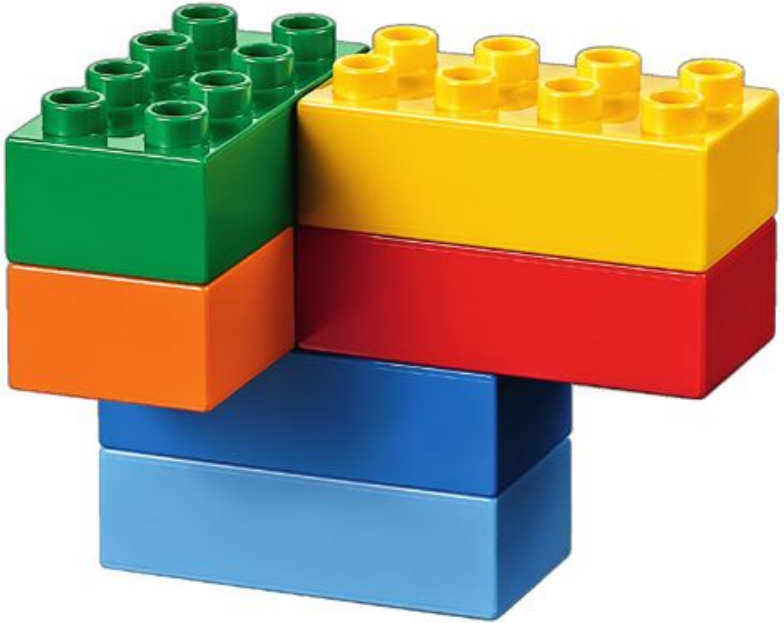
A case-study in FOLIO modularity: replacing mod-ldp with mod-reporting



Mike Taylor
Software Guy
Index Data
mike@indexdata.com

Where we're going

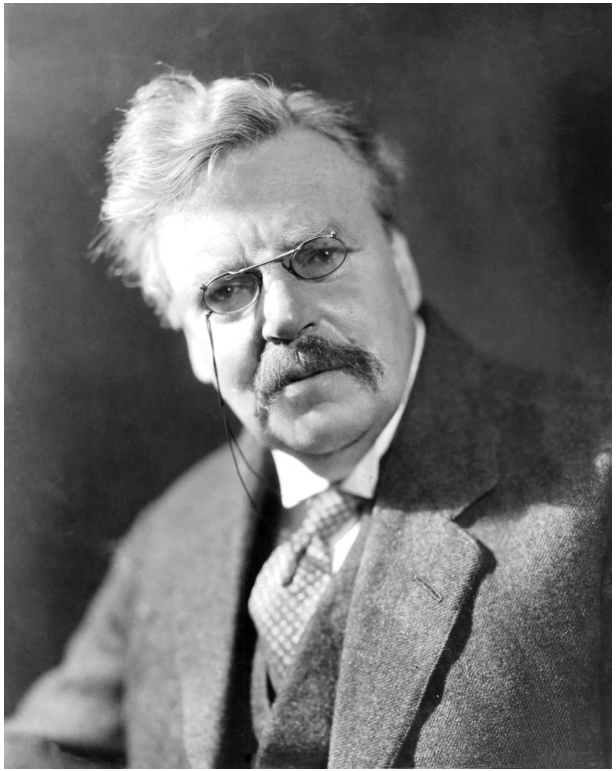
1. Philosophy
2. Modularity in FOLIO
3. Benefits of the modular architecture
4. An ecosystem of modules
5. Some light technical details
6. A case-study: mod-reporting
7. Implications

1. Philosophy

"I'm afraid I'm a practical man", said the doctor with gruff humour, "and I don't bother much about [...] philosophy."

"You'll never be a practical man till you do", said Father Brown.

— G. K. Chesterton,
The Dagger with Wings.



1. Philosophy

The design of FOLIO arose from a philosophy

1. Philosophy

The design of FOLIO arose from a philosophy

- Communities **do things better** than top-down organizations.

1. Philosophy

The design of FOLIO arose from a philosophy

- Communities **do things better** than top-down organizations.
- Communities **do better things** than top-down organizations.

1. Philosophy

The design of FOLIO arose from a philosophy

- Communities **do things better** than top-down organizations.
- Communities **do better things** than top-down organizations.
- FOLIO belongs to the library community, not to a vendor.

1. Philosophy

The design of FOLIO arose from a philosophy

- Communities **do things better** than top-down organizations.
- Communities **do better things** than top-down organizations.
- FOLIO belongs to the library community, not to a vendor.
- Any member of the community should be able to contribute.

1. Philosophy

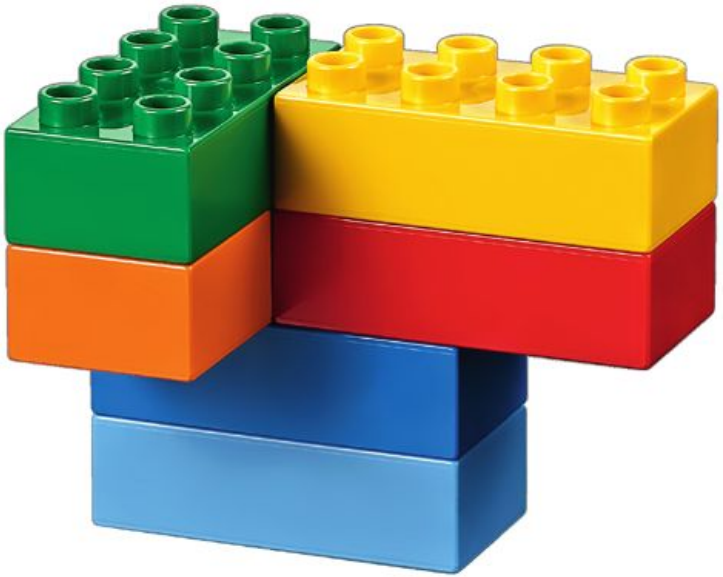
The design of FOLIO arose from a philosophy

- Communities **do things better** than top-down organizations.
- Communities **do better things** than top-down organizations.
- FOLIO belongs to the library community, not to a vendor.
- Any member of the community should be able to contribute.

No special access for insiders: we eat our own dogfood.

2. Modularity in FOLIO

FOLIO's community philosophy is reflected in its technical design: from the start, it was constructed to be modular.



2. Modularity in FOLIO

FOLIO's community philosophy is reflected in its technical design: from the start, it was constructed to be modular.

Modules are the unit of contribution.

2. Modularity in FOLIO

FOLIO's community philosophy is reflected in its technical design: from the start, it was constructed to be modular.

Modules are the unit of contribution.

But what about contributing in other ways?

- Requirements
- Documentation
- Code review
- Feedback

2. Modularity in FOLIO

FOLIO's community philosophy is reflected in its technical design: from the start, it was constructed to be modular.

Modules are the unit of **owned** contribution.

But what about contributing in other ways?

- Requirements
- Documentation
- Code review
- Feedback

2. Modularity in FOLIO

FOLIO's community philosophy is reflected in its technical design: from the start, it was constructed to be modular.

Modules are the unit of **owned** contribution.

