



IBM Software Group

IBM Tivoli Workload Automation

View, Control and Automate Composite Workloads

Mark A. Edwards
Market Manager
IBM Tivoli Workload Automation

Flora Tramontano
Product Manager
IBM Tivoli Workload Automation z/OS



@business on demand software

IBM CONFIDENTIAL

Tivoli Workload Automation is used by customers to deliver core business services

- Retailers use it to update pricing on Web sites, control inventory and replenish systems, transfer data, backup data, and control CRM and data warehouse systems
- Banks use it to process consumer and commercial transactions, provide online account services to customers, execute investment transactions, process back-end analytics and reports, and transfer and backup data
- Process and package food companies use it to schedule daily product runs, and to control inventory and deliveries
- Entertainment companies use it to schedule and load delivery trucks with media for distribution to the marketplace
- Healthcare providers use it to manage online services for customers and partners, to control back-end processes and backup data



Tivoli Workload Automation helps you to improve

- Visibility
 - ▶ to see your composite workloads and services

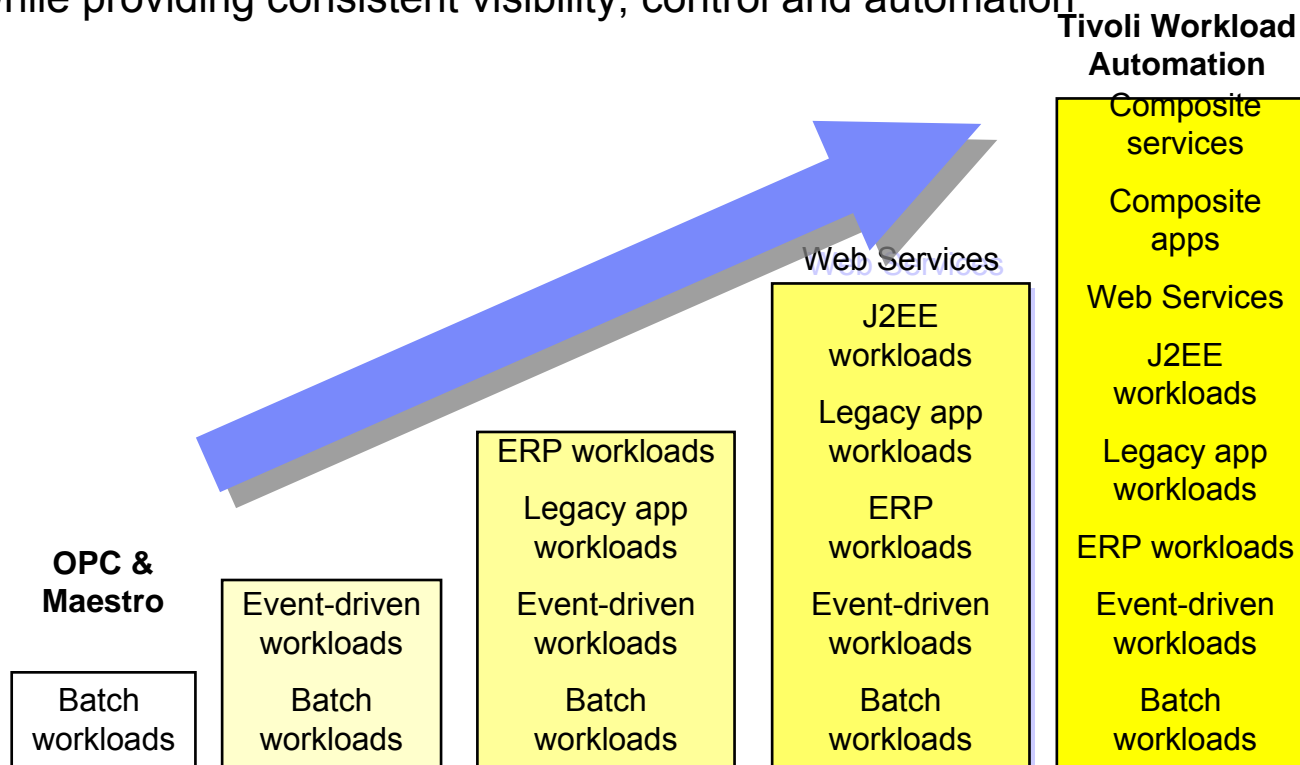
- Control
 - ▶ to *manage* composite workloads and heterogeneous applications, middleware and servers

- Automation
 - ▶ to build *agility* into operations to improve service execution reliability, performance and enhance innovation



IBM's evolution from Scheduling to Workload Automation

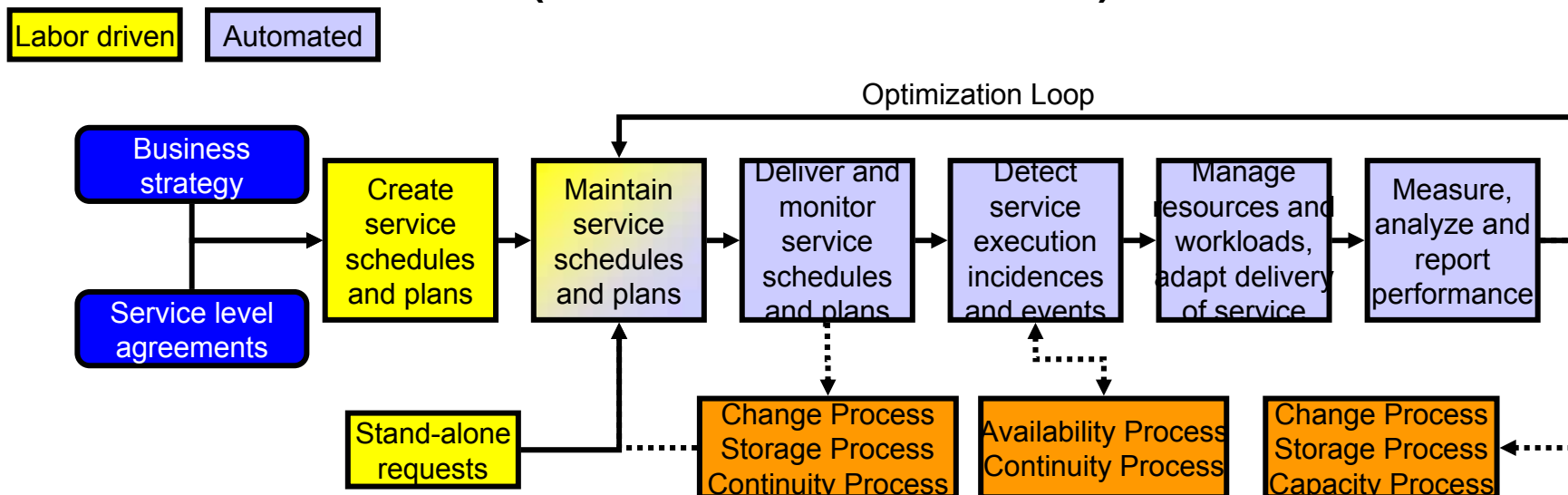
- It isn't just scheduling unattended batch jobs!
 - ▶ Service-oriented architecture with open interfacing for J2EE, Web Services and custom applications
 - ▶ Extended agents for packaged ERP, grid and z/OS systems
 - ▶ Integration with many Tivoli products for advanced services management and automation
- An enterprise backbone that drives composite workloads according to business policies while providing consistent visibility, control and automation



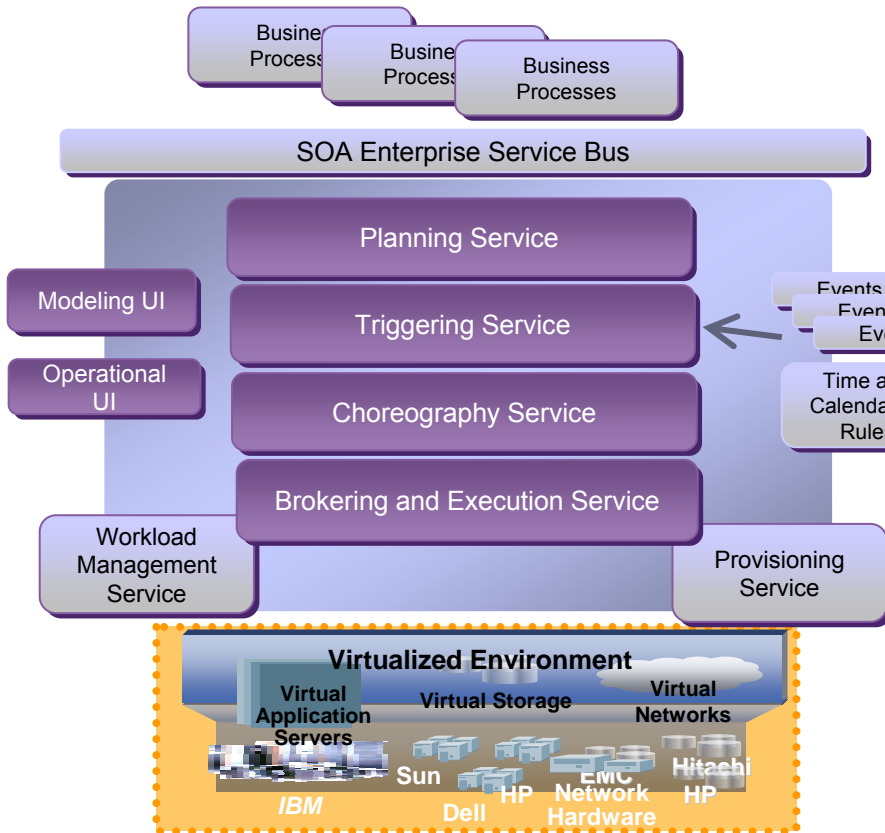
IBM's evolution from Scheduling to Workload Automation

