



Architectural Beeprints

WOLFcon 2022

Hamburg, Germany

Tod Olson & Vince Bareau





Overview

A discussion around new strategic capabilities for Folio. Some in the near future and some further away.

In this forum we will foster a conversation around some of these. A small number of strategic architectural enablers will be introduced to seed the conversation.

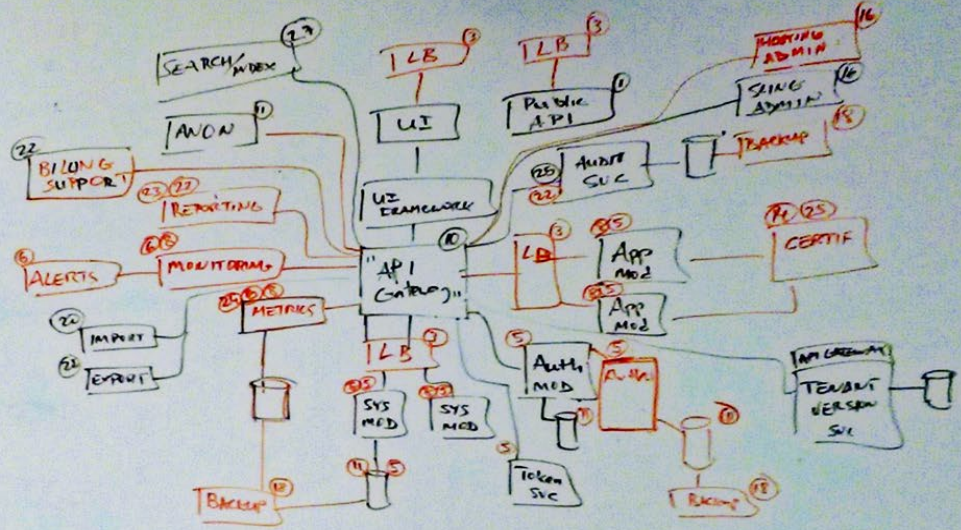
The goal is to solicit feedback around these topics and gather new or alternative ideas.



Where it Started



- ✓ 28 COLLABORATION
- ✓ 27 SEARCH
- ✓ 26 MULTITENANCY
- ✓ 25 AUDITING
- ✓ 24 - Customer-facing APIs
- ✓ 23 - Deployments Containers
- ✓ 22 - Load balancing
- ✓ 21 - Auto-scaling
- ✓ 20 - Security
- ✓ 19 - Monitoring /Alerting/Logging
- ✓ 18 - Storage
- ✓ 17 - Traceability
- ✓ 16 - Standalone/dedicated hosting
- ✓ 15 - Upgrades/Versioning
- ✓ 14 - Privacy
- ✓ 13 - Customer support
- ✓ 12 - Misc. tooling
- ✓ 11 - App validation/certification
- ✓ 10 - Sandboxing/isolation
- ✓ 9 - Administration
- ✓ 8 - Trials
- ✓ 7 - Backup/recovery
- ✓ 6 - Provisioning resources
- ✓ 5 - Migration
- ✓ 4 - Redundancy
- ✓ 3 - Billing support
- ✓ 2 - Customer-facing reporting



First draft architectural blueprint for Folio: Feb 2016 .



Where to Next?

There are certain known areas of expansion for Folio now on the near horizon.

- Some **market capabilities** have been deferred so far:
Consortial Support; App Marketplace; ...
- The growth of Folio which has introduced **challenges of scale**: data integrity; dependencies management; ...
- The **adoption of Folio** has placed demands on: release; distribution; deployments; upgrade; maintenance; ...



Some Desired Capabilities



Capability	Description	Urgency
Consortial Support	Enable shared management of resources	Soon
App Marketplace	Publish and distribute add-on FOLIO functionality	Later
Release Simplification	Reduce overhead of release process	Soon
Dependency Reduction	Simplify complex interdependencies and improve their operational management	Soon
Data Integrity	Ensure valid cross-app data references	Now
Hosting Efficiency	Reduce computing and staff loads	Soon
Ease of Development	Ease of onboarding, creating dev environments	Soon
Inventory Flexibility	Rigid areas include hierarchical data model, item statuses, metadata standards	Mixed



Enablers

To provide desired capabilities it will be necessary to create some architectural enablers. In this forum we will introduce four such enablers for discussion

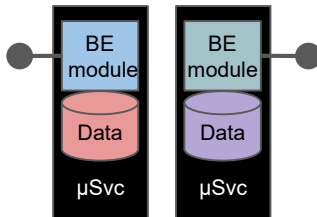
- Microservice Boundaries
- Release Packaging
- Platform Minimal
- Cross-app Data Sync

Enabler: Microservice Boundaries



Where we are

- Folio embraces microservices (sorta)
- Each backend module is considered a μ service.
- Each μ service is responsible for their own data layer.
- Only the backend module responsible for data storage is allowed to directly access it
- All other data access must be through the μ service API interfaces.
- Some older applications have implemented a separation between “business logic” and “storage” modules.

