



*Conscious Uncoupling
Drupal 8 meets Metalsmith*

Werner Glinka

Web Freelancer

Building Websites since 2005

Drupal Handyman since 2010

glinka.co

werner@glinka.co

tcdrupal.slack.com

Motivation

Why a Static Website?

What is Metalsmith?

Nunjucks Templating

The ORCA Client

The ORCA server

Questions

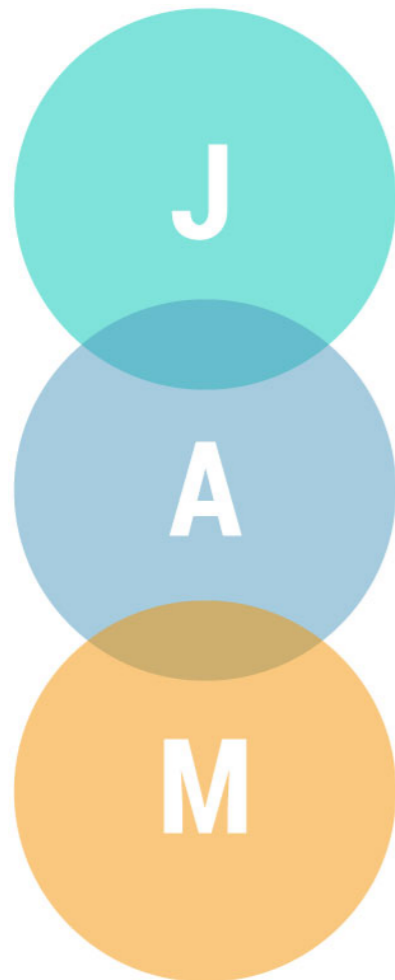
Motivation

Build an example of a cost effective small/medium business website by combining the superior content management capabilities of my favorite CMS with the benefits of a static website.

Why a Static Website

They are fast, secure, and inexpensive to host. They do NOT need any security patches, updates - no drama, no armageddons.

The JAM Stack



JavaScript

Any dynamic programming during the request/response cycle is handled by JavaScript, running entirely on the client.

APIs

All server-side processes or database actions are abstracted into reusable APIs, accessed over HTTP with JavaScript.

Markup

Templated markup should be prebuilt at deploy time, usually using a site generator for content sites, or a build tool for web apps.

