

Authentication Patterns

WIP

Option 1 - Key Vault common core service to support this request.

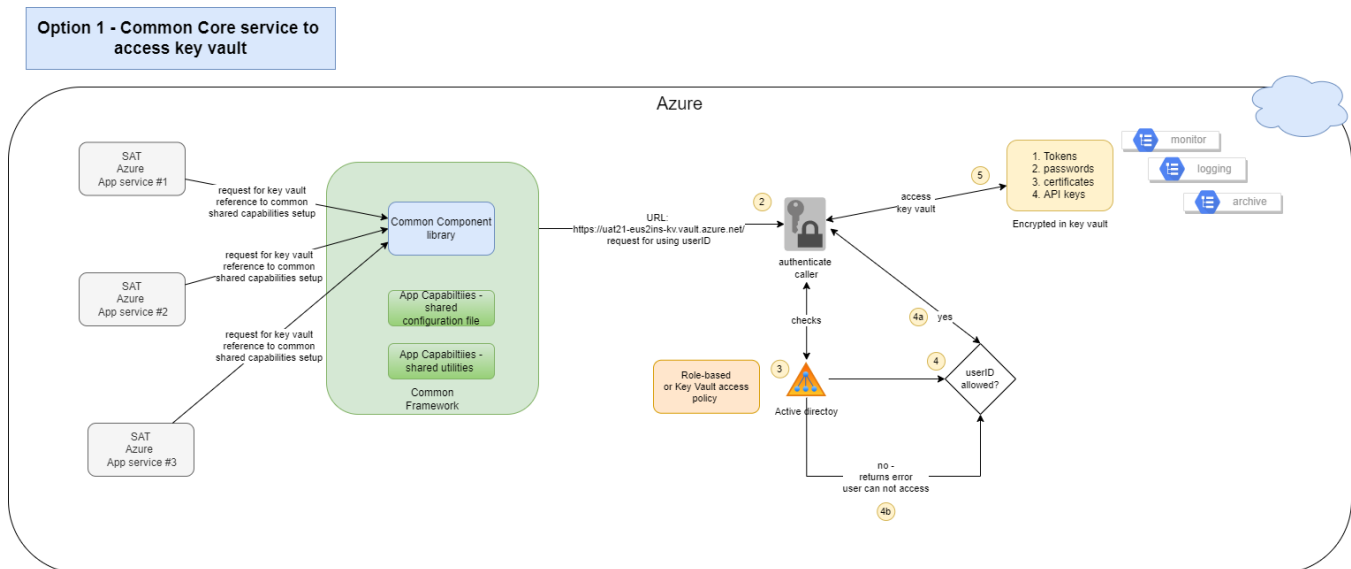
1. **Used for DB connections**
2. **Not used as common pattern for in-stock today**

Pros:

1. Allow the ease of scalability
2. Allow for common shared capability - avoid having each workstream build their own authentication framework.
3. Ensures that all workstream are using the same pattern to support authentication.

Cons:

1. New pattern not implemented anywhere - need more discovery and will take time to implement.



Option 2 - Replicate authentication framework across each project that need to use the Core Authentication service (token request)

1. **Used for service to service connections**
2. **Leverages OAuth once token is obtain via Okta**
3. **Primary pattern used for in-stock application as of current architecture.**

Pros:

1. Already implemented on core side and easy for consumer to build in their project
2. If Okta is down then token is still good for 1 hour
3. Okta SLA for recovery time = 10 mins

