CROSS-FRAMEWOR
COMPONENTS

ArcGIS Portal App

REUSABLE UI COMPONEN

Currently we use Dojo's Dijit library to create re
components in both the JS API and Por

WHY DIJITS?

- Cross-browser
- Accessible
- Localizable
- Themeable

PROBLEMS WITH DIJIT

- Lots of complexity
- No future in Dojo 2.0
- Needs wrapping to use with other to
- Hard to style

How can we build UI components with the wid
build tools, module systems and app frame

- Angular 1.0 - Developers Site, Open Data Adr
- Angular 2.0 - Insights
- Backbone - Open Data
- Ember - Open Data, Operations Dashboard, V
- Dojo - ArcGIS Online/Portal App
- jQuery - My Stories

WEB COMPONENTS

New web standard for building reusable UI co

- Custom Elements
- Shadow DOM
- Templates
- HTMl Imports

CUSTOM ELEMENTS

Create custom HTML tags like `<item-rati` `<share-button>`. These custom elements b like `<a>`, `<button>` or `<form>`.

SHADOW DOM

Create fragments of DOM to isolate styles and
components from the rest of the page

TEMPLATES

Create reusable DOM fragments that can be
components.

HTML IMPORTS

Combine HTML, CSS, and JavaScript into a sing
can be imported via HTML.

THE STATE OF WEB COMPON

Feature	Chrome	Firefox	Safar
Custom Elements	✓	✓ (Flag)	?
Shadow DOM	✓	✓ (Flag)	?
Templates	✓	✓	✓
HTML Imports	✓	X	?

Are We Componentized Yet?

POLYFILLING WEB COMPON

Use existing APIs to add support for future stand
for all browsers back to IE 9.

PRACTICAL WEB COMPONE

- Polyfilled Shadow DOM cannot encapsula
- Templates are only useful with HTML Imp
- Firefox won't support HTML Imports

CUSTOM ELEMENTS

Even without Shadow DOM, Templates, and HTM
Custom Elements are still amazingly useful. C
simple component to rate an item in Por

A SIMPLE ITEM RATING COMP

```
<item-rating
    itemid="30e5fe3149c34df1ba922e6f5bbf808f"
    numratings="6"
    rating="4.25"
></item-rating>
```

```
var itemRating = new ItemRating({
    rating: 4.25,
    numratings: 6,
    itemid: '30e5fe3149c34df1ba922e6f5bbf808f'
});

document.body.appendChild(itemRating);
```

APPS WITH COMPONENT

```
<h1>My Item</h1>

<arcgis-item-rating
    itemid="30e5fe3149c34df1ba922e6f5bbf808f"
    numratings="6"
    rating="4.25"></arcgis-item-rating>

<a onclick="document.getElementById('#share-modal').open(

<calcite-modal id="#share-modal">
    <h2>Share</h2>
    <arcgis-share-button
        itemid="30e5fe3149c34df1ba922e6f5bbf808f"></arcgi
<calcite-modal>
```

INSIDE A CUSTOM ELEME

```
class ItemRating extends HTMLElement {
  createdCallback () {
    // called when the element is first created
  }

  attachedCallback () {
        // called whenever an element is added to the DOM
  }

  detachedCallback () {
    // called when the element is removed from the DOM
  }

  attributeChangedCallback (attribute, oldValue, newValue
        // called whenever an attribute changes on an ele
    }
```

Full Source

FRAMEWORK COMPATIBIL

- Declarative API
- Programatic API
- Backbone
- Angular 1.0
- Angular 2.0
- Ember 2.0
- Aurelia
- JS API

JavaScript libraries like Dojo and jQuery woul
programatic or declarative APIs.

FUTURE PROOFING

- Move to an app framework in the future
- All app frameworks need to work with DOM
- Other teams can use components from port

CHALLENGES

- Building and Compiling
- Localization
- Accessibility
- Style Collisions
- Boilerplate

BUILDING AND COMPILIN

Require adding a compiler to compile the ES 20
We could use either Babel or TypeScript for this
before the main Dojo build runs.

LOCALIZATION - SHORT TE

Use the existing localization tools from D

```
import i18n from 'dojo/i18n!arcgisonline/nls';

class ItemRating extends HTMLElement {
    // ... use ItemRating.i18n
}

const ItemRating.i18n = i18n;
```

This cannot be a permanent solution since it
components cannot be used outside Porta

LOCALIZATION - SHORT-MID

Pass i18n strings in as attributes. This means w
existing i18n capabilities from Dojo or other fra

```
<arcgis-item-rating
    itemid="30e5fe3149c34df1ba922e6f5bbf808f"
    numratings="6"
    rating="4.25"
    i18n.rating="Rating"
    i18n.ratings="Ratings"
    i18n.poor="Poor"></arcgis-item-rating>
```

LOCALIZATION - LONG TE

Bundle localizations for each component sepa
other teams do not have to rely on an existing i
This should discussed in more detail with the

ACCESSIBILITY

Since web components are simply DOM elemen
use the standard ARIA best practices to ensure
accessible.

STYLE COLLISIONS - SHORT

Global CSS will cause all sorts of problems wi
colliding. In the short term for Portal App we ca
Calcite Web will be present.

STYLE COLLISIONS - MID T

Baring full shadow DOM support from browsers

1. Namespaced selectors `.item-rating-too`
2. Use SASS `@extend` to map `.item-rating-`
   to proper styles.

Can support both Calcite Web and Calcite Bo

STYLE COLLISONS - MID TE

Inline styles are becoming a popular way to so
problems. Libraries like Radium and CSS Module
able to solve these problems in absence of Sha

See CSS In JS for more info.

STYLE COLLISIONS - LONG T

Wait for Shadow DOM to seperate styles inside c
Probally combined with a mid-term solution t
styles with components.

BOILERPLATE

Currently custom elements require lots of boi
setting up attributes, events, properties, ect... is
time consuming.

BOILERPLATE - SOLUTIO

Use ES 7 decorators like Ember, Angular, and T

```
@attribute('itemid', String)
@attribute('rating', Number)
@attribute('numratings', Number)
@bindEvent('click', 'a', 'handleStarClick')
@watchAttribute('rating', 'updateRating')
@watchAttribute('numratings', 'updateNumRatings')
@Accessor()
class ItemRating extends HTMLElement {
  // ...

  handleStarClick () { /* ... */ }
  updateRating () { /* ... */ }
  updateNumRatings () { /* ... */ }
}
```

I have some potentially helpful decorators desi

IMPLEMENTATION

How do components interact with the Port

1. Requests item details from API
2. Builds components
3. Listens for events like `rateitem`
4. App makes API calls
5. Updates `<item-rating>` attribut

IMPLEMENTATION

As much as possible components should not kn
the JS API or the REST APIs.

UI component shouldn't be dealing with bussin
Some components might needs maps ect... were
need to break this pattern.

PROPOSED PLAN

Start implementing new item page design using

- Add compilers to build tools
- Break down items page into compone
- Begin building components
- Start to wire components to the API

LONG TERM BENEFITS

- Easy to use an app framework in the future if n
- Components can be shared with applications o
  Portal App
- App becomes highly modular and easy to reas