



# Enabling Offline Web Applications With Java™ DB

**Francois Orsini**

Java DB  
Sun Microsystems

**Kevin Henrikson**

Zimbra Desktop  
Zimbra, Inc.

<http://developers.sun.com/javadb/http://www.zimbra.com/>

Session TS-69700



# Agenda

The Offline Web

Java™ DB Web Client Store Service

How Zimbra™ Did It

Java DB/Apache Derby Community

Q&A



# The Offline Web



# The Offline Web

- Still live in a non-always connected web
- Most web applications (web apps) cannot run offline

# Enabling Offline Web Applications

- Homogeneous applications for online and offline
- Empowering web applications with Offline and local data access

# Empowering Web Applications

- Offline can also be associated with “local”
  - Local (client) data access from your web app
  - Increase performance with client persistent cache
  - Fully secure (encrypted) local data store
- Web 2.0+
  - No offline paradigm in Web v.2.0
- All about improving the user experience



# Java DB: A Compact and Fully-Featured Data Store



# Java™ DB

- Sun's supported distribution of Apache Derby
  - All development done in the Apache Derby community
- Complete relational database engine
- 100% Java technology
- Bundled in Sun Java Development Kit (JDK™) 6 and Project GlassFish™
- Supported by NetBeans™ software, Sun Java Studio Enterprise software, Eclipse
- ***The database for Java applications***



# Java™ DBn Characteristics

- Complete relational database engine
- Embeddable and client/server database
- Easy-to-use, zero-maintenance
- Small footprint (2MB)
- Standards-based [Java DataBase Connectivity (JDBC™) software, SQL92/99/2003]
- Compact, secure, mature and robust
- 100% Java technology (write once, run anywhere)

# Complete Relational Engine

- Multi-user, transactions, isolation levels, deadlock detection, crash recovery
- Fully ACID-compliant
- Complete SQL Engine including
  - views, triggers, stored procedures, functions
  - Foreign keys, check constraints, cost based optimizer
- Data caching, statement caching, write ahead logging, group commit
- Online backup/restore
- Database encryption

# Embeddable Database

- Database engine may run in application's virtual machine
  - No additional process
  - Database requests are method calls within the Java Virtual Machine (JVM™)
- Startup and shutdown controlled by application
- Just one Java Archive (JAR) file
- Invisible to the user
- Easy-to-use, zero-maintenance
- Can also run as an embeddable database server

# Java Platform Stored Procedures/Functions

```
CREATE FUNCTION SEND_MAIL(  
    TO_ADDRESS VARCHAR(320),  
    SUBJECT VARCHAR(320),  
    BODY VARCHAR(32000)) RETURNS INT  
LANGUAGE JAVA PARAMETER STYLE JAVA NO SQL  
EXTERNAL NAME 'testing.MailTest.sendSMTP_F';  
  
-- Send a Welcome e-mail when new customers are added.  
  
CREATE TRIGGER WELCOME_CUSTOMER  
AFTER INSERT ON CUSTOMER REFERENCING new_table AS newtab  
FOR EACH STATEMENT MODE DB2SQL  
SELECT SEND_MAIL(c.email, 'Welcome to AcmeWidgets', M.email_text)  
FROM newtab C, MAILINGS M  
WHERE C.TYPE = M.CUST_TYPE AND M.OFFER_TYPE = 'welcome'
```

# Java Platform SQL Function

```
public static int sendSMTP_F
    (String toAddress, String subject, String content)
{
    recipient = new InternetAddress(toAddress);
    ...

    msg = new MimeMessage(session);
    msg.setFrom(from);
    msg.setSubject(subject);
    msg.setText(content);
    msg.addRecipient(Message.RecipientType.TO, recipient);
    javax.mail.Transport.send(msg);
    return 0;
}
```

**See** <http://wiki.apache.org/db-derby/SendEmailRoutine>

# Sample Deployments

- Local client store for Web v.2.0 applications
  - Local AJAX (demo follows)
- Embeddable middle (web) tier cache
- Embeddable in rich client applications
- Read-only DB in JAR file
- Java DB on a memory stick
- Departmental client/server database



# Java DB Web Client Store Service



# LAJAX as Local AJAX

- Access to “local” client WebApp services
  - Accessible from AJAX
  - Directly from JavaScript™ technology
- A model for disconnected web apps
- Enable richer web client applications
  - e.g., offline capability
- LAJAX white paper under way



# Local Client Web Services

- “Local” WebApp client services
  - Accessible from a web application
  - Services run locally on the client host
- Javascript technology interface
- Local AJAX services
  - Asynchronous access to local services
  - Local AJAX controller
- Services installed as plug-ins or extensions

