ORACLE

セキュリティ関連新機能

Oracle Database 23c新機能セミナー

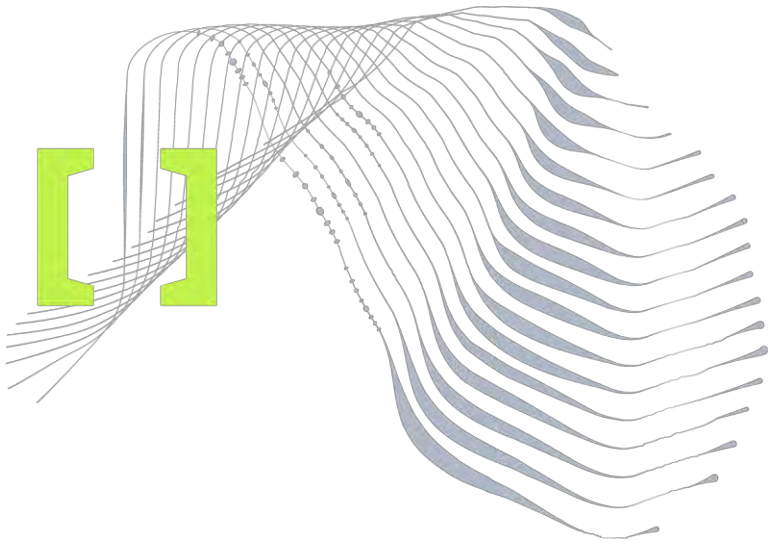
西村 克也, CISSP

日本オラクル株式会社

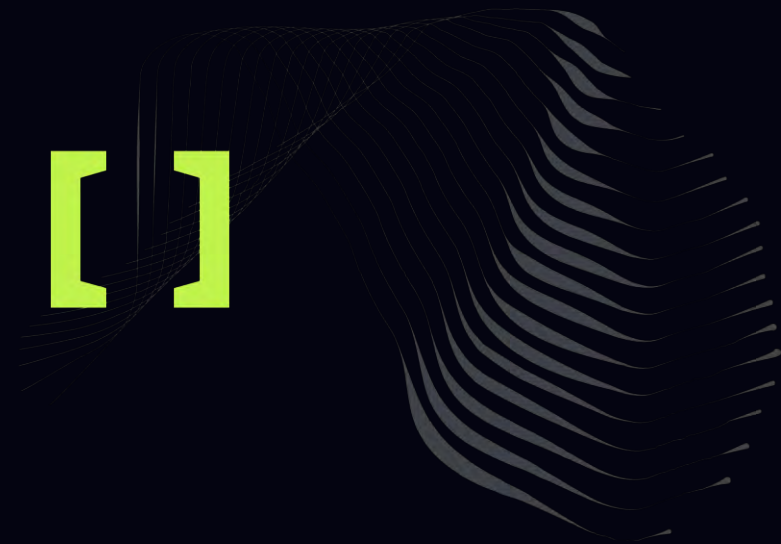
2023年10月19日

Agenda

1. SQL Firewall
2. Schema Privileges
3. Column Level Audit
4. Azure ADとのデータベース認証連携
5. その他新機能



SQL Firewall



SQL Firewall

SQLレベルで制御するデータベース・ネイティブのファイアーウォール

SQL単位でデータベースのアクセスを制御するファイアーウォール

データベース・ネイティブなので如何なる手段でもバイパスできない

アクセス許可されるSQLは、列名や条件を含めて完全一致しなければならず
SQLインジェクションや不必要なデータ参照を防止

実行されるSQLを収集しファイアーウォール・ポリシーを作成

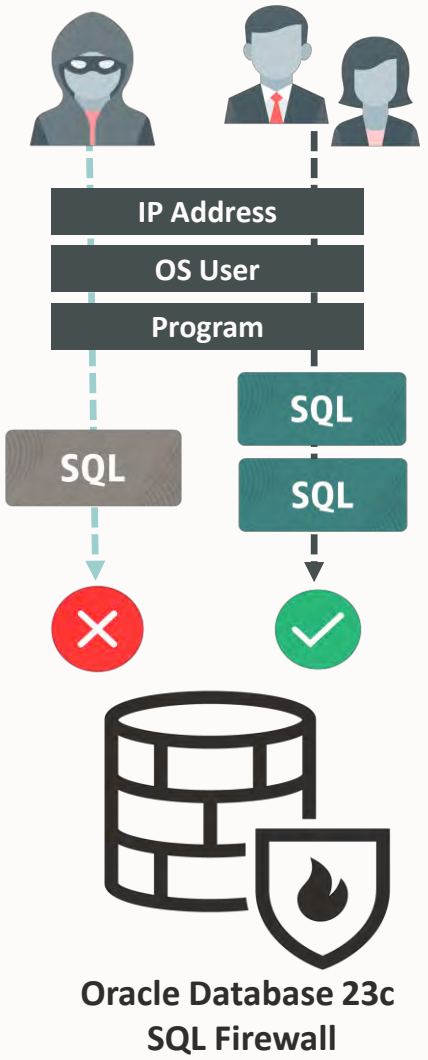
- SQL: DDL, DML
- セッション・コンテキスト: IPアドレス、OSユーザー名、プログラム名

非常に軽微なCPUオーバーヘッド

ポリシー違反のアクセスをブロックまたはログ記録だけの検知としても使用可能

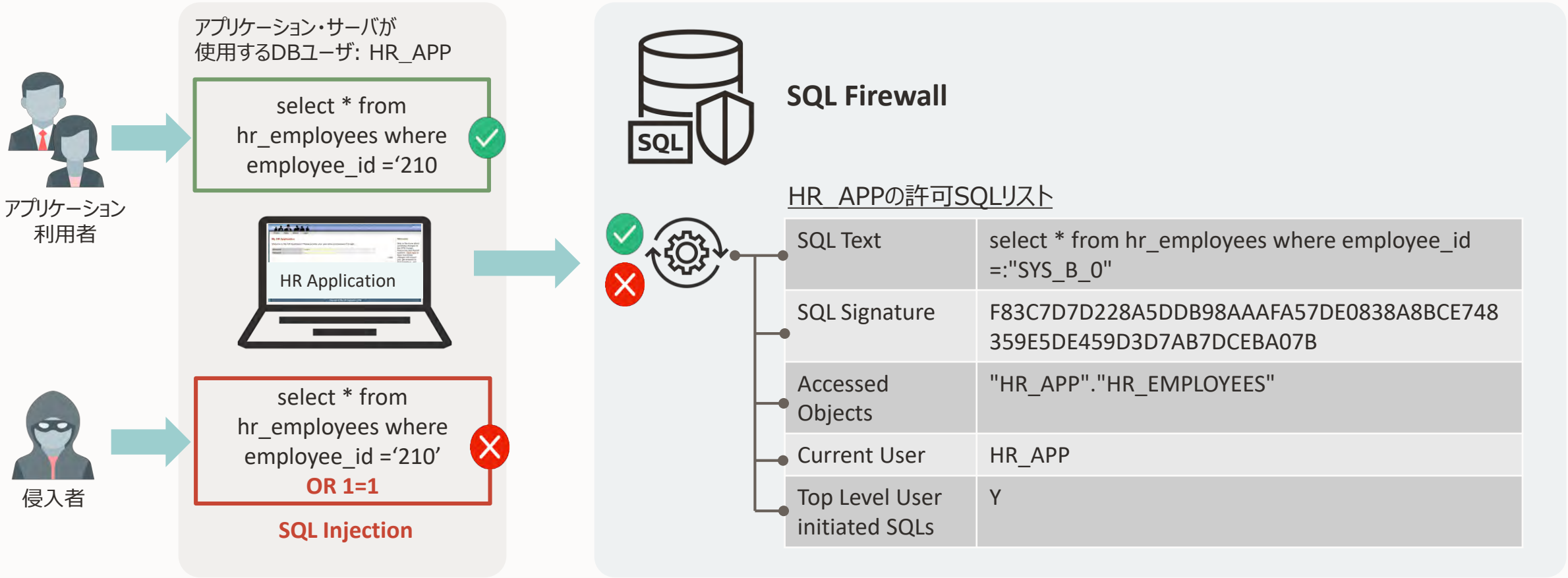
違反したログは、DBA_SQL_FIREWALL_VIOLATIONSビューで参照

AVDFまたはDatabase Vaultオプションで使用可
(※Cloudの場合、FreeもしくはBaseDB EE-HP以上)



