Data Transactions with Spring

MASTERING TRANSACTIONS

Kesha Williams
SOFTWARE ENGINEERING MANAGER & PROFESSOR

@KeshaWillz www.kesha.tech

Overview

Database transactions

Four key properties of transactions

- Atomicity
- Consistency
- Isolation
- Durability

Transaction boundaries

Concurrent transactions

TrackZilla

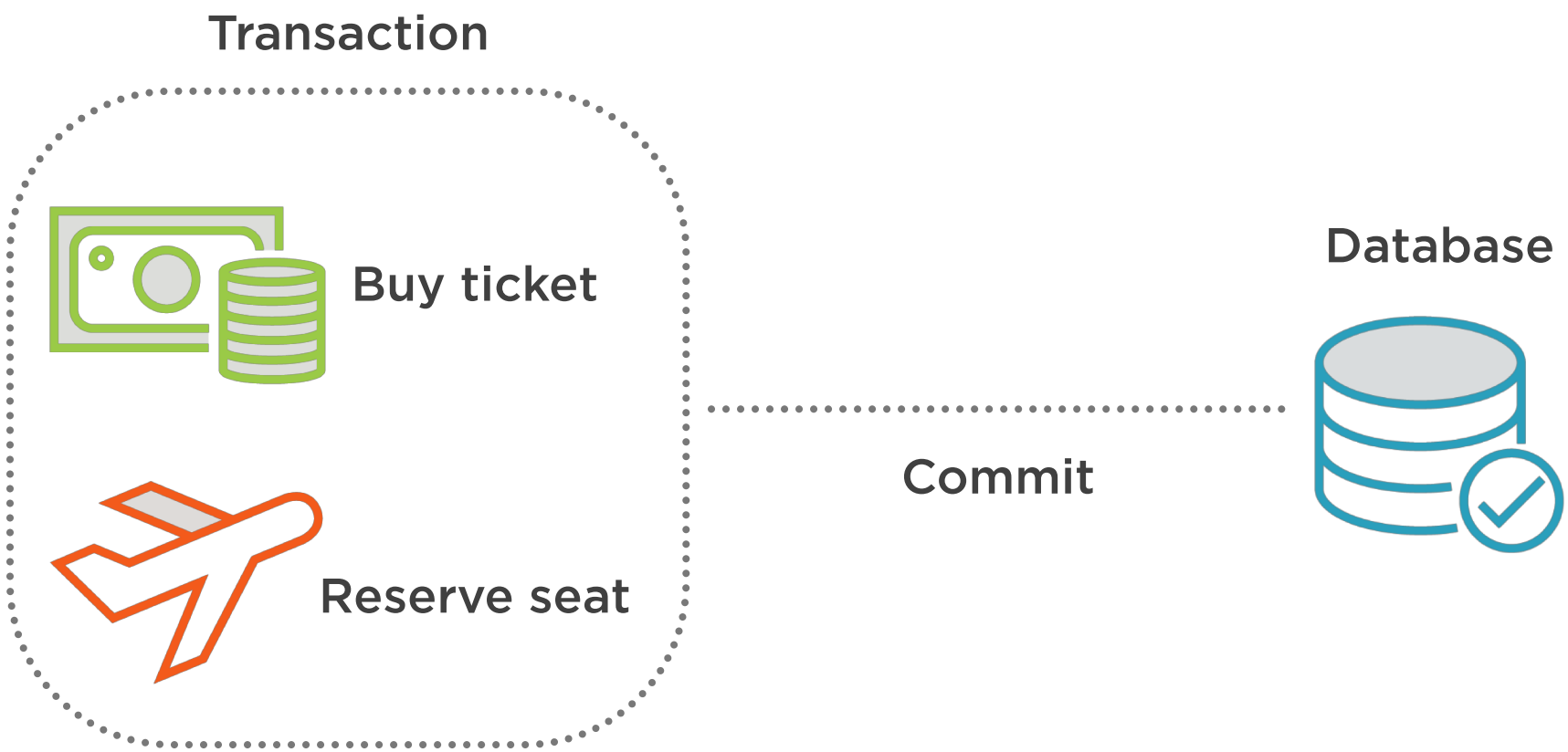
Bug tracking

Missing data

Manage transactions

Transactions

