SOLID FOUNDATIONS: BUILDING SOLUTIONS IN AN EARTHQUAKE-PRONE SHAREPOINT AREA

Sean P. McDonough

Senior Solution Architect and Consultant

Akumina

REJECTED COVER IMAGES AND TITLES

QUESTIONABLE FOUNDATIONS

Building Solid SharePoint Solutions Is Anything But A Science

#3

SHAREPOINT SOLUTION CREATION

Thought Of By Many As A Plague Of Biblical Proportions

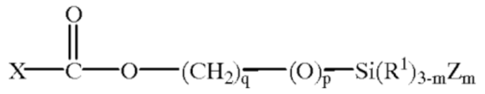
#2

LASTING SHAREPOINT SOLUTIONS

The Earth Is More Likely To Suffer A Meteor Strike

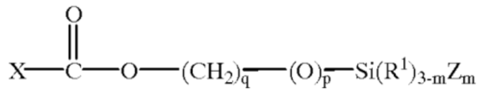
#1

A BIT ABOUT ME …

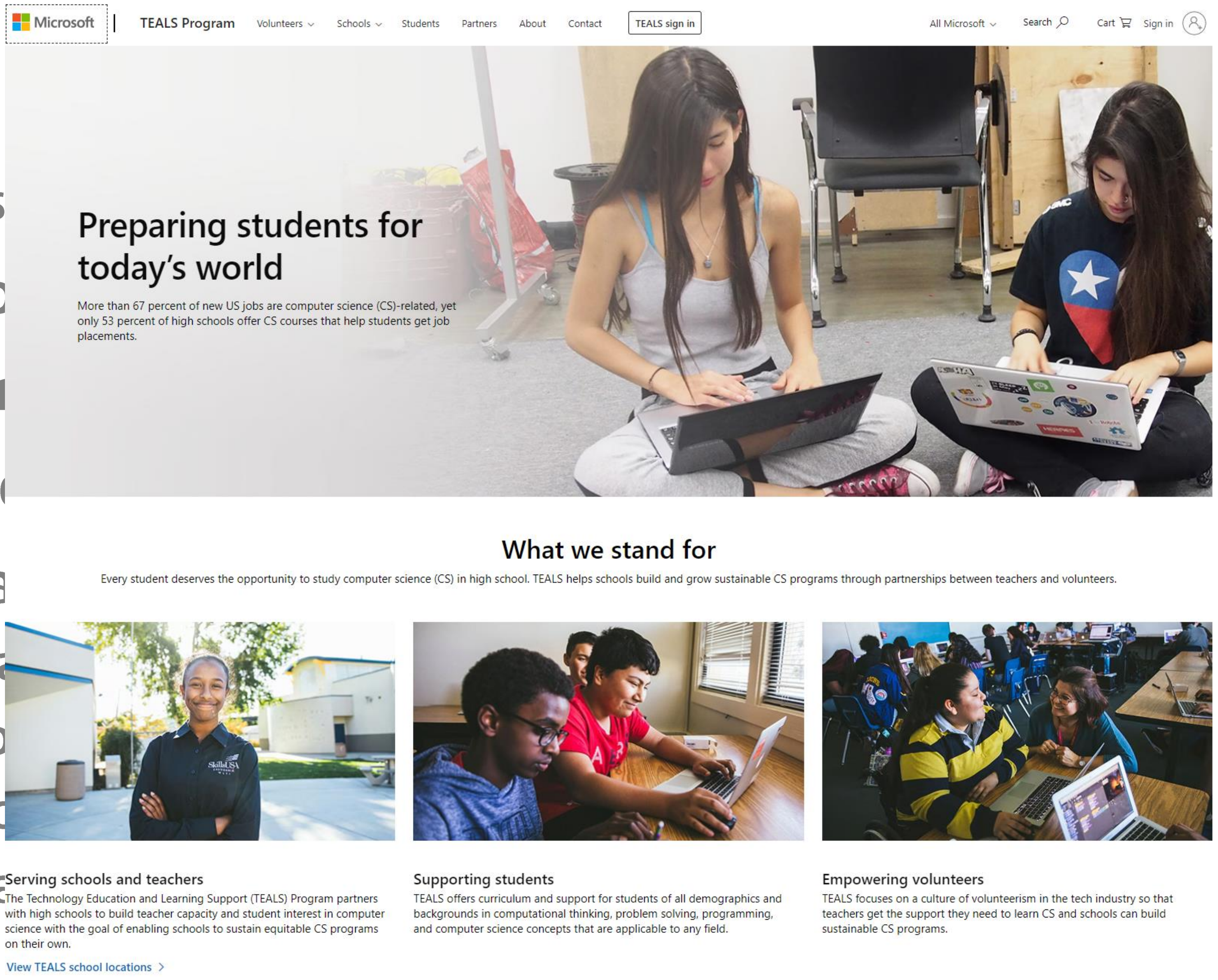


- Started professional career as a polymer chemist for P&G
  - Transitioned internally to Information Systems
- Developing software professionally since mid ’90s
- Focus primarily on SharePoint since 2004
- Awarded Microsoft MVP in 2016 (Office Apps & Services)
- Nowadays, I work for Akumina in based in Nashua, NH.
  - Operational and performance-based troubleshooting/remediation
  - Custom development of scripting aids and solutions
  - Microsoft 365 and Akumina-based intranet creation and support
  - Educational and non-profit technical services

A BIT ABOUT ME …



- Started … &G
  - Trans
- Develop
- Focus pr
- Awarde … es)
- Nowada
  - Opera … diation
  - Custo
  - Micro … pport
  - Educa



Preparing students for today's world

More than 67 percent of new US jobs are computer science (CS)-related, yet only 53 percent of high schools offer CS courses that help students get job placements.

What we stand for

Every student deserves the opportunity to study computer science (CS) in high school. TEALS helps schools build and grow sustainable CS programs through partnerships between teachers and volunteers.

Serving schools and teachers

The Technology Education and Learning Support (TEALS) Program partners with high schools to build teacher capacity and student interest in computer science with the goal of enabling schools to sustain equitable CS programs on their own.

View TEALS school locations >

Supporting students

TEALS offers curriculum and support for students of all demographics and backgrounds in computational thinking, problem solving, programming, and computer science concepts that are applicable to any field.

Empowering volunteers

TEALS focuses on a culture of volunteerism in the tech industry so that teachers get the support they need to learn CS and schools can build sustainable CS programs.

