1. **Main Benefits of SharePoint 2013.**
   1. Share - storing and sharing documents, contacts, and tasks; organizing meetings; managing business processes. Share information to social network, like Facebook, twitter.
   2. Organize - help you manage tasks, as well as their status and due dates. Keep team connected through shared information.
   3. Discover – search engine.
   4. Build – App and Cloud.
   5. Manage – Office 365
2. **Logical and physical architecture**

**Front-end web servers** - These servers publish websites, often called web applications.

**Application servers** - These servers host back-end services, such as Search services, the User Profile service, Excel Services, and so forth.

**Database servers** - These servers store configuration and content data for the entire SharePoint farm.

1. **Service Application**

Service applications are software service run in a SharePoint farm. One service application can publish to multiple servers in a farm. Service applications are intended for sharing resources and capabilities across multiple sites and servers in the same farm, or even across farms.

1. **Feature receivers**

• Feature activation

• Feature deactivating

• Feature installation

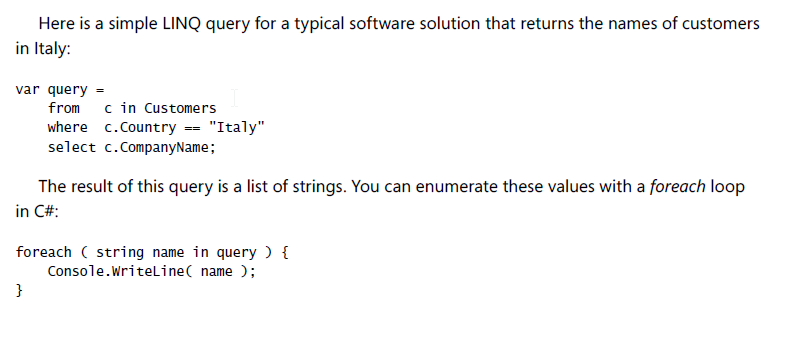
• Feature uninstalling

• Feature upgrading

1. **LINQ (Language-Integrated Query)**

LINQ is a programming model that introduces queries as a first-class concept into any Microsoft .NET language.

LINQ makes it easier to access all these various kinds of data (such as an array, an object graph, an XML document, a database, a text file, a registry key, an e-mail message, Simple Object Access Protocol (SOAP) message content, a Microsoft Office Excel file), providing a unified programming model



1. **Server-side technologies**

Server Object Model

LINQ

1. **Client-side technologies**

Client Object Model

The REST API (Representational State Transfer)

1. **The Rest API (Representational State Transfer)**

Use unique URL to query, update, delete the resource of SharePoint server.

Demo:

http://devbook.sp2013.local/\_vti\_bin/ListData.svc/DevLeapContacts(1)

1. **SharePoint APP**
   1. **App Type**

Full Page APP

App Parts

UI command extension

* 1. **Hosting Models**

**SharePoint-hosted** – On-premised host. You can take advantage of all the features of SharePoint, such

as lists, Web Parts, pages, workflows, and so on.

**Auto hosted Apps** - following this model are hosted on Microsoft Windows Azure, which can access a Microsoft SQL Azure database for managing data, too.

**Provider-hosted** - this model are almost the same as auto hosted apps. The only difference is that a provider-hosted app has to be deployed on your own hosting environment and does not necessarily use the Windows Azure environment.

Regardless of the hosting model and configuration, every SharePoint app is mainly a web application that interacts with SharePoint using the Client Object Model and the new REST API introduced with SharePoint 2013 (which is covered in Chapter 9, “The new SharePoint REST API”). One key feature of SharePoint apps is that they can be developed with any programming language or technology, as long as you host them outside SharePoint (that is, using an auto hosted or provider-hosted model). In fact, you can create a SharePoint app using PHP, Java, or any other technology capable of communicating with SharePoint via the new REST API and the OAuth protocol.

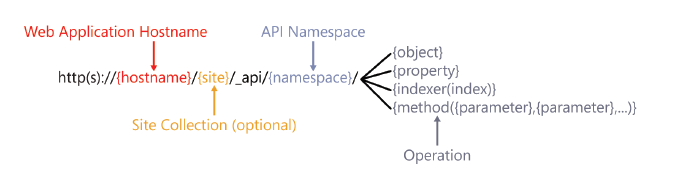
* 1. **You can convert a site to SharePoint APP**

P287

* 1. **You can use DB of Azure**

P289

1. **REST API schema**

****

1. **Remote event receivers**

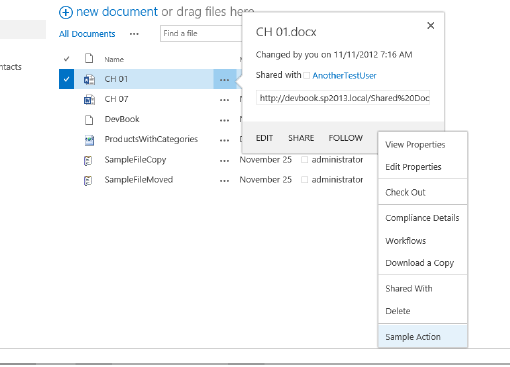
The architecture of remote event receivers uses SOAP as the protocol for communicating across the wire, because it is the most open, standard protocol for implementing cross-platform dialogs.

