

Best Practices on Migration to Progress Application Server (PAS) for OpenEdge

What to expect when implementing PAS for OpenEdge

Chad Thomson

Senior Principal Consultant





Progress®

Presenter BIO

- A seasoned software professional with over 25 years of experience delivering creative, innovative solutions to the Progress Community
- Chad is a sought-after international speaker and has been a presenter at numerous Progress and PUG Conferences
- Chad has been providing technical training for over 20 years
- Recognized expert in OpenEdge, Security, CI/CD, and all things technical

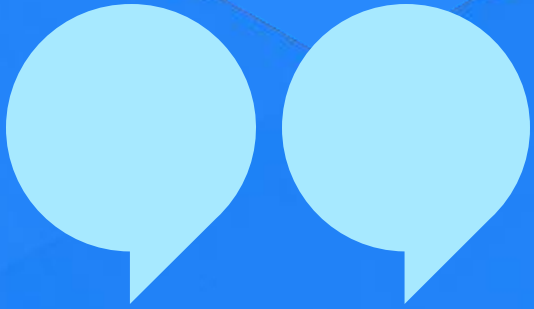


Webinar Agenda

- **Architecture Fundamentals**
Compare and contrast Classic Broker definitions to PAS for OpenEdge
- **Planning for Success**
Identify the types of activities to make the implementation a success
- **Considerations and Expectations**
Scenarios to consider and to expect
- **Q&A**

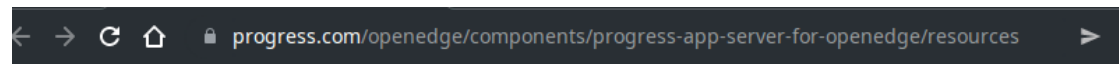


Architecture Fundamentals



"Experience is something you get just after you need it."

— Steven Wright



Migration Guide Resources

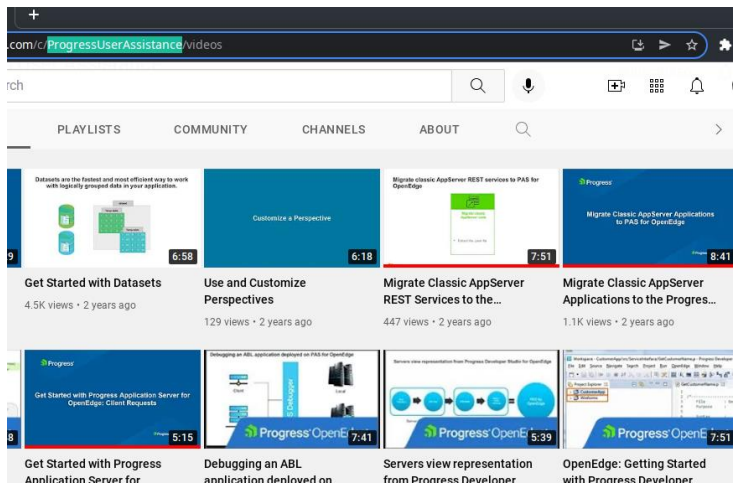
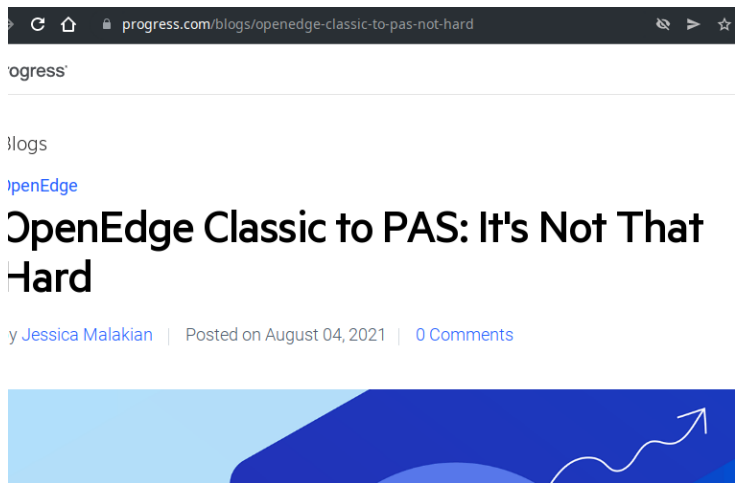
PAS for OpenEdge utilizes system resources very efficiently, easing installation, configuration and management. There are a number of ways to migrate PAS for OpenEdge, this section provides an overview of some of the ways our customers and partners have migrated.



