SE-PostgreSQLで構築する セキュアデータベース

日本セキュアOSユーザ会

海外 浩平 <kaigai@kaigai.gr.jp>

日本セキュアOSユーザ会

Japan Secure Operating System Users Group since 2007

はじめに　－思想的背景－ (1/2)

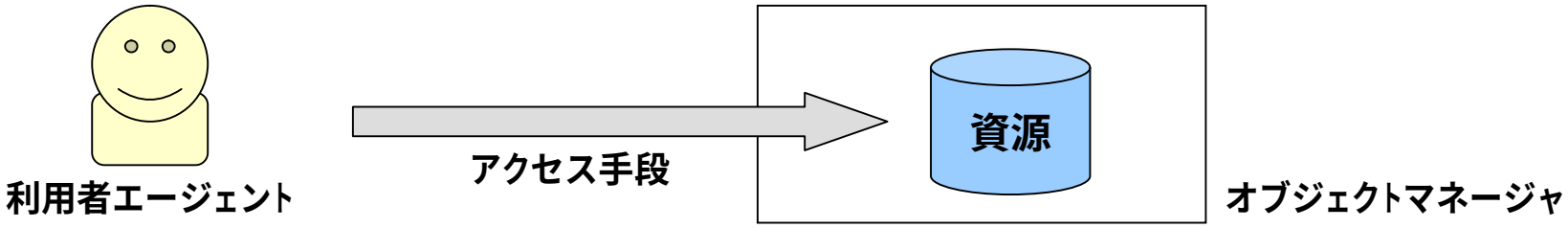
ビジネス手帳の価値：　¥1,280
個人情報の価値：　PRICELESS

- 情報システムにおける、セキュリティの役割とは？
  - **情報資産**を脅威から保全する事
- 情報資産とは
  - プライバシー、与信情報、パスワード、監査証跡、etc…
  - 特徴：　情報資産は、必ず何らかの媒体に格納される
    例：　ファイルシステム、データベース、紙、脳、石版、etc…
  - 求められるセキュリティの水準は、情報資産の”価値”しだい

はじめに　－思想的背景－ (2/2)

- 何が情報資産の価値を決めるのか？
  - ×　情報資産の格納方法
  - ○　情報資産の中身 (= コンテンツ)
- アクセス制御はどの様に機能するか？
  - ファイルシステム　　… rwxパーミッション、OSユーザ
  - データベース　　… GRANT/REVOKE構文、DBロール
  - 情報資産の”格納方法”に依存、一貫性を担保できない

