Title: Facilitating Flexible and Versatile Centrally Administered Community of Interest (CACOI) Organization and Control

Topic: C2 Concepts and Organizations

Authors: Aaron Hatcher, Kevin Foltz, and Coimbatore Chandersekaran

POC: Kevin Foltz

Name of Organization: Institute for Defense Analyses

Address: 4850 Mark Center Drive, Alexandria, VA 22311

Telephone: 703-845-6625

Fax: 703-845-6848

Email: kfoltz@ida.org

Facilitating Flexible and Versatile Centrally Administered Community of Interest (CACOI) Organization and Control

Abstract

In a Command and Control (C2) environment, we are seeing an increasing trend toward the dynamic creation of small networks with a focused purpose, commonly called communities of interest (COIs). As this trend continues we are likely to see many different combinations of systems assembled dynamically as needed. These systems will not just send data back and forth, but also share services and essentially act as one network to provide integrated and timely C2 functionality.

The Centrally Administered Community of Interest (CACOI) model was defined to cover many of the requirements of a COI in [1]. It was suggested that Microsoft .NET Server be used as the underlying operating system, specifically Windows Server 2000. The model had a few shortcomings that we seek to cover in this paper by switching to Windows Server 2003.

We propose enhancements to the CACOI model that allow the contributing communities to share data that remains under their local control and also to control the flow of data using Microsoft's Rights Management Server, a capability not present in any of the other COI models.

1 Introduction

We study collaboration among different organizations using the idea of a community of interest (COI). We consider not just sharing of data, but collaborations requiring tight integration of all elements of the collaborating organizations. Rapid assembly of collaborating partners is required, so these COIs must be established considerably more quickly and efficiently than setting up an entire new network. However, they must still be complete enough to provide the proper functionality and security measures to ensure safe interaction and sharing of resources.

There has been work in collaboration. The Department of Defense Goal Security Architecture (DGSA) [2] gives a high-level view of resource sharing over different physical networks. Peer-to-peer (P2P) networks have been studied as a simple way to share resources. In addition, some concerted efforts at collaboration have been undertaken.

The Secure Virtual Enclave (SVE) model [4] retains the primary features of a P2P COI. Resources remain under the control of the individual member domains, while each "enclave" contains architecture that maintains a list of its members. Members share these lists and use them to authenticate and authorize users from outside their local domain.

In a University of Maryland Model Dynamic Community (MDDC) [3], separate identity and authorization servers from multiple member domains combine for resource

access. Credentials are issued locally, but authorization credentials may be transferred from one domain to another. The MDDC model also includes a threshold cryptographic mechanism for joint administration, and a method for prompt removal of members from the community.

These models each address some requirements for a COI, but none fully satisfy them. The DGSA is not feasible in its current state, as there are no practical designs for implementation. The P2P, SVE, and MDDC models do not address confinement. The P2P and SVE models offer no internal controls for group-based management.

Some commercial DC models are available, but the sharing of complex resources such as databases and applications is difficult. These application level models are also subject to lower level attacks, which could undermine their security.

In [1], the idea of a Centrally Administered COI (CACOI) was introduced as a solution addressing many of the problems of previous COI models. Rather than join community members on a P2P basis, this model creates a new domain from the member domains, which allows the community administration to be centralized. In general, this model does not achieve the transparency of the P2P approaches, since users require a separate authentication for the COI. However, by encapsulating policy within the COI it offers a functionally richer model and simplifies several aspects of community operation.

In [1] the general issues of setting up COIs were discussed in depth along with detailed comparisons of various models. The rest of this section is a brief summary of the key COI requirements discussed in that paper and how the CACOI fulfills them.

We suggested using Microsoft .NET servers, specifically Windows Server 2000, to implement the CACOI model. The assumptions were that:
1. Many businesses are using Windows 2000-based operating systems, and
2. Active Directory (AD) and domain-based computing concepts allow for ready-made solutions to a lot of the COI issues.

