

ORACLE®

Oracle Coherence HotCache

An Overview

Dave Felcey
Coherence Product Manager

November 2014

ORACLE

Agenda

- 1 Overview
- 2 HotCache
- 3 Data Mappings and Transformations
- 4 Performance, Scalability and HA
- 5 Demonstration

Agenda

- 1 Overview
- 2 HotCache
- 3 Database Mappings and Transformations
- 4 Performance, Scalability and HA
- 5 Demonstration

Oracle Coherence HotCache Components

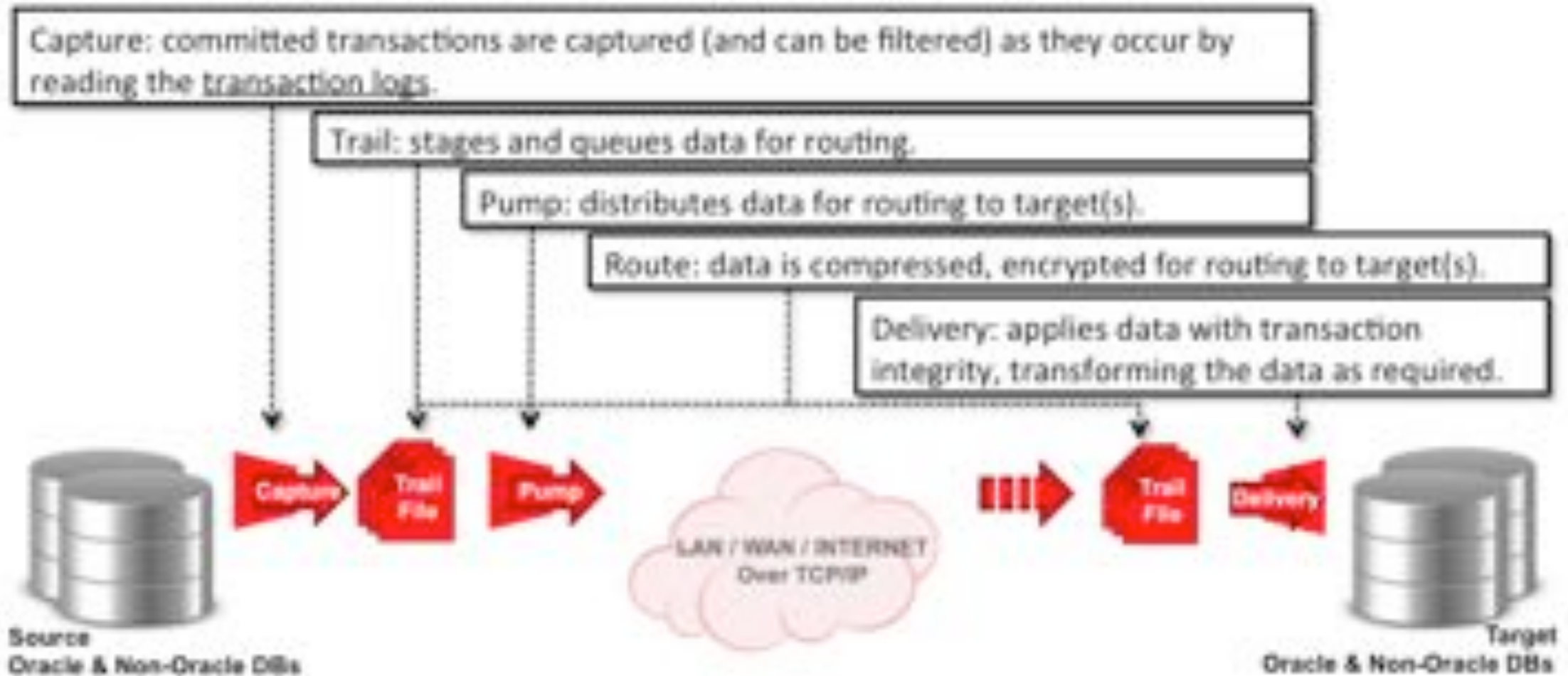
- Oracle Coherence 12.1.2+
- Oracle GoldenGate and Java Adaptor
- JPA (Java Persistence API)
- Oracle TopLink (Oracle JPA provider bundled with Coherence)
- Database (currently only Oracle Database supported)

Oracle GoldenGate Defined

No-Impact, Real-Time Data Integration & Transactional Replication

- Oracle GoldenGate is a real-time, log-based change data capture and replication software platform
- The software provides:
 - Capture
 - Routing
 - Transformation
 - Delivery
- of transactional data across heterogeneous data sources in real time