# Local Client Web Services Examples

- Local web app data caching
  - Web app data hosted/cached locally on the client
  - Synchronization with the online realm as required
    - Can be handled by the application
    - 3<sup>rd</sup> party synchronization solutions
  - Decrease of Web server access round-trips
  - Local serving of web pages
    - Lightweight web server running as a service in the browser
  - Offline web app alternative
  - Secure storage with encryption

# Local Client Web Services—How to?

- JavaScript technology interface
  - Ubiquitous
- Service logic written in Java code
  - “J” for Java technology, not just JavaScript technology
  - Richer client applications
  - Rebirth of the Applet
- (Sun) Java Plug-in software
  - Browser agnostic
  - Service can be installed as a Java Plug-in software extension
  - LiveConnect capability
  - Easy deployment/install

# Local Client Web Services (Continued)

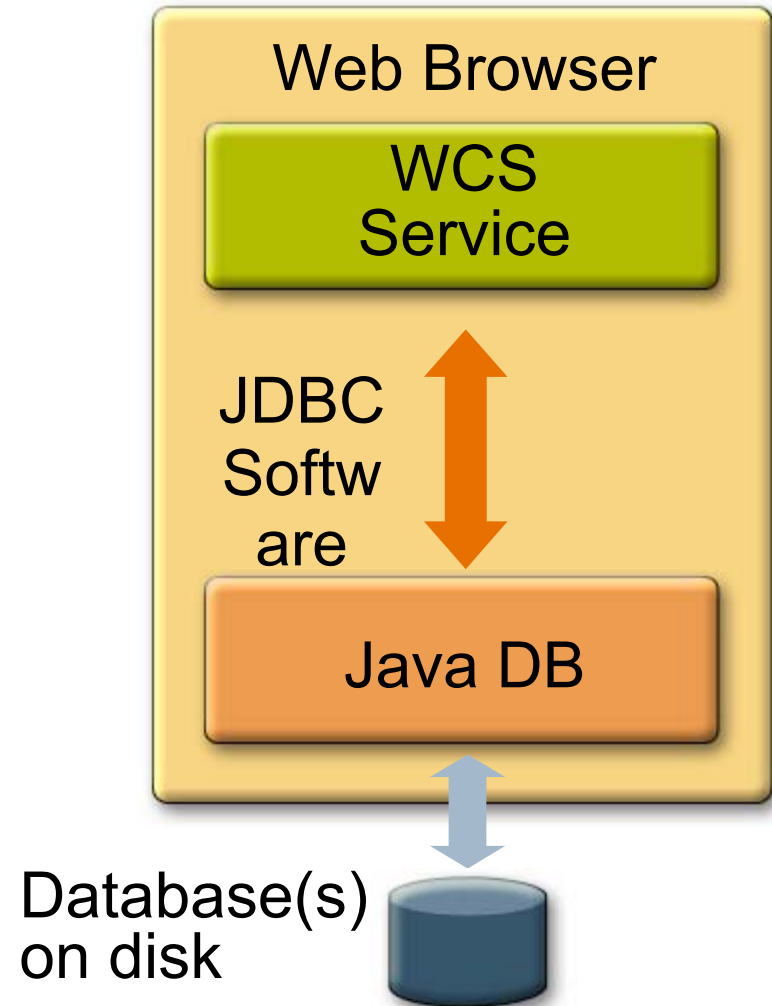
- JavaScript technology interface
  - LiveConnect to interact with core service implementation
- HTTP interface
  - Accessible from AJAX
  - Local Web Service (embedded web server)
- XMLHttpRequest prototype override
  - open() and send()
  - AJAX request(s) redirection to local web service(s)
  - Online/Offline state handling in the overridden methods

# Local Client Service via JavaScript Technology

- Interact with local service directly via JavaScript technology
  - No new syntax or else—all JavaScript technology
- Local service installed as a Java Plug-in software extension
  - Trusted, runs in Java platform Sandbox
  - Automatically installed on client host
  - Service versioning management handling
- [http://java.sun.com/j2se/1.4.2/docs/guide/plugin/developer\\_guide/extensions.html](http://java.sun.com/j2se/1.4.2/docs/guide/plugin/developer_guide/extensions.html)
- LiveConnect to interact transparently with core service implementation in Java technology
  - JavaScript technology to Java technology and vice-versa
  - Browser agnostic

# Web Client Store Service

- WCS service access via JavaScript technology
- Embedded Java DB
- ACID compliant
- Fast
- Zero administration



# Service Access via HTTP

- Embedded Web server running as a local service
  - Installed as a Java Plug-in software extension (e.g., Jetty) or running on the desktop (e.g., Service/Daemon)
  - Can serve XMLHttpRequest's
- Local AJAX controller can do online/offline redirection based on some state
- Helper classes to provide specific functionality
  - XML encoding
  - Encryption
  - HTTP redirection
  - etc.

