

Will Apps End the SharePoint Cold War?
Sean McDonough



Sean P. McDonough

SharePoint Gearhead.

Developer. Author. Evangelist.

Twitter: @spmcdonough

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

Blog: <http://SharePointInterface.com>

Book: <http://tinyurl.com/SPDRGuide2010>

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?



Developer vs. Administrator

Devs (on themselves)

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

Devs (on admins)

- They're a bunch of trolls!
- They come up with policies just to block our solutions
- They just seem to want more and more documentation





Admins (on themselves)

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

Admins (on devs)

- Their custom code tanks my farm's performance
- They need to learn how security works
- The next time I hear "it worked on my machine," I'm going to kill someone!

Cold War?

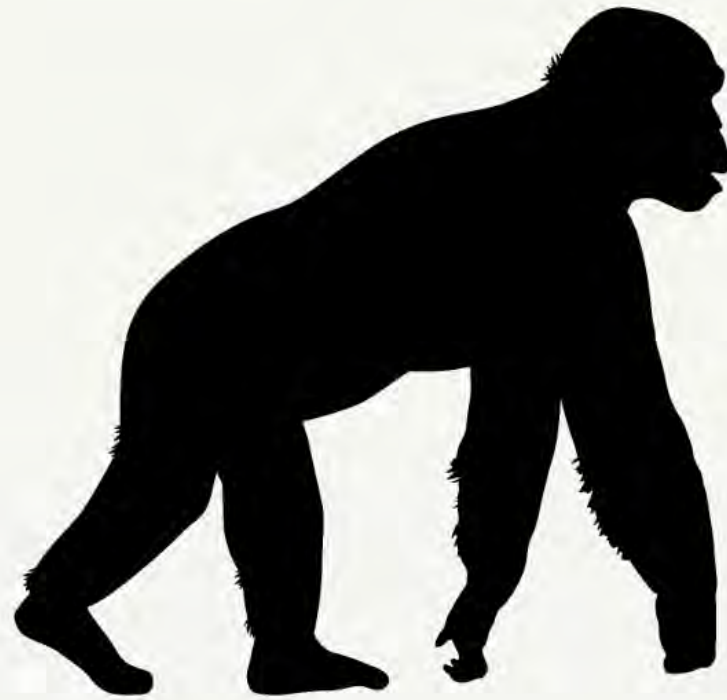


Developer vs. Administrator

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 the 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 "talks to itself"
- OAuth 2.0, underlying framework for App Web trust, added later

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 install, nothing remains



about Apps?



Are you a developer?

Get the full App Model experience to build the next wave of SharePoint solutions.

Learn more at www.microsoft.com/sharepoint/apps

Download the App Model experience

Get the full App Model experience to build the next wave of SharePoint solutions.

Learn more at www.microsoft.com/sharepoint/apps

Download the App Model experience

Get the full App Model experience to build the next wave of SharePoint solutions.

Learn more at www.microsoft.com/sharepoint/apps

Download the App Model experience

Get the full App Model experience to build the next wave of SharePoint solutions.

Learn more at www.microsoft.com/sharepoint/apps

Download the App Model experience

Get the full App Model experience to build the next wave of SharePoint solutions.

Learn more at www.microsoft.com/sharepoint/apps

Download the App Model experience

Get the full App Model experience to build the next wave of SharePoint solutions.

Learn more at www.microsoft.com/sharepoint/apps

Download the App Model experience

Get the full App Model experience to build the next wave of SharePoint solutions.

Learn more at www.microsoft.com/sharepoint/apps

Download the App Model experience

Get the full App Model experience to build the next wave of SharePoint solutions.

Standards

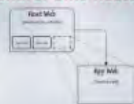
- App Model focuses on building solutions with accessible, open technologies like HTML, CSS and JavaScript
- Backend business logic can be built with any stack: Microsoft/.NET, LAMP, Java/JEE, etc.
- SharePoint data is accessed/updated through ECOW and OData compliant (proving) REST web service requests

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



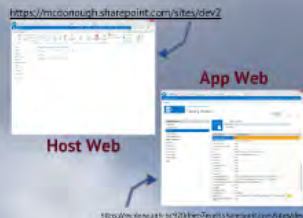
Noteworthy Points

- 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

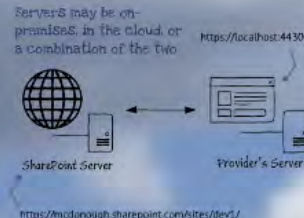


Noteworthy Points



Host Web

App Web



SharePoint Server

Provider's Server



Office 365 / Cloud / Autohosted Model

OAuth 2.0

S2S Trust

On-Premises / High Trust Model

How Apps "Surface" in SharePoint



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"> 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 install, nothing remains 	<ul style="list-style-type: none"> 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 install, nothing remains 	<ul style="list-style-type: none"> App Model focuses on building solutions with accessible, open technologies like HTML, CSS and JavaScript Backend business logic can be built with any stack: Microsoft/.NET, LAMP, Java/JEE, etc. SharePoint data is accessed/updated through ECOW and OData compliant (proving) REST web service requests 	<ul style="list-style-type: none"> Apps have no privileged access to SharePoint data; they go through web services Now App Model introduces the concept of application identity, which just "talks to itself" OAuth 2.0, underlying framework for App Web trust, added later

Oh yeah, one more area that deserves a mention

The App Store

- 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 us to download in Office 365

BROWSE FILES LIBRARY

Modify View Current View: All Documents

View Quick Edit Create View Create Column Navigate Up

E-mail a Link Alert Me RSS Feed Most Popular Items Tags & Notes

Export to Excel Open with Explorer

Form Web Parts Edit Library New Quick Step Library Settings

- 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 Find a file

✓	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
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



