

# How Plone Excels in GatsbyJS Content Mesh

Plone Conference 2019

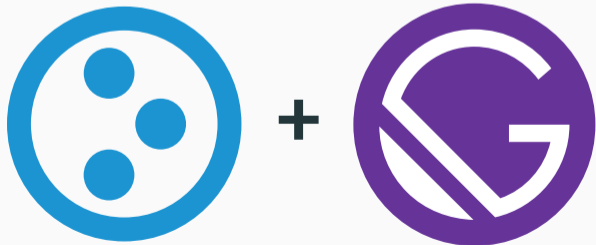
---

Asko Soukka

23.10.2019



JYVÄSKYLÄN YLIOPISTO  
UNIVERSITY OF JYVÄSKYLÄ



## Background

- Python developer since 2002
- Plone developer since 2004
- Full-time professional since 2008
- GSOC mentor since 2013
- GatsbyJS user since 2018



# University of Jyväskylä, Central Finland



# Demo



# GatsbyJS



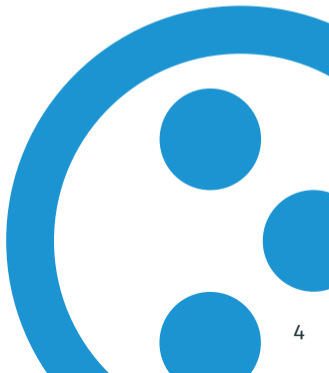
## GatsbyJS – the framework

- React based website and app generator
- GraphQL based CMS agnostic mashup framework
- Server-side rendered static deployment
- MIT-licensed open source plugin ecosystem



## GatsbyJS – the company

- Open source hobby turned into VC backed startup
- Copyrights owner, maintainer and evangelist
- Free tooling, documentation and webinars
- Paid cloud services with premium features



## Good

- "Blazingly-fast websites"
- Beginner friendly documentation
- Seamless developer experience
- Commitment on accessibility
- React ecosystem compatible





# Shortcomings

## Bad

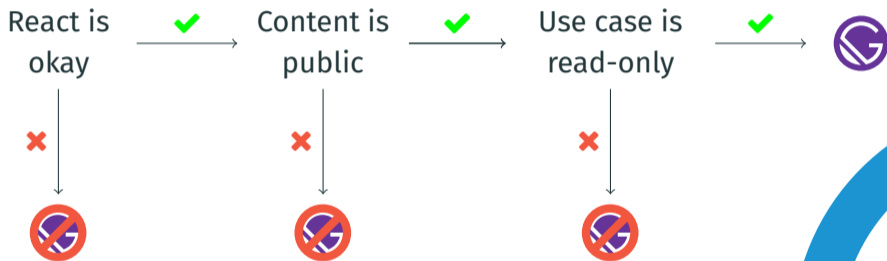
- No incremental builds
- Varying plugin quality
- GatsbyJS GitHub monorepo

## Ugly

- "Overly-complicated framework and pipeline"
- "Vendor lock-in for *just* updating static HTML"