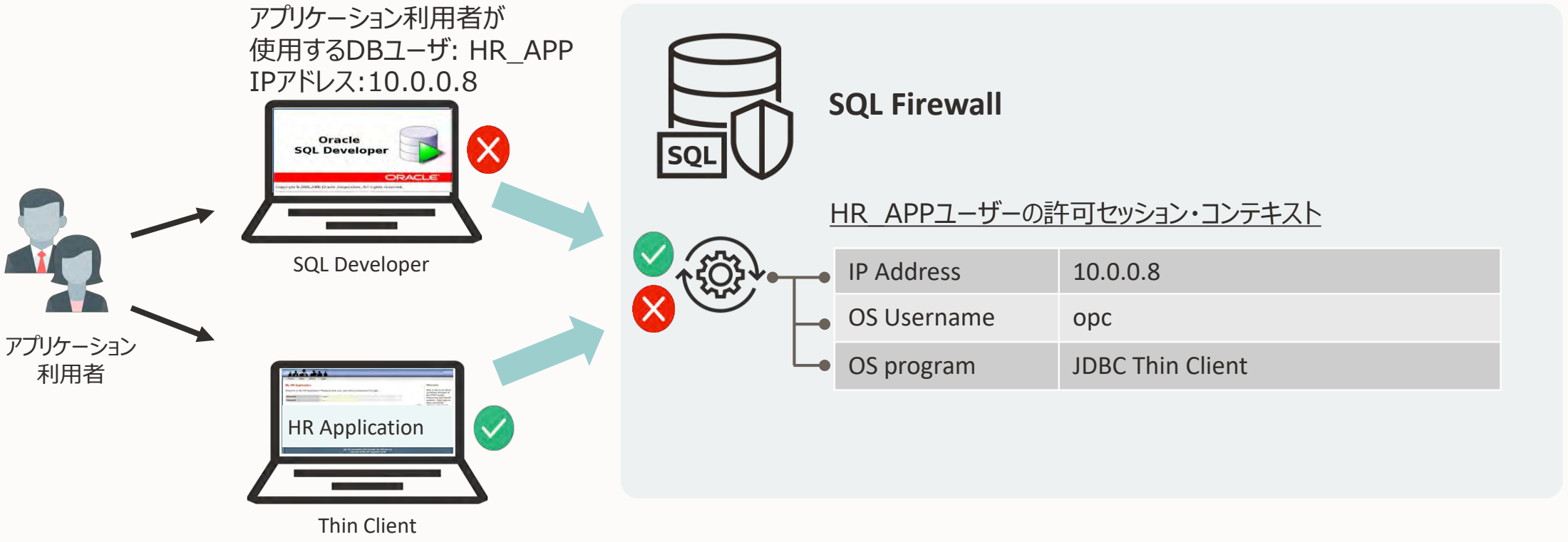
SQL Firewall

SQLリストに基づいてSQLクエリをブロックする



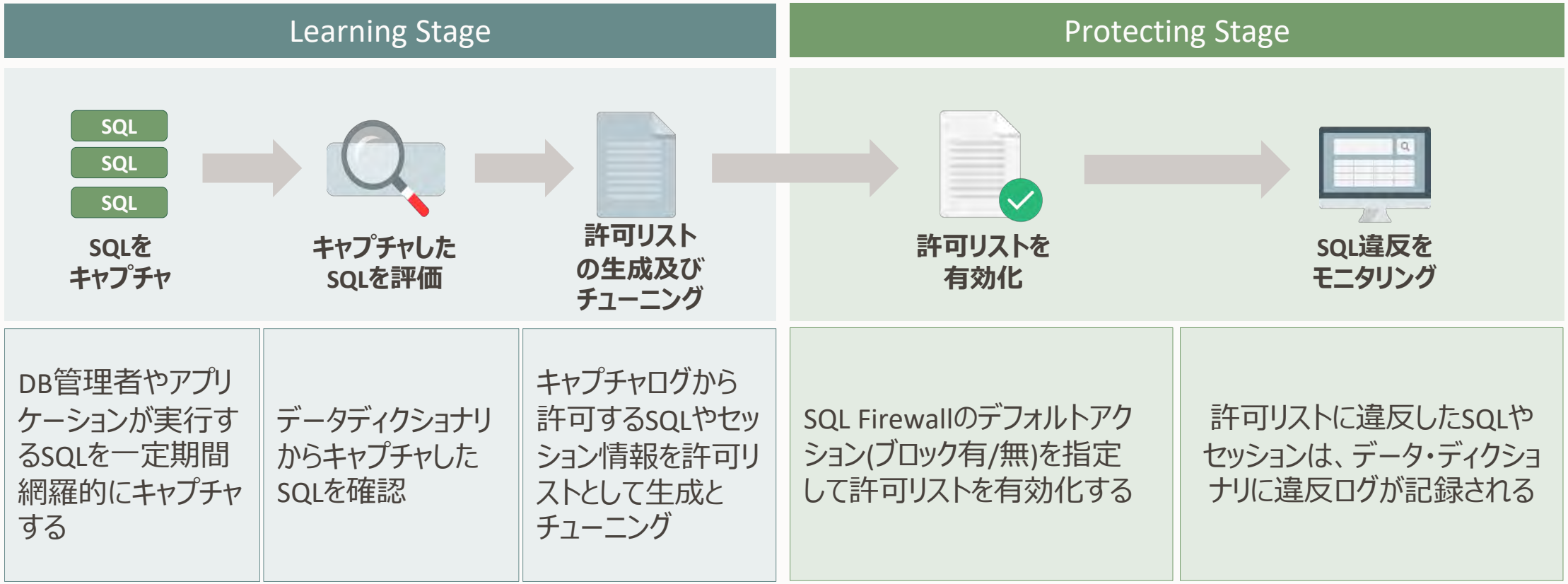
SQL Firewall

セッション・コンテキストに基づいてデータベース接続をブロックする



SQL Firewall

SQL Firewallを使用する基本的な流れ



SQL Firewall

Learning Stage ～ SQLをキャプチャ ～

SQL Firewallの有効化

```
EXEC DBMS_SQL_FIREWALL.ENABLE;
```

CDBまたはPDBごとに有効化
実行にはSQL_FIREWALL_ADMINロールが必要

SQLのキャプチャを作成

```
BEGIN
 DBMS_SQL_FIREWALL.CREATE_CAPTURE (
   username        => 'APP',
   top_level_only   => TRUE,
   start_capture    => TRUE
 );
END;
```

username
- モニター対象となるユーザーを指定
- SYS,SYSDG,SYSRAC, AUDSYS等は指定できない

top_level_only
- TRUE : ユーザーが直接実行したSQLが対象
- FALSE: PL/SQL内で実行されたSQLも含めて対象 (デフォルト)

start_capture
- TRUE: 作成と同時にキャプチャ開始
- FALSE: 任意のタイミングで開始

SQL Firewall

Learning Stage ～ SQLをキャプチャ - SQLの評価 ～

SQLのキャプチャを開始・停止

```
EXEC DBMS_SQL_FIREWALL.START_CAPTURE ('APP');
                --------  SQLを実行 --------
EXEC DBMS_SQL_FIREWALL.STOP_CAPTURE ('APP');
```

SQLをキャプチャするユーザーを指定する
キャプチャのパフォーマンスを最小化するためのパラメータ設定
- LARGE_POOL_SIZEに2GB以上を追加
- SGA_TARGETの設定値から8GB以上を追加

キャプチャ・ログは、DBA_SQL_FIREWALL_CAPTURE_LOGSから参照

```
SELECT USERNAME, SQL_SIGNATURE, SQL_TEXT, CLIENT_PROGRAM, IP_ADDRESS FROM DBA_SQL_FIREWALL_CAPTURE_LOGS;

USER     SQL_SIGNATURE     SQL_TEXT                                                              CLIENT_PROGRAM      IP_ADDRESS
-------- ------------------ ------------------------------------------------------------------- -------------------- ------------------
APP      26ABBD7CDxxxx     SELECT * FROM EMPLOYEES                                               sqlplus@db23c        127.0.0.1
APP      8B24E0F1D2xxxx     SELECT * FROM EMPLOYEES WHERE EMPLOYEE_ID=:"SYS_B_0“   JDBC                 10.0.0.58
APP      27BB61B8E0xxxx     SELECT EMPLOYEE_ID,FIRST_NAME,SALARY FROM EMPLOYEES    JDBC                 10.0.0.58
```

SQL Firewall
SQLシグネチャの生成アーキテクチャ

1. SQLからスペース、コメント、ヒントを取り除き標準化。リテラルは、バインド変数に置き換える

```
select -- comment (a.empno +1) “empNo” From /* comment */
  scoTT.Emp a  where empNo=1 and ename= ‘SCOTT’;
```

