**Caching Strategy Proposal**

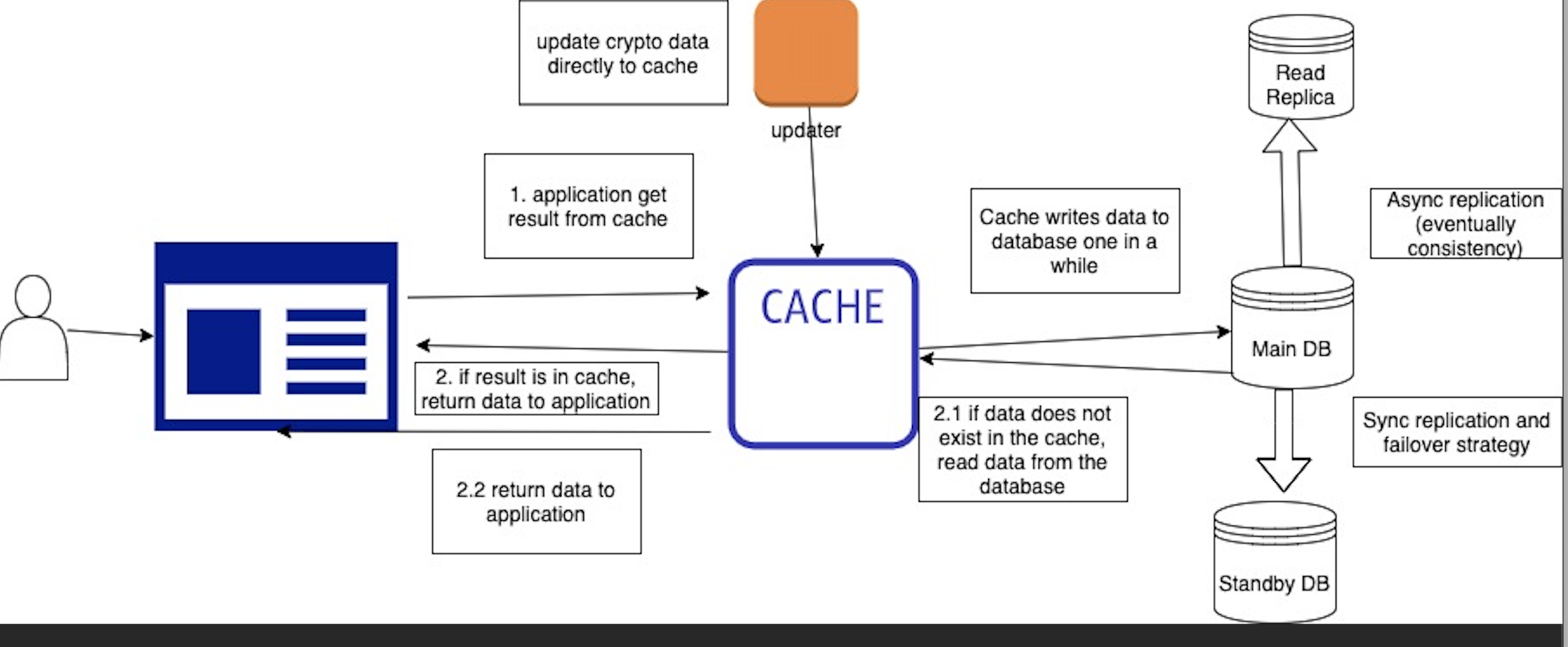
**Assumption**

The original database needs to keep updating the data frequently to get most accurate and real-time currency data.

Strategy

The strategy is based on the combination of read through and write behand caching strategies. The cache, as a main access node, is responsible for updating to database regularly and responding to the application.

The database architecture could be one main database, one standby database and several read replica databases for better performance. In addition, load balancing and auto-scaling strategy can be fulfilled using AWS services. Relational Database Service (RDS) provided by AWS can perform such database architecture. In such database design, the eventually consistency is expected.



1. Updater regularly feeds cache the latest data, and cache updates the database in a short period time. It can be based on how often the updater pushes the data to the cache.
2. The application requests the data, and cache responses directly if cache has the data. Otherwise, cache will read from the database, cache it and response to the application.

Other consideration

1. Due to the high volume loaded into the cache, the Redis is preferred in favor of node failure and auto-failover, because it provides backup and restore features compared with Memcached. In AWS services, we can configure the cache and TTL (time-to-live) using ElastiCache.