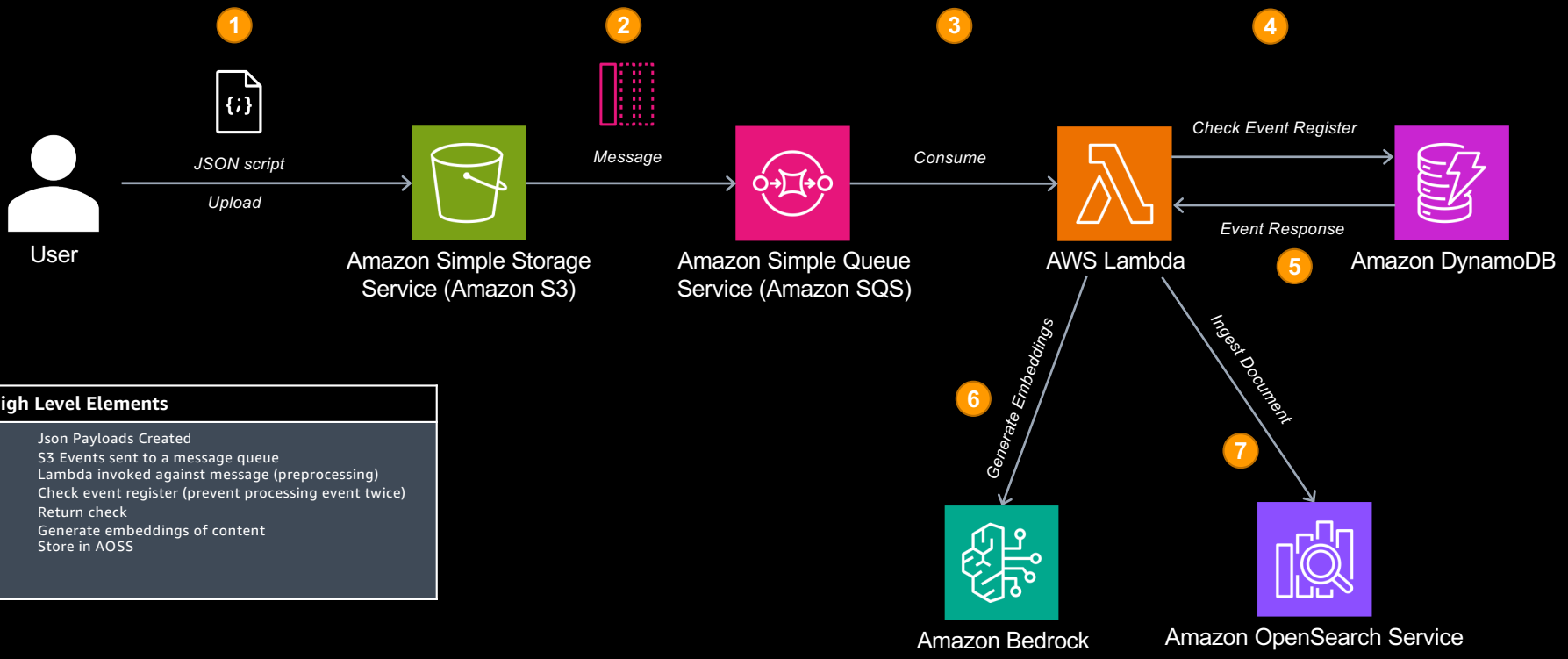
POC Technical Architecture

© 2023, Amazon Web Services, Inc. or its affiliates.