↓ 標準化

2. SQLクエリからアクセスするオブジェクト情報を取り出す

```
SELECT (A.EMPNO + “SYS_B_0”) “empNo” FROM SCOTT.EMP A
WHERE EMPNO=:”SYS_B_1” AND ENAME=:”SYS_B_3”
```

SCOTT.EMP

↓ ハッシュ(SHA-256)

3. 標準化したSQL+オブジェクト・アクセスリストをハッシュ化し、SQLシグネチャ生成

77D1C60B425F055F93F7014437EF68997
45E4C747C297D0AD799943707DD1189

SQLシグネチャはSQLを特定するためのユニークIDになる

SQL Firewall

トップ・レベルSQLキャプチャ

複数のSQLで構成されるプロシージャやファンクションなどの場合、トップレベルの実行SQLのみ、もしくは内部で実行されるSQLを含めてキャプチャするか選択可能

Exec EMPINFO(210)

```
PROCEDURE EMPINFO (id in number)
 Is
emp varchar2(2000);

Begin
..
Emp:='select ename from hr_employees
'||
'where employee_id = '''||id||'''';
..
End;
```

TOP_LEVEL=TRUE

SQL Text	EXECUTE BEGIN EMPINFO (?); END;

TOP_LEVEL=FALSE

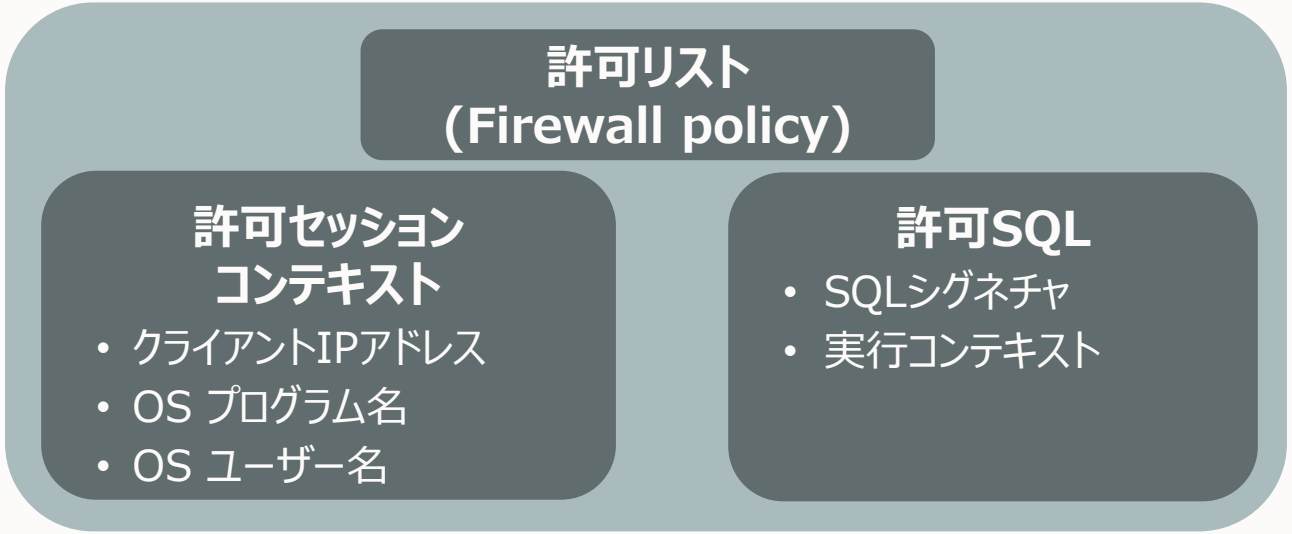
SQL Text	EXECUTE BEGIN EMPINFO (?); END;
SQL Text	SELECT ENAME FROM HR_EMPLOYEES WHERE EMPLOYEE_ID =:"SYS_B_0"

SQL Firewall

キャプチャ・ログから生成する許可リスト

キャプチャ・ログからSQL Firewallのファイアーウォール・ポリシーとなる許可リストを生成する
許可リストは、下記2種類のタイプから構成される
- 許可セッション・コンテキスト: クライアントIPアドレス、OSプログラム名、OSユーザー名
- 許可SQL: SQLシグネチャ、実行コンテキスト(表やビューなどのオブジェクト情報)

接続するDBセッションやSQLが許可リストにマッチングしない場合、SQL Firewallの違反が発生する



SQL Firewall

Learning Stage ～ 許可リストの生成 ～

許可リストを生成

```
EXEC DBMS_SQL_FIREWALL.GENERATE_ALLOW_LIST ('APP');
```

SQLキャプチャを停止後、許可リストをユーザー指定して生成
許可リストは、キャプチャ・ログのSQLとセッション情報を元に作成される

許可リストのSQLは、DBA_SQL_FIREWALL_ALLOWED_SQLビューから参照

```
SELECT ALLOWED_SQL_ID, USERNAME, SQL_SIGNATURE, SQL_TEXT FROM DBA_SQL_FIREWALL_ALLOWED_SQL;

SQL_ID   USERNAME    SQL_SIGNATURE    SQL_TEXT
--------  -------------  -------------------  ------------------------------------------------------
1            APP             26ABBD7CDxxxx    SELECT * FROM EMPLOYEES
2            APP             8B24E0F1D2xxxx    SELECT * FROM EMPLOYEES WHERE EMPLOYEE_ID=:"SYS_B_0“
3            APP             27BB61B8E0xxxx    SELECT SALARY,FIRST_NAME,EMPLOYEE_ID FROM EMPLOYEES
4            APP             982161AC03xxxx     SELECT FIRST_NAME,SALARY,EMPLOYEE_ID FROM EMPLOYEES
```

SQL Firewall

Learning Stage ～ 許可リストの生成 ～

許可リストのセッション・コンテキストは、以下それぞれのビューから参照可

- IPアドレス: DBA_SQL_FIREWALL_ALLOWED_IP_ADDR
- プログラム名: DBA_SQL_FIREWALL_ALLOWED_OS_PROG
- OSユーザ名: DBA_SQL_FIREWALL_ALLOWED_OS_USER

IPアドレスの許可リスト

```
SELECT * FROM DBA_SQL_FIREWALL_ALLOWED_IP_ADDR;
USERNAME    IP_ADDRESS
-------------   ----------------------
APP             127.0.0.1
APP             10.0.0.58
```

プログラム名の許可リスト

```
SELECT * FROM DBA_SQL_FIREWALL_ALLOWED_OS_PROG;
USERNAME    OS_PROGRAM
-------------   ----------------------
APP             sqlplus@db23c (TNS V1-V3)
APP             JDBC
```

SQL Firewall

Learning Stage ～ 許可リストのチューニング ～

生成した許可リストから必要なものだけを追加・削除するチューニングをすることが可能
ただし、下記の制限があるで注意

- SQLの場合、任意のSQLの個別追加はできない。追加の場合は、違反ログ、または、キャプチャのプロセスを再度実行した
  キャプチャ・ログから一括ですべてを追加しなければならない
- DBMS_SQL_FIREWALL.APPEND_ALLOW_LISTは、許可リストが有効時でも追加でき、即時反映される
- セッション情報(IPアドレス、プログラム名、OS名)の場合は、任意の値の追加、削除が可能

SQLをログから追加

```
BEGIN
 DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST (
   username  =>'HR',
   source       => DBMS_SQL_FIREWALL.VIOLATION_LOG
  );
END;
/
```

username
- 対象となるユーザーを指定

source
- DBMS_SQL_FIREWALL.CAPTURE_LOG : キャプチャログから
- DBMS_SQL_FIREWALL.VIOLATION_LOG: 違反ログから
- DBMS_SQL_FIREWALL.ALL_LOG: 上記合わせて

SQL Firewall

Learning Stage ～ 許可リストのチューニング ～

許可リストからSQLを削除するには、DBMS_SQL_FIREWALL.DELETE_ALLOWED_SQLを使用する
DBA_SQL_FIREWALL_ALLOWED_SQLビューから、削除したSQLのALLOWED_SQL_IDを参照する
削除したSQL_IDは再利用されない。削除したSQLを再び追加した場合、採番されているSQL IDの順番に従う

