
JavaOne℠
Sun's Worldwide Java Developer Conference

JavaOne℠

Sun's Worldwide Java Developer Conference

Database Access

Rick Cattell
JavaSoft

Database Access

1. Relational database connectivity: JDBC™
2. Higher-level relational APIs and tools
3. Object-oriented databases
4. Partners and summary

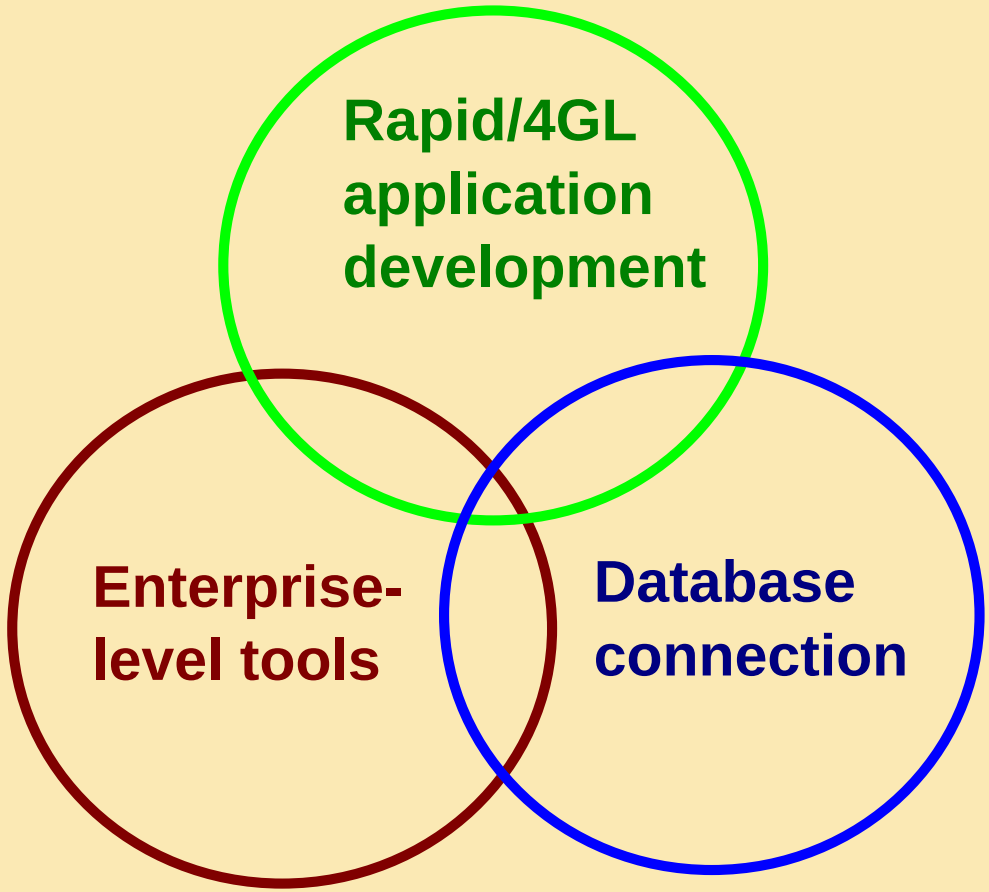
Overview

- Java™ language attractive for database applications
- Increasing interest for general applications, not just Internet
- Establish critical mass, libraries, tools for Java
- Have moved quickly with basic database connectivity
- Not trying to “do it all ourselves”: leverage partners, existing APIs

