Migration of Control-M to Job Scheduler

Integration of Job Scheduler
in a TWS Landscape

© SOS Software GmbH Berlin, Germany

April 2011

Content

- basic conditions I & II
- previous landscape (Control-M)
- current landscape (Job Scheduler)
- basic data
- installation
- migration (overview)
- migration (workflow)

Basic Conditions I

- complete replacement of Control-M (incl. server)
  - the scheduling software has to be installed on the application server
  - all scheduling tasks have to be handled by Job Scheduler
- TWS is still the enterprise tool for software automation
  - It is necessary to implement the communication between Job Scheduler and TWS
  - no changes in the TWS environment
  - the configuration of the TWS jobstreams have to remain unchanged

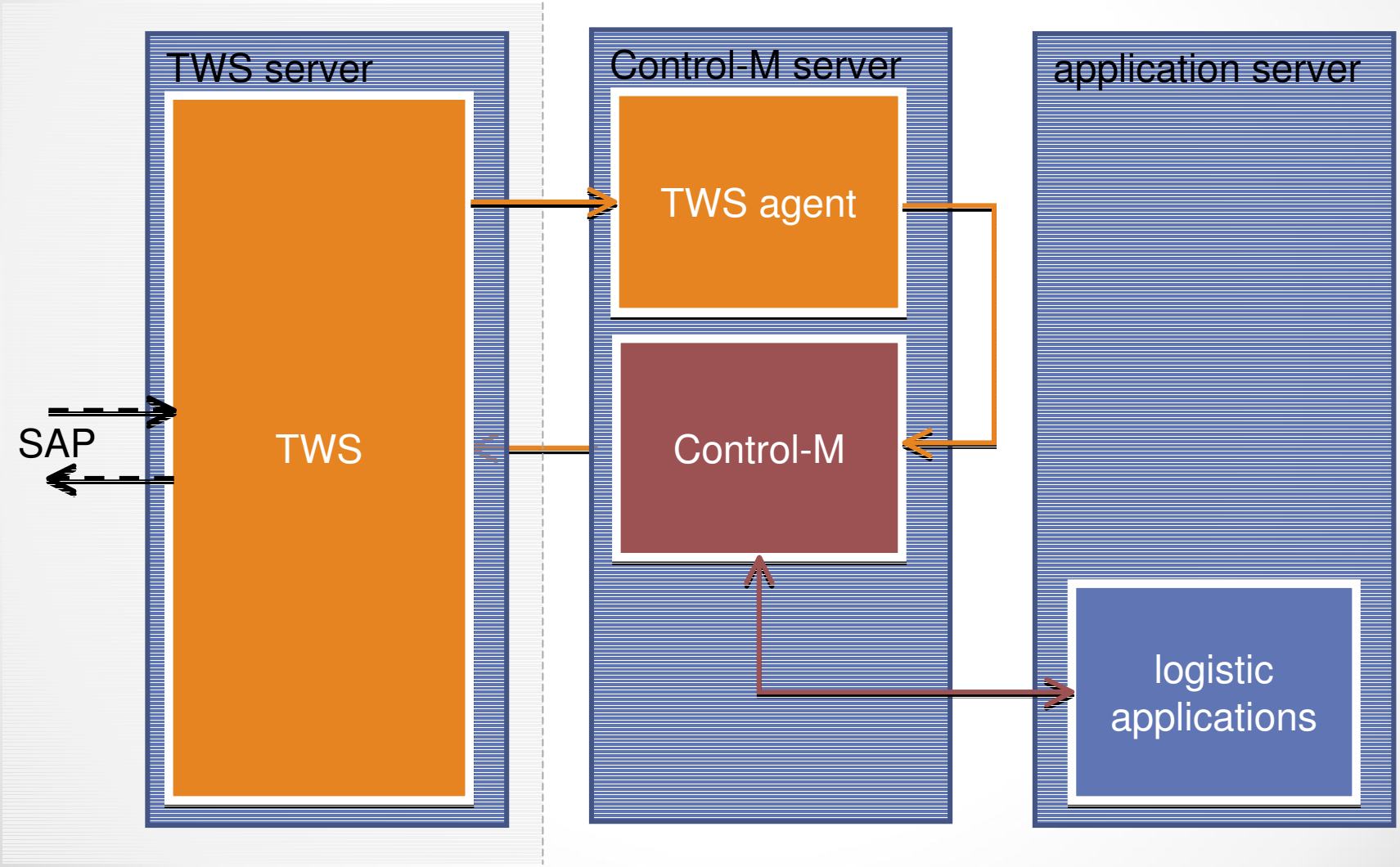
Basic Conditions II

- security conditions
  - the security conditions of the customer have to be considered (e.g. no direct ssh connection to the TWS server)
- less migration effort
  - the effort is 60 days
- cost reduction
  - approx. 80% savings
  - ROI < 1 year

Previous Landscape (Control-M)

- „Logistic Cluster“
  - two application server (production & development)
  - development server is fallback for production
- Job scheduling controlled by Control-M
  - one Control-M installation on a separte server
  - no distinction between production & development
  - Control-M handles the communication with TWS
  - Control-M handles the batch operations for the logistic applications
- TWS communication controlled by an TWS agent
  - a TWS agent is running on the Control-M server
  - it handles the communication with the TWS server

Previous Landscape (Control-M)

