Theme: Shanghai library FOLIO project

Time: August 3, 2021 07:00pm (EST) / August 4, 2021 07:00am (GMT+8)

Attendees:
Vincent Bareau (Enterprise Architect, EBSCO)
Gang Zhou (Project manager, Shanghai library)
Lucy Liu (Product Owner, Folio China)

Notes:

1. **Reference Data**
   Vince:
   https://wiki.folio.org/display/TC/Improved+Handling+of+Reference+Data+During+Upgrades
   - A long standing committee/work group was formed from the TC at least a year ago to discuss the issue of reference data and has now completed its work. The issue came from the SysOps SIG regarding how to do the upgrades. There are some questions about how customers could implement customizations and how to make it easier to carry them across upgrades, and things like that. Vince was part of the working group.
   - For different modules in folio, there is a working set of data, for example, circulation, users and special inventory, which was defined by the project release but is possibly changed by libraries for customization. They change the labels, the values, and the structures. So the problem is if they do that, when it comes time to upgrade, then you may have a conflict in consistency between the changes that are wanted to be done by the upgrade and the changes that were already done by the customization and by the librarians.
   - The working group has recommended that modules restructure how they store their data. We do not want to have the customization values competing with the default values or the system values that are part of the release.
     - There will be a set of default data that are provided by the platform with the release. It represents a pristine Folio installation - no customizations. The ability to upgrade from the prior set of Default Data to the new set of Default Data can be completely automated.
     - And there will be operational data that is used in runtime by Folio. In a pristine Folio installation the Operational Data is merely an identical copy of the Default Data. Any and all customizations are applied to Operational Data. The default data set remains untouched.
   - The release becomes a two-part release now:
     - The first part is to upgrade the default data. This is going to be very smooth and automatic. It doesn't take into account any customizations during this process. We have the ability to generate a three-way comparison (Lucy copied the table from wiki). The Default Data of the system is upgraded to reflect the new state of Default Data as specified by the new release. As part of the process, any potentially conflicting customizations in the Operational Data are identified, and a report is

generated for use in the second step below. Furthermore, any identified conflicting customizations are preserved but yield in favor of the new Default Data changes. The goal is to produce a functional upgraded Folio installation. This step can be fully automated and should be provided as part of the distribution.

Old Default	Old Operational	New Default	New Operational
base	unchanged	unchanged	unchanged
base	unchanged	changed	apply
base	changed	unchanged	keep
base	changed	changed	**review**

    - The second part is to reconcile the operational data. Any conflicting customizations that were put aside in favor of Default Data can be addressed. This is the concern of the subject matter experts and they now have the functional upgraded Folio installation from the previous step with which to reconcile the conflicting customizations. This could involve manual evaluation and changes to achieve reconciliation.
- A team will be selected to apply this approach to a candidate module. The other modules will follow.

Gang Zhou: How will the change affect libraries that have already installed folio, such as Shanghai Library?
Vince: It's just a proof of concept, which will likely occur during the Kiwi and Lotus development cycles. As with any existing module, there is currently a manual exercise required for any library to understand the differences between the new release and what they have in their system. Once the new model is added to the platform and modules, further upgrades will be simpler because the code will provide that separation between operational data and default data. Right now the problem is there is only one data bucket and everybody is putting many things in the same bucket.
Gang Zhou: What is default data? What is operational data? How to define them?
Vince: Default data only matters when you perform an upgrade. When the system runs, only the operational data matters. In definition, anything that the release provides is the default data. It's not the value or the record. It's defined by the apps. It may be the set of status, or the set of material types, etc.

2. **Data Import**

Vince:

- The issue of data import is mostly based around MARC records. A lot of attention was given to data import in the past few months because it was causing system-level issues when it was being run with large datasets. When somebody would put in ten thousand, hundred thousand or million records, it was taking a

long time. It was also causing interference in other parts of the system when it happened. It would slow down circulation if you're doing a large import at the same time. And the system would run out of memory and things would start generating errors. A lot of effort was put in optimizing data import. And the focus was to make it stable and reliable so that you can worry about performance later. So the main goal was to make it stable and reliable; and the secondary goal is to make it fast. They made a lot of changes/upgrades in this area. It's not any additional functionality that is made available. They basically made changes to the existing system.

- There are still some changes that are happening. For example, one issue that could happen is because data import is multi-threaded, if we try to import a whole bunch of records all at once, you might get duplicates because the different threads might attempt to create records in inventory for the same records that are being imported. There are some efforts to improve it. Right now, we don't yet have documentation, description of documentation or details of the authorizations. This work is ongoing.

Gang Zhou: When will the new feature be released?
Vince: They put a lot of the fixes into the latest release Juniper. Hotfix #3 has a lot of changes. The next release will contain a few more changes. The work is ongoing. Hopefully it will be completed with the next release.