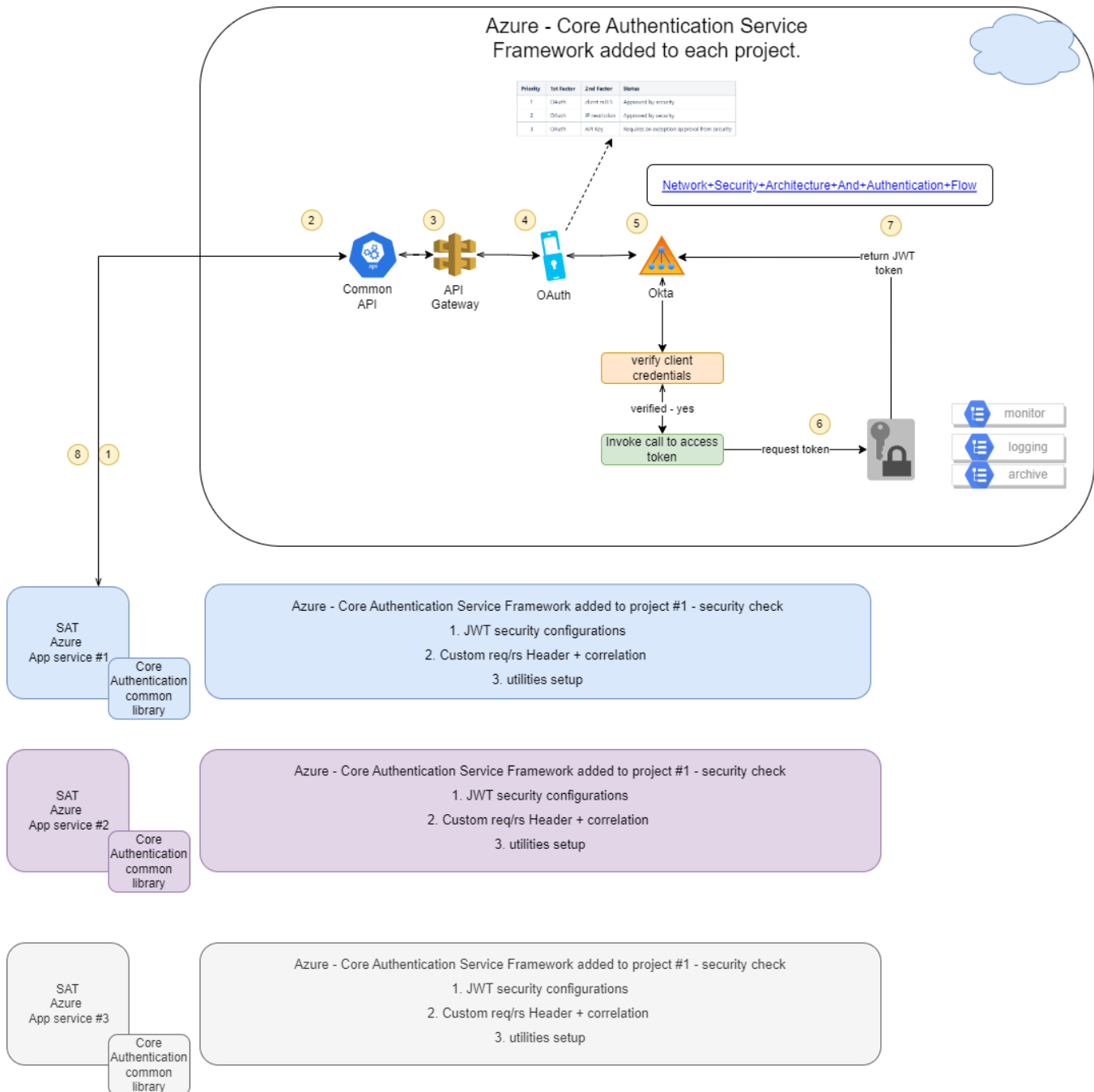
Cons:

1. More setup + maintenance on the capabilities team to rebuild the framework into each project. Changes on the authentication service will also require potential changes on the capabilities team that is calling into the authentication service.

Additional artifacts:

1. SPA replicated core authentication service code:
 - <https://github.com/krogertechnology/sat-shelf-price-audit-server/tree/develop/sat-shelf-price-audit-service/src/main/java/com/kroger/sat/shelfpriceaudit/security>

Option 2 - Replicate Core service Authentication token request framework to each project



