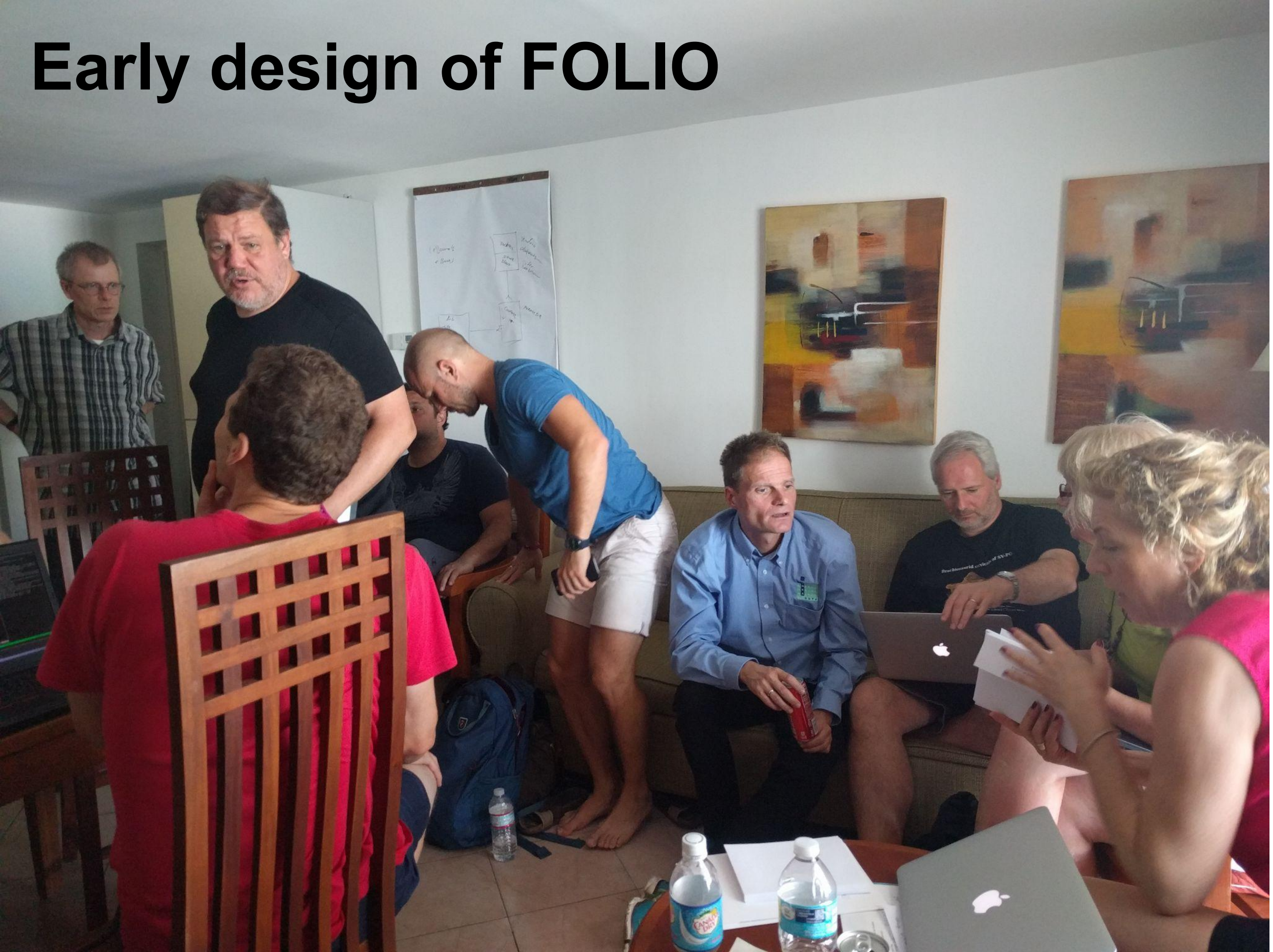
FOLIO modularity in practice

Seamless deployment of modules from multiple sources

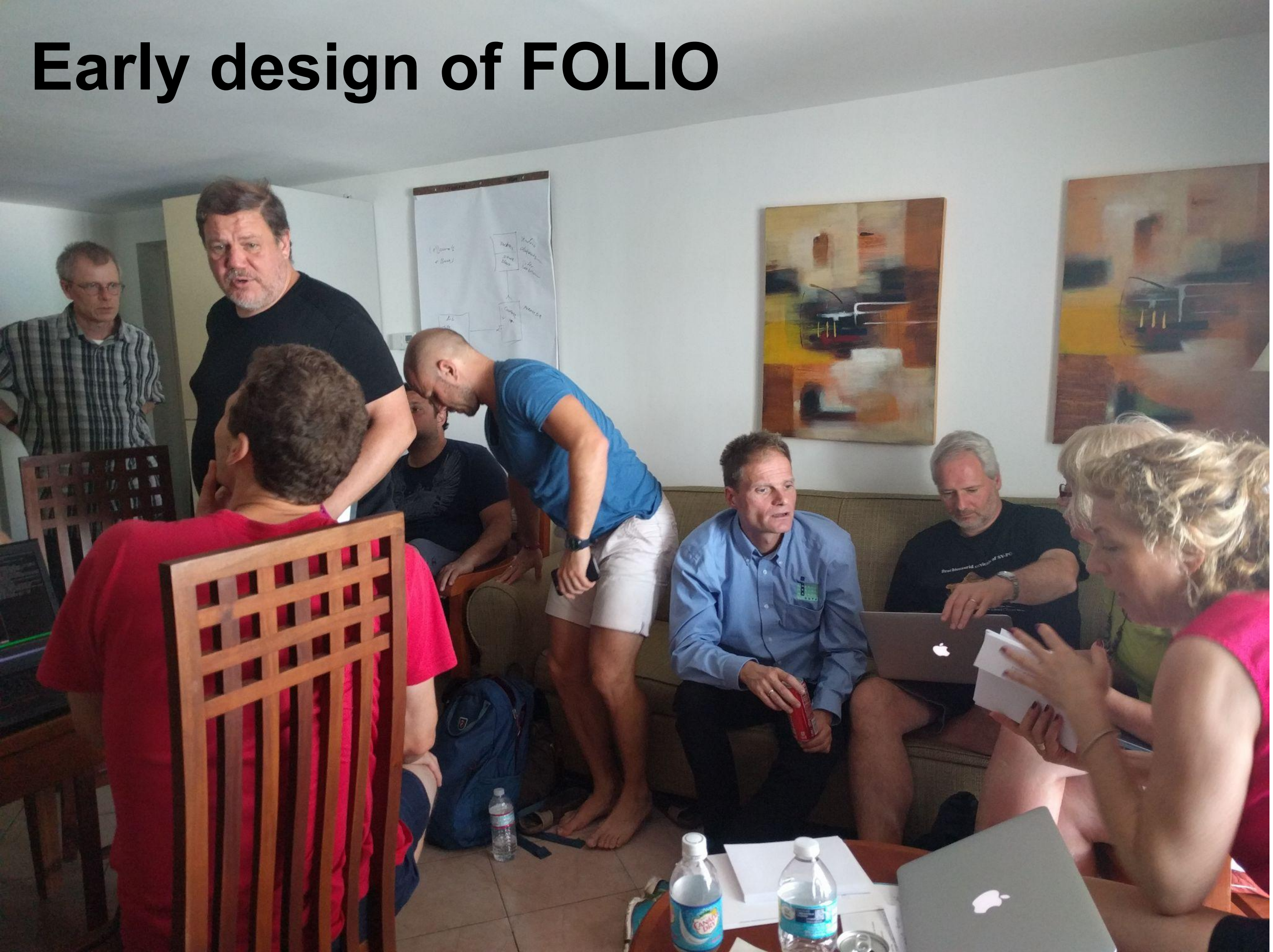
Mike Taylor
Index Data
mike@indexdata.com

