System Automation for z/OS

Performance Driven Automation

Jürgen Holtz
IBM Deutschland Entwicklung GmbH
Schönaicher Str. 220
71032 Böblingen
holtz@de.ibm.com

Copyright and Trademarks

The following names are trademarks of the IBM Corp. in USA and/or other countries and may be used throughout this presentation:

CICS, DB2, eLiza, IBM, IMS, MVS/ESA, MQSeries, NetView, OMEGAMON, RMF, RACF, S/390, Tivoli, VTAM, VSE/ESA, VM/ESA, WebSphere, z/OS, z/VM, zSeries

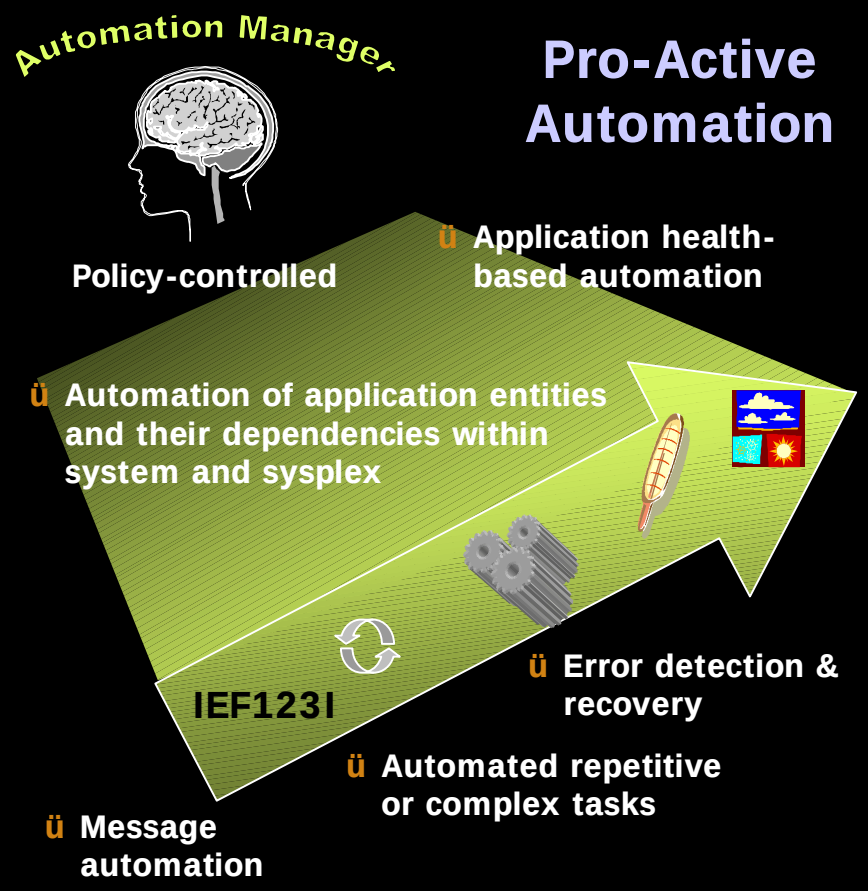
Other company, product and service names may be trademarks or service marks of others.

Agenda

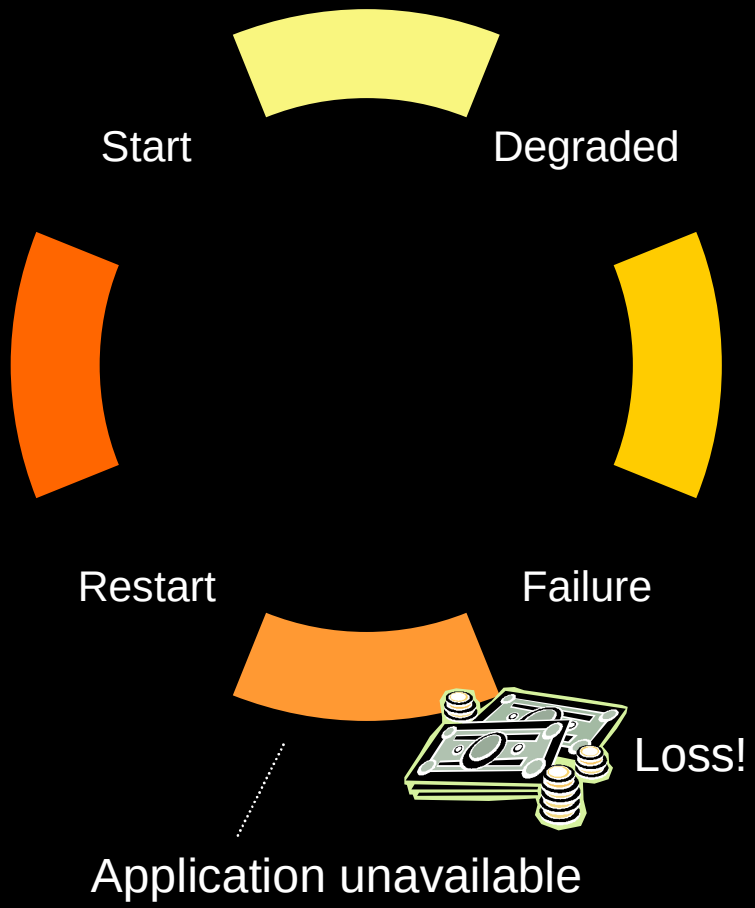
- **Motivation**
- **Resource / Exception Monitoring using OMEGAMON**
- **Exception-based Automation**
- **Monitor Resources**
  - Concepts
  - Administration and Operation
- **Health-based Automation**
- **Summary**

Automation Evolution

- **Message filtering**
- **Message automation**
- **Error detection and recovery**
- **Resource management**
  - Start, stop, recycle
  - Dependencies between resources
- **High availability for business processes**
- **Autonomic computing**
  - Understanding health of system and applications
  - Pro-active automation



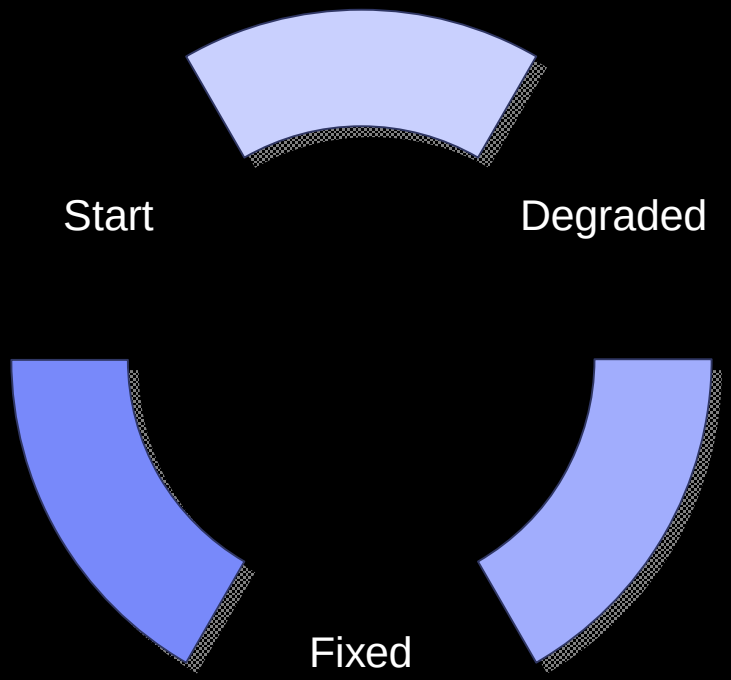
Application Life Cycle w/o Health Monitoring



- **Application state is either up or down**
- **Gradients between up and down are unknown**
- **An outage may occur when a degraded application is detected too late**
- **Damage due to outages can be measured in '$'s**

è It is important to avoid or at least reduce application repair time to achieve higher availability

