

Elio Struyf

Solutions Architect @ Valo Intranet

@eliostruyf

Things I've learned while building a product on SharePoint Modern



SharePoint and Office 365 Saturday Munich

30 November 2019 | Microsoft Munich



SharePoint and Office 365

Saturday Munich

GOLD



PLATINUM



Quest®



SILVER



HANSEVISION
the redmond integrators
Ein Unternehmen der Bechtle Gruppe



Building a product is awesome!





Old days of development



A vibrant, ornate carousel horse is the central focus. It has a white body with orange and yellow mane and tail decorations. A blue harness with gold trim is visible. The horse is mounted on a platform with red and black stripes. In the background, other colorful horses are visible, and a sign on the platform reads "BRIGHTON PIER".

Updates came in slowly

A photograph of a roller coaster track against a clear blue sky. The track is yellow and blue, curving upwards. A sign with the word "TURBO" in large, illuminated letters is mounted above the track. The letters are white with a blue outline.

TURBO

Speed is now our enemy





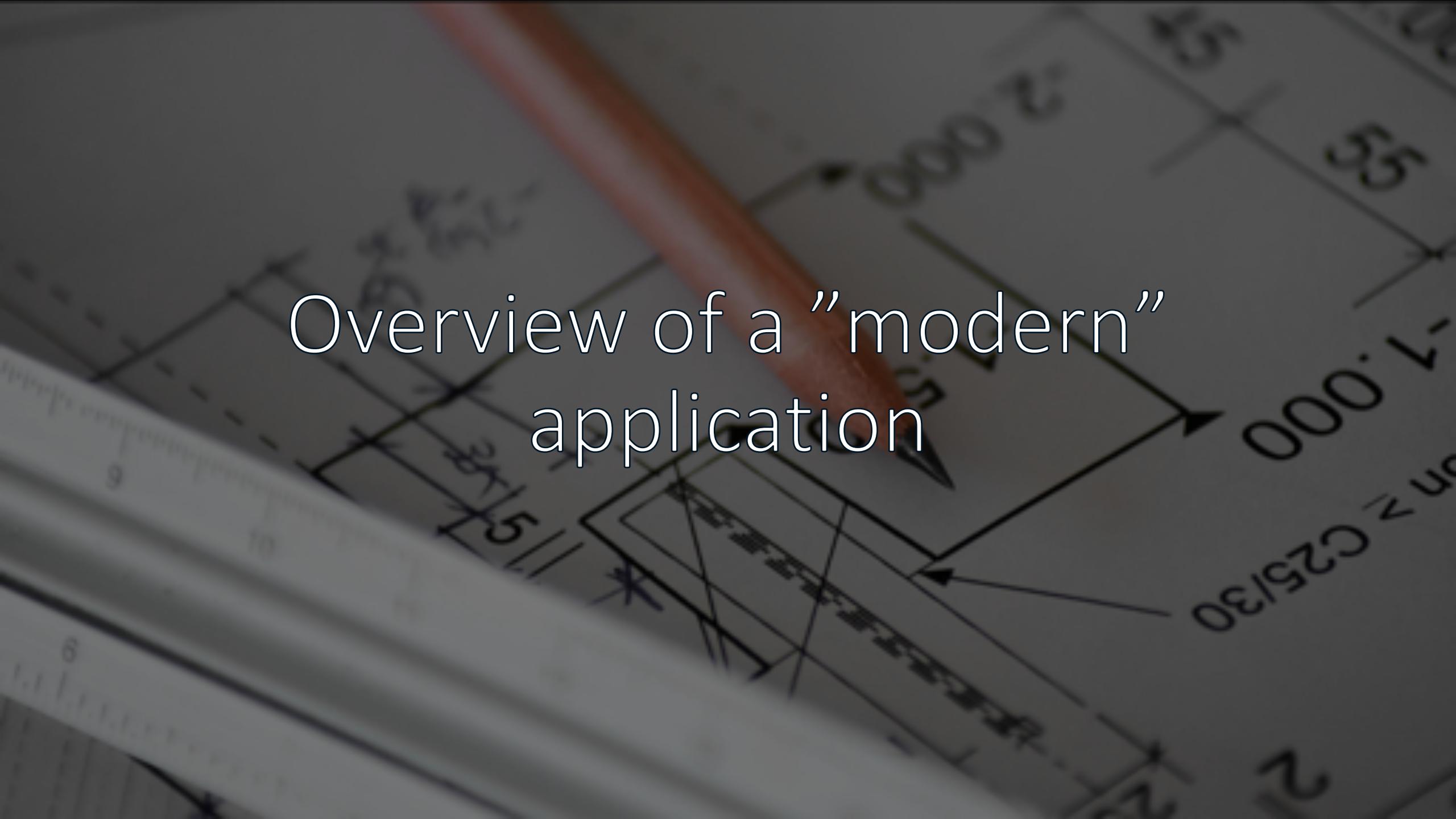
Be prepared!



