Introducing Pacific Application Server for OpenEdge (PAS for OpenEdge)

Roy Ellis
ellis@progress.com

Agenda

- Introduction to the Pacific Application Server Platform
- Introduction to the Pacific Application Server for OpenEdge
- Pacific Application Server for OpenEdge Architecture
- Performance and Scalability

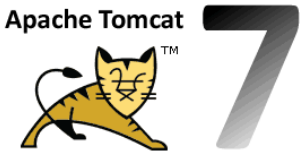
Introduction to PAS Platform

PROGRESS SPARK

Pacific Application Server Platform

- A single delivery platform for all Progress Web-based products
  - OpenEdge
  - Rollbase
  - Corticon

- Not only the application but also the web server to support it
  - Created from Apache Tomcat 7.0.55 distribution

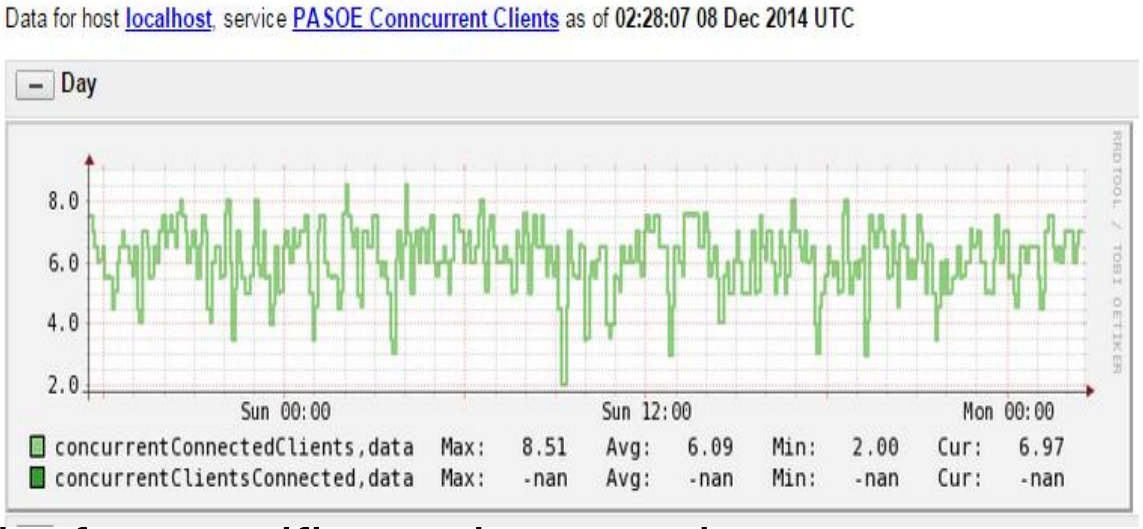


- Architected for secure operation
  - Spring Security Framework included
  - Realms and roles defined to implement access control

PROGRESS SPARK

Pacific Application Server Platform

- Easy management
  - Common set of enhanced management tools
  - No Tomcat knowledge necessary
- Open monitoring
  - JMX console/scripting
  - Any third party Tomcat monitoring tools
- Extensible
  - Each product can extend core functionality for specific product needs



PROGRESS SPARK

Pacific Application Server Platform

- Uses the Tomcat idea of “instances”
- The Progress install will:
  - Create a READ-ONLY tomcat home directory (a.k.a. CATALINA_HOME)
  - And a default instance (a.k.a. CATALINA_BASE)
- Applications run in the instances

Introduction to PAS for OpenEdge

Pacific Application Server for OpenEdge (PASOE)

- Installable application server that is a single package the functionality of:
  - Tomcat
  - Classic AppServer (the one that currently exists with OpenEdge)
  - AppServer Adapters
- Runs in Progress's unified Pacific Application Server (PAS) platform
  - Same web server installed with Rollbase-private installation and with Corticon
  - Common server administration and configuration functionality
- Supported on only 64 bit platforms (
  - Linux, Solaris, HPUX-IA, AIX, Windows 2008/2012)
- Two products
  - Development server
  - Production server

APJ
PROGRESS SPARK

Architectural Drivers

- Secure production web server
  - Installation, administration
- Simpler
  - Administration, scalability, application migration, deployment
  - AppServer connection and operating STATEs
- Customer Extensible
  - Open REST APIs for customer developed metrics, monitoring, and administration
  - Installation tailoring
- Better analysis tools
  - Built-in metrics gathering, current state queries
- Faster and optimizes resources
  - Runs same ABL application and client load with less memory and CPU consumption

