Information Integration Systems
Chapter 1. Intro: Integration Problem

SIA & SDBIS

1.1 Integration Problem

- Context and Definition.
- Solutions.

Practical “reasons” to integrate

- Different databases (with different data models).
- Disparate autonomous and heterogeneous Applications and Data Sources.
- Different Business Processes within the same enterprise or inter-organizational context.
- Different Enterprise Compartiments with specific business processes and business application solutions.

Integration Problem
Technical Context

- Databases
  - Heterogeneous Database Models.
  - Distributed Database Systems.
- Autonomous Application and Services
  - Heterogeneous Runtime/Containers.
  - Diverse programming language and platforms.
- Heterogeneous Data Interchange Formats.
- Heterogeneous Interoperability Protocols.

Integration Definition

- Interoperability of
  - Distributed,
  - Heterogeneous,
  - and Autonomous
- Enterprise Architectural Components such as
  - Systems,
  - Applications,
  - Services,
  - and Databases.

Information Integration Systems Definition

“*Information integration systems* offer uniform access to a set of autonomous and heterogeneous data sources. Sources can range from database systems and legacy systems to forms on the Web, web services and flat files. The data in the sources need not be completely structured as in relational databases. The number of sources in an *information integration application* can range from a handful to thousands.”

[Springer: Encyclopedia of Database Systems]

INTEGRATION
Conceptual Perspectives

- Database **Integration:**
  - Distributed Database Systems,
  - Federated Databases.
- Data **Integration:**
  - SQL Databases, Documents (structured, semi-structured), NoSQL, BigData, Web Data Services.
- Application **Integration:**
  - EAI Enterprise Application Integration.
- Service **Integration:**
  - service orchestration within a SOA Infrastructure,
  - service choreography within a REST/HTTP MSA(Microservices) Infrastructure.

Database/Data Integration Models

- Consolidated Database Integration (datawarehouses, datamarts, data lakes, lakehouses).
- Federated Database Integration (virtualized databases).
- Shared Data Integration (application integration data level).

EAI Enterprise Application Integration Models

- Presentation Integration (e.g. integration portals).
- Data Integration.
- Functional Integration (business oriented).

SOA Services Oriented Architecture Integration Models

- Service Mesh.
- ESB Enterprise Service Bus.
- MSA Microservices Architecture.

ESB Enterprise Service Bus

- Architecture:
  - Business Applications and Data Sources as event-driven services.
  - Central Bus to integrate event-driven services as endpoints.
- ESB middleware support services:
  - Central configuration,
  - Central Deployment,
  - Management Services.

Microservice Architecture
Spring Cloud Infrastructure Components

- Spring Boot Microservices
- Spring Cloud Discovery Service
- Spring Cloud Gateway (Edge Server)
- Other Spring Cloud Services
  - Spring Cloud Security (OAuth)
  - Spring Cloud Configuration Service (centralized configuration)
  - Spring Cloud Sleuth and Zipkin (distributed tracing)
  - Service Mesh Istio

1.2 Integration Solutions (Selection)

- Federated Database Systems (DI).
- REST Web Data Services Integration (REST.SOA).
- Enterprise Application Integration Framework (EAI).

Federated Database Systems

- Keep Source Data into original individual data sources.
- Integrate/Consolidate Source Data into a central virtual mediated database schema.

REST Web Data Services Integration (SOA)

- Keep Source Data into original individual data sources and support data interoperability by wrapping data access into Web REST Data Services.
- The integration process will be supported by the specific SOA Architecture by using specific *service interoperability* mechanisms:
  - Service Orchestration mechanisms,
  - Service Choreography mechanisms.

Enterprise Application Integration Framework (EAI)

- Keep Source Data into original individual data sources and support data interoperability centralizing access endpoints within the EAI platform/middleware.
- The integration process will be supported by the specific EAI platform Interoperability mechanisms:
  - Routing and Messaging Systems.

References for Data Integration

- AnHai Doan, Alon Halevy, Zachary Ives, Principles of Data Integration, 2012 Elsevier, Inc.
- Anthony Giordano, Data integration: blueprint and modeling techniques for a scalable and sustainable architecture, 2010, Pearson Education, Inc.
- April Reeve, Managing Data in Motion Data Integration Best Practice Techniques and Technologies, 2013 Elsevier, Inc.

References for SOA

- Erl, Thomas, Service-Oriented Architecture: Analysis and Design for Services and Microservices, PRENTICE HALL, 2017
- K. Siva Prasad Reddy, Beginning Spring Boot 2 Applications and Microservices with the Spring Framework, Apress, 2017
- Craig Walls, Spring Boot in Action, Manning; 1st edition (January 3, 2016)
- Magnus Larsson, Microservices with Spring Boot and Spring Cloud: Build resilient and scalable microservices using Spring Cloud, Istio, and Kubernetes, Packt Publishing; 2nd ed. edition (July 29, 2021)

References for EAI

- Claus Ibsen, Jonathan Anstey, Camel in Action, Manning; 2nd edition (February 18, 2018)
- Guilherme Camposo, Cloud Native Integration with Apache Camel: Building Agile and Scalable Integrations for Kubernetes Platforms, Apress (August 25, 2021)
- Jean-Baptiste Onofre, Mastering Apache Camel, Packt Publishing (June 30, 2015)