日本セキュアOSユーザ会主催
セキュアOS塾-04

SE-PostgreSQL
vs
Oracle Label Security

海外浩平 <kaigai@kaigai.gr.jp>

田口裕也 <yuya.taguchi@gmail.com>

日本セキュアOSユーザ会
Japan Secure Operating System Users Group since 2007

勉強会のきっかけ

- セキュアOS塾-03の懇親会にて…
- 海外『次の勉強会のテーマ、SE-PostgreSQLでどうだろ？』
- 田口「あー、いいっすねぇ〜」
- 『場所、オラクルさんの会議室とか借りたら刺激的じゃない？』
- 「たぶんできると思いますよ。最近は色々やってるみたいだし」
- 『じゃあそれで（多分に酒の勢いも）。ある意味超アウェー。』
- 「それだと、自分も何かしゃべらない訳にはいかないなぁ…」
- 『なら、SE-PostgreSQL vs Oracle Label Securityで対決とか』
- 「じゃあ、それでやってみますか」

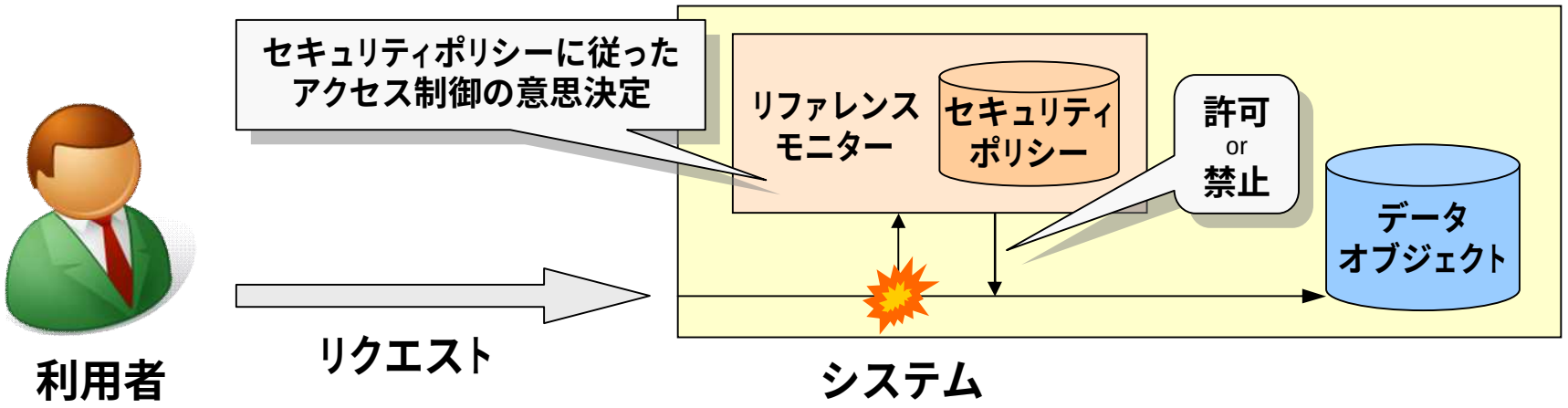
- このような流れで開催の運びとなりました。
  会議室を提供していただきました日本オラクル株式会社様の
  太っ腹ぶりに深く感謝いたします。m(_ _)m

本日のアジェンダ

1. **イントロダクション** … (**海外**)
2. SE-PostgreSQL**の概要** … (**海外**)
3. Oracle Label Security**の概要** … (**田口**)
4. **さぁ、比べてみようか** … (**海外/田口**)
5. SE-PostgreSQL**の強み** … (**海外**)
6. Oracle Label Security**の強み** … (**田口**)
7. **まとめ** … (**海外/田口**)
8. **質疑応答** … (**皆さん**)

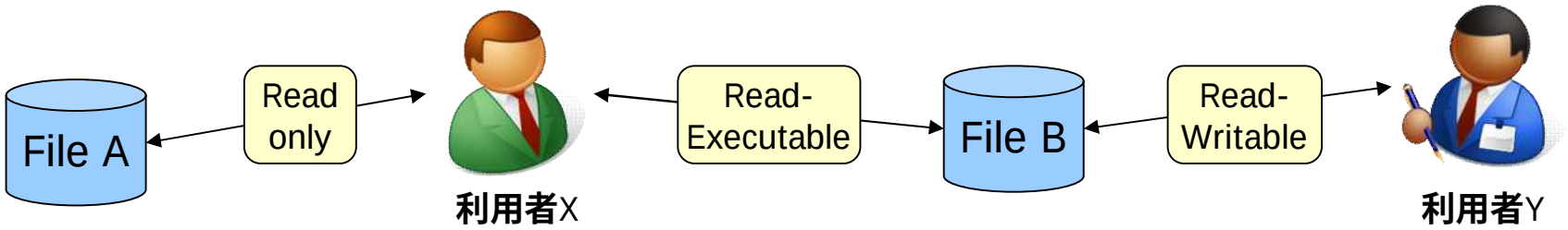
1. イントロダクション

リファレンスモニタ



- “セキュア”であるためには？
  - アプリケーションにバグ/脆弱性が無ければよい。（でも検証できない！）
  - チェックすべき箇所の最小化が重要
- リファレンスモニタ
  - 利用者のシステムに対するリクエストを例外なく捕捉、セキュリティポリシーに基づいて意思決定を行う
    - チェック漏れの可能性を極小化　… 網羅性の担保
    - セキュリティポリシーの集中管理　… 一貫性の担保

アクセス制御とは？



アクセス制御マトリックス

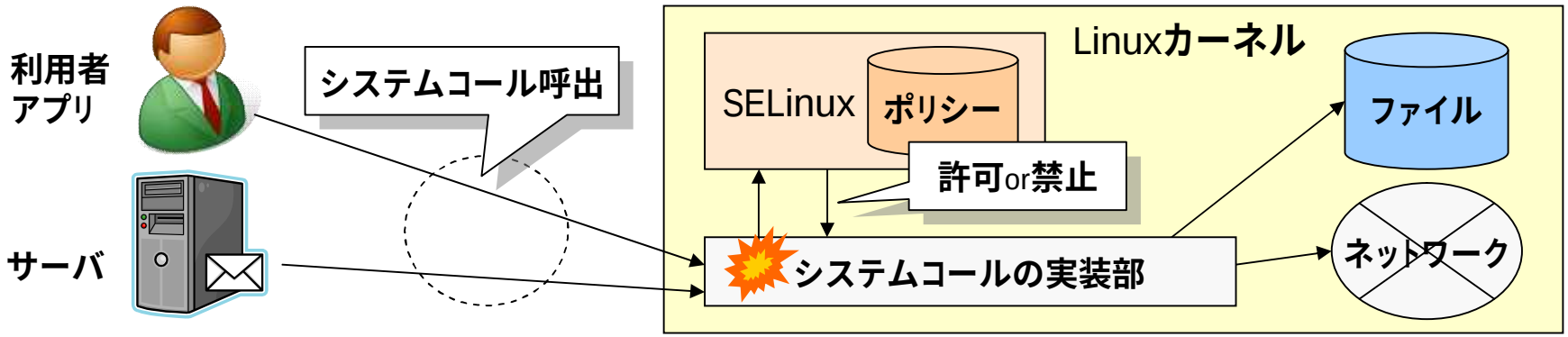
	File A	File B
利用者X	(Read)	(Read, Execute)
利用者Y	(-)	(Read, Write)

- 『誰が(Subject)』『何に(Object)』『何をできる(Action)』かを決める事
- 後は細かいアレンジ
  - “誰(Subject)”とか”何(Object)”を識別する方法
    - UserID? ファイル名? Security context?
  - “何をできる(Action)”かの種類
    - Read, Write, Execute? もっと細かく必要?

任意アクセス制御 と 強制アクセス制御

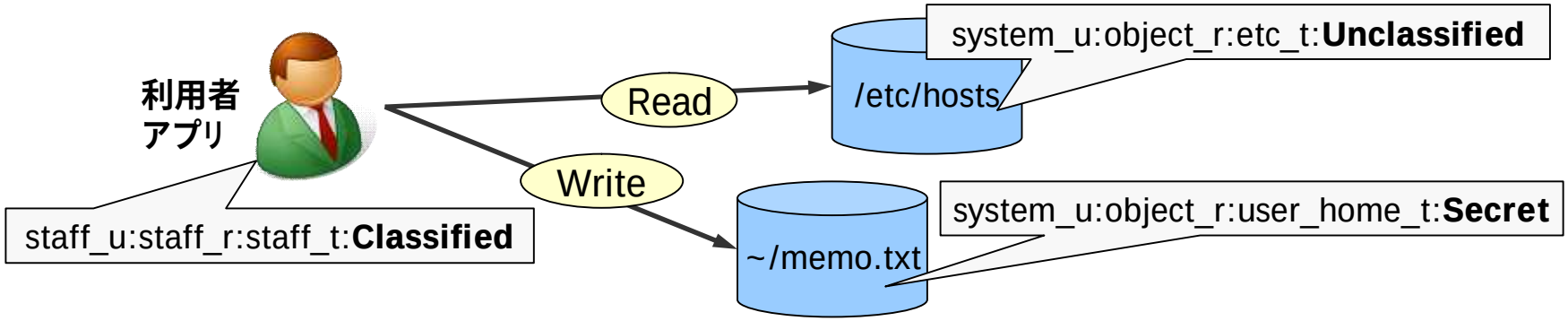
- アクセス制御とは
  - 『誰が(Subject)』『何に(Object)』『何をできる(Action)』かを決める事
  - ルールを決めるのは誰だ？
- 任意アクセス制御 (Discretionary Access Control)
  - 資源の所有者がルールを決める
    - 悪意の内部犯に対しては無力
  - 特権ユーザ(root)はチェックを回避できる
- 強制アクセス制御 (Mandatory Access Control)
  - 一元管理されたセキュリティポリシーがルールを決める
    - セキュリティポリシー = 『誰が何に何をできる』の巨大な集合
  - 全てのチェックを、漏れなく/例外なく実行する
    - 例外なく = 特権ユーザ(root)も含む

SELinux (1/2)

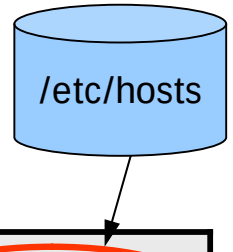


- SELinuxとは
  - Linuxカーネルにおけるリファレンスモニタ
    - カーネルへのリクエストは全てシステムコールを経由する
  - システムコールに対する強制アクセス制御を提供
- 特徴
  - 既存のアクセス制御メカニズムとは直交に機能する
  - セキュリティコンテキスト(セキュリティモデル上の識別子)と、
    セキュリティポリシー(アクセス制御のルール集)に基づくアクセス制御
    - 利用者⇔システム資源間の、あらゆる関係を記述できる
  - Linuxカーネル、RedHatEL/Fedora/etc...の標準機能

SELinux (2/2)



- セキュリティコンテキスト
  - 共通の形式で記述されるセキュリティ上の属性
  - プロセス、ファイル、ソケット、etc... あらゆるシステム資源に関連付け
    - セキュリティコンテキストが付く限り、SELinuxのモデルを適用できる
- セキュリティポリシー
  - 誰が(Subject)、何に(Object)、何をできるか(Action)の組
  - セキュリティコンテキストを用いてSubject/Objectを識別

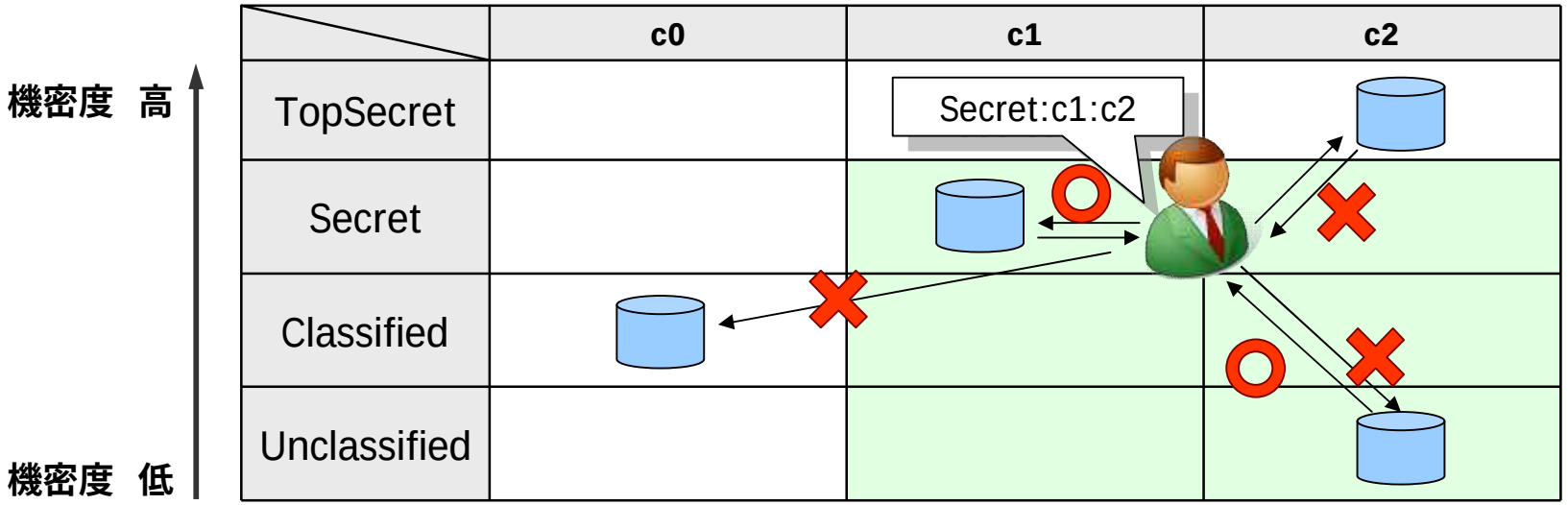


利用者 \ ファイル	Secret	Classified	Unclassified
Secret	(Read, Write)	(Read)	(Read)
Classified	-	(Read,Write)	(Read)
Unclassified	-	-	(Read,Write)

