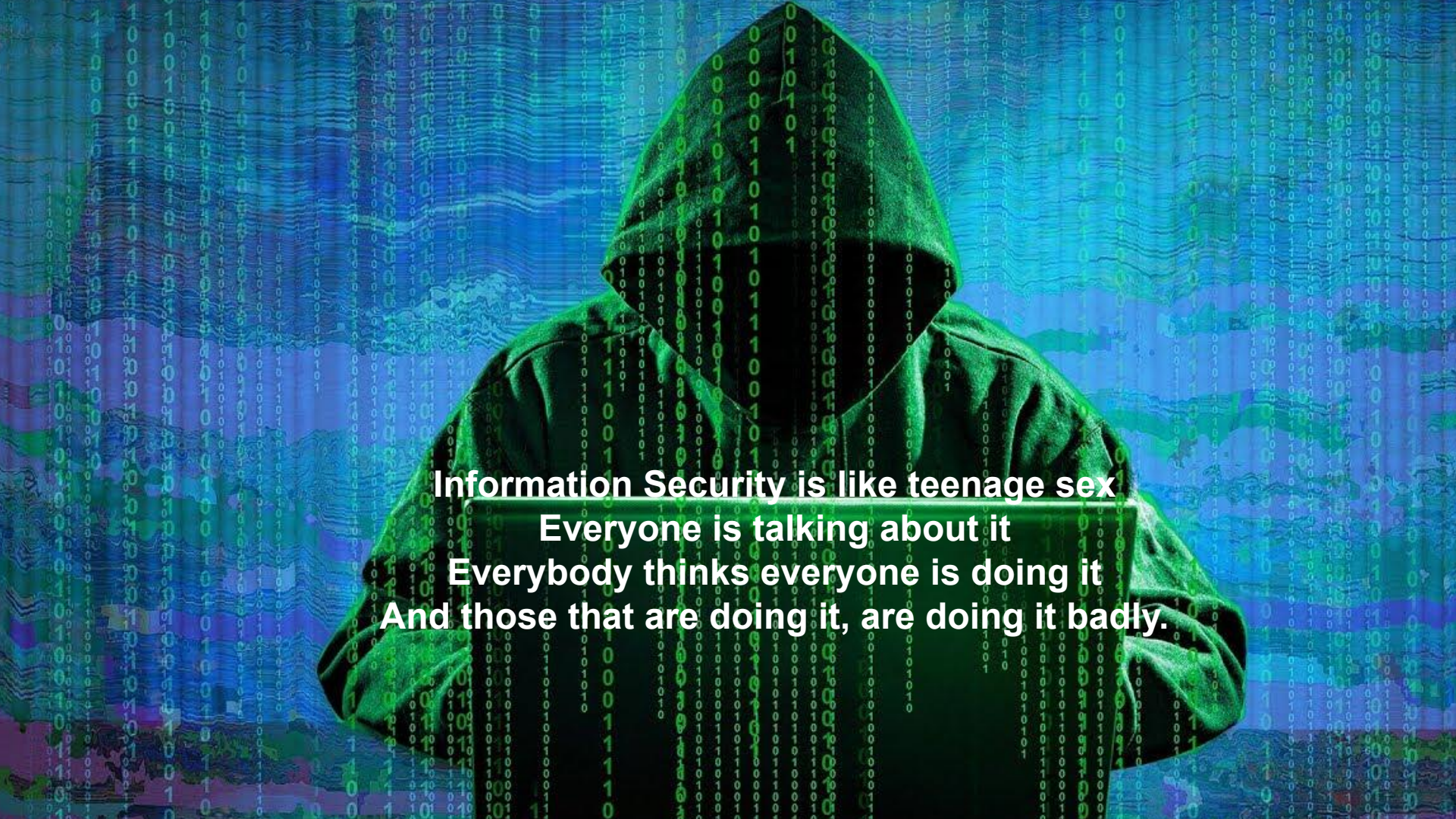
Secure your high performance code

ORACLE WIZARD .com

ORACLE ACE Director

@YourNavionPilot


Information Security is like teenage sex
Everyone is talking about it
Everybody thinks everyone is doing it
And those that are doing it, are doing it badly.


Anything can be hacked

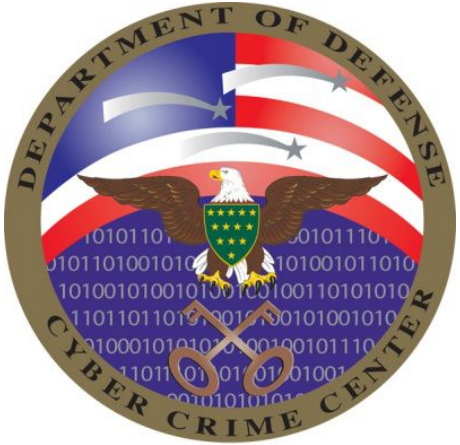

FINANCIAL CRIMES
ENFORCEMENT NETWORK

Who am I


ORACLE WIZARD
.com


ORACLE
ACE Director


ORACLE
Database Developer
Choice Awa
2015
Robert P. Lockard
Database Design


DEPARTMENT OF DEFENSE
CYBER CRIME CENTER


MARYLAND
DR EVIL
Treasure the Chesapeake

Important Terms

SQL Injection: Change a sql statement so it executes code that was not intended. Think "Change the code execution path."

Hack: Anything can be hacked. Do something it was not intended to do or something you did not think it could do.

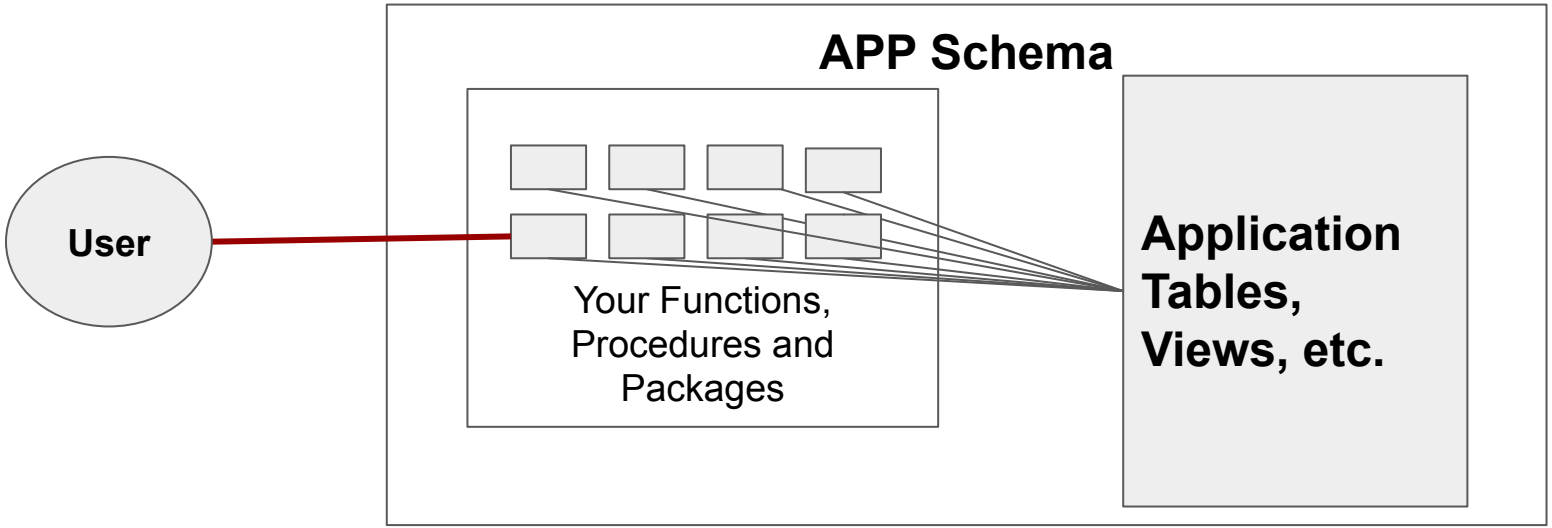
Attack Surface: Any node on the network can be attacked. That can be the UI, People, anything that touches data.

Exploit: Take advantage of a flaw or feature

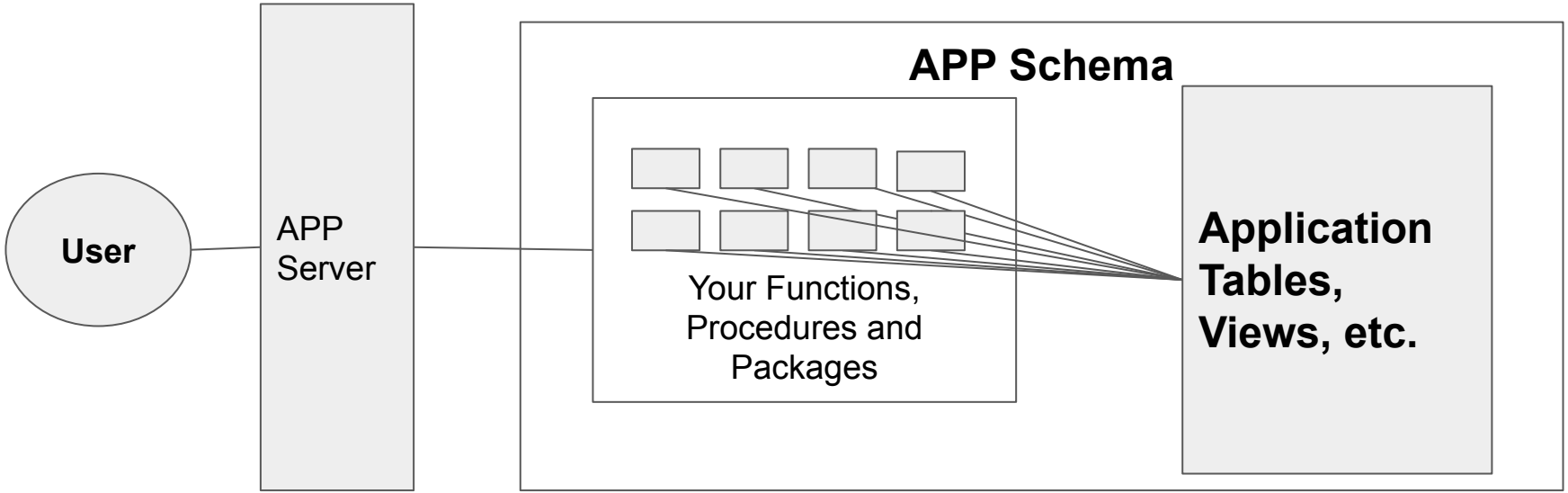
Spill: When data leaves it's protected environment. It may or may not be compromised

Leak: Data has left it's protected environment and has been compromised.