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

Development Site 2

BROWSE



Idera SharePoint Library Analyzer

Library Detail

Available Libraries

App Packages

Converted Forms

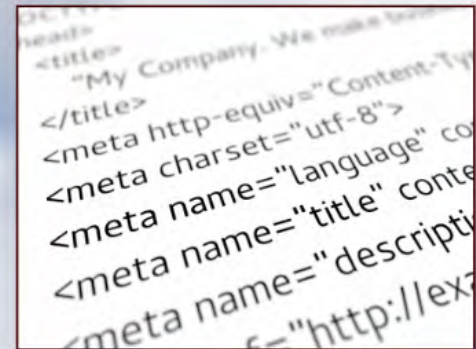
What's so special



Key App Model Aspects

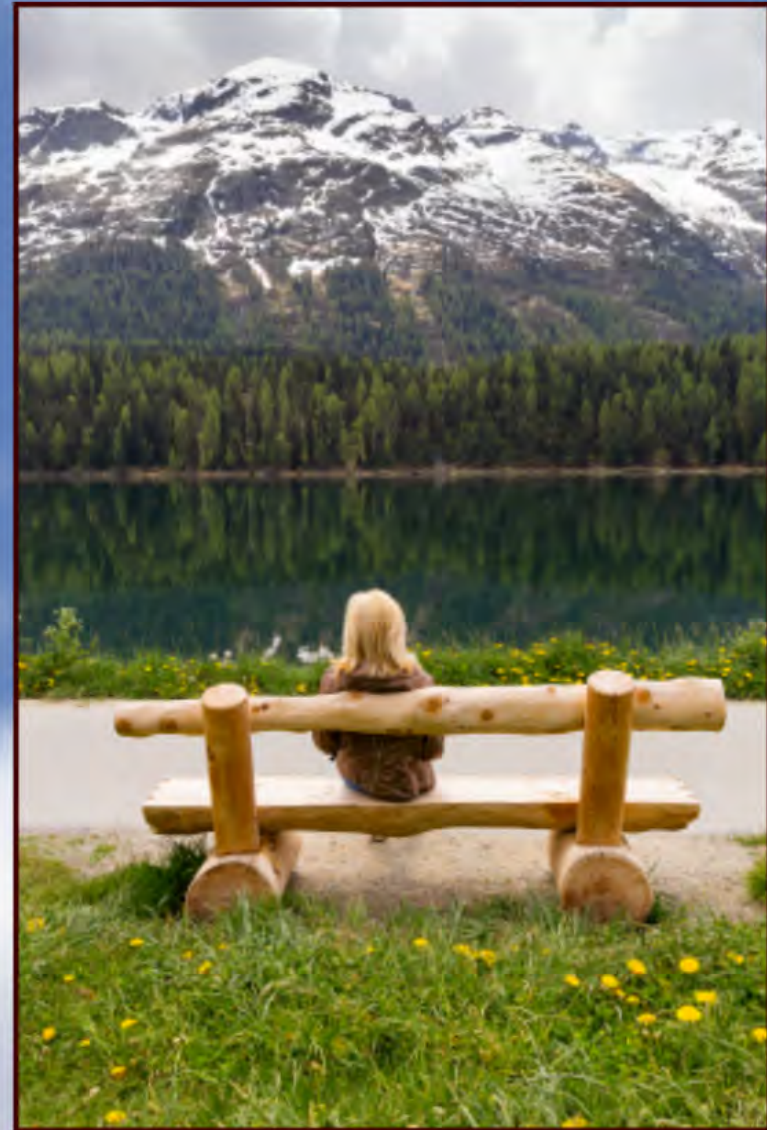


about Apps?



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 ...

Are you a developer?

If so, you might be wondering ...



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



I understand.

A blue sky with white clouds and a faint, large-scale grid pattern. The text "I'm here to help." is centered in the middle of the image.

I'm here to help.



