

**Will Apps End the SharePoint Cold War?**  
Sean McDonough



# Sean P. McDonough

SharePoint Gearhead,  
Developer and Evangelist.

My  
Company:



Twitter: @spmcdonough

About: <http://about.me/spmcdonough>

Blog: <http://SharePointInterface.com>

# The Agenda

1. Evolution of Development
  2. Examine the New App Model
  3. How Apps are Exposed
  4. Devs and Admins: Together Again!
  5. Side Notes & Demo
- + Q&A throughout!

CHECKLIST





## So why am I talking about the New App Model?

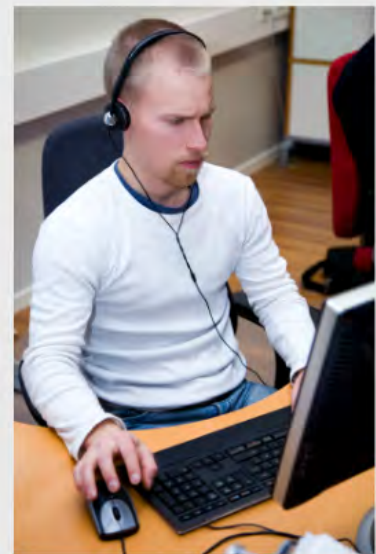
- believe it or not, my background is development
- I've been working with the new App Model since last summer
- I speak jive ...  
I mean, I speak "admin" too



Cold War?



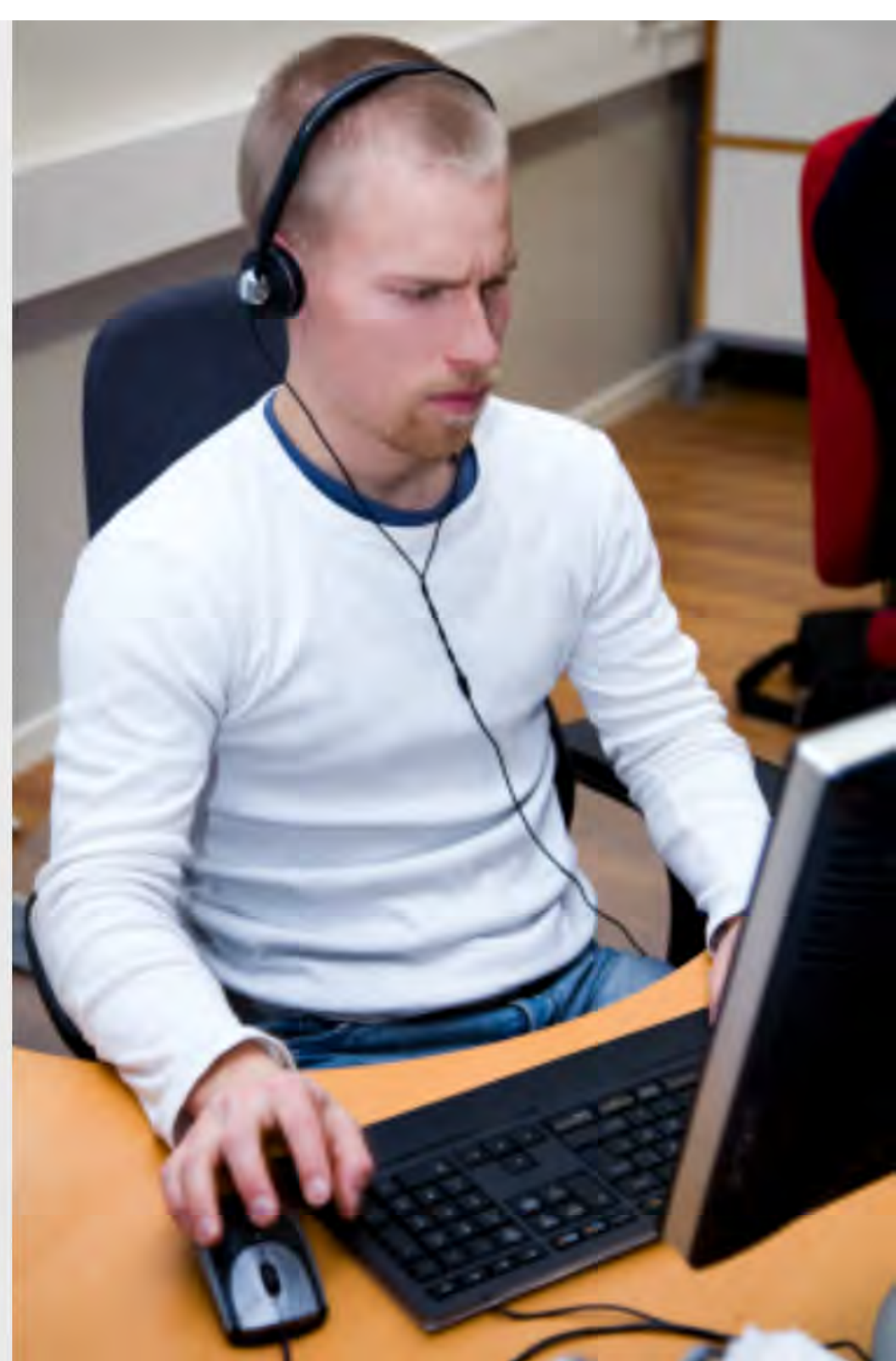
Cold War?



**Developer vs. Administrator**

# Ask a dev ...

- Building solutions for SharePoint is hard!
- Setting up SharePoint just to develop is hard
- I'm just trying to solve business problems using the tools available to me



# Ask a dev about admins ...

- They come up with policies just to block our solutions and what they do
- They just seem to want more and more documentation. I spend all day writing!



Flatline my farm and I'm gonna do some dental work!

They're a bunch of trolls!





# Ask an admin ...

- Keeping a SharePoint farm running is hard!
- Farm administration gets harder with each release
- When something goes wrong, I'm the one who gets called and yelled at

# Ask an admin about devs ...

- I tune it up, and their custom code tanks my farm's performance
- `RunWithElevatedPrivileges` isn't the norm; devs need to learn how security works
- Someone dies the next time I hear, "well, it worked on \*my\* machine"



# Cold War?



## Developer vs. Administrator

### How did it get this way?

# The Evolution of Support for Custom SharePoint Applications





2003

# 2003

## Windows SharePoint Services 2.0 SharePoint Portal Server 2003

- Server object model available to developers
- No SharePoint-specific development support
- Customizations deployed directly to servers (by hand or through MSI)
- SharePoint unaware of changes being made



2007

# 2007

Windows SharePoint Services 3.0

Microsoft Office SharePoint Server 2007

- Features and solution packages (WSPs) added to management custom code deployments
- SharePoint gains awareness of solutions that are available and deployed to the farm
- Primary mode of development still focuses on server object model





2010



2010

# 2010

## Microsoft SharePoint Foundation 2010

## Microsoft SharePoint Server 2010

- Biggest initial change comes for developers: Visual Studio support for building SharePoint solutions.
- Primary focus still on building full-trust, server-side solutions
- Client-side object model (CSOM) is introduced
- Sandboxed solutions are introduced
- Both CSOM and sandboxed solutions represent a change in how solutions are conceptualized and architected



**2013**

# 2013

Microsoft SharePoint Foundation 2013

Microsoft SharePoint Server 2013

- Introduction of the new Cloud App Model (or simply "App Model")
- Focus shifts away from full-trust, server-side code to building client-side applications
- New App Model is pushed heavily by Microsoft and is the only one that will work both on-premises and in Office 365

# The future is cloudy

## The App Model Experience

### What's so special

#### Isolation

- No code runs on the SharePoint server - it runs on the client, "another server," or both
- Apps run from a separate App Web that has a different hostname (URL) than its hosting SharePoint site (Host Web)
- Apps do not have direct access to Host Web SharePoint data



#### Key App Model Aspects



#### Security

- Apps have no privileged access to SharePoint data. They go through web services
- Now App Model introduces the concept of application identity, which just "looks like" your identity
- OAuth 2.0 is underlying framework for App Model authentication

#### Hygiene

- Apps are deployed in packages that contain content and declarative markup only: HTML, ASPX pages (but no code-behind), images, XML files, list templates, etc. No DLLs
- App data is persisted within the App Web - not the Host Web
- On occasion, entire App Web is basically wiped out resulting in clean install, nothing remains



#### about Apps?



Are you a developer?

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

SharePoint 2013 App Model

#### Standards

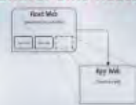
- App Model focuses on building solutions with modern web technologies like HTML, CSS and JavaScript
- Back end business logic can be built with any stack: Microsoft/.NET, LAMP, Java/JEE, etc.
- SharePoint data is accessed through CSOM and OData compliant (mostly REST web service) requests

#### App Model Hosting Options



- Scoped to web/sub-site level
- Everything runs in the browser
- All "code" is HTML + CSS + JavaScript!
- Minimal security concerns (relatively speaking)

#### SharePoint-hosted



Noteworthy Points

- Scope controlled by provider implementation
- Provider chooses their technology stack
- Code can call back to SharePoint independently
- Security can be tricky (S2S Trust and OAuth 2.0)

#### Provider-hosted



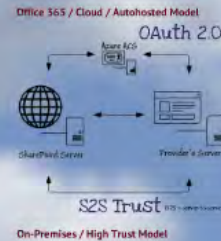
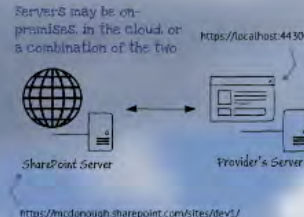
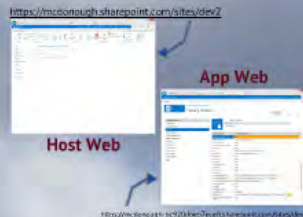
Noteworthy Points

- Special case of Provider-hosted App Model
- Technology stack is Azure-based web & SQL
- Office 365-only model (for now...)
- Security simplified by Office 365 and Azure ACS

#### Autohosted



Noteworthy Points



### How Apps "Surface" in SharePoint



Installation / Full Page Experience

User Interface Customizations

App Parts

Let's install an App

A typical SharePoint 2013 team site

### Back to the Cold War ...



### Is there hope?



The new App Model addresses many of these issues directly

- Isolation
- Hygiene
- Standards
- Security

Much of the attention has to do with poor isolation and ambiguity around custom isolation responsibilities

"Good fences make good neighbors"

### The New App Model what the changes mean

Isolation	Hygiene	Standards	Security
<ul style="list-style-type: none"> <li>Apps are deployed in packages that contain content and declarative markup only: HTML, ASPX pages (but no code-behind), images, XML files, list templates, etc. No DLLs</li> <li>App data is persisted within the App Web - not the Host Web</li> <li>On occasion, entire App Web is basically wiped out resulting in clean install, nothing remains</li> </ul>	<ul style="list-style-type: none"> <li>Apps are deployed in packages that contain content and declarative markup only: HTML, ASPX pages (but no code-behind), images, XML files, list templates, etc. No DLLs</li> <li>App data is persisted within the App Web - not the Host Web</li> <li>On occasion, entire App Web is basically wiped out resulting in clean install, nothing remains</li> </ul>	<ul style="list-style-type: none"> <li>App Model focuses on building solutions with modern web technologies like HTML, CSS and JavaScript</li> <li>Back end business logic can be built with any stack: Microsoft/.NET, LAMP, Java/JEE, etc.</li> <li>SharePoint data is accessed through CSOM and OData compliant (mostly REST web service) requests</li> </ul>	<ul style="list-style-type: none"> <li>Apps have no privileged access to SharePoint data. They go through web services</li> <li>Now App Model introduces the concept of application identity, which just "looks like" your identity</li> <li>OAuth 2.0 is underlying framework for App Model authentication</li> </ul>

Oh yeah, one more area that deserves a mention

### The App Store



- Only apps built using the new App Model can appear in the App Store
- Apps built under the new App Model are the only custom SharePoint code in Office 365



Office 365 navigation bar with tabs: BROWSE, FILES, LIBRARY. Includes icons for View, Quick Edit, Create View, Create Column, and various utility icons like E-mail a Link, Alert Me, RSS Feed, etc.