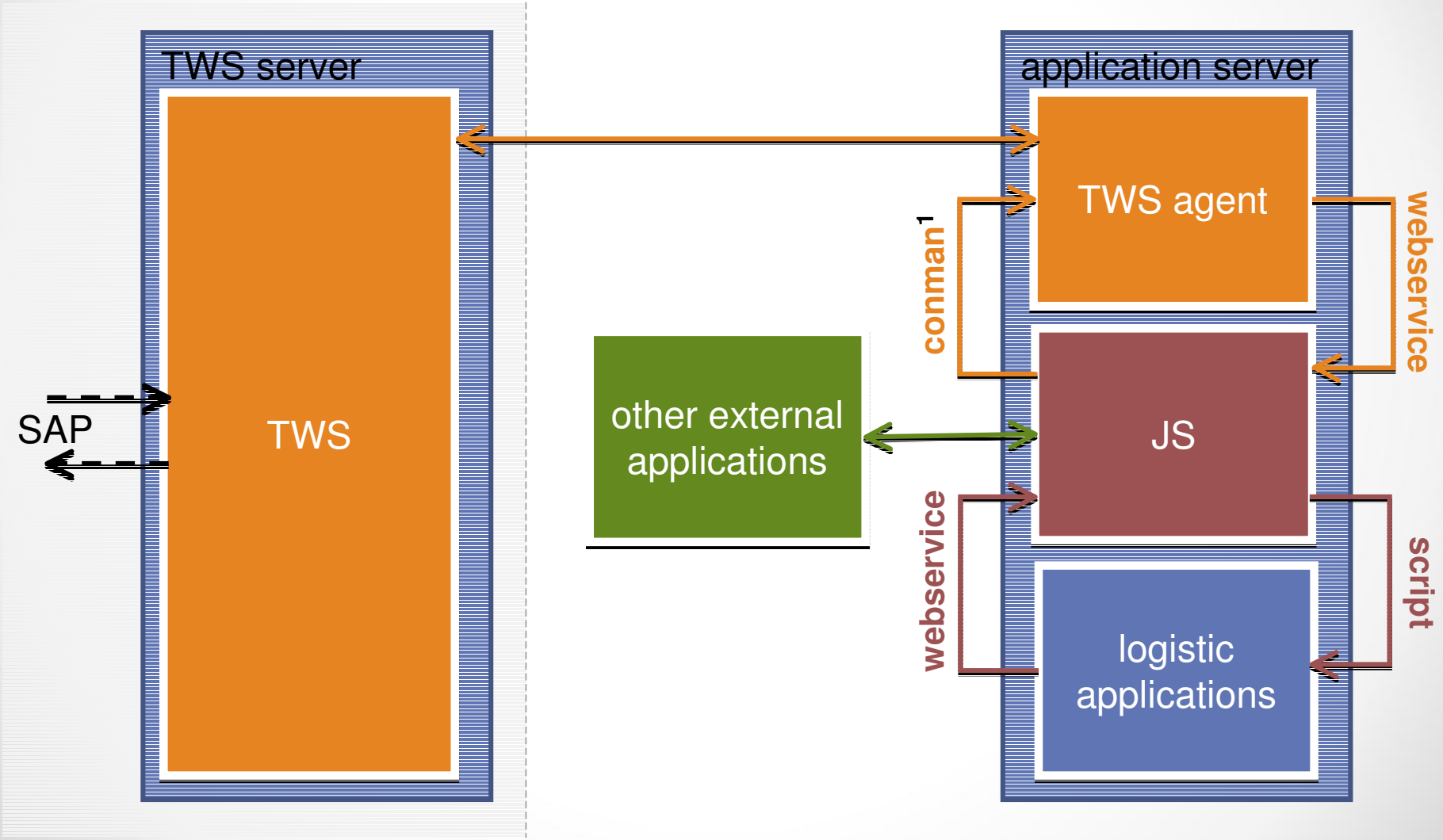


Current Landscape (Job Scheduler)

- „Logistic Cluster“
  - Two application server (production & development)
  - Development server is fallback for production
- Job scheduling controlled by Job Scheduler (JS)
  - one instance of JS installed on each application server (production & development)
  - JS handles the communication with TWS
  - JS handles the batch operations of the logistic applications
- TWS communication controlled by an TWS agent
  - a TWS agent is running on the application server
  - it handles the communication with the TWS server

Current Landscape (Job Scheduler)

[1] conman is a commandline tool of the TWS agent



Basic Data

- configuration of Job Scheduler
  - 600 jobs
  - 250 job chains
  - 100 orders
  - TWS communication via TWS agent
  - external job start via Web Service
- database
  - one database shared by two instances of Job Scheduler (production & development)
  - Oracle 11
  - 50 GB tablespace

Batch Installation

- installation of Job Scheduler in batch mode
  - configuration files just differ in a few parameters (e.g. the server names)
  - garantie for exactly the same installation on all systems
  - easy to reinstall

Migration Overview

- export the Control-M configuration
  - Control-M provides its job configuration in XML
  - one xml file per Control-M table
- automated migration
  - providing a set of XSLT stylesheets to handle the migration
  - controlled by ANT
  - 6 steps workflow:
    - configuration
    - data collection
    - build addtional ressources for Job Scheduler
    - generating Job Scheduler objects
    - provide a release version of the live folder
    - deployment to the application servers
  - < 5% manual adaption

Migration Workflow

- **configure**
  - configure a file with all CTM tables to migrate
- **collect**
  - create a „collection“ of the basic data of all jobs ★
  - create manual JS objects in singular cases
  - adapt the given Control-M data in the „collection“
  - expand the „collection“ with some „payload“ per job
- **build**
  - generate additional JS objects ★
    (e.g. syncpoint jobs, process classes, locks)
- **generate**
  - generate JS objects based on the „collection“ ★
    (jobs, orders, job chains)
- **release**
  - create a local release version of the live folder ★
- **deploy**
  - deploy the live folder to the application servers ★

★ = automated steps (XLST)