



# Upgrading Your Skillset for SharePoint 2013 App Model Development



Sean P. McDonough  
(@spmcdonough)  
National Office 365 Solution Manager  
Cardinal Solutions Group, Inc.



# Before We Begin ...

- This is a 100-level session
- We're going to tour a number of different technologies and concepts in relatively quick fashion
- My goal is to provide you with starter resources you can review at your leisure





# About Me



## Living in the Cloud



- National Office 365 Solution Manager
- Office 365 Guidance, Planning, and Implementation
- Still maintain a significant focus on SharePoint

## Previously

- Chief SharePoint Evangelist at Idera
- Product Manager at Idera (Idera SharePoint backup)
- SharePoint Team Lead at Cardinal Solutions Group

# Motivation

## SharePoint 15 ISV TAP

- SDR on the new (Cloud) App Model in Summer 2012
- Working with the new App Model since Fall 2012
- Have been through the design and deployment process

## The Rules Have Changed

- App Model development isn't the full-trust solution development you may have done in the past
- You may strip a few gears getting up-to-speed ...

# Who I'm Targeting

## Grumpy Old Devs



- Well, not entirely accurate ...
- SharePoint devs who have focused on full-trust solutions and "core plumbing" solutions without much UI/UX
- Developers who don't know SharePoint and have heard that the new App Model might be good to learn



# This talk was originally titled

"Learning How To Paint (From One Plumber to Another)"

- I've generally considered myself a plumber
- Building Apps under the new model uses different skills and technologies



y

Devs

...  
sed on  
ing"

# Who I'm Targeting

## Grumpy Old Devs



- Well, not entirely accurate ...
- SharePoint devs who have focused on full-trust solutions and "core plumbing" solutions without much UI/UX
- Developers who don't know SharePoint and have heard that the new App Model might be good to learn

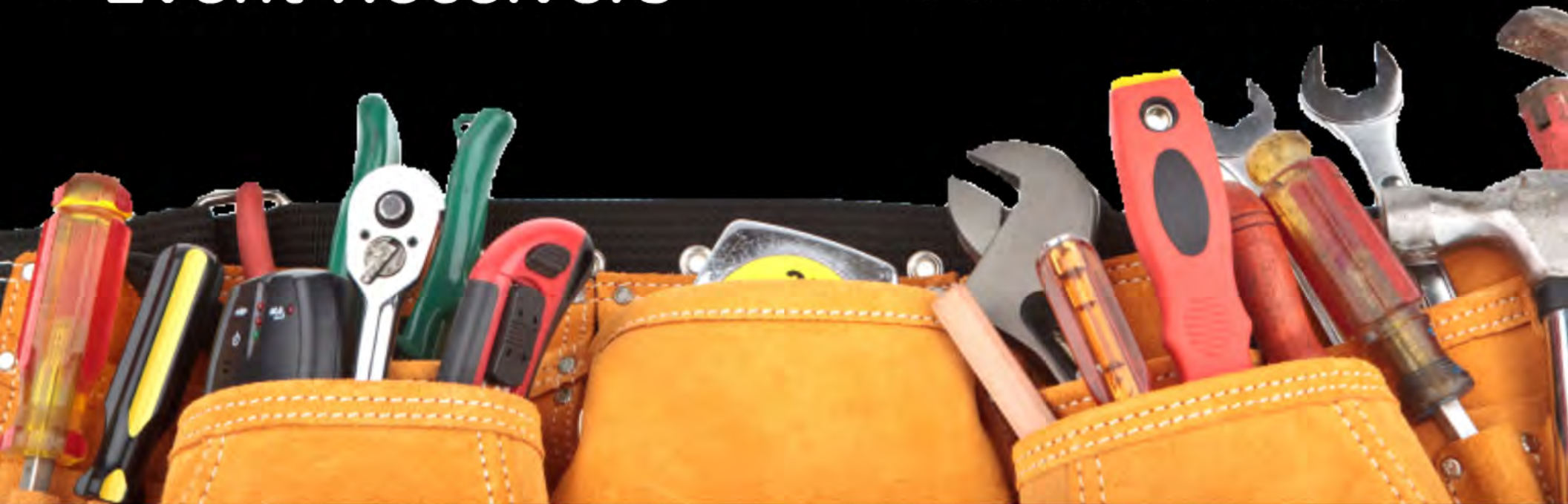
# The Tools You (May) Know

- Web Parts
- Timer Jobs
- Event Receivers
- Feature Receivers
- LAYOUTS Pages
- Custom Fields






- Web Parts
- Timer Jobs
- Event Receivers
- Feature Receivers
- LAYOUTS Pages
- Custom Fields



Bye Bye (mostly ...)





JavaScript  
Libraries

Responsive  
Design

HTML5 and  
CSS3

JavaScript  
and jQuery

CSOM and  
REST

... and hello to some new friends\*

# new friends\*

\* "friends" is probably the wrong term, but your relationship with these concepts and technologies is probably going to get better than it was before you started with Apps 😊



but first

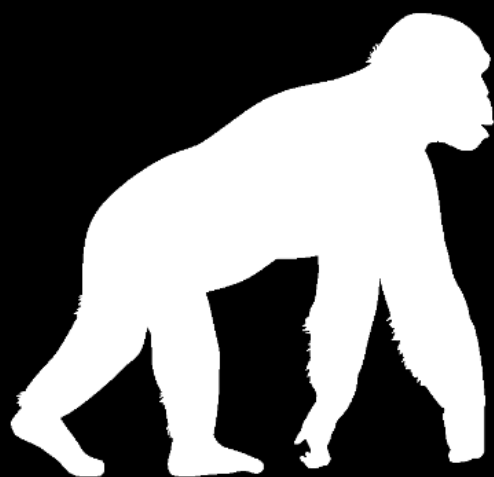
we should  
rewind a  
bit and ask



Why do we have a  
new App Model?

