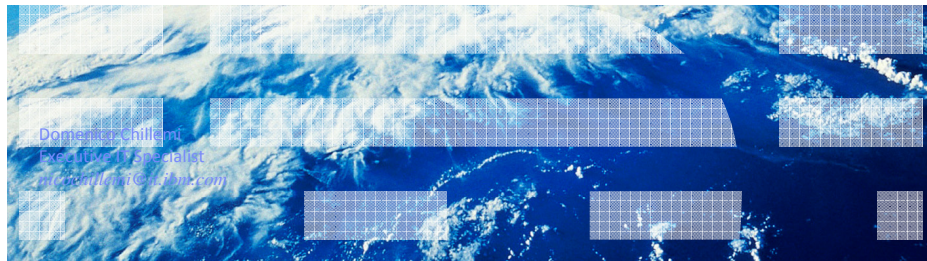


## TWS zCentric Migration Best Practices



**Tivoli** software

© 2012 IBM Corporation

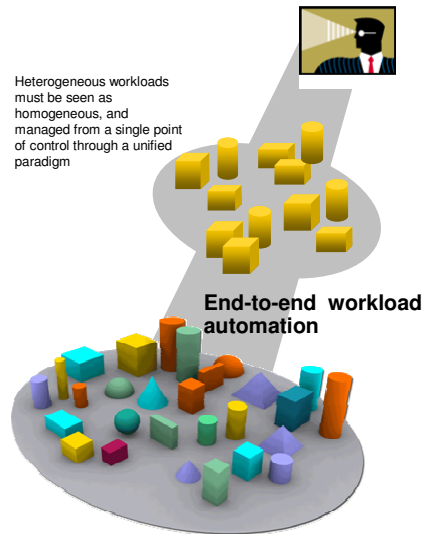
### IBM Tivoli Workload Automation end-to-end features

**IBM** **Tivoli** software

*... From a single point of control...*

- Single point of control to minimize the administrative oversight and time
- ↓
- Full impact view from the point of service delivery, for better efficiency, effectiveness and alignment of IT to business
  - Provide flexibility to establish the single point of control from any end-point

Heterogeneous workloads must be seen as homogeneous, and managed from a single point of control through a unified paradigm

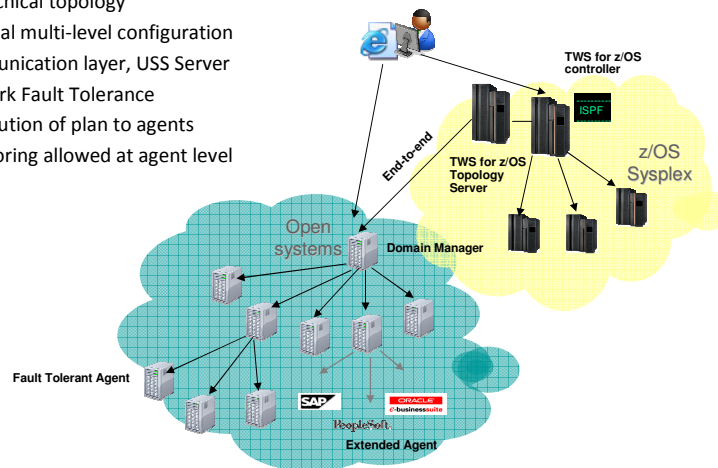


## End-to-end with fault tolerant capabilities (old technology)



It is the Plan-based End-to-end, this means:

- Hierarchical topology
- Optional multi-level configuration
- Communication layer, USS Server
- Network Fault Tolerance
- Distribution of plan to agents
- Monitoring allowed at agent level