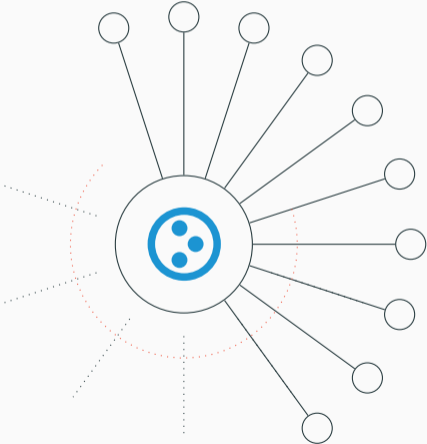
# Not for Everyone



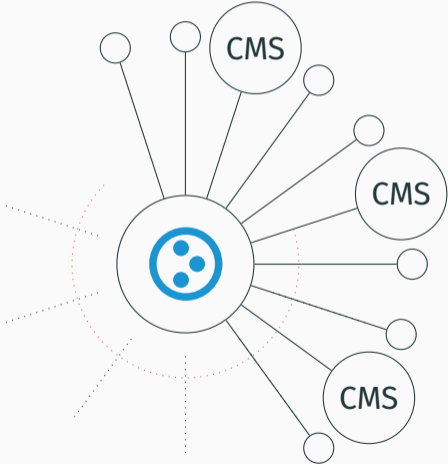
# Content Mesh



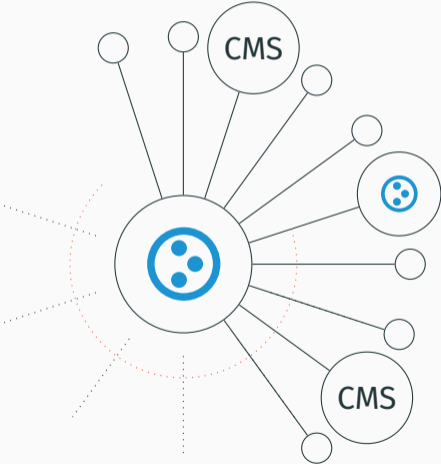
# CMS in Disruption



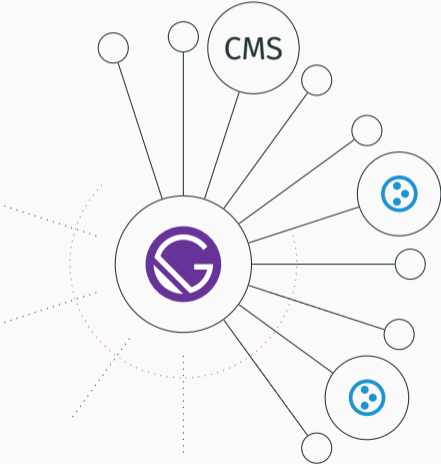
# CMS in Disruption



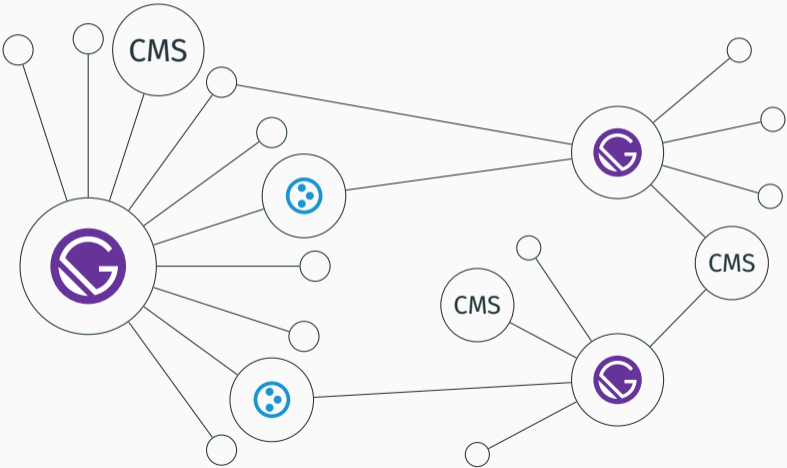
# CMS in Disruption



# CMS in Disruption



# Content Mesh





# Ecosystem



# Plugin Categories

SOURCE

# Plugin Categories

SOURCE

images

# Plugin Categories

SOURCE

TRANSFORMER

images

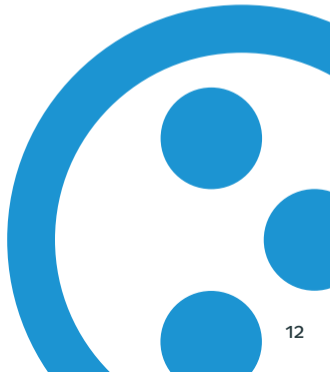
# Plugin Categories

**SOURCE**

**TRANSFORMER**

images

scales



# Plugin Categories

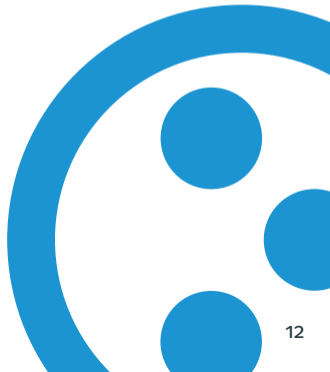
**SOURCE**

**TRANSFORMER**

**THEME**

images

scales



# Plugin Categories

**SOURCE**

**TRANSFORMER**

**THEME**

images

scales

gallery

**All plugins are equal**  
categories are not enforced



# Themes

---

## Experiment on maintainable boilerplate

- Packages with pre-configured components
- Configuration possible with plugin options
- Overridable or extendable by shadowing



