Architecture

Microsoft Dynamics CRM 4.0

The Microsoft Dynamics CRM Security Model

White Paper: “Nuts and Bolts” Series
Security and Authentication in Microsoft Dynamics CRM

Core Architecture

Date: September 2009

Microsoft Dynamics

Acknowledgements

Initiated by the Microsoft Dynamics CRM *Engineering for Enterprise* (MS CRM E^2) Team, this document was developed with support from across the organization and in direct collaboration with the following:

Key Contributors	**Technical Reviewers**
Roger Gilchrist (*Microsoft*)	Michael Ott (*Microsoft*)
Mahesh Hariharan (*Microsoft*)	Monika Borgaonkar (*Microsoft*)
	Tommy Su (*Microsoft*)
	Amy Langlois (*Microsoft*)

The MS CRM E^2 Team recognizes their efforts in helping to ensure delivery of an accurate and comprehensive technical resource in support of the broader CRM community.

Note: This paper leverages and integrates content from the Microsoft Dynamics CRM 4.0 SDK and the Microsoft Official Training course, *Administration in Microsoft Dynamics® CRM 4.0*.

MS CRM E^2 Contributors

Amir Jafri, Program Manager	Jim Toland, Content Project Manager

Feedback

Please send comments or suggestions about this document to the MS CRM E^2 Team feedback alias (entfeed@microsoft.com).

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989
Worldwide +1-701-281-6500
www.microsoft.com/dynamics

Legal Notice

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Dynamics, Microsoft Dynamics Logo, Active Directory, Excel, SharePoint, Outlook, and the Outlook Launch Icon are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

Preface

CRM E^2 Nuts and Bolts Series Overview

The MS CRM Engineering for Enterprise (E^2) *Nuts and Bolts* (NB) series is an expanding set of topical content, with each offering providing detailed information about the internal mechanisms related to a specific area of functionality within Microsoft Dynamics CRM 4.0.

NB Series offerings are designed to provide detailed technical resources that:

- Address often repeated queries to Technical aliases
- Consolidate answers, links, etc., that are generated in response to those queries
- Offer multiple levels of complementary information to support a broader, multi-perspective understanding of the topic
- Convey the baseline "principles" users require to begin to address related but tangential technical queries
- Present content using a consistent structure and "look and feel"

Audience

The target audience of the NB Series includes (but is not limited to):

- Solution Architects
- Application Architects
- Infrastructure Architects
- Consultants
- Developers

NB Article Content and Structure

Articles in the NB Series are designed to accommodate information at three independent but complementary levels (or "tiers"), which are shown in the following table:

Tier	Description
Core Architecture	High-level, architectural information; "schematic-level " view of functionality; provides contextual overview/baseline knowledge
Conceptual Application	Best practices and guidelines associated with CRM features or functionality that can be applied based on the specifics of particular implementation
Practical Application	Detailed explanations about how to address unique scenarios; practical details about resolving issues or accomplishing specific "real-world" tasks

This component of the Nuts and Bolts article *Security and Authentication in Microsoft Dynamics CRM* addresses the core architectural aspects of the Microsoft Dynamics CRM 4.0 security model.

The full breadth of coverage provided by the Nuts and Bolts article *Security and Authentication in Microsoft Dynamics CRM* includes the following:

- Core Architecture
    - *The Microsoft Dynamics CRM Security Model*
- Conceptual Application
    - *Securing Microsoft Dynamics CRM in the Enterprise*
    - *Field-level Security in Microsoft Dynamics CRM: Options and Constraints*
    - *Securing the Network Infrastructure for Microsoft Dynamics CRM*
- Practical Application
    - *Security Contexts in Microsoft Dynamics CRM*
    - *Connectivity and Firewall Port Requirements in On-Premise Deployments*

Overview

Designed to protect data integrity and privacy and to support efficient data access and collaboration, the Microsoft Dynamics CRM 4.0 security model:

- Supports a licensing model for users.
- Provides users with access only to the information that they require to do their jobs
- Categorizes types of users by role and restrict access based on those roles.
- Prevents users from accessing objects that they do not own or share.
- Supports data sharing by providing the ability to grant users with access to objects that they do not own to participate in a specified collaborative effort.

To provide context for a deeper understanding of the Microsoft Dynamics CRM 4.0 security model and how it functions to protect data integrity and privacy and to support efficient data access and collaboration, this paper provides an overview of the typical interaction scenarios associated with a CRM implementation, as well as the interaction points through which this access occurs.

This paper also provides detail about the components of the security model and their function, as well as how the components work together to accomplish the model's overall goals. This detail includes more information about the following Microsoft Dynamics CRM 4.0 topics:

- Authentication
  - Authentication methods
  - Authentication flows
  - Correlation to CRM deployment models
- Authorization
  - Role-based Security
    - Roles
    - Privileges
    - Access Levels
  - Object-based Security
    - Access rights
    - Create access
    - Sharing objects
    - Assigning objects

How Users and Services Interact with Microsoft Dynamics CRM 4.0

To provide context for a better understanding of the Microsoft Dynamics CRM 4.0 security model, it is important to be familiar with the variety of scenarios in which users and services typically interact with a CRM implementation during the course of normal business operations.

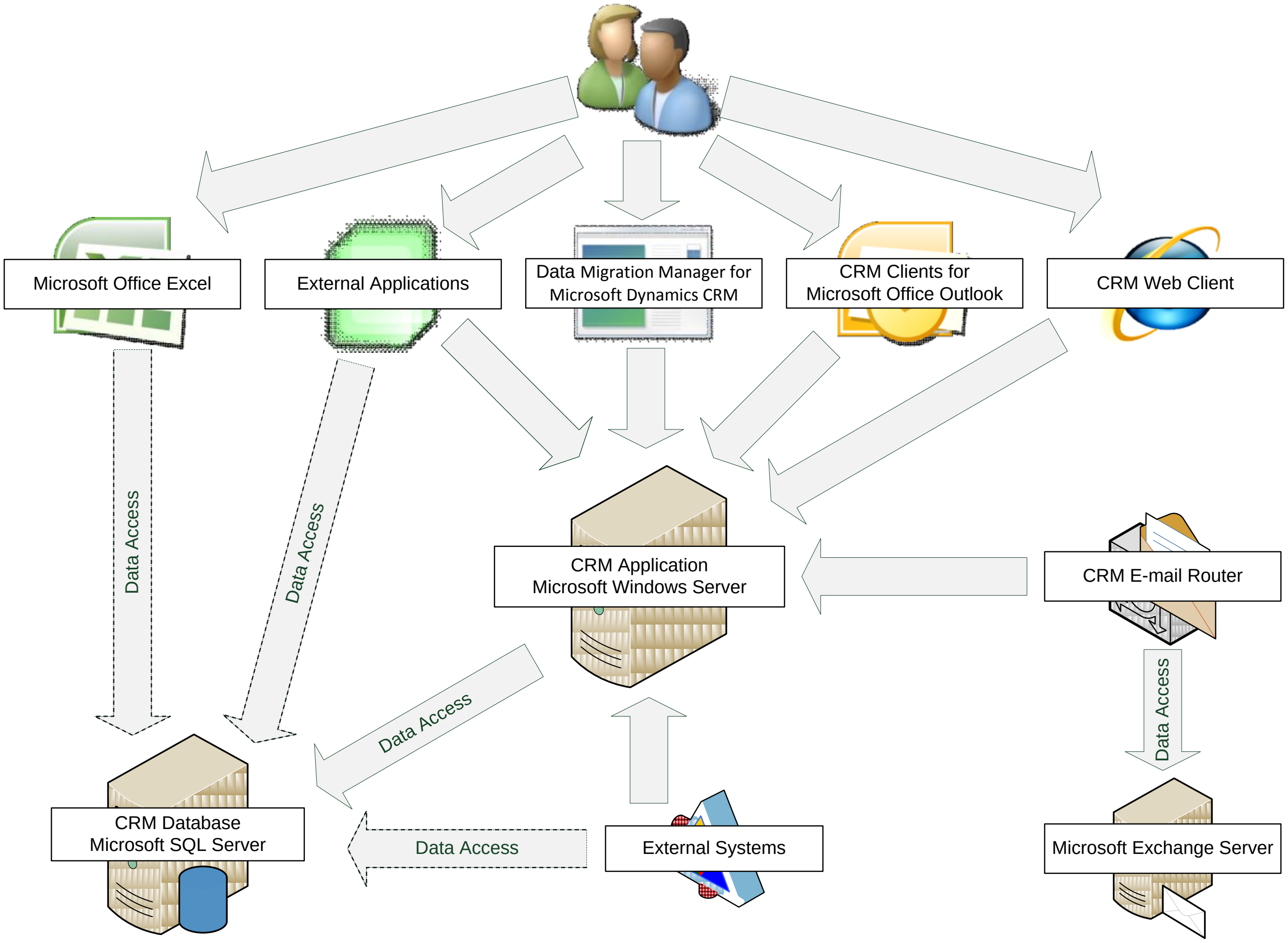
For Microsoft Dynamics CRM 4.0, these *interaction scenarios* commonly include:

- Users accessing the CRM application in their day-to-day interactions with customers and to generate reports displaying CRM data
- External systems or services accessing the CRM application, database, and SDK to retrieve data or to call functionality

From a security perspective, however, it is probably even more important to understand the related *interaction points*, or access methods and channels, that are provided by Microsoft Dynamics CRM 4.0 to enable these scenarios.

Interaction Scenarios

At a high level, external users and services typically interact with a CRM deployment in a variety of scenarios, which are depicted in the following graphic:



Note: In the graphic, Data Access arrows with dotted borders indicate access paths that are only available in a subset of Microsoft Dynamics CRM 4.0 deployment models.

- **Users**. Users access the CRM application by using the CRM Outlook clients, the CRM Web client, the CRM Mobile client, Excel, or other applications. Typically, these types of interactions reflect individual users that are accessing the CRM application to:
  - Store and retrieve customer or account data
  - Create reports that display CRM data

  In addition, in on-premise deployments, end users can access information in the CRM database by using Filtered Views.

  Note: Microsoft Dynamics CRM Online does not support access via Filtered Views.
- **External User Applications**. External, third-party (ISV) applications, such as rich clients, can also interact with the CRM platform. These applications are typically designed to provide individual users with access to CRM information they need to perform their day-to-day job functions.
- **Microsoft Office Excel**. Microsoft Dynamics CRM provides the Export-to-Excel feature, which allows users to export and access data by using Microsoft Office Excel. This approach provides end users with access to data in a format that they can manipulate by using the Excel application.
- **External Systems**. External systems, such as SharePoint or ERP applications, and connectors, such as the Microsoft BizTalk Server Adapter for Microsoft Dynamics CRM, can also interact with the CRM platform, which is typical for scenarios in which Microsoft Dynamics CRM is integrated with another system sharing data.

  In this scenario, access may be performed directly by an end user of the system, but access may also be performed by a service account retrieving data collectively and managing security externally to CRM.
- **Exchange Integration**. Microsoft Dynamics CRM also integrates with Microsoft Exchange Server, which represents yet another potential interaction scenario.
- **Data Migration Manager for Microsoft Dynamics CRM**. The Data Migration Manager for Microsoft Dynamics CRM is designed to convert and upload data from another CRM system to Microsoft Dynamics CRM. The Data Migration Manager includes default data maps to convert source data from several common CRM systems to the data format expected by Microsoft Dynamics CRM.

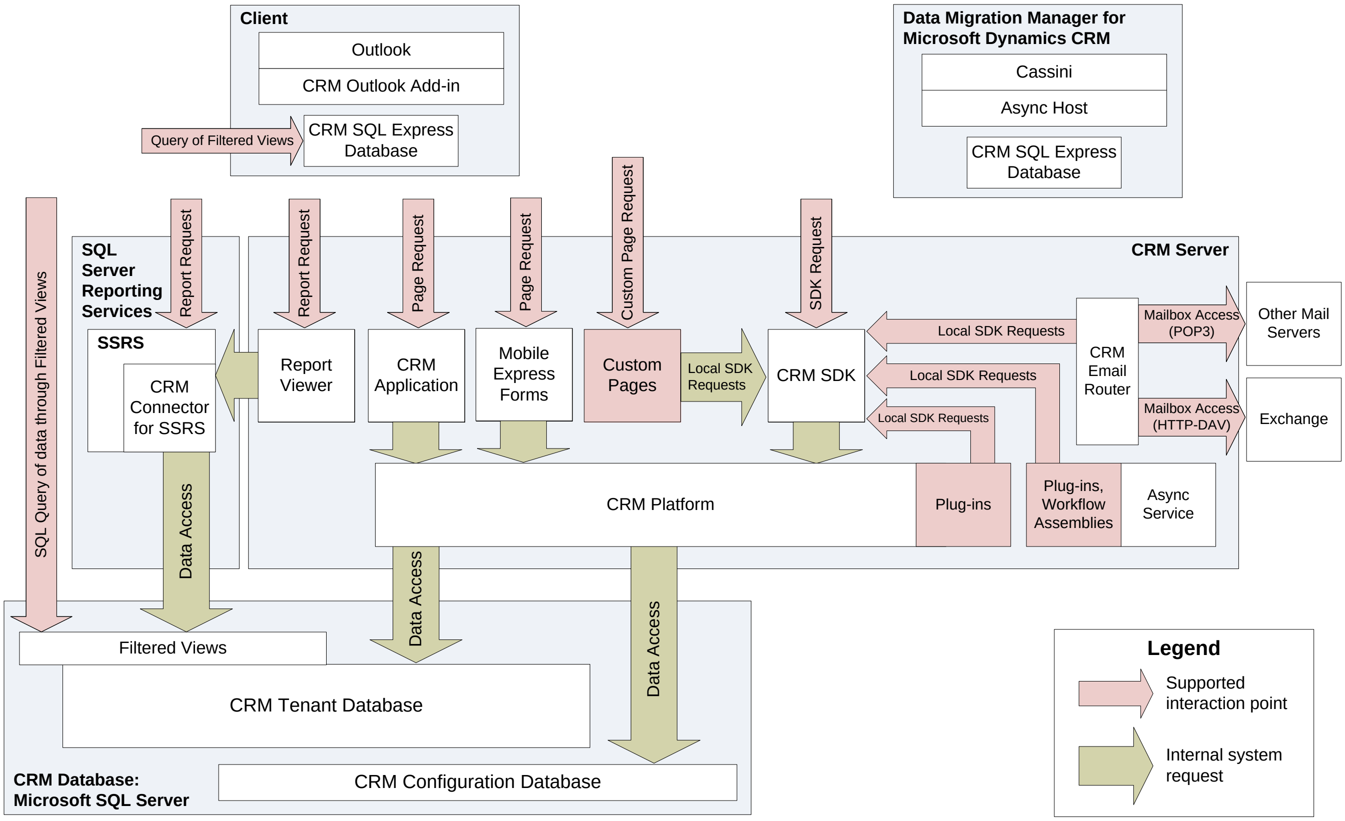
From a systems perspective, these interaction scenarios can be consolidated by considering their associated interaction points, because each interaction scenario relies on a specific:

- Method (or mechanism) for accessing the platform
- Channel or interface point through which that access occurs

Interaction Points

Microsoft Dynamics CRM 4.0 provides a number of access methods and channels that external users and systems leverage to interact with the CRM platform. Each interaction point provides access control in a way that allows for consistent treatment of data visibility while at the same time maintaining the flexibility required to implement complex situations as necessary.

The following diagram shows the key access methods and channels for interacting with the Dynamics CRM platform:



The interaction points that are associated with Microsoft Dynamics CRM 4.0 are described in the following table.

