Database Access – Application-Driven versus Data-Driven[1]

William R. Simpson
Institute for Defense Analyses, 4850 Mark Center Dr.
Alexandria, Virginia 22311

[1] The publication of this paper does not indicate endorsement by the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations.

ABSTRACT

A multitude of commercial applications rely on Database Management Systems (DBMS) that provide organized collection of data; for example, modelling the availability of airline flights and seating in a way that supports reservation and sales of air transportation. DBMSs are specially designed software applications that interact with other applications and users to capture and analyze data. A general-purpose DBMS is a software system designed to allow the definition, creation, querying, update, and administration of databases. For the purposes of this paper, we assume that the database is front-ended by web services for database access and query. This paper discusses the current approach to database access and privilege by web services. The paper then discusses the high-assurance paradigm called Enterprise Level Security. We the discuss changes that are required by a high-assurance end-to-end approach. The latter rely on a well-formed security paradigm for the enterprise.

Keywords: Database Access, DBMS, Access Control, IT Security, Integrity.

1. INTRODUCTION

Database security deals with all aspects of protecting the database content, its users, and its owners. It covers protection from intentional and unintentional unauthorized database activities by authorized privilege limited entities and unauthorized entities (e.g., a person or a computer program).

Database access control deals with controlling who (a person or a computer program) is allowed to access what information in the database and what privilege is provided. The information may comprise specific database objects (e.g., record types, specific records, data structures), certain computations over certain objects (e.g., query types, or specific queries), or use of specific access paths to the former (e.g., using specific indexes or other data structures to access information). [1-9]

This may be managed directly on an individual basis, or by the assignment of individuals and privileges to roles that are then granted entitlements.

Data security prevents unauthorized users from viewing or updating the database. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data.

Data security in general deals with protecting specific data, from corruption, destruction, or removal.

Our basic security model requires that all functionality be realized by web services. This precludes database grazing, in which the requester can peruse most of the database at once. This is to be preceded by public key infrastructure (PKI)-based mutual authentication and a transport layer security (TLS) pipeline followed by a security assertion markup language (SAML) token for access and privilege (as described in section 2). The database is organized by columns and each identity or role has permission that allow Create, Read, Update, or Delete (CRUD) functions. This paper presents the database issues in several parts.

Part 1 is this introduction.

Part 2 presents the high-assurance background on Enterprise Level Security (ELS). It includes a number of the basic concepts.

Part 3 presents the overall considerations for database operations at the enterprise level.

Part 4 intoduces the Enterprise Resource Planning (ERP) a form of database operations software that is centered around business applications.

Part 5 reviews the processes for security-hardening databases in general and specifically the ERP, including:

a. Encryption of data at rest.
b. Encryption of data in transit.
c. Claims, Access and Privilege.
d. Application Least Privilege and two paradigms for database operations (application-driven and data-driven).
e. Partisal homomorphic encryption operations.

Part 6 presents a summary.

2. ENTERPRISE LEVEL SECURITY

ELS is a high-assurance environment. For ELS, we are primarily concerned with four security principles.

- Know the Players – this is done by enforcing bi-lateral end-to-end authentication.

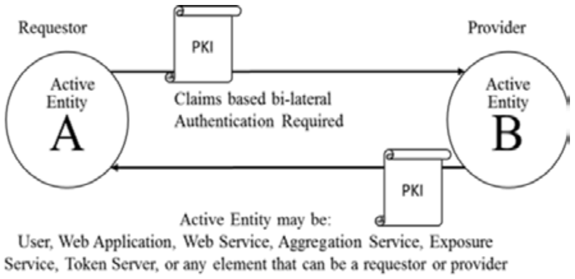

Requestor
PKI
Provider
Active Entity
A
Claims based bi-lateral Authentication Required
Active Entity
B
PKI
Active Entity may be:
User, Web Application, Web Service, Aggregation Service, Exposure Service, Token Server, or any element that can be a requestor or provider

Figure 1 Bi-lateral Authentication

In ELS, the identity certificate is an X.509 PKI certificate [10]. PKI certificates are verified and validated. Ownership is verified by a holder-of-key check.

- Maintaining Confidentiality – this entails end-to-end unbroken encryption.

ELS establishes end-to-end TLS [11] encryption (never give away private keys that belong uniquely to the certificate holder). Message authentication codes are enforced (but they are only valid when the encryption remains unbroken to the end point).

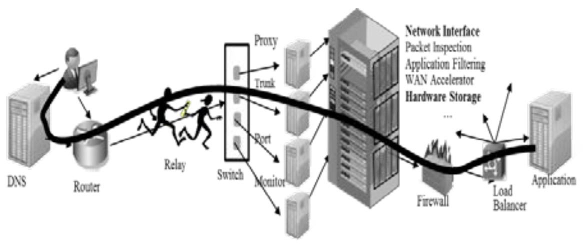


Figure 2 End-to-End Encryption

- Enforce Access Control – this is done by an authorization credential.

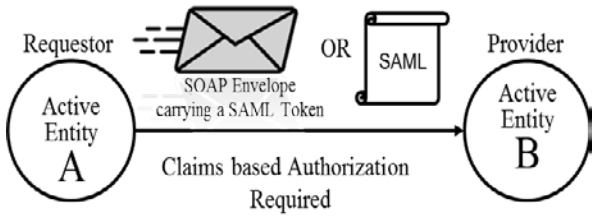

Requestor
OR
SAML
Provider
SOAP Envelope carrying a SAML Token
Active Entity A
Active Entity B
Claims based Authorization Required

Figure 3 Claims-Based Authorization

In ELS the certificate is the Security Assertion Markup Language (SAML) [12]. SAMLs are signed, and the signatures are verified and validated. The credentials of the signers are verified and validated.

- Maintain Integrity – know that you received exactly what was sent – know that content has not been modified.

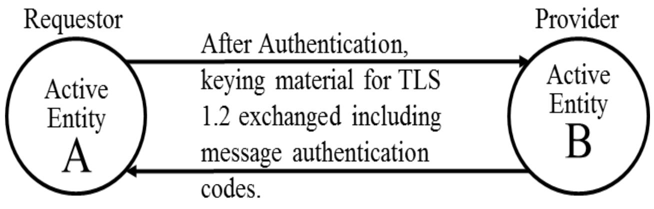

Requestor
Provider
After Authentication, keying material for TLS 1.2 exchanged including message authentication codes.
Active Entity A
Active Entity B

Figure 4 MAC and Other Integrity Measures

In ELS this is implemented by End-to-End TLS encryption with message authentication codes (MAC). Packages (like SAML tokens) are signed, and signatures are verified and validated [13].

3. OVERALL DATABASE CONSIDERATIONS

Both SQL and XML database systems were designed for precise processing of data sets. Formal rules define how operations are conducted, and well-defined outputs are produced for any given input state. This works well for traditional data models, where the requester wants the correct results for a given query at that particular time. However, these database systems only scale to certain sizes, based on their inherent architectural structures. For example, parallelism is difficult to provide while maintaining all the guarantees of SQL or XML databases. This makes it difficult to process very large data sets with reasonable performance.

Due to the important role database systems serve in the enterprise, they are protected from failures. This includes redundancy of hardware and software instances. The data itself is stored on multiple different hardware instances. These are part of different logical and software database system instances. The hardware redundancy protects against disk failures. The logical redundancy protects against software bugs or failures. Another type of redundancy related to physical redundancy is geographic distribution. Since some failures, such as electricity or natural disasters, are regional, there is at least one instance of the data being stored in a different region. This could be, for example, different military bases in in the eastern and western United States, or in North America and Europe.

The important concept is to meet Continuity of Operations (COOP) requirements for continuous operations in the event of failure of the main operational system. In addition to redundancy, extra copies are generated, maintained, synchronized, and available in standby mode so that the switch can be made automatically to promote the backup to the live operational system. The backup is updated by transaction with the primary instance to ensure consistency. When the primary data is destroyed or corrupted, some method of determining which transactions—if any—are rolled back when invoking backup data is provided. There might be a time lag between the primary and backup data sources. Ideally, the transaction would involve the backup copies so that when a transaction is complete it is also backed up; but if performance limits this capability, a method to maintain consistency when promoting the backup to primary is provided.

