# [WFA-1] Workflow Automation Platform - airtel MVP Created: 05/Sep/24 Updated: 25/Sep/24

| | |
|---|---|
| **Status:** | Draft |
| **Project:** | WorkFlowAutomation |
| **Components:** | None |
| **Affects versions:** | None |
| **Fix versions:** | MVP |

| | | | |
|---|---|---|---|
| **Type:** | Epic | **Priority:** | Medium |
| **Reporter:** | Prabhjot Singh Sethi | **Assignee:** | Prabhjot Singh Sethi |
| **Resolution:** | Unresolved | **Votes:** | 0 |
| **Labels:** | None | | |
| **Remaining Estimate:** | Not Specified | | |
| **Time Spent:** | Not Specified | | |
| **Original estimate:** | Not Specified | | |

| | |
|---|---|
| **Attachments:** | image (1).png workflow-automation-Approval LifeCycle.drawio (7259c0c8-aecd-46bc-b43b-fb25706c752b).png workflow-automation-Module Chaining.drawio (2a2fd4ca-489c-433f-80f3-8f0f8de04a95).png workflow-automation-Module-Versions.drawio.png workflow-automation-functional.drawio (1).png |
| **Issue color:** | orange |
| **Rank:** | 1\|i01803: |

## Description

This is currently purely an airtel specific requirement.

**Requirements:**

where the expectation is having a configurable workflow automation platform, providing following benefits

- A platform to automate and manage Day 0 activities for setting up servers, switch and other devices
  - where operators need to perform repetitive actions setting up initial environments and devices, this platform would enable capability of running automated workflows with environment parameters allowing seamless completion of Day 0 activities.
- Furthermore integrations should be provided with CCS for users to trigger some of these workflows allowing configuration and deployment of some of the solution consuming LBaaS and FWaaS
  - This is expected to allow getting away from the hard dependencies for the release upgrade for enabling a new type of device or function
    - which using this system would allow just by updating the relevant workflow trigger
  - Allows Airtel to ask OEMs and vendors to provide the modules for various defined operations

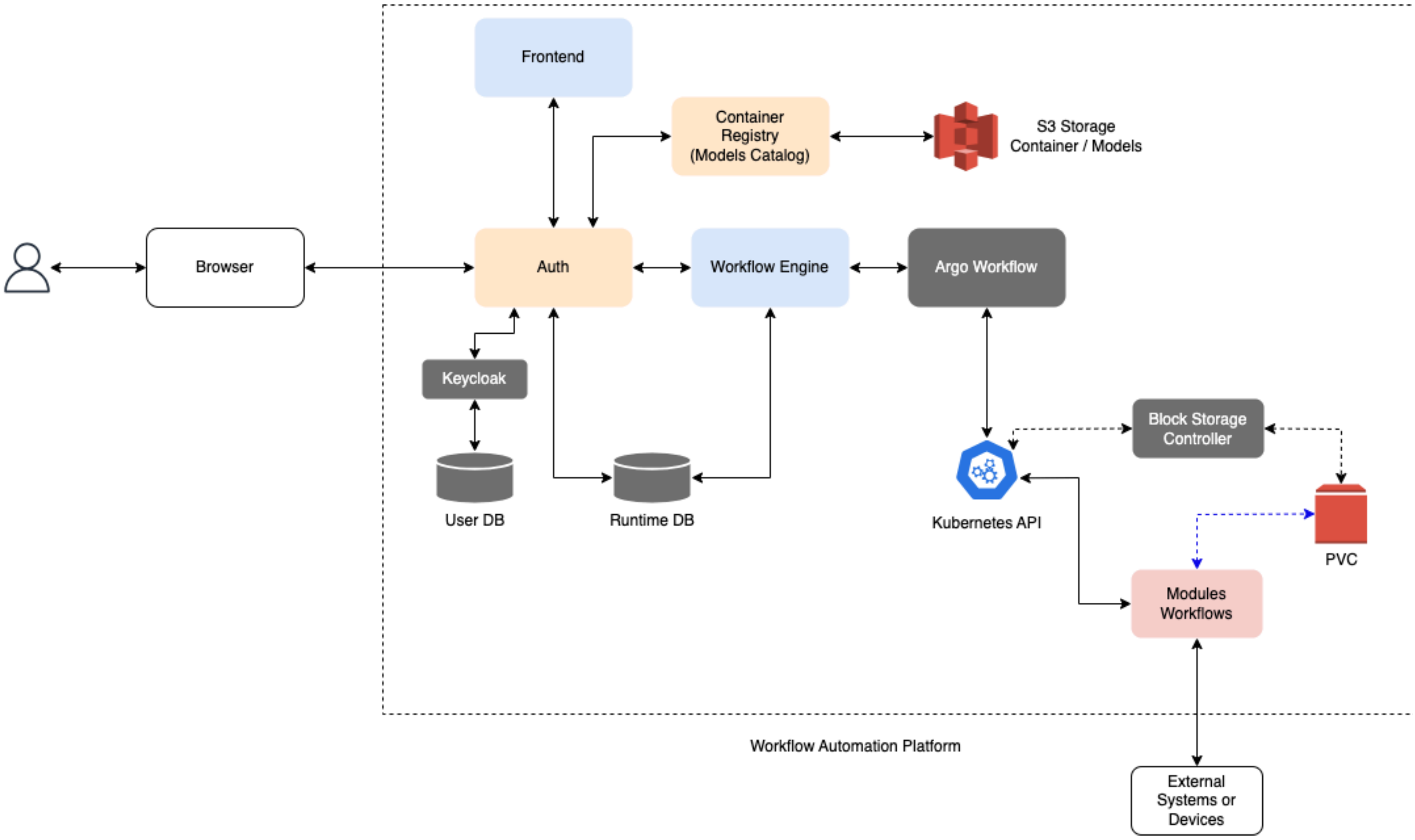
**Structure and Business flow expectations:**