How Oracle GoldenGate Works



Oracle GoldenGate

Differentiator's

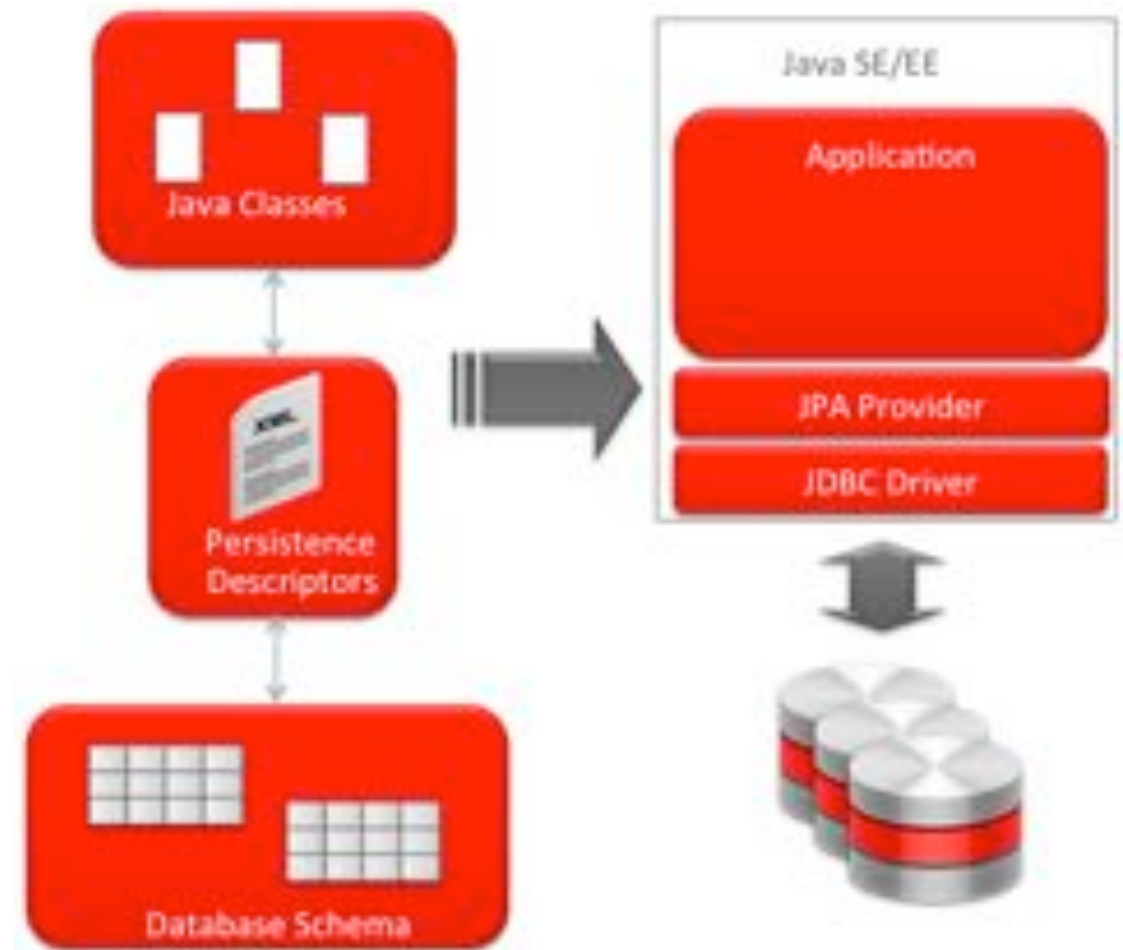


Oracle GoldenGate Supported Databases

| Databases | O/S and Platforms |
|--|---|
| Oracle GoldenGate Capture and Delivery: <ul style="list-style-type: none">• Oracle• DB2• Microsoft SQL Server• Sybase ASE• Teradata• Enscribe• SQL/MP• SQL/MX• MySQL v 5.5• JMS message queues | Linux Sun Solaris v11 Windows 2000, 2003, XP HP NonStop HP-UX IBM AIX v7.1 IBM z Series zLinux |
| Oracle GoldenGate Delivery: <ul style="list-style-type: none">• All listed above, plus:• TimesTen, Postgres• Netezza, Greenplum, HP Neoview• ETL products | IBM i Series |

Java Persistence API (JPA) - in a Nutshell

- Defines:
 - How Java objects are stored in relational database
 - A programmer API for reading, writing, and querying persistent Java objects (“Entities”)
 - A full featured query language in JP QL
 - A container contract that supports plugging any JPA runtime in to any compliant container



JPA Mappings

- Java standard Java Persistence API (JPA) mappings to map from database changes to affected objects

```
<entity class="example.model.Employee">
  <attributes>
    <id name="id">
      <generated-value />
    </id>
  </attributes>
</entity>
```

XML Mappings

OR

```
@Entity
public class Employee implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    private int id;
    private String firstName;
    private String lastName;
    private float salary;
}
```

Java Annotations

Oracle TopLink

Object Relational Mapping

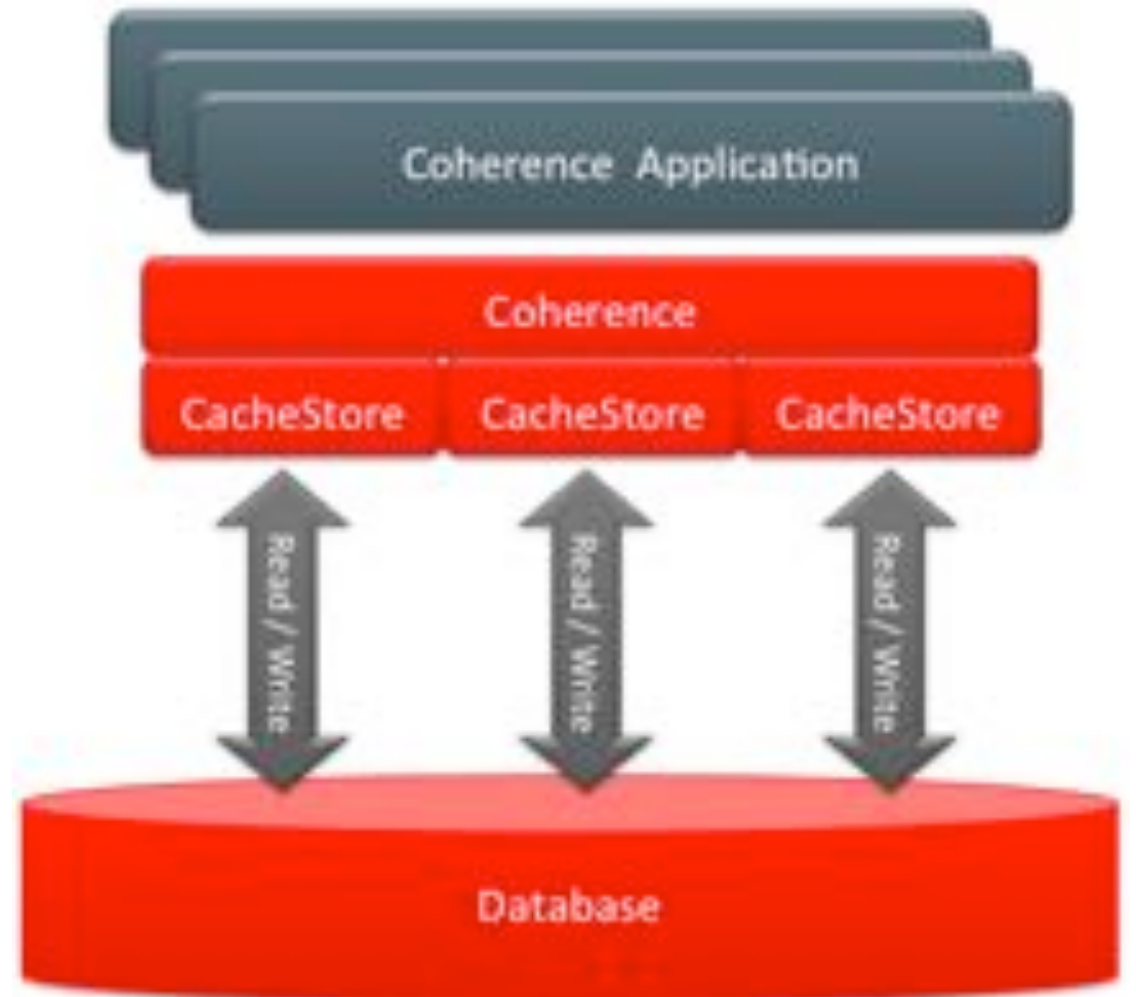
- Comprehensive Java Persistence Solution
 - Java Persistence (JPA 2.0 Reference Implementation), Java Architecture for XML Binding (JAXB 2.x), Database Web Service (DBWS)
- Strong Developer Support
 - Developed in open source EclipseLink project at Eclipse.org
 - Tool support in OEPE (Eclipse EE), NetBeans, JDeveloper
- Performance
 - Key contributor to Oracle's SpecJ World Record
 - Superior results across the board against key competitors in internal benchmarks