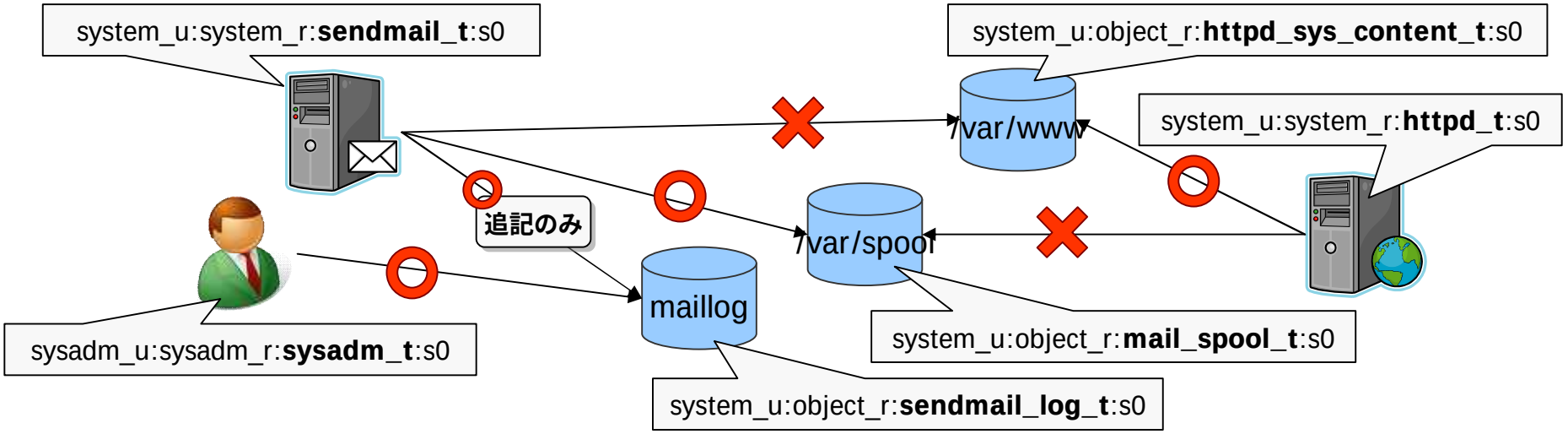
Multi Level/Category Security

- Multi Level/Category Security
  - **最も伝統的なアクセス制御ルール**
  - SELinux**もサポート。商用**UNIX、Oracle Label Security**も**
- **ルール**
  - **自分より機密度の高いデータは** Read **不可**
  - **自分と機密度の異なるデータへは** Write **不可**
  - **自分とカテゴリの異なるデータへはアクセス不可**

機密情報を、
低いレベルに流さない
= 情報フロー制御



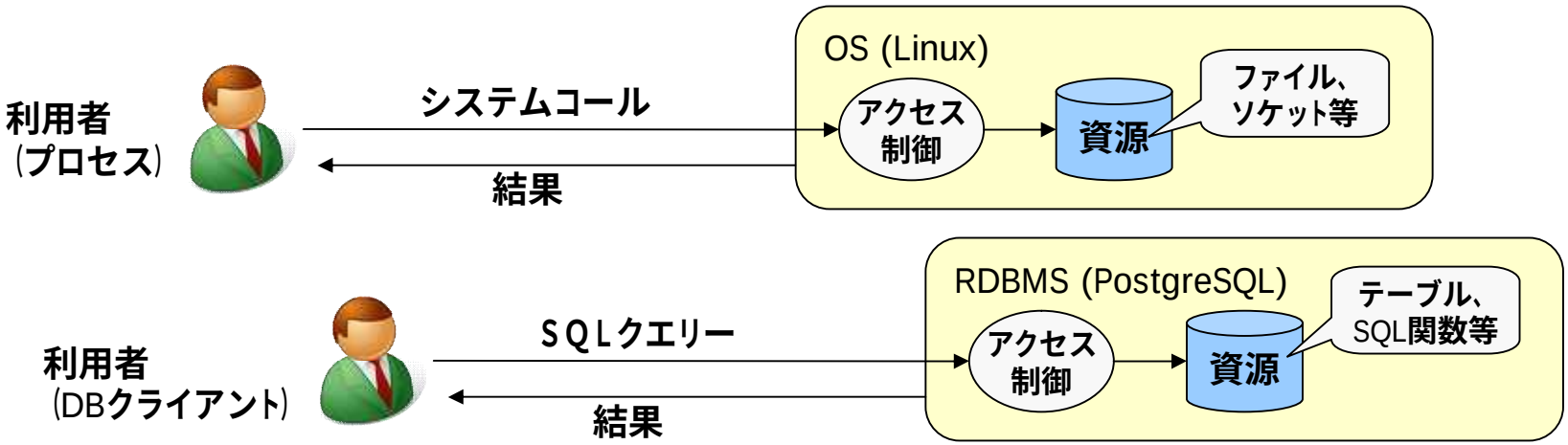
TE (Type Enforcement)



- Type Enforcement
  - SELinuxの特徴的なアクセス制御方式
  - MLS/MCSよりも柔軟なルールの記述が可能
    - それ故、標準ポリシーが未整備な頃は、設定項目が多かった…。
- ルール
  - セキュリティコンテキストの3番目のフィールド(タイプ/ドメイン)を使用
  - 個々のドメイン⇔タイプの間に、個別にアクセス制御ルールを設定できる
    - 適切な定義済みタイプを割り当てる事で、アクセス制御の設定を行う

2. SE-PostgreSQLの概要

OSとDBのアナロジー

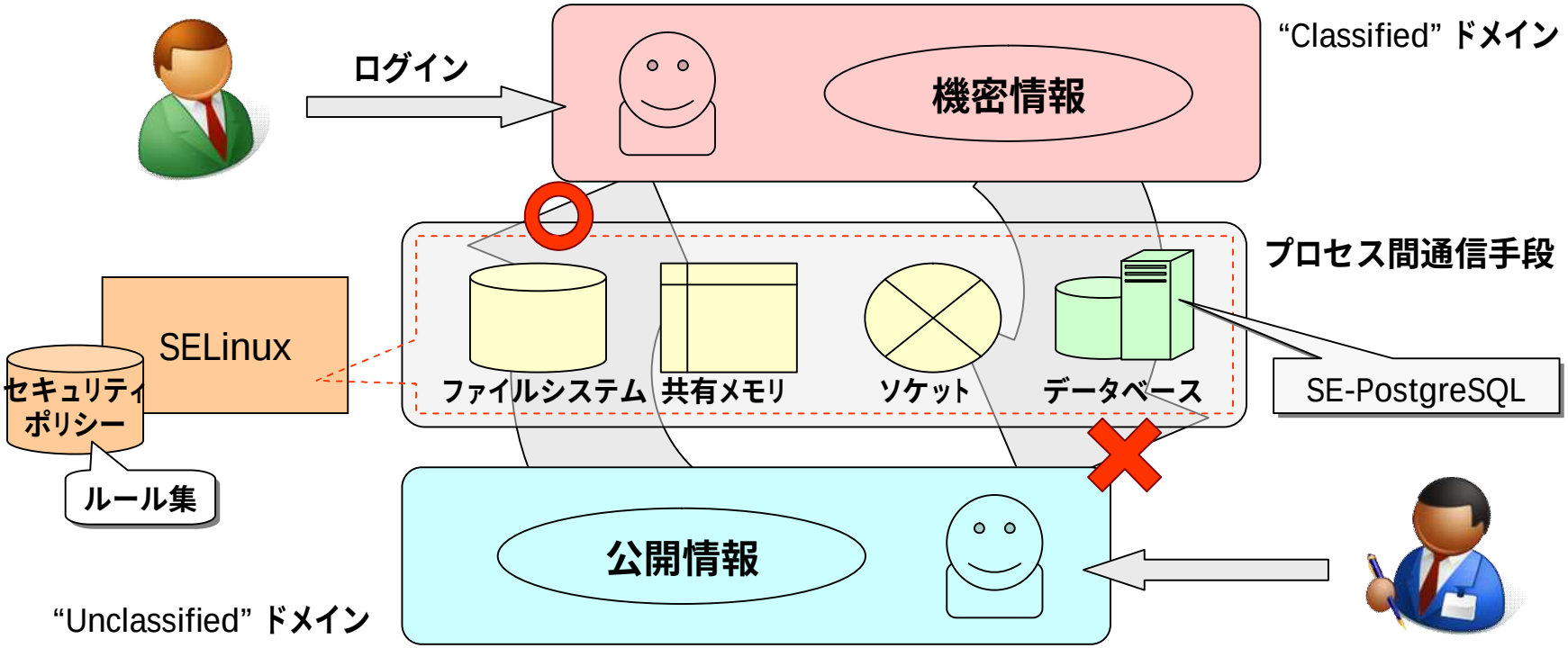


- **アナロジー** (analogy)　…〔**2つのモノの**〕**類似点**
- **OSの場合**
  - **利用者は、システムコールを用いて資源にアクセスする**
  - **OSは処理結果**(or **エラー**)**を利用者に返す**
- **RDBMSの場合**
  - **利用者は、SQLクエリーを用いて資源にアクセスする**
  - **RDBMSは処理結果**(or **エラー**)**を利用者に返す**
- **ポイント**
  - **対象となる資源、アクセス手段が異なるものの、アクセス制御モデルは共通**
  - **『誰が**(Subject)**、何に**(Object)**、何をできるか**(Action)**』をコントロール**

SE-PostgreSQLのアイデア

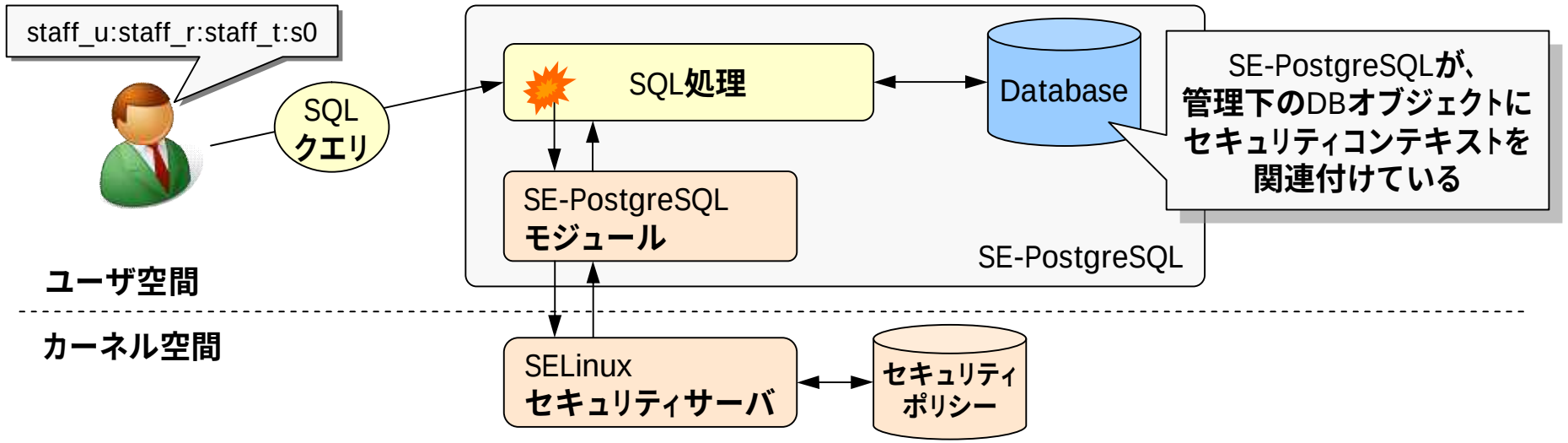
- SELinuxのセキュリティポリシーを利用して、DBへのアクセスに対して強制アクセス制御を行う
- メリット
  - システム全体で、アクセス制御の一貫性を担保できる
    - ✓ DBに保存しようがFSに保存しようが、”機密情報”は”機密情報”として扱う
    - ✓ 最初に認証すれば、利用者の権限はOSでもDBでも共通に
  - DB特権ユーザに対する制御
    - ✓ root同様に ”何でもできる人” を排除できる
- 何が必要か？
  - SELinuxのアクセス制御モデルにあてはめる
    - 誰が(Subject) … 接続元プロセスの権限
    - 何に(Object) … 対象のデータベースオブジェクト（表、列など）
    - 何をできる(Action) … DB用のパーミッションを定義
  - SELinuxの意思決定に基づいて、DBのアクセス制御を実施