Access Channel	Access Method	Description
CRM Application	CRM Web Client	A browser interface that enables end users to interact with the CRM application, or with the CRM Report Proxy when generating reports.
	CRM Outlook Client (online and offline)	An Outlook-integrated interface that enables end users to interact with the CRM application, or with the CRM Report Proxy when generating reports.
	CRM E-Mail Router	The E-mail Router is an interface between the Microsoft Dynamics CRM system and one or more Microsoft Exchange or POP3 servers for incoming e-mail, and one or more SMTP servers for outgoing e-mail. E-mail messages come into the Microsoft Dynamics CRM system through the E-mail Router.
	Mobile Express	A set of web pages hosted in a sub-directory of the CRM Application targeted at a mobile browser client

Access Channel	Access Method	Description
CRM SDK Web Service	Plug-ins, workflow assemblies	Plug-ins and workflow assemblies can make requests of the CRM platform directly by using the CRM SDK, for example to get data from another system and update Dynamics CRM when an entity instance is created or updated.
	External applications	External applications can make requests of the CRM platform directly by using the SDK, for example to surface data from or push new data into the CRM system, as is the case with the Microsoft Dynamics CRM to Dynamics GP Connector.
	Data Migration Manager for Microsoft Dynamics CRM	This utility assists users with converting and uploading data from another customer relationship management system to Microsoft Dynamics CRM.
	Custom pages	Certain implementations require that developers use custom pages in the ISV web folder to make SDK calls to modify or retrieve data. **Note**: For more information about custom pages accessing the CRM SDK Web Service, see the paper *Security Contexts in Microsoft Dynamics CRM*.
Filtered Views (CRM Database)	External applications	External applications may require access to the CRM database to perform bulk data retrieval; typical scenarios include: ▪ Leveraging database access as an integration mechanism (e.g. SQL Server Integration Services), which would be more difficult to accomplish by using the CRM SDK Web services ▪ Requiring access to volumes of CRM data, e.g. when using SQL Server Reporting Services to display a CRM report **Important**: Accessing the tables underlying the CRM database is not supported.
	CRM Export-to-Excel feature	A feature that enables end users to export CRM data to Microsoft Office Excel. The functionality involves the CRM system downloading a worksheet and then Microsoft Excel connecting directly to the database to retrieve the necessary data by using the Filtered Views exposed by CRM.

Important: CRM Online supports accessing the CRM SDK Web Service via custom pages, but it does not support hosting those pages on the CRM Server directly. For CRM Online deployments, consider hosting custom pages in Azure or any third-party data center.

Note: CRM Online does not support access to Filtered Views of the CRM Database via external applications. In addition, service provider hosted deployments typically do not support direct access to Filtered Views outside of the service provider’s firewall.

How the CRM Security Model Works

The two major aspects of a security model are authentication and authorization. While these terms are frequently interchanged, they have distinct meanings.

- *Authentication* is the process of determining if a user is who he or she claims to be. Authentication is accomplished by using a well known Identity (username) and a secret, something only the user knows (e.g. password). If the user is authenticated, the system grants the user access to the extent specified in the permission list for that user.

  Note: It is possible for one user account (A) to execute code to perform some task on behalf of another account (B) by using a technique referred to as impersonation. For additional information, see the *Authentication via Impersonation* section of this document.

- *Authorization* is the right granted to a user (or group of users) to access the system and the data stored on it. In other words, the authorization process determines which data a user can access. Authorization is typically defined by out-of-the-box designed rules and can be fine tuned by privileged users.

Important: The term "user" above refers to individuals as well as to external systems and services that potentially interact with a Microsoft Dynamics CRM implementation. For additional information about the Microsoft Dynamics CRM 4.0 security model, in the MSDN Library, see *Microsoft Dynamics CRM SDK* at http://msdn.microsoft.com/en-us/library/bb928212.aspx

Authentication

As mentioned previously, the authentication process determines if users are who they claim to be. Microsoft Dynamics CRM 4.0 leverages multiple forms of authentication to accommodate the various, supported deployment models.

Note: For additional detail about authentication in Microsoft Dynamics CRM 4.0, on the Microsoft Dynamics CRM Team blog, see the entry *CRM Authentication* at: http://blogs.msdn.com/crm/archive/2009/06/10/crm-authentication.aspx

Authentication Methods

Microsoft Dynamics CRM 4.0 supports a variety of authentication methods to accommodate the various CRM deployment models. Each authentication method is applicable to one or more CRM deployment models, as shown in the following table:

Model	Connection Type	Authentication Method	Identity
On Premise	Corporate Network	Integrated Windows Authentication	Active Directory
	Internet-facing	Forms-based Authentication	Active Directory
Hosted	Shared/Trusted Domain Network e.g. via VPN	Integrated Windows Authentication	Active Directory
	Internet-facing	Forms-based Authentication	Active Directory
CRM Online	Internet	Windows Live ID Authentication	Windows Live ID

The authentication methods supported by Microsoft Dynamics CRM 4.0 are summarized in the following sections.

Integrated Windows Authentication

With IWA, Microsoft Dynamics CRM uses standard interaction between Internet Explorer and Internet Information Server (IIS) to negotiate and authenticate a user's Active Directory Identity. This can occur using either NTLM or Kerberos, depending on the environment setup. While both security mechanisms can determine a user's identity, Kerberos offers greater security and also can be useful for delegating authentication in certain integration scenarios.

Forms-based Authentication

Similar to IWA, Forms-based Authentication requires that users have Active Directory accounts within the CRM AD forest. However, rather than automatic negotiation of credentials between IE and IIS, users are prompted with a logon form to enter their credentials directly.

Windows Live ID Authentication

The Windows Live ID service is designed to manage identity and trust within the Windows Live ecosystem. Windows Live ID provides a single-sign on experience that allows businesses and customers to use a single set of credentials (logon name and password) for accessing various Web sites or Web applications. With Windows Live ID authentication, CRM Online users navigating to http://crm.dynamics.com and choosing to log on are prompted to provide their Windows Live ID credentials.

Note: For more information about Windows Live ID Authentication, on MSDN, see the article *Introduction to Windows Live ID* at: http://msdn.microsoft.com/en-us/library/bb288408.aspx

Alternative Authentication Scenarios

By default, on-premise implementations of Microsoft Dynamics CRM 4.0 leverage Integrated Windows Authentication to accommodate access by internal users. However, many businesses also require the ability to provide external users with access to the highly sensitive information that is stored in the CRM system and to accommodate this access without having to create Active Directory trusts. Microsoft Dynamics CRM 4.0 can be configured with Microsoft Dynamics CRM 4.0 Intelligent Application Gateway (IAG) 2007 SP2 to support alternative authentication scenarios by using multi-factor authentication and federated authentication.

Important: Implementing multi-factor and federated authentication solutions is a complex task that should only be performed by experienced IT administrators. For more information about using IAG with Microsoft Dynamics CRM 4.0, see the white paper *Implementing an ADFS Solution for Microsoft Dynamics CRM by Using Intelligent Application Gateway IAG)* at: https://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=47ee7f73-6059-4b20-a305-1b8b2b23f0e9

Multi-factor Authentication

Multi-factor authentication refers to a compound implementation of two or more classes of human-authentication factors, including something:

- *Known to only the user*—Knowledge-based (for example, password, pass phrase, shared secrets, account details and transaction history, PIN, and so on).
- *Held by only the user*—Possession-based (for example, security token, smart card, shared soft tokens, mobile device, and so on).
- *Inherent to only the user*—Biological or behavior biometric traits (for example, facial recognition, fingerprint, voice recognition, keystroke dynamics, signature, and so on).

For example, many enterprise extranet/VPN solutions require both simple credentials (such as an ID and password) and hardware tokens (such as secure ID with time-based one-time password generators, smart cards that use embedded PKI solutions, etc.) to gain access. Combining "known" and "held" creates a multi-factor authentication method that significantly improves authentication strength by reducing the threat of stolen digital identities.

Federated Authentication

Federation is a trust-based agreement between two organizations with some common purpose, such that both want authentication assertions from one organization to be recognized by the other organization. As mentioned earlier, federation involves two parties; 1) the identity provider authenticates users' identity accounts so that those users can access resources in third-party networks, while 2) the resource provider permits identities authenticated by an identity provider to access resources in its network.

A *federated identity relationship* is a standards-based arrangement between organizations in which user claims from one organization are passed to and recognized by another. With federated authentication, users can therefore sign in to (and be authenticated by) the organization that manages their identity account—and then have their authentication information passed to a federated partner as needed without requiring another sign on.

Note: A federated partner that recognizes the identity provider's users and grants them access to its resources is called a resource provider.

For example, with Microsoft Dynamics CRM 4.0, developers can use a combination of IAG and Active Directory Federation Services (ADFS) to establish an authentication gateway and provide a federated authentication solution.

Authentication Flows

Regardless of the deployment model or authentication method used in an implementation, authentication occurs in one of three basic flows, as described in the following table.

Authentication Flow	Description
End user to CRM Application	Scenario involving authentication of a client application (browser application, Outlook client, console application, or a Windows Forms application) with the Microsoft Dynamics CRM application server.
SDK Client to CRM Web Service	Scenario involving authentication of a non-Microsoft Dynamics CRM (ISV) Web service, Windows service, or ASPX page that requires access to the Microsoft Dynamics CRM Web services. This type of authentication scenario can be performed in the background without user interaction, or it can be initiated by a user.
SDK Client to CRM Web Service with Impersonation	Similar to the SDK Client to Web Service authentication flow, this scenario does not require users to interact by specifying logon information in a form. In this scenario, all calls to the CRM platform are performed on a user's behalf by using impersonation. **Note**: For additional information about impersonation, see the *Authentication via Impersonation* section of this document.

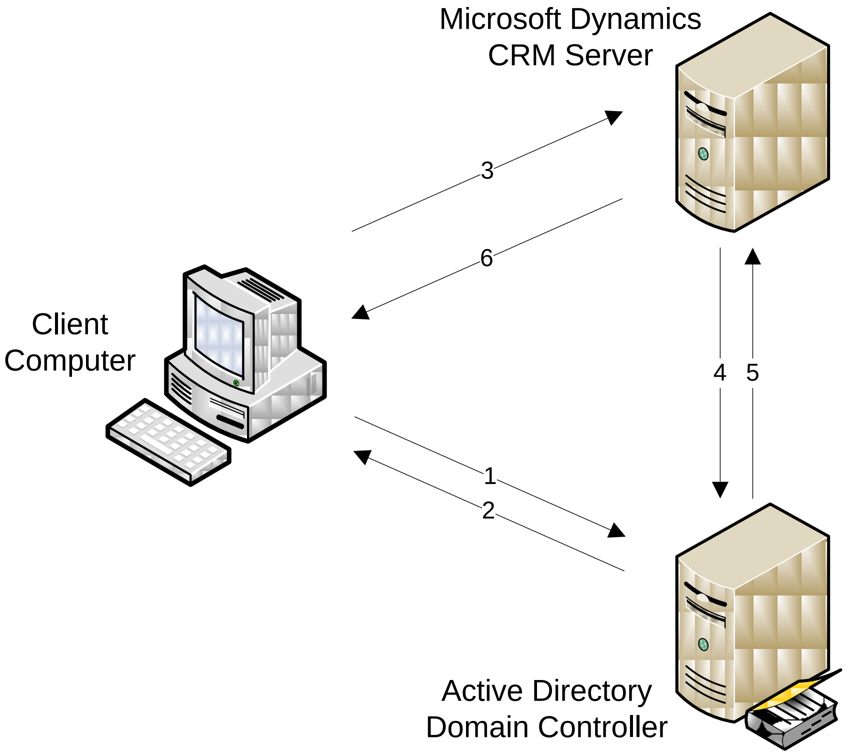
Integrated Windows Authentication in On Premise Deployments

For on-premise deployments, Microsoft Dynamics CRM typically authenticates users by leveraging Integrated Windows Authentication.

Note: For more information about the Integrated Windows Authentication in on-premise deployments, in the CRM SDK, see *Walkthrough: Using the Discovery Service with Active Directory Authentication* at: http://msdn.microsoft.com/en-us/library/bb955359.aspx

End User to CRM Application

The End User to CRM Application authentication flow in a typical on-premise deployment is shown in the following figure.

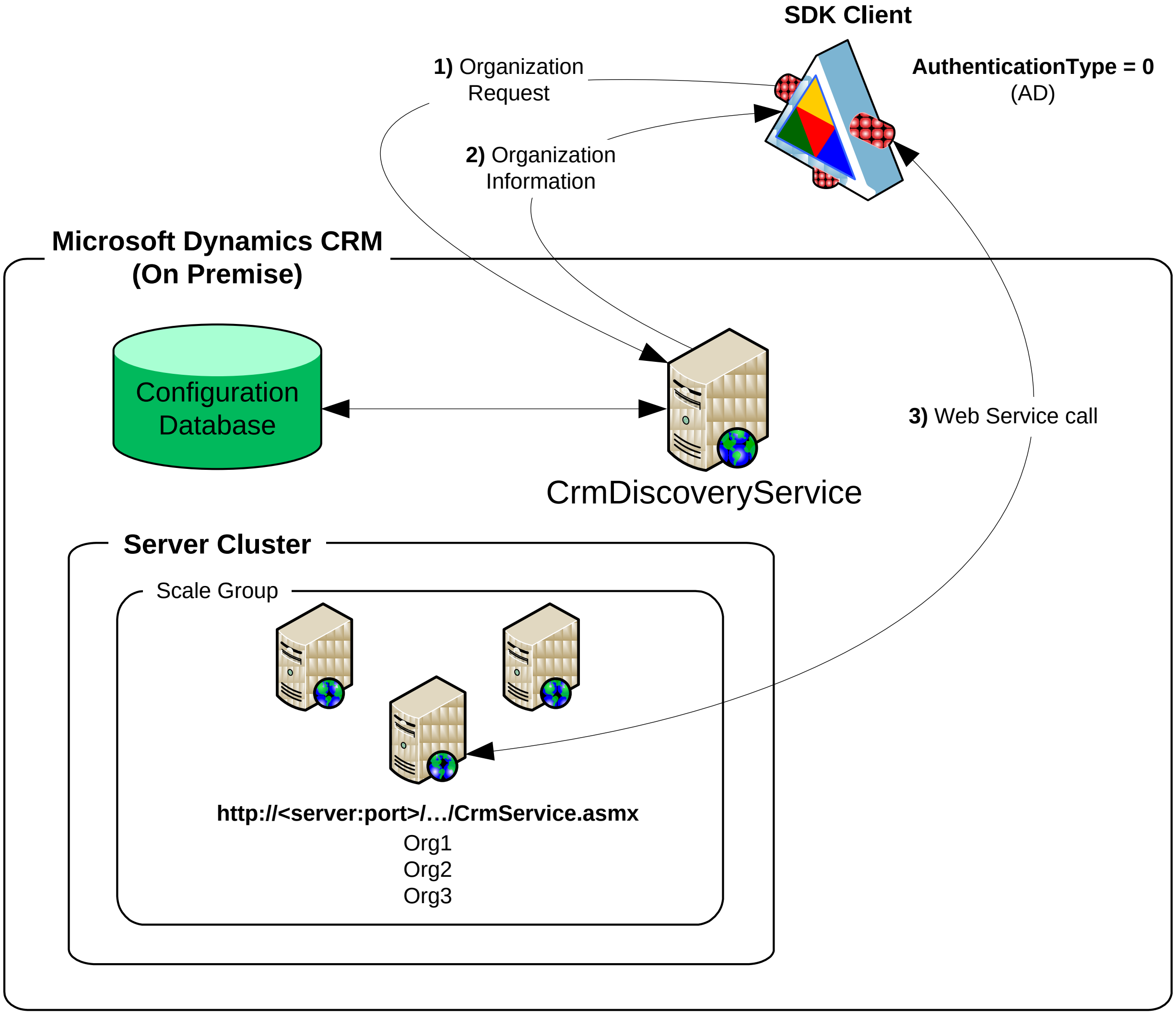

Microsoft Dynamics CRM Server
3
6
Client Computer
4
5
1
2
Active Directory Domain Controller

With end-user to CRM application communication in on-premise deployments, the Integrated Windows Authentication process involves the following steps:

1. User logs on to local Active Directory domain, which initiates a request for a token from the local domain controller.
2. Domain controller passes an authenticated token to the client computer.
3. User attempts to access the CRM application, which causes the client computer to send the authenticated token to Microsoft Dynamics CRM 4.0 server.
4. Microsoft Dynamics CRM 4.0 server verifies with the domain controller that the user's token has been authenticated.
5. Domain controller verifies that user has been authenticated.
6. CRM application, upon notification that a specific user has been authenticated, provides the appropriate level of access to the user.