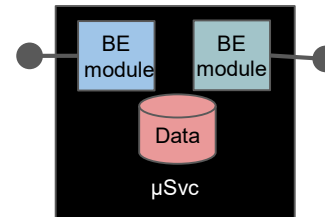


Ideas for Discussion

- Folio continues to embrace μ services (more so)
- The **Bounded Context** is considered a μ service
- A Bounded Context may consist of several backend modules
- All modules in the Bounded Context may directly access the shared data storage
- The “storage” vs. “business logic” pattern is eliminated

Benefits

- **Dependency Reduction**: reduces the number of modules
- **Ease of Development**
- Reduces latency (reduces http traffic)
- Contributes to **Data Integrity**
- Allows for extension modules

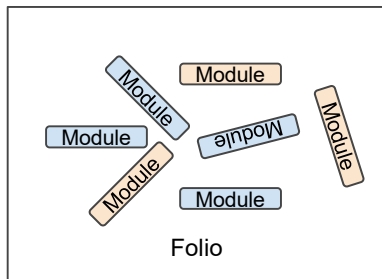


Enabler: Release Packaging



Where we are

- Folio is a monolith: a large bag of modules
- Flower releases are on a 4 month cadence
- Each Flower release is typically followed by multiple “Hot Fix” releases
- Large Testing Overhead

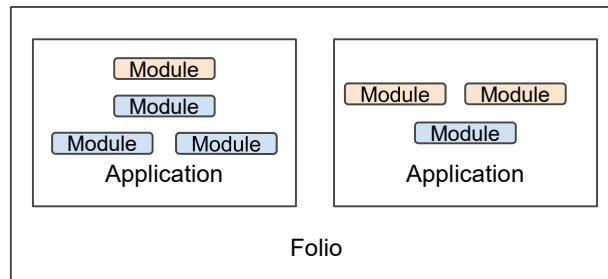


Ideas for Discussion

- Introduce the formalized concept of an **Application**
- Folio releases are comprised of a smaller number of Applications - which contain the modules
- Dependency management is moved to the Application boundary - fewer permutations

Benefits

- Tighter coupling between frontend and backend modules
- Provides for **Hosting Efficiency**
- Enabler for independent application lifecycles
- Enabler for **Folio Marketplace** or App Store





Enabler: Platform Minimal



Where we are

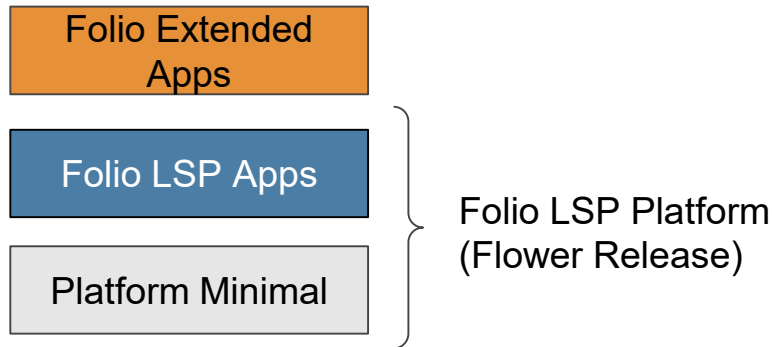
- The term “Platform” is not consistently used in Folio
- Concept: a minimal distribution of modules that allows a user to log into a UI, but provides no other functionality; a platform upon which to build functionality.
- Current work: identifying and packaging the set of modules to permit this bare-bones minimal deployment
- See [FOLIO-3253](#), [FOLIO-3548](#), etc.

Ideas for Discussion

- Defines an infrastructure layer, functionality goes on top
- This has been desired for several years
- To what degree can testing of this layer be automated?
- Can larger Platforms (e.g. Folio LSP) be composed of smaller Platforms (e.g. Platform Minimal)?

Benefits

- Provides a coherent base for functional platform distributions, i.e. FOLIO LSP distributions
- Provides a baseline for addressing bootstrapping and administrative pain points.
- Provides platform other projects can use, offers possible close integration
- **Release Simplification.** Reduces the size of Folio Flower releases when coupled with Release Packaging changes
- Provides for **Hosting Efficiency**



Enabler: Cross-App Data Sync



Where we are

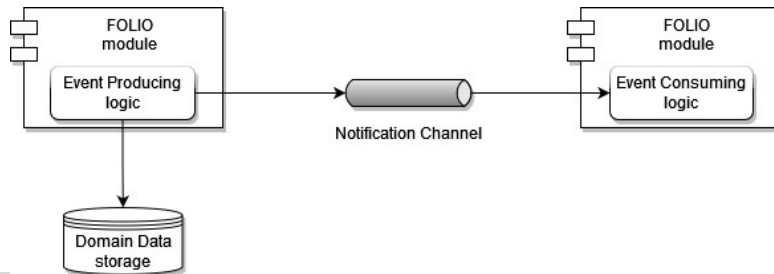
- Folio consists of many domains, each with many modules
- Modules provide a specific service in each domain
- Coordination between domains is implemented synchronously through APIs.
 - Designated “source of truth”
 - Requires deep knowledge of other modules and domains APIs and data model
 - Fragile
 - Example: Circulation State is tracked in Inventory Item; Acquisition Orders create stub Inventory Instance records.
- Existing active proposal: [FOLIO Cross-Application Data Sync Solution](#)
 - Eventual Consistency
 - Domain Event Pattern
 - Single Source of Truth

Ideas for Discussion

- Is Cross-App Data Sync suitable for all domains?
- What are acceptable latencies and how best to set the user’s expectations?
- Can Cross-App Data Sync be extended to Cross-Tenant Data sync?
- Can Cross-App Data Sync be extended to Cross-Folio Data Sync?

Benefits

- Enables **Consortial Support**
- Contributes to **Data Integrity**
- **Dependency Reduction:** loose coupling between apps as only each app concerns with modifying its own data based on event messages





Wrap up
and
Thank you



Appendix



Other Architectural Topics Not Discussed

- Cross Tenant Data
- Storage approach
 - JSONB usage
 - Centralized Storage Service
 - Localized Storage
- Elasticsearch vs Opensearch
- Roles and Permissions
- Data Privacy and Security
- Inter-Folio Integration