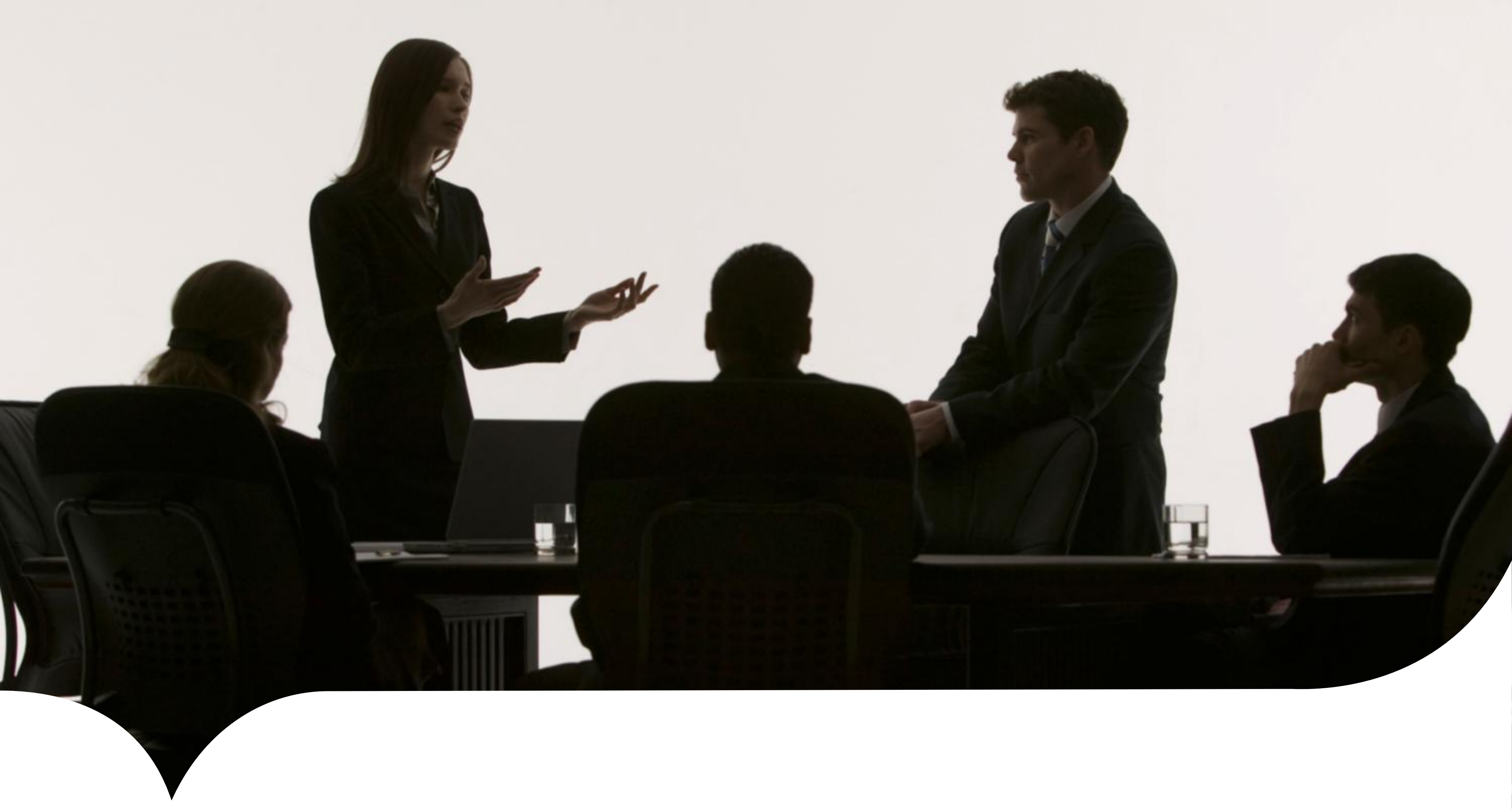
SAP BASIS Introductory Training Program

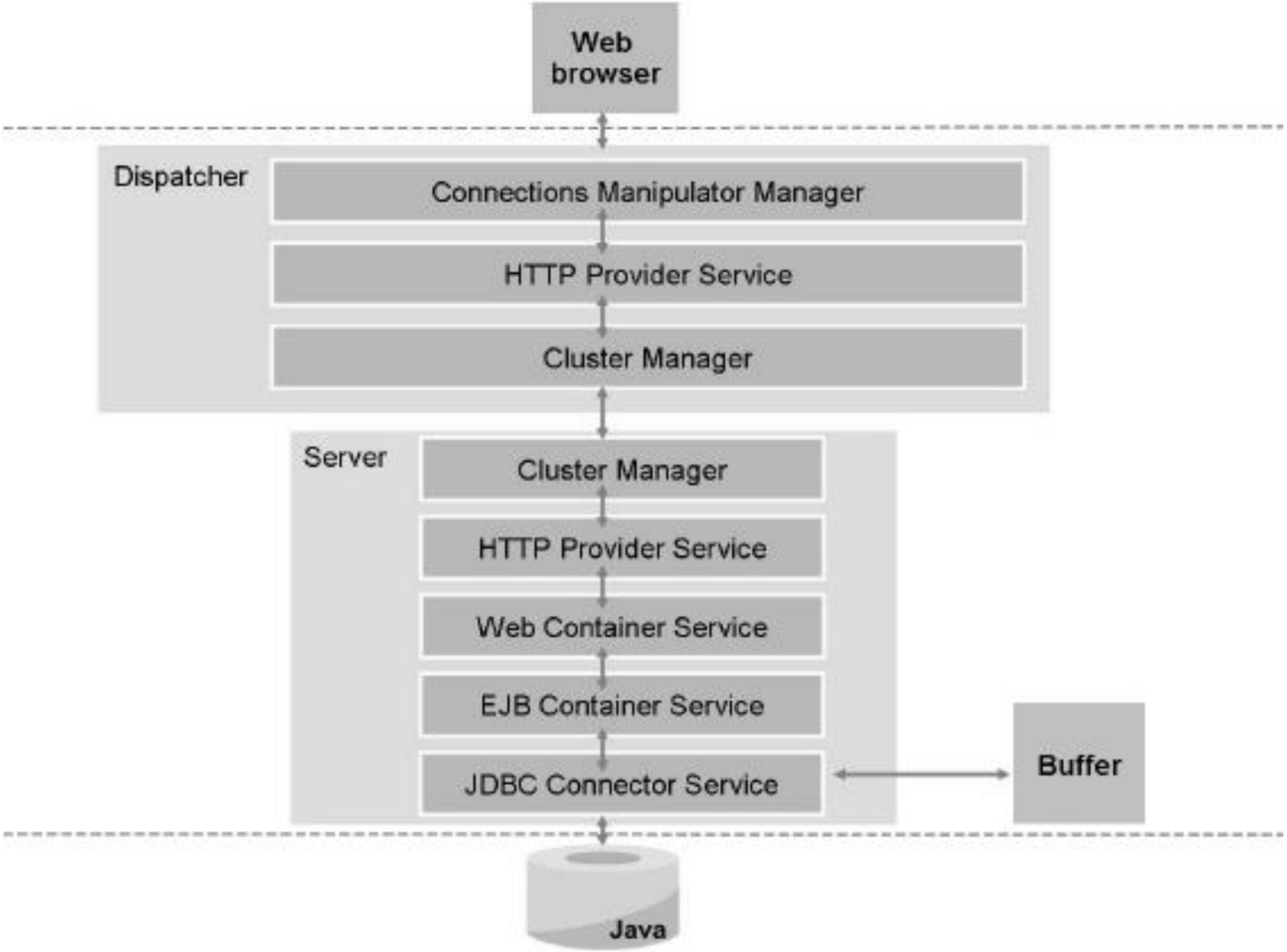
Day 5 : Agenda

- Transaction Processing in AS JAVA
- Break
- Startup/Shutdowm in AS JAVA
- Lunch Break
- System Administration in AS JAVA
- Break
- UME Concepts
- Q&A and Break Out Session



Transaction Processing in AS JAVA

Manager & Services Concept



Details on Managers & Services

1. The nodes of AS Java are split into different functional modules called managers and services. The managers form the Java Enterprise Runtime. The Java Enterprise Runtime provides basic core functions of AS Java. It is also referred to as kernel. Together with the interfaces and libraries, the services are called J2EE Engine

2. Components. The J2EE Engine Components provide programming interfaces (APIs) to the applications; the applications can then use these APIs to access the AS Java functions.

3. In case of an HTTP requests to the Java dispatcher the Connections Manipulator Manager holds a connection object with information about the client sending the request. The request is then forwarded to one of the processes of this instance by the HTTP provider service using a cluster manager.

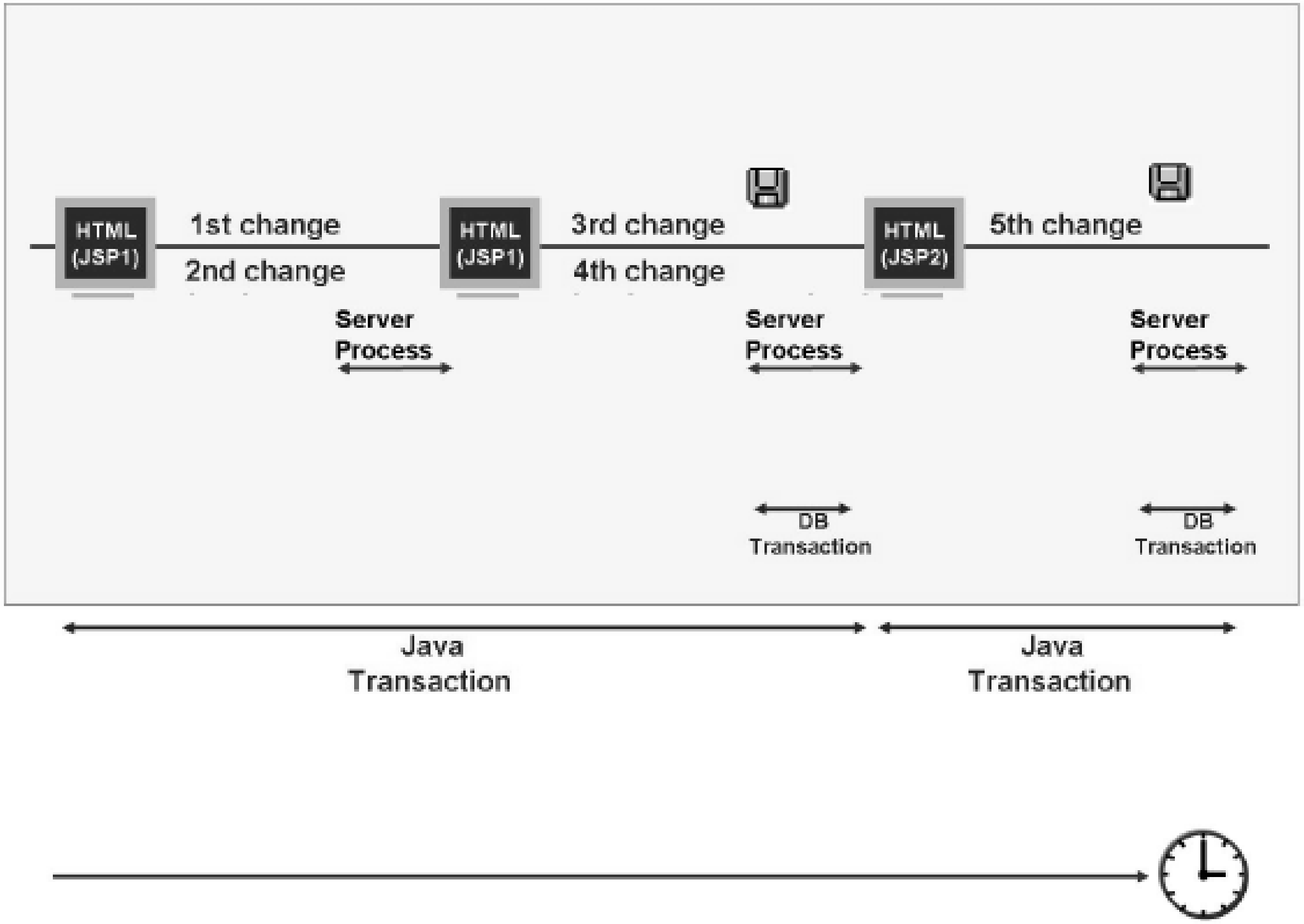
4. The cluster manager of the server process receives the request and forwards it to the HTTP provider service.

5. In Web Container Service, the presentation logic of the application is then processed. The web container service provides the processing of servlets and JavaServer Pages (JSP). The business structure logic of the application is processed in the form of EJB beans in the EJB Container Service. If, in the processing of the request, data from the database is required, the JDBC Connector

6. Service is used to establish a connection to the database and the data is requested there. If the same tables contents have already been queried by this server process, the content can be retrieved in the table buffer at application level (if buffering is allowed for the table).

7. The response to the web browser using HTTP is then returned in the same way.

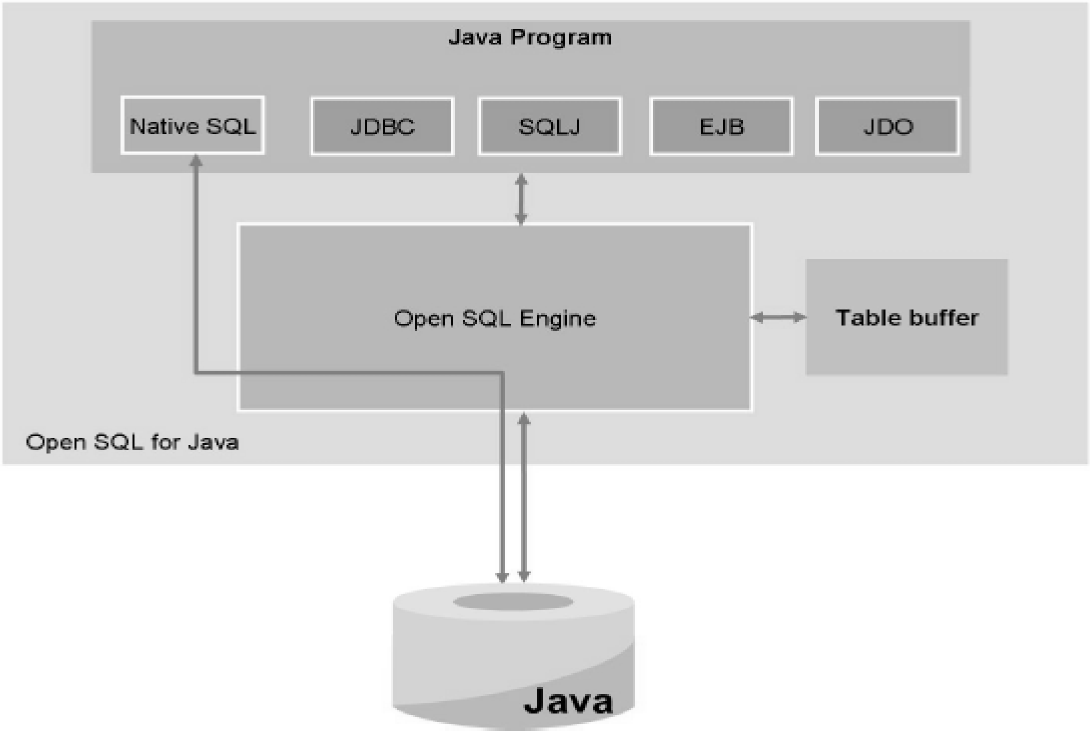
Transaction Processing in AS JAVA



Graphic Illustration of "ACID" transaction processing

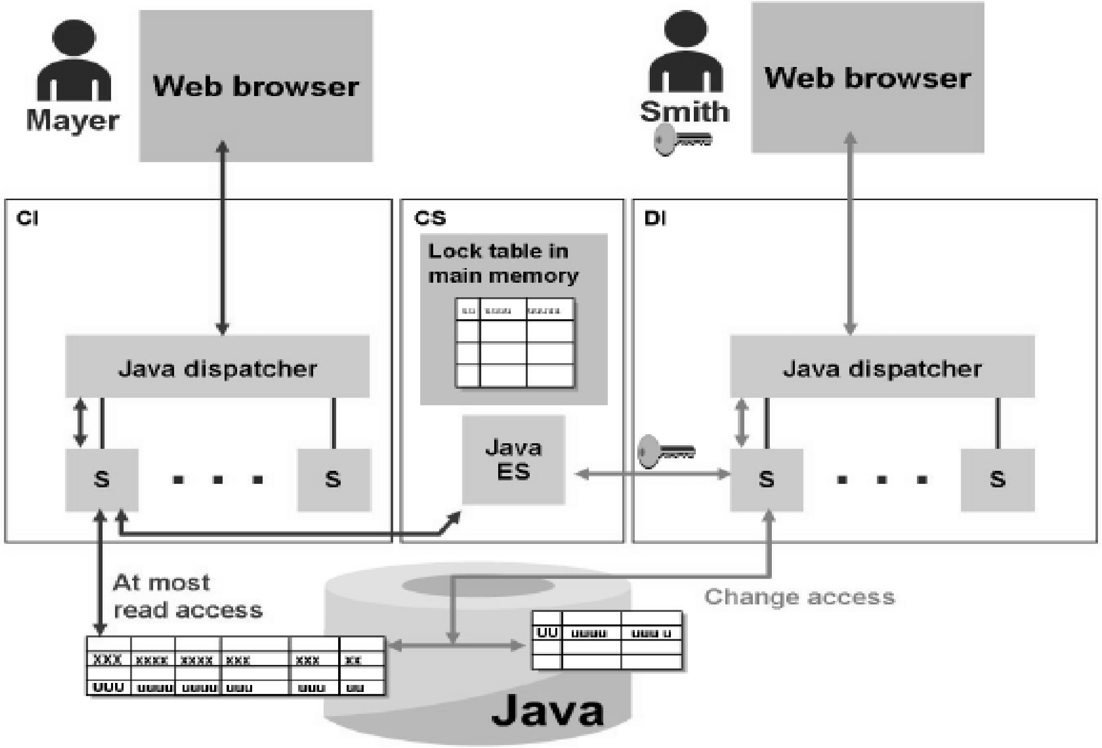
Open SQL for JAVA

- If the Java program is supposed to be portable, that is, run with a database other than the one used originally, developers can choose between Open SQL/JDBC, Open SQL/SQLJ, EJB (Enterprise JavaBeans) and JDO (Java Data Objects). If developers use Native SQL in the program, they loose the portability and cannot use the table buffer of Open SQL for Java Frameworks.