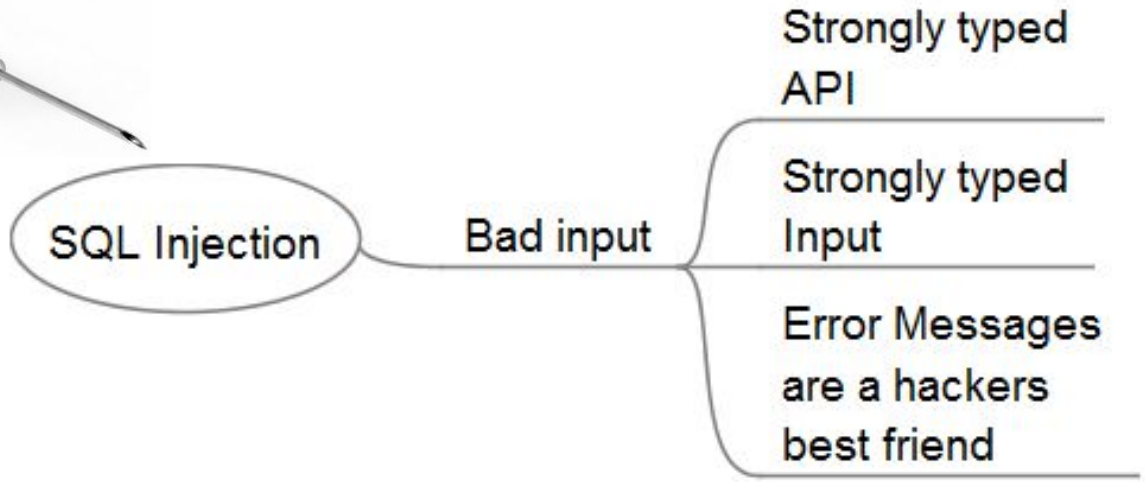
Everywhere I go.



Everywhere I go.



SQL Injection

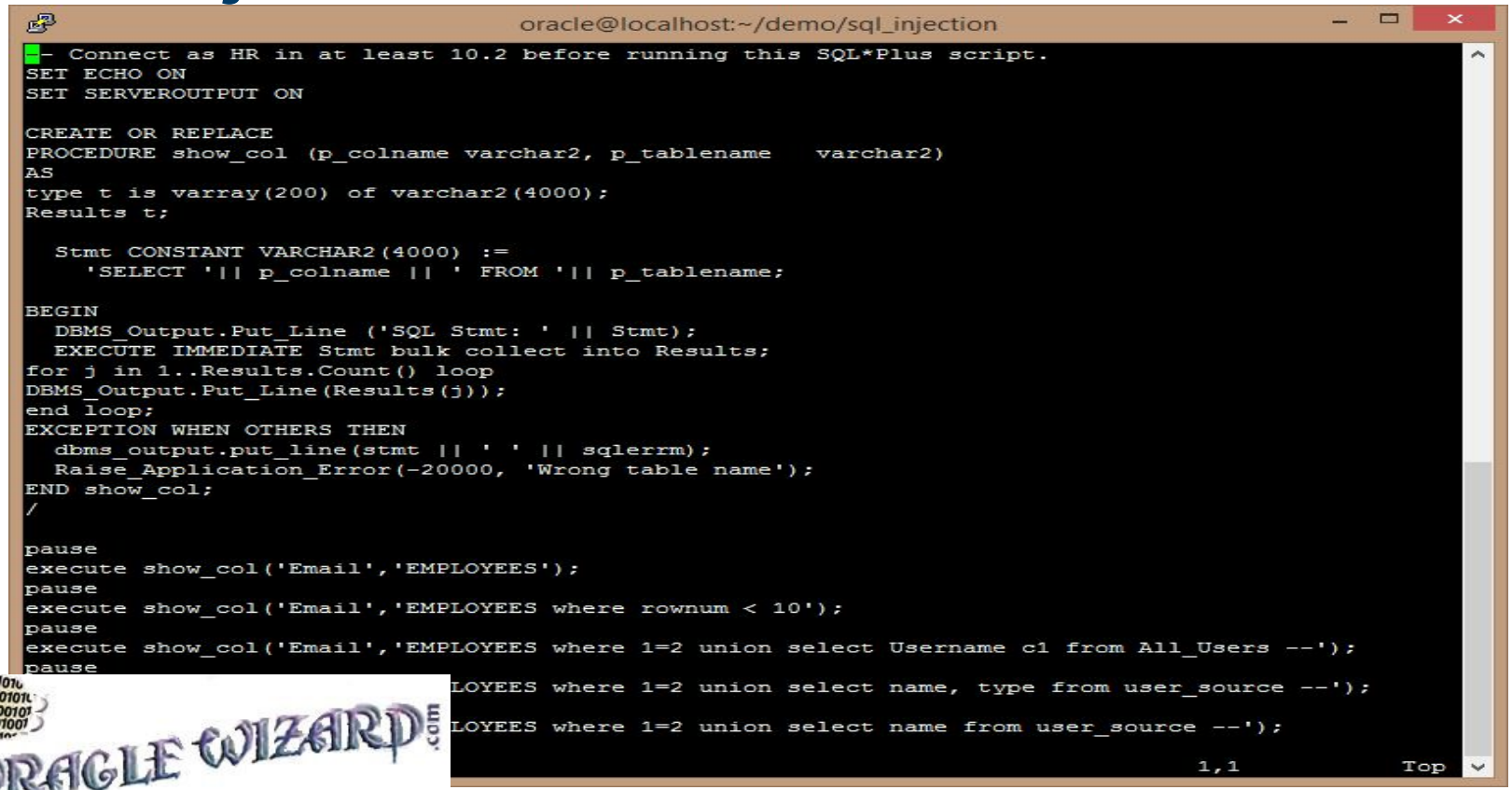


SQL Injection Demo

Create a procedure that is full of sql injection bugs.

- **concatenate strings into a sql statement.**

SQL Injection Demo



PL/SQL Secure Coding SQL Injection, extracting your source code

Not only am I getting your source code, I'm also getting all of your comments.

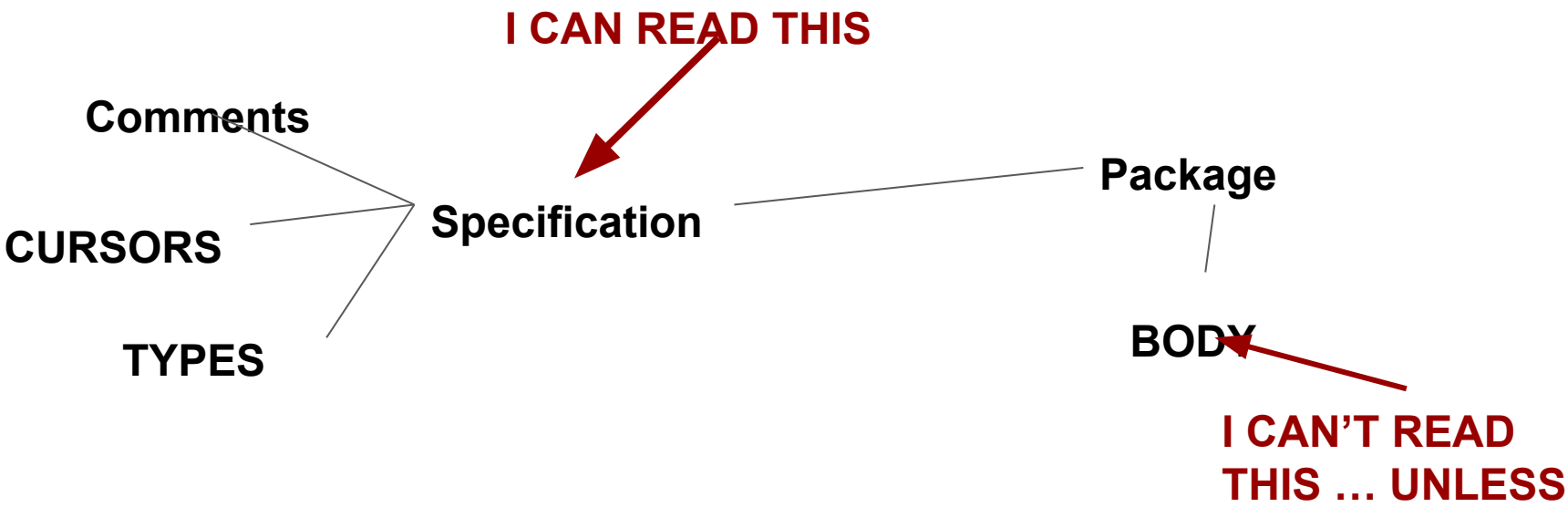
Demo - Extract from all_tables
Demo - Extract from all_source
Demo - Extract Packages

PL/SQL Secure Coding SQL Injection, extracting your source code

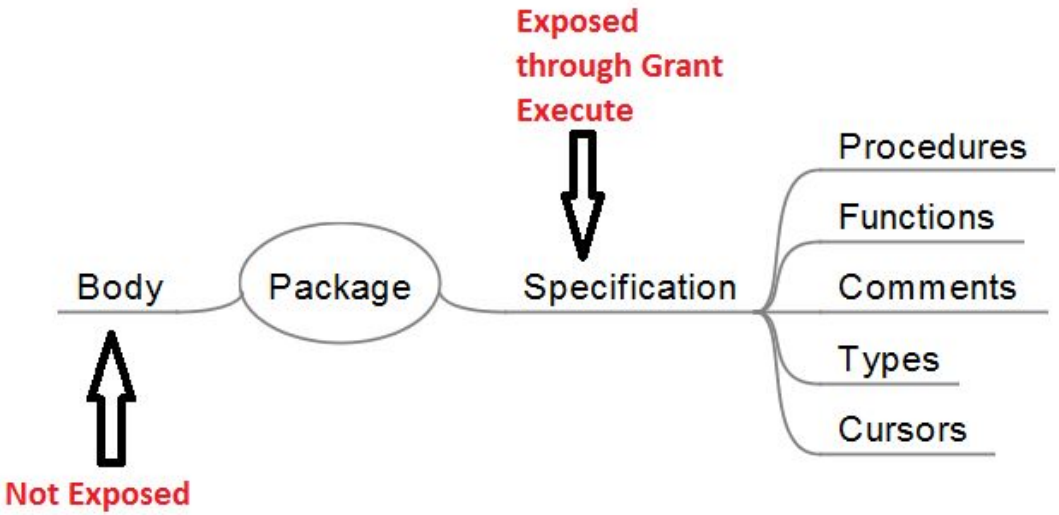
Not only am I getting your source code, I'm also getting all of your comments.

