Discovering MISP workflows

Improving automation in threat intelligence

CIRCL / Team MISP Project

MISP Project
https://www.misp-project.org/

EU MITRE ATT&CK
Community Workshop

MISP
Threat Sharing

Bringing workflows into threat intelligence platform

After multiple years, MISP users have reach a significant maturity level:

- Events with **complex TTPs, objects and attributes**;
- Exhaustive context such as **MITRE ATT&CK**, tags and relationships;
- Availability of **external modules and services** (e.g. from expansion services to third-party CTI);
- Comprehensive **processing pipelines** for threat intelligence are available;

Where is the glue?

- Initial idea came from GeekWeek7.5

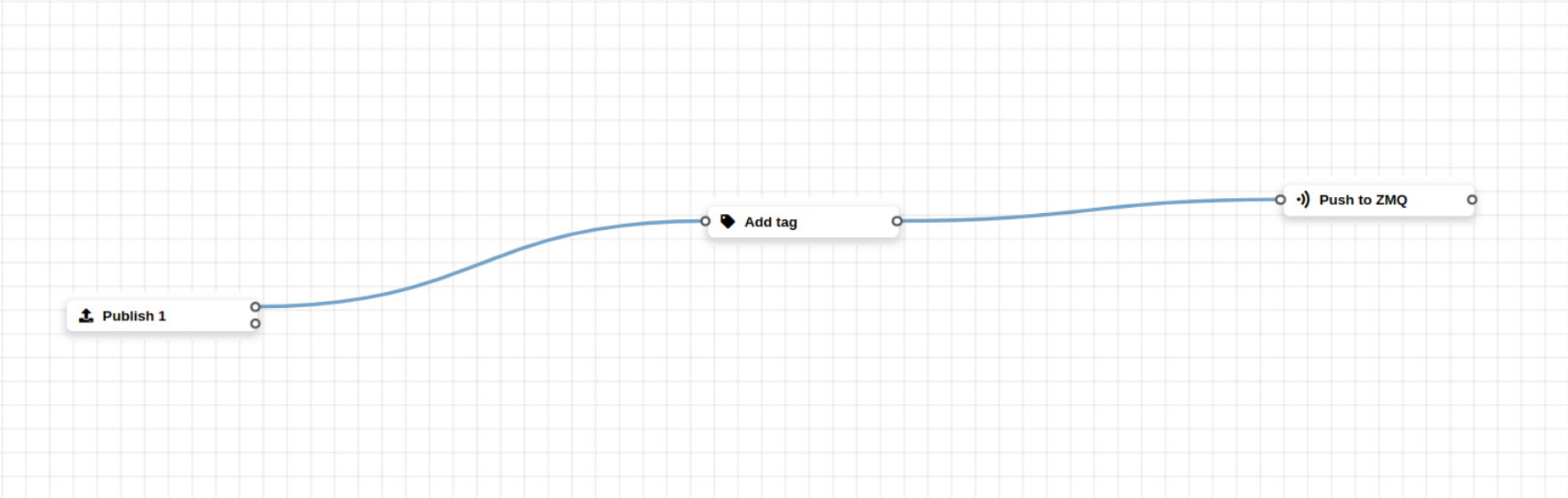
- Experienced users wanted to have a way to **trigger actions and to modify to behavior of MISP** and especially leveraging what they have in their MISP platform.
- **Creating workflows for any of the steps** in MISP (creating attributes/objects, publishing and sharing information, ...).

Simplistic overview

1. User/API Interraction
2. MISP handles the request
3. MISP executes workflows listening to the trigger

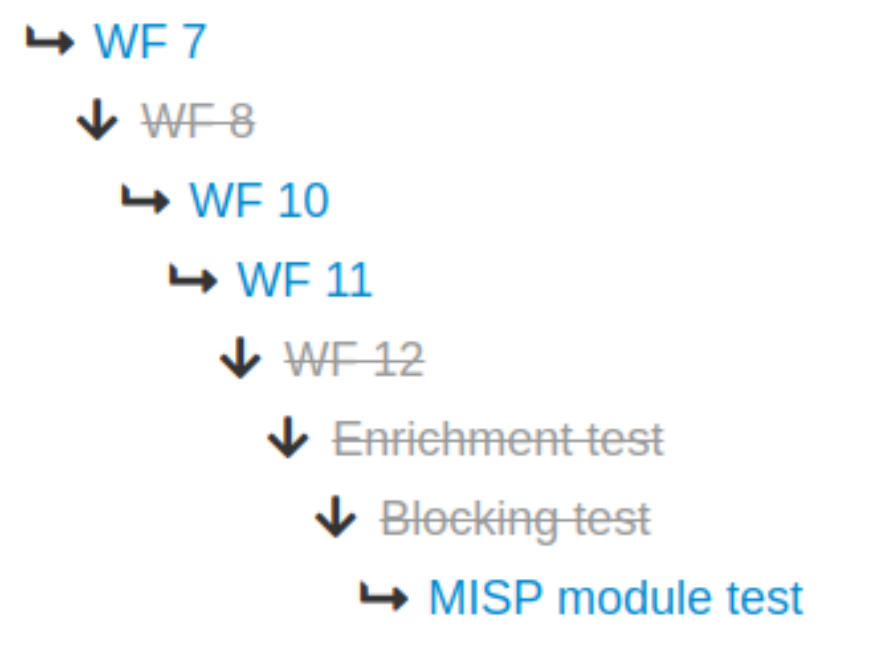
Terminology

1. **workflow**: Sequence of actions to be executed
2. **execution path**: A path composed of actions to be executed sequentially
3. **trigger**: Starting point of an `execution path`. Triggers are called when specific action are done by MISP



Workflow execution

1. A trigger is called
2. Collect workflows listening to called trigger
3. Execute workflows in the saved order



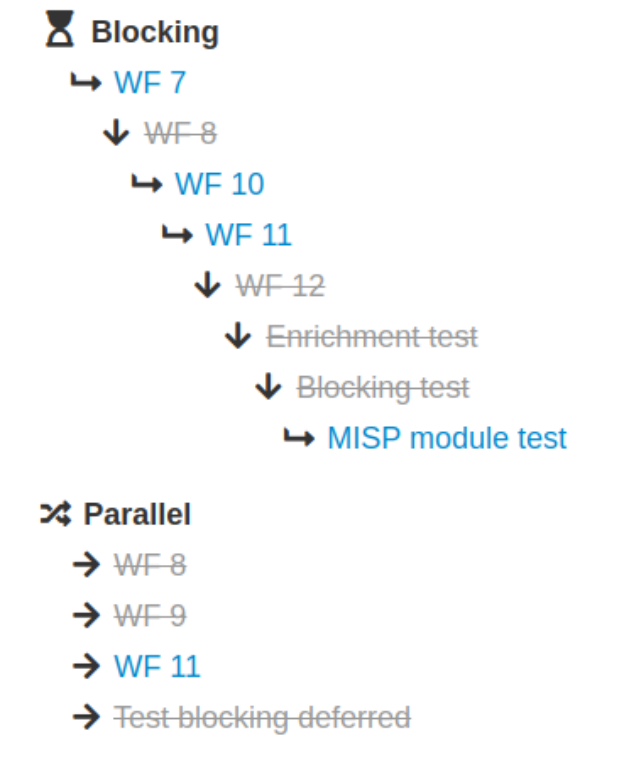
Execution Paths

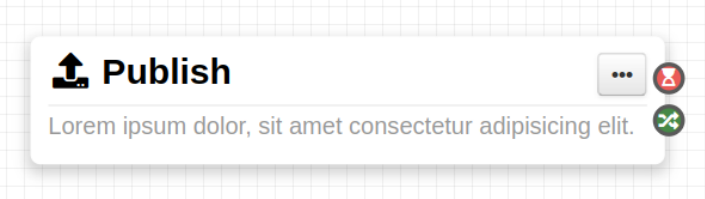
Currently 2 types of execution path:

