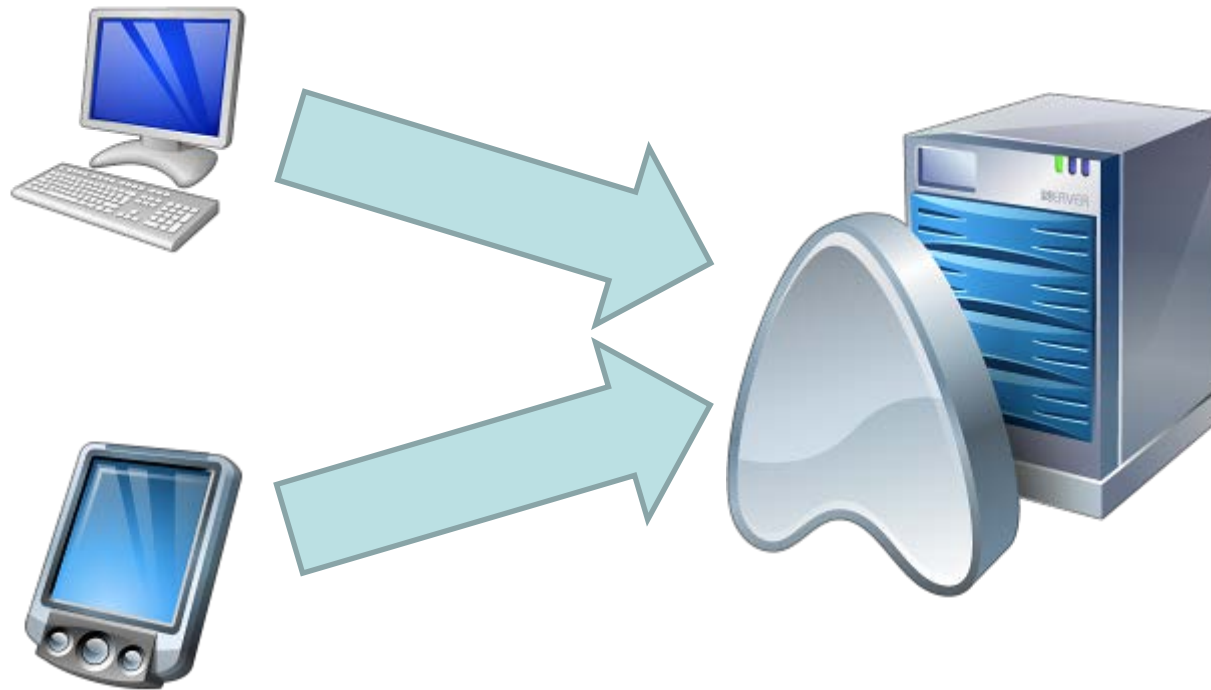
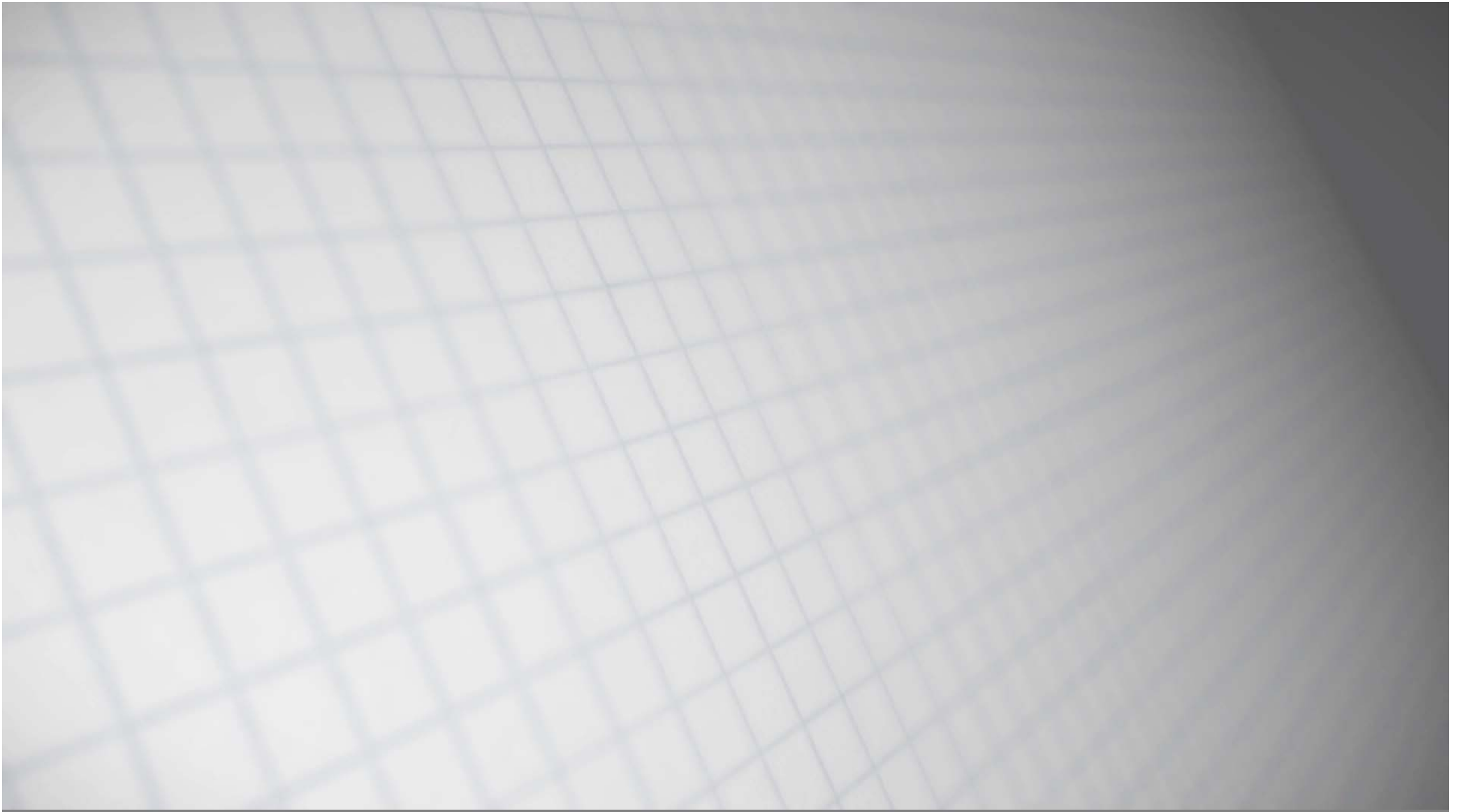