- **we are only getting the package specification.**
- **we still have copious comments in the package specification. I can use those comments to derive how your package works.**
- **we have cursor and type definitions in the package specification. Do you really need to have those in the specification? Sometime the answer is going to be yes, other times no. I always weight the moving cursors and type definitions to the package body, then go through the thought process to justify moving into the specification.**

Packages



Packages


Exposed
through Grant
Execute
Procedures
Functions
Comments
Types
Cursors
Body
Package
Specification
Not Exposed

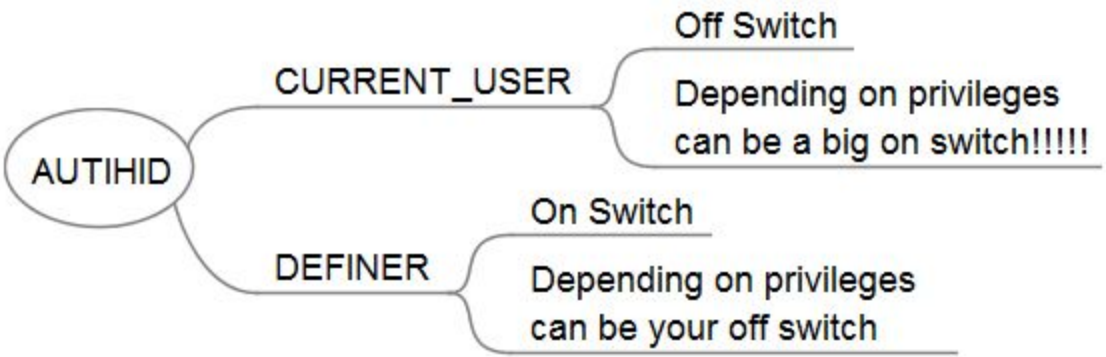
Packages

This comment is great BUT it still gives the bad guys too much information

```
84
85  create or replace PACKAGE INMS_APP.REPORTTARGET
86  AUTHID current_user
87  --ACCESSABLE BY (INMS_APP.IRP_READPRISMFILES)    -- this is a version 12c feature. implement when move to 12c.
88  AS
89      -- constant delorations
90      lVersion    CONSTANT VARCHAR2(10) := '20161026.1';
91
92      FUNCTION MAIN RETURN NUMBER;
93      FUNCTION WHAT_VERSION RETURN VARCHAR2;
94
95  END REPORTTARGET;
96  /
```

IS AUTHID CURRENT_USER A GOOD IDEA?

What happens with authid current_user is abused?
revoke inherit privileges on user <USER> from public;



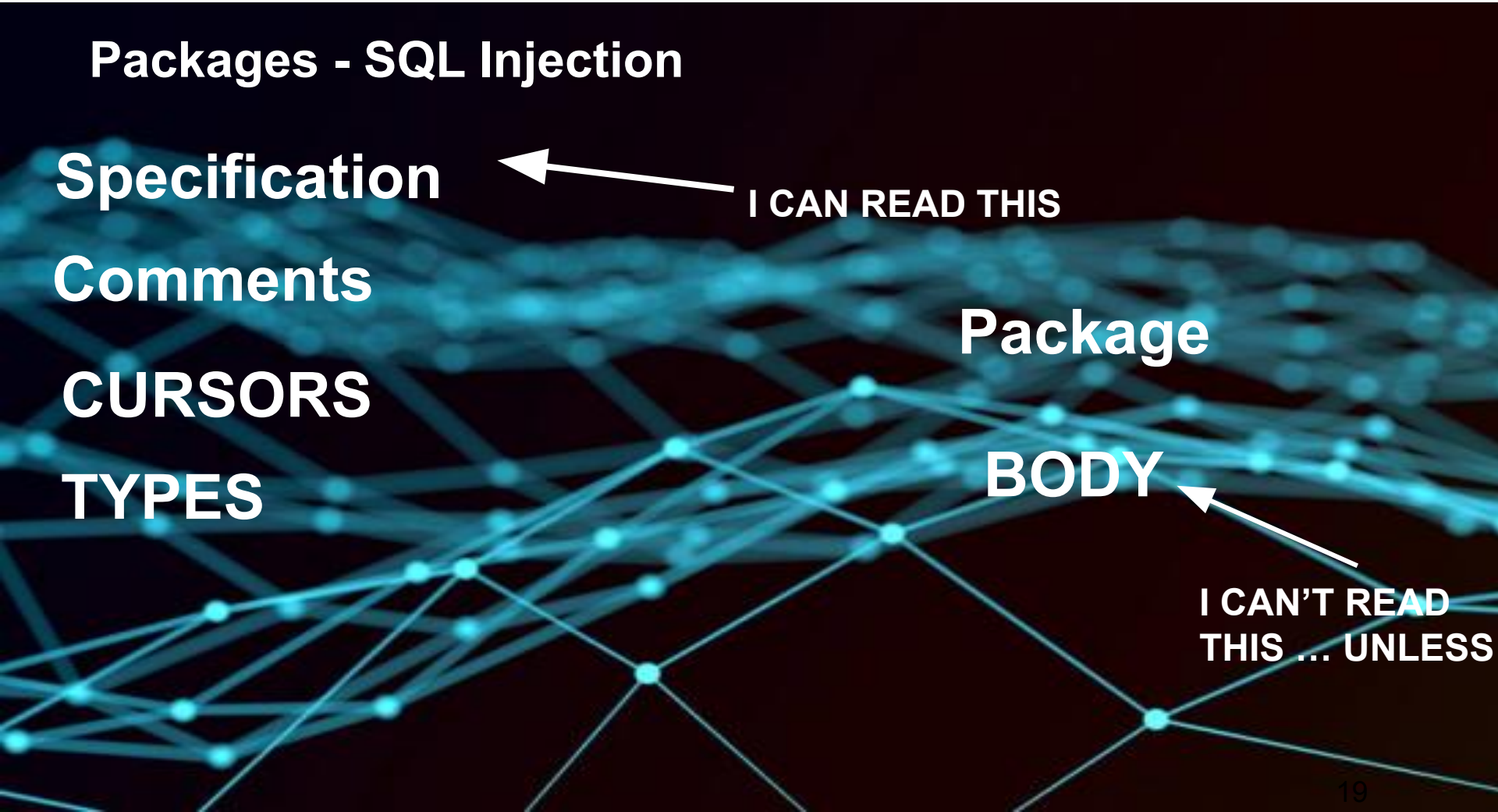
Schema Only account

Connect by proxy

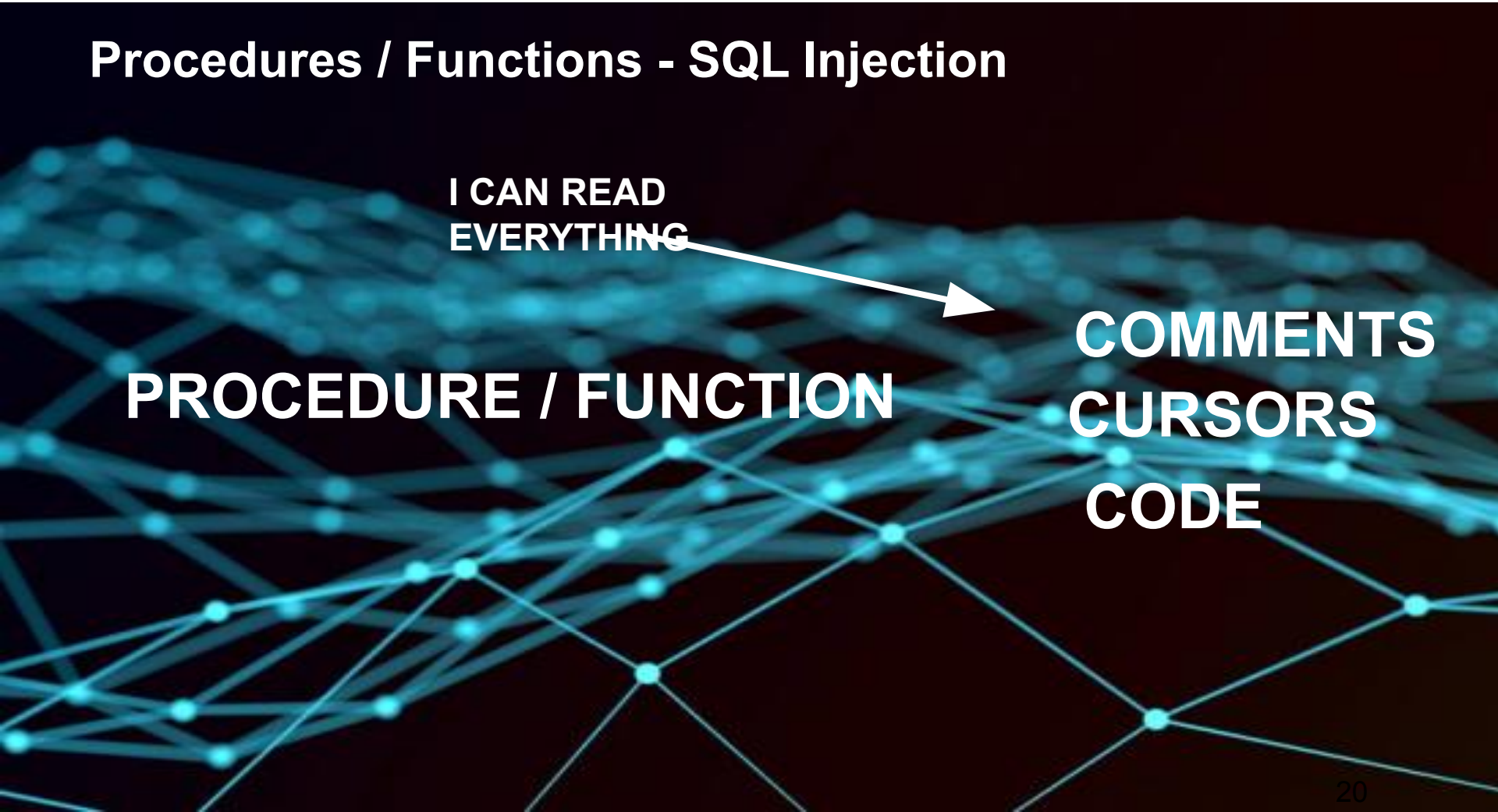
```
create user <username> no authentication

grant create session, create procedure,
create table to <username>;

alter user <username> grant connect
through <app_dba_account>;

conn rlockard[test_data1]@orclpdb1
```