SDK Client to Web Service

The SDK Client to Web Service authentication flow in a typical on-premise deployment is shown in the following figure.



With SDK Client to Web service communication in on-premise deployments, the Integrated Windows Authentication process involves:

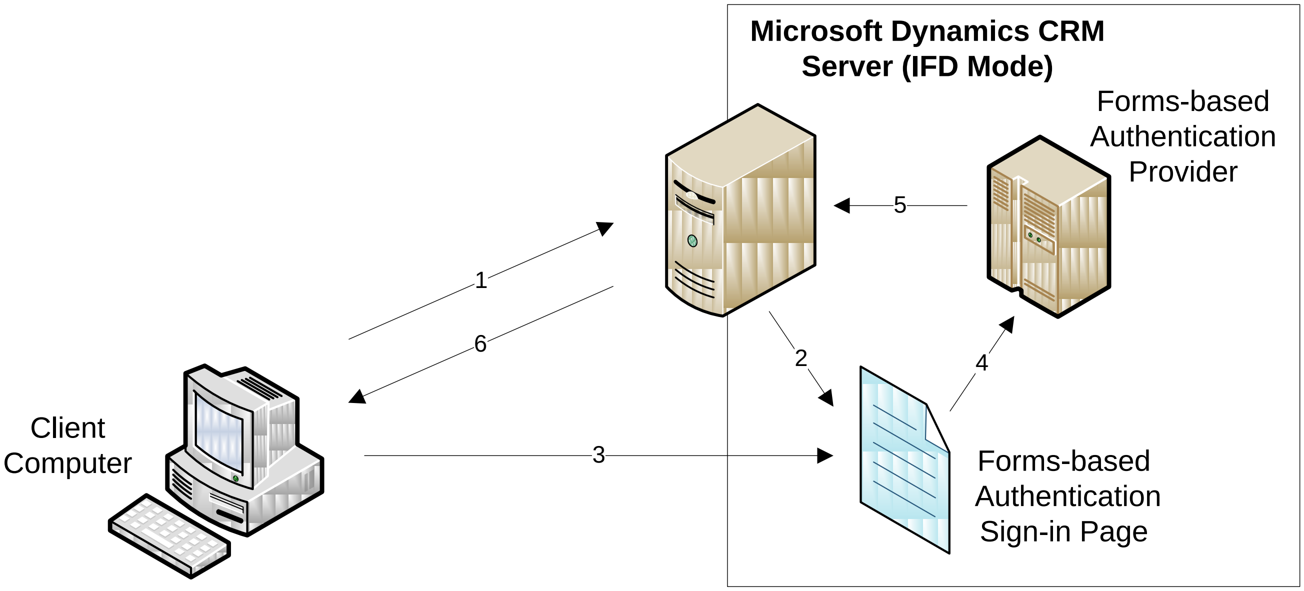
- Creating an instance of the **CrmDiscoveryService** Web service
- (1,2) (*Optional*) Obtaining a list of available organizations from the **CrmDiscoveryService** Web service; finding the target organization in the list.
  The **CrmDiscoveryService** Web service is accessed through a global URL of an on-premise Microsoft Dynamics CRM server. Each Microsoft Dynamics CRM server hosts the **CrmDiscoveryService** Web service, which is located at:
  http://<server:port>/MSCRMServices/2007/AD/CrmDiscoveryService.asmx
  Note: For sample code showing how to obtain organization information, in the Microsoft Dynamics CRM SDK, see the Active Directory authentication sample at:
  http://msdn.microsoft.com/en-us/library/cc151053.aspx
- (3) Creating an instance of the **CrmService** Web service; invoking a **CrmService** Web service method.

Forms-based Authentication in Hosted Deployments

For hosted deployments, Microsoft Dynamics CRM typically leverages Forms-based Authentication to authenticate users. Forms-based Authentication is similar to Integrated Windows Authentication, but when the **CrmDiscoveryService** Web service is accessed, the licensing model specified is a Microsoft Service Providers License Agreement (SPLA). In addition, a ticket must be obtained and set in the **CrmAuthenticationTokenValue** property value instance of the **CrmService** instance.

End User to CRM Application

The End User to CRM Application authentication flow in a typical hosted deployment is shown in the following figure.


Microsoft Dynamics CRM Server (IFD Mode)
Forms-based Authentication Provider
5
1
6
2
4
Client Computer
3
Forms-based Authentication Sign-in Page

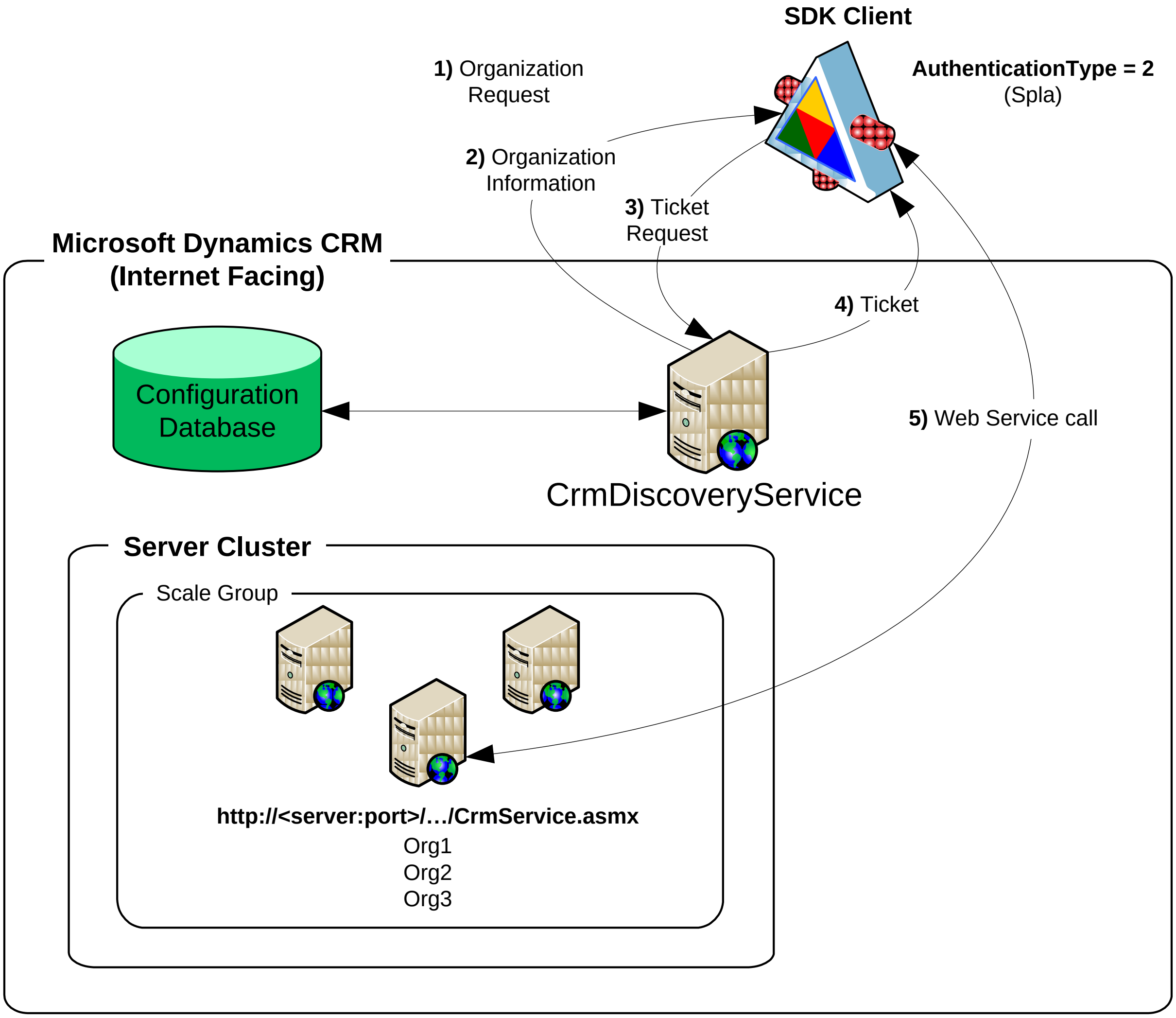
With end-user to CRM application communication in CRM hosted deployments, the Form-based Authentication process involves the following steps:

1. User attempts to access the hosted CRM Server.
2. Hosted CRM Server redirects the access request to the forms-based authentication sign-in page.
3. On the forms sign-in page, user enters Active Directory credentials (domain user name and password) and submits the form.
4. Forms sign-in page invokes the CRM Forms Authentication provider in the authentication pipeline, which attempts to validate the user.
5. If the credentials are valid, the forms authentication provider generates a token, serializes it to a cookie and attaches it to the current HTTP header and redirects the browser to CRM application. The cookie and the underlying token is proof that the Forms Authentication provider has verified the user’s identity, and the hosted CRM Server can decrypt the token to obtain the user's unique identifier.

   Note: The cookie is generated for the organization provided by the user in the URL.
6. CRM application, upon notification that a specific user has been authenticated, provides the appropriate level of access to the user.

SDK Client to Web Service

The SDK Client to Web Service authentication flow in a typical hosted deployment is shown in the following figure.



With SDK Client to Web service communication in hosted deployments, the Forms-based Authentication process involves:

- Creating an instance of the **CrmDiscoveryService** Web service proxy.
- (1,2) (*Optional*) Obtaining a list of available organizations from the **CrmDiscoveryService** Web service; finding the target organization in the list.

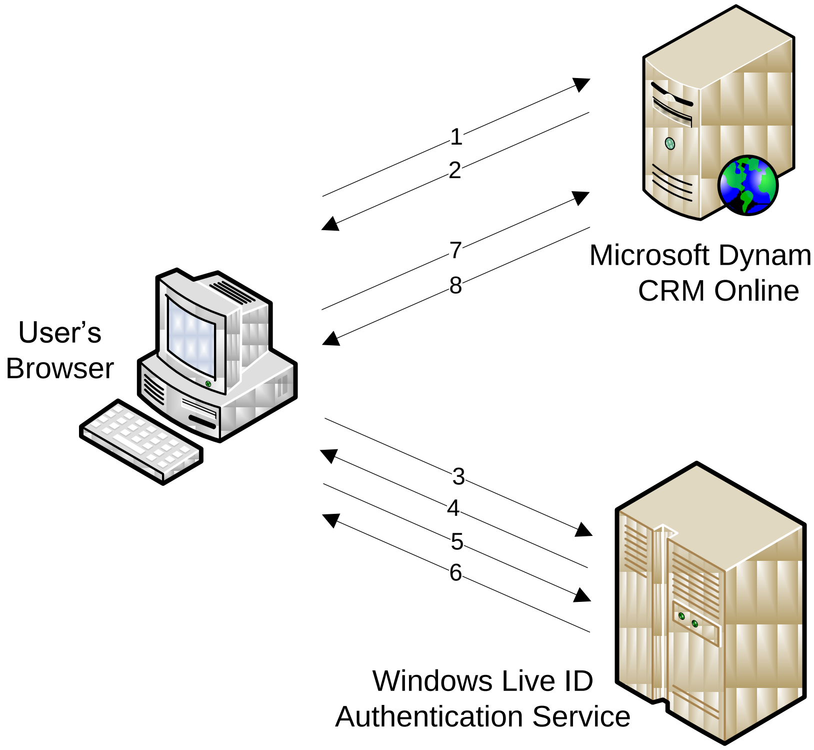
  The **CrmDiscoveryService** Web service is accessed through a global URL of a hosted (or IFD) Microsoft Dynamics CRM server. Each hosted (or IFD) server hosts the **CrmDiscoveryService** Web service, which is located at: https://<server:port>/MSCRMServices/2007/SPLA/CrmDiscoveryService.asmx
- (3,4) Obtaining a ticket (by providing the Active Directory username and password) from the **CrmDiscoveryService** Web service.
- Creating a **CrmAuthenticationToken** instance and set its **AuthenticationType**, **OrganizationName**, and **CrmTicket** property values.
- (5) Creating an instance of the **CrmService** Web service proxy; calling Web service methods.

Windows Live ID Authentication in CRM Online Deployments

CRM Online deployments leverage Windows Live ID Authentication to authenticate users.

End User to CRM Application

The End User to CRM Application authentication flow in a typical CRM Online deployment is shown in the following figure.



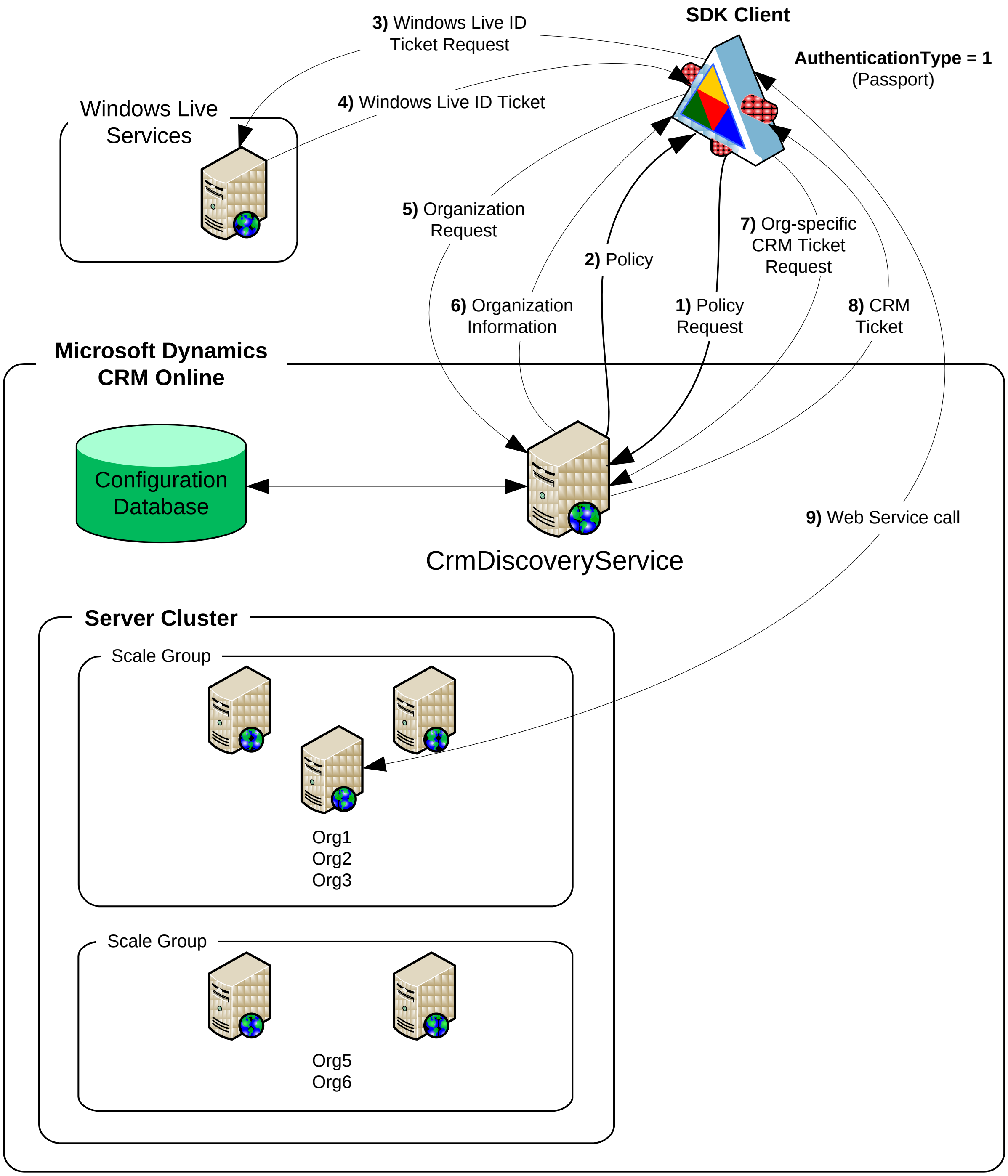
With end-user to CRM application communication in CRM Online deployments, the Windows Live ID based Authentication process involves the following steps:

1. The user, using a Web browser, visits the CRM Online Web site for the first time and has not yet signed in by using Windows Live ID.
2. The CRM Online Web application responds to the client computer with a redirect request in case the client computer does not have a LiveID cookie.
3. The client computer resends the request to the Windows Live ID authentication service.
4. The Windows Live ID authentication service prompts the user with a sign-on page.
5. On the Windows Live ID sign-in page, the user enters Windows Live ID credentials (e-mail name and password) and submits the form.
6. The Windows Live ID authentication service receives the sign-in request and attempts to validates the user's credentials.
7. If the credentials are valid, the authentication server responds by redirecting the client computer to CRM Online along with a token as a FORM POST parameter. The token is proof that Windows Live ID has verified the user's identity, and CRM Online can decrypt the token to obtain the user's unique identifier.

