Build It and They Will Come:

Custom Solutions for SharePoint 2013

Sean P. McDonough
Lead Bitsmith & Owner
Bitstream Foundry

- **Started building SharePoint solutio**

Sean P. McDonough
Lead Bitsmith & Owner
Bitstream Foundry

bitstream FOUNDRY

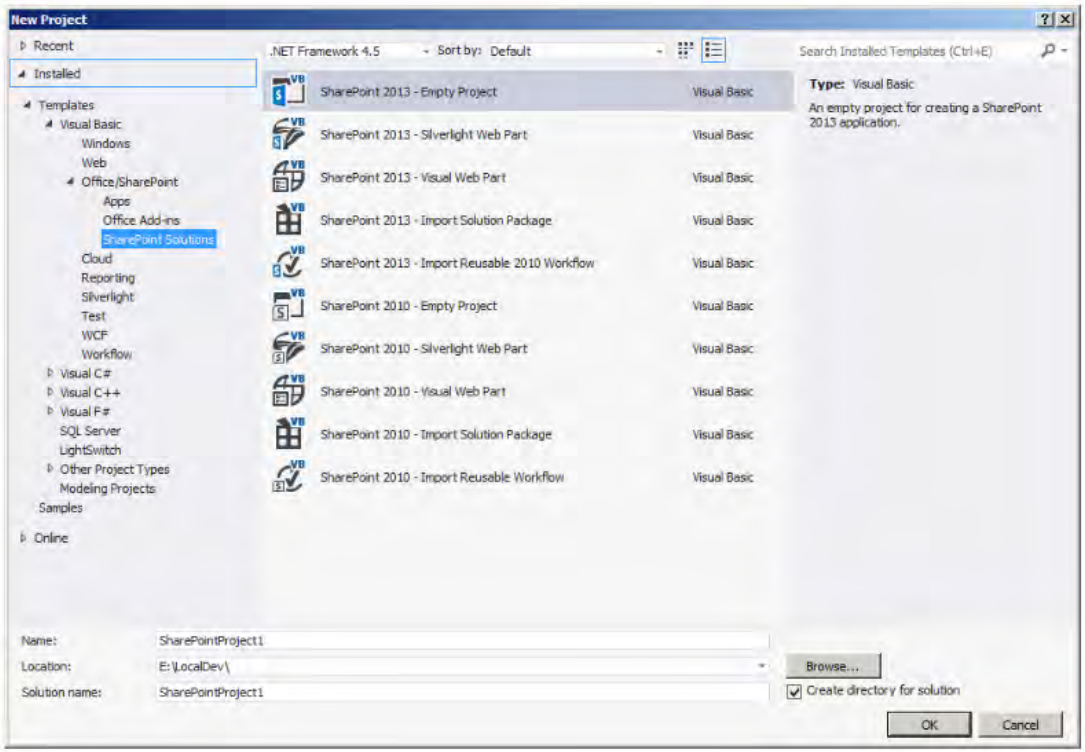
My Company

- Started building SharePoint solutions on Portal Server 2003
- Spent a chunk of time with SharePoint 2007 in Internet site development and administration for a Fortune 25 company
- Managed a product and "evangelized" for an ISV recently
- Nowadays I'm back to hands-on dev and admin
- On Twitter? Heckle me! @Spmcdonough

The Agenda

- What Stays the Same
- New 2013 Approaches
- The Implications
- Q&A Throughout!

what stays the same?



pretty much everything

well ... mostly

- ✓ WYSIWYG page editing
- ✓ Content Editor and Content Query Web Parts
- ✓ Custom lists and views

Browser-based customization and design options are doing fine

- ✓ Web Parts

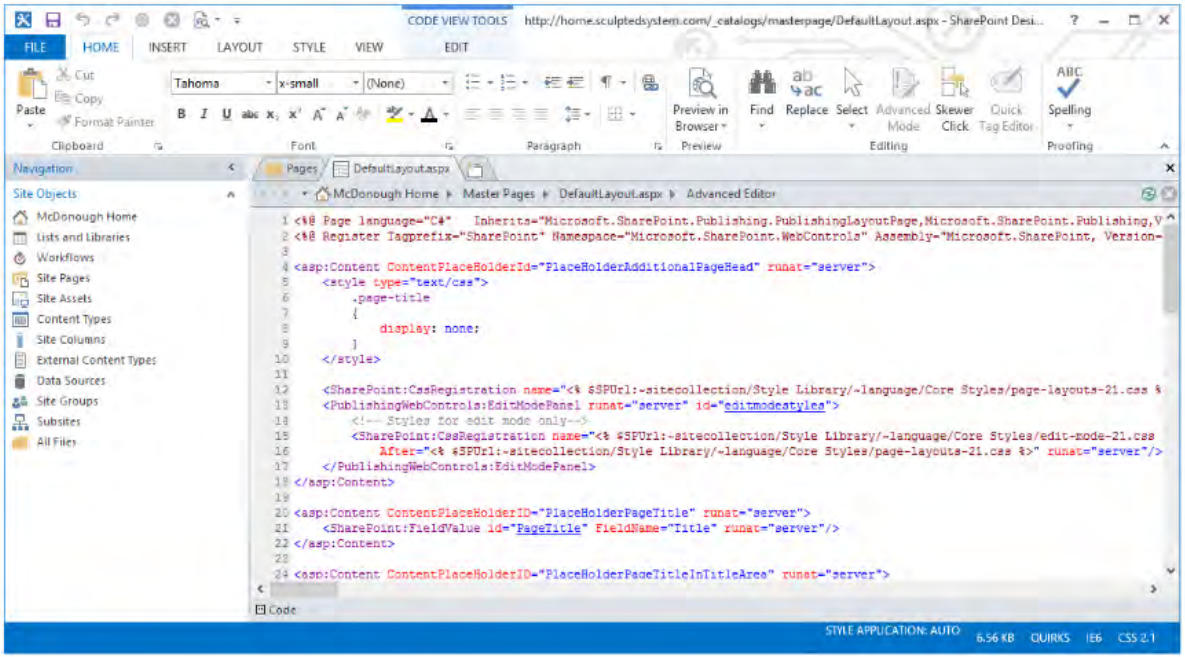
design options are doing fine

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

Full-trust solutions are alive and well

Sandboxed solutions are still around, too.

... but they have been deprecated*



SharePoint Designer became "SharePointer"

SharePoint~~Designer~~

SharePoint Designer became "SharePointer"

~~Designer~~

Customizations from previous versions may need to be "reworked" a bit

... but all in all, most of the dev options you've grown to love are still around

So .... what's changed?

Now it's all about The Cloud

- Multitenancy is a key compo

Now it's all about The Cloud

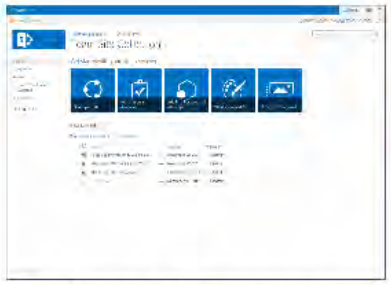
Even if you're going to keep your feet firmly planted in your own data center, The Cloud is impacting you

- **Multitenancy is a key component in the SharePoint strategy**
- **SharePoint is transforming from "platform" to "glue"**
- **Development is being pushed from server to client-side**

Development is being pushed from server to client-side

SharePoint Farm

Application Servers



Client Browser

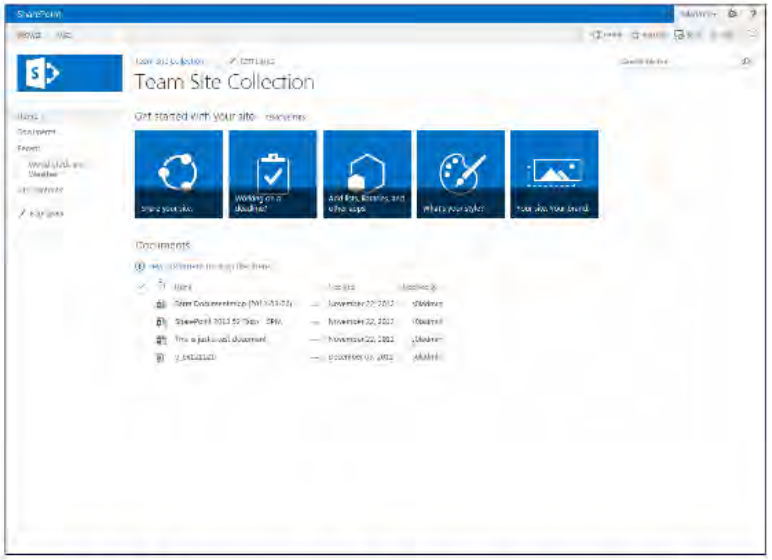
- With SharePoint 2013, the focus is here!
- Browser-based focus carries with it a strong push for standards-based implementation

The "Old" Approach to Development

- Code was written against the server object model
- Code was packaged through Features and WSPs (solution packages)
- Administrators were needed for deployment, maintenance, etc.

This doesn't work in a Cloud or multitenant environment!

SharePoint Farm



Client Browser

- With SharePoint 2013, the focus is here!
- Browser-based focus carries with it a strong push for standards-based implementation

client-side can leverage app servers

Application Servers

- Even though custom code isn't running on SharePoint farm members, it can be running somewhere else!
- Clients can call app servers directly
- App servers can call SharePoint (via web services) directly and vice-versa

Some new techniques for customization and development

- Design Manager
- Content Search Web Part (CSWP)
- The New App Model

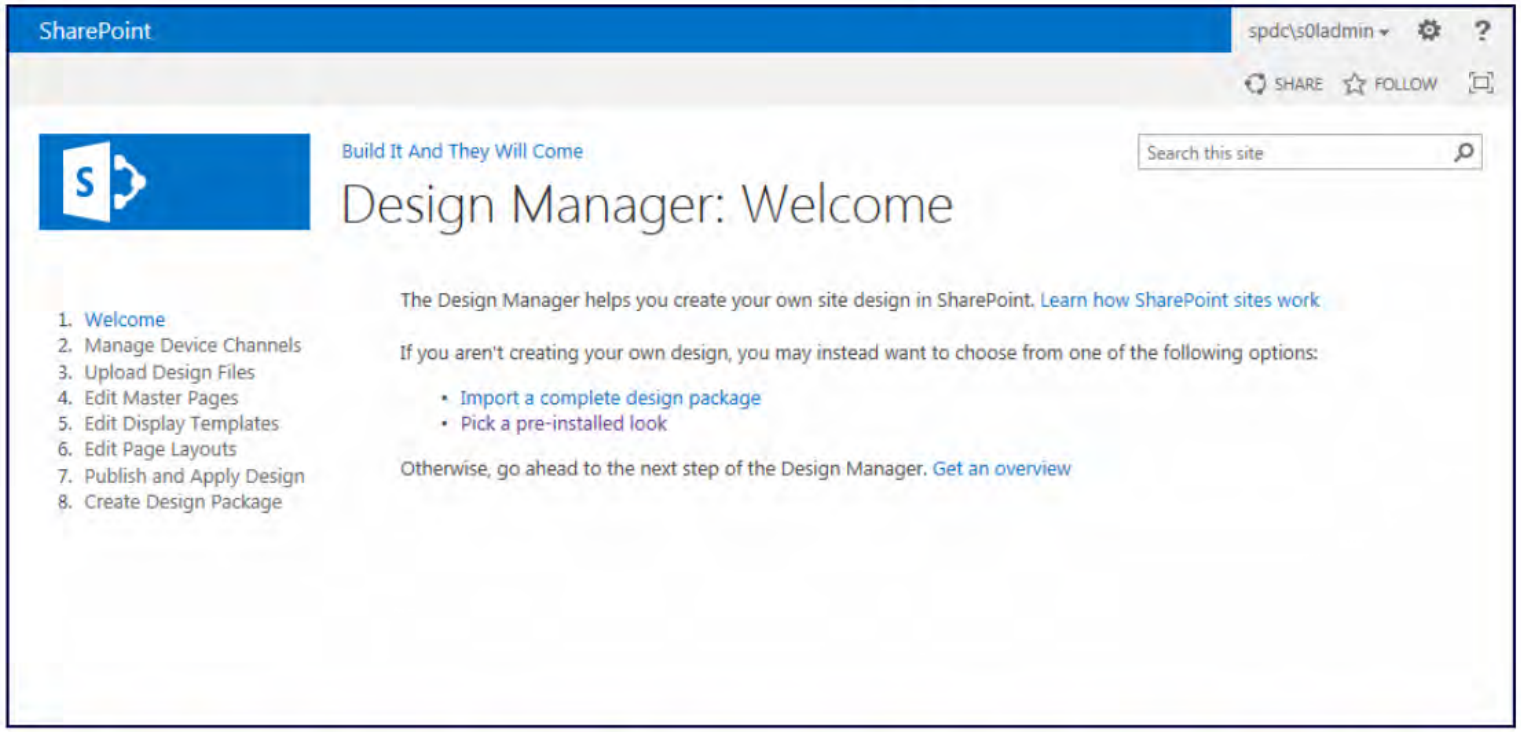
velopment

Design Manager

Content Search Web Part (CSWP)

The New App Model

What is the Design Manager?