With this, we were able to address the major requirements of a COI.

1.1 Secure Rapid Assembly

An effective COI model must be able to be created quickly and securely. We define "quick" setup to mean substantially faster than the creation of an organization's full network. If two large companies merge together, the setup for the final network may take a long time to plan out and implement. If it is essential that the companies are able to work together immediately, a COI could be created to allow for rapid data sharing and cooperation between critical departments.

The CACOI model uses the notion of a Windows Domain to satisfy this requirement. Setting up an isolated domain is very fast and, by use of Global Policy Objects (GPOs), the COI policy can be applied quickly and uniformly (once they have been decided upon

by participants). While the hardware required to set up the CACOI is a prohibitive factor, we suggest using a program like Virtual PC or Virtual Server to create the new Windows Domain Controller (DC) on simulated hardware.

This Virtual PC configuration can be saved so that you have a ready-made domain controller requiring minimal configuration when a new community must be set up, dramatically reducing setup time and complexity.

1.2 Data Isolation

The CACOI model dictates that a participating organization must copy over or move the shared data to the newly created Windows Domain. This very simple model enforces strict isolation; an organization puts what they want into the community and nothing else. Doing so does not grant any additional privileges to data that has not been uploaded and thus isolation is maintained.

1.3 Data Confinement

Because the data is all present in the domain representing the community, only a user with a login account in that domain can access the data. Since the data is in one place, we can also enforce access control and can audit resource usage, both successful and unsuccessful attempts.

2 Enhancements to the Data Sharing Model

2.1 Shortcomings of CACOI Data Sharing

One flaw of the CACOI model is the requirement that all shared data be either moved or copied to a central domain, or that the server holding the data must be joined to the domain. In either case, data must be moved. Moving or copying the data is a workable and fast solution for small or moderate amounts of data, but this solution does not scale well when there is a very large amount of data to be shared. If two research organizations are collaborating on a project and need to share the results of a few simulations, there may be terabytes of data involved. Joining their data servers to the CACOI domain is an option, but as it is currently implemented machines can only be a member of one Windows domain at a time. Users in the organization's domain that are not in the COI would lose access to that information, which is not desirable.

Also, when the data is shifted to the CACOI, the original organization loses control of their data. If the organization seeks to withdraw from the agreement they do not have the power to withdraw the resources they shared with the rest of the community. This may be viewed as a positive feature for the community as a whole but may be a very large negative for the individual organization who wants to leave the community. Organizations may not be willing to enter into the community unless they know that they can share their data yet retain the power to deny access to it at a later date. In the current model, once access is granted to that data by putting a copy in the CACOI it is *forever* granted. A simple example of this would be a large and sensitive database. A company might wish to allow queries to the database, but not access to the raw data itself.

A major requirement mentioned above was that a COI must be easily and quickly configurable and this requirement that data be copied may interfere with that. While we proposed

using Virtual PC or Virtual Server to get around the hardware cost of setting up a new domain, what about storage space for large amounts of data? It may still be necessary to purchase large amounts of storage equipment and join it to the domain, thus making the setup and configuration of a CACOI more expensive and potentially more complicated.

2.2 Strengths of CACOI Data Sharing

Whatever the flaws may have been, this shifting of data to a central location took care of many problems for us. Isolation was no longer a question; if an organization only wants specific pieces of information shared with the COI, they need only upload those specific pieces. It is impossible for users in other organizations, by virtue of being part of the COI, to gain access to any other data. This level of isolation is very desirable for organizations with proprietary or sensitive data that they need to control, especially organizations that may be in limited cooperation with a competitive rival of theirs. This level of separation in the CACOI model is something that we seek to maintain.

