VIRTUAL PRIVATE DATABASE

Mathias Magnusson, Evil Ape

2018-11-11

ME

- Mathias Magnusson
- Founder of Miracle Sweden
- My company is Evil Ape - mid 2018
- Founder of SWEOUG
  Sweden Oracle User Group
- oradbdev.mathiasmagnusson.com
  Twitter mathiasmag
  Twitter & IG evilapehq
- Oracle ACE Associate

500+ Technical Experts Helping Peers Globally

ORACLE ACE Program


ORACLE ACE Director


ORACLE ACE


ORACLE ACE Associate

---

3 Membership Tiers

- Oracle ACE Director
- Oracle ACE
- Oracle ACE Associate

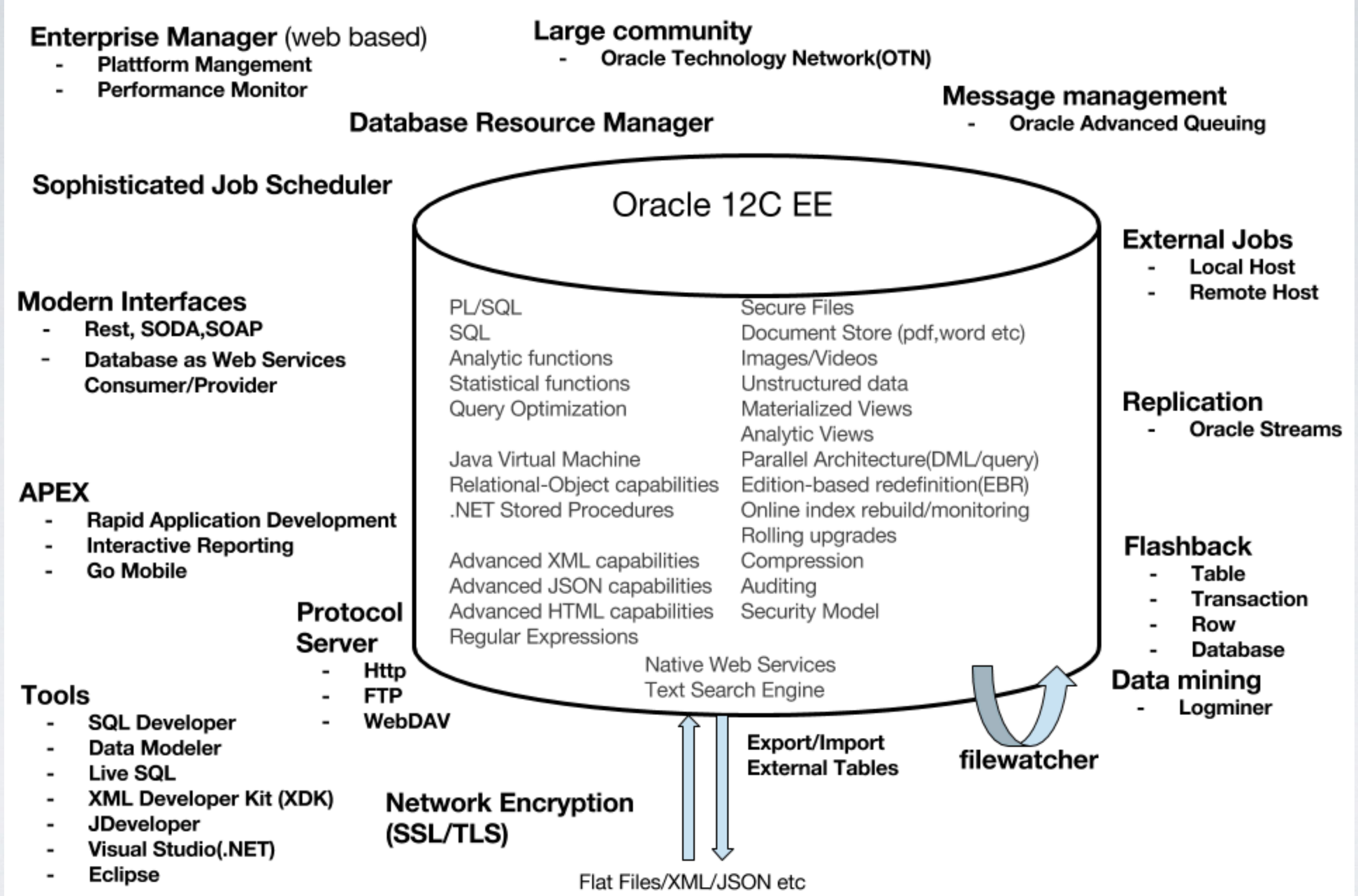
bit.ly/OracleACEProgram

Connect:

- oracle-ace_ww@oracle.com
- Facebook.com/oracleaces
- @oracleace


Oracle Groundbreakers

Nominate yourself or someone you know: acenomination.oracle.com


Enterprise Manager (web based)
- Plattform Mangement
- Performance Monitor
Large community
- Oracle Technology Network(OTN)
Message management
- Oracle Advanced Queuing
Database Resource Manager
Sophisticated Job Scheduler
Oracle 12C EE
PL/SQL
SQL
Analytic functions
Statistical functions
Query Optimization
Java Virtual Machine
Relational-Object capabilities
.NET Stored Procedures
Advanced XML capabilities
Advanced JSON capabilities
Advanced HTML capabilities
Regular Expressions
Secure Files
Document Store (pdf,word etc)
Images/Videos
Unstructured data
Materialized Views
Analytic Views
Parallel Architecture(DML/query)
Edition-based redefinition(EBR)
Online index rebuild/monitoring
Rolling upgrades
Compression
Auditing
Security Model
Native Web Services
Text Search Engine
External Jobs
- Local Host
- Remote Host
Modern Interfaces
- Rest, SODA,SOAP
- Database as Web Services Consumer/Provider
Replication
- Oracle Streams
APEX
- Rapid Application Development
- Interactive Reporting
- Go Mobile
Flashback
- Table
- Transaction
- Row
- Database
Protocol Server
- Http
- FTP
- WebDAV
Data mining
- Logminer
Tools
- SQL Developer
- Data Modeler
- Live SQL
- XML Developer Kit (XDK)
- JDeveloper
- Visual Studio(.NET)
- Eclipse
Export/Import
External Tables
filewatcher
Network Encryption
(SSL/TLS)
Flat Files/XML/JSON etc
Transportable Tablespaces, Automatic Storage Manager,
Parallel backup/recovery
© 2016 by Daniel Ekberg

WHAT IS VPD

Security policies controlling access to:

- Rows
- Columns

In English - Add a where clause dynamically

Affects

- Tables
- Views
- Synonyms

THE CONFUSION

VPD Stands for Virtual Private Database

But it is - more or less - the same thing as

FGAC - Fine Grained Access Control

RLS - Row Level Security

"By applying *fine grained access controls* to the database, you in effect, create a *virtual private database* to individual users. They can only see subsets of the data by using *row level security*."

–Connor McDonald - December 02, 2016

https://asktom.oracle.com/pls/apex/f?p=100:11:0::::P11_QUESTION_ID:9532605800346121507

