[MS-SPO]:

SharePoint Protocols Overview

This overview describes the SharePoint Protocols. This overview is intended for use in conjunction with Microsoft protocol technical specifications, publicly available standard specifications, network programming art, and systems concepts distributed through Microsoft Windows. This overview assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A protocol system document does not require the use of Microsoft programming tools or programming environments in order to implement the protocols in the system. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Preliminary Documentation. This particular Open Specifications document provides documentation for past and current releases and/or for the pre-release version of this technology. This document provides final documentation for past and current releases and preliminary documentation, as applicable and specifically noted in this document, for the pre-release version. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. Because this documentation might change between the pre-release version and the final version of this technology, there are risks in relying on this preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Abstract

This document describes the intended functionality of the SharePoint System and how the protocols in this system interact. It provides examples of some of the common user scenarios. It does not restate the processing rules and other details that are specific for each protocol. These details are described in the protocol specifications for each of the protocols and data structures that make up this system.

Revision Summary

Date	Revision History	Revision Class	Comments
9/12/2012	0.1	New	Released new document.
10/8/2012	1.0	Major	Significantly changed the technical content.
2/11/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.1	Minor	Clarified the meaning of the technical content.
2/10/2014	1.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.2	Minor	Clarified the meaning of the technical content.
7/31/2014	1.2	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	1.3	Minor	Clarified the meaning of the technical content.
2/26/2016	1.3	None	No changes to the meaning, language, or formatting of the technical content.
7/15/2016	1.3	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	1.3	None	No changes to the meaning, language, or formatting of the technical content.
7/24/2018	2.0	Major	Significantly changed the technical content.
10/1/2018	3.0	Major	Significantly changed the technical content.
3/19/2019	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/18/2019	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/20/2021	4.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

This document provides an informative overview of the protocols that are implemented by SharePoint Products and Technologies, for the purpose of communicating with client and server applications.

SharePoint Products and Technologies provides an extensible platform on which collaboration and other scenarios and features are built. Some examples include team-oriented collaboration **Web sites**, document and content publishing features, and the ability to create portals that publish data from systems external to the SharePoint Products and Technologies deployment, to name just a few. The SharePoint Products and Technologies installation usually includes multiple computers, called a **farm**.

Client applications, such as browsers, communicate with a **front-end Web server** within the farm to use the features provided by SharePoint Products and Technologies. Also, it is possible for external server applications, including those in other separate SharePoint Products and Technologies farms, to interact with front-end Web servers or a **service application** within a SharePoint Products and Technologies farm to use features from the farm or retrieve data from the farm. A SharePoint Products and Technologies front-end Web server can also communicate with service applications hosted on other servers within the farm to respond to user requests for data or to deliver a feature to a user or external server. In some cases, those service applications can be hosted on one or multiple **application servers** dedicated for that purpose within the farm.

Data created by users, as well as data used to configure and manage features and service applications provided by SharePoint Products and Technologies, is stored in databases hosted on **back-end database servers**. The main components of a SharePoint Products and Technologies farm are the front-end Web servers, application servers, and back-end database servers.

The diagram in section 2.1.1 provides a high-level overview of the system.

This document covers the protocols used between the client and server applications and the front-end Web servers for SharePoint Products and Technologies, and front-end Web servers and application servers within SharePoint Products and Technologies. Where appropriate, the document also describes the relationship between the protocols and the example scenarios in which they are used. The document is meant to facilitate an understanding of how to implement the protocols to support interoperability scenarios that involve SharePoint Products and Technologies. Many concepts that are described in this document at a high level are described in detail in [MS-WSSTS].

Some of the **Web services** and file-format protocols have changed significantly between SharePoint Products and Technologies versions. The new protocols are considered completely different than their predecessors, and cross-compatibility between versions is not supported. New versions of protocol documents feature the same titles as their predecessors, with the addition of "Version 2" or "Version 3" depending on the number of previous versions. Where protocols exist that are implemented in product versions prior to Microsoft SharePoint Server 2013 and Microsoft SharePoint Foundation 2013 those protocols are listed in separate version-specific tables in the section 2.2 protocol summary.

1.1 Glossary

This document uses the following terms:

Active Directory: The Windows implementation of a general-purpose **directory service**, which uses **LDAP** as its primary access protocol. **Active Directory** stores information about a variety of objects in the network such as user accounts, computer accounts, groups, and all related credential information used by Kerberos [MS-KILE]. **Active Directory** is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS), which are both described in [MS-ADOD]: Active Directory Protocols Overview.

Active Directory Service Interfaces (ADSI): A directory service model and a set of Component Object Model (COM) interfaces. ADSI enables Windows applications and Active Directory Domain Services (AD DS) clients to gain access to several network directory services, including AD DS.

alert: (1) An Internet message that is sent to subscribers automatically to notify them when user-defined criteria are met. Alerts are generated automatically when items such as documents, webpages, list items, sites, or other resources on a server are changed.

(2) A message that is passed to a protocol client to notify it when specific criteria are met.

alert subscription: A request to receive an Internet message automatically when user-defined criteria are met. Such messages are generated automatically when items such as documents, webpages, list items, sites, or other resources on a server are changed.

application server: A computer that provides infrastructure and services for applications that are hosted on a server farm.

ASP.NET: A web server technology for dynamically rendering HTML pages using a combination of HTML, Javascript, CSS, and server-side logic. For more information, see [ASPNET].

Asynchronous JavaScript + XML (AJAX): A web programming model that incorporates a set of web technologies including Extensible HyperText Markup Language (XHTML), cascading style sheets (CSS), Document Object Model (DOM), **XML**, Extensible Stylesheet Language Transformation (XSLT), XMLHTTPRequest (XHR), and JavaScript. AJAX is designed to make user interaction with the web more responsive.

atom feed: An **XML** structure that contains metadata about content, such as the language version and the date when the content was last modified, and is sent to subscribers by using the Atom Publishing Protocol (AtomPub), as described in [RFC4287].

audience: A named group of users that is used for targeting content.

authentication: The act of proving an identity to a server while providing key material that binds the identity to subsequent communications.

authentication mode: One of several modes in which an authentication exchange may be performed.

authorization: The secure computation of roles and accesses granted to an identity.

back-end database server: A server that hosts data, configuration settings, and stored procedures that are associated with one or more applications.

blog: A website that contains a series of posts about a subject and is arranged in reverse chronological order. Also referred to as web log.

browser-enable: The process of converting an InfoPath form template into a format that can be rendered in a web browser, and publishing it to and activating it on a protocol server that is running InfoPath Forms Services.

browser-enabled form template: A form template that is published to a protocol server that is running InfoPath Forms Services and is also activated for use on that server.

Business Connectivity Services (BCS): A set of services that enables interaction and integration with back-end data in a **line-of-business (LOB) system**, including external data and related services. BCS includes the **Business Data Connectivity (BDC)** service.

Business Data Connectivity (BDC): A shared service that stores information about business application data that exists outside a server farm. It can be used to display business data in lists, Web Parts, search results, user profiles, and custom applications. Previously referred to as Business Data Catalog.

calculate: The process by which computations in a workbook are performed.

Central Administration site: A SharePoint site that an administrator can use to manage all of the sites and servers in a server farm that is running SharePoint Products and Technologies.

change log: A log of changes, such as add and delete, that are made to objects that are stored on a **back-end database server**. Applications can use this information to identify changes that occurred on those objects.

Collaborative Application Markup Language (CAML): An XML-based language that is used to describe various elements, such as queries and views, in sites that are based on SharePoint Products and Technologies.

column: A single set of data that is displayed vertically in a worksheet or a table.

connection string: A series of arguments, delimited by a semicolon, that defines the location of a database and how to connect to it.

content database: A database that is stored on a **back-end database server** and contains stored procedures, site collections, and the contents of those site collections.

content source: A set of options for specifying the type of content to be crawled and the start addresses for the content to be indexed. A content source is defined by the protocol handler that is used to access specific systems, such as SharePoint sites, file systems, and external websites. A content source can contain up to 500 start addresses.

crawl: The process of traversing a URL space to acquire items to record in a search catalog.

crawler: A process that browses and indexes content from a content source.

credential: Previously established, **authentication** data that is used by a security principal to establish its own identity. When used in reference to the Netlogon Protocol, it is the data that is stored in the NETLOGON_CREDENTIAL structure.

dashboard: A visual interface that displays a related group of interactive scorecard and report views. It provides views into key measures that are relevant to a business practice or process. Dashboard elements provide capabilities, such as shared filters, that enable users to perform tasks such as highlighting trends, comparing data, and controlling the data that is displayed.

data connection: (1) A link between an application and a data source. Data connections can be used to query and submit data.

(2) A collection of information, such as the type and location, that defines how to connect to an external data source, such as a database, **web service**, SharePoint list, or **XML** file.

(3) A connection between an InfoPath form template and an external data source, as specified by settings in an InfoPath **form template (.xsn) file** or a **Universal Data Connection (.udc, .udcx) file**.

data macro: A component that implements application logic and enables recognition of built-in actions and tasks for list items.

data range: A set of consecutive scale-out partition keys.

database application: A set of objects, including tables, queries, forms, reports, macros, and code modules, that are stored in a database structure.

decomposition tree: A data visualization tool that facilitates analysis of complex information by using a hierarchical scheme. It enables users to find the root cause of the value of a specific, key performance indicator (KPI).

deployment package: A collection of files that can be used to deploy and manage customizations, such as add-ins, to a computer. It consists of an application manifest, a deployment manifest, and related package files.

directory service (DS): A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also **Active Directory**.

discovery case: A site that contains information relevant to an electronic discovery (eDiscovery) case such as a custodian, a **discovery source**, and saved searches.

discovery source: A repository of documents and other types of content that are relevant to the electronic discovery (eDiscovery) case.

distribution list: A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

document: An object in a **content database** such as a file, folder, **list (1)**, or **site**. Each object is identified by a **URI**.

document library: A type of list that is a container for documents and folders.

Document Workspace site: A SharePoint site that is based on a Document Workspace site template and has a template identifier value of "1". A Document Workspace site is used for planning, posting, and working together on a document or a set of related documents.

embed code: A block of data containing information needed to embed a video in a user interface element. This information can include the path to the video, the height and width of the video, and so on.

EntityInstance: A set of Field values that have a unique identity that represents a specific instance of an Entity, and are stored in a line-of-business (LOB) system.

event: (1) Any significant occurrence in a system or an application that requires users to be notified or an entry to be added to a log.

(2) An action or occurrence to which an application might respond. Examples include state changes, data transfers, key presses, and mouse movements.

external data: Data that is stored in a repository outside a workbook.

external list: A container that is within a SharePoint site and that references a set of **EntityInstances** that are sourced from a line-of-business (LOB) system. It has a customizable schema that is composed of one or more Fields.

farm: A group of computers that work together as a single system to help ensure that applications and resources are available. Also referred to as server farm.

file: A single, discrete unit of content.

font: An object that defines the graphic design, or formatting, of a collection of numbers, symbols, and letters. A font specifies the style (such as bold and strikeout), size, family (a typeface such as Times New Roman), and other qualities to describe how the collection is drawn.

form: A document with a set of controls into which users can enter information. Controls on a form can be bound to elements in the data source of the form, such as fields and groups. See also bind.

form digest validation: A type of security validation that helps prevent an attack wherein users are tricked into posting data to a server.

form file: An **XML** file that contains data that is entered into an InfoPath form by using a web browser or Microsoft InfoPath.

form library: A type of document library that is optimized for storing and displaying data in XML-based forms.

form server: A server that can host XML-based electronic forms and that supports rendering those forms in a web browser.

form template: A file or set of files that defines the data structure, appearance, and behavior of a **form**.

form template (.xsn) file: A cabinet (.cab) file with an .xsn file name extension that contains the files that comprise a form template.

forms authentication: An **authentication** method in which protocol clients redirect unauthenticated requests to an **HTML** form by using **HTTP**. If the protocol client authenticates the request, the system issues a cookie that stores the credentials or a key for reacquiring the identity. In subsequent requests, the cookie is submitted in request headers and the requests are authenticated and authorized by an **ASP.NET** event handler that uses the validation method that is specified by the protocol client.

front-end web server: A server that hosts webpages, performs processing tasks, and accepts requests from protocol clients and sends them to the appropriate back-end server for further processing.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

group: (1) A named collection of users who share similar access permissions or roles.

(2) A named collection of quick links, colleagues, or memberships for the purpose of organization.

hierarchy: A logical tree structure that organizes the members of a dimension such that each member has one parent member and zero or more child members.

home page: On the World Wide Web, a document that serves as a starting point for a set of webpages and other files in a website.

HTTP GET: An HTTP method for retrieving a resource, as described in [RFC2616].

hyperlink: A relationship between two anchors, as described in [RFC1866].

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

identity: A digital persona that is associated with two key pairs, one for encrypting data and another for signing data.

immediate translation job: A **translation job** that contains only one translation item and is executed as a higher priority than a typical translation job.

index server: A server that is assigned the task of crawling.

Internet Information Services (IIS): The services provided in Windows implementation that support web server functionality. **IIS** consists of a collection of standard Internet protocol servers such as HTTP and FTP in addition to common infrastructures that are used by other Microsoft Internet protocol servers such as SMTP, NNTP, and so on. **IIS** has been part of the Windows operating system in some versions and a separate install package in others. **IIS** version 5.0 shipped as part of Windows 2000 operating system, **IIS** version 5.1 as part of Windows XP operating system, **IIS** version 6.0 as part of Windows Server 2003 operating system, and **IIS** version 7.0 as part of Windows Vista operating system and Windows Server 2008 operating system.

inverted index: For each token that is encountered in a corpus of indexed items, a data structure that stores a list of postings that identify which documents matched and a list of occurrences that identify which position in each document.

item: A unit of content that can be indexed and searched by a search application.

legal hold: A restriction that prevents a document from being modified or transactions from being entered for a record.

Lightweight Directory Access Protocol (LDAP): The primary access protocol for **Active Directory**. Lightweight Directory Access Protocol (LDAP) is an industry-standard protocol, established by the Internet Engineering Task Force (IETF), which allows users to query and update information in a **directory service (DS)**, as described in [MS-ADTS]. The Lightweight Directory Access Protocol can be either version 2 [RFC1777] or version 3 [RFC3377].

line-of-business (LOB) system: A software system that is used to store business data and can also contain business rules and business logic that support business processes.

list: (1) A container within a SharePoint site that stores list items. A list has a customizable schema that is composed of one or more fields.

(2) An organization of a region of cells into a tabular structure in a workbook.

list item: An individual entry within a SharePoint list. Each list item has a schema that maps to fields in the list that contains the item, depending on the content type of the item.

list template: An XML-based definition of list settings, including fields and views, and optionally list items. List templates are stored in .stp files in the content database.

master secret: A key that is used to symmetrically encrypt and decrypt credentials and single sign-on (SSO) tickets.

master secret server: A protocol server that stores and can provide a master secret in response to a request from a protocol client.

Meeting Workspace site: A SharePoint site that is based on a Meeting Workspace site template and has a template ID value of "2". A Meeting Workspace site is used for planning, posting, and working together on meeting materials.

membership: The state or status of being a member of a member group. A membership contains additional metadata such as the privacy level that is associated with the membership.

metadata store: A database that is stored on a **back-end database server** and contains all stored procedures and storage for the MetadataObject types.

OLE DB: A set of interfaces that are based on the Component Object Model (COM) programming model and expose data from a variety of sources. These interfaces support the amount of Database Management System (DBMS) functionality that is appropriate for a data store and they enable a data store to share data.

Online Analytical Processing (OLAP): A technology that uses multidimensional structures to provide access to data for analysis. The source data for OLAP is stored in data warehouses in a relational database. See also cube.

Open Database Connectivity (ODBC): A standard software API method for accessing data that is stored in a variety of proprietary personal computer, minicomputer, and mainframe databases. It is an implementation of [ISO/IEC9075-3:2008] and provides extensions to that standard.

permission: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also rights.

personal site: A type of SharePoint site that is used by an individual user for personal productivity. The site appears to the user as My Site.

picture library: A type of **document library** that is optimized for storing digital pictures or graphics.

portal site: A type of SharePoint site that can act as an umbrella to other sites and can be used by a large organization.

presentation: A collection of slides that are intended to be viewed by an audience.

principal: In Kerberos, a Kerberos principal.

property promotion: A process in which a field in an InfoPath form is made available for display as a column in a SharePoint list or the name of an Outlook folder, or for use as a parameter in a Web Part connection.

property set: A set of attributes, identified by a **GUID**. Granting access to a property set grants access to all the attributes in the set.

publish: The process of making an InfoPath form template available for people to fill out by using Microsoft InfoPath or a web browser.

published: A condition of portions of a workbook that are marked as being available to the user when that workbook is processed by a protocol server.

query: A formalized instruction to a data source to either extract data or perform a specified action. A query can be in the form of a query expression, a method-based query, or a combination of the two. The data source can be in different forms, such as a relational database, XML document, or in-memory object. See also **search query**.

query server: A server that has been assigned the task of fulfilling search queries.

remote procedure call (RPC): A communication protocol used primarily between client and server. The term has three definitions that are often used interchangeably: a runtime environment providing for communication facilities between computers (the RPC runtime); a set of request-and-response message exchanges between computers (the RPC exchange); and the single message from an RPC exchange (the RPC message). For more information, see [C706].

scorecard: A report that depicts organizational and business performance by displaying a collection of key performance indicators (KPIs) with performance targets for those KPIs. Each KPI compares actual performance to goals for an area. A scorecard can be organized hierarchically and typically contains visualization tools such as trend charts and conditional formatting.

search query: A complete set of conditions that are used to generate search results, including query text, sort order, and ranking parameters.

search service application: A shared service application that provides indexing and querying capabilities.

Secure Store Service (SSS): A service that is used to store credentials for a user or a group of users. It enables applications, typically on behalf of a user, to authenticate and gain access to resources. Users can retrieve only their own credentials from the secure store.

Security Assertion Markup Language (SAML): The set of specifications that describe security assertions encoded in **XML**, profiles for attaching assertions to protocols and frameworks, request/response protocols used to obtain assertions, and the protocol bindings to transfer protocols, such as **SOAP** and HTTP.

server: A replicating machine that sends replicated files to a partner (client). The term "server" refers to the machine acting in response to requests from partners that want to receive replicated files.

service application: A middle-tier application that runs without any user interface components and supports other applications by performing tasks such as retrieving or modifying data in a database.

shape: A collection of qualifiers, such as names, and quantifiers, such as coordinates, that is used to represent a geometric object. A shape can be contained in a document, file structure, run-time structure, or other medium.

share: A resource offered by a Common Internet File System (CIFS) server for access by CIFS clients over the network. A **share** typically represents a directory tree and its included files (referred to commonly as a "disk share" or "file share") or a printer (a "print share"). If the information about the **share** is saved in persistent store (for example, Windows registry) and reloaded when a file server is restarted, then the **share** is referred to as a "sticky share". Some **share** names are reserved for specific functions and are referred to as special **shares**: IPC$, reserved for interprocess communication, ADMIN$, reserved for remote administration, and A$, B$, C$ (and other local disk names followed by a dollar sign), assigned to local disk devices.

site: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

site collection: A set of websites that are in the same **content database**, have the same owner, and share administration settings. A site collection can be identified by a **GUID** or the **URL** of the top-level site for the site collection. Each site collection contains a top-level site, can contain one or more subsites, and can have a shared navigational structure.

site definition: A family of site definition configurations. Each site definition specifies a name and contains a list of associated site definition configurations.

site membership: The status of being a member of a site and having a defined set of user rights for accessing or managing content on that site.

site template: An XML-based definition of site settings, including formatting, lists, views, and elements such as text, graphics, page layout, and styles. Site templates are stored in .stp files in the content database.

Slide Library: A type of a document library that is optimized for storing and reusing presentation slides that conform to the format described in [ISO/IEC-29500:2008].

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses **XML** technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

social data: A collection of ratings, tags, and comments about webpages and items on a SharePoint site or the Internet. Individual users create this data and, by default, share it with other users.

social networking: The use of websites and services that provide enhanced information and interaction capabilities with regard to people and resources.

social rating: A user-defined value that indicates the perceived quality of a webpage or item on a SharePoint site or the Internet. Individual users create these ratings and, by default, share them with other users.

social tag: A user-defined keyword and hyperlink to a webpage or item on a SharePoint site or the Internet. Individual users create these tags and, by default, share them with other users.

solution gallery: A gallery that is used to store solution packages.

solution package: A compressed file that can be deployed to a server farm or a site. It can contain assemblies, resource files, site and feature definitions, templates, code access security policies, and Web Parts. Solution packages have a .wsp file name extension.

task: An object that represents an assignment to be completed.

TCP/IP: A set of networking protocols that is widely used on the Internet and provides communications across interconnected networks of computers with diverse hardware architectures and various operating systems. It includes standards for how computers communicate and conventions for connecting networks and routing traffic.

term: A concept or an idea that is stored and can be used as metadata.

term label: A string that is used as the display value for a term. Each term label is associated with a specific language.

term set: A collection of terms that are arranged into and stored as a hierarchy or a flat list.

term store: A database in which managed metadata is stored in the form of **term sets** and **terms**.

translation job: A set of translation items which are created at the same time and share the same machine translation settings, including target language.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

Universal Data Connection (.udc, .udcx) file: An **XML** file that has a .udc or .udcx file name extension that contains user credentials and other authentication information that is used to connect to a data source.

user profile: A collection of properties that pertain to a specific person or entity within a portal site.

User Profile Service: A data source that stores, provides, and applies information about users.

user profile store: A database that stores information about each user profile.

video container: A data type that stores information needed to display videos on a collaboration server.

view: See form view (Microsoft InfoPath), list view (SharePoint Products and Technologies), or **View** (Microsoft Business Connectivity Services).

web application: A container in a configuration database that stores administrative settings and entry-point **URLs** for **site collections**.

web control: A server-side component that encapsulates user interface and related functionality.

Web Distributed Authoring and Versioning Protocol (WebDAV): The Web Distributed Authoring and Versioning Protocol, as described in [RFC2518] or [RFC4918].

Web Part: A reusable component that contains or generates web-based content such as **XML**, **HTML**, and scripting code. It has a standard property schema and displays that content in a cohesive unit on a webpage. See also Web Parts Page.

web server: A server computer that hosts websites and responds to requests from applications.

web service: A unit of application logic that provides data and services to other applications and can be called by using standard Internet transport protocols such as **HTTP**, Simple Mail Transfer Protocol (SMTP), or File Transfer Protocol (FTP). Web services can perform functions that range from simple requests to complicated business processes.

website: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and site.

workbook: A container for a collection of sheets.

workflow: (1) An automation of business processes that passes business documents and tasks automatically from one user to another for action, according to a defined sequence.

(2) A structured modular component that enables the automated movement of documents or items through a specific sequence of actions or tasks that are related to built-in or user-defined business processes.

workflow association: An association of a workflow template to a specific list or content type.

workflow events: An event that starts or resumes a **workflow instance**.

workflow instance: An instance of a workflow association that performs on a list item the process that is defined in a workflow template.

XML: The Extensible Markup Language, as described in [XML1.0].

1.2   References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

[MC-FPSEWM] Microsoft Corporation, "FrontPage Server Extensions: Website Management Protocol".

[MS-AADT] Microsoft Corporation, "Access Application Design Time Protocol".

[MS-ACCDT] Microsoft Corporation, "Access Template File Format".

[MS-ACMAUDWS] Microsoft Corporation, "Audit and Control Management Server Audit Web Service Protocol".

[MS-ADMINS] Microsoft Corporation, "Administration Web Service Protocol".

[MS-ADMWS] Microsoft Corporation, "Search Service Administration Web Service Protocol".

[MS-ADR] Microsoft Corporation, "Access Services Data Run Time Protocol".

[MS-ADS] Microsoft Corporation, "Access Services Data Server Protocol".

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification".

[MS-ALERTSS] Microsoft Corporation, "Alerts Service Protocol".

[MS-APPMWSP] Microsoft Corporation, "SharePoint App Management Web Service Protocol".

[MS-ART] Microsoft Corporation, "Access Run Time Protocol".

[MS-ASDT] Microsoft Corporation, "Access Server Design Time Protocol".

[MS-ASWS] Microsoft Corporation, "Access Services Protocol".

[MS-AUTHWS] Microsoft Corporation, "Authentication Web Service Protocol".

[MS-AXL2] Microsoft Corporation, "Access Application Transfer Protocol Structure Version 2".

[MS-AXL] Microsoft Corporation, "Access Application Transfer Protocol Structure".

[MS-BCSDPFFS] Microsoft Corporation, "Business Connectivity Services Deployment Package File Format".

[MS-BCSDWPS] Microsoft Corporation, "Business Connectivity Services Deployment Web Service Protocol".

[MS-BDCDP] Microsoft Corporation, "Business Data Catalog Data Web Service Protocol".

[MS-BDCMFFS] Microsoft Corporation, "Business Data Connectivity Model File Format".

[MS-BDCMP] Microsoft Corporation, "Business Data Catalog Metadata Web Service Protocol".

[MS-BDCRAWPS] Microsoft Corporation, "Business Data Connectivity Remote Administration Web Service Protocol".

[MS-BDCWPS] Microsoft Corporation, "Business Data Connectivity Web Service Protocol".

[MS-CDEPLOY] Microsoft Corporation, "Content Deployment Remote Import Web Service Protocol".

[MS-CIFO] Microsoft Corporation, "Content Index Format Structure".

[MS-COMMCSOM] Microsoft Corporation, "Community Client-Side Object Model Protocol".

[MS-CONATB] Microsoft Corporation, "Content Area Toolbox Web Service Protocol".

[MS-COPYS] Microsoft Corporation, "Copy Web Service Protocol".

[MS-CPSWS] Microsoft Corporation, "SharePoint Claim Provider Web Service Protocol".

[MS-CSOMREST] Microsoft Corporation, "SharePoint Client Query OData Protocol".

[MS-CSOMSPT] Microsoft Corporation, "SharePoint Client-Side Object Model Protocol".

[MS-CSOM] Microsoft Corporation, "SharePoint Client Query Protocol".

[MS-DADMWS] Microsoft Corporation, "Search Admin Web Service Protocol".

[MS-DMCSOM] Microsoft Corporation, "Document Management Client-Side Object Model Protocol".

[MS-DOCSWS] Microsoft Corporation, "SharePoint Document Sharing Web Service Protocol".

[MS-DOCTRANS] Microsoft Corporation, "Document Transforms Service Protocol".

[MS-DSPSTSS] Microsoft Corporation, "Data-Source Adapter SharePoint Team Services Web Service Protocol".

[MS-DWSS] Microsoft Corporation, "Document Workspace Web Service Protocol".

[MS-EAWF] Microsoft Corporation, "Group Approval Web Service Protocol".

[MS-ECTPWPS] Microsoft Corporation, "External Content Type Picker Web Service Protocol".

[MS-EDCSOM] Microsoft Corporation, "eDiscovery Client-Side Object Model Protocol".

[MS-EDINTWS] Microsoft Corporation, "eDiscovery Internal Web Service Protocol".

[MS-EDSA] Microsoft Corporation, "eDiscovery Service Application Protocol".

[MS-EMMCSOM] Microsoft Corporation, "Enterprise Managed Metadata Client-Side Object Model Protocol".

[MS-EMMWCF] Microsoft Corporation, "Metadata Shared Service WCF Service Protocol".

[MS-EMMWS] Microsoft Corporation, "Microsoft Enterprise Managed Metadata Web Service Protocol".

[MS-ESP] Microsoft Corporation, "Excel Web Services Protocol".

[MS-ESREST] Microsoft Corporation, "Excel Services REST Protocol".

[MS-ESURL] Microsoft Corporation, "Excel Services Publishing Protocol".

[MS-EXSPWS2] Microsoft Corporation, "Excel Calculation Web Service Version 2 Protocol".

[MS-EXSPWS3] Microsoft Corporation, "Excel Calculation Web Service Version 3 Protocol".

[MS-EXSPWS] Microsoft Corporation, "Excel Calculation Web Service Protocol".

[MS-EXSPXML3] Microsoft Corporation, "Excel Calculation Version 2 Web Service XML Schema".

[MS-EXSPXML] Microsoft Corporation, "Excel Calculation Web Service XML Schema".

[MS-FORMS] Microsoft Corporation, "Forms Service Protocol".