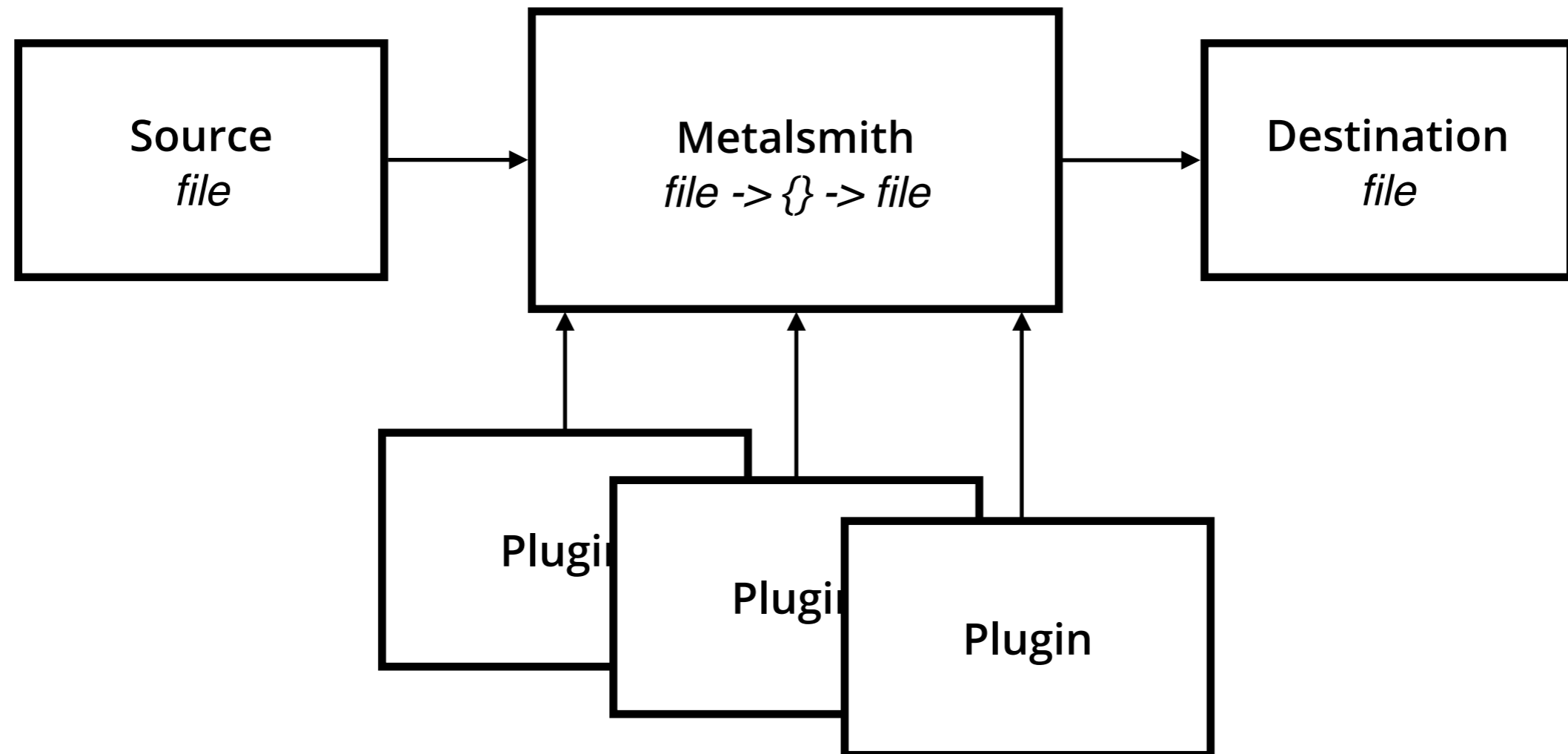
What is a Static Site Generator

Static site generators build HTML-only pages from templates and raw data. Data may be contained in JSON, YML or Markdown files. They offer some of the benefits of a CMS without the hosting, performance and security overheads.

What is Metalsmith?

“An extremely simple, pluggable static site generator”

What is Metalsmith?



What is Metalsmith?

example-file.md -> 

```

---
layout: blog-post.html


# meta data
title: Etiam Mollis Risus
description: "Vestibulum id ligula porta
..."

# page properties
blog_title: Etiam Mollis Risus
categories: category1
tags: [tag1, tagA, tag3]
author: [author2, author1]
featured_blog_post: true
featured_blog_post_order: 2

image:
  feature: default.jpg
  thumbnail: default_tn.jpg
date: 2017-03-12
body_classes: blog-post has-sidebar
---

## Nibh Fringilla Cursus
Cum sociis natoque penatibus et magnis dis
parturient montes, nascetur ridiculus mus.
Nullam id dolor id nibh ultricies vehicula
ut id elit...

```



```

{
  relative_to_source_path/example-file.md: {
    layout: blog-post.html,
    title: Etiam Mollis Risus,
    description: 'Vestibulum id ligula porta...',
    blog_title: Etiam Mollis Risus,
    categories: category1,
    tags: [tag1, tagA, tag3],
    author: [author2, author1],
    featured_blog_post: true,
    featured_blog_post_order: 2,
    image: {
      feature: default.jpg,
      thumbnail: default_tn.jpg
    },
    date: 2017-03-12,
    body_classes: blog-post has-sidebar,
    contents: '## Nibh Fringilla Cursus...',
    mode: '0777',
    stats: {}
  }
}

```

What is Metalsmith?

blog-post.html

```
<!doctype html>
<html>
<head>
  <title>{{ title }}</title>
</head>
<body>
  {{ contents | safe }}
</body>
</html>
```

What is Metalsmith?