   Note: Users are also prompted to indicate the organization or business unit to which they want to log on.
8. CRM Online, upon notification that a specific user has been authenticated, provides the appropriate level of access to the user.

SDK Client to Web Service

Similar to that in the Hosted deployment model with the exception of requiring a Windows Live ticket, the SDK Client to Web Service authentication flow in a typical CRM Online deployment is shown in the following figure.



Note: The service account represents a virtual Microsoft Dynamics CRM Online user that is the owner of any data changes made to the Microsoft Dynamics CRM Online database through Microsoft Dynamics CRM SDK Web service calls. As with any other Microsoft Dynamics CRM user account, the service account must be added to each desired organization where business data is to be modified.

With SDK Client to Web service communication in CRM Online deployments, the Windows Live ID Authentication process involves:

- (1,2) Retrieving a policy and optional organization information from the **CrmDiscoveryService** Web service.

  The **CrmDiscoveryService** Web service is accessed through the global URL of the Microsoft Dynamics CRM Online server at: https://dev.crm.dynamics.com/MSCRMServices/2007/Passport/CrmDiscoveryService.asmx
- (3,4) Retrieving a Windows Live ID (WLID) ticket from the Windows Live service using certificate-based authentication.
- (5,6) Using the WLID ticket to retrieve information about all organizations to which the user belongs. The response contains one or more organization specific **CrmService** URLs.
- (7,8) Retrieving an organization-specific CRM ticket from the **CrmDiscoveryService** Web service.
- Creating a **CrmAuthenticationToken** instance and setting its **AuthenticationType**, **OrganizationName**, and **CrmTicket** property values.
- Creating an instance of the **CrmService** class that has the **Url** property value and the **CrmAuthenticationTokenValue** property value set.
- (9) Invoking **CrmService** Web service methods.

If the ticket expires during application execution, a new ticket must be obtained and assigned to the **CrmTicket** property of the **CrmAuthenticationToken** instance. Trying to access the CrmService Web methods with an expired ticket throws a SOAP exception. The **SoapException.Detail.Innertext** property contains the error code value of "8004A101".

Certificate Requirements for CRM Online SDK Clients

Accessing the Windows Live authentication service over the Internet and obtaining a Windows Live ticket requires the use of certificates. Based on the needs of the business environment, you can purchase a certificate from a certificate provider and set up a service account in Microsoft Dynamics CRM Online. Certificates currently supported for this purpose include:

- VeriSign Secure Site
- VeriSign Secure Site Pro
- Network Solutions SSL
- VeriSign Secure Site w/EV
- VeriSign Secure Site Pro w/EV
- Entrust (standard)

Note: For more information about downloading the Sign-in Assistant software from the Windows Live Web site and associating the certificate to your Windows Live ID, on MSDN, see the article *Walkthrough: Server to Server Authentication with CRM Online* at http://msdn.microsoft.com/en-us/library/dd548517.aspx

For unique scenarios requiring the use of a Smart Client (involving user interaction) against the CRM Online service, use the ticket service library (IDCRL) that is provided in the SDK\Bin folder of the SDK samples. To access the win32 IDCRL library, the SDK provides a .NET wrapper and the associated source code (in SDK samples in the folder SDK\Server\Helpers\CS\IdCrlWrapper.

Note: For more information about the IDCRL library, on MSDN, see the article *Walkthrough: Building the IDCRL Wrapper Code for Use with CRM Online* at: http://msdn.microsoft.com/en-us/library/bb955358.aspx

Authentication via Impersonation

Impersonation is a technique by which business logic (code) is executed on behalf of a Microsoft Dynamics CRM user to provide a desired feature or service using the appropriate role and object based security. Impersonation involves two different user accounts; one user account (Account A) executes code to perform some task on behalf of another account (Account B), which must be associated with a licensed Microsoft Dynamics CRM user. This functionality allows various clients and services, such as in a workflow or custom ISV solution, to call the Microsoft Dynamics CRM Web services on behalf of a Microsoft Dynamics CRM user.

Note: For more information about impersonation, on MSDN, in the Microsoft Dynamics CRM 4.0 SDK, see the topic *Impersonation* at http://msdn.microsoft.com/en-us/library/cc151052.aspx

Impersonation in On Premise, Hosted, and Internet Facing Deployments

In on-premise, hosted, and Internet-facing deployments of Microsoft Dynamics CRM, impersonation requires that Account A be a member of the **PrivUserGroup** group in Active Directory. In these types of deployments, Account A is not required to be associated with a licensed Microsoft Dynamics CRM user, though Account B must be so associated.

In addition, impersonation in these type of deployments requires that the authentication token's **CallerID** property be explicitly set using a valid CRM system user.

Note: For information about impersonation functionality in plug-ins, in this document, see *Appendix B: Impersonation in Plug-ins*.

Authentication from an ASPX Page

On-premise, hosted, and Internet-facing deployments of Microsoft Dynamics CRM support authentication from ASPX pages, functionality that is commonly used by ISVs to call CRM SDK code on behalf of the user from ISV pages.

In hosted and Internet-facing deployments, accessing the Web services from an ASPX page requires using the class **Microsoft.Crm.Sdk.CrmImpersonator**. When included in a using statement, the **CrmImpersonator** class allows a block of code to execute under the process credentials instead of the running thread's identity. At the end of the using statement, execution will return to running under thread identity.

When using the **Create** method or **Create** message with the **CRMImpersonator** class, you must set the **ownerid** property for the entity.

Important: In on-premise deployments, Web services can be accessed from an ASPX page without using the **CrmImpersonator** class. However, to ensure that an application works seamlessly in on-premise, hosted, and IFD deployments, should consider using the **CrmImpersonator** design pattern.

For example, the following code sample reflects the proper syntax:

```
using (new CrmImpersonator())
{
    CrmAuthenticationToken token;
    if (offline == true)
    {
        token = new CrmAuthenticationToken();
    }
    else
    {
        token = CrmAuthenticationToken.ExtractCrmAuthenticationToken(Context, orgname);
    }
    token.OrganizationName = orgname;
    token.AuthenticationType = 0;

    //Create the Service
    CrmService service = new CrmService();
    service.Credentials = System.Net.CredentialCache.DefaultCredentials;
    service.CrmAuthenticationTokenValue = token;
    service.Url = crmurl;

    account account = new account();
    account.name = "Offline Impersonator: " + DateTime.Now.TimeOfDay.ToString();
    if (offline == false)
        account.ownerid = new Owner("systemuser", token.CallerId);

    service.Create(account);
}
```

Note: For more information about impersonating a user from an ASPX page, in Microsoft Help and Support, see the following resources:

- *Authentication from an ASPX Page*
  http://msdn.microsoft.com/en-us/library/cc151050.aspx
- *How to impersonate a user from Active Server Pages*
  http://support.microsoft.com/kb/248187
- *CrmImpersonator Class (Sdk Assembly)*
  http://msdn.microsoft.com/en-us/library/cc156363.aspx

Impersonation in CRM Online Deployments

For Microsoft Dynamics CRM Online, impersonation requires that both user account (A) and user account (B) be licensed Microsoft Dynamics CRM users and that each have a Windows Live ID identity. In addition, user account (A) must be assigned the **Proxy** security role to successfully impersonate user account (B).

In addition, as with other deployment types, impersonation in CRM Online deployments requires that the authentication token’s **CallerID** property be explicitly set using a valid CRM system user. You can retrieve the logged-on user's Microsoft Dynamics CRM user ID (**CallerID**) from the **CrmDiscoveryService** Web service by using the **RetrieveCrmUserIdByExternalIdRequest** request.

Note: For more information impersonation in CRM Online deployments, on MSDN, see the article *Server-to-Server Authentication with Impersonation* at http://msdn.microsoft.com/en-us/library/dd548516.aspx

Authorization

Recall that in addition to providing for user access to the system via authentication, the goals of a security model include:

- Providing users with access only to the information that they require to do their jobs
- Categorizing types of users by role and restrict access based on those roles.
- Preventing users from accessing objects that they do not own or share
- Supporting data sharing by providing the ability to grant users with access to objects that they do not own to participate in a specified collaborative effort

To determine the extent to which users have access to the system and the resources it stores, Microsoft Dynamics CRM leverages two complementary security mechanisms:

- *Role-based security* in Microsoft Dynamics CRM focuses on grouping a set of privileges together that describe the tasks that are performed for a user in a specific job function. The basic concepts of role-based security include the following:
    - Users are assigned one or more roles based on their job function or tasks
    - Roles are associated with permissions (privileges and access levels) for the different business objects (entities)
    - Users gain access to entities or groups of entities in the system via membership in a role that has been assigned the necessary privileges and access levels to perform the users' jobs
- *Object-based security* in Microsoft Dynamics CRM focuses on how users gain access to individual instances of business objects (entities).

Important: In Microsoft Dynamics CRM, a user or team that can access an entity instance within the system is a *security principal*. While roles (or privileges) cannot be assigned to a team, teams can be granted access to objects in the same way that users can. For a user that is a member of a team, the actual level of access that a user has to objects shared within a team is determined by that user's privileges. Only users can own entity instances.

Note: For additional information about the types of entities provided in Microsoft Dynamics CRM 4.0, in this document, see *Appendix C: Entity Types in Microsoft Dynamics CRM 4.0*.

Role-based Security

Role-based security in Microsoft Dynamics CRM is based on the interaction of privileges and access levels, which work together through the use of security roles.

Privileges define what actions a user can perform on each entity in Microsoft Dynamics CRM. Privileges are pre-defined in Microsoft Dynamics CRM and cannot be changed; examples of privileges include Create, Read, Write, and Delete.

Access levels indicate which records associated with each entity the user can perform actions upon. Although default access levels are assigned to each privilege, the access level can be changed. For example, if a role allows the user to delete accounts, the access level associated with the account delete privilege indicates the specific accounts that the user can delete.

Each *security role* provides a combination of privileges and access levels specific to a Microsoft Dynamics CRM job function. Users can be assigned one or more roles, and users assigned multiple roles are granted the cumulative set of privileges that are associated with all of the roles to which they belong.

Important: A licensed user must be assigned at least one role to be able to access Microsoft Dynamics CRM; users who are not assigned to a role have no privileges.

The following sections provide additional information about the roles, privileges, and access levels that are associated with Microsoft Dynamics CRM 4.0 security model.

Privileges

Microsoft Dynamics CRM uses privileges as the core of the underlying security check. Privileges are "built in" to the product and are used throughout the application and platform layers. While you cannot add or remove privileges arbitrarily or change how they are used to grant access to certain functionality, you can create new roles by using the existing privileges.

Note: Creating a custom entity also creates the privileges associated with that entity.

The privileges that apply to most entity types in Microsoft Dynamics CRM are described in the following table.

Privilege	Allows the user to…
Create	Create instances of the specified entity **Note**: A user creating an instance of an entity must be assigned to a role that provides both the **Create** and **Read** privileges for that entity.
Read	View instances of the specified entity; this controls which records are displayed on views and reports.
Write	Make changes to instances of the specified entity
Delete	Remove instances of the specified entity
Append	Associate (attach) an instance of the specified entity to another instance of that entity
Append To	Associate an instance of an entity with the selected instance
Assign	Assign ownership of an instance of the specified entity to another user.
Share	Give access to instances of specified entities to another user while maintaining access to those instances; Share an instance of the specified entity with another user or team; sharing enables another user(s) to access an instance of an entity
Reparent	Assign a different parent to an instance of the specified entity **Note**: Creating an object parented to another provides the owner of the parent with rights on the child object (the child record is explicitly shared to the owner of the parent record).
Enable/Disable	Give privileges to instances of the specified entity

Note: The **Append** and **Append To** privileges work in combination. For example if a Note is attached to a Case, a user must have the **Append** privilege on the Note and the **Append To** privilege on the Case.

For additional information about the privileges available in Microsoft Dynamics CRM, in the MSDN Library, in *Appendix A: Security Roles and Privileges*, see *Privileges by Entity* at http://msdn.microsoft.com/en-us/library/bb955027.aspx

Access Levels

The access level associated with a privilege determines (for a given entity type) the levels within the organizational hierarchy at which a user belonging to a specific role can act on that type of entity.

Important: In Microsoft Dynamics CRM 4.0, the organization hierarchy includes three primary entities. *Users* represent people who use the Microsoft Dynamics CRM application. *Teams* are arbitrary groups of users created and defined by a user in an organization. *Business units* are the structural units of an organization, as defined by a user in the organization. They are the primary container entity within the organizational hierarchy. Business unit structure determines and defines the scope of the access levels within an organization.

Microsoft Dynamics CRM provides the access levels described (in order, from least to most restrictive) in the following table:

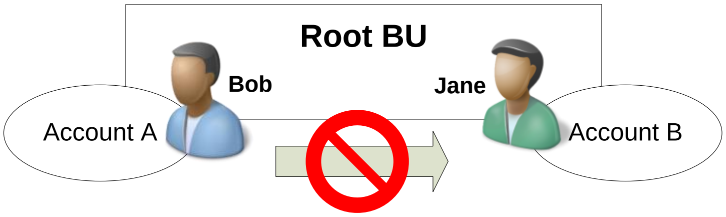
Access level	Description
Organization *(includes Parent:Child Business Units, Business Unit, and User access)*	Exposes all entity instances in the organization, regardless of the business unit hierarchical level to which the instance or the user belongs; usually reserved for managers with authority over the organization (**Example 4**)
Parent:Child Business Units *(includes Business Unit and User access)*	Exposes entity instances in the user's business unit and all subordinate business units; usually reserved for managers with authority over the business (**Example 3**)
Business Unit *(includes User access)*	Exposes entity instances in the user's business unit; usually reserved for managers with authority over the business unit (**Example 2**)
User	Exposes entity instances that the user owns, objects that are shared with the user, and objects that are shared with a team to which the user belongs; the typical level of access for sales and service representatives (**Example 1**)
None Selected	Nothing exposed

Each access level includes records that are made available by all access levels below the level that the privilege granted to the user.

Consider the following examples to better understand how privileges and access levels work in combination to secure access in a Microsoft Dynamics CRM 4.0 implementation.