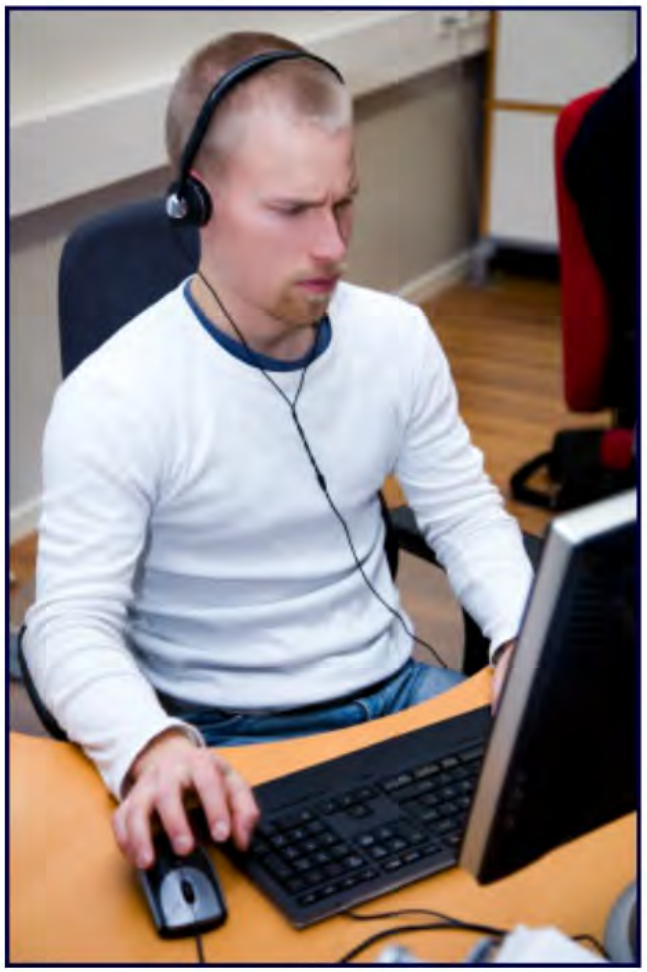


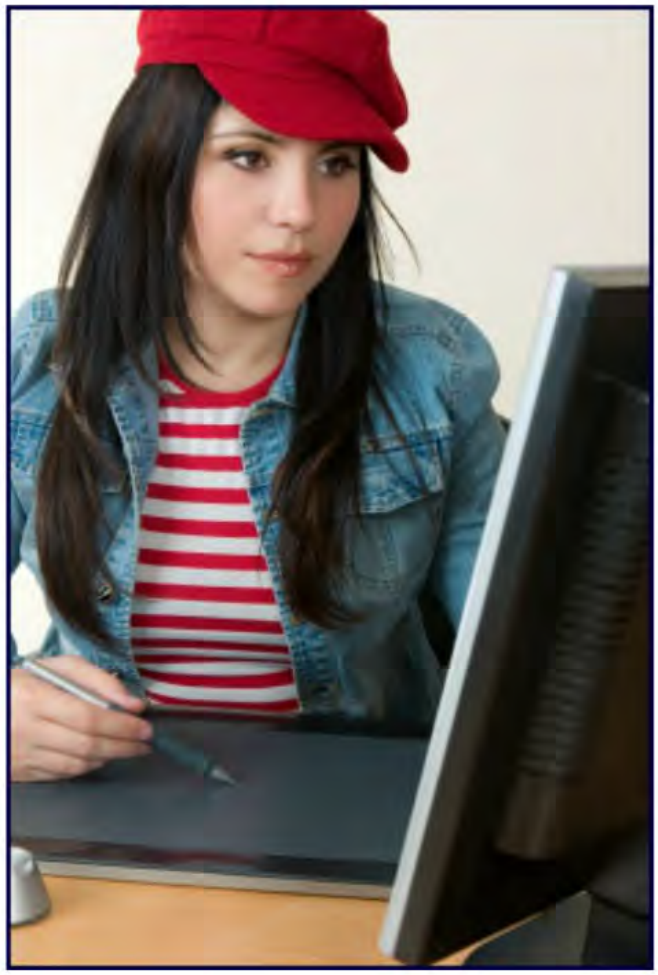
It's probably easier to start by explaining the problem the Design Manager tries to solve.

These people don't create in the same way

problem the Design Manager tries to solve.

These people don't create in the same way





nplex

It's probably easier to start by explaining the problem the Design Manager tries to solve.

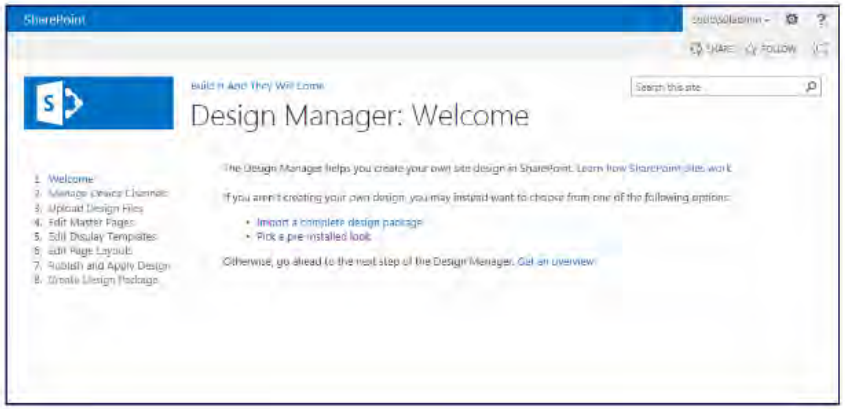
These people don't create in the same way




Developers

Designers

The Design Manager removes the need to work in code for many SharePoint-related design artifacts



The Design Manager makes a number of things easier

- Building and editing master pages
- Constructing publish page layouts
- Creating and manipulating display templates
- Packaging and applying designs
- Managing device channels

Demo

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http:/
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<title>BIATWC: A Design Manager Example</title>
</head>

<body>
    <h1>Perhaps the World's Simplest HTML Page</h1>
    <p>Any content that we'd want to insert would go here.</p>
</body>

