**DATA MESH**

# 1. Giới thiệu chung về Data Mesh

* Kiến trúc xử lý dữ liệu truyền thống:
  + Một đội xủ lý dữ liệu tập trung quản lý tất cả các dữ liệu trong tổ chức từ tất cả các đội chuyên môn khác nhau
  + Đội này có nhiệm vụ:
    - Tập hợp dữ liệu từ các nguồn khác nhau
    - Biến đổi dữ liệu cho thành các định dạng sử dụng được
    - Chuẩn bị dữ liệu để các nhóm khác hay khách hàng cuối có thể sử dụng
  + Thách thức:
    - Nhu cầu và kỳ vọng về dữ liệu rất lớn
    - Sự phân chia về dữ liệu thành Operational Data và Analytical Data với các công nghệ và kỹ thuật khác nhau
    - Data Scaling
    - Các quy trình nghiệp vụ phức tạp và dễ thay đổi
    - Giá trị thu lại từ dữ liệu chưa tương xứng với số tiền bỏ ra
* Data Mesh ra đời để giải quyết các vấn đề của kiến trúc tập trung này:
  + Trách nhiệm quản lý dữ liệu được phi tập trung hóa và chuyển về các đội chuyên môn (domain team)
  + Tăng tính linh hoạt, nhanh chóng thích ứng với các thay đổi
  + Giảm thời gian thu được giá trị từ dữ liệu
  + Giảm chi phí vận hành và hoạt động

# 2. Các nguyên lý cơ bản của Data Mesh

* Quyền sở hữu dữ liệu thuộc về các miền chuyên môn (Domain-oriented ownership)
  + Áp dụng Domain driven design (DDD) cho dữ liệu
  + Phi tập trung hóa quy trình xây dựng, quản lý và vận hành dữ liệu
  + Dữ liệu được xử lý trực tiếp bởi đội có kiến thức chuyên môn về dữ liệu đó
  + Phân loại:
    - Source-aligned domain data: Dữ liệu phân tích (Analytical data) phản ánh hoạt động nghiệp vụ của hệ thống
    - Aggregate domain data: Dữ liệu mới sinh ra bằng cách tổng hợp dữ liệu từ nhiều nguồn khác nhau
    - Customer-aligned domain data: Dữ liệu dược biến đổi để phục vụ các ca sử dụng của khách hàng
* Dữ liệu là một loại sản phẩm (Data as a product)
  + Áp dụng tư duy sản phẩm đối với dữ liệu, coi dữ liệu là một sản phẩm cung cấp cho các đội khác và cho người dùng cuối
  + Các tính chất của sản phẩm dữ liệu:
    - Khả năng tìm kiếm được (Discoverable): Các đội có nhu cầu sử dụng phải dễ dàng tìm kiếm được các sản phẩm dữ liệu cần thiết (thông qua một danh mục hay một registry tập trung)
    - Khả năng truy xuất được (Addressable): Truy xuất được nguồn gốc của sản phẩm - một địa điểm duy hất cập nhật các thông tin về Data Product (các phiên bản, thay đổi, documentation, ...)
    - Dễ hiểu (Understandable): Data Product được đặc tả bằng các ngôn ngữ dùng chung (ubiquitous language) sao cho các đội khác có thể tự sử dụng mà không cần tham khảo người tạo ra sản phẩm