Option 3 - Managed Identity setup

- Used for service to service connections

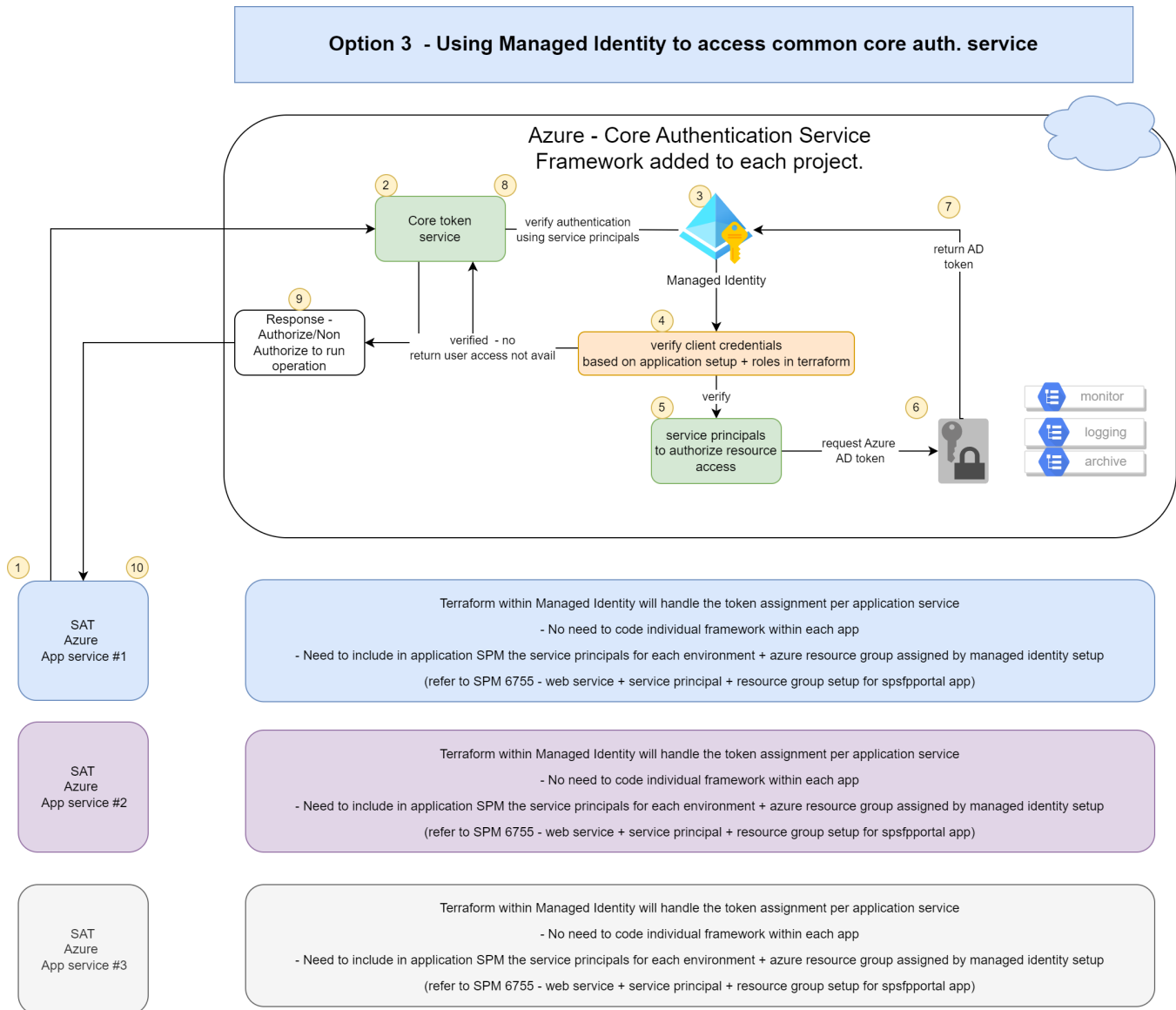
1. In discovery for in-stock

Pros:

1. Configure MI at the application level to prevent accidentals of wrong header info sent
2. Eliminate the need to use common library and risk of managing changes to the library for each of the application that uses it

Cons:

1. Need to verify this does not pose any security issues / risk since it is a new pattern



Option 4 - still in discovery - uses Microsoft DB to verify credentials.

- In POC mode

1. Aaron Rapport + Melissa Woolum is helping with exploring this new pattern

Option 4 - Microsoft DB authentication pattern