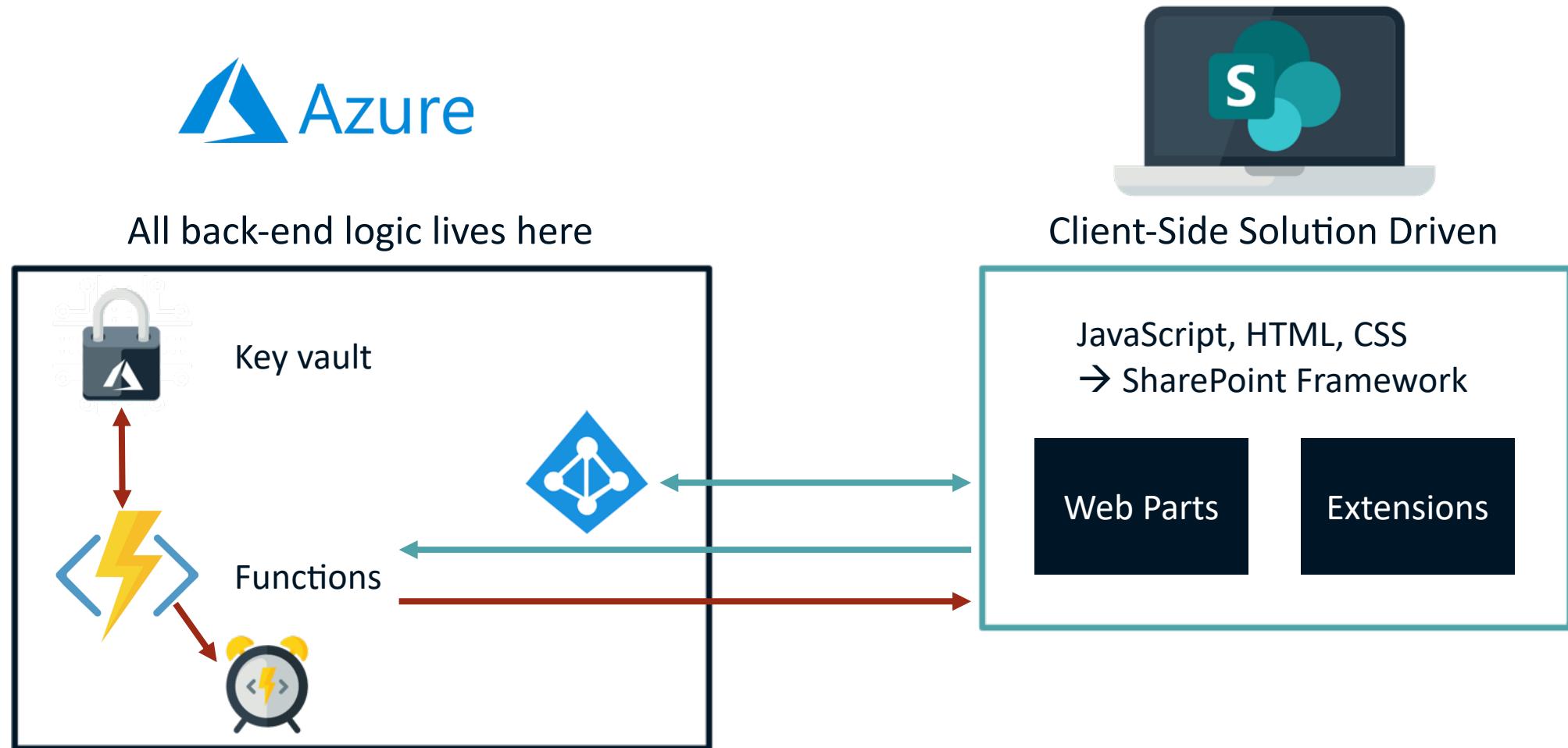
A close-up photograph of a brass sextant mounted on a wooden railing. The sextant is polished and reflects the surrounding environment. In the background, a blurred cityscape with buildings and possibly the Eiffel Tower is visible under a cloudy sky.

There is a lot to keep an eye
on

Overview of a “modern” application



Architecture overview



A photograph of a wooden desk with educational items. In the foreground, there are four colored pencils (yellow, green, red, blue) scattered on the desk. To the left, there is a stack of four books of varying thicknesses. On top of the books sits a single red apple. To the right, there is a stack of three wooden blocks. The top block is red and has the letter 'A' printed on it. The middle block is orange and has the letter 'B' printed on it. The bottom block is yellow and has the letter 'C' printed on it.

Keep yourself up to date

Channels to follow

- PnP Community Calls: <http://aka.ms/sppnp>
- <https://www.microsoft.com/en-us/microsoft-365/roadmap>
- <https://developer.microsoft.com/en-us/office/blogs>
- <https://techcommunity.microsoft.com/t5/SharePoint/ct-p/SharePoint>
- <https://azure.microsoft.com/en-in/updates/>



A close-up photograph of a person's hand gripping a power drill. The hand is wearing a dark long-sleeved shirt. The background is dark and filled with numerous small, glowing orange and yellow circular particles, resembling bokeh or sparks. The overall composition suggests a theme of craftsmanship or industrial work.

