



APEX *connect*

APEX Connect- Online Konferenz vom 05.Mai bis 06.Mai 2020

Sensible Daten schützen – Zugriffe einschränken und nachvollziehen

APEX-AUTORISIERUNG IN DER PRAXIS WARTBAR MIT PL/SQL UMSETZEN



gunther@pipperr.de

Mein Blog

<https://www.pipperr.de/dokuwiki/>



Oracle Datenbank und APEX Tips
und Tricks

Zuletzt angesehen: • [start](#) • [oracle_dbsat](#)



Bergweg 14 - 37216 Witzenhausen/Roßbach

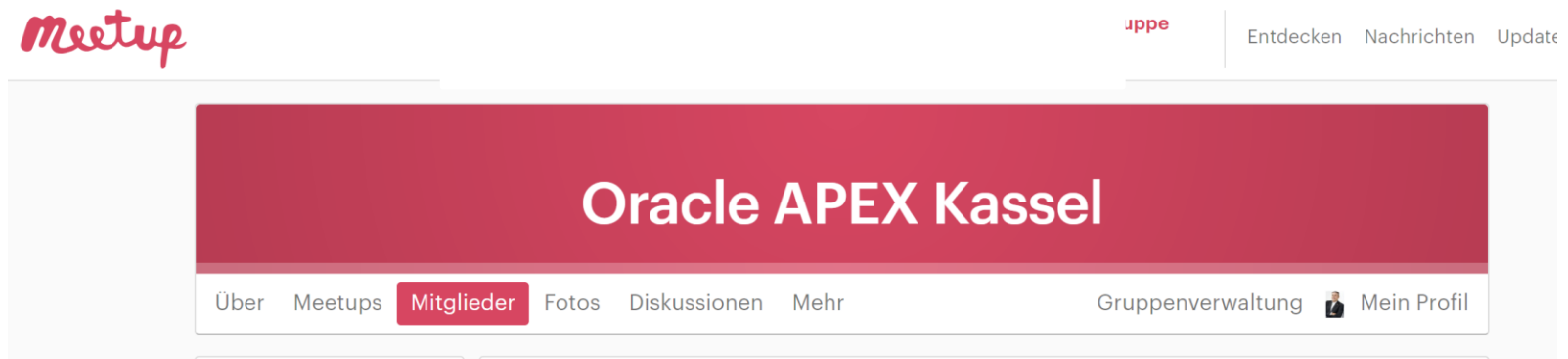
Freiberuflicher Oracle Datenbank Experte - Ich unterstütze Sie gerne in ihren Projekten.

APEX Meetup Gruppe Kassel-Göttingen

Raum für Veranstaltung in Kassel oder Göttingen gesucht, können Sie unterstützen?

Mitglieder gesucht!

<https://www.meetup.com/de-DE/Oracle-APEX-Kassel/>



Warum APEX ein alter HUT ist



<https://en.wikipedia.org/wiki/Flamen>

Schon die Römer hatten APEX



<http://thersitescorner.blogspot.com/2014/07/flamen-dialis.html>

Was heißt eigentlich APEX (1) ?

APEX =
Seltsamer
HUT



<https://en.wikipedia.org/wiki/Flamen>

Apex entered English from Latin, where it originally meant "a small rod at the top of a flamen's cap."

What's a **flamen's cap**? **Flamens** were **priests** who devoted themselves to serving just one of the many ancient Roman gods (for instance, just Jupiter or Mars).

Those priests wore **distinctive conical caps** that English speakers dubbed "**flamen's caps**."

Sixteenth- and seventeenth-century dramatist Ben Jonson was one of the few English writers known to have **used "apex" in its flamen's-cap sense**: "Upon his head a hat of delicate wool, **whose top ended in a cone**, and was thence called **apex**."

<https://www.merriam-webster.com/dictionary/apex>

Agenda

- 1 **Die Ausgangslage**
- 2 Autorisieren
- 3 Grundlagen
- 4 In der Praxis dynamisch umsetzen
- 5 Fazit



Die Ausgangslage (1)

- Eine der unbeliebtesten – aber auch wichtigsten – Aufgaben bei der Entwicklung einer Anwendung ist die Umsetzung eines **durchgängigen** Rechte und Rollen Konzepts für die **Autorisierung** der einzelnen Komponenten einer APEX Applikation.
- Die Umsetzung dieser Sicherheits-Anforderungen darf aber die Entwicklung nicht zu stark durch zu viel Komplexität einschränken

Die Ausgangslage (2)

- Die spätere Pflege und Wartung der Applikation muss verständlich bleiben
- Eine Anpassung von Rechte und Rollen Regeln muss auch ohne Programmieraufwand durchgeführt werden können
- Applikation muss „selbst lernend“ neue Items/Pages schützen / konfigurierbar erstellen

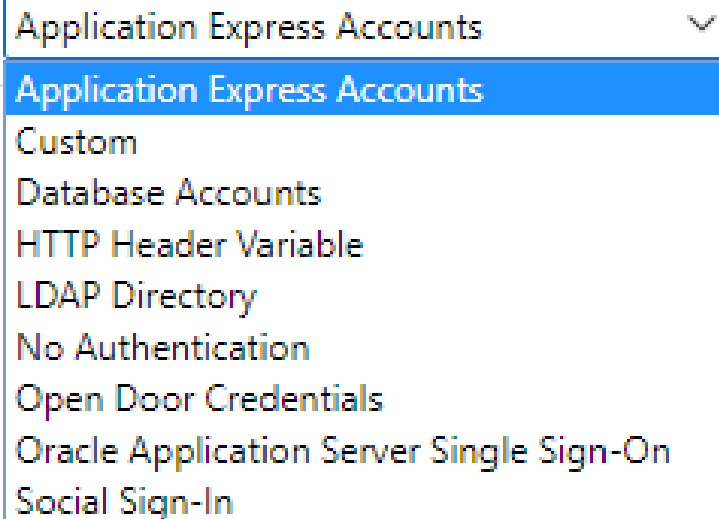
Die Ausgangslage (3)

- Auch die eigentlichen Daten soll vor unerwünschten Zugriffen geschützt werden
- Zugriff auf müssen nachvollziehbar sein

Authentifizieren und Autorisieren

Authentication

- Wie **anmelden**?
- APEX "**Authentication Scheme**"

