LAPP/SELinux

- A secure web application platform powered by SELinux -

NEC OSS Promotion Center
KaiGai Kohei
<kaigai@ak.jp.nec.com>

Self Introduction

- Working for NEC, come from Tokyo, Japan
- 6 year's experience in Linux kernel development
  - Especially, SELinux and security related region
    - SMP Scalability improvement (2.6.11)
    - XATTR Support in JFFS (2.6.18)
    - SELinux support in busybox
    - Type boundary and Multithreading (2.6.28)
    - Security-Enhanced PostgreSQL

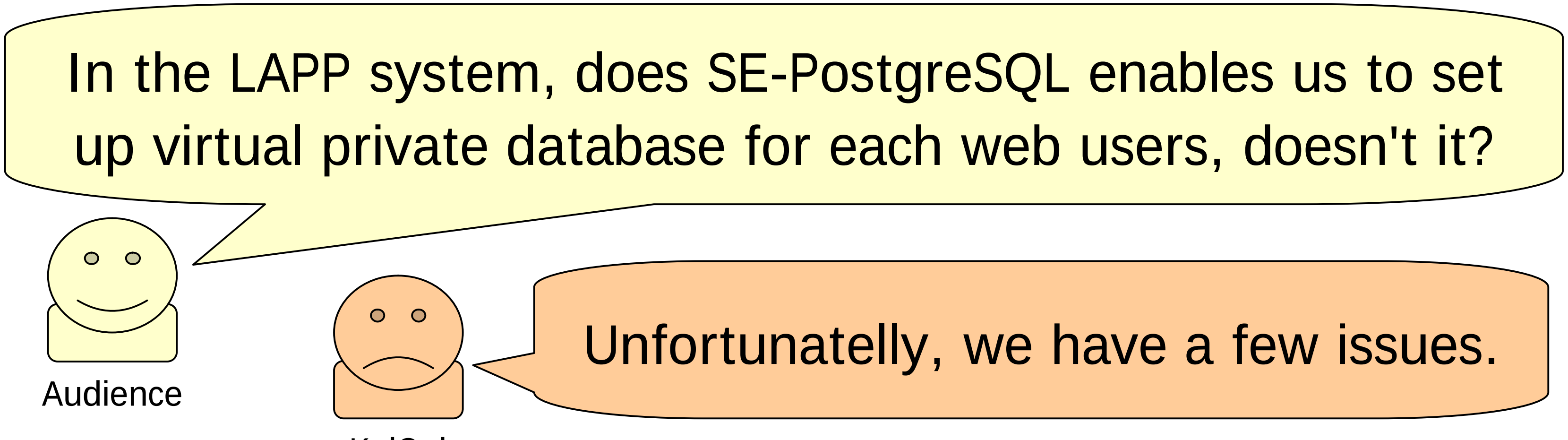
One of the core components in LAPP/SELinux

Security-Enhanced PostgreSQL

- Concept
  - System-wide consistency in access controls
    - ✓ It shares a common security policy between OS and RDBMS
  - Fine-grained mandatory access controls on DB objects
  - Client's privileges based on Labeled IPsec feature
- Status
  - Now progress in PostgreSQL v8.4 development cycle
  - Available on Fedora8 or later
- Promotions
  - Many of talks for the last 2 years....
    - ✓ SELinux Symposium, PGcon, IPA Forum, etc...
    - ➧ I got a "frequently asked question".

PGcon2008
Univ of Ottawa (23 May 2008)

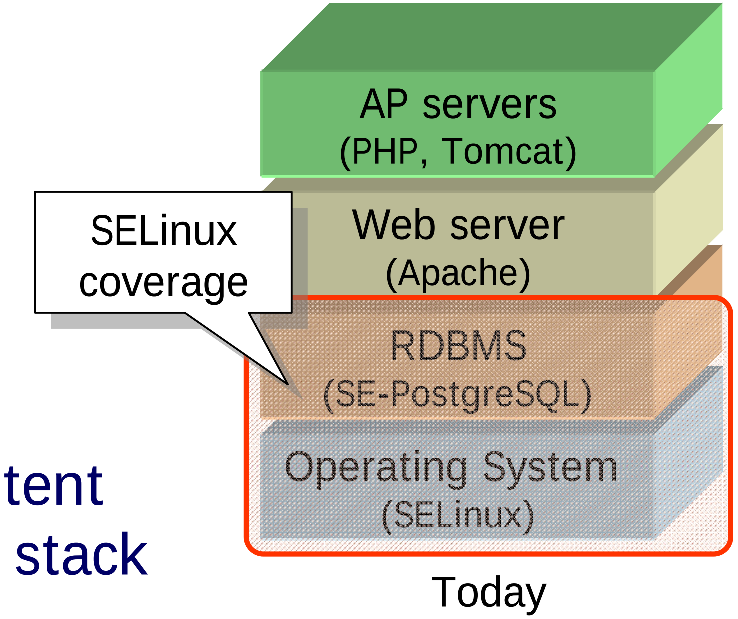
A Frequently Asked Question



In the LAPP system, does SE-PostgreSQL enables us to set up virtual private database for each web users, doesn't it?

