aws

Architecture Diagrams

Domain Consistency in Event-Driven Architectures

Domain Consistency in Event-Driven Architectures: Architecture Diagrams

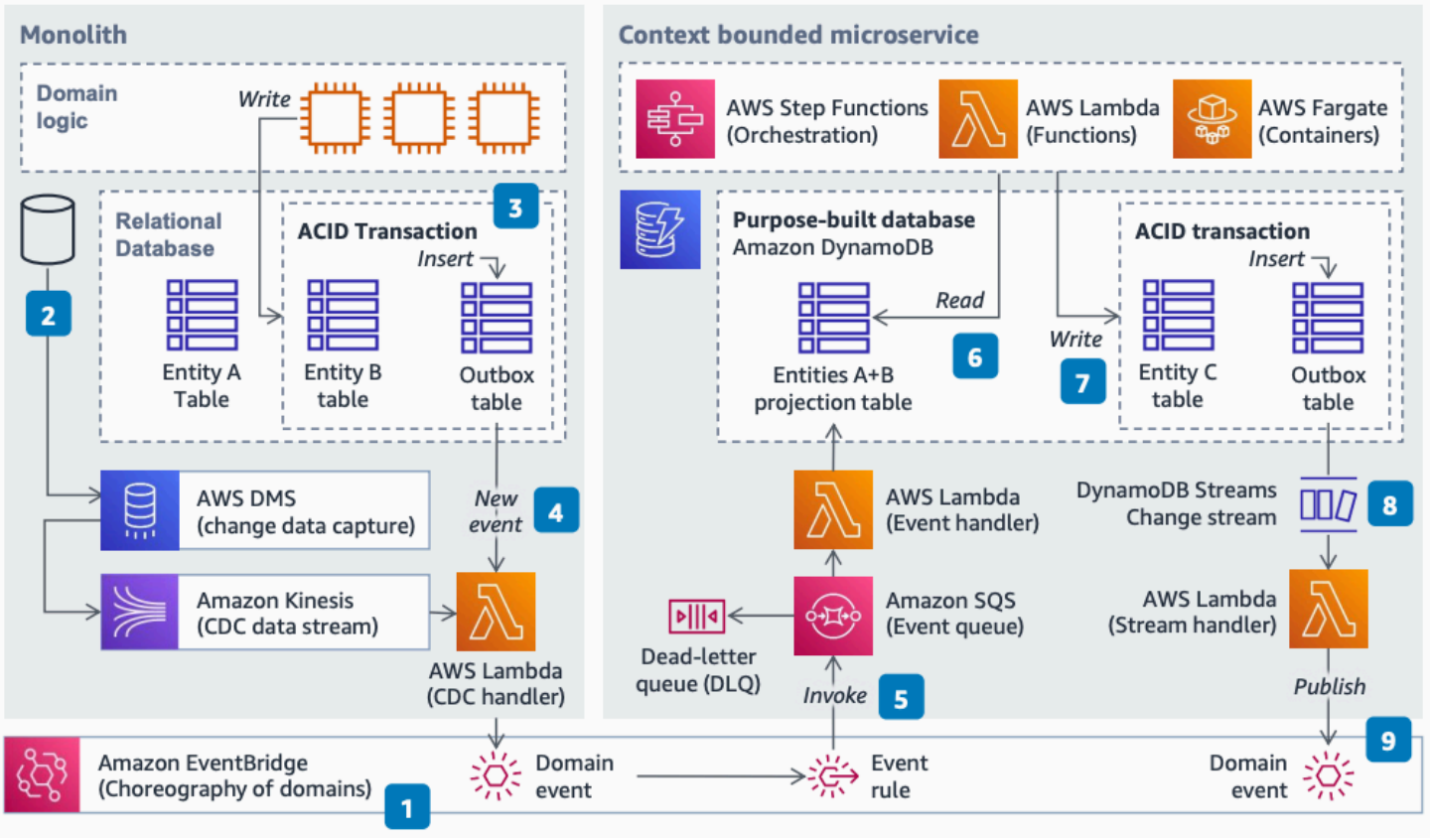
Table of Contents

Domain Consistency in Event-Driven Architectures

Publication date: **March 8, 2022 (Diagram history)**

This architecture shows how to increase the resilience of your event-driven architecture by applying the transactional outbox pattern to your domain's database transactions. You ensure that all local changes raise a corresponding domain event that other domains can project, enabling eventual consistency across bounded contexts.

Domain Consistency in Event-Driven Architectures



The following steps describe the architecture:

1. To support event choreography between your domains, set up a custom event bus using Amazon EventBridge.
2. Detect changes on your monolith's relational database by setting up change data capture (CDC) with AWS Database Migration Service (AWS DMS) and Amazon Kinesis Data Streams.

3. Create an outbox table to insert events following any database change, and wrap the write of both entities and outbox tables inside a transactional operation with atomicity, consistency, isolation, and durability (ACID).
4. Capture the inserts in the outbox table using an AWS Lambda function listening to the Kinesis CDC stream, and publish them as domain events in the event bus.
5. On your context-bounded microservices, capture the relevant external events by setting up a rule in EventBridge to push them into an Amazon Simple Queue Service queue, then handle them with a Lambda function and an Amazon SQS dead-letter queue (DLQ).
6. Use the incoming external domain events to build eventually consistent projected representations of those other domains' databases, implemented with Amazon DynamoDB tables, for local logic to read.
7. On your microservices, repeat the same transactional mechanism around both entity tables and the event outbox table.
8. Capture the inserts on the event outbox table using DynamoDB Streams.
9. Handle the data stream with a Lambda function, publishing the inserts as domain events in the event bus.

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.	March 8, 2022

Note

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