PAS for OpenEdge Technical Migration Guide

This migration guide highlights PAS for OpenEdge practices for continuous availability focusing on strategies to migrate ABL applications from Classic AppServer and WebSpeed.

[DOWNLOAD THE MIGRATION GUIDE →](#)



Progress®

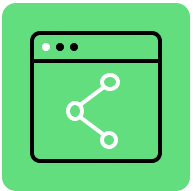
Same questions – better answers

- What is PAS for OpenEdge?
- How does it compare to classic?
- What do the differences mean to me?
- What can I expect from PAS for OpenEdge?
- How do I make sure I have a positive experience implementing PAS for OpenEdge?





Best Practice - Order of Operations



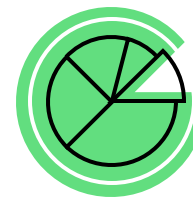
Structure

- Understand PAS for OpenEdge application architecture, deployment model, and how to apply it in your environment.



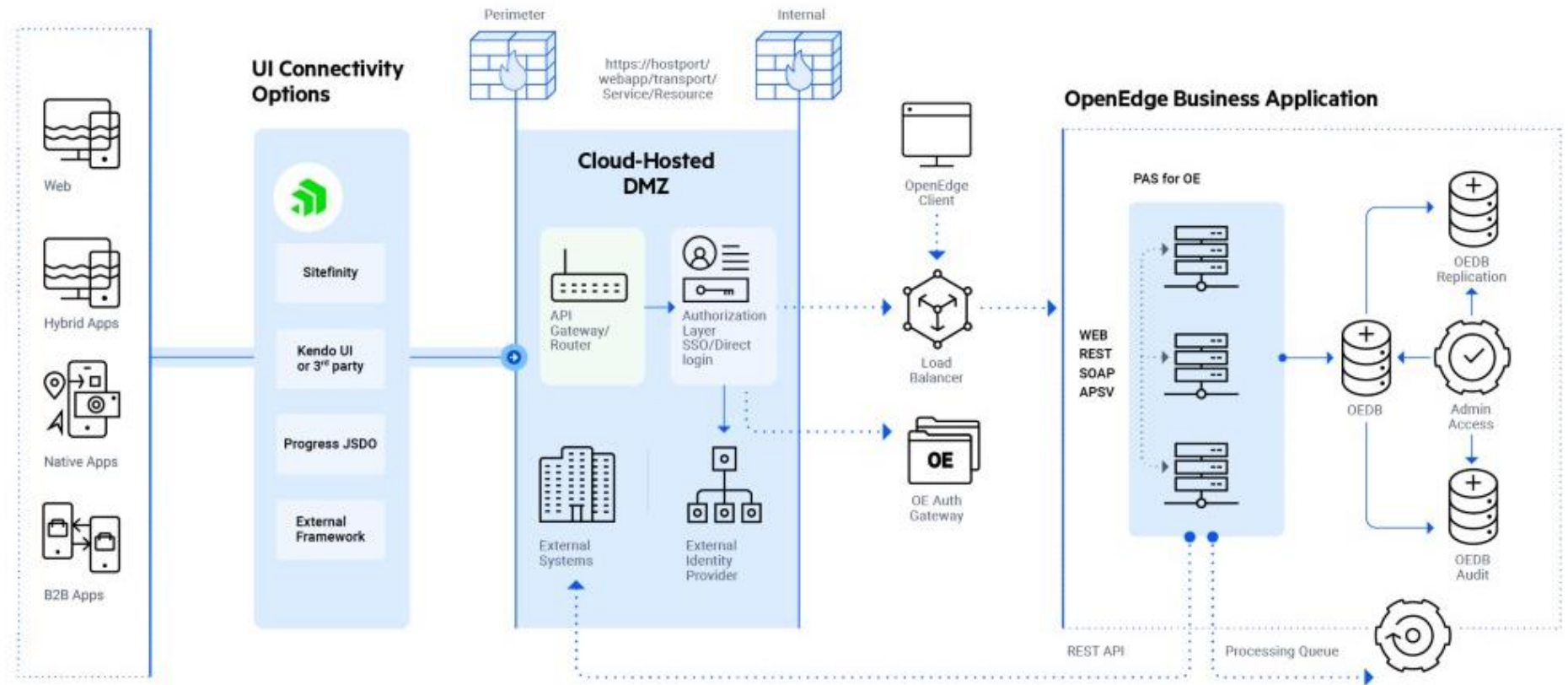
Stability

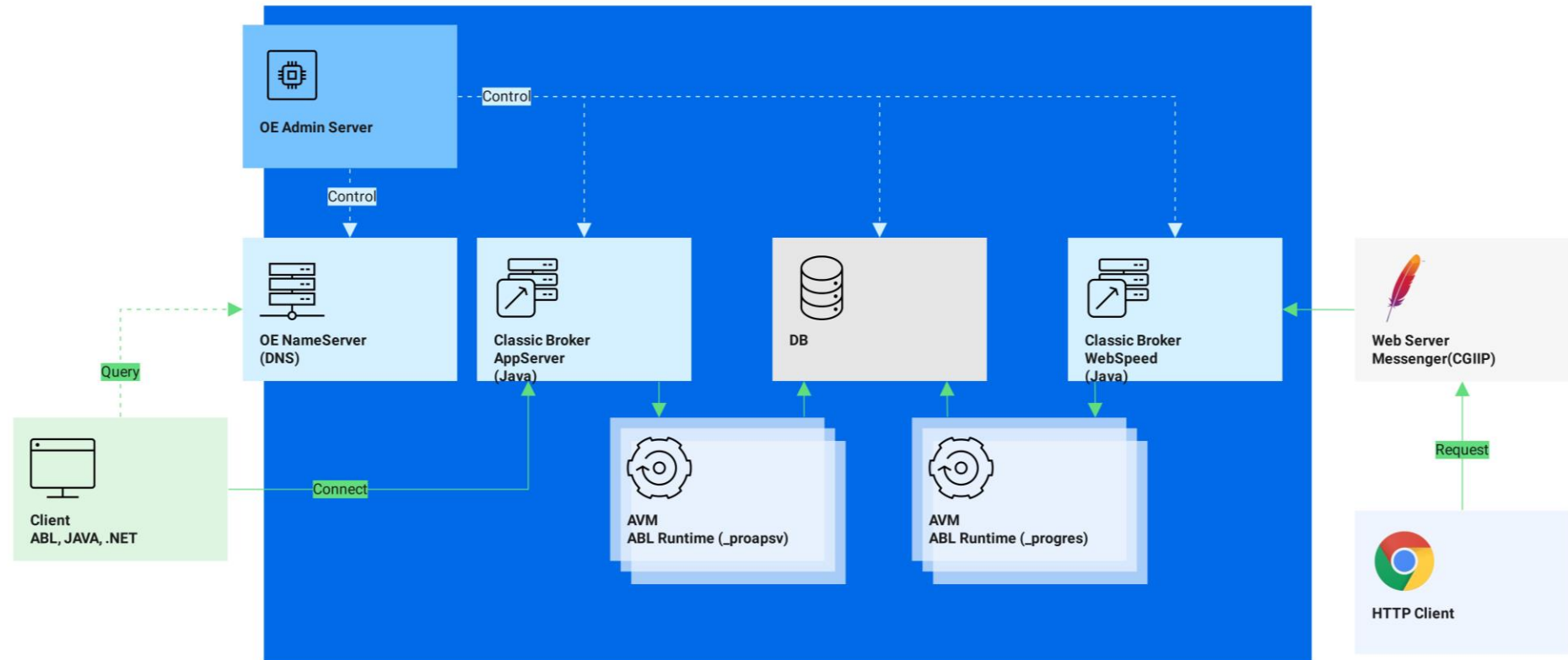
- Ability to accurately rebuild and reconfigure PAS for OpenEdge instances and related applications.



