# 

Microsoft BI Authentication and Identity Delegation

**Summary:** From straightforward client/server designs to complex architectures relying on distributed Windows services, SharePoint applications, Web services, and data sources, Microsoft BI solutions can pose many challenges to seamless user authentication and end-to-end identity delegation. SQL Server technologies and data providers expect to use Windows authentication while SharePoint Server uses Web Services Security (WS-Security). Flowing a user identity from a Windows or browser-based BI client application through a claims-based SharePoint service to a Windows backend system is not always possible due to various limitations in data providers, security protocols, and identity services. Network, forest, and federation topologies also influence the authentication flows. Familiarity with the authentication protocols and capabilities, delegation limitations, and possible workarounds is an indispensable prerequisite to delivering a positive BI user experience across the entire Microsoft BI solution stack in enterprise environments.

**Writer:** Kay Unkroth

**Technical Reviewers:** Alex Shteynberg, Andy Wu, Ariel Netz, Ashvini Sharma, Carolyn Rowe, Dave McPherson, Dave Wickert, David Hill, David Maguire, David Vugteveen, Denise Stendera, Diego Oppenheimer, Donny Rose, Dotan Elharrar, Frederic Gisbert, Guy Alroy, Haroon Ahmed, Haydn Richardson, Heidi Steen, Ileana Koller, Il-Sung Lee, Irina Gorbach, James Wu, Joanne Hendrickson, Joe Davies, John Hancock, John Sirmon, Kedar Dubhashi, Kevin Donovan, Mey Meenakshisundaram, Michiko Short, Mike Plumley, Nathaniel Scharer, Nick Simons, Nicolas Menigon, Parul Manek, Peter Brundrett, Prash Shirolkar, Reddy Duggempudi, Rob Lefferts, Robert Bruckner, Robert Skoglund, Sesha Mani, Sunil Gottumukkala, T.K. Anand, Uval Blumenfeld, V.B. Balayoghan, Venky Krishnan, Venky Veeraraghavan, Wayne Clark, Yair Tor, Craig Guyer

**Published:** January 2015

**Applies to:**

Microsoft Office 2013

Microsoft SQL Server 2012 Service Pack 1 (or higher)

Microsoft SQL Server 2014

Copyright

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.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2014 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Office 2013, Microsoft SharePoint Server, Microsoft SQL Server 2012, Microsoft SQL Server 2014, and Microsoft Windows Azure are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

[Introduction 5](#_Toc345331181)

[Personal BI Scenarios in Excel 6](#_Toc345331182)

[Client/Server Connections to Import Data 6](#_Toc345331183)

[Data Sources and their Authentication Methods 7](#_Toc345331184)

[Team BI Scenarios in SharePoint 9](#_Toc345331185)

[Multi-Tier Application Architecture 9](#_Toc345331186)

[Classic-Mode versus Claims-Mode Authentication 9](#_Toc345331187)

[Identity Delegation within a SharePoint Farm 10](#_Toc345331188)

[Client Authentication Issues 11](#_Toc345331189)

[Workbooks as a Data Source 12](#_Toc345331190)

[PowerPivot Gallery 13](#_Toc345331191)

[Anonymous Users 14](#_Toc345331192)

[Mitigating Client Authentication Issues 14](#_Toc345331193)

[Backend Authentication Issues 15](#_Toc345331194)

[Accessing Relational Configuration, Content, and Service Application Databases 16](#_Toc345331195)

[Authentication against External Data Sources 17](#_Toc345331196)

[Kerberos Constrained Delegation and Alternatives 22](#_Toc345331197)

[Accessing Analysis Services in SharePoint Mode 26](#_Toc345331198)

[Corporate BI Data Access Scenarios 30](#_Toc345331199)

[Authentication Flows in Native-Mode Reporting Services 30](#_Toc345331200)

[Windows User Authentication 31](#_Toc345331201)

[Custom User Authentication 31](#_Toc345331202)

[Custom Data Source Authentication 32](#_Toc345331203)

[Authentication Flows in Analysis Services 32](#_Toc345331204)

[Multidimensional and Tabular Data Access 32](#_Toc345331205)

[ROLAP and DirectQuery Mode 33](#_Toc345331206)

[Client Authentication over HTTP 34](#_Toc345331207)

[Custom Analysis Services Authentication over HTTP 35](#_Toc345331208)

[Federated BI Scenarios 36](#_Toc345331209)

[Multi-Forest Active Directory Environments 37](#_Toc345331210)

[Extranet Environments 38](#_Toc345331211)

[Cloud Environments 39](#_Toc345331212)

[Conclusion 40](#_Toc345331213)

# Introduction

The term Microsoft BI refers to a stack of Microsoft Office and Microsoft SQL Server technologies for data analysis and reporting. These products and technologies provide the foundation for personal, team, and corporate BI solutions that can integrate multiple distributed Windows services, SharePoint applications, Web services, and data sources so seamlessly with each other that it is often impossible for users to distinguish the various components in the overall solution architecture. Figure 1 shows the Microsoft BI reference model with the relevant products and technologies.

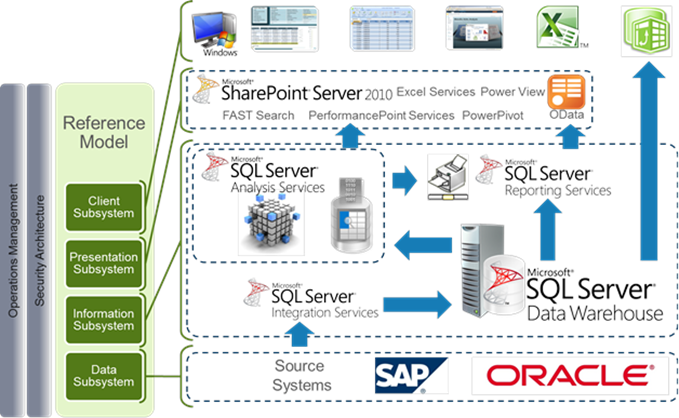


Figure 1   Microsoft BI Reference Model

The depicted reference model groups the various products and technologies into separate subsystems, but it is important to note that these subsystems do not necessarily correspond to specific tiers in a multi-tier application. BI products and technologies often serve multiple purposes, acting as systems in the client, middle, or backend tier relative to each other. For example, a Microsoft BI solution might require authentication against multiple services, while the services in turn might have to delegate the user identity to further backend systems in order to consolidate data from multiple sources. Some typical scenarios include:

* A Reporting Services report exposing a data feed combining data from multiple databases or tables into a consolidated data set.
* A SharePoint farm hosting a PowerPivot site while at the same time acting as a data source for PowerPivot workbooks that import content from a SharePoint list.
* A computer running SQL Server in a big data solution that uses a linked server and views to connect Analysis Services to a Microsoft HDInsight-based data warehouse.

The main point to take away from this flexibility is that all components and subsystems in a BI solution might have to delegate a user identity to yet another system, and every hop in the solution architecture can pose authentication and delegation challenges. In complex enterprise environments, seamless integration is sometimes challenging because the various components and subsystems might use different security protocols with different delegation capabilities and limited interoperability. Some workarounds to circumvent identity delegation barriers exist, such as stored credentials and unattended service accounts, yet these workarounds are not always applicable or addressing the full spectrum of the issues.

This technical reference guide provides an overview of the typical authentication and identity delegation requirements and limitations in distributed BI solutions based on Office 2013 and SQL Server 2012 Service Pack 1. The explanations are for technical BI professionals responsible for strategy briefings and architecture designs, delivering proof of concepts (POCs) across the breadth of Microsoft BI. The purpose of this document is to help these BI professionals to recognize integration capabilities and limitations when proposing and validating BI solution designs and system architectures. This guide assumes that the audience already has some knowledge of the Microsoft BI stack and experience building BI solutions.

# Personal BI Scenarios in Excel

Personal BI solutions enable business users to perform analysis and reporting on their own in Microsoft Excel. Through PivotTables, PivotCharts, and embedded Power View report sheets, users can combine and analyze data from a variety of sources, including relational databases, multi-dimensional or tabular data models, data feeds, SharePoint lists, and data files. It is also possible to simply copy and paste tabular data into the data model of an Excel workbook.

## Client/Server Connections to Import Data

From an Excel client perspective, a data connection is just a single hop from the client to the server. This single hop has no special authentication or delegation requirements, as long as the client is able to use the security protocols that the server supports, such as Basic, NTLM, or Kerberos. Figure 2 shows the typical data access paths.

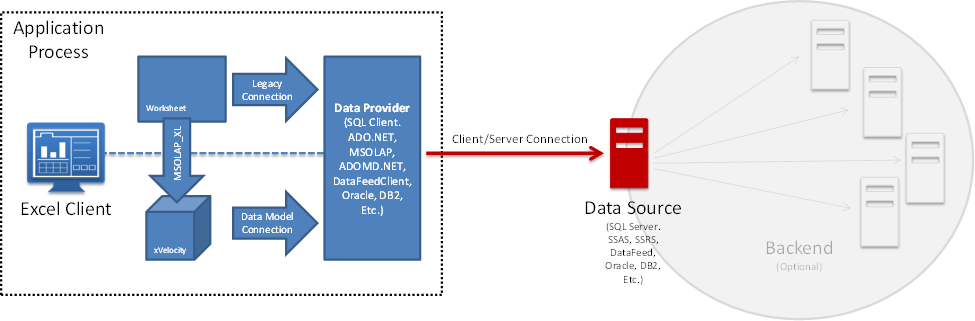


Figure 2   Client/server connections to import data into Excel

Internally, Excel 2013 supports legacy data connections as well as embedded xVelocity data connections, yet regardless of the connection type, the client typically uses a data provider or client library to access a given data source. The data source might encapsulate further systems in the backend, yet this is transparent to the client. Personal BI solutions simply use a straightforward client/server connection to access a data source as a single entity. Integrating backend systems with each other is covered later in this document in the context of team and corporate BI solutions.

**Note:** Network and forest boundaries influence what security protocols a client can use. For example, firewalls can block an external client from communicating with an internal domain controller or Kerberos key distribution center (KDC), in which case the client cannot use NTLM or Kerberos for authentication. Basic authentication can provide a workaround in this situation, as discussed in more detail under “Client Authentication over HTTP” later in this document.

## Data Sources and their Authentication Methods

The authentication method that the client must use depends on the selected data source. Loaded into the application process, a data provider or client library encapsulates the authentication and communication protocols to interact with the backend system. For example, SQL Server supports Windows as well as SQL authentication. Accordingly, the SQL Server Native Client library supports connection strings that specify *Integrated Security=SSPI* for Windows authentication and *User Id=UserName; Password=Pwd;* for SQL authentication.

If a client connects to multiple different data sources, such as SQL Server and Analysis Services, multiple data providers must be loaded into the Excel application process, and multiple authentication procedures must be completed to authenticate the user against each system. Table 1 summarizes the data sources that are relevant for personal BI solutions and their authentication methods.

Table 1   Data sources and their authentication methods

|  |  |  |
| --- | --- | --- |
| Data Source | Authentication Methods | Comments |
| SQL Server Relational Engine | Windows Authentication  User name and password on the connection string. | Can be accessed through SQL Server Native Client and ADO.NET. |
| Parallel Data Warehouse | SQL Authentication | Same as SQL Server Relational Engine, except that Windows Authentication is not available. |
| Windows Azure SQL Database | SQL Authentication | Same as SQL Server Relational Engine, except that Windows Authentication is not available. |
| Third-Party (Oracle, Teradata, Sybase, Informix, IBM DB2, and others) | User name and password on the connection string. | Can be accessed through a third-party OLE DB or ODBC data provider. |
| Apache Hadoop/Hive or Windows Azure HDInsight | User name and password on the connection string. | Can be accessed through Microsoft ODBC Provider for Hive. |
| SQL Server Reporting Services | Windows Authentication  Forms Authentication | Can be accessed via HTTP. It is also possible to connect to a report as a data feed through the DataFeed provider. |
| SQL Server Analysis Services | Windows Authentication | Can be accessed through MSOLAP, ADOMD.NET, and AMO. These data providers support TCP/IP and HTTP connections. HTTP support is provided through IIS and the Analysis Services data pump (MSMDPump). |
| Excel workbooks with embedded PowerPivot data models uploaded to a BI-enabled SharePoint farm. | Windows Authentication | Same as Analysis Services. Note that SharePoint supports Windows authentication, forms-based authentication, and SAML-based authentication, but the Analysis Services data providers only support Windows authentication. |
| Microsoft Access databases | Windows Authentication | Can be accessed through the Microsoft ACE OLEDB provider. If the database file resides on a file system that requires authentication, such as a local NTFS drive or a file share, the provider attempts to access the file in the security context of the currently logged on user. |
| Plain-text tabular data, such as Excel spreadsheets and comma- separated values (CSV) files. | Windows Authentication | Same as Access databases. |
| Web content datasets from public and private data providers on the Internet via Windows Azure Marketplace DataMarket. OData, REST and JSON-based Web Services from Twitter, Facebook, and others. | Windows Authentication  Basic Authentication  Certificate-Based Authentication  Claims-Based Authentication | Can be accessed through the DataFeed provider, which supports all common Web authentication methods, including claims-based authentication. |
| Content and metadata from SharePoint lists and document libraries. | Same as Web Content Datasets | Same as Web Content Datasets |
| Cut and paste tables from Web pages into data models. | Not applicable | These tables are static in the data model, so there is no data source that the client could connect to in order to refresh the data. |

# Team BI Scenarios in SharePoint

A user can quickly turn a personal PowerPivot workbook into a team BI solution by sharing it with others in SharePoint. When uploading a PowerPivot workbook to a document library, the solution design effectively changes from a two-tier client/server application to a multi-tier Web solution. Among other things, users can now access and interact with the workbook and its data model in a browser.