Why Java™ is Attractive for DB Applications

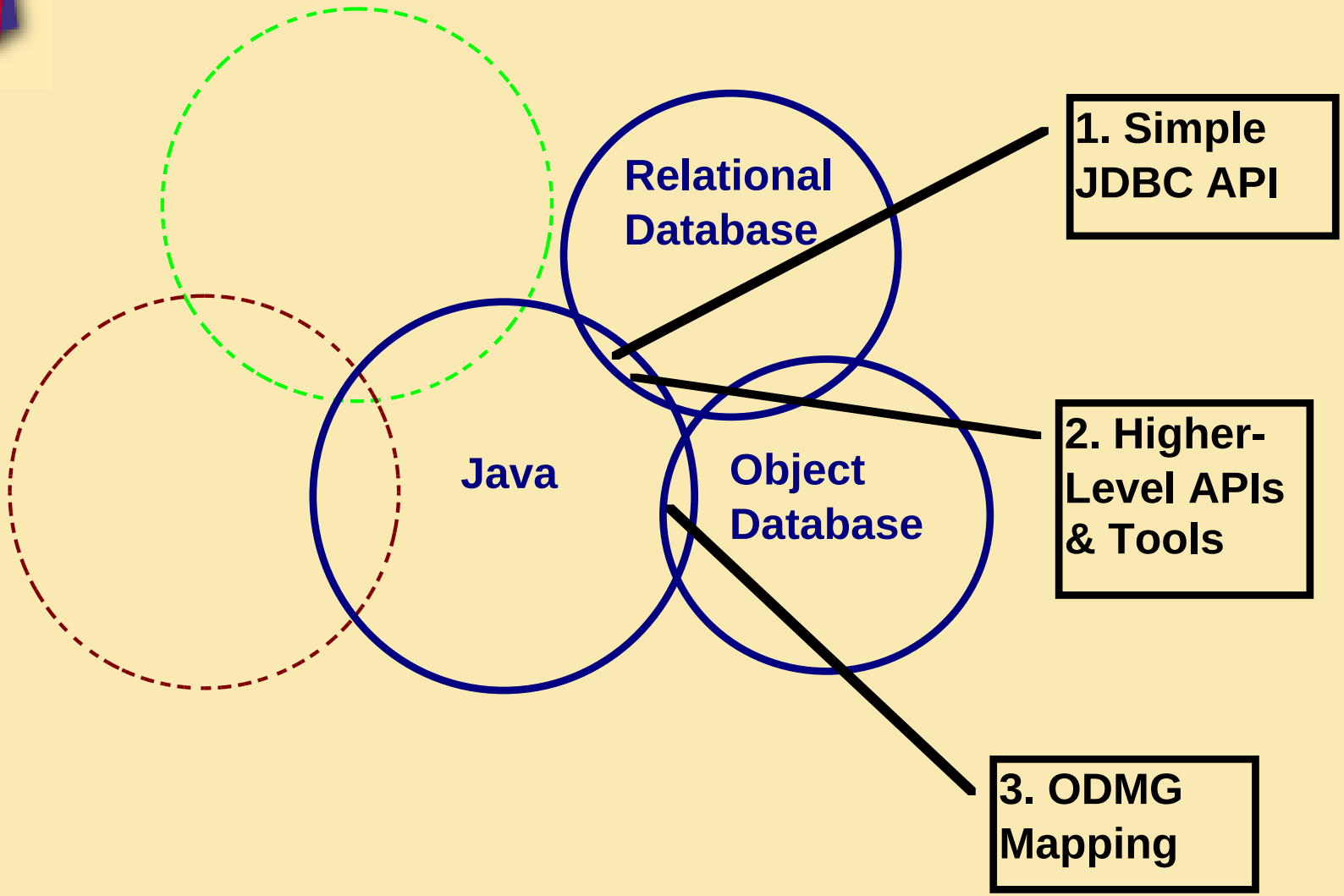
- Zero installation/administration
- Platform-independence
- Secure enterprise applications
- Internet connection
- Better language (than VB, C++, COBOL)
- Increased productivity (no clobbers, memory leaks, void*...)
- However: the Java language alone is not enough