Once the data was part of the CACOI domain, we could exercise a lot of control over it. In Windows Server 2000 domains, a user is able to share specific folders with the rest of the domain, dictating which principals can access the data through an Access Control List (ACL). Also, the user could dictate exactly which permissions these principles had (read, write, execute, modify, etc). Principals can represent users or groups, allowing for either individual or role-based authorization policies. This level of granularity granted users a large amount of control in how they chose to share their data. A user could implement a policy like "Share this folder with these three people from my domain with read/write privileges and members of the 'Marketing' group with read-only privileges". Such controls can be applied by group policy within the CACOI so, once the data was all gathered into one domain, we could implement effective confinement.

2.3 Modifications to the Data Sharing Model

While the sharing model mentioned above is used for isolation and containment once the data has already been moved to the CACOI's domain, it may be possible to use this functionality to remove the *requirement* that data be moved in the first place. To do so requires that we move from Windows Server 2000 to Windows Server 2003.

Even in Windows 2000, this information sharing model can be extended across domain boundaries through trust relationships to encompass any principal in the domain's forest due to the distributed nature of AD. If the COI was created from separate domains within the same forest, we could relax the requirement that all data be moved into the new domain and simply require that access be granted to the data for principals in the CACOI. This way, the domain that originally had ownership of the data can maintain ownership yet grant access (possibly even read-only) to members of the COI.

The issue is that often the COI will encompass organizations that aren't part of one Windows forest. In this situation, Windows Server 2000 hits its limits. A user setting an ACL for a file can only name principals within that organization's forest; any other user or group names are semantically meaningless. For this reason, we suggested upgrading the operating system to Windows Server 2003.

Windows Server 2003 extends the functionality of Windows Server 2000 by allowing a subset of the Administrators in a forest (members of the Enterprise Admins group) to set up trusts that cross forest boundaries. In practice, this means a user in a Windows 2000 domain can name a principal in another domain within their forest because Windows Server automatically creates 2-way trust relationships between domains created within the same DNS tree (like

research.fake.org and marketing.fake.org—they share the same DNS suffix). These trust relationships allow authentication requests to flow between the DCs for our fictitious marketing and research domains (AD uses DNS to allow the various DCs to find each other). When a user tries to access a resource in the marketing domain using credentials from the research domain, the DC that handles authentication requests for marketing forwards the request to the DC in the research domain. If the research domain DC says that the person supplied valid credentials, the marketing domain can trust this information (even though it didn't authenticate the user itself) and move on to making an appropriate authorization decision. Since the domains are able to get an authentication decision for principals whose records they don't have locally, they can allow users to list those principals in their ACLs and enforce the policy.

When a CACOI is created, it likely will encompass multiple, separate organizations and thus will be created as an independent domain in its own forest. Windows 2000 limits trust relationships to one forest, so sharing data with the independent domain representing the CACOI must be done by copying or moving the data to its domain. Since Windows Server 2003 allows servers to extend trust relationships across forest lines, a participating organization can establish a cross-forest trust with the CACOI. Once this trust is established, the organization can grant access to principals in the CACOI with the same granularity it had in granting access to users within the domain. Since all members of the COI have been assigned accounts in the CACOI, the participating organization can, by granting access only to their CACOI accounts, avoid having to set up complicated webs of trust with each participating organization. Since a participating organization can grant access to a large amount of data by simply sharing that folder with restricted access, there is no need to copy very large amounts to the CACOI's domain minimizing the hardware costs and setup time.

Since the mechanism for sharing the data with the COI is the same mechanism that is used *within* the COI to share data, there would be no loss of access control or granularity. This is preferable to joining the data server holding the information to the domain because, if the entire server shifts forests, the users who are part of the home organization lose control of the data. Also, this would allow the participating organization to maintain control of its data. It can audit access from COI members locally and, if it chooses to leave the community, it could remove access to its data by simply severing the trust relationship with the CACOI's domain.

Sharing the data through a cross-forest trust with the CACOI is not the only option, just an additional one. An organization can move or copy groups of files over to the CACOI's domain and share others. This allows the organization to judge the amount of control it needs to retain on a given datum and the only options are no longer "share completely" or "don't share". Now an organization can choose to share some files fully and permanently by copying them over and grant access to some more sensitive data yet retain more control over it. In the case where an organization shares data that may frequently change, sharing the local copy will ensure that the view the COI has is always up to date. The alternative is for an organization to create snapshots of the changing data and continuously upload them to the CACOI server. This may or may not be feasible and would certainly cost more bandwidth.