## Multi-Tier Application Architecture

As Figure 3 illustrates, a team BI solution effectively relies on three separate tiers for clients, SharePoint, and backend systems. Web applications on SharePoint Web Front-End (WFE) servers authenticate the clients and render the Web pages. The WFEs in turn communicate with SharePoint shared services running on application servers in the farm to retrieve the workbook content, which in turn might access backend systems outside the local farm to retrieve additional data. It is primarily the communication with the backend that can cause integration issues because user identities cannot always be delegated to external systems.

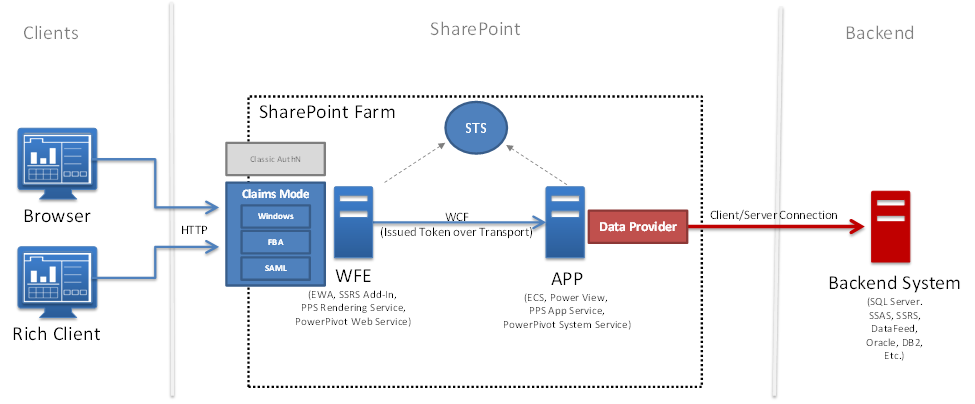


Figure 3   Team BI solution architecture

## Classic-Mode versus Claims-Mode Authentication

In the client tier, users can work with browsers or rich clients, which communicate with SharePoint Web applications on WFEs over HTTP(S). SharePoint Web applications support a variety of authentication methods, including classic-mode Windows authentication for backward compatibility as well as claims-mode Windows authentication, ASP.NET forms-based authentication, and SAML passive sign-in.

Classic-mode Windows authentication is deprecated in SharePoint 2013. It is no longer possible to configure a classic-mode Web app in SharePoint Central Administration. Administrators must use the *New-SPWebApplication* PowerShell cmdlet if a Web app must be provisioned for classic-mode authentication. However, claims-mode Windows authentication provides the same capabilities as classic-mode authentication and offers additional flexibility, such as support for multiple authentication methods on a single Web application. For this reason, and because some Office integration features no longer work with classic-mode Web apps, it is highly recommended to configure SharePoint 2013 Web applications for claims-mode authentication. For details, see the topics under “*Configure Authentication Infrastructure in SharePoint 2013*” on Microsoft TechNet at <http://technet.microsoft.com/en-us/library/jj219795.aspx>.

**Note**   In SharePoint 2010, PowerPivot for SharePoint requires classic-mode Windows authentication, yet this has changed in SharePoint 2013. In SharePoint 2013, all Microsoft BI technologies support claims-based authentication, but PowerPivot supports Windows claims-mode authentication only, despite that fact that SharePoint in claims mode also supports forms-based and SAML authentication methods.

## Identity Delegation within a SharePoint Farm

Within a SharePoint farm, independent of the incoming client authentication method, Web apps and shared services can seamlessly exchange identity information. All Web apps and shared services trust the farm’s security token service (STS) and use the *IssuedTokenOverTransport* authentication mode of Windows Communication Foundation (WCF) for intra-farm communication. The SharePoint STS is implemented as a WCF service, located in the *%ProgramFiles%\Common Files\Microsoft Shared\Web Server Extensions\15\ISAPI\STS* folder.

Whenever a user authenticates on a WFE, the SharePoint authentication module relies on the SharePoint STS to generate a claims-based security token. This SharePoint security token proves that the user was successfully authenticated and conveys the user’s identity information as claims. Windows account information is converted to Windows claims, forms-based membership information is transformed into forms-based claims, and claims from a SAML token are copied into the SharePoint security token according to the claims mappings that the SharePoint administrator defined for the identity token issuer.

In short, regardless of the incoming authentication method, every user is represented by a claims-based security token in SharePoint, and front-end Web apps can delegate this security token to shared services by using *IssuedTokenOverTransport*. The caller simply presents the claims-based security token at the Simple Object Access Protocol (SOAP) level to the shared service and the shared service can then act on behalf of the authenticated user.

While identity delegation is generally unproblematic within a single SharePoint Web application, the situation changes as soon as a Web app or shared service must access another Web app or an external backend system. It is possible to configure server-to-server authentication between SharePoint 2013 farms, Exchange Server 2013, and Lync Server 2013 (as explained in “*Configure Server-To-Server Authentication in SharePoint 2013*” at <http://technet.microsoft.com/en-us/library/jj219532.aspx>), but this does not apply to SQL Server engines or third-party data sources. Later in this document, the section “*Backend Authentication Issues*” discusses identity delegation to external backend systems in more detail.

## Client Authentication Issues

On WFEs, claims-based authentication methods are integrated into the ASP.NET request processing pipeline. If a Web app is configured for Windows authentication, it takes advantage of the authentication methods in a Windows domain environment: NTLM, Kerberos, Basic, and Anonymous. Most browsers and rich clients support Windows authentication, but the user must have a Windows account. For non-Windows users, forms-based or SAML-token authentication can be configured. Figure 4 shows the authentication settings for a Web application that supports Windows, forms-based, and SAML users.

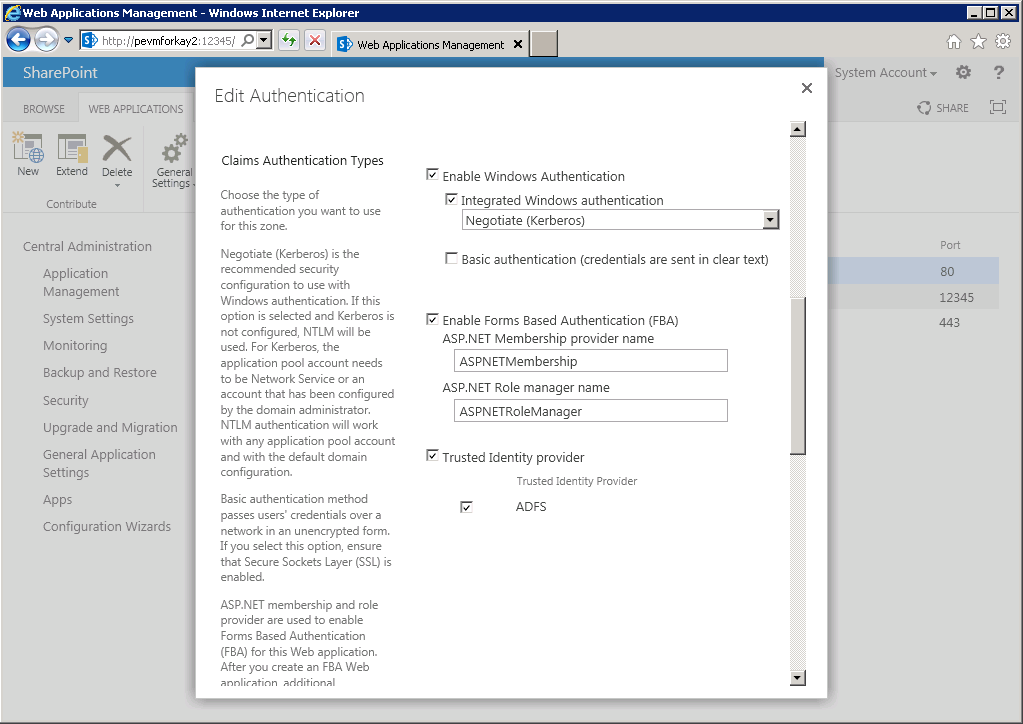


Figure 4   A Web app supporting multiple authentication types

As indicated in Figure 4, it is possible to support Windows and non-Windows users on the same Web app, but it is important to note that some PowerPivot features do not support non-Windows authentication methods and stop working if multiple authentication methods are enabled on the same Web application. Specifically, PowerPivot features that have issues include Workbooks as a Data Source and PowerPivot Gallery.

### Workbooks as a Data Source

Rich clients can access SharePoint content directly via a variety of protocols on top of HTTP(S), including SOAP, REST, WebDAV, SharePoint Remote Procedure Calls (RPCs), and File Synchronization via SOAP over HTTP Protocol. These protocols are able to handle all SharePoint authentication methods. However, rich clients do not use these protocols when accessing PowerPivot workbooks published in SharePoint as a data source.

Workbooks as a Data Source is a PowerPivot feature that enables Excel and other rich clients to access a PowerPivot workbook published in SharePoint through an Analysis Services data provider, such as MSOLAP, ADOMD.NET, or AMO, similar to a tabular Analysis Services database. The user only needs to specify the URL to the workbook as the data source and the data provider takes care of the underlying client redirection and XML/A communication. Yet, the Analysis Services data providers only support Windows authentication and are unable to deal with a selection form if the Web app supports multiple authentication methods. As a consequence, the Web app must only support Windows authentication and no other authentication methods.

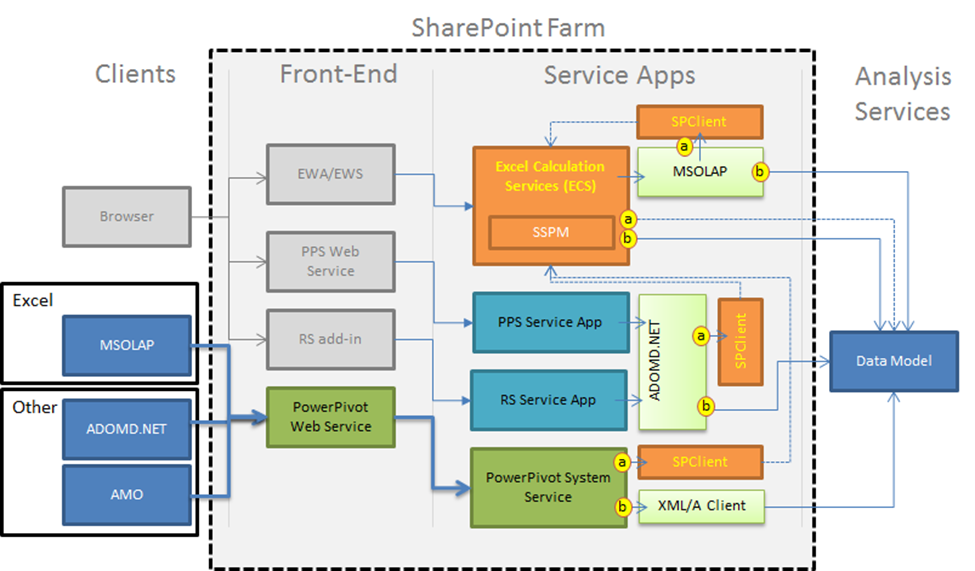


Figure 5   Workbooks as a Data Source

Figure 5 shows the Workbooks as a Data Source architecture. For clients **outside of the SharePoint farm**, the communication relies on the PowerPivot Web Service and the PowerPivot System Service to redirect the client queries and server responses, as follows:

1. The user specifies the URL of the desired PowerPivot workbook as the data source in the connection string and issues a request to open the connection.
2. The Analysis Services data provider derives the URL of the PowerPivot Web Service from the workbook URL and forwards the client request.
3. The WFE authenticates the Windows user and then the PowerPivot Web Service receives and forwards the request along with the authenticated user’s identity to the PowerPivot System Service.
4. The PowerPivot System Service, in turn, uses an SPClient component to communicate with Excel Calculation Services (ECS) in order to load the data model, again forwarding along the authenticated user’s identity.
5. ECS checks that the authenticated user has permissions to access the workbook, loads the data model on an Analysis Services server, and then returns the connection information to the PowerPivot System Service.
6. In a second step, the PowerPivot System Service establishes a direct connection to the data model using the connection information received from ECS.
7. At this point, all required connections are established to exchange requests and responses between the client application and the Analysis Services server.

