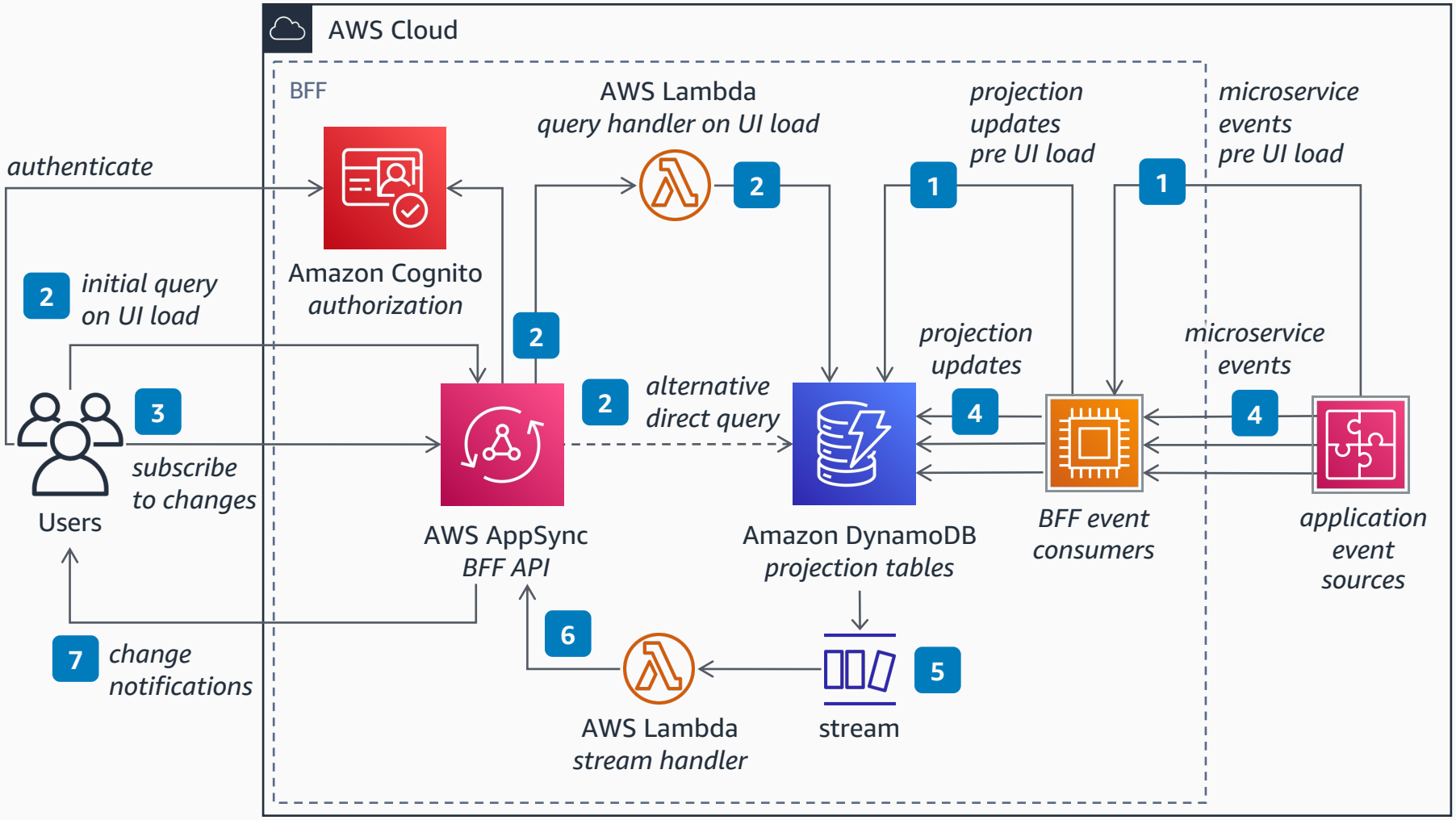
Backend for frontend using AppSync

Improve customer experience on your User Interfaces (UI) by providing real-time visual updates when your microservices raise events about mutations in domain aggregates. This architecture describes how frontend client applications can apply the Backend for Frontend (BFF) pattern to load UI-ready data projections and refresh the UI with event-driven notifications.



1. Catch the events from your application with purpose-built BFF event consumers. These are responsible for keeping a denormalized view of data in **Amazon DynamoDB** for frontend consumption.
2. On UI load, frontend clients authenticate with **Amazon Cognito**, then query the data with GraphQL by invoking the BFF API built with **AWS AppSync.** The data is then fetched in DynamoDB, either directly by **AWS AppSync** or via a BFF query handler built with **AWS Lambda**.
3. Frontend clients subscribe for any subsequent data changes using **AWS AppSync** subscriptions over WebSockets.
4. Continue to consume and process all relevant events from your application using the BFF event consumers. These consumers continuously update the denormalized frontend data view in the BFF database in real-time.
5. Subscribe to all events resulting from data changes in the BFF database using **Amazon DynamoDB Streams**, then register a trigger in **AWS Lambda** to asynchronously invoke a BFF stream-handler **Lambda** function when it detects new stream records.
6. Your BFF stream handler then invokes an empty mutation on the **AWS AppSync** GraphQL schema, purposely created to force the subscription to be triggered, thus sending a notification to clients.
7. When the change notification from **AWS AppSync** is received by the frontend clients, they can refresh the UI content.

Reviewed for technical accuracy May 6, 2022

AWS Reference Architecture