    - Đúng đắn và đáng tin cậy (Truthful & Trustworthy): Người dùng có thể kiểm tra lịch sử của Data Product từ khi load vào hệ thống qua các phép biến đổi cho tới khi cung cấp đến khách hàng và phải đảm bảo các ràng buộc về dịch vụ (SLO) với các bên liên quan. Các thông tin này phải được bao gồm trong Data Product dưới dạng metadata hoặc các biểu đồ (data linage, event sourcing)
    - Dễ tiếp cận trong tổ chức (Natively accessible): Data Product phải được thiết kế để mọi vai trò (Data Analyst, AI Engineer, App Developer, ...) đều có thể tiếp cận và sử dụng được
    - Dễ tích hợp với các hệ thống khác (Interoperable): Data Product phải có khả năng trao đổi và kết nối giữa các miền một cách dễ dàng. Định dạng file, schema, kiểu dữ liệu phải được thêm vào phần metadata của dũ liệu để đảm báo khả năng tích hợp
    - Có giá trị độc lập (Valueable on its own): Data Product phải có ý nghĩa mà không cần liên kết với các Data Product khác. Tránh việc mapping mỗi bảng trong csdl với một data product do một số bảng chỉ dùng để hỗ trợ join
    - Bảo mật (Secure): Có thể trao quyền và giới hạn truy nhập đối với các data product
  + Mỗi đội chuyên môn (domain team) phải có vai trò sở hữu và phát triển Data Product(Data product owner & Data product developer):
    - Người sở hữu đảm bảo tiến độ và chất lượng của Data Product, đem lại sự hài lòng cho khách hàng sử dụng sản phẩm
    - Người phát triển có trách nhiệm xây dựng, bảo trì và phục vụ Data Product
* Cơ sở hạ tầng dữ liệu tự phục vụ (Self-serve data platform):
  + Cung cấp một cách tự động các công cụ và dịch vụ hỗ trợ không phụ thuộc vào miền chuyên môn (domain-agnostic)
  + Giúp các đội chuyên môn dễ dàng chia sẻ và sử dụng các Data Product một cách tự động
  + Đảm bảo khả năng tích hợp và tính thống nhất về công nghệ giữa các đội làm việc trong các lĩnh vực khác nhau
  + Tránh trùng lặp về chức năng và về dữ liệu
  + Giảm thiểu chi phí vận hành
  + Cần có một đội Data Platform chuyên biệt phụ trách việc xây dựng, bảo trì và vận hành cơ sở hạ tầng dùng chung
* Phối hơp quản trị dữ liệu (Federated computational governance):
  + Mỗi đội sẽ cử ra một đại diện để tham gia vào một hội đồng chung để đặt ra các tiêu chuẩn và thực thi các chuẩn này trên các đội chuyên môn
  + Trong hội đồng cũng có đại diện của đội Data Platform và các chuyên gia bảo mật, pháp lý
  + Đảm bảo tính linh hoạt và khả năng phối hợp giữa các đội nhóm nhưng vẫn tuân thủ các quy định của tổ chức và pháp luật