Agenda

- 1 Overview
- 2 HotCache**
- 3 Database Mappings and Transformations
- 4 Performance, Scalability and HA
- 5 Demonstration

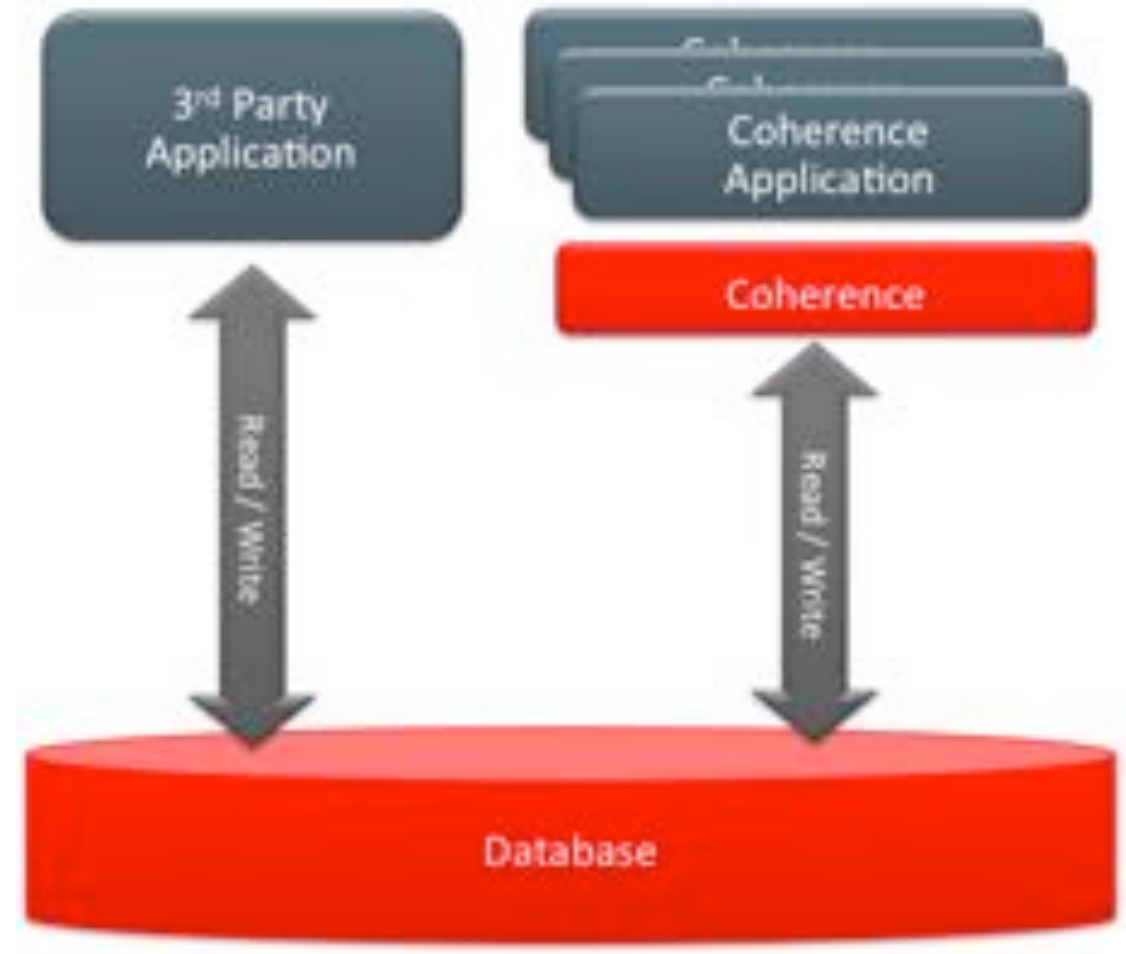
Database Backed Coherence Application

- Database access/update via CacheStore read/write
- Works well when Coherence application is the only source of database changes



