
Gatsby and GraphQL Schema Stitching with Drupal

JOHN ALBIN WILKINS / 18 JULY 2019



DECOUPLED
DAYS
2019

ABOUT ME

JOHN ALBIN WILKINS // Senior Front-end Developer and
Breaker of Designs that Expect Short Content

john.albin@amazeelabs.com

drupal.org/u/johnalbin

@JohnAlbin



**DECOUPLED
DAYS
2019**



TODAY'S DEMO CODE

MOSTLY DONE

English Español

 Search by keyword, ingredient, dish

Search

Log in

umami
FOOD MAGAZINE

Home

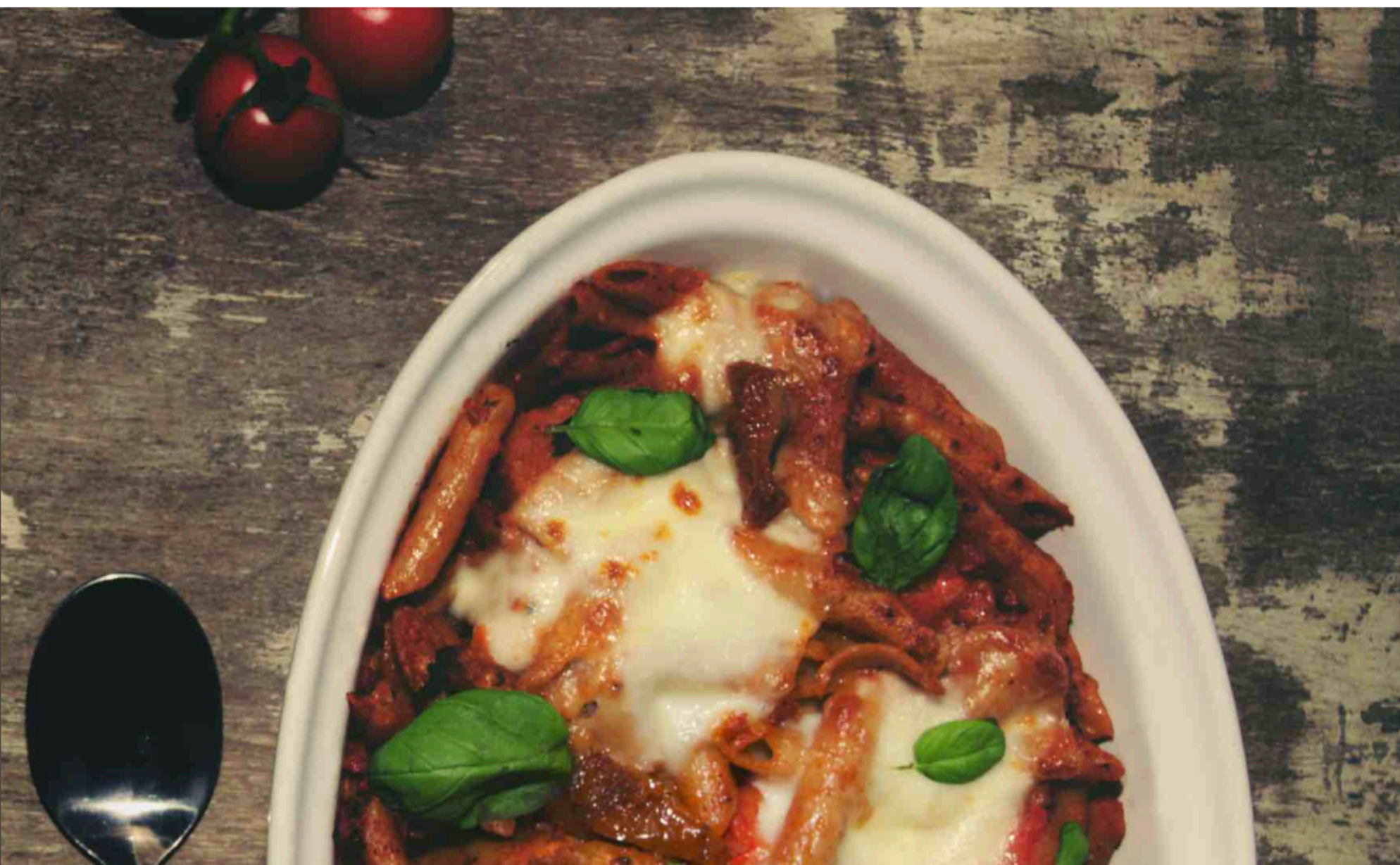
Articles

Recipes

Super easy vegetarian pasta bake

A wholesome pasta bake is the ultimate comfort food. This delicious bake is super quick to prepare and an ideal midweek meal for all the family.

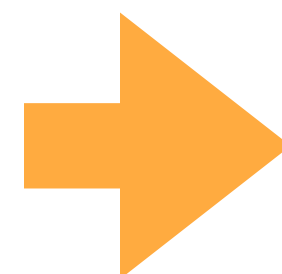
[View recipe](#)



TODAY'S DEMO CODE

MOSTLY DONE

http:///examples/using-drupal



MY GOAL:

http:///examples/using-drupal-graphql

Search by keyword, ingredient, dish Login

Umami
Food Magazine

Recipes



English Español

Search by keyword, ingredient, dish

Search

Log in

umami
FOOD MAGAZINE

Home

Articles

Recipes

Super easy vegetarian pasta bake

A wholesome pasta bake is the ultimate comfort food. This delicious bake is super quick to prepare and an ideal midweek meal for all the family.

[View recipe](#)



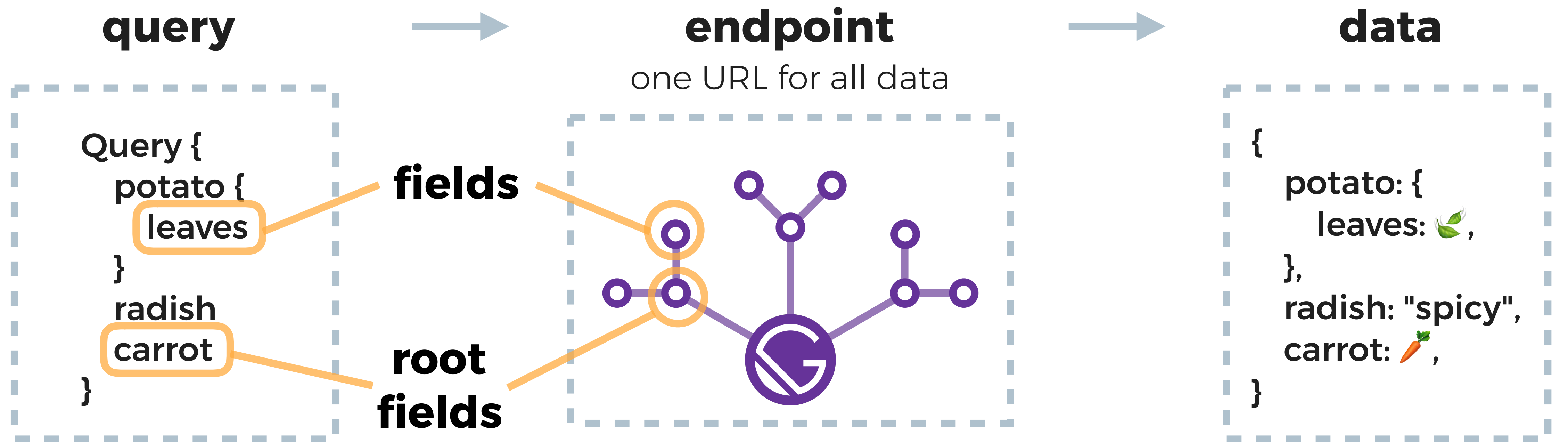
AGENDA

TALKING POINTS

- 1 GraphQL terminology
- 2 Why GraphQL and why schema stitching?
- 3 Configuring Gatsby + Drupal GraphQL
- 4 LIVE DEMO: Drupal GraphQL queries
- 5 Multilingual and Routing tips



