Memoization

In general, Memoization function cache the original function's execution result until the define timeout. Caches are mapped combination of cache key and value (execution result) pair. Cache key determines from first argument of memorized function in absence of resolver. If resolver provided and it has same set of parameters of original function then cache key receives from resolver. When resolver present but not valid then nothing will cache. If function execution exceeded timeout then original function executed again and cache the results for the given timeout.

Scenarios (When Memoization function initialize):

- Resolver and Timeout present:
    - Validated resolver before fetch cache key from resolver.
        - Resolver valid if it's a function and has same set of parameters as original function.
    - If resolver is valid then invokes resolver with same arguments passed to original function and gets cache key from resolver
    - If resolver is invalid then nothing will cache as cache key only detect from resolver if it presents.
    - If valid timeout and value found in cache based on cache key then return the result from cache otherwise execute the original function.
    - Cache (only if cache key receives from resolver and timeout set at least 1 ms) the results and timeout otherwise function result return without store in cache.
    - When timeout exceeds original function executes and cache the result and timeout again.
        - Timeout calculated upon the given timeout and System current time.
- Resolver absent and Timeout present:
    - In case of resolver absent, cache key detects from first argument of memorized function.
    - If valid timeout and value found in cache based on cache key then return the result from cache otherwise execute the original function.
    - Cache (only if timeout set at least 1 ms) the function execution result and timeout until the given timeout.
    - When timeout exceeds original function executes and cache the result and timeout again.
        - Timeout calculated upon the given timeout and System current time.

- Resolver present and Timeout absent:
    - Original function always executed and nothing will cache.
- Resolver and Timeout absent:
    - Original function always executed and nothing will cache.