Lock Management

- The lock concept of the database is used in the J2EE standard. So if application developers make sure that they implement database-independent database accesses,the application will be portable but will respond semantically different on different database platforms. For this reason and to improve response times, SAP introduced the concept of the enqueue service analogous to AS ABAP.



The JAVA Instance Concept

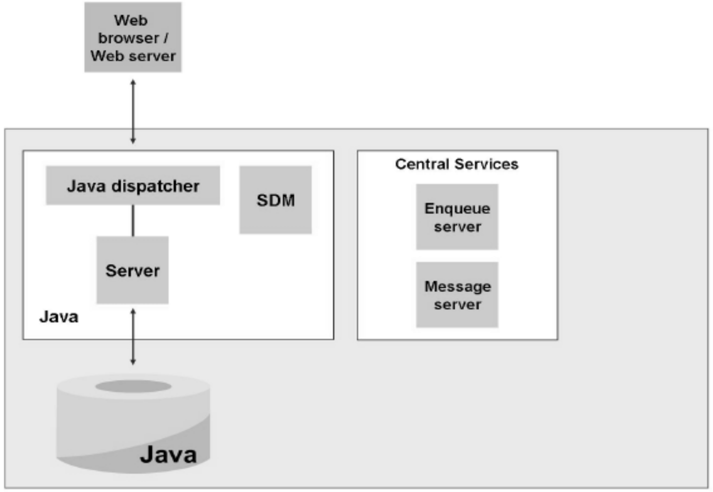
- SAP NetWeaver Application Server Java takes the instance concept of SAP Basis, which combines multiple components that are monitored, started, and stopped together into an administrative unit.
- The following components are part of a Java instance:
  - The server processes provide the infrastructure in which the J2EE applications run.
  - The Java dispatcher distributes the client requests to the free server processes of the instance.
- An instance always runs on one physical server, but there can be multiple instances on one server. Within an SAP system, an instance is defined using the system ID (SID) of the SAP system and the instance number. An SAP system consists of a database and one or more instances. These instances can either be purely ABAP or Java instances, or instances with ABAP and Java infrastructure.
- The Central Services form a special Java instance. They provide the basis of communication and synchronization within a Java cluster. The central instance is another special instance. This runs on a physical server with the Central Services.
- During its installation, the Software Deployment Manager is also installed.

JAVA Cluster Architecture

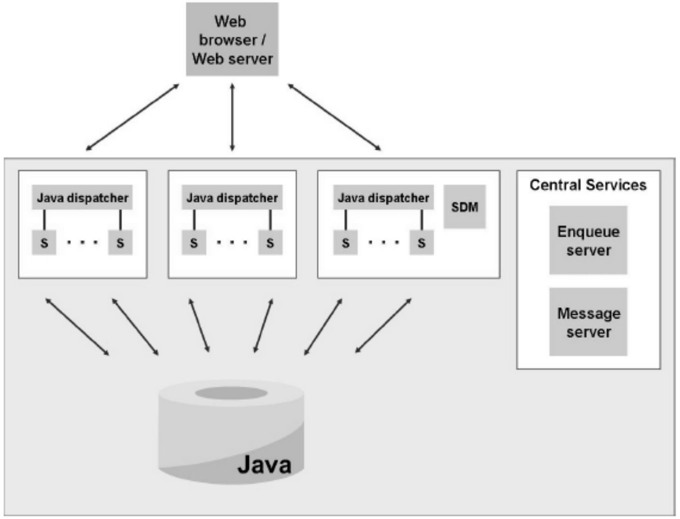
All Java components of an SAP system are known as a Java cluster. A Java cluster consists of:

- A (central) Java instance with a dispatcher and at least one server process.
- The Central Services, which contain a message server and an enqueue server.
- A database for the central storage of data.
- Optionally, additional Java instances

Minimal cluster

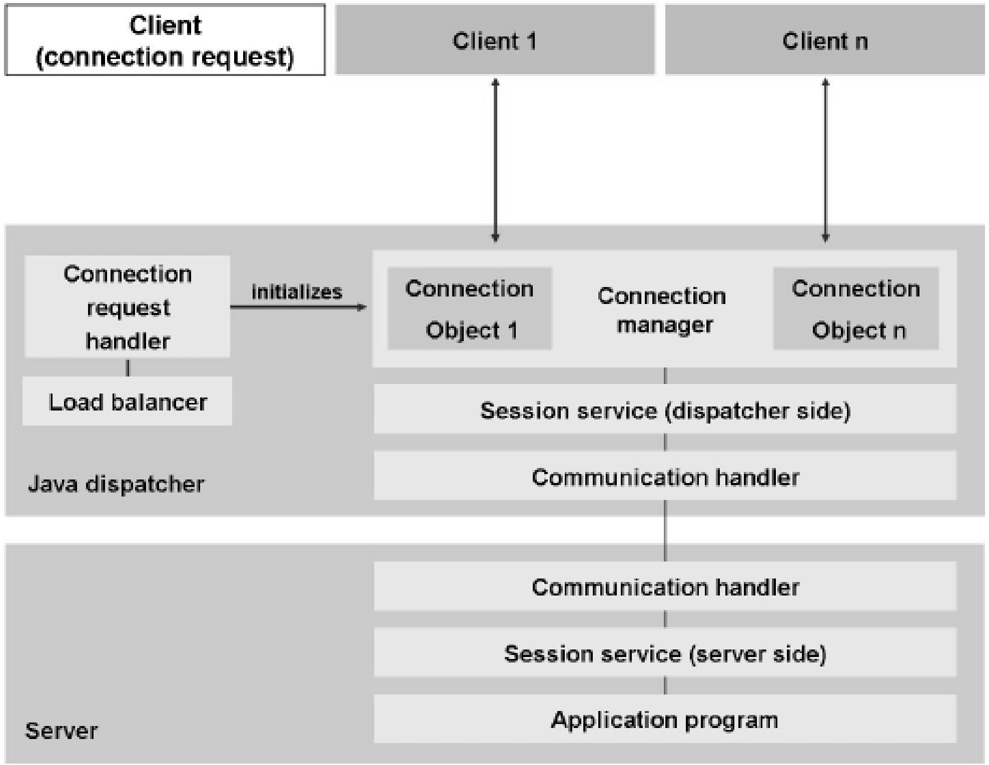


Cluster with Multiple Instances



Request Processing by the dispatcher

- Client requests to SAP NetWeaver AS Java are received by the Java dispatcher. It selects a free server process to process the request and creates the connection between the client and the server process. The dispatcher distributes new incoming requests to the available server processes using a .round robin. algorithm. If there is already a connection to the client, the request is forwarded to the server process that is already processing requests for this client.
- The individual steps in the processing of a request are outlined in the following figure.



Central Services

The Central Services run on one host and form a separate Java instance. They consist of the message service and the enqueue service.

The Central Services provide the basis for communication and synchronization for the Java cluster:

- The message service administers a list of the dispatchers and the server processes of the Java cluster. It represents the infrastructure of data exchange (for small quantities of data) between the nodes involved. In the case of load balancing between a large number of Java instances, it also provides the load balancing information for the SAP Web Dispatcher.
- The enqueue service administers logical locks that are set in a server process by the executed application program. It is also used for cluster-wide synchronization.

The Central Services are essentially required when a Java cluster is installed. They are started on a host with a separate system number and the system ID (SID) of the entire system. If the Central Services are running, other Java instances are started with the program JControl (see the unit Starting and Stopping a SAP NetWeaver AS Java).

Message & Enqueue Service

The message service performs the following tasks in the Java cluster:

- Notification of events that arise in the cluster, for example, if a node of the cluster disappears (due to failure or the instance being shut down), or when a service is started or stopped.
- Communication between different services
- Forwarding of messages and requests to all participants (broadcast)
- Prepare logon information for the SAP Web Dispatcher
- Support for message server failover
- Guaranteed message transmission
- Exchange of cache information in the cluster

The enqueue service runs on the Central Services instance of the Java cluster. It manages the lock table in the main memory and receives requests for setting or releasing locks. It uses the tried and tested SAP lock concept.

The enqueue service has the following tasks:

- Internally, it is used for synchronization within the Java cluster
- The applications can lock objects are release locks again. The enqueue service processes these requests and manages the lock table with the existing locks.

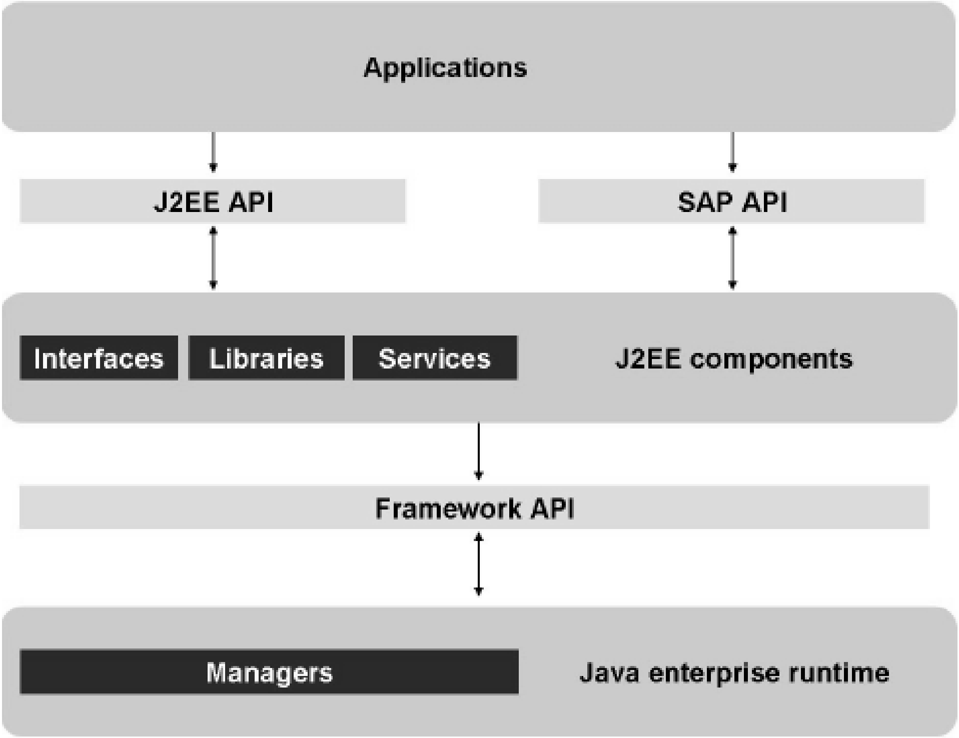
Internal Structure of AS JAVA

There are 3 logical levels

SAP Java Enterprise Runtime - provides fundamental functions of the runtime environment, such as class loading, cluster communication, management of configuration data, and so on

J2EE components - contain interfaces, libraries, and services

Applications Layer - relates to the applications that are deployed and run in SAP NetWeaver Application Server Java.

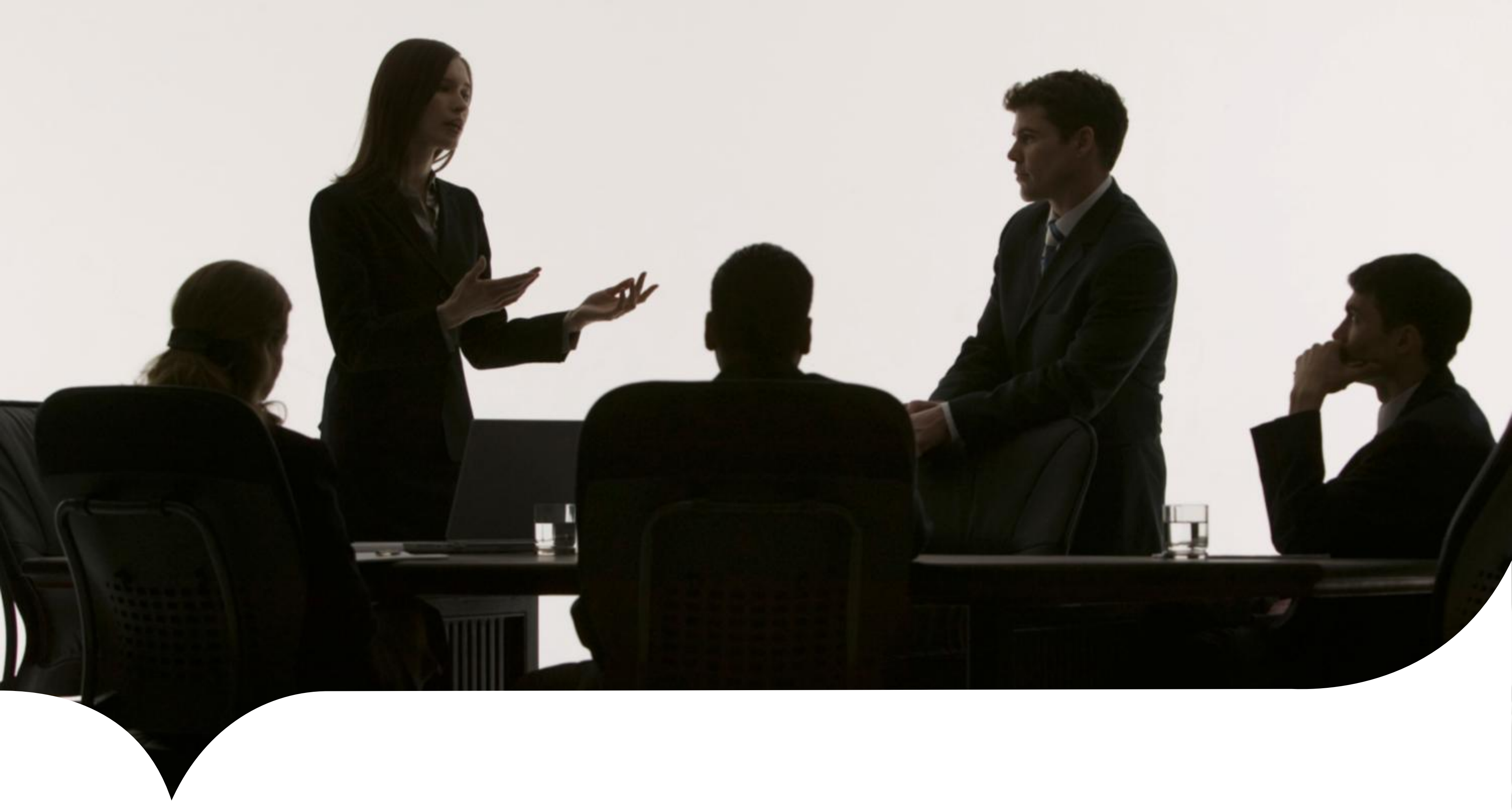


Services in the AS JAVA

The services that SAP NetWeaver AS Java provides for processing requests are defined and configured using “Services”. Service components can access & utilize functions of the runtime through the Framework API. They are the most important of J2EE components

Service	Description
Security Provider	Administration of users and groups and authorization administration. Controls access to resources or applications deployed in SAP NetWeaver AS Java.
Monitoring Service	Allows access to information about the current system status. Provides general and statistical information, among other things, about the nodes in the cluster, memory utilization, performance, applications, and user connections.
Log Configurator Service	Manages the configuration of the logging and tracing mechanism of SAP NetWeaver AS Java.
Log Viewer Service	Provides an integrated log viewer for displaying log and trace information.
Deploy Service	Manages the deployment of Java applications.
EJB Container Service	Manages all Enterprise Java Beans (session beans, entity beans, and message-driven beans), which are executed in the EJB Container of SAP NetWeaver AS Java.
HTTP Provider	Analyzes the URL of inbound HTTP requests, forwards the requests to the correct module of SAP NetWeaver AS Java for processing, and returns the responses to the client.