Furthermore, Airtel has following expectations

- Modules:
  - Allow developers / OEMs / vendors to submit newly created modules to work or interact with third party controller or device
    - Typically supporting - Python, Ansible, bash script and terraform
  - Once these modules are created it needs to be submitted for review
  - Upon Approval from the person with relevant permissions (admin) it should be made available for consumption
  - Each module can have multiple verison
  - as part of the supported lifecycle of the modules, it will require capabilities of archiving or deleting the relevant version or the complete modules itself or deprecating specific versions
- Modules Catalog:
  - All the approved modules (and their versions) needs to be rendered as part of the available modules catalog for consumption
- Workflow Templates:
  - Operations users, typically airtel operations team then will be creating pipelines using the available modules to allow delivering specific actions in the overall echo system
    - example there can be following workflow template
      - fetch details from cmdb with respect to a device
      - perform certain action
      - trigger notification email to relevant people
  - These templates will also require to support multiple versions
  - Each template also requires to follow a review approval process
- Workflow executions:
  - Once the workflow template is approved, it is ready for use
  - Users that are onboarded on the portal will have capability to start execution of workflows from the workflow templates by providing the input configuration corresponding to it
  - Only supported output of the workflow executions will be the execution logs
  - Workflow execution is not expected to hold or save any state information

**Solution Proposal:**

On high level the platform is expected to have following microservices connected together to offer the workflow automation platform



**Auth Module:** Workflow automation platform will be interfacing with the user via an existing Auth Module paired with a keycloak deployment will be providing capability of Project based multi tenancy and Authentication and Authorization layer.

**Container Registry:** Platform will also would also be including the container registry services to enable capability of building and storing container images corresponding to the modules along with base images. This will be backed by an S3 compatible storage which would work as the actual store for the container images for modules and their versions

**Frontend:** Platform will require a UI rendering for which a nex micro services would be written allowing to work with different personas of users. This microservice will be responsible for rendering the pages on the browser.

