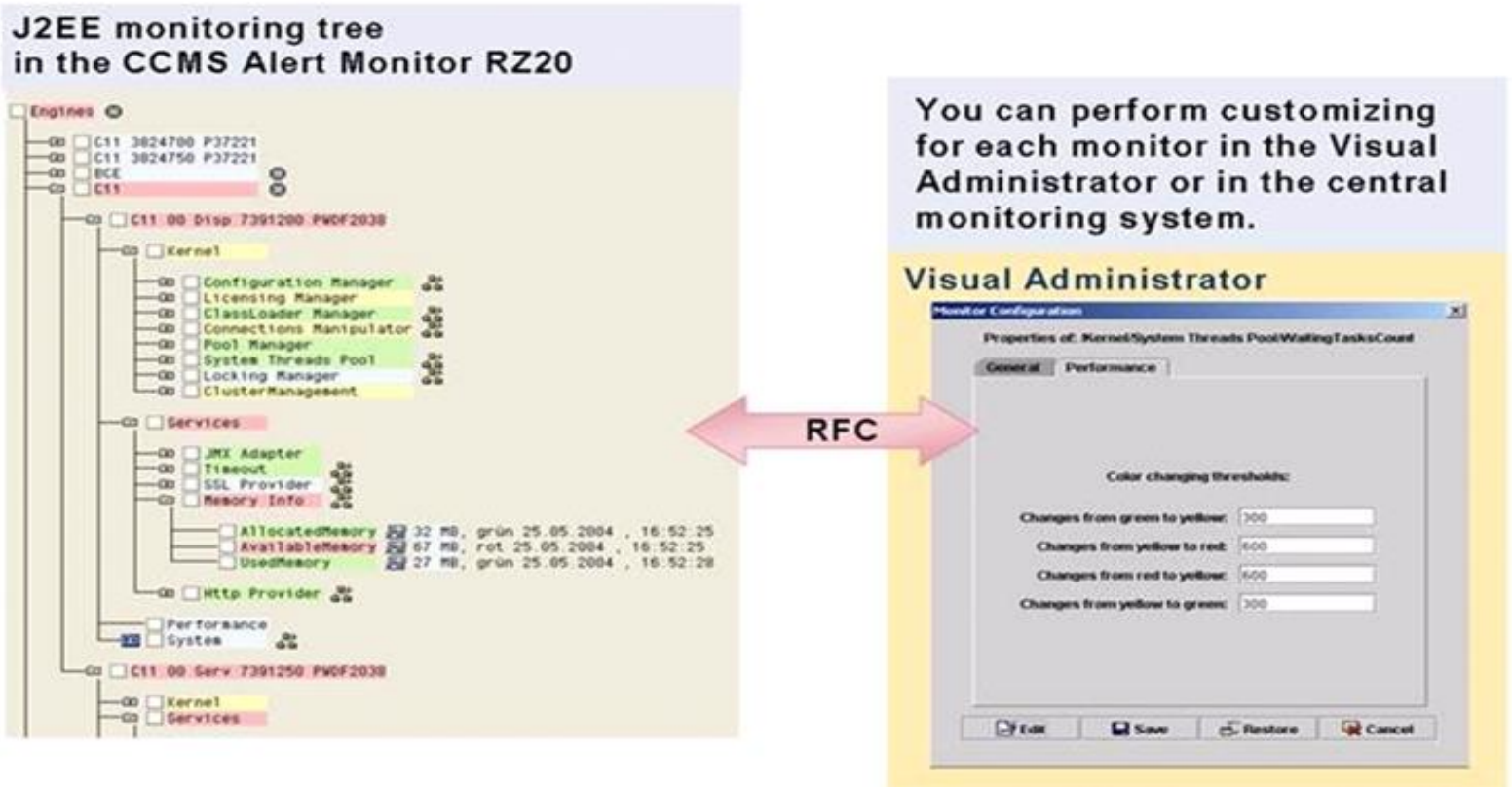
Contents of the CSMCONF



Agent Registration in the Visual Administrator

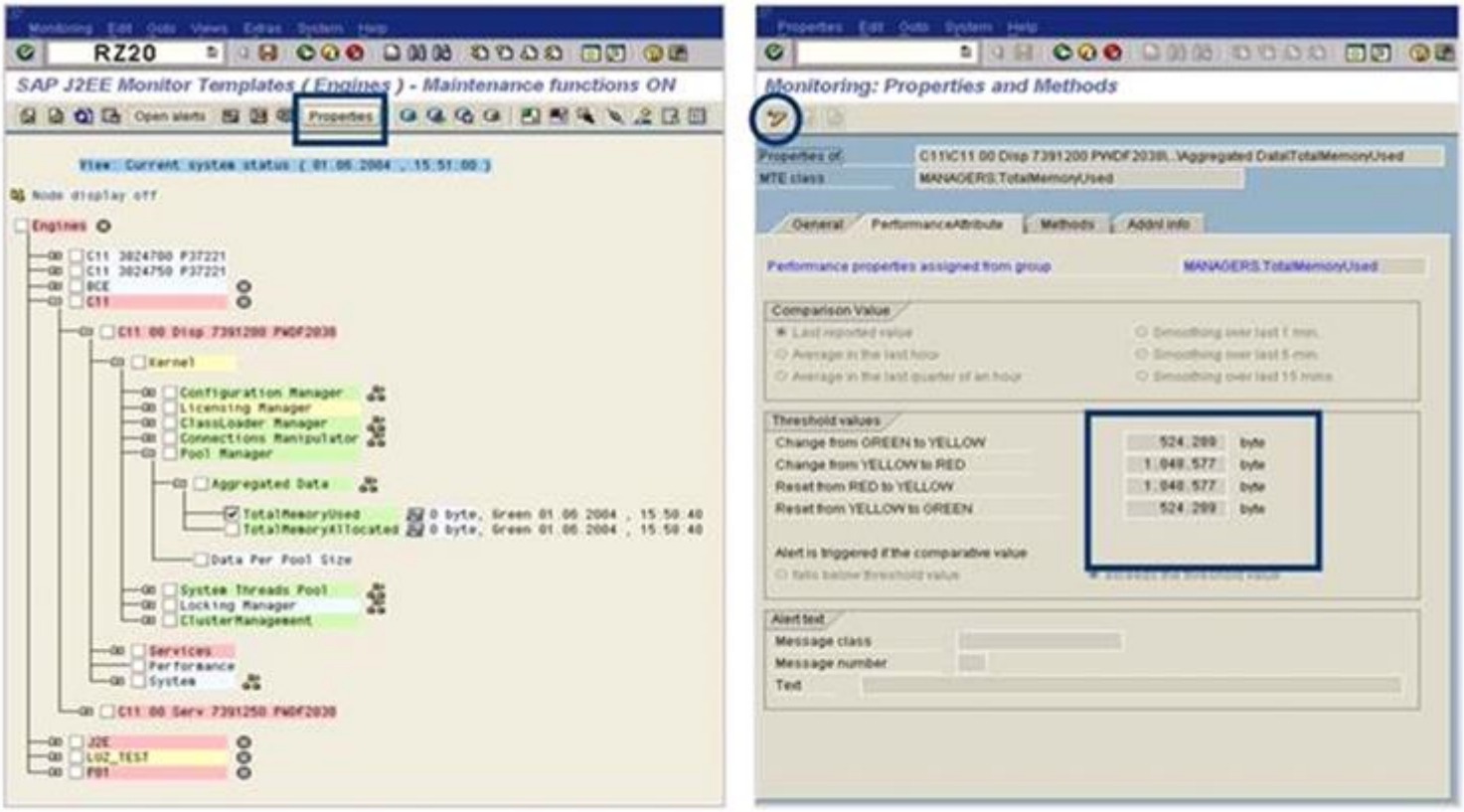


Threshold Value Maintenance



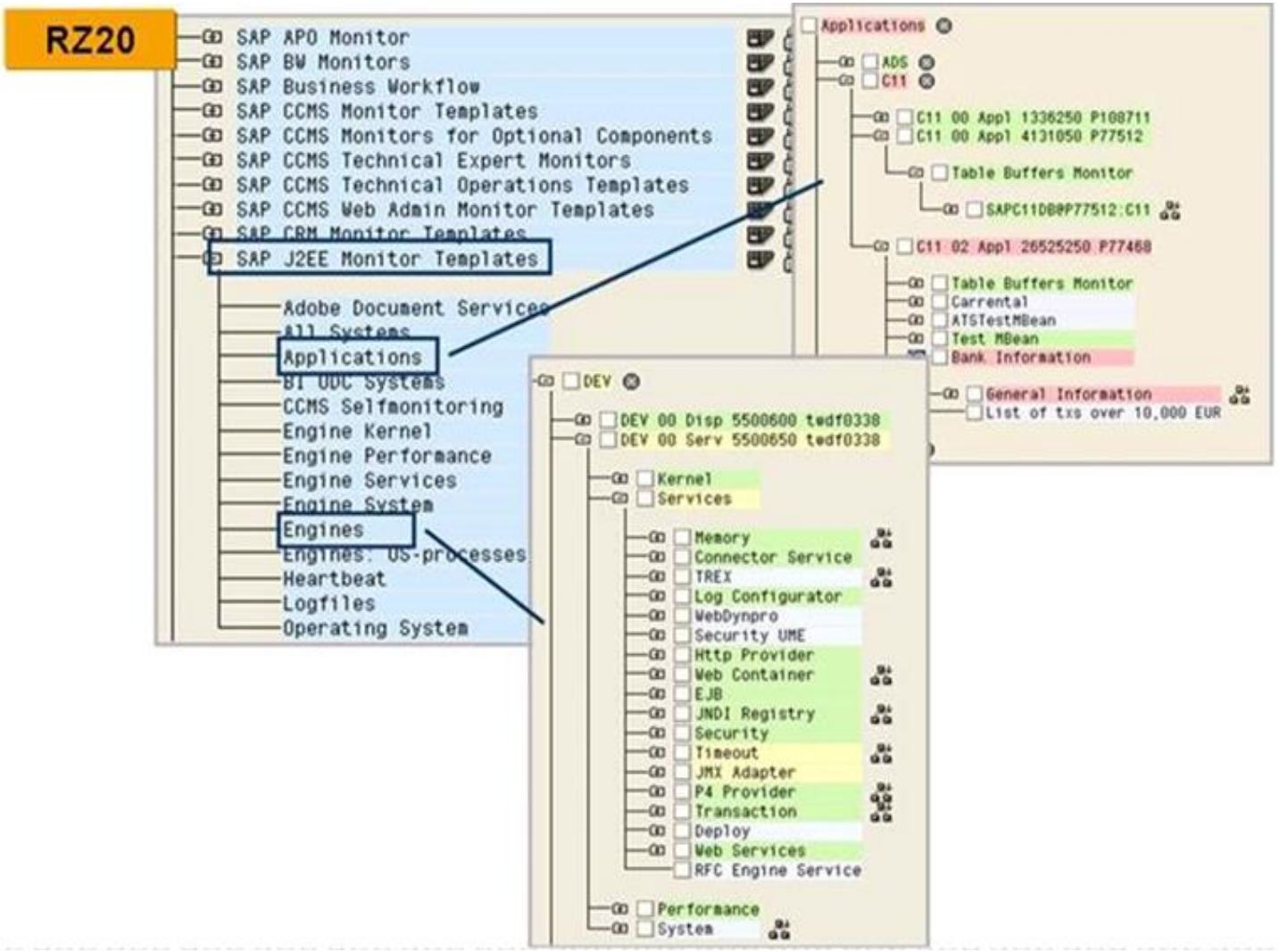
Break

Central Monitoring System

