

Building React Components for SharePoint

SharePoint Engage Conference
November 2-3, 2016

Steve Pietrek



**SharePoint Application Architect
Front End Developer**

**Cardinal Solutions
App Dev Practice Manager
Raleigh/Durham**

Contact

@spietrek
spietrek@cardinalsolutions.com

Presentation References

<https://github.com/spietrek/spengage2016>
<https://github.com/spietrek/sharepoint.react.components>

Agenda

SharePoint
Development
Challenges

React
Overview

SharePoint
Framework

Office Fabric

SharePoint Development Challenges

SharePoint 2007 Challenge

- Full Trust applications
- Code affects the whole farm
- Developers build custom solutions
- Administrators can only secure solutions with Code Access Security (CAS)
 - Hard to control what is in custom code (e.g. Elevated Privileges)
- Biggest cause of SharePoint support cases: Custom code

Developer

- Design, build, and test customizations

Administrator

- Install and monitor customizations

Site Collection Owner

- Activate and use customizations

SharePoint 2010 Challenge

- Sandbox solutions applications
- Code affects the site collection
- Developers build custom solutions
- Administrators define quotas and use resource monitors to check site collection usage
- Site collection owners deploy, activate, and implement the customizations
- Missing functionality in Sandbox Solutions

Developer

- Design, build, and test customizations

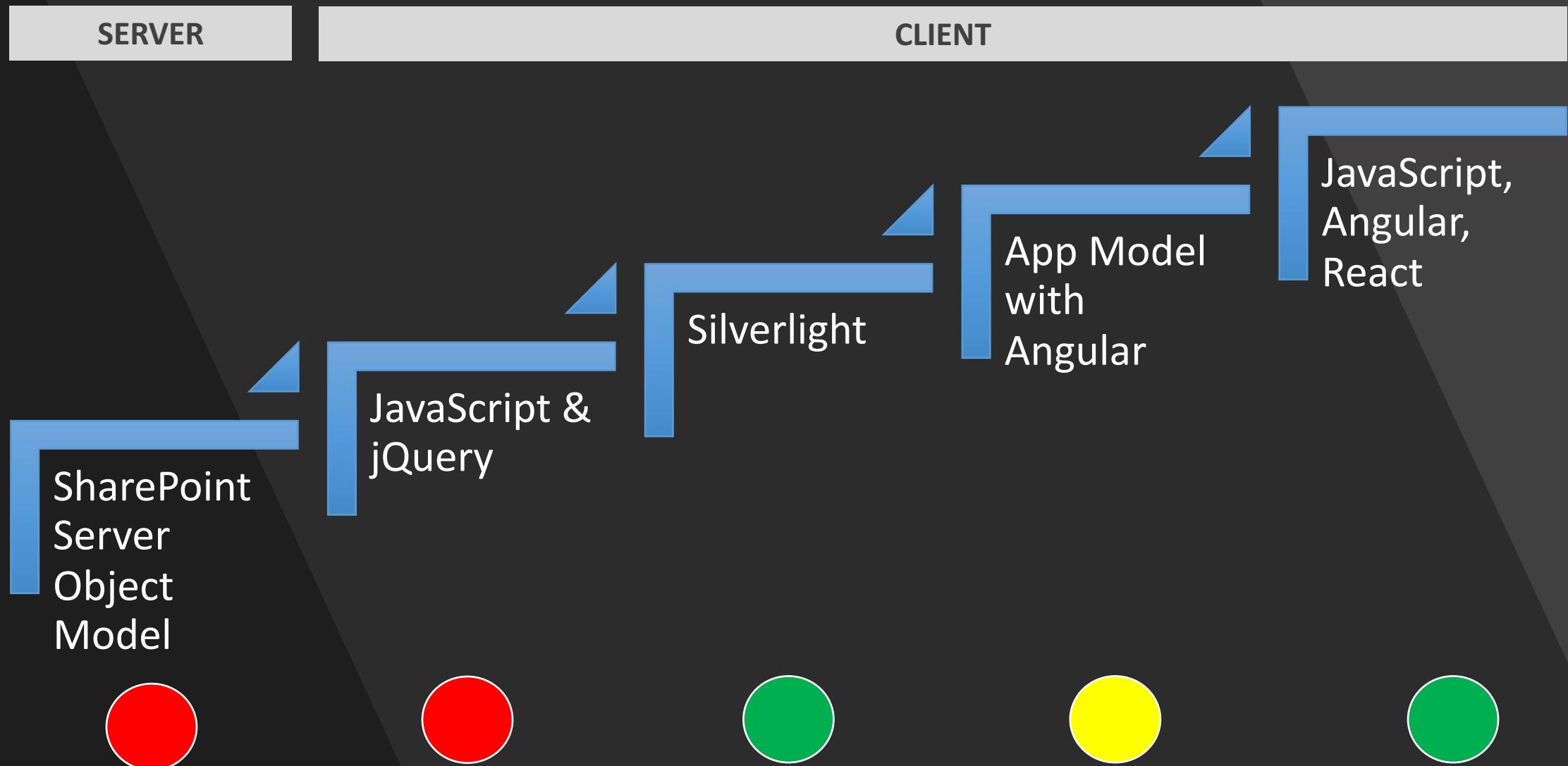
Administrator

- Set quotas and monitor customizations

Site Collection Owner

- Install customizations
- Activate and use customizations

My SharePoint Development Path



SharePoint Development Challenges



Performance
and Stability



Lifecycle
Management



Not Cloud
Ready



Resources

Client-Side Development Approach



Infrastructure

- Isolation & multitenant
- Runs in client browsers
- Reduces risk to farm
- Simplifies SharePoint upgrades
- Cloud ready

Developers

- Larger developer reach
- Use industry standards
- Use non-SharePoint technologies
- Lifecycle management
(development, deployment,
versioning, upgrades)