Transaction



Four Key Properties

Atomicity

Consistency

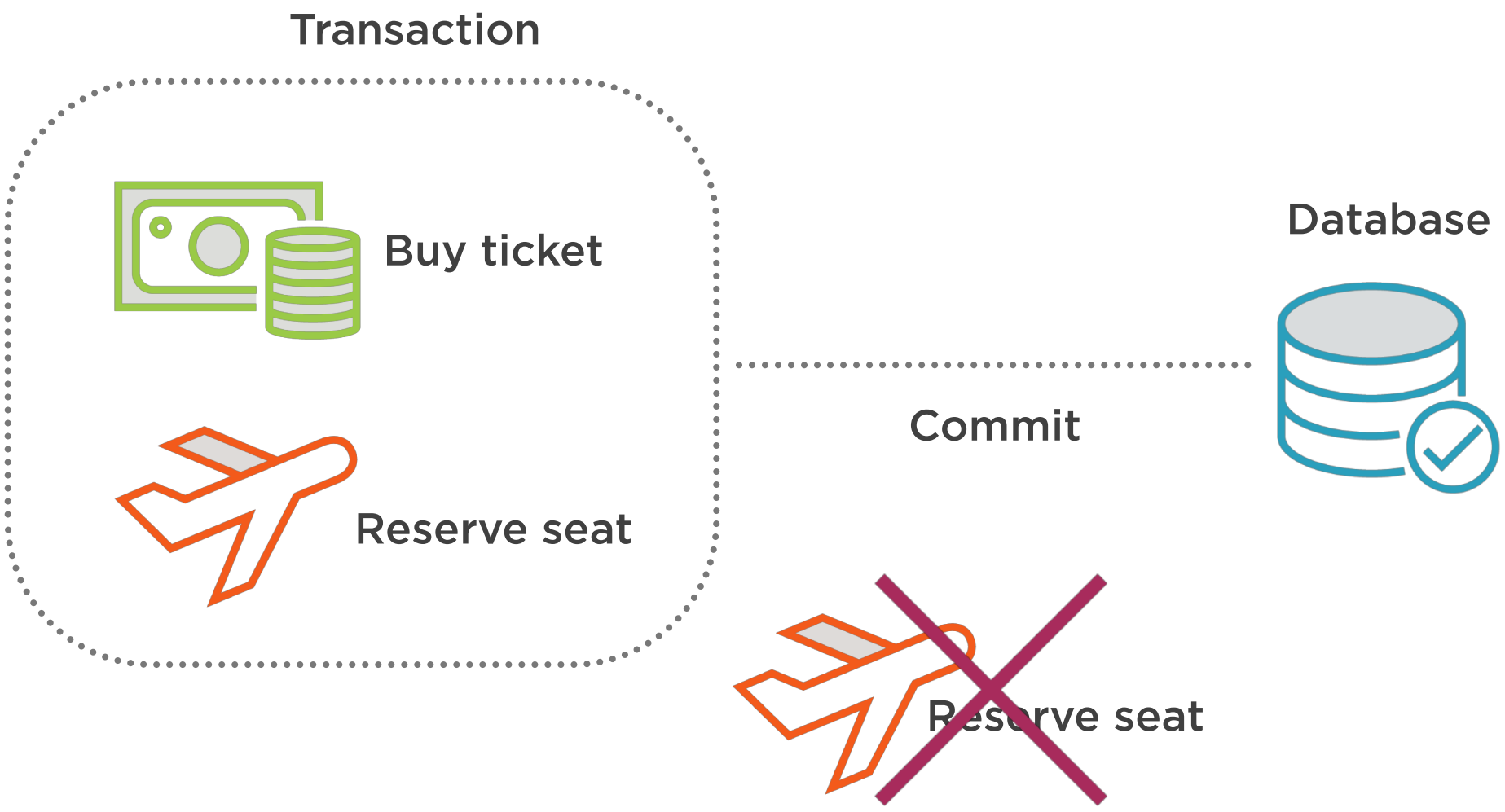
Isolation

Durability

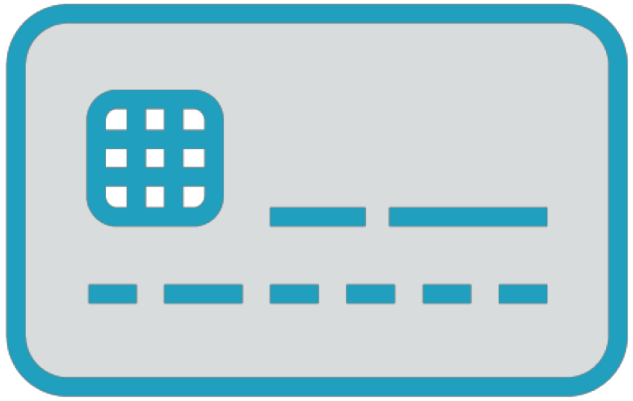
Transaction Management

Transaction management ensures data consistency and integrity.

Manage Transactions

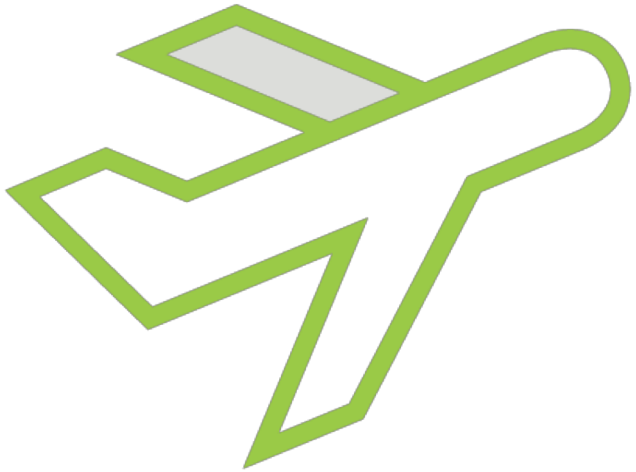


Transaction Example



Paying for a seat

Using a credit card to purchase a ticket online



Reserving a seat

Using the booking system to remove a seat from inventory

Booking Transaction

```
try {
    conn = dataSource.getConnection();
    conn.setAutoCommit(false);
    ticket.buy();
    seat.reserve();
    conn.commit();
} catch (SQLException e) {
    ...
    conn.rollback();
    ...
}
```

Transaction Types

Global transactions

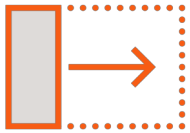
Multiple resources manage the transactions