Scale

- Monitor for liveness
- Measure metrics
- Adapt to demand





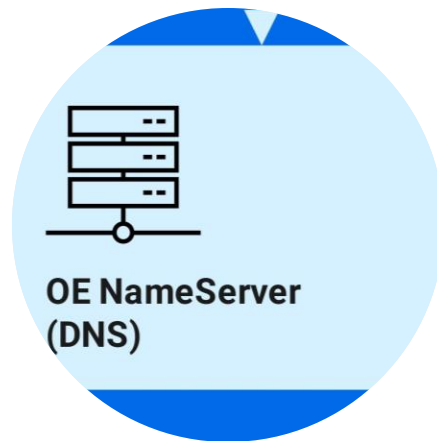
Classic Environment Components

Points of Interest - Classic Broker



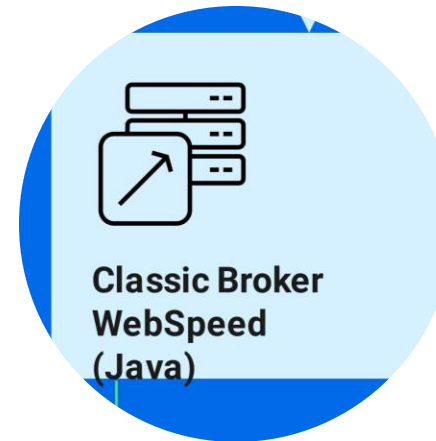
OE Admin Server

- Control and configuration of brokers



OE Name Server

- Name lookup service
- Returns IP and Port of registered (named) OE services



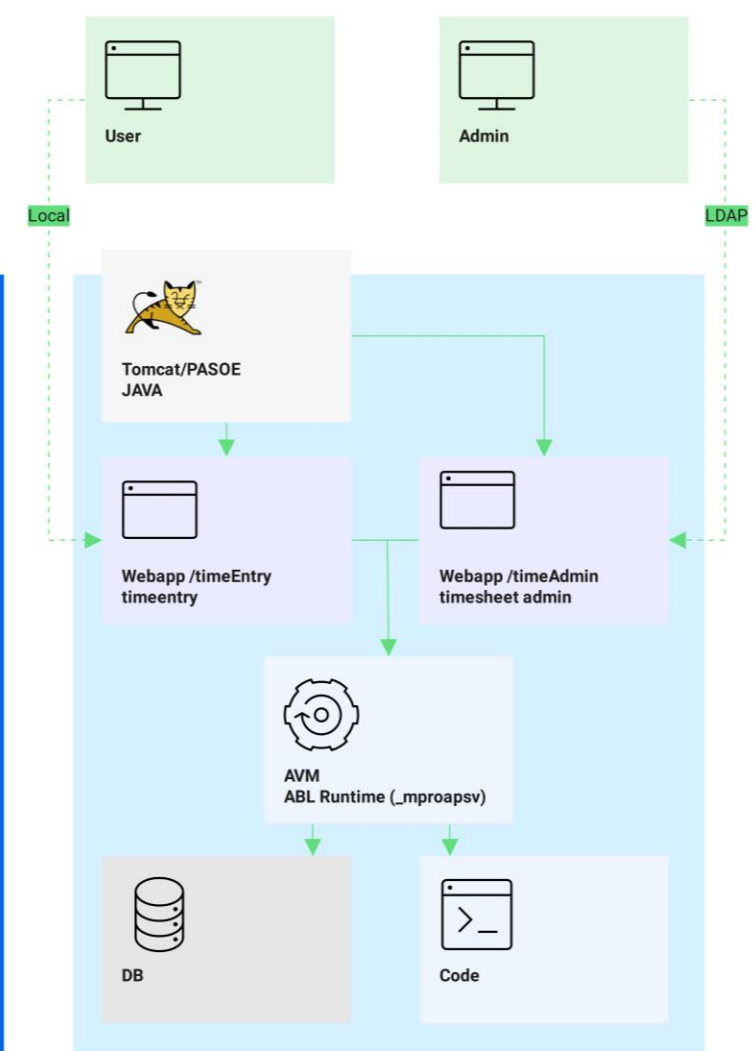
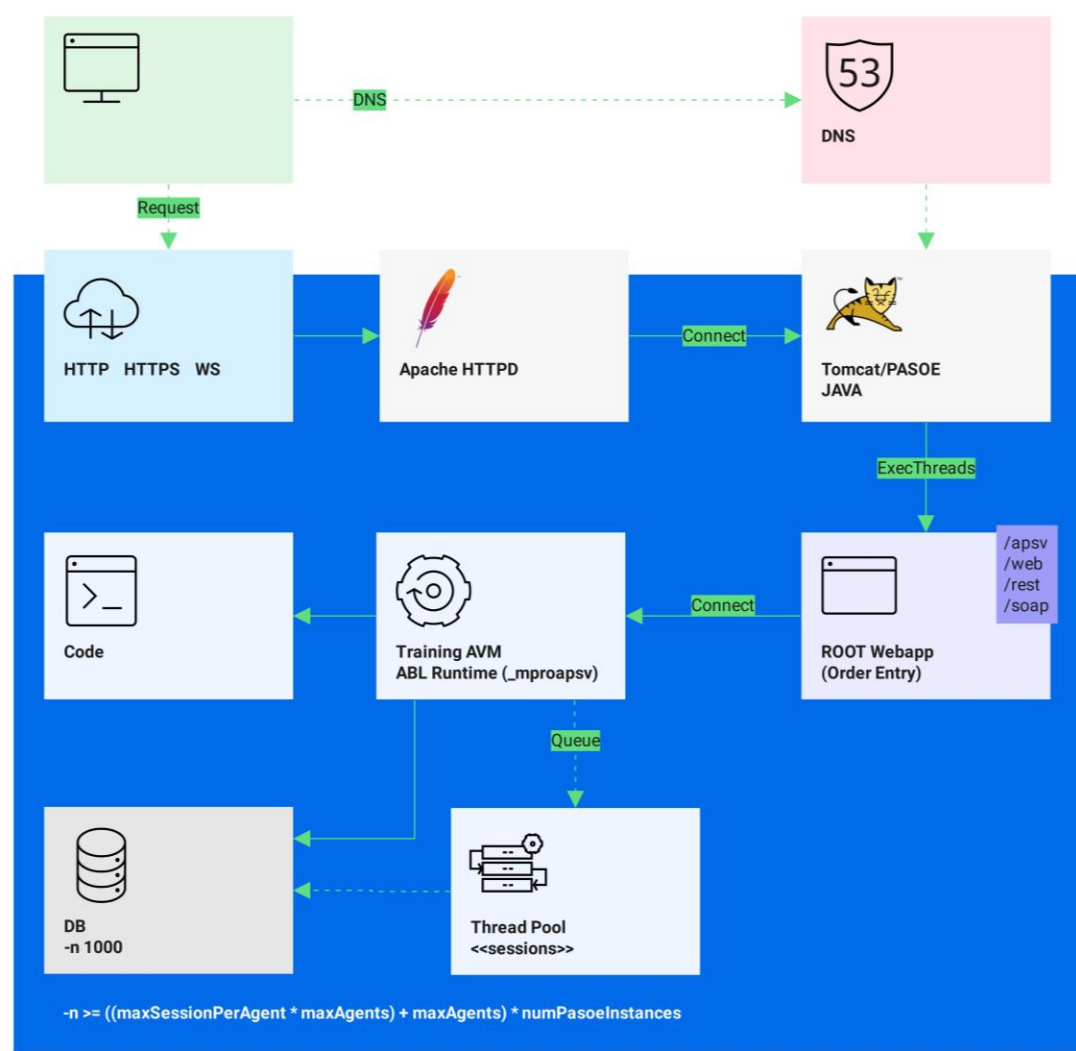
Broker