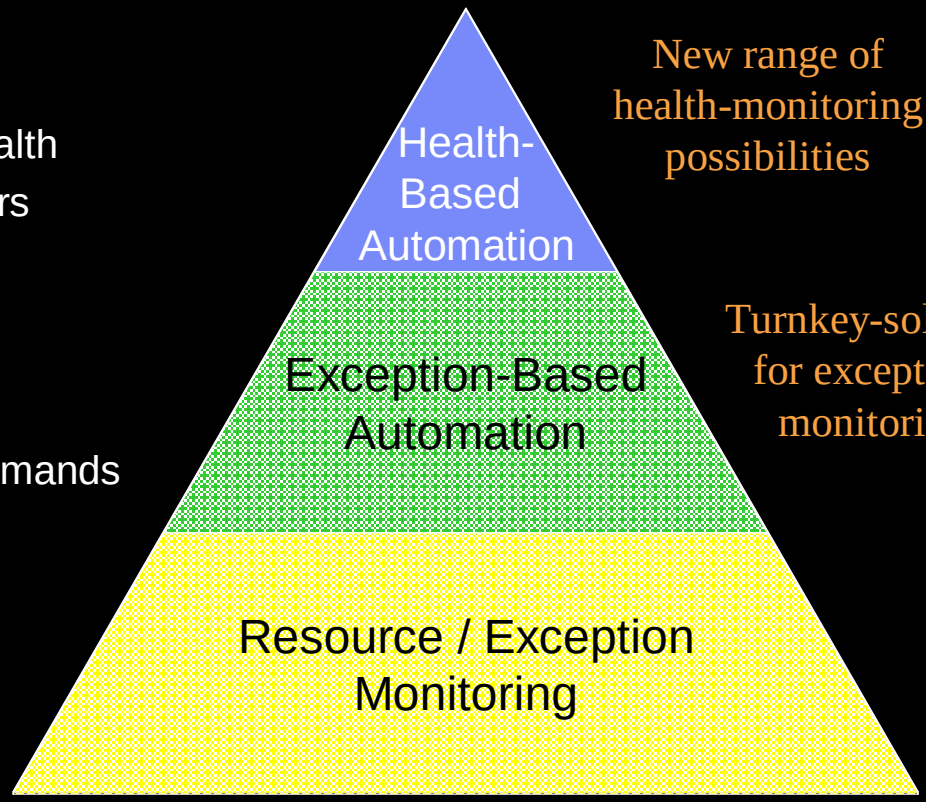
Application Life Cycle with Health Monitoring



- **Ability to detect degraded health states**
- **Possible reactions**
  - Elimination of bottlenecks
  - Provisioning of additional resources
  - Consider pro-active application move
  - Prepare for “planned” outage
- **Goal: fix the problem before a failure occurs**

How does this Relate to Automation?

- **System Automation for z/OS**
  - Monitor Resource concept
  - Determination of application health
  - Ability to act before failure occurs
- **SA z/OS and AF/OPERATOR**
  - OMEGAMON Classic interface
  - Exception Monitoring
  - Execution of OMEGAMON commands
- **OMEGAMON Classics**
  - CICS
  - DB2
  - IMS
  - MVS



New range of health-monitoring possibilities

Turnkey-solution for exception-monitoring

HW and SW performance and availability data

Agenda

- **Motivation**
- **Resource / Exception Monitoring using OMEGAMON**
- **Exception-based Automation**
- **Monitor Resources**
  - Concepts
  - Administration and Operation
- **Health-based Automation**
- **Summary**

OMEGAMON Exception Monitoring …

- **OMEGAMON LEXSY-command triggers exception analysis for**
  - System-wide exceptions, e.g. XCSA for common storage area utilization
  - Address space exceptions, e.g. WAIT for address space wait times
- **Exceptional conditions are calculated based on internal OMEGAMON cycles**
- **Example:**

```
LEXSY     OMEGAMON/MVS Exception Analysis
+ XREP Number of Outstanding Replies = 6
+ FXFR STC *MASTER*    | Fixed Frames in use = 2937
+ WSHI     *MASTER*    | Working Set Size = 12592K (High)
+ FXFR STC PCAUTH      | Fixed Frames in use = 88
+ WAIT     PCAUTH      | Wait:  8:04 DY
```

OMEGAMON Exception Monitoring *(cont.)*

- **Exception thresholds can be set and displayed with the XACB command, e.g.**

```
 XACB LIST=XCSA
: XCSA
+       DISPLAY Parameters:    THRESHOLD Parameters:    XLF Parameters:
:         State=ON              Threshold=85             Auto=OFF
:         Group=OP              Display=CLR2             Log=OFF
:         Bell=OFF              Attribute=NONE           Limit=0 (0)
:       BOX Parameters:       CYCLE Parameters:          Repeat=NO
:         Boxchar='+'           ExNcyc=0                 Persist=0
:         Boxclr=CLR2           Stop=0 (0)               SS=
:         Boxattr=NONE          Cumulative=0
```

- **In the example above, the setting for XCSA indicates that an exception is reported for CSA utilization > 85%**

Agenda

- **Motivation**
- **Resource / Exception Monitoring using OMEGAMON**
- **Exception-based Automation**
- **Monitor Resources**
  - Concepts
  - Administration and Operation
- **Health-based Automation**
- **Summary**

Exception-Based Automation using AF/OPERATOR

- **Logon to particular OMEGAMON session(s)**

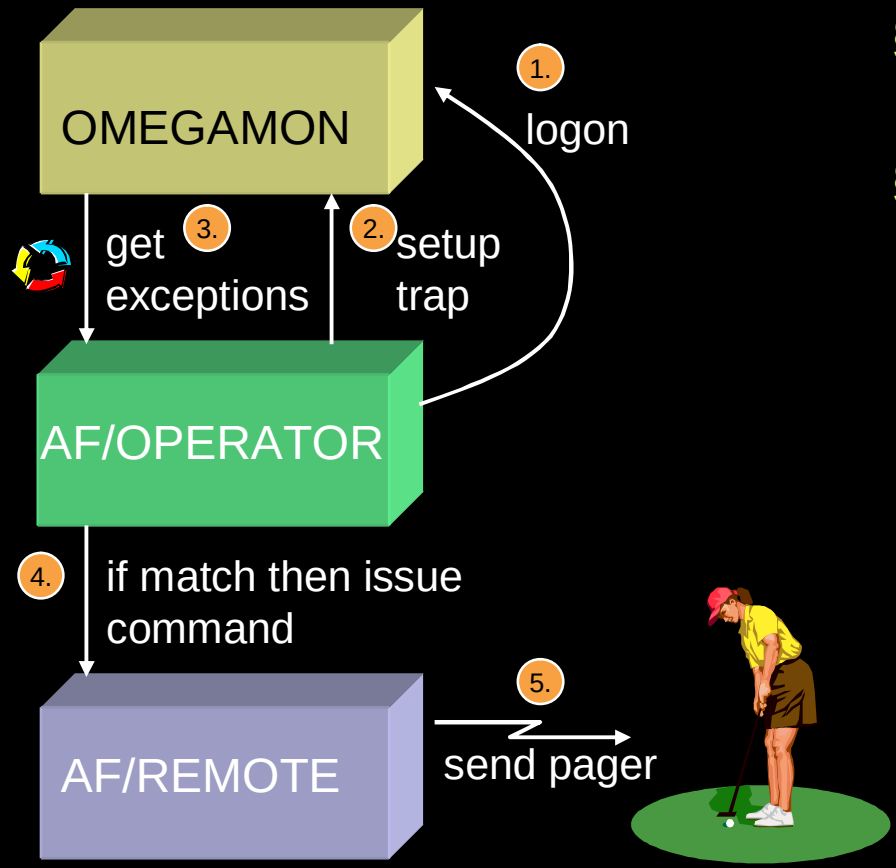
```
LOGON OMMVS APPLID(ipspm2rc) NAME(omkey4)
USERID(holtz) PASSWORD(mypw/cl3pw) INTERVAL(00:00:30)
```

- **Setup one or more traps for exceptions of interest, for example**

```
TRAP ADD(csatrap) XOM('+ XCSA *') ENABLE NOLOG
ACTION('ex donotify')
```

- **Characteristics**
  - Above trap will poll the OMEGAMON sessions every 30 seconds
  - When an exception occurs, the specified REXX script *donotify* is invoked
  - Other options
    - Selection of particular OMEGAMON through SESSION-keyword
    - Specification of an alternate command ALTACT that is issued at every n-th occurrence of an exception using MATCHLIM(n)