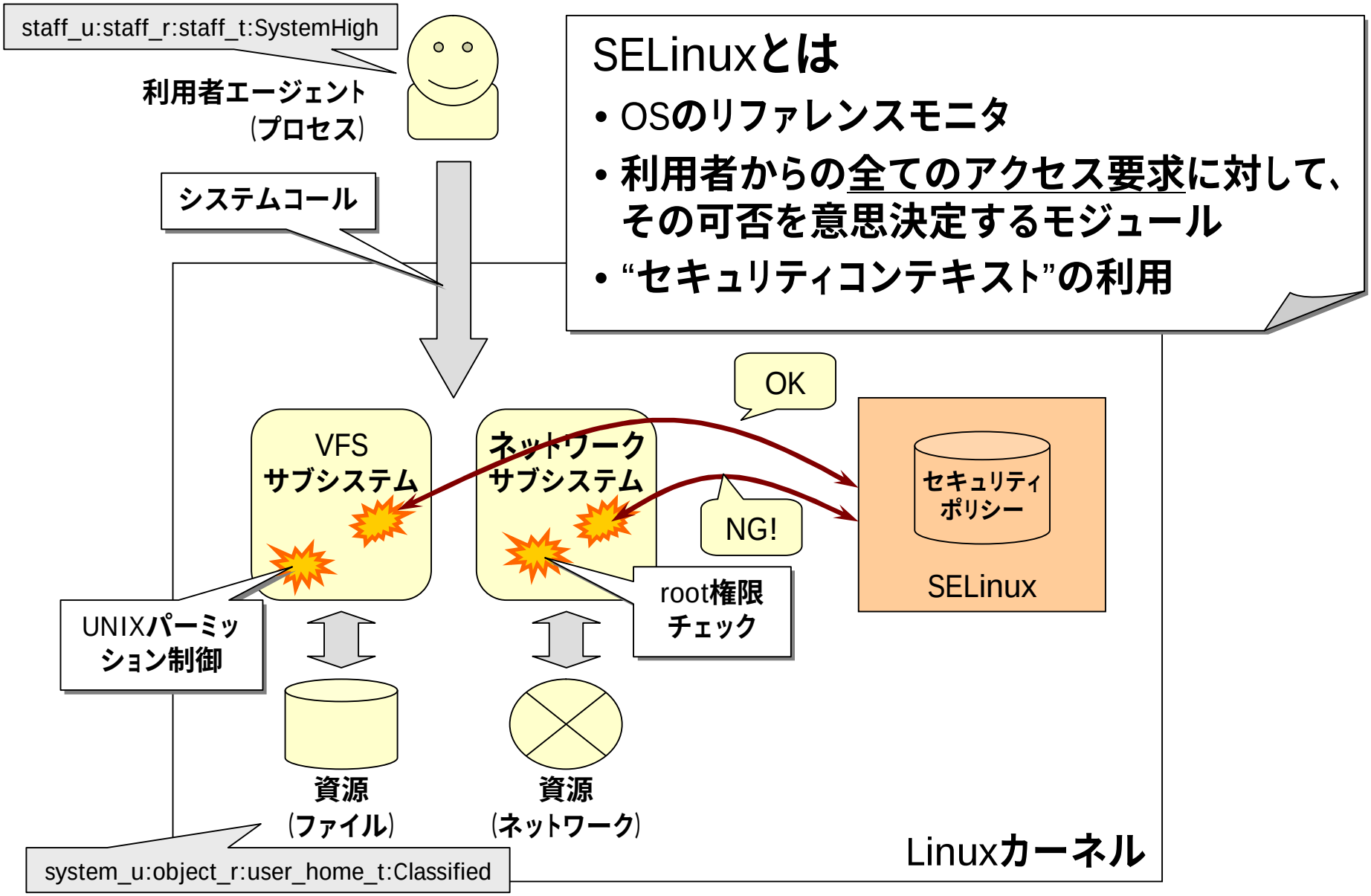
システムワイドなアクセス制御一貫性



- オブジェクトマネージャ
  - 資源へのアクセス手段／アクセス制御を提供
  - OS, DBMS, X-window
- OSとDBMSの類似性
  - OS: プロセスがシステムコールを介してファイルにアクセス
  - DBMS: クライアントがSQLを介してデータベースにアクセス
- アクセス制御の一貫性
  - 利用者エージェントは、共通の権限を持つ
  - オブジェクトマネージャは、共通のルールに基づいてアクセス制御を実施

SELinuxセキュリティポリシー

SELinux


staff_u:staff_r:staff_t:SystemHigh
利用者エージェント
(プロセス)
システムコール
SELinuxとは
• OSのリファレンスモニタ
• 利用者からの全てのアクセス要求に対して、その可否を意思決定するモジュール
• “セキュリティコンテキスト”の利用
OK
VFS
サブシステム
ネットワーク
サブシステム
NG!
セキュリティ
ポリシー
SELinux
UNIXパーミッション制御
root権限
チェック
資源
(ファイル)
資源
(ネットワーク)
system_u:object_r:user_home_t:Classified
Linuxカーネル

