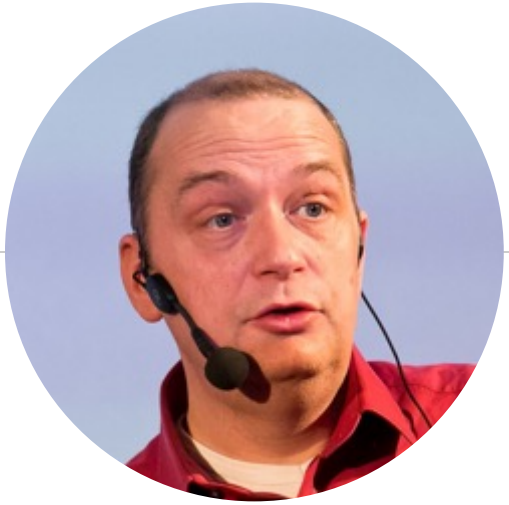
APEX, PLSQL and RESTful services II

publishing secure services



Hello!

I am Richard Martens

Welcome to this presentation.

You can find me at

- @rhjmartens
- smart4solutions.nl/blog
- https://bitbucket.org/smart4solutions/public-presentations/

smart4 solutions
IT's in our nature

Oracle ACE Pro

Oracle ACE Program

500+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical and community contributions to the Oracle community

3 membership tiers

Oracle ACE Director | Oracle ACE Pro | Oracle ACE Associate

For more details on Oracle ACE Program:
ace.oracle.com

Oracle ACE

Nominate

yourself or someone you know:

ace.oracle.com/nominate

Connect: aceprogram_ww@oracle.com Facebook.com/OracleACEs @oracleace

Agenda

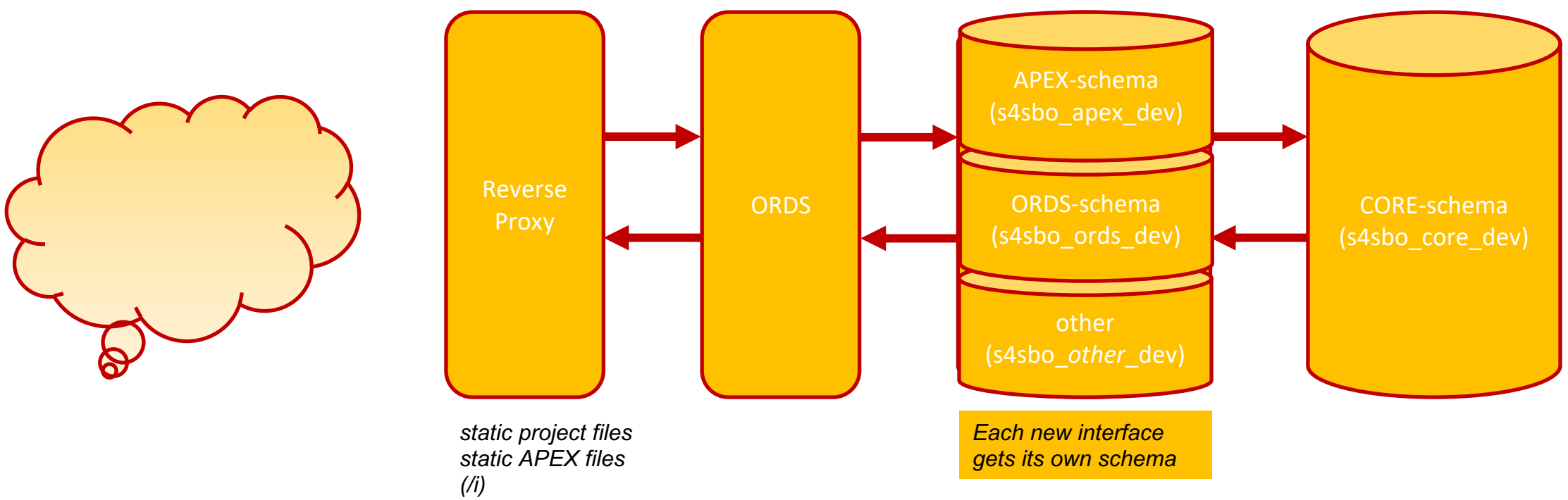
- Tools
- ORDS installation
- Create services
- Adding Authentication
- Requesting Tokens
- Adding Authorisation
- Adding a VPD on a service
- Some do’s and don’ts

