# Assumptions and Dependencies

## 1. Schema Separation as per ASN

### Assumptions:

- Each ASN will have its own schema in DynamoDB to maintain data isolation.

- Data partitioning will be based on ASN-specific partition keys.

- Role-based access control (RBAC) will be implemented to restrict access to each ASN’s data.

### Dependencies:

- AWS IAM roles and policies must be correctly configured to ensure that each ASN only accesses its own schema.

- Application logic should enforce schema separation in all queries and data access requests.

## 2. Amazon Cognito Integration

### Assumptions:

- Amazon Cognito will be used for user authentication and identity management.

- Cognito will support federated authentication with third-party identity providers (e.g., Google, Facebook, Microsoft AD).

- JWT/OAuth tokens will be validated by API Gateway before granting access to AWS services.

### Dependencies:

- Cognito must be properly configured with IAM roles to allow secure access to AWS resources.

- Third-party identity provider integration requires correct SSO configuration and token validation policies.

## 3. Payment Gateway Implementation

### Assumptions:

- API Gateway and Lambda will process payment requests before sending them to a third-party payment provider (e.g., Stripe, PayPal, Adyen).

- Secure communication (HTTPS) will be enforced for all payment transactions.

- Payment status updates will be stored in DynamoDB and trigger AWS Step Functions for post-payment workflows.

### Dependencies:

- Integration with external payment providers requires API credentials and proper webhook configurations.

- Compliance with security and financial regulations (PCI DSS) must be maintained.

## 4. IAM Security Components in AWS Architecture

### Assumptions:

- IAM policies and roles will enforce the least privilege principle.

- Each AWS service (e.g., DynamoDB, S3, Lambda, Step Functions) will have specific IAM roles with scoped permissions.

- AWS IAM session management will be used to track user activity and API calls.

### Dependencies:

- IAM policies must be continuously reviewed and updated based on security best practices.

- Logging and monitoring (via AWS CloudTrail) must be enabled to detect unauthorized access.

## 5. Security Considerations

### Assumptions:

- Data encryption will be enforced both at rest (using AWS KMS) and in transit (using TLS 1.2/1.3).

- AWS WAF will protect API Gateway from common web threats (DDoS, SQL Injection, XSS, etc.).

- AWS Secrets Manager will securely store and manage API keys, database credentials, and payment gateway secrets.

### Dependencies:

- Security policies and compliance guidelines must be followed based on industry best practices.

- Regular security audits and penetration testing must be conducted to identify vulnerabilities.

## 6. IAM Entities in AWS

### Assumptions:

- IAM Users will be assigned based on roles (Admin, ASN Manager, Finance, Support, etc.).

- IAM Roles will be mapped to specific service permissions (e.g., Lambda Execution Role, DynamoDB Read/Write Role, API Gateway Role).

- IAM Groups will be used to manage access control at an organizational level.

### Dependencies:

- AWS Organizations and Service Control Policies (SCPs) must be implemented for multi-account governance.

- IAM roles must be properly mapped to Cognito user authentication flows.