GraphQL TERMINOLOGY

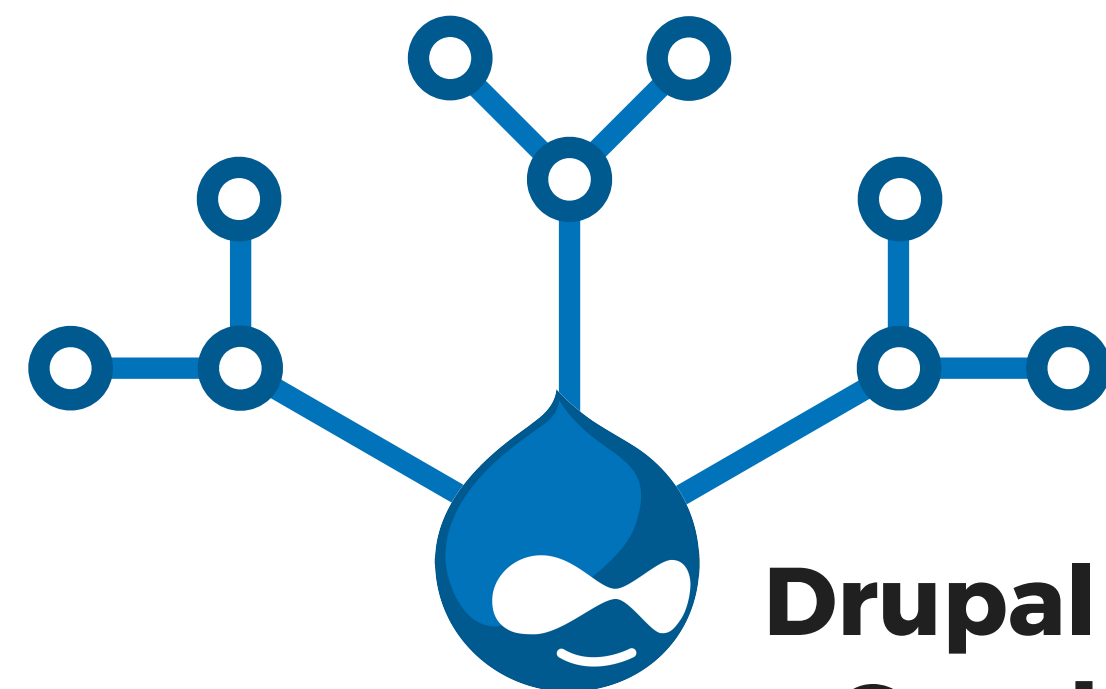
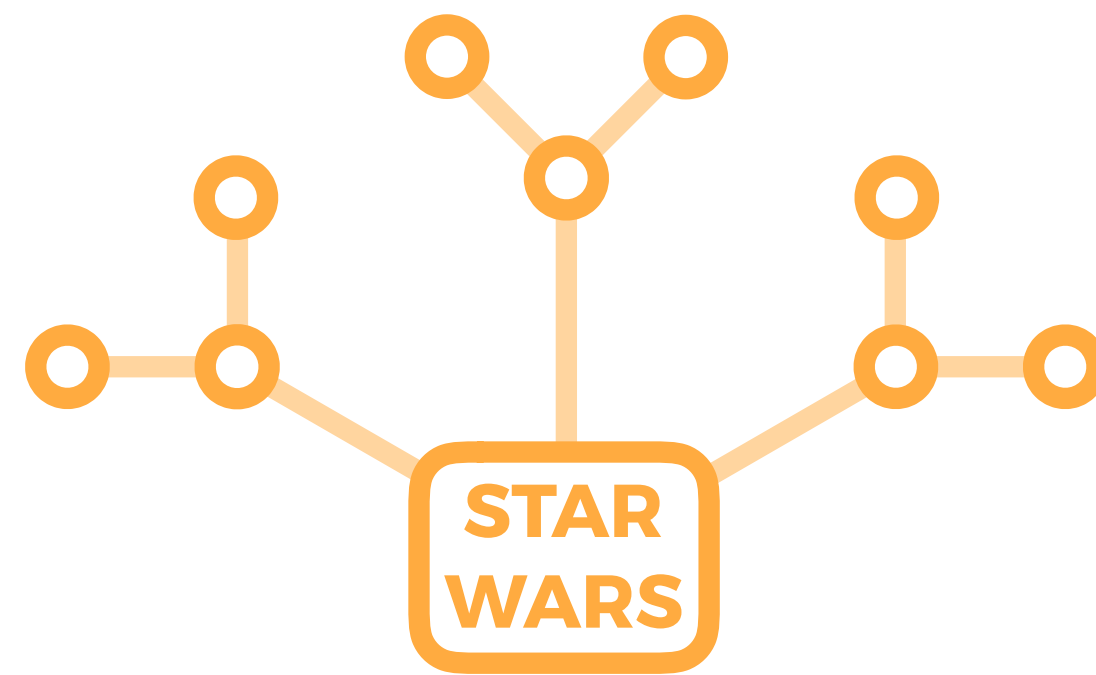
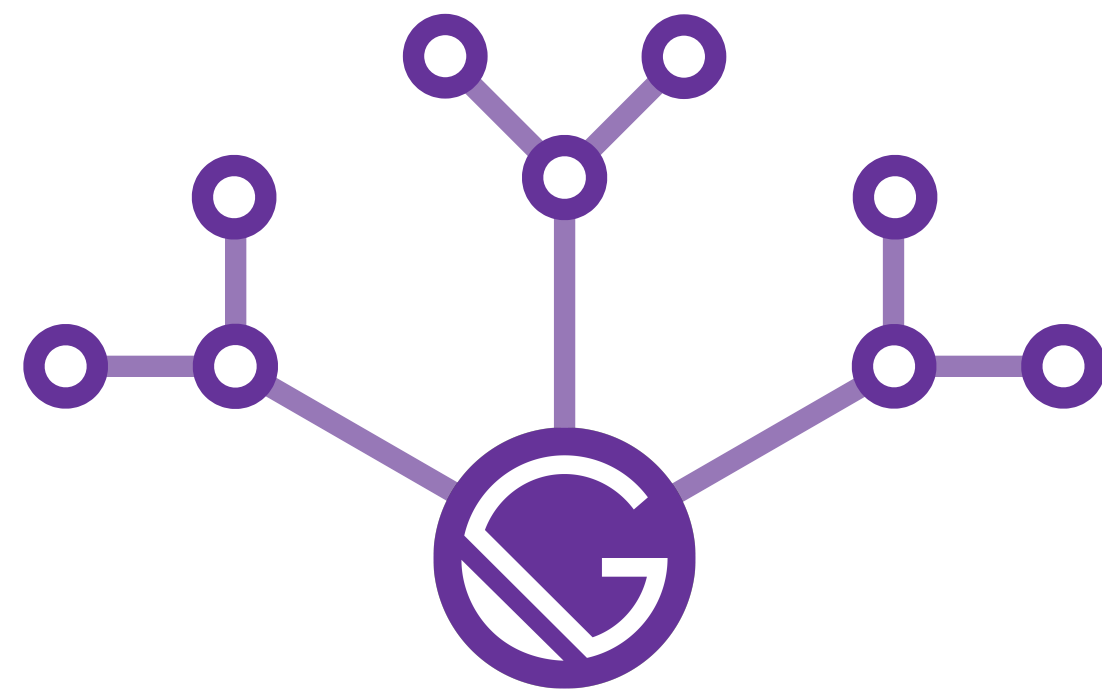


schema: description of all *fields* and their *types*

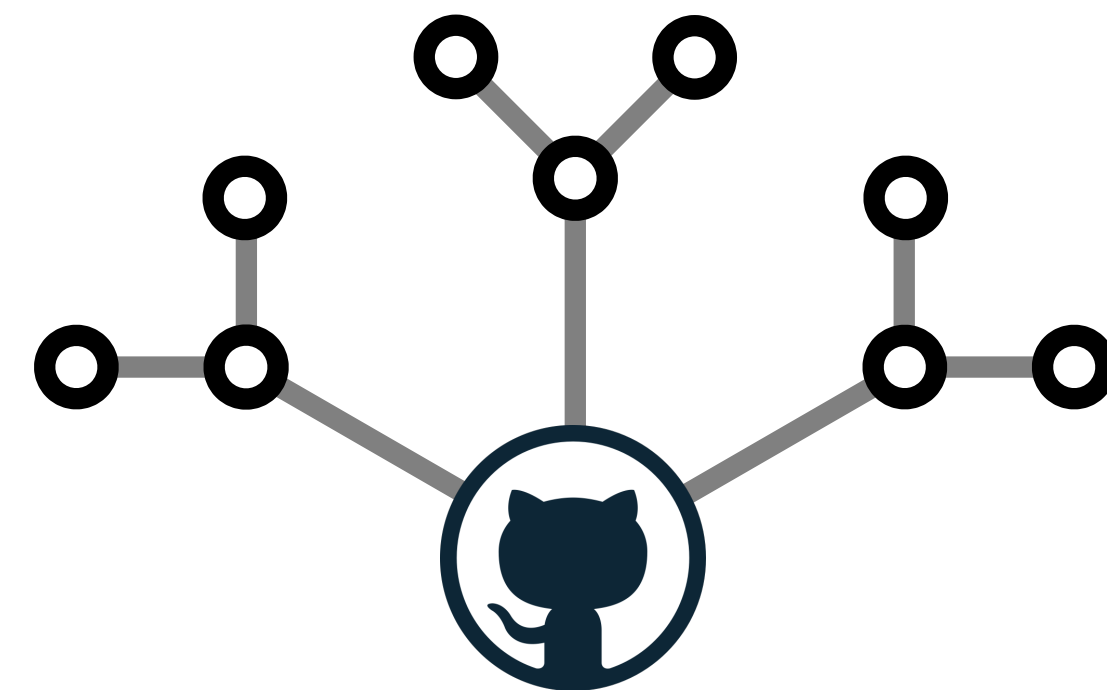
type: strong, strict definition of data

GraphQL POPULARITY

Just the new shiny?



**Drupal core
+ GraphQL module**



**GraphQL
is more than
an endpoint**



What makes GraphQL special?

REST 2.0

Example of JSON:API endpoints:

jsonapi:	{...}
data:	[]
links:	
action--action:	
href:	<i>"https://example.com/jsonapi/action/action"</i>
base_field_override--base_field_override:	
href:	<i>"https://example.com/jsonapi/base_field_override/base_field_override"</i>
block--block:	
href:	<i>"https://example.com/jsonapi/block/block"</i>
block_content--banner_block:	
href:	<i>"https://example.com/jsonapi/block_content/banner_block"</i>
block_content--basic:	
href:	<i>"https://example.com/jsonapi/block_content/basic"</i>
block_content--disclaimer_block:	
href:	<i>"https://example.com/jsonapi/block_content/disclaimer_block"</i>
block_content--footer_promo_block:	
href:	<i>"https://example.com/jsonapi/block_content/footer_promo_block"</i>
block_content_type--block_content_type:	
href:	<i>"https://example.com/jsonapi/block_content_type/block_content_type"</i>
configurable_language--configurable_language:	
href:	<i>"https://example.com/jsonapi/configurable_language/configurable_language"</i>



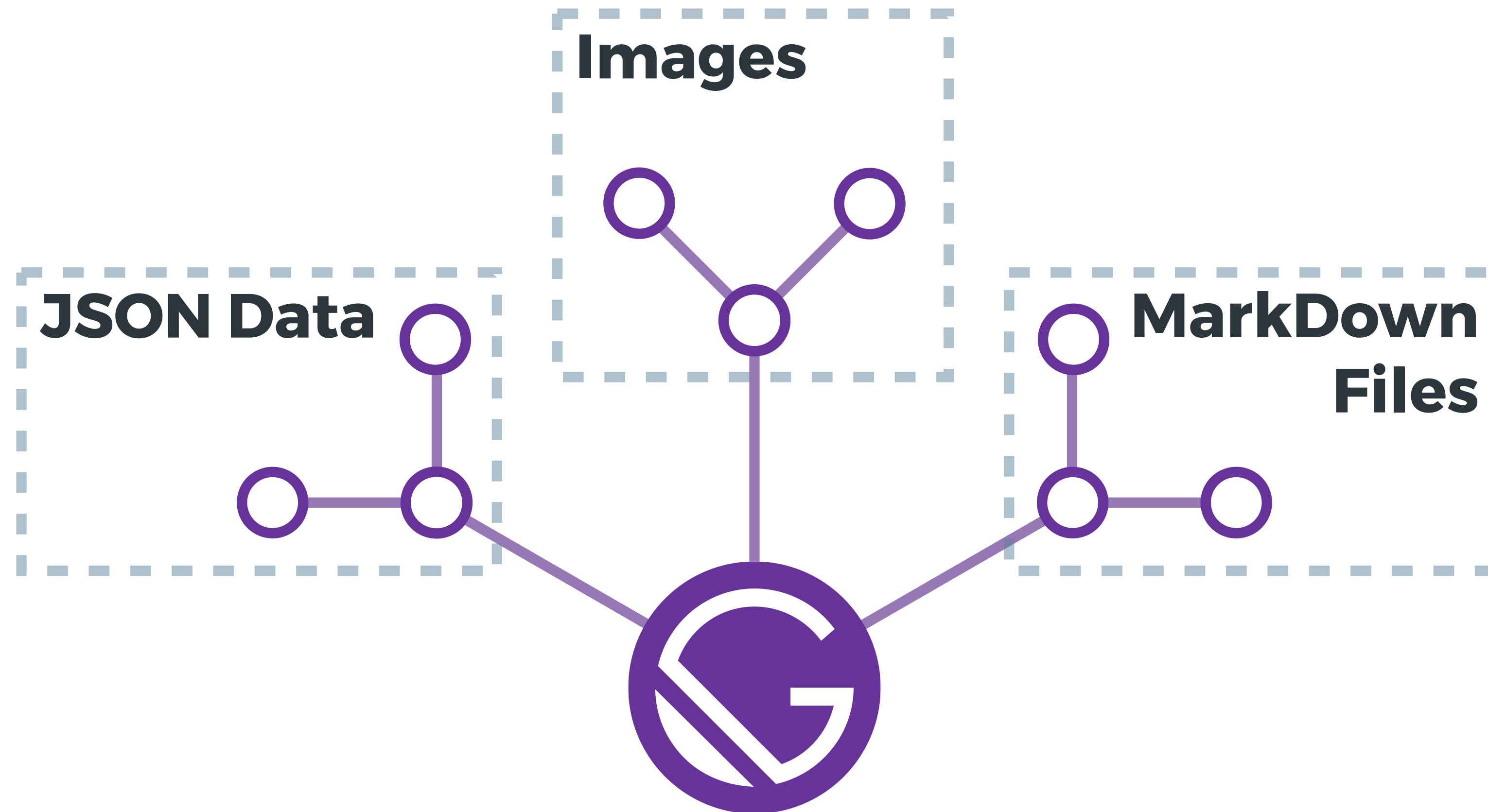
What makes GraphQL special?

