Asset Security and Data Protection Lifecycle

Comprehensive Guide to Data Governance and Protection

Course Overview

Course Objective:

The complete data protection lifecycle—from establishing governance frameworks and classifying information assets, to implementing technical controls across all data states, and ensuring compliant data disposition. This course prepares you to design and manage enterprise-level data security programs that protect organisational assets whilst meeting regulatory requirements.

Why This Matters:

Data breaches cost organisations an average of a million per incident. Proper asset security isn't just about technology—it's about understanding data's value, implementing appropriate controls based on sensitivity, and maintaining accountability throughout the data lifecycle.

Part 1: Foundation - Data Governance & Classification

What is Data Governance?

Data governance is the formal system of decision rights and accountabilities that ensures your organisation's information assets are managed as strategic resources. It answers: *Who owns this data? What can we do with it? How long must we keep it? How do we protect it?*

1.1 Establishing Your Data Management Process

01

Document Your Data Management Policy

Create a central policy that addresses:

- **Data Sensitivity Levels**
- **Ownership Assignment**
- **Handling Procedures**
- **Retention Requirements**
- **Disposal Procedures**

02

Establish Review Cycles

Annual policy reviews (minimum)
Trigger reviews when:

- New regulations affect the industry
- Major security incidents happen
- Technology infrastructure changes substantially
- Significant business changes occur (mergers, new products)

03

Ensure Policy Distribution

- Make policies accessible
- Require ack during onboarding
- Provide regular training updates
- Maintain version control

Impact:

Without documented processes, data protection becomes inconsistent and compliance gaps emerge.
A formal process ensures everyone follows the same standards and provides legal defensibility.

1.2 Defining Data Roles and Accountability

The Three Critical Roles:

Data Owner

Responsibility: Ultimate organisational accountability for specific data sets

Key Duties:

- Determines data classification level
- Defines who should have access and why
- Approves access requests
- Sets retention and handling requirements
- Bears responsibility for data breaches involving their data

Typical Role: Business unit leaders, department heads, or process owners

Data Custodian

Responsibility: Technical implementation and maintenance of controls

Key Duties:

- Implements security controls specified by
- the Data Owner
- Performs backups and restoration
- Maintains system configurations
- Monitors access logs
- Applies patches and updates

Typical Role: IT administrators, database administrators, system engineers

Data Processor/Controller

Responsibility: Handles data according to owner instructions

Key Duties:

- Processes data only as authorised
- Implements contractual security requirements
- Reports security incidents to the data owner
- Ensures compliance with regulations.
- Maintains processing records

Typical Role: Third-party vendors, cloud service providers, outsourced services

Creating Accountability:

- Document role assignments in a RACI matrix (Responsible, Accountable, Consulted, Informed)
- Include data ownership in job descriptions
- Establish escalation procedures for data-related decisions
- Create a data stewardship committee for cross-functional coordination

❑ Impact:

Clear roles prevent the "everybody's responsible means nobody's responsible" problem. When data owners are identified, they can be held accountable for protection decisions and compliance.

1.3 Establishing Your Data Classification Scheme

Why Classification Matters:

Not all data requires the same level of protection. Classification allows you to allocate security resources efficiently.

Sensitive (Restricted)

- Financial records.
- Health records
- Authentication credentials

Required Controls:

- Encryption at rest and in transit
- Multi-factor authentication for access
- Strict need-to-know access controls
- DLP monitoring
- Audit logging of all access
- Secure disposal requirements

Confidential (Internal Use Only)

- Internal policies and procedures
- Employee directory information
- Internal project documentation
- Non-public financial data
- Vendor contracts

Required Controls:

- Access controls based on job functio
- Basic encryption for remote access
- Standard backup procedures
- Controlled sharing with third parties

Public

- Marketing materials
- Published financial statements
- Press releases
- Product documentation

Required Controls:

- Integrity protection
- Version control
- Basic access controls to prevent unauthorized editing

Government/Military Classification Levels (Top Secret, Secret, Confidential)

,

1.4 Maintaining Data Inventory and Mapping Data Flows

What to Document:

- **Data Element:** Specific type of data (e.g., customer email addresses)
- **Classification Level:** Sensitivity rating
- **Data Owner:** Person accountable for the data
- **Location:** Where data is stored (servers, databases, cloud services)
- **Format:** Structured (databases) vs. unstructured (documents, emails)
- **Volume:** Approximate quantity
- **Retention Period:** How long it must be kept
- **Regulatory Requirements:** Which regulations apply (GDPR, HIPAA, PCI DSS, etc.)