Note that clients **within the farm**, such as PerformancePoint Services and Power View, work differently in the sense that the PowerPivot services are not involved in the communication path. Within the farm, Analysis Services data providers can use the SPClient component directly to communicate with ECS to load data models, retrieve the connection information, and then establish direct TCP/IP connections. Because SPClient uses the SharePoint object model, it delegates the authenticated user’s identity to ECS so that ECS can perform its access checks to verify that the user has permissions to load the data model.

**Note**   The authenticated user’s identity is not delegated to the Analysis Services server. The PowerPivot for SharePoint architecture relies on a trusted subsystem design as explained later under “Backend Authentication Issues.”

### PowerPivot Gallery

The PowerPivot Gallery is a document library type that provides interactive Silverlight-based previews and document management for published Excel workbooks and Power View reports. The previews are based on thumbnail images of worksheets and reports. For worksheets, the PowerPivot Gallery renders the thumbnails by using Excel Web Access. For Power View reports, the PowerPivot Gallery obtains the thumbnails directly from the report .rdlx files.

Particularly, the thumbnail rendering for Excel worksheets has issues with non-Windows authentication methods. When a user saves a workbook in a PowerPivot Gallery, SharePoint triggers an event, which PowerPivot Gallery intercepts to launch a Capture Snapshot process that generates a new thumbnail for every worksheet in the workbook. This unattended Capture Snapshot process must authenticate to access the workbook but is generally unable to deal with login pages, such as to enter user credentials in an ASP.NET sign-in form or to select a particular authentication method if multiple methods are enabled on the Web app.

The Capture Snapshot process is also unable to authenticate by using SAML passive sign-in because this authentication method requires a redirection of the client process to an external security token service, which the PowerPivot Gallery actively blocks so that an attacker cannot exploit the Capture Snapshot process to launch an attack against external targets. Consequentially, PowerPivot Gallery can only render and display thumbnail images for workbooks in a site collection or site that is hosted in a Web app configured for Windows authentication only.

### Anonymous Users

While SharePoint 2013 supports anonymous users accessing content without providing credentials, PowerPivot for SharePoint 2013 and Reporting Services Power View do not. Hence, anonymous authentication must not be enabled for Web apps that host PowerPivot workbooks and Power View reports. This authentication method is disabled by default.

### Mitigating Client Authentication Issues

SharePoint 2013 features a flexible Microsoft BI light-up story to provide services at varying levels to Windows and non-Windows users. Excel Services, PerformancePoint Services, and Reporting Services Power View support all forms of claims-mode authentication, including forms and SAML tokens. These services do not depend on PowerPivot for SharePoint. Hence, customers concerned about PowerPivot-related client authentication issues can opt out of deploying the PowerPivot add-in in their BI-enabled site collections without affecting other BI technologies.

Figure 6 illustrates the BI light-up story. Excel Services provides the core BI features, such as PowerPivot interactivity in the browser, as soon as an Analysis Services server is registered in the Excel Services configuration. This Analysis Services server runs in SharePoint mode but is installed outside the SharePoint farm. Other services, such as PerformancePoint Services and Power View, are immediately able to access workbooks as a data source. The PowerPivot for SharePoint 2013 add-in is only required for accessing workbooks as a data source from outside the farm, configuring scheduled data refresh for a workbook, using the PowerPivot Gallery, providing BISM link file support, and adding the PowerPivot Management Dashboard to SharePoint Central Administration.

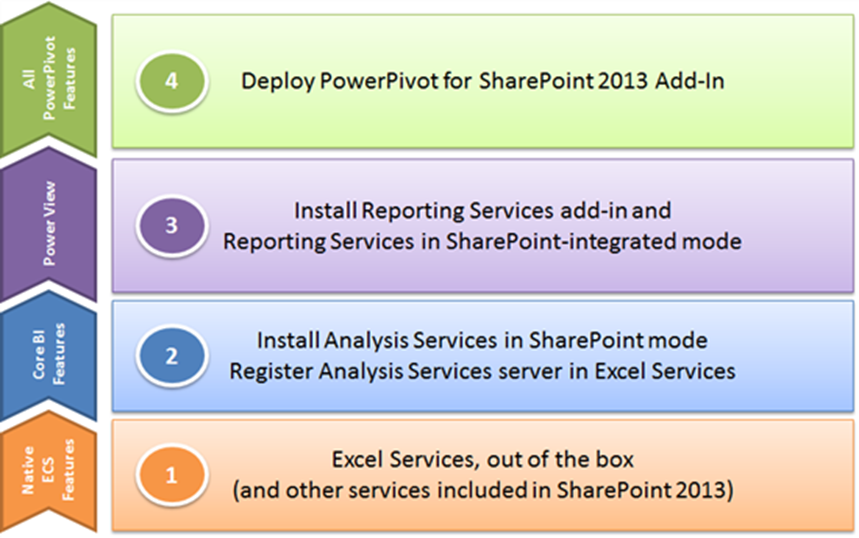


Figure 6   Microsoft BI light-up story for SharePoint 2013

If a site collection must provide advanced PowerPivot features (such as PowerPivot Gallery and Scheduled Data Refresh) to Windows, forms-based, and SAML-token users, the recommended approach is to extend the site collection’s Web application over two separate IIS Web sites that expose the same content. Windows and non-Windows users can then access the site collection using different URLs. One of the IIS Web sites should be configured only for Windows authentication so that Windows users can use all PowerPivot features. The other IIS Web site can then be configured for forms-based authentication and SAML passive sign-in with limited access to BI features. Figure 7 illustrates this configuration. For detailed configuration steps, refer to Microsoft TechNet article “*Extend Claims-Based Web Applications in SharePoint 2013*” at <http://technet.microsoft.com/en-us/library/gg276325(v=office.15)>.

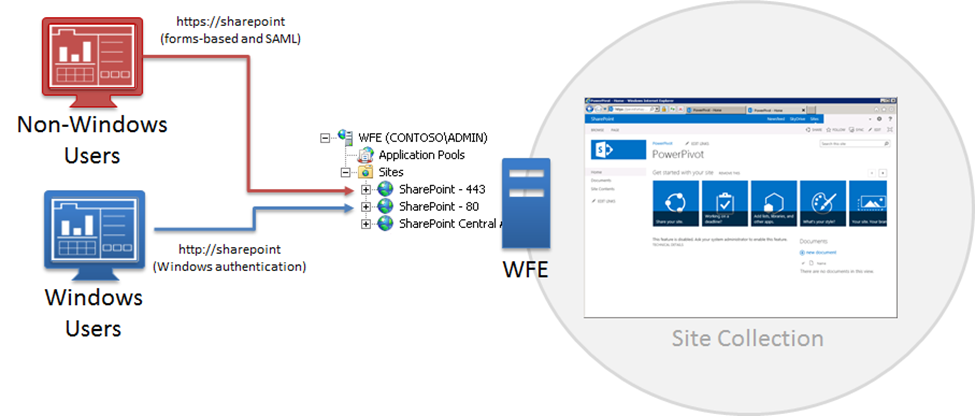


Figure 7   *Surfacing BI content to different types of users*

## Backend Authentication Issues

In SharePoint 2013, a BI-enabled farm relies on various types of backend systems with different authentication and identity delegation requirements:

* **Relational SQL Server Engine**   SharePoint 2013 servers rely on a relational instance of SQL Server 2008 R2 or SQL Server 2012 to host the configuration, content, and service application databases. User identities are not delegated to these relational databases. SharePoint services access these databases by using their own service identities.
* **External Data Sources**   Includes all data sources that BI solutions in a SharePoint farm might need to access for analysis and reporting. These are the same data sources covered under “Personal BI Scenarios in Microsoft Excel” earlier in this paper. In many cases, user credentials must be delegated to these data sources.
* **Analysis Services in SharePoint Mode**   Enables Excel Services and other SharePoint services to load, query, and process PowerPivot data models. User identities are not delegated to these BI servers. SharePoint services establish connections to PowerPivot data models by using their own service identities and then send XML/A commands to Analysis Services to communicate user information. The user information is transferred as a string for logging purposes, but not to make authorization decisions.

### Accessing Relational Configuration, Content, and Service Application Databases

The SharePoint deployment tools and wizards, such as SharePoint Configuration Wizard, Farm Creation Wizard, and PowerPivot Configuration Tool, automatically configure SharePoint account permissions and security settings for configuration, content, and service application databases in the relational SharePoint backend. For details, refer to the Microsoft TechNet article “*Account Permissions and Security Settings in SharePoint 2013*” at <http://technet.microsoft.com/en-us/library/cc678863.aspx>.

In a secure SharePoint environment, users do not have network access to configuration, content, or service application databases. Instead, they typically consume relational content indirectly. The PowerPivot architecture for Scheduled Data Refresh, shown in Figure 8, illustrates this design.

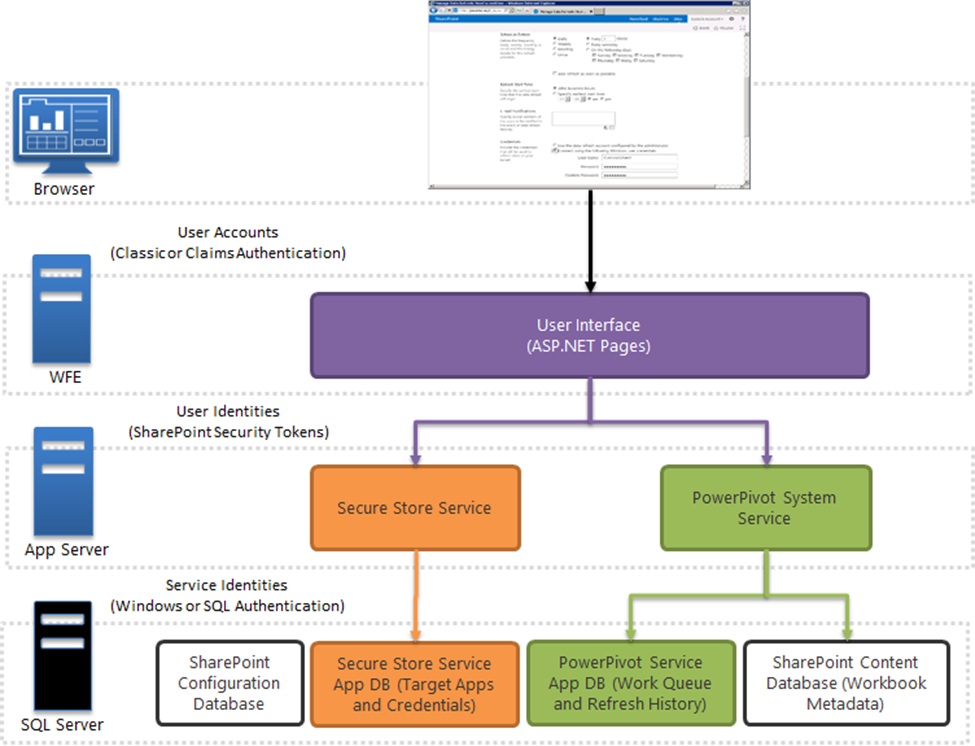


Figure 8   *Accessing content in service application databases*

The ASP.NET pages, hosted on WFEs, implement the user interface displayed in the browser. These ASP.NET pages must save and retrieve stored credentials and schedule information for PowerPivot workbooks, but instead of accessing this information directly, the ASP.NET pages use the Secure Store Service and PowerPivot System Service. These shared services receive the requests, verify that the user has the required permissions, and then carry out their tasks under their own identities. Access permissions are applied at the level of Secure Store target applications, workbooks, document libraries, and sites within SharePoint, which frees the relational backend from having to deal with user identities and individual access controls.