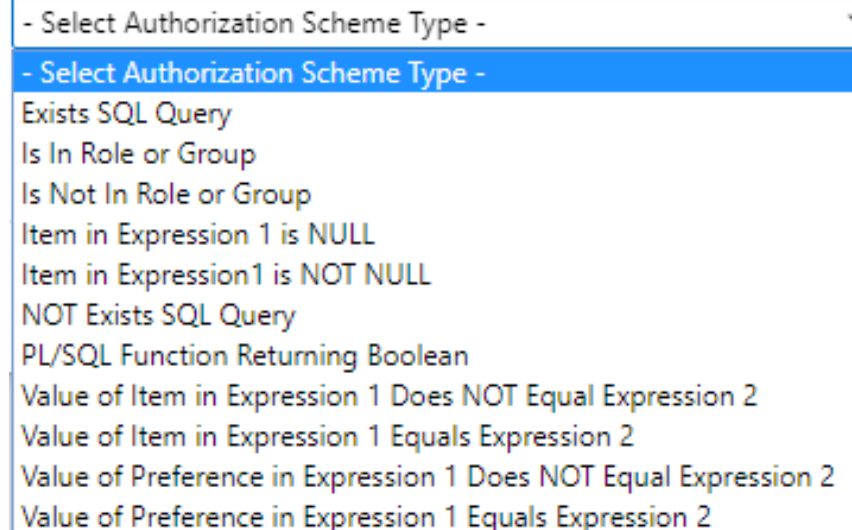


A screenshot of a dropdown menu for selecting an authentication scheme. The menu is open, showing a list of options. The first option, 'Application Express Accounts', is highlighted in blue. The dropdown has a small downward arrow icon on the right side of the header.

- Application Express Accounts
- Application Express Accounts
- Custom
- Database Accounts
- HTTP Header Variable
- LDAP Directory
- No Authentication
- Open Door Credentials
- Oracle Application Server Single Sign-On
- Social Sign-In

Authorization

- Was ist **erlaubt**?
- APEX "**Authorization Schemes**"



A screenshot of a dropdown menu for selecting an authorization scheme. The menu is open, showing a list of options. The first option, '- Select Authorization Scheme Type -', is highlighted in blue. The dropdown has a small downward arrow icon on the right side of the header.

- Select Authorization Scheme Type -
- Select Authorization Scheme Type -
- Exists SQL Query
- Is In Role or Group
- Is Not In Role or Group
- Item in Expression 1 is NULL
- Item in Expression1 is NOT NULL
- NOT Exists SQL Query
- PL/SQL Function Returning Boolean
- Value of Item in Expression 1 Does NOT Equal Expression 2
- Value of Item in Expression 1 Equals Expression 2
- Value of Preference in Expression 1 Does NOT Equal Expression 2
- Value of Preference in Expression 1 Equals Expression 2

Auf welcher Ebene die **Daten** direkt schützen?

APEX

- Die APEX Oberfläche stellt sicher, dass nur die Daten dargestellt werden die “sichtbar” sein dürfen

**Entwickler muss
alles richtig umsetzen!
Viel Logik in
der Applikation notwendig**

Datenbank

- Zugriff auf Daten “absolut” schützen
 - Oracle Virtual private database – **VPD** (nur EE)
 - Fine Grained Access Control (nur EE + Option)
 - **View Schicht** für SE

**Daten sind vor allen
Zugriffen geschützt**

Authorization

Darf der Anwender die Seite oder das Page Element sehen bzw. bedienen?

Step 1 – Rollen

Rollen und Rechte

- Typisches Vorgehen – Anforderung wird in einer Rolle / Rechte Matrix definiert

Rolle / Recht	Anmelden	Kunden- status setzen	Kunden- status auswerten	Rechenkern starten	User Rechte verwalten
Operator	<input type="checkbox"/>				
Analyst	<input type="checkbox"/>		<input type="checkbox"/>		
Admin	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>

– Usw.

Wie das in APEX umsetzen - Was bietet APEX?

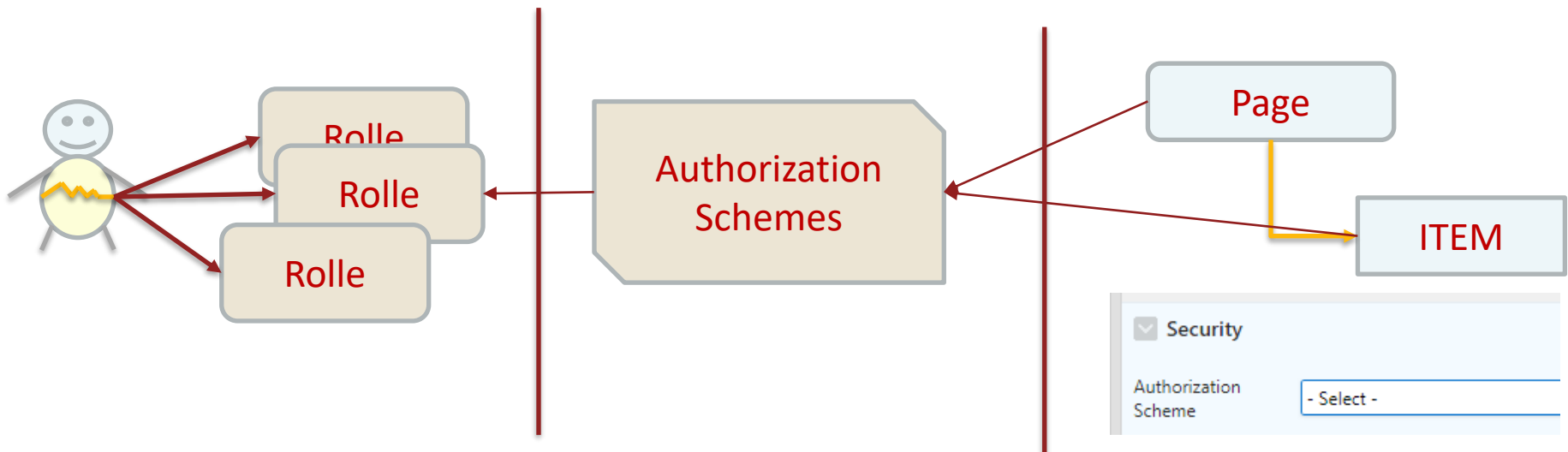
– Rollen

- In der Applikation oder im Workspace möglich
- Rollen enthalten aber keine Rechte!

