



The Trusted Technology Partner in Business Innovation

PASSION • DISCIPLINE • INNOVATION • TEAMING • INTEGRITY

# SharePoint Search App

Custom App for Advanced Searches

Ken Mears

Senior Consultant, Portals & Collaboration Practice

Neudesic LLC



# WHO THE HECK IS THIS GUY?

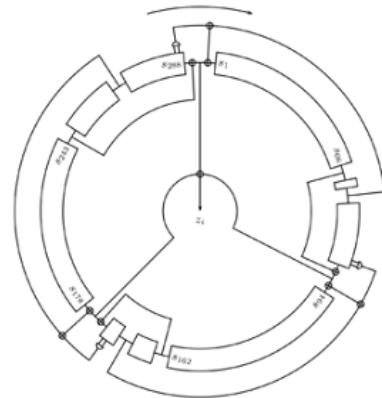
# Ken Mears

Senior Consultant  
Neudesic LLC

SharePoint Cipher – <http://krdmtech.wordpress.com/>

Twitter – [@krdmtech](#)

Email – [krdmtech@gmail.com](mailto:krdmtech@gmail.com)



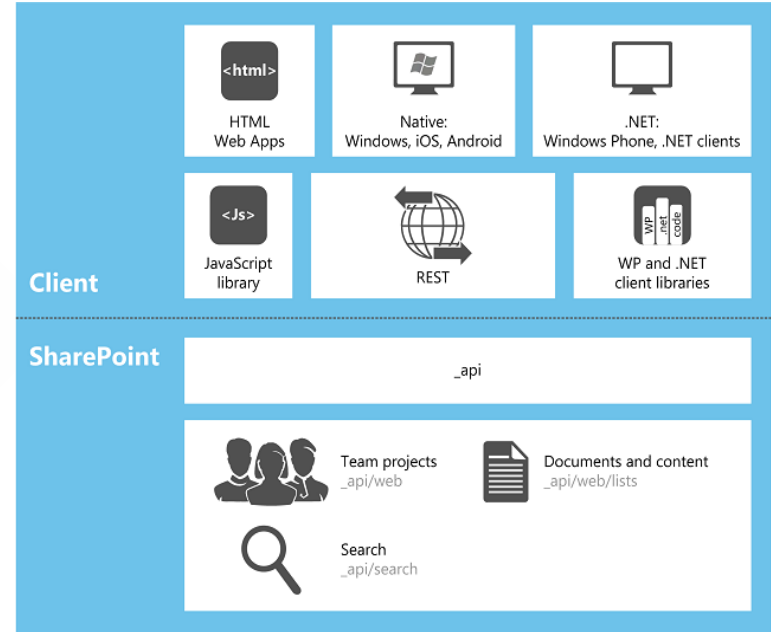
# AGENDA

- What's an App?
- App Hosting
- App Authentication
- Developing an App
- Packaging
- Demo... Let's do it!
- Questions



# SHAREPOINT 2013 APPS

- Apps are essentially web applications
- You can use any language, such as HTML, JavaScript, PHP, or .NET
- Common IDEs are Microsoft Visual Studio 2012, and a web-based tool, "Napa" Office 365 Development Tools
- The Cloud App Model supports a tiered architecture in which the app's business logic, data, and user interface (UI) can be distributed into separate components
- You can connect your app with just about any internal or public web services, take advantage of the new OAuth 2.0 support in SharePoint, and use the Representational State Transfer (REST) and client APIs (JavaScript and .NET) to integrate and connect your app with SharePoint.
- SharePoint 2013 provides a wide variety of features—such as **Search**, workflow, social networking, taxonomy, user profiles, and Business Connectivity Services (BCS)—that your apps can take advantage of.



