Progress

# OpenEdge Technologie Webinare

Webinare 2020

# Authentifizierung und Autorisierung mit PAS for OpenEdge

**Stefan Bolte**

Principal Sales Engineer

# Ihr Ansprechpartner


Stefan Bolte

stefan.bolte@progress.com

+49 221 65088070

Principal Sales Engineer

Progress Software GmbH

Köln

# OpenEdge Technologie Webinare in DACH

| Termin | Titel | Beschreibung |
|---|---|---|
| 21. April | OpenEdge 12.2 | Am 2. April 2020 wird OpenEdge 12.2 freigegeben. Ein Überblick über die Neuerungen, unter anderem zur Performance der ABL, der Datenbank und der DataServer und dem neuen Product Lifecycle. |
| 19. Mai | SQL und Co. | Der Zugriff auf Anwendungsdaten für Analyse und Reporting ist ein zentrales Thema. Wir möchten die Werkzeuge und Schnittstellen, die OpenEdge zur Verfügung stellt, im Überblick vorstellen. |
| 16. Juni | OpenEdge Schnittstellen für Web und Mobile Apps | Der Progress Application Server für OpenEdge bietet universelle Schnittstellen für den Zugriff von Webanwendungen und Apps auf die server-seitige OpenEdge Anwendung. Welche Optionen stehen Ihnen zur Verfügung? |
| 8. Juli | Continuous Integration und Continuous Delivery für OpenEdge | Für den Prozess der Software-Erstellung (Build Process) mit OpenEdge stehen Werkzeuge zur Automatisierung zur Verfügung. Das gilt auch für die Auslieferung von Updates in den Progress Application Server für OpenEdge. Diese Webinar gibt einen Einblick in die Möglichkeiten. |
| 15. September | Daten synchronisieren mit OpenEdge Change Data Capture und OpenEdge Pro2 | Teile der Daten einer OpenEdge Anwendung müssen oft in andere Anwendungen, zu anderen Standorten oder in zentrale Datenbanken redundant kopiert werden, und Änderungen danach synchronisiert werden. Mit den Replication-Triggern, dem OpenEdge Features Change Data Capture und OpenEdge Pro2 stehen mehrere Mittel bereit, um passende Lösungen zu erstellen. Wir stellen die Mittel vor. |
| **6. Oktober** | **Authentifizierung und Autorisierung mit PASOE** | **Der Progress Application Server für OpenEdge (PASOE) verwendet bekannte Technologien, um Authentifizierung und Autorisierung zu implementieren. Sie arbeiten auch gut mit externen Kompo-nenten wie LDAP Directory Servern und Secure Token Servern zusammen. Ein Einstieg ins Thema.** |

https://www.progress.com/campaigns/de/openedge/technologie-webinare-2020

Weitere Webinare: https://www.progress.com/campaigns/openedge/oe-emea-openedge-live-talks-webinar

# Agenda

- PAS for OpenEdge
- OpenEdge Identity Management
- Spring Security
- PASOE Authentifizierung und Autorisierung
- Externe Authentifizierungs-Anbieter

Progress®

# PAS for OpenEdge

# Progress Application Server

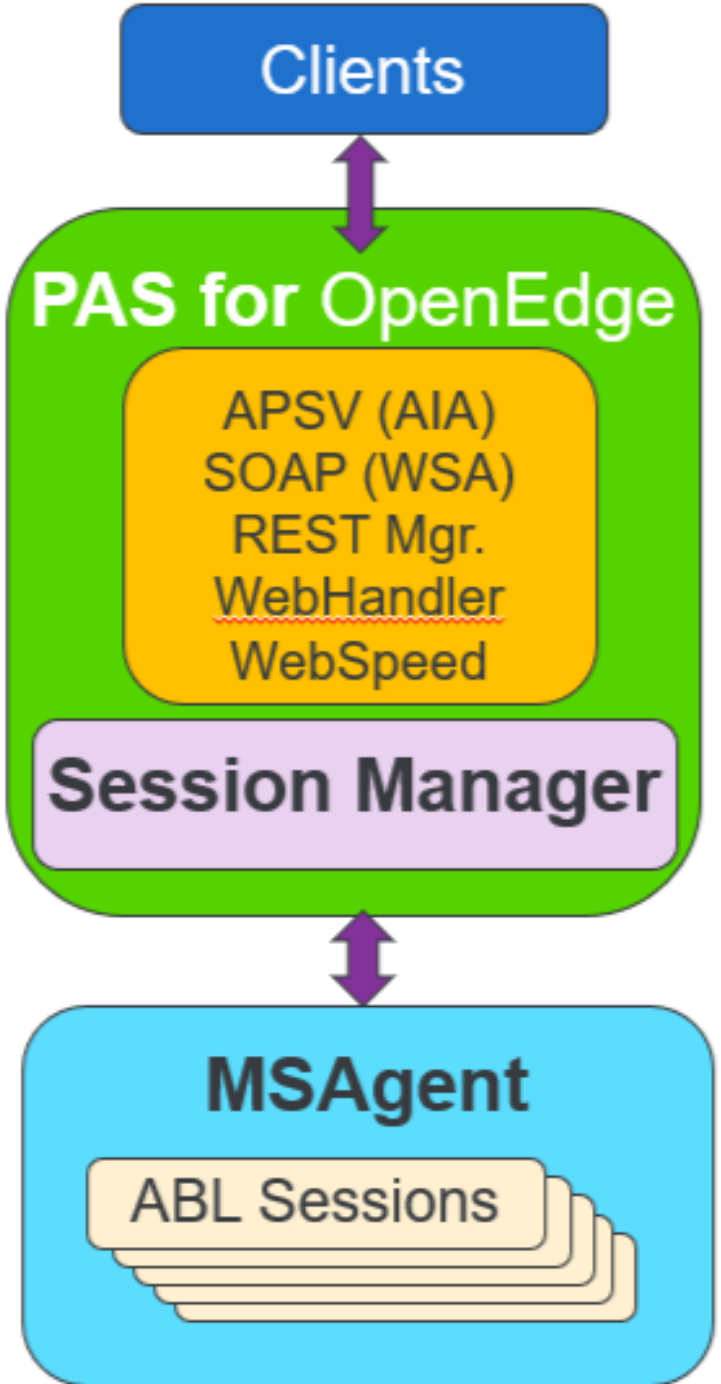
- PAS ist ein Tomcat-basierter Application-Server mit Progress-spezifischen Anpassungen und Erweiterungen
- Tomcat ist eine Komponente der J2EE-Plattform und bietet:
  - Catalina zum Ausführen von Servlets
  - Coyote http-Server
  - Jasper für Java Server Pages
  - Spring Security
- Bietet optionale Komponenten für die Fernverwaltung mit Text-, HTML- oder JMX-Proxy-APIs

# Warum ein Web Application Server?

- Die Anwendungsarchitektur muss aktuelle Entwicklungen unterstützen
  - Verschiedene UI-Technologien, einschließlich GUI, Web-Browser und Mobile
  - Anwendungen als Sammlung von Services (SOA), die über ein Service-Interface von Client aufgerufen werden.
  - Bedarf an einer standardbasierten, offenen und sicheren Schnittstellen wie REST und SOAP
  - Proprietäres OpenEdge GUI-AppServer Protokoll gekapselt in Standard-Protokoll
- Software as a Service ist der Trend
  - Entwicklung "Web- und Cloud-nativer" Anwendungen
  - Sicher und skalierbar
  - Einfaches Deployment und Betrieb in virtualisierten Umgebungen (VMs, Container)

# PAS for OpenEdge

- PASOE ist ein PAS mit integrierten Erweiterungen zur Ausführung von ABL-Code in ABL-Multi-Sessioned-Agents.
- Jeder ABL-Agent verarbeitet viele Sessions gleichzeitig (Multi-Threaded)
- Sie entwerfen, paketieren, implementieren, konfigurieren, debuggen und kontrollieren den Zugriff auf Ihre ABL-Anwendung im Kontext einer Tomcat-Webanwendung.