SharePoint Client-side Data Access

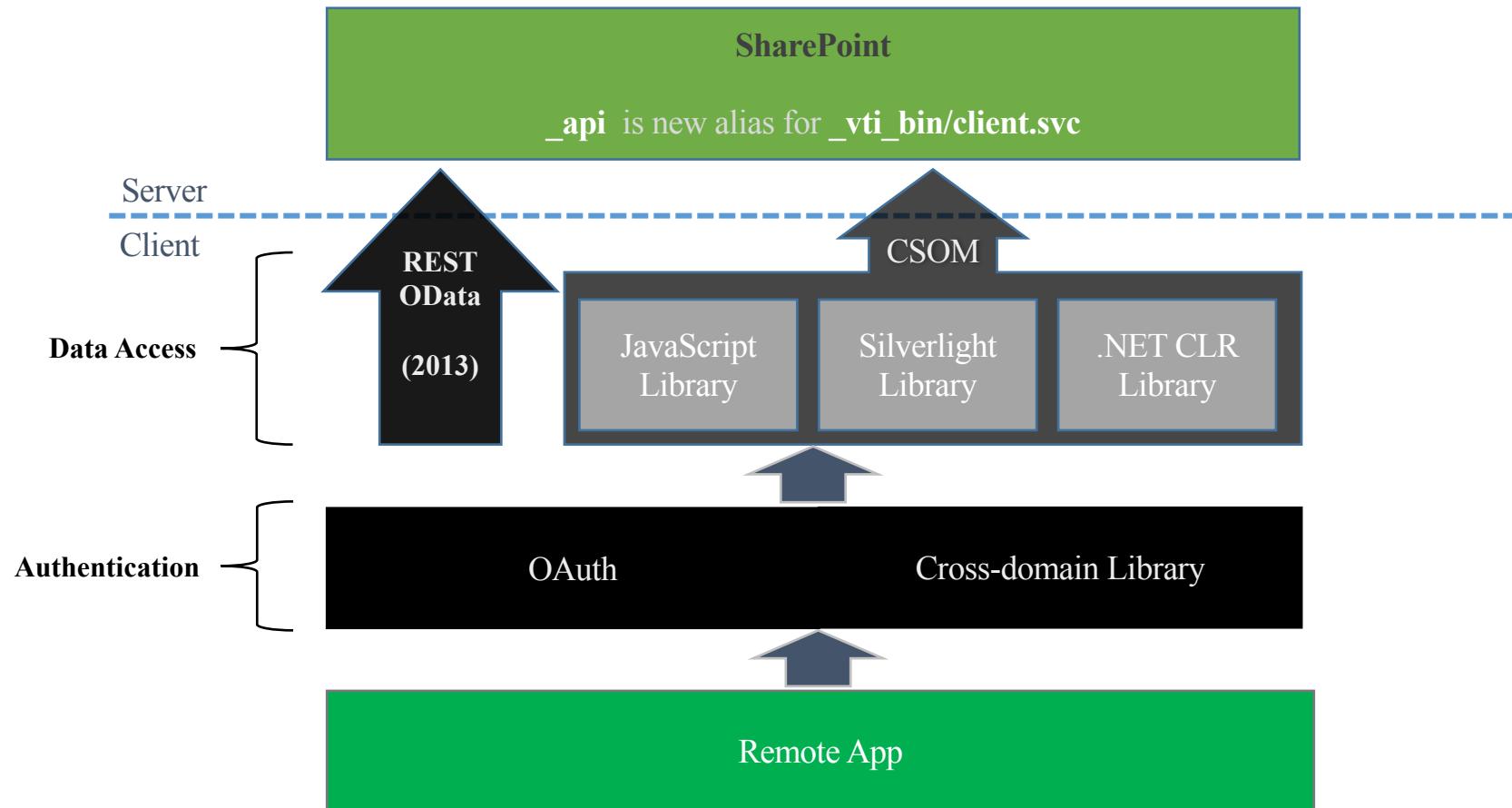
SOAP

SharePoint
2010 REST

SharePoint
2013 REST

Office
Graph

SharePoint 2013 Remote API



Note: ASMX web services have been deprecated

Key Players

- IDE: Visual Studio, VS Code, WebStorm, Brackets, Atom, Sublime Text
- Third-party JavaScript frameworks: jQuery, jQuery UI, Bootstrap, AngularJS, ReactJS, Moment, Font Awesome, SPServices, Lodash
- Browser Plug-ins: Firebug, Postman
- Task Runners: Gulp, Grunt, Webpack
- Package Managers: npm, Yarn
- Source control: Git, Visual Studio Online
- Other tools: Node.js, Fiddler, PhantomJS
- Other areas: ES6, Responsive Design, HTML5, CSS

React Overview



// comments between attributes,
// namespaced components, and
// non-alpha chars in tag/attribute name

```
<div onClick={this.handleClick}>  
  <EditInput></EditInput>  
  <FormInput></FormInput>  
  <Span>{this.state.value}</Span>  
  <FormInput hasError={true}>  
    /* here's another comment */  
    <div>  
      <input type="text" className="styled-input"/>  
    </div>  
  </FormInput>  
</div>
```



React Key Concepts

Build
Components,
Not Templates

Re-render,
Don't Mutate

Fast Virtual
DOM

Components

- Small, reusable widgets of your application
- UI layout and the UI logic are tightly coupled
- Can be individual or have a parent/child relationship
- Use state for storing data within the component
- Use props to pass data and events to a child component



Human Resources

Dep

Annual Employee Appreciation Celebration Planning is Underway!

Click here to see some of the exciting events we have in store for you!

News & Announcements



June 7, 2015

The Benefits of Blogging

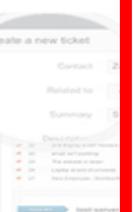
Blogging for business is not a new concept, but for many small businesses, especially service businesses, it can be an...

[Read More.](#)

June 17, 2015

Flying Pig 5K Marathon

It's the largest weekend party in town, and you can be part of it! Whether you're a runner or walker, whether you want to...

[Read More.](#)

June 23, 2015

Automate Your Business

We are excited to announce a new help desk system that will be available to your...