Priority: Focus on Sensitive Data First

- Start with data subject to regulatory requirements
- Include data that would cause significant harm if breached
- Expand to confidential data as resources permit

Inventory Tools:

- Data discovery tools (automated scanning)
- Configuration management databases (CMDB)
- Data classification software
- Spreadsheets for smaller organisations

1.4 Maintaining Data Inventory and Mapping Data Flows --Continue

Mapping Data Flows

> ### What is a Data Flow Diagram (DFD)?
>
> A visual representation showing how data moves through your systems, networks, and processes —from collection to disposal.

Components to Map:

1. **Data Sources:** Where data originates (web forms, APIs, manual entry)
2. **Processing Points:** Systems that transform or use the data
3. **Storage Locations:** Databases, file servers, backup systems
4. **Transit Paths:** Networks, VPNs, internet connections
5. **Third-Party Transfers:** Vendors, cloud providers, partners
6. **End Points:** Where data is ultimately used or disposed of

Example: E -commerce Data Flow

Customer Browser (HTTPS)→ Web Server → Application Server → Payment Gateway → Payment Processor → Bank → ← Transaction Response ← Payment Gateway ← Application Server → Database (Encrypted PAN Storage) → Backup System

1.4 Maintaining Data Inventory and Mapping Data Flows --Continue

Why Map Data Flows?

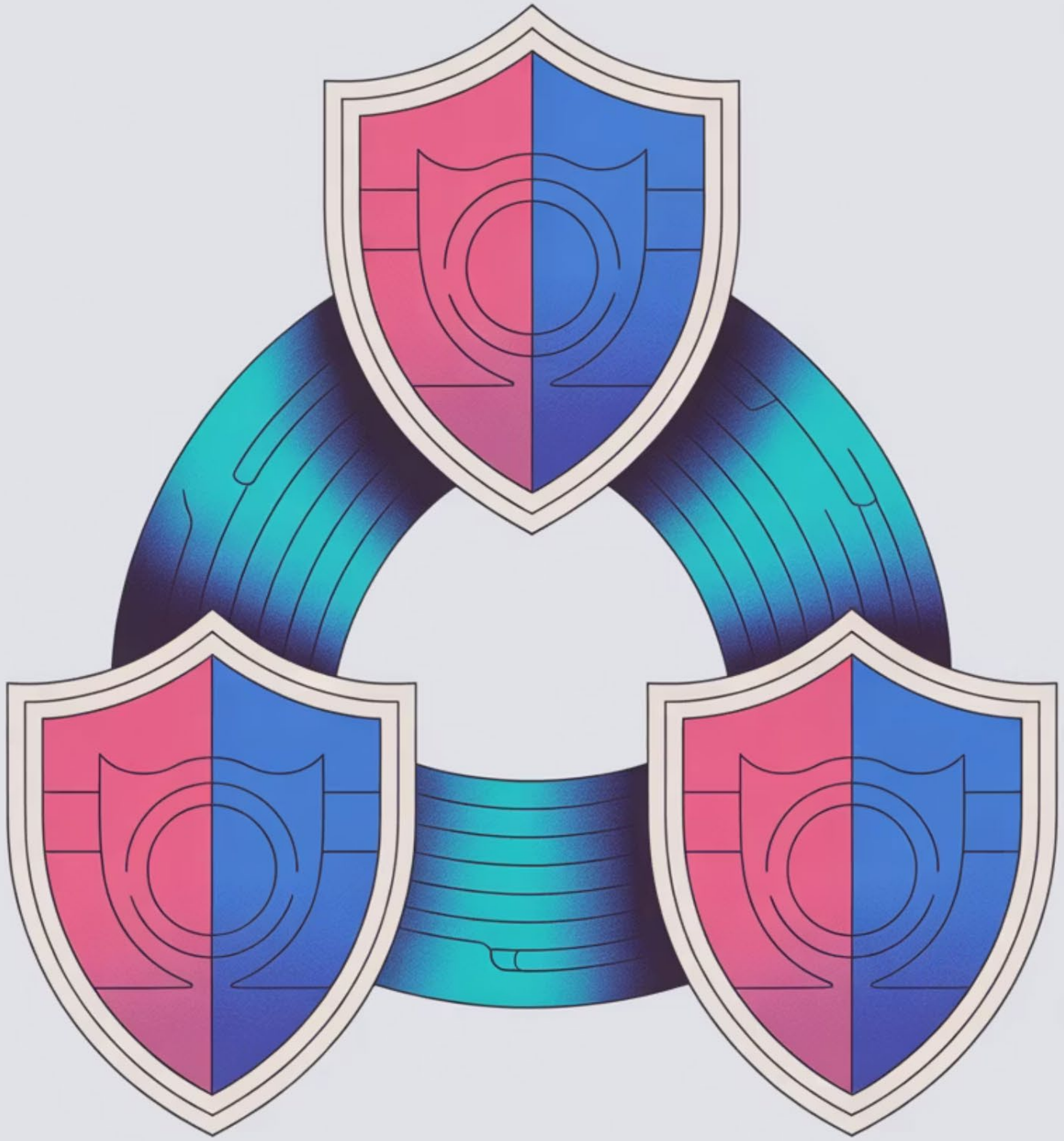
- **Identify Protection Gaps:** Find where data lacks encryption or access controls
- **Define Security Boundaries:** Establish your Data Environment
- **Scope Compliance Efforts:** Determine which systems must meet regulatory requirements
- **Support Incident Response:** Quickly trace where compromised data may have travelled
- **Facilitate Audits:** Provide auditors with clear documentation