# Warum eine neue AppServer Generation?

- Einfachheit
  - Verwaltung, Skalierbarkeit, Anwendungsmigration, Bereitstellung
  - AppServer-Verbindung und Betriebs-STATEs
- Eingebaute, standard-basierte Sicherheitsinfrastruktur
  - Spring Security eingebettet
  - Passt zu einer Vielzahl von Zugriffsverwaltungen
- Bessere Überwachung und Verwaltung
  - Integrierte Erfassung von Metriken, Abfragen zum aktuellen Status
- Höhere Leistung und Skalierbarkeit
  - Reduzierter Bedarf an RAM und CPU Leistung im Vergleich zum Classic AppServer

Progress®

# OpenEdge Identity Management

# OpenEdge Identity Management

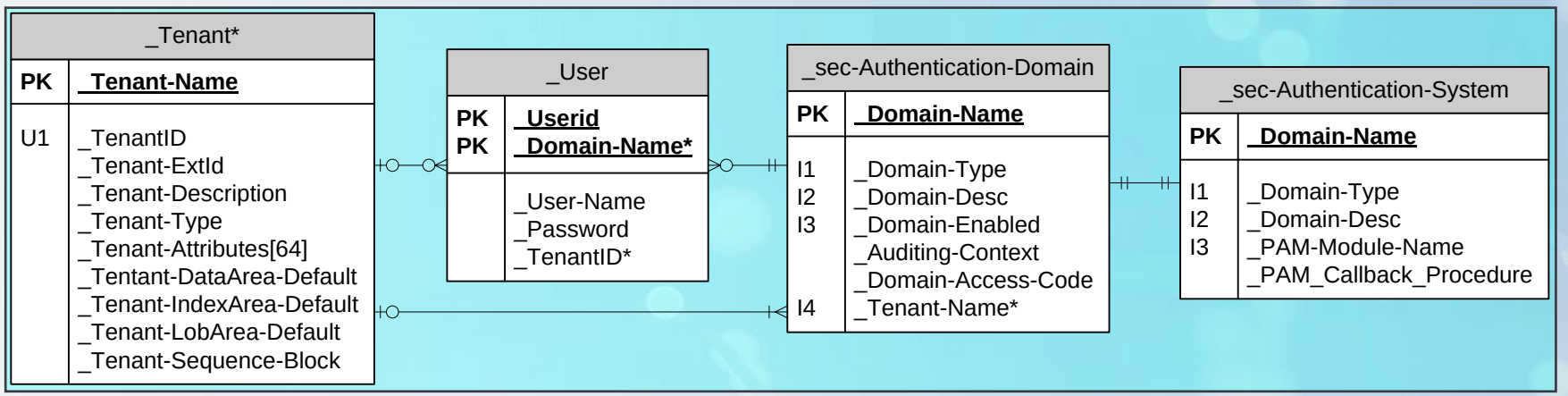
Historie der Nutzung des Identity Managements

- OpenEdge 10.1A: Auditing und Single-Sign-On
- OpenEdge 11.0: Multi-Tenancy und Domains
- OpenEdge 11.1: pluggable Authentication ABL Callback
- OpenEdge 11.2: OE REST Manager mit Spring Security
- OpenEdge 11.5: PASOE mit Spring Security

https://docs.progress.com/bundle/openedge-security-identity-management/page/What-is-Identity-Management.html

# Erweitertes User Management

- Security-Domänen (und Tenants) Teil des DB Meta-Schemas
- Eine Security-Domäne ist ein benannter Satz von Richtlinien
  - Legt fest, wie die Identität der User überprüft werden
  - _oeuser table: User Verwaltung in einer OpenEdge Datenbank
  - _oslocal: Identity wird vom Betriebssystem verwaltet
  - _extsso: Externe System wie LDAP, Active Directory, etc.
  - Und User defined authentication system: Eigener ABL code



# OpenEdge Identity Management

## Client-Principal

- Vertrauenswürdiger Berechtigungs-schlüssel mit Identität und Rolle(n)
- Mit Checksumme gesichertes Identitäts-Token
- In der ABL ein handle-basiertes Objekt
- Eingeführt mit OE-Auditing in OE 10.1A

**CLIENT-PRINCIPAL**

**Domain:** Application
**User-ID:** Joshua
**Tenant-Name:** Acme
**Tenant-ID:** 203
**Login-token:** BW3G1&2G1836D872
**Login-date:** 6/12/11 08:15:33.12
**Login-expires:** 6/12/11 19:30.00.00
**State:** Login
**Roles:** Manager
**App-data:** Company=ABC Corp
...
**Seal:** AC63Galx98wBwuuw2

# Erzeugen des Client-Principals

- Wird meist implizit erstellt. Möglichkeiten:
  - beim Anmelden eines Users bei einer Domäne (definiert in einer OpenEdge Datenbank).
  - Spring Security Beans im PASOE, je nach Authentication Provider
  - OpenEdge Authentication Gateway
- Kann auch manuell erzeugt werden: eigene Authentifizierung.

```
DEFINE VARIABLE hCP as HANDLE.

SECURITY-POLICY:LOAD-DOMAINS(domain-db-name).

CREATE CLIENT-PRINCIPAL hCP.

hCP:INITIALIZE(qualifiedUserid, PassPhrase).  /* userid@domain */

SECURITY-POLICY:SET-CLIENT(hCP). /* authenticate, login and seal */
```

# Nutzen des Client Principals

- Identity Token wird zwischen Client, AppServer und Datenbank weitergegeben
- Definieren der Identität eines Session-Free Aufrufs
- Serialisieren (Export-Principal () Methode) eines Client Principals
  - für das Speichern Wiederherstellen eines Session-Kontextes mit Identität

```
SECURITY-POLICY:SET-CLIENT(hCP). /* aktuelle Login-Session */

IsValidIdentity = SET-DB-CLIENT(hCP). /* abweichende DB-Identity */

IsValidUser = SETUSERID("admin","admin"). /* nicht mehr verwenden! */
```

Progress®

# Spring Security

# Spring Security

- Spring Security ist ein leistungs-starkes und anpassbares Framework für Authentifizierung und Zugriffskontrolle
- Eingebaut in PASOE
  - Spring wird automatisch gestartet, wenn Sie eine Instanz starten.
- Alle Requests müssen den Spring-Sicherheitsprozess durchlaufen, um ein Sicherheits-Token zu generieren

# Spring Security Process

- Spring übernimmt die Zugangsdaten aus dem Request
- Eine Reihe von Java Beans implementieren den Prüfprozess
- Am Ende des Prozesses wird ein `Authentication` Object (Java) erstellt und befüllt, dass den Servlets zur Verfügung steht.
- PASOE hat erweiterte Beans für die Integration mit OpenEdge und dem OpenEdge Authentication Gateway

| |
|---|
| **Browser submits** "*authentication credentials*" |
| "*Authentication mechanism*" **collects the details** |
| **An** "*authentication request*" **object is built** |
| *Authentication request* **sent to an AuthenticationManager** |
| *AuthenticationManager* (this is responsible for passing requests through a chain of *AuthenticationProviders')* |
| "*Authentication provider*" will ask a `UserDetailsService` to provide a `UserDetails` object |
| The resultant `UserDetails` object (which also contains the `GrantedAuthority[]s`) will be used to build the fully populated `Authentication` object. |
| If "*Authentication mechanism*" **receives** back the fully populated `Authentication` object, it will deem the request valid, put the `Authentication` into the `SecurityContextHolder`; and cause the original request to be retried.<br>If, on the other hand, the `AuthenticationProvider` rejected the request, the authentication mechanism will ask the user agent to retry. |
| `AbstractSecurityInterceptor` authorizes the regenerated request and throws Java exceptions. (Asks `AccessDecisionManager` for decision.) |
| `ExceptionTranslationFilter` translates the exceptions thrown by `AbstractSecurityInterceptor` into HTTP related error codes |
| Error code 403 – if the principal has been authenticated and therefore simply lacks sufficient access<br>Launch an `AuthenticationEntryPoint` – if the principal has not been authenticated which is an authentication mechanism |