</html>
```

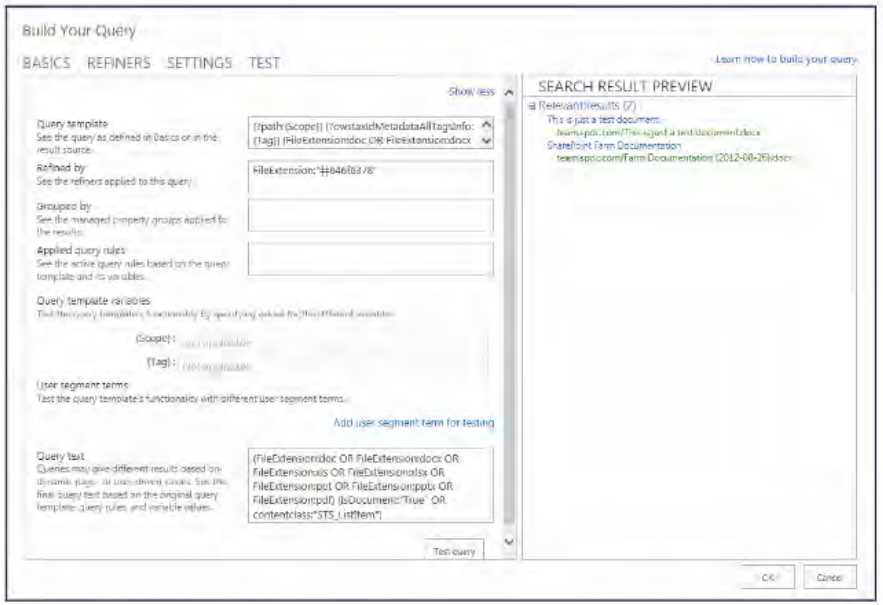
Wait … why the Design Manager Demo?

Needed to look closer …

... to understand where we're going next

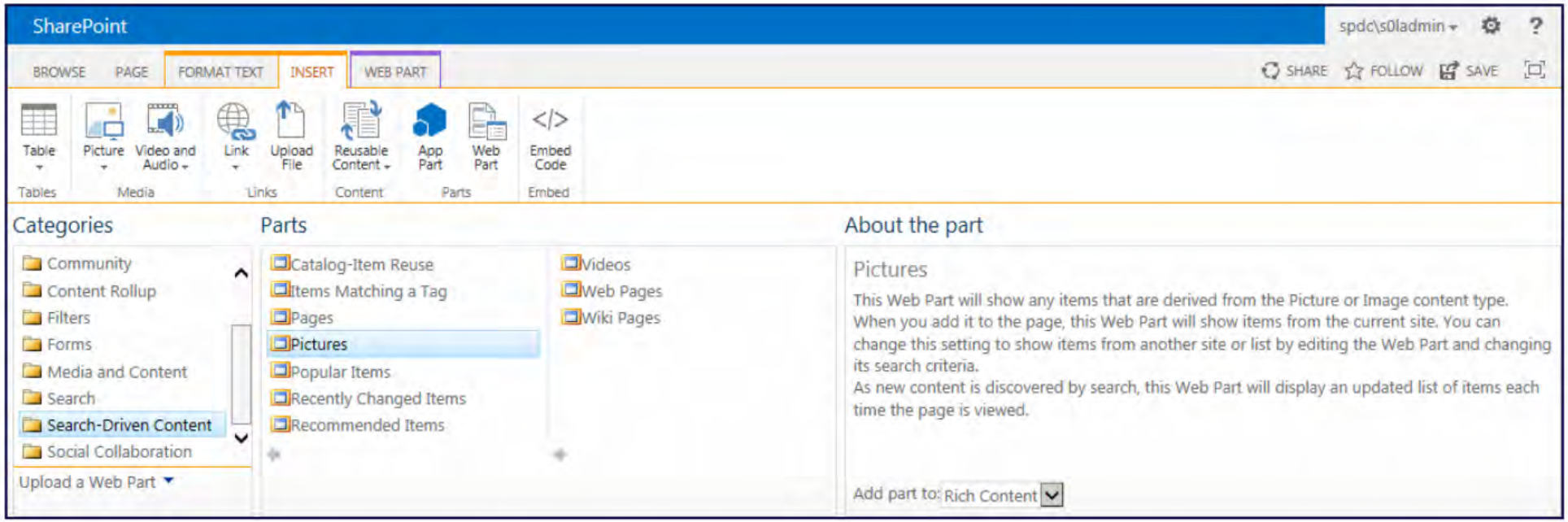
Content Search Web Part (CSWP)

Content Search Web Part (CSWP)



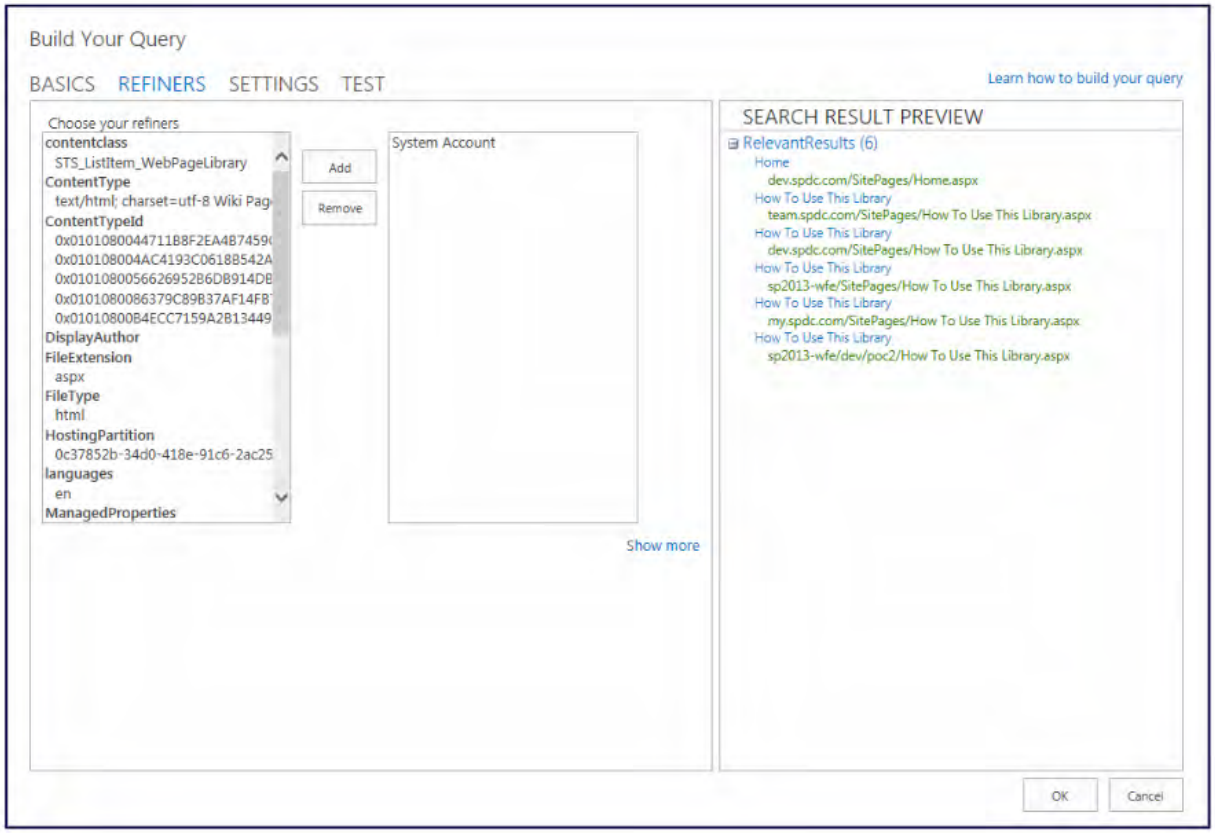
- Uses Search results to dynamically generate contents at run-time
- Includes a powerful query builder to easily construct queries with parameters, refiners, scope restrictions, sort rules, and much more
- Results shown are highly customizable using display templates

Start simple with pre-configured options



Search-Driven Content Web Parts

Step-up to the full power when needed



=

Demo

```
<html xmlns:mso="urn:schemas-microsoft-com:office:office" xmlns:msdt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882">
<head>
<title>BIATWC: Item Entry Template</title>