- **Blocking**: Execution is stoped in case of error
  - Current workflow's blocking execution path is **stopped**
  - Any other blocking path of next workflows **will not be executed**
- **Non-blocking**/Deferred: Stop execution for current path only
  - Current execution path is **stopped**
  - **Resume** execution of remaining paths
  - Paths from other workflow will be **executed**

Execution Order and Execution Types

- **Blocking** paths from all workflows are executed first in the saved order
- If any blocking executions failed, the action that called the trigger will **be stopped**
- **Parallel/Deferred** paths from all workflows are executed. The order is irrelevant





Publishing example

Example:

1. An Event is published
2. MISP starts the publishing process
3. MISP executes a workflow listening to the trigger
   - **execution success**: Proceed publishing
   - **execution failure**: Stop publishing, log the reason and report the failure to the user

Execution context

- Workflow can be triggered by any users
- However, the user for which the workflow executes is the workflow creator
- This is to make sure users with a higher privilege will have their workflow correctly executed

Workflow modules

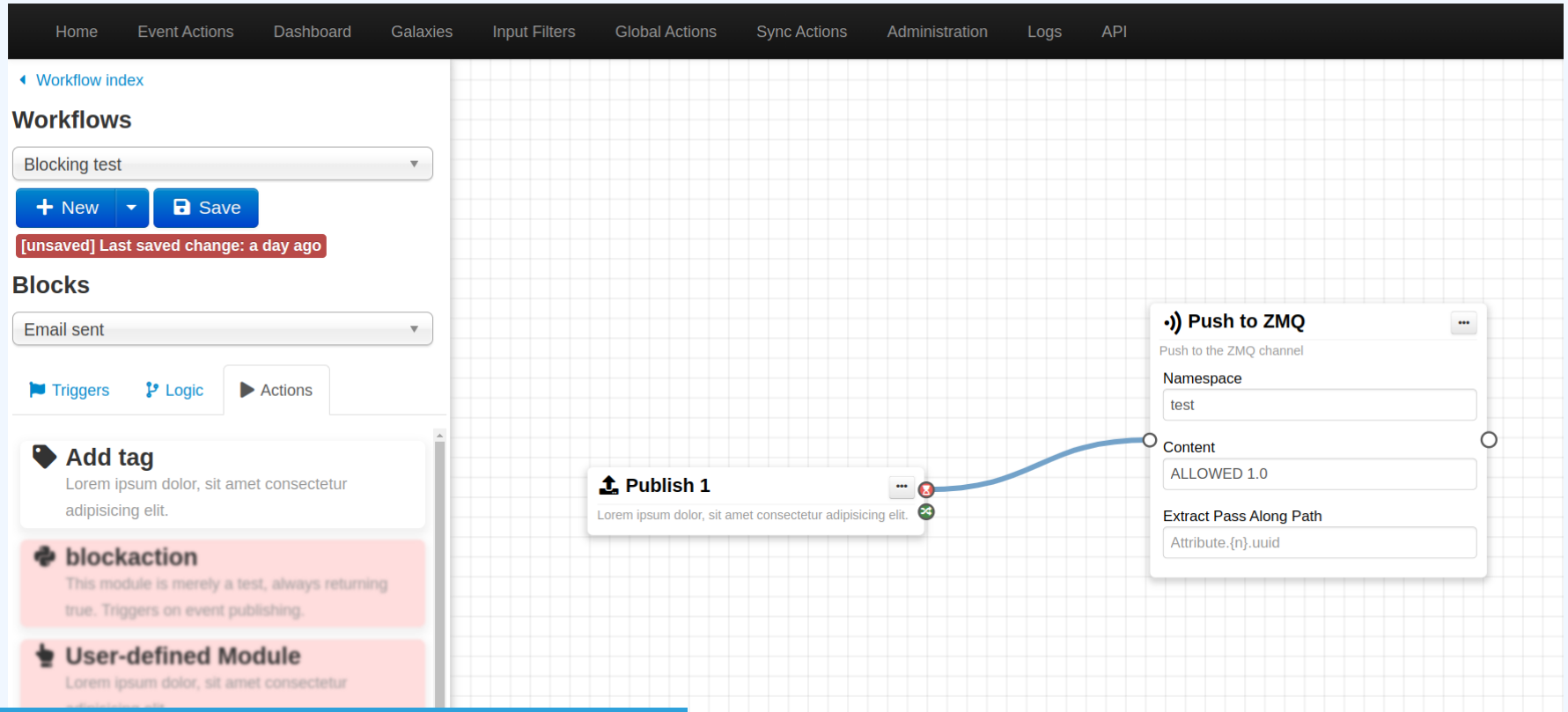
- 3 types of modules
  - `trigger`: Entry point of the execution
    - Event publish, email about to be sent, feed data about to be saved, ...
  - `logic`: Allow to redirect the execution flow.
    - IF condition, fork the blocking execution into a non-blocking one, ...
  - `action`: Modules that can modify data, prevent execution or perform additional actions
    - Publish to ZMQ, perform enrichments, block the execution, ...

Workflow modules

- `action` modules can be from 2 sources
  - `app/Model/WorkflowModules/action/[module_name].php`
    - Written in PHP
    - They can use MISP's built-in functionalities (restsearch, enrichment, push to zmq, ...)
    - Faster and easier to interact with for those having internal knowledge of MISP
  - `From the misp-module service`
    - Written in Python
    - They can use any python libraries
    - Easier to write
    - New module type `action`
- Both systems are **plug-and-play**