Let us talk about building
things

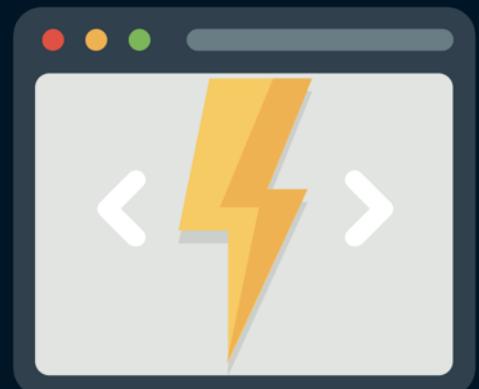
SPFx runs in the context of the user





Azure Functions tips

- ! Don't forget to secure them with Azure AD
- ! You'll be using app-only permissions
 - Check the SharePoint permissions on both ends
- ! Try to create a messaging system
 - Users will not wait > 10 seconds
 - Queues great for this in combination with Queue Triggered function
 - Create a HTTP Message Broker → creates the queue message





Navigation

Crazy how nature does that



Important checks while navigating

- ✓ Check page URL
- ✓ Check list ID + item ID
- ✓ Check the HubSite ID
- ✓ Check the current UI language
- ✓ Check if you are creating a new page

Depending on the type of change:

- Application customizer needs a reload
- Page needs to be refreshed
- Component updates
- Aborting fetch calls

