COMMUNITY mojaloop MEETING

BizOps framework workstream

PI 16 Goal - Biz Ops Framework

Confidence Vote:4

Goal

✓ ✓

Key Epics Objectives

1. **As a compliance officer I access the audit trail of operator activities**
   Story 1: Deliver the read only logs that capture the information required to perform an audit.

2. As a hub operator I would like to deploy Biz Ops framework micro-front end/s into the financial portal so that I can take advantage of new functionality while keeping existing functionality

3. As a Hub Operator I would like to enforce a ~~maker-check~~ segregation of duties security policy so that I can enforce an RBAC policy ✓

4. As a infrastructure engineer, I would like to deploy BizOps framework along with the current OSS IaC ✓

Not Doing now but important next & Risk / Issues

1.

Extra resource/skills that would be valuable:

- **Hub Business process experts**
- **Infrastructure skills**
- **Backend Software Engineers**
- **UI/UX software engineers**

Risks you might need to track:

- **AVAILABILITY OF SME AND RESOURCES TO ACHIEVE THIS - ITS AMBITIOUS**

Success Defined How?

1. This RBAC solution in ~~production~~ **UAT environment** with WynePay ✓
2. Community Collaboration on documentation. ✗
3. Community Collaboration in building Micro-frontends with their supportive backend APIs to extend hub process supportive functionality. ✗
4.

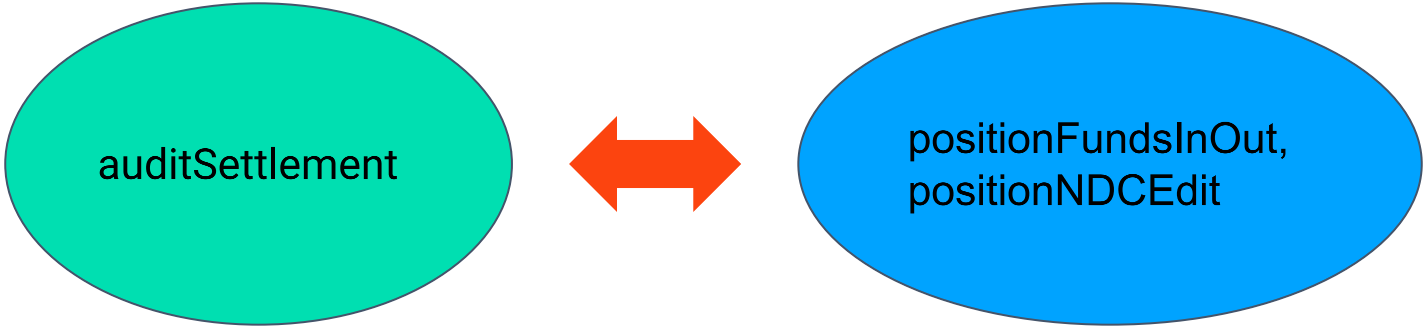
1. Demonstrate a ~~maker checker~~ **segregation of duties** RBAC ✓
2. Pick Positive and negative tests and demonstrate how this would be applied to a running system with integrity review checks. ✓
3. Document how a security policy change would be made and reviewed and pushed to live. ✓

Demo - Financial Portal V3

1. Trace a transfer
2. Reconciliation Processes
   a. Positions
   b. Deferred net settlement
   c. Reports
3. RBAC
   a. Assignment of role - participant DFSP access.
   b. IaC - integration

Segregation of duties enforcement

Mutually exclusive sets of permissions.



auditSettlement

positionFundsInOut,
positionNDCEdit

System ensures that no two roles can be assigned to a user that violates the exclusion set.

Team that made this all possible

Vijay Kumar

Kevin Leyow

Yevhen Kyriukha

Sridevi Miriyala

David Fry

Community collaboration

What is the easiest way to contribute to the BizOps initiative:

1. Add new reports

2. Enhance current micro-frontends

   E.g.

   a. Enhance Settlement UIs
   b. Enhance Settlement API controls
   c. Enhance participant lifecycle controls

3. Extend by adding new micro-frontends

   E.g.