Tools used

- PL/SQL Developer
- SQL Developer (only if needed ☺)
- APEX
- Postman

ORDS Installation

Standard infrastructure dev



First line of defence

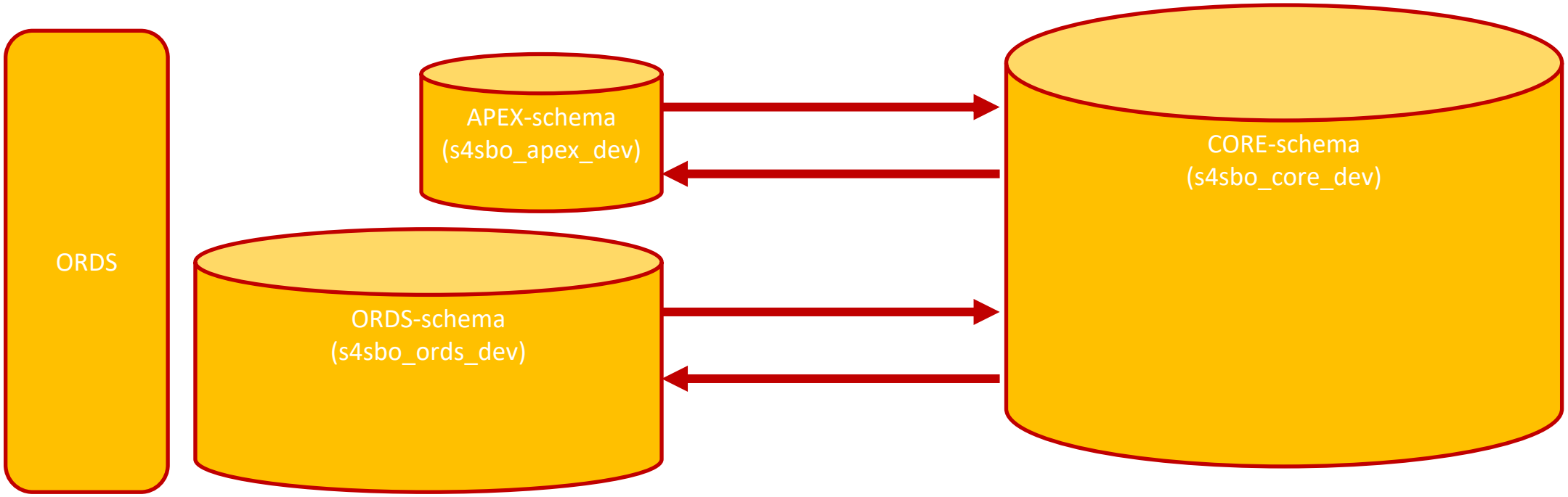
- All data resides in the “core” schema
- Only data that needs to be exposed is granted to the “ords” schema

- Each template (service) gets its own package / view in the ”ords” schema
  - The more fine-grained you can get the easier it will become to make changes in a later stage (manageability)
- Services should not call objects straight from the “core” schema

- These assumptions are the same for our APEX applications

- Also makes a clear division between “business” logic and “display” logic

Standard infrastructure dev



- views on tables inside CORE
- packages using objects in CORE
- All per module-template-handler ("fine-grained")

Our first service

- No authentication at all
- Enable the “ords” schema
  - Do not use alias that also exists as an APEX workspace
- Enabling a view
  - Choose an alias
  - ORDS takes care of the JSON

Our second service

- Still no authentication
- Enabling a package
  - Choose an alias
  - The request is a POST
  - Single values can be returned (OUT) as varchar or number
  - Arrays can be returned (OUT) as sys_refcursor
  - ORDS takes care of the JSON..
- P_ prefixes gives ugly json
- No P_ prefixes gives ugly PL/SQL

Adding authentication

Authentication consists of

- A role
  - ORDS.CREATE_ROLE
  - USER_ORDS_ROLES
- A privilege
  - Defines what is allowed (authentication-wise) through patterns or modules
  - Receives roles (define what roles have what privilege)
  - ORDS.DEFINE_PRIVILEGE
  - USER_ORDS_PRIVILEGES
  - USER_ORDS_PRIVILEGE_ROLES
  - USER_ORDS_PRIVILEGE_MAPPINGS