A Simple Plugin

```
---
layout: blog-post.html

# meta data
title: Etiam Mollis Risus
draft: true
description: "Vestibulum id ligula porta ..."

# page properties
blog_title: Etiam Mollis Risus
categories: category1
tags: [tag1, tagA, tag3]
author: [author2, author1]
featured_blog_post: true
featured_blog_post_order: 2

image:
  feature: default.jpg
  thumbnail: default_tn.jpg
date: 2017-03-12
body_classes: blog-post has-sidebar
---

function plugin() {
  return function(files, metalsmith, done){
    setImmediate(done);
    Object.keys(files).forEach(function(file){
      var thisFile = files[file];
      if (thisFile.draft) delete files[file];
    });
  };
}

## Nibh Fringilla Cursus
Cum sociis natoque penatibus et magnis dis
parturient montes, nascetur ridiculus mus.
Nullam id dolor id nibh ultricies vehicula
ut id elit..
```

What is Metalsmith?

A Simple Build Script

```
Metalsmith(__dirname)
  .source('sourcepath')
  .destination('destpath')
  .use(markdown())
  .use(permalinks())
  .use(layouts())
  .build(function(err) {
    if (err) throw err;
  });
```

What is Metalsmith?

`example-file.md`

```
{  
  relative_to_source_path/example-file.md: {  
    layout: blog-post.html,  
    title: Etiam Mollis Risus,  
    ...  
    contents: '## Nibh Fringilla Cursus...',  
    ...  
  }  
}
```

What is Metalsmith?

Invoking the Markdown Plugin

```
{  
  relative_to_source_path/example-file.html: {  
    layout: blog-post.html,  
    title: Etiam Mollis Risus,  
    ...  
    contents: '<h2>Nibh Fringilla Cursus</h2>...',  
    ...  
  }  
}
```

What is Metalsmith?

Invoking the Permalink Plugin

```
{  
  relative_to_source_path/example-file.html: {  
    layout: blog-post.html,  
    title: Etiam Mollis Risus,  
    ...  
    contents: '<h2>Nibh Fringilla Cursus</h2>...',  
    path: 'example-file',  
    ...  
  }  
}
```


What is Metalsmith?

Invoking the Layouts Plugin

```
{
  relative_to_source_path/example-file.html: {
    layout: blog-post.html,
    title: Etiam Mollis Risus,
    ...
    contents: '<!doctype html>
      <html>
      <head>
        <title>Etiam Mollis Risus</title>
      </head>
      <body>
        <h2>Nibh Fringilla Cursus</h2>...
      </body>
      </html>',
    path: 'example-file',
    ...
  }
}
```

Nunjucks

A templating language for JavaScript

Block inheritance

Includes

Flow Control

Macros

Nunjucks

A templating language for JavaScript

```
{{ variable | filter }}
```

```
{% extends "parent.html" %}
```

```
{% include "header.html" %}
```

```
{% block left %} ... {% endblock %}
```

```
{% if hungry %} ... {% elif tired %} ... {% else %} ... {% endif %}
```

```
{% for item in items %} ... {% else %} ... {% endfor %}
```

```
{% macro field(name, value='', type='text') %} ... {% endmacro %}
```

```
{# I am a comment #}
```