補足：SELinuxセキュリティポリシー

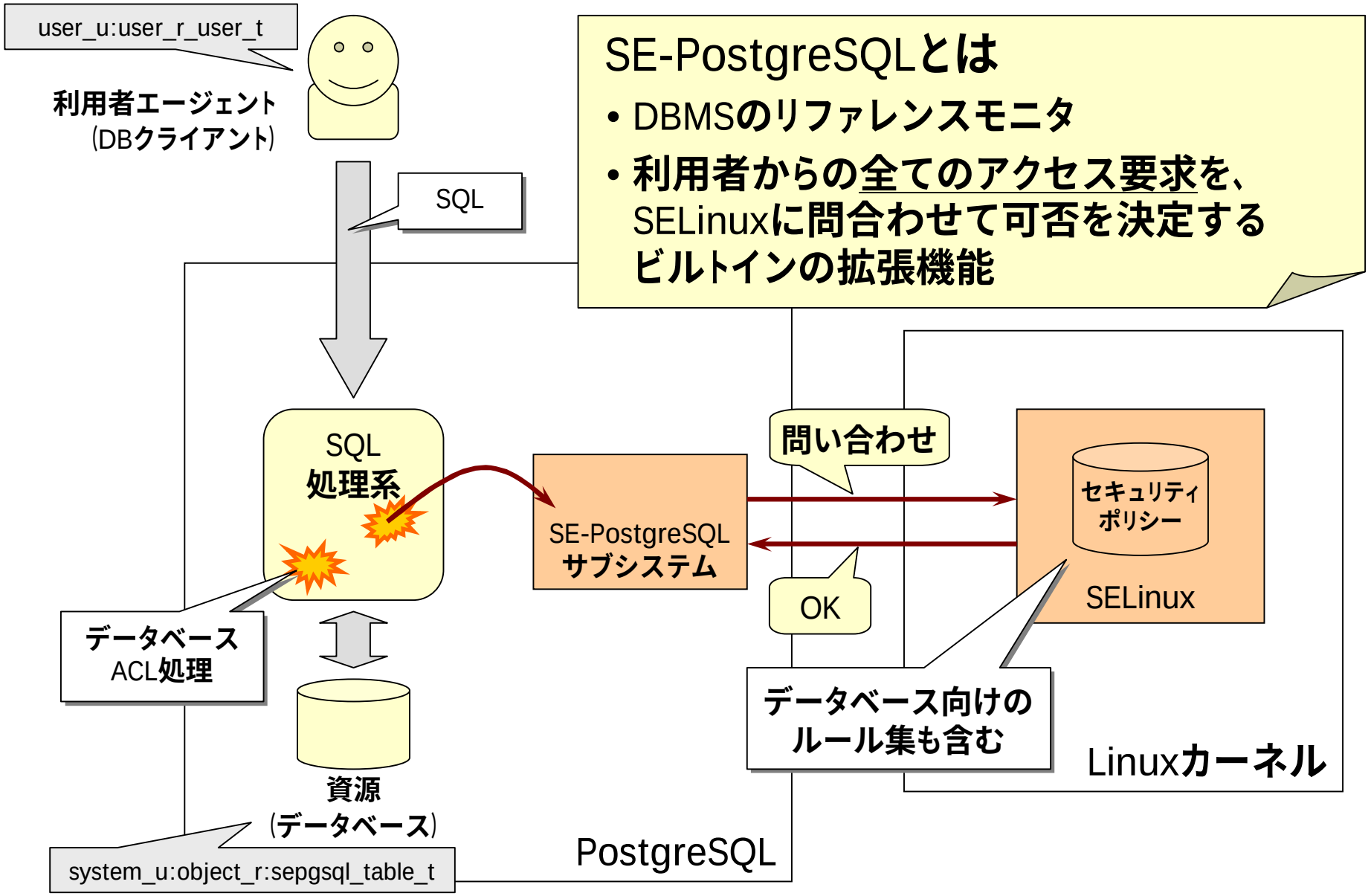
- セキュリティコンテキスト
  - SELinuxがアクセス制御に利用する識別子
    - プロセス、ファイル、ソケットetc...、全て共通の書式

```
unconfined_u:unconfined_r:unconfined_t:SystemHigh
system_u:object_r:user_home_t:Classified
```

Domain/Type　　Range

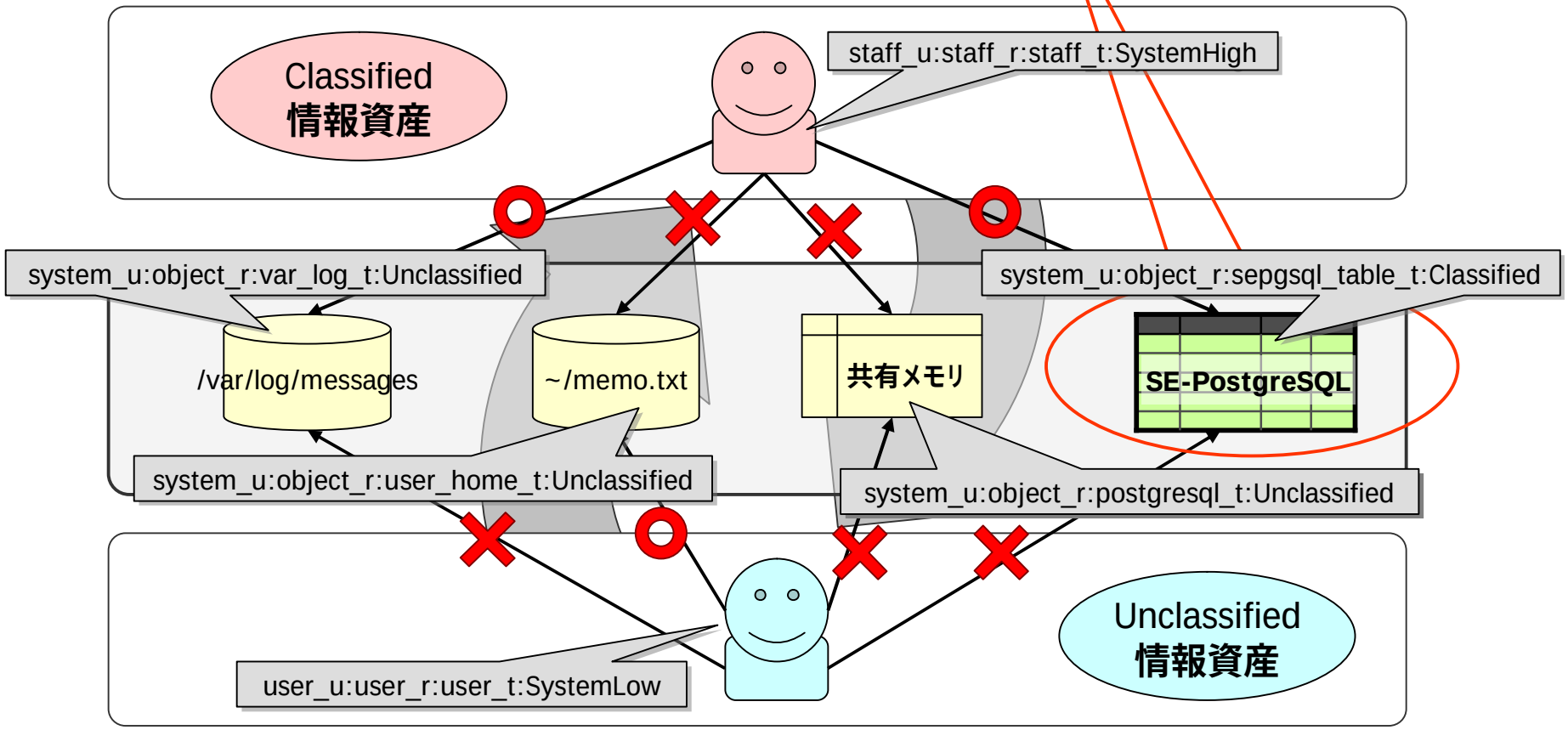
- セキュリティポリシー
  - 『誰が(subject)』『何に(object)』『何をできる(action)』ルール集
  - ホワイトリスト方式
  - TE:　Domain/Type間にルールを記述
  - MCS:　Rangeの上下/包含関係を制約するルールを記述