PASOE VS Legacy AppServer



Marko Rüterbories, Consultingwerk Ltd.
marko.rueterbories@consultingwerk.de



Consultingwerk Ltd.



- Independent IT consulting organization
- Focusing on **OpenEdge** and **related technology**
- Located in Cologne, Germany, subsidiary in UK
- Customers in Europe, North America, Australia and South Africa
- Vendor of developer tools and consulting services
- 27 years of Progress experience (V5 ... OE11)
- Specialized in GUI for .NET, Angular, OO, Software Architecture, Application Integration

Introduction

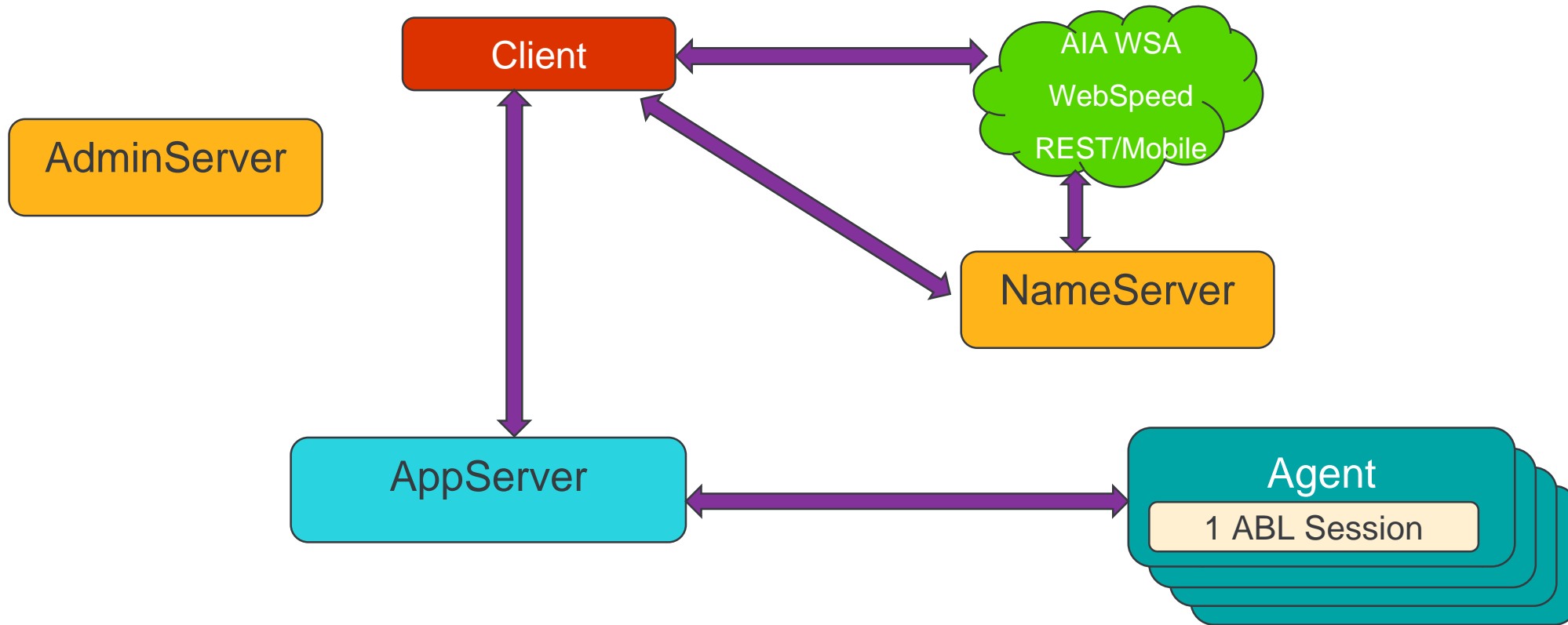
- Legacy is classic 😊
- Flexible and extensible part of a Service-oriented architecture
- Standards-based transaction engine
- Provides user interface independence
- Enables business logic to be easily distributed and reused
- Centralized business logic provides a single point to manage access to data

Agenda

- **Overview legacy AppServer**
- Overview PASOE
- Comparison
- Setup and Configuration
- Client connection from the ABL
- Deployment Options / Strategies