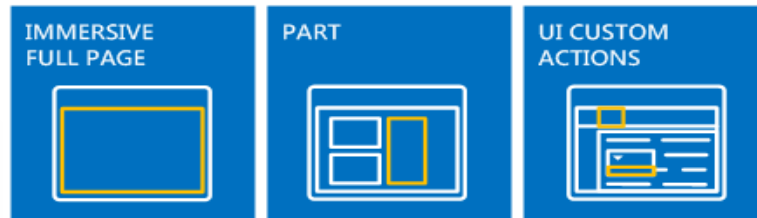
# APP CATALOG

- The App Catalog is a site collection where apps can be deployed to allow them to be shared with every other site collection/site in that web application.
- There can only be one app catalog per web application
- There can be multiple app catalogs in a farm
- During the creation of the app catalog you will be asked to supply a **Primary Site Collection Administrator**. This user is the gate keeper for all apps in this catalog.
- The app catalog administrators can also be given the control over the SharePoint Store. This will force users to “Request” apps from the store and the admins will need to approve or deny those requests.
- You will also be asked to supply a list of **End Users**. These users are the only set of users that will be able to see apps in the app catalog



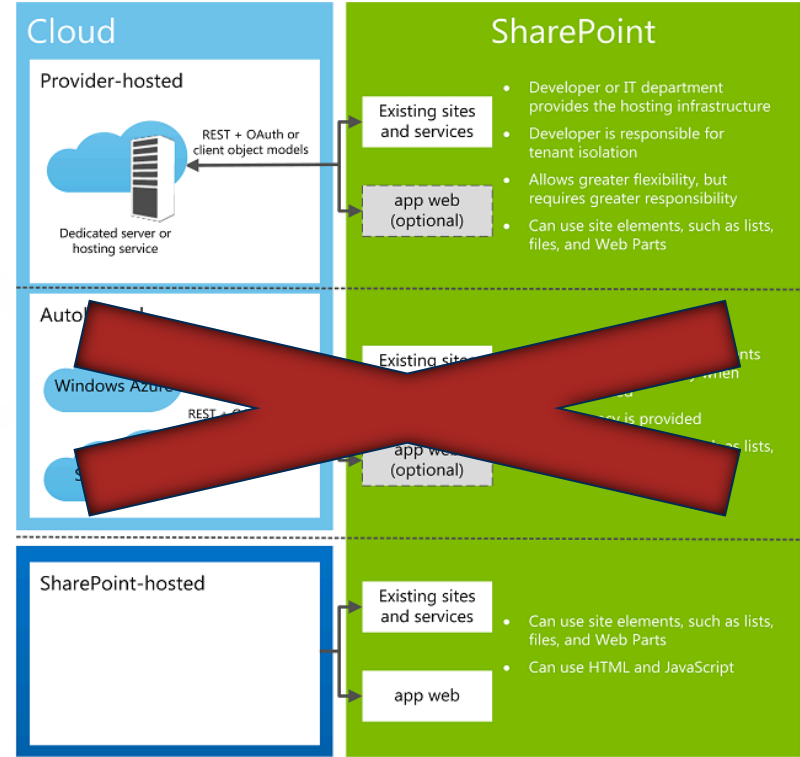
# FAMILIAR UX (USER EXPERIENCE)

Apps for SharePoint fit seamlessly into the SharePoint website where they are installed and therefore bring data and functionality to the users' familiar work environment and provide them with a familiar user experience. An app for SharePoint provides a fully immersive experience and optionally can extend some of the existing UI, such as in menus, or by providing embeddable parts for other pages. App parts surface an iFrame element that contains content from your app. Ribbon or context menu extensions make your app available on list items, documents, or anywhere else a ribbon is shown.



