



# Helm charts v14

Overview, status, next steps

<https://github.com/mojaloop/charts>



# PI-18 Contributors

- David Fry
- Juan Correa
- Kevin Leyow
- Lewis Daly
- Miguel de Barros
- Shashikant Hirugade
- Tom Daly
- Vijay Kumar
- Yevhen Kyriukha





# Agenda

- Goals
- Status
- Next steps

# Goals



- 1) Take full advantage of **Helm's** capabilities
- 2) Separate “**backend** services” from Mojaloop charts
- 3) Mojaloop deployments to support **k8s** versions beyond **1.21**
- 4) Cleaner Helm charts and configuration
- 5) Alignment to “**Reference** Architecture standards” where possible

# Goals #1

## Take full advantage of Helm's capabilities



1. Support for Common template function libraries to ensure consistency throughout Mojaloop charts
2. Simplified charts
  - a. No more separate requirements.yaml
  - b. Dependencies consolidated in Chart.yaml
  - c. Remove duplicated template functions by using Common Template library ← improved maintainability, less bugs

# Goals #2

## Separate backend services from Mojaloop charts



### 1. Decoupling of external dependency charts

- a. e.g. Database (MySQL, MongoDB), Message-broker (Kafka, Zookeeper), etc  
=> **significant improvement in flexibility, and upgradability**
- b. Easier to replace deprecated or unmaintained charts dependencies
- c. Improved upgradability as Mojaloop charts are managed separately
- d. Flexibility in on how dependencies are deployed and managed on-prem, hosted, etc

### 1. **Best-practice Alignment** to Mojaloop's Upgrade Strategy Guide:

<https://bit.ly/38hfe0q>

### 2. **Blog post** - Helm & Configuration to Resolve External Dependencies & Improve Upgradability in Mojaloop v14: <https://bit.ly/3rOxoO9>



# Goals #3

## Mojaloop deployments to support k8s versions beyond 1.21



1. Support **latest/future k8s APIs**
  - a. Ingress **networking.k8s.io/v1** support required for k8s v1.22
  
1. Support for **Docker** and **Containerd container runtimes**
  - a. Mojaloop v13 Percona backend dependency - doesn't work with Containerd

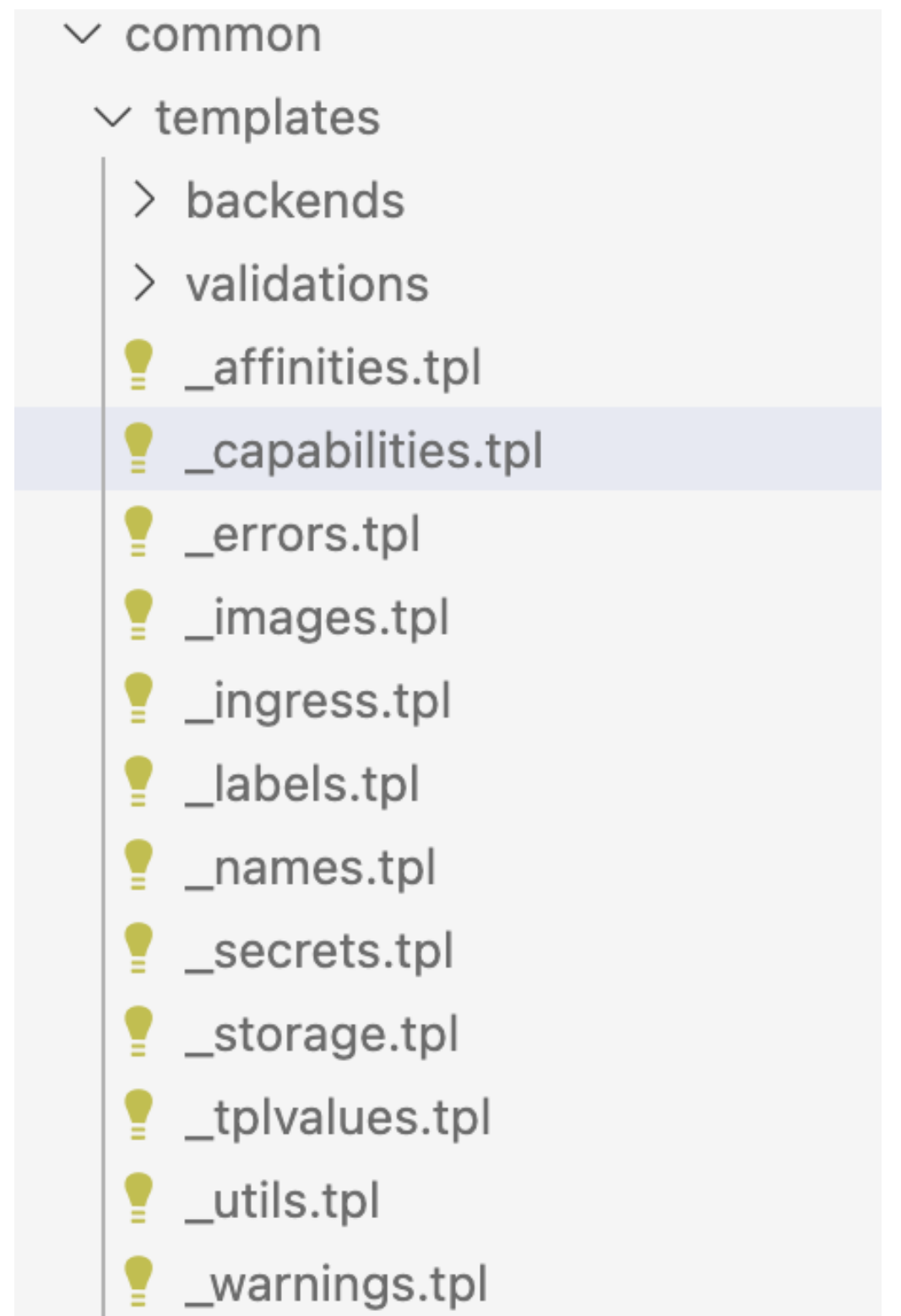
Would have been a significant advantage to have the above changes ready for Microsoft Azure deployments.