https://en.wikipedia.org/wiki/Spring_Security

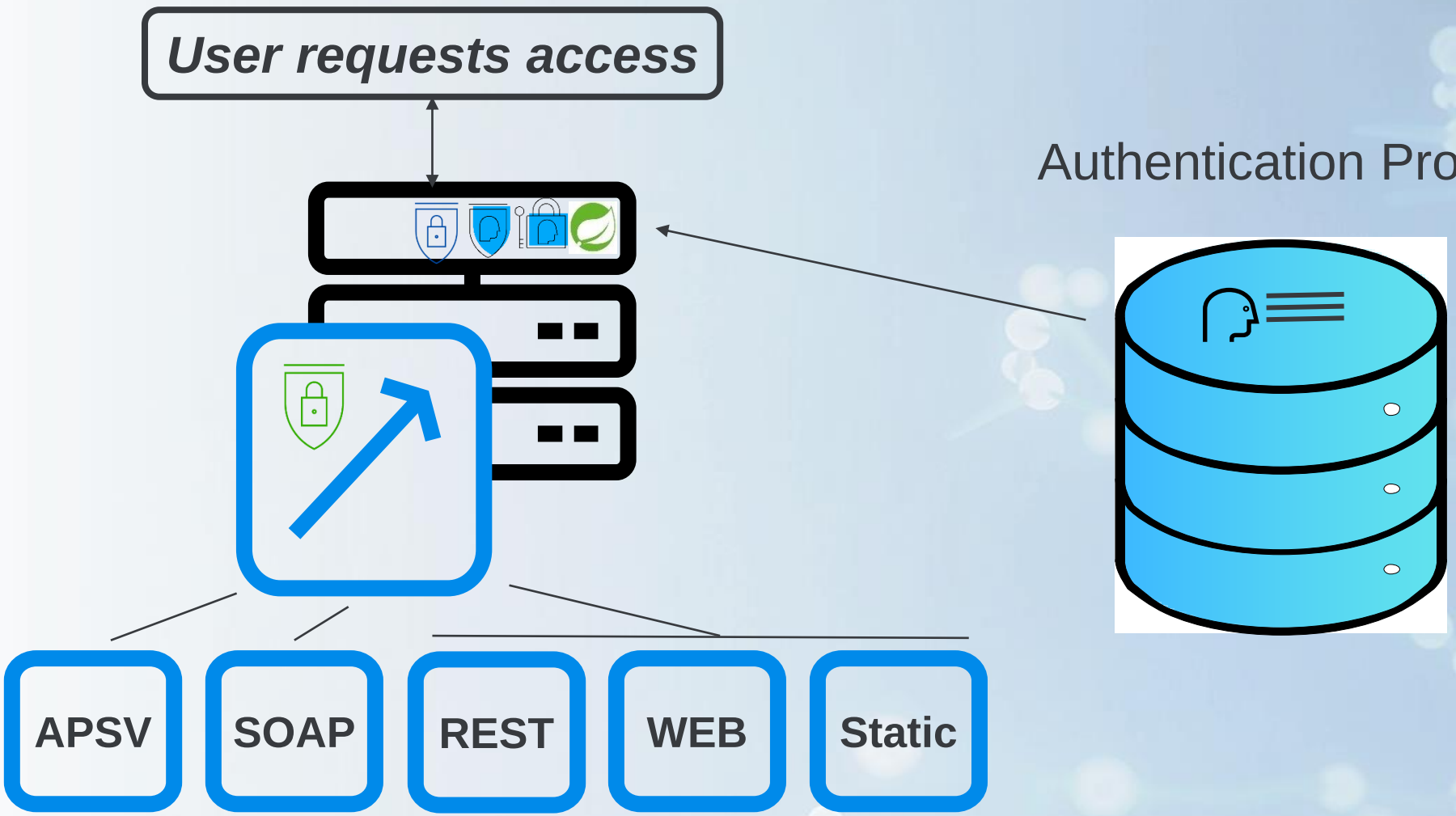
# Authentication Object, ein Security Token

- Ein Objekt, das sowohl Benutzeranmeldeinformationen als auch zusätzliche Informationen über die Rollen und Berechtigungen des Benutzers enthält
- PASOE erzeugt ein entsprechendes Client-Principal Object
  - Bietet Flexibilität - ich entscheide, welcher Provider verwendet wird
    - Anonym oder Textdatei, LDAP, Active Directory, OERealm, OpenEdge Authentication Gateway usw.
  - Versiegelt – Kann nicht durch eine andere Person genutzt werden
  - Zeitlich beschränkt - beschränkt das Risiko, dass andere dieses Token verwenden