PAS for OpenEdge Production versus Development Products

PAS for OE Development	PAS for OE Production
Non-secure configuration	Secure configuration
Test server instance in $WRKDIR	No test server instances
Remote administration included Tomcat remote admin enabled OpenEdge remote admin enabled	Remote administration optional Tomcat remote admin optional OpenEdge remote admin optional
Built-in oeabl web application (ROOT) All transport deployed and enabled	Built-in oeabl web application (ROOT) All transports deployed but disabled
Restricted 5 concurrent requests 1 agent	Unrestricted concurrent requests number of agents

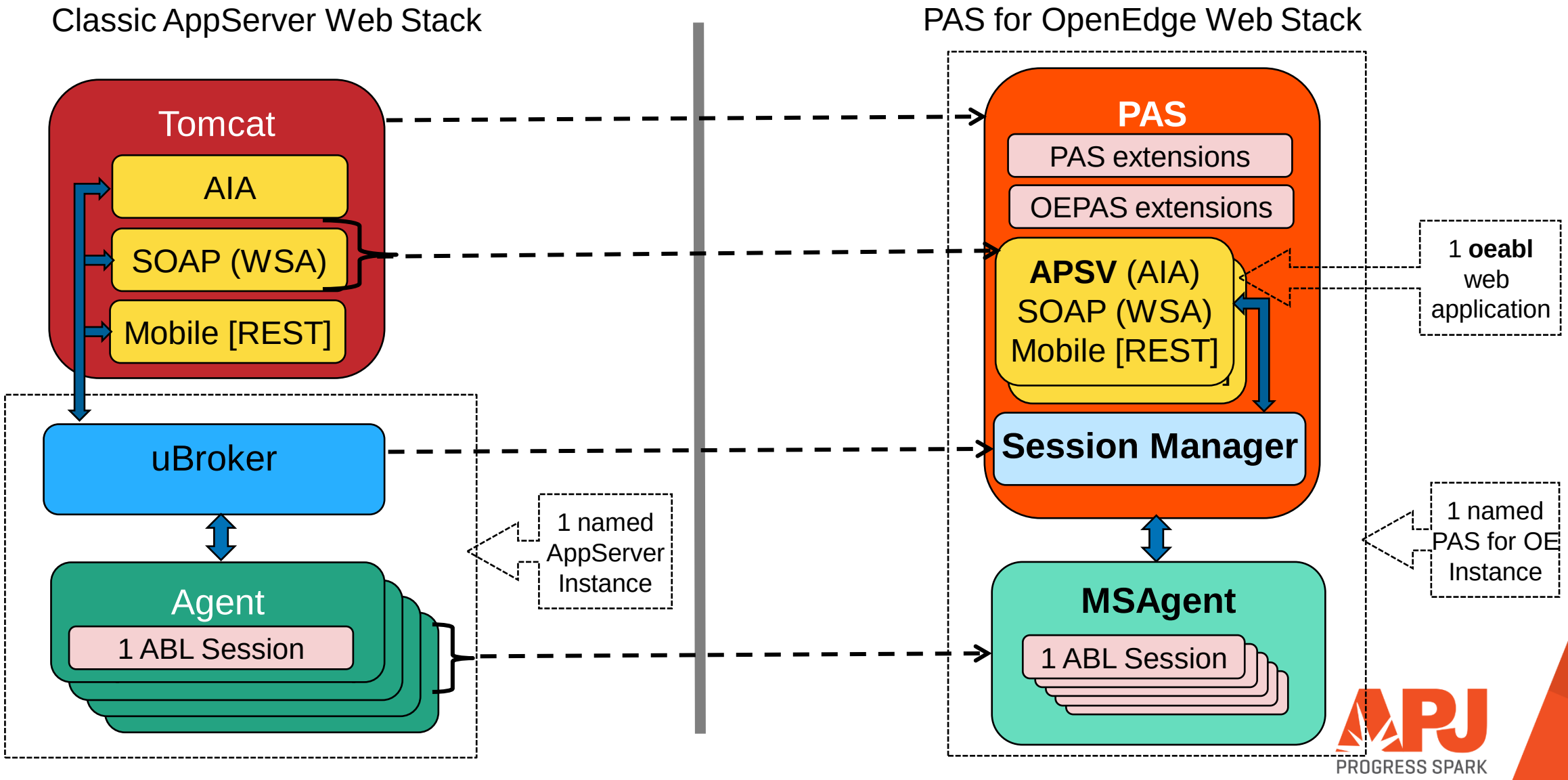
PROGRESS SPARK

PAS for OpenEdge Architecture

PASOE Architectural Concepts and Terms

- **PAS** is a Java web application server with Progress extensions
  - Optional web application for remote administration using text, html, or JMX proxy APIs
- **ABL** services for Pacific Application Server (ABLPAS)
  - An collection of Java web applications and resources installed and into a **PAS**
  - Java web applications use multi-session agent OS process (**MSAgent**)