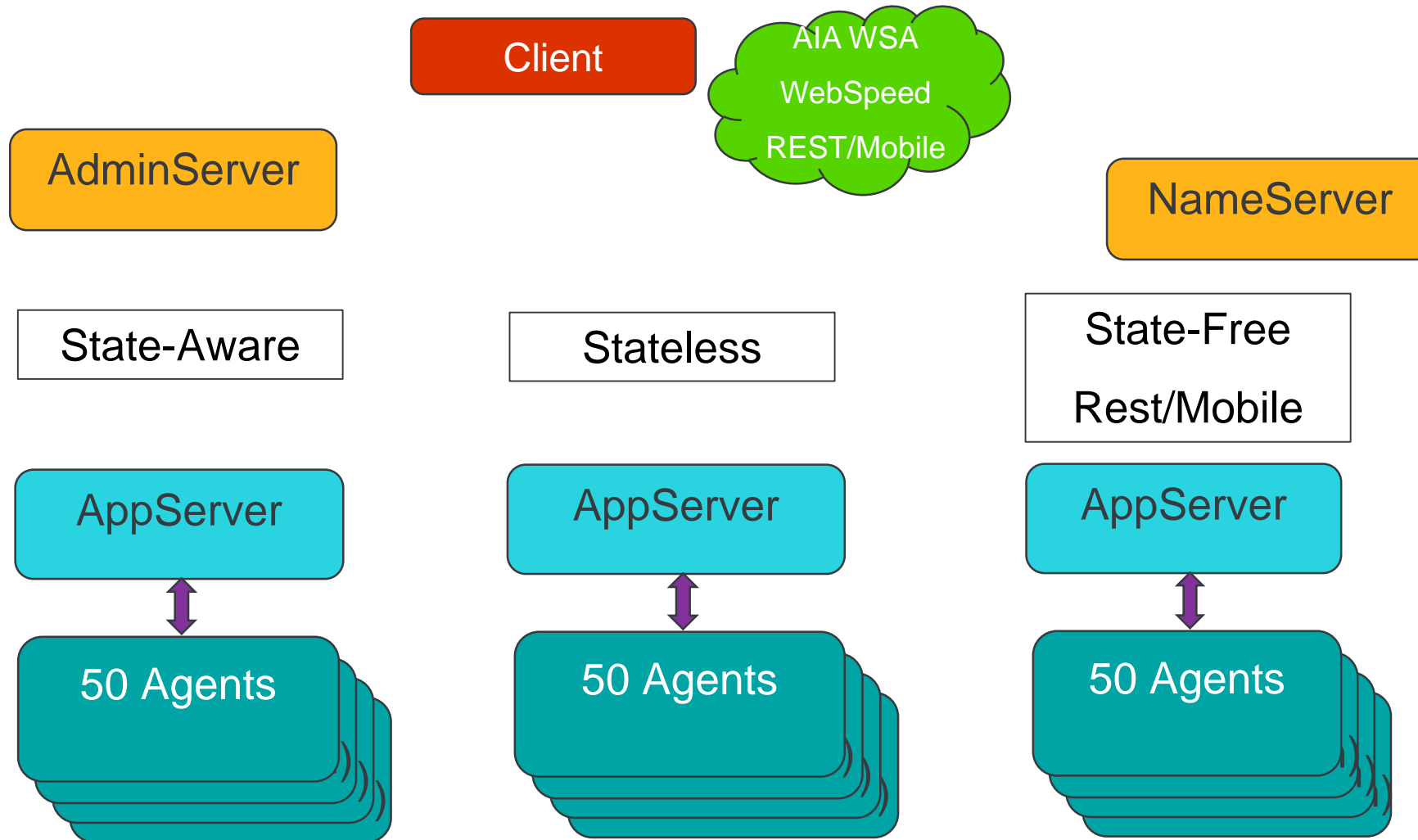
Overview legacy AppServer



Overview legacy AppServer

- AppServer Broker
 - Manages pools of re-usable AppServer agents
 - Manages client connections and requests for AppServer agents to execute ABL
- AppServer Agents
 - Executes OpenEdge ABL procedures
 - The reuse of AppServer agents is determined by Operating mode

