Empowered by Innovation NEC

LAPP/SELinux

SE-PostgreSQLを用いたセキュアWebアプリケーション基盤

KaiGai Kohei <kaigai@ak.jp.nec.com>
NEC OSS推進センター

自己紹介

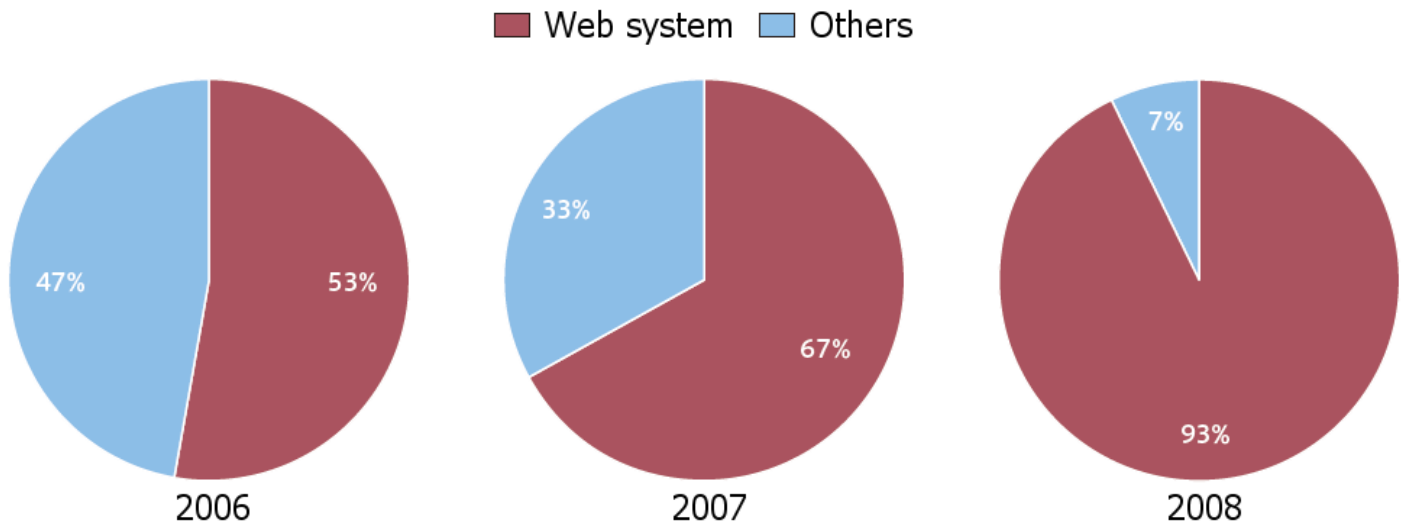
- **海外 浩平**
- NEC OSS推進センター勤務
- SELinuxなどLinuxカーネル開発に貢献
  - SMPスケーラビリティの改善
  - 一連の組込みプラットフォームへの移植
  - SE-PostgreSQLの開発
  - スレッド単位のセキュリティコンテキスト機能のサポート

- **最近の関心は、Webシステムのセキュリティ**

Empowered by Innovation NEC

1. Background
2. SE-PostgreSQL
3. Apache/SELinux Plus
4. LAPP/SELinux

Webシステムに迫る脅威



Targets of significant security incidents

(出展: JSOC侵入傾向分暦レポート, vol.12, LAC)

- Webシステムに対する攻撃が急増
  - 電子商取引の拡大と軌を一にするものと推定されている
- 既存のセキュリティ対策では十分に対応可能か？

LAPP - 典型的なWebアプリケーションスタックとして

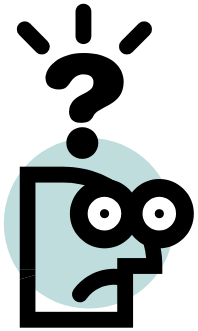
LAPP

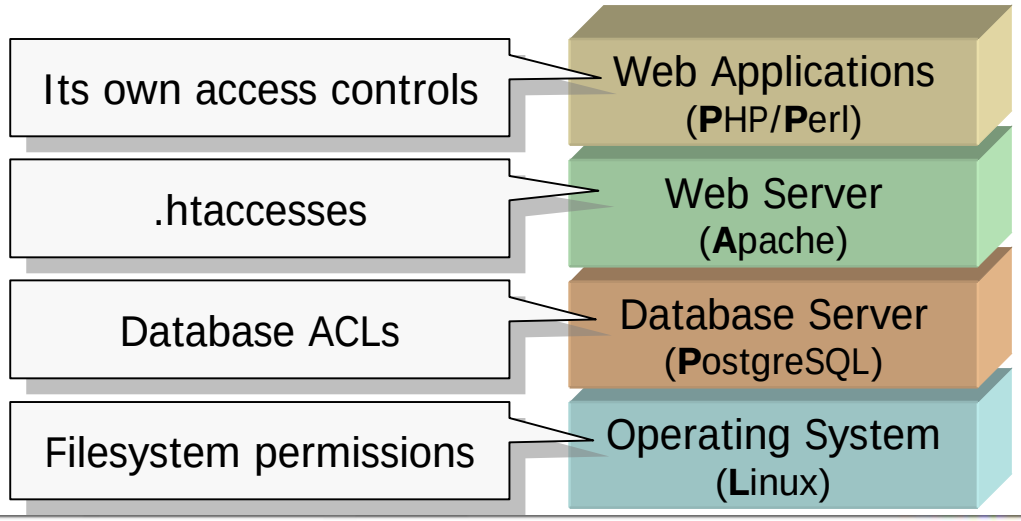
- 全OSSのWebアプリケーションスタック。LAMPとしても知られる。
  - Linux, Apache, PostgreSQL and PHP/Perl

セキュリティ上の不満

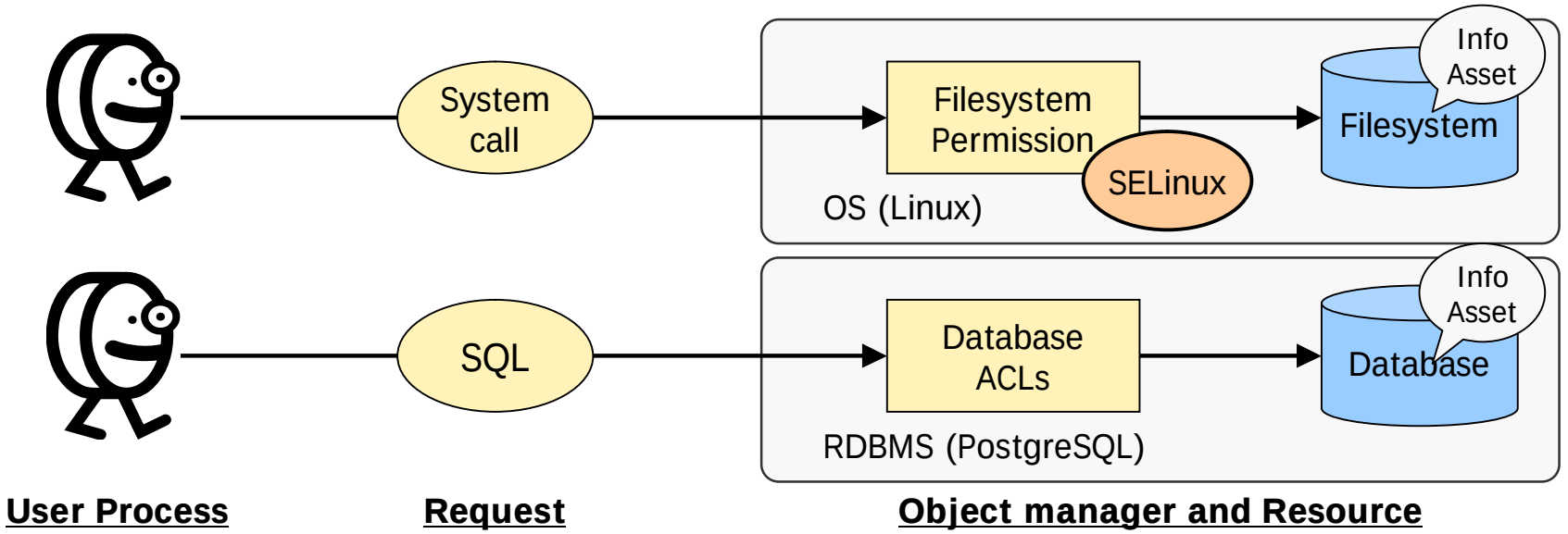
- 各レイヤーが独自のアクセス制御を実施する
  - ➡ 一貫性の欠如
- 個々のWeb利用者に応じた権限が付与されない
  （つまり、全てのセキュリティはWebアプリの品質に依存！）
  - ➡ 網羅性の欠如