- Home
- Notebook
- Documents
- Apps in Testing
- Samples
- Developer Center
- Recent
  - Idera SharePoint Library Analyzer
- Site Contents
- EDIT LINKS

+ new document or drag files here

All Documents









✓	Name	Modified	Modified By
There are no files in the view "All Documents"			

/start.asp

Documents - All Documents

Outlook Calendar People Newsfeed SkyDrive Sites Admin

SHARE

 Alert Me	 RSS Feed	 Most Popular Items	 Tags & Notes	 Connect to Outlook	 Examine Library Details	 Connect to Office	 Export to Excel	 Form Web Parts	
Share & Track			Tags and Notes	Connect & Export			 Open with Explorer	 Edit Library	
								 New Quick Step	Customize Library

ag files here

file

modified By

"All Documents".



BROWSE



Idera SharePoint Library Analyzer

# Library Details

Details | Compare | About

## Available Libraries

- App Packages
- Converted Forms
- Documents**
- Form Templates
- List Template Gallery
- Master Page Gallery
- Site Assets
- Site Pages
- Solution Gallery
- Style Library
- Theme Gallery
- Web Part Gallery
- wfpub



Name: Documents

Web Address: <https://mcdonough.sharepoint.com/sites/dev2/Shared Documents/Forms/AllItems.aspx>

Description:

Library Property	Current Setting/Value
<b>Enable Versioning?</b>	false
Entity Type Name	Shared_x0020_Documents
Force Checkout?	false
Has External Data Source?	false
Hidden?	false
ID	109f44b3-4097-4338-ae78-55c6167a47b9
Image URL	/_layouts/15/images/itdl.png?rev=23
IRM Enabled?	false
IRM Expire	false
IRM Reject	false
Is Application List?	false
Is Catalog?	false
Is Private?	false
Item Count	0
Last Item Deleted Date	2013-01-21T13:44:36Z (Monday, January 21, 2013 1:44:36 PM)
Last Item Modified Date	2013-01-21T13:44:36Z (Monday, January 21, 2013 1:44:36 PM)
List Item Entity Type Full Name	SP.Data.Shared_x0020_DocumentsItem
Multiple Data List?	false
No Crawl?	false
On Quick Launch?	true
Parent Web URL	/sites/dev2
Schema XML	<List DocTemplateUrl="/sites/dev2/Shared Documents/Forms/template.dotx&... <a href="#">more</a>
Server Template Can Create Folders?	true



https://mcdonough-bc920dbeb7ece8.sharepoint.com/sites/

Development Site 2

BROWSE



Idera SharePoint Library Analyzer

# Library Detail

## Available Libraries

App Packages

Converted Forms



# Key App Model Aspects

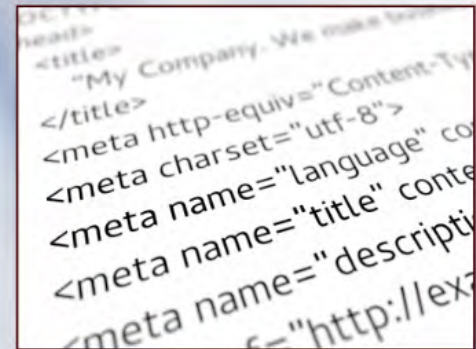
# What's so special



## Key App Model Aspects

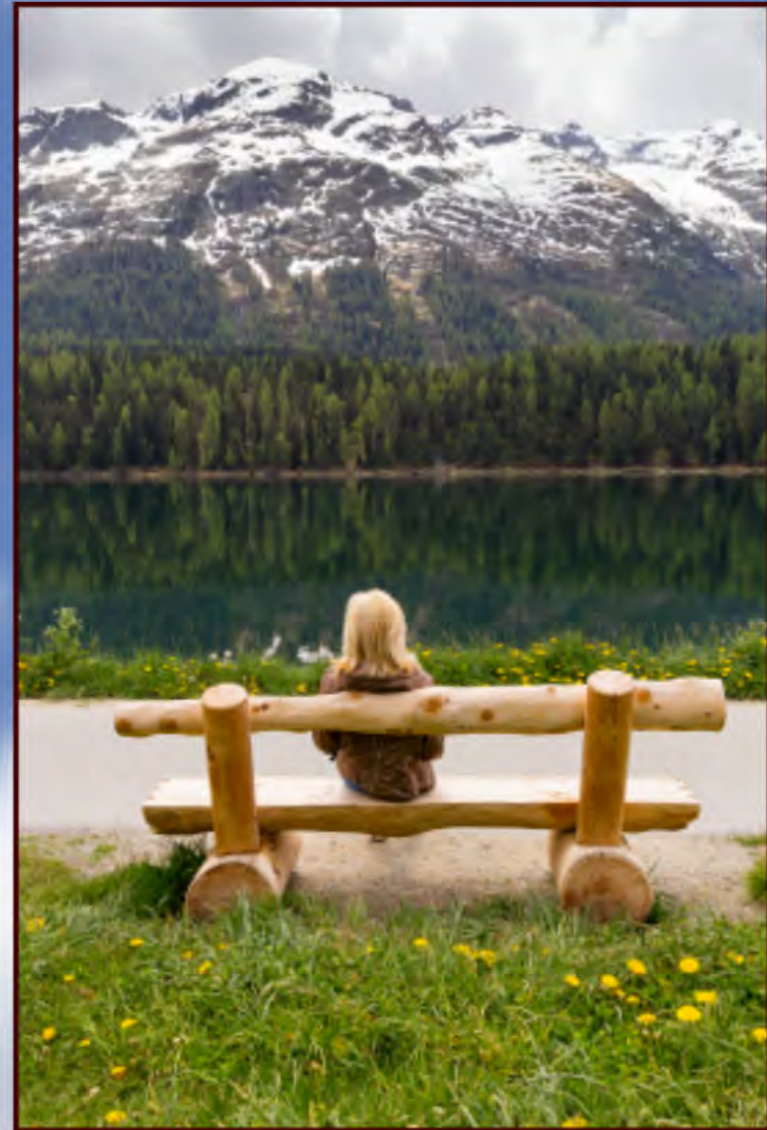


# about Apps?



# Isolation

- No code runs on the SharePoint server - it runs on the client, "another server," or both
- Apps run from a separate App Web that has a different hostname (URL base) than its hosting SharePoint site (Host Web)
- Apps do not have direct access to Host Web SharePoint data



# Hygiene



- Apps are deployed as packages that contain content and declarative markup only: HTML, ASPX pages (but no code-behind), images, XML files, list templates, etc. No DLLs.
- App data is persisted within the App Web - not the Host Web
- On uninstall, entire App Web is basically wiped out resulting in clean uninstall. Nothing remains.

# Standards

```
<title>  
  "My Company. We make  
</title>  
<meta http-equiv="Content-Type"  
<meta charset="utf-8">  
<meta name="language" co  
<meta name="title" conte  
<meta name="descripti  
s="http://ex?
```

- App Model focuses on building solutions with presentation tier technologies like HTML, CSS, and JavaScript
- Back-end business logic can be built on/with any stack: Microsoft/.NET, LAMP, Java/J2EE, etc.
- SharePoint data is accessed/updated through CSOM and OData-compliant (mostly) REST web service endpoints

# Security



- Apps have no privileged access to SharePoint data; they go through web services
- New App Model introduces the concept of application identity (versus just "user identity")
- OAuth 2.0 is underlying framework for App and User-based trust



# Are you a developer?

If so, you might be wondering ...



How the #\$@! am I supposed  
to build an application with  
these constraints?



I understand.



I'm here to help.