– "Authorization Schemes" auf diese Rollen

- Müssen zuvor im Detail definiert werden

– "Authorization Schemes" auf Komponenten Ebene in APEX jeweils zuweisen

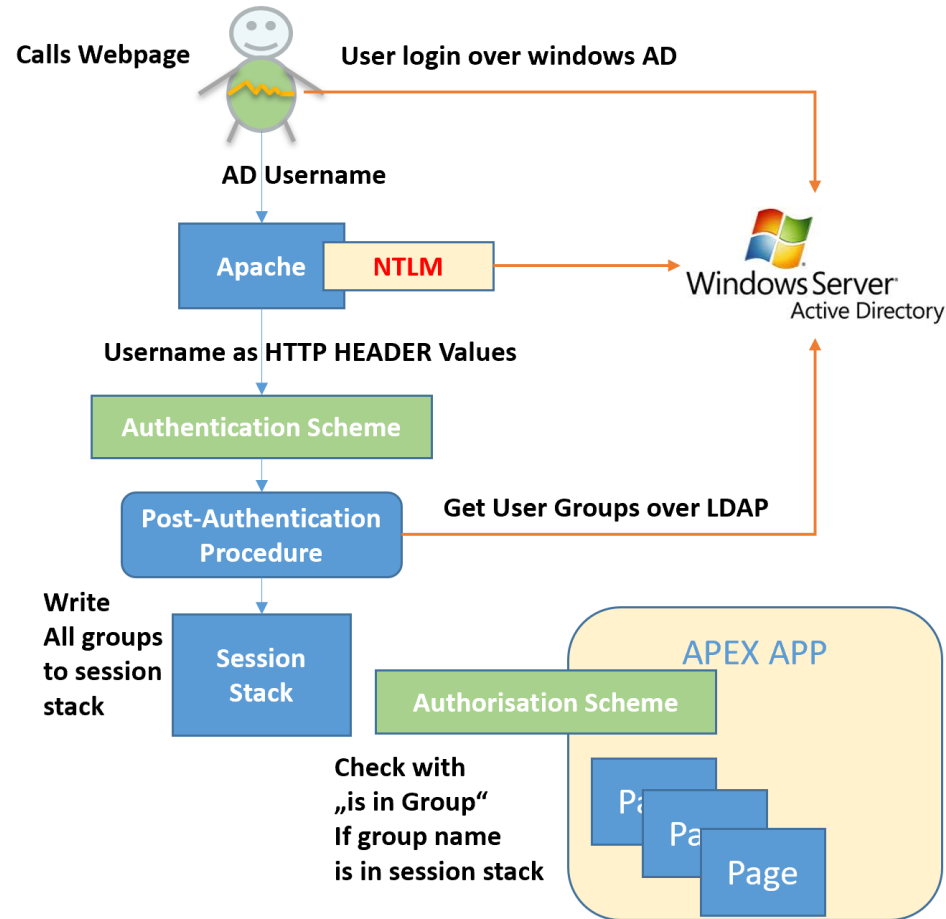


1. Schritt – Wie kommt der User zu seiner Rolle?

- Statisch zuweisen
 - In der Applikation mit "Application Access Control"
 - Vorteil – In der Applikation verwaltbar
 - Nachteil – Zuordnung zu den eigentlichen Usern wird nicht mit deployed
 - Im Workspace über "Groups"
 - Vorteil – Für alle Applikationen gültig
 - Nachteil – Gruppen müssen außerhalb der Applikation gepflegt und angelegt werden und müssen auch zu allen anderen Applikationen passen
- Dynamisch beim Login zuweisen
 - Z.B. um die Windows AD Rollen in APEX zu verwenden
 - APEX API `apex_authorization.enable_dynamic_groups`

Windows Rollen in APEX integrieren

- AD Integration mit Apache auf einem Windows Server



APEX API

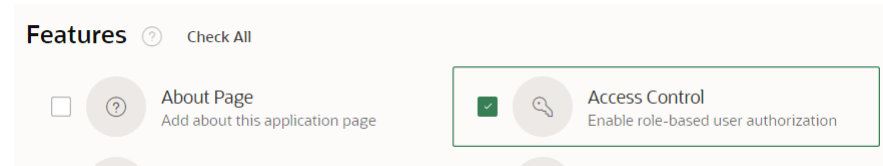
`apex_authorization.enable_dynamic_groups`

https://www.pipperr.de/dokuwiki/doku.php?id=prog:oracle_apex_active_directory_integration

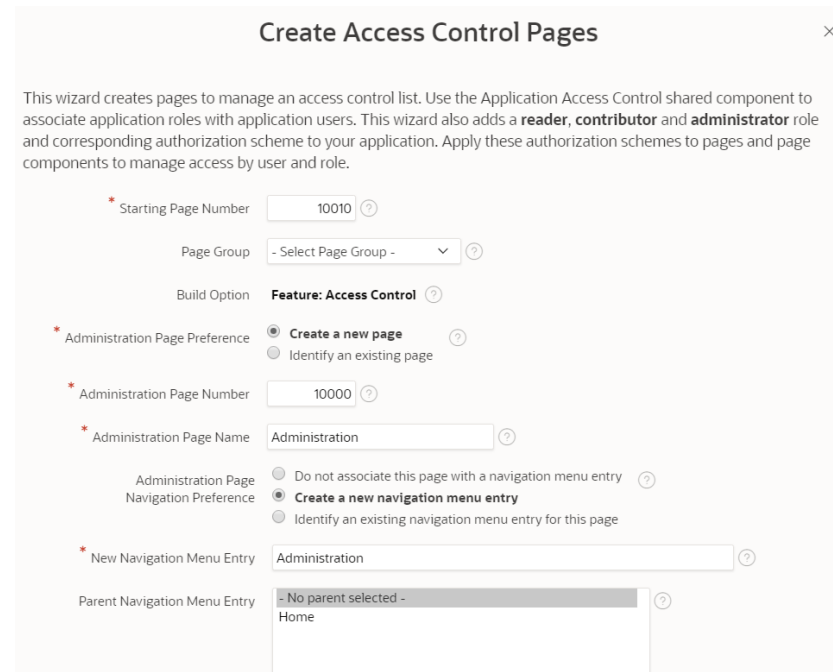
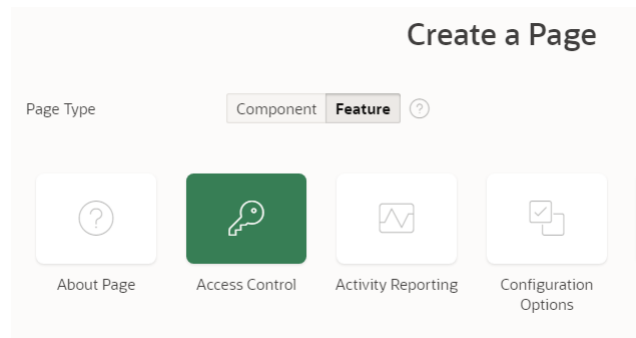
https://www.pipperr.de/dokuwiki/doku.php?id=prog:oracle_rest_data_service_tomcat#single_sign_on_mit_dem_apache_einrichten