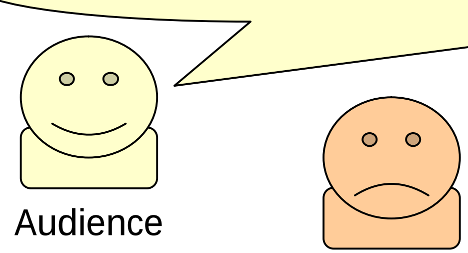
Unfortunatelly, we have a few issues.

- A few issues:
  - Not separated domains
  - Multi-threading web application
- Our goal
  - SELinux as a foundation of consistent access controls on whole of LAPP stack



A Frequently Asked Question

In the LAPP system, does SE-PostgreSQL enables us to set up virtual private database for each web users, doesn't it?



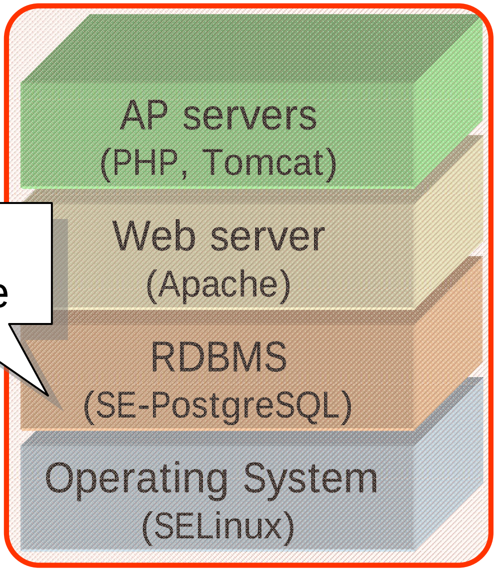
Unfortunatelly, we have a few issues.

- A fe
  - N
  - Multi-
- Our goal
  - SELinux as a foundation of consistent access controls on whole of LAPP stack