3. **Kafka Direct**

Vince:

- We have Kafka now in the project on the platform. But until recently Kafka was hidden. Modules did not have direct access to use Kafka. They have access to a module called mod-pubsub, which was a wrapper that provided its functionality but using Kafka under the covers. Mod-pubsub was originally created to help the data import process. Originally data import was really meant to be automation for a small number of records. But then it was decided that data import would also be responsible for doing the initial data load - the first time data migration to folio, and potentially for large numbers of data imports. Mod-pubsub was insufficient because of that.
- So they did a redesign of data import to use Kafka directly about a year ago. Now they bypass Okapi. In the meantime, some other components also want to make direct use of Kafka. So the folio platform has been formalizing how other modules can make use of Kafka. Because direct use of Kafka has the characteristics that it bypass Okapi, it is bypassing the entire security model, which means if a module is going to use Kafka, it can go into Kafka, it can read topics from other tenants and from other modules. The security model restricts everything to one tenant when Okapi is involved. In order for Kafka direct available to the community, the community needs to figure out a security model.
- The security model is basically the following:
    - First, we need to do data encryption on transport into Kafka using the mTLS support that Kafka has built into it. It’s a SSL-based protocol and it relies on certificates. We are going to use the certificates to do the transport.

    - Then, we are going to isolate each tenant's data into topics instead of sharing one topic everywhere.
    - Once we have everything in place, we are going to use Kafka access control list ACLs to restrict access to the topics. So we restrict access to a topic to the specific modules and only to the tenant making the request. Basically we are going to introduce the isolation of tenant data within Kafka using these tools.
    - Eventually, we are going to connect to another project - Secrets Storage. We are going to store Kafka credentials, including the certificates we need, into that secret storage.
    - We are going to implement it in three phases:
        - First, stamp up the mTLS transport encryption and authentication as well as the ACLs access control list. But we are not going to separate the tenant data. This has been done for some pieces that rely on Kafka, for example, data import, ElasticSearch, pubsub, remote storage. First phase is done.
        - Second, multi-tenant work. We are going to separate the topics per tenant and add credentials per tenant so that we can do the isolation. Right now, the security authentication and the ACLs are in place, there is not a separation of topics for each tenant per module.
        - Third, connect these up with the centralized secrets storage.

Gang Zhou: Is it possible that Kafka becomes the data bus across apps and across tenants in folio in the future?
Vince: It's possible.
Gang Zhou: Any documents around the Kafka model? Any rules or designs?
Vince: It's under work. We have drafts of the Kafka security model
https://wiki.folio.org/display/~mage.air/Kafka+Security
https://wiki.folio.org/display/~mage.air/Kafka+usage
Data import is an exemplar of how we should use it and has some relevant documentation. ElasticSearch has already used Kafka as a bus. The documentation of ES is not complete right now.
Gang Zhou: Kafka might be used in ElasticSearch for communications between apps and between folio and outside apps. Any discussions in Kafka regarding this scenario, beyond data import?
Vince:

- Yes. It's already built into ElasticSearch on the platform. There's inventory storage that has a provider client for Kafka. Whatever changes are made there, they are posted to Kafka. And those messages are consumed by ES to update the indexes on inventory items.
- Another big effort is remote storage, which is integration with external systems for storing physical books. That's also using Kafka Direct as well. We are not allowing direct access to Kafka to external systems. It's closed to external connections. We have not discussed extending it yet. That's for later. That will post a security challenge that we haven't addressed yet.

4. **Secrets Storage**
   https://wiki.folio.org/display/DD/FOLIO+secrets+management
   Vince:
   - We don't have centralized secrets storage right now. But individual modules have created secret storage on their own and implemented it. They store secrets in different ways and in different places.
   - The idea is to provide a centralized secrets storage for Folio that would be available to any application/module to store the secrets and to be configurable so that you can configure it to use any of the chosen secret storage system for a particular deployment.
   - The proposed solution is to provide two Java libraries. One will work with Vert.x modules and one with Spring Way modules. they will be based on the code already implemented in Edge API Utils, which is a common library for creating edge APIs. That's the starting point. Modules will include this library, which will provide abstract but unified interfaces to do secret storage. Then there will be configuration to decide which actual secret storage being used. So we just need to worry about the abstract interface and no need to consider each individual secret storage. This doesn't have to be hosted in cloud storage. This will support local tools, for example Vault.
   - This is currently in the design stage. It's possible that some remote storage modules will apply this design first. But it's not decided yet. We are soliciting feedback.

   Gang Zhou: Any application to mod-search or OAI-PMH in the secrets management?
   Vince: That's the goal. All these modules already do secret management. Once we agree on how it should work, we can move the individual secret storage to a centralized one.
   Gang Zhou: When will this be released?
   Vince: It will be two releases. We need to implement it with one module and then decide how to implement it with other modules. My guess is probably not until after the Lotus release.

5. **System Users**
   Vince:
   - There's a problem in Folio that we only have one kind of user and all the users are about the same. The way Okapi works is you must have a user in each request. The issue is we have certain things, for example, edge APIs for availability check, in which case it doesn't matter who the user is. The way we are doing the availability check is to create an "institutional user" which will show up in the User app. The problem is that librarians might accidentally delete that user and requests can no longer be made in this circumstance.
   - It's still in the discussion/argument phase. Mikhail is leading the discussion

6. **Updates from Shanghai**
   Gang Zhou:
   - The VuFind went live. Patrons can search the library collections at SHL. They can log in to my libraries, and check the loans and create requests under VuFind.

VuFind is connected with Folio. It’s operational. This is for all print collections, more than 40 million items.

- We have different projects in folio, for example, the open stack project in circulation, checkout on mobile phone, net borrower, etc. All these will use the circulation modules.