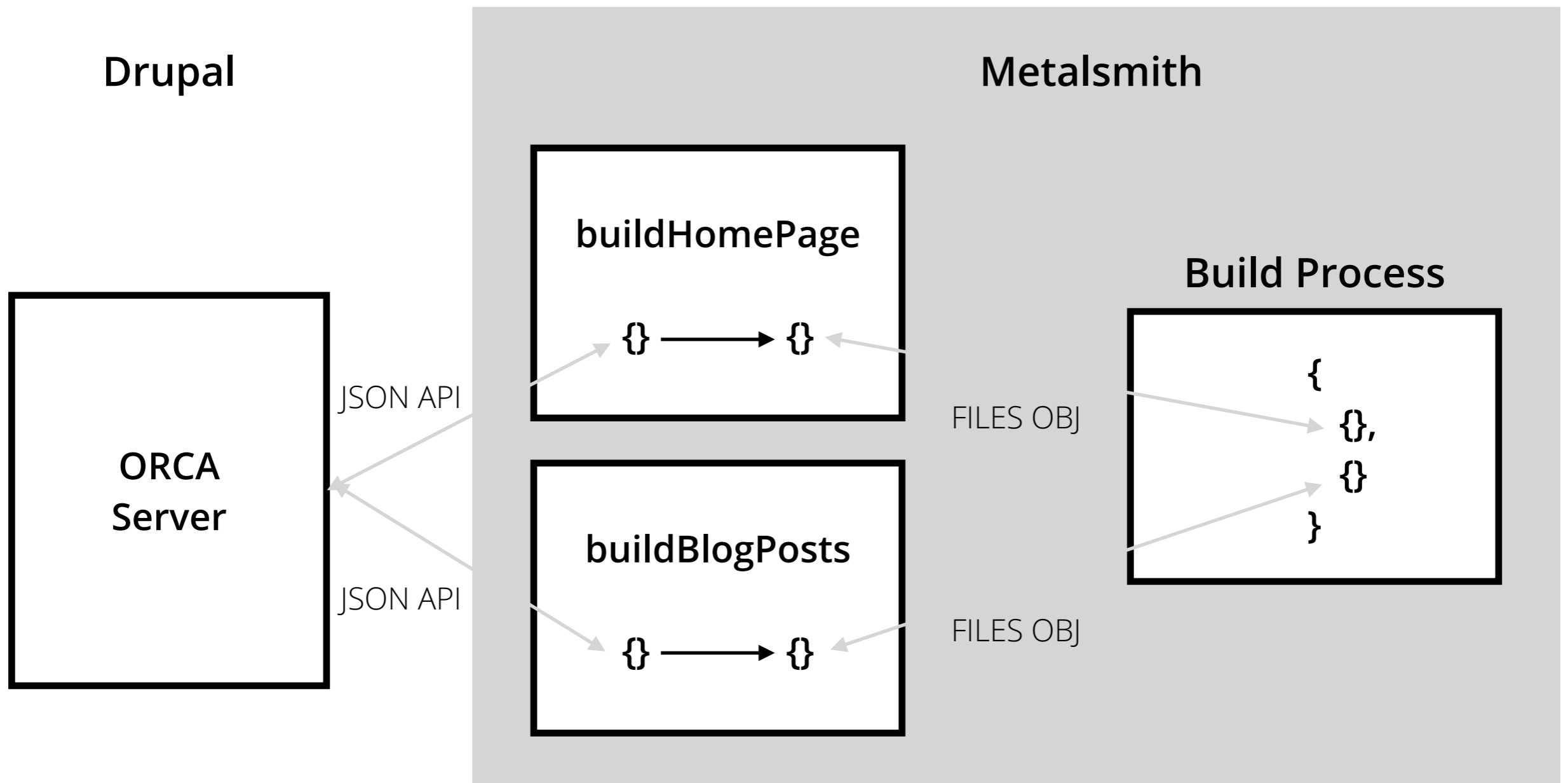
ORCA Client

It is not a Website

Unlike most headless implementations, the ORCA client is NOT a website. Rather, it is a static-site-generator build process that requests content from the ORCA server during build time. Once the site is deployed it doesn't communicate with the ORCA server at all.

ORCA Client

The Build Process



Gulp

A Javascripts Task Runner to Automate Tasks

Sass to CSS compilation

ES6 to ES5 Javascript transpilation

Updating browsers when saving a file

Bundling and minifying files for production

Gulp

The buildDev Task

```
gulp.task("buildDev", function (cb) {  
  "use strict";  
  sequence([  
    "vendorScripts",  
    "scripts",  
    "styles"  
  ],  
  "metalsmith",  
  cb  
);  
});
```

Gulp

The default Task

```
gulp.task("default", ["buildDev"], function () {
  "use strict";
  browserSync.init({
    server: {
      baseDir: "build"
    },
    open: false
  });

  gulp.watch([
    "./dev/scripts/**/*",
    "./dev/styles/**/*",
    "./dev/content/**/*",
    "./dev/layouts/**/*",
    "./dev/sources/**/*"
  ], ["refresh"]);
});
```


Continuous Deployment

Hosting on Netlify

Continuous deployment works by connecting a git repository to a Netlify site and keeping the two in sync.

It works for plain static sites, but it's even more powerful when you're using a static site generator or a front end build tool like Gulp.