# 3. Cách thiết kế Data Product

* Data Product là đơn vị nhỏ nhất cấu thành Data Mesh, đóng vai trò là các node trong lưới dữ liệu
* Thiết kế dựa trên chức năng cung cấp (affordance):
  + Serve:
    - Customer first
    - Multimodal Output Port
    - Imutable Data
    - Replayable
  + Consume:
    - Input Ports: Data Source(internal services, external source), other Data Products, self
    - Locality: Dữ liệu có thể được di chuyển đến gần với nơi tiêu thụ để tăng tốc độ tính toán
  + Transform:
    - Nonprogrammatic (SQL, GraphQL, ...) vs programmatic (Apache Spark, Apache Beam, ...)
    - Thiết kế theo kiểu pipeline (Dataflow)
    - Lưu lại các artifact trong quá trình transform như code, configuration, các bảng và view được cache trong quá trình biến đổi
  + Discover & Understand:
    - Có cơ chế Data Product discovery tương tự service discovery cho software
    - Self registration
    - Global URI for data discovery
    - Phải bao gồm schema, metadata và documentation
    - Đưa ra các thông số thống kê (metrics)
  + Compose:
    - Kết hợp được từ nhiều nguồn khác nhau với các định dạng khác nhau
    - Tìm kiếm được các Data Product liên quan đến nhau
    - Các Data Product phảihỗ trợ việc kết hợp các dữ liệu có quan hệ với nhau
  + Manage:
    - Toàn bộ life-cycle quản lý bởi domain-team một cách tự động, phát triển độc lập với các Data Product khác
    - Manifest
  + Govern:
    - Policy-as-code
    - Sidecar Pattern
    - Các quy định được chuẩn hóa với toàn bộ tổ chức như mã hóa, bảo mật và quyền truy cập
  + Observe:
    - Log, Trace, Metric
    - Structered and standardized format
    - Observability as a data product
* Ví dụ:
  + Table hoặc View trong cơ sở dữ liệu
  + Các file chứa dữ liệu không có cấu trúc như file ảnh, audio nhưng phải bao gồm metadata
  + Các file đơn giản không có schema như JSON, CSV, excel
  + Các file có chứa dữ liệu có cấu trúc như Protobuf, Parquet, Avro, ...
  + Các data stream thời gian thực
  + Các data stream log lại các sự kiện phục vụ hoạt động nghiệp vụ
  + REST API
  + Các đặc trưng được trích xuất phục vụ cho AI/ML
  + Dashboard / report để trực quan hóa dữ liệu
* [[Template Data Product Canvas](https://www.datamesh-architecture.com/data-product-canvas):](https://www.datamesh-architecture.com/data-product-canvas)
  + Miền nghiệp vụ (Domain): Nhóm có nhiệm vụ phụ trách và phát triển Data Product
  + Tên (Data Product Name): Tên của sản phẩm phải tuân thủ theo các naming convention như trong các ngôn ngữ lập trình sao cho thật dễ hiểu với người sử dụng
  + Khách hàng và ca sử dụng (Customer & Use case): Xác định rõ các khách hàng và ca sử dụng của từng khách hàng như đối với một sản phẩm phần mềm
  + Đầu ra (Output Ports): Nêu rõ định dạng và giao thức (file, table, API, ...) mà Data Product expose
  + Metadata: Các thông tin khác về sản phẩm mà người sử dụng muốn biết như quyền sở hữu, quyền truy cập, schema, ...
  + Đầu vào (Input Ports): Nêu rõ Data Product đã sử dụng những nguồn dữ liệu nào và định dạng, giao thức tiếp nhận các nguồn đó
  + Thiết kế chính (Data Product Design): Miêu tả toàn bộ quá trình thiết kế Data Product từ mức độ đặc tả đến code để tạo ra Data Product
  + Khả năng quan sát được (Observability): Khả năng quan sát được trạng thái bên trong dựa vào các thông số bên ngoài
    - Operational Metrics: Liệt kê phiên bản, thay đổi, freshness, các thông số thống kê
    - Quality Metrics: Yêu cầu về chức năng như độ chính xác, tính hoàn chỉnh, tính toàn vẹn
    - Service Level Objectives (SLOs): Đảm bảo tính tin cậy và chính xác, thông báo khi phát hiện có xảy ra bất thường
  + Ngôn ngữ dùng chung (Ubiquitious Language): Liệt kê thuật ngữ chung giữa những người phát triển và người sử dụng từ các miền chuyên môn khác nhau
  + Phân loại (Classification): Một trong ba loại Source-aligned, Aggregate, Customer-aligned

# 4. Một số nền tảng công nghệ hỗ trợ triển khai Data Mesh

* dbt (Data Build Tool):
  + Được xây dựng trên tầng Storage chủ yếu tập trung vào hỗ trợ việc biến đổi dữ liệu (T trong ETL)
  + Có thể nhận dữ liệu và lưu dữ liệu vào và từ nhiều nguồn khác nhau (Snowflake, Databricks, AWS S3, Google BigQuery, ...) dưới nhiều định dạng khác nhau
  + Có kết nối với git repo hỗ trợ việc quản lý các phiên bản
  + Lập trình viên chủ yếu viết bằng ngôn ngữ tương tự SQL và điều chỉnh cấu hình bằng file YAML
  + Có công cụ trực quan hóa Data Linage từ nguồn
  + Hỗ trợ tự động kiểm thử (singular test, generic test, data freshness, ...)
  + Dễ dàng sinh documentation chỉ bằng một câu lệnh
  + Hỗ trợ CI/CD pipeline và job scheduling
* [Data Mesh Manager](https://demo.datamesh-manager.com/demo801706003247)