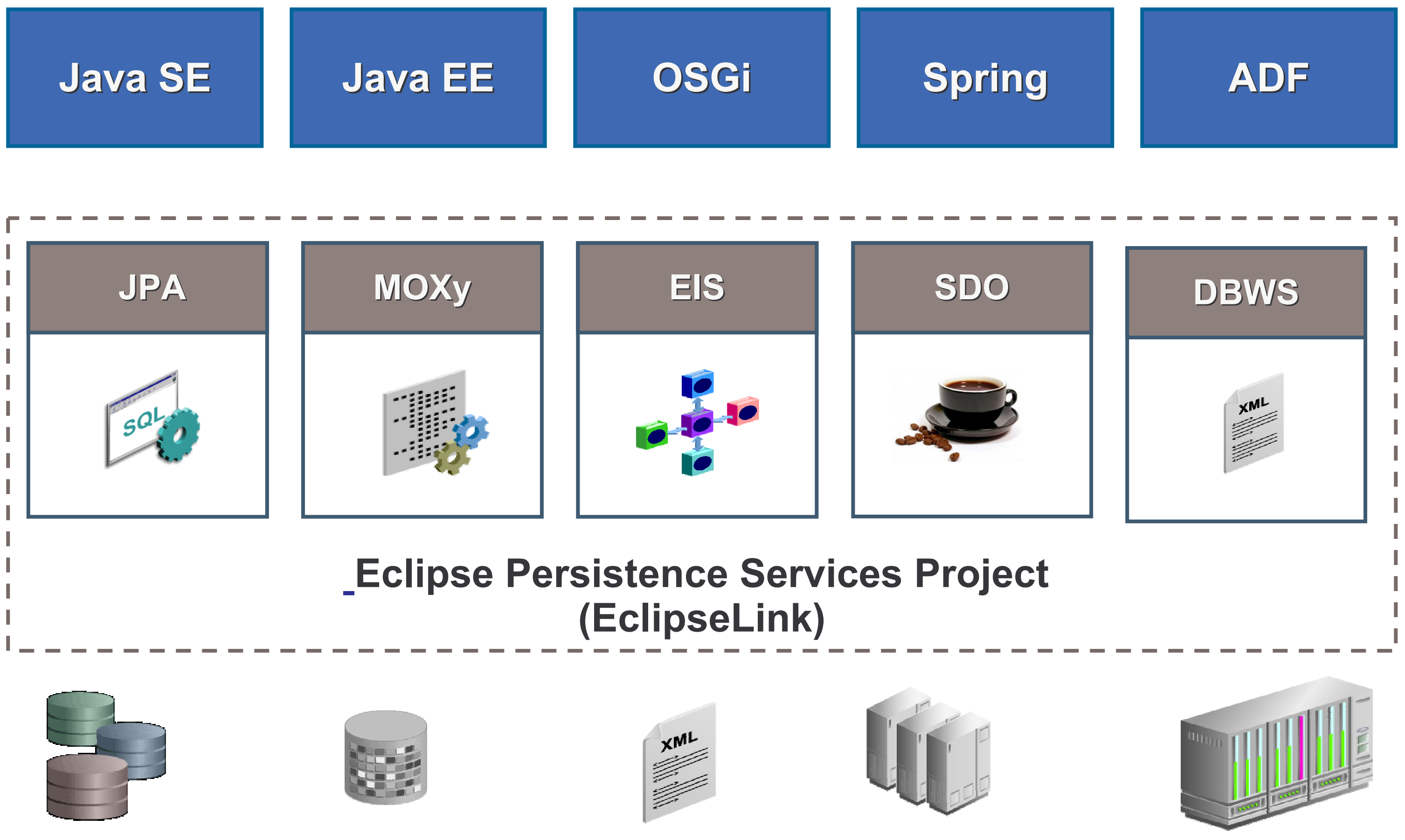
jax 09

Konferenz für Java™, Enterprise Architekturen, SOA

EclipseLink: High-Performance Java Persistence

Doug Clarke – Oracle Corp.

