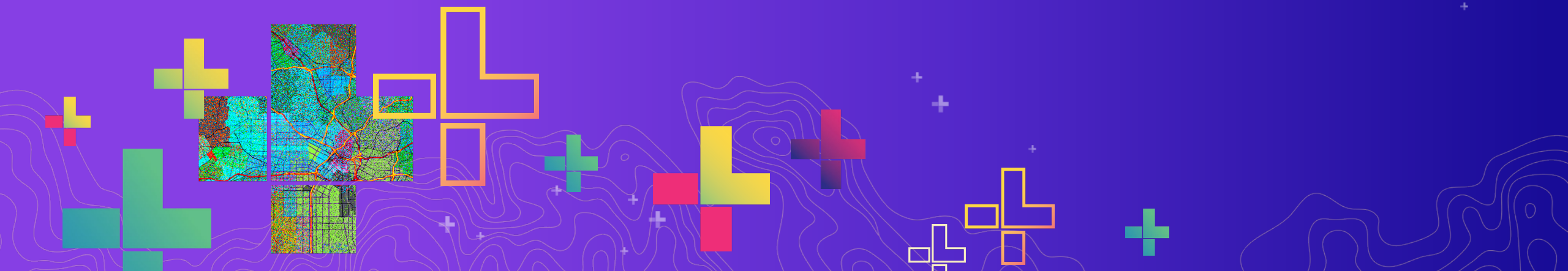




Web AppBuilder for ArcGIS: A Deep Dive in Enterprise Deployments

David Jacobs and Nick Brueggemann

2020 ESRI DEVELOPER SUMMIT | Palm Springs, CA



What we'll cover...

Subhead Here

- Who are We?
- What is “Enterprise” Development?
- App Building Process and Pipeline
- Source Control Patterns
- Multi-Environment Deployment Considerations
- Architectural Patterns
- Tips and Tricks



Who Are We?



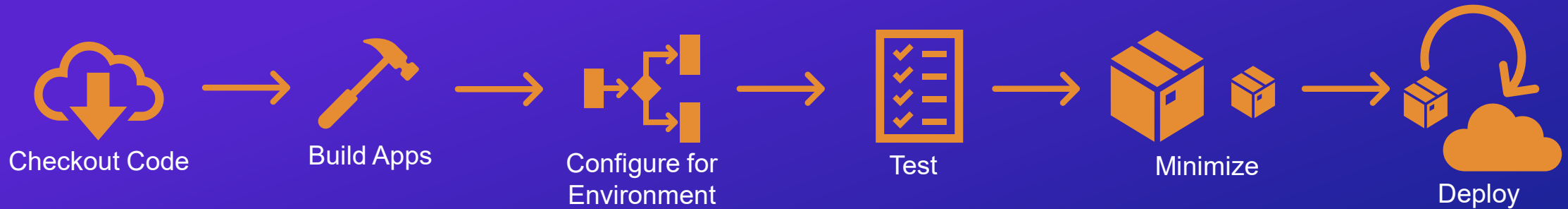
Enterprise Development

What do we consider to be “Enterprise” Development?

- Generally it's a project that has more than one developer contributing
- Typically it's a project that is deployed into multiple environments (i.e development, staging, and production)
- Usually it's a project that has grown into a significant amount of complexity
 - Application code that doesn't “fit” cleanly into a single widget
 - Out of the Box widgets have been customized or extended
 - WAB code itself may have been modified

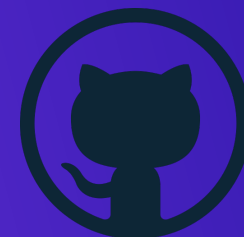
Build Pipeline

- What is a build pipeline?
- What are the steps in a build pipeline?



* The main takeaway here is to go from code checkout to deployment with no manual steps

Source Control



Source Code Management Paradigms

- Essential for team collaboration
- There are a few different ways to do this
- We have a recommended approach but one size doesn't always fit all.

Source Code Management Paradigms

No Source Control



Source Control Everything

- arc-gis-web-appbuilder
 - client
 - stemapp
 - ...
 - docs
 - server

Source Control a Single Application

- arc-gis-web-appbuilder
 - client
 - stemapp
 - ...
 - docs
 - server
 - apps
 - 2

