

Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems *

Dr. Sam Chamberlain

U.S. Army Research Laboratory
ATTN: AMSRL-CI-CT
APG, MD 21005-5067
USA

Email: sam.chamberlain@us.army.mil
URL: <http://www.arl.army.mil/~wildman>

ABSTRACT

In any information system, a critical feature is the ability to link together disparate pieces of data and information via relationships. One way to greatly facilitate this task is to provide a common technique for identifying the pieces so that they can be conveniently referenced. Arbitrary linking of data can be accomplished by standardizing one field across disparate data sources. This is the objective of enterprise identifiers (EID). If data can be globally identified using a common scheme, then one can spontaneously reference and retrieve arbitrary pieces of information with minimal prior coordination. A strategy for accomplishing this task was described by the author in a paper [1] at last year's 6th International Command & Control Research & Technology Symposium. This paper describes the implementation of an EID seed server and some of the ongoing issues encountered and being addressed.

1.0 INTRODUCTION

The term enterprise identifier is the general form of the term enterprise key (EK) that comes from the relational database community. An EK is a surrogate key¹ (SK) that is unique across the whole enterprise; in other words, it is a universal SK (USK). For example, if the enterprise is the DoD, then no two data items may have the same EID within the DoD unless they represent the exact same object. To facilitate interoperability, the format of an EID must be technology independent. It must be usable by relational databases, object databases, formatted flat files, or any other formatted source of data. This is an important characteristic for integrating disparate data systems. For a list of sources of information on topics related to implementing EIDs, see [2].

When a database management system creates (e.g., inserts) data, it must obtain an EID. This EID is attached to the data for its lifetime, that is, it is a persistent label. Because the data has an EID, no matter where it propagates within the enterprise, its EID is guaranteed never to collide with any other EIDs already resident in

* This paper was previously published in the *Proceedings of the 7th International Command and Control Research and Technology Symposium*; Quebec City, Quebec, Canada; 16-20 September 2000. This work is being transitioned via the Global Force Management (GFM) project under the leadership of the U.S. DOD Joint Staff/J-8 and the Under Secretary of Defense for Personnel and Readiness. See: http://www.dodccrp.org/events/2002/7th_ICCRTS/Tracks/pdf/109.pdf for previous publication.

¹ Surrogate Key: a single attribute, candidate key from which no information can be gleaned about the entity it identifies.

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 01 DEC 2005		2. REPORT TYPE N/A		3. DATES COVERED -	
4. TITLE AND SUBTITLE Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRL-CI-CT APG, MD 21005-5067 USA				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited					
13. SUPPLEMENTARY NOTES See also ADM202135, RTO-MP-IST-042. Coalition C4ISR Architectures and Information Exchange Capabilities (Les architectures C4ISR et les capacites d'echange d'information en coalition), The original document contains color images.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 31	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems

any database². EIDs serve the same purpose in data sources that Uniform Resource Identifiers (URI) serve in the web community³. However, this scheme for EIDs maintains three important characteristics:

- All EIDs are surrogate keys (fit in a single attribute, no embedded intelligence),
- All EIDs have the same size and form (in this implementation 64-bits),
- The size is chosen to be as small as possible.

The reason for these characteristics is to support transactions over a wide variety of military data systems that may have to operate in constrained communication environments. Low bandwidth or degraded communication conditions will likely exist in wireless, military systems for a significant duration. Any URI scheme adopted for military use must also operate under these conditions. With a few simple modifications, this EID scheme can be easily modified to meet registration requirements as a URI, but at the expense of substantial increases in bandwidth and space utilization.

The primary challenge in creating an EID system is developing an approach to guarantee that EIDs are unique, while ensuring that they can be easily obtained and cause no performance degradation. An additional objective is to provide a flexible scheme for allocating EIDs that allows the system managers as much control and freedom as possible. A common approach for producing globally unique values is by concatenating two smaller values. In this implementation, devices called EID Servers (ES) create globally⁴ unique EIDs by appending a locally managed suffix to a globally unique prefix obtained from an authoritative source. The prefix is called an EID Seed and the authoritative source is called an EID Seed Server (ESS). Figure 1 illustrates this approach.

An enterprise identifier can be composed by combining unique identifiers.

A four byte (32 bit), globally unique “EID Seed” prefix is followed by a locally created and managed four byte (32 bit) suffix.

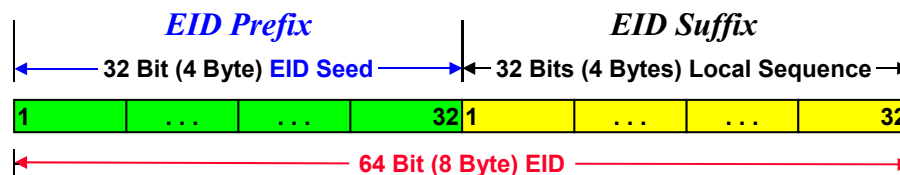


Figure 1: EID Composition

Figure 2 illustrates the EID allocation architecture. It includes three levels of components: EID users (on the right), EID servers (in the middle), and EID seed servers (on the left). EID users (usually databases) “connect” to an ES to obtain EIDs. An ES is any computer program that provides EIDs to requestors. It can be 20 lines of Java™ code (as one might find embedded in a PDA⁵) or a complex program that resides on a separate machine and serves many databases. The topic of this paper is the ESS that passes out 32-bit EID seeds to its registered users. This allows nearly 4.3 billion EID servers to be established, each able to dispense 4.3 billion EIDs for a total allocation of 18 billion-billion EIDs.

² The phrase “within the enterprise” refers to the other data sources that are also using the same EID scheme.

³ See: <http://www.ietf.org/rfc/rfc2396.txt> for the latest RFC on URIs and <http://www-old.ics.uci.edu/pub/ietf/uri/> for a comprehensive list of related topics.

⁴ The term “globally” refers to all the members of the enterprise.

⁵ PDA – Personal Digital Assistant, for example, a Palm Pilot.®

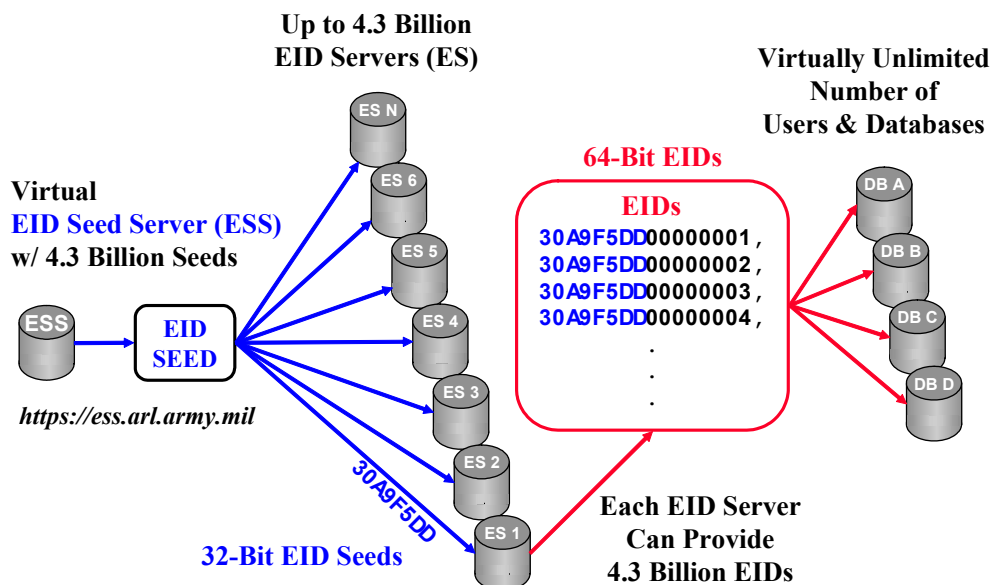


Figure 2: An EID Server Architecture

To avoid confusion, one must understand the difference between EIDs, EID seeds, EID servers, and EID seed servers. To reiterate these differences, an EID is a 64-bit value that is used to uniquely identify a database record or object. Internally, it may be represented as a 64-bit binary value, two 32-bit binary values⁶, or as sixteen hexadecimal characters.⁷ EIDs are obtained from EID servers (ES) that create them by appending a locally controlled 32-bit suffix to a 32-bit prefix. The prefix is an EID seed that is obtained via an EID seed server (ESS) on which one must have an account. Observe that the reason that an ES can produce nearly 4.3 billion EIDs is because of the size of the locally controlled suffix. The fact that the prefix and suffix are both 32-bits is a coincidence, or perhaps better stated, they are convenient numbers to use with computers. Any sized prefix and suffix will work. The reason 64-bits was selected was that it was determined that this was the smallest value that could accomplish the task. Larger values will work equally well, but due to bandwidth considerations, the smallest possible values were selected.