* 1. **Scopes and types of receivers**

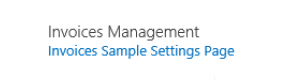
Scope: Single list item, List, Website, APP

1. **Web Parts**
   1. **Classic Web Parts**
   2. **Visual Web Parts**
2. **Customizing the UI**
   1. **Custom Actions**

**■CustomAction -** Creates a new custom action to define a new control on a ribbon, a new menu item on a standard menu, or a new link on a settings page

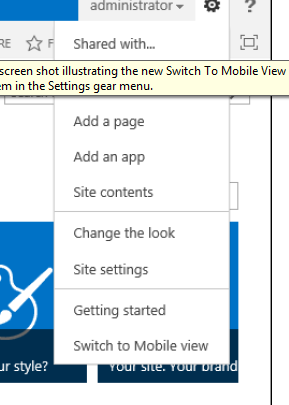
****

**■CustomActionGroup -** Creates a new group of custom actions for better usability from the perspective of the end user

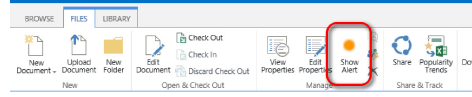
****

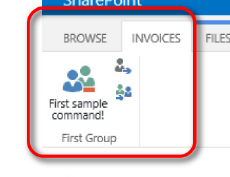
**■HideCustomAction -** Hides an existing custom action defined by another custom action or implemented by default in SharePoint

**■Server-side custom actions**

****

* 1. **Custom Ribbons**

****

****

* 1. **Custom Content**

**Images and generic content.**

**Application pages.**

**Content pages, web part pages and galleries.**

* 1. **Status bar and notification**
  2. **Dialog Framwork**

1. **Web Templates**

**The Core Techniques**

* 1. **Site Definitions**

A site definition is a site model defined on the file system and stored in the folder SharePoint15\_Root\TEMPLATE\SiteTemplates of every front-end server. Saving files on the file system is an ancient habit that requires direct access to the servers in your server farm. Thus, it is not suitable for Microsoft Office 365 or SharePoint Online.

* 1. **Feature Stapling**

Feature stapling is a technique that enables you to customize existing site definitions by adding custom features to extend the site definition.

* 1. **Site Templates**

A site template is an exported snapshot of an already existing site instance, with or without its content. You can use a site template, which is basically just a WSP package, to replicate a site instance from one environment to another, as long as the base site definition is available on the target environment, too. In addition, you can import site templates into Microsoft Visual Studio 2012 to create custom template projects. To save a site template from an already existing site instance, simply click Save Site As Template in the Site Actions menu group on the Site Settings page. The one exception, however, is that a site template cannot be created from a site in which publishing features are enabled, so it’s only usable in team and collaboration sites.

* 1. **Web Templates**

A web template is a specific Web Template feature that you can create using Microsoft Visual Studio 2012 and provision using a sandboxed solution. A web template enables you to define a custom site model for future reuse. You can deploy web templates at the site collection level, through a sandboxed solution, or at the farm level using a full trust WSP solution package. Because web templates can use a sandboxed solution for deployment, you can use them against Office 365 and SharePoint Online, too.

* 1. **Modules**

Modules for uploading files.

1. **Business Connectivity Services**

Business Connectivity Services (BCS) is a fundamental service application of Microsoft SharePoint 2013. It provides capabilities to read and write data from external systems, such as line-of-business (LOB) applications, web services, databases, and any other external sources that offer a suitable connector.

In this way, we could use the native list to access to external data directly without coding.

1. **Workflow**

Every workflow is made of a set of activities or at least a single activity. An activity is the smallest unit of execution for a workflow and can be considered a single step of a workflow. Activities can be created by developers, or can be taken from the list of activities already available out of the box.

* 1. **Sequence**

Represents a sequential workflow in which you define an explicit entry point (start) and a well-known exit point (end). The workflow instances will be executed from the start to the end, stepping through multiple steps and following some branches, without the capability to step backward through the flow. The only way to step backward in a sequential workflow is to define multiple nested loops, but doing so would result in overly complex and difficult-to-manage workflows.

* 1. **Flowchart**

Defines a flow that behaves like a flow diagram; it can be traversed from start to finish, and specific conditions and rules can return the flow to previously completed steps.

* 1. **State machine**

Defines a state machine flow, which is made of a set of states and rules to transfer the flow. It is the most suitable solution for implementing human-interactive flows, because end-user behavior is unpredictable and nondeterministic.

All of the above models are made up of various activities. You can mix the three models within a single workflow definition.