REST 2.0

- **Single endpoint**
- **Query language**
- **Data-storage agnostic + per-field storage**

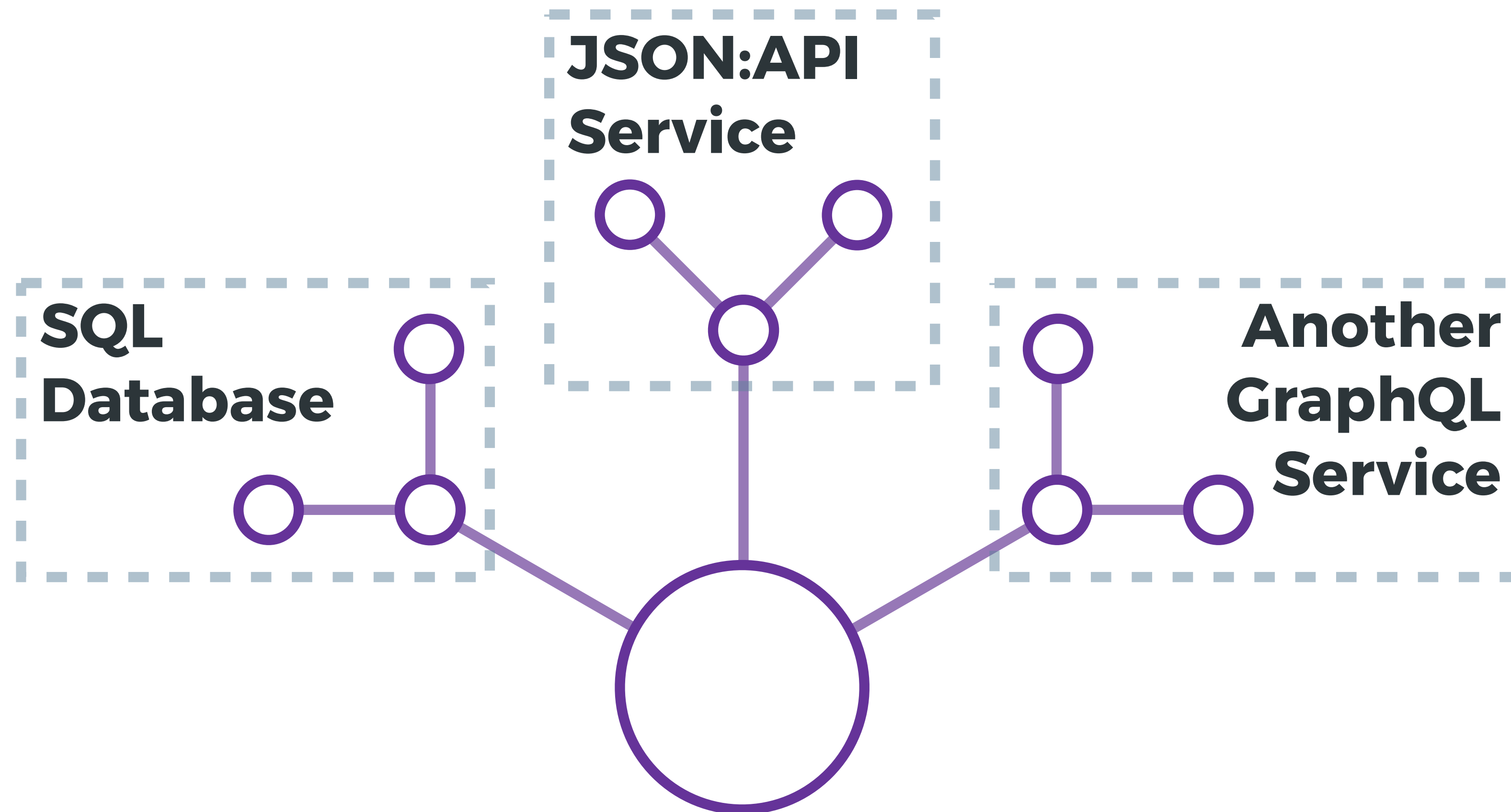
FIELD-SPECIFIC STORAGE

EXAMPLE: GATSBY PLUGINS



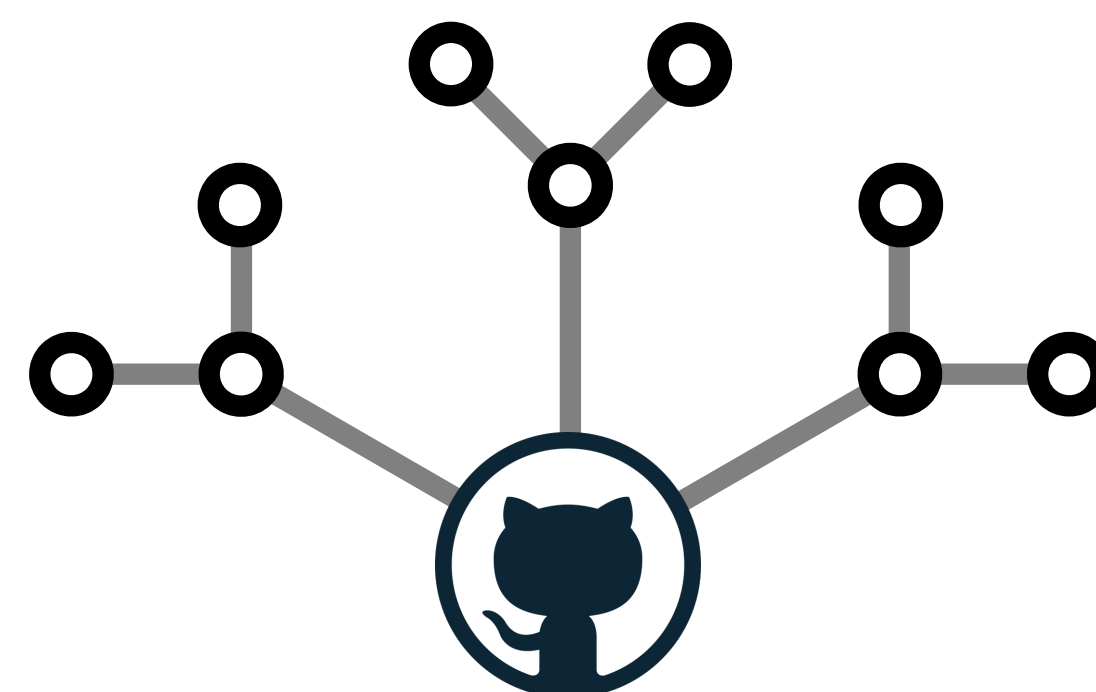
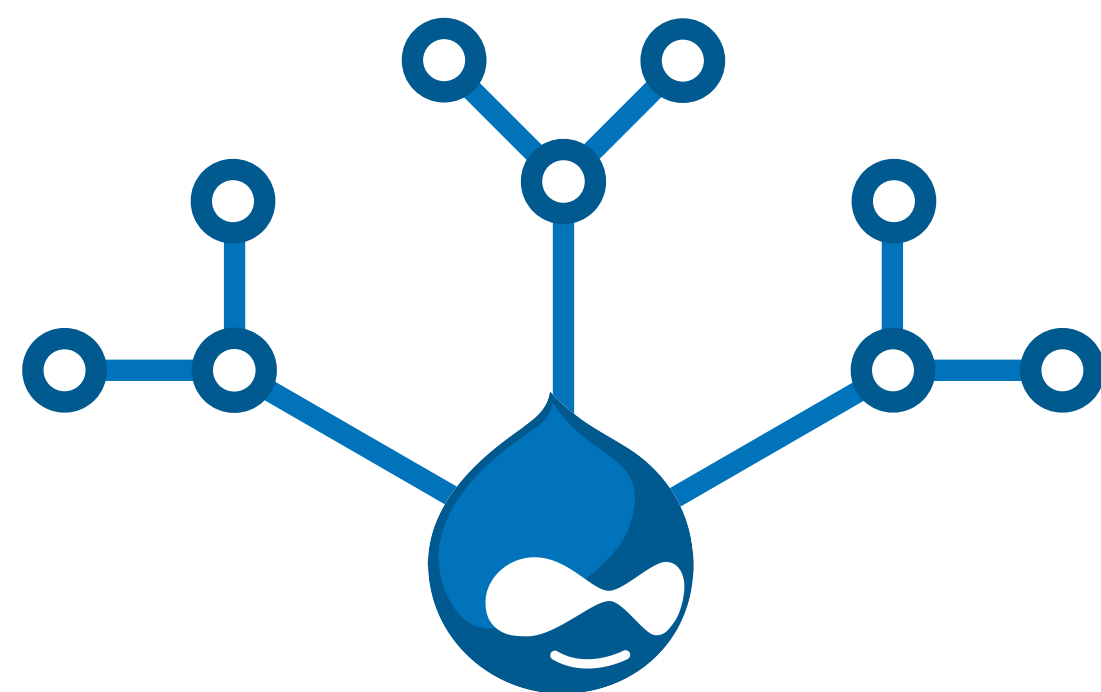
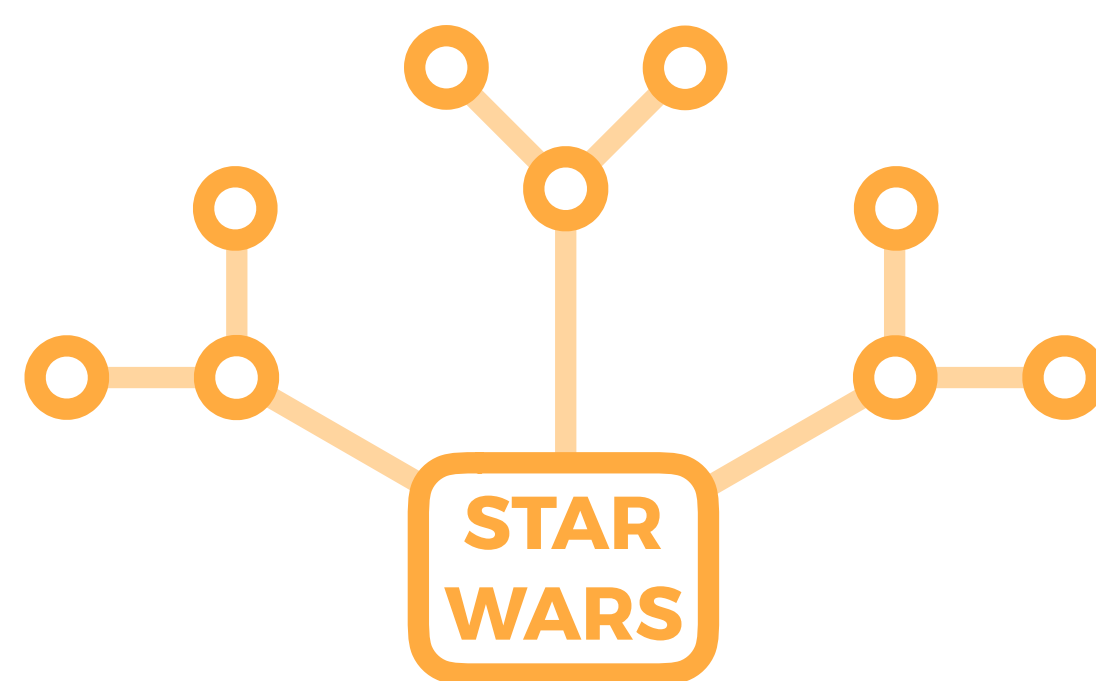
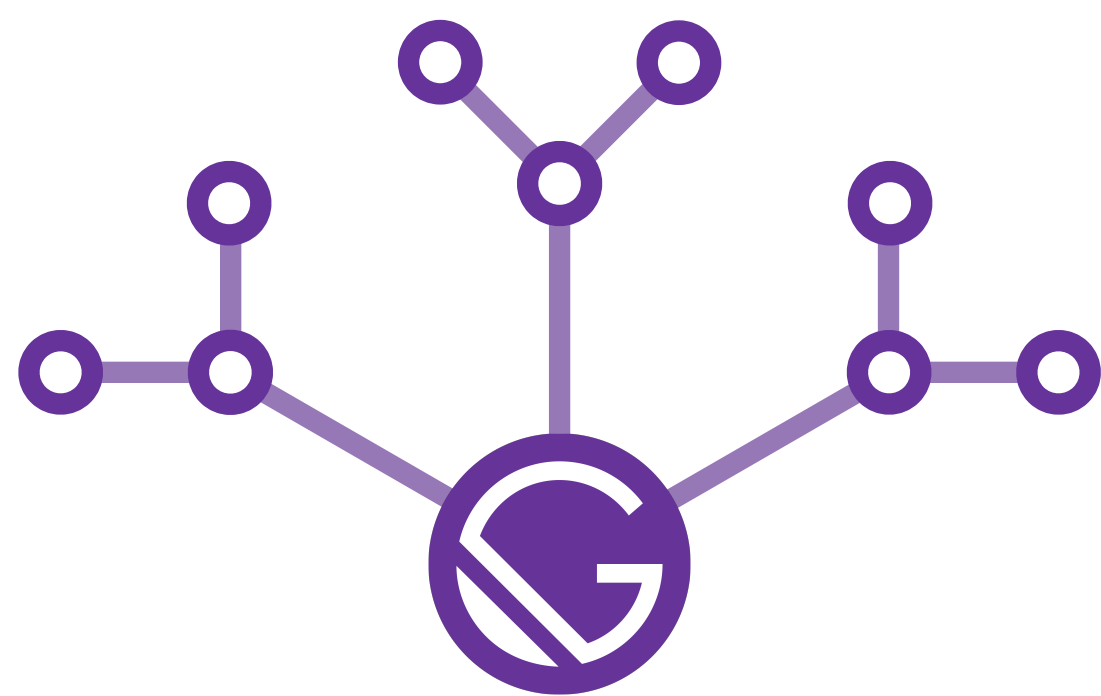
UNIFIED GRAPH

