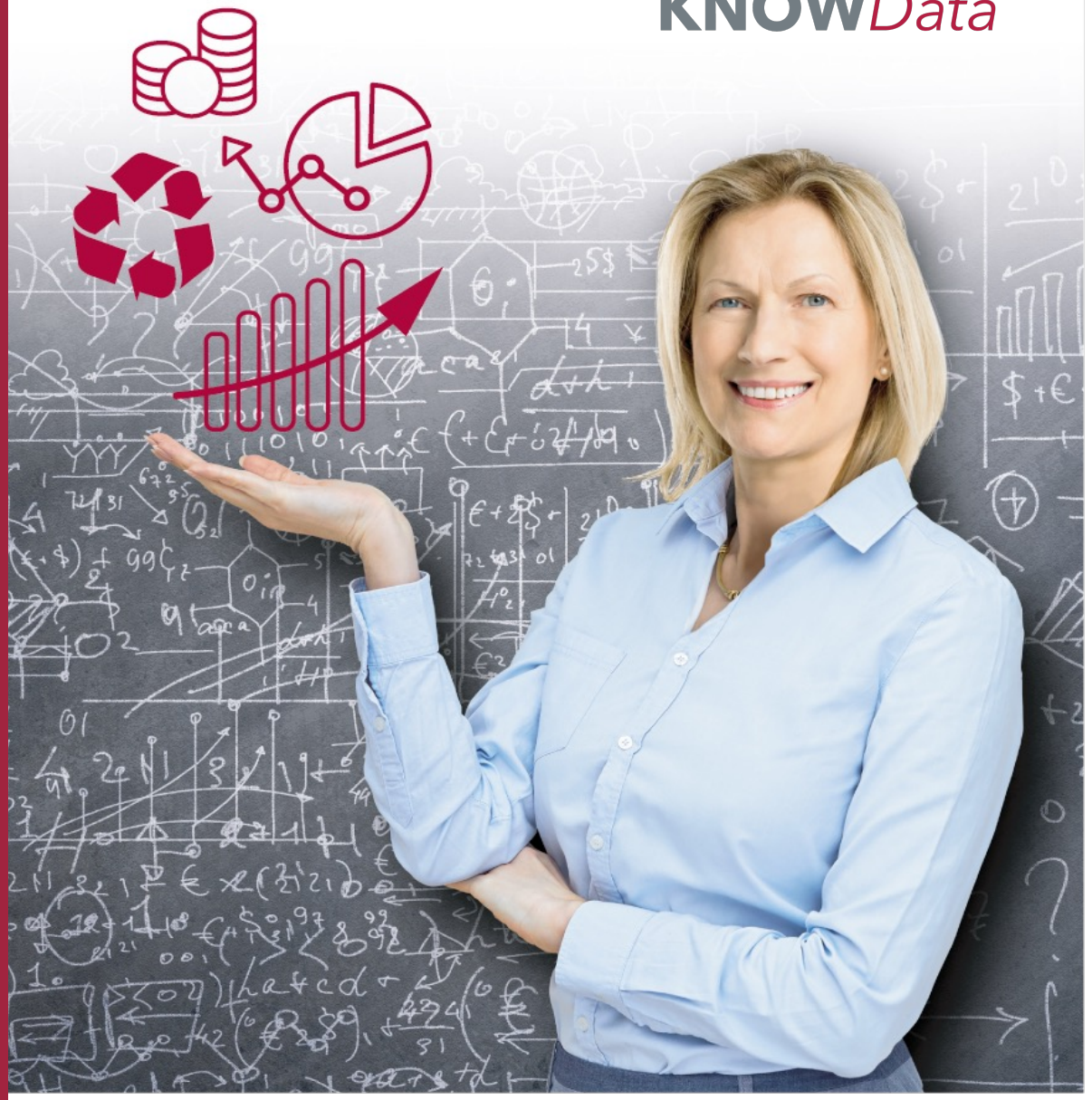
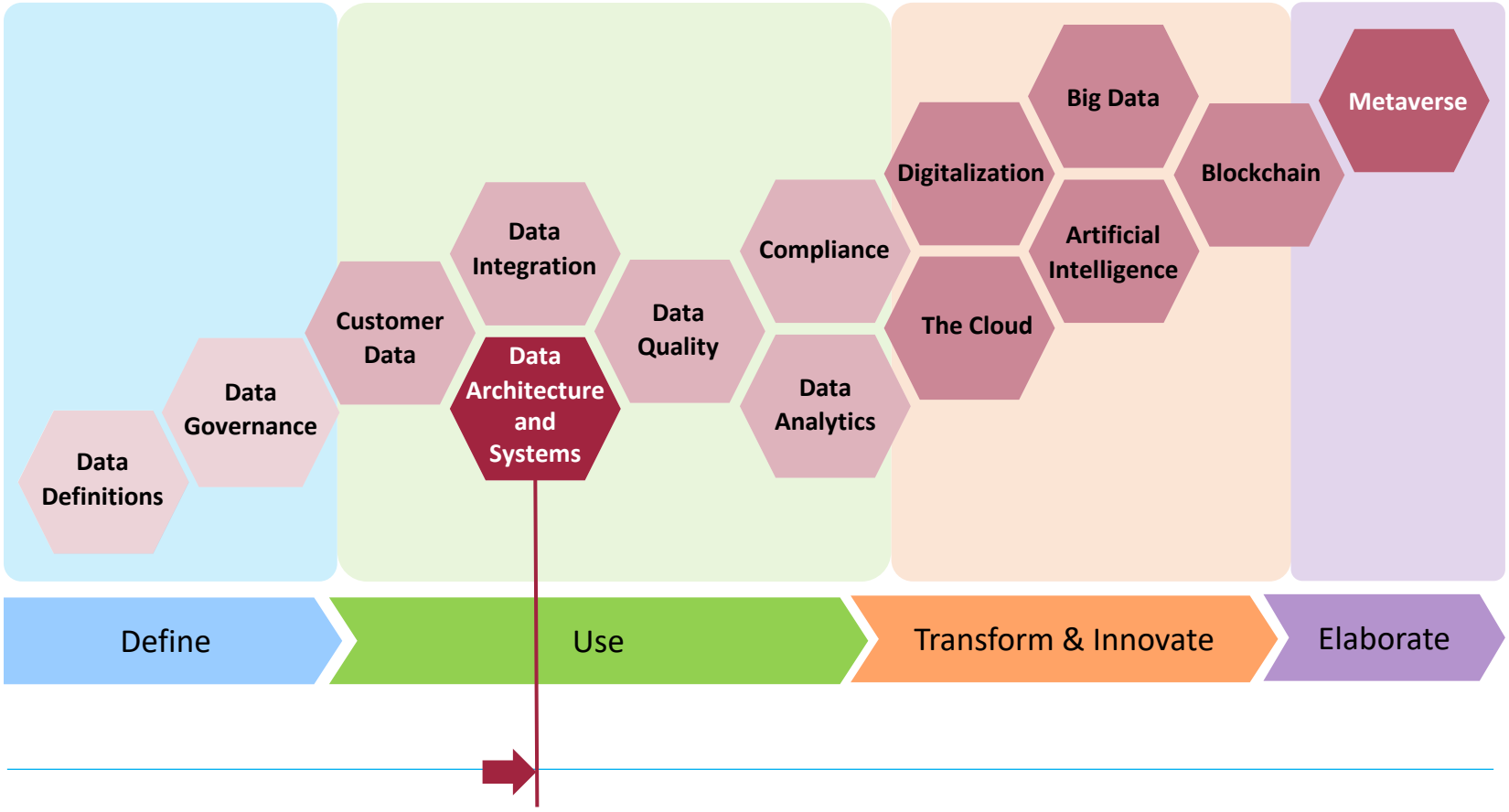
Data Architecture and Systems



KNOWData

May 10, 2026

Lionel Pilorget


Data Definitions
Data Governance
Customer Data
Data Integration
Data Architecture and Systems
Data Quality
Compliance
Data Analytics
Digitalization
The Cloud
Big Data
Artificial Intelligence
Blockchain
Metaverse
Define
Use
Transform & Innovate
Elaborate

Structure of the presentation

Data Architecture

Data Landscape

- Data Digital Ecosystem
- Data Modelling
- Data Inventory Data Catalogue
- Data Mesh

The Reality of Data

- Unstructured Data Gaps
- Approaches and Governance
- Solutions and Implementation

DataOps

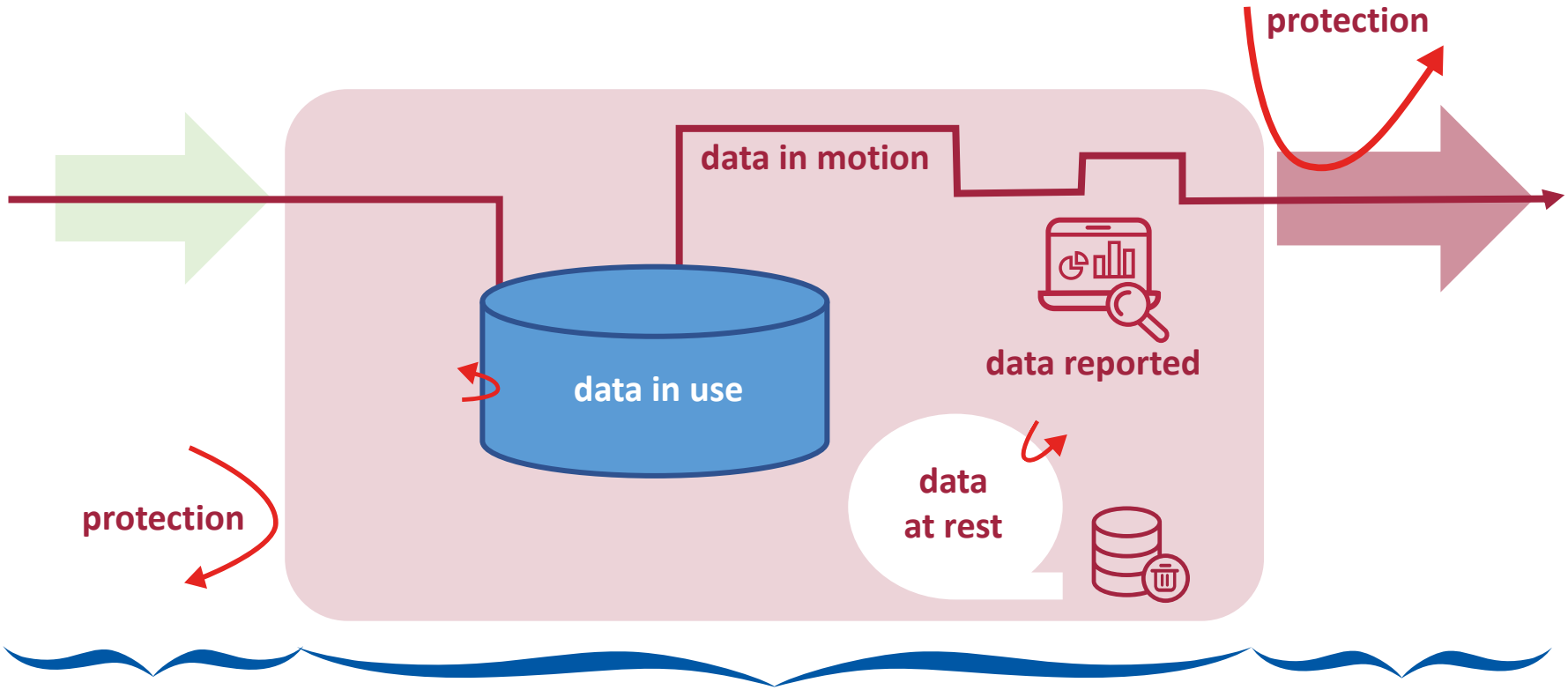
- Store
- Move
- Operate
- Recover

Data Protection

- Data-at-Rest
- Data-in-Motion
- Data-in-Use

The Fate of Data

In Use, in Motion, at Rest



Collect, process, store, protect, analyze and discard

Acquire *Transmit*

Data Digital Ecosystem

A **Data Digital Ecosystem** refers to the interconnected network of **data, systems, processes**, and **stakeholders** within an organization that collectively enable the collection, storage, processing, analysis, and utilization of information in a digital environment.

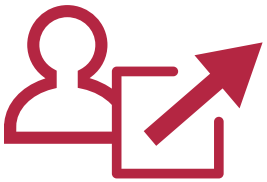
Data Assets

Technologie Infrastructure

SW and Applications

Governance and Security

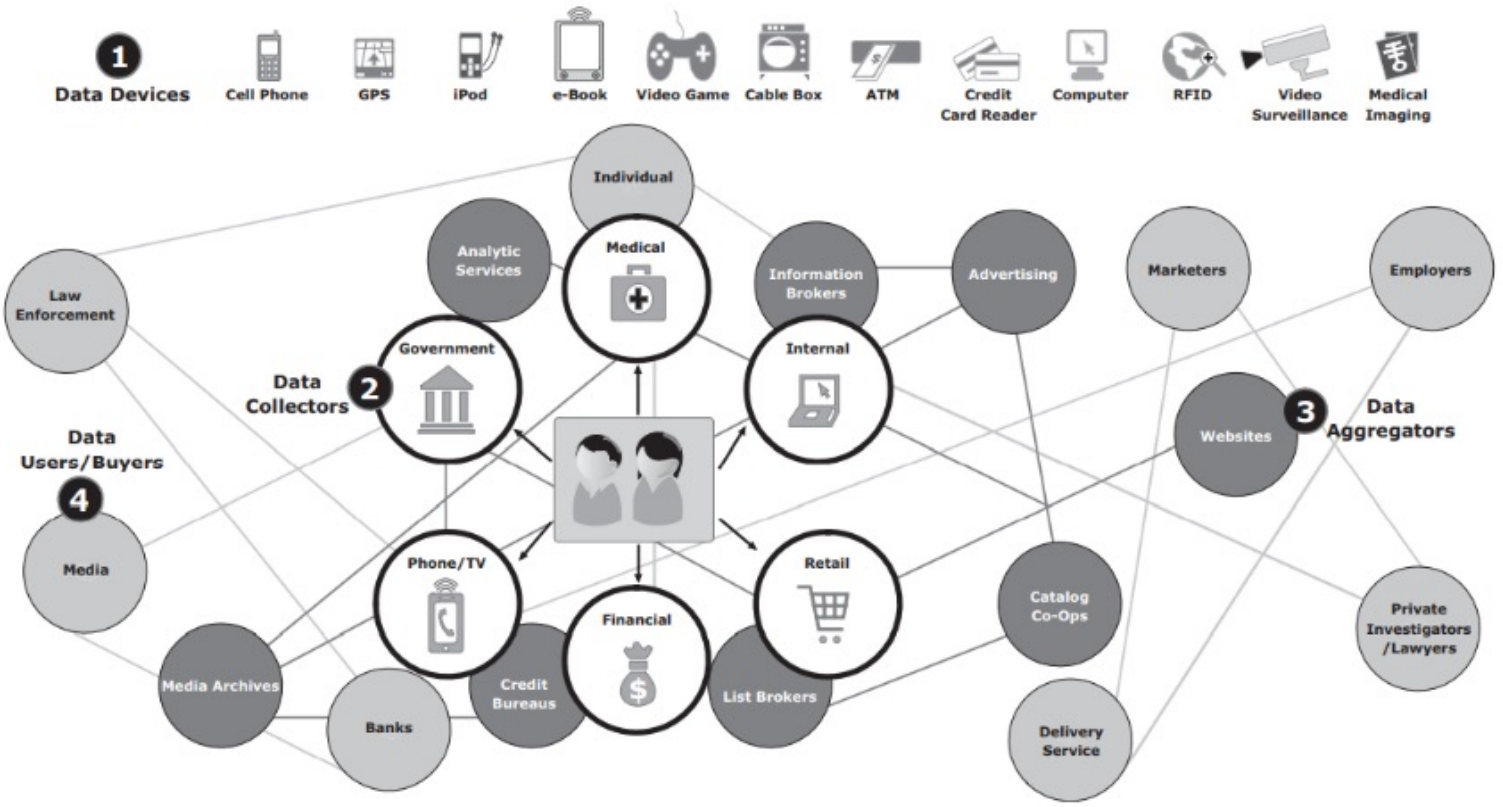
Employees



External partners and Stakeholders

Connected Data Ecosystem

1. Devices that collect data from multiple locations and also generate new data about this data (metadata)
2. Data collectors who gather data from devices and users.
3. Data aggregators that compile the collected data to extract meaningful information
4. Data users and buyers who benefit from the information collected and aggregated by others in the data value chain.



