HP Praesidium/ Authorization Server

Topics

- What is It?
- Why Use It?
- Elements: Entitlements, Privileges, Profiles, Users
- Administration
- Architecture
- Web Authorization
- Platform Support
- Future Directions

What is It?

The *HP Praesidium/Authorization Server* is a middleware product providing authorization services to *end-user applications*, using

- a set of *rules* designed by the customer to model the business needs of the enterprise and
- sets of *privileges* defined for *registered users* that specify requirements or limits on each user,

that are

- stored in a *central, replicated database*,
- accessed through *client libraries* and *replicated servers*, and
- managed by a set of *administration tools*.

Why Use It?

- Enforce *consistent policy* across applications
  - Authorization rules and privileges can be set up by security policy administrators.
  - Enforcement of policy is out-sourced to Authorization Server.
- Reduce application *development costs* and *time*
- Application developers don't need to write custom authorization code.
- Reduce *administration costs*
  - Single-point of security administration for multiple applications.
  - Admin interface using ubiquitous web browsers.

Rule Examples: Funds Trading

- A junior trader can initiate a trade transaction for up to $100,000 in fund A and $200,000 in fund B.
- A senior trader can initiate a trade transaction for up to $500,000 in fund A and $1,000,000 in fund B.
- Bob is a junior trader, so he can trade up to $100,000 in fund A and $200,000 in fund B. He also has permission to trade up to $200,000 in fund C.
- Sue is a junior trader, but she is temporarily serving as a senior trader from April 1 through April 5.
- Some trades are executed in currencies other than US$, but the equivalent trade limits apply, based on current exchange rates.

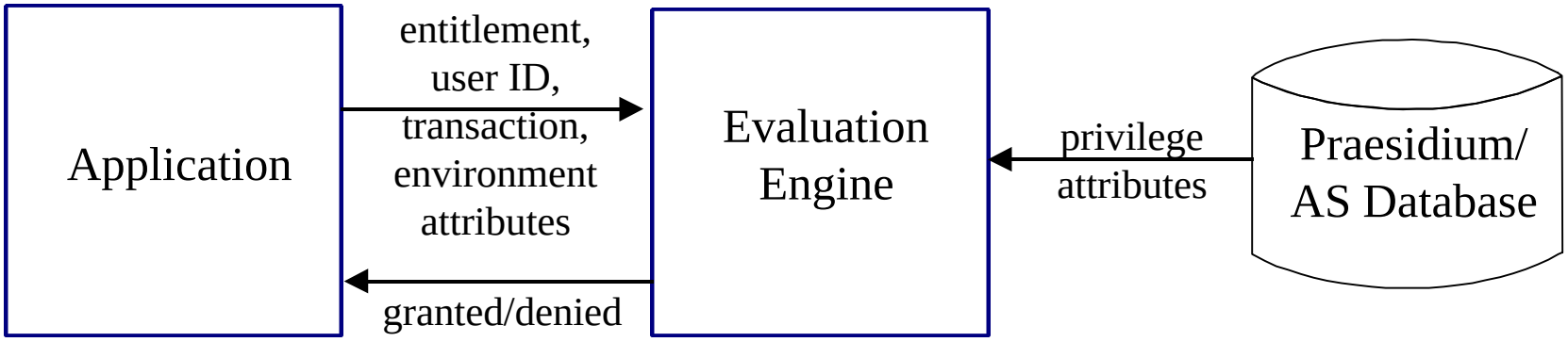
HP Praesidium/
Authorization Ser

Entitlement Rules and Privileges

- *Transaction attributes* are provided by the application.
  - trade_fund, trade_amount
- *Privilege attributes* are defined for users and stored in the Pr/AS databases. Privileges are *enabled* over a time range.
  - For Bob:
    - Funds_Trade:allowed_fund="A", max_amount="100000"
    - Funds_Trade:allowed_fund="B", max_amount="200000"
    - Funds_Trade:allowed_fund="C", max_amount="200000"
- Built-in *environment attributes* provide time information.
  - odss_i_time, odss_i_day, odss_i_date
- A *rule* is a boolean expression that compares transaction, privilege, and environment attributes.
  - trade_fund = allowed_fund and trade_amount <= max_amount