許可リストからSQL IDを指定して削除

```
BEGIN
 DBMS_SQL_FIREWALL.DELETE_ALLOWED_SQL (
   username          => 'APP',
   allowed_sql_id   => 1
  );
END;
/
```

username
- 対象となるユーザーを指定

allowed_sql_id
- DBA_SQL_FIREWALL_ALLOWED_SQLの ALLOWED_SQL_ID列を参照

SQL Firewall

Learning Stage ～ 許可リストのチューニング ～

セッション・コンテキストの追加は、DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT
削除は、DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXTを使用する

コンテキストタイプは、以下を指定。ワイルドカードの使用可

- IPアドレス: DBMS_SQL_FIREWALL.IP_ADDRESS
- OSプログラム名: DBMS_SQL_FIREWALL.OS_PROGRAM
- OSユーザ名: DBMS_SQL_FIREWALL.OS_USERNAME

指定したIPアドレスを許可リストに追加

```
BEGIN
 DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT(
   username     => 'APP',
   context_type => DBMS_SQL_FIREWALL.IP_ADDRESS,
   value        => '10.0.0.100');
END;
/
```

指定したIPアドレスを許可リストから削除

```
BEGIN
  DBMS_SQL_FIREWALL.DELETE_ALLOWED_CONTEXT (
    username     => 'APP',
    context_type => DBMS_SQL_FIREWALL.IP_ADDRESS,
    value        => '10.0.0.58');
END;
/
```

SQL Firewall

Protecting Stage ～ 許可リストの有効化 ～

許可リストを有効化するには、DBMS_SQL_FIREWALL.ENABLE_ALLOW_LISTを実行する
違反するSQLやセッションを検出した場合のアクションをブロック、または検出のみのいずれかで選択

許可リストを有効化する

```
BEGIN
DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST(
username=>'APP',
enforce=>DBMS_SQL_FIREWALL.ENFORCE_ALL,
block=>TRUE
);
END;
/
```

許可リストの停止・削除

```
exec DBMS_SQL_FIREWALL.DISABLE_ALLOW_LIST('APP');
exec DBMS_SQL_FIREWALL.DROP_ALLOW_LIST('HR');
```

username
- モニター対象となるユーザーを指定

enforce
- DBMS_SQL_FIREWALL.ENFORCE_ALL : すべて
- DBMS_SQL_FIREWALL.ENFORCE_CONTEXT : セッションのみ
- DBMS_SQL_FIREWALL.ENFORCE_SQL : SQLのみ

block
- TRUE: 許可リストに該当しない場合はSQLをブロックする
- FALSE: ブロックせずはしないが違反ログには記録される

SQL Firewall

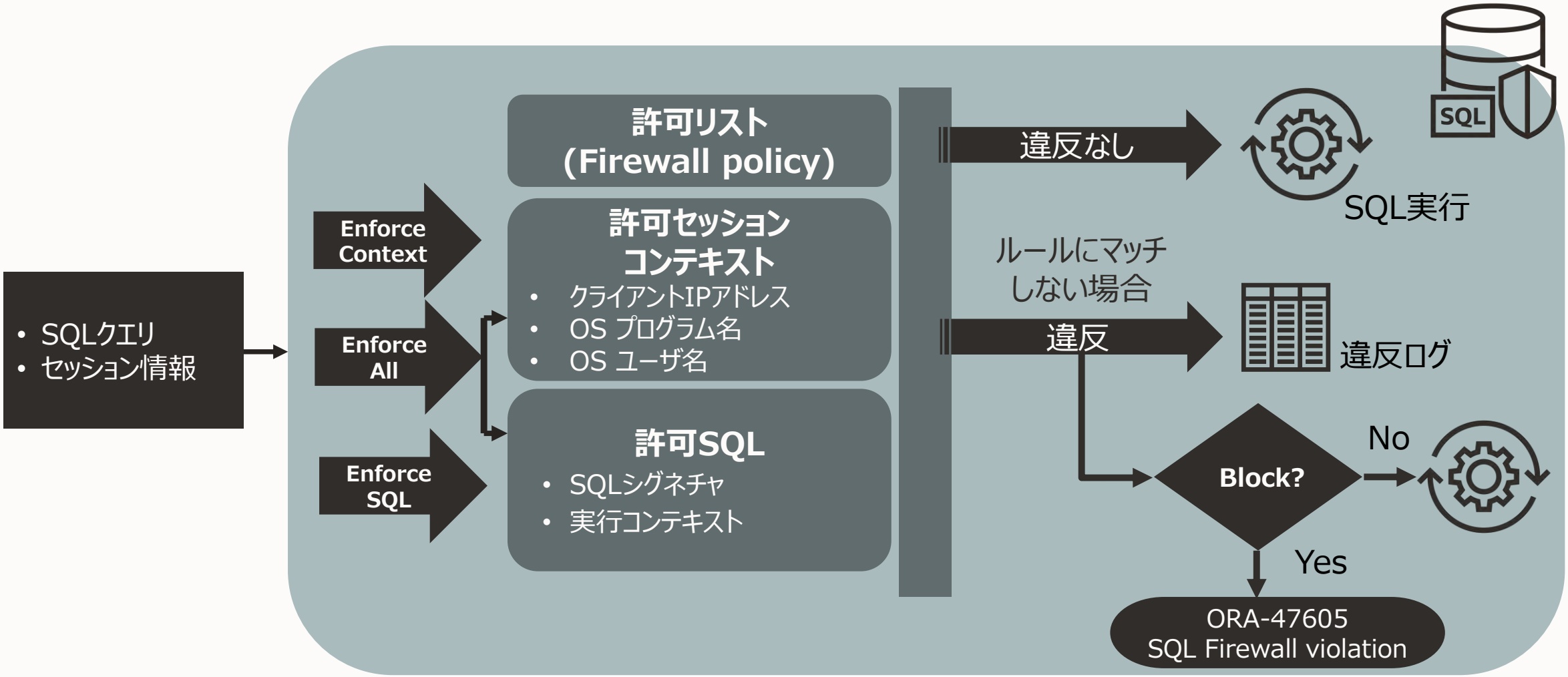
Protecting Stage ～ 違反ログのモニタリング ～

許可リストに違反するSQLやセッションは、違反ログとしてDBA_SQL_FIREWALL_VIOLATIONSに記録される
OCCURRED_AT列 (TIMESTAMP(6) WITH TIME ZONE)は、違反した時間を表す
許可リストの有効時にBlockを指定しない場合は、FIREWALL_ACTION列は、Allowedの表記になる
監査ログにも同様に記録される (audit_type='SQL Firewall' , action_name='SQL VIOLATION')

```
SELECT TO_CHAR(to_timestamp_tz(OCCURRED_AT) at time zone 'Asia/Tokyo', 'yyyy/mm/dd hh24:mi:ss tzd')EVENTTIME, USERNAME,
SQL_TEXT, FIREWALL_ACTION, CAUSE FROM DBA_SQL_FIREWALL_VIOLATIONS;

EVENTTIME                    USERNAME    SQL_TEXT                                    FIREWALL ACTION     CAUSE
--------------------------   ----------  ------------------------------------------  ------------------  -----------------
2023/09/19 15:57:32 JST      APP         SELECT *  FROM EMPLOYEES                    Blocked             SQL violation
2023/09/19 15:57:55 JST      APP         SELECT SALARY,EMAIL EMPLOYEES               Blocked             SQL violation
2023/09/19 16:01:40 JST      APP         null                                        Blocked             Context violation
```

SQL Firewall
許可リストのフローチャート



SQL Firewall
ログのパージ

ディスク・スペースの節約のために、定期的に不必要なログをパージすることが推奨される
SQL Firewallのログのパージには、DBMS_SQL_FIREWALL.PURGE_LOGを使用する

SQL Firewallのログのパージ

```
BEGIN
 DBMS_SQL_FIREWALL.PURGE_LOG (
   username    => 'APP ',
   purge_time  => '2023-09-19 00:00:00.00 -08:00',
   log_type    => DBMS_SQL_FIREWALL.ALL_LOGS
  );
 END;
/
```

username
- 対象となるユーザーを指定

purge_time
- 指定した時刻より前のログ・レコードをパージする
- このパラメータを省いた場合は、すべてのレコードが対象となる

log_type
- DBMS_SQL_FIREWALL.CAPTURE_LOG : キャプチャ・ログのみ
- DBMS_SQL_FIREWALL.VIOLATION_LOG : 違反ログのみ
- DBMS_SQL_FIREWALL.ALL_LOGS: 上記二つ両方

SQL Firewall
許可リストのエクスポート・インポート

生成した許可リストは、DBMS_SQL_FIREWALL.EXPORT_ALLOW_LISTによってCLOBにエクスポート可能
空のCLOBを指定し、許可リストはJSONフォーマットとして格納される
DBMS_SQL_FIREWALL.IMPORT_ALLOW_LISTで指定した許可リストをインポート
インポートは、既存の許可リストに追加され、重複するSQLやセッションの情報は追加しない