INHERIT PRIVILEGES

Packages - SQL Injection



Procedures / Functions - SQL Injection



Packages - SQL Injection

```
84
85  create or replace PACKAGE INMS_APP.REPORTTARGET
86  AUTHID current_user
87  --ACCESSABLE BY (INMS_APP.IRP_READPRISMFILES)    -- this is a version 12c feature. implement when move to 12c.
88  AS
89      -- constant delorations
90      lVersion    CONSTANT VARCHAR2(10) := '20161026.1';
91
92      FUNCTION MAIN RETURN NUMBER;
93      FUNCTION WHAT_VERSION RETURN VARCHAR2;
94
95  END REPORTTARGET;
96  /
```

SYNONYMS

USE SYS.DBMS_OUTPUT

Limit the number of ways to get to your data.
Trusted Path. The wrong way to do it.

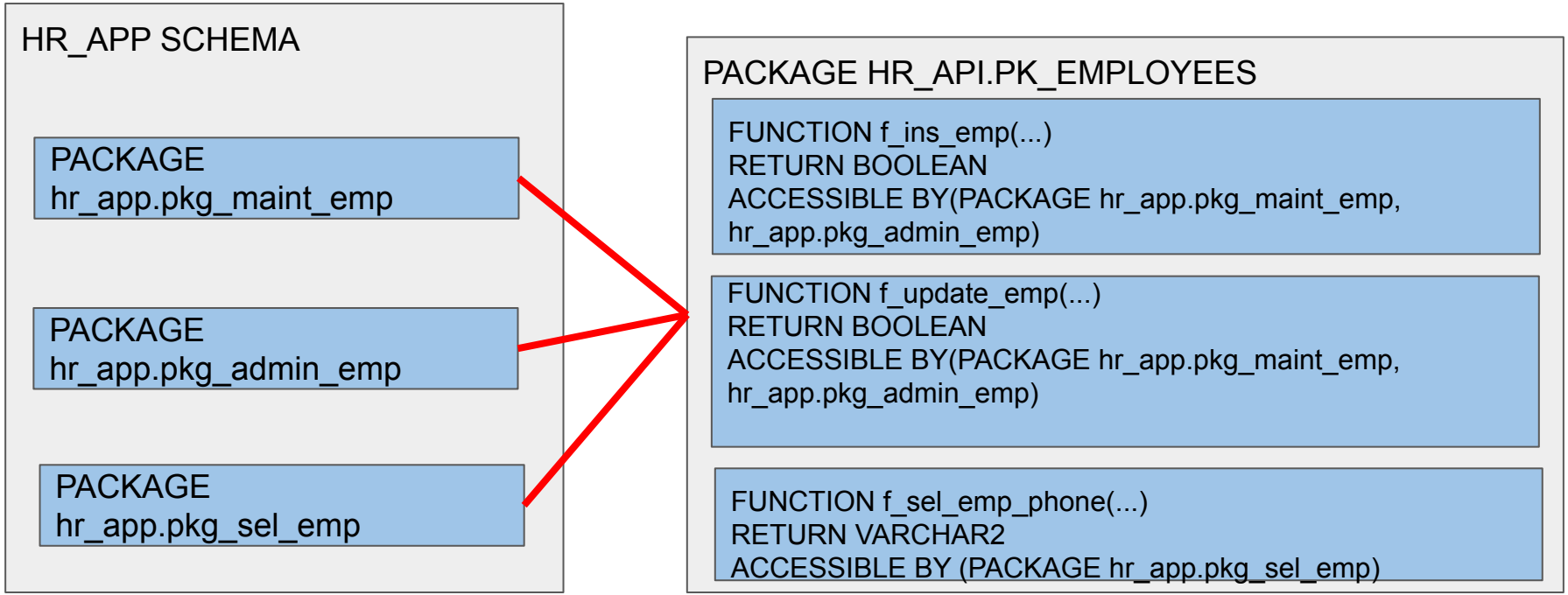
Because this is a procedure a hacker can extract your code.

Because authid not specified the procedure is created with definer rights

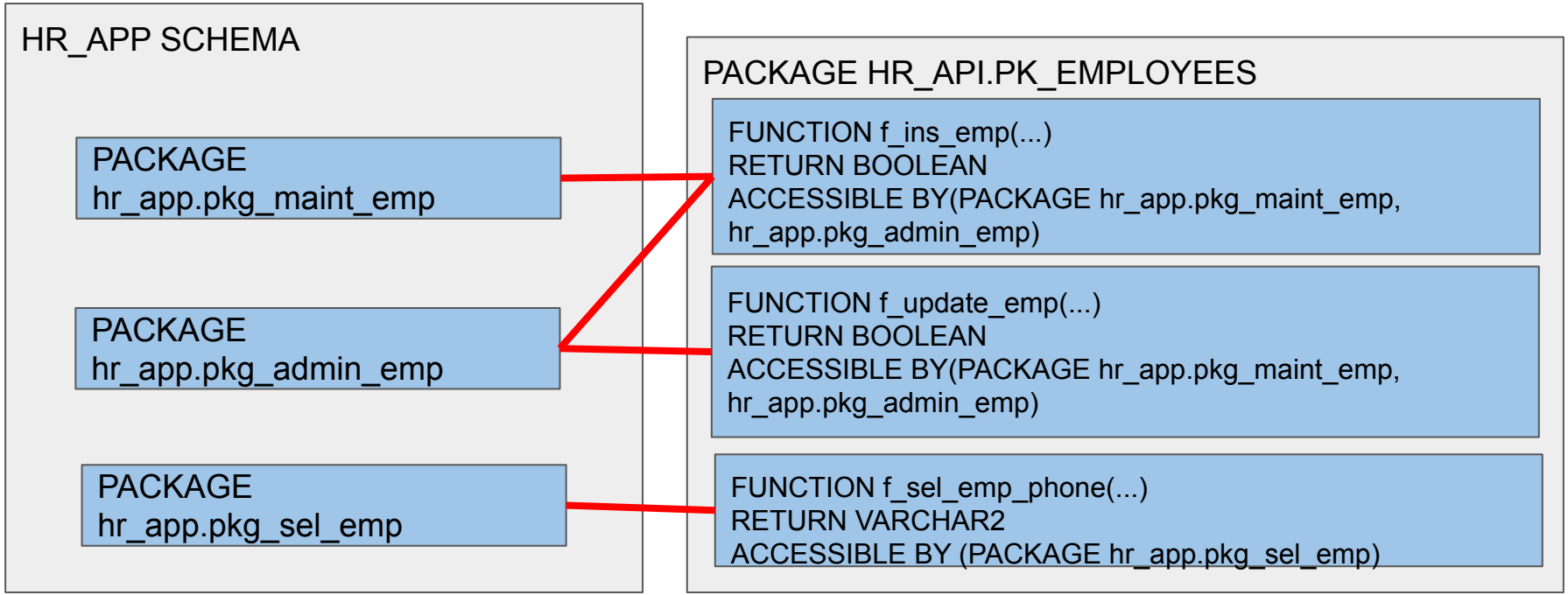
```
005 CREATE OR REPLACE PROCEDURE process_one_row (table_in   IN VARCHAR2,
006                                              where_in   IN VARCHAR2)
007 IS
008    l_block   VARCHAR2 (32767)
009       :=    'DECLARE l_row '
010          || table_in
011          || '%ROWTYPE; BEGIN SELECT * INTO l_row FROM '
012          || table_in
013          || ' WHERE '
014          || where_in
015          || '; /* Then do stuff with that row... */'
016          || ' END;';
017 BEGIN
018
019    DBMS_OUTPUT.put_line (l_block);
020
021    EXECUTE IMMEDIATE l_block;
022 EXCEPTION
```