  - *Optional* web application for remote administration using REST APIs
  - ***ABLPAS*** web services can be installed into the **PAS** of other PSC product installations
- **PASOE** is an OpenEdge installed **PAS** with a pre-installed **ABLPAS** services
- You design, package, deploy, configure, debug, and control access to your ABL application in the context of a web application running in a web server

PAS for OpenEdge Architectural Concepts and Terms

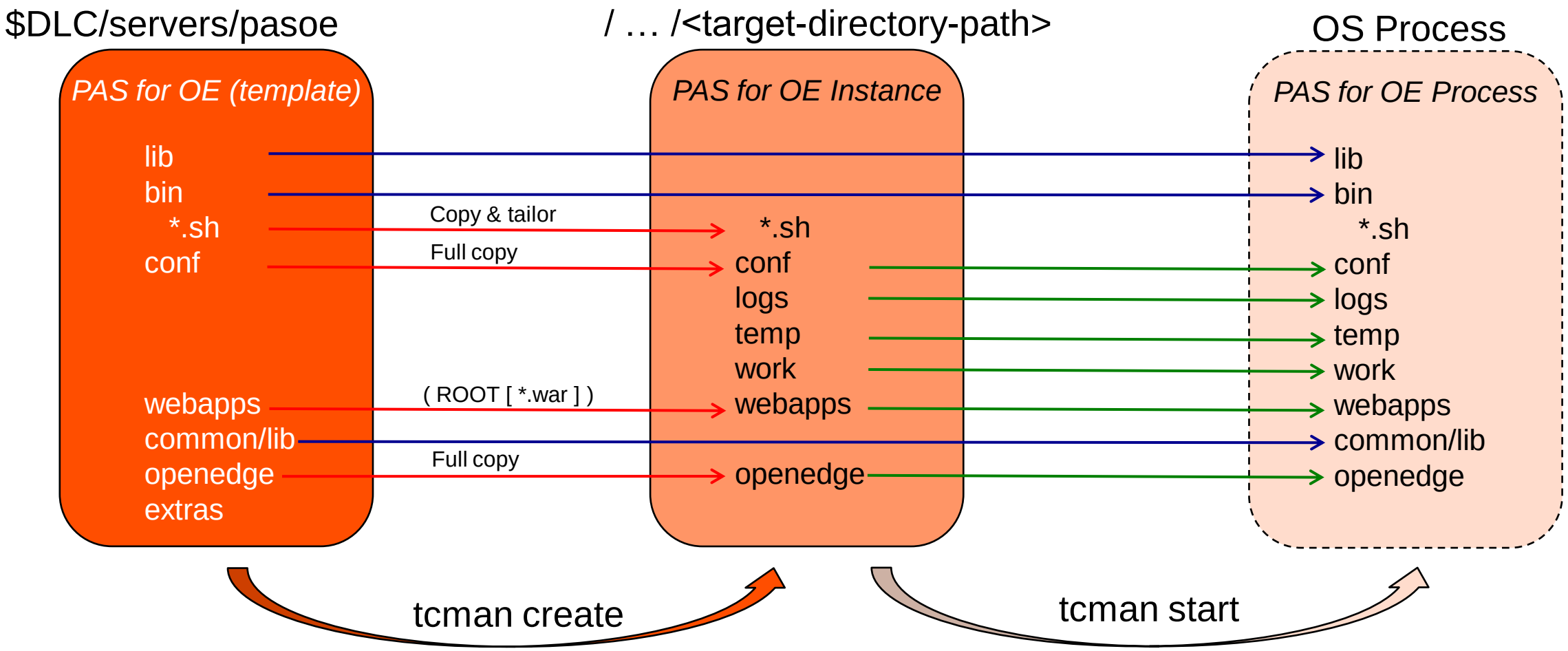


PROGRESS SPARK

Classic AppServer Instances versus PAS for OpenEdge Instances

Classic AppServer	PAS for OpenEdge
Each AppServer instance has a name	Each PAS for OE instance has a name
AppServers instance resides inside OE install directory structure	PAS for OE instance resides outside the OE install directory structure
AppServer instances lost at OE uninstall	PAS for OE instance lives beyond OE uninstall
Broker and adapter instances are non transferable to other OE installs	PAS for OE instances are transferrable between OE installs
Cannot package and redeploy an instance	An instance can be packaged and redeployed