Serializability is a property of a database such that there is a valid sequence of events that could have resulted in the database state for all externally visible states of the database. In particular, intermediate internal states involved in transactions are not accessible to external entities. This ensures that actions are taken in the correct order. In some cases, the database manager reorders requests for performance or other reasons. However, this can result in inconsistencies. For example, if one application queries the value of two cells and another application increments their values, the query of the first cell might get the updated value while the second cell gets the old value, which could violate assumptions about the underlying content. Different guarantees on serializability are possible in database systems. Stricter guarantees limit optimizations, and looser guarantees allow more optimization.

Triggers are actions taken within a database system based on changes that take place to its data. These are useful for maintaining internal consistency. Instead of writing complex applications to scan the data after each operation to check consistency and take appropriate actions, triggers can be used to automatically do this whenever a change is made to a database.

Indexes allow faster querying. They provide a different way to organize data, one that allows queries to be processed faster than if searching an entire table. Instead of using a search that looks through the table entries to find values that match a query, the index is used to directly point to the rows in the table that have that value. One implementation involves a binary tree to search for the value in logarithmic time instead of linear time.

Data cubes and data grids are a way to use a database to store aggregate information across multiple categories. For example, if a database keeps track of sales, with attributes for location, date, and type, a data cube would include automatically generated entries that sum across all locations, all dates, all types, any combination of two categories, and a grand total across all data. This makes aggregating queries fast, since it involves a single lookup instead of aggregating across large numbers of entries. Other variations of cubes allow different dimensionality (e.g., two-dimensional) or aggregation functions (e.g., average or max), but the same concepts apply.

Checkpointing is a method of periodically saving the database state. The challenge is to do this in a consistent way while updates and queries are being run on the database. Saving the entire state of a database is a time-consuming activity, so shutting down other activity while this is done is not an acceptable solution for performance reasons. The goal of checkpointing is to save the current state in such a way that as far as all requesters can tell the state was saved at a given point in time.

Logging captures the time and nature of all requests. When combined with checkpointing, logging allows reconstruction of a "broken" database. The last checkpointed version can be loaded, and then the history in the log files can be repeated starting at the time of the checkpoint. This requires checkpointing information to be included in the log files.

Databases can allow or deny actions to different users based on authorization rules. In ELS systems, services are the front end of all databases, so these rules are not needed for individual web browser requesters. However, for an exposure service that reads only from a database, it would be beneficial to allow read-only permission for that service, so database authorization rules are important for ELS systems even though the database authorization is not strictly part of the ELS security model.

Although the underlying data structures are not always made available at the database service layer of abstraction, certain common patterns should be made available or used, such as B-Trees and hash tables. A B-Tree provides a balanced tree structure to optimize query performance, and a hash table provides a constant time lookup for arbitrary data sets.

Recent news reports are replete with instances of database compromises, theft of data, and other incursions. To understand the vulnerabilities, we should first examine how database applications are currently organized.

4. ENTERPRISE RESOURCE PLANNING

Access to a database is normally organized along the lines of a business software solution. Enterprise resource planning (ERP) is business management software—typically a suite of integrated applications—that a company can use to collect, store, manage, and interpret data from many business activities, including product planning, cost, manufacturing, service delivery marketing, and sales. The form is complex and involves self-sufficiency on everything from security to external interfaces. The basic form is shown in the Figure 5.

The complexities associated with ERP are due, in part, to control of security and other aspects. However, since the ERP does not use web service approaches (end-to-end approaches with distributed authentication, and authorization), they must be treated as legacy and untrusted from the high-assurance standpoint.

ERP as a Legacy System
There are two aspects of interfacing with a legacy system. The first is control of the attributes associated with an identity, and the second is the sanitization and checking of communications and data that an untrusted system provides. Figure 6 shows a somewhat simplified version of the ERP with the security interfaces in place. In the figure, external interfaces have been removed since they will not be discussed.

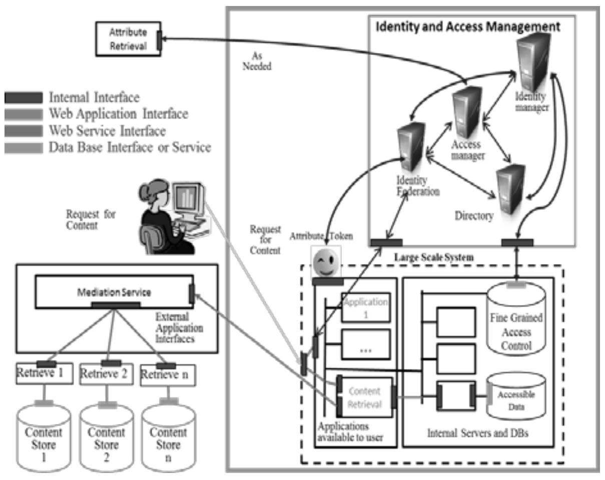


Figure 5 ERP Overall Organization

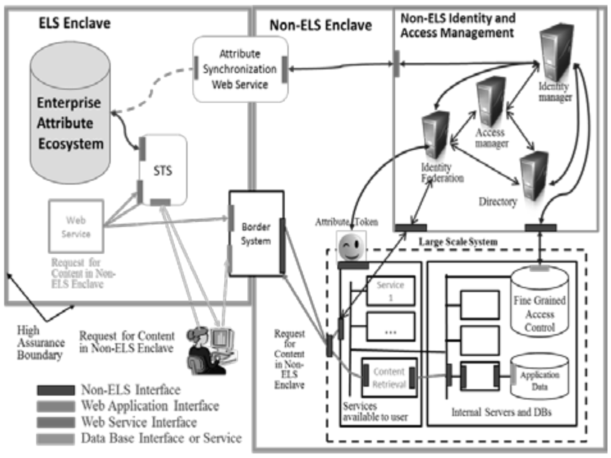

ELS Enclave
Non-ELS Enclave
Non-ELS Identity and Access Management
Attribute Synchronization Web Service
Enterprise Attribute Ecosystem
STS
Web Service
Request for Content in Non-ELS Enclave
Border System
High Assurance Boundary
Request for Content in Non-ELS Enclave
Identity manager
Access manager
Identity Federation
Directory
Attribute Token
Large Scale System
Request for Content in Non-ELS Enclave
Service 1
...
Content Retrieval
Services available to user
Fine Grained Access Control
Application Data
Internal Servers and DBs
Non-ELS Interface
Web Application Interface
Web Service Interface
Data Base Interface or Service

Figure 6 ERP as a Legacy System

ERP Attribute System Synchronization
Large-scale legacy systems (such as the ERP) maintain their own attributes for users and establish roles and privileges based upon those attributes. Current updating may or may not happen based upon manual input. Notifications of changes within the enterprise are not uniform (and often not in a form easily used by legacy system administrators), and updating may or may not occur. This leads to an unacceptable delay in adjusting to changes within the enterprise. Attribute synchronization between the legacy system and the Enterprise Attribute System (EAS) may be undertaken to allow a more automated adjustment to the tempo of operations. This requires a synchronization system and changes to the legacy system for compatibility in definitions of roles, etc.

The synchronization system described above must meet the following considerations:

- Not all EAS entities will be in legacy databases; however, all legacy identities should be in the EAS.
- Each Identity in a legacy database has roles and privileges based upon attributes.
- Each identity in the legacy has enterprise-level attributes.
- Maps must be established between the EAS attributes and the legacy roles and privileges.
- Local naming may differ from that for Enterprise Distinguished Name (DN), but it can be mapped to the enterprise identity.
- Some EAS attributes currently map to legacy roles and privileges.

- The enterprise must add and maintain other attributes in the EAS that are needed to establish legacy roles and privileges.
- For Identities that occur in both, EAS contains attributes (for each identity) that can map to legacy roles and privileges.
- EAS will be maintained and reflect the current attributes of all its active entities, including those attributes added for legacy mappings.
- The legacy (Access Manager Interface) will import (periodically or on demand) changes to attributes of identities in its stores as reflected by changes in EAS identity attributes.
- The legacy will remap its roles and privileges based upon updated attributes.
- The legacy roles and privileges will be synchronized and current with EAS attributes after each update mapping.