je t'aim

Languages / Localization

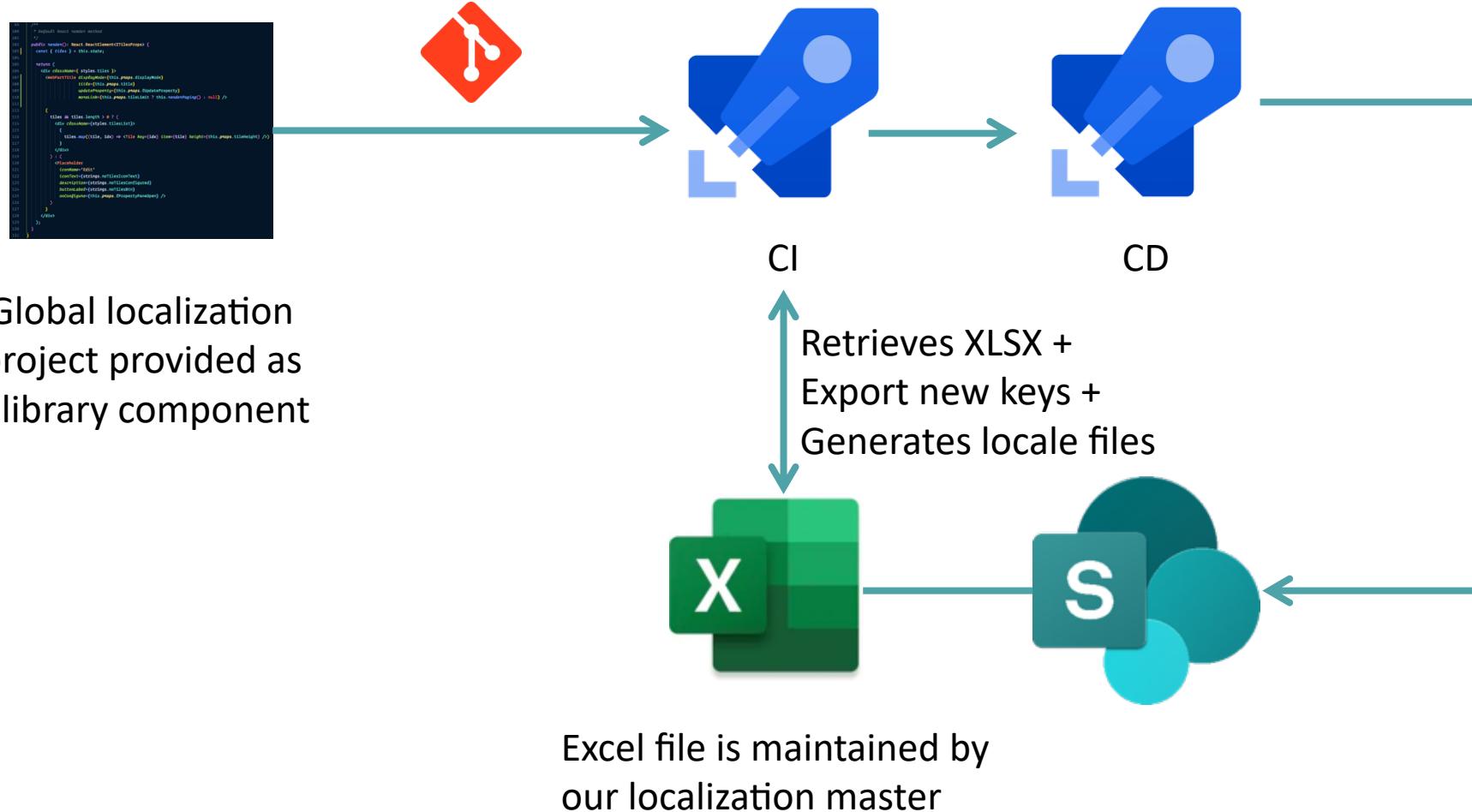
Standard SPFx localization story

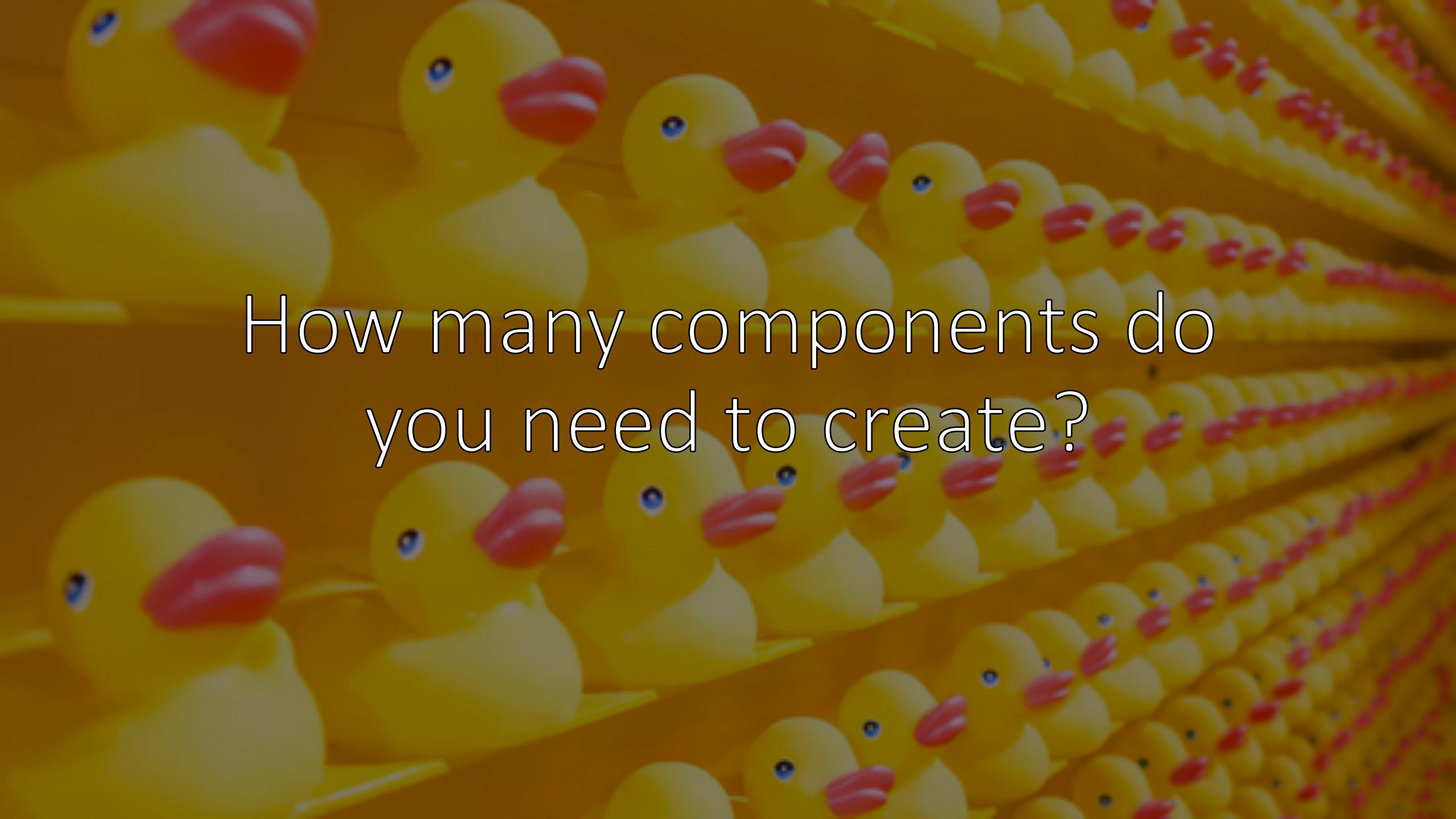
- Each web part / extension has its own localization file
 - `/src/webparts/<your-webpart>/loc/<locale>.js`
- Developers **must** maintain these files
 - “Easy” for 1-3 languages → updates need to happen in each file
 - Harder when you need to support +20 languages
- It is JS, things can go wrong!

SPFx localization solutions

- Create the localization files as TS files
→ Catch issues during build
- One global localization file
 - ✓ Easy to maintain
 - ✓ Sharing localization keys is easier
 - ! Developer must still maintain it
- Use an extension - <https://elst.es/vscode-localization>
 - ✓ Automatically add entries in all locale files
 - ✓ Generate a CSV file that can be owned by someone else
- Create your own system

How we do localization



A large number of yellow rubber ducks with red beaks and blue eyes are swimming in water. They are arranged in several rows, creating a sense of depth. The ducks are facing towards the right side of the frame.

How many components do
you need to create?



It can become very slow!

