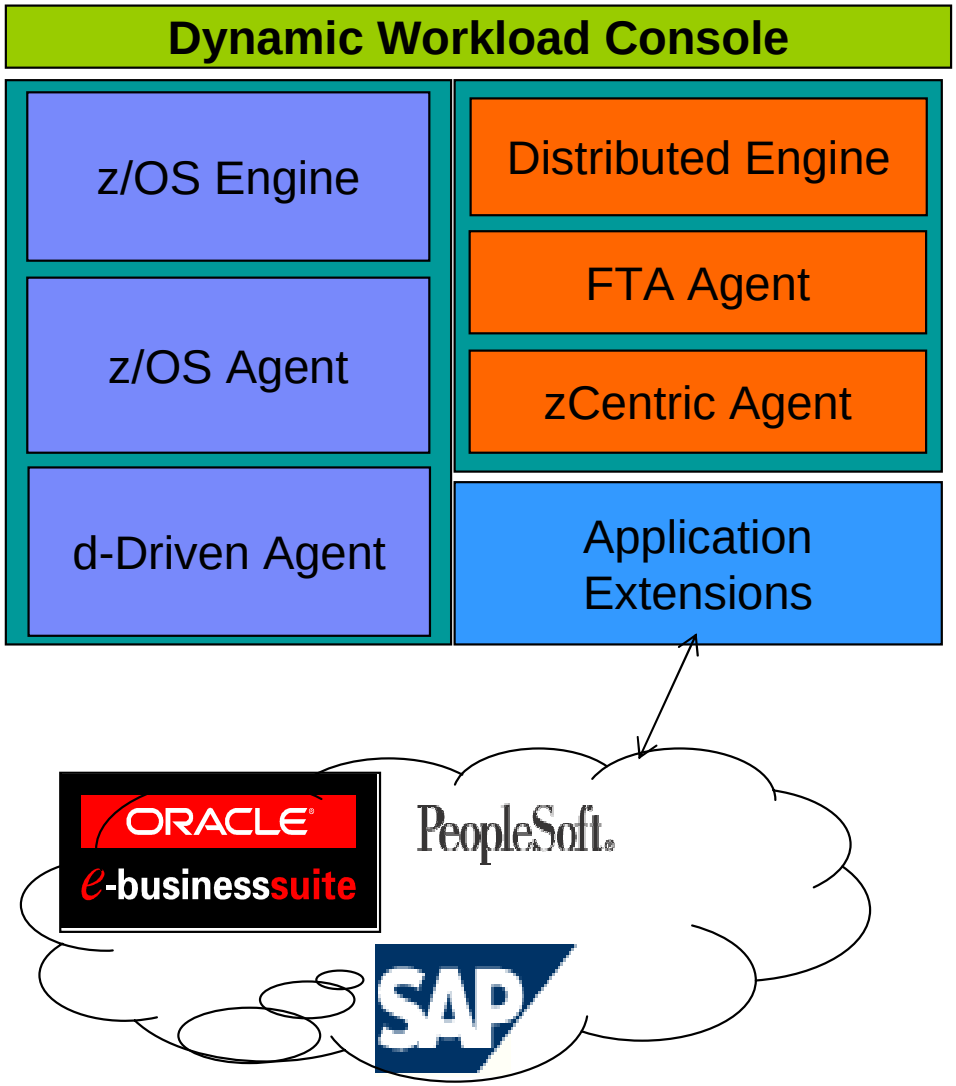
Domenico Chillemi
Executive IT Specialist
nicochillemi@it.ibm.com
+39 335 5999629