## Experiment on maintainable boilerplate

- Packages with pre-configured components
- Configuration possible with plugin options
- Overridable or extendable by shadowing

JBOA

## Experiment on maintainable boilerplate

- Packages with pre-configured components
- Configuration possible with plugin options
- Overridable or extendable by shadowing

JBOA

???

## Experiment on maintainable boilerplate

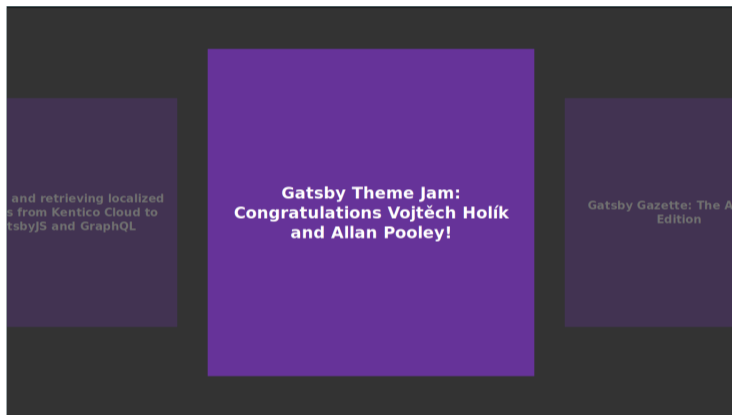
- Packages with pre-configured components
- Configuration possible with plugin options
- Overridable or extendable by shadowing

JBOA

???

PROFIT

## Example "theme" gatsby-theme-coverflow



[datakurre.github.io/gatsby-theme-coverflow](https://datakurre.github.io/gatsby-theme-coverflow)

# Example "theme" configuration

```
{
  resolve: "gatsby-theme-coverflow",
  options: {
    path: "gatsby-blog",
    query: `{ allCoverPages: allFeedGatsbyBlog {
      edges {
        node {
          title
          link
        }
      }
    } }`
  }
}
```

**Plone**





## Enablers

- Hierarcical content management
- Customizable content types
- Flexible publication workflows
- Effective and expiration dates
- High quality extensible REST API



## Curated use-cases

- Standalone spin-off web sites
- CMS part of GatsbyJS web apps
- Content configuration management



## Curated use-cases

- Standalone spin-off web sites
- CMS part of GatsbyJS web apps
- Content configuration management

## Seen challenges

- WYSIWYG → WYSIWYM
- Release management



## Done

- Incremental Plone content sourcing
- Selectable site root with `baseURL` option
- Support for rich text, images and files
- Rich text to React w/ Gatsby links and images
- GraphQL queries by `_path` or by hierarchy
- Websocket-ready for instant `develop` updates

## To do

- Support custom rich text, image and file fields

*Issue:* Only fields `text`, `image` and `file` are supported

- Path prefixing or namespacing support for multiple plugin instances

*Issue:* Instances may override each others' nodes



# Get Started

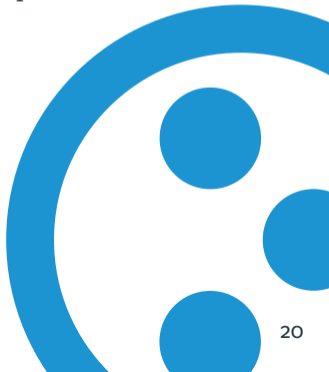


## Create a new project

```
$ npx gatsby new my-gatsby-project
$ cd my-gatsby-project
$ yarn add git+https://github.com/collective/gatsby-source-plone
```