ERP Border System

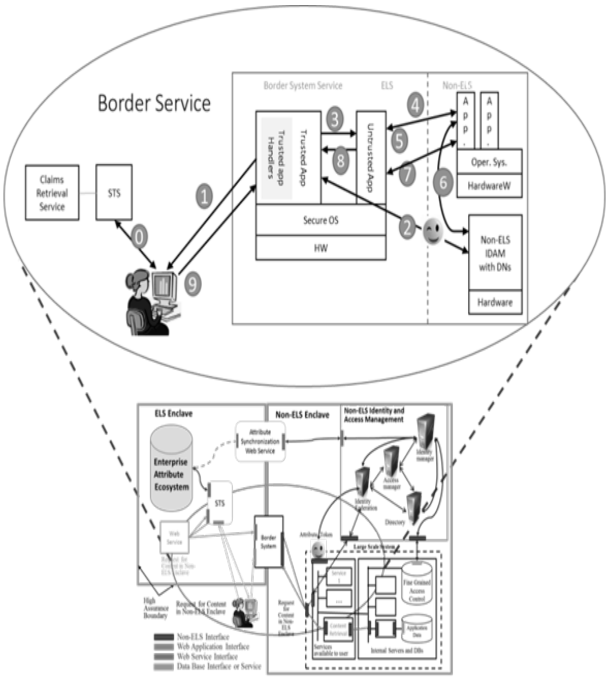


Figure 7 Legacy Border System Functionality

The extent to which the border system must translate/mediate communications depends on the Non-ELS system. The border system is in the ELS enclave unless the particular Non-ELS enclave opts to provide some of the functionality in its environment, which is preferable because PKI enabled Non-ELS enclaves should synchronize access and privilege attributes with the EAS to simplify the process. However, the main assumption is that Non-ELS systems do not change. Access and privilege resides with the Non-ELS enclave Identity and Access Management (IDAM). For Figure 7, the communications flows are:

0. Standard ELS flows for browser SAML request to STS, with call to EAS for claims. Browser window info includes the STS name, Application Name, Border service name, and content object name.
1. The browser calls Border System Service (BSS) using standard ELS Authentication and Authorization.
2. The trusted application within BSS creates a NEW trusted application for a specific user call, NEW trusted application calls Non-ELS identity system with its own credentials [if the identity system is DN-enabled and PKI-enabled or else has an account with the identity system and an authentication credential recognized by the trust system] with request for local identity and authentication credentials based on DN of the original requester.
3. The trusted application fork/execs untrusted application and passes local identity authentication credential and application name/content object name to untrusted application via local IP.
4. The untrusted application establishes a connection to non-ELS Application using local ID and authentication credentials.
5. The untrusted application sends a request to non-ELS Application of content object name.
6. The non-ELS Application sends a request to identity system with local ID of Untrusted Application for authorization credentials.
7. The non-ELS Application determines access, retrieves content from store, and sends response(s) to untrusted Application.
8. The untrusted application passes a response to the trusted application (at session close, the untrusted application terminates) special handling, including sanitization—offline.
9. The new trusted application sends the content to the browser (on session close the NEW trusted application terminates and clears state). Note that these can also be threads of a single process.

5. HARDENING ERP DATABASE SYSTEMS

ERPs contain a number of applications (for specific functionality), and a back-end that will provide attributes related to identities. These applications are normally identity-based and often username/password-enabled. The sources of attributes may contain local stores that need to be maintained for the specific database, and of course, all of the interfaces need to be secured. To discuss the hardening of the ERP, we have further simplified the diagram, as shown in Figure 8.

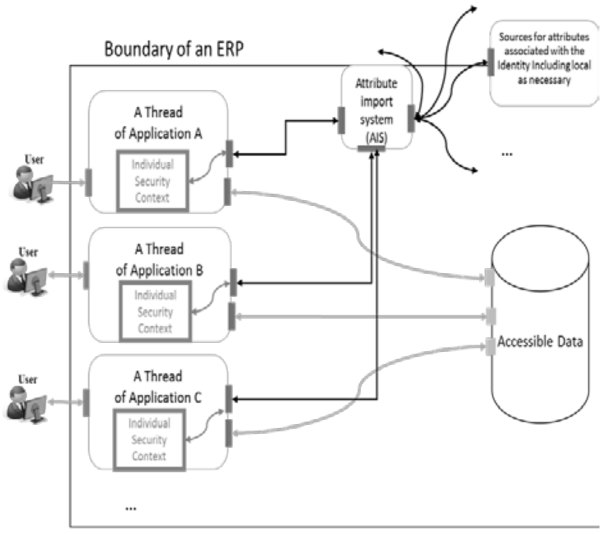
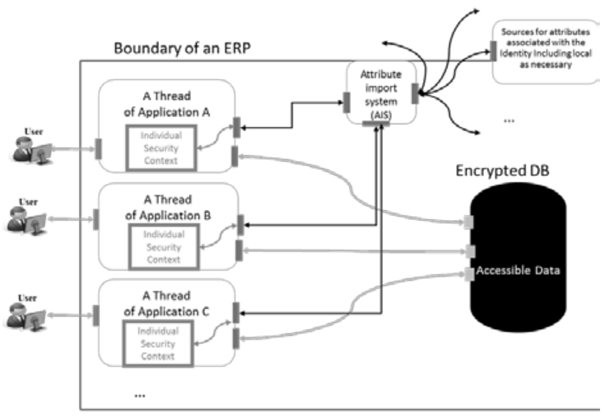

Boundary of an ERP
Sources for attributes associated with the Identity Including local as necessary
Attribute import system (AIS)
...
User
A Thread of Application A
Individual Security Context
User
A Thread of Application B
Individual Security Context
User
A Thread of Application C
Individual Security Context
Accessible Data
...

Figure 8 The Simplified Organization of an ERP

Hardening will occur in five stages, discussed below:
1. Encryption of data at rest;
2. Encryption of data in transit;
3. Claims-based identity, access, and privilege;
4. Hardening the Application for Accessing Databases
5. Applying homomorphic encryption processes to the database.

Hardening Stage One – Encryption of Data at Rest
Figure 9 shows the basic idea. All data within the database is encrypted as it sits in storage. This stops any threats that are present from benefitting from picking up the entire stored

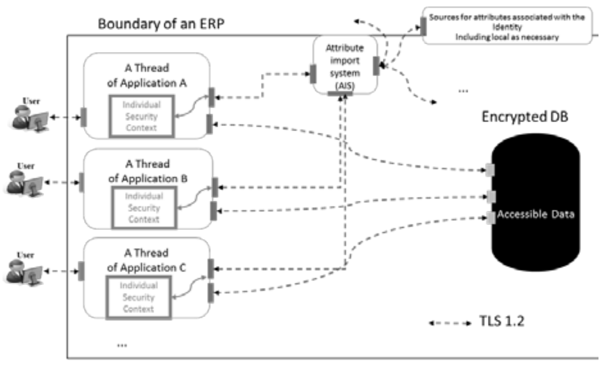
database and exporting it. Many database systems already do this. It should be automatic and built into the storage system.



Boundary of an ERP
Sources for attributes associated with the Identity Including local as necessary
Attribute import system (AIS)
A Thread of Application A
A Thread of Application B
A Thread of Application C
Individual Security Context
User
Encrypted DB
Accessible Data

Figure 9 Encryption of Data at Rest

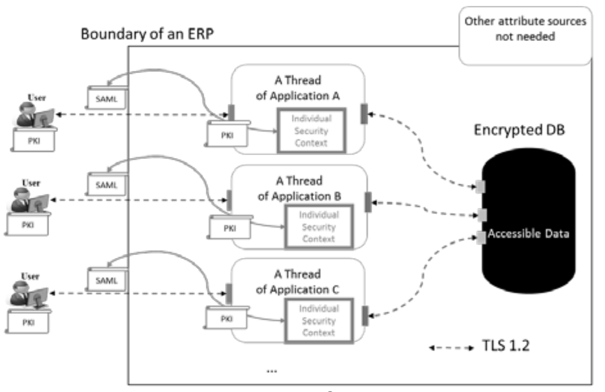
Hardening Stage Two – Encryption of Data in Transit
Although this may seem obvious, many of the implementations have not encrypted all of the links. Links within the spaces considered to be the ERP are often considered trusted, but as we have seen again and again, such spaces may contain threats, at least for a while. The preferred process for enterprise-level security is TLA-based bi-lateral authentication.