Overview legacy AppServer

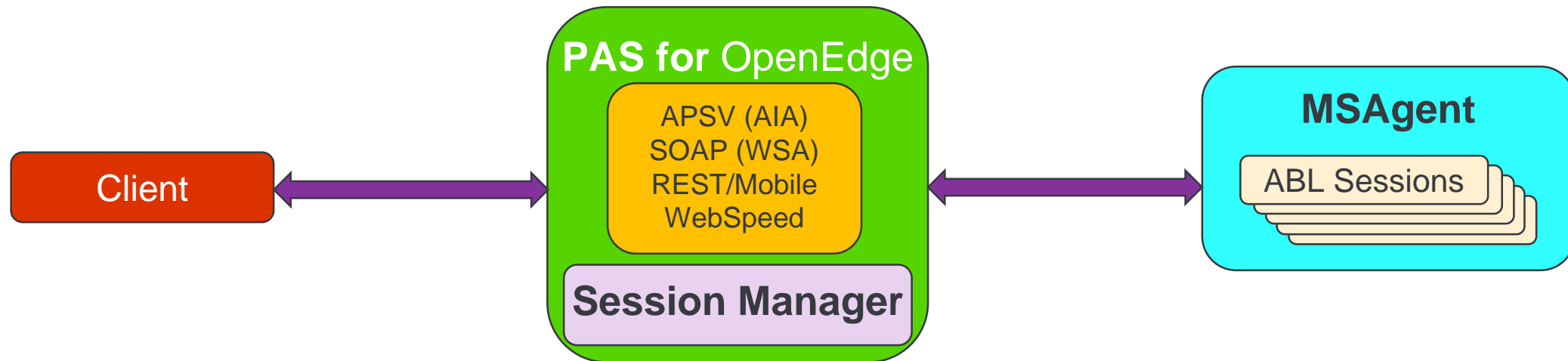


Agenda

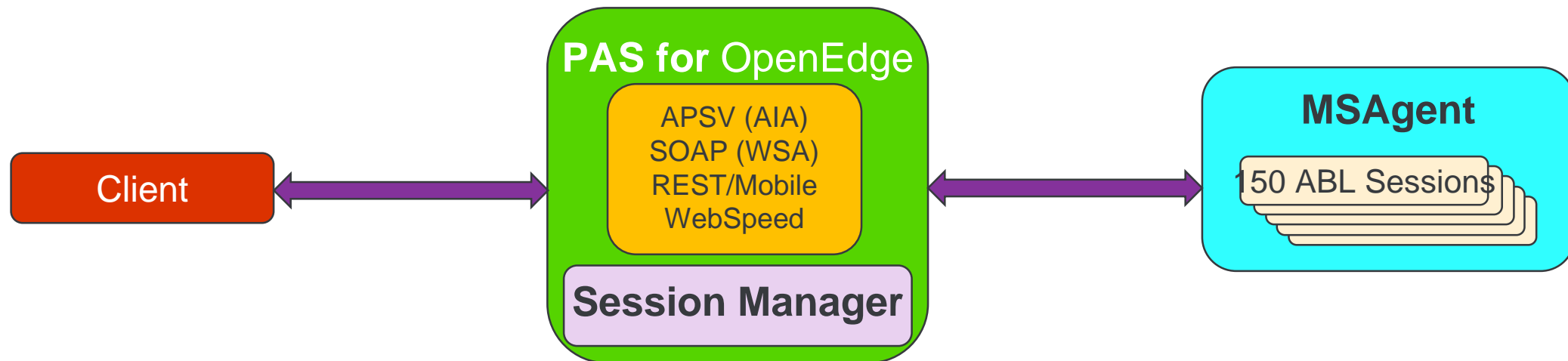
- Overview legacy AppServer
- **Overview PASOE**
- Comparison
- Setup and Configuration
- Client connection from the ABL
- Deployment Options / Strategies



Overview PASOE

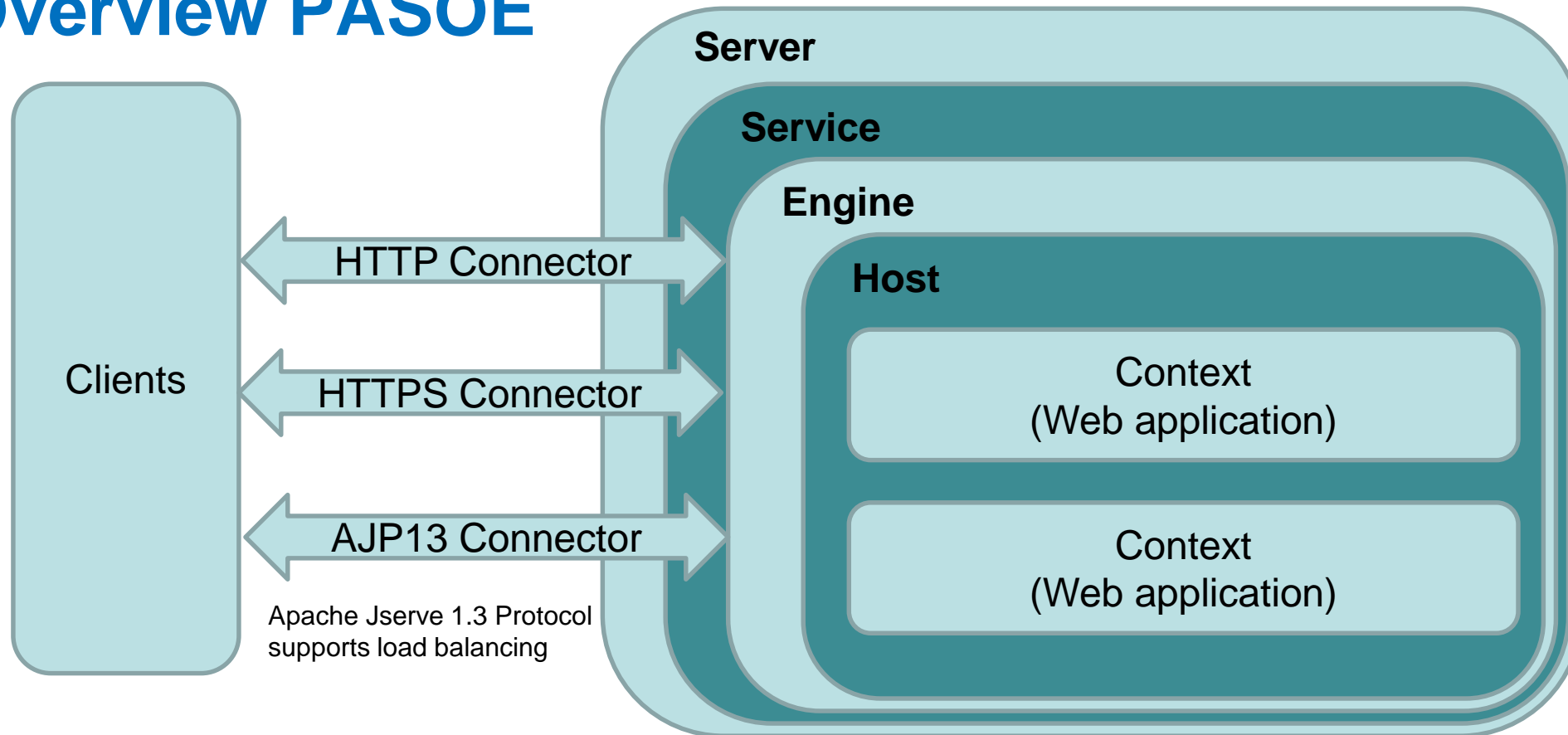


Overview PASOE



- Just one Multi-Session Agent able to utilize all kind of clients and operation modes