Huge SQL Injection Bug

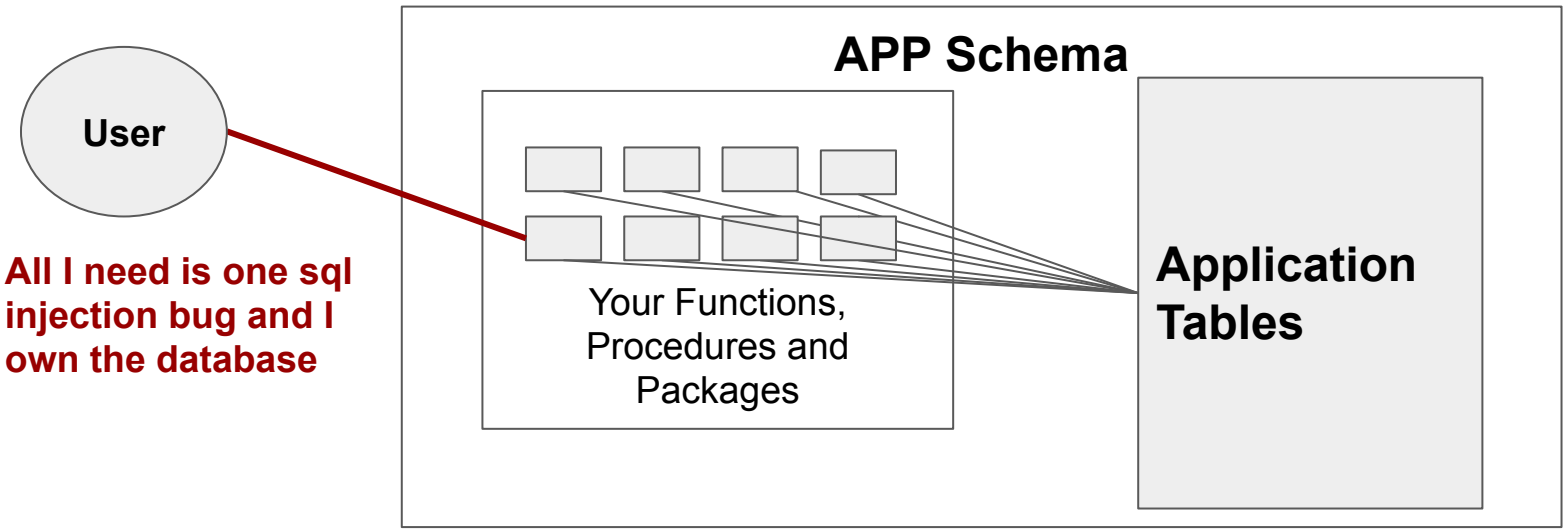
ACCESSIBLE BY 12.1



ACCESSIBLE BY 12.2 ENHANCEMENTS

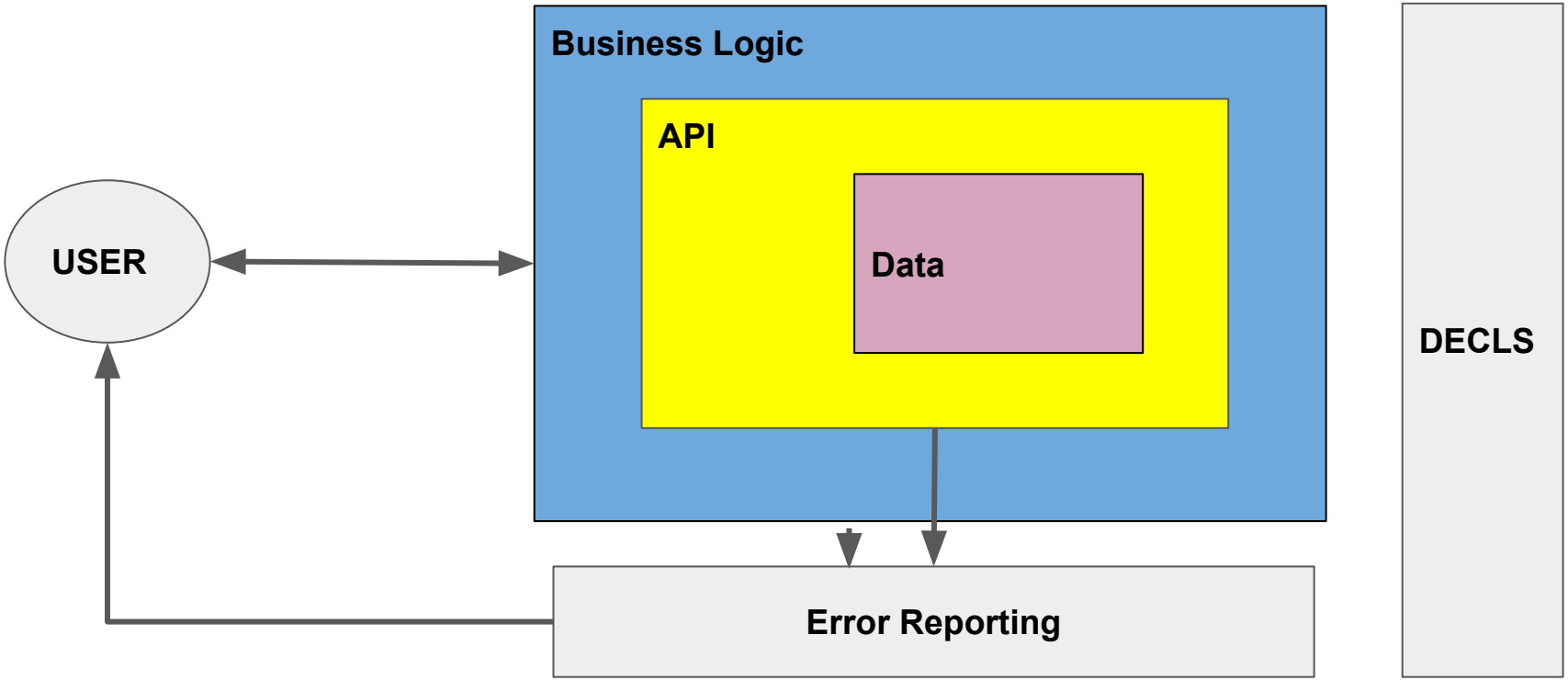


Limit the number of ways to get to your data. Trusted Path. The wrong way to do it.



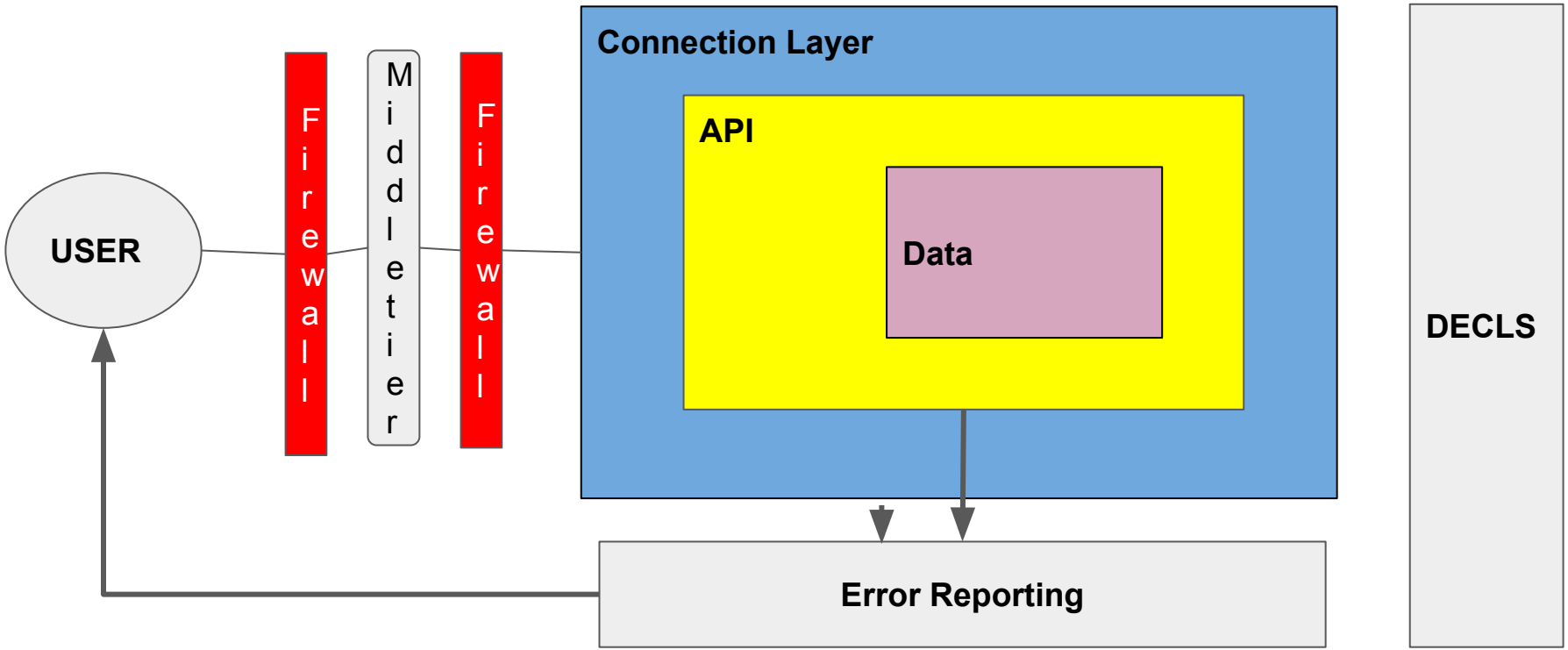
All I need is one sql injection bug and I own the database