Boundary of an ERP
Sources for attributes associated with the Identity Including local as necessary
Attribute import system (AIS)
A Thread of Application A
A Thread of Application B
A Thread of Application C
Individual Security Context
User
Encrypted DB
Accessible Data
TLS 1.2

Figure 10 Encryption of Data in Transit

Hardening Stage Three – Claims Identity, Access, and Privilege
This stage is more difficult, but much more rewarding. Figure 11 below shows that a number of interfaces are completely eliminated and their vulnerabilities are no longer a problem. Many of these old software interfaces went outside of the ERP system. The claims-based system will be assumed from this point forward. The hardening does not stop here, however, and the fourth stage is discussed in the next few sections.



Other attribute sources not needed
Boundary of an ERP
User
SAML
PKI
A Thread of Application A
A Thread of Application B
A Thread of Application C
Individual Security Context
Encrypted DB
Accessible Data
TLS 1.2

Figure 11 Implementation of Claims-based Identity, Access, and Privilege

Hardening Stage Four – Least Privilege for Application
We will proceed from the position that the databases are claims-driven as opposed to account-driven. Most database systems maintain accounts for users. These accounts have privileges and status for each individual, are expensive to maintain, and are error-prone. With ELS we have two basic credentials, the PKI for identity and SAML for access and privilege. The basic security model indicates that all functionality is realized by web services. This precludes database grazing, which is a situation in which the requester can peruse most of the database at once. This is to be preceded by PKI-based mutual authentication and a TLS pipeline and followed by a SAML token for access and privilege. However, that still leaves two paradigms for database operations (application-driven and data-driven). To illustrate the difference, the following section contains a notional example whereby a financial database is accessed by an individual who has the credentials of a Financial Analyst.

Role-Based Access
The enterprise financial database (EFD) has many predefined roles. These are determined by the data owner, and placed in the format of an Access Control requirement (ACR) for storage in the enterprise service registry. The roles may be arbitrarily complex since the claims engine will compute whether or not they are satisfied and provide any variables or restrictions requested. A few are defined below:

1. Financial analyst is determined by position, training, and job identifier.
   Financial Analyst =>
   a. manager and above, AND
   b. job identifier=xxx12, AND
   c. training=[basic finance (within last 5 years) AND financial Analysis (within last 5 years)] OR [BS, accounting or finance (within last 10 years)] OR waiver.
   RESTRICT
   a. sub area q unless supervisor is corporate director or above.
   b. data restricted to current location code. AND
   c. cannot update any project over $5M UNLESS a waiver is issued for the individual AND
   d. Additional restrictions may be included.
2. Financial Supervisor is determined by position, training, and job identifier.
   Financial Supervisor =>
   a. manager and above, AND
   b. job identifier=xxx14, AND
   c. training=[basic finance (within last 5 years) AND Financial Analysis (within last 5 years)] OR [BS, Accounting or finance (within last 10 years)] OR waiver is issued for the individual.
   RESTRICT
   a. cannot update any project over $5M until he has been using the system 6 months, OR
   b. waiver is issued for the individual.
3. Financial Auditor is determined by training and job identifier.
   Financial Auditor =>
   a. job identifier=xx316, AND
   b. training=[basic finance (within last 5 years) AND Financial Analysis (within last 3 years) AND Financial Audit (within last 3 years)) OR (MS, Accounting or finance (within last 15 years)) OR waiver.

RESTRICT
   a. data restricted to audit location code.
   b. …
4. Bookkeeper …
5. Quality Control Specialist …
6. Administrator …
7 ….

Application-Driven Access
Each of the roles must be coded for operations. For illustration we will deal only with the Financial Analyst in this example—who, in this case, is Fred2345432, or just Fred. The evolution is the normal preparation for the access and privilege associated with an application or service. The figures below show the evolution of the access control, which involve most of the services in the enterprise attribute ecosystem.

The process begins with the generation of a SAML token.

Table 1 Basic SAML for Database Operations

SAML: Assertion		
Version ID	Version 2.0	Required
ID	SAML ID	Required
Issue Instant	Timestamp	Required
Issuer	(content)	Required
Signature	(content)	Required
Subject	User	Required - X.509 Identity
SAML: Attribute Statement		
Subject	User	For local use
Claims include Roles: and restrictions	(content)	may include parameters.
SAML: Conditions		
NotBefore	(content)	Timestamp
NotAfter	(content)	Timestamp
Audience	(content)	Target Service

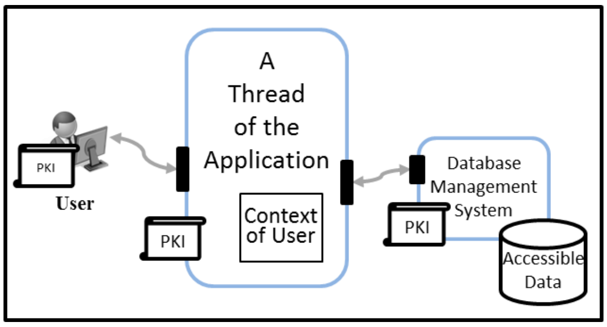


Figure 12 Application-Driven Access

The application (through the use of SAML) has the security context of the user. The application has full privilege with the database and is trusted to limit the user to his/her security context.

Application-Driven Annotated Example
Fred is the Chicago Branch Manager. The definitions of the various roles can be used to compute Fred's claims. Fred is evaluated based upon the enterprise data and he is provided a claim of Financial Analyst but with some restrictions as shown in the table below.

Table 2 Basic Data Evaluation for Fred

Financial Analyst =>	Fred	Claims Engine for Fred
manager and above, AND	Chicago Branch Manger	True
job identifier = xxx12, AND	Job code =43212	True
training=[basic finance (within last 5 years) AND (Financial Analysis (within last 5 years)] OR [BS, Accounting or finance (within last 10 years)] OR waiver.)	Training = training on basic AND finance(8/4/2012), AND (Financial analysis (6/5/2010), BS Mathematics Purdue (6/1/2000),) OR On the enterprise Training Waiver list group for Financial Analysts (TWFIN)	True AND True AND (False or True) =True Overall Fred = Financial Analyst
RESTRICTIONS		
sub area q unless supervisor is corporate director or above.	supervisor (all billets report to 43200 or 43201) is Field Office Manager	False Supply Notq token to application
Data restricted to location code. AND	Location Code = Chicago	Supply Chicago token to application
Cannot update any project over $5M UNLESS a waiver is present in the enterprise stores	Not in enterprise group for ($5Mupdatewaiver)	False supply Not$5M+ token to application

Table 3 SAML for Fred (Application-Driven)

SAML: Assertion	
Version ID	Version 2.0
ID	X34.?thik045ml23
Issue Instant	12:11:00 06 May 2014
Issuer	www.securitytokenserver3.net
Signature	(content)
Subject	Fred- X.509 Identity
SAML: Attribute Statement	
Subject	Fred2345432
Claims: Role = Financial Analyst Data= Notq, Chicago, Not$5M+	
SAML: Conditions	
NotBefore	12:11:00 06 May 2014
NotAfter	12:16:00 06 May 2014
Audience	www.mysqldata2.net