Understanding PAS for OpenEdge Instance Run-time

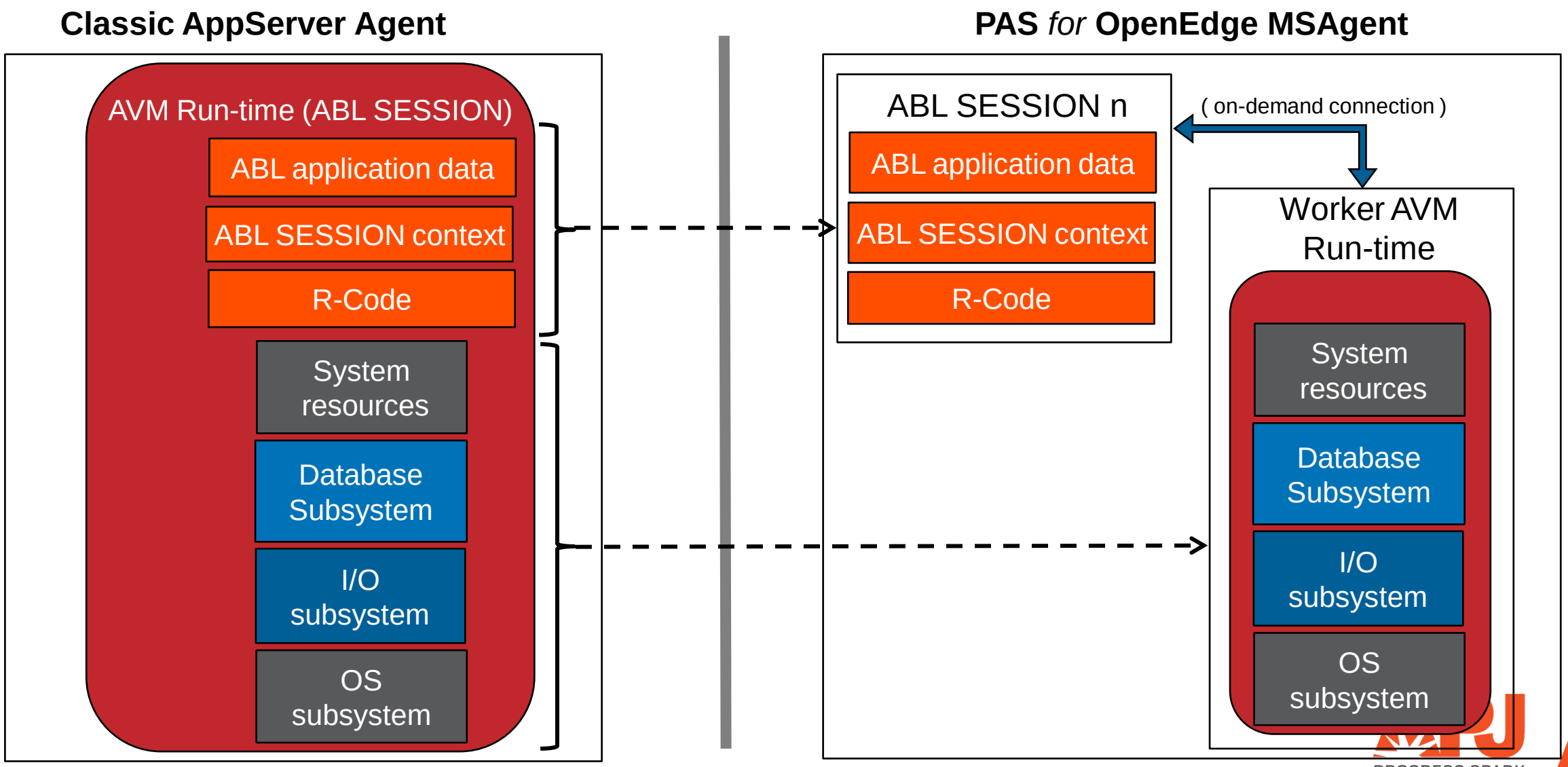


PROGRESS SPARK

Classic AppServer Configuration versus PAS for OE Configuration

Classic AppServer instance	PAS for OpenEdge instance
All instances located in common ubroker.properties file	Each instance located in its conf/openedge.properties file
All properties in ubroker.properties	Properties in multiple property files catalina, appserver, & openedge
No named service(s)	Named service(s)
Fixed STATE Model	No STATE – STATE set by client
Environment variables defined in common ubroker.properties	Environment variables defined in instance's bin/<app-name>_setenv.{sh|bat}