**Dr. Elisabeth Kübler-Ross**

The Five

Stages of Grief

1. Denial
2. Anger
3. Bargaining
4. Depression
5. Acceptance

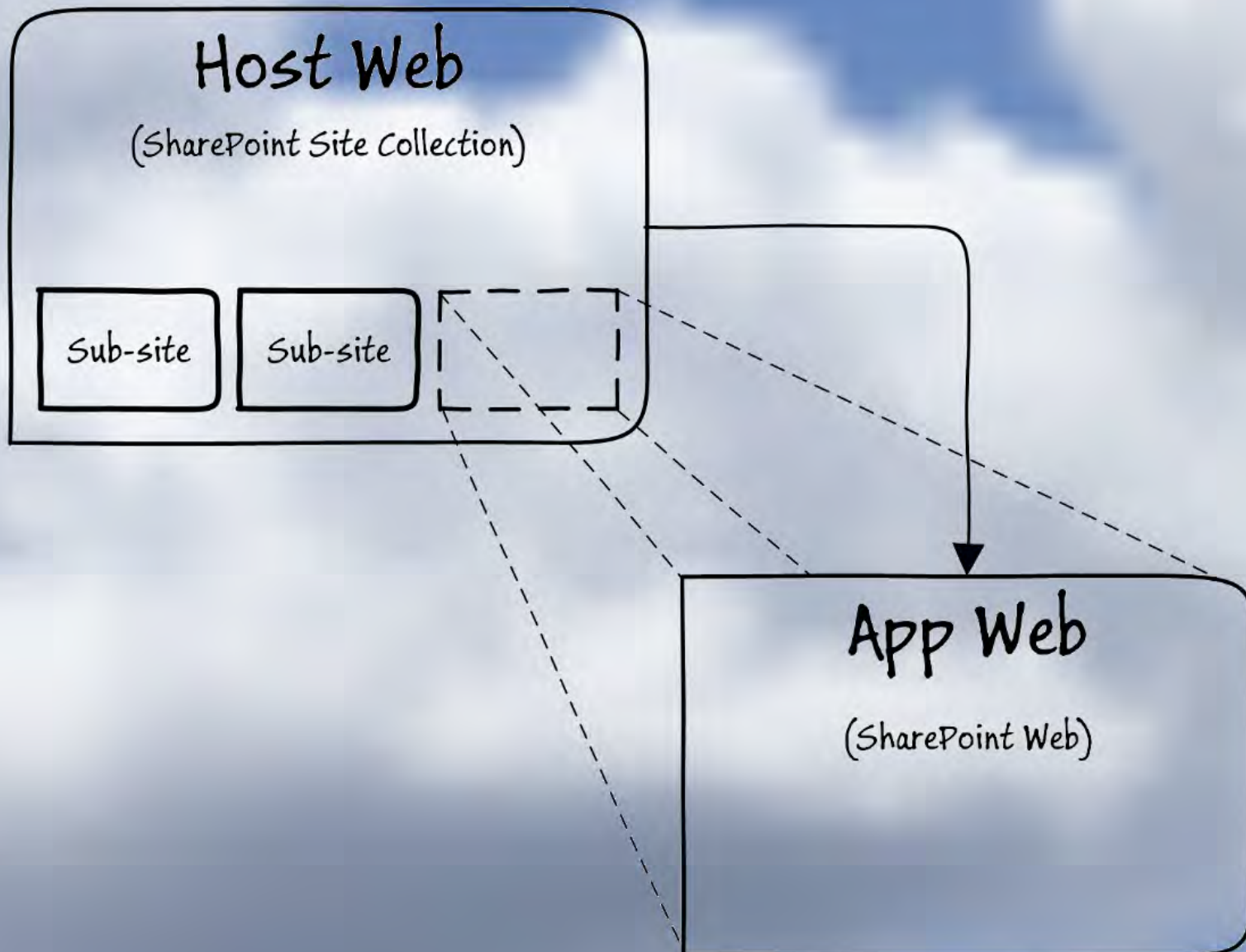


*There are 3*

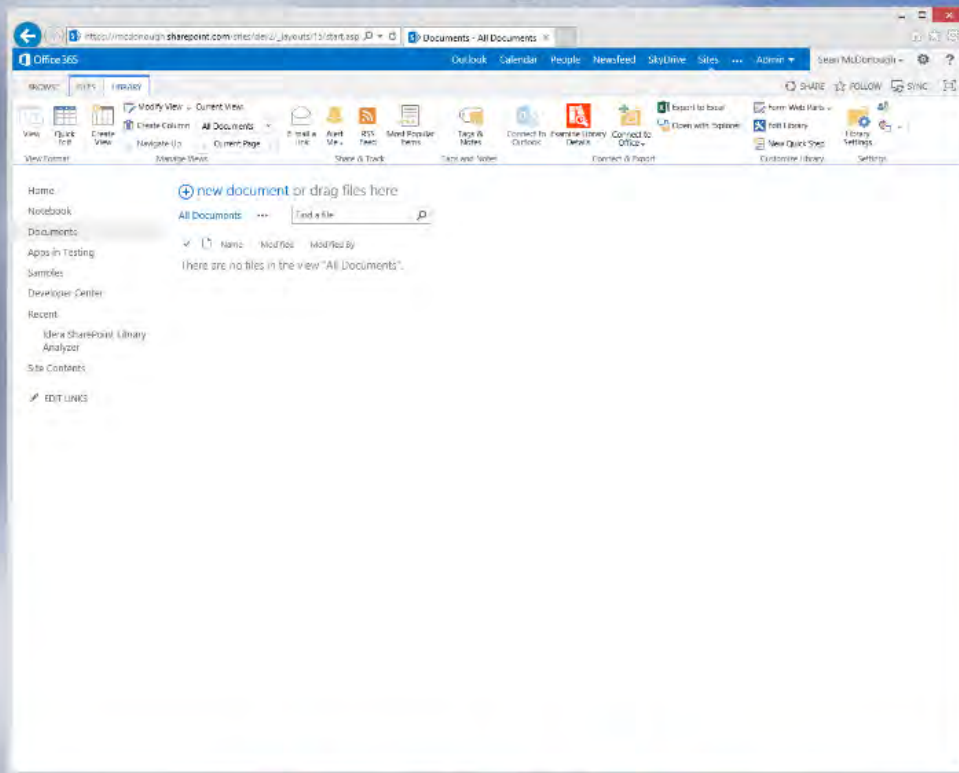
# App Model Hosting Options



# SharePoint-hosted



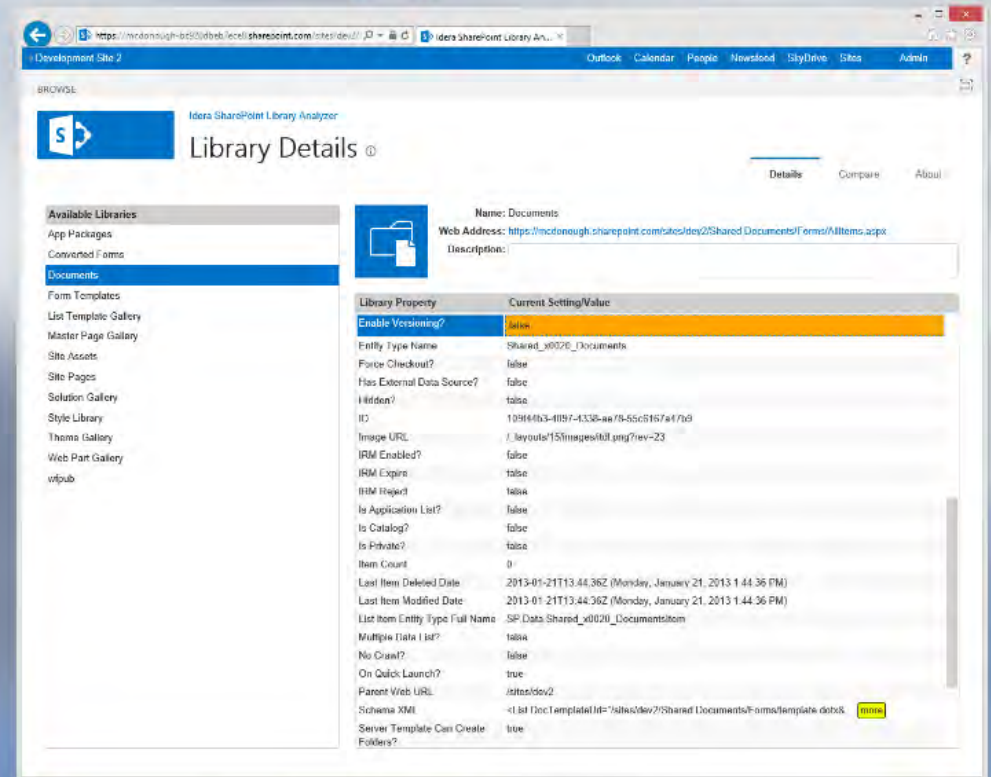
<https://mcdonough.sharepoint.com/sites/dev2>