# The Evolution of Custom Application Development

*SharePoint Edition*



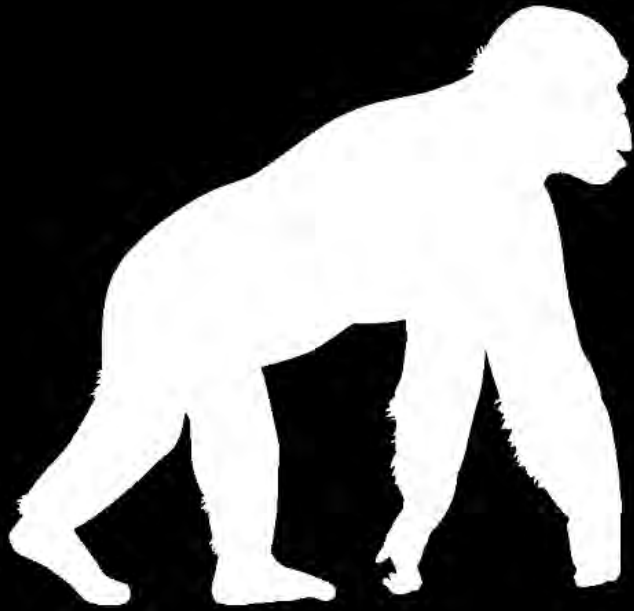
2003

2007

2010

2013





2003

- Server-side only
- No development support
- No deployment support
- SharePoint unaware of any changes being made



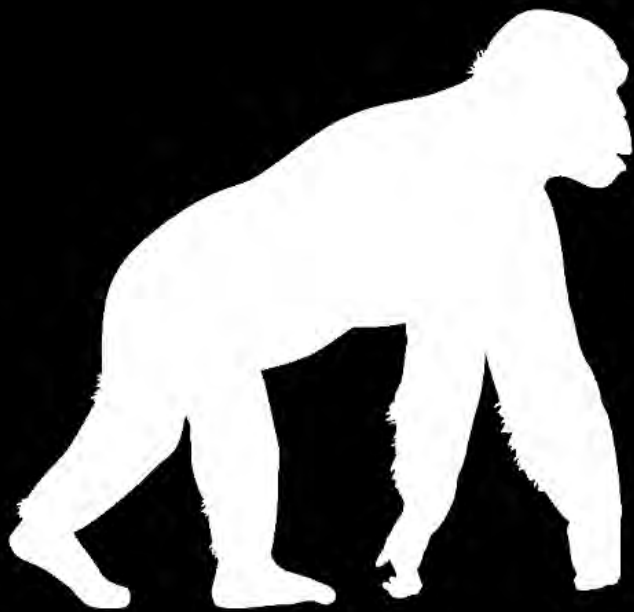
2007

- Server-side mostly
- Community dev support
- Solution packages
- SharePoint aware of WSP-deployed changes



2010

- Server-side mostly with some client-side
- VS-based dev support
- Client-Side Object Model (CSOM)
- Sandboxed solutions



# 2003

- Server-side only
- No development support
- No deployment support
- SharePoint unaware of any changes being made



# 2007

- Server-side mostly
- Community dev support
- Solution packages
- SharePoint aware of WSP-deployed changes



# 2010

- Server-side mostly with some client-side
- VS-based dev support
- Client-Side Object Model (CSOM)
- Sandboxed solutions

Upgrades, security, troubleshooting, performance ...



# 2013

# The Cloud

- The Cloud App Model (or simply "App Model") is new with SharePoint 2013
- Full Visual Studio development and deployment support
- Sidesteps limitations associated with sandboxed solutions
- All-but-guarantees a better operations scenario ("love your admin!")
- Only feasible model for Office 365 and multi-tenant cloud environments





# 2013

# The Cloud

- The Cloud App Model (or simply "App Model") is new with SharePoint 2013
- Full Visual Studio development and deployment support
- Sidesteps limitations associated with sandboxed solutions
- All-but-guarantees a better operations scenario ("love your admin!")
- Only feasible model for Office 365 and multi-tenant cloud environments

**! CODE NO LONGER RUNS ON SHAREPOINT SERVERS !**

wait ... did you just say  
that code doesn't run on  
SharePoint servers?



wait ... did you just say  
that code doesn't run on  
SharePoint servers?



**Correct.**

- Code runs on another (non-farm) server and/or in the user's browser based on the model/variant chosen
- All interactions with "proper" SharePoint data occur through CSOM and/or REST
- No more server-side object model access
- No more free passes on security



# Correct.

- Code runs on another (non-farm) server and/or in the user's browser based on the model/variant chosen
- All interactions with "proper" SharePoint data occur through CSOM and/or REST
- No more server-side object model access
- No more free passes on security

It's okay if you need to rage about this for a bit. Go ahead. I'll wait.