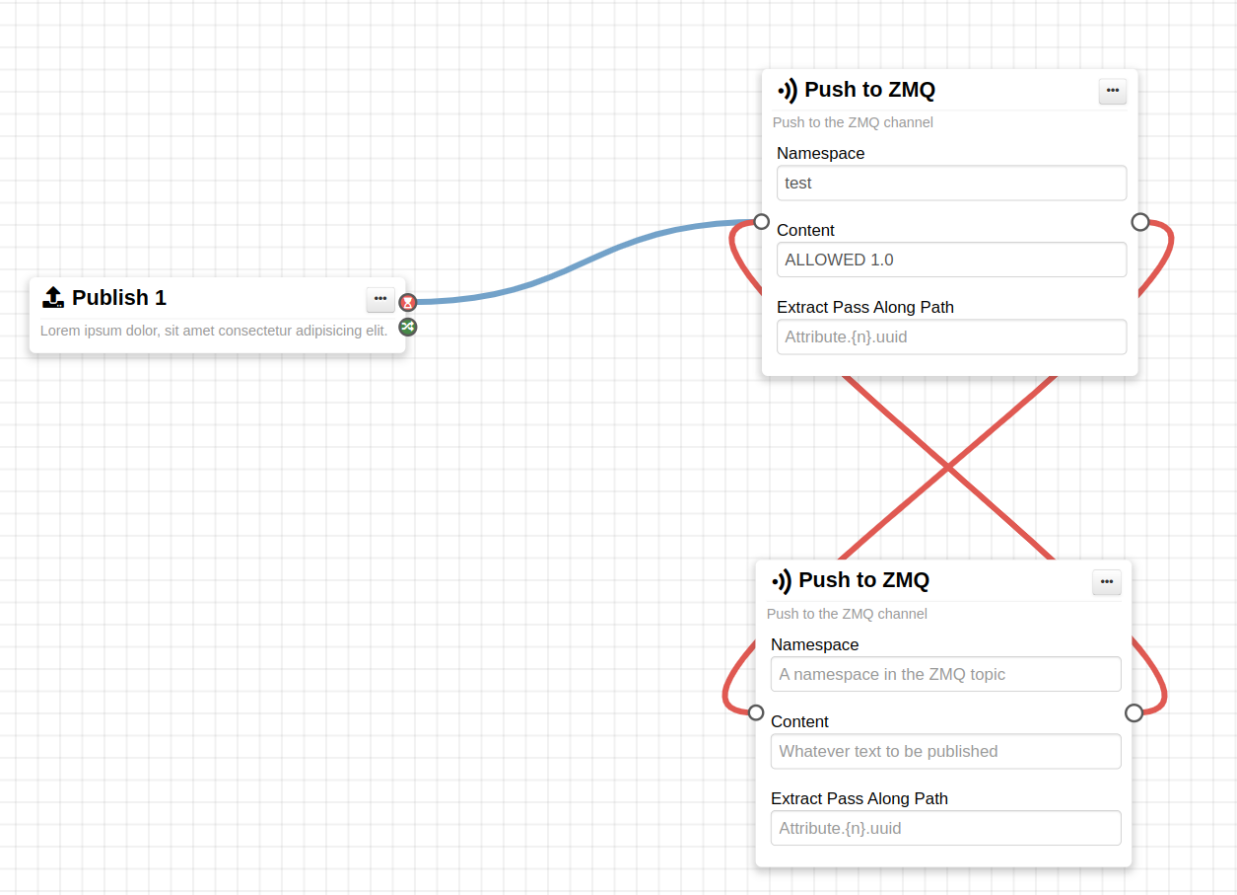
Creating a workflow with the editor

1. Drag a `trigger` module from the side panel to the canvas
2. Drag an `action` module from the side panel to the canvas
3. From the `trigger` output, drag an arrow into the `action` input (left side)
   - You can choose between a `blocking` and `non-blocking` execution path by using the associated trigger output