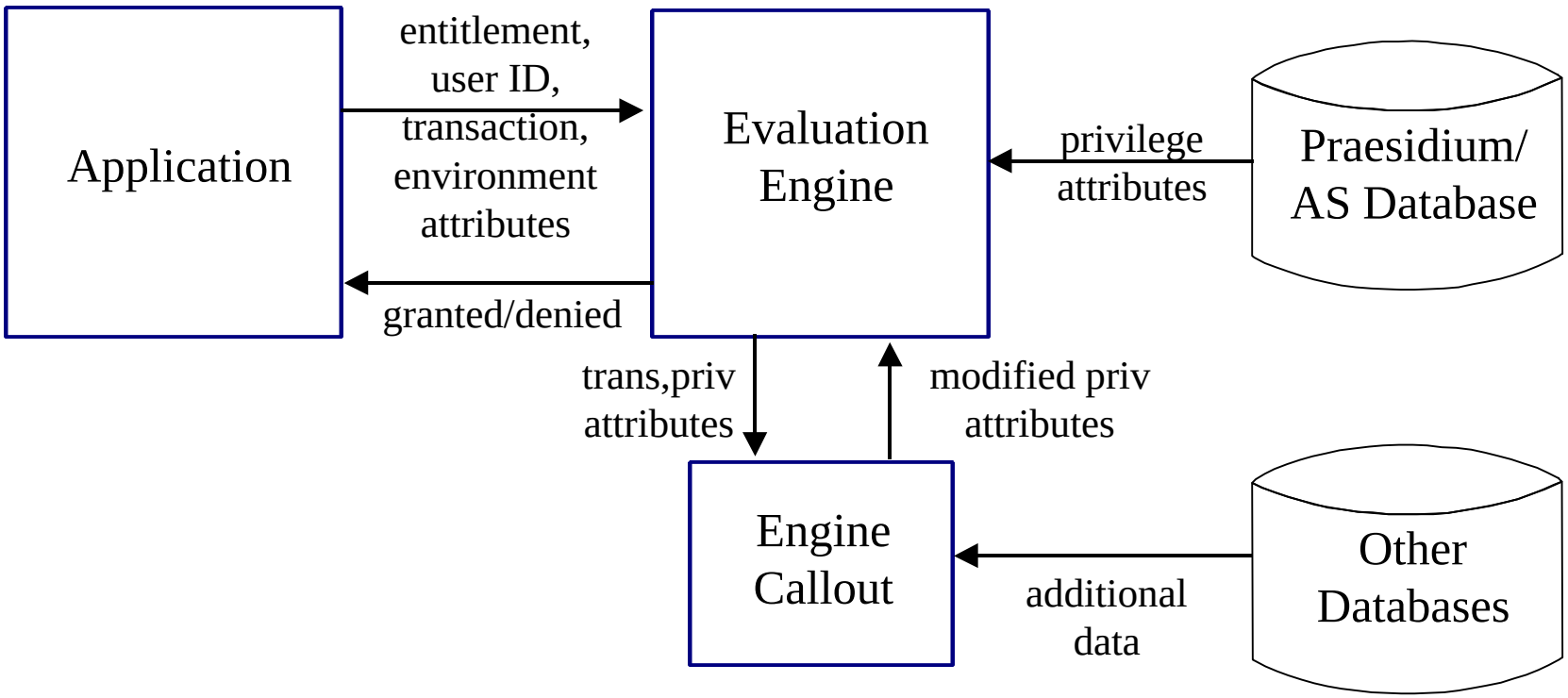
Rule Evaluation

- For each enabled privilege for the user and entitlement:
  - Plug in transaction attribute values from the application.
  - Plug in privilege attribute values from the privilege.
  - Evaluate the rule.
  - If true, authorization is *granted*.
  - If false, continue.
- If all matching privileges have been tried and failed, authorization is *denied*.



Evaluation Callouts

- *Evaluation callouts* provide custom data and processing during rule evaluation.
  - Example: convert trade_amount from foreign currency to US$, using a table of current exchange rates.