概念図



- FSもDBも、プロセス間通信手段の一つにすぎないと考える
- SELinuxはプロセス間通信を介した情報フローを制御する
  - ✓ 機密度の高い情報が、それを扱う資格のない人に漏えいしない
- プロセス間通信手段が違っても、アクセス制御の結果が違うのは変

SE-PostgreSQLのアーキテクチャ



- **セキュリティコンテキストの管理**
  - **利用者** ... SELinuxのAPIを利用、**接続元プロセス**のモノを取得
  - DBオブジェクト ... SE-PgSQLが個々のDBオブジェクトに関連付け
- **アクセス制御の意思決定**
  - 重要なポイントにセキュリティフックを挿入、SE-PgSQLモジュールを呼び出し
  - SQLクエリによる要求を許可すべきか否か、SELinuxに問合わせ
  - SELinuxは自身のセキュリティポリシーを探索して意思決定
  - その結果に基づき、SE-PgSQLは「許可」または「禁止」のステータスを返す

SE-PostgreSQLの動作例

```
postgres=# CREATE TABLE drink (
    id      int primary key,
    name    text,
    price   int
) SECURITY_CONTEXT = 'system_u:object_r:sepgsql_ro_table_t:s0';
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "drink_pkey" ...
CREATE TABLE
```

セキュリティコンテキストを指定できる

```
postgres=# SELECT * FROM drink;
 id | name  | price
----+-------+-------
 10 | water |   120
(1 row)

postgres=# INSERT INTO drink VALUES (11, 'coke', 130);
ERROR:  SELinux: security policy violation
```

DB特権ユーザの権限で接続している

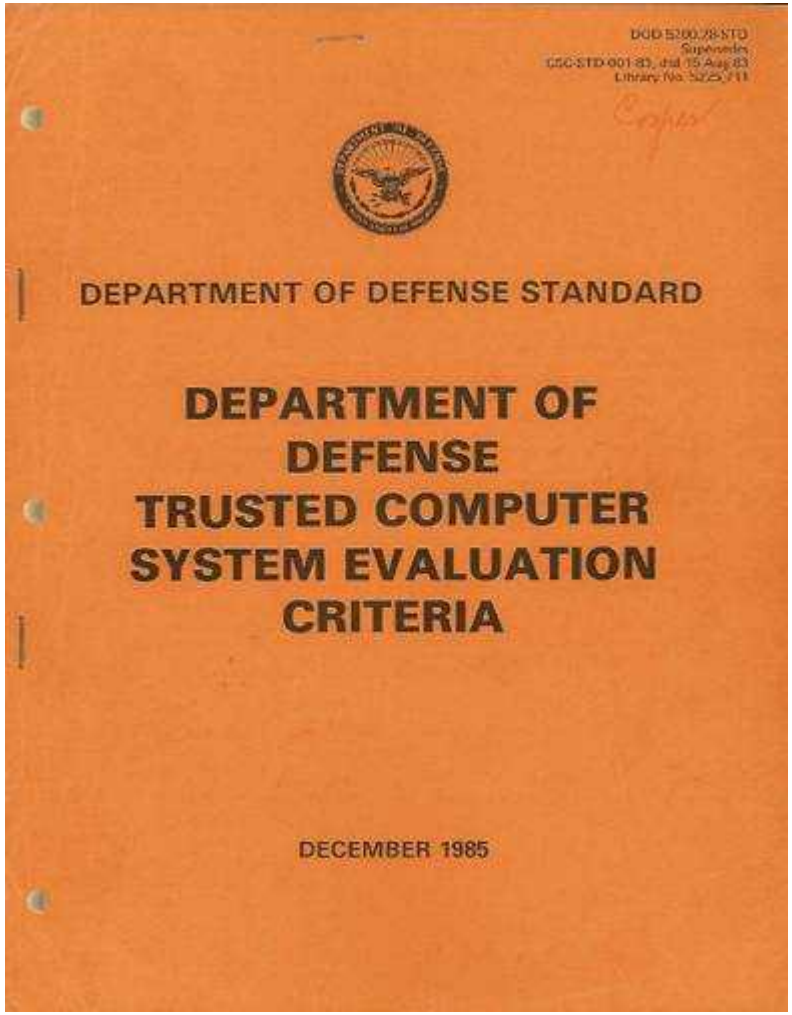
sepgsql_ro_table_t (読込み専用) なので、SELECT可能だが、UPDATEはエラー

```
postgres=# SELECT test_func(2);
ERROR: SELinux: security policy violation
```

テーブル、カラム、タプルだけでなく、SQL関数、スキーマ等に対しても アクセス制御を行うことが可能

3. Oracle Label Securityの概要

TCSEC （オレンジブック）



TCSECとは　http://itpro.nikkeibp.co.jp/word/page/10005255/

Trusted Oracleの歴史は古く

UNITED STATES
DEPARTMENT OF DEFENSE

Certificate of Appreciation
Presented to -

Oracle Corporation

This certificate is presented in recognition of the contribution you have made in the Computer Security Effort through your success in developing a product,
Trusted Oracle7 (B1)

which meets the criteria for placement on the Evaluated Products List for Trusted Computer Systems.

Director
NATIONAL COMPUTER SECURITY CENTER

1994年4月7日にTCSEC B1認定を取得
http://www.oracle.com/technology/deploy/security/seceval/pdf/tcsec_to7013.gif

Trusted Oracleの歴史は古く

shortcuts | GETTING STARTED | DOWNLOADS | DOCUMENTATION | FORUMS | ARTICLES | SAMPLE CODE | TUTORIALS

Oracle and the US TCSEC

Oracle pursued and successfully completed TCSEC evaluations of its Oracle7 and Trusted Oracle7 database server products.

Oracle does not participate in the TCSEC any longer as TCSEC has been superseded by Common Criteria.

For a matrix of Oracle security evaluations currently in progress as well as those completed please read the go to Oracle Security Evaluations Status.

Overview of the TCSEC

Published first in 1983, the US Trusted Computer System Evaluation Criteria (TCSEC, also known as the "Orange Book") has been used since then for the evaluation of operating systems. In April 1991, the US National Computer Security Center (NCSC) published the Trusted Database Interpretation (TDI) which sets forth an interpretation of these evaluation criteria for database management systems and other layered products. Products are evaluated against the TCSEC and the TDI at predefined classes from D (lowest) up to A1 (highest).

Evaluation Status

Product	Class	Certificate
Oracle7 Database Server, Release 7.0.13.1 (No longer supported)	C2	Certificate
Trusted Oracle7 Database Server, Release 7.0.13.1 (No longer supported)	B1	Certificate

Oracle7:C2，Trusted Oracle 7:B1

http://www.oracle.com/technology/deploy/security/seceval/oracle_tcsec-validations.html

Oracle Database セキュリティの進化

Oracle Database 11g
- Data Masking
- TDE Tablespace Encryption
- Oracle Total Recall
- Oracle Audit Vault

Oracle Database 10g
- Oracle Database Vault
- Transparent Data Encryption (TDE)
- Real Time Masking

Oracle Database 9i
- Secure Config Scanning
- Fine Grained Auditing
- Oracle Label Security

Oracle8i
- Enterprise User Security
- Virtual Private Database (VPD)
- Database Encryption API
- Strong Authentication

Oracle7
- Native Network Encryption
- Database Auditing
- Government customer

Trusted Solarisとの組み合わせ

Trusted Solaris™ 7 Technical White Paper

White Paper

Sun microsystems

Trusted Oracleの例

前の例とは反対に、Trusted Oracle などのトラステッド・アプリケーションを実行するには、複数レベルで操作できるアプリケーションの単一のコピーを実行する必要があります。

Trusted Oracle をトラステッド・アプリケーションとして実行すると、Trusted Solaris の必須アクセス制御方針をバイパスする特権が与えられます。これは、システム全体のセキュリティ方針を妨げることなく、これらの特権を使用するように信頼されているためです。Trusted Solaris は、ファイルなどのオブジェクトに対する必須アクセス制御を実行しますが、Trusted Oracle は、Trusted Solaris オブジェクト内の行などのデータベース・オブジェクトに対して必須アクセス制御を実行します。このため、Trusted Oracle のプロセスは、Trusted Solarisオブジェクト内のデータベースオブジェクトに対して必須アクセス制御を実行するように、特権が与えられ、信頼されている必要があります。

Trusted Oracle は、トラステッド・アプリケーションであるため、ユーザーのセキュリティ・クリアランスによって、ユーザーが、データベースの下位レベルのデータの読み取り、上位レベルのデータへの書き込み、下位レベルのデータへの書き込みができるようにする機能などの特権処理を提供します。この機能によって、管理者や特権ユーザーは、データベース全体のエクスポートおよびインポート、または情報の再ラベリングなどの必要な管理機能を実行できます。これらの機能は、トラステッド・データベース・アプリケーションの実行も支援します。

特権検出モニタ

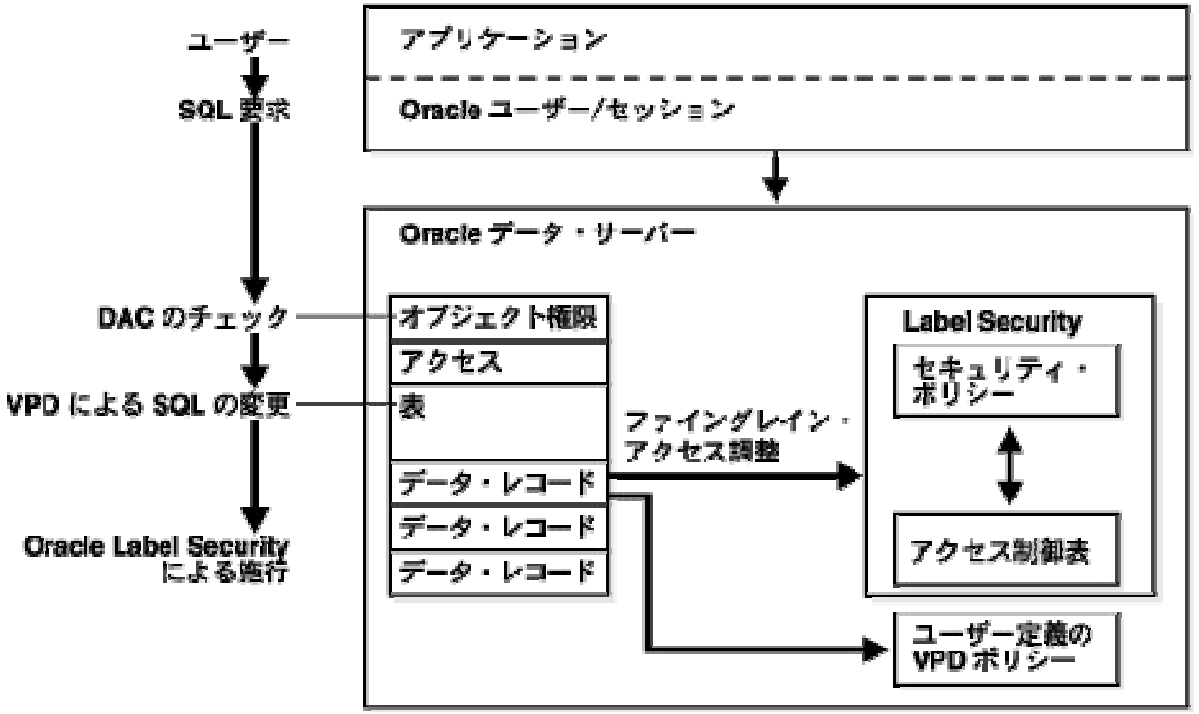
特権検出モニタにより、サードパーティのプログラムに必要な特権を速やかに判別することができます。これは、"rumpd（run privilege detector）"によって起動され、通常はシステム管理者が、独立した開発システムで実行します。特権検出モニタを使うには、特殊デバッ

31

http://jp.sun.com/products/wp/solaris/T_Sol7WP.pdf

Oracle Label Security

- OLSはVPDアーキテクチャ上で動作するLBAC



4. さぁ、比べてみようか

日本セキュアOSユーザ会
Japan Secure Operating System Users Group since 2007

Oracleの行レベルアクセス制御

- Virtual Private Database (VPD)
  - Oracle Database Enterprise Editionの標準機能
  - **ルールの定義によるアクセス制御**

- Oracle Label Security (OLS)
  - Oracle Database Enterprise Editionのオプション製品
  - **データ・ラベルの定義によるアクセス制御**