# Goal #4

## Cleaner Helm charts and configuration



1. Common template functions (\*.tpl) library
  - a. standard handling of labels, matchers, etc
  - b. resolving backend-dependencies configurations globally, locally, etc
  - c. standard parsers/transforms for configurations, files, etc
  - d. standard handling Ingress k8s API version differences
1. Flattened chart directory structure
  - => much more intuitive, easier to maintain





# Goal #5

## Alignment to “Reference Architecture standards”



1. Chart standards: <https://bit.ly/3Kb5uIB>

1. Charts structures and names are aligned to Ref Architecture (e.g. Bounded Contexts) and Mojaloop naming conventions:

a. ml-api-adapter API → **fspiop-transfer-api-svc**

b. account-lookup-service API → **fspiop-account-lookup-api-svc**

# PI-18 Status

1. Established charts repo
  - a. initial directory structure
  - b. example backend services (mysql, kafka etc)
  - c. common \_tpl helpers
  - d. sample charts
    - i. fsiop-transfer-api-svc (ml-api-adapter API)
    - ii. admin-api-svc (central-ledger admin API)
2. BOP charts and account-lookup-service have been added

```
✓ mojaloop
  ✓ account-lookup-service
    > templates
    ≡ Chart.lock
    ! Chart.yaml
    ! values.yaml
    > admin-api-svc
    > bof
    > chart-admin
    > chart-service
    > common
  ✓ example-backend
    > templates
    ≡ Chart.lock
    ! Chart.yaml
    ! values.yaml
    ≡ values.yaml.bak
    > fspiop-transfer-api-svc
    > mojaloop
    > repo
    > reporting-events-processor-svc
    > reporting-hub-bop-api-svc
    > reporting-hub-bop-experience-api-svc
    > reporting-hub-bop-positions-ui
    > reporting-hub-bop-role-ui
    > reporting-hub-bop-settlements-ui
    > reporting-hub-bop-shell
    > reporting-hub-bop-trx-ui
    > reporting-legacy-api
    > role-assignment-service
    > security-hub-bop-kratos-ui
    > security-role-perm-operator-svc
  > repo
  > scripts
```



# PI-18 Status



CircleCI pipeline updated

1. Helm chart linting

1. Validates Deployment of chart(s) across multiple(3) versions of k8s:  
v1.20, v1.21 & v1.22
- a. verify that chart(s) deploy
  - b. test that health endpoints are accessible and live

1. Automation investigation

- a. populate\_values
  - i. => investigate automation of some parts of chart production.



# Next Steps



1. Migrate charts that support **testing capabilities** (e.g. testing-toolkit, simulators...)
2. Migrate minimum **core-services** (e.g. ledger, adapters, quotes...) charts that can be functionally verified using **Helm Test**
3. Integrate **Helm Test verifications** into **CI-CD** (CircleCI) pipeline to ensure chart(s) work against multiple k8s version
4. **Complete migration** of remaining core-services charts (e.g. settlements...)
5. **Complete migration** of remaining charts (e.g. bulk, pisp...)