MVP
Microsoft® Most Valuable Professional

Microsoft TEALS Program

akumina

bitstream FOUNDRY

GOAL OF TODAY'S PRESENTATION

- SharePoint's plethora of app options over the years
- Strategic decision points and questions to ask when designing a solution using SharePoint
- Overview of options currently considered "viable" for solutions
- Avoiding painting yourself into a corner with a solution approach
- Knowing what's deprecated/dead
- But first …

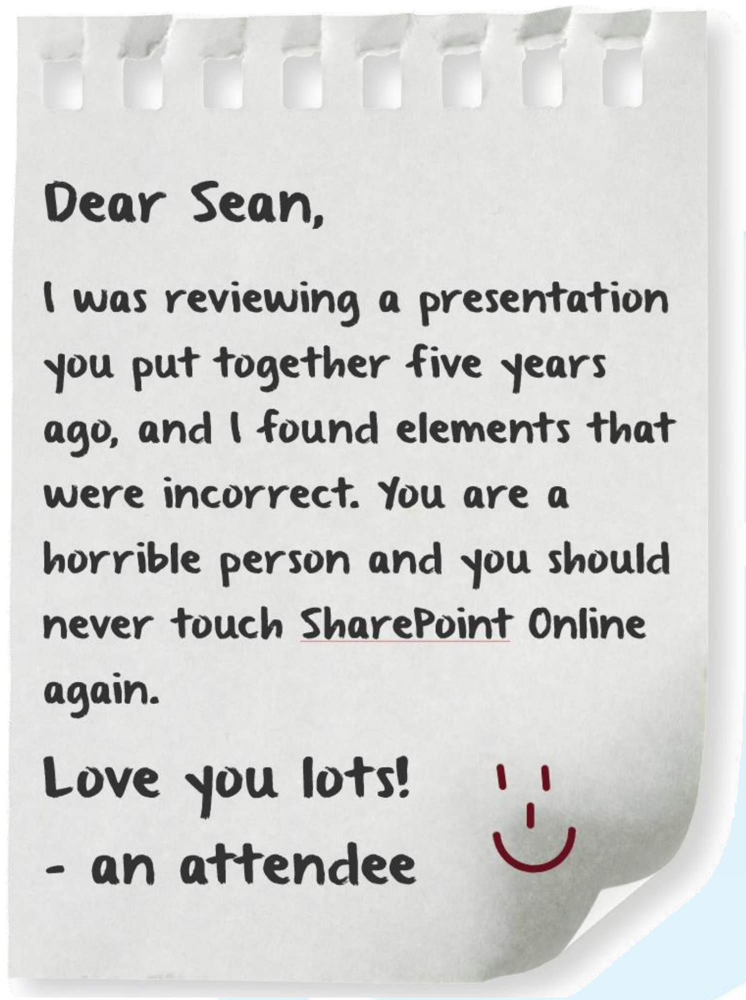
A BIT ABOUT "BORN ON" DATES

A BIT ABOUT "BORN ON" DATES

- This is Microsoft 365 (M365) and SharePoint Online (SPO).
- Microsoft refers to these services as "evergreen services."
- That designation carries some implications …
- Some of the information I'm going to share has a "born on" date and will expire in time.

A BIT ABOUT "BORN ON" DATES (CONTINUED)

Please don't look me up in five years and send me "fan mail" because I presented something that is no longer accurate due to a SharePoint (Online) service change. You may laugh, but it ~~will happen~~ has happened!



Dear Sean,

I was reviewing a presentation you put together five years ago, and I found elements that were incorrect. You are a horrible person and you should never touch SharePoint Online again.

Love you lots!

- an attendee

OPTIONS OVER THE YEARS …

SOLUTIONS TECHNOLOGY TIMELINE

- SharePoint Portal Server (SPS) 2003, WSSv2, and earlier
  - Power Users: FrontPage 2003
  - Devs had one option: compiled code running on the server
  - Zero developer assistance or support (aside from Orca and MSIs)
- Microsoft Office SharePoint Server (MOSS) 2007, WSSv3
  - Power Users: SharePoint Designer 2007, WF-based options*, InfoPath
  - Developers: (Full-Trust) Features/Solutions, SOAP-based web services*, SPServices
- SharePoint Server 2010, SPF 2010
  - Power Users: SharePoint Designer 2010, WF-based options*, InfoPath
  - Developers: Visual Studio Support for Feature/Solution creation, SOAP-based web services, SPServices, WCF-based web services, PowerShell, …

bitstream FOUNDRY

<INSERT TECHTONIC SHIFTING HERE />

HUH? (A.K.A. SOLUTIONS TIMELINE CONTINUED)

SharePoint 2013 and Beyond ("The Rift")

- Power-U
  - ~~Share~~
  - ~~InfoPa~~
  - ~~Share~~
  - ~~Share~~
  - Power
  - Extern

dd-Ins

s

(CSOM)

PFx)

mbedded

KEY DECISION POINTS AND QUESTIONS

DIRECTIONAL PLATFORM CONSIDERATIONS

- On-premises, in the cloud, or both?
  - Probably the single biggest factor that will drive additional decisions and design.
  - Some design options are only available on-prem (and vice-versa for the cloud).
  - Cloud doesn't necessarily mean SPO (but in the majority of cases, it does).
  - Microsoft's approach to innovation is "mobile first, cloud first."