[MS-FPSE] Microsoft Corporation, "FrontPage Server Extensions Remote Protocol".

[MS-FQL2] Microsoft Corporation, "Fast Query Language Version 2 Protocol".

[MS-FSDAP] Microsoft Corporation, "Forms Services Design and Activation Web Service Protocol".

[MS-FSFDP] Microsoft Corporation, "Forms Services Feature Detection Protocol".

[MS-FSPP] Microsoft Corporation, "Forms Services Proxy Web Service Protocol".

[MS-FSSHTTPB] Microsoft Corporation, "Binary Requests for File Synchronization via SOAP Protocol".

[MS-FSSHTTPD] Microsoft Corporation, "Binary Data Format for File Synchronization via SOAP".

[MS-FSSHTTP] Microsoft Corporation, "File Synchronization via SOAP over HTTP Protocol".

[MS-GLOADWS] Microsoft Corporation, "Global Admin Web Service Protocol".

[MS-IMAGS] Microsoft Corporation, "Imaging Service Protocol".

[MS-INFODCF] Microsoft Corporation, "InfoPath Data Connection File Download Protocol".

[MS-IPDSP] Microsoft Corporation, "InfoPath Digital Signing Protocol".

[MS-IPFF2] Microsoft Corporation, "InfoPath Form Template Format Version 2".

[MS-IPFFX] Microsoft Corporation, "InfoPath Form File Format".

[MS-IPFF] Microsoft Corporation, "InfoPath Form Template Format".

[MS-KQL] Microsoft Corporation, "Keyword Query Language Structure Protocol".

[MS-LISTSWS] Microsoft Corporation, "Lists Web Service Protocol".

[MS-MAR] Microsoft Corporation, "Microsoft Office SharePoint Server (MOSS) Analytics Reporting Protocol".

[MS-MEETS] Microsoft Corporation, "Meetings Web Services Protocol".

[MS-METAWEB] Microsoft Corporation, "MetaWeblog Extensions Protocol".

[MS-NOTESWS] Microsoft Corporation, "MS Search Lotus Notes Web Service Protocol".

[MS-OAUTH2EX] Microsoft Corporation, "OAuth 2.0 Authentication Protocol Extensions".

[MS-OCPROTO] Microsoft Corporation, "Office Client Protocols Overview".

[MS-OFBA] Microsoft Corporation, "Office Forms Based Authentication Protocol".

[MS-OFFICIALFILE] Microsoft Corporation, "Official File Web Service Protocol".

[MS-OMS] Microsoft Corporation, "Office Mobile Service Protocol".

[MS-OOWQWS] Microsoft Corporation, "Office Online Web Query Web Service Protocol".

[MS-OSALER] Microsoft Corporation, "Alerts Interoperability Protocol".

[MS-OSCO] Microsoft Corporation, "Office Server ClickOnce Manifest Structure".

[MS-OUTSPS] Microsoft Corporation, "Lists Client Sync Protocol".

[MS-PASCWS] Microsoft Corporation, "PowerPoint Automation Services Conversion Web Service Protocol".

[MS-PEOPS] Microsoft Corporation, "People Web Service Protocol".

[MS-PERMS] Microsoft Corporation, "Permissions Web Service Protocol".

[MS-PLSP] Microsoft Corporation, "Published Links Web Service Protocol".

[MS-PPSAPP] Microsoft Corporation, "PerformancePoint Services Application Server Protocol".

[MS-PPSAS] Microsoft Corporation, "PerformancePoint Services Authoring Service Protocol".

[MS-PPSDECO] Microsoft Corporation, "PerformancePoint Services Decomposition Tree Protocol".

[MS-PRIMEPF] Microsoft Corporation, "Deployment Package Format".

[MS-PRSTFR] Microsoft Corporation, "ADO XML Persistence Format".

[MS-PUBCSOM] Microsoft Corporation, "Publishing Client-Side Object Model Protocol".

[MS-PUBWS] Microsoft Corporation, "Publishing Web Service Protocol".

[MS-QSSWS] Microsoft Corporation, "Search Query Shared Services Protocol".

[MS-SADMWS] Microsoft Corporation, "Search Site Administration Web Service Protocol".

[MS-SEARCH] Microsoft Corporation, "Search Protocol".

[MS-SHDACCWS] Microsoft Corporation, "Shared Access Web Service Protocol".

[MS-SITED3S] Microsoft Corporation, "Site Data 2003 Web Service Protocol".

[MS-SITEDATS] Microsoft Corporation, "Site Data Web Service Protocol".

[MS-SITESS] Microsoft Corporation, "Sites Web Service Protocol".

[MS-SLIDELI] Microsoft Corporation, "Slide Library Web Service Protocol".

[MS-SLXV] Microsoft Corporation, "Silverlight XAML Vocabulary Specification 2008".

[MS-SOCCSOM] Microsoft Corporation, "SharePoint Social Client-Side Object Model Protocol".

[MS-SPACSOM] Microsoft Corporation, "SharePoint Analytics Client-Side Object Model Protocol".

[MS-SPAFWI] Microsoft Corporation, "SharePoint Activity Feed Web Interfaces Protocol".

[MS-SPASA] Microsoft Corporation, "SharePoint Analytics Service Application Protocol".

[MS-SPDIAG] Microsoft Corporation, "SharePoint Diagnostics Web Service Protocol".

[MS-SPEMAWS] Microsoft Corporation, "SharePoint Email Web Service Protocol".

[MS-SPLCHK] Microsoft Corporation, "SpellCheck Web Service Protocol".

[MS-SPPTC] Microsoft Corporation, "User Code Execution Protocol".

[MS-SPRSS] Microsoft Corporation, "RSS 2.0 Format Extensions".

[MS-SPS2SAUTH] Microsoft Corporation, "OAuth 2.0 Authentication Protocol: SharePoint Profile".

[MS-SPSCRWL] Microsoft Corporation, "SPSCrawl Web Service Protocol".

[MS-SPSETWS] Microsoft Corporation, "SharePoint Subscription Settings Web Service Protocol".

[MS-SPSTWS] Microsoft Corporation, "SharePoint Security Token Service Web Service Protocol".

[MS-SPTWS] Microsoft Corporation, "Service Platform Topology Web Service Protocol".

[MS-SPWFCSOM] Microsoft Corporation, "SharePoint Workflow Client-Side Object Model Protocol".

[MS-SQP2] Microsoft Corporation, "MSSearch Query Version 2 Protocol".

[MS-SQP] Microsoft Corporation, "MSSearch Query Protocol".

[MS-SRCHCSOM] Microsoft Corporation, "Search Client Query Protocol".

[MS-SSAS] Microsoft Corporation, "SQL Server Analysis Services Protocol".

[MS-SSP] Microsoft Corporation, "Single Sign-On Protocol".

[MS-SSWPS] Microsoft Corporation, "Secure Store Web Service Protocol".

[MS-STSSYN] Microsoft Corporation, "StsSync Data Structure".

[MS-SYS-ARCHIVE] Microsoft Corporation, "Windows System Overview", https://docs.microsoft.com/en-us/openspecs/windows_protocols/MS-WINPROTLP/df36f95e-6a6b-48d6-a3ae-35a17674f546

[MS-TDS] Microsoft Corporation, "Tabular Data Stream Protocol".

[MS-TMPLDISC] Microsoft Corporation, "Template Discovery Web Service Protocol".

[MS-TSCSOM] Microsoft Corporation, "Translation Services Client-Side Object Model Protocol".

[MS-TSWS] Microsoft Corporation, "Translation Services Web Service Protocol".

[MS-UDCX] Microsoft Corporation, "Universal Data Connection 2.0 XML File Format".

[MS-UGS] Microsoft Corporation, "UserGroup Web Service Protocol".

[MS-UPFCWS] Microsoft Corporation, "User Activity Feed Cache Web Service Protocol".

[MS-UPIEWS] Microsoft Corporation, "User Profile Import and Export Web Service Protocol".

[MS-UPSCDS] Microsoft Corporation, "User Profile Synchronization (UPS): Configuration Data Structure".

[MS-UPSCP] Microsoft Corporation, "User Profile Synchronization (UPS): Configuration Protocol Extensions".

[MS-UPSCWS] Microsoft Corporation, "User Profile Service Application Caching Web Service Protocol".

[MS-UPSDWS] Microsoft Corporation, "User Profile Social Data Web Service Protocol".

[MS-UPSLDAP] Microsoft Corporation, "User Profile Synchronization (UPS): Lightweight Directory Access Protocol Version 3 Extensions".

[MS-UPSMODS] Microsoft Corporation, "User Profile Synchronization (UPS): Management Objects Data Structure".

[MS-UPSSCXPP] Microsoft Corporation, "User Profile Synchronization (UPS): Schema Exchange Protocol Profile".

[MS-UPWCFWS] Microsoft Corporation, "User Profile Property Service Application Web Service Protocol".

[MS-USPSWS] Microsoft Corporation, "User Profile Service Web Service Protocol".

[MS-USRPCH] Microsoft Corporation, "User Profile Change Log Web Service Protocol".

[MS-VERSS] Microsoft Corporation, "Versions Web Service Protocol".

[MS-VGSFF] Microsoft Corporation, "Visio Graphics Service (.vdw) File Format".

[MS-VGSP] Microsoft Corporation, "Visio Graphics Service Protocol".

[MS-VIEWSS] Microsoft Corporation, "Views Web Service Protocol".

[MS-VSDX] Microsoft Corporation, "Visio Graphics Service VSDX File Format".

[MS-WDVME] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Protocol: Microsoft Extensions".

[MS-WDVMODUU] Microsoft Corporation, "Office Document Update Utility Extensions".

[MS-WDVSE] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Protocol: Server Extensions".

[MS-WDV] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Protocol: Client Extensions".

[MS-WEBDAVE] Microsoft Corporation, "Web Distributed Authoring and Versioning Error Extensions Protocol".

[MS-WEBSS] Microsoft Corporation, "Webs Web Service Protocol".

[MS-WMCSOM] Microsoft Corporation, "Work Management Client-Side Object Model Protocol".

[MS-WMS] Microsoft Corporation, "Work Management Service Web Service Protocol".

[MS-WORDSWCF] Microsoft Corporation, "Word Automation Services WCF Service Protocol".

[MS-WPPS] Microsoft Corporation, "Web Part Pages Web Service Protocol".

[MS-WSSCAML] Microsoft Corporation, "Collaborative Application Markup Language (CAML) Structure".

[MS-WSSCAP] Microsoft Corporation, "Windows SharePoint Services Collaborative Application Protocol".

[MS-WSSHP] Microsoft Corporation, "HTTP Windows SharePoint Services Headers Protocol".

[MS-WSSREST] Microsoft Corporation, "ListData Data Service Protocol".

[MS-WSSTS] Microsoft Corporation, "Windows SharePoint Services".

[MS-WWSP] Microsoft Corporation, "Workflow Web Service Protocol".

[MSDN-ASPWP] Microsoft Corporation, "ASP.NET Web Parts Overview", http://msdn.microsoft.com/en-us/library/hhy9ewf1.aspx

[MSDN-ESO] Microsoft Corporation, "Excel Services Overview", http://msdn.microsoft.com/en-us/library/ms546696.aspx

[MSDN-EXCELS] Microsoft Corporation, "Introducing Excel Services", http://msdn.microsoft.com/en-us/library/ms519581.aspx

[MSDN-MOSSWF] Microsoft Corporation, "Workflows in SharePoint Server 2010 (ECM)", http://msdn.microsoft.com/en-us/library/ms549489.aspx

[MSDN-OLEDB] Microsoft Corporation, "Microsoft OLE DB", https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ms722784(v=vs.85)

[MSDN-OpenDBConnectivity] Microsoft Corporation, "Microsoft Open Database Connectivity (ODBC)", http://msdn.microsoft.com/en-us/library/ms710252.aspx

[MSDN-SharePoint2013-SDK] Microsoft Corporation, "SharePoint 2013", http://msdn.microsoft.com/en-us/library/jj162979.aspx

[MSDN-SharePointSDK] Microsoft Corporation, "SharePoint Products and Technologies SDK: 2010 API Reference (Technical Preview)", July 2009, http://msdn.microsoft.com/en-us/library/ee557253(office.14).aspx

[MSDN-WSSWF] Microsoft Corporation, "Workflow Development for SharePoint Foundation", http://msdn.microsoft.com/en-us/library/ms414613.aspx

[MSDN-WSSWP] Microsoft Corporation, "Web Parts in SharePoint Foundation", http://msdn.microsoft.com/en-us/library/ms476318.aspx

[MSFT-ADC] Microsoft Corporation, "Active Directory Collection", March 2003, http://technet2.microsoft.com/WindowsServer/en/library/6f8a7c80-45fc-4916-80d9-16e6d46241f91033.mspx

[RFC2518] Goland, Y., Whitehead, E., Faizi, A., et al., "HTTP Extensions for Distributed Authoring - WebDAV", RFC 2518, February 1999, http://www.ietf.org/rfc/rfc2518.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.rfc-editor.org/rfc/rfc2616.txt

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt

[RFC4510] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", June 2006, http://www.rfc-editor.org/rfc/rfc4510.txt

[SAMLToken1.1] Lawrence, K., Kaler, C., Monzillo, R., et al., "Web Services Security: SAML Token Profile 1.1", February 2006, http://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTokenProfile.pdf

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[SOAP1.2-2/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part2-20070427

[TSQL-Ref] Microsoft Corporation, "Transact-SQL Reference (Transact-SQL)", http://msdn.microsoft.com/en-us/library/dd884419.aspx

[WSFederation] Kaler, C., Nadalin, A., Bajaj, S., et al., "Web Services Federation Language (WS-Federation)", Version 1.1, December 2006, http://specs.xmlsoap.org/ws/2006/12/federation/ws-federation.pdf

[WSTrust] IBM, Microsoft, Nortel, VeriSign, "WS-Trust V1.0", February 2005, http://specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf

[XMLSCHEMA] World Wide Web Consortium, "XML Schema", September 2005, http://www.w3.org/2001/XMLSchema

2   Functional Architecture

The following sections describe the functional architecture of the SharePoint system.

2.1   Overview

SharePoint Products and Technologies comprises a Web-based platform that provides both an infrastructure for building and deploying applications and a set of built-in capabilities and features. The following sections describe the high-level topology, architecture, and deployment of SharePoint Products and Technologies. For a technical description of SharePoint Products and Technologies concepts, see [MS-WSSTS].

2.1.1   Deployment Topology

SharePoint Products and Technologies can be deployed in a wide range of topologies to provide high availability and various degrees of scale-out, depending on the requirements of particular deployment scenarios. A particular deployment of one or more **servers** is called a **farm**. The following figure shows the protocol topology for SharePoint Products and Technologies. This figure shows the components deployed on separate servers, but the components can also run compatibly on a single server.
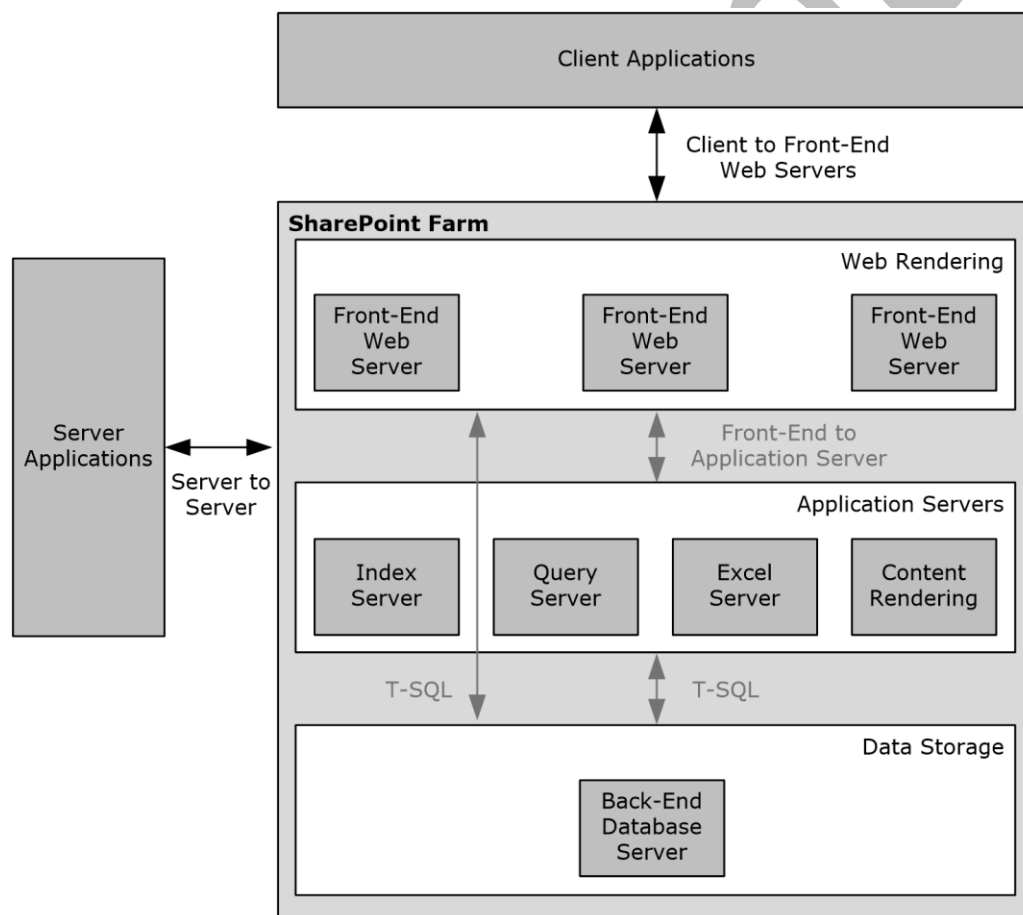
Figure 1: SharePoint Products and Technologies deployment topology

Servers of each type can be added to a deployment of SharePoint Products and Technologies as needed by the specific usage scenario. These additional servers, such as **front-end Web servers**, supply additional computing resources to provide load-balancing and scalability.

The following sections briefly describe the SharePoint Products and Technologies deployment topology.

2.1.1.1 SharePoint Products and Technologies Farm

As shown in the figure in section 2.1.1, a SharePoint Products and Technologies deployment can be scaled-out across multiple **servers** in a **farm** deployment to increase throughput and availability.

SharePoint Products and Technologies supports network load-balancing technologies that distribute user requests across multiple servers in a farm. These individual **front-end Web servers** are stateless; that is, any front-end Web server in the farm is prepared to handle any user request in the same way as any other front-end Web server in the farm.

To provide additional resource management, SharePoint Products and Technologies also supports the ability to deploy servers as **application servers** to provide dedicated infrastructure and services for specific service applications or features that are provided by a farm. For example, a specific server could be created as the Excel Services application server within a farm and be the resource to do all Excel Services calculations for user requests to that farm. This allows any front-end Web servers to offload that possibly resource-heavy work and avoid affecting other user requests.

From a back-end storage perspective, as the SharePoint Products and Technologies deployment grows and the capacity of an individual back-end database server is fully consumed, additional back-end database server resources can be deployed by adding additional servers that host completely separate content databases. Different **site collections** can be deployed into those separate **content databases**, and when a user request comes to a particular front-end Web server, that front-end Web server will retrieve the **site** content strictly from the appropriate back-end database. This provides the ability to load-balance across multiple back-end resources, but does require manual placement of high-load sites into separate content databases.

2.1.1.2 Web Infrastructure

SharePoint Products and Technologies provides the ability to create and manage **websites**. It does so within the framework of two operating-system components: **Internet Information Services (IIS)** and **ASP.NET**, which provide core services for **Hypertext Transfer Protocol (HTTP)**, process management, and programmability.

2.1.1.2.1 Web Topology

In **IIS**, a **web application** acts as both a container for **websites** and a process isolation boundary. A deployment of SharePoint Products and Technologies includes three types of web applications in IIS:

- A **Central Administration site** web application that handles the administration of websites and of the farm running SharePoint Products and Technologies.

- A **service application** web application that enables key services spanning multiple websites.

- Zero or more run-time web applications that handle user requests.

A functioning farm includes at least one run-time web application. Such a web application is necessary for creating websites for user content and applications. In a common usage scenario, a browser accesses each web application via an independent **TCP/IP** port. For example, the Central Administration site web application might function over port 1002 and be accessed via the http://myserver:1002 **URL**. The run-time web application might function over TCP/IP port 80, which is the default for HTTP.

This separation among the web applications provides both simplicity and security:

- Simplicity for users who do not need to see the complexity that is associated with administration and service applications.

- Security, because the user content, service applications, and Central Administration site all run in separate processes with separate **permissions** and **authentication**.

The following figure shows the configuration of web applications in IIS.
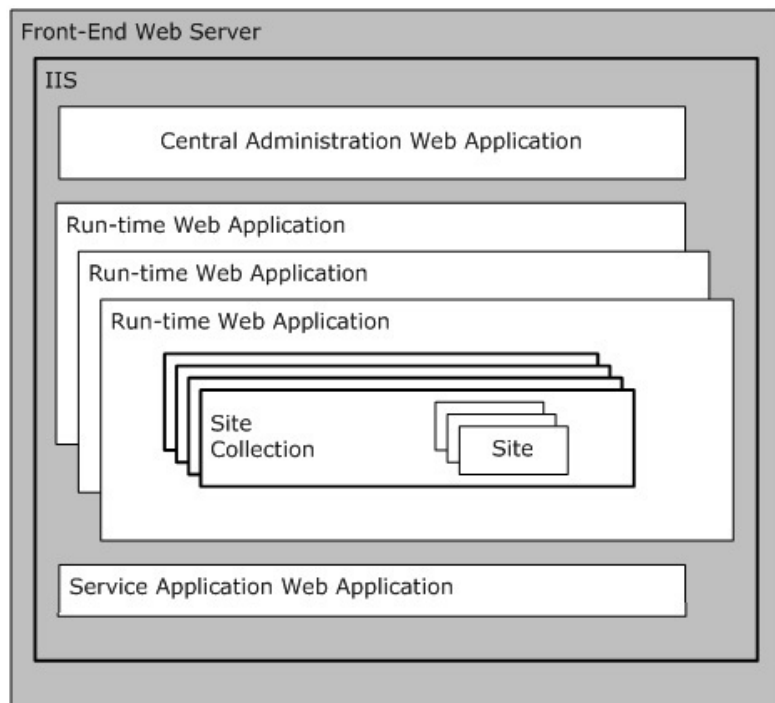
Figure 2: Configuration of web applications in IIS

In this configuration, the run-time web applications render all of the user content and contain one or more **site collections**. The architecture of SharePoint Products and Technologies allows for a large number of site collections in a single deployment. A single deployment can also support more than one run-time web application. Groups of site collections can thus be organized onto separate process boundaries, which in turn can be useful for supporting different modes of authentication and for separating content into different levels of security on the front-end Web server.

2.1.1.2.2 Service Applications

A **service application** exposes functionality that has been configured for providing extended capabilities across site collections and **web applications** or potentially across multiple farms running SharePoint Products and Technologies. A service application can be implemented as a **Web service**. Service applications that are implemented as Web services are configured and hosted by the service application web application, as described in the overview of Web topology (section 2.1.1.2.1).

For example, a full-text search can be used to aggregate content that exists both across many site collections and outside of SharePoint Products and Technologies. To enable such a search, the **search service application** defines the search capability, which each run-time web application then accesses individually.

Every **farm** running SharePoint Products and Technologies includes zero or more service applications. The service applications that are provided with SharePoint Products and Technologies include:

- Excel Services: Storing, calculating, and displaying spreadsheets, as described in section 2.1.3.1.

- Microsoft Access Services: Storing, creating, and managing database applications, as described in section 2.1.3.2.

- User profiles: Creating **user profiles** and **audiences**, as described in section 2.1.3.4.

- Search services: Full-text querying, as described in section 2.1.3.5.

- Indexing: Building content indexes for full-text querying, as described in section 2.1.3.5.

- **Business Connectivity Services (BCS)**: Querying data from external systems, as described in section 2.1.3.6.

Service applications use claims-based **identity** when communicating with each other, and they can use it when communicating with external systems. For more information about claims-based identity, and how claims-based identity is used with user **authentication**, see section 2.1.2.4.1. For more information about service applications see [MSDN-SharePointSDK].

2.1.1.2.3 Web Components

Because of their integration with IIS and ASP.NET, SharePoint Products and Technologies implements a full **Web server** that is capable of storing and delivering user content over HTTP and HTTP extensions. Web pages on SharePoint sites use the ASP.NET framework for rendering and executing Web content. Such Web pages include those that are installed by default as well as those that are customized. In addition, SharePoint Products and Technologies uses ASP.NET **Web Parts** for customizing and reusing Web content.

2.1.1.2.4 Web Parts

By allowing users to easily add and change information that appears on individual Web pages, ASP.NET Web Parts provide a powerful and easy-to-use mechanism for customizing the presentation of **websites**.

For example, a user might decide to change the layout of components on a **site's home page** by removing some content and changing the page's layout or appearance. With the user interface (UI) components displayed on the website, the user can customize the page by changing settings on the Web Parts and by adding and removing Web Parts. Web Parts can provide this level of flexibility, because they are represented as separate resources that are not stored on the base ASP.NET page. Instead, Web Parts are ASP.NET components for which the executable code is deployed on the front-end Web server. All the settings and customizations for a Web Part are stored in a **content database** and associated with the location of the particular page on the particular site. For information about the types of databases provided by SharePoint Products and Technologies, see section 2.1.1.3.1.

A detailed technical specification of Web Parts as they are used in SharePoint Products and Technologies is available. For more information, see [MS-WSSTS] section 2.1.2.15.2. Information about the development and utilization of Web Parts is also available. For more information, see [MSDN-WSSWP]. Finally, a general overview of ASP.NET Web Parts is available. For more information, see [MSDN-ASPWP].

2.1.1.3 Data Storage

The following sections provide a summary of the physical and logical storage that is provided by SharePoint Products and Technologies.

2.1.1.3.1 Physical Storage

All the information that is stored by SharePoint Products and Technologies is contained in relational databases in one or more instances of Microsoft SQL Server in the data storage layer. Individual front-end Web servers do not serve as repositories for user data. Instead, they store only the configuration

or application information that can be easily redeployed. **Application servers** can store some constructed data that is derived from user data, such as a set of full-text indexes. Such constructed data can take time to reconstruct. The data storage is intended to be completely abstracted from SharePoint site users and administrators. In other words, normal usage of the farm running SharePoint Products and Technologies and interoperability with the protocols does not require any direct interaction with the data storage environment.

At a high level, the following are the relational databases:

- **Content database:** Stores all user and **site** content.

- **Configuration database:** Stores topology and configuration information for the complete deployment. Every farm has exactly one configuration database.

- **Central Administration content database:** Stores content associated with the **Central Administration site**.

- **Service applications databases:** Stores content and configuration information for service applications, such as the **User Profile Service**. The capabilities of service applications are described in section 2.1.1.2.2.

SharePoint Products and Technologies front-end Web servers and application servers use T-SQL language [TSQL-Ref] with the Tabular Data Stream Protocol [MS-TDS] as the transport to communicate with the back-end database server.

2.1.1.3.2 Logical Storage

Logically, all the user content that is stored by SharePoint Products and Technologies is located within the container **hierarchy**. A simplified perspective is as follows:

- **Site collections:** A **site collection** serves as the top-level boundary of storage and security for SharePoint Products and Technologies. A site collection also serves as the unit of scale, which means that the architecture of SharePoint Products and Technologies allows for a large number of site collections. Each site collection can contain a hierarchy of **sites** and can directly contain **lists (1)** and **document libraries**.

- **Sites:** A site provides a mechanism for subdividing the URL namespace, security, and organization of content within a site collection. Each site can contain sites, lists (1), and document libraries.

- **Lists and libraries:** Lists (1) and libraries act as the basic containers for content that is stored in SharePoint Products and Technologies. In other words, users store and access individual pieces of content from lists (1) and libraries. A list (1) can contain **list items**, and a library can contain **documents**. A SharePoint library stores documents with their associated metadata. Client applications can interact with the documents in SharePoint libraries by using either **file** operation protocols (section 2.1.2.3) or authoring integration protocols (section 2.1.2.5). Both lists (1) and libraries can be organized further into a hierarchy of folders.

