cloud based ecommerce web architecture

Building Blocks for a successful e-commerce platform

Front- end

HTML CSS JS

Back-end

Java python Firebase

php Ruby aws

Database & Storage

teradata. IBM DB2 Microsoft SQL Server

Amazon RDS SQL ORACLE DATABASE MySQL

SQL snowflake PostgreSQL

SAP S/4 HANA

Payment Gateways

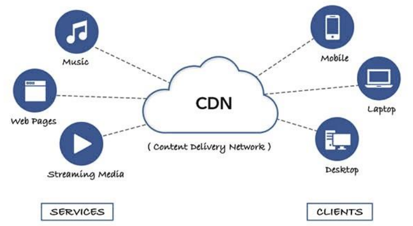
PayPal Payoneer bitcoin VISA pay

Pay mastercard G Pay Skrill RuPay

Maestro xoom Alipay DISCOVER wepay

Slim Pay Braintree cirrus stripe AMERICAN EXPRESS

Content Delivery Network



Security

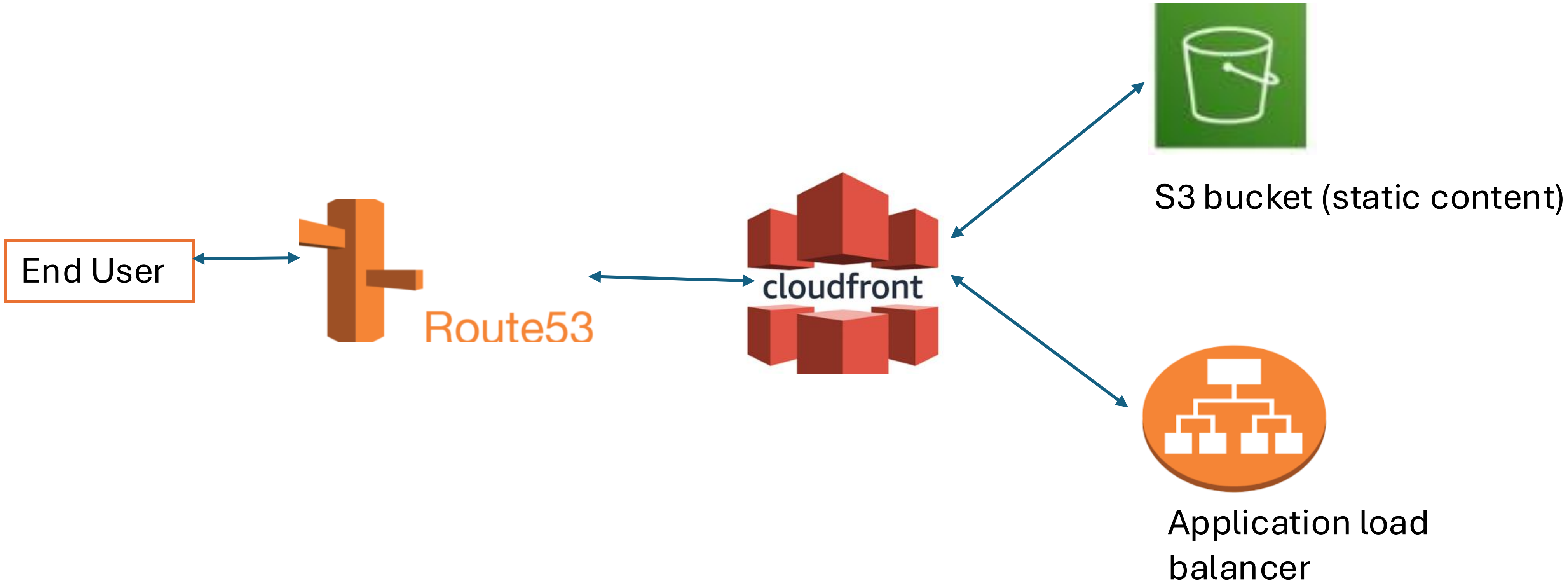
Harden Servers With ServerSecurePLUS

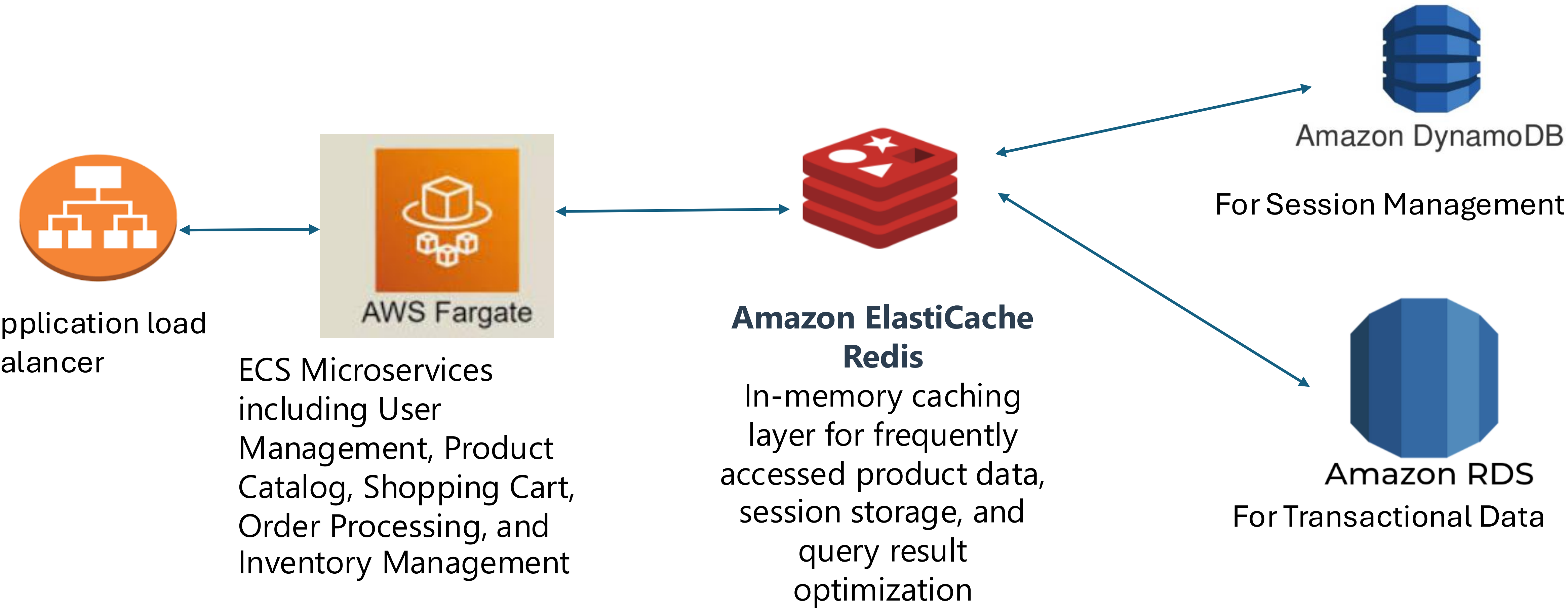
PCI Compliant Server Setup

Off-Server Backups With Acronis Cyber Backup

Prevent Intrusion With a Hardware Firewall

Encrypt Transactions With an SSL Certificate


End User
Route53
cloudfront
S3 bucket (static content)
Application load balancer



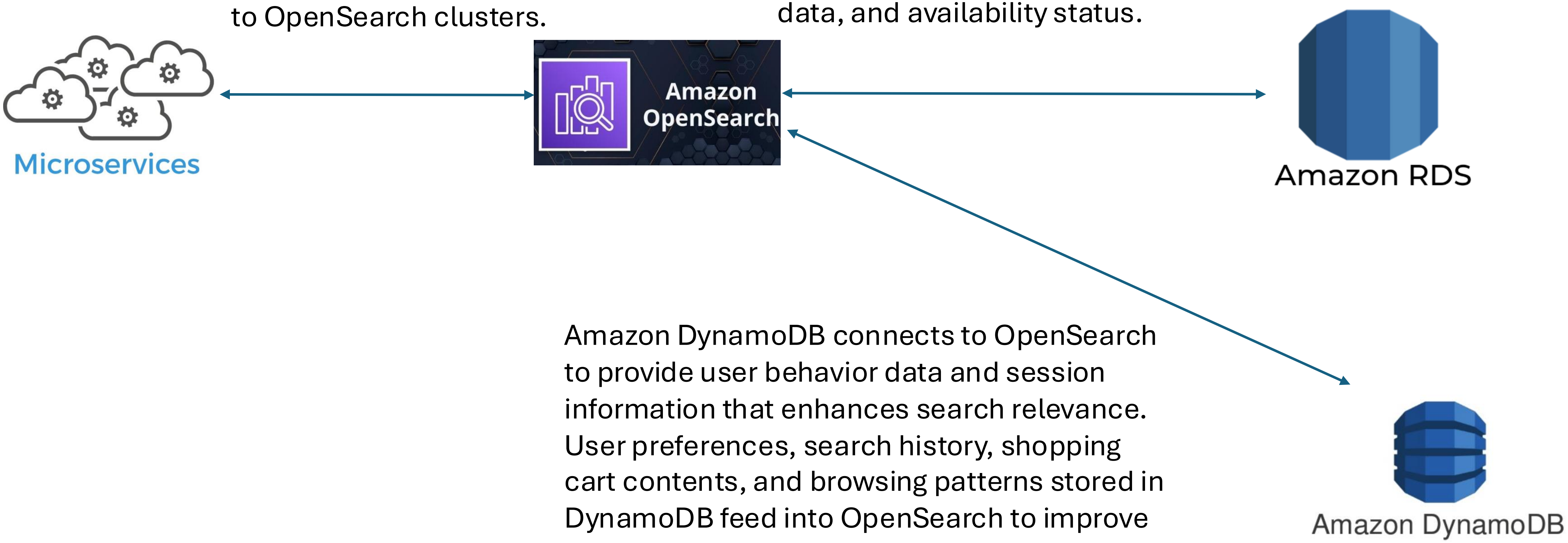
Application load balancer

ECS Microservices including User Management, Product Catalog, Shopping Cart, Order Processing, and Inventory Management

Amazon ElastiCache Redis
In-memory caching layer for frequently accessed product data, session storage, and query result optimization

For Session Management

For Transactional Data



ECS containerized microservices send product search requests, filter queries, and recommendation requests to OpenSearch clusters.

This connection enables OpenSearch to index product catalog information, including product descriptions, specifications, categories, pricing data, and availability status.

