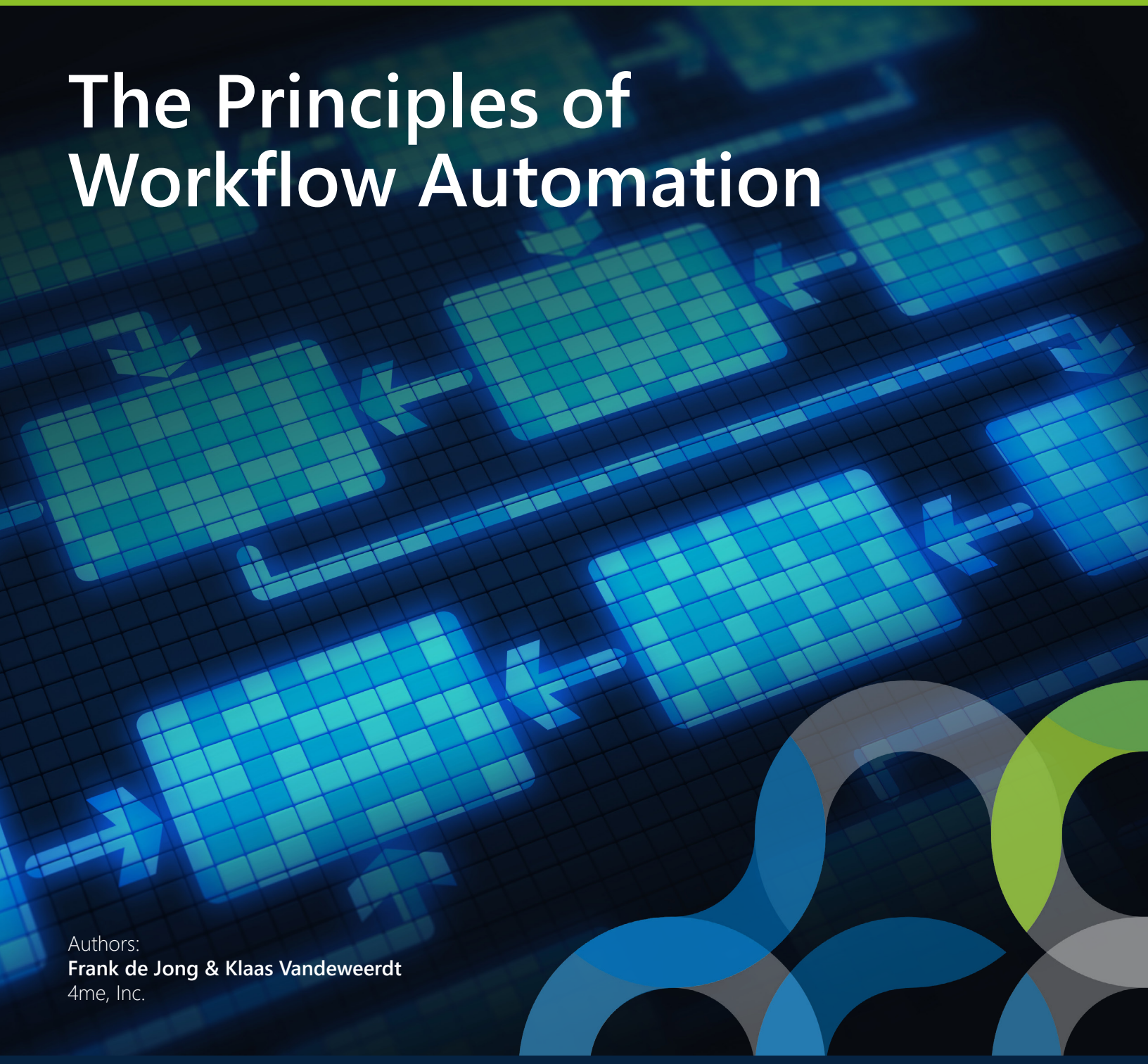
WHITE PAPER

The Principles of Workflow Automation



Authors:
Frank de Jong & Klaas Vandeweerdt
4me, Inc.

