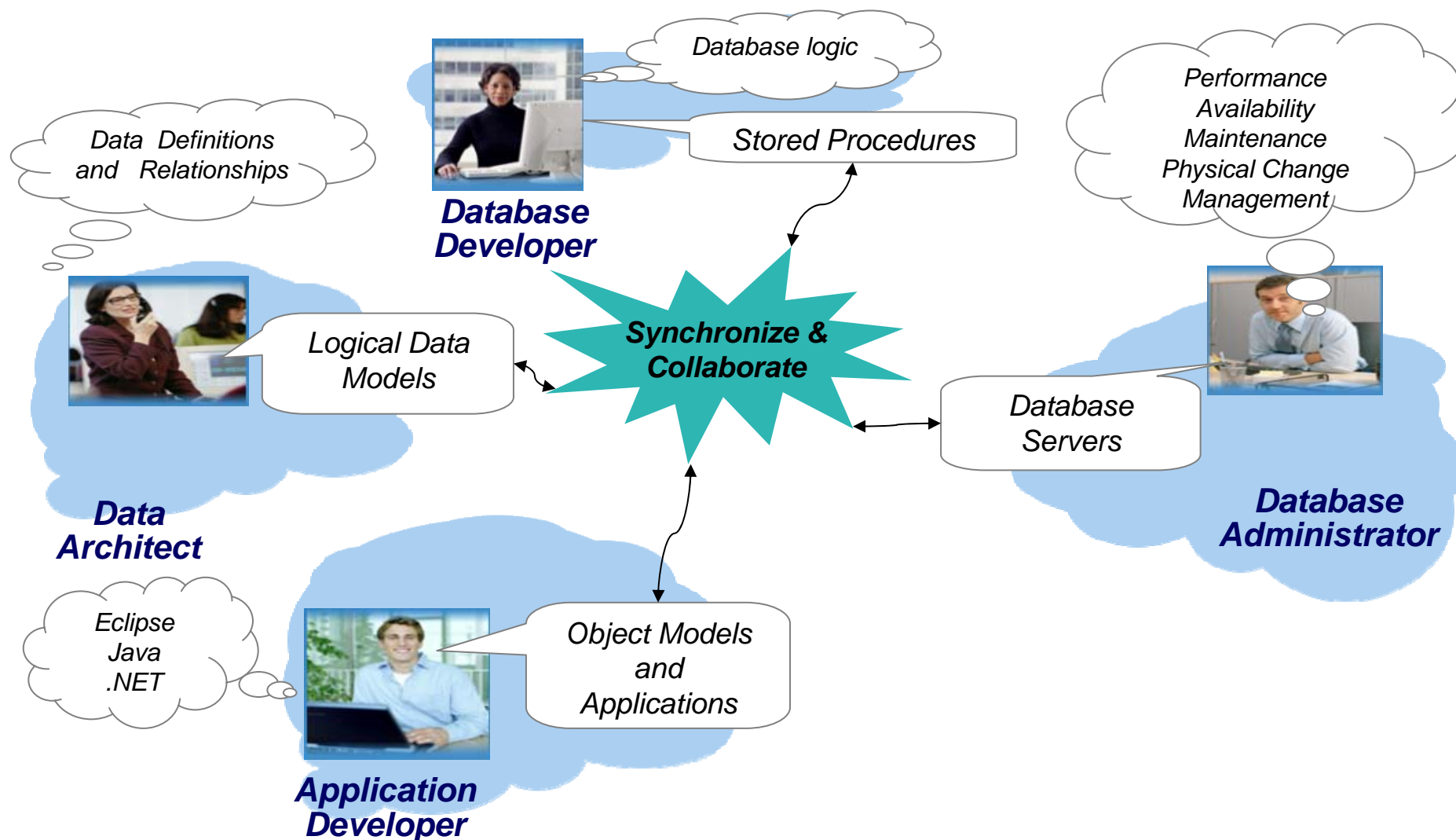# Modern Application Development with DB2