Overview PASOE



- Progress proprietary network protocol has been dropped
- Tomcat does not speak TCP

Agenda

- Overview legacy AppServer
- Overview PASOE
- **Comparison**
- Setup and Configuration
- Client connection from the ABL
- Deployment Options / Strategies



Comparison

Legacy Appserver

- Must have an AdminServer
- AdminServer starts the NameServer and AppServer, then the AdminServer is just used to monitor and manage
- Client connects to NameServer (optional) and is then redirected to the AppServer
- If the Client needs to connect through the internet (HTTP/HTTPS) you must install and configure separate and distinct adapters AIA/WSA/REST/Mobile

Comparison

PASOE

- Start the PAS for OE instance
- Client connects using HTTP/HTTPS always
- AdminServer is not necessary (except for running OEMgmt on the same machine or publishing)
- There is NO NameServer (load balancing and fault tolerance are handled another way)

Comparison

- **Session Models and Operating Modes** for legacy AppServer
 - Specified when AppServer is configured
 - Determines how client requests are dispatched to individual Application Server Agents
 - Considerations at Design time
 - Application context
 - Performance optimization

Comparison

- Connection management
 - Session managed model
 - Client sends requests over „bound“ connection
 - Requests are handled sequentially
 - Operating Modes
 - State-reset (Progress V8.2)
 - State-aware (Progress V9)
 - Stateless (Progress V9)

Comparison

- Connection management
 - Session free model
 - Client sends request on any available connection
 - Requests handled in parallel
 - Operating Mode
 - State-free (OpenEdge 10.0A)

Comparison (Architecture, capabilities)

- *Operating Modes*
 - State-reset
 - One client per Application Server Agent
 - Session state is reset on disconnect
 - State-aware
 - One client per Application Server Agent
 - Session state is maintained across connections
 - Stateless
 - Connection managed by AppServer Broker
 - Many clients per application server agent
 - Context must be managed programmatically

Comparison

- AppServer operating modes can be programmatically adopted to PASOE
- Unbound by default
- Bound session can be achieved by
 - Having the client run a remote persistent procedure
 - Setting SESSION:SERVER-CONNECTION-BOUND-REQUEST in the Connect and Disconnect procedure's
- Only ABL clients, .NET and Java OpenClient can use bound sessions
- Web, SOAP and REST clients can not use this

Comparison

Migrating classic state-reset and state-aware operating mode

To achieve the same client **bound** and **reset** behavior in PASOE, include the following as the first statement in your Connect event procedure:

```
SESSION:SERVER-CONNECTION-BOUND-REQUEST = TRUE.
```

This will fully disable the execution of Activate and Deactivate event procedures for client requests over this connection until this attribute is set to FALSE, but leave them available for use by session-free client connections.

Comparison

Also include the following as the last statements before exiting your Disconnect event procedure:

```
SESSION:SERVER-CONNECTION-BOUND-REQUEST = FALSE.  
QUIT. (only for state-reset)
```

The QUIT statement in PASOE effectively performs the same ABL session reset operation that occurs in the classic AppServer after the Disconnect event procedure executes.

Comparison

- **Migrating classic stateless and state-free operating mode**
- No changes are required to event procedures to migrate applications running on a classic stateless AppServer.

Comparison

- More details on migrating from legacy Appserver to PASOE
- https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/pasoe-migrate-develop/migrating-appserver-operating-modes.html

Comparison

- Startup and Shutdown procedures

Procedure name	Legacy AppServer	PASOE	PASOE procedure name
		Executed on Server startup	agentStartupProc
		Executed on Server shutdown	agentShutdownProc
svrStartupProc	Executed on Session startup		sessionStartupProc
svrShutdownProc	Executed on Session shutdown		sessionShutdownProc
svrConnectProc	Executed on Client connect		sessionConnectProc
svrDisconnProc	Executed on Client disconnect		sessionDisconnProc
svrActivateProc	Executed on Server startup	Executed on Session activate	sessionActivateProc
svrDeactivateProc	Executed on Server shutdown	Executed on Session deactivate	sessionDeactivateProc

Comparison

Full comparison of event procedures can be found here:

https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/pasoe-migrate-develop%2Fcomparing-event-procedures-between-the-appserver.html%23

Comparison

Improvements in PASOE

- Multi-Session Agent
 - Single process supports multiple, concurrent, ABL sessions
 - Sessions share resources
 - Manages shared memory connects
- Session Manager integrated with Tomcat
 - No separate Java processes for Ubroker, NameServer or AdminServer (optional)
 - Removes one network copy in stack

Comparison

	Classic	PAS for OE	Improvement
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			YMMV
WebSpeed performance			
Round trip			YMMV

Agenda

- Overview legacy AppServer
- Overview PASOE
- Comparison
- **Setup and Configuration**
- Client connection from the ABL
- Deployment Options / Strategies



Setup and Configuration

The screenshot shows the Progress OpenEdge Explorer interface. The browser address bar indicates the URL is localhost:9090/resources/resourcegrid.jsp. The user is logged in as 'admin'. The main content area displays a table of resources for the 'desktop-2lcvicv' environment. The table has columns for 'Select', 'Resource', 'Type', 'Status', and 'Alerts'. There are two main resource groups: 'AdminServer' (1 item) and 'AppServer' (6 items). The 'AdminServer' group contains one resource, 'desktop-2lcvicv', which is of type 'AdminServer' and is 'Online'. The 'AppServer' group contains six resources, all of type 'AppServer'. One resource, 'desktop-2lcvicv.TestServer1', is 'Running', while the other five are 'Not Runni...'. A 'Resource Summary' panel on the right shows a 'Resource Count' of 0.

