



Architecture Diagrams

Backend for Frontend Using API Gateway



Backend for Frontend Using API Gateway: Architecture Diagrams

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

BFF API Gateway **i**

Backend for Frontend Using API Gateway 1

Further reading 2

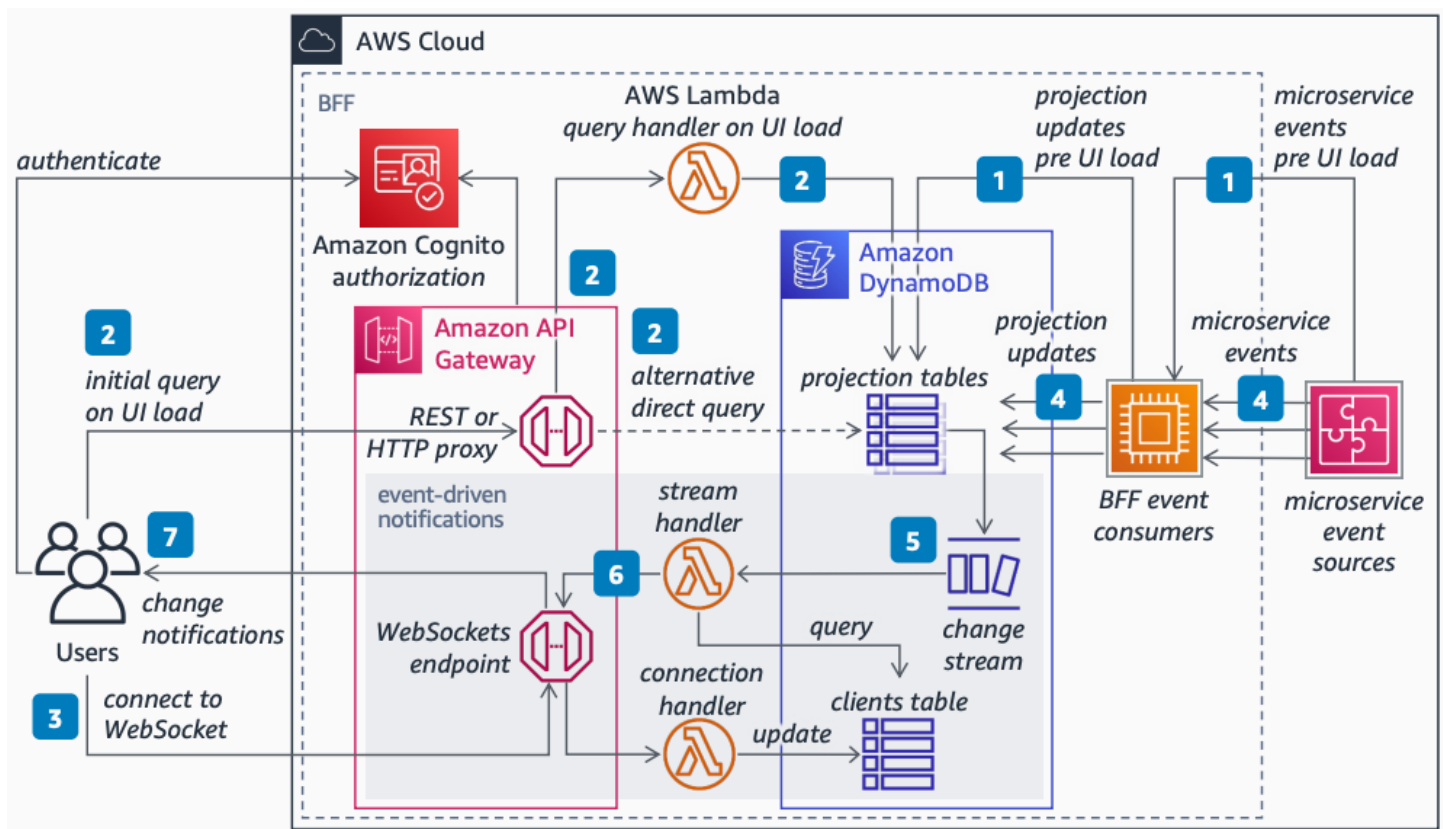
Diagram history 2

Backend for Frontend Using API Gateway

Publication date: April 27, 2022 ([Diagram history](#))

This architecture shows how frontend client applications apply the Backend for Frontend (BFF) pattern to load UI-ready data projections and refresh the UI with event-driven notifications. You use [Amazon API Gateway](#) WebSockets to push real-time updates when microservices raise events about mutations in domain aggregates.

Backend for Frontend Using API Gateway



The following steps describe the architecture:

1. Purpose-built BFF event consumers catch events from your application and update denormalized data projections in [Amazon DynamoDB](#) for frontend consumption.
2. On UI load, frontend clients authenticate with [Amazon Cognito](#), then query data by invoking the BFF API built with API Gateway. The API fetches data from DynamoDB directly or through a BFF query handler built with [AWS Lambda](#).

3. Frontend clients subscribe for subsequent data changes by connecting to a BFF WebSocket endpoint provided by API Gateway, which triggers an update of the connected clients table.
4. BFF event consumers continue to process all relevant events from your application and update the denormalized frontend data view in real time.
5. Amazon DynamoDB Streams captures all events from data changes in the BFF database. A Lambda trigger asynchronously invokes a BFF stream-handler function when it detects new stream records.
6. The BFF stream handler pushes notifications to clients connected to API Gateway WebSockets.
7. Frontend clients receive the change notification from API Gateway and refresh the UI content.

Further reading

For additional information, refer to the following resources:

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	April 27, 2022

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.