3 The Controlled Dissemination of Data within a COI

Our updated CACOI model uses Windows Server 2003 which introduces another possibility—data flow control using Microsoft's Rights Management Server. To see why this is desirable, we should point out the shortcomings of the current models.

If an organization wanted to participate in a COI (any model) it would likely share resources, possibly just a few Word documents, with select members of the other organizations.

If the organization has very sensitive data, it may be important that only the specified people see the documents. Isolation will ensure that only the documents that want to be shared are shared, and confinement can ensure that only the people who are supposed to have access to the document can access it.

3.1 A Hole in the System

This may sound like a solved problem, but this is not technically the case—"confinement", as implemented in most COI models, only ensures that the resource will be accessed from *a given server* by the people who are allowed to access it. There is no protection against the person passing the document to someone else by copying it onto a memory stick or simply emailing the file over the internet. Also, a compromised computer (worm/virus) could make very sensitive data available to people who aren't meant to see it.

If we encrypt the documents we can ensure that, in the case of a worm, the data will be useless to those not authorized to view it. However, there is a problem with just encrypting a document; if the document is intended to be shared with someone, they're going to need to be handed a decrypted form of the document or the decryption key (so that they can *produce* a decrypted form). Once they have the decrypted document, nothing can stop them from passing that on to others.

To gain true confinements, what is needed are "sticky rights", rights that are passed with the document and enforced at every access attempt. The document should be encrypted so that the file by itself is not useful and decryption should occur in a controlled manner. By using a trusted program, we can also pass along privileges with the document (stating that it may or may not be copied, printed, saved, etc) which now severely limits the ability of a malicious insider to pass on the information to an unintended source.

This is concept behind RMS. When a document is to be protected, it should be encrypted so that nobody can read it without the right key. To determine who has the rights to read it, an XML (specifically XrML) tag should be permanently attached that records the creator and the privileges they dictate to various principals. In the next section we cover how it affects common usage.

3.2 Controlling Data Flow Control in the CACOI Using RMS

To get RMS functioning in a new domain, an administrator needs to install the RMS Server on a member server. Once the RMS Server is configured and provisioned, any computer that's going to connect to the CACOI need only have RMS aware applications, like Microsoft Outlook or Office, and install the RMS client software. As we're using Windows Server 2003 as our server and assuming Windows 2000 or XP as the client machines, the assumption that Outlook and Office will be used is a small one, and not even fully necessary. Any RMS enabled word processor and email client can stand in their place but, as RMS is a new technology, there are currently few RMS enabled applications to choose from.

While a Microsoft Exchange Server is not necessary for RMS to function, it is necessary to send and protect email. Also, installation of the Exchange Server creates the Global Address List (GAL) that RMS can use to make naming principals easier. Thus, we recommend installation of an Exchange Server in the CACOI domain.

It is worth noting that producing and sharing unprotected documents is not hindered in any way. When the RMS client software is installed, Word continues to make unprotected documents by default. The noticeable change is that a button (and menu item) for permissions becomes active and, if you click on it, your account and your machine automatically contact the RMS Server, register themselves, and gain the keys necessary to produce protected data.

The protected file can now be passed around like any other file but is not useful to someone who is not authorized to view it. If you attempt to open it in Word, Word will display an error message telling you that you don't have permission to view the file and tell you how to contact the author of the content. Attempting to bypass the restrictions by viewing it in a hex editor, you can see the XrML header that specifies the location of the RMS server that protected the document, the owner, the public key of the server, and lots of other metadata. However, you cannot view the actual contents of the document because it is encrypted. The XrML header includes a digital signature as well so attempting to simply make yourself the document's owner by changing fields in the header won't work either.