### Authentication against External Data Sources

External data sources are different from the relational database backend of SharePoint in the sense that users typically want to access the data sources with their own credentials. BI solutions can perform analysis and reporting by directly connecting to external data sources or by first importing the data into an embedded xVelocity data model. Direct connectivity helps to avoid data refresh operations whenever the data source is updated with new information, which can be time consuming. On the other hand, importing data into an embedded model provides best performance and modeling flexibility. It can also lessen a BI solution’s dependency on external data connectivity, provided alternative means are available to refresh the data, such as by re-uploading a workbook that was manually refreshed on a client computer. Embedded data models are available in BI solutions based on Excel workbooks. For technologies that do not support xVelocity data models, another option for external data sources is to use Excel workbooks. For more information, see the section “[Workbooks as a Data Source](#_Workbooks_as_a)”.

#### Direct Connectivity to Data Sources

Table 2 summarizes the connectivity and authentication capabilities for Microsoft BI-related SharePoint shared services, namely Excel Services, PerformancePoint Services, Visio Services, and Power View. Note that these shared services support similar authentication settings, although the way to store credentials and the range of supported data sources varies to some degree from service to service.

Table 2   Authentication capabilities of SharePoint shared services

|  |  |  |
| --- | --- | --- |
| Shared Service | Data Source | Authentication Method |
| Excel Services | SharePoint lists | SharePoint user identity |
| Excel workbooks | SharePoint user identity |
| SQL Server | Windows user identity (integrated security), Secure Store Service, unattended service account, or SQL authentication |
| Analysis Services | Windows authentication (integrated security), Secure Store Service, or unattended service account |
| Third-party OLE DB/ODBC/ADO.NET | User-name and password in the connection string. |
| Visio Services | SharePoint lists | SharePoint user identity |
| Excel workbooks | SharePoint user identity |
| SQL Server | Windows user identity (integrated security), Secure Store Service, unattended service account, or SQL authentication |
| Third-party OLE DB/ODBC/ADO.NET | User-name and password in the connection string. |
| PerformancePoint Services | SharePoint lists | SharePoint user identity |
| Excel workbooks | SharePoint user identity |
| SQL Server | Windows user identity (integrated security), Secure Store Service, unattended service account, or SQL authentication |
| Analysis Services | Windows authentication (integrated security), Secure Store Service, or unattended service account |
| Power View | Workbook data models | SharePoint user identity |
| Analysis Services | Windows authentication (integrated security) or credentials stored in a Reporting Services data source (RSDS) file |
| Reporting Services in SharePoint-integrated mode | SharePoint lists | SharePoint user identity |
|  | SQL Server | Windows user identity (integrated security) or credentials stored in a Reporting Services data source (RSDS) file |
|  | Windows Azure SQL Database | Credentials stored in a Reporting Services data source (RSDS) file |
|  | Analysis Services | Windows user identity (integrated security) or credentials stored in a Reporting Services data source (RSDS) file |
|  | XML Web Services or documents | Windows user identity (integrated security) or no credentials |
|  | Oracle, SAP NetWeaver BI, Hyperion Essbase, or other third-party OLE DB/ODBC/ADO.NET | Credentials stored in a Reporting Services data source (RSDS) file |

#### Authentication Settings for Data Sources that Support Windows Authentication

Excel users connecting a workbook to a data source by using Windows authentication (*Integrated Security=SSPI*), such as a SQL Server database or a tabular Analysis Services data model, can configure Excel Services authentication settings in the workbook for each individual data connection. These settings determine how Excel Services connects to the data source after the workbook is uploaded to SharePoint. There are three options available in the Excel Services Authentication Settings dialog box, which can be displayed by clicking on Connections in Excel, on the Data tab, then selecting the desired connection, clicking Properties, switching to the Definition tab, and then clicking on Authentication Settings:

* **Use the authenticated user’s account**   This is the default. It instructs Excel Services to impersonate the authenticated user. SharePoint authenticates the user by using Windows authentication (NTLM or Negotiate) and creates a SharePoint security token. As long as the inbound authentication method was Windows, Excel Services then translates the SharePoint security token back into a Windows identity by using Claims to Windows Token Service (C2WTS). Excel Services then connects to the external data source in the context of the Windows user identity. Because C2WTS returns an impersonation-level Windows token, Kerberos constrained delegation (KCD) must be configured between the C2WTS and the external data source (unless the data source is hosted locally on the server, which is typically not the case).
* **Use a stored account**   Instructs Excel Services to retrieve the user name and password in the context of the authenticated user from the specified Target Application ID in Secure Store Service. Excel Services then uses the Windows *LogonUser* function to authenticate the user, which eliminates the need for Kerberos constrained delegation to access the external data source.
* **None**   Instructs Excel Services to use its unattended service account credentials. The corresponding Application ID must have been specified in the Excel Services configuration settings. Excel Services retrieves the credentials from the Secure Store Service in the context of its own service account and then uses the Windows *LogonUser* function to authenticate the user.

**Note** Credentials in the connection string can override the Excel Services authentication settings to establish the connection to the data source. For example, a SQL Server data source can use SQL Server authentication in which case Excel Services first applies its own authentication settings and then establishes the connection at which point the SQL Native Client uses the user name and password from the connection string to log on to the data source. The Excel Services authentication settings are effective if the data source uses Windows authentication and the connection string specifies integrated security (Integrated Security=SSPI).

#### Centralizing Authentication Settings in a Data Connection Library

Excel provides the ability to export data connection definitions from a workbook into Office Data Connection (ODC) files by clicking on Export Connection File in the Connection Properties dialog box, which can be displayed in Excel by clicking on Connections on the Data tab, then selecting the desired connection, clicking Properties, and then switching to the Definition tab. The ODC file can be stored locally or in a data connection library in SharePoint.

An ODC file is an XML document that stores among other things the connection information to access a data source, including authentication settings. The data connection library, on the other hand, is a specialized type of document library that enables administrators to maintain a central repository of data connections in SharePoint. For example, administrators can publish preconfigured data connections with stored credentials. Users can then simply consume these data connections by clicking on Existing Connections on the Data tab and then browsing for and selecting the desired ODC file. If the data connection parameters must be changed later on (for example, if a different Secure Store target application must be specified), it is only necessary to update the corresponding ODC file. Clients automatically pick up the changes.

It is noteworthy that ODC files created in Excel 2010 or Excel 2013 can also be used with Visio and Visio Services. Visio Services respects the authentication settings for data-connected diagrams published in SharePoint 2013. For more information, see the Microsoft TechNet article “Data authentication for Visio Services in SharePoint Server 2013” at <http://technet.microsoft.com/en-us/library/ff829835(office.15).aspx>.

Also note that the SharePoint authentication settings only apply to Excel Services and Visio Services. They do not apply to Excel 2013 or Visio 2013 because these rich clients work independently from SharePoint 2013. For data sources that use Windows authentication, these clients connect in the context of the currently logged-on user. They do not use Secure Store credentials or unattended service accounts.

**Note:** ODC files are clear-text files. Users with access to the data connection library can examine the connection details, such as the connection string to the data source.

#### PerformancePoint Services Authentication Settings

Unlike Excel workbooks and Visio diagrams, PerformancePoint dashboards use PerformancePoint Services Data Connection (.PPSDC) files to retrieve connection information for a data source. PPSDC files are XML files created in Dashboard Designer and stored in a PerformancePoint Data Connections library. Their use and purpose is similar to ODC files. In fact, a PerformancePoint Data Connections library supports the same content types as standard data connection libraries do so PPSDC files and ODC files can be maintained in the same location. However, PPSDC files and ODC files are not interchangeable. Excel 2013 and Visio 2013 cannot use PPSDC files and PerformancePoint dashboards cannot use ODC files.

Despite the file format differences, PerformancePoint Services authentications options are very similar to Excel Services and Visio Services authentication settings. Table 3 summarizes the authentication settings.

Table 3   PerformancePoint Services authentication settings

|  |  |  |
| --- | --- | --- |
| PerformancePoint Services Authentication Setting | Excel Services Authentication Setting | Comments |
| Unattended service account | None | All users connect to the specified data source by using the same credentials stored in the target application ID registered as the unattended service account in the PerformancePoint Services configuration. |
| Use a stored account | Use a stored account | All users connect by using the same credentials stored in the specified target application ID. |
| Per-user identity | Use the authenticated user’s account | PerformancePoint Services impersonates the authenticated SharePoint user by using C2WTS and KCD. |

#### Reporting Services and Power View Authentication Settings

SQL Server Reporting Services and Power View use yet another content type in SharePoint to maintain authentication settings for shared data sources in the form of Reporting Services data source (RSDS) files. There is no preconfigured document library, but the Report Data Source content type can be added to any SharePoint document library when Reporting Services Power View is installed and configured in the farm.

Reporting Services data sources support the following authentication settings:

* **Windows authentication (integrated) or SharePoint user** Uses the credentials of the user who opened the report. This option requires C2WTS and KCD and is only available for users who initially authenticated against SharePoint using Windows authentication. Also note that subscriptions and data alerts are not available with this authentication setting.
* **Prompt for credentials**   Prompts the user to enter credentials when running a report that uses this shared data source.
* **Select Stored credentials**   Instructs the report server to retrieve the user name and password from the RSDS file. Credentials are stored in encrypted format. There is no dependency on Secure Store Service.
* **Credentials are not required**   Runs the report by using credentials stored in the connection string or by using a least-privilege account that is configured on the report server. If this account is not configured on the report server, users will be prompted for credentials.

In addition, RSDS files define how Reporting Services uses stored or supplied credentials:

* **Use as Windows credentials**   Triggers a Windows *LogonUser* call using the supplied credentials, which requires the supplied credentials to belong to a Windows user account. This option must be selected if the data source is using Windows authentication.
* **Set execution context to this account**   Causes the report server to set the execution context on the backend server. For SQL Server databases, this option relies on the SETUSER function, provided the Reporting Services service account is a member of the *sysadmin* fixed server role or the *db\_owner* fixed database role. For Analysis Services, this option relies on the *EffectiveUserName* command, provided the Reporting Services service account is an administrator of the data model in Analysis Services.

Note also that Reporting Services and Power View can use custom data sources embedded in a report as an alternative to a shared data source centrally maintained in a document library. For example, a user can create a Power View report based on a workbook in a PowerPivot Gallery without having to use an RSDS file. The report uses a custom data source of type *Microsoft BI Semantic Model for Power View* with the URL of the underlying PowerPivot workbook in the connection string and the authentication setting *Windows authentication (integrated) or SharePoint user*, which can be verified by opening the SharePoint context menu of the report in the PowerPivot Gallery in All Documents View, then clicking on the Manage Data Sources option, and then clicking on the name of the custom data source, such as *EntityDataSource*. Custom data sources support the same authentication settings as shared data sources.

### Kerberos Constrained Delegation and Alternatives

SharePoint shared services that want to delegate the identity of a SharePoint user to an external data source that uses Windows authentication (integrated security) must retranslate the user’s SharePoint security token back into a Windows identity. The Claims to Windows Token Service (C2WTS) facilitates this process. This is a Windows service, installed as part of Windows Identity Foundation (WIF). By default, C2WTS uses the Local System account. If the service account is changed to a domain account, the account must be granted special permissions, such as the right to Act as Part of the Operating System. For details, refer to KB article 2722087 “*How to configure Claim to Windows Token Services in SharePoint 2010 with Kerberos Authentication*” at <http://support.microsoft.com/kb/2722087>.

#### Identity Flow based on Kerberos Constrained Delegation

Figure 9 illustrates the identity flow across SharePoint to a Windows backend service. The Web app on the SharePoint WFE must use either classic-mode or claims-mode Windows authentication. The user information is then delegated to the SharePoint shared service through WFC and *IssuedTokenOverTransport*, as explained earlier in this paper. Next, the shared service locally invokes C2WTS through WIF in order to translate the SharePoint security token back into a Windows token, provided the inbound authentication method was Windows. C2WTS extracts the User Principal Name (UPN) from the SharePoint security token and contacts an Active Directory domain controller to obtain an impersonation-level Windows token by using Service for User (S4U) Kerberos Extensions. Provided Kerberos Constrained Delegation (KCD) was properly configured between the C2WTS service and the external data source, the SharePoint shared service can then delegate the user’s Windows identity to the data source.

**Note**   KCD is only required for accessing remote data sources. Data sources hosted locally on a SharePoint application server are accessible without KCD.