Exception Alarming via AF/REMOTE

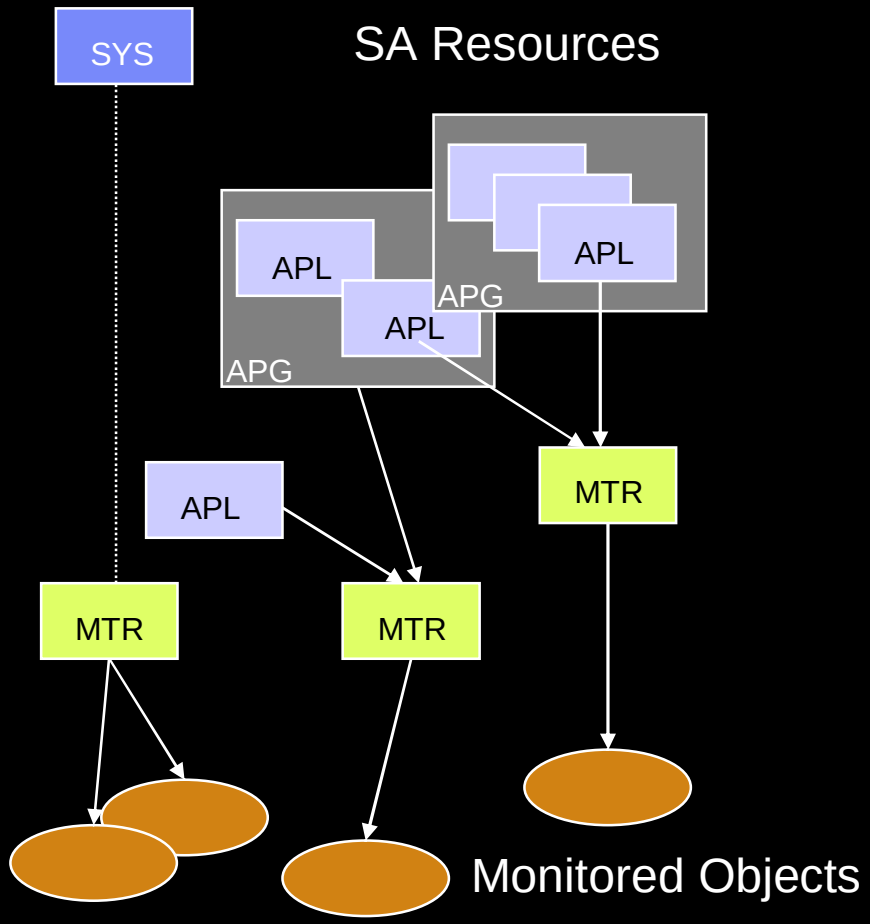


- **Scenario: Monitoring and Alarming**
- **Components**
  - OMEGAMON exception thresholds are defined
  - AF/OPERATOR establishes session with OMEGAMON to trap interesting exceptions
  - AF/REMOTE script is defined to send a pager or an e-mail when triggered by automation

Agenda

- **Motivation**
- **Resource / Exception Monitoring using OMEGAMON**
- **Exception-based Automation**
- **Monitor Resources**
  - Concepts
  - Administration and Operation
- **Health-based Automation**
- **Summary**

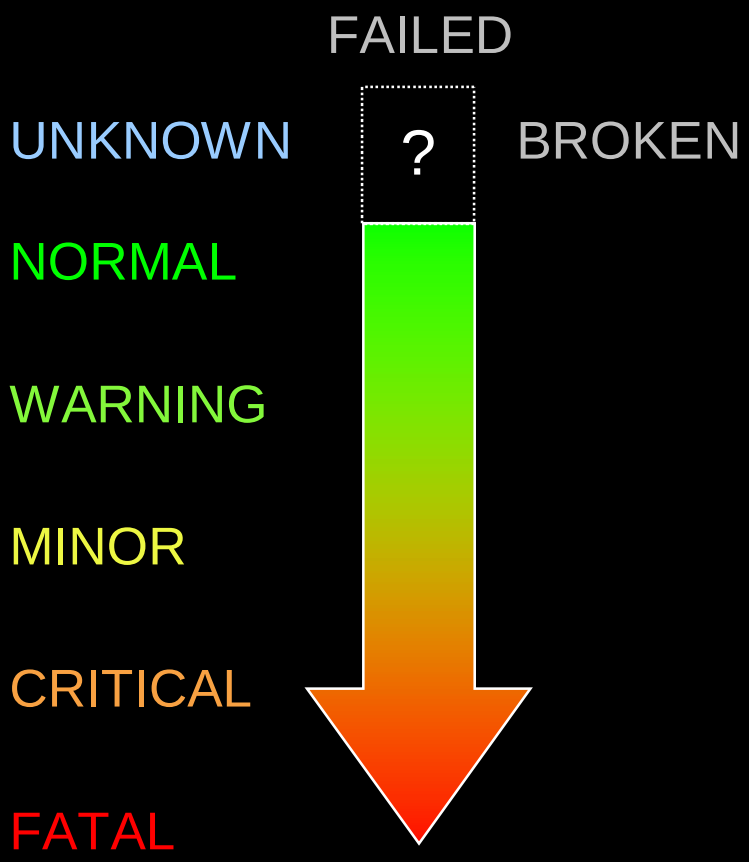
Monitor Resources – At a Glance



- **Resource in the automation policy**

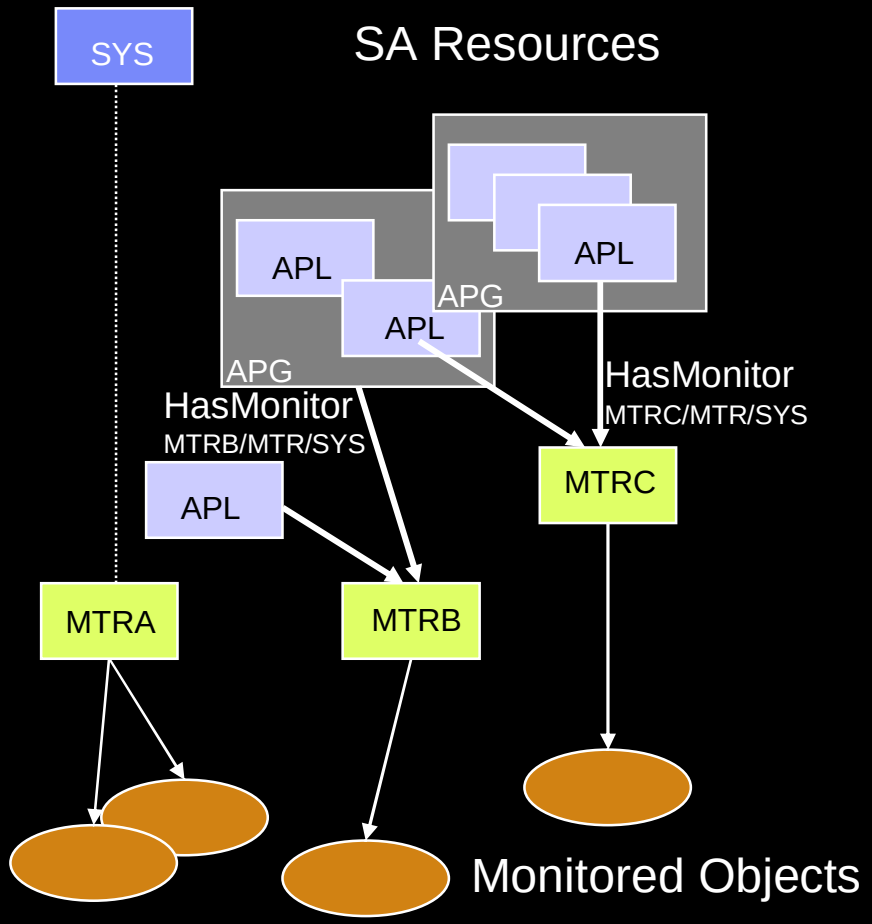
  Name: *monitor*/MTR/*system*
- **Obtains and holds health state of the object it monitors (job, device, file system, etc.)**
- **Typically associated with an application (APL) or application group (APG)**
- **Health state**
  - Obtained either periodically or based on an event
  - Propagated to associated APL and APG