Amazon DynamoDB connects to OpenSearch to provide user behavior data and session information that enhances search relevance. User preferences, search history, shopping cart contents, and browsing patterns stored in DynamoDB feed into OpenSearch to improve search result ranking algorithms and enable personalized product recommendations

USER FLOW

STEP:1

Service Interactions:

Browser → Route 53 → CloudFront → WAF → Load Balancer

Initial Request Flow

User browser makes DNS request to Route 53, which resolves to CloudFront edge locations. CloudFront checks cache for static content and forwards dynamic requests to the Application Load Balancer.

STEP:2 (User Login/Registration)

Authentication Flow:

Frontend → API Gateway → User Authentication Service → RDS (user validation)
Authentication Service → DynamoDB (session creation) → ElastiCache (session cache)

User Session Establishment

User submits credentials through frontend to API Gateway. Authentication Service validates against RDS user database, creates session in DynamoDB, and caches session data in ElastiCache for fast subsequent requests.

STEP:3 (Product Search and Browse)

Search Processing:

Frontend → Search Service → OpenSearch (query execution)
Search Service → User Recommendation Service (personalized ranking)

Search and Discovery

User search query goes to Search Service, which queries OpenSearch for relevant products. User Recommendation Service personalizes result rankings based on user history.

STEP:4 (Product Detail Page View)

Product Information Assembly:

Frontend → Product Detail Service → ElastiCache → RDS (product data)
Product Detail Service → Inventory Service → DynamoDB (stock check)
Product Detail Service → Recommendation Service → OpenSearch (related products)

Product Information Loading

Product Detail Service retrieves product information from cache/database, Inventory Service checks real-time stock levels in DynamoDB, and Recommendation Service generates related product suggestions using OpenSearch analytics.

STEP:5 (ADD PRODUCT TO CART)

Cart Management:

Frontend → Shopping Cart Service → Inventory Service (availability check)
Shopping Cart Service → DynamoDB (cart persistence) → ElastiCache (cart cache)
Shopping Cart Service → Recommendation Service (cross-sell items)

Cart State Management

Shopping Cart Service validates product availability through Inventory Service, persists cart state in DynamoDB, caches in ElastiCache for performance, and triggers Recommendation Service for cross-sell suggestions.

STEP:6 CHECK OUT AND Payment

Payment Processing:

Frontend → Order Processing Service → Payment Gateway (payment processing)
Order Processing Service → Inventory Service → DynamoDB (stock reservation)
Order Processing Service → RDS (order creation) → Lambda (notification triggers)

Order Completion

Order Processing Service coordinates payment through external gateway, reserves inventory in DynamoDB, creates order record in RDS, and triggers Lambda functions for email notifications and order confirmations.