"Here is a thing we made"

2. Modularity in FOLIO

The heart of every FOLIO system is a platform consisting of:

- Okapi, the API gateway that manages modules

2. Modularity in FOLIO

The heart of every FOLIO system is a platform consisting of:

- Okapi, the API gateway that manages modules
- Well-defined specifications for what a module is (discussed below)

2. Modularity in FOLIO

The heart of every FOLIO system is a platform consisting of:

- Okapi, the API gateway that manages modules
- Well-defined specifications for what a module is (discussed below)
- A handful of core modules to handle authentication and related concerns

2. Modularity in FOLIO

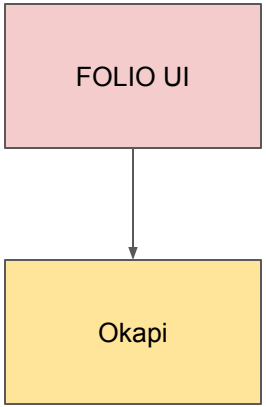
The heart of every FOLIO system is a platform consisting of:

- Okapi, the API gateway that manages modules
- Well-defined specifications for what a module is (discussed below)
- A handful of core modules to handle authentication and related concerns

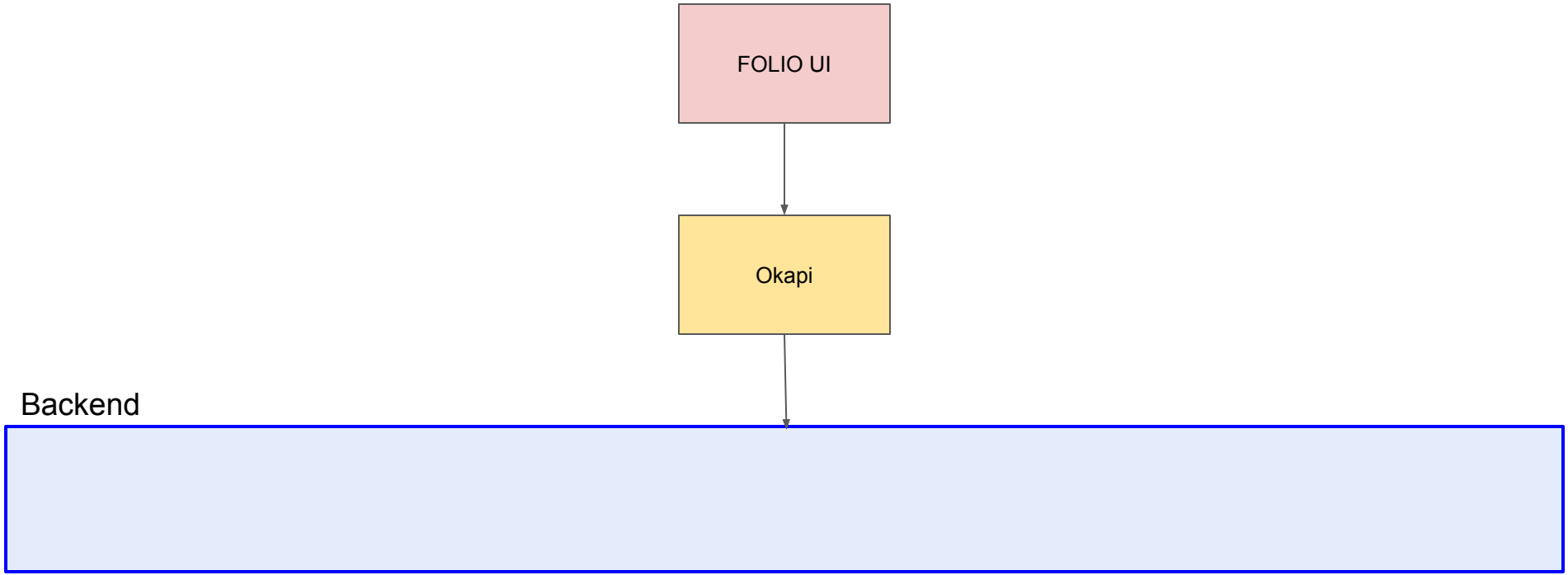
This provides a context in which any module from any source can be deployed.

