KALAMUNA

Let's Play.

Prototyping

Design + Content-first approaches to prototyping

KALAMUNA

Let's Play.

About Us

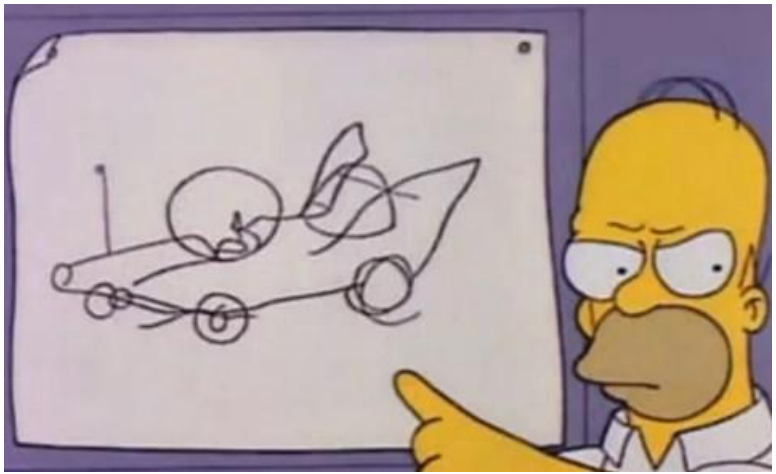
Kalamuna makes the Internet for for mission-driven organizations driven to tinker, critique, and change the way things are. We specialize in design, strategy, user experience, and development.

Trial vs Error

to build sites that MATTER

Prototyping = Solutions

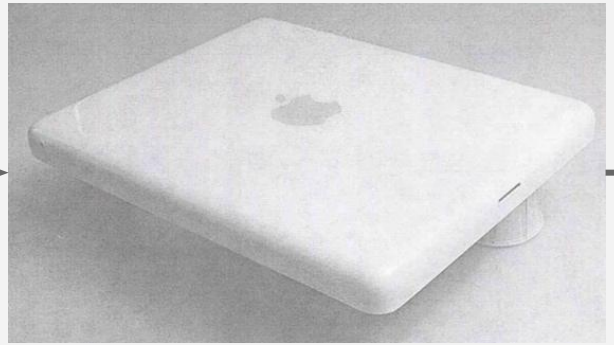
Can you imagine a world without prototyping?



nature is weird

KALAMUNA

iPad Prototypes



KALAMUNA

Why Prototyping

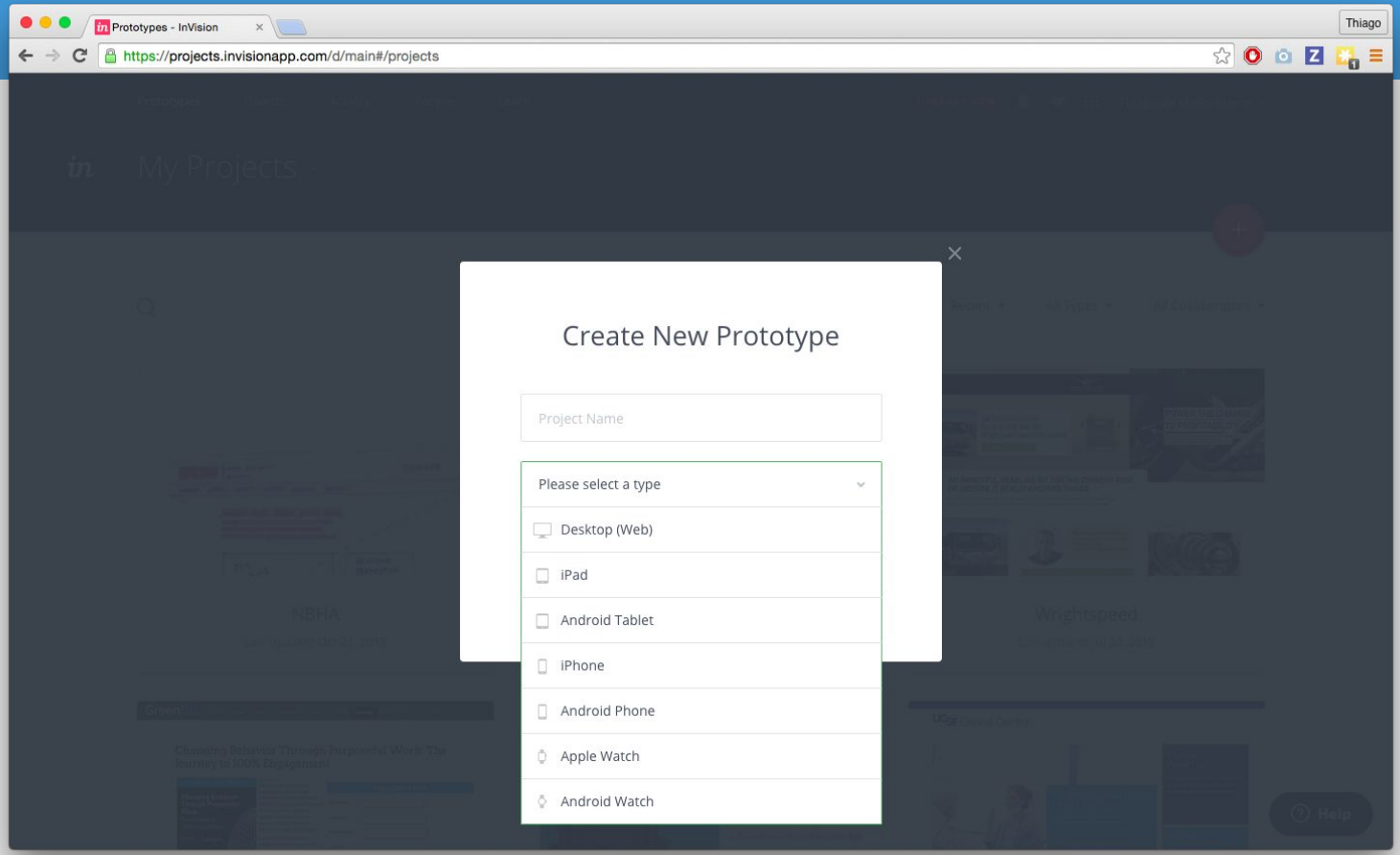
- ACTUALLY test early
- Get stakeholder feedback early
- Removed waterfall-effect from passing designs back and forth
- Avoid solving hard problems in Drupal if features aren’t awesome

Prototyping for the Web

Paper prototypes

KALAMUNA

Some prototypes are only responsive-ish



KALAMUNA

Let's use the native medium

http://alistapart.com/blog/post/15-years-of-dao

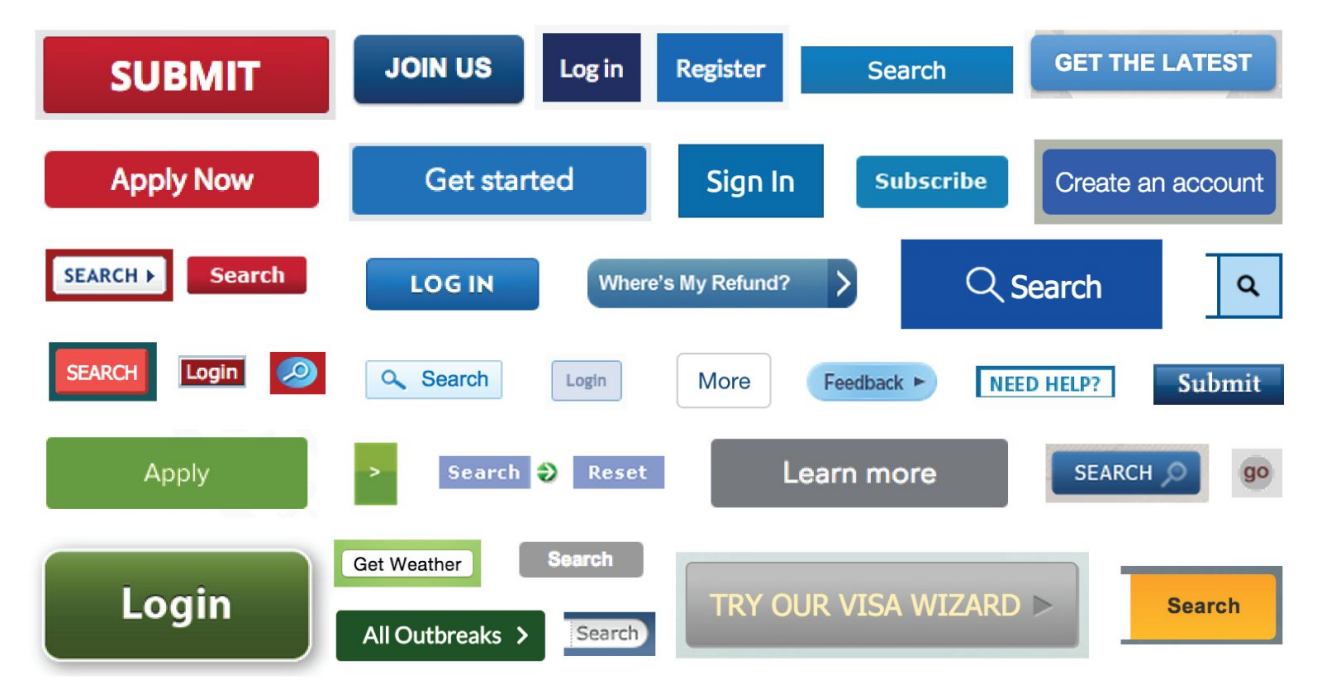
```
<!doctype html>
<html>
<head>
    ase href="/" />
    eta charset="utf-8">
    eta http-equiv="X-UA-Compatible" content="IE=edge">
    eta name="viewport" content="width=device-width, initial-scale=1">

    nk rel="shortcut icon" href="images/favicon.ico" type="image/vnd.
    k rel="apple-touch-icon" href="images/touch-icon.png">
    k rel="icon" href="images/favicon.png">
    [if IE]><link rel="shortcut icon" href="images/favicon.ico"><![e
     or, set /favicon.ico for IE10 win -->
     name="msapplication-TileColor" content="#f00">
     name="msapplication-TileImage" content="/images/tileicon.png">

  <title>
    Kalastatic
  </title>
```