Break



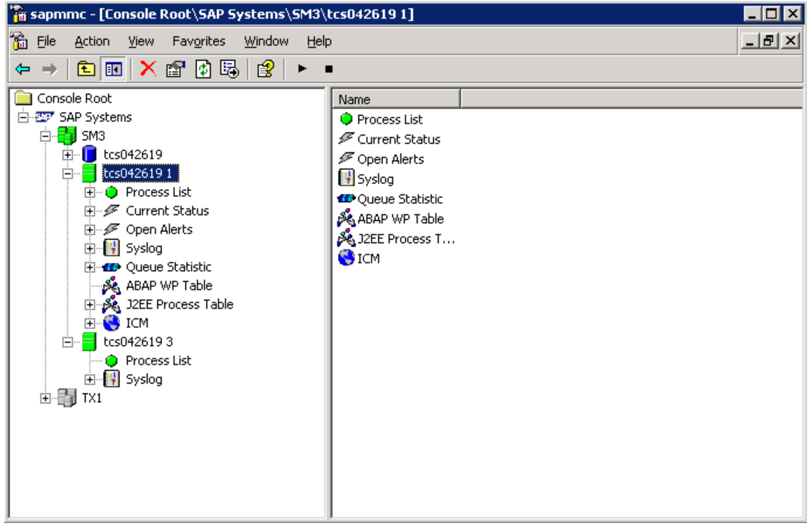
Startup & Shutdown of AS JAVA Instances

Startup & Shutdown of AS JAVA Systems

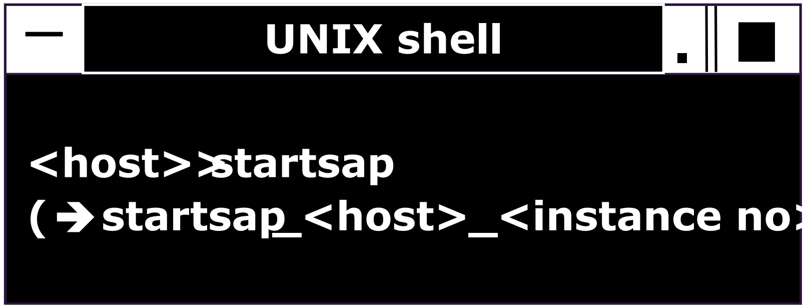
The startup commands for AS ABAP are also valid for AS JAVA systems. Except for the option “r3” in startsap/stopsap command line executable , the option to be used is “j2ee”

Similar to AS ABAP , both command line and SAP MMC are used for starting and stopping of AS JAVA systems

Windows OS



Unix OS

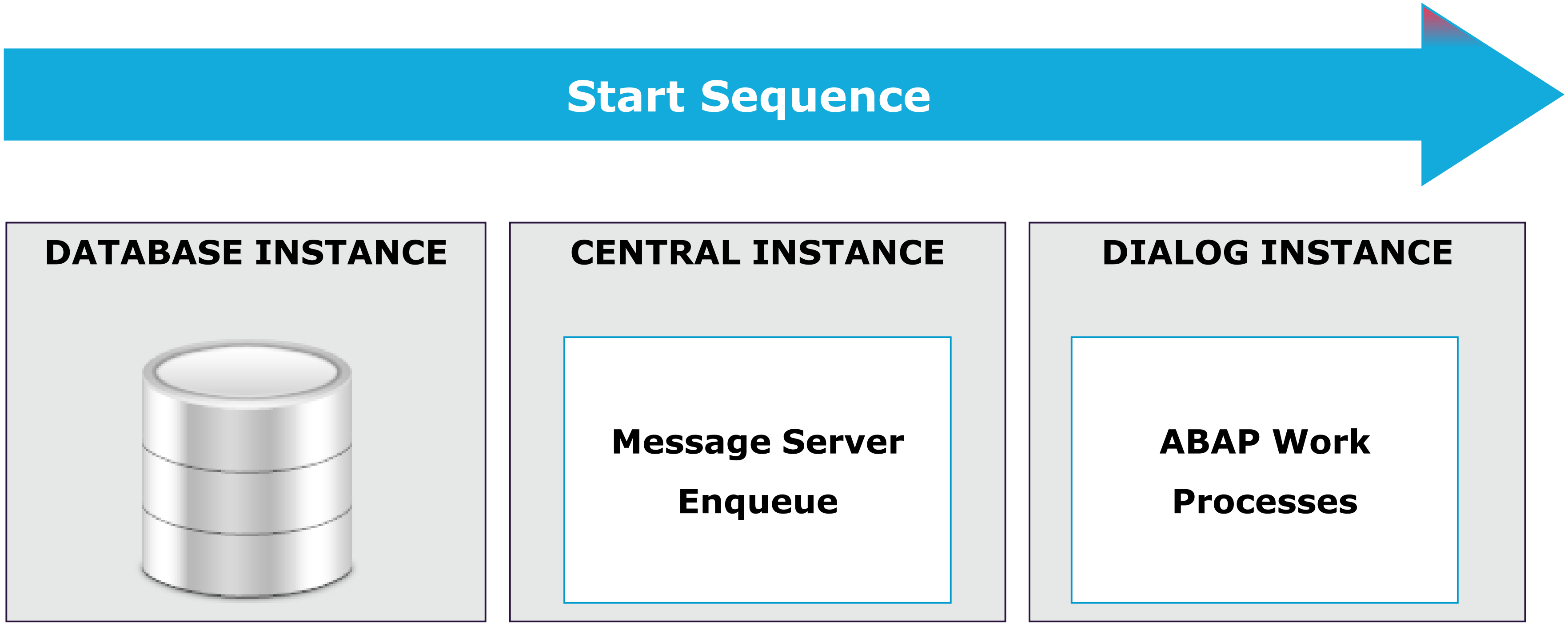


```
<host>>startsap
(→startsap_<host>_<instance no
```

Sequence of Startup

The SAP System comprises of Database Instance , Central Services and finally the multiple Dialog Instances

The sequence of startup is extremely important. Starting the processes of sequence will result in an inconsistent state and may spawn zombie processes on the OS which will have to killed



Start Sequence

DATABASE INSTANCE

CENTRAL INSTANCE

Message Server

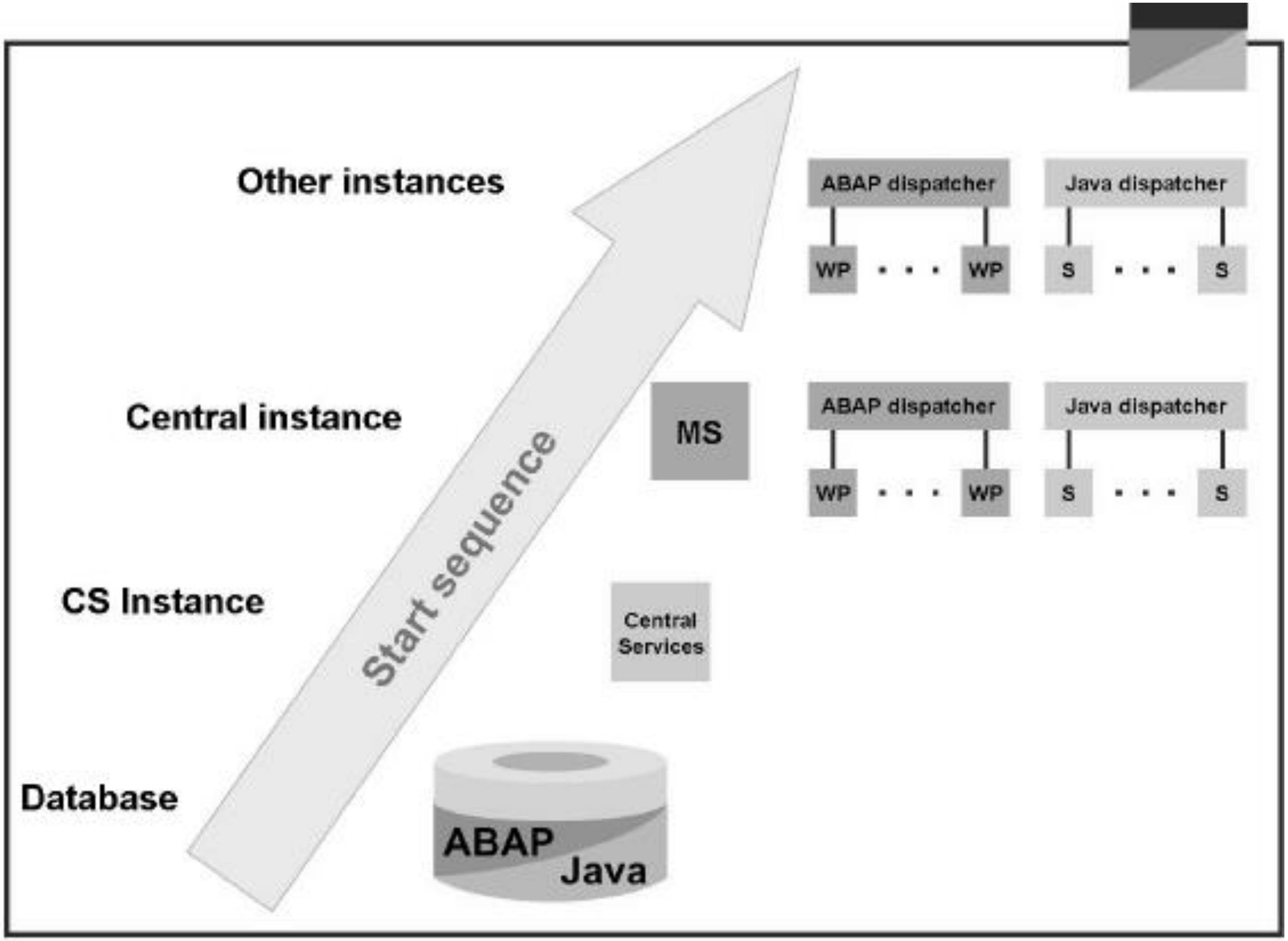
Enqueue

DIALOG INSTANCE

ABAP Work

Processes

Starting AS JAVA Processes

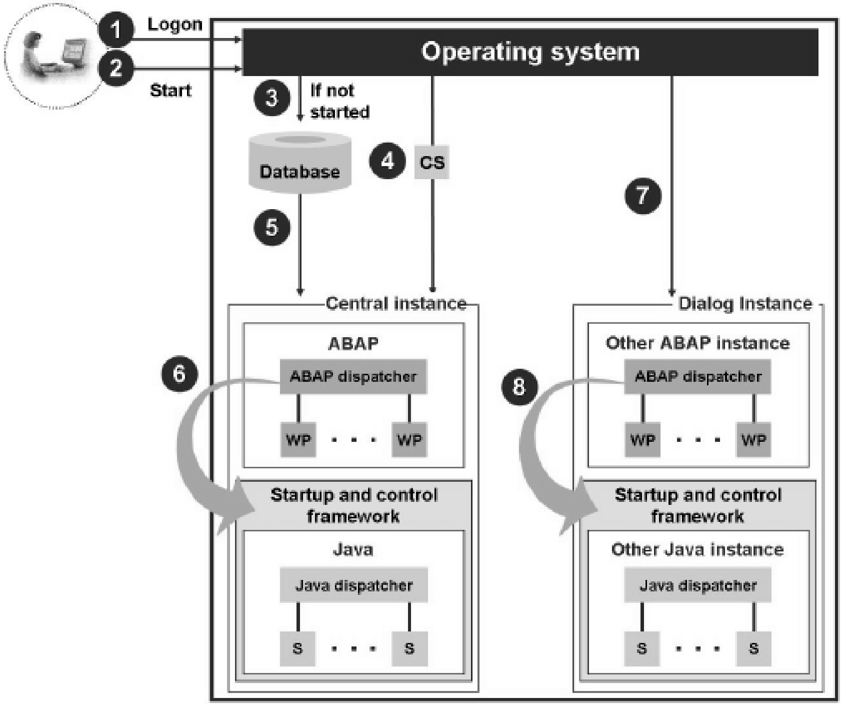


Stopping AS JAVA in UNIX

The stopping of an SAP system is performed in reverse sequence. The dialog instances are always stopped first, and then the central instance.

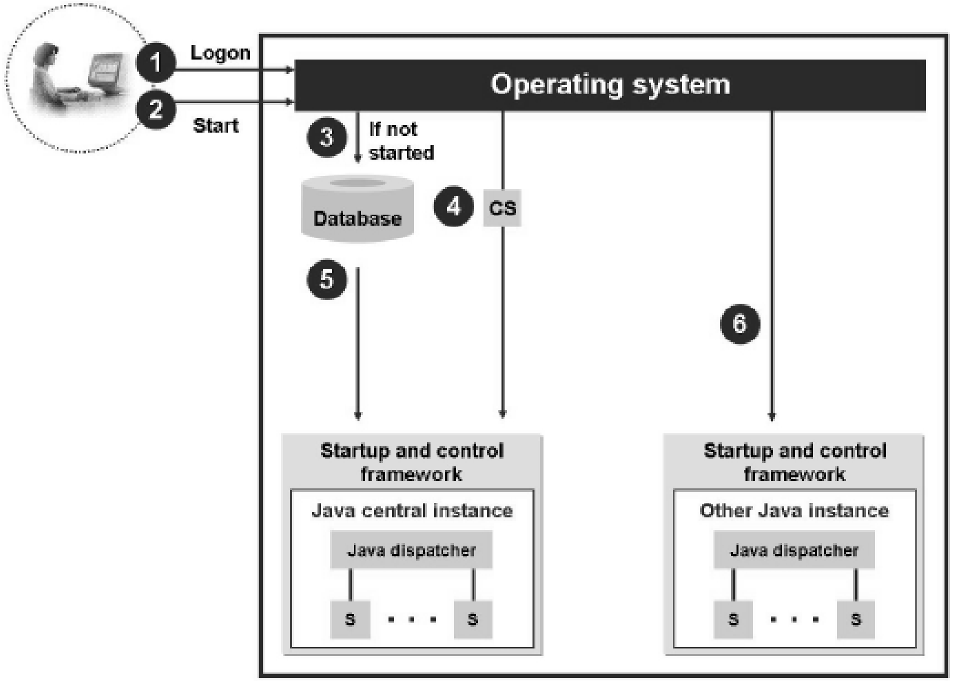
For SAP system installations on the Microsoft Windows operating system, the database is not stopped at the same time. This can be done with the tools of the relevant database. In the case of SAP system that are installed on UNIX operating systems, the database is generally stopped at the same time.

Starting/Stopping of Dual Stack Systems



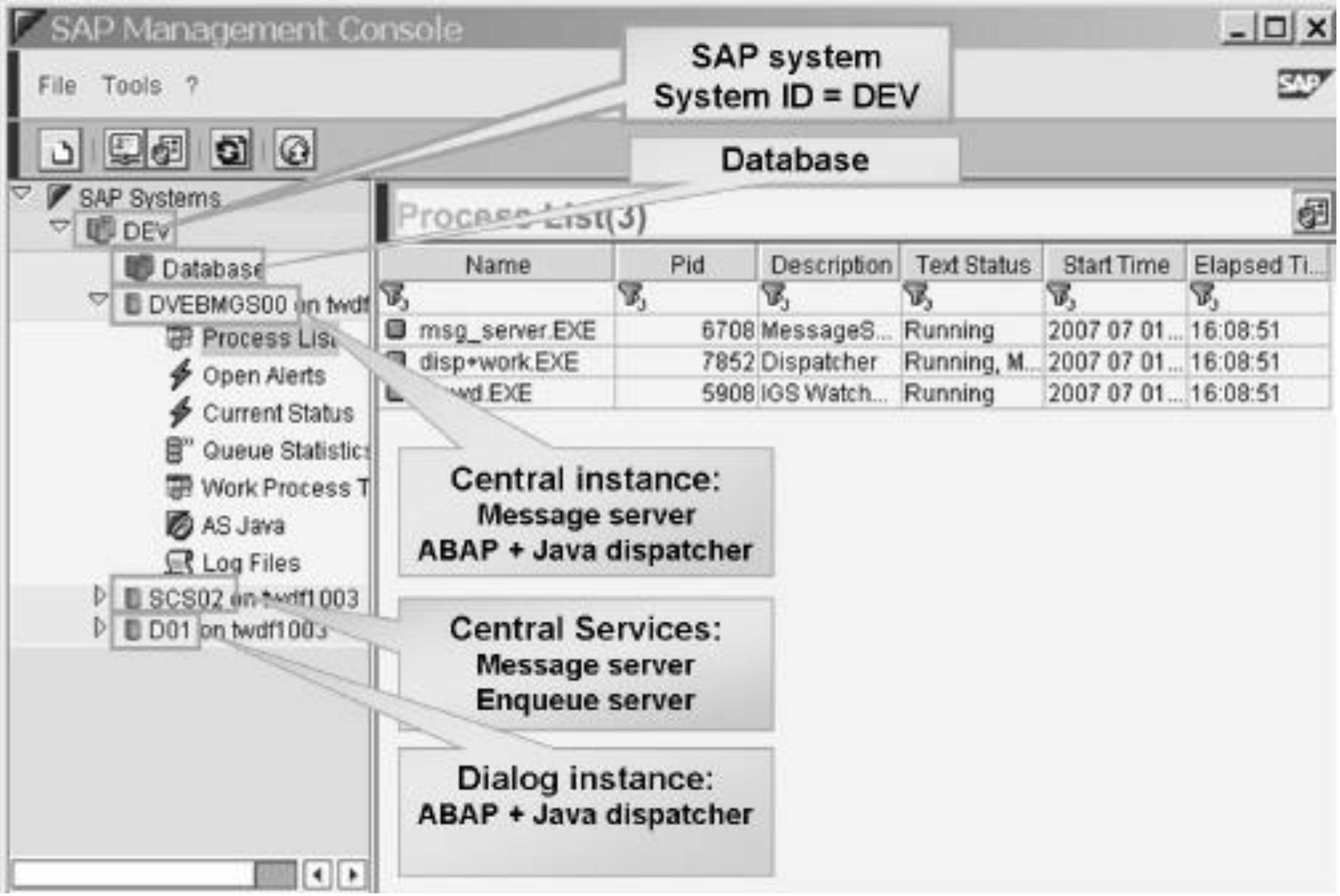
During the installation of an SAP system, you can add a Java stack to each ABAP instance. The instances are started and stopped using appropriate tools at operating system level - such as the SAP Management Console (SAP MC). Within an instance,the Java stack is controlled by the ABAP dispatcher. In concrete terms, this means that the start and stop processes are triggered by the ABAP dispatcher. To do this,the ABAP dispatcher sends a start command to the so-called Startup and Control Framework. The corresponding Java cluster elements are started using the Startup and Control Framework.