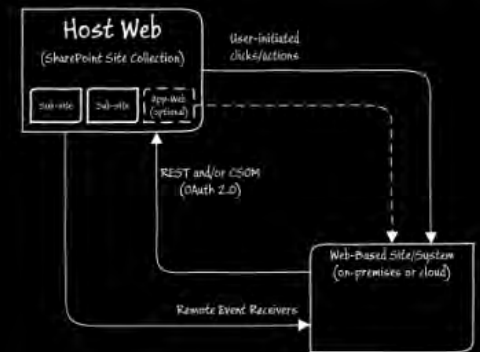
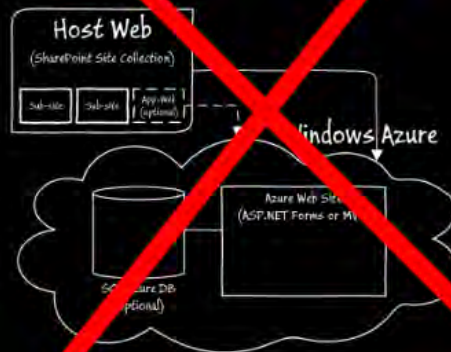
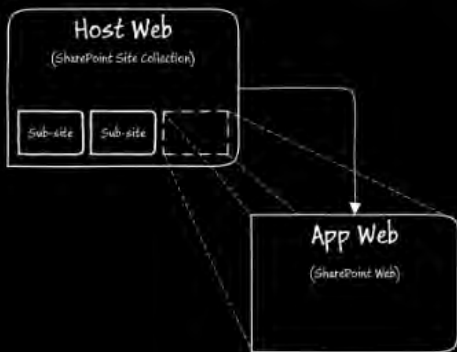
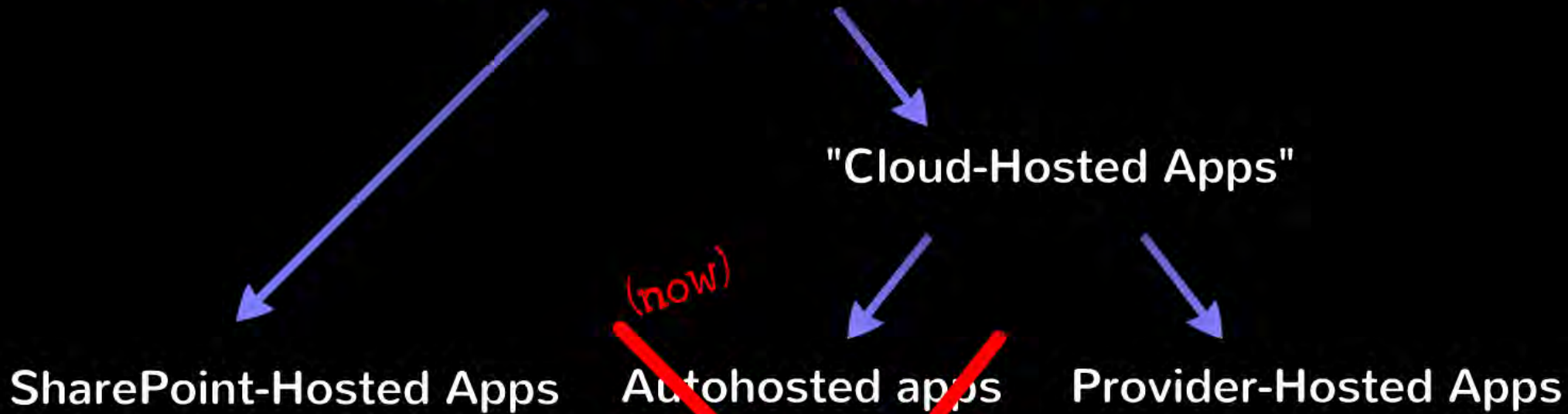
It's okay if you need to rage about this for a bit. Go ahead. I'll wait.



# App Model is Here to Stay

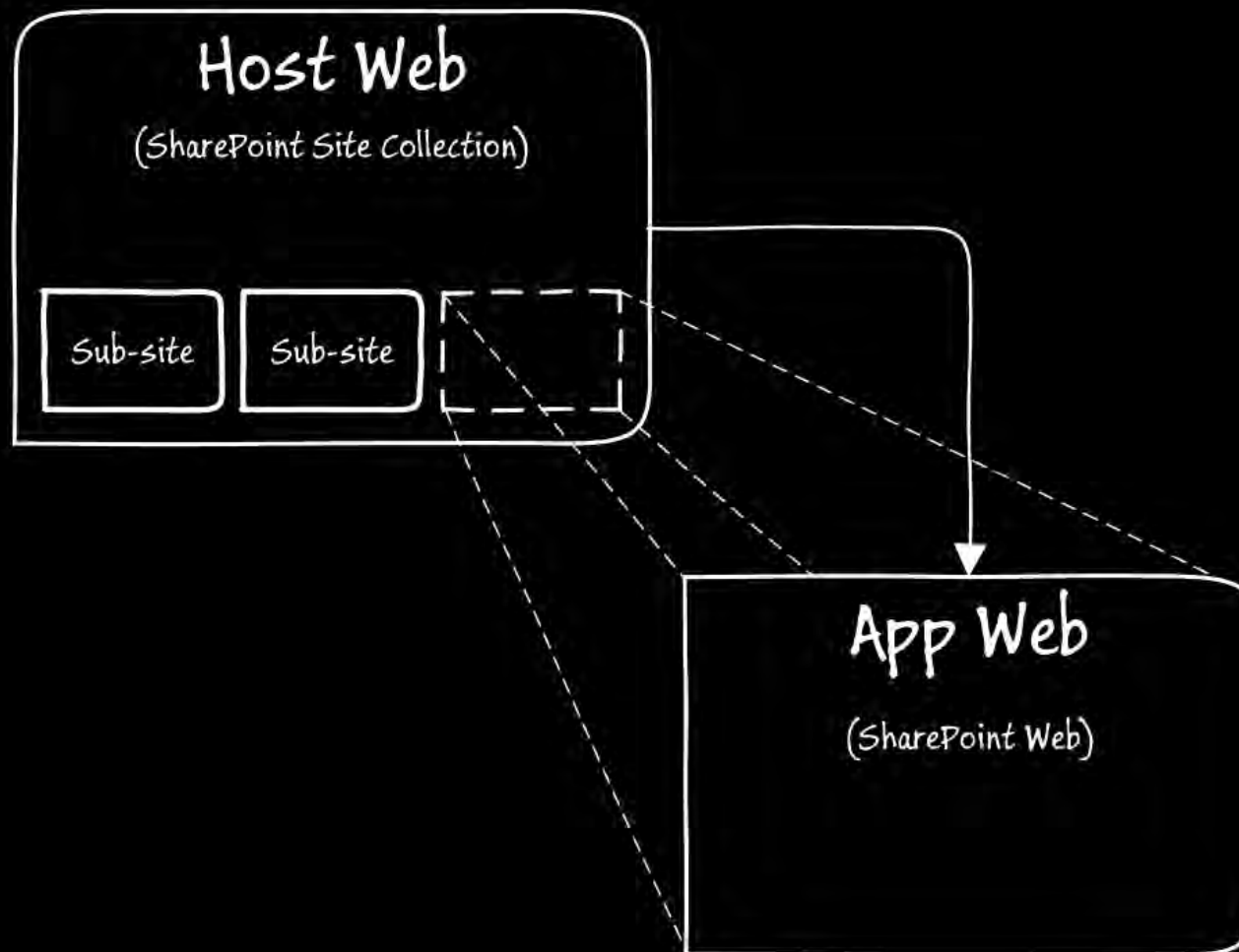
- Improves SharePoint performance
- Generally speaking, it simplifies upgrades (to farm and custom code)
- Simplifies troubleshooting
- Is open to developers on other stacks (e.g., J2EE)
- Needed for The Cloud