Early design of FOLIO

Early design of FOLIO

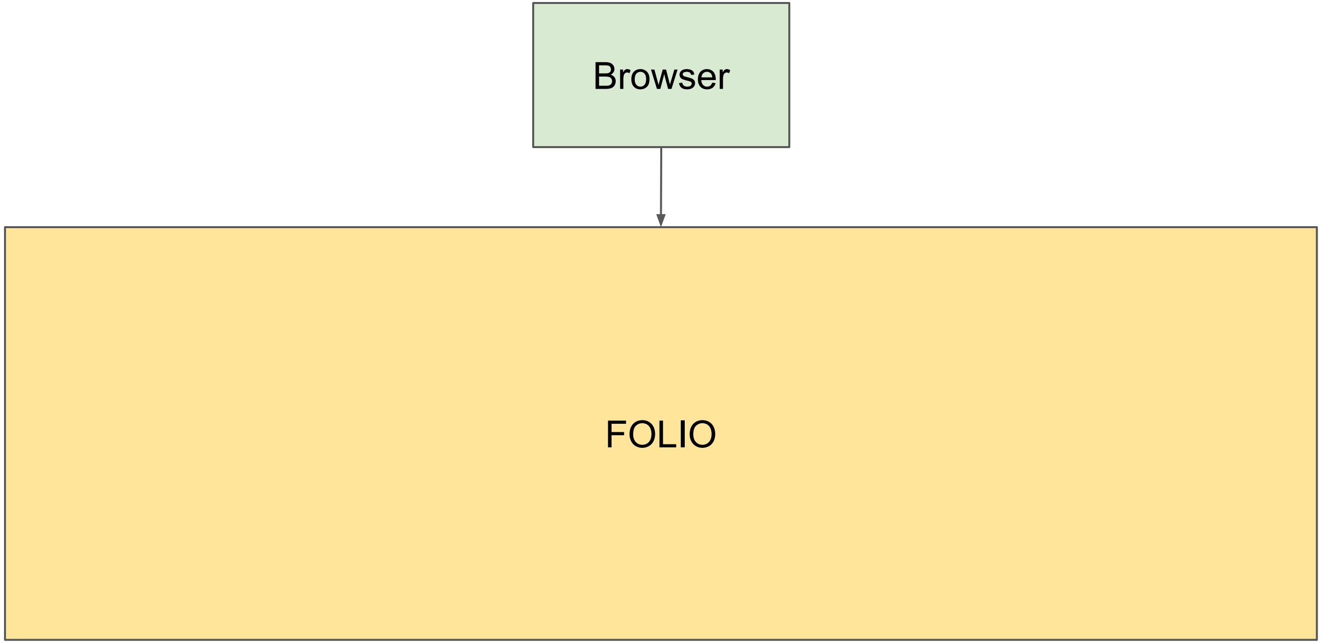


Early design of FOLIO

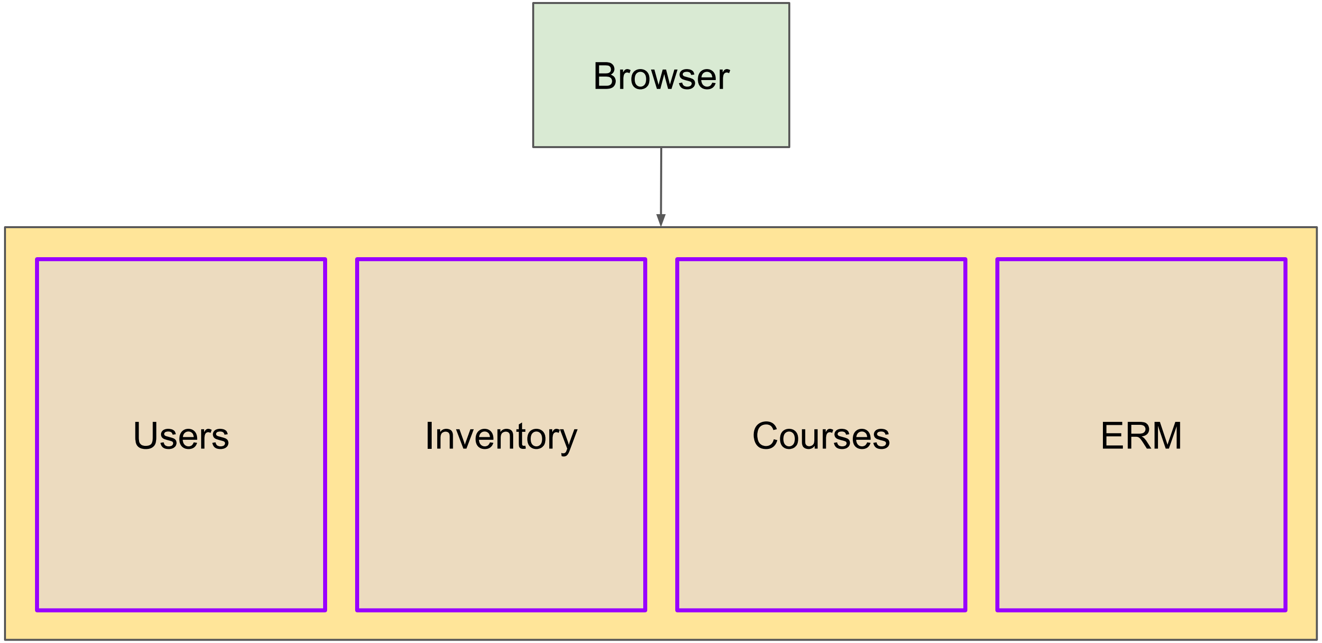


2015

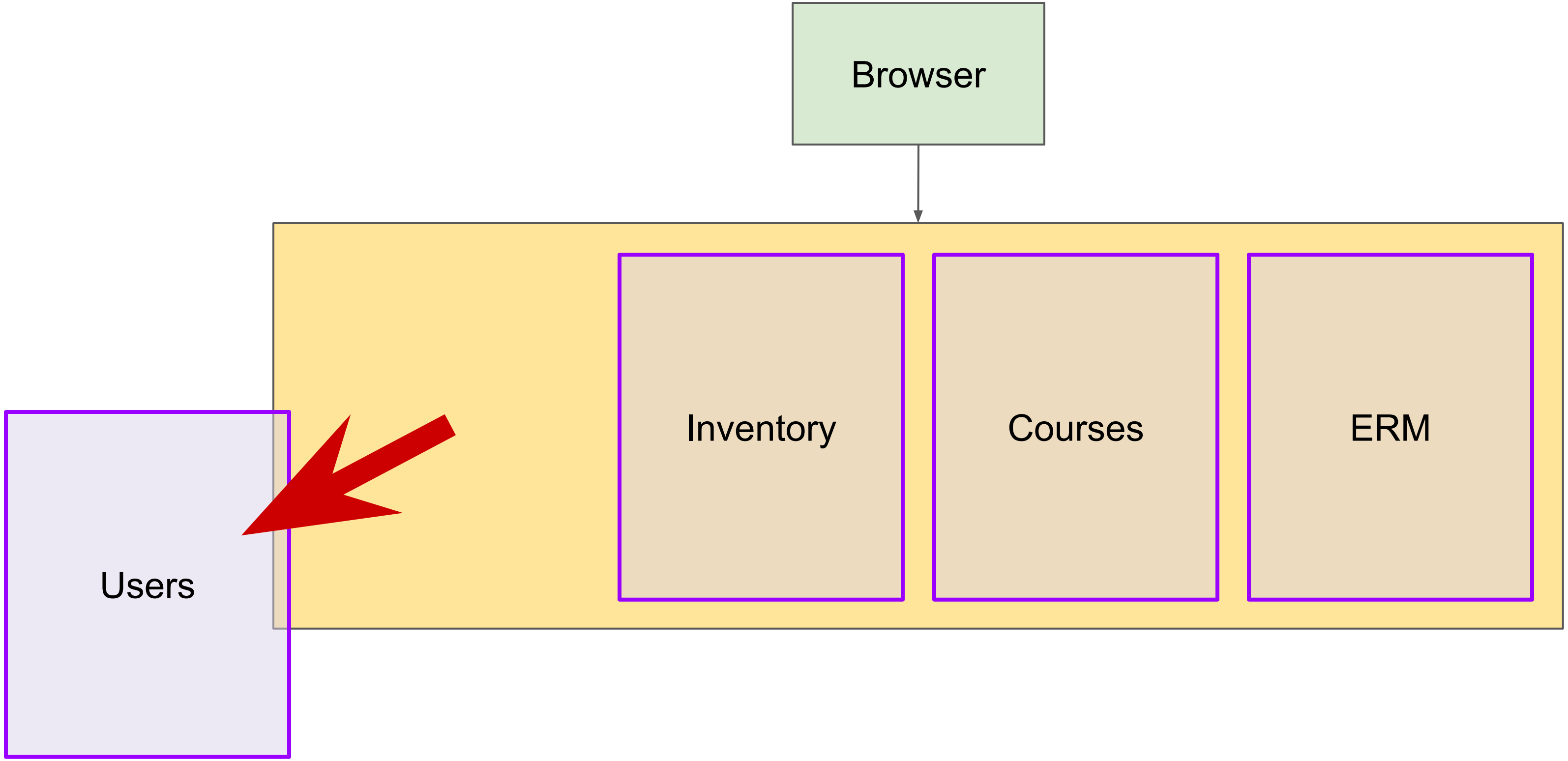
FOLIO was designed to be modular



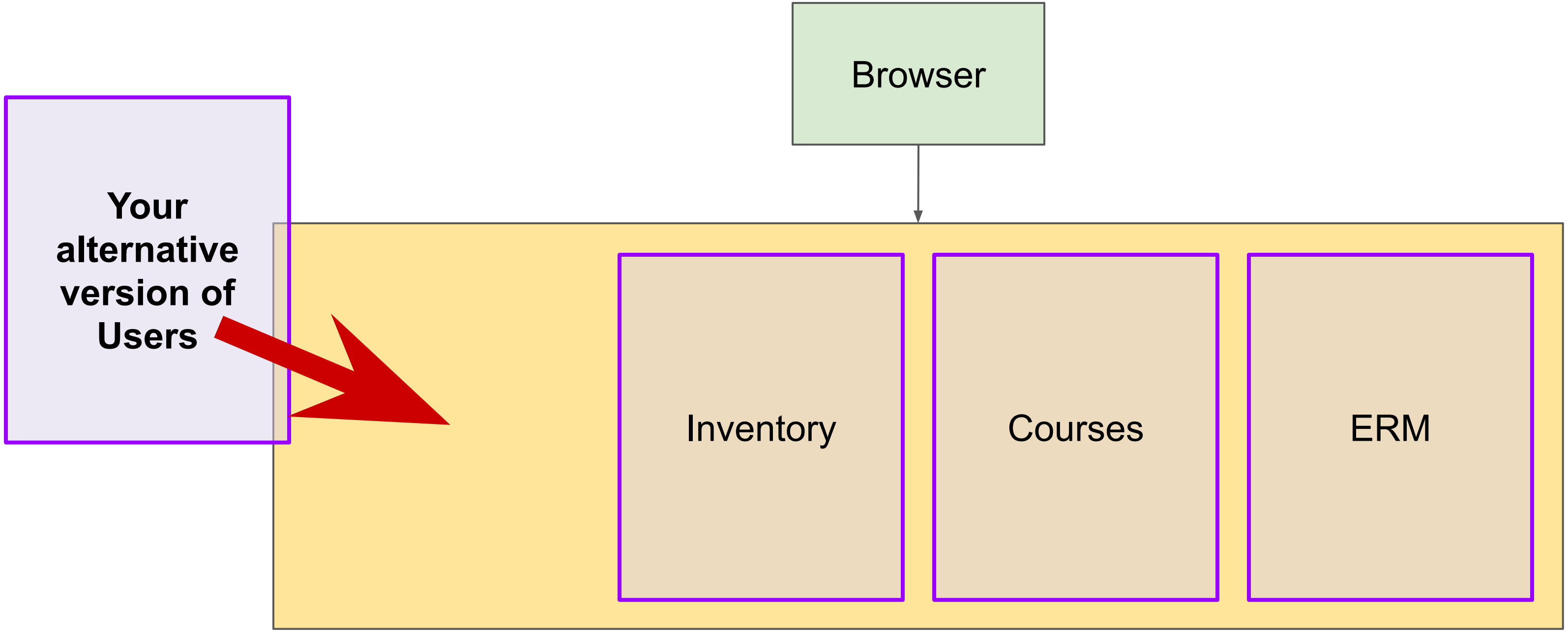
FOLIO was designed to be modular



Dependencies on interfaces, not implementations



Substituting interface-compatible modules



Browser

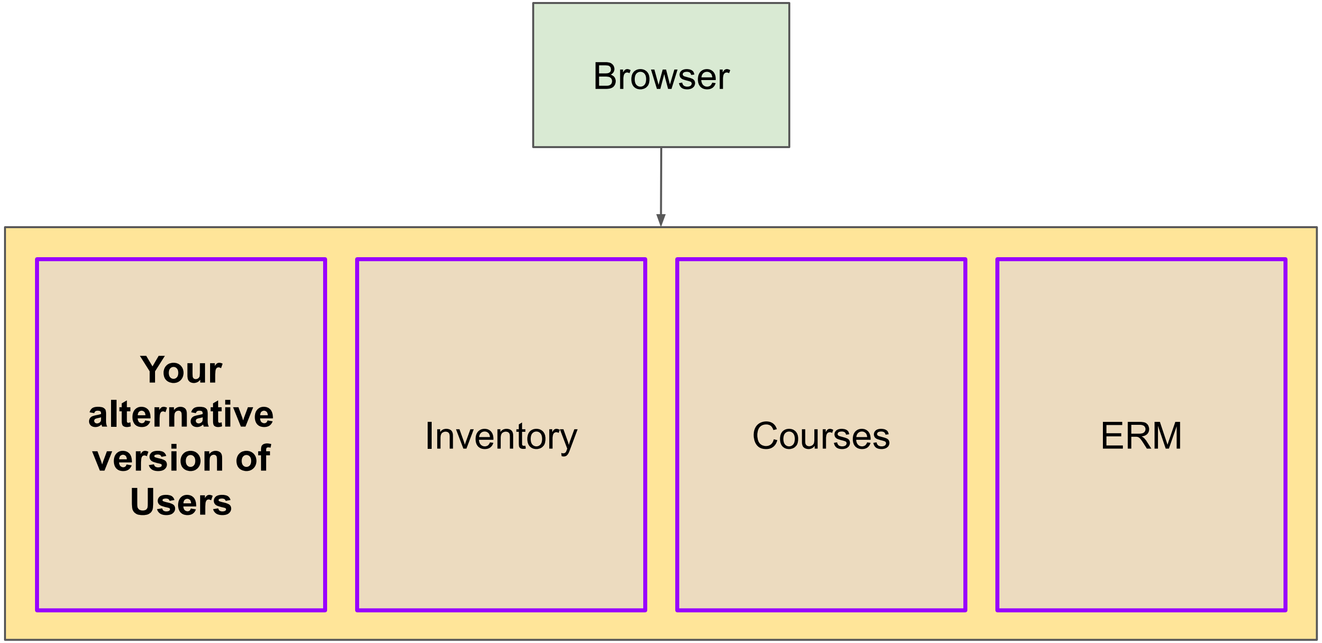
Your alternative version of Users

Inventory

Courses

ERM

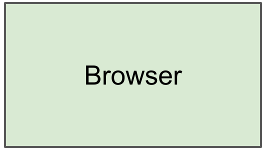
Substituting interface-compatible modules



A quick refresher on FOLIO architecture

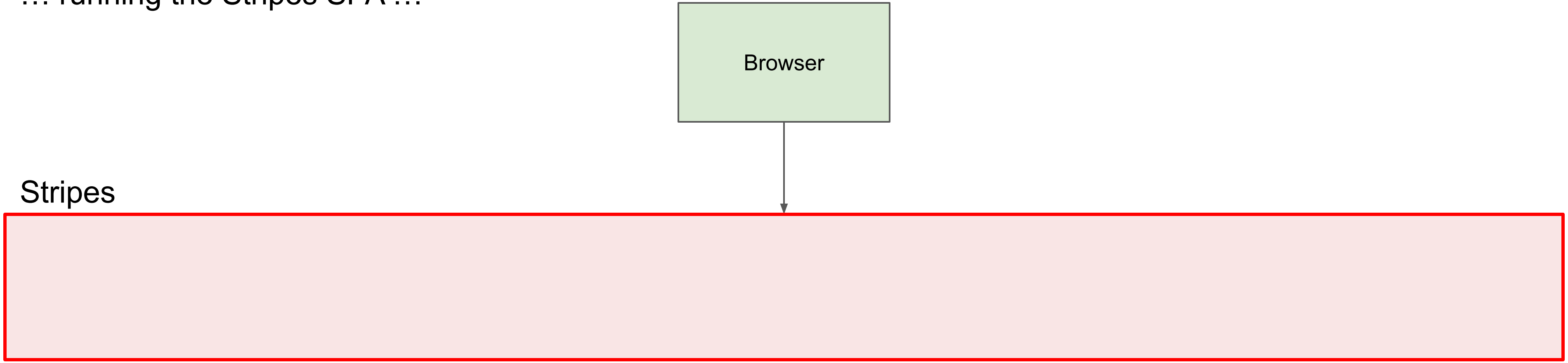
- The user’s Web browser
- Stripes – FOLIO’s single page application (SPA)
- The Okapi API gateway
- The FOLIO backend
- Stripes modules
- Backend modules
- Apps