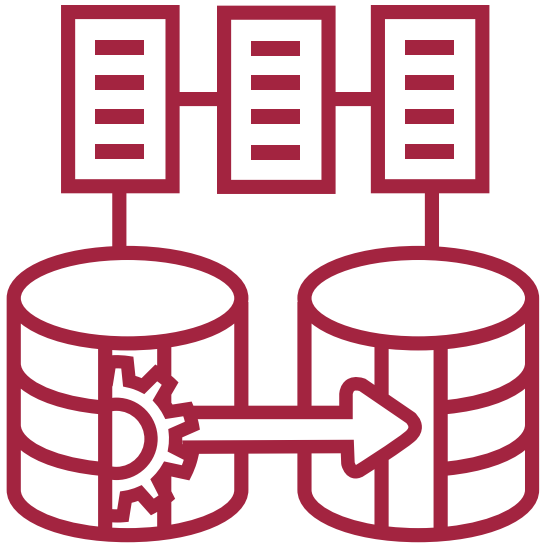
Source: Gnanasundaram S., Shrivastava A., Information Storage and Management

Key Questions about Data to be Answered

Data Identification & Classification	• What are the critical data entities (e.g., customer, product, transaction)? • How is data classified (PII, sensitive, public)? • What metadata is tracked (ownership, lineage)?
Data Governance & Quality	• Who owns and stewards the data? • What are the data quality rules and metrics? • How is data lineage tracked? • What policies enforce compliance?
Data Storage & Location	• Where is data stored (on-prem, cloud, hybrid)? • What storage formats are used (structured, unstructured)? • What are retention & archival policies?
Data Security & Compliance	• How is access controlled (RBAC, encryption)? • How is data anonymized/masked?
Data Usage & Analytics	• Who consumes the data? • What are the SLAs for data access? • How is metadata exposed (data catalogues)?
Scalability & Performance	• How does the system handle growth (sharding, partitioning)? • What are performance benchmarks?
Data Movement & Integration	• Which are data how transferred? • What are the latency requirements (real-time vs. batch)? • How are pipelines monitored?
Cost & Efficiency	• What are the cost drivers (storage, compute)? • How is lifecycle management optimized? • Are there unused datasets to deprecate?

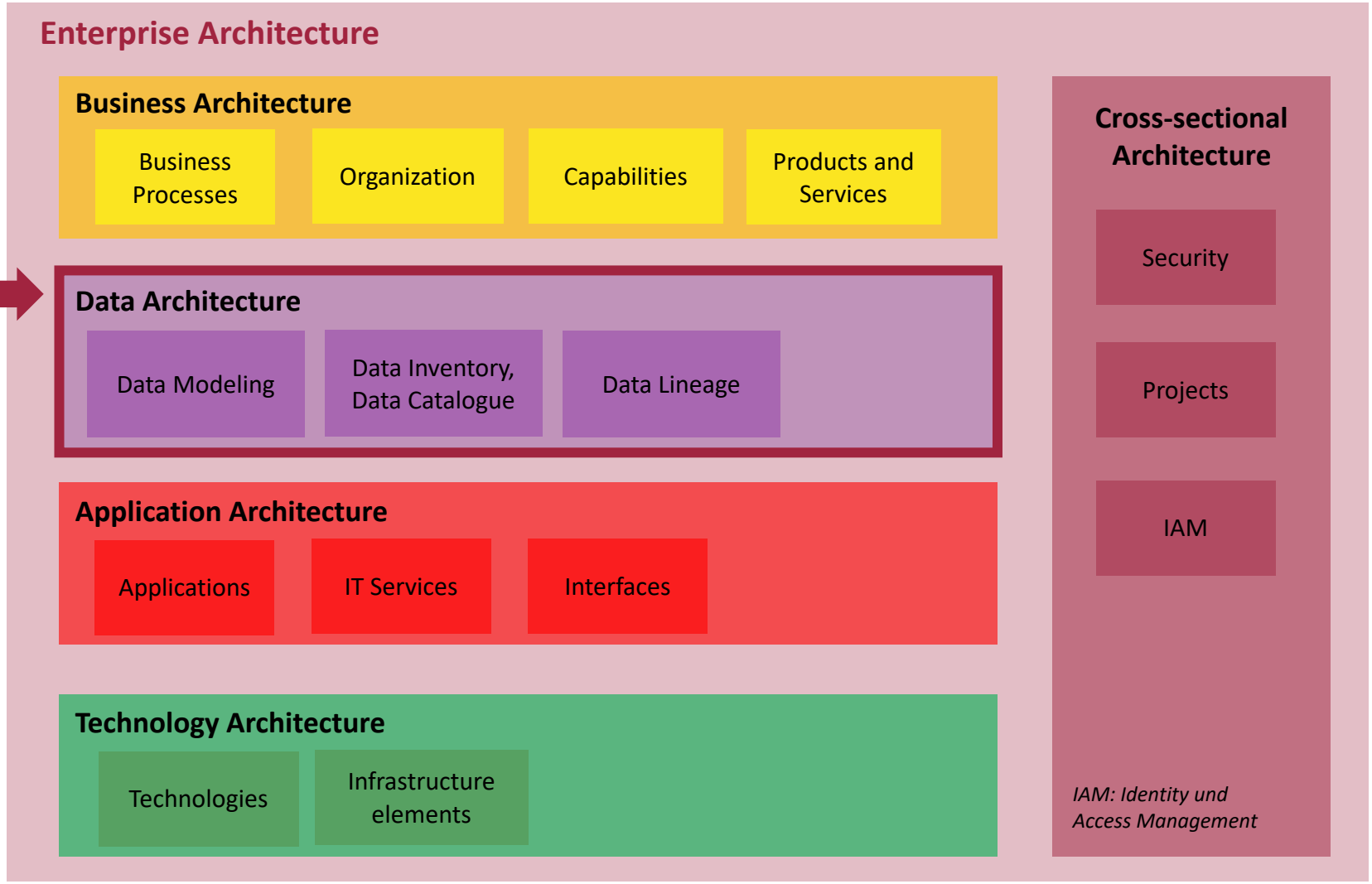
Is Data Architecture the Answer?

Data architecture is the design and development of data systems, models, and structures to store, manage, and access all types of data in a way that meets the business needs.



Data Architecture as Part of Enterprise Architecture

Inspired from the TOGAF Framework



Data Modeling: Identify and Define Business Objects

A business object is a core concept or entity that represents a real-world thing, role, or event relevant to a business process. It can be **tangible** (Product, Customer, Invoice) or **intangible** (Contract, Policy, Event, Agreement).

Understand the Business Context
- What does the organization *do*?
- What are its goals, customers, products, and value streams?

Identify Key Processes
- Look at common activities: e.g., "Sell a product," "Manage a customer," "Deliver a service"
- Use use-case diagrams, value chains, or event storming

Extract Nouns and Roles
- From process descriptions, extract nouns: these are likely business objects -> "A ***customer*** places an ***order*** for a ***product***"
- Identify roles (e.g., supplier, employee) and documents (invoice, quote, report)

Group and Refine
- Group similar objects and eliminate duplicates
- Ask: "Is this object reused across multiple processes?" If yes, it's likely a key business object.

Examples of Business Objects

Business Area	Business Objects
Retail	Customer, Product, Order, Shopping Cart, Payment, Delivery, Review
Banking	Customer, Account, Transaction, Card, Loan, Branch, Statement, Asset
HealthCare	Patient, Appointment, Doctor, Prescription, Treatment, Invoice, Medical Record
Manufacturing	Product, Machine, Supplier, Order, Work Order, Component, Quality Check, Distribution Center
Insurance	Policy holder, Policy, Claim, Premium, Risk, Agent, Contract

Shape the Data Domains

A **data domain** is a logical grouping of related data elements (i.e., business objects) that reflect a specific area of business responsibility or capability

Group Objects by Business Function or Purpose

Ask:
- Which objects "belong together" logically?
- Do they support the same process?
- Are they owned by the same team or department?

Define each Domain

For each group:
- Give it a clear name (e.g., "Customer Domain")
- Describe its purpose
- List its business objects
- Note relationships with other domains

Identify Cross-Domain Relationships

- E.g., an Order in the "Sales Domain" references a Customer in the "Customer Domain"

Validate against Business Structure

Ask:
- Do the domains map to actual teams, capabilities, or departments?

Examples of Data Domains

Domain	Purpose	Key Business Objects
Customer	Manage customer profiles and preferences	Customer, Address, Loyalty Card
Product	Manage product catalogue and categories	Product, Category, Inventory
Sales Orders	Track sales and transactions	Order, Payment, Invoice
Fulfilment	Handle shipping and delivery	Delivery, Warehouse, Carrier
Marketing	Promotions and engagement	Campaign, Promotion, Coupon

Domain	Purpose	Key Business Objects
Customer	Manage client data	Customer, Address, Identity
Account	Banking services and financial products	Account, Transaction, Statement
Credit	Manage loans and credit lines	Loan, Interest Rate, Credit Limit
Asset	Manage owned and traded financial assets	Portfolio, Security, Investment, Asset Valuation
Compliance	Regulatory and audit controls	KYC Record, Risk Rating, Audit Log

Domain	Purpose	Key Business Objects
Patient	Personal and medical history	Patient, Contact, Insurance
Appointment	Scheduling and check-in	Appointment, Clinic, Doctor
Treatment	Clinical services and results	Treatment, Diagnosis, Prescription
Billing	Financial operations	Invoice, Payment, Insurance Claim

Domain	Purpose	Key Business Objects
Product Design	Develop and maintain product structure	Product, Component, BOM (Bill of Materials)
Production	Manage manufacturing operations	Work Order, Machine, Shift, Line
Supply Chain	Handle procurement and logistics	Supplier, Purchase Order, Delivery
Quality Control	Ensure product standards	Inspection, Defect, Test Report

Conceptual Data Model

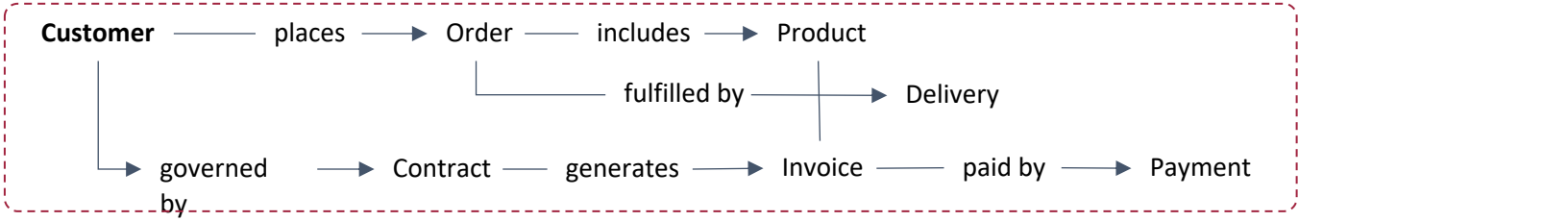
Generic Model

Entities

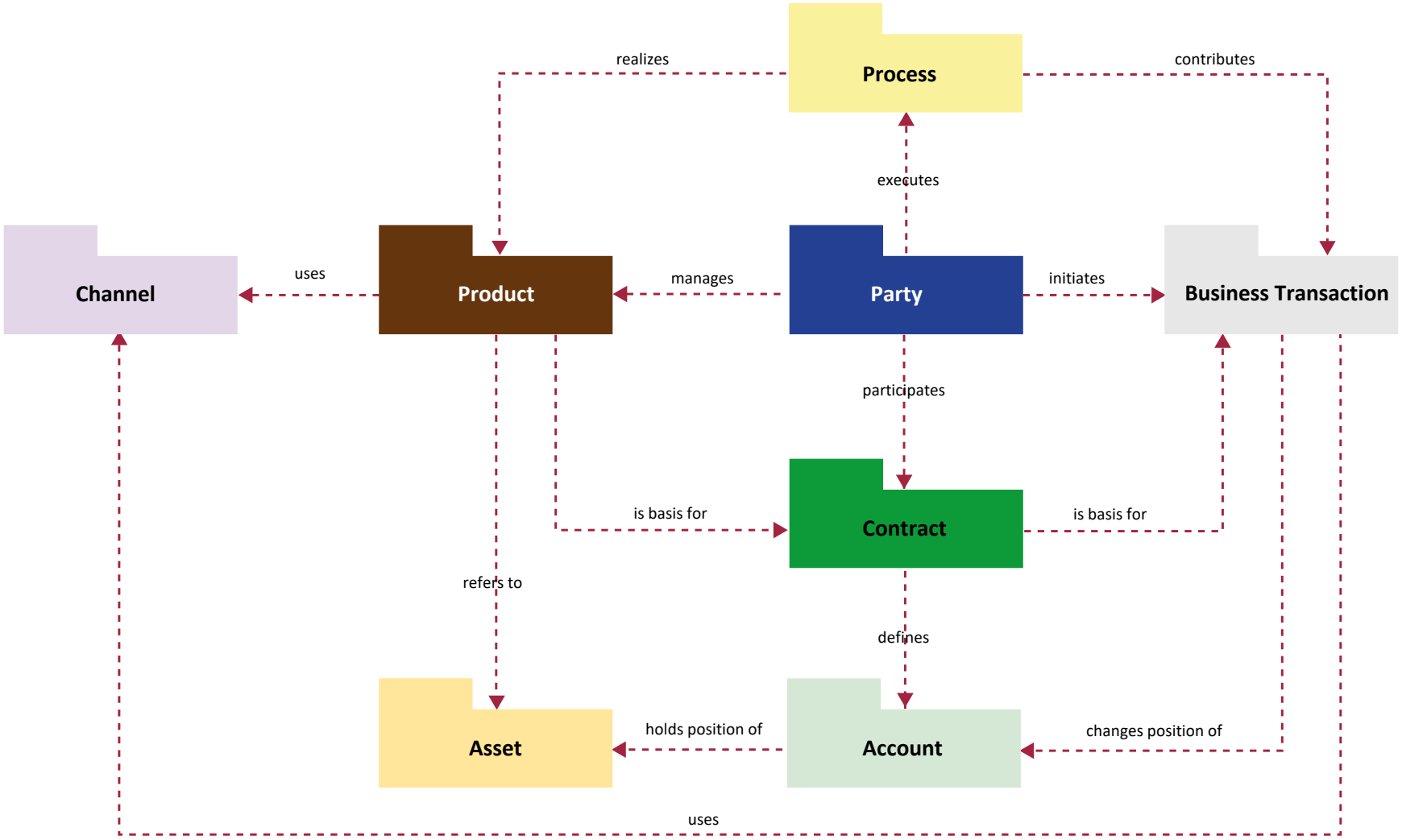
- Customer
- Product / Service
- Order / Transaction
- Invoice / Payment
- Employee / Agent
- Supplier / Partner
- Contract / Agreement
- Delivery / Fulfilment
- Campaign / Promotion