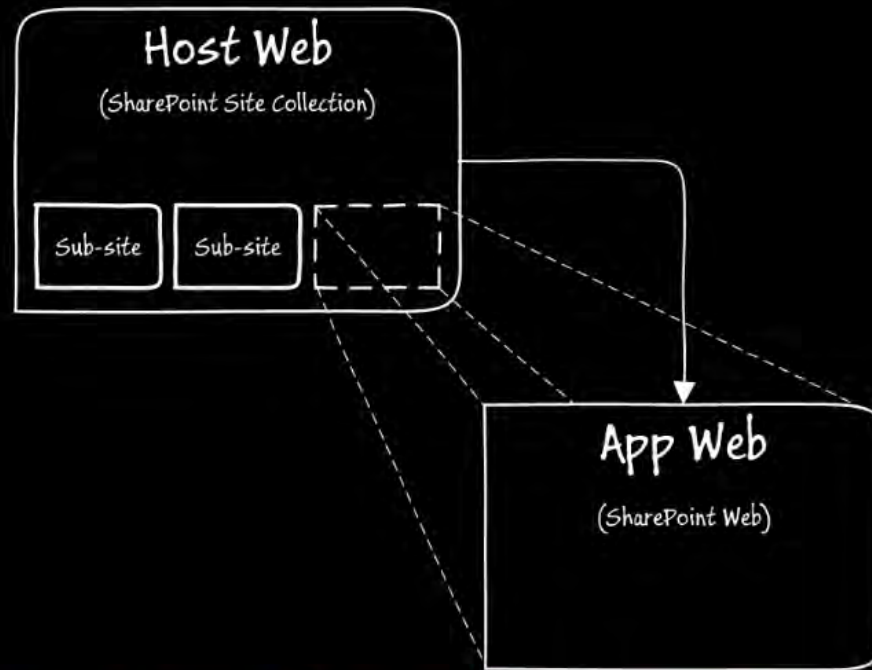
# The App Model





# SharePoint-Hosted Apps





# SharePoint-Hosted Apps

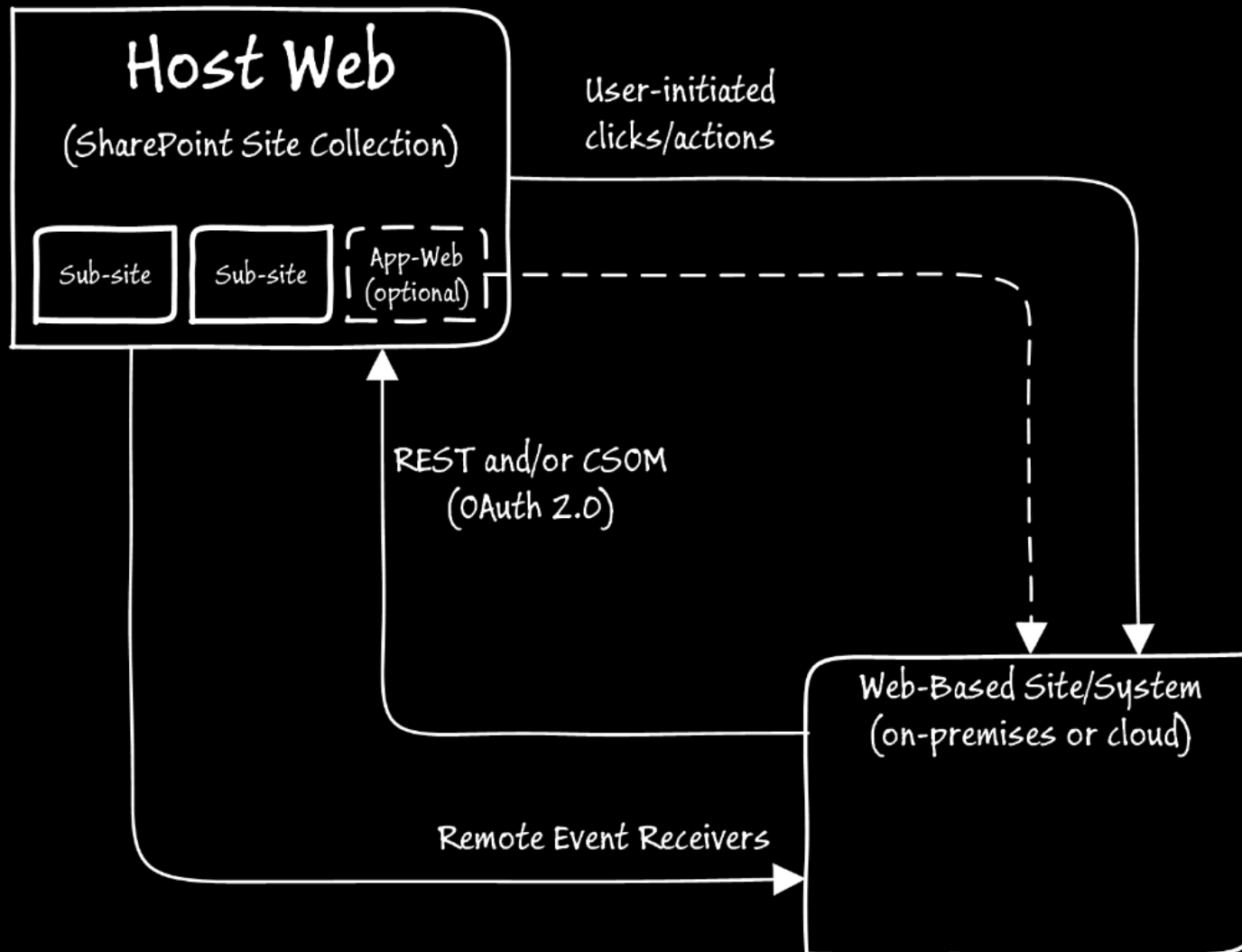
## Pros

- Easier to develop
- Fewer security concerns
- Easier to deploy
- Less complex

## Cons

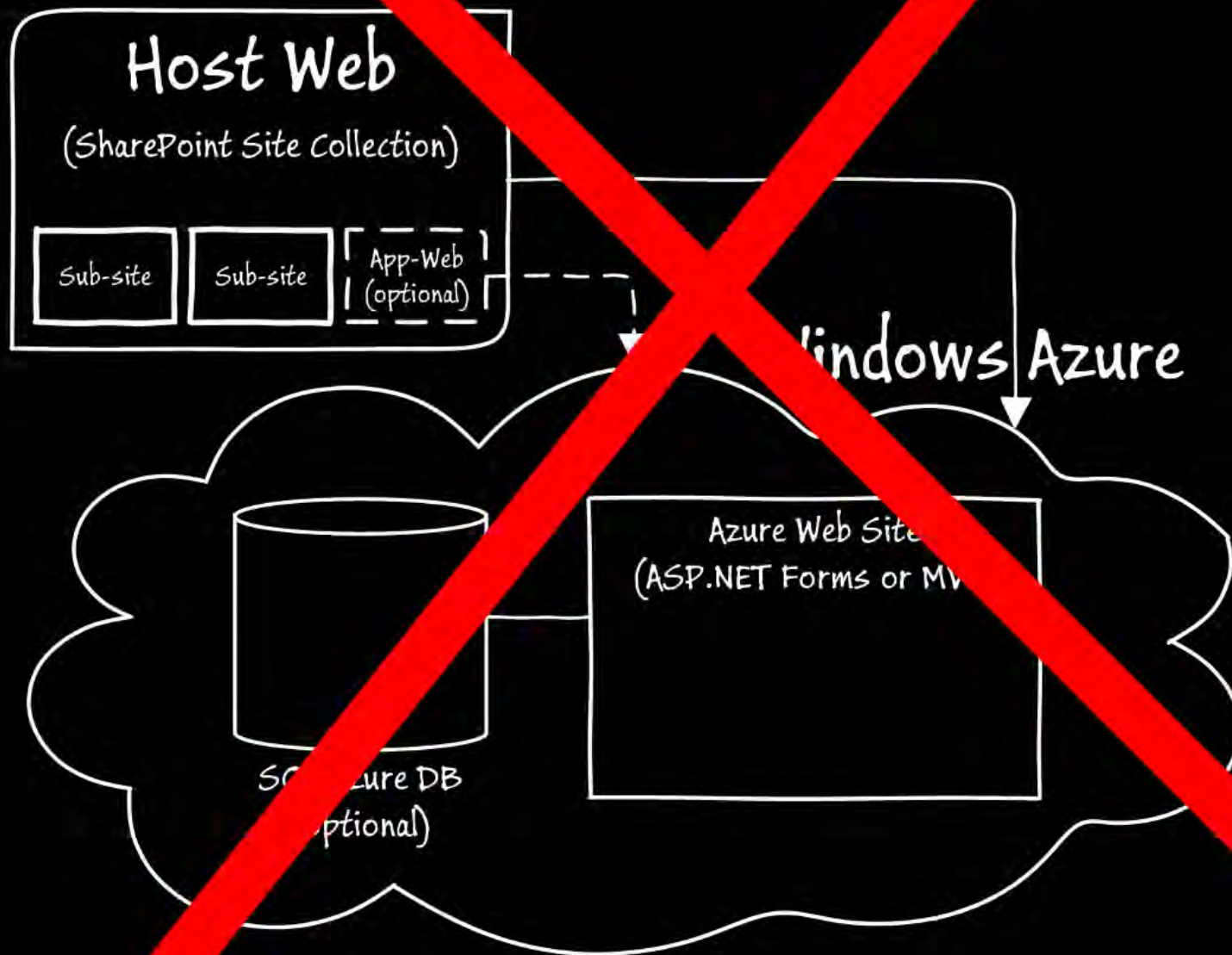
- No compiled code
- No unattended execution