We call it

LAPP/SELinux

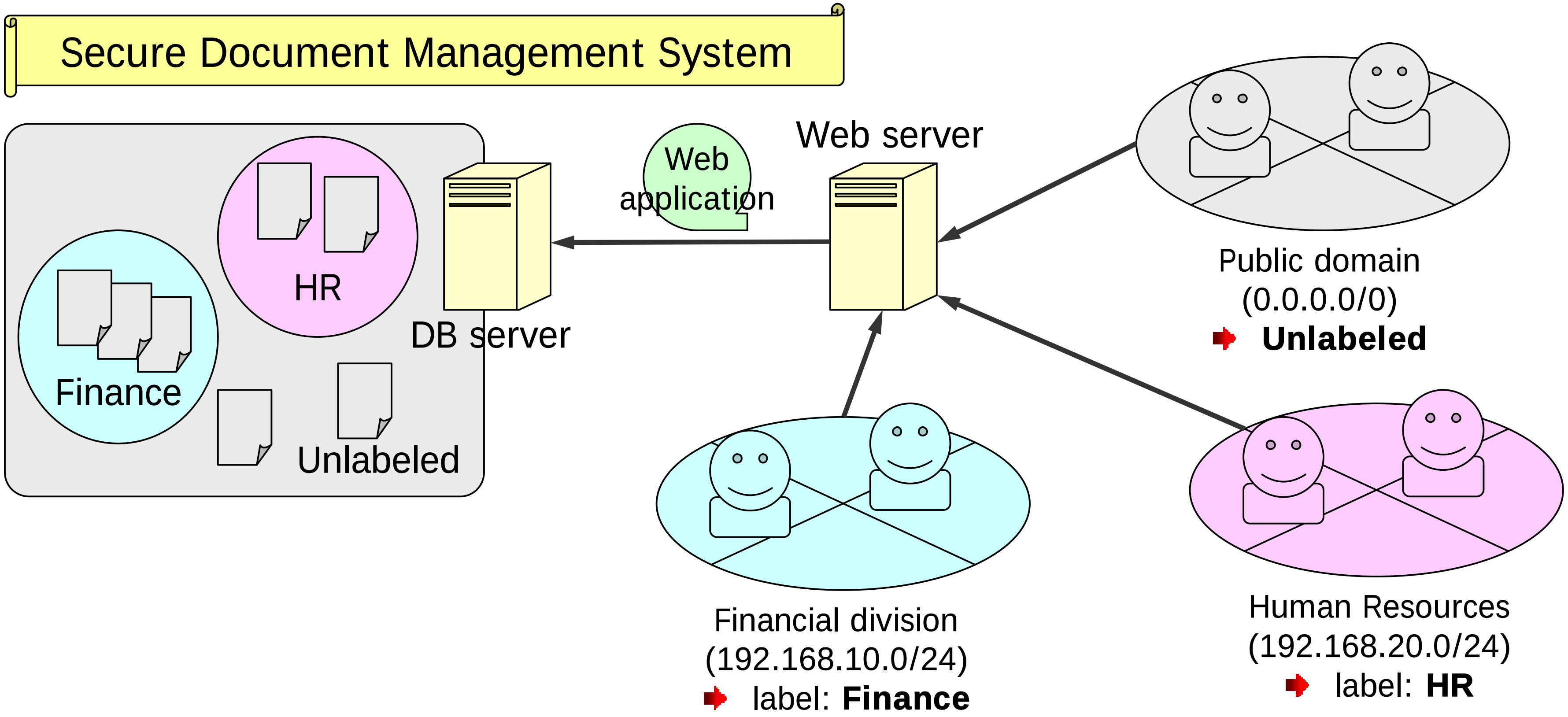
SELinux coverage



Future

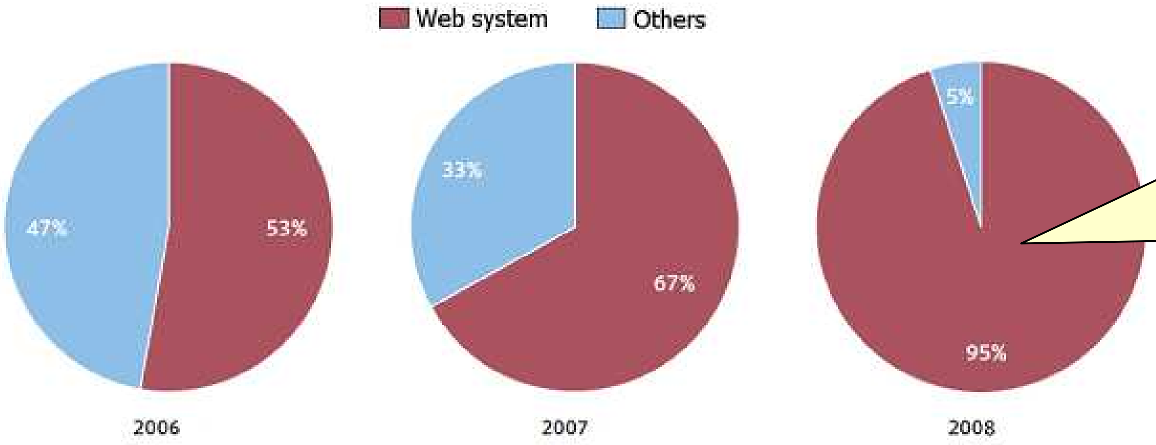
Example: A system image of LAPP/SELinux

- Web application works with correct security context
- DB objects are labeled, and MAC policy is applied on accesses
- ➡ Correct access controls, even if Web-application is very buggy!

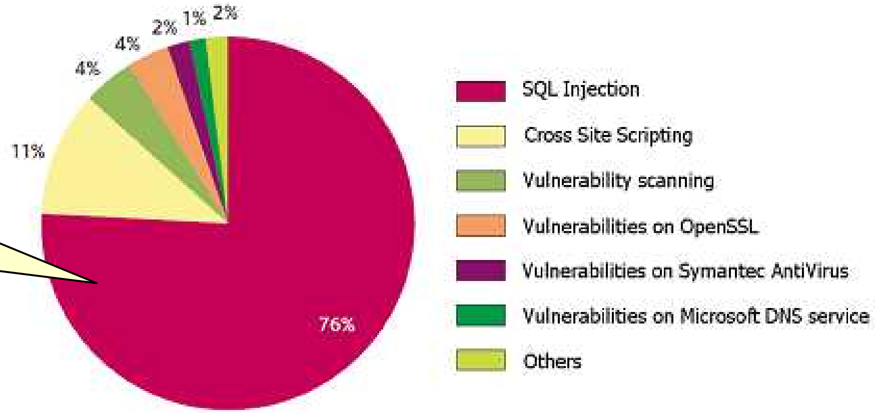


Background: Web application is a Nightmare!

- A security vendor in Japan reported as....



95% of attacks targeted on web applications, 2008.



76% of attacks tried SQL injection, 2008.

Source: Vulnerability Analysis Report vol.11, Lac Inc

Can SELinux provide a solution?

Yes, we can!