TWS for z/OS Update V8.6 General Overview

PCTY2012
Pulse Comes to You

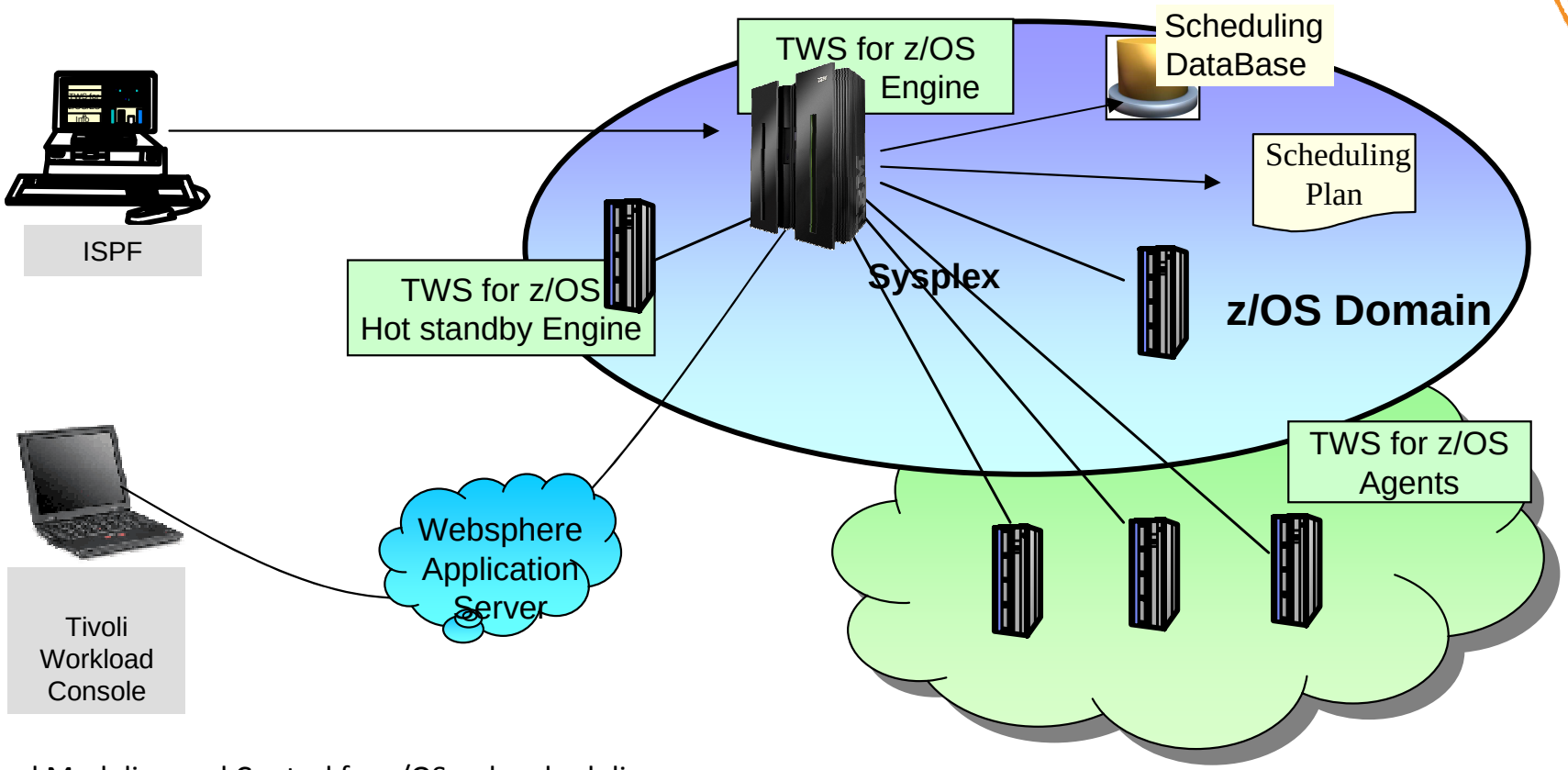
Tivoli Workload Automation Overview

IBM Batch scheduler: The Big Picture



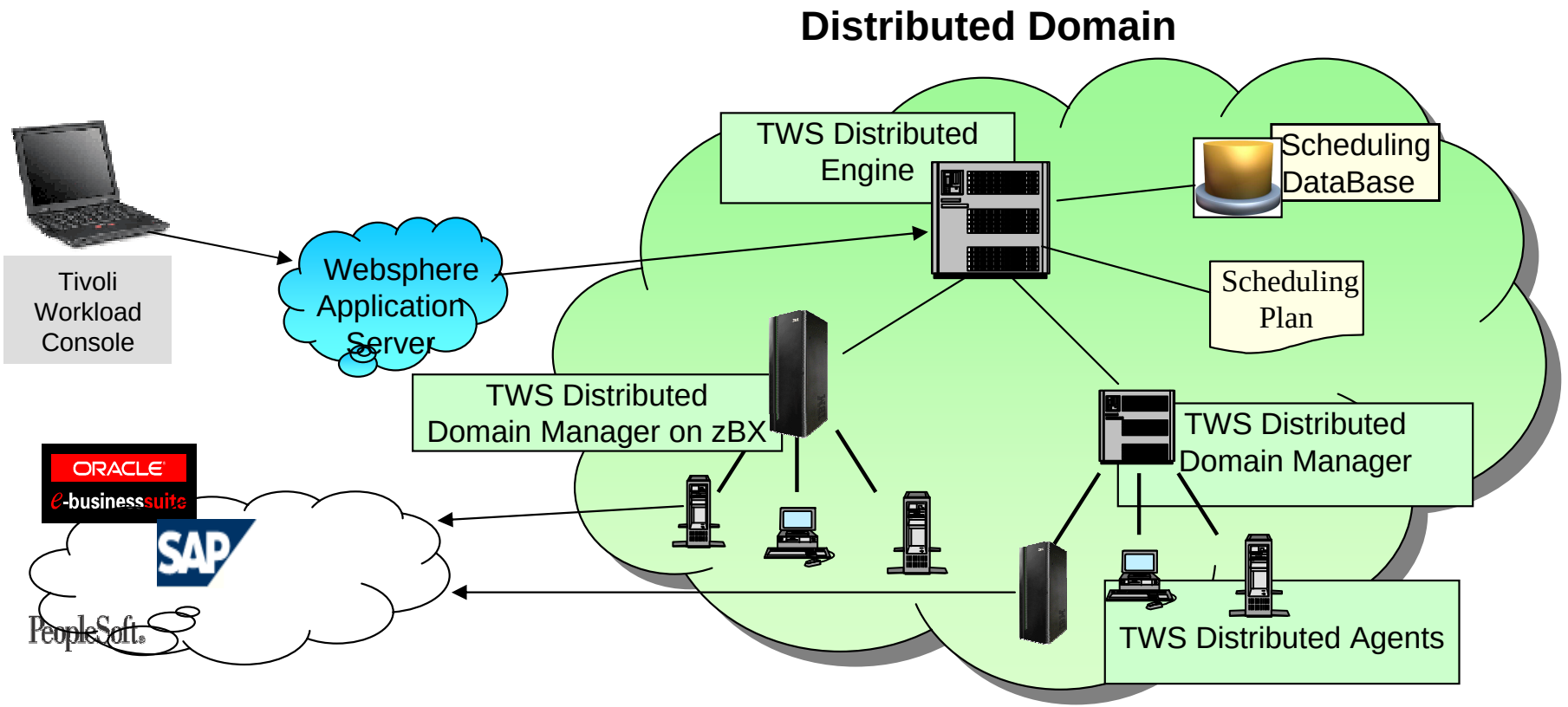
- IBM Tivoli Workload Scheduler
  - Distributed Scheduler (Maestro technology)
  - Dynamic Workload Console with distributed Connectors
  - Fault Tolerant Agent (Maestro technology)
  - zCentric static and dynamic Agents (new technology)
- IBM Tivoli Workload Scheduler for z/OS
  - z/OS Scheduler (OPC technology)
  - Dynamic Workload Console with the z/OS Connector
  - d-Driven z/OS Agent (new technology)
- IBM Tivoli Workload Scheduler for Applications
  - Application integration with SAP R/3
  - Application integration with Oracle e-Business Suite
  - Application integration with People Soft
  - Application integration for emerging workloads

TWS z/OS Configuration



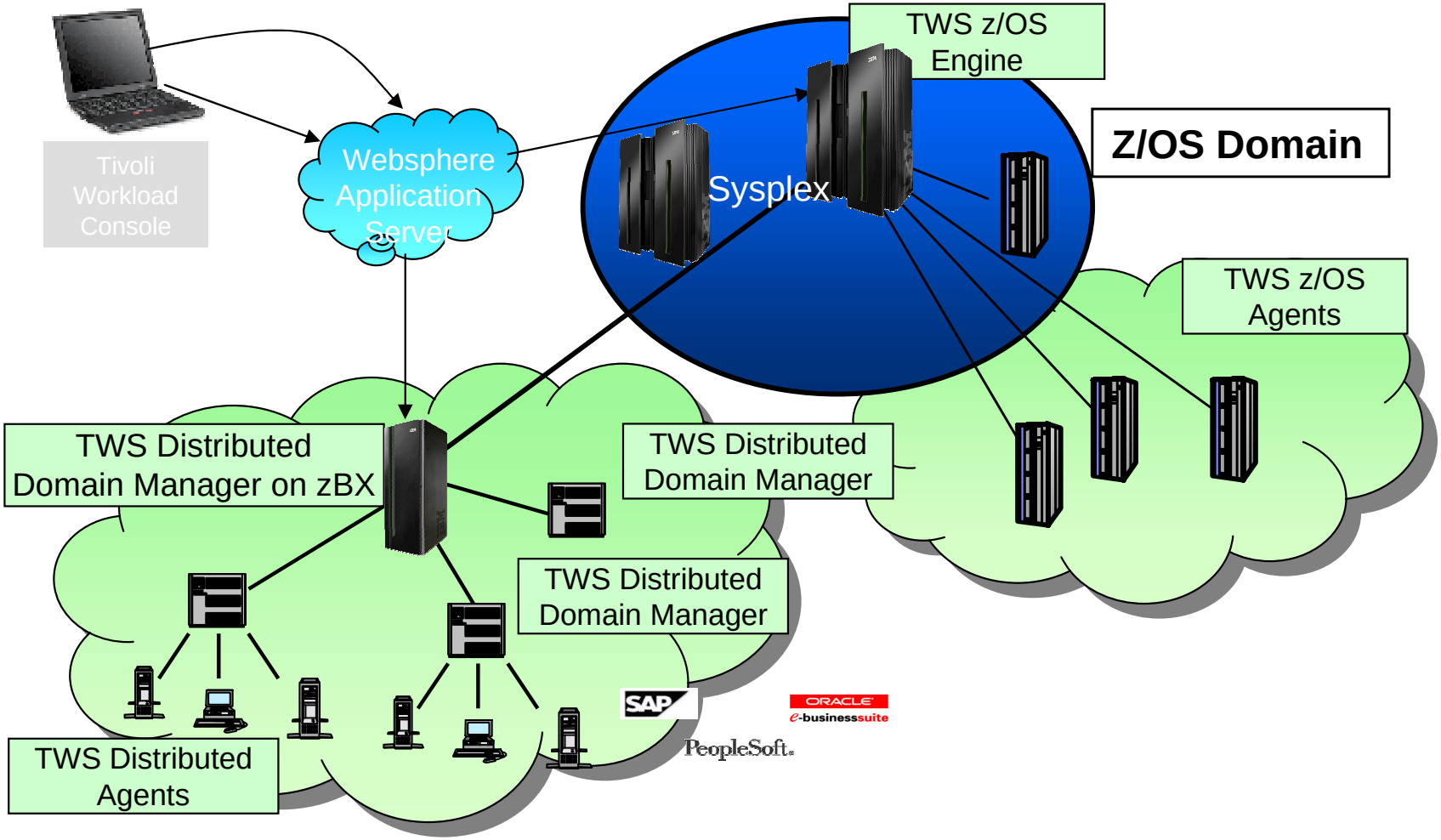
- Centralized Modeling and Control for z/OS only scheduling
- TWS for z/OS Engine
  - Owns the scheduling database, drives all the scheduling activities, and implements all Scheduling control logic
- TWS for z/OS Server
  - Manages communication with TWS components located outside the Controller Addtess Space
- TWS for z/OS Agents
  - Receives jobs to be executed by Engine, interface JES and SMF, sends back events to the Engine.
  - Connected to Engine via XCF, Shared DASD, TCPIP or SNA.