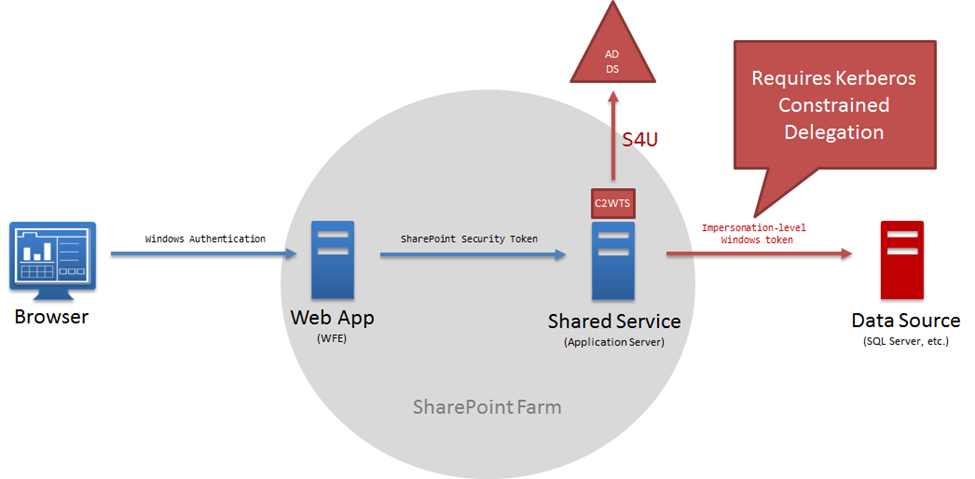


Figure 9  Delegating a Windows identity to a backend system

#### Kerberos Constrained Delegation Issues

There are a number of issues with this identity flow. One issue is that SharePoint cannot delegate the identity of a non-Windows user to the external data source because shared services do not invoke the C2WTS service for forms-based or SAML-based users even if their tokens include a UPN claim that has a matching user account in Active Directory. Only domain users authenticated by using classic-mode or claims-mode Windows authentication are supported.

Another issue is configuration complexity concerning KCD. Prior to Windows Server 2012, configuring KCD required a domain administrator to accomplish several steps in Active Directory for registering Service Principal Names (SPNs) and enabling domain accounts for delegation. KCD also required the SharePoint farm and the external data sources to reside in the same Windows domain. It was not possible to delegate identities across domain boundaries. Although Windows Server 2012 KCD mitigates these issues (as described in "KCD Improvements in Windows Server 2012" in this paper), not every Active Directory environment includes Windows Server 2012 domain controllers yet.

**Note:** While Kerberos unconstrained delegation can also cross domain and forest boundaries, the tokens created by C2WTS will only work with constrained delegation.

#### Identity Flow based on Stored Credentials

If the Windows user’s identity cannot be delegated all the way to the data source, stored credentials or an unattended service account must be used to establish the connection. This implies that the user identity flow effectively stops at the SharePoint application server.

Stored credentials may be an acceptable workaround in situations where C2WTS or KCD cannot be used. C2WTS or KCD are not required because the application server can directly authenticate against the data source, but it is important to point out that the user identities might now be hidden from the data source, which is an issue if the data source is supposed to return different result sets depending on the user. Particularly in large environments, it is often impossible to maintain stored credentials for every user individually.

#### Identity Flow based on EffectiveUsername

Analysis Services offers yet another way to work around KCD limitations by means of a special connection string parameter called *EffectiveUsername*, which members of the server administrator role on the Analysis Services instance can use to act on behalf of another user. Figure 10 illustrates the use of *EffectiveUsername* in a PowerShell script that queries Analysis Services for the name of the current user.

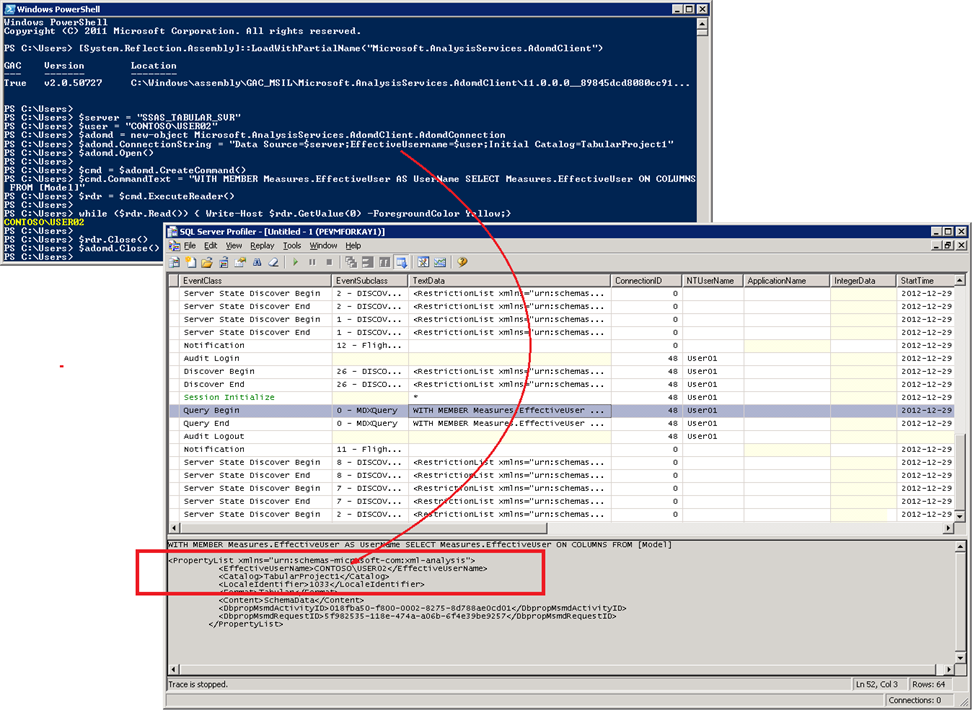


Figure 10   Testing the EffectiveUsername parameter

As the PowerShell script in Figure 10 demonstrates, any client can use the *EffectiveUsername* property to pass the name of a Windows domain account to Analysis Services as long as the user making the connection has administrator permissions on the data model. Only Analysis Services administrators can make a connection using the *EffectiveUsername* parameter. Note also that only domain accounts can be specified in the *EffectiveUsername* parameter because Analysis Services uses S4U Kerberos Extensions similar to the C2WTS service to impersonate the user. S4U Kerberos Extensions require Active Directory. If a local Windows account is specified, Analysis Services returns an error to the client opening the connection stating that there are currently no logon servers available.

Although *EffectiveUsername* relies on S4U Kerberos Extensions, KCD is not required unless Analysis Services must access remote resources, such as a SQL Server database when operating in DirectQuery mode. Hence, for data models that do not use DirectQuery mode, *EffectiveUsername* can provide an alternative to KCD for user authentication against Analysis Services in the backend. Excel Services, PerformancePoint Services, and Reporting Services Power View can use *EffectiveUsername* if the corresponding service accounts are granted server administrator permissions in Analysis Services. Power View uses *EffectiveUsername* if the option *Set execution context to this account* is enabled in the data source definition. If Kerberos fails, Reporting Services initiates a second connection attempt with its own service identity and NTLM authentication, passing the Power View user identity on the connection string. Excel Services and PerformancePoint Services, on the other hand, must be configured explicitly. For details, refer to the TechNet article “*Use Analysis Services EffectiveUserName in SharePoint Server 2013*” at <http://technet.microsoft.com/en-us/library/jj219741.aspx>.

**Note**   SharePoint shared services can only use *EffectiveUserName* if the user was authenticated using classic-mode or claims-mode Windows authentication. This feature is not available to delegate the identity of a non-Windows user to Analysis Services.

#### KCD Improvements in Windows Server 2012

Windows Server 2012 enables cross-domain and cross-forest delegation scenarios that previous Windows Server versions did not support. The new version also eliminates KCD configuration complexities by moving the authorization decision to the resource owners. KCD configuration no longer requires Domain Admin privileges. The back-end systems now authorize which front-end service accounts can impersonate users against their resources. To specify which front-end service accounts can impersonate users, a resource owner with administrative permissions to the back-end service account can use the following PowerShell cmdlets:

* New/Set-ADComputer [-name] <string> [-PrincipalsAllowedToDelegateToAccount <ADPrincipal[]>]
* New/Set-ADServiceAccount [-name] <string> [-PrincipalsAllowedToDelegateToAccount <ADPrincipal[]>]

To take advantage of the new KCD capabilities, the BI environment must meet the following requirements:

* One or more domain controllers in the front-end and backend domains must be running Windows Server 2012
* The SharePoint servers must be running Windows Server 2012
* The back-end server must be running Windows Server 2003 or later
* The back-end server account must be configured with the accounts that are permitted for impersonation

For more information about KCD Improvements in Windows Server 2012, refer to the Microsoft TechNet article “What's New in Active Directory Domain Services (AD DS)” at <http://technet.microsoft.com/en-us/library/hh831477.aspx>.

### Accessing Analysis Services in SharePoint Mode

Analysis Services in SharePoint mode is a special kind of backend system, specifically designed for SharePoint 2013 connectivity without KCD requirements. Unlike other external data sources, including multidimensional and tabular Analysis Services data models, Analysis Services in SharePoint mode relies on a trusted subsystem design in which the Analysis Services servers in the backend fully trust the SharePoint Web applications and shared services to act on behalf of the users. To establish this trusted subsystem, the SharePoint farm account, Web applications, and shared services must have server administrator permissions in Analysis Services. Figure 11 shows a configuration for separate Excel Services, PerformancePoint Services, and Power View service accounts.

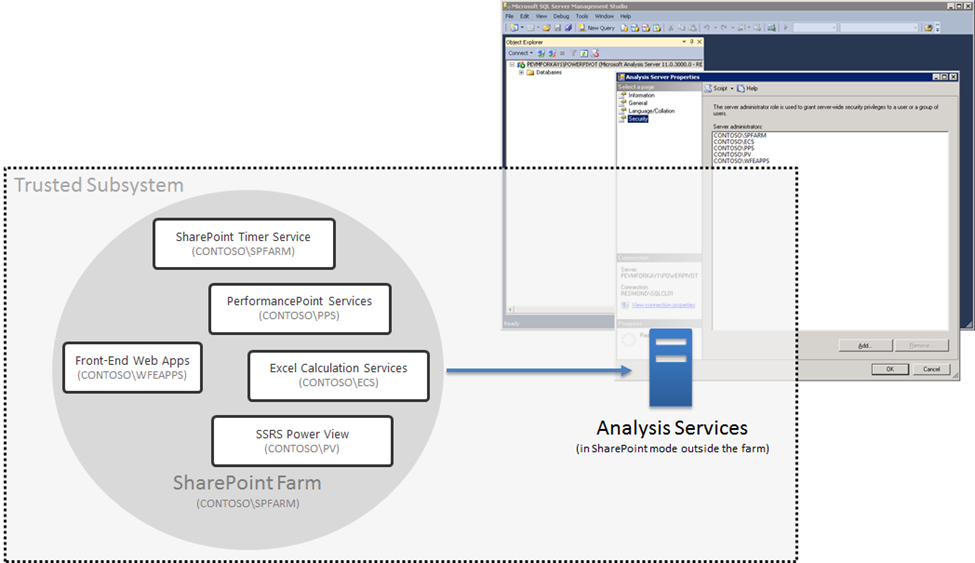


Figure 11   *Granting SharePoint system accounts Analysis Services admin permissions to establish a trusted subsystem*

#### Connecting to a Data Model in SharePoint Mode

Earlier in this paper, Figure 5 explained how the Workbooks as a Data Source architecture enables SharePoint shared services to load and access PowerPivot data models. As explained, ECS first loads the model and then the requesting shared service establishes a direct TCP/IP connection. For Analysis Services in SharePoint mode to accept the connection request, the service must have server administrator permissions. Analysis Services in SharePoint mode does not accept user connections.

Having established a direct connection, SharePoint can use the *SetAuthContext* command to change the user name on the connection. Again, only Analysis Services server administrators can use this command, it is only available in SharePoint mode, and it must be the first command sent after the connection is established. The user name is an arbitrary string. It does not need to correspond to a user name in Active Directory. Because the *SetAuthContext* command has no impersonation or delegation requirements, it works for any type of SharePoint user identity: Windows-, forms-, and SAML-based.

Figure 12 shows how SharePoint shared services use the *SetAuthContext* command. In this example, a user who was authenticated in SharePoint by using SAML passive sign-in is interacting with the data model through Excel Services. The authentication provider in SharePoint was configured to use the user’s email address to identify the user. Accordingly, Excel Services sends the email address as the user name to Analysis Services. The characters in front of the email address in the *SetAuthContext* token indicate that this is a claims-based user.