The future runs on System z

Curt Cotner
IBM Fellow,Vice President and
CTO for IBM Database Servers

Data Lifecycle Management

Database logic

Stored Procedures

Performance
Availability
Maintenance
Physical Change
Management

Data Definitions
and Relationships

**Database
Developer**

Synchronize &
Collaborate

Logical Data
Models

Database
Servers

**Data
Architect**

**Database
Administrator**

Eclipse
Java
.NET

Object Models
and
Applications

**Application
Developer**

IBM Data Studio
A Foundation for Innovation

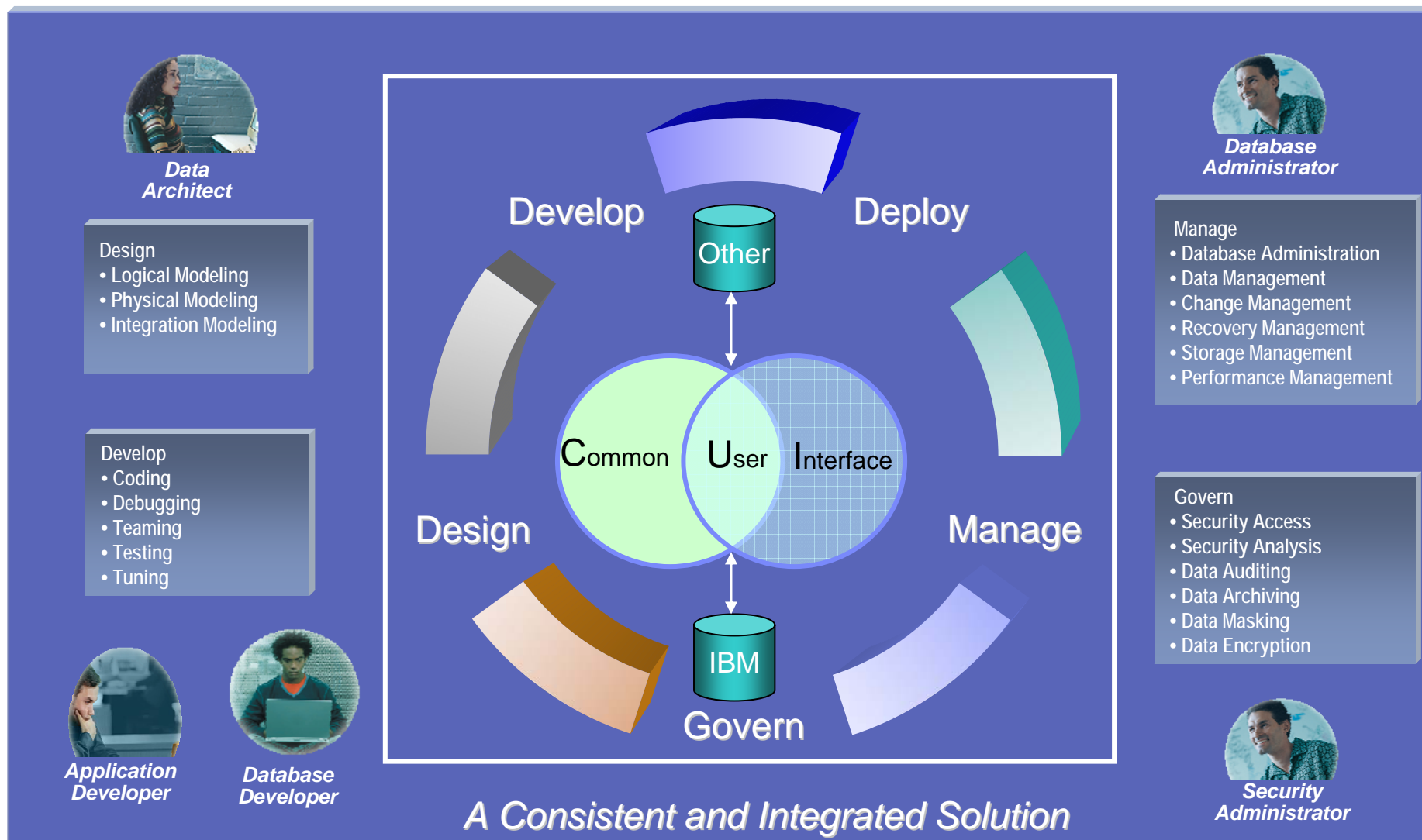| |
|---|
| **IBM Data Studio is a comprehensive data management solution** |

It empowers you to effectively

Design, develop, deploy and manage
*your data, databases and database applications throughout the data
management lifecycle utilizing a consistent and integrated user interface*

IBM Data Studio

Data Architect

Design
- Logical Modeling
- Physical Modeling
- Integration Modeling

Develop
- Coding
- Debugging
- Teaming
- Testing
- Tuning

Application Developer

Database Developer

Database Administrator

Manage
- Database Administration
- Data Management
- Change Management
- Recovery Management
- Storage Management
- Performance Management

Govern
- Security Access
- Security Analysis
- Data Auditing
- Data Archiving
- Data Masking
- Data Encryption