Profiles

- A *profile* is a collection of privileges for a class of users.
  - Junior_trader:
    - Funds_Trade:allowed_fund="A", max_amount="100000"
    - Funds_Trade:allowed_fund="B", max_amount="200000"
  - Senior_trader:
    - Funds_Trade:allowed_fund="A", max_amount="500000"
  - Funds_Trade:allowed_fund="B", max_amount="1000000"
- A profile can
  - be assigned to a user, who inherits the privileges in the profile.
  - contain privileges for more than one entitlement.
  - contain other nested profiles.

Registered Users

- A registered user has a *type*, *name* and an *ID*.
  - Type indicates security domain for the name.
    - DCE
    - X.500
    - customer-defined
  - Name is used in admin interfaces.
  - ID is used for privilege retrieval.
- *Create puser* callouts
  - custom mapping of user names to IDs.
  - can retrieve IDs from a database (e.g. X.500 directory).

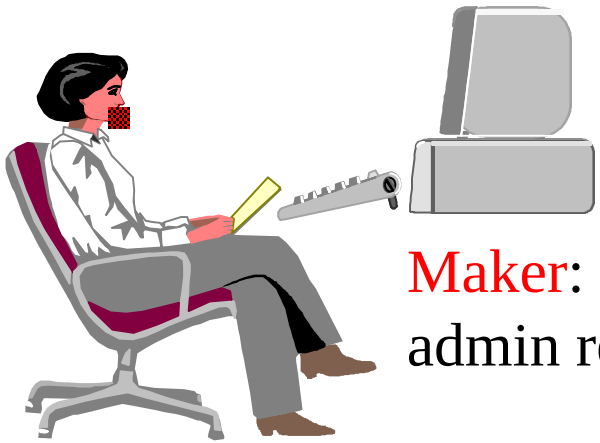
Administrative Tools

- Web Admin GUI
  - uses HTML screens and CGI programs.
  - available through any web browser.
  - requires a Netscape web server in a DCE cell.
- authu_batch
  - provides a command-line interface.
  - must be run in a DCE cell.
  - used to configure base entitlements, profiles, and users.
- authu_maint
  - used to synchronize and backup Pr/AS databases.

Administration Authorization

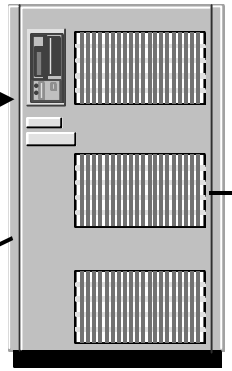
- Praesidium/AS uses pre-defined *base entitlements* to authorize admin requests. Example:
  - ODSS_CREATE_ENT - create an entitlement
- A user must have privileges for these entitlements to execute admin requests.
- The base *ODSS_ADMIN_PROFILE* collects privileges for all of the admin entitlements. This is assigned to admin users.
- The base *cell_admin* principal is initially assigned the ODSS_ADMIN_PROFILE.

Maker/Checker Approval



Maker: submits admin request

Checker: reviews admin request and approves, denies, or resubmits it.