Challenge: Shared Database

- Coherence applications backed by a relational database can suffer from stale caches due to 3rd party database updates.
- Existing Coherence solutions:
 - Cache Expiry
 - Inefficient—expires fresh data
 - Refresh Ahead
 - Inefficient pull model refreshes fresh data
 - Possible high latency

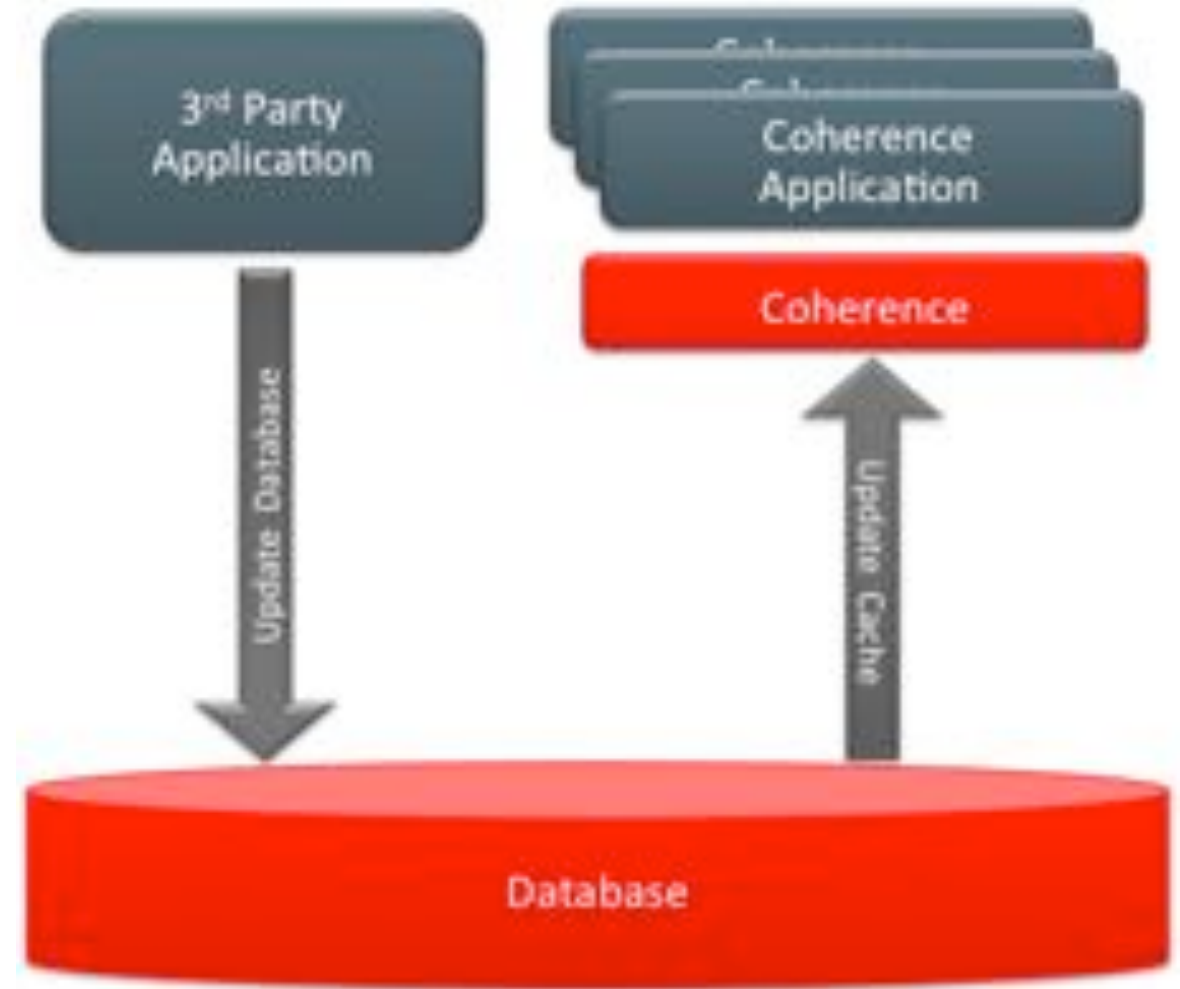


“Custom” Alternative Approaches

- Custom Solutions
 - Queuing and triggers
 - Dual updates
 - Oracle JDBC DCN
 - Polling
- All have some or all of the below limitations
 - Require database changes
 - Have limited scalability or latency characteristics
 - Are database specific (like Oracle AQ)
 - Can miss transactions
 - Need duplicates to be dealt with to provide HA