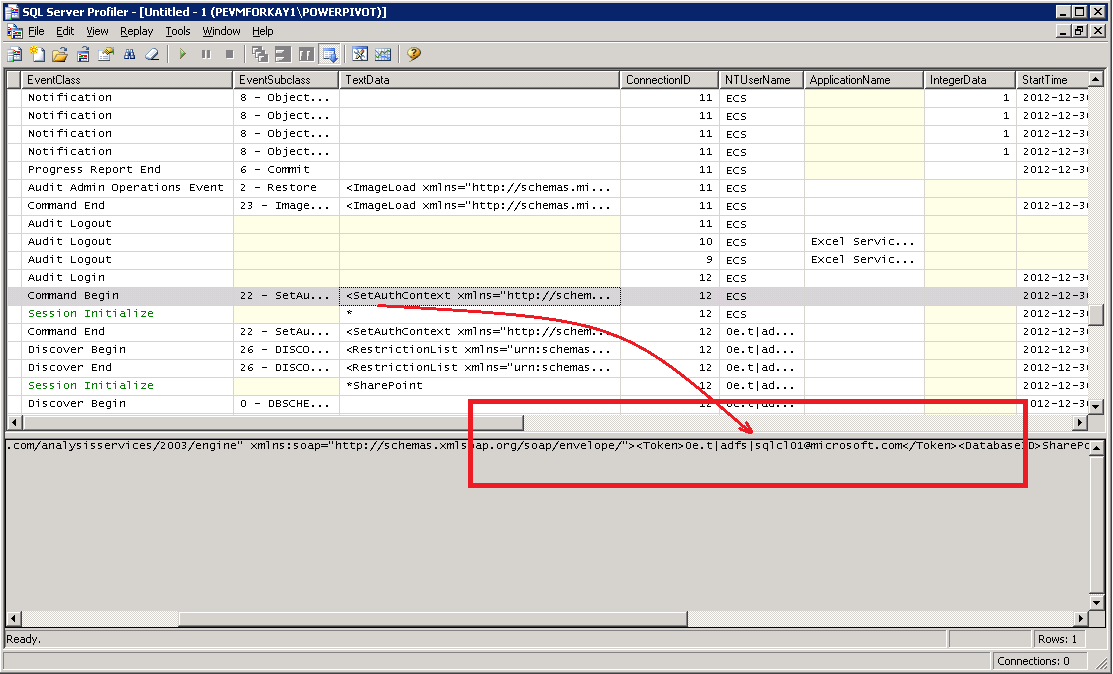


Figure 12   The *SetAuthComtext* command in a SQL Profiler Trace

**Note**   The *SetAuthContext* command does not impersonate or authenticate a user in Analysis Services for the purposes of granting or denying access to a data model. PowerPivot data models do not provide security features. Access controls are applied to the workbooks in SharePoint. Excel Services loads the data model if the user has access to the workbook.

#### Processing PowerPivot Data Models

Because the *SetAuthContext* command does not actually impersonate or sign in a user, it cannot be used as a workaround to delegate a user identity to Analysis Services for the purposes of data refresh, such as when a user clicks on Refresh Selected Connection or Refresh All Connections on the Data menu of a PowerPivot workbook in the browser. Instead, Excel Services conveys the impersonation information in an out-of-line binding within the processing command that it sends to Analysis Services in SharePoint mode, as shown in Figure 13.



Figure 13   Impersonating a user to refresh a PowerPivot data model

The impersonation information within the processing command depends on the Excel Services authentication settings defined for the data connection, as follows:

* **Use the authenticated user’s account**   Excel Services impersonates the authenticated user by using C2WTS and then sends the Account name of the Windows identity to Analysis Services. The password is not retrievable for the currently logged-on user.
* **Use a stored account or the unattended service account**    Excel Services retrieves the user name and password from the specified Target Application ID in Secure Store Service and then includes both Account name and Password in the ImpersonationInfo.

The difference between these authentication settings is significant because the authenticated user’s account option only works for interactive Windows users due to its C2WTS and KCD dependencies, as Figure 14 illustrates. Stored accounts, on the other hand, work for all types of SharePoint users because, given account name and password, Analysis Services can use the standard Windows *LogonUser* function to authenticate the user. The stored credentials refer to a Windows account that Excel Services can use instead of an interactive non-Windows user identity.

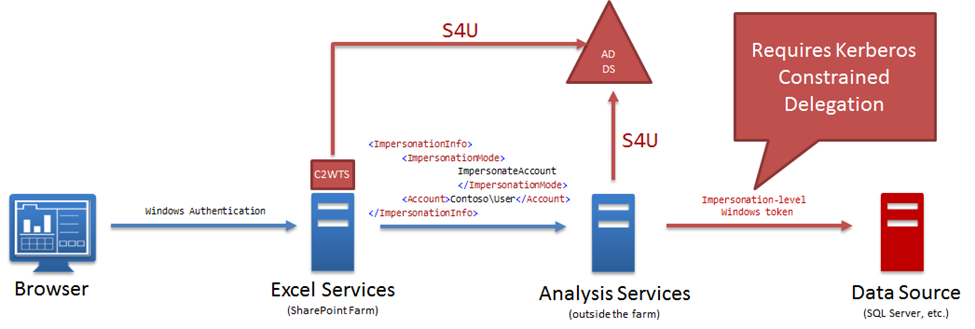


Figure 14   Impersonating a Windows user for data refresh

If Excel Services does not submit a password along with the account name in the ImpersonationInfo, Analysis Services must use S4U Kerberos Extensions to impersonate the user. This requires the right to Act as Part of the Operating System, which Analysis Services does not have by default because it is a very high-privileged permission practically equivalent to running as Local System. So, even with Windows authentication in SharePoint and the C2WTS running on application servers, interactive data refresh won’t succeed until the Analysis Services service account is granted the right to Act as Part of the Operating System. There is also the issue that S4U returns an impersonation-level Windows token, which requires KCD for Analysis Servcies to access remote data sources.

If Analysis Services cannot be granted the right to Act as Part of the Operating System for any reason or KCD cannot be configured, users must switch the Excel Services authentication settings in their workbooks from Windows authentication to *Use a stored account* or *None*. Alternatively, it is possible to configure scheduled data refresh and specific the credentials in the refresh settings. Scheduled Data Refresh always works with stored credentials since there is no interactive user.

# Corporate BI Data Access Scenarios

Corporate BI describes a set of tools that help business users align their objectives and activities with overall company goals, objectives, and metrics, such as for organizational planning, budgeting, forecasting, and financial consolidation. The main distinguishing factor is that corporate BI solutions are more formally developed and centrally maintained by BI professionals in the IT department whereas personal and team BI solutions are primarily the domain of information workers and power users in business units. Typical corporate BI solutions include:

* **SharePoint-based dashboards, scorecards, strategy maps, reports, and other tools**  Based on PerformancePoint Services and Reporting Services in SharePoint-integrated mode, but also PowerPivot and Power View. Because these solutions take advantage of a SharePoint infrastructure, the user authentication and identity delegation requirements are the same as for team BI solutions in SharePoint discussed in the previous sections.
* **Reporting Services reports**   Managed by using SQL Server Management Studio and Report Manager on a report server running in native mode. Reporting Services is designed and tested with Windows Authentication, but it is also possible to build custom authentication extensions to broaden the audience of corporate reporting solutions.
* **Analysis Services data models**   Hosted on a server running Analysis Services in multidimensional or tabular mode. The Analysis Services engine only supports Windows Authentication, but it is possible to deploy custom HTTP handlers or modules to extend the authentication capabilities when using the HTTP data pump.

## Authentication Flows in Native-Mode Reporting Services

Reporting Services in native mode supports the same authentication settings for custom and shared data sources as Reporting Services in SharePoint-integrated mode. The report server can use Windows authentication (integrated) to flow the identity of the user who opened the report all the way to the data source (if KCD is configured), prompt the user for credentials, use stored credentials, or connect to the external data source without requiring explicit credentials. Figure 15 illustrates the corresponding authentication flows.

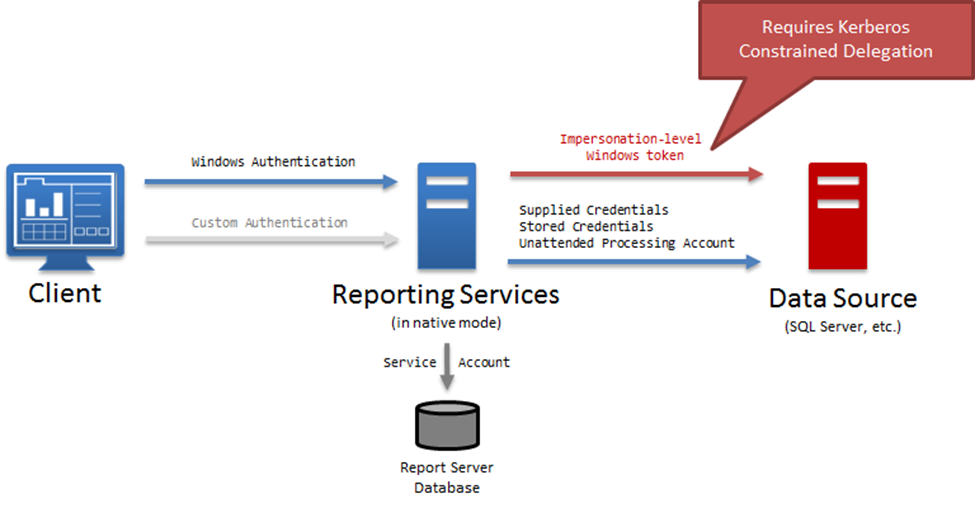


Figure 15   Native-mode SSRS authentication flows

### Windows User Authentication

On the front-end side, Reporting Services supports Windows authentication (integrated) with delegated or impersonated credentials, but Basic authentication and forms-based authentication can also be enabled by configuring the *AuthenticationType* parameter in the *RSeportServer.config* file located in the *%ProgramFiles%Microsoft SQL Server\<instance name>\ReportServer* folder. For detailed information, refer to the Microsoft TechNet article “Authentication with the Report Server” at <http://msdn.microsoft.com/en-US/library/bb283249(v=sql.110).aspx>.

**Note**   ASP.NET must be configured for Windows Authentication. By default, the Web.config files for the Report Server Web service and Report Manager include the <authentication mode="Windows"> setting. If this setting is changed to <authentication mode="Forms">, the Windows Authentication for Reporting Services will fail.

### Custom User Authentication

The default Windows authentication extension requires each user accessing a report server to have a valid Windows user account. The account must have access to the Web server hosting the report server, and must be assigned to Reporting Services roles to gain access to specific report server operations. For users who do not have a Windows account, a custom extension can provide the required authentication logic. One common scenario for using custom authentication is to support extranet access. Another is to support internal users that do not have user accounts in Active Directory. For details, read the Microsoft TechNet article “*Configure Custom or Forms Authentication on the Report Server*” at <http://msdn.microsoft.com/en-us/library/cc281383(v=sql.110).aspx>.

### Custom Data Source Authentication

With custom user authentication, Reporting Services can no longer delegate user identities to an external data source. Stored credentials must be used. It is likewise impossible to set the execution context to a user account because the backend system cannot authenticate a non-Windows account. For example, sending the name of a non-Windows user to Analysis Services by using the *EffectiveUserName* command only results in a connection error, as explained earlier in this paper. Reporting Services supports data processing extensions, which can be used to customize how to connect to an external data source and retrieve data, but the usefulness is limited if the data source requires a Windows identity, such as Analysis Services. For details about the SSRS data processing API, refer to the Microsoft TechNet article “*Data Processing Extensions Overview*” at <http://msdn.microsoft.com/en-us/library/ms152816.aspx>.

## Authentication Flows in Analysis Services

Unlike Reporting Services, Analysis Services in multidimensional or tabular mode typically does not delegate user identities to the backend data sources. A noteworthy exception is DirectQuery tabular mode.

### Multidimensional and Tabular Data Access

A multidimensional or tabular data model is an analytical database derived from the tables and views of one or multiple underlying data sources. Similar to PowerPivot in Excel, Analysis Services can connect to relational databases, multi-dimensional or tabular data models, data feeds, SharePoint lists, and data files. Yet unlike PowerPivot workbooks, Analysis Services data models do not connect to data sources that use Windows authentication in the context of the currently logged on user. Instead, Analysis Services data models use processing accounts to import the full set of data and then implement their own security layer to provide each user with specific access to the analytics data, as illustrated in Figure 16.