Flow control can be applied not just to documents that may be sent off via email but to emails themselves because sometimes sensitive information is present in the contents of an email. It is possible to compose an email in Outlook, list the recipients from the GAL, and dictate that they cannot forward (or print or copy) the email to others. They will not be able to open the protected email until they've been authenticated to the RMS server (usually occurs automatically) and Outlook will enforce the distribution policy once the email has been decrypted.

This added layer of protection to documents and emails adds effective flow control to the CACOI model so that true containment is possible. We hope that this extra protection will make organizations that have sensitive or proprietary material more likely to share it with other members of the COI.

3.3 Possible Configurations

In the domain of our CACOI, we suggest placing an Exchange Server and also an RMS server to allow for document protection. When a participant creates a document that they want protected, they can create it using their account from the CACOI and apply RMS protections to it. As they can specifically name the principals from the CACOI that they want to grant access to and exactly which privileges they seek to grant, this creates a very powerful setup.

However, it may be the case that organizations usually produce documents in their own domain and later send them to the CACOI. If this is the case, the user may have to send a copy of the document unprotected, open it with the account the user has there, and then apply protections. While this is possible, it does not keep the document protected the entire time; there is a window in which someone else could access the file without restrictions. If even such a small window is unacceptable, there are other configuration options available.

RMS, like Windows Server 2003 itself, has the ability to create trusts with other RMS servers that cross the Windows forest boundary. This means that an organization that already has an RMS server present in its home domain can set up a trust with the RMS server in the CACOI. This would allow the user to create the document using the account in their home domain, add principals from the CACOI to the ACL, and then send it to the CACOI fully protected. All users or groups within the COI can be granted privileges without separately having to establish trusts between every home domain involved.

4 Conclusion

The updated CACOI Model using Windows Server 2003 not only enhances the capabilities of the original but adds features not present in any of the other models. Due to the addition of cross-forest trusts, organizations not have the option to share data with the COI using the CACOI model but without having to copy or move the data over. An organization would be able to maintain full control over its data because it has the ability to sever the trust relationship at any time.

The presence of technology that can apply and enforce “sticky rights” in Windows Server 2003 also allows us to enforce true confinement and controlled data dissemination. Sharing data with the COI need not be as much of a risk, which hopefully will make organizations with highly sensitive data (proprietary, classified, etc) more comfortable taking part in such communities.

5 Future Work

We plan to review the robustness and scalability of the architecture and the richness of the policies. As an example, when sharing data with a COI through Cross-Forest trusts, it is not clear who owns the auditing and other metadata associated with the resource and access attempts. This gets more complicated as the number of COIs and organizations grows. We’d also like to perform an analysis of how these new modifications (organizations being able to withdraw their data, etc) affect the rules and dynamics of the community.

6 References

[1] Kevin Foltz, Coimbatore Chandersekaran, "Sharing Resources through Dynamic Communities," in Proceedings of The Tenth International Command and Control Research and Technology Symposium (ICCRTS 2005), Falls Church, VA, June 2005.

[2] Department of Defense (DoD) Goal Security Architecture (DGSA), Center for Information Systems Security (CISS), Defense Information System Security Program (DISSP), Version 3.0, 30 September 1995.

[3] H. Khurana, “Negotiation and Management of Coalition Resources,” Ph.D. Thesis, 2002, Department of Electrical and Computer Engineering, University of Maryland, College Park, MD.

[4] D. Shands, R. Yee, J. Jacobs, E. J. Sebes, “Secure Virtual Enclaves: Supporting Coalition Use of Distributed Application Technologies,” Proceedings of the Network and Distributed Systems Security Symposium, San Diego, February 2000.

Appendix

Included here are diagrams illustrating the changes to our CACOI model.