- Tivoli Workload Automation
 - ▶ Helps view, control and automate the entire process of delivering composite workloads across heterogeneous applications and systems

IT Service Execution Process Activities (Formal ITUP / PRM-IT Flow)



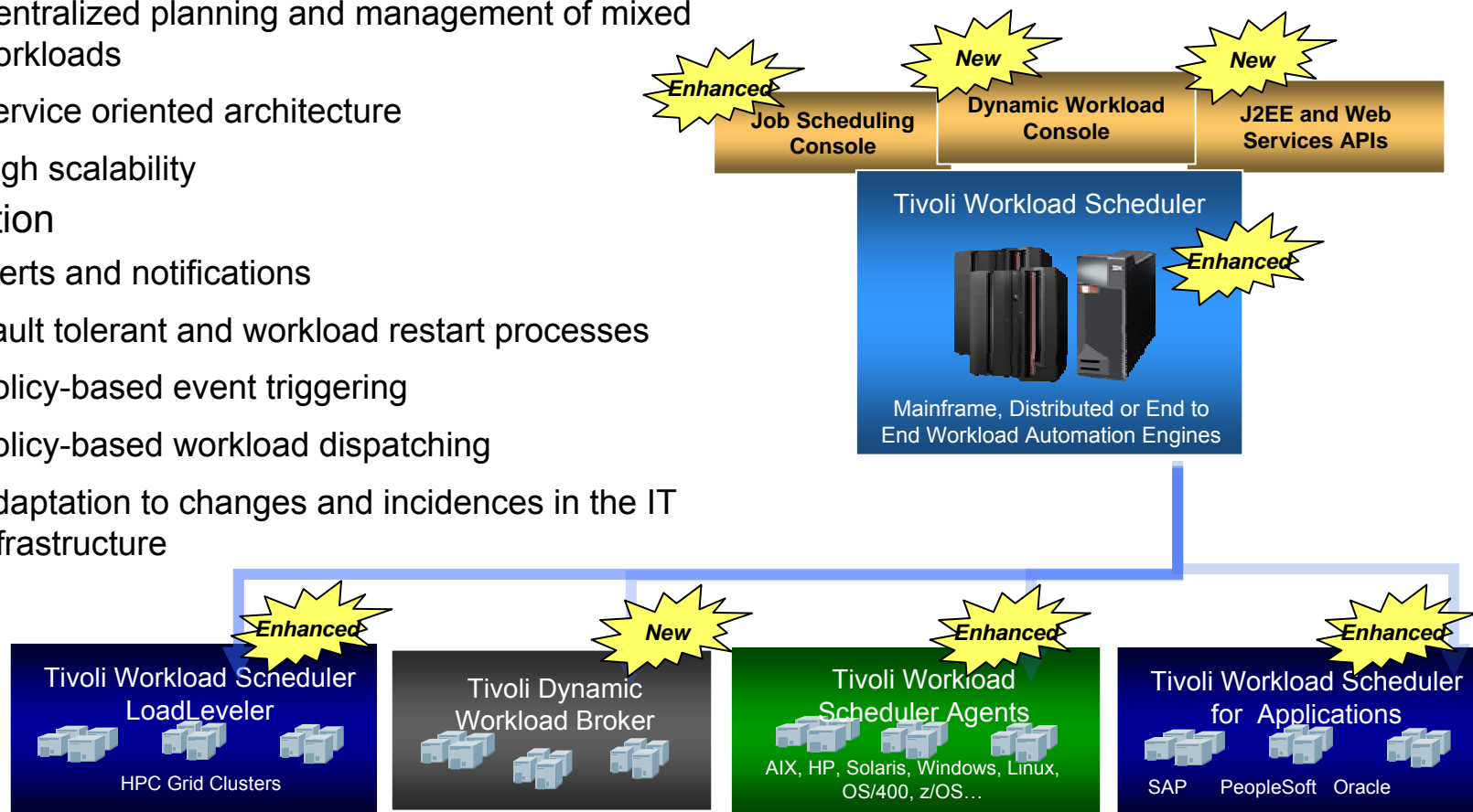
Tivoli Workload Automation services



- Integrate Tivoli Workload Automation managed workloads with composite business services through SOA
- Model and plan time-triggered workloads and policy-based event-triggered workloads
- Choreograph composite workloads and resolve dependencies throughout heterogeneous infrastructures
- Prioritize and broker workloads to best available resources
- Consolidate management of all enterprise workloads while virtualizing heterogeneous IT infrastructures

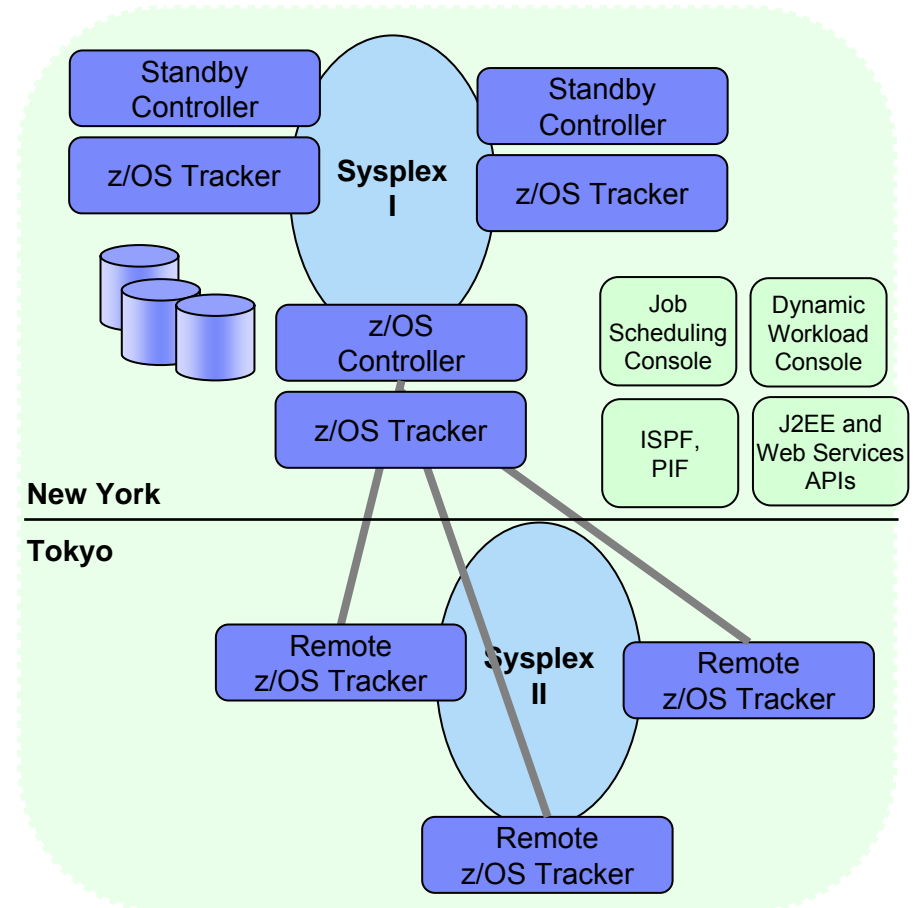
Tivoli Workload Automation components

- **Visibility**
 - ▶ Single, consistent view for monitoring, reporting and auditing
- **Control**
 - ▶ Centralized planning and management of mixed workloads
 - ▶ Service oriented architecture
 - ▶ High scalability
- **Automation**
 - ▶ Alerts and notifications
 - ▶ Fault tolerant and workload restart processes
 - ▶ Policy-based event triggering
 - ▶ Policy-based workload dispatching
 - ▶ Adaptation to changes and incidences in the IT infrastructure