Dr. Elisabeth Kübler-Ross

The Five

Stages of Grief

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

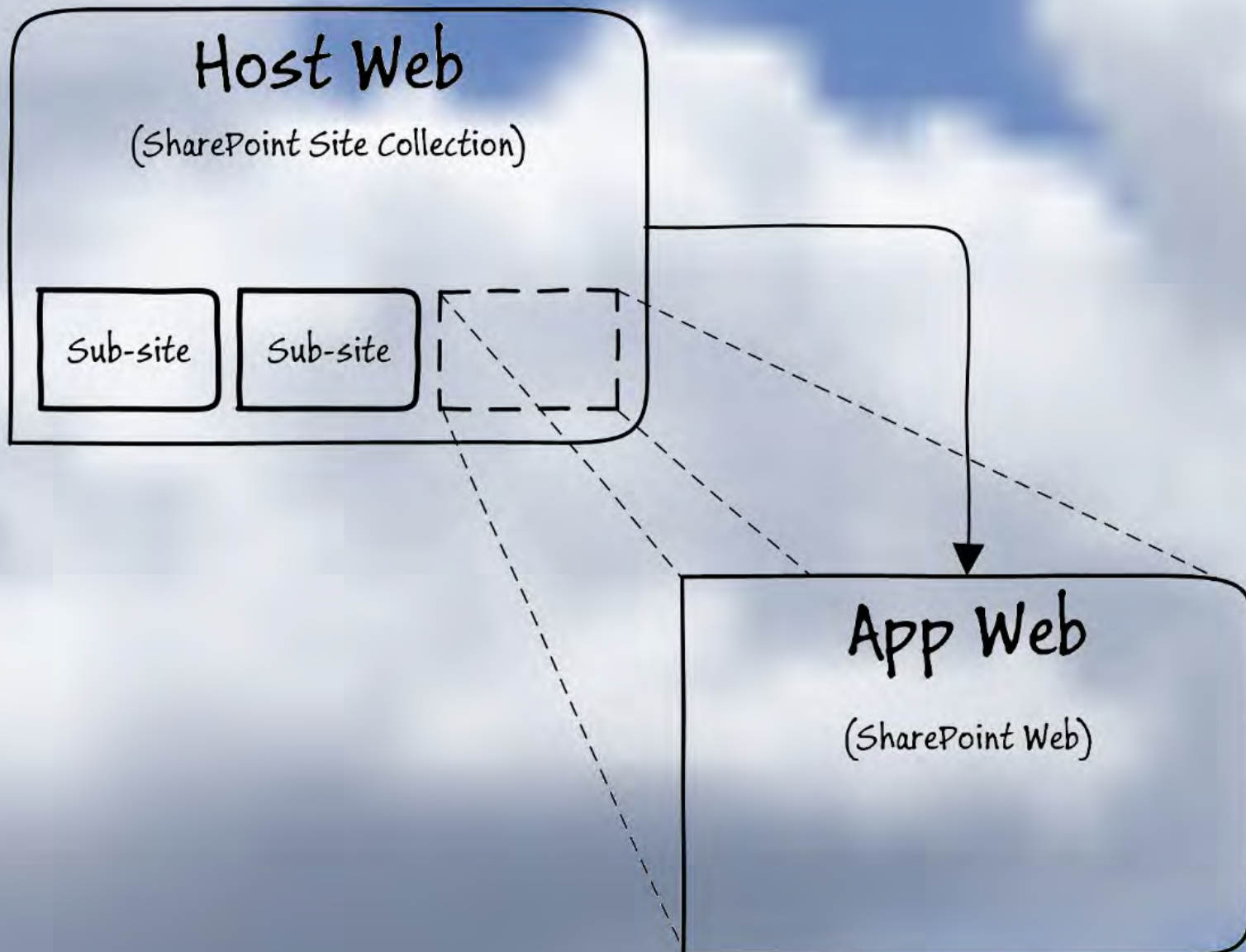


There are 3

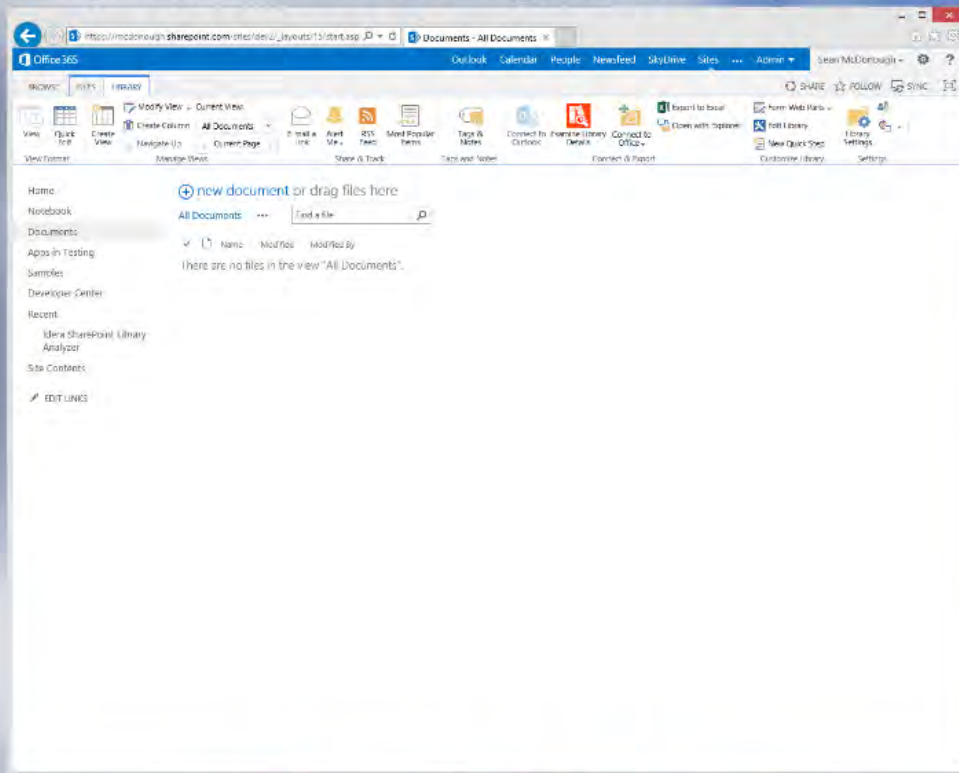
App Model Hosting Options



SharePoint-hosted

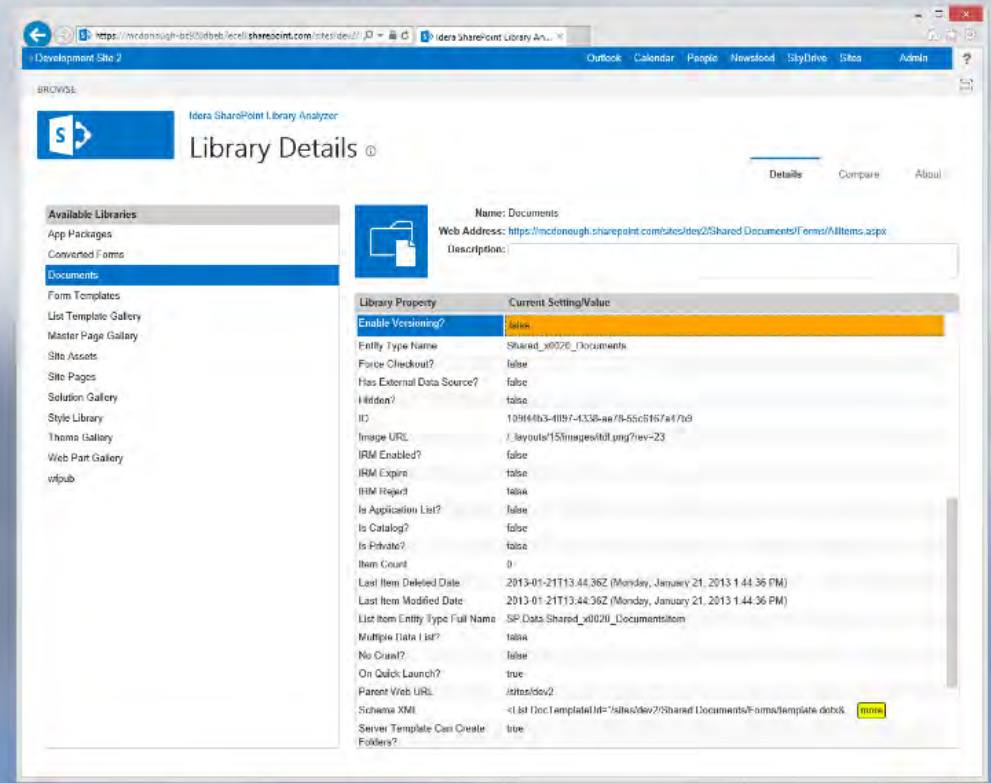


<https://mcdonough.sharepoint.com/sites/dev2>



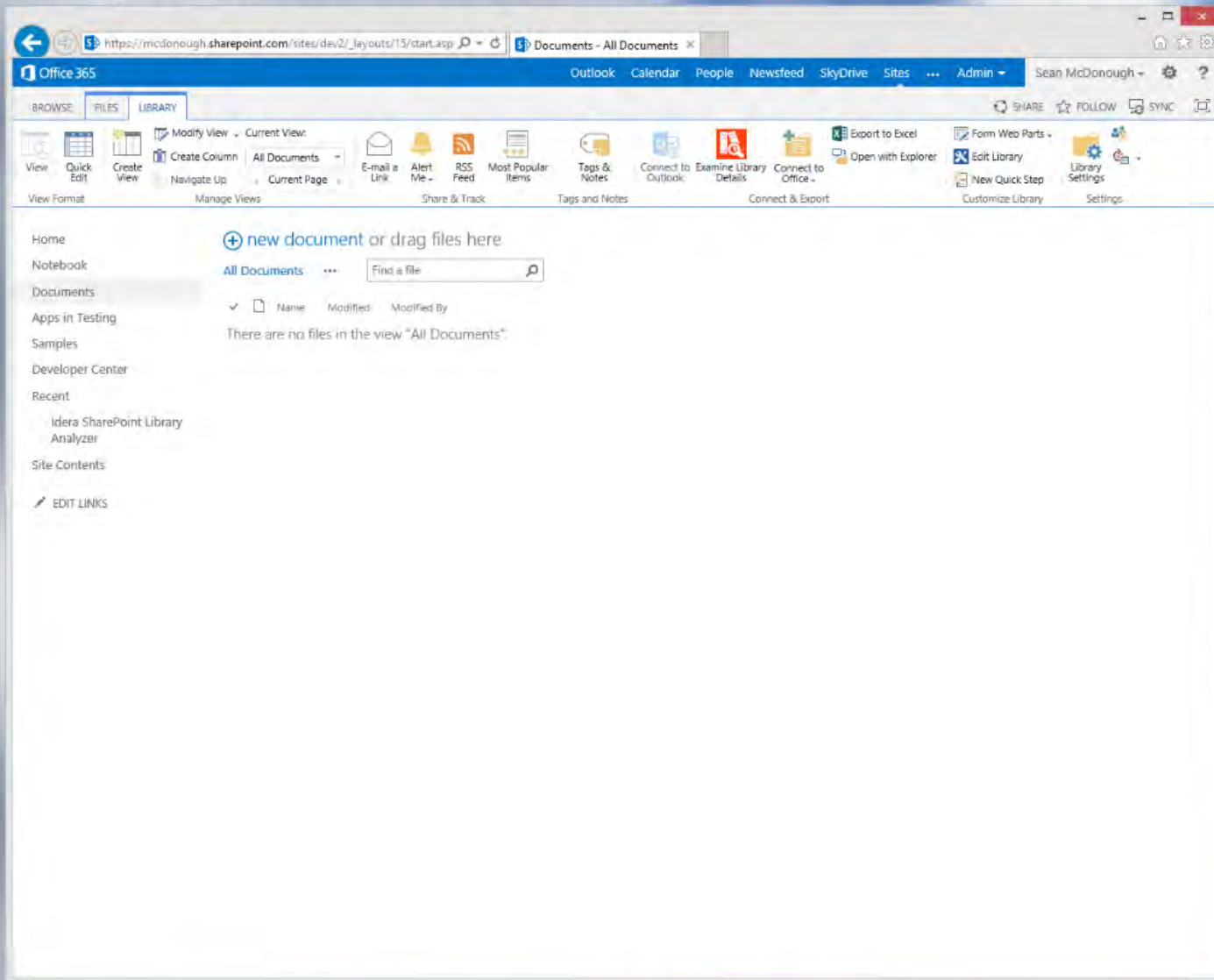
Host Web

App Web

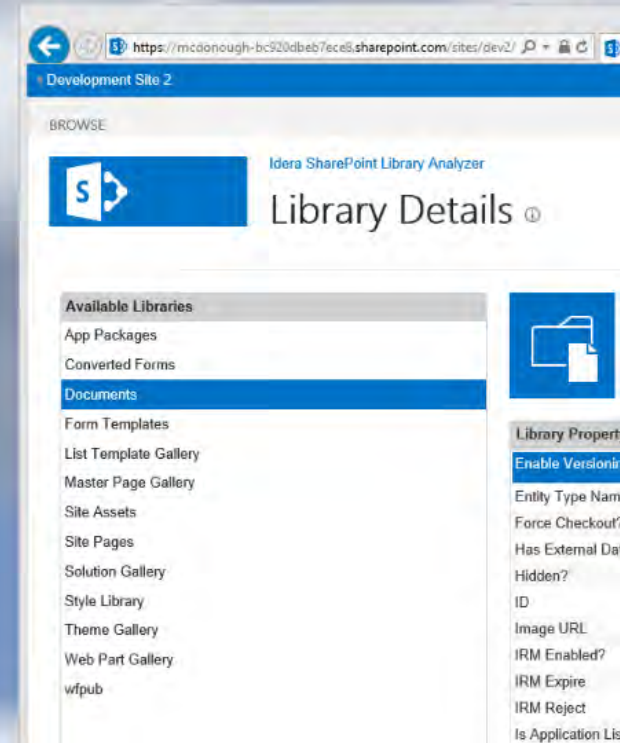


<https://mcdonough-bc920dbeb7ece8.sharepoint.com/sites/dev2>

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



Host Web



App

App Web

The screenshot shows the Idera SharePoint Library Analyzer web application. The browser address bar displays the URL: <https://mcdonough-bc920dbeb7ece8.sharepoint.com/sites/dev2/>. The application header includes navigation links for Outlook, Calendar, People, Newsfeed, SkyDrive, Sites, and Admin. The main content area is titled "Library Details" and shows information for a library named "Documents".