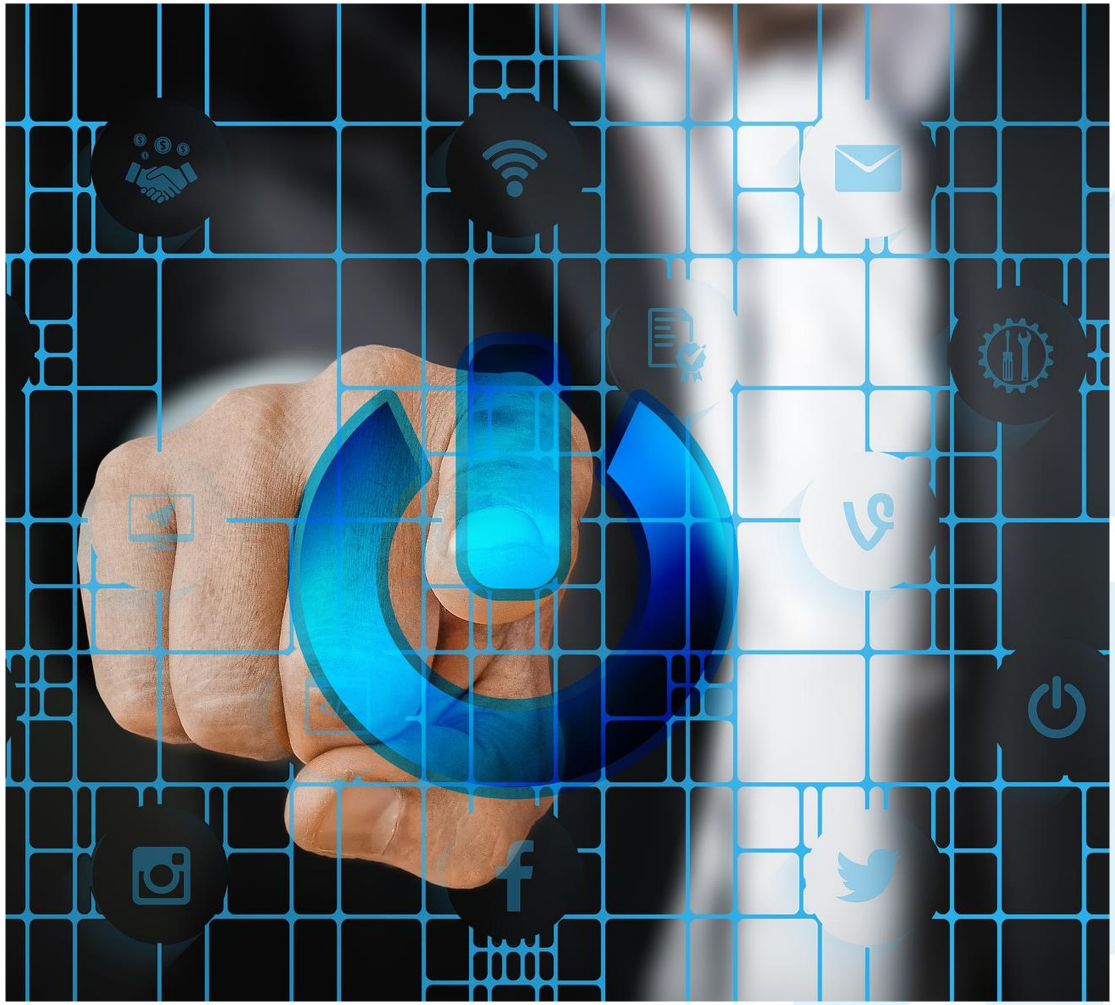
DIRECTIONAL PLATFORM CONSIDERATIONS

- Will you remain on that (SP) platform for the anticipated life of the solution?
  - What is the expected and *realistic* life expectancy of the solution?
  - Will your solution need to migrate or transform (e.g., from on-prem to the cloud)?
  - Will there be a platform upgrade (i.e., a SharePoint upgrade) during the lifetime of the solution?
  - Solution "upgrades" are never truly practical with SharePoint ...

bitstream FOUNDRY

DIRECTIONAL PLATFORM CONSIDERATIONS

- What (and where) are the non-SP interfaces for systems that will be part of the solution architecture?
  - You may want to put your solution in SharePoint, but that doesn't mean that all of the data your solution processes resides in SharePoint.
  - Some systems can't be reached from the cloud or on-premises: web services, legacy systems, others.
  - Hybrid connectors are available to bridge the on-prem/cloud gap in some cases (e.g., PowerAutomate).



TECHNICAL CONSIDERATIONS

- What sort of user interface / user experience (UI/UX) is needed?
  - Are there specific capabilities that must be present in the UI?
  - Do you need to meet a particular WCAG standard? (hint: the answer is probably “yes”).
  - Does your application need to do any heavy/CPU-intensive work?
  - What is your ability to envision your solution as a web application?
  - Are your UX/UI requirements clear?
  - Does your solution even *require* a user interface?

TECHNICAL CONSIDERATIONS

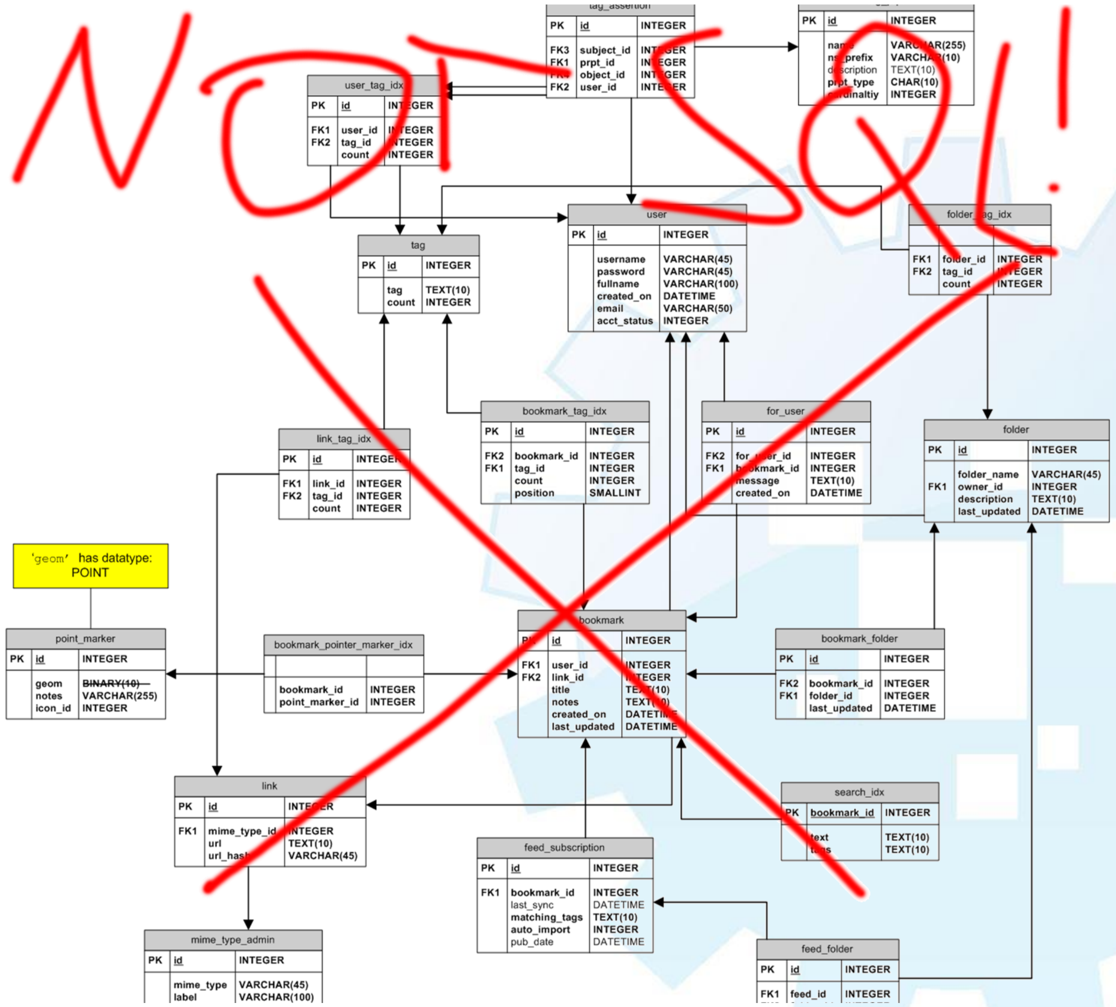
- Does your application need to be able act as someone other than the current user?
    - Impersonation or an ability to act in an app/administrative capacity is provided with some solution approaches but absent in others.
    - Most browser-based applications and those that run client-side cannot impersonate another user.
    - Understand your application use cases before you decide!

