TA7919 - Investigate capability required in TAS

Link to task in Rally: https://rally1.rallydev.com/#/120708787896d/detail/task/219474211044?fdp=true

Preliminary (unreviewed) SDD from ePharmacy team for Recent Transactions Report (RTR)

TAS ePhmn SDD US3695 v1.0.docx

Projected User Experience

User logon

UI evaluate roles and stations coming from IAM

- Would make a call to business service to determine what divisions they have access to.

Resulting in the user seeing their listing of the divisions they have access to.

The user would put a check box next to the desired divisions, and enter or select a date range

Then run the report via a button on the page.

- A business service would take the divisions selected in the UI, the date ranges desired, etc. to retrieve the desired data for the specific report.

Meeting notes 5/25

IAM /SSO

According to ePharmacy SDD for the RTR the ePharmacy teams wants to present the user with CPACS to divisions based upon access.

- Want SSO in TAS so once they log in to TAS, they can view what they need.
- Tableau uses SSO integrated with AD.

Set up roles and stations integration with IAM

- IAM role based access. Attributes include set of roles
- Another attribute called station - provides the info about access to location.
- Configured access does not change often but mechanisms exist so it may be changed.
- Probably better to say It's *attribute based access*.

Read through the config service and display only the divisions they (the current report user) have access to.

> The list of divisions should come from a service. It will be reused by other parts of the system.
>
> TAS Core to create a business service retrieved from VistA via FHIR server.

Info from Mike McDougall email 5/29 0933

Timeline UML

Latest based on my research:

The tableau javascript API has a mechanism for applying filters before the report is rendered.

https://onlinehelp.tableau.com/current/api/js_api/en-us/JavaScriptAPI/js_api_concepts_filtering.htm

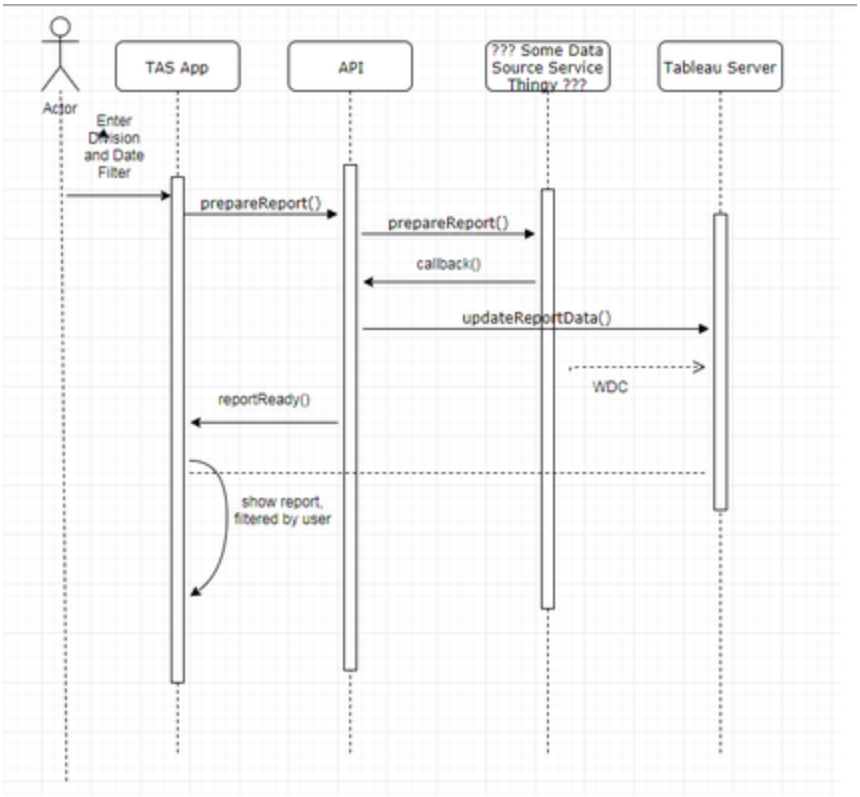
However I think this mechanism would only work with a Tableau native connector (Tableau Data Extract, Azure, otherDB etc) for “prefiltering” since Tableau knows how to pre filter data against it’s native connectors. *According to Joel from Tableau, using the native connector would be the preferred method to utilize in this use case. Continues that Pre-filtering does NOT have to be set to a Tableau data source exclusively.*

If we have to use a live Web Data Connector (WDC) I think we can use join filtering to establish a prefilter -http://tableau.github.io/webdataconnector/docs/wdc_join_filtering . It looks like we can chain two service calls with this mechanism without using the “Interactive Phase” of WDC. *According to Joel from Tableau, using the native connector would be the preferred method to utilize in this use case and is the best option over WDC.*

In the context of the uml below that would mean the prepareReport () function would actually be more like a prepare filters function to pass into

the Tableau JS api and the WDC.

We also need a mechanism to render the appropriate filter data after the report is rendered. With the JS api we can create filter UI with normal HTML/Angular dynamically based on parameters passed in. Then we can use the JS api to apply the filter based on the users selection. Here is an example of that - https://github.com/tableau/js-api-samples/blob/master/filter.html



UML diagram from Patrick Whalen

Real Time Reporting Data Sync Functionality

TAS API includes a queuing mechanism. So if VistA can put data out to TAS the data can be processed inbound to the TAS API using the queuing mechanism.

Question: Would this be delivered as a FHIR message?

- Sending data itself - could we do it through the FHIR server? yes - just a non FHIR notification then retrieve the data through the FHIR server.
- VistA would need to know how to call out to TAS.
- VistA would need a confirmation that the message was put on the queue.
- Would look exactly like the FSC prototype.

Sync data with Azure repo that is informed of updates to the data set which would need to be uploaded OR use local "in memory" table in Tableau for the repo.

- Might be easier to sync to Azure but sync to Tableau DE may be better performance at report runtime.
- Query for changes to pull data from changed instances.
- Or to PUSH data from the VistA instances.
- Or notifies us there was a change so we can go out an pull it.

Scenario:

> Someone makes a change in Chicago. Where does the notification go?
>
> Mike Cordi - To the TAS app. Through the nginx sever. Use a service or Queue to receive the notifications and process them.

Refreshed VistA data that needs to be integrated into report repository or a new report needs to be identifiable. The ePay team is implementing a new Date Time stamp field in ePay VistA files.

It may be necessary for the ePharmacy VistA files to be similarly modified to include a reliable trigger or a reliable change date in VistA.

Potential approaches to data sync

Live pull from VistA(s)

How would this cache to Azure in realtime?

Azure from data that is pulled when needed to support the realtime request.

130 rest calls to be executed every 5 min.

Data is accumulated and the user configures the request in order to pare down the accumulated data to view only the desired info.

It was pointed out that this sounds similar to just doing the request and getting the report back.

A hybrid approach may be possible.

- A background service spawns rest call.
- Goes out retrieves the data and stores it in a repo
- Then calling the repo to assemble the desired report.

FHIR Server

FHIR Server needs to be updated providing resources identified in the mapping sheet and also needs to be updated to read the new RPC's provided by dev team through VistA Link.

Need to ensure cert from VistA Link installed on FHIR server.

Refer to eRX Draft SDD and Initial Mapping Sheet in TA7918

VistA Link

VistA Link needs to provide access to RPC's.

RPC's

RPC's need to be developed in M.