

Understanding Java Persistence API



Antonio Goncalves

JAVA CHAMPION

@agoncal www.antoniogoncalves.org



Previous Module



Database access

Read, create, update, delete data

Query, paginate

JDBC

JPA

Higher level of abstraction

Overview



What is a relational database?

What is an ORM tool?

What is Java Persistence API?

What it's not

Metadata for mapping

API to CRUD and query objects

Java SE and Java EE



What Is a Database?



Organized collection of data

Model our business requirements

Process information

Files

Schema-less

Relational databases



What Is a Relational Database?



Collection of tables

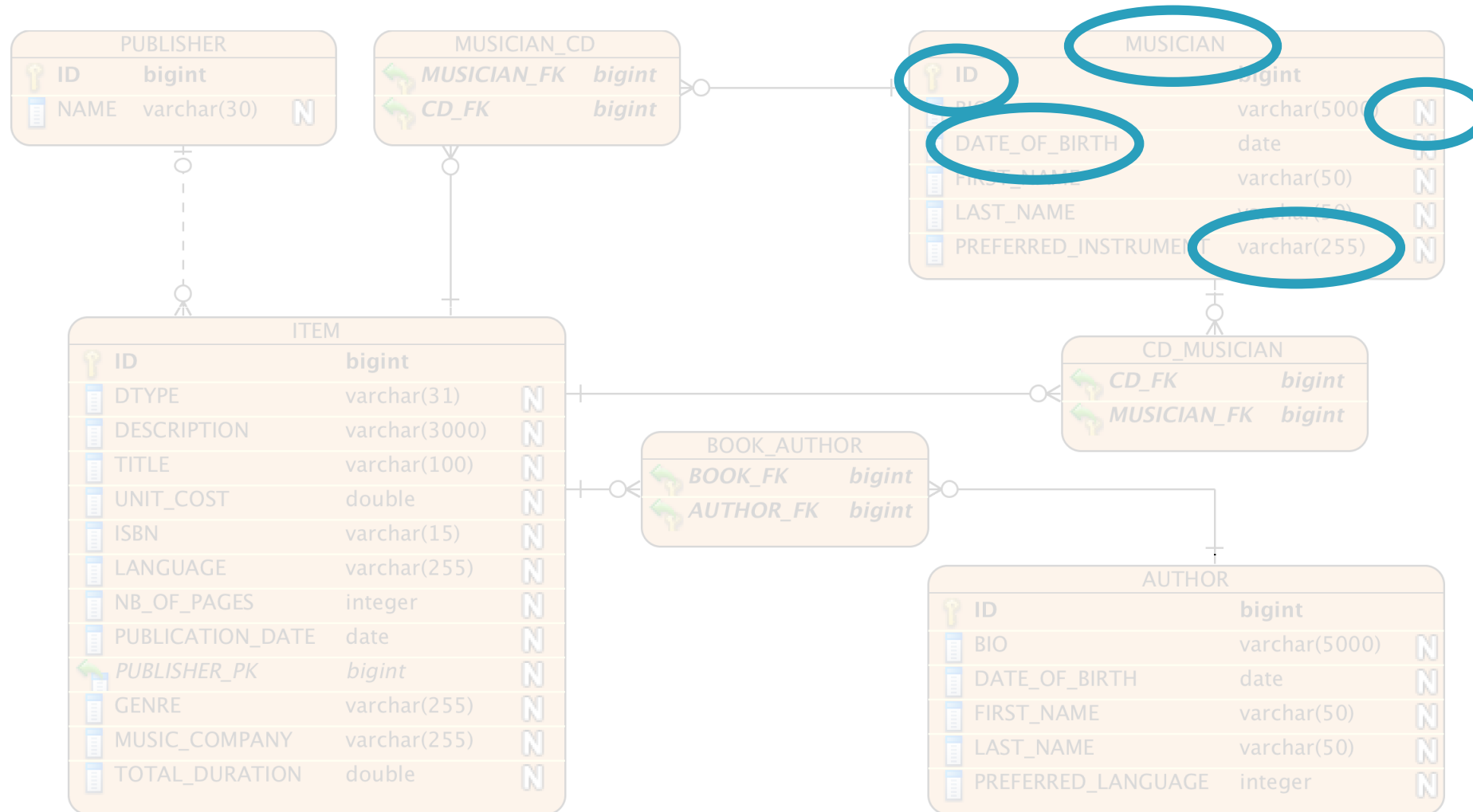
Row

Columns

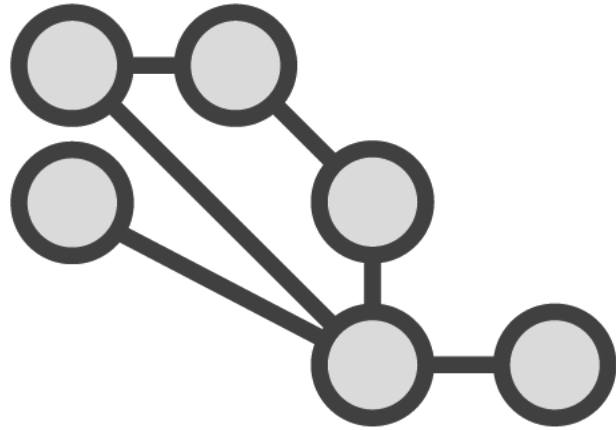
Relationships

Identity through primary keys

Database Diagram



What Is Object Oriented Programming?



