

Pacific Application Server pour OpenEdge

Introduction

David Cleary
Principal Software Engineer
November 20, 2014

Traduction par Gilles QUERRET
Riverside Software
PUG France – 16 décembre 2014

PROGRESS
EXCHANGE 2014

Disclaimer

- This presentation is for informational purposes only. You are cautioned that any information contained in this presentation may change in the course of product development.
- This presentation may not be interpreted as any commitment on behalf of Progress and future development, timing and release of any products, features or functionality described in this presentation remains at the sole discretion of Progress.

***Final ESAP Release
Yesterday
Gold Candidate Last
Night***

Disclaimer 2

- Cette presentation n'est pas délivrée par une personne de Progress Software, donc les informations contenues dans ces slides sont à prendre avec encore plus de pincettes
- Pour tout reclamation, contactez directement Laurent Kieffer 😊

Agenda

- Introduction PASOE
- Architecture PASOE
- Performance et montée en charge

Introduction PASOE

Plateforme Pacific Application Server

- **Créé à partir d'une distribution Apache Tomcat 7.0.55**
- **Configuré par défaut en mode “production”**
 - L'application ROOT par défaut est remplacée
 - L'application Tomcat Manager n'est pas installée
 - Déploiement automatique désactivé
 - Port d'arrêt désactivé sous Unix
 - JMX désactivé
 - Filtre des crawlers Web activé
- **Spring Security Framework**
- **Realms et roles définis pour configure le contrôle d'accès**
- **Lignes de commandes spécifiques pour la configuration et la gestion du serveur**
- **Implémentation par défaut pour Progress Rollbase et Progress Corticon**

Pacific Application Server pour OpenEdge (PASOE)

- **Serveur web fusionnant en un seul paquet les fonctionnalités de :**
 - Tomcat
 - AppServer classique (celui actuellement livré avec OpenEdge)
 - Adaptateurs AppServer
- **Fonctionne sur la plateforme unifiée Pacific Application Server**
 - Même serveur web que pour les installations Rollbase et Corticon
 - Fonctionnalité d'administration et de configuration
- **Uniquement disponible sur les plateformes 64 bits** (Linux, Solaris, HPUX-IA, AIX, Windows 2008/2012)
- **Deux produits: “Development server” et “Production server”**

Motivations

- **Serveur Web sécurisé**
 - Installation, administration
- **Plus simple**
 - Administration, montée en charge, migration d'application, déploiement
 - Modes de connexion AppServer et STATEs
- **Customer Extensible**
 - API REST ouvertes pour le développement de métriques personnalisées, la supervision et l'administration
 - Personnalisation de l'installation
- **Meilleurs outils d'analyse**
 - Collecte automatique de métrique et du statut des requêtes en cours
- **Plus rapide et optimisé**
 - Exécution de code identique avec moins de mémoire et de CPU

PASOE - Production vs Development

PAS for OE Development

- Configuration non sécurisée
- Instance de test dans \$WRKDIR
- Administration à distance
 - Tomcat remote admin activé
 - OpenEdge remote admin activé
- Nombre limité de requêtes simultanées

PAS for OE Production

- Configuration sécurisée
- Pas d'instance de test
- Administration à distance optionnelle
 - Tomcat remote admin optionnel
 - OpenEdge remote admin optionnel
- Couches “transports” désactivées
- Non limité

Pacific Application Server for OpenEdge Architecture (PASOE)