Host Web

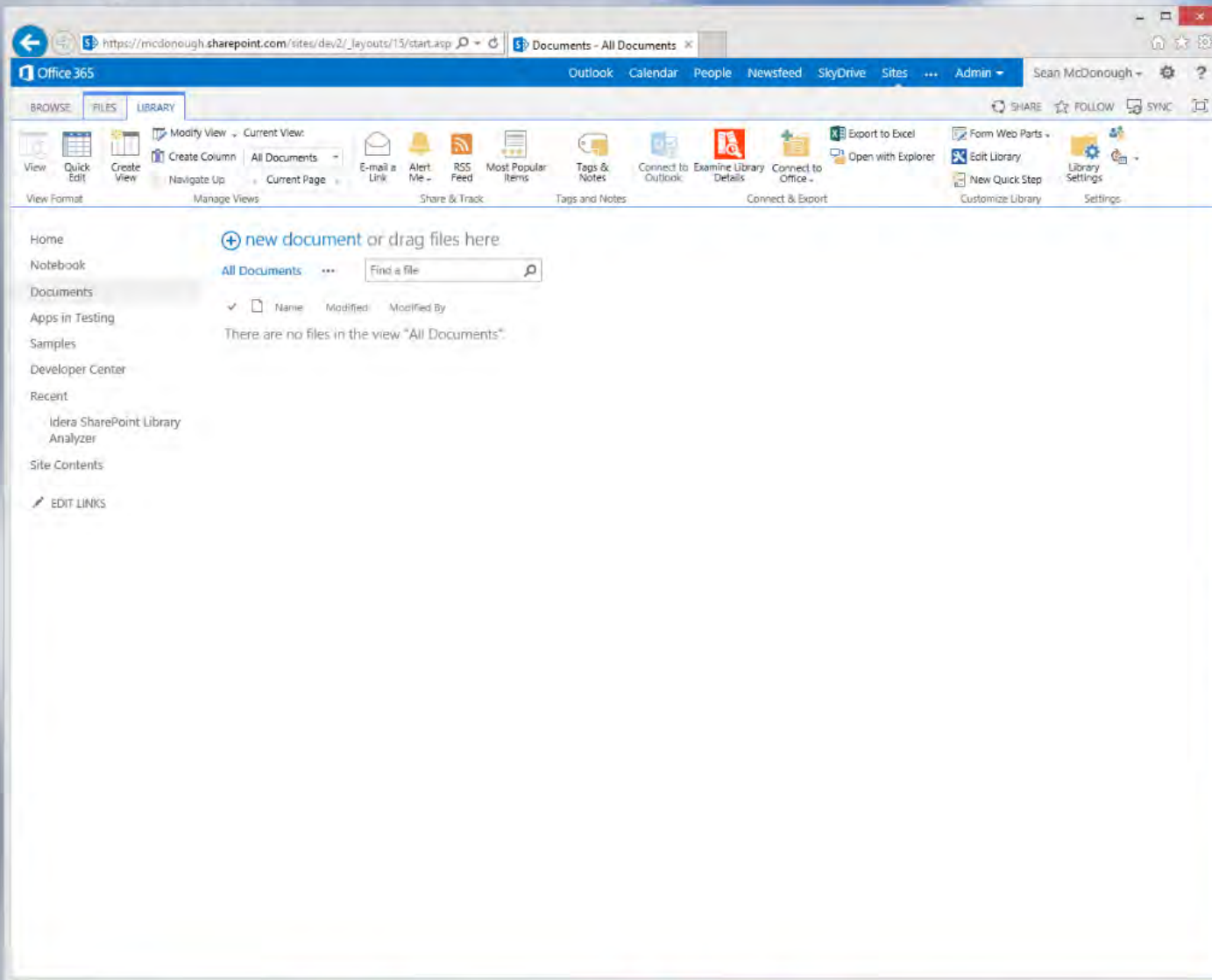


App Web

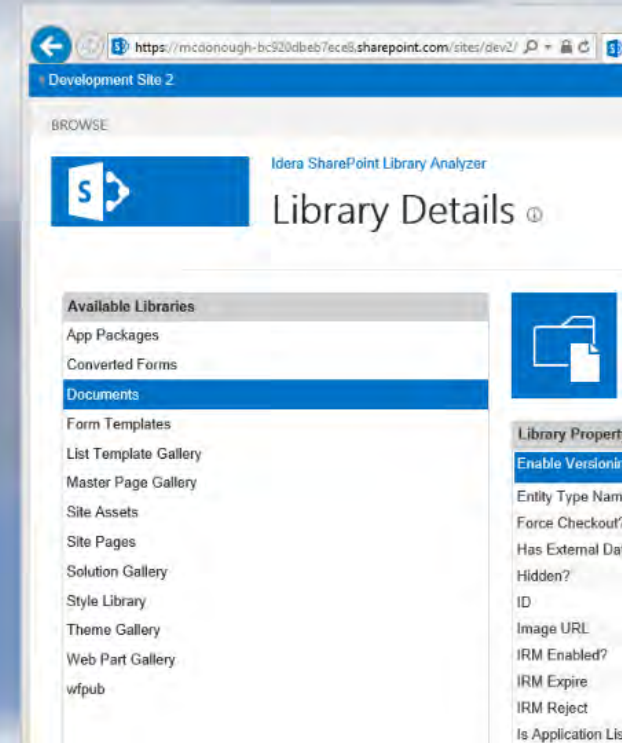


<https://mcdonough-bc920dbeb7ece8.sharepoint.com/sites/dev2>

<https://mcdonough.sharepoint.com/sites/dev2>

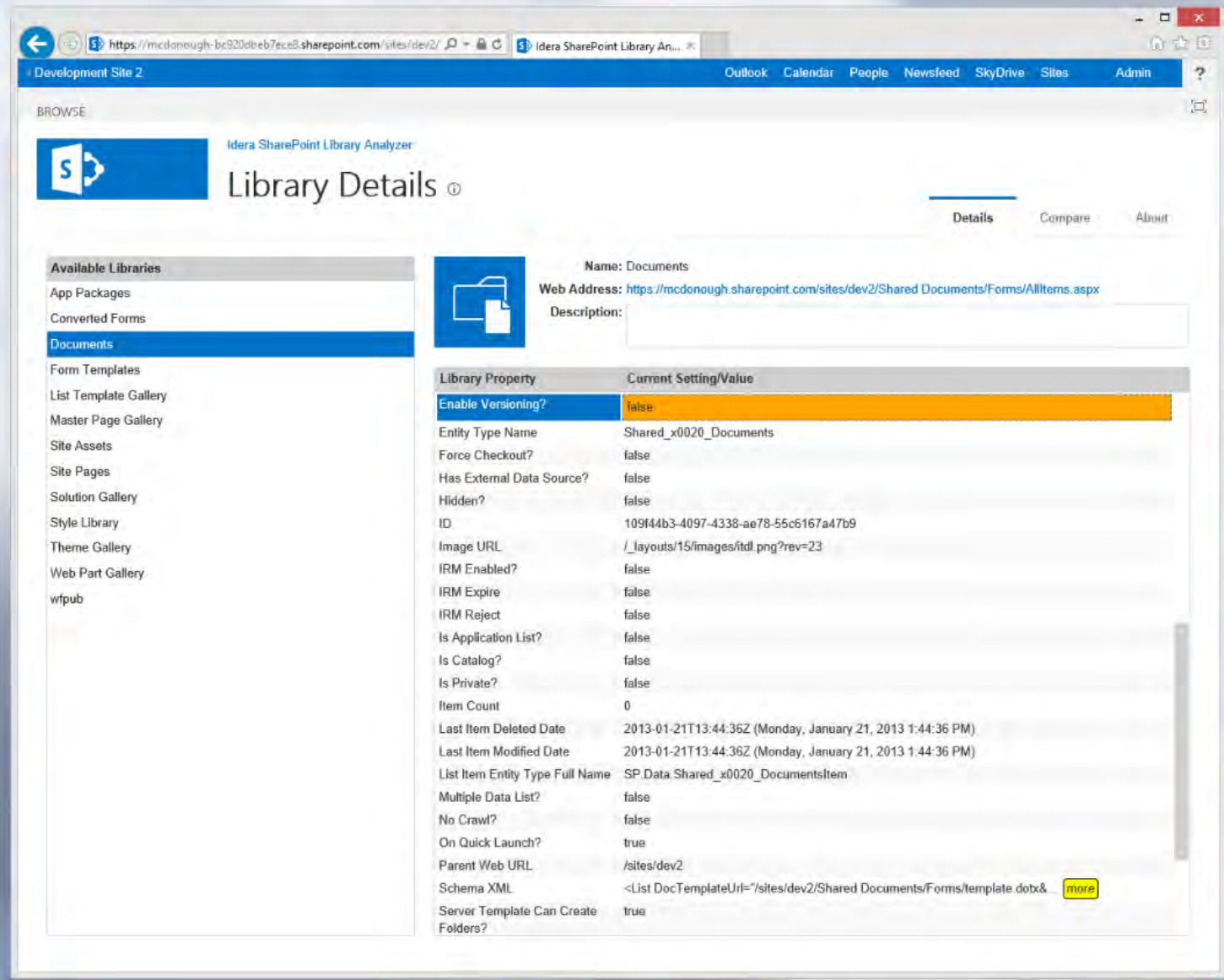


**Host Web**





# App Web



The screenshot displays the Idera SharePoint Library Analyzer interface. The browser address bar shows the URL: <https://mcdonough-bc920dbeb7ece8.sharepoint.com/sites/dev2/>. The page title is "Library Details" for the "Documents" library. The interface includes a navigation menu on the left with options like "Available Libraries", "App Packages", "Converted Forms", "Documents", "Form Templates", "List Template Gallery", "Master Page Gallery", "Site Assets", "Site Pages", "Solution Gallery", "Style Library", "Theme Gallery", "Web Part Gallery", and "wpub". The main content area shows the library's name, web address, and a table of properties.

**Library Details**

Name: Documents  
Web Address: <https://mcdonough.sharepoint.com/sites/dev2/Shared Documents/Forms/AllItems.aspx>  
Description:

Library Property	Current Setting/Value
Enable Versioning?	false
Entity Type Name	Shared_x0020_Documents
Force Checkout?	false
Has External Data Source?	false
Hidden?	false
ID	109f44b3-4097-4338-ae78-55c6167a47b9
Image URL	/_layouts/15/images/itdl.png?rev=23
IRM Enabled?	false
IRM Expire	false
IRM Reject	false
Is Application List?	false
Is Catalog?	false
Is Private?	false
Item Count	0
Last Item Deleted Date	2013-01-21T13:44:36Z (Monday, January 21, 2013 1:44:36 PM)
Last Item Modified Date	2013-01-21T13:44:36Z (Monday, January 21, 2013 1:44:36 PM)
List Item Entity Type Full Name	SP.Data.Shared_x0020_DocumentsItem
Multiple Data List?	false
No Crawl?	false
On Quick Launch?	true
Parent Web URL	/sites/dev2
Schema XML	<List DocTemplateUrl="/sites/dev2/Shared Documents/Forms/template.dotx&... more
Server Template Can Create Folders?	true

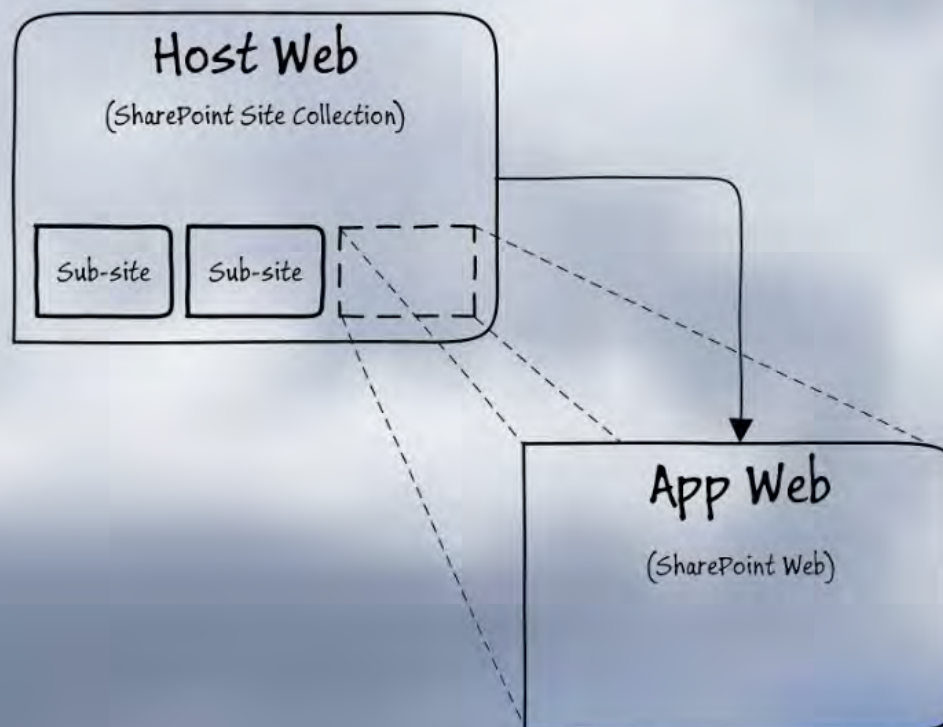
Web



<https://mcdonough-bc920dbeb7ece8.sharepoint.com/sites/dev2/>

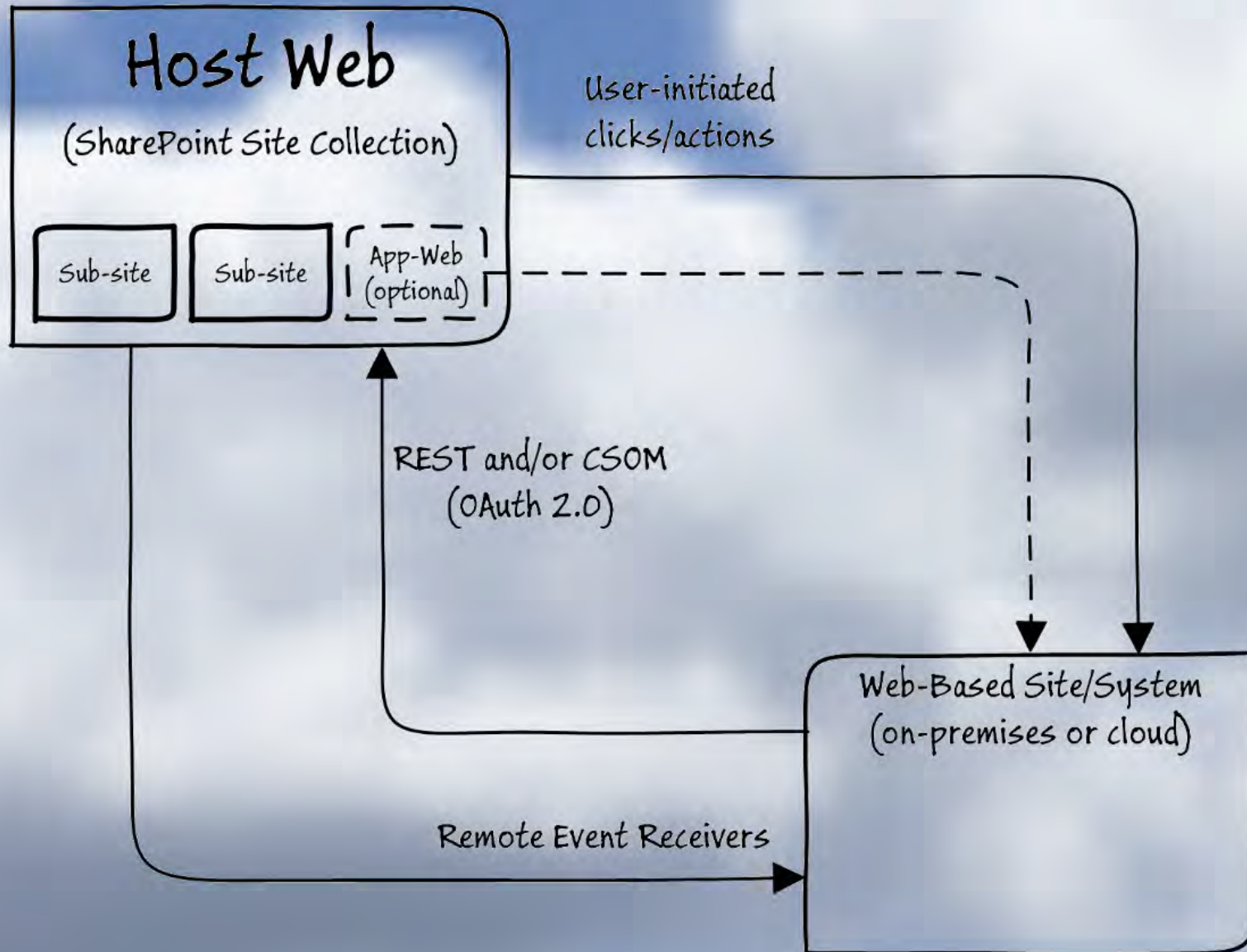
- Scoped to web/sub-site level
- Everything runs in the browser
- All "code" is HTML + CSS + JavaScript!
- Minimal security concerns (relatively speaking)

## SharePoint-hosted



Noteworthy  
Points

# Provider-hosted



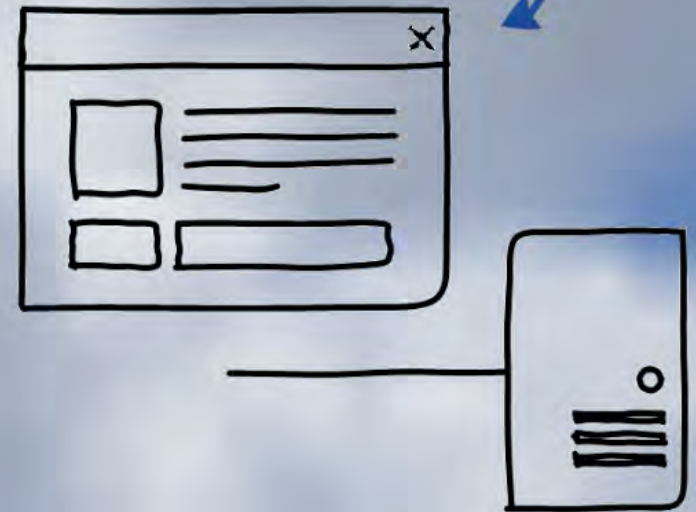
Servers may be on-premises, in the cloud, or a combination of the two