Objects

Classes

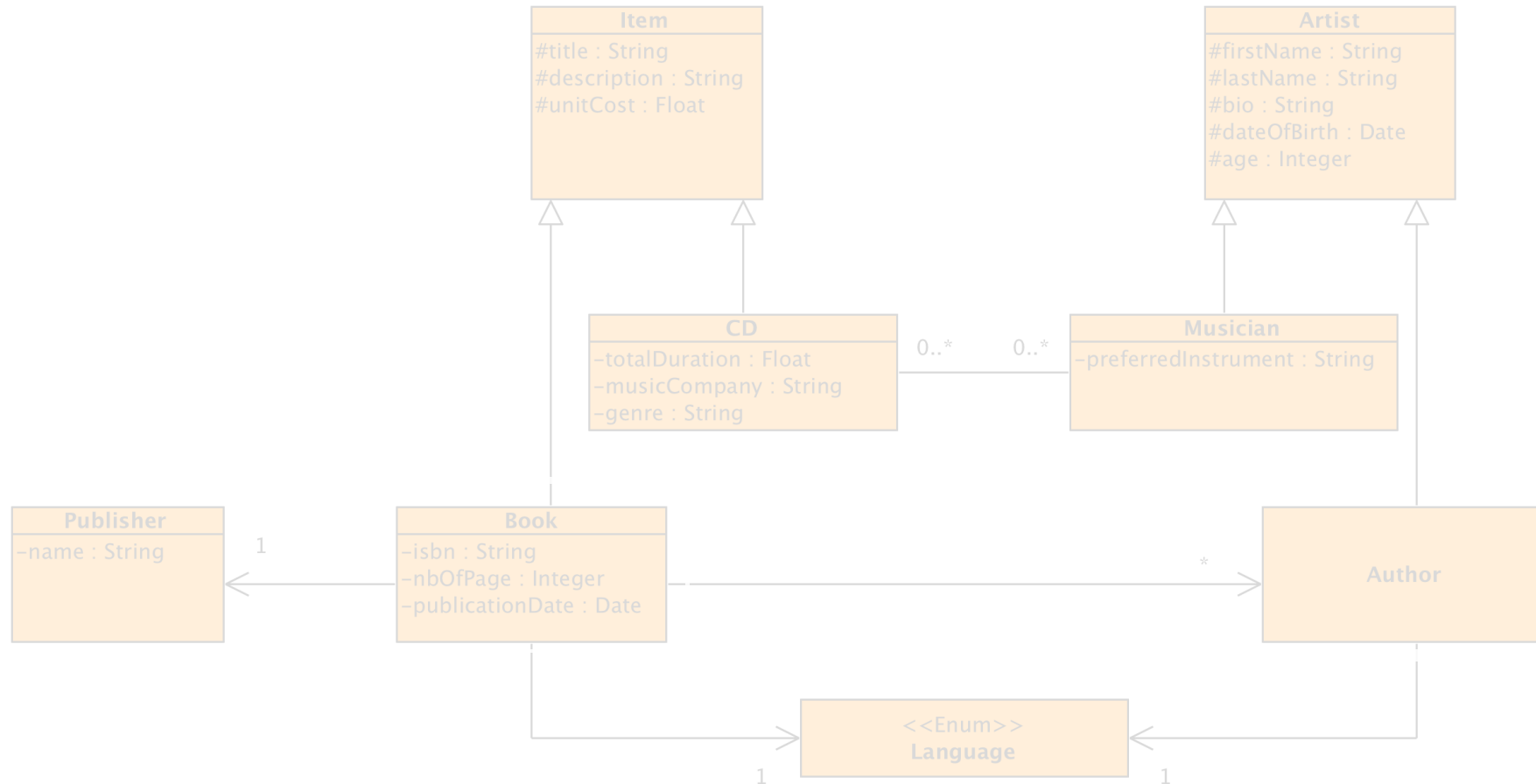
Inheritance

Reference to other classes

Abstract classes, interfaces, enumerations...

Encapsulate state and behavior

Class Diagram



What Is Object Relational Mapping?



Bring relational databases and objects together

Objects have transient state

Relational databases store state

Mapping between objects and tables

Java Persistence API

What Is Java Persistence API?



Abstraction above JDBC

ORM

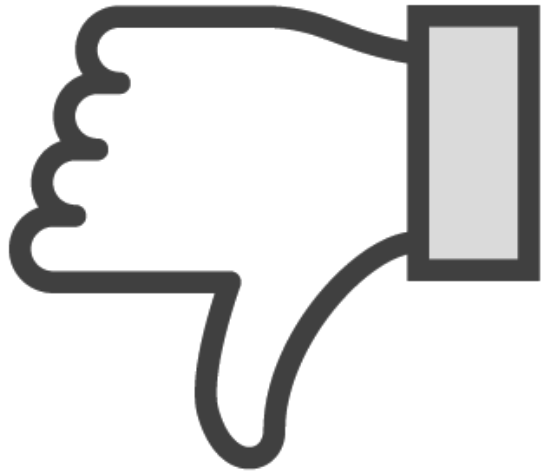
Entity Manager

- CRUD operations
- Java Persistence Query Language

Transactions

Life cycle

What Isn't Java Persistence API?