Health States



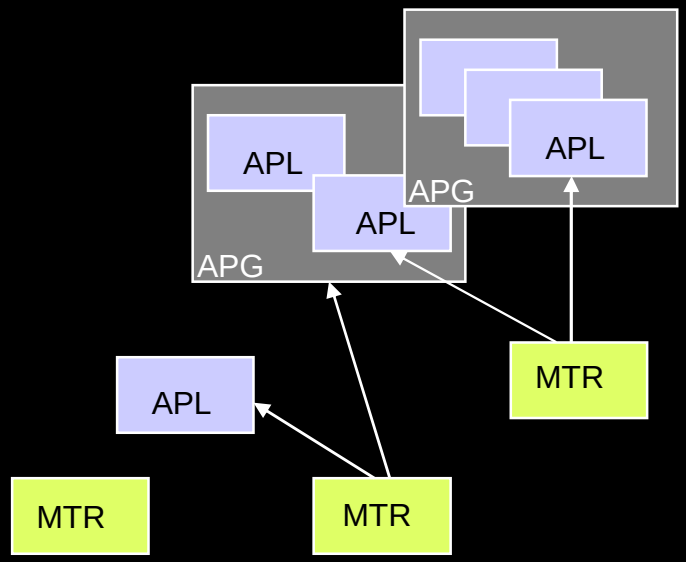
- **The MTR determines an health state based on its observations**
  - 5 regular health states: NORMAL, WARNING, MINOR, CRITICAL, and FATAL
  - UNKNOWN: health state has not yet been determined
  - FAILED: MTR failed and will be rescheduled
  - BROKEN: MTR failed and monitoring stopped
- **The health state is tracked by the automation manager**
- **The automation manager**
  - Propagates the health state to resources related to the MTR
  - Computes an accumulated health state
  - Triggers actions, if specified in the automation policy based on individual health state

HasMonitor Relationship



- **MTR is connected from APL or APG via *HasMonitor* relationship**
- **One MTR can be connected to zero or more APLs/APGs**
- **One APL/APG can have zero or more MTRs connected**
- **MTRs cannot be members of APGs and cannot have other MTRs**

Health Status Accumulation



- **Health states are accumulated by the automation manager**
  - Over all MTRs
  - Over all group members
  - Over multiple group nesting levels, if required
- **General rule: most severe health state counts**
- **Health status is ‘N/A’ for APLs or APGs without MTR**

Compound State

- The compound state is the result of the aggregation of the six resource states managed by the automation manager
- A compound state PROBLEM propagated to an APG can trigger automation manager decisions for MOVE and SERVER groups
- MTRs

Desired Status	Observed Status	Health Status	Compound Status
AVAILABLE	AVAILABLE	UNKNOWN,NORMAL	SATISFACTORY
AVAILABLE	AVAILABLE	WARNING,MINOR,CRITICAL	DEGRADED
AVAILABLE	AVAILABLE	FATAL	PROBLEM
UNAVAILABLE	SOFTDOWN	N/A	SATISFACTORY

- APLs/APGs

Desired Status	Observed Status	Health Status	Compound Status
AVAILABLE	AVAILABLE	N/A (no MTRs)	SATISFACTORY
AVAILABLE	AVAILABLE	UNKNOWN,NORMAL	SATISFACTORY
AVAILABLE	AVAILABLE	WARNING,MINOR,CRITICAL	DEGRADED
AVAILABLE	AVAILABLE	FATAL	PROBLEM
UNAVAILABLE	SOFTDOWN	N/A (no MTRs)	SATISFACTORY
UNAVAILABLE	SOFTDOWN	UNKNOWN,NORMAL	SATISFACTORY
UNAVAILABLE	SOFTDOWN	WARNING,MINOR,CRITICAL	SATISFACTORY
UNAVAILABLE	SOFTDOWN	FATAL	INHIBITED

Monitor Resource Life Cycle

e.g. Mo-Fr, 08:00-16:00

1. *monitor*
2. *determine health state*
3. *capture message*

- **Default: MTR is always active unless something else is specified**
- **MTR can be connected to a service period defining the window of availability**
- **An MTR can be activated or deactivated when the supporting resource is made available or unavailable, respectively**
- **Upon activation, an optional activation command can be issued for setup**
- **While the MTR is active**
  - Monitor command Is issued
  - Health state is determined
  - Message explaining health state is captured
- **Upon deactivation, an optional deactivation command can be issued for cleanup**

Active Monitor Resource

- **An active MTR runs periodically according to interval specified in customization dialog**
- **Health state is determined based on polling the monitored object(s)**
- **Simple example: Test of network connection to some TCP/IP host**

```
/* REXX */
parse arg ipHost
Rc_Normal = 3
Rc_Fatal  = 7

If CNMEPING('-q' ipHost) = 1 then
  lrc = Rc_Normal
else
  lrc = Rc_Fatal

return lrc
```

Passive Monitor Resource

- **An MTR is passive if no interval is specified in the customization dialog**
- **A passive MTR determines health state based on events coming from the monitored object(s) à messages**
- **Health state must be updated in response to such messages using the generic command INGMON**
- **Simple example: MTR JES2MON is monitoring $HASP9202 issued by JES2**
  - Meaning: Potential JES2 main task loop
  - NetView automation table snippets created automatically based on policy definition:

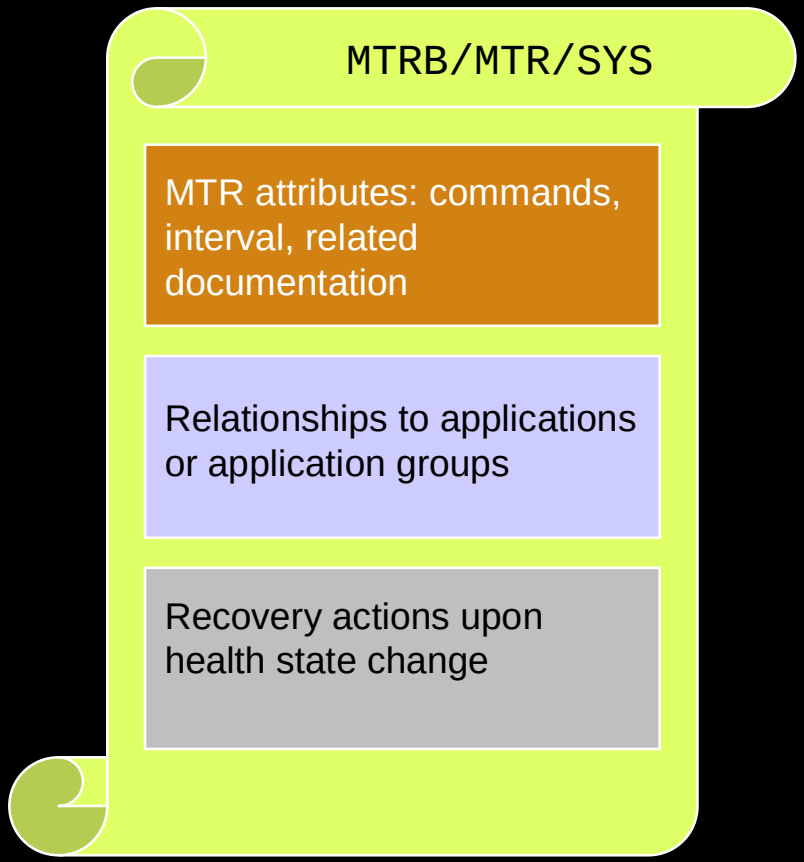
New with SA z/OS V3.1

```
NetView AT condition. . . . . .
MSGID = '$HASP9202'


NetView AT action 1 . . . . . .
EXEC(CMD('INGMON JES2MON STATUS=CRITICAL') ROUTE(ONE %AOFOPJESOPER%))
```

Agenda

- **Motivation**
- **Resource / Exception Monitoring using OMEGAMON**
- **Exception-based Automation**
- **Monitor Resources**
  - Concepts
  - Administration and Operation
- **Health-based Automation**
- **Summary**

Defining a Monitor Resource



- **MTRs are defined in the customization dialog under policy entry MTR**
- **A definition consists of**
  - Attributes such as commands, a polling interval, size of historical message log
  - Relationships to other resources
  - Recovery actions driven by SA upon detected change of health state
- **MTR may be assigned an optional service period (SVP)**
- **One or more systems must be selected where the MTR is to be instantiated**