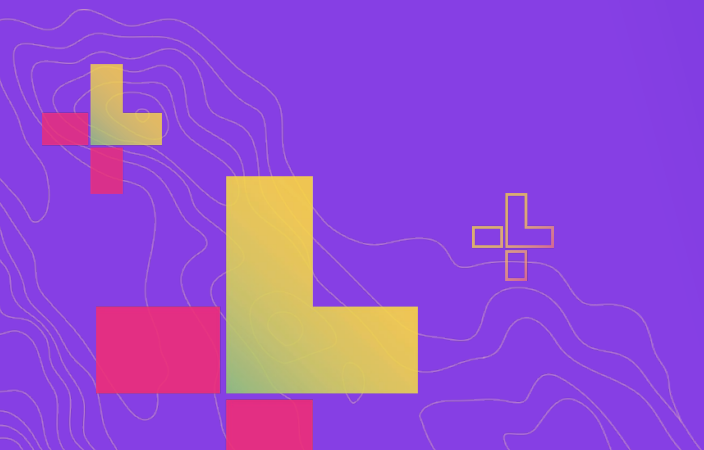
Source Control Only your Code

- my-project
 - build
 - src
 - custom
 - libs
 - widgets
 - wab *

* Added to gitignore



Build Apps



App Build Process

Create Web Apps (outside of WAB)

- Steps to create a Web Application outside of WebApplicationBuilder
- Gulp example
- Manual example

App Build Process

Overview

stemapp



images
jimu.js
libs
themes
widgets

gulp install

.apps/MyApp



images
jimu.js
libs
themes
widgets

your repo



images
jimu.js
libs
themes
widgets

Build



App Build Process

Intro to Task Runners for Automation

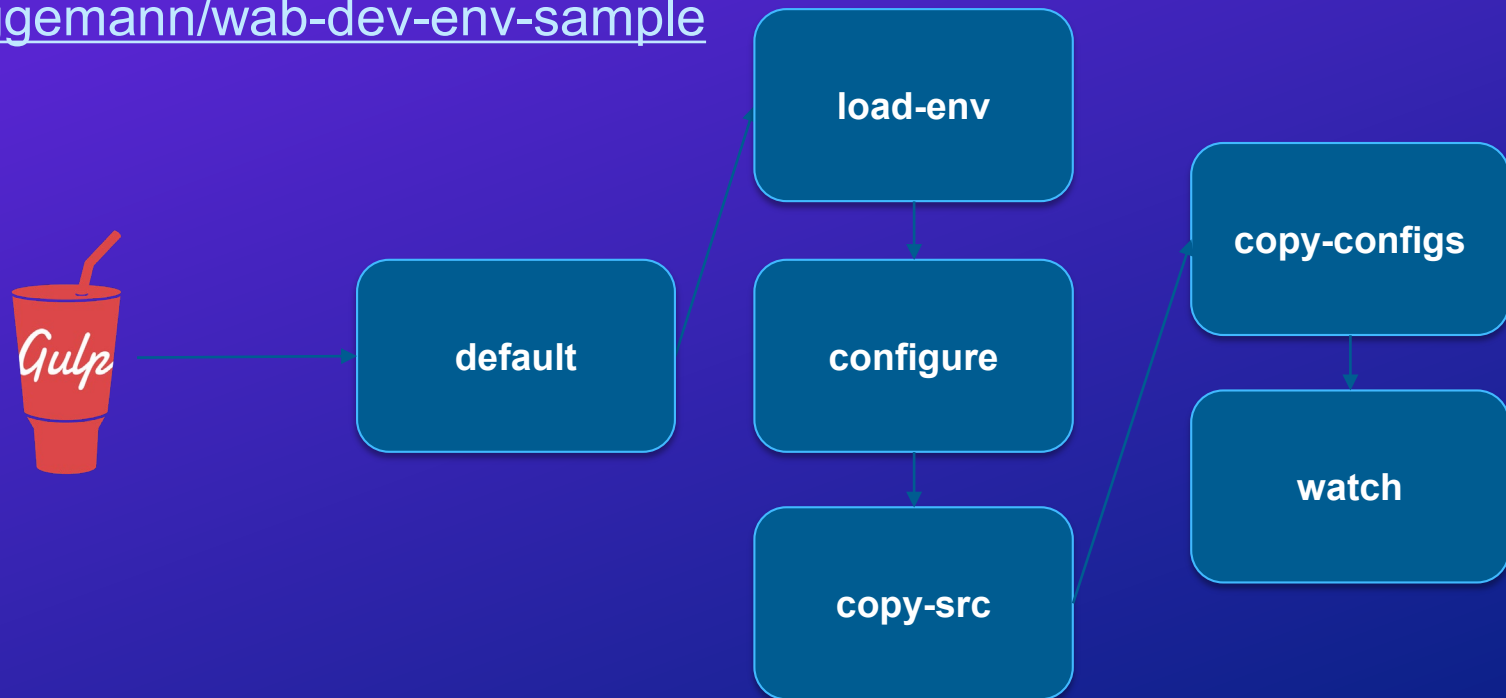
- Both of these are task runners that run using NodeJS
- Used for automating build steps
- They are similar but different. Use whatever you're comfortable with.
- Gulp has a rich set of available plugins: <https://gulpjs.com/plugins/>