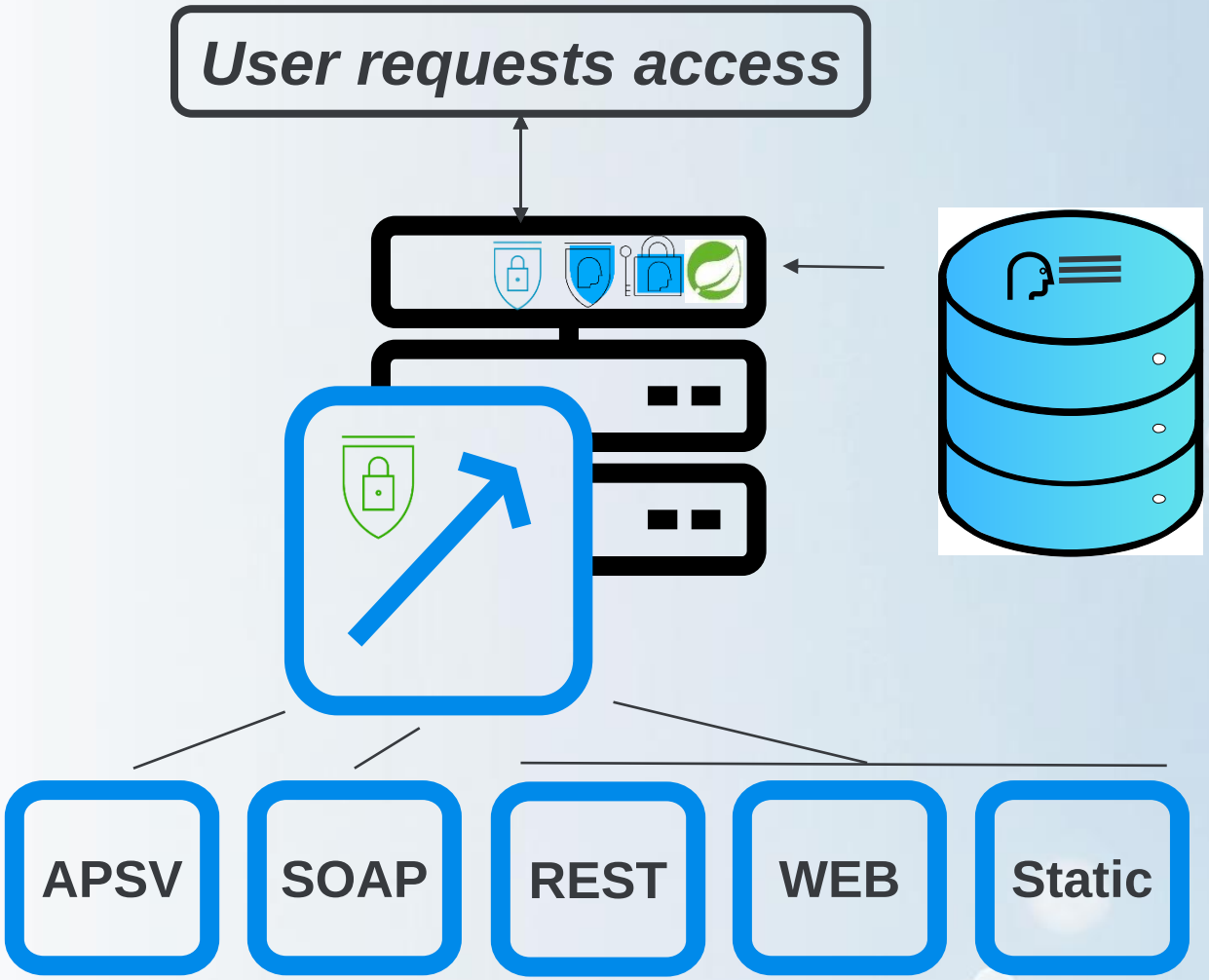
Progress®

# PASOE Authentifizierung und Autorisierung

# Unterstützte Request Protokolle



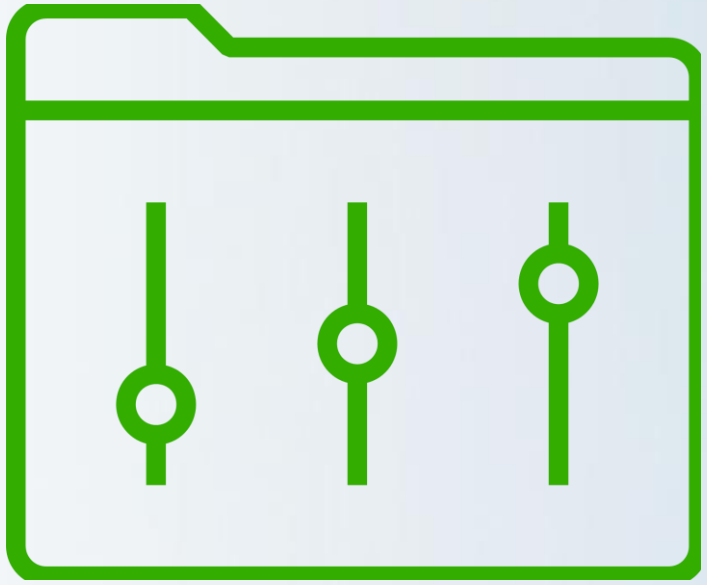
# Unterstützte Authentication Provider




**Authentication Provider**

- (None)
- Local Security (user.properties) und ExtLocal (verschlüss. Passwörter)
- OERealm
- OE Single-Sign-On
- STS (OpenEdge Authentication Gateway)
- LDAP und Active Directory
- SAML2 und OAuth2

# Vergleich PASOE 11.5/.6 und danach



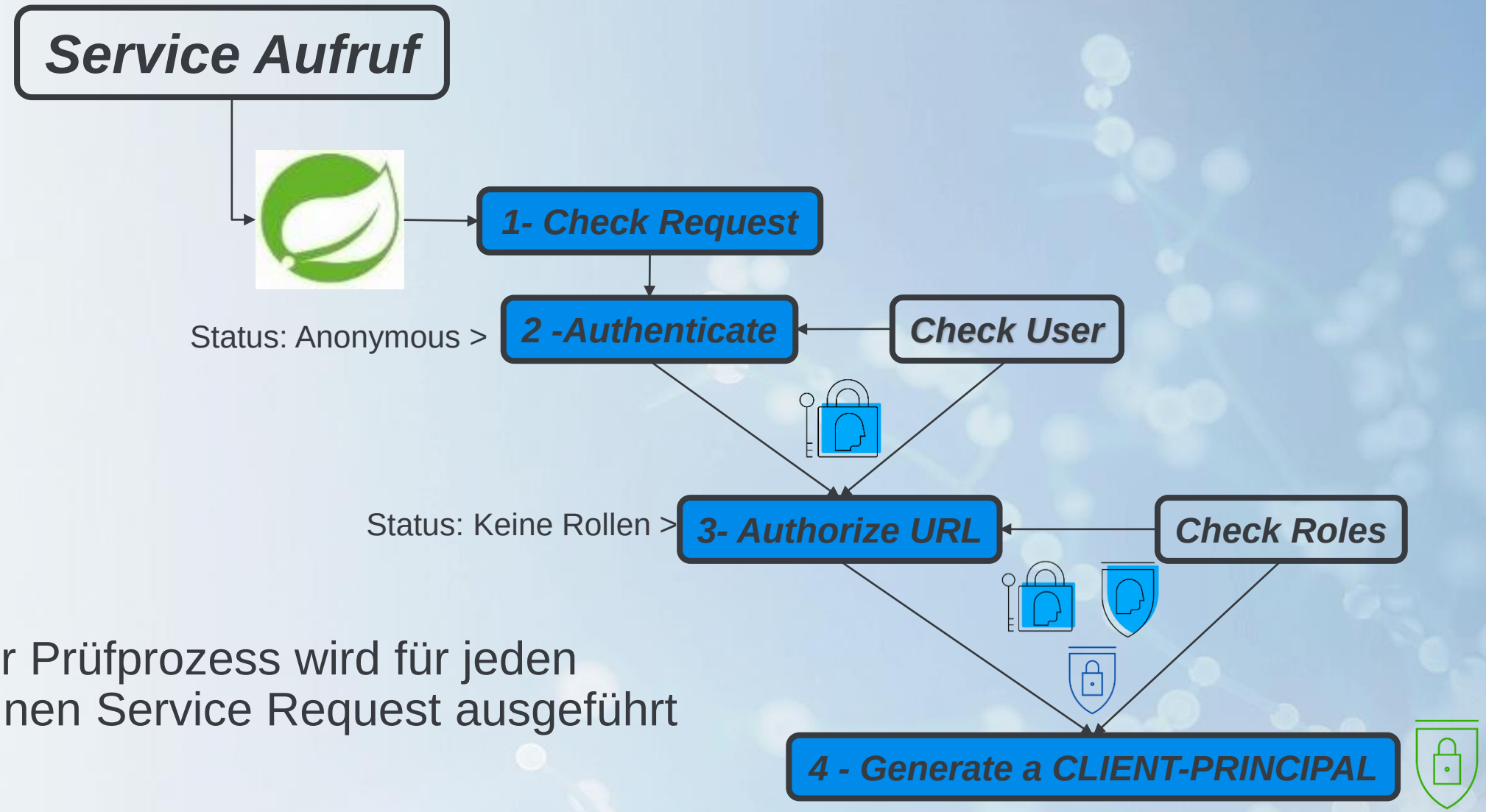
- **Configuration files**
  - 11.6 nutzt .xml files
  - 11.7.x verwendet zwei Dateien
    - `oeablSecurity.properties`
    - `oeablSecurity.csv`
  - Lesson learned
    - xml Struktur Pflege war fehleranfällig
    - Properties und .csv file sind einfacher zu
      - editieren
      - debuggen
      - migrieren

# Development vs Production modes

- Modus Development
  - Hat eine Standardinstanz, oepas1
  - Alle Protokolle sind aktiv
  - Kann sowohl ABL Source-Code als auch r-Code ausführen
  - Hat einen MS-Agent mit maximal fünf gleichzeitigen Sessions
  - Sicherheitsprüfung ist deaktiviert
- tcman create … -Z dev

- Modus Production
  - Hat keine Standardinstanz
  - Kann nur kompilierten Code ausführen
  - Die Sicherheit ist so eingestellt, dass jeder Zugriff verweigert wird
  - Alle Protokolle sind deaktiviert
  - Keine Beschränkungen hinsichtlich der Anzahl der Agents oder gleichzeitiger Sessions
- tcman create … -Z prod

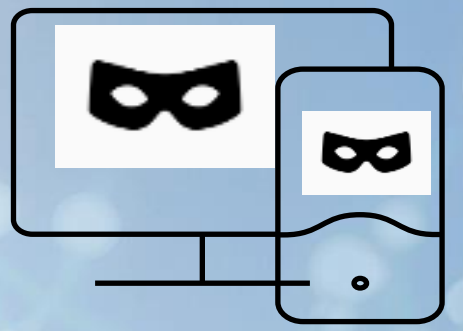
# Authentifizierung und Autorisierung in PASOE




Dieser Prüfprozess wird für jeden einzelnen Service Request ausgeführt

# Schritt 1: Gültigkeit des Request prüfen

- Spring wendet Standard Filter an
  - HTTPS Filter
  - HTTPS [SSL/TLS] Client login Filters
  - CORS (Cross-origin resource sharing) Filter
  - CSRF (Cross site request forgery) Filter
- Kurz: Ist die Anfrage konform und zuverlässig?
  - Es sind zusätzliche Schritte erforderlich, um TLS, CORS usw. zu aktivieren. Diese Prüfungen gehen über das Thema Spring Security hinaus.