Build target: DEBUG

```
[09:49:39] Using gulpfile ~/nodejs/sessions/spn119/gulpfile.js
[09:49:39] Starting gulp
[09:49:39] Starting 'bundle'...
[09:49:39] Starting subtask 'configure-sp-build-rig'...
[09:49:39] Finished subtask 'configure-sp-build-rig' after 3.63 ms
[09:49:39] Starting subtask 'pre-copy'...
[09:49:39] Finished subtask 'pre-copy' after 13 ms
[09:49:39] Starting subtask 'copy-static-assets'...
[09:49:39] Starting subtask 'sass'...
[09:49:39] Finished subtask 'copy-static-assets' after 28 ms
[09:49:40] Finished subtask 'sass' after 127 ms
[09:49:40] Starting subtask 'tslint'...
[09:49:40] [tslint] tslint version: 5.12.1
[09:49:40] Starting subtask 'tsc'...
[09:49:40] [tsc] typescript version: 2.9.2
[09:49:42] Finished subtask 'tsc' after 1.47 s
[09:49:42] Finished subtask 'tslint' after 2.22 s
[09:49:42] Starting subtask 'post-copy'...
[09:49:42] Finished subtask 'post-copy' after 88 µs
[09:49:42] Starting subtask 'collectLocalizedResources'...
[09:49:42] Finished subtask 'collectLocalizedResources' after 2.32 ms
[09:49:42] Starting subtask 'configure-webpack'...
[09:49:42] Finished subtask 'configure-webpack' after 267 ms
[09:49:42] Starting subtask 'webpack'...
[09:49:44] Finished subtask 'webpack' after 2.39 s
[09:49:44] Starting subtask 'configure-webpack-external-bundling'...
[09:49:44] Finished subtask 'configure-webpack-external-bundling' after 554 µs
[09:49:44] Starting subtask 'copy-assets'...
[09:49:44] Finished subtask 'copy-assets' after 15 ms
[09:49:44] Starting subtask 'write-manifests'...
[09:49:45] Finished subtask 'write-manifests' after 288 ms
[09:49:45] Finished 'bundle' after 5.33 s
[09:49:45] ======[ Finished ]=====
[09:49:46] Project spn119 version:8.8.1
[09:49:46] Build tools version:3.9.26
[09:49:46] Node version:v8.9.4
[09:49:46] Total duration:7.71 s
```

Build target: DEBUG

```
[09:47:55] Using gulpfile ~/nodejs/spfx/microsoft/sp-portal-showcase/solution/gulpfile.js
[09:47:55] Starting gulp
[09:47:55] Starting 'bundle'...
[09:47:55] Starting subtask 'configure-sp-build-rig'...
[09:47:56] Finished subtask 'configure-sp-build-rig' after 8.17 ms
[09:47:56] Starting subtask 'pre-copy'...
[09:47:56] Finished subtask 'pre-copy' after 22 ms
[09:47:56] Starting subtask 'copy-static-assets'...
[09:47:56] Starting subtask 'sass'...
[09:47:56] Finished subtask 'sass' after 946 ms
[09:47:56] Starting subtask 'tslint'...
[09:47:57] [tslint] tslint version: 5.12.1
[09:47:57] Starting subtask 'tsc'...
[09:47:57] [tsc] typescript version: 2.9.2
[09:47:57] Finished subtask 'copy-static-assets' after 1.77 s
[09:48:03] Finished subtask 'tslint' after 6.2 s
[09:48:03] Finished subtask 'tsc' after 6.23 s
[09:48:03] Starting subtask 'post-copy'...
[09:48:03] Finished subtask 'post-copy' after 113 µs
[09:48:03] Starting subtask 'collectLocalizedResources'...
[09:48:03] Finished subtask 'collectLocalizedResources' after 9.49 ms
[09:48:03] Starting subtask 'configure-webpack'...
[09:48:04] Finished subtask 'configure-webpack' after 415 ms
[09:48:04] Starting subtask 'webpack'...
[09:48:38] Finished subtask 'webpack' after 34 s
[09:48:38] Starting subtask 'configure-webpack-external-bundling'...
[09:48:38] Finished subtask 'configure-webpack-external-bundling' after 452 µs
[09:48:38] Starting subtask 'copy-assets'...
[09:48:40] Finished subtask 'copy-assets' after 1.53 s
[09:48:40] Starting subtask 'write-manifests'...
[09:48:40] Finished subtask 'write-manifests' after 493 ms
[09:48:40] Finished 'bundle' after 45 s
[09:48:41] ======[ Finished ]=====
[09:48:41] Project sp-starter-kit version:1.5.2
[09:48:41] Build tools version:3.9.26
[09:48:41] Node version:v8.9.4
[09:48:41] Total duration:48 s
```

Splitting could be an option

Multiple projects

- ✓ Faster build times
- ✓ Easier to debug
- ✓ Lower impact when you need to do a hotfix

- ! Watch out for code duplication
- ! Localization
- ! Packaging and installations
- ! Project management -> Check out monorepo (Lerna)



A close-up photograph of a plate of nachos. The nachos are made of tortilla chips topped with melted cheese, diced tomatoes, and other toppings. A hand is visible on the right side, holding a piece of the nacho. The background is slightly blurred.