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
- wpub

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

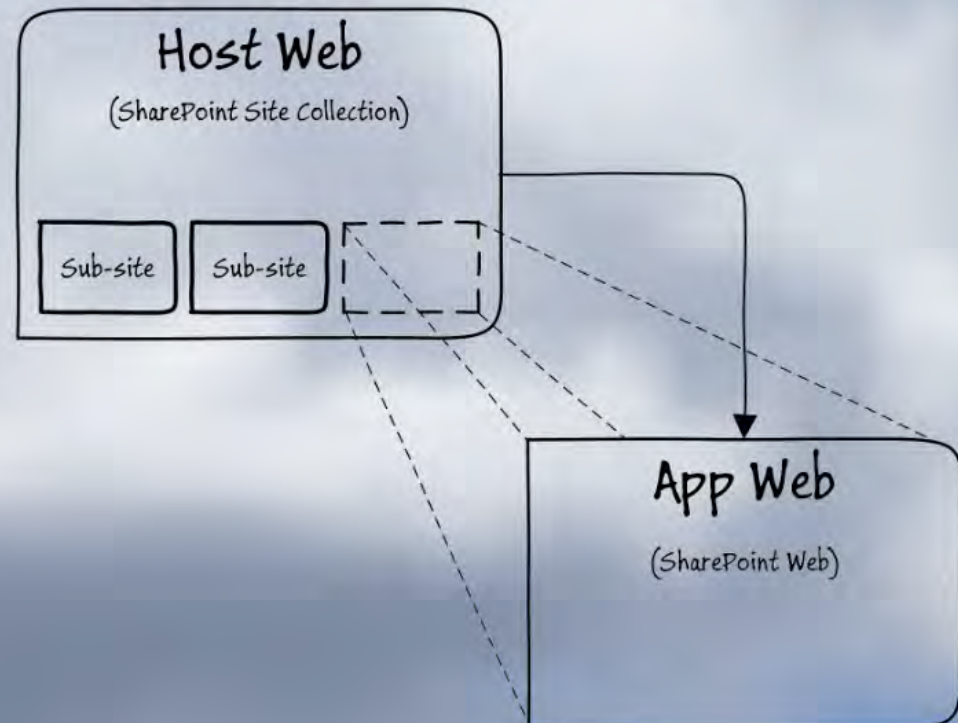
Web



<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)

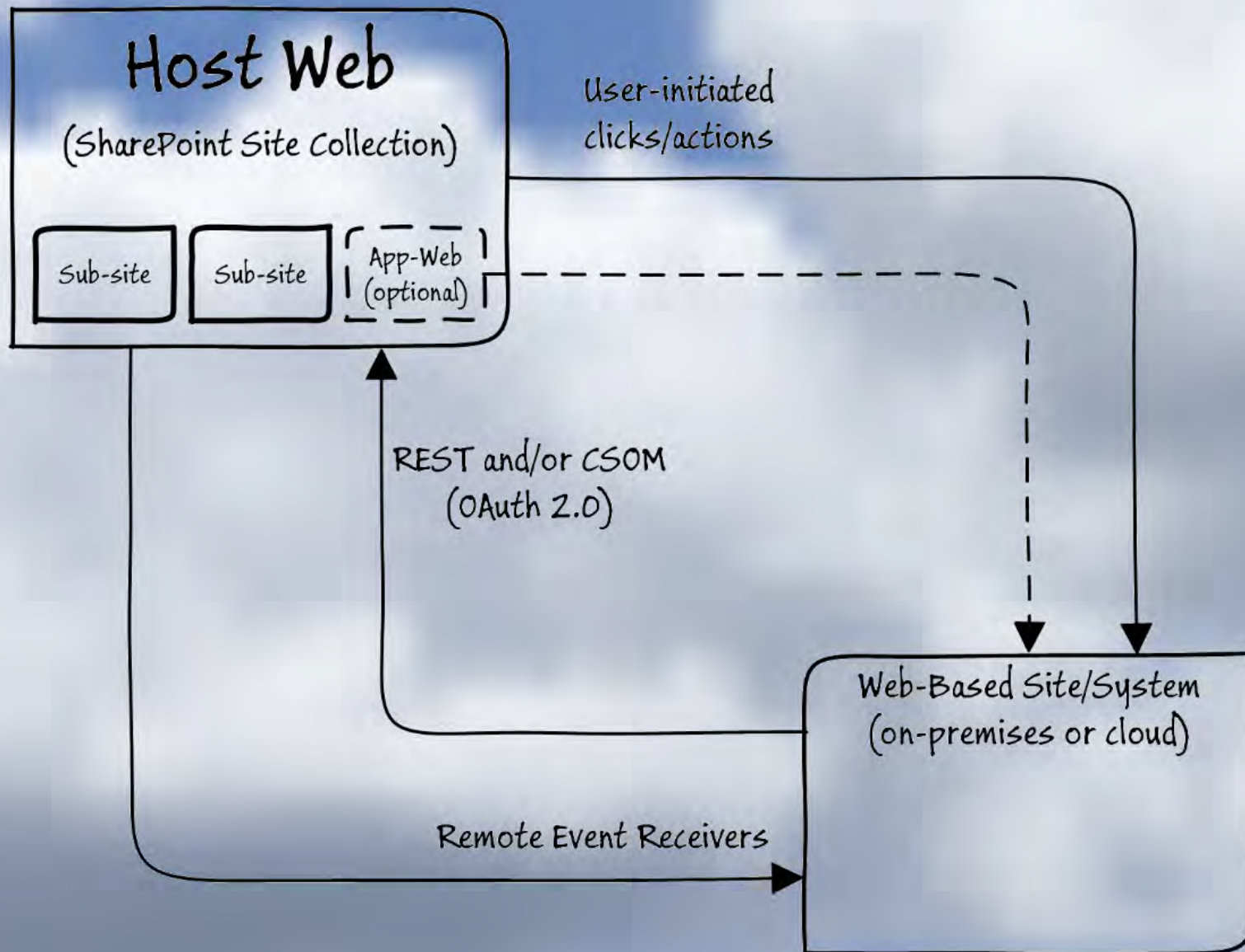
SharePoint-hosted



↑

Noteworthy
Points

Provider-hosted



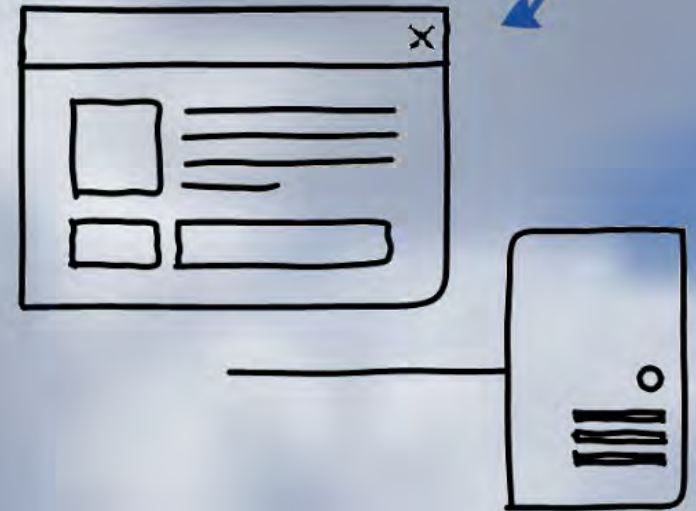
Servers may be on-premises, in the cloud, or a combination of the two