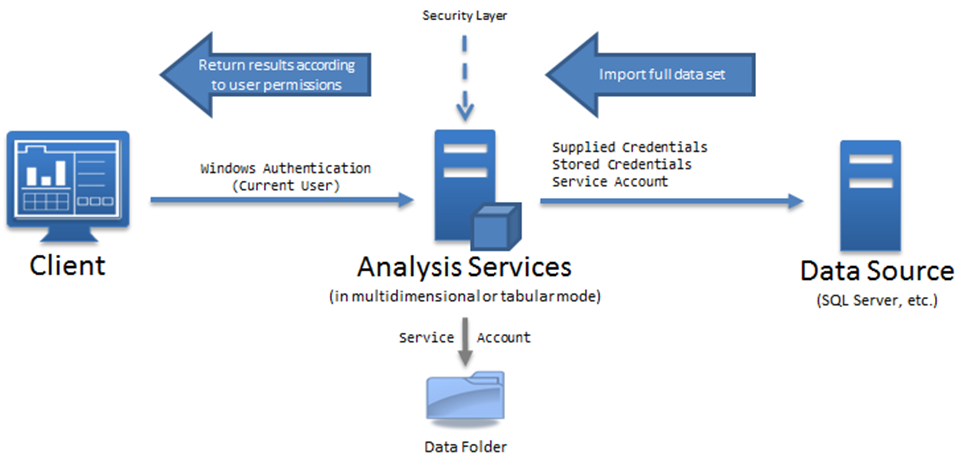


Figure 16   Standard authentication flows in Analysis Services

Although the logic for loading data into a multidimensional model is more complex than the logic for loading data into a tabular data model, the underlying data source definitions are basically the same. For data sources that support Windows authentication, it is possible to define impersonation settings. For multidimensional models, switch to the Impersonation Information tab in Data Source Designer. For tabular modes, use the Impersonation Information page in the Table Import Wizard or click on the Impersonation button in the Edit Connection dialog box. The following impersonation settings are typically used:

* **Specific Windows user name and password**   Analysis Services uses a dedicated, least-privilege Windows user identity created specifically for data access purposes.
* **Service account**   Analysis Services uses the security credentials associated with the Analysis Services instance that manages the data model.
* **Out-of-line bindings**   Analysis Services uses the account name and password supplied as part of an ImpersonationInfo node in an out-of-line binding, as discussed in the context of processing PowerPivot data models earlier in this paper.

For details about Analysis Services impersonation settings for data sources, refer to the Microsoft TechNet topics “*Set Impersonation Options (SSAS - Multidimensional)*” at <http://msdn.microsoft.com/en-us/library/ms187597.aspx> and “*Impersonation (SSAS Tabular)*” at <http://technet.microsoft.com/en-us/library/gg492180.aspx>.

### ROLAP and DirectQuery Mode

By default, Analysis Services imports data from the data sources into the data model to provide best query performance. This implies, however, that changes at the data source level are not available for analysis until the data model is reprocessed again to import the updates. The source data volume might also exceed the Analysis Services capacity, making it difficult to complete import operations successfully. In order to accommodate these situations, Analysis Services data models can access data and aggregates directly from a relational database.

Multidimensional data models operating in Relational Online Analytical Processing (ROLAP) store mode generate SQL statements to fetch data from a relational database as needed to satisfy query requests. The SQL connection uses the Analysis Services service account or stored credentials. The current user’s credentials cannot be used because Analysis Services may fetch more data from the data source than the user is allowed to access and may cache this data to satisfy queries from other users as well. Regardless of the multidimensional storage mode, the security settings must be defined in the Analysis Services data model.

Tabular data models, on the other hand, can use DirectQuery mode, which can delegate credentials of the current user to connect to the data source as an alternative to stored credentials or the Analysis Services service account. Note that KCD must be configured between Analysis Services and remote data sources to delegate credentials successfully, as Figure 17 illustrates. Note also that Analysis Services can only delegate user credentials for direct queries against the relational data source. This impersonation setting is invalid for InMemory, DirectQuerywithInMemory, and InMemorywithDirectQuery query modes. For details about DirectQuery mode capabilities and limitations, refer to the Microsoft TechNet topic “*DirectQuery Mode (SSAS Tabular)*” at <http://technet.microsoft.com/en-us/library/hh230898.aspx>.

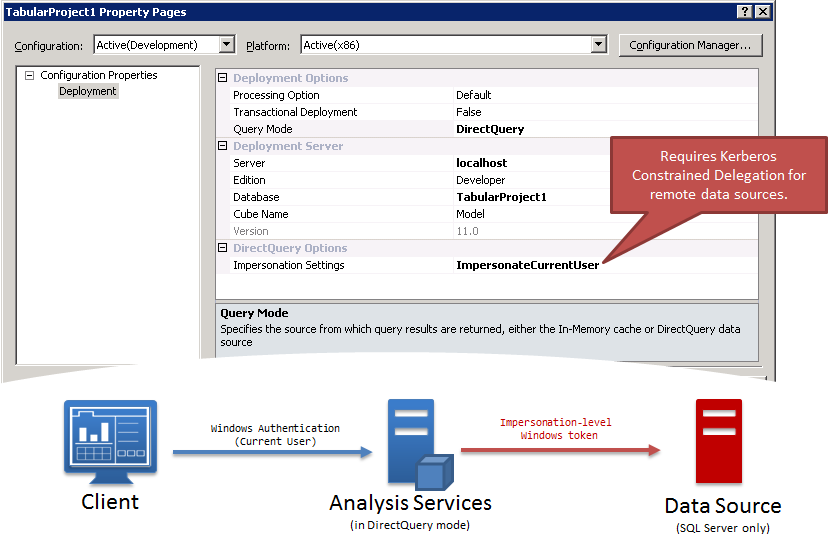


Figure 17   Identity delegation in DirectQuery mode

### Client Authentication over HTTP

On the front-end side, Analysis Services requires Windows authentication (integrated). This might be an issue for clients in non-trusted domains or network locations without direct TCP/IP connectivity to the ports of the Analysis Services instance hosting the data model. In these situations, it can be useful to deploy the Analysis Services data pump (MSMDPUMP.dll), which is an IIS ISAPI extension to provide HTTP(S) connectivity. Because MSMDPUMP is hosted in IIS, it can support alternative authentication methods, such as Anonymous and Basic authentication. For details about how to configure MSMDPUMP, refer to the Microsoft TechNet topic “*Configure HTTP Access to Analysis Services on Internet Information Services (IIS) 7.0*” at <http://msdn.microsoft.com/en-us/library/gg492140(v=sql.110).aspx>.

Figure 18 illustrates the data pump architecture for HTTP-based access to Analysis Services. The client can use any of the Analysis Services data providers to connect. The supplied credentials must match the authentication mode enabled on the IIS server. For NTLM, Kerberos, and Anonymous authentication the option Use Windows Authentication should be used in the client, which corresponds to Integrated Security = SSPI on the connection string. If Basic authentication is enabled in IIS, the user name and password must be specified explicitly on the connection string through the *User Id* and *Password* parameters.

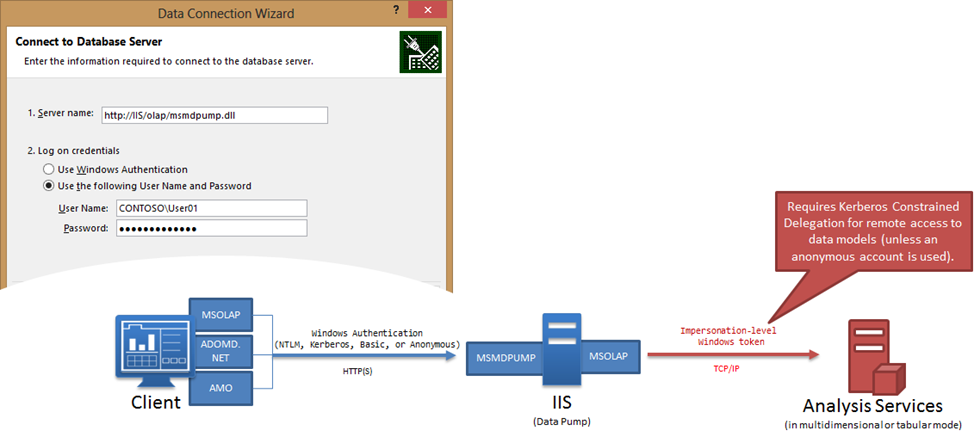


Figure 18   MSMDPUMP architecture for HTTP-based access to Analysis Services

The data pump on the IIS server, in turn, uses MSOLAP to connect to the Analysis Services data model. Depending on the authentication method enabled in IIS, MSMDPUMP can authenticate against Analysis Services by either delegating the user credentials or by using the user account specified in the Anonymous Authentication module in IIS Manager. If Analysis Services is running on a remote computer and the client connection uses any authentication method other than Anonymous, KCD must be configured for IIS to delegate user credentials successfully. Anonymous authentication does not require KCD because IIS can perform a direct logon with the configured anonymous user account.

**Note**   IIS supports Forms and Digest authentication in addition to NTLM, Kerberos, Basic, and Anonymous. However, the Analysis Services data providers do not support these authentication methods. Thin XML/A clients that communicate directly with Analysis Services over HTTP(S) might be able to use these additional authentication methods, but Excel, Power View, SQL Server Management Studio, and other tools that use MSOLAP, ADOMD.NET, or AMO do not.

### Custom Analysis Services Authentication over HTTP

IIS provides rich extensibility features, including support for custom ASP.NET HTTP handlers and modules, which can be used to provide additional authentication modules and other security checks in front of the Analysis Services data pump. A good example is the SSASProxy Custom HTTP Security Sample solution, available for download at <http://code.msdn.microsoft.com/office/SSASProxy-Custom-HTTP-7db6d016>.

The SSAS Proxy HTTP handler must be deployed in an IIS application that supports HTTP Anonymous authentication. It uses Basic authentication to validate user credentials against a custom identity store and then modifies the client’s XML/A BeginSession request to add the authenticated user identity in the CustomData field before passing the request on to the data pump. MSMDPUMP in turn establishes the connection to Analysis Services by using the account configured in the Anonymous authentication module of IIS. Having received the custom data, Analysis Services can access the user identity information through the CustomData() function. This provides a foundation for custom security controls in a data model. Figure 19 illustrates the SSAS Proxy architecture.

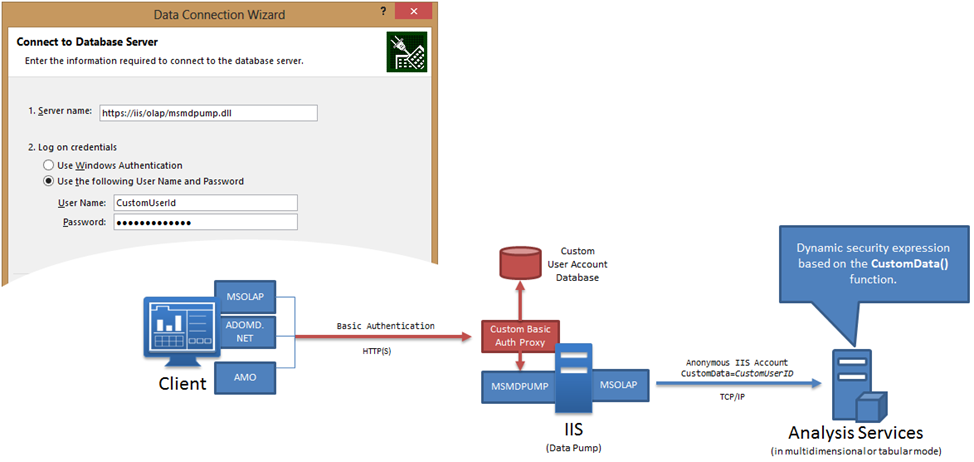


Figure 19   A custom security solution for Analysis Services

# Federated BI Scenarios

A federated environment consists of multiple autonomous realms, domains, or forests that establish security boundaries and trust relationships around user accounts and resources so that users can seamlessly access resources across security boundaries without having to use multiple separate sets of credentials.

* **Multi-forest environments**   Enterprise environments usually consist of multiple Active Directory forests to meet strict security requirements of business units, accommodate different directory schema needs, or because of mergers and acquisitions. Typical deployments are based on a resource-forest or cross-forest topology. Although a multi-forest Active Directory environment supports Windows authentication between clients and servers, it is not always possible to delegate user identities to backend systems in all locations.
* **Extranet environments**   Extranets provide internal and external users with access to resources in a perimeter network. One-way forest trusts between Extranet and internal forests provide internal Windows users with access to extranet resources. External users, such as vendors and partners, can get seamless access if federation trusts are established for Web single sign-on (SSO). Note however that not all Microsoft BI technologies support Web SSO.
* **Cloud environments**   Cloud environments are similar to extranets in that users can seamlessly access resources if federation trusts have been established for Web SSO. The main difference is that there are no Active Directory forest trusts between the cloud and the corporate network so internal users can no longer use Windows authentication. It is also difficult for cloud systems to access data sources on-premises.