Schema-less databases

NoSQL

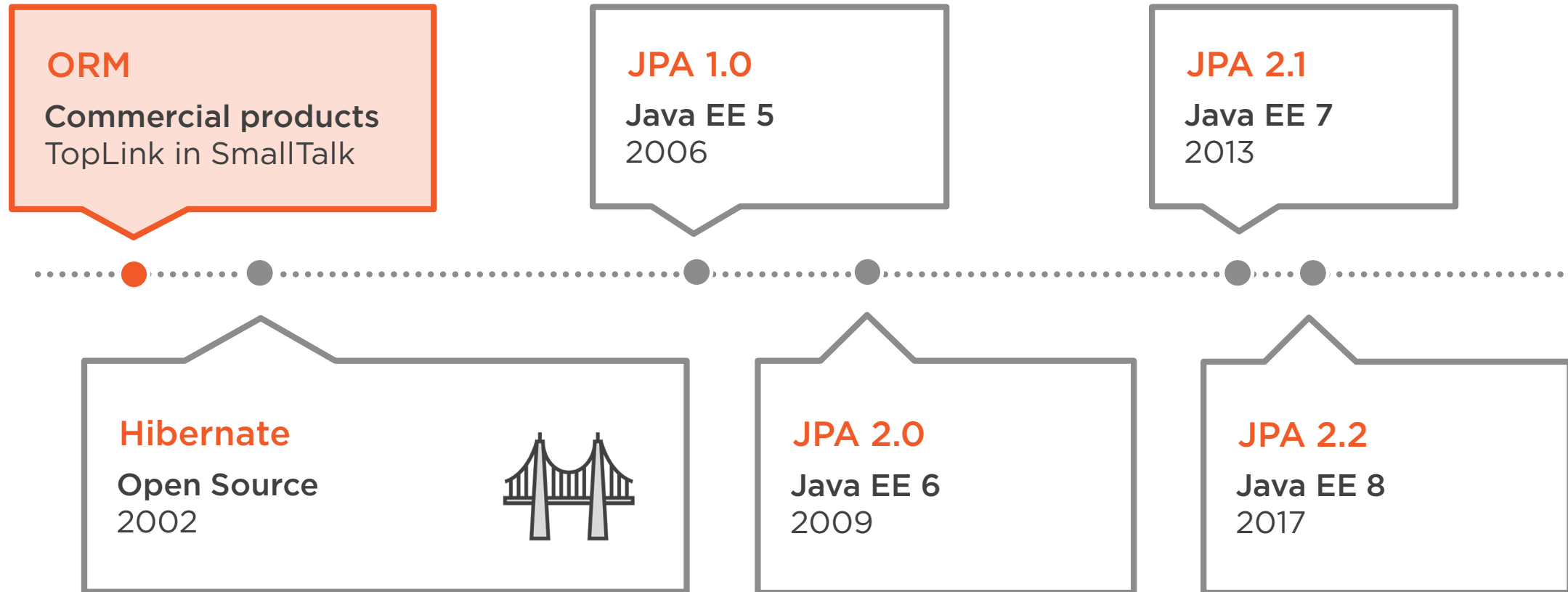
JSon

XML

JPQL \neq SQL



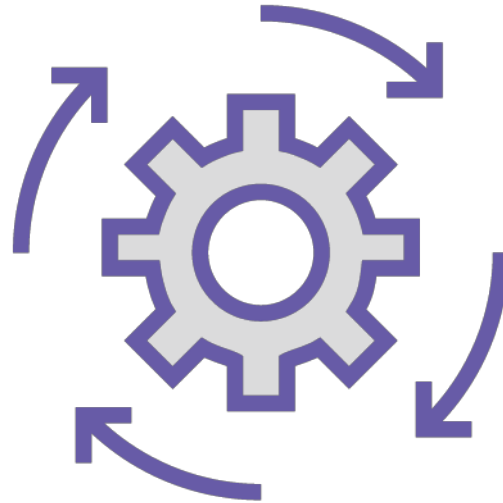
A Brief History of JPA



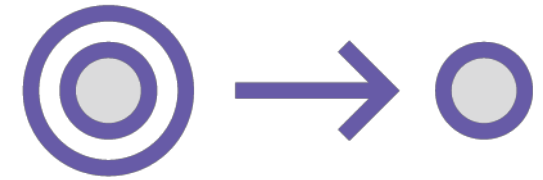
JPA Specification



Specification



Java Community
Process



Java Specification
Request



https://jcp.org/en/jsr/detail?id=338

JSR

Community

Expert Group

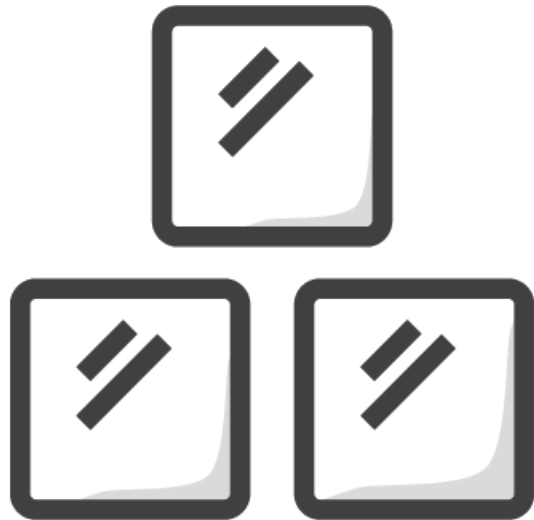
[Summary](#) | [Proposal](#) | [Detail \(Summary & Proposal\)](#) | [Nominations](#)

JSRs: Java Specification Requests

JSR 338: Java™ Persistence 2.2

Stage	Access	Start	Finish
Maintenance Release	Download page	04 Aug, 2017	
Maintenance Review Ballot	View results	06 Jun, 2017	19 Jun, 2017
Maintenance Draft Review	Download page	22 May, 2017	05 Jun, 2017
Final Release	Download page	22 May, 2013	
Final Approval Ballot	View results	09 Apr, 2013	22 Apr, 2013
Proposed Final Draft	Download page	01 Mar, 2013	
Public Review Ballot	View results	22 Jan, 2013	04 Feb, 2013
Public Review	Download page	21 Dec, 2012	21 Jan, 2013
Early Draft Review 2	Download page	31 Oct, 2012	30 Nov, 2012

JPA Implementations



Runtime provider

Implement the JSR 338

EclipseLink

Hibernate

Open JPA

No vendor locking

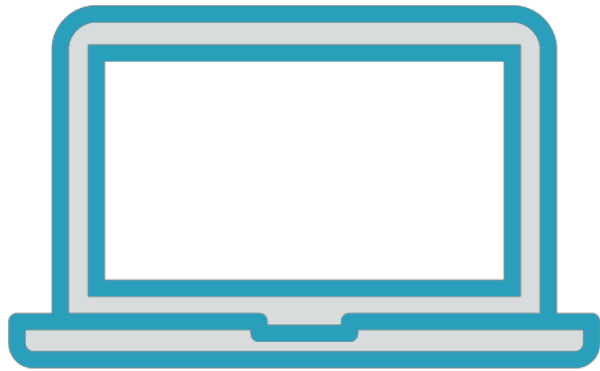


JPA Packages

Package	Description
➡ javax.persistence	Core API
➡ javax.persistence.criteria	Criteria API
➡ javax.persistence.metamodel	Metamodel API
➡ javax.persistence.spi	SPI for JPA providers



Where Can We Use JPA?