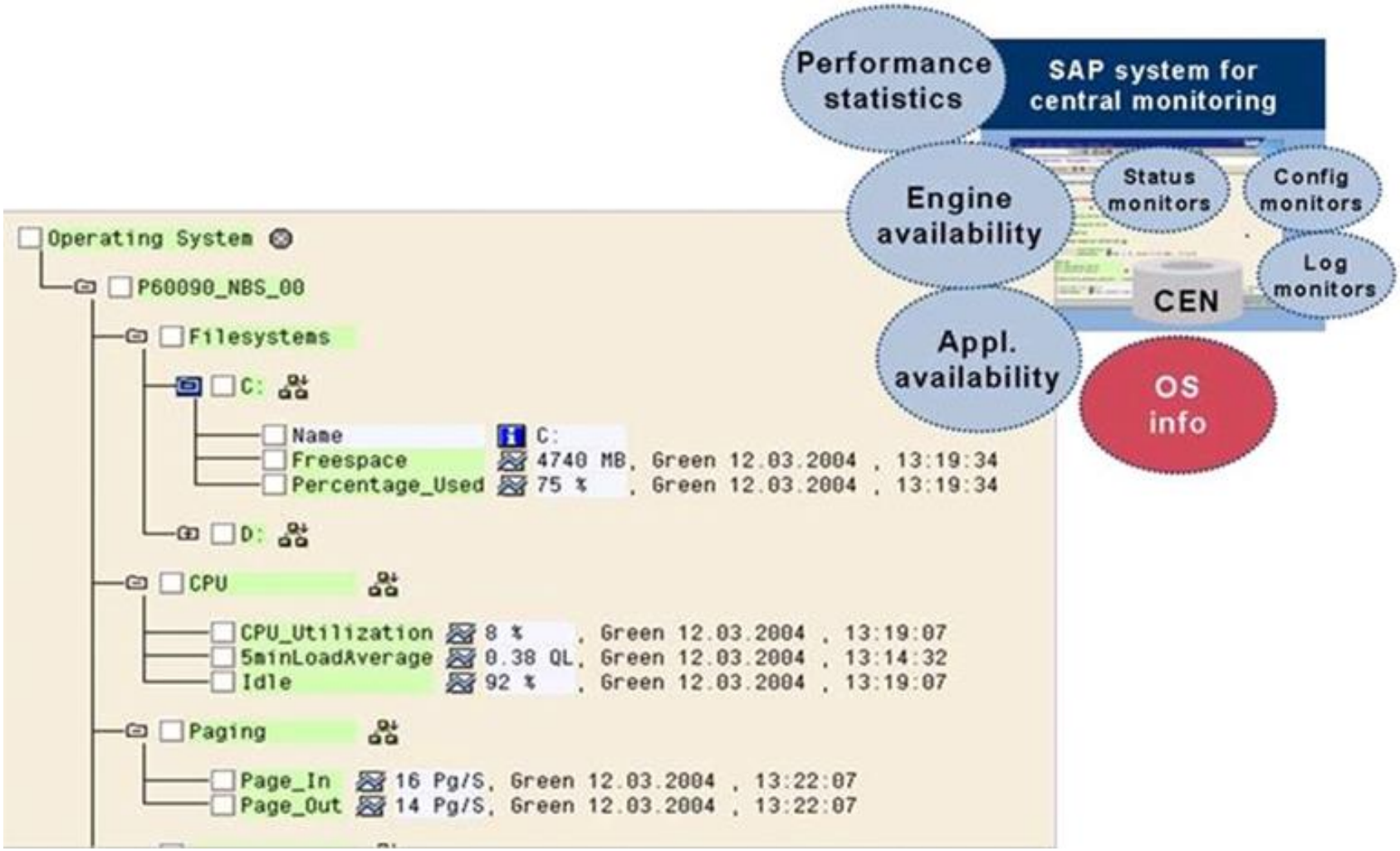


All monitoring configuration data is persistent and is stored in the Java database.

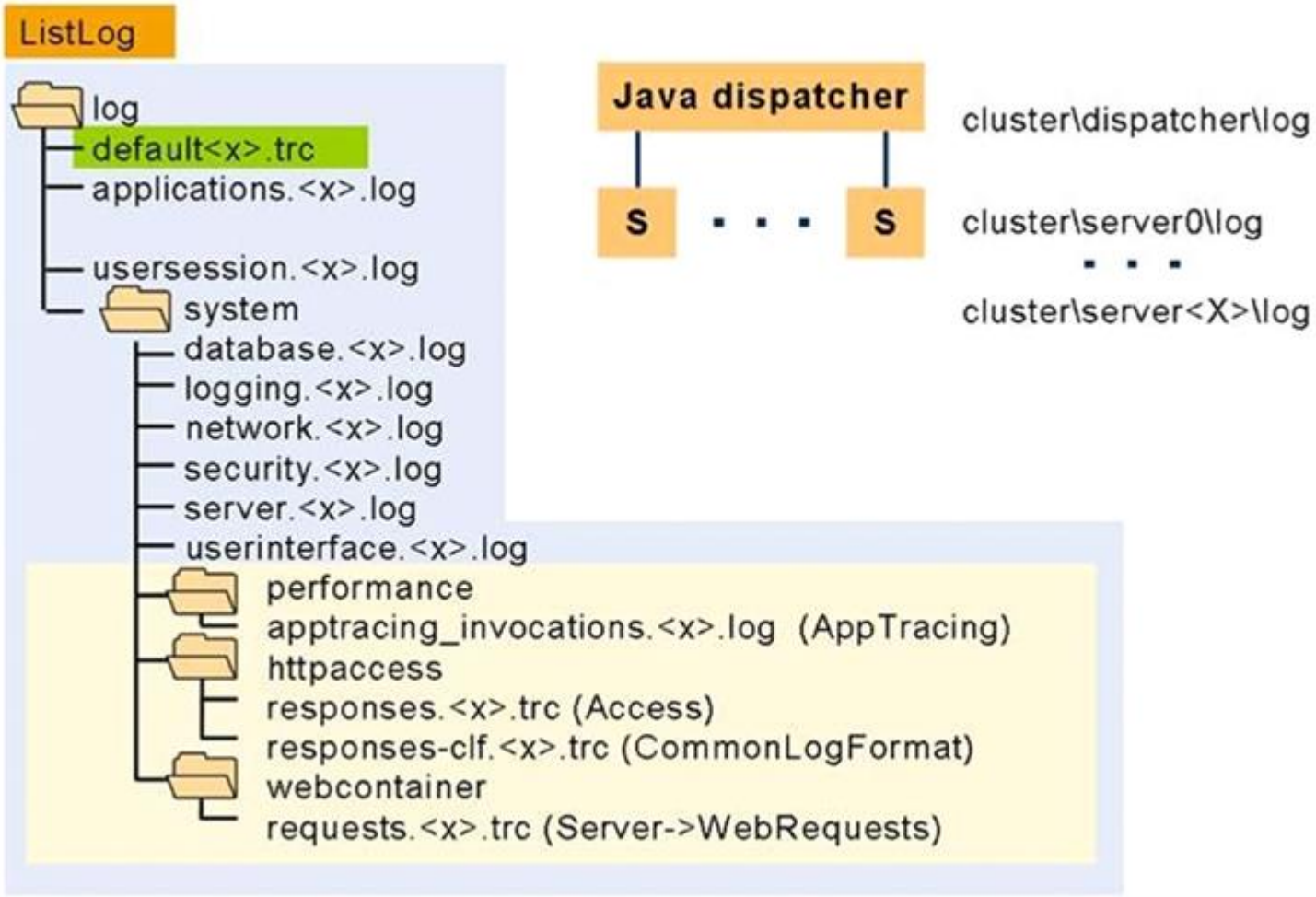
Display in Transaction RZ20



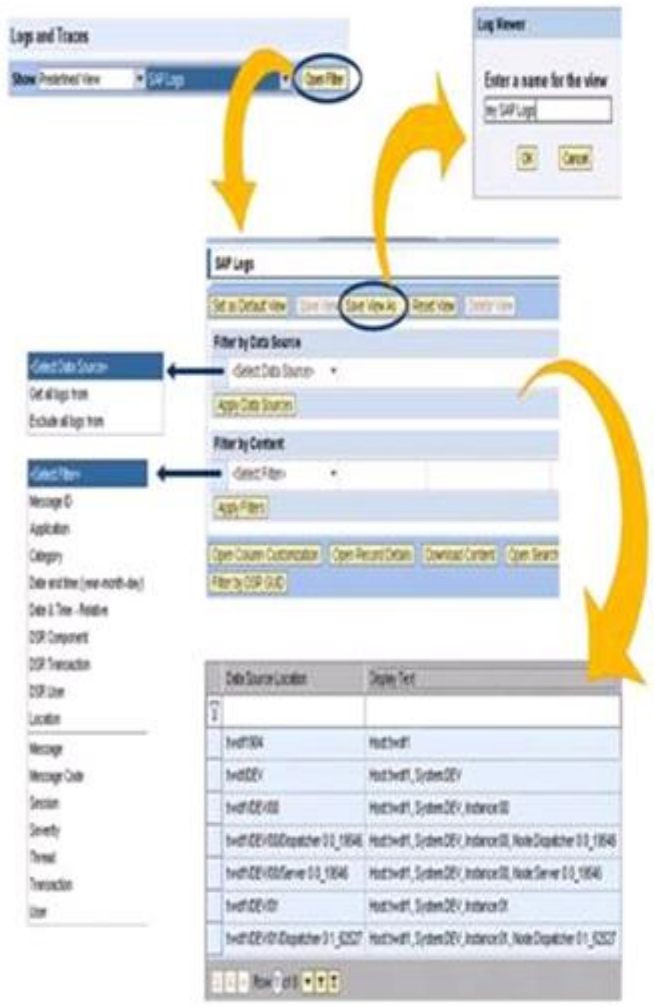
Operating System Information in Transaction RZ20



