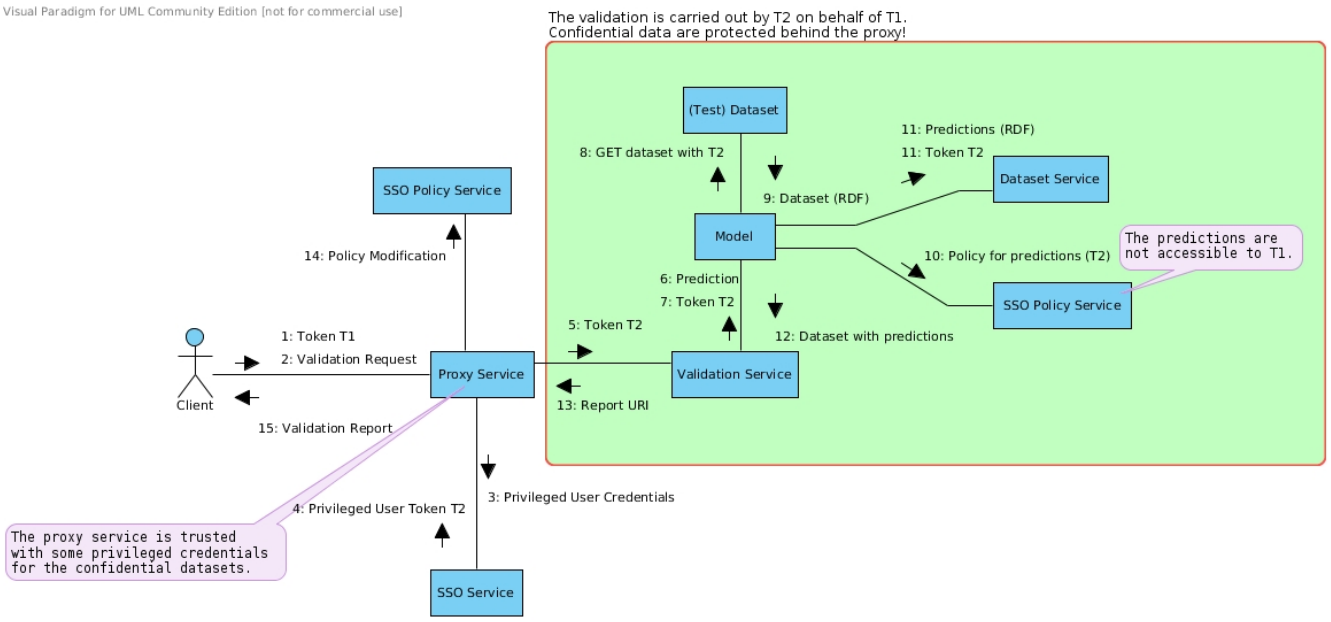
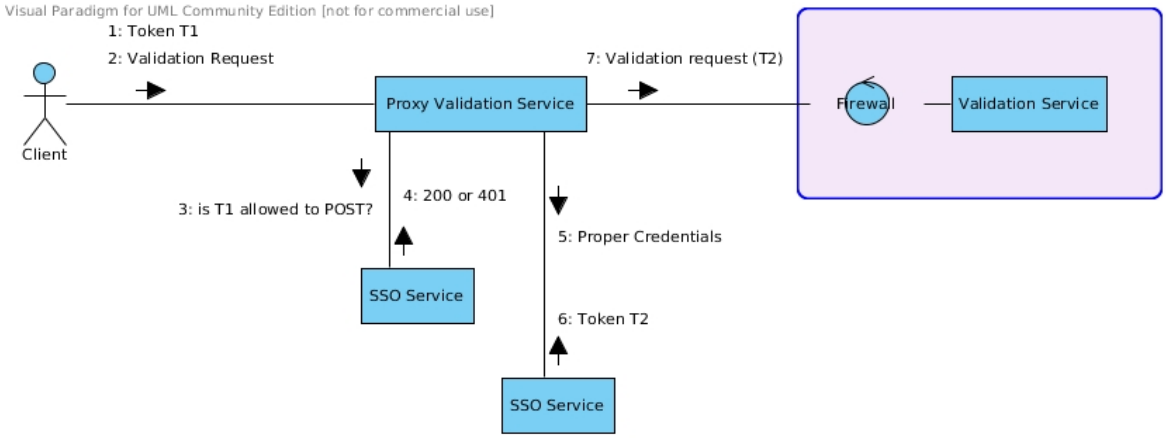
Validation against confidential data

The use case of “validation against confidential data” consist in creation of validation reports for datasets that are not accessible by any means to a user. No prediction data or parts of the initial dataset should be exposed to the user and any intermediate resources related to the test data should be protected in such a way that the end user is denied access. A validation report for the model is created along with a proper policy that allows the end user to access it. Whether the confidential data should be available for validation is to be decided by the creator. If yes, then a *trusted proxy service* is endowed with (GET) privileges on these data, *i.e.* is given privileged credentials. The proxy retains a database relating confidential dataset URIs with credentials to access them. End users address to that service to perform a validation for a dataset to which they don't have access.