SE-PostgreSQL



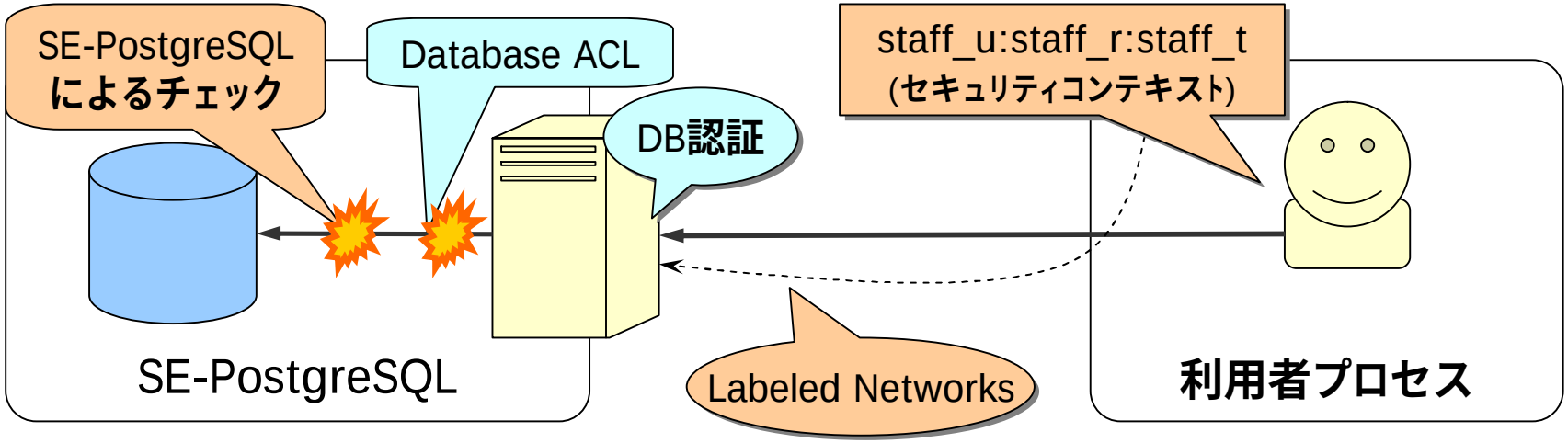
セキュリティポリシーの一元化

- 利用者エージェントの権限は、OS/DBの双方に対して共通
- OS/DBの双方が、集中管理されたセキュリティポリシーを利用する
- 情報フローの経路に拠らず一貫したアクセス制御