List Logs in the File System



Log Viewer in the NWA - Filters



Predefined Views

Last 24 Hours
Shows log and trace entries for the last 24 hours

SAP Logs
Shows log entries but no trace entries

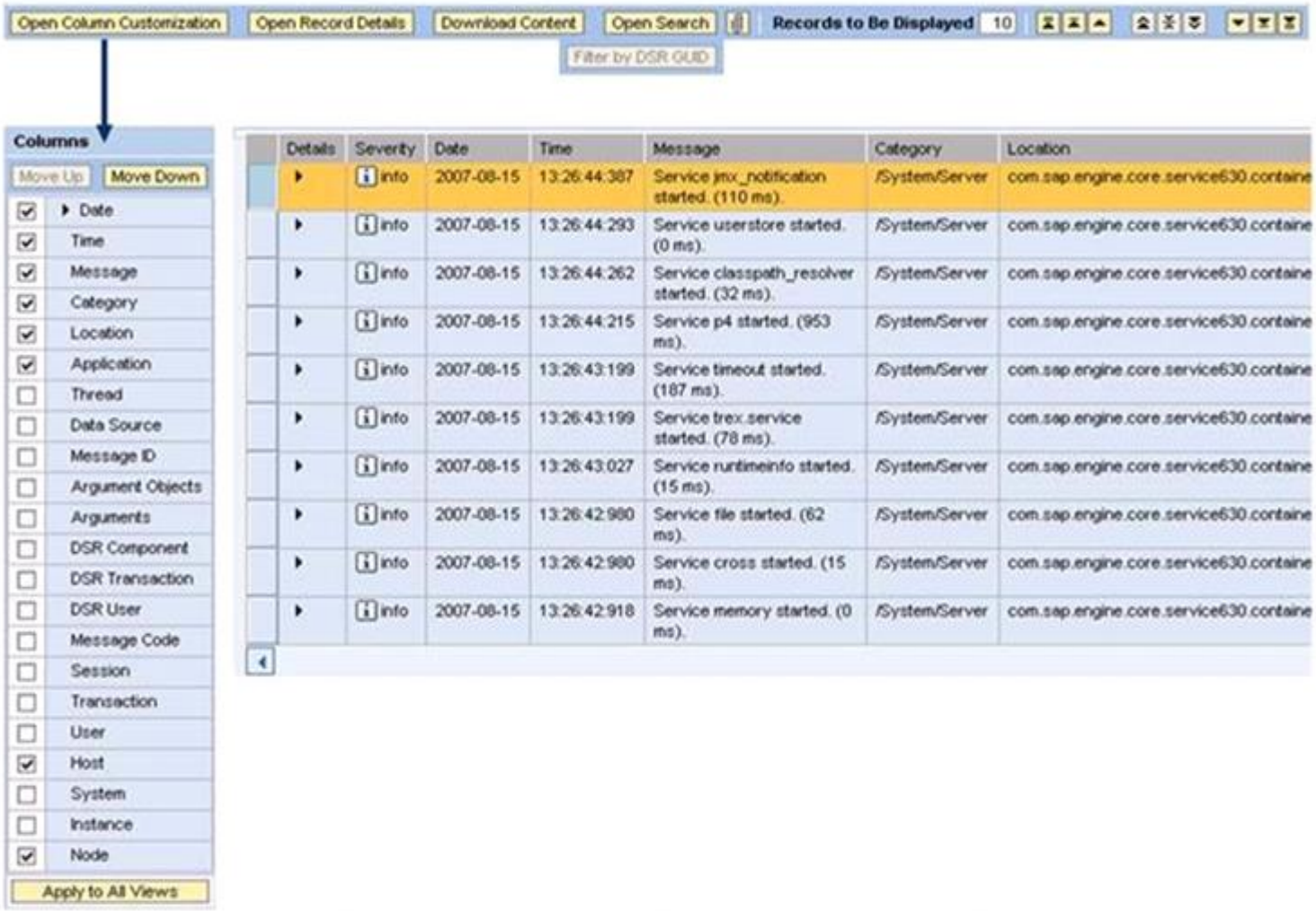
Alert
Shows log entries with severity level “Error” or “Fatal”

Default Trace
Shows trace entries but no log entries

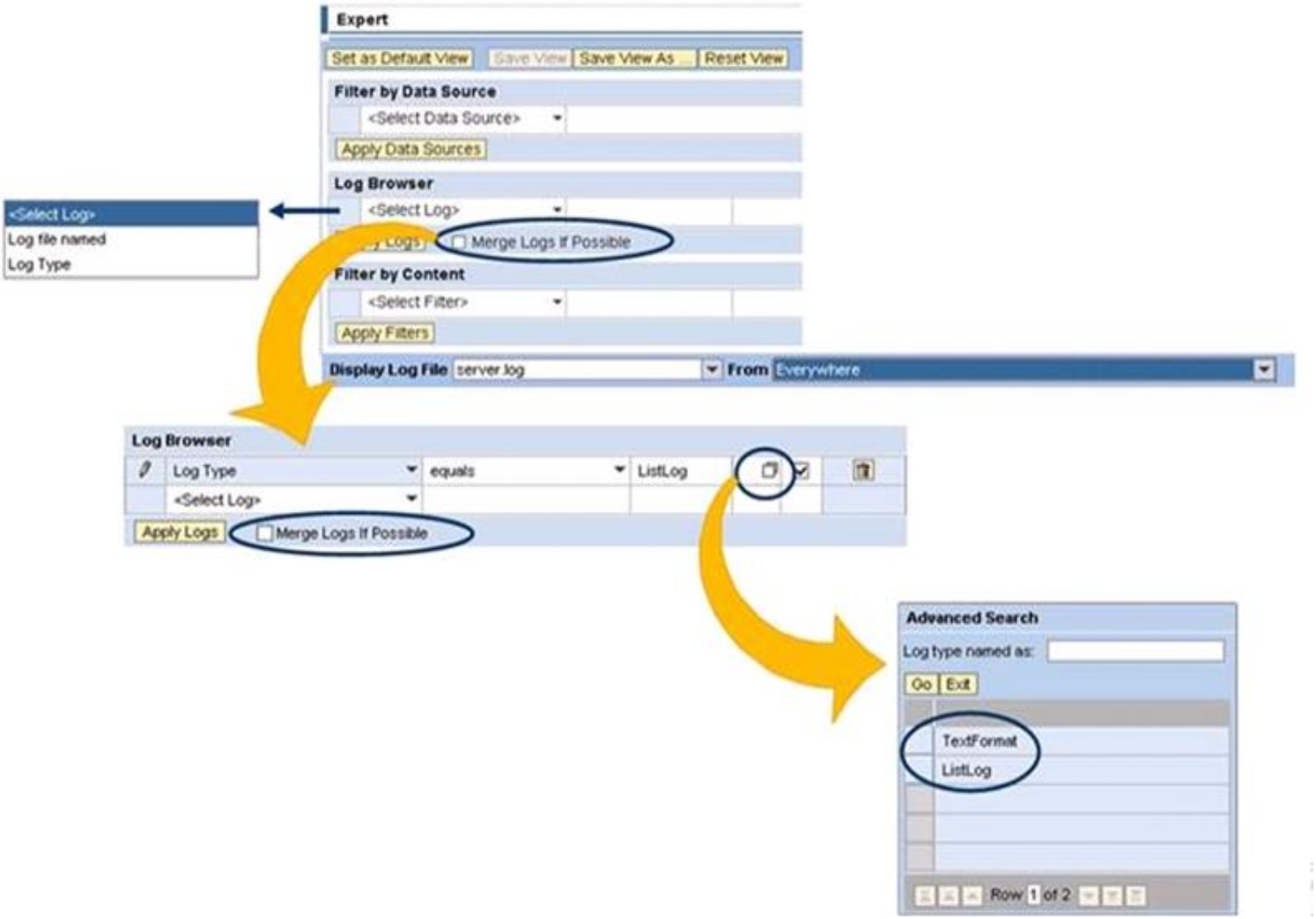
Expert
Shows all log and trace entries without restriction

Text Formatted Traces
Shows file contents which are not of type “ListLog”

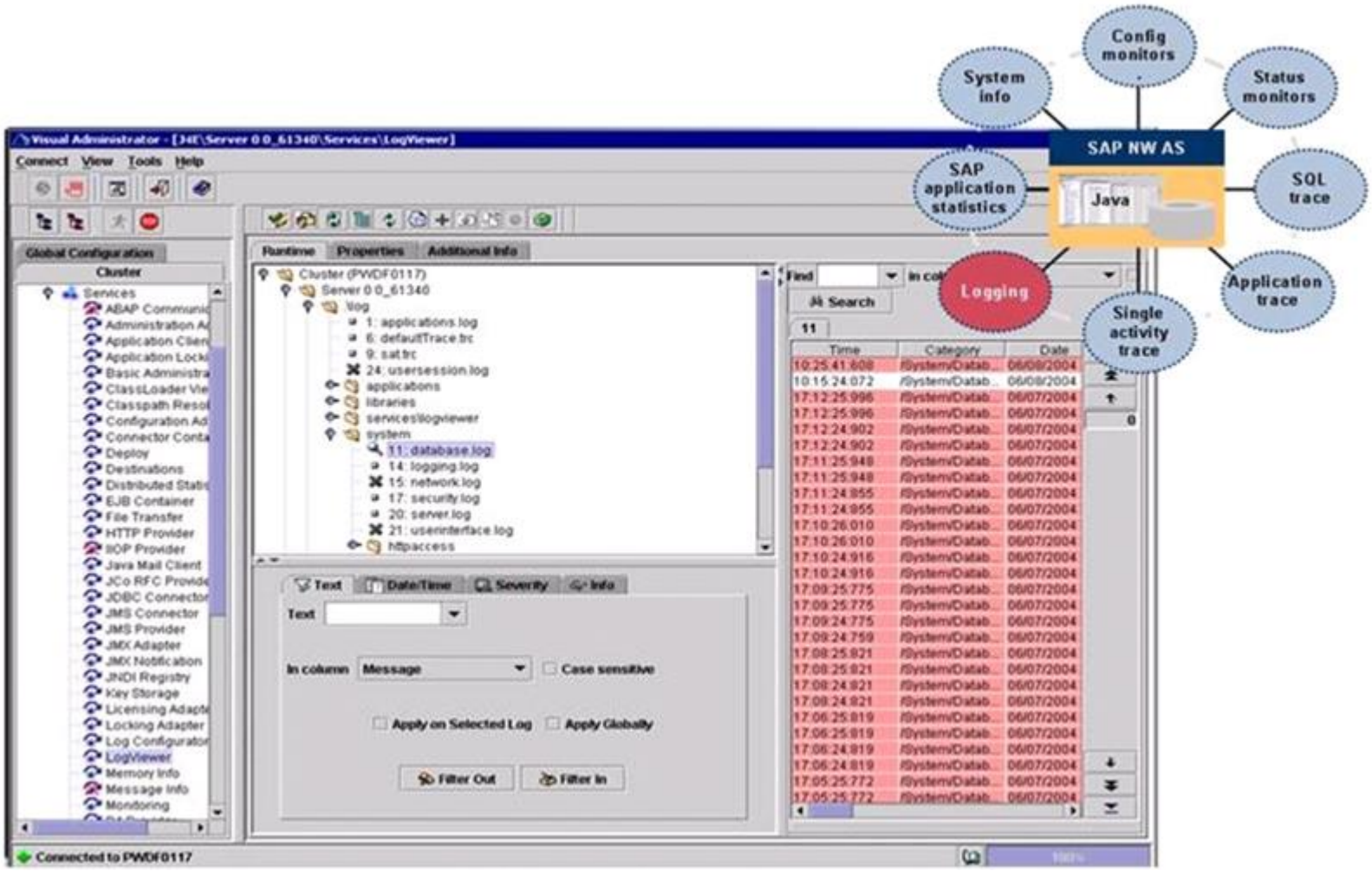
Log Viewer in the NWA: Column Customization