PROGRESS
EXCHANGE 2014

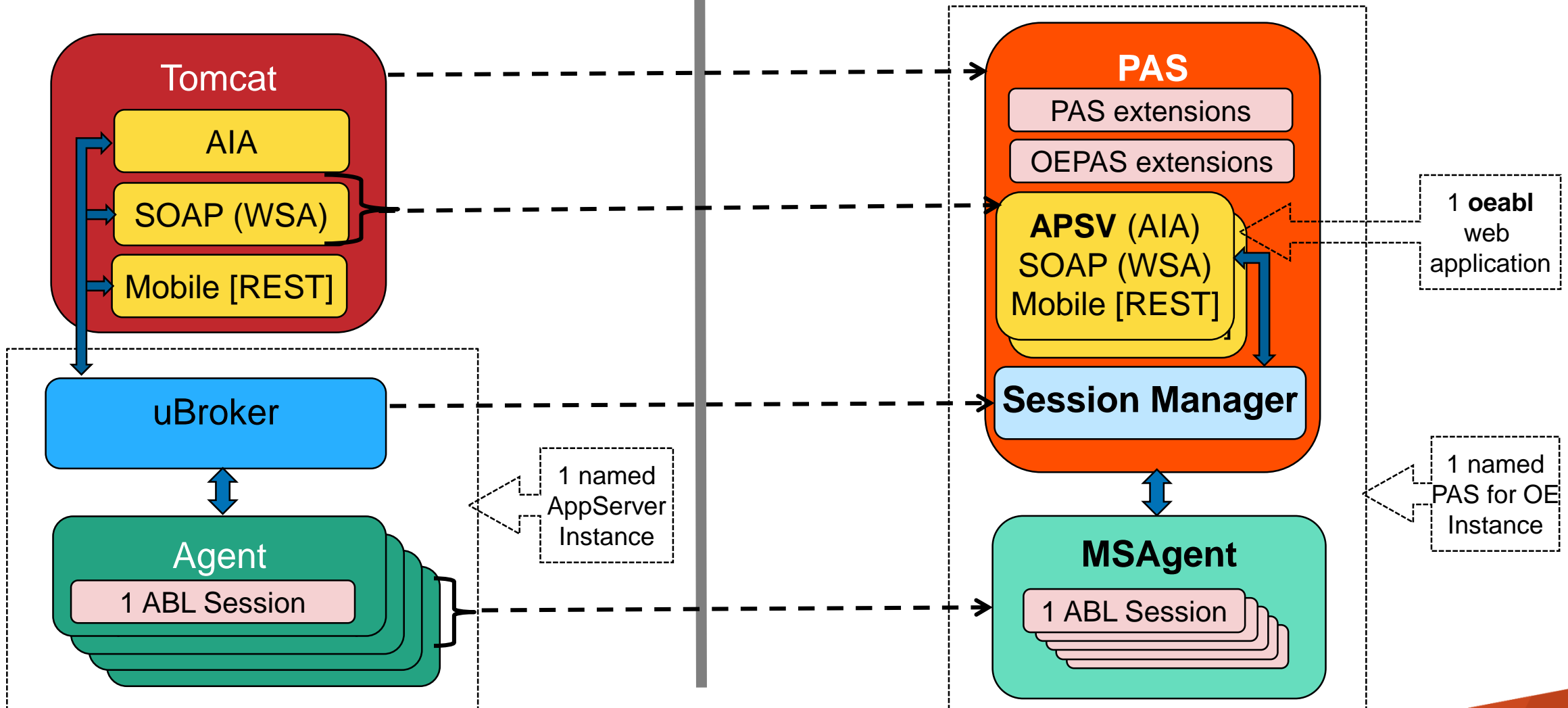
PAS for OpenEdge - Concepts d'architecture et abréviations

- **PAS** est un serveur d'applications Web avec les extensions Progress
 - Application optionnelle d'administration à distance par texte, html ou JMX
- **Services ABL pour Pacific Application Server (ABLPAS)**
 - Collection d'applications Web Java installée sur un PAS
 - Ces applications utilisent le processus "Agent multi-session" (**MSAgent**)
 - Application Web *optionnelle* d'administration à distance (par API REST)
 - Les services Web **ABLPAS** peuvent être installés sur d'autres produits PSC
- **PASOE** est un PAS avec les services **ABLPAS** préinstallés
- Vous concevez, empaquetez, déployez, configurez et déboguez votre application ABL dans une application Web