2. Modularity in FOLIO

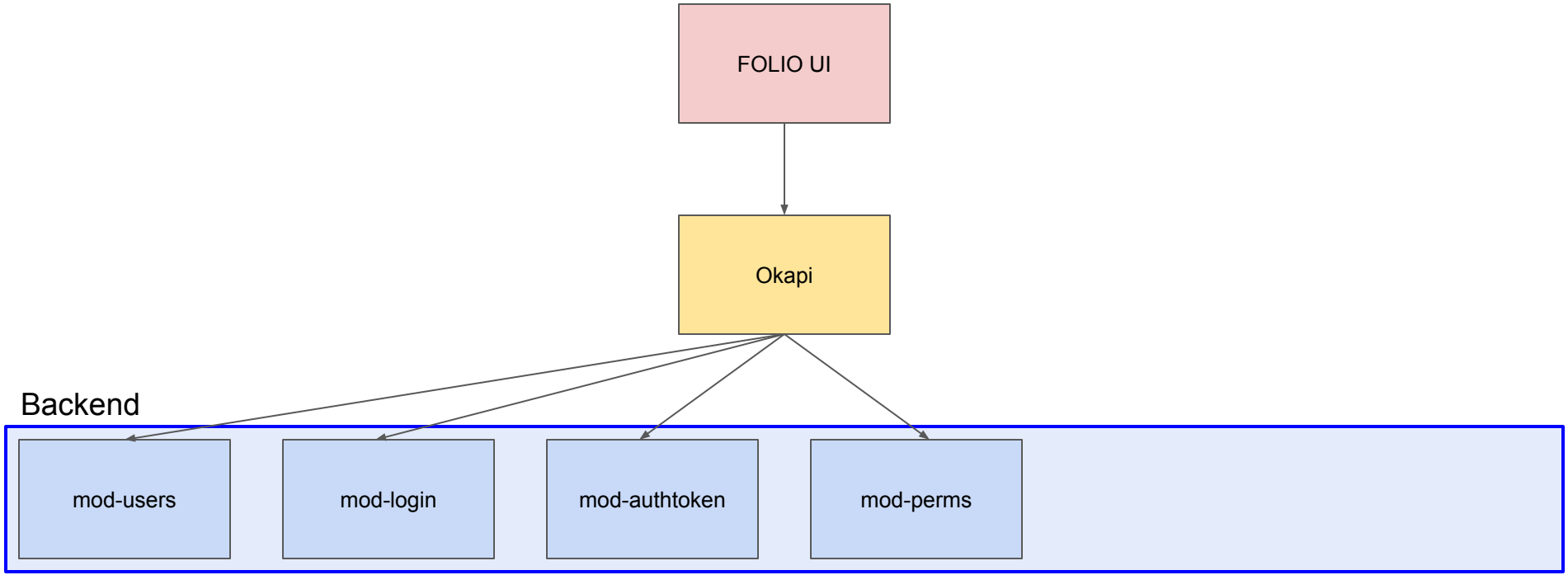
2. Modularity in FOLIO



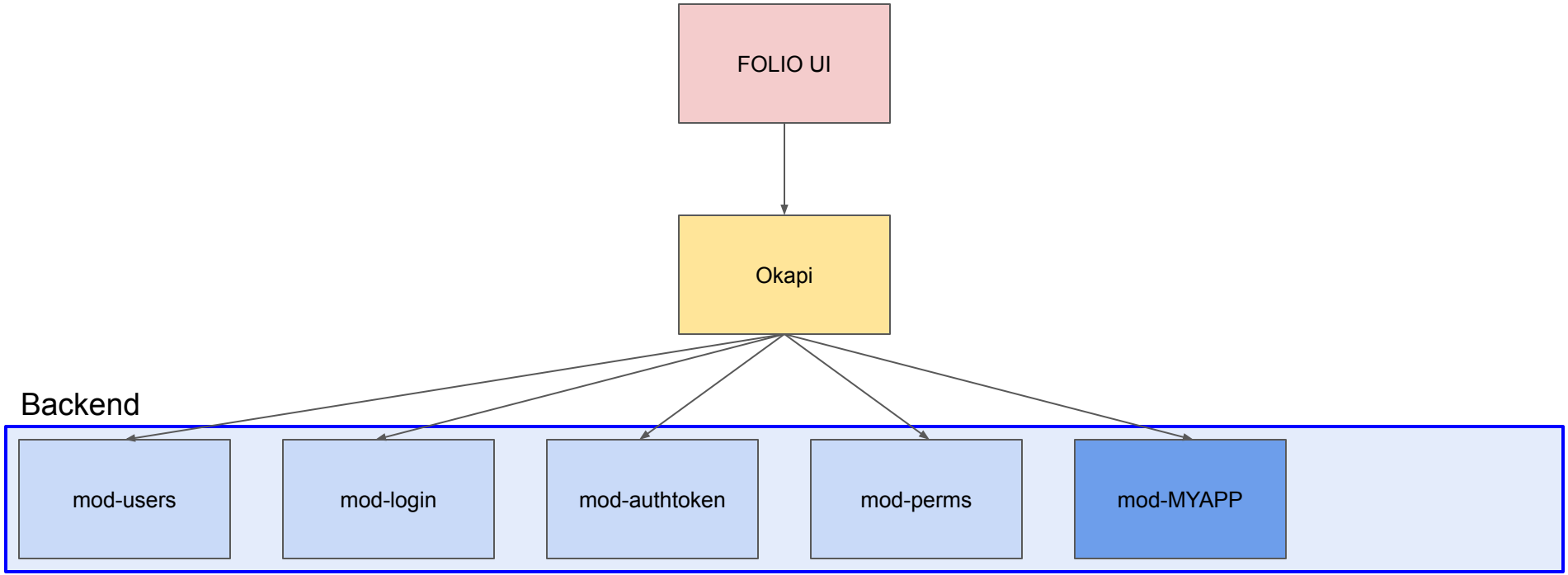
2. Modularity in FOLIO



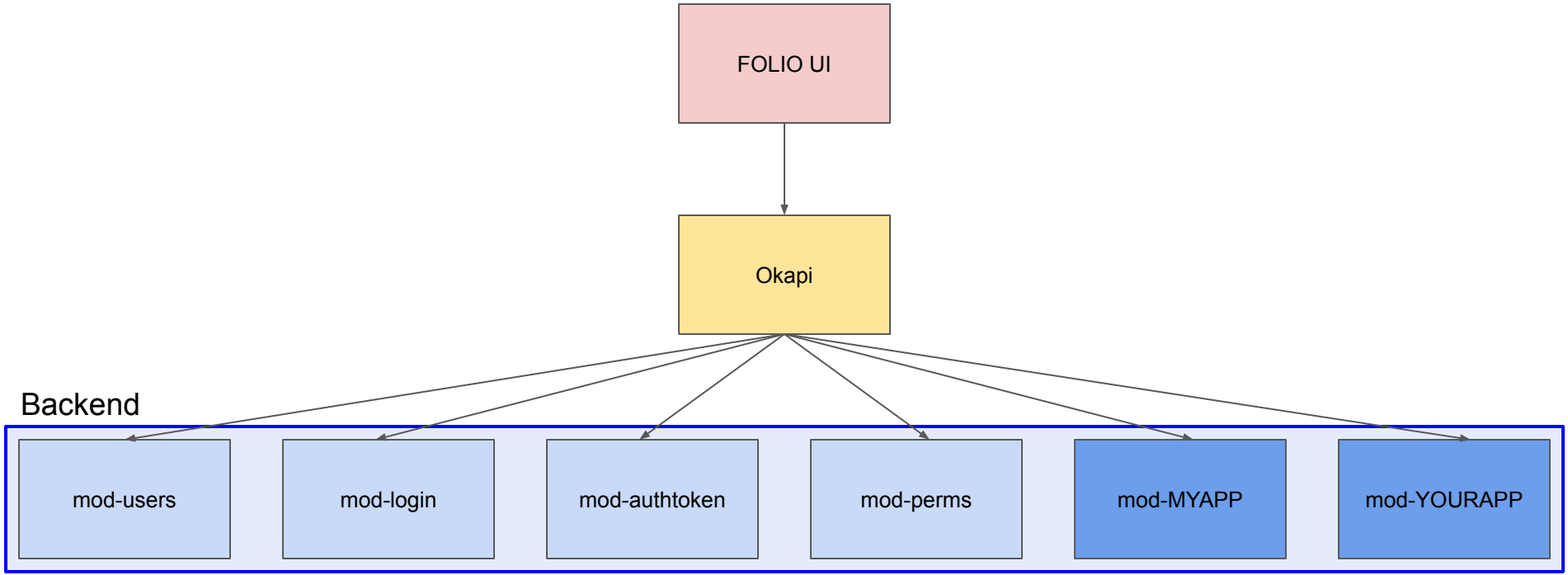
2. Modularity in FOLIO



2. Modularity in FOLIO



2. Modularity in FOLIO



2. Modularity in FOLIO

(The UI is also modular, but this talk is not concerned with that.)

3. Benefits of the modular architecture

This vision of modularity is crucial to FOLIO's appeal to the library community, because it lowers the bar to participation:

3. Benefits of the modular architecture

This vision of modularity is crucial to FOLIO's appeal to the library community, because it lowers the bar to participation:

- New functionality can be added as a module

3. Benefits of the modular architecture

This vision of modularity is crucial to FOLIO's appeal to the library community, because it lowers the bar to participation:

- New functionality can be added as a module
- Individual libraries may create modules that meet their needs

3. Benefits of the modular architecture

This vision of modularity is crucial to FOLIO's appeal to the library community, because it lowers the bar to participation:

- New functionality can be added as a module
- Individual libraries may create modules that meet their needs
  - Or hire developers to do so

3. Benefits of the modular architecture

This vision of modularity is crucial to FOLIO's appeal to the library community, because it lowers the bar to participation:

- New functionality can be added as a module
- Individual libraries may create modules that meet their needs
  - Or hire developers to do so
  - Or crowdfund development of widely needed modules

3. Benefits of the modular architecture

This vision of modularity is crucial to FOLIO's appeal to the library community, because it lowers the bar to participation:

- New functionality can be added as a module
- Individual libraries may create modules that meet their needs
  - Or hire developers to do so
  - Or crowdfund development of widely needed modules
- Different libraries can deploy different sets of modules, lowering operating costs.

3. Benefits of the modular architecture

None of this needs “permission” from a central authority.

3. Benefits of the modular architecture

None of this needs "permission" from a central authority.

There's one-one to say "no".

3. Benefits of the modular architecture

None of this needs "permission" from a central authority.

There's one-one to say "no".

(At least in principle.)

4. An ecosystem of modules

4. An ecosystem of modules

Example: a library might need a module for booking rooms.

4. An ecosystem of modules

Example: a library might need a module for booking rooms.

They could create that module themselves.

4. An ecosystem of modules

Example: a library might need a module for booking rooms.

They could create that module themselves.

And make it available for other FOLIO libraries to use.

4. An ecosystem of modules

Example: a library might need a module for booking rooms.

They could create that module themselves.

And make it available for other FOLIO libraries to use.

This is exactly what happened with Course Reserves:

- Needed by FLO, the Fenway Library Organization in Boston.
- Developed by Index Data with their funding
- Now used by almost all US Academic libraries

4. An ecosystem of modules

FOLIO was started in 2015, in Sint Maarten.

4. An ecosystem of modules

FOLIO was started in 2015, in Sint Maarten.

The first modules (mod-users and mod-inventory) came in 2016.