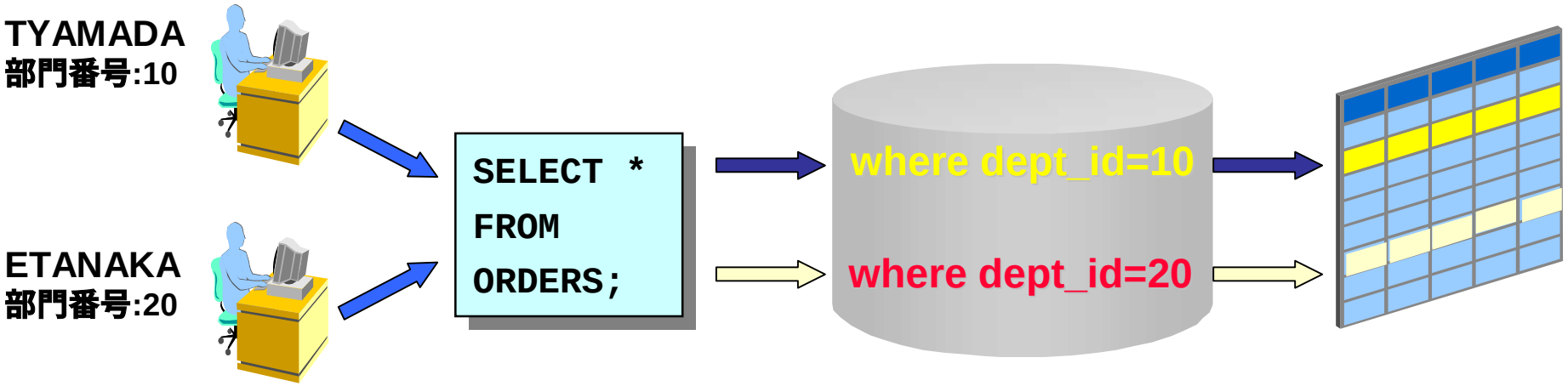
・開発の目的は、政府や国防など非常に機密性の高い情報を扱い、
データを完全に分離して格納しなければならない場面で利用 (MLSの実現)

・Label Based Access Contorol(LBAC)：

表の各行にラベルと呼ばれるセキュリティ情報を格納する列を追加し、
各ユーザーが持つラベルと比較することにより、アクセス制御を実現

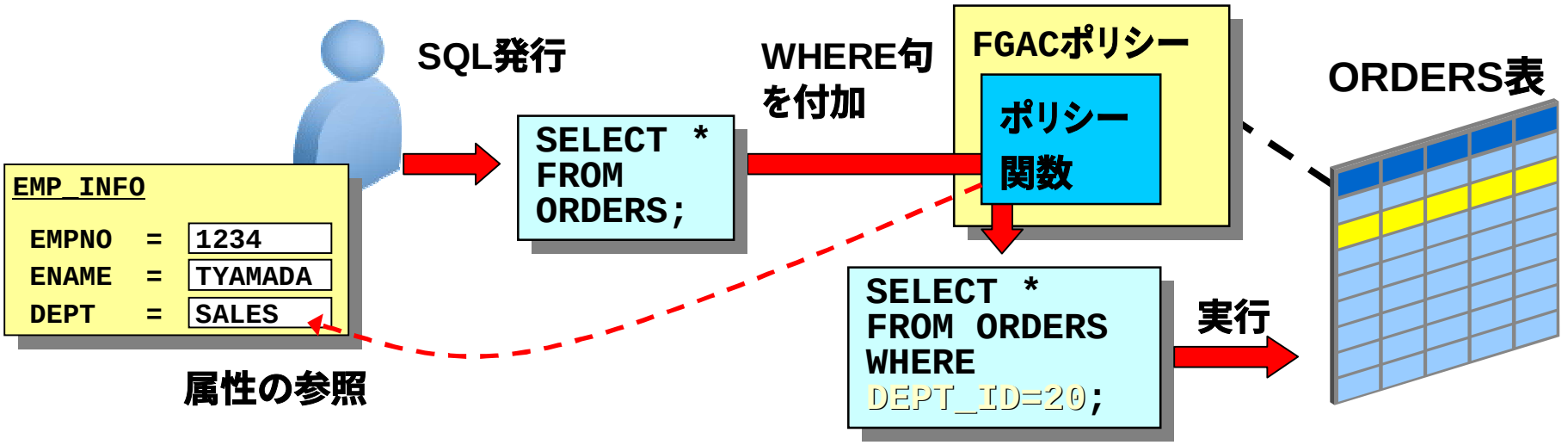
Virtual Private Database (VPD)

- VPD = アプリケーション・コンテキスト ＋ FGAC
  - アプリケーション・コンテキスト
    - ユーザー識別情報を保持するしくみ
  - ファイングレイン・アクセス・コントロール (FGAC)
    - WHERE句を生成／付加するしくみ



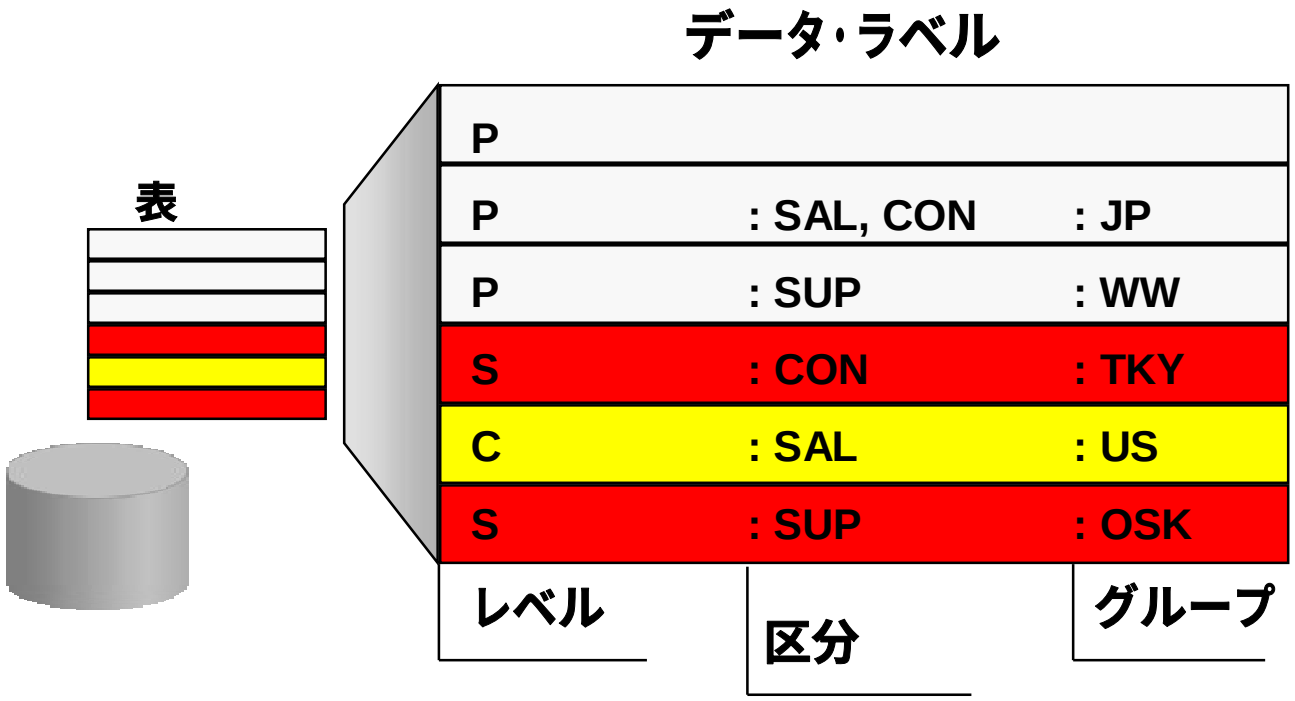
ファイングレイン・アクセス・コントロール (FGAC)

- 表に付加されたFGACポリシーに基づいて、SQL文にWHERE句を生成／付加する
- FGACポリシーに、WHERE句の生成ロジックをポリシー関数 (PL/SQLファンクション) として作成



データ・ラベル

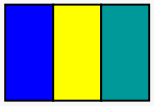
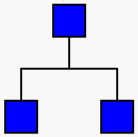
- 表の各行に格納されるラベル
  - ラベル列は、物理的に各行に格納される
  - ラベル自体は、ユーザーから参照できなくすることも可能



ラベルの構成要素

ラベル = レベル : 区分 : グループ

ラベルの例
S:SAL:JP
C:SUP,CON:TKY

要素	説明	例 (短縮名)
レベル 	データの重要度・機密度をあらわします。数値による、直線型の階層構造をもち、大きな数値を持つレベルほど重要なデータであることを示します。レベルは必須項目であり、すべてのラベルはレベルを持ちます。	SENSITIVE(S) CONFIDENTIAL(C) PUBLIC(P)
区分 	データのカテゴリーを表します。データの種類ごとの分割などをおこない、操作対象を分割することができます (オプション)。	SALES(SAL) SUPPORT(SUP) CONSULTING(CON)
グループ 	データにツリー型の階層構造を持たせることができます。上位の階層のグループをユーザー・ラベルに持っている場合、下位の階層のグループのラベルをデータ・ラベルとして持つデータにもアクセス可能となります (オプション)。	JAPAN(JP) TOKYO(TKY) OSAKA(OSK)

ラベル要素の例

レベル

短縮名	通常名	番号
S	SENSITIVE	80
C	CONFIDENTIAL	50
P	PUBLIC	1

重要

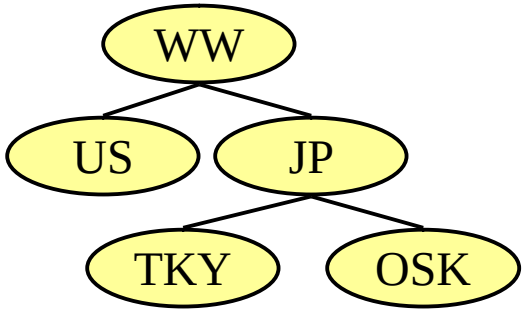
区分

短縮名	通常名	番号
SAL	SALES	10
SUP	SUPPORT	20
CON	CONSULTING	30

レベルのみ、番号の大小に意味を持つ。
数値が大きいほど重要度が高い。

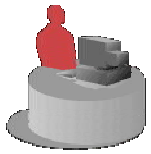
グループ

短縮名	通常名	番号	親グループ
WW	WORLD_WIDE	1000	
US	UNITED_STATES	1100	WW
JP	JAPAN	1200	WW
TKY	TOKYO	1210	JP
OSK	OSAKA	1220	JP



ラベルを使用したアクセス制御

OP1



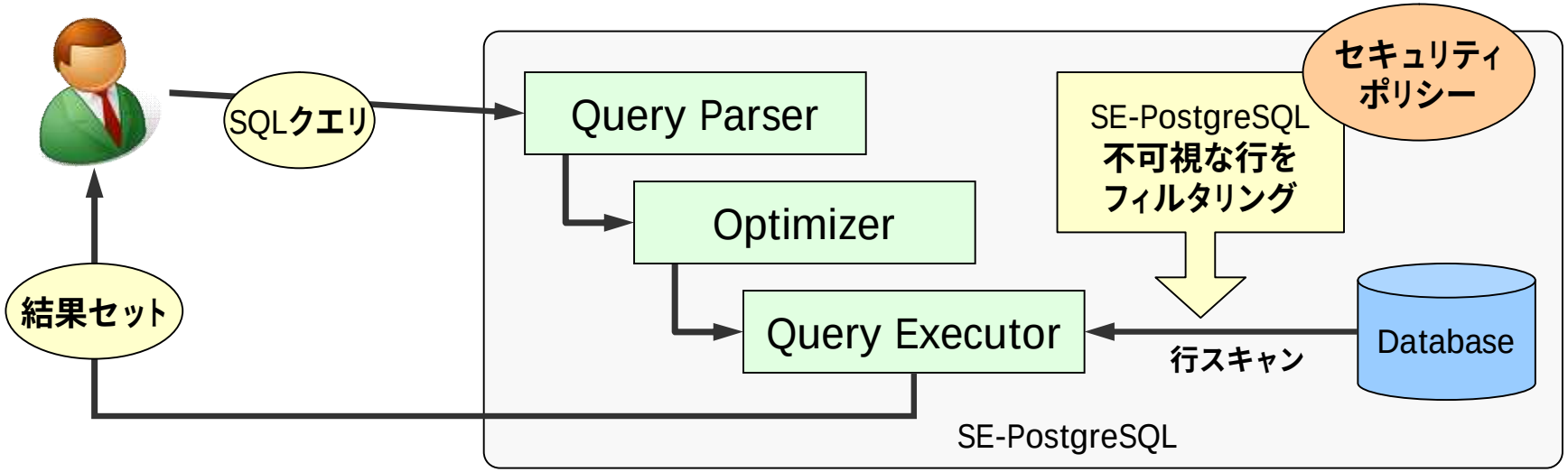
Common:Woman:Japan

OP2

Secret:Man:Osaka

ID	名前	ラベル	OP1	OP2
1	SCOTT	Common:Man:Tokyo	×	×
2	JAMES	Common:Man:Osaka	×	○
3	NANCY	Secret:Woman:Japan	×	×
4	MIKE	Secret:Man:Japan	×	×
5	JANE	Common:Woman:Tokyo	○	×
6	JO	Common:Woman:Osaka	○	×
7	SMITH	Common::	○	○

SE-PgSQL: 行レベルアクセス制御



- SQL処理の流れ
  - Query Parser … クエリ文字列を解析して内部形式に変換する
  - Optimizer … 最適なクエリ実行計画を作成する
  - Query Executor … 実行計画に基づいてテーブルをスキャンする
- WHERE句書き換えを使わない理由
  - Oracle社/IBM社の特許に抵触する可能性がある :-)
  - Optimizerに邪魔されたくない