SharePoint Server



<https://localhost:44300>

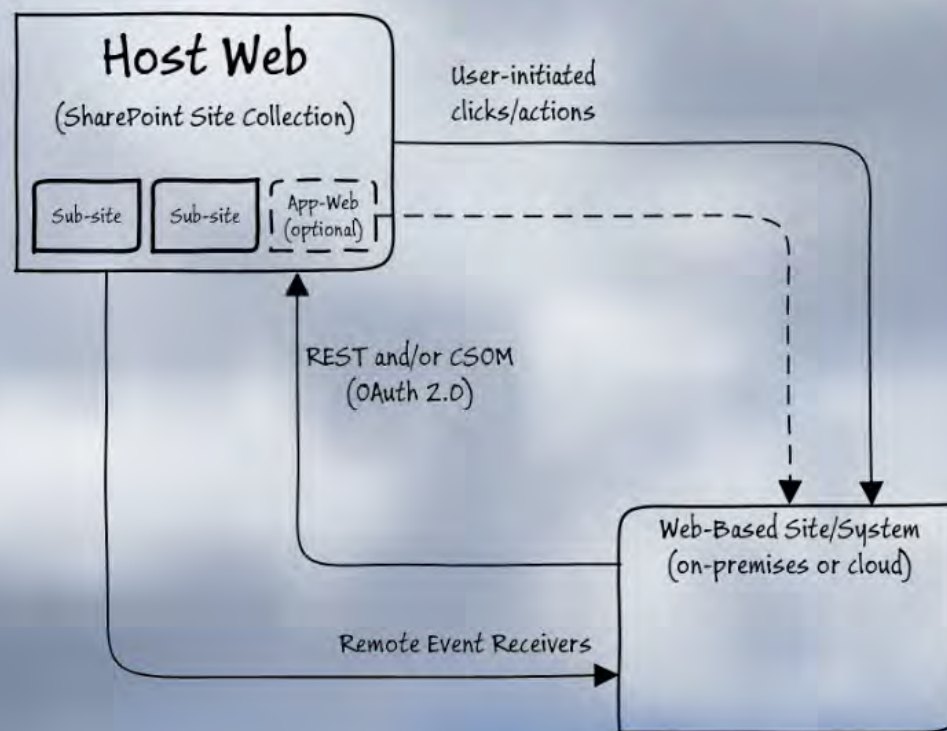


Provider's Server

<https://mcdonough.sharepoint.com/sites/dev1/>

- Scope controlled by provider implementation
- Provider chooses their technology stack
- Code can call back to SharePoint independently
- Security can be tricky (S2S Trust and OAuth 2.0)

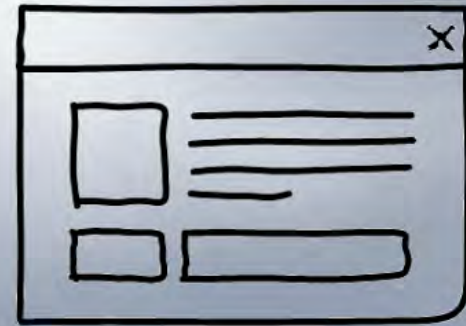
## Provider-hosted



Noteworthy  
Points



SharePoint Server



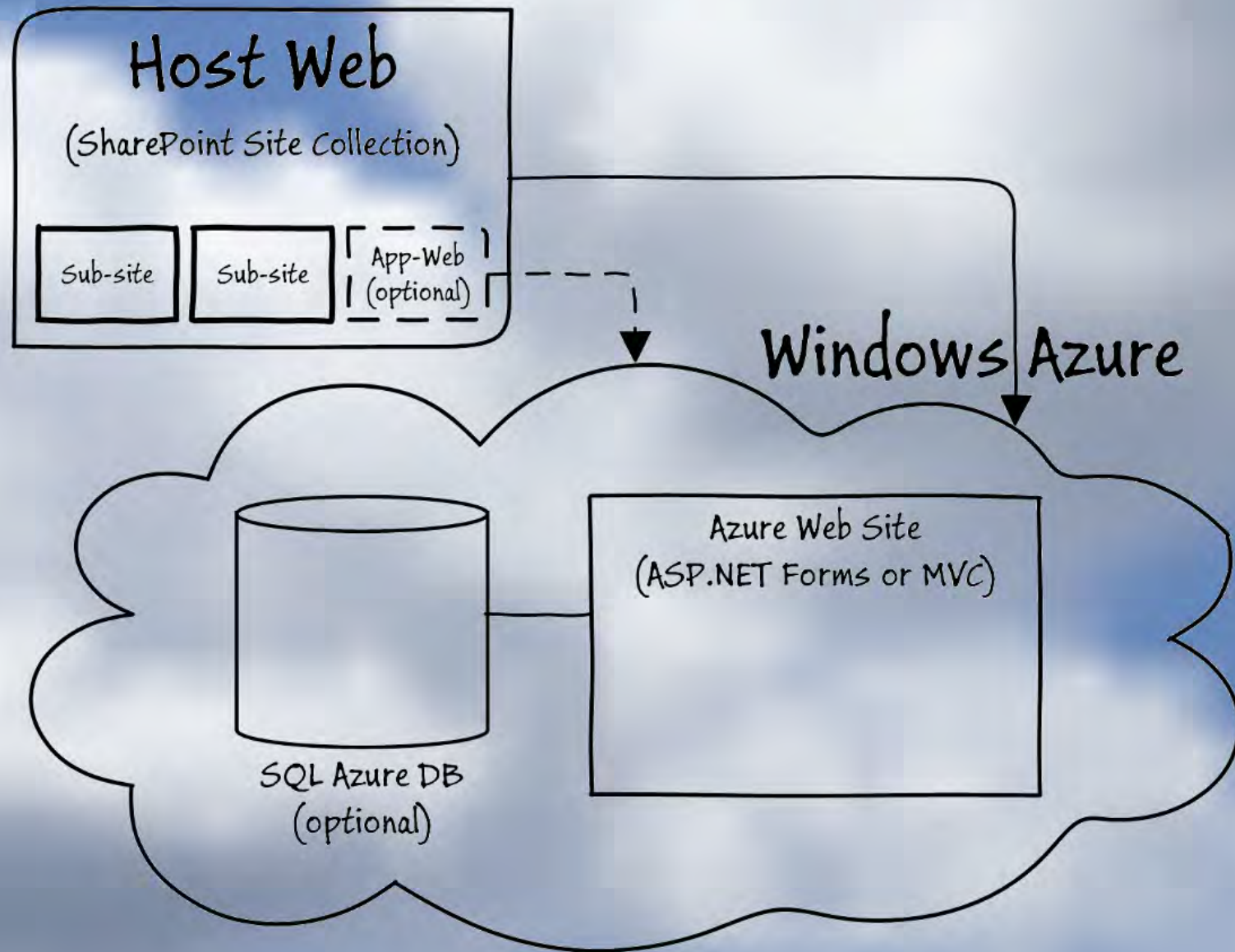
Provider's Server



**S2S Trust** (S2S = server-to-server)

**On-Premises / High Trust Model**

# Autohosted

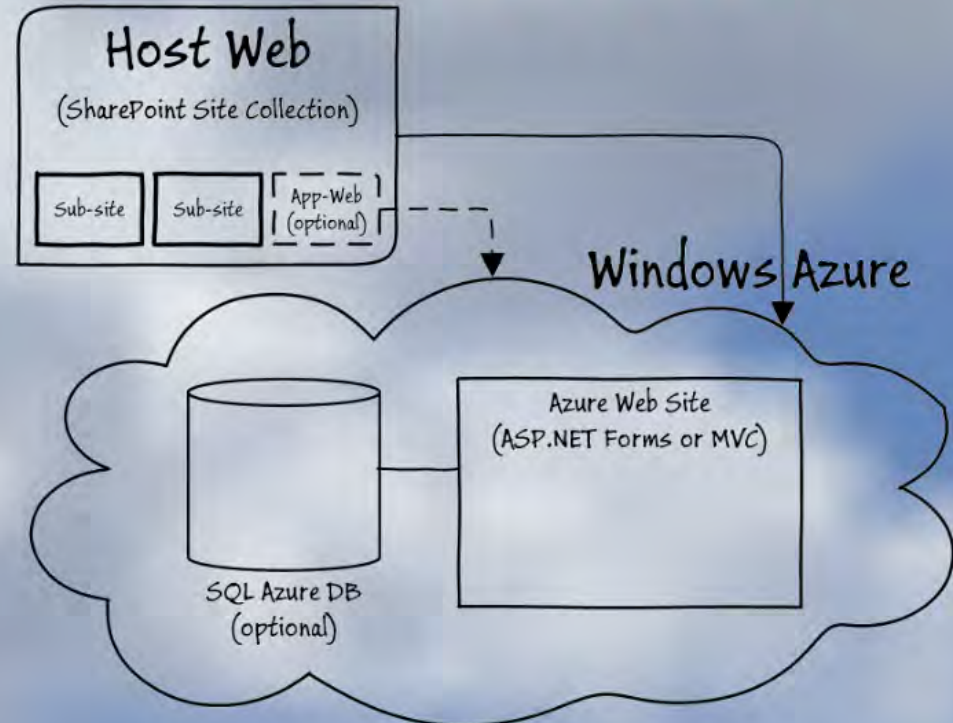


- Special case of Provider-hosted App Model
- Technology stack is Azure-based web & SQL
- Office 365-only model (for now ...)
- Security simplified by Office 365 and Azure ACS



Noteworthy  
Points

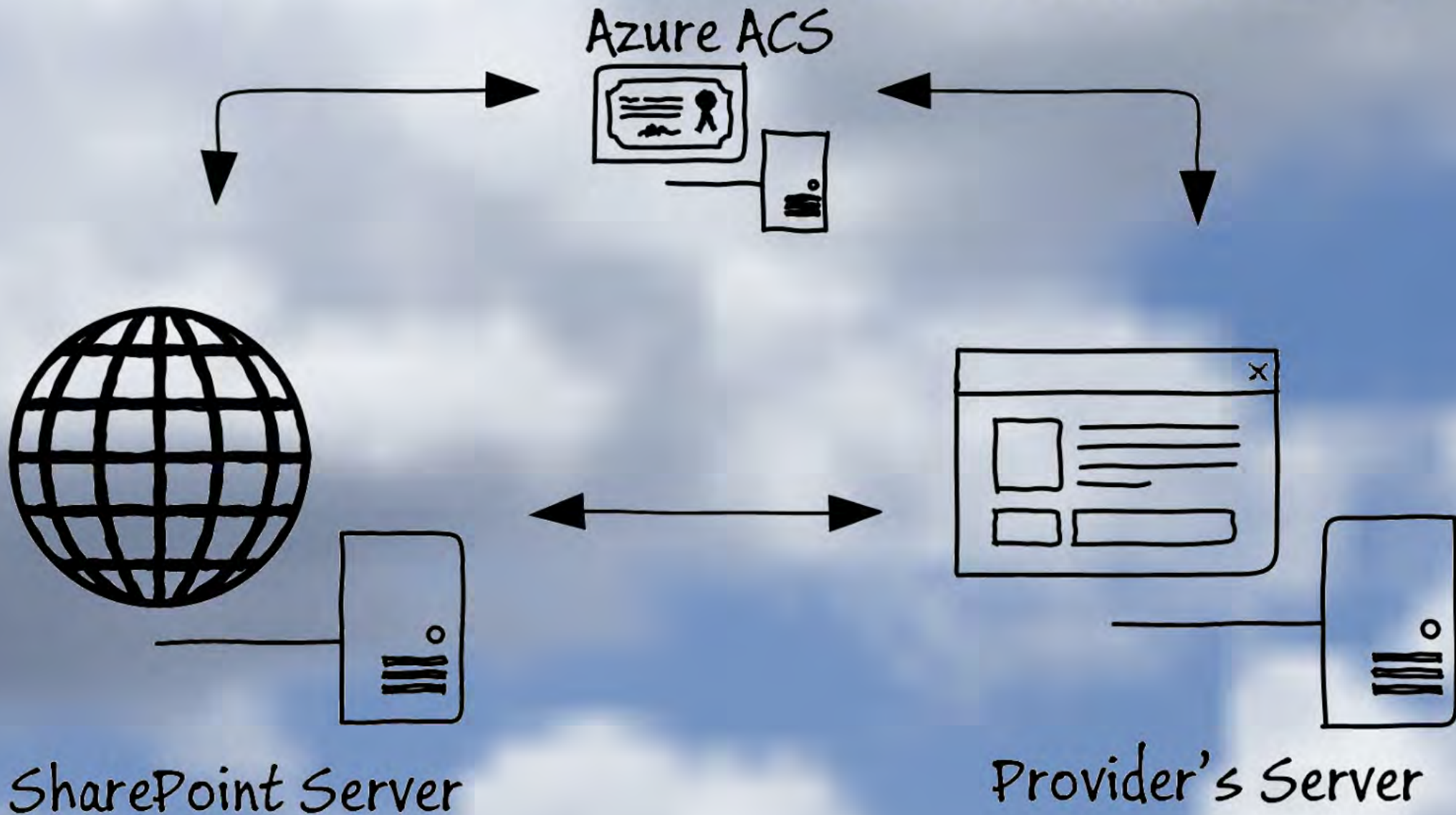
## Autohosted





# Office 365 / Cloud / Autohosted Model

## OAuth 2.0



# How Apps "Surface" in SharePoint



# A typical SharePoint 2013 team site

A Poor Man's Attempt at a SharePoint 2013 Page

http://www.no-talent-design.com/Pages/default.aspx

SharePoint joe.nobody

BROWSE PAGE SHARE FOLLOW SYNC EDIT

Sample Site Collection EDIT LINKS

Home Get started with your site REMOVE THIS

- Documents
- Recent
- Site Contents

EDIT LINKS

Share your site.