Operations not allowed

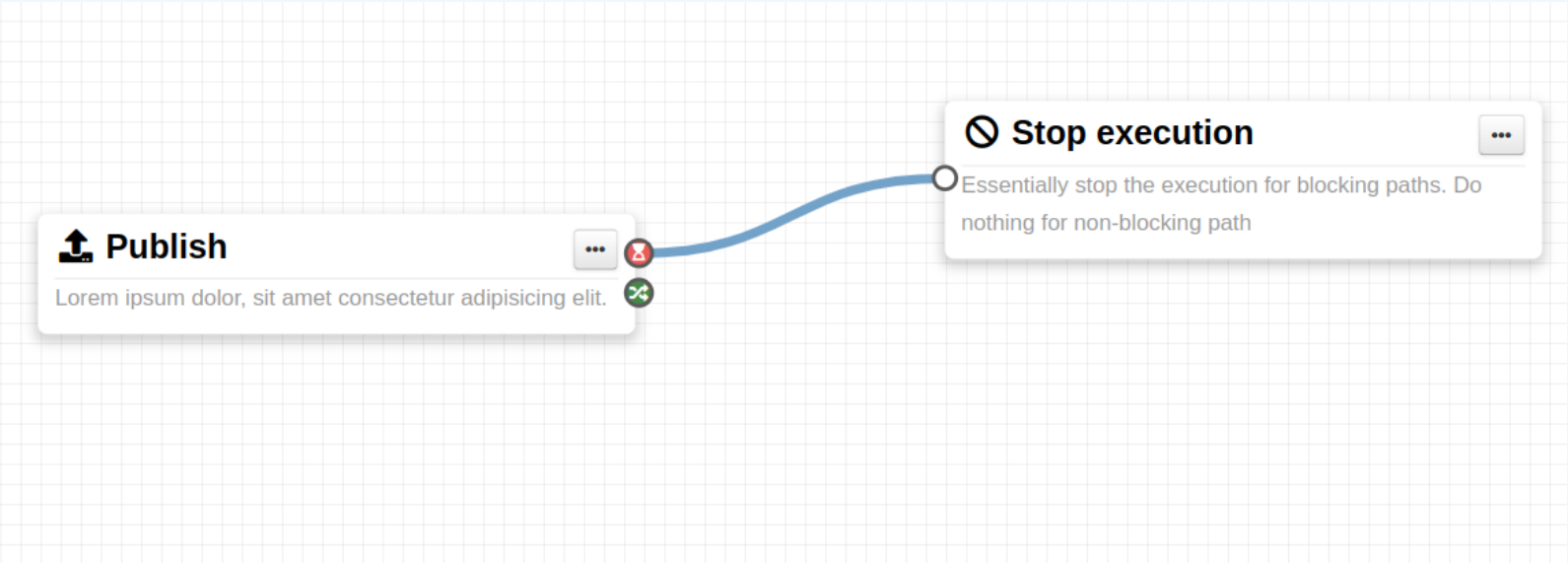
- Create an execution loop



- Use the same trigger twice

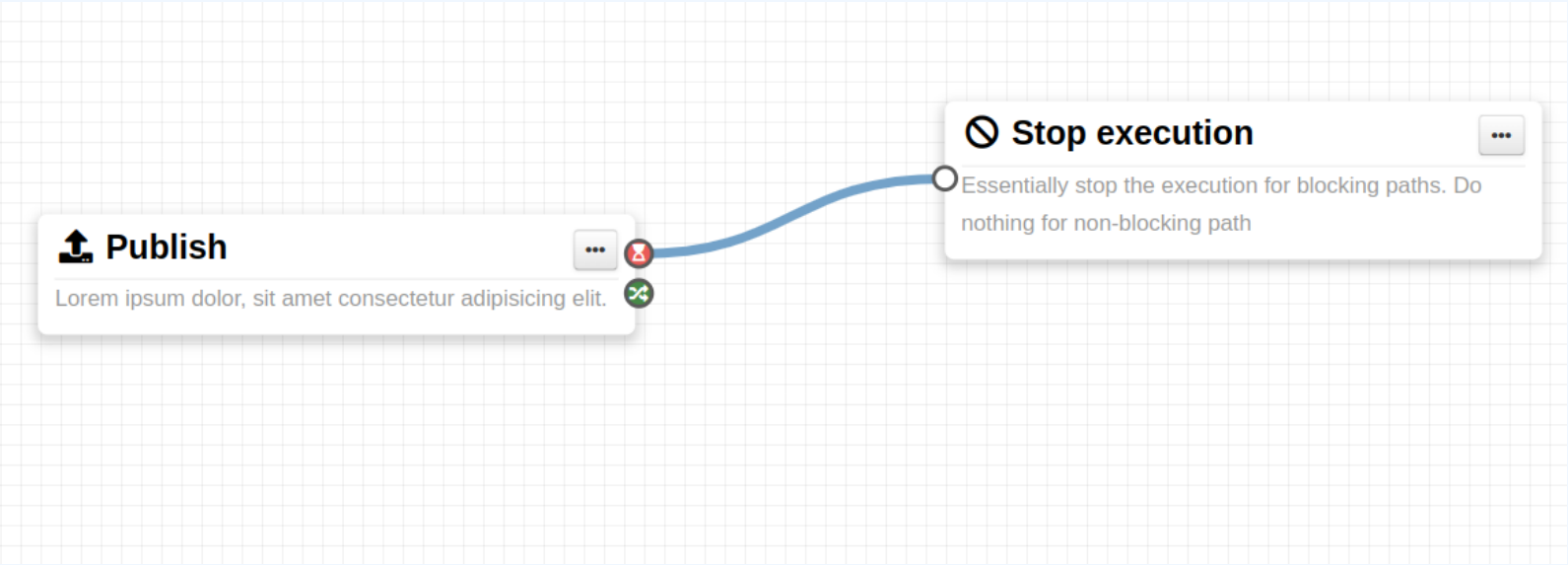
Learning by examples

Workflow example 1



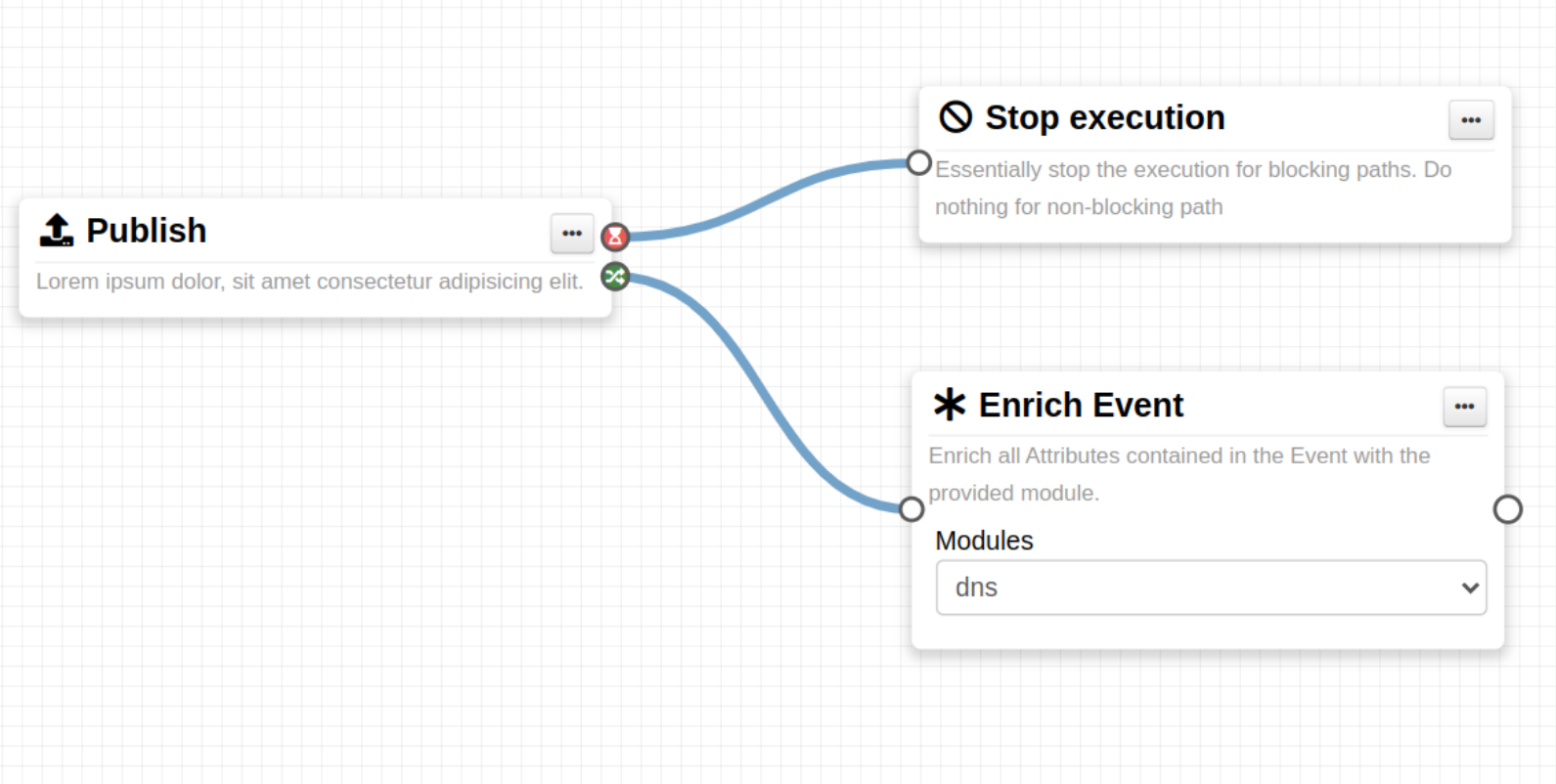
1. Will the next blocking path (from another workflow) be executed?

Workflow example 1: Answers



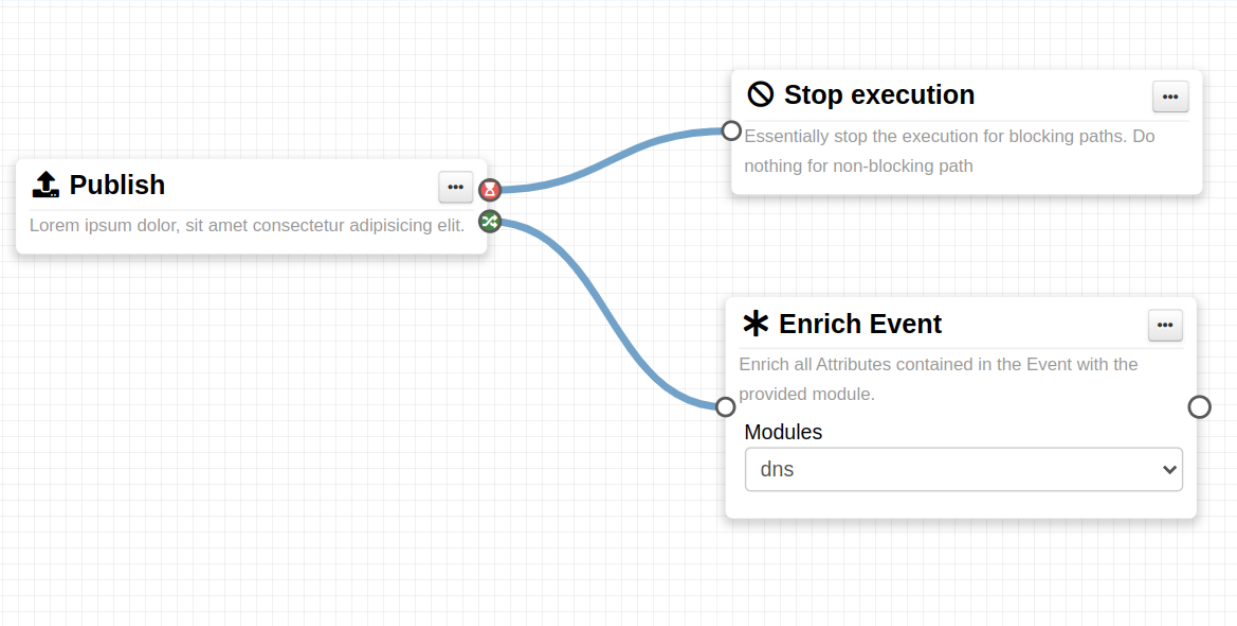
1. Will the next blocking path (from another workflow) be executed?
   - **No**. We are in a blocking path. As the execution has been stopped, no other blocking paths will be executed.