KALAMUNA

Characteristics of a Solid Web Prototyping Solution

A Style Guide!

No guide = no style

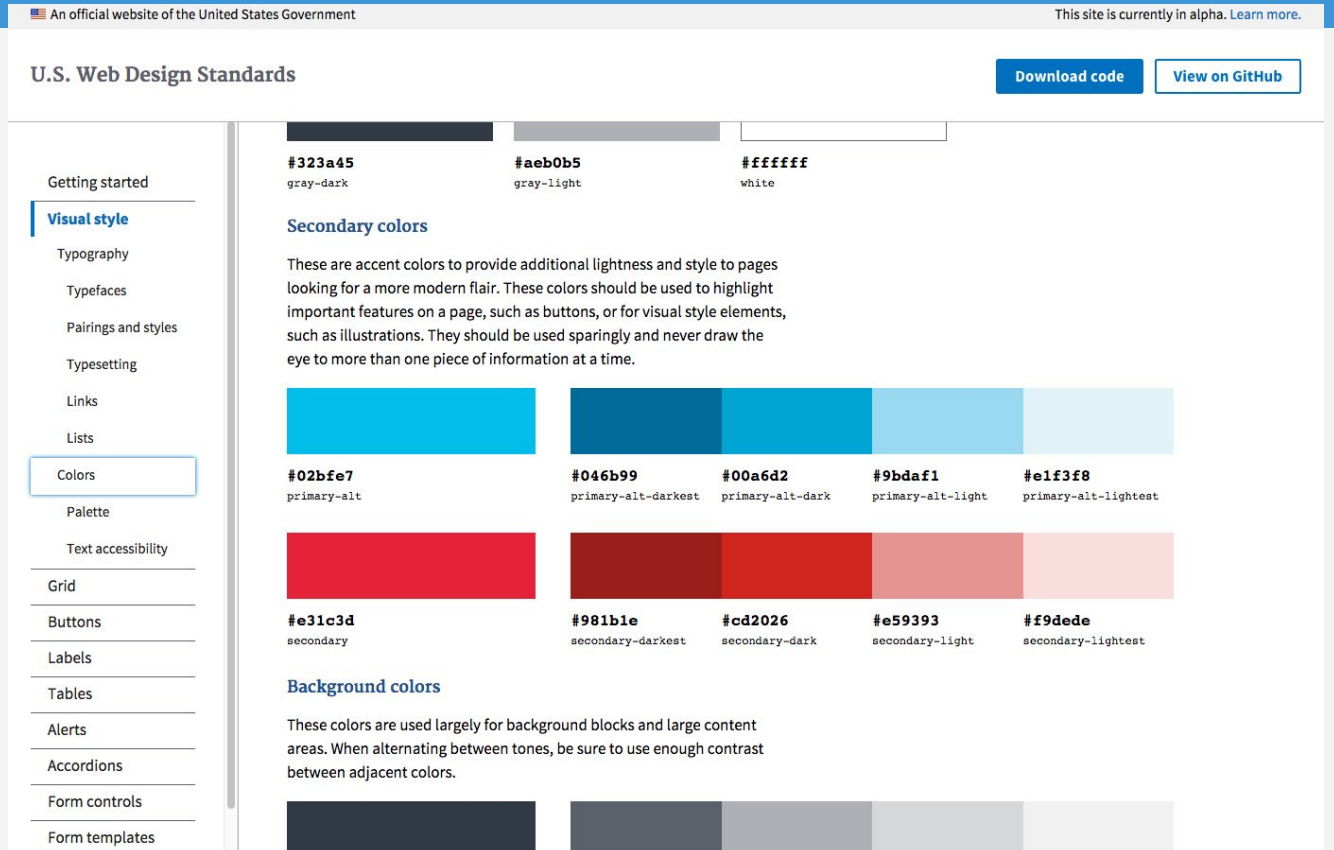


KALAMUNA

the new US standard (in alpha)