TECHNICAL CONSIDERATIONS

- Does your solution need to store or persist data in some way?
  - Most applications have this requirement in some form.
  - What is the nature of the data to be persisted? Complex objects? Scalar values?
  - Can the data be stored (or structured for storage) in SharePoint?
  - Do you have a need for elevated or even "extreme" data security?
  - Do you have GDPR and/or data sovereignty requirements?

TECHNICAL CONSIDERATIONS

- Does your application have a need to work with relational data?
  - Does your data need referential integrity, triggers, constraints, very specific data type definitions, etc?
  - SharePoint is **NOT an RDMS** and has **NEVER BEEN ONE**!
  - This doesn't mean that SharePoint data is unorganized, but you won't achieve third normal form with it.
  - SharePoint has (lightweight) Lookup Columns. Don't mistake these for true RDMS relational capabilities.



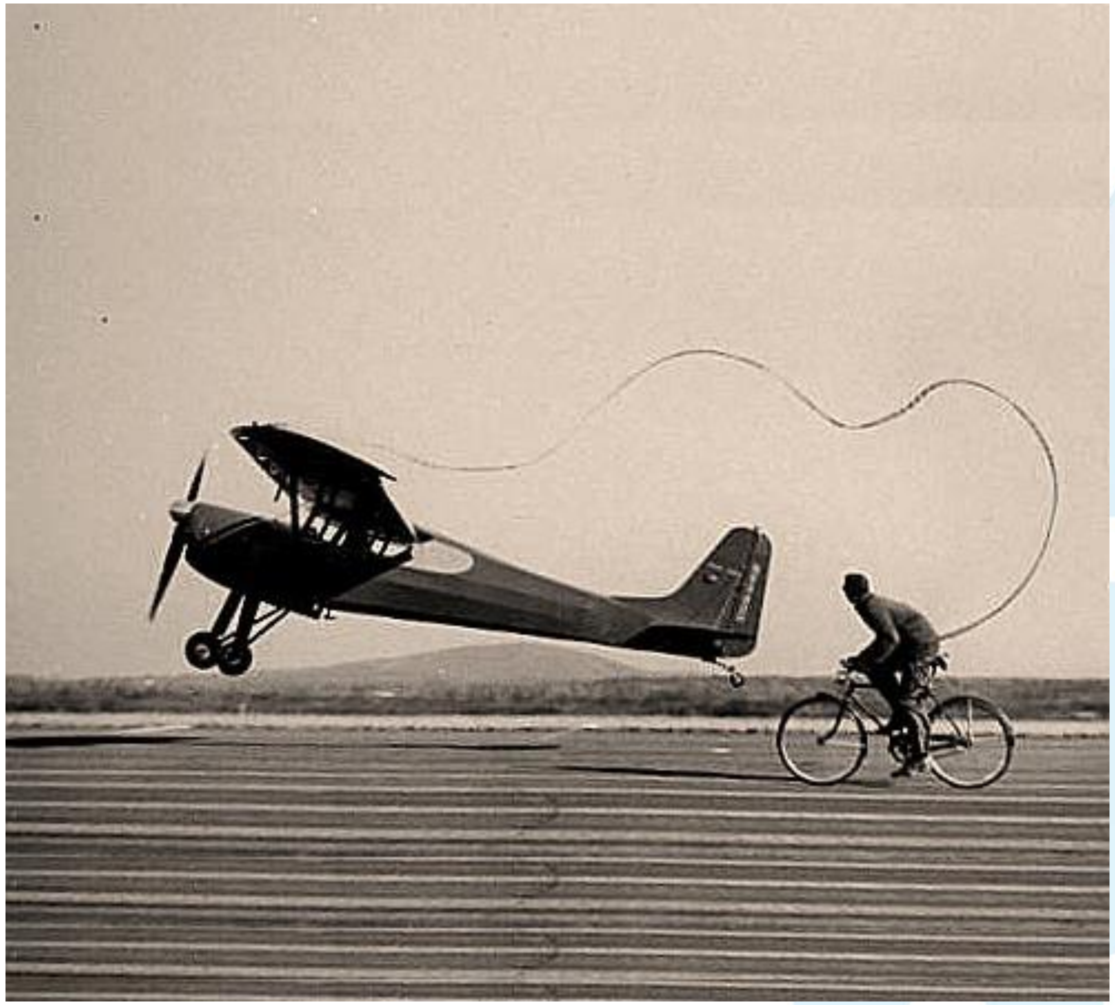
OTHER RELEVANT CONSIDERATIONS

- I've been over some of the common considerations that I've seen and faced over the years.
- The landscape is changing under our feet constantly – truly.
- By the time you're ready to design your solution, there are likely to be additional questions and criteria.
- Did I miss something you would consider especially significant in your "solutioning" process?

SOLUTION DESIGN OPTIONS

THERE'S NO SINGLE MAGIC ANSWER

- As with everything SharePoint, "it depends."
  - Each of the items and concerns we've discussed thus far factor into the ultimate approach taken.
  - Hard-blockers and "absolutely not" circumstances tend to be rare.
- The thoughts/suggestions I'm about to share should not be thought of in absolute terms.
  - It's always possible to find situations which contraindicate an approach!



FULL-TRUST SOLUTIONS

- With few exceptions, the days of full-trust, compiled solutions and SharePoint Features that run in the SharePoint farm are behind us.
  - Full-trust solutions are not permitted in SPO and most "clouds."
  - Full-trust solutions are not required to impersonate users or execute code in the context of an application-only identity.
  - Full-trust solutions originally developed a bad name due to developers writing code that had (unintentional) negative effects on SharePoint farms.
- Are there cases when I might not want to rule-out full-trust solutions?
  - If you *know* you will not be going to the cloud in the application's lifetime.
  - If you're adapting an existing full-trust solution for on-premises use (see 1st bullet)
  - If the development team is struggling with the "tech curve" (don't laugh!)
  - If you are truly building something that can only be built with full-trust code.