Workflow example 2



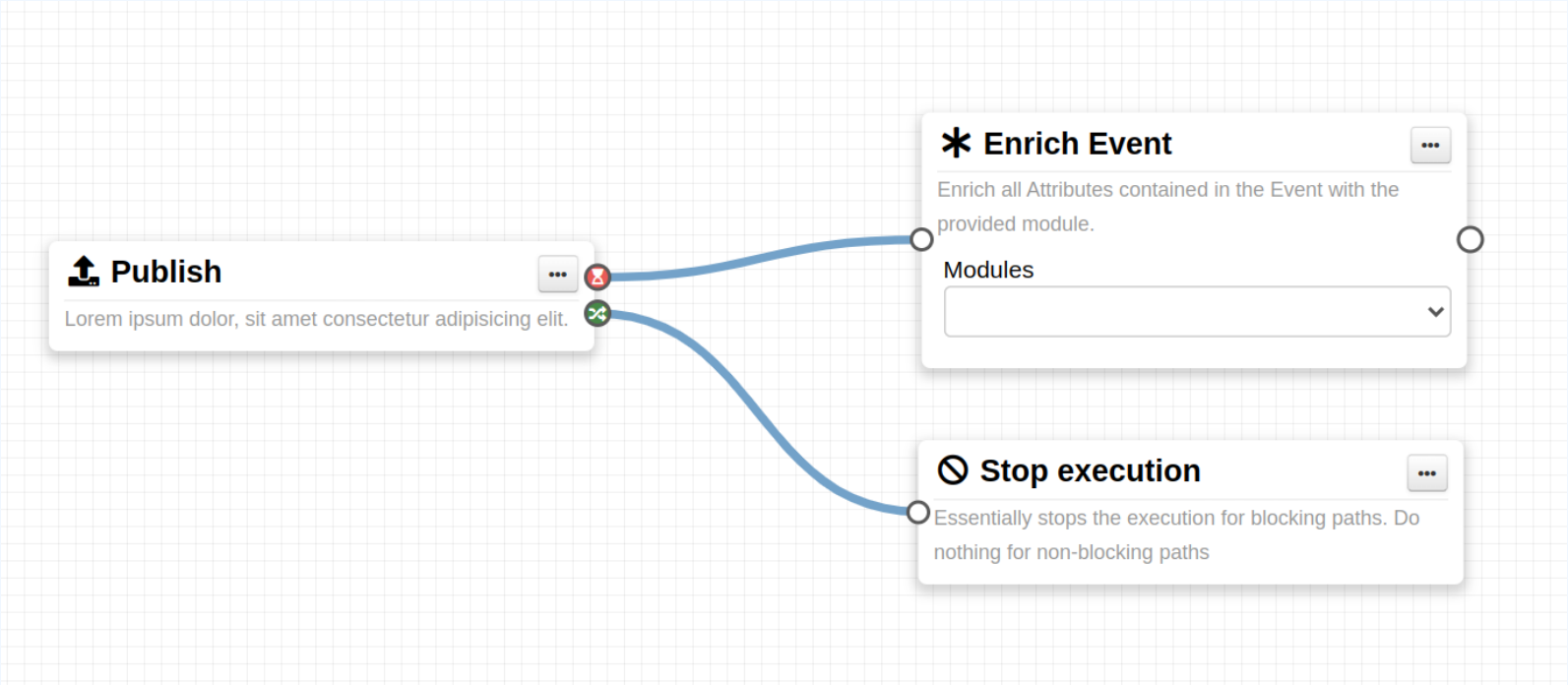
1. Will the next blocking path (from another workflow) be executed?
2. Will `Enrich Event` module be executed?

Workflow example 2: Answers



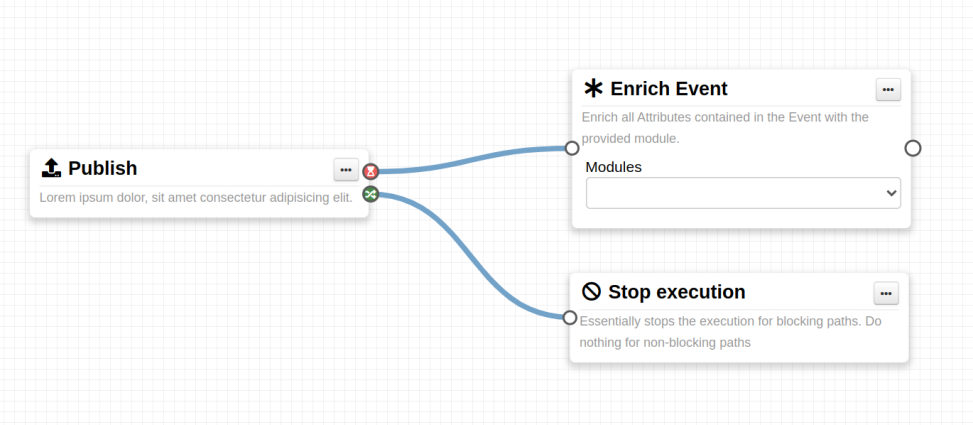
1. Will the next blocking path (from another workflow) be executed?
   - **No**. Same reason that before
2. Will `Enrich Event` module be executed?
   - **Yes**. The module is in the non-blocking path. Regardless of the result of the blocking path, it will be executed.