Relationships

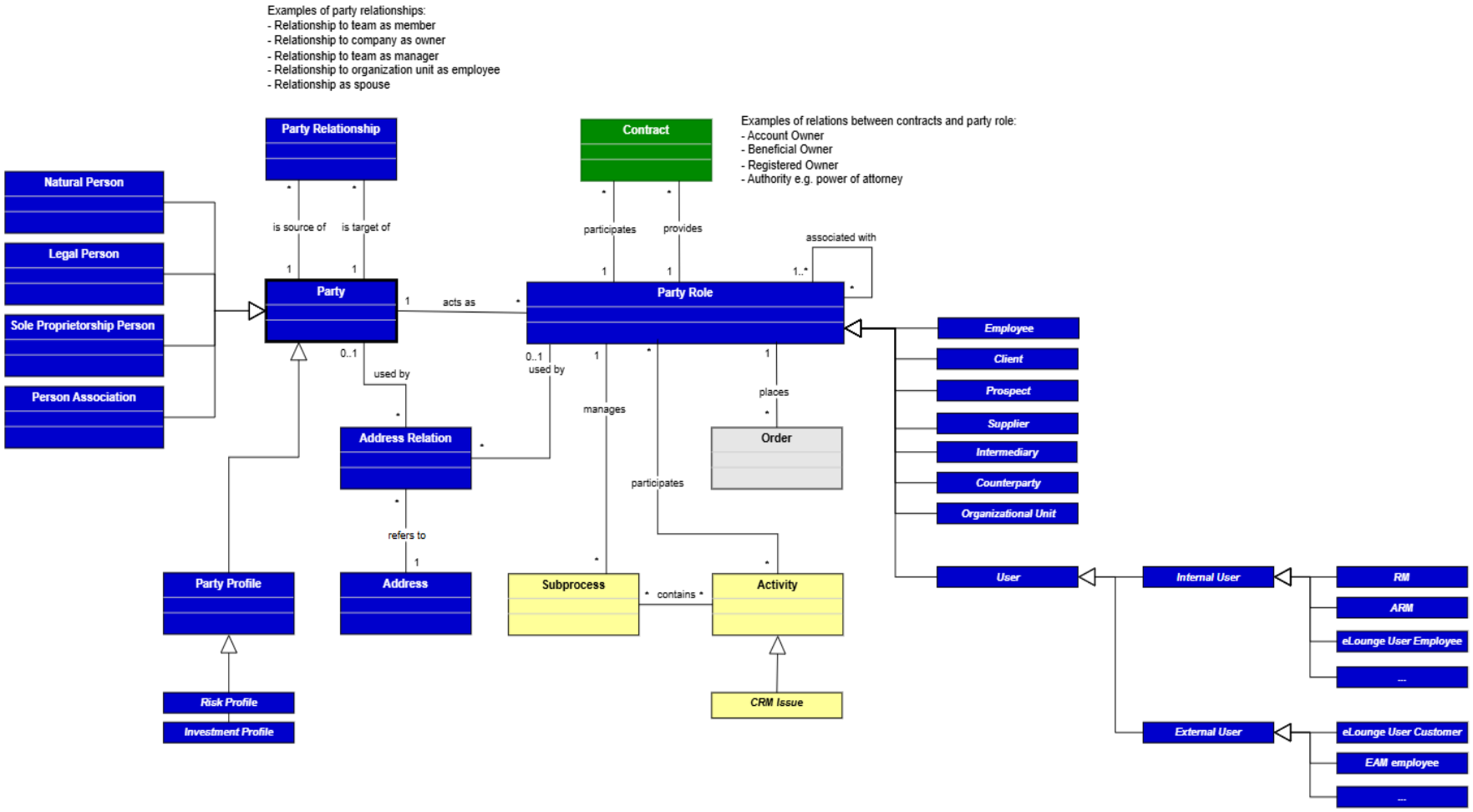
- A Customer places one or more Orders
- An Order includes one or more Products
- An Order generates an Invoice
- A Customer makes a Payment for an Invoice
- An Employee or Agent handles an Order or Customer
- A Supplier provides Products
- A Campaign promotes Products
- A Contract governs the relationship with a Customer or Supplier
- A Delivery fulfils an Order



Example of Data Modeling in Banking



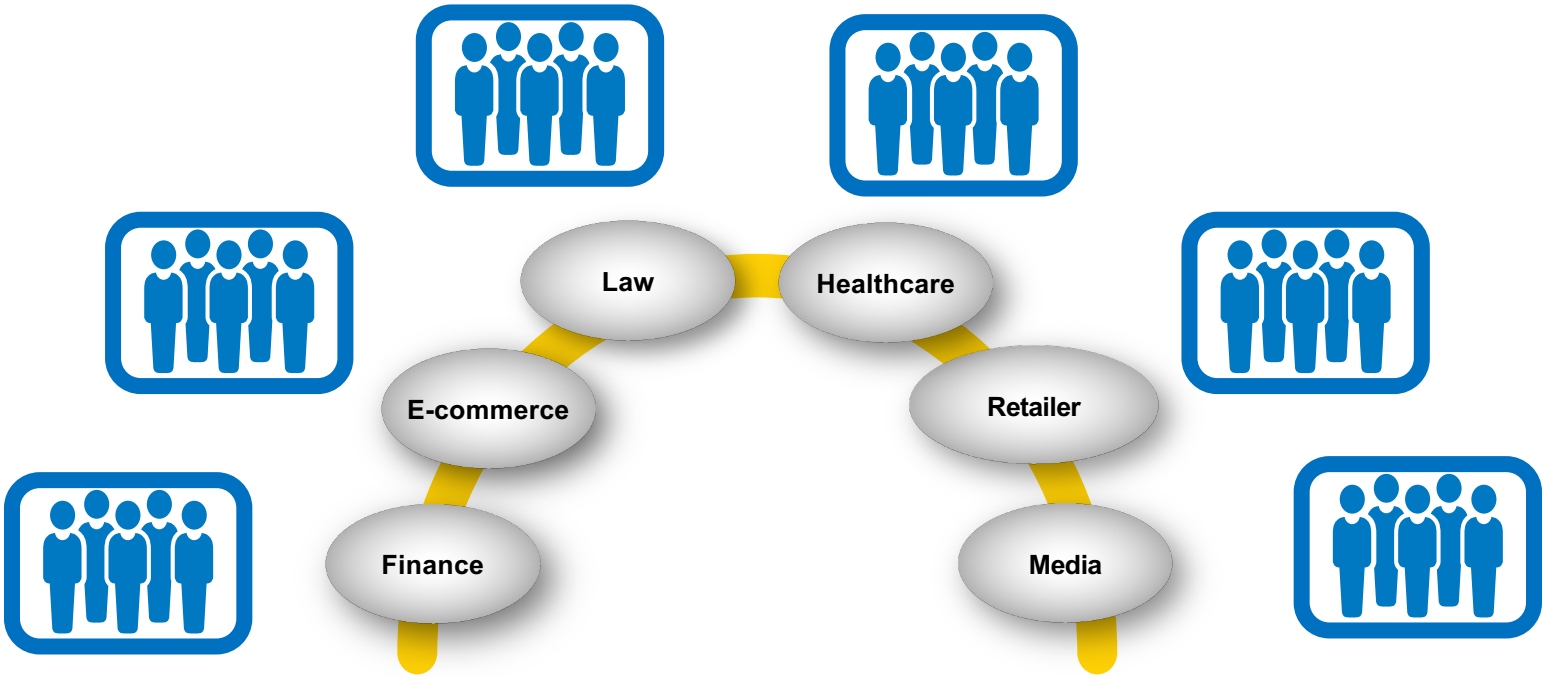
Detailed Data Modeling for the Object “Party”



Design a Conceptual Data Model

Data architecture model for the following business

- Financial Institution
- E-commerce Platform
- Law firm
- Healthcare services
- Global retailer
- Media company



Data Inventory

What data do we have, and where are they stored?

A **data inventory** is a **comprehensive, high-level listing of all datasets** within an organization, including their location, format, owner, and purpose.

Tools can be used to automate the data inventory process, ensuring accurate and up-to-date records of the data assets.

- Location
- Format
- Owner
- Purpose
- Criticality score

- Metadata
- Dependencies
- Update Frequency
- Volume
- Classification

Why is Data Inventory important?

Data Inventory:

- Helps to comply with **data protection regulations** by providing a clear understanding of what data is collected & how it is used
- Allows for **better management of data assets**, ensuring that data is organized, accessible & maintained properly, which can enhance operational efficiency
- By identifying and categorizing **sensitive data**, a data inventory helps to assess potential risks & implement appropriate security measures to protect against data breaches
- Helps **identifying opportunities** for data integration across departments, leading to more cohesive data strategies and **improved analytics capabilities**

Data Catalogue

What data are available, what does it mean, and how can I use it?

A **Data Catalogue** is a **centralized, dynamic and searchable repository** that organizes, enriches, and manages metadata about data assets, enabling users to discover, understand, and trust the data they work with.

A Data Catalogue:

- Provides detailed information about data sources, including data types, formats, lineage, & usage statistics, which helps users assess data quality & relevance.
- Enhances data discoverability by offering advanced search capabilities, allowing users to quickly locate datasets based on keywords, tags, or other attributes.
- Includes features for collaboration, such as user comments, ratings & data stewardship roles, fostering a culture of shared knowledge & best practices within organizations.
- Can integrate with various data sources & tools, enabling seamless access to data across different platforms, enhancing data governance and compliance efforts.

Domain-Centric Data Catalogue

Organizing the catalogue around data domains as self-contained units, each with domain team ownership and consistent metadata standards

Data Domain	Key Business Objects	Attributes (Examples)	Data Catalogue Items	Example Value
Sales	Order	order_id (UUID), customer_id (FK), order_date, total_amount	Orders API (REST endpoint) Sales Daily Report (CSV)	{"order_id": "a1b2c3", "total_amount": 99.99}
	Customer	customer_id, name, email, loyalty_tier	Customer Master (DB table), Customer Segmentation (Dashboard)	{"name": "John Doe", "loyalty_tier": "Gold"}
Inventory	Product	product_id, name, price, stock_quantity, supplier_id	Product Catalogue (API), Low Stock Alerts (Kafka topic)	{"name": "Wireless Headphones", "stock_quantity": 42}
	Warehouse	warehouse_id, location, capacity	Warehouse Locations (GIS dataset)	{"location": "Berlin", "capacity": "5000sqft"}
Finance	Payment	payment_id, order_id, amount, status	Payments API, Revenue Forecast (Power BI)	{"amount": 99.99, "status": "Completed"}
	Invoice	invoice_id, due_date, tax_amount	Invoice PDFs (S3 bucket)	{"due_date": "2024-05-30", "tax_amount": 19.99}
Marketing	Campaign	campaign_id, budget, start_date, channel	Campaign Performance (Google Analytics link)	{"budget": 5000, "channel": "Social Media"}

Relevance of a Data Catalogue

A **Data Catalogue**:

- Helps to maintain control over data assets by providing visibility into data lineage, ownership & compliance, ensuring that data is used responsibly & ethically
- Helps users to understand the quality & reliability of data, promoting consistent usage across the organization by providing metadata & context
- Allows employees to easily find & access the data they need, reducing time spent searching for information & increasing overall productivity
- Supports data analysts & scientists by providing them with the necessary resources to conduct analyses more effectively, leading to more accurate insights & informed business strategies

Key differences between a Data Inventory and a Data Catalogue

Topic	Data Inventory	Data Catalogue
Purpose	Basic listing of all data assets in an organization	Enriched metadata platform to organize, search, and manage data
Scope	Broad and high-level (includes all data sources)	Focused on discoverable, usable, and curated data sets
Detail Level	Often minimal: name, location, owner	Rich: metadata, lineage, profiling, tags, usage, quality
Users	IT, compliance, data governance	Data analysts, data scientists, business users, data engineers
Metadata Included	Basic metadata (name, type, source, owner)	Technical + business metadata, data lineage, classification
Interactivity	Typically, static or updated periodically	Dynamic, searchable, with collaboration features
Tooling	Often manual (spreadsheets, basic tools)	Specialized tools (e.g., Alation, Collibra, Google Data Catalogue)

Definition of Data Lineage

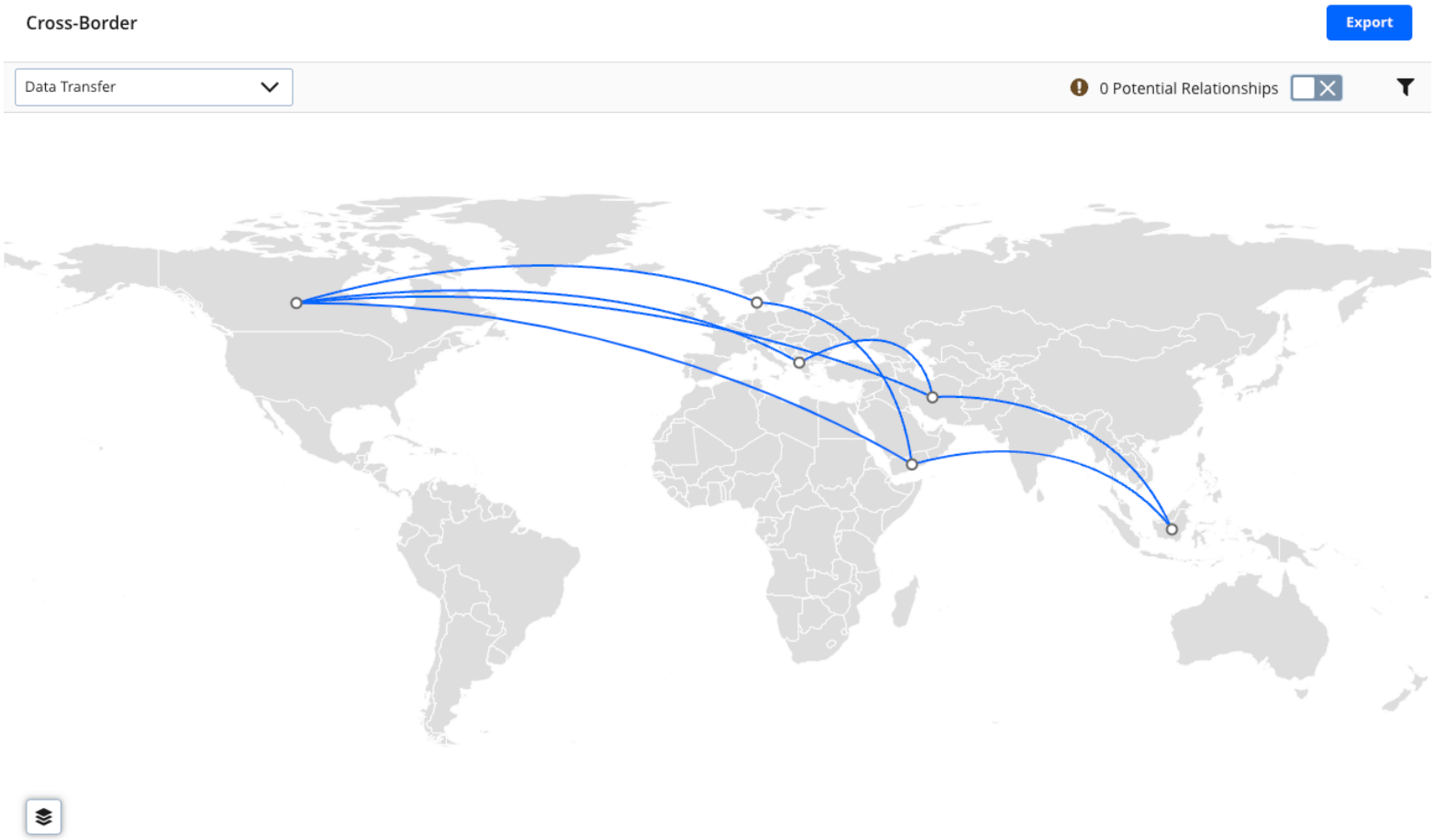
Data lineage is the record of the flow, transformation, and lifecycle of data as it moves from its original source to its final destination.

It shows where data comes from, how it changes, and where it goes, often in visual or traceable form.

