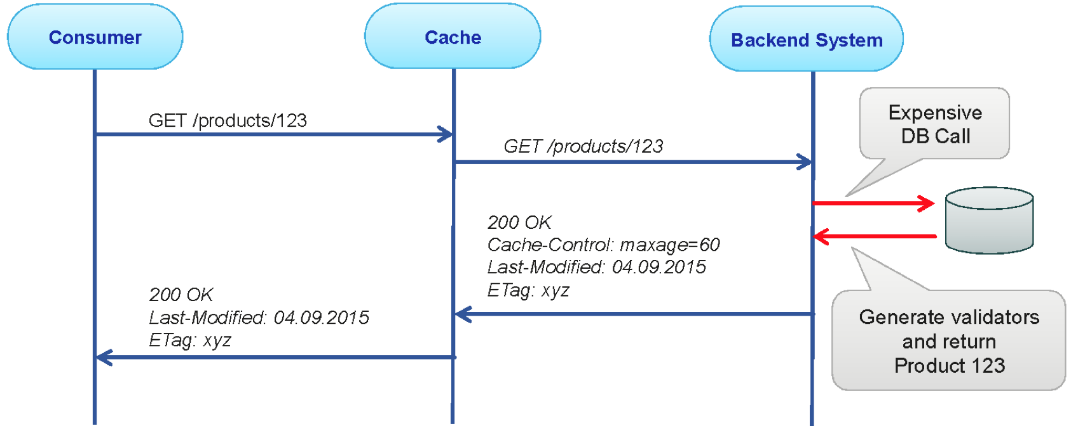
# Caching

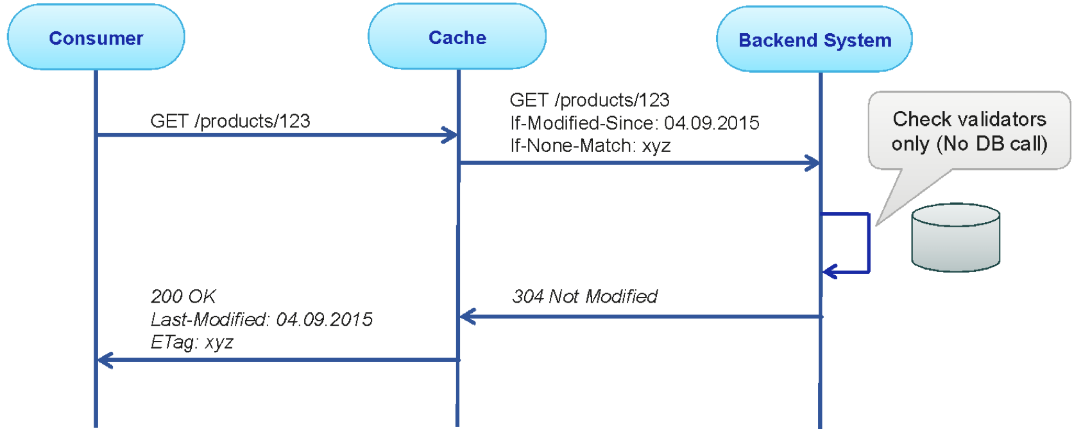
The Caching Pattern has been introduced to improve the performance and load management of the API. In many cases the information requested from the API is fairly static, with limited change over time (typical examples: address information on locations, price information, …). In these cases the API does not necessarily have to retrieve the information from back-end systems, but keep it in memory close to the API, called a Cache. Only if the information is not available in the Cache, the integration retrieves the information from the back-end. Based on the frequency of the retrieval and the size of the Cache, this information from the back-end could be added. The following charts describe an example of the caching pattern:

The Consumer requests information on the product 123 from the product API. The Cache does not have valid information on the product in the store and sends the request to the back-end system which retrieves the information.

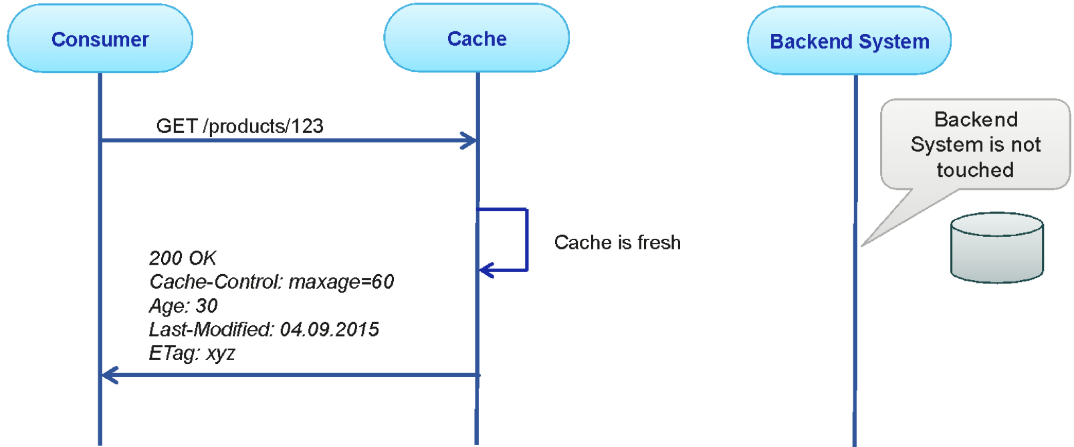
As this is the first request of the information the Cache does not keep the information as frequently used in local memory, but returns it back to the consumer.



The second request for the product now triggers the cache to inspect the information retrieved. As the information has not changed it is likely to be static. Also the information has an expiration time added, so the Cache can keep the information local for any following request.



Any following request will now be handled by the Cache unless the information is expired in the Cache or (because of changes in the back-end system) removed from the Cache.



It is important that the Cache is refreshed (i.e. deleted) if the back-end is changed, or the maximal Cache storage period (age of the information) exceeds the allowed age of the information.