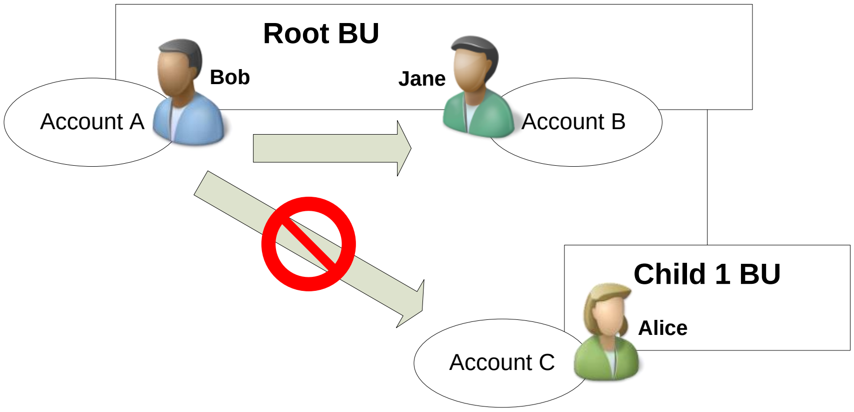
Example 1

- Bob and Jane are in the Root BU
- Bob owns Account A, and Jane owns Account B
- Bob has the Read Account privilege at **User** depth
- Bob can read Account A but not Account B



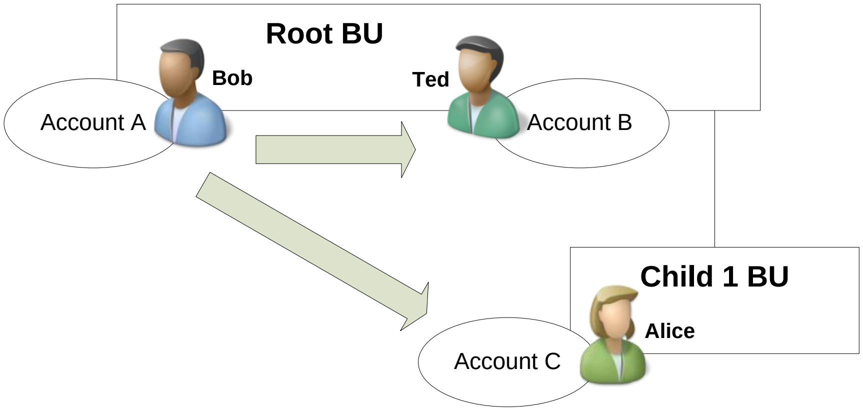
Example 2

- Bob and Jane are in the Root BU, and Alice is in the Child 1 BU
- Bob owns Account A, Jane owns Account B, and Alice owns Account C
- Bob has the Read Account privilege at **Business Unit** depth
- Bob can read Account A and Account B, but not Account C



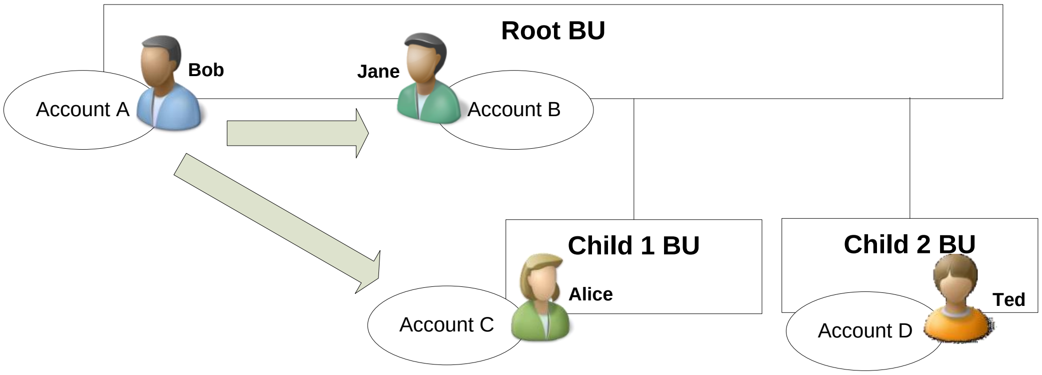
Example 3

- Bob and Jane are in the Root BU, and Alice is in the Child 1 BU
- Bob owns Account A, Jane owns Account B, and Alice owns Account C
- Bob has the Read Account privilege at **Parent:Child Business Units** depth
- Bob can read Account A, Account B, and Account C



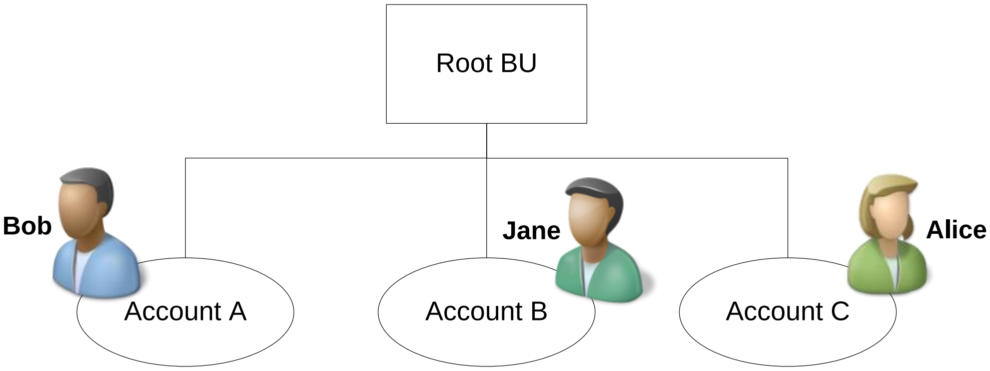
Example 4

- Bob and Jane are in the Root BU, Alice is in the Child 1 BU, and Ted is in the Child 2 BU
- Bob owns Account A, Jane owns Account B, Alice owns Account C, and Ted owns Account D
- Alice has the Read Account privilege at **Organization** depth
- Alice can read Account C and Account D



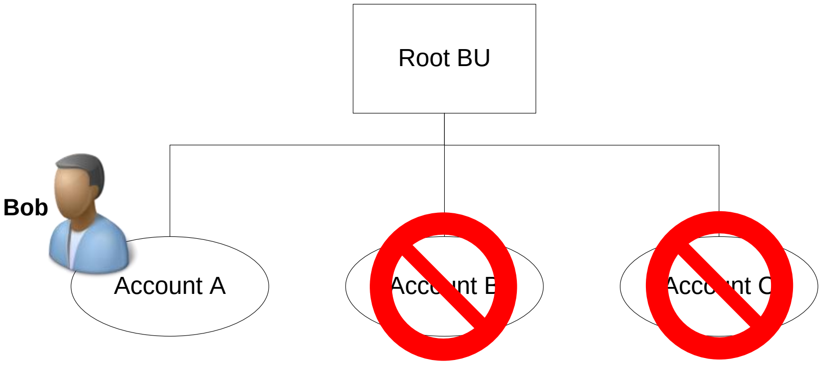
Example 5

- Bob, Jane, and Alice are in the Root BU
- Bob and Jane have the Read Account privilege at **User** depth
- Alice has the Read Account privilege at **Business Unit** depth

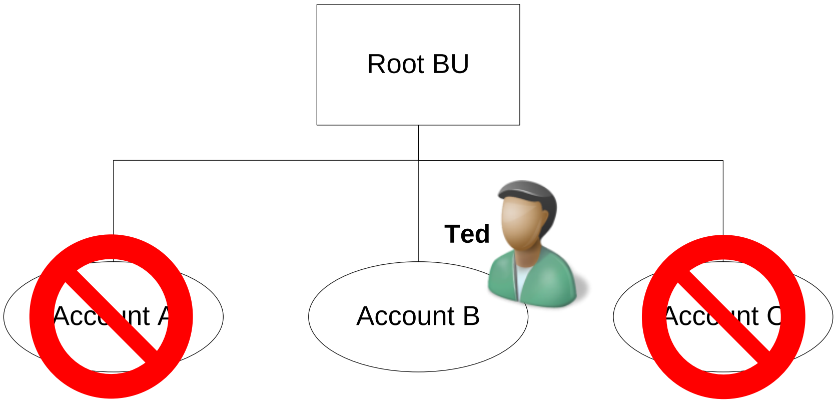


In this environment:

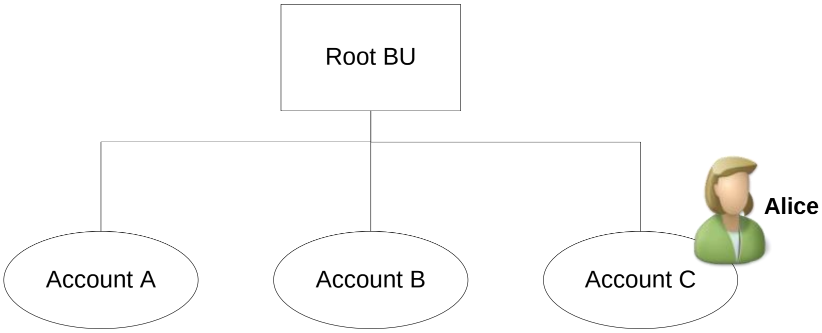
- Bob can read Account A, but not B or C



- Jane can read Account B, but not A or C



- Alice can read Accounts A, B, and C



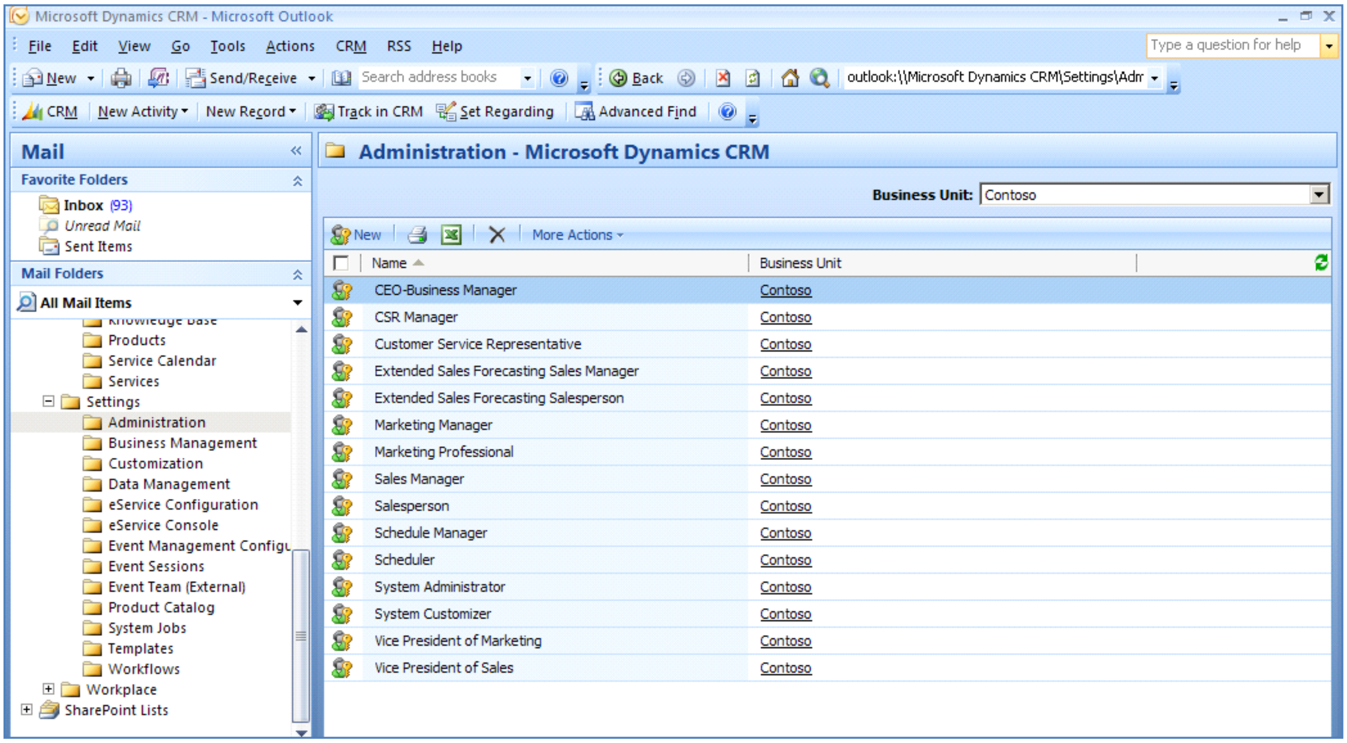
Note: For additional information about and examples how privileges and access levels work in combination to secure access in a Microsoft Dynamics CRM 4.0 implementation, in this document, see *Appendix D: Security Access Levels in Microsoft Dynamics CRM 4.0*.

Security Roles

Microsoft Dynamics CRM includes a set of pre-defined security roles that reflect common user roles. Each security role is associated with a set of privileges that determines the user's access to information within the company. Following security best practices, each pre-defined role provides access to the minimum amount of business data required to perform the job.

Note: For a complete listing of the pre-defined roles (and their associated responsibilities) in Microsoft Dynamics CRM, in the MSDN Library, under Microsoft Dynamics CRM 4.0, see *Appendix A: Security Roles and Privileges* at http://msdn.microsoft.com/en-us/library/bb954998.aspx

For example, the security roles that are associated with the Microsoft Dynamics CRM 4.0 deployment at Contoso, a fictitious company, are shown in the following screenshot:



Administration - Microsoft Dynamics CRM

Business Unit: Contoso

Name	Business Unit
CEO-Business Manager	Contoso
CSR Manager	Contoso
Customer Service Representative	Contoso
Extended Sales Forecasting Sales Manager	Contoso
Extended Sales Forecasting Salesperson	Contoso
Marketing Manager	Contoso
Marketing Professional	Contoso
Sales Manager	Contoso
Salesperson	Contoso
Schedule Manager	Contoso
Scheduler	Contoso
System Administrator	Contoso
System Customizer	Contoso
Vice President of Marketing	Contoso
Vice President of Sales	Contoso

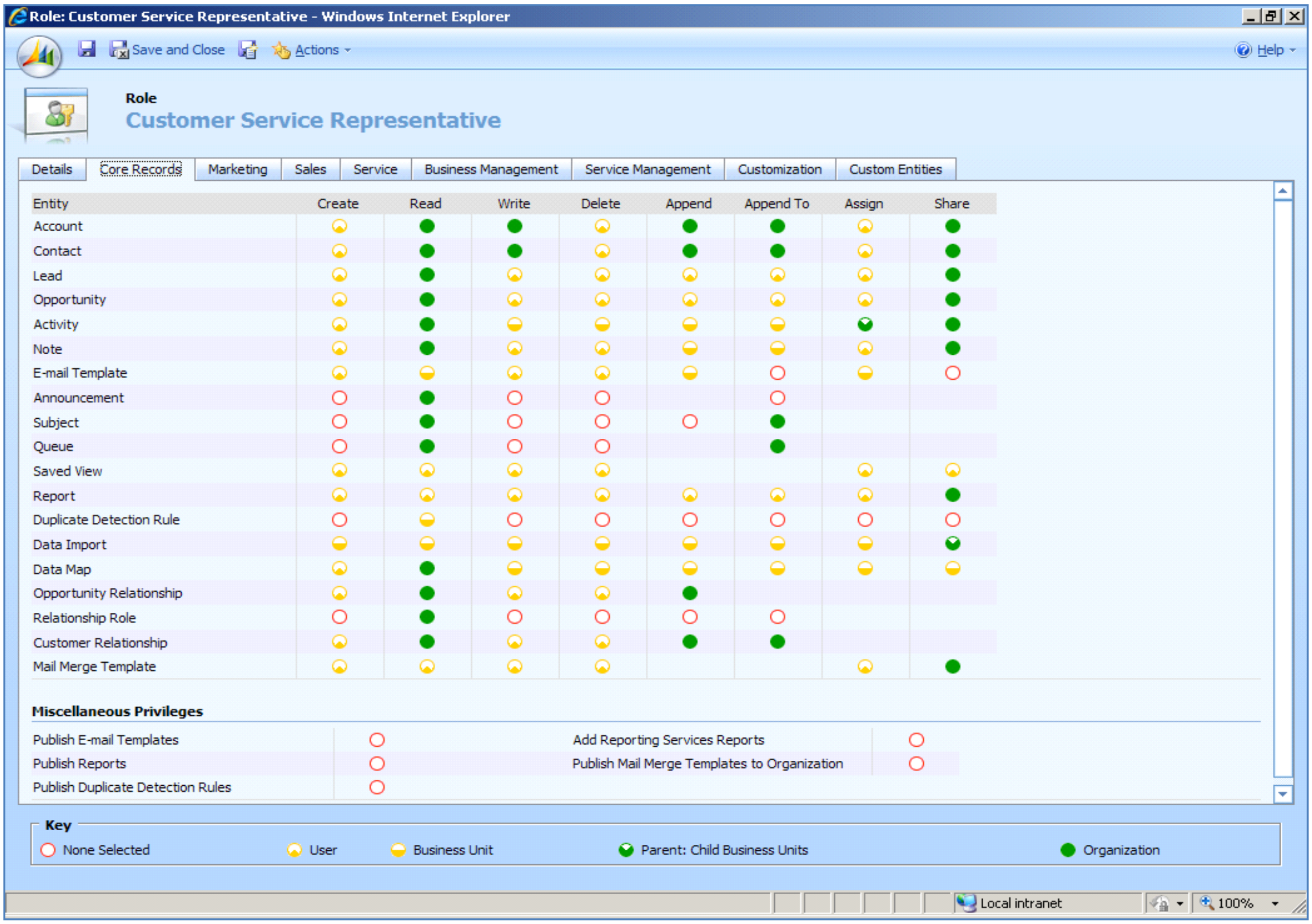
While you cannot modify privileges at the user level, you can define custom roles within Microsoft Dynamics CRM to accommodate the unique types of users within an organization.

