CLAIMS

What is claimed is:

1. A data synchronization method, comprising:

receiving, by an application, a payment service request;

determining, by ~~an~~the application, to-be-queried user account data~~:~~

~~obtaining a version identifier corresponding to the to-be-queried data from a cache; and~~

~~when it is determined that~~ of a user based upon the ~~version identifier corresponding to the to-be-queried data is inconsistent with a specified version identifier, querying the to-be-queried data from a database, and replacing the to-be-queried data stored in the cache with the to-be-queried data obtained from the database.~~payment service request;

~~2. The method according to claim 1, wherein the method further comprises: when it is determined that the version identifier corresponding to the to-be-queried data is consistent with the specified version identifier, obtaining the to-be-queried data from the cache.~~

~~3. The method according to claim 1, wherein determining, by an~~ identifying, using the application, ~~to-be-queried data specifically comprises:~~

~~determining~~ a data identifier ~~corresponding to the~~associated with to-be-queried user account data; ~~and~~

~~obtaining a~~identifying, using the application and the data identifier, cached user account data in a cache associated with the data identifier and a first version identifier corresponding to the ~~to-be-queried data~~cached user account data in the cache associated with the data identifier, wherein the first version identifier includes time information of a change from a ~~cache specifically comprises:~~

~~obtaining to-be-queried data~~ previous version identifier corresponding to the ~~data identifier from the cache based on the determined data identifier; and~~

~~obtaining a~~cached user account data to the first version identifier corresponding to the ~~to-be-queried data from the to-be-queried data.~~cached user account data;

~~4. The method according to claim 1, wherein determining, by an application, to-be-queried data specifically comprises:~~

~~determining a data identifier corresponding to the to-be-queried data; and~~

~~obtaining a~~identifying a current version identifier of the application, wherein the

application is associated with a database storing a plurality of pieces of database data, wherein each piece of the plurality of pieces of database data is associated with a version identifier that is the same as the current version identifier of the application; and

in response to determining that the first version identifier corresponding to the ~~to-be-queried data from a cache specifically comprises~~cached user account data in the cache associated with the data identifier does not match the current version identifier of the application:

~~determining an actual identifier that matches the data identifier from the cache; and~~

~~obtaining the version identifier corresponding to the to-be-queried data from the actual identifier.~~

~~5. The method according to claim 1, wherein replacing the to-be-queried data stored in the cache with the to-be-queried data obtained from the database specifically comprises:~~

~~invalidating the to-be-queried data stored in the cache; and~~

~~storing the to-be-queried data obtained from the~~obtaining, using the application, database ~~into the cache.~~

~~6. The method according to any one of~~ data from the ~~1 to 5, wherein the method further comprises:~~

~~in response to detecting a change to version information of the application, determining the specified version identifier based on changed version information.~~

~~7. A data synchronization apparatus, comprising:~~

~~a determining module, configured to determine to-be-queried data;~~

~~an acquisition module, configured to obtain a version identifier corresponding to the to-be-queried data from a cache; and~~

~~a processing module, configured to: when it is determined that the version identifier corresponding to the to-be-queried data is inconsistent with a specified version identifier, query the to-be-queried data from a~~ database~~, and replace the to-be-queried data stored in the cache with the to-be-queried data obtained from the database.~~ associated with the data identifier; and

~~8~~replacing, using the application, the cached user account data in the cache associated with the data identifier with the database data from the database associated with the data identifier.

2. The ~~apparatus~~method according to claim ~~7, wherein the processing module is~~

configured1, further comprising:

in response to: when it is determined determining that the first version identifier corresponding to the to be queriedcached user account data is consistentin the cache associated with the specifieddata identifier matches the current version identifier, obtain of the to be queriedapplication, obtaining, using the application, the cached user account data fromin the cache associated with the data identifier.

93. The apparatusmethod according to claim 71, wherein identifying the first version identifier corresponding to the cached user account data in the cache associated with the data identifier comprises:

obtaining, based on the data identifier, the cached user account data in the cache associated with the data identifier; and

identifying, from the cached user account data in the cache associated with the data identifier, the first version identifier.

4. The method according to claim 1, wherein the determining module determines a dataidentifying the first version identifier corresponding to the to-be-queried data; andcached user account data in the cache associated with the data identifier comprises:

the acquisition module obtains to-be-queried data corresponding to the data identifier from the cache based on the determined data identifier; and obtains a version identifier corresponding to the to-be-queried data from the to-be-queried data.identifying, in the cache,

10. The apparatus according to claim 7, wherein the determining module determines a data identifier corresponding to the to-be-queried data; and

the acquisition module determines an actual identifier that matches the data identifier from the cache; and obtains the version identifier corresponding to the to-be-queried data from the actual identifier.; and

11. identifying, from the actual identifier, the first version identifier.

5. The apparatusmethod according to claim 71, wherein the processing module invalidates the to-be-queried replacing the cached user account data stored in the cache; and stores associated with the to-be-queried data obtained fromidentifier with the database into the cache.data from the database associated with the data identifier comprises:

12invalidating the cached user account data in the cache associated with the data identifier;

and

storing, in the cache, the database data from the database associated with the data identifier.

6. The ~~apparatus~~method according to ~~any one of claims 7 to 11~~claim 1, further comprising:

in response to detecting a change to version information of the application, determining the current version identifier of the application based on changed version information.

7. The method according to claim 1, wherein the ~~apparatus further comprises:~~current version identifier of the application is determined based on current version information of the application.