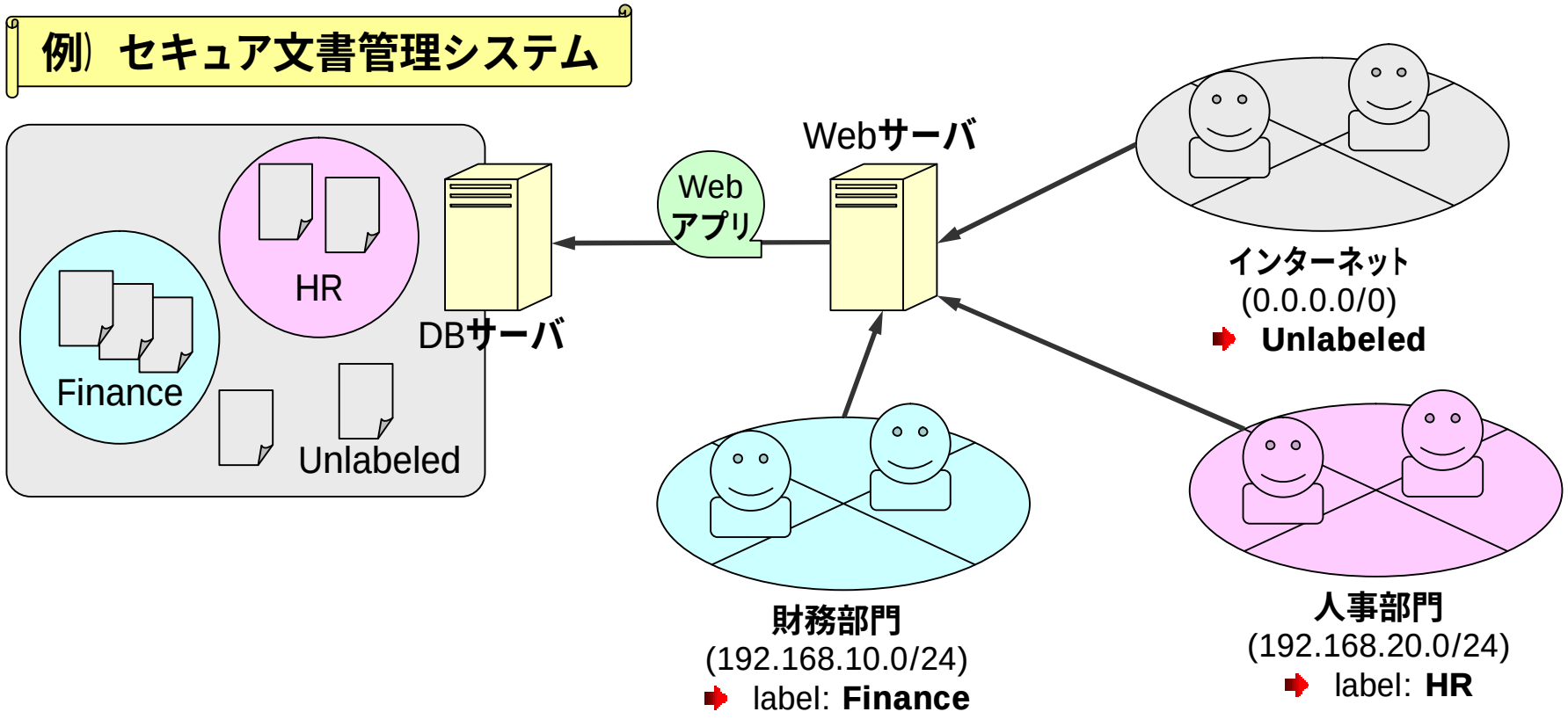
利用者の権限

- Labeled Networks
  - **接続元プロセスのセキュリティコンテキストを取得する機能**
  - **DB認証とは無関係に、クライアントの権限を決定する**
- SE-PostgreSQL**では**
  - **接続元プロセスの権限に基づいてアクセス制御**
    (DBユーザとは無関係！)
  - **利用例**：**Webアプリには機密情報を参照させない**



目標とするシステムイメージ

- 制約された権限の下で動作する Web アプリケーション
- ファイル/データベースへのアクセスを全てチェック
- Webアプリがバグってても、不正なアクセスは許さない



例：行レベルのアクセス制御

```
postgres=# SELECT security_label, * FROM drink;
                security_label                 | 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_label’ **システム列**
  - **行単位のセキュリティコンテキストを参照/更新できる**
- **行レベルアクセス制御**
  - **権限のない行は、**SELECT**や**UPDATE**の対象からフィルタリングされる**
  - **強制的に**WHERE**句/**JOIN**～**ON**句に条件を付加するイメージ**

例：行レベルのアクセス制御

```
postgres=# SELECT security_label, * FROM drink;
                security_label                | id |   name   | price
----------------------------------------------+----+----------+-------
 system_u:object_r:sepgsql_ro_t               |  1 | water    |   100
 system_u:object_r:sepgsql_ro_t               |  2 | tea      |   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)
```

‘Unclassified’ **な利用者からは、行がフィルタリングされて見える**

- ‘security_label’ **システム列**
  - **行単位のセキュリティコンテキストを参照/更新できる**
- **行レベルアクセス制御**
  - **権限のない行は、**SELECT**や**UPDATE**の対象からフィルタリングされる**
  - **強制的に**WHERE**句/**JOIN**～**ON**句に条件を付加するイメージ**

例：行レベルのアクセス制御