Starting/Stopping a pure AS JAVA instance



Pure Java instances can not be managed by the ABAP dispatcher. The start and stop processes can be initiated using appropriate operating system commands -such as the SAP Management Console (SAP MC) or the Microsoft Management Console (SAP MMC) under Windows as well as the scripts startsap and stopsap under UNIX.

Start/Stop Using MMC Console



Name	Pid	Description	Text Status	Start Time	Elapsed Ti...
msg_server.EXE	6708	MessageS...	Running	2007 07 01...	16:08:51
disp+work.EXE	7852	Dispatcher	Running, M...	2007 07 01...	16:08:51
wd.EXE	5908	IGS Watch...	Running	2007 07 01...	16:08:51

SAP system
System ID = DEV

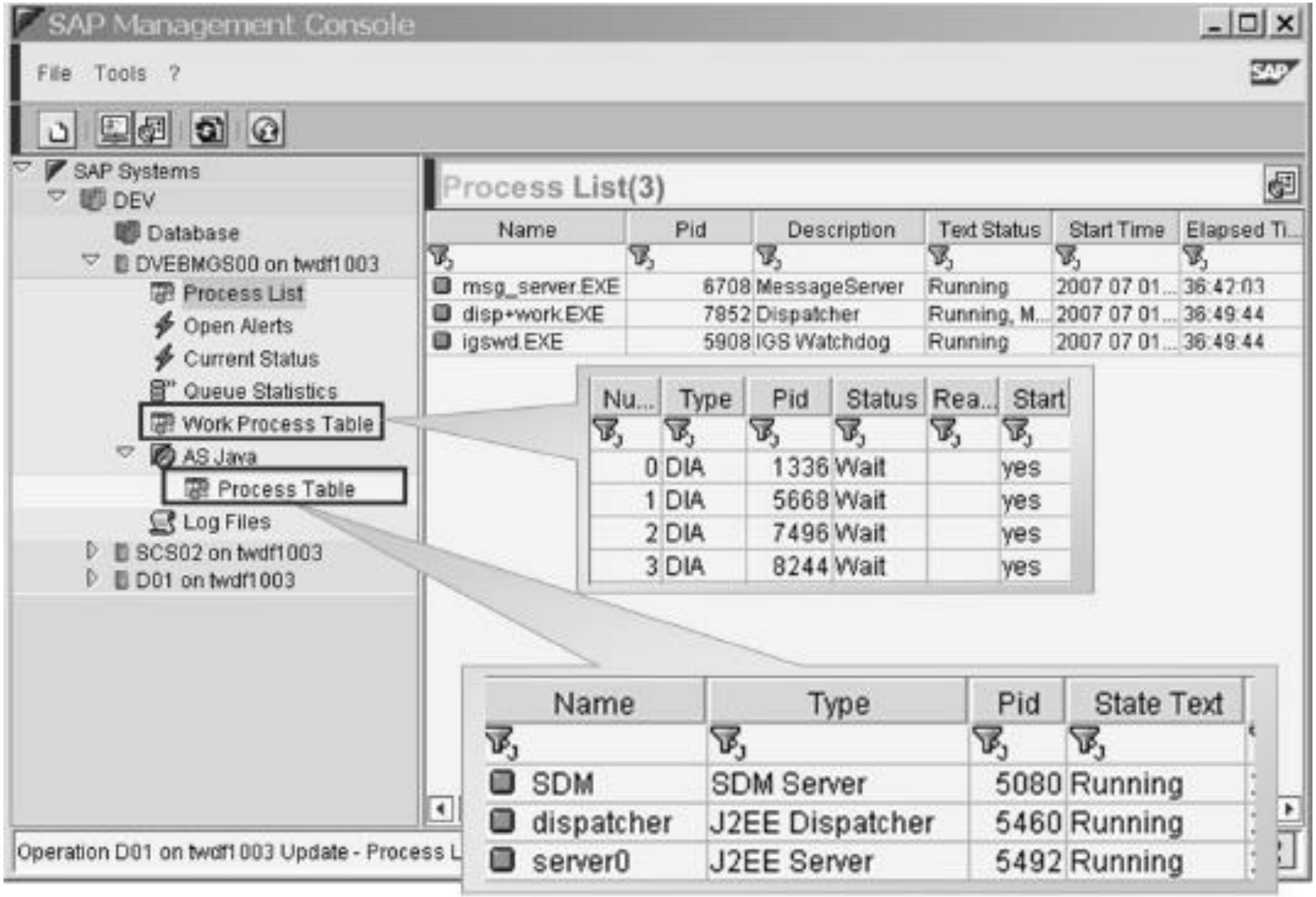
Database

Central instance:
Message server
ABAP + Java dispatcher

Central Services:
Message server
Enqueue server

Dialog instance:
ABAP + Java dispatcher

Process Information



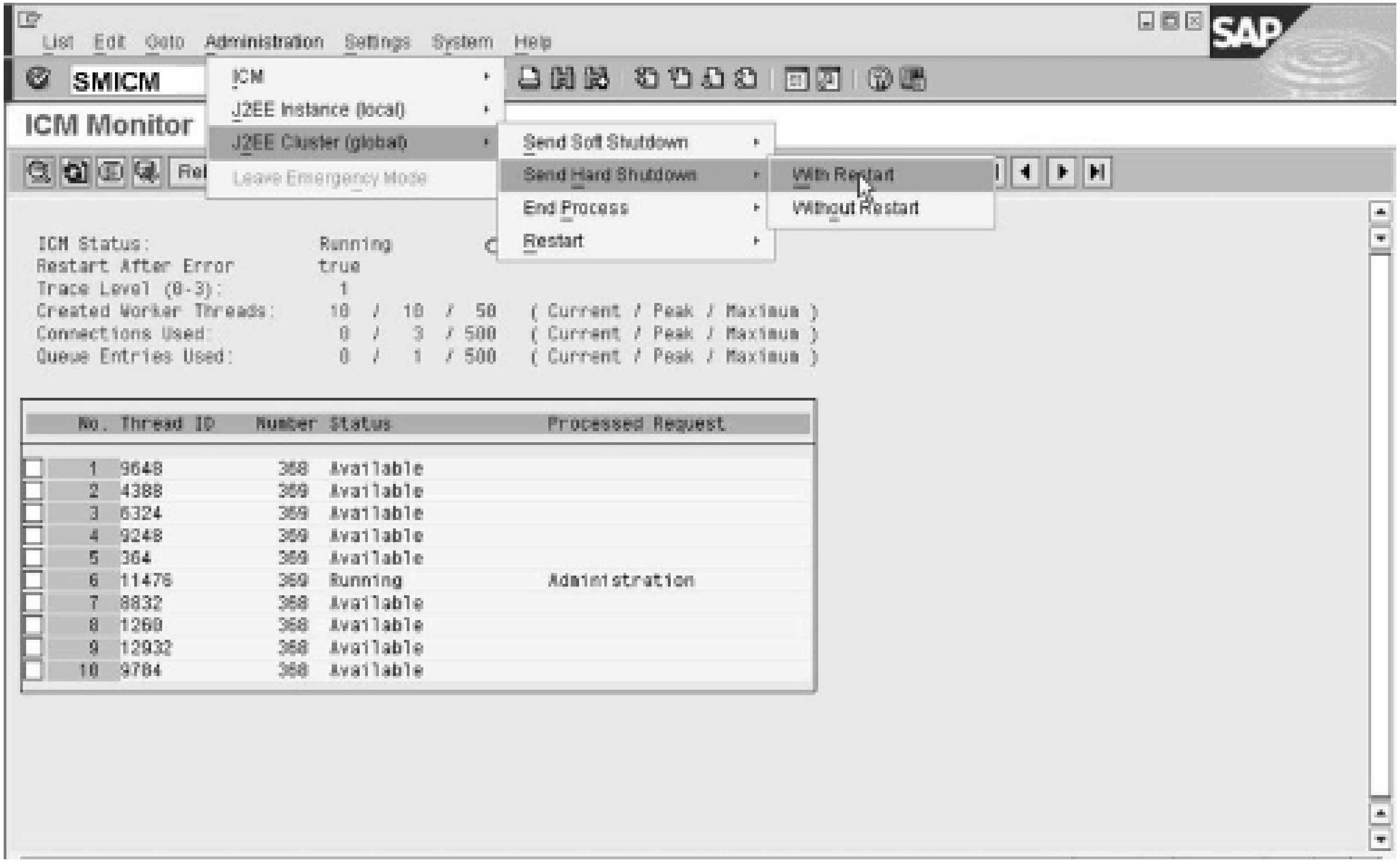
Name	Pid	Description	Text Status	Start Time	Elapsed Ti...
msg_server.EXE	6708	MessageServer	Running	2007 07 01...	36:42:03
disp+work.EXE	7852	Dispatcher	Running, M...	2007 07 01...	36:49:44
igswd.EXE	5908	IGS Watchdog	Running	2007 07 01...	36:49:44

Nu...	Type	Pid	Status	Rea...	Start
0	DIA	1336	Wait		yes
1	DIA	5668	Wait		yes
2	DIA	7496	Wait		yes
3	DIA	8244	Wait		yes

Name	Type	Pid	State Text
SDM	SDM Server	5080	Running
dispatcher	J2EE Dispatcher	5460	Running
server0	J2EE Server	5492	Running

Operation D01 on twdf1003 Update - Process L

Start/Stop using SMICM



```
ICM Status:                 Running
Restart After Error         true
Trace Level (0-3):          1
Created Worker Threads:     10  /  10  /  50    ( Current / Peak / Maximum )
Connections Used:           0   /  3   / 500    ( Current / Peak / Maximum )
Queue Entries Used:         0   /  1   / 500    ( Current / Peak / Maximum )
```

```
   No. Thread ID    Number Status          Processed Request

□   1  9648         368    Available
□   2  4388         369    Available
□   3  6324         369    Available
□   4  9248         369    Available
□   5  364          369    Available
□   6  11476        369    Running         Administration
□   7  8832         368    Available
□   8  1260         368    Available
□   9  12932        368    Available
□  10  9784         368    Available
```

SMICM Details

In the case of SAP NetWeaver AS ABAP + Java, it is possible to allow the ABAP stack to continue running, and only stop and then restart the Java stack.

You do this using transaction SMICM. You can either start/stop the (local) instance onto which you are logged in the transaction SMICM or start/stop all the instances in the (global) Java cluster (see figure: Starting and Stopping the Java Stack of an SAP NetWeaver AS ABAP + Java from the Transaction SMICM).

It is not possible, and also not useful, to stop only the ABAP stack and leave the Java stack started in the case of AS ABAP + Java.

Break

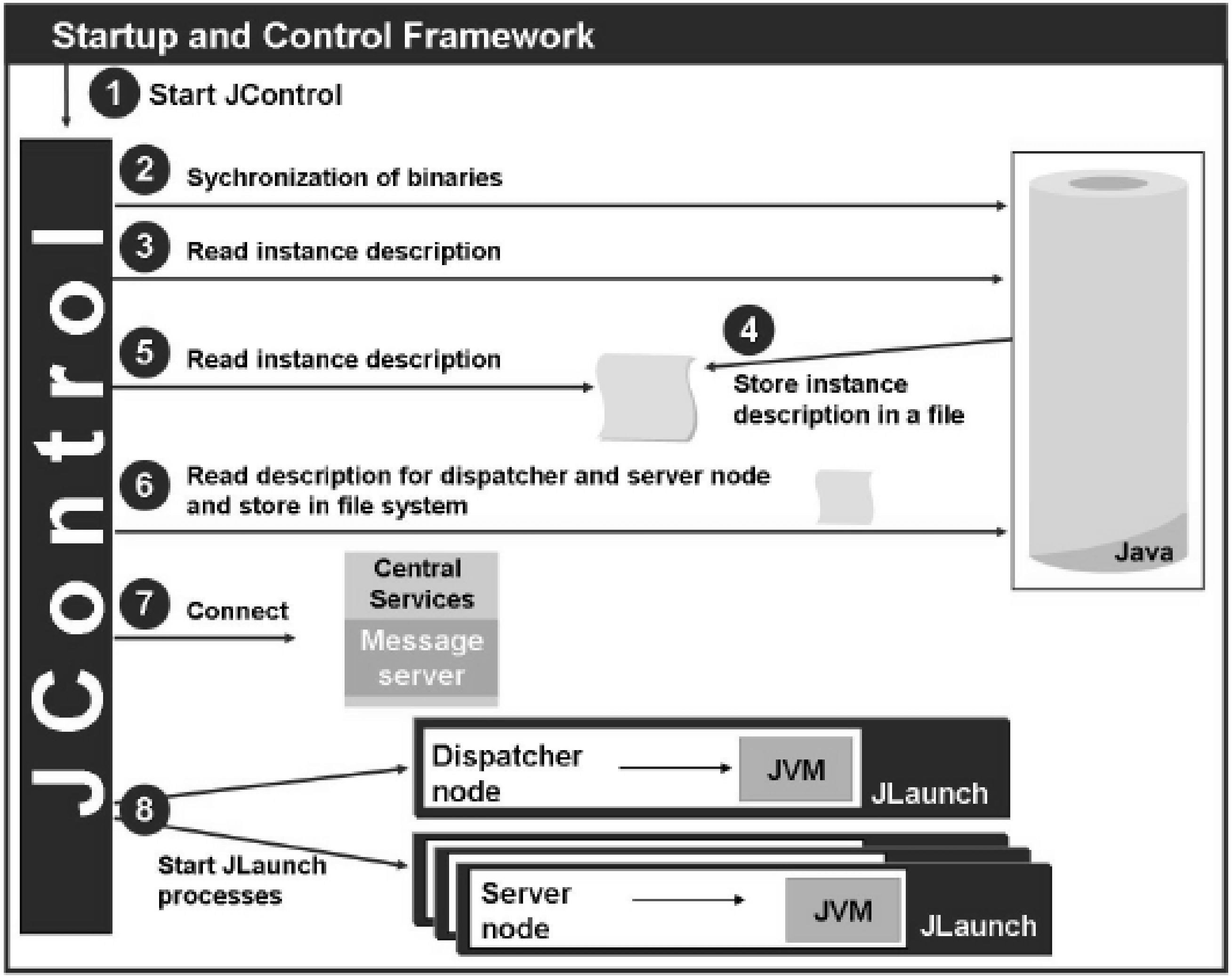
Start/Stop in UNIX

- For SAP systems that are installed under UNIX, you run the scripts startsap and stopsap to initiate the start and stop processes. If multiple SAP instances are installed on one physical server, you add the instance name to the names of the scripts startsap and stopsap.
- The startsap script can be called with the following options:
  - DB: starts the database system
  - R3: starts the instances and associated processes for the instance
  - J2EE: starts the instances and associated processes for the instance
  - ALL: starts the database system and the instance (default setting, can be omitted)
- The stopsap script can be called with the following options:
  - DB: stops the database system with the help of the stopdb script
  - R3: stops the instances and associated processes for the instance
  - J2EE: stops the instances of the SAP system
  - ALL: stops the database system and the instance (default setting, can be omitted)
- To start the SAP system, the startsap script calls the sapstart process with the start profile specified in the script in the variable *START_FILES*.

Continued..