Original CACOI Structure

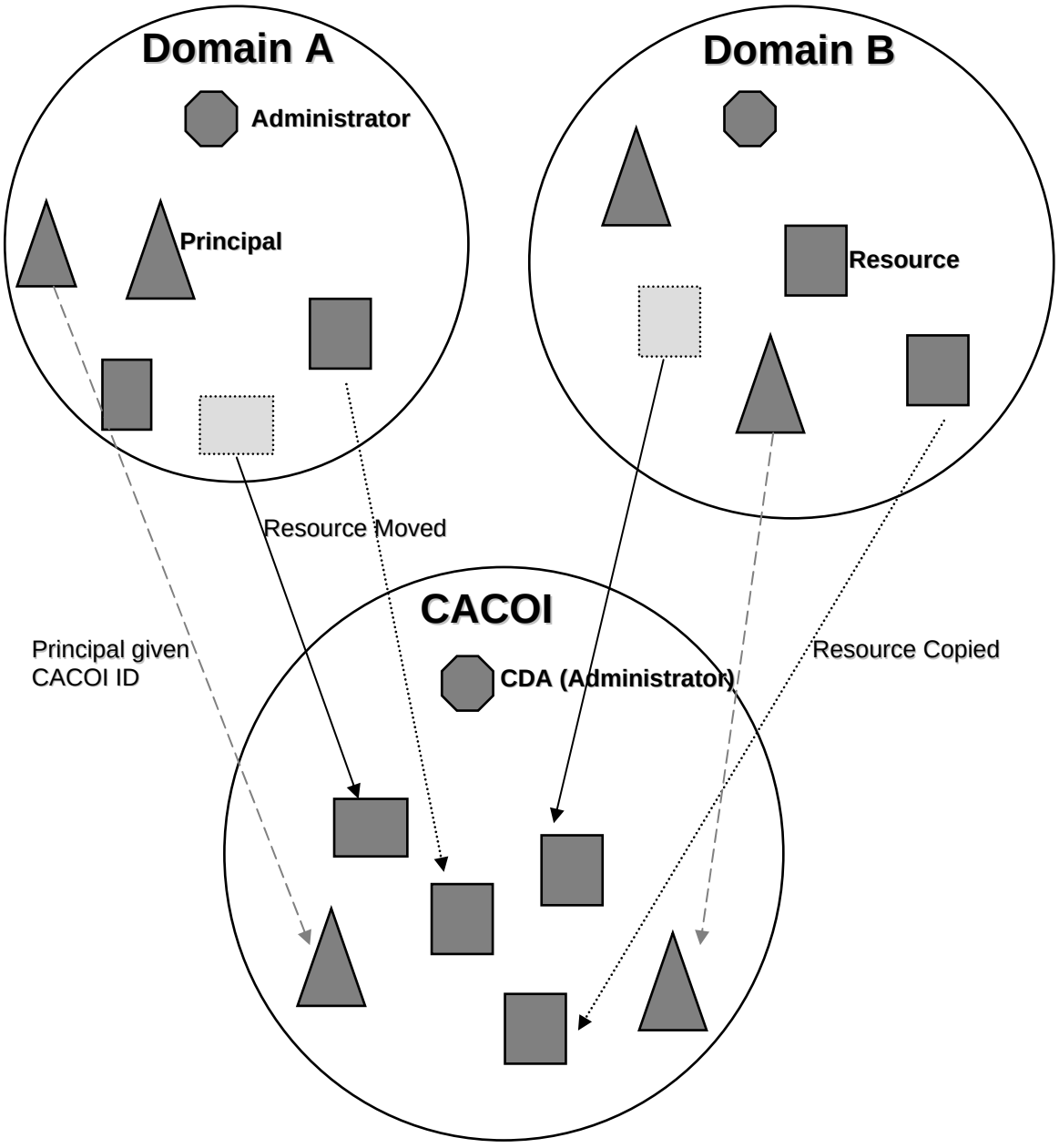

Figure 1: Structure of a CACOI. Note that all community resources are moved or copied into the community domain, and principals have new identities in the CACOI.