New Components for Enterprise Apps

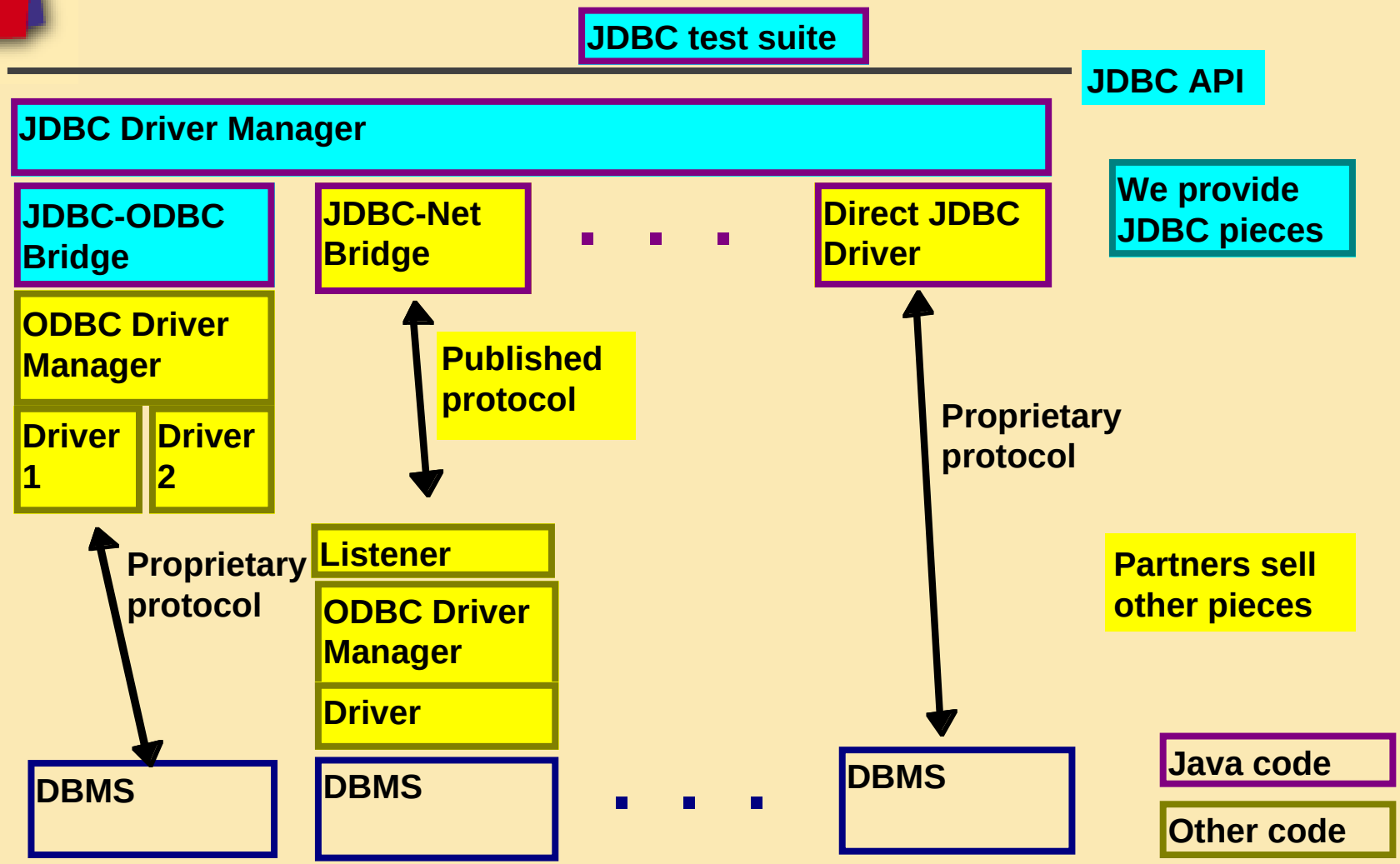


Next two panels cover work in all of these areas

Database Connectivity



1. SQL Interface: JDBC



JDBC Product

- Existing C database APIs not practical for Java
- Need solution yesterday; easy to leverage ODBC
- Synergy for ISVs producing connectivity and tools
- Partnership with Intersolv and others
- See `http://splash.javasoft.com/jdbc` for more info

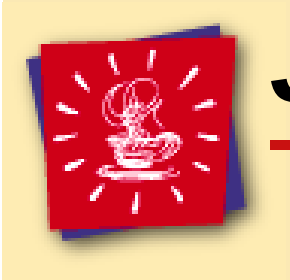
March	June	September
Alpha Spec Alpha DM Endorsees	FCS Spec, DM Drivers available Beta bridge, tests	More drivers available FCS ODBC bridge FCS test suite

JDBC Design

- Generally patterned after ODBC and X/Open CLI
- Faithful to Java language with ease-of-use emphasis
- Low-level API; build other APIs on top
- Two kinds of users:
  - Programmers (need ease of use)
  - Programs (need completeness, performance)
- Two modes of operation:
  - Untrusted applets and drivers on Internet
  - Trusted code accessing company DBMS servers

JDBC Features

- Connection, Statement, ResultSet interfaces
- PreparedStatement and CallableStatement for compilation and stored procedures
- Metadata and dynamic access
- Dynamic driver selection and loading
- Database naming based on URLS; typically `jdbc:<subprotocol>:<identifier>`
- ... More details in Graham Hamilton’s talk



JDBC Example

```
…
  Connection conn =
     DriverManager.getConnection(
     “jdbc:odbc:sales”);
  Statement stmt =
     conn.createStatement();
  ResultSet rs = stmt.executeQuery(
     “SELECT Name, Sales FROM Customers”);
  while (rs.next()) {
     String name = rs.getString(“Name”);
     int sales = rs.getInt(“Sales”);
     …
```