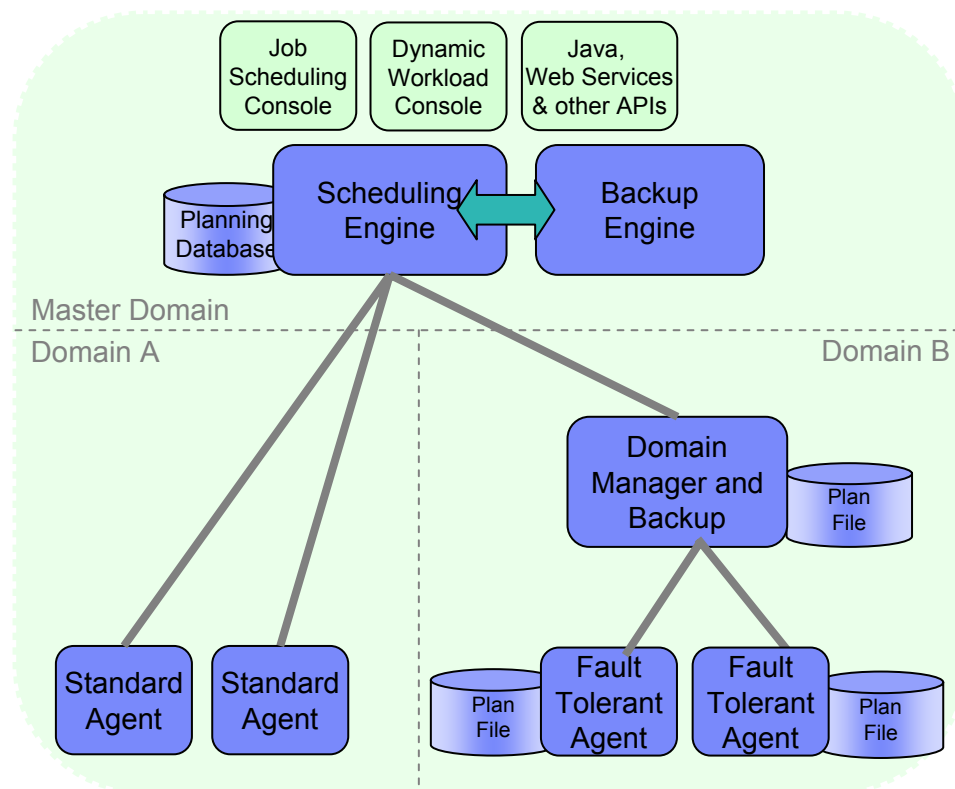
Mainframe workload automation

- Tivoli Workload Scheduler for z/OS
 - ▶ Environment: z/OS
- View
 - ▶ Centralized view of models, plans and production workloads
 - ▶ Integration with Tivoli Enterprise Portal and Tivoli Business Systems Manager
- Control
 - ▶ Centralized management of heterogeneous, composite workloads
 - ▶ Policy-based calendar-, time- and event-triggered planning and modeling
 - ▶ Highly secure and scalable
- Automation
 - ▶ Fault tolerant and workload recovery processes
 - ▶ Critical path analysis
 - ▶ Dispatching of workloads to best available resources
 - ▶ Integration with Tivoli Storage Manager to coordinate data backups with application workload events and plans
 - ▶ Integration with Tivoli System Automation to start, stop and move applications



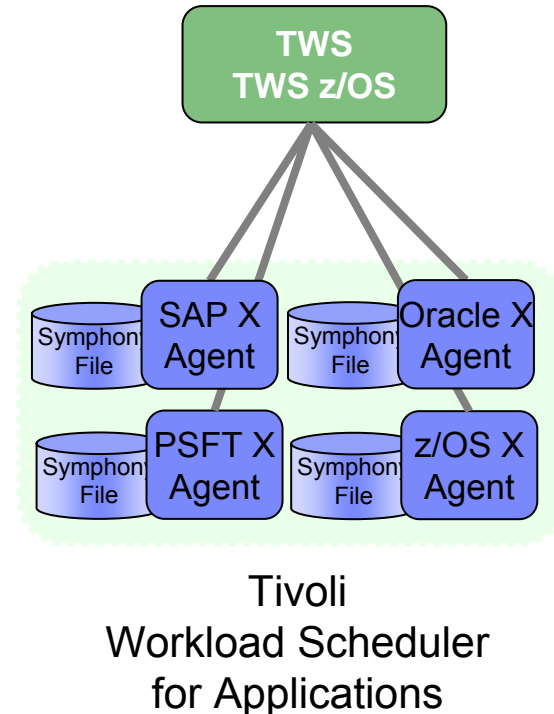
Distributed workload automation

- **Tivoli Workload Scheduler**
 - ▶ Environment: UNIX, Windows, Linux, i5/O5
- **View**
 - ▶ Centralized view of models, plans and production workloads
 - ▶ Integration with Tivoli Enterprise Portal and Tivoli Business Systems Manager
- **Control**
 - ▶ Centralized management of heterogeneous, composite workloads
 - ▶ Policy-based calendar-, time- and event-triggered planning and modeling
 - ▶ Highly secure and scalable flat or hierarchical topology
- **Automation**
 - ▶ Fault tolerant and workload recovery processes
 - ▶ Event filtering and automation engine
 - ▶ Integration with Tivoli Storage Manager to coordinate data backups with application workload events and plans
 - ▶ Integration with Tivoli System Automation to start, stop and move applications



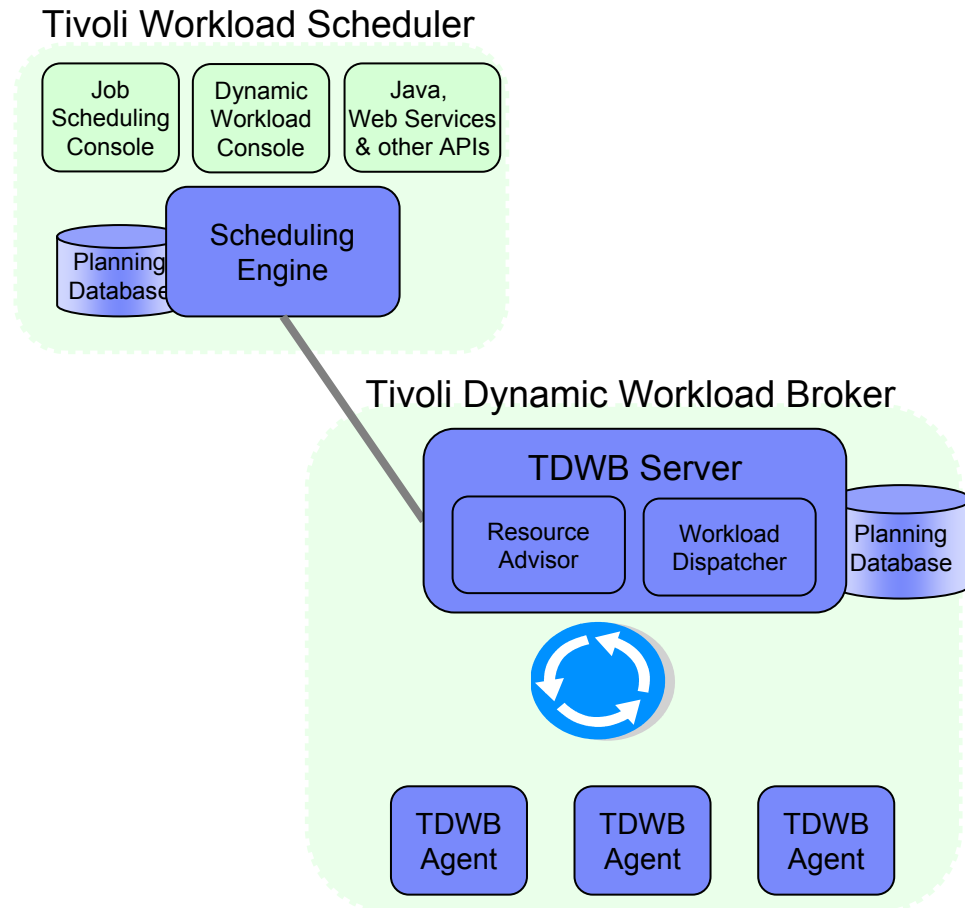
ERP system workload automation

- Tivoli Workload Scheduler for Applications
 - ▶ Environment: UNIX, Windows, Linux, z/OS
- View
 - ▶ Plan, model and track heterogeneous, composite workloads
- Control
 - ▶ Extend Tivoli Workload Scheduler automation capability to ERP and non-natively supported systems
 - ▶ Submit and restart SAP workloads through Tivoli Workload Scheduler
- Automation
 - ▶ Fault tolerance, workload recovery and load balancing
 - ▶ Resolve dependencies between SAP and non-SAP workloads