キーワード … Analogy



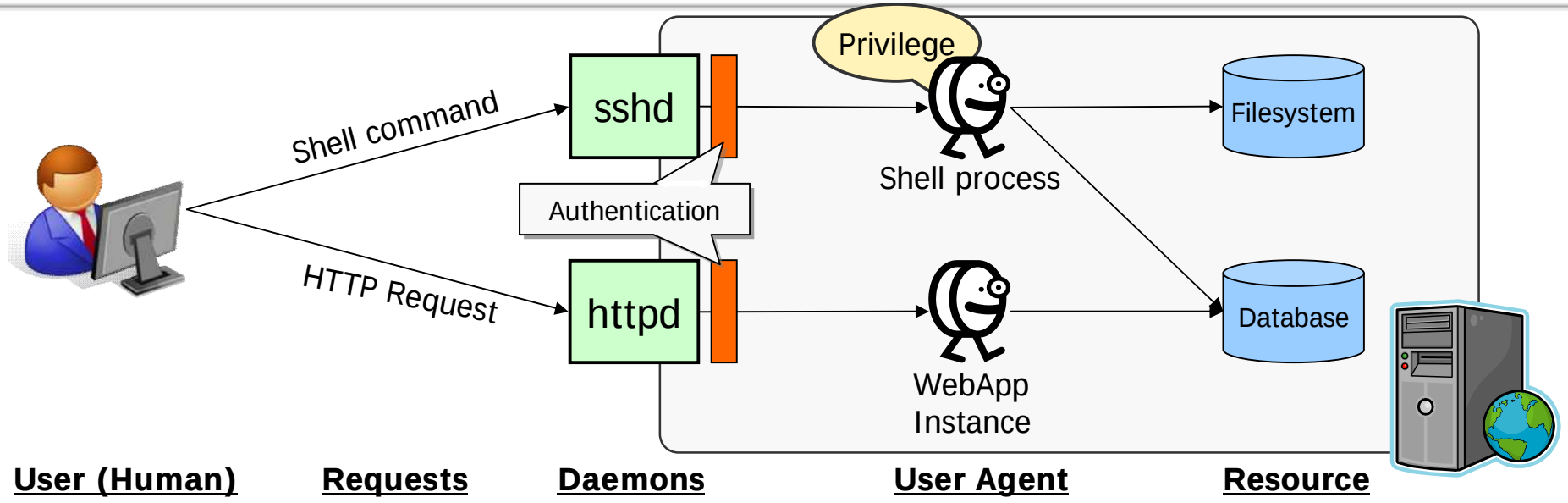


OSとDBの間のアナロジー



- ユーザプロセスと情報資産の関係に注目
- 情報資産を保存／参照する手段の違い
  - システムコール：ファイルシステム ⇔ ＳＱＬ：データベース
- アクセス制御とは、特定の利用者/資源のペアに対して何の操作が許可/拒否されるのかを決定する事
  - ＯＳとＤＢで共通のセキュリティモデルが利用できない理由は無い
    アクセス制御の "一貫性" を保証する

SSHとWebの間のアナロジー



- ユーザ = 人間; "エージェント" が人間 (利用者) の代わりに働く
- エージェントは、利用者を正しく反映した権限を持つ
  - 認証によって、利用者を識別し、権限を設定する
  - Httpdは、個々の権限を付与することなくWebアプリを起動する
  - OS/DBは、誰がエージェントの背後にいるのかを判断できない
- 利用者に応じた権限を、Webアプリにも設定する必要がある

アナロジーから学べること

SE-PostgreSQL

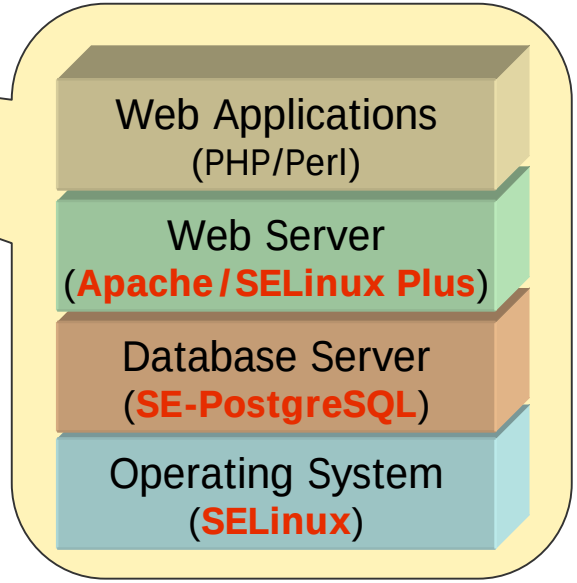
- SELinuxを利用して、SQLに対して追加的なアクセス制御
- アクセス制御における一貫性の担保

Apache/SELinux Plus

- SELinuxを利用して、Webアプリの権限をコントロール
- Webアプリにおける、アクセス制御の網羅性を担保

LAPP/SELinux

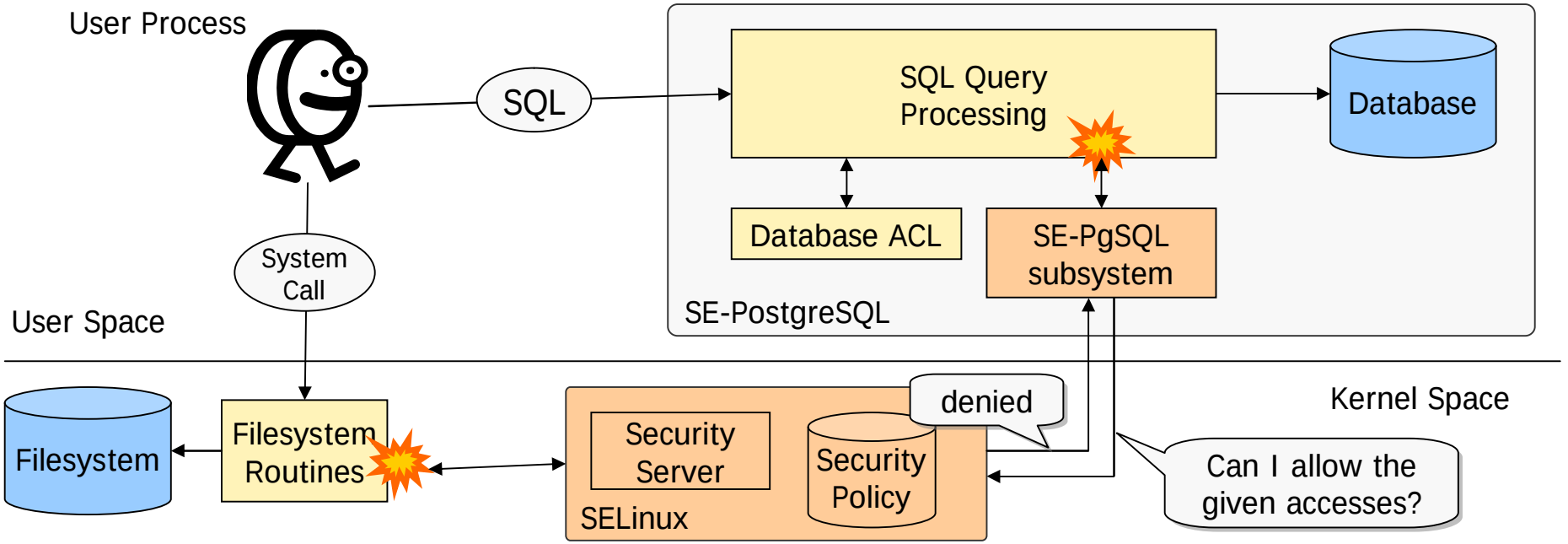
- LAPPスタックにおけるSELinuxの応用
  - SELinux + SE-PostgreSQL + Apache/SELinux Plus