2. Higher-Level Relational Tools and APIs

- Embedded SQL for Java language
- Integration with application-building tools
- Integration with 3-tier tools: RMI, CORBA
- Enterprise repository and administration tools
- Object/relational mapping

Object/Relational Mapping: Declarations

SQL:
```
CREATE TABLE CUSTOMER (
        CUSTID INTEGER NOT NULL,
        ADDRESS VARCHAR(50),
        SALESREP INTEGER,
        PRIMARY KEY (CUSTID),
        FOREIGN KEY (SALESREP)REFERENCES SALES);
```

Java:
```
class Customer {
        int CustID;
        String address;
        Sales salesRep }
```

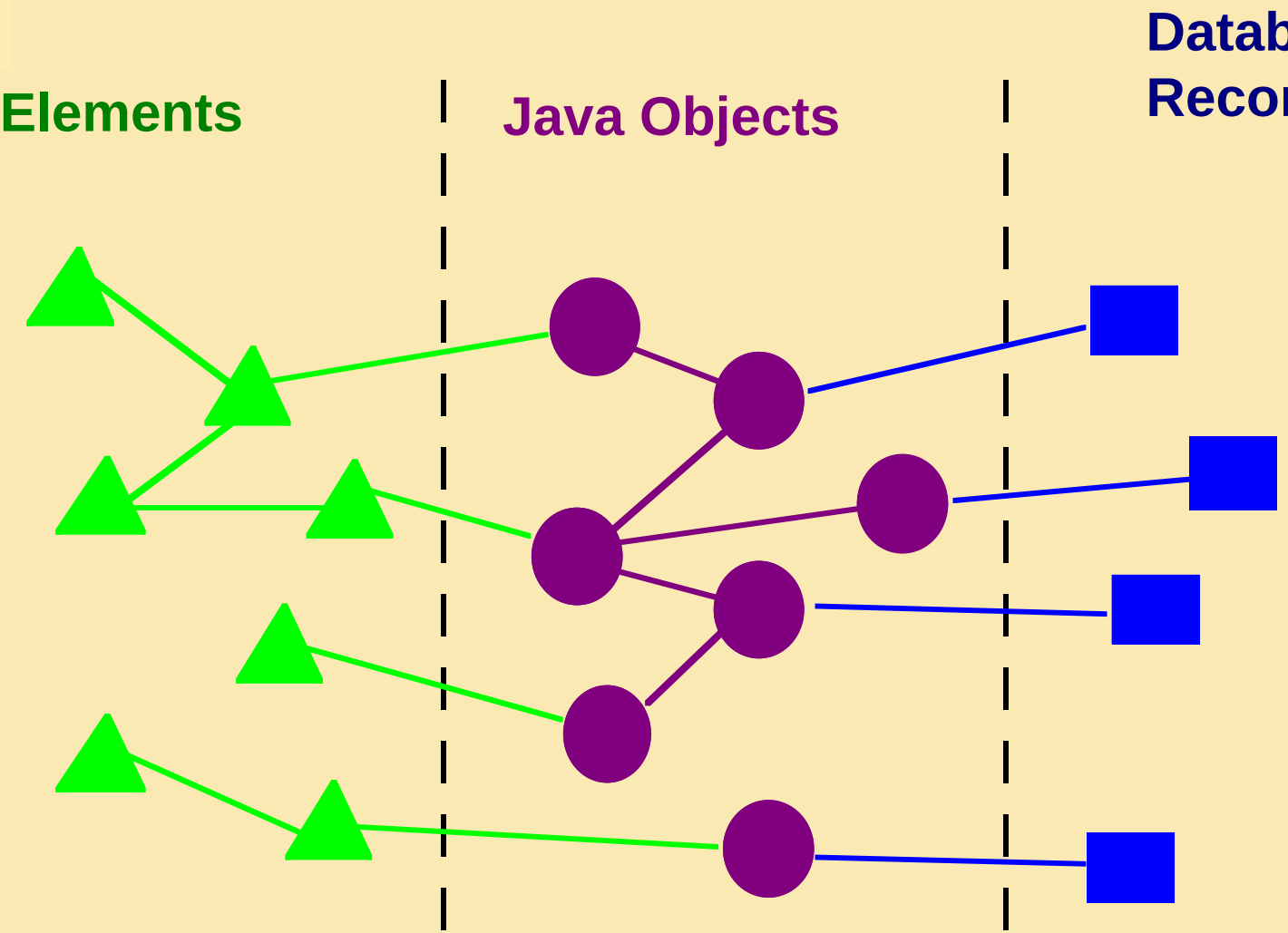
Automatically create Java class for each table in database schema (or vice versa), plus provide tools to embellish for many-to-one mappings, relationships, etc.