- Java process
- Maintains pool of Agents (runtime AVM)

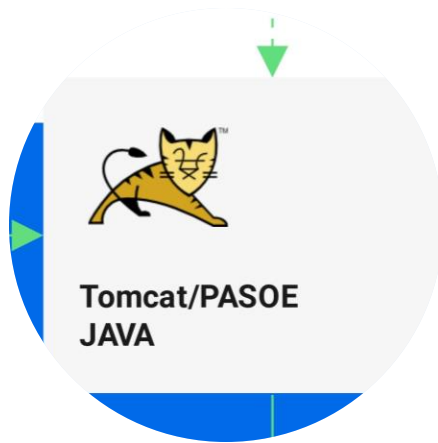


Agent

- Runtime ABL execution process (AVM)
- One per desired concurrent request

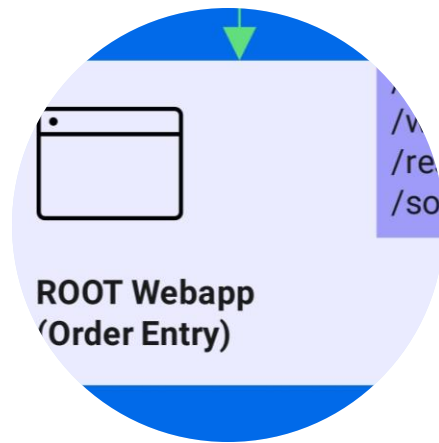


Points of Interest – PAS for OpenEdge Services



Tomcat/PAS

- Heart of the platform
- Customized, extended components



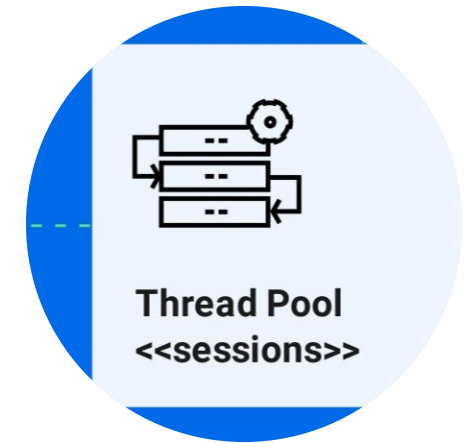
Webapp

- Individual entry point to AVM configuration
- Security, authentication, RBAC, logging