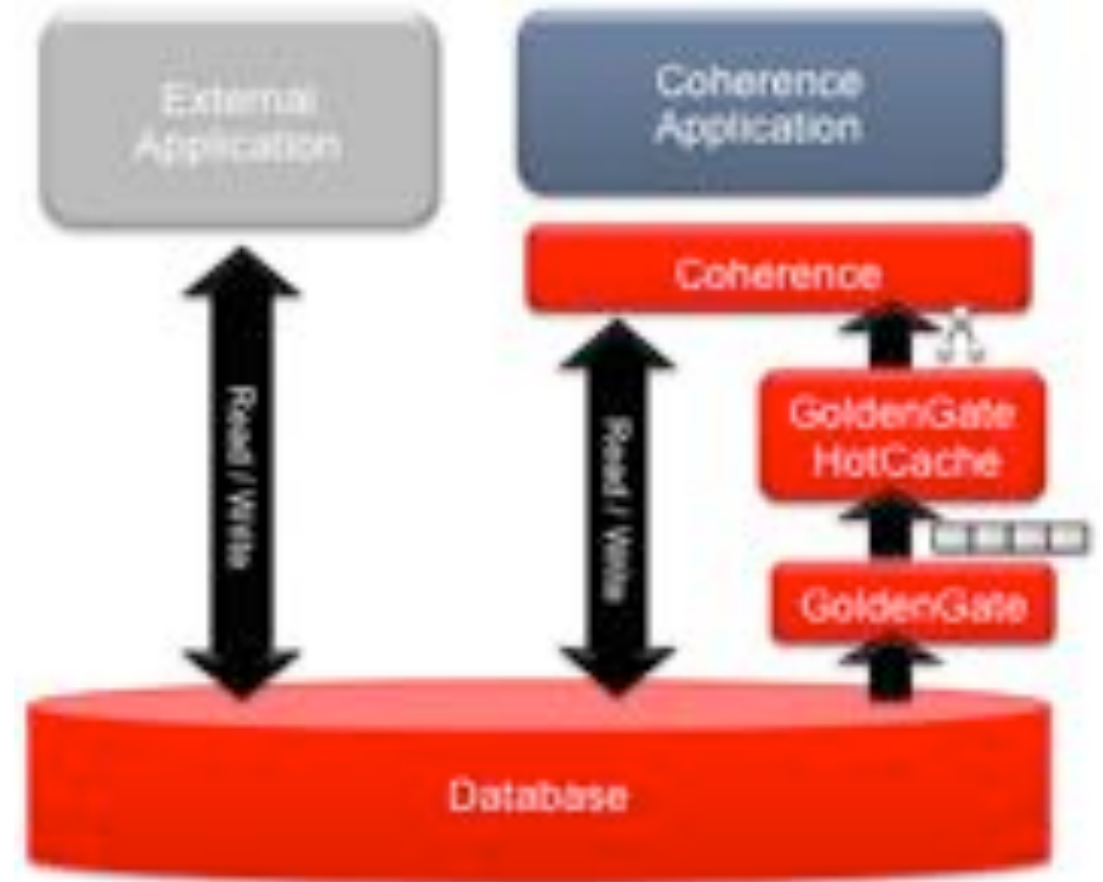
Ideal Solution: Update Cache on database change

- Ideally
 - Event Driven
 - Low Latency
 - Efficient
 - Non invasive
 - **Understand what objects in what caches are affected by a change to a given column in a given table.**



Coherence “HotCache”

- Built on TopLink, it integrates with GoldenGate to push database changes into Coherence cache
 - Maps data changes to affected objects using JPA metadata
 - Zero to minimal change to Coherence application
 - Efficient push model—only stale data is processed
 - Low latency—push happens on change
 - No database changes required



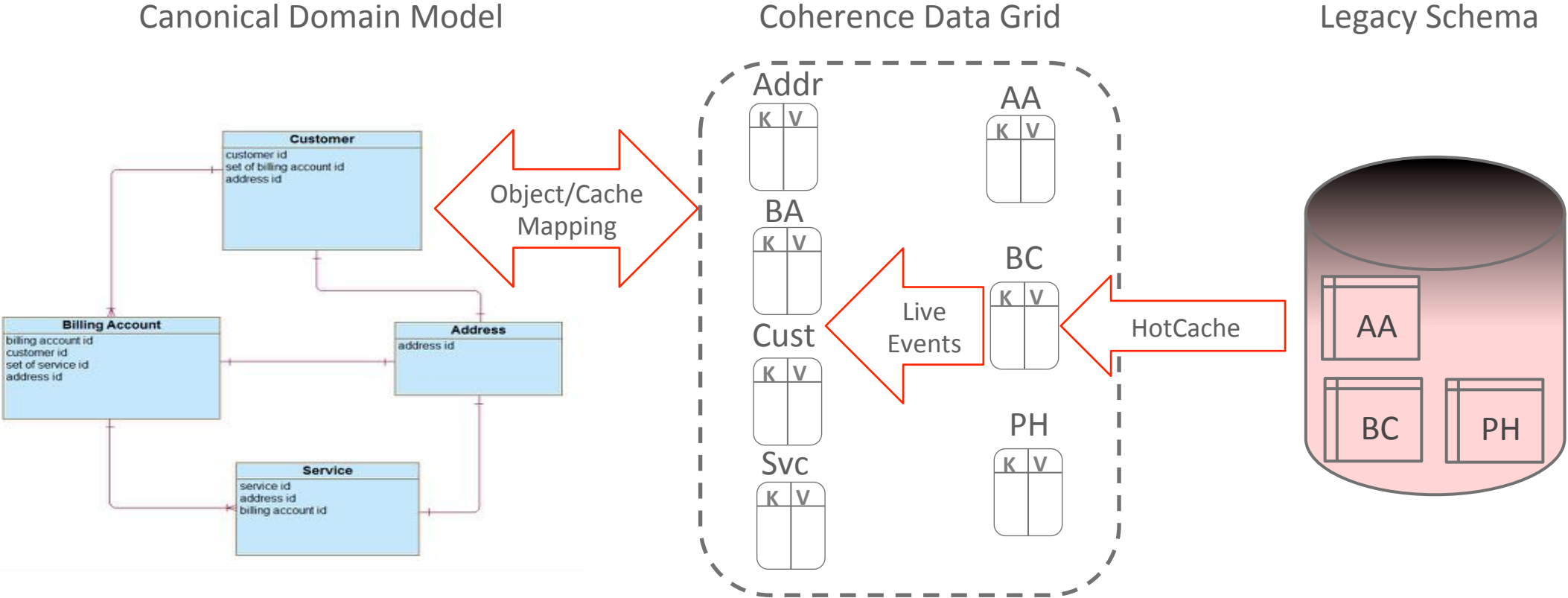
Coherence HotCache and JPA

- Coherence GoldenGate Adapter identifies what objects in what cache need to be updated based on database change
- Coherence Entry Processors are used to update data in place
- JPA optimistic locking supported
 - Only applies change if newer than cache value
 - Sequence or timestamp fields can be
- JPA converter's can be used to transform data types