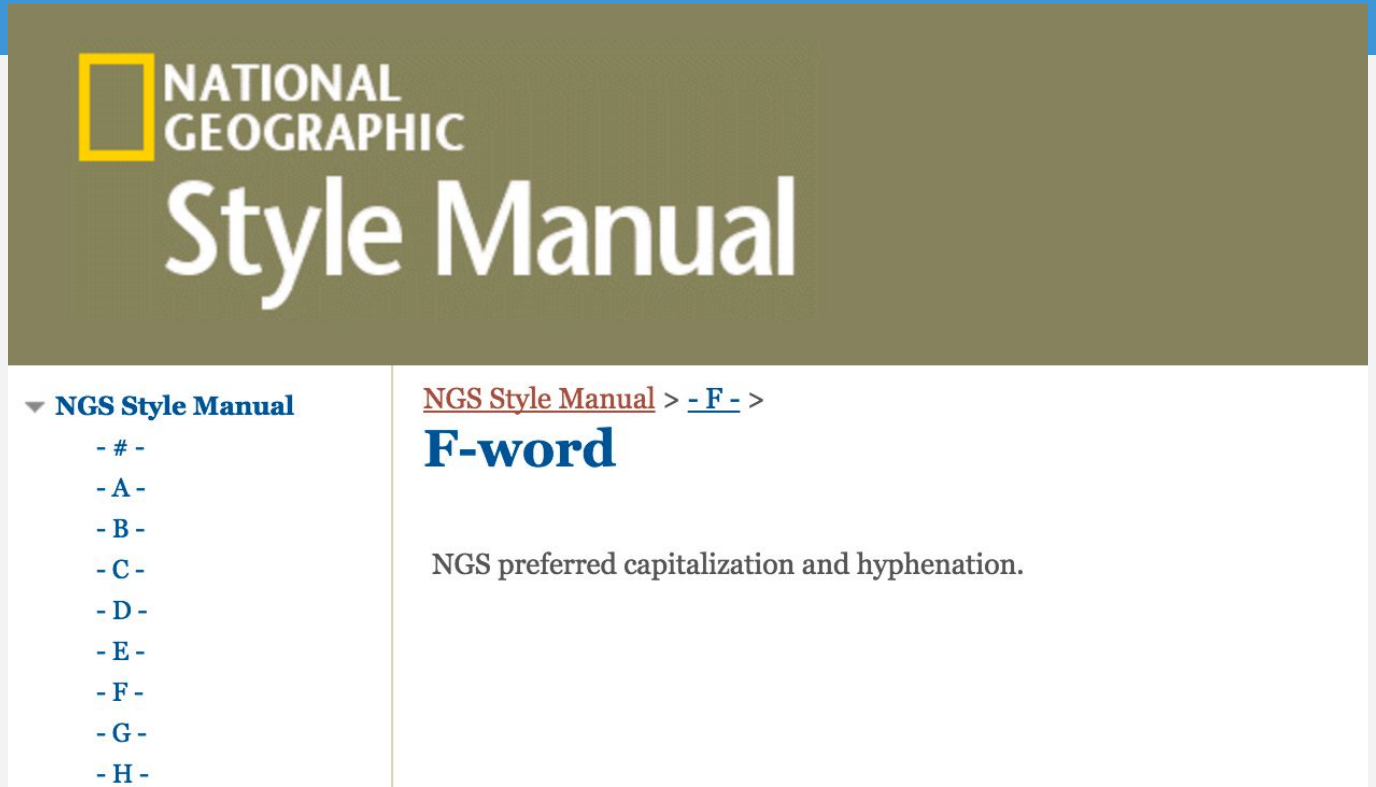
KALAMUNA

the US standard



KALAMUNA

Styleguides can also be about words



Component Architectures

Atoms

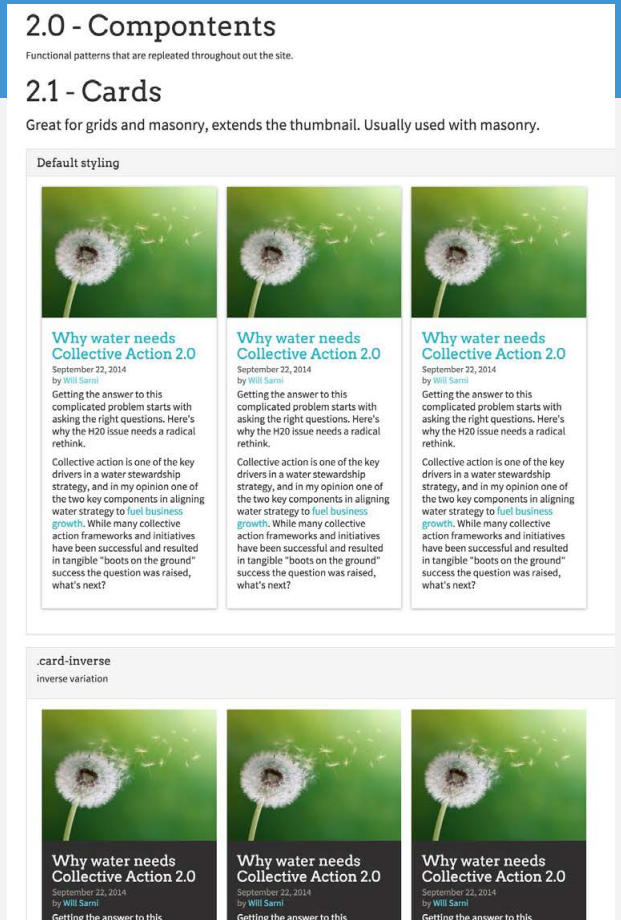
Brad Frost's "Atomic Design"

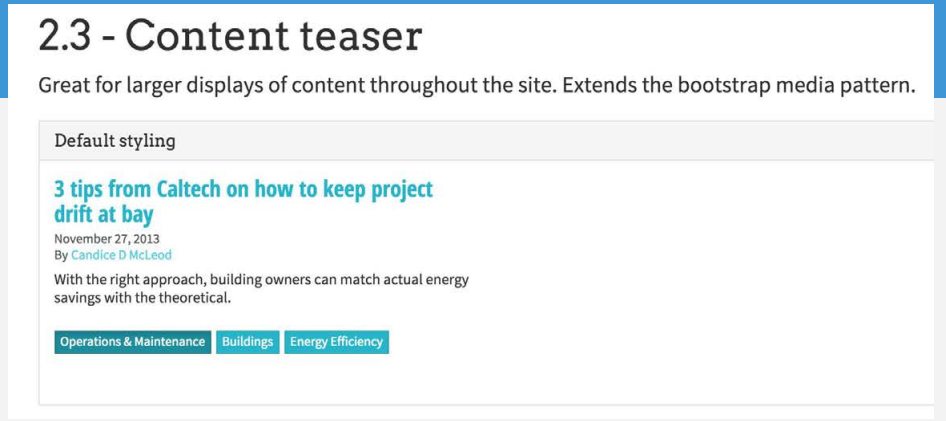
<input> <a> <h3> <h1 class="logo">

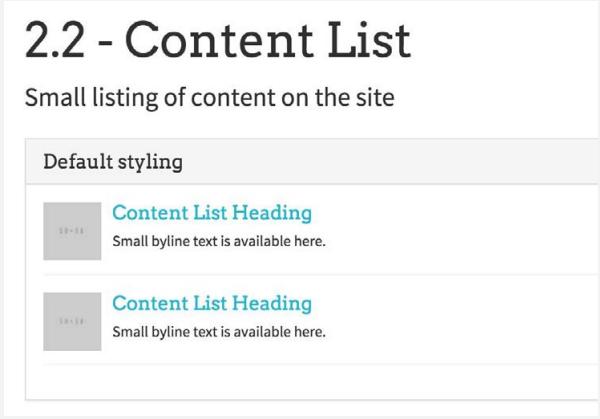
Molecule

Organism

Some Drupal-y Components





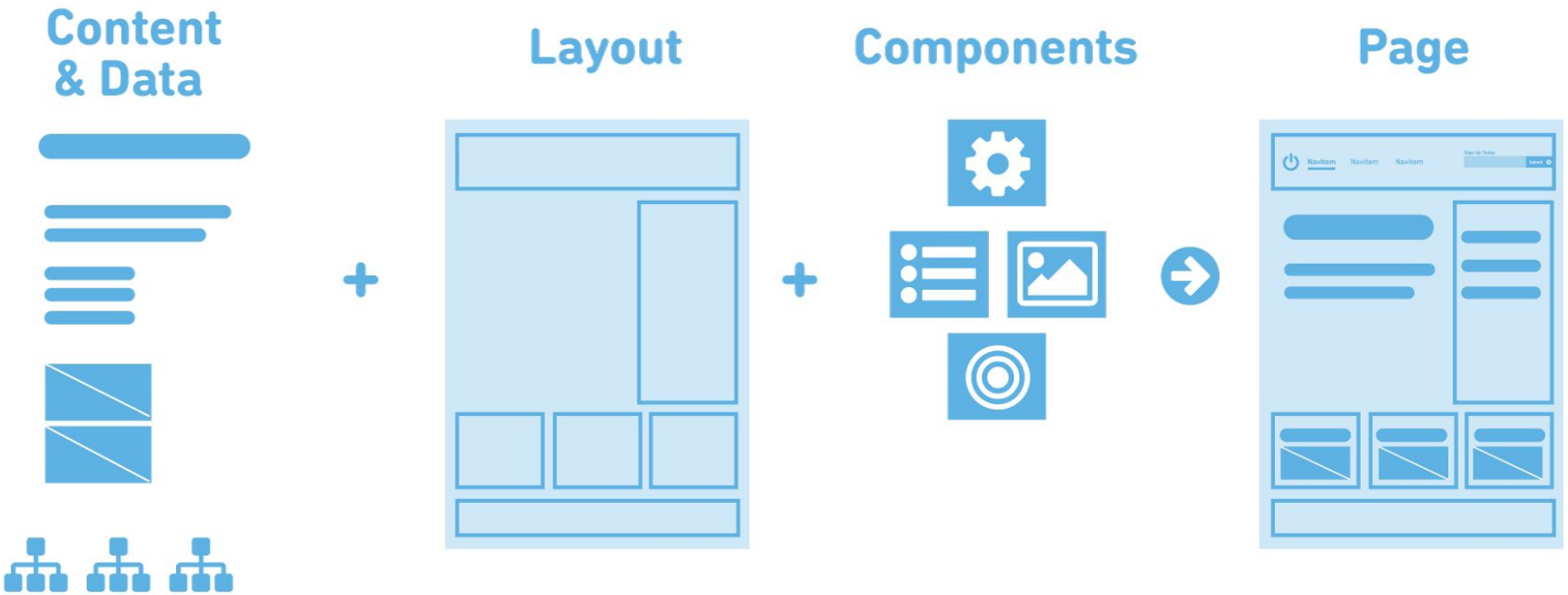


KALAMUNA

Why it Matters?