TWS Distributed Configuration



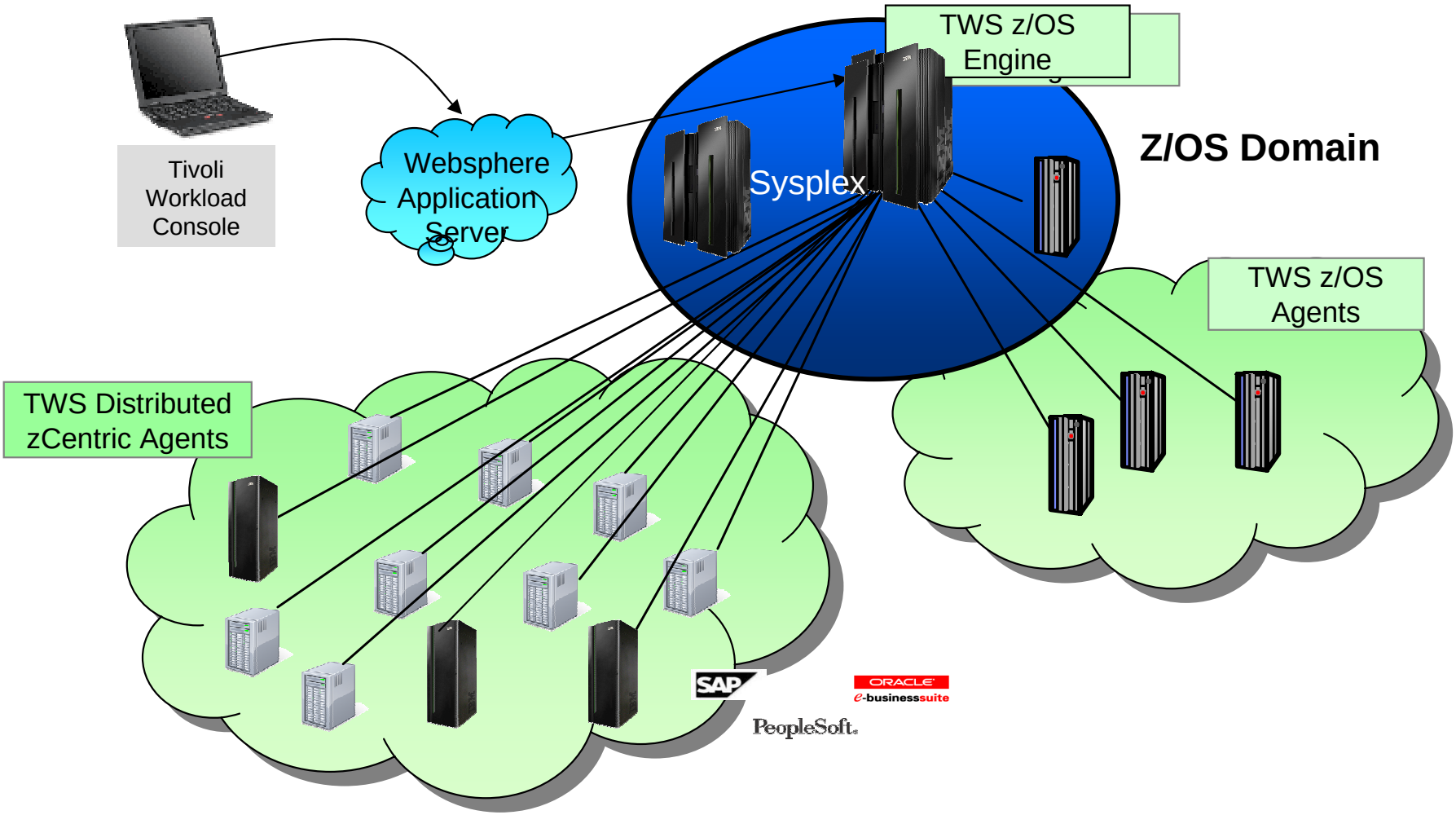
- Centralized Modeling and Control for Distributed only scheduling
- TWS Distributed Engine
  - Owns the scheduling database, collects events from the underlying Agents and resolves cross-domain dependencies
- TWS Distributed Agents
  - Intelligent agents capabale to work autonomously
  - Connectivity to ERP applications

TWS Plan Based End-to-End Configuration



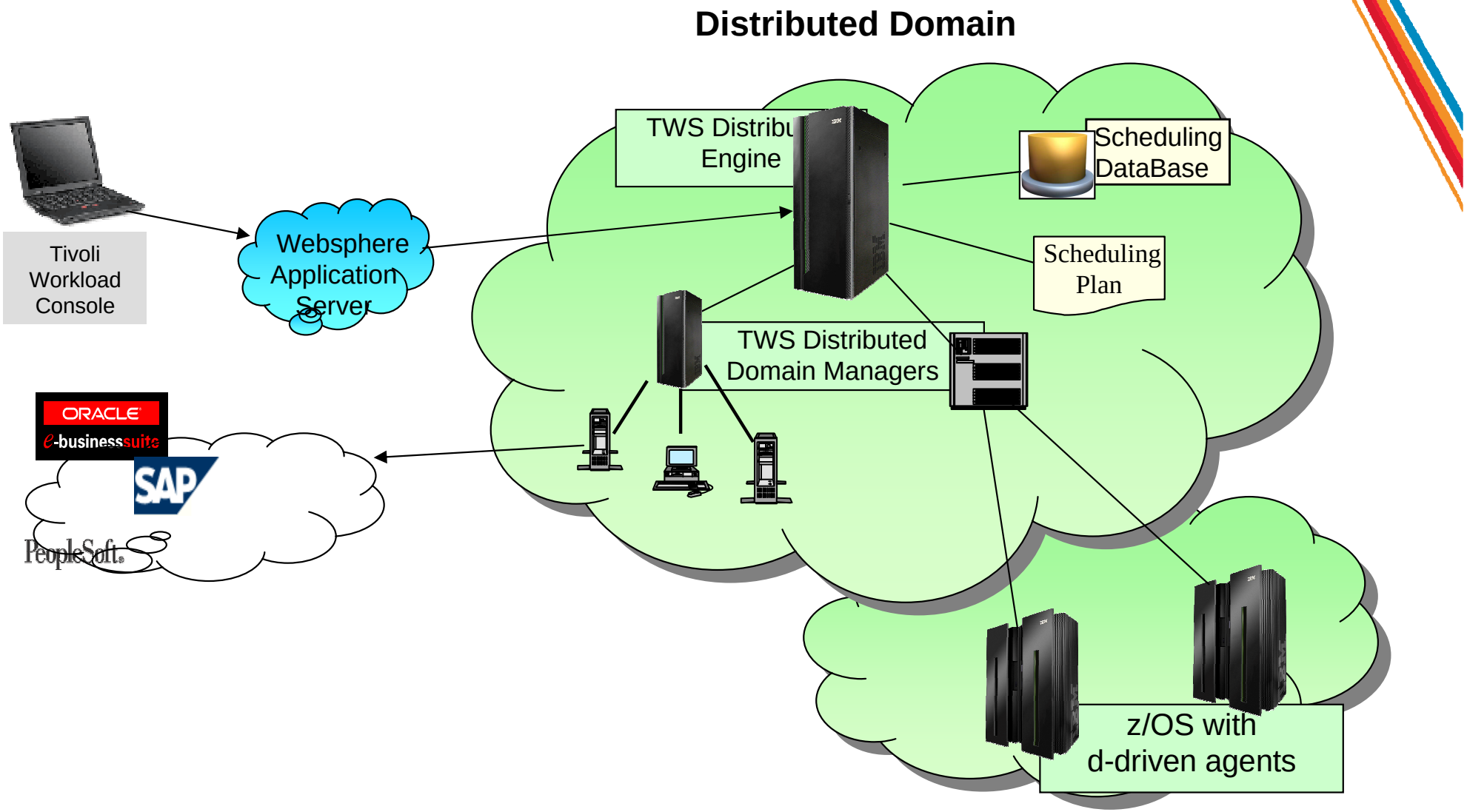
- Centralized Modeling and Control from z/OS environment for all Enterprise scheduling (z/OS and Distributed)

TWS zCentric End-to-End Configuration



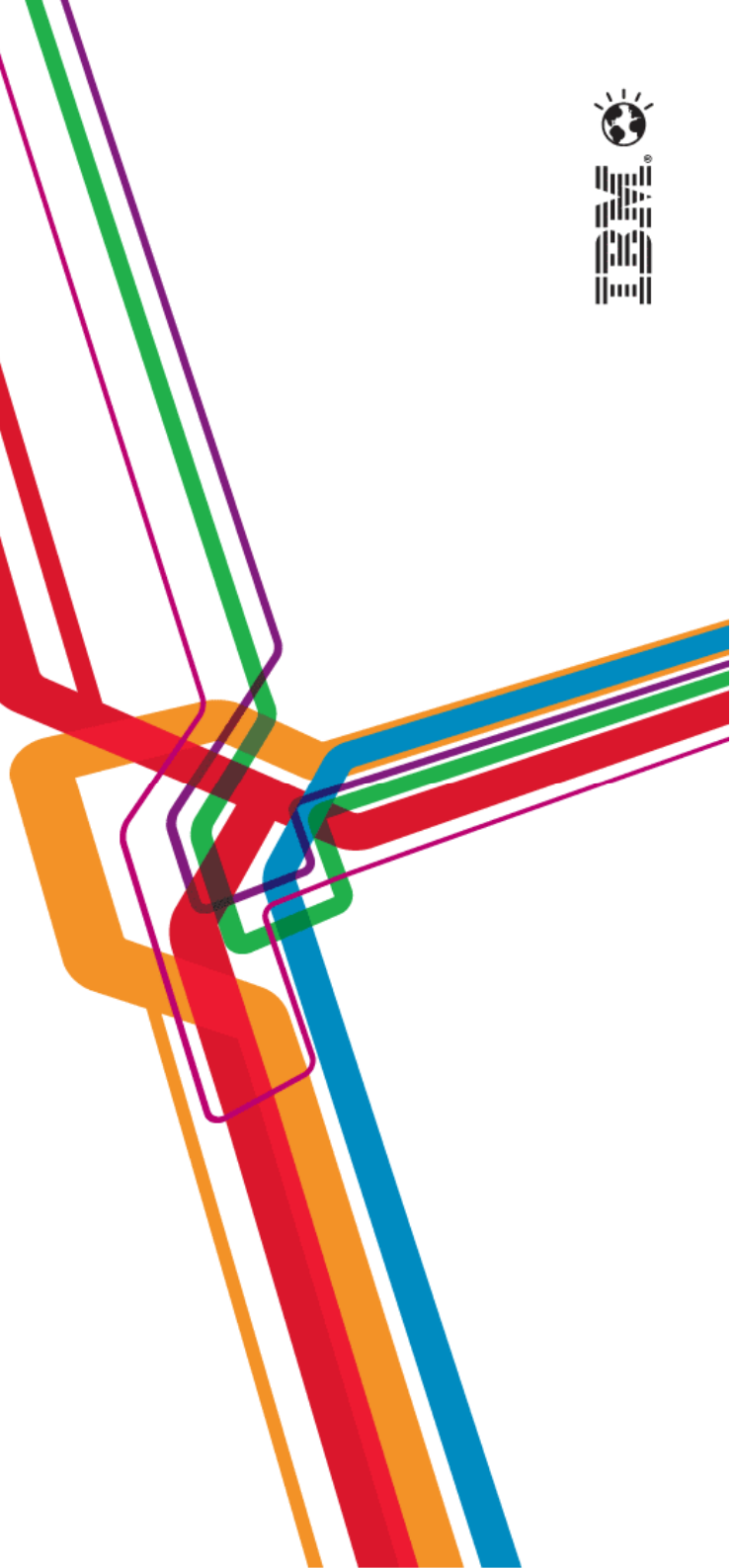
- Centralized Modeling and Control from z/OS environment simple http protocol