Externe Bedrohungen

# Schritt 2: Authentifizieren

- Spring prüft Ihre Konfigurations-Dateien, um zu entscheiden
  - Welche Art von Authentifizierung?
  - Direkte Anmeldung (Benutzername & Passwort)
    - Wo finde ich den Authentifizierungs-Provider?
  - Single Sign On (SSO)
    - Ist bereits authentifiziert, muss aber verifiziert werden

Authentifizierung

- Kurz: Überprüfe ich die Anmeldedaten der Benutzer? Wenn ja, wo finde ich die richtige Liste, um zu überprüfen, ob die Benutzer die sind, für die sie sich ausgeben

# Schritt 3. Autorisieren

- Spring Security erteilt Rechte auf der Basis von Rollen
- Basierend auf Ihrer Konfiguration überprüft Spring Folgendes:
  - Beschränken wir den Zugriff auf verschiedene Arten von Anwendungen (REST, Web usw.)?
  - Beschränken wir den Zugriff auf Teile der Anwendung (Buchhaltung, Finanzen, Humankapital)
- Kurz: Schränke ich den Zugriff auf meine Anwendung basierend auf der Rolle des Benutzers in der Organisation ein?

Autorisierung

# Schritt 4. CLIENT-PRINCIPAL erzeugen

- OpenEdge nutzt ein CLIENT-PRINZIPAL
  - Liefert Domänen und Rollen
- Spring erzeugt ein Authentication Object
- PASOE transformiert mittels der von Progress eingebauten Spring Security Beans (OEClientPrincipalFilter) das Spring Authentication Object in ein korrespondierendes, versiegelts OpenEdge Client-Principal
  - Die Attribute werden übertragen
  - In der Agent Session wird das Client-Principal in die Security-Policy geladen.

CLIENT-PRINCIPAL

- Kurz: Wir verwandeln das Spring-Token in ein CLIENT-PRINCIPAL, um ABL-spezifische Details hinzuzufügen.

# PASOE Security Konfiguration

- Die meisten Parameter finden sich in .property files
  - oeablSecurity.properties
  - oeablSecurity.csv (URL zu Spring Access Pattern Zuordnung)
  - users.properties
- Hierarchie: Files im ./conf Verzeichnis von
  - PASOE Home
  - Instance Base
  - ABL Application
  - Web Application, dort in ./WEB-INF

https://knowledgebase.progress.com/articles/Knowledge/How-does-the-spring-security-hierarchy-work-with-PASOE

# Zugriffskontrolle (Access Control)

- Bei der Authentifizierung werden einem Benutzer eine oder mehrere Rollen zugewiesen
- Man kann einen Standard angeben (Default)
- PASOE (Spring) gewährt/verweigert Rollen den Zugriff auf Kombinationen aus URL + HTTP-Methode
- URL-Muster in oeablSecurity.csv
- Requests werden mit den Mustern verglichen, und Zugriff auf URL-Pfade (statische Dateien, Service Aufrufe) werden entsprechend gewährt.

# Spring-Access-Expressions

- https://docs.spring.io/spring-security/site/docs/3.0.x/reference/el-access.html

| Expression | Description |
|---|---|
| `hasRole([role])` | Returns `true` if the current principal has the specified role. |
| `hasAnyRole([role1,role2])` | Returns `true` if the current principal has any of the supplied roles (given as a comma-separated list of strings) |
| `principal` | Allows direct access to the principal object representing the current user |
| `authentication` | Allows direct access to the current `Authentication` object obtained from the `SecurityContext` |
| `permitAll` | Always evaluates to `true` |
| `denyAll` | Always evaluates to `false` |
| `isAnonymous()` | Returns `true` if the current principal is an anonymous user |
| `isRememberMe()` | Returns `true` if the current principal is a remember-me user |
| `isAuthenticated()` | Returns `true` if the user is not anonymous |
| `isFullyAuthenticated()` | Returns `true` if the user is not an anonymous or a remember-me user |

# Beispiel: Modus „Local“ für Tests

- Konfiguriere das Security Model in oeablSecurity.properties
  - http.all.authmanager =local
  - client.login.model =form or basic
- User in <instance>\webapps\<app>\WEB-INF\users.properties
  - <username>=<password>,<list of roles>, <enabled/disabled>
  - stefan=123456,ROLE_PSCUser,enabled
- Default users in Development Mode:
  - <instance>\conf\tomcat-users.xml
- PASOE Instanz neu starten

# User Session mit Nicht-ABL Clients

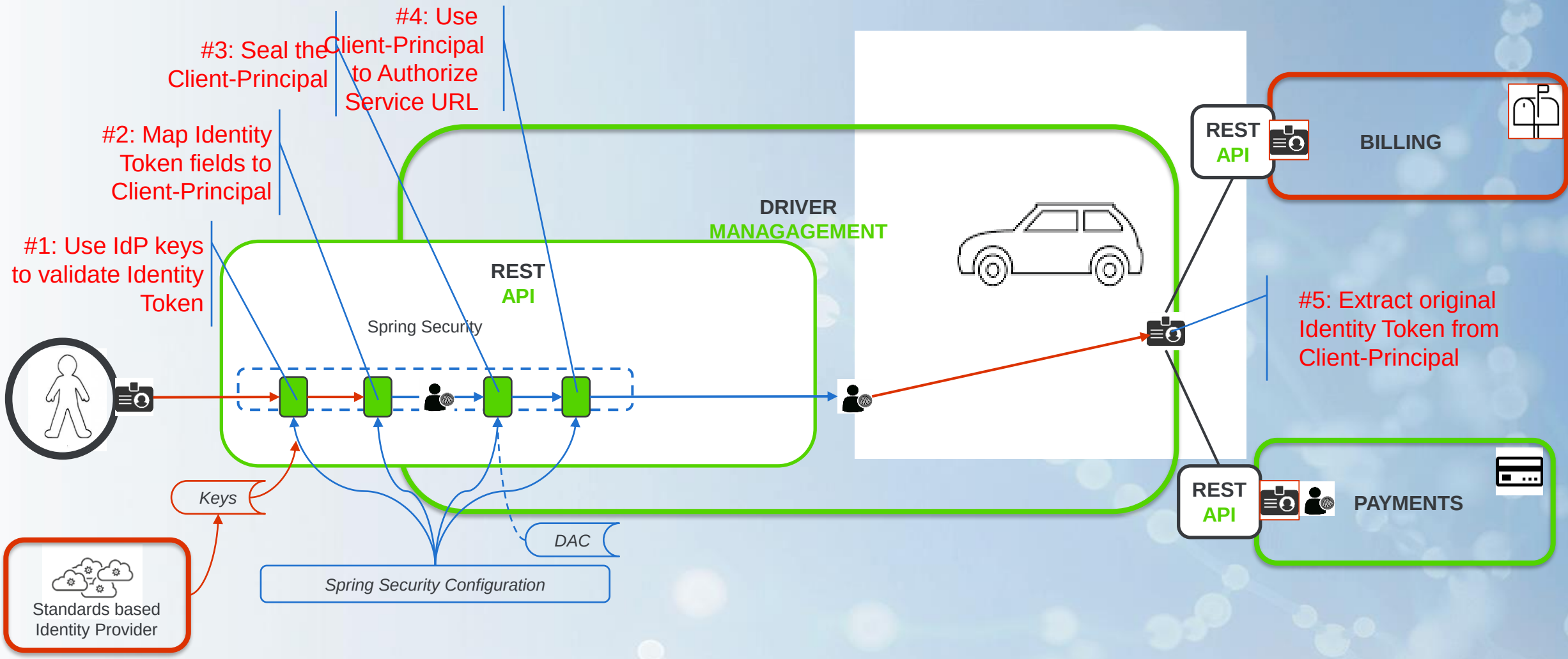
- Die PASOE Spring Security Implementation dehnt die OpenEdge SSO Funktionalität auf http(s)-Clients (Browser, REST, WebHandlers) aus.
- Das PASOE `OEClientPrincipalFilter` Bean erzeugt ein SSO Access Token. Es ist ein Base64-kodiertes und versiegeltes CLIENT-PRINCIPAL.
- Ein optionales Refresh-Token ist eine eindeutige UID (String-Token), die einem CLIENT-PRINCIPAL-Token zugeordnet ist.
- Voraussetzung ist HTTP FORM Authentication. Das SSO Access Token wird in http-Headern transportiert. Die Zuordnung zur Session wird vom Session Manager übernommen.
- Zugriff auf das Client Principal in der ABL Session wie immer: `SESSION:CURRENT-REQUEST-INFO:GETCLIENTPRINCIPAL().`

https://docs.progress.com/bundle/pas-for-openedge-management/page/Extend-OpenEdge-SSO-to-web-applications.html