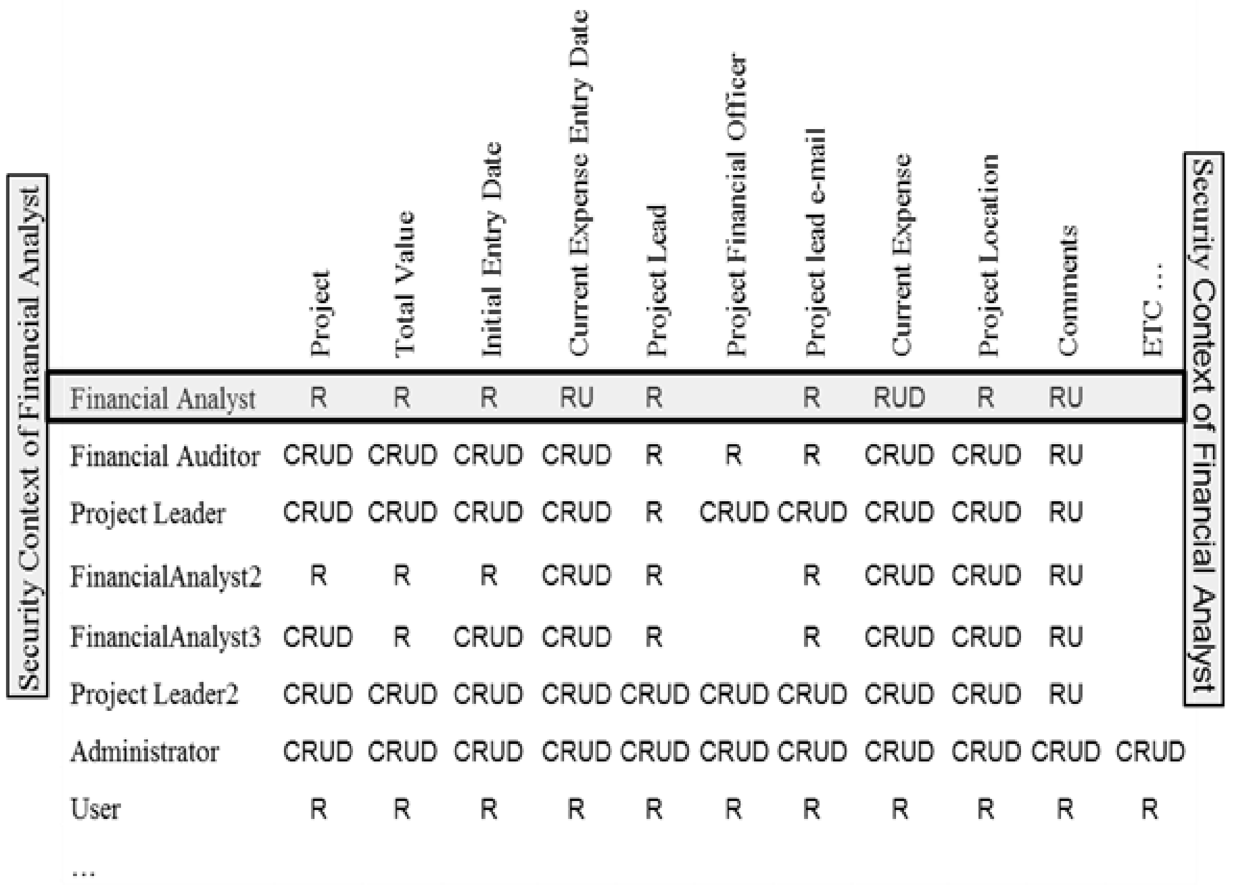
For databases, the application-driven approach has the following advantages and disadvantages:

Advantages:
1. The data owner does not have to know the database schema in order to specify access and privilege.
2. The service controls Fred's interaction with the database.
3. Database administrators may or may not establish CRUDs for the role in question.

Disadvantages:
1. The service developer must know the database specifics.
2. The service is granted full access to the database (to accommodate the different users.
3. The service computes what is allowable (CRUD) and send computed SQL for what it believes are reasonable requests consistent with Fred's authorities.

Data-Driven Operations
A number of additional requirements are needed for data-driven applications:

1. Database schema must be known to the developer of the access control requirements. Assume column authorization defined CRUDs
2. Elements in the database (when they represent the same thing in the enterprise attribute store (EAS) must be identical (and common definition) to the elements in the EAS. (Example: Location code in the database is a three-character code. It must be the same code in the EAS—when multiple databases use the same value, they must all have the same representation as the EAS).
3. The database must be prepared: The column CRUD permissions are set in the database for each role (Figure 13).
4. CRUD by role:


Security Context of Financial Analyst
Security Context of Financial Analyst

	Project	Total Value	Initial Entry Date	Current Expense Entry Date	Project Lead	Project Financial Officer	Project lead e-mail	Current Expense	Project Location	Comments	ETC ...
Financial Analyst	R	R	R	RU	R		R	RUD	R	RU	
Financial Auditor	CRUD	CRUD	CRUD	CRUD	R	R	R	CRUD	CRUD	RU	
Project Leader	CRUD	CRUD	CRUD	CRUD	R	CRUD	CRUD	CRUD	CRUD	RU	
FinancialAnalyst2	R	R	R	CRUD	R		R	CRUD	CRUD	RU	
FinancialAnalyst3	CRUD	R	CRUD	CRUD	R		R	CRUD	CRUD	RU	
Project Leader2	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	RU	
Administrator	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD
User	R	R	R	R	R	R	R	R	R	R	R
...											

Figure 13 CRUD by Role

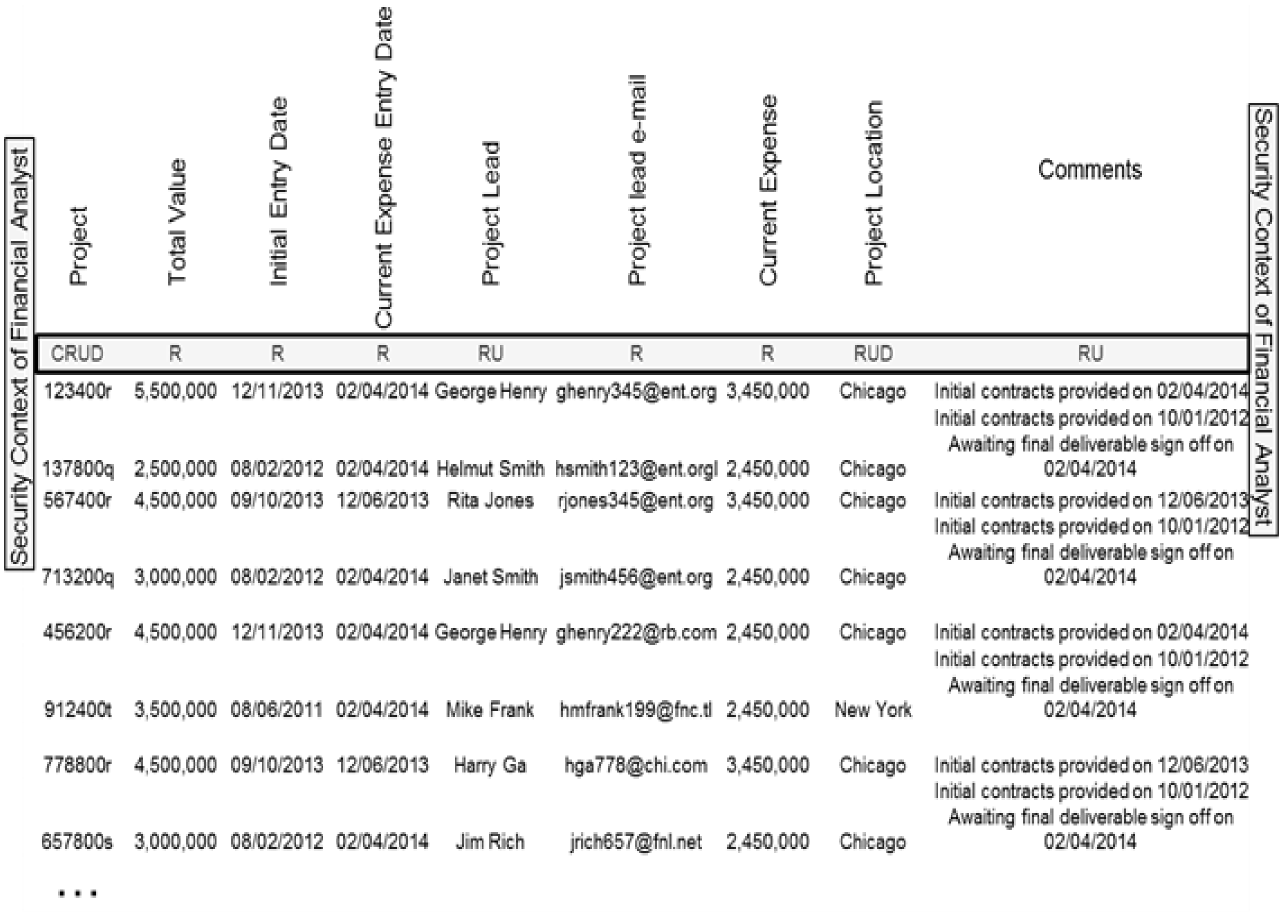
a. A view template is created (Figure 14) by Role showing all columns that a role can view.
b. A stored program is created that will provide a tailored view for each role as tailored by the individual attributes in the CRUD security of the role. [14, 15].
c. This view can only be restricted, not enhanced. If enhancement is desired, a new role must be defined.