## Add a Plone source in ./gatsby-config.js with

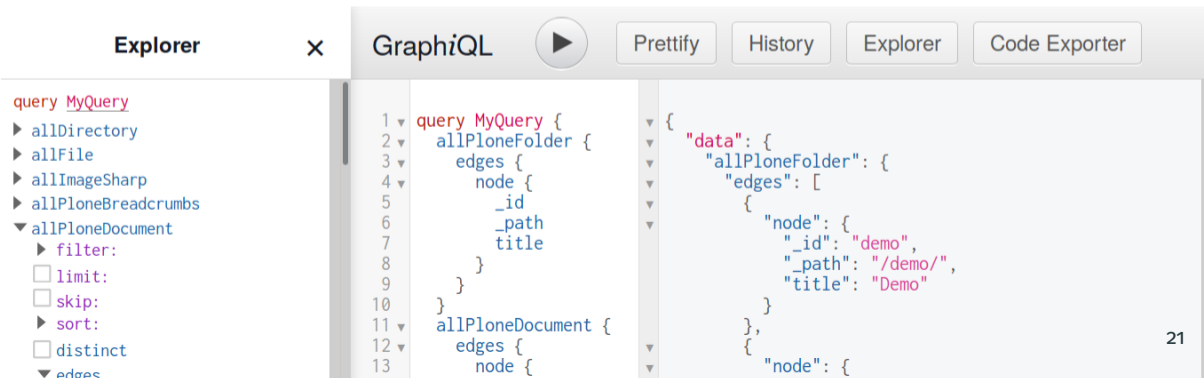
```
plugins: [
  { resolve: 'gatsby-source-plone',
    options: {
      baseUrl: 'https://plonedemo.kitconcept.com/en',
    },
  },
],
```



## Start GatsbyJS development server

```
$ yarn gatsby develop
```

**Open browser at** [http://localhost:8000/\\_\\_\\_graphql](http://localhost:8000/___graphql)



The screenshot shows the GraphQL Playground interface. On the left is the Explorer panel with a tree view of the schema. The main area is split into two panes: the query editor on the left and the JSON response on the right. The query is a GraphQL query for a 'demo' document. The response is a JSON object with a 'data' field containing the document details.

**Explorer** × **GraphiQL** [Play] [Prettify] [History] [Explorer] [Code Exporter]

```
query MyQuery {
  allDirectory
  allFile
  allImageSharp
  allPloneBreadcrumbs
  allPloneDocument {
    filter:
    limit:
    skip:
    sort:
    distinct
    edges
  }
}
```

```
1 query MyQuery {
2   allPloneFolder {
3     edges {
4       node {
5         _id
6         _path
7         title
8       }
9     }
10  }
11 allPloneDocument {
12   edges {
13     node {
```

```
{
  "data": {
    "allPloneFolder": {
      "edges": [
        {
          "node": {
            "_id": "demo",
            "_path": "/demo/",
            "title": "Demo"
          }
        }
      ]
    },
    {
      "node": {
```



## Learn GatsbyJS

- [gatsbyjs.org/tutorial](https://gatsbyjs.org/tutorial)
- [collective.github.io/gatsby-source-plone](https://collective.github.io/gatsby-source-plone)

## Ask for help

- [github.com/collective/gatsby-source-plone/issues](https://github.com/collective/gatsby-source-plone/issues)



## Web site examples

- [github.com/collective/gatsby-source-plone](https://github.com/collective/gatsby-source-plone)
- [github.com/collective/gatsby-starter-plone](https://github.com/collective/gatsby-starter-plone)

## Landing page example

- [github.com/datakurre/gatsby-starter-plone-brochure](https://github.com/datakurre/gatsby-starter-plone-brochure)

## Content configuration example

- [github.com/datakurre/gatsby-theme-ds-player](https://github.com/datakurre/gatsby-theme-ds-player)



[datakurre.github.io/ploneconf2019](https://datakurre.github.io/ploneconf2019)