Monitor and Activate / Deactivate Commands

- **All commands are executed in NetView PIPE with EXPOSE COMMAND stage**
- **Command echo and any message produced are written to NETLOG**
- **Return codes**
  - Activate and deactivate commands: RC=0, otherwise MTR state is BROKEN
  - Monitor commands: (see table below)

Return Code	Health State	Monitor State	Meaning
1	UNKNOWN	BROKEN	Unrecoverable error. SA stops monitoring the object
2	UNKNOWN	FAILED	Temporary failure. SA continues monitoring the object
3	NORMAL	ACTIVE	Normal operation of the monitored object
4	WARNING	ACTIVE	Same as NORMAL but with some degradation
5	MINOR	ACTIVE	Same as WARNING but more severe
6	CRITICAL	ACTIVE	Same as MINOR but more severe
7	FATAL	ACTIVE	Same as CRITICAL but more severe
8	“DEFER”	ACTIVE	Deferred processing of health state within automation table

Recovery Activities

- **MTR definitions can hold commands that are executed once**
  - When the health state changes (no health state specified)
  - When the health state changes to the specified value
- **If there are multiple commands for one health status, the commands are executed in the sequence specified**
- **Example: JES3 SPOOL monitoring using MTRs**

```
Healthstate    Automated Function/''*'
Command Text
FATAL          JESOPER
AOFRJ3RC JES3 SPOOLSHORT 00:01

NORMAL         JESOPER
AOFRJ3RC JES3 SPOOLSHORT RESET
```

Change to FATAL: Issues commands grouped by PASSes specified in SPOOLSHORT policy within APL JES3. Increment PASS-count every minute.

Change to NORMAL: reset PASS count and stop SPOOL-recovery.

Operating MTRs from NCCF

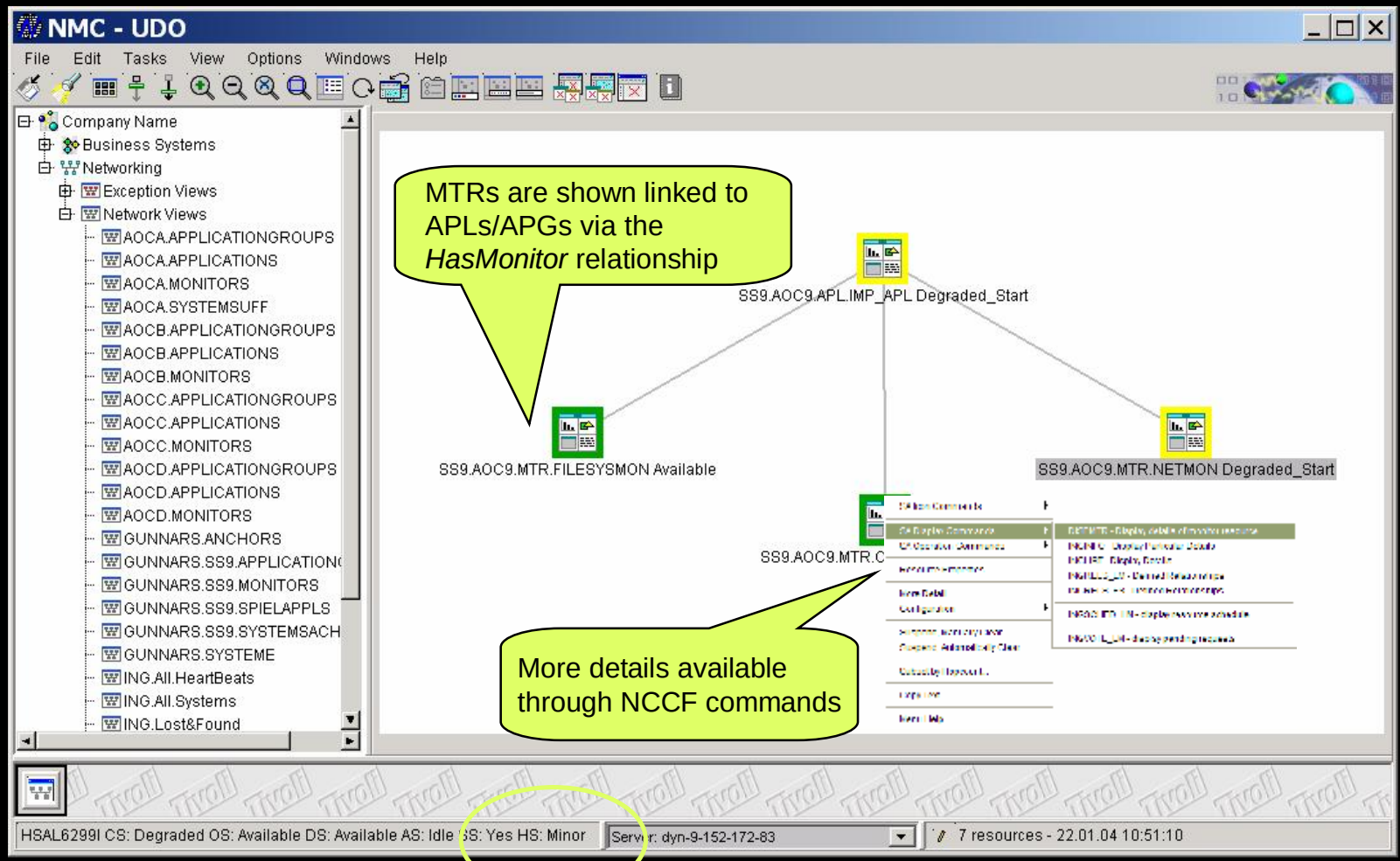
- **INGLIST lists all resources including health state (scroll right)**

```
INGKYST0                    SA z/OS  - Command Dialogs     Line  1     of 5
 Domain ID    = IPUN9      -------- INGLIST   ---------     Date = 03/23/05
 Operator ID = BHOL             Sysplex = SYSPLEX1          Time = 08:44:09
 CMD: A Update    B Start     C Stop      D INGRELS   E INGVOTE   F INGINFO
      G Members   H DISPTRG   I INGSCHED  J INGGROUP  M DISPMTR   / scroll
 CMD Name         Type System    Compound      Desired      Observed    Nature
 --- ------------ ---- --------  ------------  -----------  ----------  --------
  _  APLGROUP     APG  AOC9      SATISFACTORY  AVAILABLE    AVAILABLE   BASIC
  M  APLMON1      MTR  AOC9      SATISFACTORY  AVAILABLE    AVAILABLE
  _  APLMON2      MTR  AOC9      SATISFACTORY  AVAILABLE    AVAILABLE
```

- **DISPMTR displays detailed information about a monitor and the reason for the current health state**

```
INGKYMO0                    SA z/OS  - Command Dialogs     Line  1     of 1
Domain ID    = IPUN9      -------- DISPMTR   ---------     Date = 03/23/05
Operator ID = BHOL             Sysplex = AOC9PLEX          Time = 08:40:38

CMD: A Reset   B Start   C Stop   D Details   E INGVOTE  F INGINFO  I INGSCHED

CMD Monitor      System    Status        Health       Last monitored
--- ------------ --------  ------------  ------------ -------------------
 _  APLMON1      AOC9      ACTIVE        NORMAL       2005-03-23 08:40:10
```

