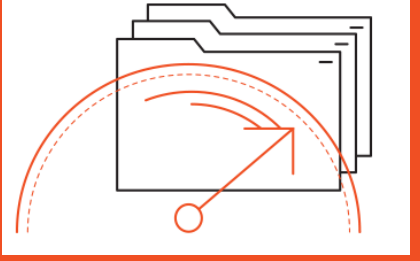


Enhanced WebSpeed

Webspeed …. I am back

PROGRESS

OpenEdge 11.6 – WebSpeed!!!

- Modernize your Progress OpenEdge web apps through enhanced Progress Application Server (PAS) support for WebSpeed

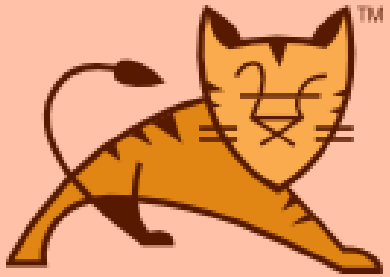
- Achieve improved performance and web server scalability

- Leverage existing investment, while minimizing development costs and risks

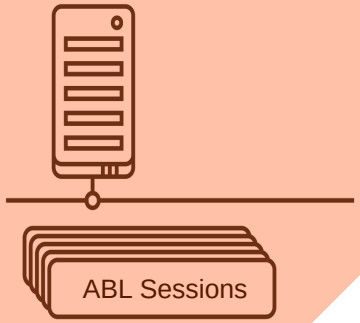
PAS for OpenEdge with WebSpeed

UNIFIED AND HIGHLY SCALABLE APPLICATION SERVER

Helping enterprises reduce the total cost of ownership and maintenance



Built on Apache Tomcat Web server



Highly scalable multi-session architecture

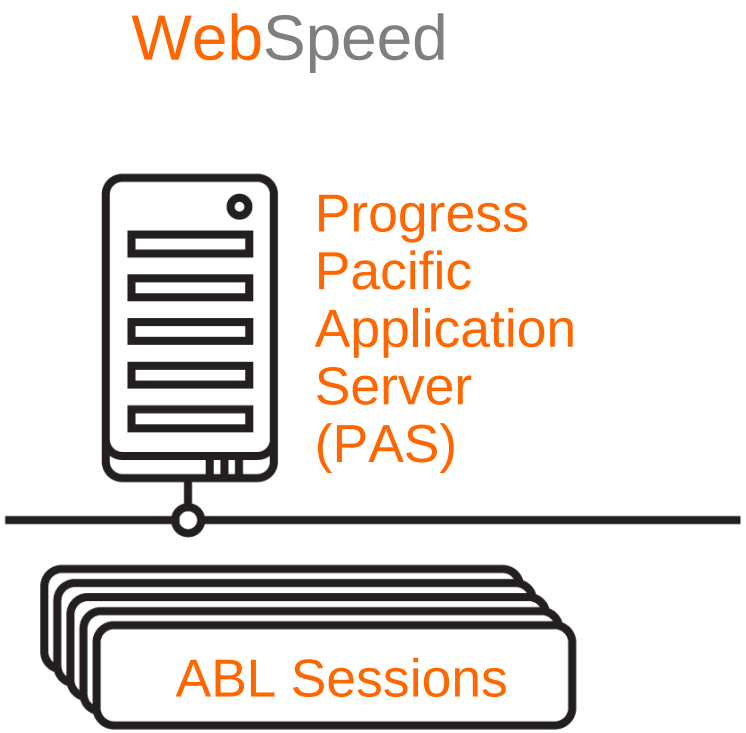
Production ready, out-of-the-box security