Java SE



Java EE

A JPA Entity

@Entity

```
public class Book {  
  
    @Id  
  
    private Long id;  
  
    private String title;  
  
    private String description;  
  
    private Float unitCost;  
    private String isbn;  
  
    // Constructors, getters & setters  
}
```



Mapping Metadata

```
@Entity
@Table(name = "t_book")
public class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @Column(name = "book_title", nullable = false)
    private String title;
    @Column(length = 2000)
    private String description;
    @Column(name = "unit_cost")
    private Float unitCost;
    private String isbn;

    // Constructors, getters & setters
}
```



CRUD Operations

```
public class Main {  
  
    static EntityManagerFactory emf =  
        Persistence.createEntityManagerFactory("myPU");  
    static EntityManager em = emf.createEntityManager();  
  
    public static void main(String[] args) {  
  
        Book book = new Book(4044L, "H2G2", "Scifi Book", "5678-5678", 247);  
        em.persist(book);  
  
        book = em.find(Book.class, 4044L);  
  
    }  
}
```

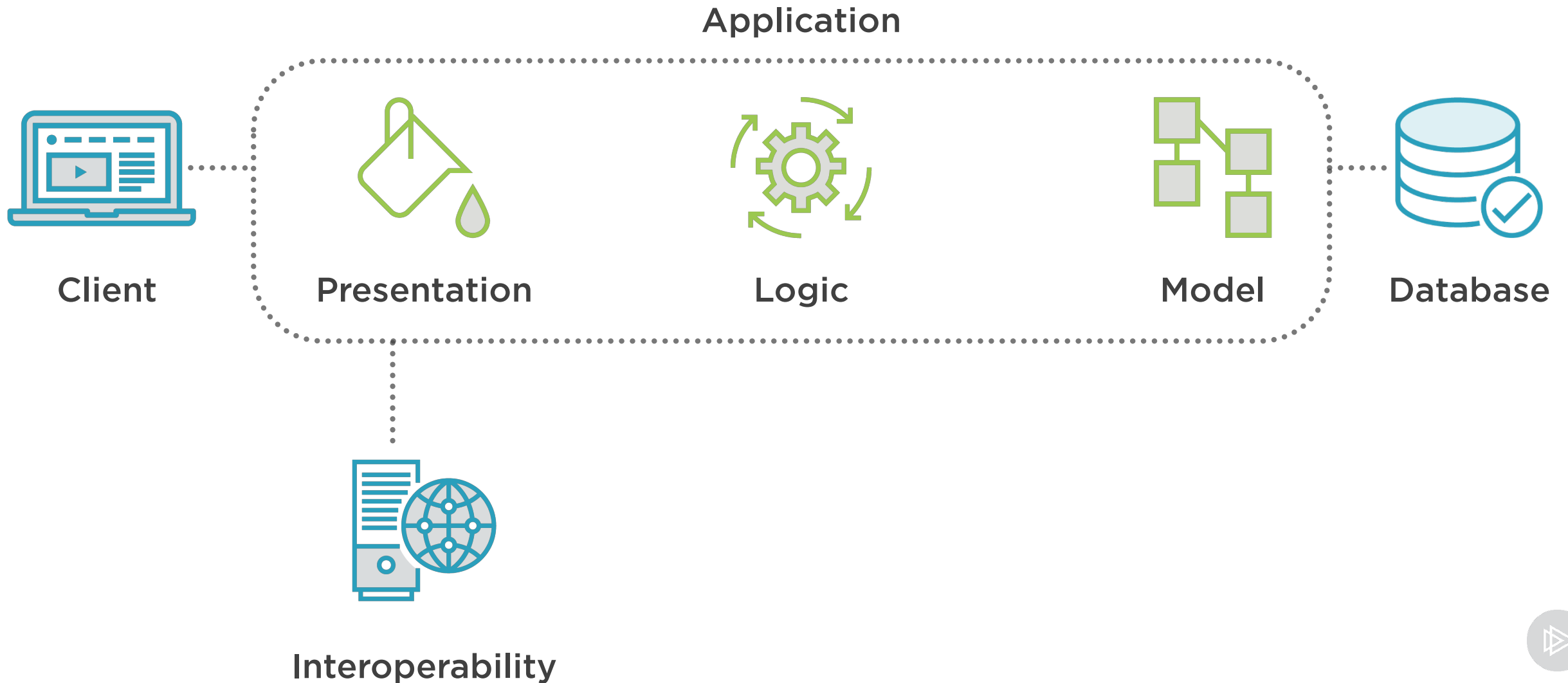


JPQL Queries

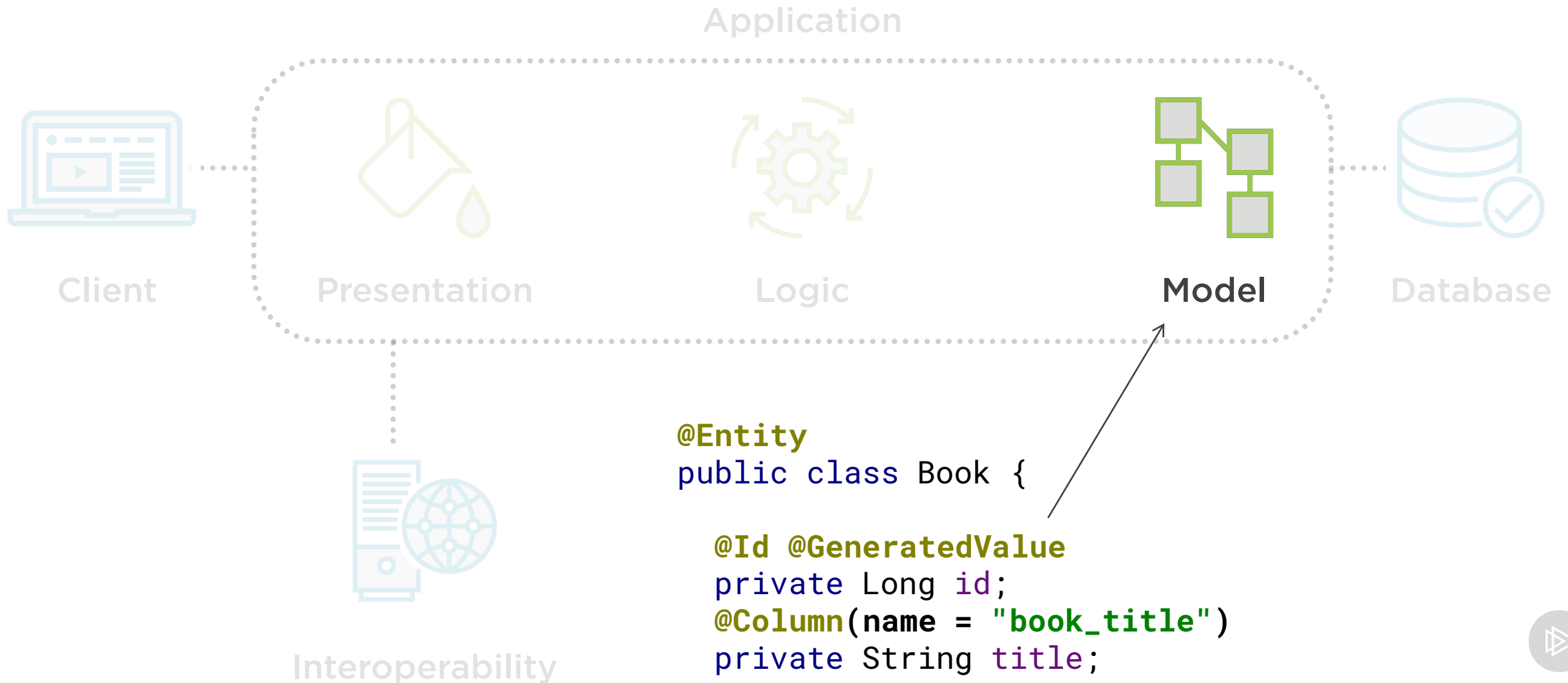
```
public class Main {  
  
    static EntityManagerFactory emf =  
        Persistence.createEntityManagerFactory("myPU");  
    static EntityManager em = emf.createEntityManager();  
  
    public static void main(String[] args) {  
  
        Book book = new Book(4044L, "H2G2", "Scifi Book", "5678-5678", 247);  
        em.persist(book);  
  
        book = em.find(Book.class, 4044L);  
  
        TypedQuery<Book> query = em.createQuery(  
            "Select b From Book b order by b.title", Book.class);  
        query.getResultList();  
    }  
}
```



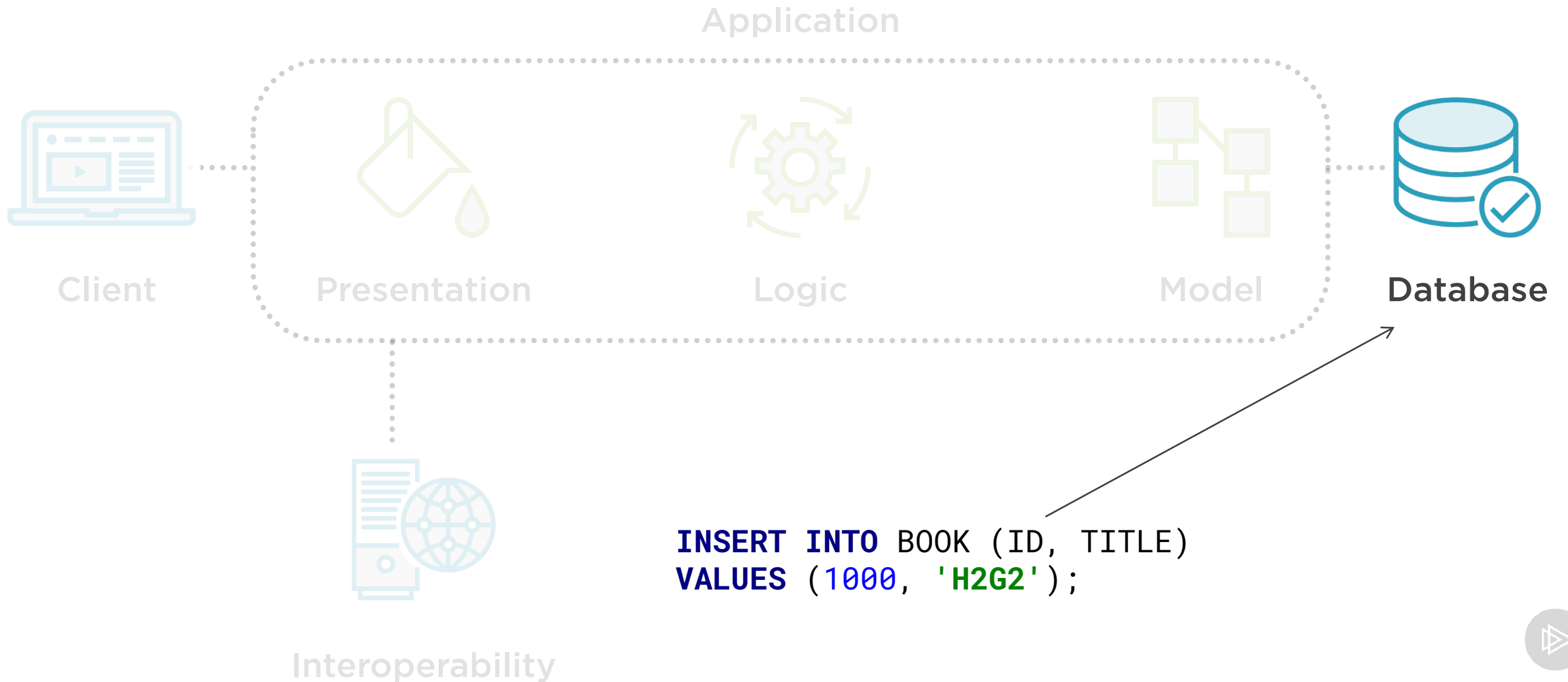
Where Can We Use JPA?