- **Folders:** Folders act as organizational tools for categorizing the **items** or documents that exist in a list (1) or library. Folders enable browsing and permissions management. A folder can contain subfolders as well as individual list items and documents.

- **List items and documents:** In SharePoint Products and Technologies, the basic units of content are the list item and the document. In a list (1), the unit of content is the list item, which can be an individual **task**, announcement, or contact entry. In a library, the unit of content is the document with its associated metadata.

Each instance of the types of content described in this section will have its own URL that can be used to directly access and link to the content. In addition, users can create lists (1) and customize them with new **columns**, **forms**, and **views**.

For more information, see [MS-WSSTS] section 2.1.2.

2.1.1.4 Protocol Areas

These protocols described in this document enable communication between SharePoint Products and Technologies farms and other client and server applications and intra-farm communication between web front-end servers and application server.

The protocols are implemented by using **HTTP** or extensions to HTTP, such as **SOAP Web services**. For more information, see [SOAP1.1], [SOAP1.2-1/2007], [SOAP1.2-2/2007], [RFC2616], and [RFC2818].

Because HTTP and SOAP are used, these protocols can be used by **web applications**, tunneled through corporate network firewalls, and accessed through load-balancing technologies.

The protocols are divided into two main groups: core platform and extended capabilities.

The core platform protocols cover the following areas:

- **Site** administration

- **List (1)** data

- **File** operations

- Security and **identity**

- Authoring integration

- Extensibility

- SharePoint client-side object model

The extended capabilities protocols cover the following areas:

- Excel Services

- Access Services

- InfoPath Forms Services

- **User Profile Service**

- Search services

- **BCS**

- Content Management Service

- PerformancePoint Services in Microsoft SharePoint Server 2010

- Translation Services

- Workflow Services

- Work Management Service

- Social Services

- PowerPoint Automation Services

- User Code Execution Services

- Word Automation Services

- Visio Graphics Services

Sections 2.1.2 and 2.1.3 summarize the core platform and extended capabilities groups of protocols.

2.1.2 Core Platform

The core platform provides administrative services, a security and identity model, and the basic physical and logical storage for content.

2.1.2.1 Site Administration

Users access data and capabilities by interacting with individual SharePoint sites through a Web browser or other client application. Sites (2) are organized hierarchically, so a site (2) can contain zero or more subsites. A **hierarchy** of sites (2) is contained in a **site collection**, and multiple site collections can be contained in a single **farm** running SharePoint Products and Technologies. Technical details about site collections and sites (2) are available. For more information, see [MS-WSSTS] section 2.1.2.

The client application, whether it is a Web browser or other client application, communicates with the farm by means of HTTP-based protocols, as described in section 2.1.1.2.3. Every site collection and every site (2) within a farm has its own URL, so access patterns frequently involve users connecting directly to a site (2) by means of the URL and then navigating or searching within the site's content hierarchy.

SharePoint Products and Technologies provides generic site (2) administration capabilities, such as the basic creation, configuration, and inspection of all types of sites (2) and site collections. SharePoint Products and Technologies also supports additional site (2) administration functionality for specific types of sites (2), such as **Document Workspace sites** and **Meeting Workspace sites**. These sites (2) are built from specific **site definitions** and target specific usage scenarios. Document Workspace sites facilitate the collaboration of multiple authors on a single **document** with supporting collateral. A Meeting Workspace site provides a **website** for organizing content that is associated with a meeting, such as agendas, participants, follow-up details, and meeting collateral.

Another site (2) administration capability entails linking the **membership** of a SharePoint site to an e-mail **distribution list**. If the deployment has been configured appropriately with a directory provider, a client application can use membership information to create and configure a distribution list in an external directory.

2.1.2.2 List Data

The **list (1)** data protocols enable core SharePoint list and library data access for creating and working with lists (1).

2.1.2.3 File Operations

The file operations protocols that are implemented by SharePoint Products and Technologies provide basic browse, open, edit, save, and rename file operations. These capabilities are exposed through clients such as Windows Internet Explorer and Microsoft Office 2013. At a high level, two primary technologies provide this interoperability: Microsoft FrontPage Server Extensions and **Web Distributed Authoring and Versioning Protocol (WebDAV)**. Earlier versions of Microsoft Office and Microsoft Windows implemented both FrontPage Server Extensions and WebDAV, whereas the newer versions of these clients are moving towards implementations that use only WebDAV. A description of how Office 2013 use these protocols to communicate with SharePoint Products and Technologies is available. For more information, see [MS-OCPROTO] section 2.1.2.

SharePoint Products and Technologies also supports the ability to extract document metadata and link information from specific document file formats. This is called document **property promotion**.

2.1.2.4 Security and Identity

Security for SharePoint Products and Technologies can be divided into two separate but related areas: **authentication** and **authorization**. Authentication is the process of reliably validating the **identity** claims of an individual user. Authorization is the process of correctly determining which permissions that a user is supposed to have within the farm running SharePoint Products and Technologies.

2.1.2.4.1 Authentication

For user **authentication**, SharePoint Products and Technologies relies on external authorities to appropriately validate a user's identity and, optionally, **group (1)** or role membership. By default, this external authority is Active Directory Domain Services (AD DS), but SharePoint Products and Technologies supports pluggable security authentication that allows any appropriately configured authentication provider to validate a user's identity. Such a provider, which is called an authentication system, provides a list of users and group (1) memberships that are available to a web application (1) in SharePoint Products and Technologies. A technical specification of an authentication system is available. For more information, see [MS-WSSTS] section 2.1.4.

Microsoft SharePoint Server 2013 and later also implement claims-based identity, an authentication system based on different industry standards as described in [MS-SPSTWS] section 1. This capability exists in addition to Windows Challenge/Response (NTLM) and the Kerberos protocol. The implementation of claims-based identity is standards based. Users who deploy SharePoint Server 2013 and later can use any protocol or service that implements these standards to provide identity to SharePoint Products and Technologies. In addition, when applications running on SharePoint Server 2013 and later need to make external Web service calls, Microsoft SharePoint Server will issue a **Security Assertion Markup Language (SAML)** token that identifies the user and application pool identity. For more information, see [WSTrust], [WSFederation], and [SAMLToken1.1].

2.1.2.4.2 Authorization

Authentication system, **group (1)**, and user identities are known as security **principals**, which are granted permissions on specific objects in SharePoint Products and Technologies. Permissions can be set directly on standard types of content, such as **site collections**, **sites**, **lists (1)**, libraries, folders, and items. In addition, permissions can be inherited from containers by the objects that exist in those containers. To simplify the administrative overhead that is associated with assigning permissions, SharePoint Products and Technologies implements the concept of groups (1), allowing a collection of principals to be assigned the same permissions and roles. A description of how authorization is implemented is available. A technical description of permissions is also available. For more information, see [MS-WSSTS] section 2.1.4.

2.1.2.5 Authoring Integration

In addition to basic file operations, SharePoint Products and Technologies implements several protocols that can be used by authoring applications, such as Office 2013 and later to provide specific, enhanced experiences for content creation. For example, SharePoint Products and Technologies implements protocols that facilitate scenarios in which a client application produces content that is targeted for the **server**. Such content might include a message for a **blog** or slides for a **Slide Library**.

SharePoint Server 2013 and later also implement protocols to help synchronize document content between client applications. These protocols allow a client application to incrementally upload or download changes to documents along with any metadata associated with the changes from SharePoint libraries. These protocols enable coauthoring features in Microsoft Word 2013 and

Microsoft PowerPoint 2013 and later as well as document synchronization in Microsoft SharePoint Workspace 2010.

For more information, see [MS-OCPROTO] sections 2.1.10, 2.1.12, and 2.1.18.

2.1.2.6 Extensibility

SharePoint Products and Technologies implements support for content import and export, **workflow (1)** integration, and Web page and **Web Part** customization. For information about workflow (1) in SharePoint Products and Technologies, see [MSDN-MOSSWF] and [MSDN-WSSWF].

2.1.2.7 SharePoint Client-Side Object Model

SharePoint Server 2013 and Microsoft SharePoint Foundation 2013 and later implement a client-side object model. The SharePoint Client-Side Object Model enables a client application to make batch requests to a front-end Web server running SharePoint Products and Technologies to perform operations on core platform objects such as **lists (1)**, **sites**, and users. The operations can include method calls, property read operations, and property write operations.

Through the SharePoint Client-Side Object Model, a client application has access to much of the same functionality that is exposed by the SOAP Web services mentioned in section 2.1.2. The SharePoint Client-Side Object Model differs from the SOAP Web services in that it is optimized to support the placement of a batch of operations into a single request by the client application. The operations can also span object types, whereas the SOAP Web services are organized by individual object type. For more information, see [MS-CSOMSPT] section 3.2.1.

The SharePoint Client-Side Object Model can also be accessed using common web technologies via a REST methodology.

2.1.2.8 Configuration Services

SharePoint Products and Technologies implements the protocols described in the table in section 2.2.1.8 to enable clients or services to discover details of features available in a SharePoint Products and Technologies farm, such as **service applications** enabled on the farm; share configuration data and management of groups of **site collections**; or discover database connection information for databases used by applications deployed in the SharePoint Products and Technologies farm.

2.1.3 Extended Capabilities

The extended capabilities build on the core platform that was described in the preceding sections.

2.1.3.1 Excel Services

Excel Services is composed of three main components: Microsoft Excel Calculation Services (ECS), Excel Web Access (EWA), and Excel Web Services.

The ECS component loads and calculates workbooks, applies user changes to workbooks, accesses external data that the workbooks might reference, and saves workbooks back to the SharePoint Products and Technologies content database. ECS can be deployed on a separate application server in the farm.

EWA interacts with ECS and renders the **HTML** representation of workbooks. Using EWA, a user can view and interact with a workbook. Excel Web Services provides programmatic access to Excel Services capabilities. Using Excel Web Services, a user can load, modify values in, calculate, and retrieve the calculated results of a workbook. ECS also allows more than one user to edit the same workbook at the same time.

The following figure shows the high-level architecture of Excel Services and how it can be deployed in the farm topology. Though ECS can be deployed on a separate **server** as shown in this diagram and as described in section 2.1.1.1, it can also be configured to run on the same server with EWA and Excel Web Services.

For technical information about the protocols implemented by Excel Services, see [MS-ESP], [MS-ESURL], and [MS-ESREST]. For technical information about Excel Calculation Web Services XML Specification, see [MS-EXSPXML3]. For technical information about the protocol implemented by Excel Calculation Web Services, see [MS-EXSPWS3].
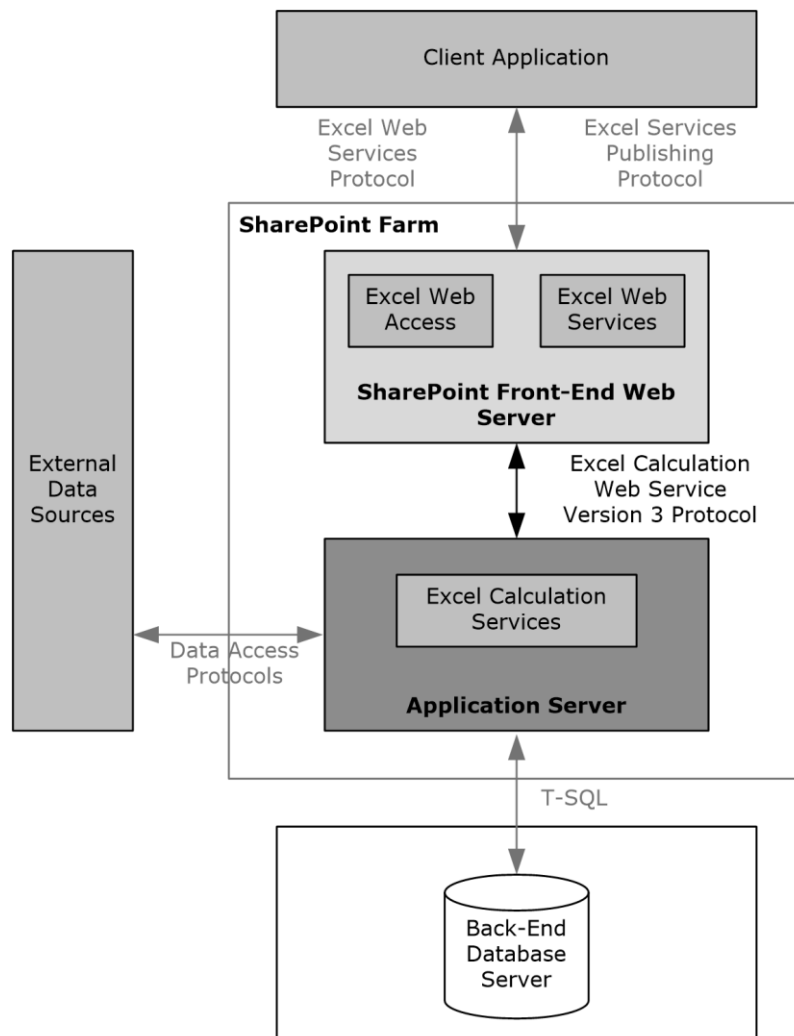
Figure 3: High-level architecture of Excel Services components

Excel Services is used to create, update, format, **calculate**, and save **workbooks** on a server. Excel Services provides the following core capabilities:

- The ability to **publish** workbooks from a client application, such as Microsoft Excel 2013, to the server. Users can then view and interact with these workbooks via a Web browser. This ability enables browser-based interaction with the workbooks from machines that do not have a spreadsheet application installed.

- The ability to programmatically access workbook contents via Excel Web Services, which is a **Web services** protocol. This ability enables client applications to load, edit, interact with, calculate, and save workbooks that have been **published** to SharePoint Server 2013.

Excel Calculation Services provides the core calculation and workbook processing for Excel Services. A description of the architecture of this application server and the protocol communication with it is available. For more information, see [MSDN-EXCELS].

As in the usage scenario for viewing workbooks in the browser, it is possible to programmatically access and manipulate server-based workbooks by using Web services. Operations available via the Excel Web Services API include opening, setting ranges in, refreshing **external data** in, recalculating, getting ranges from, and saving a workbook.

Workbooks that are loaded by Excel Calculation Services can reference external data. The data can be in relational or multidimensional databases. Similar to a client spreadsheet application, Excel Calculation Services uses the following data access APIs to retrieve information from such data sources:

- **Open Database Connectivity (ODBC)**

- **OLE DB**

- OLE DB for **Online Analytical Processing (OLAP)**

If a workbook contains elements that reference external data, such as PivotTable or data access formulas, Excel Calculation Services will perform the data retrieval by using one of the preceding methods.

A common way to extend these features to new data sources is to install an appropriate ODBC driver for the data source, thereby making it available to Excel Services. ODBC is a data access API that is commonly used. ODBC is not a wire protocol. Many ODBC drivers already exist for different data sources. When communicating with SQL Server data sources, the OLE DB providers included with the Microsoft .NET Framework are used. These providers use the Tabular Data Stream (TDS) protocol to communicate with the server running SQL Server. General descriptions of OLE DB and ODBC are available. For more information, see [MSDN-OLEDB] and [MSDN-OpenDBConnectivity].

Excel Calculation Services can communicate not only with strictly relational data sources but with multidimensional databases. For this purpose, SharePoint Products and Technologies can install an OLAP driver that communicates with SQL Server by using the SQL Server Analysis Services Protocol. For more information, see [MS-SSAS].

Two types of protocols are used for interoperability with Excel Services. In addition to the Excel Services Publishing Protocol, as described in [MS-ESURL], workbook publishing from client authoring applications to Excel Services also uses the file operations protocols as described in section 2.2.1.3.

2.1.3.2 Access Services

SharePoint Server implements Access Services, which is used to create **database applications** and manage data. Access Services provides the following core capabilities:

- The ability to **publish** database applications from a client application, such as Microsoft Access 2013, to the **server**. Users can then view and edit data in these databases via a Web browser. This ability enables browser-based viewing and interaction with the databases from machines that do not have a database application installed.

- The ability to programmatically access database contents via a Web services protocol. This ability enables client applications to retrieve and interact with databases that have been **published** to SharePoint Server 2013.

Access Services, when used with an Access web application, is composed of two main components: Access Services Data Server (described in [MS-ADS]), which retrieves and manages data from SharePoint Products and Technologies storage for the database applications, and Access Services Web Services which provides the Web-based rendering for Access Services database applications. The following figure illustrates this architecture.
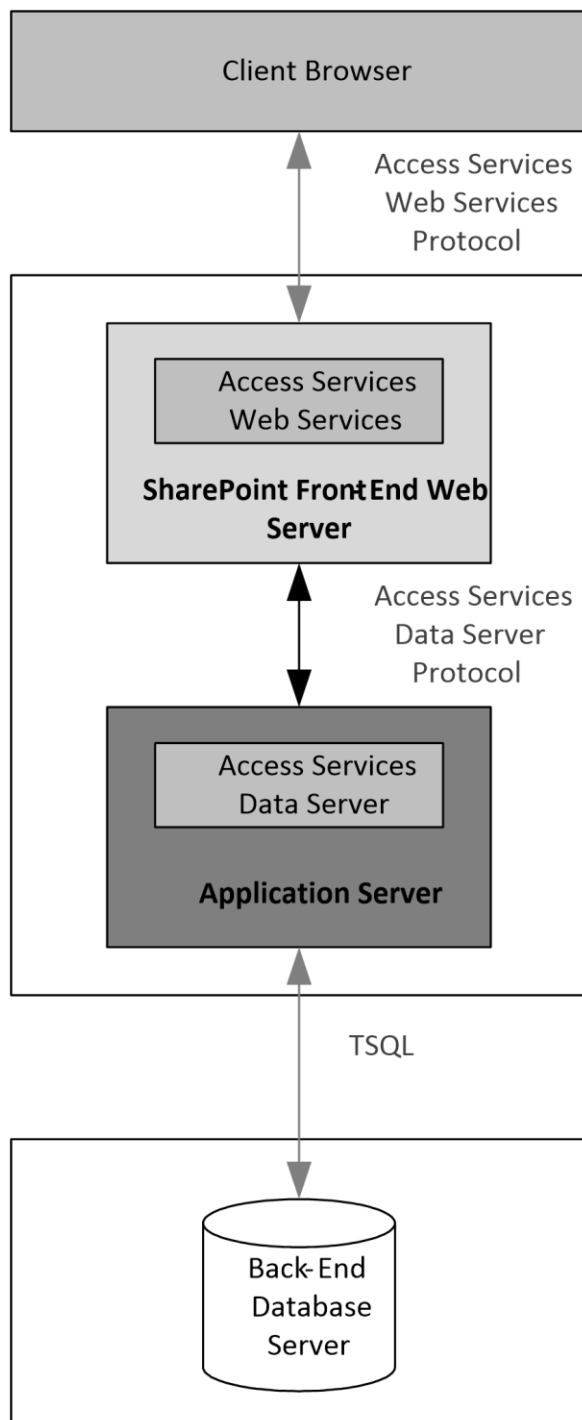
Figure 4: High-level architecture of Access Services components for an Access web app

When used with a Microsoft Access Services web app, Access Services is composed of different components. The Access Services Data Run Time (described in [MS-ADR]) is the protocol that retrieves and manages data from a SQL Server back-end databases. Access Services Web Services provide the client-side rendering for database applications. The following figure illustrates this architecture.
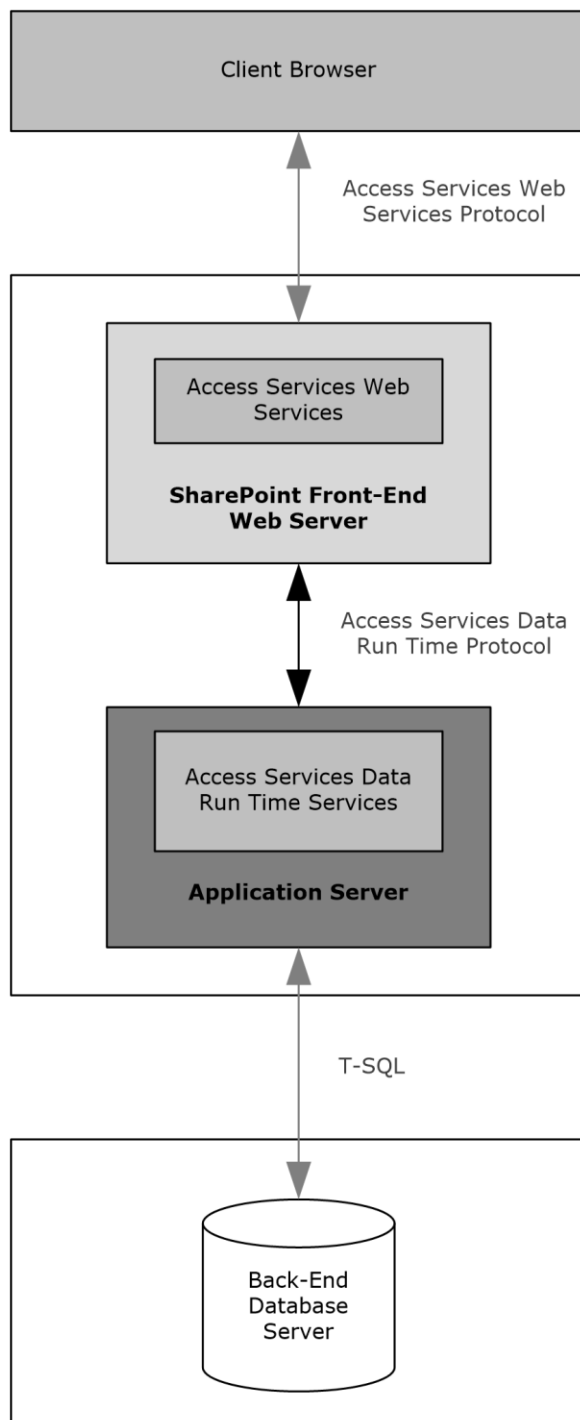
Figure 5: High-level architecture of Access Services components for a Microsoft Access Services Preview web app

A user can publish two types of Access Services database applications to SharePoint Server 2013 from a client application, such as Access 2013: Access web application and Access Services.

For Access web application, Access Services creates a **site** for each published database application. Access 2013 creates a **list (1)** for each table in the database and uploads the data from the database to the lists (1). Access Services Data Server retrieves and manages data from the storage in SharePoint Products and Technologies for the database applications. For more information, see [MS-ADS]. A user can also get application reference files for the Access 2013 database application. The application reference files connect to the lists (1), **forms**, and reports in the database application that was published and allows the user to edit them in Access 2013. For more information, see [MS-AXL] section 1.3.7. These applications use Access Services Web Services to communicate with the server. For more information, see [MS-ASWS].

For Access Services, Access Services creates a site for each published database application and a database in SQL Server. These applications use the Access Application Design Time Protocol and the Access Server Design Time Protocol to communicate with the server. For more information, see [MS-AADT] and [MS-ASDT]. These applications use the Access Run Time Protocol and the Access Services Data Run Time Protocol to send and receive data to and from the server. For more information, see [MS-ART] and [MS-ADR].

2.1.3.3 InfoPath Forms Services

InfoPath Forms Services in SharePoint Server 2013 enables users to fill out **form templates** in a Web browser. This ability requires that InfoPath Forms Services be enabled and that the **browser-enabled form template** be published to the **server**. For more information, see [MS-OCPROTO] section 2.2.3.

SharePoint Server 2013 also implements the ability of form-editing applications, such as Microsoft InfoPath 2013, to store **forms** as content in document libraries. This ability relies on the file operations protocols as described in section 2.2.1.3.

2.1.3.4 User Profile Service

SharePoint Products and Technologies implements a **User Profile Service** that stores information about a user population. This information might include names, e-mail addresses, and telephone numbers. The User Profile Service aggregates information from multiple data sources and provides a uniform interface to this information at both the user level and the programming level. The User Profile Service exposes a number of capabilities including consistent ways to query across the user data, categorize users into **groups (1)**, create indexes of user data for full-text searches, access a consistent **change log** of all the modified user data, and synchronize the user data with multiple SharePoint sites in a **farm**.
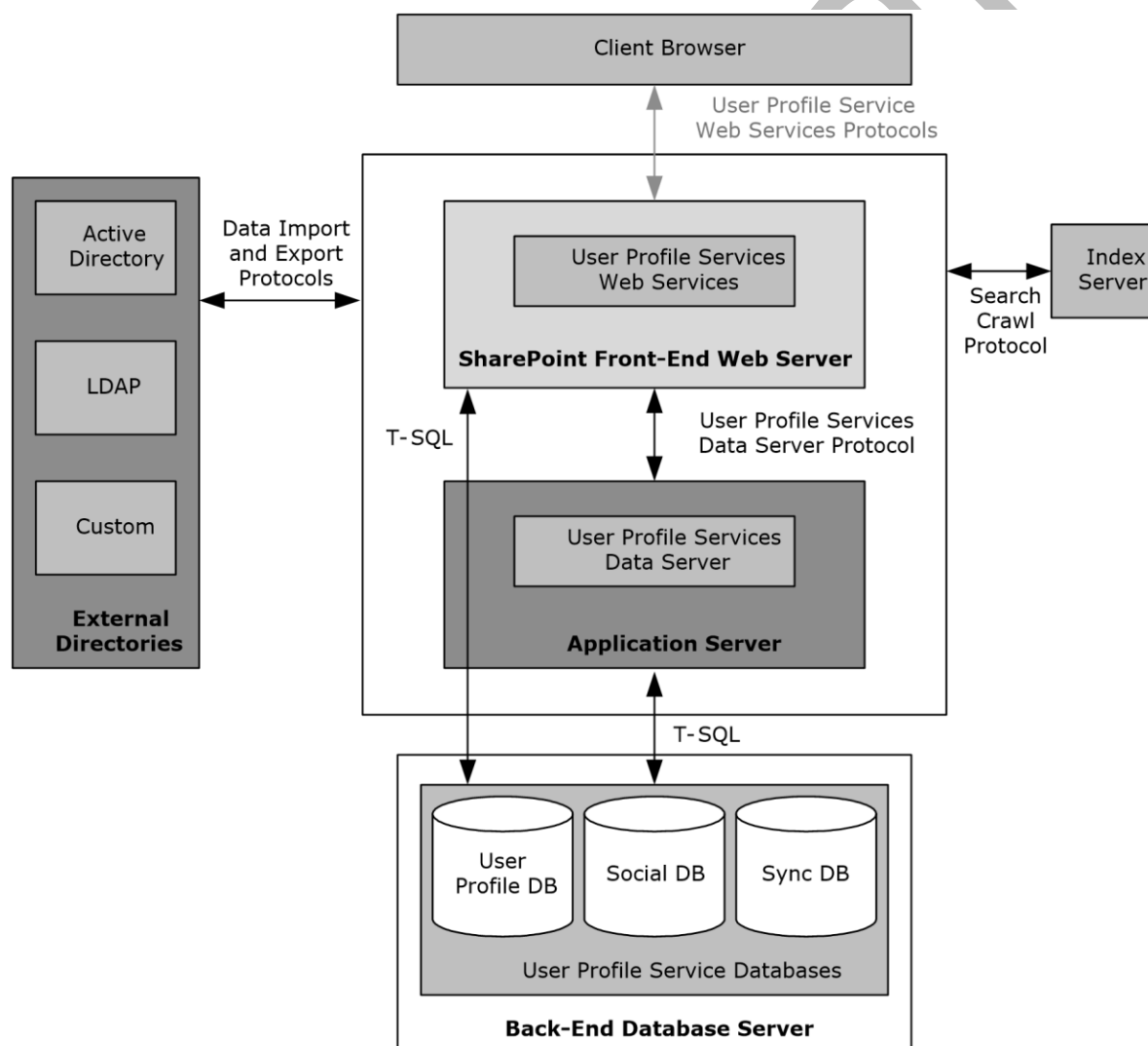
The User Profile Service is implemented as a **service application**, as described in section 2.1.1.2.2. Multiple **site collections** can thus refer to one central User Profile Service for information about users.

A profile is intended to be a description of a person. In common usage scenarios, each profile represents one person. In some situations, it represents one identity (for example, a user account). The profile consists of a set of identifiers (for example, a user name, an e-mail address, and an alias) and a **list (1)** of values for additional properties. SharePoint Products and Technologies also implements support for properties that contain multiple values, such as a property that contains a list (1) of skills for a person, and for properties that are linked to specific, restricted lists (1) of allowed values. Access to properties can be restricted so that only a specific group (1) of related users can see a particular value. The User Profile Service in SharePoint Products and Technologies aggregates the information from multiple data sources, within SharePoint Products and Technologies **sites** and externally as described and [MS-UPIEWS], into a single repository, and provides a uniform user experience for this data. In addition, the User Profile Service provides programmatic access to this data as described in [MS-UPSCWS].