ABL Sessions and AVM Run-times as Independent Resources



PAS for OpenEdge ABL Application Architecture Models

- Client controls ABL session model via traditional AppServer *–sessionModel* option
- Stateless session model (i.e. Session-Free)
  - Client's request can be executed in any ABL session, in any Agent, in any PAS for OE server
  - A client can execute concurrent ABL requests
  - User context managed entirely by the server application and client code
- Stateful session model (i.e. Session-Managed)
  - Used for classic AppServer State-Reset, State-Aware, and Stateless
  - Each client's request routed to same PASOE server, MSAgent, and ABL session (until DISCONNECT)
  - A user's context is stored within an Agent's ABL session between requests
  - Supports Automatic Transactions [that span multiple requests]
  - A client can execute pipelined async requests [to single ABL session]

APJ
PROGRESS SPARK

RDBMS Self-service Connections

Classic AppServer	PAS for OpenEdge
One connection per DB per Agent process: no ABL SESSION sharing	One connection per DB per Agent process: shared by all ABL SESSIONs
Connected @ ABL SESSION startup or CONNECT	Connected on 1st SESSION startup or CONNECT statement
Disconnected @ ABL SESSION shutdown or DISCONNECT	Disconnected when **last** SESSION shuts down or **last** DISCONNECT
PROMON show one connection per ABL SESSION	PROMON shows one connection per SESSION + 1 Agent (admin) SESSION
PROMON disconnect one ABL SESSION at a time	PROMON disconnects ALL SESSIONs when Agent (admin) SESSION disconnected

APJ
PROGRESS SPARK

PROMON - RDBMS Self-service Connections

User Control: by user number for all tenants

Usr:Ten	Name	Domain	Type	Wait	Table:Part	Dbkey	Trans	PID	Sem	Srv	Login Time
0	root	0	BROK	--	0	0	0	8068	0	0	04/01/15 19:14
5	root	-4	SELF/PASA	--	0	0	0	9596	2	0	04/01/15 19:19
6	root	0	SELF/PASN	--	6	1412160	0	9596	3	0	04/01/15 19:19
7	root	0	SELF/PASN	--	6	924864	0	9596	3	0	04/01/15 19:19
8	root	-4	SELF/PASN	--	0	0	0	9596	4	0	04/01/15 19:19
9	root	-4	SELF/PASN	--	6	030912	0	9596	4	0	04/01/15 19:19
10	root	-4	SELF/PASN	--	6	1427776	0	9596	5	0	04/01/15 19:19
11	root	0	MON	--	0	0	0	1359	5	0	04/02/15 15:10

PROMON - RDBMS Self-service Connections

User Control: by user number for all tenants

Usr:Ten	Name	Domain	Type	Wait	Table:Part	Dbkey	Trans	PID	Sem	Srv	Login Time
0	root	0	BROK	--	0	0	0	8068	0	0	04/01/15 19:14
5	root	-4	SELF/PASA	--	0	0	0	9596	2	0	04/01/15 19:19
6	root	0	SELF/PASN	--	6	1412160	0	9596	3	0	04/01/15 19:19
7	root	0	SELF/PASN	--	6	924864	0	9596	3	0	04/01/15 19:19
8	root	-4	SELF/PASN	--	0	0	0	9596	4	0	04/01/15 19:19
9	root	-4	SELF/PASN	--	6	030912	0	9596	4	0	04/01/15 19:19
10	root	-4	SELF/PASN	--	6	1427776	0	9596	5	0	04/01/15 19:19
11	root	0	MON	--	0	0	0	1359	5	0	04/02/15 15:10

PROMON - RDBMS Self-service Connections

User Control: by user number for all tenants

Usr:Ten	Name	Domain	Type	Wait	Table:Part	Dbkey	Trans	PID	Sem	Srv	Login Time
0	root	0	BROK	--	0	0	0	8068	0	0	04/01/15 19:14
5	root	-4	SELF/PASA	--	0	0	0	9596	2	0	04/01/15 19:19
6	root	0	SELF/PASN	--	6	1412160	0	9596	3	0	04/01/15 19:19
7	root	0	SELF/PASN	--	6	924864	0	9596	3	0	04/01/15 19:19
8	root	-4	SELF/PASN	--	0	0	0	9596	4	0	04/01/15 19:19
9	root	-4	SELF/PASN	--	6	030912	0	9596	4	0	04/01/15 19:19
10	root	-4	SELF/PASN	--	6	1427776	0	9596	5	0	04/01/15 19:19
11	root	0	MON	--	0	0	0	1359	5	0	04/02/15 15:10