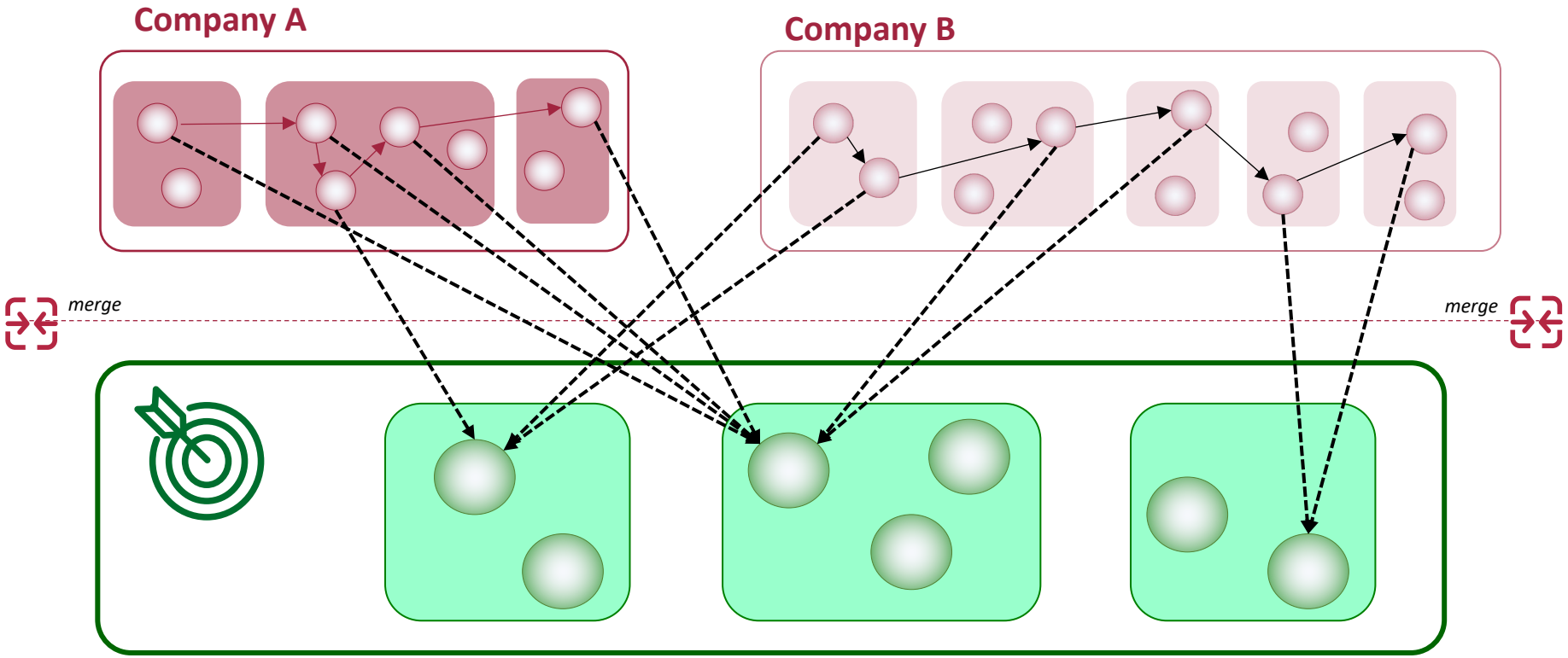
Data lineage is essential for:

- ✅ **Trust & Transparency** – Understand how data was generated or transformed
- ✅ **Impact Analysis** – Know what downstream systems are affected by changes
- ✅ **Debugging & Troubleshooting** – Trace errors back to their origin
- ✅ **Compliance & Auditing** – Prove how data complies with regulations (e.g., GDPR)
- ✅ **Data Governance** – Enable responsible data stewardship and quality control

Data Lineage Use Case 01:
Visualizing Cross-border Data Transfers

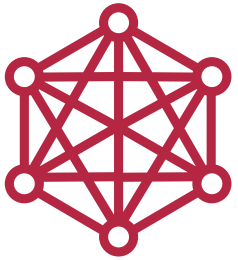


Data Lineage Use Case 02:
System Consolidation for Mergers & Acquisitions



Data Mesh

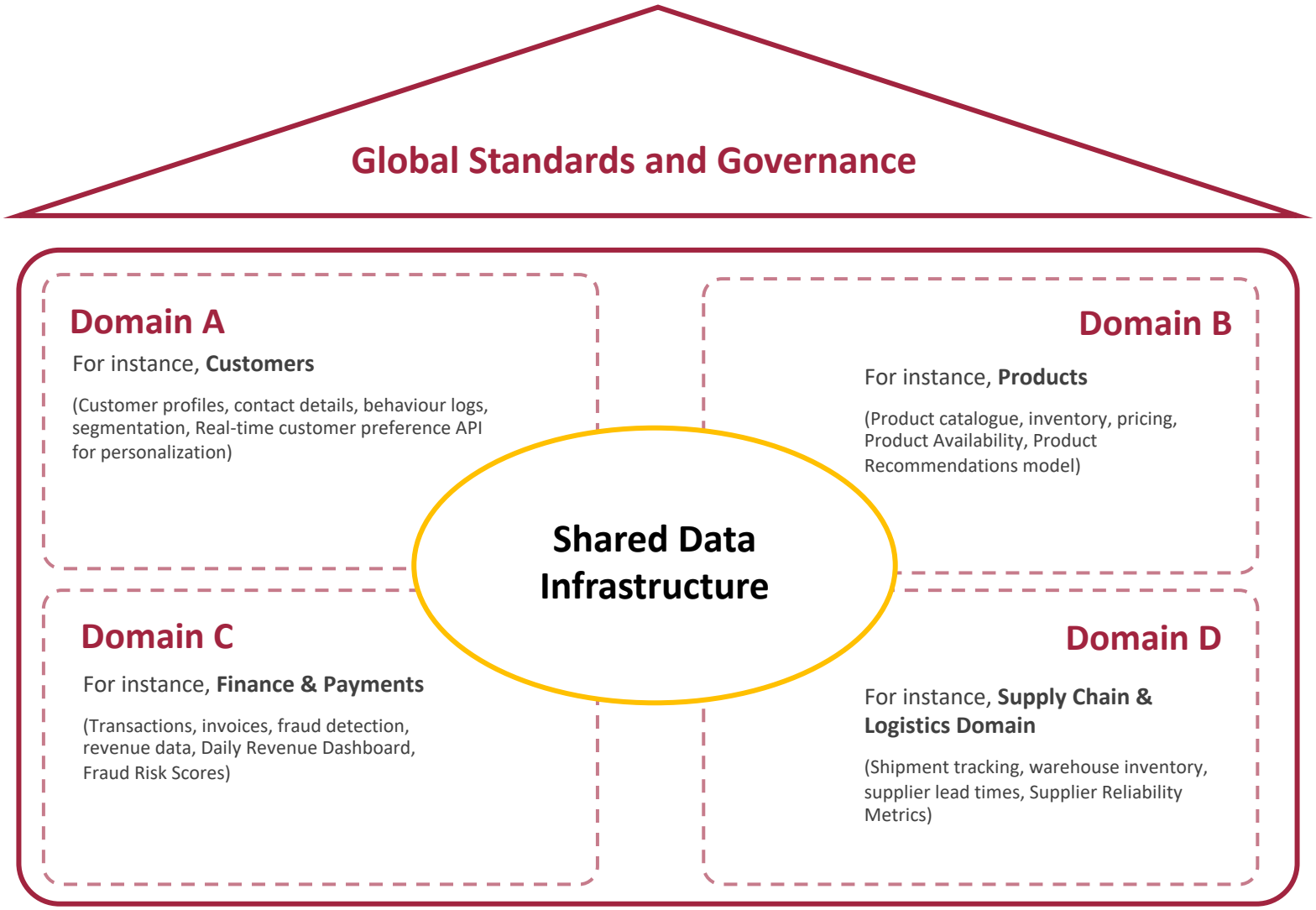
Data Mesh is a decentralized, domain-oriented approach to data architecture and organizational design that treats **data as a product**, emphasizing ownership, scalability, and self-service access. It shifts away from monolithic data lakes or warehouses by distributing responsibility to domain-specific teams (e.g., marketing, finance) while maintaining global interoperability.



Key Principles of Data Mesh

1. **Domain-Oriented Ownership**
   - Data responsibility is distributed to the teams that are closest to the data (e.g., marketing, sales, operations).
   - These domain teams are responsible for producing, maintaining, and serving their data.
2. **Data as a Product**:
   - Each dataset is treated like a product with dedicated owners, clear documentation, quality standards, and service-level agreements (SLAs).
   - Users (analysts, data scientists, etc.) are treated as customers of the data.
3. **Self-Serve Data Infrastructure**:
   - A shared platform provides tools and services (e.g., ingestion, processing, security, governance) to enable domain teams to manage their data autonomously.
4. **Federated Computational Governance**:
   - Governance is not centralized but coordinated across domains, using shared standards and policies. It ensures data security, privacy, and compliance while preserving agility.

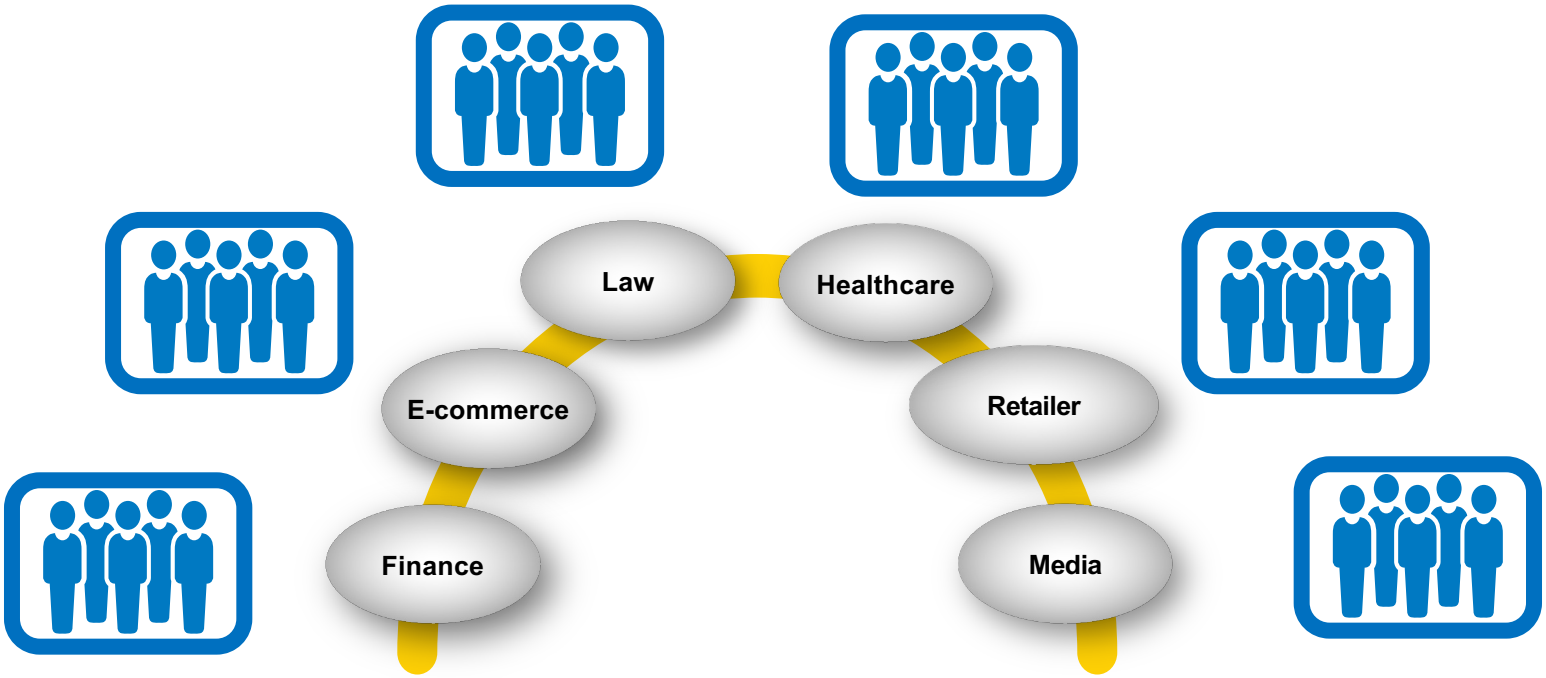
High Level Architecture of Data Mesh



Define Data Mesh

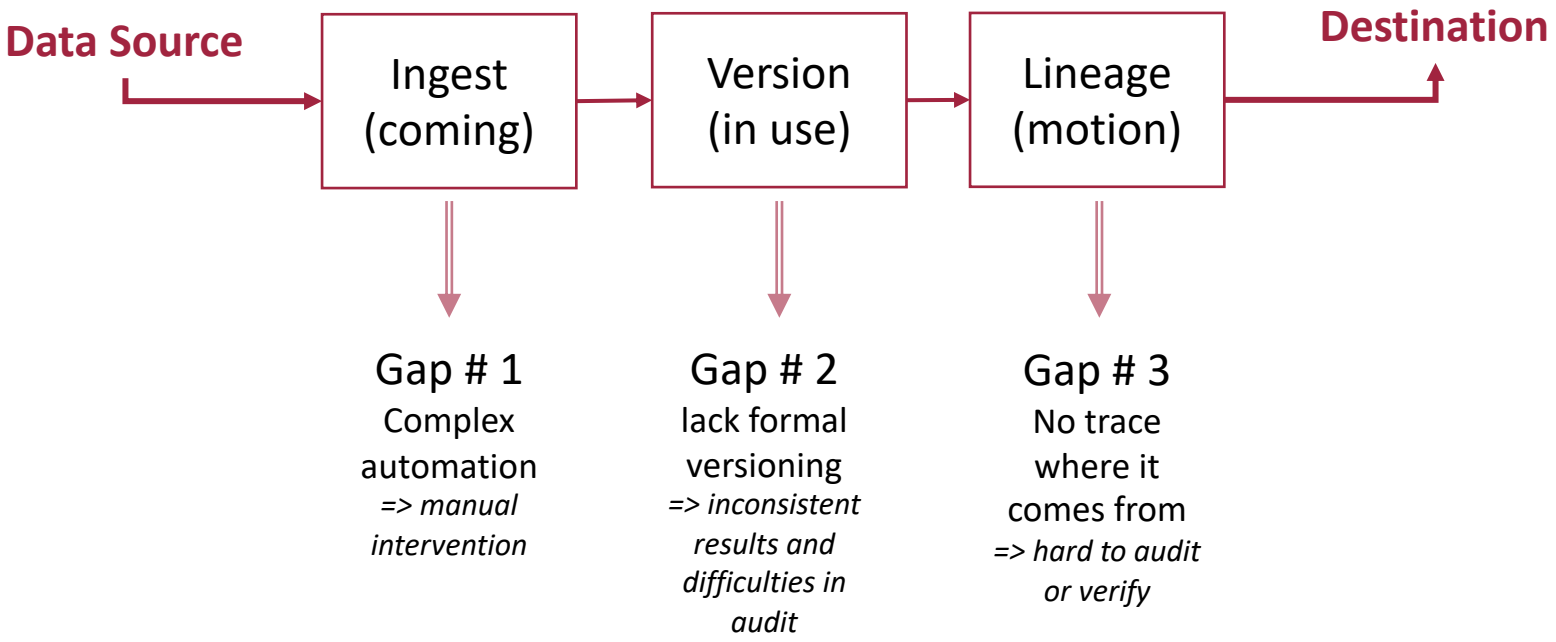
For the following businesses

- Financial Institution
- E-commerce Platform
- Law firm
- Healthcare services
- Global retailer
- Media company



The Reality of Data: Unstructured Data Gaps

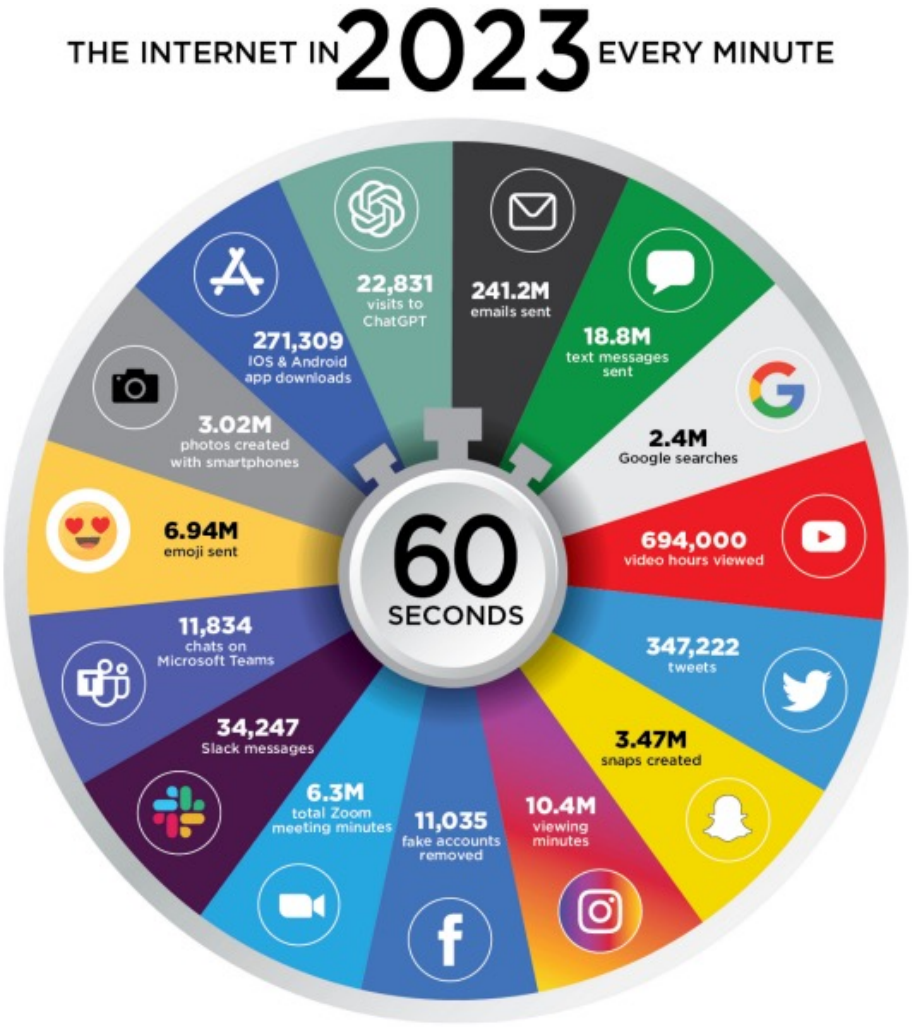
The Three Fundamental Gaps: Ingestion, Versioning, Lineage



- ✓ Structured data flow smoothly through all three stages
- ≠ Unstructured data break at every stage

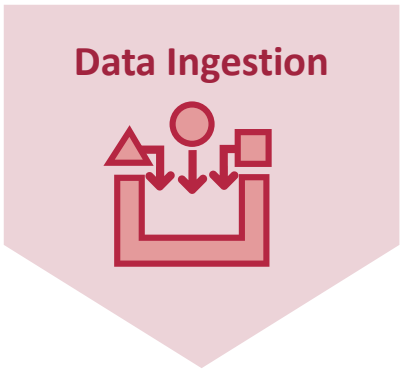
Where Do Unstructured Data Live ?