Local transactions

One resource manages the transaction

Spring Framework Advantages

Spring Framework



Transactions managed seamlessly

Consistent programming model across global and local transactions

Benefits from different transaction management strategies

Transaction Management Types

Programmatic

Developer writes custom code to manage transactions and set boundaries

Declarative

Developer separates transaction management from business code

Isolation Levels

DEFAULT

Default from underlying database

READ_UNCOMMITTED

Read uncommitted changes

READ_COMMITTED

Read only committed changes

REPEATABLE_READ

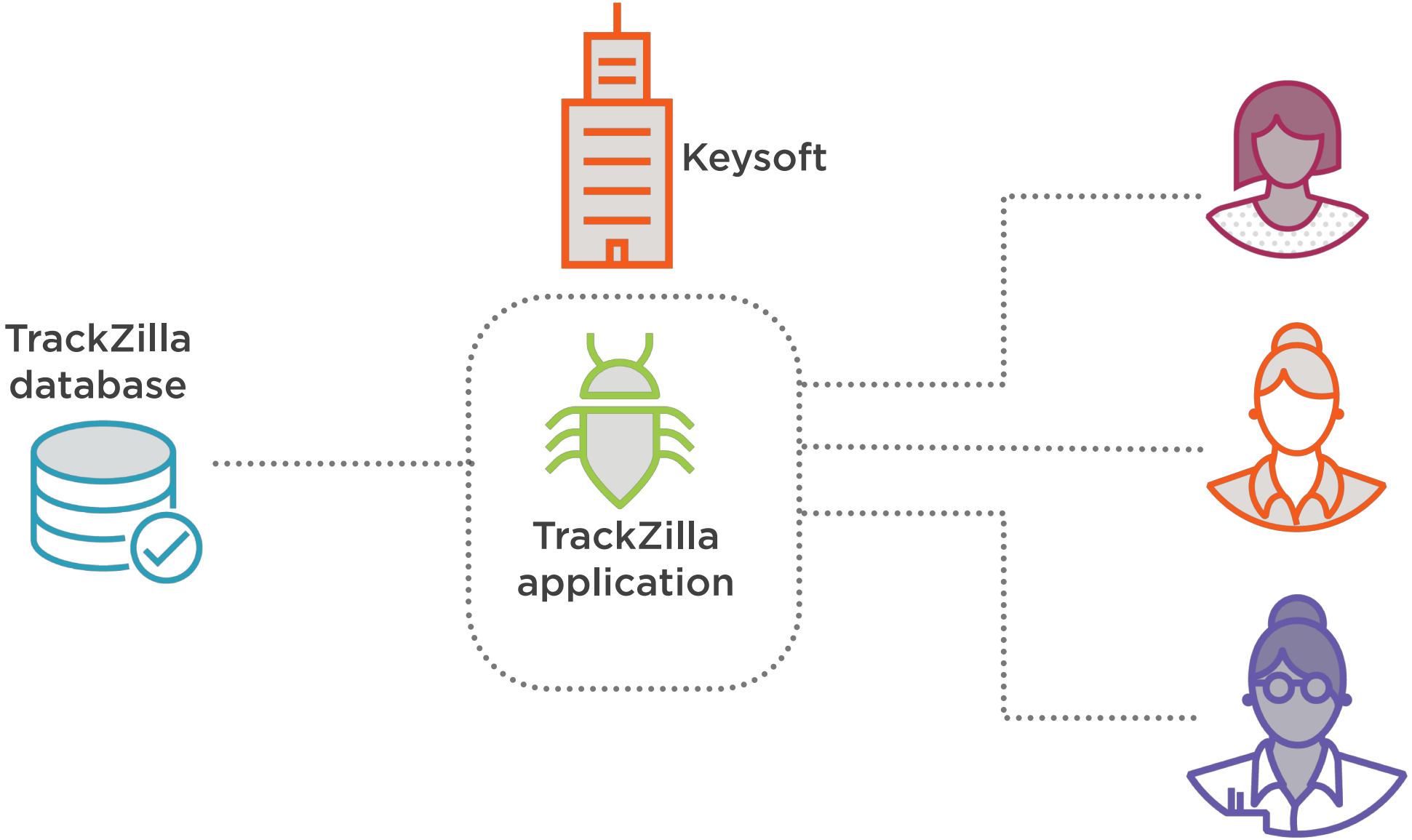