Security Administrator

Develop

Deploy

Other

Common User Interface

Design

Manage

IBM

Govern

A Consistent and Integrated Solution

IBM Data Studio
A Consistent and Productive work environment
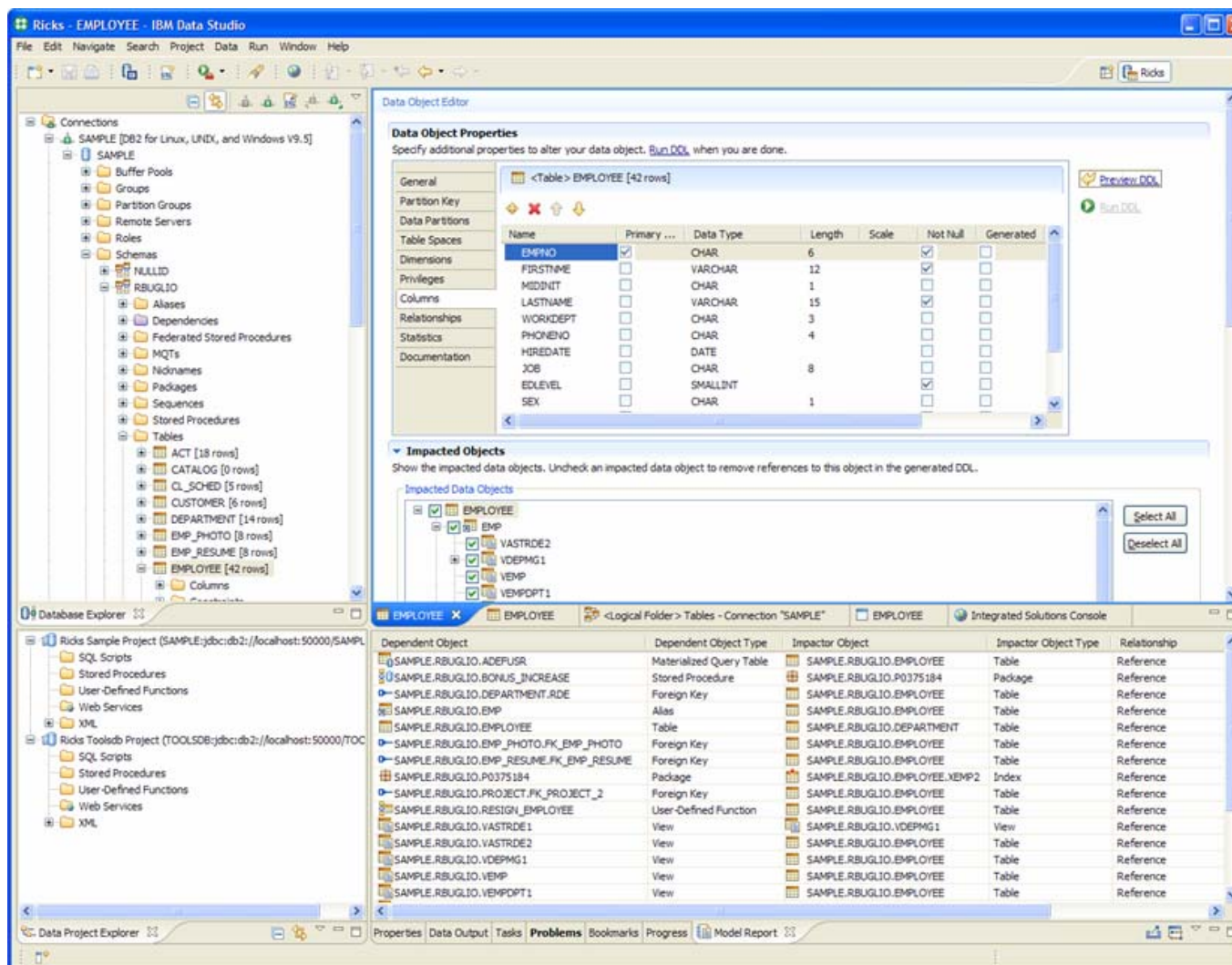
IBM Data Studio
ER Diagramming

