Implied (rules-driven) authorizations - an enhancement to the Roles Database

Jim Repa (repa@mit.edu)

ITAG

May 28, 2008

What do we already have?

- Before discussing new “implied” authorization features, let’s review what we already have in the Roles Database
- MIT’s Roles Database has been in production since 1998, supporting explicitly-granted authorizations for Financials (SAP), Budget, HR, Payroll, Student Systems, EHS, etc.

What we have already - continued

- Central system for maintaining authorizations for multiple target applications
- Generic way of defining authorizations as 3-part “Subject + Verb + Object” structure (Person + Function + Qualifier) with inheritance; easy to maintain and audit
- Authorizations (and Functions) defined in understandable business terminology
- Distributed maintenance of auths. - departmental administrators control access to dept’s resources

What we have already - continued

- UI for maintaining authorizations (PowerBuilder app, web-based interface currently in testing phase)
- Consuming applications can use either…
  - Periodic batch feeds of authorization information
  - Real-time lookups via web service
- Consuming applications enforce the authorizations

Examples of explicit authorizations (old)

- “JOEUSER” can “SPEND OR COMMIT FUNDS” for “cost object 123457”
- “FREDUSER” can “SPEND OR COMMIT FUNDS” for “Funds Center 123456 (all cost objects under the department of Biology)”
- “JSMITH1” can “REPORT ON HR INFORMATION” for “10000322 (EECS)”

Business model for explicit authorizations

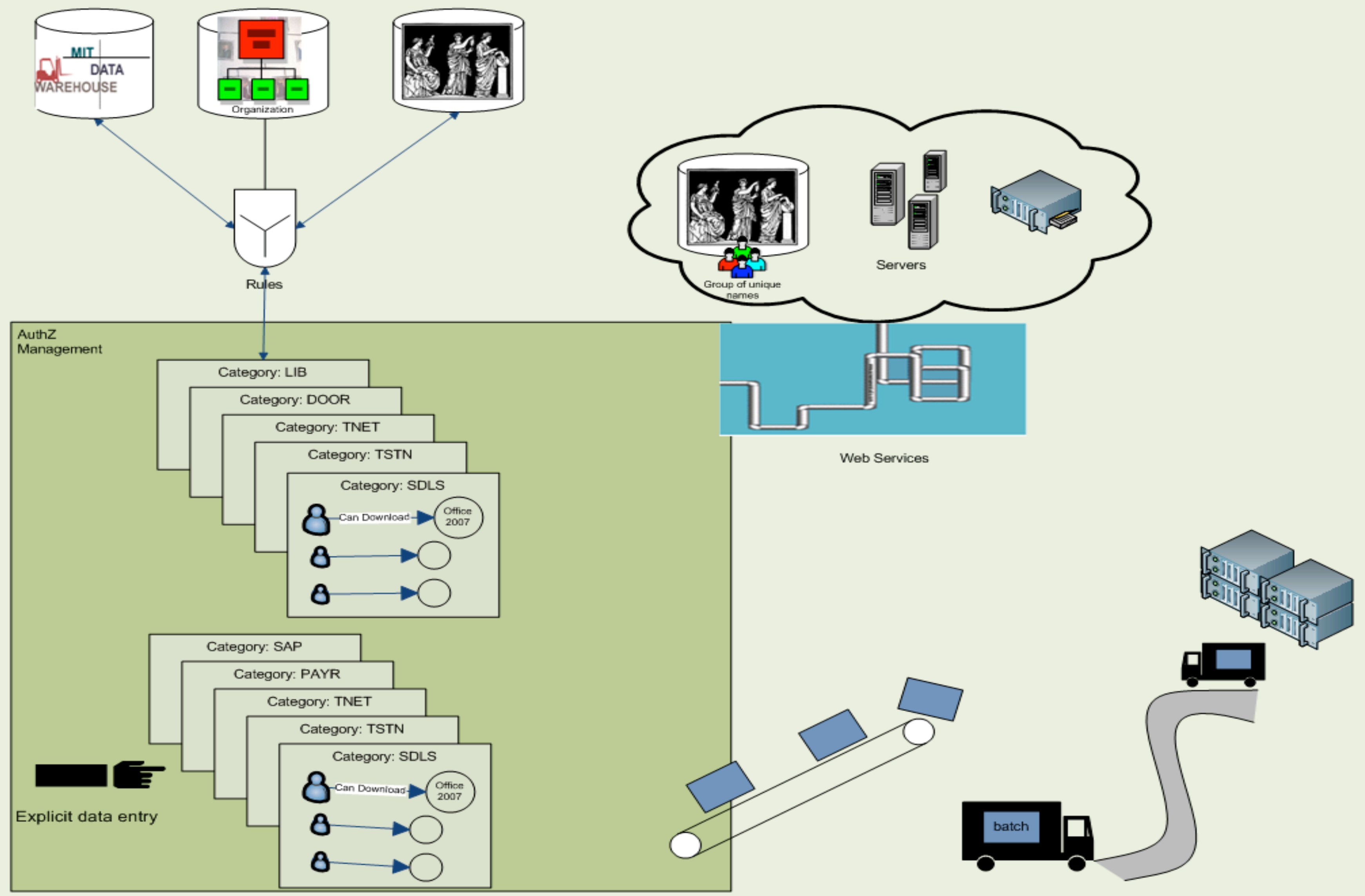
- Department head or central office decides who has control over a category of authorizations
- Designated administrators (called “Primary Authorizers”) decide who needs authorizations and assign authorizations to appropriate individuals
- For MIT’s financial, HR, etc., auths., THERE IS NO WAY TO DERIVE AUTHORIZATIONS from job titles or other known attributes of individuals (too many exceptions, impractical) - thus, we explicitly grant authorizations