4. An ecosystem of modules

FOLIO was started in 2015, in Sint Maarten.

The first modules (mod-users and mod-inventory) came in 2016.

Eight years on, it has grown to 114 back-end modules
(as of the current snapshot build).

SIDEBAR

How are modules made available to the community?
How can libraries know to trust them?

SIDEBAR

How are modules made available to the community?
How can libraries know to trust them?

This is more of an organizational problem than technical one.

We can talk more about it in the Q&A if there is interest.

5. Some light technical details

A FOLIO module is a computer program that fulfils certain requirements.

5. Some light technical details

A FOLIO module is a computer program that fulfils certain requirements.

It can be written in any language (though most are in Java).

5. Some light technical details

A FOLIO module must fulfil certain requirements:

- It must respond to WSAPI requests.

5. Some light technical details

A FOLIO module must fulfil certain requirements:

- It must respond to WSAPI requests.
- WSAPIs are typically RESTful and use JSON payloads.

5. Some light technical details

A FOLIO module must fulfil certain requirements:

- It must respond to WSAPI requests.
- WSAPIs are typically RESTful and use JSON payloads.
- It must provide a machine-readable description of the requests it accepts and the responses it can give.

5. Some light technical details

A module's description must specify:

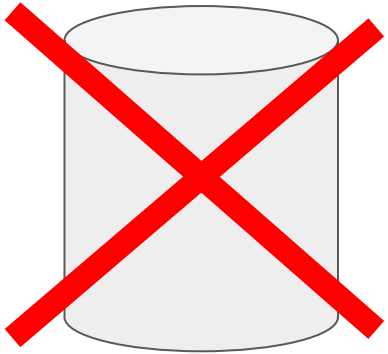
- What interfaces the module requires
- What interfaces it provides.

5. Some light technical details

A module's description must specify:

- What interfaces the module requires
- What interfaces it provides.

All communication between modules is via these interfaces: *there is no shared database.*



5. Some light technical details

A module's description must specify:

- What interfaces the module requires
- What interfaces it provides.

All dependencies are on *interfaces*.

Not on modules
(which are implementations).

5. Some light technical details

An example interface description (from mod-ldp):

```
{
  "id" : "ldp-query",
  "version" : "1.2",
  "handlers": [
    {
    "methods": [ "GET" ],
    "pathPattern" : "/ldp/db/log",
    "permissionsRequired": [ "ldp.read" ]
    },
    ... etc, ...
  ]
}
```

5. Some light technical details

FOLIO enforces a partitioning of concerns: integration between modules is by contract-defined API-based communication.

5. Some light technical details

FOLIO enforces a partitioning of concerns: integration between modules is by contract-defined API-based communication.

Consumers of an interface neither know nor care what module implements it. (And vice versa.)

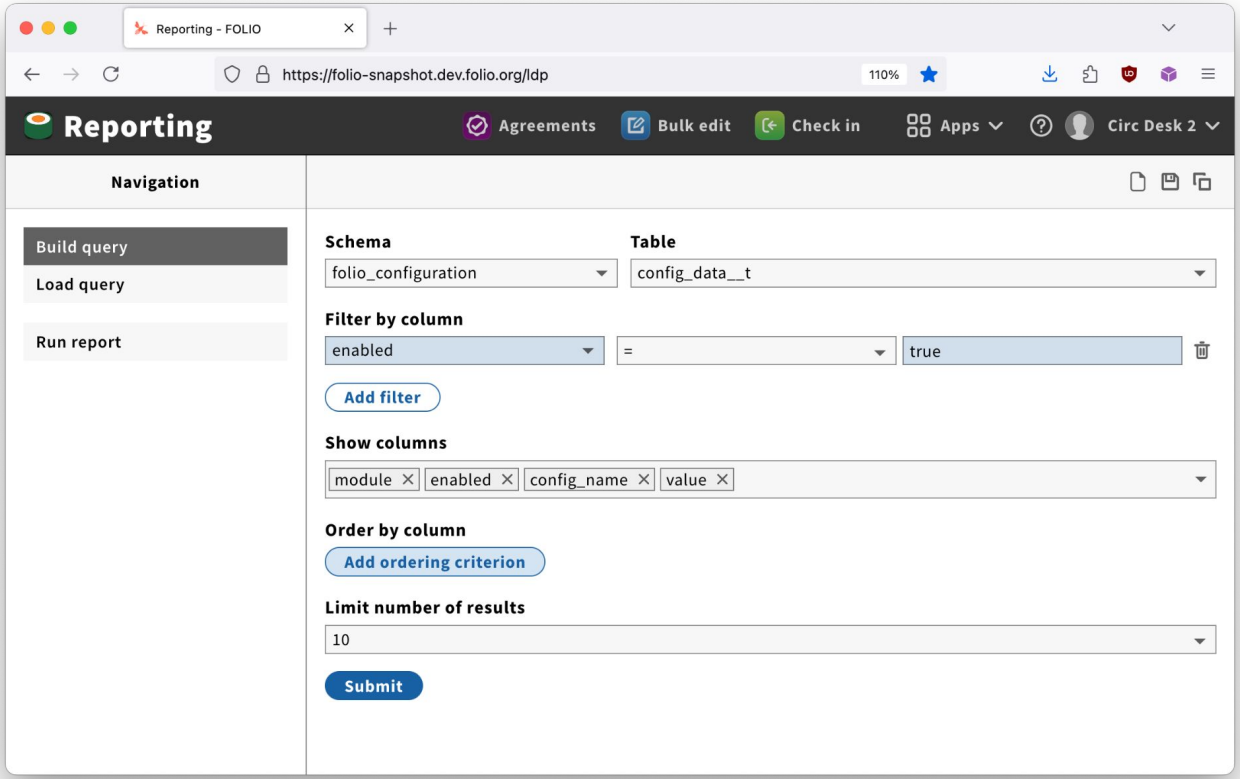
5. Some light technical details

FOLIO enforces a partitioning of concerns: integration between modules is by contract-defined API-based communication.

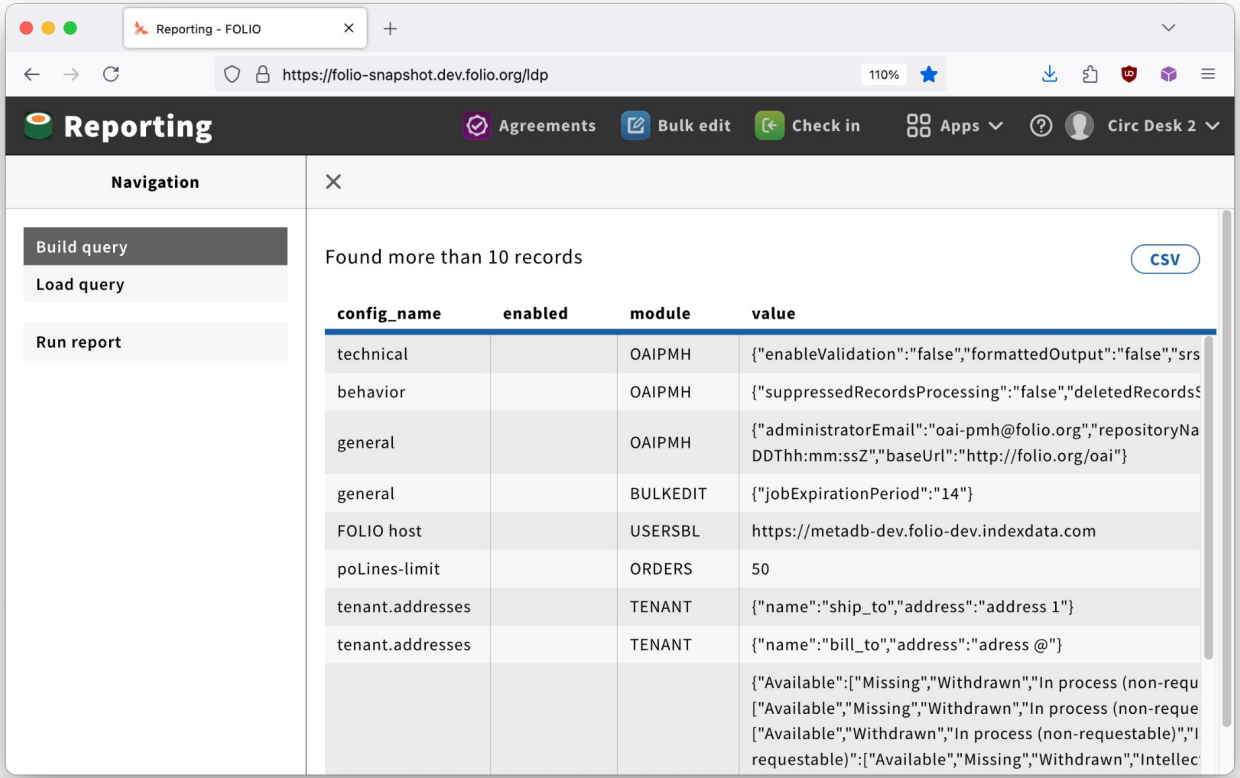
Consumers of an interface neither know nor care what module implements it. (And vice versa.)