Secure Architecture 1



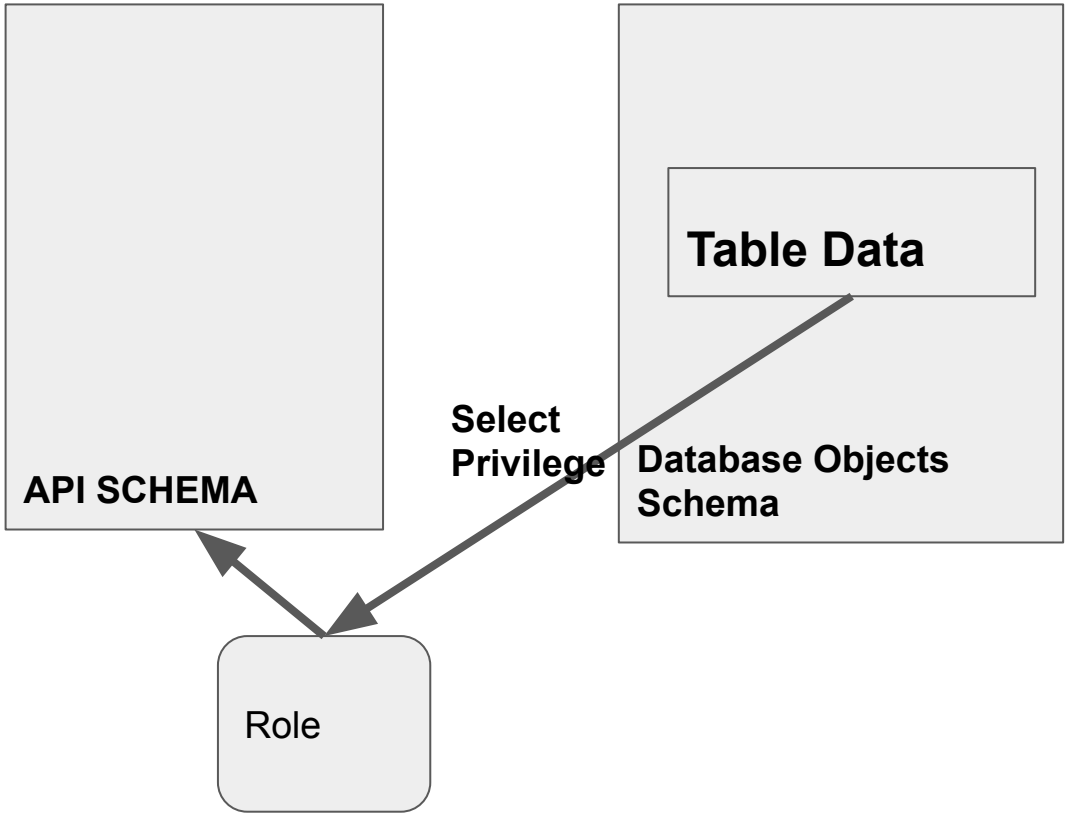
Based on the original work of Bryn Llewellyn