行レベル制御と最適化の危ない関係

- PostgreSQLのユーザ定義関数
  - Optimizerへのヒントとして “重さ” を指定できる
  - WHERE句を評価する時に、“軽い”関数から先にチェックする
- どうなるか？

```
postgres=# CREATE VIEW v AS SELECT * FROM t WHERE b = current_user;
CREATE VIEW
postgres=# CREATE FUNCTION test_func(v) RETURNS bool LANGUAGE sql
               COST 0.0001 AS 'INSERT INTO s (x) VALUES ($1); SELECT true';
CREATE FUNCTION
postgres=# SELECT * FROM v WHERE test_func(v);
 a |   b
---+--------
 1 | kaigai
(1 row)
postgres=# SELECT * FROM s;
     x
------------
 (1,kaigai)
 (2,yuyat)
(2 rows)
```

ビューのWHERE句を評価するより先に、
ユーザ定義関数を実行している。
→ 表 s に、隠し行の内容がリーク

重要

PostgreSQLでは、行レベルセキュリティの目的で
ビューを利用した対象の絞込みは危険。
同じ理由で、WHERE句書換えも技術的にダメ

SE-PgSQL: 行レベルアクセス制御

```
postgres=# SELECT security_context, * FROM drink;
               security_context                | id |  name   | price
-----------------------------------------------+----+---------+-------
 system_u:object_r:sepgsql_ro_table_t          |  1 | water   |   100
 system_u:object_r:sepgsql_ro_table_t          |  2 | coke    |   120
 system_u:object_r:sepgsql_table_t             |  3 | juice   |   130
 system_u:object_r:sepgsql_table_t             |  4 | coffee  |   180
 system_u:object_r:sepgsql_table_t:Classified  |  5 | beer    |   240
 system_u:object_r:sepgsql_table_t:Classified  |  6 | sake    |   320
(6 rows)
```

- security_context **システム列**
  - **行単位で付与されたセキュリティコンテキストを表示/更新する**
- SELECT**したら**
  - Classified**権限のない人は、**”beer”**と**”sake”**がフィルタリングされる**
- UPDATE**したら**
  - Classified**権限のない人は、**”beer”**と**”sake”**を更新できない**
  - “water”**と**”coke”**は、読み込み専用なのでデータを更新できない**
- **実際には、あたかも**WHERE**句に条件が付加されたように動作する**

SE-PgSQL: セキュリティポリシー

```
postgres=# SELECT security_context, * FROM drink;
               security_context                | id |  name  | price
-----------------------------------------------+----+--------+-------
 sy                                            |  3 | tea    |   100
 sy                                            |  4 | coffee |   180
 system_u:object_r:sepgsql_table_t:Classified  |  5 | beer   |   240
 system_u:object_r:sepgsql_table_t:Classified  |  6 | sake   |   320
(4 rows)
```

3番目のフィールド
= Type Enforcementルール

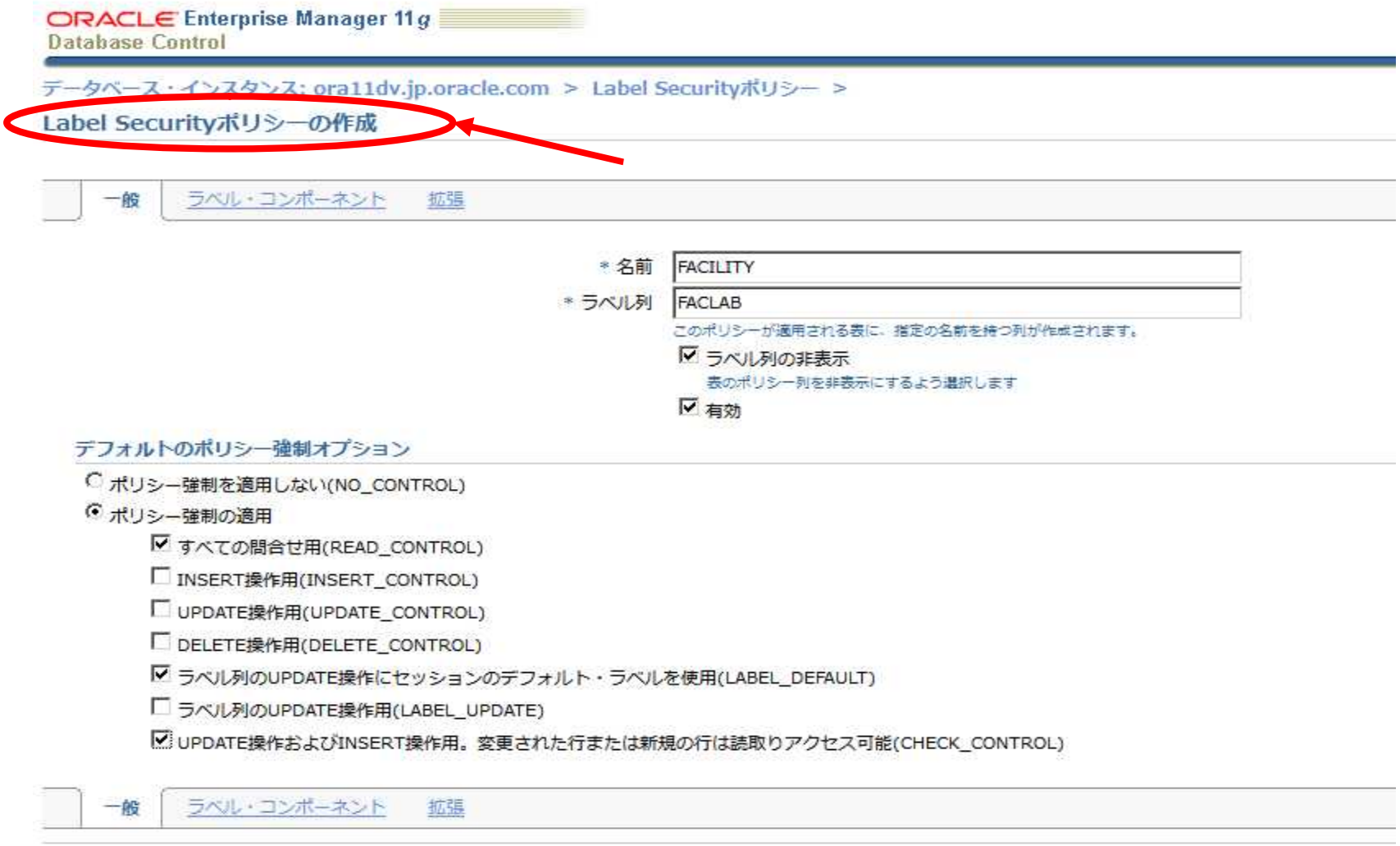
4番目のフィールド
= Multi Level Securityルール

- Type Enforcementルール
  - 個々のドメイン/タイプの間に、許可すべき権限セットが規定されている。
  - 定義済みラベル例： sepgsql_fixed_table_t, user_sepgsql_proc_exec_t
- Multi Level Securityルール
  - 利用者/DBオブジェクト間の上下関係/包含関係に基づく。
  - 定義済みラベル例： s0, s0:c1, s0:c0.c1023 (=SystemHigh/別名)
- 定義済みセキュリティポリシー
  - Fedora 9以降では標準パッケージに含まれる。RHEL6にも統合予定
  - 基本的に、ポリシー利用者が書くべきものではない。

OLS: ポリシーの作成


ORACLE Enterprise Manager 11g
Database Control
データベース・インスタンス: ora11dv.jp.oracle.com
ホーム
パフォーマンス
可用性
サーバー
スキーマ
データ移動
ソフトウェアとサポート
記憶域
制御ファイル
表領域
一時表領域グループ
データファイル
ロールバック・セグメント
REDOログ・グループ
アーカイブ・ログ
ASMに移行
ローカル管理表領域
データベース構成
メモリー・アドバイザ
自動UNDO管理
初期化パラメータ
データベース機能使用状況の検索
Oracle Scheduler
ジョブ
チェーン
スケジュール
プログラム
ジョブ・クラス
ウィンドウ
ウィンドウ・グループ
グローバル属性
自動化メンテナンス・タスク
統計管理
自動ワークロード・リポジトリ
AWRベースライン
リソース・マネージャ
スタート・ガイド
コンシューマ・グループ
コンシューマ・グループ・マッピング
プラン
設定
統計
セキュリティ
ユーザー
ロール
プロファイル
監査設定
Oracle Label Security
仮想プライベート・データベース・ポリシー
アプリケーション・コンテキスト

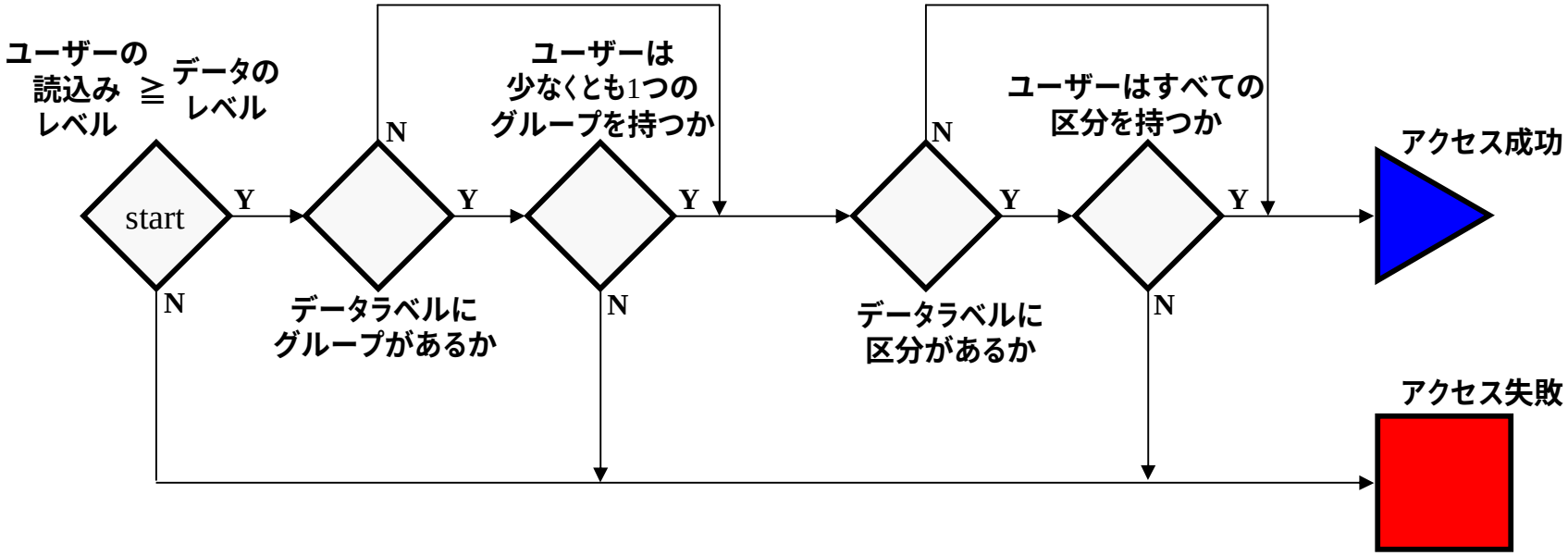
OLS: ポリシーの作成


ORACLE Enterprise Manager 11g
Database Control
データベース・インスタンス: ora11dv.jp.oracle.com > Label Securityポリシー >
Label Securityポリシーの作成
一般
ラベル・コンポーネント
拡張
* 名前
FACILITY
* ラベル列
FACLAB
このポリシーが適用される表に、指定の名前を持つ列が作成されます。
ラベル列の非表示
表のポリシー列を非表示にするよう選択します
有効
デフォルトのポリシー強制オプション
ポリシー強制を適用しない(NO_CONTROL)
ポリシー強制の適用
すべての問合せ用(READ_CONTROL)
INSERT操作用(INSERT_CONTROL)
UPDATE操作用(UPDATE_CONTROL)
DELETE操作用(DELETE_CONTROL)
ラベル列のUPDATE操作にセッションのデフォルト・ラベルを使用(LABEL_DEFAULT)
ラベル列のUPDATE操作用(LABEL_UPDATE)
UPDATE操作およびINSERT操作用。変更された行または新規の行は読取りアクセス可能(CHECK_CONTROL)
一般
ラベル・コンポーネント
拡張