Multiple modules can provide alternative implementations of the same interface.

6. A case-study: mod-reporting



6. A case-study: mod-reporting



6. A case-study: mod-reporting

FOLIO's Reporting app allows users to search in Metadb.

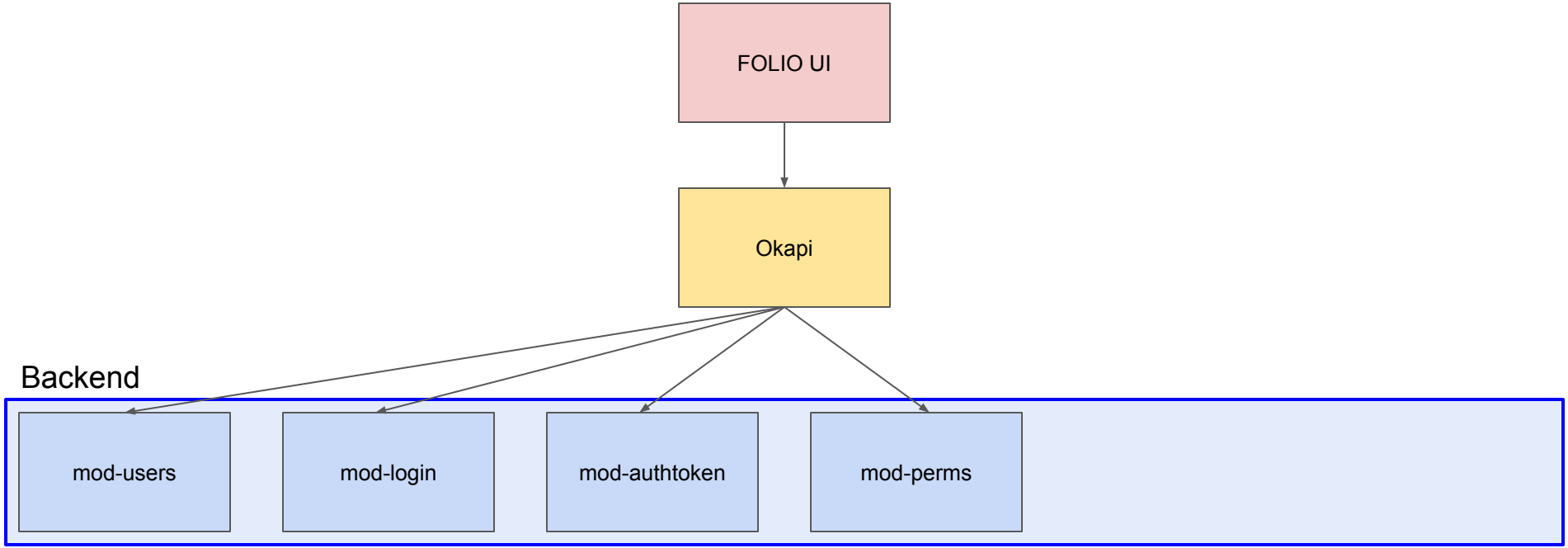
This is a relational database that contains normalized records harvested from FOLIO.

They are continually updated as they change, and history is maintained.

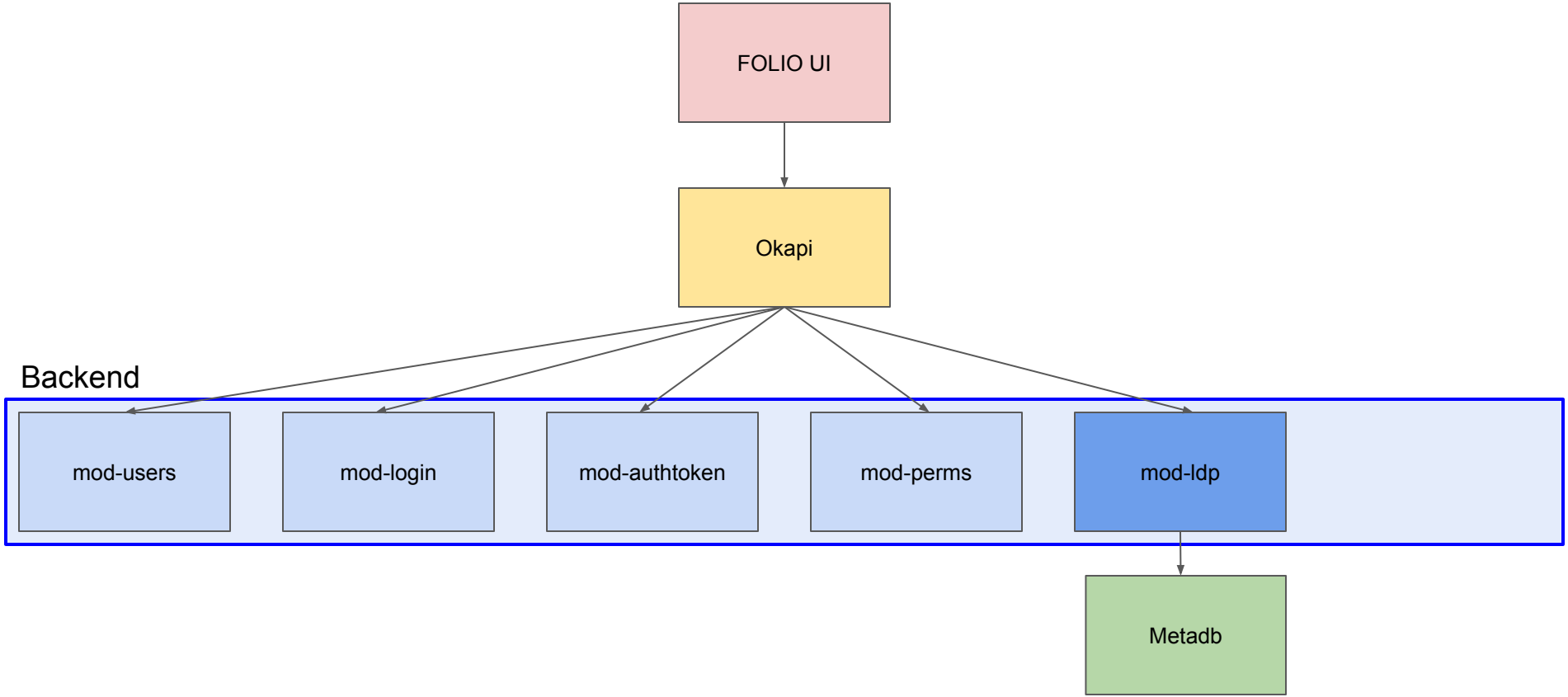
6. A case-study: mod-reporting

Apart from the UI component, the app is implemented by a back-end module (mod-ldp) which communicates on the UI's behalf with Metadb.