EACH FIELD CAN HAVE ITS OWN SOURCE



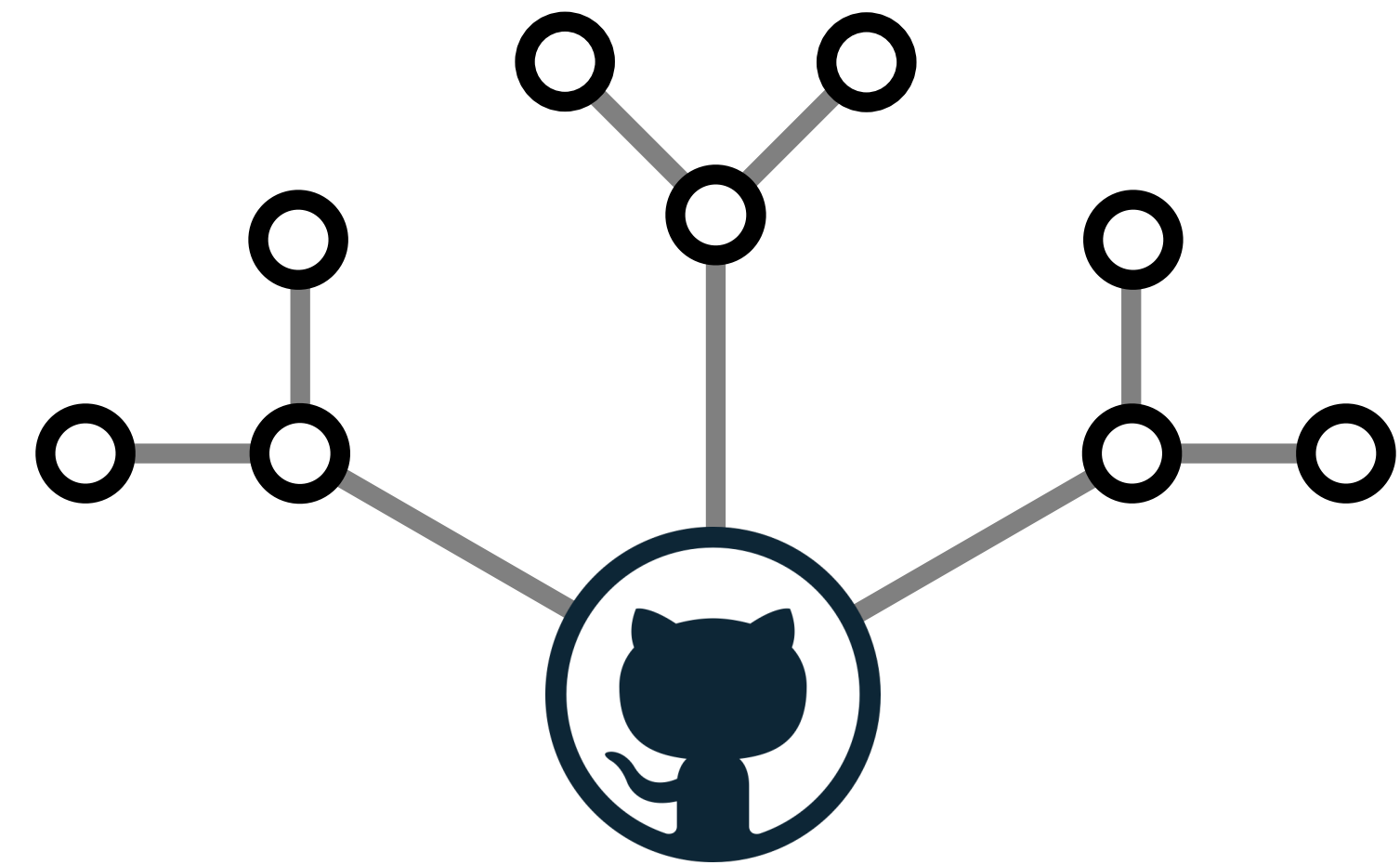
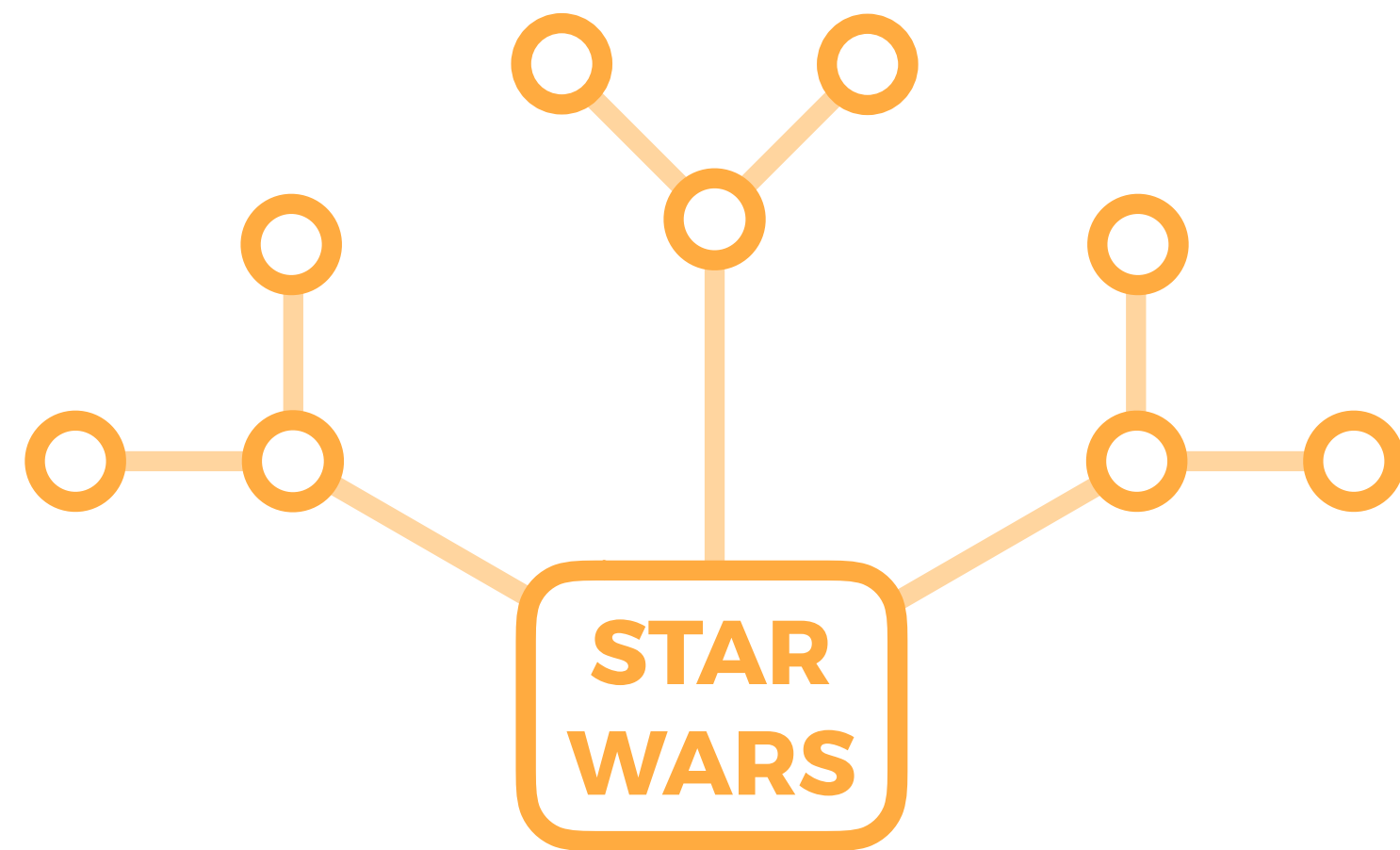
MERGING GRAPHS