Application Access Control aktivieren

- Beim Anlegen der APP über den APP Wizard als Feature einschalten





- Nachträglich über



Ein/Ausschalten über

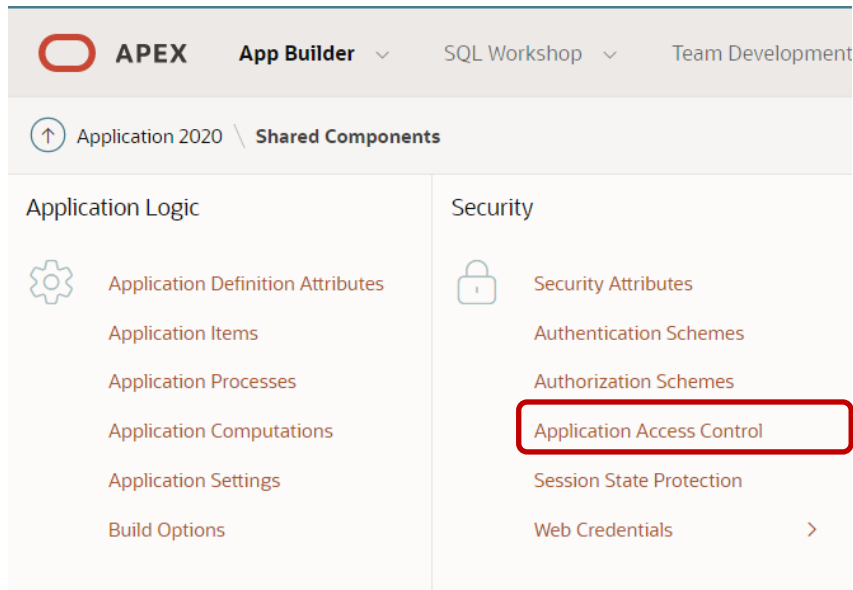
- Shared Components /Build Options

↑ Application 2020 \ Shared Components \ Build Options					
Build Options Utilization History					
Q ▾ Go   Actions ▾					
Name ↑ ▾	Application	Status	Default on Export	On Upgrade Keep Status	Comment
Feature: Access Control	2020	Include	Include	No	Incorporate role based user authentication within your application and manage username mappings to application roles.

↑ Application 2020 \ Shared Components \ Build Options \ Create / Edit	
Build Option	
Show All Attributes Associated Components	
Attributes	
* Application	2020 APEX_CONNECT_2020 ?
* Build Option	Feature: Access Control ?
Status	Include ?
Default on Export	- Same as Current Status - ?
On Upgrade Keep Status	<input type="checkbox"/> ?
Comments	Incorporate role based user authentication within your application and manage username mappings to application roles.

Application Access Control – Rollen verwalten

- In der Applikation Rollen und User definieren
 - Basis für ein passendes "Authorization Schemes"



The screenshot shows the 'Application Access Control' page. It has tabs for 'Show All', 'Roles', and 'User Role Assignments'. The 'Roles' tab is active. Below the tabs is a search bar with a magnifying glass icon, a 'Go' button, and an 'Actions' dropdown. A table lists the roles:

Role	Static Identifier
ADMINISTRATOR	ADMINISTRATOR
ANALYST	ANALYST

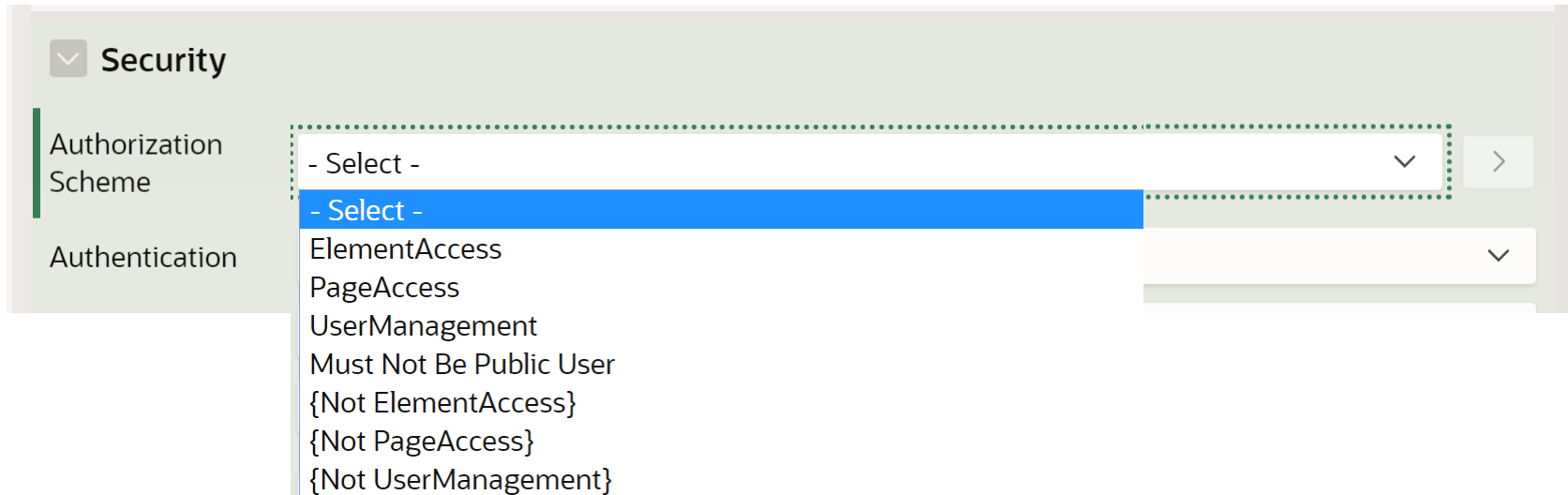
Name der Rollen NIE mehr ändern!!

- Kann über PL/SQL gesteuert werden
 - API – APEX_ACL

Step 2 – Authorization Scheme

Wie schütze ich ein Element in APEX?

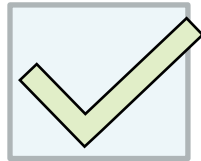
- Auf den Security Eigenschaften eines Elements ein "Authorization Scheme" hinterlegen
 - Z.B. auf der PAGE Ebene



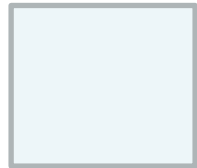
Für alle Seite unter **Utilities / Cross Page Utilities / Grid Edit** of all Pages einstellen

APEX "Authorization Scheme"

- Ein “pass/fail” Checks für alle Elemente der APEX Applikation



PASS



FAIL

- Kann auf den meisten Komponenten / Ebenen in APEX direkt definiert werden
 - Check muss erfüllt oder NICHT (NOT) erfüllt werden

APEX "Authorization Scheme"

- Beim ersten Aufruf auf Session Ebene oder je PAGE View möglich
 - Performance beachten!
 - Auf den Evaluation Point achten!
 - Die notwendigen Runtime Parameter **:APP_COMPONENT_TYPE**, **:APP_COMPONENT_ID** und **:APP_COMPONENT_NAME** können nur auf „Once per component“ und “Always (No Caching)” ausgelesen werden!