PAS for OpenEdge - Concepts d'architecture et abréviations

Couche AppServer web classique

Couche Web PAS for OpenEdge



Classic AppServer Instances versus PAS for OpenEdge Instances

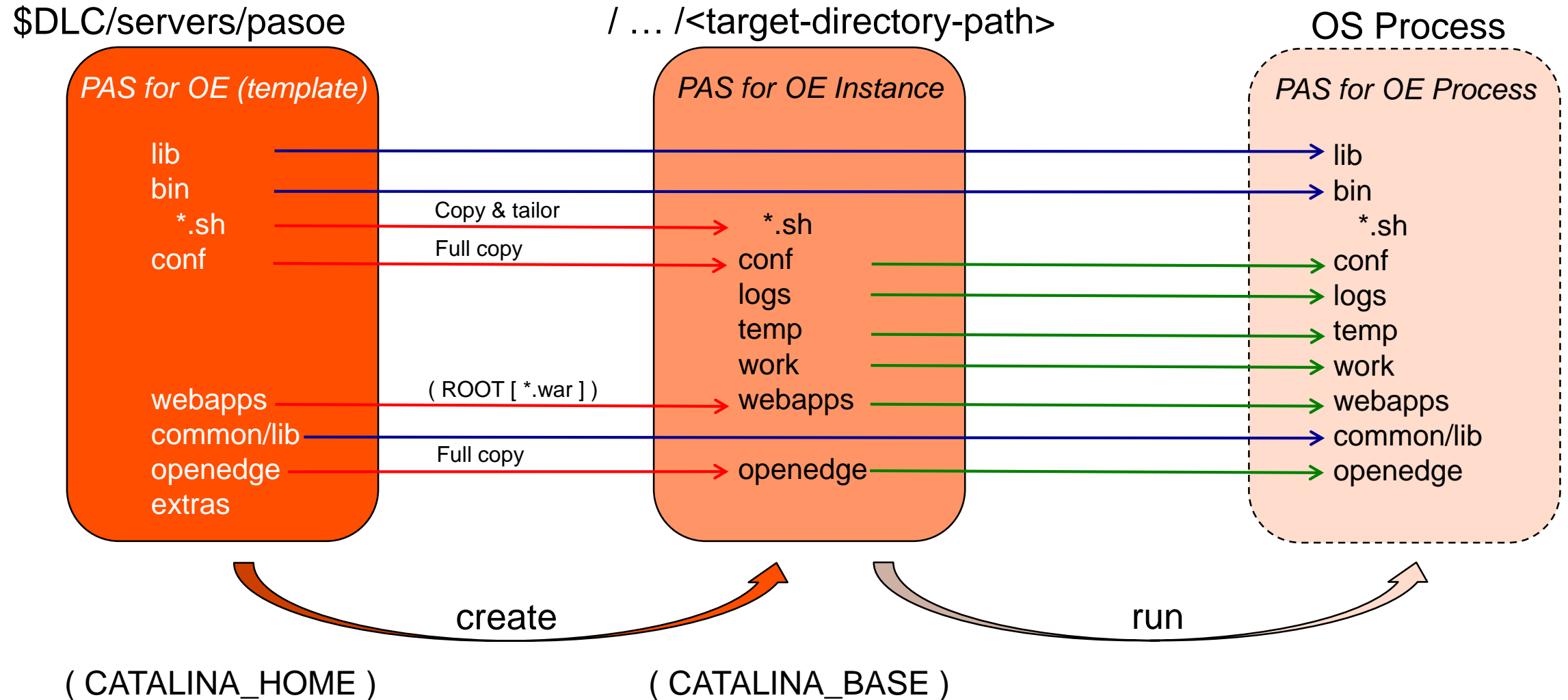
AppServer classique

- Chaque instance d'Appserver a un nom
- Les instances sont définies dans le repertoire d'installation de Progress
- Les instances sont supprimées lors de la désinstallation
- Les instances de broker et d'agents ne sont pas transférables
- Pas de packaging d'une instance

PAS for OpenEdge

- Chaque instance PAS for OE a un nom
- Les instances sont définies en dehors de l'installation Progress
- Les instances ne sont pas supprimées lors de la désinstallation
- Les instances sont transférables
- Une instance peut être package et redéployée