A user at a web browser …

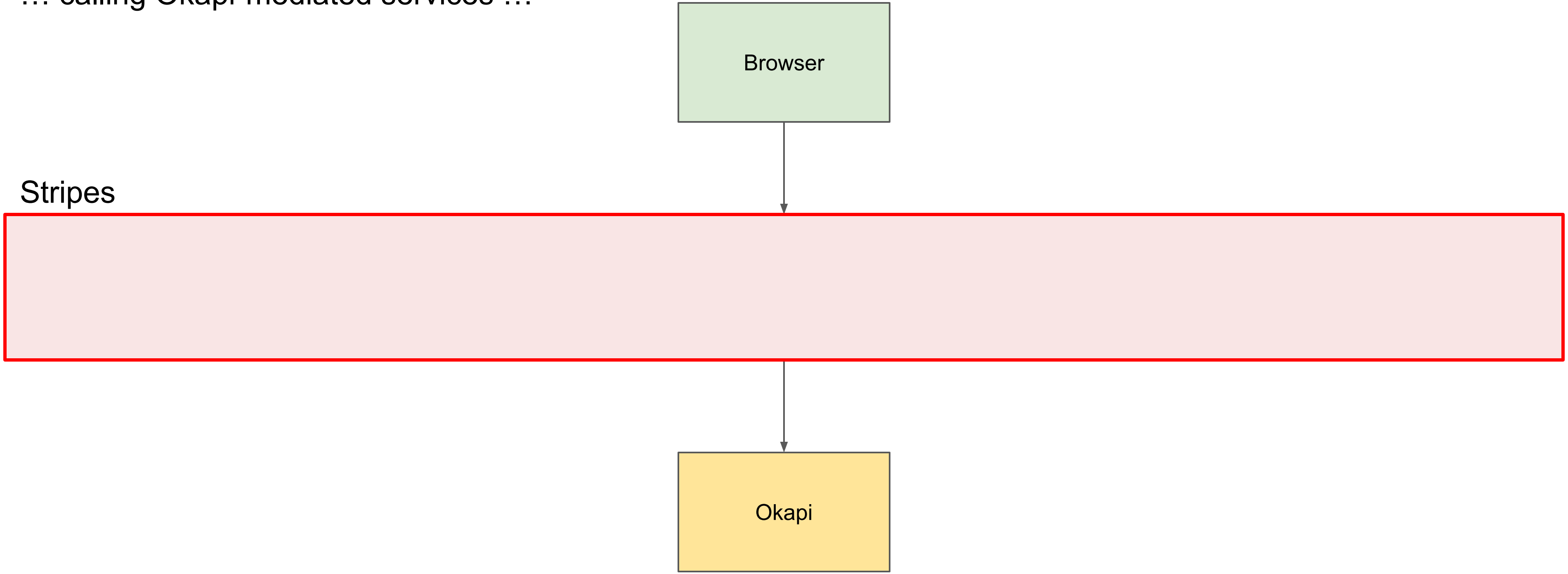


Browser

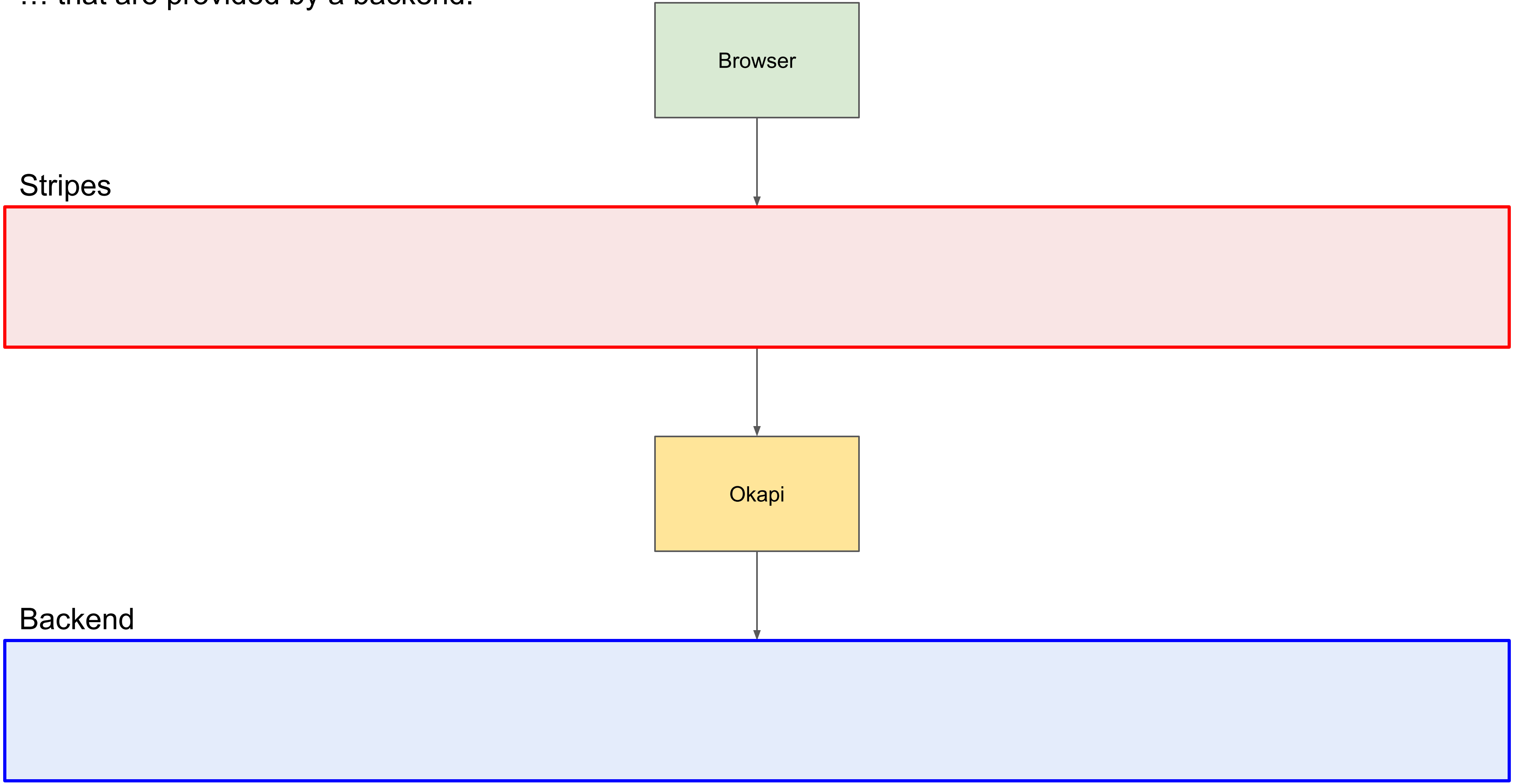
… running the Stripes SPA …



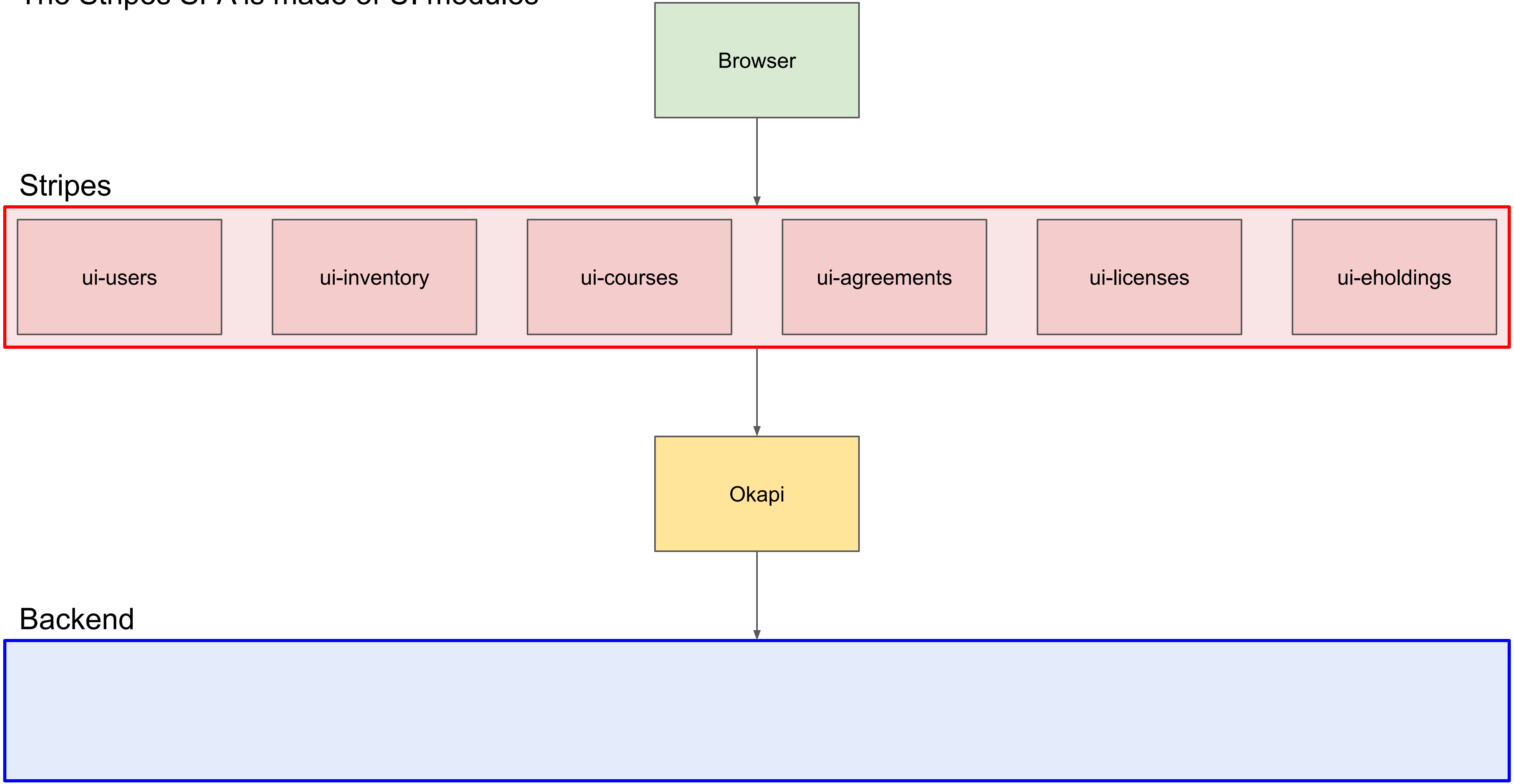
… calling Okapi-mediated services …



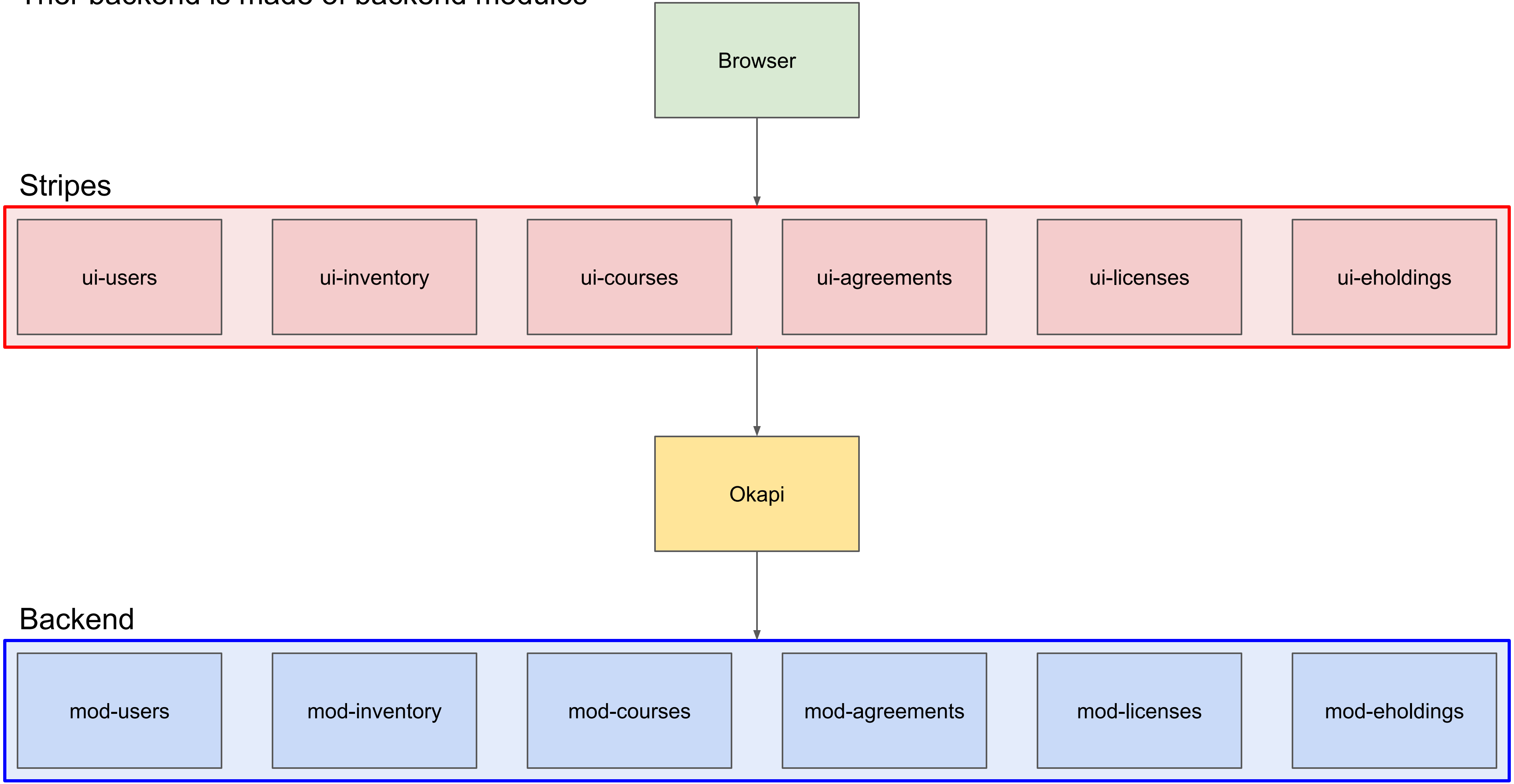
… that are provided by a backend.