<!--[if gte mso 9]><xml>
<mso:CustomDocumentProperties>
<mso:TemplateHidden msdt:dt="string">0</mso:TemplateHidden>
<mso:ManagedPropertyMapping msdt:dt="string">
    'Link URL'{Link URL}:'Path',
    'Display Text'{Display Text}:'Title',
    'FileExtension',
    'SecondaryFileExtension'
</mso:ManagedPropertyMapping>
<mso:MasterPageDescription msdt:dt="string">This display template controls an individual item in the full page of
<mso:ContentTypeId msdt:dt="string">0x0101002039C03861C64EC4A04F5361F385106603</mso:ContentTypeId>
<mso:TargetControlType msdt:dt="string">;#Content Web Parts;#</mso:TargetControlType>
<mso:HtmlDesignAssociated msdt:dt="string">1</mso:HtmlDesignAssociated>
</mso:CustomDocumentProperties>
</xml><![endif]-->
</head>
```

Make the right choice

Use the CSWP when

- When you need content from outside the site collection
- When content selection is parameterized
- When you have access to on-premises SP2013 Enterprise
- When you need complete control over display formatting
- When it's okay if content shown is a little "stale"

In other cases, the CQWP may be an alternative

Last stop on this ride ...

The SharePoint 2013 Cloud App Model

(or just "App Model")

Before going to far, we should probably ask the question:

Why do we even have a new App Model?

To understand the "why," we need to rewind a bit and look at ...

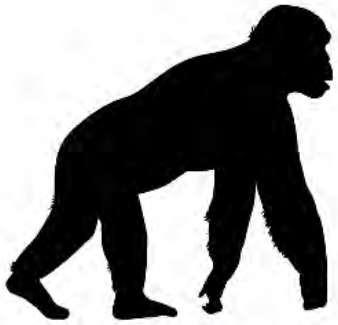
The Evolution of Custom Application Development

SharePoint Ed

and look at ...

The Evolution of Custom Application Development