- Reusability, and consistency
- Markup needs assessed as planning and not afterthought
- Serve as a guide for future work
- Identify styling regressions as a project evolves
- Results in smaller style assets (DRY)

Layouts & Pages



- Organisms + Molecules: like blocks in a Layout.
- Add content to a layout and it is a Page

Content Workflows

Content

- Lorem ipsum dolar sit amet
- It’s not an afterthought!
- The site adapts to it (not vice versa)
- It has mass and takes up space (it matters)
- Updating content/images should be easy
- The Dean will love the design with his face in it

Testing

Tests

- Continuous Integration
- Automated Build Tests
- Linting (JS,HTML and CSS/SCSS)
- Accessibility Reports

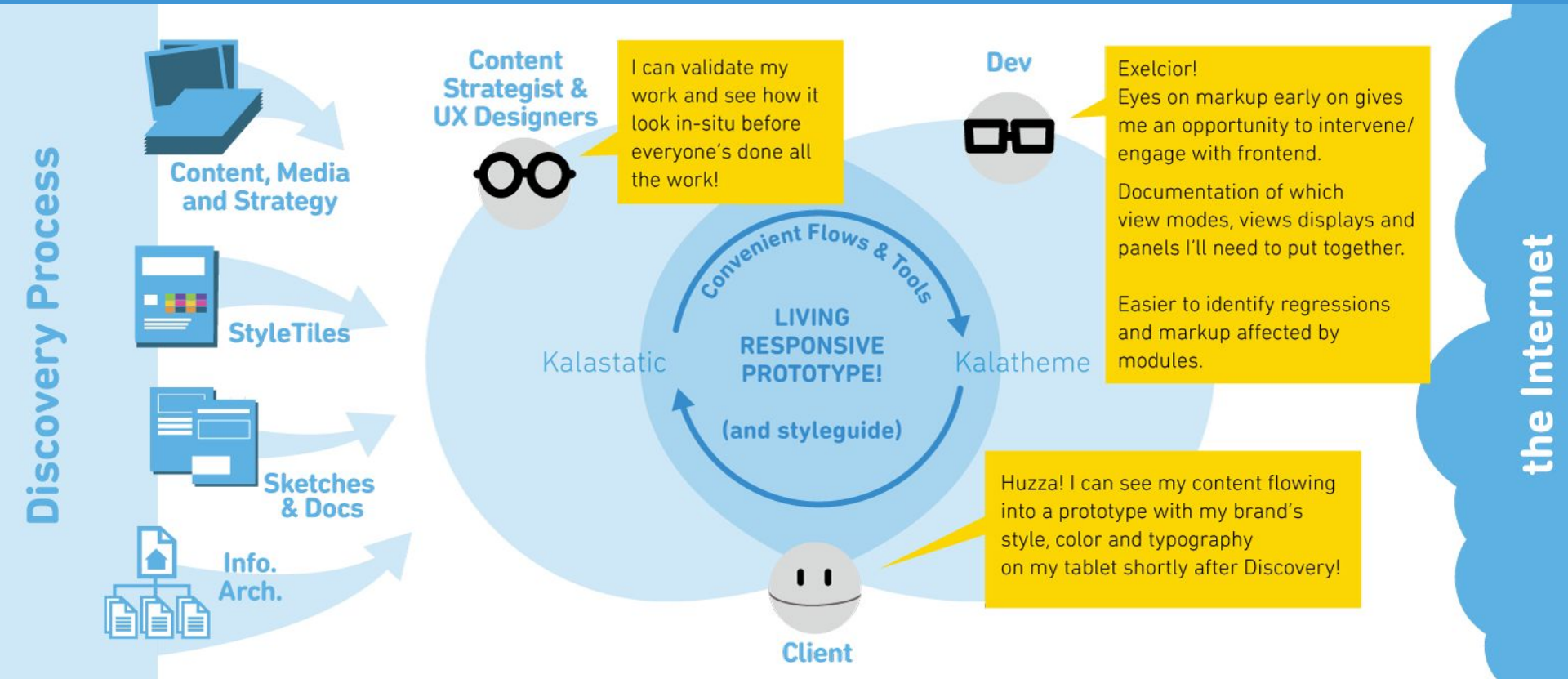
Welcome to Kalastatic

a static framework for prototyping
responsive sites, styleguides,
and designing in-browser.

Why?

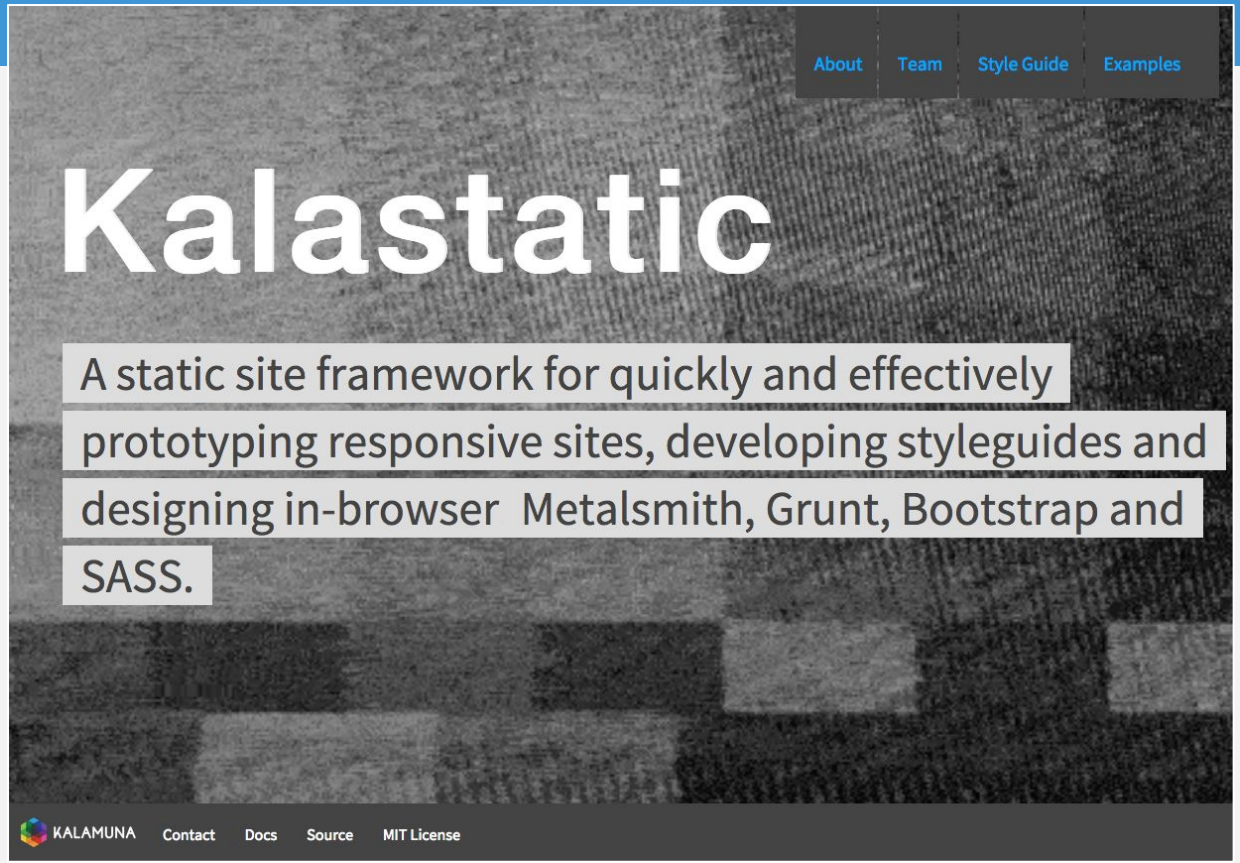
With Kalastatic we close the gap early between **visual design**, **content strategy** and **frontend development**.

Strategy, Design and Code – Together!



