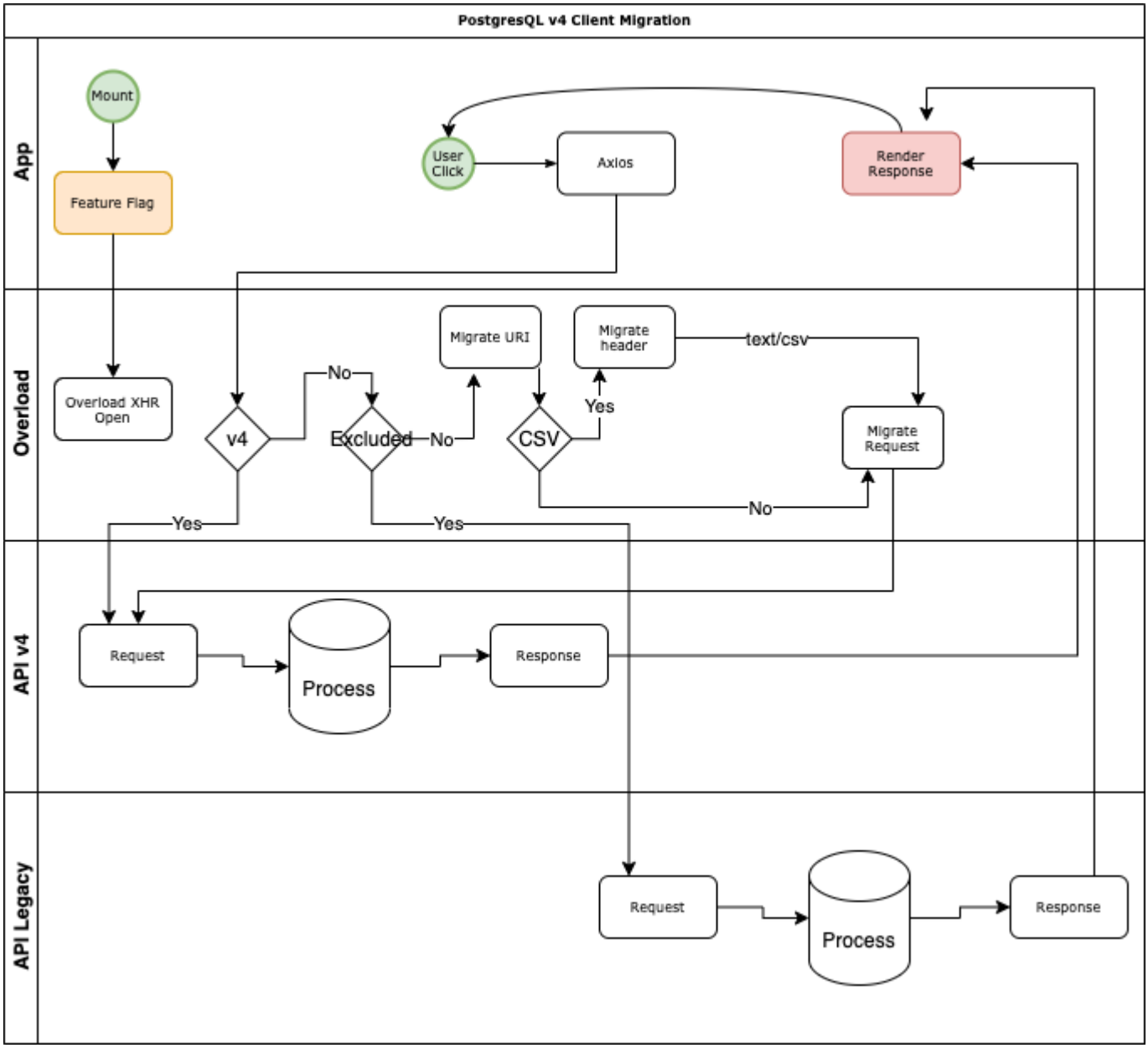
Q:Cyber Client App v4 API Migration | PostgresQL

This document describes a technical approach to migrating from vX to v4 API end-points in the Client App. The most significant constraint is that the migration must exist behind a feature flag configured for each unique environment.

It would be desirable to encapsulate conditional logic to facilitate testing and future clean-up (technical debt). The client contains 41k lines of custom code. Currently 68 lines of code contain a hard-coded reference to v1/v2/v3 API within 29 files. Adding conditional logic to those 29 files would cause short-term quality issues and long-term maintenance challenges.

This proposed solution overloads a browser asynchronous method (XHR). It encapsulates URI conversion, URI exclusion, and API bump-down logic while enforcing the configured feature-flag.



Identified task requirements:

1. Establish a global feature flag: `workflows.ACDP.admon-postgres`
2. Conditionally change the URI to point to the new end-point based on the feature flag
3. Conditionally remove “CSV” from the URI and modify header based on feature flag and existence of “CSV” in the source URI.
4. Swap the position of hostname/obs to obs/hostname for v1 requests
5. Fetch node by ID response will return only the data being displayed instead of the whole blob.

Notes:

It is suggested that we commit to sending dates in raw epoch timestamp format consistently. The human readable format can be transmitted as well. The rationale is that we may have some I18N requirements in the near future that would require client-side manipulation of time/date objects.

The cyber app does not implement the Fetch syntax for asynchronous communication. This is significant because it guarantees that all asynchronous calls will be processed via XHR.