5. View restrictions are by column but apply to rows (Example: Project Location = ‘Chicago’).
6. When more than one role is in SAML, the application must ask the requester which role is being exercised.
7. We assume for this example a column-organized relational database. The claims can be built for any database and the former is for illustration only. For the database, the permissions are defined in terms of CRUD, normally by columns. The database also applies these CRUD elements for the role. In an identity-based access control system, they would be written for each identity. The use of roles and restrictions simplifies the definitions for an appropriate view to be computed.

Transfer of the SAML to a stored program in the database to set the view for the role as limited by other factors. For example, Columns and their CRUDSs are set in the stored view for each role. Rows are restricted by setting acceptable values in various columns. The stored program validates the SAML, resolves the rol, and sets the view in the security context of the role (for the application of CRUDs to be transferred to the application for further transmittal to the user). The application must have at least four SQL queries programmed in. These include:

1. Execute stored program for view and security context.
2. Create – New entry in the stored view and security context.
3. Update – (column, row) in the stored view and security context are updated.
4. Delete – (column, row) in the stored view and security context are deleted.

Any violation of the CRUD for the context view returns an error.


Security Context of Financial Analyst

Project	Total Value	Initial Entry Date	Current Expense Entry Date	Project Lead	Project lead e-mail	Current Expense	Project Location	Comments
CRUD	R	R	R	RU	R	R	RUD	RU
123400r	5,500,000	12/11/2013	02/04/2014	George Henry	ghenry345@ent.org	3,450,000	Chicago	Initial contracts provided on 02/04/2014
137800q	2,500,000	08/02/2012	02/04/2014	Helmut Smith	hsmith123@ent.orgl	2,450,000	Chicago	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
567400r	4,500,000	09/10/2013	12/06/2013	Rita Jones	rjones345@ent.org	3,450,000	Chicago	Initial contracts provided on 12/06/2013
713200q	3,000,000	08/02/2012	02/04/2014	Janet Smith	jsmith456@ent.org	2,450,000	Chicago	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
456200r	4,500,000	12/11/2013	02/04/2014	George Henry	ghenry222@rb.com	2,450,000	Chicago	Initial contracts provided on 02/04/2014
912400t	3,500,000	08/06/2011	02/04/2014	Mike Frank	hmfrank199@fnc.tl	2,450,000	New York	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
778800r	4,500,000	09/10/2013	12/06/2013	Harry Ga	hga778@chi.com	3,450,000	Chicago	Initial contracts provided on 12/06/2013
657800s	3,000,000	08/02/2012	02/04/2014	Jim Rich	jrich657@fnl.net	2,450,000	Chicago	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
• • •								

Security Context of Financial Analyst

Figure 14 View Template for Financial Analyst

Data-Driven Annotated Example

Fred is evaluated by the claims engine, and claims are slightly modified based upon the database schema and the instructions to the stored program as shown in the table below.

Table 4 Modified SAML Data for Fred (Data-Driven)

SAML: Assertion	
Version ID	Version 2.0
ID	X34.?thik045ml23
Issue Instant	12:11:00 06 May 2014
Issuer	www.securitytokenserver3.net
Signature	(content)
Subject	Fred - X.509 Identity
Subject	Fred2345432
Claims: **Role = Financial Analyst** **Restrict: "Project" ≠ ??????q** **Restrict: "Project Location"= "Chicago"** **Restrict: "Total Value" >=5,000,000** (content)	
SAML: Conditions	
NotBefore	12:11:00 06 May 2014
NotAfter	12:16:00 06 May 2014
Audience	www.mysqldata2.net

The application authenticates itself to the database and triggers the stored program—the SAML for Fred is transferred as shown in Figure 15.

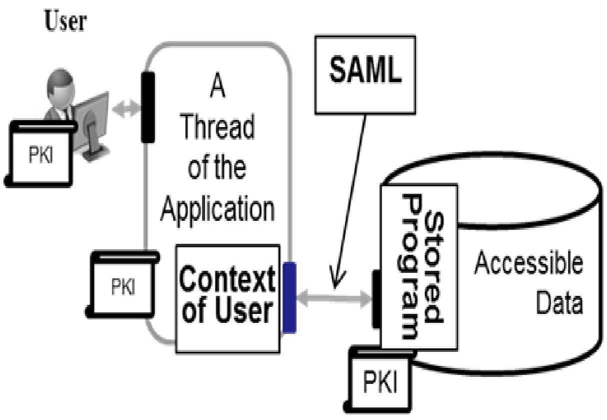
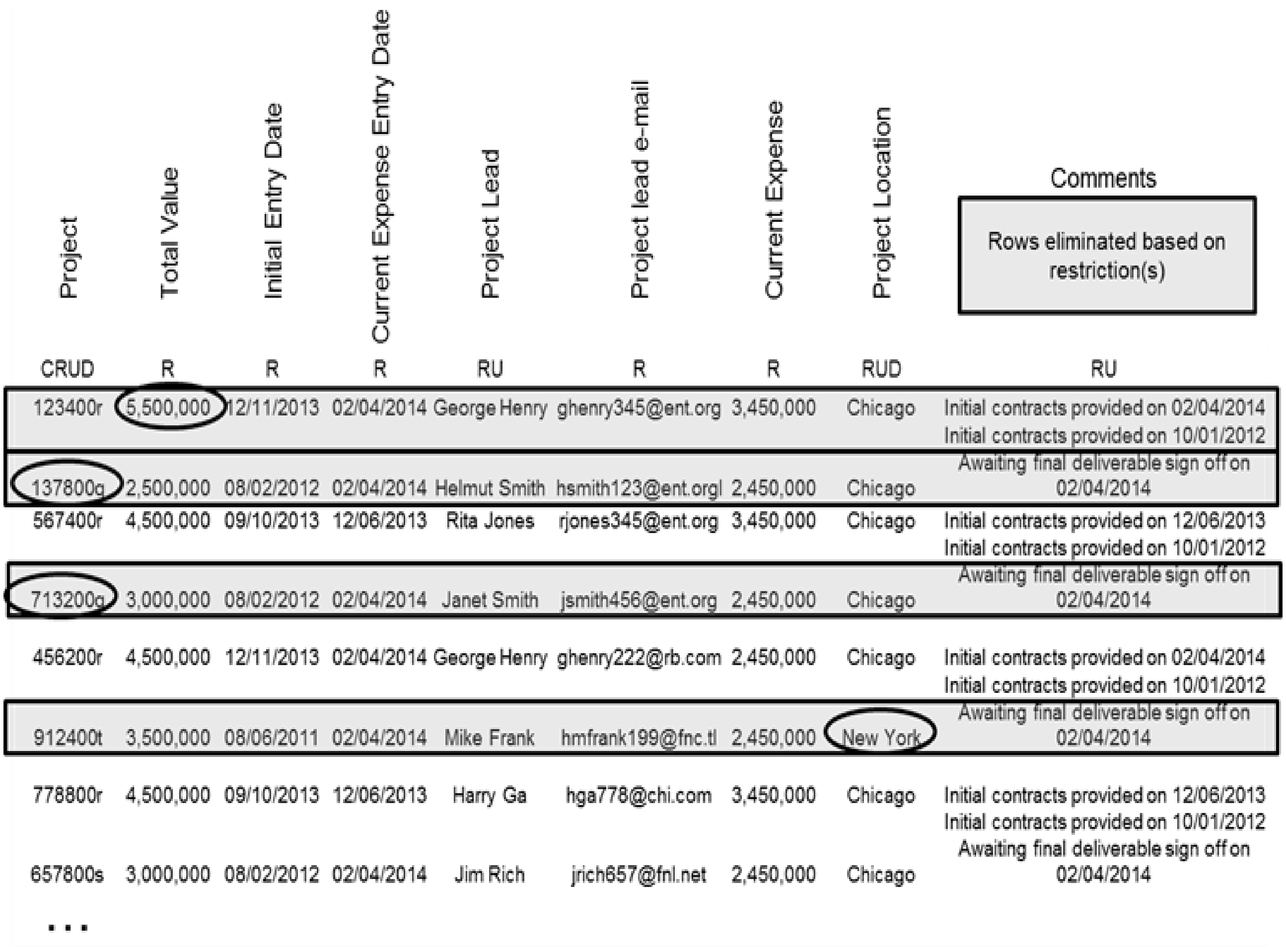