APJ
PROGRESS SPARK

Configuring PAS for OpenEdge Client Connections

- **REST/Mobile clients:** No differences (change URL in http clients)
- **SOAP clients:** No differences (redeploy WSDL with new port URL)
- **OpenEdge clients**
  - PAS for OpenEdge only has one connection model : HTTP(S)
  - Reference: *Connecting to AppServers Using a URL*
  - The Classic AppServer's ***aia-path*** field becomes the PAS for OpenEdge's oeable **web application** name
    - Default service : ROOT web application *"/"*
    - *sales service : sales web application "/sales"*
  - All PAS for OpenEdge connections are specified using the URL connection format
    - *-url* ***http***[***s***]***://host***[*:****port***]***/oeabl-path***

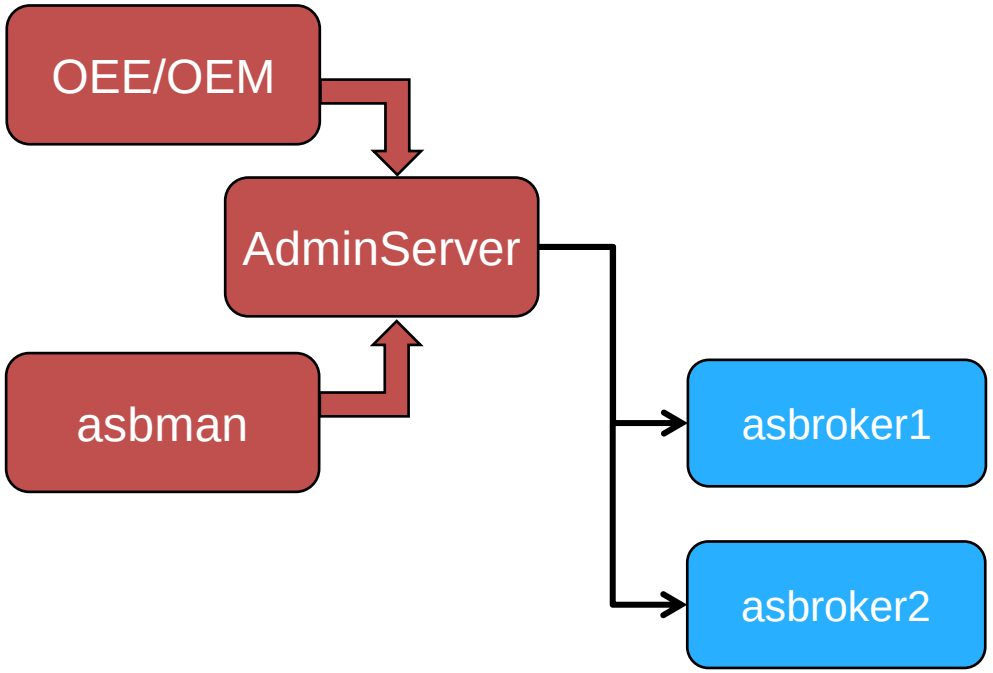
PROGRESS SPARK

Configuring and Using AppServer Event Procedures

- agentStartupProc & agentShutdownProc
  - Executes one time when starting / stopping an MSAgent OS process
- PAS for OpenEdge renamed classic AppServer event procedures:
  srvrXxxxx → sessionXxxxx
- sessionStartupProc & sessionShutdownProc
  - Executed in classic AppServer when the Agent's single ABL SESSION started/stopped
  - Executes in PAS for OE when the MSAgent starts/stops each ABL SESSION
- sessionConnectProc & sessionDisconnectProc
  - Same as classic AppServer for all Session-Managed client connections
- sessionActivateProc & sessionDeactivateProc
  - Executed in classic AppServer on every Stateless & State-Free client request
  - Follows traditional Stateless model

