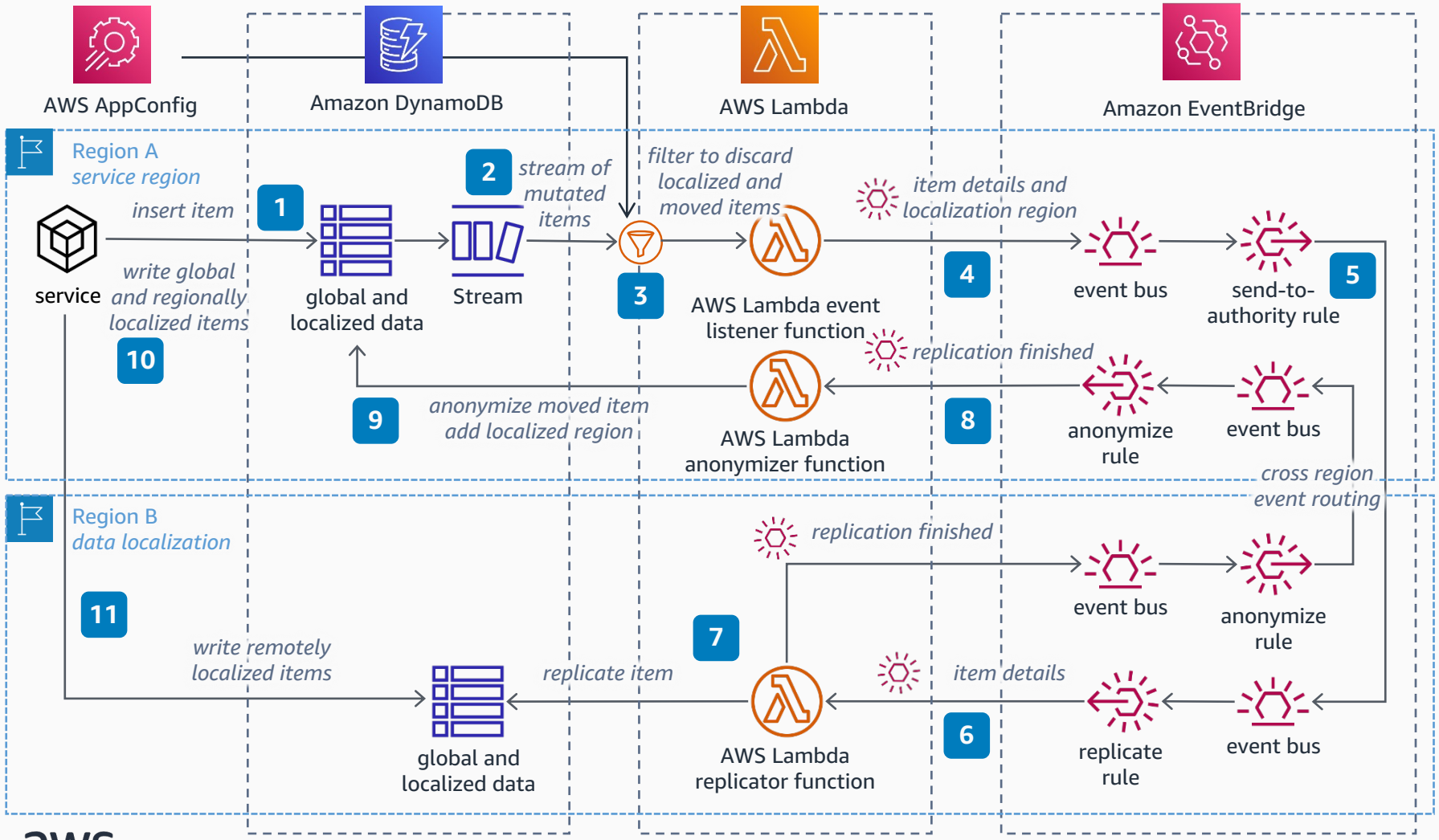
DynamoDB Global Replication with Data Localization

Use this architecture for global data-item replication in DynamoDB with scalability and performance while working to comply with data localization mandates for data to be stored in a specific Region only.



1. On your main AWS Region, set your services to add items with a localization-authority field into a regional **Amazon DynamoDB** table.
2. Enable **Amazon DynamoDB Streams** to capture events about mutated items and handle the events with an **AWS Lambda** event-listener function.
3. Map localization authorities to Regions with **AWS AppConfig**. On the function, read the map, then discard changes on items localized in the current region and items already moved using event-filtering.
4. Raise an event containing the mutated item details and the localization region to an event bus in **Amazon EventBridge**.
5. Set up **EventBridge** rules for cross-Region event routing. Send global items to all Regions. Send remotely localized items only to the localization Region defined on the payload.
6. On destination Regions, handle received item details with an **AWS Lambda** replicator function.
7. Replicate received items in regional **DynamoDB** tables, then raise an event to the Region of origin, indicating that the replication finished.
8. On the Region of origin, handle the finished event with an anonymizer **Lambda** function.
9. Set the function to anonymize sensitive information from localized moved items, stamping them with the localized Region that they moved into. Ignore global items.
10. When your services are updating items, only update global items and localized items that belong to the same Region they're stored in.
11. If a localized item to be updated has already moved to a different Region, make a cross-region call to the **DynamoDB** of the localized Region to update the record. A similar cross-region call is required to read anonymized fields from localized items that were already moved.

Reviewed for technical accuracy March 10, 2023

AWS Reference Architecture