



## Architecture Diagrams

# Backend for Frontend Using AppSync



# Backend for Frontend Using AppSync: 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 AppSync** ..... **i**

Backend for Frontend Using AppSync ..... 1

Further reading ..... 2

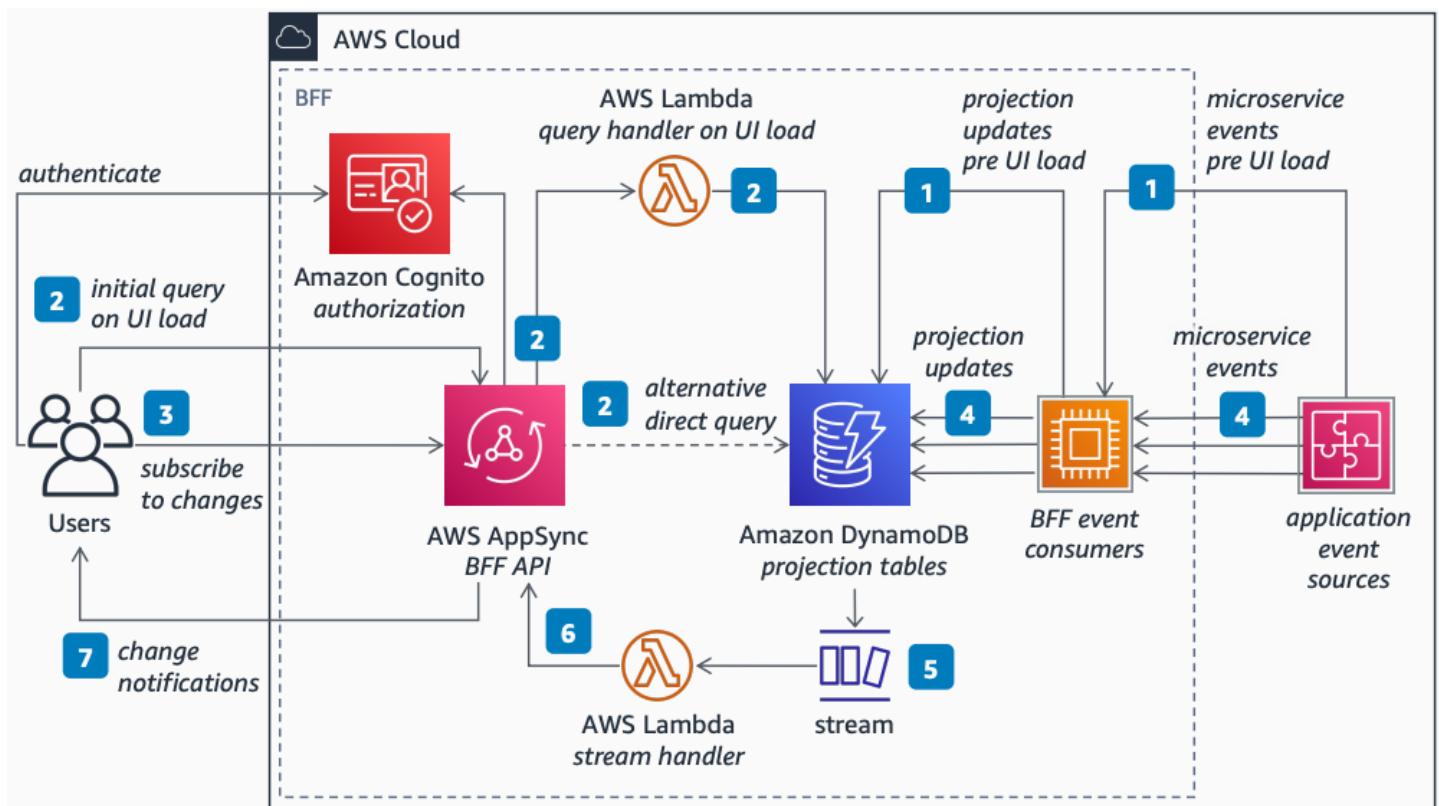
Diagram history ..... 2

# Backend for Frontend Using AppSync

Publication date: May 6, 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 AppSync](#) GraphQL subscriptions over WebSockets to push real-time updates when microservices raise events about mutations in domain aggregates.

## Backend for Frontend Using AppSync



The following steps describe the architecture:

1. Purpose-built BFF event consumers catch events from your application and keep a denormalized view of data in [Amazon DynamoDB](#) for frontend consumption.
2. On UI load, frontend clients authenticate with [Amazon Cognito](#), then query data with GraphQL by invoking the BFF API built with Amazon AppSync. 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 using Amazon AppSync subscriptions over WebSockets.
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 invokes an empty mutation on the Amazon AppSync GraphQL schema to force the subscription to trigger and send a notification to connected clients.
7. Frontend clients receive the change notification from Amazon AppSync 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        |
|-------------------------------------|-------------------------------------------------|-------------|
| <a href="#">Initial publication</a> | Reference architecture diagram first published. | May 6, 2022 |

### Note

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