DFD Best Practices:

- Update flows when systems change
- Include encryption status at each stage
- Colour-code by data sensitivity
- Mark regulatory boundaries clearly
- Document data transformations (when data is modified)

Impact:

Organisations that don't know where their sensitive data resides cannot adequately protect it. The 2023 Verizon DBIR found that 82% of breaches involved data the organisation didn't know it had or where it was stored. An accurate inventory and data flow mapping prevents "shadow data" from becoming your security blind spot.



Part 2: Technical Controls - Protecting Data in All States

Understanding the Three Data States

Every piece of data exists in one of three states at any given time:

Data at Rest

Stored on hard drives, SSDs, databases, backups, archives

Data in Transit

Moving across networks, between systems, over the internet

Data in Use

Actively being processed in system memory (RAM) or CPU

Each state requires different protection strategies.

2.1 Access Controls - The Foundation of Data Protection

Principle: Need to Know & Least Privilege

Need to Know: Users should only access data required to perform their specific job functions, nothing more.

Least Privilege: Users should have the minimum level of access (read, write, modify, delete) necessary to do their jobs.

Implementation Framework

01

Role-Based Access Control (RBAC)

- Define job roles and their data access requirements
- Group users into roles (e.g., Sales Rep, Finance Analyst)
- Assign permissions to roles, not individuals
- Assign users to appropriate roles

02

Configure Access Control Lists (ACLs)

- Set file system permissions on sensitive directories
- Configure database permissions at the table/column level
- Implement application-level access controls
- Use network segmentation to control system access

03

Implement Regular Access Reviews

- Quarterly reviews of user permissions
- Immediate removal when roles change or ends
- Challenge "exceptions" to standard role permissions
- Document all access decisions

04

Enforce Separation of Duties

- No single person should control an entire sensitive process
- Divide critical functions (e.g., one person initiates payments, another approves)
- Prevent conflicts of interest

2.1 Access Controls - The Foundation of Data Protection --continue

Example:

Role: Customer Service Representative

Access Granted:

- Customer contact information (Read/Write)
- Order history (Read Only)
- Support ticket system (Read/Write)

Access Denied:

- Customer payment information
- Financial reports
- System configuration

Advanced Access Control Techniques:

Attribute -Based Access Control (ABAC)

- Access based on attributes (department, clearance level, location, time of day)
- More granular than RBAC

Mandatory Access Control (MAC)

- System enforces access based on data classification and user clearance
- Used in military and government environments
- Users cannot change permissions on data they create

Impact:

The 2023 Cost of Insider Threats Report found that excessive access privileges contributed to 45% of insiderelated incidents. Proper access controls are your first line of defence against both external attackers and insider threats.

2.2 Encrypting Data at Rest

> ### What is Encryption at Rest?
>
> Encryption at rest transforms stored data into unreadable ciphertext, protecting it if physical storage media is stolen or improperly disposed of.