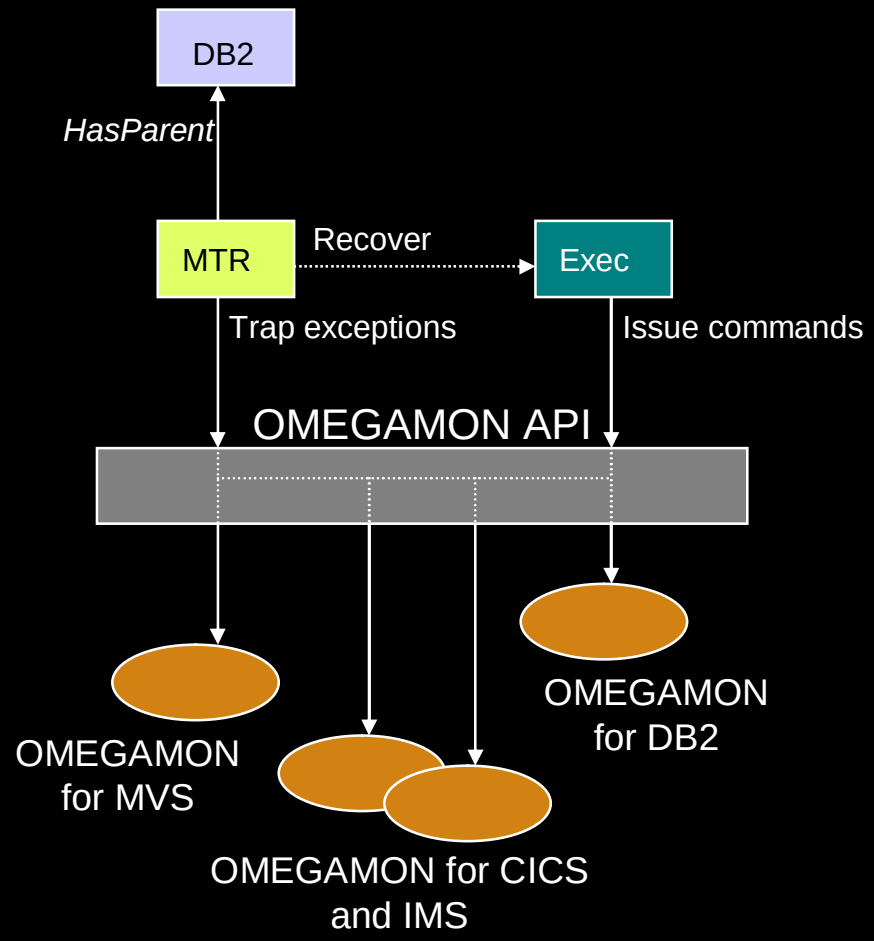
Operating MTRs from NMC



Agenda

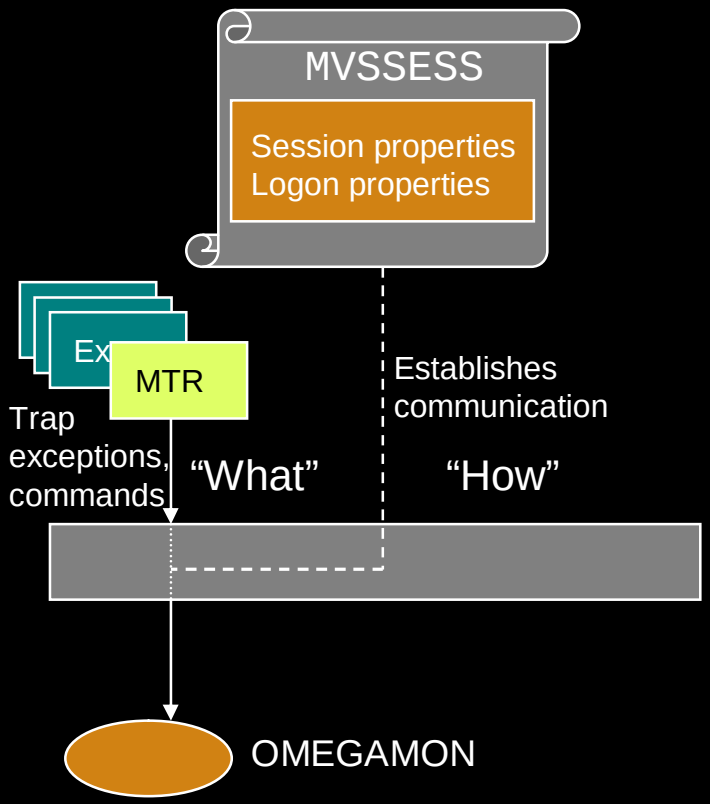
- **Motivation**
- **Resource / Exception Monitoring using OMEGAMON**
- **Exception-based Automation**
- **Monitor Resources**
  - Concepts
  - Administration and Operation
- **Health-based Automation**
- **Summary**

SA OMEGAMON Interoperation – Value



- **Use of performance and availability information for application automation**
  - More facts, more accurate decisions
  - Sources: MVS, DB2, CICS, IMS
- **Provides API to communicate with OMEGAMON monitors to**
  - Obtains and filters installation-defined exceptional conditions
  - Sends commands to OMEGAMON, for example to respond to such conditions
- **Provides exception monitor based on the Monitor Resource concept**
  - Monitors „interesting“ set of exceptions
  - Sets application health state based on existence of such exceptions
  - Provides means to react and resolve exceptional conditions