Administration with OpenEdge Management

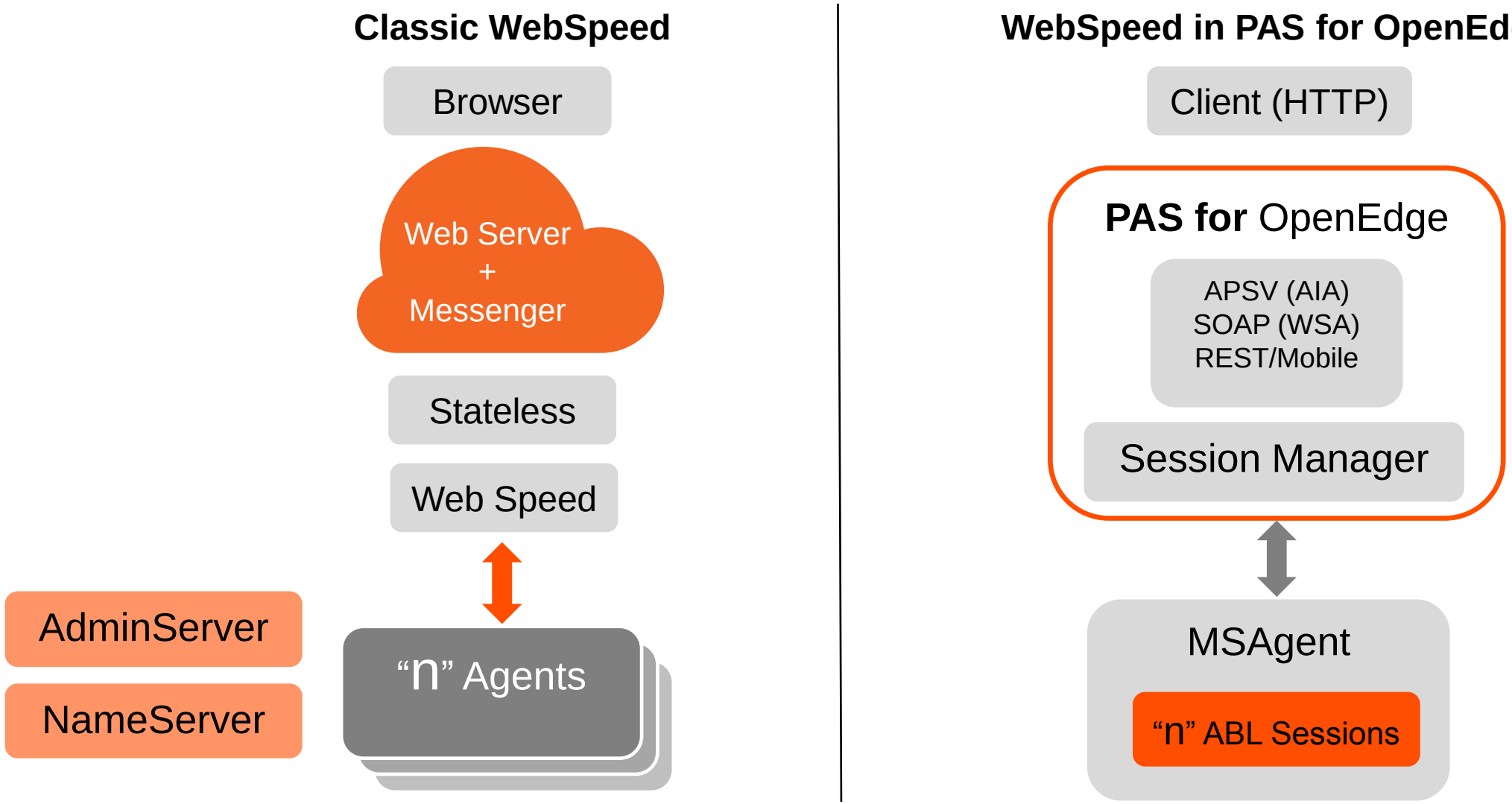
PAS for OpenEdge with WebSpeed Support

Next generation Application Server
that integrates multiple functions

- ✓ **Same core server** for Rollbase, Corticon, and OpenEdge AppServer / WebSpeed ABL applications
- ✓ ABL applications run in the new **multi-session agent,** which uses a single OS process for improved performance and scalability
- ✓ Simpler administration / deployment and is **preconfigured** to operate as a **production-ready** server



Classic WebSpeed vs. PAS for OpenEdge

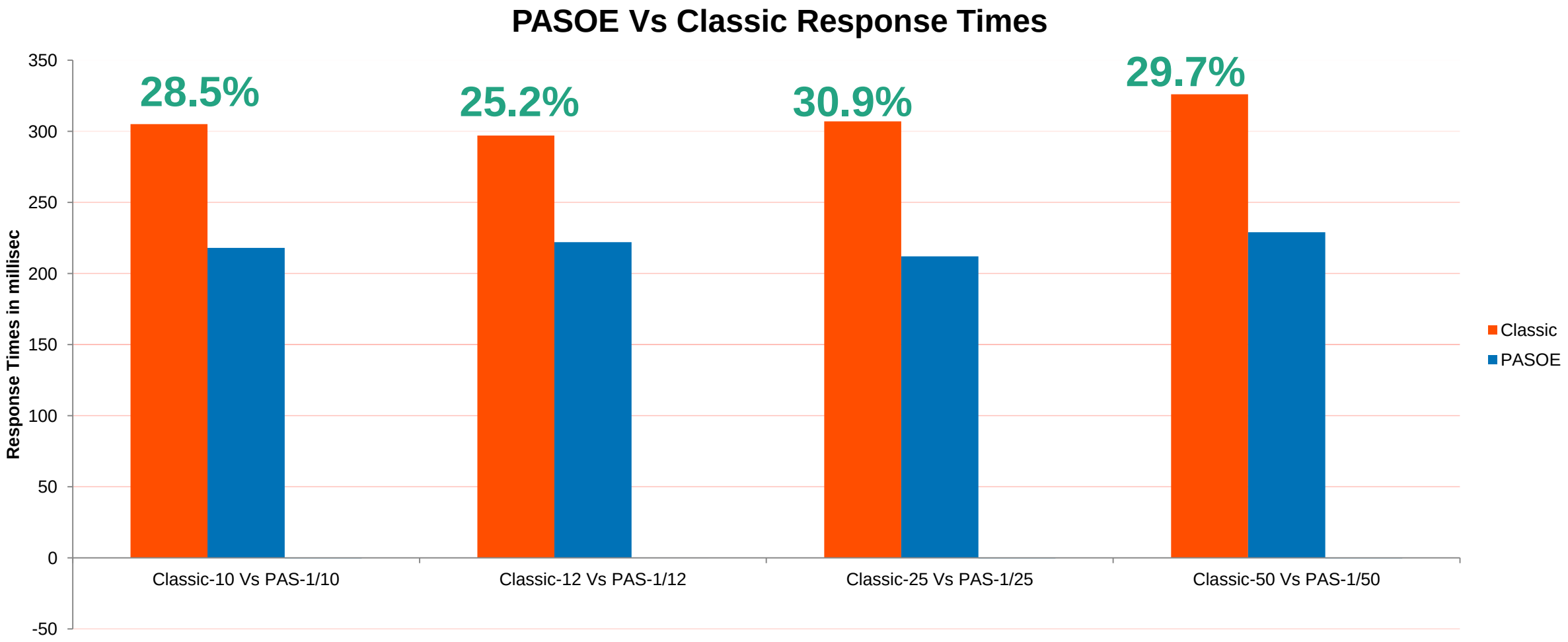


WebSpeed Migration Benefits

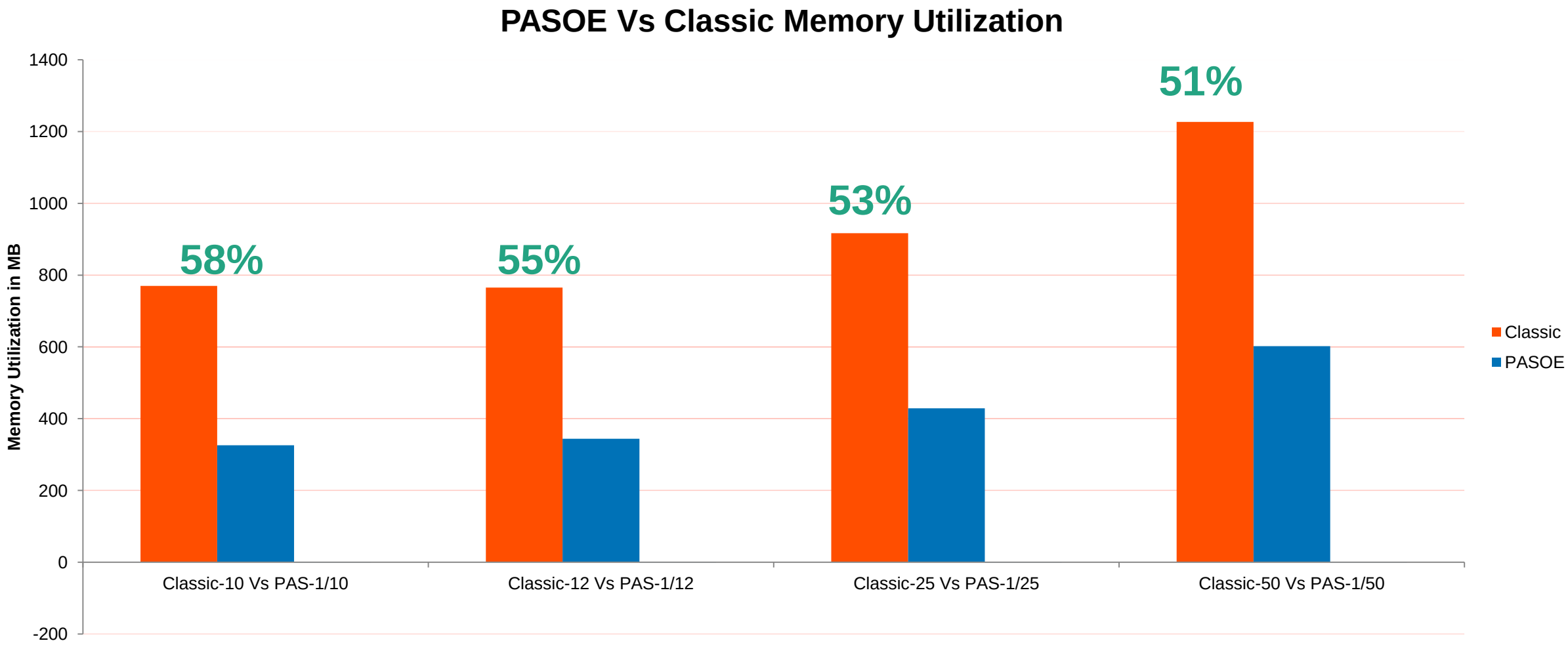
- Uses less system resources- more scalable
- Runs existing* WebSpeed application code
- Supports web application security requirements