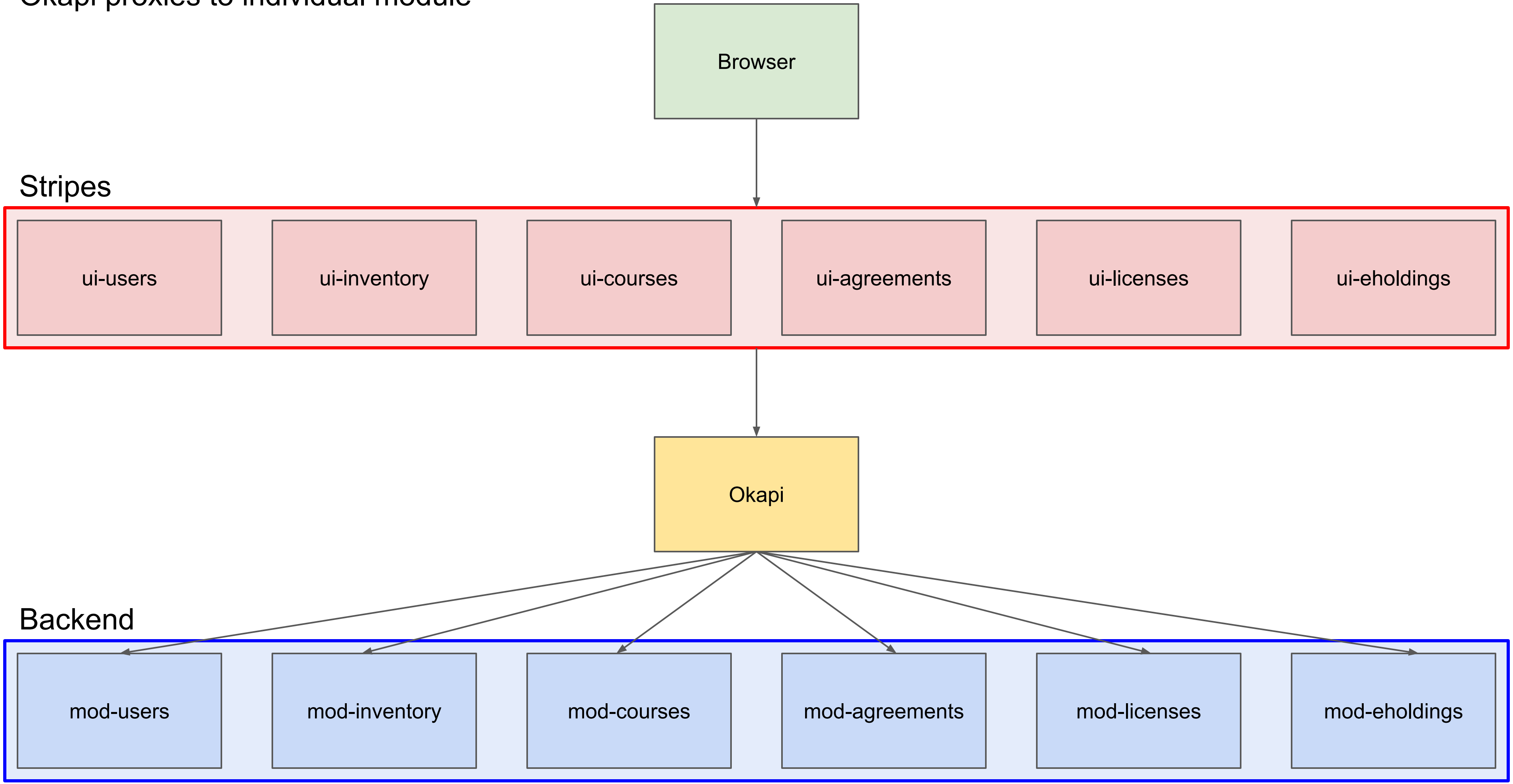
The Stripes SPA is made of UI modules



Ther backend is made of backend modules



Okapi proxies to individual module



Browser

Stripes

ui-users

ui-inventory

ui-courses

ui-agreements

ui-licenses

ui-eholdings

Okapi

Backend

mod-users

mod-inventory

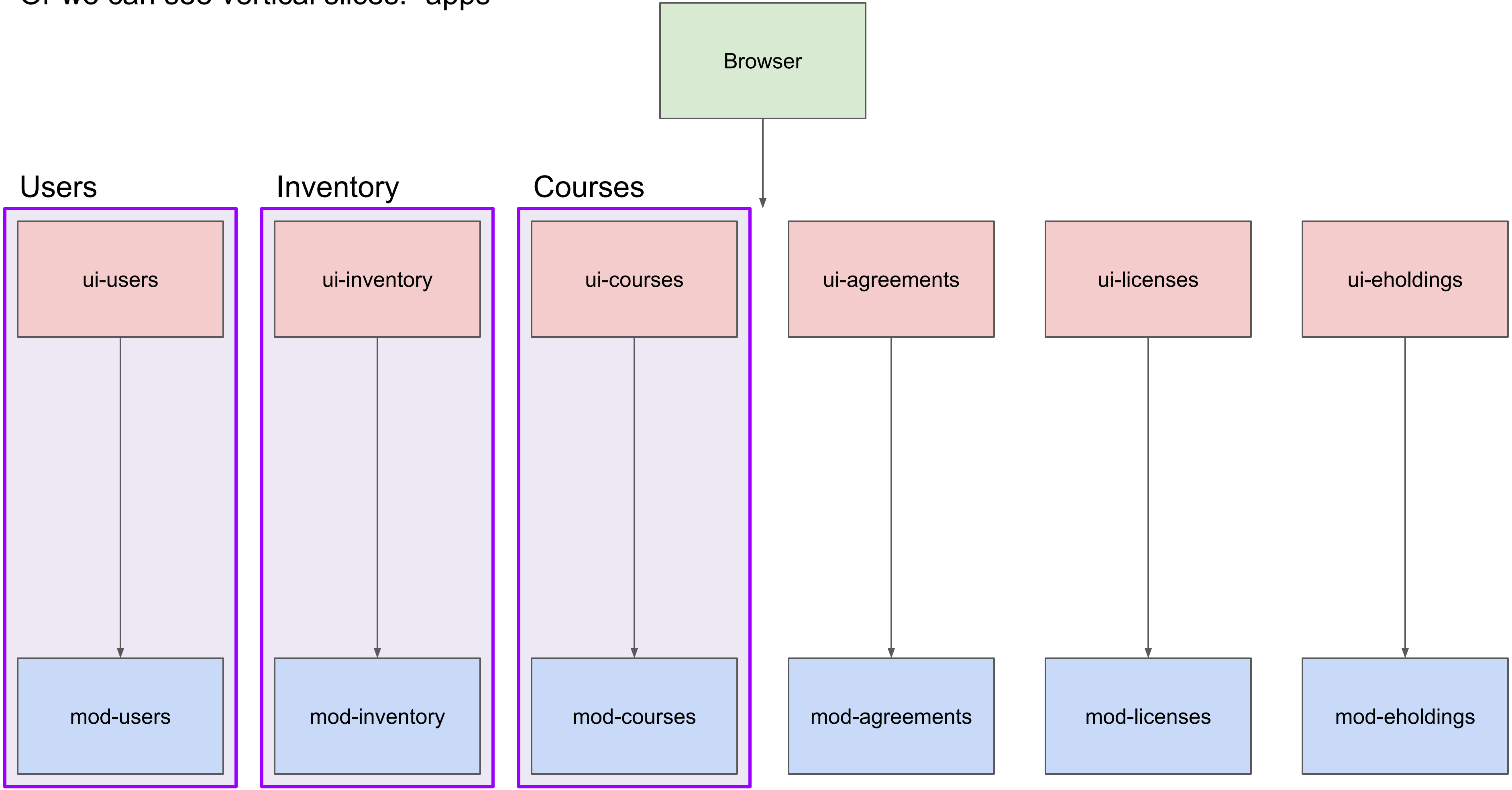
mod-courses

mod-agreements

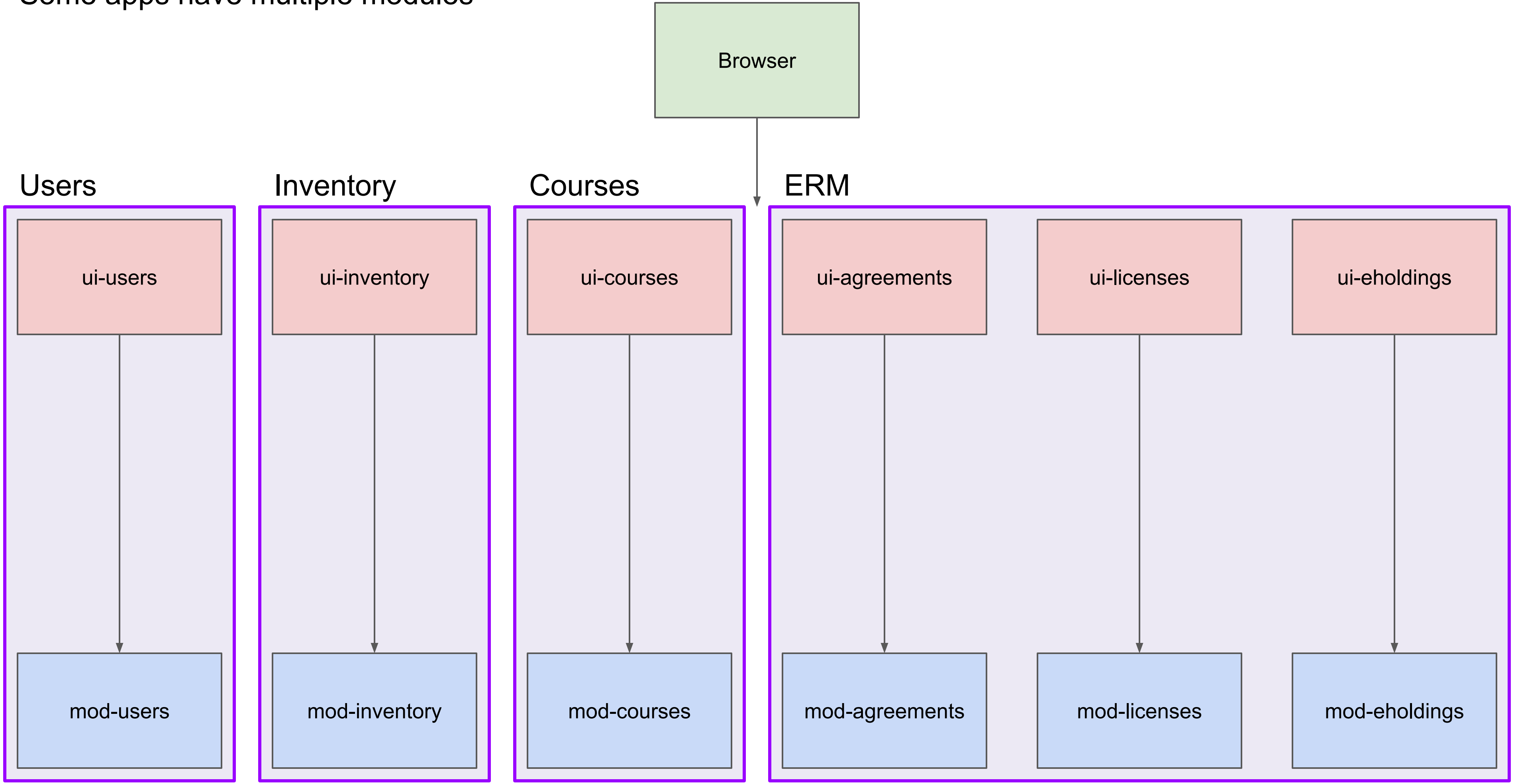
mod-licenses

mod-eholdings

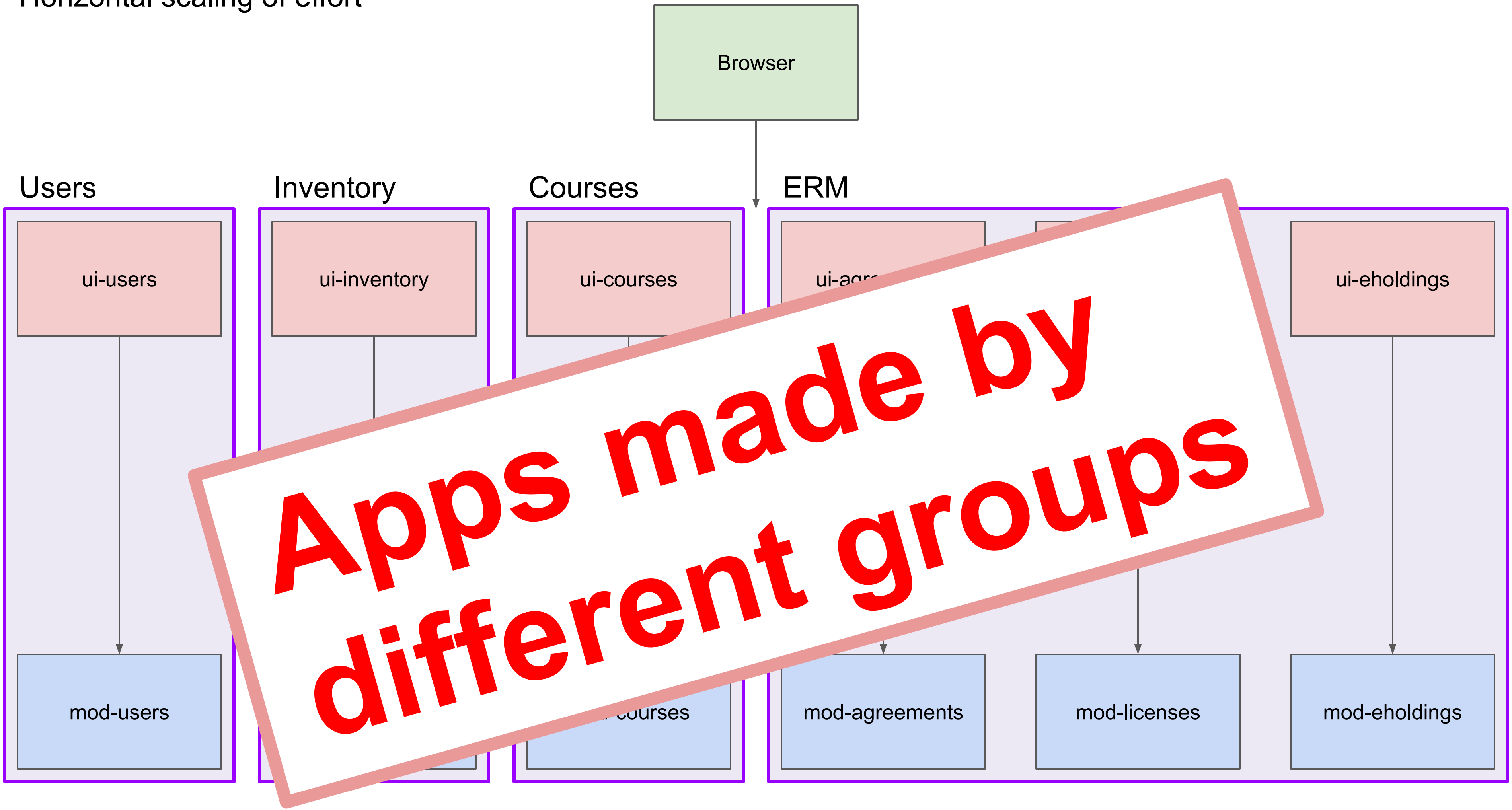
Or we can see vertical slices: “apps”