For more information visit: **4me.com**

Introduction

Workflow automation has become a necessity for many organizations. Many industries suffer from a shortage of skilled workers, while organizations need to be able to respond quickly to opportunities and changes in demand to remain competitive.

The implementation of workflow automation can bring numerous benefits to an organization, including increased productivity, improved accuracy and consistency, a higher level of security and compliance, and higher customer satisfaction levels. These results do not happen overnight, however. Besides excellent technology, successful workflow automation requires strong governance and a structured, consistent approach.

This white paper provides important principles for orchestrating and automating workflows using the 4me platform. With the release of the 4me Workflow Automator, 4me now provides all the required capabilities for end-to-end workflow automation. However, the principles in this white paper still apply when using another automation platform.

Workflow Orchestration

The 4me Service Management platform provides all the required capabilities for modeling, executing, and monitoring all workflows in your organization. All tasks in the workflow and their order and dependencies are identified and included in the workflow template. During the execution of the workflow, these tasks are assigned in a predefined order to the responsible teams or specialists. Using automation rules, the management and administration of workflows can be automated to a high extent. Reports and analytics provide valuable metrics for reviewing and optimizing performance. In the remainder of this white paper, we will refer to the capabilities described above as **workflow orchestration**.

Automating Workflow Execution

The 4me Workflow Automator provides powerful functionality to further optimize and automate the execution of your workflows. The platform is low code, does not require development skills, and comes with over 1,000 out-of-the-box connectors for well-known systems and data sources.

Connectors, triggers, and actions are combined into automation recipes that can be reused and shared. The logic in a recipe allows you to execute the steps in a certain order.

Data exchange between 4me and the Workflow Automator is based on the 4me GraphQL API. This API allows very precise and flexible queries, minimizing the number of required API calls, which optimizes performance and helps you not to exceed any API quotas. Another advantage of using the GraphQL API is the automatic schema update. A schema describes the structure of the data, the relationships between entities, and possible actions that can be performed on this data. When the 4me schema is updated, the changes are immediately available in the Workflow Automator. The user interface of the 4me connector will automatically update as well. This means there is no need to update connectors when using the GraphQL API.

While these capabilities provide a lot of flexibility and possibilities, they also introduce a common pitfall: redundancy. You now have two platforms allowing you to model and automate workflows. Spreading data and activities across multiple platforms has several downsides, including increased complexity, increased costs, increased risk of unauthorized access to data, and inadequate reporting.

To avoid the redundancy pitfall and optimally benefit from the power of both platforms, we recommend automating your workflows according to the following principles:

Orchestrate workflows in 4me

Workflows, including all required tasks, are modeled and maintained in 4me. Ensure all tasks, including those that may be automated, are added to the workflow templates. This approach has several advantages:

- Careful planning. When the entire end-to-end workflow is designed in one place, overlooking essential steps is less likely.
- Traceability. Status, progress, and metrics about costs and performance are available from a single set of related records on a single platform.
- Fallback when automation fails or is unavailable. Having all the details of the workflow available allows teams to manually complete workflow tasks as a fallback in case of automation issues.
- Segregation of duties. Only people with the appropriate roles are able to access data and modify workflow details.

Everything-as-a-workflow

Adopt the habit of always and only triggering automations from a workflow in 4me. Whenever an automation is required, a related workflow should be present in 4me, with an automation task for triggering the desired automation recipe.

Scheduled automations can easily be triggered using recurring workflows. When an automation is successfully completed, the accompanying task in 4me is completed. This provides traceability and makes it very easy to identify failed or delayed automations.

Automate the execution of single workflow tasks

An automation recipe should take care of the completion of a single task. Subsequent automations are triggered from subsequent tasks in the workflow. Do not trigger follow-up automations directly within the automation platform if they are not necessary to complete a single task.

The desired level of traceability is determined during the creation of the workflow template. A single task can contain multiple actions if these can all be completed by a single specialist without the need to involve others.

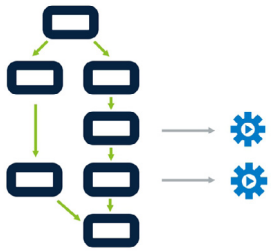
Aim for reusability

Always aim for maximum reusability of automation recipes. Task templates in 4me should be relatable to a single automation. Each time the task template is reused in a new workflow template, the accompanying automation is also reused. We will provide a practical example later in this white paper.