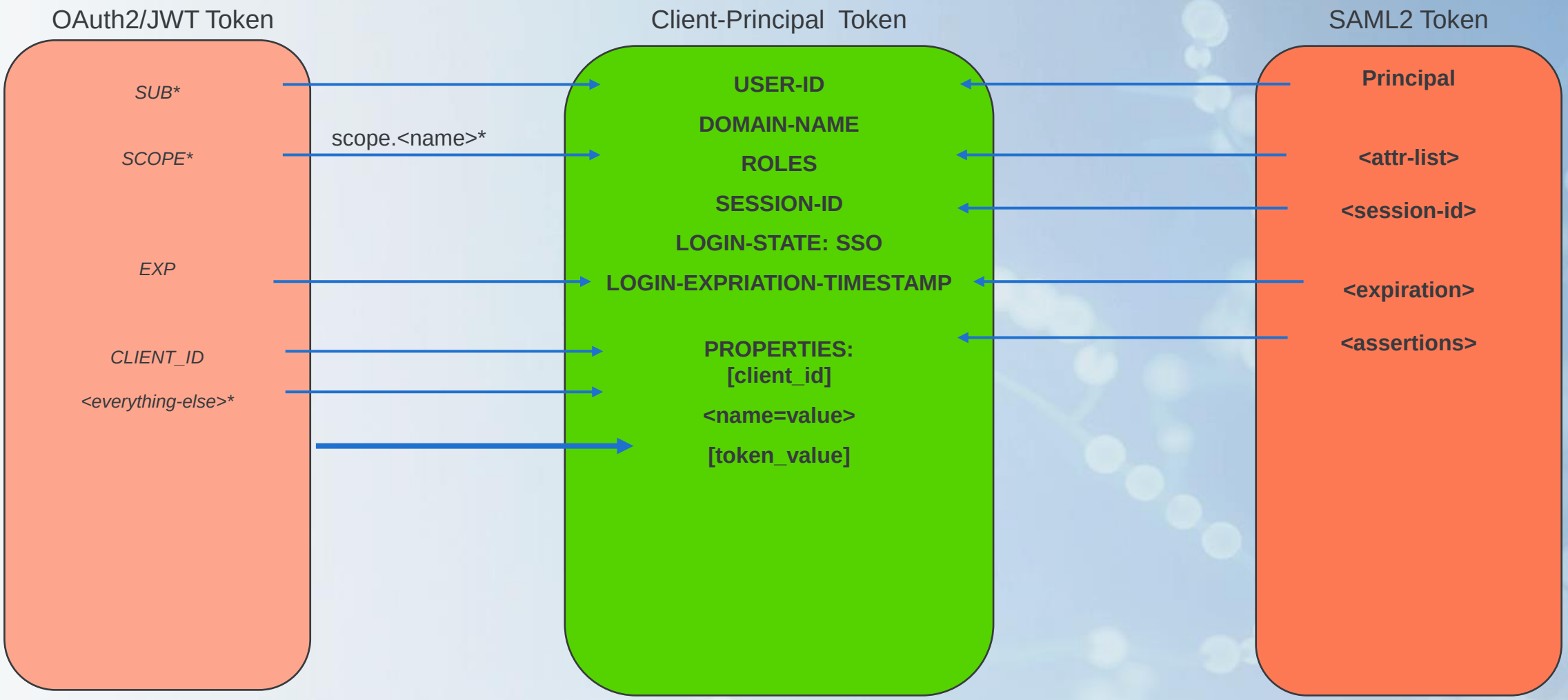
Progress®

# Externe Authentifizierungs-Anbieter

# PASOE mit externen Token (JWT/OAuth2/SAML2)

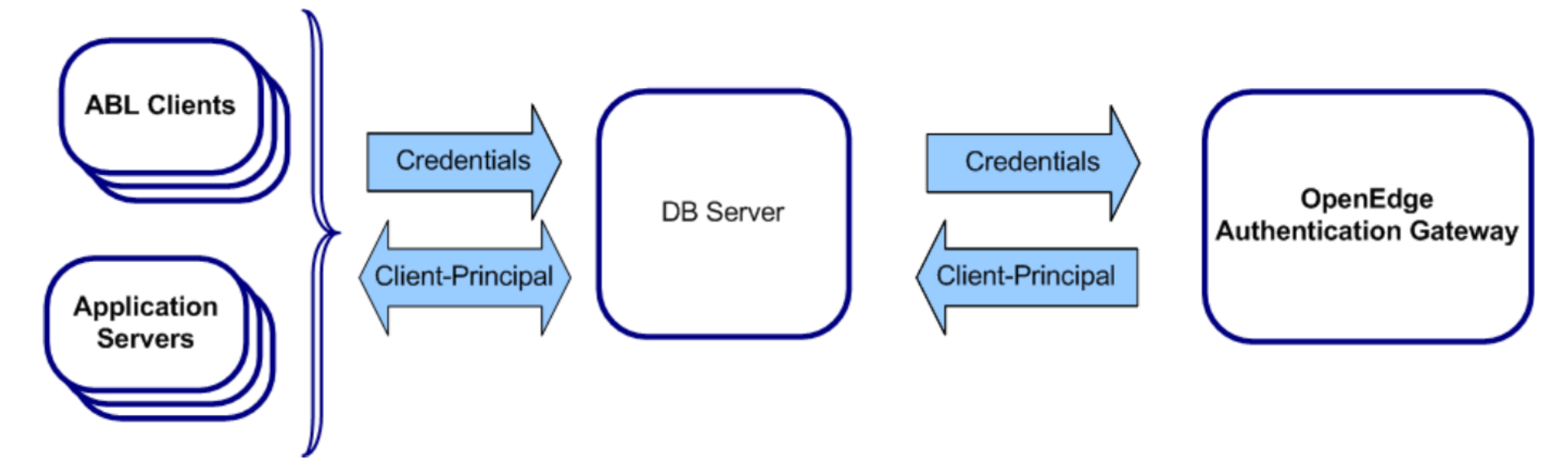


# JWT/OAuth2/SAML2 zu CP Token Mapping



# OpenEdge Authentication Gateway

- Zentralisiert die Authentifizierungs- und Autorisierungsprozess zwischen einer OpenEdge-Datenbank, ihren Clients und einem OpenEdge Authentication Gateway-Server.