However, implied auths are practical for some applications

- There are other areas where we can define unambiguous rules, to derive authorizations from known attributes about people, such as their status as a student/faculty/staff or other known attributes
- Examples
  - Many people at MIT have access to licensed online library materials
  - Large sets of people at MIT have access to licensed downloadable software
  - People with known roles within laboratory areas (tracked by EHS system) have implied access to report or record training or laboratory materials

We will support both explicit and rules-based authorizations

- To accommodate a broader set of applications, we will add support for rules-based implied authorizations
- Web services will allow applications to look-up authorizations for individuals, and will get results based on the union of implied (rules-based) and explicitly-granted authorizations



MIT DATA WAREHOUSE

Organization

Rules

Group of unique names

Servers

Web Services

AuthZ Management

Category: LIB

Category: DOOR

Category: TNET

Category: TSTN

Category: SDLS

Can Download

Office 2007

Category: SAP

Category: PAYR

Category: TNET

Category: TSTN

Category: SDLS

Can Download

Office 2007

Explicit data entry

batch

What features are we adding?

- Authorizations based on evaluation of rules and known data about people
- Enhanced web services that will look up both explicit and implied (rules-based) authorizations
- A UI for maintaining rules

First applications to use implied authorizations

- MIT Libraries (access to licensed online library materials)
- SDLS (distribution of licensed software at MIT)

Later applications to use implied authorizations

- Warehouse reporting on EHS-related training and laboratory information based on known data people’s laboratory roles (PIs, etc.) - use the new service to replace a custom-written program in use within Roles DB
- Card Office control of prox-card access to student housing

How will rules work?

- Build a format that is simple enough to make it easy to implement, but complex enough to support known and anticipated requirements
- Organize input data about people into “relations” (person-verb-object) and base rules on this sort of data
  - Input data example 1: JOE, IS FACULTY, for EECS
  - Input data example 2: SUE, HAS COMPLETED, subject 9.123

Example of rules

If user has this role or attribute…	*…for this object*	*…generate implied auth for this function…*	*…and this qualifier (object or group)*
STUDENT / FACULTY / STAFF	Any DLC at MIT	Access library materials	Group1a of licensed materials
STUDENT	Sloan School	Access library materials	Acme Managemnt journal

More rules examples

If user has this role or attribute…	*…for this object*	*…generate implied auth for this function…*	*…and this qualifier (object or group)*
Principal Investigator	Any DLC/PI object	View training data for postdocs	The same DLC/PI object
Is Resident	Any MIT residence	Has keycard access for front door	The same MIT residence

Web service methods - initial set

- Can a given person perform a given function with a given object (qualifier)?- returns TRUE or FALSE
  - Example: Can user="JOEUSER" do function "ACCESS LIBRARY MATERIALS" for "Encyclopedia Britannica online"?
- Given a person and a function, list objects for which the person has access - returns a list of objects
  - Example: Given user="JOEUSER" and function = "ACCESS LIBRARY MATERIALS", return a list of objects (i.e., library materials that the person can view online)

Web service methods - more

- List or maintain rules
  - Allow an application to build its own rules-maintenance UI that uses central rules and authorizations service
- List or maintain hierarchy of qualifiers (objects) that people can have access to
  - Allow an application to build its own object-maintenance UI that maintains the objects for which people can do certain actions
  - An example might be to build a UI to maintain the list of downloadable licensed software controlled by SDLS

Technology to be used

- Initially, continue to use the same tools as existing Roles Database
  - Oracle database with PL/SQL stored procedures
  - Periodic batch feeds using Perl scripts with Oracle DBI module
  - Web services for real-time access by applications
  - Web-based UI using combination of Perl CGI scripts and Java (AJAX) development
- We are considering building an “open source” version of the Roles DB using mySQL instead of Oracle, with generic scripts intended for broader use than just MIT

More detailed documents

- The following web page has links to web-based documents describing implied authorizations and related topics in more detail
  - http://web.mit.edu/repa/www/implied_auths_links.html