

# Use case: transform application workflow

This set of use cases is presented to address a common need: providing a means to modify the I/O workflow within Fedora, based on conditions set by internal or external services.

By modifying the workflow we mean making an operation fail or succeed, or transforming the data involved in the operation both for storage and retrieval.

## Use cases

1. Perform structural validation on nodes
  - a. User ingests a node by providing some properties and some mixins, or adds or removes a mixin in a node.
  - b. Before persisting the node, the metadata is checked against a configuration file/database to verify that the conditions to apply those mixins (mandatory properties, conflicting/disjointed mixins, etc.) are met.
  - c. If validation conditions are met, the operation proceeds as planned in Fedora; else Fedora throws a validation exception.
2. Perform data validation on properties
  - a. User updates some properties in a node, or creates a node with some properties.
  - b. Property values are checked against a config file/database if data type(s), cardinality, range (both numeric and RDFS), etc. are indicated for that property.
  - c. If validation conditions are met, the operation proceeds as planned in Fedora; else Fedora throws a validation exception.
3. External authorization providers
  - a. User performs an operation on a node.
  - b. Fedora checks ACLs interesting the user identity, the node and the operation involved from an external authorization provider.
  - c. ACLs can be stored in Fedora for convenience, but the external authZ provider facilitates reuse of them for ancillary applications (e.g. external index).
  - d. If ACL conditions are met, Fedora proceeds with the operation; if not, Fedora throws an authorization exception.
4. I/O transformation
  - a. Client creates a session by authenticating via REST API and passing a "myheader\_dept=Imaging" header. This denotes the department the user belongs to.
  - b. Fedora parses the value of "myheader\_dept" header via the [HTTP Header Principal Provider](#).
  - c. Client creates a node during this session.
  - d. Fedora adds the "myns:dept\_created" property with the value "Imaging" (or a URI obtained through an external search) to the new node.
  - e. A similar workflow can be adopted to filter out, transform or enrich response data in a retrieval operation.

## Proposed approach

The above use cases could be solved by exposing some event hooks that allow to attach custom functionality before the node is persisted. By attaching a custom method to a hook, one can have Fedora respond to the response of these methods and transform the application workflow.

## Doubts/Challenges

Which events do we want to trigger? Which points of the workflow do we want to expose as hooks? Do many parts of the code need to be changed if hooks are provided at different points?

Will transactions add complexity to the whole?