3

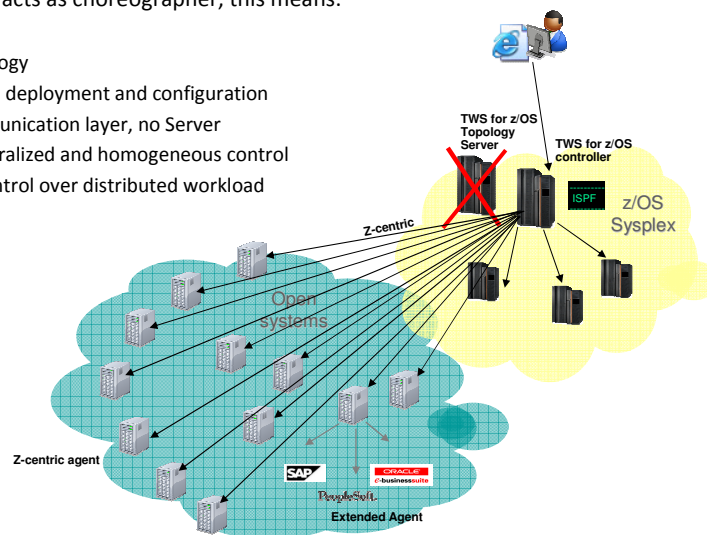
© 2012 IBM Corporation

## End-to-end with z Centric capabilities (new technology)



The TWS controller acts as choreographer, this means:

- Flat topology
- Simplified deployment and configuration
- No communication layer, no Server
- Fully centralized and homogeneous control
- Direct control over distributed workload



4

© 2012 IBM Corporation

## z Centric – feature exploitation (1)

- The z centric feature can be easily activated, by customizing the controller PARM member, by:

- Setting the HTTPPTS parameters
- Adding in the ROUTOPTS statement the HTTP/HTTPS parameters

- It's very easy to add, delete or modify z centric WSs.

- There is a MODIFY command dynamically reloading the destination definitions:  
**/F TW1A,RFRDEST**

```
HTTPPTS TCPJOBNAME('TCPJOB')
HOSTNAME(9.168.107.126)
HTTPPORTNUMBER(511)
CONNTIMEOUT(60)
JOBLOGRETRIEVAL(ONDEMAND)
JOBLOGMAXLINES(15000)
JOBLOGSECTION(FIRSTLAST)
JLOGTHREADNUM(3)
USRMEM(USRINF0)
SSLAUTHMODE(CAONLY)
SSLKEYRING(EQORING)
SSLKEYRINGTYPE(SAF)
SSLPORT(5111)
```

```
ROUTOPTS XCF(TW1AMC1,TW1AAA,TW1AAB,TW1AZZ)
HTTP(DSTWIN:'9.168.101.39'/31114,
DSTAIK:'9.168.101.65'/32114)
HTTPS(DSTSSL:'9.168.101.83'/31114)
```

- The same Z centric agent can run workload for various TWS for z/OS controllers at the same time

- This greatly simplify the infrastructure
- Reduces the maintenance effort



5

© 2012 IBM Corporation

## z Centric – feature exploitation (2)

- It is possible to define more Z centric WSs having the same destination, this:

- Allows to globally act on a subset of the overall workload ran by a certain server
- can be very useful if the same server runs workload related to different LOBs.

- The Z centric WSs grant the same flexibility of the z/OS WSs, this means:

- Open time intervals;
- Alternate WSs;
- Parallel servers

- It is possible to use the TWS for z/OS variables to tailor in a “centralized way” the workload on many distributed servers.

- In the example on the right the supplied variable related to the extended op. name, is used to parameterize the remote file name of an FTP job

```
//JOBREC
JOBTYP(//file transfer)
TRANSFERTYPE(UPLOAD)
TRANSFERMODE(TEXT)
SERVER(9.168.101.41)
LOCALFILE(c:\fileinput.txt)
REMOTEFILE(&XJOBNAM)
PROTOCOL(FTP)
TIMEOUT(1)
MINPORT(1035)
MAXPORT(1035)
PASSIVEMODE(YES)
JOBUSR(root)
```

Operation Extended Name:  
/tmp/MyFTPLocalTextFile.txt

6

© 2012 IBM Corporation

- The z centric agents support the **filewatch** feature

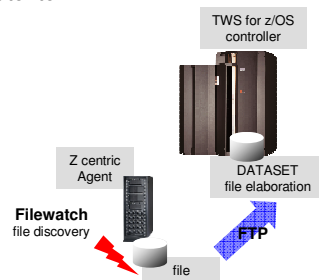
- It's an executable able to perform advanced file discovery
- File creation, deletion, edition. An example:

```
filewatch -condition wcr -filename C:\ftpd\ftp.file -int 30 -deadline 0
```

- It can be very usefully integrated with the FTP job executor to automate file discovery and transfer scenarios.