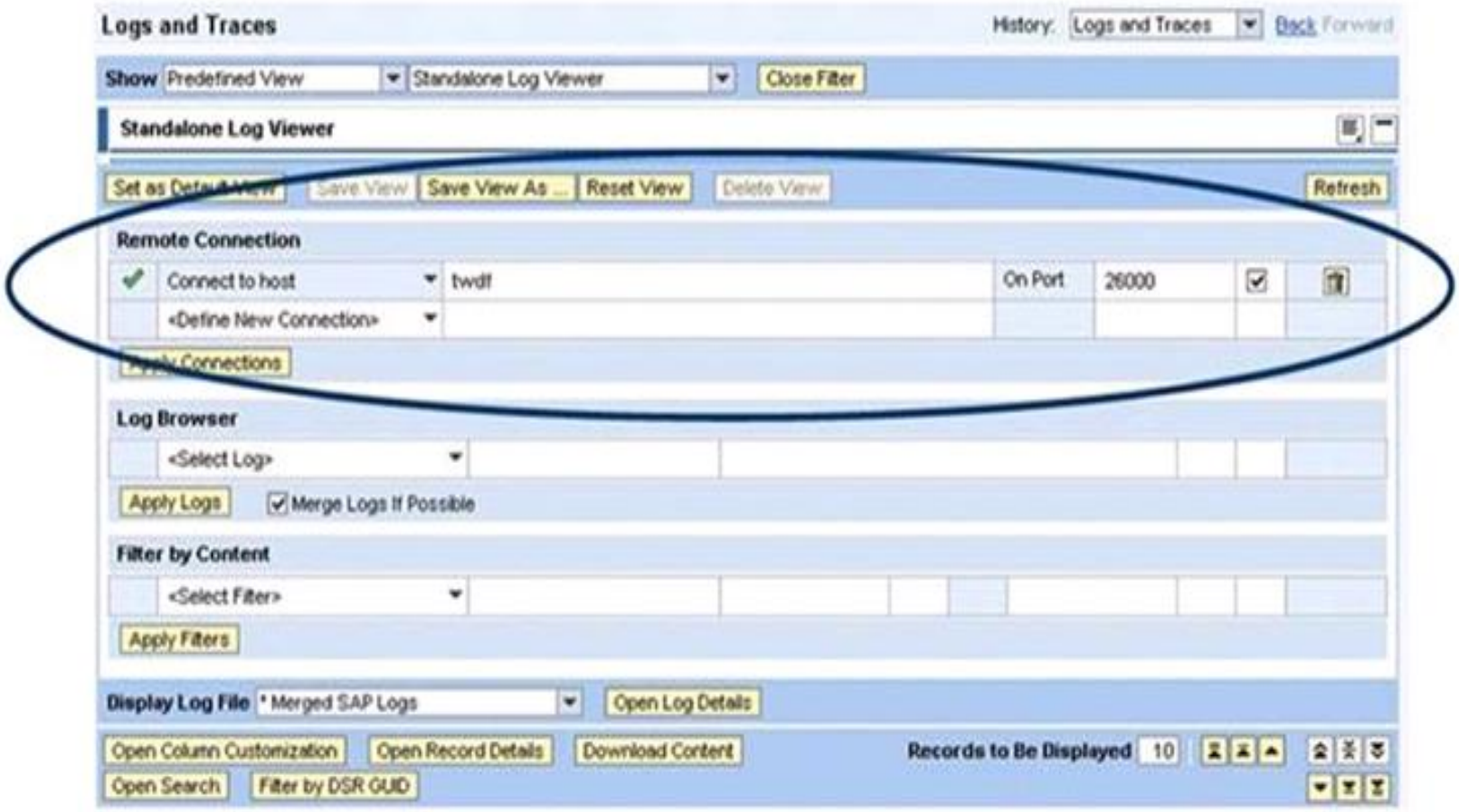
Log Viewer in the NWA: Expert View



Integrated Log Viewer



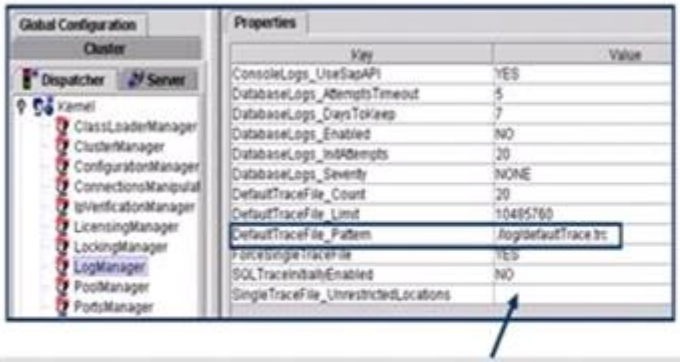
Log Viewer in the NWA: Standalone



Default Trace

All traces are written in one file:

usr\sap\<SID>\<instance>\j2ee\cluster\server0\log\defaultTrace.trc



Specify a location for a separate trace file (location name → Log Configurator Service)

You must update and restart the server

Functions of the Log Configuration

Service:

Change the severity

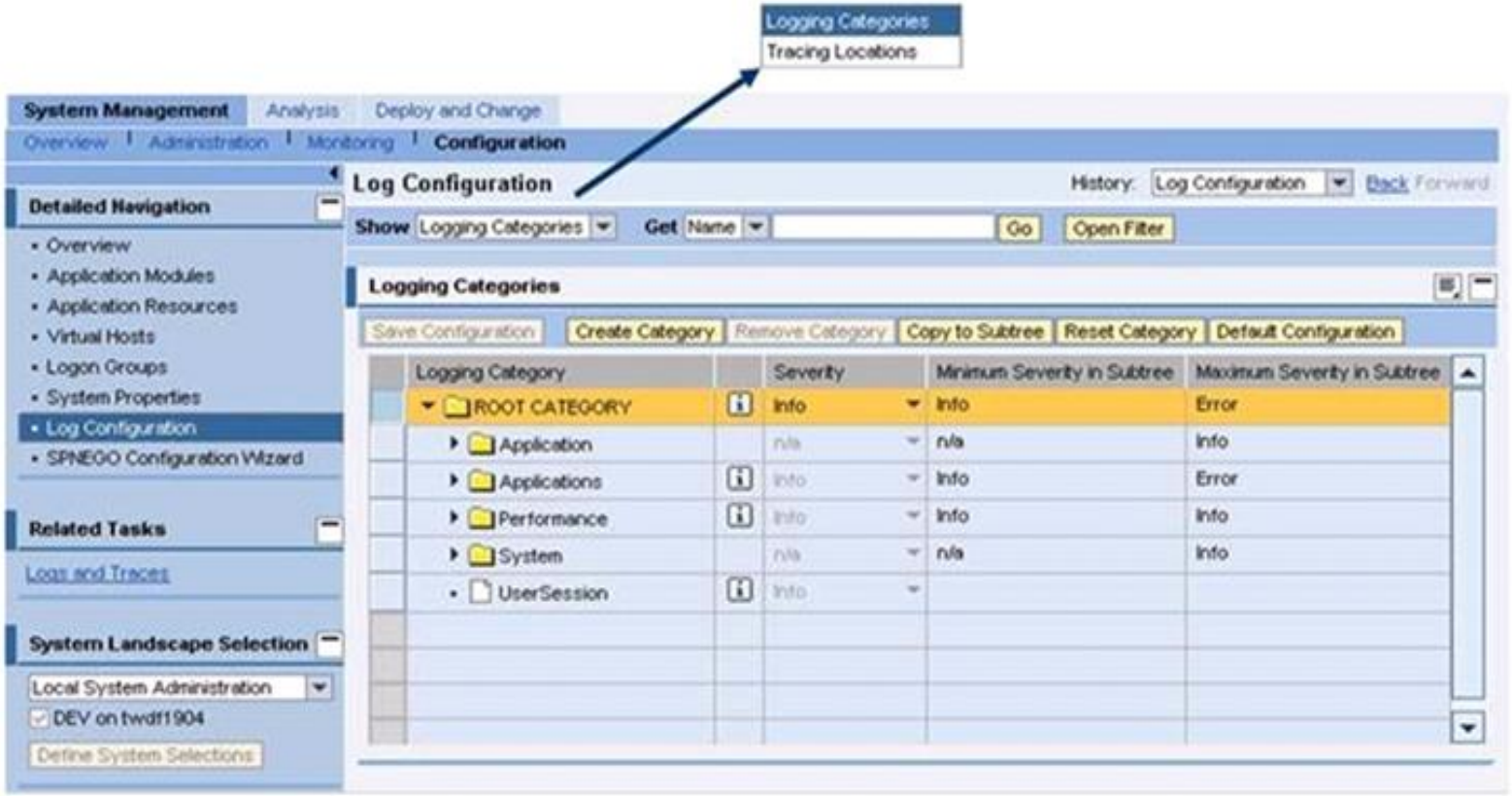
Add, change, and delete log destinations (storage locations)