Secure Architecture 2

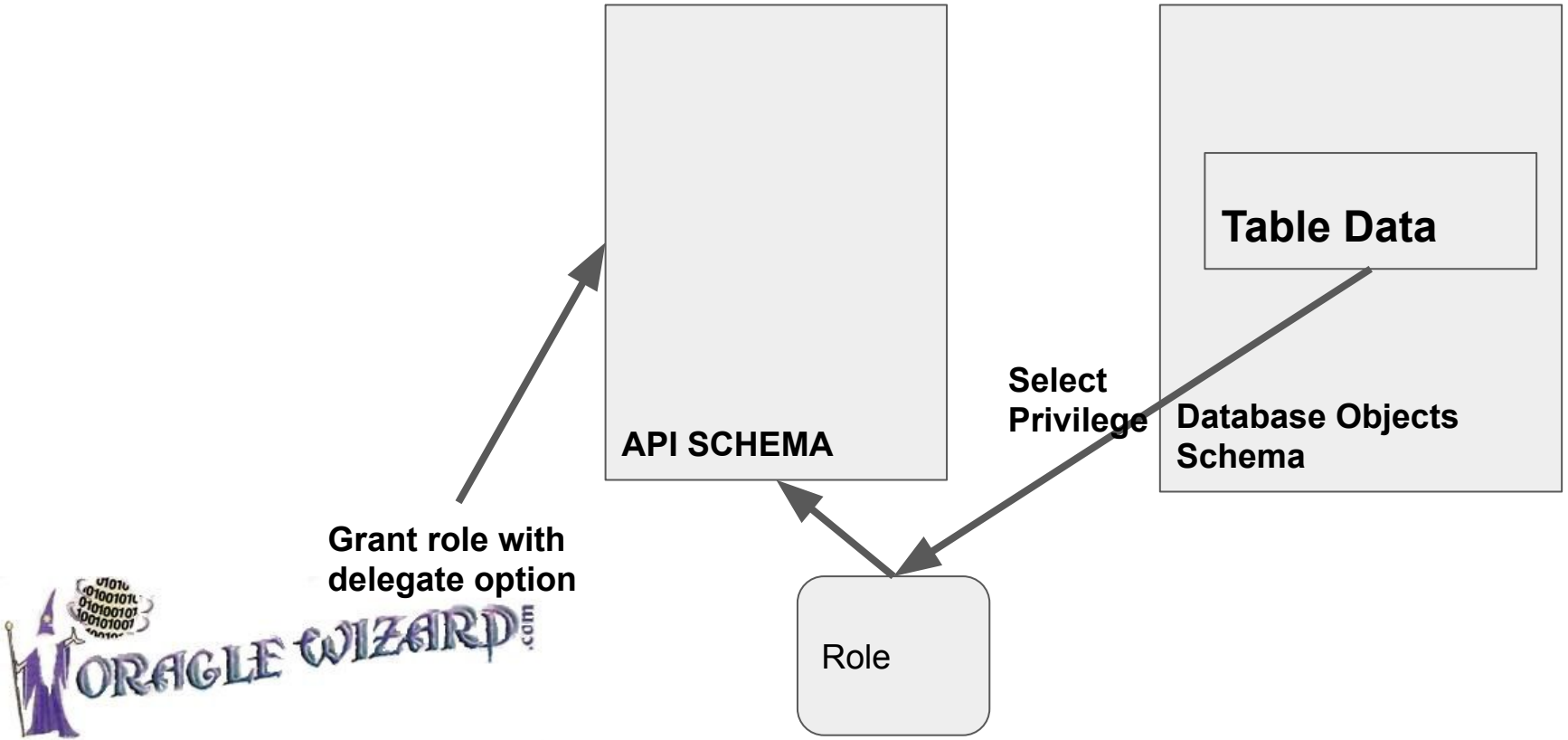


Based on the original work of Bryn Llewellyn

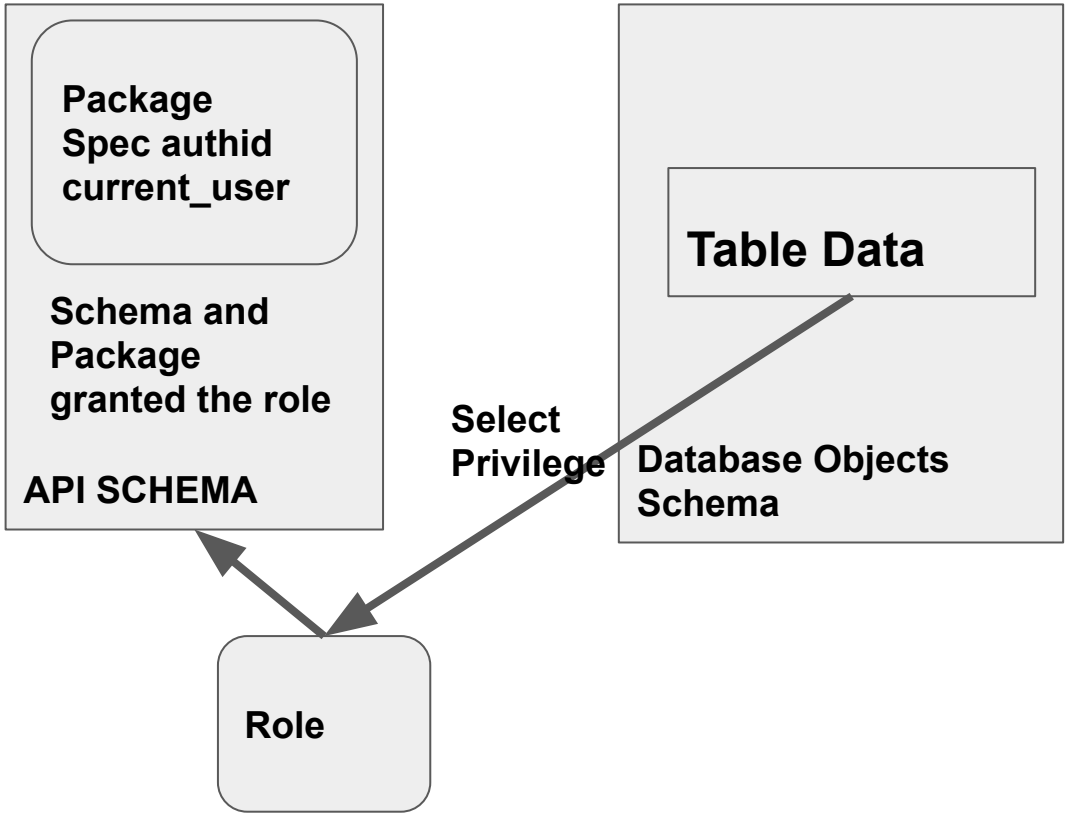
Code Based Access Control



Code Based Access Control

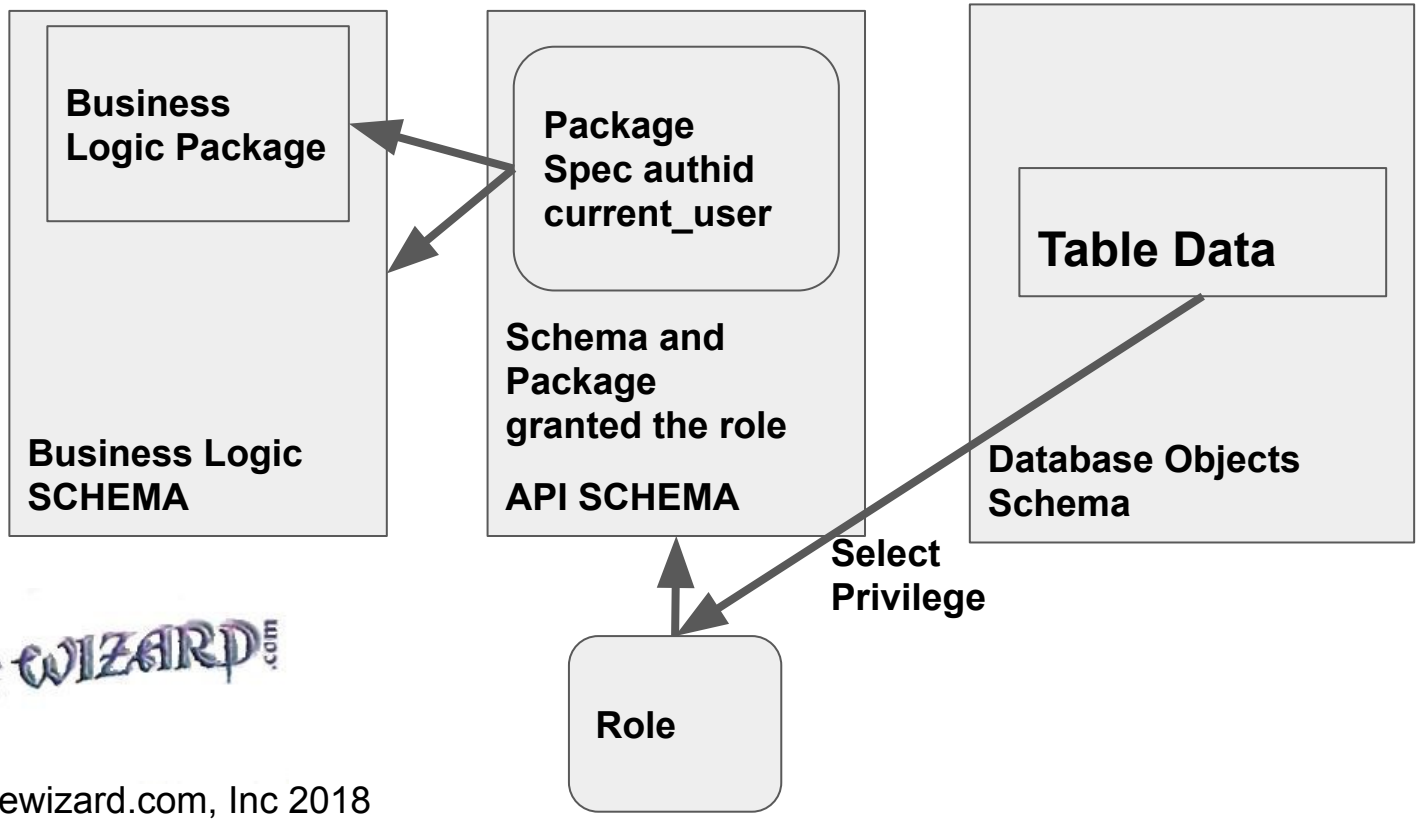


Code Based Access Control

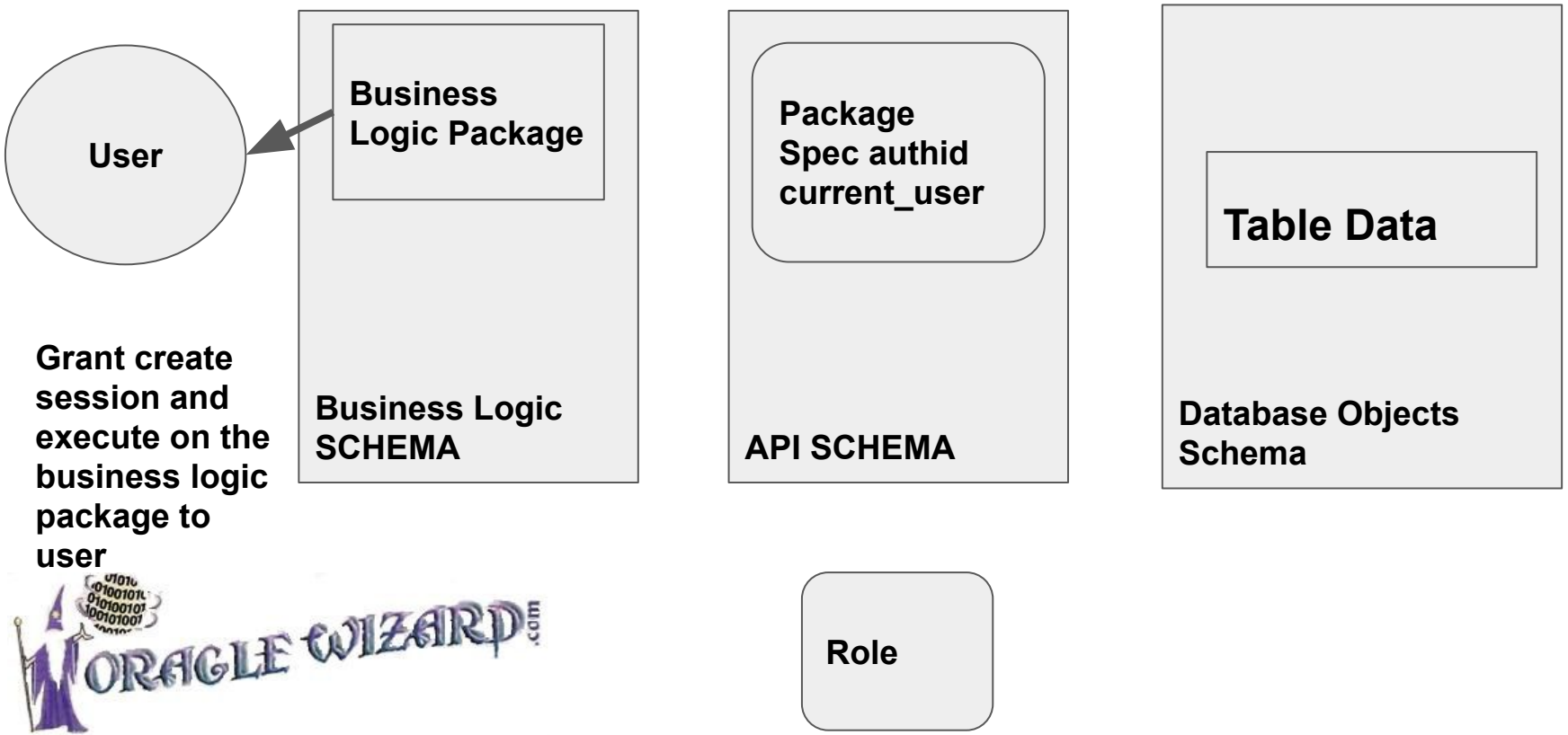


Code Based Access Control

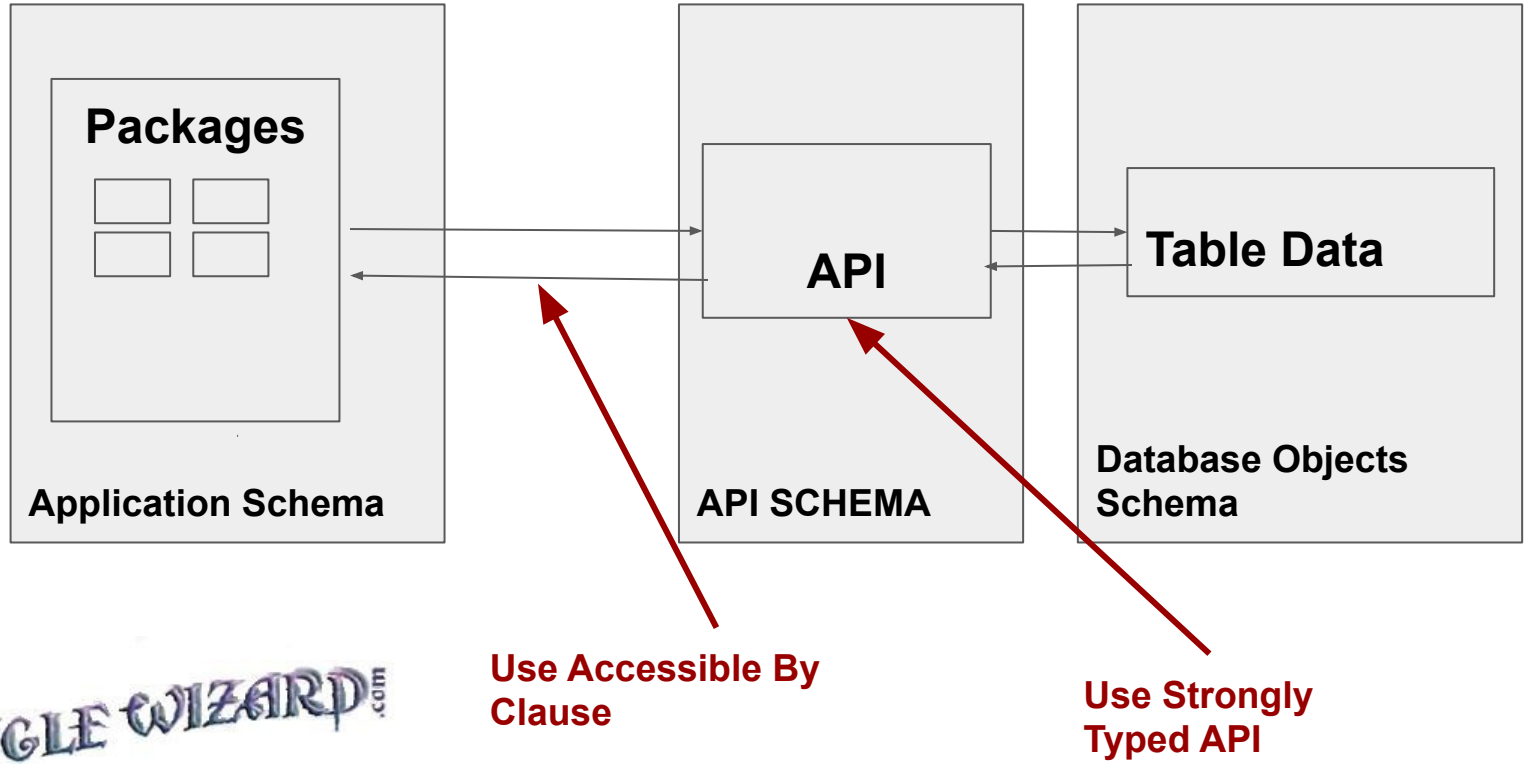
Grant execute on package to Business logic shema and accessible by to the business logic package.procedure