SharePoint Edition



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

2013

- The App Model is brand-now w
- Full Visual Studio development
- Sidesteps limitations associate
- All-but-guarantees a better ope
- Only model that will work in Of

Share

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 ...

ePoint Edition

2013

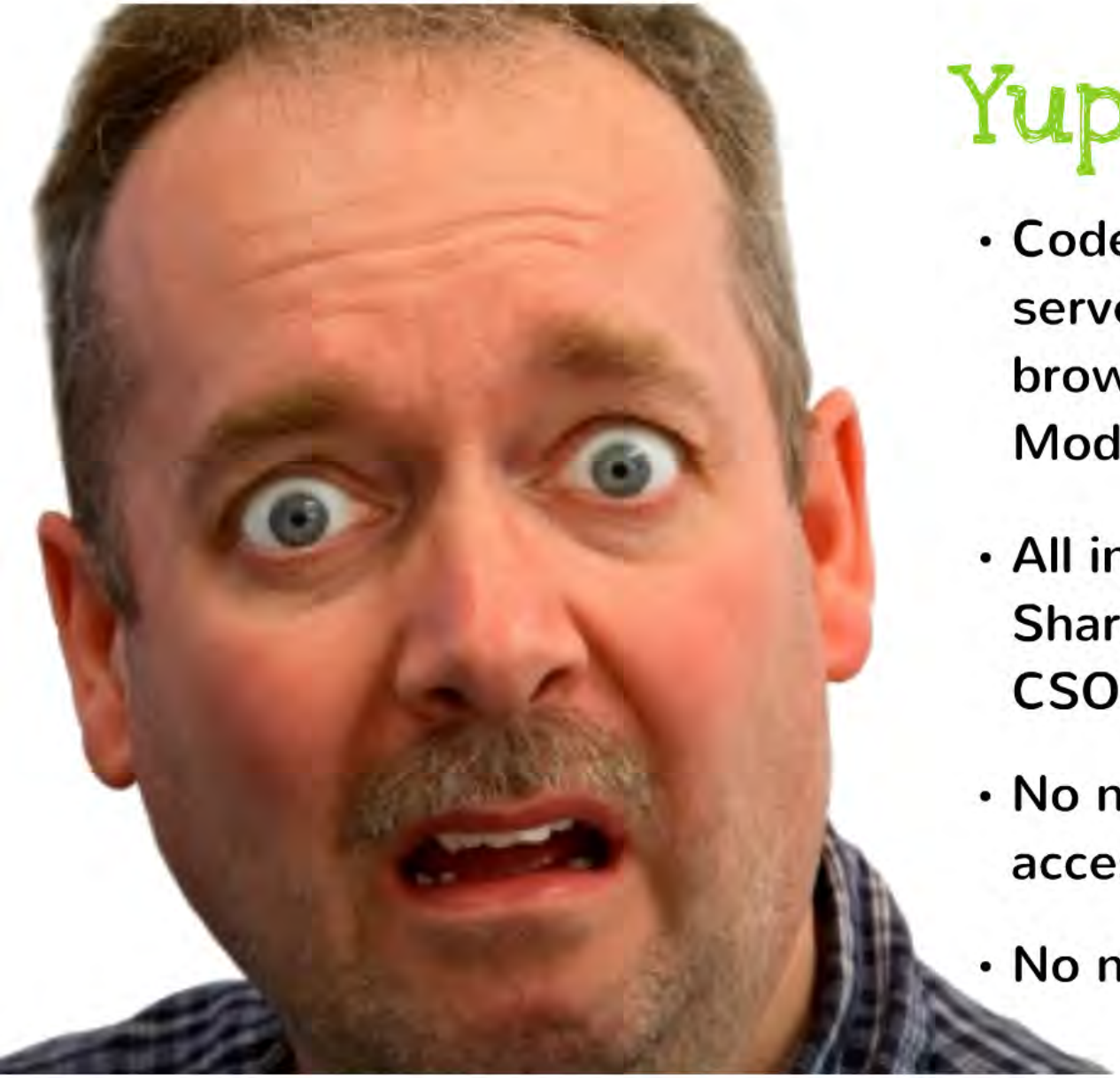
The Cloud

- The App Model is brand-now 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 model that will work in 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?

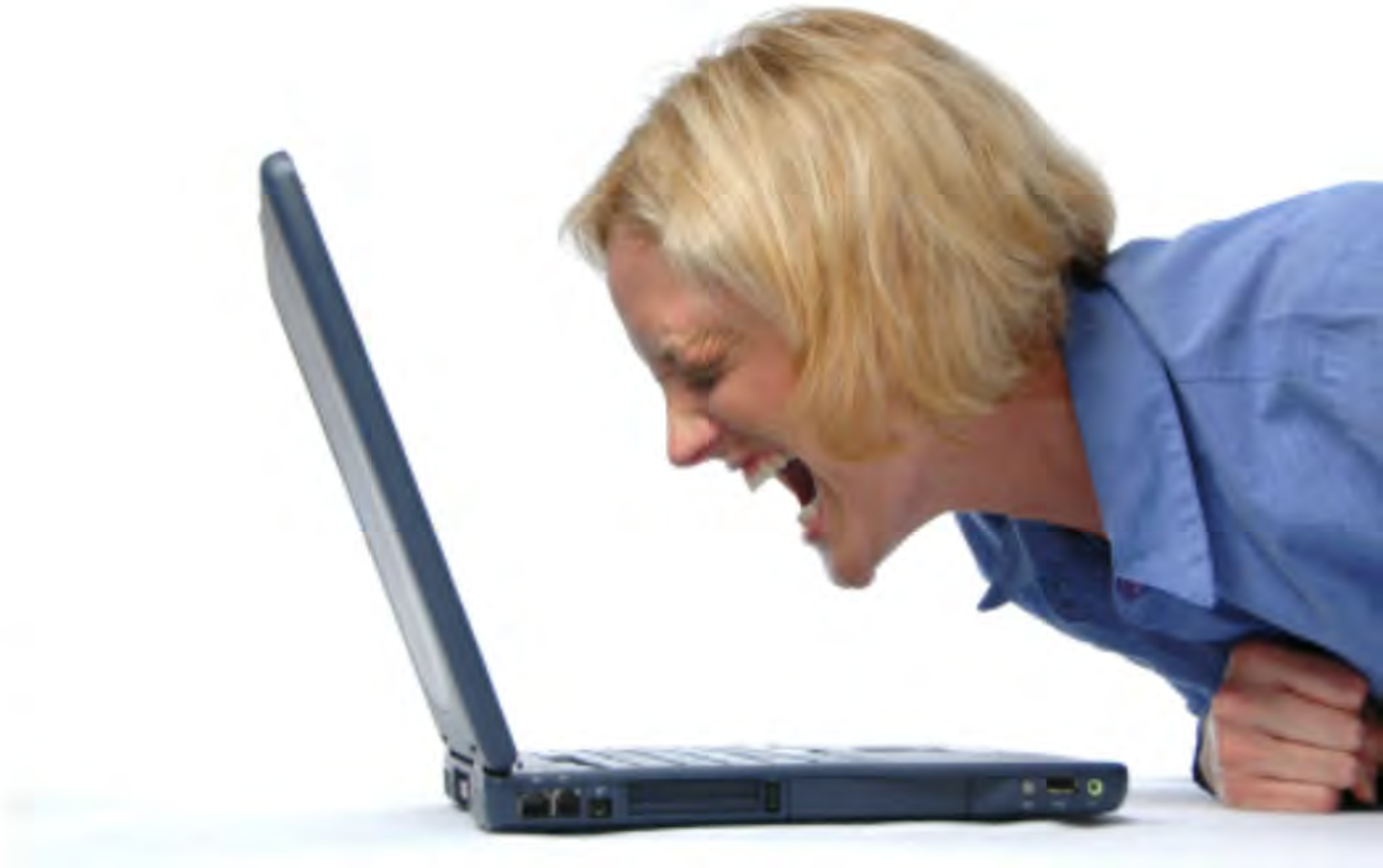


Yup. That's Right.

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

Feel the need to rage a bit?

It's okay



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

Why the App Model is here to stay

5. Acceptance

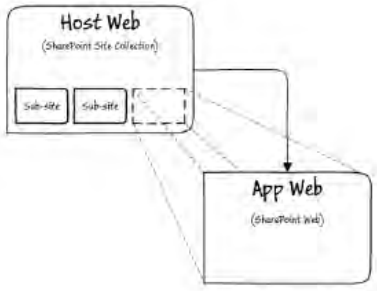
Why the App Model is here to stay

- Improves SharePoint performance
- Simplifies upgrades (to farm and custom code)
- Simplifies troubleshooting
- Is open to developers on other stacks (e.g., J2EE)
- Needed for The Cloud and Office 365

The App Model

"Cloud-Hosted Apps"

SharePoint-Hosted Apps



ros

develop

ecurity concerns

deploy

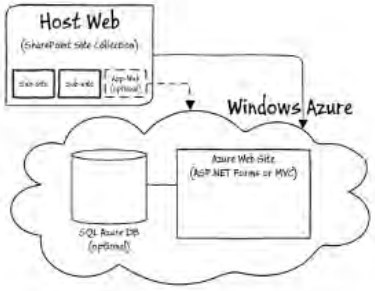
Cons

No compiled code

No unattended execution

Generally less capable

Autohosted Apps ("Azure-Hosted Apps")



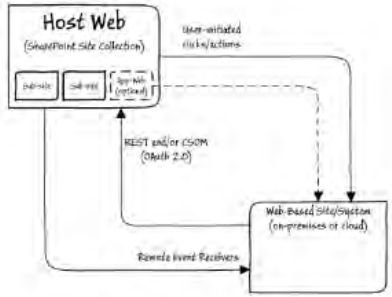
Pros

Much more capable

able but

mplex

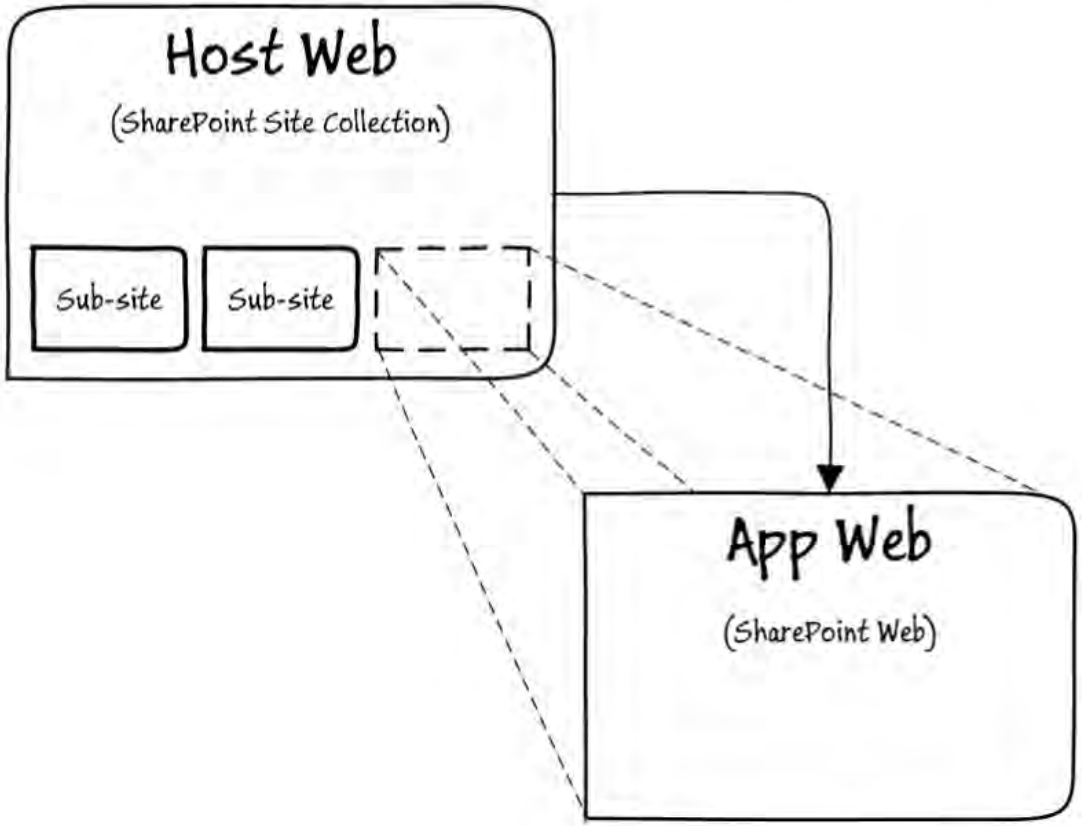
Provider-Hosted Apps



Cons

(Potentially) complex deployment