   Platform configuration, Platform management, Liquidity management, Participant lifecycle management, Agreement (quoting) management, Account lookup and discovery management, Third-party-initiated payments management, Fees (interchange and billing) management, Reporting and analytics

Adding new Reports

```
apiVersion: mojaloop.io/v1
kind: MojaloopReport
metadata:
 name: {{ printf "%s-%s" .Release.Name "dfsp-settlement-detail" | trimPrefix "-"
}}
spec:
 endpoint:
   path: /dfspSettlementDetail
   params:
     - name: settlementId
       required: true
     - name: fspid
       required: true
 queries:
   - name: dfspInfo
     query: |
       SELECT participantId, name FROM participant WHERE name = :fspid AND name
!= 'Hub'
   - name: report
     query: |
       SELECT …
           WHERE
               tF.isValid
               AND s.settlementId = :settlementId
               AND (pPayee.name = :fspid OR pPayer.name = :fspid)
 template: |
   <!DOCTYPE html>
   <html lang="en">
   <head>
       <style>
           table {
               font-family: arial, sans-serif;
               border-collapse: collapse;
               width: 100%;
               display: block;
```

dfspSettlementDetail.yaml

Enhancing a UI - backend API

Repositories in mojaloop github.

https://github.com/mojaloop/reporting-hub-bop-settlements-ui

https://github.com/mojaloop/reporting-hub-bop-positions-ui

https://github.com/mojaloop/reporting-hub-bop-role-ui

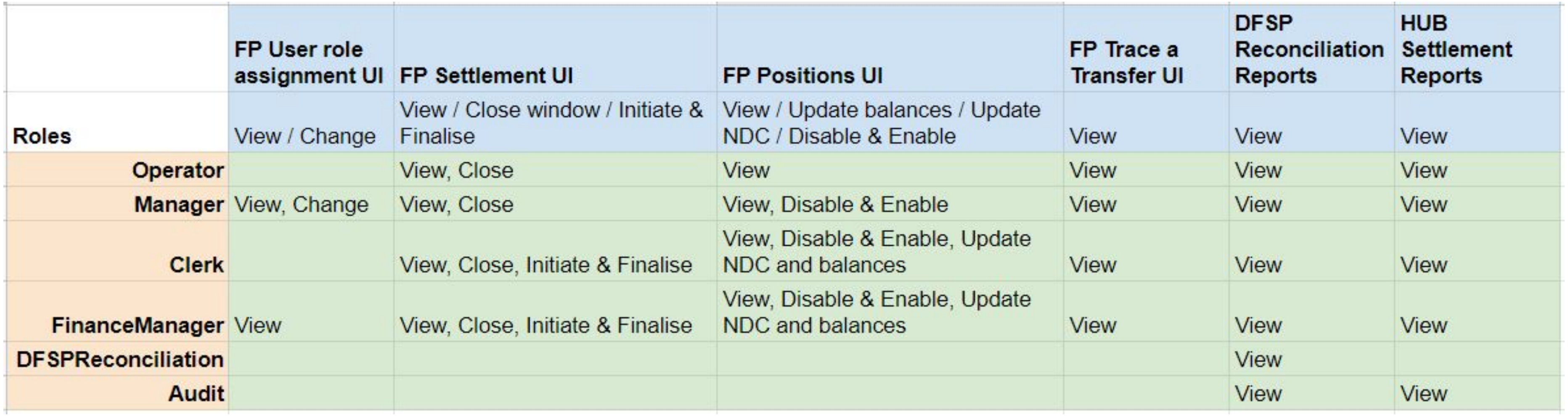
https://github.com/mojaloop/reporting-hub-bop-trx-ui

New Micro-frontend

1. Scheme customisation
2. Extend framework

https://github.com/mojaloop/microfrontend-boilerplate

IaC - standard roles



	FP User role assignment UI	FP Settlement UI	FP Positions UI	FP Trace a Transfer UI	DFSP Reconciliation Reports	HUB Settlement Reports
Roles	View / Change	View / Close window / Initiate & Finalise	View / Update balances / Update NDC / Disable & Enable	View	View	View
Operator		View, Close	View	View	View	View
Manager	View, Change	View, Close	View, Disable & Enable	View	View	View
Clerk		View, Close, Initiate & Finalise	View, Disable & Enable, Update NDC and balances	View	View	View
FinanceManager	View	View, Close, Initiate & Finalise	View, Disable & Enable, Update NDC and balances	View	View	View
DFSPReconciliation					View	
Audit					View	View

Financial Portal v3 Demo → IaC

COMMUNITY mojaloop MEETING

Demo 1: **RBAC Demo from within the IaC** (Infrastructure as code)

In this demo Vijay Kumar shows us the inner workings of the IaC design from the perspective of the IaC environment. This not only allows us to understand the parts of the RBAC and how security is controlled, but it demonstrates this functionality can be enabled and controlled in a production environment. Epic 2189 and epic 2305 are the work items that are being demonstrated.