2.0 ACCESSING THE EID SEED SERVER

In April 2002, the US Army Research Laboratory (USARL) established a prototype ESS. To obtain EID seeds, one must obtain an ESS account. Obtaining an account is simple, is open to anyone who wishes to apply, and is accomplished by visiting the URL <https://ess.arl.army.mil/>. When one applies, they must first agree to abide by four basic criteria — they will ensure that:

1. their point of contact information is always kept current,
2. the ESSs they establish produce 64-bit EIDs using a bona-fide EID seed obtained from an ESS as a prefix,

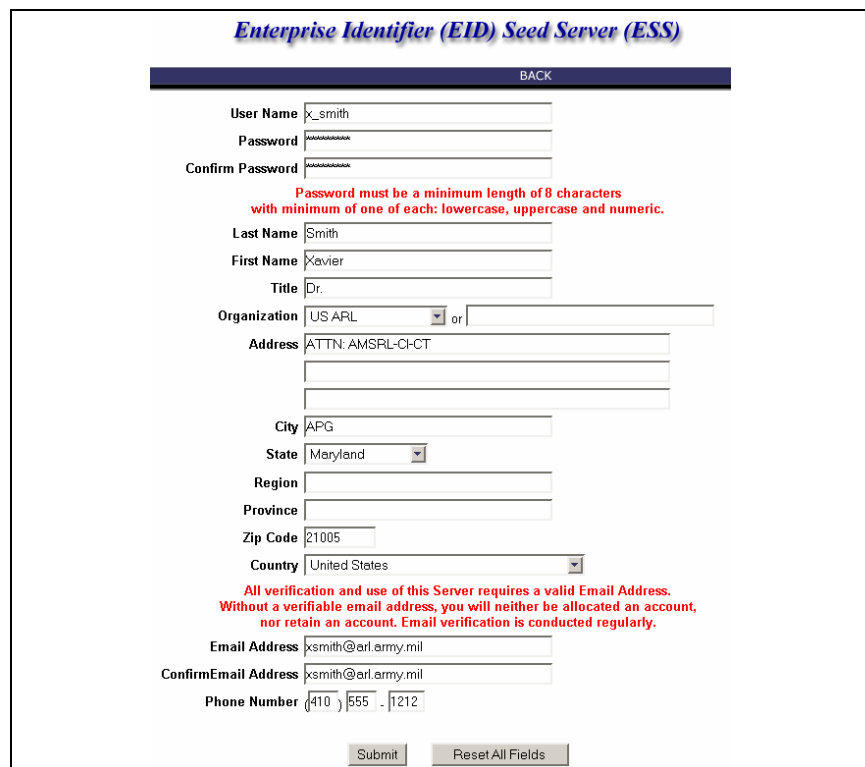
⁶ Some databases do not yet support 64-bit binary values.

⁷ In hexadecimal notation, an ASCII character represents a pattern of four bits. Therefore, the 16 ASCII characters 0-9 and A-F are used to denote the 16 combinations of four bits: 0000 through 1111. Note that it is a coincidence that 16 characters are also required to represent a 64-bit EID.

Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems

3. EIDs are maintained in either binary form as a single 64-bit field, or two 32-bit fields, or as 16 characters using hexadecimal notation, and
4. the ES includes a mechanism to guarantee that EIDs are never duplicated. This implies that the ES must have some type of backup scheme to prevent re-use in the event of a power loss or major malfunction.

Once one has agreed to these four conditions, the requester is transferred to a registration page to enter the usual official contact information. The current page is illustrated in Figure 3. Each user recommends a unique user name to allow them to log on to the ESS. All contact information can be updated; but two items, organization and email address, require the concurrence of an ESS account administrator. The reason for this is to maintain control of who actually is using the ESS. If organizational email addresses are used, the account representative may change for an organization without any ESS administrator involvement. The “Region” and “Province” fields are to facilitate international users. New fields can be easily added as required.



Enterprise Identifier (EID) Seed Server (ESS)

BACK

User Name

Password

Confirm Password

Password must be a minimum length of 8 characters
with minimum of one of each: lowercase, uppercase and numeric.

Last Name

First Name

Title

Organization or

Address

City

State

Region

Province

Zip Code

Country

All verification and use of this Server requires a valid Email Address.
Without a verifiable email address, you will neither be allocated an account,
nor retain an account. Email verification is conducted regularly.

Email Address

Confirm Email Address

Phone Number

Figure 3: The Current ESS Registration Page

Once the request is submitted, an email is sent to the email address specified that includes information on how to validate the email address. This uses the common validation technique of having the requestor log on to a machine to enter a code sent in the email. Once this is accomplished, an email is sent to the ESS administrators notifying them that an account request is pending from a validated source. An ESS administrator may then approve the ESS user account.

Anyone from any organization or country may request an account. Because a goal of the prototype is to encourage the use of EIDs and to gain an experience base, it is unlikely that anyone will be turned down.

Eventually, this may change and accounts may be scrutinized to limit subscribers to those with a bona fide requirement or official capacity to create official data. Some association with the US Department of Defense (DoD) could be required (e.g., one is a member of, or doing work for, a defense establishment). Examples are officials in corporate information offices (CIO); acquisition organizations or companies, such as program executive officers (PEO), project managers (PM), and systems builders; and the simulation community. To handle non-traditional cases, a human will control the account authorization process. This is the role of the ESS account administrators. The current requirement is to respond to a request within 24 hours, although the goal is to respond within minutes.

Once approved, a requestor becomes a subscriber and may request EID seed accounts. For most people, one account is sufficient. However, it is envisioned that a single person may require several accounts because they are handling several end-users or may have security considerations. Consequently, a user may set up several accounts. To create an EID account, a subscriber logs on to the ESS and goes to the “New Account” page. This page is illustrated in Figure 4. An EID seed account does not have to have a globally unique name since it is associated with an ESS user account. Therefore, ESS users can select any name that is meaningful to them.

Enterprise Identifier (EID) Seed Server (ESS)

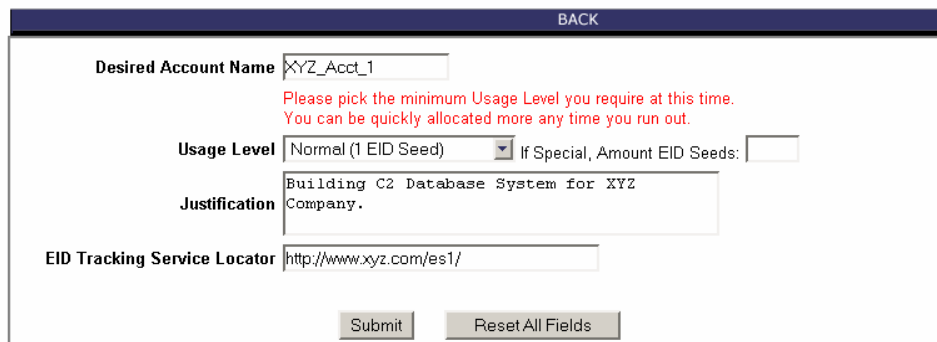


Figure 4: The New EID Account Page

2.1 Usage Level

There are four usage levels available for an EID seed account: Normal (1 EID Seed), Moderate (10 EID Seeds), Heavy (100 EID Seeds), and Special (> 100 EID Seeds). The objective is to provide the EID seed user with as many seeds as they require, but no more. For this reason, users are encouraged to be frugal and may increase their usage level at a later time. Recall that one EID seed supports the creation of 4.3 billion EIDs. Therefore, for many users one EID seed is enough. If they use up their 4.3 billion EIDs, they simply increase their EID usage and obtain another EID seed.

As explained in [1], the primary reason for requiring more than one EID is database performance. The use of EIDs must not cause any database performance degradation or system management problems. If a set of database management systems (DBMS) is tightly coupled over a high bandwidth communications environment, then a single ES (established with a single EID seed) may be adequate to serve many systems without degradation. However, if one has many distributed systems that are widely dispersed, then an EID seed may be required for each system. This is the expected situation for battle command systems and is the

Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems

environment for which this implementation is based. If a project manager is building 12,000 tactical systems, then 12,000 EID seeds may be required. If every person has a wearable computer, then every one of those systems may require a separate EID seed.