Resources / desktop-2lcvicv

Resources

--Search on resource name or tag | --Type-- | --Status-- | --Group By--

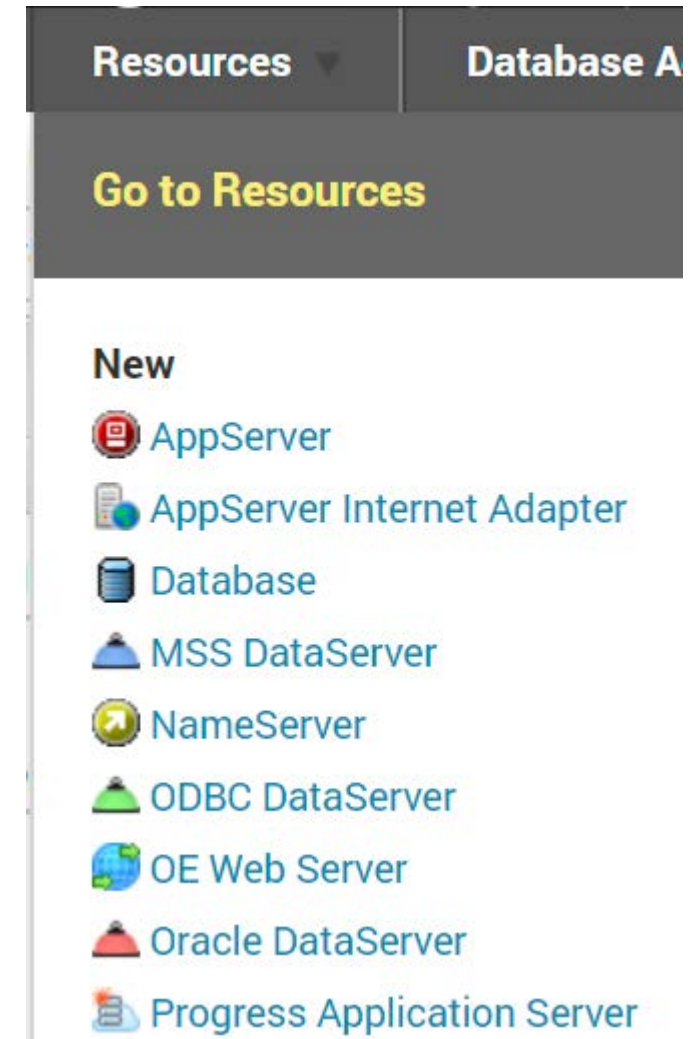
Select	Resource	Type	Status	Alerts
[-]	AdminServer (1 Item)			
<input type="checkbox"/>	desktop-2lcvicv	AdminServer	Online	
[-]	AppServer (6 Items)			
<input type="checkbox"/>	desktop-2lcvicv.TestServer1	AppServer	Running	
<input type="checkbox"/>	desktop-2lcvicv.asbroker1	AppServer	Not Runni...	
<input type="checkbox"/>	desktop-2lcvicv.bpsbroker1	AppServer	Not Runni...	
<input type="checkbox"/>	desktop-2lcvicv.esbbroker1	AppServer	Not Runni...	
<input type="checkbox"/>	desktop-2lcvicv.icfrepos	AppServer	Not Runni...	
<input type="checkbox"/>	desktop-2lcvicv.restbroker1	AppServer	Not Runni...	

Resource Summary

Resource Count: 0

Setup and Configuration

- Add new Resource from the Resources Toolbox



Setup and Configuration

- Add new AppServer Resource
- Name the Instance
- Save - the new Instance is created
- Configure the AppServer as needed

Progress® OpenEdge® Explorer

Resources Database Administration

Resources / OpenEdge / AppServer / New AppServer

AppServer
Create new AppServer

SAVE CANCEL

Please select a new AppServer name

New AppServer name*: TestServer1

Setup and Configuration

- Broker configuration
 - Port number (default 3090) may already in use
 - Operating Mode

AdminServer: desktop-2lcvicv
AppServer: TestServer1
Configuration

EDIT

[Broker](#) [Agent](#) [SSL](#) [Messaging](#) [Environment Variables](#)

General

Auto start:	<input type="checkbox"/>
Port number:	3190
Working directory:	@{WorkPath}
Operating Mode:	State-reset
Enable NetBios:	<input checked="" type="checkbox"/>

Owner Information

Username:	Not set
Group name:	Not set
Password:	Not set

Controlling NameServer

Register with NameServer:	<input checked="" type="checkbox"/>
Controlling NameServer:	NS1
Registration mode:	Register-IP
Registration host name:	Not set

Setup and Configuration

- Agent configuration
 - Startup parameters
 - PROPATH

AdminServer: desktop-2lcvicv
AppServer: TestServer1
Configuration

EDIT

Broker Agent SSL Messaging Environment Variables

General

Server executable file:	"@{Startup\DLC}\bin_proapsv.exe"
Server startup parameters:	-db C:\Work\AppServerTalk\DB\sports2000.db -H localhost
PROPATH:	@{WinChar Startup\PROPATH};@{WorkPath};c:\Work\AppServerTalk\ABL
Minimum port number:	2002
Maximum port number:	2202
Flush statistical data:	255