SCHEMA STITCHING



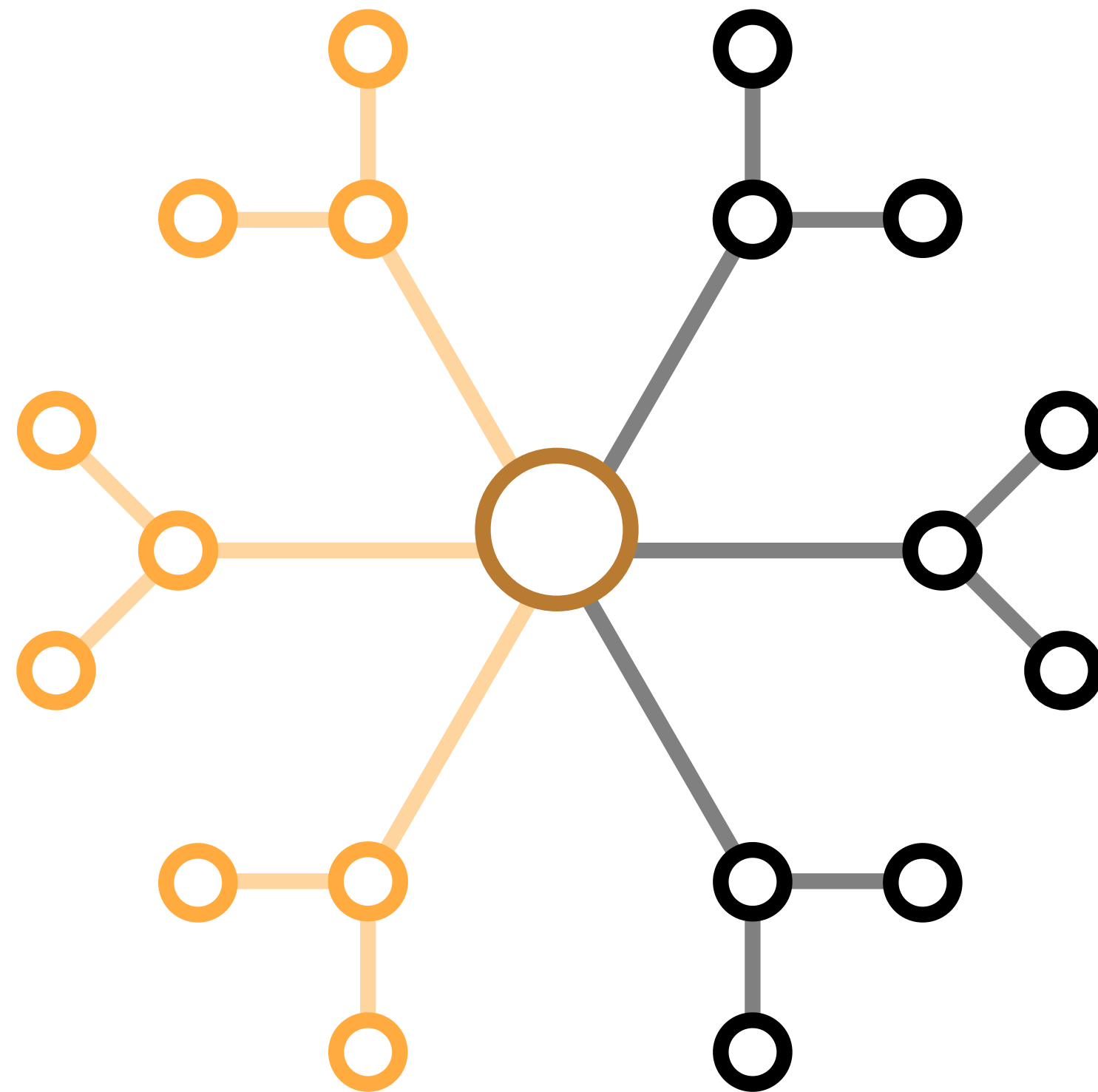
SCHEMA STITCHING

SIMPLE CASE



SCHEMA STITCHING

SIMPLE CASE



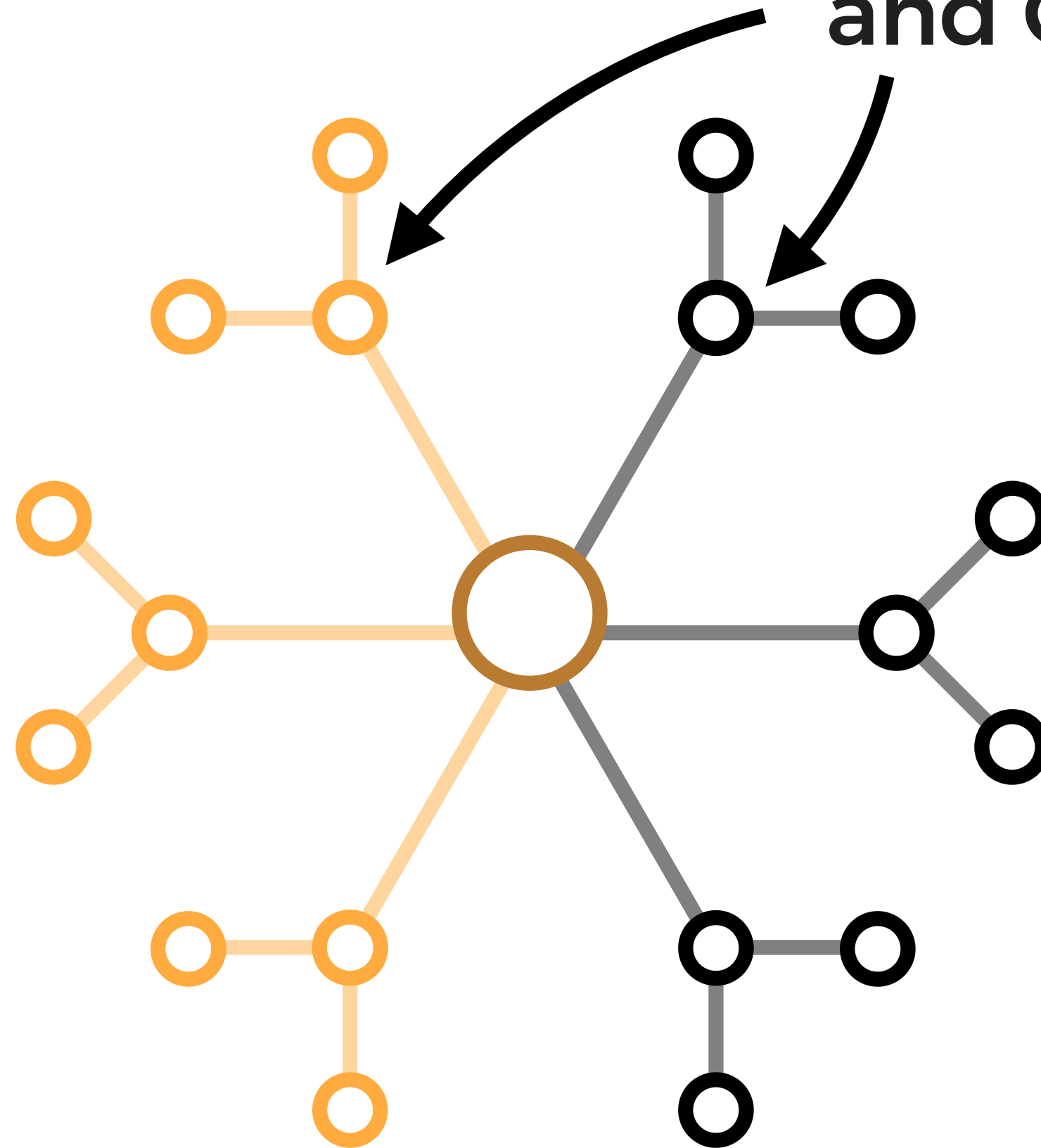
SCHEMA STITCHING

SIMPLE CASE

can't have both
Star Wars' allFiles
and GitHub' allFiles

REQUIREMENTS:

- **Unique root fields**



SCHEMA STITCHING

SIMPLE CASE

REQUIREMENTS:

- **Unique root fields**
- **Unique types**



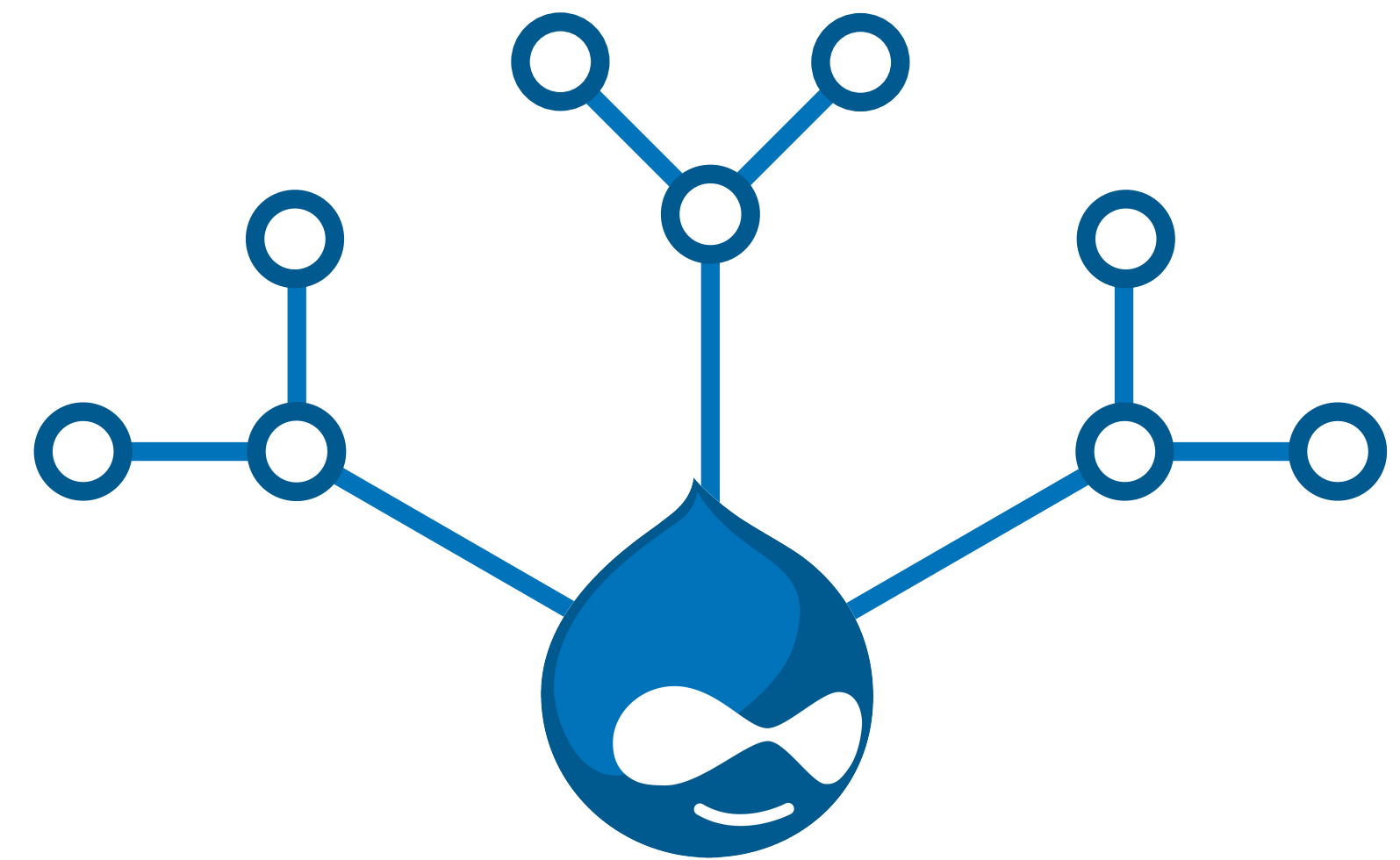
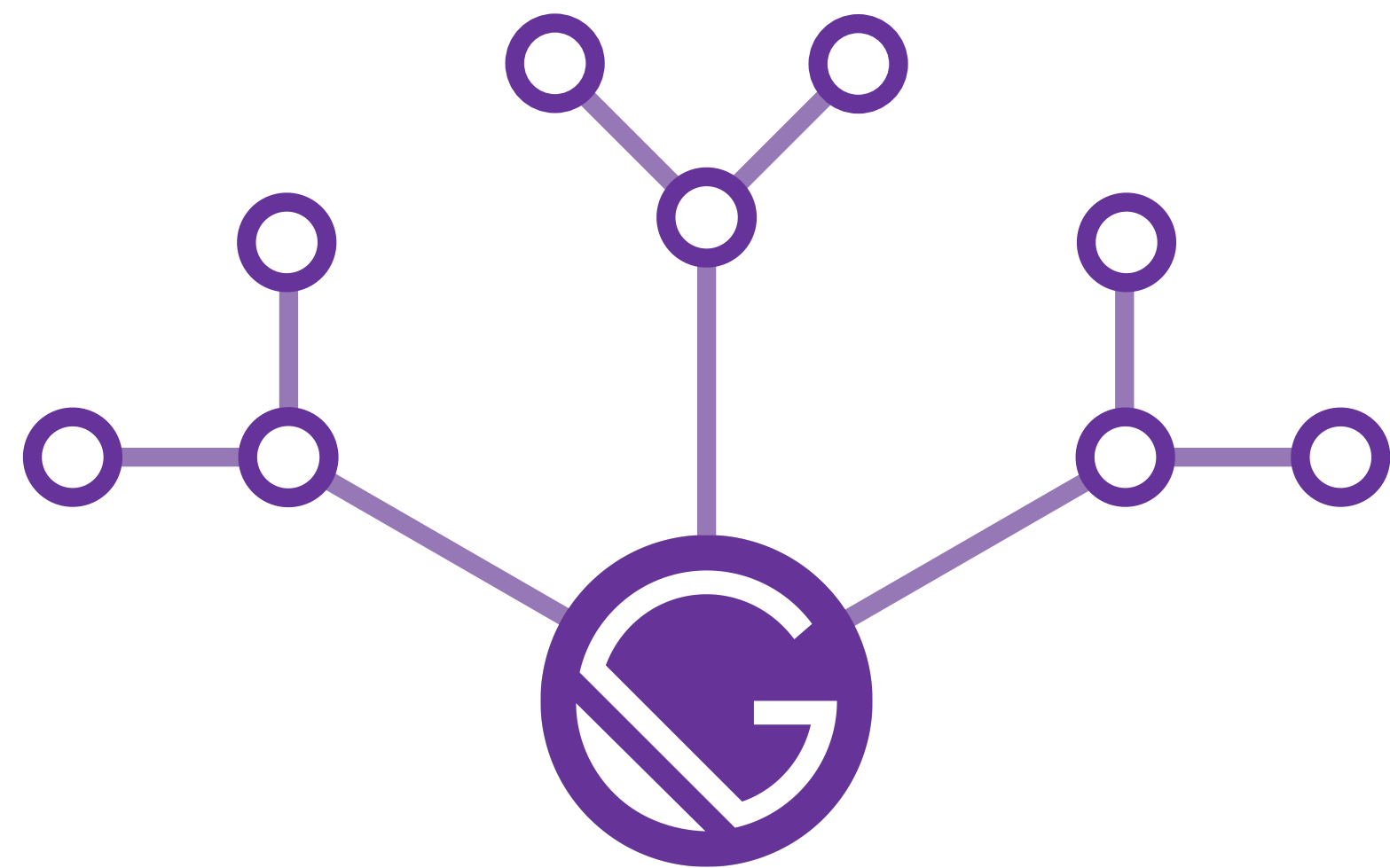
```
Person {  
  name: String,  
  weapons: Weapon[],  
}
```



```
Person {  
  name: String,  
  repositories: Repo[],  
}
```