Some apps have multiple modules

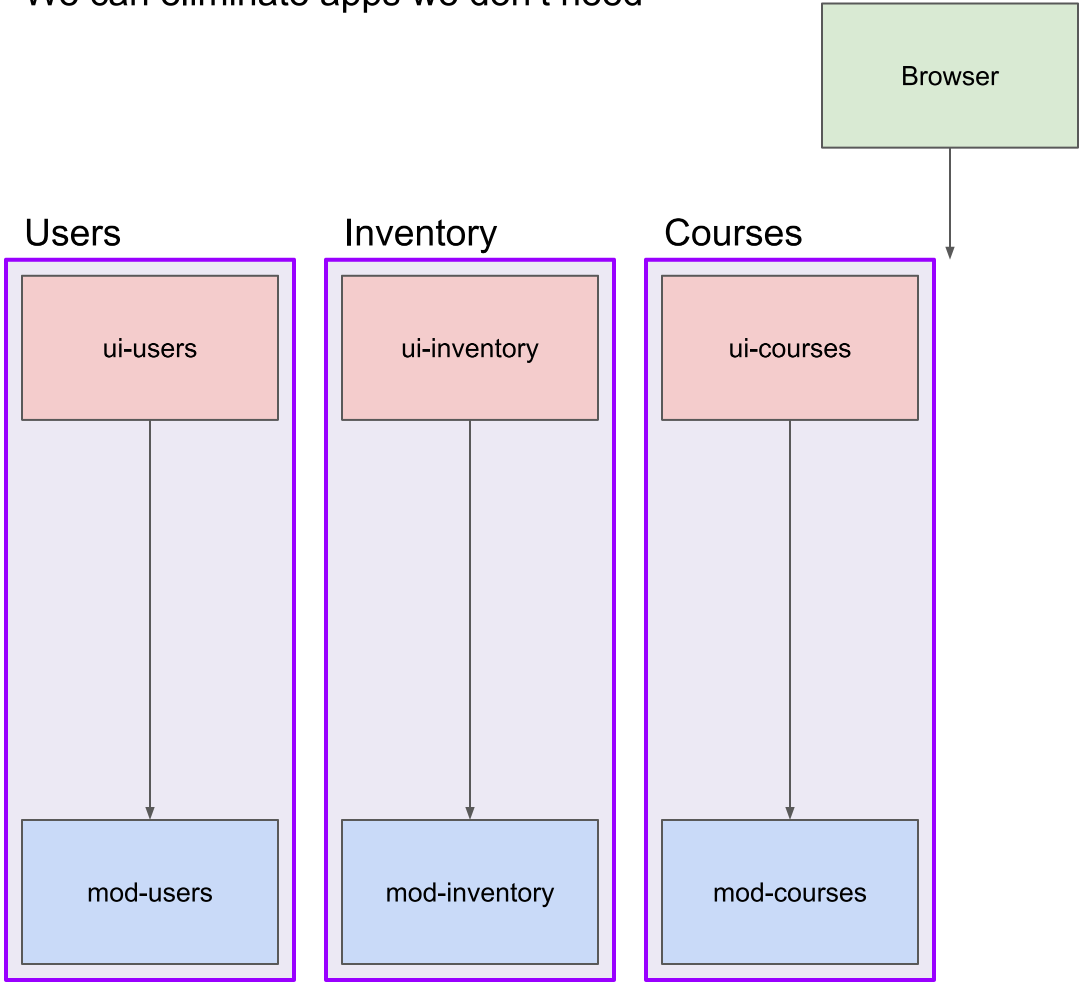


Horizontal scaling of effort

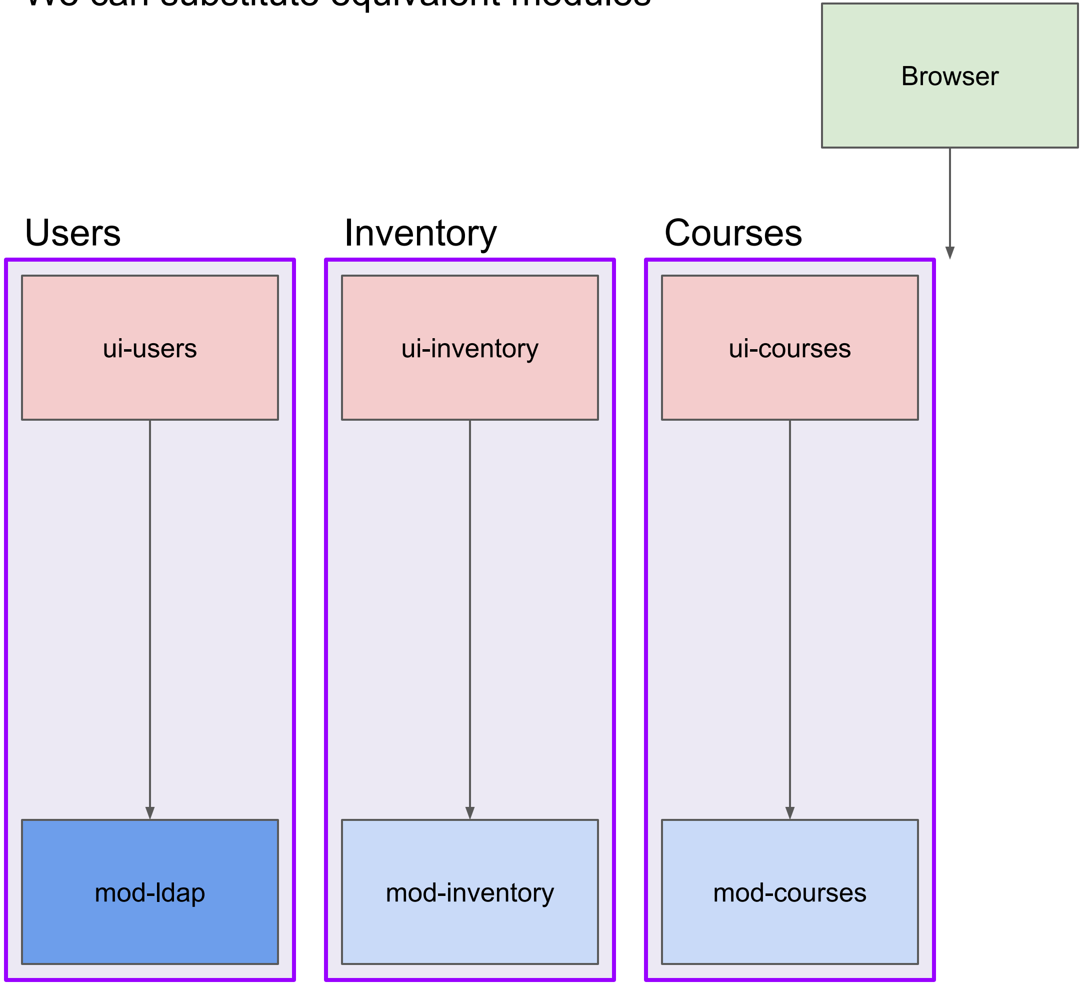


Apps made by different groups

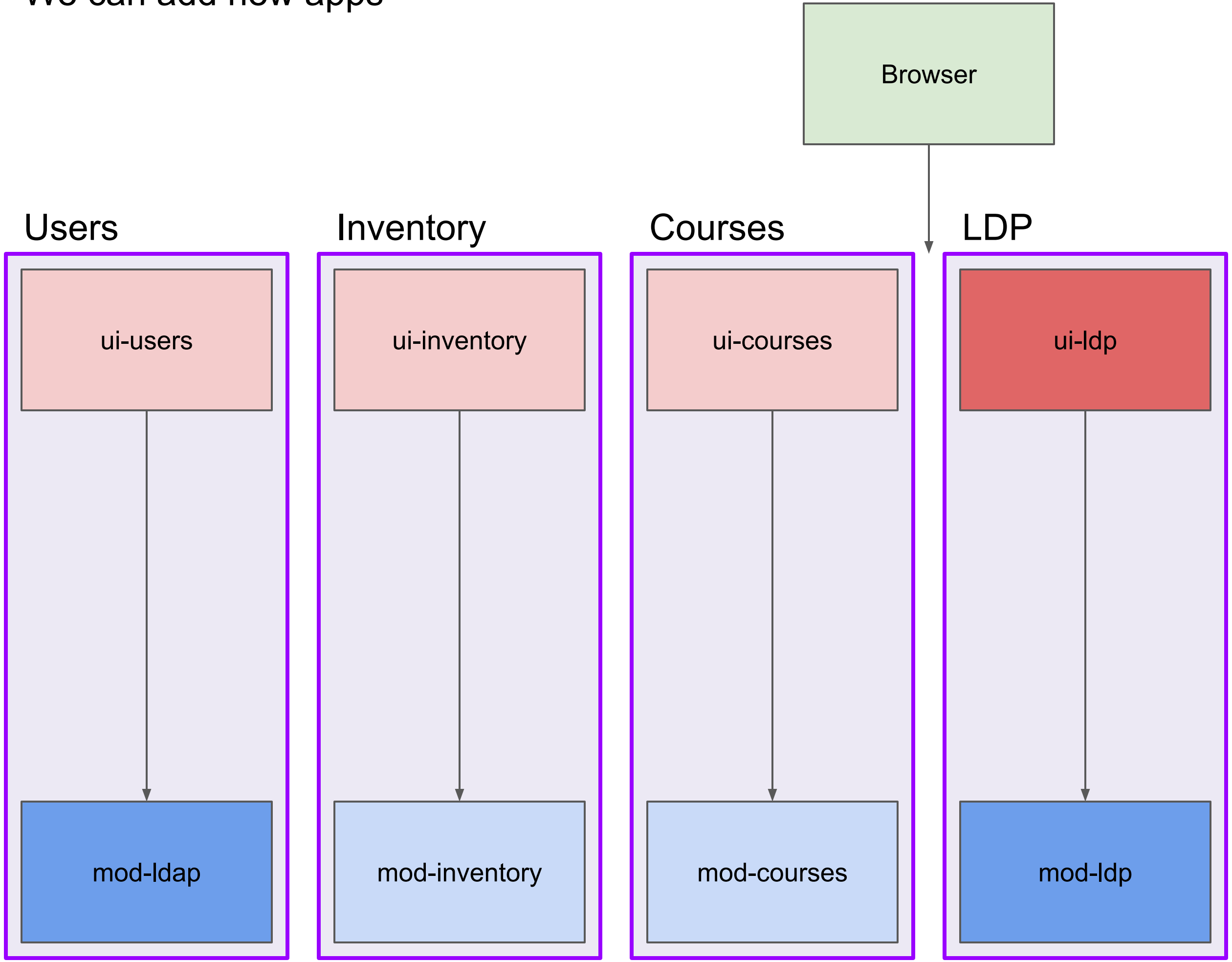
We can eliminate apps we don’t need



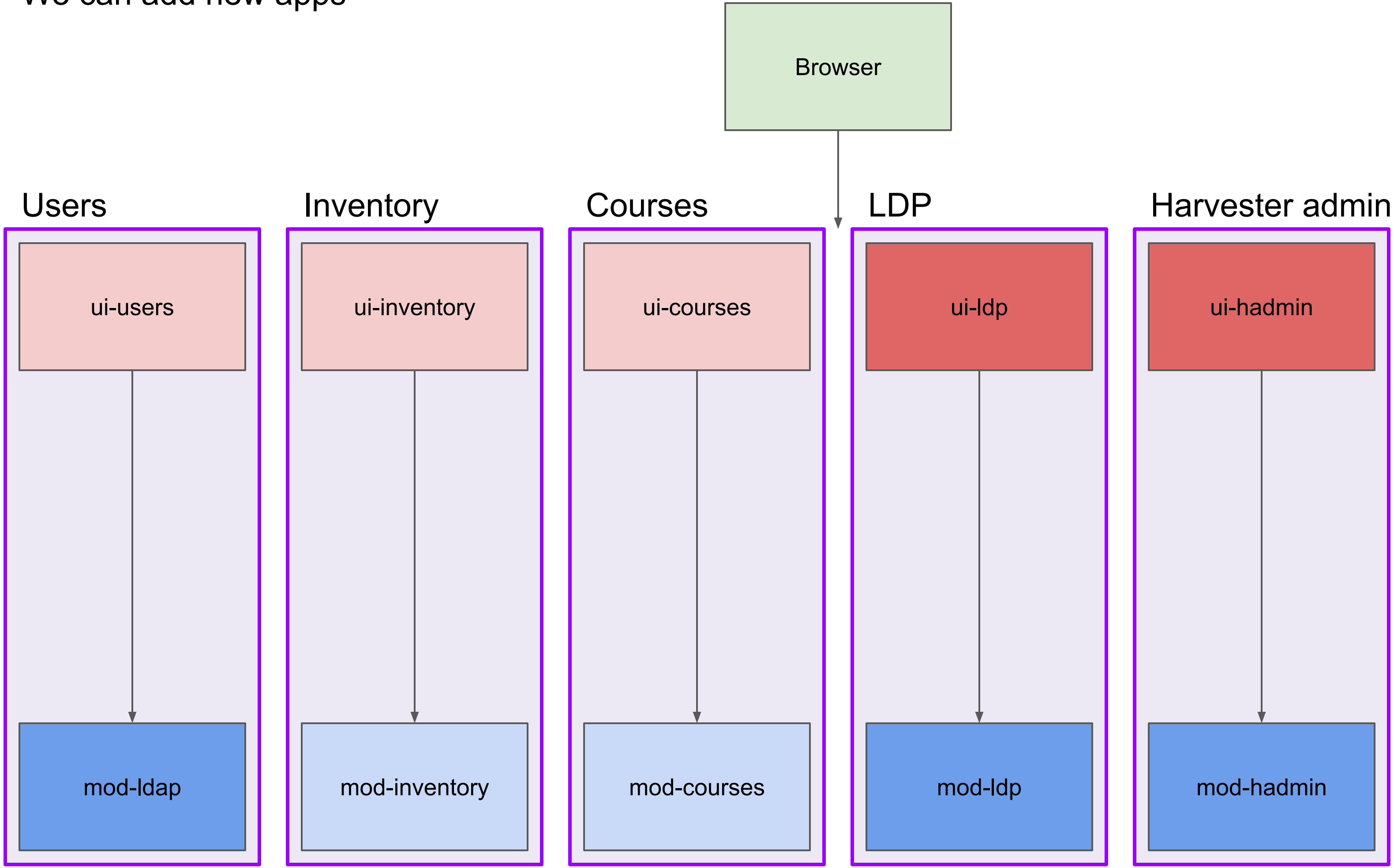
We can substitute equivalent modules



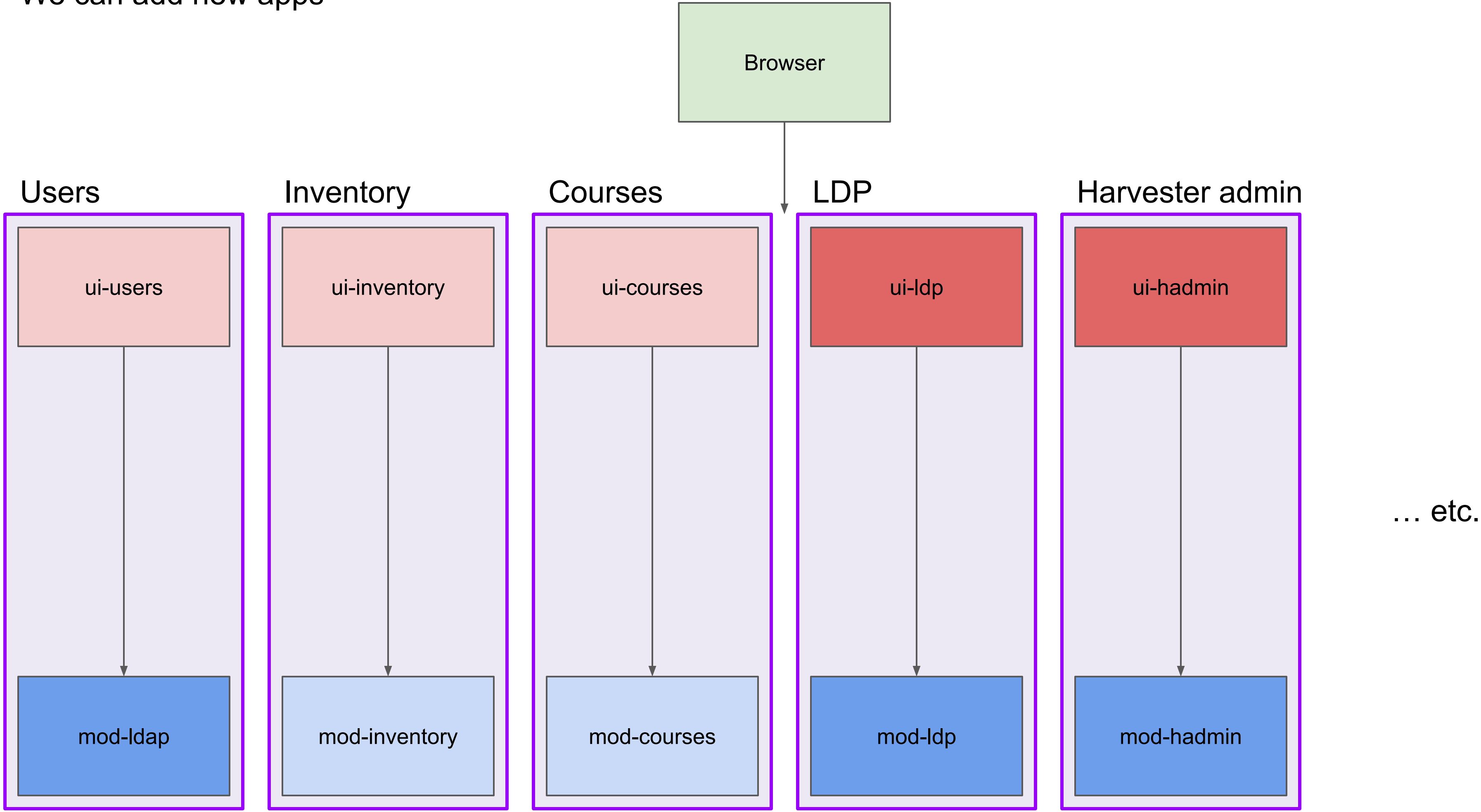
We can add new apps



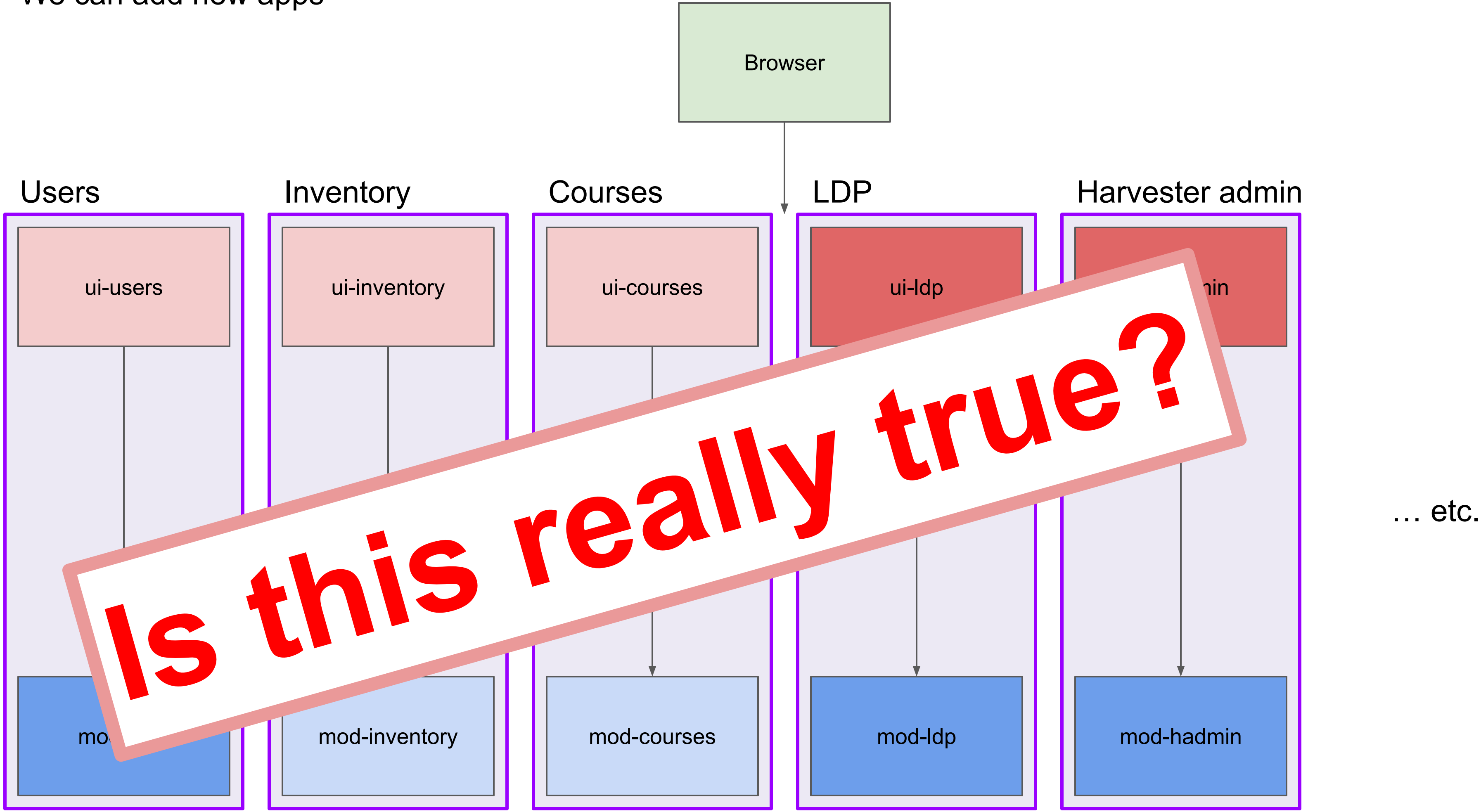
We can add new apps



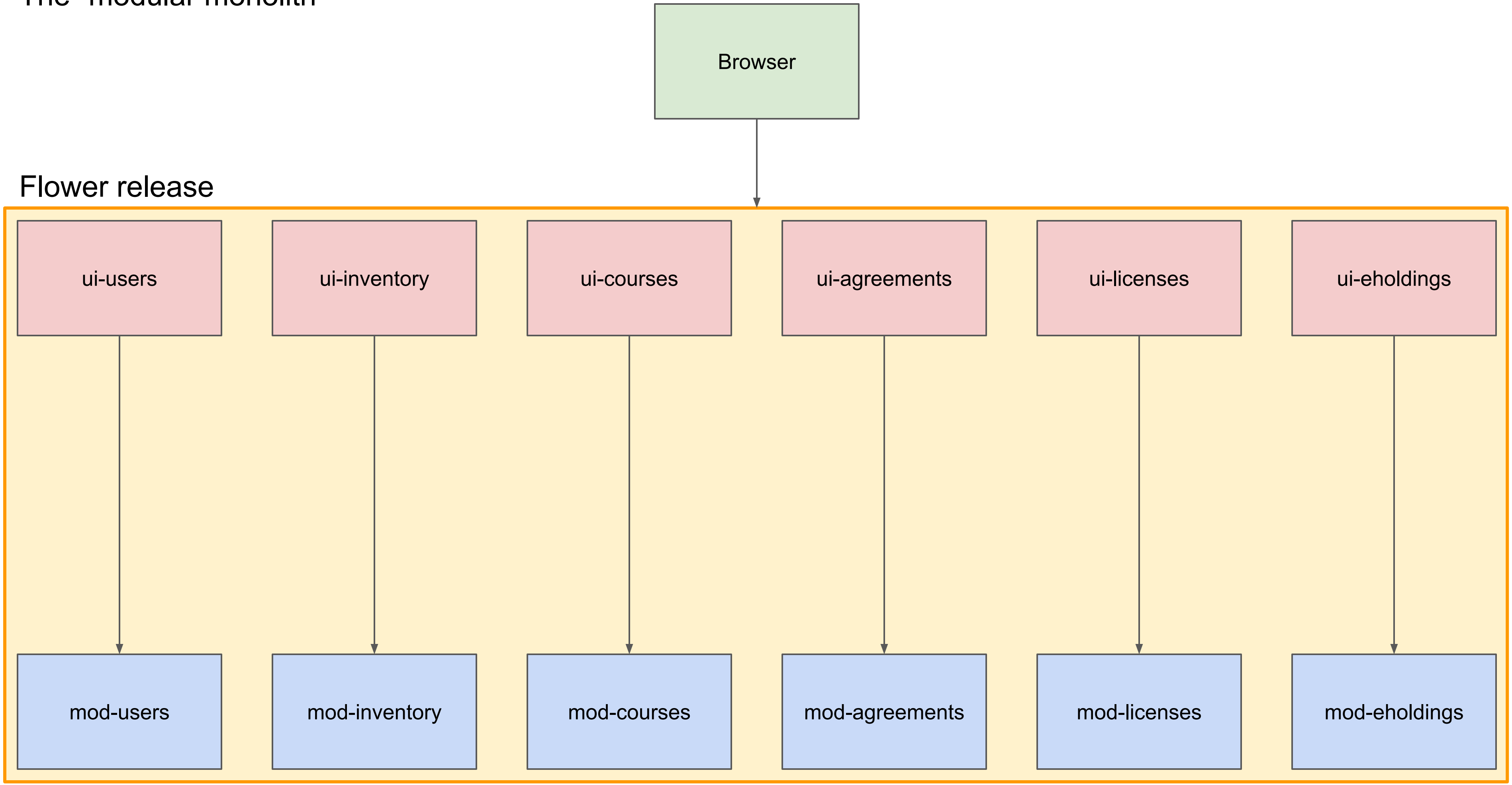
We can add new apps



We can add new apps



The “modular monolith”



Browser

Flower release

ui-users

ui-inventory

ui-courses

ui-agreements

ui-licenses

ui-eholdings

mod-users

mod-inventory

mod-courses

mod-agreements

mod-licenses

mod-eholdings

Flower releases

- Stable, tested, reliable
- But the result is that we usually have all and only the core set of modules

Flower releases

- It’s hard to get new code into a flower release (for mostly good reasons)

Flower releases

- It’s hard to get new code into a flower release (for mostly good reasons)
- Where is the flourishing ecosystem of competing apps from different providers?

Result: a modular monolith

1. The base platform is too big to build custom FOLIO subsets on
2. There are technical difficulties in adding apps not in the
3. There is no way to package modules tog
4. There is no way to certify th ch as internationali
5. There load apps, or download them from
6. There ver apps that do exist
7. Installing apps (UI and backend modules) is cumbersome
8. There is no Okapi Console that can be used to enable/disable modules, etc.