- Generally less capable
- Scoped to just a sub-site

# Provider-Hosted Apps

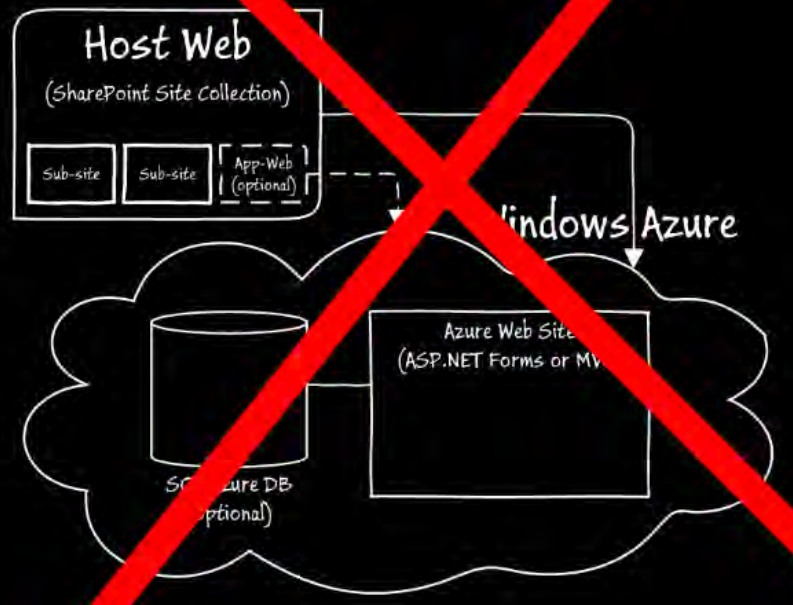




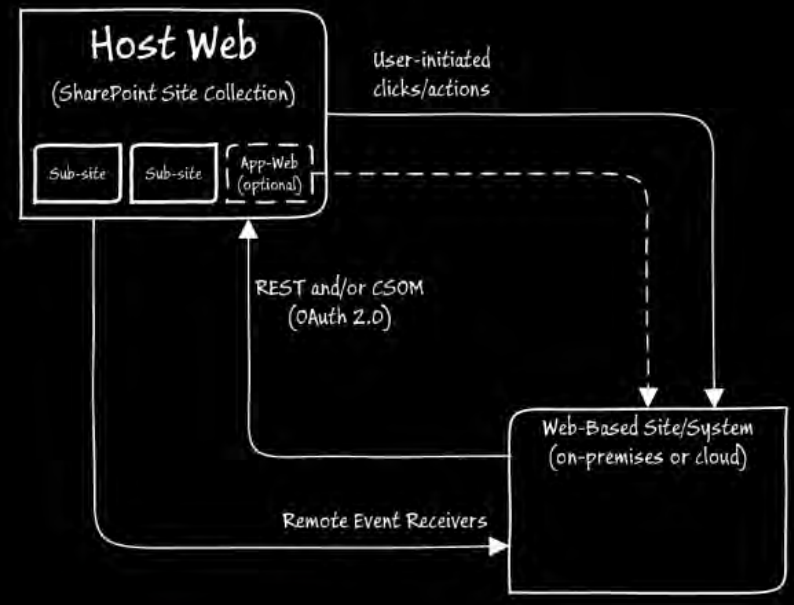
# Autohosted apps



# Autohosted apps



# Provider-Hosted Apps



# Cloud-Hosted Apps

## Pros

- Much more capable
- Apps can act on their own
- Open to non-SharePoint devs
- You control everything