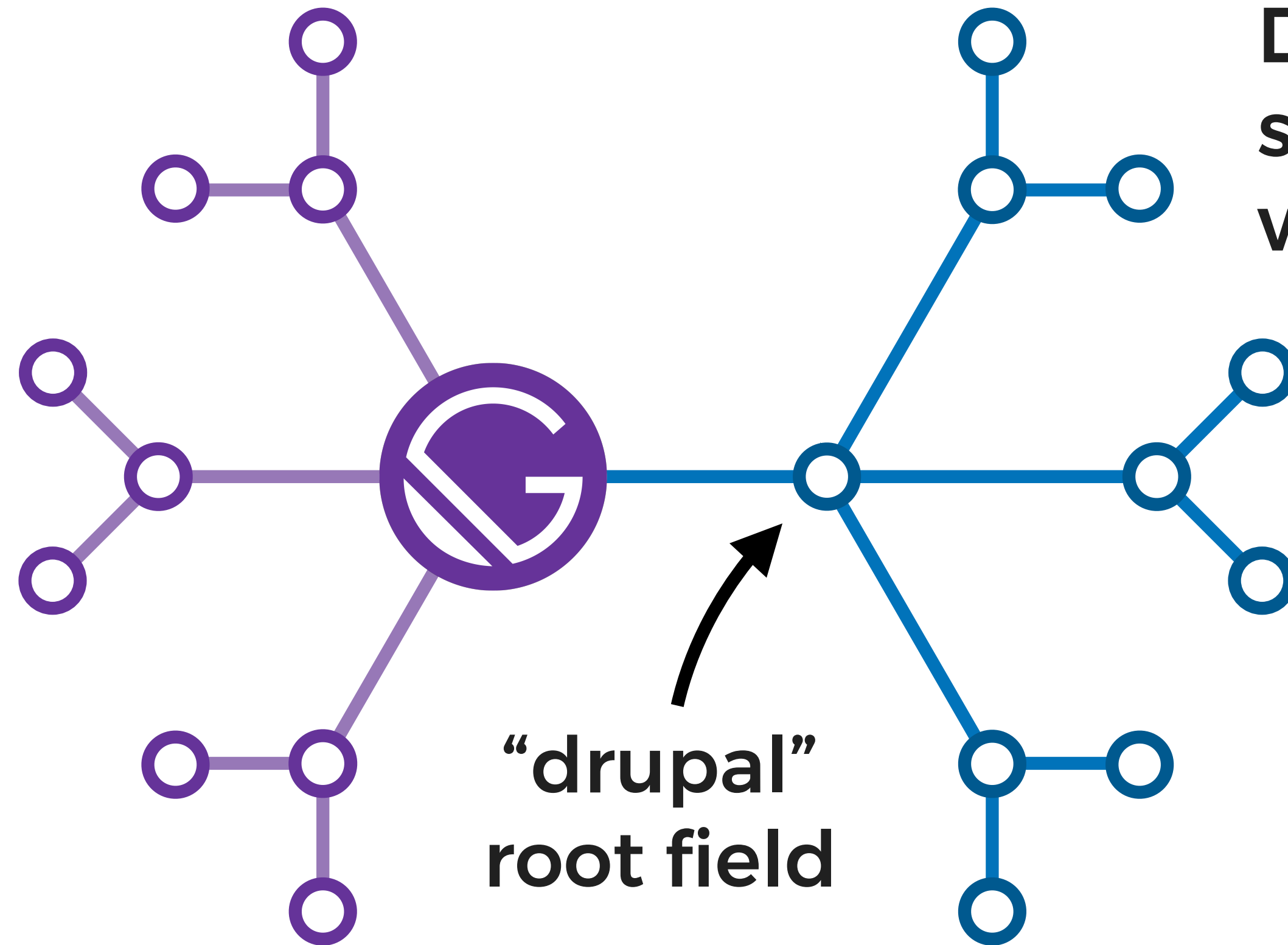
SCHEMA STITCHING

ADVANCED CASE



SCHEMA STITCHING

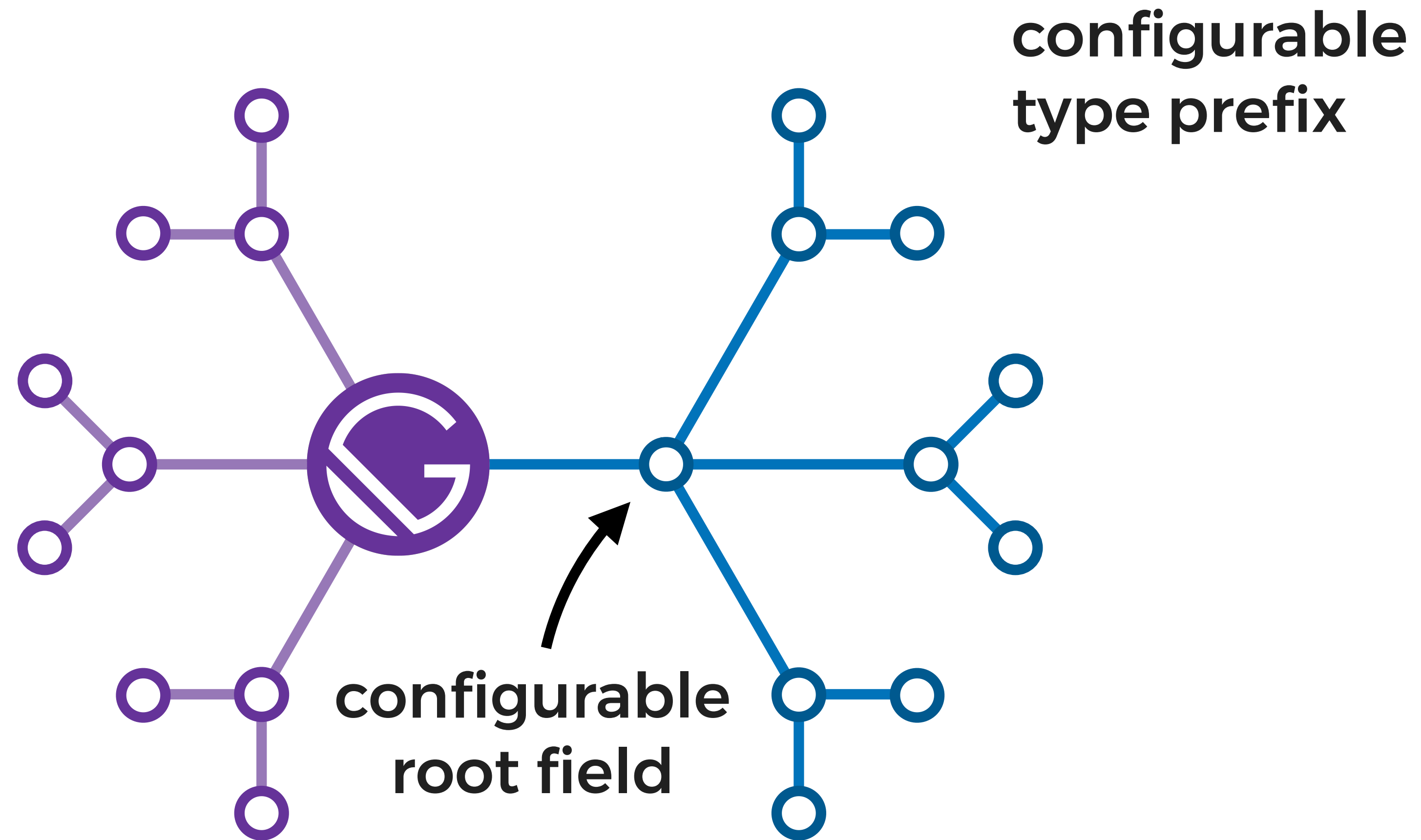
ADVANCED CASE



All types from
Drupal GraphQL
schema prefixed
with “Drupal_”

GATSBY-SOURCE-GRAPHQL

ADVANCED CASE



GATSBY-SOURCE-GRAPHQL

CONFIGURATION

```
// gatsby-config.js
```

```
module.exports = {  
  plugins: [  
    {  
      resolve: "gatsby-source-graphql",  
      options: {  
        typeName: `Drupal`,  
        fieldName: `umami`,  
        url: `https://umami.example.com/graphql`,  
      }  
    }  
  ],  
}
```

GATSBY-SOURCE-GRAPHQL

CONFIGURATION

```
// gatsby-config.js
```

```
module.exports = {  
  plugins: [  
    {  
      resolve: "gatsby-source-graphql",  
      options: {  
        typeName: `Drupal`,  
        fieldName: `umami`,  
        url: `https://umami.example.com/graphql`,  
      },  
    },  
    {  
      resolve: "gatsby-source-graphql",  
      options: {  
        typeName: `SWAPI`,  
        fieldName: `swapi`,  
        url: `https://swapi.graph.cool/`,  
      },  
    },  
  ],  
}
```


AUTHENTICATION

SECURING YOUR DRUPAL ENDPOINT

▼ GRAPHQL

- GraphQL** ▶ Base module for integrating GraphQL with Drupal.
- GraphQL Breadcrumbs test** ▶ Test for breadcrumbs.
- GraphQL Core** ▶ Provides type system plugins and drivers on behalf of core modules.
- GraphQL Twig** ▶ Render twig templates with graphql data.
- GraphQL Views** ▶ Adds support for views.