Working on a deadline?

Add lists, libraries, and other apps.

What's your style?

Your site. Your brand.

Documents

new document or drag files here

<input checked="" type="checkbox"/>	Name	Modified	Modified By
<input type="checkbox"/>	Developers are Lousy Artists	... January 31, 2013	<input type="checkbox"/> joe.nobody

Let's install an App

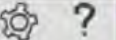


http://www.no-talent-design.com/Pages/default.aspx

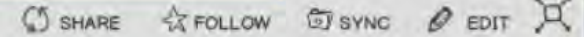


SharePoint

joe.nobody



BROWSE PAGE



Sample Site Collection EDIT LINKS

Search this site


# Sample Site Collection

- Home
- Documents
- Recent
  - My Test App
- Site Contents
- EDIT LINKS


Get started with your site REMOVE THIS




Share your site.




Working on a deadline?



Add lists, libraries, and other apps.



What's your style?



Your site. Your brand.

## Documents

+ new document or drag files here

✓	📄	Name	Modified	Modified By
	📄	Developers are Lousy Artists	... January 31, 2013	👤 joe.nobody



- App available through link
- also in Site Contents



Home

Documents

Recent

My Test App

Site Contents

 EDIT LINKS

So

Ge

Sh

Doc



# Immersive / Full Page Experience



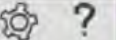


http://www.no-talent-design.com/Pages/default.aspx

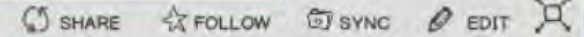


SharePoint

joe.nobody



BROWSE PAGE




Sample Site Collection [EDIT LINKS](#)

Search this site


# Sample Site Collection

- Home
- Documents
- Recent
  - My Test App
- Site Contents
- [EDIT LINKS](#)


Get started with your site [REMOVE THIS](#)




Share your site.




Working on a deadline?



Add lists, libraries, and other apps.



What's your style?



Your site. Your brand.

## Documents

[+](#) new document or drag files here

✓	Name	Modified	Modified By
	Developers are Lousy Artists	... January 31, 2013	<input type="checkbox"/> joe.nobody





Share your site.

Working on a  
deadline?

Add lists, libraries, and  
other apps.

## Documents

 new document or drag files here

<input checked="" type="checkbox"/>		Name		Modified	Modified By
		Developers are Lousy Artists	...	January 31, 2013	<input type="checkbox"/> joe.nobody

# User Interface Custom Actions

A Poor Man's Attempt at a SharePoint 2013 Page

http://www.no-talent-design.com/Pages/default.aspx

SharePoint joe.nobody

BROWSE PAGE SHARE FOLLOW SYNC EDIT

Sample Site Collection EDIT LINKS

Search this site

Home Get started with your site REMOVE THIS

Documents

Recent My Test App

Site Contents EDIT LINKS

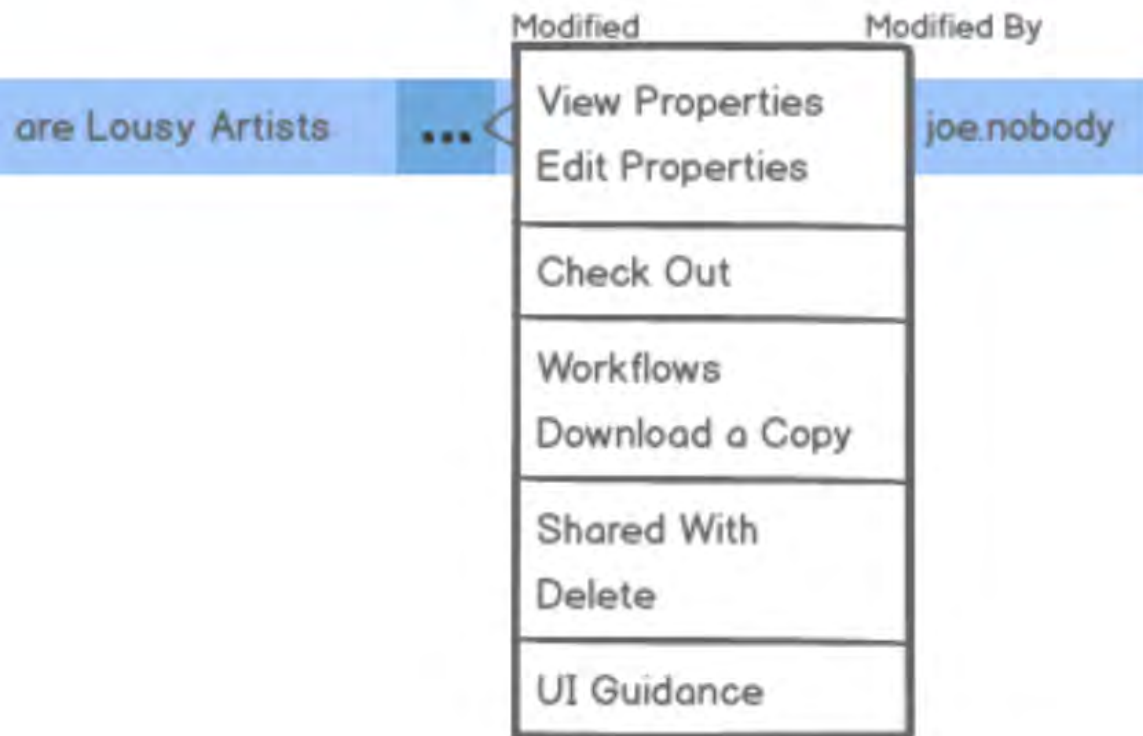
Share your site. Working on a deadline? Add lists, libraries, and other apps. What's your style? Your site. Your brand.

Documents

+ new document or drag files here

✓	Name	Modified	Modified By
✓	Developers are Lausy Artists		joe.nobody

- View Properties
- Edit Properties
- Check Out
- Workflows
- Download a Copy
- Shared With
- Delete
- UI Guidance



- this is an ECB custom action
- ribbon custom actions also possible

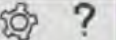


http://www.no-talent-design.com/Pages/default.aspx

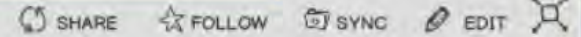


SharePoint

joe.nobody



BROWSE PAGE




Sample Site Collection EDIT LINKS

Search this site


# Sample Site Collection

- Home
- Documents
- Recent
  - My Test App
- Site Contents
- EDIT LINKS


Get started with your site REMOVE THIS




Share your site.




Working on a deadline?



Add lists, libraries, and other apps.



What's your style?



Your site. Your brand.

## Documents

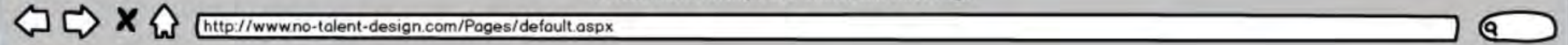
+ new document or drag files here

✓	Name	Modified	Modified By
	Developers are Lousy Artists	... January 31, 2013	□ joe.nobody



# App Parts

A Poor Man's Attempt at a SharePoint 2013 Page



SharePoint joe.nobody ▾ ⚙ ?

BROWSE PAGE SHARE FOLLOW SYNC EDIT




Sample Site Collection EDIT LINKS


## Sample Site Collection

- Home
- Documents
- Recent
  - My Test App
- Site Contents
- EDIT LINKS


Get started with your site REMOVE THIS




Share your site.




Working on a deadline?



Add lists, libraries, and other apps.



What's your style?

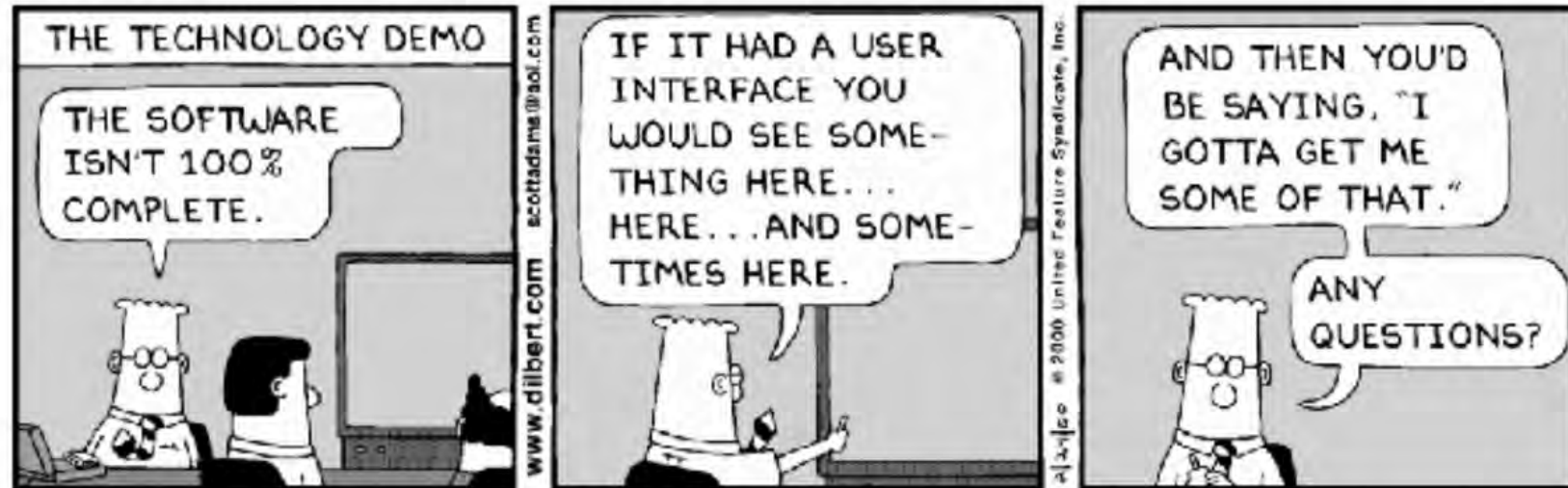


Your site. Your brand.

### My Test App



### My Test App



- handled like a web part
- basically an IFRAME in Host Web

# Back to the Cold War ...



Is there hope?

“Yes!”







