DATA FLOW ARCHITECTURE

A screenshot of a diagram

Description automatically generated

**1. User Interaction and DNS Resolution:**

* Users interact with the system by visiting domain-specific URLs.
* AWS Route 53 handles DNS resolution, directing recall users to **recall.com** and store users to **store.com**.

**2. Traffic Distribution through Application Load Balancers (ALBs):**

* Route 53 resolves the domain name and directs incoming traffic to the appropriate ALB.
* The **Recall ALB** receives traffic intended for the recall management interface and routes it to the web servers in the Auto Scaling Group within Recall Private Subnet 1.
* The **Store ALB** receives traffic for the store interface and directs it to the web servers in the Auto Scaling Group within Store Private Subnet 2.
* ALBs ensure high availability and distribute incoming application traffic across multiple targets, improving scalability.

**3. Web Server Processing in Auto Scaling Groups:**

* The web servers, likely EC2 instances, process incoming requests. These instances automatically scale out/in based on demand, thanks to Auto Scaling Groups.
* The recall web servers present the management interface and handle user interactions for recall initiation and tracking.
* The store web servers provide the store-facing interface for acknowledging recalls and managing inventory responses.

**4. Backend Services:**

* Backend servers in both recall and store subnets handle the application's business logic and communicate with the database for transaction management.
* These services are crucial for processing recall data, user actions, and coordinating the flow of notifications.

**5. Database Interactions with Amazon RDS:**

* Amazon RDS is used for structured data storage, holding recall information, user data, and transaction logs. RDS ensures data persistence, reliability, and automatic backups.
* The backend services interact with RDS to retrieve and update data as part of the recall process.

**6. Message Queuing with Amazon SQS:**

* Amazon SQS handles inter-service messaging, decoupling components and enabling asynchronous communication.
* Messages are queued based on priority (high, medium, low) to ensure timely processing of recalls. The backend services enqueue and dequeue messages for processing.
* The DLQ captures messages that fail to be processed multiple times, allowing for troubleshooting without impacting the system's flow.

**7. Notification Polling and Delivery:**

* The store's backend servers use the Notification Poller Lambda function to check SQS for new messages.
* Retrieved messages trigger the Notification Poller to send out alerts via SES for email notifications and SNS for SMS or push notifications.
* SES and SNS are managed services that provide reliable delivery of notifications to end-users, supporting a variety of communication channels.

**8. External Adapters for Third-Party Integrations:**

* Both recall and store systems utilize external adapters, potentially Lambda functions, which integrate with third-party services for additional capabilities, such as posting updates to collaboration tools like Microsoft Teams.
* These adapters provide flexibility in extending the system’s functionality and allow for integration with external systems via APIs or webhooks.

**9. Monitoring, Logging, and Storage:**

* AWS CloudWatch is utilized for monitoring the operational health and performance of the application, setting alarms, and visualizing logs.
* Amazon S3 provides durable object storage for logs and other documents, serving as a centralized repository for audit trails and system backups.
* VPC Flow Logs capture information about the IP traffic going to and from network interfaces within the VPC, offering insights into traffic patterns and potential security issues.

**10. Security and Compliance:**

* The entire system is designed with security in mind; IAM roles and policies govern access, encryption-at-rest and in-transit protect data, and AWS WAF and Shield offer additional layers of protection against common web exploits and DDoS attacks, respectively.

This comprehensive AWS-based architecture is designed to be highly available, scalable, secure, and resilient, ensuring that the recall notification system can handle variable loads, secure user data, and provide timely notifications in a robust and maintainable environment.