Issues need to be considered

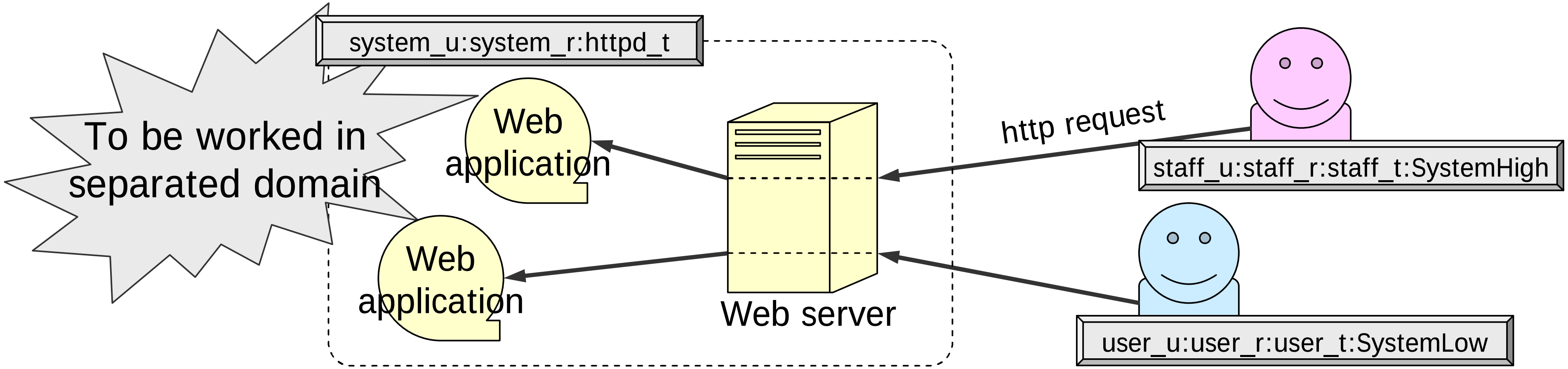
- Not a separated domain
- Multi-threading web application

Primarily, how should it be considered?

- Definitions
  - **Access control** is the ability to permit/deny uses of paticular resources by particular users.
  - **User** is a human, not a computer program.
  - **Process** is an agent of user in computer system.
    - ➡ So, access control has to apply its policy on processes as if it is a user himself.
- How should the web be considered in this context?
  - **User** accesses paticular resources via its **agent**.
  - **User** accesses paticular resources via web interface, and it invokes web-application as its **agent**.
  - ➡ No fundamental differences are here!

Issue: Not a separated domain

- Privileges of web applications
  - Web server handles all the HTTP request by itself.
  - ➡ OS does not consider it as works of a agent of clients.
  - ➡ Web application has to apply **its own access controls**
- Issues in this scheme
  - How to make sure web-app's access controls are not flaw?
  - Who does it actually requires to access on resources?

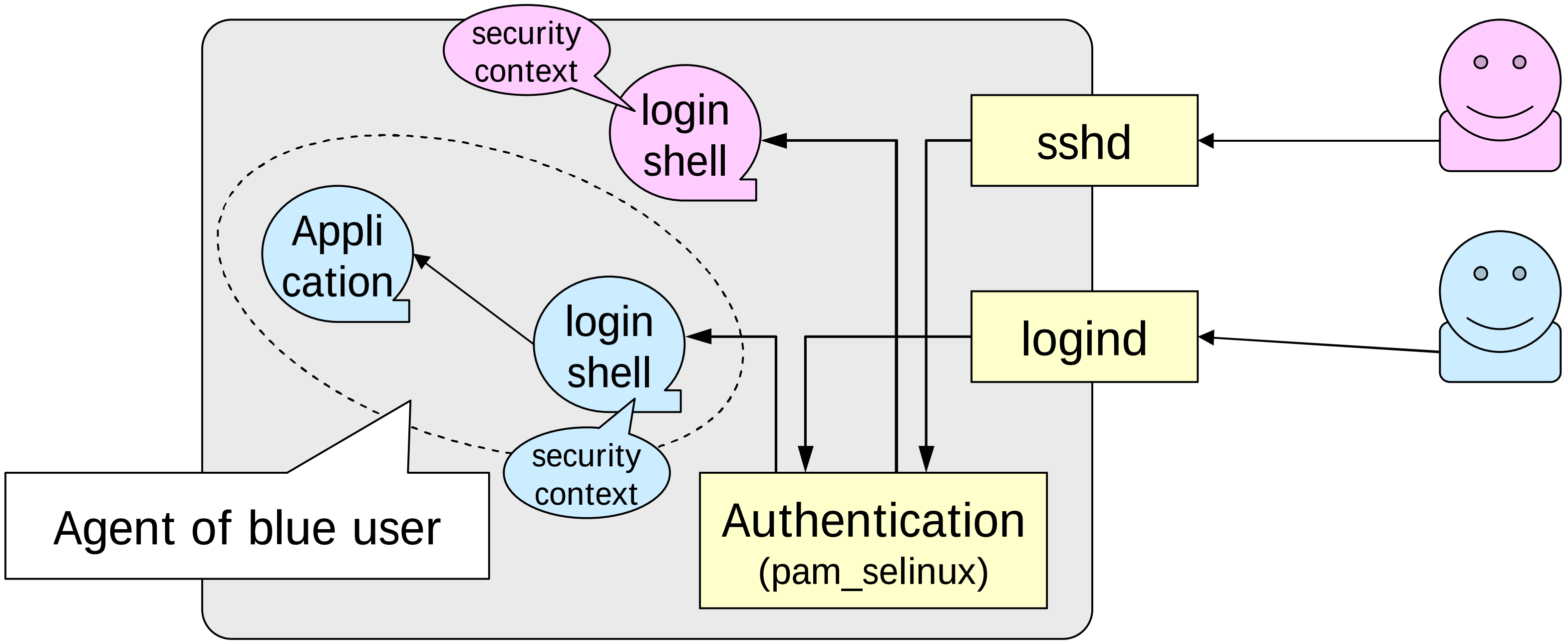

system_u:system_r:httpd_t
To be worked in separated domain
Web application
Web application
Web server
http request
staff_u:staff_r:staff_t:SystemHigh
user_u:user_r:user_t:SystemLow