Adding authentication

We also have

- A client
  - Allowed to request a token through {{base-url}}/oauth/token
  - Receives roles
  - OAUTH.CREATE_CLIENT
  - OAUTH.GRANT_CLIENT_ROLE
  - USER_ORDS_CLIENTS
  - USER_ORDS_CLIENT_PRIVILEGES
  - USER_ORDS_CLIENT_ROLES

- Privileges can be given out to clients and to roles.
- The given views should be run as the user owning the objects (clients, privs and roles)

Requesting token

- Before the REST call we must get a token
- {{base-url}}/oauth/token
- Client-id and secret are essential
- See part I of this series

Adding Authorisation

- ORDS implicit parameters
  - body, body_text, content_type, *current_user*, forward_locaton, fetch_offset, fetch_size, page_offset, page_size, row_offset, row_count, status_code
  - They act like bind-variables
  - There is no V function (yet)
- With current_user, the USER_ORDS_* views, and maybe some authorisation-tables we can define a poor mans VPD
- We must add the where condition ourselves.

VPD I - preparation

```
oracle-> sqlplus / as sysdba

SQL*Plus: Release 18.0.0.0.0 - Production on Mon Oct 10 13:41:06 2022
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.


Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> alter session set container=xepdb1;

Session altered.

SQL> grant execute on dbms_rls to presrest_ords_dev;

Grant succeeded.

SQL> _
```

VPD ii – add policy

```
begin
  dbms_rls.add_policy(object_schema          => 'CORE_SCHEMA'
                     ,object_name            => 'TABLE_NAME'
                     ,policy_name            => 'name'
                     ,function_schema        => 'ORDS_SCHEMA'
                     ,policy_function        => 'package.function'
                     ,sec_relevant_cols      => 'HIDDEN,COLS'
                     ,sec_relevant_cols_opt => dbms_rls.all_rows
end;
/
```

VPD iii – policy function

- Return null when not applicable
- Return ‘1=1’ when applicable

```
--return null; -- policy is OFF
--return '1=0'; -- policy is ON
if l_reccount > 0
then
   l_retval := 'deptno=' || substr(p_role_name, -2);
else
   l_retval := null;
end if;
```

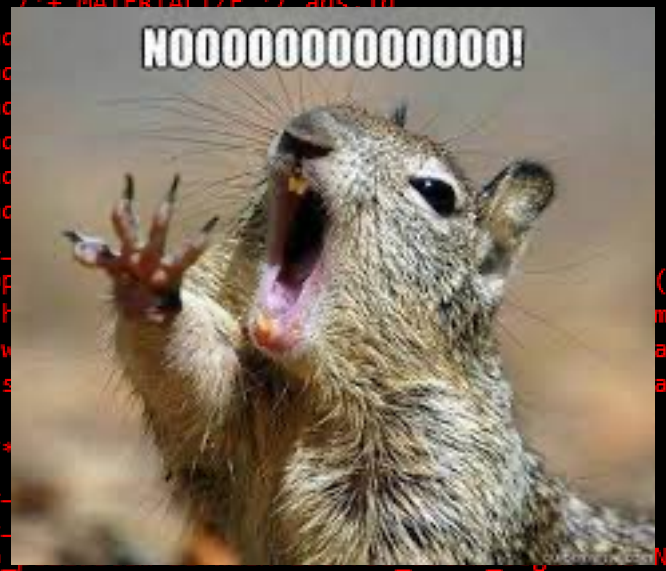
- Return value is added to the where condition by the database.

Demo

1. Add policy
    - Policy should be on a table, not a view
2. Add (packaged) predicate returning function
3. Do a regular SQL