Empowered by Innovation NEC

1. Background
2. SE-PostgreSQL
3. Apache/SELinux Plus
4. LAPP/SELinux

SE-PostgreSQLのアーキテクチャ

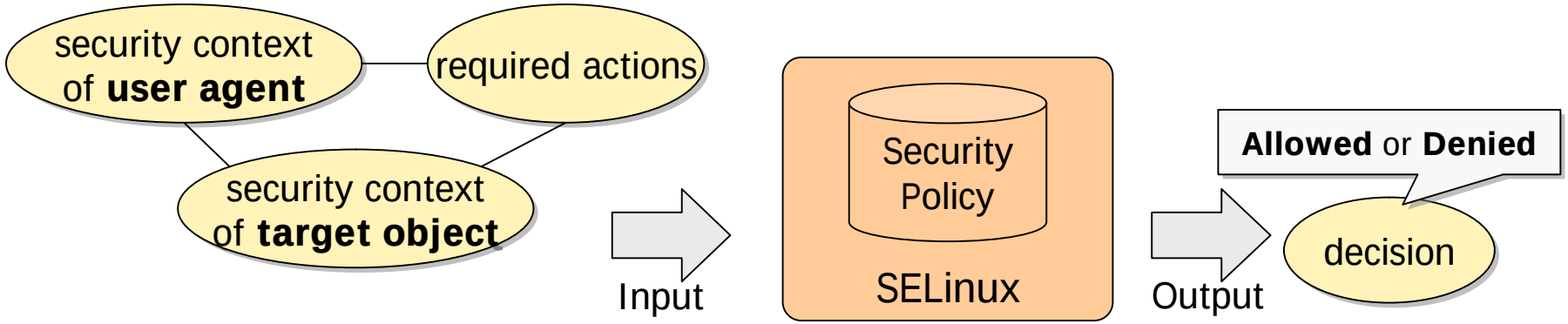


- SELinuxはシステムコール処理をフック
- SE-PostgreSQLもSQLクエリ処理をフック

- SELinuxはセキュリティポリシーに基づいてアクセス制御の意思決定を行なう
- SELinuxの意思決定に基づいて、
  SE-PostgreSQLはSQLクエリ処理の実行の可否を制御する

➡ 一元管理されたセキュリティポリシーが、OSとDBを共に制御する事を意味する

SELinuxにおける意思決定



SELinux は関数のように振舞う

- SELinuxは入力に対して "許可" または "拒否" の二値を返す
- カーネルの実装は、SELinuxを呼び出し、その意思決定に従っている
- セキュリティコンテキストのペアを入力できる限り、
  アプリケーションもまたこの仕組みを利用する事ができる

セキュリティコンテキスト

- SELinux固有の識別子で、プロセスやその他のオブジェクトに付与
- ファイル、ネットワーク、etc... ⇒ カーネルが管理
- DBオブジェクト、etc... ⇒ アプリケーションが管理しなければならない

security_context システム列

一般のテーブルのセキュリティコンテキスト

```
postgres=# SELECT security_context, * FROM drink;
            security_context              | id | name  | price
------------------------------------------+----+-------+-------
 system_u:object_r:sepgsql_table_t:s0     |  3 | juice |   130
 system_u:object_r:sepgsql_table_t:s0     |  4 | cofee |   180
 system_u:object_r:sepgsql_table_t:s0:c0  |  5 | beer  |   240
 system_u:object_r:sepgsql_table_t:s0:c0  |  6 | sake  |   320
 system_u:object_r:sepgsql_table_t:s0:c1  |  7 | wine  |   380
 system_u:object_r:sepgsql_table_t:s0:c1  |  8 | tea   |   140
(6 rows)
```

システムカタログのセキュリティコンテキスト

System catalog

```
postgres=# SELECT security_context, attname, attnum FROM pg_attribute
        WHERE attrelid = 'drink'::regclass AND attnum > 0;
            security_context              | attname | attnum
------------------------------------------+---------+--------
 system_u:object_r:sepgsql_table_t:s0     | id      |      1
 system_u:object_r:sepgsql_table_t:s0     | name    |      2
 system_u:object_r:sepgsql_ro_table_t:s0  | price   |      3
(3 rows)
```

利用者の権限

- SE-PostgreSQLはピアプロセスの権限を適用する
  - データベース認証には依存しない
  - SELinuxは通信相手プロセスのセキュリティコンテキストを取得するAPIを提供
    - `getpeercon(3)` を参照のこと
- Labeled IPsec
  - リモートプロセスに対してセキュリティコンテキストを通知する
  - IPsecの拡張として実装、kernel-2.6.18以降で対応

```
[ymj@saba ~]$ id -Z
uid=1002(ymj) gid=100(users) groups=100(users) ¥
        context=staff_u:staff_r:staff_t:s0-s0:c0.c15

[ymj@saba ~]$ psql -q postgres -U dbguest
postgres=> SELECT sepgsql_getcon(), current_user;
            sepgsql_getcon             | current_user
---------------------------------------+--------------
 staff_u:staff_r:staff_t:s0-s0:c0.c15 | dbguest
(1 row)
```

SE-PostgreSQLの利用例 (1/2)

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:Unclassified |  1 | water   |   100
 system_u:object_r:sepgsql_ro_table_t:Unclassified |  2 | coke    |   120
 system_u:object_r:sepgsql_table_t:Unclassified    |  3 | juice   |   130
 system_u:object_r:sepgsql_table_t:Unclassified    |  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
 staff_u:object_r:sepgsql_table_t:Unclassified     |  7 | soda    |   150
```

- SELECTした時
  - Unclassified の利用者からは、Classified行はフィルタリング
- UPDATE/DELETEした時
  - 読込み専用 (sepgsql_ro_table_t) 行の更新/削除をスキップ
  - ただし、Classified利用者は、Classified行でも更新/削除できる
- 新しい行をINSERTした時
  - 新しい行に対して、デフォルトのセキュリティコンテキストが付与される

SE-PostgreSQLの利用例 (2/2)

表/列レベルアクセス制御

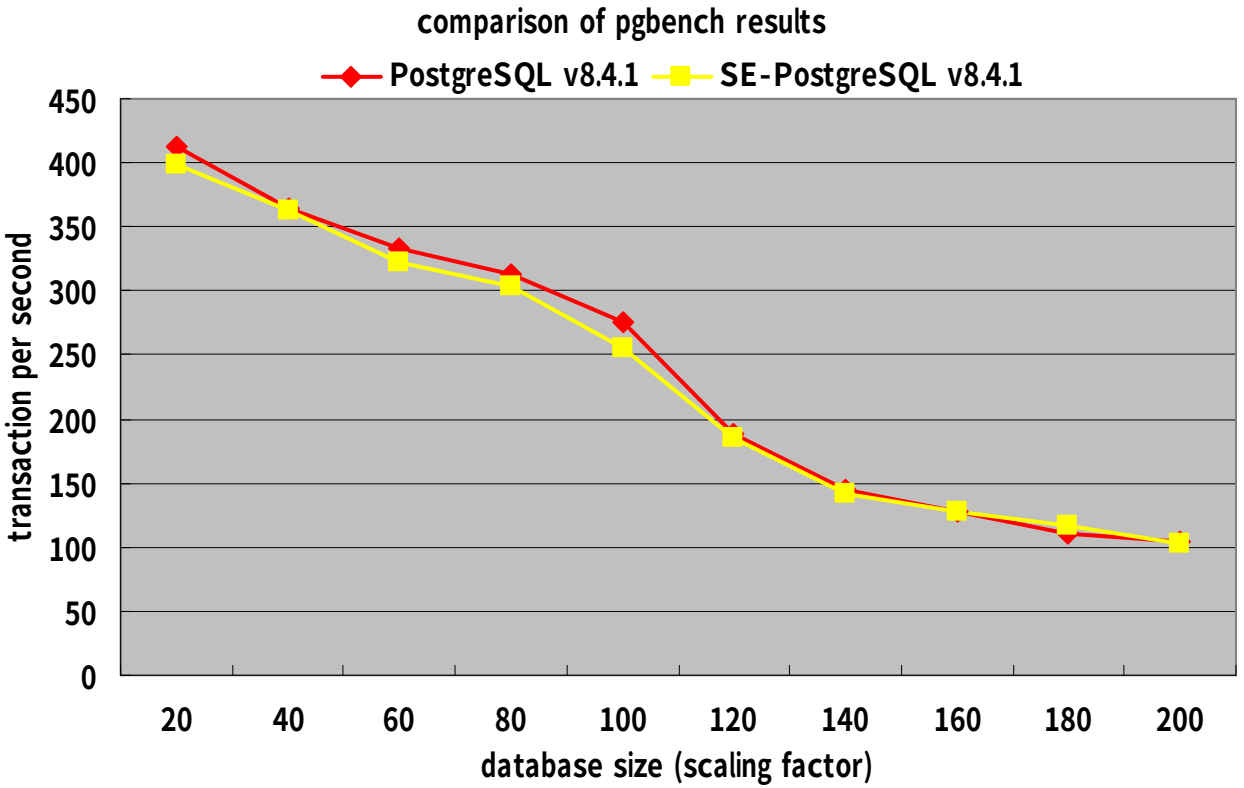
```
postgres=# CREATE TABLE customer (
      cid       integer primary key,
      cname     varchar(32),
      ccredit   varchar(32)
          SECURITY_CONTEXT = 'system_u:object_r:sepgsql_secret_table_t:s0'
      );
CREATE TABLE
```

セキュリティコンテキストを、
特定の表や列に設定する事が可能

