2021-1-5 Meeting Notes

Date
05 Jan 2021

Attendees
Paula Sullenger
Holly Mistlebauer
Jakub Skoczen
Tod Olson
Marie Widigson
Philip Robinson
Mark Arnold
Anya Arnold
Debra Howell
Jean M. Pajerek
Kyle
Patrick Roth
Natalie Sommerville
Natascha Owens
Molly Driscoll
Martina Tumulla
Elizabeth German
Karen Newbery
@jessica Janecki
@Andrew Clark
@Mary?
Dennis Benndorf
Ian Walls
Patty Wanninger
Christie Thomas
Kelly Drake
Jenn Colt
Charlotte Whitt
Jackie Gottlieb
Bob Scheier
Ann-Marie Breaux
Jacquie Samples
Thomas C. Wilson
Peter Murray
Jesse Koennecke
## Goals

### Discussion items

<table>
<thead>
<tr>
<th>Time</th>
<th>Item</th>
<th>Who</th>
<th>Notes</th>
</tr>
</thead>
</table>
| Optimi     |      | Jakub Skoczen & Holly Mistlebauer        | Not be able to handle update conflicts has been an issue for over a year and a half - Issue# 2027  
Was discussed as a Tech Debt item at last WOLFCon and Optimistic Locking was selected as best strategy - UXPROD-1752 - This ticket is mostly about the backend development  
This feature has been ranked as R1 by most institutions  
There is no silver bullet for solving this problem - all have pros & cons  
- OL is suitable when risk of collisions is low and update duration is short. It is “optimistic” that updates won’t often collide  
- OL allows for detection and prevention of update conflicts, it does not provide functionality for combining or merging conflicting updates. Such functionality can be built “on top” of OL, e.g. in the UI  
- Alma uses pessimistic locking which locks records for a longer time, JIRA uses yet another strategy. In Sierra when batch loading and a record is busy you just get a report at end of load that lists these. In the UI case you get a message saying the record is busy and you cannot get in until it is released by the user.  
- In Q3 2020 Platform team has stated working on a design and implementation proposal for OL support in FOLIO  
- https://wiki.folio.org/display/DD/Optimistic+locking+proposal  
- **RMB-710** - SPIKE: design protocol and implementation for optimistic locking  
- **R1 2021: OL platform support implementation**  
  - In R1: **RMB-727** - Implement support for optimistic locking  
  - This allows to either detect (log a warning in the system logs) or Prevent (return an error to the client/UI) record update conflicts  
  - Needs to be explicitly enabled by a developer for a specific db table and module (e.g., “instances” table in mod-inventory)  
  - As of yet, no stories have been created to enable OL in specific modules/tables - unlikely to rollout in R1 because of the tight upcoming deadline  
  - GBV requested a way to do this without full development - there is a way to do this  
  - Platform team has started grooming a story that would provide a way for an implementer to enable detection (or prevention) for all tables and modules across the system through an environment variable: **RMB-777** - Optimistic locking environment variable option  
- **R2 rollout:**  
  - Preliminary plan assumed enabling conflict “detection” in R1 to verify the extent of the problem and “enabling” prevention in R2  
  - Prevention mechanism is not backward compatible - clients/UI need to be updated to handle error resulting from detected conflicts  
  - R2 stories for rolling out conflict prevention and corresponding changes need to be created  
- AMB: If it will be added to Inventory, then I think Data Import would have to make some changes to deal with those reports, when the conflict is caused by Data Import and a User making changes to the same record at once. There is nothing now in JIRA that reflects the work and size of work to be done.  
- OL in Inventory: **UIIN-1245** - Implement optimistic locking in Inventory  
- R2 would be clearer if we can get R1 to give report - Jakub, yes, want this for R1. Code freeze is mid-Feb  
- Is there a specific team of people who need to look for these issues? or an institution? - Cornell would be interested, Chicago will see if anyone has time for this. This will be important for consortia where maybe 5 libraries are sharing a record - rate of collisions will increase. OL may not be the best solution for frequent collisions. Enabling this for R1 to collect frequency data could help us make a better solution  
- AMB - trickiest part will be the flag in R1 - just now moving places to Honeysuckle and Iris work us just starting. Seems to be a little late to implement a Juniper solution. Jakub - yes, this is a big challenge. Hopefully that the institutional testing will help us.  
- Tod - maybe can look at our experience with current systems - where are the collisions happening. FOLIO won’t change that number.  
- AMB - anything patron-facing will be more high-cost, could affect someone trying to checkout, for example  
  - Ian will check with Koha to see how they handle and what the volume is  

## Future topics

| January 12 - Receiving demo |
Action items

☐