SharePoint Server



<https://localhost:44300>

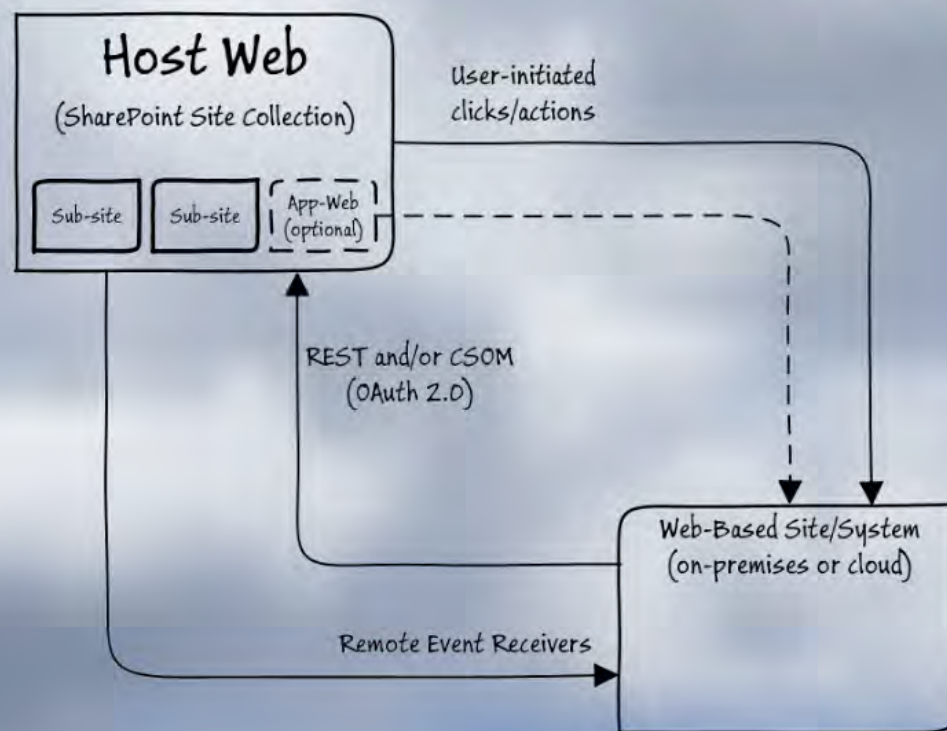


Provider's Server

<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)

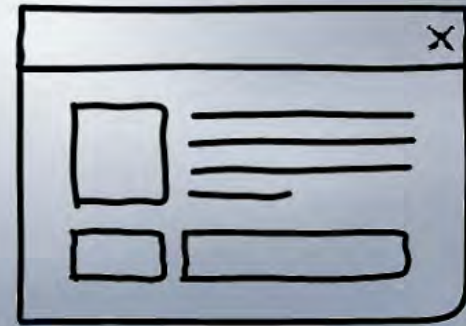
Provider-hosted



Noteworthy
Points



SharePoint Server



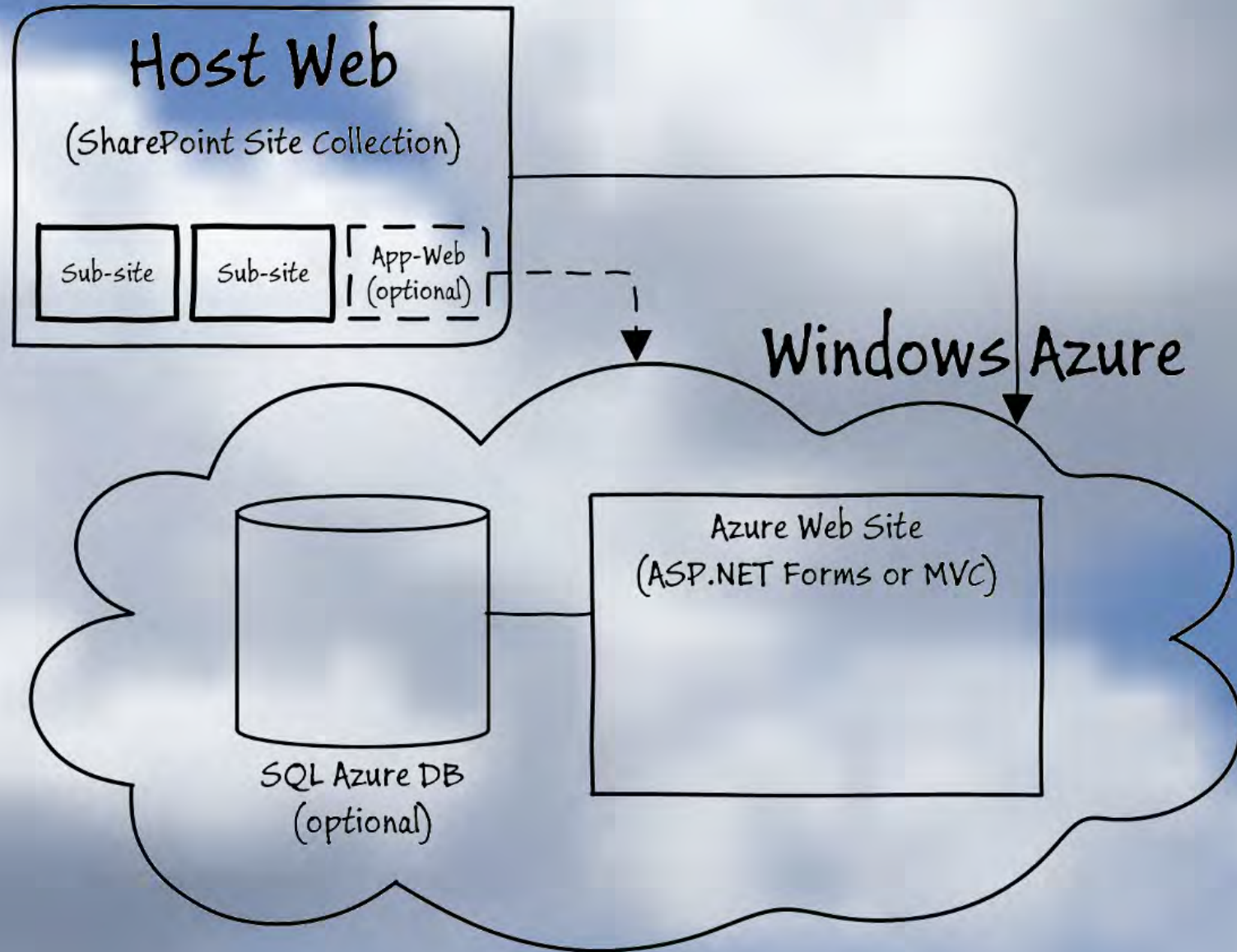
Provider's Server



S2S Trust (S2S = server-to-server)