Think about sharing common
logic

Sharing common logic

- SPFx library component
 - ✓ Option for sharing code between projects
 - ! Only one library component version can be used
 - ! Be sure to keep the version in sync
- Private npm dependencies (Azure Artifacts)
 - ✓ Each project can use its own version
 - ✓ Not SPFx specific
 - ✓ Will work for classic and on-prem projects
 - ! Could lead to larger bundles (talking about KBs)



1.0.0.0

1.2.3.4

1.5.2.1

Thank
you!

SP doesn't care

Solution versioning

- Update version in **package-solution.json**
- The version only applies to the **feature framework** and **SharePoint feature definitions**
- Code and assets will be updated as soon a new version is uploaded





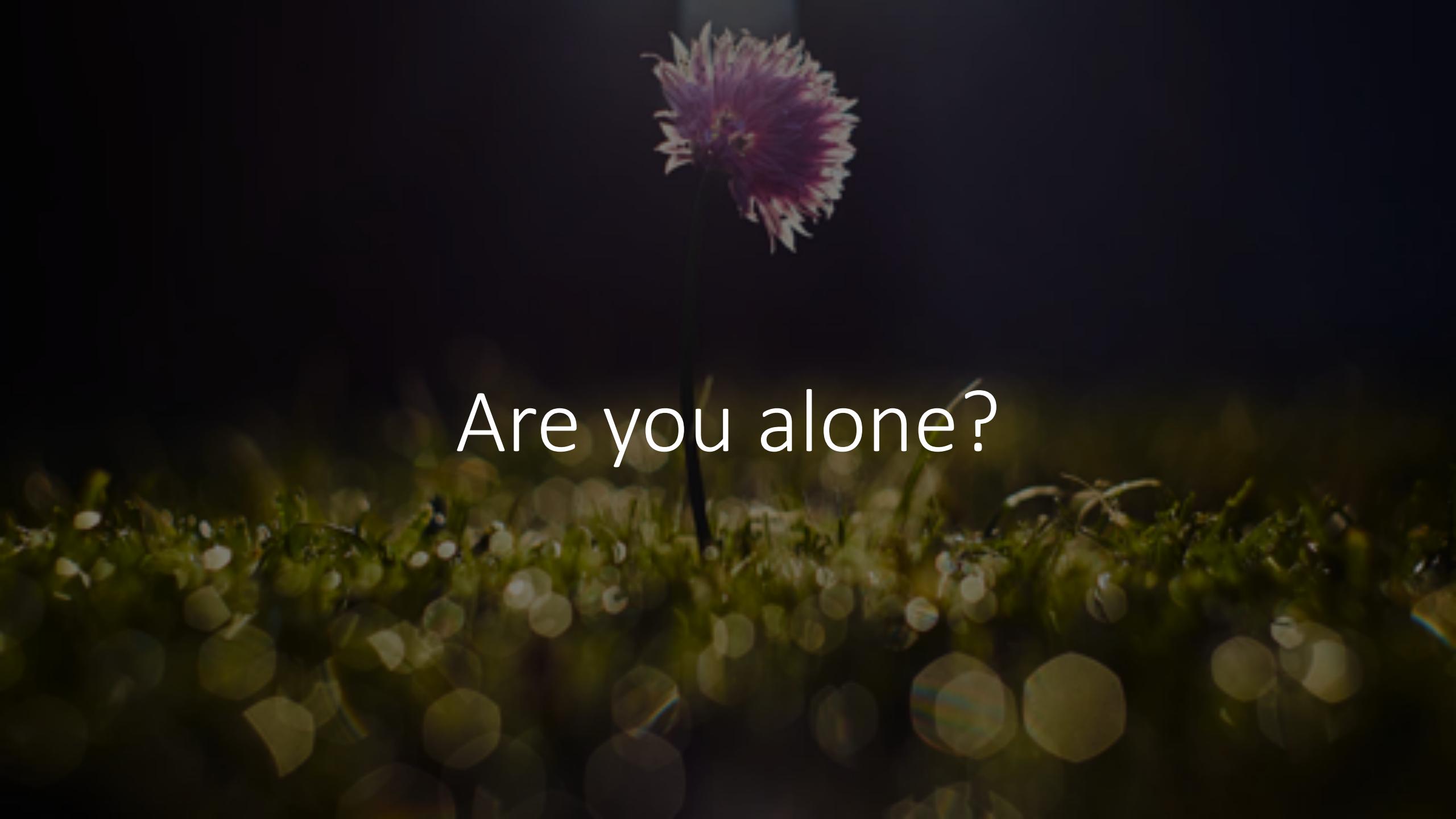
Challenges



The one that doesn't want to die

Supporting IE11

- ! Speed is the biggest issue
- ! Slow due to all the polyfills required (ex.: fetch)
- ! SPFx memory leaks (1,5GB after two minutes)
- ! Styling



Are you alone?

Are you the only third-party on the tenant?

✓ If you don't do DOM manipulations, you should be OK

! Watch out with AadHttpClient

- Shares permission scopes
- You are not in control

✓ Use isolated web parts (>=1.8.0 SPFx)

✓ Gain control by using your own Azure AD APP

- Use the AadTokenProvider





What works today,
could fail tomorrow

Be prepared for failure!