Prototyping Examples

http://test-kalastatic.at.kalamuna.com



KALAMUNA

kalastatic + kalatheme in love

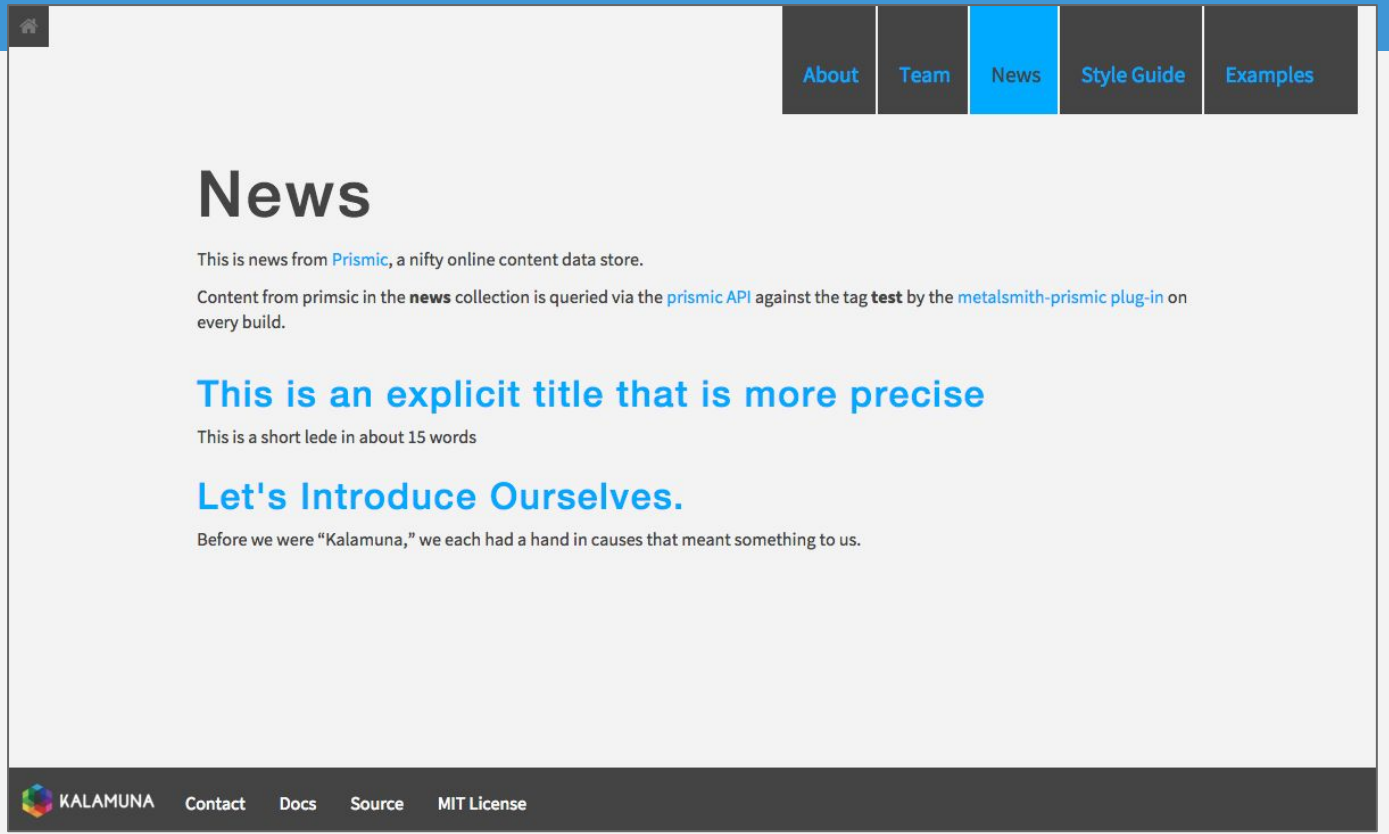
kalastatic imported into kalatheme (vs vanilla kalatheme)

> prototype in Drupal

> kalastatic.module configuration

Application Integrations

Prismic: pulling json into the prototype



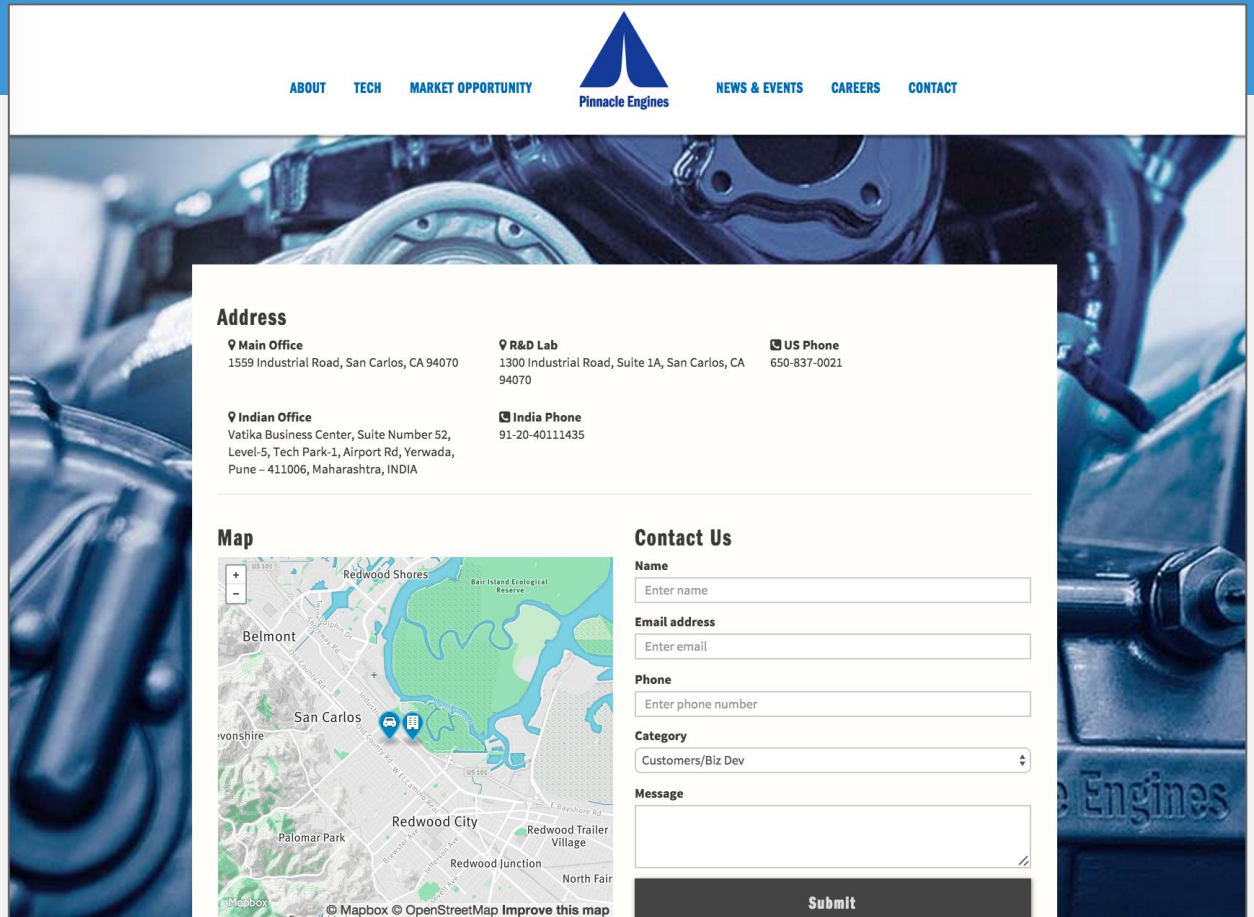
KALAMUNA

Drupal 7/8 integration: more decoupling!