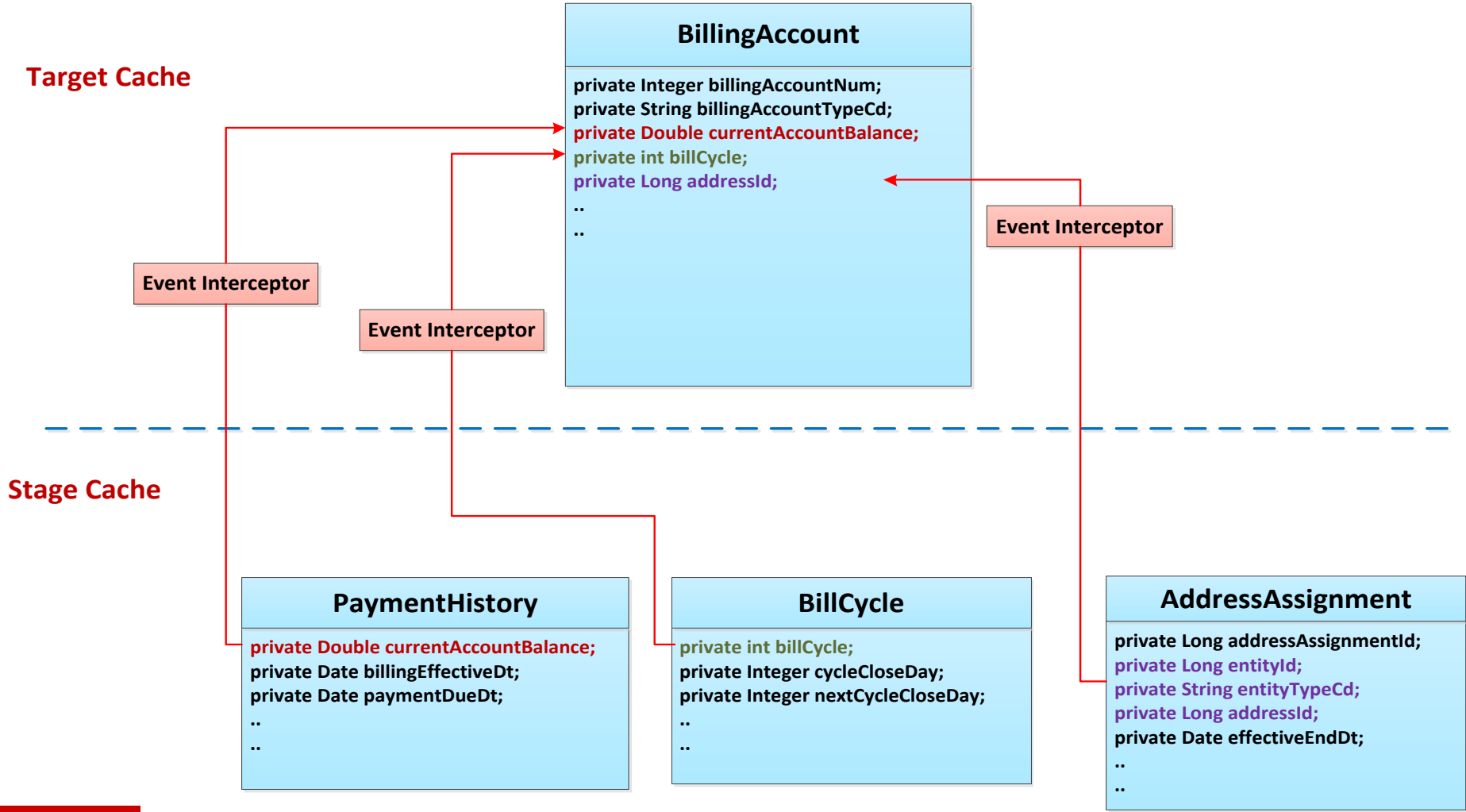
Agenda

- 1 Overview
- 2 HotCache
- 3 Database Mappings and Transformations**
- 4 Performance, Scalability and HA
- 5 Demonstration

Data Transformation with Coherence Live Events



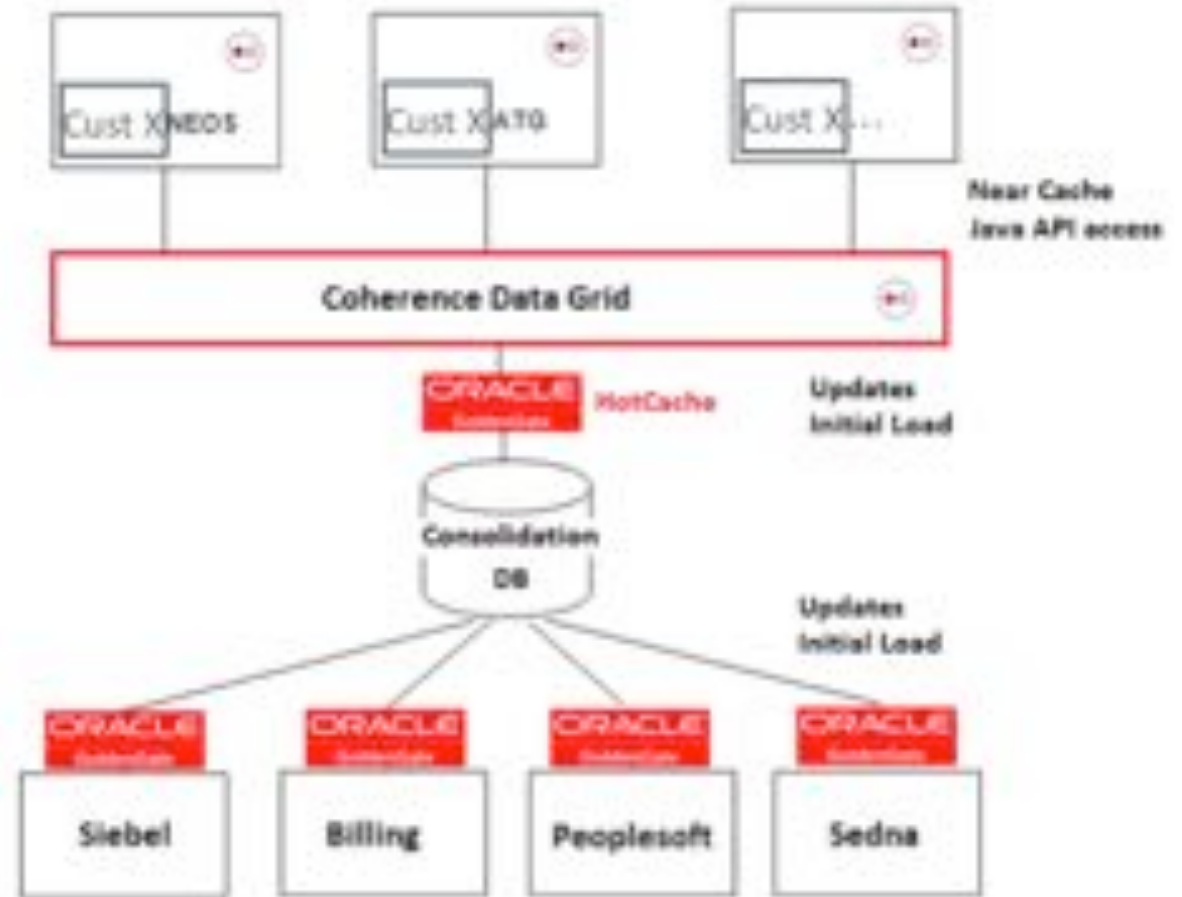
Update Dependent Objects with Event Interceptors