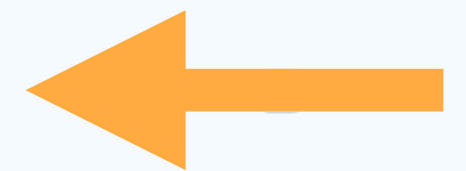
▼ WEB SERVICES

- HAL** ▶ Serializes entities using Hypertext Application Language.
- HTTP Basic Authentication** ▶ Provides the HTTP Basic authentication provider

AUTHENTICATION

SECURING YOUR DRUPAL ENDPOINT

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
GraphQL			
Administer query maps Allows users to import and edit query map configuration.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Bypass field security Allows the user to manually resolve insecure fields.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Execute arbitrary GraphQL requests Allows users to execute arbitrary GraphQL requests.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Execute persisted GraphQL requests Allows users to execute persisted GraphQL requests.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Use the GraphQL Voyager Allows users to use the GraphQL Voyager interface.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Use the GraphQL explorer Allows users to use the GraphQL interface.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>



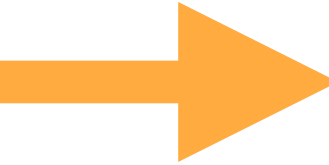
AUTHENTICATION

SECURING YOUR DRUPAL ENDPOINT

```
// gatsby-config.js
```

```
// btoa will base64-encode your "USERNAME:PASSWORD" string.  
const btoa = require("btoa");
```

```
module.exports = {  
  plugins: [  
    {  
      resolve: "gatsby-source-graphql",  
      options: {  
        typeName: `Drupal`,  
        fieldName: `umami`,  
        url: `https://example.com/graphql`,  
        headers: {  
          Authorization: `Basic ${btoa(  
            process.env.ENDPOINT_USERNAME + ":" + process.env.ENDPOINT_PASSWORD  
          )}`  
        }  
      }  
    }  
  ],  
}
```



AUTHENTICATION

SECURING YOUR DRUPAL ENDPOINT

```
// gatsby-config.js
```

```
// btoa will base64-encode your "USERNAME:PASSWORD" string.  
const btoa = require("btoa");
```

```
module.exports = {  
  plugins: [  
    {  
      resolve: "gatsby-source-graphql",  
      options: {  
        typeName: `Drupal`,  
        fieldName: `umami`,  
        url: `https://example.com/graphql`,  
        headers: {  
          Authorization: `Basic ${btoa("admin:admin")}`  
        }  
      }  
    },  
  ],  
}
```



LIVE DEMO



GATSBY 2.5'S GRAPHQL APIS

LOW-LEVEL GRAPHQL APIS

- **Node APIs** <https://www.gatsbyjs.org/docs/node-apis/>
 - **createResolvers**
 - **createSchemaCustomization**
- **Actions** <https://www.gatsbyjs.org/docs/actions/>
 - **addThirdPartySchema**
 - **createFieldExtension**
 - **createTypes**

GATSBY 2.5'S GRAPHQL APIS

LOW-LEVEL GRAPHQL APIS

<https://www.gatsbyjs.org/blog/tags/graphql>



Mikhail Novikov

GraphQL sommelier. Did Reindex, GraphQL Delegation, Schema
Stitching, Launchpad. Organizer @GraphQLFinland.

[@freiksenet](#)

Improvements to Schema Customization API - Available in Gatsby 2.5.0

Today we are releasing further improvements to the schema customization that we've released in version 2.2.0 . You can use them with Gatsby...



[Mikhail Novikov](#) on May 17, 2019

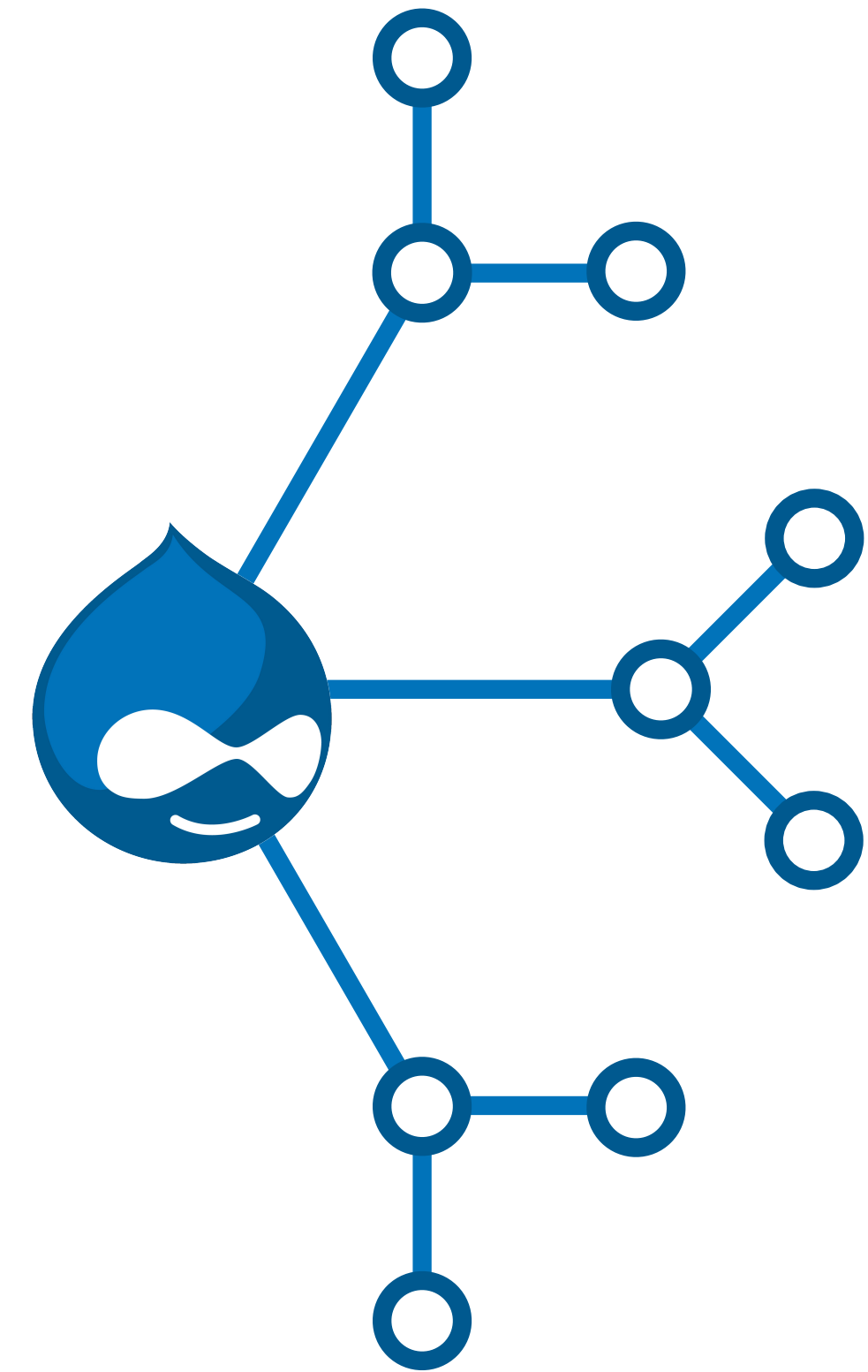
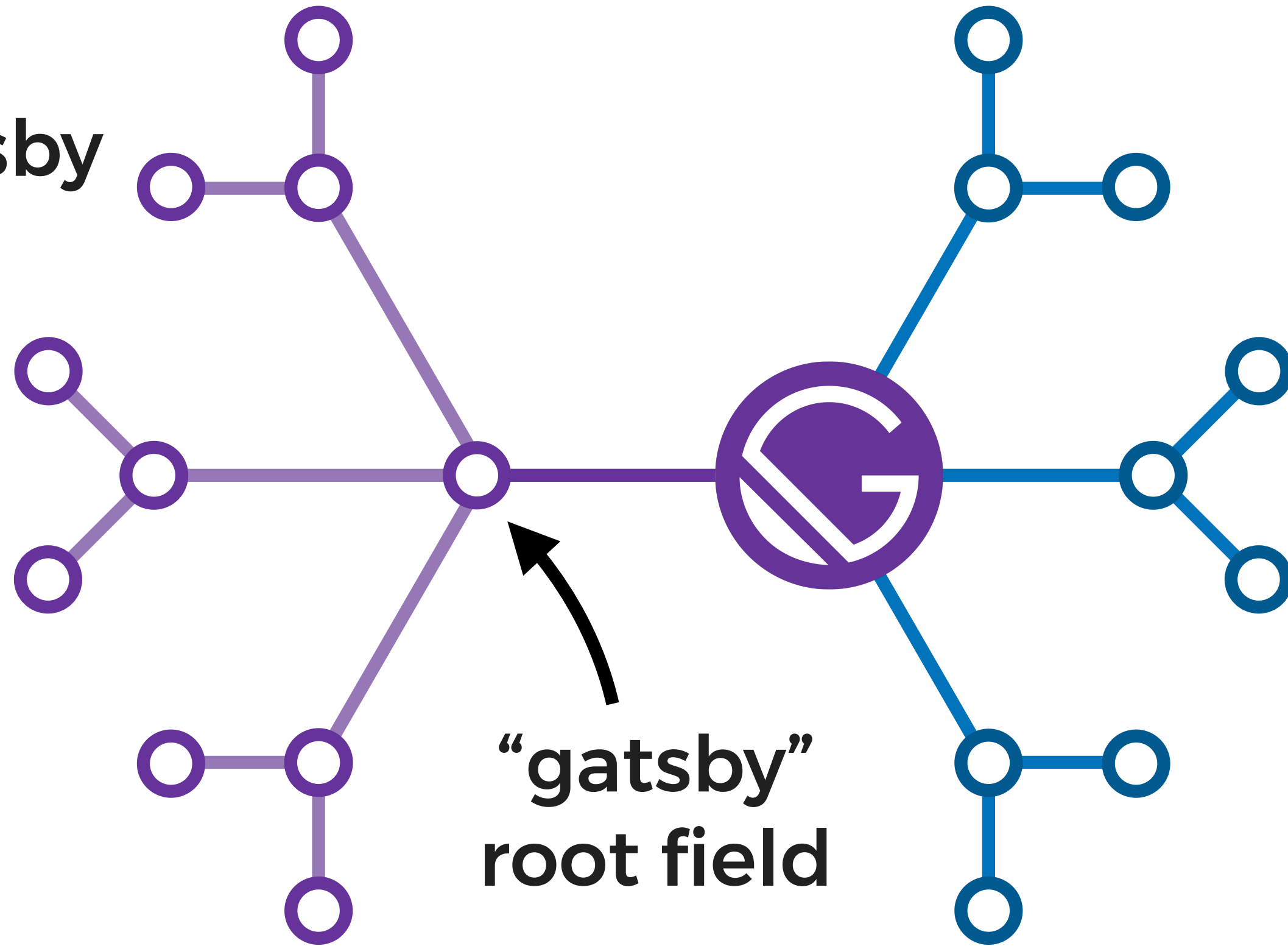
MY DREAM