- Just create an application running z centric jobs where:

1. The first one runs the **filewatch** executable
2. Its successor run an FTP job.
3. Subsequent jobs perform the file content elaboration



- The settings defined in the EQQUX001 override the other settings. A typical error scenario:
  - If a job ends in error with ext. status OSUB
  - It may be the user submitting the job is that defined in the exit and not the right one
- The TWS controller has to resolve the IP addr of the z centric agents and vice versa. A typical error scenario:

- The TWS for z/OS user interface shows a jobs in “started” status
- The job has been really submitted on the server hosting the z centric agent
- In this case setting the HTTPPTS HOSTNAME keyword can solve the issue.

- Consider that the TWS controller tries to connect a Z centric agent only when it has to run the first daily job.

- Check on the real status of the agent by scheduling a TSO WSSTAT command, i.e.

```
WSSTAT SUBSYS(TW1A) WSNAME(ZAGT) STATUS(A)
```

- This can be very useful for agents running workload during the night

```
HTTPPTS TCPJOBNAME('TCPJOB')
HOSTNAME(9.168.107.126)
HTTPPORTNUMBER(511)
CONN TIMEOUT(60)
JOBLOGRETRIEVAL(ONDEMAND)
JOBLOGMAXLINES(15000)
JOBLOGSECTION(FIRSTLAST)
JLOGTHREADNUM(3)
USRMEM(USRINFO)
SSLAUTHMODE(CAONLY)
SSLKEYRING(EQORING)
SSLKEYRINGTYPE(SAF)
SSLPORT(51111)
```

- The symphony file is currently the bottleneck of a FTA End to End plan based configuration
- The FTA E2E agent normally requires a very high implementation effort
- The zCentric E2E agent can be implemented in less than 5 minutes
- Domain managers are more difficult to maintain
- FTA workstation cannot be changed in the Current Plan
- No SCRIPTLIB needed for zCentric E2E
- ...and many more minor advantages/disadvantages

- Identify FTA workstations
- Upgrade all FTA agents to TWS 8.6
- 
- **OR**
- Install the zCentric agent 8.6 on any system related to each FTA workstation
- Customize the TWS for z/OS Controller
- Operate in TWS to perform the migration

- First of all we need to identify all the FTA workstations you want to migrate
- A good best practise is to install the same agent level (zCentric and FTA) in all systems to migrate and this means
  - Upgrade all FTA agents to TWS 8.6
  - This installs also zCentric Agent 8.6 (demo scenario)
- In alternative install the zCentric agent 8.6 on any system related to each FTA workstation
  - Do this if FTA was not upgraded

- The TWS for z/OS Controller must be made capable to run the same jobs on both the agents
- zCentric E2E statements need to be added in the Controller PARMLIB member
  - ROUTOPTS
  - HTTPOPTS
- Consider to reflect in the Controller FTA and zCentric parameters
  - Controller PARM member for zCentric
  - E2E Server PARM members for FTA
- zCentric WS names do not need to be specified in zCentric statements, just the destinations

- Choose the right method of migration for each workstation
  - “all at once”  
**fast but dangerous**  
good with small number of jobs submitted on that workstation
  - “step by step”  
**slow but much more secure**  
good for big environments
- Depending on the choice, the activity to do is a little different
- Non centralized scripts need to be transferred to JOBLIB

**NOTE:** The best thing to do before starting a migration is to start with a clean Current Plan, where possibly all jobs completed, and with the JOBSUBMIT parameter set to NO

- Before starting Controller save the Database VSAM Files (AD and WS)
- Go in the Workstation Description
- Change every FTA workstation definition with a zCentric definition
  - specify the destination name
  - put Y in the zCentric field
- Extend the Current Plan
- When you are sure, stop the server and deactivate the E2E parameters

- Before starting Controller save the Database VSAM Files (AD and WS)
- Go in the Workstation Description
- Define a zCentric Workstation for each distributed machine where an FTA workstation is in place, with the destination pointing to the same distributed machine
- Go to the Application Description
- Start from the first application you would like to migrate
- Go inside the application and change the FTA WS names with the zCentric WS associated with the same destination names in all application operations
- Extend the Current Plan and see if all the jobs of that application occurrence complete correctly
- Start with another application

**This best practise will be shown in the demo**