# Aufgaben

- Bereitstellen eines sicheren Authentifizierungs-Gateways, das die gesamte Benutzeranmeldung und SSO an externe Benutzerverwaltungen sowie die Generierung/Validierung von Client-Principal-Sicherheits-Token kapselt
- Alle ABL-Clients und Datenbank-Server, die im "Secure-Client"-Modus laufen, verwenden ausschließlich das Authentication Gateway.
- Verschlüsseln/Verschlüsseln aller Übertragungen von User-id/pwd & Client-Principal-Sicherheitstoken-Daten auf unverschlüsselten Netzwerkkanälen
- Aktualisieren der Audit-Ereignisse mit zusätzlichen Informationen

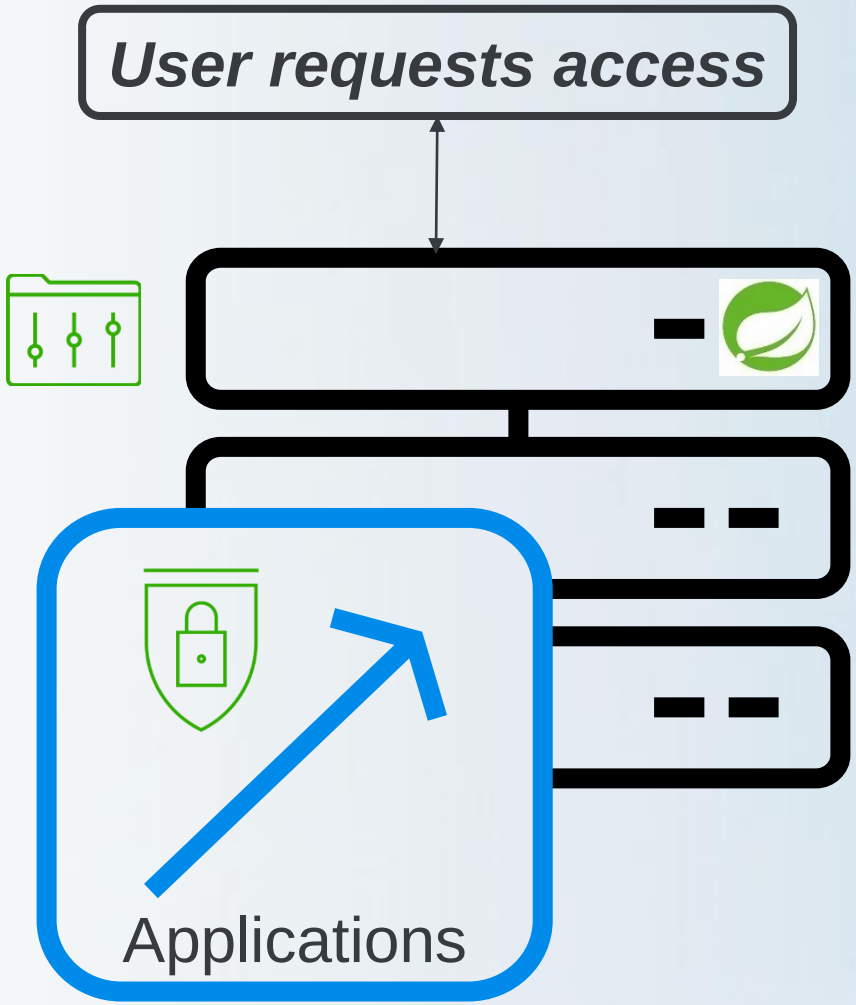
# OE Authentication Gateway (OEAG) mit PASOE

- Benutzerauthentifizierung zentralisiert in einem neuen OpenEdge Authentication Gateway
  - Unterstützt autorisierte Client-Verbindungen zur DB-Betriebsart "STS-Authentifizierung".
  - Wenn die Anwendung bereits das Client Principal verwendet, werden keine Code-Änderungen zur Verwendung des Authentication Gateway benötigt.
- Reicht die Authentifizierung an LDAP, Active Directory, Keon, u. a. weiter
- Client-Kommunikation mit dem Authentication Gateway verschlüsselt sensible Daten (z.B. Benutzername & Passwort; z.B. Client Principal)
- Zusätzliche Client-Sicherheit:
  - Datenbankserver blockieren ältere Clients, die nicht in der Lage sind, Authentication Gateway-Operationen zu unterstützen, und validieren vom OEAG ausgestellte Client-Principals, bevor sie den Zugriff erlauben.
  - Nicht berechtigte ABL-Clients werden an der Nutzung des Authentifizierungs-Gateways durch STS-Zugangsschlüssel gehindert
  - Manuell generierte Client-Principals durch ABL-Code-Module erfordern die Domain-Zugangscodes der DB: alle anderen werden blockiert

Progress®

# Zusammenfassung

# PASOE und Spring Security

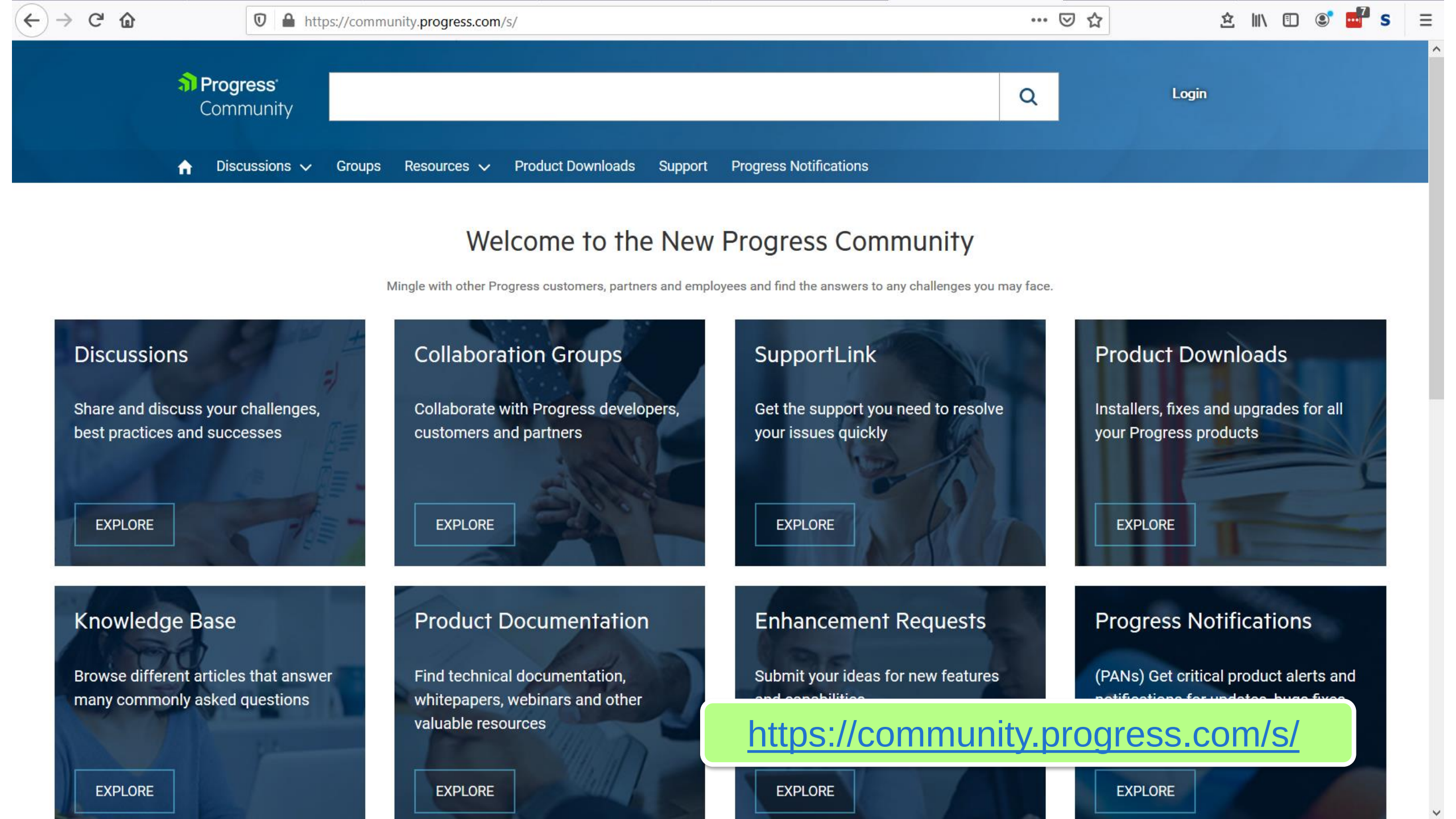



- Erste Verteidigungslinie für die Zugangskontrolle
  - Java-Industriestandard
  - Immer eingeschaltet, läuft immer zuerst
  - Ist unabhängig von ihrer Anwendung
  - Blockiert die Anfrage, wenn Authentifizierung und Autorisierung fehlschlagen
  - Erstellt immer ein Sicherheits-Token
- Einfache Verwendung mit vorhandenem Code
  - Generiert automatisch ein CLIENT-PRINCIPAL aus dem Spring-Sicherheitstoken, damit Sie es in Ihren vorhandenen Anwendungen verwenden können
- Leicht zu konfigurieren
  - Verwendung von .property- und .csv-Dateien
    - OpenEdge erledigt 99% der Details
  - Kann den Authentifizierungsanbieter leicht wechseln

https://docs.progress.com/bundle/pas-for-openedge-management/page/Secure-PAS-for-OpenEdge-instances.html