許可リストのエクスポート

```
create table allow_list_json(a number, b clob);
insert into allow_list_json values(1,empty_clob());
Declare
 json clob;
Begin
 select b into json from allow_list_json where a=1 for update;
 DBMS_SQL_FIREWALL.EXPORT_ALLOW_LIST (
  username  => 'HR' ,
  allow_list  => json);
  update allow_list_json set b=json where a=1;
End;
/
```

許可リストのインポート

```
Declare
 json clob;
Begin
 select b into json from allow_list_json where a=1;
 DBMS_SQL_FIREWALL.IMPORT_ALLOW_LIST (
  username  => 'HR' ,
  allow_list  => json);
End;
/
```

SQL Firewall

メタデータのエクスポート・インポート

SQL Firewallのキャプチャログや許可リスト等のメタデータは、Data Pumpによるエクスポート/インポートが可能
テストDBでキャプチャしたログやチューニングした許可リストを本番DBに移行するための利用を想定
インポートには、ADMINISTER SQL FIREWALLの権限が必要
インポートは、既存のメタデータに対して追記される

エクスポート

```
SQL> CREATE DIRECTORY sqlfw_dir AS '/home/oracle';
$ expdp username/password@service_name  FULL=Y DIRECTORY=sqlfw_dir  INCLUDE=SQL_FIREWALL DUMPFILE=sql_fw.dmp
```

インポート

```
$ impdp  username/password@service_name FULL=Y DIRECTORY=sqlfw_dir  INCLUDE=SQL_FIREWALL dumpfile=sql_fw.dmp
```

SQL Firewall
管理インターフェース

2つの管理オプション

DBMS_SQL_FIREWALL パッケージ

- PL/SQLによる基本のCLI管理

Oracle Data SafeのSQL Firewall管理機能

- Data Safeのインタフェースから GUI管理
- SQL Firewallの設定や違反ログなどビジュアライズされたGUIによる直感的な操作
- EventsとNotificationsと連携することで、許可リストの違反が発生した場合のアラート通知が可能

OCIにある3種類のFirewallの違い

	 Web Application Firewall	 Network Firewall	 SQL Firewall
目的	Webアプリケーションの脆弱性を悪用した攻撃からWebサイトを保護する	インターネットやVCN内のネットワーク・トラフィックを監視し不正アクセスからネットワークを保護する	SQLレベルでデータベースの不正アクセスを保護する
保護対象	WEBアプリケーション	ネットワーク	データベース
対応する主な攻撃	XSS(クロスサイトスクリプティング)、CSRF、DDoS、OS・SQLインジェクション等	DDoSやネットワークスキャン、マルウェア、C&Cなどの不正アクセス全般	SQLインジェクションや不正なSQLクエリ
対象プロトコル	HTTPS、HTTP	TCP、UDP、ICMP、その他	SQL、JDBC/ODBC等
保護ポリシー	600以上の保護ルール(Oracle管理)	Pala Alto独自の脅威検出エンジン	ユーザー自身でSQLポリシーを作成
特徴	Webアプリケーションのコード修正や設定変更なく、迅速な導入が可能	保護対象の広さと汎用性が高く、侵入から攻撃までの様々な不正アクセスを検出	DBのネイティブ機能なので、バイパスできない最高レベルのSQLアクセス制御
考慮事項	Webアプリケーションに応じた適切な保護ルールの選択、運用ルールの検討が必要	配置場所に応じてVCNのネットワークの再設計が必要	厳格なSQL単位でのポリシーのため稼働後の運用ルールの検討が必要

Schema Privileges

Schema Privileges

アクセス範囲をスキーマ・レベルに限定

従来のスキーマへのアクセス付与にはそれぞれの表ごとの指定が必要
また、新規の表が作成された場合には、改めて付与しなければならない

- GRANT SELECT ON HR.EMPLOYEES to <ユーザー名>

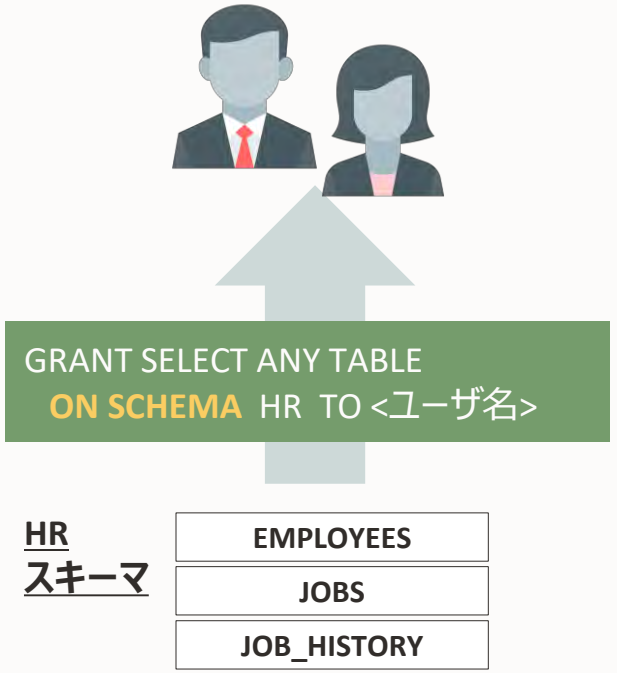
従来のSELECT ANY TABLE権限は、対象がDB内のすべての表になってしまう
強力なシステム権限

Schema Privilegesは、指定したスキーマのすべて表とおよび新規作成表に対して
包括的に権限を付与することが可能

GRANT **SELECT ANY TABLE ON SCHEMA <スキーマ名>** to <ユーザー名>

アップグレードやパッチ適用間に作成される表やDWHのデータ集計時の表など
一時的に作成される表へのアクセス権の利便性なども向上

同様にINSERT/UPDATE/DELETE ANY TABLEで使用できる



Schema Privileges
使用例

SYSユーザで操作

#BaseDB 23cにあるHRスキーマを使用する

#TESTユーザを作成し、HRのEMPLOYEES表のアクセス権を付与

```
CREATE USER TEST IDENTIFIED “password”;
GRANT CREATE SESSION TO TEST;
GRANT SELECT ON HR.EMPLOYEES TO TEST;
```

#HRスキーマ単位でSELECTを付与

```
GRANT SELECT ANY TABLE ON SCHEMA HR to TEST;
```

#新規表を作成

```
CREATE TABLE HR.NEW_TABLE(COL1 NUMBER);
```

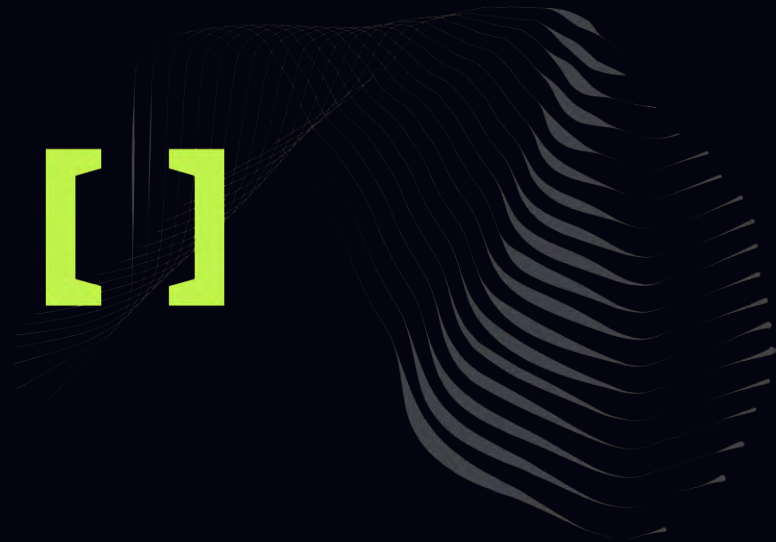
TESTユーザで操作

```
SELECT * FROM HR.EMPLOYEES;   -->   EMP表には当然アクセス可能
SELECT * FROM HR.JOBS;   --> EMP表以外は権限がないのでアクセス不可
ERROR at line 1:
ORA-00942: table or view does not exist
```

```
SELECT * FROM HR.JOBS;   --> HRのすべての表にSELECTが可能
```

```
SELECT * FROM HR.NEW_TABLE;   --> 追加された表も同様にアクセス可能
```