```
postgres=> SELECT * FROM customer;
LOG:  SELinux: denied { select } ¥
    scontext=staff_u:staff_r:staff_t:Unclassified ¥
    tcontext=system_u:object_r:sepgsql_secret_table_t:Unclassified ¥
    tclass=db_column name=customer.ccredit
ERROR:  SELinux: security policy violation
postgres=> SELECT cid, cname FROM customer;
 cid | cname
-----+-------
  10 | jack
  13 | adam
  14 | liza
(3 rows)
```

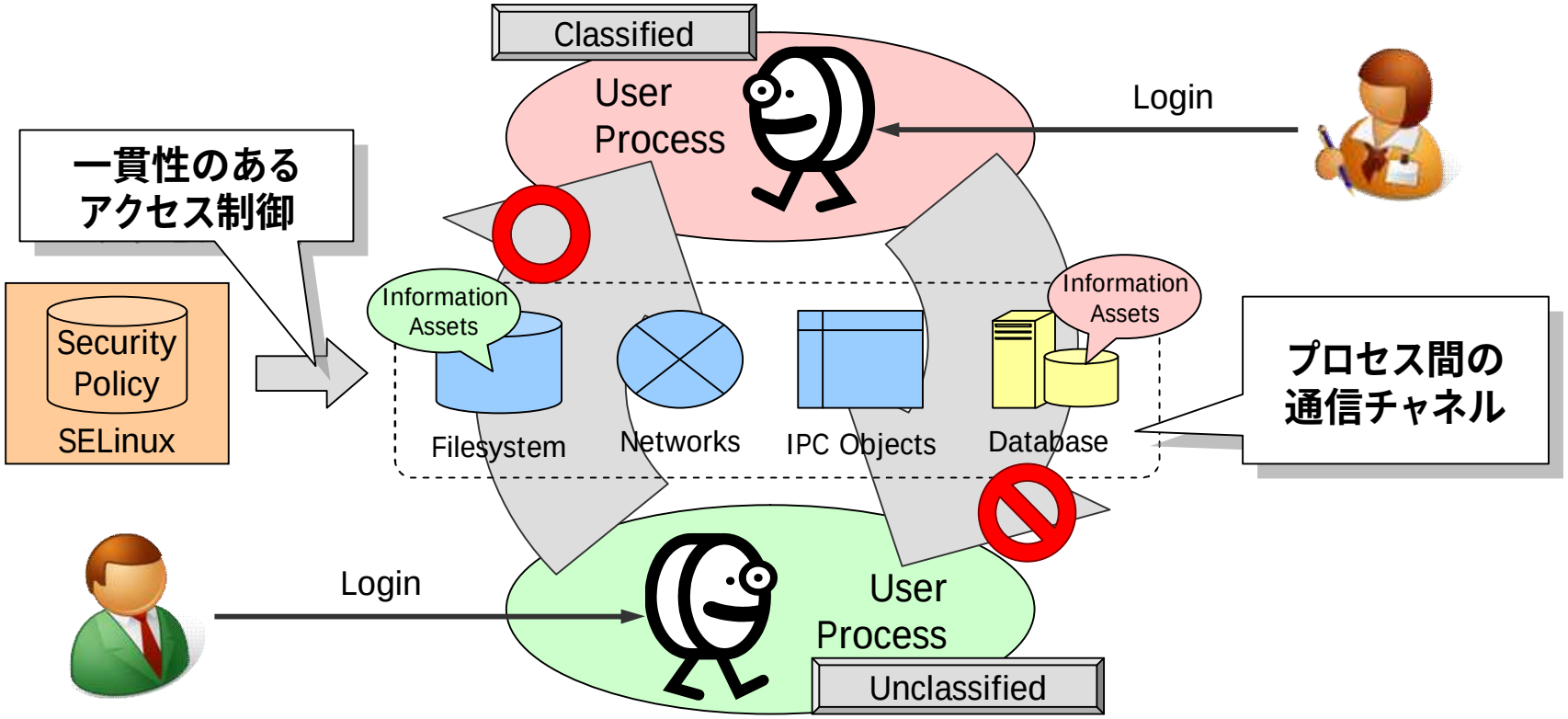
SE-PostgreSQLは、権限のない利用者が"Secret"とラベル付けされた列を参照する事を禁止

SE-PostgreSQLのパフォーマンス



- 2~4%程度のパフォーマンス上のトレードオフ
  - userspace AVCがカーネル呼び出しを最小化
- 測定環境
  - CPU Xeon (2.33GHz) Dual, Mem: 2GB (shared_buffer=512m)
  - **pgbench -c 2 -t 200000 による測定**

システムイメージ: System-wide consistency in access control

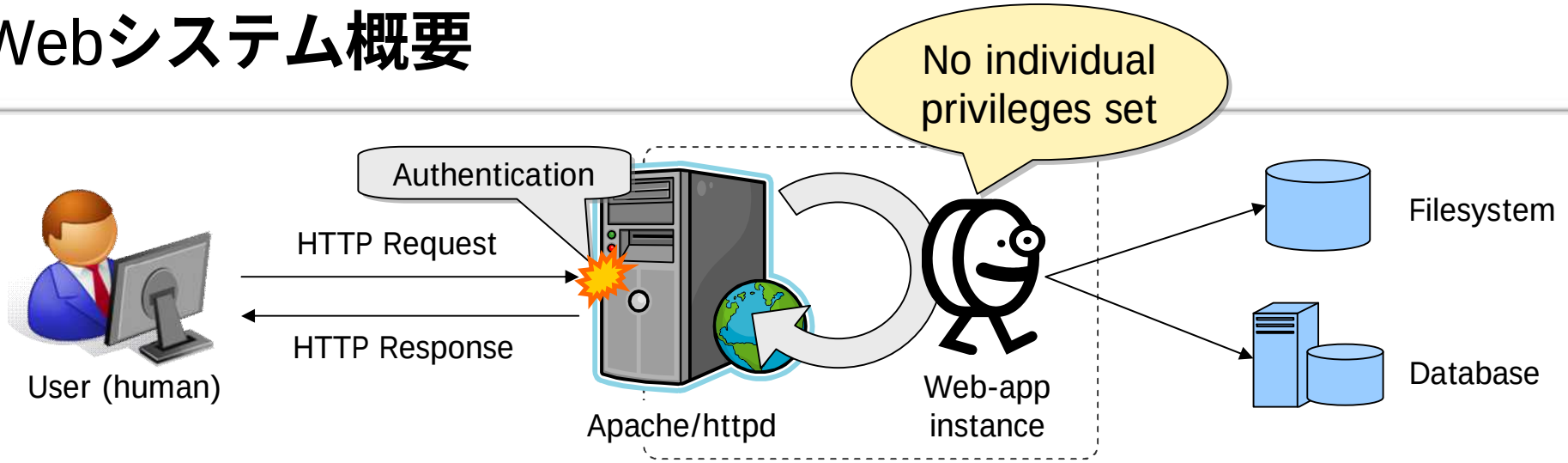