Much of the agitation has to do with poor boundaries and ambiguity around custom solution responsibilities

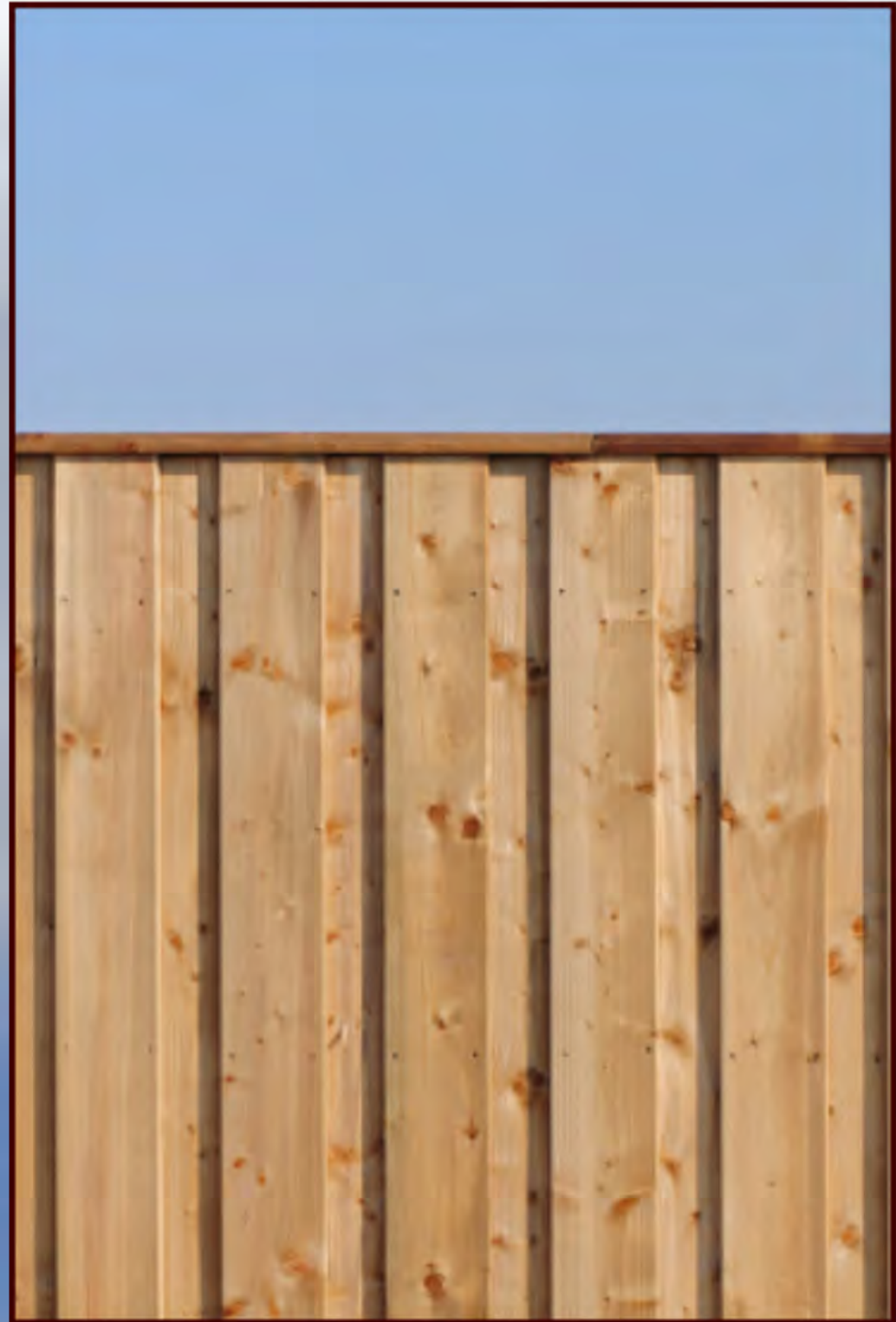
The new App Model addresses many of these issues directly



- Isolation
- Hygiene
- Standards
- Security

- Isolation
- Hygiene
- Standards
- Security

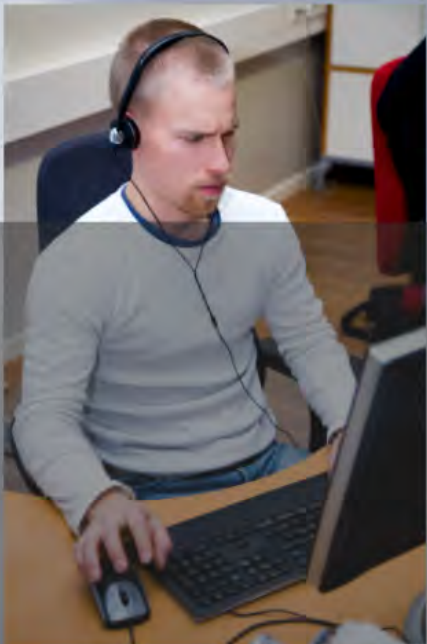
<sup>PP</sup>“Good fences  
make good  
neighbors”<sup>PP</sup>



# The New App Model

what the changes mean

Developer



Administrator



# The New App Model

## What the changes mean

Developer



Administrator



### Isolation

- No code runs on the SharePoint server - it runs on the client, "another server," or both
- Apps run from a separate App Web that has a different hostname (URL base) than its hosting SharePoint site (Host Web)
- Apps do not have direct access to Host Web SharePoint data



- Say goodbye to the server object model all access through .api via CSOM, JSON, REST, etc.
- Object model not as robust as server OM App Model is not appropriate for all app types!
- Can't take context for granted Just about everything is an explicit series of interactions with the SharePoint environment
- Greater patch and hotfix freedom Solution code only loosely tied to farm



- Say goodbye to farm performance issues related to custom code Code is now running on other servers you may or may not manage
- Troubleshooting boundaries are clearer Look at the URL - it tells you where to investigate
- Service packs, hotfixes, and CUs (generally) shouldn't break custom code Farm and solution upgrades not tightly coupled

### Hygiene

- Apps are deployed as packages that contain content and declarative markup only: HTML, ASPX pages (but no code-behind), images, XML files, list templates, etc. No DLLs.
- App data is persisted within the App Web - not the Host Web
- On uninstall, entire App Web is basically wiped out resulting in clean uninstall. Nothing remains.



- Bye-bye provisioning code, hello declarative markup! no FeatureCascaders, provisioning scripts, etc.
- Think hard about persisted values/items Particularly for SharePoint-hosted Apps, since App Webs are wiped-out on uninstall
- Can be a real asset Especially if your needs can be met with an autohosted app where provisioning is "automatic"



- Different (but overall fewer) demands for administrators and their time No longer manual WSP deploys and Feature activations, but app catalog must be managed
- Centralized management Apps may be requested by users and managed in app catalog; includes licensing support
- Administrators can put the mop away No more cleaning up dead artifacts, DLLs, lists, webs, and other items that were left behind.

### Standards

- App Model focuses on building solutions with presentation tier technologies like HTML, CSS, and JavaScript
- Back-end business logic can be built on with any stack: Microsoft/.NET, LAMP, Java/J2EE, etc.
- SharePoint data is accessed/updated through CSOM and OData-compliant (mostly) REST web service endpoints



- If you're not a "front end developer," it's time to learn some new skills HTML, CSS, JavaScript, jQuery, async patterns, etc.
- If you weren't a SharePoint (or even .NET) developer, you now have an "in" Provider-hosted Apps in particular
- This helps SharePoint's image with the rest of the world Small point to us but important when considering the viability of the platform in the long-term



- Impact to administrators isn't tremendous (versus developers). More non-microsoft, non-.NET developers may be accessing SharePoint with SharePoint 2013
- Focus becomes assisting at the web service interfaces ... and security, which is up next!

### Security

- Apps have no privileged access to SharePoint data; they go through web services
- New App Model introduces the concept of application identity (versus just "user identity")
- OAuth 2.0 is underlying framework for App and User-based trust



- You need to understand both user and application security. The full-trust, "free pass" days of RunWithElevatedPrivileges() are long gone.
- Your apps need to declare what they need It all goes in the AppManifest.xml file
- Security may be no issue at all ... or require substantial investment in time/effort App Model used determines (to a large degree) how easy or hard security will be.



- Apps require claims-based authentication Still using NTLM with SharePoint 2013. It's time to switch over.
- OAuth 2.0 and S2S Trusts will require admins to learn some new tricks Getting a farm set up for those items is a non-trivial effort
- Learn a bit about Azure and Office 365 with SharePoint 2013, there's a good chance you'll be involved in a conversation about it

# Isolation

- No code runs on the SharePoint server - it runs on the client, "another server," or both
- Apps run from a separate App Web that has a different hostname (URL base) than its hosting SharePoint site (Host Web)
- Apps do not have direct access to Host Web SharePoint data



- Say goodbye to the server object model  
all access through `_api` via CSOM, JSOM, REST, etc.
- Object model not as robust as server OM  
App Model is not appropriate for all app types!
- Can't take context for granted  
Just about everything is an explicit series of interactions with the SharePoint environment
- Greater patch and hotfix freedom  
Solution code only loosely tied to farm



- Say goodbye to farm performance issues related to custom code  
*Code is now running on other servers you may or may not manage*
- Troubleshooting boundaries are clearer  
*Look at the URL - it tells you where to investigate*
- Service packs, hotfixes, and CUs (generally) shouldn't break custom code  
*Farm and solution upgrades not tightly coupled*



# Hygiene

- Apps are deployed as packages that contain content and declarative markup only: HTML, ASPX pages (but no code-behind), images, XML files, list templates, etc. No DLLs.
- App data is persisted within the App Web - not the Host Web
- On uninstall, entire App Web is basically wiped out resulting in clean uninstall. Nothing remains.



- Buh-bye provisioning code, hello declarative markup!  
no `FeatureReceivers`, provisioning scripts, etc
- Think hard about persisted values/items  
Particularly for SharePoint-hosted Apps, since App webs are wiped-out on uninstall
- Can be a real asset  
Especially if your needs can be met with an autohosted app where provisioning is "automagic"



- Different (but overall fewer) demands for administrators and their time

No longer manual .WSP deploys and Feature activations, but app catalog must be managed

- Centralized management

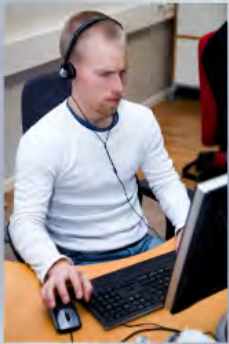
Apps may be requested by users and managed in app catalog; includes licensing support

- Administrators can put the mop away

No more cleaning up dead artifacts, DLLs, lists, webs, and other items that were left behind.

# Standards

- App Model focuses on building solutions with presentation tier technologies like HTML, CSS, and JavaScript
- Back-end business logic can be built on/with any stack: Microsoft/.NET, LAMP, Java/J2EE, etc.
- SharePoint data is accessed/updated through CSOM and OData-compliant (mostly) REST web service endpoints



- If you're not a "front end developer," it's time to learn some new skills  
HTML, CSS, JavaScript, jQuery, async patterns, etc.
- If you weren't a SharePoint (or even .NET) developer, you now have an "in"  
Provider-hosted apps in particular
- This helps SharePoint's image with the rest of the world  
Small point to us, but important when considering the viability of the platform in the long-term



- Impact to administrators isn't tremendous (versus developers).

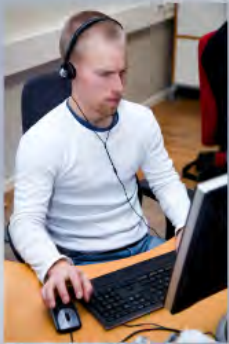
*More non-Microsoft, non-.NET developers may be accessing SharePoint with SharePoint 2013*

- Focus becomes assisting at the web service interfaces ...  
*... and security, which is up next!*

# Security

- Apps have no privileged access to SharePoint data; they go through web services
- New App Model introduces the concept of application identity (versus just "user identity")
- OAuth 2.0 is underlying framework for App and User-based trust

## The viability of the platform in the long-term



- You need to understand both user and application security.  
The full-trust, "free pass" days of `RunWithElevatedPrivileges()` are long gone.
- Your apps need to declare what they need  
It all goes in the `AppManifest.xml` file
- Security may be no issue at all ... or require substantial investment in time/effort  
App Model used determines (to a large degree) how easy or hard security will be.