Column Level Audit



Column Level Audit
表の列をトリガーにした監査アクション

監査アクションの対象を従来の表単位から列単位での特定が可能

CREATE AUDIT POLICY ポリシー名 **ACTIONS** DML(カラム名) **ON** テーブル名

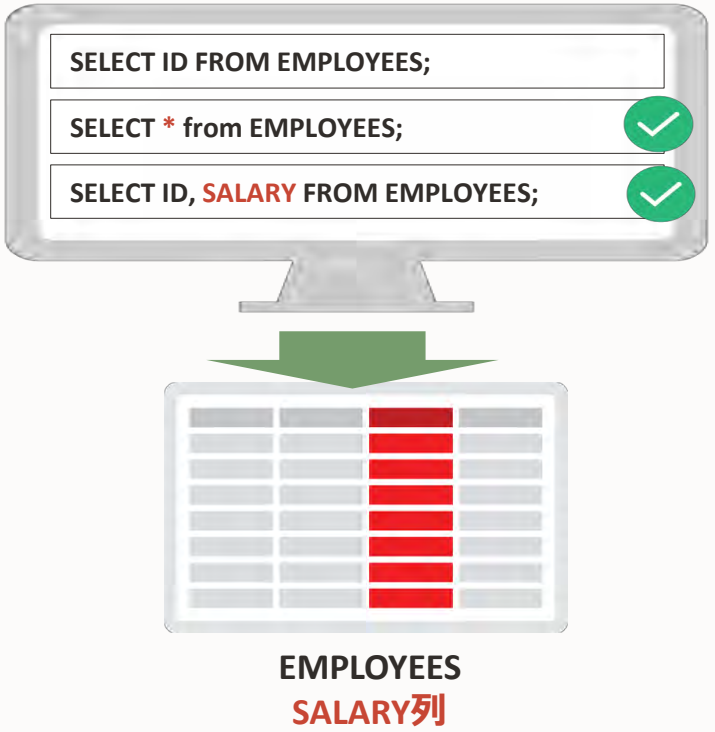
例) EMPLOYEES表のSALARY列のSELECT, UPDATEを対象

```
CREATE AUDIT POLICY policy1 ACTIONS  SELECT(SALARY), UPDATE(SALARY)
 ON HR.EMPLOYEES;

AUDIT POLICY policy1;
```

ログは、UNIFIED_AUDIT_TRAILビューをから参照

監査対象を絞り込むことは、監査ログのストレージ領域の肥大を抑え
DMLのパフォーマンスへのオーバーヘッドも最小限にすることができる



Unified Audit

Oracle Databaseの監査機能

データベース・オブジェクト、権限、ユーザー・アクション等、監査対象をグループ化したポリシーベースの監査
事前定義済みの監査ポリシーで必要とされる最小限の監査項目をカバー
ユーザー・セッション情報(IPアドレス、ユーザー名、プログラム名等)を監査条件にログ出力の絞り込み
監査ログはデータベース内の内部表として格納され、UNIFIED_AUDIT_TRAILビューで参照可
SYSユーザー監査、Recovery Manager、Data Pump、SQL*Loader等のログも統合
推奨される監査のベストプラクティス

- すべてのアクションを監査対象ではなく、コンプライアンス要件やセキュリティの懸念のあるアクションにフォーカス
- DBMS_AUDIT_MGMTパッケージで定期的にログをパージ。別サービスと連携したログの長期保管も検討

例)表に対するDML監査ポリシー

```
CREATE AUDIT POLICY Policy1  ACTIONS UPDATE ON HR.EMP, DELETE ON HR.EMP_EXD;
AUDIT POLICY Policy1;
```

Unified Audit

Unified Auditの定義済みポリシー

ポリシー名	ポリシーの内容	デフォルト
ORA_LOGON_FAILURES	ログイン失敗のみ	Yes (DBCAでDB作成時)
ORA_SECURECONFIG	セキュリティ監査の必須要件として求められる基本的なデータベースの構成管理に関連した操作	Yes (DBCAでDB作成時)
ORA_DATABASE_PARAMETER	データベースのパラメータ変更に関連した操作	No
ORA_ACCOUNT_MGMT	ユーザー・アカウントの変更や権限に関連した操作	No
ORA_CIS_RECOMMENDATIONS	CISベンチマークで求められる監査要件に関連した操作	No
ORA_RAS_POLICY_MGMT, ORA_RAS_SESSION_MGMT	Real Application Securityに関連した操作	No
ORA_DV_AUDPOL	Oracle Database Vaultの DVSYS, LBACSYSスキーマのオブジェクトに関連した操作	No
ORA_DV_AUDPOL2	Database Vaultのレルムやコマンドルールに関連した操作	No

Unified Audit

ORA_SECURECONFIGポリシー

主にデータベース管理に関連する操作に対して監査ポリシー
表などのオブジェクトに対する監査設定はされていないので、ユーザー用途に応じたDML監査を追加する

ALTER ANY TABLE CREATE ANY TABLE DROP ANY TABLE	ALTER ANY PROCEDURE CREATE ANY PROCEDURE DROP ANY PROCEDURE	GRANT ANY PRIVILEGE GRANT ANY OBJECT PRIVILEGE GRANT ANY ROLE
CREATE USER DROP USER ALTER USER	ALTER DATABASE ALTER SYSTEM AUDIT SYSTEM	ALTER ANY SQL TRANSLATION PROFILE CREATE ANY SQL TRANSLATION PROFILE DROP ANY SQL TRANSLATION PROFILE
CREATE ANY LIBRARY	CREATE ANY JOB CREATE EXTERNAL JOB	CREATE SQL TRANSLATION PROFILE
CREATE PUBLIC SYNONYM DROP PUBLIC SYNONYM	EXEMPT ACCESS POLICY EXEMPT REDACTION POLICY	TRANSLATE ANY SQL
PURGE DBA_RECYCLEBIN	LOGMINING	ADMINISTER KEY MANAGEMENT
BECOME USER CREATE PROFILE ALTER PROFILE DROP PROFILE	CREATE ROLE ALTER ROLE DROP ROLE SET ROLE	CREATE DATABASE LINK ALTER DATABASE LINK DROP DATABASE LINK
CREATE DIRECTORY DROP DIRECTORY	EXECUTE ON DBMS_RLS ALTER DATABASE DICTIONARY	CREATE PLUGGABLE DATABASE DROP PLUGGABLE DATABASE ALTER PLUGGABLE DATABASE

Unified Audit
条件による監査対象の絞り込み

特定のユーザーのUPDATE,DELETE文のDMLを監査

```
CREATE AUDIT POLICY Policy1 ACTIONS UPDATE ON EMPLOYEES, DELETE ON EMP_EXTENDED;
AUDIT POLICY Policy1 BY UserX, UserY;
```

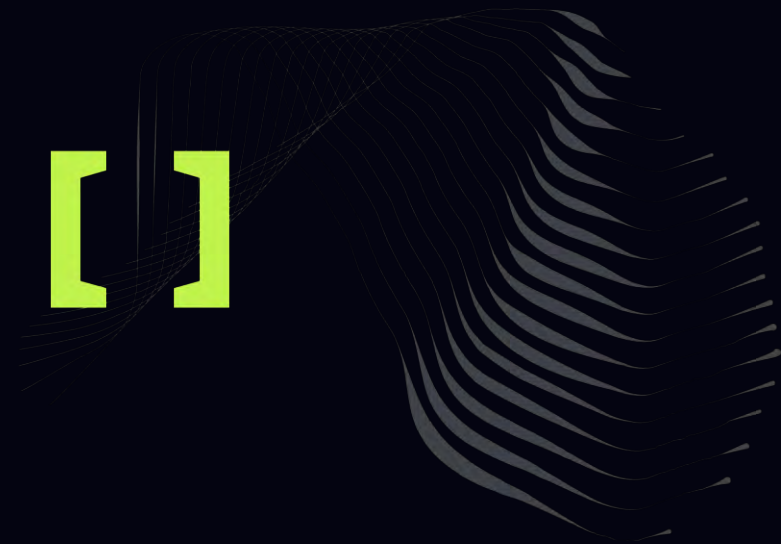
WHEN句で、IPアドレスがNULL(ローカル接続)の場合のみの条件を追

```
CREATE AUDIT POLICY Policy2 ACTIONS UPDATE ON EMPLOYEES, DELETE ON EMP_EXTENDED
WHEN 'SYS_CONTEXT(''USERENV'',''IP_ADDRESS'') IS NULL’
AUDIT POLICY Policy2;
```