- SELinuxは全てのプロセス間通信チャネルを制御 (Data-Flow-Control)
  - No read-up, No write-down
- SE-PostgreSQLによってRDBMSをこの枠組みに組み込む事が可能に
  - アクセス制御の視点から見た時、FSとDBの間に違いなどない

Empowered by Innovation NEC

1. Background
2. SE-PostgreSQL
3. Apache/SELinux Plus
4. LAPP/SELinux

Webシステム概要



利用者のリクエストを処理する手順

1. 利用者はHTTP要求を送出
2. Apache/httpdはHTTP認証を実行（するかも）
3. 利用者のエージェントとしてWebアプリを実行
   - Webアプリの権限セットは、Webサーバプロセスと同一
4. Apache/httpdはHTTPレスポンスを応答

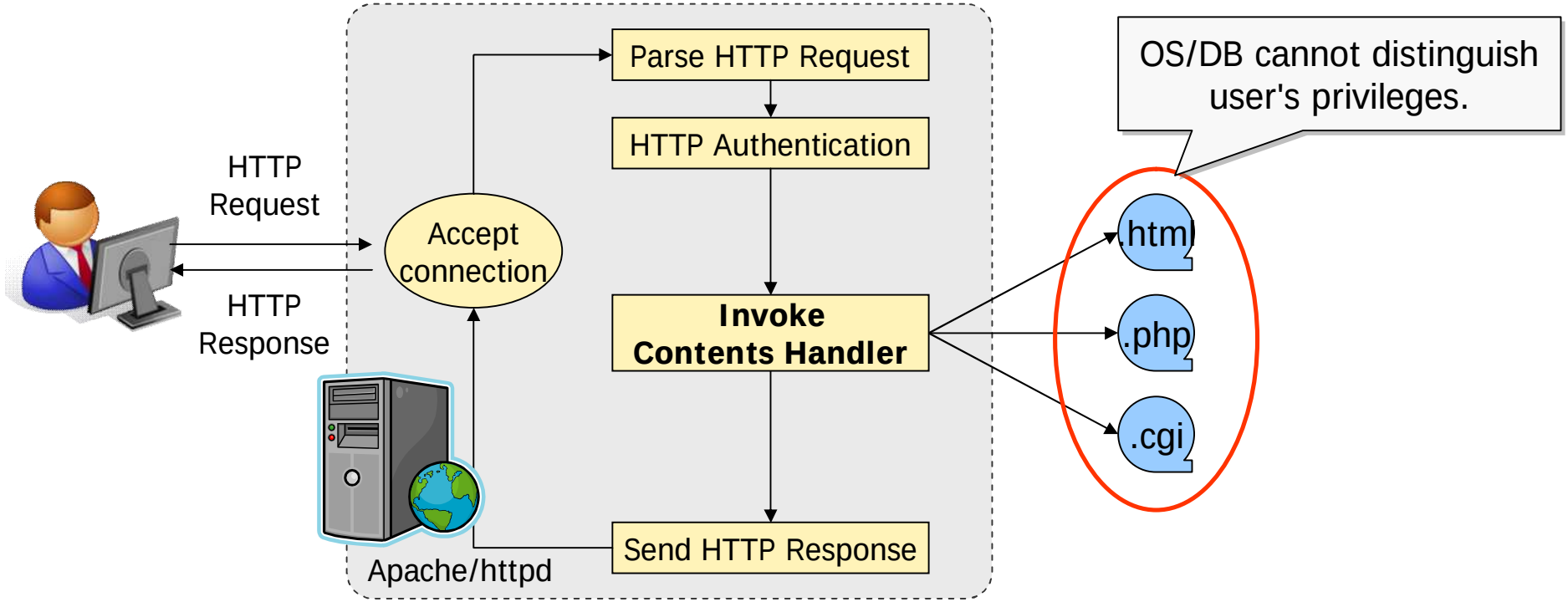
我々の頭痛の種とその処方箋

- OS/DBは、利用者エージェントに意味あるアクセス制御を実施できない
- Webアプリの実行前に、利用者に応じた権限を付与する必要がある
- **Apache/SELinux Plusモジュール**

Apache/SELinux Plus (1/2)

Apache（SELinuxサポートなし）

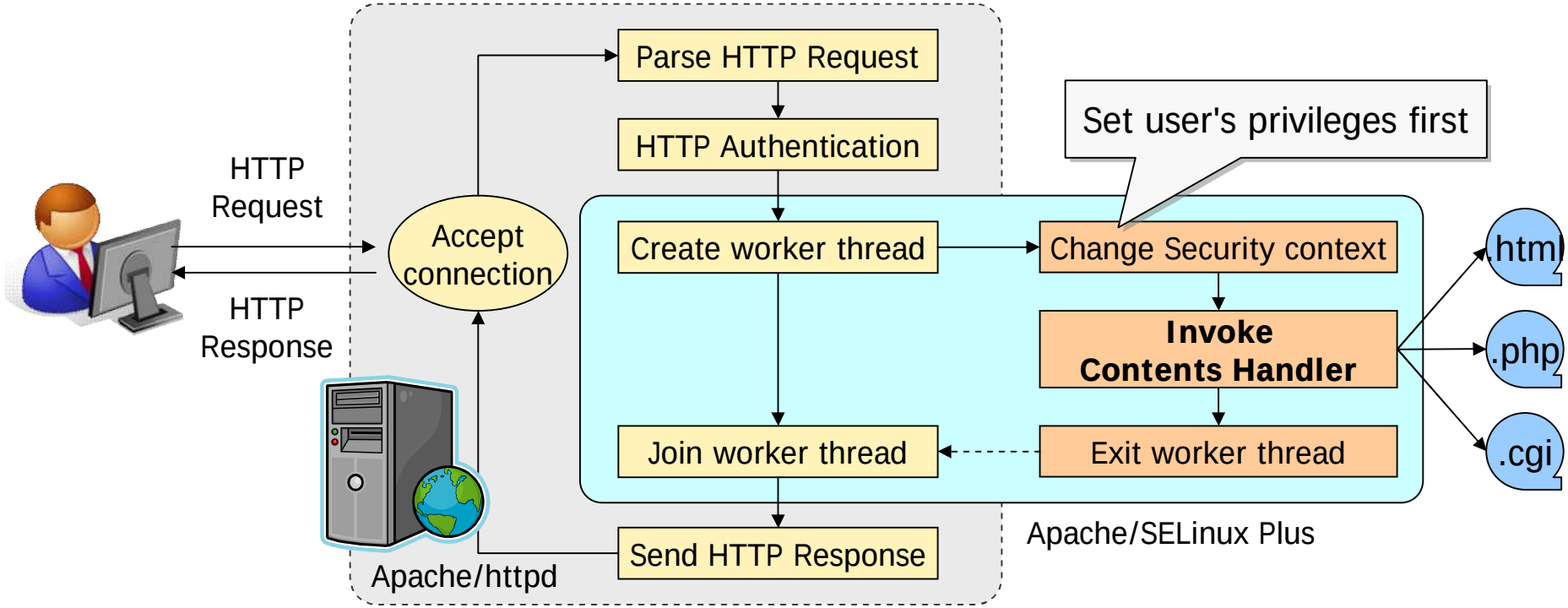
1. 利用者からのHTTP要求
2. HTTP認証を適用（するかも）
3. **サーバプロセスと同一の権限**で、要求されたコンテンツハンドラを実行
   - 単純に、アクセス制御の負担をWebアプリケーションに押し付けている