Data Transformations in a Database

Intermediate Database for Staging Tables

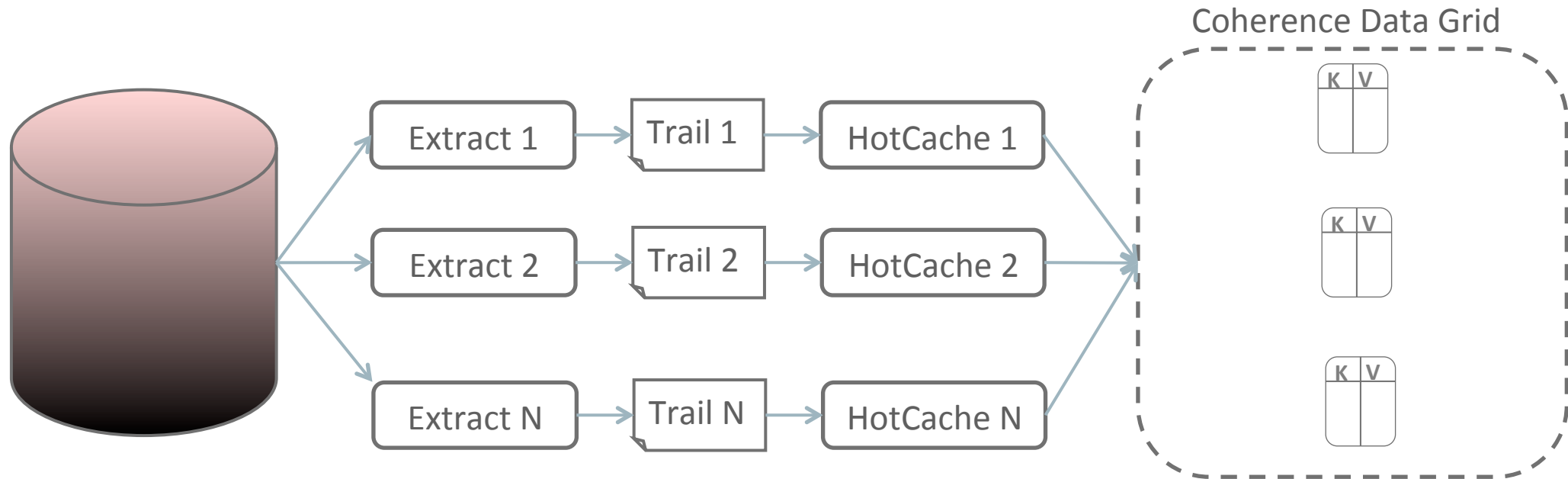
- Real-time cache to protect enterprise resources
- Intermediate database used to transform and de-normalize shared data
- 2 stage replication by GoldenGate



Agenda

- 1 Overview
- 2 HotCache
- 3 Database Mappings and Transformations
- 4 Performance, Scalability and HA**
- 5 Demonstration

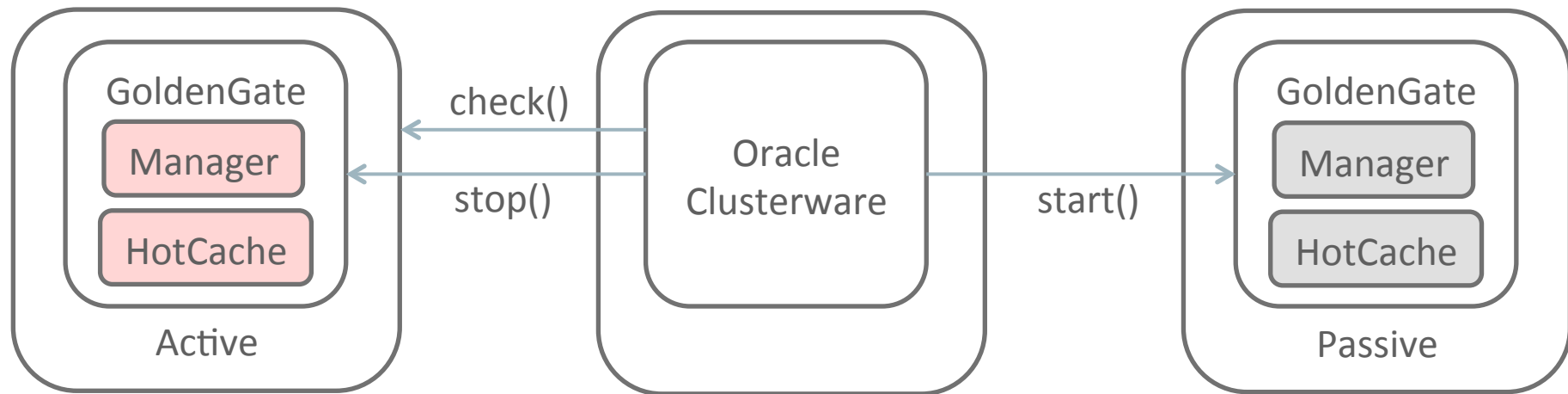
Scaling HotCache via Parallel Data Flows