- Well-defined deployment models & industry standard API's for monitoring
- Migration path to full HTTP request/response handling & present day UI technologies

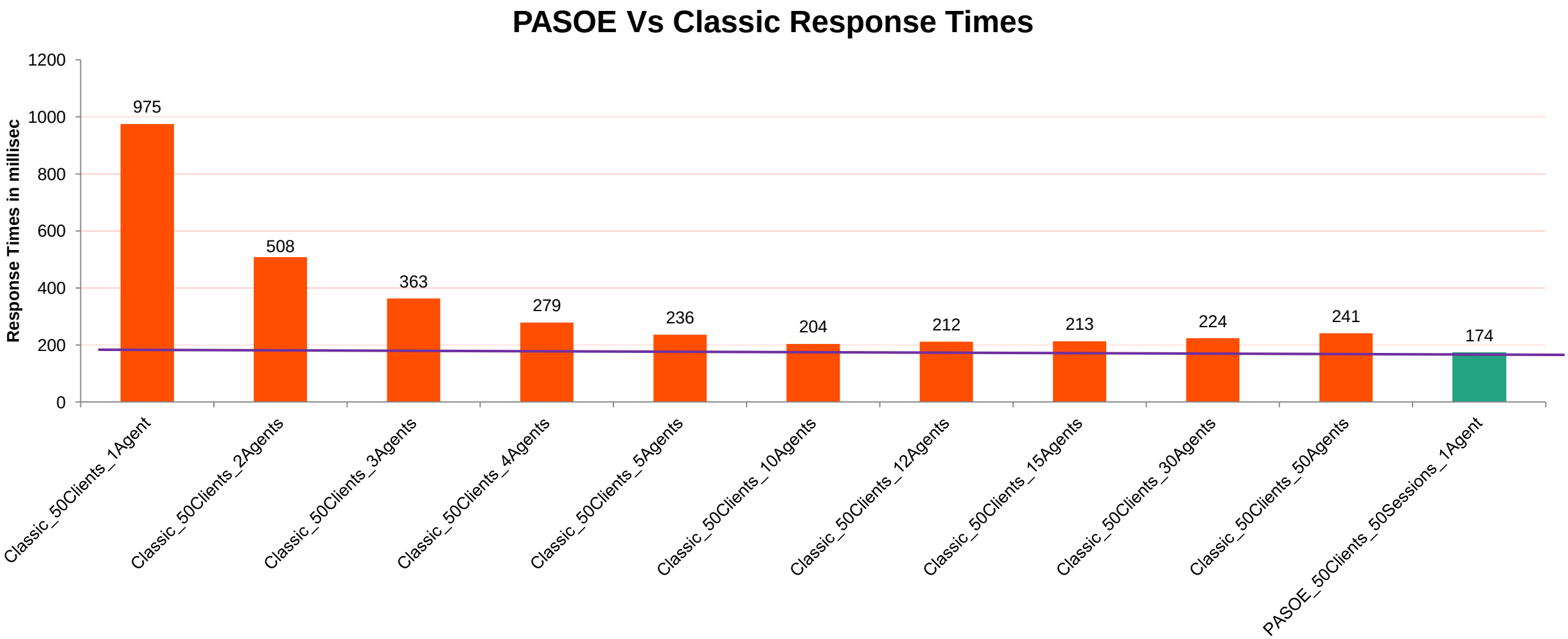
Webspeed - 25KB of Payload on Linux



Webspeed - 25KB of Payload on Linux

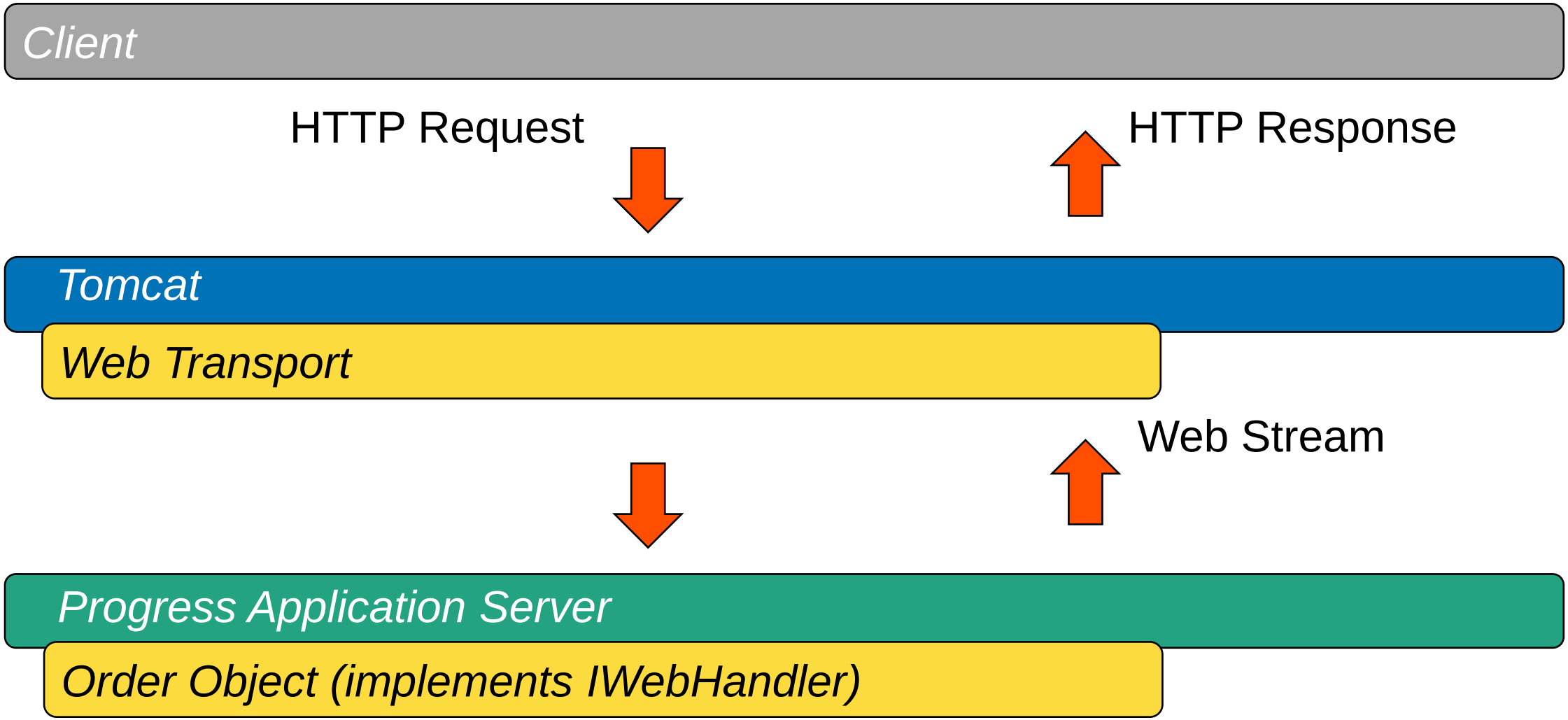