For example, John is given a Salesperson role, which requires him to accept all leads assigned to him. However, the administrator wants John to be able to reassign the leads that are assigned to him. To provide for this scenario, the administrator could either:

- Modify the Salesperson role by incorporating this specific privilege
- Create a new role (by modifying a copy of an existing role) that incorporates this specific privilege, and then add John to this role.

In this case, however, unless all users who are assigned the Salesperson role also require the additional privilege, it is recommended that administrators use the second option (establish a new role) so that privilege is granted only to John. Even if all users who are assigned the Salesperson role require the additional privilege, it may be advantageous to create a second role. For example, this approach can be useful for granting Export privileges.

The privileges and access levels that are associated with the Customer Service Representative role at Contoso are shown in the following screenshot:



Role: Customer Service Representative - Windows Internet Explorer
Save and Close
Actions
Help

Role

Customer Service Representative

Details | Core Records | Marketing | Sales | Service | Business Management | Service Management | Customization | Custom Entities

Entity	Create	Read	Write	Delete	Append	Append To	Assign	Share
Account	◔	●	●	◔	●	●	◔	●
Contact	◔	●	●	◔	●	●	◔	●
Lead	◔	●	◔	◔	◔	◔	◔	●
Opportunity	◔	●	◔	◔	◔	◔	◔	●
Activity	◔	●	◒	◒	◒	◒	◕	●
Note	◔	●	◔	◔	◒	◒	◔	●
E-mail Template	◔	◒	◔	◔	◒	○	◒	○
Announcement	○	●	○	○		○		
Subject	○	●	○	○	○	●		
Queue	○	●	○	○		●		
Saved View	◔	◔	◔	◔			◔	◔
Report	◔	◔	◔	◔	◔	◔	◔	●
Duplicate Detection Rule	○	◒	○	○	○	○	○	○
Data Import	◒	◒	◒	◒	◒	◒	◒	◕
Data Map	◔	●	◒	◒	◒	◒	◒	◒
Opportunity Relationship	◔	●	◔	◔	●			
Relationship Role	○	●	○	○	○	○		
Customer Relationship	◔	●	◔	◔	●	●		
Mail Merge Template	◔	◔	◔	◔			◔	●

Miscellaneous Privileges

Publish E-mail Templates	○	Add Reporting Services Reports	○
Publish Reports	○	Publish Mail Merge Templates to Organization	○
Publish Duplicate Detection Rules	○		

Key

○ None Selected ◔ User ◒ Business Unit ◕ Parent: Child Business Units ● Organization

Object-based Security

Object-based security applies to individual instances of entities and is provided by using access rights. The relationship between an access right and a privilege is that access rights apply only after privileges have taken effect. For example, if users do not have the privilege to read accounts, they will be unable to read any account, regardless of the access rights another user might grant them to a specific account through sharing.

Access Rights

An access right is granted to a user for a particular entity instance. The following table describes the access rights that are provided in Microsoft Dynamics CRM 4.0.

Right	Enumeration Name	Controls whether the user can…
Read	ReadAccess	Read an entity instance.
Write	WriteAccess	Update an entity instance.
Assign	AssignAccess	Assign an entity instance to another user.
Append	AppendAccess	Attach another entity instance to the specified entity instance. The Append and Append To access rights work in combination with one another. Every time that a user attaches one entity instance to another, the user must have both rights. For example, when you attach a note to a case, you must have the Append access right on the note and the Append To access right on the case for the operation to work.
Append To	AppendToAccess	Append the entity instance to another entity instance. The Append and Append To access rights work in combination with one another. Each time that a user attaches one entity instance to another, the user must have both rights. For example, when you attach a file to a note, you must have the Append access right on the file attachment and the Append To access right on the note for the operation to work.
Share	ShareAccess	Share an entity instance with another user or team. Sharing gives another user access to an entity instance.
Delete	DeleteAccess	Delete an entity instance.

Create Access

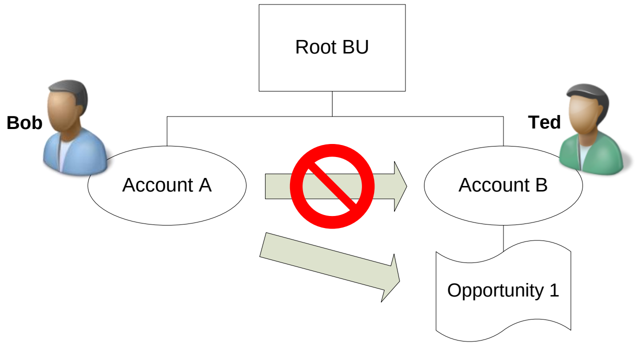
The right to create an instance of an entity is not included in the previous table because this right does not apply to an individual instance, but instead to a class of entities. Therefore, Create is handled as a privilege instead of as an access right. By default, the user who creates an entity instance will have all rights on that entity instance, unless his or her other privileges forbid a specific right. The Create privilege controls whether you can create an entity instance. You can have the Create privilege with Business Unit, Parent:Child Business Unit, or Organization access level, and you will be able to create instances for other users. However, you can create instances for yourself only if you have Create and Read privileges.

Note: For more information about dependencies that relate to Create privileges, in the Microsoft Dynamics CRM 4.0 SDK, see *Security Dependencies* at http://msdn.microsoft.com/en-us/library/bb955133.aspx

Sharing Entity Instances

Sharing provides the ability for users to allow other users or teams access to specific customer information, which can be useful for sharing data with users that are assigned to roles that have only the **User** access level. Consider the following example:

- Bob and Ted are both assigned the Salesperson role, which has **User** Read and Write access to accounts
- Ted owns Account B and shares Opportunity 1 with Bob, with Read rights
- Bob can now read Opportunity 1, but not Account B



Microsoft Dynamics CRM provides the following sharing capabilities:

Capability	Description
Share	A user who has share privileges on a given entity type can share instances of that type with any other user or team in Microsoft Dynamics CRM.
Share rights	To share an entity instance with another user, grant access rights (Read, Write, Delete, Append, Assign, and Share) to the other user for that entity instance. Access rights on a shared entity instance can be different for each user with whom the entity instance is shared. However, you cannot grant a user rights that he or she would not automatically have for that type of entity based on the role assigned to that user. For example, when you share an account with a user that does not have Read privileges on accounts, the user will not be able to see that account.
Modify share	Alter the rights granted to a shared entity instance after it has been shared.
Remove share	When you share an entity instance with another user or team, you can stop sharing the instance at a later time. After you remove sharing for an entity instance, the other user or team loses access rights to the instance.

A user might have access to the same entity instance in more than one context. For example, a user might share an entity instance directly with specific access rights, and that user might also be on a team with which the same entity instance is shared with different access rights. In this case, the user would have the cumulative the rights in each context.

Important: For security reasons, it is important to develop the practice of sharing only the necessary objects, or entity instances, among the smallest set of users possible, and to grant only the minimum access required for users to do their jobs.

Note: For a list of the entities that support sharing, in the Microsoft Dynamics CRM 4.0 SDK, see *GrantAccess Message (CrmService)* at http://msdn.microsoft.com/en-us/library/bb959430.aspx

Assigning Entity Instances

Anyone with Assign privileges on an entity instance can assign that object to another user. When an entity instance is assigned to a new user, the new user becomes the owner of the entity instance and its related entity instances. The original user loses ownership of the entity instance but automatically shares it with the new owner.

In Microsoft Dynamics CRM 4.0, the system administrator can decide for an organization whether entity instances should be shared with previous owners or not after the assign operation. If **Share with previous owner** is chosen, then after the assign operation the previous owner shares the entity instance with all access rights. Otherwise the previous owner does not share the entity instance and therefore may not have any access to the instance, depending on his or her privileges.

Note: For a list of entities that support Assign, in the Microsoft Dynamics CRM 4.0 SDK, see *Assign Message (CrmService) at* http://msdn.microsoft.com/en-us/library/bb959372.aspx

Cascading Rules

In Microsoft Dynamics CRM, certain actions, such as sharing and assigning, on a parent entity instance can affect child entity instances based upon the *cascading rules* that are configured on the relationships between the parent object and its child objects.

Dynamics CRM actions that are can be controlled by using cascading rules include:

- Assign
- Delete
- Merge
- Reparent
- Share
- Unshare

The cascading rules in Microsoft Dynamics CRM 4.0 are described in the following table.

Rule	Description
Cascade All	Perform the action on the specified entity instance and all related entity instances.
Cascade None	Perform the action on the specified entity instance only. Do not cascade to related entity instances.
Cascade Active	Perform the action on the specified entity instance and all related entity instances that are active or open.
Cascade User Owned	Perform the action on the specified entity instance and all related entity instances that are owned by the same user as this entity.
Remove Link	Perform the action on the specified entity instance and remove the link to the related entity instance. No changes are made to the related entity instance.
Restrict	Applies to delete only. The delete is not allowed if there are other entity instances that reference the ID of the entity instance being deleted.

Note: For more information about cascading rules in Microsoft Dynamics CRM 4.0, in the Microsoft Dynamics CRM 4.0 SDK, see the topic *Cascading Rules* at: http://msdn.microsoft.com/en-us/library/bb955296.aspx

Sharing and Inheritance

A child entity instance inherits the sharing properties of the parent entity instance according to the cascading rules configured for the parent entity instance.

Sharing is maintained on individual entity instances. An entity instance inherits the sharing properties from its parent and also maintains its own sharing properties. Therefore, an entity instance can have two sets of sharing properties—one that it has on its own and one that it inherits from its parent.

For example, Bob and Ted are working on a high-priority lead. Bob creates a new lead and two activities, shares the lead with Ted, and selects cascade sharing, which also gives Ted access to the associated activities. Ted makes a call and sends an e-mail regarding the new lead. Bob sees that Ted has contacted the company two times, so he does not make another call.

Note: Bob could also have explicitly shared the two activities with Ted without having to share the lead.

Removing the share of a parent entity instance removes the sharing properties of objects (entity instances) that it inherited from the parent. That is, all users who previously had visibility into this entity instance no longer have visibility. Notice that certain child objects might still be shared to some of these users if they were shared individually.

Summary

There are several best practices and guidelines to consider as you approach configuring an organization's Microsoft Dynamics CRM security model.

Best Practices

When configuring the Microsoft Dynamics CRM security model to accommodate a specific business, keep in mind the following best practices.

1. **Understand existing data security strategies before implementing**.

   You must understand whether or not existing organizational structures must map directly to the business unit structure and security in the deployment of Microsoft Dynamics CRM.

   In some cases, established data management strategies can be used to control data integrity and job function privileges. Sometimes you will be required to help formalize these strategies.

   As you plan the organizational model that you will deploy with Microsoft Dynamics CRM, you must determine if:

   - Existing data security strategies are the best fit for your CRM deployment, considering both short and long term priorities.
   - Changes to strategy must be implemented given the new tools being implemented.

2. **Understand if existing job functions must map directly to security roles**.

   By having a better understanding of the responsibilities each person has within the company, you can clearly define what data they must access and if security roles must map directly to job functions. People who perform the same job function must have a standard role and privileges.

   This results in:

   - Simpler deployment of the application
   - Standardized training for the users by job function
   - More efficient change management

3. **Compare standard security roles with existing job functions**.

   If different job functions require different security privileges, a role must be established for each job function. Map the default roles within Microsoft Dynamics CRM to the organization's job functions. Document any job function that cannot be mapped, and create a new role to reflect its requirements.

   Consider the following during the planning process:

   - If an existing role in the system provides sufficient privileges for the user's job functions
   - If the role provides too many privileges based on the user's job functions
   - If there are unique job functions that require creating or modifying an existing role

4. **Create tiered security roles if several job functions require the same security privileges and access levels**.

   In some deployments where there are non-traditional job functions or a large number of different job functions, creating a separate security role for each job function can result in unacceptable redundancies. These redundancies slow the deployment process and create inefficiencies when you make changes to security. To implement tiered security:

   - Create a base security role that is assigned to all users. This role must contain the most restrictive privileges and access levels, but must include all privileges needed by all users. This role is assigned to every new user.
   - Additional security roles can be created to add specific privileges to specific users. Where appropriate, you can add a logical set of privileges in each additional role. For example, when you create a custom entity, you might add only the Read privilege for the new entity to the base role and give full access to the new entity in the additional security role. Assign the new security role to those users who will be managing the data in the new entity.

Additional Considerations

In addition to the best practices highlighted above, guidelines for leveraging the Microsoft Dynamics CRM security model include the following:

- Strictly limit the number of people assigned the role of System Administrator.
- Create roles according to the security best practice of least privilege, providing access to the minimum amount of business data required for the task; assign users the appropriate role(s) for their job.
- When appropriate, use sharing to grant specific users specific rights on individual objects, rather than granting broader privileges on all objects of a given type.
- Use teams to create cross-functional groups to share specific objects across the team.
- Train users with sharing access rights to share the least amount of information required.

Important: If a user needs additional access levels or rights, create a new role by copying an existing role, augment it with the necessary privileges, and then add the user to the new role. A user's rights are the union of all the rights associated with the roles to which he or she has been assigned; do not grant additional privileges to the original role privileges when those privileges are required by only a few users.

Appendix A: Term List

Descriptions of key terms associated with security and authentication in Microsoft Dynamics CRM 4.0 are listed in the following table.

Term	Description
Access levels	A security role setting for a privilege that determines for a given entity type at which levels within the organization hierarchy a user can act on that entity type; each privilege can have up to four access levels: **User**, **Business Unit**, **Parent:Child Business Units**, **Organization**.
Authentication	The process by which the system validates a user's logon information. A user's name and password are compared against an authorized list, and if the system detects a match, access is granted to the extent specified in the permission list for that user.
Authorization	The right granted an individual to use the system and the data stored on it. Authorization is typically set up by a system administrator and verified by the computer based on some form of user identification, such as a code number or password.
Business units	One of three entities comprising the Microsoft Dynamics CRM organizational and business structure, business units represent the structural units of an organization, as defined by a user in the organization. They are the primary container entity within the organizational hierarchy. Business unit structure determines and defines the scope of the access levels in an organization.
Cascading rules	Configuration settings that control how certain actions, such as sharing or assigning, on a parent entity instance affect the child entity instances of that parent object.
Federated Identity Relationship	A standards-based arrangement between organizations in which user claims from one organization are passed to and recognized by another. Users can therefore sign in to (and be authenticated by) their identity provider—the organization that manages their identity account—and then have their authentication information passed to a federated partner as needed without being required to sign in again.
Federation	A trust-based agreement between two organizations with some common purpose, such that both want authentication assertions from one organization to be recognized by the other
Filtered Views	SQL database filtered views that are used for business data access. Fully compliant with the Microsoft Dynamics CRM security model, filtered views exist for all Microsoft Dynamics CRM business objects (entities). Data in filtered views is restricted at the organization level, the business unit level, and the owner level.
Impersonation	The ability of a thread to run in the security context of a security principal different from the security principal that started the process. This is usually so that a process can gain access to resources on behalf of a user.

Term	Description
Interaction point	An access channel to Microsoft Dynamics CRM 4.0 that provides access control in a way that allows consistent treatment of data visibility across channels while at the same time maintaining the flexibility required to implement complex situations as necessary.
Object-based security	Object-based security applies to individual instances of entities and is provided by using access rights. Access rights and privileges are related in that access rights apply only after privileges have taken effect.
Privilege	Authorizes a user to perform a specific action on a specific entity type; privileges apply to an entire class of objects, rather than individual instances of objects.
Role	A defined set of privileges within the organization. The security role assigned to a user determines which tasks the user can perform and which parts of the user interface the user can view. All users must be assigned at least one security role to be able to access the system.
Role-based security	The fundamental concept in role-based security is that privileges are assigned to defined categories of users (known as roles) rather than to individual users. When a user is assigned to one of these roles, he or she is assigned the set of privileges associated with that role. A user who is not assigned to a role does not have any privileges.
Security principal	A person or group that can own or access an entity instance within the system. There are two types of security principals within Microsoft Dynamics CRM: users and teams.
Teams	One of three entities comprising the Microsoft Dynamics CRM organizational and business structure, teams represent arbitrary groups of users created and defined by a user in an organization.
Users	One of three entities comprising the Microsoft Dynamics CRM organizational and business structure, users represent the people who use the Microsoft Dynamics CRM application.