SELinux and security context

- SELinux
  - It can provide various kind of object managers its decision on access controls.
    - ✓ Operating system, RDBMS, X-Window system, ...
  - Its decision come from security context of agent and resources to be accessed.
  - ➡ How should correct security context be assigned to the agent?
- Strategies
  - Authentication
  - Labeled Networking Technology
  - Do nothing

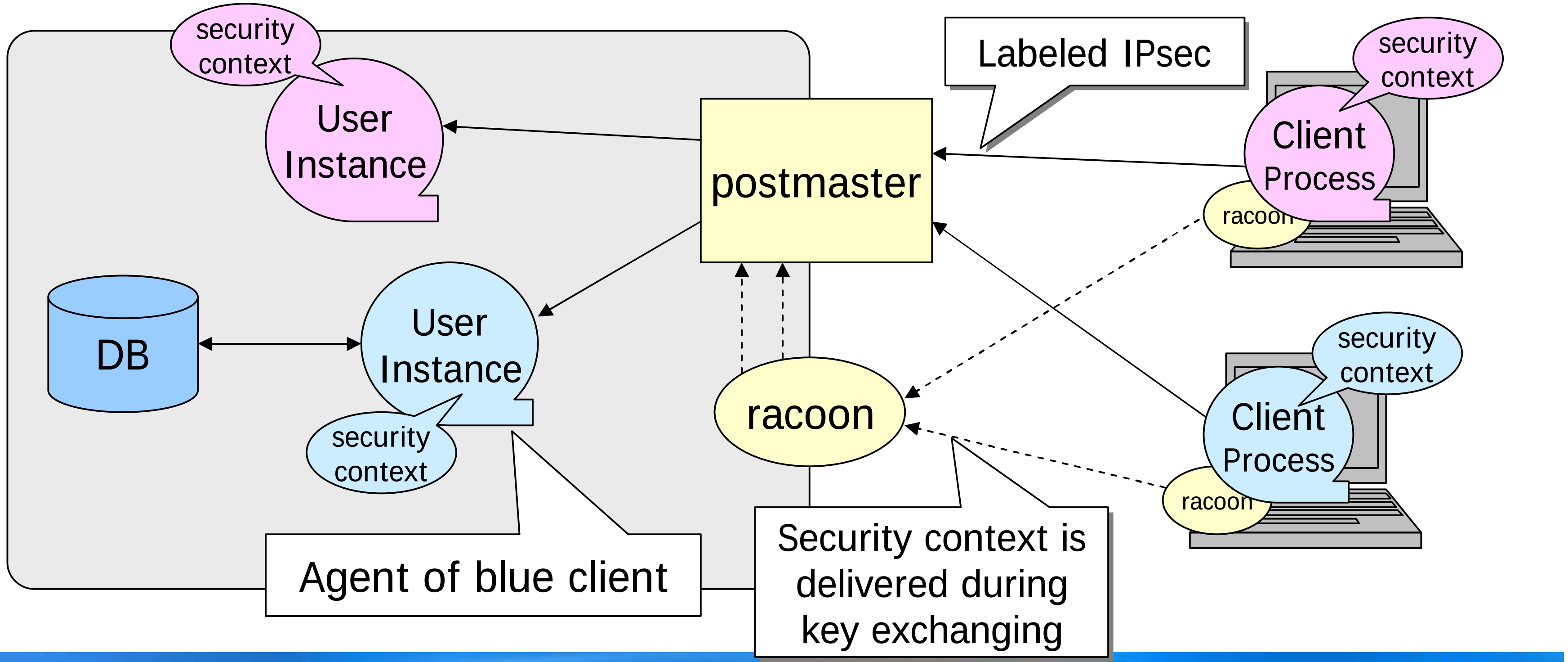
User/Security context assignment (1/3)

- Strategy.1 Authentication
  - It assigns a security context to agent during authentication based on user’s identifier.
    - ✓ Case examples: Operating System