Webspeed - 5.3KB of Payload on Linux



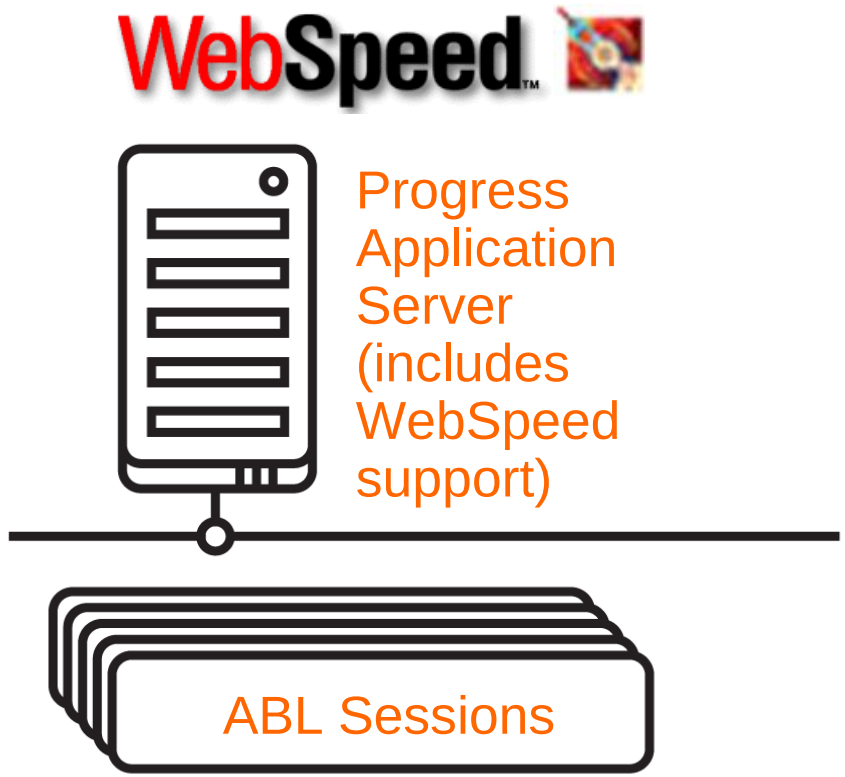
OpenHTTP

Introduction



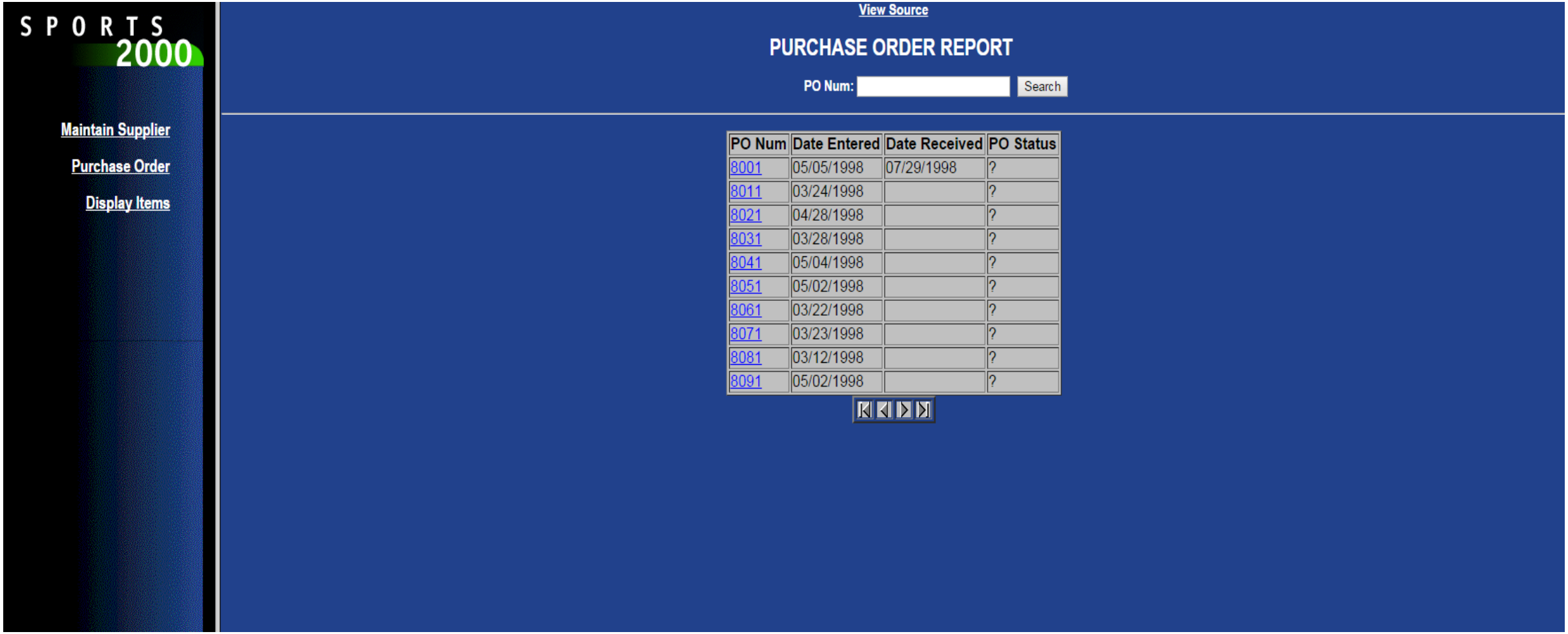
WebSpeed OpenHTTP

- Writing your own IWebHandler implementation from scratch can be complex
- Most applications will not use all HTTP verbs
- Some behavior is frequently common for all handlers