**Goggles
must be
worn**

What can you do?

- Logging!
- Create a debug mode
 - Show installed package versions
 - Log web part properties
 - Log context
 - ...



?loadSPFX=true&debugManifestsFile=
<https://localhost:4321/temp/manifests.js>

A close-up photograph of a man with dark hair and a serious expression. He is wearing a dark blue shirt. His right hand is extended towards the camera, with his index finger pointing directly at the viewer. The background is a plain, light color.

It's not always your fault



Testing is THE most
important thing

A vibrant, graffiti-covered brick wall. Key elements include: a large, stylized yellow cartoon character with a wide, toothy grin and a striped pattern on its body; a black and white skull with red eyes; a portrait of a man with a mustache and glasses; and a small figure of a person riding a bicycle. There are also various other spray-painted shapes, letters, and a small sign that says "I WANT CHANGE".

Visual UI Tests

#287: Puppeteer running on Azure DevOps

Manually run today at 10:03 by Elio Struyf → puppeteer-splx-tests [master] ← aeffel

Logs Summary Tests

Summary ↗

1 Run(s) Completed (1 Passed, 0 Failed)

3

Total tests

+3



3 ● Passed
0 ● Failed
0 ● Others

100%

Pass percentage

↑ 100%

9s 800ms

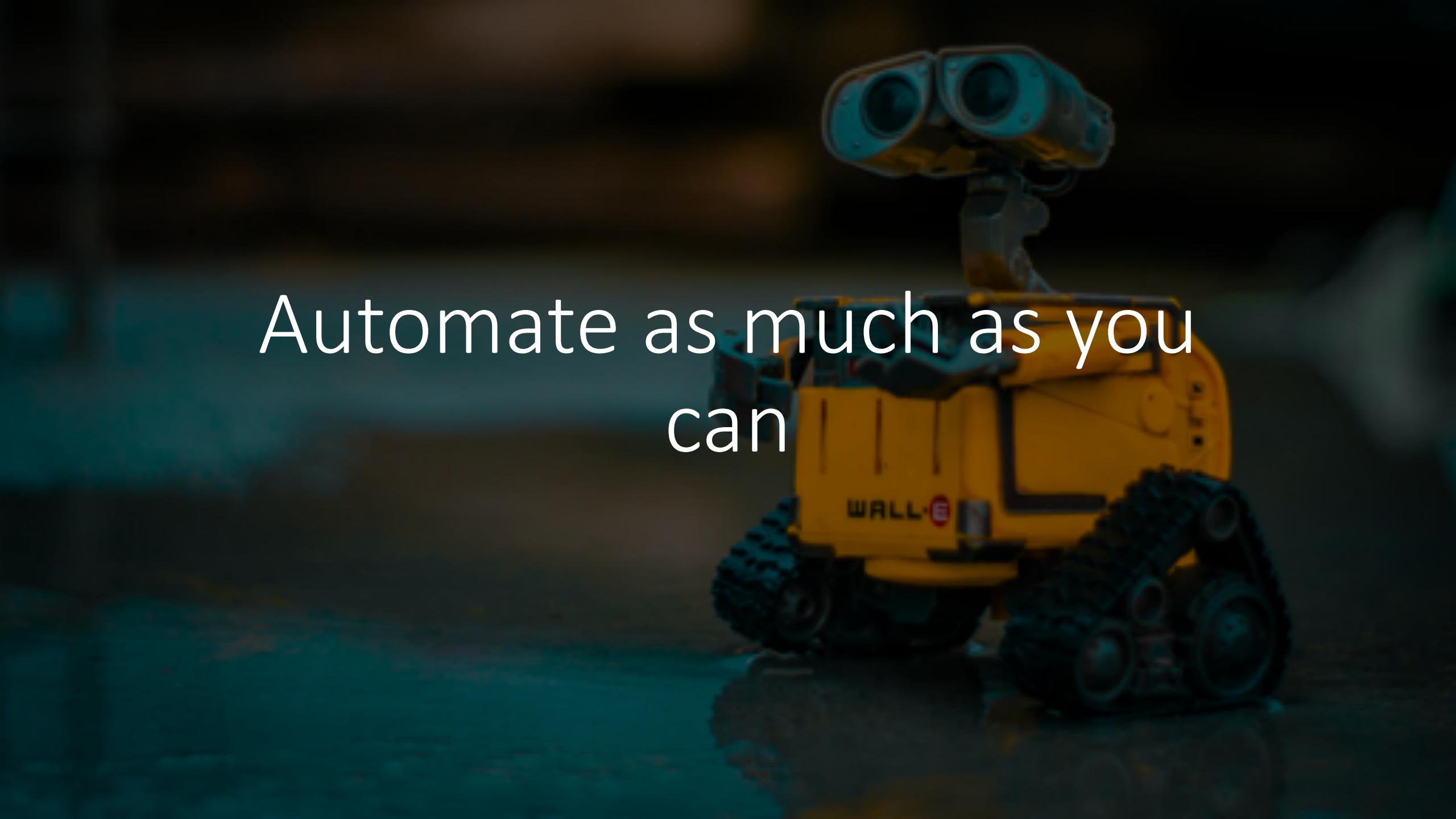
Run duration ⓘ

↑ +9s 800ms

 Bug ↗  Link

▼ Filter by test name

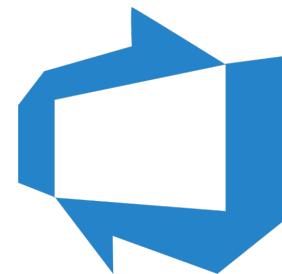
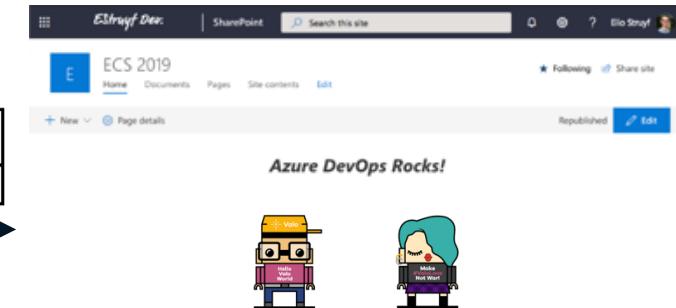
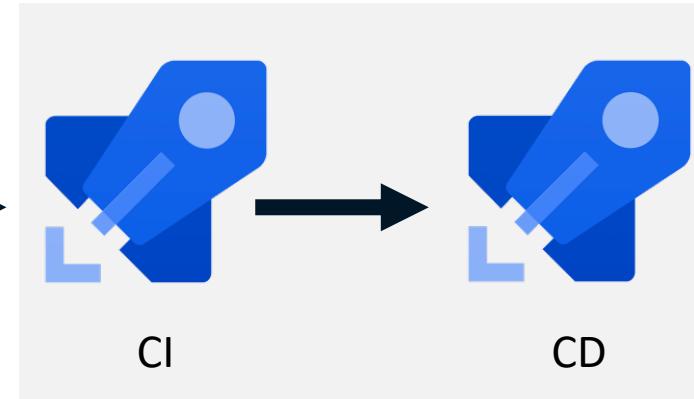
Test	Duration
+ ✓ Tests on a site page (not news) (3/3)	00:00:00.020
✓ Tests on a site page (not news) Should load the page	00:00:00.010
✓ Tests on a site page (not news) Check if caption element is present in the web part	00:00:00.007
✓ Tests on a site page (not news) Check if caption text is equal to "Automated UI Tests"	00:00:00.003

A yellow and blue robot with large, expressive eyes and a small antenna on its head stands in a dark, rocky landscape. The robot's body has the word "WALL-E" printed on it. It appears to be looking directly at the viewer.

Automate as much as you
can

Automate your releases

```
99
100 // Default React render method
101 */
102