Workflow example 3



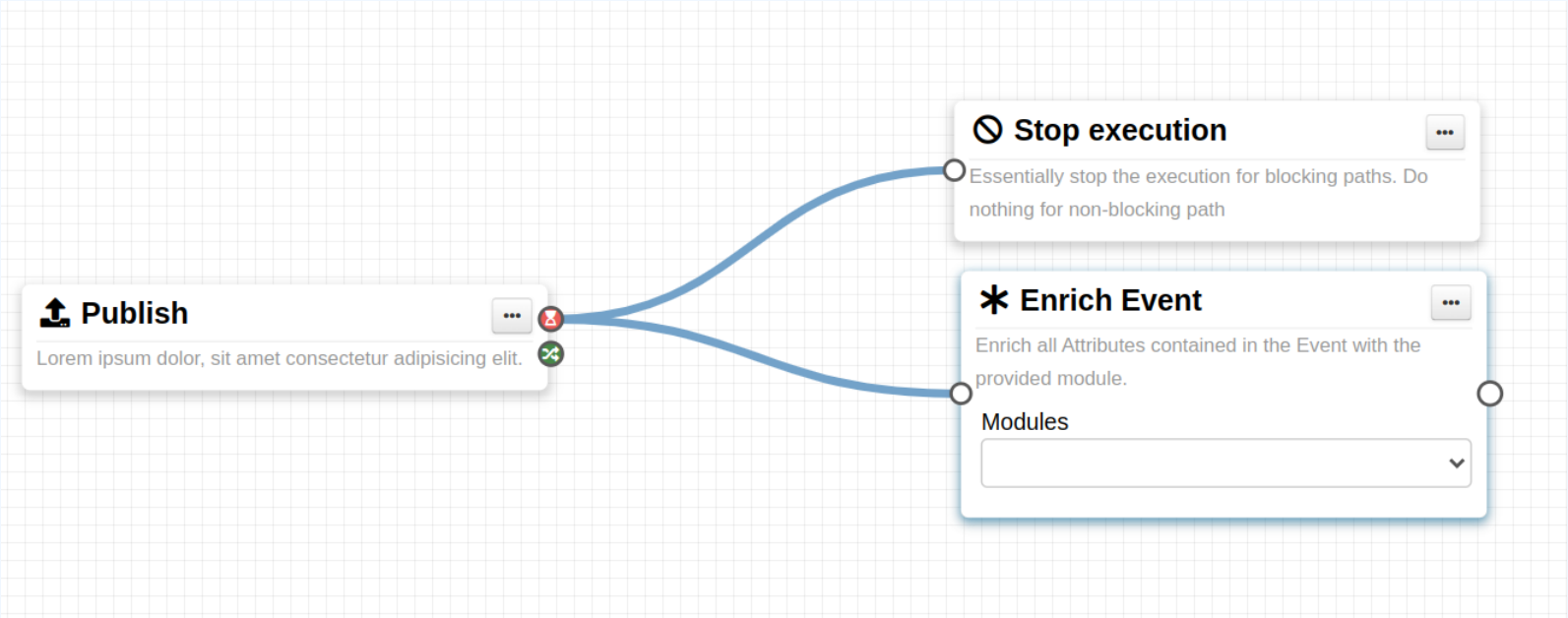
1. Will `Enrich Event` module be executed?
2. Will the next blocking path (from another workflow) be executed?

Workflow example 3: Answers



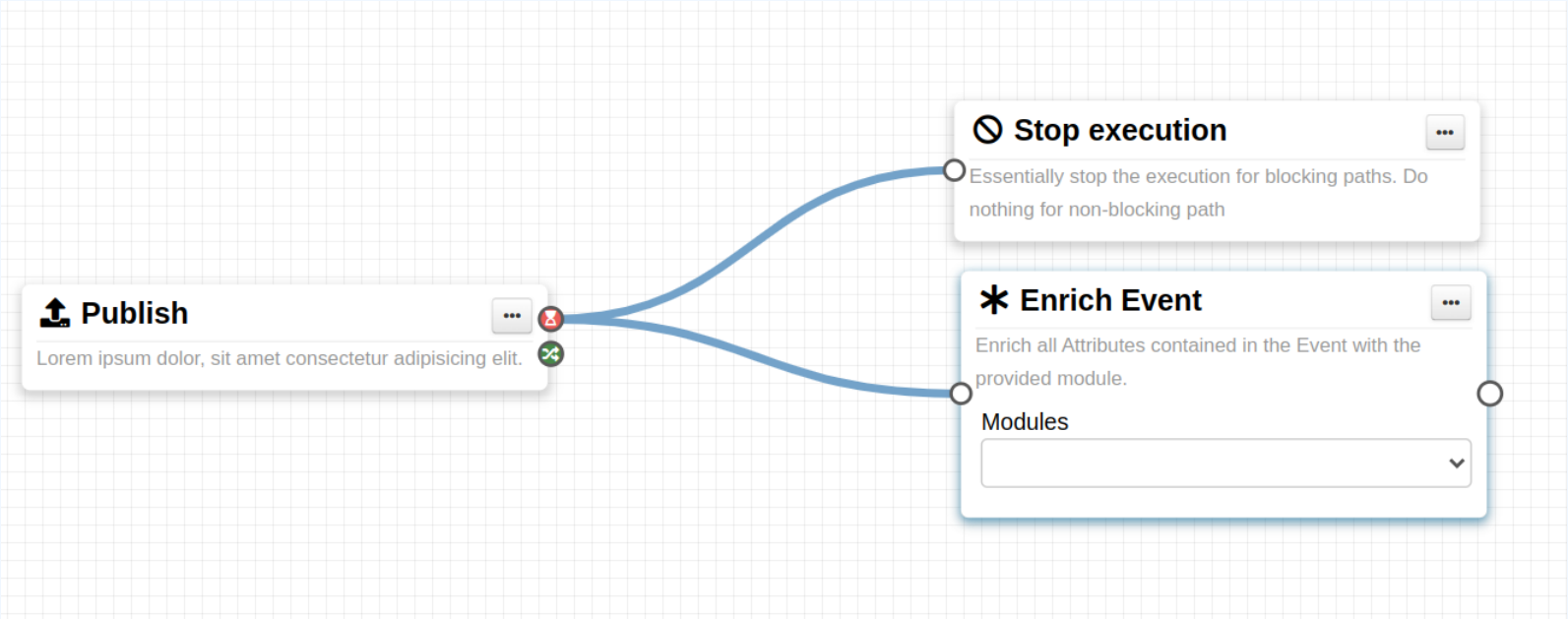
1. Will `Enrich Event` module be executed?
   - **Yes**
   - The blocking path is executed before the non-blocking one
   - The result of the non-blocking path has no influence on the blocking one
2. Will the next blocking path (from another workflow) be executed?
   - **Yes**
   - The blocking path is executed before the non-blocking one
   - The result of the non-blocking path has no influence the execution of other workflows

Workflow example 4



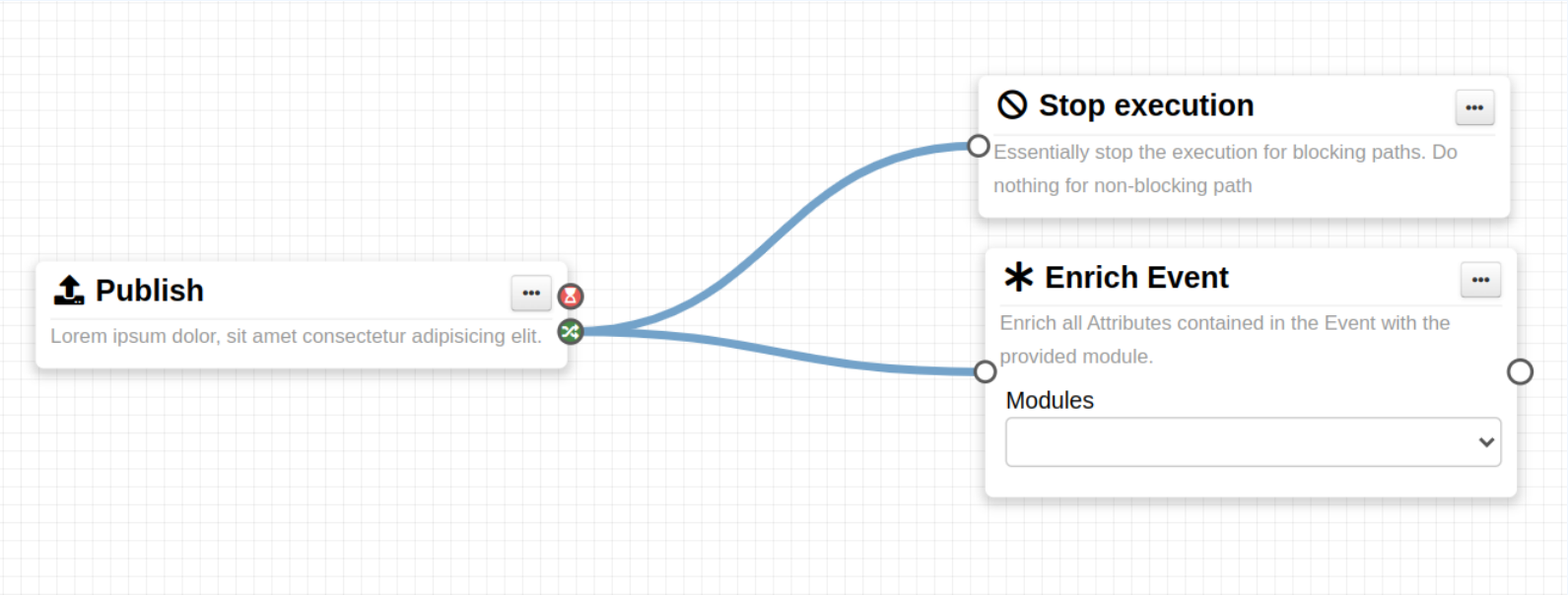
1. Will `Enrich Event` module be executed?