In addition to the core concept of a profile, SharePoint Products and Technologies implements support for grouping profiles together. One mechanism for doing so is to import key grouping information from other sources, such as e-mail **distribution lists** or **site memberships**. SharePoint Products and Technologies also allow the calculation of collections of user profiles based on profile metadata information. Such a collection is called an **audience**. For example, all the managers of a company can comprise an audience. Web pages or client applications can then target information to this audience.

The User Profile Service also implements support for adding **social data** to any **Uniform Resource Identifier (URI)** and for retrieving social activity **events (1)** information, such as service anniversaries, distribution list memberships, and social data activity on a URI. The URI can point to a resource or to content inside or outside of SharePoint Products and Technologies. Social data can be in the form of numerical ratings, a note that is represented by lines of text, and keywords or **terms**. Keywords or terms are managed by the Content Management Service as described in section 2.1.3.7. Common usage scenarios for this metadata include rating a document, tagging content with something familiar to the user (for example, "social computing" for a document that is related to social computing), and writing a descriptive line of text as a note about some content.

The following figure shows a high-level view of the User Profile Service protocol architecture.

Figure 6: High-level view of the User Profile Service protocol architecture

2.1.3.4.1 Core Service

The core User Profile Service describes the set of protocols that enable an implementer to access and modify the data and metadata stored in the service. The User Profile Service Web Service (described in [MS-USPSWS]) allows data operations such as creating, reading, updating, or deleting user profiles. In addition, user profiles can be searched by identifier such as name, e-mail address, or alias.

The User Profile Service databases store and manage user information.

When accessing data from the user profile database, the use of the User Profile Service Web Service Protocol [MS-USPSWS] is recommended.

The key pieces of data stored in the User Profile Service database are the following:

- Properties

- Organizations

- Profiles

- Multiple values

- Term Sets (Vocabularies)

- Colleagues

- Memberships

- Profile change log

- Social data

2.1.3.4.1.1   Properties

The User Profile Service database schema defines what the profile object contains. The database schema is a list of properties, their data types, and optionally, valid property values. System properties are used by SharePoint Products and Technologies and cannot be deleted. Non-system (normal or "regular") properties can be deleted. Administrators and implementers can create only non-system properties; however, there are no fixed limits on the number of properties that can be created for a profile.

2.1.3.4.1.2   Organizations

Organizations can be created to represent an entity, such as a business division or a working group containing members, delegates, contact information, and additional properties defined in the schema. Organizations can have a hierarchical relation with other organizations. This hierarchical relationship can then be used for actions such as searching for specific users related to the organization or its child organizations.

2.1.3.4.1.3   Profiles

A profile is a record for one user, and in some situations represents one identity (for example, a user account). The profile consists of a set of identifiers (such as user name, e-mail address, and alias) and a list of values for additional properties that are defined in the schema. Organizations can also have profiles, with identifiers such as the organization name, owner, delegates, and a **list (1)** of values for additional properties that are defined in the schema.

2.1.3.4.1.4   Multiple Values

Many scenarios require more than one value for a given property, for example, the **list (1)** of a user's skills. The User Profile Service Web service enables the storage of multiple, individually addressable property values. This functionality is typically used to **query** all profiles that contain a specific value for a property (for example, find all users who have a "Skill" called "programming").

2.1.3.4.1.5   Vocabularies

Term sets allow the user to select an existing value, or if it does not exist, to enter a value specific to the user. This value is then made available for all future selections. For example, an **Interests** profile property might allow the user to select from a term set of interests that others have used to describe themselves or to enter a new interest, for example, "skiing". When subsequent users update their profile, the vocabulary contains the word "skiing".

Term sets that are stored and managed by the term store defined in the Content Management Service section 2.1.3.7.This allows a term set administrator to define centrally controlled terms with the full benefit of other functionality implemented by the Content Management Service. Such centrally controlled term sets are closed vocabularies for users to use or select.

2.1.3.4.1.6   Colleagues

The profile service stores relationships between users, as well as metadata about users. The **list (1)** of other users that the user knows is named "colleagues".

2.1.3.4.1.7   Memberships

The profile service also stores and models a user's affiliations with groups or organizations. The service models two types of groups: SharePoint sites and **distribution lists**. The creation of distribution lists and security group memberships is accomplished by using the User Profile Import/Export Web Service Protocol [MS-UPIEWS].

2.1.3.4.1.8   Profile Change Log

As with any database store, it is important to identify and access the information that has changed. The uses for this information are many. For example, when synchronizing two profile stores, the changes can be applied from one database to another.

The change log is a chronological list of changes, and each change has a time stamp. The change log service provided by the User Profile Change Log Web Service Protocol ([MS-USRPCH]) returns the changes that occurred in the database after a given time stamp or social activity.

Any property or membership change triggers the creation of an entry in the change log.

2.1.3.4.1.9   Social Data

The User Profile Service facilitates a user applying social data on a **URI**. A user can select, for example, a document or an arbitrary URL and attach additional social metadata to the URL. Some common scenarios for using metadata include the ability to rate, comment, bookmark, or add keywords to content.

SharePoint Server 2013 and later adds support for front-end web servers saving social data posts and feed entries, and, generating consolidated feeds from an in-memory cache maintained on the Profile Service **servers**. The protocols used for that cache are described by [MS-UPFCWS] section 1.

2.1.3.4.2 Data Import and Export

The profile service described in section 2.1.3.4.1 is useful only when there is a large body of data on which applications and features are built. Enterprise systems already contain a significant amount of information about users, whether in enterprise **line-of-business (LOB) systems**, directories, or

other databases. The profile service imports data from most available sources by using the protocol defined in [MS-UPIEWS]. Data is imported into a synchronization database and then into the profile store, by using the protocols described in [MS-UPSCDS], [MS-UPSLDAP], [MS-UPSSCXPP], [MS-UPSMODS], and [MS-UPSCP].

The profile service can import data from three sources:

- An **Active Directory** directory

- A **Lightweight Directory Access Protocol (LDAP)** v3 directory

- The SharePoint Products and Technologies **Business Data Connectivity (BDC)**

Once the data is available in the user profile store, the profile service can export the data to the SharePoint Products and Technologies content database. The User Profile Service can also export data back to Active Directory and Lightweight Directory Access Protocol (LDAP) directories. The protocol used for import and export is defined in [MS-UPIEWS]. The SharePoint Products and Technologies BDC remains import only.

2.1.3.4.2.1   Active Directory

AD DS can provide a rich store of user information, as described in [MSFT-ADC]. It contains identifying information, descriptive metadata (such as office number, title, and phone number), and relationship information (such as manager and groups a user belongs to). The profile service can import that information into the database. For each directory, the administrator creates a directory connection and describes which directory to connect to, what **credentials** to use, and what subset of the data to import.

A User Profile Service administrator can define properties that will be exported back to AD DS. A popular scenario for the export functionality is to send a user's picture back to be stored in AD DS.

2.1.3.4.2.2   LDAP v3 Directory

SharePoint Products and Technologies also supports directories that implement the LDAP v3 standard described in [RFC4510]. Importing from LDAP v3 works in much the same way as importing from AD DS. However, **groups (1)** are not imported from an LDAP v3 directory.

A User Profile Service administrator can define properties that will be exported back to LDAP. A popular scenario for the export functionality is to send a user's picture back to be stored in LDAP.

2.1.3.4.2.3   Business Data Connectivity (BDC)

SharePoint Products and Technologies supports importing user data from external systems that are configured by using the **Business Data Connectivity (BDC)**. The Business Data Connectivity service is part of **Business Connectivity Services (BCS)**. For information about Business Connectivity Services, see section 2.1.3.6.

This action does not specify any particular wire protocol, because BDC connects to whatever protocol is provided by the back-end system. The BDC information provides a consistent interface to Web services and databases through a declarative XML syntax. The User Profile Service uses BDC to import data from the data sources that BDC supports. For example, the human resources (HR) database contains the employee's hire date, which is not **published** to the company directory. The profile service can create a profile for the employee from the directory entry and complete the employee's **hiredate** property by importing that value from the HR database.

One major limitation to importing from BDC is that profiles cannot be created or removed by an import. The BDC import only comments existing profile records.

2.1.3.4.3 SharePoint Products and Technologies Content Database

SharePoint Products and Technologies stores user information in every **site collection** for various uses such as security, name display, and presence. The user record in each site collection is created by copying information from the directory; however, the record is not updated automatically. SharePoint Products and Technologies updates the records, on a schedule that the administrator sets, by tracking user information that has changed in the User Profile Service database since the last time it synchronized data.

SharePoint Products and Technologies scans every content database for new user records and updates them with the latest information from the User Profile Service database. It also locates the records for users whose information has changed in the User Profile Service database and updates those as well.

2.1.3.4.4 User Profile Service Database Indexing

SharePoint Products and Technologies implements full-text search of information stored in the user profile database. For example, users can find the profiles of other users by searching on interesting keywords related to expertise, name, or any other information stored in the User Profile Service. This capability is implemented by using the search infrastructure that is described in the various protocol specifications for the **search service application**, which requires that the User Profile Service database is first indexed. This indexing is done using the protocols defined in [MS-SPSCRWL].

2.1.3.4.5 Personalization and Targeting

It is possible to use the information in the User Profile Service database for personalization and content targeting. For example, a company's employee **portal site** can be customized to show something specific to the job role of the employee to people in different job areas of the company for example, rather than showing the same content to everyone, a person who works in "Sales" will see something different about a given employee than what someone who works in "Development" might see.

SharePoint Products and Technologies addresses the challenge of partitioning people into specific groups through the use of **audiences**. Audiences define groups of people by their characteristics. An audience can be referred to as a dynamic group that is implemented through the use of rules, which are periodically evaluated in batch through a process called audience compilation.

2.1.3.4.6 Social Metadata

The User Profile Service facilitates a user applying social data on a **URI**.

A user can select, for example, a document or an arbitrary URL and attach additional social metadata to the URL.

Some common scenarios for using metadata include the ability to:

- "Rate" a document, content, or person.

- Add a note or comment to content.

- Bookmark content with some predefined keywords, such as "a favorite".

- Add tags to the content to categorize the content or highlight it with colleagues.

The Web services described in [MS-UPSDWS] social metadata to the corresponding URI. In the case of bookmarks and tags, the User Profile Service works with the Content Management Service (section 2.1.3.7) to store the value of the tag; the User Profile Service database only stores the corresponding **GUID**.

For all forms of metadata, four key pieces of information are stored in the User Profile Service database:

1. Who added the metadata: the record identifier of the user in the profile store.

2. When was it added: a time stamp.

3. Where it was applied: the URI.

4. Which social metadata it is: for rating, it is a number ranging from 1-5; for note, it is the text of the note; and for tags and bookmarks, it is the GUID returned from the Content Management Service.

2.1.3.5 Search Services

Search services provide the ability for a user to quickly see a list of the most relevant documents in a large document collection, based on a specified keyword query. A large document collection can have tens of millions of documents. This quick query resolution requires the creation and maintenance of the **inverted index** in SharePoint Server 2013. SharePoint Products and Technologies implements the protocols that are necessary for creating the indexes, including those for crawling (traversing the URL space of the documents that the user wishes to search and synchronizing the changes in the URL space), retrieving the documents, parsing the documents, performing tokenization, and creating the inverted index. SharePoint Products and Technologies also implements the front-end protocols that are necessary for executing queries against these data structures and for administering the search configuration.

Both SharePoint Foundation 2013 and SharePoint Server 2013 and later implement search protocols. In SharePoint Foundation 2013, the protocols are limited to the subset that is necessary for crawling and providing a search capability across the content that is stored in SharePoint Foundation 2013. The SharePoint Products and Technologies search service provides search capabilities across the content in all the SharePoint Foundation 2013 **sites** in the enterprise as well as across other **websites**, file shares, and other **content sources**. Unless otherwise specified, the concepts, configuration, and protocols apply to search services in SharePoint Foundation 2013 and SharePoint Server 2013 and later.

The following is a description of the architecture and components of the subsystems in the search functionality in the SharePoint Products and Technologies search services, along with the protocols involved in interoperability between the components.

2.1.3.5.1 Concepts

Search services use two basic operating concepts: crawls and queries. Crawls involve retrieving the content of the **websites**, profile repositories and other content sources by the search service, processing the content, including parsing of the documents, extracting metadata, linguistic processing (such as tokenization and stemming) and generation of the inverted index data structures that are subsequently used for fast querying. The processing steps are for illustration of a typical search engine and exact details are specific to the search service implementation. Queries involve the end user or a client application issuing queries typically containing one or more keywords, as well as other restrictions specifying the set of documents to be retrieved.

2.1.3.5.2 Crawling

The crawl operation is a background process that involves traversal of the URL space configured by the user. The crawl operation is implemented by the search service. If the URL space is considered a graph, where each node is a URL and the edges are created from **hyperlinks** (in the case of HTML) or other types of links between URLs, the basic crawl operation procedure consists of an exhaustive graph traversal.

Processing of a URL during the crawl involves retrieving the document data from the document repository specified by the URL, parsing the document, and determining the set of links to follow.

Subsequent crawl operations do not have to reindex every document in the URL space. These incremental crawls can be of two types: repositories that can provide detailed information about the

list of changes that need to be indexed and repositories that cannot provide this detailed information. SharePoint Products and Technologies includes the former. This type of repository monitors which documents within the repository change, and thus can provide change logs. A change log is the list of URLs of changed and deleted documents to reindex. For all other repository types, the set of URLs in the crawl history is added to the crawl queue and a crawl operation is initiated. By comparing the time stamp of each document that is crawled to its time stamp in the crawl history, the system can selectively reindex only those documents that have changed or have been deleted.

2.1.3.5.3 Querying

A query against the search service can be executed using [MS-QSSWS], [MS-SRCHCSOM] or [MS-SEARCH] protocols. The query language can be similar to a user typing the query, as described in [MS-SEARCH] section 2.2.10, or advanced query syntax (FQL) as described in [MS-FQL2] section 2.

2.1.3.6 Business Connectivity Services

Business Connectivity Services (BCS) provides access to data from server applications that are external to the farm running SharePoint Products and Technologies, thus bridging the gap between SharePoint sites and business applications. Through Business Connectivity Services, SharePoint Server 2013 and later have built-in support for displaying data from databases and Web services. Business Connectivity Services also facilitates the integration of data from line-of-business applications, such as SAP, Siebel, and other databases, into **lists (2)**, Web Parts, search services, user profiles, and custom applications. However, Business Connectivity Services does not specify a protocol for communicating with each line-of-business application. The SharePoint Server 2013 and later implementation of Business Connectivity Services supports the ability to write back to these data sources, as well.

BCS consists of two parts: a **Business Data Connectivity (BDC) metadata store** and an object model. The BDC service uses metadata to describe the APIs of the line-of-business applications. After the API description of a line-of-business application is registered in the metadata store, it provides dynamic data-access connections to the back-end data without the need for additional code.

The metadata model abstracts the underlying physical sources and provides a consistent and uniform way to work with different line-of-business applications. The metadata for each data source defines the programming objects that applications interact with and the methods that are available for client applications. Although the metadata model provides a consistent way to work with different data sources, BCS does not specify a generic protocol for communicating with each source.

For more information, see [MSDN-SharePoint2013-SDK].

Note that SharePoint Server 2013 and later also implements a capability for securely storing user credentials for authenticating to a line-of-business application. This capability is called the **Secure Store Service (SSS)**.

2.1.3.7 Content Management Service

SharePoint Server 2013 and later implement enterprise content management capabilities on top of the core storage, platform, and file operations infrastructure that is described in this document. These capabilities include document management and versioning, metadata management, tools for propagating and managing content across multiple installations, records management tools, and support for converting documents from one format to another.

2.1.3.8 PerformancePoint Services

PerformancePoint Services 2010 allows users to author and **publish** business intelligence data to a Web Part page in SharePoint Products and Technologies. This page, also known as a **dashboard** in PerformancePoint Services 2010, provides users with the ability to analyze their business data based on business rules and metrics. Dashboards can be filled with content representing business

performance indicators (such as grids, charts, **scorecards**, and other reports). This content is composed of metadata (such as the names of dimensions, measures, or data source references), calculation definitions, and formatting information (such as colors or layout information).

The Web services for PerformancePoint Services 2010 enable a protocol client to:

- Create, read, update, validate, and delete a variety of business logic content in a content store in SharePoint Products and Technologies.

- Create, read, update, and delete annotations that exist on a scorecard.

- Retrieve metadata, such as the names of dimensions or measures, and data sets via data source references.

- Retrieve a **list (2)** of content subtypes and component information from a configuration store.

A typical scenario for using these services involves creating an extensible content-editing application that enables users to create, publish, and maintain some of the content that is used in a larger business-intelligence system. A manager for a large business, for example, might wish to create a chart that shows the daily sales volume for a specific product. The chart would contain active areas that would allow the manager to click those areas for additional information, such as the breakdown of the net profit and cost for that product.

2.1.3.9 Translation Services

SharePoint Server 2013 and later implement Translation Services, which is used to automatically machine translate SharePoint content. Translation Services allows documents in SharePoint document libraries to be translated in one of two ways:

- With queued **translation jobs**, large numbers of documents can be submitted to a queue which is processed over time. Protocol clients can add single files, folders, or entire document libraries to a job. The translated output files can be saved to either the same document library or a different library for storing localized content. The protocol client can query the status of a previously submitted job.

- **Immediate translation jobs** are used for smaller, higher priority jobs, such as when a user wishes to translate a single document interactively. Immediate jobs bypass the queue and are processed as fast as possible. As with queued translation jobs, output files can be automatically saved to any document library.

2.1.3.10   Office Online Web Query Service

The Office Online Web Query Web Service enumerates details about files in the Office Online repository. Microsoft SharePoint 2010 Products and Technologies has a user interface for displaying **solution packages** stored in the Office Online repository, downloading a solution package, and storing it in a **site collection**'s **solution gallery**.

2.1.3.11   Workflow Services

Workflow Services manages workflows on SharePoint content. Workflow Services enables remote clients to manage **workflow associations**, to send **workflow events**, to control or to query **workflow instances**, and to manage workflow (2) definitions and actions.

2.1.3.12   Work Management Service

SharePoint Server 2013 and later implement work management capabilities on top of the core storage, platform, and file operations infrastructure that is described in this document. These capabilities include accessing and manipulating **tasks** assigned to a user.

2.1.3.13   Social Services

The SharePoint Social Client-Side Object Model Protocol provides types, methods, and properties to enable a protocol client to access social data for the current user. The set of types, properties and methods provide the ability for remote clients to send and retrieve data from **site** social features. Social data includes:

- User profiles

- Blogging

- Activity Feeds

- Following people and content

- Sharing documents

2.1.3.14   PowerPoint Automation Services

PowerPoint Automation Services provides the capability to convert a **presentation** from one file format to another. A typical scenario for using PowerPoint Automation Services is a file conversion application that enables users to convert presentation files to a different format for archiving purposes.

PowerPoint Automation Services is composed of the PowerPoint Automation Services Conversion Web Service Protocol [MS-PASCWS], which allows a protocol client to pass a presentation file to the protocol server and to receive from the protocol server a converted file in a different format.

2.1.3.15   User Code Execution Service

SharePoint Products and Technologies implements a capability to securely host custom solutions and executable code that comes from customers of the system. It provides better controls for the security of execution of custom code, as well as mechanisms to monitor and control the usage of resources by that code.

One requirement of these mechanisms is to move the execution of the code to separate processes and separate **servers**, to provide better layers of isolation and monitoring. Therefore, two roles are defined in a user-code execution process: a front-end Web server that initiates requests to run user code on behalf of a user, and a code execution back-end service that fulfills the execution request. Communication between these servers is described in the User Code Execution Protocol [MS-SPPTC].

2.1.3.16   Word Automation Services

Word Automation Services provides the capability to convert large numbers of input files in groups, each group using a specific set of conversion preferences (for example, embed **fonts** in the output file).

The Word Automation Services WCF Service Protocol as described in [MS-WORDSWCF] and the following diagram illustrates the relationship between this protocol and the service components.
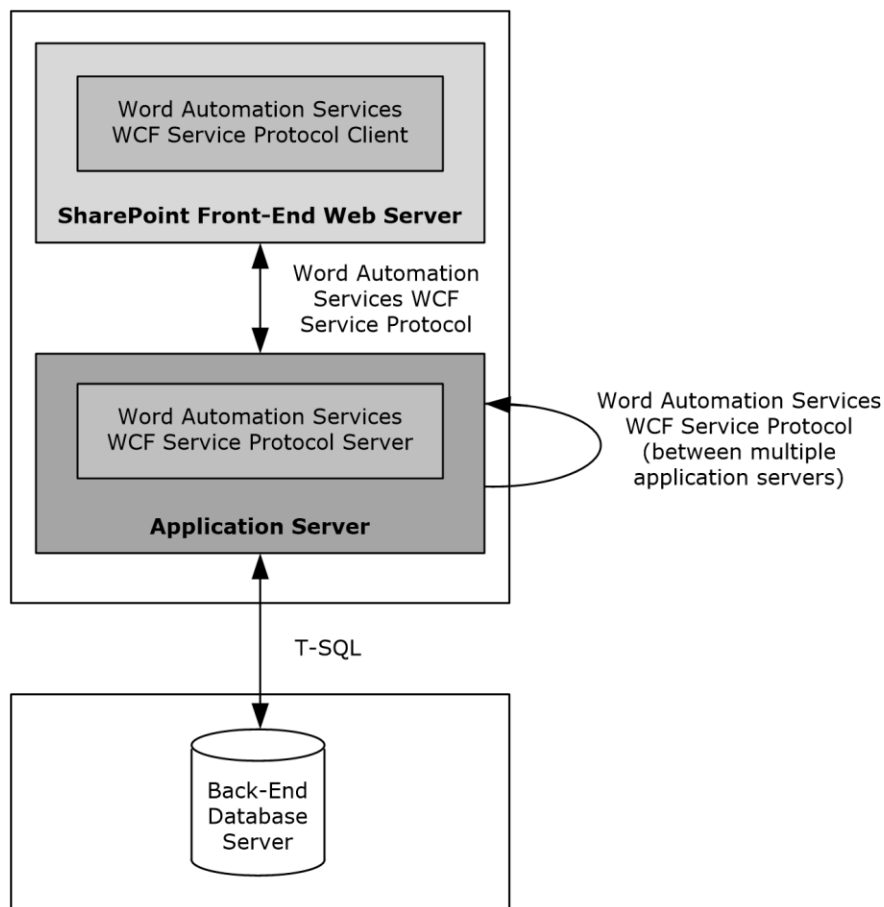
Figure 7: Relationship between the components of Word Automation Services

The Word Automation Services WCF Service Protocol is used to convert documents from one file format to another file format, as described in [MS-WORDSWCF]. This protocol allows a protocol client to create and manage a persistent queue of document conversions. It also allows the protocol client to initiate the conversion of documents in the queue from one file format to another file format. The protocol server maintains the queue and initiates conversions based on requests from a protocol client.

2.1.3.17   Visio Graphics Service

The Visio Graphics Service enables the **server** to render a drawing page in a Web drawing as a raster or vector drawing. The rendered drawing can then be viewed in a Web browser. The Web drawing can include connections to data sources that the Visio Graphics Service can query to update information displayed in the drawing page. The Visio Graphics Service also enables comments on the drawing page to be created, viewed, and edited in a Web browser.

The typical scenario for using these services is to create a Web drawing by using a client such as Microsoft Visio 2010 or Microsoft Visio 2013, storing the drawing on the server, and then viewing it in a Web browser. For example, a user can create a Web drawing that shows a business process and operational metrics for the steps in the process. The operational metrics are linked to data in a database. Users can view the Web drawing in a Web browser and see the business process overlaid with the current operational data from the database.

2.2 Protocol Summary

The tables in this section provide a comprehensive list of the member protocols of the SharePoint Front-End Protocols System. The member protocols are grouped according to their primary purpose.

2.2.1 Core Platform Protocols

This section lists the front-end protocols that are used to interoperate with the core platform components of SharePoint Products and Technologies.

2.2.1.1 Site Administration

The protocols in the following table enable **site** administration in SharePoint Server 2013. These protocols are generic across all **site collections** and sites.

Protocol name	Description	Short name
Administration Web Service Protocol	Provides the ability to create and delete site collections and to retrieve information about a site collection, such as which languages are supported by the site collection.	[MS-ADMINS]
Webs Web Service Protocol	Provides the ability to access and modify sites and the attributes of those sites in a site collection. This protocol can also be used to modify the design and layout of the sites.	[MS-WEBSS]

The protocols in the following table enable site administration for **Document Workspace sites** and **Meeting Workspace sites** in SharePoint Server 2013.

Protocol name	Description	Short name
Document Workspace Web Service Protocol	Provides the ability to create, edit, and delete Document Workspace sites. This protocol also allows the protocol client to organize the content in the Document Workspace sites by using folders.	[MS-DWSS]
Meetings Web Services Protocol	Provides the ability to create, edit, and delete Meeting Workspace sites.	[MS-MEETS]

The protocol in the following table enables **site membership** and directory integration in SharePoint Server 2013.

Protocol name	Description	Short name
SharePoint Email Web Service Protocol	Provides the ability to create and configure **distribution lists**. SharePoint Products and Technologies includes a simple implementation of this Web service that can be configured to use **Active Directory Service Interfaces (ADSI)** to transmit requests to AD DS.	[MS-SPEMAWS]

2.2.1.2 List Data

The protocols in the following table enable core SharePoint list and library data access in SharePoint Server 2013.

Protocol name	Description	Short name
Lists Web Service Protocol	Provides the ability to create, edit, delete, and configure **lists (1)** and list (1) data and to configure settings and metadata that are specific to lists (1).	[MS-LISTSWS]
Views Web Service Protocol	Provides methods for adding a list (1) view and for retrieving a specific view or the collection of list (1) views.	[MS-VIEWSS]
Forms Service Protocol	Provides methods for getting a list (1) of **forms** from the **server** and for retrieving an individual form from that list (1). Protocol clients can use these methods to display viewing and editing forms for individual list items to users.	[MS-FORMS]
ListData Data Service Protocol	Provides the ability to create, edit, delete, and retrieve list (1) list (2) data.	[MS-WSSREST]

The structures in the following table enable SharePoint list and library data access in SharePoint Server 2013.

Structure name	Description	Short name
Collaborative Application Markup Language (CAML) Structure	Describes the use of **Collaborative Application Markup Language (CAML)** to describe queries, views, and lists (1) across multiple external and internal protocols that are implemented by SharePoint Products and Technologies.	[MS-WSSCAML]
ADO XML Persistence Format Protocol	Describes the subset of the ADO XML Persistence Format used to represent tabular data, such as list (2) data returned by [MS-LISTSWS].	[MS-PRSTFR]

The protocol in the following table enables **alerts (2)** for SharePoint list (1) and library data in SharePoint Server 2013 and later.

Protocol name	Description	Short name
Alerts Service Protocol	Allows a protocol client to list and delete **alert subscriptions**, which specify when and how notifications are sent to users when changes are made to content that is stored on the server.	[MS-ALERTSS]

The protocol in the following table enables the Office Mobile Service Protocol. SharePoint Server 2013 can use this protocol to send an alert (1) to a protocol server, and the protocol server sends the alert (1) to a mobile phone.

Protocol name	Description	Short name
Office Mobile Service Protocol	Enables a protocol client, such as SharePoint Server 2013, to transmit mobile messages to a protocol server.	[MS-OMS]

The protocols in the following table enable additional ways to access list (2) data in SharePoint Server 2013. These protocols are implemented for historical interoperability with Microsoft Office Excel 2003 and with Microsoft Office FrontPage 2003, which was superseded by Microsoft Office SharePoint Designer 2007. For common usage scenarios, implementing these protocols is not advised because the same capabilities are available through the Lists Web Service Protocol, as described in [MS-LISTSWS].