# XMLHttpRequest Override

- JavaScript programming language is a prototype-based language
- JavaScript programming language override of `XMLHttpRequest.prototype.open` and `XMLHttpRequest.prototype.send`
  - `XMLHttpRequest`'s submitted the same way
- Local AJAX controller can do online/offline redirection based on some state



# Data Synchronization

- Not always required
  - Depends on the application
- Conflicts resolution is the biggest problem
- At the application level
  - Zimbra desktop
  - Offline Derby Google Calendar (demo part of Derby)
- Database level
  - Daffodil Replicator with Derby  
<http://sourceforge.net/projects/daffodilreplica/>



# DEMO

## Offline Web App



# Web App With Embedded Java DB

- A web application accessing Java DB from a local client store service
- Java DB runs as a local secure store—it is exposed to the web application through a JavaScript technology interface (LiveConnect) or a local HTTP service (AJAX)
- Allows local caching of data on the client side
  - No network connectivity to interact with the data store
  - Store sensitive, private, confidential data on the client's host, instead of a remote server (or both)
  - Java DB can encrypt the local database
  - Local store for loosely connected web apps

# Demo—Things to Remember

- Ease-of-deployment over a large user base (e.g., consumer desktops)
- Transparent-embeddable and zero-administration
  - invisible to the end user
- ACID RDBMS—high levels of durability and consistency to prevent data loss
- Ease-of-upgrade (using Firefox or Java Web Start software)
- Small footprint
- Highly secure to ensure desktop data is safe

# Demo—More Information

- Demo code publicly available at <http://developers.sun.com/prodtech/javadb/>
- For more information see <http://blogs.sun.com/roller/page/FrancoisOrsini>



# How Zimbra Did It



# Zimbra™ Offline

- Features
  - Linux, Win32 and Mac
  - Firefox, Internet Explorer and Safari
  - Identical AJAX interface to online version
  - Micro Server for sync, persistence, and search
    - Derby (Meta Data, User Profile)
    - Lucene (Full text search)
    - Jetty (HTTP, JavaServer Pages™)
    - Filesystem for messages

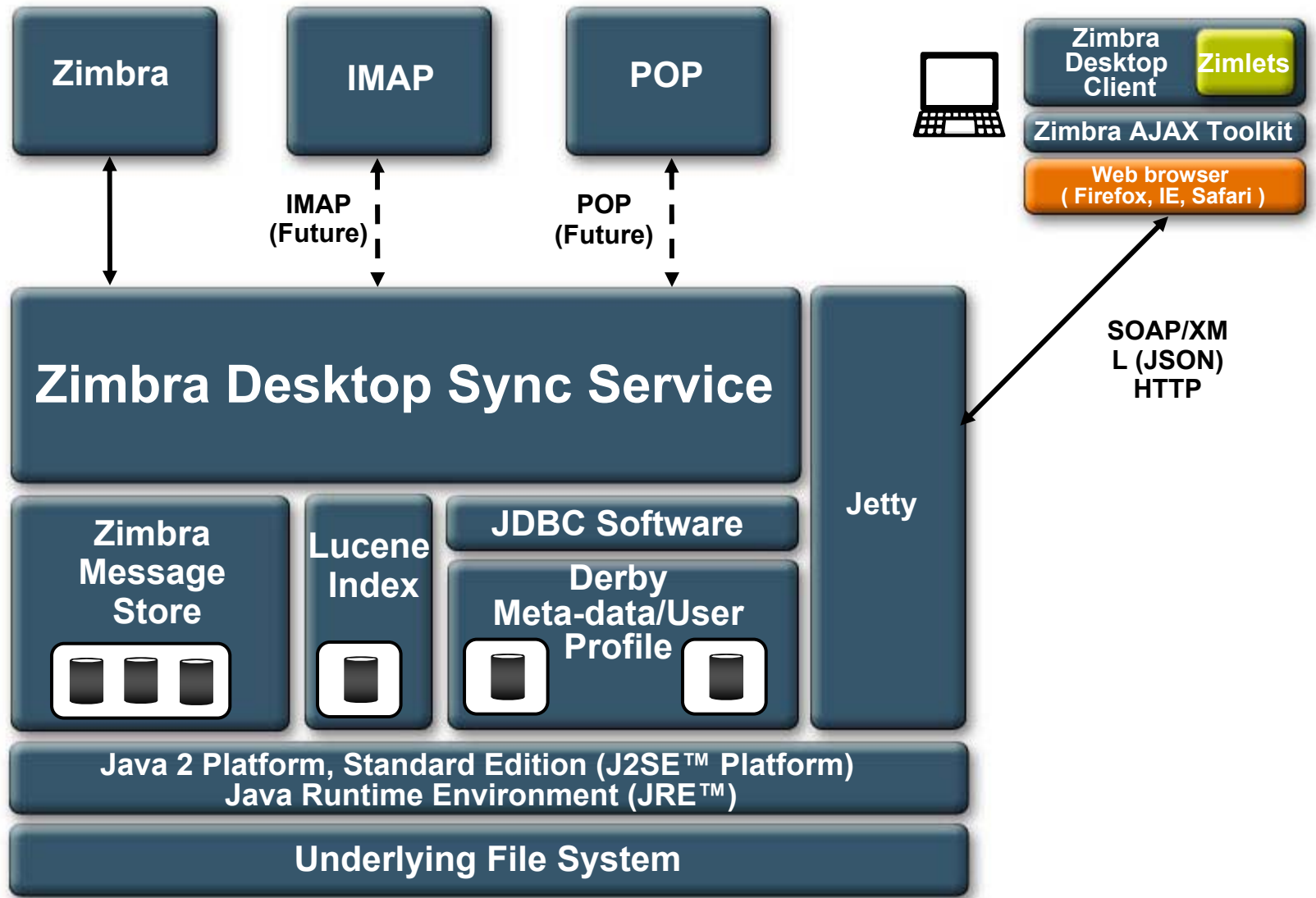
# Zimbra Offline (Continued)