- Database and SAP instances can each be individually started and stopped with the options *DB* or *R3*, e.g. startsap R3 DVEBMGS00.
- Caution**:** The option *J2EE* can be used in the same way as the option *R3*. In the case of SAP NetWeaver AS ABAP + Java, both the ABAP stack and the Java stack are started and stopped.

JAVA Startup and Control Framework



Details on the Framework

SAP provides a separate Startup and Control Framework for SAP Web Application Server Java. This Framework is used to start, stop, and monitor the Java stack processes within the SAP system instances - but not, however, the Central Services.It consists of the following processes:

JControl:

- JControl starts, stops, and monitors the processes of a Java instance . Primarily the server and dispatcher processes. SAP Signal Handling is implemented with JControl, to forward the start and stop commands to the processes of the Java stack.
- JControl restarts terminated processes, ends hanging processes, and sends a shutdown signal to the processes of the Java stack.
- JControl reads the description of the instance from profile files.
- JControl starts the JLaunch processes, which in turn start Java nodes such as dispatchers
  or servers.
- JControl creates a shared memory segment for the internal administration data of all JLaunch processes.

JLaunch:

- JLaunch starts a Java program, loads a Java VM (JVM) in its own address space and assumes the function of the corresponding cluster element. The parameterizing of the JVM is read before the loading.
- JLaunch receives commands from the JControl process (through named pipes) to stop cluster elements such as dispatchers or servers.
- The JLaunch process ends itself if its parent process JControl is no longer running.

AS JAVA Profile Files

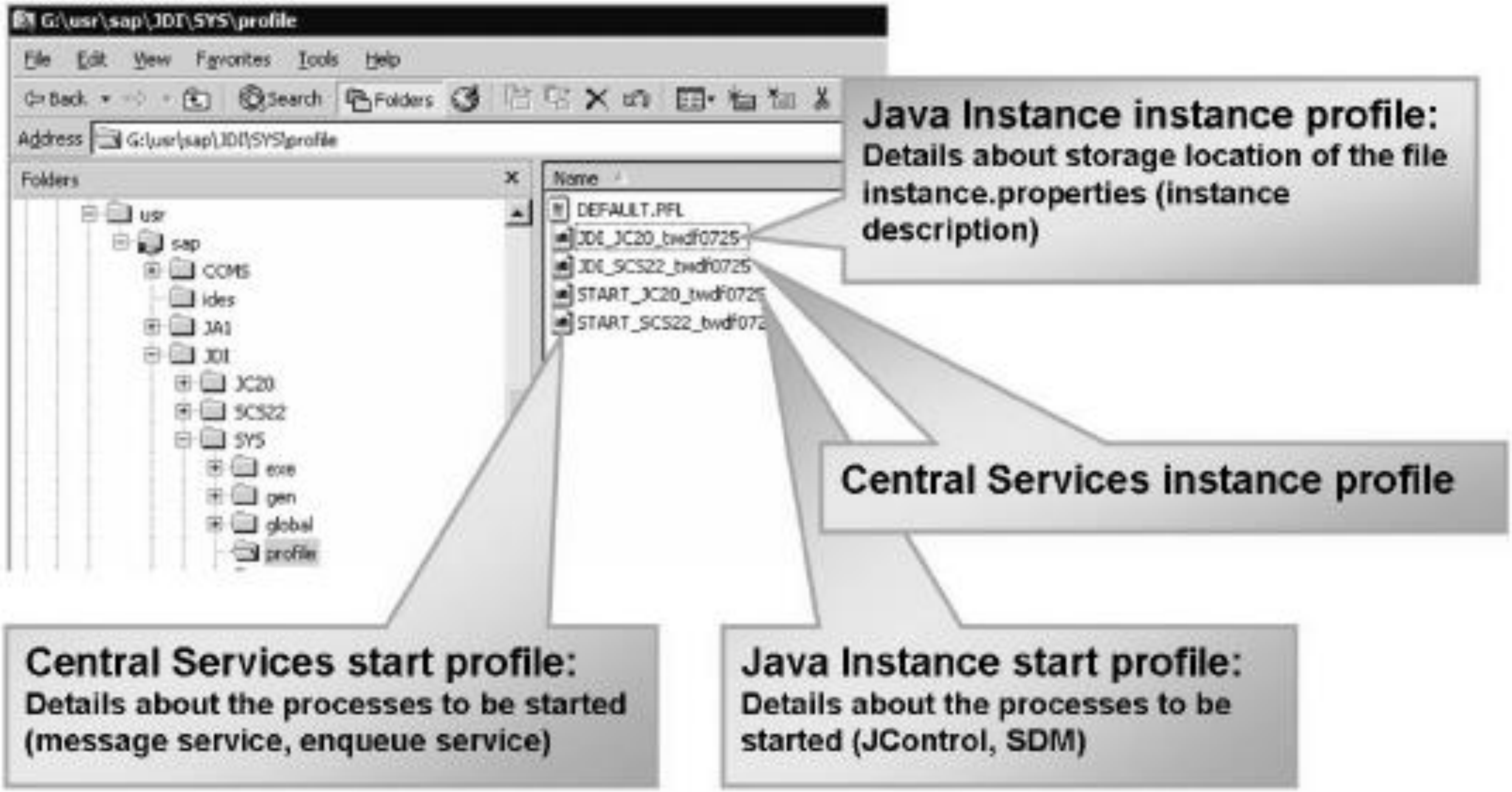
Profile files are first read when a Java instance is started. The JControl process then reads the files instance.properties and bootstrap.properties.

The profile files are located on the operating system in the directory DIR_PROFILE (Microsoft Windows: <drive>:\usr\sap\<SID>\SYS\profile or UNIX: /usr/sap/<SID>/SYS/profile) and are generated at installation time.

There are three different profile files:

- Default profile (Default.pfl),
- Start profile (START_<instance>_<host>), and
- Instance profile (<SID>_<instance>_<host>)

Location in Windows



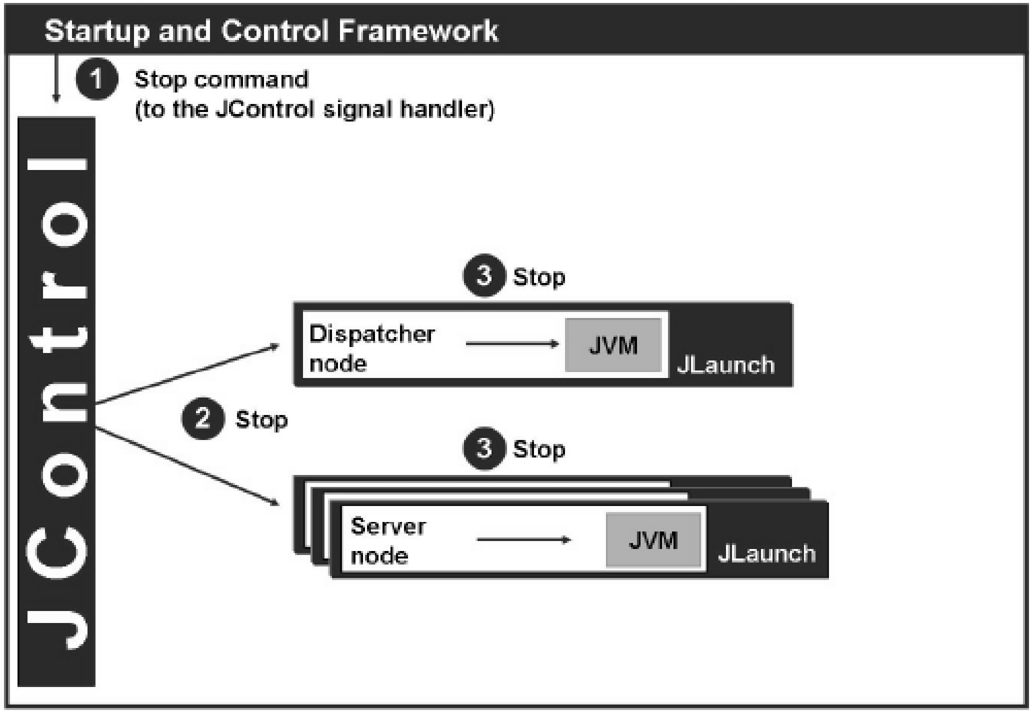
Java Instance instance profile: Details about storage location of the file instance.properties (instance description)

Central Services instance profile

Central Services start profile: Details about the processes to be started (message service, enqueue service)

Java Instance start profile: Details about the processes to be started (JControl, SDM)

Stop process using Startup and Control Framework

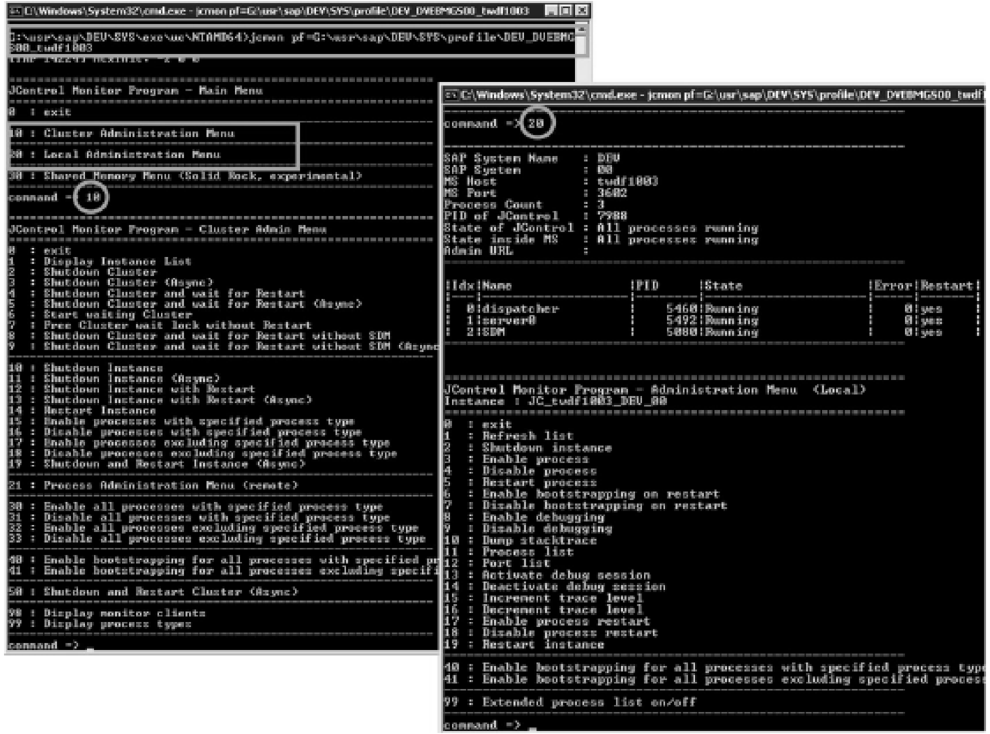


The Signal Handler of JControl receives a stop signal from the stopsap script (UNIX) or from the Windows Service.

JControl forwards the signal through named pipes to all running JLaunch processes (Java servers
and dispatchers).

The JLaunch process of a Java cluster element should react to this notification (soft shutdown) within a certain period of time. After this time has expired,the JLaunch process is terminated by JControl. When this is done, the JVM within the JLaunch process is ended, as is the JVM of the JLaunch process.

JCMON Tool



```
C:\Windows\System32\cmd.exe - jcmon pf=G:\usr\sap\DEV\SYS\profile\DEV_DVEBMGS00_twdf1003

G:\usr\sap\DEV\SYS\exe\uc\NTAMD64>jcmon pf=G:\usr\sap\DEV\SYS\profile\DEV_DVEBMG
S00_twdf1003

============================================================
JControl Monitor Program - Main Menu
============================================================
0  : exit
------------------------------------------------------------
10 : Cluster Administration Menu
------------------------------------------------------------
20 : Local Administration Menu
------------------------------------------------------------
30 : Shared Memory Menu (Solid Rock, experimental)
------------------------------------------------------------
command => 10

============================================================
JControl Monitor Program - Cluster Admin Menu
============================================================
0  : exit
1  : Display Instance List
2  : Shutdown Cluster
3  : Shutdown Cluster (Async)
4  : Shutdown Cluster and wait for Restart
5  : Shutdown Cluster and wait for Restart (Async)
6  : Start waiting Cluster
7  : Free Cluster wait lock without Restart
8  : Shutdown Cluster and wait for Restart without SDM
9  : Shutdown Cluster and wait for Restart without SDM (Async
------------------------------------------------------------
10 : Shutdown Instance
11 : Shutdown Instance (Async)
12 : Shutdown Instance with Restart
13 : Shutdown Instance with Restart (Async)
14 : Restart Instance
15 : Enable processes with specified process type
16 : Disable processes with specified process type
17 : Enable processes excluding specified process type
18 : Disable processes excluding specified process type
19 : Shutdown and Restart Instance (Async)
------------------------------------------------------------
21 : Process Administration Menu (remote)
------------------------------------------------------------
30 : Enable all processes with specified process type
31 : Disable all processes with specified process type
32 : Enable all processes excluding specified process type
33 : Disable all processes excluding specified process type
------------------------------------------------------------
40 : Enable bootstrapping for all processes with specified pr
41 : Enable bootstrapping for all processes excluding specifi
------------------------------------------------------------
50 : Shutdown and Restart Cluster (Async)
------------------------------------------------------------
98 : Display monitor clients
99 : Display process types
------------------------------------------------------------
command => _
```

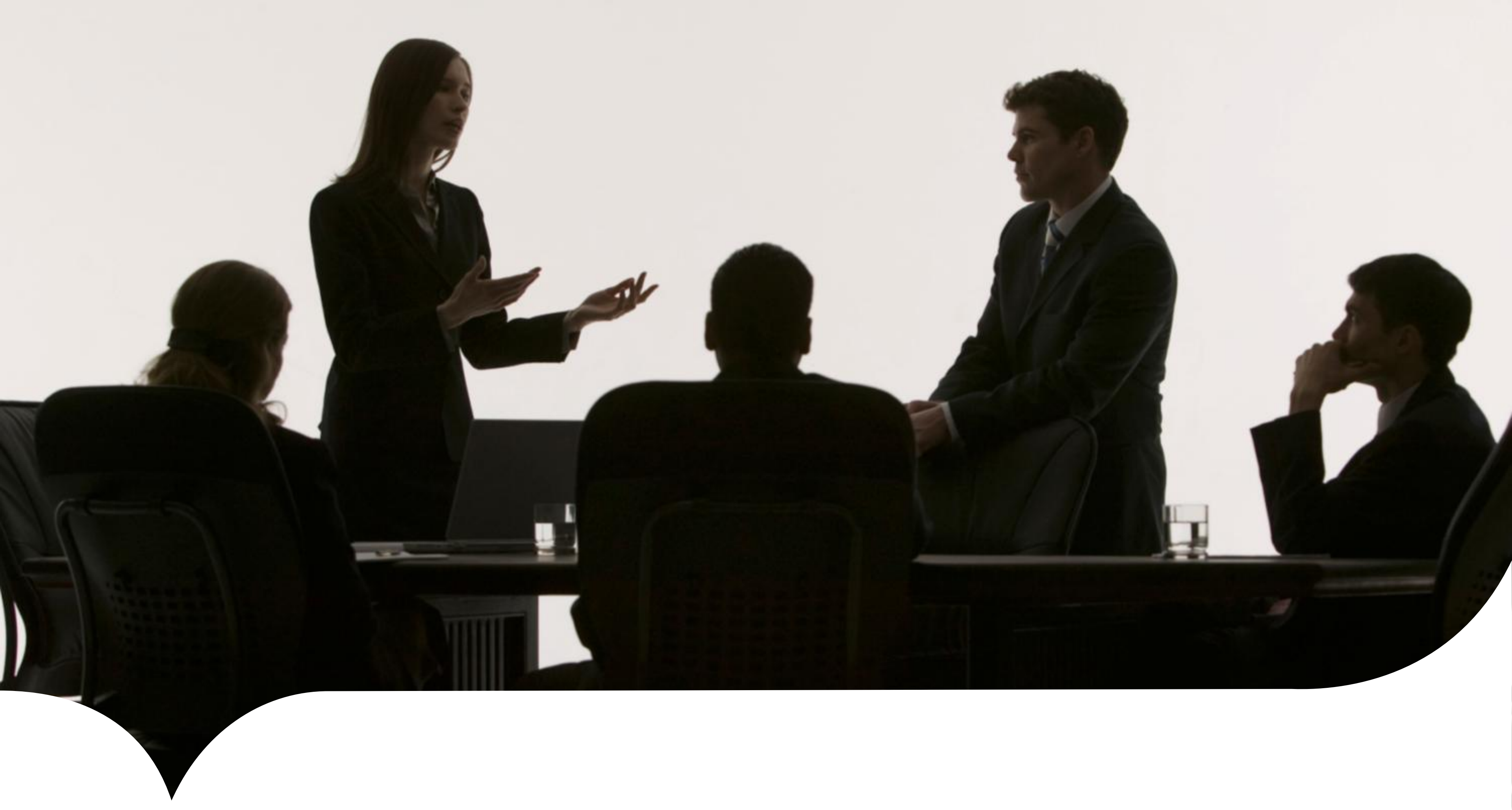
```
C:\Windows\System32\cmd.exe - jcmon pf=G:\usr\sap\DEV\SYS\profile\DEV_DVEBMGS00_twdf
------------------------------------------------------------
command => 20

------------------------------------------------------------
SAP System Name    : DEV
SAP System         : 00
MS Host            : twdf1003
MS Port            : 3602
Process Count      : 3
PID of JControl    : 7988
State of JControl  : All processes running
State inside MS    : All processes running
Admin URL          :
------------------------------------------------------------

|Idx|Name                |PID      |State                |Error|Restart|
|---|--------------------|---------|---------------------|-----|-------|
|  0|dispatcher          |     5460|Running              |    0|yes    |
|  1|server0             |     5492|Running              |    0|yes    |
|  2|SDM                 |     5080|Running              |    0|yes    |
------------------------------------------------------------


============================================================
JControl Monitor Program - Administration Menu  (Local)
Instance : JC_twdf1003_DEV_00
============================================================
0  : exit
1  : Refresh list
2  : Shutdown instance
3  : Enable process
4  : Disable process
5  : Restart process
6  : Enable bootstrapping on restart
7  : Disable bootstrapping on restart
8  : Enable debugging
9  : Disable debugging
10 : Dump stacktrace
11 : Process list
12 : Port list
13 : Activate debug session
14 : Deactivate debug session
15 : Increment trace level
16 : Decrement trace level
17 : Enable process restart
18 : Disable process restart
19 : Restart instance
------------------------------------------------------------
40 : Enable bootstrapping for all processes with specified process typ
41 : Enable bootstrapping for all processes excluding specified proces
------------------------------------------------------------
99 : Extended process list on/off
------------------------------------------------------------
command => _
```

The JCmon tool can be used to monitor the JControl process. JCmon is part of the Startup and Control Framework, and is located in the JControl/JLaunch home directory, that is, in the executable directory /usr/sap/<SID>/<instance>/SYS/exe/....