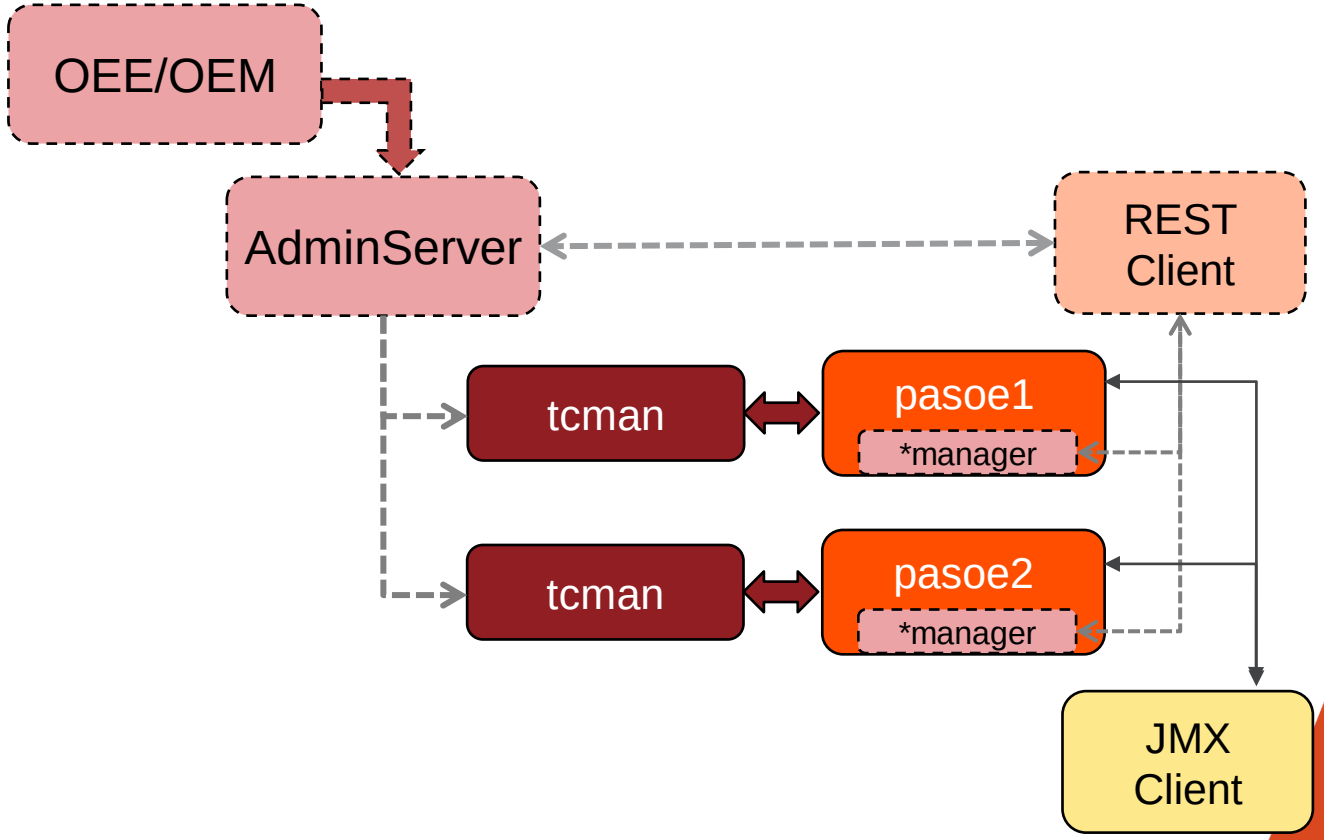
Classic AppServer versus PAS for OpenEdge Administration