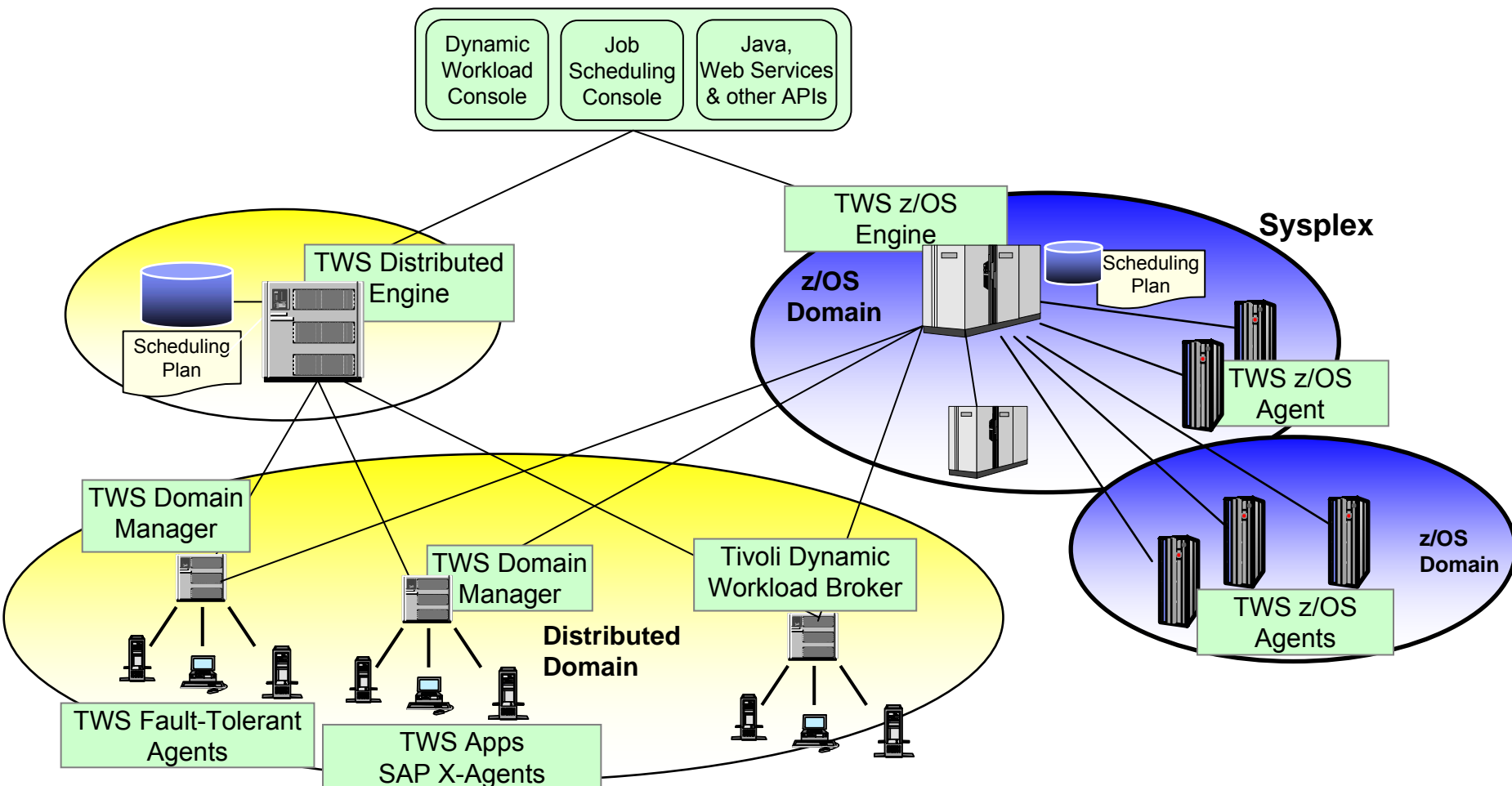


Distributed dynamic workload brokering

- Tivoli Dynamic Workload Broker
 - ▶ Environment: UNIX, Windows, Linux
- Stand-alone product
 - ▶ Add-on product to Tivoli Workload Scheduler
 - ▶ Users, applications and application servers can also submit workloads directly
- Policy-based dispatching
 - ▶ Matches workload requirements, priorities and attributes to IT resource attributes, performance and availability
 - ▶ Policy-based IT resource utilization
 - ▶ Automatic discovery of configuration changes
- Advanced automation
 - ▶ Integrated with Tivoli Provisioning Manager to provision resources on demand
 - ▶ Integration with Tivoli CCMDB to map Workload Automation into formal IT management processes



Tivoli Workload Automation in an end-to-end environment



- Centralized planning, monitoring and control of end-to-end environments
- Mainframe-centric, distributed-centric or mixed-mode management

Tivoli Workload Automation value

- Automatically manage composite workloads
 - ▶ Policy-based execution of workloads while resolving all mixed workload dependencies across heterogeneous IT resources
- Automatically execute services to meet contracted levels
 - ▶ Policy-based event filtering and triggering of workloads
- Automatically manage and adapt to planned configuration changes
 - ▶ Identify configuration changes and incorporate into workload execution without manually updating plans and choreography
- Automatically adapt to unplanned incidences
 - ▶ Monitor workloads and IT resources by exception
 - ▶ Generate alerts
 - ▶ Restart failed workloads
 - ▶ Failover scheduling engine and domain managers with no loss of service or historical or in-flight data
 - ▶ Adapt service execution to unplanned incidences by relocating workloads to available IT resources without manually updating plans and choreography

Tivoli Workload Automation value (continued)

- Automatically shrink batch windows
 - ▶ High scalability and high performance Workload Automation infrastructure
 - ▶ Centrally monitor workloads and IT resources on exceptions basis, generate alerts
 - ▶ Automatic adaptation to unplanned incidences

- Automatically adapt to spikes in workload volumes
 - ▶ Employ policy-based resource utilization to avoid overloading or idling IT resources
 - ▶ Distribute workloads across available IT resources
 - ▶ Provision resources on demand (through integration with Tivoli Provisioning Manager or similar)

- Automatically manage virtualized infrastructures
 - ▶ Adapt to virtual configuration changes automatically without manually updating plans and choreography
 - ▶ Distribute workloads across available virtual resources based on availability, performance and policies
 - ▶ Provision virtual resources on demand (through integration with Tivoli Provisioning Manager or similar)



Tivoli Workload Automation value (continued)

- IBM vision, investment, commitment and support to
 - ▶ Elevate Workload Automation in strategic priority
 - ▶ Help to improve your ability to deliver reliable and scalable composite enterprise workloads and services
 - ▶ Help to reduce your administrative and operations expenses and complexity, and increase return on investment for existing IT resources



Scheduling to Workload Automation evolution in TDWA

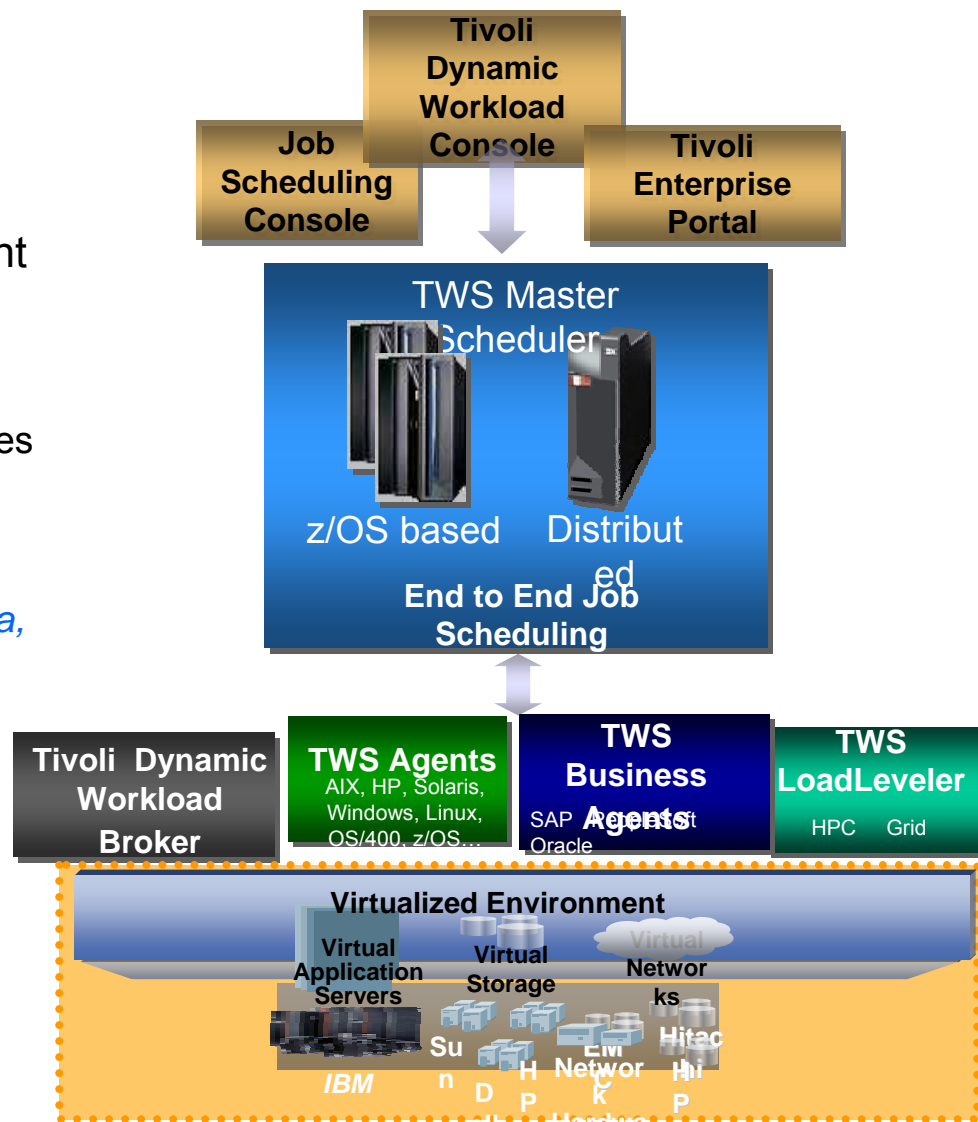
- **Composite Workload** management backbone