Evaluation Point	
Validate authorization scheme:	<div><input type="radio"/> Once per session</div> <div><input type="radio"/> Once per page view</div> <div><input checked="" type="radio"/> Once per component</div> <div><input type="radio"/> Always (No Caching)</div>

APEX "Authorization Scheme" - Anlegen

■ Welche Möglichkeiten haben wir?

Create Authorization Scheme

Use this page to define an authorization scheme. By creating an authorization schemes, you can protect applications, pages, and application components and extend the security provided by your application authentication scheme. You can use authorization schemes to identify additional security beyond simple user authentication. For example a user with administration rights may need access to more navigation bar icons, pages, and tabs than other users.

Application: **2020 APEX_CONNECT_2020**

* Name:

* Scheme Type: **- Select Authorization Scheme Type -**

Identify error message displayed when scheme violated:

Validate authorization scheme: ☒ **Once per session**
☐ Once per page view
☐ Once per component
☐ Always (No Caching)

Comments:

- Select Authorization Scheme Type -

- Select Authorization Scheme Type -

Exists SQL Query

Is In Role or Group

Is Not In Role or Group

Item in Expression 1 is NULL

Item in Expression1 is NOT NULL

NOT Exists SQL Query

PL/SQL Function Returning Boolean

Value of Item in Expression 1 Does NOT Equal Expression 2

Value of Item in Expression 1 Equals Expression 2

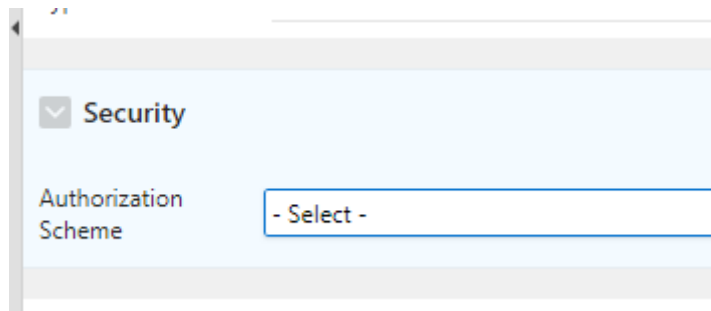
Value of Preference in Expression 1 Does NOT Equal Expression 2

Value of Preference in Expression 1 Equals Expression 2

Performance => PL/SQL möglichst wenig verwenden

"Authorization Scheme" **sicher** verwenden

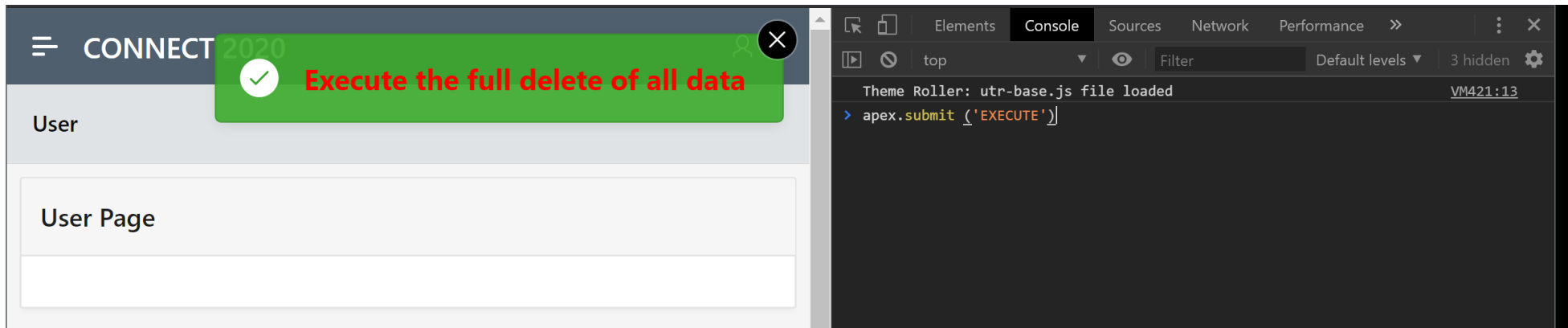
- Sicherheit
 - "Authorization Scheme" blendet ein Element nur aus!
 - Der Prozess dahinter muss aber auch mit dem gleichen "Authorization Scheme" geschützt werden, damit dieser nicht doch über eine URL aufgerufen werden kann!



Einen “**versteckten**” Button in APEX **aufrufen**

- F12, Java Script Konsole aufrufen, Submit absetzen

```
apex.submit ('<BUTTON_NAME>')
```



Ist das ausreichend dynamisch?

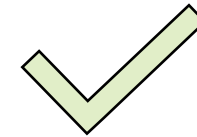
■ Problem:

- NUR ein "Authorization Scheme" pro Element möglich
- Eine fachliche Rolle mit vielen fachlichen Rechten kann zu vielen "Authorization Schemes" führen, um jeden notwendigen Technischen Schritt dahinter auch abzusichern
- Bei mehr als x Rollen mit mehr als xxx Rechten auf vielen Seiten wird die Konfiguration von einzelnen "Authorization Schemes" pro Rolle/Recht Kombination sehr unübersichtlich
- Wie Navigationselemente mit den dazugehörigen Seitenrechten synchronisieren?
- Wie mit den dazugehörigen "Processes" der Seiten?
- Wie das zum Schutz von „Links“ in Berichten verwenden?

Lösung: Eigene Implementierung in PL/SQL

- Voraussetzung
 - Item/Element in APEX kann eindeutig identifiziert werden

• Page => PAGE_ID



• Page ITEM =>



- ID des Page ITEM kann sich beim Deployment in PROD ändern!
- Lösung => **Static ID** auf Page Item vergeben – Page Item immer über PAGE_ID und STATIC_ID im eigenen Code referenzieren

Mit PL/SQL eigene Lösung implementieren

- Wichtigste Information
 - Aus welchem Element wurde das "Authorization Scheme" aufgerufen
 - apex_debug.message('app_component_id = ' ||
:APP_COMPONENT_ID);
 - apex_debug.message('app_component_name = ' ||
:APP_COMPONENT_NAME);
 - apex_debug.message('app_component_type = ' ||
:APP_COMPONENT_TYPE);

Lösung in der Praxis – Seitenaufruf pro Rolle (1)

- Tabelle mit allen Rollen und Seiten
- In der Tabelle wird definiert, mit welcher Rolle welche Seite aufgerufen werden kann
- Admin Seite zur Pflege der Zuordnung
- PL/SQL Funktion auf Basis der Tabelle prüft ob User / Seite und Rolle passen => return true => Zugriff erlaubt
 - Falls Seite noch nicht existiert, eintragen und Admin benachrichtigen
- "Authorization Scheme" anlegen (PL/SQL)
- Auf jeder Seite das gleiche "Authorization Scheme" definieren
- Rechte Tabellen initialisieren bzw. jede Seite aufrufen