## Cons

- (Potentially) complex deployment
- Tougher security configuration
- Requires external hosting
- Still not a full-trust solution

# Cloud-Hosted Apps

## Pros

- Much more capable
- Apps can act on their own
- Open to non-SharePoint devs
- You control everything

## Cons

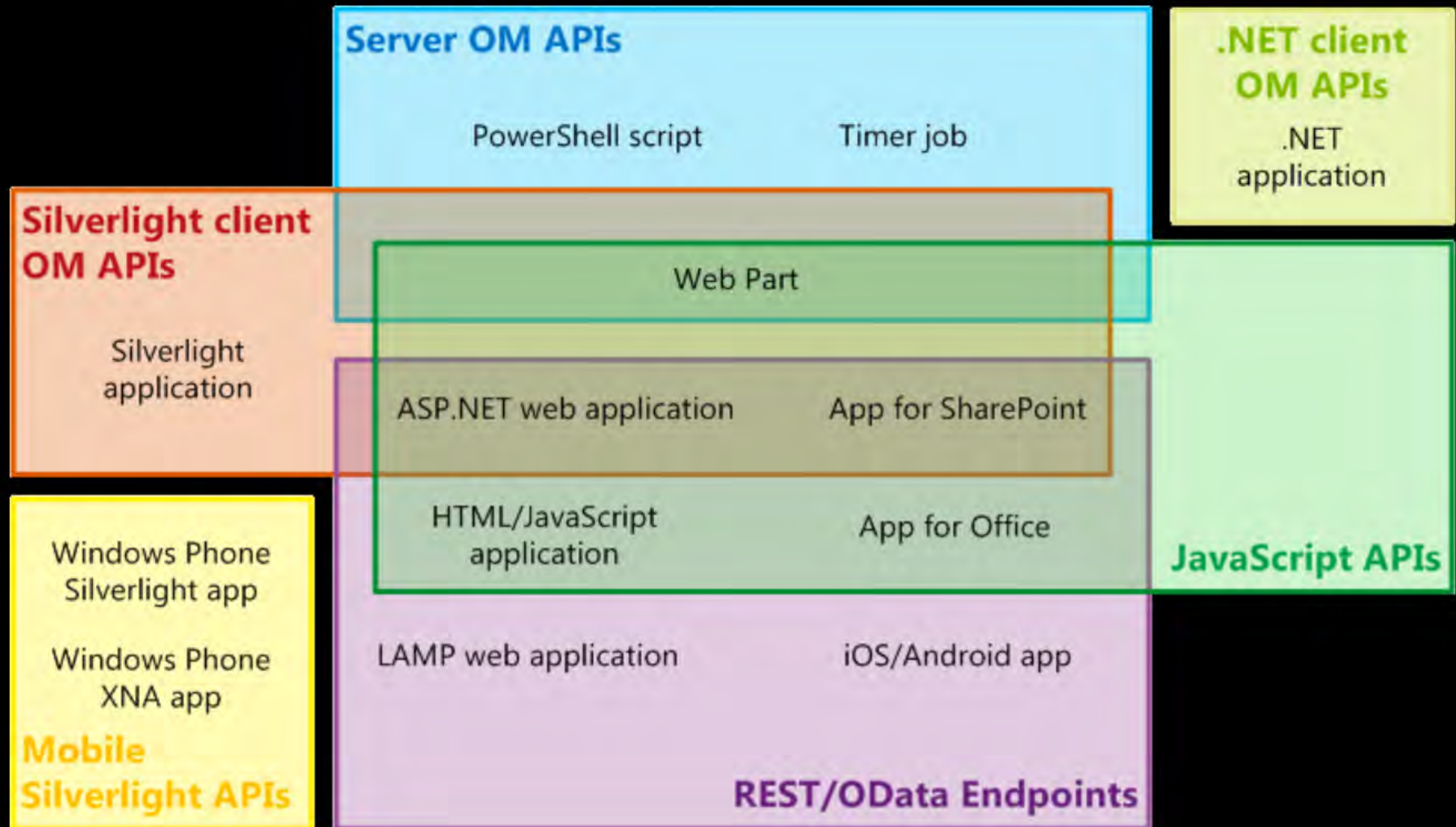
- (Potentially) complex
- Tougher security
- Requires external services
- Still not a full-fledged app

## Microsoft killed autohosted apps in May 2014

- Richard diZerega has a great summary on this topic
- My take: too many questions about who would actually be paying for the auto-provisioned Azure services!



# Choose the right API and strategy for the task



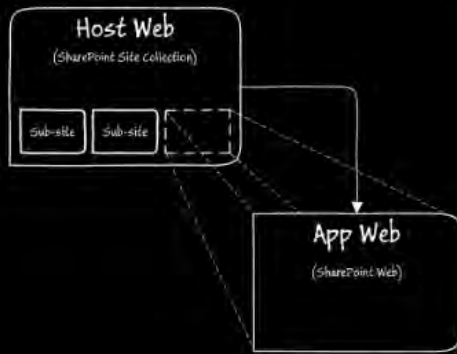
# The App Model

"Cloud-Hosted Apps"

SharePoint-Hosted Apps

~~Autohosted apps~~

Provider-Hosted Apps



SharePoint-Hosted Apps

Pros

- Easier to develop
- Fewer security concerns
- Easier to deploy
- Less complex

Cons

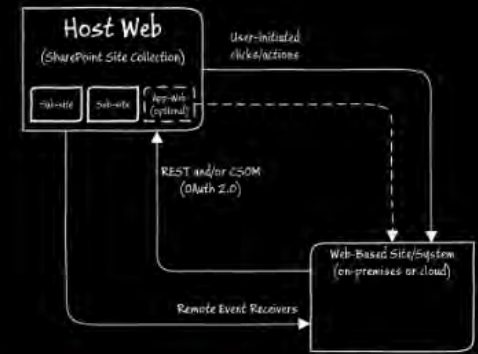
- No compiled code
- No unattended execution
- Generally less capable
- Scoped to just a sub-site



Cloud-Hosted Apps

Pros

- Much more capable
- Apps can act on their own
- Open to non-SharePoint devs
- You control everything



Cons

- (Potentially) complex deployment
- Tougher security configuration
- Requires external hosting
- Still not a full-trust solution

# Common App Model Themes

- Embrace web standards
- Build rich experiences
- Get off the server!