JCmon can be started with the command JCmon pf=<SAP instance profile>.

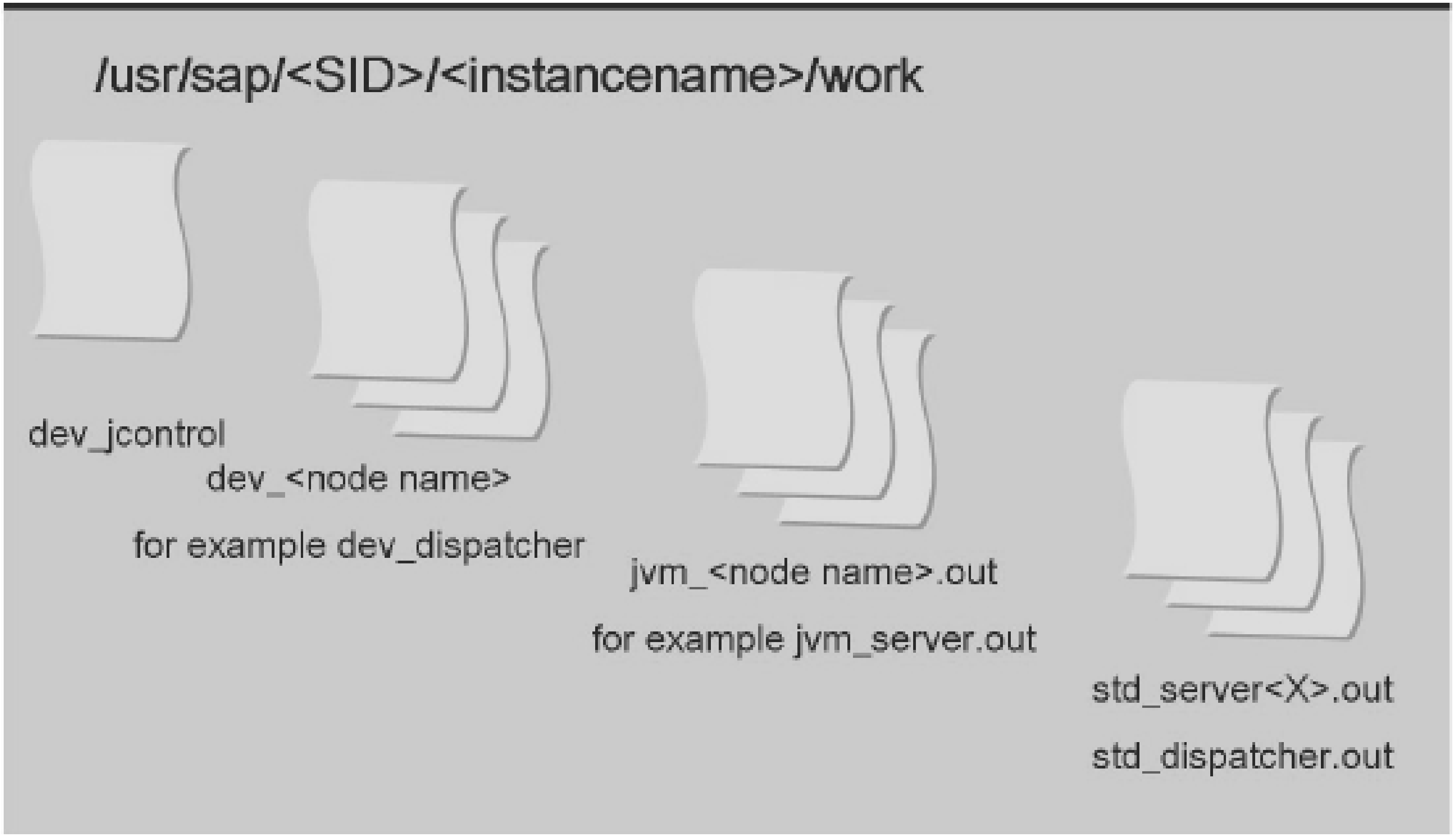
JCmon provides an administration interface for elements in the Java cluster that can be called from the operating system.

Lunch Break



System Administration In AS JAVA

Logs & Trace Files in AS JAVA


/usr/sap/<SID>/<instancename>/work
dev_jcontrol
dev_<node name>
for example dev_dispatcher
jvm_<node name>.out
for example jvm_server.out
std_server<X>.out
std_dispatcher.out

Logs & Trace Files in AS JAVA

The trace and log files are stored in the work directory of an instance. This directory is called /usr/sap/<SID>/<instance name>/work (UNIX) and analogously in the Microsoft Windows environment.

dev_jcontrol is the trace file for the JControl process. dev_jcontrol is the most important trace file for problem messages when starting NetWeaver AS Java. Current messages are written at the end of the file.

dev_<node name> is the trace file for JLaunch processes. The trace file dev_<node name> is written for each started JLaunch process, and therefore for every dispatcher and server process.

jvm_<node name>.out is the output file for the Java Virtual Machine (JVM). This JLaunch process represents a Java node such as a dispatcher or a server and therefore a JVM. The output of a JVM is forwarded to the file jvm_<node name>.out in the work directory of a Java instance.

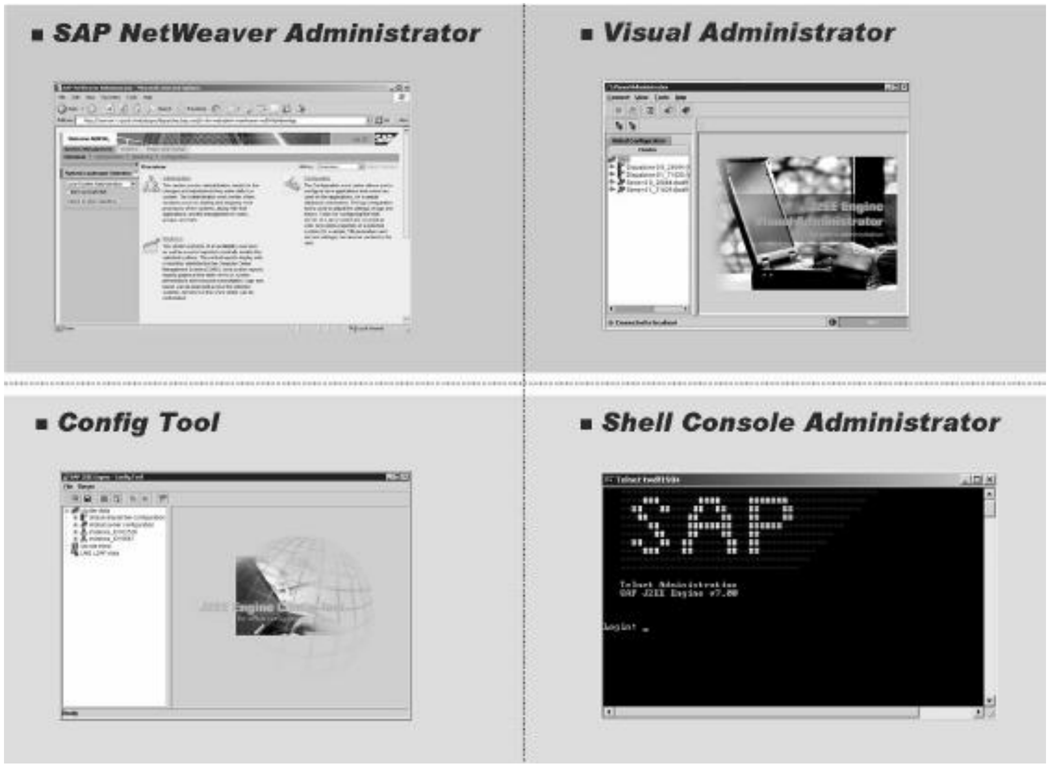
std_server<X>.out and std_dispatcher.out are the default output files for the started managers and services of the the corresponding nodes.

For all of the log files listed above, you will also find log files in the workdirectory

with the ending .old, which can also often be used for troubleshooting.

Administration & Configuration Tools in AS JAVA

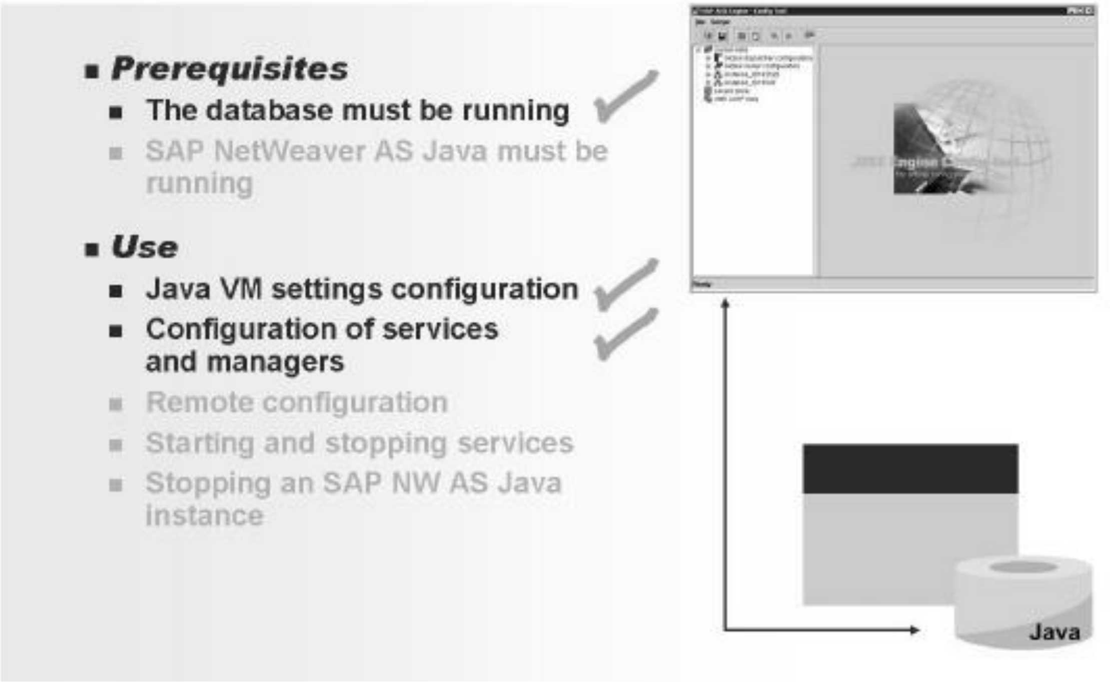
SAP Netweaver Administrator
Visual Administrator
Config Tool
Shell Console



- SAP NetWeaver Administrator
- Visual Administrator
- Config Tool
- Shell Console Administrator

```
Telnet Administration
SAP J2EE Engine v7.00


login: _
```

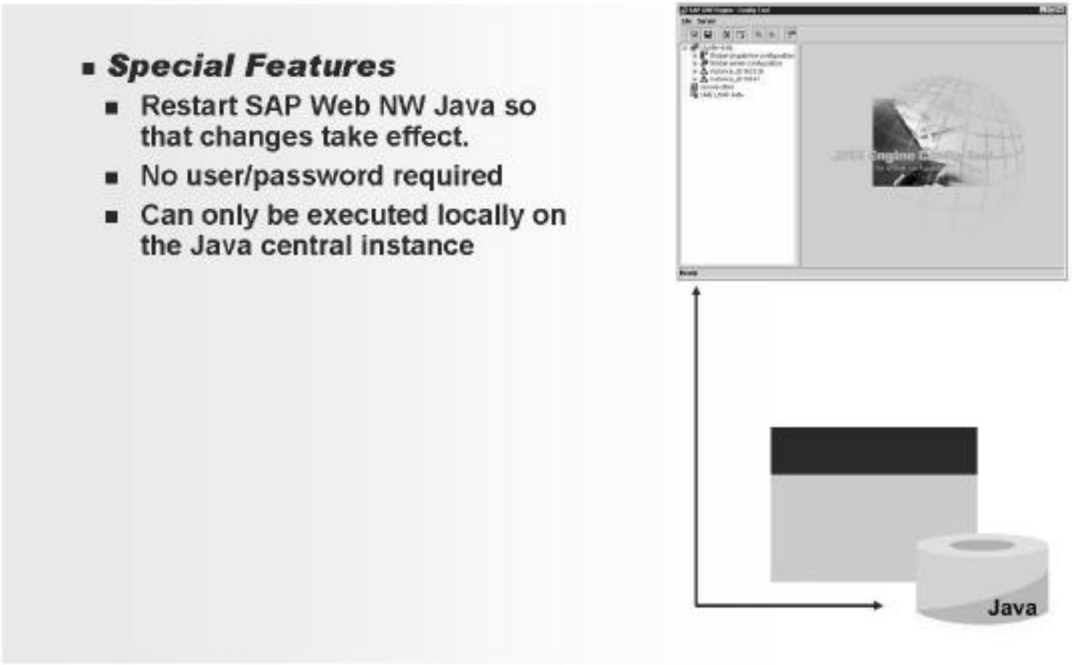
Config Tool



- ***Prerequisites***
  - The database must be running ✓
  - SAP NetWeaver AS Java must be running
- ***Use***
  - Java VM settings configuration ✓
  - Configuration of services and managers ✓
  - Remote configuration
  - Starting and stopping services
  - Stopping an SAP NW AS Java instance

The Java VM Settings (the settings of the Java Virtual Machine) of SAP NetWeaver AS Java can only be maintained with the Config Tool. You can use the Config Tool to configure the managers and services of SAP NetWeaver AS Java.

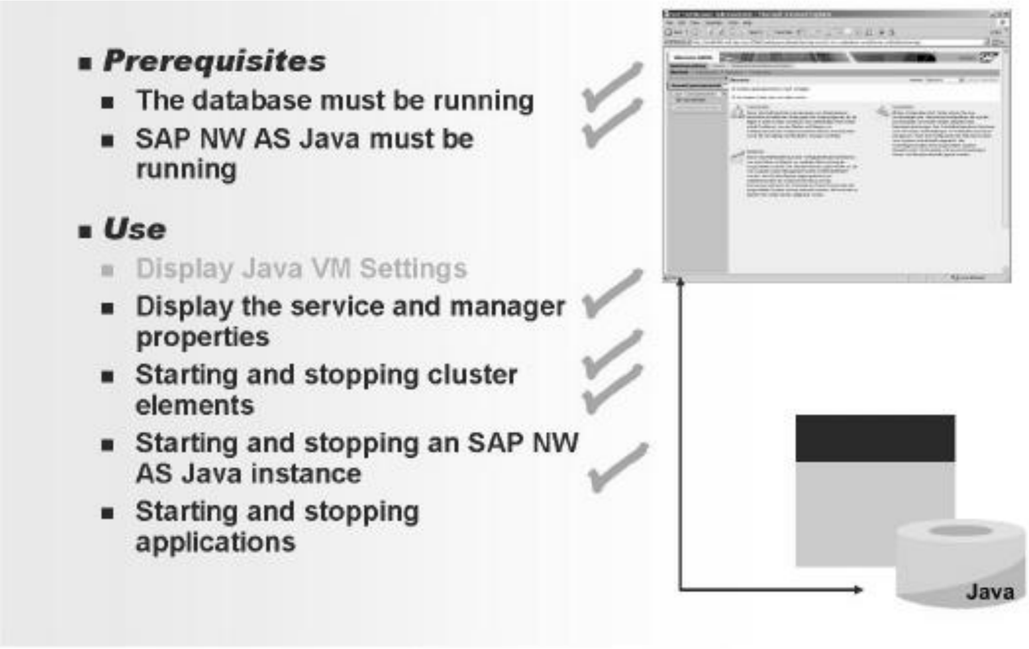
Features of the Config Tool



- ***Special Features***
  - Restart SAP Web NW Java so that changes take effect.
  - No user/password required
  - Can only be executed locally on the Java central instance

- The settings that you make with the Config Tool only take effect when the SAP
- NetWeaver AS Java is started; that is, you must restart the SAP NetWeaver AS Java after maintaining settings.
- The Config Tool is called on the host on which the central instance of the SAP
- NetWeaver AS Java is running. Authentication is performed using the logon to the operating system of the host.