SharePoint-Hosted Apps

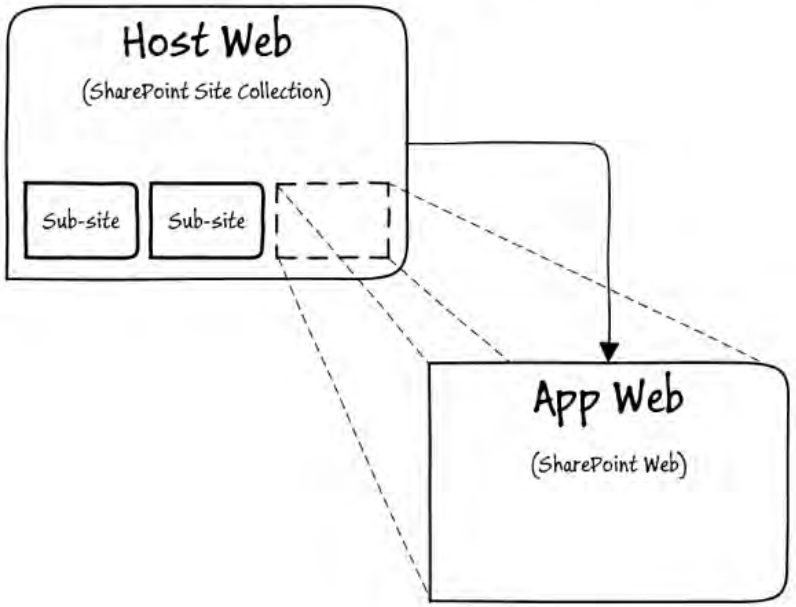


Pros

Cons

SharePoint-Hosted Apps

Simplest of new App Model options



Pros

Easier to develop
Fewer security concerns
Easier to deploy
Less complex

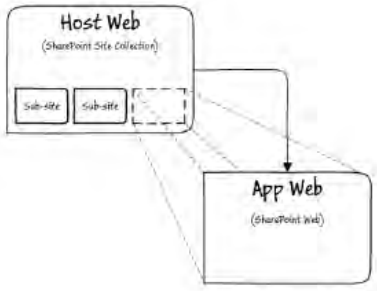
Cons

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

The App Model

"Cloud-Hosted Apps"

SharePoint-Hosted Apps



ros

develop

ecurity concerns

deploy

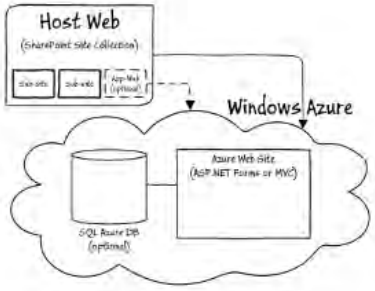
Cons

No compiled code

No unattended execution

Generally less capable

Autohosted Apps ("Azure-Hosted Apps")



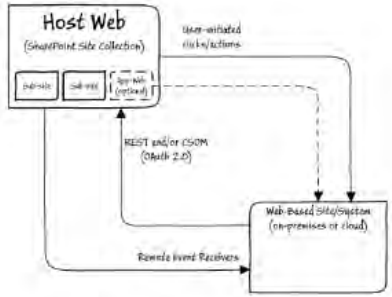
Pros

Much more capable

able but

mplex

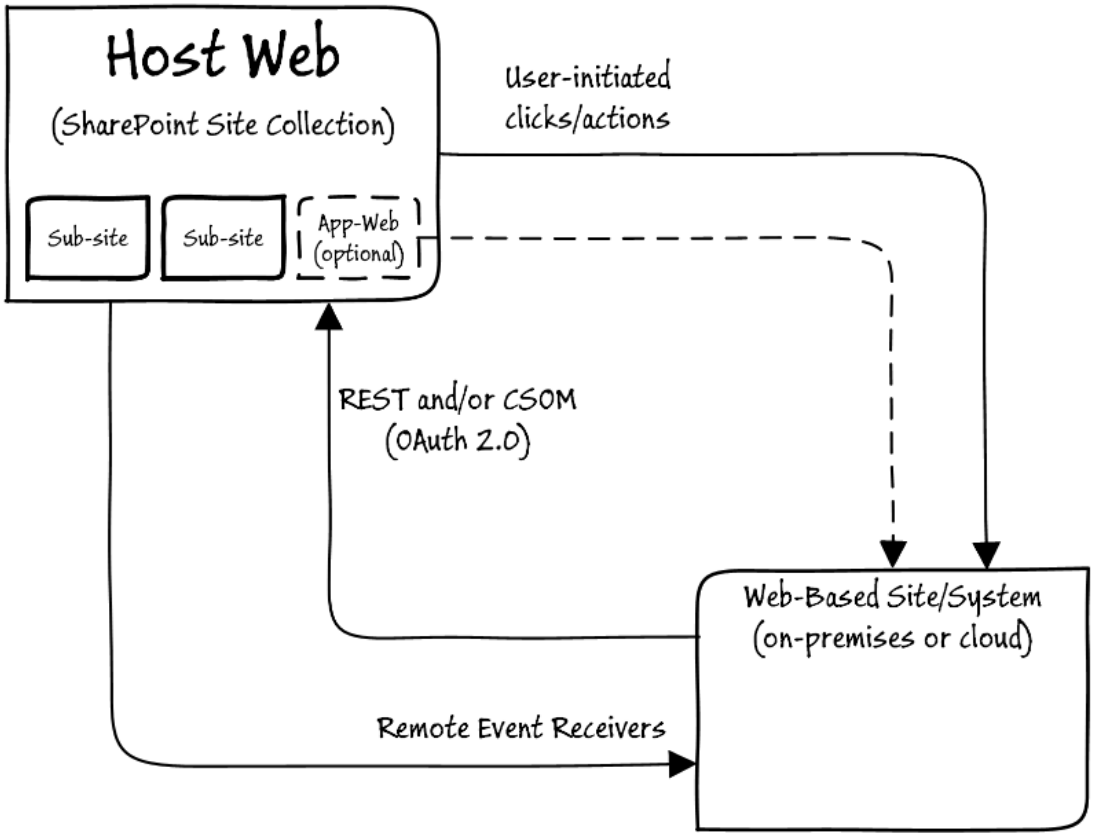
Provider-Hosted Apps



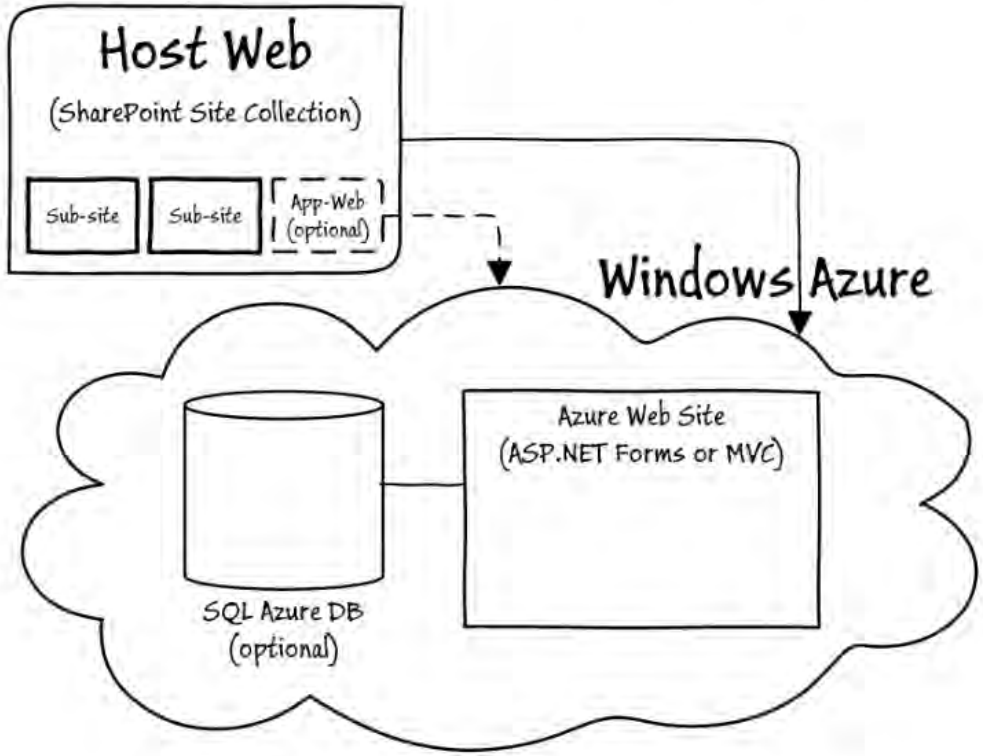
Cons

(Potentially) complex deployment

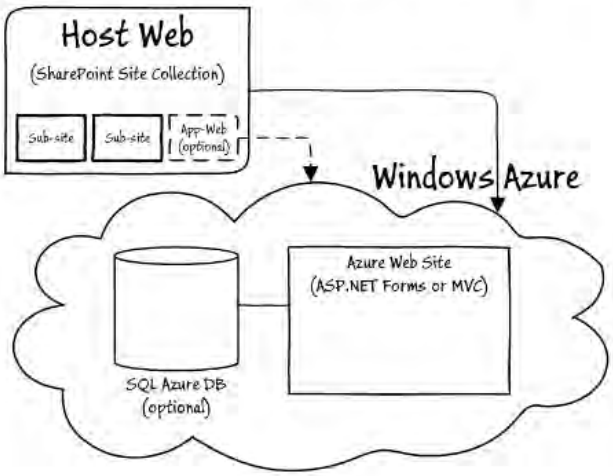
Provider-Hosted Apps



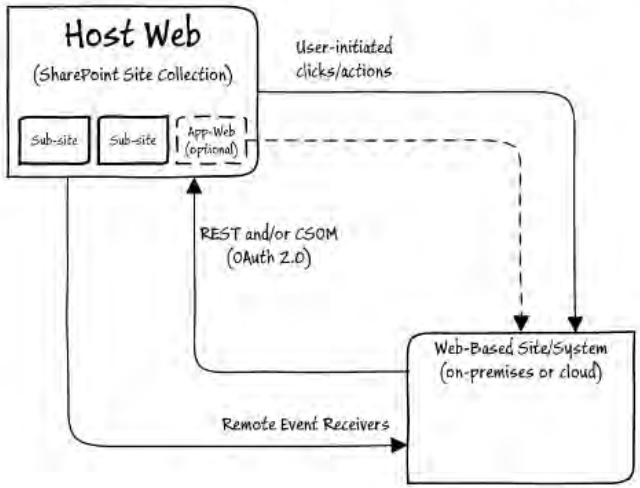
Autohosted Apps ("Azure-Hosted Apps")



Autohosted Apps ("Azure-Hosted Apps")



Provider-Hosted Apps



Pros

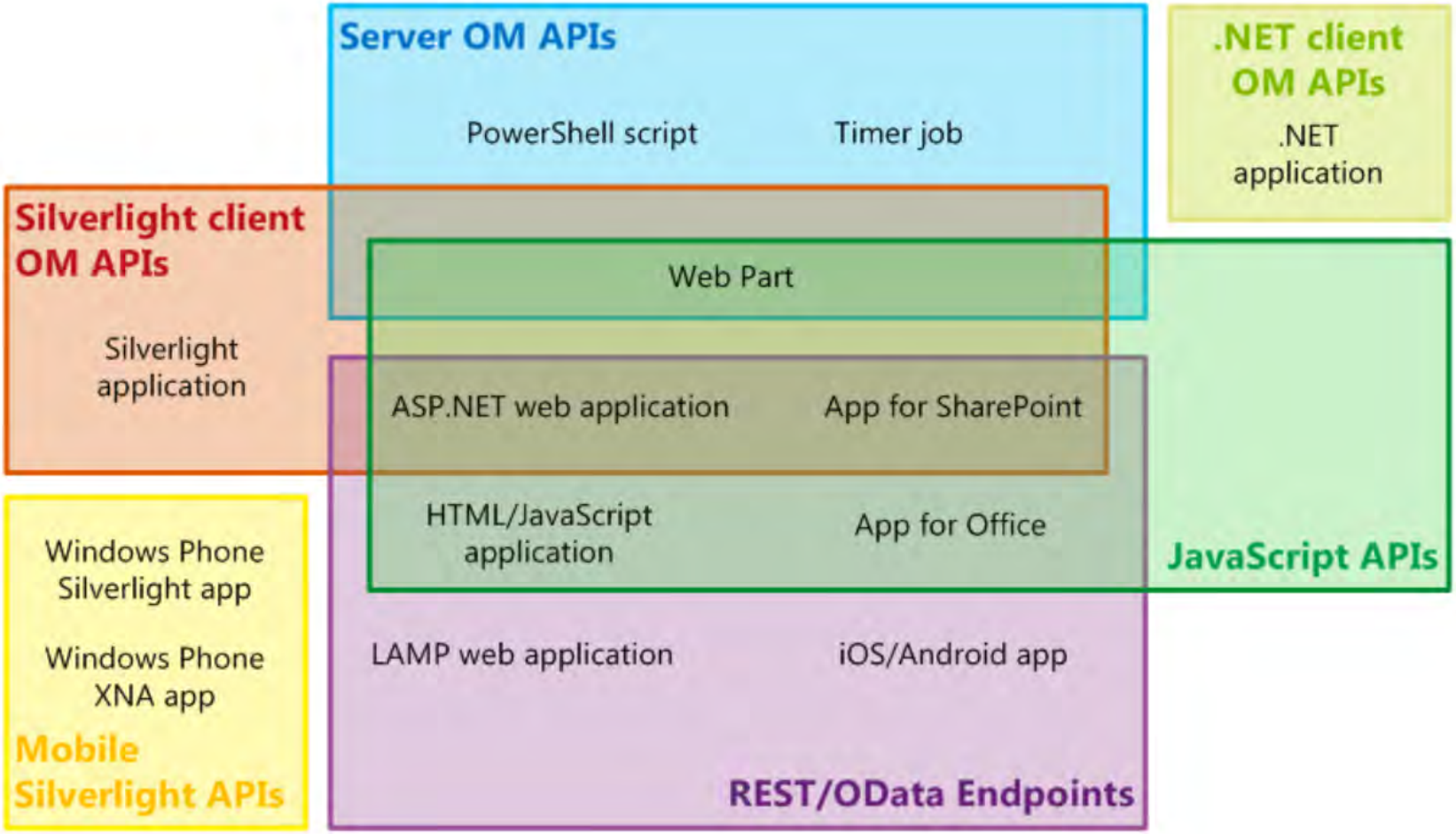
Much more capable
Apps can act on their own
Open to non-SharePoint devs
You control everything

More capable but more complex

Cons

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

Choose the right API and strategy for the task

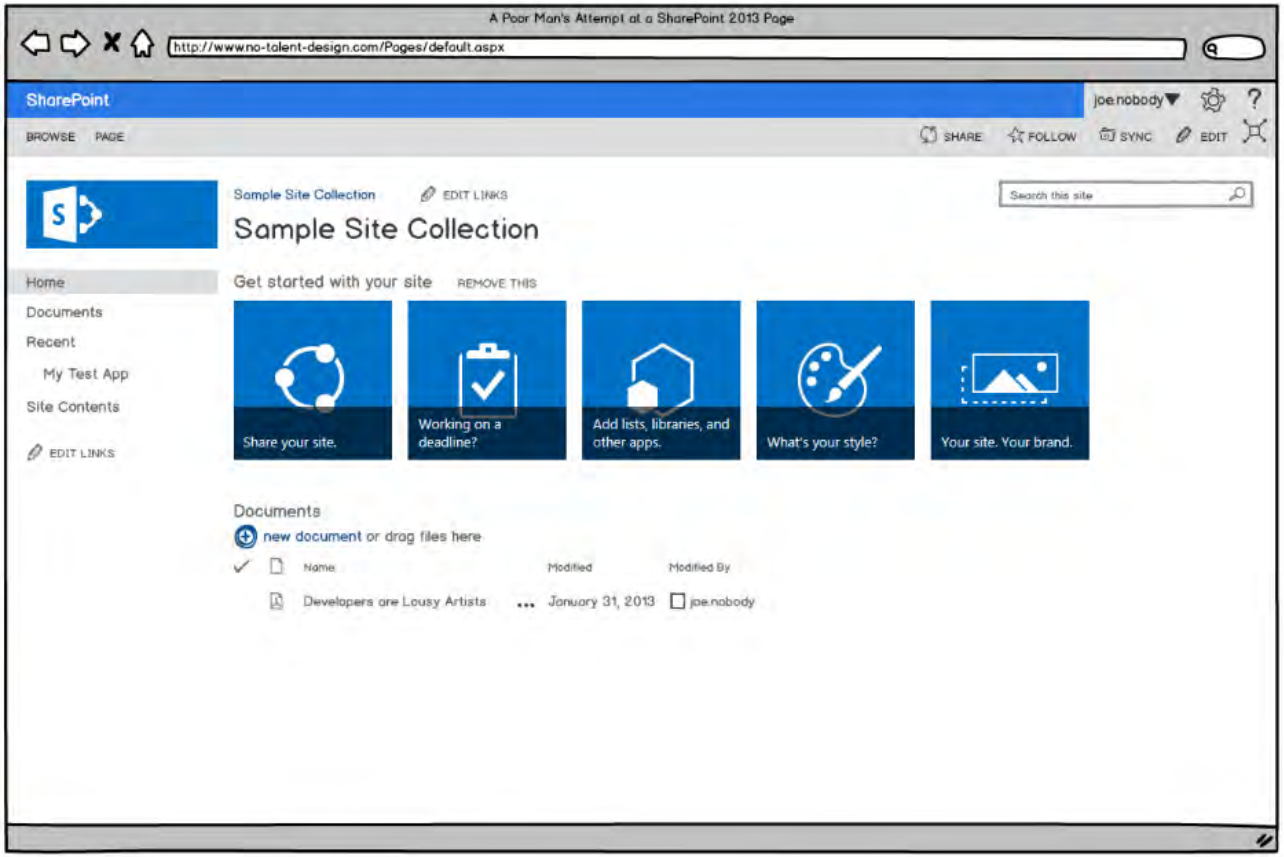


So you might be wondering …

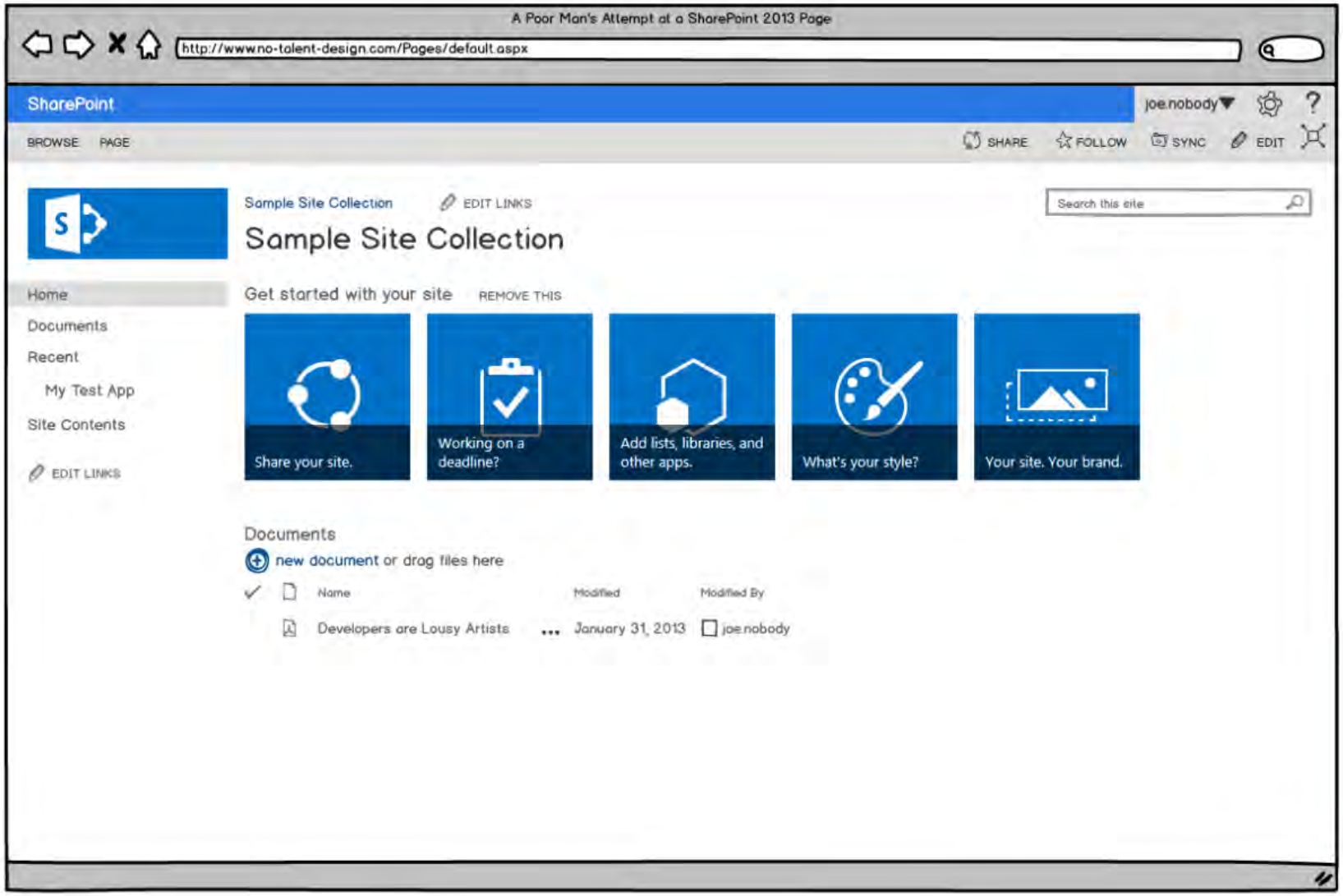
If Apps are isolated from SharePoint, how do they show up in a site?

Apps can be surfaced in three ways

Team Site Mockup