Don't read this

Our team is addressing all these issues

Our team is addressing all these issues

- Index Data team:
  - Mike Taylor (dev)
  - Wayne Schneider (ops)
  - Jason Skomorowski (dev)
  - David Crossley (ops)

Our team is addressing all these issues

- Index Data team:
  - Mike Taylor (dev)
  - Wayne Schneider (ops)
  - Jason Skomorowski (dev)
  - David Crossley (ops)

- With help from:
  - Michal Kuklis, Cornell (UI developer)
  - Zak Burke (EBSCO) (actual wizard)

1. Smaller, lighter base platform (backend)

- We often want to add modules, but we also want to pare down to a lean subset
- We want to be able to make lean standalone apps, e.g. checkin/checkout

1. Smaller, lighter base platform (backend)

- We often want to add modules, but we also want to pare down to a lean subset
- We want to be able to make lean standalone apps, e.g. checkin/checkout
- At present the only platform supported by the OLF is `platform-complete` (which is what we use to build https://folio-snapshot.dev.folio.org/)
- This has 140 backend and edge modules, 29 UI apps, 30 UI and 18 UI plugins

1. Smaller, lighter base platform (backend)

- We often want to add modules, but we also want to pare down to a lean subset
- We want to be able to make lean standalone apps, e.g. checkin/checkout
- At present the only platform supported by the OLF is `platform-complete` (which is what we use to build https://folio-snapshot.dev.folio.org/)
- This has 140 backend and edge modules, 29 UI apps, 30 UI and 18 UI plugins

- **We are working on a much smaller `platform-core`**

1. Smaller, lighter base platform (UI)

- Difficult because Stripes logs in with `mod-users-bl` instead of `mod-login`, and that pulls in many other modules
- We need to remove the hardwired concept of "service points" from Stripes

1. Smaller, lighter base platform (UI)

- Difficult because Stripes logs in with `mod-users-bl` instead of `mod-login`, and that pulls in many other modules
- We need to remove the hardwired concept of "service points" from Stripes

- **We have prototype code for a leaner approach to login**
- **We have a “handler” system that can assign service-points**

2. Remove unexamined assumptions in the code

- Historically, all Stripes packages had to be in the `@folio` namespace

2. Remove unexamined assumptions in the code

- Historically, all Stripes packages had to be in the `@folio` namespace

- **We have code to support UI modules in any namespace**

3. A way of packaging apps

- UI modules exist, and backend modules exist, but there is no app package.

3. A way of packaging apps

- UI modules exist, and backend modules exist, but there is no app package.

- **FAM format (FOLIO Application Metadata)**
- **JSON-based**

3. A way of packaging apps: FAM format

```
{
  "name": "harvester-admin",
  "version": "0.0.1"
}
```

3. A way of packaging apps: FAM format

```
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester"
}
```

3. A way of packaging apps: FAM format

```
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester",
  "elements": [
    {
      "type": "ui",
      "required": true
    }
  ]
}
```

3. A way of packaging apps: FAM format

```json
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester",
  "elements": [
    {
      "type": "ui",
      "required": true,
      "descriptor": "https://reg.indexdata.com/_/proxy/modules/indexdata_harvester-admin-0.1.0"
    }
  ]
}
```

3. A way of packaging apps: FAM format

```
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester",
  "elements": [
    {
      "type": "ui",
      "required": true,
      "descriptor": "https://reg.indexdata.com/_/proxy/modules/indexdata_harvester-admin-0.1.0",
      "url": "https://repository.folio.org/repository/npm-folio/@indexdata/harvester-admin"
    }
  ]
}
```

3. A way of packaging apps: FAM format

```
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester",
  "elements": [
    {
      "type": "ui",
      "required": true,
      "descriptor": "https://reg.indexdata.com/_/proxy/modules/indexdata_harvester-admin-0.1.0",
      "url": "https://repository.folio.org/repository/npm-folio/@indexdata/harvester-admin"
    },
    {
      "type": "backend",
      "required": true,
      "descriptor": "https://reg.indexdata.com/_/proxy/modules/mod-harvester-admin-0.1.0-SNAPSHOT",
      "url": "ghcr.io/indexdata/mod-harvester-admin:v0.1.0-SNAPSHOT.7"
    }
  ]
}
```

4. A way of certifying apps

- We need to know who published the app and is taking responsibility for it

4. A way of certifying apps

- We need to know who published the app and is taking responsibility for it
- We also want to know who has made claims about
  - UX design
  - internationalization
  - accessibility
  - security
  - *etc.*

4. A way of certifying apps

```
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester",
  "elements": [...]
}
```

4. A way of certifying apps

```
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester",
  "elements": [...],
  "certified": [
    {
      "type": "published",
      "certifier": "indexdata.com",
    }
  ]
}
```

4. A way of certifying apps

```
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester",
  "elements": [...],
  "certified": [
    {
      "type": "published",
      "certifier": "indexdata.com",
      "signature": "1a2b3c4d5e6f7g8h9i0j"
    }
  ]
}
```

4. A way of certifying apps

```
{
  "name": "harvester-admin",
  "version": "0.0.1",
  "displayName": "Harvester admin",
  "description": "Admin console for the Index Data Harvester",
  "elements": [...],
  "certified": [
    {
      "type": "published",
      "certifier": "indexdata.com",
      "signature": "1a2b3c4d5e6f7g8h9i0j"
    },
    {
      "type": "ux",
      "certifier": "samhaeng.com",
      "signature": "1234567890abcdefghij"
    }
  ]
}
```

5. A place to upload apps and their components

An app consists of several parts:

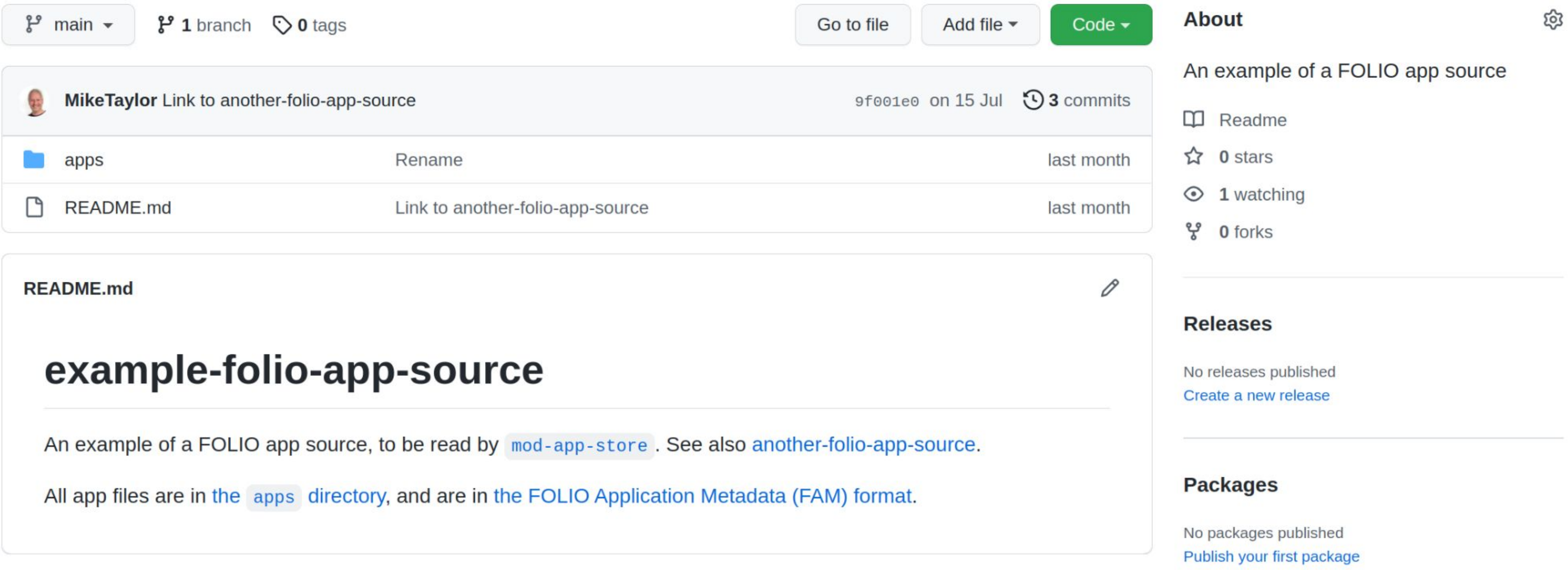
- The FAM file
- Module descriptors for UI and backend modules
- NPM packages for UI modules
- Docker containers for backend modules

They all need to go somewhere.

5. A place to upload apps: FAM files

FAM files in GitHub repositories

e.g. https://github.com/MikeTaylor/example-folio-app-source



An example of a FOLIO app source

MikeTaylor Link to another-folio-app-source 9f001e0 on 15 Jul 3 commits

apps	Rename	last month
README.md	Link to another-folio-app-source	last month

README.md

example-folio-app-source

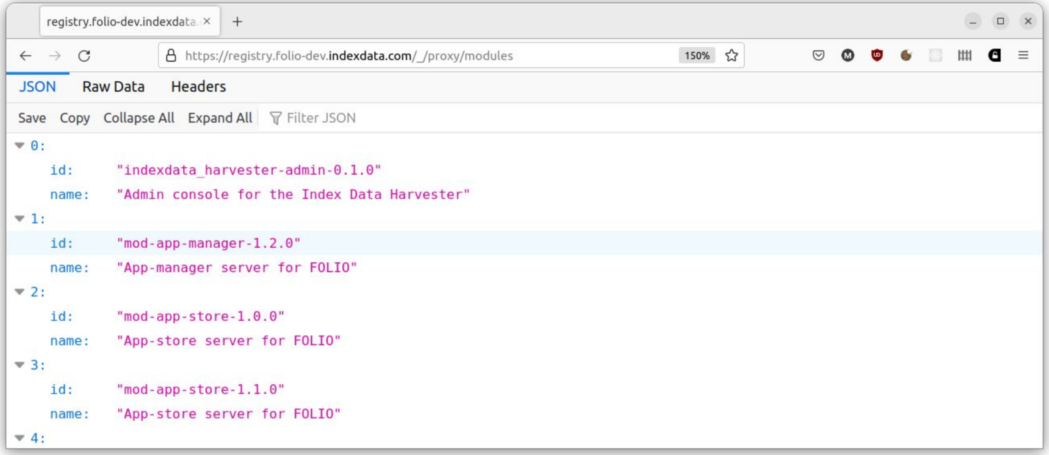
An example of a FOLIO app source, to be read by `mod-app-store`. See also another-folio-app-source.

All app files are in the `apps` directory, and are in the FOLIO Application Metadata (FAM) format.

5. A place to upload apps: module descriptors

Module descriptors in publicly accessible Okapi instances:

- The FOLIO project's registry at http://folio-registry.aws.indexdata.com
- Index Data's registry at http://registry.folio-dev.indexdata.com
- Any other registry that anyone stands up



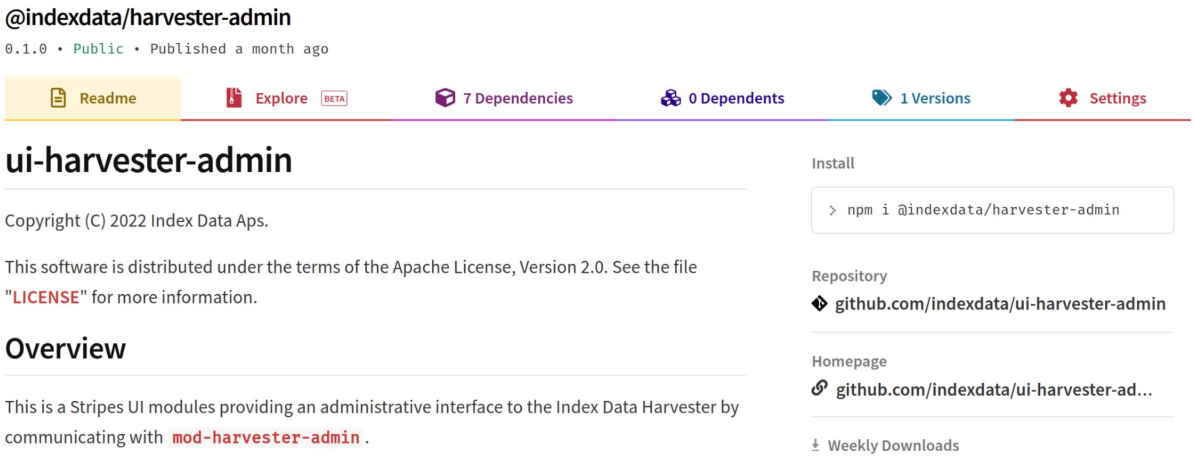
https://registry.folio-dev.indexdata.com/_/proxy/modules

```
▼ 0:
    id:     "indexdata_harvester-admin-0.1.0"
    name:   "Admin console for the Index Data Harvester"
▼ 1:
    id:     "mod-app-manager-1.2.0"
    name:   "App-manager server for FOLIO"
▼ 2:
    id:     "mod-app-store-1.0.0"
    name:   "App-store server for FOLIO"
▼ 3:
    id:     "mod-app-store-1.1.0"
    name:   "App-store server for FOLIO"
▼ 4:
```

5. A place to upload apps: NPM packages

NPM packages in any NPM registry:

- The FOLIO NPM registry (where core modules live)
- The global NPM registry
- Any other registry that anyone stands up



@indexdata/harvester-admin

0.1.0 • Public • Published a month ago

ui-harvester-admin

Copyright (C) 2022 Index Data Aps.

This software is distributed under the terms of the Apache License, Version 2.0. See the file "LICENSE" for more information.

Overview

This is a Stripes UI modules providing an administrative interface to the Index Data Harvester by communicating with `mod-harvester-admin`.