[Read More.](#)

12 Projects	
Red Projects	
!	Firetruck →
!	Irongate →
All Projects	
	Bladerunner →
▲	Constantine →
▲	Excalibur →
!	Firetruck →
!	Irongate →
	Kryptonite →
	Mad Hatter →
▲	Moonshine →
	Orion →
▲	Revolution →
▲	Sputnik →
	Zircon →

Props vs State

	prop	state
Can you use a plain JavaScript object to store data?	Yes	Yes
Can get initial value from parent Component?	Yes	Yes
Can be changed by parent Component?	Yes	No
Can set default values inside Component?	Yes	Yes
Can change inside Component?	No	Yes
Can set initial value for child Components?	Yes	Yes
Can change in child Components?	Yes	No
Does change trigger render event?	Yes	Yes

```
    );
    return (
      <div className="component-container">
        <div className={styles.slider}>
          <Slider {...settings}>
            {this
              .props
              .announcements
              .map(function (announcement, index) {
                return (
                  <div key={'announcement-' + index}>
                    <AnnouncementsItem item={announcement}/>
                  </div>
                );
              })
            }
          </Slider>
        </div>
      </div>
    );
  }
}
```

JSX

- Syntax extension to JavaScript
- Combines templates with JavaScript
- Declarative description of the UI inline in JavaScript code
- Use inside if statements and for loops, assign to variables, accept as arguments, return from functions
- Wrap JavaScript expressions in {} to be used within an element
- Preprocessor translates syntax into plain JavaScript objects

Deployment Workflow

Local Test

- Build process
- Disable web security (--disable-web-security) errors in Chrome
- Functional testing

Upload Source Code to Document library

- Same folder structure as local environment
- Only the build files are needed
- Upload source code

SharePoint Deployment & Test

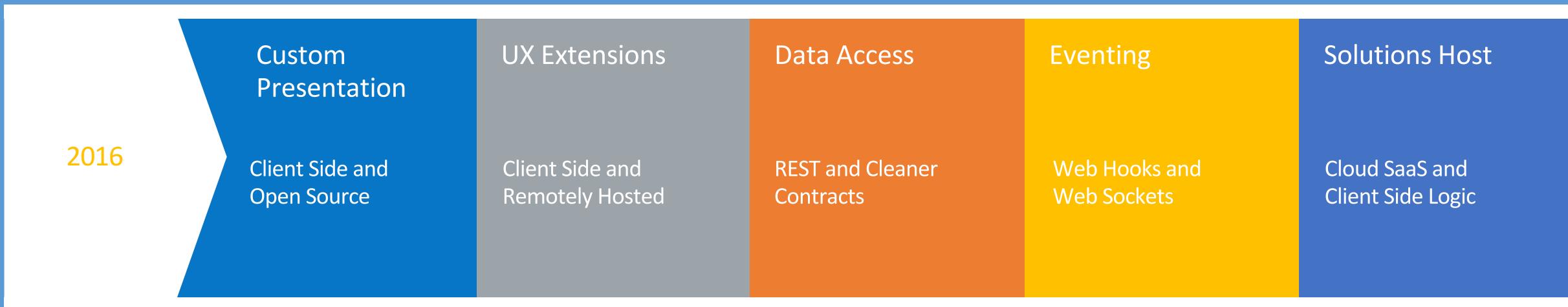
- Add Content Editor to Page
- Add markup
- Functional testing