TWS d-Driven End-to-End Configuration



- Manage small z/OS batch environments through a Distributed Master Domain Manager

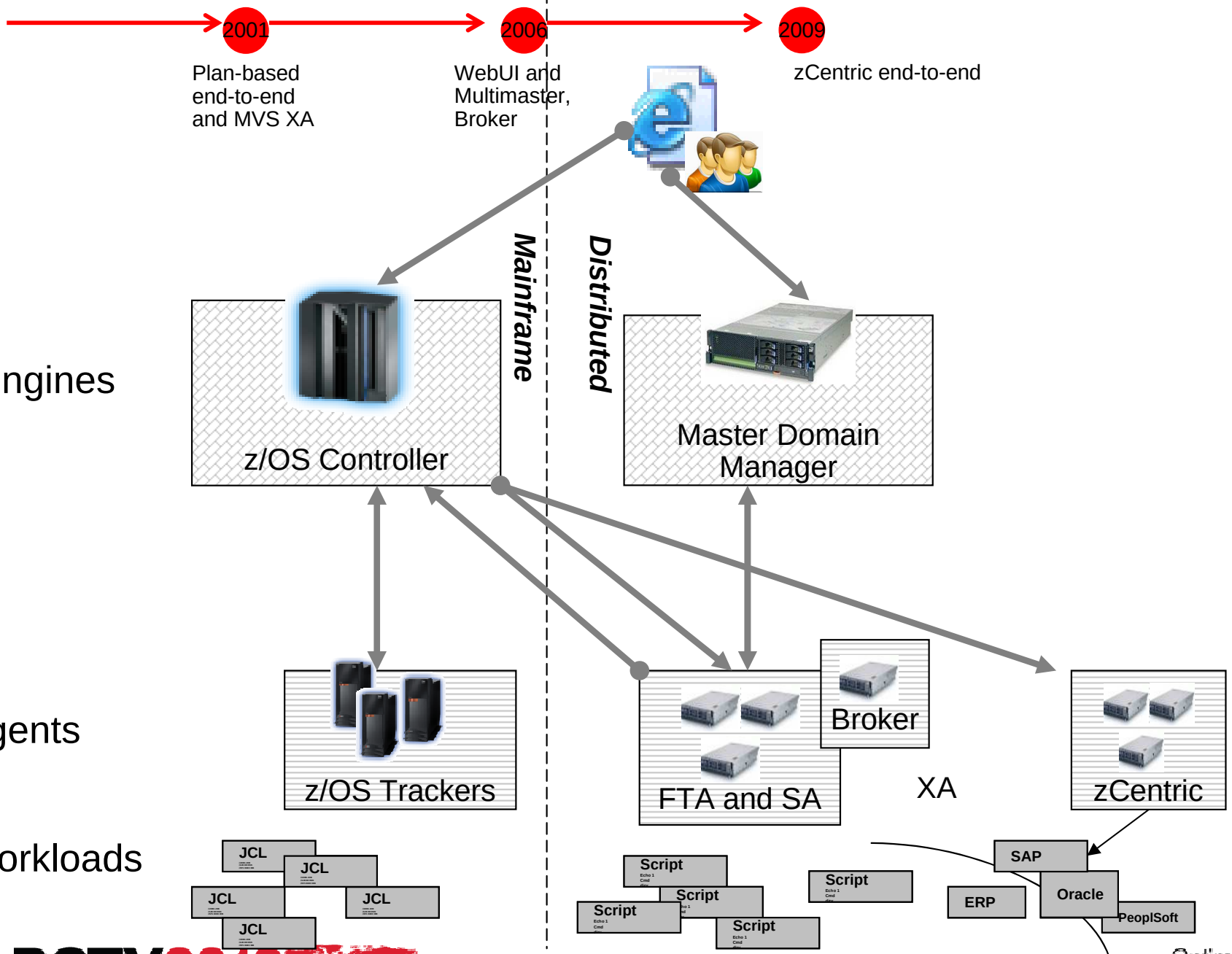
End to End Evolution



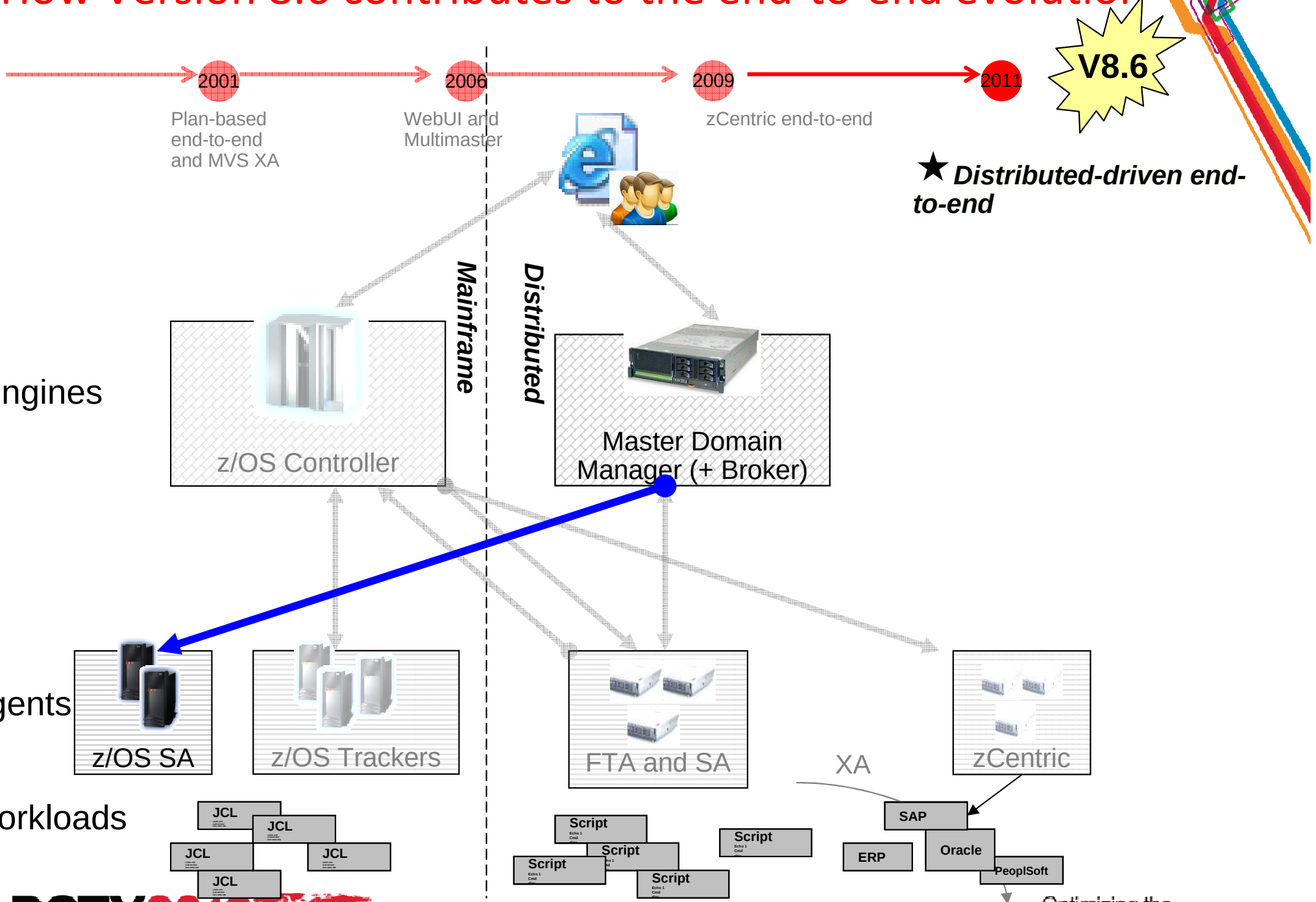
