**Search Item using elastic search and in-memory cache**

**Status**

Proposed

**Context**

* Enable quick search facility
* Most frequent search items are in an in-memory cache

**Decision**

The proposed system would use Elastic search to support text search. The data retuned based on the search would be persisted in a distributed cache. The most frequently used search results would be persisted in a distributed local cache. Searchable text would be updated by subscribing to a message broker topic upon product updating.

Apache Kafka can be used as message broker and Apache Ignite would use as the distributed cache mechanism.

Diagram

Description automatically generated

Architecture characteristics that required:

* Performance
* Scalability
* Responsiveness

Alternatives :

1. Use simple message queue instead of message broker
2. User regular DB to persist searchable text
3. Use REST API to perform search and return result

**Consequences**

Architecture tradeoff analysis

* Using Message broker instead of message queue
  + Ease of distributivity
  + Send updates as events
  + Multiple subscribers
  + Less probability of message loss
* Usage of distributed near cache hybrid
  + This enables fast access to frequently searched items (MFU) and enables large number of cached items (full cache) as well.
* Usage of Elastic search
  + This enables Index lifecycle management with high performance text search

Architecture fitness function

* A search result should be returned to the client within 2 seconds in a normal network conditions.