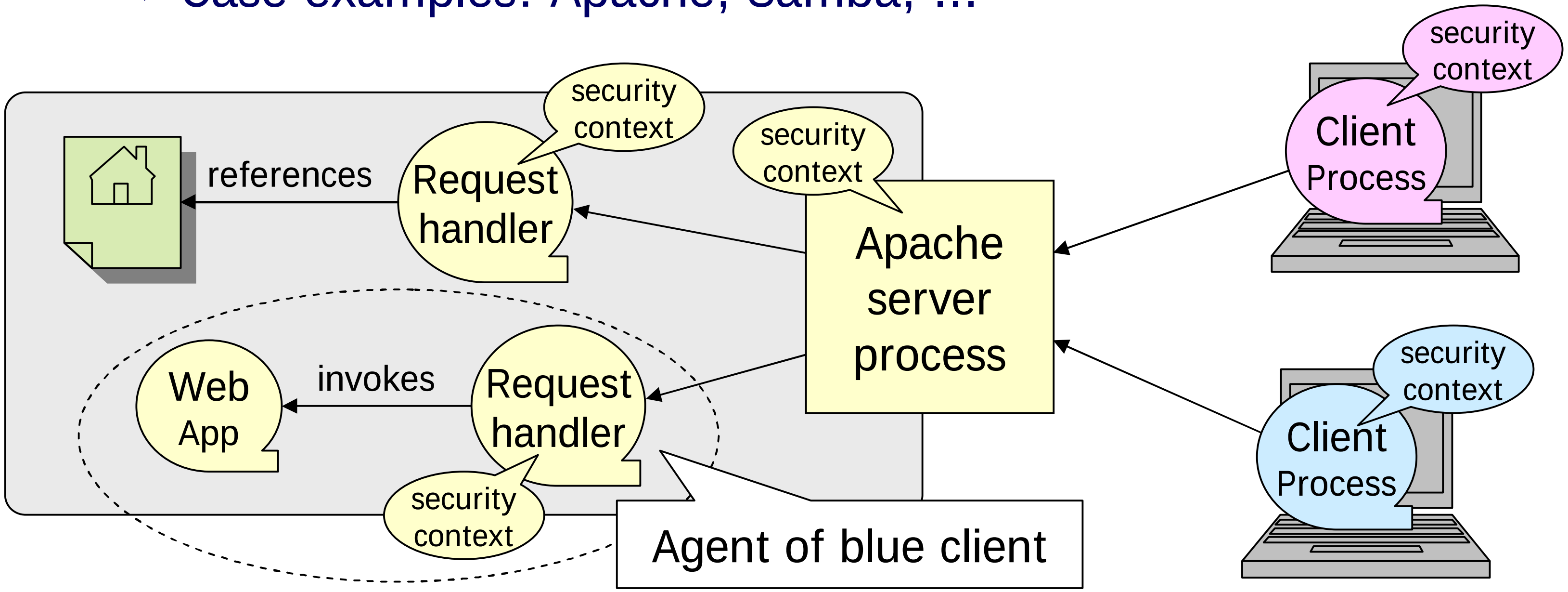
User/Security context assignment (2/3)

- Strategy.2 Labeled Networking Technology
  - It assigns a security context on agent based on the peer entity's one.
    - ✓ Case examples: SE-PostgreSQL, XACE/SELinux, Xinetd



User/Security context assignment (3/3)

- Strategy.3 Do nothing
  - It does not assign individual security context on agent.
    - ✓ Case examples: Apache, Samba, …



- Correct security context should be assigned on agent whenever user begins to use a system, but …

Solution

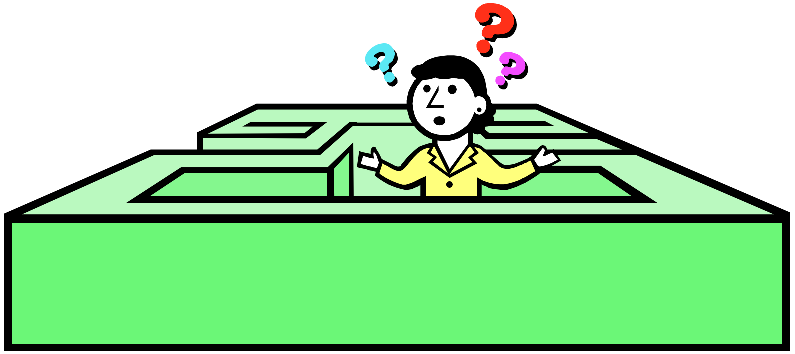
- Rules
  - Any agent should be assigned correct security context whenever user begin to use the system via agent.
    - ✓ User can execute a command via shell program.
    - ✓ User can refer a document via web-interfaces.
      - ➡ No fundamental differences.
  - It allows various strategies to determine security context.
- Items to be enhanced on Web server
  - It determine a security context of request handler.
  - It assigns it just before invocation of request handler.

⇩

- Web application can work under SELinux restriction!

Issue: Multi-threading web application

- Restriction
  - SELinux didn't allow to assign individual security context for each threads within a process.
    - It is quite natural restriction due to domain separation!
  - Some of applications handle user's request in multithreaded backends.
    - ✓ Apache 2.x, Tomcat, ...
  - ➡ We need to consider a reasonable solution.



Idea: Bounds Domain (1/2)

- What is bounds domain?
  - A domain with a hierarchical boundary of its privileges.
  - Bounded one cannot have any permission when its bounds domain does not have them.