Classic AppServer



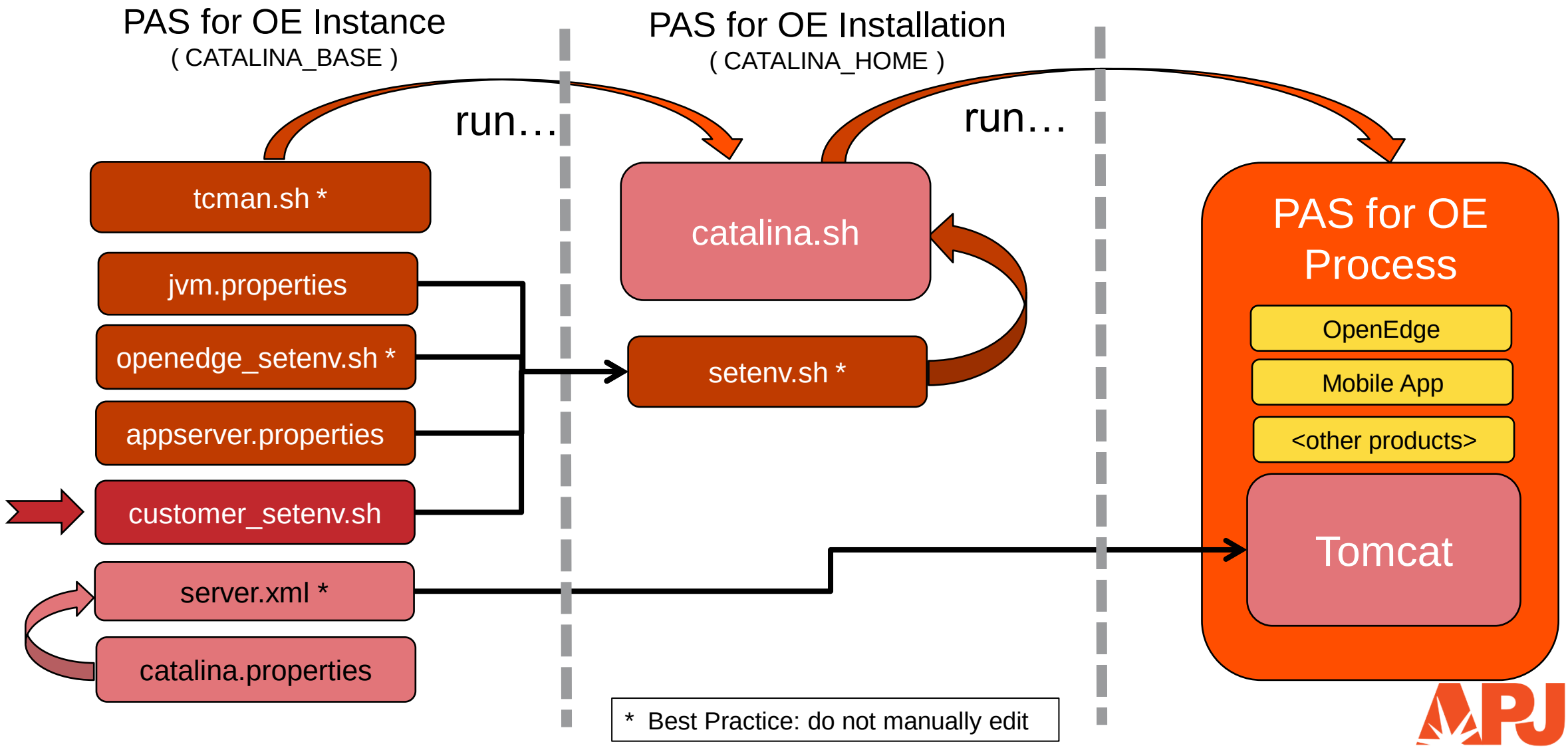
Legend:
- always exists
- optional install & use

PAS for OpenEdge



PROGRESS SPARK

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



PROGRESS SPARK

Where Do I Find My…

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	$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 loggin 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

Resource Sharing in Initial Release

Shared	Not Shared
Self-service OpenEdge database connections	Temp-tables / ProDatasets
Procedure libraries	Sockets (including SOAP and AppServer connections)
R-code	Network OpenEdge database connections
Promsg files	LBI & DBI files
OS threads	ABL session memory

Application Deployment Options

- SOAP service
  - SOAP service descriptor .wsm generated by Proxygen
  - Deploy .wsm file via PASOE command line tool *deploySOAP*
- REST service
  - REST/Mobile service descriptor .paar file exported from PDSOE (same as 11.3)
  - Deploy .paar file from PASOE command line tool *deployREST*
- ABL r-code / libraries
  - Same as traditional AppServer (it's all about PROPATH)
- As a OEABL Java web application (.war)
  - Create .war file from OEABL service with embedded ABL code and SOAP/REST services
- Deploy PASOE instance as a ZIP file with pre-installed OEABL services (manual)

Performance and Scalability

Pacific Application Server for OpenEdge - Performance

	Classic AppServer	PAS for OE	Difference
Scalability			
Client connections	221	1312	493%
Server Resources			
CPU	10 CPUs	5.2 CPUs	192%
Memory	2.1 GB	670 MB	313%
Transactions	203 tps	1698 tps	736%
Client performance			
OpenEdge	472ms	340ms	138%

Performance and Scalability Improvements

- Threads vs Processes
  - Single multi-threaded process supports multiple, concurrent, ABL sessions
  - Single-threaded process supports single ABL session
- Session Manager integrated with Tomcat
  - No separate Java Ubroker process
  - Removes one network copy in stack
- Initial sessions created at startup
  - numInitialSessions -> numInitialAgents
  - Helps service connection storms

Q&A


APJ
PROGRESS SPARK