Notice, however, that the DBMS dispersion problem can be addressed in another way. If every system is likely to create 4.3 billion pieces of data, then a separate EID seed, or EID prefix, is required for each system. This equates to solving the problem at the ESS level. The alternative is for the system manager to solve the problem at the ES level by allocating the locally controlled, EID suffix into blocks. For example, the 4.3 billion EID suffixes could be broken into 100 43 million EID blocks and dispersed into 100 distributed systems. As a system nears the end of its block of EIDs, the manager obtains a new EID seed and creates another set of blocks to be distributed. It is this flexibility that is afforded by the properties of surrogate keys (i.e., no embedded intelligence) that makes managing EIDs so simple.

Just as with ESS user accounts, an ESS administrator approves and enables individual EID seed accounts. Once approved, the ESS user may obtain EID seeds up to the maximum number authorized for the EID seed account without communicating with an ESS administrator. EID seeds are obtained using the “Get New Seed” page as is illustrated in Figure 5. This page provides information on the usage of the EID seeds for a particular account. In this case, it states that this account has 1 EID seed remaining. The user can request up to that number. When this amount is entered and the “GET EID” button is pressed, the ESS allocates that number of EIDs to the EID seed account. Simultaneously, the user is sent an email stating that EID seeds were allocated to the specified account.

Enterprise Identifier (EID) Seed Server (ESS)

Logoff	Home	New Account	Update Contact Info	Increase Usage	Get New Seed	Find EID	Tutorial
--------	------	-------------	---------------------	----------------	--------------	----------	----------

x_smith

Last logged in: 04/11/2002

Amount of Logins: 6

Next Distributed EID: 01000002

EID seeds		
Account	Remaining	Requesting
XYZ_Acct_1	1	1 GET EID

Figure 5: The Get EID Seed Page

When users log on to the ESS, they are placed in their ESS User Area (i.e., homepage) as is illustrated in Figure 6. This page lists all of their accounts, their status, the number of EID seeds remaining, and the number of EID seeds obtained. A user can acquire a list of their EID seeds by selecting the “List” text as is illustrated in Figure 7. In this example, this account has EID seed “01000002” that was allocated to it on 11 April 2002. Therefore, this ESS user may establish an ES and assign this seed to be its EID prefix. The ESS guarantees that no other ESS subscriber has been assigned this seed, and therefore, no EID created using it will ever collide with any EID created by another ESS subscriber. The status field and modification date will be described in the following section on EID tracking.

An interesting ramification of this approach is that *the enterprise* is now defined as the set of ESS subscribers. This set is currently not constrained by service, governmental, or national boundaries. This facilitates an open

medium by which to experiment with and experience the advantages of EIDs. It is the author's experience that standardization edicts are rarely successful. Consequently, the use of EIDs is proposed as a strictly voluntary action. However, it is expected that EID users will rapidly realize the advantages of using EIDs and the practice will quickly spread. For two organizations to implement EIDs, all that is required is an agreement by the parties to get ESS accounts and abide by the four basic rules. Further, as explained in [1], although there are significant advantages for database maintenance when using EIDs as primary keys, all the interoperability advantages are obtained when they are used as alternate keys. Thus, EIDs can be implemented without modifying the existing primary key system in a legacy database. If the legacy system is a relational database, all that is required is the addition of a single new column in each data table to contain an EID. It is then a simple procedure to assign an EID to each row.

Enterprise Identifier (EID) Seed Server (ESS)

Logoff Home New Account Update Contact Info Increase Usage Get New Seed Find EID Tutorial			
x_smith			
Last logged in: 04/11/2002			
Amount of Logins: 7			
Next Distributed EID: 01000003			
LIST All Requests Disapproved Disabled			
User Area			
ALL ACCOUNTS for Xavier Smith (x_smith)			
EID seeds			
ACCOUNT NAME	Remaining	In Use	STATUS
Main_Acct_1	1	0 (List)	Approved
XYZ_Acct_1	0	1 (List)	Approved

Figure 6: ESS User's Area (Home Page)

LIST All Requests Disapproved			
User Area			
EIDs for Account XYZ_Acct_1			
EID seed	Assign Date	State*	Modification Date
01000002	04/11/2002	Active	04/11/2002
* Clicking on State will toggle to next available state.			
List EIDs ONLY			

Figure 7: Listing the EID Seeds of an EID Seed Account

Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems

2.2 EID Tracking

With any distributed DBMS, procedures should be implemented to ensure that references to data are accompanied by the data referenced. In other words, if a field in one database table references a row of another database table, then that row must be present. If this is violated, a referential integrity error occurs. This situation is no different for EIDs as other types of keys. Although it should rarely happen, it is possible to be given an EID that cannot be found in the distributed database. However, since EIDs have no embedded intelligence, it is often impossible to devise a scheme to track down the data item that caused the referential integrity error. To facilitate a solution to this problem, an EID tracking system is being developed.

Although an EID has no embedded intelligence, it is known that an EID prefix is an EID seed that was obtained from the ESS. Therefore, one can at least discover who was given an EID seed by contacting the ESS. Current ESS policy is that the contact information associated with an ESS user account is available to all subscribers. This feature is executed using the “Find EID” option. Figure 8 illustrates both a “Find EID” page and a corresponding response page. In this example, an EID of “0100000200050FFE” is entered, and the corresponding contact information is provided. The EID prefix is parsed from the EID (first 8 hexadecimal characters) and that EID seed is used to query the ESS. In this case, the basic contact information entered by the ESS user is returned: name, phone number, and email address. It would be up to the inquirer to then contact the EID seed owner and ask about the data item tagged with the EID “0100000200050FFE.” Of course, it is totally at the EID seed owner’s discretion to divulge the information. This is similar to IP addresses. One is not *required* to register an IP address with a hostname via a domain name service. However, most people do this to allow their hostname, or URL⁸, to be used instead of a raw IP address. The same practice is true with EIDs.

FIND EID seed

You need to provide either an EID (16 HEX characters) or EID seed (8 HEX characters)

Enter an EID / EID seed in HEX format:

FIND EID seed

EID seed 0100000200050FFE
Assignment Date 04/11/2002
Last Modified 04/11/2002
Name of POC Xavier Smith
Phone Number (410) 555-1212
Email Address x_smith@arl.army.mil
EID Tracking Service Locator <http://www.xyz.com/es1/>

Figure 8: The Find EID Request and Response Pages

What is desirable is to have an automated system to do EID searches. This is the function of the EID Tracking Service Locator (TSL). In Figure 8, a URL has been entered to indicate that this ESS user has implemented a tracking system for some or all of the EID seeds it has obtained. This allows the system to

⁸ URL: Uniform Resource Locator, more commonly known as a web address.

automatically forward the “Find EID” command to the URL (along with the source information about who is asking). This capability, and the associated protocol specification, is currently under development. The concept is that this process continues until the DBMS containing the data is discovered. Then, a “Get EID” command is issued to the DBMS and the resulting data is sent to the requestor in the form of an XML document⁹. This requires that a mechanism be developed to convert the “Find EID” command into a local “Get EID” command that executes a query in the native database query language of the appropriate DBMS and converts the result into XML code to be returned to the requestor. In other words, a tracking server system must include a “Get EID” function that interacts with the local DBMS’s.

A recursive EID tracking system is illustrated in Figure 9. This example shows several different levels of complexity that are based upon the implementation decisions of the system manager. The line marked with an ‘A’ shows the simplest level that occurs when an ES (labeled “ES3”) is embedded in a DBMS. The user’s TSL allows the ESS to redirect the “Find EID” request directly to the source of the data. If the user is tracking the listed EID, then the ES executes a Get EID command to the database, which in turn generates the XML code that is returned to the requestor.¹⁰ In actuality, this is probably the most likely scenario because the single EID seed with a single independent database is expected to be the most common situation. Under this condition, the data that is tagged with the requested EID is discovered with only a single redirection by the ESS of the Find EID operation.