103 public render(): React.ReactElement<ITilesProps> {
104   const { tiles } = this.state;
105
106   return (
107     <div className={ styles.tiles }>
108       <WebPartTitle displayMode={this.props.displayMode}>
109         {title}</WebPartTitle>
110         {updateProperty= {this.props.updateProperty}}
111         {moreLink= {this.props.tileLimit ? this.renderPaging() : null}} </>
112     </div>
113     {tiles.length > 0 ? (
114       <div className={ styles.tilesList }>
115         {titles.map((tile, idx) => <Tile key={idx} item={tile} height={this.props.tileHeight} />)
116       </div>
117     ) : (
118       <Placeholder
119         iconName='Edit'
120         iconText={strings.noTilesIconText}
121         description={strings.noTilesConfigured}
122         buttonLabel={strings.noTilesBtn}
123         onConfigure={this.props.onPropertyPaneOpen} />
124     )
125   )
126   </div>
127 >;
128 }
129
130
131 }
```



The full story

- We push everything immediately to our shared-DEV environment
- We do nightly provisioning
 - Flush the whole environment
 - Reinstall all components
 - Azure AD Apps
 - Azure Functions
 - SharePoint sites
 - Deploy SPPKGs

 PnP PowerShell



Things to consider

- ! Flat site structure are great, but ...
 - Lots of cross-site collection calls → calls cannot be batched
 - Have a good caching story
- ! Watch your bundle sizes
 - Don't be blinded by it
 - Use CDN – Azure CDN or Office 365 (client-side asset packaging)
 - Browser caches files → biggest impact on cold load or expired cache
- ! Test on different environments



THANK YOU!

Don't forget to
rate the session!



SharePoint and Office 365 Saturday Munich

30 November 2019 | Microsoft Munich



Elio Struyf

Solutions Architect at Valo Intranet



BIWUG



-
- @eliostruyf
 - www.eliostruyf.com
 - elio@struyfconsulting.be

Office Development MVP
Azure / DevOps / Office 365 / SharePoint

A black and white photograph of a man with glasses and a dark t-shirt with a circular pattern. He is gesturing with his hands, palms facing each other, as if explaining something. He is positioned on the left side of the frame.

Slides and related links

<https://elst.es/modern-product>