Protocol name	Description	Short name
Data-Source Adapter SharePoint Team Services	Enables a protocol client to obtain structured tabular data from a server and provides access to metadata about the server and how	[MS-DSPSTSS]

Protocol name	Description	Short name
Web Service Protocol	the tabular data is organized.	
Windows SharePoint Services Collaborative Application Protocol	Enables a protocol client to access and manipulate lists (1) and list (2) views on the server.	[MS-WSSCAP]

The protocols in the following table enable synchronization with Microsoft Outlook 2013 in SharePoint Server 2013 and later.

Protocol name	Description	Short name
Lists Client Sync Protocol	Enables the transfer of organized collections of data, referred to as "items," between a server and a client. Determines the processing and logic to maintain local copies of SharePoint list (1) items in Outlook 2013.	[MS-OUTSPS]
StsSync Structure	Instantiates a connection between Outlook 2013 and a SharePoint list (1) by using a URL format.	[MS-STSSYN]
Alerts Interoperability Protocol	Extracts data from an alert (2) sent from SharePoint Products and Technologies to Outlook 2013.	[MS-OSALER]

2.2.1.3 File Operations

The protocols in the following table enable file operations in SharePoint Server 2013 and later.

Protocol name	Description	Short name
HTTP Extensions for Distributed Authoring -- WEBDAV	Provides file access and content management.	[RFC2518]
Web Distributed Authoring and Versioning (WebDAV) Protocol: Client Extensions	Provides file access and content management over the Internet. These extensions to WebDAV provide enhanced features, such as new error codes and additional TCP ports.	[MS-WDV]
Web Distributed Authoring and Versioning (WebDAV) Protocol: Microsoft Extensions	Provides extensions and limitations to WebDAV as implemented by SharePoint Products and Technologies.	[MS-WDVME]
Web Distributed Authoring and Versioning (WebDAV) Protocol: Server Extensions	Provides extensions to WebDAV, including headers that both enable the file types that are not currently manageable and optimize protocol interactions for file system clients.	[MS-WDVSE]
Web Distributed Authoring and Versioning Error Extensions Protocol	Describes the extended error codes that SharePoint Products and Technologies can return to a client application. The extended errors include: Document checked out Minor version limit extended Missing necessary document properties.	[MS-WEBDAVE]
FrontPage Server Extensions Remote Protocol	Provides file access functionality that is similar to that of WebDAV, allowing a **website** to be presented as a **share**.	[MS-FPSE]

Protocol name	Description	Short name
FrontPage Server Extensions: Website Management Protocol	The FrontPage Server Extensions: Website Management Protocol is a superset of a smaller protocol known as FrontPage Server Extensions Remote Protocol, as described in [MS-FPSE]. The larger protocol is used to perform a wider array of website administration tasks, including theme management, site navigation, and document repository tasks.	[MC-FPSEWM]
Hypertext Transfer Protocol -- HTTP/1.1	Provides application-level methods for distributed, collaborative, and hypermedia information systems. This generic, stateless, protocol can be used for many tasks, including those for file access. If other protocols are not supported, Microsoft Office clients use **HTTP GET** for read-only access to files.	[RFC2616]
SharePoint Document Sharing Web Service Protocol	Allows protocol clients to enable sharing of documents stored in a document library and to generate or retrieve anonymous access links for guest users to access documents without sign-in.	[MS-DOCSWS]

The protocols in the following table enable file operations in Microsoft SharePoint 2007 Products and Technologies and SharePoint 2010 Products and later.

Protocol name	Description	Short name
Office Document Update Utility Extensions Protocol	Provides extensions to WebDAV, including header updates, a property that enables optimized protocol interaction for synchronization, and a property that allows the **server** to send the virus infection status of a document to client applications. The Document Workspace feature of Microsoft® Office clients uses these extensions to synchronize documents with SharePoint Products and Technologies.	[MS-WDVMODUU]

2.2.1.4 Security and Identity

The protocols in the following table enable user **authentication** and identification in SharePoint Server 2013 and later.

Protocol name	Description	Short name
Authentication Web Service Protocol	Identifies the **authentication mode** that is in use by a **web application**, and enables user authentication when **forms authentication** is the authentication mode.	[MS-AUTHWS]
People Web Service Protocol	Enables a protocol client to resolve and find **principals**.	[MS-PEOPS]
Office Forms Based Authentication Protocol	Enables a protocol client to authenticate to a **server** configured to use Office Forms Based Authentication.	[MS-OFBA]
OAuth 2.0 Authentication Protocol: SharePoint Profile	Describes the profile for server to server authentication performed by an application server with other application services.	[MS-SPS2SAUTH]
OAuth 2.0 Authentication Protocol Extensions	Describes extensions to the OAuth 2.0 Authentication Protocol. These extensions consist of additional parameters in the request URI and the JSON objects returned in the HTTP response body. These extensions provide additional functionality, such as finer granularity of token expiry periods, which is incremental over what is provided by the base specification.	[MS-OAUTH2EX]

The protocols in the following table enable claims-based identity in SharePoint Server 2013 and later.

Protocol name	Description	Short name
SharePoint Claim Provider Web Service Protocol	Enables a protocol client to enumerate, search for, and resolve the security principals that are available in SharePoint Server 2013 and later.	[MS-CPSWS]
SharePoint Security Token Service Web Service Protocol	Describes the restrictions placed on the messages for interoperating with the SharePoint Server 2013 and later implementation of a Security Token Service.	[MS-SPSTWS]

The protocols in the following table enable SharePoint Products and Technologies authorization in SharePoint Server 2013.

Protocol name	Description	Short name
Permissions Web Service Protocol	Enables client applications to manage permissions to server **sites** and **lists (1)** through a Web service. This protocol describes the messages for adding, modifying, retrieving, and removing permissions.	[MS-PERMS]
UserGroup Web Service Protocol	Enables the protocol client to define, modify, and retrieve information about users or **groups (1)** and the role definitions to which they are assigned in a specific site or **site collection**.	[MS-UGS]

2.2.1.5 Authoring Integration

The protocol in the following table enables extended HTTP headers in SharePoint Server 2013 and later.

Protocol name	Description	Short name
HTTP Windows SharePoint Services Headers Protocol	Enables previously undefined behaviors by means of new headers and messages, which are extensions to HTTP mechanisms. The new application scenarios that these extensions introduce include authenticating client connections, communicating error conditions, sending complex data, interacting with information rights management systems, interacting with antivirus systems, and interacting with **crawlers**.	[MS-WSSHP]

The protocols in the following table enable scenario-specific content authoring in SharePoint Server 2013 and later.

Protocol name	Description	Short name
MetaWeblog Extensions Protocol	Describes extensions and modifications to the MetaWeblog API (MWA). These extensions and modifications allow client applications to get and set the text and attributes of posts on a **blog**.	[MS-METAWEB]
Slide Library Web Service Protocol	Enables a protocol client to send a request to the protocol server and to then receive information about the existence of named slides, about all slides, about slides that are specified by their identifiers, or about slides that contain specific search strings.	[MS-SLIDELI]
Imaging Service Protocol	Enables a protocol client to retrieve, upload, and organize images in a **picture library**.	[MS-IMAGS]

The protocol in the following table enables diagnostic submission only in SharePoint Server 2013 and later.

Protocol name	Description	Short name
SharePoint Diagnostics Web Service Protocol	Enables a protocol client to submit details about an error report—details such as the call stack, an error message, or the operating environment.	[MS-SPDIAG]

The protocols in the following table enable file synchronization in SharePoint Server 2013 and later.

Protocol name	Description	Short name
File Synchronization via SOAP over HTTP Protocol	Enables a client application to make a SOAP Web service request that allows for incrementally uploading or downloading file changes, along with the related metadata changes, from a **server** that supports the protocol. This protocol enables the coauthoring feature in Word 2013 and PowerPoint 2013 and later.	[MS-FSSHTTP]
Binary Requests for File Synchronization via SOAP Protocol	Enables a protocol client to synchronize the state of a structured file that is hosted by a protocol server.	[MS-FSSHTTPB]
Binary Data Format for File Synchronization via SOAP	Provides a schema for representing traditional file data (a stream of bytes) efficiently in the storage model described in [MS-FSSHTTP].	[MS-FSSHTTPD]

The protocol in the following table enables coauthoring in SharePoint Server 2013 and later.

Protocol name	Description	Short name
Shared Access Web Service Protocol	Enables a client application to determine whether a document is being coauthored.	[MS-SHDACCWS]

2.2.1.6 Extensibility

SharePoint Products and Technologies implements a number of protocols that are designed specifically to provide extensibility and customization. These protocols can be divided into three major areas:

- Content import and export protocols
- **Workflow (1)** integration protocols
- Web page and **Web Part** customization protocols

2.2.1.6.1 Content Import and Export

The protocols and structure in the following table enable importing and exporting content from SharePoint Server 2013 and later.

Protocol or structure name	Description	Short name
Sites Web Service Protocol	Provides methods for importing and exporting a **site** from a **server**, retrieving **site template** information, and obtaining a **form digest validation** value for authenticating a secure submit request that was made to the server.	[MS-SITESS]

Protocol or structure name	Description	Short name
RSS 2.0 Format Extensions	Provides methods for serializing data from a **website**.	[MS-SPRSS]
Deployment Package Format	Defines a set of **XML** and binary files to encapsulate site content and relationships. This file format enables the portability of user data.	[MS-PRIMEPF]

2.2.1.6.2 Workflow Integration

The protocols in the following table enable **workflow (1)** development and administration as well as user interaction in SharePoint Server 2013 and later.

Protocol name	Description	Short name
Workflow Web Service Protocol	Starts new workflows (1), changes existing ones, and queries for status. A common usage scenario for this protocol is managing a document approval process.	[MS-WWSP]
Web Part Pages Web Service Protocol	Describes a format for workflow (1) definitions that will be deployed to a **server** running SharePoint Products and Technologies.	[MS-WPPS]

2.2.1.6.3 Web Page and Web Part Customization

As described in section 2.1.1.2.4, SharePoint Products and Technologies implements a capability for a Web page to integrate multiple pieces of content through the Web Part mechanism.

The protocols in the following table enable Web Part page customization in SharePoint Server 2013 and later.

Protocol name	Description	Short name
Web Part Pages Web Service Protocol	Enables a protocol client to gather the information that is necessary to author Web pages that use **server** resources, and allows the authoring of execution logic that will react to changes in the server state.	[MS-WPPS]
Content Area Toolbox Web Service Protocol	Enables a protocol client to enumerate the **Web controls** that are available for use on a Web page that is stored on a protocol server. A common usage scenario involves a Web page editing application that queries the protocol server for Web controls and then displays those controls so that the user can add them to a Web page.	[MS-CONATB]

The protocol in the following table enables **workflow (1)** development and administration as well as user interaction in both SharePoint 2007 Products and Technologies and SharePoint 2010 Products.

Protocol name	Description	Short name
Group Approval Web Service Protocol	Submits or approves Group Approval documents. A common usage scenario for this protocol involves a custom Group Approval application that allows users to remotely access Group Approval	[MS-EAWF]

Protocol name	Description	Short name
	data.	

2.2.1.7 SharePoint Client-Side Object Model

The protocols in the following table enable the SharePoint Client-Side Object Model in SharePoint Server 2013 and later.

Protocol name	Description	Short name
SharePoint Client Query Protocol	Enables a protocol client to execute a batch of method invocations, property read operations, and property write operations on logical objects or class types.	[MS-CSOM]
SharePoint Client-Side Object Model Protocol	Defines the types, methods, and properties of objects such as **sites** and **lists (1)**.	[MS-CSOMSPT]
SharePoint Client Query OData Protocol	Enables a protocol client to use common web technologies to access types, methods, properties and objects such as sites and lists (1).	[MS-CSOMREST]

2.2.1.8 Configuration Services

The protocols in the following table enable the retrieval, sharing, and storage of configuration data in SharePoint Server 2013 and later and SharePoint Foundation 2013.

Protocol name	Description	Short name
SharePoint App Management Web Service Protocol	Allows the protocol client to retrieve a set of **data ranges** the protocol server maintains and the corresponding database **connection strings**, as well as a flag for each database indicating whether the database is cloud-based.	[MS-APPMWSP]
SharePoint Subscription Settings Web Service Protocol	In enterprise and hosting scenarios, a number of **site collections** need to use a shared set of configuration data. This allows for the potential of some configuration data for this group of site collections to be delegated to additional administrators. To facilitate these scenarios, this protocol has the ability to set, remove, and read **property sets** for a particular set of site collections. Applies to SharePoint Foundation 2013.	[MS-SPSETWS]
Service Platform Topology Web Service Protocol	Identifies the communication between the protocol client and the protocol server that is used to obtain a list of service applications and their properties. This protocol enables the protocol client to discover the list of service applications known to the protocol server. It also enables the protocol client to retrieve the list of endpoint URIs for a service application so that it can communicate with it using a protocol supported by that service application. Applies to SharePoint Foundation 2013.	[MS-SPTWS]

2.2.2 Extended Capabilities Protocols

This section lists the external protocols that are used to interoperate with the extended capabilities of SharePoint Products and Technologies.

2.2.2.1 Excel Services

The protocols in the following table enable Excel Services in SharePoint Server 2013.

Protocol name	Description	Short name
Excel Web Services Protocol	Enables a protocol client to access workbooks that are stored on a **Web server** or in a location that can be reached by using a Universal Naming Convention (UNC) path.	[MS-ESP]
Excel Services Publishing Protocol	Enables a client application, such as Excel 2013, to form the URL and associated parameters for a query string to display the workbook in the browser after the workbook is **published** to the protocol server.	[MS-ESURL]
Excel Services REST Protocol	Enables a client application to access information stored in **workbooks** that are maintained by a protocol server. The information returned by the protocol can be in one of several formats, such as **HTML**, image, and as an **atom feed**. This protocol client also allows for values to be inserted into cells of the workbook.	[MS-ESREST]
Excel Calculation Version 2 Web Service XML	Identifies the structure used to send a workbook from the back-end server to the front-end server; this is an XML format as described in [XMLSCHEMA].	[MS-EXSPXML3]
Excel Calculation Web Service Version 3 Protocol	Accesses workbooks stored on a front-end Web server or in a location that can be reached by using a UNC path. For an overview about the specific implementation of Excel Services that implements this protocol, see [MSDN-ESO]. Applies to SharePoint Server 2013.	[MS-EXSPWS3]

The protocol in the following table enables access to Excel Services workbooks in both SharePoint 2007 Products and Technologies and SharePoint 2010 Products.

Protocol name	Description	Short name
Excel Calculation Web Service XML	Identifies the structure used to send a workbook from the back-end server to the front-end server; this is an XML format as described in [XMLSCHEMA].	[MS-EXSPXML]

The protocol in the following table enables access to Excel Services workbooks in SharePoint 2010 Products only.

Protocol name	Description	Short name
Excel Calculation Web Service Version 2 Protocol	Accesses, enables editing, and saves workbooks stored on a front-end Web server, or accesses workbooks that can be reached by using a UNC path. For an overview about the specific implementation of Excel Services which implements this protocol, see [MSDN-ESO]. This is implemented in Microsoft SharePoint Server 2010.	[MS-EXSPWS2]

The protocol in the following table enables access to Excel Services workbooks in SharePoint 2007 Products and Technologies only.

Protocol name	Description	Short name
Excel Calculation Web Service Protocol	Accesses workbooks stored on a front-end Web server or in a location that can be reached by using a UNC path. For an overview about the specific implementation of Excel Services that implements this protocol, see [MSDN-ESO]. This is implemented in Microsoft Office SharePoint Server 2007.	[MS-EXSPWS]

2.2.2.2 Access Services

The protocols and structures in the following tables enable Access Services in SharePoint Server 2013.

For Access web application, the following protocols and structures are used.

Protocol or structure name	Description	Short name
Access Services Protocol	Enables a client application, such as Access 2013, to retrieve user profiles, execute data macros, perform operations on the data list data, trigger compilation of a database application, and manage the Access services site version.	[MS-ASWS]
Access Application Transfer Protocol Structure	Defines a format for describing **database application** components.	[MS-AXL]
Access Template File Format	Enables the creation of the structure and content of an Access web application.	[MS-ACCDT]
Access Services Data Server Protocol	Enables a client application, such as Access 2013, to operate the data, trigger a data macro, and retrieve information for the protocol server or data macro from SharePoint Server 2013.	[MS-ADS]

For Access Services, the following protocols and structures are used.

Protocol or structure name	Description	Short name
Access Application Design Time Protocol	Enables a client application, such as Access 2013, to modify the behavior of database applications and to create, modify, and delete objects in such applications.	[MS-AADT]
Access Application Transfer Protocol Structure Version 2	Defines a format for describing database application components.	[MS-AXL2]
Access Services Data Run Time Protocol	Enables a client application, such as Access 2013, to operate the data or objects, retrieve information for the protocol server or objects, and modify the behavior for database applications that have been **published** to SharePoint Server 2013.	[MS-ADR]
Access Services Data Server Protocol	Enables a client application, such as Access 2013, to operate the data, trigger a data macro, and retrieve information for the protocol server or data macro from SharePoint Server 2013.	[MS-ADS]
Access Run Time Protocol	Enables a client application, such as Access 2013, to send and receive data, to and from database applications that have been published to SharePoint Server 2013.	[MS-ART]

Protocol or structure name	Description	Short name
Access Server Design Time Protocol	Enables a client application, such as Access 2013, to create and delete database applications on SharePoint Server 2013.	[MS-ASDT]

2.2.2.3 InfoPath Forms Services

The protocols and structures in the following table enable InfoPath Forms Services in SharePoint Server 2013.

Protocol or structure name	Description	Short name
Forms Services Design and Activation Web Service Protocol	Allows a protocol client to communicate with a **form server** to validate, **browser-enable**, or manipulate settings of **form template** files in supported scenarios.	[MS-FSDAP]
Forms Services Feature Detection Protocol	Allows a protocol client to detect whether InfoPath Forms Services is present and enabled on the **server**. InfoPath Forms Services is necessary to **publish** a browser-enabled **form template (.xsn) file**.	[MS-FSFDP]
InfoPath Form File Format Structure	Enables a form server, with an associated form template (.xsn) file, to render and edit **form** data from a **form file** in a Web browser.	[MS-IPFFX]
Universal Data Connection 2.0 XML File Structure	Provides a container for **data connection (3)** information.	[MS-UDCX]
InfoPath Data Connection File Download Protocol	Allows a protocol client to retrieve a **Universal Data Connection (.udc, .udcx) file**.	[MS-INFODCF]
Forms Services Proxy Web Service Protocol	Forwards **SOAP** messages for a client application, and returns the targeted Web service response.	[MS-FSPP]
InfoPath Form Template Format Version 2 Structure	Enables a form server to render and edit form data in a Web browser.	[MS-IPFF2]
InfoPath Digital Signing Protocol	Enables a form server client to digitally sign a form file.	[MS-IPDSP]

The protocol in the following table enables InfoPath Forms Services in SharePoint 2007 Products and Technologies only.

Protocol name	Description	Short name
InfoPath Form Template Format Structure	Enables a form server to render and edit form data in a Web browser.	[MS-IPFF]

2.2.2.4 User Profile Service

The protocols in the following table enable the User Profile Service in SharePoint Server 2013.

Protocol name	Description	Short name
User Profile Synchronization (UPS): Management Objects Data Structure	Identifies the format for the objects that are used to run and determine status for synchronization between data sources.	[MS-UPSMODS]
User Profile Synchronization (UPS): Lightweight Directory Access Protocol Version 3 Extensions	Identifies the extensions to the **Lightweight Directory Access Protocol (LDAP)** [RFC4510] that are used in the communication sequences between a Synchronization Service and directory **servers**.	[MS-UPSLDAP]
User Profile Synchronization (UPS): Schema Exchange Protocol Profile	Identifies the User Profile Synchronization Schema Exchange Protocol Profile [MS-UPSSCXPP]. This profile enables protocol clients to discover metadata for a server implementation of the User Profile Synchronization (UPS): Configuration Data Structure [MS-UPSCDS].	[MS-UPSSCXPP]
Published Links Web Service Protocol	Retrieves a **list (1)** of URLs and descriptive details about each URL that a protocol client can inspect to decide whether each URL is useful for its particular application. A common usage scenario involves providing a list (1) of SharePoint sites to be displayed in an Open or Save dialog box.	[MS-PLSP]
User Profile Service Web Service Protocol	Enables a protocol client to retrieve user profile data as well as data that is common among users. This protocol also enables a protocol client to manage other user profile properties, such as links, **group (2)** memberships, and colleagues. This protocol can be used to manage an enterprise social network and to enable users to search for people within their organization for communication purposes.	[MS-USPSWS]
User Profile Change Log Web Service Protocol	Allows client applications to query and synchronize the changes from a user profile store. This protocol provides querying for changes to specific user profiles or to all profiles on the server. The protocol is also designed to be used by search engines to incrementally **crawl** data from the user profile store.	[MS-USRPCH]
User Profile Social Data Web Service Protocol	Enables a protocol client to add, update, remove, and retrieve **social tags**, notes, and **social ratings** on documents and Web pages that are stored on the server.	[MS-UPSDWS]
SharePoint Activity Feed Web Interfaces Protocol	Enables a protocol client to retrieve **social networking** activity data for the users of a **website**.	[MS-SPAFWI]
User Activity Feed Cache Web Service Protocol	Identifies the communication that allows protocol clients to store data related to social feed posts and feed metadata in an in-memory cache on a middle-tier **application server**.	[MS-UPFCWS]
User Profile Import and Export Web Service Protocol	Enables importing and exporting user profile and member group data from the **user profile store**. A common usage scenario involves a synchronization application that runs at fixed intervals to keep the user profile store and a Lightweight Directory Access Protocol (LDAP) DS in sync.	[MS-UPIEWS]
User Profile Service Application Caching Web Service Protocol	Allows protocol clients to access user profile information through the middle-tier application server tier, rather than the back-end database server tier. The middle-tier application server caches commonly used user profile fields and expose them to protocol clients that can read these fields through this protocol.	[MS-UPSCWS]

Protocol name	Description	Short name
User Profile Property Service Application Web Service Protocol	Allows protocol clients to start interacting with the User Profile Service. A typical scenario is a protocol client connecting to a **site** and accessing user profile information relevant to the logged-on user.	[MS-UPWCFWS]

2.2.2.5 Search Services

The protocols in the following table enable search services in SharePoint Server 2013.

Protocol name	Description	Short name
Search Protocol	Allows a protocol client to issue **search queries** to a protocol server and to then receive the results.	[MS-SEARCH]
Site Data 2003 Web Service Protocol	Supports the creation of site indexes by external **index servers**. This protocol allows the creation of indexes from earlier implementations of SharePoint Products and Technologies.	[MS-SITED3S]
Site Data Web Service Protocol	Supports the creation of site indexes by external indexing services.	[MS-SITEDATS]
SPSCrawl Web Service Protocol	Allows protocol clients to read the value of any **item** within the context of a **site** or **service application**.	[MS-SPSCRWL]
Search Client Query Protocol	Allows a protocol client to issue search queries against a protocol server hosting a **search service application**.	[MS-SRCHCSOM]
MS Search Lotus Notes Web Service Protocol	Enables the protocol client to enumerate and retrieve content and properties from a Lotus Notes Domino Server.	[MS-NOTESWS]
Search Query Shared Services Protocol	Allows protocol clients to execute queries, retrieve search results, and record click-through items on search results. This is a WCF based protocol.	[MS-QSSWS]
Fast Query Language Version 2 Protocol	Specifies the advanced query language to execute search queries.	[MS-FQL2]
Keyword Query Language Structure Protocol	This document specifies the structure of the Keyword Query Language (KQL). KQL is a language for expressing search criteria.	[MS-KQL]

The protocol in the following table enables analytics search services in SharePoint Server 2013.

Protocol name	Description	Short name
SharePoint Analytics Client-Side Object Model Protocol	Allows a protocol client to retrieve historical usage information about **events (2)** that occurred for items stored on the protocol server.	[MS-SPACSOM]

The protocols in the following table enable search services in SharePoint 2010 Products.

Protocol name	Description	Short name
Search Site Administration Web Service Protocol	Enables the protocol client to configure and control the specific SharePoint site functions such as Best Bets and Scopes.	[MS-SADMWS]
MSSearch Query Version	Allows a protocol client to communicate with a **query server** to	[MS-SQP2]

Protocol name	Description	Short name
2 Protocol	issue queries. The search service running on the query server helps efficiently organize the extracted features of a collection of documents.	
Search Admin Web Service Protocol	Enables the protocol client to configure and control the crawling function of the search subsystem. It allows the protocol client to define the URL space of the crawl by configuring starting points and restriction rules, thereby determining the credentials used during the crawl by the index server and controlling when the crawl process occurs. Applies to SharePoint Server 2010.	[MS-DADMWS]

The protocol in the following table enables the index format in both SharePoint 2007 Products and Technologies and SharePoint 2010 Products.

Protocol name	Description	Short name
Content Index Format Structure	Identifies the format of the internal full-text index.	[MS-CIFO]

The protocols in the following table enable search services in Office SharePoint Server 2007.

Protocol name	Description	Short name
Search Service Administration Web Service Protocol	Enables the protocol client to configure and control the crawling function of the search subsystem. It allows the protocol client to define the URL space of the crawl by configuring starting points and restriction rules, thereby determining the credentials used during the crawl by the index server and controlling when the crawl process occurs.	[MS-ADMWS]
Global Admin Web Service Protocol	Enables remote administration of the Search application. Applies to Office SharePoint Server 2007.	[MS-GLOADWS]
MSSearch Query Protocol	Allows a protocol client to communicate with a query server to issue queries. The search service running on the query server helps efficiently organize the extracted features of a collection of documents. Applies to Office SharePoint Server 2007.	[MS-SQP]

2.2.2.6 Business Connectivity Services

The protocols in the following table enable **Business Connectivity Services (BCS)** in SharePoint Server 2013.

Protocol name	Description	Short Name
Business Data Catalog Metadata Web Service Protocol	Describes an interface for protocol clients to retrieve information about interfaces of other line-of-business applications and annotations of these interfaces.	[MS-BDCMP]
Business Data Connectivity Remote Administration Web Service Protocol	Enables protocol clients to create, store, and retrieve information about the interfaces of **LOB systems** and the annotations of these interfaces.	[MS-BDCRAWPS]

Protocol name	Description	Short Name
Business Connectivity Services Deployment Package File Format	Describes the content of a **deployment package** for a BCS solution; the extensions that are specific to BCS for the Visual Studio Tools for Office ClickOnce Schema, as described in [MS-OSCO]; and a file format that is specific to BCS for representing the properties of an **external list**.	[MS-BCSDPFFS]
Office Server ClickOnce Manifest Structure	Used to create a deployment package for deploying one or more customizations and their related components to a computer.	[MS-OSCO]
Business Connectivity Services Deployment Web Service Protocol	Describes the utility functions that can be used in the creation of development or administration tools and applications interacting with BCS solutions.	[MS-BCSDWPS]
External Content Type Picker Web Service Protocol	Enables protocol clients to find and resolve **EntityInstances** on a protocol server.	[MS-ECTPWPS]
Business Data Connectivity Model File Format	Identifies an XML structure used to describe the API of one or more line-of-business (LOB) systems and how that structure can be mapped to a well-defined interface of canonical operations. This structure is typically consumed by a run-time engine that translates canonical interface operation execution requests into the native LOB system's interface operation execution requests.	[MS-BDCMFFS]
Business Data Connectivity Web Service Protocol	Identifies an interface for protocol clients to retrieve information about interfaces of other line-of-business applications and annotations of these interfaces.	[MS-BDCWPS]
Secure Store Web Service Protocol	Identifies an interface for protocol clients to store and retrieve credentials and related information that is typically used to authenticate to line-of-business (LOB) systems.	[MS-SSWPS]

The protocols in the following table were used by Business Connectivity Services (BCS) in Office SharePoint Server 2007.

Protocol name	Description	Short Name
Business Data Catalog Data Web Service Protocol	Describes an interface that protocol clients can use to search software systems that store business data and process rules for an instance of a particular entity.	[MS-BDCDP]
Single Sign-On Protocol	Identifies a **remote procedure call (RPC)**-based protocol used by protocol clients to obtain a **master secret** from a **master secret server**.	[MS-SSP]

2.2.2.7 Content Management Service

The protocols in the following table enable content management in SharePoint Server 2013.