Where to Encrypt Data at Rest

End -User Devices

Laptops and desktops: Use full-disk encryption

- Windows: BitLocker
- macOS: FileVault
- Linux: LUKS (Linux Unified Key Setup)

Mobile devices: Enable device encryption

- iOS: Encryption enabled by default (verify it's on)
- Android: Enable encryption in security settings

Removable media: USB drives, external hard drives

Servers and Infrastructure

- **File servers:** Encrypt sensitive file shares
- **Application servers:** Encrypt application data directories
- **Databases:** Implement database encryption
- **Virtual machines:** Encrypt VM disk images
- **Backup systems:** Encrypt backup data

2.2 Encrypting Data at Rest -- Continue

Encryption Approaches

Full-Disk Encryption (FDE)

What: Encrypts entire storage device

Pros:

- Simple to implement
- Protects all data automatically
- Good for lost/stolen devices

Cons:

- Doesn't protect data when system is running
- All data decrypted when disk is mounted
- Doesn't protect against compromised OS

File-Level Encryption

What: Encrypts individual files or folders

Pros:

- Granular control over what's encrypted
- Different keys for different files
- Protection persists when files are moved

Cons:

- Users must remember to encrypt files
- Management overhead
- File metadata may remain visible

Database Encryption

Transparent Data Encryption (TDE) - Storage Layer

What: Database encrypts data files on disk

Pros:

- Transparent to applications
- No code changes required
- Protects against storage theft

Cons:

- Data is decrypted when accessed by database
- No protect against SQL inject or compromised accounts
- Database administrators can access plaintext

Application-Layer Encryption (Column-Level)

What: Encrypts specific data fields before storing in database

Pros:

- Data remains encrypted in database
- Even DBAs cannot read sensitive fields
- Protects against database compromises
- Recommended for highest sensitivity data

Cons:

- Requires application code changes
- More complex to implement
- Can impact database performance (indexing, searching)

2.2 Encrypting Data at Rest -- Continue

Encryption Approaches

Cryptographic Standards

- **Use FIPS 140-2/140-3 validated cryptography**
- **Algorithms:** AES-256, RSA-2048 or higher
- **Minimum:** AES-128 or stronger
- **Recommended:** AES-256
- **Key Management:** Cryptographic keys must be protected with strong access controls

Key Management Best Practices

1. **Separate keys from encrypted data:** Never store encryption keys on the same system as encrypted data
2. **Use key management systems (KMS):** Dedicated systems for key generation, storage, rotation
3. **Implement key rotation:** Change keys periodically (annually minimum, quarterly recommended)
4. **Protect key access:** Require multi-factor authentication to access keys
5. **Establish key recovery procedures:** Plan for key loss without creating security gaps
6. **Document key lifecycle:** Track key creation, distribution, usage, rotation, and destruction

Impact:

According to the Ponemon Institute, lost or stolen laptops account for 20% of data breaches.
For regulated data, encryption at rest is often mandatory and failure to implement it can result in significant fines.

2.2 Encrypting Data at Rest -- Continue

2.3 Encrypting Data in Transit

> **What is Encryption in Transit?**
> Encryption in transit protects data whilst it moves across networks—preventing interception, eavesdropping, and tampering during transmission.

Critical Use Cases for Transit Encryption

External Communications (Internet)

- Web traffic (HTTPS)
- Email (S/MIME, PGP)
- File transfers (SFTP, FTPS)
- Remote access (VPN)
- API communications
- Payment transactions

Internal Communications (Corporate Networks)

- Database connections
- Server-to-server communications
- Wireless networks
- Administrative access

Why Encrypt Internal Traffic?

Many breaches start with external compromise but escalate through unencrypted internal networks. Assume your network perimeter will be breached—encrypt internally to limit damage.