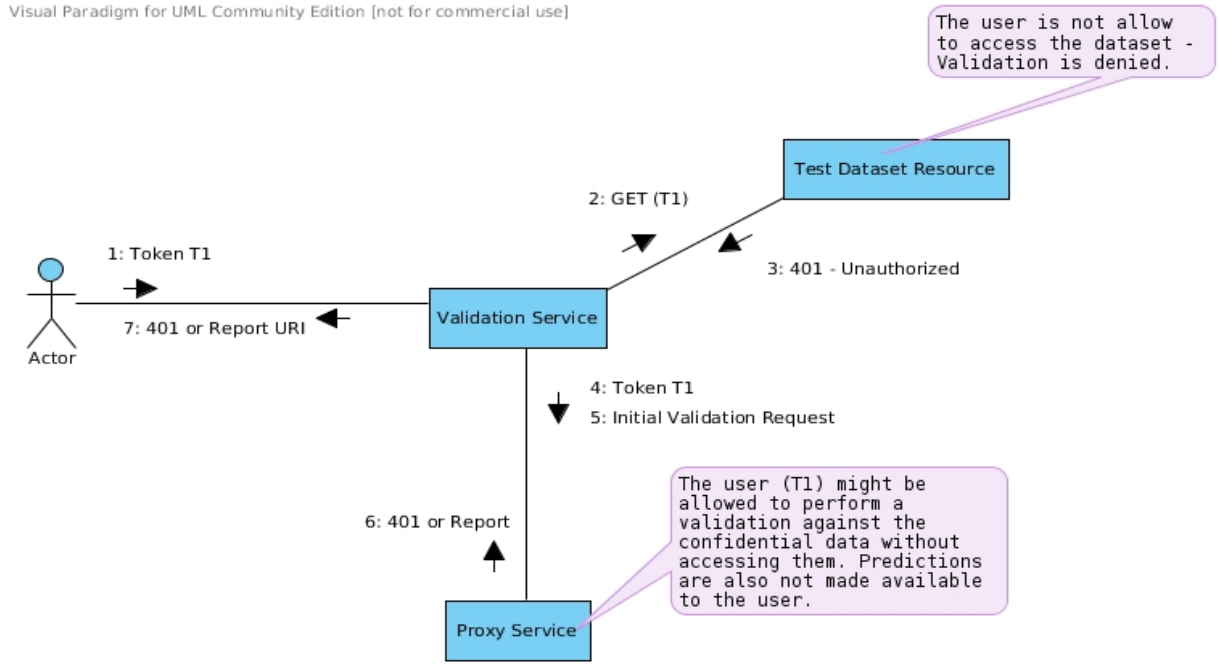


The proposed structure is represented in Figure 1. Being generic enough, can be used to tackle similar problems where the user needs to access parts of the results that only a more privileged user can access or acquire. The end user provides its token (T1) to the proxy service along with a validation request. The client is not actually aware whether it refers to an actual validation service or to a proxy since the request is identical with the difference that the actual service is protected behind the proxy. The proxy service acquires a privileged token T2 that corresponds to the test dataset URI that the user has provided. If the proxy service does not have any stored credentials for the test dataset then it tries to use the token T1 (i.e. in that case T2 = T1 which will be successful only in the case that the end user has access to the test dataset and the model). The proxy service has the option to deny access to the end user specifying some SSO policy for its URI while the validation service may have a different policy itself. This way it is possible to control which users can use the proxy service and/or the validation service directly. Furthermore service providers might consider hiding completely the actual validation service behind a firewall for additional security thus allowing it to be accessible only via the proxy.

In Figure 2 we see that it is possible that the end user is denied access to the proxy service and in the presence of a firewall no validation is possible and the validation service is utterly protected from unwanted requests (even though might not make much sense in denying access to a user to the validation service).



Finally one more alternative is that the validation service can address to the proxy in case it receives a 401 from the test dataset resource acting on behalf of the user (T1).



This scheme has its advantages as it can control on two independent layers the validation of data. Users address to the actual validation service in all cases which takes on the task of accessing the trusted proxy service only if a 401 is received while processing the request on behalf of T1. In case of an initial 401, the validation service attempts to find an authentication token T2 and retry the request. In this case proxy service providers can consider hiding the proxy service behind a firewall.