CLAIMS

1. A method for synchronizing message data between at least two clients, the method comprising:

receiving a request to initiate a shared session from a first client, the request including a first client identifier;

based on the received request, providing a session identifier to the first client associated with the first client identifier;

associating the first client identifier with the session identifier;

receiving a request from a second client to be associated with the session identifier;

associating the second client identifier with the session identifier;

receiving message data from the first client;

adding the message data to a queue associated with the session identifier;

adding the message data together with the session identifier to a database;

providing the message data from the queue to the second client based on the session identifier;

receiving, at a load balancer, the message data from the first client;

determining, at the load balancer, an instance of a websocket server based on an availability of resources;

sending the message data to the determined instance of the websocket server; and

providing, from the websocket server, the message data to a websocket storage, the websocket storage including the queue.

2. The method of claim 1, further comprising:

receiving, at the load balancer, the message data from the websocket storage; and

providing the message data from a queue of the websocket storage to the second client.

3. The method of claim 1, further comprising:

receiving, at the load balancer, the message data from the first client;

determining, at the load balancer, an instance of the websocket server based on an availability of resources;

sending the message data to the determined instance of the websocket server; and

providing, from the websocket server, the message data to a content service cluster, the content service cluster storing the message data in the database.

4. The method of claim 1, wherein the message data includes object information and session identifier information, the object information being stored in a location of the database that is different than the session identifier.

5. The method of claim 4, wherein the object information comprises a graphical object displayed at a display associated with the first client for an augmented reality session.

6. The method of claim 1, wherein the message data and the session identifier are stored in an in-memory database.

7. A content services platform comprising:

a processor; and

a memory storing instructions that, when executed by the processor, configure the content services platform to:

receive a request to initiate a shared session from a first client, the request including a first client identifier;

based on the received request, provide a session identifier to the first client associated with the first client identifier;

associate the first client identifier with the session identifier;

receive a request from a second client to be associated with the session identifier;

associate the second client identifier with the session identifier;

receive message data from the first client;

add the message data to a queue associated with the session identifier;

add the message data together with the session identifier to a database;

provide the message data from the queue to the second client based on the session identifier;

receive, at a load balancer, the message data from the first client;

determine, at the load balancer, an instance of a websocket server based on an availability of resources;

send the message data to the determined instance of the websocket server; and

provide, from the websocket server, the message data to a websocket storage, the websocket storage including the queue.

8. The content services platform of claim 7, wherein the instructions further configure the content services platform to:

receive, at the load balancer, the message data from the websocket storage; and

provide the message data from a queue of the websocket storage to the second client.

9. The content services platform of claim 7, wherein the instructions further configure the content services platform to:

receive, at the load balancer, the message data from the first client;

determine, at the load balancer, an instance of the websocket server based on an availability of resources;

send the message data to the determined instance of the websocket server; and

provide, from the websocket server, the message data to a content service cluster, the content service cluster storing the message data in the database.

10. The content services platform of claim 9, wherein the message data includes object information and session identifier information, the object information being stored in a location of the database that is different than the session identifier.

11. The content services platform of claim 10, wherein the object information comprises a graphical object displayed at a display associated with the first client for an augmented reality session.

12. The content services platform of claim 7, wherein the message data and the session identifier are stored in an in-memory database.

13. A non-transitory computer-readable storage medium, the computer-readable storage medium including instructions that when executed by a computer, cause the computer to:

receive a request to initiate a shared session from a first client, the request including a first client identifier;

based on the received request, provide a session identifier to the first client associated

with the first client identifier;

associate the first client identifier with the session identifier;

receive a request from a second client to be associated with the session identifier;

associate the second client identifier with the session identifier;

receive message data from the first client;

add the message data to a queue associated with the session identifier;

add the message data together with the session identifier to a database;

provide the message data from the queue to the second client based on the session identifier;

receive, at a load balancer, the message data from the first client;

determine, at the load balancer, an instance of a websocket server based on an availability of resources;

send the message data to the determined instance of the websocket server; and

provide, from the websocket server, the message data to a websocket storage, the websocket storage including the queue.

14. The computer-readable storage medium of claim 13, wherein the instructions further configure the computer to:

receive, at the load balancer, the message data from the websocket storage; and

provide the message data from a queue of the websocket storage to the second client.

15. The computer-readable storage medium of claim 13, wherein the instructions further configure the computer to:

receive, at the load balancer, the message data from the first client;

determine, at the load balancer, an instance of the websocket server based on an availability of resources;

send the message data to the determined instance of the websocket server; and

provide, from the websocket server, the message data to a content service cluster, the content service cluster storing the message data in the database.

16. The computer-readable storage medium of claim 15, wherein the message data includes object information and session identifier information, the object information being stored in a location of the database that is different than the session identifier.

17. The computer-readable storage medium of claim 13, wherein the message data and the session identifier are stored in an in-memory database.