Figure 15 Posting SAML to Stored Program

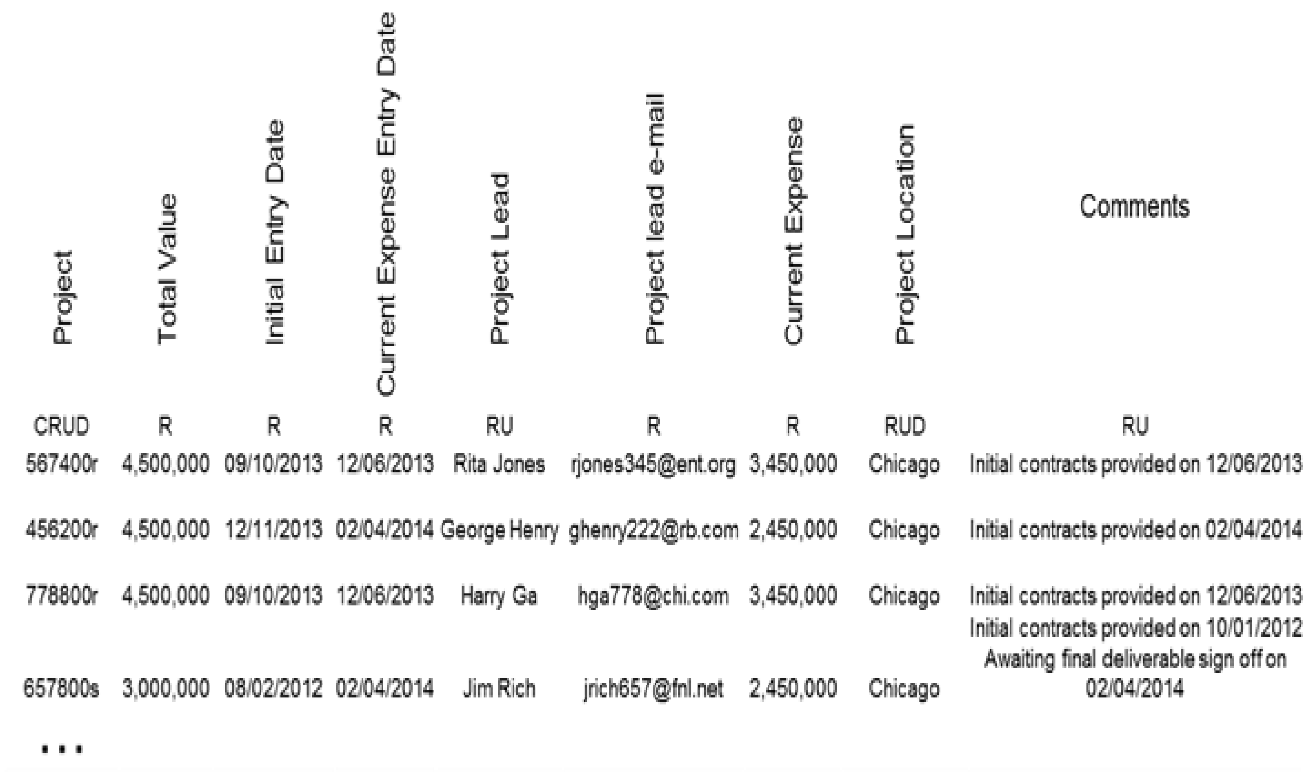
The stored program verifies and validates the SAML and pulls up the view template stored in the permissions for Financial Analyst. The stored program then applies the restrictions to the view. This restricted or tailored view is provided to the application for action. Actions are performed, but only in the context of the CRUDs in the tailored view. The view is then updated for further work. The stored program modifies the view as shown in Figures 16 and 17.



Rows eliminated based on restriction(s)

Project	Total Value	Initial Entry Date	Current Expense Entry Date	Project Lead	Project lead e-mail	Current Expense	Project Location	Comments
CRUD	R	R	R	RU	R	R	RUD	RU
123400r	5,500,000	12/11/2013	02/04/2014	George Henry	ghenry345@ent.org	3,450,000	Chicago	Initial contracts provided on 02/04/2014
137800g	2,500,000	08/02/2012	02/04/2014	Helmut Smith	hsmith123@ent.orgl	2,450,000	Chicago	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
567400r	4,500,000	09/10/2013	12/06/2013	Rita Jones	rjones345@ent.org	3,450,000	Chicago	Initial contracts provided on 12/06/2013
713200g	3,000,000	08/02/2012	02/04/2014	Janet Smith	jsmith456@ent.org	2,450,000	Chicago	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
456200r	4,500,000	12/11/2013	02/04/2014	George Henry	ghenry222@rb.com	2,450,000	Chicago	Initial contracts provided on 02/04/2014
912400t	3,500,000	08/06/2011	02/04/2014	Mike Frank	hmfrank199@fnc.tl	2,450,000	New York	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
778800r	4,500,000	09/10/2013	12/06/2013	Harry Ga	hga778@chi.com	3,450,000	Chicago	Initial contracts provided on 12/06/2013
657800s	3,000,000	08/02/2012	02/04/2014	Jim Rich	jrich657@fnl.net	2,450,000	Chicago	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
. . .								

Figure 16 Tailoring the View for Data-Driven Access and Privilege



Project	Total Value	Initial Entry Date	Current Expense Entry Date	Project Lead	Project lead e-mail	Current Expense	Project Location	Comments
CRUD	R	R	R	RU	R	R	RUD	RU
567400r	4,500,000	09/10/2013	12/06/2013	Rita Jones	rjones345@ent.org	3,450,000	Chicago	Initial contracts provided on 12/06/2013
456200r	4,500,000	12/11/2013	02/04/2014	George Henry	ghenry222@rb.com	2,450,000	Chicago	Initial contracts provided on 02/04/2014
778800r	4,500,000	09/10/2013	12/06/2013	Harry Ga	hga778@chi.com	3,450,000	Chicago	Initial contracts provided on 12/06/2013
657800s	3,000,000	08/02/2012	02/04/2014	Jim Rich	jrich657@fnl.net	2,450,000	Chicago	Initial contracts provided on 10/01/2012 Awaiting final deliverable sign off on 02/04/2014
. . .								

Figure 17 Tailored View for Financial Analyst Fred

The CRUDS in the database will be enforced for the restricted view. Figure 18 shows the exchange with the user. Only accessible data leaves the database.

The A in figure 18 is:

A scripted exchange with the application about a user request related to the tailored view (requests are not filtered).

The B in figure 18 is:

SQL Requests (Read is assumed in the view):

1. Create – New entry—stored view and security context
2. Update – (column, row)—stored view and security context
3. Delete – (column, row)—stored view and security context

No other SQL requests are allowed.

For databases, the data-driven approach has the following advantages and disadvantages:

Advantages:

1. The service has limited access to the database.
2. The database controls Fred’s interaction with the database based upon Fred’s credentials.
3. Database administrators must establish CRUDs for the role in question.
4. The SQL authority of the service is limited and verified by the database.