# APP HOSTING OPTIONS

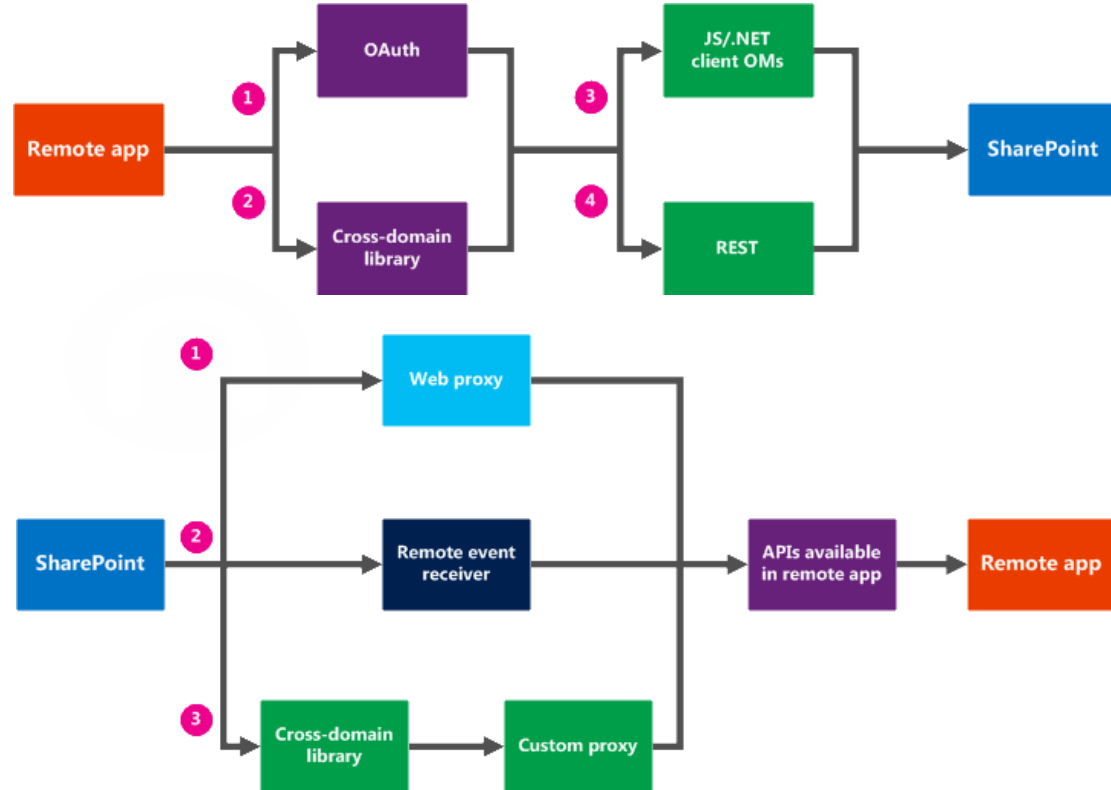
- **Cloud-hosted:** Includes at least one remote component and may also include SharePoint-hosted components. Within this category are two important types of apps:
  - **Provider-hosted:** These apps include components that are deployed and hosted outside the SharePoint farm. They are installed to the host web, but their remote components are hosted on another server.
  - **Auto-hosted:** Provider-hosted apps whose remote components are provisioned and deployed for you on Windows Azure. As with a provider-hosted app, an auto-hosted app for SharePoint interacts with a SharePoint website but also uses resources and services that are located on a remote site that is hosted by Windows Azure. The SharePoint 2013 installation provisions and deploys these resources for you.
- **SharePoint-hosted:** Apps where all components are hosted on either an on-premises or Office 365 SharePoint farm. SharePoint-hosted apps are installed on a SharePoint 2013 website, called the host web. They have their resources hosted on an isolated sub-site of a host web, called the app web.





# APP AUTHENTICATION

- When you are dealing with accessing SharePoint data from a remote app, you have to decide whether you want to authenticate and communicate to SharePoint by using (1) *OAuth*, or (2) the *cross-domain library*. Then, for the data access API, you must decide between (3) the *client object model (JavaScript/.NET client object models)*, or (4) *Representational State Transfer (REST)*.
- When you are accessing the data on your remote app from SharePoint, you have to decide whether you want to use (1) the *web proxy*, (2) *remote event receivers*, or (3) the *cross-domain library with a custom proxy page* to authenticate and communicate with external services or apps. You have to use the available APIs in the remote app to retrieve or manipulate data.



# APP AUTHENTICATION (CONT'D)

You have to consider several aspects when you work with data in your app. For example, what route is the data using? Is it coming from or going through the server? Is it going through the client? Is it OK to authenticate as the logged-on user? Does the app need elevated privileges?