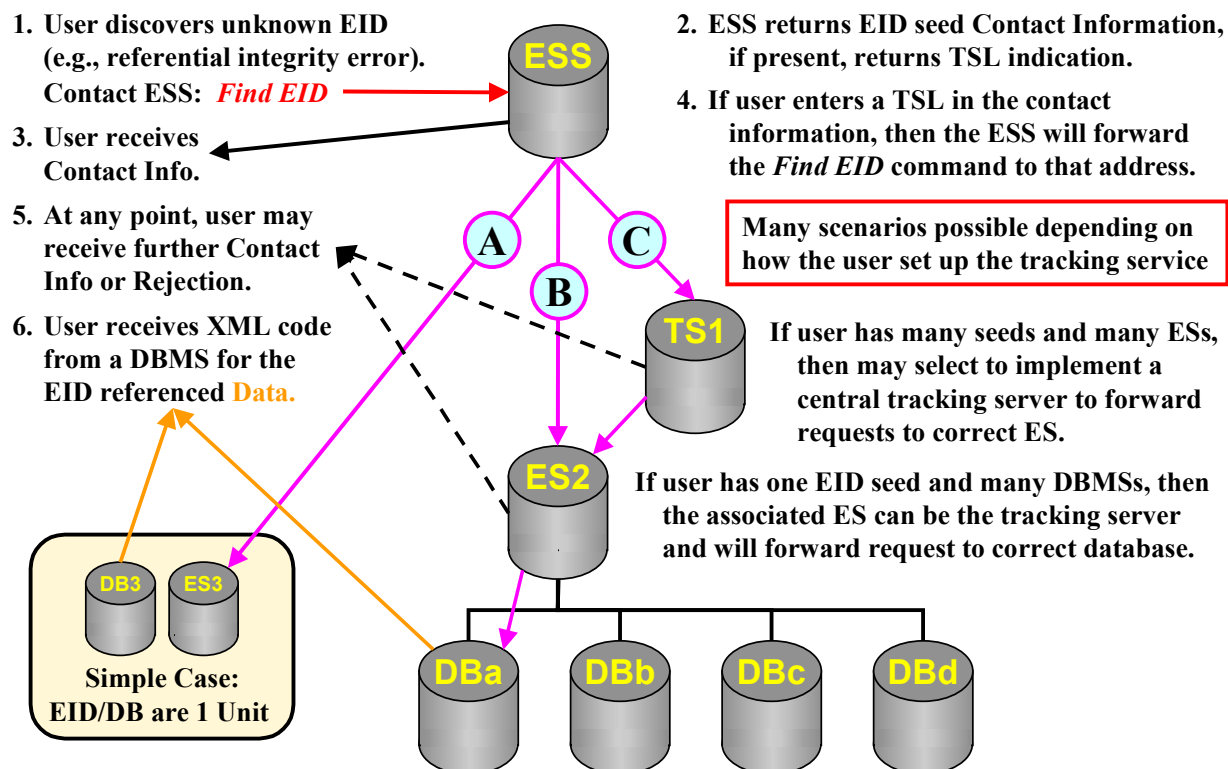


Figure 9: A Recursive EID Tracking System

⁹ XML: The Extensible Mark-up Language, a standard for describing the structure of information; see <http://www.w3.org/XML/>.

¹⁰ Alternatively, the DBMS could intercept the Find EID command and execute this process.

Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems

The line marked with a 'B' illustrates a slightly more complex level in which the ES manager has decided to implement EID suffix blocking as was explained in Section 2.1. Because only the user (not the ESS) has any knowledge of the blocking (or any allocation) scheme, a "Find EID" request must first be directed to the ES for further redirection. In this case, the ES (labelled "ES2") forwards the "Find EID" command to the correct DBMS (e.g., DBa). Alternatively, it could forward a "Get EID" command to the appropriate DBMS and handle the results of the query by converting it into XML code and returning it to the requestor.

The line marked with a 'C' illustrates the most complex level. In this case, a user has an EID seed account with multiple EID seeds each with its own ES. When this occurs, the user must have an intermediate tracking service established to forward the "Find EID" command to the correct ES. This is because of the decision to associate the TSL with an EID seed account, and not with the individual EID seeds. It does not matter if a user has a normal, moderate, heavy, or special EID seed account, each account may be associated with only a single TSL. Originally, it was considered to allow a TSL to be designated for each EID seed, but it was decided that this would place an unnecessary burden on the ESS and also on the ESS users who would have to contact the ESS for what are essentially local changes to their tracking service. By having a single TSL per EID seed account, the burden of effort is divided equitably between the ESS and the user's local systems.

Observe that from the ESS's perspective, all three of these cases are the same. When a "Find EID" command is executed, the ESS simply retrieves the EID seed account under which the EID was allocated. If a TSL field entry exists, it forwards the "Find EID" command to the URL entered in the TSL field. In reality, all these examples are figurative since there are numerous ways to implement tracking servers and ESSs. All functions may reside on a single machine (with a single hostname), each on a separate machine, or some combination of machines. It is this flexibility that allows users to manage their data and the allocation of EIDs as they determine locally. This allow the ESS to be as unintrusive as possible.

An ESS characteristic that was debated is whether visibility of user contact data should be mandatory (as it is currently), or whether it should be user selectable. In other words, should the ESS users be able to specify whether the "Find EID" command returns their contact information. One argument is that there may be users who do not wish to be identified (e.g., intelligence organizations). It was decided that the ESS should remain completely open and independent of any organizational security issues (another advantage of the properties of surrogate keys). If an ESS user does not want to be identified, then a proxy can always be hired to be their ESS account manager (one of the reasons for allowing multiple EID seed accounts per user). Having a "front" is not a novel idea in many businesses. Many organizations specialize in the practice of masking the identity or true character of other organizations. This is not the business of the ESS. All that is required is that its users abide by its four stipulations, which includes maintaining current contact information.

Observe that there are no guarantees that a tracking service will be provided for an EID. All that is guaranteed is that contact information for an EID seed will be provided. A user is not required to provide a TSL or any tracking services. Furthermore, a user can arbitrarily provide tracking for some EIDs and not others. This means that a requestor may receive a refusal notice at any level of the tracking function exemplified in Figure 9. Note that this is also true when tracking IP addresses within the Internet.

Finally, there is a user settable parameter to indicate the activation state of an allocated EID seed. This is a toggled value that a user can set for each EID seed as is illustrated in Figure 10. There are several reasons for this field, the most prominent being the time lapse between the acquisition of an EID seed and the establishment of an ES that uses that seed. When a user executes the "Get EID" command (see Figure 5), the EID status is set to "inactive." Once an ES is established and is activated, the user can toggle this field to indicate that the EID seed is now in use. Currently, this parameter is used only as an indicator (e.g., to display

the status). So a user can toggle this variable without any effect on the ESS. For example, a user may change the state to “inactive” to indicate that an ES is no longer active or operational. This does not automatically imply that the tracking services for the ES are switched off, but only that it is no longer available for dispensing EIDs. Eventually, this value may be used in ESS functions like “Find EID.” Future uses for this field are still being discussed.

User Area			
EIDs for Account XYZ_Big_Acct			
EID seed	Assign Date	State*	Modification Date
01000003	04/12/2002	Active	04/12/2002
01000004	04/12/2002	Active	04/12/2002
01000005	04/12/2002	Active	04/12/2002
01000006	04/12/2002	Inactive	NOT MODIFIED
01000007	04/12/2002	Active	04/12/2002
01000008	04/12/2002	Active	04/12/2002
01000009	04/12/2002	Active	04/12/2002
0100000a	04/12/2002	Inactive	NOT MODIFIED
0100000b	04/12/2002	Inactive	NOT MODIFIED
0100000c	04/12/2002	Inactive	NOT MODIFIED
0100000d	04/12/2002	Active	04/12/2002
0100000e	04/12/2002	Active	04/12/2002
0100000f	04/12/2002	Inactive	NOT MODIFIED
01000010	04/12/2002	Inactive	NOT MODIFIED
01000011	04/12/2002	Inactive	NOT MODIFIED
01000012	04/12/2002	Active	04/12/2002

Figure 10: EID Seed Activation

3.0 SUMMARY

To achieve interoperability across automated information systems (AIS), a critical feature is the ability to link together disparate pieces of data and information. One way to greatly facilitate this task is to provide a common technique for identifying data and information so that they can be conveniently referenced. Arbitrary linking of data can be accomplished by standardizing one field across disparate data sources. This is the objective of enterprise identifiers (EID). If data can be globally identified using a common scheme, then one can spontaneously reference and retrieve arbitrary pieces of information with minimal prior coordination. This can be accomplished using alternate keys, thus having little (or no) affect on legacy data systems.