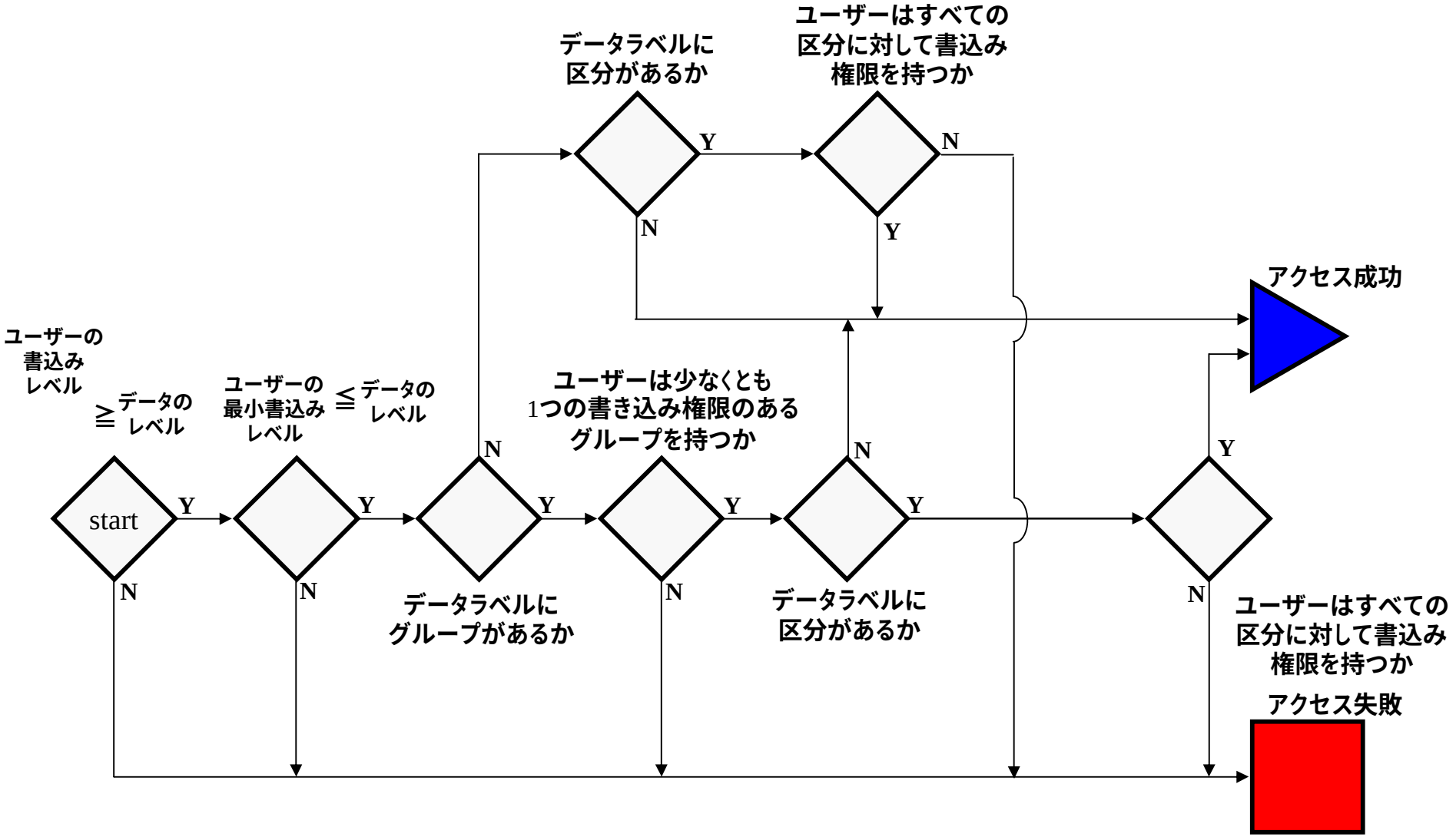
OLS: ポリシーの作成


ORACLE Enterprise Manager 11g
Database Control
データベース・インスタンス: ora11dv.jp.oracle.com > Label Securityポリシー >
Label Securityポリシーの編集: FACILITY
一般
ラベル・コンポーネント
拡張
レベル
レベルは、ラベル付けする情報の機密性を示すランキングです。情報の機密性が高いほどレベルも高くなります。どのラベルにもレベルを1つ設定する必要があります。レベル(および他の各ラベル・コンポーネント)には詳細名と
のは短縮名のみです。ラベル操作の際は、短縮名のみを使用します。
削除
すべて選択 | 選択解除
選択 詳細名 短縮名 数値タグ
PUBLIC P 1000
SENSITIVE SEN 2000
HIGHLY_SENSITIVE HS 3000
5行追加
区分

Oracle label Security の評価 (読込)



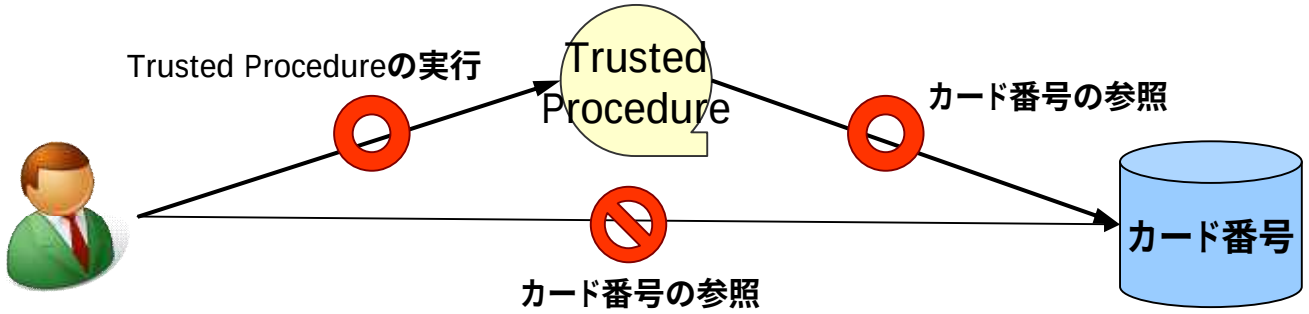
Oracle label Security の評価 (書込)



Trusted Procedure

- Trusted Procedureとは
  - 特定のSQL関数を実行した時に、利用者の権限を変更する
  - 類似機能：Security Definer関数
    - SQL関数の所有者にユーザIDを切り替えて動作
    - OS上のSetUIDプログラムに相当
- メリット
  - 特定のデータを、任意のSQLによるアクセスから隠し、アクセス手順を限定的にする事ができる。

例） クレジットカード番号をWebアプリのSQLで参照させたくない
しかし、決済時に伏字の番号を表示する必要がある



SE-PgSQL: Trusted Procedure

- SE-PostgreSQLのTrusted Procedure
  - SQL関数の実行時に、ドメイン遷移を発生させる事ができる。
  - 要は、一時的に利用者の権限を変更する仕組み

```
postgres=# CREATE FUNCTION show_credit(int) RETURNS text LANGUAGE sql
 SECURITY_CONTEXT = ‘system_u:object_r:sepgsql_trusted_proc_t:s0’
 AS 'SELECT regexp_replace(ccredit, ''[0-9]+-'', ''xxxx-'', ''g'')
                                    FROM customer WHERE $1 = uid';
CREATE FUNCTION
```

伏字にして呼び出し元に返す

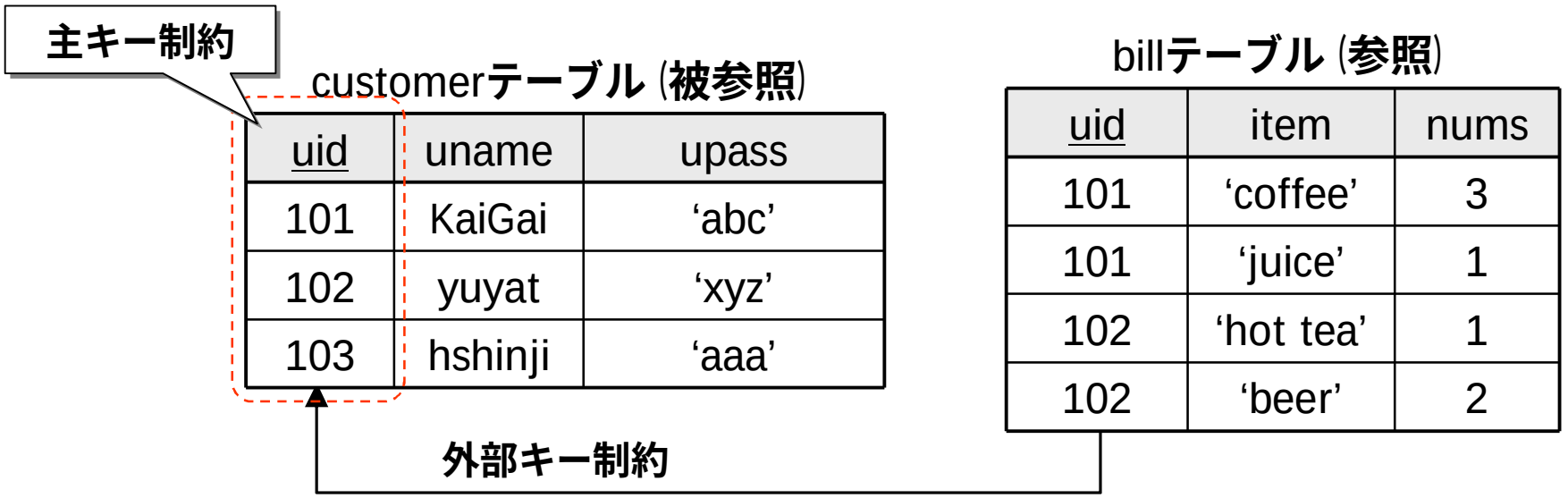
```
postgres=# SELECT uid, uname, ucredit FROM customer;
ERROR:  SELinux: security policy violation

postgres=# SELECT uid, uname, show_credit(uid) FROM customer;
 uid | uname  |     show_credit
-----+--------+---------------------
 101 | KaiGai | 1111-xxxx-xxxx-xxxx
 102 | yuyat  | 5555-xxxx-xxxx-xxxx
(2 rows)
```

OLS: Trusted Procedure

- トラステッド・ストアド・プログラム・ユニット
  - Oracle Label Security権限が付与されている
    ストアドプロシージャ、ファンクション、パッケージのこと。
  - ユーザーに対して認可の範囲を超えるアクセスを許可する場合に使う
    - 上位ラベルのCONFIDENTIALを持つユーザーが下位ラベルのSENSITIVE行へデータの挿入が業務上どうしても必要なときには、トラステッド・ストアド・プログラムにWRITEUP権限を付与し、ユーザーはトラステッド・ストアド・プログラムを経由して実行することができる。

OLS: Trusted Procedure

制約と行レベル制御 (1/2)



- **主キー制約、UNIQUE制約**
  - **特定の列において、複数の行が同じ値を持てない**
- **外部キー制約**
  - **参照側テーブルに存在する値は、必ず被参照テーブルにも存在する**
  - **参照/被参照関係を破壊するような更新削除を禁止する**

制約と行レベル制御 (2/2)

- 問題は？
  - “見えない行”と重複する主キーや、参照/被参照の関係
  - “見えない行”が存在する事を推測できてしまう
  - Covert Channel (隠れチャネル)と呼ばれる現象
- SE-PostgreSQLでは
  - 制約を破壊するようなデータ操作には、即座にエラーを返す
  - 主キー/外部キー制約を守った上での、行レベル制御
  - Covert Channel対策は対象外
- Oracle Label Securityでは
  - あまり考慮されていない予感
  - ラベル列を隠すオプション (HIDE) はある

SE-PgSQL: 監査ログ機能

- 監査ログ機能
  - 誰が何に何をしたか、記録保存するための機能
  - セキュリティ事故が発生した際に、後で確認する事ができる
- SE-PostgreSQLでは
  - OSと同じ形式でログファイルに出力
    (将来的には Linux標準 auditd との統合を予定)
  - ファイルに書き出されたログの保全はOS(SELinux)の役割

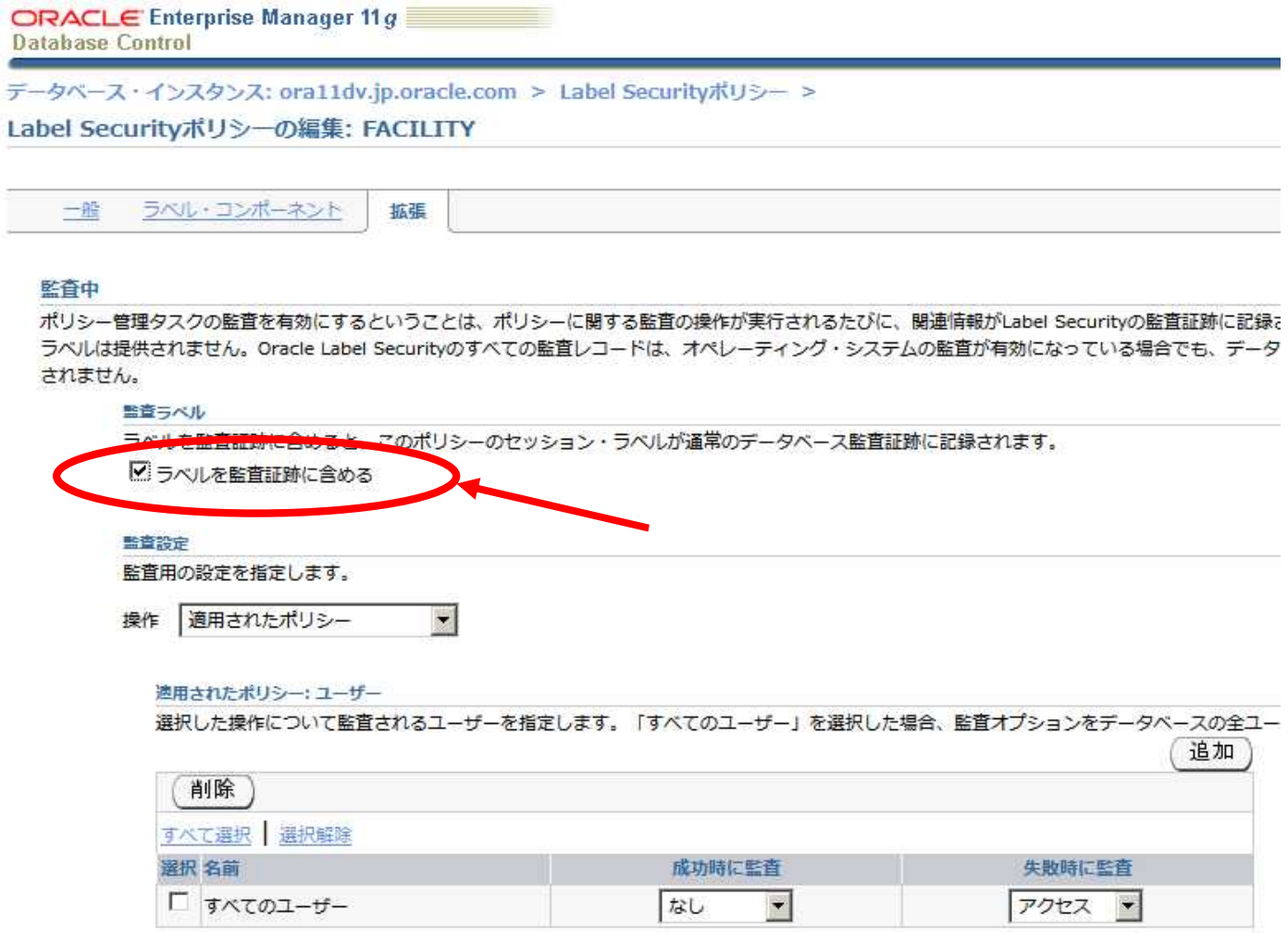
```
LOG:  SELinux: denied { insert }
  scontext=unconfined_u:unconfined_r:sepgsql_test_t:SystemLow-SystemHigh
  tcontext=system_u:object_r:sepgsql_ro_table_t:SystemLow
  tclass=db_table name=drink
STATEMENT:  INSERT INTO drink VALUES (11, 'coke', 130);
           :
LOG:  SELinux: granted { execute }
  scontext=unconfined_u:unconfined_r:unconfined_t:SystemLow-SystemHigh
  tcontext=unconfined_u:object_r:sepgsql_proc_exec_t:s0
  tclass=db_procedure name=sepgsql_getcon
STATEMENT:  SELECT sepgsql_getcon();
```