6. A case-study: mod-reporting



6. A case-study: mod-reporting



6. A case-study: mod-reporting

mod-ldp is … not great:

- old code
- written by a third party
- using tools alien to its maintainers
- tenant separation is questionable
- insecure in places
- badly in need of updates.

6. A case-study: mod-reporting

mod-ldp is … not great:

- old code
- written by a third party
- using tools alien to its maintainers
- tenant separation is questionable
- insecure in places
- badly in need of updates.

Rather than fix it, we decided to replace mod-ldp with a new module, mod-reporting, written in Go.

6. A case-study: mod-reporting

The interface description from mod-reporting:

6. A case-study: mod-reporting

The interface description from mod-reporting:

```
{
  "id" : "ldp-query",
  "version" : "1.2",
  "handlers": [
    {
    "methods": [ "GET" ],
    "pathPattern" : "/ldp/db/log",
    "permissionsRequired": [ "ldp.read" ]
    },
    ... etc, ...
  ]
}
```


POLICE PUBLIC CALL BOX
POLICE PUBLIC CALL BOX

5. Some light technical details

An example interface description (from mod-ldp):

```
{
  "id" : "ldp-query",
  "version" : "1.2",
  "handlers": [
    {
    "methods": [ "GET" ],
    "pathPattern" : "/ldp/db/log",
    "permissionsRequired": [ "ldp.read" ]
    },
    ... etc, ...
  ]
}
```

6. A case-study: mod-reporting

The interface description from mod-reporting:

```
{
  "id" : "ldp-query",
  "version" : "1.2",
  "handlers": [
    {
    "methods": [ "GET" ],
    "pathPattern" : "/ldp/db/log",
    "permissionsRequired": [ "ldp.read" ]
    },
    ... etc, ...
  ]
}
```

6. A case-study: mod-reporting

The new module provides the same interface as the old.

That means the new module is plug-compatible with the old.

6. A case-study: mod-reporting

The new module provides the same interface as the old.

That means the new module is plug-compatible with the old.

It has been deployed in place of its predecessor without problems.

6. A case-study: mod-reporting

The new module provides the same interface as the old.

That means the new module is plug-compatible with the old.

It has been deployed in place of its predecessor without problems.

The existing UI module has not been changed to accommodate it, and continues to work exactly as before.