- ▶ *Policy-based* and *SLA-based* control of any workload, in a *virtualized* environment

- Dynamic IT resource utilization *optimization*
- *Automatic discover* of IT resource changes and adapt workload
- *Critical Path* management
- Plan optimization based on *historical data, trend analysis and what if analysis*

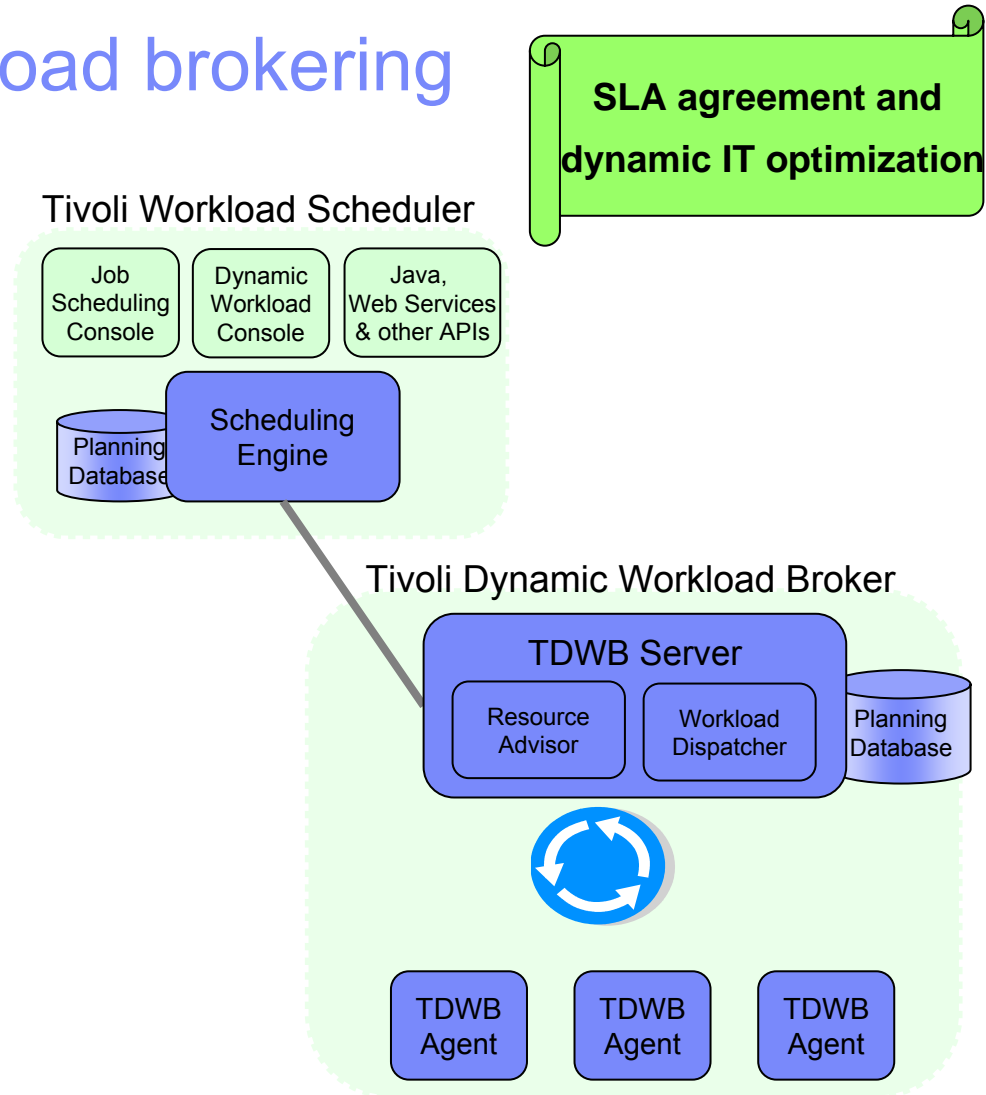
- ▶ Ability to manage and control *new triggering criteria and job sequencing conditions*

- ▶ *Automation* to reduce costs and improve efficiency and *evolution* to latest technology



Distributed dynamic workload brokering

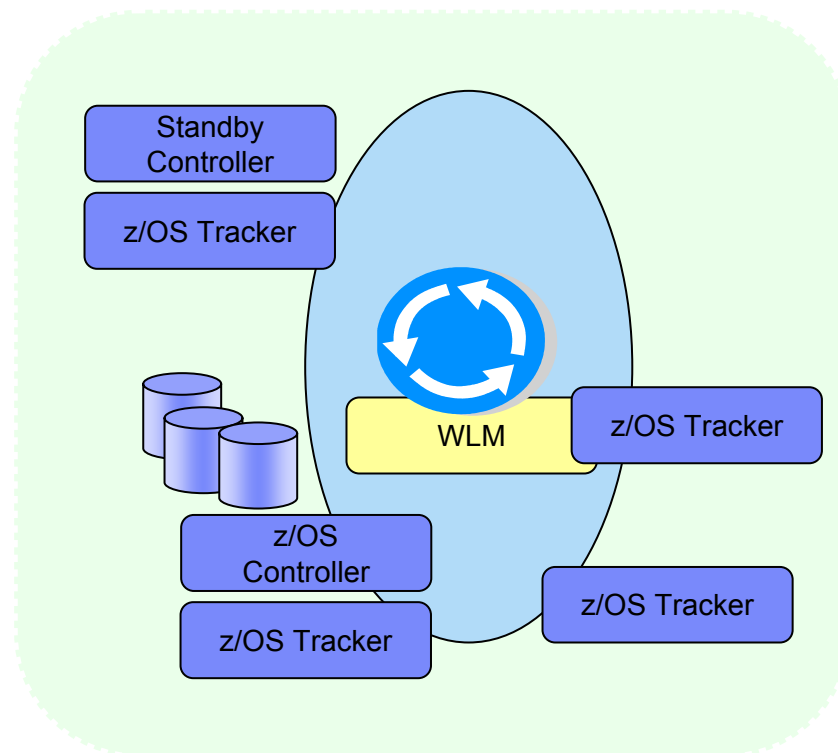
- Tivoli Dynamic Workload Broker
 - ▶ Environment: UNIX, Windows, Linux
- Stand-alone product
 - ▶ Add-on product to Tivoli Workload Scheduler
 - ▶ Users, applications and application servers can also submit workloads directly
- Policy-based dispatching
 - ▶ Matches workload requirements, priorities and attributes to IT resource attributes, performance and availability
 - ▶ Policy-based IT resource utilization
 - ▶ Automatic discovery of configuration changes
- Advanced automation
 - ▶ Integrated with Tivoli Provisioning Manager to provision resources on demand
 - ▶ Integration with Tivoli CCMDB to map Workload Automation into formal IT management processes