## Inbound data connectivity

- **OAuth:** An open protocol that enables secure authorization in a simple and standard way. OAuth enables users to approve an application to act on their behalf without sharing their user name and password. You can use OAuth with server-side code. It is a good option if you need to run a non-interactive process, or if you need to elevate privileges to other than those of the logged-on user.
- **Cross-domain library:** A client-side alternative in the form of a JavaScript file (**SP.RequestExecutor.js**) hosted in the SharePoint website that you can reference in your remote app as long as they are in the same internet zone. The cross-domain library allows you to interact with more than one domain in your remote app page through a proxy. This is a good option if you prefer your app code to run in the client rather than in the server, or if there are connectivity barriers, such as firewalls, between SharePoint and your remote infrastructure.

## Inbound data connectivity options: Which one should I use?

The following table lists the common requirements and scenarios you might encounter when you are building apps. An **x** in the column indicates which option—OAuth or cross-domain library—you can use in each case.

Requirement/Scenario	OAuth	Cross-domain library
I use client-side technologies (HTML + JavaScript).		x
I want to use REST interfaces.	x	x
There is a firewall between SharePoint and my remote app, and I need to issue the calls through the browser.		x
My app needs to access resources as the logged-on user.	x	x
My app needs to elevate privileges to other than those of the current logged-on user.	x	
My app needs to act on behalf of a user other than the one who is logged on.	x	
My app needs to perform operations only while the user is logged on.	x	x
My app needs to perform operations even when the user is not logged on.	x	



# APP AUTHENTICATION (CONT'D)

## Outbound data connectivity

- **Web proxy:** As a developer, you can use the web proxy exposed in client APIs such as the JavaScript/.NET client object models. When you use the web proxy, you issue the initial request to SharePoint. In turn, SharePoint requests the data to the specified endpoint and forwards the response back to your page. Use the web proxy when you want the communication to occur at the server level.
- **Remote event receivers:** You can use remote event receivers to handle events that occur to an item in the app, such as a list, a list item, or a web. These events resemble those in a traditional SharePoint solution, except that they can work with the remote components of the app for SharePoint.
- **Custom proxy page for the cross-domain library:** You can use the cross-domain library to access data in your remote app if you provide a custom proxy page that is hosted in the remote app infrastructure. As the developer, you are responsible for the custom proxy page implementation and must deal custom logic, such as the authentication mechanism, to the remote app. Use the cross-domain library with a custom proxy page if you want the communication to occur at the client level.

## Outbound data connectivity options: Which one should I use?

The following table lists the common requirements and scenarios you might encounter when you are building apps. An x in the column indicates which option—web proxy library or cross-domain library with a custom proxy page—you can use in each case.

Requirement/Scenario	Web Proxy	Cross-domain library with custom proxy page	Remote event receivers
I use client-side technologies (HTML + JavaScript).	x	x	
I cannot add pages or components to the remote app or service.	x		x
I want to use REST interfaces.	x	x	x
I want to use the JavaScript CSOM.	x	x	
I want to use the .NET CSOM.	x		x
There is no direct connectivity between the SharePoint server and my app. I need to issue calls through the browser.		x	
My app needs to access resources as the logged-on user.	x	x	



# DEVELOPING THE APP

- A new domain is needed for all Apps in the SharePoint farm

- Setup App Catalog
- Create app domain forward lookup zone
- Create \* Hosts(A)'s for all IP addresses
- Run PowerShell script to create App service application

```
add-psnapin "Microsoft.Sharepoint.Powershell"
```

```
$account = Get-SPManagedAccount Domain\Account
```

```
Remove-SPServiceApplicationPool -Identity SettingsServiceAppPool
```

```
$appPoolSubSvc = New-SPServiceApplicationPool -Name SettingsServiceAppPool -Account $account
```

```
$appSubSvc = New-SPSubscriptionSettingsServiceApplication -ApplicationPool
```

```
$appPoolSubSvc -Name SettingsServiceApp -DatabaseName SettingsServiceDB
```

```
$proxySubSvc = New-SPSubscriptionSettingsServiceApplicationProxy -ServiceApplication
```

```
$appSubSvc
```

## Before you begin

Buy a domain for apps from a domain provider

**ContosoApps.com**

## On your DNS Server



- 1 Create a forward lookup zone for apps

**ContosoApps.com**

- 2 Create a CNAME alias from the app domain to the SharePoint domain

**\*.ContosoApps.com -> SharePoint.Contoso.com**

## For SSL (https:// URLs)



- 3 Create a wildcard SSL certificate for the new app domain

**https://\*.ContosoApps.com**

## On your SharePoint Servers



- 4 Configure the Subscription Settings service application by using Windows PowerShell.