PCTY2012

Pulse Comes to You

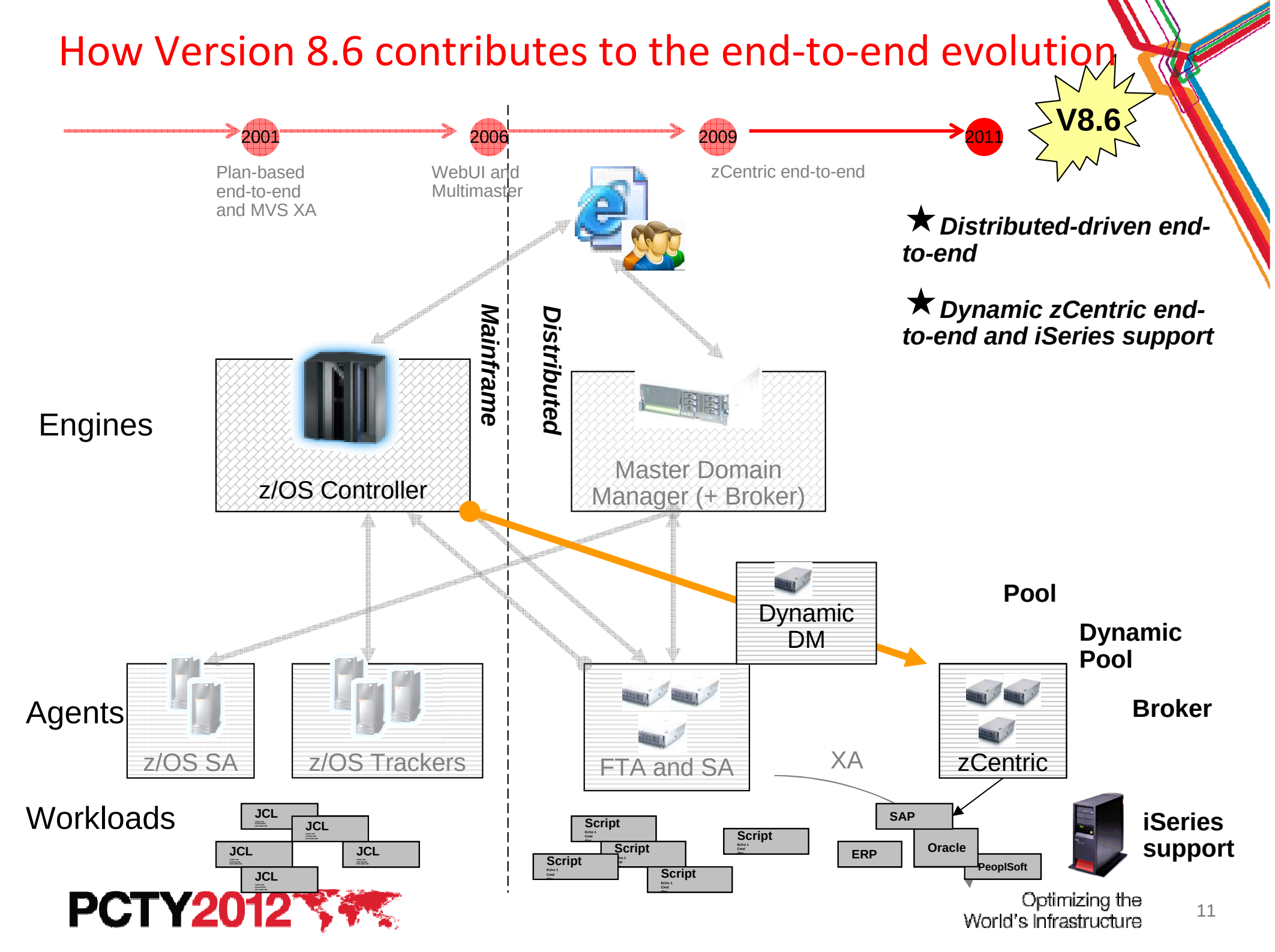
How Version 8.6 contributes to the end-to-end evolution



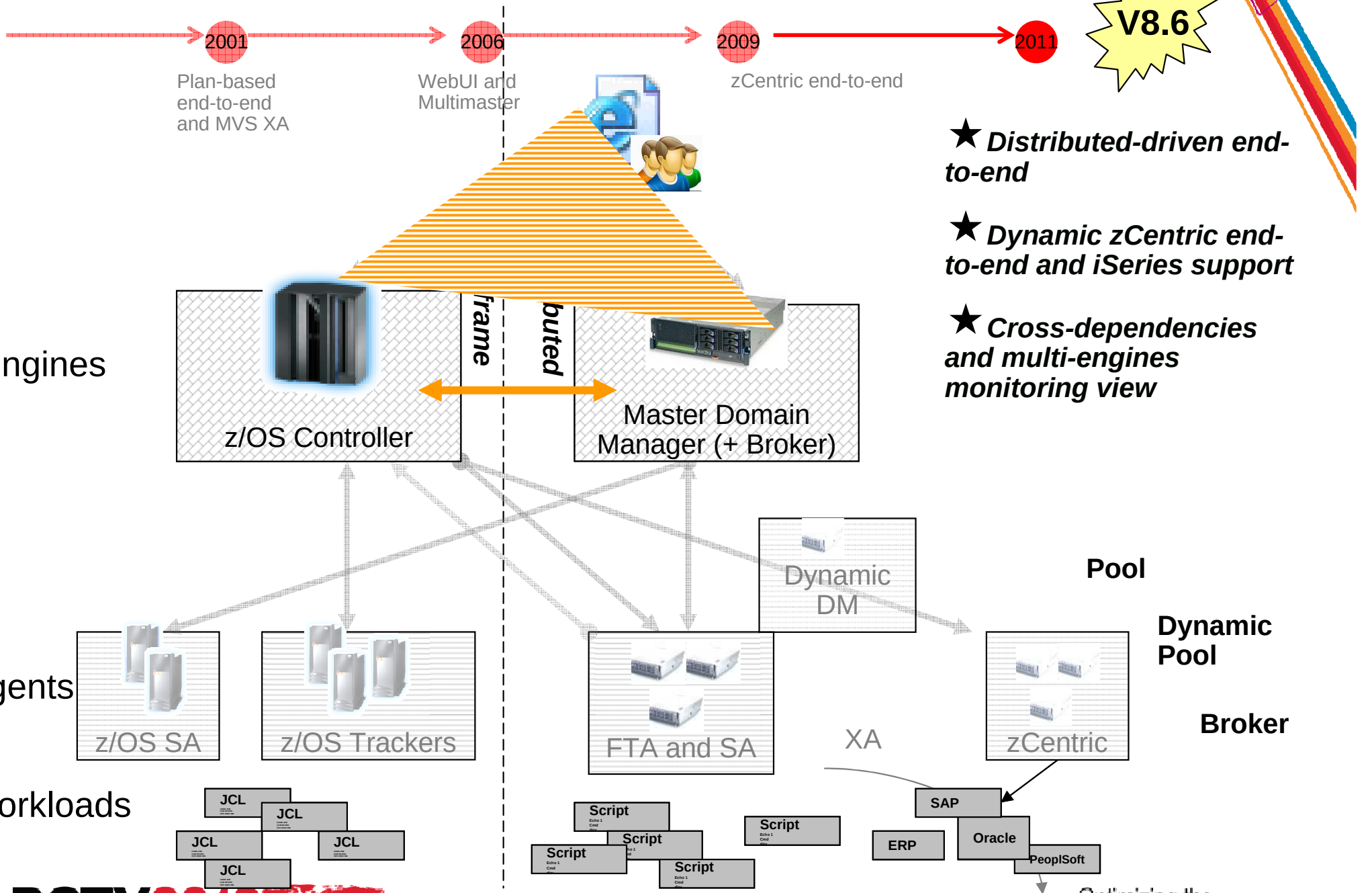
How Version 8.6 contributes to the end-to-end evolution