z/OS Dynamic workload brokering

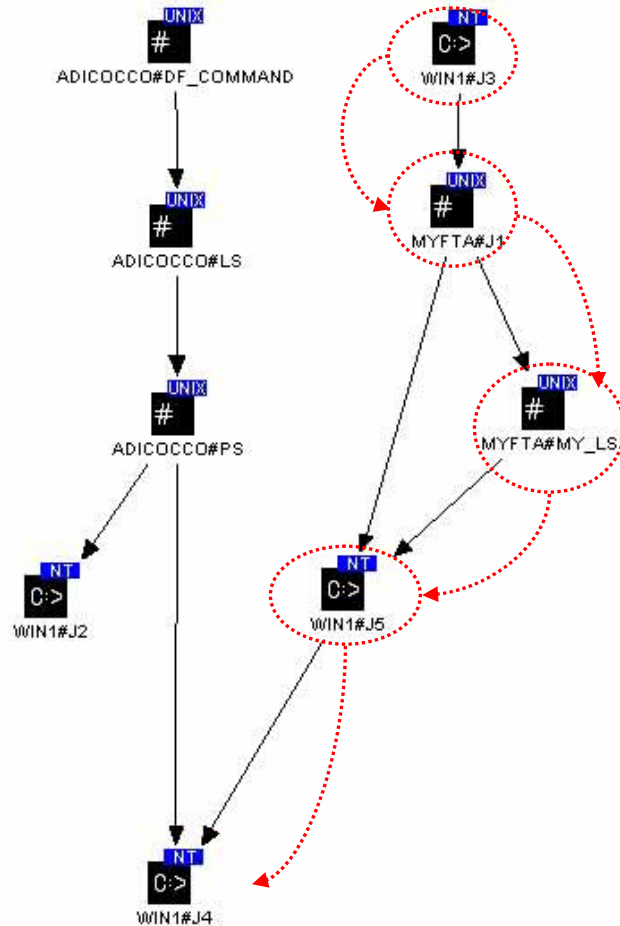
- TWS integrates with Workload Manager (WLM), leveraging Scheduling Environment (SE) and Service Class objects
- New integration with WLM SE has been provided
 - ▶ Dynamic routing of workload to MVS systems in the Sysplex based on best available resources
 - ▶ SE becomes part of TWS operation
 - ▶ SE availability status is checked before jobs submission
 - ▶ Automatic re-submission of jobs at SE availability status change
- Integration with WLM Service Class has been enhanced in TWS
 - ▶ WLM Service class can be defined at TWS operation level
 - ▶ Jobs will be promoted to the specified WLM Service Class if they are on the Critical Path

**SLA agreement and
dynamic IT optimization**



Critical Path feature

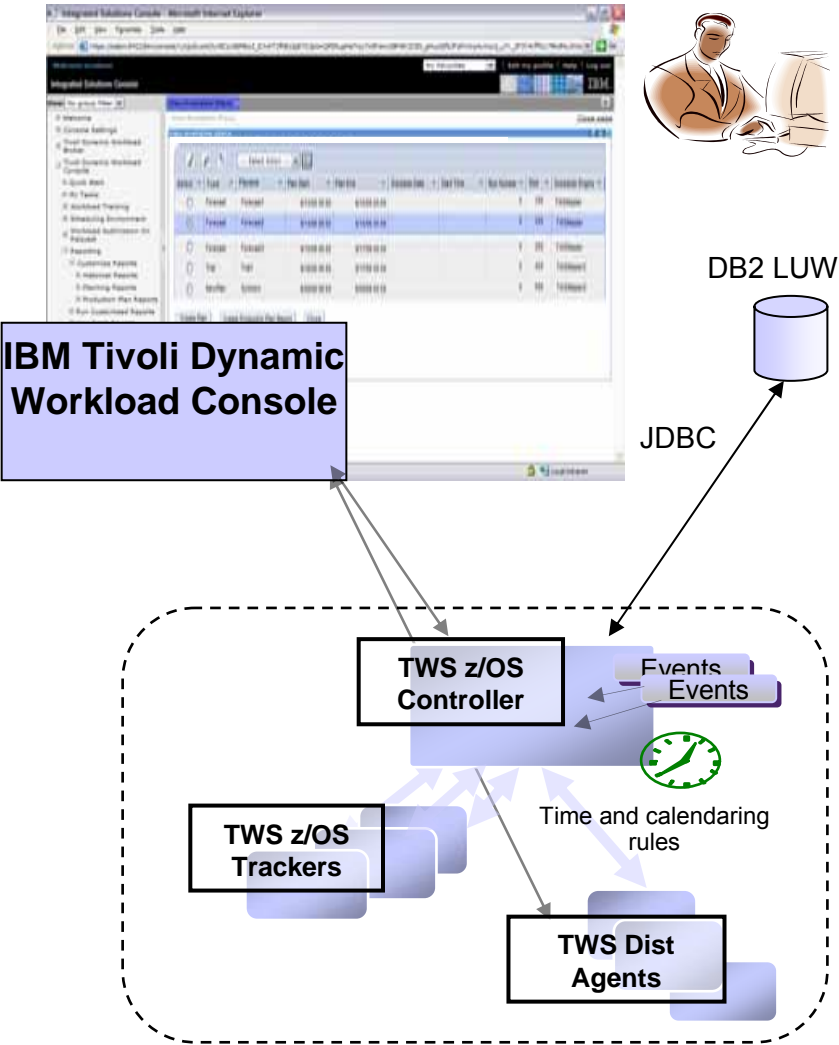
SLA agreement



- Proactive management of milestone jobs (that must meet SLAs)
 - ▶ Jobs that must not miss their deadlines can be flagged as critical
 - ▶ New views are provided to monitor Critical Jobs and their Critical Paths
 - ▶ Critical Paths to critical jobs are calculated at Daily Plan
 - ▶ Jobs on Critical Paths are automatically promoted when they are late
 - Internal priority is leveraged
 - WLM Service Class are leveraged
- Dynamic Critical Path is going to be implemented in the near future

Reporting feature

SLA agreement and plan optimization



- Pre-defined and user-defined Reports management (Tivoli Dynamic Workload Console capability)
 - ▶ Data for reporting is stored into (and extracted from) a distributed DB2
- Customize, generate, view, print, save and share Reports
- Report types
 - ▶ Job Run History
 - ▶ Job Run Statistics
 - ▶ Workstation workload summary
 - ▶ Workstation workload runtime
 - ▶ Production plan (distributed only)
 - ▶ Custom SQL reports

Reporting feature – some example

SLA agreement and plan optimization

▶ Job Run History and Statistics

- Job Run History – measure business compliance and SLA
 - Ended in Error, late jobs, missed deadlines, rerun indicators
- Job Run Statistics – workload scheduling forecasting
 - Success/error rates, min/max/average duration, late and long duration statistics

Job Run History

Report Description: The report collects the historical job-execution data during a time interval. It will allow to detect which jobs ended in error as well late jobs, missed deadline, long duration, rerun indicators for reruns, etc.

Report Date: Tuesday, February 27, 2007 5:21:34 PM UTC
 Report Type: JobRunHistory
 Total Rows: 6

Job Run History Listing

Job Name	Workstation (Job)	Job Stream Name	Workstation (Job Stream)	Scheduled Time	Actual Start Time	Started Late (delay hh:mm)	Ended Late (delay hh:mm)	Status	Rerun type
JOB1	FCARTERS	JS1	FCARTERS	Monday, February 5, 2007 4:00:00 AM UTC	Monday, February 5, 2007 3:43:00 PM UTC			Successful	Regular Job
JOB2	FCARTERS	JS1	FCARTERS	Monday, February 5, 2007 4:00:00 AM UTC	Monday, February 5, 2007 3:43:00 PM UTC			Error	Regular Job
JOB3	FCARTERS	JS1	FCARTERS	Monday, February 5, 2007 4:00:00 AM UTC	Monday, February 5, 2007 3:43:00 PM UTC			Successful	Regular Job
JOB4	FCARTERS	JS2	FCARTERS	Monday, February 5, 2007 4:00:00 AM UTC	Monday, February 5, 2007 3:43:00 PM UTC			Successful	Regular Job
JOB4	FCARTERS	JS2	FCARTERS	Monday, February 5, 2007 4:00:00 AM UTC	Monday, February 5, 2007 3:45:00 PM UTC			Successful	Job Every

Job Name: JOB1
 Workstation Name: FCARTERS
 Script: dir
 Login User: fcarteri
 Job creator: fcarteri

Runs by status

Runs by status	% of Total runs
Successful	100.00%
Error	0.00%
Total	1.0
Total Reruns	0

Runtime exceptions

Runtime exceptions	% of Total runs
Started Late	0.00%
Ended Late	0.00%
Long Duration	1.0000%