```
postgres=# SELECT security_label, * FROM drink;
                security_label                 | 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             |  5 | tea     |   140
 system_u:object_r:sepgsql_table_t             |  6 | milk    |   200
(6 rows)
```

‘Read Only’ **なので、**SELECT**は可能だが、**
UPDATE**や**DELETE**の対象からは除外される**

- ‘security_label’ **システム列**
  - **行単位のセキュリティコンテキストを参照/更新できる**
- **行レベルアクセス制御**
  - **権限のない行は、**SELECT**や**UPDATE**の対象からフィルタリングされる**
  - **強制的に**WHERE**句/**JOIN**～**ON**句に条件を付加するイメージ**

例: 列レベルのアクセス制御 (1/2)

```
postgres=# select * from customer;
ERROR:  SELinux: denied { select }
    scontext=unconfined_u:unconfined_r:sepgsql_test_t
    tcontext=system_u:object_r:sepgsql_secret_table_t
    tclass=db_column name=customer.credit
postgres=# select cid, cname from customer;
 cid | cname
-----+--------
   1 | kaigai
   2 | yamada
   3 | kimura
(3 rows)
```

- テーブル、カラム、SQL関数etc...もセキュリティコンテキストを持つ
- 権限のないDBオブジェクトを利用した場合、事前にエラーを返す
  - SELECT権限のないカラムの参照、DELETE権限のないテーブルの削除
  - 単純に結果をフィルタリングする行レベルの制御とは異なる
  - ✓ “行レベル”では、クエリの実行前にアクセス対象を確定できない :-)

例：列レベルのアクセス制御 (2/2)

```
postgres=# select * from customer;
ERROR:  SELinux: denied { select }
    scontext=unconfined_u:unconfined_r:sepgsql_test_t
    tcontext=system_u:object_r:sepgsql_secret_table_t
    tclass=db_column name=customer.credit
postgres=# select cid, cname, show_credit(cid) from customer;
 cid | cname  |     show_credit
-----+--------+---------------------
   1 | kaigai | 1111-xxxx-xxxx-xxxx
   2 | yamada | 5555-xxxx-xxxx-xxxx
   3 | kimura | 9999-xxxx-xxxx-xxxx
(3 rows)
```

- Trusted Procedure
  - SQL関数の実行中だけ、クライアントの権限を変更する
  - この例では、”credit”列を伏字にして返却
- 応用例
  - 決済システムはクレジットカード番号を参照可能
  - Webアプリケーションは、伏字でしか参照できない

ディストリビューションの対応状況

- Fedora Project
  - sepostgresqlパッケージ採用（Fedora 8～）
  - SELinux標準セキュリティポリシーでのサポート
- Red Hat EL/Cent OSなど
  - RHEL5では、カーネル/ポリシー共に古すぎるため不可
  - RHEL6向けに、EPELリポジトリ経由での提供を検討
    - ※ EPEL = Extra Packages for Enterprise Linux
- PostgreSQL本家
  - v8.4 での採用に向けて議論を進めている途中
    http://wiki.postgresql.org/wiki/CommitFest:2008-11
  - 但し、一部機能に関しては v8.5 へ延期 (泣
  - なので、しばらくは Fedora 向けパッケージを使って下さい (汗

インストール

```
[root@masu ~]# yum install -y sepostgresql
        :
================================================================================
 Package         Arch    Version                Repository         Size
================================================================================
Installing:
 sepostgresql    i386    8.3.6-2.1518.fc11      rawhide           2.1 M

Transaction Summary
================================================================================
        :
Installed:
  sepostgresql.i386 0:8.3.6-2.1518.fc11