Demo – SharePoint & React

SharePoint Framework

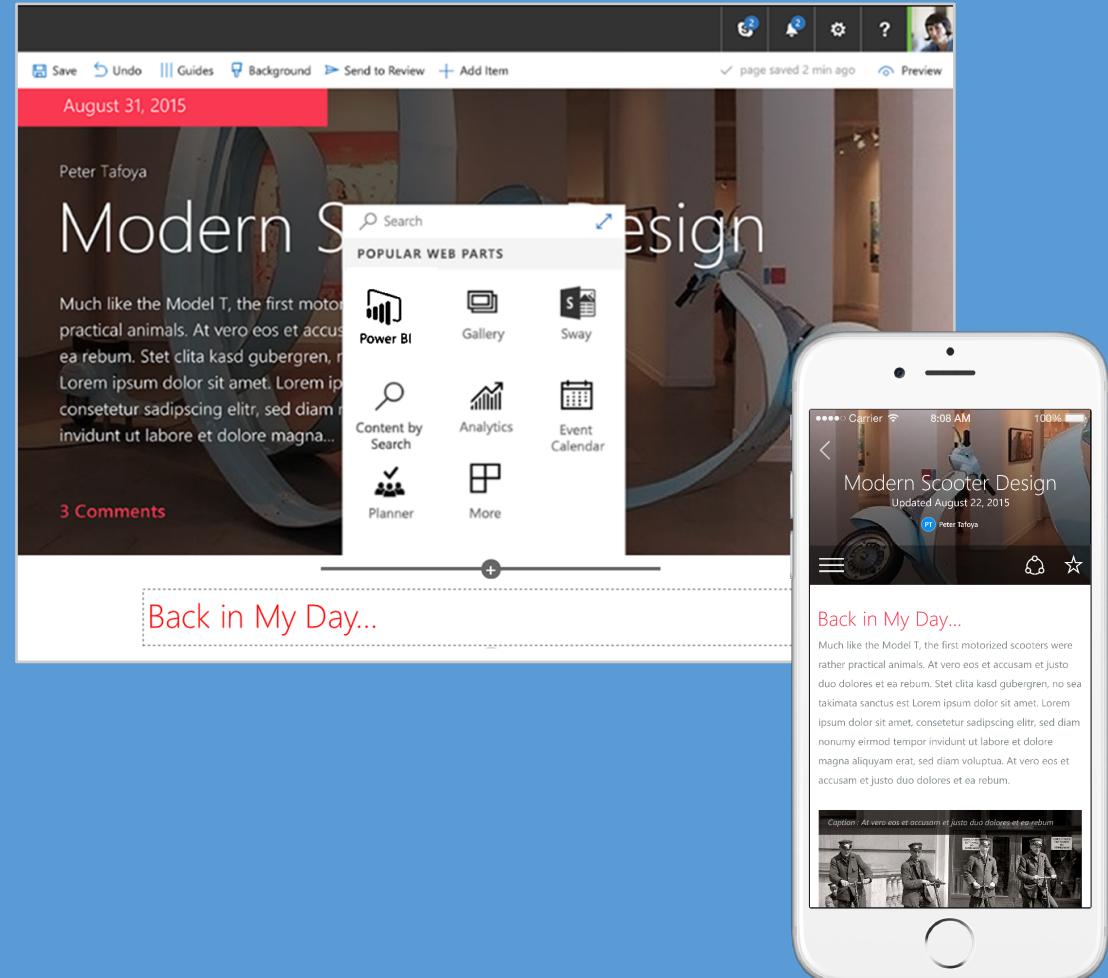
SharePoint Framework

- New Microsoft-recommended, client-side framework coming to SharePoint Online and SharePoint 2016
- Build applications using the latest JavaScript frameworks including Angular and React
- Use the latest tools: NodeJS, NPM, Gulp, Yeoman, TypeScript