Logging Setting

Server log filename:	@{WorkPath}\TestServer1.server.log
Server logging level:	Basic
Append to server log file:	<input checked="" type="checkbox"/>
Server logging entry types:	ASPlumbing,DB.Connects
Server log file threshold size:	0
Maximum number of server log files:	3
Server watch dog interval:	60

Pool Range

Initial number of connections:	1
--------------------------------	---

Setup and Configuration

- Agent configuration
 - Number of Agents to start
 - Startup and Shutdown Procedures
Used f.e. to initialize the backend session

Pool Range

Initial number of servers to start:	1
Minimum servers:	1
Maximum servers:	10

Advanced Features

4GL debugger enabled:	<input type="checkbox"/>
Activate procedure:	<i>Not set</i>
Deactivate procedure:	<i>Not set</i>
Connect procedure:	<i>Not set</i>
Disconnect procedure:	<i>Not set</i>
Startup procedure:	<i>Not set</i>
Shutdown procedure:	<i>Not set</i>
Startup procedure parameters:	<i>Not set</i>
Execution Time Limit:	0

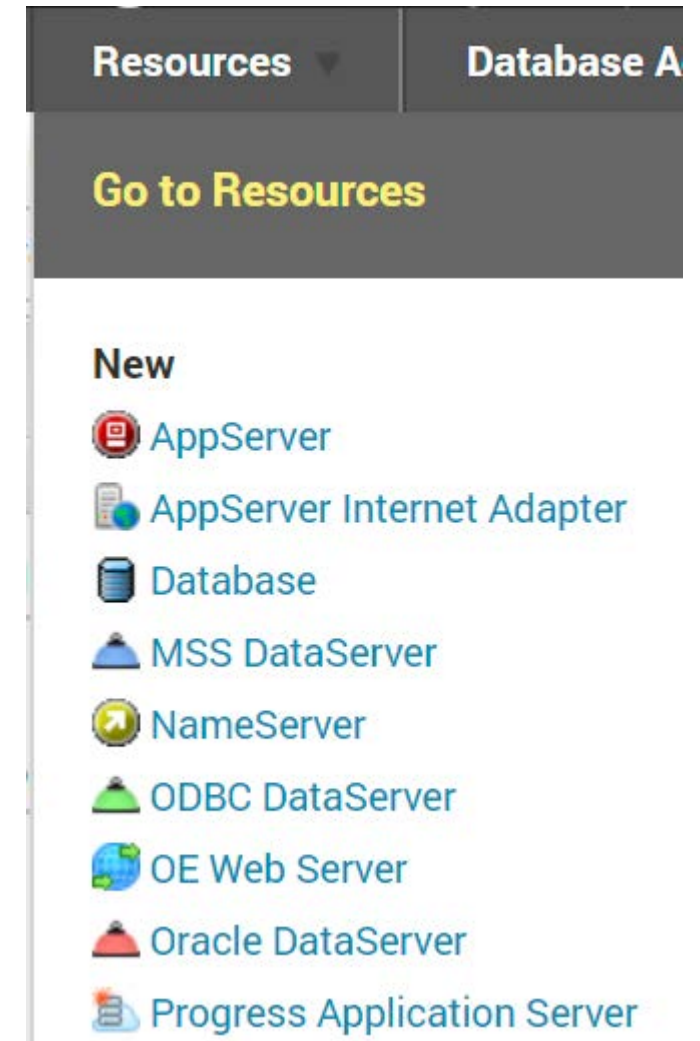
Setup and Configuration

- Ubroker.properties

```
[UBroker.AS.TestServer1]
  appserviceNameList=TestServer1
  brkrDebuggerKeyAlias=default_server
  brokerLogFile=@{WorkPath}\TestServer1.broker.log
  controllingNameServer=NS1
  environment=TestServer1
  initialSrvrInstance=1
  keyAlias=default_server
  mqBrokerLogFile=@{WorkPath}\TestServer1.mqbroker.log
  mqServerLogFile=@{WorkPath}\TestServer1.mqserver.log
  portNumber=3190
  PROPATH=@{WinChar Startup\PROPATH};@{WorkPath};c:\Work\AppServerTalk\ABL
  srvrLogFile=@{WorkPath}\TestServer1.server.log
  srvrStartupParam=-db C:\Work\AppServerTalk\DB\sports2000.db -H localhost
  uuid=83a1dda7d9d8c1fe:-47431f8:15fa1ff2833:-75fb
```

Setup and Configuration

- Add new Resource from the Resources Toolbox



Progress Application Server

Create new Progress Application Server

CANCEL CREATE

Create Progress Application Server

Instance Name*:	<input type="text" value="PASTestServer"/>
Location:	<input checked="" type="radio"/> Local <input type="radio"/> Remote
URL:	<input type="text" value="https://host:port/oemanager"/> <input checked="" type="checkbox"/> No host verification for SSL
Instance Directory:	<input type="text" value="c:\Work\AppServerTalk\pasoe"/>
HTTP port*:	<input type="text" value="8080"/>
HTTPS port*:	<input type="text" value="8443"/>
Shutdown port*:	<input type="text" value="8081"/>
	<input checked="" type="checkbox"/> Autostart
Tomcat manager login*:	<input type="text" value="tomcat"/>
Tomcat manager password*:	<input type="password" value="....."/>
OpenEdge manager authentication:	<input checked="" type="radio"/> Use Tomcat manager username and password: <input type="radio"/> Login to OE Manager with username and password:
OpenEdge manager login:	<input type="text"/>
OpenEdge manager password:	<input type="password"/>

Setup and Configuration

- Add new Progress Application Server
- Name the new Instance of PASOE
- Define the Instance Directory
- On Windows - define the Shutdown port
- Supply Tomcat manager the login and password details

Setup and Configuration

- Define Startup parameters
- Set PROPATH
- Set Session procedures

AdminServer: desktop-2lcvicv
Progress Application Server: PASTestServer
Application configuration for PASTestServer

Optionally provide parameter value to be passed to the agent startup procedure.