Complete!
[root@masu ~]# /etc/init.d/sepostgresql initdb
Initializing database:                                     [  OK  ]
[root@masu ~]# /etc/init.d/sepostgresql start
Starting sepostgresql service:                             [  OK  ]
```

- Fedoraでは非常に簡単
- むしろ、現状ではFedora以外での利用をお勧めできない orz
- カーネル/セキュリティポリシー/周辺コマンドetc...の対応状況

DB設計の方針 (1/2)

- **標準のセキュリティポリシーを利用する**
  - **セキュリティ設計は基本的に難しい**
    - **専門家の作成したセキュリティポリシーの利用**
      = ベストプラクティス
  - TE/MCS**の中から、必要なものをピックアップ**
- **定義済みタイプの一例**
  - sepgsql_table_t … **読み書き可能**
  - sepgsql_secret_table_t … **一般ユーザのアクセス不可**
  - sepgsql_fixed_table_t … SELECT/INSERT**のみ可**
  - **staff**_sepgsql_proc_exec_t
    … staff_t **ドメインのみ実行可能な**SQL**関数**

DB設計の方針 (2/2)

- デフォルトのセキュリティコンテキスト
  - テーブル: sepgsql_table_t (無印PostgreSQLと同じ)
  - カラム/タプル: テーブルのタイプを継承
  - カテゴリ: 利用者のカテゴリを継承
- 例
  - 利用者 `system_u:system_r:httpd_t:s0:c3`
  - テーブル `staff_u:object_r:sepgsql_fixed_table_t:s0`
  - INSERTした列 `staff_u:object_r:sepgsql_fixed_table_s:s0:c3`
- 拡張SQL構文