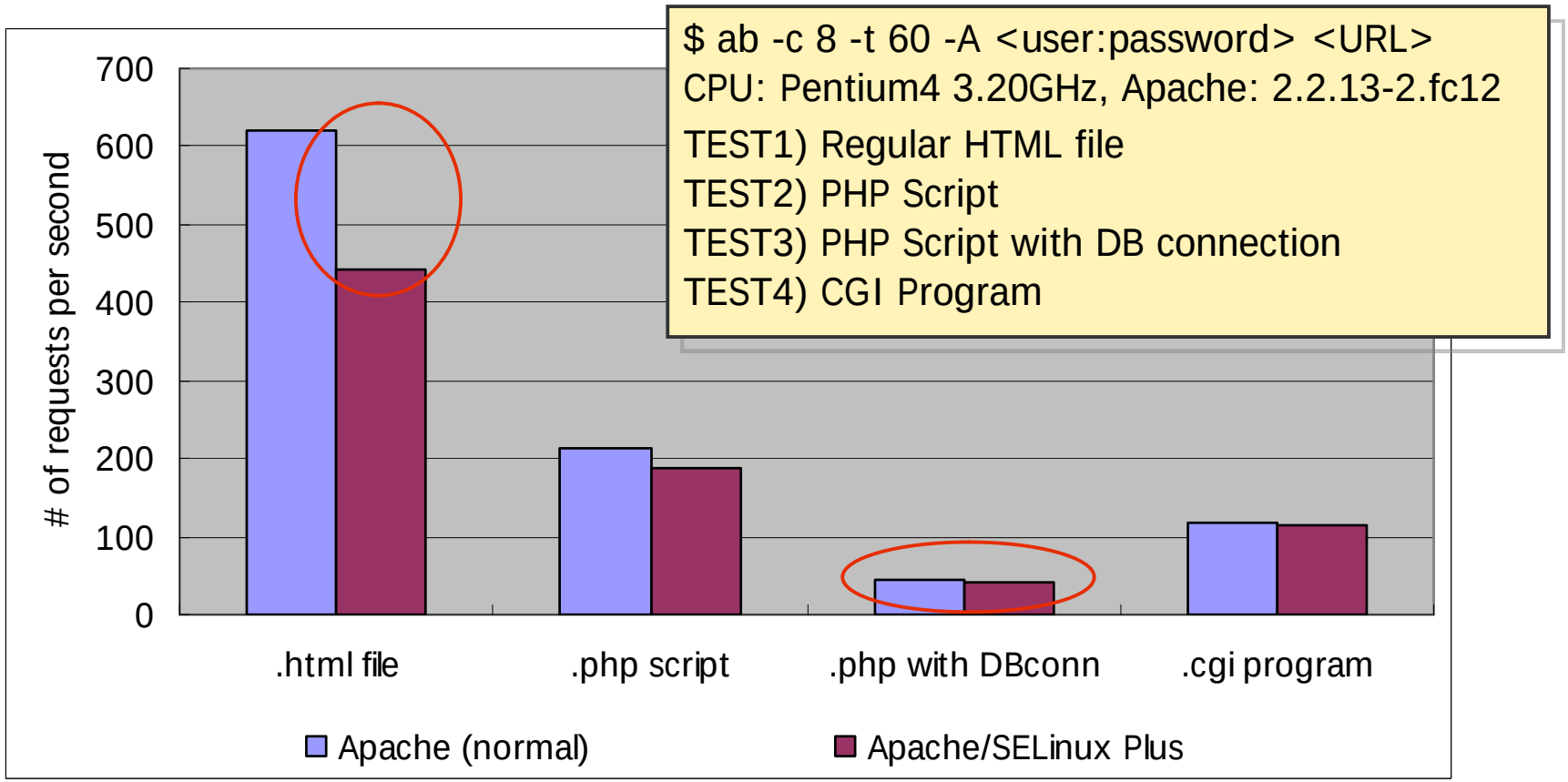
Apache/SELinux Plus (2/2)

Apache/SELinux Plus

1. **利用者からのHTTP要求**
2. **HTTP認証を適用（するかも）**
3. **ワンタイムの作業スレッドを生成、親はその終了を待つ**
4. **作業スレッドは、利用者の権限を自分自身に設定。**
   その後、コンテンツハンドラを実行する

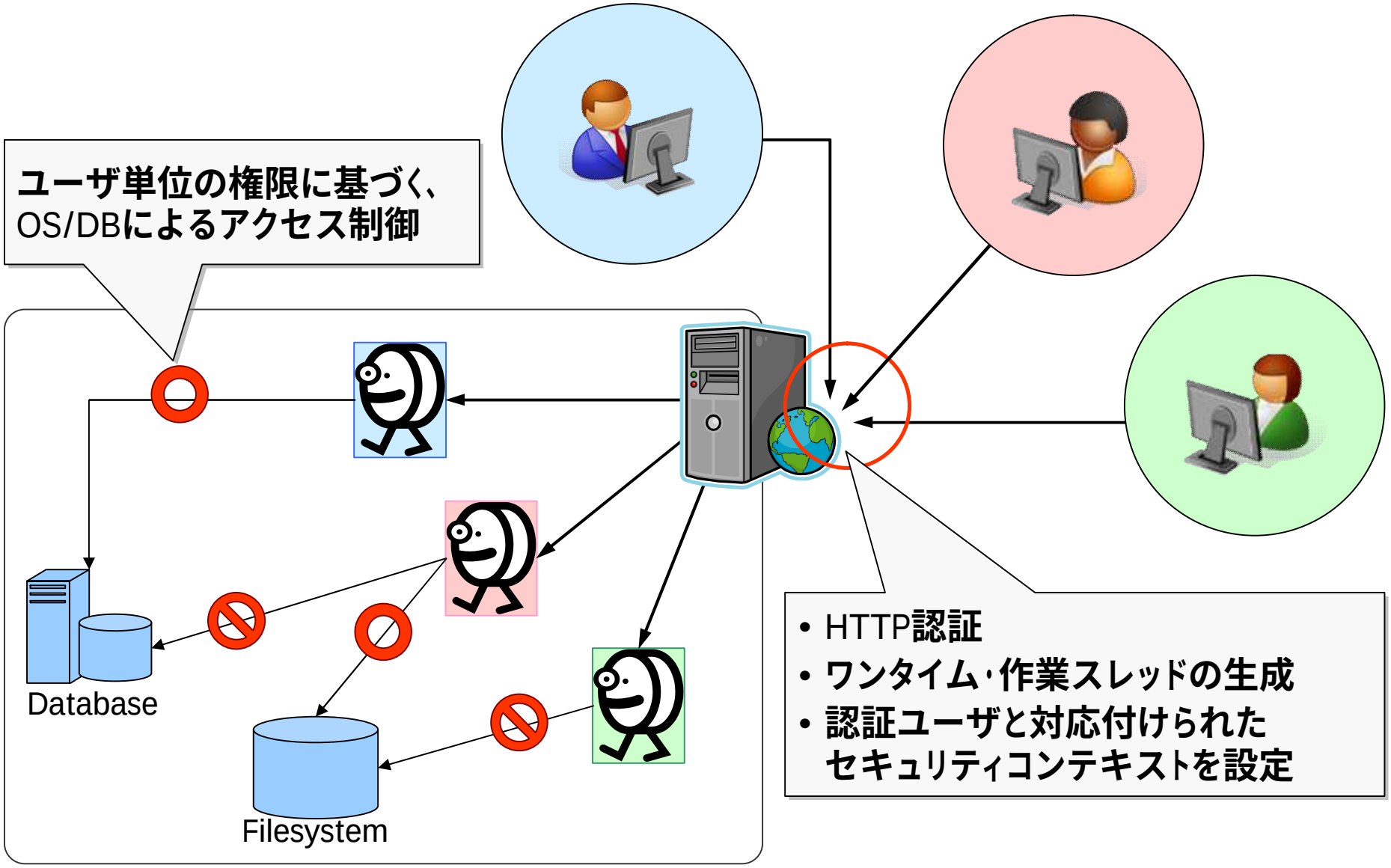


Apache/SELinux Plusのパフォーマンス

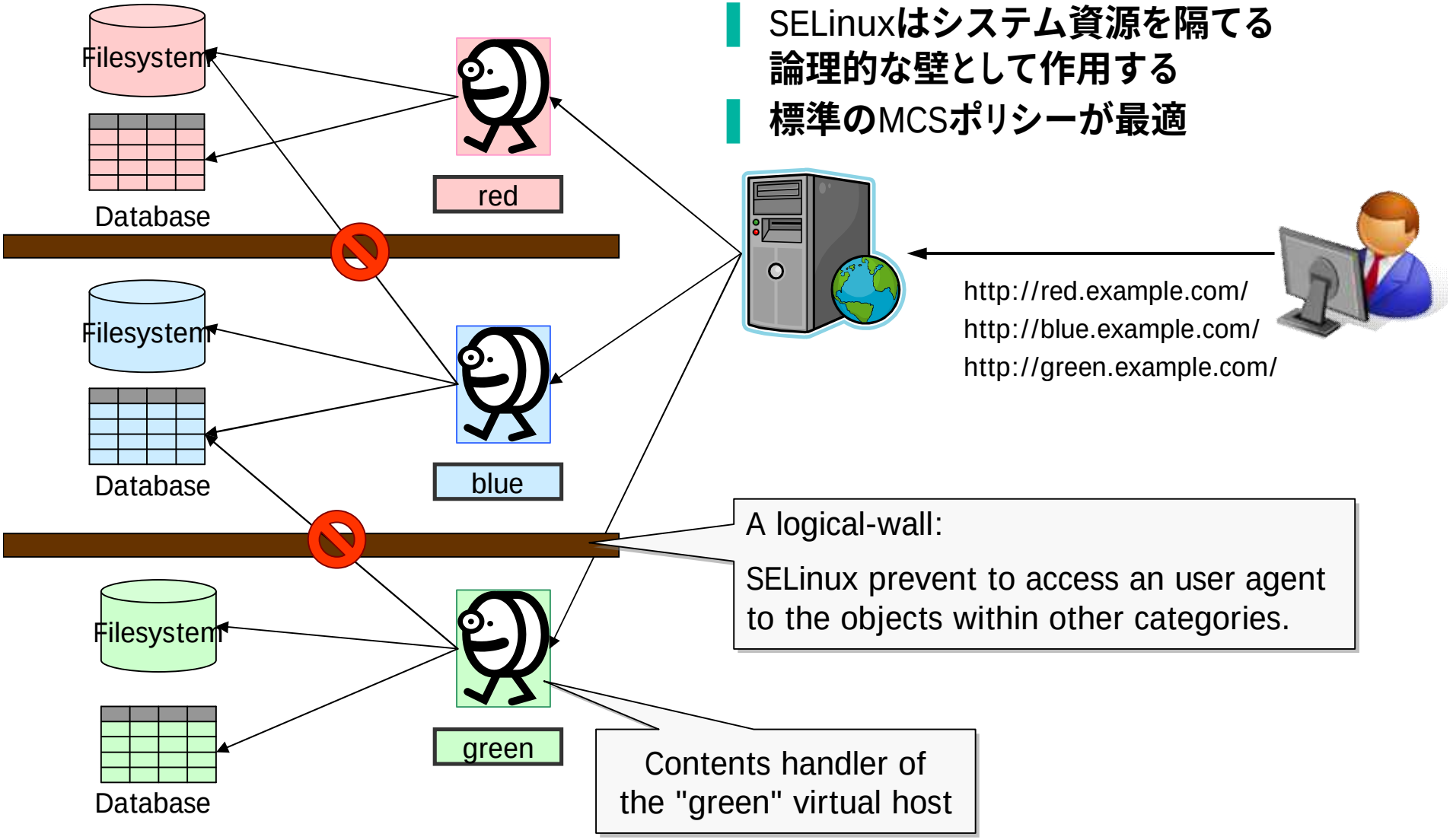


- スレッド生成/権限の付与は、軽いリクエストに比べて相対的に重い
- 主たるターゲットに対しては、ほとんど差は無い (Web+DBアプリ)
  - DBクエリの実行は、作業スレッドの生成よりも大きなコスト

システムイメージ (1/2)：Webユーザ単位の権限付与



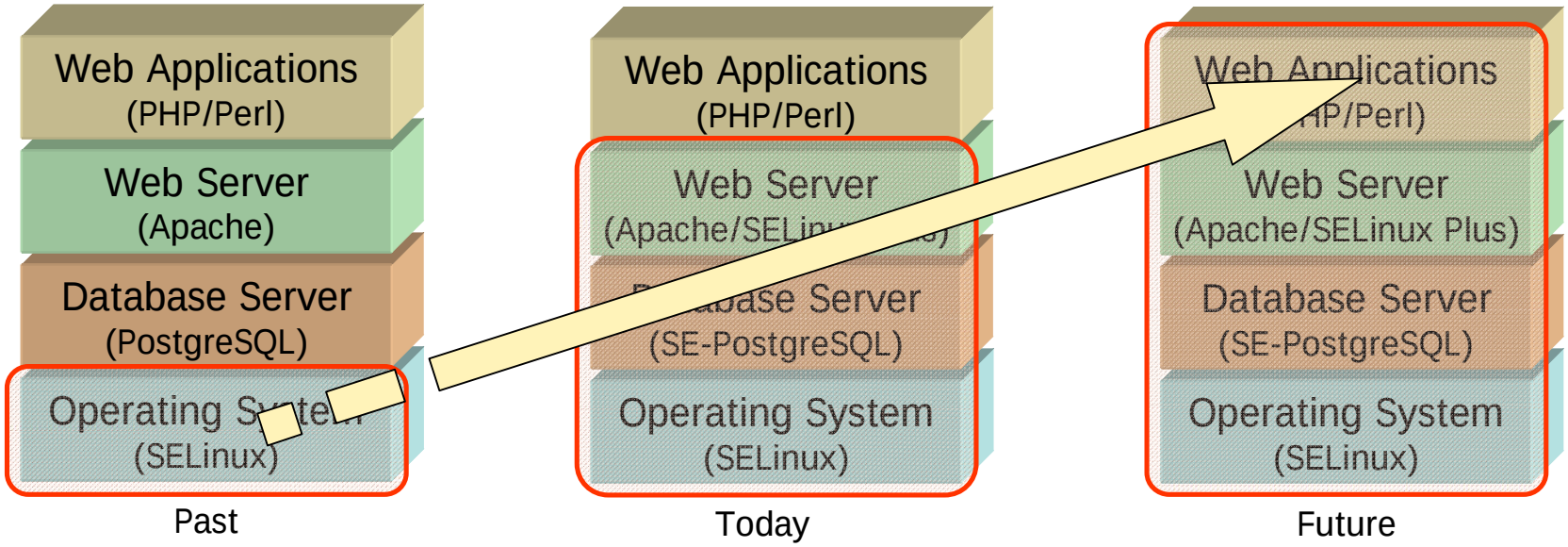
システムイメージ (2/2)：Virtual Host単位の分離



Empowered by Innovation NEC

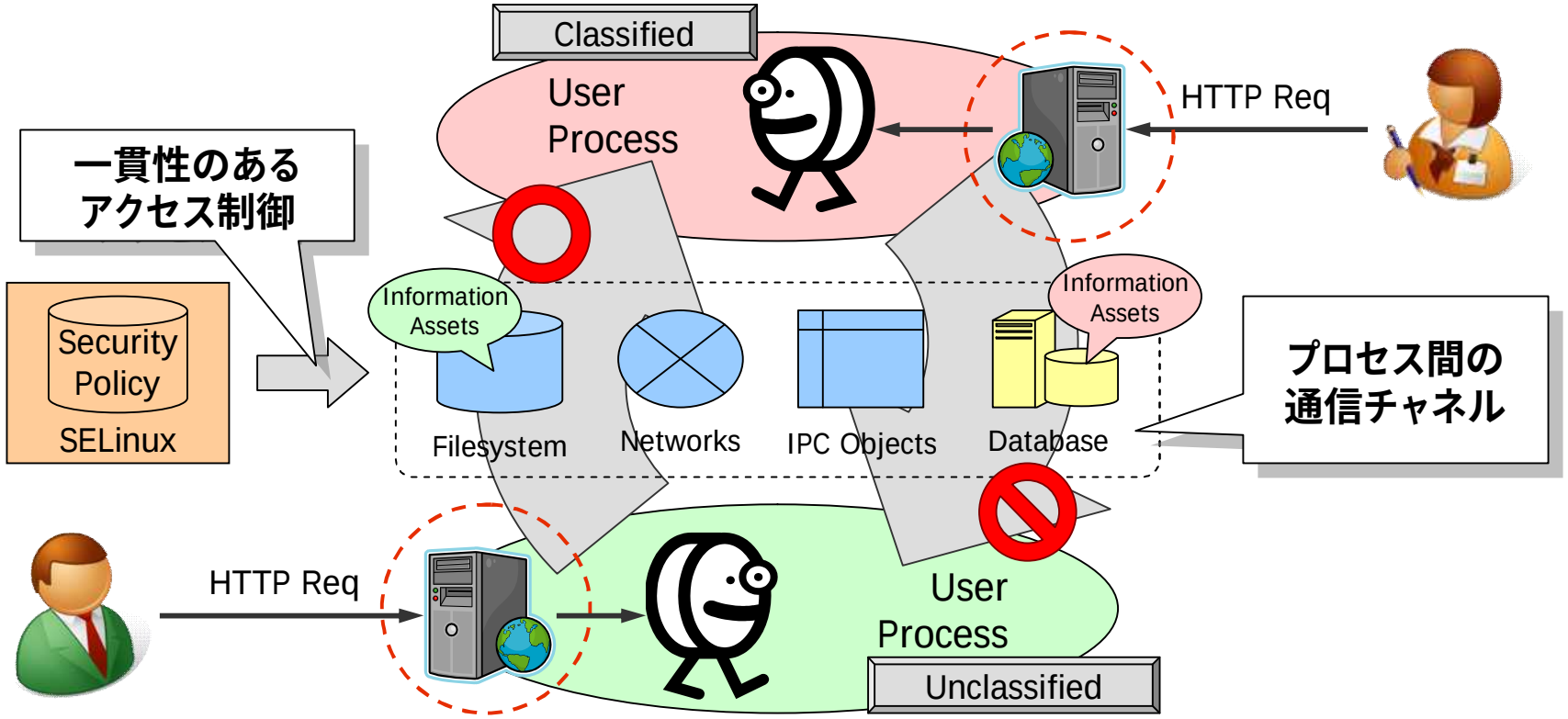
1. Background
2. SE-PostgreSQL
3. Apache/SELinux Plus
4. LAPP/SELinux

SELinuxの進化



- **先史時代：強制アクセス機能は存在しなかった**
- **過去：　SELinuxがOSの機能として採用される**