EIDs serve the military database community in the same way that URIs serve the web community. This implementation of EIDs maintains three important characteristics:

- All EIDs are surrogate keys (fit in a single attribute, no embedded intelligence),
- All EIDs have the same size and form,
- The size is chosen to be as small as possible.

Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems

The reason for these characteristics is to support transactions over a wide variety of military data systems that may have to operate in constrained communication environments. Any URI scheme adopted for military use must also operate under these conditions. With a few simple modifications, this EID scheme can be easily modified to meet registration requirements as a URI, but at the expense of substantial increases in bandwidth and space utilization.

This paper describes an implementation of a prototype EID seed server (ESS) and some of the challenges encountered that are being addressed. Primary characteristics include simplicity, flexibility, and unintrusiveness. To accomplish these characteristics, a three-tiered approach is used that is based upon the common idea of producing globally unique values by concatenating a locally managed unique value to an existing globally unique value. Perhaps the most difficult task in implementing EIDs is obtaining an agreement within the enterprise to use them. For this reason, it is recommended that the use of EIDs continue to be voluntary until users have an adequate opportunity to try them and experience their benefit for themselves. In this implementation of EIDs, the enterprise is defined as the set of ESS subscribers. Perhaps no other single agreement between data managers will have a larger payoff than agreeing on this very basic feature of their data.

4.0 REFERENCES

- [1] S. Chamberlain: "An Enterprise Identifier Strategy for Global Naming Across Arbitrary C4I Systems," Proceedings of the 6th International Command and Control Research and Technology Symposium; US Naval Academy, Annapolis, MD; 19-21 Jun 2001.
http://www.dodccrp.org/events/2001/6th_ICCRTS/Cd/Tracks/Papers/Track2/059_tr2.pdf
- [2] Papers related to Surrogate Keys, Enterprise Keys, and Enterprise Identifiers:
 - E.F. Codd:
 - "Extending the Relational Model to Capture More Meaning,"
ACM Transactions on Database Systems, Vol 4(4), Dec 1979.
 - C.J. Date:
 - "An Introduction to Database Systems, Volume II," Addison-Wesley, Reading, MA, 1983;
ISBN 0-201-14474-3
 - "Relational Databases: Selected Writings," Addison-Wesley, Reading, MA, 1986.
ISBN 0-201-14196-5
 - T. Johnston:
 - "Primary Key Reengineering Projects: The Problem;" *DM Review*, February, 2000,
<http://www.dmreview.com/master.cfm?NavID=55&EdID=1866>.
 - "Primary Key Reengineering Projects: The Solution;" *DM Review*, March, 2000,
<http://www.dmreview.com/master.cfm?NavID=55&EdID=2004>
 - "De-Embedding Foreign Keys," *DM Direct* (online): Part 1, June 2, 2000.
http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2308.

- “De-Embedding Foreign Keys,” *DM Direct* (online): Part 2, June 9, 2000.
http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2322.
- “De-Embedding Foreign Keys,” *DM Direct* (online): Part 3, June 16, 2000.
http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2331.
- “De-Embedding Foreign Keys,” *DM Direct* (online): Part 4, June 23, 2000.
http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2341.
- M. Lonigro, Mike:
 - “The Case for the Surrogate Key”, Intelligent Enterprise Database Programming and Design On-line, May 1998; <http://www.dbpd.com/vault/9805extra.htm>.
- “Enterprise Identifiers For Logistics -
An Approach in Support of Army Transformation Initiatives,”
http://www.arl.army.mil/~wildman/PAPERS/log_study.html.

**Implementation of an Enterprise Identifier
Seed Server for Joint and Coalition Systems**





Implementation of an Enterprise-Wide Identifier Seed Server for Joint and Coalition Systems*

NATO RTO Symposium on
Coalition C4ISR Architectures and
Information Exchange Capabilities

The Hague, The Netherlands
27-28 September 2004

Sam Chamberlain, Ph.D.
US Army Research Laboratory
sam.chamberlain@us.army.mil
410- 278-8948 // DSN 298
<http://www.arl.army.mil/~wildman>

* Sponsored in part by the DoD C4ISR Cooperative Research Program (www.dodccrp.org), Dr. Dave Alberts, Director, Research, ASD(C3I)



Recent History (FY 2004)

- Under the *US DOD Net Centric Data Strategy* (9 May 2003), **Communities of Interest (COI)** are to be established to address organization and maintenance of data.
- In the summer of 2003, a COI for **Global Force Management (GFM)** was established by the Joint Staff Force Structure Directorate (J-8) and co-chaired by the US DOD Deputy Under-Secretary of Defense (Personnel & Readiness)/Readiness to tackle the challenges imposed by the *Net-Centric Data Strategy*.
- The major impetus for the establishment of the GFM-COI is the development of reliable and maintainable data sources in a net-centric environment to support decisions relating to force management for systems such as the new **Defense Readiness Reporting System (DRRS)**.
- A project called the **GFM Enterprise Data Initiative (EDI)** is underway to investigate and evaluate GFM data structure, creation, management, and accessibility (e.g., via web service enabling).
- Part of this project addresses a key management strategy for DoD data sources. This has resulted in a **DoD Directive** currently being staffed to implement **Enterprise –Wide Identifiers (EwIDs)** for tagging arbitrary DoD data. EwIDs were previously called **EIDs** in the author's literature.



The Problem:

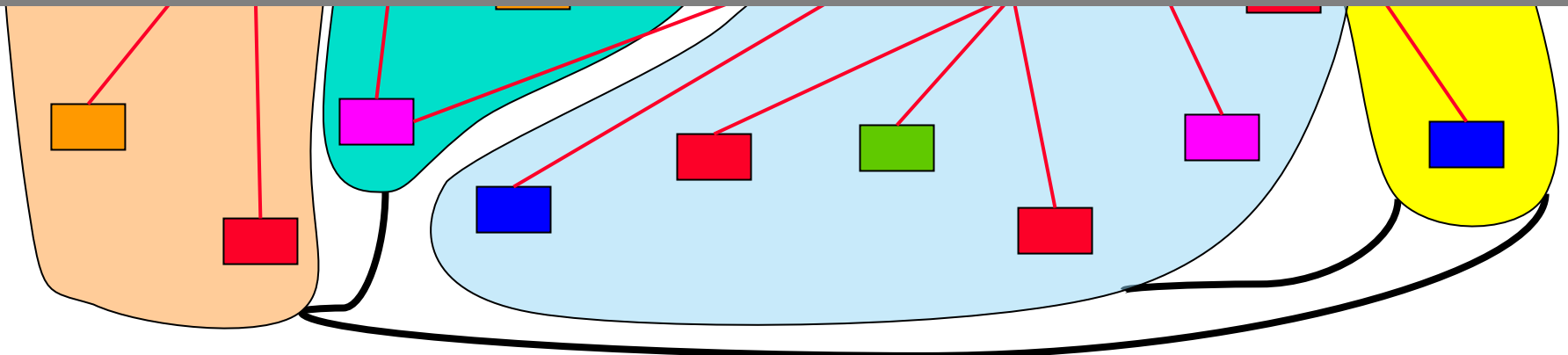
**Building Associations Between
“Pieces of Information”**