Understanding PAS for OpenEdge Instance Run-time



Classic AppServer Configuration versus PAS for OE Configuration

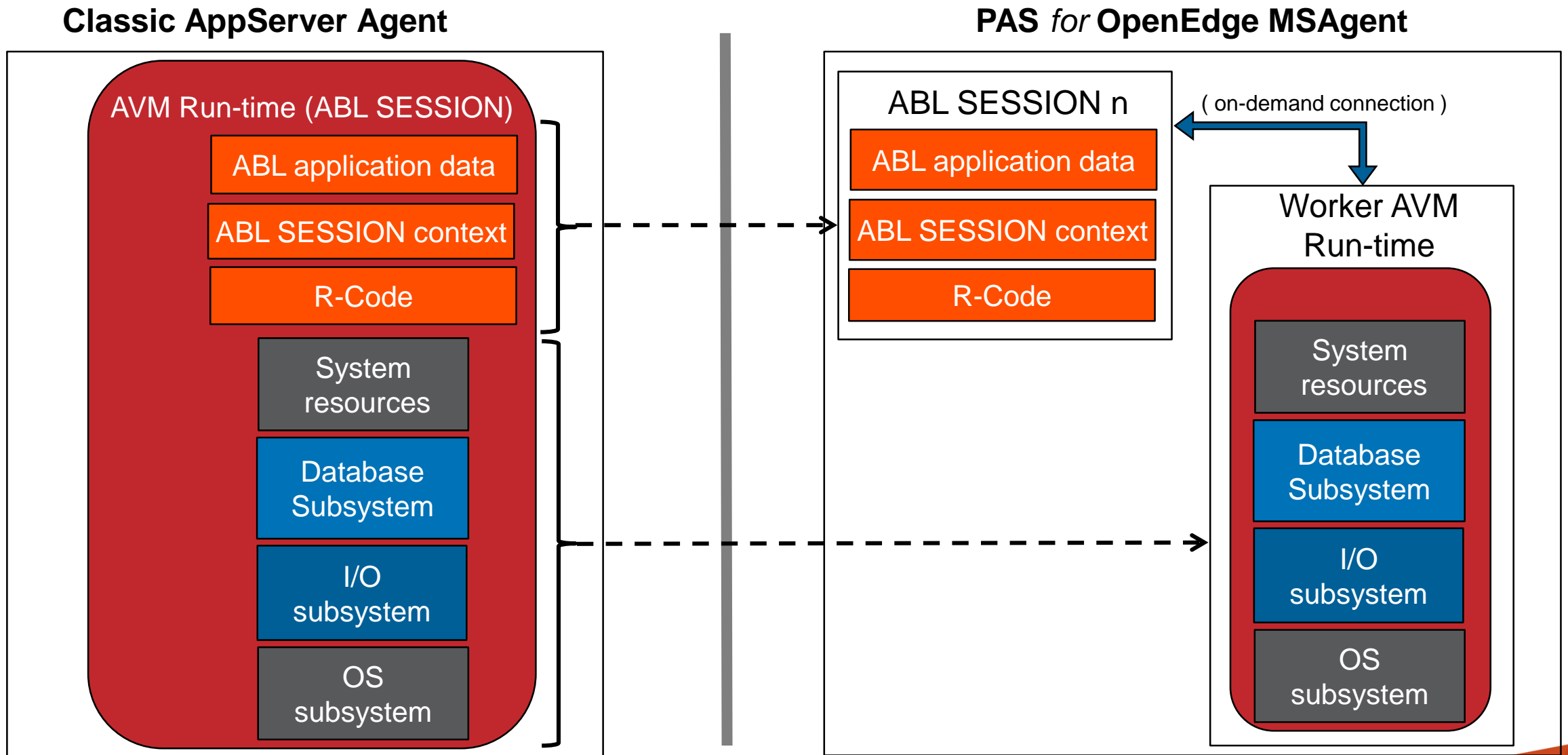
Classic AppServer Instance

- Toutes les instances sont centralisées dans le fichier `ubroker.properties`
- Propriétés également dans `ubroker.properties`
- Operating mode défini à la création
- Environnement également dans `ubroker.properties`