… WAIT - THERE IS MORE …

OLS - Oracle Label Security

Database Vault

Both are solutions using RLS…FGAC…VPD

AND WHILE WE'RE AT IT

VPD is a poor acronym

Virtual - no not at all, not as in virtualisation

Private - well it is still in the same shared database

Database - true, but no more or less than w/o VPD

Virtually private? - kind of, but rarely used that way

LICENSE

Included with Oracle EE since 8i

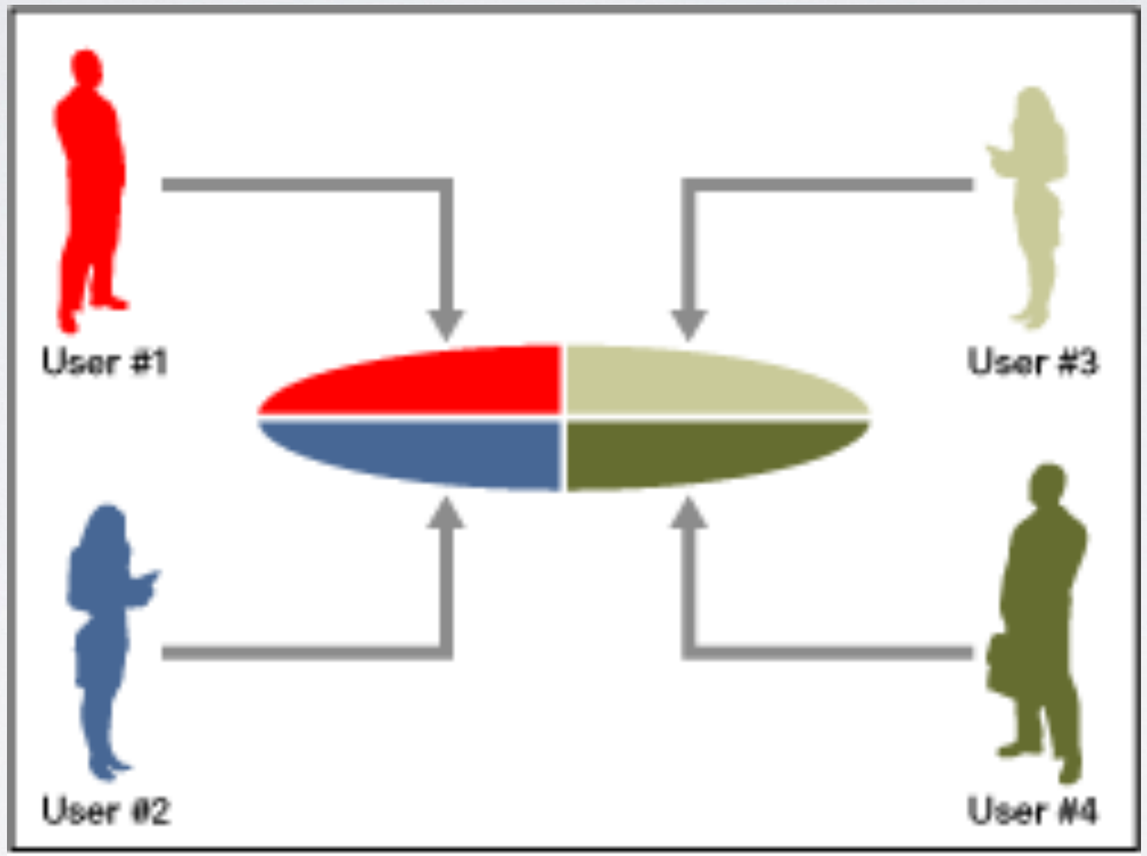
Not included with SE/SE1/SE2

LETS DIG IN

Discuss what it is and how it can be used

Finish with demonstration

HOW IT WORKS



Credit: Oracle

HIGH LEVEL BENEFITS

Security - Avoid incorrect modifications

Simplicity - Add once to a table and done

Flexibility - Can have multiple policies on one table

Result -> Data Integrity

PARTS OF VPD PROTECTION

A Table

A Function returning a condition

A policy definition

THE TABLE

```
create table vpd_test
    (vpd_id       number         not null
    ,info_text    varchar2(50)   not null
    ,owning_user  varchar2(8)    not null);
insert into vpd_test(1, 'Row 1, usrA', ‘USRA’);
insert into vpd_test(2, 'Row 2, usrA', ‘USRA’);
insert into vpd_test(3, 'Row 3, usrB', ‘USRB’);
insert into vpd_test(4, 'Row 4, usrB', ‘USRB’);
grant select on vpd_test to usra,usrb;
```

Two users can see all data in the table

THE FUNCTION

```
create function vpd_cond
          (p_in_schema in varchar2
          ,p_in_table  in varchar2) return varchar2 is
begin
  return q'#owning_user = '#' || user || '';
end vpd_cond;
```

A function that returns a condition.

Hardcoded literal for user owning the row.

THE POLICY

```
dbms_rls.add_policy
   (object_schema   => 'vpd_mgmt'
   ,object_name     => 'vpd_test'
   ,policy_name     => 'vpd_cond'
   ,function_schema => 'vpd_mgmt'
   ,policy_type     => dbms_rls.context_sensitive
   ,policy_function => 'vpd_mgmt.vpd_cond'
   ,statement_types => 'select,insert,update,delete');
```

Attach policy to an object referring to the function.

RESULT

With the policy in place

- User usrA can see row 1 and 2 only
- User usrB can see row 3 and 4 only
- User vpd_mgmt can see no rows at all

```
insert into vpd_test(1, 'Row 1, usrA', ‘USRA’);
insert into vpd_test(2, 'Row 2, usrA', ‘USRA’);
insert into vpd_test(3, 'Row 3, usrB', ‘USRB’);
insert into vpd_test(4, 'Row 4, usrB', ‘USRB’);
```

SEE THE ISSUE?

THE ISSUE

Where statement becomes

```
where owning_user = 'USRA'
```

A hardcoded literal that has many possible values

Massive parsing

Not good in a busy database

THE SOLUTION

Common wisdom - use bind variables…

WELL…

We cannot pass bind variables from a function

So… Lets mimic it with an application context

In the function, we replace this

```
return q'#owning_user = '#' || user || '';
```

with this

```
return q'#owning_user = sys_context('USERENV'
                                    ,'SESSION_USER')
```

Now the where clause becomes:

```
where owning_user = sys_context('USERENV'
                                ,'SESSION_USER')
```

That is then same no matter who is logged on!

SO MUCH MORE
SO LITTLE TIME

Lets finish up with trying to show this

+

How to use if in/with APEX