App Build Process

Digging into the build process

- With Gulp you build concise tasks for each build step
- Build as many as you need
- Checkout other WAB sessions for more on this
- <https://github.com/nbrueggemann/wab-dev-env-sample>



Configure for Environment



Multi-Environment Configuration

- Typically configurations point to services your widget depend on
- There may need to change for deployments to different environments

Typical Config

```
{  
  "service": {  
    "url": "https://sampleserver6.arcgisonline.com/arcgis/rest/serv..."  
  }  
}
```

But We Need

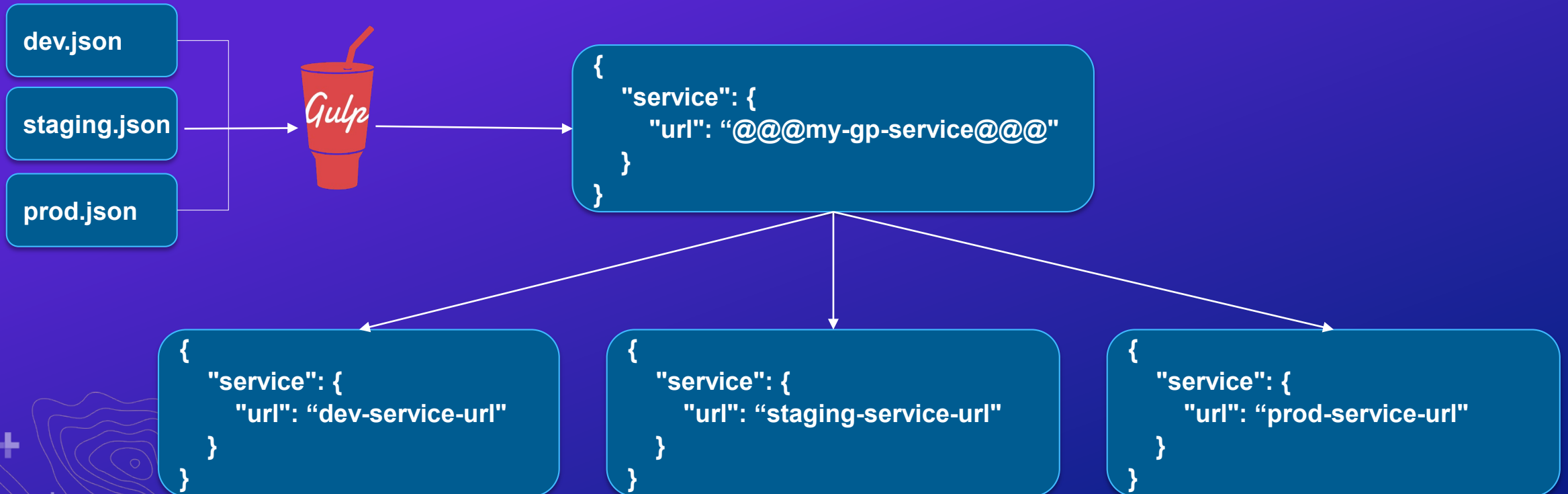
```
{  
  "service": {  
    "url": "dev-service-url"  
  }  
}
```

```
{  
  "service": {  
    "url": "staging-service-url"  
  }  
}
```

```
{  
  "service": {  
    "url": "prod-service-url"  
  }  
}
```

Multi-Environment Configuration

- We use gulp-replace as a way to build these configurations for us
- Target one env at a time: gulp --env=dev (yargs for arguments)



Multi-Environment Configuration

Demo Code:

<https://github.com/nbrueggemann/wab-dev-env-sample>



Test

Minimize



Deployment



Deployment

- We often deploy to tomcat web servers. The automated process we use is as follows:
 - Zip up application using gulp-zip
 - Change extension to .war since that what tomcat expects.
 - Use node-ssh package to ssh into web servers are move the .war file to the server

Code Beyond Widgets

- Focus is typically on widgets and themes but customizing doesn't have to stop there
- Very valid reasons for putting code outside of widget
 - Shared code between widgets
 - Core application business logic
- Where should this code go?
 - custom/libs
 - “custom” folder is used so your code doesn't end up in the root of libs folder in wab.

Code Beyond Widgets

Shared code between widgets

Space-Time Filter

Select a start and end date/time filter for your data query.
Note: All times are in UTC (GMT, Zulu).