OLS: 監査ログ機能

- OLSでは
  - **標準的なOracle Databaseの監査機能の拡張**

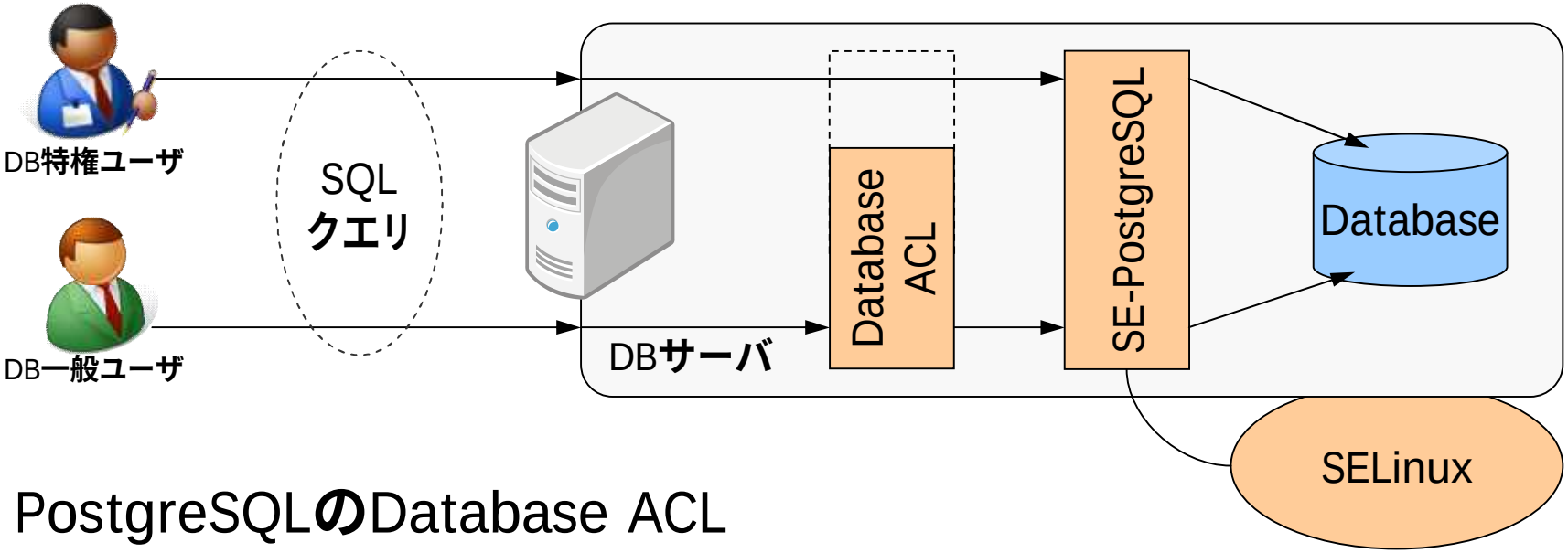
    Oracle Label Security**独自の管理操作およびポリシー権限の使用を記録できる**
  - **データベースの監査機能を使うので、監査を有効にするパラメータAUDIT_TRAILを設定すること**

OLS: 監査ログ機能


ORACLE Enterprise Manager 11g
Database Control
データベース・インスタンス: ora11dv.jp.oracle.com > Label Securityポリシー >
Label Securityポリシーの編集: FACILITY
一般
ラベル・コンポーネント
拡張
監査中
ポリシー管理タスクの監査を有効にするということは、ポリシーに関する監査の操作が実行されるたびに、関連情報がLabel Securityの監査証跡に記録さ
ラベルは提供されません。Oracle Label Securityのすべての監査レコードは、オペレーティング・システムの監査が有効になっている場合でも、データ
されません。
監査ラベル
このポリシーのセッション・ラベルが通常のデータベース監査証跡に記録されます。
ラベルを監査証跡に含める
監査設定
監査用の設定を指定します。
操作
適用されたポリシー
適用されたポリシー: ユーザー
選択した操作について監査されるユーザーを指定します。「すべてのユーザー」を選択した場合、監査オプションをデータベースの全ユー
追加
削除
すべて選択
選択解除
選択
名前
成功時に監査
失敗時に監査
すべてのユーザー
なし
アクセス

5. SE-PostgreSQLの強み

SE-PgSQL: 強制アクセス制御

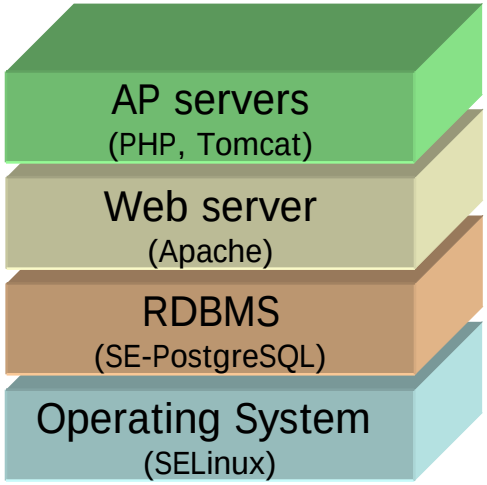


- PostgreSQLのDatabase ACL
  - DB特権ユーザ (superuser) はアクセス制御を回避
  - オブジェクトの所有者が任意にGRANT/REVOKEできる
- SE-PostgreSQLの強制アクセス制御
  - 特権ユーザを含む、全ての利用者に例外なくアクセス制御を適用
  - セキュリティポリシーは、アクセス権の変更も含めて集中制御
- OS上のUNIX Permission/SELinuxの関係に酷似

LAPP/SELinux (1/2)

- LAPP: Linux, Apache, PostgreSQL, PHP/Perl
  - **典型的なOSS-Webアプリケーションスタック**
- **現状**
  - **攻撃の大半はWebアプリケーションに対して（90%以上！）**
  - **個々のWebアプリの品質に依存したセキュリティには限界**
- **問題点**
  - **レイヤー毎にアクセス制御ルールが異なる**
  - **本当に漏れはないか？確認する術が無い**

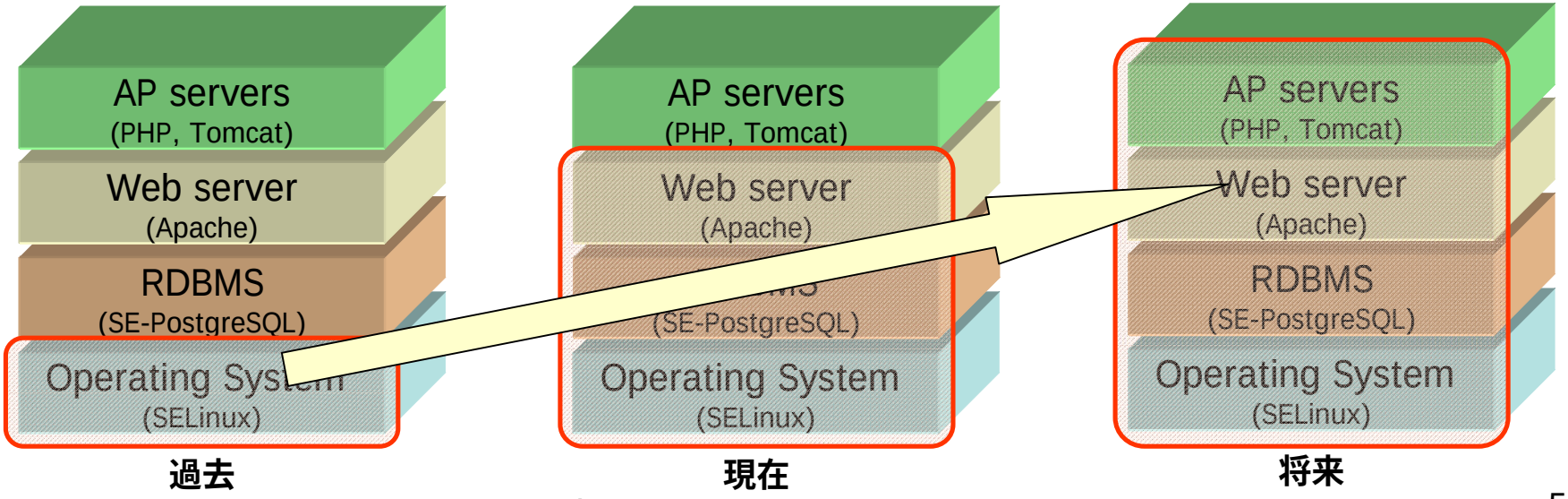
㈱ラック　侵入傾向分析レポート vol.12 より引用



LAPPスタック

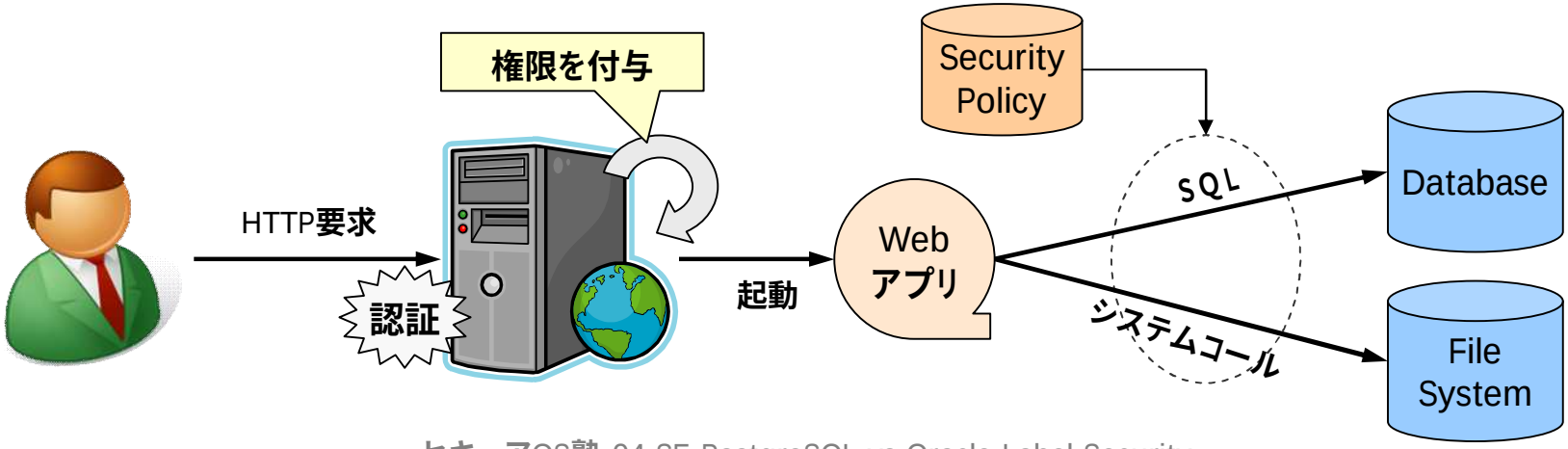
LAPP/SELinux (2/2)

- **一貫性**
  - **全てのレイヤーでSELinuxのセキュリティポリシーを適用**
  - **Webサーバによる認証、OS/DB(SE-PostgreSQL)でのアクセス制御**
- **網羅性**
  - **OS/DB/Web、全ての実行パスが例外なく経由するパスでチェック**
  - **特権ユーザに対しても、アクセス制御の例外とはならない**
- **セキュアなWebアプリケーション基盤の実現**



