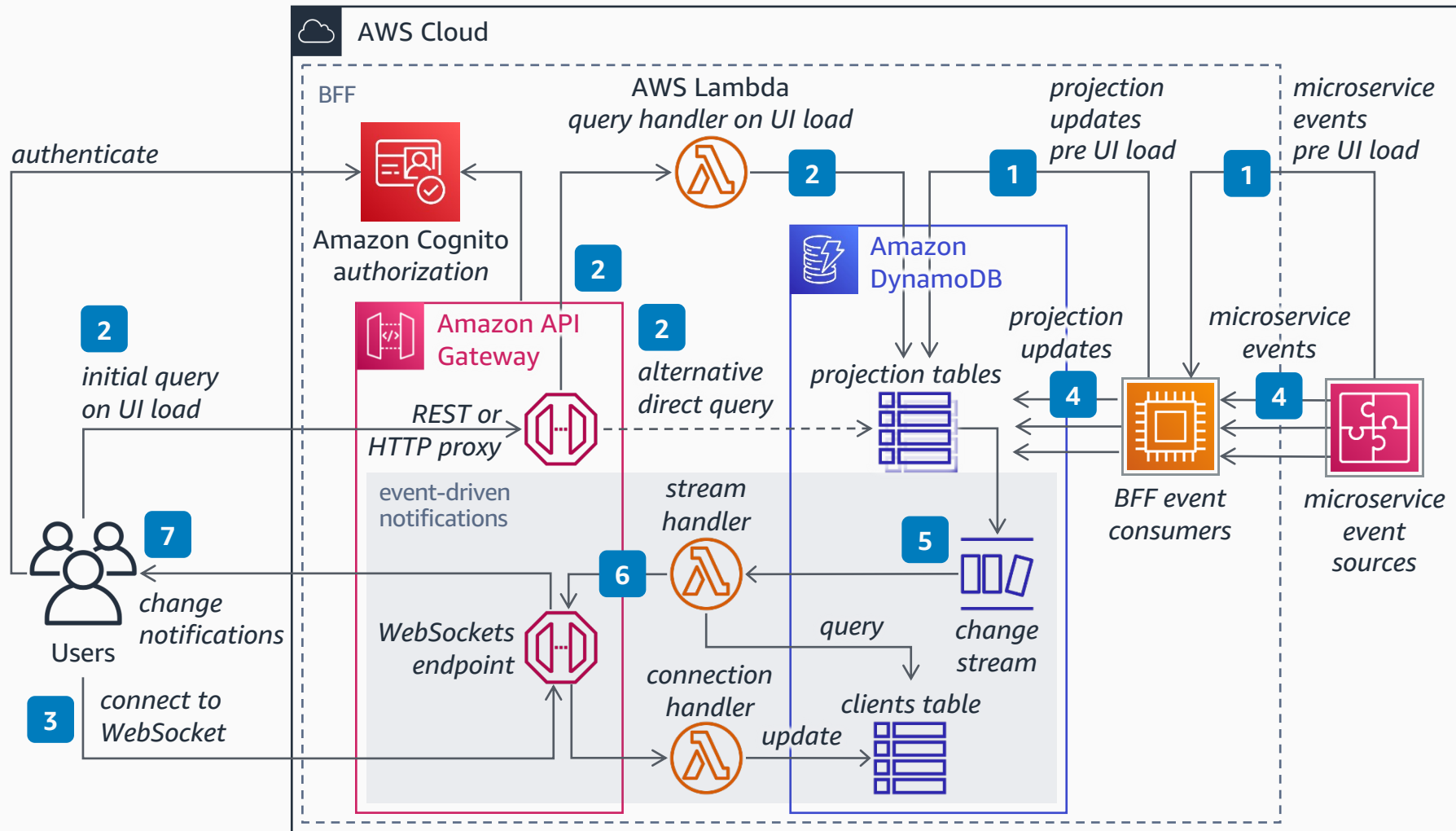


Backend for frontend using API Gateway

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 updating denormalized data projections in **Amazon DynamoDB** for frontend consumption.
- 2 On UI load, frontend clients authenticate with **Amazon Cognito**, then query the data by invoking the BFF API built with **Amazon API Gateway**. The data is then fetched in **DynamoDB**, either directly by API Gateway or via a BFF query handler built with **AWS Lambda**.
- 3 Frontend clients subscribe for any subsequent data changes by connecting to a BFF WebSocket endpoint provided by **API Gateway**, which triggers the update of the “connected clients” table.
- 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 pushes notifications to clients connected to **API Gateway**'s WebSockets.
- 7 When the change notification from **API Gateway** is received by the frontend clients, they can refresh the UI content.