Add, change, and delete log formatters

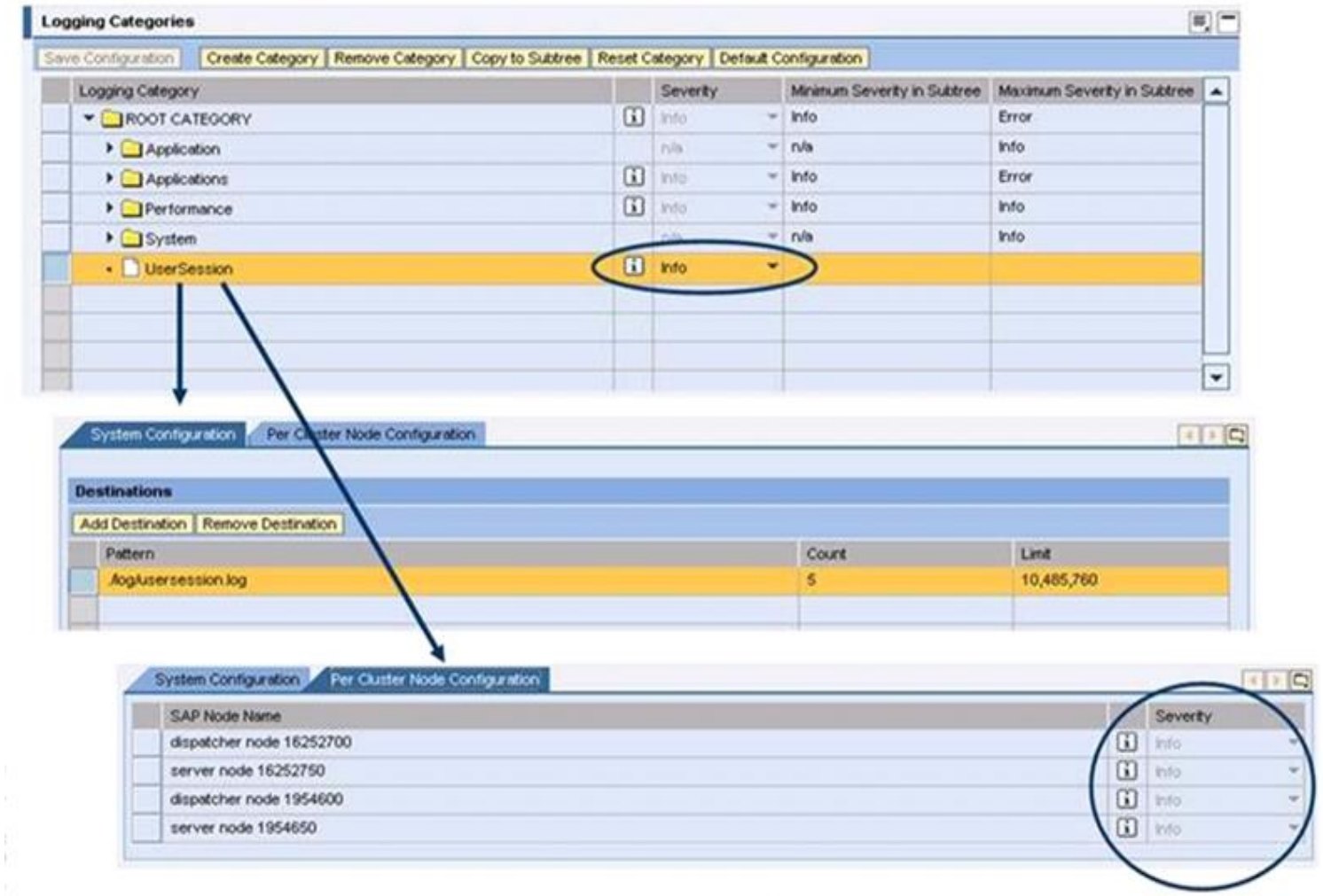
Add, change, and delete log controllers

Archive log files

Log and Trace Configuration in the NWA



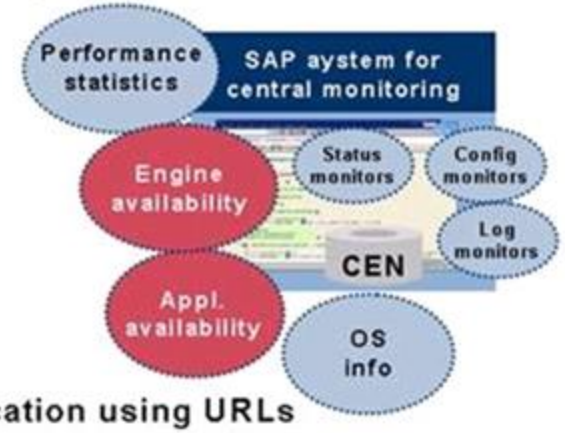
Severities in the NWA



Availability Monitoring

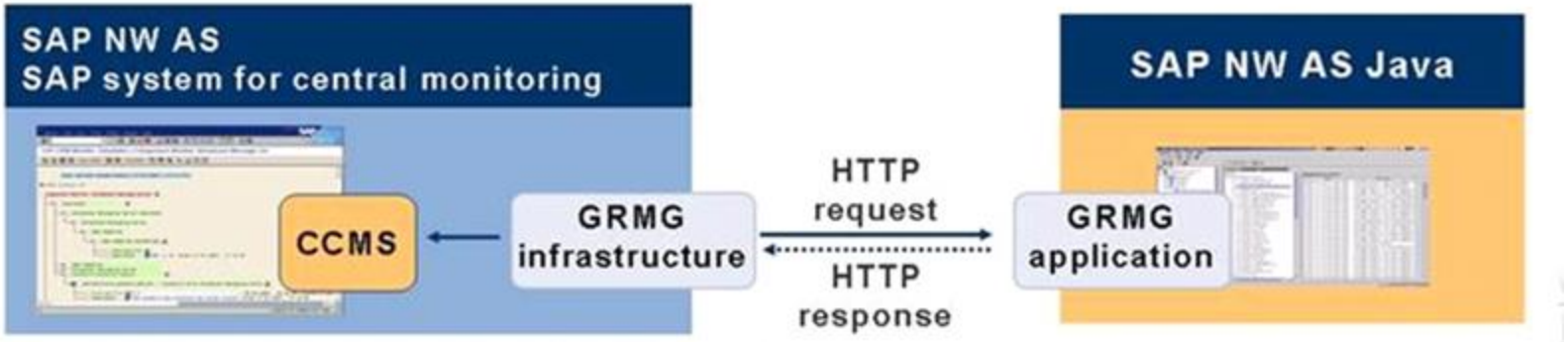
GRMG: Generic Request and Message Generator

Central infrastructure for availability monitoring of Java-based components and applications

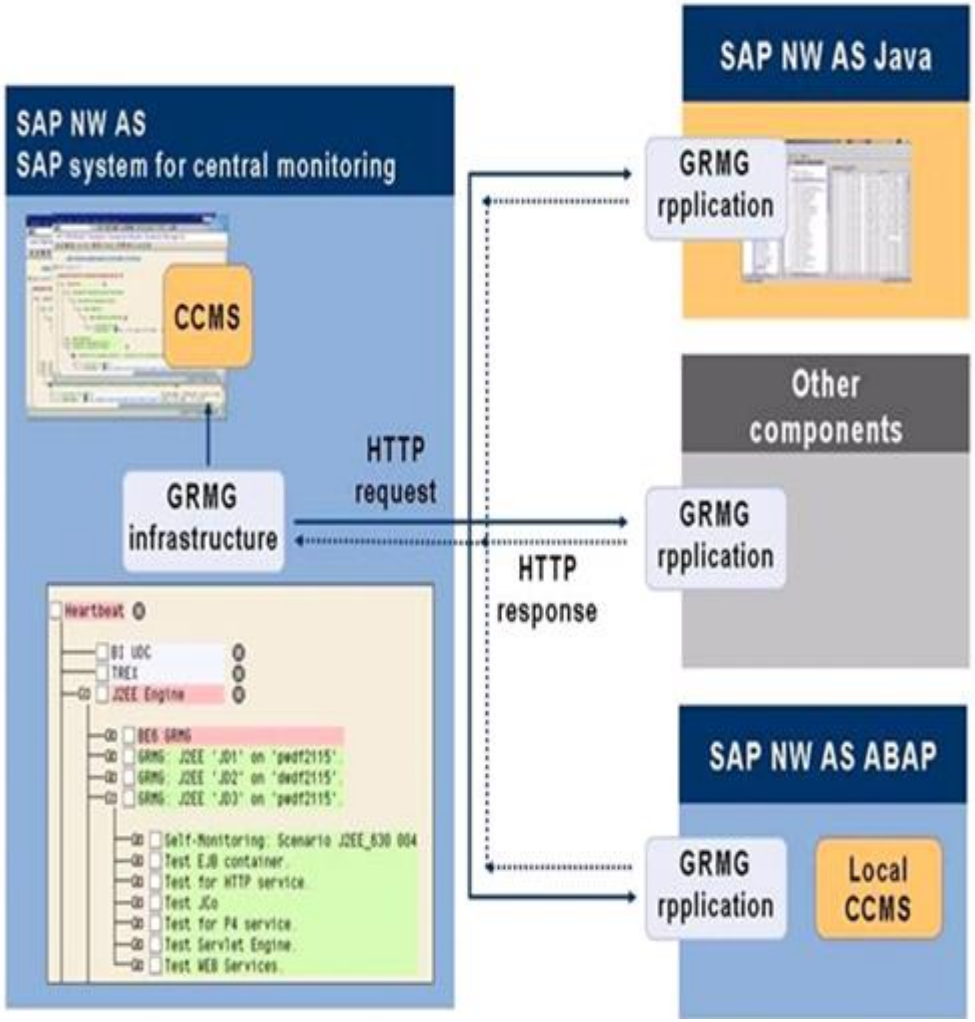


Functionality:

- GRMG infrastructure periodically calls GRMG application using URLs
- GRMG request (XML) is sent to the GRMG application to check availability
- GRMG application returns a response (XML)