Startup Parameters and Environment Log Settings Performance Tuning and Limits Network Environment Configure advanced settings

Startup Parameter and Environment Settings

Agent working directory:
Agent startup parameter:
PROPATH:

Session procedures

Activate procedure:
Deactivate procedure:
Connect procedure:
Disconnect procedure:
Session startup procedure:
Startup procedure parameter:
Shutdown procedure:

Agent procedures

Startup procedure:

Setup and Configuration

- Openedge.properties

```
[AppServer]
collectMetrics=1
applications=PASTestServer
statusEnabled=1
allowRuntimeUpdates=0

[AppServer.SessMgr.PASTestServer]
agentLogEntryTypes=ASPlumbing,DB.Connects
agentStartupParam=-T "${catalina.base}/temp" -db C:\Work\AppServerTalk\DB\sports2000.db -H localhost
collectMetrics=3
agentLogFile=${catalina.base}/logs/PASTestServer.agent.log

[AppServer.Agent]
numInitialSessions=5
agentTableParam=-n 200
sessionStartupProcParam=
sessionConnectProc=
agentStartupProc=
noSessionCache=0
lockAllExtLib=
agentMinPort=62002
sessionStartupProc=
usingThreadSafeExtLib=
workDir=${CATALINA_BASE}/work
binaryUploadMaxSize=0
```

Setup and Configuration

- Openedge.properties

```
[AppServer.Agent.PASTestServer]
numInitialSessions=2
uuid=http://DESKTOP-2LCVICV:8443/PASTestServer
keyAliasPasswd=
PROPATH=c:\Work\AppServerTalk\ABL,${CATALINA_BASE}/webapps/ROOT/WEB-INF/openedge,${CATALINA_BASE}/opene
```


Setup and Configuration

- Another way of creating a new PASOE instance is using the tcman command
c:\progress\OpenEdge117_64\servers\pasoe\bin\tcman create -p 8830 -s 8831 -p 8832 -j 8833 TestPASOE
- The tcman command can also be used for the Startup and shutdown of a PASOE instance
{pasoe folder}\bin\tcman {start|stop}
- TRIM Agents (that were the old days 😊)
- On PDSOE you need to TRIM Session, not the whole MSAgent

Setup and Configuration

Classic AppServer tools

- asbman –query
- OpenEdge Explorer
- OpenEdge Management

PASOE

- Management REST API
 - The “oemanager” application must be installed for access to the REST API
- JMX access
 - Using Jconsole
 - Locally by PID
 - Remotely - “tcman.sh/.bat feature JMXLifecycle=on”
- Nagios
 - 3rd party tool recommended by Tomcat for monitoring
 - Uses the tools above to monitor, alert and graph
- OpenEdge Explorer and Management

Agenda

- Overview legacy AppServer
- Overview PASOE
- Comparison
- Setup and Configuration
- **Client connection from the ABL**
- Deployment Options / Strategies



Client connection from the ABL

- REST/Mobile clients: URL still uses ../rest/..
- SOAP clients: ../soap/.. instead of ../wsa/..
- WebSpeed: ../web/.. Instead of ../cgi-bin/cgi.sh/..
- OpenEdge clients URL connection format only
 - URL [http://myhost:8810/\(appname\)/apsv](http://myhost:8810/(appname)/apsv)
No (appname) assumes ROOT -URL <http://myhost:8810/apsv>
- Reference: *Connecting to AppServers Using a URL*

NOTE: Since all of the old web servlets are now encapsulated in PASOE we use these “transports” to identify the communication: rest, soap, apsv and web

Client connection from the ABL

- A SERVER object is used to connect from ABL source code to either type of AppServer

```
DEFINE VARIABLE hServer AS HANDLE NO-UNDO.  
CREATE SERVER hServer.
```
- Connecting to legacy AppServer

```
hServer:CONNECT ("-AppService TestServer1 -H localhost -s  
3190").
```
- Connecting to PASOE

```
hServer:CONNECT ("-URL http://localhost:8080/apsv").
```

Client connection from the ABL

- Demo
 - Get simple response
 - Get data from DB
- Show how to use the same sourcecode in a local session to connect to a legacy AppServer and PASOE

Agenda

- Overview legacy AppServer
- Overview PASOE
- Comparison
- Setup and Configuration
- Client connection from the ABL
- **Deployment Options / Strategies**



Deployment Options / Strategies

- Many roads lead to Rome 😊
- Options you may need
 - Port forwarding from a Gateway
 - Rewrite Rules (www.myDomain.com/{AppName})
 - LoadBalancing
- Never use PASOE Development as a production instance!!!
 - Not only because of licensing

PASOE Development

PASOE Production

Can compile code

Cannot compile code

Non-secure configuration

Secure configuration

Test server instance in \$WRKDIR

No test server instances

Remote administration included

Remote administration optional

Tomcat remote admin enabled

Tomcat remote admin optional

OpenEdge remote admin enabled

OpenEdge remote admin optional

Built-in oeabl web application (ROOT)

Built-in oeabl web application (ROOT)

All transport deployed and enabled

All transports deployed but disabled

Restricted: 5 concurrent requests and 1 agent

Unrestricted: concurrent requests and agents

defaultHandler=

defaultHandler=

OpenEdge.Web.CompatibilityWebHandler

OpenEdge.Web.DefaultWebHandler

Questions