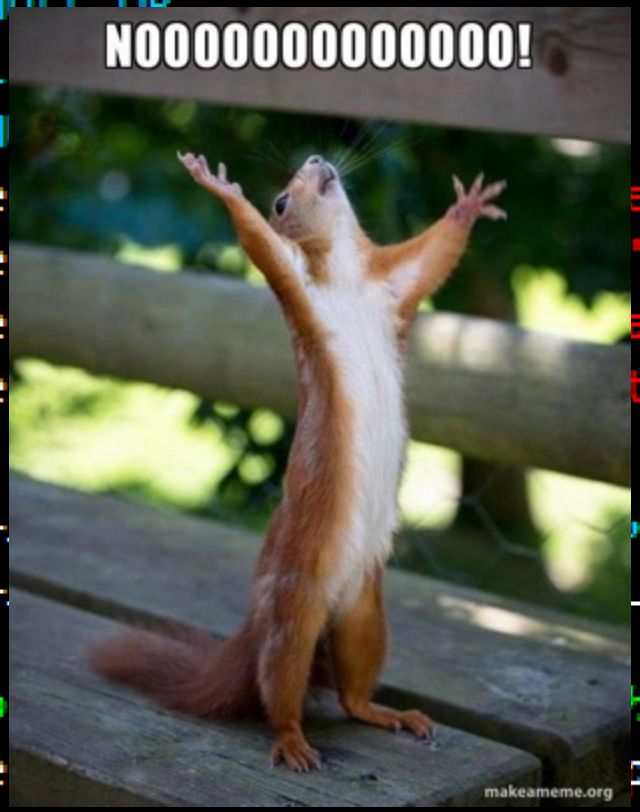
Gathering data do’s and don’ts

```
IF p_woonplaats IS NOT NULL OR
   p_straatnaam IS NOT NULL OR
   p_postcode IS NOT NULL OR
   p_huisnummer IS NOT NULL THEN
  -- Er zijn zoekcriteria meegegeven voor adres gegevens. Daarom query opbouwen met selectie op adres
  l_query := q'#WITH ads_sel as (SELECT /*+ MATERIALIZE */ ads.id
                                      ,ac
                                      ,ac
                                      ,ac
                                      ,ac
                                      ,ac
                                      ,ac
                                  FROM rbl_
                                  WHERE upp                                    (ads.postcode))
                                  AND ads.h                                    mer)
                                  AND ads.w                                    ats)
                                  AND ads.s                                    aam)
                                  )
                ,loc_ref AS (SELECT /*
                                  FROM rbl_
                                  JOIN ads_
                                  WHERE :b_                                    NVL(loc.rbl_datum_einde, :b_peildatum)
                                  AND  :b_peildatum BETWEEN loc.datum_begin AND NVL(loc.datum_einde, :b_peildatum)
                               )#';
  IF p_naam IS NOT NULL THEN
    -- ook de with clause op rbl namen toegevoegen
    l_query := l_query || '
               , nam as (SELECT /*+ MATERIALIZE */ DISTINCT(nam.rel_referentie)
                           ,  score(1) score
                        FROM  rbl_namen_vw nam
                        WHERE contains(nam.naam, ''fuzzy({' || p_naam ||
```