On-Premises / High Trust Model

Autohosted

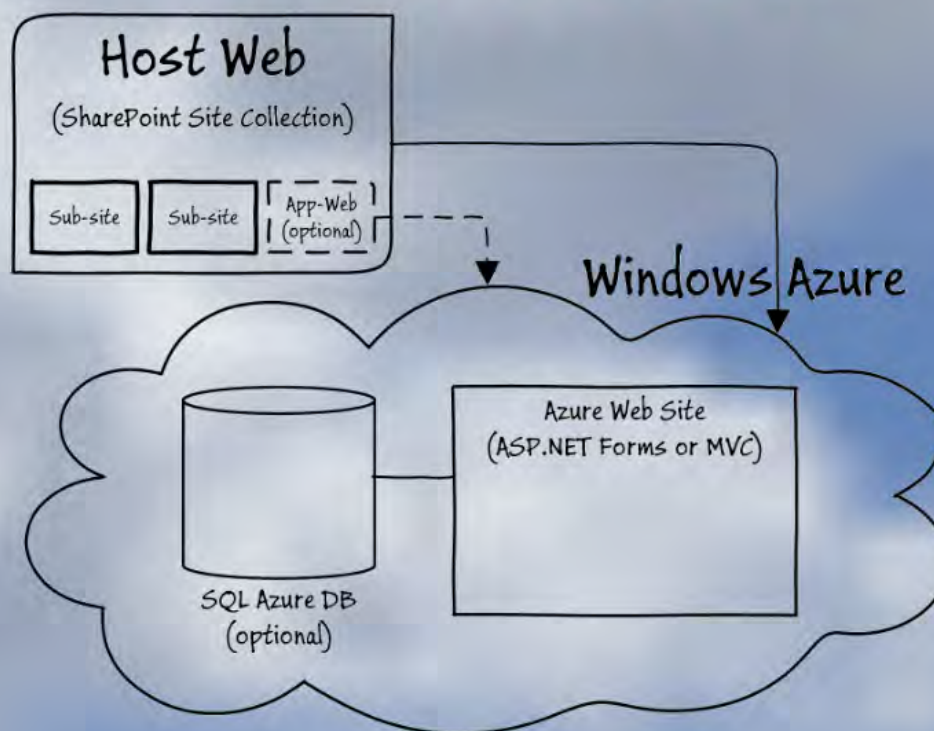


- Special case of Provider-hosted App Model
- Technology stack is Azure-based web & SQL
- Office 365-only model (for now ...)
- Security simplified by Office 365 and Azure ACS



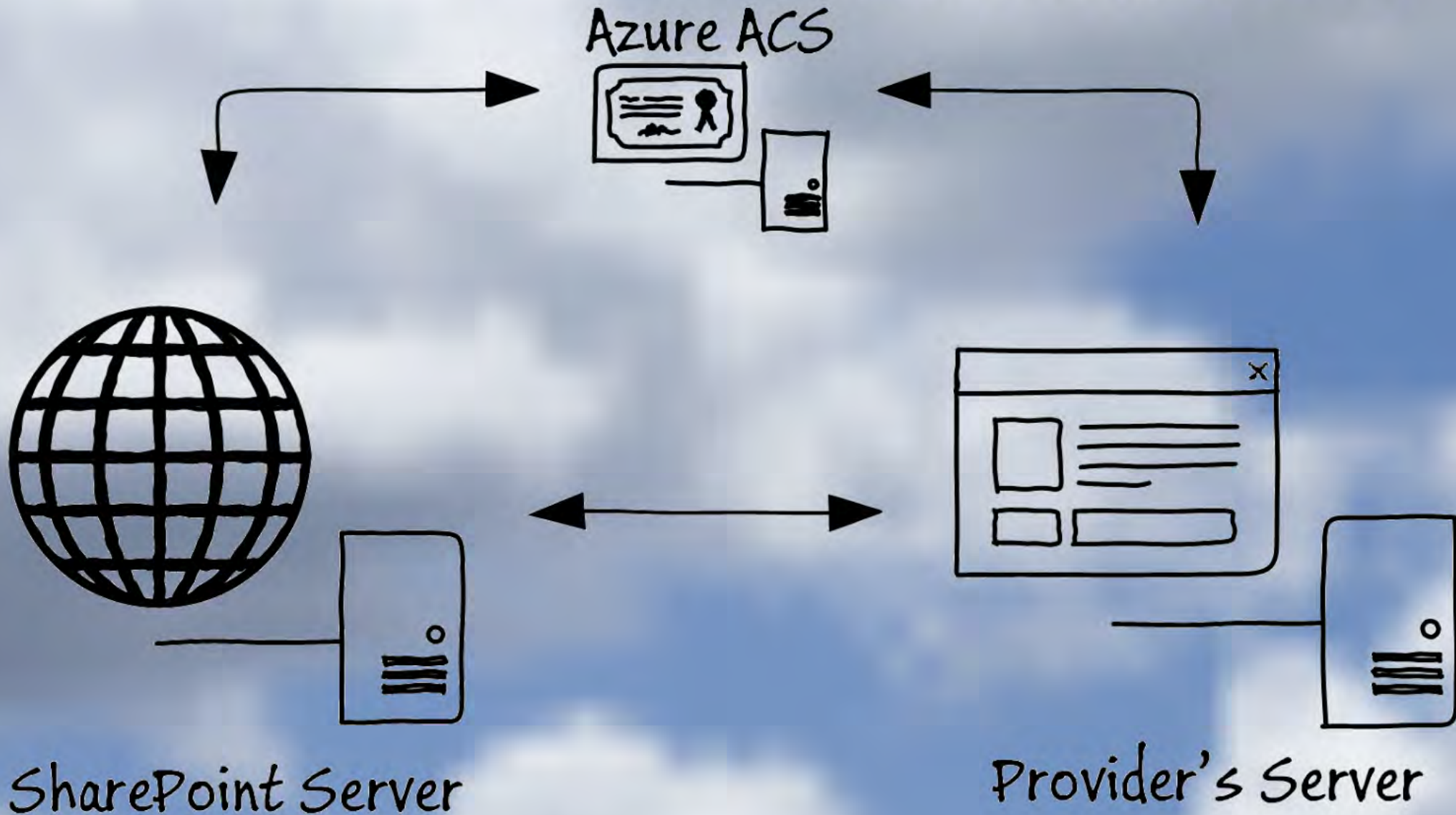
Noteworthy
Points

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

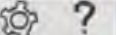


http://www.no-talent-design.com/Pages/default.aspx

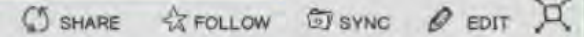


SharePoint

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



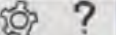


http://www.no-talent-design.com/Pages/default.aspx

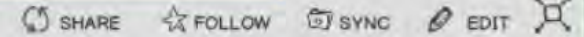


SharePoint

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	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

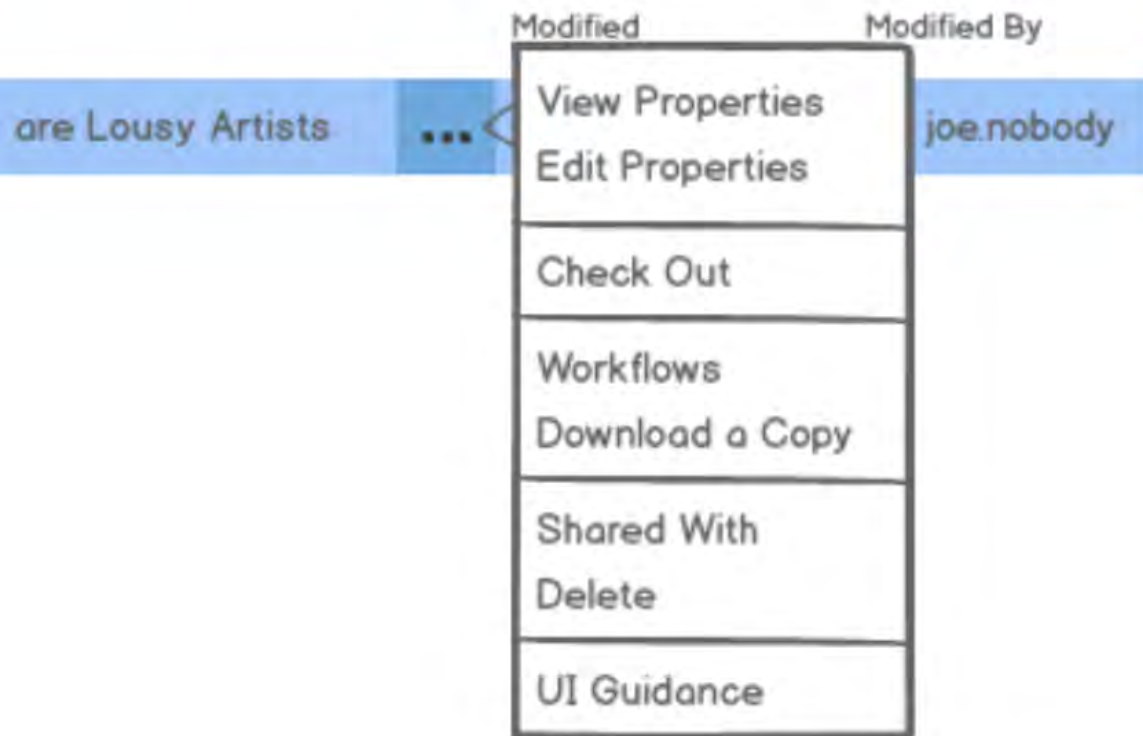
Share your site. Working on a deadline? Add lists, libraries, and other apps. What's your style? Your site. Your brand.