Establishing Relationships Between Data Items

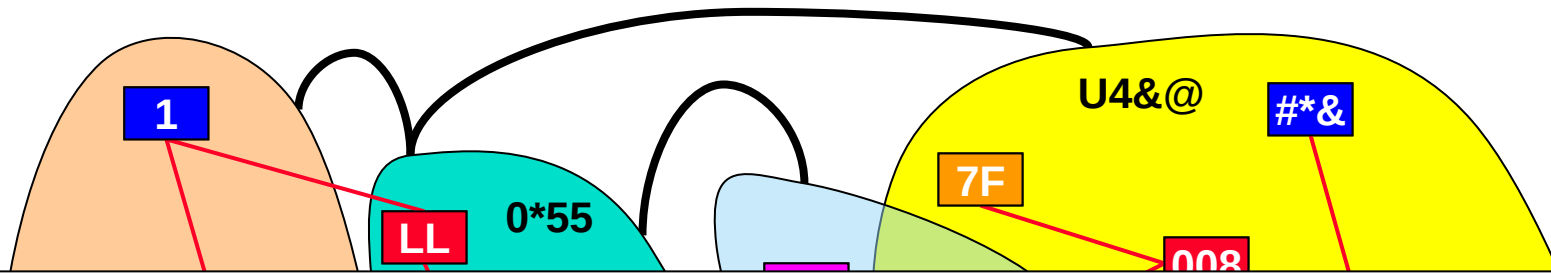
**THERE IS TREMENDOUS POWER IN THE
RELATIONSHIPS BETWEEN THE PIECES.**

THAT IS, THE **LINKS ARE AS IMPORTANT AS THE **NODES**
(SOMETIMES, EVEN MORE SO!)**





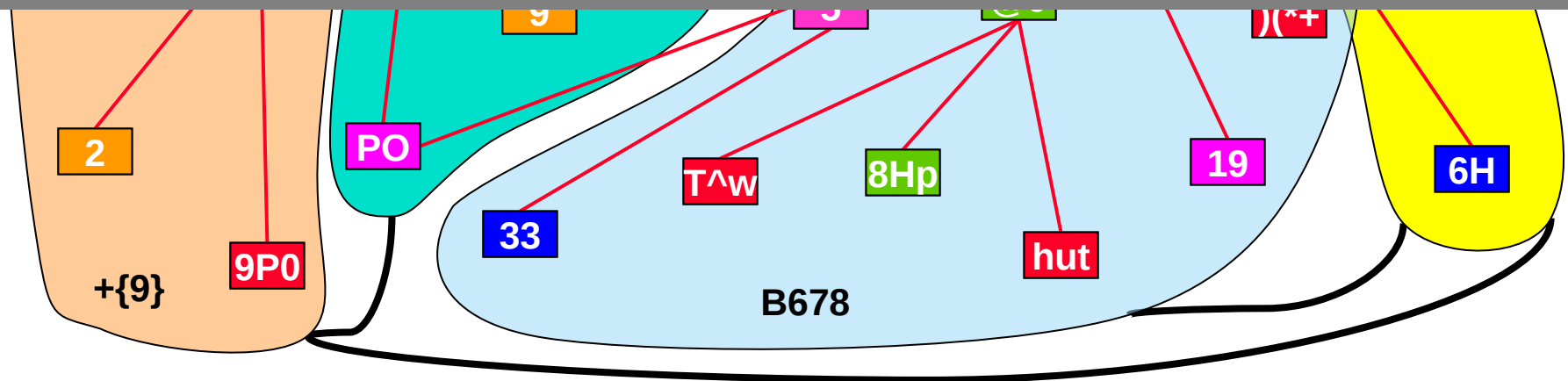
Technically, Names Must Only Be Unique, Not Consistent



But This Makes Interoperability Difficult.

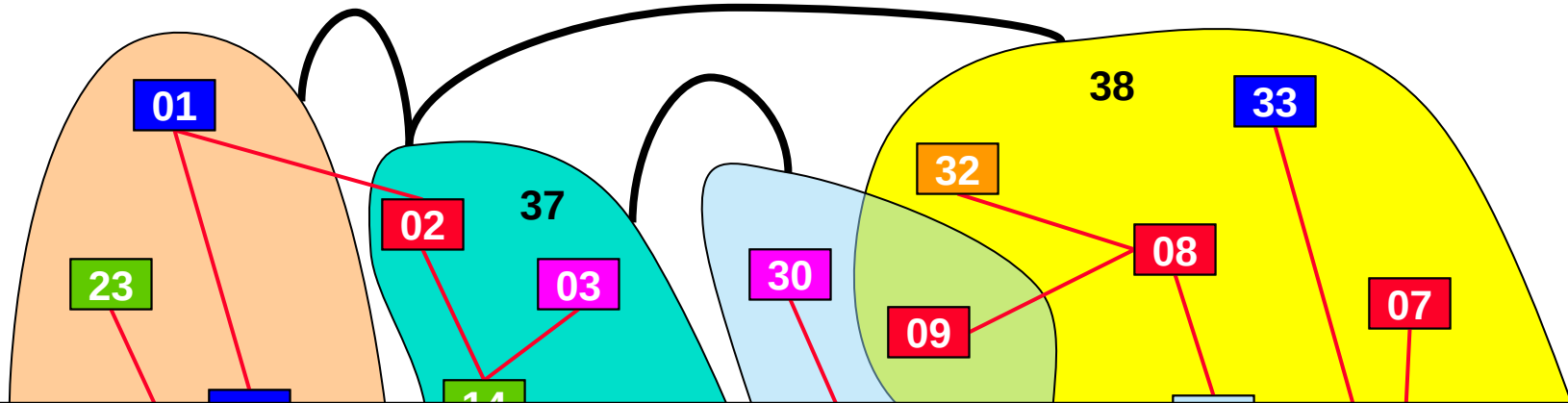
It is highly advantageous to have a standard for this component of the architecture because it facilitates “plug and play” of information;

Plus, it is difficult to ensure uniqueness without a standard.

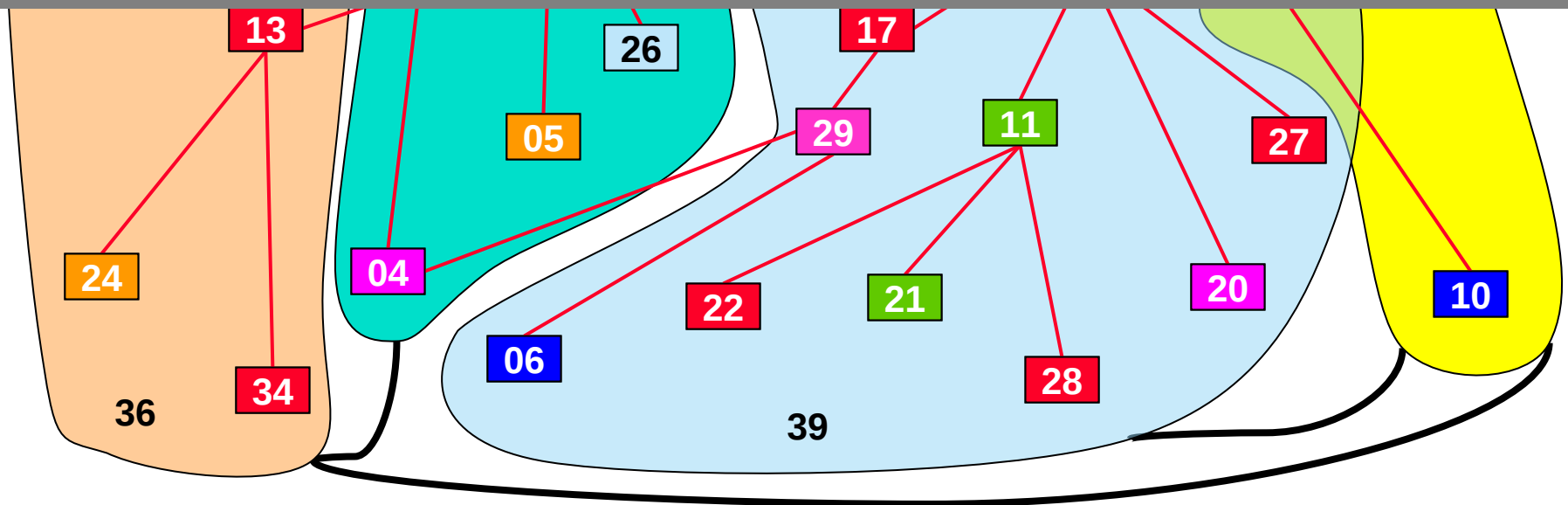




A Common Naming Convention Across The Enterprise Is Highly Advantageous



This Is a Main Feature of Enterprise-Wide Identifiers (EwID)
A Unique Value in a Common Format





Review - Identifier Terms (Initially, A Relational Database Perspective)

- **General:** EwIDs are the general form of Enterprise Keys (EK) documented in the relational database community. Here, EwID is a data-type, not a field name.
- **Primary Key (PK)** is a set of fields in a database table that uniquely identifies an entry.
 - When building relationships, the primary key is often built by combining primary key fields from the associated tables plus some new fields.
 - This produces primary keys of many different sizes and “shapes.”
- **Surrogate keys (SK)** are single attribute PKs with no embedded intelligence.
 - One can glean nothing about the data from the key that identifies it.
 - Therefore, they are impervious to change (there is no reason to change them).
- **Enterprise-wide keys (EK)** are SKs that are unique across all the tables of the enterprise.
 - Ideally, they all have the same size and shape (i.e, a single attribute),
 - One needs a way to guarantee uniqueness.
- **Business Keys** are alternate keys, often with intelligence, for use by the database users.
- **Object IDs** are the object technology equivalent of a PK and typically have EK properties within the object database.