- Support for large datasets multi-GB
- License
  - MPL/ZPL—Open Source
- Availability
  - Zimbra Desktop Alpha

<http://www.zimbra.com/desktop>



# Zimbra Micro Server Architecture



# Offline Alternatives

- Apollo (Adobe)
- Dojo Offline Toolkit (Dojo/Sitepen)
- Slingshot (Joyent/Magnetk)
- Firefox 3 Offline Support (Mozilla)
- XUL Runner (Mozilla)
- Roll your own
  - Salesforce.com
  - TiddlyWiki

# Challenges for the Developer

- Selecting what to take offline
  - User offline needs
  - Security risks
- Sync needs to be thought thoroughly
  - Application level most of the time
- Application support
  - Upgrade path
  - Possible increase in maintenance

# Zimbra Desktop—More Information

- Zimbra Open Source Community
  - <http://www.zimbra.com/forums>
- Zimbra Desktop
  - <http://www.zimbra.com/desktop>
- Zimbra Blog
  - <http://www.zimbra.com/blog>



# Java DB Community



# Apache Derby Community

- Apache Software License v.2.0
- Anyone can contribute
- Active contributors become committers through community vote
- Apache Derby community is
  - growing at fast pace
  - a very active one
  - a great place to learn more about database internals

# Next Release Features (10.3)

- Security improvements
  - DBA Powers
  - Secure Server by Default
  - SSL/TLS between client and server
- Language-based ordering
- Alter table enhancements (drop/rename column)
- More performance improvements
- BLOB/CLOB enhancements
- More info at
  - <http://wiki.apache.org/db-derby/DerbyTenThreeRelease>

# Participate!

- <http://db.apache.org/derby>
  - Download, read docs
- JIRA  
<http://issues.apache.org/jira/browse/DERBY>
  - Report bugs, submit patches
- [derby-user@apache.org](mailto:derby-user@apache.org)
  - Discuss experience, get help, give feedback
- [derby-dev@apache.org](mailto:derby-dev@apache.org)
  - Discuss developer issues



# Derby Integration

- Apache ActiveMQ
- Apache Roller
- Apache Cocoon
- Apache Geronimo
- Apache JDO
- Apache Xalan
- Daffodil Replicator
- Data Direct SequeLink
- DB Visual Architect
- Drone IRC Bot
- Eclipse
- Project Glassfish
- Hibernate
- IBM DB2 Everyplace
- IBM DB2 JDBC Universal Driver
- IBM WebSphere App Server
- ISQL-Viewer
- Java DB
- JBoss
- JPOX
- Jython
- Kodo 3.3.3
- Maven
- NetBeans Software
- Zimbra
- Red Hat Application Server
- AntHill Pro
- Sequoia (C-JDBC)
- SQuirreL SQL
- Sun Java Enterprise System
- Sun Java System Portal Server
- Sun Java Studio software
- Sun Java Platform, Enterprise Edition
- Sun Java System Service Registry
- SUSE Linux 9.3
- Zend core for IBM
- Tomcat



# Q&A





# Enabling Offline Web Applications With Java™ DB

**Francois Orsini**

Java DB  
Sun Microsystems  
<http://developers.sun.com/javadb/>

**Kevin Henrikson**

Zimbra Desktop  
Zimbra, Inc.  
<http://www.zimbra.com/>

Session TS-69700