- hal+json
- RSS

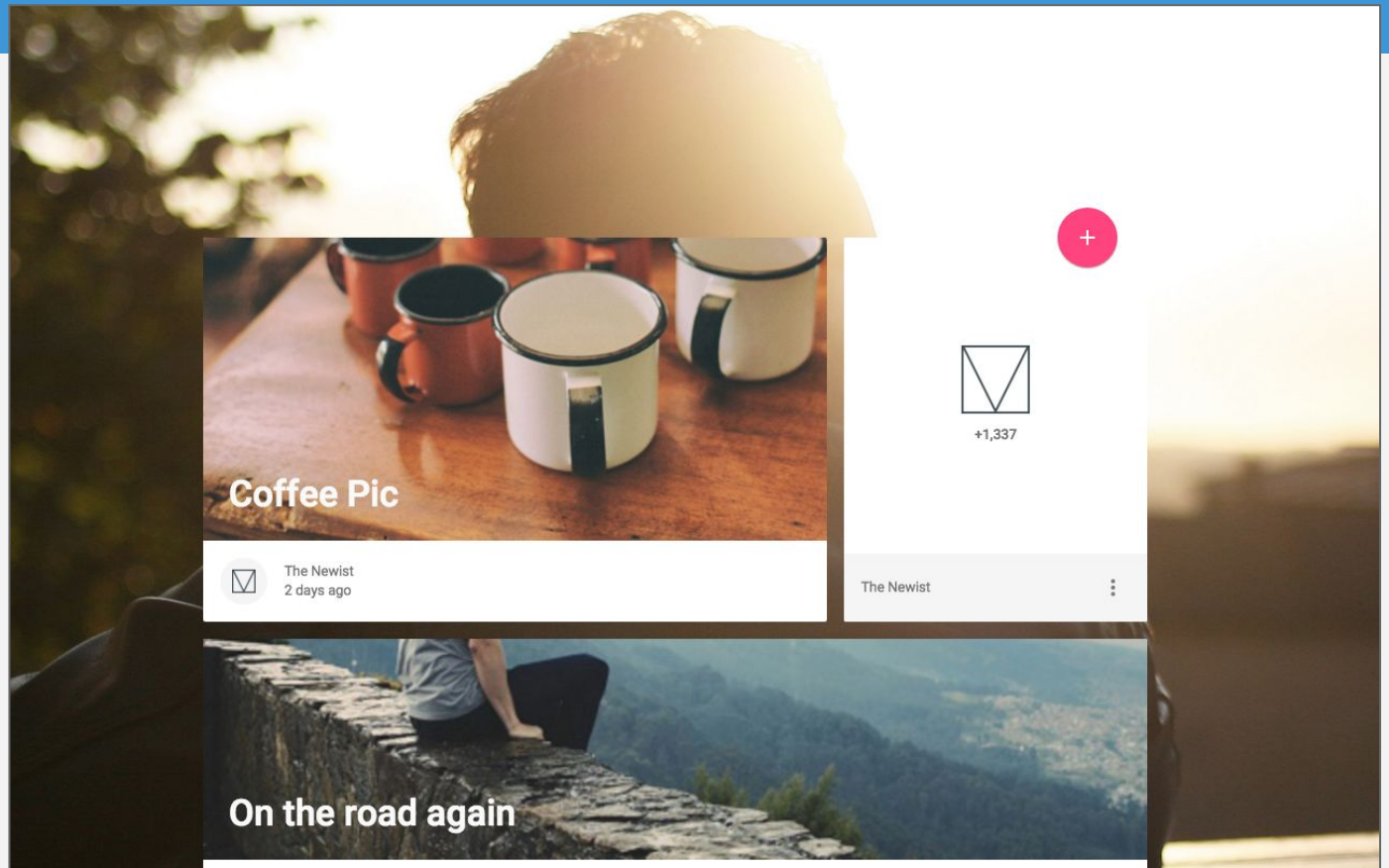
KALAMUNA

Pinnacle Engines: static site resources shared to Wordpress



KALAMUNA

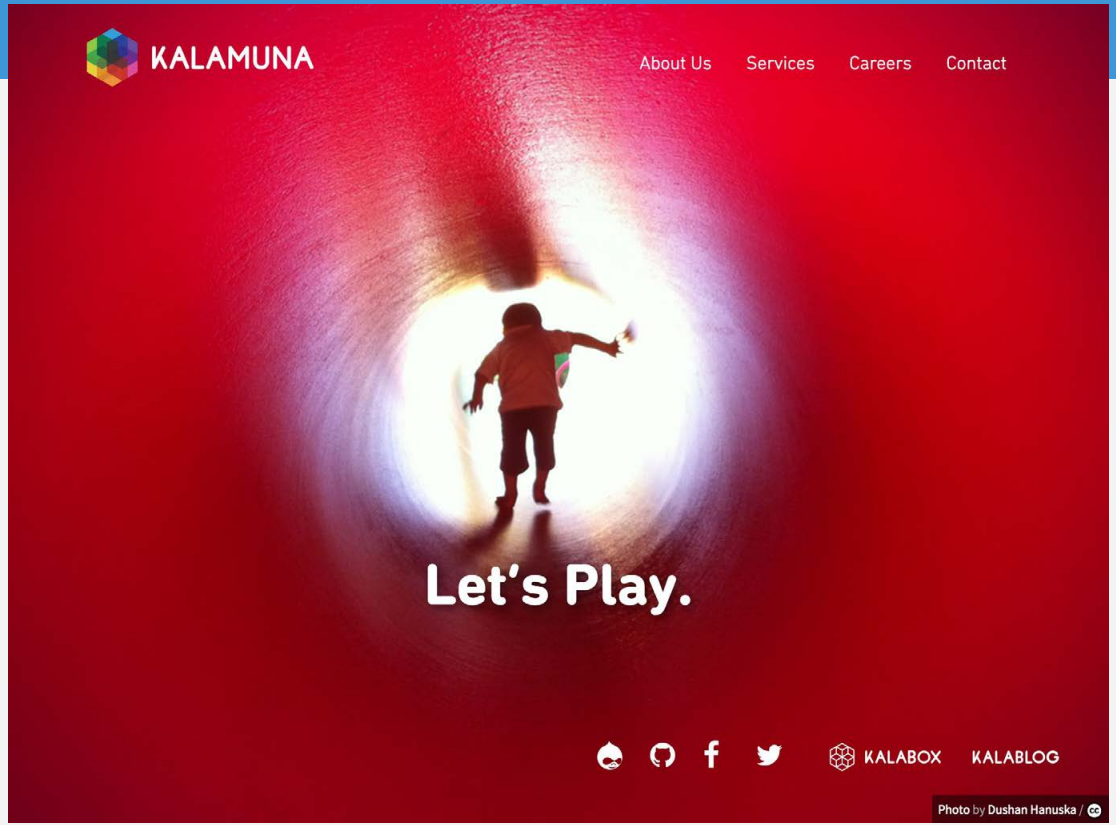
material design lite



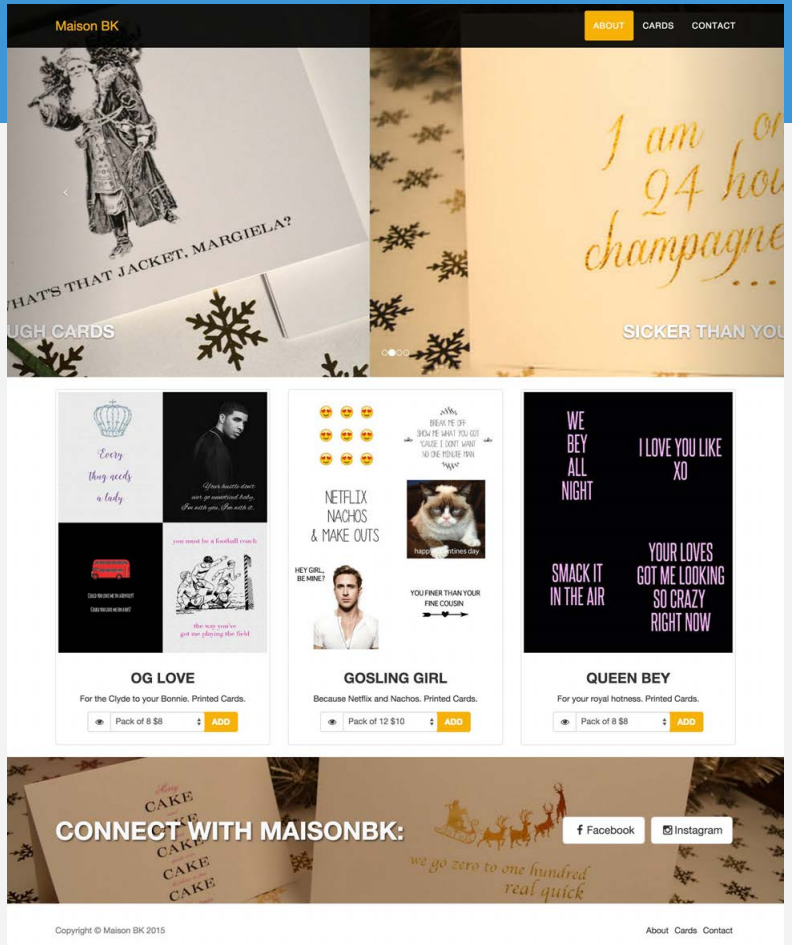
KALAMUNA

Examples in the wild

http://www.kalamuna.com

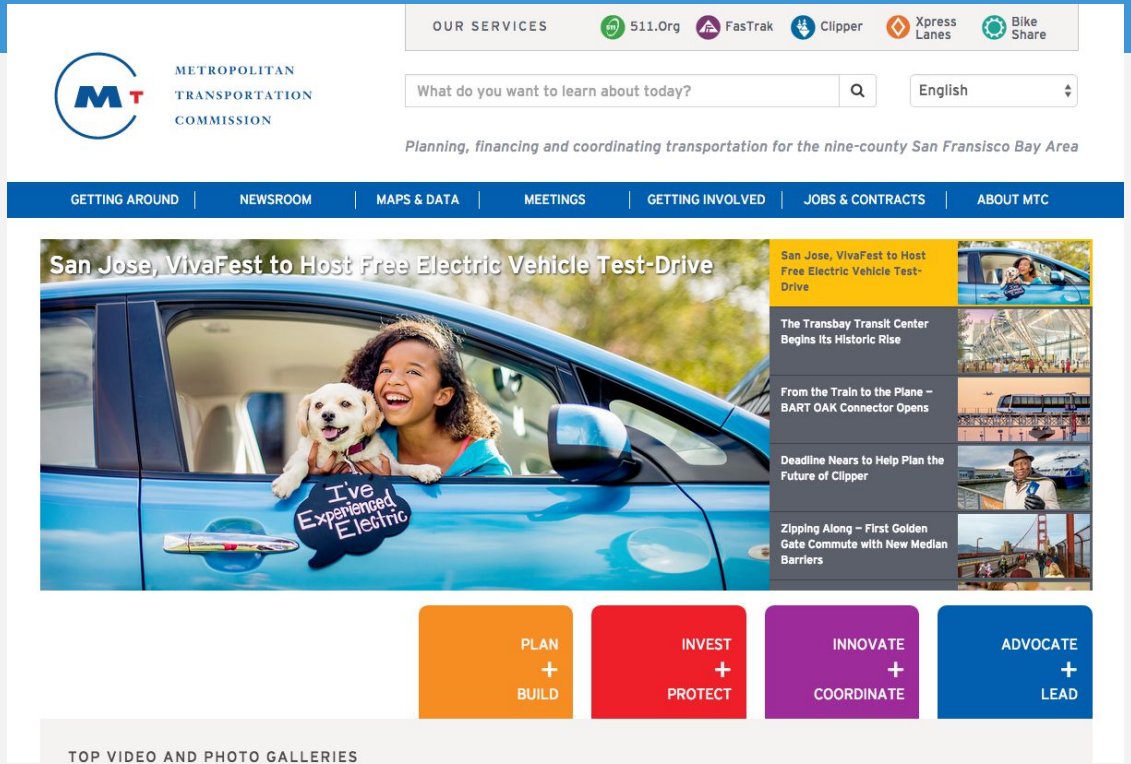


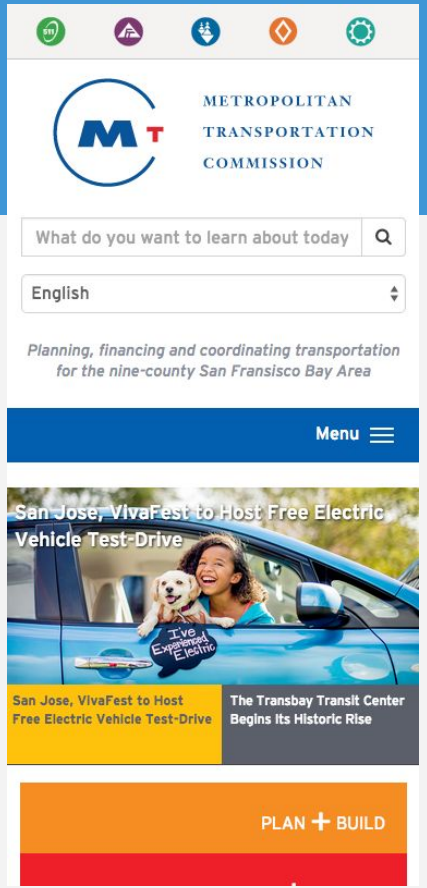
Maison BK: bootstrap prototype + e-commerce



KALAMUNA

Metropolitan Transportation Commission





KALAMUNA

THE FUTURE:

from blueprint to framework

Application Framework

github.com/kalamuna/kalastatic-cli

```
index.html.jade
styles.css.scss
scripts.js.jsmin.browserify
```

- Drush-like updating
- Support multiple templating engines
- multiple frameworks, like Google Material Design: http://mdl-blog-kalastatic.at.kalamuna.com

KALAMUNA

Thank You!

www.kalamuna.com

mallis@kalamuna.com
rob@kalamuna.com

Q & A

This section contains supplemental slides that speak more to the technologies that power kalastatic

Features

HOW is this different than patternlab?

- more advanced, modern (attractive) tools
  - no compass = node-sass
  - easy to ingest libraries
- no PHP dependencies
- Content strategy
- Flexible beyond styleguides
  - can ingest json
  - metalsmith can do anything!
  - can be used to make a real website!

KALAMUNA

Features

- Easy, single command installation
- Managed dependencies
- Built in webserver
- Livereload: Saving files reloads the browser
- Remote device access: Load your local on a mobile device!
- You can make a real site with this thing
- JSON ingestion from remote sources if you want it! (drupal 8?)

Features

CONVENIENCE UTILITIES

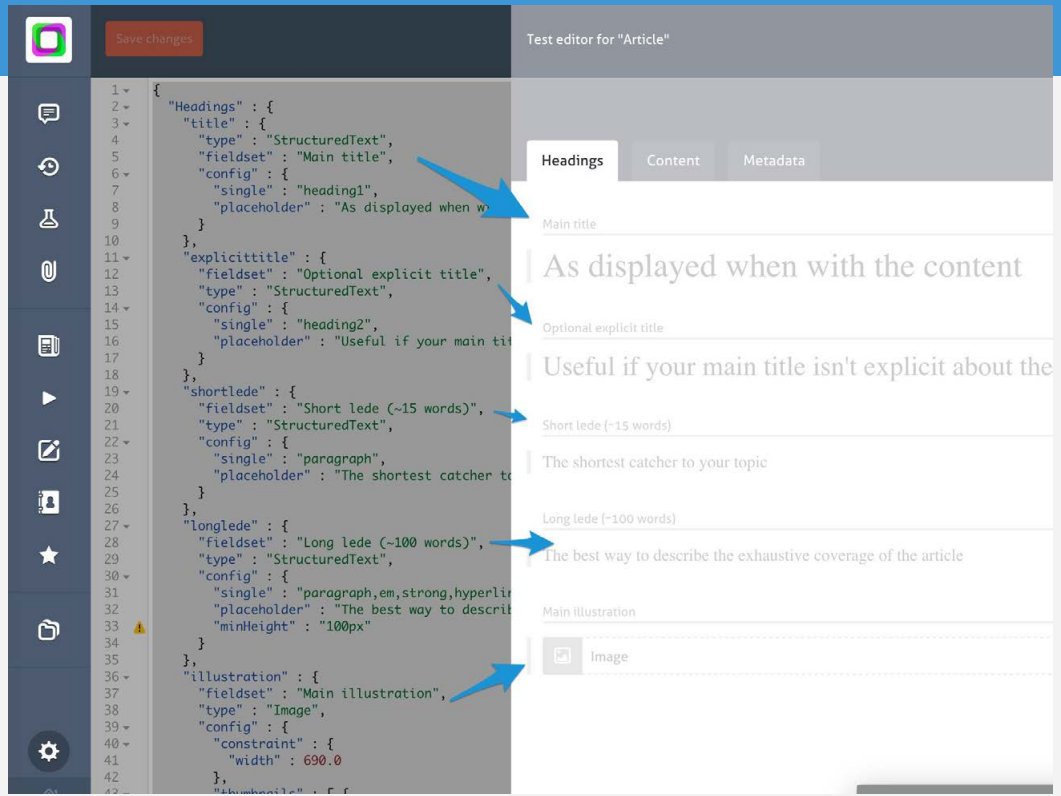
- Cache busting
- Deep linking (URL fragments)
- Character limit filters
- IE backward-compatibility

PRISMIC

Prismic – a head wearing masks