- DB schema must be amenable (related tables in same trail)
- One HotCache throughput: 700-3000 TPS depending on HW, configuration
- This approach has been tested to 18,000 TPS

HotCache High Availability

- Coherence is already HA
- Oracle Clusterware manages redundant GoldenGate HotCache processes
- <http://www.oracle.com/technetwork/middleware/goldengate/overview/ha-goldengate-whitepaper-128197.pdf>



Monitoring HotCache


Products and Services > Oracle Fusion Middleware > Data Integration > GoldenGate > Management Pack > Overview

Management Pack for Oracle GoldenGate

Streamlined Management and Monitoring

Management Pack for Oracle GoldenGate includes Oracle Enterprise Manager Plug-in, Oracle GoldenGate Monitor, and Oracle GoldenGate Director. It simplifies management and monitoring of Oracle GoldenGate solutions across the enterprise.

 [Data Sheet \(PDF\)](#)



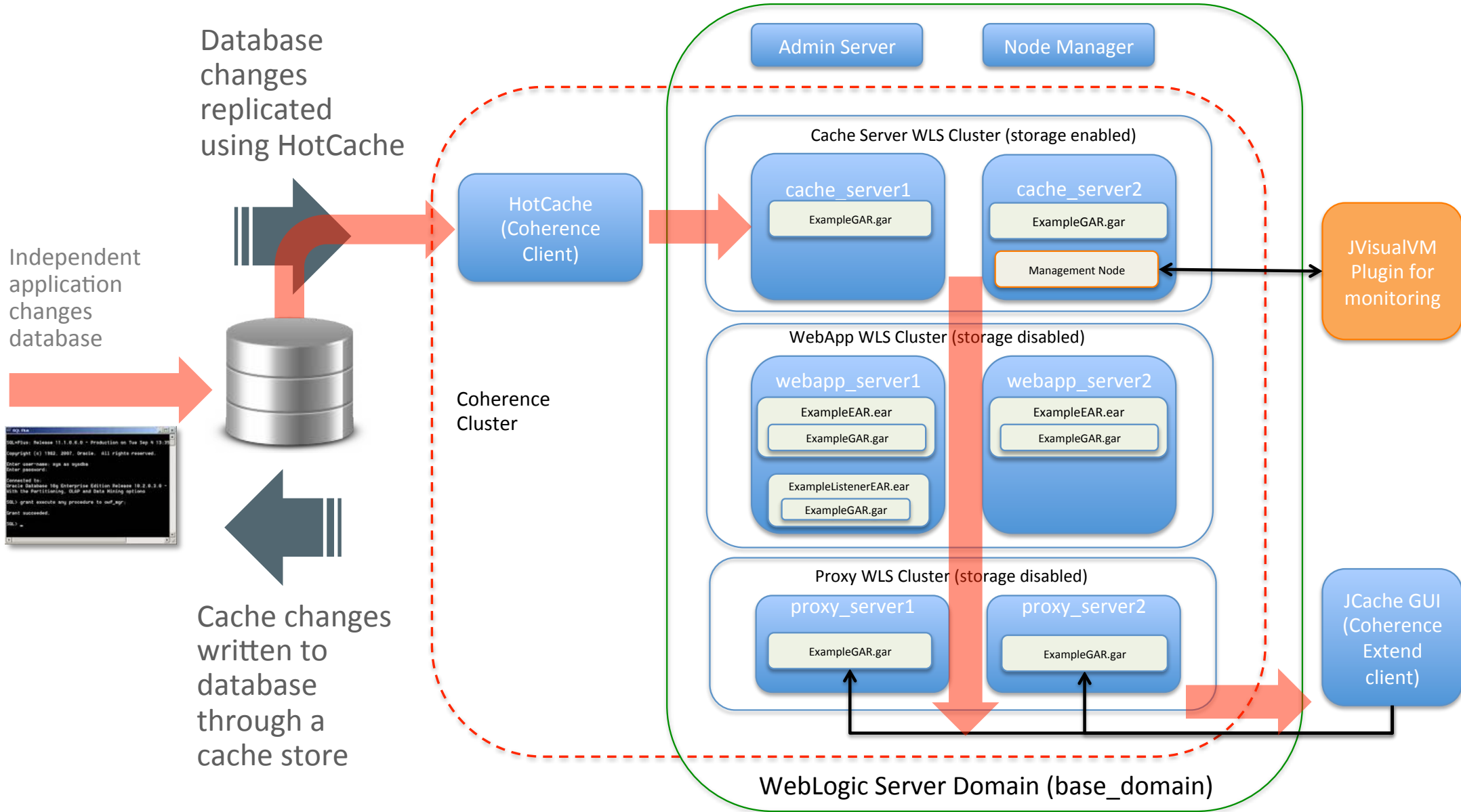
> Download Price & Buy

Agenda

- 1 Overview
- 2 HotCache
- 3 Database Mappings and Transformations
- 4 Performance, Scalability and HA
- 5 Demonstration**

Demonstration

HotCache in Action



ORACLE®