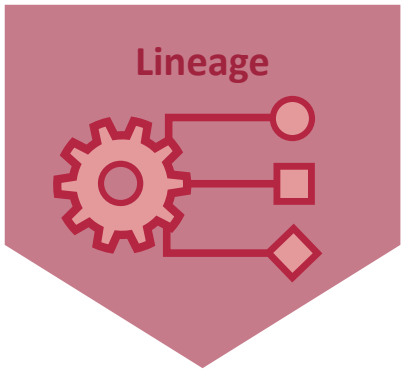
The Scale of Unstructured Data



Source: ediscoverytoday.com/wp-content/uploads/2023/04/2023-Internet-Minute-Infographic-Source-Stats.pdf/

How Do Enterprises Close the Gaps?





Automation

Version Policies

Provenance Tracking

Audit Trails

ECM: Enterprise Content Management

DMS: Document Management System

Why Version Control Alone Fails?

Five fundamental limitations

- **Lack of Standardized Structure**
  - Unstructured data has no fixed schema → difficult to diff, merge, or track changes automatically.
  - Example: Word documents or PDFs cannot be versioned like code files
- **Large File Sizes**
  - Media files, datasets, or archives are too big for traditional version control systems.
  - Storing multiple versions consumes storage and slows operations
- **Complex Metadata & Context**
  - Meaning of changes depends on context (e.g., business process, author, workflow state)
  - Standard VCS cannot capture metadata like approvals, document lifecycle, or business rules
- **Collaboration Complexity**
  - Multiple users may work on overlapping content in parallel (documents, presentations)
  - Conflicts are hard to resolve automatically without clear workflows
- **Limited Tooling Support**
  - Most VCS tools (Git, SVN) are optimized for text/code, not rich documents or multi-format content
  - Lack of integration with workflows and governance systems

Best Practices for Unified Version Control

Control vs. Speed

Aspect	Centralized Versioning	Distributed Versioning
Model	Single central repository (e.g., SVN, SharePoint)	Decentralized clones (e.g., Git, Mercurial)
Collaboration	Lock-modify-unlock (serialized changes)	Branch-merge (parallel workflows)
Offline Work	Limited (requires server connection	Fully supported (local commits)
Use Cases	• Legal documents with strict approvals • Monolithic data systems	• Code/data science projects • Agile content teams

Centralized

Centralized versioning fits traditional workflows where control, compliance, and access management are key

Distributed

Distributed models suit agile, tech-heavy teams needing speed, autonomy, and flexible branching

Enterprise Content Management (ECM) Overview

Enterprise content management, a form of content management, combines the capture, search and networking of documents with digital archiving, document management and workflow. It includes the challenges involved in using and preserving a company's internal (often unstructured) information in all its forms.
Most ECM solutions focus on business-to-employee (B2E) systems.

Source: en.wikipedia.org/wiki/Enterprise_content_management

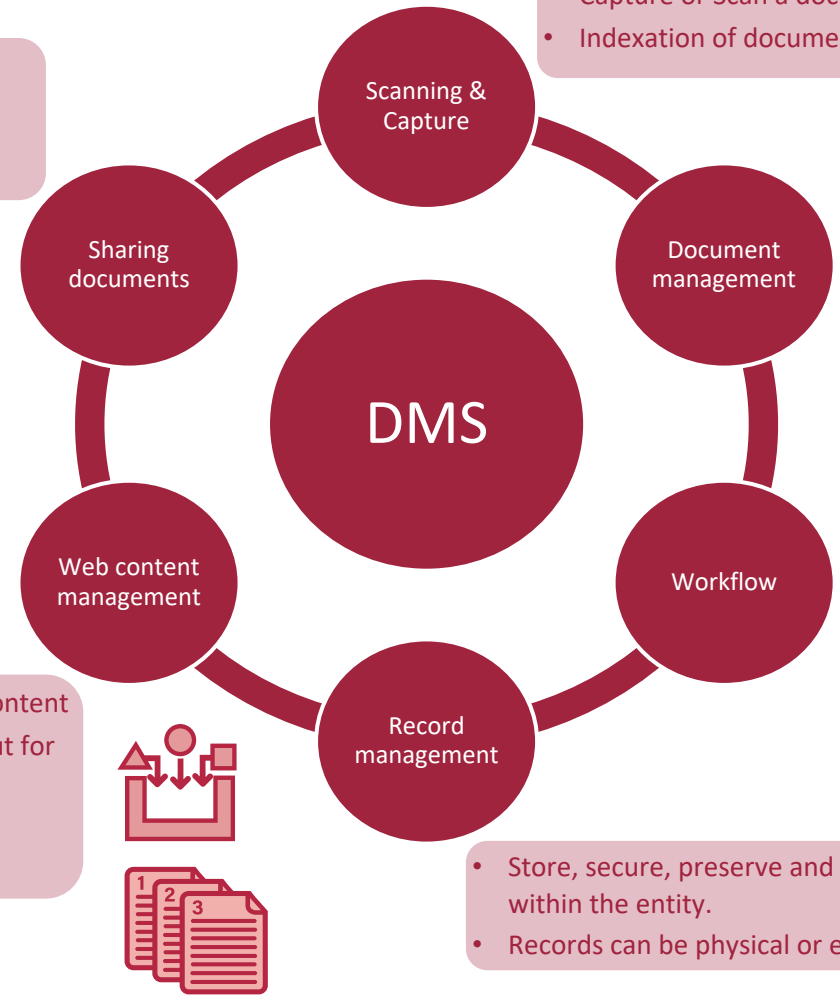
Document Management Solution (DMS) Explained

A Document Management System (DMS) is usually a computerized system used to store, share, track and manage files or documents. Some systems include history tracking where a log of the various versions created and modified by different users is recorded. The term has some overlap with the concepts of content management systems. It is often viewed as a component of enterprise content management (ECM) systems and related to digital asset management, document imaging, workflow systems and records management systems.

Source: en.wikipedia.org/wiki/Document_management_system

DMS Functionalities Mapped to Data Gaps

How scanning, versioning, workflows, and records close the gaps



- Capture or Scan a document
- Indexation of documents

- Share and access documents throughout the entity
- Sharing can be on-site or at distance

- store and keep track of different versions modified by different users

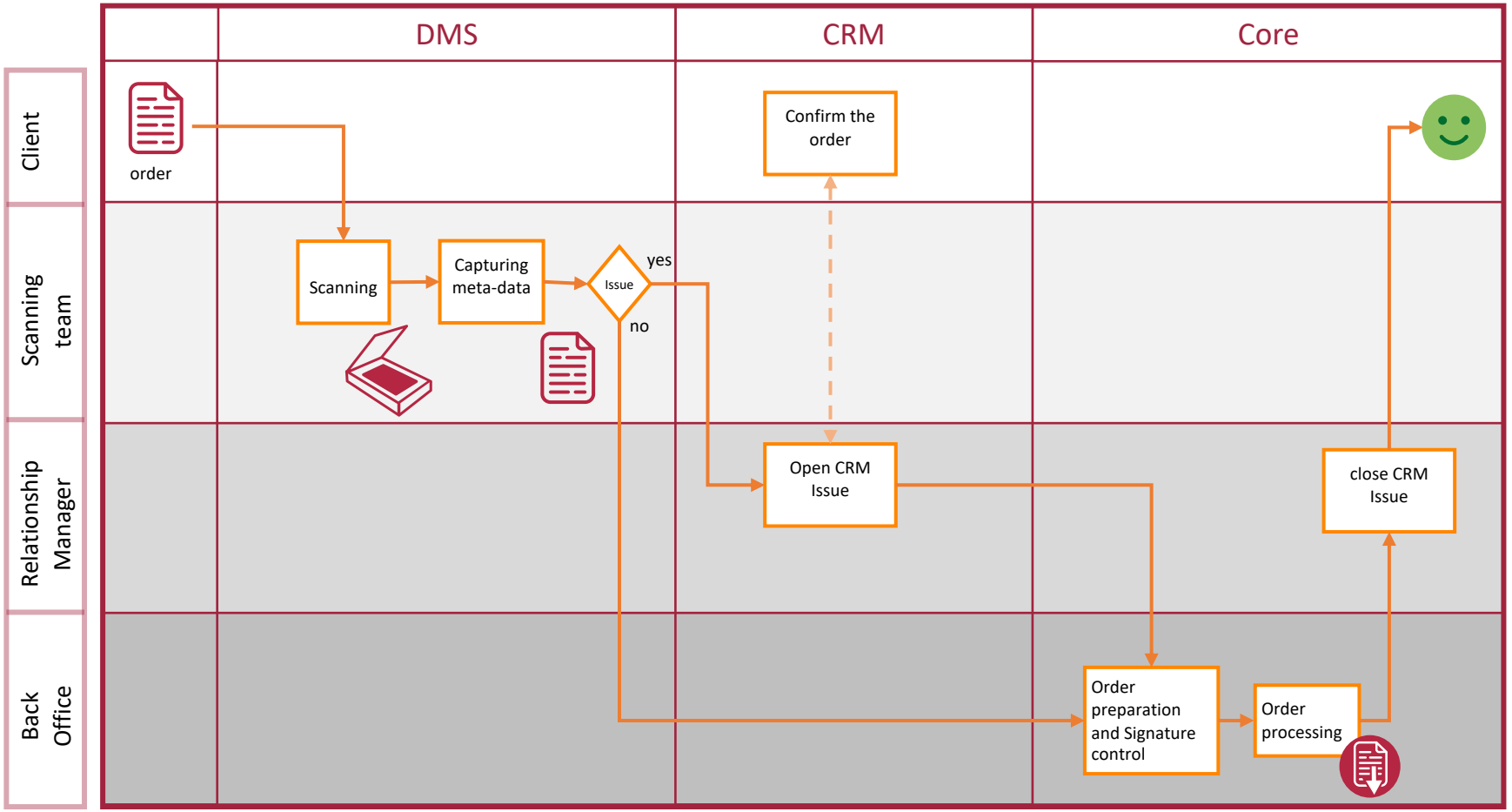
- set-up, perform and monitoring a defined workflow
- Introduce document routing based on configured rules

- create and manage website content
- manage documents and output for multiple author editing and participation

- Store, secure, preserve and track records within the entity.
- Records can be physical or electronic

DMS Automation in Practice

Example: Scan to Order Processing



The Reality: Major Efforts Required for Success

What it takes to make DMS work

Pre-conditions

IT Infrastructure / Security

Objective:
Installation of new SW and system up-and running, including security

Main activities:
- HW purchasing
- SW installation
- DEV, TEST, PROD environments up and running
- Operation Manuals
- Security concept
- Design authorisations
- Upload users and assign roles
- Operation manual for security

Digitalisation

Migration Archive System

Objective: Archived documents available in the new system

Main activities:
- Migrate documents
- Reconciliation
- Security rules
- Decommissioning former Archive System

Scan-to-Archive

Objective: Digitalise paper documents

Main activities:
- Define document catalogue
- Collect paper documents
- Design "Client Dossiers"
- Scan and digitalise documents

Scan-to-Mailbox

Objective: Introduce digital mailbox within the DMS solution

Main activities:
- Define scanning process and logistics
- Setup Mailbox and authorisations
- Design business process and workflows
- Define operating procedures and partners

Use

Piloting & Training

Objective: performant users and lean processes

Main activities:
- Training concept
- Pilot
- Trainings
- Support
- CRM issues
- Assessments (tool use, user feedback)

IT → **Business**

Roles & Responsibilities in Closing Data Gaps

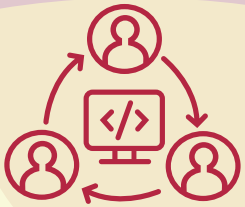
Closing ingestion, versioning, and lineage gaps requires aligned roles and responsibilities across the enterprise

IT / Data Architecture

- System design & integration
- Scalability & performance
- Metadata frameworks
- Search/index infrastructure

Business Units / Users

- Define document types & workflows
- Identify content usage needs
- Tagging and searchability input
- Drive user adoption



Compliance / Governance

- Retention policies
- Access control & security
- Legal/regulatory requirements
- Audit readiness

Best Practices for Content-Driven Collaboration

1. Cross-Functional Teams

- Involve *data architects, content managers, legal, and business teams* in design

2. Tools & Standards

- Metadata Harmony: Use tools like Collibra or Microsoft Purview to tag documents and datasets consistently
- Automation: Deploy AI/ML to extract data from documents (e.g., invoices → ERP)

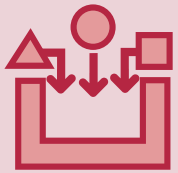
3. Success Metrics

- Reduced manual work (e.g., auto-classifying documents)
- Faster compliance audits (track documents + data lineage)

Summary: Achieving Governance Across All Gaps

Key Practices and Success Factors for Managing Unstructured Data

Data Ingestion

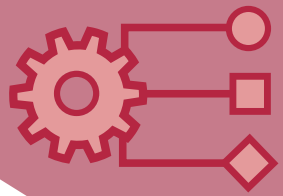


- Standardized capture
- Workflow integration
- Metadata tagging
- Tool: ECM / DMS

Versioning

- Unified version policy
- Reproducible pipelines
- Audit and monitoring
- Tool: ECM / DMS

Lineage



- Provenance logging
- Metadata frameworks
- Automated workflow tracking
- Tool: ECM / DMS

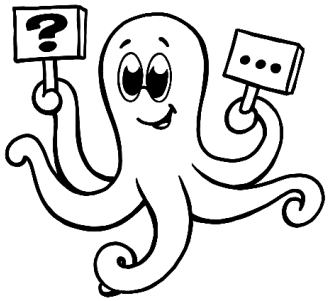
- Executive sponsorship
- Cross-functional collaboration
- Governance and compliance policies embedded in processes
- Enterprise solutions: ECM, DMS, collaboration platforms
- User adoption and training

Data Operations

DataOps ensures that data is reliably delivered from source to consumption

Store → Move → Operate → Recover

Why storing data?