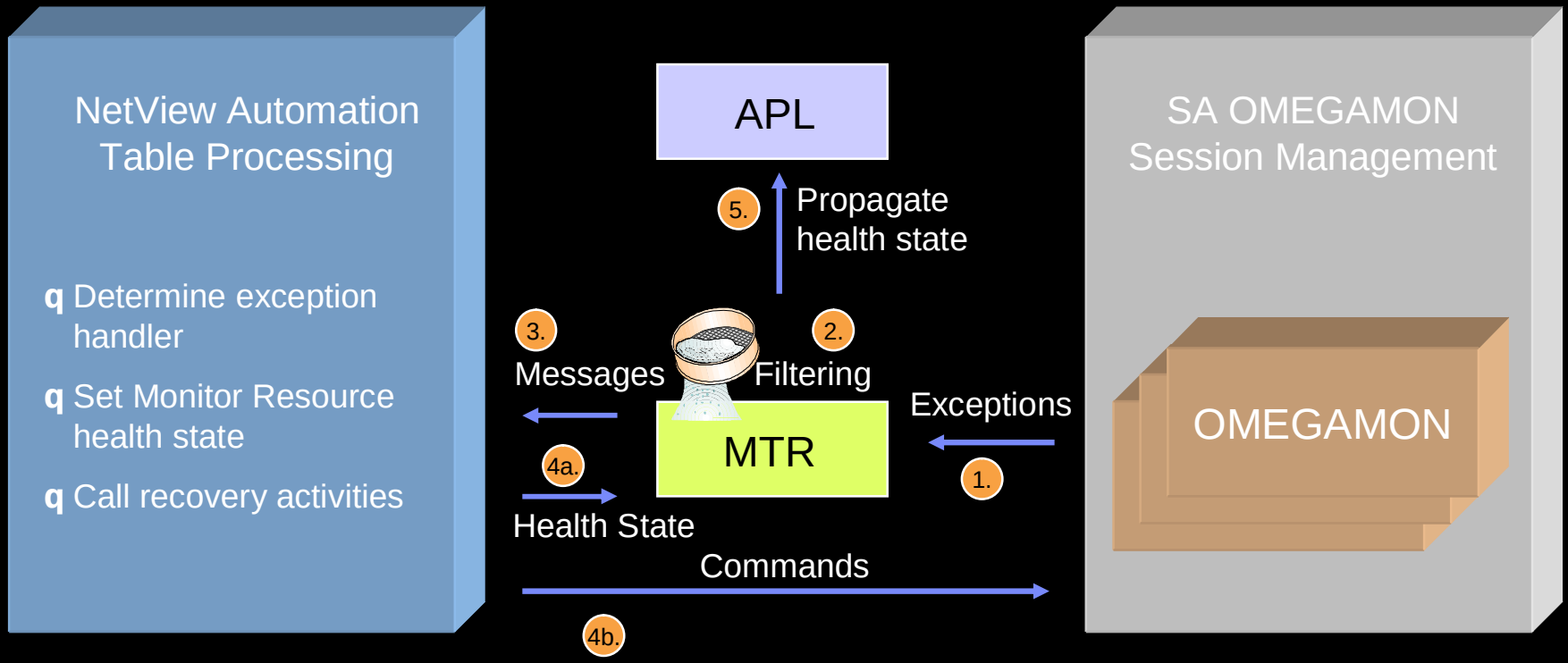
SA OMEGAMON Sessions



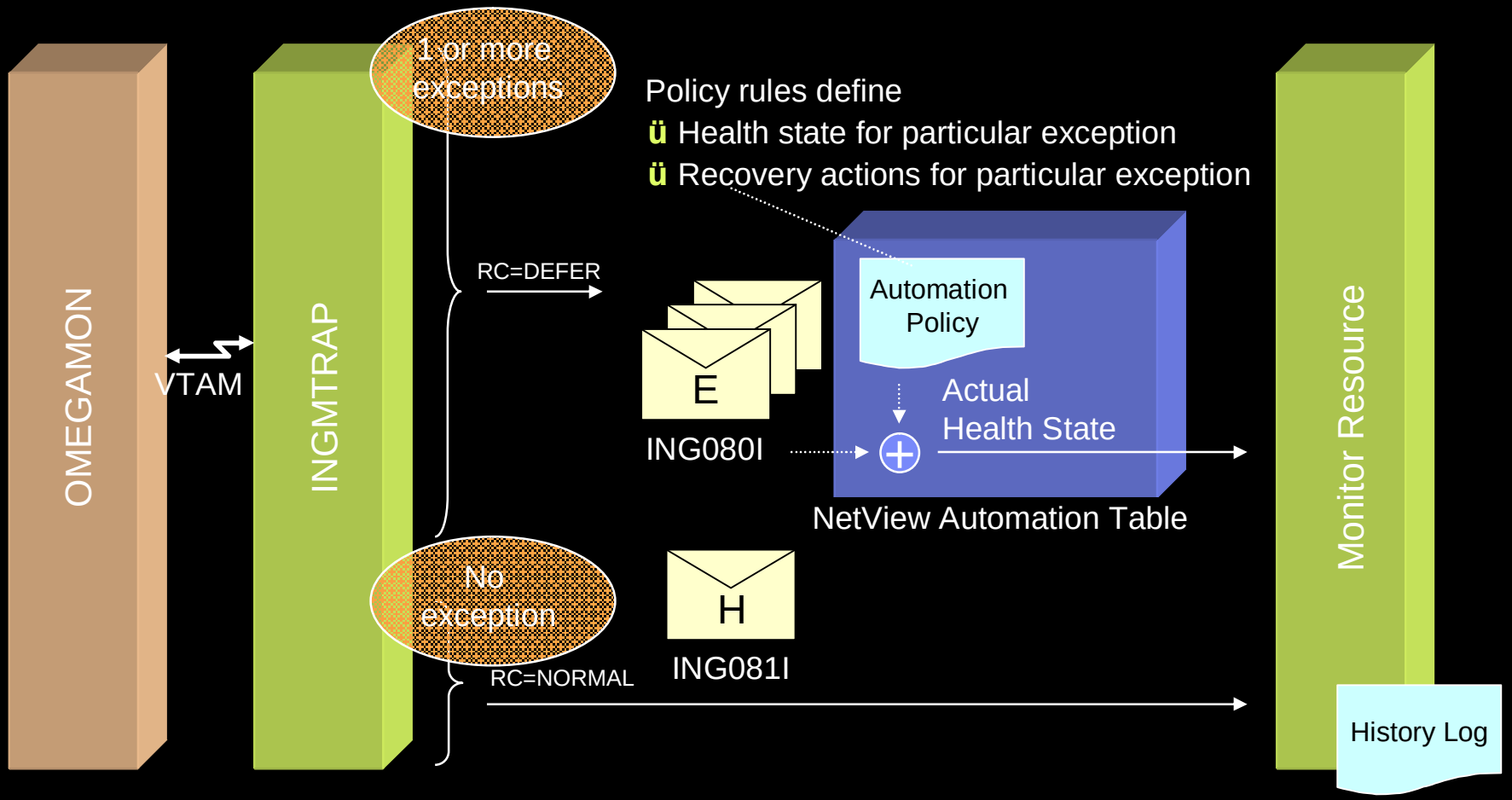
- **OMEGAMON sessions are defined as policy items in the network policy (NTW)**
- **A definition consists of**
  - Session attributes to identify and control VTAM session
  - User attributes to enable logon
- **A session can be used by multiple operators**
  - Automation operators, for example running Monitor Resource commands
  - Human operators
- **Separate automation operators are reserved to control one or more sessions**

Exception Monitoring Architecture

- **Active MTR is used to periodically retrieve OMEGAMON exceptions**
- **Health state processing and recovery will be driven via the NetView automation table created out of the SA policy**



From an Exception to a Health State



System Automation OMEGAMON API

- **Command INGOMX serves as interface between operators and a particular OMEGAMON session**
- **Possible interactions**
  - Call OMEGAMON exception analysis and find interesting exceptions
  - Enter one or more OMEGAMON commands, for example to collect additional performance information or to remove a bottleneck
- **Monitor command INGMTRAP serves as a customized interface to INGOMX primarily intended to**
  - Find interesting exceptions in the context of a monitor command
  - Drive NetView automation table processing to set application health state and for recovery

Exception-Monitoring using System Automation for z/OS

- **Define a Monitor Resource that periodically issues INGMTRAP, e.g.**

```
INGMTRAP TRAP,NAME=omsy4mvs,XTYPE=XCSA
```

- **Define an exception entry within the MESSAGES/USER DATA policy for the Monitor Resource, e.g.**

```
+ XCSA
```

à issue command *donotify* <parms>

- **When exception trips, a message like below is generated**

```
ING080I MYMON/MTR/KEY4 OMSY4MVS OMIIMVS + XCSA
Warning: Allocated CSA = 44%  (1428K out of 3264K)
```

- **Characteristics**
  - Each time monitor command is executed, exception analysis is done
  - Within the automation policy you can also set a health state and define a series of commands for escalation or define different sets of commands depending on exception text
  - Exception handling can be disabled while recovery is in progress

Issue OMEGAMON Commands from NetView Console

- **OMEGAMON commands can be issued from the NetView console using INGOMX, e.g.**

```
INGOMX EX,NAME=omsy4mvs,CMD=csaa
```

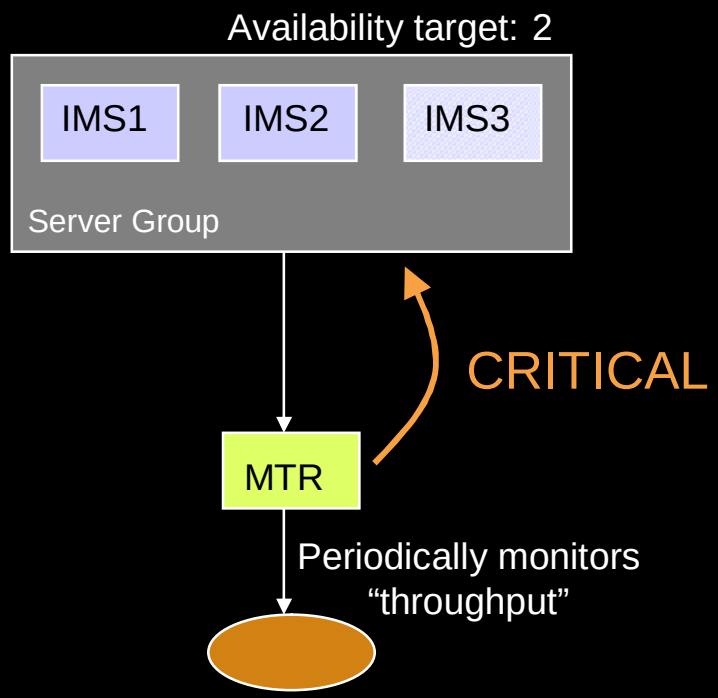
```
 CSAA    SUMMARY
+
+           System
+           Maximum   Pre-CSAA    Orphan        Usage
+           -------   --------  -------  ----------------0___2___4___6___8___100
+     CSA     3264K     1287K         0     1287K   39.4%|------>               |
+    ECSA   307336K    76925K         0    76925K   25.0%|---->                 |
+     SQA     1672K      604K         0      604K   36.1%|------>               |
+    ESQA   144892K    22834K         0    22834K   15.8%|-->                   |
```

OMEGAMON Session Management

- **INGSESS is the operator command to manage OMEGAMON sessions**
  - Start sessions manually to test connection and authorization
  - Stop sessions to do maintenance
  - Show additional session attributes, e.g. logon data, timeout, statistics