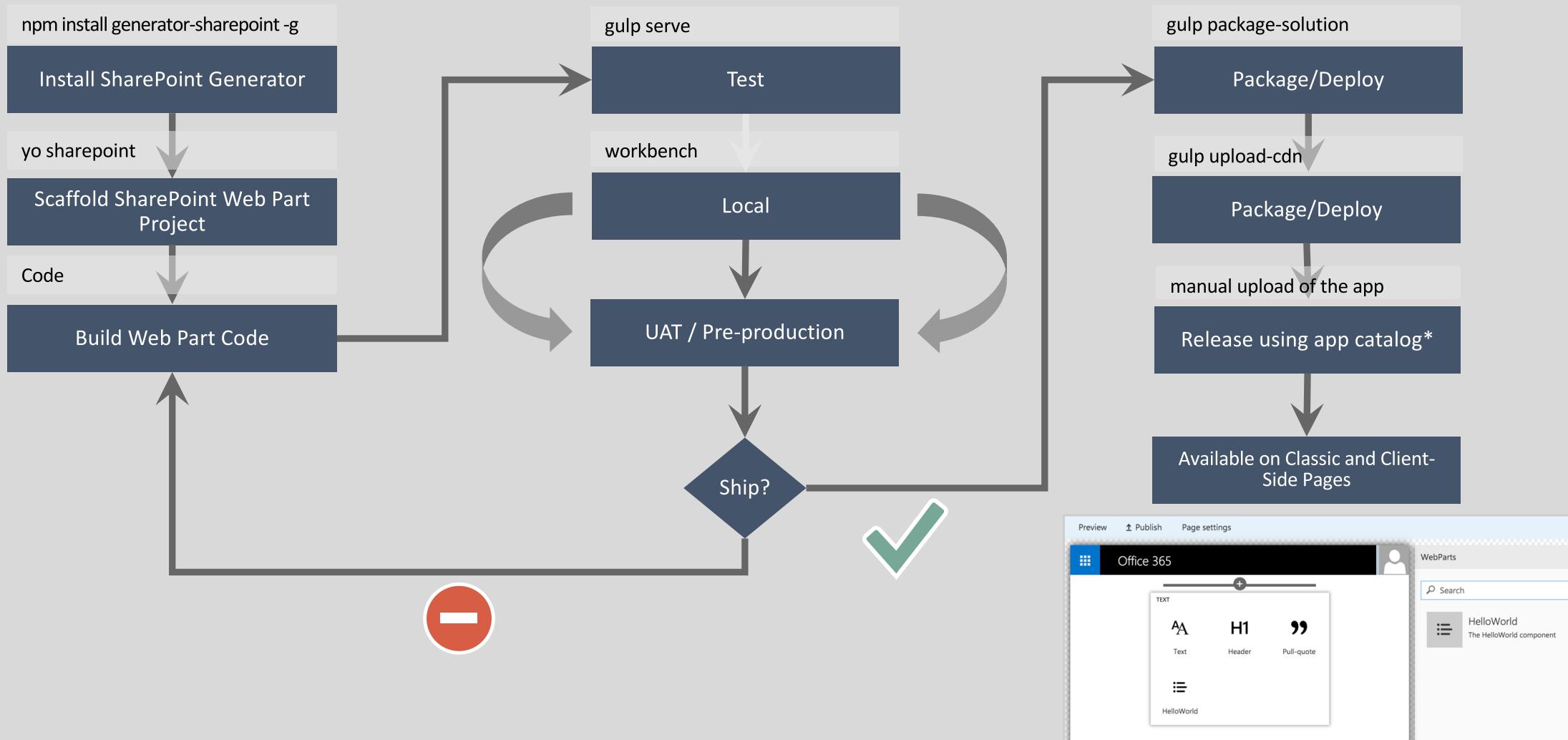


Scenarios

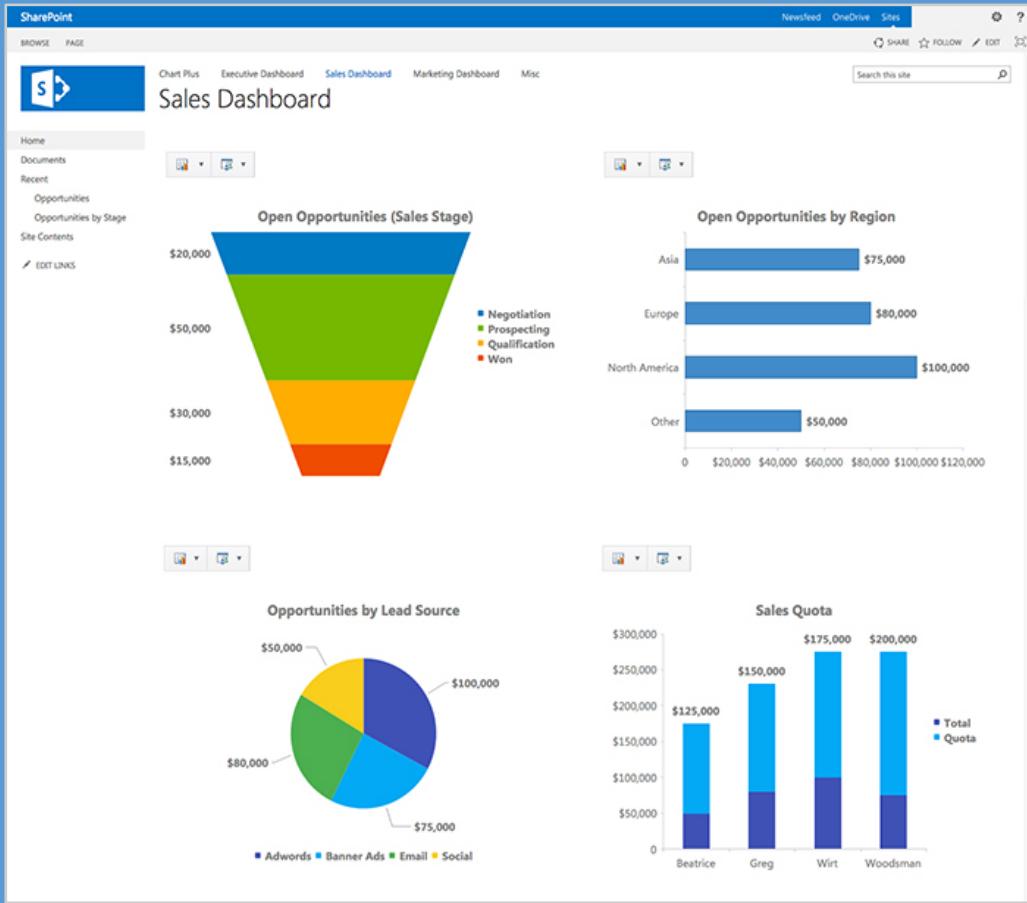
- Enhancing the out of the box experience
- Branding custom portals
- Building custom Intranet Portals
- Building custom Web Parts
- Building vertical focused applications
- Integrating with external systems



Client-side Web Part Build Flow



Client-Side Web Parts



Web Part Value

End users add functionality to SharePoint Experiences
Configurable, reusable, purpose built components
built by developers
Framework for connecting related components
Context aware

Client-side Web Parts remain Web Parts

- Built for the modern, JavaScript-driven web
- Can run directly inside a SharePoint page, new and existing
- Web framework agnostic
- Support both on-premises and Office 365

Demo – SharePoint Framework

Office Fabric

Office Fabric

- <http://dev.office.com/fabric>
- “The official front-end framework for building experiences that fit seamlessly into Office and Office 365”
- Language options: Vanilla JavaScript, ReactJS, and AngularJS
- Follows Material Design guidelines

The image shows three main sections of the Fabric design system. On the left, under 'Styles', there is a purple background featuring the letters 'Aa' in white, followed by a color palette consisting of four squares in teal, red, yellow, and brown. Below this is a section titled 'Styles' with a description. In the center, under 'Icons', there is a purple background with three circular icons: a globe, a document with the Microsoft logo, and a compass. Below this is a section titled 'Icons' with a description. On the right, under 'Components', there is a purple background showing a screenshot of a user interface with a navigation bar, a content area with a document icon, and a footer. Below this is a section titled 'Components' with a description.

Styles	Icons	Components
Fabric gives you access to Segoe, Microsoft's official typeface, along with the color palette, type ramp, icons, and responsive grid for Office 365.	Fabric includes Office's official product icons. Fabric also provides a suite of product and document symbols, so you can use the same metaphors we use.	Components are the building blocks of your UI. Fabric has a variety of components, including navigation, commands, containers, and content.

Demo – Office Fabric