Install

```
> npm i @indexdata/harvester-admin
```

Repository

github.com/indexdata/ui-harvester-admin

Homepage

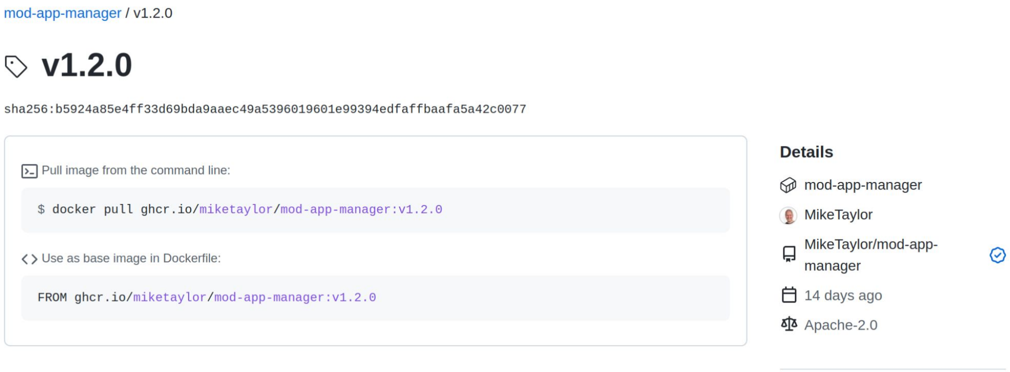
github.com/indexdata/ui-harvester-ad...

Weekly Downloads

5. A place to upload apps: Docker containers

Docker containers in any Docker registry:

- DockerHub
- FOLIO's DockerHub mirror
- GitHub packages at https://ghcr.io
- Any other registry that anyone stands up



mod-app-manager / v1.2.0

v1.2.0

sha256:b5924a85e4ff33d69bda9aaec49a5396019601e99394edfaffbaafa5a42c0077

Pull image from the command line:

```
$ docker pull ghcr.io/miketaylor/mod-app-manager:v1.2.0
```

Use as base image in Dockerfile:

```
FROM ghcr.io/miketaylor/mod-app-manager:v1.2.0
```

Details

- mod-app-manager
- MikeTaylor
- MikeTaylor/mod-app-manager
- 14 days ago
- Apache-2.0

5. Where should you upload your artifacts?

- We at Index Data have made our own choices
- Yours may be different, and that's fine.
- We don't want an Index Data monolith,
  parallel with the existing FOLIO monolith
- We want an ecosystem populated by many actors

6. Towards an app-store

- Anyone can publish FAM files, but how can they be discovered?

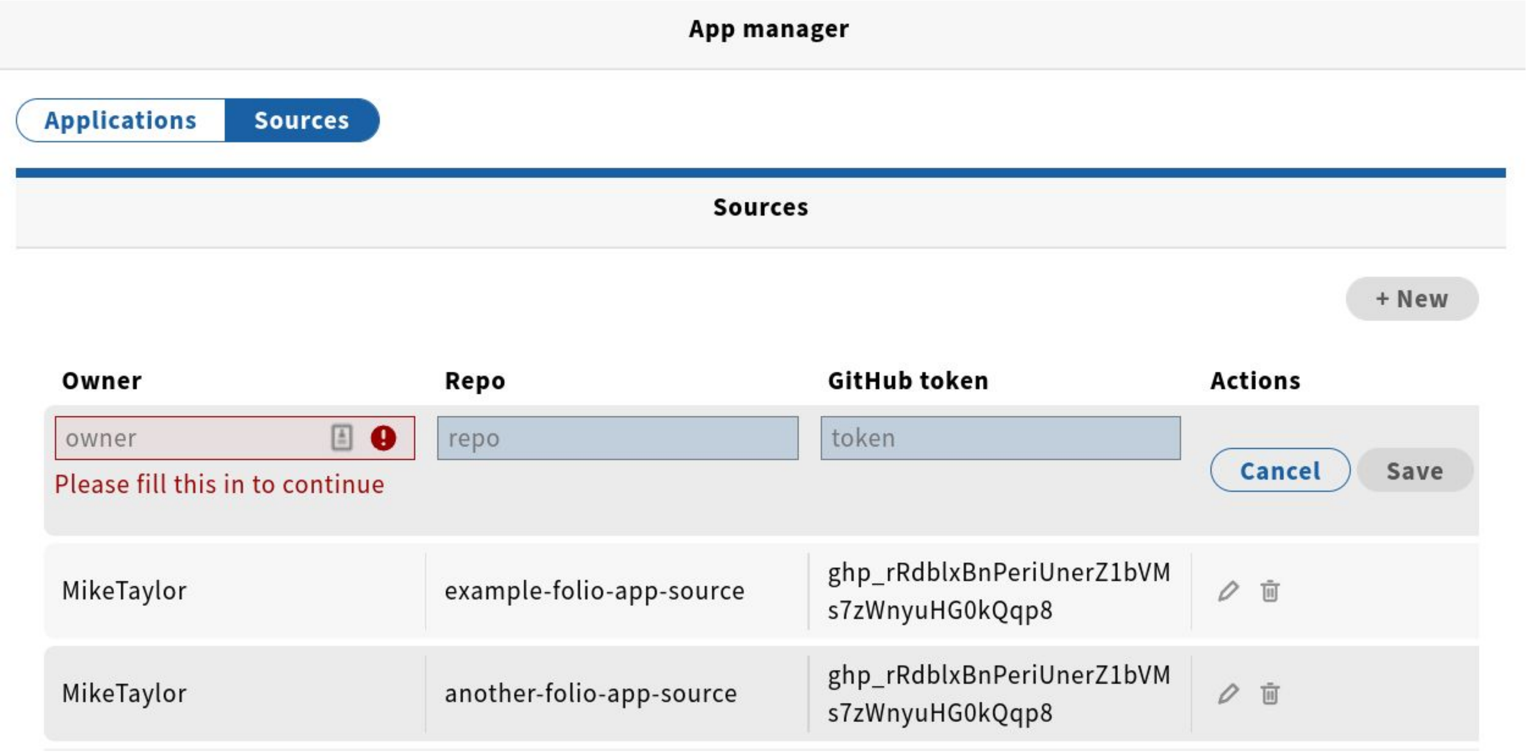
6. Towards an app-store

- Anyone can publish FAM files, but how can they be discovered?

- **We have a prototype `mod-app-manager`**
- **Its configuration stores details about GitHub repositories**
- **It can return all the FAM files from all those repositories**

6. Towards an app-store

We have a primitive `mod-app-manager` UI in a branch of the Developer Console

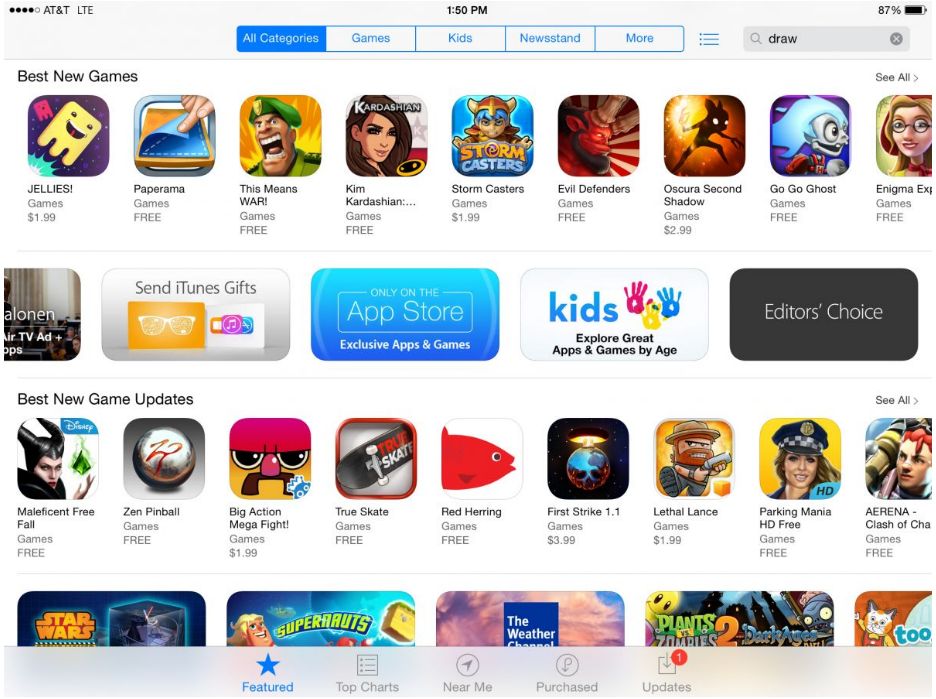


Owner	Repo	GitHub token	Actions
owner Please fill this in to continue	repo	token	Cancel Save
MikeTaylor	example-folio-app-source	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	
MikeTaylor	another-folio-app-source	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	

6. Towards an app-store

This is the beginning of an app-store that can be used in several ways

- Platform administrators:
  - Add and deploy apps
- Tenant administrators
  - Enable and disable apps
- Users
  - Make installation requests



7. Installing apps

Installation is a cumbersome specialist process involving devops time

7. Installing apps

Installation is a cumbersome specialist process involving devops time

We have a script, `add-app-to-platform`

- **It reads a FAM file**
- **It inserts the module descriptors of all elements**
- **It deploys backend modules**

7. Installing apps

Installation is a cumbersome specialist process involving devops time

We have a script, `add-app-to-platform`

- **It reads a FAM file**
- **It inserts the module descriptors of all elements**
- **It deploys backend modules**

- **(It does not yet build and publish new Stripes bundles)**

7. Installing apps

Deployment is different for different vendors

7. Installing apps

Deployment is different for different vendors

`add-app-to-platform` uses a simple deployment-plugin API

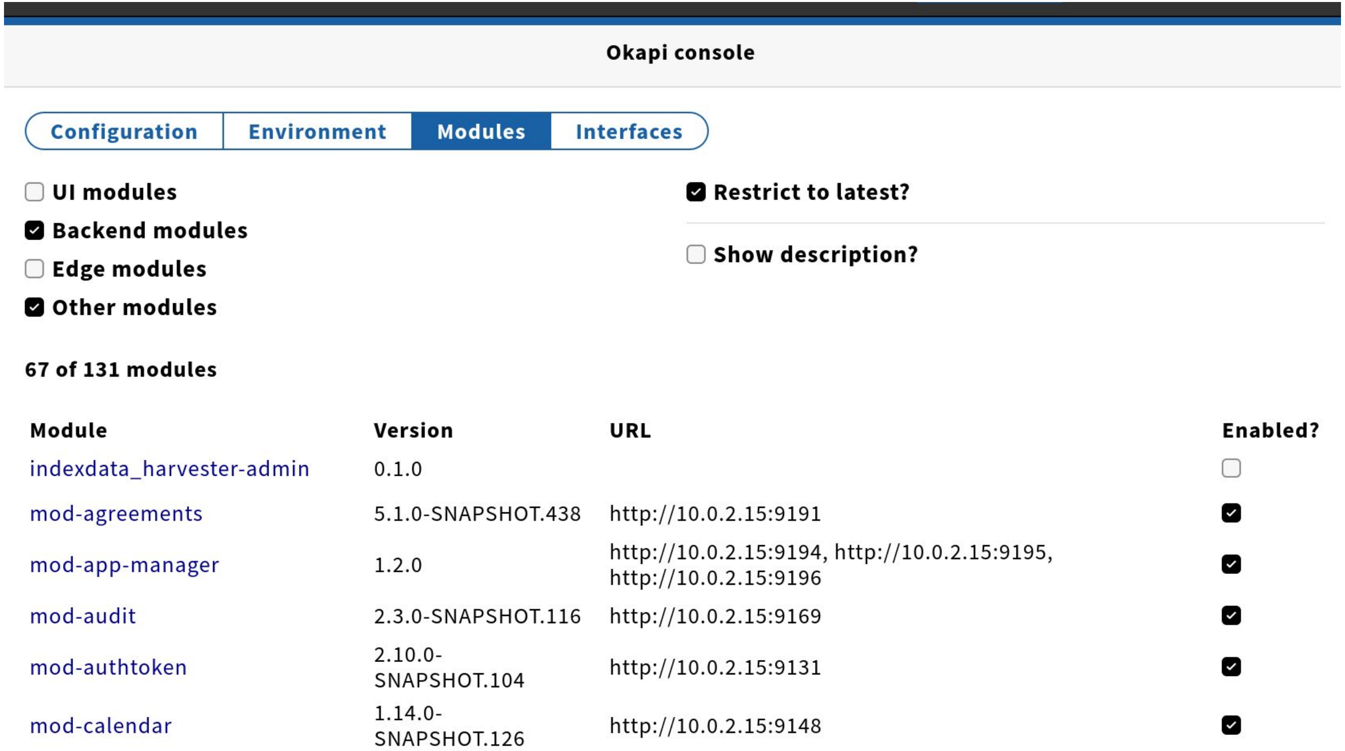
- **There is currently an `okapi` plugin using Okapi's deployment mechanism**
- **We plan a Kubernetes plugin**
- **Others are possible, e.g. DockerSwarm**

7. Installing apps

- We want to build a FOLIO module that can do these things
- Then we can make a UI to allow all this to be done within FOLIO itself

8. The Okapi console

- **Prototype is currently hidden in Developer Settings**



Okapi console

☐ **UI modules**
☑ **Backend modules**
☐ **Edge modules**
☑ **Other modules**

☑ **Restrict to latest?**

☐ **Show description?**

67 of 131 modules

Module	Version	URL	Enabled?
indexdata_harvester-admin	0.1.0		☐
mod-agreements	5.1.0-SNAPSHOT.438	http://10.0.2.15:9191	☑
mod-app-manager	1.2.0	http://10.0.2.15:9194, http://10.0.2.15:9195, http://10.0.2.15:9196	☑
mod-audit	2.3.0-SNAPSHOT.116	http://10.0.2.15:9169	☑
mod-authtoken	2.10.0-SNAPSHOT.104	http://10.0.2.15:9131	☑
mod-calendar	1.14.0-SNAPSHOT.126	http://10.0.2.15:9148	☑

Conclusion: why this matters

- We want everyone to be able to contribute — not just a few anointed organizations
- We want the barrier-to-entry to be very low, while retaining quality stamps
- We want no single chokepoint to making apps available in FOLIO
- We want a functioning ecosystem with competition
- We want FOLIO to outlive EBSCO and Index Data

FOLIO modularity in practice

Seamless deployment of modules from multiple sources

Mike Taylor
Index Data
mike@indexdata.com