Enterprise (wide) Identifiers (EwID)

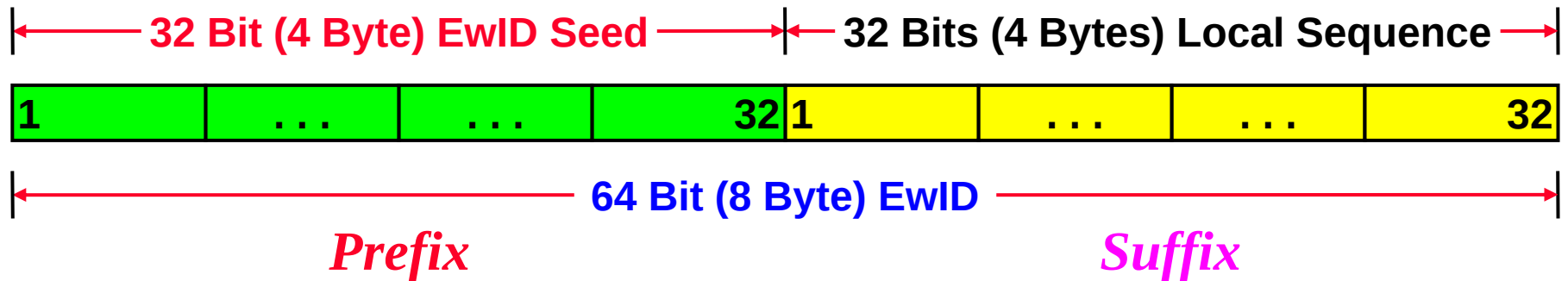
- **Identifier:** a property that uniquely distinguishes an item.
[*The most basic requirement of data.*]
- **EwID:** An identifier that is unique across the enterprise (e.g., DoD).
- **Fundamental (Required) Characteristics:**
 - It includes no information about the entity it identifies (“surrogate key” property from relational databases).
 - It is a fixed size (ease of software development and interoperability).
- **Recommended Characteristics:**
 - Size: 64 bits is the smallest size that will do the job (bandwidth is a consideration)
 - Allocation Scheme: Global Prefix, Local Suffix for simplicity.
- When any data is created, it is tagged with an EwID that remains associated with it for its life.
- Technical Challenge: to guarantee uniqueness without sacrificing reliability and performance (i.e., no bottlenecks).



EwID Formulation Scheme

An enterprise-wide identifier to uniquely identify *any item* in *any database* can be composed by combining unique identifiers.

First, a globally unique, four byte (32 bit) “**EwID Seed**” is obtained from an **EwID Seed Server**.

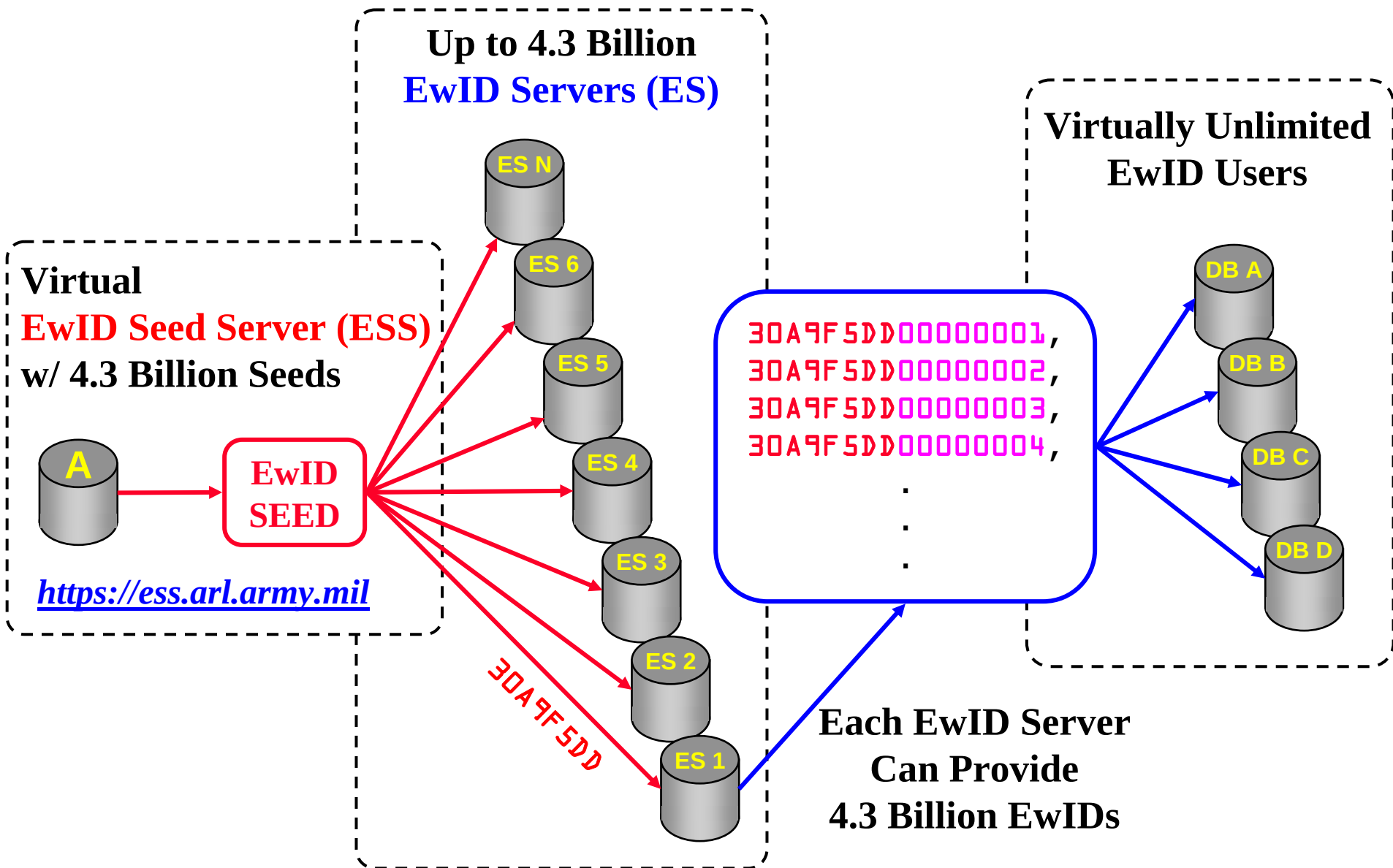


Then, an **EwID Server** is established to provide **EwIDs** to users by producing globally unique, eight byte (64 bit) **EwIDs** by appending a locally controlled, unique, four byte (32 bit) suffix to the EwID Seed prefix.

The common, eight byte (64-bit) *enterprise-wide identifier* format allows 2^{64} bit patterns = 18.45×10^{18} , or 18.45 Exa-identifiers, or 18 Billion Billion Unique Entities to be tracked. In other words ... **4.3 billion EwIDs can be produced from each of the 4.3 billion EwID Seeds.**



Enterprise-wide Identifier Allocation Hierarchy





What's An EID Server [ES]? (1 of 2)

- **Any program that passes out EIDs and abides by three criteria:**
 1. The EIDs produced must be 64-bit values with a prefix that is a bona-fide EID seed obtained from an ESS,
 2. The EIDs are maintained in one of three forms:
 - a. Binary, as a single 64-bit field, or
 - b. Binary as two 32-bit fields, or
 - c. 16 character ASCII String using hexadecimal notation.
 3. It includes a mechanism to guarantee that EIDs are never duplicated. This implies that the ES must have some type of backup scheme to prevent re-use in the event of a power loss or major malfunction.
- **An ES can reside on a mainframe, a PDA, or anything in between.**



What's An EID Server [ES]? (2 of 2)

- **Major Functions Are Determined Locally by the ES Owner:**
 - **ALLOCATION SCHEME:**
 - Minimum: $\text{New EID} = \text{Previous EID} + 1$ and store New EID.
 - Maximum: As elaborate as required (blocked values, etc.).
 - **ACCESS PERMISSIONS:**
 - Single user: only one program/machine/user has access.
 - Multi-User: many programs/machines/users have access.
 - **SECURITY FEATURES:**
 - Encrypted links
 - Password protection
- **Optionally, may provide a tracking service - who/what received an EID.**



What's An EID Seed Server [ESS]? (1 of 2)

- A special set of servers that pass out 32 bit EID Seeds to subscribers.
- ESS Interface
 - Password protected and encrypted (SSL).
- Users must register to obtain an account:
 - Currently, there is a “person in the loop” to grant authorization.
 - Approved subscriber receives a password for a *user account*; then may create several *EID seed accounts*.
 - EID Seed accounts have a maximum number allocated, but this value can be increased by a request.
 - Subscriber must keep point of contact information current.