Protocol name	Description	Short Name
Copy Web Service Protocol	Provides a method for copying files—along with their associated properties, such as the creation date and author—to one or more locations on a single **server** or on multiple servers running SharePoint Products and Technologies.	[MS-COPYS]
Document Transforms	Enables a protocol client to call a file conversion service that	[MS-DOCTRANS]

Protocol name	Description	Short Name
Service Protocol	converts a file from one format to another.	
Official File Web Service Protocol	Enables the submission of files to a repository for the purpose of long-term retention according to a records management policy.	[MS-OFFICIALFILE]
Publishing Web Service Protocol	Provides access to functionality that is related to publishing and content management. This protocol, which is used in Web content management scenarios, allows for the control of page layout, page status, and multilingual translation.	[MS-PUBWS]
Versions Web Service Protocol	Enables a protocol client to programmatically view and manage the versions of **files** on the server. This protocol enables retrieving all the versions of a file, deleting the previous versions of a file, and restoring a file to a previous version.	[MS-VERSS]
SpellCheck Web Service Protocol	Enables a protocol client to check the spelling of a set of words.	[MS-SPLCHK]
Metadata Shared Service WCF Service Protocol	Describes a set of operations for reading, writing, and managing metadata in a centrally managed metadata store.	[MS-EMMWCF]
Microsoft Enterprise Managed Metadata Web Service Protocol	Describes a set of operations that enable protocol clients to read, write, and manage metadata in a centrally managed metadata store.	[MS-EMMWS]
Template Discovery Web Service Protocol	Enables a protocol client to retrieve links to the document templates that are associated with a **document library** on the protocol server.	[MS-TMPLDISC]
eDiscovery Internal Web Service Protocol	This protocol enables a protocol client to perform UI interaction operations in support of an application that manages a **discovery case**. For example, it enables a protocol client to validate whether the identifier of a location that a user has entered is a valid location to use as a **discovery source** or to update statistics for the discovery case once a discovery source is added.	[MS-EDINTWS]
Enterprise Managed Metadata Client-Side Object Model Protocol	Enables a protocol client to access enterprise managed metadata in a **term store** on a protocol server. The enterprise managed metadata is accessed by using types, methods, and properties defined by this protocol. Some typical scenarios in which a protocol client could use this protocol are to retrieve **terms**, add terms, add **term labels**, or remove terms.	[MS-EMMCSOM]
Publishing Client-Side Object Model Protocol	Enables a protocol client to manage publishing **websites** on a protocol server through types, methods, and properties defined by this protocol. For example, a protocol client can retrieve a tree of menu entries from the protocol server. A protocol client can also update the managed navigation data on the protocol server to modify URLs, the hierarchy, and other aspects of menu navigation.	[MS-PUBCSOM]
Document Management Client Side Object Model Protocol	Enables a protocol client to manage publishing videos on a protocol server through types, methods and properties defined by this protocol. For example, a protocol client can access video data, upload a video, and view and update the **embed code** for a **video container**.	[MS-DMCSOM]
eDiscovery Client-Side Object Model Protocol	Enables a protocol client to manage a discovery case through types, methods, and properties defined by this protocol. For example, a protocol client can retrieve a discovery case from the protocol server, retrieve the associated discovery sources, and perform operations such as initiating a **legal hold** on those	[MS-EDCSOM]

Protocol name	Description	Short Name
	discovery sources.	
Content Deployment Remote Import Web Service Protocol	Enables a protocol client to remotely manage the import of a set of content that was previously exported from a SharePoint Products and Technologies protocol server to another protocol server.	[MS-CDEPLOY]
eDiscovery Service Application Protocol	Enables a protocol client to manage actions that can be applied to the supported sources of content. Actions can be invoked and deleted. The protocol also provides the means to get the status of the actions that have already been sent to the protocol server. A typical scenario for using this protocol is an application used to centrally manage the different sources of content.	[MS-EDSA]
SharePoint Analytics Service Application Protocol	This protocol enables a protocol client to send usage **events (2)** to the protocol server. The protocol server performs implementation specific aggregation of those events (2) and produces historical usage information based on those events (2). This protocol further enables a protocol client to retrieve the aggregated usage information from the protocol server. A typical scenario for using this protocol is a reporting application that allows the users to visualize the usage of an item during its lifetime.	[MS-SPASA]

The protocol in the following table enables Analytics Services only in SharePoint 2010 Products

Protocol name	Description	Short Name
Microsoft Office SharePoint Server (MOSS) Analytics Reporting Protocol	Provides methods for logging events (2) for analyzing and retrieving analytical data.	[MS-MAR]

2.2.2.8 PerformancePoint Services

The protocols in the following table enable PerformancePoint Services 2010 in SharePoint Server 2013.

Protocol name	Description	Short name
PerformancePoint Services Authoring Service Protocol	Describes a Web service that enables reading, writing, and performing basic operations on **dashboards** that are hosted on a protocol server. The methods of this protocol center mainly on the ability to author and edit such dashboards.	[MS-PPSAS]
PerformancePoint Services Decomposition Tree Protocol	Allows a protocol client to perform contribution analysis on a data value by using a data source or data source context that is known to the protocol server. This protocol allows a protocol client to retrieve either grouped, named actions that apply to a specific data value or named properties and values that are associated with that node, and to then retrieve additional values that contribute to the value. The protocol server returns data points and provides or denies summary data for the data points that it does not return.	[MS-PPSDECO]
PerformancePoint Services Application Server	Identifies the communications used between PerformancePoint Services on a **front-end web server** and a back-end server (2)	[MS-PPSAPP]

Protocol name	Description	Short name
Protocol	storing business intelligence metadata.	
Audit and Control Management Server Audit Web Service Protocol	Enables communication between a protocol server that handles audit control features and a protocol client that consumes the web service to implement audit control features of the document control system.	[MS-ACMAUDWS]

2.2.2.9 Translation Services

The protocol in the following table enables Translation Services in SharePoint Server 2013.

Protocol name	Description	Short name
Translation Services Client-Side Object Model Protocol	Allows a protocol client to automatically machine translate documents stored in SharePoint document libraries.	[MS-TSCSOM]
Translation Services Web Service Protocol	This protocol enables a protocol client to create and manage a persistent queue of document translations. It also allows the protocol client to initiate the machine translation of documents in that queue from one language to another.	[MS-TSWS]

2.2.2.10 Office Online Web Query Service

The protocol in the following table enables Office Online Web Query in SharePoint 2010 Products only.

Protocol name	Description	Short name
Office Online Web Query Web Service Protocol	Allows a protocol client, such as SharePoint 2010 Products, to enumerate details about files in the Office Online repository.	[MS-OOWQWS]

2.2.2.11 Workflow Services

The protocol in the following table enables Workflow Services in SharePoint Server 2013.

Protocol name	Description	Short name
SharePoint Workflow Client-Side Object Model Protocol	Allows a protocol client to manage and interact with **workflow (1)** functionality such as **workflow associations**, **workflow events**, and **workflow instances**.	[MS-SPWFCSOM]

2.2.2.12 Work Management Service

The protocol in the following table enables the Work Management Service in SharePoint Server 2013.

Protocol name	Description	Short name
Work Management Client-Side Object Model Protocol	Allows a protocol client to access and manipulate **tasks** assigned to a user.	[MS-WMCSOM]
Work Management Service Web Service Protocol	Allows protocol clients to refresh or synchronize a user's task information through a middle-tier **application server**. The protocol client sends a request for a user to the protocol server, and the **server** responds with an acknowledgement of the request and refreshes the task information asynchronously.	[MS-WMS]

2.2.2.13 Social Services

The protocol in the following table enables the SharePoint Social Services Client Side Object Model in SharePoint Server 2013.

Protocol name	Description	Short name
SharePoint Social Client-Side Object Model Protocol	Allows a protocol client to access feeds and social data stored on a **site**, User Profile Service databases and memory cache.	[MS-SOCCSOM]
Community Client-Side Object Model Protocol	Defines types, methods, and properties that protocol clients use to communicate with and manipulate aspects of a community site or a **personal site** on a protocol server. The set of types, properties, and methods provide the ability for remote clients to perform moderation tasks on a community site such as reporting content as inappropriate and dismissing the complaints by users that content is inappropriate. The set of types, properties, and methods also provide the ability for remote clients to work with **lists (1)** on a user's personal site.	[MS-COMMCSOM]

2.2.2.14 PowerPoint Automation Services

The protocol in the following table enables presentation file conversion in SharePoint Server 2013.

Protocol name	Description	Short name
PowerPoint Automation Services Conversion Web Service Protocol	Allows a protocol client to convert a presentation from one file format to another file format. It allows the protocol client to pass a presentation file to the protocol server and to receive from the protocol server a converted file in a different format.	[MS-PASCWS]

2.2.2.15 User Code Execution Service

The protocol in the following table enables user code to be run remotely. This protocol is used by the User Code Execution Service in SharePoint Server 2013.

Protocol name	Description	Short name
User Code Execution	Used to remotely run user code within a **server farm** on one or	[MS-SPPTC]

Protocol name	Description	Short name
Protocol	more **front-end web servers** that are primarily dedicated to running user code rather than responding to Web requests.	

2.2.2.16   Word Automation Services

The protocols in the following table enable file conversion in SharePoint Server 2013.

Protocol name	Description	Short name
Word Automation Services WCF Service Protocol	Identifies the Windows Communication Foundation (WCF) service that is used to convert documents from one file format to another file format.	[MS-WORDSWCF]

2.2.2.17   Visio Graphics Service

The protocol in the following table enables the Visio Graphics Service in SharePoint Server 2013.

Protocol name	Description	Short name
Visio Graphics Service Protocol	Provides methods for retrieving Web drawings as raster or vector-formatted drawing pages.	[MS-VGSP]

The file formats in the following table enable the Visio Graphics Service in SharePoint Server 2013.

Protocol name	Description	Short name
Visio Graphics Service (.vdw) File Format	Provides the file that stores information about VGSFF Web drawings, which include drawing pages, **shapes**, fonts, images, **data connections (1)**, and diagram update information.	[MS-VGSFF]
Visio Graphics Service VSDX File Format	Provides the file that stores information about VSDX Web drawings, which include drawing pages, shapes, images, comments, data connections (1), and recalculation information.	[MS-VSDX]

2.3   Environment

The following sections identify the context in which the system exists. This includes the systems that use the interfaces provided by this system of protocols, other systems that depend on this system, and how components of the system communicate.

2.3.1   Dependencies on This System

The following system depends on the SharePoint Protocols system:

▪   The Office Client Protocols system, as described in [MS-OCPROTO].

2.3.2   Dependencies on Other Systems/Components

The SharePoint Front-End Protocols system depends on the following systems:

- The Windows system, as described in [MS-SYS-ARCHIVE].

- The Active Directory system, as described in [MS-ADTS] section 1

SharePoint Server 2013 depends on the following systems/components to function:

- The 64-bit edition of Windows Server 2008 R2 operating system with Service Pack 1 (SP1) or 64-bit edition of Windows Server 2012 operating system

- Microsoft .NET Framework 4.5

- Windows PowerShell 3.0

- The 64-bit edition of Microsoft SQL Server 2008 R2 Service Pack 1 (SP1) or 64-bit edition of Microsoft SQL Server 2012

2.4   Assumptions and Preconditions

The SharePoint Front-End Protocols system necessitates the following assumptions and preconditions:

- The protocol clients can access the **front-end Web servers** running SharePoint Products and Technologies via one or more established IP addresses.

- The functional components of the **servers** running SharePoint Products and Technologies are started collectively, and the front-end Web servers running SharePoint Products and Technologies accept requests from the protocol clients.

- The front-end Web servers can access the **back-end database servers** and have the appropriate permissions to access data in the content and configuration databases.

- The front-end Web servers and the back-end Web servers are running either the same versions of SharePoint Products and Technologies or versions that are within an acceptable range.

- If AD DS provides **authentication**, the **directory service (DS)** is accessible to the servers running SharePoint Products and Technologies. All the necessary ports and gateways of any intermediate firewalls, routers, or connection points between components of the system are open for communication.

The scope of this list of assumptions and preconditions is intended to be implementation-independent and limited to the system level.

2.5   Use Cases

The following use cases are provided to facilitate understanding of the SharePoint Front-End Protocols system overall.

Core platform use cases:

- Create a **site collection**

- Create a **site**

- Create a **list (1)** and add a **view**

- Get list data

- Create a file in a SharePoint library from a client computer

- Export and import site content

- Add a **Web Part** to a page and update its properties

Extended capabilities use cases:

- Excel Services: **Publish** a **workbook** to a SharePoint library

- Excel Services: Interact with a workbook by using a browser

- Excel Services: Interact with a workbook by using an application

- Access Services: Publish a **database application** to SharePoint Products and Technologies

- Access Services: Interact with a database application that has been **published** to SharePoint Products and Technologies

- InfoPath Forms Services: Publish a **form template** and enable browser rendering

- User Profile Service: Retrieve a user's profile

- User Profile Service: Tag a document

- Search services: Issue a **search query**

- **BCS**: Create an external content type

- BCS: Read an external content type

- BCS: Update an external content type

- BCS: Deploy a BCS solution

- BCS: Update a BCS solution

- Content Management Service: Get **term set** data and add term

- PerformancePoint Services 2010: Interact with published **dashboards** on a Web Part page

- Translation Services: Translate SharePoint documents by machine

- Work Management Service: Retrieve a user **task** list

- Work Management Service: Update a task

- Work Management Service: Change the order of a task

These use cases are not intended to provide a thorough and complete model of the system for any implementation.

2.5.1 Core Platform Use Cases

2.5.1.1 Create a Site Collection

This use case describes how a client application creates a new **site collection**. The client application specifies a **site definition** or **site template** to use when creating the new site collection.

The following diagram illustrates this process.
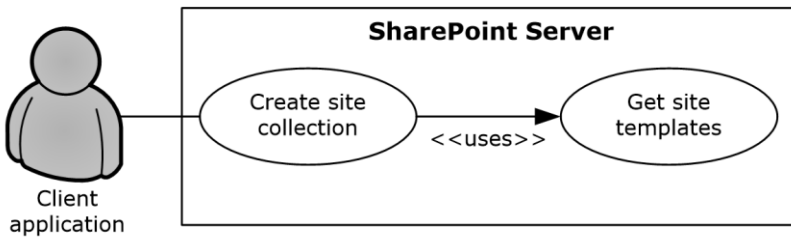
Figure 8: Process for creating a site collection

Preconditions

- The client application can communicate with the **Central Administration site**.

- The client application has been authenticated as a member of the Farm Administrators **group (1)**.

Steps

1. The client application requests a list of available site definitions and site templates from the **server**.

2. The server returns the list of available site definitions and site templates.

3. The client application selects a Web template and specifies the URL, title, and description for the site collection.

4. The client application calls the method to create a new site collection using the specified information.

5. The server creates a site collection with the specified information.

Error

- If the specified URL is invalid or already in use, the server will not create a new site collection.

Post-conditions

- A site collection is created at the URL that was specified by the client application.

- The site collection is not created, because the specified URL is invalid or already in use.

2.5.1.2 Create a Site

This use case describes how a client application creates a new **site**. The client application specifies a **site definition** or **site template** to use when creating the new site.

The following diagram illustrates this process.
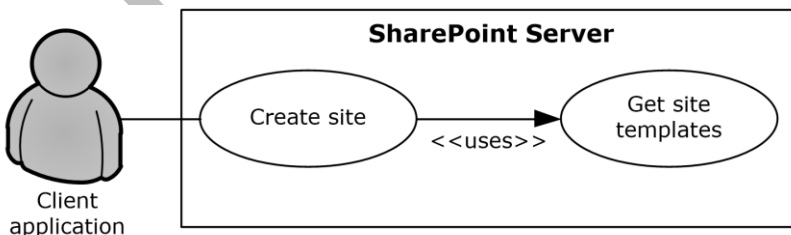
Figure 9: Process for creating a site

Preconditions

- The client application can communicate with the SharePoint site collection.

- The client application has been authenticated as a member of the Site Owner **group (1)**.

Steps

1. The client application requests a **list (1)** of available site definitions and site templates from the **server**.

2. The server returns the list (1) of available site definitions and site templates.

3. The client application selects a Web template and specifies the URL, title, and description for the site.

4. The client application calls the method to create a new site using the specified information.

5. The server creates a site with the specified information.

Error

- If the specified URL is invalid or already in use, the server will not create a new site.

Post-conditions

- A site is created at the URL that was specified by the client application.

- The site is not created, because the specified URL is invalid or already in use.

2.5.1.3 Create a List and Add a View

This use case describes how a client application creates a new **list (1)** and adds a view. The client application specifies the **list template** to use when creating the new list (1).

The following diagram illustrates this process.
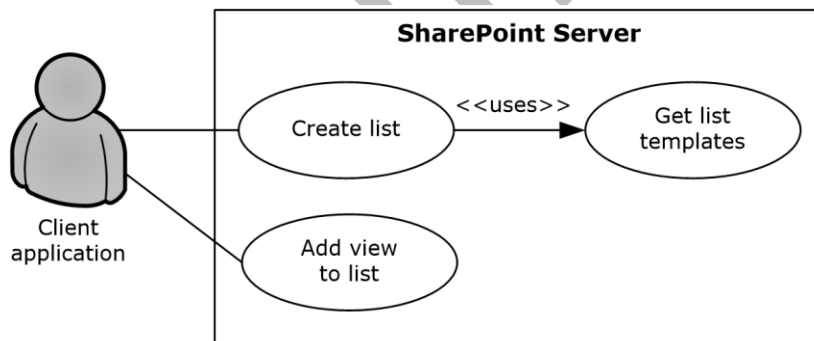
Figure 10: Process for creating a list (1) and adding a view

Preconditions

- The client application can communicate with the SharePoint site.

- The client application has been authenticated as a member of the Contributor **group (1)**.

Steps

1. The client application requests a list of list templates from the **server**.

2. The server returns the list of list templates.

3. The client application selects a list template and specifies the title and description for the new list (1).

4. The client application calls the method to create a new list (1) and passes the specified information.

5. The server creates a list (1) with the specified information.

6. The client application selects the list (1) that was created and specifies a view name and view fields.

7. The client application calls the method to add a view to the list (1) and passes the specified information.

8. The server adds a view to the list (1).

Post-conditions

- A new list (1) exists on the **site**.

- The new list (1) contains the new view.

2.5.1.4 Get List Data

This use case describes how a client application gets list data for display, building reports, or other common scenarios.

The following diagram illustrates this process.
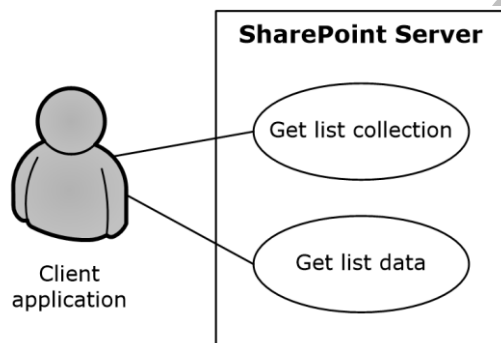
Figure 11: Process for getting list data

Preconditions

- The client application can communicate with the SharePoint site.

- The client application has been authenticated and has read permission on the **list (1)**.

Steps

1. The client application requests a list of the lists (1) that exist on the **site** from the **server**.

2. The server returns the list of lists (1) on the site.

3. The client application selects a list (1).

4. The client application requests data for the selected list (1).

5. The server returns the data for the list (1).

Post-condition

- The selected list (1) is displayed.

2.5.1.5 Create a File in a SharePoint Library from a Client Computer

This use case describes the simplest way to create a file by using the SharePoint Front-End Protocols system. The actor is the user, who creates a text file named hello.txt that contains the text "hello" in a SharePoint library.

The following diagram illustrates this file creation process.
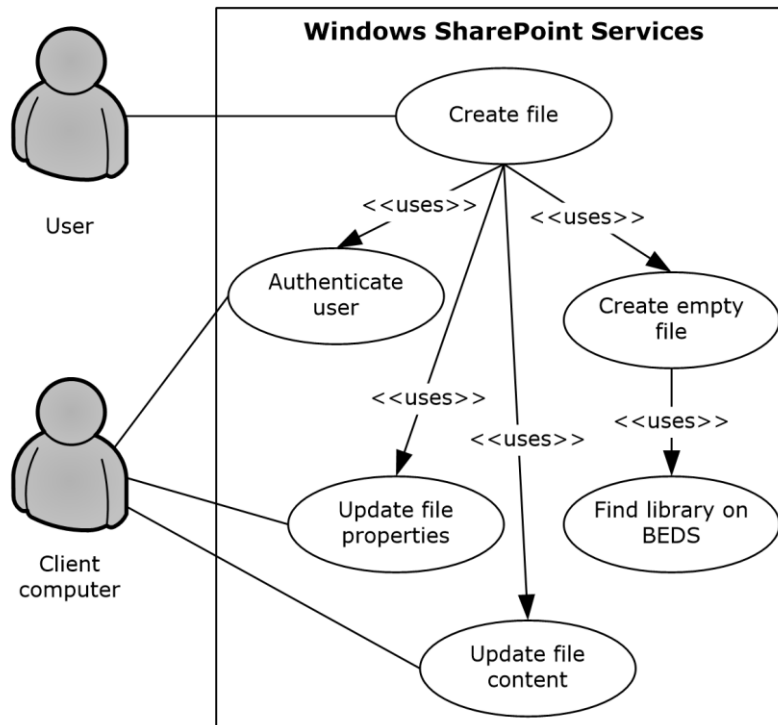
Figure 12: Process for creating a file in a SharePoint library from a client computer

Preconditions

- The user has permissions for an existing SharePoint library.

- The user is logged on to a client computer running Windows 7 operating system with an authenticated Windows session and can access the SharePoint Foundation 2013 site that contains the library. (Client computers running other versions of Windows might have different steps.)

- In the **Command Prompt** window, the user types the following command:
  echo hello >**server**site****doclib**\hello.txt**
  (where *server* is the name of the **server**, *site* is the name of the **site**, and *doclib* is the name of the document library).

Steps

1. The **front-end Web server** running SharePoint Foundation 2013 authenticates the user.

2. SharePoint Foundation 2013 finds the location of the document library on the **back-end database server** ("BEDS" in the figure) and verifies that the user has access to it.

3. SharePoint Foundation 2013 creates an empty file in the library and confirms success to the client computer.

4. The client computer updates the file properties and file contents.

Errors

- If the user does not have permissions, the client computer notifies the user that access was denied.

- If the client computer cannot connect to the front-end Web server, the client computer notifies the user of the error.

- If the client computer cannot update the file properties or file contents, the client computer notifies the user of the error.

Post-conditions

- The file is created, and the file properties and file contents are updated.

- The file is not created and the user is notified of the error.

2.5.1.6 Export and Import Site Content

This use case describes how a client application can export and import **site** content. Client applications export and import site content to migrate content from one site to another.

The following diagram illustrates the export and import process.
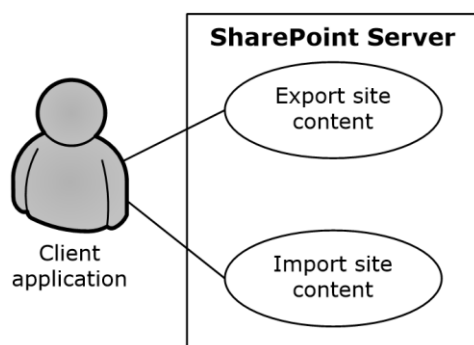
Figure 13: Process for exporting and importing site content

Preconditions

- The client application can communicate with the SharePoint site.

- The client application has been authenticated to the **server**, has open Web permissions and manage Web permissions for the source site, and has site creation permissions for the target site.

Steps

1. The client application calls the method to export the site and passes the URL of the site to export, the file name for the content migration package, and the directory in which to place the package.

2. The server packages the site content and places the content migration package in the specified directory.

3. The client application calls the method to import the site and passes the URL for the resulting site and the URL of the content migration package.

4. The server imports the content from the content migration package to the specified URL.

Errors

- If a site at a specified URL is not accessible, the server will return an error.

- If the directory from which to write and read the content migration package is not accessible, the server will return an error.

- If the client application lacks sufficient permissions to export or import the site, the server will return an error.

Post-conditions

- The destination site contains all the content from the content migration package.

- The content is not exported or imported, because an error occurred.

2.5.1.7 Add a Web Part to a Page and Update Its Properties

This use case describes how a client application adds a Web Part to a page and updates the properties of that Web Part. Web Parts function as a mechanism for customizing the presentation of a **website**.

The following diagram illustrates the Web Part add and update process.
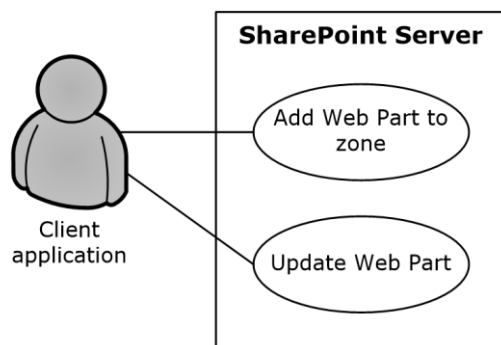
Figure 14: Process for adding a Web Part to a page and updating its properties

Preconditions

- The client application can communicate with the SharePoint site.

- The SharePoint site has a page that contains Web Part zones.

- The client application has been authenticated as a member of the Contributor **group (1)**.

- The Web Part to add is available on the **server**.

Steps

1. The client application specifies the URL of a page, the Web Part zone, the Web Part zone index, and the markup for the Web Part.

2. The client application calls the method to add a Web Part to a Web Part zone.

3. The server adds the Web Part to the Web Part zone.

4. The server returns the **GUID** that uniquely identifies the newly added Web Part.

5. The client application calls the method to update the Web Part and passes the same URL of the page, the GUID that identifies the Web Part, and the new properties.

6. The server updates the properties of the Web Part.

Error

- If the Web Part markup is invalid, the server will return an error.

Post-conditions

- The page contains a new Web Part with updated properties.

- The page does not contain a new Web Part with updated properties, because the Web Part markup is invalid.

2.5.2 Extended Capabilities Use Cases

2.5.2.1 Excel Services

2.5.2.1.1 Publish a Workbook to a SharePoint Library

This use case describes how a client application **publishes** a workbook to a SharePoint library.

The following diagram illustrates this process.
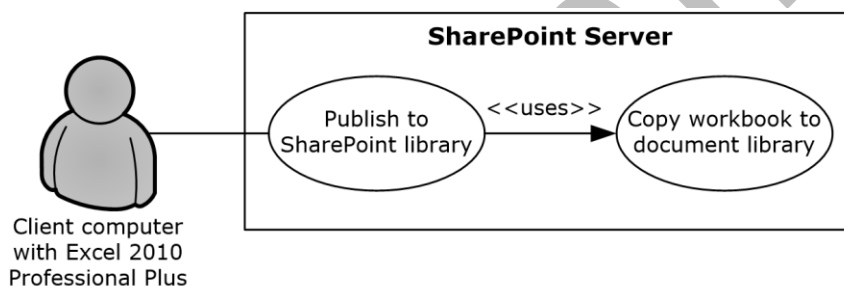
Figure 15: Process for publishing a workbook to a SharePoint library

Preconditions

- Excel Services is enabled on a farm running SharePoint Products and Technologies.

- The user has created a workbook in Excel 2013.

- The user has permission to save files to SharePoint Products and Technologies.

Steps

1. The user initiates publishing from Excel 2013.

2. Upon a request from Excel 2013, SharePoint Products and Technologies creates a file in a SharePoint library.

3. The workbook opens in a browser by using a URL that is specified according to the Excel Services Publishing Protocol, as described in [MS-ESURL].

Errors

- If the user does not have permission to create content in a SharePoint library, an error will be returned.

- If the workbook exceeds the size limit set in SharePoint Central Administration, an error will be returned.

- If the workbook contains unsupported features, the workbook will not open in the browser and an error will be returned.

Post-conditions

- A file exists in a SharePoint library.

- A file does not exist in a SharePoint library, because an error occurred.

2.5.2.1.2 Interact with a Workbook by Using a Browser

This use case describes how a browser uses Excel Services to interact with a workbook that has been **published** to SharePoint Products and Technologies.