Availability Monitoring with the GRMG



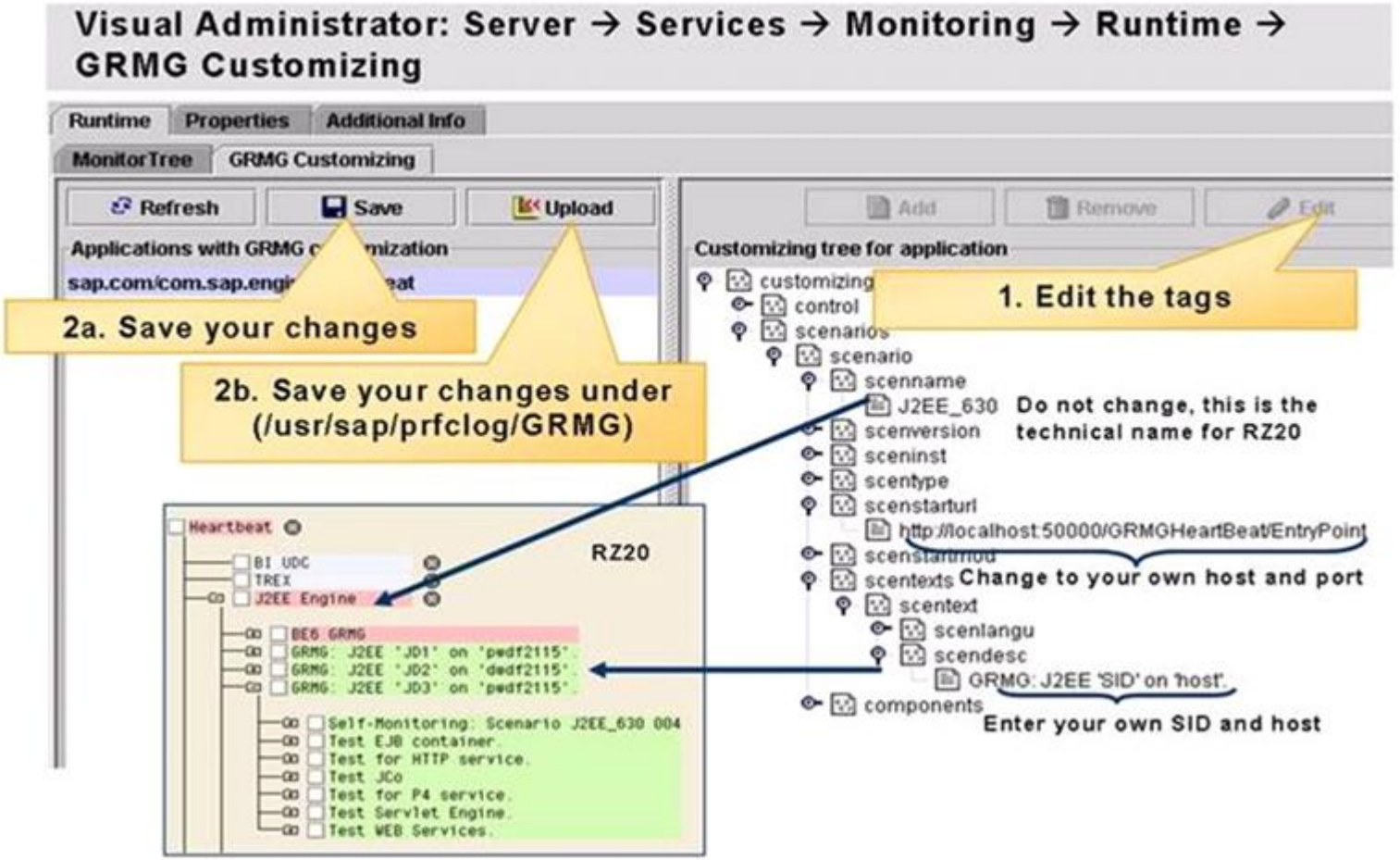
Setting Up GRMG Monitoring:

Instrument the application for GRMG monitoring

Perform the technical Customizing for monitoring

GRMG applications

Editing Customizing.xml (Step 1)

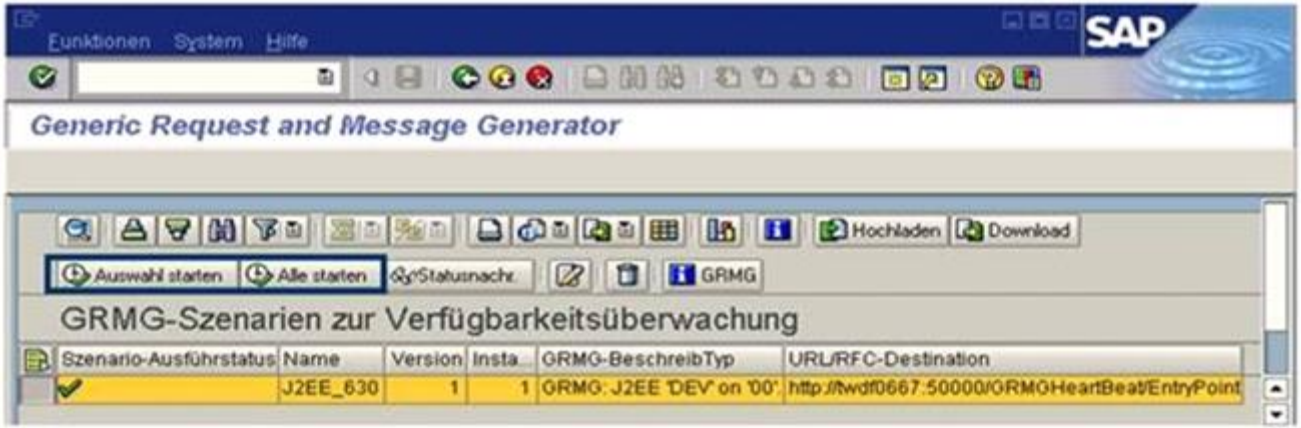


Upload GRMG Customizing File (Step 2)

Automatic Upload Using an Agent

Manual Upload

Starting the GRMG Scenarios (Step 3)



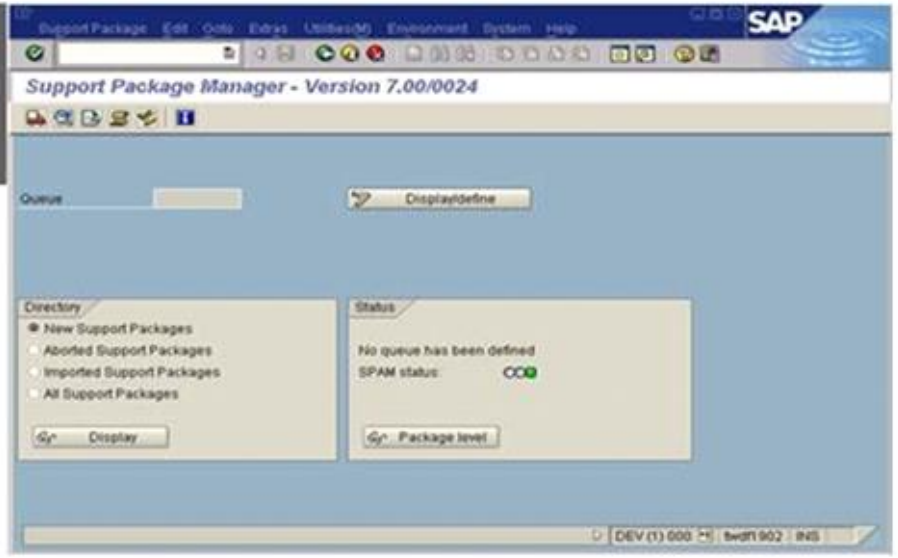
Availability (GRMG): Display in RZ20

RZ20 → SAP J2EE Monitor Templates → Heartbeat → J2EE Engine

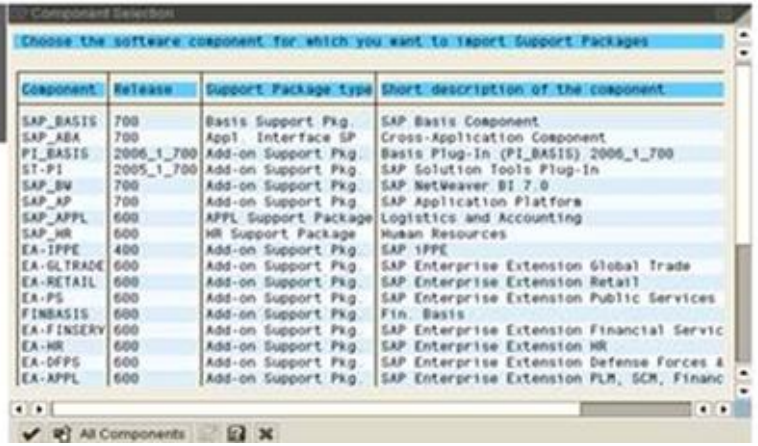
Break

ABAP Support Packages Using Transaction SPAM

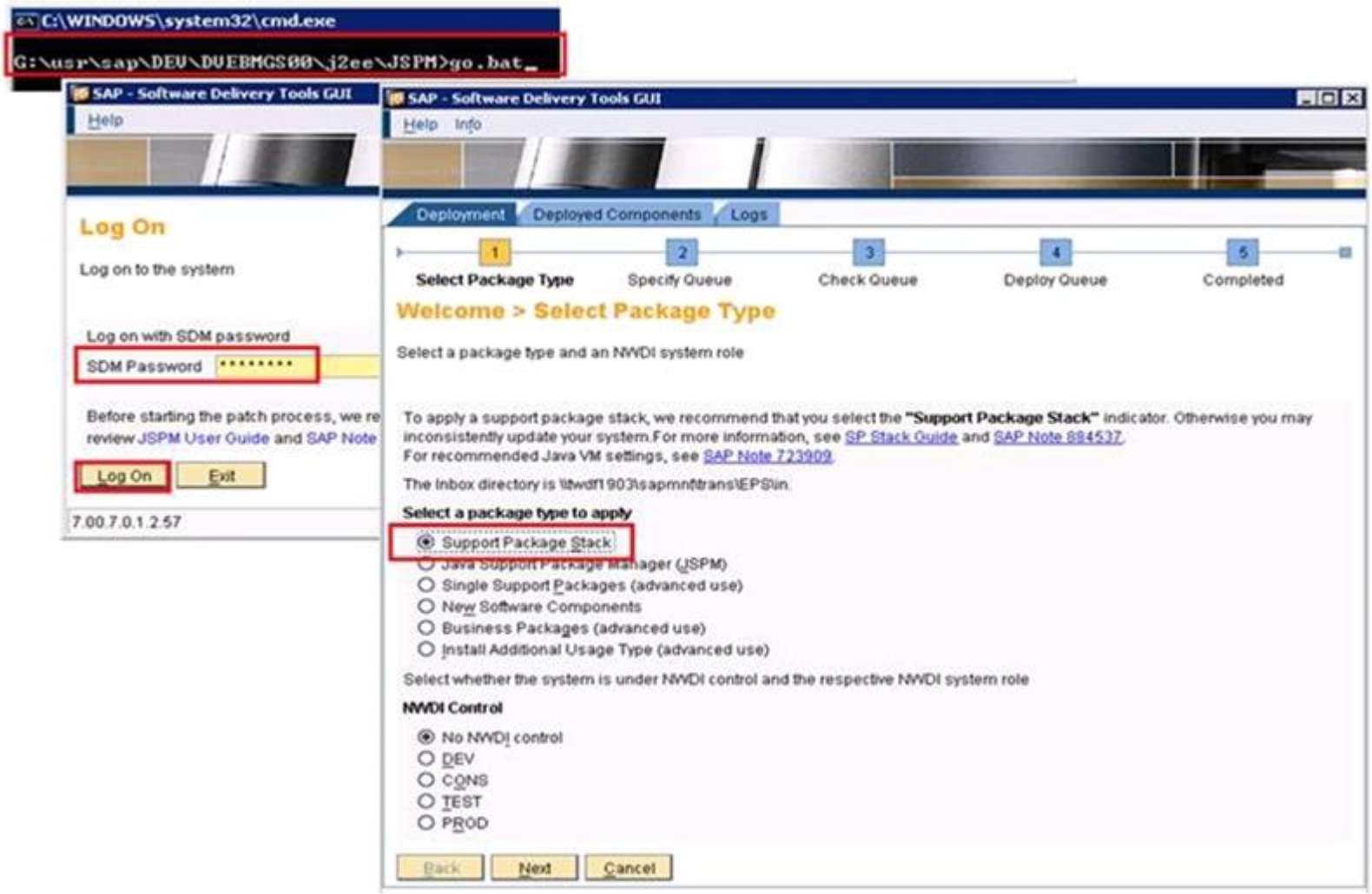
Transaction SPAM:
- Import SPAM/SAINT Update
- Define Support Package Queue
- Import Support Packages