FULL-TRUST ALTERNATIVES

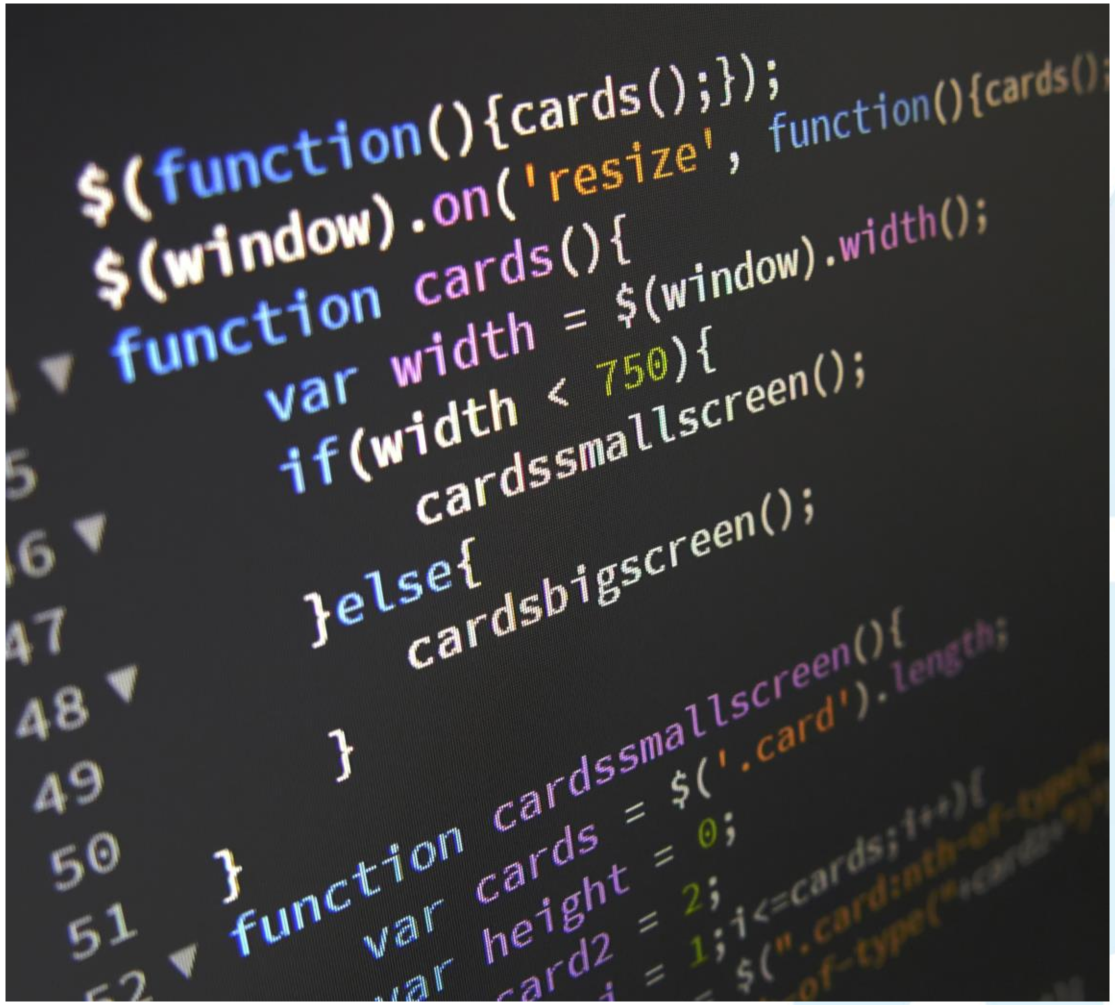
- Microsoft typically doesn’t deprecate technologies without some viable alternative approach being available.
  - You won’t get pushed out of the plane without a (functioning) parachute.
- Consider Provider-Hosted Add-Ins in conventional full-trust scenarios.
  - PHAs treat SharePoint as a web service.
  - PHAs run on a separate app server.
  - PHAs can be built in any language.
  - PHAs can operate non-interactively.
  - PHAs can impersonate!

THE CLIENT-SIDE PUSH

- Microsoft makes no secret of it.
  - Development and customization is best done somewhere other than in/on the SharePoint farm.
  - Since 2013, the expansion of dev capabilities has pushed further into browser and client-side territory.
- Exciting (and maybe a little scary to some) options exist!
  - Unfamiliar to many older SP devs.
  - The rest of this session will be spent on the client-side of the equation.

CLIENT-SIDE OPTIONS FOR DEVELOPMENT

- Typically focus on JavaScript
  - JS-based development has been possible since 2007 in some form.
  - Newer REST-based web services.
  - Client-Side Object Model (CSOM).
  - SPFx and other frameworks (React).
- Going to JS-based solutions allowed Microsoft to modernize antiquated SP dev approaches
  - Modern dev stack more common.
  - No longer need special tools.*



```
$(function(){cards();});
$(window).on('resize', function(){cards()
function cards(){
    var width = $(window).width();
    if(width < 750){
        cardssmallscreen();
    }else{
        cardsbigscreen();
    }
}
function cardssmallscreen(){
    var cards = $('.card').length;
    var height = 0;
    var card2 = 2;
```

CLIENT-SIDE OPTIONS FOR DEVELOPMENT

- Hate JavaScript? It's not your only option, you know …
  - Older SharePoint developers tend to fare better with C# and ASP.NET.
  - In addition to JavaScript, CSOM libraries exist for .NET.
  - REST web services are platform-agnostic and callable with a browser.
  - Yes, this means you can use Java, PHP, or any connected platform with SP.
- JavaScript snippets, jQuery, and "simpler" JavaScript approaches
  - They still exist and they still work … and they're still not very ALM friendly.
  - If you're unfamiliar with modern web stacks/tooling*, use what works for you.
  - If you're comfortable with modern tooling and web frameworks, try SPFx.
  - SPFx allows you to use React, other JS frameworks, and TypeScript.
  - Building admin tools and/or scripting components? Don't forget PowerShell!