8. A non-transitory, computer-readable medium storing one or more instructions executable by a computer system to perform operations comprising:

receiving, by an application, a payment service request;

determining, by the application, to-be-queried user account data of a user based upon the payment service request;

identifying, using the application, a data identifier associated with to-be-queried user account data;

identifying, using the application and the data identifier, cached user account data in a cache associated with the data identifier and a first version identifier corresponding to the cached user account data in the cache associated with the data identifier, wherein the first version identifier includes time information of a change from a previous version identifier corresponding to the cached user account data to the first version identifier corresponding to the cached user account data;

identifying a current version identifier of the application, wherein the application is associated with a database storing a plurality of pieces of database data, wherein each piece of the plurality of pieces of database data is associated with a version identifier that is the same as the current version identifier of the application; and

in response to determining that the first version identifier corresponding to the cached user account data in the cache associated with the data identifier does not match the current version identifier of the application:

obtaining, using the application, database data from the database associated with

the data identifier; and

replacing, using the application, the cached user account data in the cache associated with the data identifier with the database data from the database associated with the data identifier.

9. The non-transitory, computer-readable medium according to claim 8, wherein the operations comprise:

in response to determining that the first version identifier corresponding to the cached user account data in the cache associated with the data identifier matches the current version identifier of the application, obtaining, using the application, the cached user account data in the cache associated with the data identifier.

10. ~~a monitoring module, configured to:~~ The non-transitory, computer-readable medium according to claim 8, wherein identifying the first version identifier corresponding to the cached user account data in the cache associated with the data identifier comprises:

obtaining, based on the data identifier, the cached user account data in the cache associated with the data identifier; and

identifying, from the cached user account data in the cache associated with the data identifier, the first version identifier.

11. The non-transitory, computer-readable medium according to claim 8, wherein identifying the first version identifier corresponding to the cached user account data in the cache associated with the data identifier comprises:

identifying, in the cache, an actual identifier that matches the data identifier; and

identifying, from the actual identifier, the first version identifier.

12. The non-transitory, computer-readable medium according to claim 8, wherein replacing the cached user account data in the cache associated with the data identifier with the database data from the database associated with the data identifier comprises:

invalidating the cached user account data in the cache associated with the data identifier; and

storing, in the cache, the database data from the database associated with the data identifier.

13. The non-transitory, computer-readable medium according to claim 8, wherein the operations comprise:

in response to detecting a change to version information of the ~~apparatus, determine~~application, determining the ~~specified~~current version identifier of the application based on changed version information.

~~13.~~

14. The non-transitory, computer-readable medium according to claim 8, wherein the current version identifier of the application is determined based on current version information of the application.

15. A ~~data synchronization device~~computer-implemented system, comprising~~-~~:

one or more ~~processors~~computers; and ~~a~~

one or more computer memory~~, wherein the memory stores a program,~~ devices interoperably coupled with the one or more computers and ~~the program is configured to be used~~having tangible, non-transitory, machine-readable media storing one or more instructions that, when executed by the one or more ~~processors to~~computers, perform ~~the following steps~~one or more operations comprising:

receiving, by an application, a payment service request;

determining, by the application, to-be-queried user account data of a user based upon the payment service request;

identifying, using the application, a data identifier associated with to-be-queried user account data;

identifying, using the application and the data identifier, cached user account data in a cache associated with the data identifier and a first version identifier corresponding to the cached user account data in the cache associated with the data identifier, wherein the first version identifier includes time information of a change from a previous version identifier corresponding to the cached user account data to the first version identifier corresponding to the cached user account data;

identifying a current version identifier of the application, wherein the application is associated with a database storing a plurality of pieces of database data, wherein each piece of the plurality of pieces of database data is associated with a version identifier that is the

same as the current version identifier of the application; and

in response to determining that the first version identifier corresponding to the cached user account data in the cache associated with the data identifier does not match the current version identifier of the application:

obtaining, using the application, database data from the database associated with the data identifier; and

replacing, using the application, the cached user account data in the cache associated with the data identifier with the database data from the database associated with the data identifier.

16. The computer-implemented system according to claim 15, wherein the operations comprise:

in response to determining ~~to be queried data;~~

~~obtaining a~~that the first version identifier corresponding to the ~~to be queried data from a cache; and~~cached user account data in the cache associated with the data identifier matches the current version identifier of the application, obtaining, using the application, the cached user account data in the cache associated with the data identifier.

~~when it is determined that the~~

17. The computer-implemented system according to claim 15, wherein identifying the first version identifier corresponding to the ~~to be queried~~cached user account data ~~is inconsistent~~in the cache associated with ~~a specified~~the data identifier comprises:

obtaining, based on the data identifier, the cached user account data in the cache associated with the data identifier; and

identifying, from the cached user account data in the cache associated with the data identifier, the first version identifier~~, querying the to be queried data from a database, and~~.

18. The computer-implemented system according to claim 15, wherein identifying the first version identifier corresponding to the cached user account data in the cache associated with the data identifier comprises:

identifying, in the cache, an actual identifier that matches the data identifier; and

identifying, from the actual identifier, the first version identifier.

19. The computer-implemented system according to claim 15, wherein replacing the ~~to-be-queried~~cached user account data ~~stored~~ in the cache associated with the ~~to-be-queried~~ data ~~obtained~~identifier with the database data from the database~~.~~ associated with the data identifier comprises:

invalidating the cached user account data in the cache associated with the data identifier; and

storing, in the cache, the database data from the database associated with the data identifier.

20. The computer-implemented system according to claim 15, wherein the operations comprise:

in response to detecting a change to version information of the application, determining the current version identifier of the application based on changed version information.