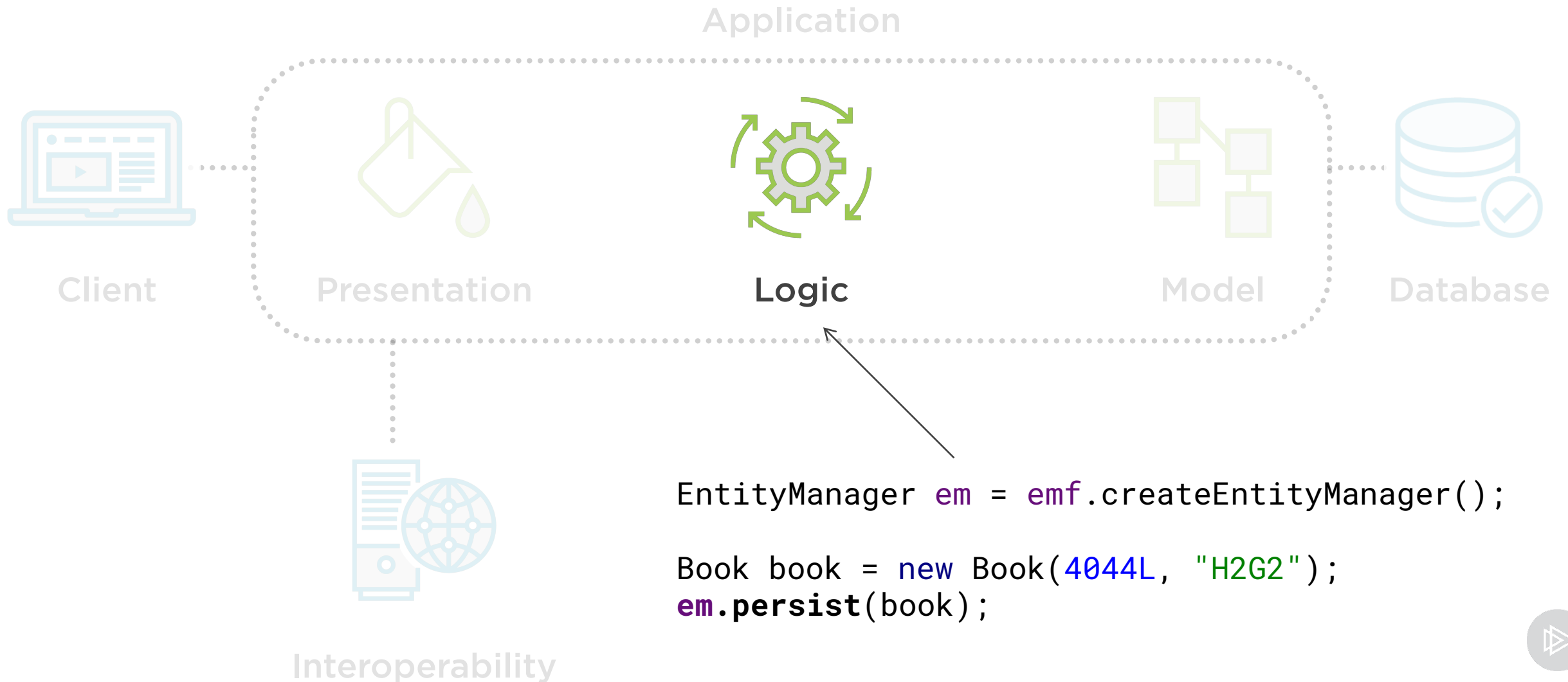
Business Model



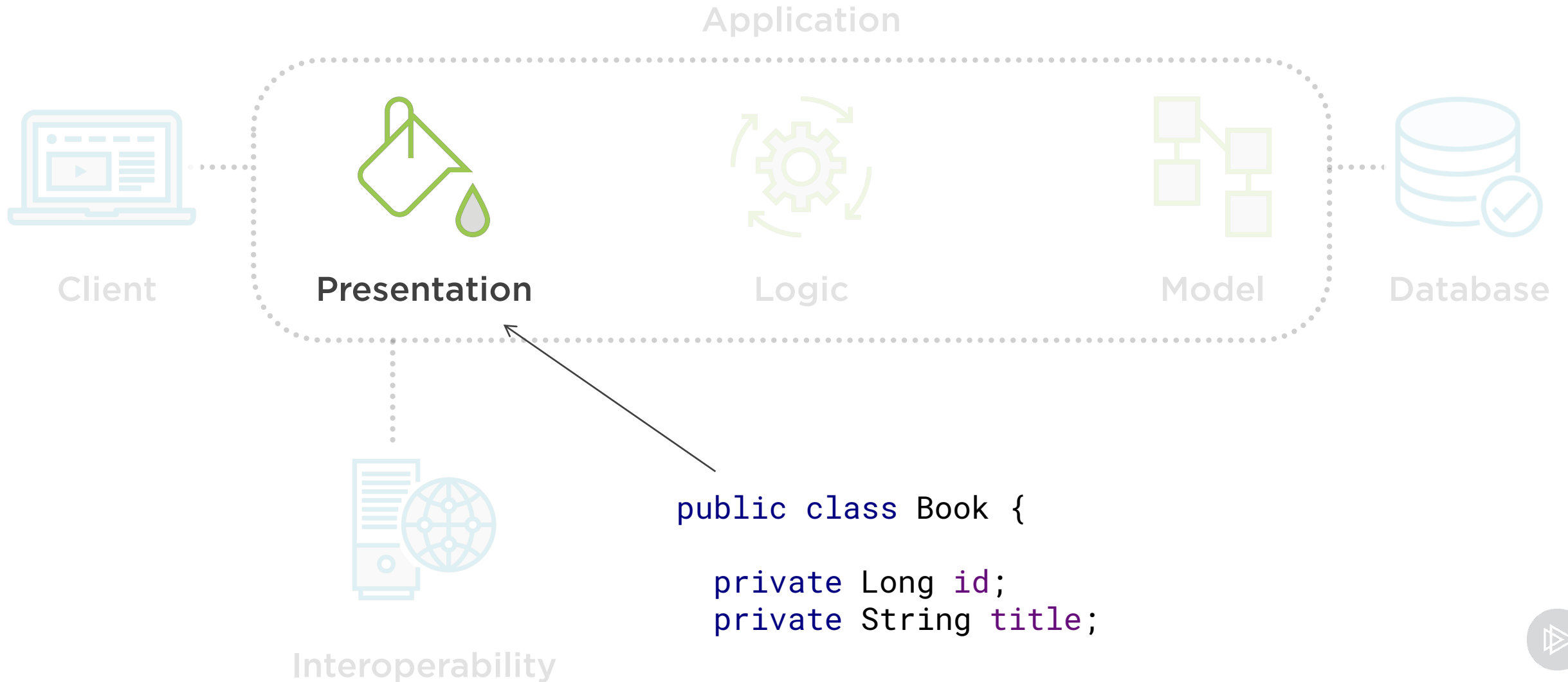
Relational Database



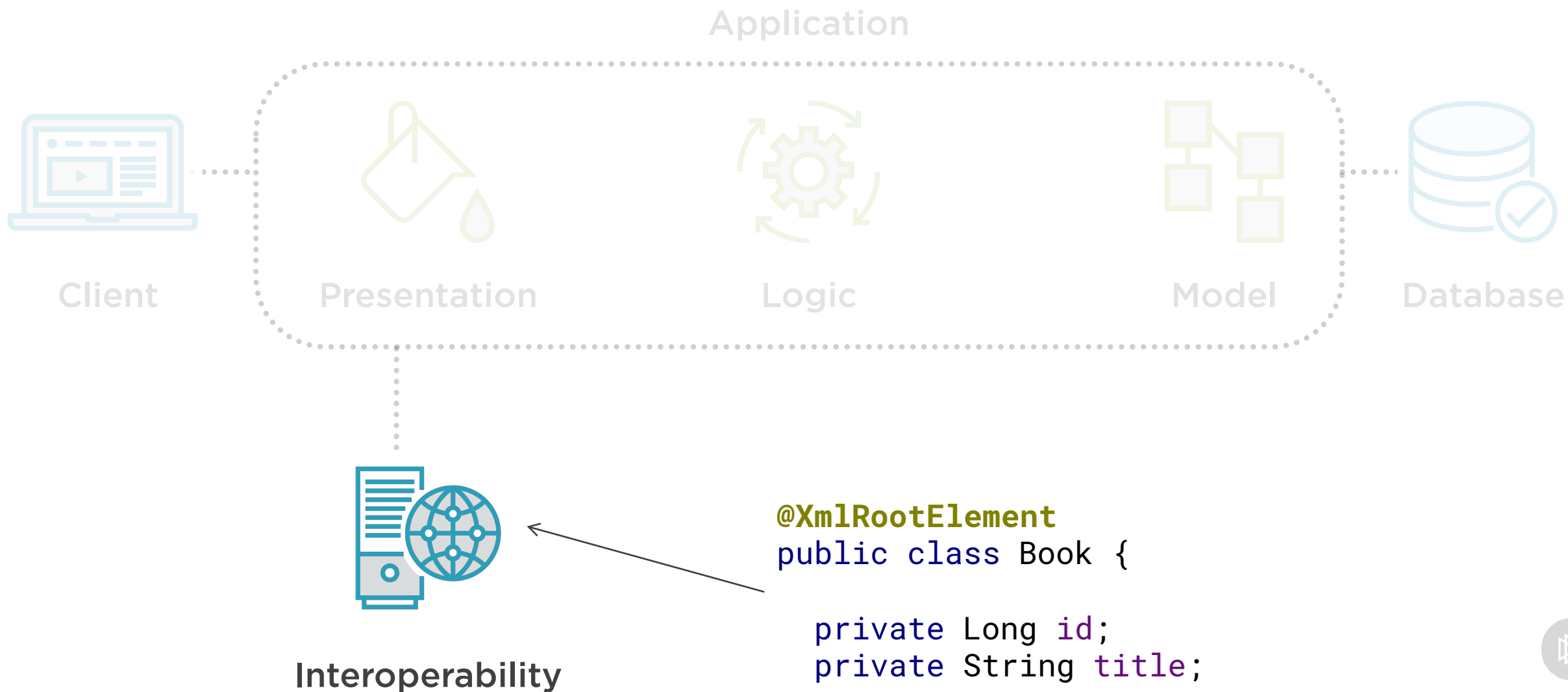
Business Logic



Presentation