2.3 Encrypting Data in Transit -- Continue

Encryption Protocols and Technologies

Transport Layer Security (TLS)

Current Version: TLS 1.3 (preferred)

- **Acceptable:** TLS 1.2 (phase out by 2025)
- **Deprecated:** TLS 1.0, TLS 1.1, SSL 2.0, SSL 3.0 (never use)

TLS Use Cases:

- **HTTPS:** Web applications, APIs
- **SMTPS:** Email transmission
- **FTPS:** Secure file transfer
- **Database connections:** Postgres, MySQL, SQL Server

TLS Configuration Best Practices:

- Use strong cipher suites (AES-GCM preferred)
- Disable weak ciphers (RC4, DES, 3DES)
- Enable Perfect Forward Secrecy (PFS)
- Use certificates from trusted Certificate Authorities (CAs)
- Implement certificate pinning for critical applications
- Monitor certificate expiration

Secure Shell (SSH)

Protocol: OpenSSH (current standard)

Use Cases:

- Remote server administration
- Secure file transfer (SFTP, SCP)

- IPsec protocol standard
- SSL/TLS VPNs (easier to deploy)

- IPsec VPNs (more secure for high-risk users)

2.3 Encrypting Data in Transit -- Continue

VPN Configuration:

- Require multi-factor authentication
- Use split tunnelling carefully (or disable)
- Implement always-on VPN for remote workers handling sensitive data
- Monitor VPN logs for anomalies

Protecting Authentication in Transit

> **Critical Requirement:**
>
> All authentication credentials MUST be encrypted during transmission.

What Must Be Protected:

- Usernames and passwords
- Session tokens and cookies
- API keys and secrets
- Multi-factor authentication codes
- Biometric data
- Cryptographic keys

Strong cryptography and security protocols must be used to safeguard sensitive cardholder data during transmission over open, public networks. Primary Account Numbers (PANs) must be unreadable during transmission.

2.4 Network Segmentation and Data Isolation

What is Network Segmentation?

Dividing your network into separate zones based on data sensitivity and business function—limiting the blast radius when breaches occur.

Why Segment?

If an attacker compromises one system, segmentation prevents lateral movement to more sensitive systems. Think of it as having fire doors in a building—containing the damage.

Segmentation Strategies

Zero Trust Segmentation

- **Principle:** Never trust, always verify
- **Approach:** Micro-segmentation where every connection is authenticated and authorised
- **Implementation:** Software-defined networking, identity-based access

Perimeter -Based Segmentation

- **Traditional approach:** Separate networks by function
- **Implementation:** VLANs, firewalls, routers with ACLs

Key Security Zones

DMZ (Demilitarised Zone)

- **Purpose:** Buffer zone between internet and internal network
- **Systems:** Public-facing web servers, mail servers, DNS
- **Protection:** Cannot directly access internal network

Trusted vs. Untrusted Networks

2.4 Network Segmentation and Data Isolation -- Continue

Implementation Techniques

□ VLANs (Virtual Local Area Networks) □ Firewalls and Access Control Lists □ Air Gapping

Security Level Inheritance Principle

> Critical Rule:
> When systems of different security levels interact, the entire system must be secured to the highest required level.

Example:

Impact:
The Target breach occurred because payment systems weren't properly segmented
From vendor access. Network segmentation is one of the most effective controls for limiting breach impact.

- **Purpose:** Isolate systems that store, process, ortransmit data
- **Requirements:**
  - Network segmentation from out -of-scope systems
  - Strong access controls
  - Regular scope validation
  - **Never use production data in test environments**

2.5 Data Loss Prevention (DLP)

> ### What is DLP?
> Automated tools that identify, monitor, and protect sensitive data from unauthorised access, use, or transmission—preventing data from leaving your control.

DLP Deployment Models

Network DLP

Location: email or web gateways

Protection: Monitors and blocks data in transit

Use Cases:

- Prevent email the customer.
- Block upload of sensitive files to cloud.
- Detect sensitive data being transmitted.

Endpoint DLP

Location: Installed on end-user devices.

Protection: Monitors and controls data at rest and in use

Use Cases:

- Prevent copying sensitive files to USB drives
- Block screenshots of sensitive applications
- Control printing of confidential documents
- Encrypt files automatically based on classification

Cloud DLP

Location: Cloud access security broker (CASB) or cloud-native DLP

Protection: Monitors data in cloud

Use Cases:

- Detect sensitive data uploaded to SaaS
- Monitor cloud storage repositories

DLP Detection Methods

Content Inspection

- **Pattern matching:** Regular expressions.
- **Keywords:** Specific terms indicating sensitive content
- **Exact data matching:** Hashes of known sensitive files

Contextual Analysis

- File type and size
- User role and behaviour
- Destination and recipient
- Time and location
- Data classification labels