MT Agent

- Multi-threaded Agent
- Configured AVM runtime
- Multiple AVM Sessions (thread)

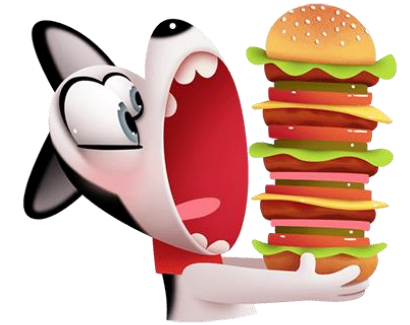


Agent Sessions

- One configured agent
- Multiple threads VS
- Several Agent Processes



Care and Feeding - Configuration



Classic Broker

Control and Config	Target
Start, Stop, Status	wtbman, asbman
Config Management	mergeprop
Configuration Stored	ubroker.properties

PAS for OpenEdge

Control and Config	Target
Start, Stop, Status	pasman, tcman
Config Management	oeprop, secprop
Configuration Stored (per-instance, per-webapp)	catalina.properties appserver.properties openedge.properties server.xml oeablSecurity.properties oeablSecurity.csv logging.xml

Pro Tip: use DLC/bin/paspropconv to kick-start migrating classic broker definitions to PAS (one-time)



Logging – Who doesn't want more logs?

Classic Broker

Source	Target
Broker	broker.log
Agent	agent.log

PAS for OpenEdge

Source	Target (per-instance)
Tomcat	catalina.* <host>.access.*.log
Spring/PAS	<webapp>.*.log <webapp>.authn.log <webapp>.authz.log
Agent	<webapp>.agent.*.log

Pro Tip: fine-tune ABL logging using the OE Logging framework, available in OpenEdge.Core.pl



- Start in <webapp>/WEB-INF/logging.xml
- Identify what area needs the logging changes
 - Tomcat – for high-level access info (start/stop)
 - Spring/PAS – Java, internal workings
 - Agent – typical "***" or ABL event messages

Planning for Success



"Experience is what you gain when things don't go according to plan."

— unknown (sounds like me, though)



Evaluate your application environment

Network

- Identify firewalls between service consumers (clients) and brokers
- Have a network layout diagram that demonstrates end-to-end communication path

Classic Services

- Identify broker definitions, configuration, and codebase

Known Clients

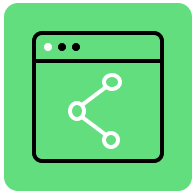
- Identify clients to existing services and how they interact

Ancillary Services

- Identify external or third-party applications that may be accessing your services (e.g. external REST clients, open client).

Progress®

Evaluate your operations



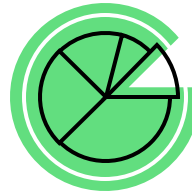
Wants, Needs

- Are current processes critical to operation?
- Are there aspects that haven't been considered?



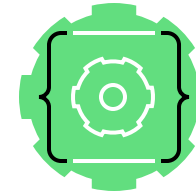
Application Source

- Inventory application assets - entirety
- Evaluate code - **be honest**
- Expect changes - **make them**
- Cleanly rebuild your application



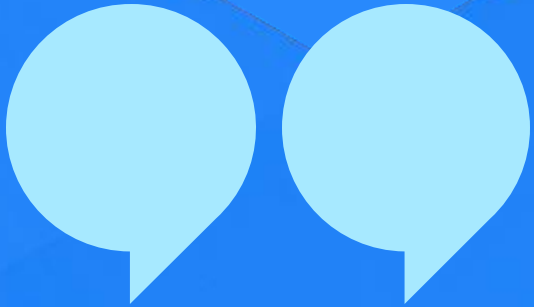
Owners

- Who does what, and when?
- Get SME sign-on
- Agree on metrics for success
- Include launch go/no-go



Target Goals

- What is your authentication and authorization strategy?
- Do you need end-to-end TLS?



"If everything goes right, we get a good experience. If everything goes wrong, we get a good story."