## Multi-Forest Active Directory Environments

In multi-forest Active Directory environments, trust relationships define how users in one realm, domain, or forest can access resources in other realms, domains, or forests. Moreover, trust relationships impact how the systems can delegate user identities. While Windows Server 2012 extends the reach of KCD across domain and forest boundaries, limitations still exist and might require the use of stored credentials in specific situations, as illustrated in Figure 20.

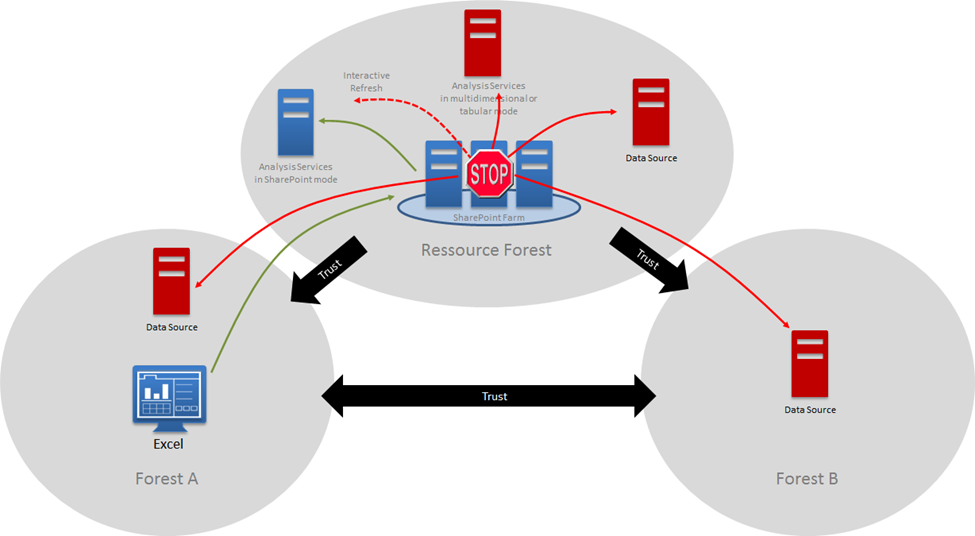


Figure 20   Delegation limitations in multi-forest environments based on Windows Server 2012

In the depicted scenario, a user in forest A has access to resources in all other forests, yet the resource forest hosting the SharePoint farm only has one-way trusts defined. The trust relationships enable users to access sites and services in the farm, but they do not support S4U Kerberos Extensions within the resource forest to obtain impersonation tokens for user accounts from forest A or B. The SharePoint servers cannot contact domain controllers in other forests. As a consequence, SharePoint in the resource forest cannot use C2WTS to translate its security tokens into Windows identities. Because practically all Microsoft BI-related SharePoint services rely on C2WTS, Windows identities cannot be delegated until trusts are also established in the opposite direction so that the resource-forest services can communicate with the domain controllers in the account forests.

Note that the S4U dependencies also affect Analysis Services in multidimensional and tabular mode when using the *EffectiveUsername* connection string parameter because *EffectiveUsername* requires Analysis Services to impersonate the user. On the other hand, Analysis Services in SharePoint mode is less affected due to the trusted subsystem design. The trusted subsystem design enables SharePoint shared services to communicate user names to Analysis Services in SharePoint mode without requiring impersonation, as explained earlier in this paper. The only exception is an interactive refresh of a workbook’s data connections in the browser using the authenticated user’s identity because Excel Services cannot determine the authenticated user’s identity by using C2WTS.

Note also that these explanations assume domain controllers and SharePoint servers running Windows Server 2012. KCD in previous versions of Windows Server is limited to a single domain. In these environments, even with two-way trust relationships to support S4U Kerberos Extensions, user identities cannot be delegated to external data sources unless the data sources are moved or replicated to the local domain of the SharePoint servers.

## Extranet Environments

Extranets are environments with limited or no access to the corporate network. For security reasons, it is usually not acceptable to establish two-way trusts. In a recommended configuration, illustrated in Figure 21, there is a one-way Active Directory trust relationship from the extranet to the corporate network so that internal users can access extranet resources with their Windows credentials. In this configuration, the same S4U Kerberos limitations apply as in resource forests with one-way trust relationships to account domains. Running on extranet servers, C2WTS and Analysis Services cannot impersonate internal users through S4U, making it necessary to use stored credentials. In configurations without any Active Directory trusts, on the other hand, corporate users must authenticate the same way as external users.

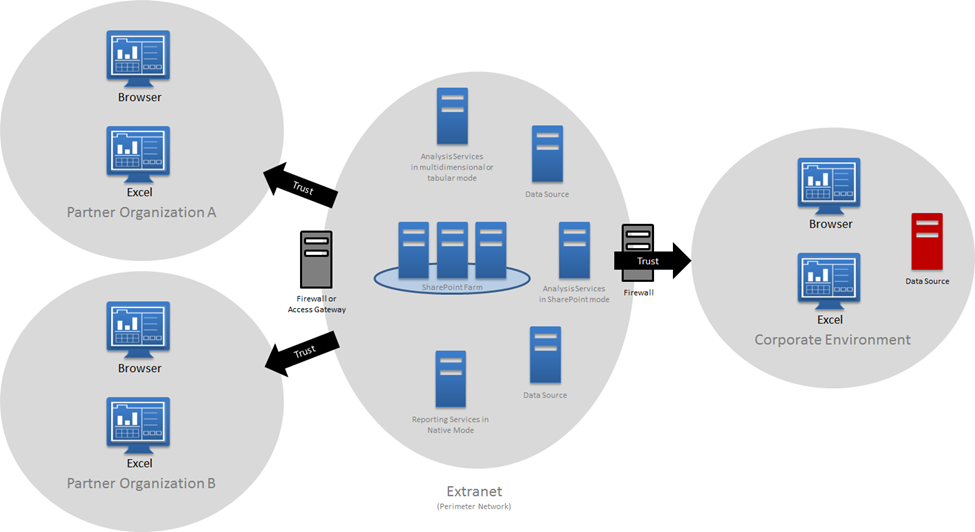


Figure 21   Microsoft BI solutions in an extranet environment

For external users and partner organizations, an enterprise organization might maintain separate extranet accounts, establish federation trusts, or provide virtual private network (VPN) or DirectAccess connectivity to support Kerberos authentication. Extranet accounts can be maintained in Active Directory or in separate account stores. Web-based security standards, such as WS-\* and OAuth, provide the basis to broker federated trust relationships between organizations and enable SSO without requiring Active Directory and Kerberos. VPNs and DirectAccess are frequently used to provide mobile corporate users with connectivity. VPNs and DirectAccess essentially project the external users into the corporate environment through encrypted communication channels.

In extranet environments, SharePoint farms frequently rely on forms-based authentication or SAML token sign-in. The account store can be a relational database, a third-party LDAP directory (such as Sun Directory or IBM TAM/WebSEAL), or a system that supports SAML 2.0 (such as ADFS, PING Identity, or CA Federation Manager/SiteMinder). It is also possible to integrate social identity providers with SharePoint, such as Microsoft ID, Gmail, Facebook, and Twitter. However, external users authenticating against SharePoint by using these non-Windows authentication methods cannot take advantage of C2WTS or *EffectiveUsername* capabilities to convey their identities to backend systems, as discussed in detail earlier in this document. Stored credentials and unattended service accounts are the only available options.

It is also important to point out that Reporting Services in native mode and Analysis Services in multidimensional or tabular mode only support Windows authentication out of the box. Reporting Services can accommodate additional methods through custom authentication modules, yet Analysis Services is not as flexible because the data providers on the client side only support Windows authentication (NTLM, Kerberos, Basic, or Anonymous). Even if the Analysis Services data pump could support forms-based authentication or SAML token sign-in, MSOLAP, ADOMD.NET, and AMO do not use these methods on the client side.

Microsoft Forefront Unified Access Gateway 2010 (UAG) can help to address these issues by authenticating external users through Basic, forms-based, SAML-based methods and then mapping the users to shadow accounts in the extranet Active Directory forest. Having translated external identities into Windows shadow accounts, KCD can be used to delegate the identities from the access gateway to the application servers and backend systems. S4U Kerberos Extensions also work against the shadow accounts, yet the overhead associated with maintaining shadow accounts might be considerable. For details, refer to the Forefront UAG TechCenter at <http://technet.microsoft.com/en-us/forefront/ee907407>.

## Cloud Environments

Cloud environments, such as Microsoft Office 365, support SSO through Active Directory Federation Services (AD FS) 2.0 and directory synchronization between on-premises Active Directory Services (AD DS) and Windows Azure Active Directory (Windows Azure AD). Users can then access SharePoint Online and other cloud services with their corporate credentials and do not have to maintain separate user names and passwords. For details, refer to the “Single sign-on roadmap” at <http://onlinehelp.microsoft.com/en-us/office365-enterprises/hh125004.aspx>.

# Conclusion

The Microsoft BI stack is based on Windows client and server technologies and provides an optimal user experience if all systems in the solution architecture use Windows authentication. Systems that do not support Windows authentication typically require stored credentials, such as a user name and password in a connection string, data connection file, or report definition. Maintaining stored credentials is burdensome, especially if passwords expire frequently, and they are potentially less secure than Windows credentials because the passwords are retrievable. On the other hand, retrievable passwords enable the Windows systems in the BI solution architecture to use the *LogonUser* function for authentication, which eliminates the need to enable Kerberos Constrained Delegation to address the so-called double-hop authentication scenario.

Prior to Windows Server 2012, Kerberos Constrained Delegation represented configuration challenge. It required Active Directory domain administrator permissions and was limited in scope to a single Windows domain. It was not possible to delegate a user identity to a system in another domain. Windows Server 2012 eliminates these issues by moving the delegation decision to the resource owners. Domain admin permissions are no longer required to configure KCD and it is possible to cross domain and forest boundaries. A Windows system can now delegate a user identity to another system in a neighboring forest. Data sources no longer need to be moved or replicated to different domains for identity delegation reasons.

While the Windows Server 2012 improvements help to broaden the reach of Microsoft BI solutions, they do not eliminate the need for two-way trust relationships to support S4U Kerberos Extensions. Claims to Windows Token Service and Analysis Services use S4U to obtain impersonation-level Windows tokens in specific situations. If S4U does not work, SharePoint cannot translate its internal security tokens into Windows identities and can therefore not delegate user identities to external data sources. If Analysis Services cannot use S4U, it cannot support the *EffectiveUsername* connection string parameter or data model processing requests that only specify a user name in the impersonation information. Stored credentials or unattended service accounts must be used for backend authentication in these cases.

Even with full S4U support, it is important to keep in mind that SharePoint does not translate security tokens into Windows identities if the incoming user authentication method is forms- or SAML-based. This is particularly an issue in environments that do not support Windows authentication at the front-end, such as extranets and the cloud. It is likewise difficult to publish Reporting Services and Analysis Services solutions to external users without requiring at least Basic authentication. With Windows authentication, external users are required to use a separate set of extranet credentials. This might be avoided if an access gateway is available to authenticate the user based on Web security standards, mapping the authenticated user to a shadow account to present it as a Windows identity. Forefront Unified Access Gateway provides this capability.

In the majority of cases, stored credentials and shadow accounts can provide acceptable workarounds to establish sufficient authentication flows in distributed BI solutions. The implementation is not always seamless and comes with administrative and maintenance overhead, especially if systems with different authentication methods and security protocols are involved. Data connection libraries in SharePoint can help to centralize the maintenance of stored credentials to access external data sources and Scheduled Data Refresh can provide an alternative to interactive data refreshes of PowerPivot data models, eliminating the need to delegate an authenticated user’s identity to the data sources. In other cases, it might not be acceptable to hide user identities behind stored credentials. In these cases, the network and Active Directory topologies might have to be redesigned to accommodate the Kerberos requirements for end-to-end user authentication and delegation.

**For more information:**

<http://www.microsoft.com/sqlserver/>: SQL Server Web site

<http://technet.microsoft.com/en-us/sqlserver/>: SQL Server TechCenter

<http://msdn.microsoft.com/en-us/sqlserver/>: SQL Server DevCenter