1 QUERY FOR 2 GRAPHS

build time

run time

All types
from Gatsby
prefixed
with
"Gatsby_"

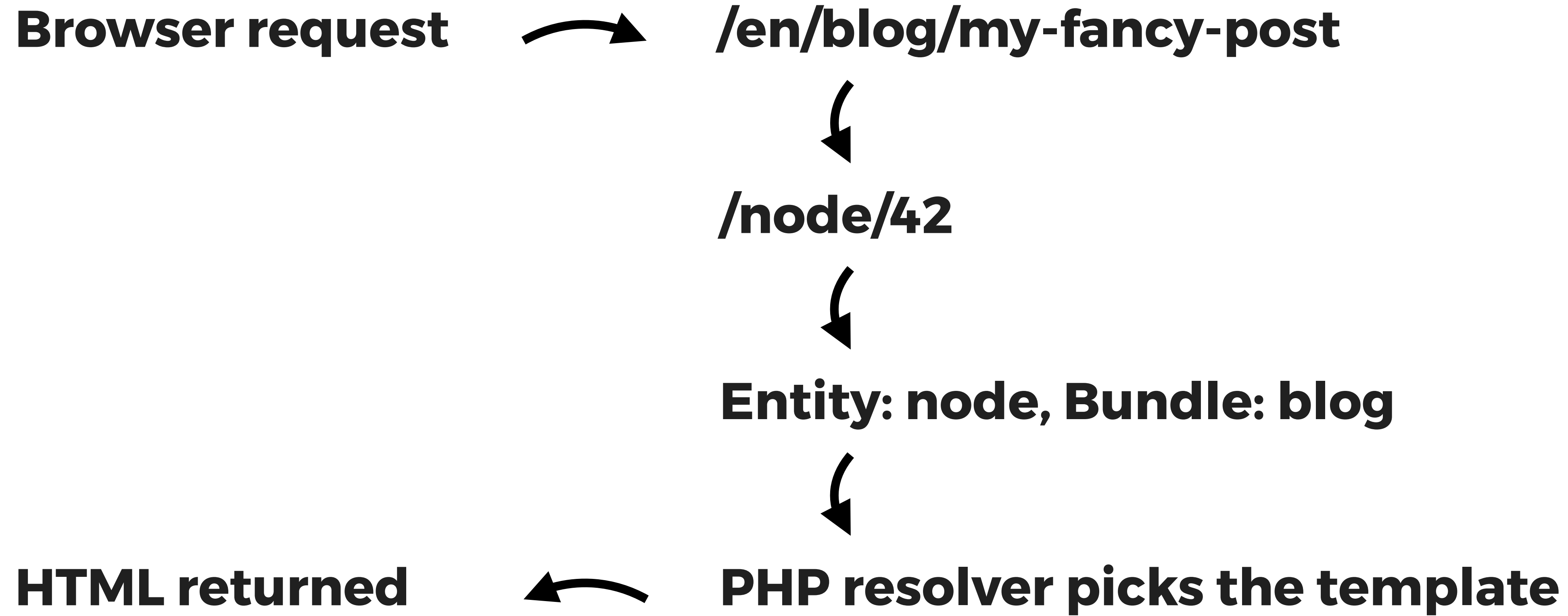


**MULTILINGUAL
GRAPHQL
QUERIES**



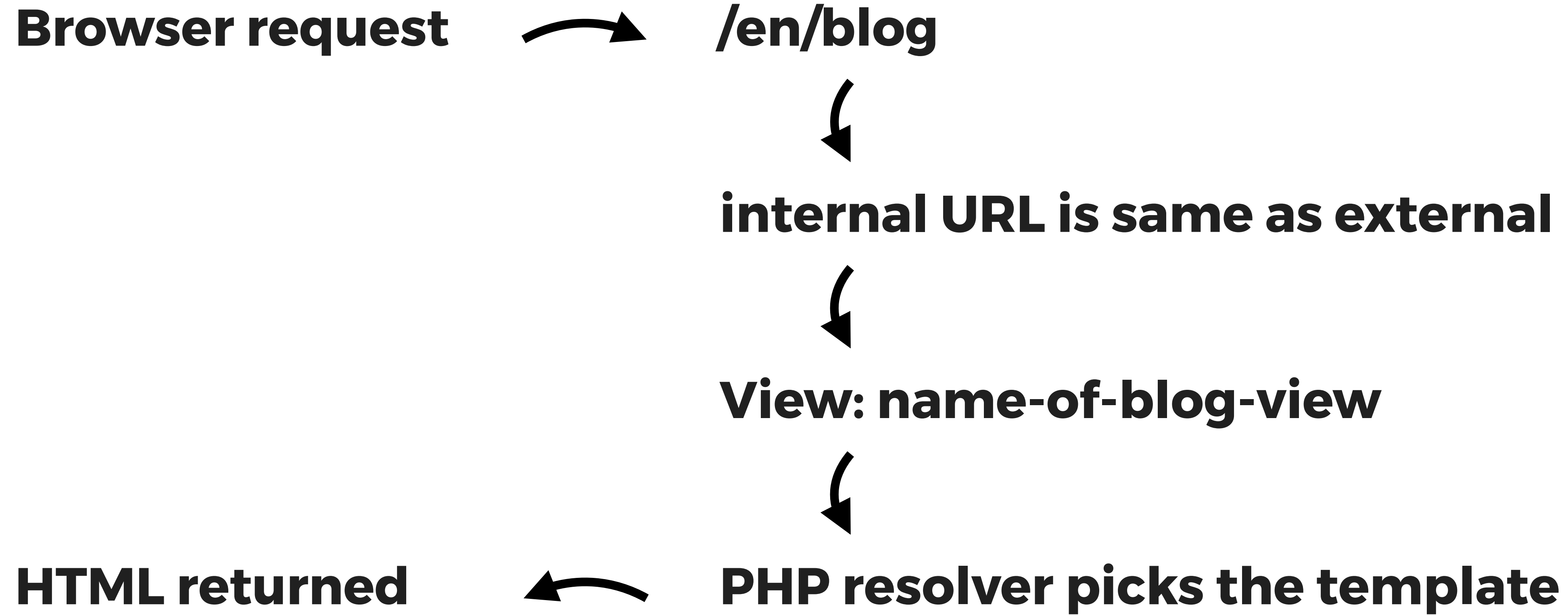
DRUPAL ROUTING: ENTITY

SIMPLIFIED EXAMPLE



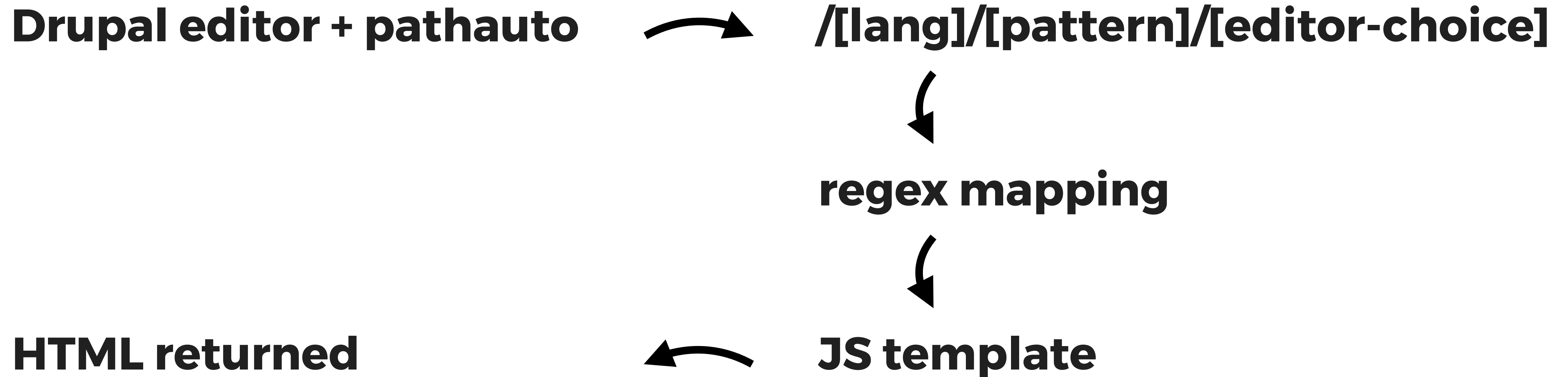
DRUPAL ROUTING: VIEWS

SIMPLIFIED EXAMPLE



GATSBY ROUTING

DRUPAL MANAGED PATHS USING PATHAUTO



GATSBY ROUTING

gatsby-node.js

// Specify how Drupal-configured paths should map to React components.

```
const pathMapping = [
  {
    test: /^\/articles\/.+\$/,
    component: path.resolve(`./src/templates/node-article.js`)
  },
  {
    test: /^\/recipes\/.+\$/,
    component: path.resolve(`./src/templates/node-recipe.js`)
  },
  {
    test: /^\/recipe\/-category\/.+\$/,
    component: path.resolve(`./src/templates/taxonomy-term-recipe-category.js`)
  },
  {
    test: /^\/tags\/.+\$/,
    component: path.resolve(`./src/templates/taxonomy-term-tags.js`)
  },
  {
    test: /^\/articles$/
```

GATSBY ROUTING

gatsby-node.js

```
    component: path.resolve(`./src/templates/taxonomy-term-tags.js`)
  },
  {
    test: /^\/articles$/,
    component: path.resolve(`./src/templates/index-articles.js`)
  },
  {
    test: /^\/recipes$/,
    component: path.resolve(`./src/templates/index-recipes.js`)
  },
  {
    test: /^\/$/,
    component: path.resolve(`./src/templates/index.js`)
  },
  // This is the catch-all since Page node URLs follow no pattern.
  {
    test: /^\/.+$/,
    component: path.resolve(`./src/templates/node-page.js`)
  }
];
```

GATSBY ROUTING

gatsby-node.js

```
{  
  test: /^\/.+\$/,  
  component: path.resolve(`./src/templates/node-page.js`)  
}  
];
```

```
const getComponent = (language, slug) => {  
  const url =  
    slug === `/${language}` ? "/" : slug.replace(`/${language}/`, "/");  
  for (let i = 0; i < pathMapping.length; i++) {  
    if (pathMapping[i].test.test(url)) {  
      return pathMapping[i].component;  
    }  
  }  
  throw new Error(`No template found for ${slug}`);  
};
```


QUESTIONS

AND MAYBE ANSWERS

- Slides: Follow me on Twitter: @JohnAlbin
- Demo code: github.com/AmazeeLabs/umamiDemo



THANK YOU



JOHN ALBIN WILKINS
SENIOR FRONT-END DEVELOPER

john.albin@amazee.com
drupal.org/u/johnalbin
@JohnAlbin



**DECOUPLED
DAYS
2019**