— Simon Sinek



Critically evaluate progression

**Still, Be
Honest**

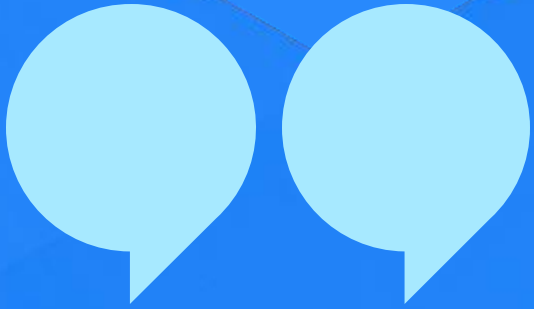
**Test, Test,
Test some
more**

**Deploy,
tear-down,
deploy**

**Concurrent
load testing**

**Test for
failure and
recovery**

Considerations and Expectations



"The secret to success is having low realistic expectations"

— Warren Buffett (original)

Progress® Classic Broker Migration

AppServer		WebSpeed	
Service Type	Expectation, Result	Service	Expectation, Result
REST, SOAP	Service URI will be very similar [<service>]/rest/<resource> [<service>]/soap/<resource>	Uses Messenger	Manage URI mapping to /web/transport
	Re-use classic .paar, .wsm definitions	Application uses Mapped-HTML model	Re-implement with webhandler classes
	For full control of REST swap-out to WebDataObject implementation	CGIWrapper code	Need to use CompatibilityWebHandler Reimplement using webhandler classes
AIA	URI Transition from /aia to /apsv	Application has references to full URI	Change to using paths relative to /web/transport
ABL, OpenClient	Update connection string -URL https://.../apsv Client dictates session model, add -session-Model state-free	Static Assets, where do they go?	PAS can host in /WEB-INF/static/ folder Host in DMZ or on CDN



PRO TIP: Keep track of your instances

	A	B	C	D	E	F	G	H
1			Ports					
2	Instance Name	Location	HTTPS	HTTP	AJP	Health	AgentMin	AgentMax
3	instance1	/srv/instance1	8700	8701	8702	8703		
4								
5	instance2	/srv/appinst2	8800	8801	8802	8803		
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								



Pro Tip #2: use -1 for the shutdown port

I DON'T ALWAYS TEST



**BUT WHEN I DO IT'S IN
PRODUCTION**

Target Production - today

- Start now - *Practice the way you play*
- Use CLI instance tooling pasman, tcman
- Create instances using "-Z prod"
- Learn to recognize which component requires your focus
 - Tomcat
 - Spring (security, RBAC)
 - PAS Classes (session manager, logging)
 - Agent (ABL code, "***" errors)



Secure by default, Features are Opt-in

Opt-in early – at development time

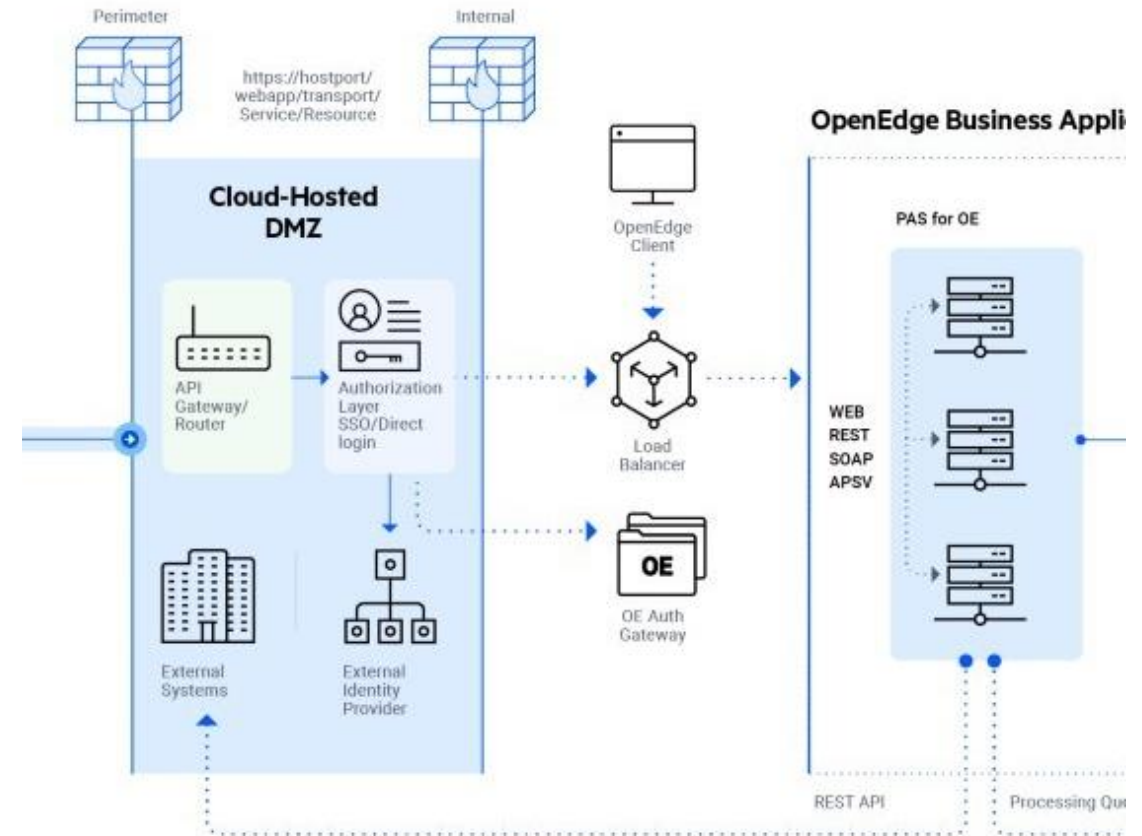
- Enable HTTPS transport
- Create and manage SSL/TLS certificates
- Script certificate management
- Let's Encrypt, certbot
- Let Spring manage Authentication and URI RBAC
- Application code should focus on Authorization
- Integrate with authentication services
 - Oauth2
 - Idap/ad
 - OpenEdge Authentication Gateway