**Workflow Engine:** Workflow Engine is the core service for the platform which would be providing the functional capabilities of the platform described in the subsequent sections

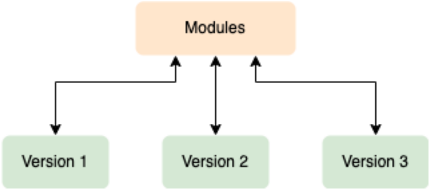
**Argo Workflow:** The platform looks forward to leverage opensource Argo Workflow engine which will be used by the platform to trigger the execution of created pipelines. Where Argo Workflow engine is an extension of Kubernetes platform with capability to executing the pipelines in sequence along with available integrations with Object storage providing capability to store/save results from workflow or the execution logs.

Since Argo workflow engine underneath depends on Kubernetes platform, scalability of parallel executions will be limited by the platform capability of handling multiple pods. As part of the proposal we should be leveraging **CKP** as the Kubernetes platform while being deployed over and above virtual machines to leverage infrastructure in an optimised manner. However, otherwise the platform will be useable with any Kubernetes platform

**Note:** By definition the platform is working as a workflow executor and thus will be functional in a stateless manner, where the platform will not store any contextual information as part of an individual workflow execution apart from the execution logs. Wherever it is expected to work on a contextual information the workflow pipeline is expected to include modules / functional blocks to fetch the relevant information from different sub systems like CCS, CMDB etc.

### Modules and its Life Cycle:

Modules are the basic fundamental blocks available in the system, which will be defined to perform a very specific function or an execution for an external device or system. Each module will maintain multiple versions, along with the information on creator, approvers etc as part of the catalog entry.



A User will have capability to create these modules or versions for an existing module and submit the request for review. as part of which they will require to configure following:

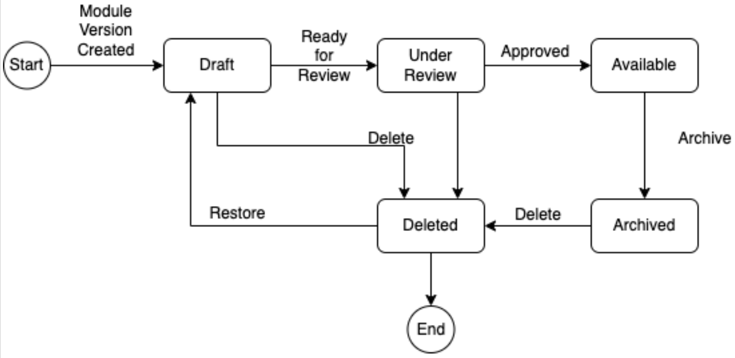
- Base container image
  - out of the box support will be added for Python / Golang / Ansible / Bash and Terraform
  - Custom images also should be accomodateable
- Functional Logic in the form of a file upload or on browser editor or a git repo
  - if added using git repo the - checked out code will be frozen as part of container image build submission for module review / catalog
- Defining set of system level inputs needed as environment variables
- If Persistent Storage is required for reading the file based inputs from previous stages of workflow and providing the output files to further stages of workflow
  - There will be an option for module to define if this storage will be mounted as RW or RO

  - path for this storage will be made available as part of the ENV variable
  - Size of the storage will be limited to base 100 Mib per workflow

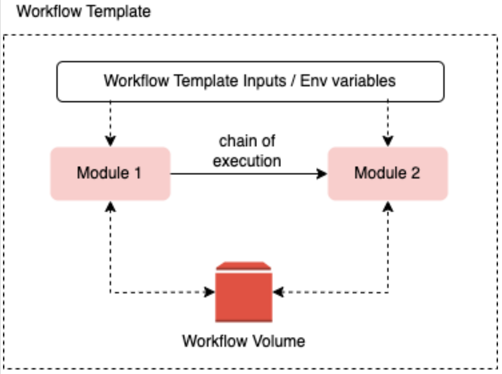
Expectations from Modules:

- Modules will be expected to consume environment level inputs from ENV variables for context like org / project / az etc
- whereas, the stage level inputs from the files from presistent storage

Module LifeCycle Workflow:



Following diagram depicts module chaining inside a workflow

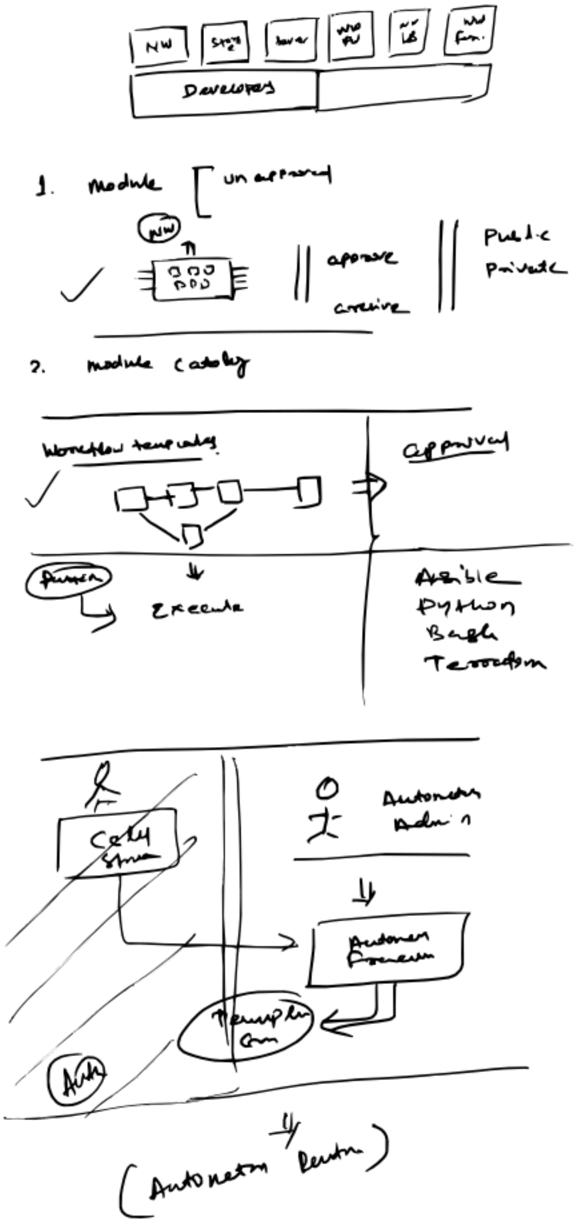


**Future Considerations:**

Following functionalities are desireable but won’t be included as part of the effort right now.

- Security considerations for model repositories
  - Artifact vulnerability scanning
  - Artifact notary signing

**Additional Reference Notes:**



Comments

Comment by Manik Sidana [ 23/Sep/24 ]

**A note on argo-workflows version:**

argo-workflows follow semantic versioning. argo-workflows patch release are typically released every 3-4 weeks. Minor release follow after the rc releases. In the rece
months(post v3.5), however, release planning is typically community driven. The community also takes care of release documentation for upgrading from one minor re
about breaking changes. The recent most release is v3.5.11, released in Sep, 2024. As argo-workflows is a pure community driven project, each release introduces a
suggested to use Recent-1 version so that community fixes would be readily available in case of blocker issues while integration. As a result, it is recommended to use
new features[https://terrytangyuan.github.io/2023/08/14/argo-workflows-v3.5/#:~:text=Users%20no%20longer%20need%20to,in%20%2311132%20by%20Jason%20Meridth.&text=Additionally%2C%20thanks%20to%20%2310145%20by,ca
would be missed out. As argo-workflows are intended to be used majorly from the perspective of offering workflows, ready-made artifact push to S3 and a few others,

**Argo releases timeline:**
v3.5.1- Nov 4, 2023
v3.5.0 - Oct 13, 2023
v3.4.4 - Nov 30, 2022
v3.4.1 - Oct 1, 2022
v3.4.0 - Sep 19, 2022

Comment by Manik Sidana [ 25/Sep/24 ]

As discussed yesterday, we would be going ahead with argo-workflow version 3.5.X and not 3.4, considering 3.4 is more than an year old. However, for 3.5.X patch re
X.

Generated at Thu Sep 26 06:46:06 UTC 2024 by Sushant Kumar Jha using Jira 1001.0.0-SNAPSHOT#100266-rev:70ee8b5693c3bd64816dba1cc8d498bf7ed9192a.