Documents

+ new document or drag files here

✓	Name	Modified	Modified By
✓	Developers are Lousy 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

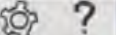


http://www.no-talent-design.com/Pages/default.aspx

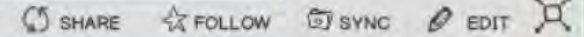


SharePoint

joe.nobody



BROWSE PAGE



Sample Site Collection [EDIT LINKS](#)

Search this site

Sample Site Collection

Home

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 Lousy Artists	... January 31, 2013	joe.nobody



App Parts

A Poor Man's Attempt at a SharePoint 2013 Page

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

SharePoint joenobody

BROWSE PAGE SHARE FOLLOW SYNC EDIT

Sample Site Collection EDIT LINKS

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

THE TECHNOLOGY DEMO

THE SOFTWARE ISN'T 100% COMPLETE.

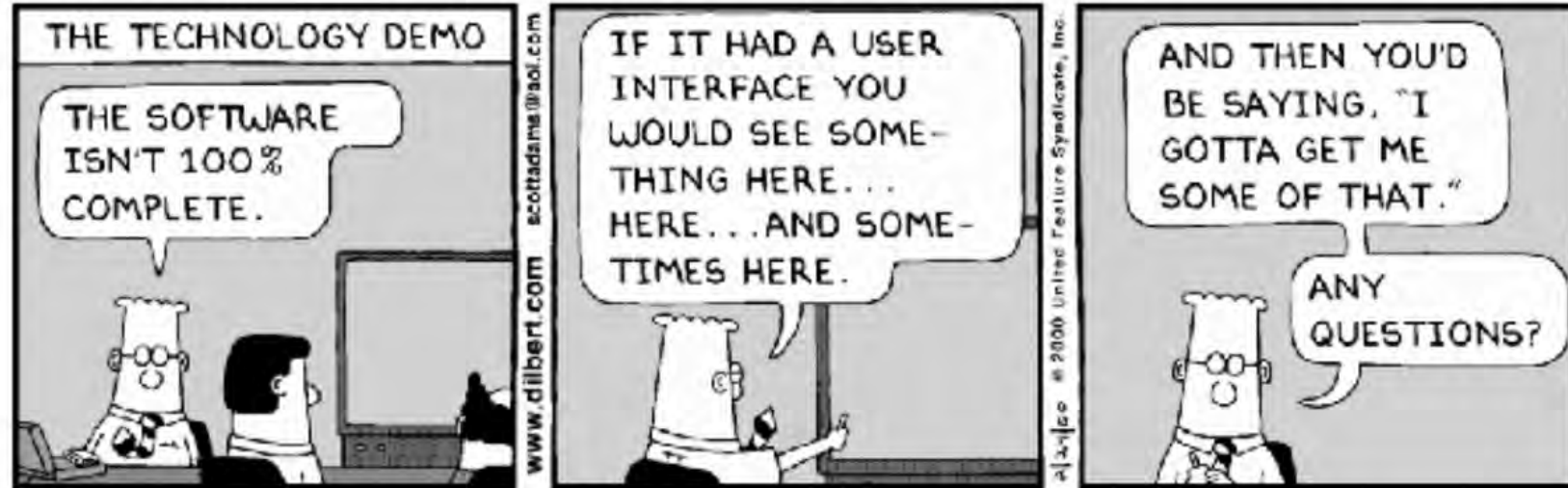
IF IT HAD A USER INTERFACE YOU WOULD SEE SOMETHING HERE... HERE... AND SOMETIMES HERE.

AND THEN YOU'D BE SAYING, "I GOTTA GET ME SOME OF THAT."

ANY QUESTIONS?

www.dilbert.com scottadams@ad.com © 2000 United Feature Syndicate, Inc.

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

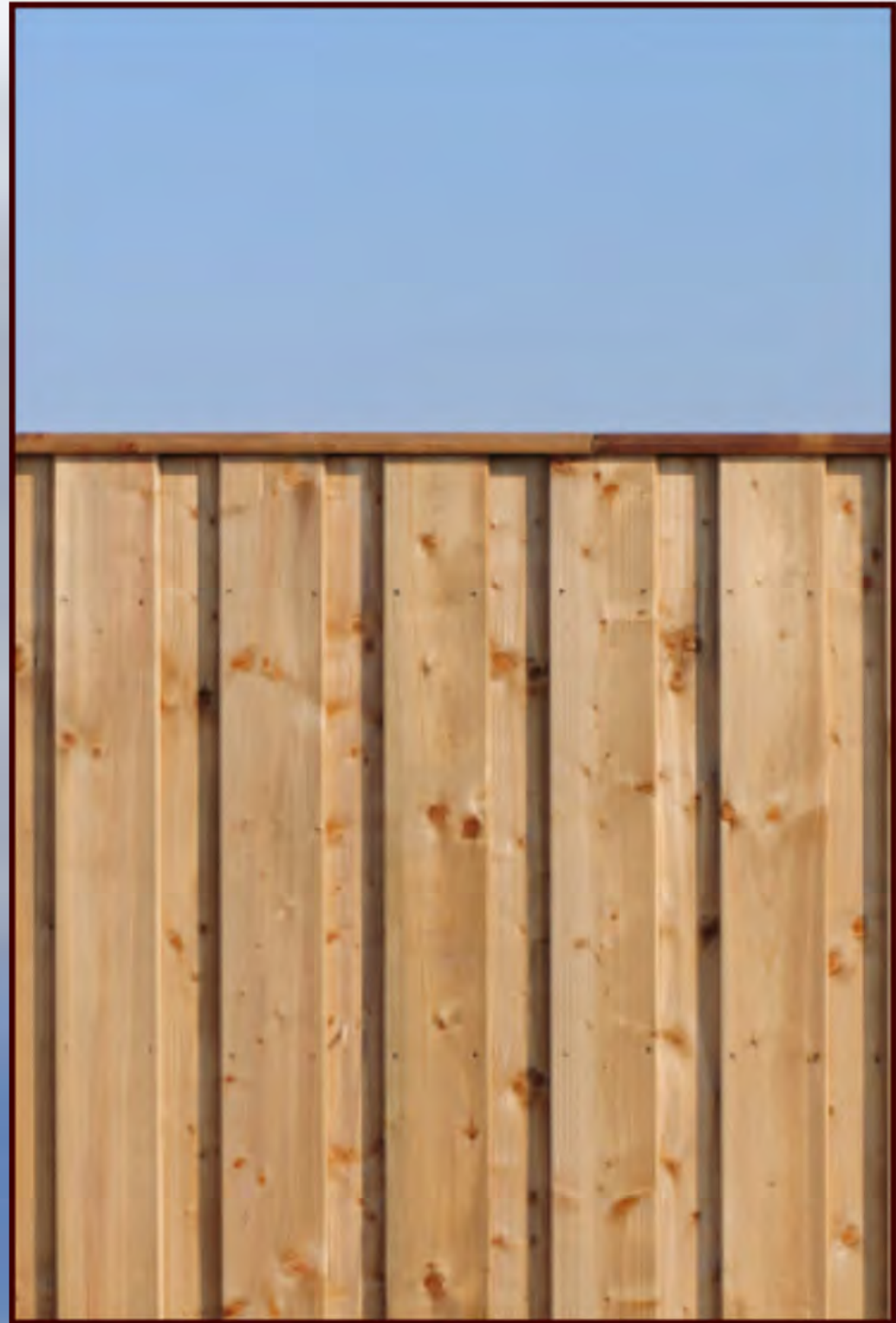
The new App Model addresses many of these issues directly