ABAP-Components of SAP ECC, which can be updated using SPAM.

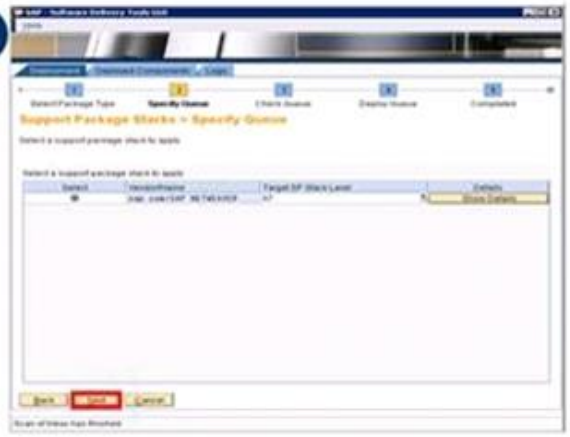


JSPM: Java Support Package 1/2



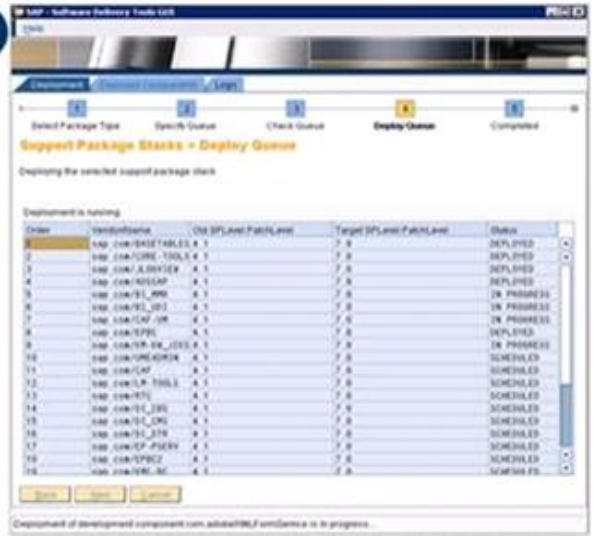
JSPM: Java Support Package 2/2

1

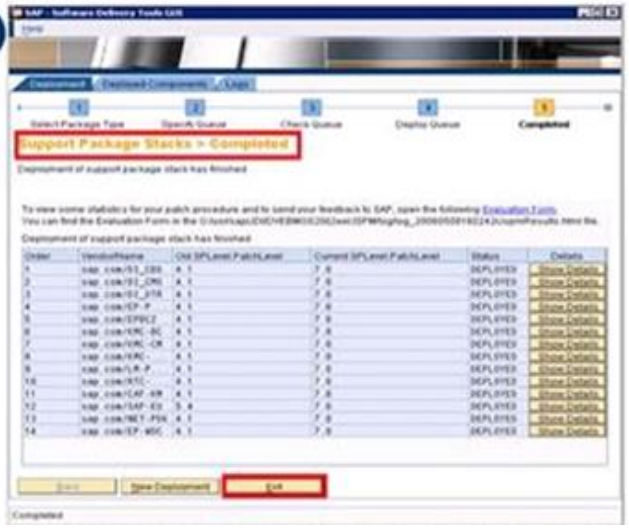


2

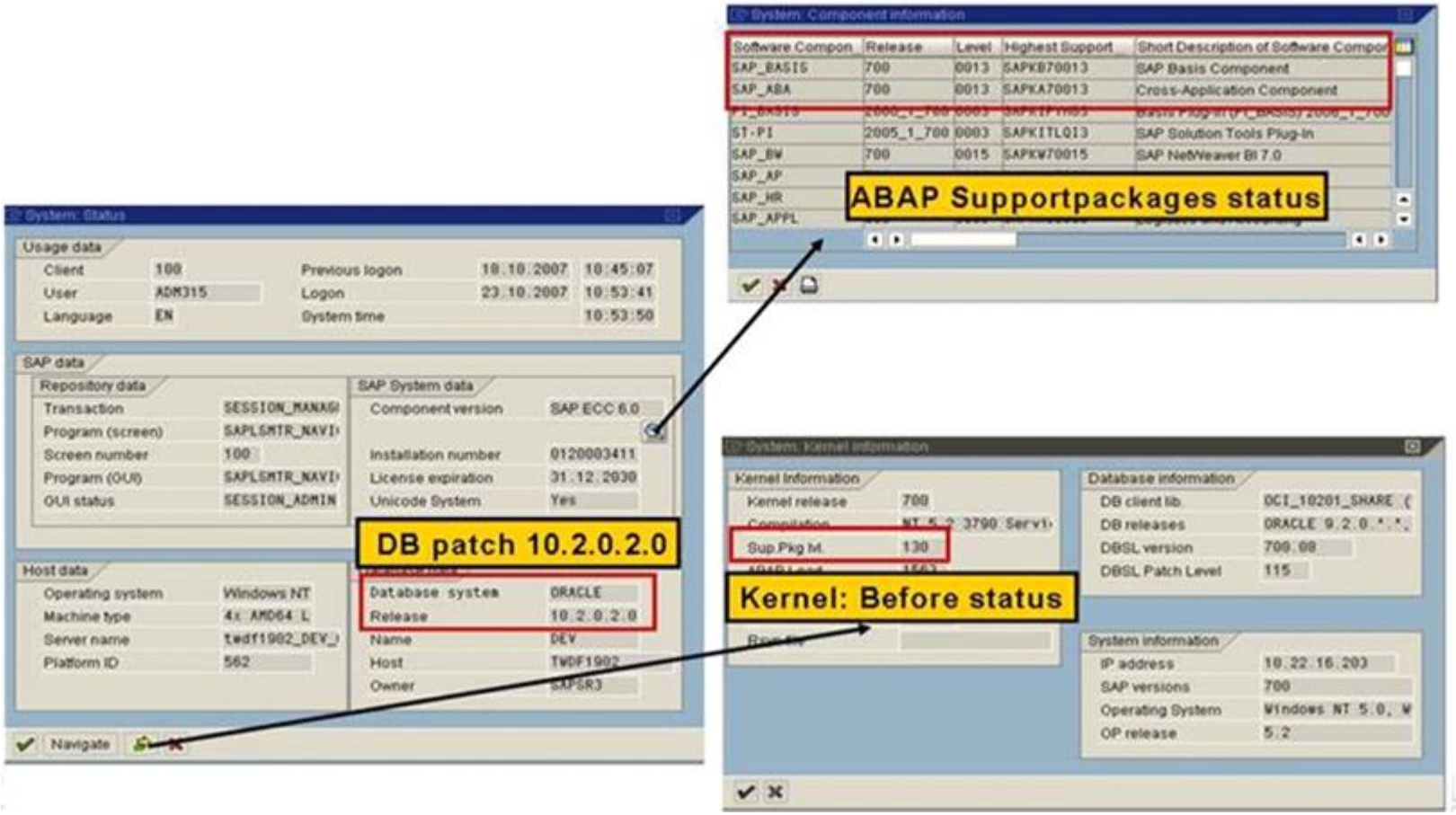
3



4



Check System Patch Details



Kernel Update

New Directory Structure for Kernel

Central Instance:

new:*sapglobalhostsapmnt**SAPSID**sys**exe**codepage**platform***

(*codepage*:UC = Unicode systems, NUC = non-Unicode systems)

old: *sapglobalhostsapmnt**SAPSID**sys**exe**run***

Java Add-in instances:

new:*sapglobalhostsapmnt**SAPSID**sys**exe**UC***

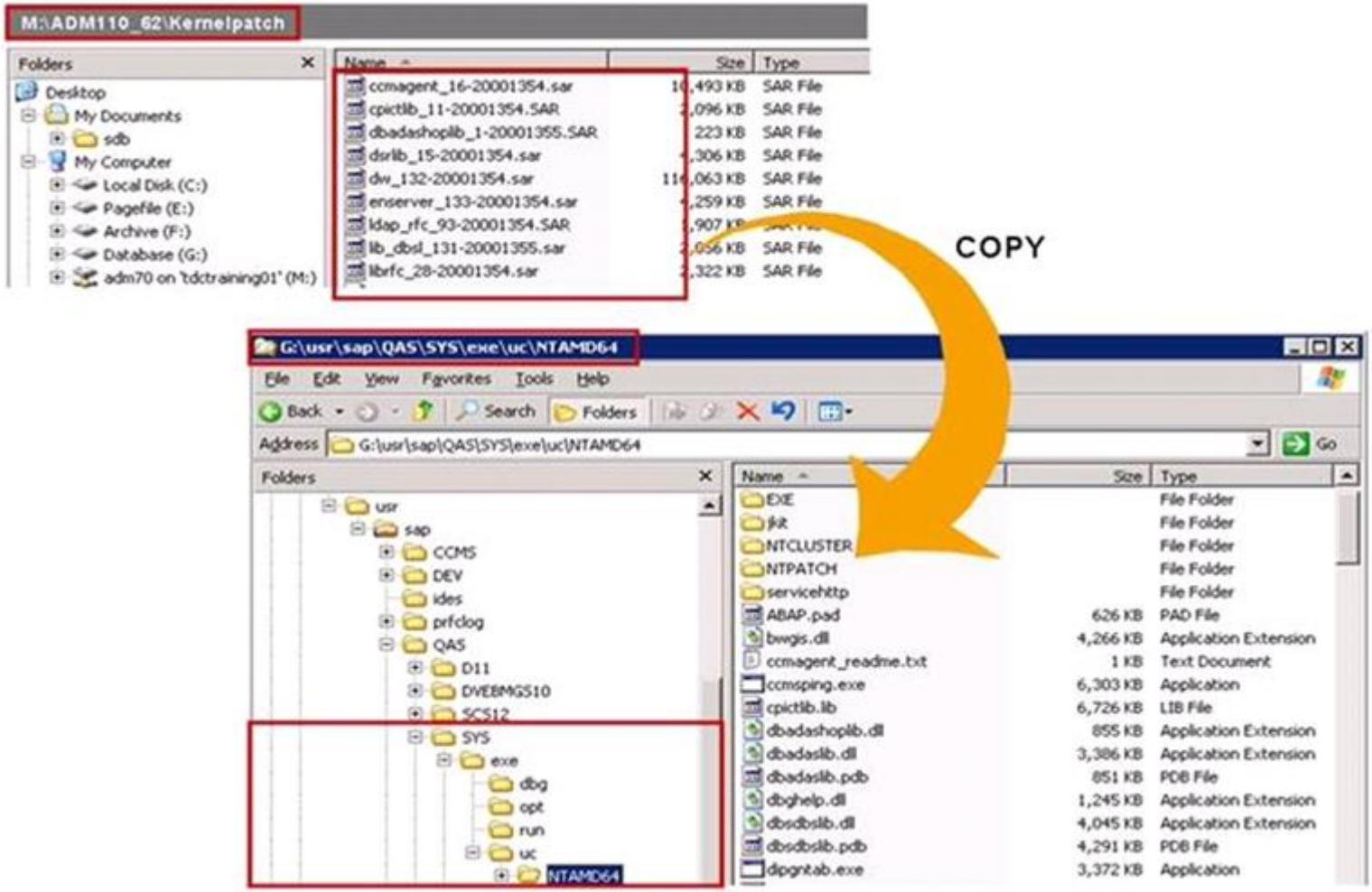
old:*sapglobalhostsapmnt**SAPSID**sys**exe**runU***

Central instance uses an instance-specific executable directory (*sapglobalhostsaploc**SAPSID**DVEBMGS##**exe*)**

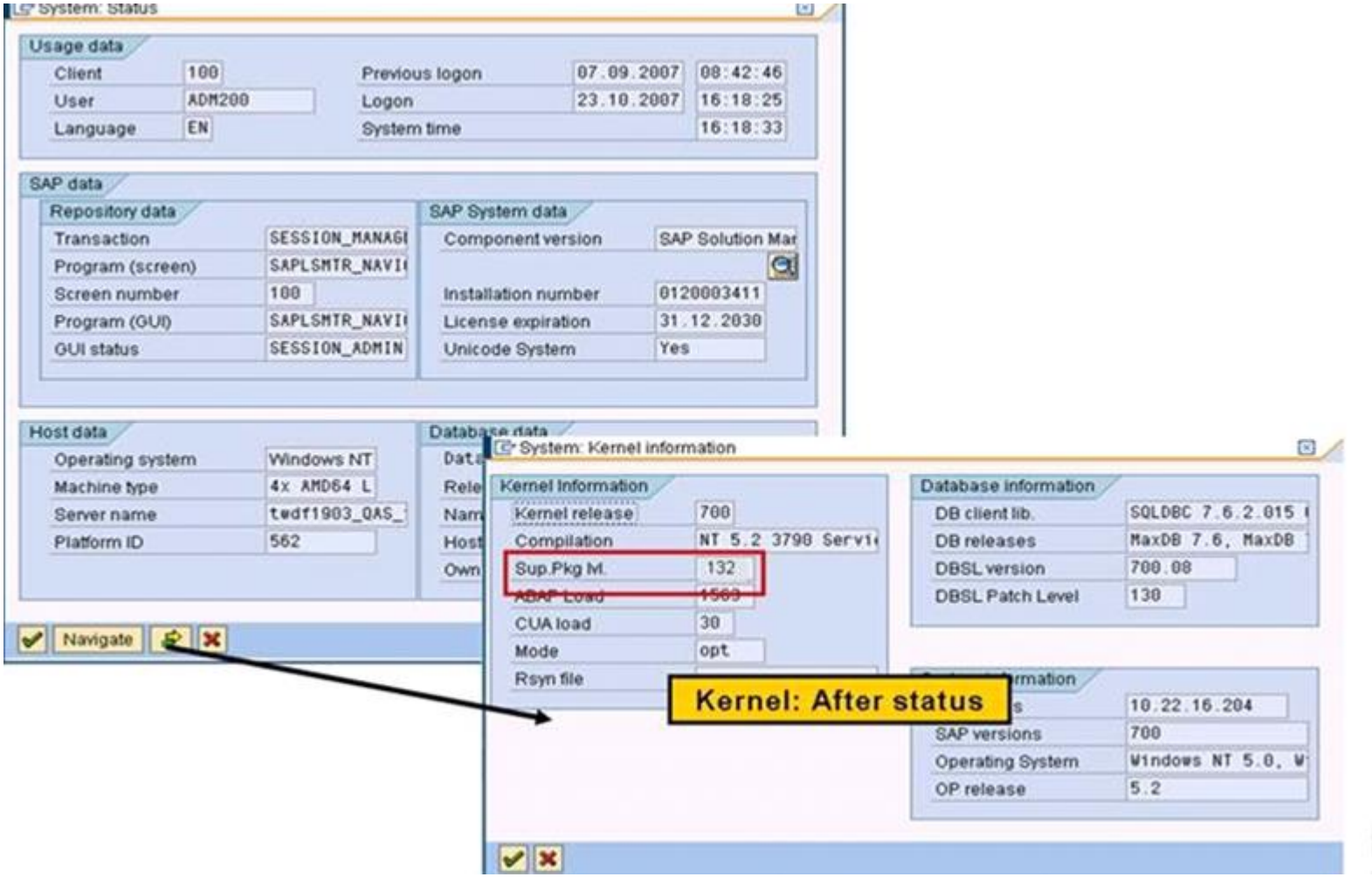
An instance should not directly use

sapglobalhost**sapmnt**SAPSID**SYS**exe**run

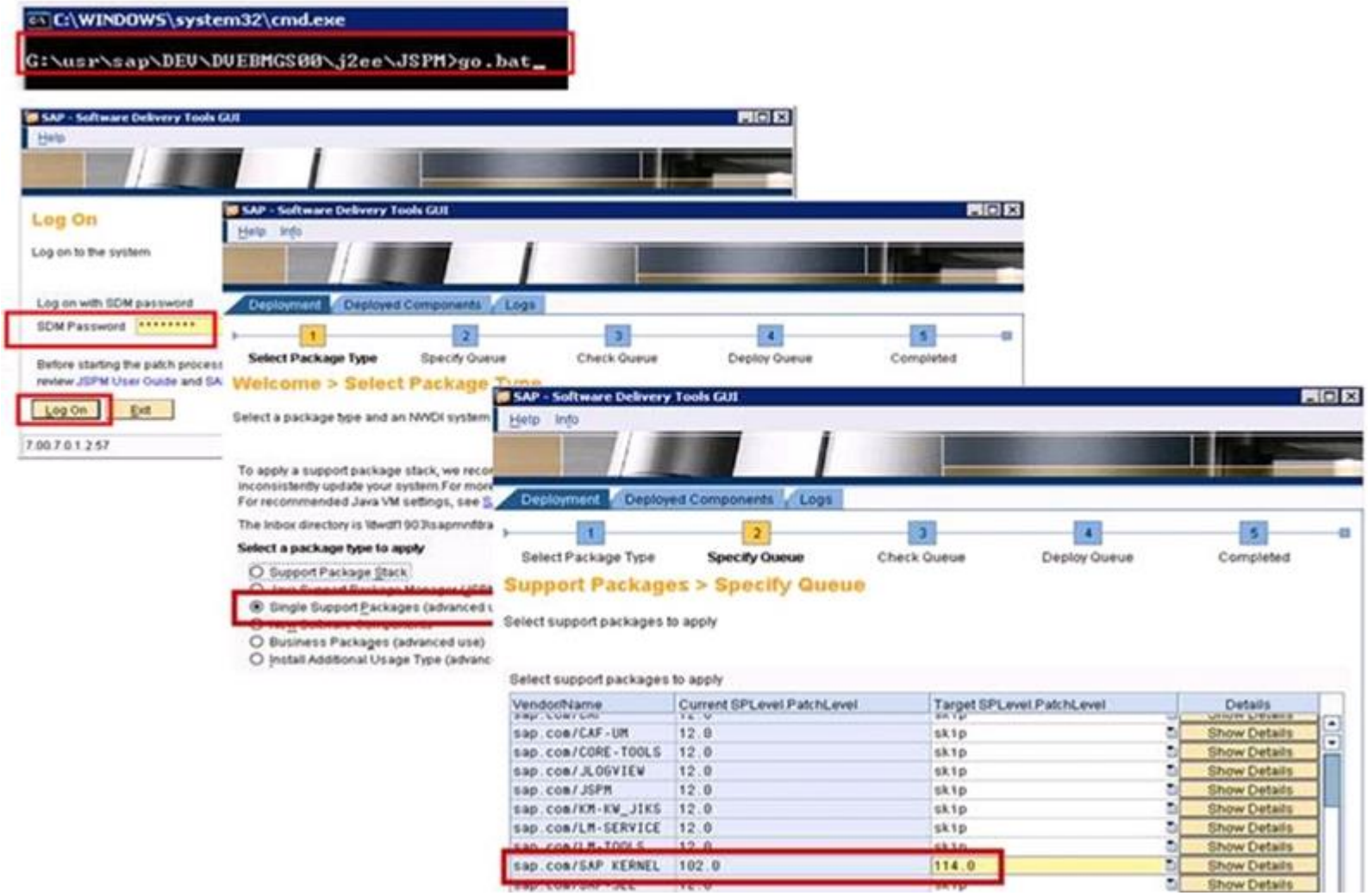
Applying a Kernel Patch Manually



Kernel Patch Level After the Patch



Apply a Kernel Patch using JSPM 1/2



Q&A Breakout Session


COLLABORATIVE BUSINESS EXPERIENCE