Modified CACOI Structure

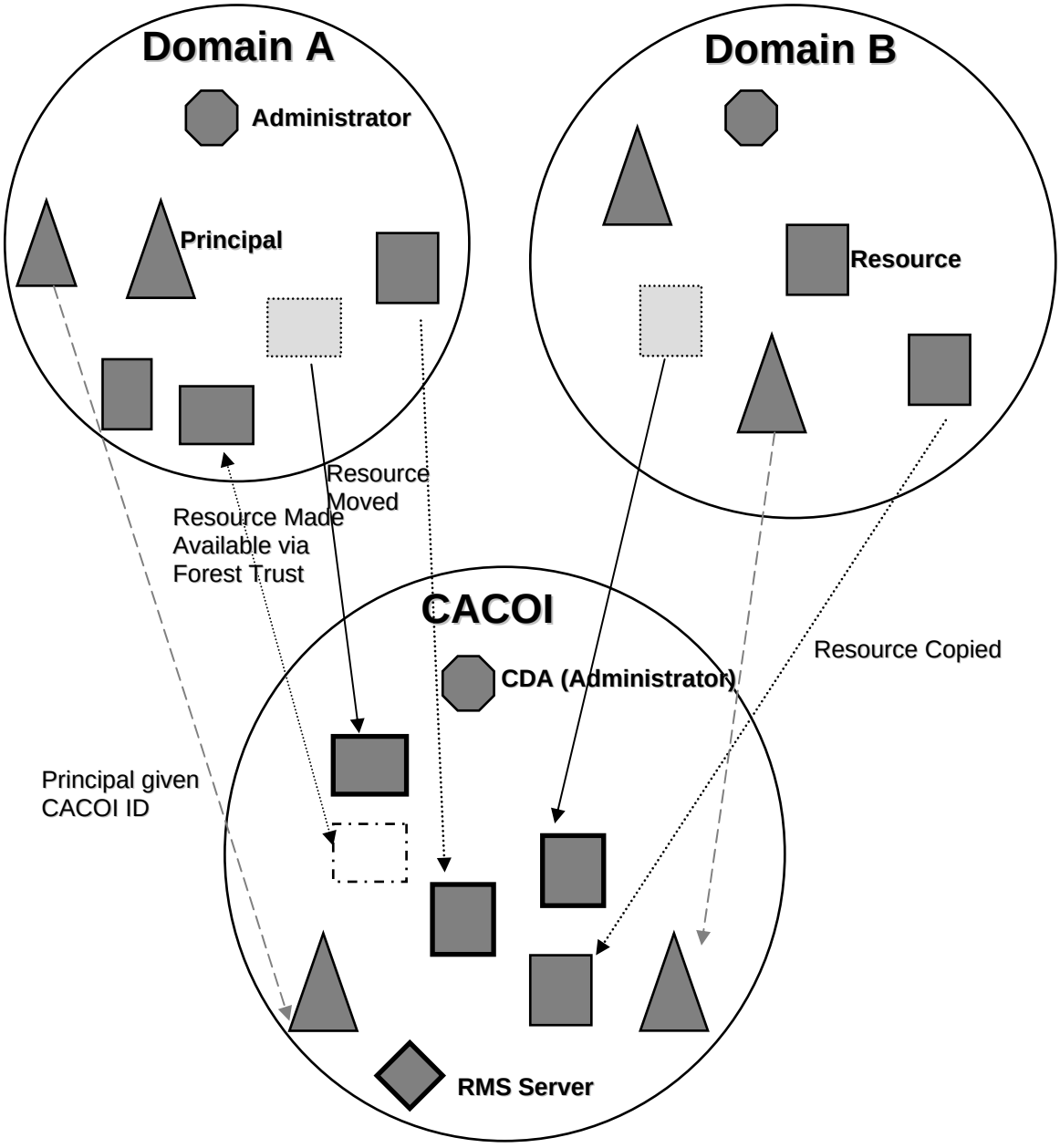

Figure 2: The updated CACOI. Note that some resources are made accessible to the community but remain in their original domains. Resources with dark blocks are RMS protected