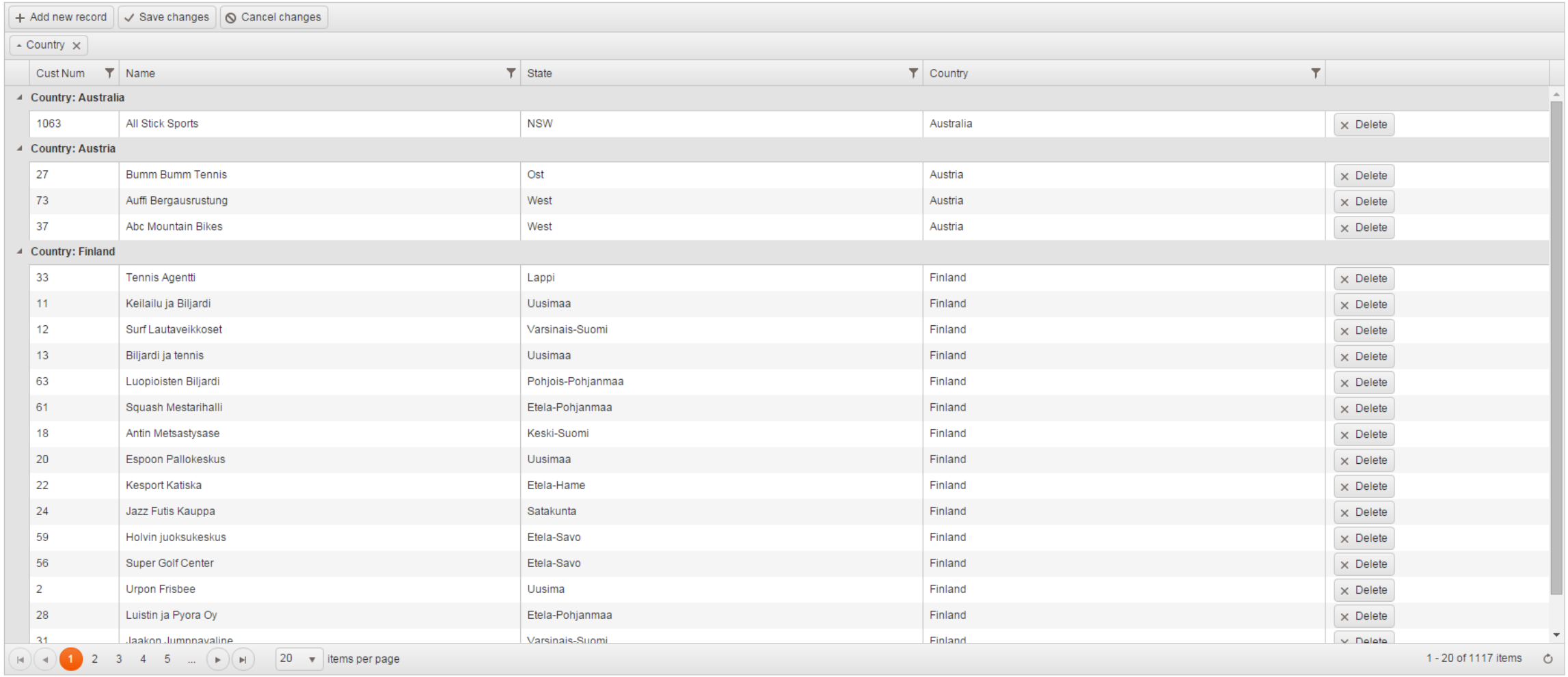


Demo Webspeed PASOE

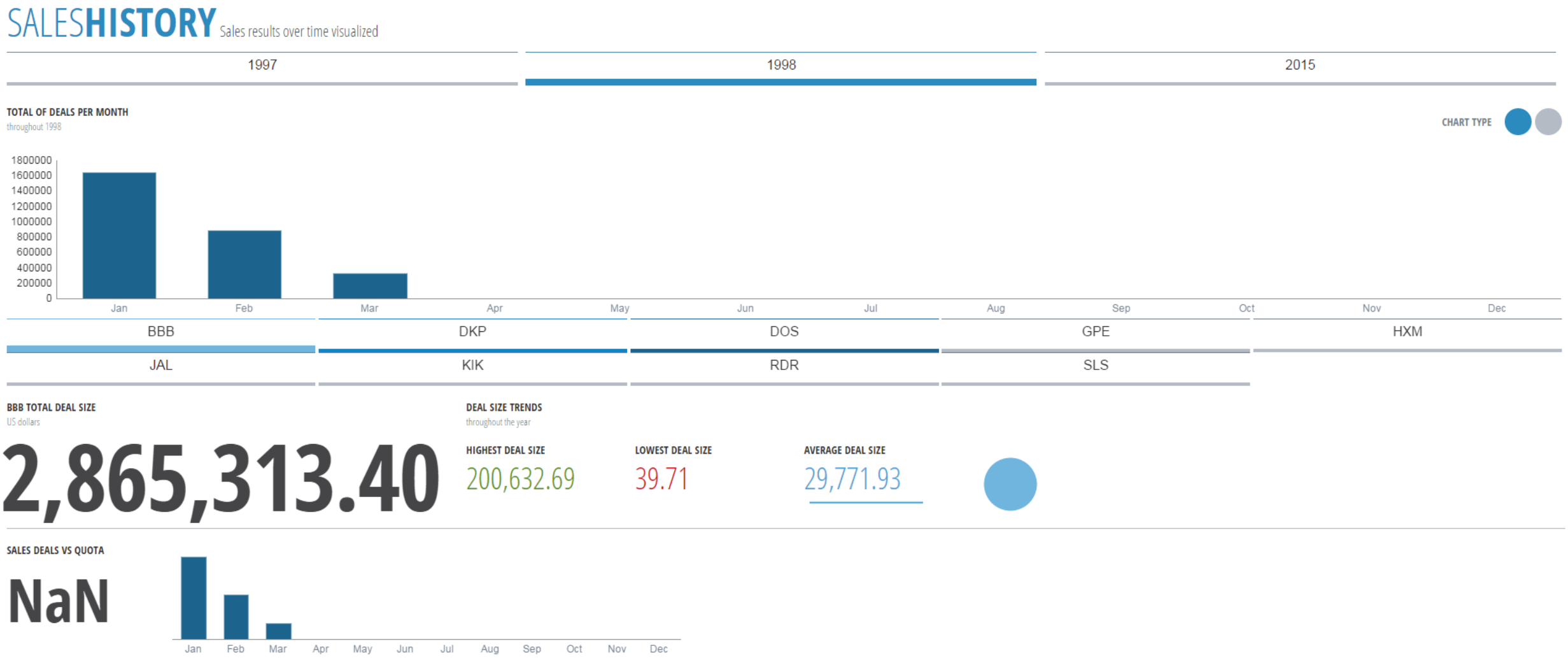
If you remember of Webspeed like this



What do you think about Webspeed like that ?