- **現在：　SELinuxのカバレッジ範囲はOS以外にも拡大**
  - SE-PostgreSQL, Apache/SELinux Plus, X-ACE/SELinux, sVirt, …
- **未来：　Webアプリケーションスタックの全域をカバー**

LAPP/SELinuxの概念図

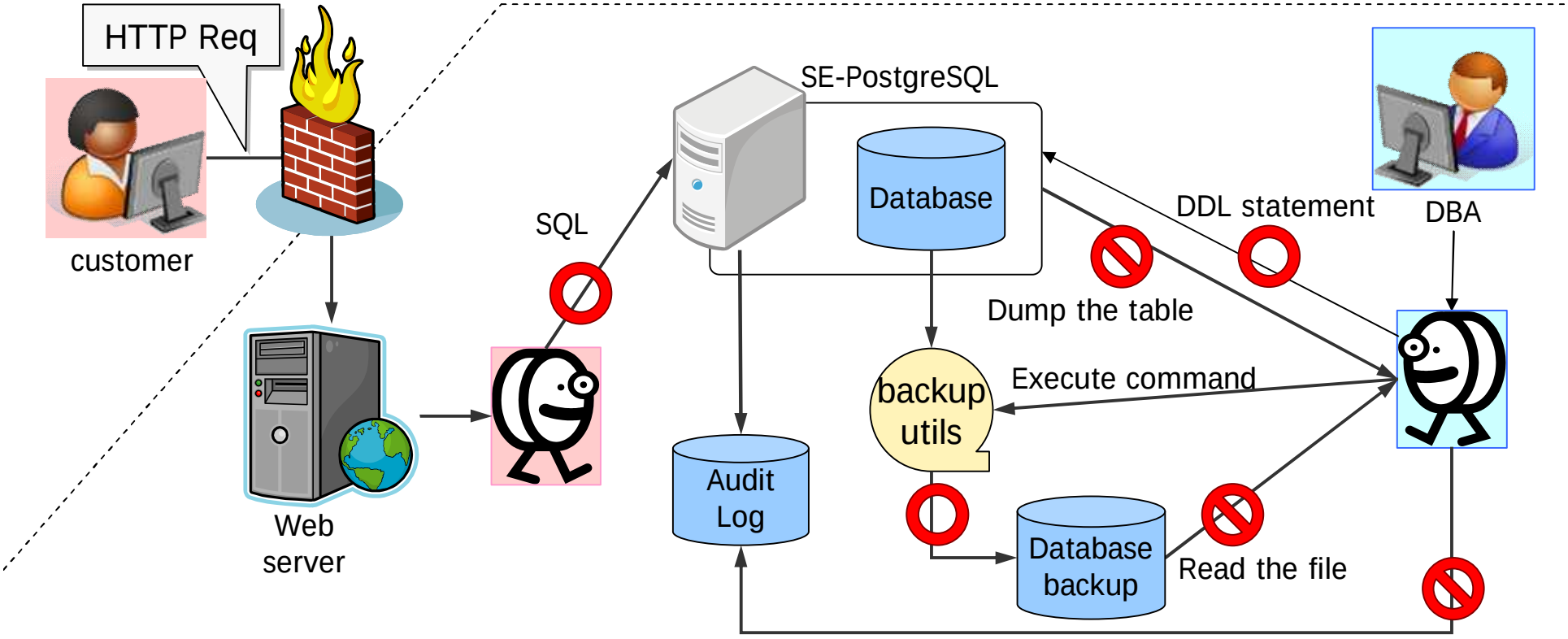


- SE-PostgreSQLは、アクセス制御におけるシステムワイドな一貫性を提供
- Apache/SELinux plusは、正しい権限でWebアプリを実行する事を可能に
- LAPP/SELinuxは、アクセス制御における『一貫性』と『網羅性』を共に担保して、Webシステムを構築する事を可能にします。

LAPP/SELinux applicability: 内部犯行の防止

Database Administrator (DBA)

- **通常、Databaseに対する全ての操作が許可されている**
- **だが、不必要な情報への参照は禁止されるべき。例えば、バックアップの中身**
- **SELinuxによる一貫したアクセス制御が内部犯による情報漏えいを防止できる**



原則

情報資産の価値

- コンテンツがその価値を決める。情報の保管手段ではない。
- 同じ利用者/情報資産のペアに対して、一貫性のあるアクセス制御が必要

アクセス制御の目的

- 特定の利用者（人間）と情報資産の間に、許可/拒否される操作の組を定義する事
  - 『人間』と『情報』との間の関係である事をお忘れなく！

LAPP/SELinuxにおける原則

- 共通のセキュリティ識別子 = セキュリティコンテキスト
- 共通のアクセス制御意思決定 = セキュリティポリシー