Read identical values multiple times

SERIALIZABLE

Read identical rows multiple times

Transaction Management in Action

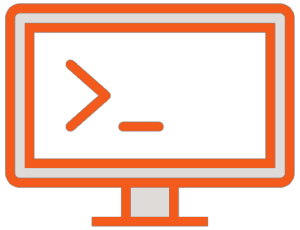
TrackZilla Case Study



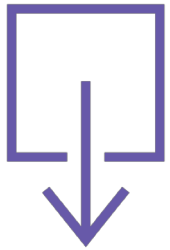
Prerequisites

JDK 1.8 +

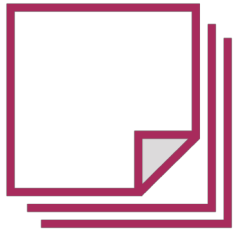
Text Editor



Maven 3.0 +

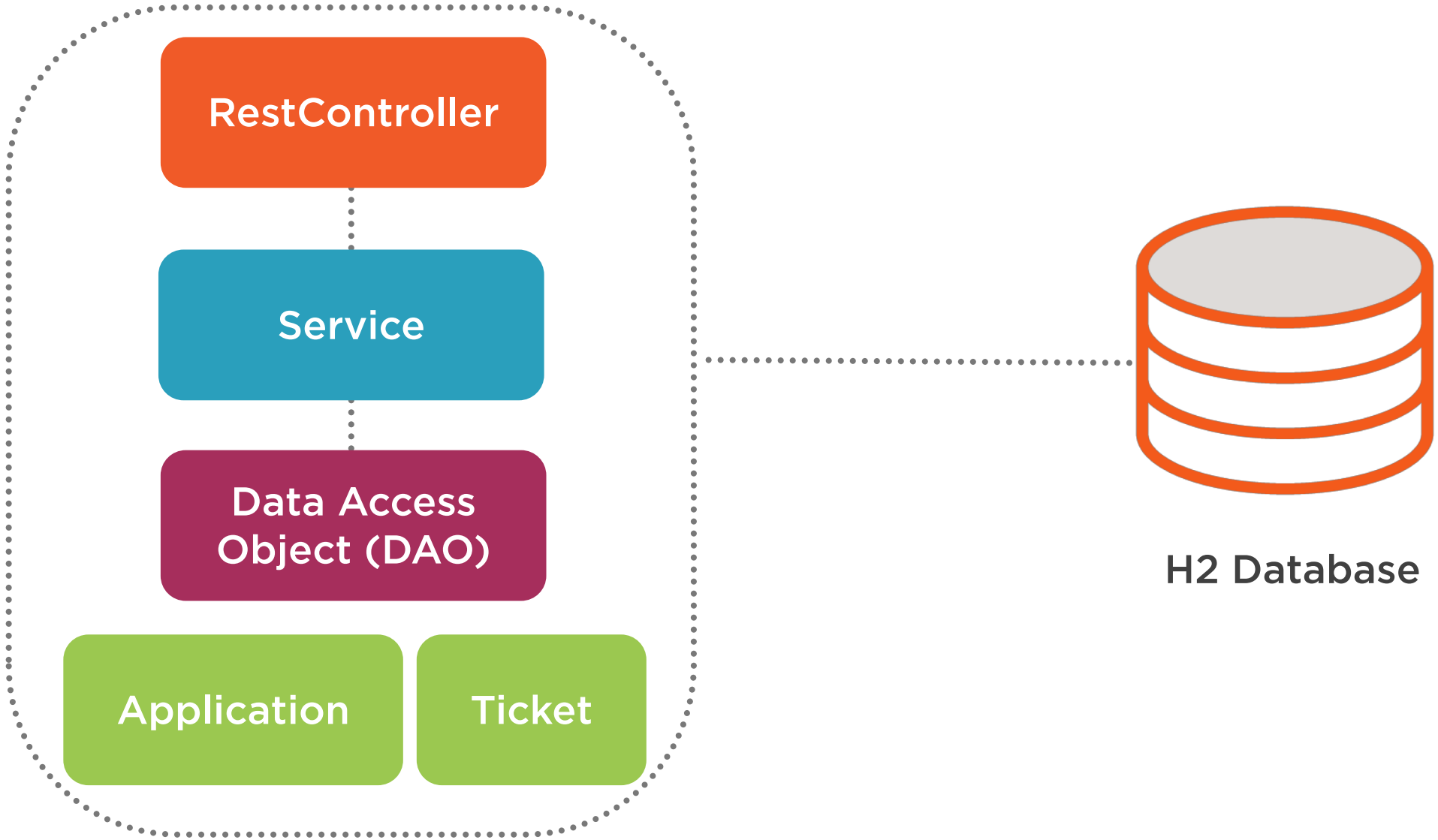


Git



Postman

TrackZilla Architecture



Demo

TrackZilla Spring Boot application

- Controller
- Service

Domain model

- Data Access Object

H2 database

Summary

Transactions

- Why they are important?

Spring's support for transactions

TrackZilla use case

Different transactions managers