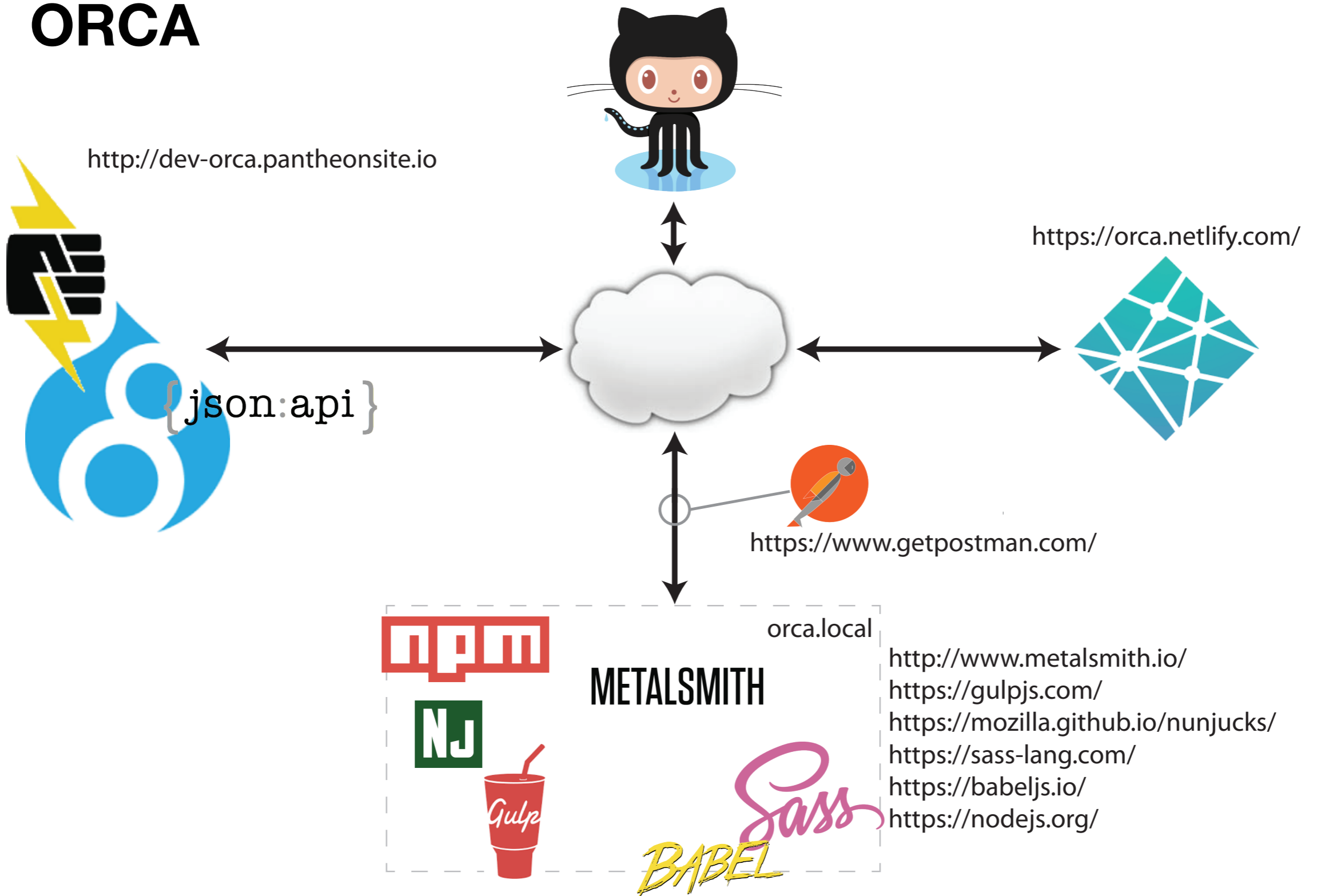
Netlify will run your build command and deploy the result whenever you push to your git repo.