PAS for OE Instance

- Chaque instance dans le fichier `conf/openedge.properties`
- Propriétés dans plusieurs fichiers du repertoire `conf/`
- Pas d'operating mode
- Définition des variables d'environnement :
`bin/<app-name>_setenv.{sh|bat}`

ABL Sessions and AVM Run-times as Independent Resources



PAS for OpenEdge ABL Application Architecture Models

- **Le client contrôle le mode de session (via l'option `--sessionModel`)**
- **Mode stateless** (i.e. Session-Free)
 - Une requête peut être exécutée depuis n'importe quelle session ABL
 - Un client peut exécuter plusieurs requêtes en même temps
 - Contexte géré par le client et le serveur
- **Mode Stateful** (i.e. Session-Managed)
 - Utilisé pour les modes State-Reset, State-Aware et Stateless
 - Chaque requête vers la même session ABL
 - Le contexte est stocké dans la session ABL entre chaque requête
 - Support des transactions automatiques
 - Un client peut exécuter un pipeline de requêtes asynchrones

Connexions DB self-service

AppServer classique

- Une connexion DB par agent (pas de partage)
- Connexion au démarrage de la session ABL ou au CONNECT
- Déconnexion à l'arrêt de la session ABL ou au DISCONNECT
- PROMON affiche une connexion par session ABL
- PROMON déconnecte une session ABL à la fois

PAS for OpenEdge

- Une connexion DB par MSAgent : partagé par toutes les sessions
- Connexion au démarrage de la première session (ou au CONNECT)
- Déconnexion lors de l'arrêt de la dernière session (ou DISCONNECT)
- Promon affiche une connexion par session + une session Agent (admin)
- PROMON déconnecte toutes les sessions ABL lorsque l'agent (admin) est déconnecté

Configuration des connexions (sur le client)

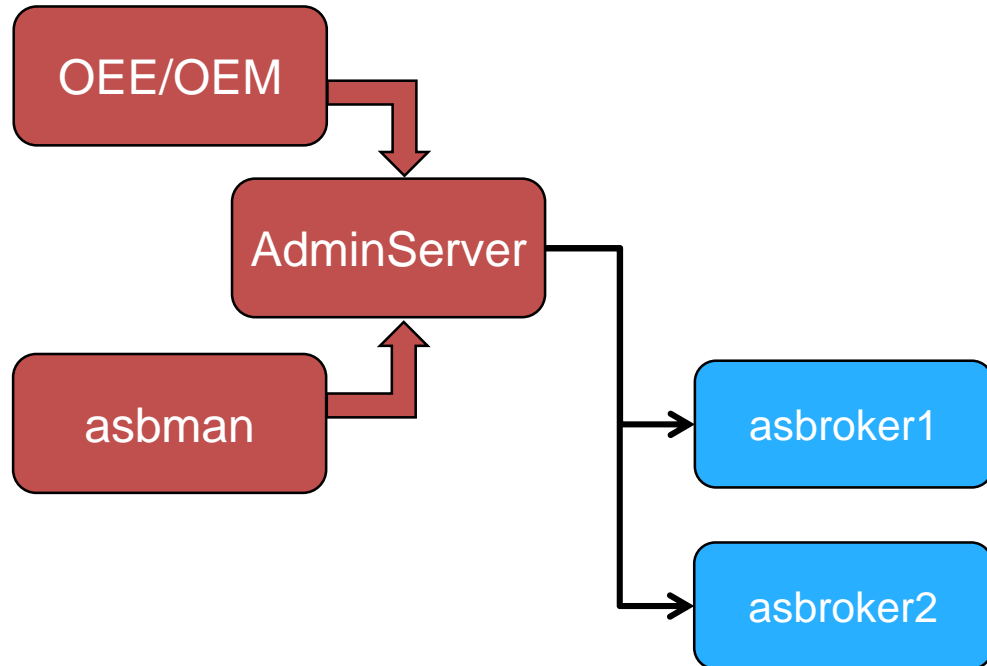
- **Clients REST/Mobile:** Pas de difference (changement d'URL)
- **Clients SOAP :** Pas de différence (redéploiement WSDL sur nouveau port)
- **Clients OpenEdge**
 - Un seul mode de connexion : HTTP
 - Référence: *Connecting to AppServers Using a URL*
 - Le champ ***aia-path*** devient le nom de **l'application Web** PAS for OpenEdge
 - Service par défaut: ROOT web application *“/”*
 - Service *sales* : *sales web application* *“/sales”*
 - Toutes les connexions se font avec le format URL
 - *-url http[s]://host[:port]/oeabl-path*