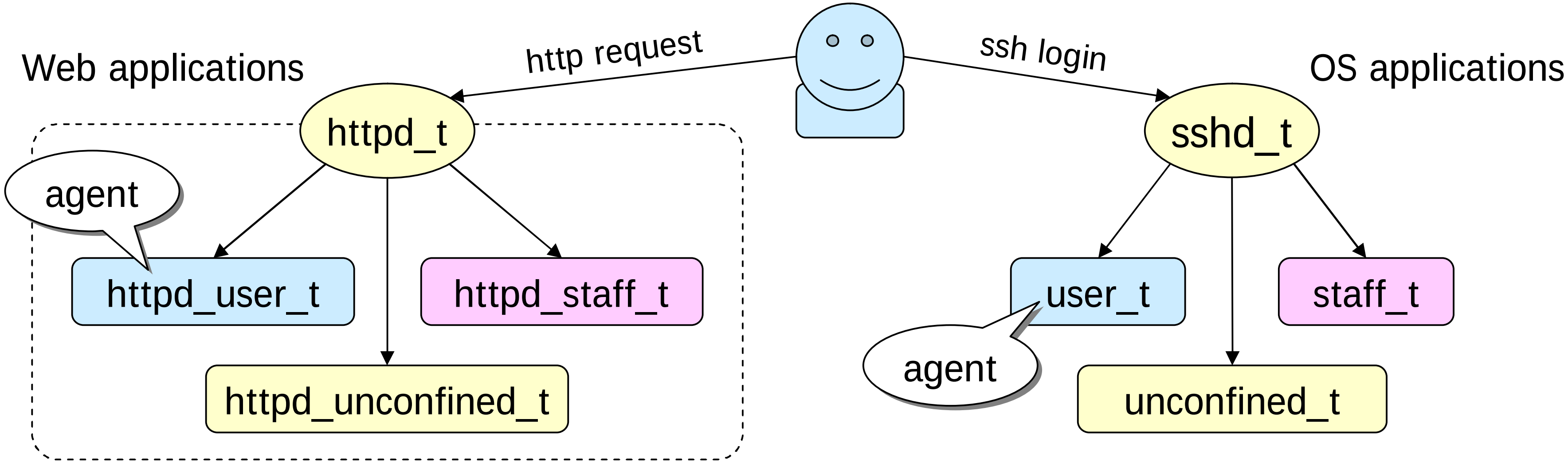
Example:

```
typebounds httpd_t httpd_child_t;
allow httpd_t       etc_t : file { getattr read };
allow httpd_child_t etc_t : file {         read ~~write~~ };
```

- A new **typebounds** statement defines a hierarchical relationship between two domains.
- **httpd_child_t** cannot have **file:{write}** due to lack of permissions on **httpd_t** which is the parent.
- It means child domain always has equal or smaller privilleges.

Idea: Bounds Domain (2/2)

- What does it make possible?
  - We can ensure that all the threads work within a process's privileges, even if they have individual domains.
  - ➡ Prerequisite of per-thread domain
  - We can also consider **`httpd_user_t`** as a restricted mode of **`httpd_t`** domain in this case.

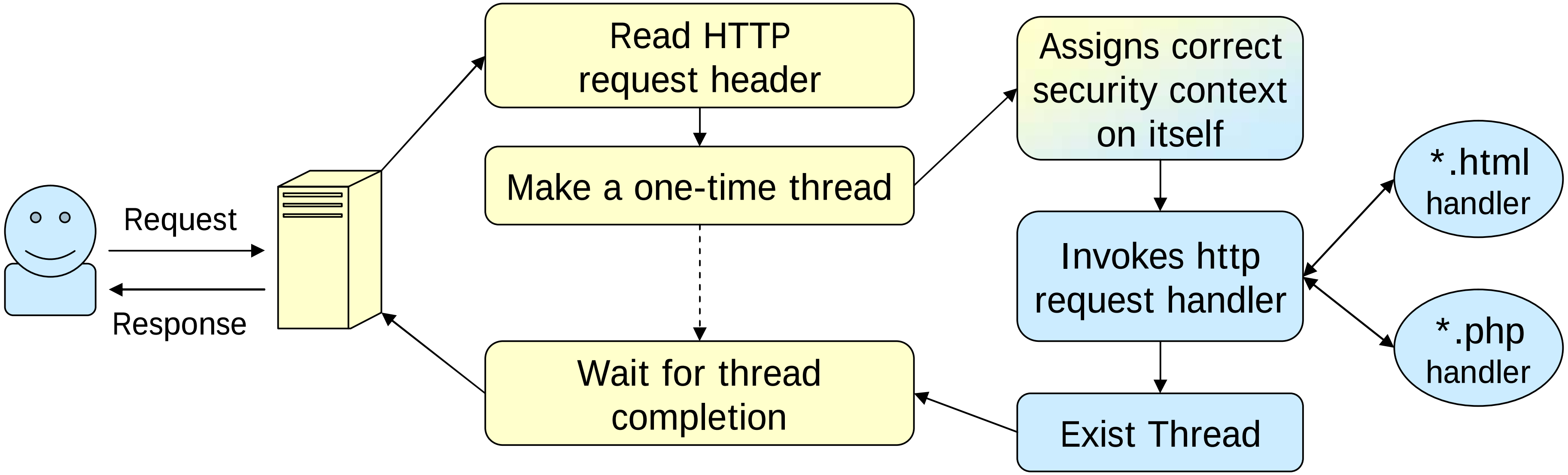


Apache/SELinux plus (1/2)

- What is Apache/SELinux plus?
  - An extension of Apache/httpd.
  - It assigns individual security context before invocation of request handler.
  - Currently, it determines the security context based on HTTP authentication or source IP address.
    - ✓ Note that it allows additional various strategies.
- What does it make possible?
  - It enables to associate an idea of “web user” and security context of SELinux.
    - Per web-user privileges on PHP scripts, static web contents, and so on...