Lösung in der Praxis – Seitenaufruf pro Rolle (2)

- Tabelle mit allen Rollen und Seiten inkl. Pflegemaske

Administration \ Page to Role

Q

Search: All Text Columns

Go

Actions

Edit

Save

Add Row

Reset

Page					Role							
		Page Id ↑	Page Name	Menu Id	Operator Readonly	Operator	Operator Senior	Analyst	Analyst Senior	User Admin	Administrator	Developer
<input type="checkbox"/>		0	Global Page - Desktop	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/>		1	Home	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<input type="checkbox"/>		1000	Dash Board	-	No	No	No	Yes	Yes	No	Yes	No
<input type="checkbox"/>		2000	Client Selection	-	No	No	No	No	No	No	No	No
<input type="checkbox"/>		2010	Client Overview	-	No	No	No	No	No	No	No	No
<input type="checkbox"/>		2020	Client History	-	No	No	No	No	No	No	No	No
<input type="checkbox"/>		2021	Client Detail Modal	-	No	No	No	No	No	No	No	No

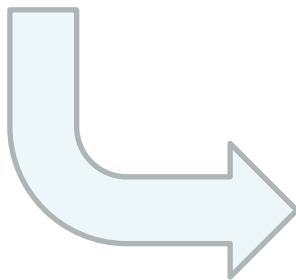
```
create table T_SEC_PAGE_ACCESS (  
  DDE_SK  
  ,PAGE_NAME  
  ,PAGE_ID  
  ,MENUE_ID  
  ,OPERATOR_READONLY  
  ,OPERATOR  
  ,OPERATOR_SENIOR  
  ,ANALYST  
  ,ANALYST_SENIOR  
  ,USER_ADMIN  
  ,ADMINISTRATOR  
  ,DEVELOPPER  
)  
;  
  
number(15) not null constraint t_sec_page_access_id_pk primary key  
varchar2(255)  
number(15)  
number(15)  
varchar2(1) default 'N'  
varchar2(1) default 'N'  
varchar2(1) default 'N'  
varchar2(1) default 'N'  
varchar2(1) default 'N'  
varchar2(1) default 'N'  
varchar2(1) default 'N'  
varchar2(1) default 'N'
```

Nachteil – Rollen nicht dynamisch
Vorteil – Pflege über GRID einfacher

Lösung in der Praxis – Seitenaufruf pro Rolle (3)

- PL/SQL Funktion auf Basis von **APEX_APPL_ACL_USER_ROLES** und der eigenen Tabelle

```
select listagg(u.role_name,':') WITHIN GROUP (ORDER BY u.role_name)
into v_return
from APEX_APPL_ACL_USER_ROLES u
where u.application_id=p_app_id
and u.user_name=p_username
group by u.user_name;
```



```
select
decode(OPERATOR_READONLY, 'N', null, 'OPERATOR_READONLY')
||': '||
decode(OPERATOR, 'N', null, 'OPERATOR')
||': '||
decode(OPERATOR_SENIOR, 'N', null, 'OPERATOR_SENIOR')
||': '||
decode(ANALYST, 'N', null, 'ANALYST')
||': '||
decode(ANALYST_SENIOR, 'N', null, 'ANALYST_SENIOR')
||': '||
decode(USER_ADMIN, 'N', null, 'USER_ADMIN')
||': '||
decode(ADMINISTRATOR, 'N', null, 'ADMINISTRATOR')
||': '||
decode(DEVELOPPER, 'N', null, 'DEVELOPPER')
into v_role_needed
from T_SEC_PAGE_ACCESS
where page_id=p_page_id;
```

Vergleich: ob Rolle des Users in dieser Liste ist =>

Lösung in der Praxis – Seitenaufruf pro Rolle (4)

- "Authorization Scheme" anlegen (PL/SQL)
 - Gibt TRUE oder FALSE zurück

Authorization Scheme

* Scheme Type PL/SQL Function Returning Boolean ?

* PL/SQL Function Body ?

```
v_check:=PKG_DDE_APEX_UTILS.checkPageAuthorisation (  
    p_user      => :APP_USER  
    , p_application_id => :APP_ID  
    , p_page_id   => :APP_PAGE_ID  
    , p_component_id => :APP_COMPONENT_ID  
    , p_component_name => :APP_COMPONENT_NAME  
    , p_component_type => :APP_COMPONENT_TYPE );
```

* Identify error message displayed when scheme violated

To access the user &APP_USER, you need the correct role!

Evaluation Point

Validate authorization scheme:

- ☐ Once per session ?
- ☐ Once per page view
- ☒ **Once per component**
- ☐ Always (No Caching)

Variable nicht immer gefüllt!
Auf den "Evaluation Point" achten!