Actions ALLだけれども、JDBC接続の特定のAPサーバのアクセスは除く条件を指定

```
CREATE AUDIT POLICY Policy3 ACTIONS ALL ON EMPLOYEES
WHEN 'SYS_CONTEXT(''USERENV'',''HOST'') IN ‘’xxxxx.jp.oracle.com,xxxxx.jp.oracle.com‘’ AND
SYS_CONTEXT(''USERENV'',''MODULE'') NOT IN ''JDBC Thin Client'''
AUDIT POLICY Policy3;
```

Azure ADとのデータベース認証連携



Azure ADとのデータベース認証連携

トークン・ベースの外部ユーザー認証

Azure ADとOAuth2トークンによるOracle Databaseの認証と認可

Oracle DatabaseとAzure AD間の通信はTLSにより暗号化され
セキュアにトークンの受け渡しが可能

Azure ADのユーザー・ロールとOracle Databaseのスキーマを
マッピングすることで、Azure ADによる統合的なユーザー管理が可能

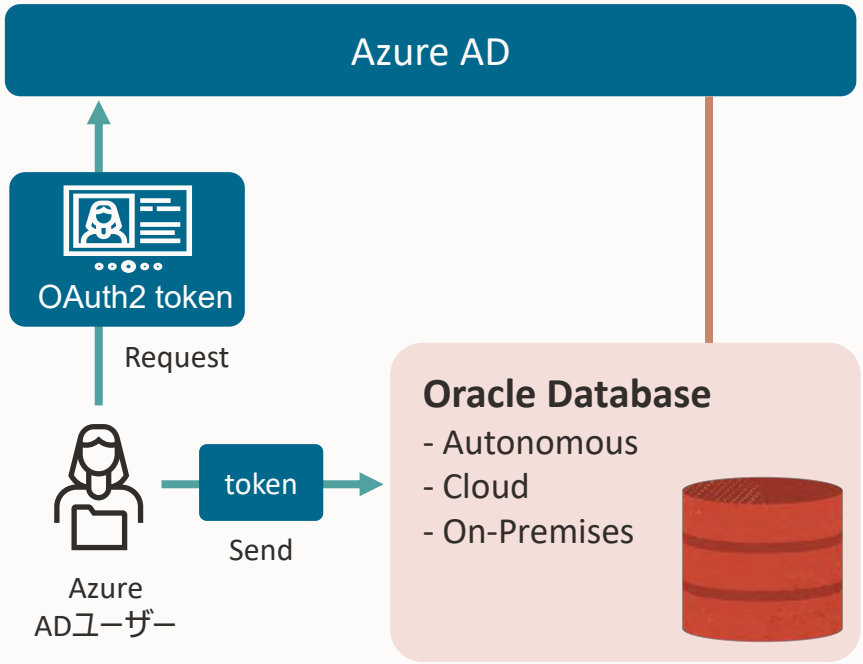
JDBC-thin、ODP.NETクライアントはネイティブクラウド認証に対応

23cの新機能として開発され、先行してAutonomous Database及び
19cにバックポート済み

対象データベース：

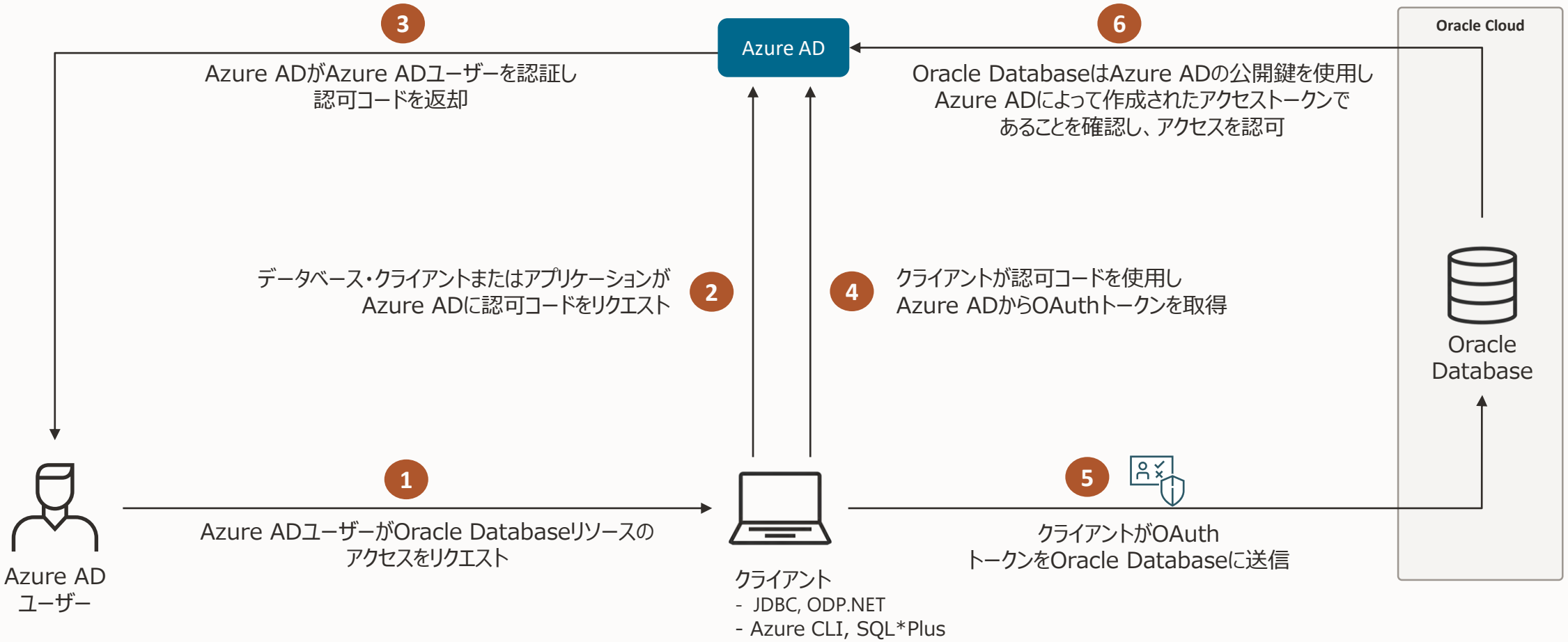
- Oracle Database 19c(19.18～)、23c (※21cは対象外、OSはLinuxのみ)
- Autonomous Database on Shared/Dedicated Exadata Infrastructure
- Oracle Base Database Service, Oracle Exadata Cloud Service

同様にIdentity Domainsとの認証連携が可能 (※クラウドのみ)