No deploying without committing and pushing first

Easy collaboration through pull requests

ORCA

<https://github.com/wernerglinka/orca>



ORCA Server

A Drupal 8 Minimal Installation

Enabled in core:

admin_toolbar
admin_toolbar_extra_tools
text-editor
ckeditor
field UI
serialization
datetime

Plus:

jsonapi
admin_toolbar
insert
module_filter

ORCA Server JSON API

Implements the JSON API spec

Provides several features that streamline the interface:

- Collections that include filtering and sorting
- Resource embedding that minimizes http requests
- Sparce fields that shortens the http response to only necessary data

There is an excellent JSON API video series on YouTube:

`https://www.youtube.com/watch?v=-ZL3EAhnwc&index=1&list=PLZ0Q_ZMpYrZsy0-3IstImK1okrpfAjuMZ`

ORCA Server JSON API

```
“nid”: [
  {
    “value”: “1234”
  }
]
```



```
“nid”: “1234”
```

ORCA Server

Content Structure

Content is organized “page centric”.

There are two “page” content types:

Home Page

Blogpost

and “supporting” content types:

Blog Author

Projects

Services

Testimonials

This structure minimizes http requests. Just one request is needed to load the home page for example as JSON API allows us to embed all information into one http response.

ORCA Server

Building the API with Postman

The screenshot shows the Postman interface with a REST client request. The request is a GET call to the following URL:

```
http://dev-orca.pantheonsite.io/jsonapi/node/home_page?_format=api_json&filter[titleFilter][condition][path]=title&filter[titleFilter][condition][value]=Orca Home Global&fields[node--home_page]=field_site_name,field_welcome_text,field_welcome_bg_img&include=field_welcome_bg_img&fields[file--file]=url
```

The request parameters are:

Key	Value	Description
<input checked="" type="checkbox"/> _format	api_json	
<input checked="" type="checkbox"/> filter[titleFilter][condition][path]	title	
<input checked="" type="checkbox"/> filter[titleFilter][condition][value]	Orca Home Global	
<input checked="" type="checkbox"/> fields[node--home_page]	field_site_name,field_welcome_text,field_welcome_bg_img,field_abou...	
<input checked="" type="checkbox"/> include	field_welcome_bg_img,field_about_image	
<input checked="" type="checkbox"/> fields[file--file]	url	

The response is a JSON object with the following structure:

```
{  "data": [    {      "type": "node--home_page",      "id": "8ee7074a-2c7b-4f26-88ab-03242d5098f8",      "attributes": {        "field_about_byline": {          "value": "When you arrive at a fork in the road, take it",          "format": "basic_html",          "processed": "When you arrive at a fork in the road, take it"        },        "field_about_prose": {          "value": "<p>Morbi leo risus, porta ac <strong>consectetur ac, vestibulum at eros</strong>. Donec id elit non mi porta gravida at eget metus. Donec sed odio dui. Aenean eu leo quam. <em>Pellentesque ornare sem</em> lacinia quam venenatis vestibulum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Sed posuere consectetur est at lobortis.</p>\r\n",          "format": "basic_html",          "processed": "<p>Morbi leo risus, porta ac <strong>consectetur ac, vestibulum at eros</strong>. Donec id elit non mi porta gravida at eget metus. Donec sed odio dui. Aenean eu leo quam. <em>Pellentesque ornare sem</em> lacinia quam venenatis vestibulum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Sed posuere consectetur est at lobortis.</p>"        }      }    }  ]}
```

Lando

A Local Dev Environment for Drupal

Integrates with hosting providers like Pantheon
Mimics production environment locally.

Standardize your teams dev environments and tooling on OSX, Windows and Linux.

Store all of the above in a version controlled config file called `.lando.yml`

Lando

Preconfigured Recipes

Backdrop

Drupal 6

Drupal 7

Drupal 8

Pantheon

WordPress

**Using Lando makes synch with Pantheon a one
step process**

Lando

Working with Lando is Simple

Install Lando

Make a directory

```
> lando init pantheon
```

```
> lando pull
```

```
> lando start
```

Now your local site mirrors the code, files and db of the Pantheon instance

If you want to synch your local work with Pantheon

```
> lando push
```



Questions?



*Conscious Uncoupling -Drupal 8 meets
Metalsmith*

If you are looking for an **example** of a decoupled
Drupal 8 site feeding content to a Metalsmith
build process, you have come to the right place.