Current UTC Time: Tue, 27 Feb 2018 20:15:13 UTC

Start Time

February

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10

End Time

February

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Current Time Filter:
From: Tue, 27 Feb 2018 00:00:00 UTC
To: Tue, 27 Feb 2018 20:13:18 UTC

Spatial Filter: 4,037,678 Square Miles

Feature Counts:
Counts: 23,654

Attribute Filter

Current Time Filter:
From: Tue, 27 Feb 2018 00:00:00 UTC
To: Tue, 27 Feb 2018 20:13:18 UTC

Spatial Filter: 4,037,678 Square Miles

Feature Counts:
Counts: 23,654

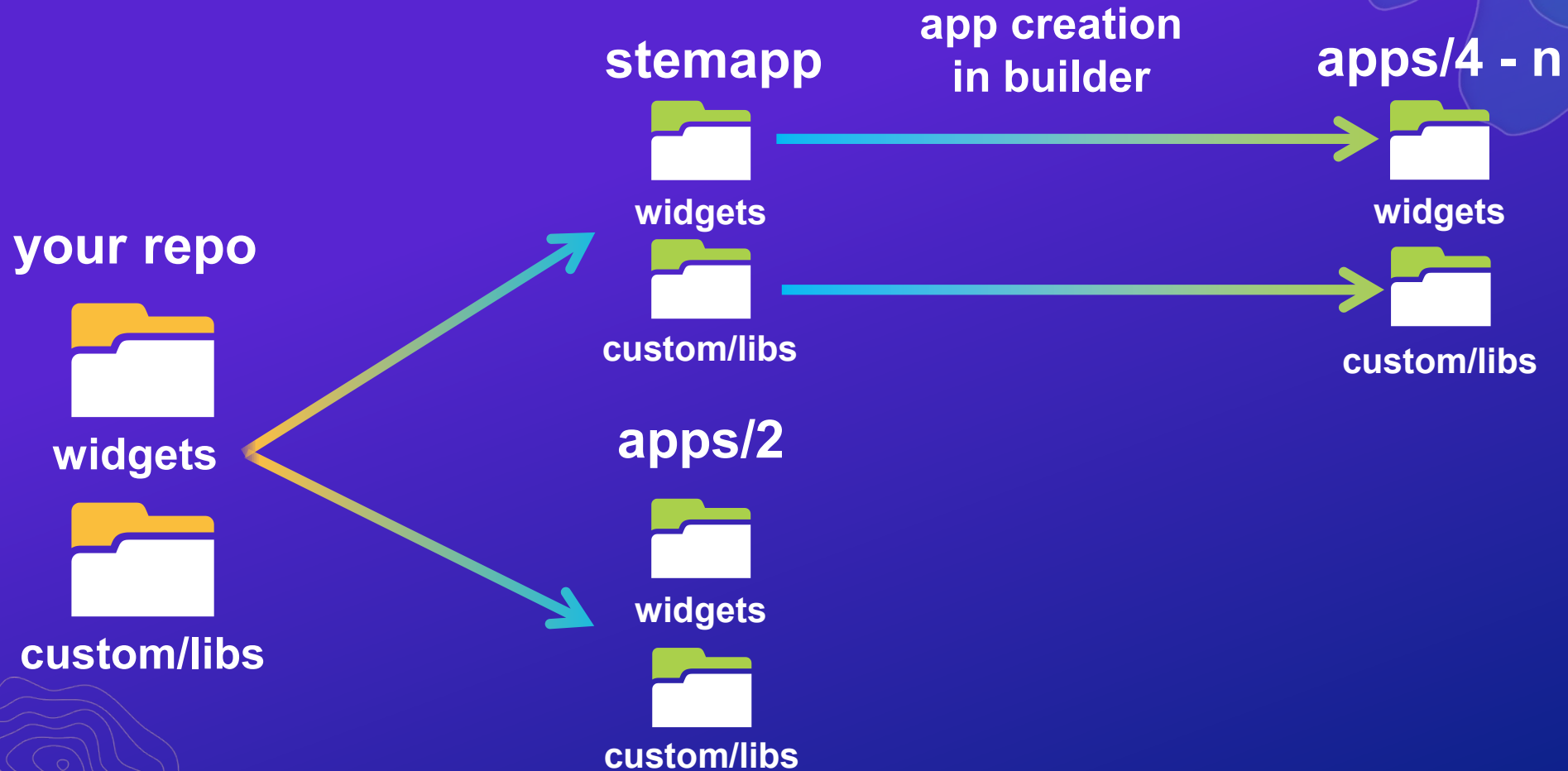
Reset Apply

Shared UI

- my-project
 - src
 - custom
 - libs
 - filter-status
 - Dijit files...

Code Beyond Widgets

Update build process to handle libs folder



Code Beyond Widgets

Drawbacks

- You lose the ability to pick up your widget and share it by itself.

Additional Tips and Tricks

- Peek into WAB and take a look around
- Understanding WAB configs



Getting to know jimv.js Better

- Web App Builder dev docs are good. But we have the code so we might as well dig into it!
- There's some good utilities available that aren't talked about. Examples:
 - CSVUtils
 - GeoJsonConverters
 - A whole slew of dijits