- Prevent **loss** due to accidents
- **Analyze to extract** insights, discover trends, and make informed decisions
- **Create record** of past events
- **Train AI** on stored data to recognize patterns and make predictions

Storage is about cost, performance, and lifecycle, and not just saving data

Common Storage Tiers

Tier	Storage Type	Speed	Cost	Usage
Hot Tier	SSDs, NVMe	🔥 Fastest	💸 High	Frequently accessed data (e.g. active files, real-time apps)
Warm Tier	High-performance HDDs	⚡ Medium	💰 Medium	Occasionally accessed data (e.g. recent backups, logs)
Cold Tier	Low-cost HDDs or slow disks	🐢 Slower	💵 Low	Rarely accessed data (e.g. old reports, compliance data)
Archive Tier	Tape, cloud archive, deep cloud	🧊 Very slow	🪙 Very low	Regulatory or long-term data storage (e.g. 7+ year retention)

Hard Disk Drive (HDD): Uses spinning magnetic platters and a moving read/write head (mechanical)
Solid State Drive (SDD): Flash memory (no moving parts)
Non-Volatile Memory Express (NVMe): SSD using a much faster protocol

Architectures for Connecting Storage

DAS - Direct-Attached Storage

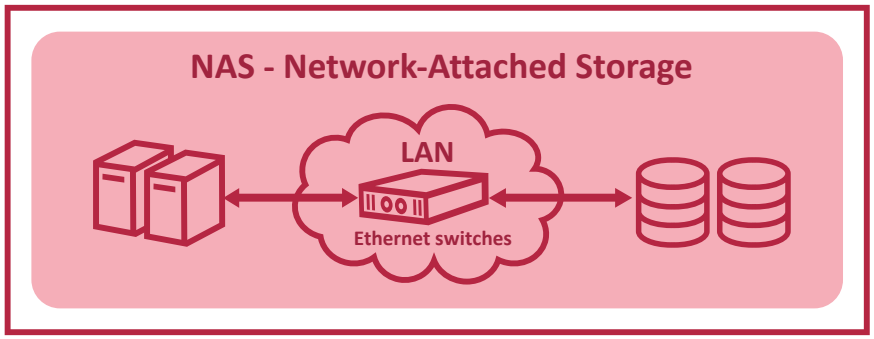
Storage that is **directly connected** to a single computer or server

Simple to set up, Fast for local access, Low cost
Not shareable, Limited scalability

NAS - Network-Attached Storage

Dedicated storage device **connected to a network**, accessed via file-level protocols like NFS, SMB, or FTP

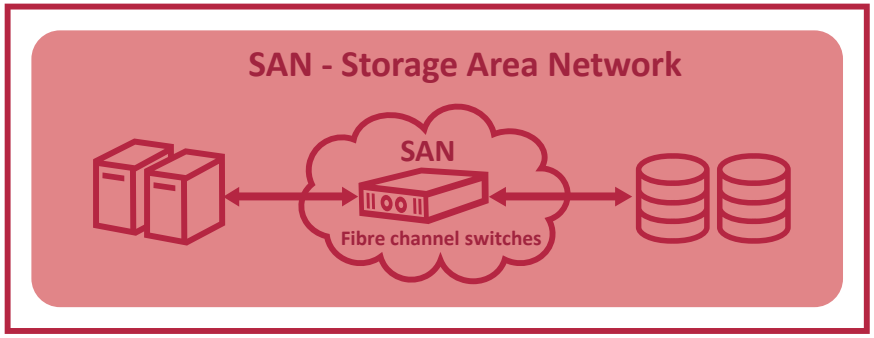
Shared access, Easy to share files across devices, User-friendly
File-level access only, Slower than SAN for high workloads



SAN - Storage Area Network

High-speed, block-level **storage** network

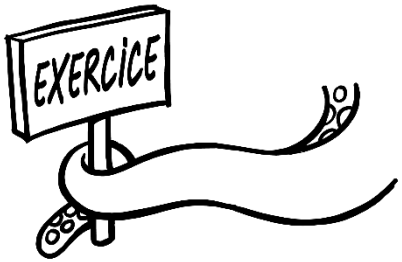
Extremely fast, Highly scalable, Centralized management
Expensive, Complex setup

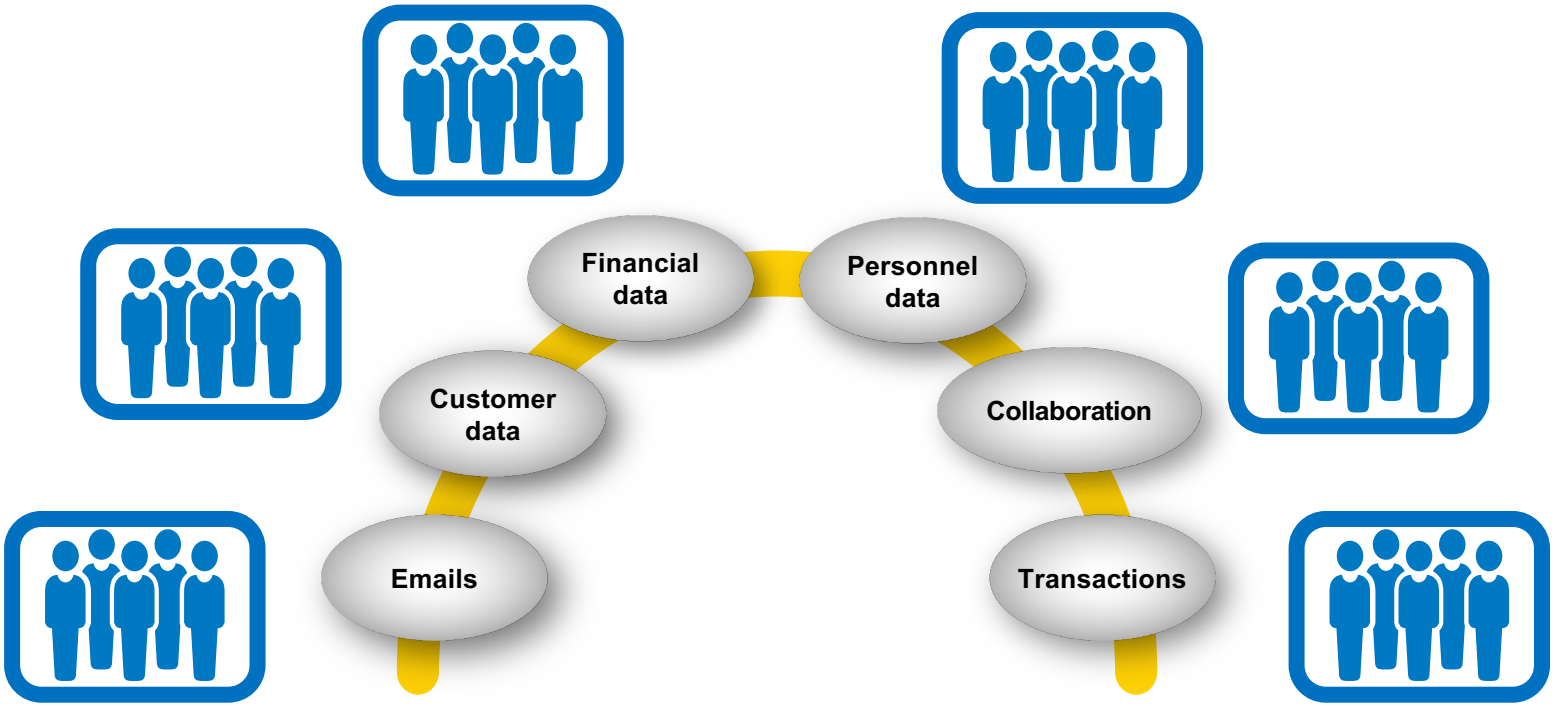


Storage Rules

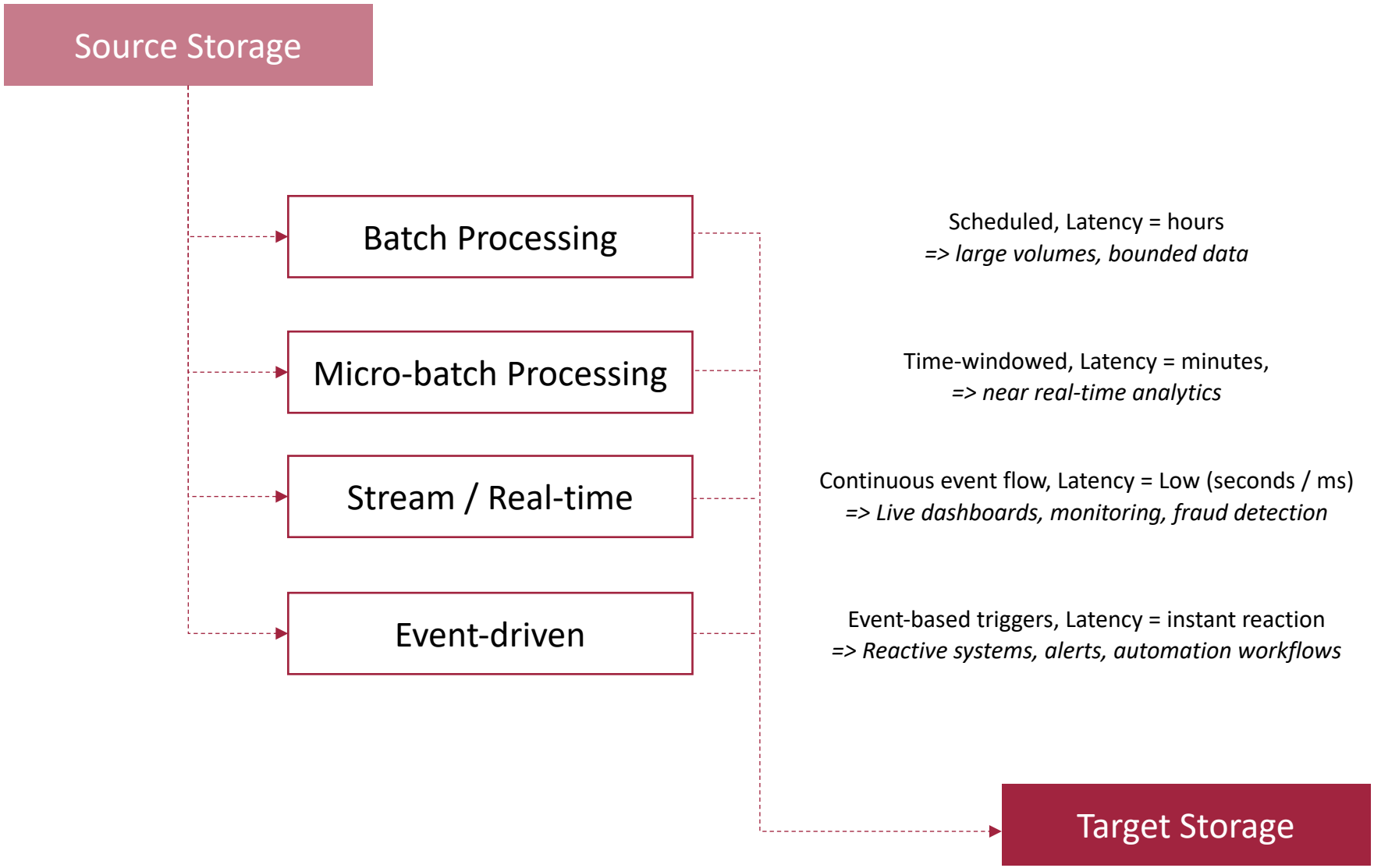
Define rules for the storage of the following data

- Emails
- Customer data
- Financial data
- Personnel data
- Collaboration platform
- Transactions



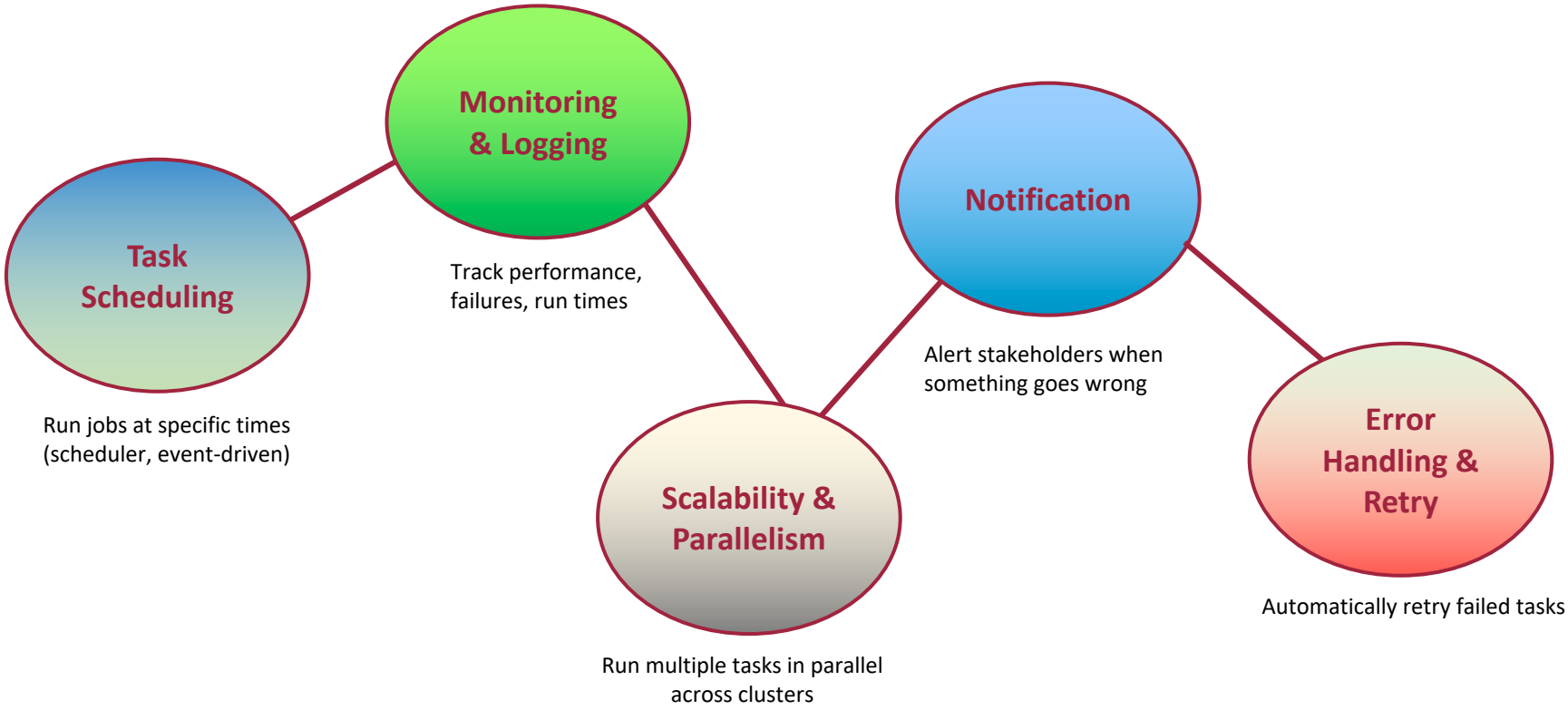


From Storage at Rest to Data in Motion



Data Pipeline Orchestration

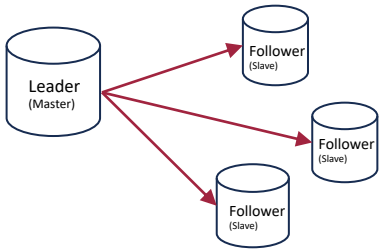
Data Pipeline Orchestration is the process of coordinating, scheduling, and managing complex data workflows — ensuring that tasks run in the correct order, handle dependencies, recover from failures, and operate efficiently across systems



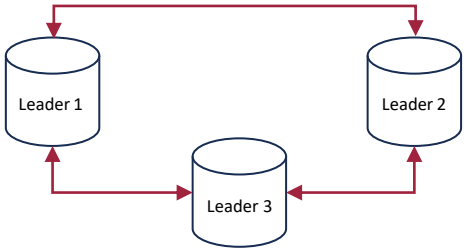
Replication Strategies

Data replication involves creating and maintaining copies of data across different locations or systems to improve availability, fault tolerance, and performance

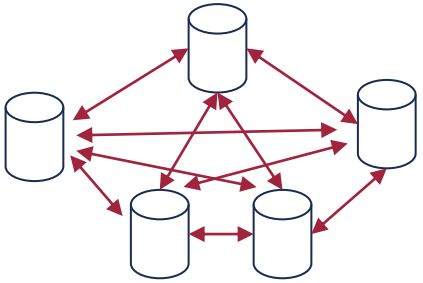
Single Leader



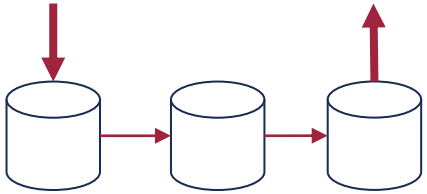
Multi-Leader



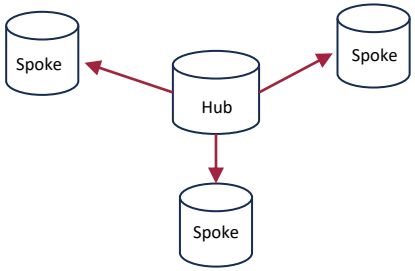
Peer-to-Peer



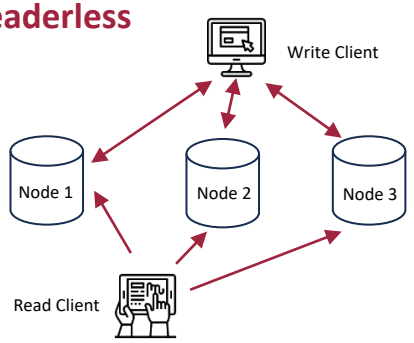
Chain



Hub-and-Spoke



Leaderless



Operate: System Availability and Downtimes

Availability (%)	Downtime per Year	Class
99% ("Two Nines")	~3.65 days	Basic reliability
99.9% ("Three Nines")	~8.76 hours	Common for cloud
99.99% ("Four Nines")	~52.6 minutes	High availability
99.999% ("Five Nines")	~5.26 minutes	Mission-critical