Duration (hh:mm) Date Cpu Consumption

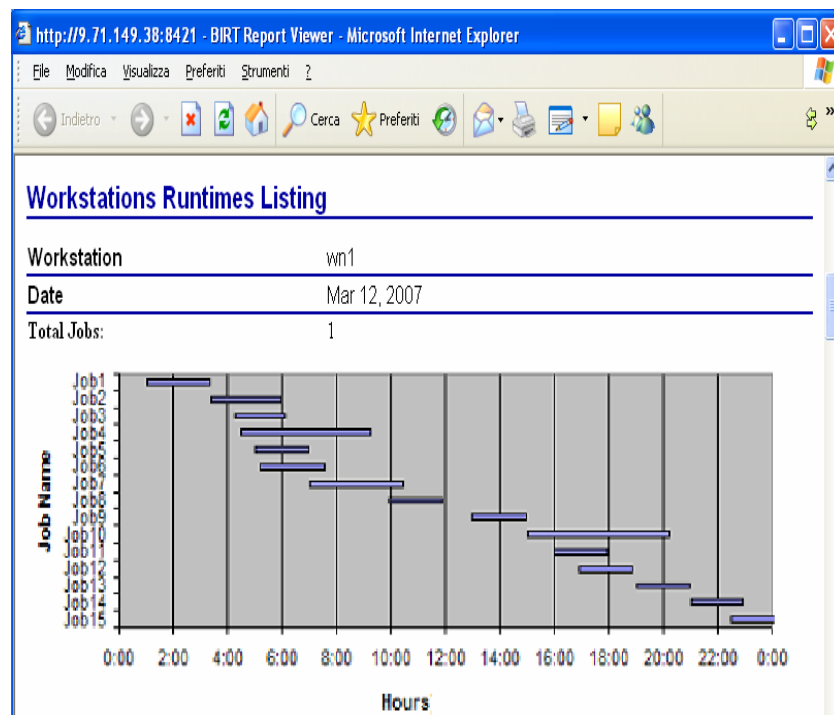
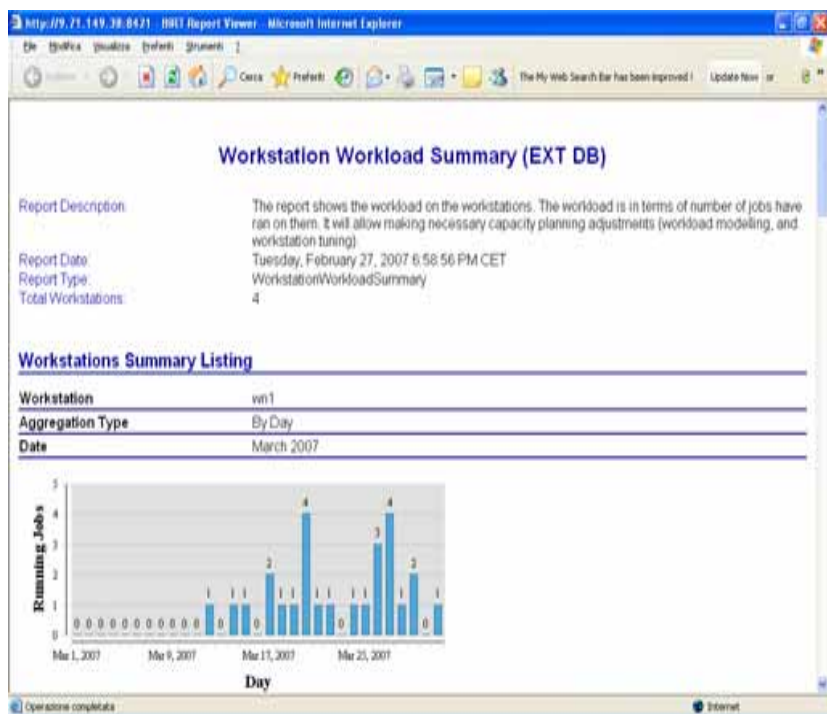
Duration (hh:mm)	Date	Cpu Consumption
Last Run 00:01	Monday, February 5, 2007 3:43:00 PM UTC	0
Min 00:01	Monday, February 5, 2007 3:43:00 PM UTC	0
Max 00:01	Monday, February 5, 2007 3:43:00 PM UTC	0
Average 00:01	Monday, February 5, 2007 3:43:00 PM UTC	0

Reporting feature – some example

SLA agreement and
plan optimization

▶ Workstation Workload Summary and Runtimes

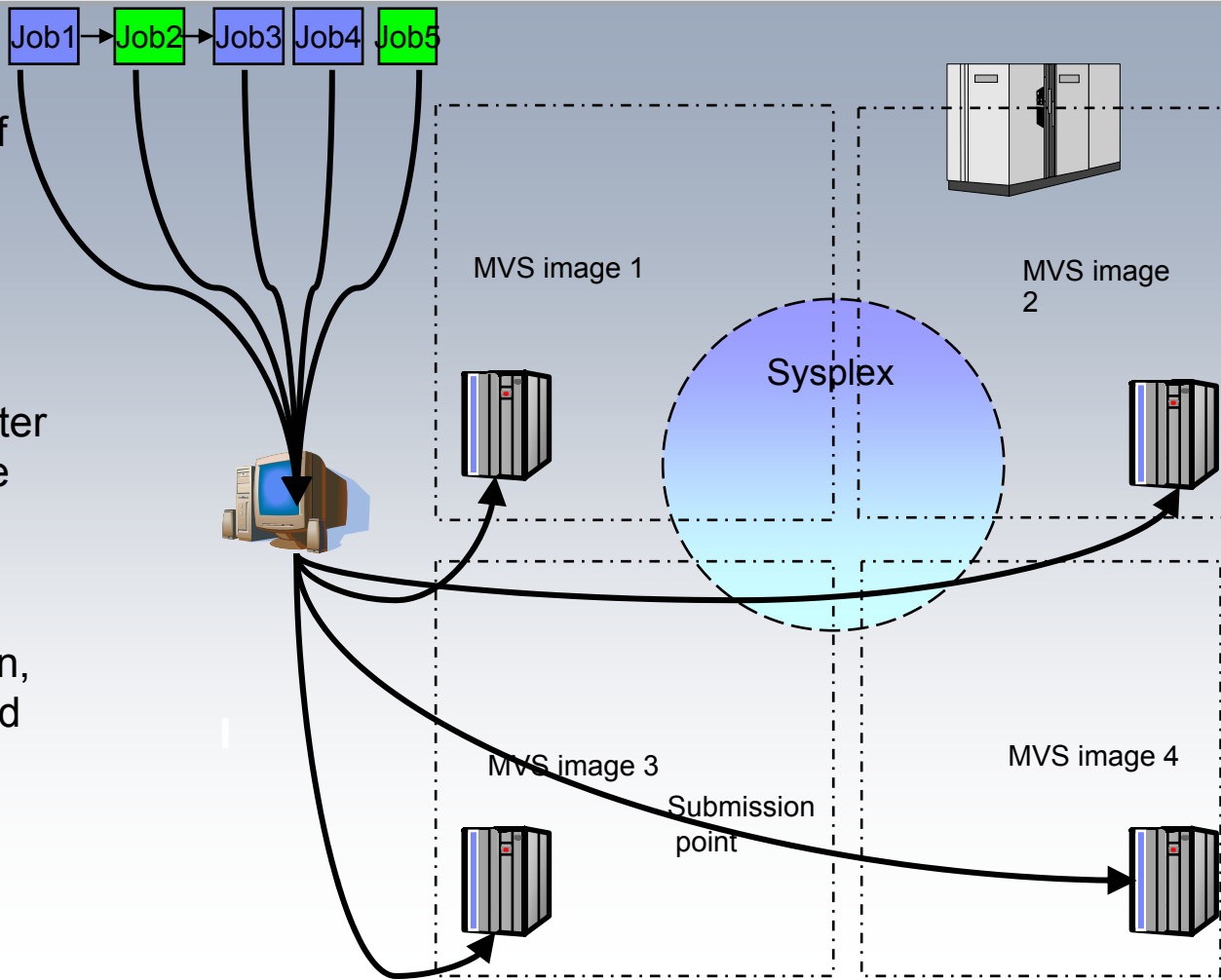
- Workstation Workload Summary – capacity planning adjustments
 - Workload by workstation, aggregated by hour, day, etc.
- Workload Workstation Runtime – monitor and tune workload capacity
 - Job run times and durations on the workstation



"Virtual" pooling workstation

SLA agreement

- ▶ Automatic distribution of workload to different destinations and alternate workstation
- ▶ Traditional z/OS computer workstations might have only one "destination"
- ▶ New "virtual" workstation, multiple destinations and availability info



Distributed Event-driven scheduling

- Extends TWS calendar based planning to unplanned world
- Mix plan-driven scheduling with reception and combination of real-time “events”
- Automatic notification for failures in scheduling infrastructure or problems in scheduled jobs

Sample scenarios

- If file */abc/file1* is FTP'ed on *machine-x* and TWS *job** is successful, start TWS *jobstream-y*
- If msg “*EGS0243 Tr A terminated*” issued in file */var/log1* on *machine-x*, start TWS *job_tr1*
- If event “*job_exec_tr1=succesfull*” is received in 1 hour send an Informational email to joe.smith@company.com otherwise send an Error email to the same user