Appendix B: Impersonation in Plug-ins

Microsoft Dynamics CRM obtains the pre-entity and post-entity images that are passed to plug-ins in the execution context on behalf of the impersonated system user. Any business logic executed within a plug-in, including Web service method calls, is governed by the security privileges of the impersonated user.

Important: Microsoft Dynamics CRM Online does not support the use of plug-ins.

Plug-ins execute under the security account that is specified on the **Identity** tab of the **CRMAppPool Properties** dialog box. By default, **CRMAppPool** uses the **Network Service** account identity. If the **CRMAppPool** identity is changed to a system account other than Network Service, the new identity account must be added to the **PrivUserGroup** group in Active Directory.

Impersonation during plug-in registration

One method to impersonate a system user within a plug-in is by specifying the impersonated user during plug-in registration. When registering a plug-in, if the **impersonatinguserid** field of the **sdkmessageprocessingstep** or **SdkMessageProcessingStepRegistration** class instance is set to a specific Microsoft Dynamics CRM system user, Web service calls made by the plug-in execute on behalf of the impersonated user. If the **impersonatinguserid** field is set to a value of **null** or **Guid.Empty** during plug-in registration, the calling/logged on user or the standard "system" user is the impersonated user.

Whether the calling/logged on user or "system" user is used for impersonation is dependent on the request being processed by the pipeline and is beyond the scope of this documentation.

Impersonation during plug-in execution

Impersonation that was defined during plug-in registration can be altered in a plug-in at run time. Even if impersonation was not defined at plug-in registration, plug-in code can still use impersonation. The following discussion identifies the key properties and methods that play a role in impersonation when making Web service method calls in a plug-in.

The platform passes the impersonated user ID to a plug-in at run time through the **IPluginExecutionContext.UserId** property. The **UserId** property can have either of the following values:

- *Initiating user* - The **impersonatinguserid** property of the **sdkmessageprocessingstep** or **SdkMessageProcessingStepRegistration** class instance is set to **null** or **Guid.Empty** at plug-in registration.
- *Impersonated user* – The **impersonatinguserid** property is set to a valid system user ID at plug-in registration.

If you specify an impersonated user during plug-in registration, you should set up the Web service proxy in the plug-in by passing a value of **true** to the **CreateCrmService** method or the **CreateMetadataService** method. Passing a value of **true** indicates to use the ID in the **IPluginExecutionContext.UserId** property as the impersonated user.

The following code example shows how to do this.

```
[C#] ICrmService service = context.CreateCrmService(true);
```

This is equivalent to the following code:

```
[C#] ICrmService service = context.CreateCrmService(context.UserId);
```

To ignore any impersonating user set during plug-in registration, use the following code.

```
[C#] ICrmService service = context.CreateCrmService(false);
```

When a value of **false** is passed to the **CreateCrmService** or the **CreateMetadataService** method, the platform uses the built-in "system" account to execute Web service method calls made by your plug-in code. The "system" account is a high privilege user account with some restrictions. For example, the "system" account cannot create a task activity.

The **InitiatingUserId** property of **IPluginExecutionContext** contains the ID of the system user that called the Web service method that ultimately caused the plug-in to execute. The following code shows how to create a Web service proxy to make Web method calls on behalf of the initiating user.

```
[C#] ICrmService service = context.CreateCrmService(context.InitiatingUserId);
```

If the **impersonatinguserid** property is set during plug-in registration, this line of code effectively ignores that setting for any Web method calls to the Web service.

For plug-ins executing offline, any entities created by the plug-in are owned by the logged on user. Impersonation in plug-ins is not supported while in offline mode.

Note: When you register a plug-in by using the plug-in registration sample tools that are provided with the SDK, Web service methods invoked by the plug-in always execute under the account of the calling or logged on user. The tools do not offer impersonation as a supported feature. For more information about the plug-in sample code, in the Microsoft Dynamics CRM 4.0 SDK, see *Plug-in Sample Code* at http://msdn.microsoft.com/en-us/library/cc151207.aspx

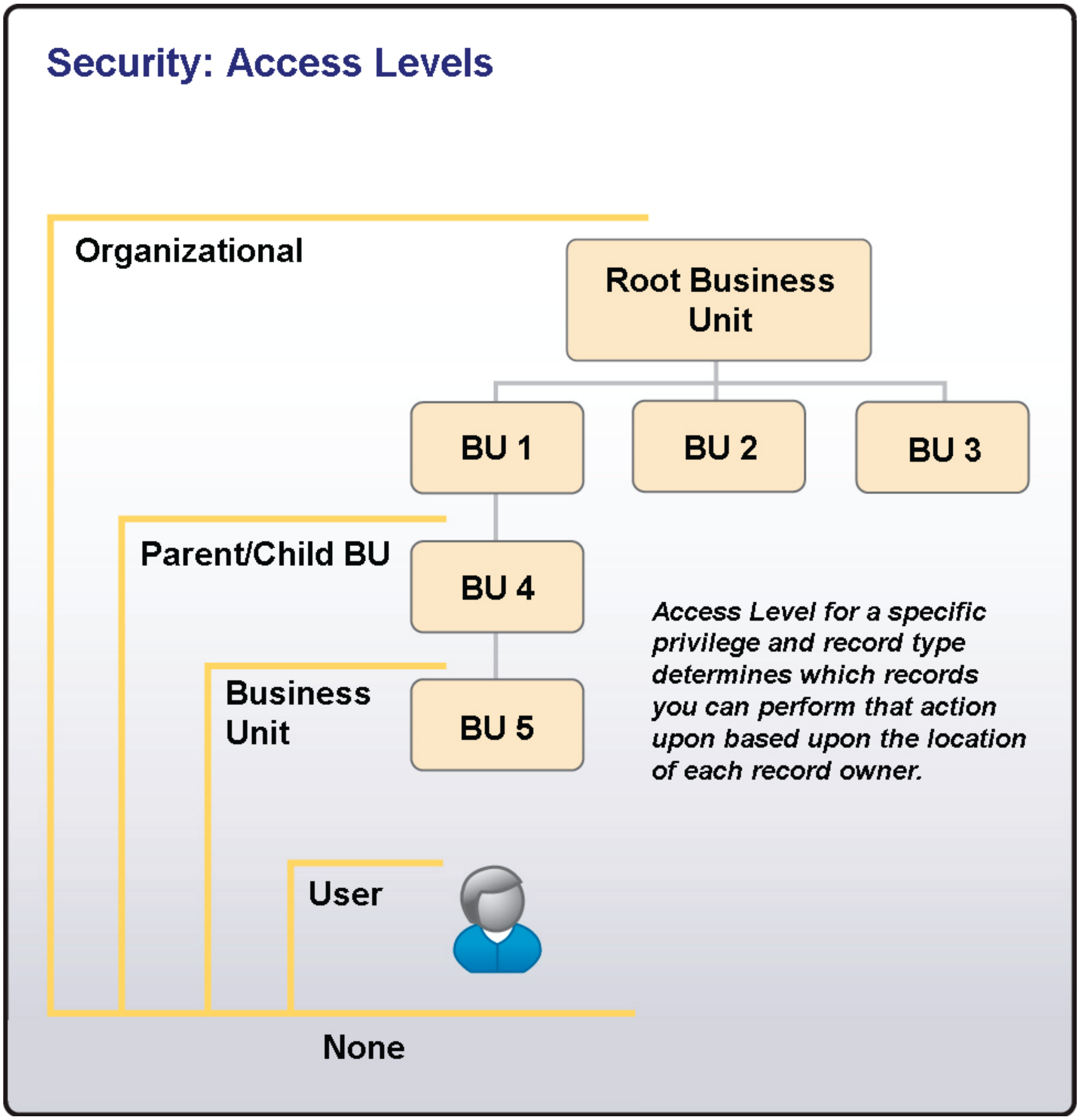
Appendix C: Entity Types in Microsoft Dynamics CRM 4.0

Microsoft Dynamics CRM 4.0 includes the entity types shown in the following table:

Entity Type	Examples	Characteristics
User-Owned	Account, Contact, Lead, …	▪ Have a specific owner attached to them ▪ Can be shared to other users or teams ▪ Can be assigned to other users ▪ Access is determined by privilege depth on that object and sharing
Business-Owned	Business Unit, Role, SystemUser	▪ Same as User-Owned but without assign/share ▪ Access is determined by privilege depth
Organization-Owned	Product, Territory, ContractTemplate, …	▪ Access is determined by organization membership (i.e. all Organization-Owned entities can have only Organization privilege depths)
Child Entities	SalesLiteratureItem, QuoteDetail, …	▪ Access is determined through parent object ▪ Example: Privilege to read ContractDetail == Privilege to read Contract ▪ Example: AccessCheck(user, ContractDetail A, Read) == AccessCheck(user, ContractDetail A's parent Contract, Read)

Appendix D: Security Access Levels in Microsoft Dynamics CRM 4.0

The basic relationship between the various access levels proved in Microsoft Dynamics CRM 4.0 are shown in the following graphic:



Security: Access Levels

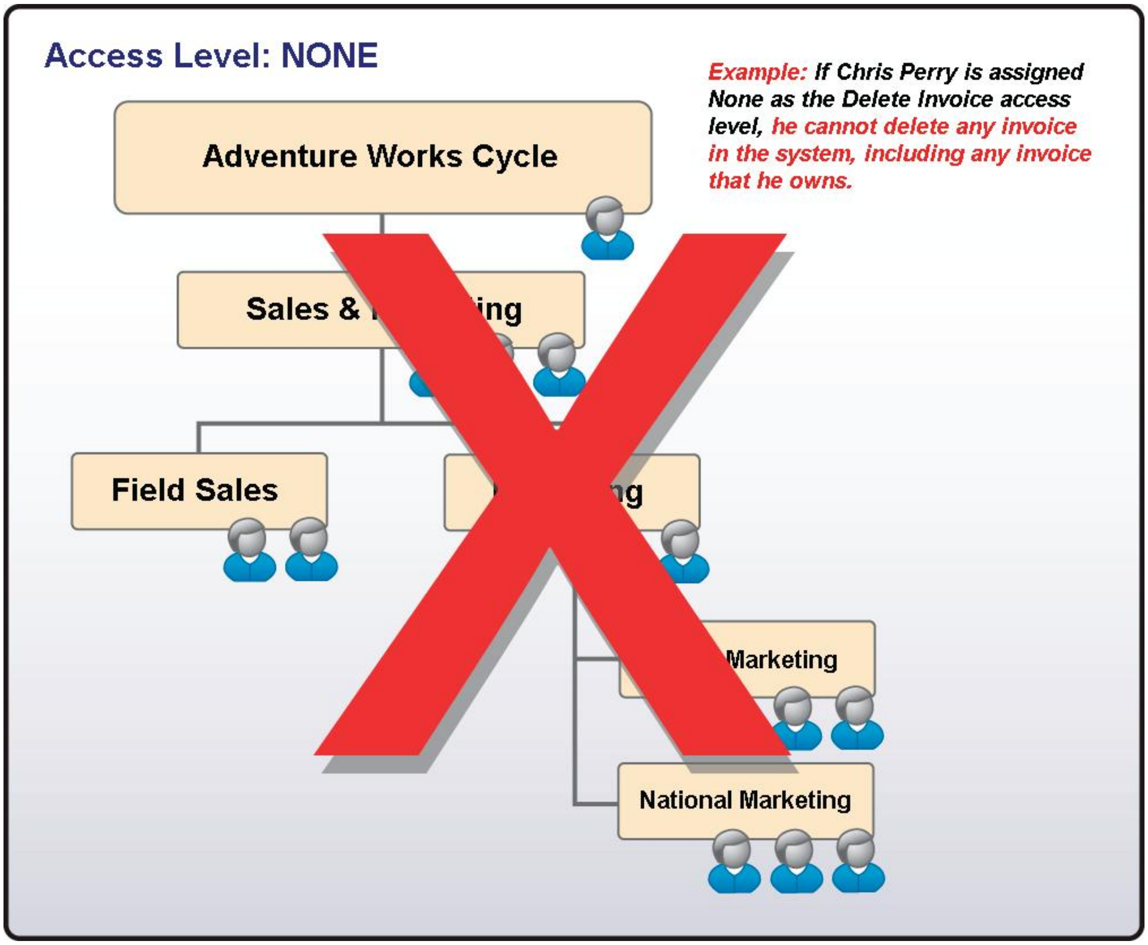
Access Level for a specific privilege and record type determines which records you can perform that action upon based upon the location of each record owner.

Access Level - None

The **None** access level restricts the user from performing an action on any records within that entity - *even on records owned by the user*. A privilege is not assigned to a security role if the access level is set to **None**. Conversely, a privilege is assigned to a role when the access level is changed from **None** to another value.

Example: Gail Erickson is the Sales Manager for Adventure Works' Western Region. Adventure Works has decided that there are some privileges the Sales Manager must be restricted from performing, such as creating, writing, and deleting Views. To guarantee this, the System Administrator creates a copy of the default Sales Manager role and assigns the **None** access level to the Create, Write, and Delete privilege for the Views entity. Gail is assigned this new, customized role instead of the default Sales Manager role.

Another example is provided in the following graphic:


Access Level: NONE
Example: If Chris Perry is assigned None as the Delete Invoice access level, he cannot delete any invoice in the system, including any invoice that he owns.
Adventure Works Cycle
Sales &
ing
Field Sales
g
Marketing
National Marketing

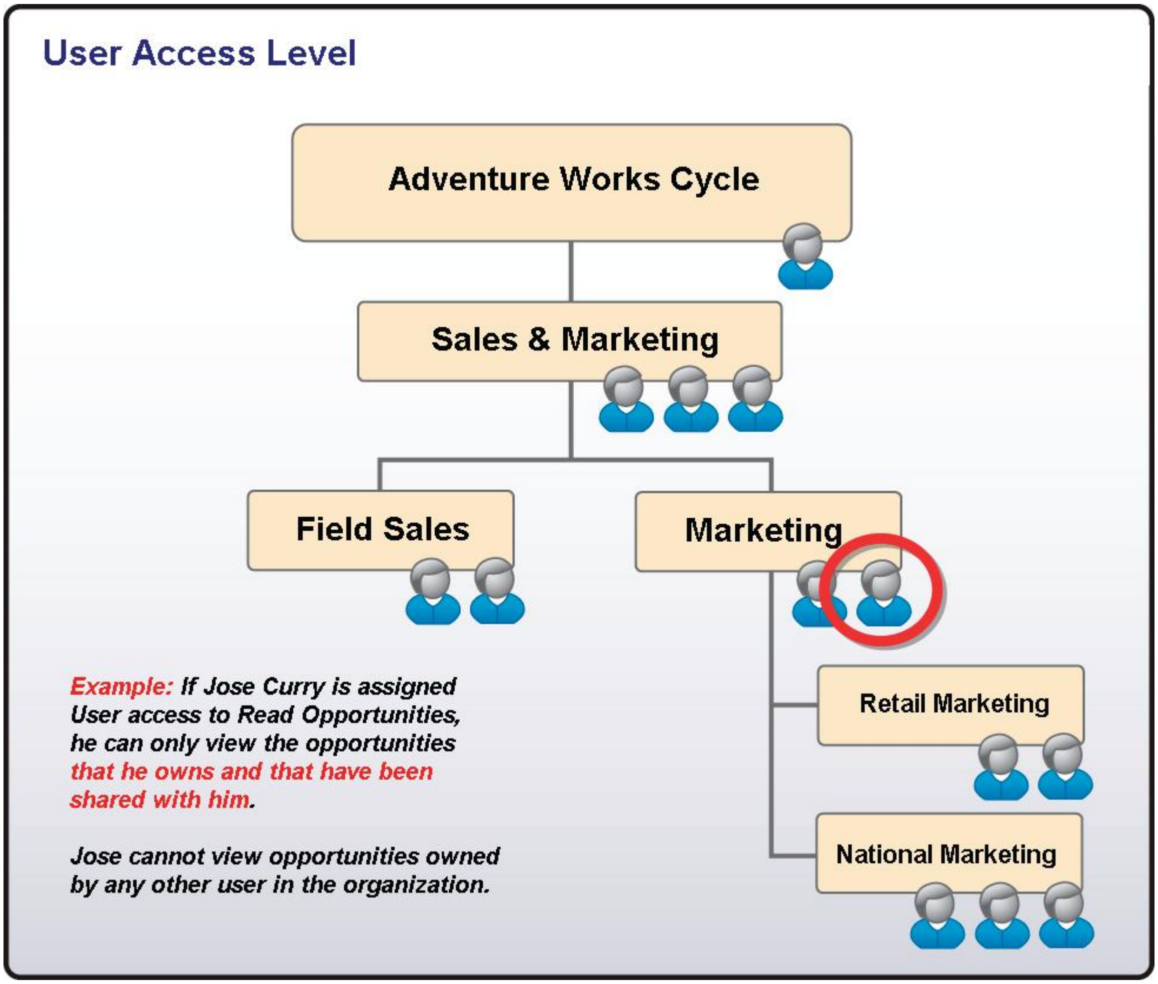
Access Level - User

Except for the **None** access level, **User** access is the most restrictive of the remaining levels that provide some form of access. If your role provides **User** access for a specific entity and privilege, you can only perform that action on the following records for that particular entity:

- Records you own
- Records owned by someone else but are shared with you
- Records shared with a team in which you are a member

Example: In Adventure Works Cycle, Douglas Hite is a Customer Service Representative in the Customer Support business unit. Douglas has "User Account Create" and "User Account Write" access. The **User** level access for these two privileges enables Douglas to create new Accounts and edit (change) any records that are assigned to him, shared with him by other users, or shared with any team in which he is a member.