- Apps require claims-based authentication  
Still using NTLM? with SharePoint 2013, it's time to switch over.
- OAuth 2.0 and S2S Trusts will require admins to learn some new tricks  
Getting a farm set up for these items is a non-trivial effort
- Learn a bit about Azure and Office 365  
with SharePoint 2013, there's a good chance you'll be involved in a conversation about it

# The New App Model

## What the changes mean

Developer



Administrator



### Isolation

- No code runs on the SharePoint server - it runs on the client, "another server," or both
- Apps run from a separate App Web that has a different hostname (URL base) than its hosting SharePoint site (Host Web)
- Apps do not have direct access to Host Web SharePoint data



- Say goodbye to the server object model all access through .api via CSOM, JSON, REST, etc.
- Object model not as robust as server OM App Model is not appropriate for all app types!
- Can't take context for granted Just about everything is an explicit series of interactions with the SharePoint environment
- Greater patch and hotfix freedom Solution code only loosely tied to farm



- Say goodbye to farm performance issues related to custom code Code is now running on other servers you may or may not manage
- Troubleshooting boundaries are clearer Look at the URL - it tells you where to investigate
- Service packs, hotfixes, and CUs (generally) shouldn't break custom code Farm and solution upgrades not tightly coupled

### Hygiene

- Apps are deployed as packages that contain content and declarative markup only: HTML, ASPX pages (but no code-behind), images, XML files, list templates, etc. No DLLs.
- App data is persisted within the App Web - not the Host Web
- On uninstall, entire App Web is basically wiped out resulting in clean uninstall. Nothing remains.



- Bye-bye provisioning code, hello declarative markup! no FeatureCascaders, provisioning scripts, etc.
- Think hard about persisted values/items Particularly for SharePoint-hosted Apps, since App Webs are wiped-out on uninstall
- Can be a real asset Especially if your needs can be met with an autohosted app where provisioning is "automatic"



- Different (but overall fewer) demands for administrators and their time No longer manual WSP deploys and Feature activations, but app catalog must be managed
- Centralized management Apps may be requested by users and managed in app catalog; includes licensing support
- Administrators can put the mop away No more cleaning up dead artifacts, DLLs, lists, webs, and other items that were left behind.

### Standards

- App Model focuses on building solutions with presentation tier technologies like HTML, CSS, and JavaScript
- Back-end business logic can be built on with any stack: Microsoft/.NET, LAMP, Java/J2EE, etc.
- SharePoint data is accessed/updated through CSOM and OData-compliant (mostly) REST web service endpoints



- If you're not a "front end developer," it's time to learn some new skills HTML, CSS, JavaScript, jQuery, async patterns, etc.
- If you weren't a SharePoint (or even .NET) developer, you now have an "in" Provider-hosted Apps in particular
- This helps SharePoint's image with the rest of the world Small point to us but important when considering the viability of the platform in the long-term



- Impact to administrators isn't tremendous (versus developers). More non-microsoft, non-.NET developers may be accessing SharePoint with SharePoint 2013
- Focus becomes assisting at the web service interfaces ... and security, which is up next!

### Security

- Apps have no privileged access to SharePoint data; they go through web services
- New App Model introduces the concept of application identity (versus just "user identity")
- OAuth 2.0 is underlying framework for App and User-based trust



- You need to understand both user and application security. The full-trust, "free pass" days of RunWithElevatedPrivileges() are long gone.
- Your apps need to declare what they need It all goes in the AppManifest.xml file
- Security may be no issue at all ... or require substantial investment in time/effort App Model used determines (to a large degree) how easy or hard security will be.



- Apps require claims-based authentication Still using NTLM with SharePoint 2013. It's time to switch over.
- OAuth 2.0 and S2S Trusts will require admins to learn some new tricks Getting a farm set up for those items is a non-trivial effort
- Learn a bit about Azure and Office 365 with SharePoint 2013, there's a good chance you'll be involved in a conversation about it



Oh yeah, one more  
area that deserves  
a mention ...

# The App Store

# The App Store

Office

Sign in | United States

HOME MY OFFICE PRODUCTS SUPPORT IMAGES TEMPLATES **STORE**

Search for apps

Office Store

What's new Office Products **Apps for Office and SharePoint...**

Excel Outlook Project **SharePoint** Word [What are apps?](#) [App Dev Center](#)

### The Quest Social Hub app for SharePoint

Track your Twitter, Facebook, and LinkedIn channels

[Try it for free](#)

Grid of app tiles for SharePoint, including Quest News, Microsoft SharePoint Sum, @Windows8, Wall Street Journal, Quest SharePoint Facebook, Quest SharePoint, My Quest SharePoint, and Fabrikam Software.

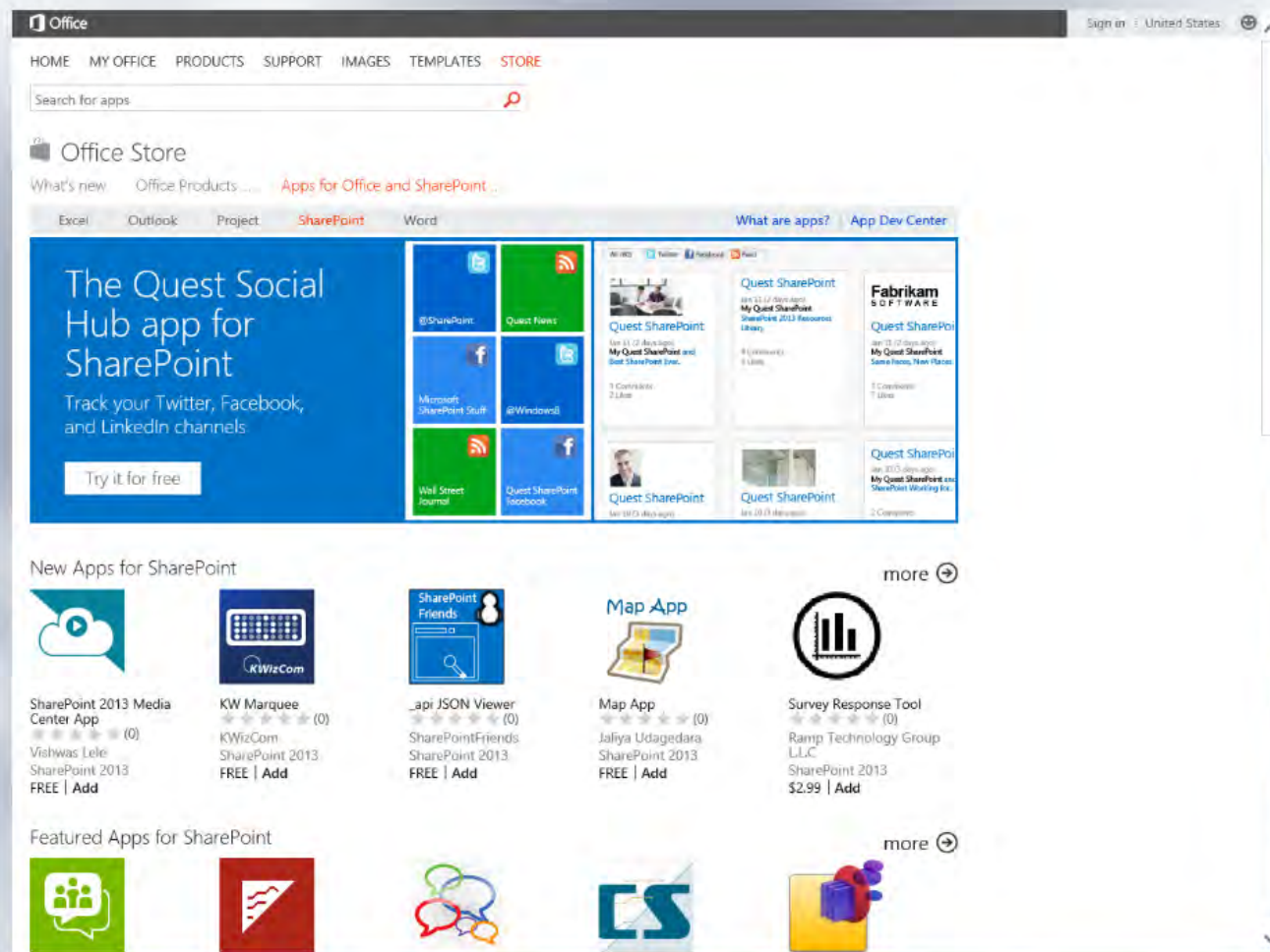
### New Apps for SharePoint

more →

 <b>SharePoint 2013 Media Center App</b> ★★★★★ (0) Vishwas Lele SharePoint 2013 <b>FREE   Add</b>	 <b>KW Marquee</b> ★★★★★ (0) KWizCom SharePoint 2013 <b>FREE   Add</b>	 <b>SharePoint Friends</b> ★★★★★ (0) _api JSON Viewer SharePointFriends SharePoint 2013 <b>FREE   Add</b>	 <b>Map App</b> ★★★★★ (0) Jaliya Udagedara SharePoint 2013 <b>FREE   Add</b>	 <b>Survey Response Tool</b> ★★★★★ (0) Ramp Technology Group L.L.C SharePoint 2013 <b>\$2.99   Add</b>
--	---	---	---	---

### Featured Apps for SharePoint

more →



The screenshot displays the Office Store interface for SharePoint. At the top, there are navigation links: HOME, MY OFFICE, PRODUCTS, SUPPORT, IMAGES, TEMPLATES, and STORE. A search bar is present with the text "Search for apps". Below this, the "Office Store" header is visible, followed by "What's new" and "Office Products". The main content area is titled "The Quest Social Hub app for SharePoint" and includes a "Try it for free" button. Below this, there are sections for "New Apps for SharePoint" and "Featured Apps for SharePoint". The "New Apps for SharePoint" section lists several apps: SharePoint 2013 Media Center App, KW Marquee, \_api JSON Viewer, Map App, and Survey Response Tool. The "Featured Apps for SharePoint" section shows icons for various apps.

- Only apps built under the new App Model can appear in the App Store
- Apps built under the new App Model are the only option for custom code in Office 365

# The One Slide Sum-Up

- Many pain points and areas of friction between developers and administrators arose from the mechanisms provided for SharePoint customization
- The new App Model addresses many of these problems by fundamentally changing the way SharePoint is customized and extended
- Not every development challenge can be tackled with the new App Model, but many can with some re-thinking in how solutions are architected and built
- The new App Model is the only way forward for Office 365, the App Store, and SharePoint's cloud-centric future

# 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))

## Why Administrators will Giggle Like Schoolgirls About SharePoint 2013's New App Model

- <http://blog.idera.com/sharepoint/why-administrators-will-giggle-like-schoolgirls-about-sharepoint-2013s-new-app-model/>

## Build apps for Office and SharePoint

- <http://msdn.microsoft.com/en-us/office/apps/fp160950.aspx>

## UX design for apps in SharePoint 2013

- <http://msdn.microsoft.com/en-us/library/office/apps/fp179934>

# Important aspects of the app for SharePoint architecture and development landscape

- <http://msdn.microsoft.com/en-us/library/office/apps/fp179922>

# How to: create custom actions to deploy with apps for SharePoint

- <http://msdn.microsoft.com/en-us/library/jj163954.aspx>

# Custom Ribbon Button Image Limitations with SharePoint 2013 Apps

- <http://sharepointinterface.com/2013/01/22/custom-ribbon-button-image-limitations-with-sharepoint-2013-apps/>

# What's new in authentication for SharePoint 2013

- <http://technet.microsoft.com/en-us/library/jj219758.aspx>



## Configure an environment for apps for SharePoint (SharePoint 2013)

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

## Introducing the SharePoint 2013 Application Model

- <http://www.binarywave.com/blogs/eshupps/Lists/Posts/Post.aspx?ID=266>

## Apps for SharePoint (Office Store)

- <http://office.microsoft.com/en-us/store/apps-for-sharepoint-FX102804987.aspx>



# Sean P. McDonough

SharePoint Gearhead,  
Developer and Evangelist.

My  
Company:



Twitter: @spmcdonough

About: <http://about.me/spmcdonough>

Blog: <http://SharePointInterface.com>