- 5 Configure the App Management service application (Central Administration or Windows PowerShell).



- 6 Configure the App URLs in Central Administration

**Domain: ContosoApps.com  
Prefix: Apps**



# DEVELOPING THE APP (CONT'D)

## — Configure App URLs

- In the App domain box, type the isolated domain that you created for hosting apps
- In the App prefix box, type a name to use for the URL prefix for apps

`http://app-bdf2016ea7dacb.contosoapps.com/sites/SPC/Poll`

The diagram illustrates the components of the URL `http://app-bdf2016ea7dacb.contosoapps.com/sites/SPC/Poll` using arrows and labels:

- `app` is labeled **App prefix (tenant)** (orange text).
- `bdf2016ea7dacb` is labeled **App ID** (green text).
- `contosoapps.com` is labeled **App domain** (red text).
- `Host web` is labeled **Host web** (blue text).
- `SPC` is labeled **App name** (purple text).
- `Poll` is labeled **App name** (purple text).



# PACKAGING AND DISTRIBUTING APPS

Apps for SharePoint are distributed as an app package. Depending on where the app is hosted, the package may contain different components.

- For provider-hosted apps, where you or your IT department hosts the app on a dedicated server or third-party hosting service, the package might just contain the app manifest.
- \* Auto-hosted app packages may include items to be deployed on the Windows Azure cloud.
- SharePoint-hosted apps might contain some SharePoint-related components, in addition to the app manifest.

If you use "Napa" Office 365 Development Tools or Visual Studio 2012 to build apps and deploy directly from them, the tools let you publish your app into a file that has an .app extension. You can then distribute the app in one of two ways:

- Apps that are meant to run within a particular organization can be deployed using the App Catalog.
- Apps for the Office Store can be submitted via the seller hub. Always remember to add licensing checks to the app.



# DEMO



Visual Studio



SharePoint



# QUESTIONS?





# REFERENCES

Apps for SharePoint overview

[http://msdn.microsoft.com/library/office/apps/fp179930\(v=office.15\)](http://msdn.microsoft.com/library/office/apps/fp179930(v=office.15))

Choose patterns for developing and hosting your app for SharePoint

<http://msdn.microsoft.com/en-us/library/fp179887.aspx>

Data access options for apps in SharePoint 2013

<http://msdn.microsoft.com/en-US/library/office/apps/fp179897>

Host webs, app webs, and SharePoint components in SharePoint 2013

<http://msdn.microsoft.com/en-us/library/fp179925.aspx>

SharePoint 2013 - App Development

<https://www.nothingbutsharepoint.com/sites/devwiki/articles/Pages/SharePoint-2013-App-Development.aspx>

Publish apps for SharePoint

<http://msdn.microsoft.com/en-us/library/jj164070.aspx>

Convert autohosted app to provider-hosted app

[http://msdn.microsoft.com/EN-US/library/office/dn722449\(v=office.15\).aspx](http://msdn.microsoft.com/EN-US/library/office/dn722449(v=office.15).aspx)

Manage the App Catalog in SharePoint 2013

<http://technet.microsoft.com/en-us/library/fp161234.aspx>

Configure Environment

<http://technet.microsoft.com/en-us/library/fp161236.aspx>

Configure an Environment for Apps for SharePoint 2013

<https://www.nothingbutsharepoint.com/sites/devwiki/articles/pages/configure-an-environment-for-apps-for-sharepoint-2013.aspx>

Planning the Infrastructure Required for the new App Model in SharePoint 2013

<http://blogs.technet.com/b/speschka/archive/2012/09/03/planning-the-infrastructure-required-for-the-new-app-model-in-sharepoint-2013.aspx>

Prepare your SharePoint 2013 farm for App development and debugging

<http://blogs.msdn.com/b/how24/archive/2013/06/14/prepare-your-sharepoint-2013-farm-for-app-development-and-debugging.aspx>