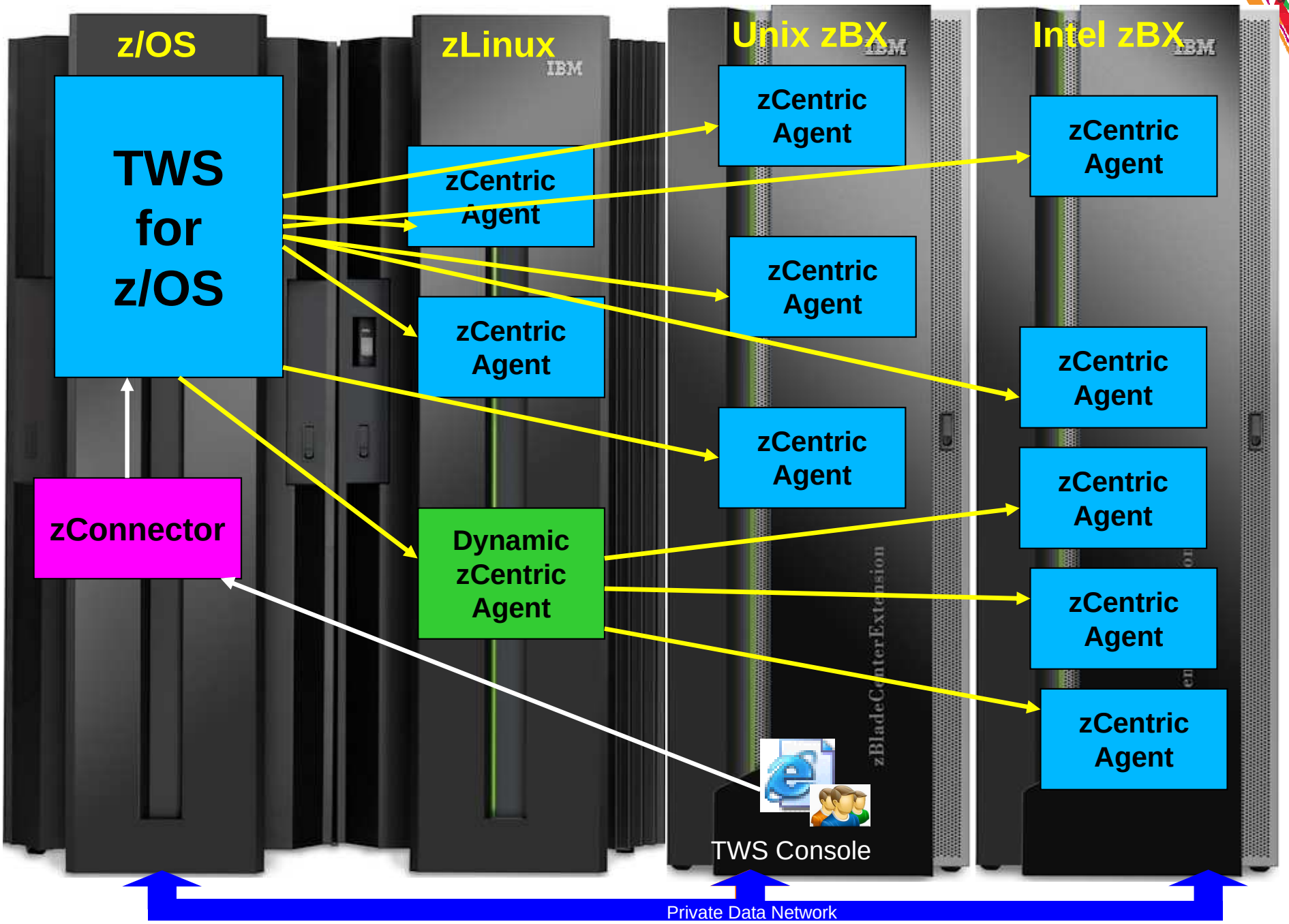
How Version 8.6 contributes to the end-to-end evolution



How Version 8.6 contributes to the end-to-end evolution



Best fit End to End on zEnterprise



Batch Modernization

How Version 8.6 contributes to the Batch Modernization

Examples of batch modernization

Transform: batch applications using modern languages (ex. COBOL to Java)

Re-use: existing applications with business oriented Web Services

TWA V8.6 helps supporting transformation and re-using projects

- **Invoke scheduling services as Java API**
  - **Through zConnector, now running also on z/OS**
- **Enable wrapping existing scheduling services with web services**
  - **Edit and submit jobstreams with variable substitution**
- **Embrace scheduling of Java and Web Services**

★ ***Reduce costs offloading MIPS to zAAP***

★ ***Enable easy remote access to scheduling services***

★ ***Re-use of existing processes running rather than encouraging a re-write***

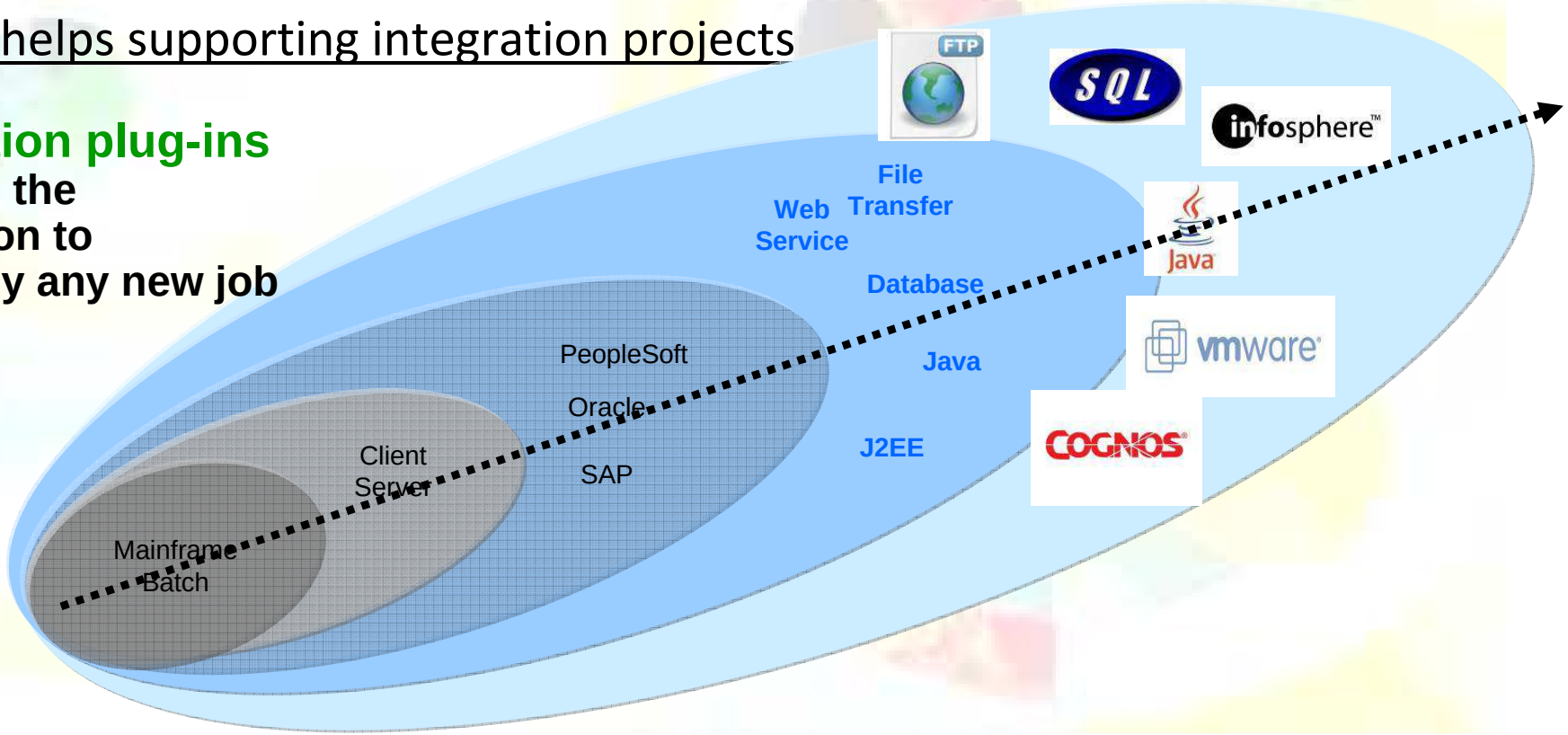
How Version 8.6 contributes to the Batch Modernization

Examples of batch modernization

Integrate: legacy applications with new applications

TWA V8.6 helps supporting integration projects

- **Application plug-ins to extend the automation to potentially any new job types**



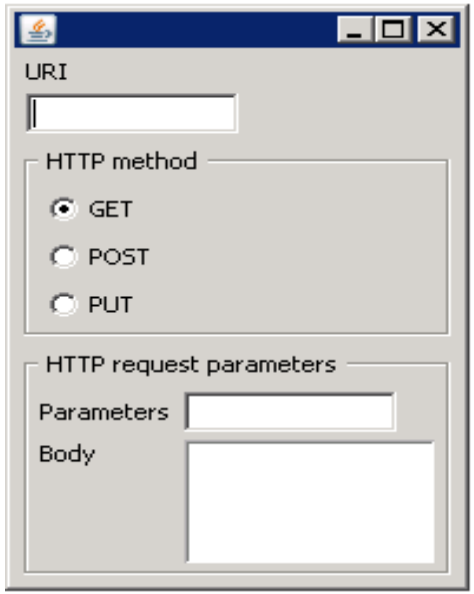
Application Plug-ins

What we offer to 3° parties (Clients, BP, System Integrators) for new Application Plug-ins implementation