```
CREATE TABLE audit (
    id    serial primary key,
    data  text
) security_label = 'system_u:object_r:sepgsql_fixed_table_t:s0'
```

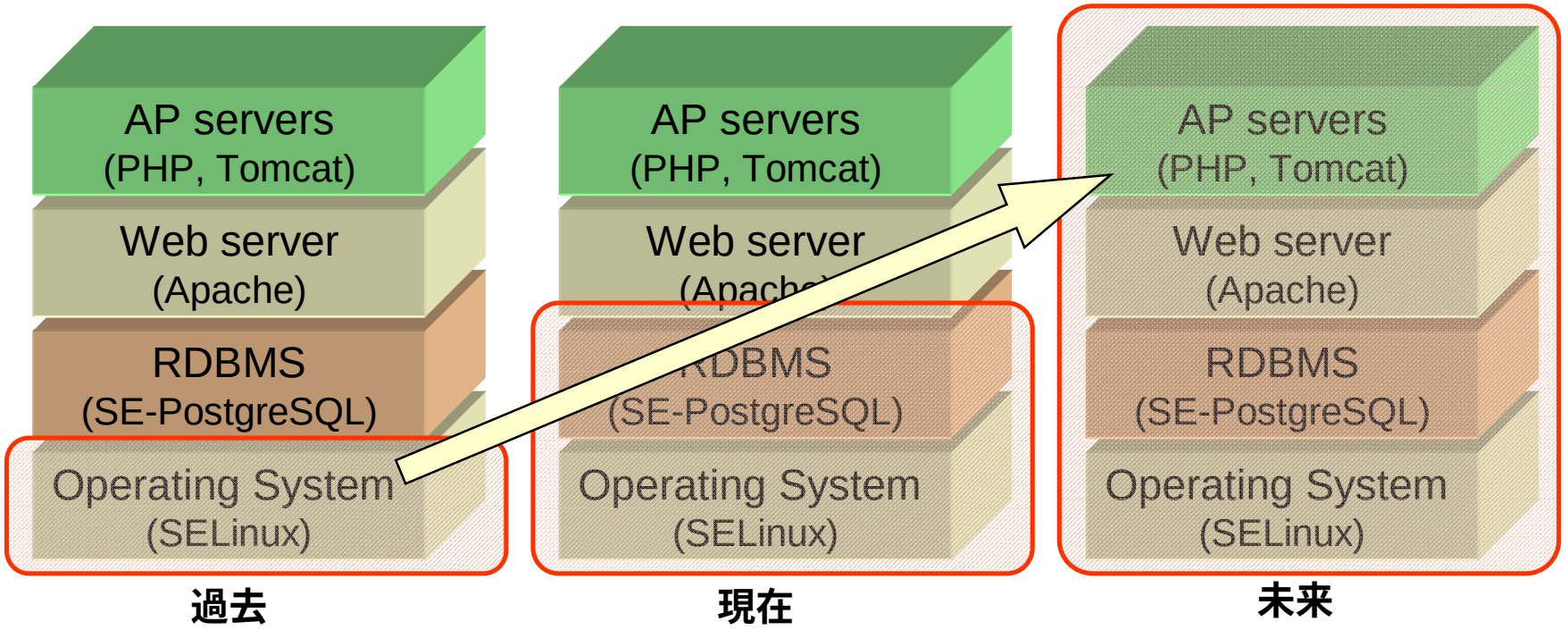
デモンストレーション

日本セキュアOSユーザ会

Japan Secure Operating System Users Group since 2007

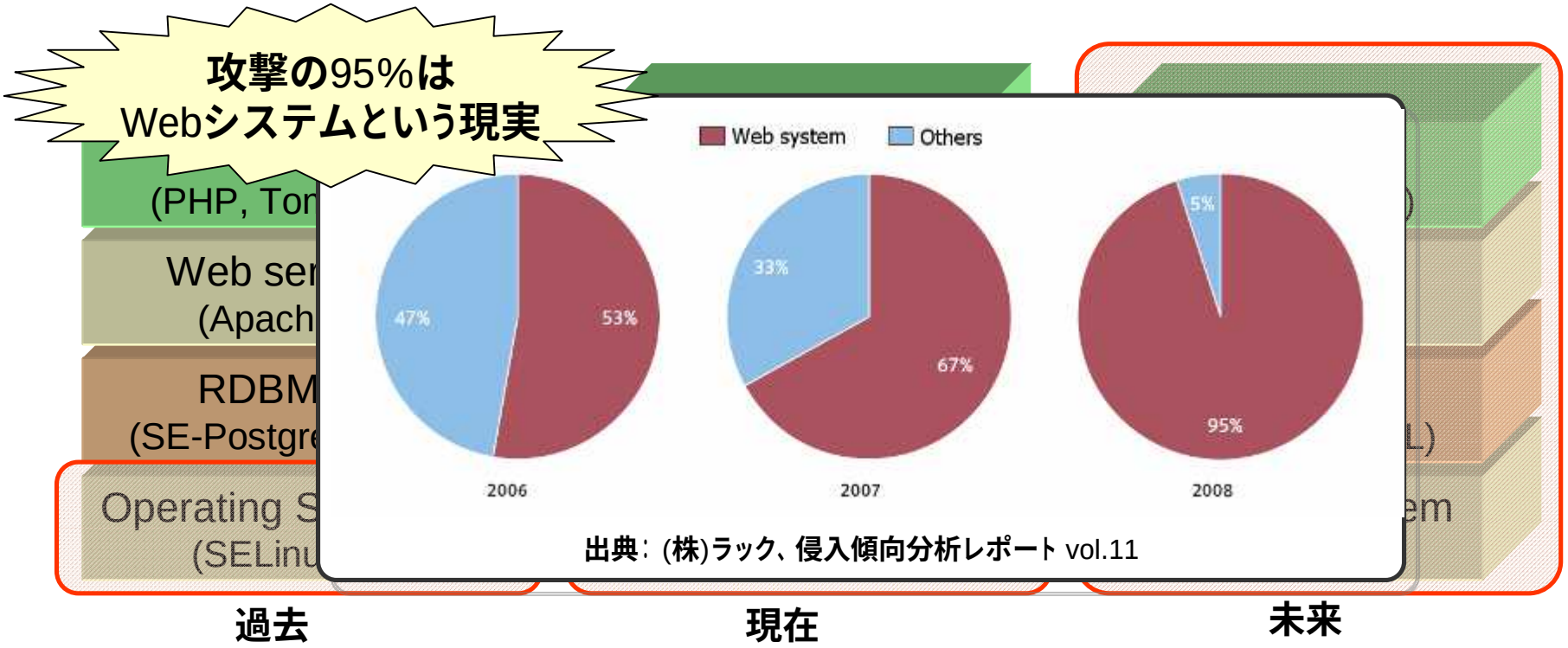
今後の方向性

- LAPPスタック全体をSELinuxで保護
  - ターゲットは web アプリケーション
  - SaaS化/Cloud-Computingの流れ
  - 抜本的なセキュリティの改善は必須



今後の方向性

- LAPPスタック全体をSELinuxで保護
  - ターゲットは web アプリケーション
  - SaaS化/Cloud-Computingの流れ
  - 抜本的なセキュリティの改善は必須



セキュアOSユーザ会

- about us
  - **セキュアOS技術を中心に、技術者、利用者etc...の情報交換・交流・議論を目的としたコミュニティ**
  - Web: http://www.secureos.jp/
  - ML: users-ml@secureos.jp
- **セキュアOS塾**
  - **3～4ヶ月に一回くらいの勉強会**
  - **平日夜/都内/懇親会**
  - **過去のテーマ**
    - SELinux**入門**
    - LAPP/SELinux
    - TOMOYO Linux
    - POSIX Capability

'08/10/29 **セキュア**OS**塾**-01
(於・港区勤労者福祉会館)

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

業務連絡

- 展示：　セキュアOSユーザ会
  - 5階A展示教室
  - TOMOYO Linux, SE-PostgreSQL, LAPP/SELinux

- セミナー情報：16:15～／8階A教室
  セキュアOS上でのアプリケーション動作検証
  ～OSS ECM Alfrescoの例～
  - 講師：才所 秀明（Linuxコンソーシアム）
  - CMSアプリケーション等を導入する場合、現状セキュアOS機能をOFFに されてしまう場合が殆どです。本セッションでは、OSS ECM (Enterprise Content Management) AlfrescoのセキュアOS上での動作検証結果や その効果を紹介します。

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