Statistical Analysis

- Machine learning models
- Behavioural analytics
- Anomaly detection

2.5 Data Loss Prevention (DLP) -- Continue

DLP Response Actions

Block
Prevent the action entirely

Quarantine
Hold for admin review

Alert
Notify security team

Encrypt
Automatically encrypt sensitive data

Log
Record for audit purposes

User notification
Warn user about policy violation

Implementation Roadmap

Phase 1: Discovery (Monitor Only)

- Deploy in monitoring mode
- Identify where sensitive data exists
- Build baseline of normal data flows
- Tune policies to reduce false positives

Phase 2: Detection and Alerting

- Alert on policy violations
- Investigate incidents
- Refine policies based on real-world use

Phase 3: Prevention and Enforcement

- Enable blocking of high-risk violations
- Enforce automatic encryption
- Implement user education workflows

Phase 4: Integration and Optimisation

- Integrate with SIEM for correlation
- Automate incident response workflows
- Continuously tune and optimise

2.5 Data Loss Prevention (DLP) -- Continue

DLP Policy Examples

- **Trigger:** Detect patterns matching credit card numbers (Luhn algorithm)
- **Action:** Block transmission via email or web upload
- **Exception:** Allow within authorised payment applications

Personally Identifiable Information (PII)

- **Trigger:** Multiple PII elements in single file (name + DOB)
- **Action:** Require encryption if transmitted externally
- **Alert:** Notify security team

Intellectual Property Protection

- **Trigger:** Files marked as "Confidential" or "Trade Secret"
- **Action:** Block upload to personal cloud storage
- **Allow:** Transfer to approved corporate cloud with encryption

2.6 Data Masking and Display Controls

What is Data Masking?

Techniques that obscure sensitive data, revealing only what's necessary for a specificpurpose —protecting confidentiality whilst maintaining usability.

Masking Techniques

Redaction (Truncation)

Method: Show only partial data

Example:

- Credit card: **** **** **** 1234
- Email: j***@example.com

Masking (Character Substitution)

Method: Replace characters with X, *, or other symbols

Example: Visal DOEUK becomes XXXXXXXXXX

Tokenisation

Method: Replace sensitive data with nonsensitive surrogate value

Example: Replace credit card 4532 1234 5678 9010 with token TKN-7839-4821

Use: Token can be used in systems

Advantage: No sensitive data in most systems

Encryption -Based Masking

Method: Encrypt data, show encrypted value

Advantage: Reversible for authorised users

Anonymisation

Method: Remove identifying information completely

Use: Analytics, testing, research

Example: Remove names, addressesfrom data sets

Pseudonymisation

Method: Replace identifying fields with pseudonyms

Reversible: Yes, with lookup table

Compliance: Recognised as privacyenhancing technique

2.6 Data Masking and Display Controls --Continue

Display Requirements

Primary Account Number (PAN) Display Rules:

Mandatory Masking

- When displaying PAN on screen, papereceipt…
- Maximum visible: First 6 digits (BIN) and last 4 digits
- Example: 123456******1234

Who Can See Full PAN

- Only individuals with legitimate business need
- Document business justification
- Implement technical controls to prevent unnecessary access
- Log all full PAN access

Receipt Requirements

- Paper receipts: Maximum last 4 digits visible
- Electronic receipts: Same masking requirements
- Expiration date: Must NOT appear on customer receipts
- Name: Acceptable if needed for matching signature

Implementation in Applications

Database Views

- Create masked views for different user roles
- Full data in base table
- Masked views for general users

```
CREATEVIEWcustomer_masked  AS
SELECTcustomer_id, CONCAT(LEFT(email,2),
***@***') as email FROM customers
```

Dynamic Data Masking

- Database feature (SQL Server, Oracle, PostgreSQL)
- Automatic masking based on user permissions
- No application code changes required

Application -Layer Masking

- Mask in application code based on user role
- Maintain full data in database
- Present masked version to UI