Authorization Server

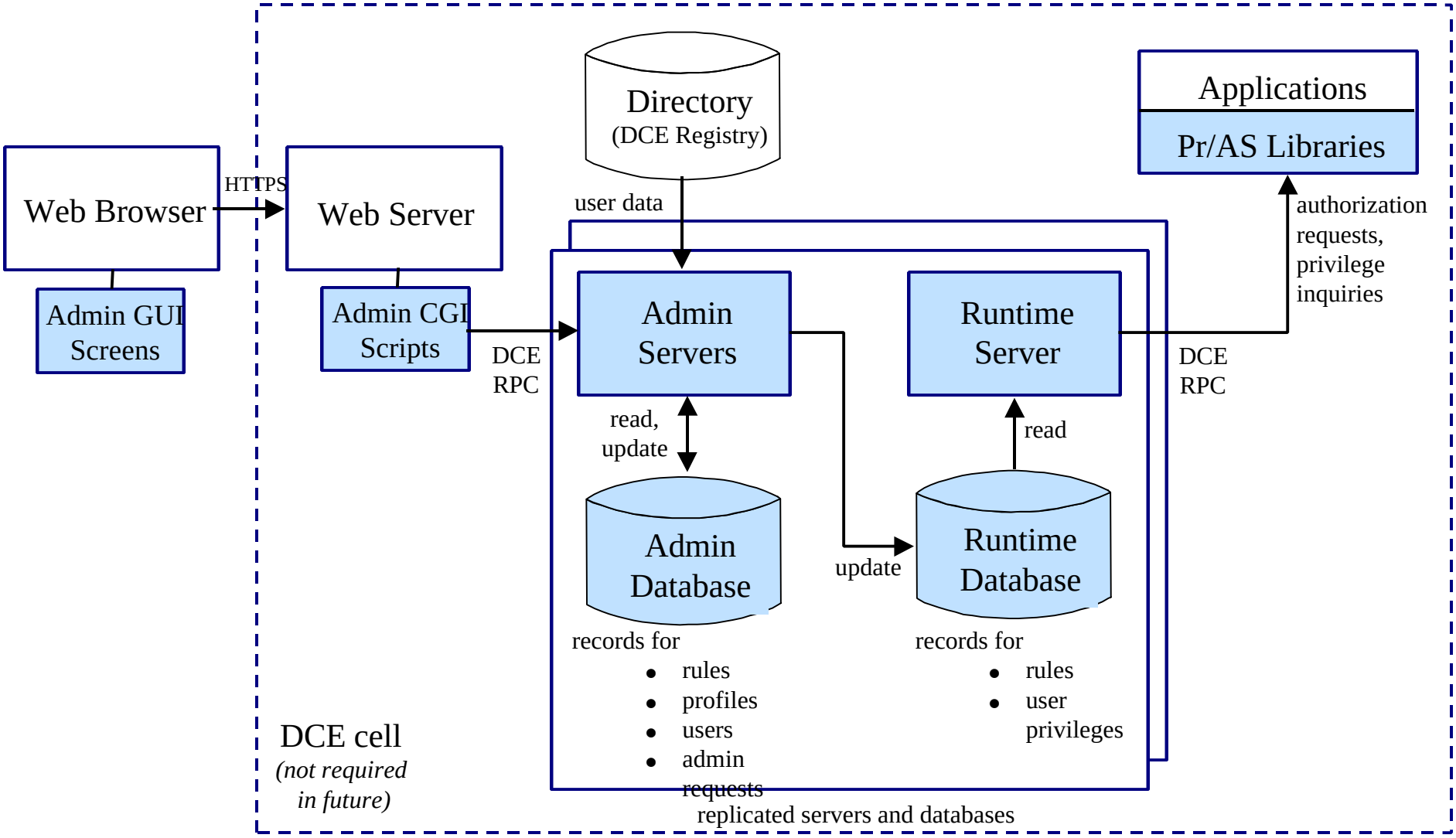
Request Queues

make:
request 1
check:
request 2
pending:
request 3
lock:
request 4
success:
request 5
request 6
fail:
request 7
deny:
request 8
resubmit:
request 9