masks are like content types, but defined in json!

- edit article
- view content



KALAMUNA

“view” of stories of type **news** tagged **test**

/src/news.md

```
---
template: news.html
title: News
pageSlug: news
prismic:
  news:
    query: '[[:d = at(document.tags,["test"])]]'
---

This is news from [Prismic](https://kalastatic.prismic.io)!
```

article “teaser”

/templates/partials/news-teaser.html

```
<article class="teaser">
  <div class="wrapper">
    <a href="/article/{{id}}/{{slug}}">{{data.explicittitle.html|safe}}</a>
    {{data.shortlede.html|safe}}
  </div>
</article>
```

article “node”

/templates/news-article.html

```
{% extends 'html.html' %}

{% block contents %}

  {% for article in prismic.news.results %}
    <main role="main">
      <article>
        {{article.data.title.html|safe}}
        <figure class="article-img">
          {{article.data.illustration.html|safe}}
        </figure>
        </div>
        {{article.data.content.html|safe}}
      </article>
    </main>
  {% endfor %}

{% endblock %}
```

TECHNOLOGIES

Technologies

Dependency Management

We use npm for running tasks whenever it saves us from adding additional grunt dependencies.

Development Tools

Using Grunt allows us to run a local development environment with live reloading, as well as deployment.

Technologies

Front-End Dependencies

Bower downloads all of our frontend libraries:

- Bootstrap
- jQuery
- Holder.js
- and anything else we might need…

Technologies

Front-End Framework

- Most popular CSS/JS/HTML framework
- Responsive
- Large library of components
- 4.0 right around the corner!
  - SASS Native
  - Flexbox
  - Cards
  - etc

Technologies

CSS Pre-Processor

We use node-sass which is a wrapper for LibSass.

LibSass is written in C/C++ and as a result is much faster than RubySass.

This speeds up our workflows and removes any Ruby dependencies from our stack.