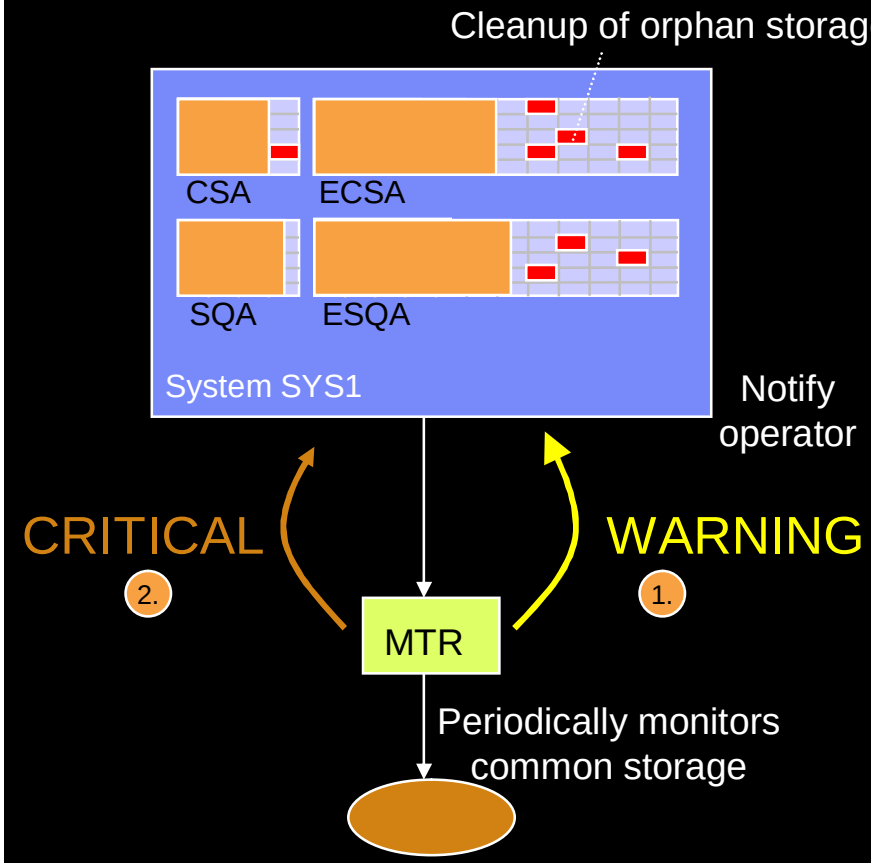
```
INGKYSS0                    SA z/OS  - Command Dialogs     Line  1     of 8
Domain ID   = IPUN9        -------- INGSESS   ---------    Date = 03/23/05
Operator ID = BHOL              System  = AOC9              Time = 08:08:56

CMD:  B Start session   C Stop session   D Details

CMD Session     System   Type     Status     Appl-id  User id  SessOper
--- ----------- -------- -------- ---------- -------- -------- --------
 _  CICSKY41    OMIICICS AOC9     ACTIVE     IPSPOC0  SAOM     AOFSES01
 _  DB2SGG4     OMIIDB2  AOC9     INACTIVE   IPSPD2C  SAOM     AOFSES02
 _  DB2SG14     OMIIDB2  AOC9     MAINT      IPSPD2C  SAOM     AOFSES03
 _  IMS742CR    OMIIIMS  AOC9     INACTIVE   IPSPOI0  SAOM     AOFSES01
 _  OMSY4MVS    OMIIMVS  AOC9     AUTHFAIL   IPSPM2RC SAOM     AOFSES02

Command ===>
 PF1=Help    PF2=End      PF3=Return                          PF6=Roll
                          PF9=Refresh                       PF12=Retrieve
```

Sample Scenario: Application Provisioning



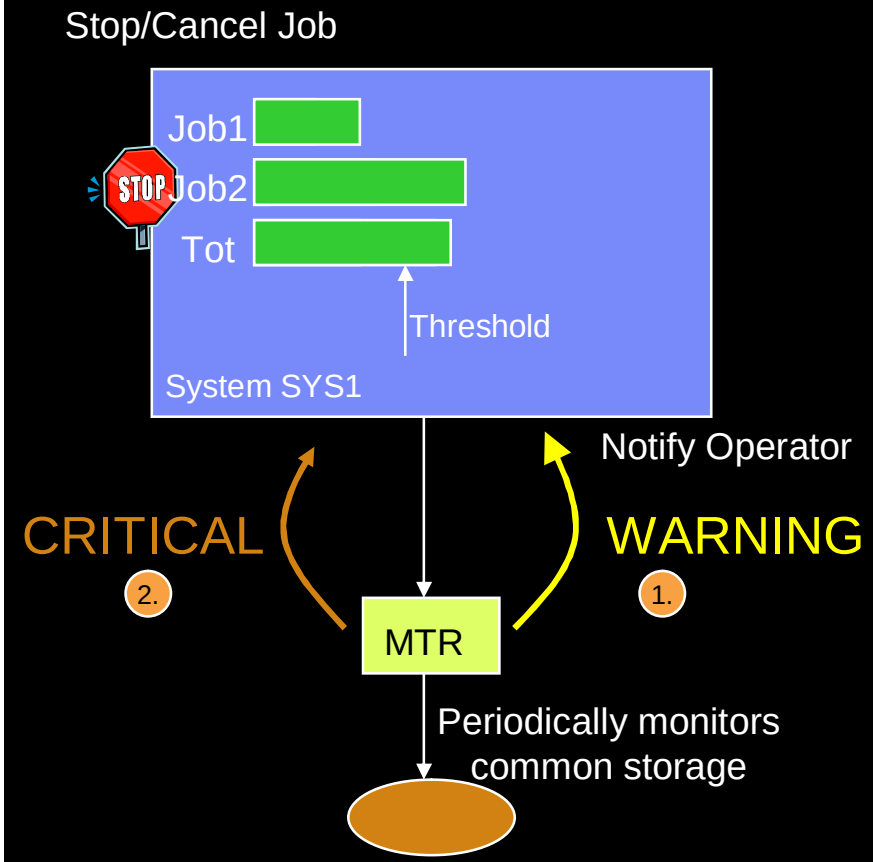
- **Uses server group concept in SA with**
  - Predefined instances
  - Variable availability target based on business demand
  - MTR monitoring transaction throughput and deriving health state
- **Intention: provide new application instance when throughput becomes CRITICAL**
- **Results:**
  - Increase of availability target based on health state CRITICAL causes SA to start a new server instance
  - Optionally other resources are terminated, if active

Other Scenarios: Common Storage Health



- **Common storage health**
  - Use of OMEGAMON common storage analyzer
  - Determine overall usage of common storage areas
    - SQA below and above
    - CSA below and above
  - Set health state and notify operator
  - Optionally, determine orphan storage and release it

Other Scenarios: Looping Job Detection



- **Processor health**
  - Use of OMEGAMON CPU-related commands and exceptions
  - Determine exceptional utilization of overall system
  - Determine exceptional utilization of single address spaces
  - Set health state and notify operator
  - Optionally, stop/cancel address space assumed to be looping

Other Scenarios *(cont.)*

- **Monitoring CICS connections to other CICS, DB2 and/or IMS**
  - Automatic start of missing connection
- **Monitoring transient CICS queues**
  - Increase priority of the update transaction for faster unload of queue
- **DB2, MQ archive log management**
  - Assistance to increase archive logs
- **Enqueue monitoring**
  - Automatic cancel of job holding enqueue resource for too long
  - Automatic detection of transactions holding CICS-enqueue
- **...**

Agenda

- **Motivation**
- **Resource / Exception Monitoring using OMEGAMON**
- **Exception-based Automation**
- **Monitor Resources**
  - Concepts
  - Administration and Operation
- **Health-based Automation**
- **Summary**

Summary

- **IBM automation products are tightly integrated with OMEGAMON**
  - System Automation and AF/OPERATOR utilize OMEGAMON monitors
  - System Automation provides Monitor Resources to determine health state and for health-based automation based on OMEGAMON data
- **Understanding the application health can lead to**
  - Higher availability
  - Higher efficiency
  - Improved IT service management

Bibliography

- **Related Documentation**
  - SA z/OS V3.1 Defining Automation Policy (SC33-8262)
  - SA z/OS V3.1 User's Guide (SC33-8263)
  - SA z/OS V3.1 Programmer's Reference (SC33-8266)
  - SA z/OS V3.1 Customizing and Programming (SC33-8260)
  - Command Reference Manual AF/OPERATOR Version 340
  - User's Guide AF/OPERATOR Version 340
- **White Paper**
  - IBM Tivoli System Automation for z/OS V2.3: *A Primer to Monitor Resources*, Paul Quigley
  - Checkout SA-homepage: http://www-1.ibm.com/servers/eserver/zseries/software/sa/