1. **Authentication and authorization infrastructure**
   1. **Authentication infrastructure**

Uses the Windows infrastructure, providing support for NTLM, Kerberos, Anonymous, Basic, and Digest authentication. X.509 Certificate Authentication is not supported, unless you manually configure users’ certificate mapping rules within Internet Information Services (IIS). It works both in classic mode and claims-based mode.

* 1. **Forms-Based Authentication (FBA)**

Utilizes a username-and-password HTML form that queries a membership provider on the back end. By default, it includes providers for LDAP and SQL Server; however, you can develop custom providers of your own. FBA is based on the standard forms authentication provided by Microsoft ASP.NET, which resides at the very core of SharePoint. It works only in claims-based mode.

* 1. **SAML Token-Based Authentication**

**Claims-based authentication**

It employs the concept of claims identity, representing each user’s identity as tokens made of claims. A claim is a statement, asserted by an issuer, about a subject, which is assumed to be true by the reader, due to a trust relationship between the reader and the issuer. Claims are issued by a claim provider and packaged into a security token, which is emitted by a security token service, which can also be an identity provider or can use an external identity provider. The identity provider is a service that authenticates the end users, based on a specific set of credentials. The power of claims-based authentication arises from the fact that claims-based identities are cross platform and can lead your solutions to provide single-sign-on capabilities on multiple platforms using a standard, secure, and reliable protocol.

**Forms-Based Authentication**

When you configure FBA, you gain the capability to authenticate your users against an external repository of users. By default, this can be an LDAP or Microsoft SQL Server database built using the standard SQL Membership Provider of ASP.NET. Of course, you can also develop custom membership providers of your own, querying any kind of users’ repository.

**Authorization infrastructure**

No matter which authentication mode and methods you choose, authorization in SharePoint is always managed the same way. This is a great feature that makes life easier for SharePoint administrators, because they do not need to care about the front-end authentication environment.

Authorization in SharePoint is based on permission levels, which are a formal definition of a set of permissions. Permission levels can be assigned to users (SPUser) or groups (SPGroup). Both SPUser and SPGroup inherit from SPPrincipal, which is the base class for every principal in SharePoint,

**Claims-based authentication and WS-Federation**

1. **Purpose**

The problem is evident: you and everyone else in today’s digital world have too many sets of credentials to remember, manage, and keep safe. Wouldn’t a better solution be to decouple applications and software solutions from their authentication environments, while taking advantage of a shared set of credentials? In the ideal digital world, you should authenticate once, at the very beginning of the day, and use a worldwide single-sign-on infrastructure.

1. **Real Environment**

Many software solutions authenticate their users just because they need to authorize access to resources or functionalities based on users’ identities. They do not really need, however, to collect and maintain users’ credentials. From an authorization viewpoint, it suffices to have some information about the users to cluster them in groups or audiences and authorize access to resources based on their properties.

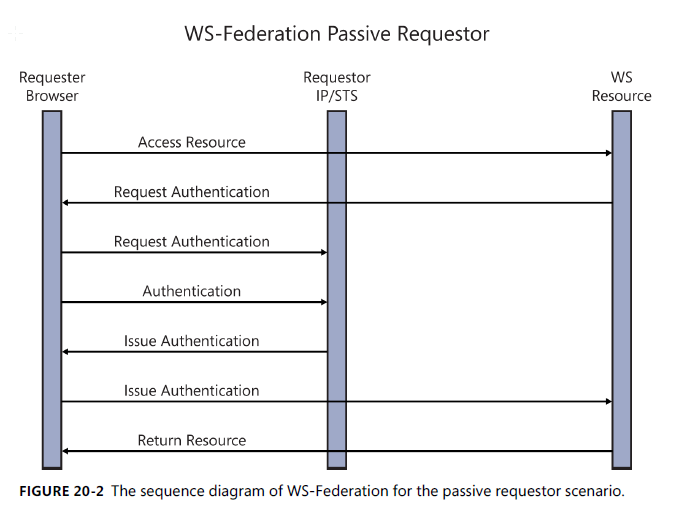
1. **Simplified Solution**

Pushed by these ideas, a few years ago the software market started working on the goal of defining an authentication infrastructure that could be externalized and that could identify every user as a digital identity. In Chapter 19, “Authentication and authorization infrastructure,” you learned that a digital identity is essentially a set of claims. Remember that a claim is a statement that is asserted by an issuer about a subject; this statement is assumed to be true by the reader, due to a trust relationship between the reader and the issuer. The externalized authentication provider is generally defined as the identity provider (IP) and often publishes an STS. The application or software solution externalizing the authentication process is called the service provider or relying party. The consumer, who uses the service provider for authenticating with the IP, is generally called the subject. Figure 20-1 portrays an extremely simplified authentication architecture employed by a software solution that uses externalized authentication.

WS == Web Service



From a technology viewpoint, these concepts use such specifications as WS-Security, WS-Trust, and WS-\*. The final goal of these specifications is to allow for implementing a token-based authentication and authorization system, in which third parties can issue tokens.



Read at

Today’s software solutions always require user authentication and authorization