EclipseLink Project



EclipseLink 1.0

- JPA: Object-Relational
  - JPA 1.0 with many advanced features
  - Simplified configuration of using annotations and/or XML
  - All leading RDMS with platform specific features
    - Best ORM for the Oracle Database
- MOXy: Object-XML Binding (JAXB)
- SDO: Service Data Objects
- EIS using JCA Resource Adapters
- Containers
  - Java EE: WebLogic, OracleAS, WebSphere, GlassFish/SunAS, JBoss
  - Java SE, Web Containers, Spring, and OSGi

EclipseLink Developer Tool Support

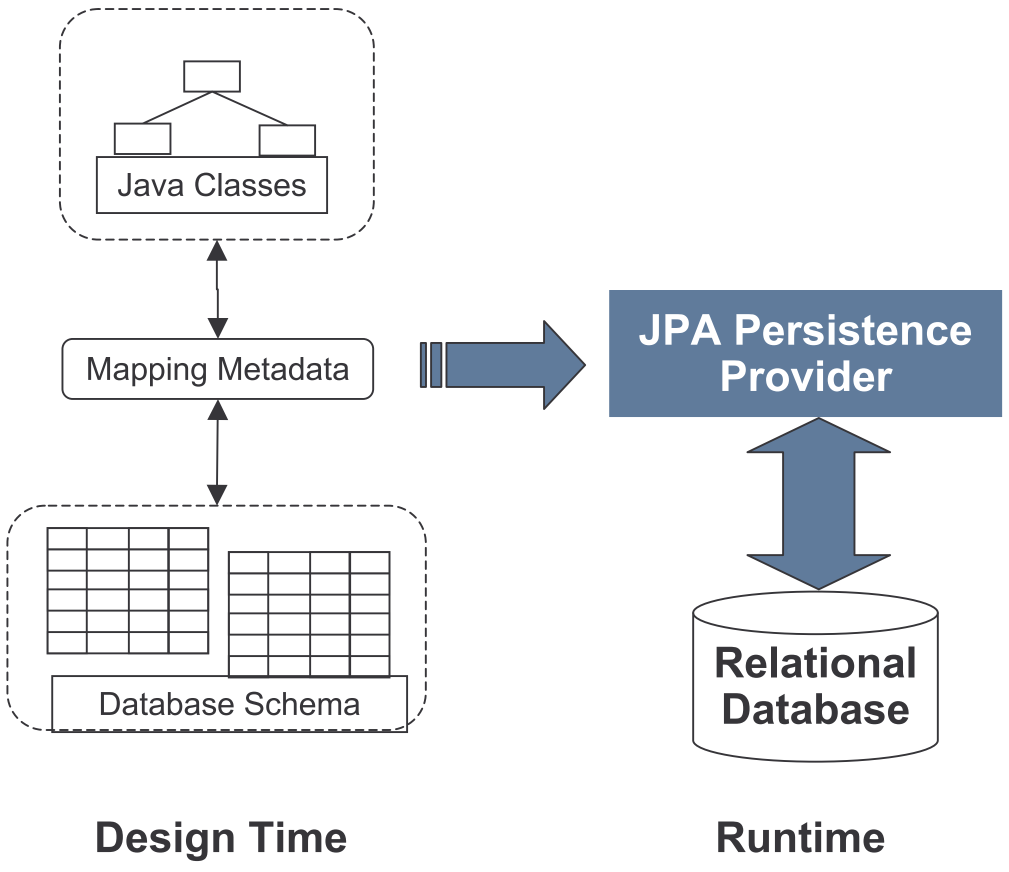
- EclipseLink is a Runtime Project but supported by IDEs
- Eclipse IDE
  - EclipseLink support included by Dali in Eclipse 3.4 (Ganymede)
- MyEclipse
- Oracle
  - JDeveloper 11g: JPA, Native ORM, OXM, and EIS mapping
  - Oracle Enterprise Pack for Eclipse (OEPE)
- NetBeans
- Standalone Workbench: Native ORM, OXM, EIS

EclipseLink: Distributions

- Oracle
  - TopLink 11g
  - WebLogic Server and Oracle AS
- Sun GlassFish v3 (SunAS)
  - Replaces TopLink Essentials (JPA 2.0 Reference Implementation)
- Spring Source
  - Spring Framework
  - Spring OSGi Bundle Repository
- Others actively investigating