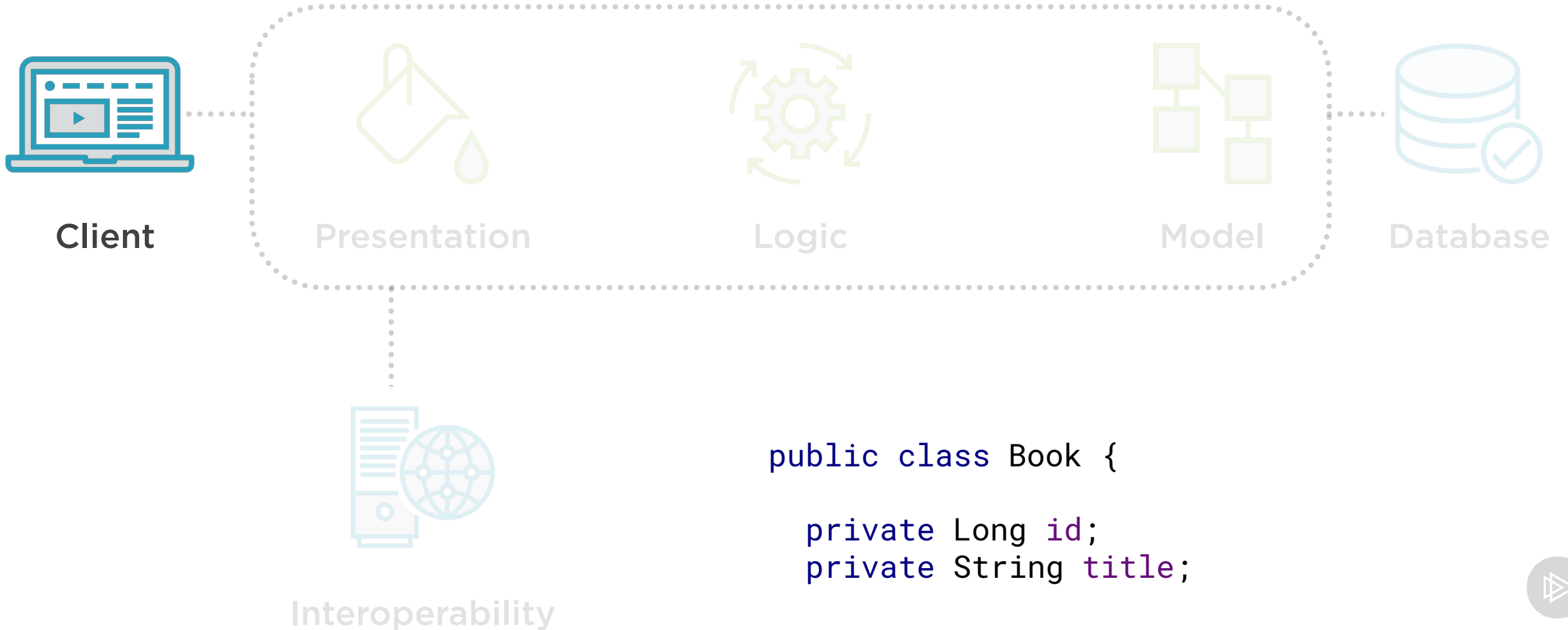


Interoperability

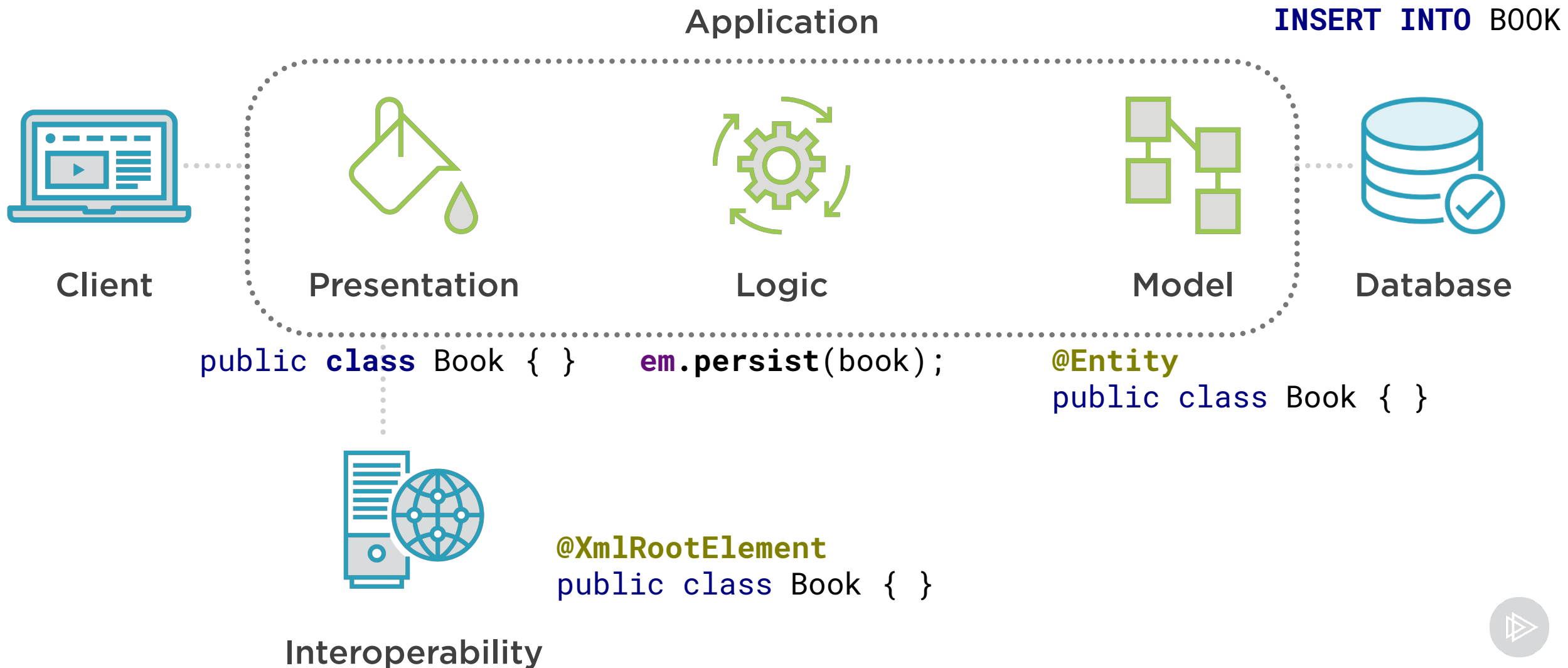


Client

Application



JPA in Our Tiered Architectures



Summary



What is a relational database?

What is Java Persistence API?

How to develop an entity?

How to manipulate an entity?

JPA in n-tier architecture

Next Module



Manipulating and mapping entities

Entity manager

Persistent unit

CRUD operations

Basic mapping metadata