Workflow example 4: Answers



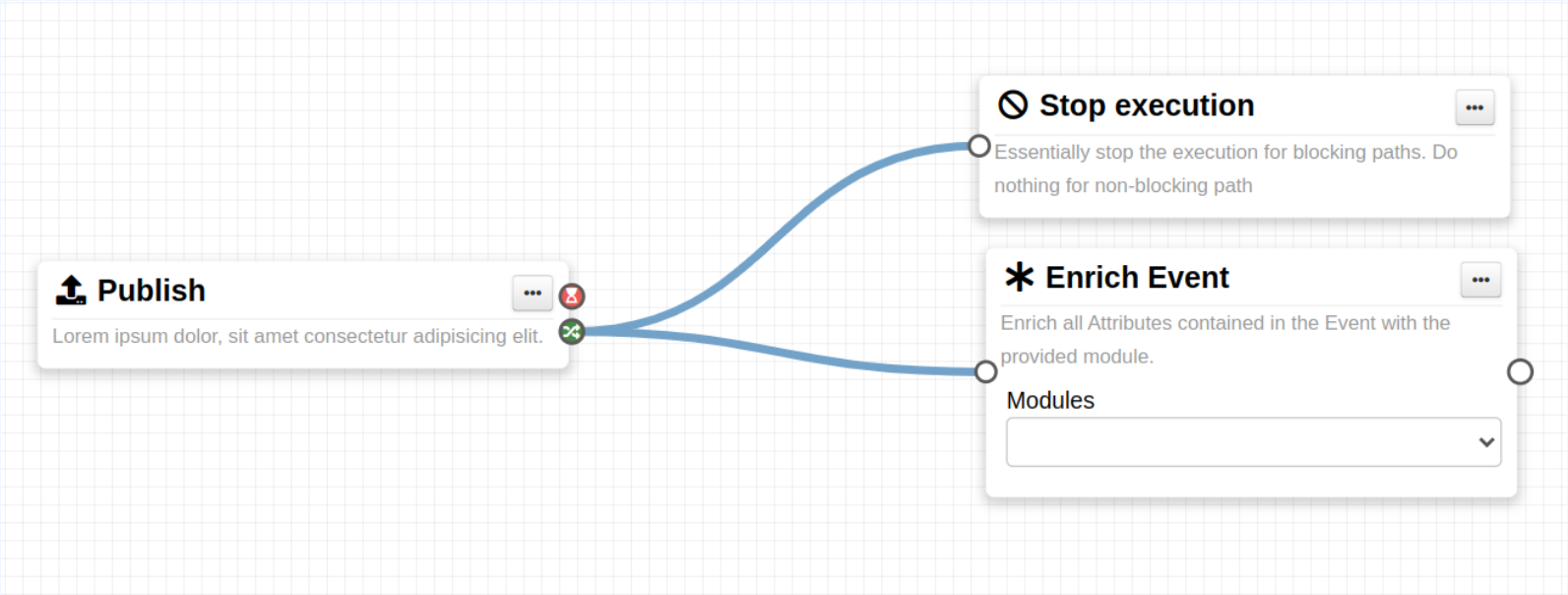
1. Will `Enrich Event` module be executed?
   - **Yes** and **No**. The execution order for the same output is not guaranteed
   - If `Stop execution` is executed first, it's a no.

Workflow example 5



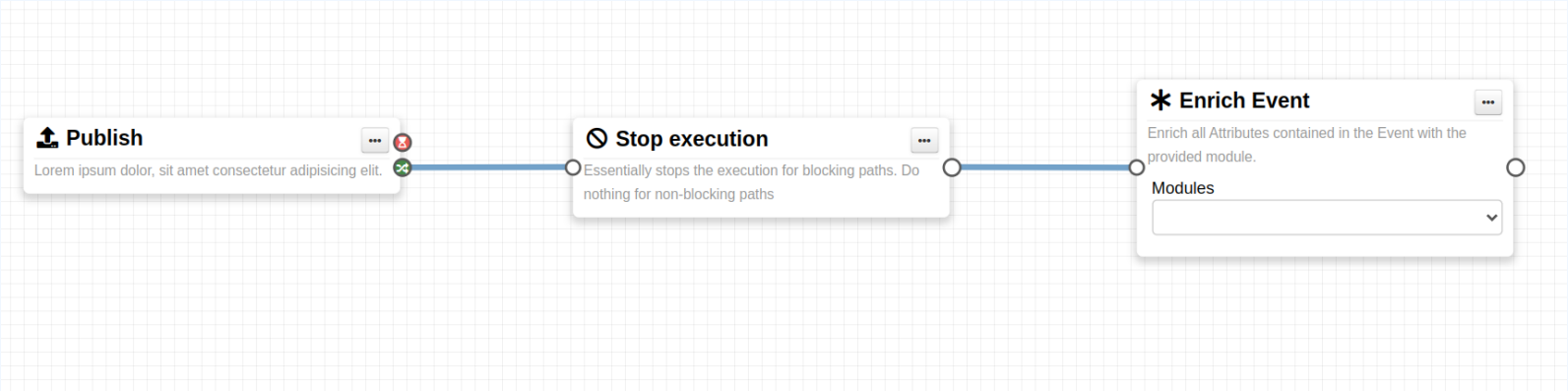
1. Will `Enrich Event` module be executed?