Payment Gateway Integration: Core Component After CDN Optimization

1. WHAT IT DOES:

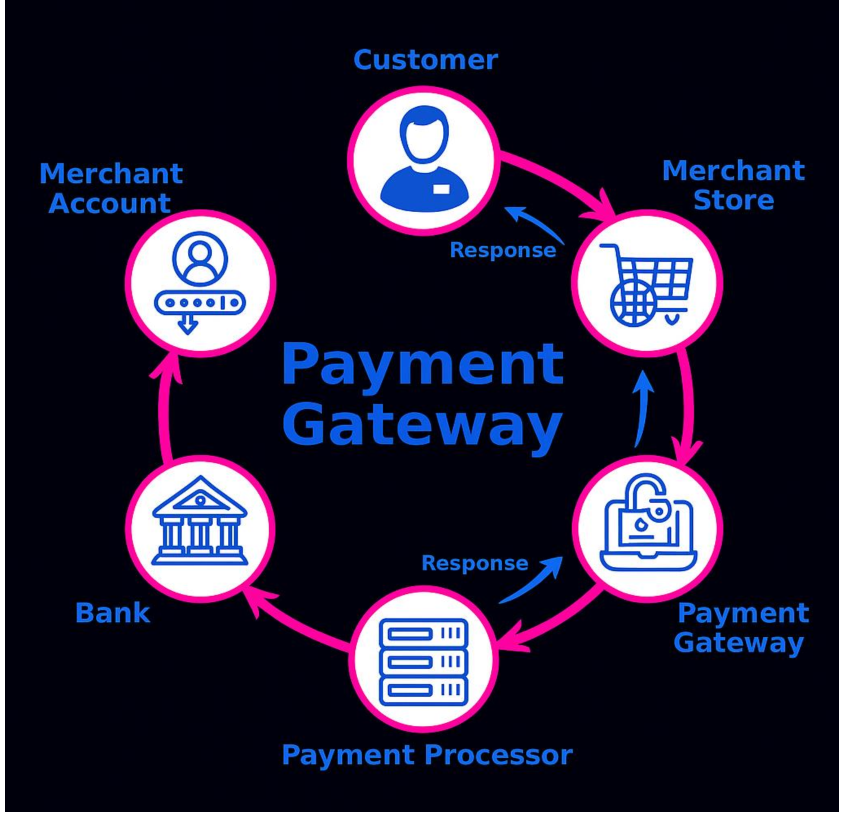
- Offers Multiple Payment Options
- Handles Authorization, Refunds, and Recurring Billing

2. WHY IT MATTERS:

- Security
- Buyer Convenience
- Developer Simplicity & Reliability

Think of Payment Gateway as your "virtual cashier."

When a buyer checks out, out Payment Service hands off their card details to Stripe/PayPal, which talks to the bank, runs fraud checks, and returns an "approved" or "declined" response. Thus, letting us finalize the order without ever touching sensitive card data.



Security Measures and Protocols

Basics of establishing trust with the customer/ Goals of security measures on e-commerce platforms:

- Privacy
- Integrity
- Authentication
- Non- Repudiation

Protocols

- Widespread CDN (content delivery network)
- Multi-factor authentication
- Secure server layer (SSL) certificates
- Firewalls
- Anti- Malware software
- PCI-DSS (payment card industry data security standard) compliance

Maintenance and vulnerability assessment

- Report vulnerability options (White hat hackers)
- Periodic security audit

Ecommerce Security: Importance, Issues & Protection Measures, www.getastra.com/blog/knowledge-base/ecommerce-security/. Accessed 17 July 2025.

Keeping you connected to everything from Namecheap., and Cora Quigley. "How a CDN Can Help Protect against Ddos Attacks." *Namecheap Blog*, 3 Nov. 2020, www.namecheap.com/blog/how-a-cdn-can-help-protect-against-ddos-attacks/.

Logistic API Integration

- Automates shipment tracking and order fulfillment.
- Provides real-time shipping updates and delivery status.
- Supports multiple carriers (DHL, FedEx, UPS).
- Enables rate comparison and returns management.
- Reverse Logistics.

Customer Relationship

- Centralizes customer interactions and order history.
- Syncs data between Customer Relationship Management, Enterprise Resource Planning, and logistics systems.
- Enhances customer experience with personalized updates.
- Integrates with marketing tools for targeted campaigns(ML algorithm for searches).