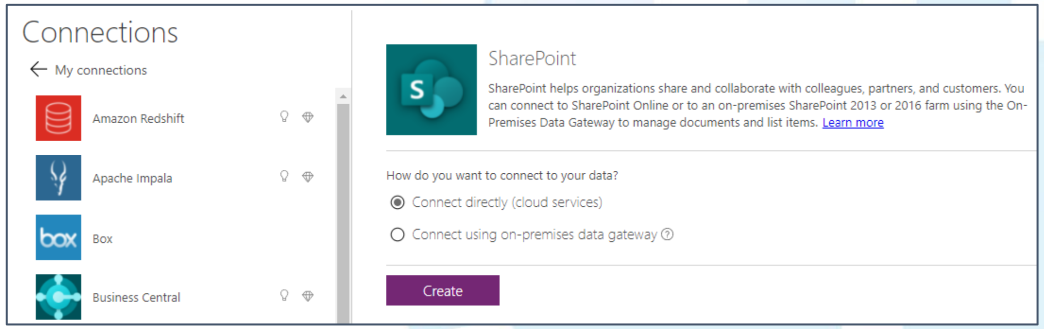
POWER-USER DEVELOPMENT OPTIONS

- Microsoft has been lowering the bar and cost of entry for building solutions and solving problems in M365.
  - (Formerly) known as "power-users," everyone is now considered a "citizen developer."
  - The goal is to remove or minimize the amount of technical knowledge needed to solve business problems.
  - This has bred a new series of tools and helper applications that are accessible to just about everyone.

POWER-USER DEVELOPMENT OPTIONS

- Meet the (old) Microsoft Power Platform:
  - Use Power BI for data visualization and data dashboards.
  - Employ Power Apps to rapidly create user interfaces that can be paired with business logic and connectors.
  - Power Automate excels at orchestrating (long-running) business processes non-interactively.
  - Power Virtual Agents can be used to build AI-powered bots with minimal technical knowledge required.

Power BI Power Apps Power Automate Power Virtual Agents


Connections

← My connections

Amazon Redshift

Apache Impala

Box

Business Central

SharePoint

SharePoint helps organizations share and collaborate with colleagues, partners, and customers. You can connect to SharePoint Online or to an on-premises SharePoint 2013 or 2016 farm using the On-Premises Data Gateway to manage documents and list items. Learn more

How do you want to connect to your data?

[✓ Connect directly (cloud services)]

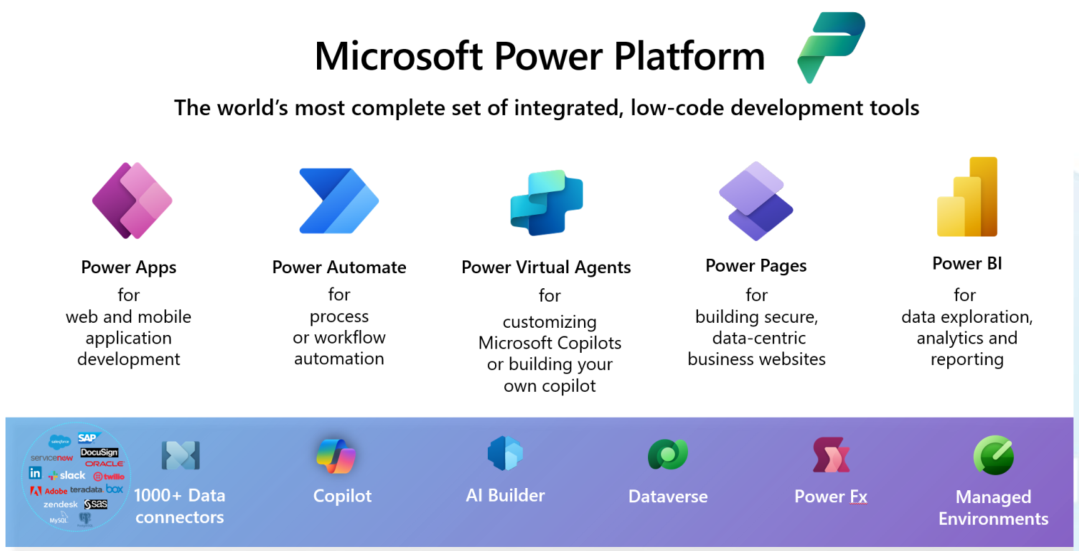
Connect using on-premises data gateway

Create

bitstream FOUNDRY

POWER-USER DEVELOPMENT OPTIONS

- Meet the (current) Microsoft Power Platform:
  - Tools of the previous Power Platform +
  - Power Pages
  - AI Builder
  - Dataverse
  - Power Fx
  - … and more.
- The current favorite flavor is, of course, Copilot (in its various forms).