Disadvantages:

1. The data owner does have to know the database schema in order to specify access and privilege.
2. Views are moved multiple times.
3. The service computes what is allowable (CRUD) and sends computed SQL for what it believes are reasonable requests.

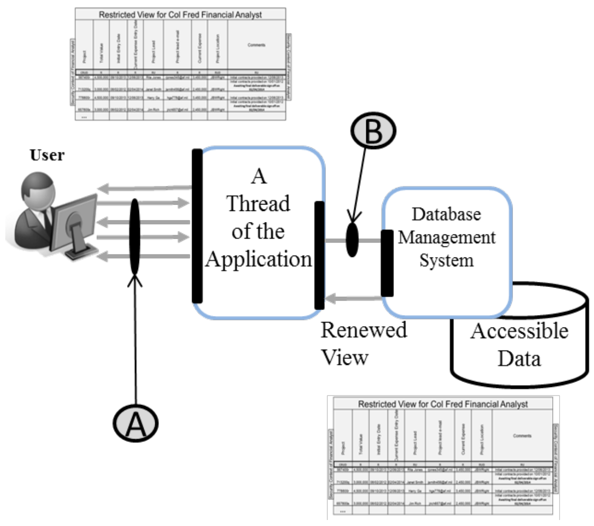

Restricted View for Col Fred Financial Analyst
B
User
A
Thread
of the
Application
Database
Management
System
Renewed
View
Accessible
Data
A
Restricted View for Col Fred Financial Analyst

Figure 18 Data-Driven Exchange with User

Hardening Stage Five – Homomorphic Encryption

Homomorphic techniques allow encrypted queries of encrypted data. The techniques have been proven in the lab, but are currently impractical. Of current concern is partial homomorphic encryption of databases, which may be practically applied to Sequential Query Language (SQL) database queries. [16-20].

6. SUMMARY

We have reviewed the basic approaches to the restriction of database access, and the assignment of privilege with databases. The common approach to a web service front end of a Database Management System (DBMS) requires the web service to restrict access and privilege based upon the user context. In doing this it must be provided with full access and privilege to the database, and be trusted to limit user access and privilege. We reviewed the high-assurance security paradigm and the changes that must be made for hardening the security associated with database operations. The suggested approaches build increasing security by adding user-tailored restrictions directly into the database, and they provide the web service fronting the DBMS with the same privilege as the user. At the same time, it restricts SQL queries to a fundamental set that will be enforced by the view developed within the database and not at the web service. A final area, yet to be developed is the application of partial homomorphic techniques that keeps all transactions encrypted.

This research is part of a body of work for high-assurance enterprise computing using web services. Elements of this work include bi-lateral end-to-end authentication using PKI credentials for all person and non-person entities, a separate SAML credential for claims-based authorization, full encryption at the transport layer, and a defined federation process. Many of the elements of this work are described in [21-26].

REFERENCES

[1]. Jeffrey Ullman 1997: First course in database systems, Prentice–Hall Inc., Simon & Schuster, p. 1, ISBN 0-13-861337-0.

[2]. Tsitchizris, D. C. and F. H. Lochovsky (1982). Data Models. Englewood-Cliffs, Prentice–Hall.

[3]. Beynon-Davies P. (2004). Database Systems 3rd Edition. Palgrave, Basingstoke, UK. ISBN 1-4039-1601-2.

[4]. Ken North, “Sets, Data Models and Data Independence,” Dr. Dobb’s, 10 March 2010.

[5]. William Hershey and Carol Easthope, “A set theoretic data structure and retrieval language,” Spring Joint Computer Conference, May 1972 in ACM SIGIR Forum, Volume 7, Issue 4 (December 1972), pp. 45–55, DOI=10.1145/1095495.1095500.

[6]. Description of a set-theoretic data structure, D. L. Childs, 1968, Technical Report 3 of the CONCOMP (Research in Conversational Use of Computers) Project, University of Michigan, Ann Arbor, Michigan, USA.

[7]. Feasibility of a Set-Theoretic Data Structure: A General Structure Based on a Reconstituted Definition of Relation, D. L. Childs, 1968, Technical Report 6 of the CONCOMP (Research in Conversational Use of Computers) Project, University of Michigan, Ann Arbor, Michigan, USA.

[8]. “TeleCommunication Systems Signs up as a Reseller of TimesTen; Mobile Operators and Carriers Gain Real-Time Platform for Location-Based Services.” Business Wire. 2002-06-24.

[9]. “Structured Query Language (SQL).” International Business Machines. October 27, 2006. Retrieved 2007-06-10.

[10]. Internet Engineering Task Force (IETF), RFC 2459, Internet X.509 Public Key Infrastructure, January 1999.

[11]. Request for Comments: The Transport Layer Security (TLS) Protocol Version 1.2. http://tools.ietf.org/html/rfc5246, August 2008

[12]. P. Mishra et al. Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005.

[13]. William List and Rob Melville, IFIP Working Group 11.5, Integrity In Information, Computers and Security,

Volume 13, Issue 4, pp. 295–301, Elsevier, doi:10.1016/0167-4048(94)90018-3, 1994.

[14]. My SQL stored Programs and Views. http://docs.oracle.com/cd/E19078-01/mysql/mysql-refman-5.0/stored-programs-views.html#stored-routines-syntax

[15]. Purdue on Using stored procedures to set views https://www.cs.purdue.edu/homes/ninghui/projects/Topics/DB_FineGrained.html

[16]. Boneh, D., Gentry, C., Halevi, S., Wang, F., Wu, D.J.: Private database queries using somewhat homomorphic encryption. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 102–118. Springer, Heidelberg (2013), Practical Packing Method in Somewhat Homomorphic Encryption 49.

[17]. Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., Koshiba, T.: Packed homomorphic encryption based on ideal lattices and its application to biometrics. In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds.) CD-ARES Workshops 2013.

[18]. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999).

[19]. Raluca A. Popa, Catherine M.S. Redfield, Nickolai Zeldovich, Hari Balakrishnan "CryptDB: Processing Queries on an Encrypted Database," Comm. ACM, vol. 55, no 9, Sept. 2012 (also Proc. of 23rd ACM SoSP, Sept. 2011).

[20]. R. A. Popa, F. H. Li, N. Zeldovich, "An Ideal-Security Protocol for Order-Preserving Encoding," Proc. of the 2013 IEEE Symposium on Security and Privacy, San Francisco, CA, May 2013. Coimbatore Chandersekaran and William R. Simpson, The 3rd International Multi-Conf. on Engineering and Technological Innovation: IMETI2010, Volume 2, "A SAML Framework for Delegation, Attribution and Least Privilege," pp. 303–308, Orlando, FL, July 2010.

[21]. William R. Simpson and Coimbatore Chandersekaran, The 3rd International Multi-Conference on Engineering and Technological Innovation: IMETI2010, Volume 2, "Use Case Based Access Control," pp. 297–302, Orlando, FL., July 2010.

[22]. William R. Simpson and Coimbatore Chandersekaran, International Journal of Computer Technology and Application (IJCTA), "An Agent-Based Web-Services Monitoring System," Vol. 2, No. 9, September 2011, pp. 675–685.

[23]. William R. Simpson, Coimbatore Chandersekaran and Ryan Wagner, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering and Computer Science 2011, Volume I, "High Assurance Challenges for Cloud Computing," pp. 61–66, San Francisco, October 2011.

[24]. Coimbatore Chandersekaran and William R. Simpson, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering 2012, The 2012 International Conference of Information Security and Internet Engineering, Volume I, "Claims-Based Enterprise-Wide Access Control," pp. 524–529, London, July 2012.

[25]. William R. Simpson and Coimbatore Chandersekaran, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering 2012, The 2012 International Conference of Information Security and Internet Engineering, Volume I, "Assured Content Delivery in the Enterprise," pp. 555–560, London, July 2012.

[26]. Coimbatore Chandersekaran and William R. Simpson, International Journal of Scientific Computing, Vol. 6, No. 2, "A Uniform Claims-Based Access Control for the Enterprise," December 2012, ISSN: 0973-578X, pp. 1–23.