Apache/SELinux plusとの連携

- Apache/SELinux plusとは
  - Webアプリの実行前に、HTTP認証に基づいてSELinuxのセキュリティコンテキストを設定する、Apache 2.2.x用モジュール
  - 最初に一度だけ認証・権限付与すれば、ファイルへのアクセスでも、DBへのアクセスでも、共通のセキュリティポリシーで制御できる
    - 一貫性の担保
  - Webアプリケーションの実行前に権限を縮退させるので、仮にWebアプリがバグっていても何もできない
    - 網羅性の担保



SE-PostgreSQLの強み

- 強制アクセス制御
  - DBAに対しても例外なく適用
- システムワイドなアクセス制御一貫性
  - DBやFSといったデータ格納“方法”の違いを、
    アクセス制御の意思決定の違いに波及させない
  - OSやDB、各レイヤー毎に別個の認証を行わずに済む
- お値段 :-)
  - SELinux, SE-PostgreSQL, Apache/SELinux plusなどは全て公開
  - Fedora 11以降で利用可能、RHEL6向けにも提供予定

6. Oracle Label Securityの強み

事例: 某金融サービス企業

- 目標
  - 2TBを超える顧客情報の機密性を確保しつつ、10のデータベースを統合
  - コールセンター業務、会員登録処理、請求処理、EDIを統合
- ソリューション
  - 10のデータベースをOracle Label Securityを導入したシングルインスタンスに統合。Single Sign-onによるアクセス制御を実装
  - 統合により、セキュリティを損なうことなく、管理コストを20%削減することに成功
  - Oracle Label Securityの行レベルアクセス制御により、ユーザーの内部アクセスおよび約3000の提携ユーザーからの外部アクセスそれぞれに厳密なアクセス制御を施行

http://www.oracle.com/technology/deploy/security/database-security/pdf/jcb2apac0103ver2.pdf

事例: PCASSOプロジェクト

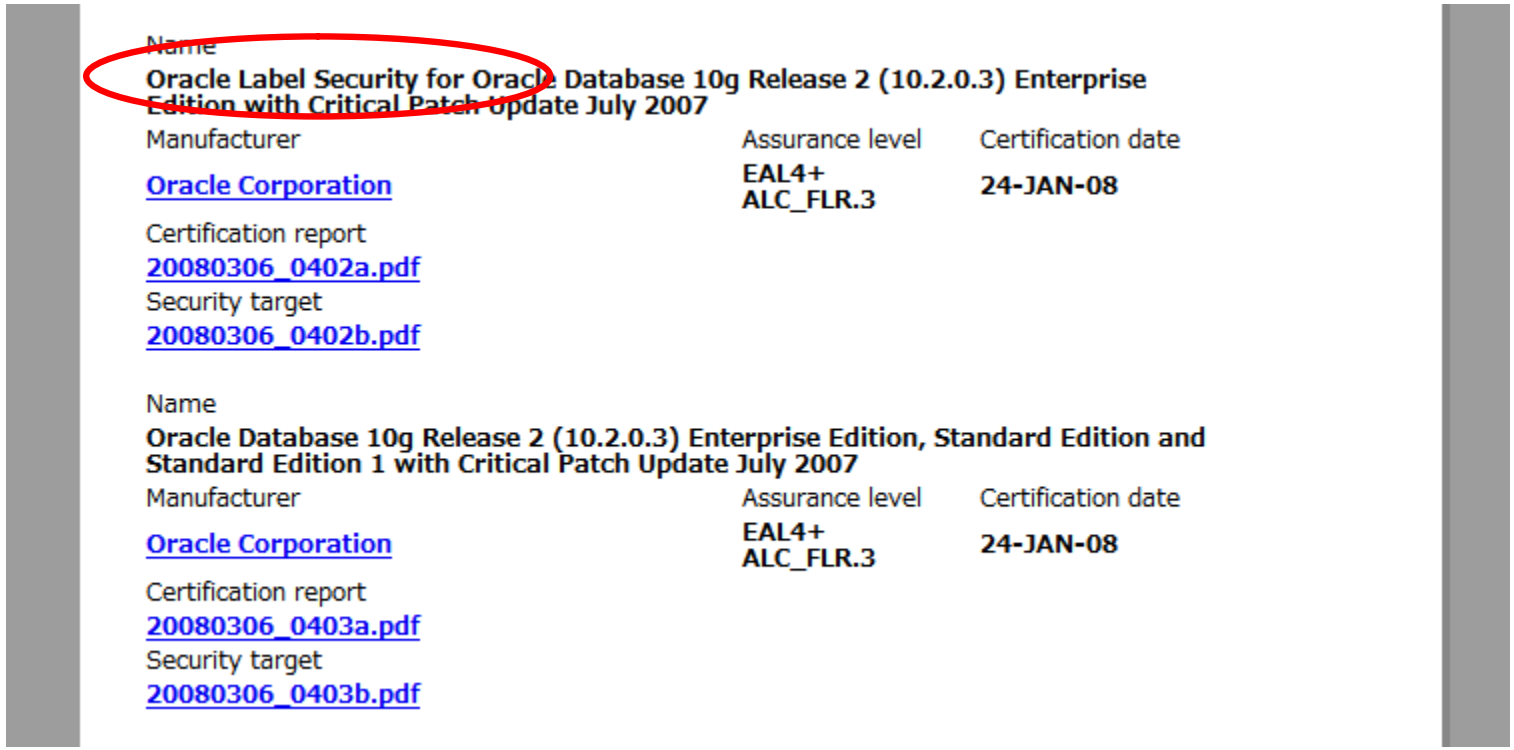
- Science Applications International Corporation(SAIC) ,
  the University of California, San Diego, School of Medicine(UCSD)
- http://medicine.ucsd.edu/pcasso/
- **目標**
  - **178,000人以上の患者情報を、必要な権限を持ったユーザーだけにインターネット経由でアクセスできるようにアクセス制御を行う**
- **ソリューション**
  - **Oracle Label SecurityのLBACを利用することで、機密情報である患者情報を、適切な権限を持つユーザー（例えば、主治医と患者本人のみ）にのみ参照できるように制御を実施**

PCASSOシステム

医療機関が管理する患者情報をインターネットに公開して、患者本人や担当医師だけが院外から安全に閲覧できるシステム

ISO/IEC15408

- ISO/IEC15408　DBMSPPでEAL4+を取得済み



Name
Oracle Label Security for Oracle Database 10g Release 2 (10.2.0.3) Enterprise Edition with Critical Patch Update July 2007

Manufacturer	Assurance level	Certification date
Oracle Corporation	EAL4+ ALC_FLR.3	24-JAN-08

Certification report
20080306_0402a.pdf
Security target
20080306_0402b.pdf

Name
Oracle Database 10g Release 2 (10.2.0.3) Enterprise Edition, Standard Edition and Standard Edition 1 with Critical Patch Update July 2007

Manufacturer	Assurance level	Certification date
Oracle Corporation	EAL4+ ALC_FLR.3	24-JAN-08

Certification report
20080306_0403a.pdf
Security target
20080306_0403b.pdf

http://www.commoncriteriaportal.org/products_DB.html#DB

Oracle Label Security 11g もEAL4+も申請中

http://www.oracle.com/technology/deploy/security/seceval/security-evaluations.html?_template=/ocom/print

他のOracle製品との統合

- Oracle Database Vault

  ラベルをDatabase Vault **の要素として使用可能。**
  ユーザラベルをDatabase Vault**のルールで使用することにより、**
  データベース、SQL**コマンド、テーブルへの強制アクセス制御を実現**

- Oracle Identity Management(Oracle Internet Directry)

  ラベルポリシー、ユーザラベル認可の集中管理

- Oracle Advanced Security

  表領域の暗号化、ネットワーク暗号化、バックアップデータの暗号化

Oracle Label Security って実は、、、

- Oracle Label SecurityってMACではないです。。

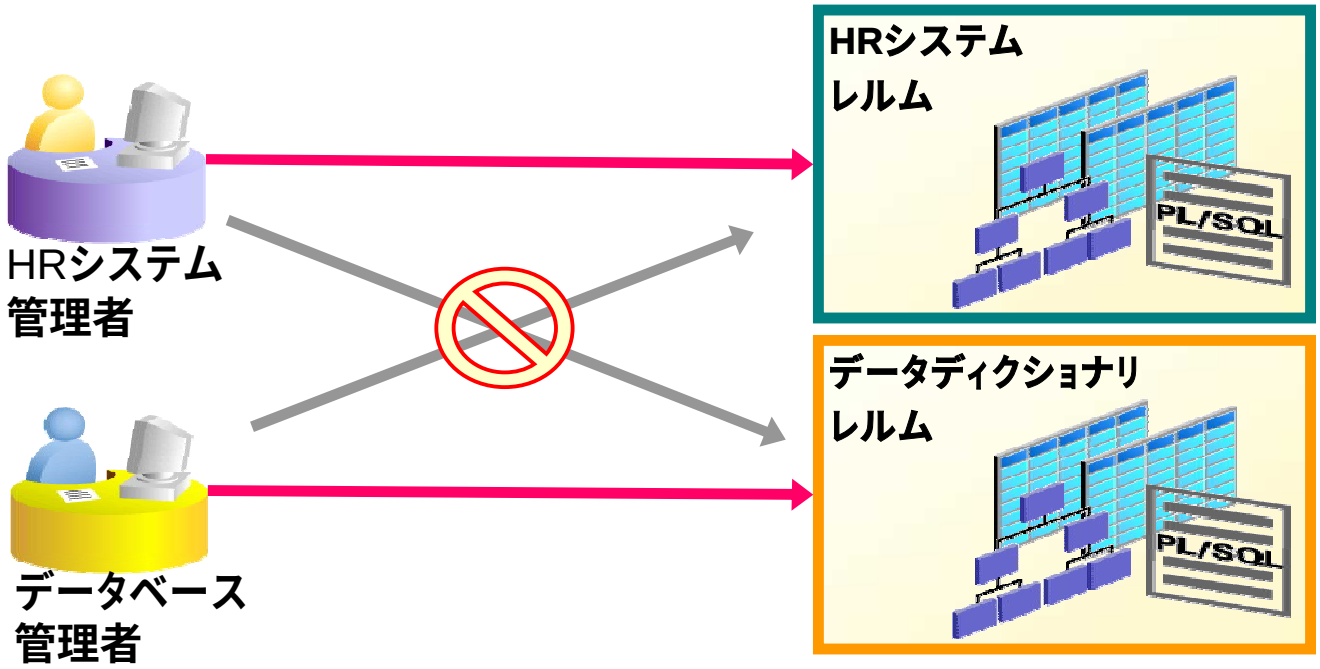
- DBAを制御できるわけではない。
  VPDはDBAだと回避できる。
  → Label SecurityはVPDの機能上実装している

- MACを実現しているのはDatabase Vault
  → Label SecurityとDatabase Vaultを組み合わせる

Oracle Database Vault

各レルムの認可を受けたユーザのみが、そのレルムで保護されたオブジェクトに対するシステム権限の行使、その他 DDL の発行 (実行権限は別途必要)を許可される

- 認可を受けたレルムに閉じた DBA の役割を持つ
- レルム上のオブジェクトに対する SELECT/DML/EXECUTE 権限を含む適切なロールを作成し、適切なユーザ(アプリケーション・ユーザ)に付与する
- 認可を受けていないレルムに対する、システム権限でのアクセスや DDL はレルム違反エラーとなる



他製品との連携

- Oracle Database Vaultの管理コンソール

7. まとめ

本日のサマリー (1/2)

- 問題意識
  - アプリケーション任せのセキュリティは根本的に危ない
    - アクセス制御の基準に間違いはない？ (一貫性)
    - 本当にアクセス制御の漏れはない？ (網羅性)
- アプローチ
  - バグ/脆弱性は根元(OS/DB)で抑えた方が間違いがない
    - OSの場合 ⇒ システムコール処理は誰も回避できない
    - DBの場合 ⇒ SQLクエリ処理は誰も回避できない
  - リファレンスモニタ
    - '80年代のセキュリティ研究に起源
    - OSへの適用例がSELinuxやTrusted Solaris
    - DBでも同じ構造が使える
      - SE-PostgreSQL や Oracle Label Security へ

本日のサマリー (2/2)

- SE-PostgreSQL
  - SELinuxを核にOS/DBのセキュリティポリシーを一元化
    - LAPP/SELinuxなどで、システム全体を保護する事が可能
      (System-wide consistency in access control)
  - OSSとして公開、Fedora/RHEL系でパッケージが標準化
  - 基本機能はOLSを凌駕するが、実績はまだ未知数
- Oracle Label Security
  - VPD上にマルチレベルセキュリティポリシーを実装
  - 商用RDBMSとしての豊富な実績
    - ISO/IEC15408認証も取得 (EAL4+; Jul 2007)
  - Oracle ミドルウェア群との連携
    - Oracle Database Vault, Oracle Identity Management, ...

8. 質疑応答

ありがとうございました

なお、本日の資料は、下記URLよりダウンロード可能です

http://www.secureos.jp/index.php?events/jsosjk04

日本セキュアOSユーザ会

Japan Secure Operating System Users Group since 2007