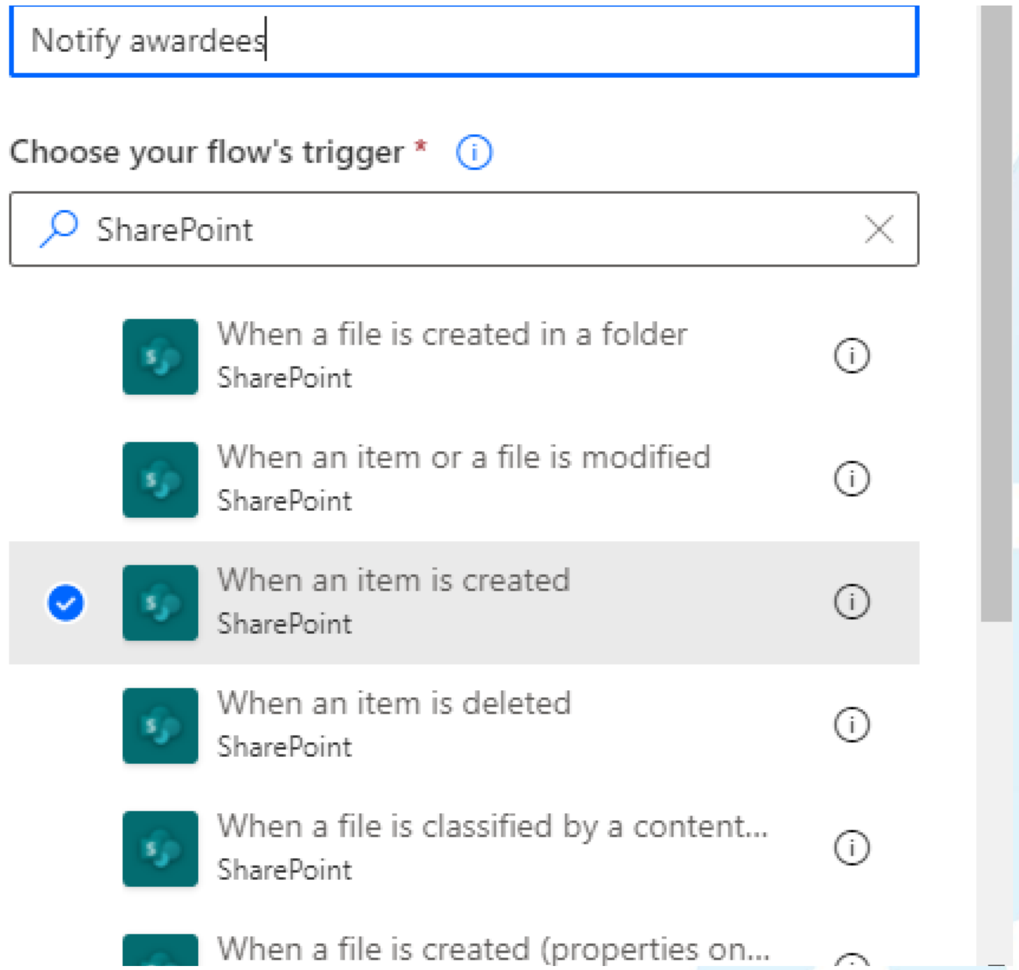

Microsoft Power Platform

The world's most complete set of integrated, low-code development tools

Power Apps	Power Automate	Power Virtual Agents	Power Pages	Power BI
for web and mobile application development	for process or workflow automation	for customizing Microsoft Copilots or building your own copilot	for building secure, data-centric business websites	for data exploration, analytics and reporting

POWER-USER DEVELOPMENT OPTIONS

- Power Automate is of particular interest to us when building SharePoint-centric solutions.
  - SharePoint 2010 Workflows have been disabled for use in SPO.
  - SharePoint 2013 Workflows will suffer the same fate in the not-so-distant future (you have until 4/2/2026).
  - Power Automate is the "named successor" for workflow needs.
  - PowerAutomate permits the creation of the same style of business solution as SharePoint Workflow


Notify awardees

Choose your flow's trigger *

SharePoint

When a file is created in a folder
SharePoint

When an item or a file is modified
SharePoint

✓ When an item is created
SharePoint

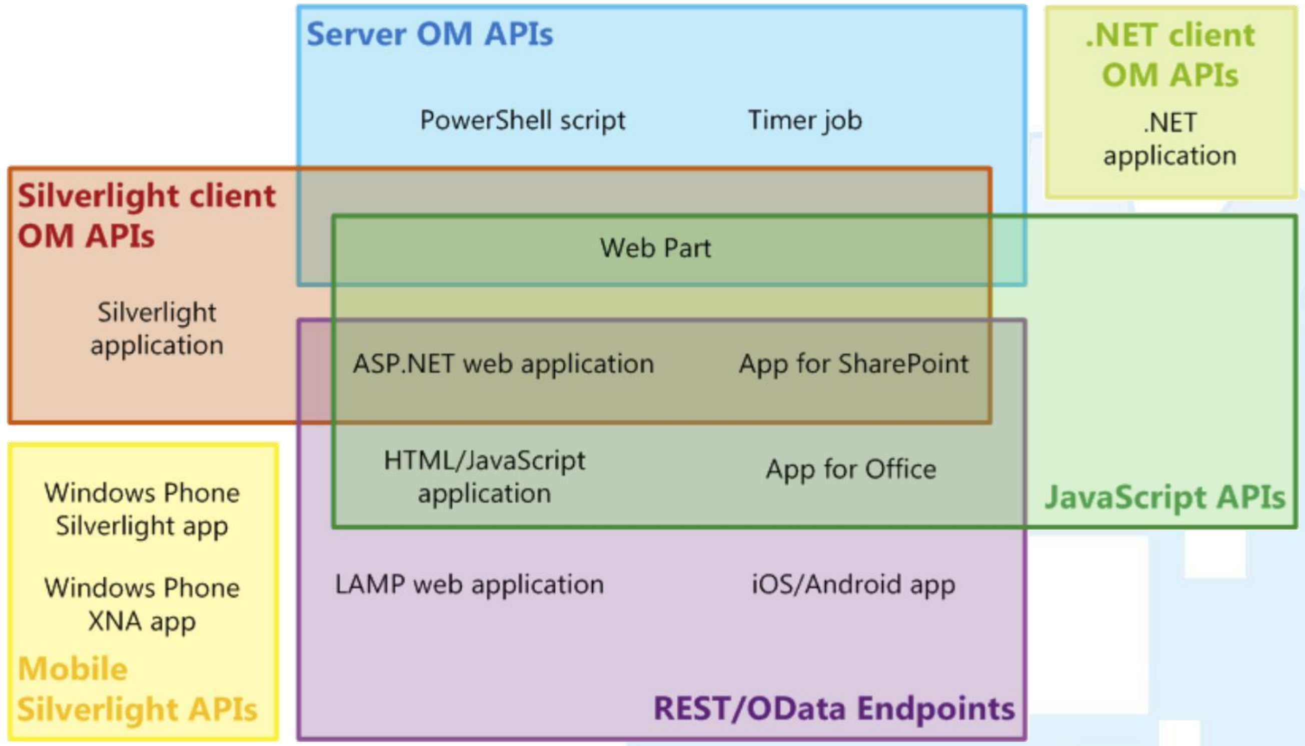
When an item is deleted
SharePoint

When a file is classified by a content...
SharePoint

When a file is created (properties on...

OTHER OPTIONS

- No two business solutions are the same or have the same requirements/goals.
- There are countless approaches and technologies from which to build a SharePoint-based solution.
- Microsoft is constantly rolling out new technologies, platforms, web services, and tools to build solutions.



AVOIDING THE CORNER ...

DON'T PAINT YOURSELF INTO A CORNER

- Avoid the situation with some honest and realistic planning
  - How long will the solution likely stay in operation?
  - Are you due to migrate or shift from on-prem to the cloud?
  - How can solution dependencies be minimized?
  - Can decisions be made to reduce potential maintenance and ongoing changes over time?
  - Can low(er) tech approaches work?
  - Can you avoid re-inventing the wheel with 3rd party apps/tools?

DON'T PAINT YOURSELF INTO A CORNER

- All things being equal, if you have the ability to decide …
  - Choose a client-side strategy over a server-side strategy.
  - Opt for open web-standards and web services (e.g. REST-based).
  - Microsoft is pushing the JavaScript development stack, so SPFx and associated technologies (like React) will probably have a longer life.
  - Follow Microsoft development guidance and suggestions.