# Common App Model Themes

- Embrace web standards
- Build rich experiences
- Get off the server!

*This was scary for me at first!*

I'll share some of the resources that helped me make the transition





JavaScript  
Libraries

Responsive  
Design

HTML5 and  
CSS3

JavaScript  
and jQuery

CSOM and  
REST

HTML5

and CSS3



Essential pieces in any App user interface

HTML5  
and CSS3



Essential pieces in any App user interface

# HTML5 and CSS3



HTML: HyperText Markup Language

- The content you want to share

CSS: Cascading Style Sheets

- How the content is presented

Yes, very  
basic

Mark Brown & Jerry Honeycutt



Using  
Special Edition

# Using HTML 4

Fourth Edition

Comprehensive. Authoritative. Recommended.

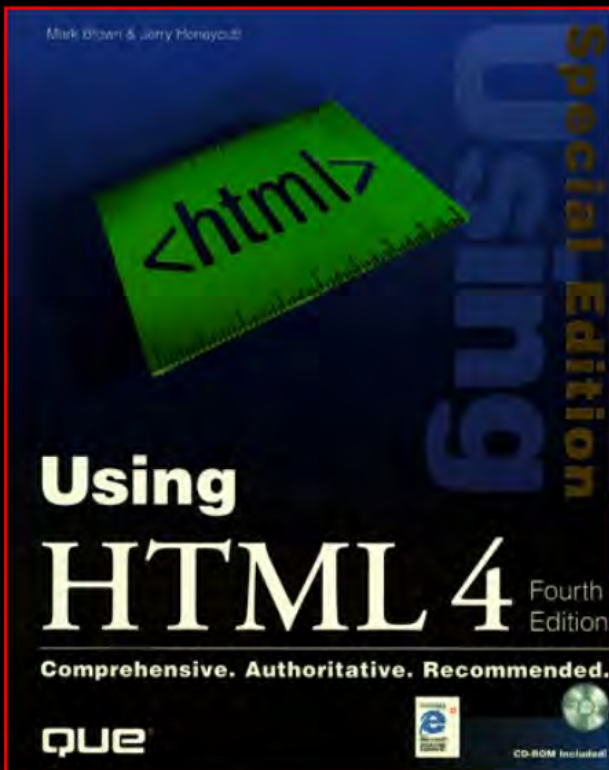
que



CD-ROM Included!

This is  
my old  
HTML  
reference





This is  
my old  
HTML  
reference

Published  
in 1997!





This is  
my old  
HTML  
reference

Published  
in 1997!



- Been under a rock?
- If you're used to ASP.NET web forms, you probably have some bad behaviors





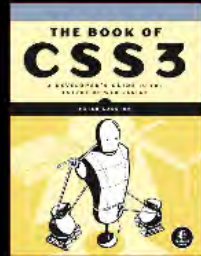
# Thoughts for Grumpy Old Devs





# Thoughts for Grumpy Old Devs

- Get a good HTML and CSS reference  
*at least understand the latest spec updates*
- Stop using tables and tools that abuse them  
*Tables are no longer okay for arbitrary formatting*
- Get out of the habit of using inline styles everywhere  
*It's a bad habit that tends to come with too much ASP.NET dev*
- It's okay to use CSS3. Really  
*The spec isn't as volatile as some might have you believe*







JavaScript  
Libraries

Responsive  
Design

HTML5 and  
CSS3

JavaScript  
and jQuery

CSOM and  
REST

# Responsive Design





# Responsive Design

- Fluid design and layout typically through a grid system (or framework) and CSS3 media queries
- Flexible images/media resize to container as needed

Why?

# Why?

Two big reasons



# The one you've probably heard

- Traffic to your Apps is going to come from a variety of devices and form factors



# Why?

Two big reasons

# Why?

Two big reasons

The one you probably haven't heard (yet)

- Apps provide HTML for a variety of containers

IMMERSIVE  
FULL PAGE



A diagram showing a white rectangular border containing a smaller yellow rectangular border, representing an immersive full page container.

PART



A diagram showing a white rectangular border containing a smaller white rectangular border, which in turn contains two smaller white rectangular borders (one square, one vertical rectangle) and a yellow rectangular border, representing a part container.

UI CUSTOM  
ACTIONS



A diagram showing a white rectangular border containing a complex UI layout with various elements, including a yellow rectangular border, representing a UI custom actions container.



# Thoughts for Grumpy Old Devs





# Thoughts for Grumpy Old Devs

- Get (or build) a CSS-based grid or framework  
*Tons of free ones available*
- Browser sniffing: not so cool these days  
*Still in the arsenal, but a rigid approach*
- Learn the basics of media queries  
*Not hard to write and leverage*



JavaScript  
Libraries

Responsive  
Design

HTML5 and  
CSS3

JavaScript  
and jQuery

CSOM and  
REST



# JavaScript and jQuery




# The Nuts and Bolts of Many Solutions

## JavaScript and jQuery



- JavaScript = "function" *loosely speaking*
- jQuery = "events & behavior"

What I actually mean  
when I say this 

Nuts and Bolts

# Provider-hosted Apps

- helpful for rich UI/UX
- essential for smooth client-side processing & behavior (e.g., single-page apps, or SPAs)

# SharePoint-hosted Apps

- Your sole option for business logic
- Yes, this means your App is written entirely in JavaScript!