Configuration des événements AppServer

- **agentStartupProc & agentShutdownProc**
 - Exécuté une fois au démarrage / arrêt du processus MSAgent
- **Renommage des procédures classiques dans le PAS for OpenEdge:**
svrXxxxx → sessionXxxxx
- **sessionStartupProc & sessionShutdownProc**
 - Exécuté auparavant lors du démarrage du _proapsv
 - Exécuté maintenant lors du démarrage d'une session dans le MSAgent
- **sessionConnectProc & sessionDisconnectProc**
 - Pas de changement (pour les connexions Session-Managed)
- **sessionActivateProc & sessionDeactivateProc**
 - Exécuté sur les Appservers classiques à chaque requête Stateless et State-free
 - Modèle Stateless traditionnel

Classic AppServer versus PAS for OpenEdge Administration

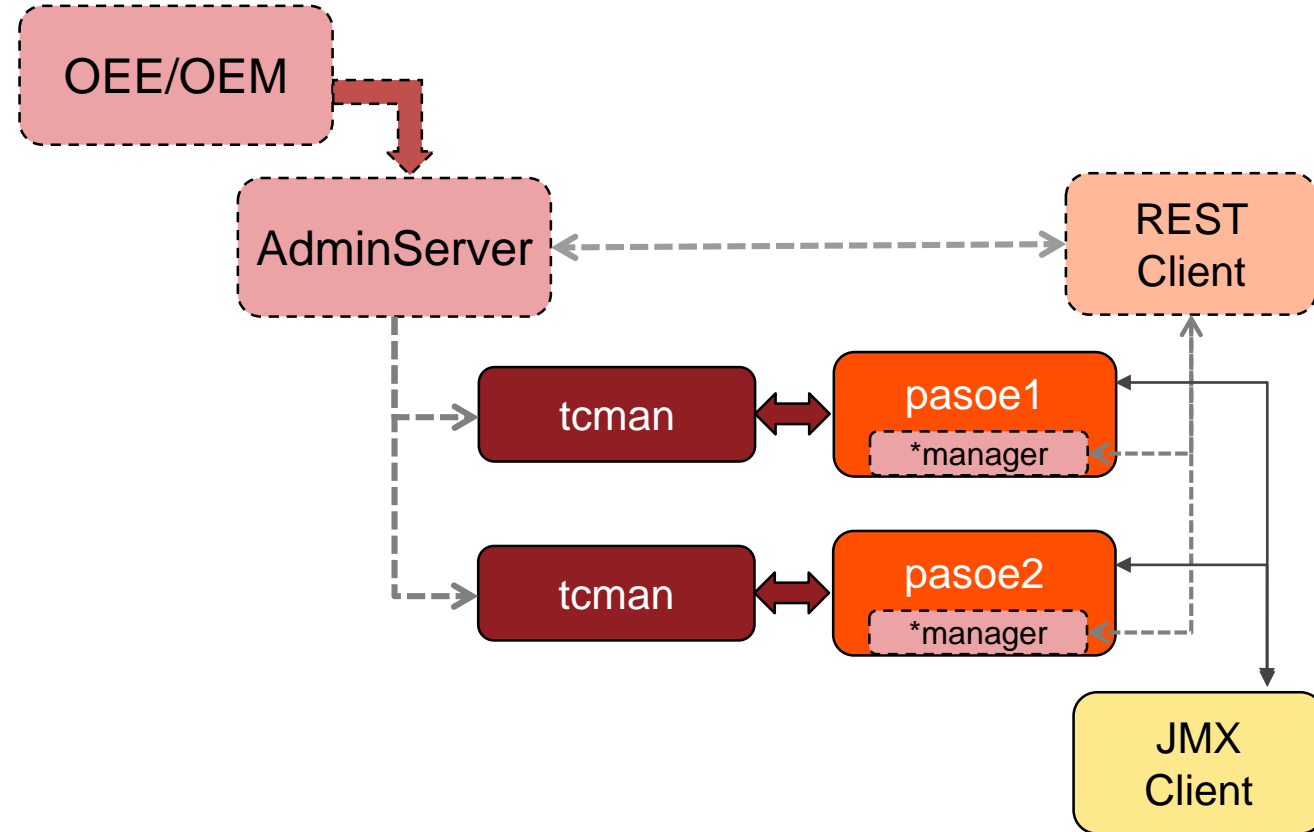
Classic AppServer



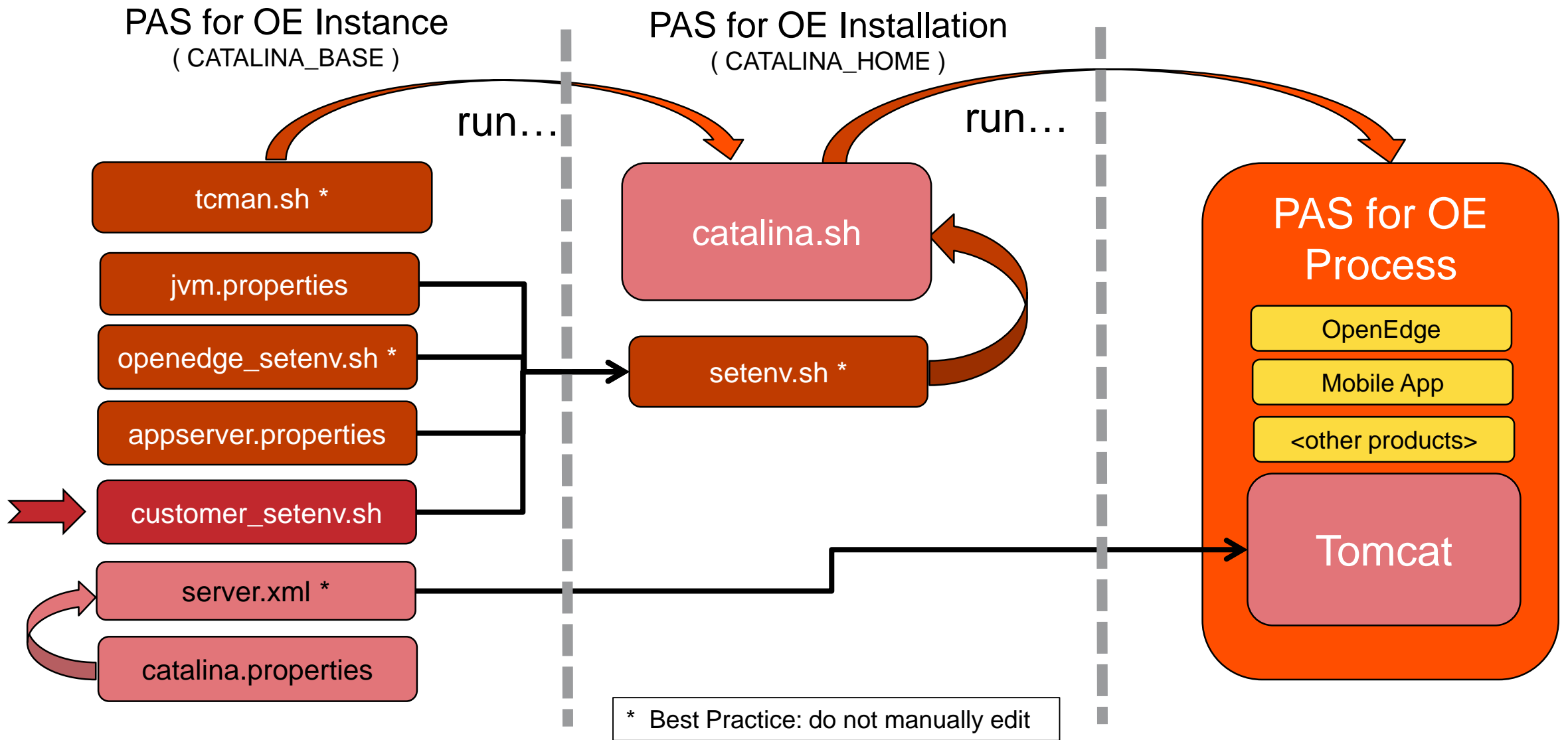
Legend:

- always exists
- - - - optional install & use

PAS for OpenEdge



PAS for OpenEdge Instance Startup Process (or What comes from Where...)



Où trouver...

PAS for OE's DLC & WRKDIR environment variable definition	\$CATALINA_BASE/bin/openedge_setenv.sh
Application's environment variable definitions	\$CATALINA_BASE/bin/<app-name>_setenv.sh
PAS for OE installation path environment variable definition PAS for OE instance's path environment variable definition PAS for OE instance's temporary file directory path definition	\$CATALINA_BASE/bin/{tcman startup shutdown configtest version}.sh
PAS for OE configuration properties	\$CATALINA_BASE/conf/appserver.properties \$CATALINA_BASE/conf/catalina.properties \$CATALINA_BASE/conf/jvm.properties
Optional deployment files (web applications & other files)	\$CATALINA_HOME/extras
PAS for OE & web application log files	\$CATALINA_BASE/logs
PAS logging configuration	\$CATALINA_BASE/conf/logging.xml (ref Tomcat logging)
PAS for OE session manager & MSAgent logging configuration	\$CATALINA_BASE/conf/openedge.properties
oeabl web application logging configuration	\$CATALINA_BASE/ROOT/WEB-INF/logging.xml
oemanager remote admin web application login configuration	\$CATALINA_BASE/oemanager/WEB-INF/logging.xml
Test user accounts and roles	\$CATALINA_BASE/conf/tomcat-users.xml
PAS for OE instance registration list	\$CATALINA_HOME/conf/instances.unix
JAVA_HOME & JSE_HOME environment variable definition	\$CATALINA_BASE/bin/javacfg.sh

Partage des ressources (dans la release 11.5)

Partagé

- Connexion aux bases de données en shared memory
- Bibliothèques de procédures
- R-code
- Promsg files
- OS threads

Non partagé

- Temp-tables / ProDatasets
- Sockets (y compris connexions SOAP et AppServer)
- Connexion aux bases de données par TCP/IP
- Fichiers LBI & DBI
- Mémoire privée session ABL

Déploiement des applications

■ Service SOAP

- Génération d'un fichier .wsm par le Proxygen
- Utilisation de la ligne de commandes **deploySOAP** pour déployer le .wsm

■ Service REST

- Export d'un fichier .paar par PDSOE (même manière que 11.3)
- Utilisation de la ligne de commandes **deployREST** pour déployer le .paar

■ R-Code ou bibliothèques (PL)

- Identique à un appservers traditionnel (tout est dans le PROPATH)

■ En tant qu'application Web Java (.war)

- Création d'un fichier .war depuis un service OEABL avec le code source et les services inclus

■ Déploiement instance PASOE en tant que fichier ZIP avec les services préinstallés (manuel)

Performance and Scalability

PROGRESS
EXCHANGE 2014

Améliorations de performance et de montée en charge

- Threads vs Processes
 - Un processus multi-threadé supporte plusieurs sessions ABL simultanées
 - Un process single-thread ne supporte qu'une seule session ABL
- Gestionnaire de sessions intégré à Tomcat
 - Plus de processus Java Ubroker séparé
 - Une couche réseau en moins
- Nombre initial de sessions créé au démarrage
 - numInitialSessions -> numInitialAgents
 - Gestion plus simple d'un grand nombre de connexions

Early Access Available

Interested?

Contact

Early-Software-Admin@progress.com



PROGRESS