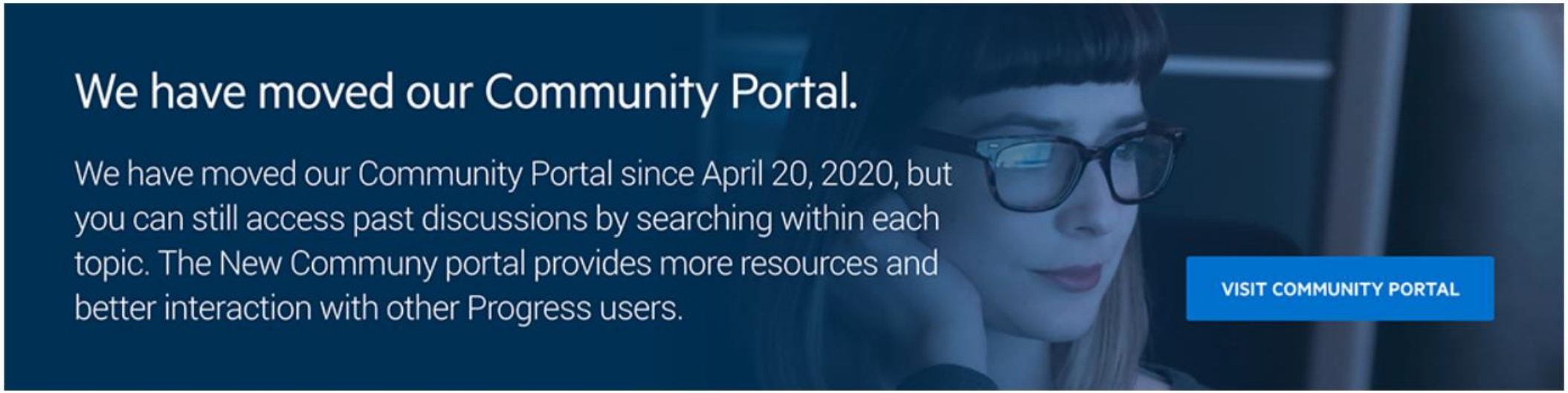
Progress®

# Nehmen Sie teil!


https://community.progress.com/s/
Progress Community
Login
Discussions
Groups
Resources
Product Downloads
Support
Progress Notifications
Welcome to the New Progress Community
Mingle with other Progress customers, partners and employees and find the answers to any challenges you may face.
Discussions
Share and discuss your challenges, best practices and successes
EXPLORE
Collaboration Groups
Collaborate with Progress developers, customers and partners
EXPLORE
SupportLink
Get the support you need to resolve your issues quickly
EXPLORE
Product Downloads
Installers, fixes and upgrades for all your Progress products
EXPLORE
Knowledge Base
Browse different articles that answer many commonly asked questions
EXPLORE
Product Documentation
Find technical documentation, whitepapers, webinars and other valuable resources
EXPLORE
Enhancement Requests
Submit your ideas for new features
EXPLORE
Progress Notifications
(PANs) Get critical product alerts and
EXPLORE
https://community.progress.com/s/

https://community-archive.progress.com

Progress Community | Archive of former Progress Community Discussions




Home

# Community Archive Index

https://community-archive.progress.com/

COGNITIVE SERVICES

Corticon (265)

DATA CONNECTIVITY AND INTEGRATION

OPENEDGE

OpenEdge BPM (214)

OpenEdge RDBMS (1166)

OpenEdge Development (6145)

PROGRESS USER GROUPS

Progress User Groups (164)

PUG Germany - Technical (6)

PUG Germany - General (49)

Progress OpenEdge

# OpenEdge Ideas

Your portal to submit new ideas (enhancement requests) for Progress OpenEdge. Anyone can submit and vote on ideas, and these will be reviewed and prioritized by the Product Management team. If you submitted or subscribe to an idea, you will get automatic email notifications of comments and significant status changes.

Add a new idea

My ideas 0
My votes 1
My subscriptions 1

FILTER BY STATUS

New 19
Under review 214
Already exists 11
Will not implement 332
Future consideration 85
In progress 8
Shipped 13

FILTER BY CATEGORY

Application Server 7
BPM 12
Development 381
Install and Deployment 50
OpenEdge Database 200

Recent | Trending | Popular

Search ideas

76 VOTE

## Complete OO functionality in ABL

Currently there are quite some OO features missing in ABL that are common in other OO languages like Java or C#:An extended list of wanted features is available at http://www.oehive.org/oowishlist, but I would like to repeat some here which I thin...

Created 6 months ago by Lieven De Foor
Development

In progress 0

19 VOTE

## Add server-side join support for open query statement (static query)

We would like to have SSJ support for static queries (using the open query statement) like:DEFINE QUERY q1 FOR Customer, order SCROLLING.OPEN QUERY q1 FOR EACH customer NO-LOCK, EACH order OF customer NO-LOCK.

Created about 2 months ago by Heino Vander Sanden
Other

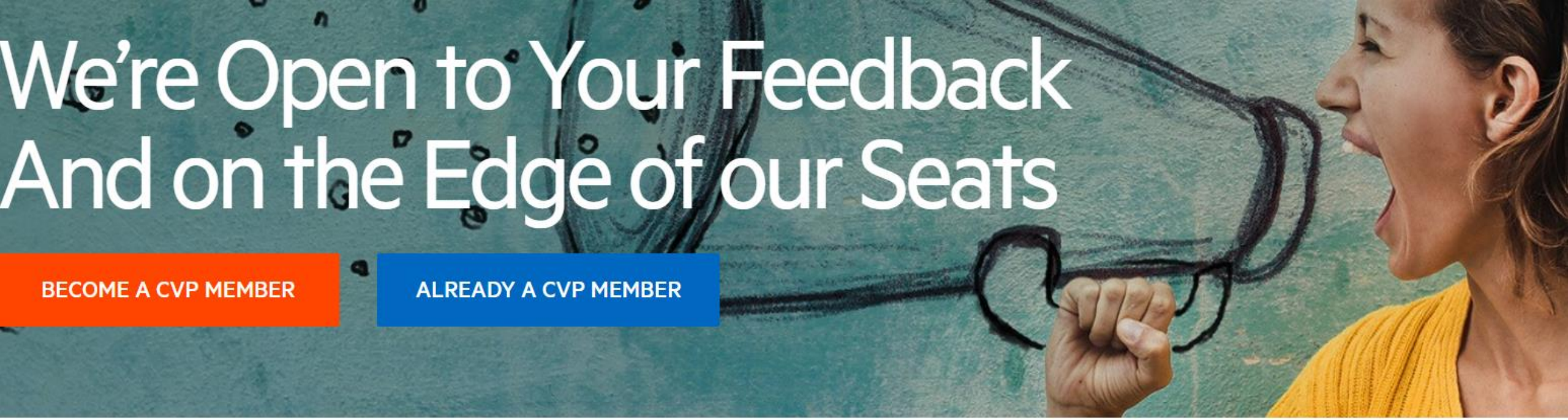
In progress 0

69 VOTE

## Unused Code

Provide optional compiler warnings for variables and properties which are not referenced in the compile unit.There is another Idea posted with similar ask here https://progresssoftware.aha.io/ideas/ideas/OPENEDGE-I-693Variables and properties that...

https://openedge.ideas.aha.io




# Delivering software that meets your high expectations is our number one priority.

Nun zu finden unter Community.Progress.com -> Groups, rechte Seite, Liste der CVP Gruppen.

OpenEdge CVP https://community.progress.com/s/group/0F94Q000000HakDSAS/openedge-cvp

Progress®

# Fragen?


Progress®