Technologies

JavaScript Pre-Processor

Compiling JavaScript for the browser is hard, so we use Browserify.

Browserify lets us `require('modules')` in the browser, and bundles up all the dependencies.

This speeds up our workflows and even speeds up browser JavaScript performance.

Technologies

METALSMITH

Templating

Extremely simple, pluggable, static site generator.

Takes source files, manipulates them via chainable plugins, and outputs results.

Metalsmith

Metalsmith

```
{
  "plugins": {
    "metalsmith-collections-convention": {},
    "metalsmith-metadata": {
    "config": "config.yaml",
    "navigation": "navigation.yaml"
    },
    "metalsmith-markdown": {},
    "metalsmith-include": {},
    "metalsmith-permalinks": {
    "pattern": ":url",
    "relative": false
    },
    "metalsmith-swig-helpers": {
```

Metalsmith - collections

A key Metalsmith plugin is metalsmith-collections

Collections are bundles of content that can be iterated through, they can be passed off to partials for rendering and allow for design patterns such as indexes, slideshows, and can allow individual bits of content to be used in multiple contexts.

Not unlike of like custom queries, and views.

Metalsmith

```
---
pattern: collections/news/*.md
sortBy: order
---

All news articles.
```

Custom template filters

Swig has straightforward html-yness.

We chose it due to it's relative similarities to **twig**, aligning us towards a D8 future.

It also supports **partials**, and **template inheritance** out of the box.

We rolled up some custom swig filters to get a bit of extra functionality like **email obfuscation**, **excerpts**, and **slug matching**.

We're probably switching to Mozilla's Nunjucks!

templating example

```
<section class="page-content">
    <h1>{{title}}</h1>
    <div class="contents">
        {{contents|safe}}
    </div>
</section>

<section id="news" class="news teasers">
  {% for news in collections.news %}
    {% include "./partials/news-teaser.html" with news only %}
  {% endfor %}
</section>
```

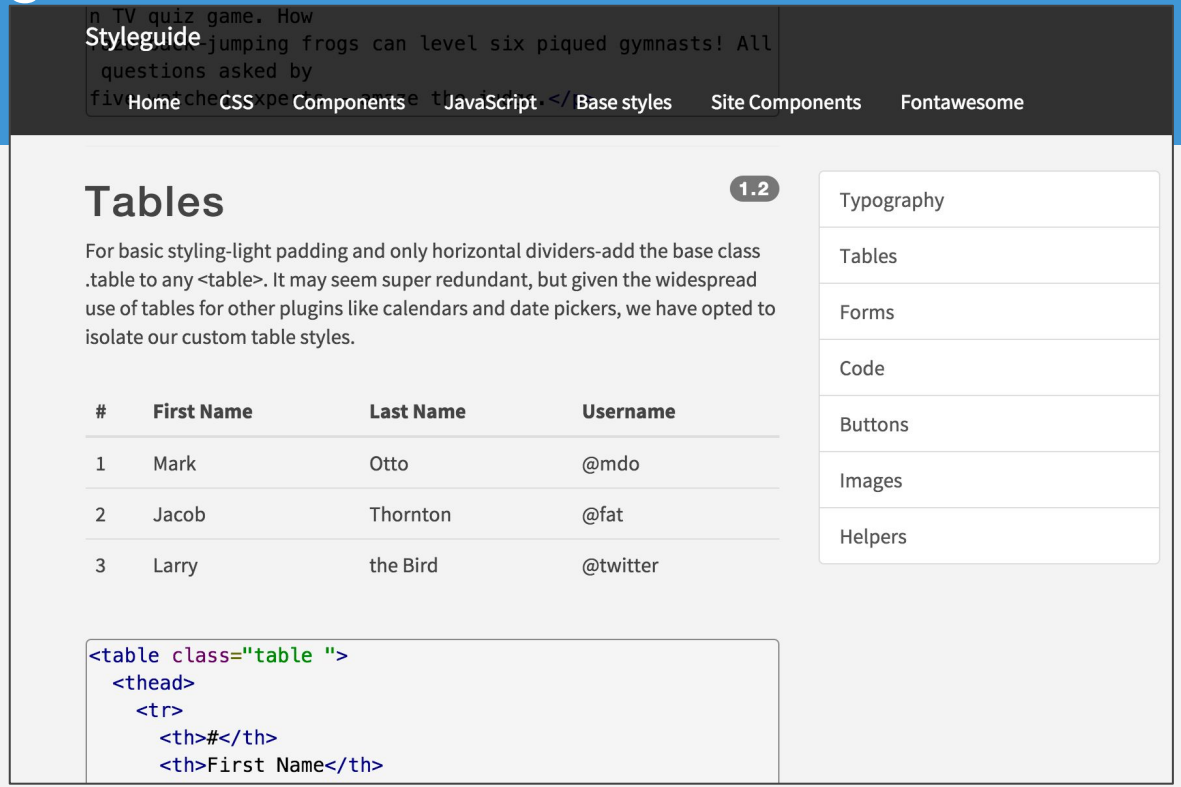
KSS

KSS

- Knyle Style Sheets is a CSS documentation specification that enables automated generation of ‘living’ style guides.
- It is **not** a preprocessor, CSS framework, naming convention, or specificity guideline.
- It works with any flavour of CSS—CSS, SCSS, LESS.
- We use kss-node, by Drupal’s own johnalbin.

KSS allows us to document our components style and markup, and easily identify/avoid front-end regressions.

KSS Styleguide



KALAMUNA

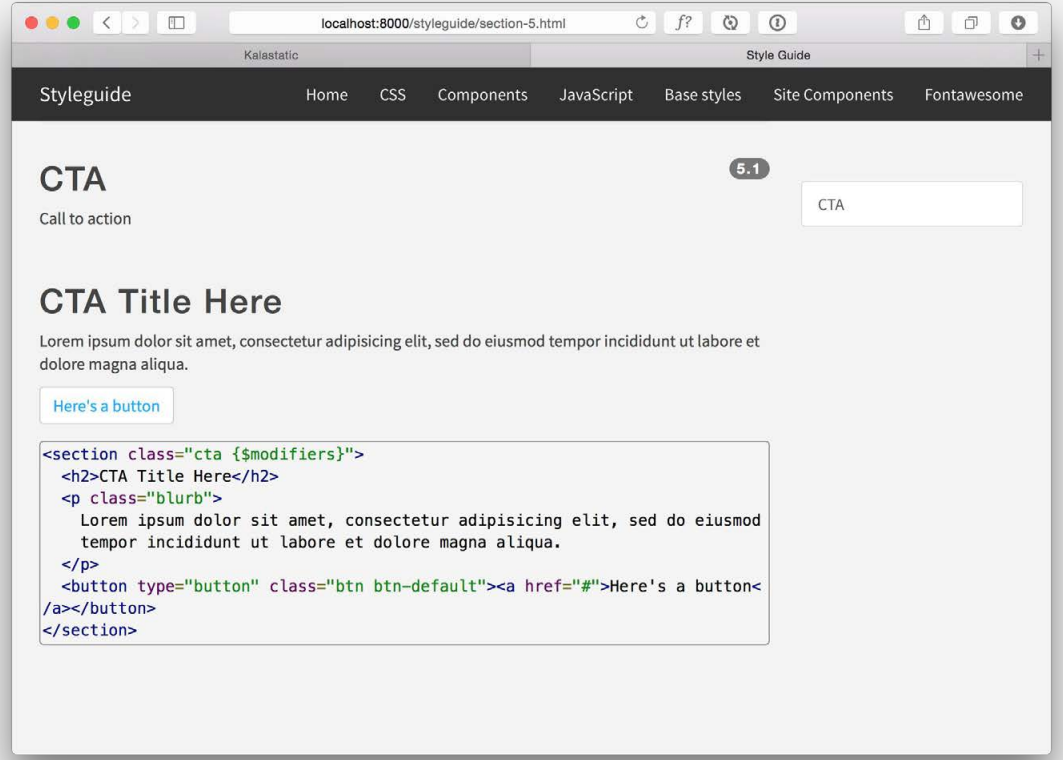
github.com/kalamuna/**kss-boostrap**

KSS

```
/*
CTA

Call to action

Markup:
<section class="cta {$modifiers}">
  <h2>CTA Title Here</h2>
  <p class="blurb">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua.
  </p>
  <button type="button" class="btn btn-default"><a href="#">Here's a button</a></button>
</section>

Styleguide 5.1
*/

.cta-wrapper {
  @extend .clearfix;
  clear: both;
  background: #dedede;
  .cta {
    text-align: center;
  }
}
```

KALAMUNA

KSS



KALAMUNA