- **Tivoli Workload Scheduler Integration Workbench**
  - Takes you through the **creation** of your plug-ins

- Two main phases:
  - Panels creation (AUIML file)
  - Execution method creation (JSDL)

Application Plug-in (jar file)



- After **deployment** of new Application Plug-in, you will be able to manage the new job type in the same fashion as all other TWA job typesd creation (JSDL)

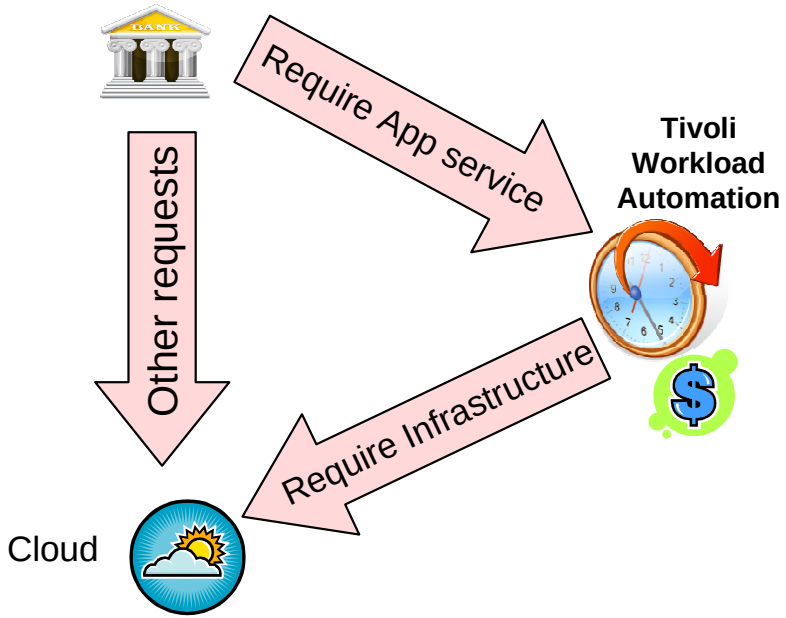
Batch Cloud

Cloud and workload automation

Cloud accelerates the need for workload automation

Value proposition

★ TWS has moved to the center between the request for business services and the demand for Cloud infrastructure and resources: it allows to manage mission-critical, end-to-end workloads through fluid and elastic cloud resources



Capabilities

- **TWS v8.6** provides TSAM integration for
  - Immediate provisioning of workload automation ready environment, for temporary demand
  - Elastic provisioning of workload automation resource for intermittent demand

Business Value

★ ***Reduce resource, installation and customization costs and increase productivity***

★ ***Adapt quickly in the context of limited resources***

Cloud workload automation benefits

- **Elastic scaling**
  - Automatic provisioning of additional TWS agents in case of delay in processing TWS workflows. Delay triggers TSAM request for new agent provisioning.

- **Provisioning of a batch-ready environment**
  - Request through TSAM a batch-ready execution environment. The owner will be able to use that environment and schedule jobs.

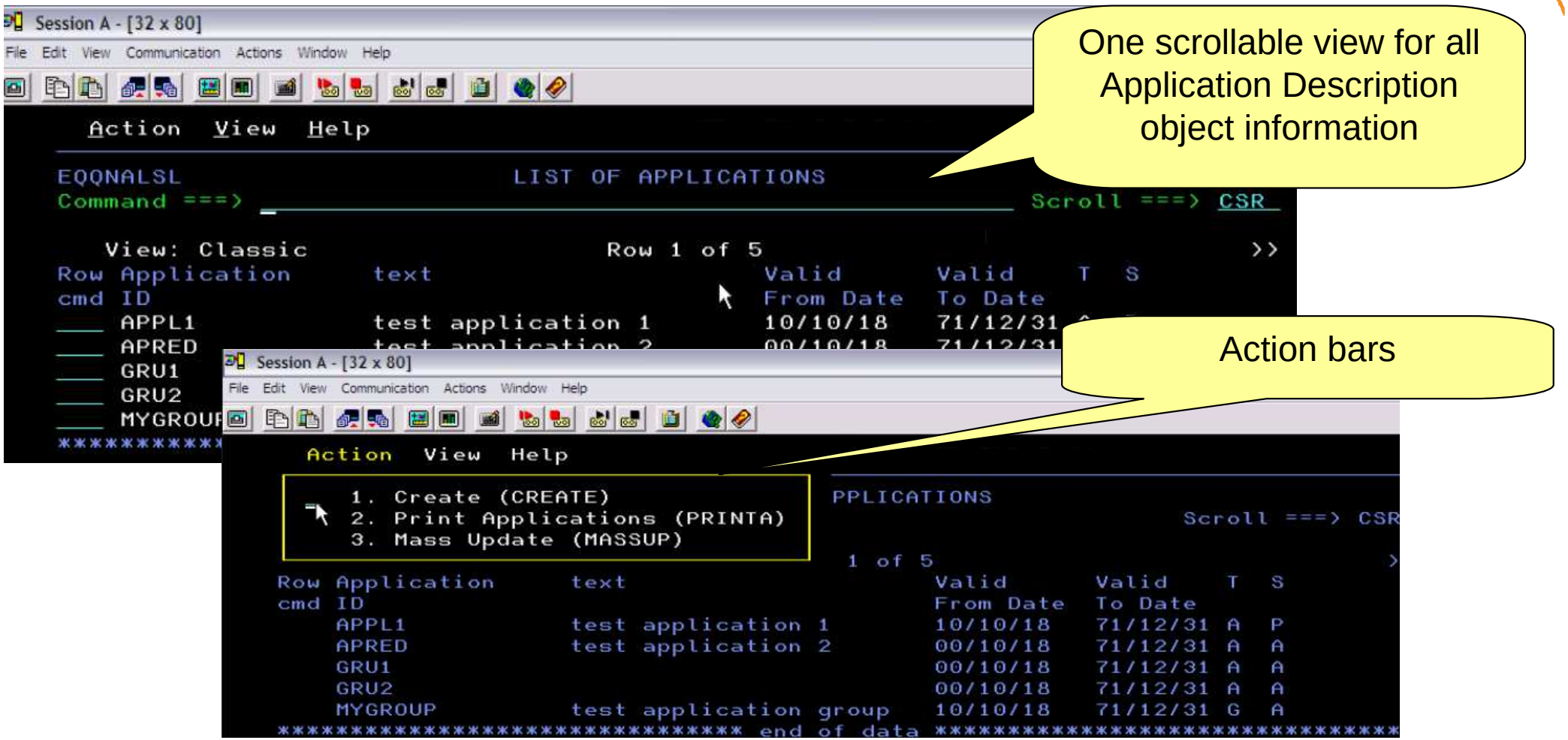
- **WEE failover**
  - Automatic provisioning of a server capable to satisfy job resource requirements, to recover the a job in error. TSAM is invoked as a recovery action and it provides the needed server.