1. Talk through IaC pipeline
2. Summary of RBAC design using online docs
3. Demo process for creating a new role
4. Demo process for adding permissions to roles Via GitLab and a PR
5. Demo process for creating a new user and assigning a role

Demo 2: **Assigning DFSP access to a report**

In this demo Yevhen Kyriukha showcases the work that he has done to enhance the legacy restful reporting API. The legacy reporting API now has a new method for adding, editing and deploying reports. These reports can be configured to enforce RBAC authorisation control that checks both that the correct DFSP access and the correct report permissions have been granted. The work effort is encapsulated in epic 2220.

1. Demo the new deployment of legacy report YML with new RBAC permission.
2. Demo/explain how a DFSP would access the report Via OAuth2.
3. Demo how permissions are assigned to a hub user to access the settlement report
4. Demo how permissions and participant access is assigned to a DFSP settlement report
5. Demo how to access the different report output formats that include xls, cvs, html and json.

Demo 3: **Tracing a Transfer**

In this demo Sri Miriyala showcases the tracing of a transfer UI functionality. She explains how problems can be traced starting from the dashboard drilling down into a listview and subsequent transfer details. Sri then demonstrates how to interpret the event traces and explains how this information can be used to identify responses from the individual handlers in Mojaloop. This work is encapsulated by epic 2389.

1. Summary of what this functionality aims to achieve
2. Demo Dashboard
3. Demo filters
4. Demo grid filters and sorting
5. Demo transfer detail modal
6. Demo transfer event tracing while comparing to stages of a transfer diagram

Demo 4: **Tracing a Transfer Reporting API backend**

As part of the delivery of this workstream, a powerful GraphQL API was built along with an event's processor and event reporting database. Yevhen Kyriukha demonstrates the technical part of this API showcasing the power of the versatile graphQL environment, and how this has been implemented with RBAC in a way that maximises data object reuse while combining multiple sources of data. This work is encapsulated by epic 2389.

1. Demo Architecture Diagram with online docs
2. Demo Graph QL demo running transfer summary queries
3. Discuss RBAC permissions, resources and how parts of the solution can be reused.
4. Demo how Graph QL is obtaining data from more that one backend source
5. Demo how to use the graph QL self documentation

Demo 5: **Tracing a Transfer - Settlements**

In this demo Sri Miriyala showcases the tracing of a transfer UI functionality specifically focusing on settlement and the tracing of settlement events. She demonstrates sending transfers and settling within the TTK demo mobile simulator, and then settling from the BizOps framework. Sri then demonstrates how to interpret the settlement event traces and explains how this relates back to the bizOps documentation on settlement.

1. TTK demo showing transfer and settlement.
2. Send another transfer.
3. Settle using Finance Portal.
4. Trace transfers showing settlement events.
5. Demonstrate how to interpret settlement events using biz Ops documentation.

More information - Documentation links

BizOps documentation on Settlement
https://docs.mojaloop.io/mojaloop-business-docs/HubOperations/Settlement/settlement-basic-concepts.html
Link to a settlement example
https://docs.mojaloop.io/mojaloop-business-docs/HubOperations/Settlement/ledgers-in-the-hub.html#settlement-an-example

BizOps documentation on RBAC
https://docs.mojaloop.io/mojaloop-business-docs/HubOperations/RBAC/Role-based-access-control.html

BizOps Framework documentation on RBAC
https://docs.mojaloop.io/business-operations-framework-docs/guide/SecurityBC.html

BizOps Framework documentation on reporting backend functionality
https://docs.mojaloop.io/business-operations-framework-docs/guide/ReportingBC.html

BizOps Framework documentation on Micro-services and JAMStack
https://docs.mojaloop.io/business-operations-framework-docs/guide/Microfrontend-JAMStack.html