The following diagram illustrates this process.
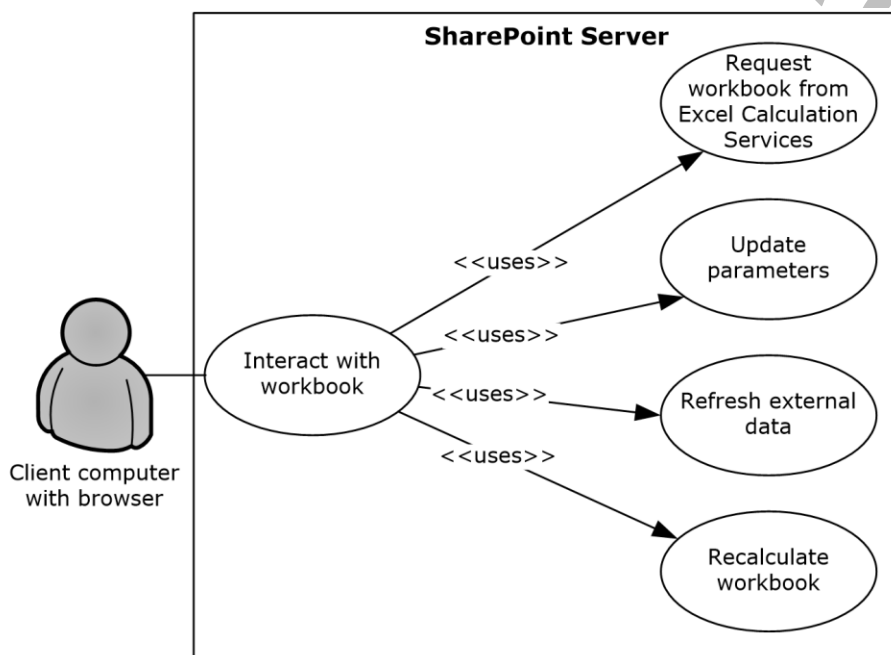
Figure 16: Process for interacting with a workbook by using a browser

Preconditions

- Excel Services is enabled on a farm running SharePoint Products and Technologies.

- The user has permissions for the requested workbook.

- The user is running a supported browser.

Steps

1. The user requests the workbook by either clicking it in the UI of a SharePoint library, typing the URL of the file in the browser, or opening a Web page that displays all or part of the workbook.

2. The viewable range of the workbook appears in the browser.

3. If parameters exist, the user can enter the values in the parameter pane.

4. If external **data connections (2)** exist, the user can request that the data be refreshed.

5. The user can request that the workbook be recalculated.

Errors

- If the user does not have permissions for the workbook, an error will be returned.

- If the workbook contains unsupported features that prevent it from opening, an error will be returned.

- If the workbook is connected to external data for which the user does not have permissions, an error will be returned.

Post-conditions

- The values of the workbook are recalculated according to the changes, and the workbook is displayed in the browser.

- The workbook is not displayed in the browser, because an error occurred.

2.5.2.1.3 Interact with a Workbook by Using an Application

This use case describes how an application uses Excel Services to interact with a workbook that has been **published** to SharePoint Products and Technologies.

The following diagram illustrates this process.
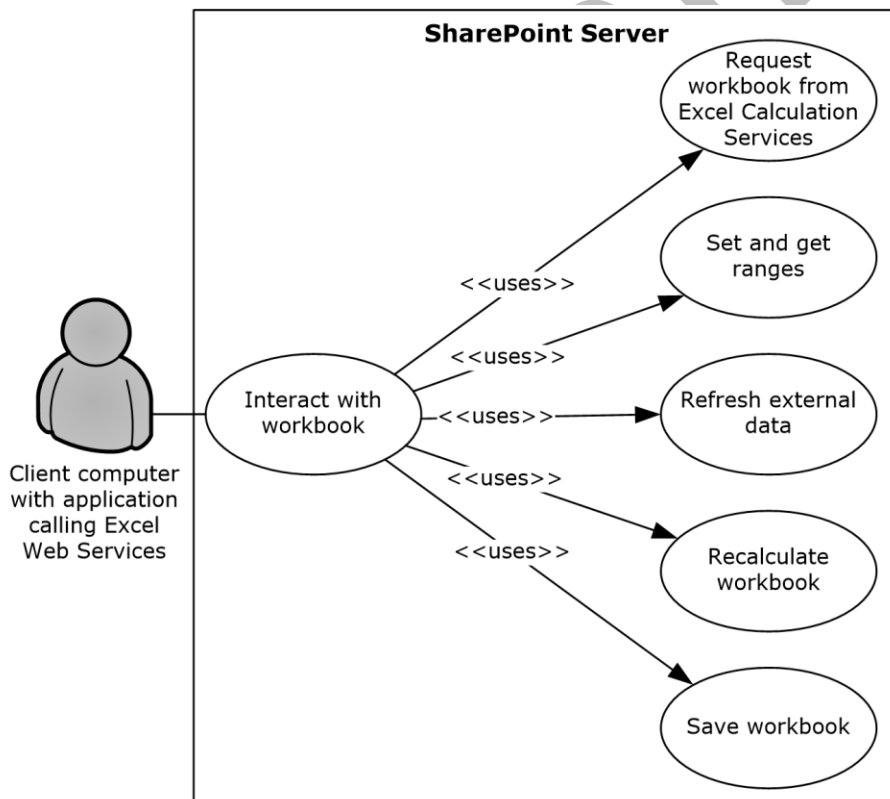
Figure 17: Process for interacting with a workbook by using an application

Preconditions

- Excel Services is enabled on a farm running SharePoint Products and Technologies.

- The user who is running the application has permissions for the requested workbook.

Steps

1. The application requests the workbook by using the URL of the file.

2. The application can get a range of data from the workbook.

3. The application can set a range of data in the workbook.

4. The application can refresh external data in the workbook.

5. The application can recalculate the workbook.

6. The application can save changes made to the workbook back to the file that is stored in SharePoint Products and Technologies.

7. The application can close the workbook.

Errors

- If the user who is running the application does not have permissions for the workbook, an error will be returned.

- If the workbook contains unsupported features that prevent it from opening, an error will be returned.

- If the workbook is connected to external data for which the current user does not have permissions, an error will be returned.

Post-conditions

- The application has retrieved any necessary data from the workbook.

- The workbook has been updated as requested by the application.

- The workbook has not been updated, because an error occurred.

2.5.2.2 Access Services

2.5.2.2.1 Publish a Database Application to SharePoint Products and Technologies

This use case describes how a client application **publishes** a database application to SharePoint Products and Technologies.

The following diagram illustrates this process.
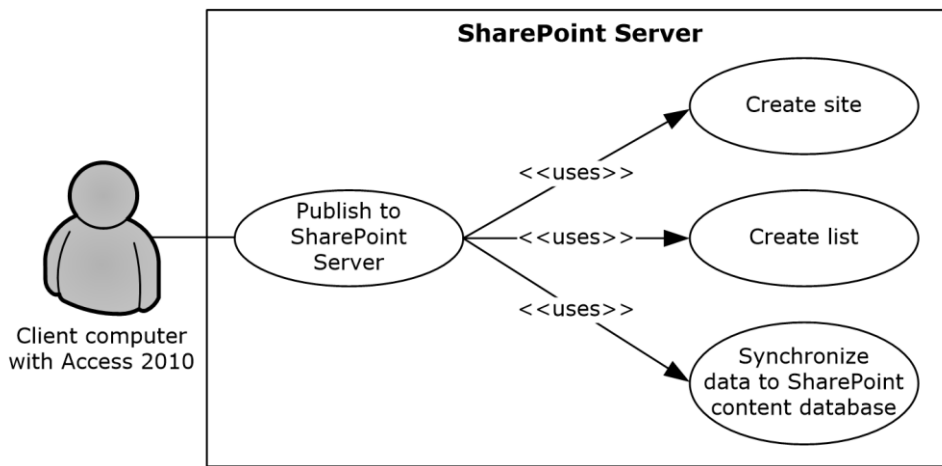
Figure 18: Process used to publish a database application to SharePoint Products and Technologies

Preconditions

- Access Services is enabled on a farm running SharePoint Products and Technologies.

- The user has built a database application in Access 2013.

- The user has been authenticated and the permission to create a **site**.

Steps

1. The user initiates publishing from Access 2013.

2. Upon a request from Access 2013, Access Services creates a SharePoint site according to the Access Services Site Template, as described in [MS-ASWS] section 3.1.1.1.

3. Upon a request from Access 2013, Access Services creates SharePoint **lists (1)** to represent tables in the Access 2013 database application.

4. Upon a request from Access 2013, Access Services synchronizes the database object structures.

5. Upon a request from Access 2013, Access Services synchronizes data from the tables to the SharePoint lists.

6. Upon a request from Access 2013, Access Services creates pages in SharePoint Products and Technologies for the **forms** and reports of the Access 2013 database application. Upon user request, the pages will be rendered in **HTML** in the browser.

Post-condition

- A SharePoint site that contains the published application has been created.

2.5.2.2.2 Interact with a Database Application that Has Been Published to SharePoint Products and Technologies

This use case describes how a client application interacts with a database application that has been published to SharePoint Products and Technologies.

The following diagram illustrates this process.
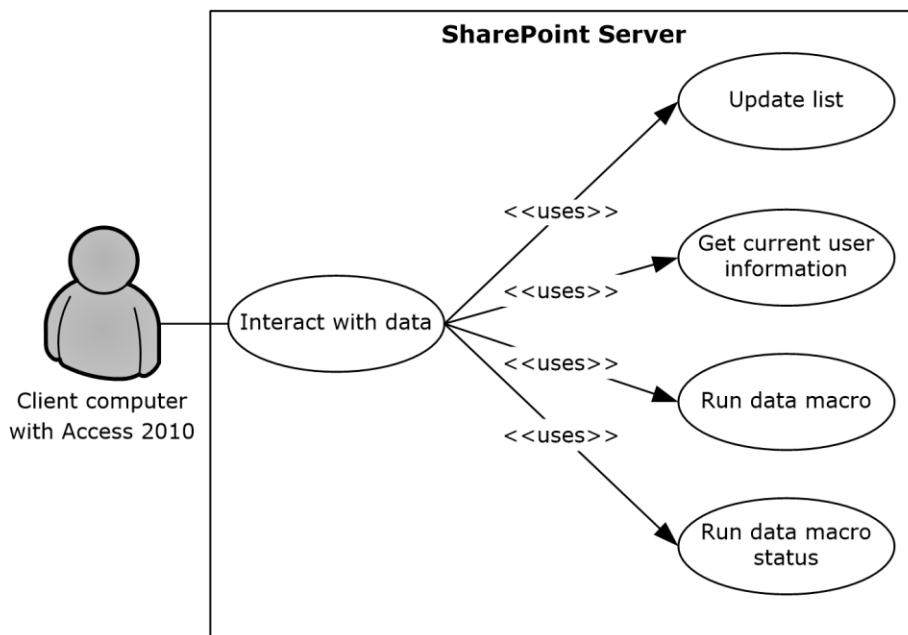
Figure 19: Process for interacting with a database application that has been published to SharePoint Products and Technologies

Preconditions

- Access Services is enabled on a farm running SharePoint Products and Technologies.

- A user can use Access 2013 to interact with a database application that has been published to SharePoint Products and Technologies.

Steps

1. After a database application created with Access 2013 has been published to SharePoint Products and Technologies, the data is stored in the content database. A user can still update the data by using Access 2013 through Web services, as described in [MS-ASWS]. Upon a request from Access 2013, Access Services updates the SharePoint **lists (1)** that store the data.

2. Access 2013 can retrieve the current user information from Access Services.

3. A **data macro** can be triggered from Access 2013. Access Services will run the data macro on SharePoint Products and Technologies and track the status of the data macro instance.

Error

- If the specified list (1) to be updated does not exist, the **server** will return an error.

Post-conditions

- The data operation that was performed in Access 2013 is synchronized to the SharePoint site that contains the application.

- The data macro that was triggered in Access 2013 is run on the farm running SharePoint Products and Technologies.

2.5.2.3 InfoPath Forms Services

2.5.2.3.1 Publish a Form Template and Enable Browser Rendering

This use case describes how a user both **publishes** a **form template** to a SharePoint Products and Technologies-hosted document library by using the InfoPath Form Designer client application, and enables browser rendering for that form template by means of InfoPath Forms Services. This allows other end users to fill out such a **form** using their Web browsers and submit that information back to the SharePoint Products and Technologies **server** for review, analysis or further action but the original publishing user.

The following diagram illustrates this process.
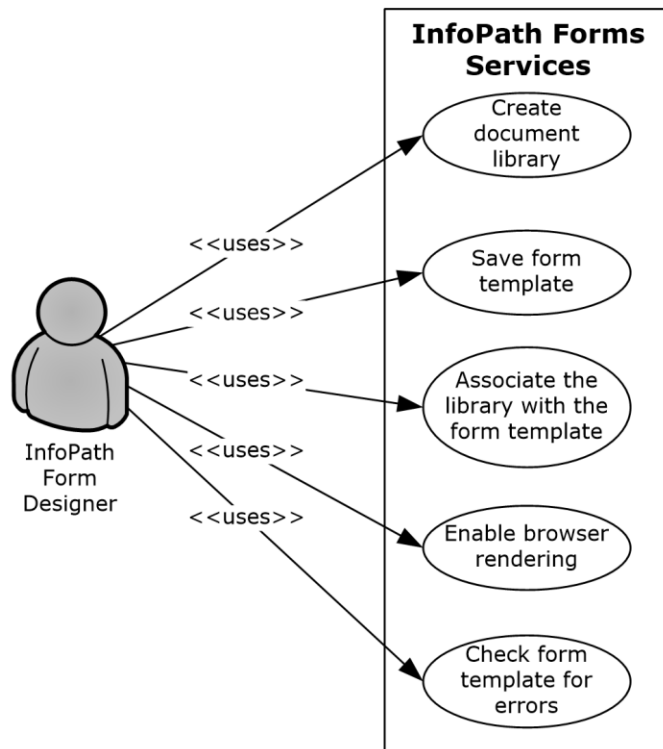
Figure 20: Process used to publish (1) a form template and enable browser rendering

Preconditions

- InfoPath Forms Services is enabled.
- The client application can communicate with the SharePoint site.
- The user has been authenticated as a member of the designer **group (1)**.

Steps

1. The user initiates publishing from InfoPath 2013.
2. Upon a request from InfoPath Designer, SharePoint Products and Technologies responds with information about the **site** that is provided by the user.
3. The user provides parameters such as a **form library** name.
4. Upon a request from InfoPath Designer, SharePoint Products and Technologies creates a document library.
5. InfoPath Designer saves the form template to the document library.
6. Upon a request from InfoPath Designer, SharePoint Products and Technologies associates the SharePoint library with the template.
7. Upon a request from InfoPath Designer, InfoPath Forms Services enables browser rendering for the form template (.xsn) file.

Error

- If an error occurs during the browser-enable process, InfoPath Designer will issue a call to check the form template for errors.

Post-conditions

- A published form template is accessible by using the Web browser at a URL that is associated with the document library.

- A published form template is inaccessible, because an error occurred during the browser-enable process.

2.5.2.4 User Profile Service

2.5.2.4.1 Retrieve a User's Profile

This use case describes how a client application can retrieve a user's profile, that is, all the information in the user's profile that is accessible to the client application. The information that is presented to the client application is based on the privacy level set on the user profile properties, either by the administrator as a default or by the user. For example, a user can set the privacy settings so that only colleagues will be able to see the user's birthday.

The following diagram illustrates the retrieval process.
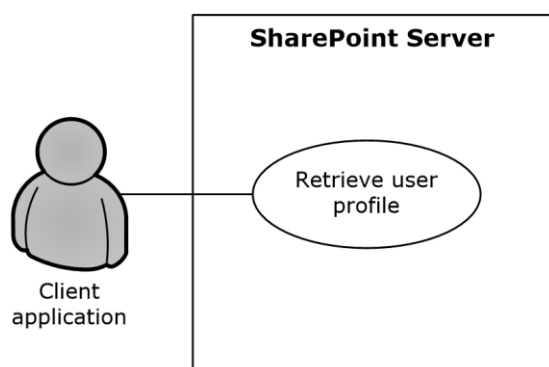
Figure 21: Process for retrieving a user's profile

Preconditions

- The client application has permissions for accessing and reading the user's profile.

- The requested user exists in the user profile store.

Steps

1. The client application requests access to a user's profile.

2. The User Profile Service retrieves the properties of the user profile, filtered by the privacy settings.

3. The **User Profile Service** returns the properties to the client application, formatted in the **site template** that has been applied to the user's personal profile **site**.

Error

- If the client application does not have access and read permissions for the user's profile, an access denied error will be returned.

Post-conditions

- The client application displays the user's profile, which typically includes a picture, information about the user, and any additional properties that have been entered and shared by the user.

- The user's profile is not displayed, because an access denied error was returned.

2.5.2.4.2 Tag a Document

This use case describes how a client application applies a tag to a document.

The following diagram illustrates this process.
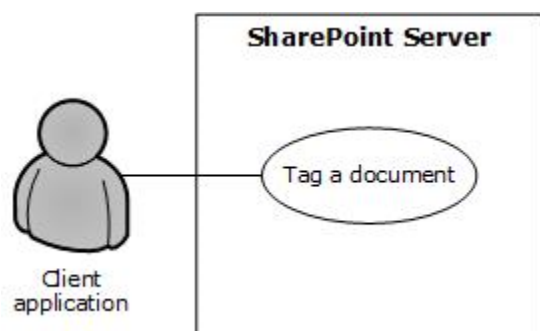
Figure 22: Process for tagging a document

Preconditions

- The client application has permission to access the location of the document.

- The client application has permission to apply tags by using the User Profile Service.

- The Content Management Service is provisioned and operational. For more information, see section 2.1.3.7.

Steps

1. The client application requests matching terms from the term store of the Content Management Service by providing the initial letters of the tag (for example, "soc" for "social computing" as an intended tag).

2. The Content Management Service returns a set of suggested terms to the client application, as described in section 2.5.2.7.1.

3. Upon the acceptance of a suggested term, a **GUID** for the term is stored in the **user profile store** along with the location of the document, information about the client application's identity, and the time of tag creation.

Error

- If the client application does not have permission to apply tags in the User Profile Service, tagging access will be disabled.

Post-conditions

- The client application detects the tag that is applied to the document location.

- The client application cannot tag a document, because tagging access was disabled.

2.5.2.5 Search Services

2.5.2.5.1 Issue a Search Query

This use case describes how a client application retrieves the search results for a **search query** that it issued against a SharePoint site. After the client application receives the results, common uses include displaying them in the UI and allowing the user to select a particular result to view more information.

The following diagram illustrates the query process.
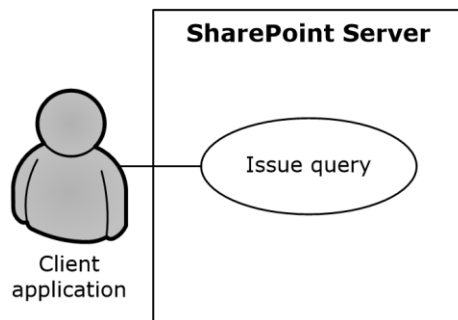
Figure 23: Process for issuing a search query

Preconditions

- The client application has permissions for accessing and executing queries against the SharePoint site.

- The SharePoint site is configured to use search services from a **search service application**.

Steps

1. The client application requests the results for a particular query.

2. The search service retrieves the URLs and properties that satisfy the query.

3. The search service returns the results to the client application in the format that is specified by the configuration of the Search Center.

Errors

- If the client application does not have permissions for the SharePoint site, an access denied error will be returned.

- If the SharePoint site is not set up to use the search services from a search service application, an error will be displayed in the UI when the client application issues a query.

Post-conditions:

- The client application displays the search results along with properties such as the URL, author, last modified date, file size, title, and summary.

- The search results are not displayed, because an error was returned.

2.5.2.6 Business Connectivity Services

2.5.2.6.1 Create an External Content Type

This use case describes how a client application creates an external content type and **publishes** to the **BDC metadata store** on SharePoint Server.

The following diagram illustrates this process.
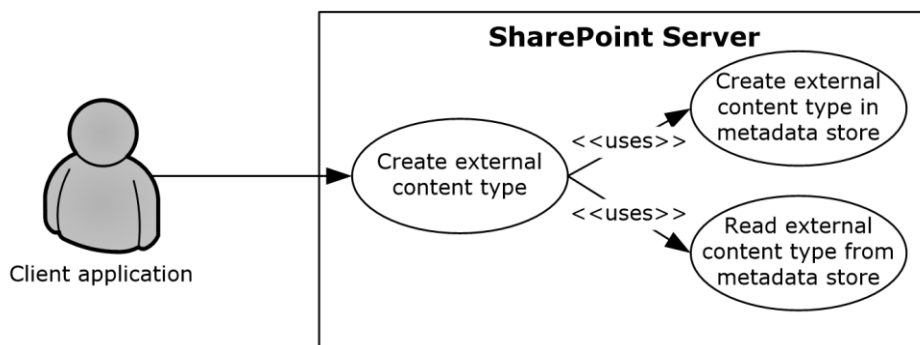
Figure 24: Process for creating an external content type

Preconditions

- The Business Data Connectivity service is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated as a member of the designer **group (1)**.

Steps

1. The protocol client calls the method to create an external content type.
2. The protocol server will establish a connection to the metadata store and create the external content type.
3. The protocol server will return the new external content type to the protocol client.

Errors

- If the client application does not have permissions for the SharePoint site, an access denied error will be returned.

- If the SharePoint site is not set up to use **Business Connectivity Services (BCS)** from a service application, an appropriate error will be returned.

- If the current user does not have permissions to update the BDC metadata store, an appropriate error will be returned.

Post-conditions

- The external content type is successfully published to the BDC metadata store.

- The external content type is not published to the BDC metadata store, because an error was returned.

2.5.2.6.2 Read an External Content Type

This use case describes how a client application reads an external content type from the **BDC metadata store** on SharePoint Server.

The following diagram illustrates this process.
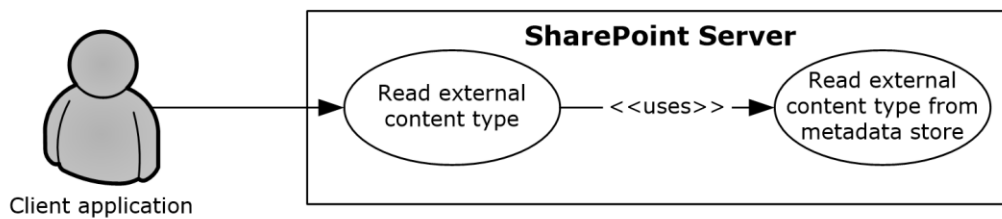
Figure 25: Process for reading an external content type

Preconditions

- The BDC service is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated as a member of the designer **group (1)**.

Steps

1. The protocol client calls the method to read an external content type.
2. The protocol server will establish a connection to the metadata store and read the external content type.
3. The protocol server will return the external content type to the protocol client.

Errors

- If the client application does not have permissions for the SharePoint site, an access denied error will be returned.

- If the SharePoint site is not set up to use **BCS** from a service application, an appropriate error will be returned.

- If the current user does not have permissions to access the BDC metadata store, an appropriate error will be returned.

Post-conditions

- The external content type is read and returned to the protocol client.

- The external content type is not read or returned to the protocol client, because an error was returned.

2.5.2.6.3 Update an External Content Type

This use case describes how a client application updates an external content type store in the **BDC metadata store** on SharePoint Server.

The following diagram illustrates this process.
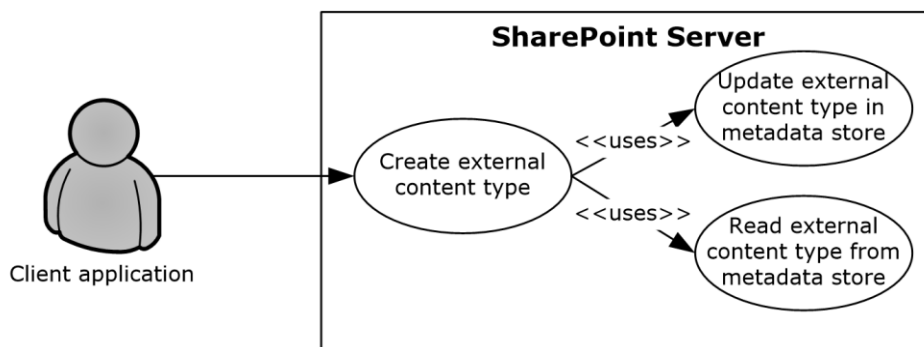
Figure 26: Process for updating an external content type

Preconditions

- The BDC service is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated as a member of the designer **group (1)**.

Steps

1. The protocol client initiates a request for reading an External Content Type by calling the update method.
2. The protocol server will establish a connection to the metadata store and update the external content type.
3. The protocol server will return the updated external content type to the protocol client.

Errors

- If the client application does not have permissions for the SharePoint site, an access denied error will be returned.

- If the SharePoint site is not set up to use **BCS** from a service application, an appropriate error will be returned.

- If the current user does not have permissions to update the BDC metadata store, an appropriate error will be returned.

Post-conditions

- The external content type is successfully updated in the BDC metadata store.

- The external content type is not updated in the BDC metadata store, because an error was returned.

2.5.2.6.4 Deploy a Business Connectivity Services Solution

This use case describes how a client application can synchronize with a **BCS** solution **published** in the form of a **list (1)** on SharePoint Server.

The following diagram illustrates this process.
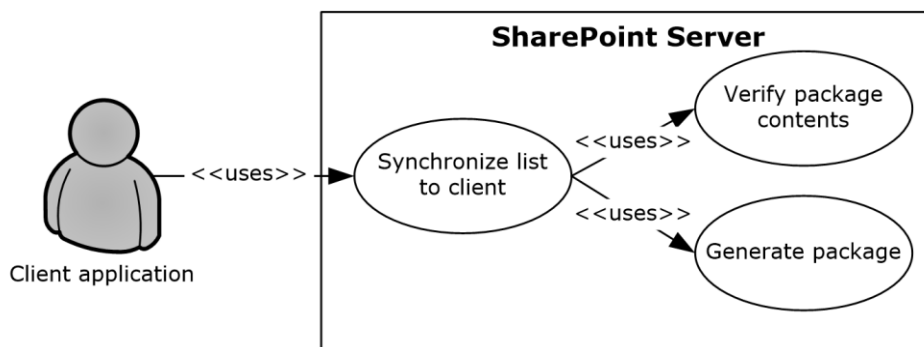
Figure 27: Process for deploying a Business Connectivity Services solution

Preconditions

- BCS is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated and has read access.

Steps

1. The protocol client initiates a request for synchronizing the list (1) with a client application.
2. The protocol server will verify the package existence and will verify that the external content type and the related external content types are current.
3. If they are not current, the protocol server will generate a new package with the current external content type and the related external content types.
4. The protocol server will return the package URL and the solution identifier to the protocol client.

Error

- If an error occurs during the package generation process, the protocol server will return an appropriate error code to the protocol client.

Post-conditions

- The solution package is accessible to the protocol client from a URL.
- The solution package is inaccessible to the protocol client from a URL, because an error occurred during the package generation process.

2.5.2.6.5 Update a Business Connectivity Services Solution

This use case describes how a client application can update a **BCS** solution **published** in the form of a **list (1)** on SharePoint Server.

The following diagram illustrates this process.
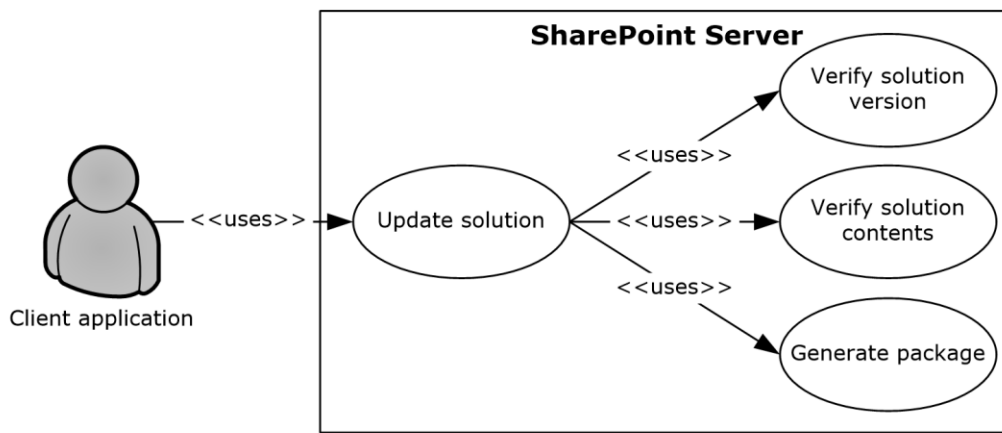
Figure 28: Process for updating a Business Connectivity Services solution

Preconditions

- BCS is enabled on the SharePoint site.
- The client application can communicate with the SharePoint site.
- The user has been authenticated and has read access.