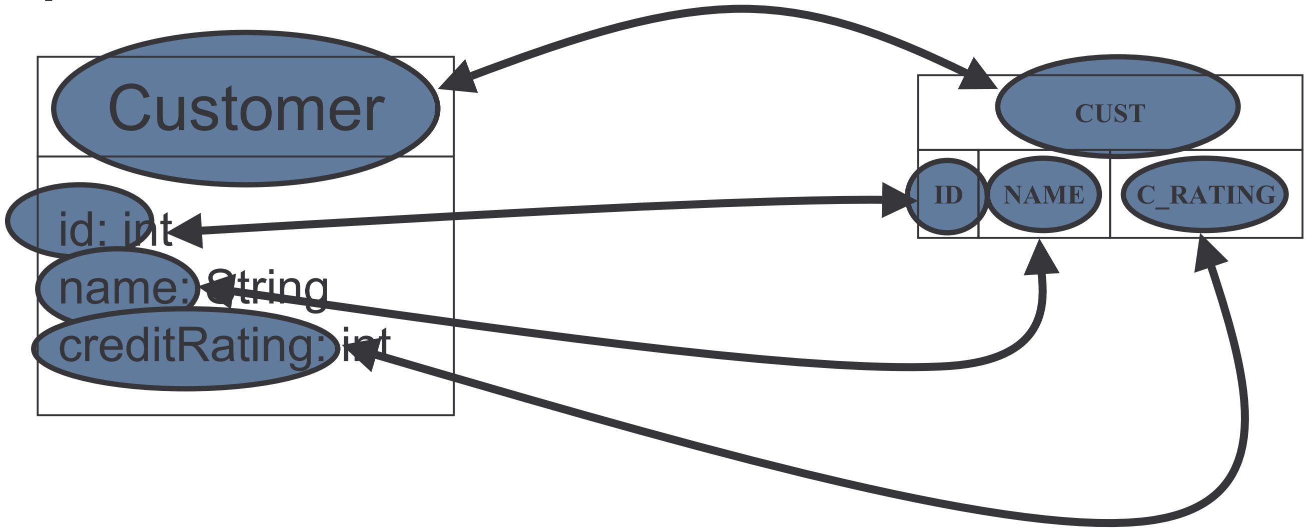
JPA: Object-Relational Persistence

Where does EclipseLink JPA fit?



Mapping

- The activity of ‘Mapping’ is the process of connecting objects/attributes to tables/columns



Standard JPA Mappings

- Core JPA Mappings
  - Id
  - Basic
  - Relationships
    - OneToOne
    - ManyToOne
    - OneToMany
    - ManyToMany
  - And more…
- Annotations and/or XML

EclipseLink JPA Config

- Standard JPA 1.0 (portable)
  - Persistence.xml with EclipseLink properties
  - Mapping: Annotations and/or orm.xml
  - Query hints
- EclipseLink JPA
  - Standard JPA +
  - EclipseLink annotations
  - EclipseLink orm.xml

Advanced Mapping Example

```
@Entity
@Cache(type=SOFT_WEAK, coordinationType=SEND_OBJECT_CHANGES)
@OptimisticLocking(type=CHANGED_COLUMNS)
@Converter(name="money", converterClass=MoneyConverter.class)
public class Employee {
  @Id
  private int id;

  private String name;

  @OneToMany(mappedBy="owner")
  @PrivateOwned
  private List<PhoneNumbers> phones;

  @Convert("money")
  private Money salary
  …
}
```

Flexible Mappings

- @BasicCollection - stores a collection of simple types, such as String, Number, Date, etc., in a single table
- @BasicMap - stores a collection of key-value pairs of simple types, such as String, Number, Date, etc., in a single table
- @PrivateOwned - supports cascade delete
- @JoinFetch - enables the joining and reading of a referenced object(s) in the same query as the source object
- @Mutable - indicates that the value of a complex field itself can be changed or not changed (instead of being replaced)
- @Transformation - enables the mapping of a single field to to one or more database columns.
- @VariableOneToOne - supports OneToOne mappings to an interface rather than an Entity
- @ReadOnly - makes an Entity read only