Another benefit of reusability is that workflow managers can now determine which tasks should be automated by reusing the relevant task template. There is no need for the involvement of an automation specialist.

Think beyond the happy flow

Exceptions happen. Some exceptions, like delays or timeouts, can be handled by automation recipes. Failures that impact the workflow should be handled by exception flows in the workflow template. If this is not possible or feasible, the automation task should be set to the status 'Failed'. This will cause the workflow to be set to the status 'Progress Halted'. The Workflow Manager is notified and can determine the next steps.



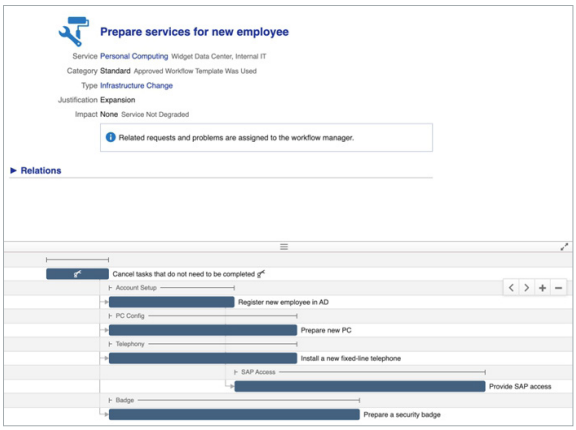
Schematic example of a workflow and related automation recipes. Note the 1:1 relationship between steps in the workflow and the automations.

A Practical Example

Let's look at a practical example using a simple workflow for onboarding a new employee. This workflow contains separate tasks for preparing accounts and infrastructure. This is a simplified example to explain the overall approach and the various functionalities involved. A more complex (and perhaps more realistic) example will follow the same principles.

Creating the workflow template in 4me

The template for the example workflow is displayed below. For the scope of this white paper, the first two tasks are the most relevant.



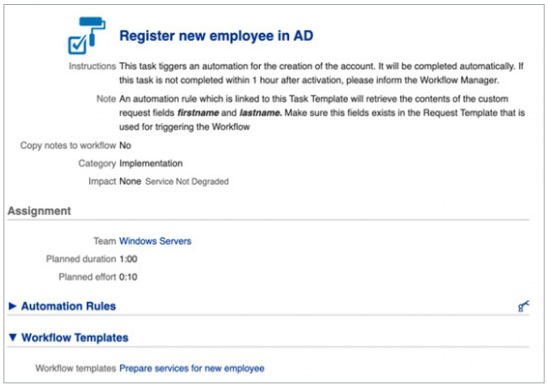
Screenshot of a Workflow Template in 4me for onboarding a new employee

Canceling unnecessary tasks

The first task is for the Workflow Manager; to determine if all tasks in a certain instance of the workflow are required. For example, the task of installing a fixed-line telephone might be irrelevant for a new employee who will not be working in an office. The cancellation of these tasks can be determined with the information provided when the request that triggered this workflow was created. Canceling tasks can be automated using automation rules.

Automating tasks

The next task in the workflow will be assigned to the Windows Servers team to create a new Active Directory account.



Screenshot of a Task Template in 4me

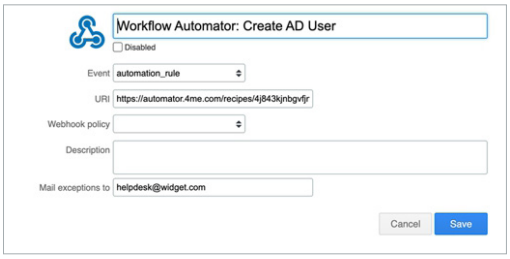
This task will be an ideal candidate for automation.
It involves five steps:

Step 1: Create the automation recipe

The first step is to create an automation recipe for creating the AD account. We could decide to create a single recipe for the creation of an account. Another option would be to combine the creation, updates, and deactivation of an account in a single recipe.
The automation recipe can be triggered remotely and expects a certain payload with essential information like the name, department, and job description of the new employee. This recipe should be triggered from 4me, including the correct payload.