Gathering data do’s and don’ts

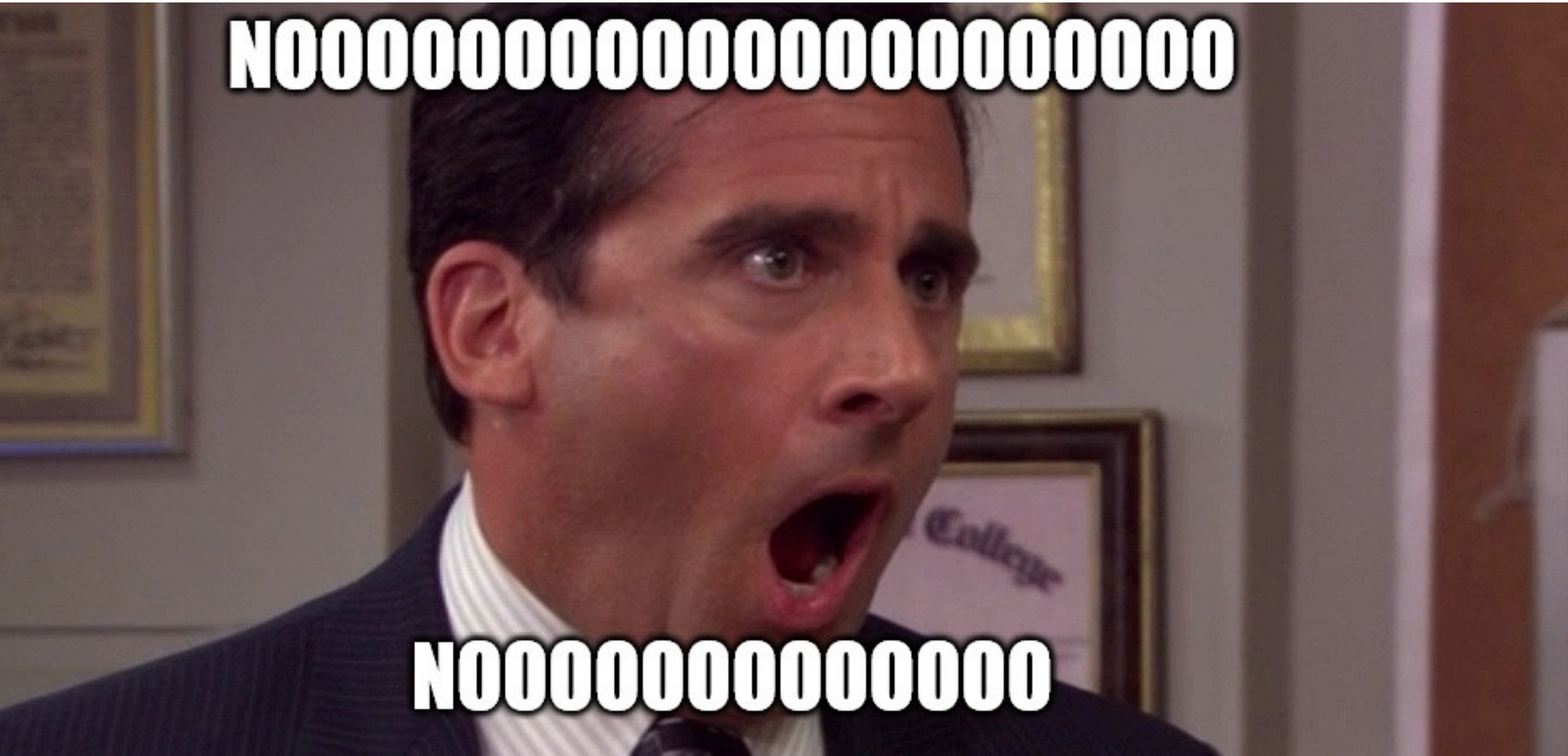
```
IF p_woonplaats IS NOT NULL OR
   p_straatnaam IS NOT NU
   p_postcode IS NOT NUL
   p_huisnummer IS NOT N
  dbms_sql.bind_variable                    er', p_huisnummer);
  dbms_sql.bind_variable                    ', upper(p_postcode));
  dbms_sql.bind_variable                    am', p_straatnaam);
  dbms_sql.bind_variable                    ts', p_woonplaats);
END IF;
dbms_sql.bind_variable(l                    , p_peildatum);
dbms_sql.bind_variable(l                    _offset);
IF p_limit <> 0 THEN
  -- Haal 1 rij extra op                    hasMore
  dbms_sql.bind_variable                    p_limit + 1);
END IF;
l_ret := dbms_sql.execute(l_cur_id);
c_rel := dbms_sql.to_refcursor(l_cur_id);
```