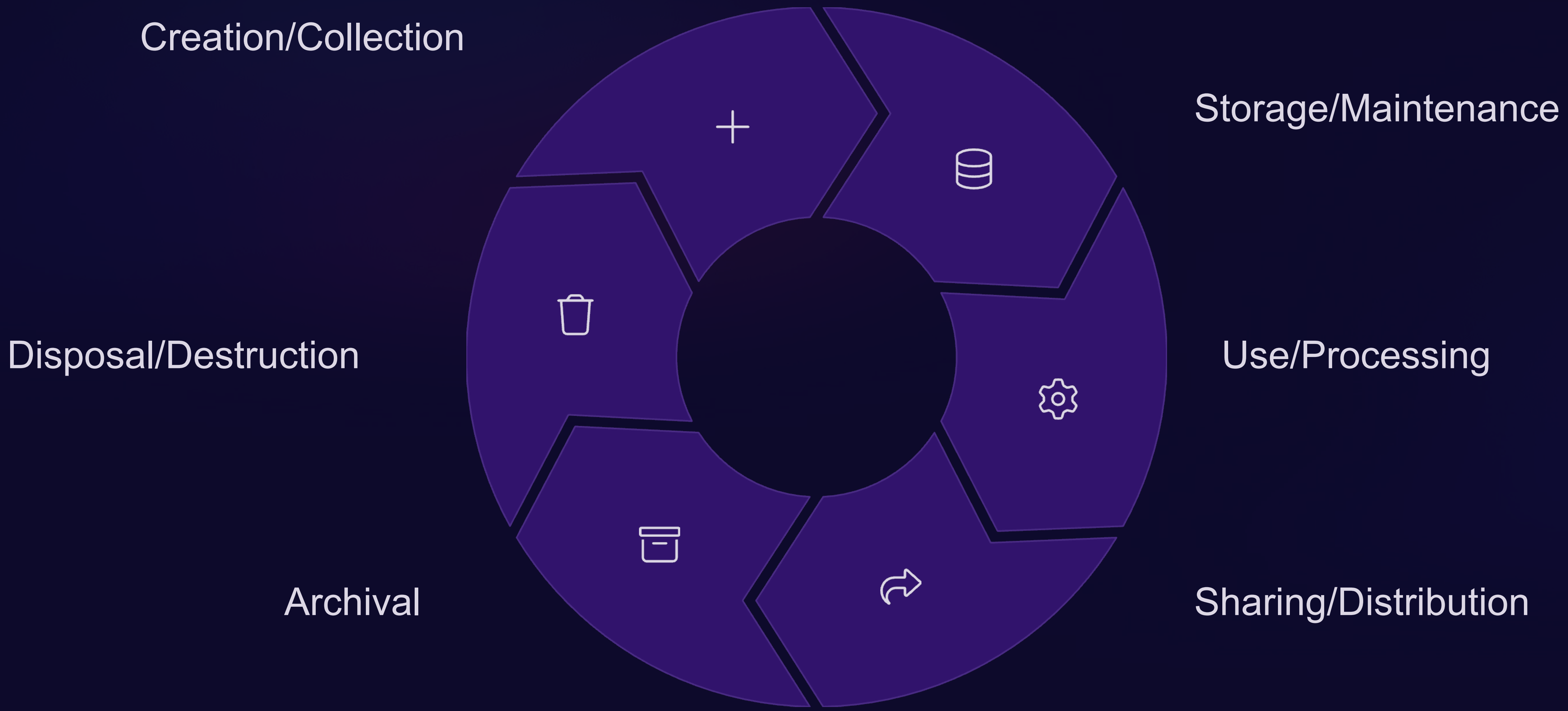
API Response Filtering

- Mask sensitive fields in API responses
- Use different API endpoints for different access levels

Part 3: Data Lifecycle Management - Retention & Disposal

The Data Lifecycle

Every piece of data goes through distinct phases:



Proper lifecycle management addresses security, compliance, and efficiency at each phase.

3.1 Data Retention Policies

Why Retention Matters:

- **Legal compliance:** Laws require certain data be kept for specific periods
- **Risk reduction:** Unnecessary data increases breach risk and storage costs
- **Litigation management:** Supports or defends legal claims

Developing Retention Requirements

Step 1: Identify Legal Requirements

Research regulations applicable to your industry and data types:

Financial Records

- Tax records: 7 years (IRS)
- Accounting records: 7 years
- Payroll records: 4 years (FLSA)

Employment Records

- Personnel files: 3 years after termination
- I-9 forms: 3 years after hire or 1 year after termination (whichever is later)
- Benefits records: 6 years (ERISA)

Healthcare (HIPAA)

- Medical records: 6 years from creation or last use
- State laws may require longer (some states: lifetime + 10 years)