SAP NWA



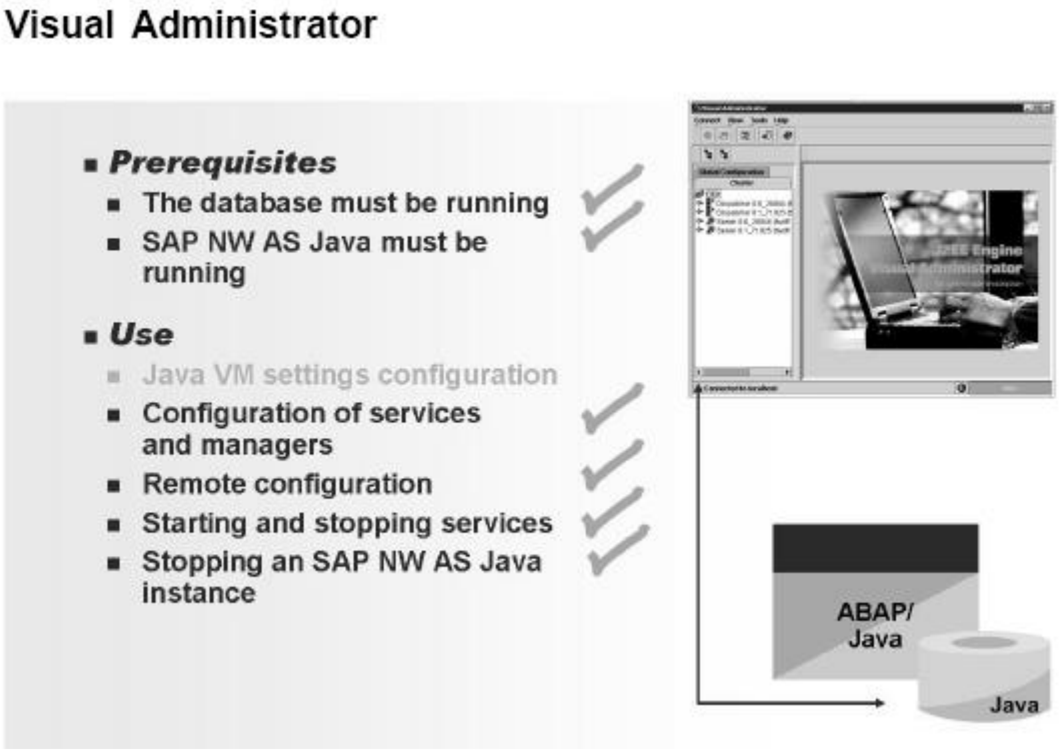
- ***Prerequisites***
  - The database must be running ✓
  - SAP NW AS Java must be running ✓
- ***Use***
  - Display Java VM Settings
  - Display the service and manager properties ✓
  - Starting and stopping cluster elements ✓
  - Starting and stopping an SAP NW AS Java instance ✓
  - Starting and stopping applications ✓

- The SAP NetWeaver Administrator (NWA) combines the most important administration and monitoring tools for Java and ABAP systems in a new, browser-bases user interface. The most important benefits offered by NWA are:
- You no longer need to switch between different tools for the administration, troubleshooting and problem analysis of your entire SAP NetWeaver system landscape.
- Instead, you have for Java, and increasingly for ABAP systems, a central, cross-landscape administration tool which enables you to start and stop instances,
- check configuration settings and logs and monitor the correct functioning of the components.

NWA displays all the monitoring infrastructure data, i.e. alerts together with the current and historical values.

The GUI adheres to the current guidelines concerning interface design, is easy to use, task-oriented and complete. If you use Web Dynpro, it runs in the browser.

Visual Administrator

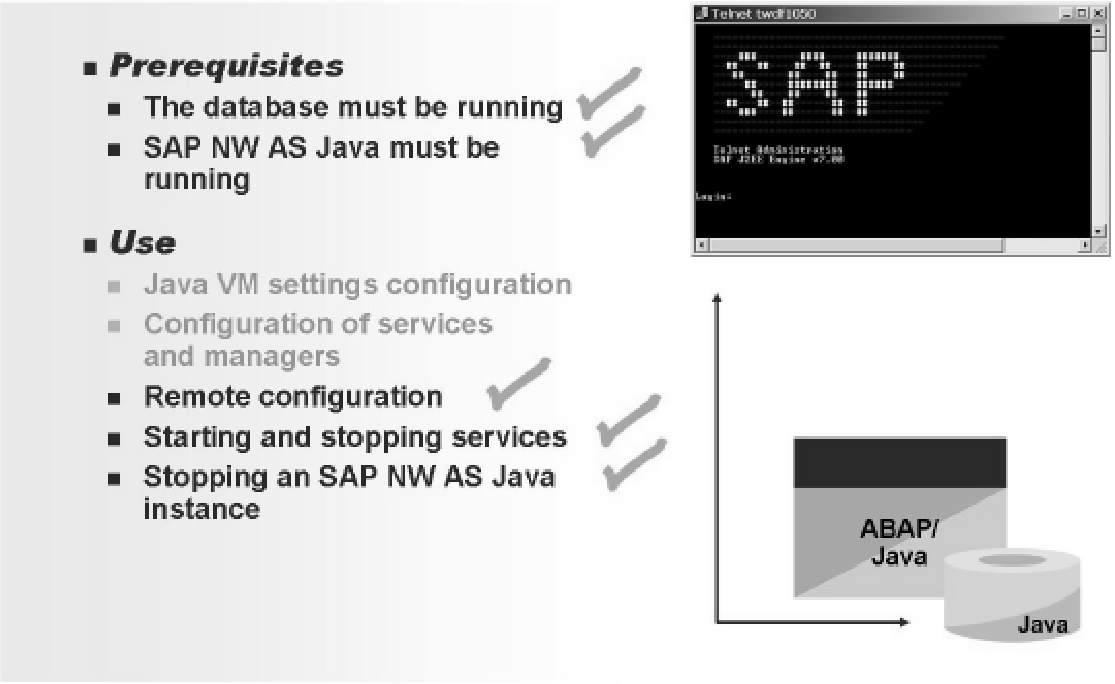


Visual Administrator

- ***Prerequisites***
  - The database must be running ✓
  - SAP NW AS Java must be running ✓
- ***Use***
  - Java VM settings configuration
  - Configuration of services and managers ✓
  - Remote configuration ✓
  - Starting and stopping services ✓
  - Stopping an SAP NW AS Java instance ✓

With the Visual Administrator, you log on to the P4 port of an SAP NetWeaver AS Java instance. It is therefore necessary that the SAP NetWeaver AS Java instance is started. The Visual Administrator can also be installed on a host on which no SAP NetWeaver AS Java instance is installed. In this way, it can be used for remote administration. You can use the Visual Administrator to configure the services and managers of all SAP NetWeaver AS Java instances in the Cluster. The changes to the selected parameters take effect during runtime. You can use the Visual Administrator to start or stop services. You cannot change the Virtual Machine parameters with the Visual Administrator.

Shell Console Administrator



- ***Prerequisites***
  - The database must be running ✓
  - SAP NW AS Java must be running ✓
- ***Use***
  - Java VM settings configuration
  - Configuration of services and managers
  - Remote configuration ✓
  - Starting and stopping services ✓
  - Stopping an SAP NW AS Java instance ✓

```
Telnet Administration
SAP J2EE Engine v7.00

Login:
```

You can use Shell Console Administrator to display basic information for an SAP NetWeaver AS Java system in a simple way. Shell Console Administrator is also suitable for starting and stopping services. You cannot change VM parameters using Shell Console Administrator.

Memory Management in AS JAVA

The memory area of a Java Virtual Machine (JVM or VM) is mainly divided into three areas, which are called the young generation, the tenured generation, and the permanent generation. A "generation" reserves space in the address area of the host.

On start-up, the Virtual Machine allocates operating system memory for each "generation". This initially allocated memory area is called "initial" or initial size.

This occupancy of this space is administered internally by the VM.Once the initial space has been used, the VM allocates further operating system memory space in stages up to a maximum amount. This maximum value is defined in "max size".

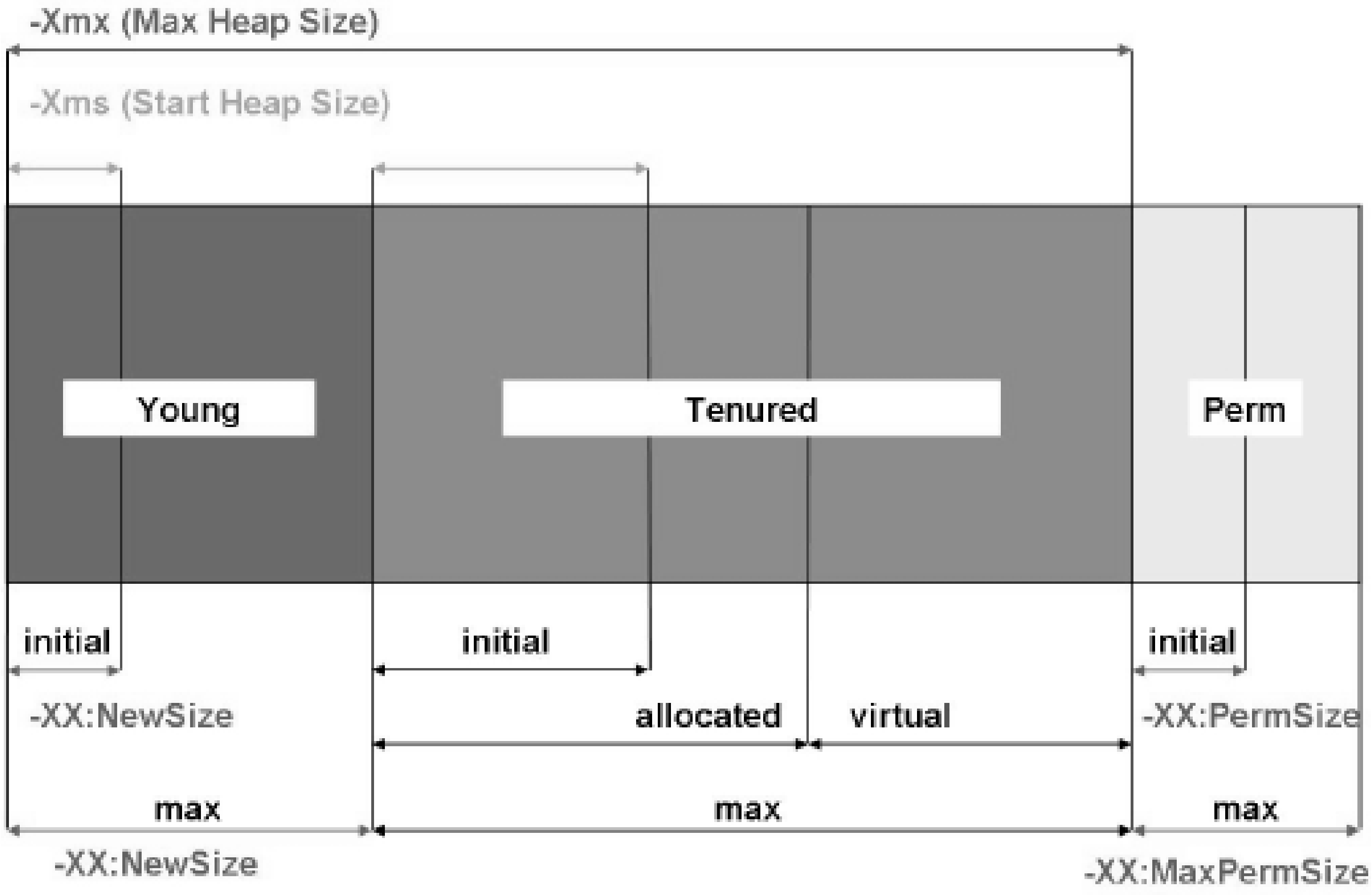
The VM automatically takes care of the allocation of memory space for Java applications. The memory space is implicitly assigned when an object is created.

Even if a large amount of memory is required, this does not mean that the system is at risk. The VM determines which objects are no longer used and releases the memory areas which they currently occupy. This task is performed by the Special Java Agent named Garbage Collector (GC) which forms part of the VM. Its job is to prevent the occurrence of situations in which there is a danger of memory bottlenecks.

The memory space that is available is called the available memory or allocated memory. Since this space is reserved by the operating system, it is also referred to as .reserved. space since the entire space up to the maximum size is .available.. The space that is not yet reserved is called virtual memory. However, this should not be confused with the .virtual memory. of the operating system. If less space is required, the memory is returned to the operating system, also in stages.

The reserved memory space (available memory) is potentially available to the VM. However, it does not have to be used in full. The memory space that is actually used by Java applications is referred to as used memory.

Memory Allocation in AS JAVA



Explanation of the Term “Generation”

The three main memory areas of the VM, the young, tenured, and permanent generations differ from one another due to the data stored in them. The objects that have been newly created by the applications are stored in the young generation

Objects that have been required for a longer period of time by an application are automatically moved to the tenured generation. The newer objects are in the young generation and the older objects are in the tenured generation.

Objects that are permanently required by the VM, such as classes and methods, are stored in the permanent generation

Objects that are no longer required by the applications are automatically removed from the generation. This process is known as garbage collection.

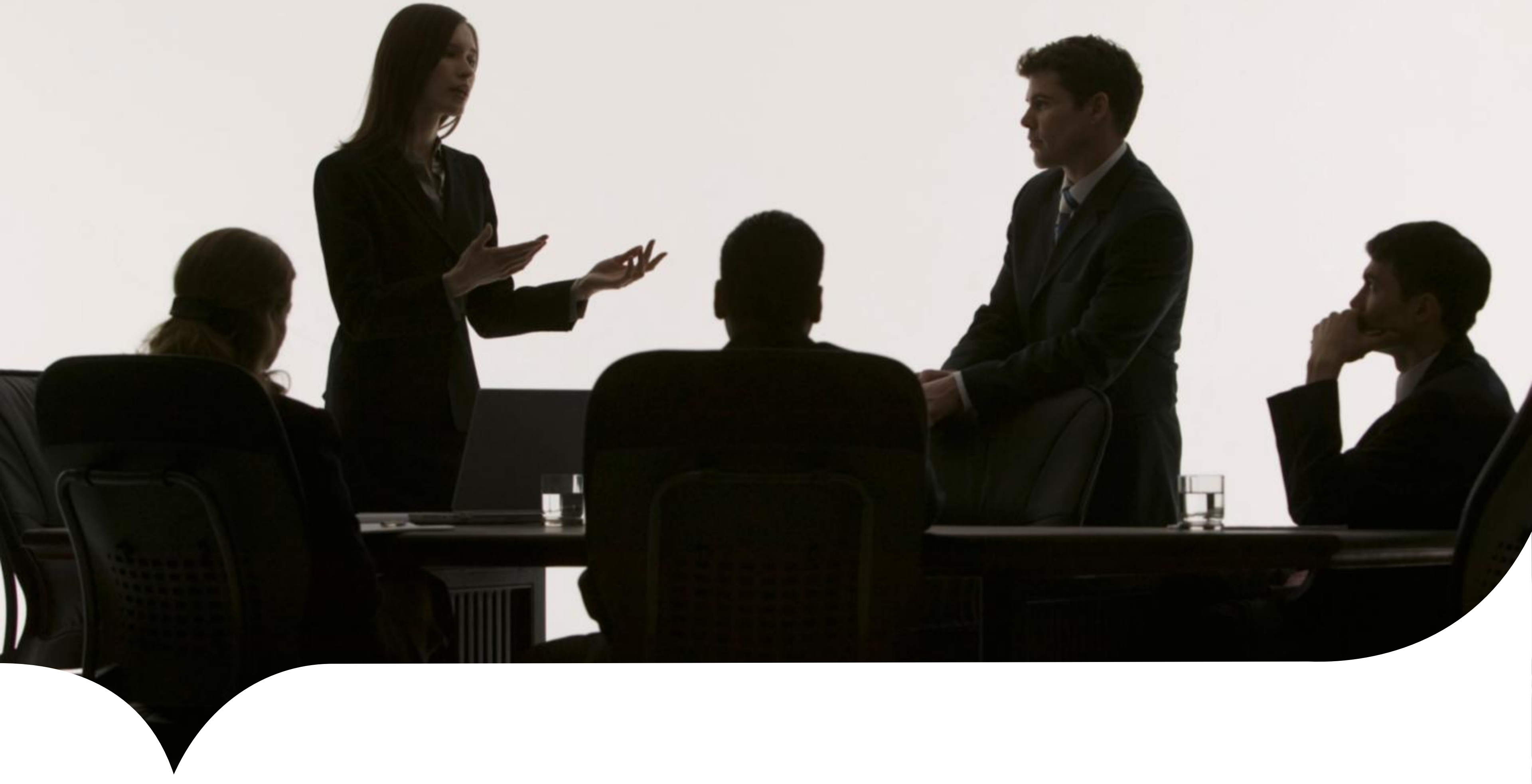
The generations have an initial and a maximum size. For the young generation, you can define the initial size with the parameter -XX:NewSize, and the max size with the parameter -XX:MaxNewSize.