- プラットフォーム機能を最大限活用する = SELinux

プロジェクトの現在と、これまでの歴史

Status

- Linuxカーネルは LAPP/SELinuxが必要とする全ての機能をサポート
- Fedora Projectでは sepostgresql 及び mod_selinux パッケージを提供
- SE-PostgreSQLは PgSQL開発者コミュニティで議論の途中

History

- '06/09 SE-PostgreSQLの開発スタート
- '07/03 SELinux Symposium & Developer Summit 2007 (Baltimare, USA)
- '07/08 Fedora にて SE-PostgreSQL のパッケージが採用 (F8 or later)
- '07/11 IPA未踏ソフトにて"スーパークリエーター"として受賞
- '08/03 The PostgreSQL conference 2008 (Ottawa, CA)
- '08/05 SE-PostgreSQLをv8.4.x開発サイクルに提案 (継続)
- '08/12 スレッド単位セキュリティコンテキスト機能がマージ (2.6.28 or later)
- '09/04 Fedora にて Apache/SELinux Plus のパッケージ採用(F11 or later)
- '09/07 SE-PostgreSQLをv8.5.x開発サイクルに提案
- '09/10 Japan Linux Symposium 2009 での発表
- '09/11 Japan PostgreSQL Conference 2009 での発表

Any Questions?

Empowered by Innovation NEC

Thank you!