And this one ?



Advantages of PAS for OE

WebSpeed on PAS for OE

- supports standard HTTP verbs and status codes
- supports all media types for uploading and downloading
- no need to run a seperate webserver
- AppServer/WebSpeed/WebServer in one instance
- Access to everything in the http request message (i.e. Header, body and query parameters)

Classic WebSpeed

- only supports GET and POST, no way to control status codes
- supports a limited set of file types for uploading
- additional webserver required
- Seperate appserver and webspeed brokers
- Limited access to the request message

Why Upgrade to 11.6 for WebSpeed?

- Performance!
- Less Resource Requirements
- Better Security

PROGRESS

https://www.progress.com/openedge/whats-new

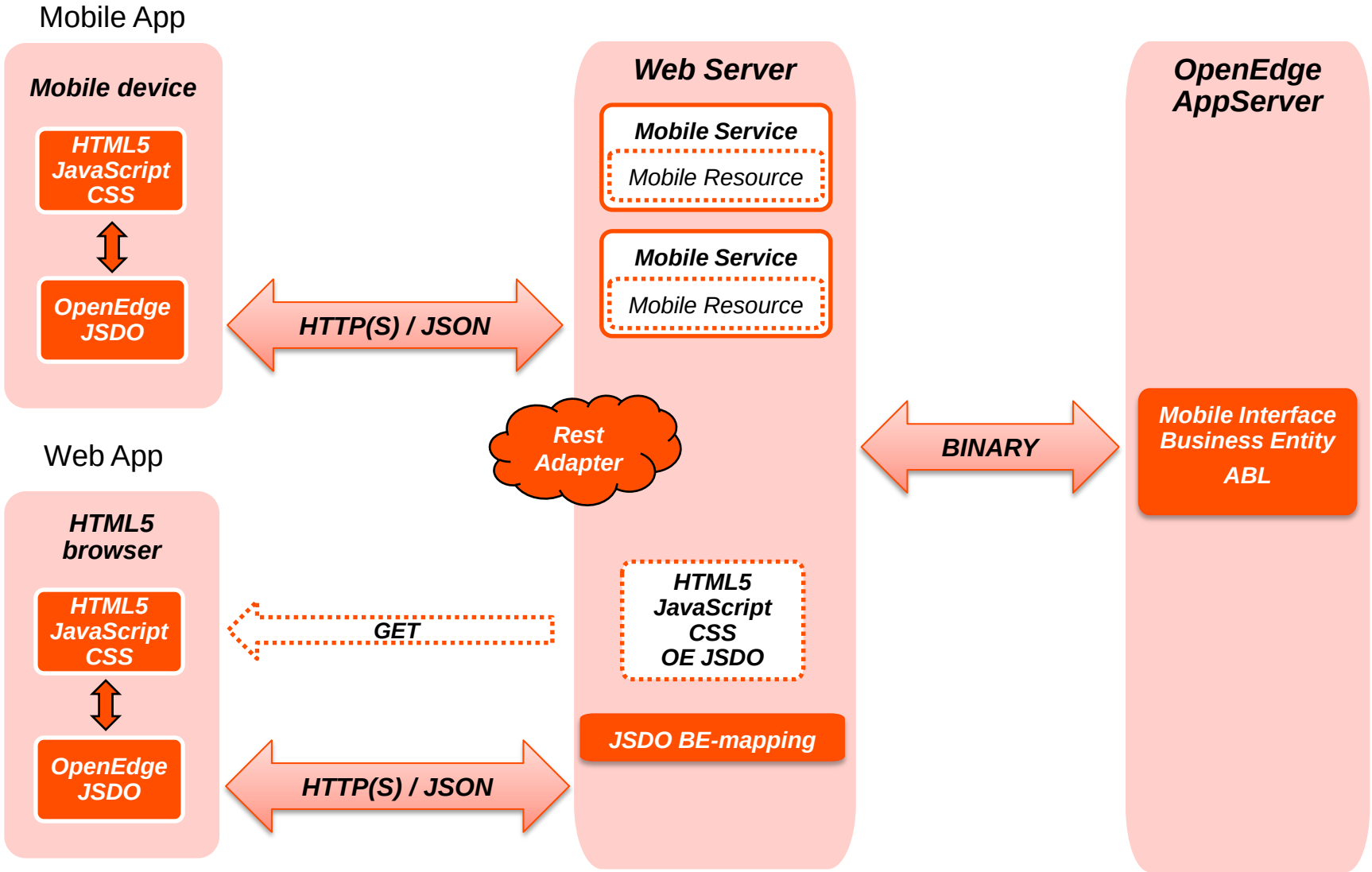
Modernisation Web avec Telerik KENDO UI

PROGRESS

Agenda

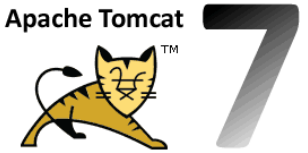
- **Classic AppServer**
  - mobile architecture
- **Pacific Application Server for OpenEdge (PAS for OE)**
  - what is PAS for OE?
  - mobile architecture
- **JavaScript Data Object (JSDO)**
  - what is the JSDO?
  - why should I use the JSDO?
  - Kendo DataSource and the JSDO
- **Demo**
- **Model View ViewModel (MVVM)**
- **Questions**

Architecture – Classic OpenEdge AppServer



Pacific Application Server for OpenEdge

- Enterprise class web/application server that integrates multiple functions
  - Web Server (Tomcat), OpenEdge AppServer, OpenEdge AppServer adapters