Apache/SELinux plus (2/2)

- Internal design
  - It makes a one-time thread just before invocation of request handler, and parent waits for its completion.
  - The thread assigns correct security context on itself, then invokes request handler.
  - The thread exist, and parent wakes up.



Demonstration

- Security context of agent based on HTTP authentication
  - Result set of DB query depends on security context
    - It also applied on references to static contents

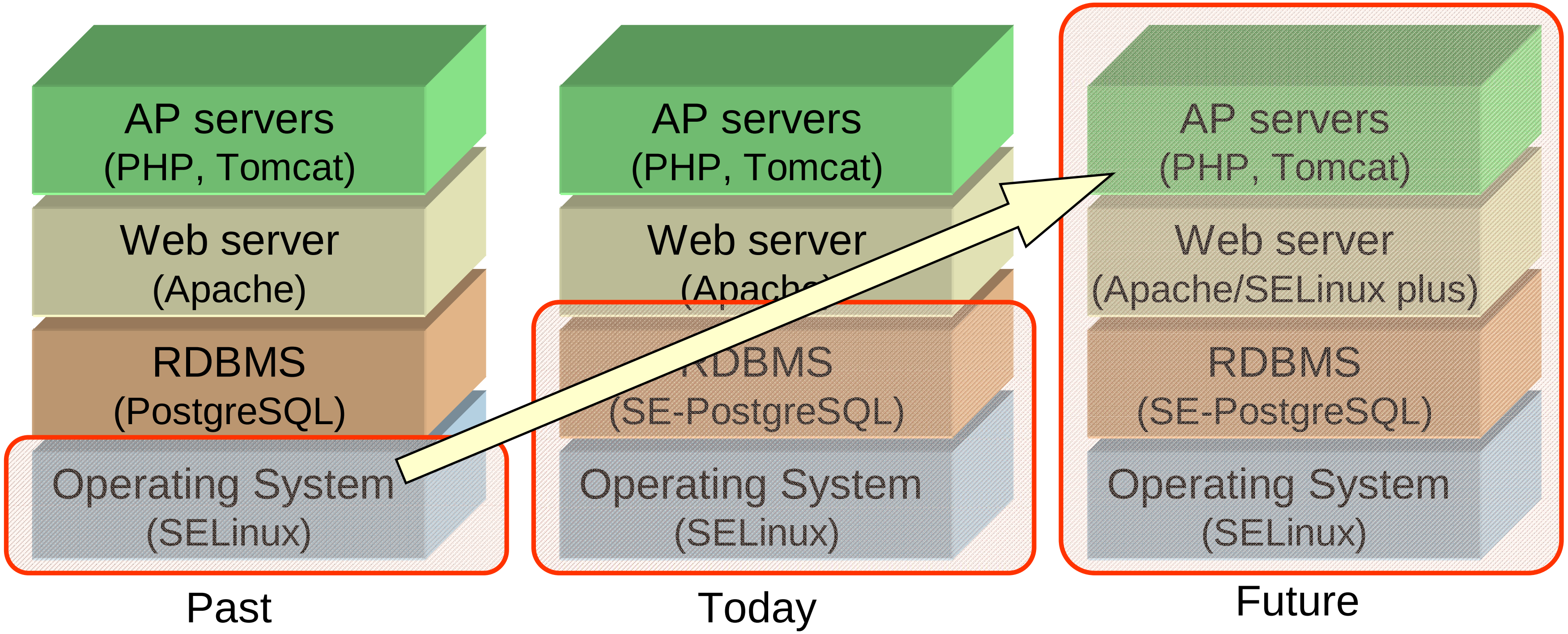
Current status of LAPP/SELinux

- Kernel features
  - 2.6.28 got support bounds-domain and multi-threading.
  - SELinux toolchain also supports bounds-domain.
- SE-PostgreSQL
  - Currently, we are working under PostgreSQL v8.4 development cycle.

    http://wiki.postgresql.org/wiki/CommitFest:2008-11
- Apache/SELinux plus
  - Also published at http://code.google.com/p/sepgsql/
  - Planed to propose it for upstreamed apache/httpd, next to the SE-PostgreSQL.

Future visions

- SELinux as a common foundation of whole of web application stack (LAPP).
  - Consistent privileges and decisions in access control for various kind of web applications.
  - Fine-grained mandatory access control policy



Any questions?

Thank you!