Concern (Identifier: Description)		*Con#4: How does the system handle raw_data?*
Ranking criteria (Identifier: Name)		*Cr#1: memory usage* *Cr#2: performance* *Cr#3: scalability*
Options	***Identifier: Name***	*Con#4-Opt#1: Internal Caching*
	Description	*Given the fact that the data flow from raw_data DBMS to the message stream is not continuous, but rather a flow-on-demand, as in data flows according to the OID from the latest fab_data capture, the flow needs to be discrete and the data needs to be cached before being passed through the message stream. The discritization of the flow and the caching of the data is done using the kafka streams application API (inside the streaming platform).*
	Status	*This option is decided.*
	Relationship(s)	-
	Evaluation	*Cr#1: Memory usage becomes minimal as we make the flow discrete (as opposed to a continuous flow), and the memory of the raw_data topic can be set to recycle with the memory of fab_data topic.* *Cr#2: Aside from simple checks to see which OID needs to be fetched to the raw_data topic, the performance can be evaluated as O(n), with n being the number of fab_data instances being streamed. However, O(n) represents the worst case scenario in which all OIDs are new, but if an OID is in the cache then it doesn't have to be fetched again, which enhances the performance.* *Cr#3: Since the API used for this option is the same as the one used by the bulk of the system, then this option scales up with the system.*
	Rationale of decision	*This option satisfies the requirements, enhances the performance, and scales up with the system.*
	Identifier: Name	*Con#4-Opt#1: External Caching*
	Description	*Given the fact that the data flow from raw_data DBMS to the message stream is not continuous, but rather a flow-on-demand, as in data flows according to the OID from the latest fab_data capture, the flow needs to be discrete and the data needs to be cached before being passed to the message stream. The discritization of the flow and the caching of the data is done by providing an external application (outside the streaming platform) that takes the latest OID from the message stream, then fetches the recipe from raw_data DBMS then provides a dataflow to the raw_data topic.*
	Status	*This option is rejected.*
	Relationship(s)	-
	Evaluation	*Cr#1: Memory usage increases as we need to route the OID to the external caching application, then the provided recipes will be consumed cintinuously by raw_data topic, which increases the load inside the message stream.* *Cr#2: Performance decreases as we route the OID to the external caching application.* *Cr#3: Since we are using an external application for this option, then it requires special maintainance and specific development cycles to guarantee its scalability.*
	Rationale of decision	*The option is rejected because it impacts performance negatively, increases memory usage, and doesn't scale up with the system.*