Beispiel – Page Authorization Scheme

```
declare
  v_check boolean:=false;

begin

  apex_debug.message('app user           = ' || :APP_USER);
  apex_debug.message('app_id           = ' || :APP_ID);
  apex_debug.message('page id          = ' || :APP_PAGE_ID );
  -----
  apex_debug.message('app_component_id   = ' || :APP_COMPONENT_ID);
  apex_debug.message('app_component_name = ' || :APP_COMPONENT_NAME);
  apex_debug.message('app_component_type = ' || :APP_COMPONENT_TYPE);
  -----

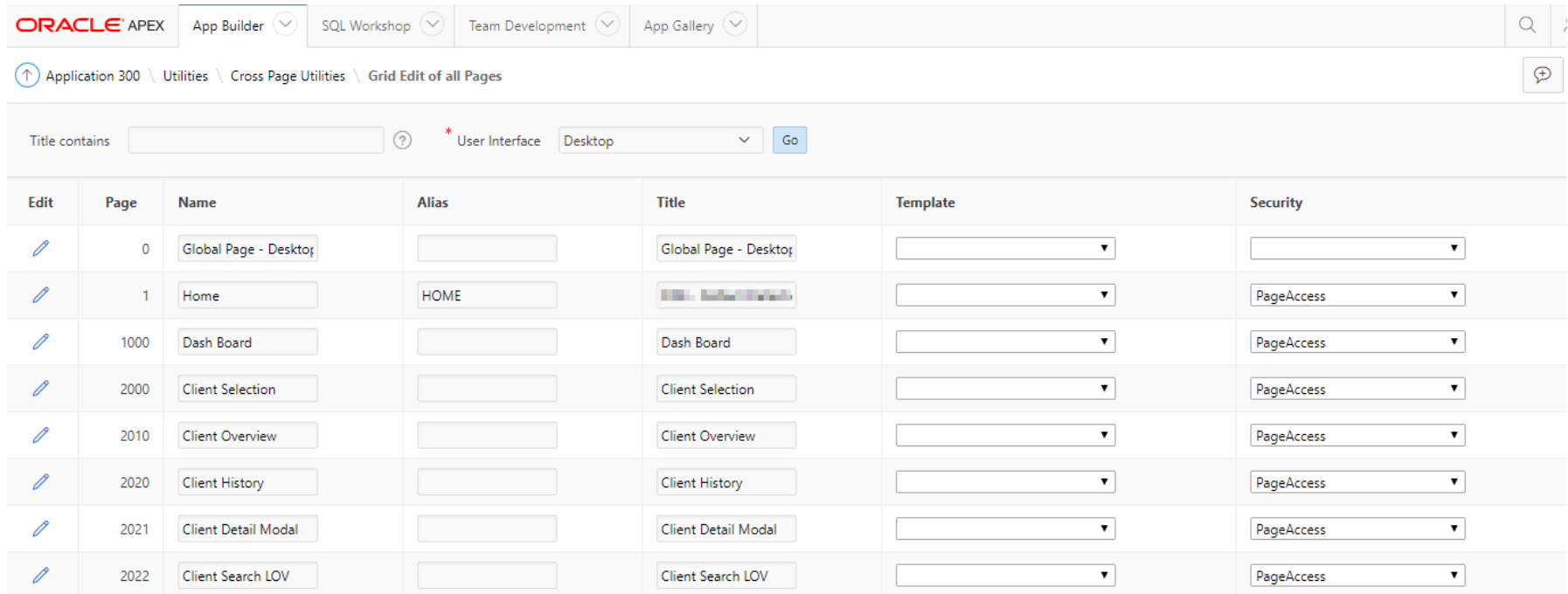
  v_check:=PKG_APEX_SECURITY.checkPageAuthorisation (
                                p_user           => :APP_USER
                              , p_application_id => :APP_ID
                              , p_page_id       => :APP_PAGE_ID
                              , p_component_id  => :APP_COMPONENT_ID
                              , p_component_name => :APP_COMPONENT_NAME
                              , p_component_type => :APP_COMPONENT_TYPE );









  return v_check;

end;
```

Lösung in der Praxis – Seitenaufruf pro Rolle (5)

- Auf jeder Seite das gleiche "Authorization Scheme" definieren
 - Utilities / Cross Page Utilities / Grid Edit of all Pages





Edit	Page	Name	Alias	Title	Template	Security
	0	Global Page - Desktop		Global Page - Desktop		
	1	Home	HOME			PageAccess
	1000	Dash Board		Dash Board		PageAccess
	2000	Client Selection		Client Selection		PageAccess
	2010	Client Overview		Client Overview		PageAccess
	2020	Client History		Client History		PageAccess
	2021	Client Detail Modal		Client Detail Modal		PageAccess
	2022	Client Search LOV		Client Search LOV		PageAccess

Login Seite und Seite 0 nicht einschränken

Gleiches Prinzip auf Komponenten Ebene

- Ein "Scheme" für Regions und Buttons
- Steuertabelle + Pflegemaske

		APEX Element				Role							
<input type="checkbox"/>		Page Id ↑	Page Name	Id	Name	Operator Readonly	Operator	Operator Senior	Analyst	Analyst Senior	User Admin	Admin-istrator	Dev-elopper
▼		Type: APEX_APPLICATION_BUTTONS											
<input checked="" type="checkbox"/>		3200	Set Manual Def...	5709199726006517	Set █████ █████	No	Yes	Yes	No	No	No	Yes	No
▼		Type: APEX_APPLICATION_PAGE_REGIONS											
<input type="checkbox"/>		1	Home	10727856194810026	Waiting for approval to set the trigger status to...	No	Yes	Yes	No	No	No	Yes	No
<input type="checkbox"/>		1	Home	7537196315184848	List of █████ in the actual █████ run	No	No	No	Yes	Yes	No	Yes	No
<input type="checkbox"/>		1	Home	25569281368664048	User Report	No	No	No	No	No	Yes	Yes	No
<input type="checkbox"/>		1	Home	2292619037962038	█████ Engine Background Jobs	No	No	No	No	No	No	Yes	No
<input type="checkbox"/>		1000	Dash Board	8150354292050313	Count of distinct █████ numbers pe...	No	No	No	Yes	Yes	No	Yes	No
<input type="checkbox"/>		1000	Dash Board	1963432565773430	█████ Counter Summary (For each █████ cou...	No	No	No	Yes	Yes	No	Yes	No
1 rows selected													
Total 7													

Achtung!

ID nicht stabil bei Export/Import – Page ID und
STATIC_ID des Elements verwenden

Demo – PL/SQL "Authorization Scheme"

- Demo



Source Code: https://github.com/gpipperr/APEX_CONNECT_2020_Dynamic_Roles_and_Rights

Wie Links in einem Report schützen?

- In einem Bericht soll ein Link zum Bearbeiten nur dann aktiv sein, wenn :
 - Der Anwender die entsprechenden Rechte in APEX besitzt
 - Der Datensatz für seine Rolle auch freigeschaltet ist.

https://www.pipperr.de/dokuwiki/doku.php?id=prog:apex_authorization_scheme_protect_link

In PL/SQL ein "Scheme" abfragen

```
create or replace function checkRowAccess( p_row_val varchar2 ,p_sec_check varchar2 ,p_security_scheme varchar2 ,p_link_to varchar
, p_link_item varchar2
, p_session varchar2
, p_app_id varchar2)
return varchar2
is
    v_return          varchar2(8000);
    v_admin_user       boolean:=false;
    v_row_link         varchar2(8000);

begin
    -- Link erzeugen
    v_row_link := '<a href="'
        || APEX_UTIL.PREPARE_URL( p_url => 'f?p='
        || p_app_id
        || ':'
        || p_link_to
        || ':'
        || p_session
        || '::NO::'
        || p_link_item
        || ':'
        || p_row_val
        , p_checksum_type => 'SESSION')
        || '"></a>';

    pürfe nur wenn nötig
    if p_sec_check = '1' then
        v_admin_user := apex_util.public_check_authorization(p_security_scheme => p_security_scheme );
    end if;

    if p_sec_check = '1'
    and v_admin_user=false then
        v_return:=p_row_val;
    else
        v_return:=v_row_link;
    end if;

return v_return;

end checkRowAccess;
```

Beispiel um einen Link in einem Bericht abzusichern

Ab 19.1 - APEX_AUTHORIZATION.IS_AUTHORIZED (
p_authorization_name IN VARCHAR2)
RETURN BOOLEAN;