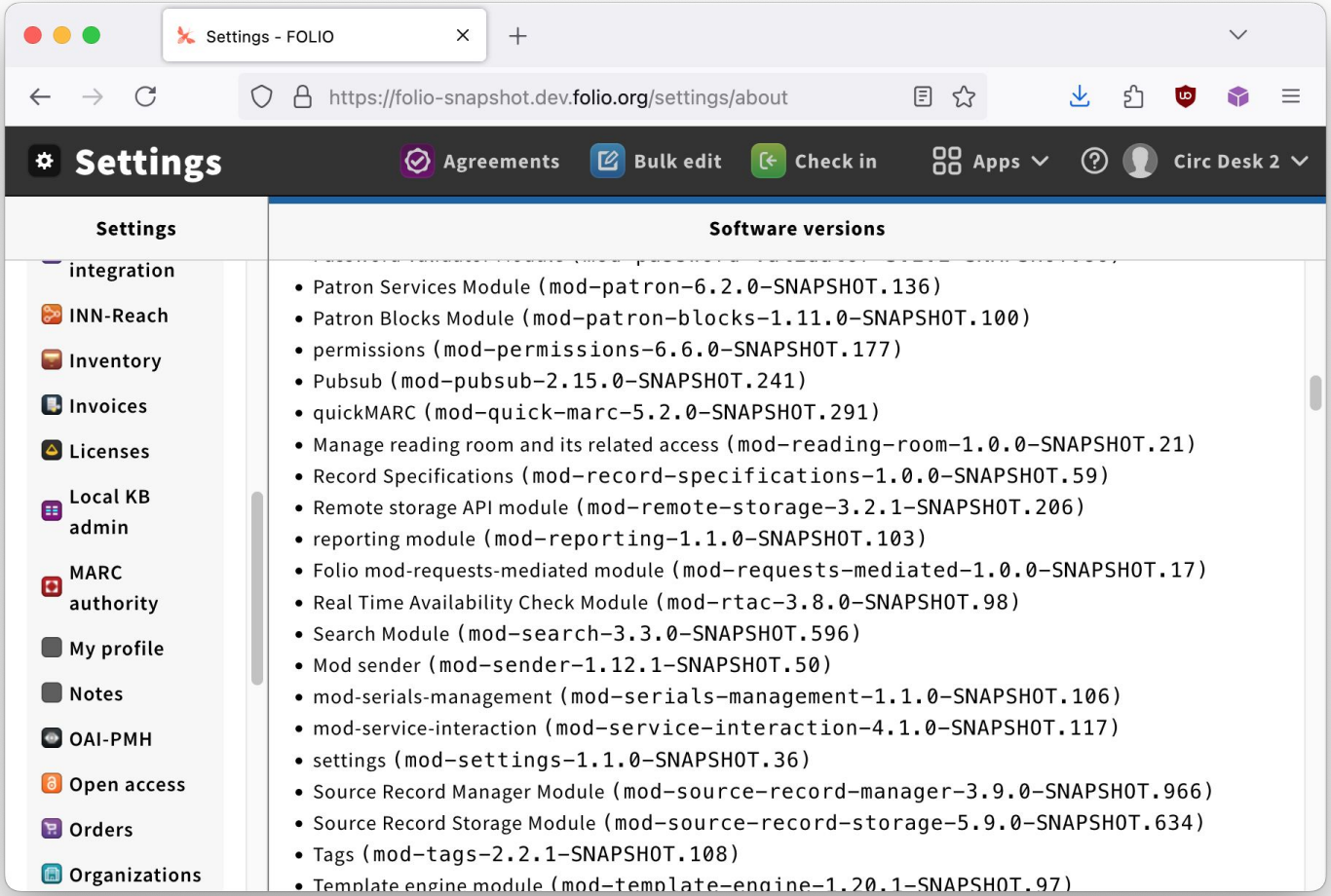
6. A case-study: mod-reporting

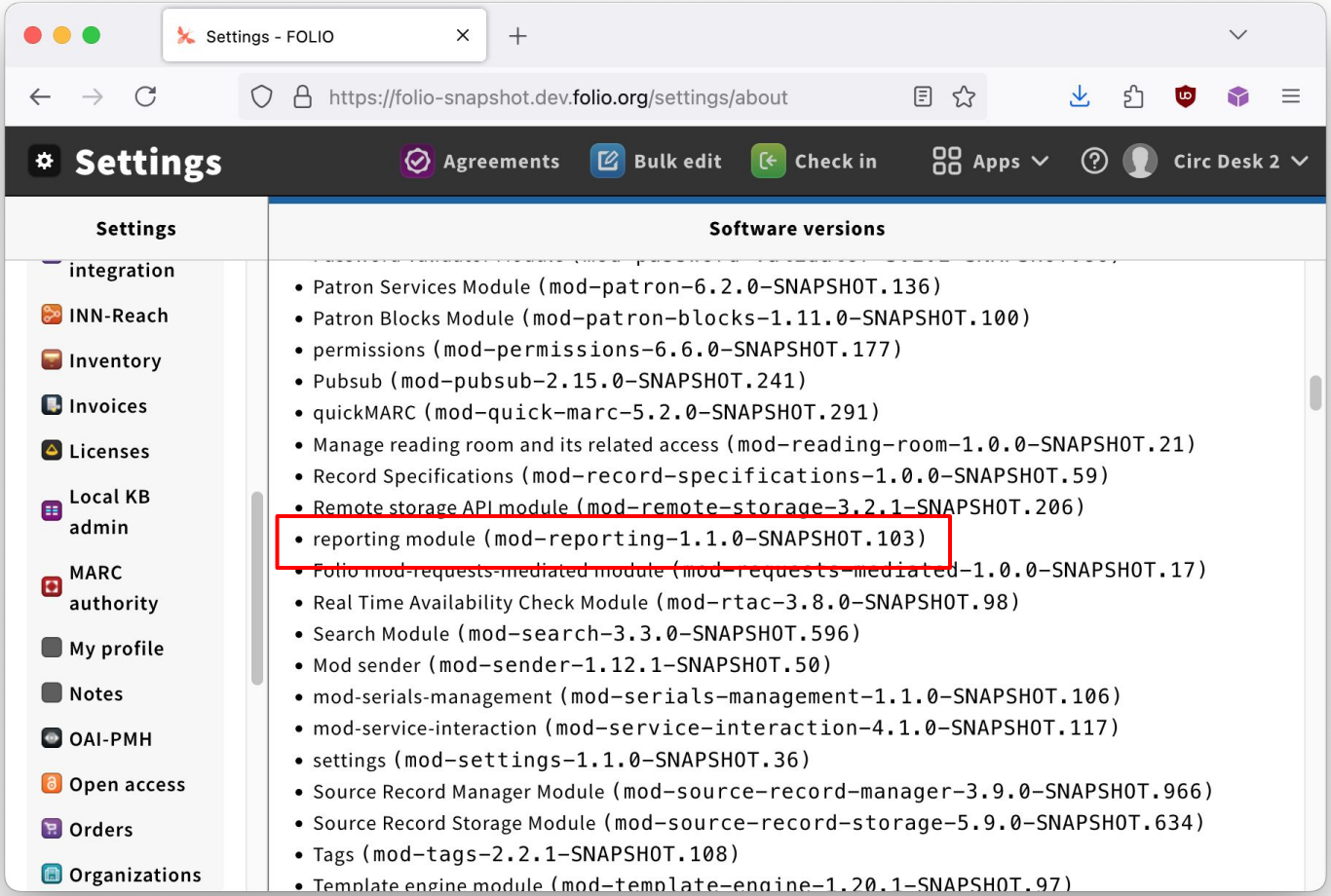
The new module provides the same interface as the old.

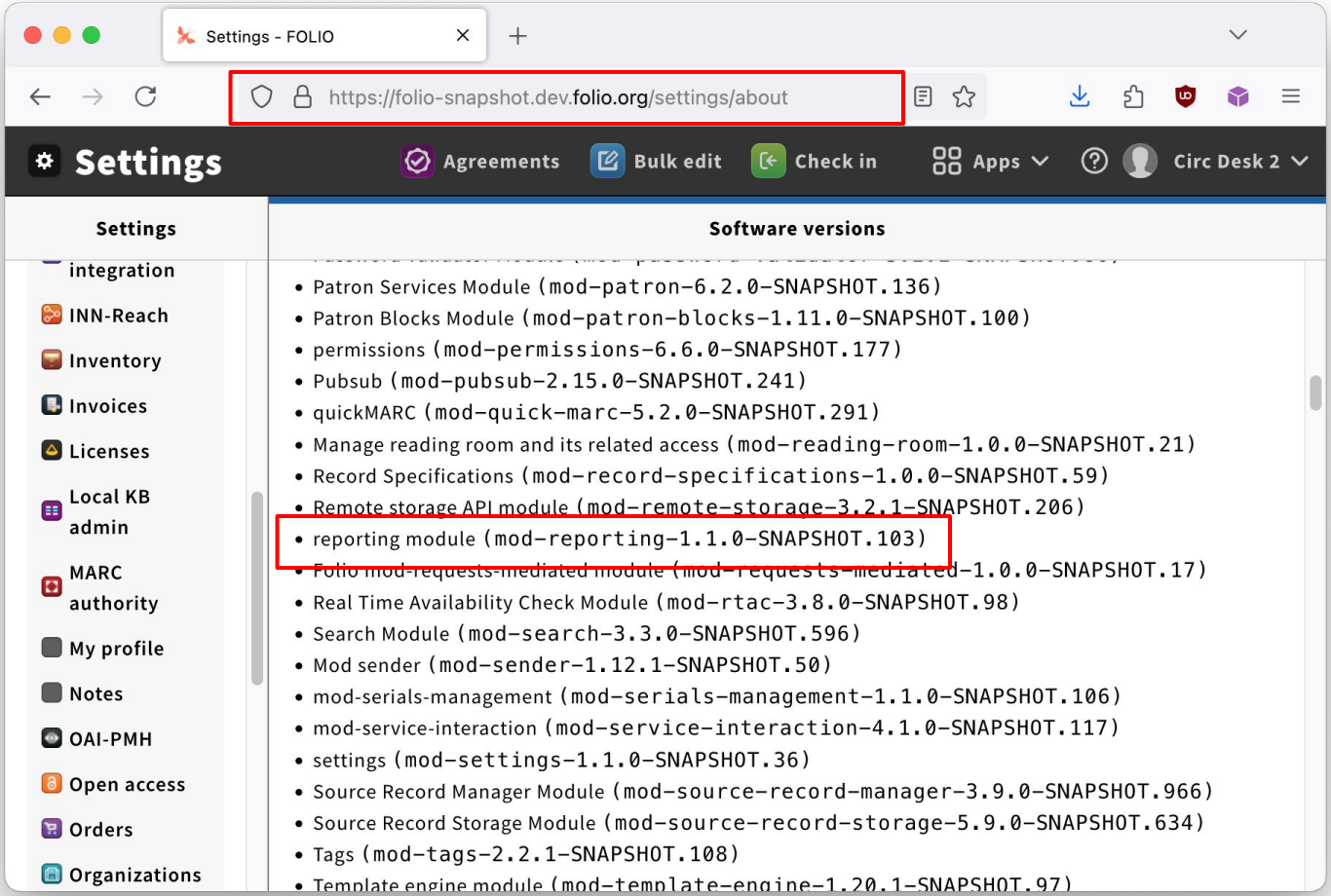
That means the new module is plug-compatible with the old.

It has been deployed in place of its predecessor without problems.

The existing UI module has not been changed to accommodate it, and continues to work exactly as before.