- Isolation
- Hygiene
- Standards
- Security

- Isolation
- Hygiene
- Standards
- Security

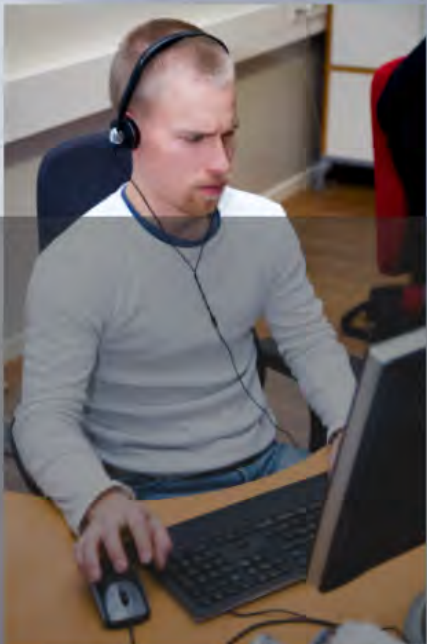
^{PP}“Good fences
make good
neighbors”^{PP}



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, 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, 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 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, 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

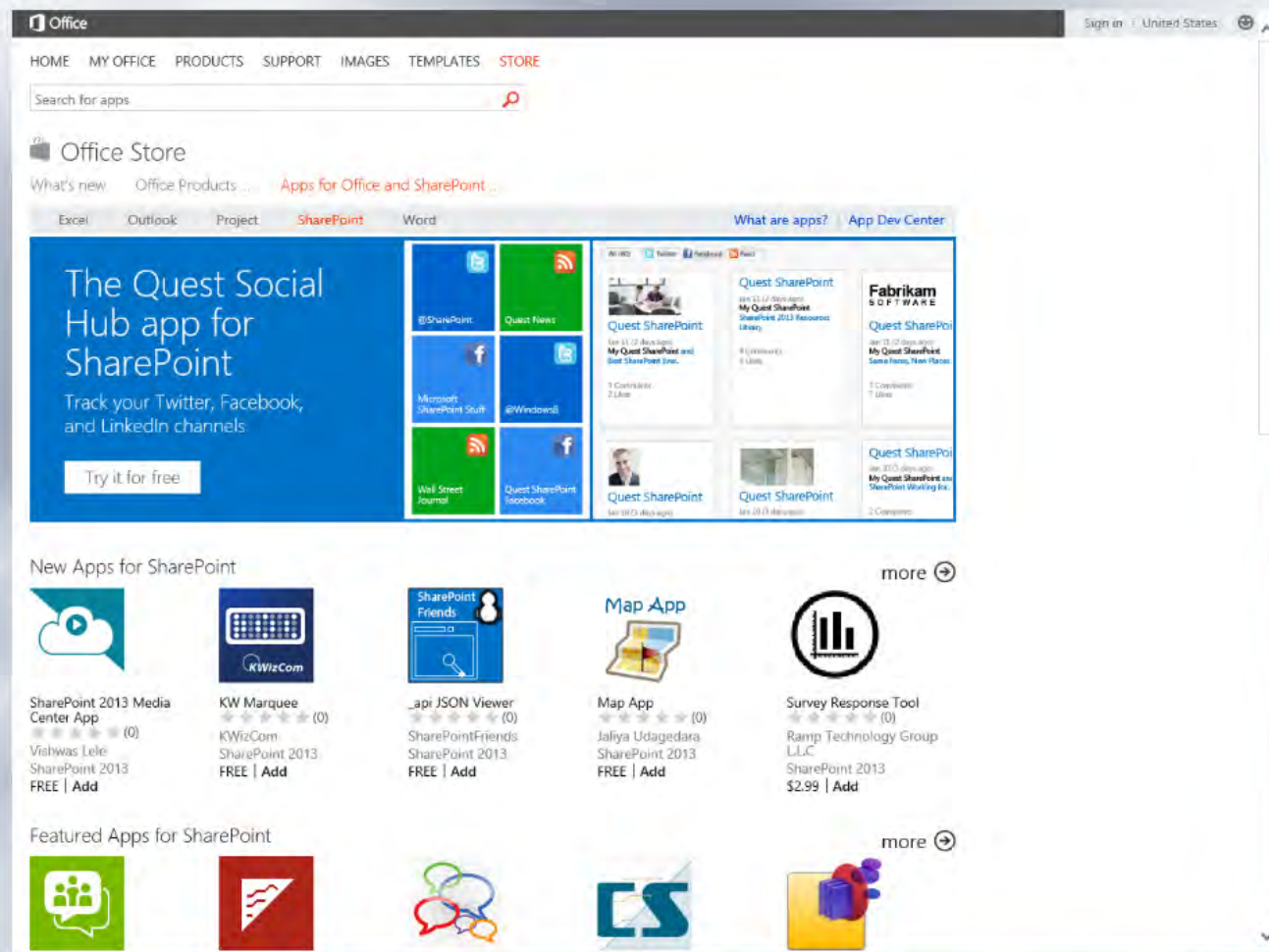


Oh yeah, one more
area that deserves
a mention ...

The App Store

The App Store


The screenshot displays the Office Store interface. At the top, there's a navigation bar with 'Office' on the left and 'Sign in' and 'United States' on the right. Below this is a menu with 'HOME', 'MY OFFICE', 'PRODUCTS', 'SUPPORT', 'IMAGES', 'TEMPLATES', and 'STORE'. A search bar is present with the text 'Search for apps'. The main content area is titled 'Office Store' and includes a sub-menu with 'What's new', 'Office Products', and 'Apps for Office and SharePoint...'. Below this is a secondary navigation bar with 'Excel', 'Outlook', 'Project', 'SharePoint', and 'Word', along with links for 'What are apps?' and 'App Dev Center'. The main content area features a large blue banner for 'The Quest Social Hub app for SharePoint' with a 'Try it for free' button. To the right of the banner is a grid of app tiles, including '@SharePoint', 'Quest News', 'Microsoft SharePoint Sum', '@Windows8', 'Wall Street Journal', and 'Quest SharePoint Facebook'. Below the banner is a section titled 'New Apps for SharePoint' with a 'more' link. This section contains five app cards: 'SharePoint 2013 Media Center App' (FREE | Add), 'KW Marquee' (FREE | Add), '_api JSON Viewer' (FREE | Add), 'Map App' (FREE | Add), and 'Survey Response Tool' (\$2.99 | Add). At the bottom is a 'Featured Apps for SharePoint' section with a 'more' link, showing five app icons: a green group icon, a red document icon, a colorful speech bubble icon, a blue 'CS' icon, and a yellow cube icon.



The screenshot displays the Office Store interface. At the top, there are navigation links for 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". A prominent banner for "The Quest Social Hub app for SharePoint" is featured, with a "Try it for free" button. Below the banner, there are sections for "New Apps for SharePoint" and "Featured Apps for SharePoint". The "New Apps for SharePoint" section lists several apps with their icons, names, and prices:

App Name	Developer	Price
SharePoint 2013 Media Center App	Vishwas Lele	FREE
KW Marquee	KWizCom	FREE
_api JSON Viewer	SharePointFriends	FREE
Map App	Jaliya Udagedara	FREE
Survey Response Tool	Ramp Technology Group LLC	\$2.99

The "Featured Apps for SharePoint" section shows a row of app icons, including a green icon with people, a red icon with a document, a colorful icon with speech bubbles, a blue icon with a document, and a yellow icon with a cube.

- 
- 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. Author. Evangelist.

Twitter: @spmcdonough

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

Blog: <http://SharePointInterface.com>

Book: <http://tinyurl.com/SPDRGuide2010>