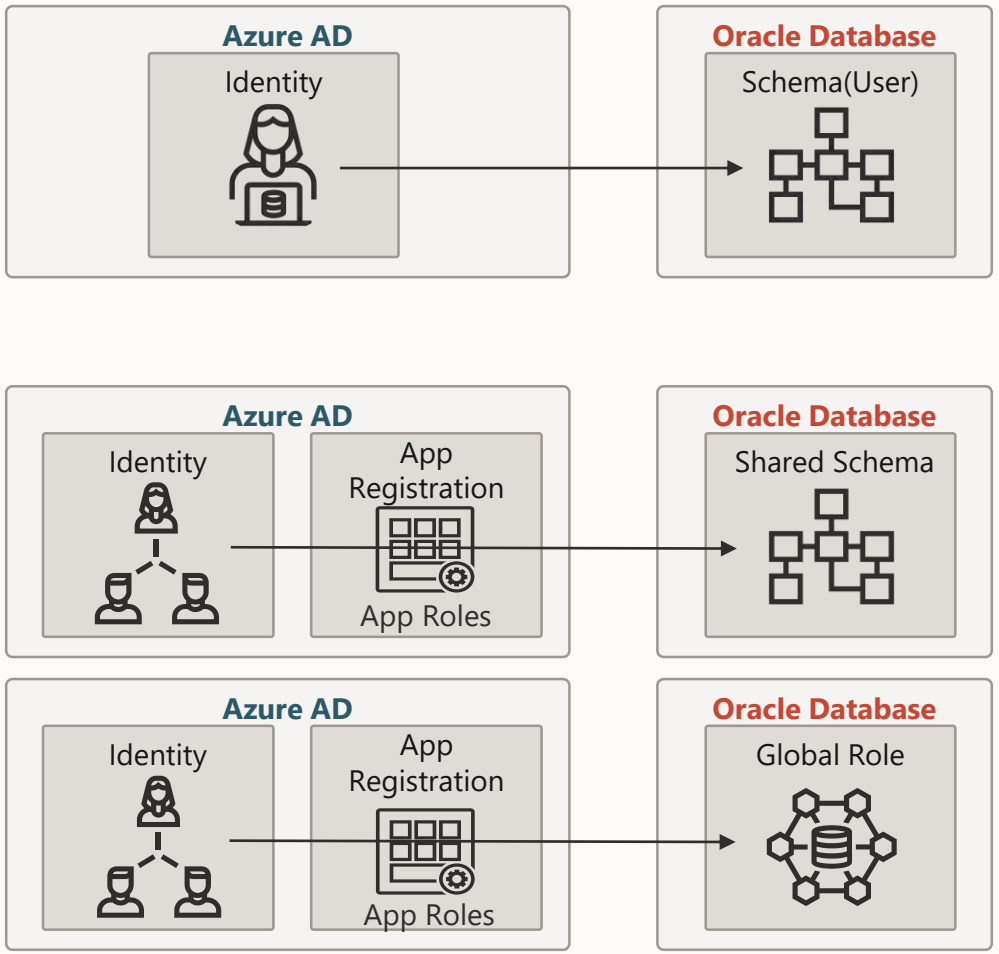
Azure ADとのデータベース認証連携

認証・認可フロー



Azure ADとのデータベース認証連携

Azure ADユーザーとDBスキーマのマッピング方式



排他的マッピング

- Azure ADユーザーとOracle Databaseスキーマを直接マッピング
- ADユーザの追加・削除時には、同様にDB側のスキーマの修正に必要
- ADユーザーと同数のDBスキーマが必要

```
CREATE USER peter IDENTIFIED GLOBALLY AS
'AZURE_USER=peter@example.com'
```

グローバル・マッピング

- Azure ADアプリロールとOracle Database Sharedスキーマもしくは Oracle Database Globalロールをマッピング
- ADユーザの追加・削除時には、DB側の変更なくAD側のみで完結
- ADアプリロールと同数のDBスキーマ(ユーザ)・ロールを用意

```
CREATE USER dba_azure IDENTIFIED GLOBALLY AS
'AZURE_ROLE=AZURE_DBA'
```

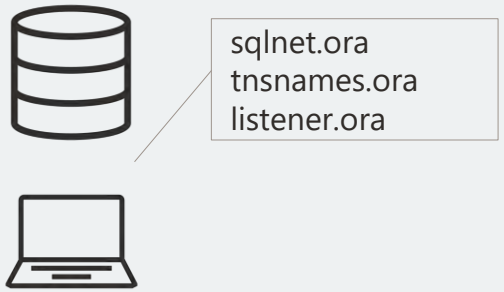
```
CREATE ROLE sales_role IDENTIFIED GLOBALLY AS
'AZURE_ROLE=SalesGroup'
```

Azure ADとのデータベース認証連携

必要な設定手順の流れ

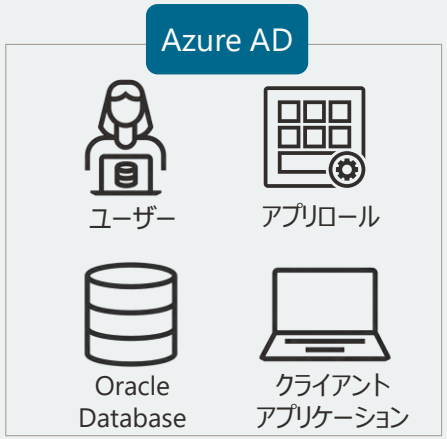
Oracle DatabaseでTLS通信を有効化

- ✓ Walletの作成
- ✓ 構成ファイルの修正
- ✓ システム再起動
- ✓ クライアントサーバーの構成



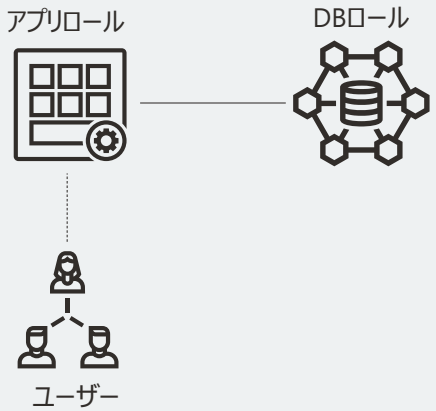
Azure ADの構成

- ✓ Oracle Databaseの登録
- ✓ クライアントアプリケーションの登録
- ✓ アプリロールの作成
- ✓ ユーザー作成



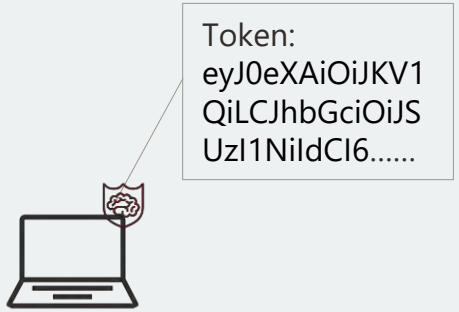
Oracle Databaseの構成

- ✓ Azure ADによる外部認証の有効化
- ✓ Azure ADの情報登録
- ✓ アプリロールとOracle Databaseグローバル・ロールのマッピング

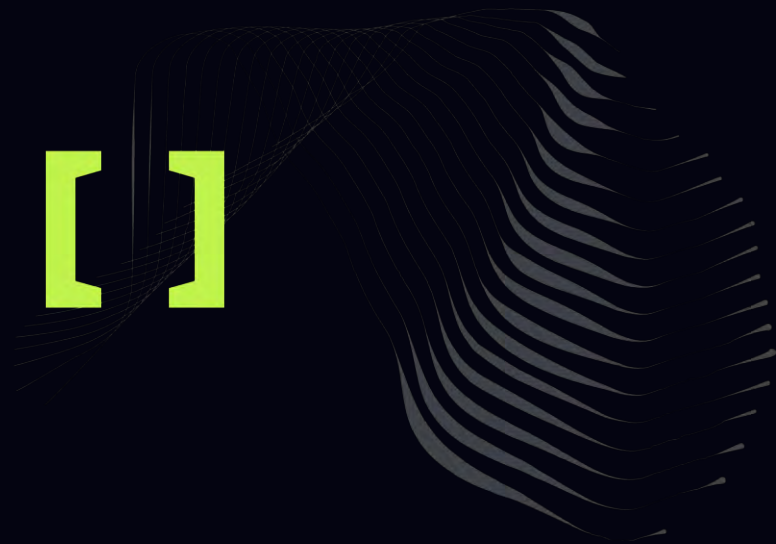


クライアントアプリケーションの構成

- ✓ トークンの取得
- ✓ 構成ファイルの修正



その他の新機能



その他の新機能

- パスワードの長さを1024バイトまでサポート
  - 従来は30バイトまで
  - IDCSやIdentity Domainsなどのクラウドと同等のパスワード・ポリシーが適用可能

- Read Onlyユーザー・セッション
  - ユーザーを参照のみに変更し、INSERTやDELETE,CREATEなどの作成や更新は一切できない
  - 既存の権限を上書きするので、仮にDBA権限を付与されていても強制的に参照のみになる

```
#Read Onlyユーザーの作成・変更
CREATE USER <ユーザー名> READ ONLY;
ALTER USER <ユーザー名> READ ONLY;

#Read Onlyユーザーでは更新系SQLは実行できない
SQL> CREATE TABLE TEST(COL1 NUMBER);
ERROR at line 1:
ORA-28194: Can perform read operations only
```

```
#Read Writeに変更する場合
ALTER USER <ユーザー名> READ WRITE;
```

その他の新機能

- FIPS_140パラメータの統合
  - それぞれの機能で設定が必要だったFIPS_140-2の対応を一つのパラメータによって制御が可能に
  - 対応する機能: Transparent Data Encryption、DBMS_CRYPT、TLS、Native Network Encryption

```
$ORACLE_HOME/ldap/adminにfip.oraファイルを作成し、下記を追記する
FIPS_140=TRUE
```

- TDEのデフォルト暗号化アルゴリズムをAES256に変更
  - 従来は、列暗号化はAES192、表領域暗号化はAES128
- 大文字・小文字を区別しないパスワードのサポート終了
  - 23cにアップグレード後は、大文字・小文字を区別しないパスワードは使用できない
- 従来型の監査(AuditコマンドでOSやXML形式での監査)のサポート終了
  - 23cにアップグレード後は、設定済みの従来監査は機能はするが、新規や追加設定には対応せず削除のみ

Oracle Database セキュリティ新機能
参考リンク

SQL Firewall

Schema Privileges

Column Level Audit

Oracle Base DatabaseでTLS通信を有効化する

Azure ADのトークンでOracle Base Databaseに認証する

ありがとうございました