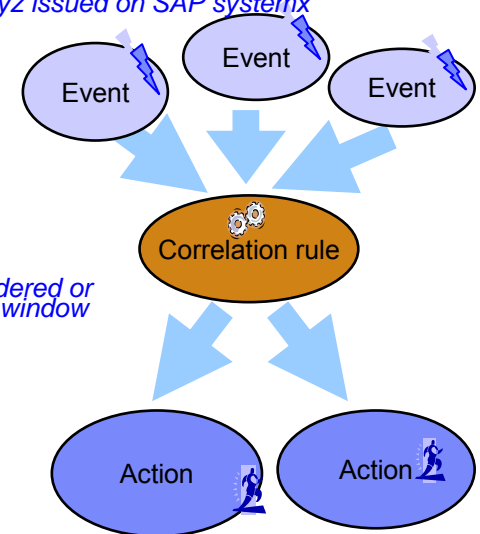
Event-triggered workload management

Events

- ▶ *File created, modified, deleted*
- ▶ *Message logged to a file*
- ▶ *Job/jobstream status change*
- ▶ *Return Code condition*
- ▶ *TWS agent status*
- ▶ *Event xyz issued on SAP systemx*

Correlation rules

- ▶ *Filter*
- ▶ *Combination of events (ordered or random) in a specific time window*

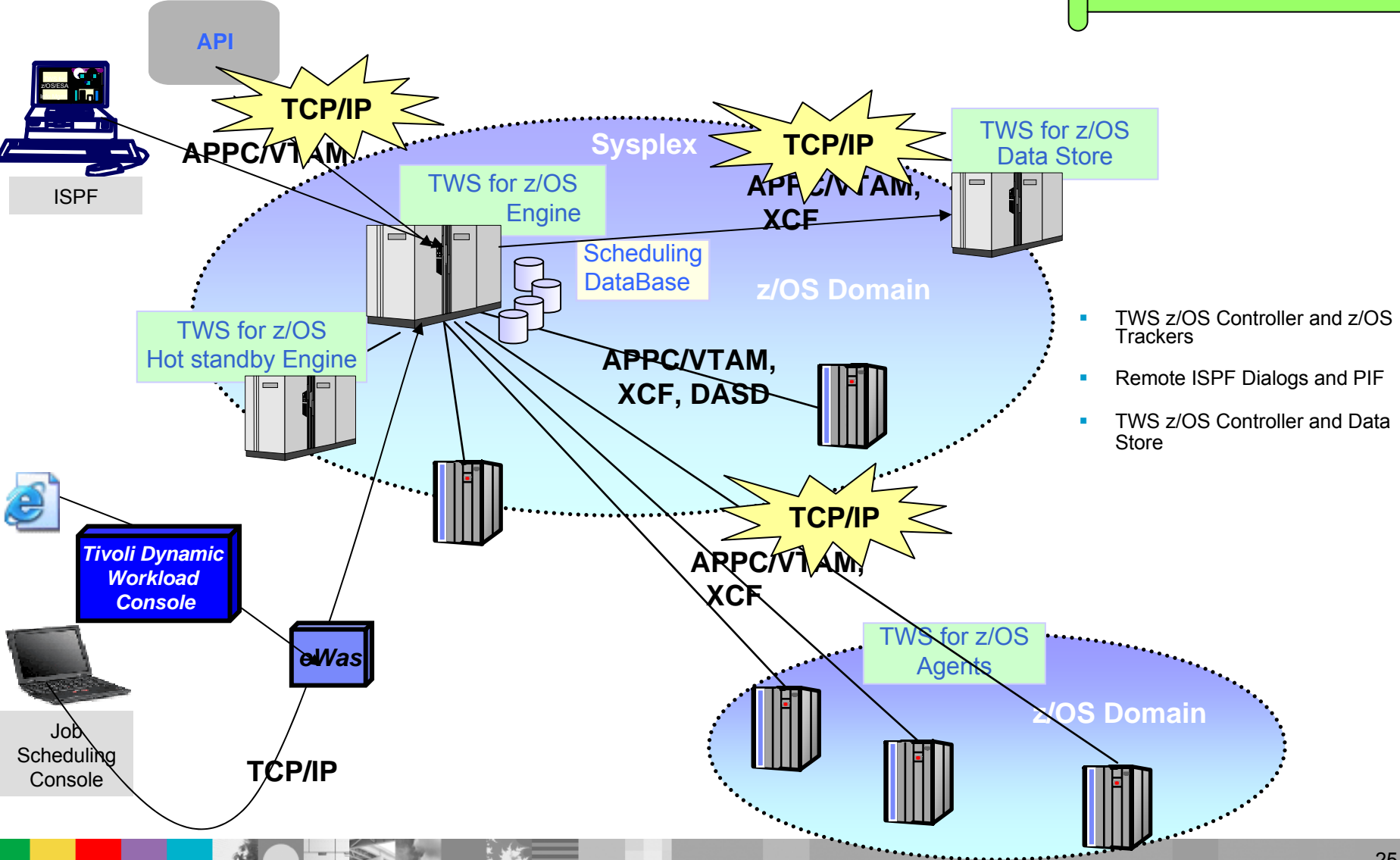


Actions

- ▶ *Submit a TWS job/jobStream*
- ▶ *TWS answers a prompt*
- ▶ *Send an e-mail*
- ▶ *Send An event to TEC*
- ▶ *Write a message in the message log*

z/OS TCP/IP Connection

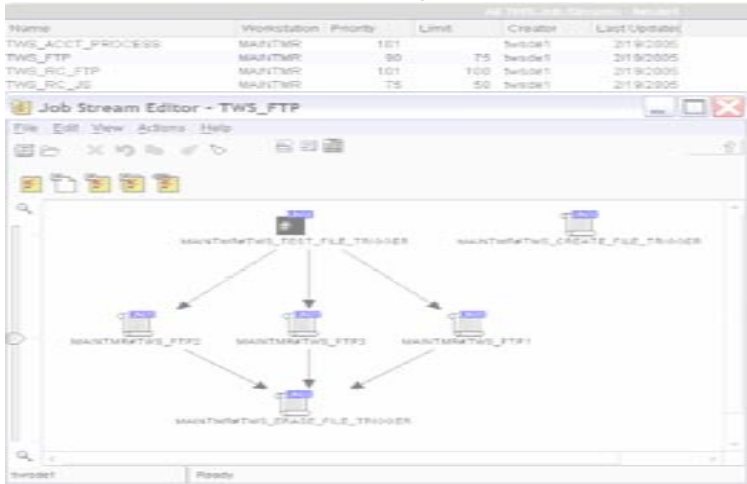
Evolving with System evolution



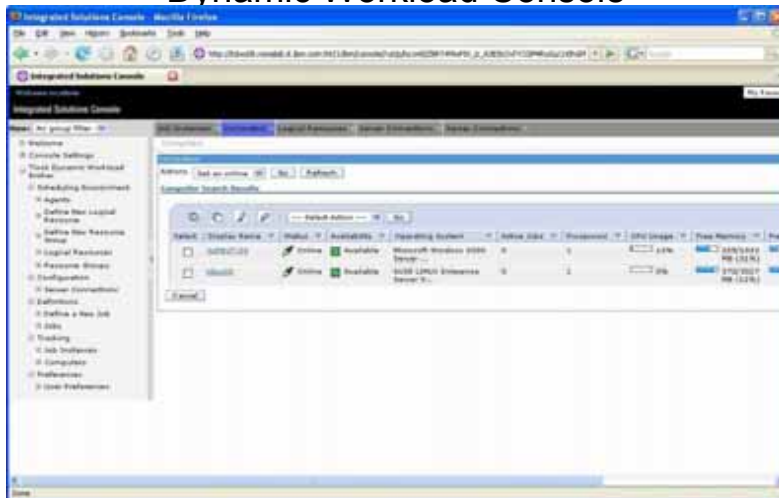
- TWS z/OS Controller and z/OS Trackers
- Remote ISPF Dialogs and PIF
- TWS z/OS Controller and Data Store

Single point of control

Job Scheduling Console



Dynamic Workload Console



Automation

- Job Scheduling Console
 - ▶ Drag-and-drop object-based planning and modeling
 - ▶ Filterable views of all jobs and jobstreams including all dependencies

- Dynamic Workload Console
 - ▶ Monitor and manage
 - Exceptions-based monitoring
 - Monitor and tune workload throughput and performance
 - Take manual action
 - Trial forecasting
 - ▶ Report production plans and performance
 - Standard reports (templates)
 - User-defined reports
 - Historical execution data and statistics
 - Audit reports for compliance

Thank You for Joining Us today!

Go to www.ibm.com/software/systemz to:

- ▶ Replay this teleconference
- ▶ Replay previously broadcast teleconferences
- ▶ Register for upcoming events