Availability

% Availability = Total Uptime / (Total Downtime + Total Uptime)×100

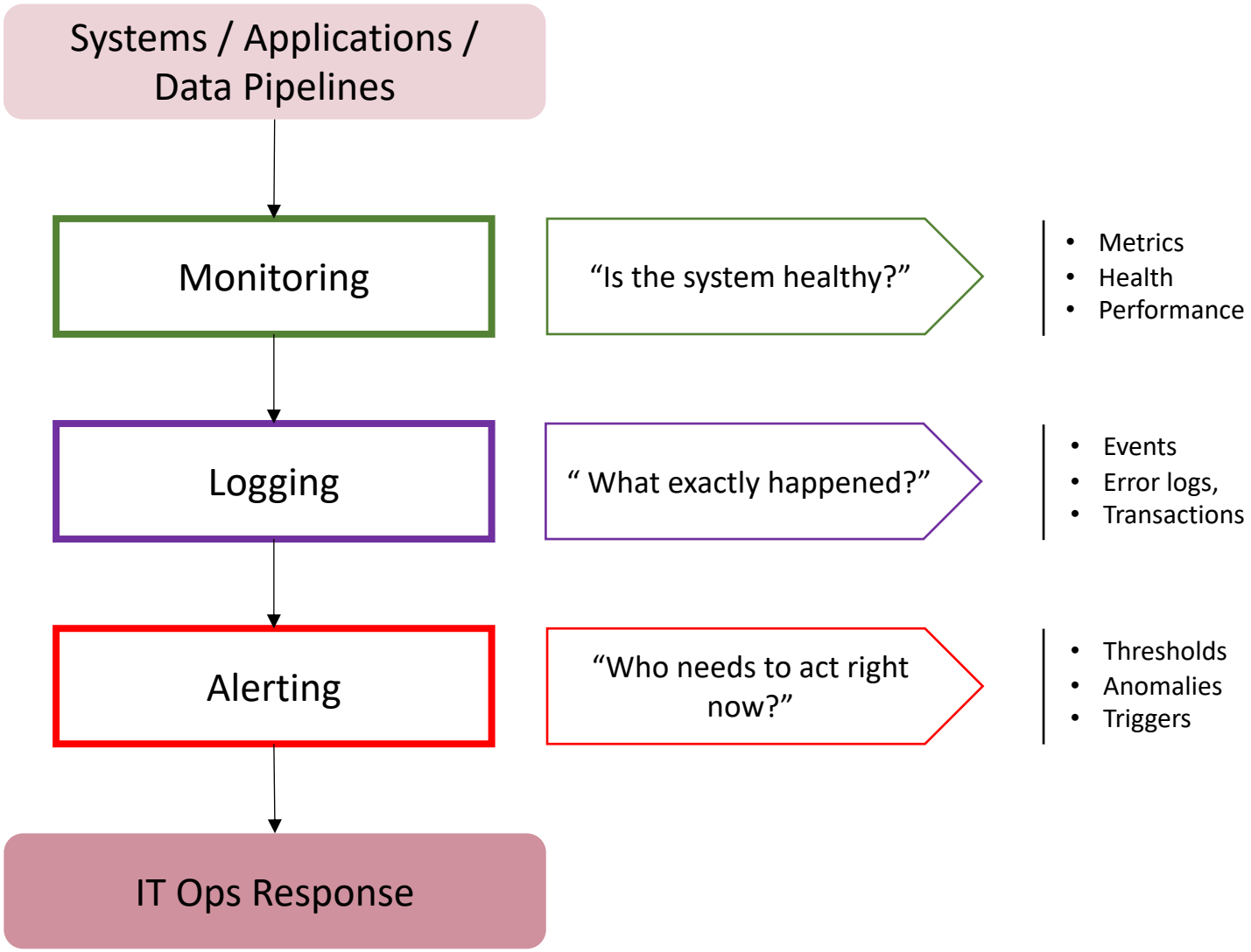
Mean Time Between Failures (MTBF)

MTBF = Total Operational Time / Number of Failures

Mean Time To Repair (MTTR)

MTTR = Total Downtime / Number of Failures

Operate: Monitoring > Logging > Alerting



The System Log

System Log

To be able to recover from failures that affect transactions, the system maintains a log to keep track of all transaction operations that affect the values of database items, as well as other transaction information that may be needed to permit recovery from failures. The log is a sequential, append-only file that is kept on disk, so it is not affected by any type of failure except for disk or catastrophic failure.

Log records

Based on unique transaction-id that is generated automatically by the system for each transaction and that is used to identify each transaction

1. **[start_transaction, *T*].** Indicates that transaction *T* has started execution.
2. **[write_item, *T*, *X*, *old_value*, *new_value*].** Indicates that transaction *T* has changed the value of database item *X* from *old_value* to *new_value*.
3. **[read_item, *T*, *X*].** Indicates that transaction *T* has read the value of database item *X*.
4. **[commit, *T*].** Indicates that transaction *T* has completed successfully, and affirms that its effect can be committed (recorded permanently) to the database.
5. **[abort, *T*].** Indicates that transaction *T* has been aborted.

Source: R.Elmasri and S.B. Navathe, Fundamentals of Database Systems

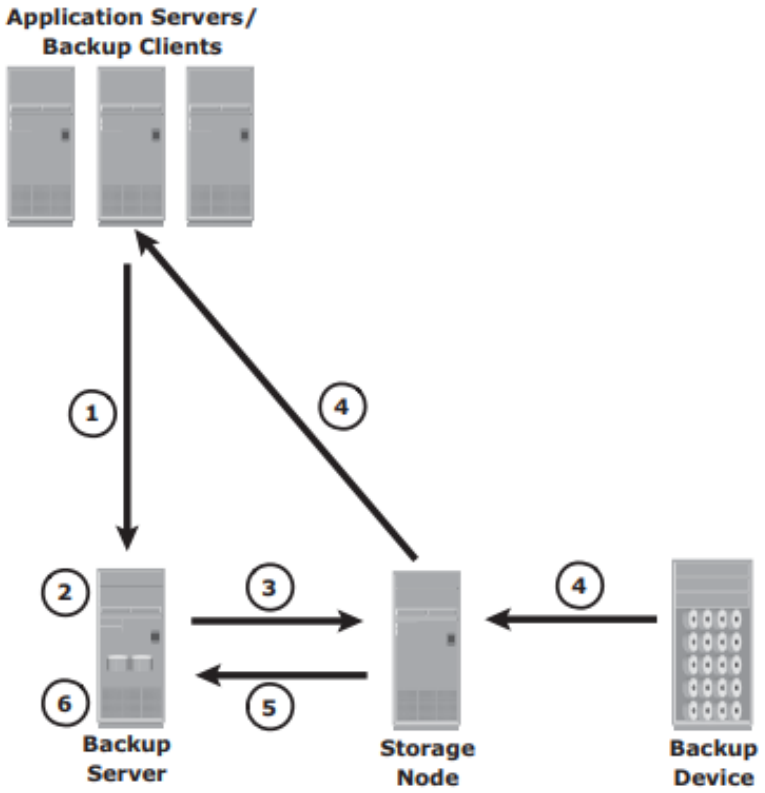
Types of Failures

1. **A computer failure** (system crash): A hardware, software, or network error occurs in the computer system during transaction execution. Hardware crashes are usually media failures—for example, main memory failure.
2. **A transaction or system error**: Some operation in the transaction may cause it to fail, such as integer overflow or division by zero. Transaction failure may also occur because of erroneous parameter values or because of a logical programming error.3 Additionally, the user may interrupt the transaction during its execution
3. **Local errors or exception conditions detected by the transaction**: During transaction execution, certain conditions may occur that necessitate cancellation of the transaction. For example, data for the transaction may not be found. **Concurrency control enforcement**: Transactions aborted because of serializability violations or deadlocks
4. **Disk failure**: Some disk blocks may lose their data because of a read or write malfunction or because of a disk read/write head crash
5. **Physical problems and catastrophes**: power or air-conditioning failure, fire, theft, sabotage, overwriting disks or tapes by mistake, and mounting of a wrong tape by the operator

Source: R.Elmasri and S.B. Navathe, Fundamentals of Database Systems

Restoring Data

Backups are used to restore data in case of data loss or corruption, for instance when an important file or email is corrupted



1. The backup client requests the backup server for data restore.
2. The backup server scans the backup catalog to identify data to be restored and the client that will receive data.
3. The backup server instructs the storage node to load backup media in the backup device.
4. Data is then read and sent to the backup client.
5. The storage node sends restore metadata to the backup server.
6. The backup server updates the backup catalog.

Source: Gnanasundaram S., Shrivastava A. (2012) Information Storage and Management

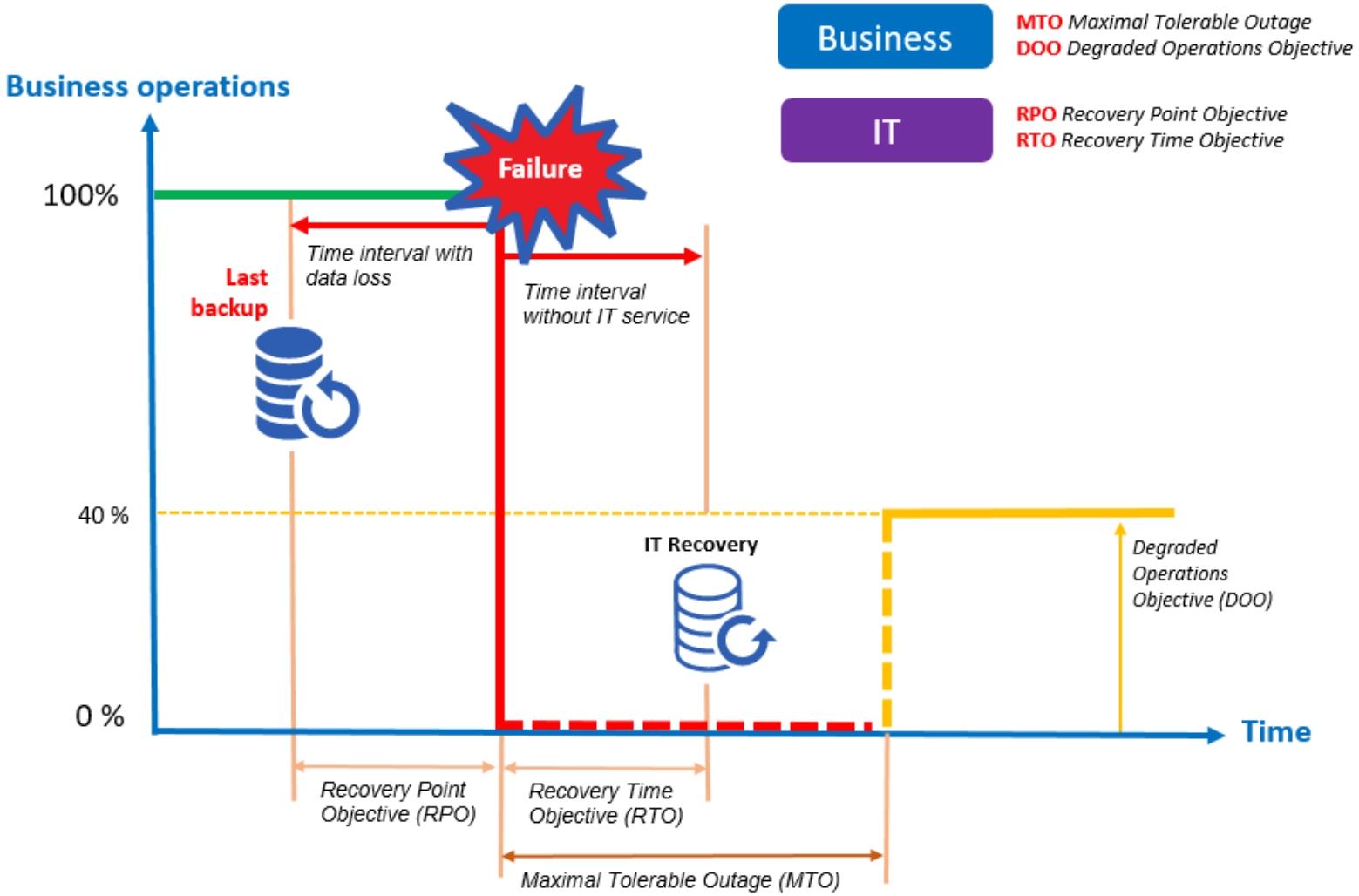
A Brief History of Data Backup

Era	Backup Method	Medium	Comment
1950s–60s	Manual	Magnetic tape	Mainframes
1970s–80s	Local, Manual	Floppy, tape	PCs emerge
1990s	Automated scripts	CD/DVD, RAID	Backup software becomes standard
2000s	Remote/networked	NAS, SAN	Start of online backups
2010s	Cloud, BaaS	Cloud, VMs	DRaaS, compliance focus
2020s	AI-driven, continuous	Immutable, snapshot	Ransomware resistance, automation

PC: Personal Computer
CD: Compact Disc
DVD: Digital Versatile Disc
RAID: Redundant Array of Independent Disks
NAS: Network Attached Storage

SAN: Storage Area Network
BaaS: Backup as a Service
VM: Virtual Machine
DRaaS: Disaster Recovery as a Service
AI: Artificial Intelligence

Recovery Principle

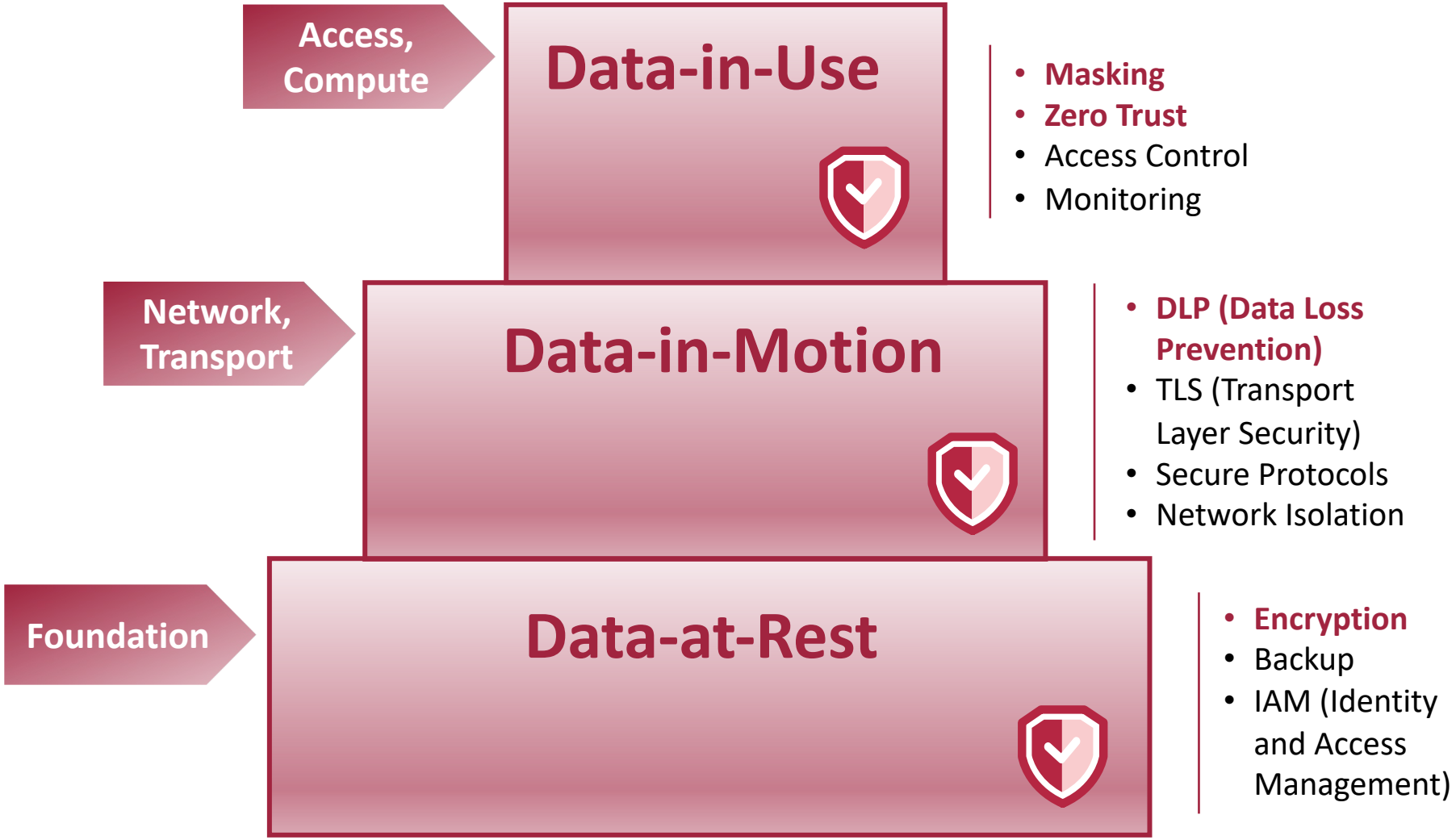


Best Practices for DataOps

Automate	=> Speed and reliability
Implement CI/CD for Data	=> Faster and safer deployment
Use Version Control	=> Traceability and rollback
Reuse Components	=> Efficiency and standardization
Optimize Scalability & Performance	=> Improved query times and infrastructure cost reduction
Use Infrastructure as Code	=> Consistency across environments
Secure Data & Comply with Regulations	=> Breach prevention and risk mitigation

Data Protection

Protecting Data Across All States: at Rest, in Motion, in Use



Encryption Strategies

Encryption strategies are structured approaches to protect data by converting it into unreadable code for unauthorized users

Data at Rest Encryption
- Protects stored data (e.g., files, databases)
- Common methods: Full Disk Encryption (FDE), File-level Encryption, Database TDE

⇒ *Example: BitLocker, VeraCrypt, Oracle TDE*

Data in Transit Encryption
- Protects data moving across networks
- Uses protocols like TLS/SSL, HTTPS, SSH

⇒ *Example: Encrypting API communication or web traffic*

End-to-End Encryption (E2EE)
- Only the sender and recipient can decrypt the data
- Prevents intermediaries (including service providers) from accessing content

⇒ *Example: Signal, WhatsApp*

Symmetric
- Same key for encryption and decryption (e.g., AES)

Asymmetric
- Public key encrypts, private key decrypts (e.g., RSA)

Hybrid Encryption
- Combines symmetric and asymmetric encryption for performance and security
- Common in secure messaging and SSL/TLS

Data Loss Prevention (DLP)

DLP is a security technology for the prevention of sensitive data from being accessed, shared, or lost

How to guarantee the data integrity and how to keep data from reaching a person or entity not privileged to the information?