Ingest Trusted Article

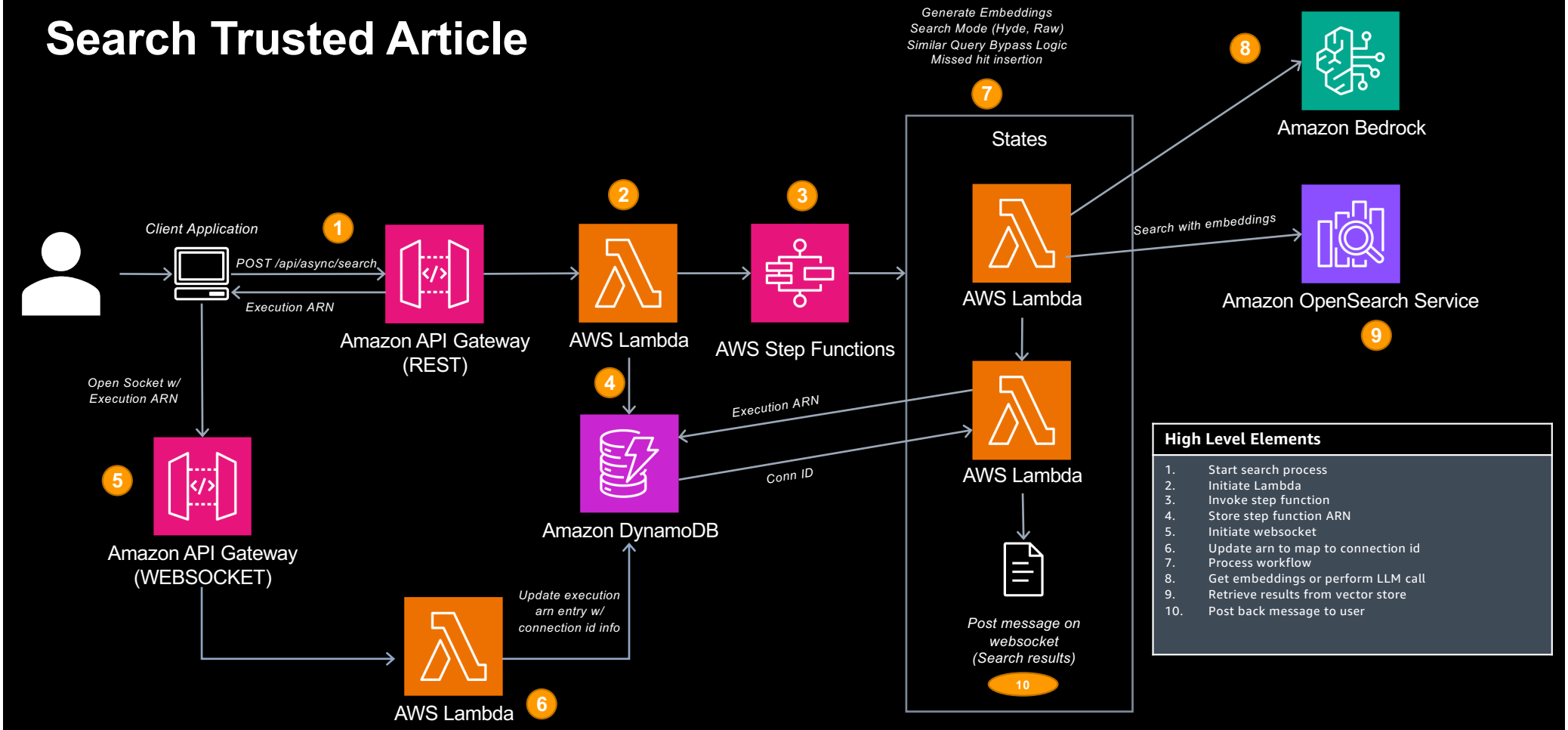


High Level Elements

1. Json Payloads Created
2. S3 Events sent to a message queue
3. Lambda invoked against message (preprocessing)
4. Check event register (prevent processing event twice)
5. Return check
6. Generate embeddings of content
7. Store in AOSS

© 2023, Amazon Web Services, Inc. or its affiliates.

Search Trusted Article

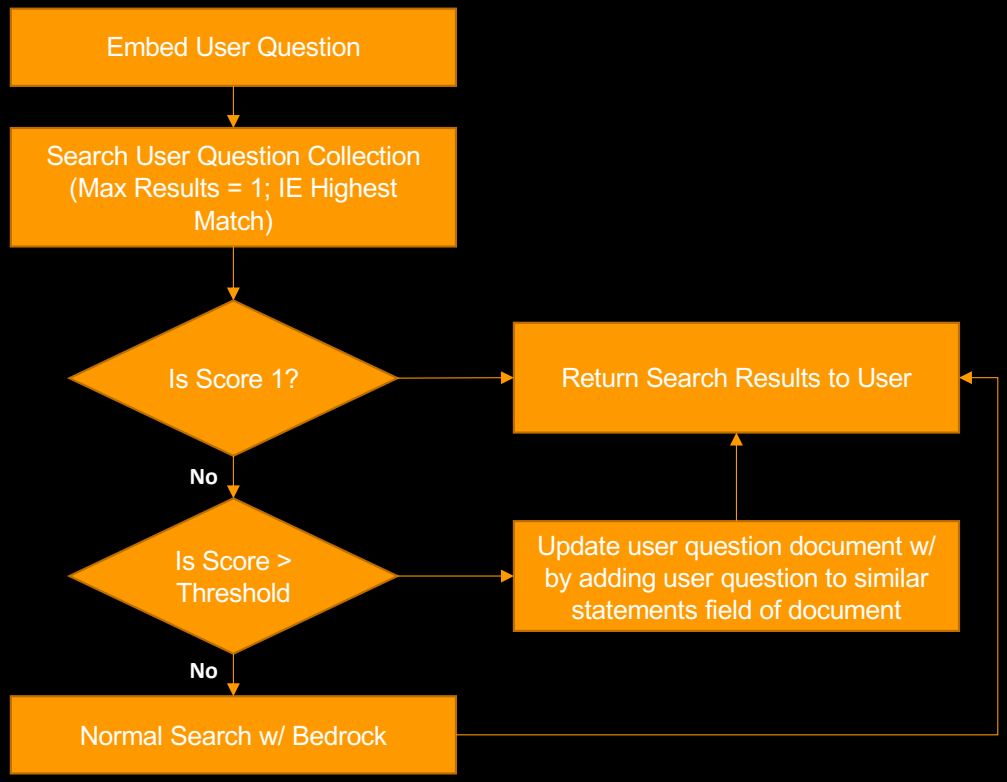


High Level Elements

1. Start search process
2. Initiate Lambda
3. Invoke step function
4. Store step function ARN
5. Initiate websocket
6. Update arn to map to connection id
7. Process workflow
8. Get embeddings or perform LLM call
9. Retrieve results from vector store
10. Post back message to user

© 2023, Amazon Web Services, Inc. or its affiliates.

Similar Query Bypass Logic



Note: *in production we will want to optimize this workflow as well as the data that is stored.*

Currently, user query also includes the document results of matches.

We could strip these and perform a regular search for documents (but bypass Bedrock).

Improvement: *We can utilize caching with Amazon ElastiCache to provide the user with even better response times as well as bypass reads/writes on Amazon OpenSearch Serverless*

Consideration: *Document results when new documents are ingested when pertaining to similar query bypass logic. Possible check would be stored mappings of doc ids in our query versus top 3 that search returned. Mismatches mean we must regenerate Bedrock response and restore accordingly.*

Consideration: *OpenSearch is eventually consistent; have potential to address these mismatch "timings" of documents through clean up routines or address through a processed queue of data (SQS) instead.*

Consideration: *Consider introducing Amazon ElastiCache into production workflows for further latency improvements*

Amazon ElastiCache

© 2023, Amazon Web Services, Inc. or its affiliates.