Steps

1. The protocol client initiates a request for updating a solution.
2. The protocol server will verify the package existence and will verify that the external content type and the related external content types are current.
3. If they are not current, the protocol server will generate a new package with the current external content type and the related external content types.
4. The protocol server will return the package URL and the solution identifier to the protocol client.

Error

- If an error occurs during the package generation process, the protocol server will return an appropriate error code to the protocol client.

Post-conditions

- The solution package is accessible to the protocol client from a URL.
- The solution identifier returned to the protocol client is different, indicating a newer version of the solution is available.
- The solution package is not updated, because an error occurred during the package generation process.

2.5.2.7 Content Management Service

2.5.2.7.1 Get Term Set Data and Add a Term

This use case describes how a client application first gets the data for a **term set** from a **term store** that is managed by a protocol server and then adds an additional **term** to that term set. After a client application gets the data, other common uses include displaying the term set and validating user-supplied data against the term set.

The following diagram illustrates the get and add processes for a term set.
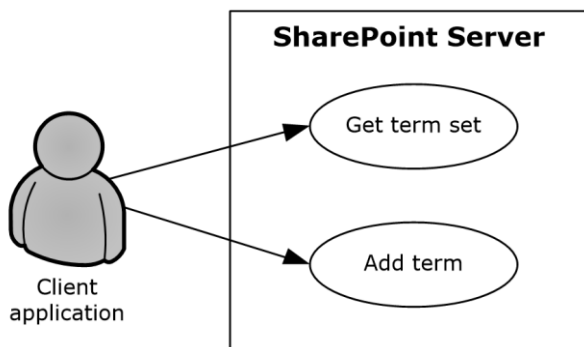
Figure 29: Process for getting term set data and adding a term

Preconditions

- The client application can communicate with the term store.

- The client application has been authenticated, has read permission and has write permission for the term store.

- The client application has the identifiers for the term sets in the term store.

Steps

1. The client application requests the term set from the term store.

2. The **server** returns the term set.

3. The client adds a term to the term set in the specified term store.

4. The server returns the added term.

Error

- If the term set is not open, the term store will not create a new term.

Post-conditions

- A new term exists in the term set.

- A new term is not added to the term set, because the term set is not open.

2.5.2.8 PerformancePoint Services

2.5.2.8.1 Interact with Published Dashboards on a Web Part Page

This use case describes how a client application interacts with **published dashboards** on a Web Part page in SharePoint Products and Technologies. A user can analyze business data (if the user has the appropriate permissions for the SharePoint **list (1)** in which that data is stored); comment on the data; filter or run calculations on the data according to business rules; and view the data through the **decomposition tree**, which provides a visual view of how the data unfolds from a rolled-up **OLAP hierarchy** (as described in [MS-PPSDECO]).

The following diagram illustrates the process of interacting with published dashboards.
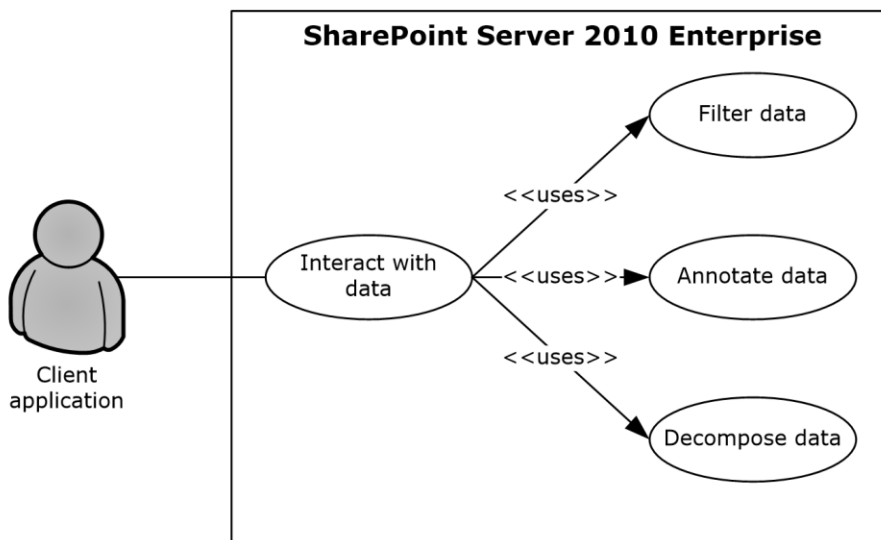
Figure 30: Process for interacting with dashboards on a Web Part page

Preconditions

- PerformancePoint Services 2010 has been set up, and the **site collection** level feature has been started in SharePoint Server 2013.

- A dashboard from PerformancePoint Services 2010 has been published to SharePoint Server 2013.

- Microsoft Silverlight version 3.0 has been installed on the client computer. For more information, see [MS-SLXV].

Steps

1. The browser on the client computer requests a published dashboard from Microsoft SharePoint Server 2010 Enterprise.

2. PerformancePoint Services 2010 retrieves the requested objects from the SharePoint list (1) service.

3. To return data to the rendered dashboard, PerformancePoint Services 2010 executes a data source query for each object that requires one.

4. The user drills through the rendered data, sending an **Asynchronous JavaScript + XML (AJAX)** request for each data request that the browser initiates.

5. In **scorecard** objects, the user can right-click a cell and comment the value in the cell.

6. In charts and analytic grids, the user can analyze data in a decomposition tree that is rendered in Silverlight.

Error

- If a user does not have access to a dashboard item (for example, the analytic grid, chart, or scorecard) in the SharePoint list (1), that user will receive an error in the Web Part zone of the Web Part page.

Post-conditions

- The user views the result of the dashboard that was acted upon.

- The dashboard result cannot be viewed, because the user does not have access to the dashboard item.

2.5.2.9 Translation Services

2.5.2.9.1 Machine Translate SharePoint Documents

This use case describes how a client application interacts with SharePoint to translate documents stored in document libraries.

The following diagram illustrates the process of translating documents.
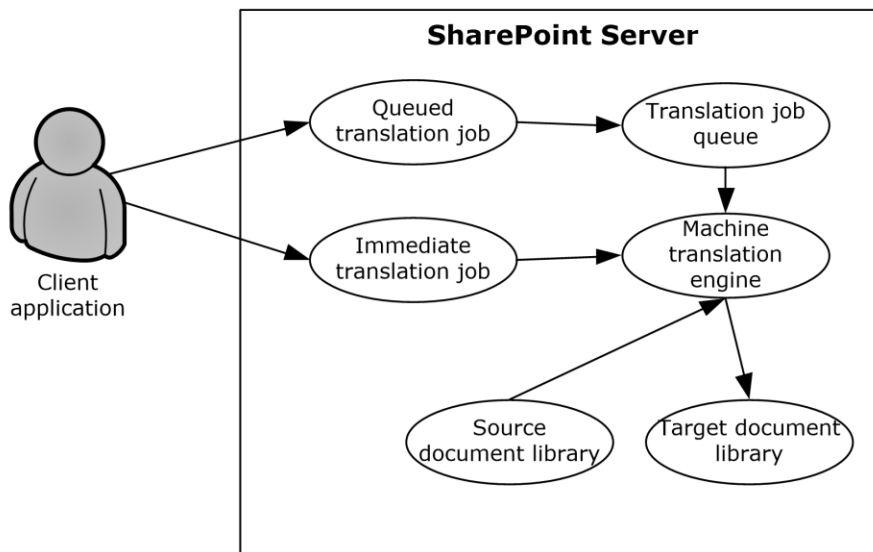
Figure 31: Process for translating documents in a document library

Preconditions

- Translation Services has been set up.
- A SharePoint site with source and target document libraries has been set up.
- The source library contains files in a format supported by Translation Services.

Steps for a queued translation job

1. The client application submits a **translation job** with several documents from the source library.
2. The translation job is stored in the job queue.
3. When Translation Services has capacity, the translation job is fetched from the queue and sent to the machine translation engine.
4. The machine translation engine downloads documents from the source document library.
5. The machine translation engine translates the documents.
6. The machine translation engine stores the translated documents in the target document library.

Steps for an immediate translation job

1. The client application submits an **immediate translation job** with a single document from the source library.

2. The machine translation engine downloads the document from the source document library.

3. The machine translation engine translates the document.

4. The machine translation engine stores the translated document in the target document library.

Error

- If the user that initiated the translation job does not have read access to the source document library or does not have write access to the target document library, the translation will fail.

Post-conditions

- The target document library contains machine translated versions of documents from the source document library.

2.5.2.10 Work Management Service

2.5.2.10.1 Retrieving a User Task List

This use case describes how a client application retrieves a list of **tasks** assigned to a user. The items returned are loaded from a cache of tasks aggregated across the system and external task providers.

The following diagram illustrates the retrieval process:
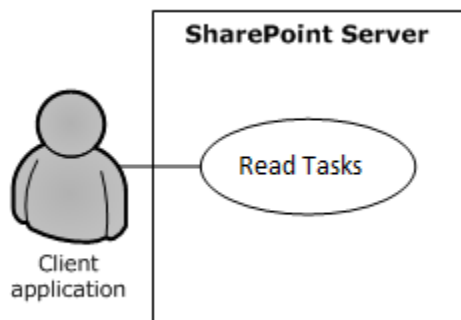
Figure 32: Process for retrieving a user's task list

Preconditions

- The Work Management Service has been set up.
- The User Profile Service has been set up.
- The client application has been authenticated, has read permission and has write permission for his or her **personal site** associated with the User Profile Service, where tasks are cached.
- The client application can communicate with his or her SharePoint site.

Steps

1. The client application requests the tasks from the **server**.

2. The server returns the tasks.

Error

- If the user does not have permissions to the task cache, this will return an error.

Post-conditions

- None; this is a read operation.

2.5.2.10.2   Updating a Task

This use case describes how a client application updates a **task** from the user's task cache and the original source task corresponding to the cached copy.

The following diagram illustrates the update process:
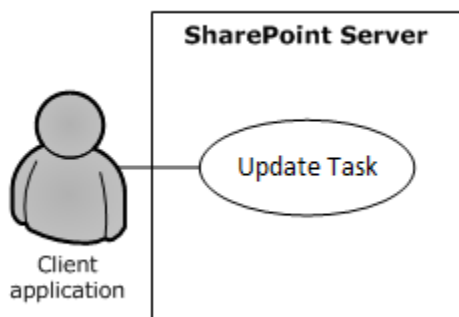
Figure 33: Process for updating a task on the user's task list

Preconditions

- The Work Management Service has been set up.
- The **User Profile Service** has been set up.
- The client application has been authenticated, has read permission and has write permission for his or her personal SharePoint site associated with the User Profile Service, where tasks are cached.
- The client application can communicate with his or her SharePoint site.

Steps

1. The client application requests an update to a task on the **server**.
2. The server updates the task in the task cache and original source location with the values provided in the request.
3. The server returns information about whether the update was successful or not.

Error

- If the user does not have write permissions to the task cache or original source item, this will return an error.

Post-conditions

- Fields on the task will be updated with new values.
- Fields on the task will remain the same if the user did not have sufficient permissions.

2.5.2.10.3   Changing the Order of a Task

This use case describes how a client application changes the position of a **task** relative to other tasks in a user-defined sort order. This sort order affects the order in which tasks are returned depending on the query for reading tasks.

The following diagram illustrates the retrieval process:
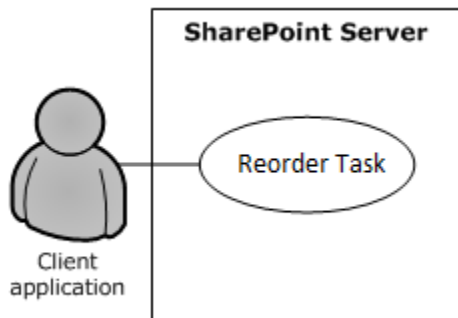
Figure 34: Process for reordering a task on the user's task list

Preconditions

- The Work Management Service has been set up.

- The **User Profile Service** has been set up.

- The client application has been authenticated, has read permission and has write permission for his or her **personal site** associated with the User Profile Service, where tasks are cached.

- The client application can communicate with his or her SharePoint site.

Steps

1. The client application requests to change the position of a task on the **server**.

2. The server updates the position of the task in the task cache.

Error

- If the user does not have write permissions to the task cache, this operation will fail.

Post-conditions

- The position of a task will be updated.

- The position will remain the same if the user did not have sufficient permissions to edit the task cache.

2.6   Versioning, Capability Negotiation, and Extensibility

The SharePoint Products and Technologies front-end and back-end **servers** perform explicit version verifications.

2.7   Error Handling

The SharePoint Front-End Protocols system does not handle errors at the system level for cross-protocol error states. The individual protocol documents describe the errors that the protocols return and what they mean for the system. How to handle the errors, based on the protocol descriptions, is determined by the implementer.

2.8 Coherency Requirements

This system has no special coherency requirements beyond the specific details that are covered by the individual protocol documents.

2.9 Security

A detailed overview of **authentication**, authorization, and the core concepts that are used to manage security in SharePoint Products and Technologies is available.

2.10 Additional Considerations

There are no additional considerations.

3 Examples

The examples in the following sections provide more information about the use and operation of the SharePoint Front-End Protocols system, especially interactions between system components. Protocol-level examples can be found in the individual protocol documents. The following system-level examples are provided in this document:

- Create a new **site** and access **list (1)** data

- Create a file from a client computer

These examples are subject to the following, possible common errors:

- Client unable to connect to front-end **server**.

- Front-end server unable to connect to Active Directory Domain Services (AD DS) to authenticate the user.

- Front-end server unable to connect to **back-end database server**.

3.1 Example 1: Create a New Site and Access List Data

This example describes how to create a new **site**, create a new **list (1)**, and access the list (1) data. This example builds on the use cases covered in "Create a Site" (section 2.5.1.2), "Create a List and Add a View" (section 2.5.1.3), and "Get List Data" (section 2.5.1.4). In this example, a client application creates a new subsite, creates a new list (1) on that subsite, populates the list (1) with data, and gets a filtered set of **list items**.

This scenario assumes that:

- The client application can communicate with the SharePoint site.

- The client application has permission to create a subsite, create a list (1), read list items, and write list items.

The following steps illustrate the site creation and list access processes, assuming that no errors occur.

1. The client application calls the method in the Site Web Service Protocol (as described in [MS-SITESS]) that retrieves the list of available **site templates** and **site definitions**.

2. The **server** returns the collection of available site templates and site definitions.

3. The client application calls to the method in the Site Web Service Protocol that creates a subsite.

4. The server returns the URL to the newly created subsite.

5. The client application uses the URL of the newly created subsite to call the method in the Webs Web Service Protocol (as described in [MS-WEBSS]) that gets the collection of **list templates** for the subsite.

6. The server returns the collection of list templates for the new subsite.

7. The client application calls the method in the Lists Web Service Protocol (as described in [MS-LISTSWS]) that adds a list (1).

8. The server returns the properties and fields of the new list (1).

9. The client application calls on the method in the Lists Web Service Protocol that adds items to a list (1).

10. The server returns the list items that were added to the list (1).

11. The client application calls the method in the Lists Web Service Protocol that gets list items, and passes a custom query (as described in [MS-WSSCAML] section 2.2).

12. The server returns the requested list items.

The following figure shows the sequence of steps for this example.
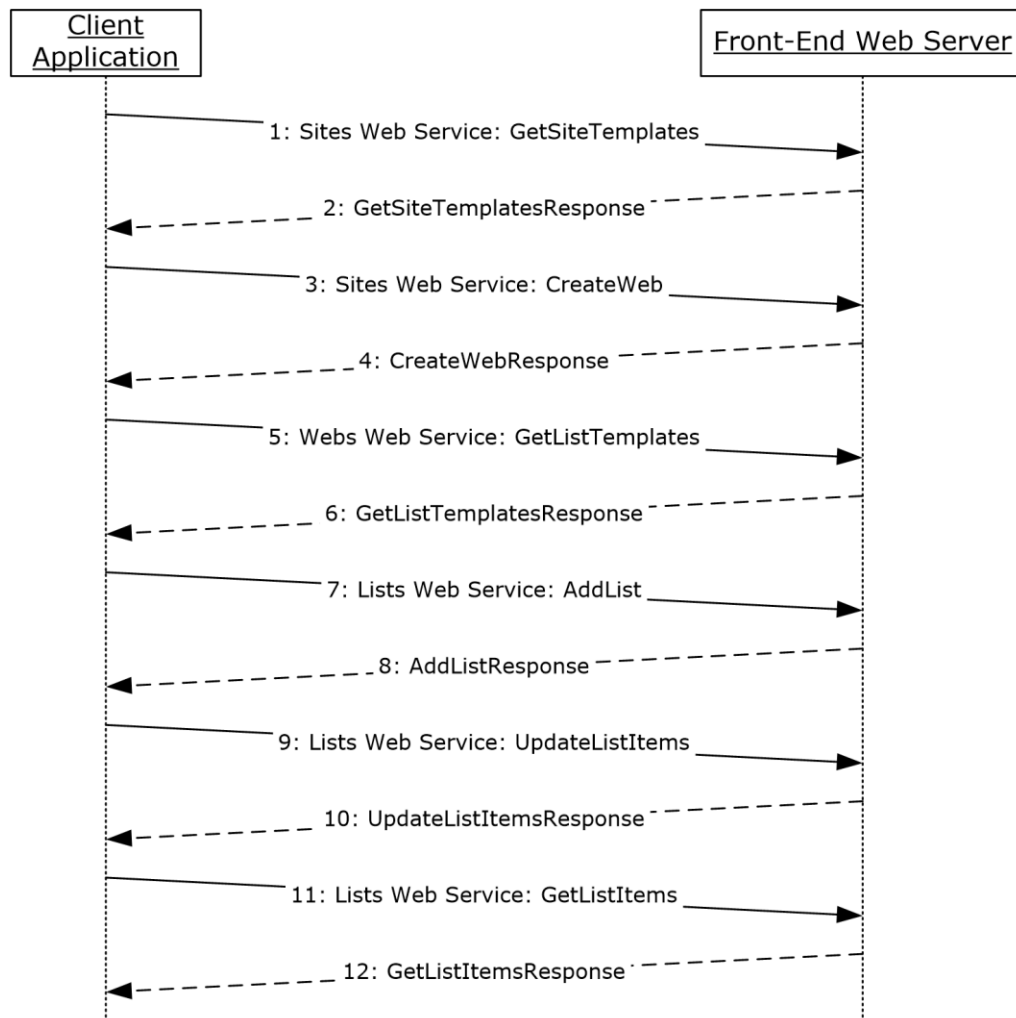
Figure 35: Sequence for creating a new site and accessing list data

3.2   Example 2: Create a File from a Client Computer

This example describes how to create a file from the client by using the protocols covered in this system. This example uses the "Create a File in a SharePoint Library from a Client Computer" use case that is described in section 2.5.1.5.

It helps to provide an understanding of how user interaction with the SharePoint Products and Technologies protocols involves communication with the back-end **servers**.

This example assumes the following:

▪ The user has permissions to an existing SharePoint library named "http://server/site/doclib".

- The user is logged on to a client computer running Windows 7 operating system (other client versions will have different steps), with an authenticated Microsoft Windows session, and can access the Windows SharePoint Services site containing the library.

- Using a Windows command line, the user begins by typing the following command:

    echo hello >\\server\site\doclib\hello.txt

The following steps illustrate the file creation process by a user using a Windows command line, assuming that no errors occur.

Some of the following steps use the SharePoint Products and Technologies programming API; for more information about these APIs, see [MSDN-SharePointSDK].

To highlight a specific case, step 3 in this example is created by calling the SharePoint Products and Technologies programming API: SPGroup.Users.Add(). This example is called out because the actual steps generated between the **front-end Web server** and **back-end database server** might be different when the request is initiated by user interaction with the front-end Web server using a browser or when using the Windows command line, as in this case.

1. After the user initiates the **echo** command, the client sends a WebDAV request (as described in [RFC2518]) to the front-end Web server, asking it to perform a PUT operation on the hello.txt file in the document library.

2. The front-end Web server (IIS) authenticates the user with AD DS. In practice, this might involve multiple LDAP requests with AD DS, especially if the user has not visited the site before.

3. AD DS responds with multiple LDAP results.

4. In multiple round trips with the back-end database server, the front-end Web server locates the content database for the document library and confirms that the library exists.

5. The back-end database server returns multiple objects for the **site collection**, **website**, and library to the front-end Web server.

6. In multiple round trips with the back-end database server, the front-end Web server creates an empty file in the document library, and then, if successful, also verifies that the user has permissions to access and write to the document library.

7. The back-end database server returns multiple result sets as part of the process to create the file.

8. The front-end Web server returns a WebDAV response, saying the file was created successfully.

9. The client sends a WebDAV HEAD request to the front-end Web server, with the URL to the hello.txt file in the document library, to verify the success of the previous call.

10. In multiple round trips with the back-end database server, the front-end Web server retrieves the file.

11. The back-end database server returns multiple result sets as part of the process to retrieve the file.

12. In response to the HEAD request, the front-end Web server sends a response saying the request was successful.

13. Then the client sends a WebDAV PUT request to the front-end Web server, which will have multiple parts to upload the file and update its properties.

14. The client sends a WebDAV request to the front-end Web server with an **XML** payload that has the file properties from the client.

15. The client sends a WebDAV request to the front-end Web server with an XML payload that has the file content; in this example, that content is simply the word "hello".

16. In multiple round trips with the back-end database server, the front-end Web server updates the file and its properties in the document library.

17. The back-end database server returns multiple result sets as part of the process to update the files.

18. On completing the update, the front-end Web server sends a WebDAV response, saying the request was successful.

The following sequence diagram illustrates the message sequence for this example. It shows the consolidation of multiple front-end Web servers to back-end database server actions, and multiple front-end Web servers to AD DS actions and into single flows.

This diagram does not document some of the initial interactions between client and server that optionally happen on some clients and that can also depend on whether the client has connected to the **site** previously to verify whether the server is able to support WebDAV.
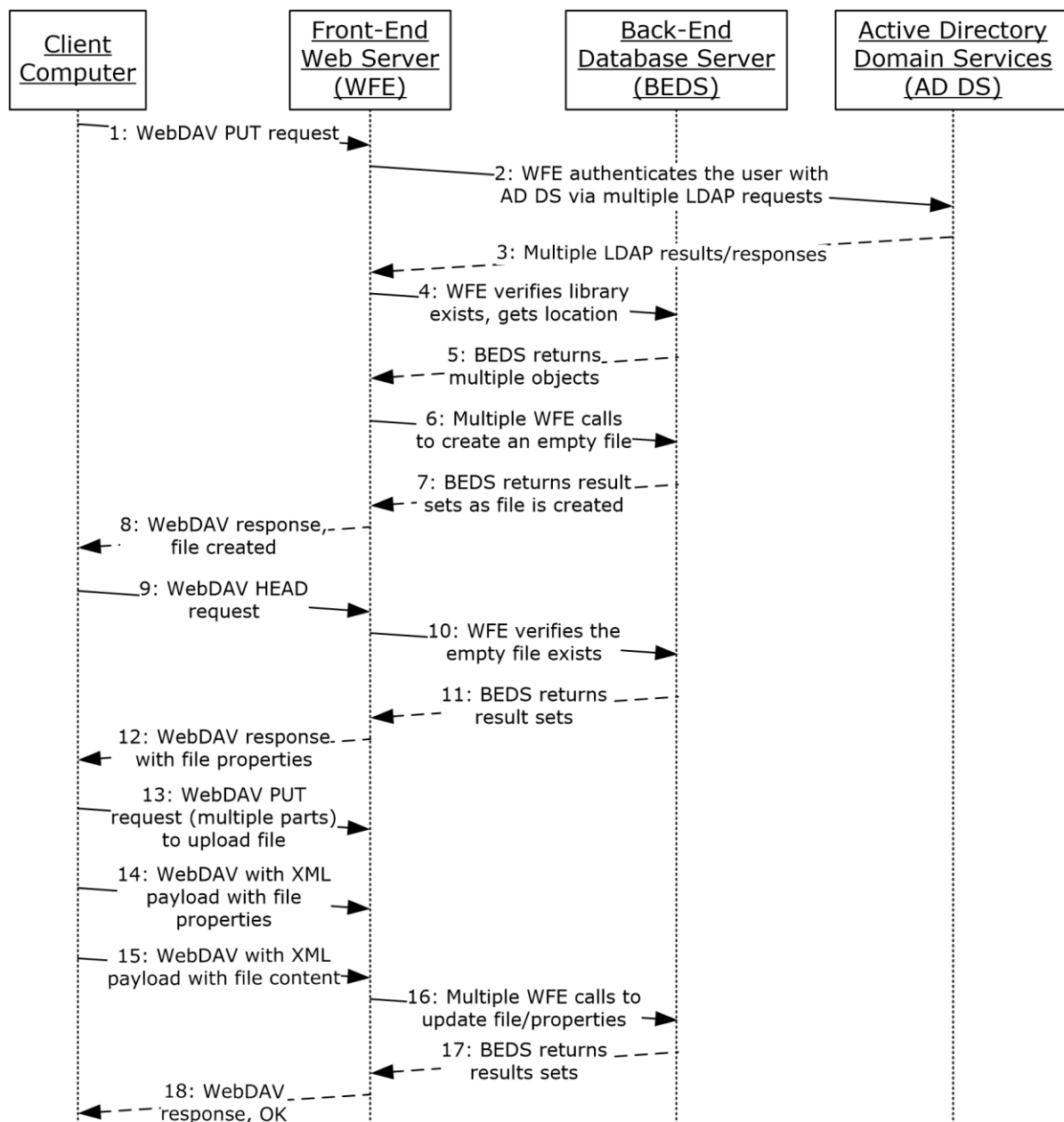
Figure 36: Sequence for creating a file from the client

4 Microsoft Implementations

There are no variations in the behavior of the SharePoint Front-End Protocols system in different versions of Windows SharePoint Services and Microsoft SharePoint Server beyond those described in the specifications of the protocols supported by the system, as listed in section 2.2.

The information in this specification is applicable to the following versions of Windows SharePoint Services and SharePoint Server:

- Microsoft Office SharePoint Portal Server 2003

- Windows SharePoint Services 2.0

- Microsoft Office SharePoint Server 2007

- Microsoft Office SharePoint Server 2007 CHT, CHS, KOR, JPN LANGPAK

- Microsoft Office SharePoint Designer 2007

- Windows SharePoint Services 3.0

- Microsoft SharePoint 2007 Products and Technologies

- Microsoft FAST Search Server 2010 for SharePoint

- Microsoft FAST Search Server 2010 for SharePoint Internet Sites

- Microsoft SharePoint Workspace 2010

- PerformancePoint Services in Microsoft SharePoint Server 2010

- Microsoft SharePoint Designer 2010

- Microsoft SharePoint Designer 2010

- Microsoft SharePoint Server 2010

- Microsoft SharePoint Server 2010

- Microsoft SharePoint Server 2010 JPN, KOR, CHT, CHS LANGPAK

- Microsoft SharePoint Server 2010 Enterprise

- Microsoft SharePoint 2010 for Internet Sites Enterprise

- Microsoft SharePoint 2010 for Internet Sites Standard

- Microsoft SharePoint Foundation 2010

- Microsoft SharePoint 2010 Products and Technologies

- Microsoft SharePoint Designer 2013

- Microsoft SharePoint Server 2013

- Microsoft SharePoint Foundation 2013

- Microsoft SharePoint Server 2019

- Microsoft SharePoint Server Subscription Edition Preview

Exceptions, if any, are noted in the following section.

4.1 Product Behavior

5 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
All	Clarified applicability in later versions of SharePoint Products and Technologies.	Minor
4 Microsoft Implementations	Updated list of supported products.	Major

6 Index