Step 2: Create a webhook in 4me

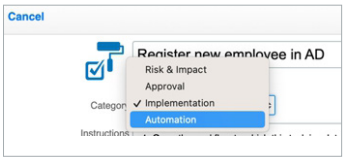
A trigger to an external system in 4me is called a webhook. For each recipe, a specific webhook must be created. An example of a webhook configuration is displayed below. The event field defines how and when the webhook is triggered. In this example, it is set to automation_rule to ensure we can trigger this webhook from an automation rule. The URI field contains the URI of the application that should receive the webhook. In our example, this will be the 4me Workflow Automator. A webhook policy allows you to encrypt the contents (payload) of the webhook so the receiver can be sure the information was not tampered with.



Screenshot of a Webhook in 4me

Step 3: Update the task template

Update the task template in 4me and add an automation rule. Tasks that will be completed by automation should not be assigned to teams or specialists to avoid unnecessary notifications and cluttering the Inbox in 4me. For this reason, the category of the task template for creating the AD account is updated to 'Automation'. This will remove the assignment from the task.

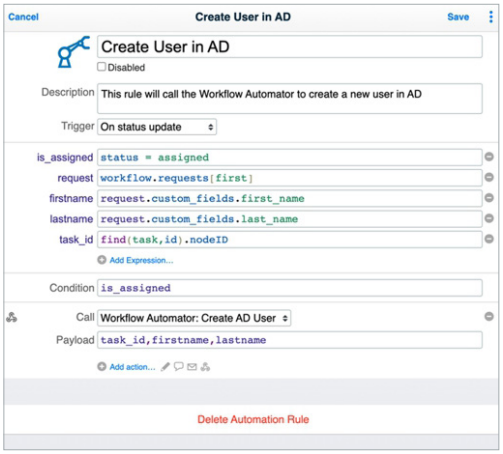


Updating the category of the Task Template

Step 4: Add an automation rule to the task template

The webhook we created in step 2 should be triggered as soon as the task for creating the AD account is assigned. This trigger should also include the expected payload.

4me's automation rules allow us to do this. Below is a (simplified) example of an automation rule:



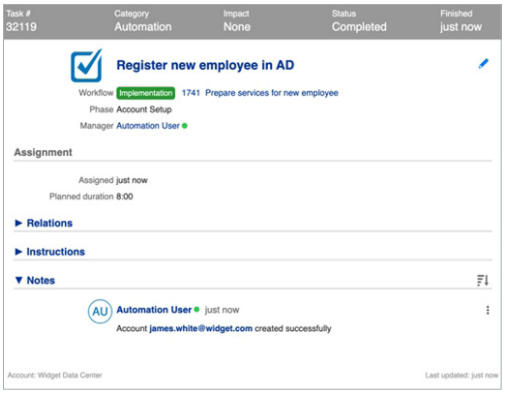
Screenshot of an Automation Rule in 4me

The first name and last name of the new employee are retrieved from the request. The model, id, and action fields are used to generate the payload and indicate the type of record the webhook call originates from, the ID of the task, and the desired action of the automation recipe.

With this information, the automation recipe can create the request AD account and update the task in 4me accordingly.

Step 5: Execute the workflow and review the result

The final step is to start another instance of the workflow and review the result of the automated task.



Updating the category of the Task Template

This example shows how the various functionalities of 4me intertwine. For specialists using 4me, the demonstrated steps will look very familiar. There is no special 'automation knowledge' required to prepare the automation of the workflow, which also allows for clear segregation of duties. Workflow Managers maintain and register workflows in 4me and decide when to trigger the integration platform. The integration specialists then take care of creating and maintaining the automation recipes. A clear description of the expected payload for each recipe will allow 4me specialists to create the necessary automation rules for preparing the data and constructing the payload.

Conclusion

In this white paper, we outlined best-practice principles for automating workflows. With the addition of the 4me Workflow Automator to the 4me platform, customers have all the required capabilities to fully orchestrate workflows and their automation in one platform. We also explained that the same principles apply when organizations use third-party automation solutions.

The most important prerequisite for the successful orchestration of workflow automation is a structured and consistent approach. Orchestration is done in the service management system, and workflow task execution is handled by the automation solution(s).