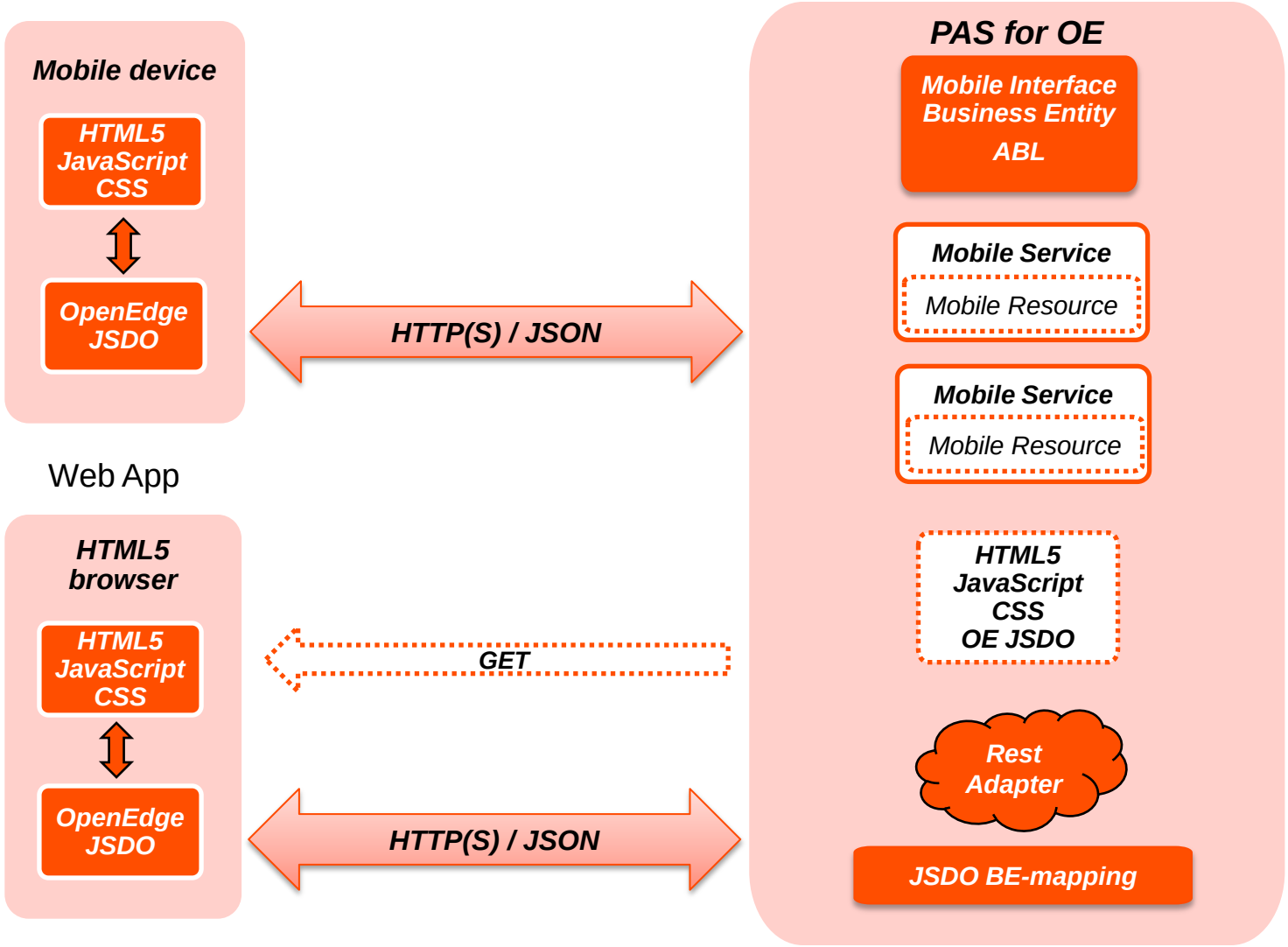


- Highly scalable
  - Multi-Session agent
  - Tomcat with industry standard load balancing

- Spring Security framework built-in

- No seperate AppServers required for different state-models
- Little to no ABL code changes when migrating

Architecture – Pacific Application Server for OE

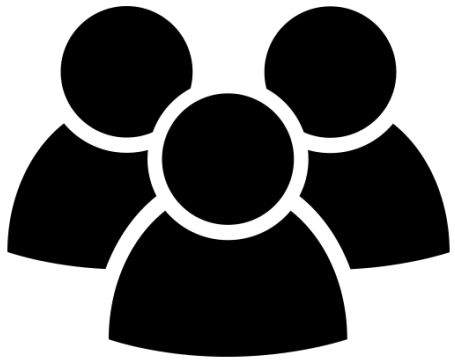


- Less overhead
- Reduced number of OS processes
- Better overall system performance
- One place to deploy

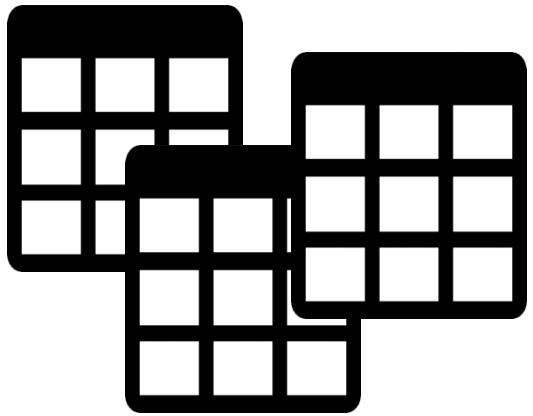
JSDO – JavaScript Data Object

- Client side object provides access to mobile resources exposed as a web service
  - Resources are ABL business logic and OpenEdge data
- Encapsulates functionality needed to manipulate client side data
- Communicates with an AppServer sending and receiving data
  - Can execute CRUD operations as defined in the Business Entity
  - Can invoke methods on an AppServer
- JSDO properties and methods
  - A JavaScript program or a JavaScript framework uses the JSDO methods to make requests of the JSDO

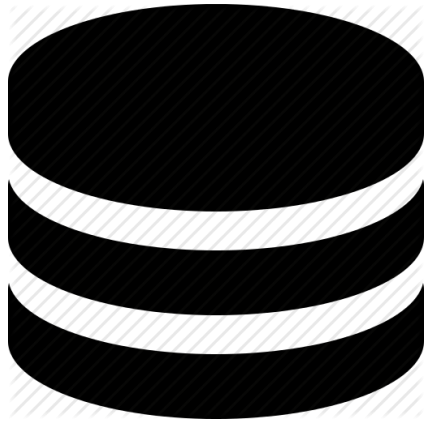
Why should I use the JSDO?