Custom Data Types & Conversions

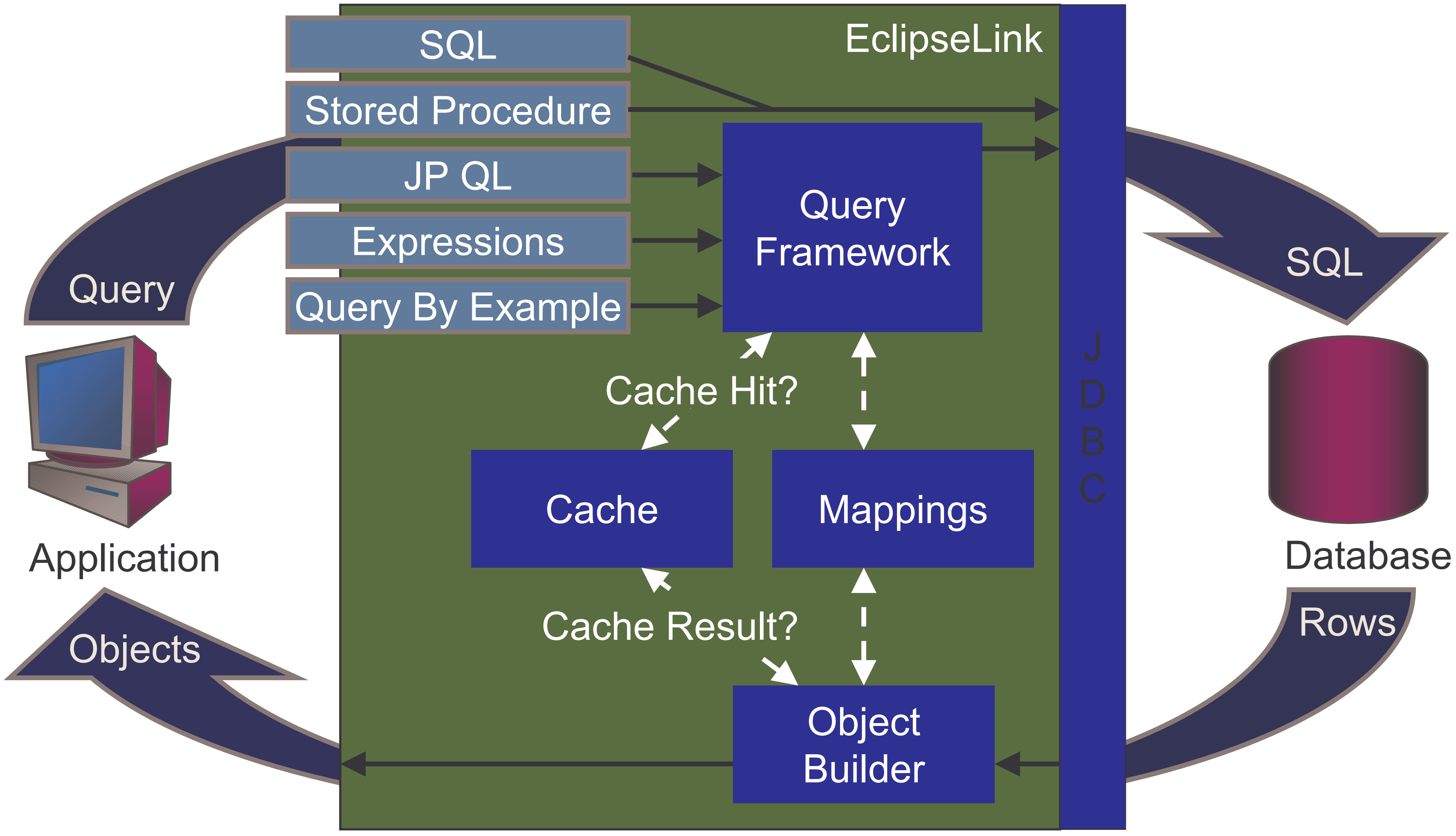
- New converter mappings for type conversion and user defined types include:
  - @Converter, @TypeConverter, @ObjectTypeConverter
  - @StructConverter
  - @Convert

```
@Entity
@Converter(
      name="Currency",
      converterClass=CurrencyConverter.class)
public class Employee {
  @Convert("Currency")
  private Currency salaryCurrency;
```

Query Framework

- Queries can be defined using
  - Entity Model: JPQL, Expressions, Query-by-example
  - Native: SQL, Stored Procedures
- Customizable
  - Locking, Cache Usage, Refreshing
  - Optimizations: Joining, Batching, parameter binding
  - Result shaping/conversions
  - Stored Procedure support

EclipseLink Query Execution



Stored Procedure Query

- Stored procedure usage has been simplified through the
  - **@NamedStoredProcedureQuery**
  - **@NamedStoredProcedureQueries**
- These annotations encapsulate stored procedure calls as named queries.
- Client code is unaware that the query they are executing is a stored procedure and not a JP QL or native SQL query

```
ERROR: undefined
OFFENDING COMMAND: G00GFFEncoding

STACK:

/Encoding
```