Workflow example 5: Answers



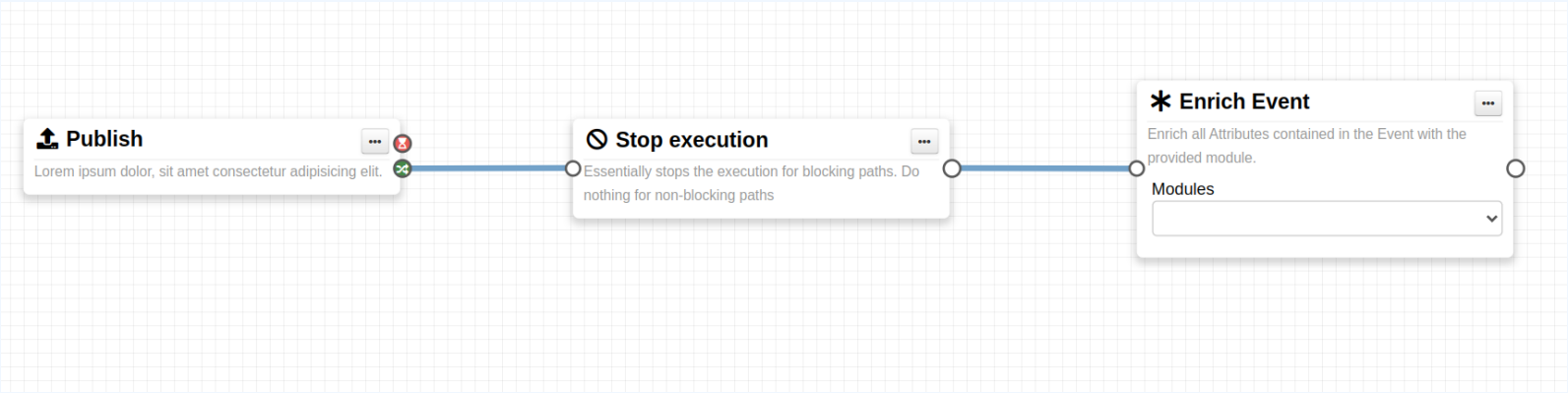
1. Will `Enrich Event` module be executed?
   - **Yes**. The execution order for the same output is not guaranteed
   - However, as we are in a non-blocking path, the outcome of the execution of another path has no impact

Workflow example 6



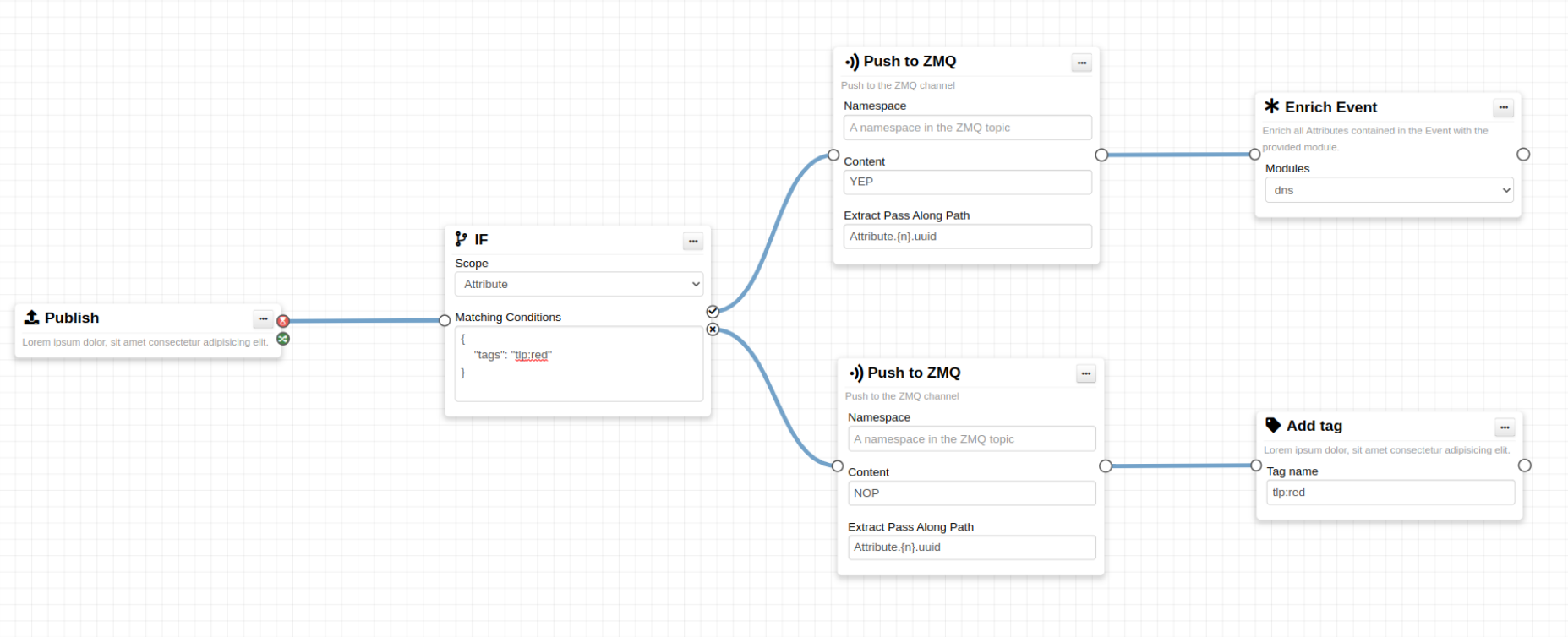
1. Will `Enrich Event` module be executed?

Workflow example 6: Answers



1. Will `Enrich Event` module be executed?
   - **No**. Even if we are in a non-blocking path, if the current execution path is blocked, the execution will be stopped

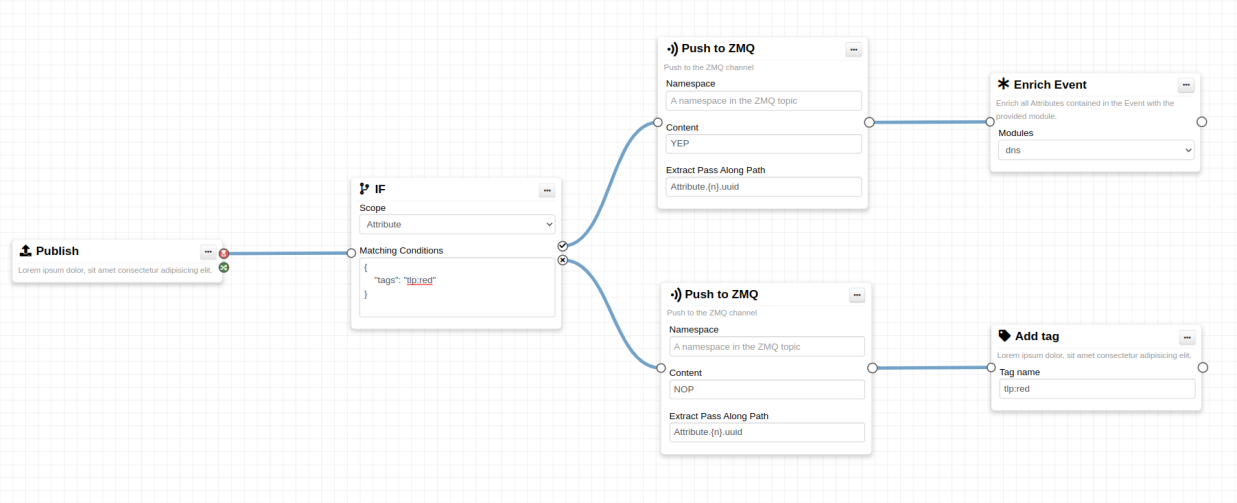
Workflow example 7



Category	Type	Value	Tags
Network activity	ip-src	185.194.93.14	tlp:red x
Network activity	domain	circl.lu	tlp:white x

1. Will `Enrich Event` module be executed?
2. Will `circl.lu` have a tag attached to it?

Workflow example 7: Answers



Category	Type	Value	Tags
Network activity	ip-src	185.194.93.14	tlp:red
Network activity	domain	circl.lu	tlp:white

1. Will `Enrich Event` module be executed?
   - **Yes**. The event contains an attribute satisfying the matching condition
2. Will `circl.lu` have a tag attached to it?
   - **No**. The event contains an attribute satisfying the matching condition. The `else` part will not be executed.