Settings - FOLIO
https://folio-snapshot.dev.folio.org/settings/about
Settings
Agreements
Bulk edit
Check in
Apps
Circ Desk 2
Settings
Software versions
integration
INN-Reach
Inventory
Invoices
Licenses
Local KB admin
MARC authority
My profile
Notes
OAI-PMH
Open access
Orders
Organizations
Patron Services Module (mod-patron-6.2.0-SNAPSHOT.136)
Patron Blocks Module (mod-patron-blocks-1.11.0-SNAPSHOT.100)
permissions (mod-permissions-6.6.0-SNAPSHOT.177)
Pubsub (mod-pubsub-2.15.0-SNAPSHOT.241)
quickMARC (mod-quick-marc-5.2.0-SNAPSHOT.291)
Manage reading room and its related access (mod-reading-room-1.0.0-SNAPSHOT.21)
Record Specifications (mod-record-specifications-1.0.0-SNAPSHOT.59)
Remote storage API module (mod-remote-storage-3.2.1-SNAPSHOT.206)
reporting module (mod-reporting-1.1.0-SNAPSHOT.103)
Folio mod-requests-mediated module (mod-requests-mediated-1.0.0-SNAPSHOT.17)
Real Time Availability Check Module (mod-rtac-3.8.0-SNAPSHOT.98)
Search Module (mod-search-3.3.0-SNAPSHOT.596)
Mod sender (mod-sender-1.12.1-SNAPSHOT.50)
mod-serials-management (mod-serials-management-1.1.0-SNAPSHOT.106)
mod-service-interaction (mod-service-interaction-4.1.0-SNAPSHOT.117)
settings (mod-settings-1.1.0-SNAPSHOT.36)
Source Record Manager Module (mod-source-record-manager-3.9.0-SNAPSHOT.966)
Source Record Storage Module (mod-source-record-storage-5.9.0-SNAPSHOT.634)
Tags (mod-tags-2.2.1-SNAPSHOT.108)


Settings - FOLIO
https://folio-snapshot.dev.folio.org/settings/about
Settings
Agreements
Bulk edit
Check in
Apps
Circ Desk 2
Settings
Software versions
integration
INN-Reach
Inventory
Invoices
Licenses
Local KB admin
MARC authority
My profile
Notes
OAI-PMH
Open access
Orders
Organizations
• Patron Services Module (mod-patron-6.2.0-SNAPSHOT.136)
• Patron Blocks Module (mod-patron-blocks-1.11.0-SNAPSHOT.100)
• permissions (mod-permissions-6.6.0-SNAPSHOT.177)
• Pubsub (mod-pubsub-2.15.0-SNAPSHOT.241)
• quickMARC (mod-quick-marc-5.2.0-SNAPSHOT.291)
• Manage reading room and its related access (mod-reading-room-1.0.0-SNAPSHOT.21)
• Record Specifications (mod-record-specifications-1.0.0-SNAPSHOT.59)
• Remote storage API module (mod-remote-storage-3.2.1-SNAPSHOT.206)
• reporting module (mod-reporting-1.1.0-SNAPSHOT.103)
• Real Time Availability Check Module (mod-rtac-3.8.0-SNAPSHOT.98)
• Search Module (mod-search-3.3.0-SNAPSHOT.596)
• Mod sender (mod-sender-1.12.1-SNAPSHOT.50)
• mod-serials-management (mod-serials-management-1.1.0-SNAPSHOT.106)
• mod-service-interaction (mod-service-interaction-4.1.0-SNAPSHOT.117)
• settings (mod-settings-1.1.0-SNAPSHOT.36)
• Source Record Manager Module (mod-source-record-manager-3.9.0-SNAPSHOT.966)
• Source Record Storage Module (mod-source-record-storage-5.9.0-SNAPSHOT.634)
• Tags (mod-tags-2.2.1-SNAPSHOT.108)


Settings - FOLIO
https://folio-snapshot.dev.folio.org/settings/about
Settings
Agreements
Bulk edit
Check in
Apps
Circ Desk 2
Settings
Software versions
integration
INN-Reach
Inventory
Invoices
Licenses
Local KB admin
MARC authority
My profile
Notes
OAI-PMH
Open access
Orders
Organizations
• Patron Services Module (mod-patron-6.2.0-SNAPSHOT.136)
• Patron Blocks Module (mod-patron-blocks-1.11.0-SNAPSHOT.100)
• permissions (mod-permissions-6.6.0-SNAPSHOT.177)
• Pubsub (mod-pubsub-2.15.0-SNAPSHOT.241)
• quickMARC (mod-quick-marc-5.2.0-SNAPSHOT.291)
• Manage reading room and its related access (mod-reading-room-1.0.0-SNAPSHOT.21)
• Record Specifications (mod-record-specifications-1.0.0-SNAPSHOT.59)
• Remote storage API module (mod-remote-storage-3.2.1-SNAPSHOT.206)
• reporting module (mod-reporting-1.1.0-SNAPSHOT.103)
• Real Time Availability Check Module (mod-rtac-3.8.0-SNAPSHOT.98)
• Search Module (mod-search-3.3.0-SNAPSHOT.596)
• Mod sender (mod-sender-1.12.1-SNAPSHOT.50)
• mod-serials-management (mod-serials-management-1.1.0-SNAPSHOT.106)
• mod-service-interaction (mod-service-interaction-4.1.0-SNAPSHOT.117)
• settings (mod-settings-1.1.0-SNAPSHOT.36)
• Source Record Manager Module (mod-source-record-manager-3.9.0-SNAPSHOT.966)
• Source Record Storage Module (mod-source-record-storage-5.9.0-SNAPSHOT.634)
• Tags (mod-tags-2.2.1-SNAPSHOT.108)

7. Implications

Our experience provides an example of how such code upgrades can be performed.

7. Implications

Our experience provides an example of how such code upgrades can be performed.

It provides a pathway for piecewise refreshment of parts of FOLIO as they age out.

7. Implications

A similar technical approach can also enable the substitution of different-*behaving* implementations of an API.

7. Implications

A similar technical approach can also enable the substitution of different-*behaving* implementations of an API.

For example:

- A plug-compatible mod-users alternative that uses LDAP as the source of truth

7. Implications

A similar technical approach can also enable the substitution of different-*behaving* implementations of an API.

For example:

- A plug-compatible mod-users alternative that uses LDAP as the source of truth
- A plug-compatible mod-fqm-manager replacement that runs queries against MetaDB

7. Implications

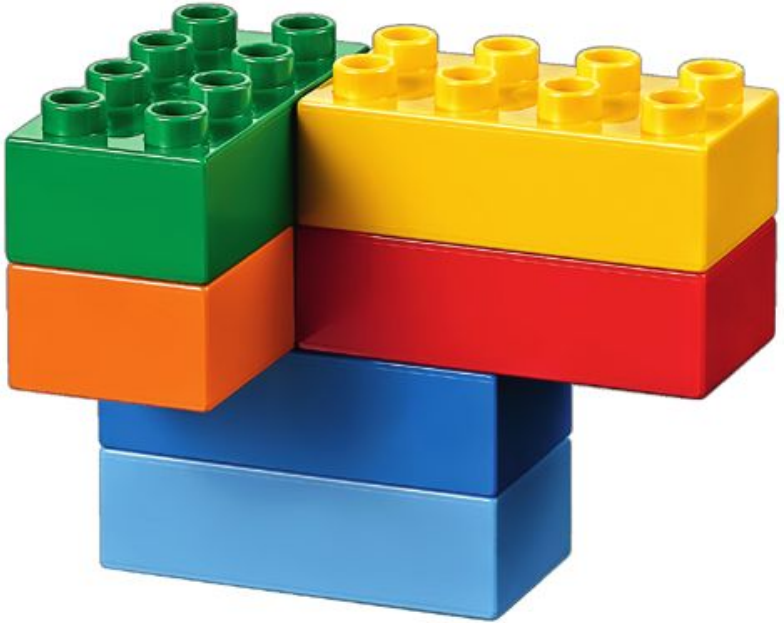
Crucially, *anyone can do this*.

7. Implications

Crucially, *anyone can do this*.

Modularity is empowering!

A case-study in FOLIO modularity: replacing mod-ldp with mod-reporting



Mike Taylor
Software Guy
Index Data
mike@indexdata.com