Another example is provided in the following graphic:



Example: If Jose Curry is assigned User access to Read Opportunities, he can only view the opportunities that he owns and that have been shared with him.

Jose cannot view opportunities owned by any other user in the organization.

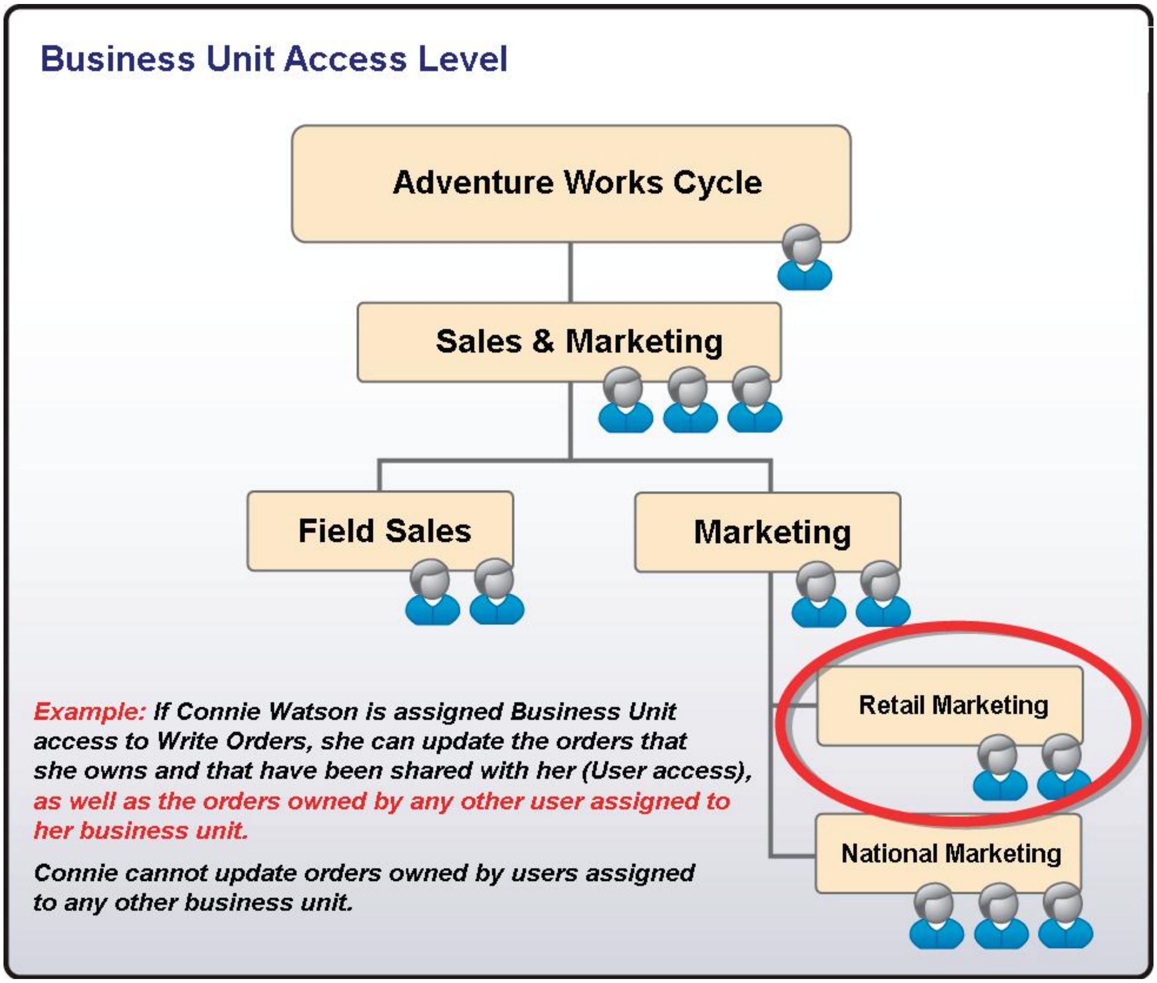
Access Level - Business Unit

Business Unit access is the next step up from **User** level access. **Business Unit** access for a specific entity and privilege gives you the following:

- User access rights
- Access to records owned by or shared with other users assigned to the same business unit as you

Example: Stefan DelMarco is the Customer Support Manager at Adventure Works Cycle. He manages the Customer Service representatives and is required to assign and review all accounts and cases assigned to these representatives. Assigning him "Business Unit Case Create" access enables him to create cases for any customer assigned to the Customer Support business unit. Similarly, if Stefan has "Business Unit Account Delete" access, he can delete any Account record that is owned by him or any user who is assigned to the Customer Support Business Unit.

Another example is provided in the following graphic:



Example: If Connie Watson is assigned Business Unit access to Write Orders, she can update the orders that she owns and that have been shared with her (User access), as well as the orders owned by any other user assigned to her business unit.

Connie cannot update orders owned by users assigned to any other business unit.

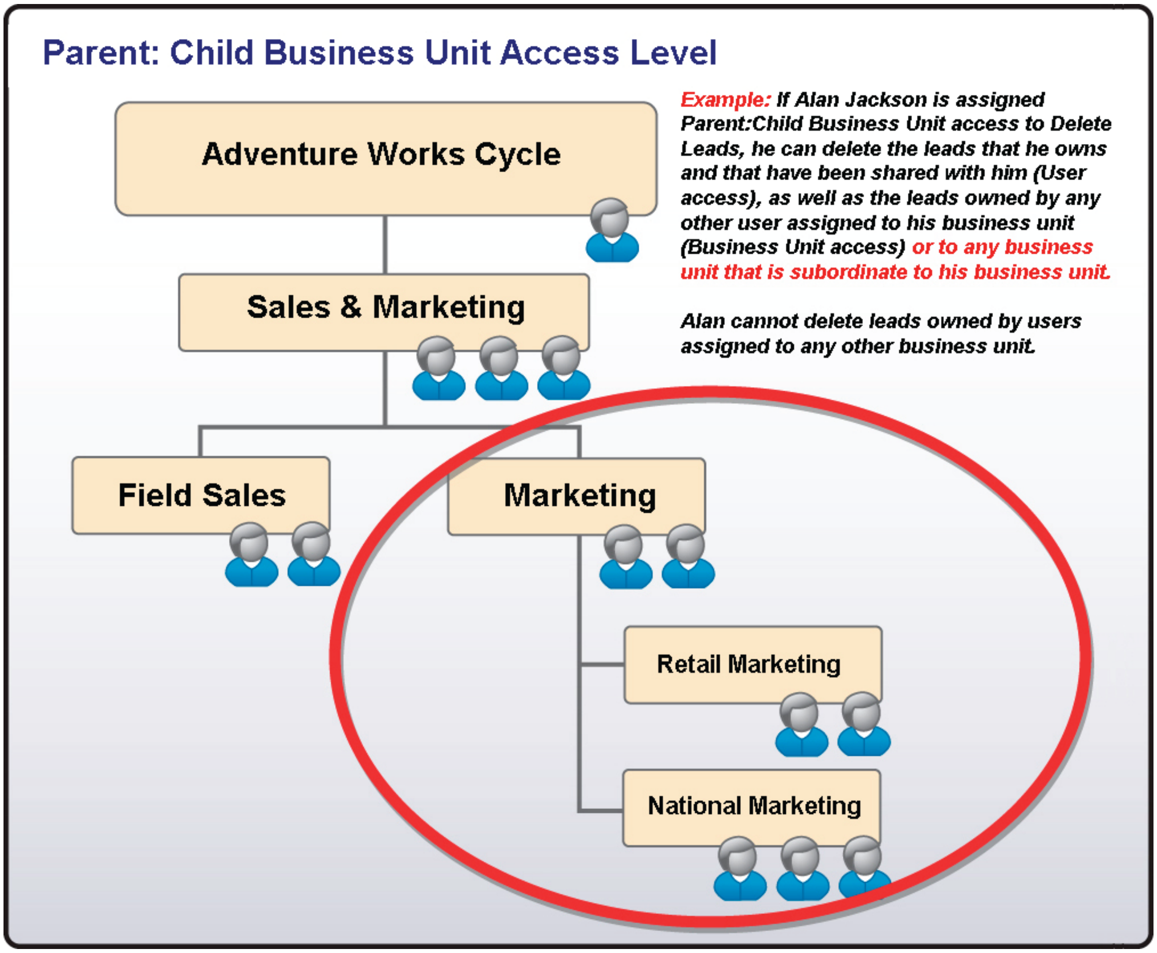
Access Level – Parent:Child Business Unit

Parent:Child Business Unit access is the next step up from **Business Unit** access. **Parent:Child Business Unit** access for a specific entity and privilege gives you the following:

- User and Business Unit access rights.
- Access to records owned by users and shared with users who are assigned to any business unit subordinate to your business unit, regardless of how deep in the organizational structure the user's business unit appears.

Example: Mary Baker is VP of Sales and Marketing for Adventure Works Cycle. She manages all the Sales and Marketing representatives for the Field Sales and Marketing Divisions. By assigning Mary "Parent:Child Opportunity Read" access, she can view all opportunities owned by any user assigned to the Sales & Marketing business unit or any one of its child business units. Because the Adventure Works Cycle, Customer Care, Customer Support, and OEM Support business units are not subordinate to Mary's business unit, she cannot view opportunities owned by users assigned to those business units.

Another example is provided in the following graphic:



Parent: Child Business Unit Access Level

Example: If Alan Jackson is assigned Parent:Child Business Unit access to Delete Leads, he can delete the leads that he owns and that have been shared with him (User access), as well as the leads owned by any other user assigned to his business unit (Business Unit access) or to any business unit that is subordinate to his business unit.

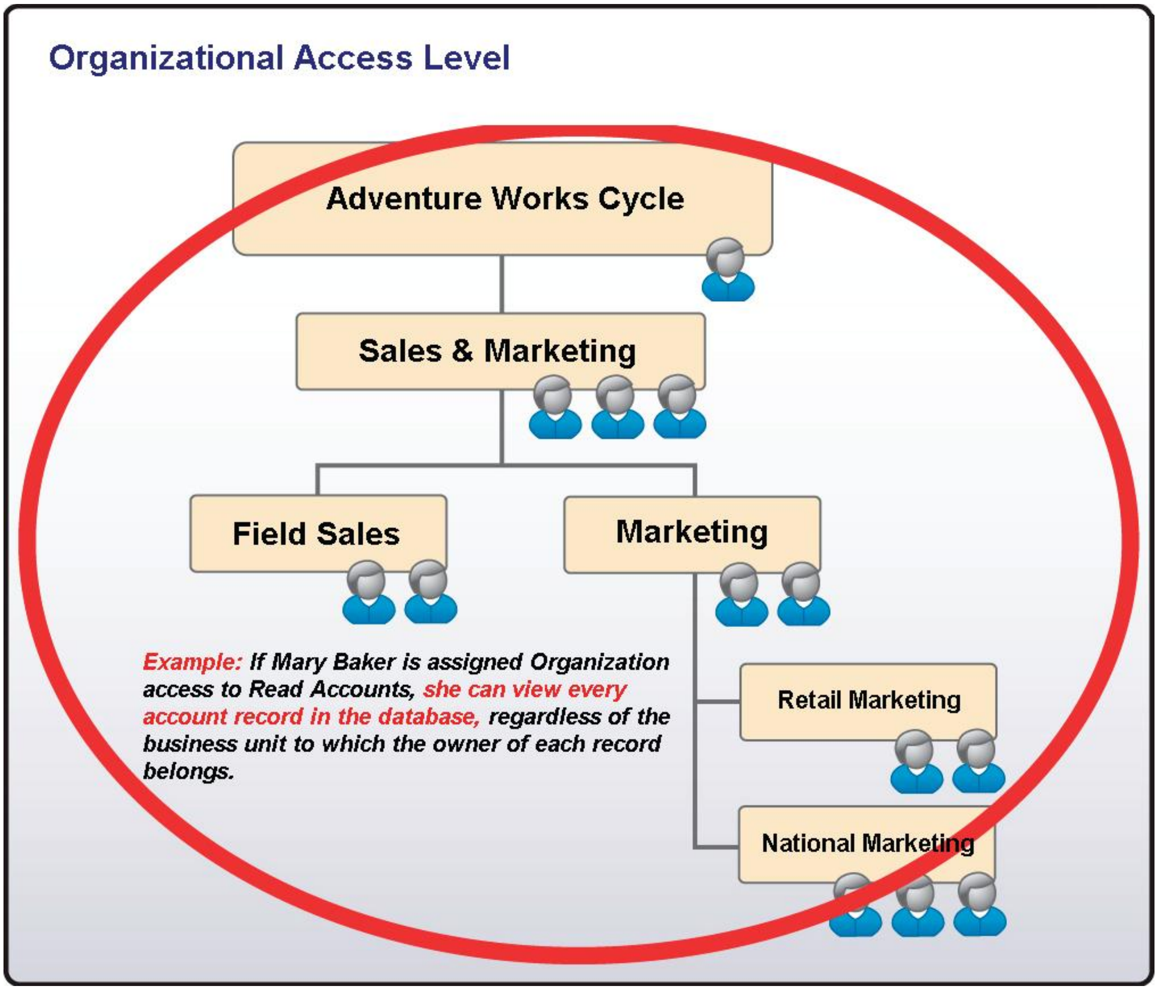
Alan cannot delete leads owned by users assigned to any other business unit.

Access Level - Organization

Organization access is the least restrictive of all access rights. **Organization** access for a specific entity and privilege allows you to perform that action on records *owned by any user within the entire organization*, regardless of the business unit to which the owner belongs. There are no access restrictions with **Organization** access.

Example: David Lawrence is the System Administrator for Adventure Works Cycle. He requires the ability to reassign ownership of any record in the system, regardless of the business unit to which the owner of the record belongs. If his System Administrator role gives him Organization Lead Assign access, David can reassign any lead that is entered in the system, regardless of who owns the record.

Another example is provided in the following graphic:



Example: If Mary Baker is assigned Organization access to Read Accounts, she can view every account record in the database, regardless of the business unit to which the owner of each record belongs.