Usability

New ISPF panels

Scenarios

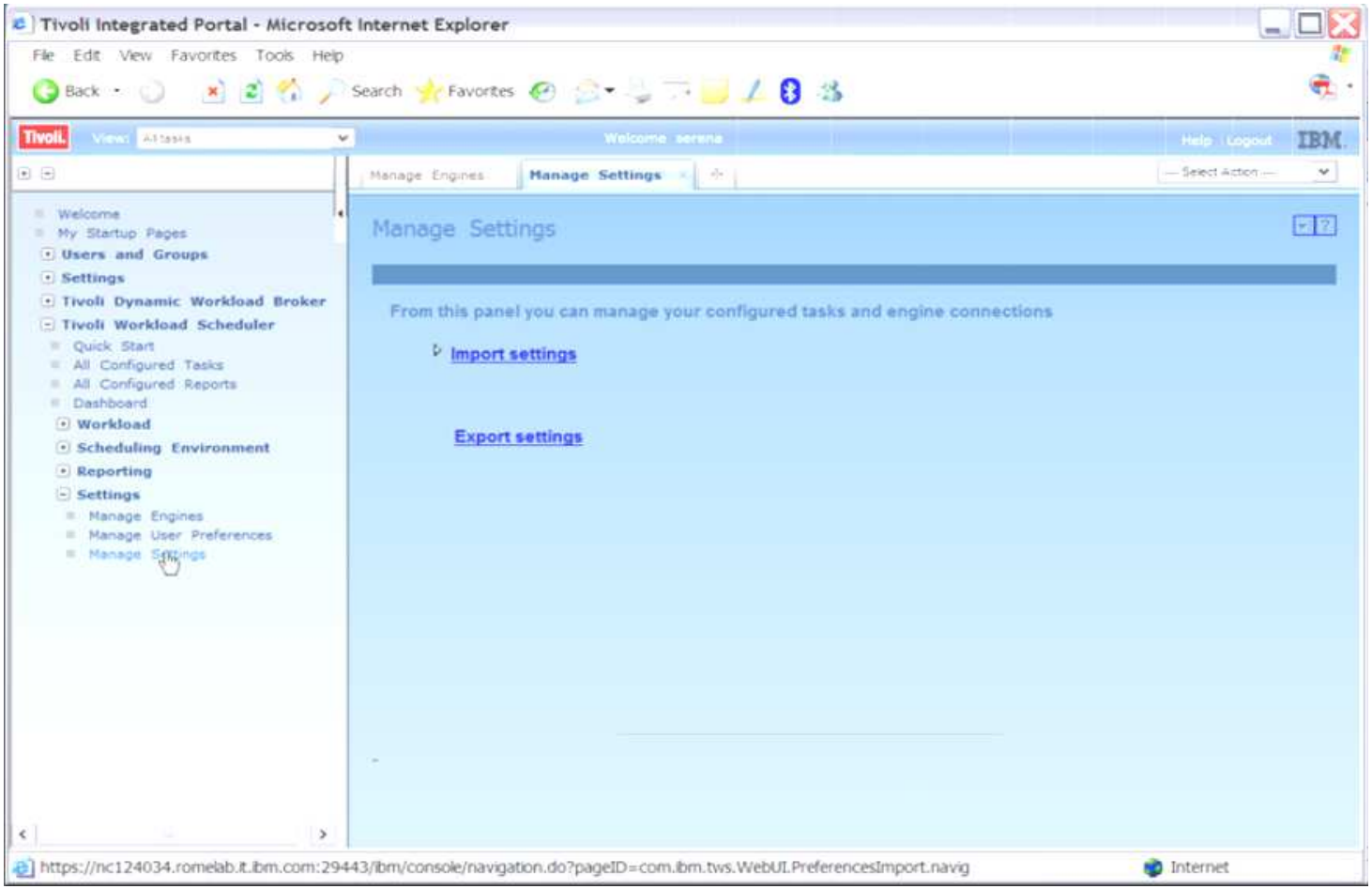
- *Customers require to rapidly navigate across ISPF TWS Dialogs, to accelerate operations, increase speed-up and more confidence with the product interface.*



One scrollable view for all Application Description object information

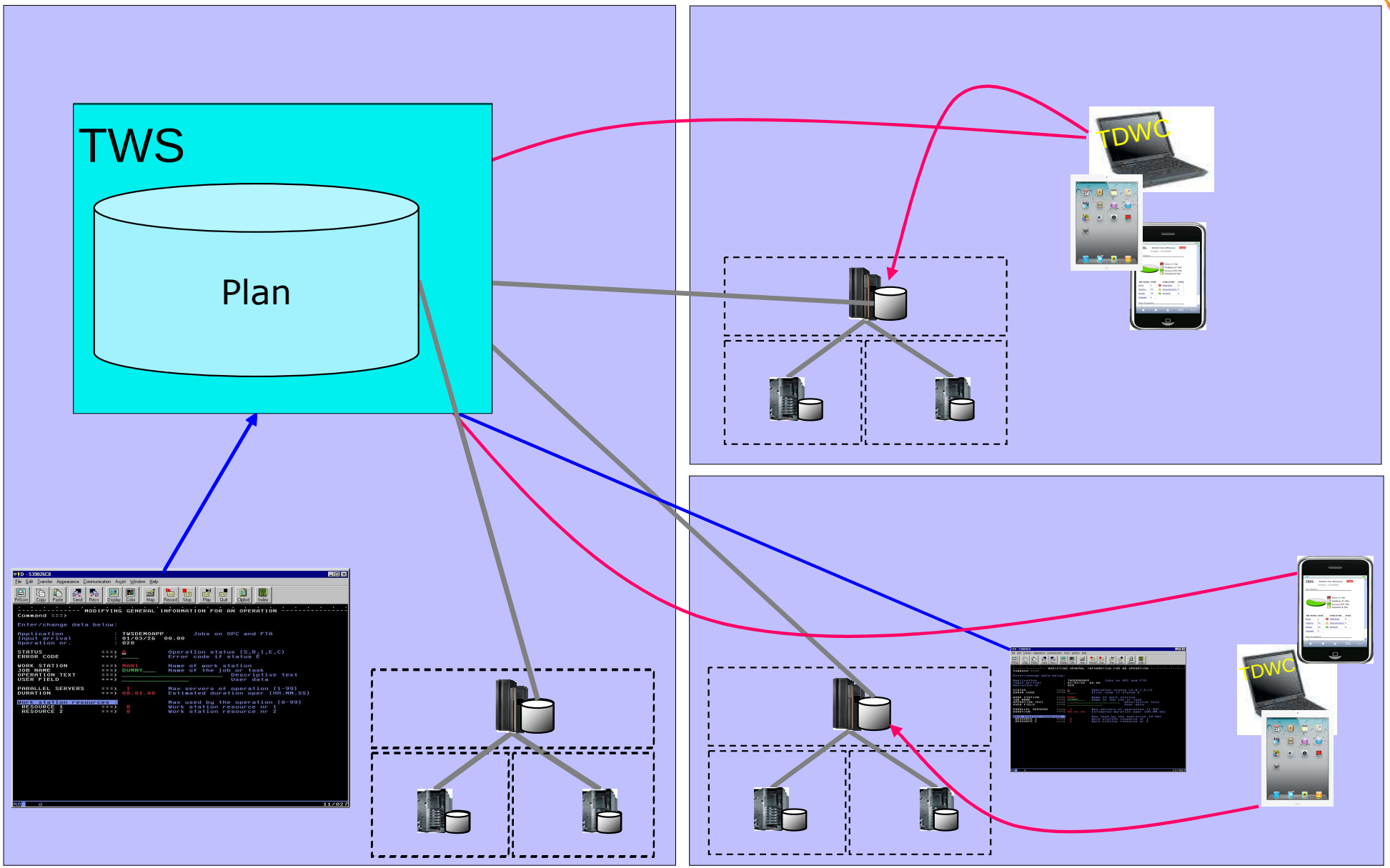
Action bars

Tivoli Dynamic Workload Console



- XML support
- Easy to unload definitions (engines, tasks)
- Easy to import unloaded information

Operating on demand



Thank you

obrigado
merci
gracias
graciñas
hvala
obrigada
dank
thank
tesekkur
gratias
gràcies
gråces
rahmat
salamat
plural
tänan
takk
khun
asante
kasih
kié
wel
terima
mauruuru
mamnun
aabhar
much
pers
mèsi
grazie
basque
kop
tapadh
ni
akoze
mura
dziękuję
enkosi
leibh
edirem
faafetai
didi
kyay
shukriya
nirringrazzjak
arigatô
grazzi
bayarlalaa
blagodaram
dhanyavaad
dhanyabaad
sagolun
grassie
respectful
gracie
sheun
dhanyavadalu
blagodaria
madloba
mahalo
aahe
informal
saha
Barak
maith
dhanyavadagalu
familiar
gratzias
misaotra
meharbani
aguyjé
ju
akun
sobodi
chai
je
hamnida
sag
ago
kha
dankie
danke
chnorakaloutioun
Allahu
khob
krap
najis
ederim
milesker
bedankt
leat
dankon
akpé
formal
singular
mercé
mersi
maita
kiitos
deu
nanni
beacôp
asko
cám
manana
sana
dakujem
terima
diakuiu
agimus
dhanyavad
sah
faleminderit
grandmercé
diolch
dîky
aabhari
olun
muchas
nižžik
ačiū
ngiyabonga
dziakuju
dankewol
po
pa
ko
fyri
nandri
agat
raibh
kam
lava
paldies
köszönöm
chokrane
děkuji
dieuf
marahaba
barkal
northern
djiere
efharisto
eskerrik
motashakkeram
danki
côps
mulţumesc
fiik
ekkur
mochchakkeram
nirringrazzjak
arigatô
grassie
sagolun
gracie
sheun
aahe
mahalo
madloba
blagodaria
dhanyavadalu
respectful
dhanyabaad
dhanyavaad
bayarlalaa
grazzi
ju aguyjé
akun
sobodi
chai
je
hamnida
sag
ago
kha
dankie
danke
chnorakaloutioun
Allahu
khob
krap
najis
ederim
edirem
faafetai
didi
kyay
shukriya
mamnun
pers
mèsi
thank
dank
terima
mauruuru
grandmercé
diolch
dîky
aabhari
olun
muchas
nižžik
ačiū
ngiyabonga
dziakuju
dankewol
po
pa
ko
fyri
nandri
agat
raibh
kam
lava
paldies
köszönöm
chokrane
děkuji
dieuf
marahaba
barkal
northern
djiere
efharisto
eskerrik
motashakkeram
danki
côps
mulţumesc
fiik
dhanyavadagalu
Barak
maith
saha
informal
tänan
familiar
gratzias
misaotra
meharbani
aguyjé
ju