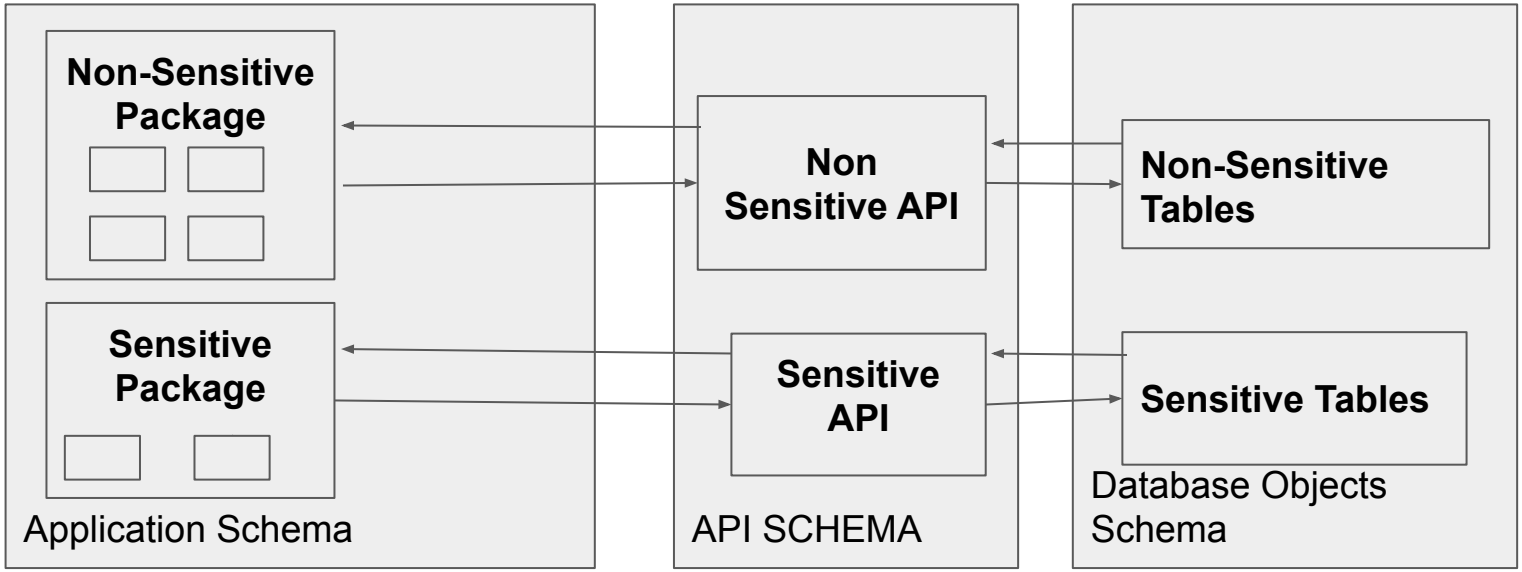
Code Based Access Control



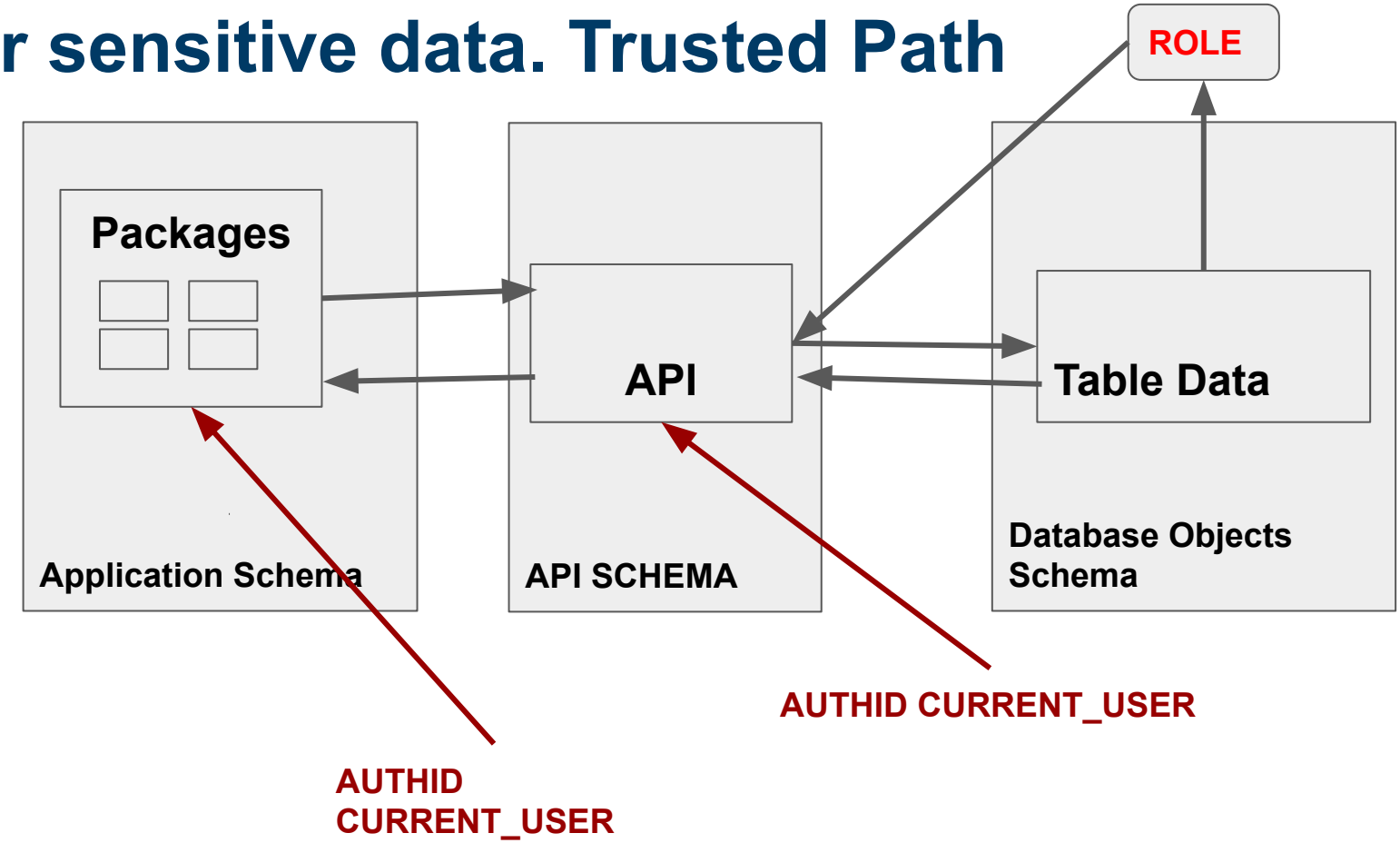
Limit the number of ways to get to your sensitive data. Trusted Path



Split sensitive and non sensitive packages



Limit the number of ways to get to your sensitive data. Trusted Path



Code Based Access Control

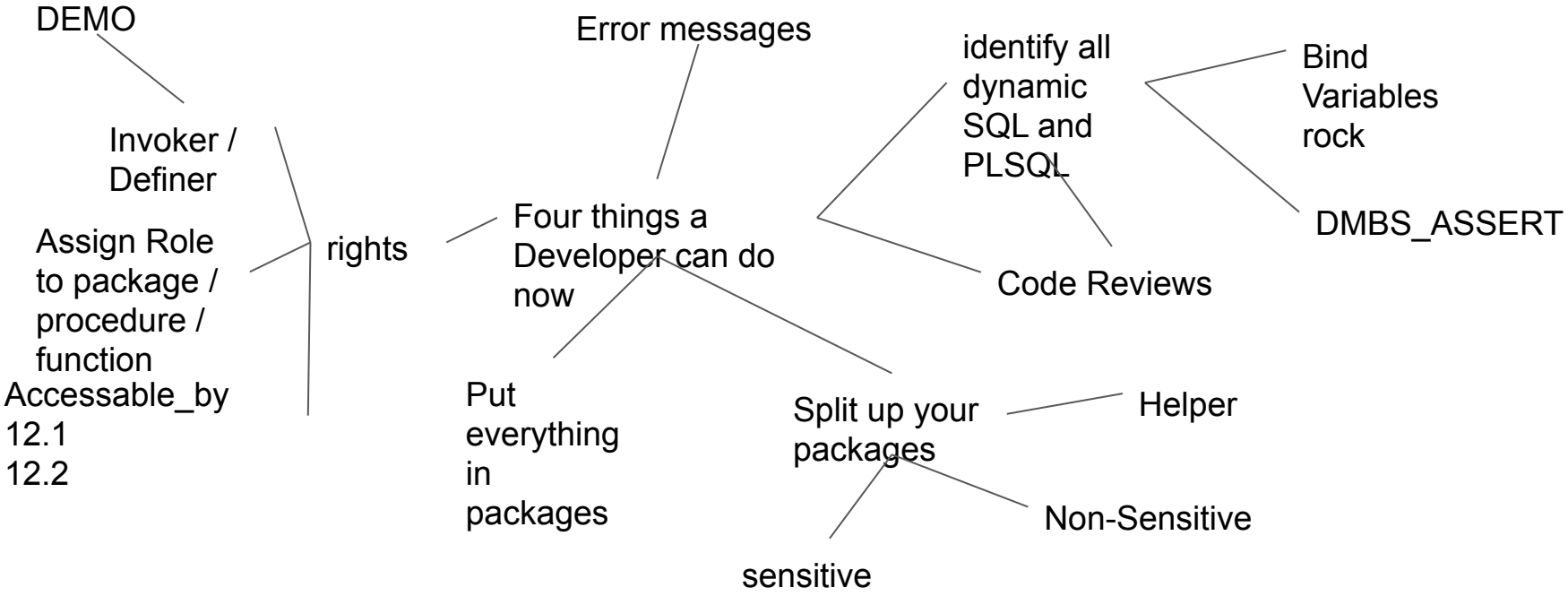
Error Messages

```
EXCEPTION WHEN OTHERS THEN
  error_log.errorstack_pkg.pMain(pErrorId => iErrorId);
  raise_application_error(-20000, 'An error was raised, check log # ' || iErrorId);
END;
```

Putting it all together

4 things developers can do now to improve security… in process …..

Oooopsy, I lied, there are more than 4 things. :-)



Contact Information

Robert P. Lockard
Oraclewizard, Inc.
Hubzone Certified
Small Veteran Owned Business
Glen Burnie, MD
USA

email: rob@oraclewizard.com
twitter: @YourNavionPilot
blog: www.oraclewizard.com
youtube: www.youtube.com/user/n4281k