- Multi-user change tracking
- Session management
- Invoke methods on AS

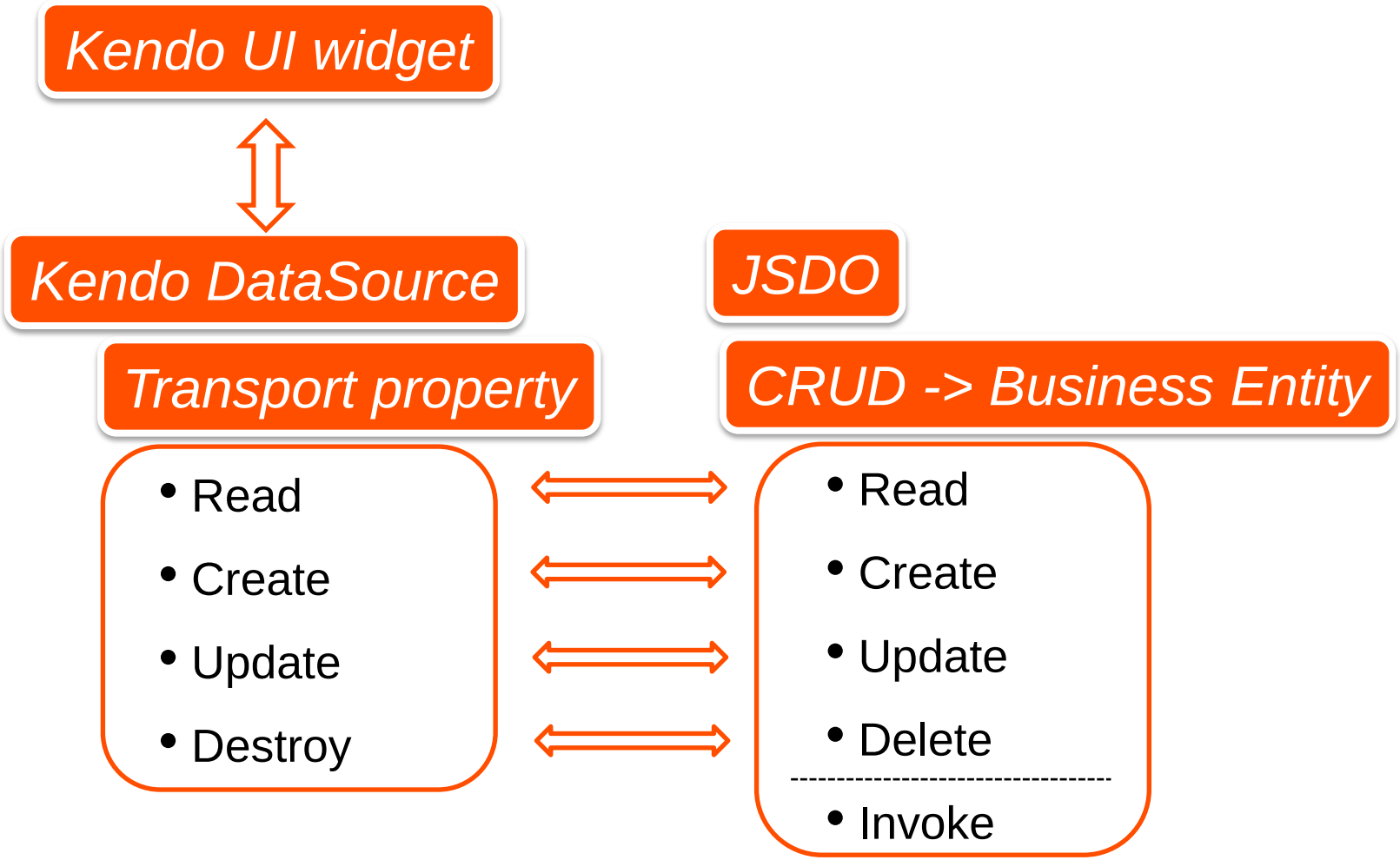


- Multi-table support
- Multiple CRUD actions
- Access to data definitions



- Local storage
- Any JavaScript framework
- No knowledge of URI’s

Kendo DataSource and the JSDO



Let’s Build!

Create PAS for OE instance (and deploy oemanager)

Create BE’s & business logic

Create and expose the ABL Service

Consume Mobile Service using the JSDO

Binding JSDO to Kendo UI controls (dataSource)

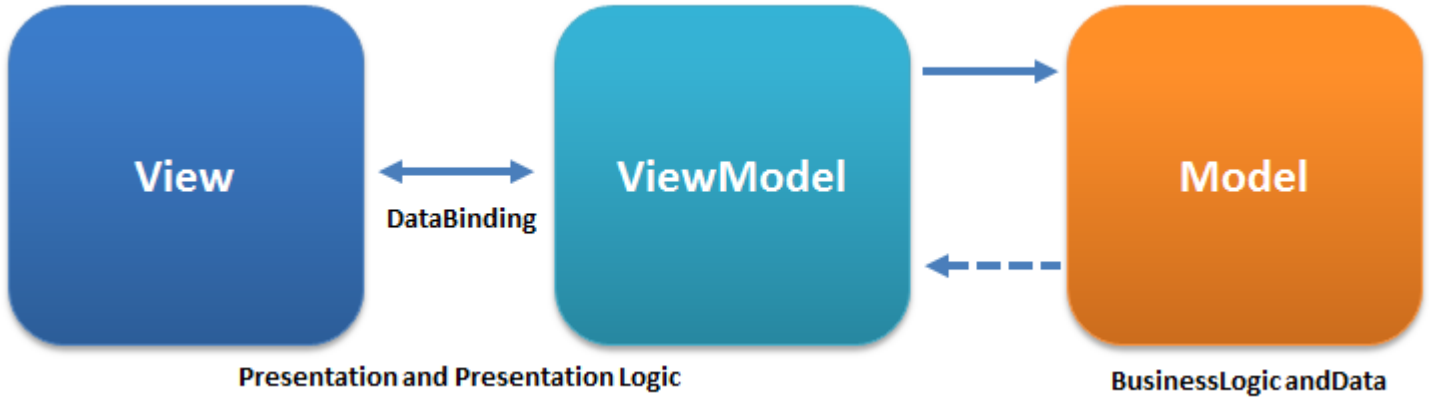
Recap...

Create PAS for OE instance	Create BE's & business logic	Create and expose the Mobile Service	Consume Mobile Service using the JSDO	Binding JSDO to Kendo UI widgets (DataSource)
1 command (tcman) to manage, configure and monitor	**Wizard to easily create Business Entities from a database table or include file**	**1 service that can contain multiple ProDataSets with multiple tables**	**Instantiate JSDO session** **Setup the CRUD operations once and reuse those**	**After binding JSDO to the Kendo DS, it can be used for any widget**

MVVM – seperating logic from UI

Model View ViewModel (MVVM) is a design pattern which helps developers separate the Model (the data) from the View (the UI).

The View-Model part of MVVM is responsible for exposing the data objects from the Model in such a way that those objects are easily consumed in the View.



Q & A