surfaced in three ways



IMMERSIVE FULL PAGE

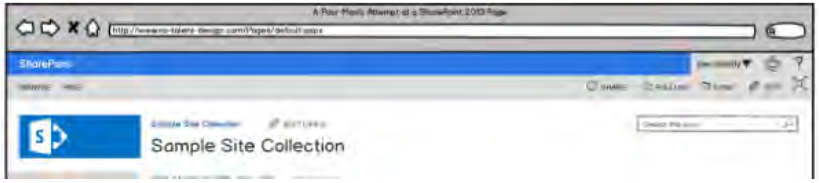
UI CUSTOM ACTIONS

PART

IMMERSIVE FULL PAGE

- App gets complete control of a page
- Can look like it's part of a SharePoint site, but not a requirement

UI CUSTOM



User Interface Guidance

http://www.mycloudhost.com/uiguidance.html

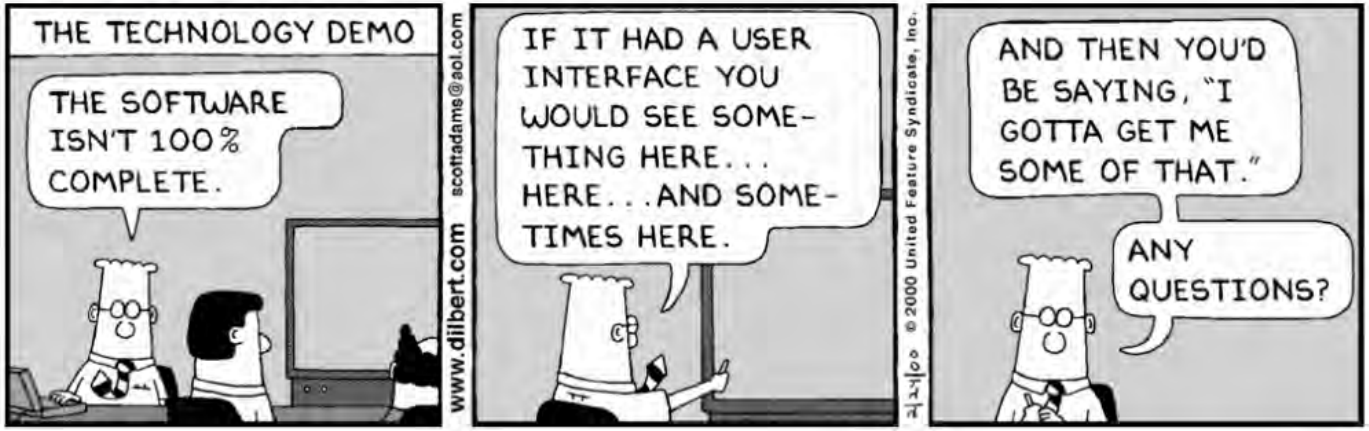
Sample Site Collection

BROWSE

My Test App

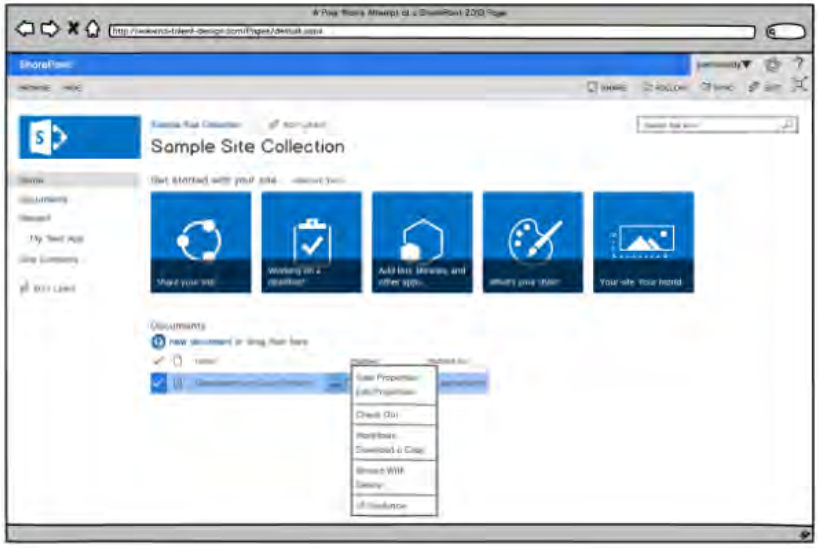
User Interface Guidance

Welcome to the Immersive Page Experience!



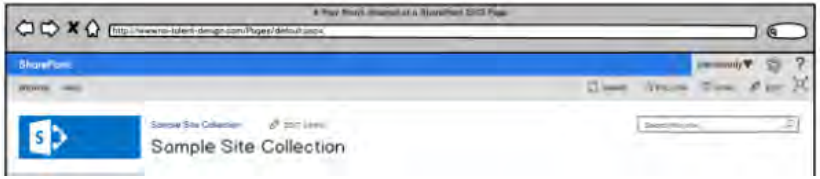
SharePoint site, but not a requirement

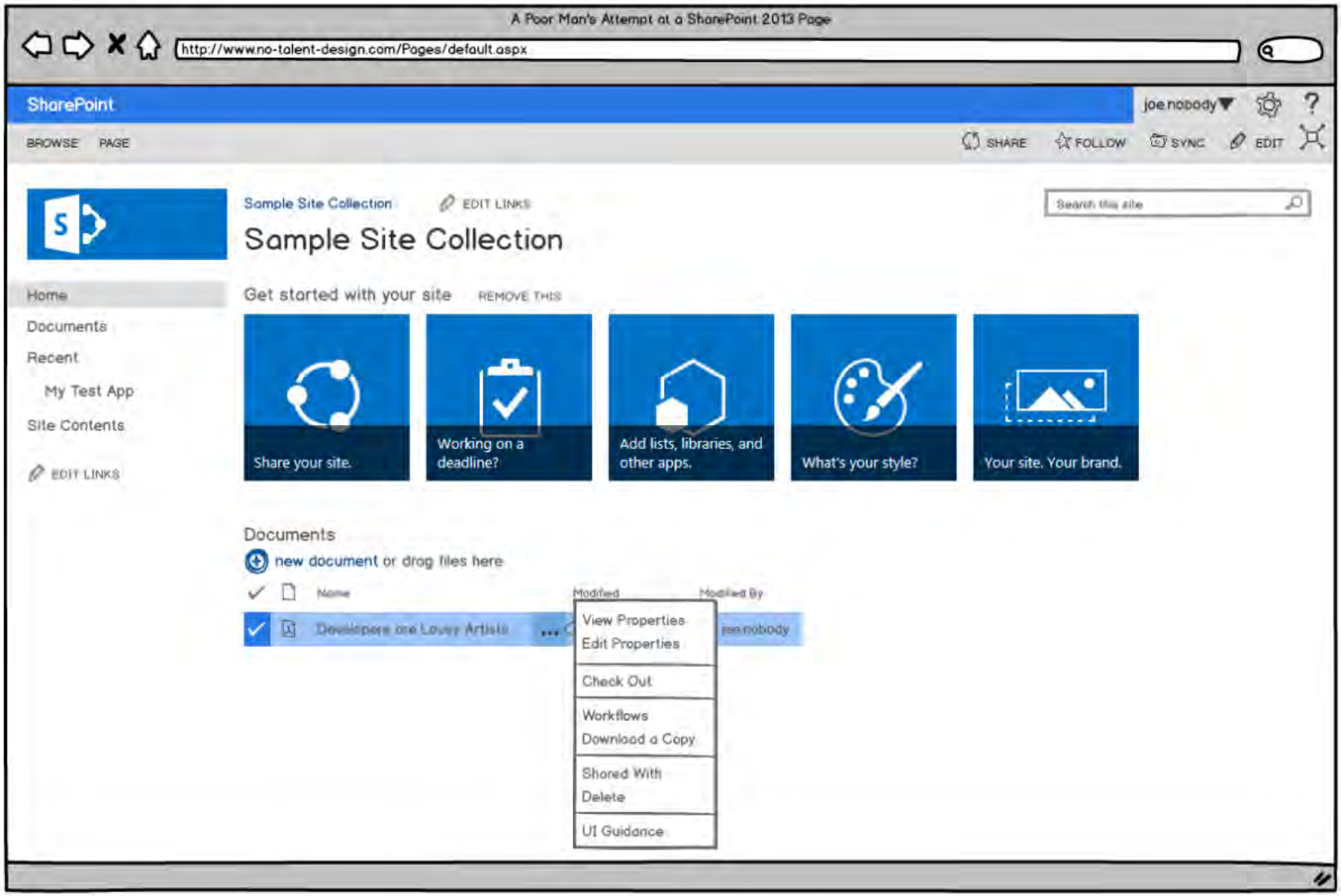
UI CUSTOM ACTIONS

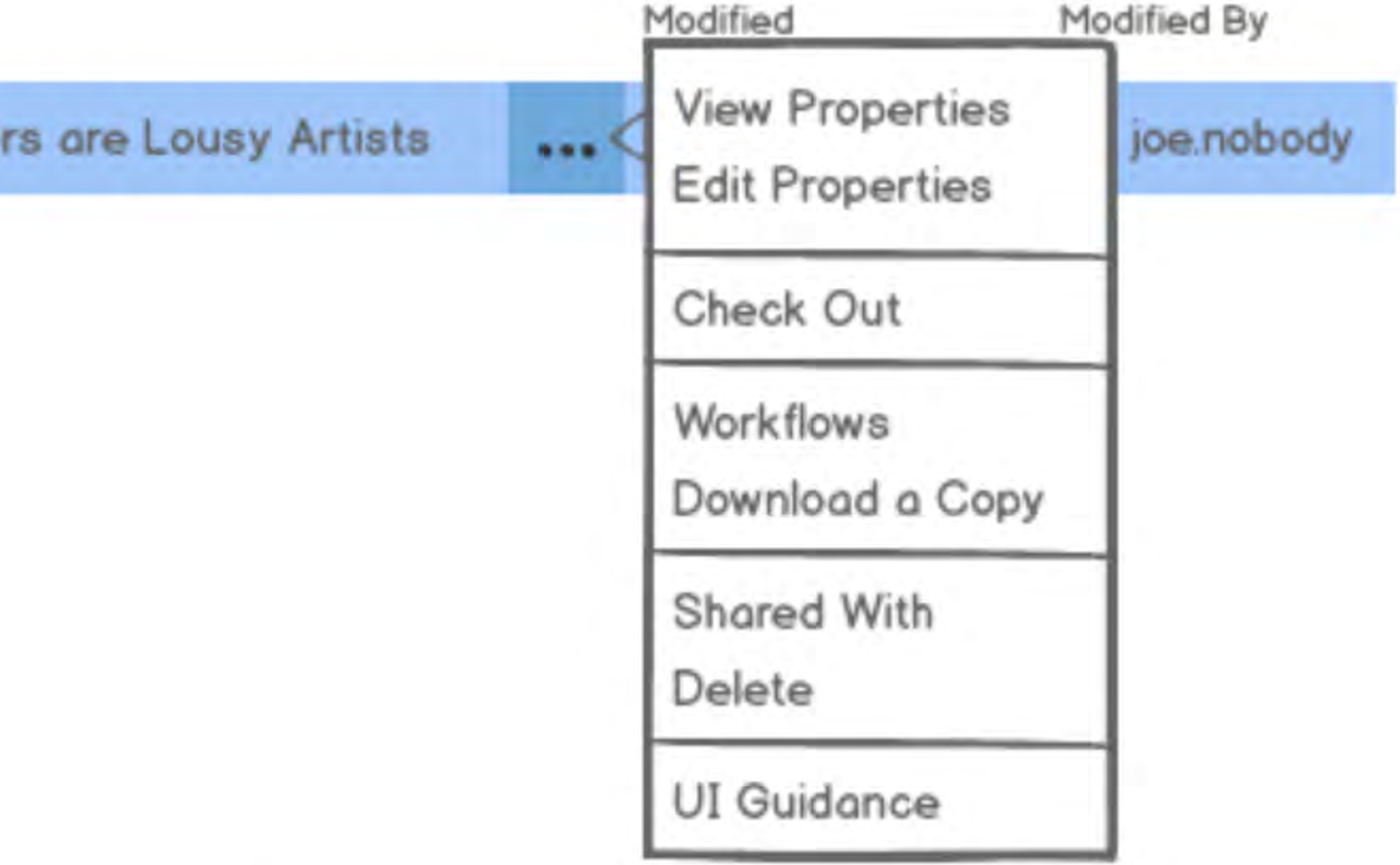


- Ribbon custom actions (buttons) are one possibility
- Edit Control Block (ECB) custom actions are the other option (as shown)

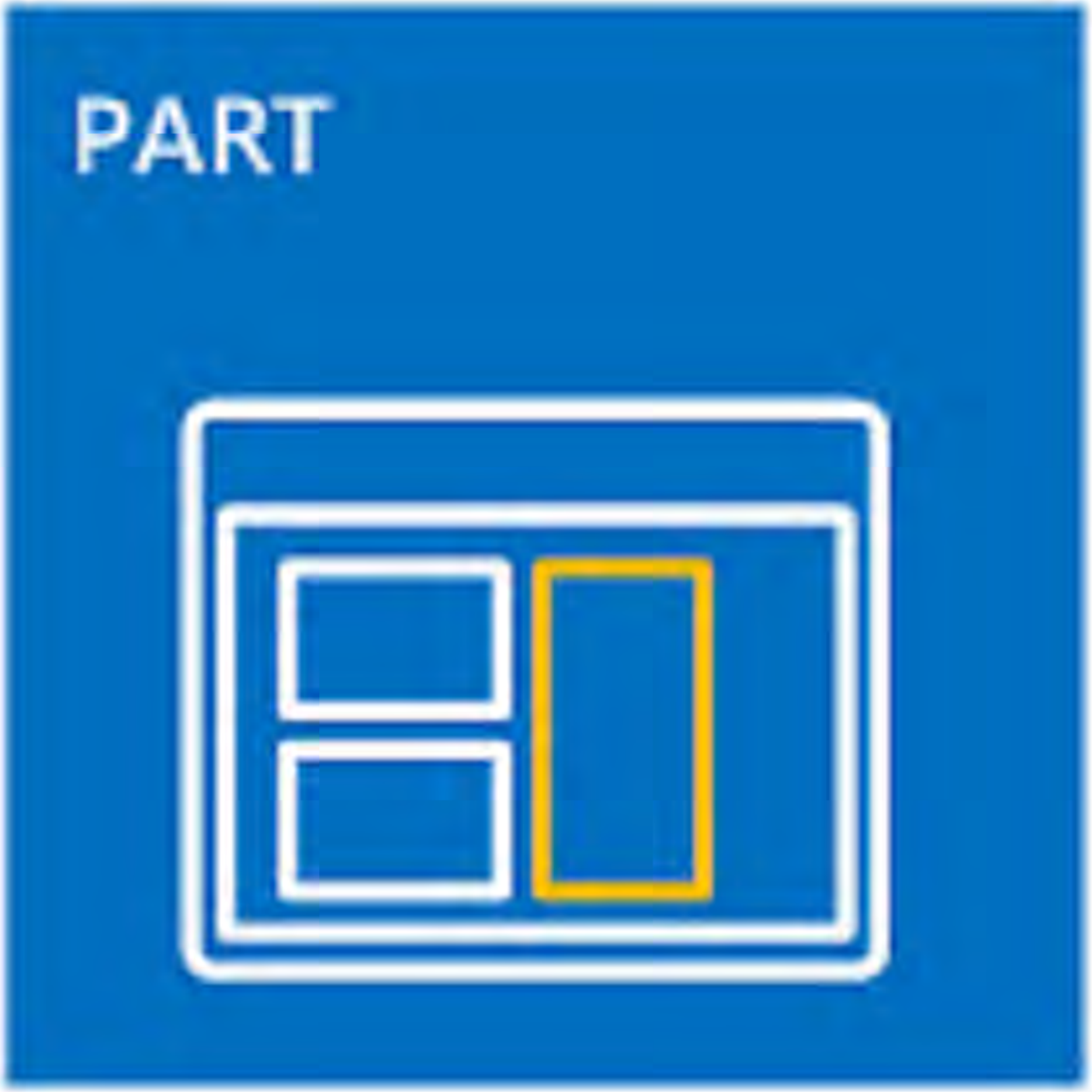
PART

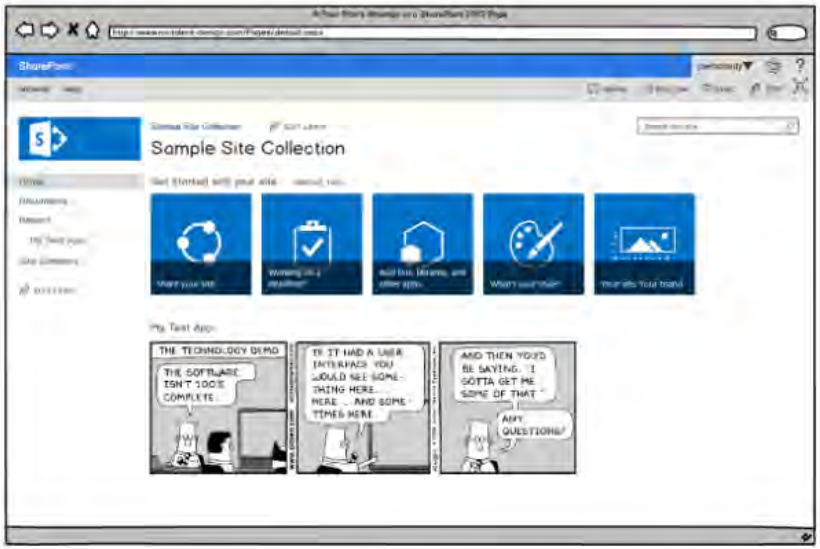



A Poor Man's Attempt at a SharePoint 2013 Page
http://www.no-talent-design.com/Pages/default.aspx
SharePoint
joe.nobody
BROWSE
PAGE
SHARE
FOLLOW
SYNC
EDIT
Sample Site Collection
EDIT LINKS
Search this site
Sample Site Collection
Home
Documents
Recent
My Test App
Site Contents
EDIT LINKS
Get started with 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
joe.nobody
View Properties
Edit Properties
Check Out
Workflows
Download a Copy
Shared With
Delete
UI Guidance

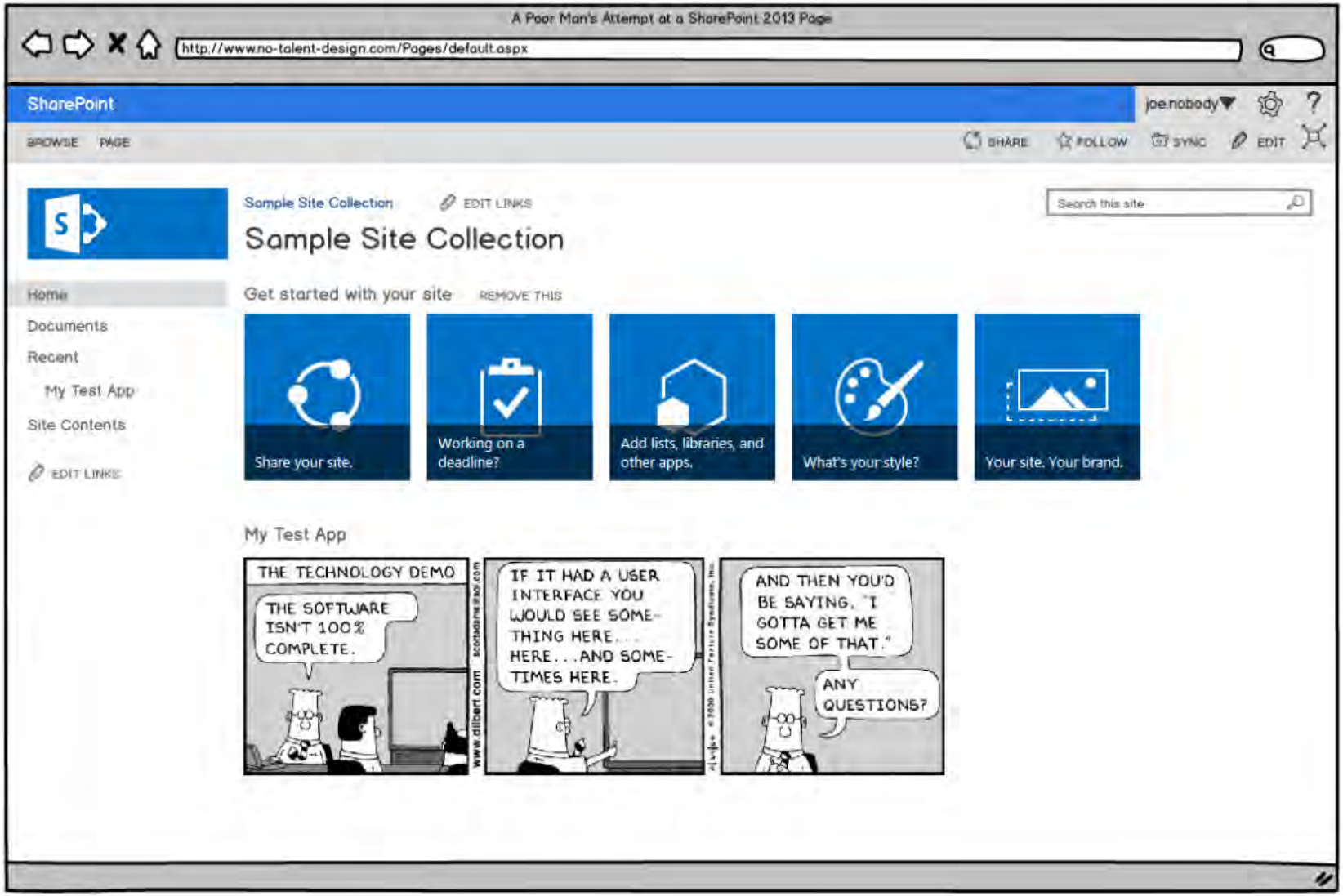

Modified
Modified By
rs are Lousy Artists
...
joe.nobody
View Properties
Edit Properties
Check Out
Workflows