DON'T PAINT YOURSELF INTO A CORNER

- Here are some technologies that, as a rule, you shouldn't use to build solutions or include in your solutions – even if they appear "okay" at first glance:
  - SharePoint 2010 Workflows.
  - SharePoint 2013 Workflows.
  - Silverlight Apps.
  - InfoPath Forms and Solutions.
  - SharePoint Designer (any version).
- When in doubt, check support!

bitstream FOUNDRY

SUMMARY

- There is no one-size-fits-all solution approach
- A technology choice in one scenario may be contraindicated in another
- A little extra effort seeking help and guidance from someone who has been building SharePoint solutions for a long time is worth the time and investment.

Final Questions?

REFERENCES AND RESOURCES

SharePoint

- https://en.wikipedia.org/wiki/SharePoint

History of SharePoint: Then and Now

- https://www.eswcompany.com/history-sharepoint-future/

(Archives) Microsoft FrontPage 2003: Getting Started with FrontPage

- https://www.uwec.edu/kb/article/microsoft-frontpage-2003-getting-started-with-frontpage/

Orca.exe

- https://docs.microsoft.com/en-us/windows/win32/msi/orca-exe

REFERENCES AND RESOURCES (CONTINUED)

SharePoint Designer 2007

- https://www.kean.edu/~ocisweb/downloads/SharepointDesigner_2007/

Index of SharePoint Web Services

- https://blog.crsw.com/sharepoint/index-of-sharepoint-web-services/

Windows SharePoint Services 3.0

- https://docs.microsoft.com/en-us/previous-versions/office/developer/sharepoint-services/bb931737(v=office.12)

Getting Started with SharePoint 2010 Developer Tools in Visual Studio 2010

- https://docs.microsoft.com/en-us/previous-versions/office/developer/sharepoint-2010/gg131919(v=office.14)

REFERENCES AND RESOURCES (CONTINUED)

MOSS 2007 Web Services – Easier Way to Populate the SharePoint Information

- http://jenkinsblogs.com/2009/01/19/jenkins-blog-moss-2007-web-services-easier-way-to-populate-the-sharepoint-information/

SPServices

- https://sympmarc.github.io/SPServices/index.html

WCF Services in SharePoint Foundation 2010

- https://docs.microsoft.com/en-us/previous-versions/office/developer/sharepoint-2010/ff521586(v=office.14)

Welcome to the Microsoft SharePoint 2010 SDK

- https://docs.microsoft.com/en-us/previous-versions/office/developer/sharepoint-2010/ee557253(v=office.14)

REFERENCES AND RESOURCES (CONTINUED)

Install and Configure Workflow for SharePoint Server

- https://docs.microsoft.com/en-us/sharepoint/governance/install-and-configure-workflow-for-sharepoint-server

SharePoint Development Documentation

- https://docs.microsoft.com/en-us/sharepoint/dev/

Database Normalization (Explained in Simple English)

- https://www.essentialsql.com/get-ready-to-learn-sql-database-normalization-explained-in-simple-english/

SQL - Contraints

- https://www.tutorialspoint.com/sql/sql-constraints.htm

REFERENCES AND RESOURCES (CONTINUED)

Data Sovereignty

- https://en.wikipedia.org/wiki/Data_sovereignty

Transforming Your SharePoint Full Trust Code to the Office App Model

- https://channel9.msdn.com/Events/Ignite/2015/BRK4125

Office Add-ins platform overview

- https://docs.microsoft.com/en-us/office/dev/add-ins/overview/office-add-ins

Get started creating provider-hosted SharePoint Add-ins

- https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/get-started-creating-provider-hosted-sharepoint-add-ins

bitstream FOUNDRY

REFERENCES AND RESOURCES (CONTINUED)

Upgrading Your Skillset for SharePoint 2013 Add-In Development

- https://sharepointinterface.com/wp-content/uploads/2015/10/upgrading-your-skillset-for-sharepoint-2013-add-in-development-sps-twin-cities-fall-2015.pdf

Get to know the SharePoint REST service

- https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/get-to-know-the-sharepoint-rest-service?tabs=csom

Complete basic operations using SharePoint client library code

- https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/complete-basic-operations-using-sharepoint-client-library-code

REFERENCES AND RESOURCES (CONTINUED)

Build your first SharePoint client-side web part (Hello World part 1)

- https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/build-a-hello-world-web-part

Choose the right API set in SharePoint

- https://docs.microsoft.com/en-us/sharepoint/dev/general-development/choose-the-right-api-set-in-sharepoint

How It Feels to Learn JavaScript in 2016

- https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f

REFERENCES AND RESOURCES (CONTINUED)

Microsoft Power Platform

- https://www.microsoft.com/en-us/power-platform

Getting Started with Power Automate

- https://docs.microsoft.com/en-us/power-automate/getting-started

SharePoint 2010 Workflow Retirement

- https://support.microsoft.com/en-us/office/sharepoint-2010-workflow-retirement-1ca3fff8-9985-410a-85aa-8120f626965f

Seriously, It's Time, Just Say "No" to InfoPath

- https://www.markrackley.net/2019/08/11/seriously-its-time-just-say-no-to-infopath/

REFERENCES AND RESOURCES (CONTINUED)

Silverlight End Of Support

- https://support.microsoft.com/en-us/windows/silverlight-end-of-support-0a3be3c7-bead-e203-2dfd-74f0a64f1788

Lifecycle: SharePoint Designer 2013

- https://docs.microsoft.com/en-us/lifecycle/products/sharepoint-designer-2013

Search Products and Services Lifecycle Information

- https://docs.microsoft.com/en-us/lifecycle/products/

Web Content Accessibility Guidelines (WCAG)

- https://docs.microsoft.com/en-us/compliance/regulatory/offering-wcag-2-1

REFERENCES AND RESOURCES (CONTINUED)

Satya Nadella: Mobile First, Cloud First Press Briefing

- https://news.microsoft.com/2014/03/27/satya-nadella-mobile-first-cloud-first-press-briefing/

What is an on-premises data gateway?

- https://docs.microsoft.com/en-us/power-automate/gateway-reference

How to write a cmdlet

- https://docs.microsoft.com/en-us/powershell/scripting/developer/cmdlet/how-to-write-a-simple-cmdlet

Microsoft TEALS Program

- https://www.microsoft.com/en-us/teals

REFERENCES AND RESOURCES (CONTINUED)

Akumina

- https://www.akumina.com/

Sean P. McDonough

sean@SharePointInterface.com

sean@BitstreamFoundry.com

Company: bitstream FOUNDRY and akumina®

LinkedIn: https://www.linkedin.com/in/smcdonough/

Twitter: @spmcdonough

Blog: https://SharePointInterface.com/

About: https://spmcdonough.com/

bitstream FOUNDRY