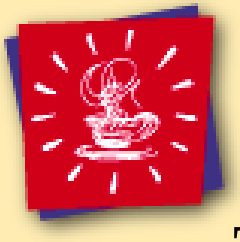
Object/Relational Mapping: Code

```
...
Transaction t = Transaction.create();
CustomerSet cs = CustomerSet.query(mycustomers);
Sales s = c.salesRep;  /* follows foreign key */
c.address = newAddress;  /* obtains write lock */
s.sales = s.sales + thisOrder;
t.commit();  /* writes c and s back to database */
...
```

Java transaction object, database object, etc; transparently fetch/store Java objects from tables, map foreign keys to Java references

Object/Relational Mapping + GUI Tools





3. Object Databases for Java

- Transparent persistence for Java objects
- Java ideal for object databases since “safe”
- Cached performance with more powerful data structures than relational
- More transparent and faster than “pickling”
- Not a JavaSoft product; working with ODMG consortium (Object Database Management Group): voting members GemStone, IBEX, O2, Object Design, Objectivity, POET, UniSQL, Versant

ODMG Timetable

- Formed working group, chaired by me
- Already reached agreement on initial specification
- Several vendors well along on implementation
- See `http://www.odmg.org/` for more ODMG info

April	June	December
ODMG WG Draft spec	Review spec Java hook?	Freeze spec Initial products

ODMG Java Binding

- ODMG class library: Database, transaction, query, and collection classes
- Persistence orthogonal to type: Existing classes can have persistent and transient instances
- Persistence by reachability: All objects reachable from designated named root database objects become persistent on transaction.commit
- ODMG object model: Integrity constraints, can share data with persistent C++ and Smalltalk programs
- Full database functionality: SQL2 superset OQL, transaction per thread, object locking

ODMG Example

```
Database.open(“University”,Database.ReadWrite);
Transaction t = new Transaction;
...
SetOfStudent mathematicians = Students.query
  "exists s in this.takes: s.course.name=\"math\"");
Student joe = Students.select(“id = 4132”);
Professor oldAdvisor = joe.advisor;
joe.advisor = newAdvisor;
joe.address = “123 Main Street”;
t.commit()
...
```

4. Database Partners

Companies	Products this year
Bluestone	Sapphire/Web database app-builder
Borland	InterClient JDBC driver for InterBase, Latté app-builder
BulletProof	JAGG JDBC/ODBC drivers and JDesignerPro database app-builder
IBM	DB2 JDBC driver. CGI scripting, app-building tools, Java stored procedures/user functions / triggers, Data Access Builder and Visual Age object/relational mapping on JDBC

Database Partners

Companies	Product this year
Imaginary	Postgres95 and mSQL JDBC drivers, mSQL DBMS
Informix	Database app-building tools, DBMS
Intersolv	SequeLink JDBC-net driver, JDBC drivers, ODBC drivers, joint development of JDBC/ODBC bridge and test suites
O2 Technology	Object database, object/relational mapping
Object Design	Object database, object/relational mapping, JDBC driver
Open Horizon	Connection JDBC-net driver, security services, directory services, TP services
OpenLink	JDBC drivers
Oracle	Database application building tools, DBMS
POET	Object Database

Database Partners

Companies	Product this year
SAS	Java JDBC driver for Share*Net
SCO	SQL-Retriever JDBC/ODBC drivers
Spider Software	NetDynamics web/database application builder for JDBC and ODBC
Sybase	Optima++ Java database application development tools, DBMS
Symantec	JDBC drivers, Café app bulding tools and libraries
Versant	Object database
Visigenic Software	JDBC drivers, OpenChannel JDBC-net driver, ODBC drivers, source licensee partner
WebLogic	T3Server JDBC-net driver, dbKona API

Database Partners

Companies	Product this year
Working Set	DataRamp JDBC-net driver and server
XDB	JetConnectPro JDBC/ODBC drivers, DB GUI classes on AWT, JetStream RDBMS/gateway

For more partner info see JDBC web page
and JavaOne panels on database access, engines, and tools

Summary

- JDBC: Quick to market, access legacy and relational database, direct use and generated code
- Higher-level: Object-relational mapping, embedded SQL, integration with tools/repository
- Object database: High-performance persistence for small footprint, embedded use
- Partners: Not trying to do it all ourselves; focus on APIs so pieces plug together