Gathering data do’s and don’ts

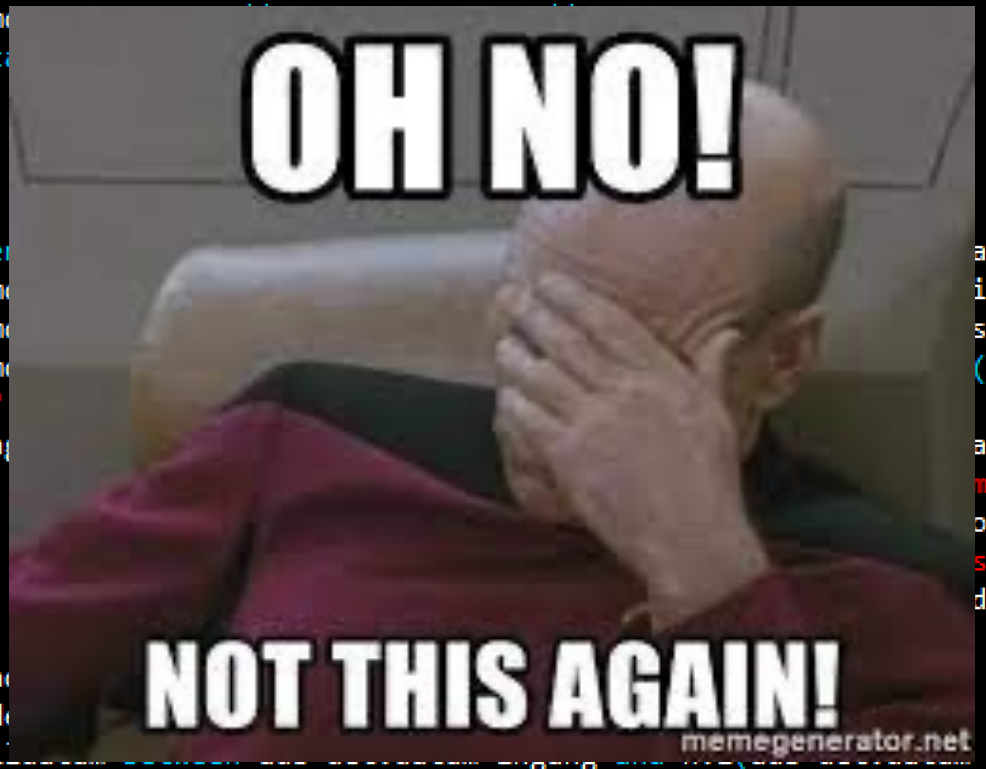
```
 l_rel_obj   := NEW json_object_t;
 l_ads_obj   := NEW json_object_t;
 l_bat_array := NEW json_array_t;
 rbl_audit.log_object(p_alg_id => p_alg_id
                     ,p_object => rbl_audit.c_audit_object_rbl_relaties
                      p_identificatie => p_rel_referentie);

rbl_igla_chec

rbl_igla_chec

                                 );

rbl_igla_check.put_json_value(i_igla_string => 'RBL_RELATIES_VW.DATUM_BEGIN'
                             ,i_naam        => 'DatumBegin'
                             ,i_waarde      => to_char(r_rel.datum_begin, util.datetime_format)
                             ,io_json_obj   => l_rel_obj
                             ,i_user        => p_current_user
                             );
```

Gathering data do’s and don’ts



Gathering data do's and don'ts

```
l_clob := '{"Status": "Ok", "Medewerkers":[';
for r_mdr in (select '{"id": ' || apex_json.stringify(mdr.id) || ', "naam": ' || apex_json.stringify(mdr.naam) || ', "sapNummer": ' ||
                    apex_json.stringify(mdr.sapnummer) || ', "datumInDienst": ' || apex_json.stringify(mdr.ingangsdatum) || ', "datumUitDienst
                    apex_json.stringify(m
                    apex_json.stringify(c
                                                                                                amen) || ', "voorletters": ' ||
                    apex_json.stringify(m                                                       ingify(mdr.voorvoegsel) || ', "netwerkUserna
                    apex_json.stringify(m                                                       stringify(mdr.geboortedatum) || ', "roepnaam
                    apex_json.stringify(m                                                       (mdr.emailadres) ||
                    (select ', "adressen"
                           (select lista                                                        adres)) || ', "straat": ' ||
                                                                                                mer": ' || apex_json.stringify(ads_det.huisn
                                                                                                peging) || ', "postcode": ' ||
                                                                                                s": ' || apex_json.stringify(ads_det.woonpla
                                                                                                det.telefoonnummer_mobiel, ads_det.telefoonn
                              from rmr_a
                             where ads_d
                               and l_pei                                                        einde, l_peildatum)) || ']'
                      from rmr_adressen_vw ads
                     where ads.mdr_id = mdr.id
                       and l_peildatum between ads.datum_ingang and nvl(ads.datum_einde, l_peildatum)
```




hear of something without knowing the origin

Gathering data do’s and don’ts

```
select d.deptno
      ,d.dname
      ,d.loc
from srs_dept_vw d
where upper(d.deptno || '#' || d.dname || '#' || d.loc) like upper('%' || :search || '%')
```