Anwendung:

Application 2020 \ Share

Authorization Schemes

LinkAccessADMINISTRATOR	PL/SQL Function Returning Boolean	Always (No Caching)
LinkAccessANALYST	PL/SQL Function Returning Boolean	Always (No Caching)
LinkAccessOPERATOR	PL/SQL Function Returning Boolean	Always (No Caching)

```
v_check:=PKG_APEX_SECURITY.checkUserInRole( p_username => :APP_USER
                                             , p_role    => 'OPERATOR'
                                             , p_app_id  => :APP_ID );

return v_check;
```

```
1 select SEC_SK
2       , checkRowAccess(SEC_SK           -- p_row_val
3       , '1'                           -- ,p_sec_check
4       , 'LinkAccess'||EDITOR_ROLE      -- ,p_security_scheme
5       , '10'                           -- ,p_link_to
6       , 'P10_SEC_SK'                   -- ,p_link_item
7       , '&SESSION.'                     -- ,p_session
8       , '&APP_ID.'                      -- ,p_app_id
9       )
10     AS LINK_TEXT
11     , CUSTOMER
12     , RABATT_LEVEL
13     , EDITOR_ROLE
14 from T_SECURE_DATA_EXAMPLES
```

Build Options

Build Options

- Konditional Elemente der Applikation freischalten oder sperren
- Ideal um Elemente aus der Entwicklung in der Produktion zu deaktivieren
- Kann über PL/SQL gesteuert werden
 - `APEX_UTIL.SET/GET_BUILD_OPTION_STATUS` API

Kein echtes Security feature – Ein weiterer Baustein für komplexe Problemstellungen

Die Daten schützen

Was ist mit den angezeigten Daten?



Daten möglichst auf der Datenbank schützen

- Virtual Private Database

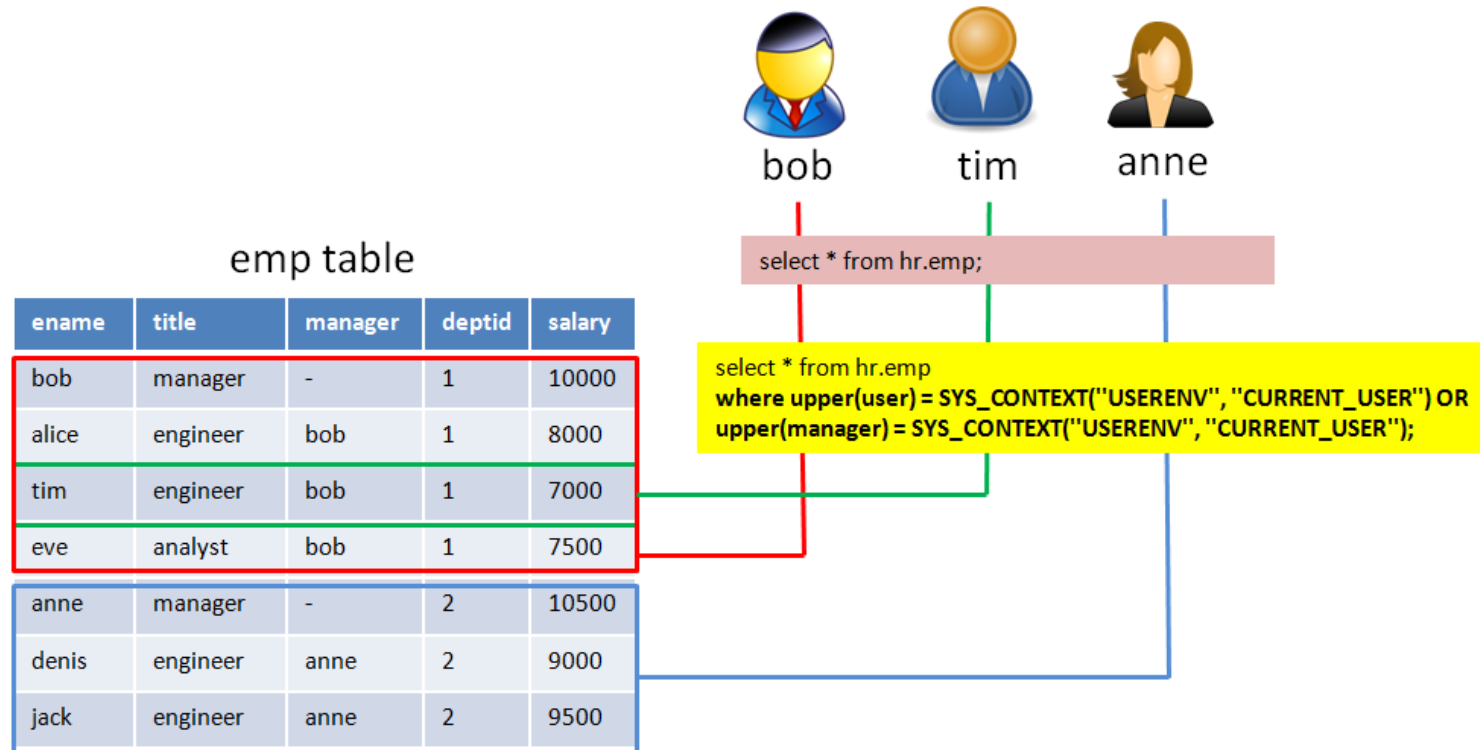


Bild: <http://mohamednabeel.blogspot.com/2014/08/oracle-virtual-private-database-vpd.html>

Lizenz?

- Oracle Label Security
 - \Rightarrow EE + Extra Option notwendig
- VPD - Virtual Private Database
 - \Rightarrow EE ohne Extra Option
- XE? Enthalten
- SE?
 - View Schicht implementieren

Summary

Fazit „APEX Sichern“

- Daten möglichst in der DB selber sichern
- Möglichst immer die Standard Methoden verwenden
- Falls es Komplex wird, ein eigenes Framework aufsetzen und gut testen

Source Code: https://github.com/gpipperr/APEX_CONNECT_2020_Dynamic_Roles_and_Rights

Mehr

- Quellen
 - [Matt Mulvaney](https://explorer.co.uk/apex-authentication-and-authorisation-for-forms-developers/) => <https://explorer.co.uk/apex-authentication-and-authorisation-for-forms-developers/>
- Blog Gunther Pippèrr
<https://www.pipperr.de/dokuwiki/doku.php>
 - Wieder mal eine andere Skript Library
 - <https://github.com/gpipperr/OraPowerShell>
- Bildmaterial : <https://pixabay.com>

Source Code: https://github.com/gpipperr/APEX_CONNECT_2020_Dynamic_Roles_and_Rights

F&A

Fragen



APEX-Autorisierung

Fragen ?