Progress®

Pro Tip: Reverse Proxy



- First step into a larger world
- Cannot 'scale' without one
- Eases configuration
- May improve application responsiveness
- Load balancing, caching, routing
 - Apache HTTPD
 - Nginx
 - Traefik, AWS ALB



Progress Professional Services Team Introduces a New Progress Application Server (PAS) for OpenEdge Task Force

by [Alex Oliveri](#)

| Posted on January 06, 2022

| [0 Comments](#)

Be sure to check out the great blog detailing Progress Services' PAS for OpenEdge Task Force initiative

<https://tinyurl.com/PASforOETask>

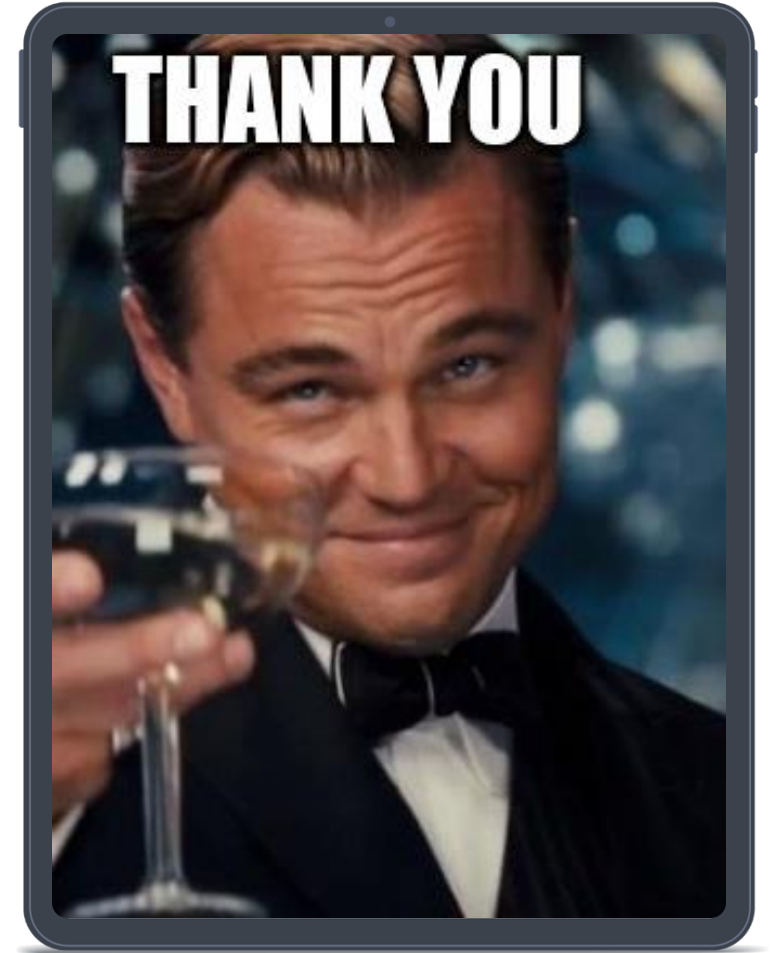
If you are interested in learning more about PAS for OpenEdge best practices or would like implementation help, we'd love to hear from you. Reach out to your Services Sales Rep and we can set up a call to hear about your specific requirements.



Progress® Thanks for attending

Here's to having a successful
PAS for OpenEdge experience.

Presentation slides and configuration examples available on GitHub
<https://github.com/ChadThomsonPSC/webinar-pasoe-migration>



Q&A