Gathering data do's and don'ts

```
procedure dsearch(p_deptno in varchar2
                 ,p_dname  in varchar2
                 ,p_items  out sys_refcursor) is
   lgr logger_oo := new logger_oo(p_scope => $$plsql_unit || '.search');
begin

   lgr.add_param(p_name => 'client_identifier', p_value => sys_context('userenv', 'client_identifier'));
   lgr.add_param(p_name => 'p_deptno', p_value => p_deptno);
   lgr.add_param(p_name => 'p_dname', p_value => p_dname);

   lgr.log_start;

   open dsearch.p_items for
      select d.deptno
            ,d.dname
            ,d.loc
            ,cursor (select e.empno
                           ,e.ename
                           ,e.job
                           ,e.mgr
                           ,e.hiredate
                           ,e.sal
                           ,e.comm
                           ,e.deptno
                     from   srs_emp_vw e
                     where  e.deptno = d.deptno) as employees
      from   srs_dept_vw d
      where  1 = 1
      and    (d.deptno = p_deptno or p_deptno is null)
      and    (upper(d.dname) like upper('%' || p_dname || '%') or p_dname is null);

   lgr.log_end;

end dsearch;
```

Gathering data do's and don'ts

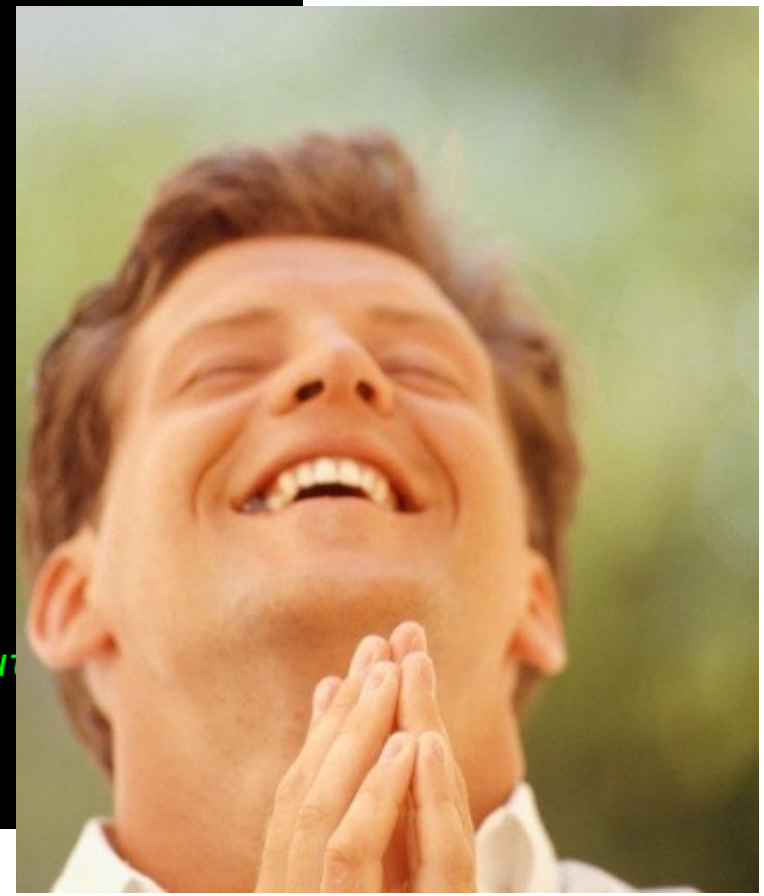
- Options to generate the JSON:
  - By the ORDS, we don't have to do anything!
    - SQL inside the handler
    - OUT parameter (SYS_REFCURSOR or PLSQL-Collection)
  - By using JSON generating functions in Oracle 12c and up
    - Your mime-type must be properly set
  - By using APEX_JSON package

Gathering data do’s and don’ts

```
select json_object (
         key 'departments' value (
           select json_arrayagg(
                    json_object (
                      key 'department_name' value d.dname,
                      key 'department_no' value d.deptno,
                      key 'employees' value (
                        select json_arrayagg (
                                 json_objec
                                   key 'emp
                                   key 'emp
                                 )
                               )
  from   emp e
  where  e.deptno = d
)


d

s
```

Gathering data do's and don'ts

```
declare
   l_cursor sys_refcursor;
begin

   open l_cursor for
      select e.empno  as "employee_number"
             ,e.ename  as "employee_name"
             ,e.deptno as "department_number"
      from   emp e
      where  rownum <= 2;

   --apex_json.initialize_clob_output;

   apex_json.open_object;
   apex_json.write('employees', l_cursor);
   apex_json.close_object;

   --dbms_output.put_line(apex_json.get_clob_ou
   --apex_json.free_output;
end;
```



Thanks

Any questions ?

You can find me at

- @rhjmartens
- rmartens@smart4solutions.nl