Download a Copy
Shared With
Delete
UI Guidance

custom actions are the other option (as shown)





- App Parts look and feel like Web Parts - no Web Part connections, though
- App Parts are basically just IFRAMEs to an App custom page



- App Parts look and feel like

A related thought …

If you have another software product (SaaS), this integration and exposure pattern simplifies making your offering available through SharePoint.

Demo

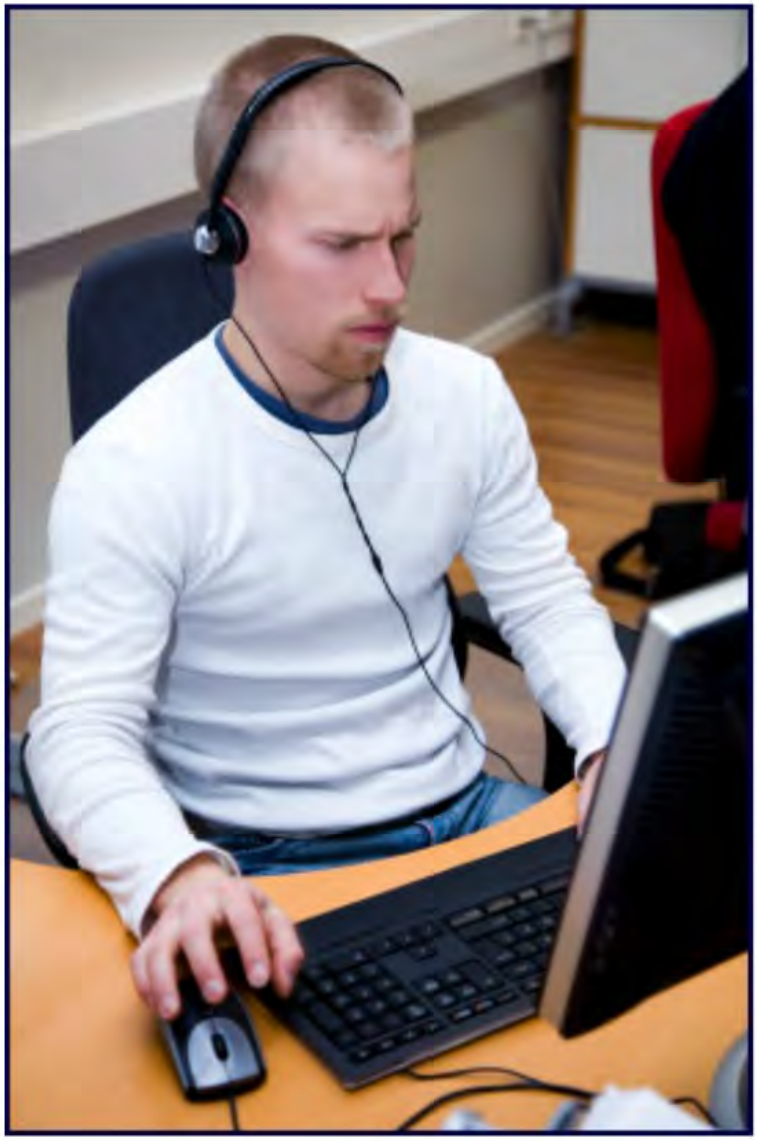
The Implications

gets better!

For devs, it's

Admins

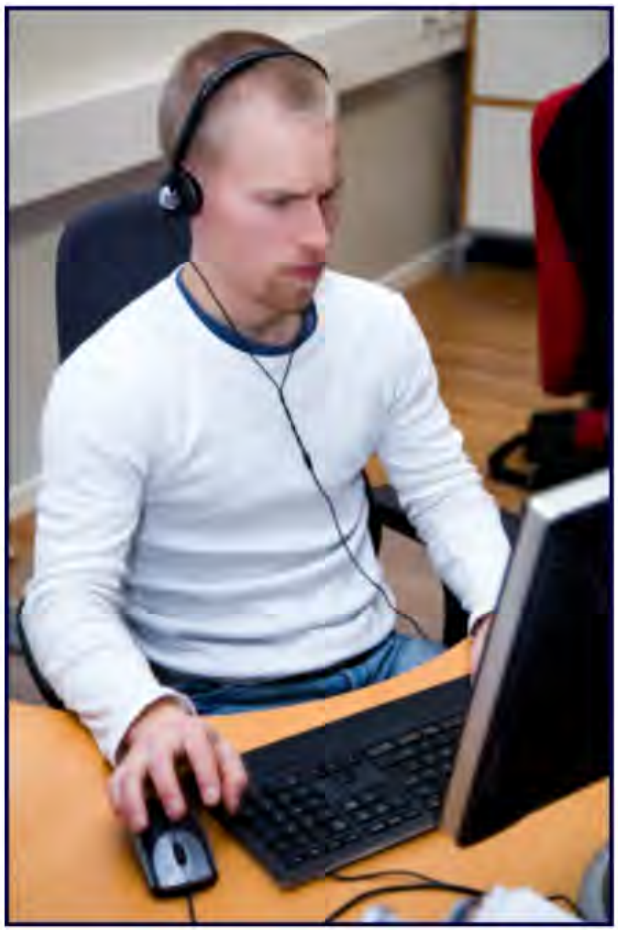
Devs



For admins, life generally gets better!

- Customization and development doesn't impact farm performance
- Troubleshooting boundaries get much clearer
- Less involvement in deployment and fewer related issues
- Patching and platform upgrades can proceed independent of code!

For devs, it's something of a mixed bag



- Grumpy old devs: time to learn some new skills
- Client-side focus means no more free passes on security, SharePoint context, and other "givens"
- Decoupling now makes it easier for continuous integration, TDD, and other previously difficult practices

Build it and they'll come?

The jury's out, but SharePoint 2013 provides plenty of customization and development options to suit your needs

Sean P. McDonough

customization and development options to suit your needs

Sean P. McDonough

SharePoint Gearhead, Developer, and Evangelist

Twitter: @spmcdonough

Blog: http://SharePointInterface.com

About: http://about.me/spmcdonough