What's An EID Seed Server [ESS]? (2 of 2)

- **ESS provides several services:**
 - **New EID seed/block of seeds.**
 - **Account Tracking Service –**
“who” was given a particular EID Seed.
POC data returned; a URL may be included.
(This is the “Find EID” command.)
 - **Future: Validation service (for EID servers).**
 - **Future: User option - Recursive Tracking Service.**



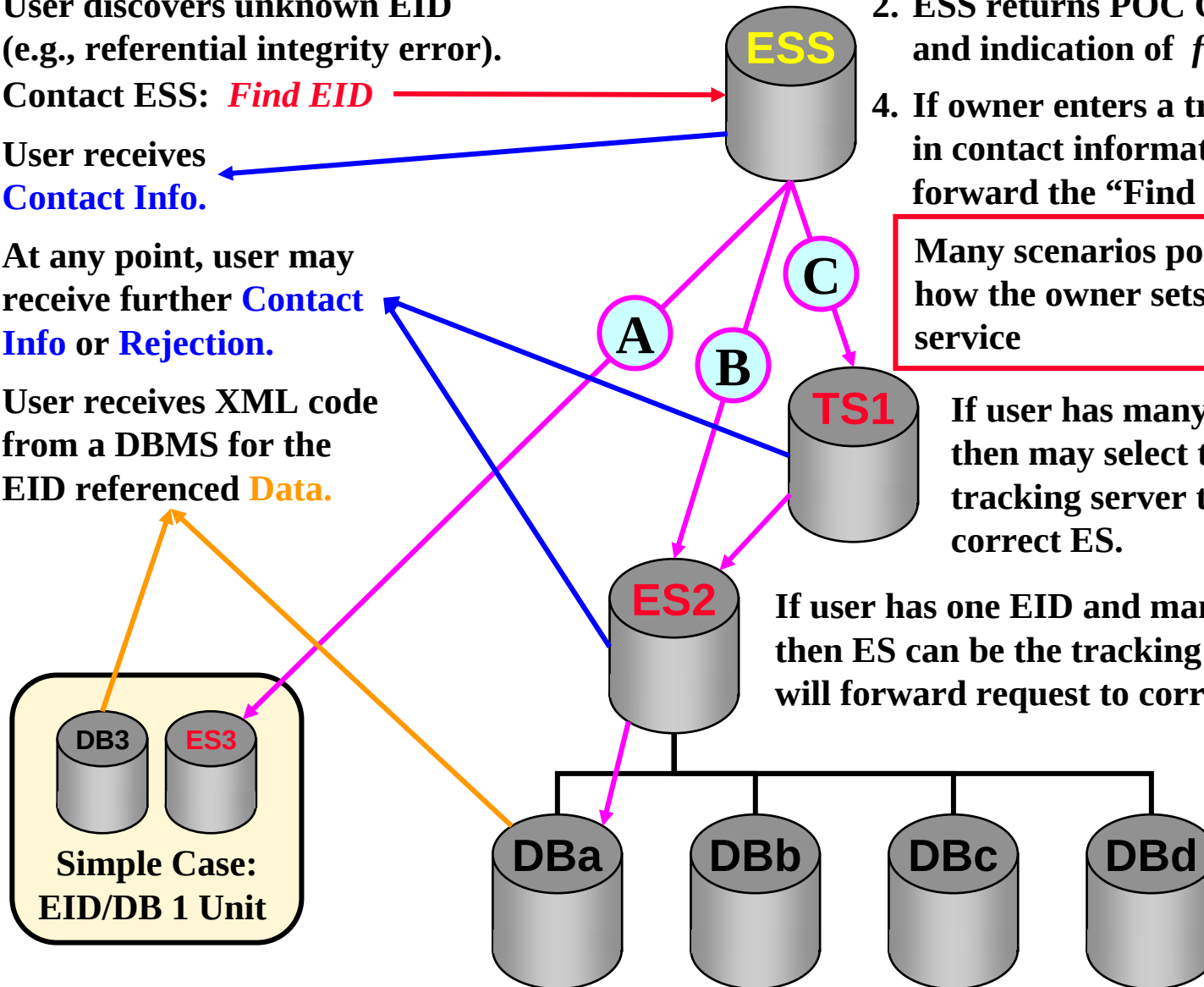
- 6. User receives XML code from a DBMS for the EID referenced Data.**

- 2. ESS returns POC Contact Information and indication of *finished* or *forwarded*.**
- 4. If owner enters a tracking service URL in contact information, then the ESS will forward the “Find EID” command.**

Many scenarios possible depending on how the owner sets up the tracking service

If user has many seeds and many ESs, then may select to implement a central tracking server to forward requests to correct ES.

If user has one EID and many DBMSs, then ES can be the tracking server and will forward request to correct database





EwID Summary

- Data identification will have to be accomplished somehow. This is but one of many possible techniques; the hard part is the task of selecting one.
- EwID is a Data Type, not an attribute (column) name
- Obtaining EwID Seeds is not intended to be a real-time process. This occurs when the systems are built and configured.
- EwID Seeds are free (see: <https://ess.arl.army.mil>)
- EwID characteristics & advantages:
 - No embedded information – they give away no information
 - Registration-based, this allows them to be compact & efficient (no waste)
 - Simple, fixed size – easy for software engineers to use
 - Easy to implement (add to legacy DBs as Alt Keys)
 - Data Miner's Dream – all data is tagged with a common structure



Some References

- [1] S. Chamberlain, G. Sprung (Joint Staff): “A Unifying Strategy for Data Integration for Global Force Management” *Proceedings of the 9th International Command and Control Research and Technology Symposium*; IDA (House of Engineers), Copenhagen, Denmark; 14-16 September 2004; Presented 14 June 2004. <http://www.arl.army.mil/~wildman/PAPERS/c2rt04.html>.
- [2] S. Chamberlain: “Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems,” *Proceedings of the 7th International Command and Control Research and Technology Symposium*; Quebec City, Quebec, Canada; 16-20 September 2002; Presented 17 September 2002; <http://www.arl.army.mil/~wildman/PAPERS/7thc2rt.html>.
- [3] “Application of Enterprise Identifiers (EID) To Automated Information Systems For Personnel,” *US Army Studies Program Report*, September 2002; http://www.arl.army.mil/~wildman/PAPERS/per_study.html.
- [4] S. Chamberlain: “An Enterprise Identifier Strategy for Global Naming Across Arbitrary C4I Systems,” *Proceedings of the 6th International Command & Control Research & Technology Symposium*, USNA, Annapolis, MD; 19 - 21 June 2001; http://www.arl.army.mil/~wildman/PAPERS/6th_Intl_01.pdf.
- [5] “Enterprise Identifiers For Logistics - An Approach in Support of Army Transformation Initiatives,” *US Army Studies Program Report*, December 2000; http://www.arl.army.mil/~wildman/PAPERS/log_study.html.
- [6] M. Boller, “Common Understanding for Transformation Brigades,” *Military Review*, Sep-Oct 2000. <http://www-cgsc.army.mil/milrev/English/SepOct00/boller.htm>.
- [7] T. Johnston: “Primary Key Reengineering Projects: The Problem;” *DM Review*, February, 2000, <http://www.dmreview.com/master.cfm?NavID=55&EdID=1866>.
- [8] T. Johnston: “Primary Key Reengineering Projects: The Solution;” *DM Review*, March, 2000, <http://www.dmreview.com/master.cfm?NavID=55&EdID=2004>.
- [9] T. Johnston: “De-Embedding Foreign Keys,” *DM Direct* (online):
Part 1, June 2, 2000. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2308.
Part 2, June 9, 2000. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2322.
Part 3, June 16, 2000. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2331.
Part 4, June 23, 2000. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2341.
- [10] S. Chamberlain: “Default Operational Representations of Military Organizations,” *Army Research Laboratory Technical Report: ARL-TR-2172*; February 2000. <http://www.arl.army.mil/~wildman/PAPERS/tr2172.html>.
- [11] M. Lonigro, Mike: “The Case for the Surrogate Key”, *Intelligent Enterprise Database Programming and Design On-line*, May 1998. <http://www.dbpd.com/vault/9805extra.htm>.