You can define the corresponding values for the permanent generation (also abbreviated to the perm generation) with the parameters -XX:PermSize and -XX:MaxPermSize

You cannot directly define the initial and maximum sizes of the tenured generation. These are calculated from the parameters for the young generation and the parameters -Xmx and -Xms.

The parameter -Xmx is called the max heap size and defines the total size of the young and tenured generations.

The parameter -Xms is called the start heap size or initial heap size and defines the total initial size of the young and tenured generations.



BREAK

User Management in Java & UME

User Management Engine (UME)

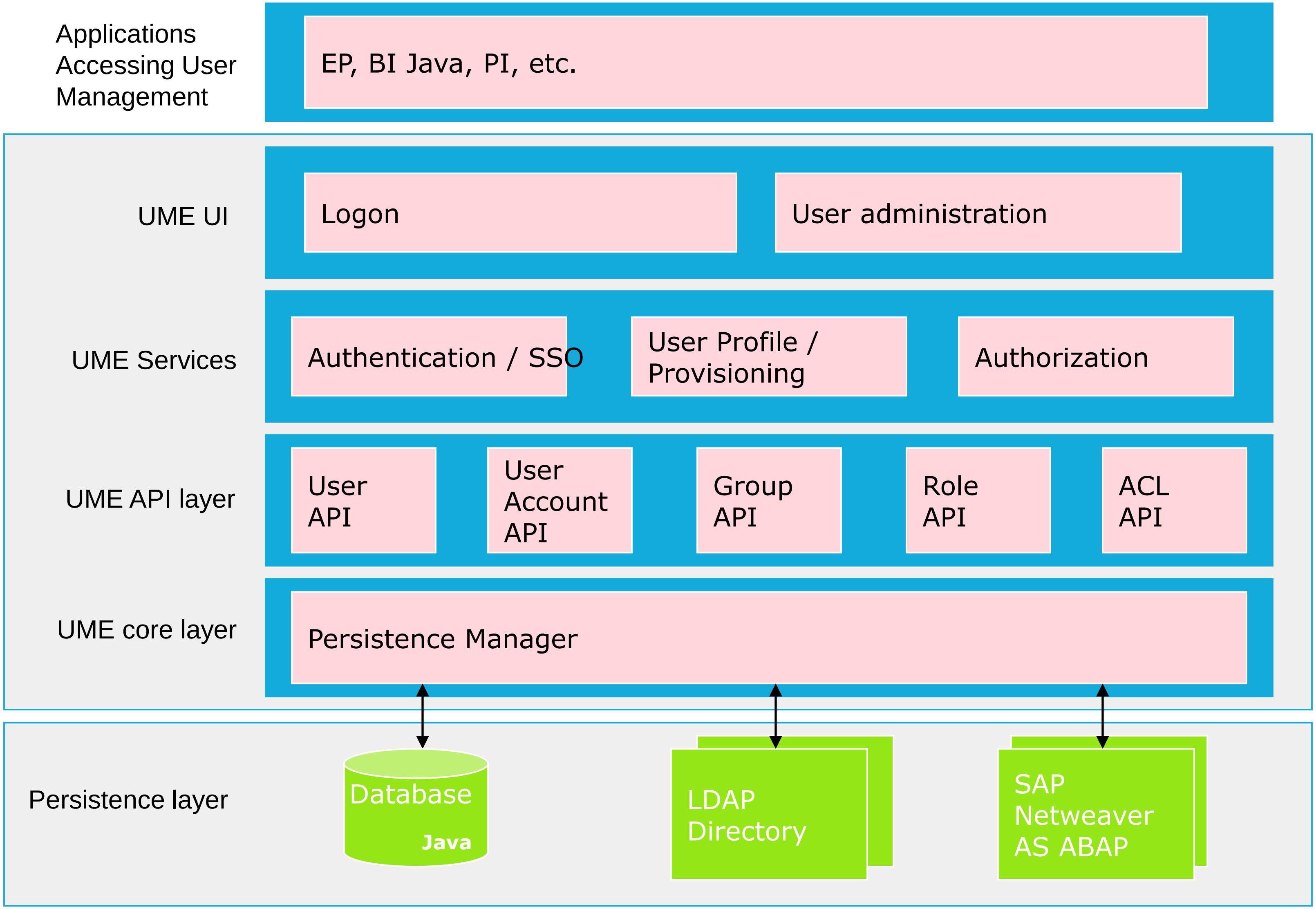
The user management engine (UME) provides a centralized user management for all Java applications and can be configured to work with user management data from multiple data sources. It is seamlessly integrated in the SAP NetWeaver Application Server (AS) Java as its default user store and can be administrated using the administration tools of the AS Java.

The UME runs as a service in the AS Java and is set up as the default user store.

UME Functional areas are,

- UME core layer
- UME API layer
- UME services
- UME UI

Architecture



Applications Accessing User Management

EP, BI Java, PI, etc.

UME UI

Logon

User administration

UME Services

Authentication / SSO

User Profile / Provisioning

Authorization

UME API layer

User API

User Account API

Group API

Role API

ACL API

UME core layer

Persistence Manager

Persistence layer

Database Java

LDAP Directory

SAP Netweaver AS ABAP

Features

User Administration – Identity management enables administrators to perform routine administration tasks such as creating or searching for users and groups, and assigning users and groups to roles. You can also configure the UME for e-mail notification, whereby e-mails are automatically sent to users or administrators on specific events, for example, if an administrator locks a user account, the user receives an e-mail informing him or her of the change.

Security Setting – You can define a password policy including settings such as minimum and maximum length of passwords, number of failed logons before a user is locked, and so on.

Self Service Scenarios – UME provides self-service scenarios that allow users to register themselves as new users or to change their own data (address, password, and so on). It is also possible to set up an approval workflow, whereby administrators approve newly registered users.

Security Logging – The UME logs important security events, such as successful and failed user logons, and creation or modification of users, groups, and roles.

Import & Export User Data – The UME enables you to import and export user data from and to external systems.

Virtual Groups – The UME enables you to define virtual groups based on the content of a user attribute.

Companies – Enables you to support delegated user administration

Tools for Configuration

UME Administration Console:

- You can use the UME Administration Console running in the web browser to modify selected settings without it being necessary to know the technical parameter names (path: URL /useradmin → Configuration).

Configuration Tool (Offline Configuration Editor Mode):

- Only in Offline Configuration Editor Modus are you able to access all the UME settings (path: cluster_data → server → cfg → services → Propertysheetcom.sap.security.core.ume.service).

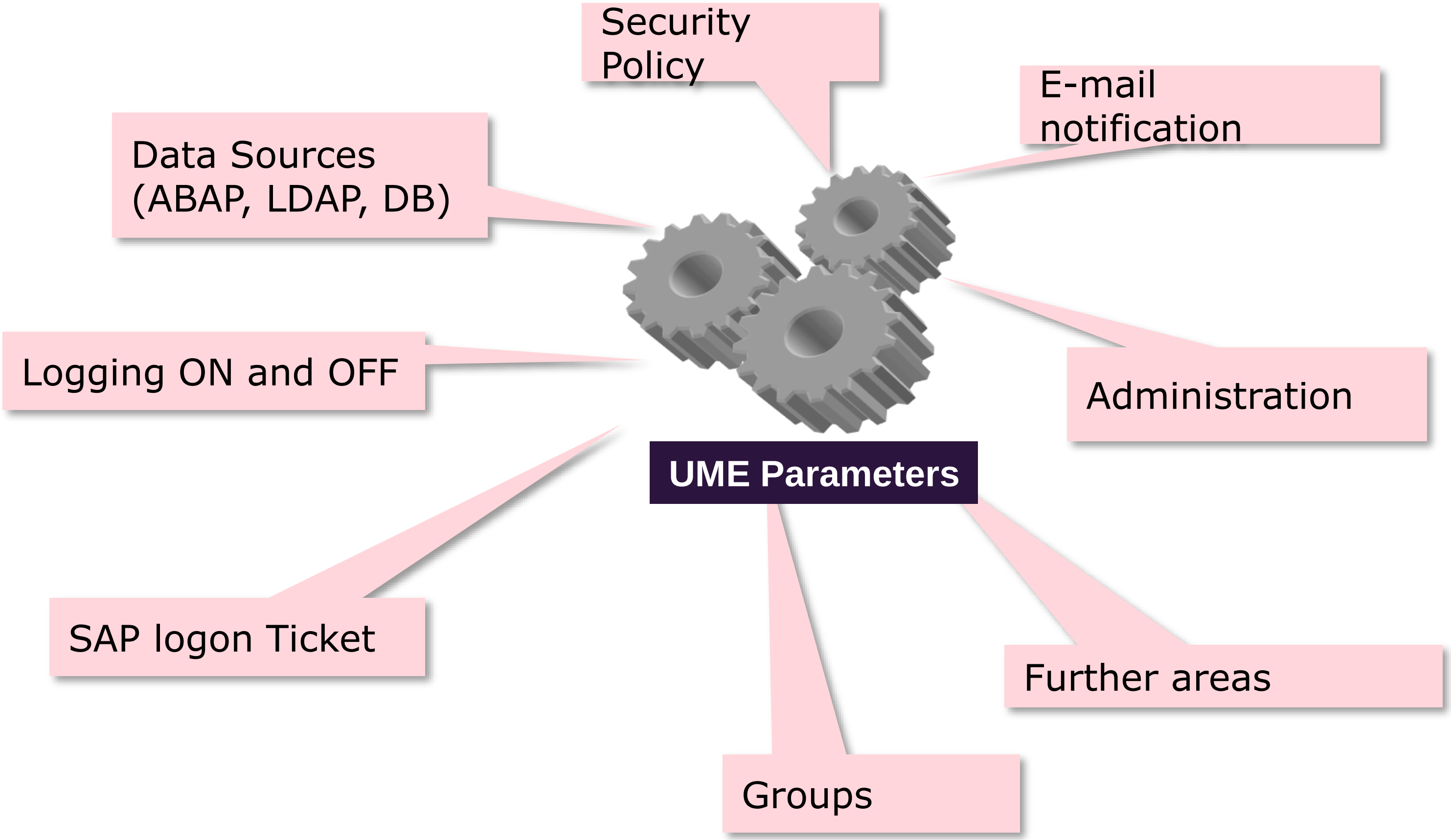
Configuration Tool (simple mode):

- In the Configuration Tool's simple mode, you will see an area in which you can make settings specially for the LDAP Server data source (path cluster_data → UME LDAP data).

UME Configuration iView :

- If the usage type EP Core has been installed in your SAP NetWeaver system, you can use the portal interface to access an iView for UME configuration. This offers similar setting options to the UME Administration Console (path System Administration → System Configuration → UME Configuration).

Parameters



UME Parameters

Security Policy

E-mail notification

Data Sources (ABAP, LDAP, DB)

Logging ON and OFF

Administration

SAP logon Ticket

Further areas

Groups

Parameters

- UME Data Sources
- Editing UME Properties
- Configuring the Security Policy for User ID and Passwords
- Configuring E-Mail Notification
- Configuring Self-Registration
- Configuring Self-Management
- Enabling Users to Reset Their Own Password
- Logon Screen Customization
- Configuring Delegated User Administration Using Companies
- Configuring Virtual Groups
- Adding Custom Attributes to the User Profile
- Allowing Users to View the Contact Information of Other Users
- Additional Customizing Options

Additional Information

For the most recent changes in the UME, see SAP Note 720590 User Management Engine (UME) on WAS 6.30 and higher.

For more information about configuring the UME, see UME Configuration. Here you can find information on configuring the data sources that UME uses to read and write user management data, and other configuration options.

For more information about administration with UME, see UME User Administration. The UME provides an administration console for performing administrative tasks such as searching for and creating users, groups, and roles.

For more information about troubleshooting and monitoring options for the UME, see Troubleshooting. This section also includes information about configuring the emergency user.

For reference material on the UME, see UME Reference. This includes information on the UME properties and configuration files.

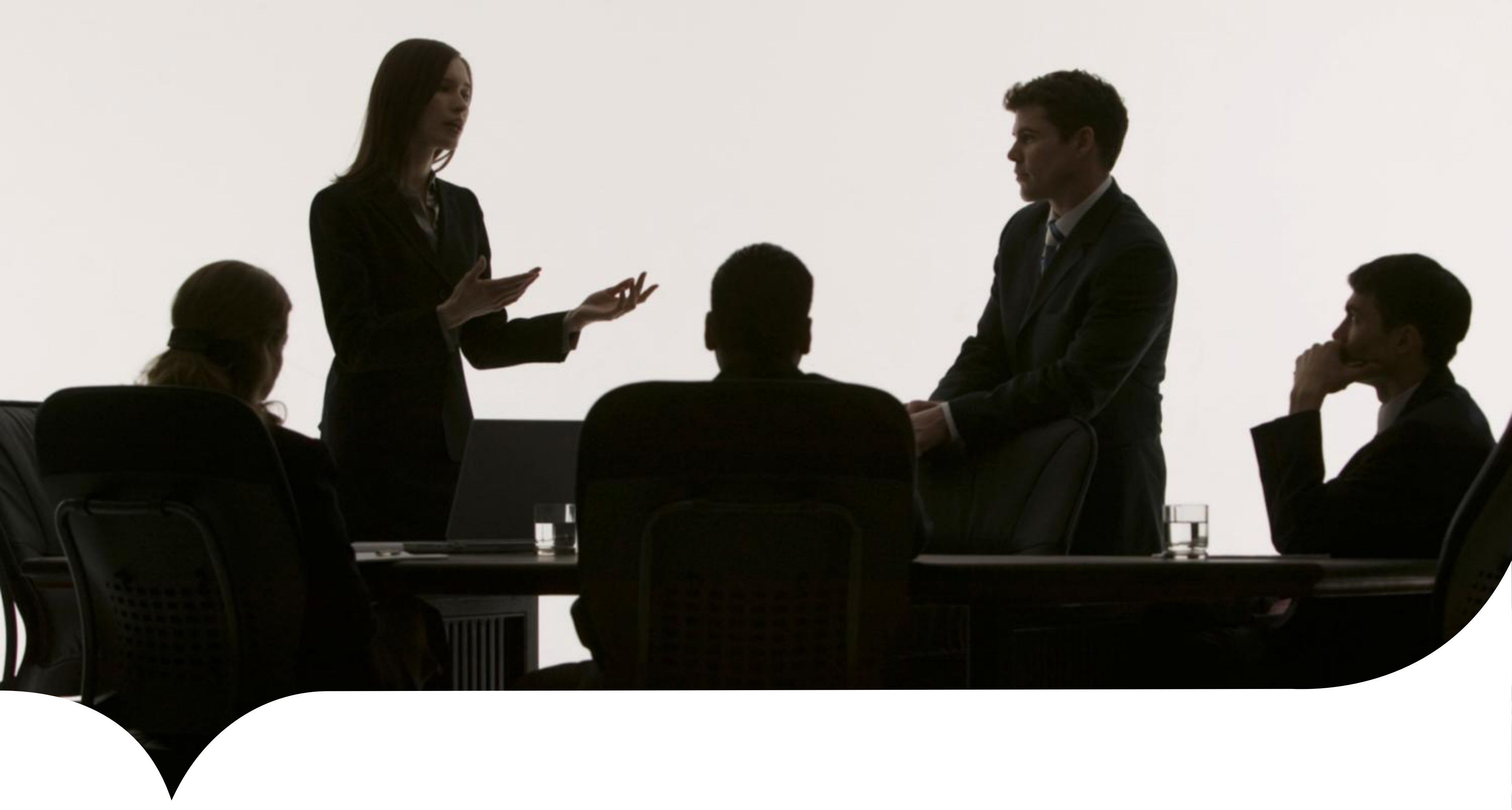
Summary

User Management in Java

- UME Concepts
- Managing Users, Groups, and roles
- Principles, Roles, Groups

Internet Communication

- ITS, ICM and ICF
- Web Dispatcher



Break out session

COLLABORATIVE BUSINESS EXPERIENCE