Main Goals of DLP

- **Prevent data breaches** and leaks
- **Enforce compliance** with regulations (e.g., GDPR, HIPAA, PCI DSS)
- **Protect intellectual property** and trade secrets
- **Control insider threats** (accidental or malicious)

DLP in Action: From Discovery to Enforcement

A step-by-step approach to data loss prevention

Discover & Classify

Identifies sensitive data (e.g., credit cards, health records) using:

- **Pattern matching** (e.g., regex for U.S. Social Security Numbers)
- **Machine learning** (contextual analysis)
- **Fingerprinting** (exact data matching)

Monitor & Analyse

Tracks data movement across:

- Emails
- Cloud apps
- Printers
- USB devices

Enforce Policies

Takes action based on rules:

- **Block**: Prevent sending a file with PCI data.
- **Encrypt**: Auto-encrypt sensitive attachments.
- **Quarantine**: Hold suspicious files for review.
- **Alert**: Notify admins of policy violations

Employee tries to email customer PII externally -> Block email + alert security team

Developer uploads source code to public GitHub -> Detect keywords (e.g., "API key") + block upload

Healthcare worker copies PHI to a USB drive -> Encrypt file or disable USB access

Credit card numbers stored unencrypted in a database -> Identify + flag for remediation

Example of DLP Implementation

Severity	E-mails	Web Upload	Endpoint Protection
High	**Blocked*,** with a notification from the DLP system *Can only be released by IT Security + Data Protection*	**Blocked** with a message when uploading *The block cannot be overridden by the originator*	**Blocked** with a message when uploading *The block cannot be overridden by the originator*
Medium	**Blocked*,** with a notification from the DLP system *Can be released by superiors*	**No Blocking** *A DLP case is created and checked 'post-mortem' in each case by superiors*	**Blocked** *can be released by the originator on their own responsibility by means of a dialogue and by entering a reason.* *DLP case is created and checked 'post-mortem' in each case*
Low	**Blocked*,** with a notification from the DLP system *can be released by the originator on their own responsibility*	**No Blocking** *A DLP case is also created for this and checked on a random basis*	**Blocked** *can be released by the originators on their own responsibility by means of a dialogue and by entering a reason.* *DLP case is created.*
Info	**Blocked*** - but automatically released	**No Blocking** *A DLP case is also created for this and checked randomly*	**No Blocking** *a DLP case is also created for this and checked on a random basis.*

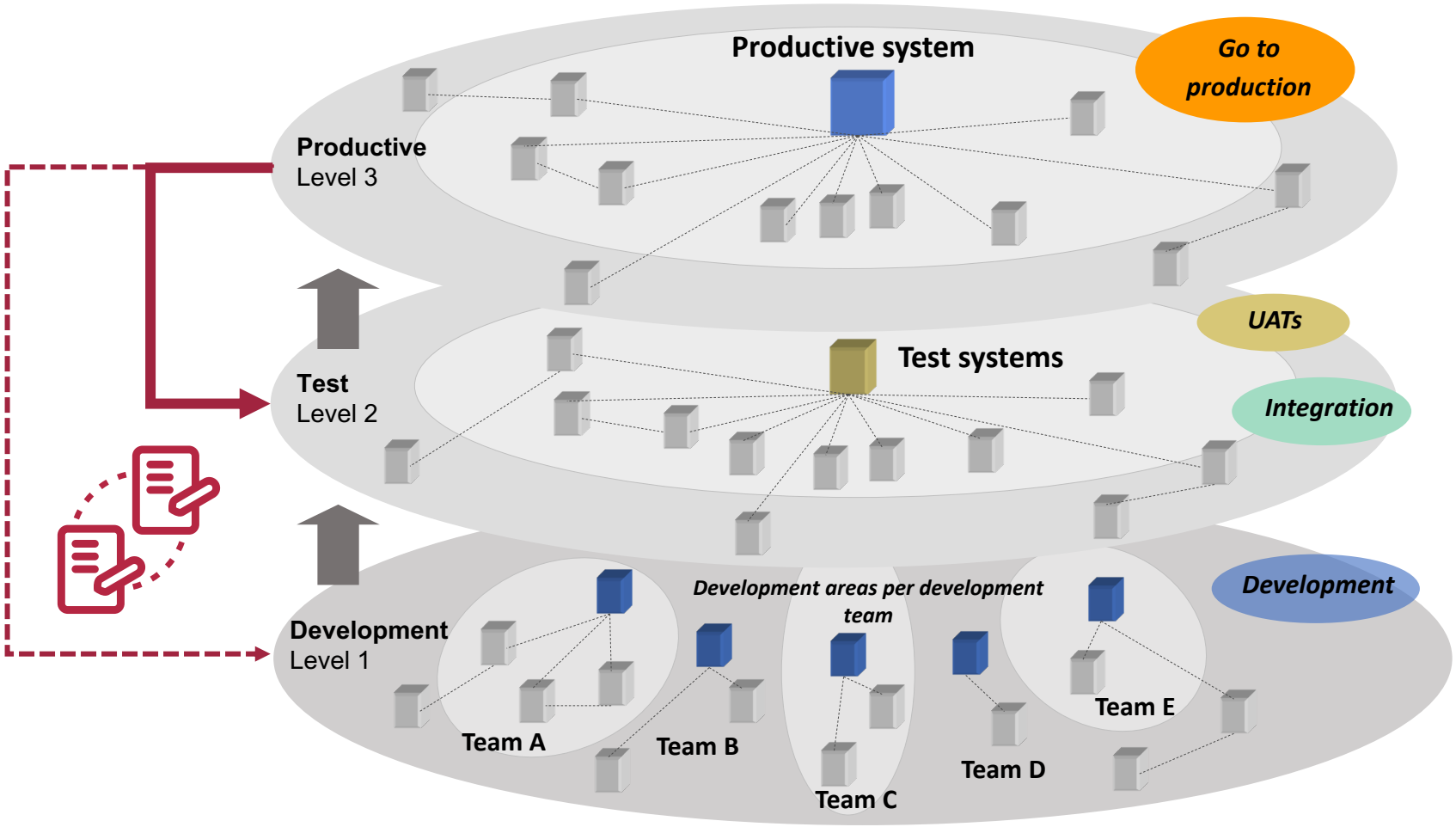
(*) Blocked for emails means that the email is moved to the so-called quarantine.
The email remains there until an automatic or manual decision is made as to whether the email can be released or must remain blocked.

Data Masking

Data masking is a data protection technique used to hide or obfuscate sensitive data by replacing it with fictional but realistic values. It allows organizations to use data for testing, development, analytics, or training without exposing real personal or confidential information.

Technique	Description	Example
Static Masking	Permanently replaces sensitive data in a database copy	A production DB is copied for testing, and real names are replaced with fake ones
Dynamic Masking	Hides data in real-time based on user permissions (original data stays intact)	A call centre agent sees only the last 4 digits of a credit card
Redaction/Nulling	Completely removes or blacks out sensitive data	SSN: ***-**-****
Pseudonymization	Replaces identifiers with fake but realistic data (reversible with a key)	"John Doe" → "User123" (can be mapped back if needed)
Shuffling	Randomly reorders values within a column (e.g., swapping names)	Employee salaries are shuffled so no real person is linked to their actual salary
Data Substitution	Replaces real data with realistic but fictional data	"Paris" → "Berlin" (for location data)
Tokenization	Replaces sensitive data with random tokens (mapped in a secure vault)	Credit card 1234-5678-9012-3456 → 8X9F-2P3Q-6R1S-4T5U

Copying and Anonymizing Data



Data Anonymization

Name	Phone	Number plate	Client number
John Sample	+41 33 222 454 11	BS 45798	BP 234-555-888H

randomly replaced | fixed substitution | partial information | central key without CID

Name	Phone	Number plate	Client number
Adqx Hakft	+41 11 111 11 11	BS 11111	BP 234-555-888H

Client number	Account number	Currency	Account balance
BP 234-555-888H	CH9300762011	CHF	2'000

central key without CID | substitution with pseudonym | General information | No client reference

Client number	Account	Currency	Balance
BP 234-555-888H	CH5604850412	CHF	2'000

Anonymization: make re-identification impossible through sufficient destruction of information
Pseudonymization: only allow re-identification through protected information (using table or function)

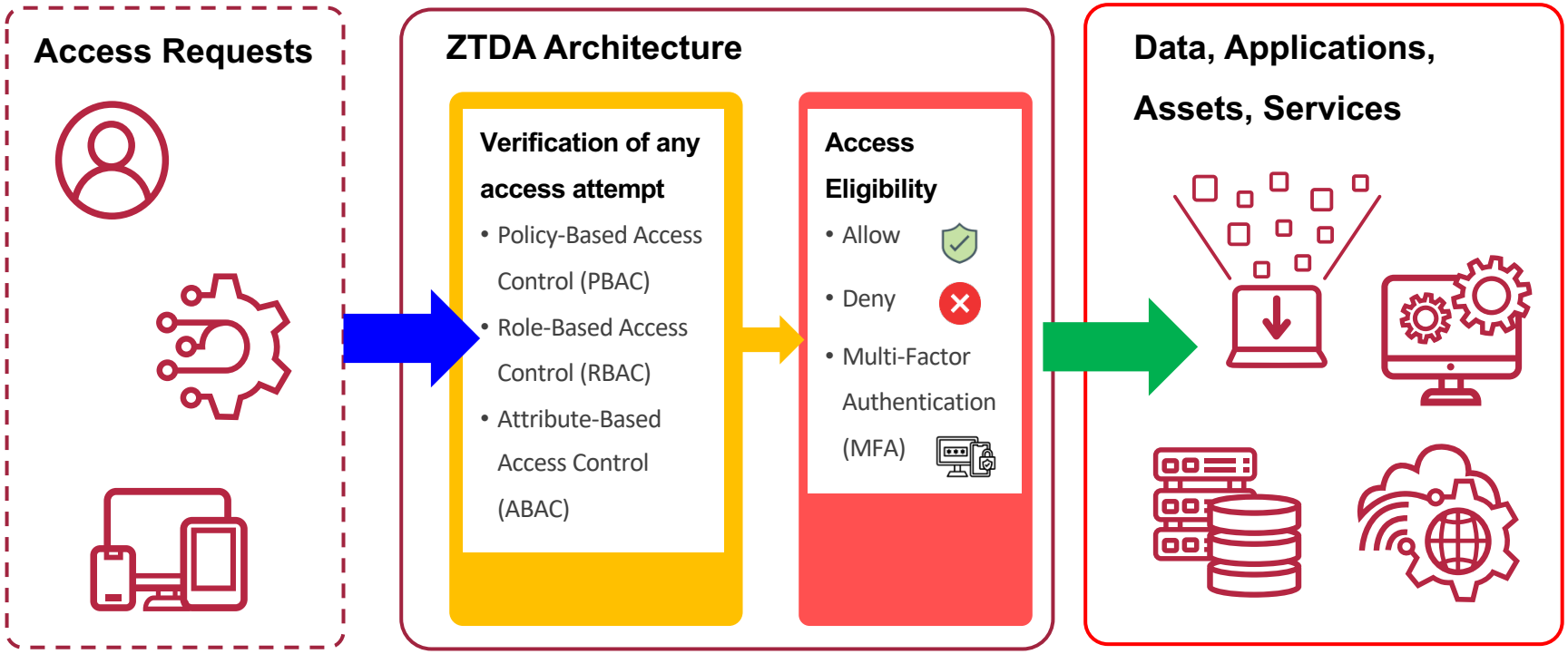
The central challenge remains the degree of anonymization while preserving at the same time test-relevant information

Using Data for Testing Purposes

Assessment criteria	Productive Data	Anonymized Data	Synthetic Data
Confidentiality / data protection	problematical	partly given	good
Data Search	laborious	difficult	Tailor-made for test cases
Data usage	problematical	problematical	generated again and again
Data Specificity	not specific	not specific	exactly the defined properties
Data history	Inherently available	Inherently available	possible, but difficult
Data volume	Production volumes	Production volumes	large volumes well producible
New data	Manually producible	Manually producible	programmable
Smoke tests	possible with few selected data sets	possible with few selected data sets	Newly generated required data
Integration tests	Possible, but always changes of data	Possible, but always changes of data	Tailor-made for test cases
User Acceptance Tests (UATs)	Possible, but always changes of data	Possible, but always changes of data	exactly the defined properties
Last / Performance Tests	good for history, otherwise complex	good for history, otherwise complex	quite possible, except for history

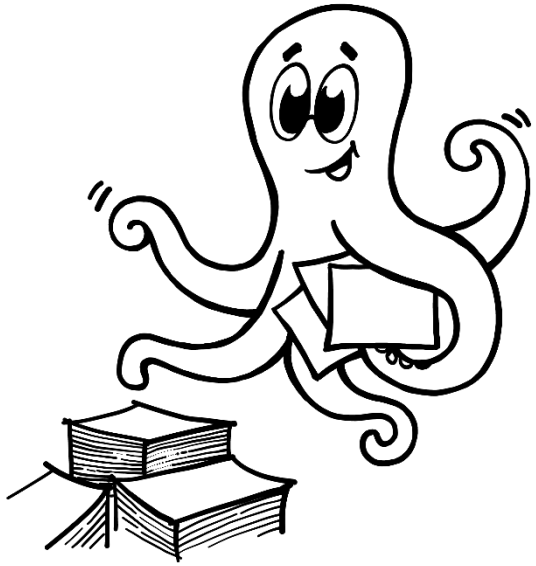
Zero-Trust Data Access (ZTDA)

ZTDA is a security model that enforces strict identity verification, least-privilege access, and continuous monitoring for every request to access data—regardless of whether the request comes from inside or outside a network. "Never trust, always verify."



Take away and must know!

- To understand key concepts of data architecture
- To understand the meaning of Enterprise Content Management
- To know the use of Document Management Systems
- To understand the principles of data storage and activities of data operation management
- To recognize critical data protection and security challenges and to know the best practices of data loss prevention



Literature

- DoDAF (2010) The DoDAF Architecture Framework Version, U.S. Department of Defense
- Elmasri R., Navathe S.B. (2016) Fundamentals of Database Systems, Pearson
- Gnanasundaram S., Shrivastava A. (2012) Information Storage and Management, John Wiley & Sons, Inc
- Kaijser P. (1995) Data protection in communications and storage, IFIP International Federation for Information Processing
- Kampffmeyer U. (1994) Was ist ECM Enterprise Content Management? Project Consult

KNOWLEDGE

www. know-ledge.ch