HP Praesidium/ Authorization Ser

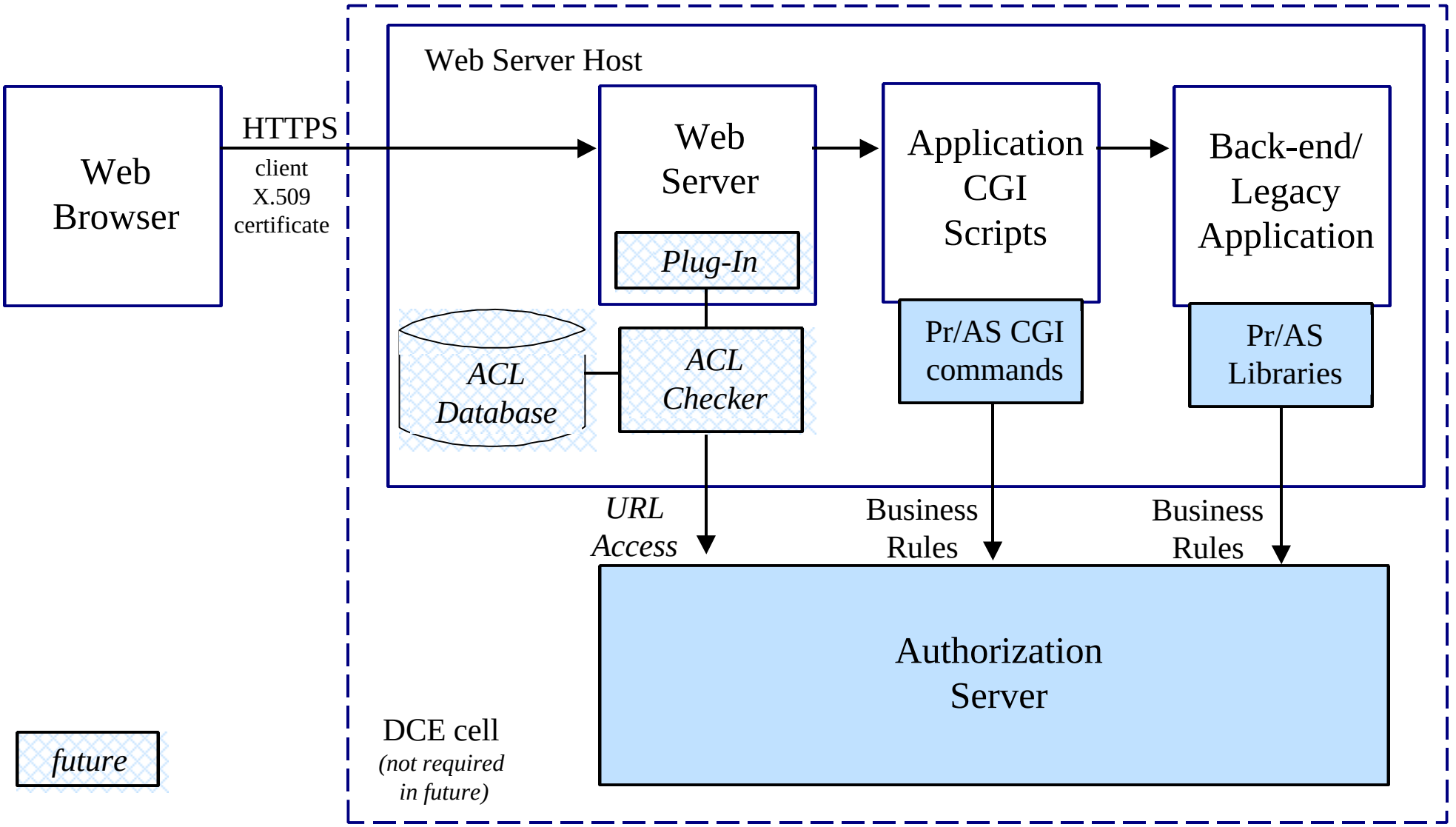
Architecture

HP Praesidium/
Authorization Ser



Web Authorization using Pr/AS

HP Praesidium/
Authorization Ser



HP Praesidium/
Authorization Ser

Platform Support

Platform	Authorization Client	Authorization Server
HP-UX	yes	yes
Windows NT	yes	future
Windows 3.1	privilege inquiry only	
AIX	yes	
Solaris	yes	

Future Directions

- Eliminate DCE dependencies.
  - Use underlying IPsec for secure communications.
  - Support multiple authentication methods for administrators.
    - DCE
    - X.500/SSL
    - Local (NT, Unix) user login
- Add Plug-in Access Control for web authorization.
  - Provide ACLs on web objects.
  - Support Netscape and Microsoft web servers.
  - Provide ACL Editor for easy ACL administration
    - Java applet in web browser
    - Explorer-like format: document tree and object ACLs
  - Escape to full Pr/AS rule authorization for special cases.

Future Directions

- Integrate with other directories and user administration.
  - Netscape Directory Service
  - Microsoft Active Directory (when available)
- Move admin database into directory.
  - Extend directory admin tools for rule and privileges.
  - Allow fine-grain delegation of administration.
- Replace runtime database with in-memory tables.
  - Lower product cost.
  - Improve performance.
- Re-design rule evaluation engine.
  - Improve performance.
  - Allow for embedding engine in applications.