What I actually mean



**INSTANT**

# JavaScript



A comprehensive tutorial in JavaScript for Web developers with HTML knowledge

JavaScript applications demonstrated in detail

Covers Dynamic HTML and its relationship to JavaScript

- ▲ Extensive coverage of both Netscape's JavaScript and Microsoft's Jscript
- ▲ Includes details of the ECMAScript standard
- ▲ Full reference for core language features and compatibility issues



Nigel McFarlane

This is my old  
JavaScript  
reference from

1997

**INSTANT**

# JavaScript



A comprehensive tutorial in JavaScript for Web developers with HTML knowledge

JavaScript applications demonstrated in detail

Covers Dynamic HTML and its relationship to JavaScript

- ▲ Extensive coverage of both Netscape's JavaScript and Microsoft's Jscript
- ▲ Includes details of the ECMA Script standard
- ▲ Full reference for core language features and compatibility issues



Nigel McFarlane

This is my old  
JavaScript  
reference from

1997

JavaScript is not a "toy" language

# JavaScript is not a "toy" language

## Lots of Goodness ...

- object oriented
- function & lambda-based programming
- prototypal inheritance
- object literal notation and JSON

## ... but some ??? stuff too

- global object / common namespace
- loose typing and easy (slippery?) coercion
- function-level (not block-level) scoping
- no compilation (and thus no compile-time help)





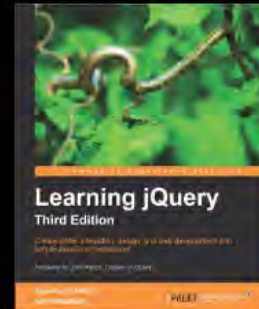
# Thoughts for Grumpy Old Devs



# Thoughts for Grumpy Old Devs

- just like HTML/CSS, get some good (newer) references

*"JavaScript: The Good Parts" is fantastic*



- if you want or need a more "classical" approach, consider learning about TypeScript

*Try giving JavaScript a "fair shake" first, though*

- there are tools that can help you along the way

*VS web Essentials, ReSharper, JSLint, and more*





JavaScript  
Libraries

Responsive  
Design

HTML5 and  
CSS3

JavaScript  
and jQuery

CSOM and  
REST



# JavaScript Libraries

# JavaScript Libraries



Chances are that someone else tried  
to climb the mountain before you

with a little luck, they packaged the  
result as a JavaScript library



# Libraries I Found Helpful when Starting Out ...

- jQuery (of course!)
- jQuery UI
- Modernizr
- JsRender
- Google's Cookie Library



# Libraries that are Changing the Landscape Now:

- Data.js
- Breeze.js
- Knockout
- AngularJS

It's oftentimes worth trying to find a library before you "roll your own" ...



JavaScript  
Libraries

Responsive  
Design

HTML5 and  
CSS3

JavaScript  
and jQuery

CSOM and  
REST

CSOM and  
REST





**CSOM** : Client-Side Object Model

**REST** : Representational State Transfer

This is how your  
App will interact  
with SharePoint

CSOM  
REST



- `_api` web services all-the-way
- even for SharePoint-hosted Apps
- no more "free" `SPContext`
- no more `RunWithElevatedPrivileges()`
- "so long" to synchronous operations



# Thoughts for Grumpy Old Devs





# Thoughts for Grumpy Old Devs

- Prepare to spend some time learning AJAX and JSON  
*ASynchronous operations and calling patterns are quite different than what you're probably used to*
- CSOM, REST, or both? Learn when to use each  
*Choice generally depends on App Model variant used ...*
- Can't get there from here?  
*Learn to use the web proxy and cross-domain library*



JavaScript  
Libraries

Responsive  
Design

HTML5 and  
CSS3

JavaScript  
and jQuery

CSOM and  
REST



Do you feel a bit like this?

App Model

You





# It's okay

It doesn't hurt as bad  
as you might think



- Patch-up Visual Studio
- Get an Office 365 Tenant
- Activate your Azure dev benefits
- Play around and see what you can create with the new model!

# Some resources ...

This document contains the references and resources to accompany my "Upgrading Your Skillset for SharePoint 2013 App Model Development" session. This is by no means an exhaustive list, but it does contain resources that I found extremely helpful as I sought to come up-to-speed with development and associated concerns pertaining to SharePoint 2013's new App Model.



If you have comments, questions, or other feedback, then I encourage you to visit my blog (<http://SharePointInterface.com>) or contact me via email ([sean@sharepointinterface.com](mailto:sean@sharepointinterface.com)) or Twitter ([@spmcdonough](https://twitter.com/spmcdonough)). I'd love to hear from you!

Last updated on October 23, 2014

- Sean McDonough

## App Model Ramp-Up

- **Apps for SharePoint Overview.** This MSDN page provides an end-to-end walk-through on the new SharePoint 2013 App Model, its characteristics, what it seeks to address, and more. <http://tinyurl.com/pfp-appsoverview>
- **Web Essentials for Visual Studio.** This Visual Studio extension (available from within Visual Studio itself) provides a number of accelerators and helpers to improve your experience when working with JavaScript, CSS, and other App Model technologies. <http://vswebessentials.com/>
- **App Approaches to Common SharePoint Customizations.** Microsoft's Richard [dizerega](#) put this together to compare how many common full-trust customizations can be achieved (or not) under the new App Model. This is a great article to use as a starting point for researching whether or not something can or should be done with the App Model. <http://tinyurl.com/pfp-approaches>
- **Autohosted Apps are Dead ... Big Deal!** This is another Richard [dizerega](#) post where he discusses Microsoft's decision to decommission its [autohosted](#) apps option, leaving us with SharePoint-hosted and provider-hosted app options. It's a good read. <http://tinyurl.com/pfp-autohosted>

## HTML and CSS

- **The Definitive Guide to HTML5.** This book by Adam Freeman is an end-to-end reference on HTML5. It is nearly 1000 pages in length and is a good, detailed reference. <http://tinyurl.com/pfp-html5guide>
- **Mozilla Developer Network.** The Mozilla Developer Network (MDN) is an excellent online resource for HTML, CSS, and various other browser-based technology topics. Although w3schools.com hits tend to appear first in search engine results, I've found the content on the MDN to be more accurate and "reputable" overall.

- Items I found helpful while getting up-to-speed
- available on my blog under "Resources"

# My blog



<http://SharePointInterface.com>

My  
Employer:



Twitter: @spmcdonough

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