ER Diagram Viewer
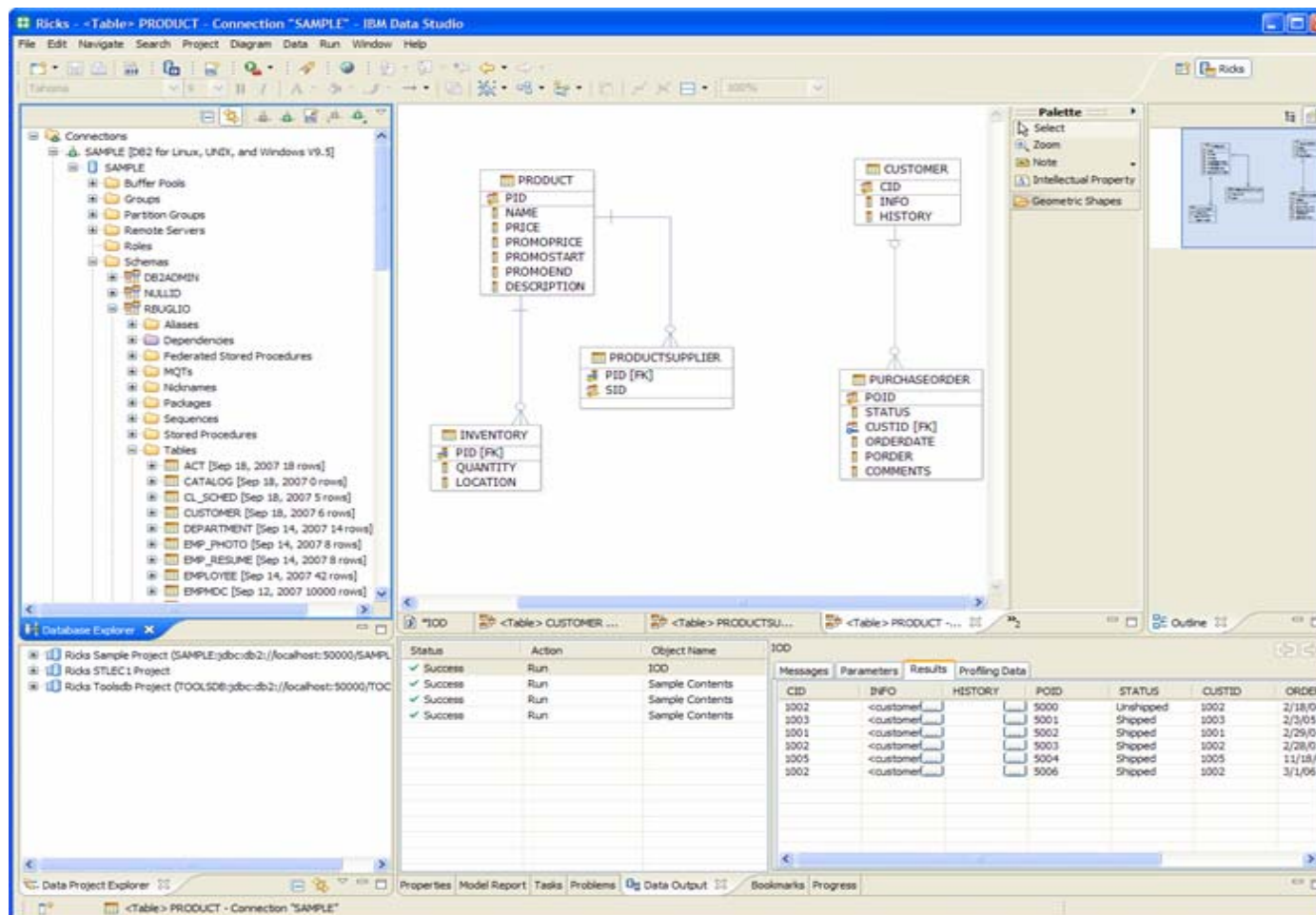
Easily create, view and share ER diagrams

IBM Data Studio
Query Management

Integrated Query Editor

- **Express yourself with optimal queries**

  - Content assistance for database objects

  - Rapid interactive end-user feedback

  - Extensible templates

  - Multiple SQL statement testing

  - SQL assistance and XQuery assistance

SQL and XQuery together
using an integrated editor

IBM Data Studio
XML Management

XML Editors
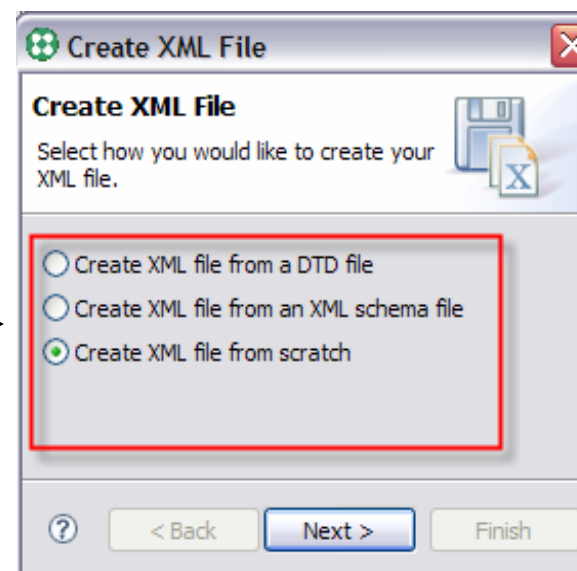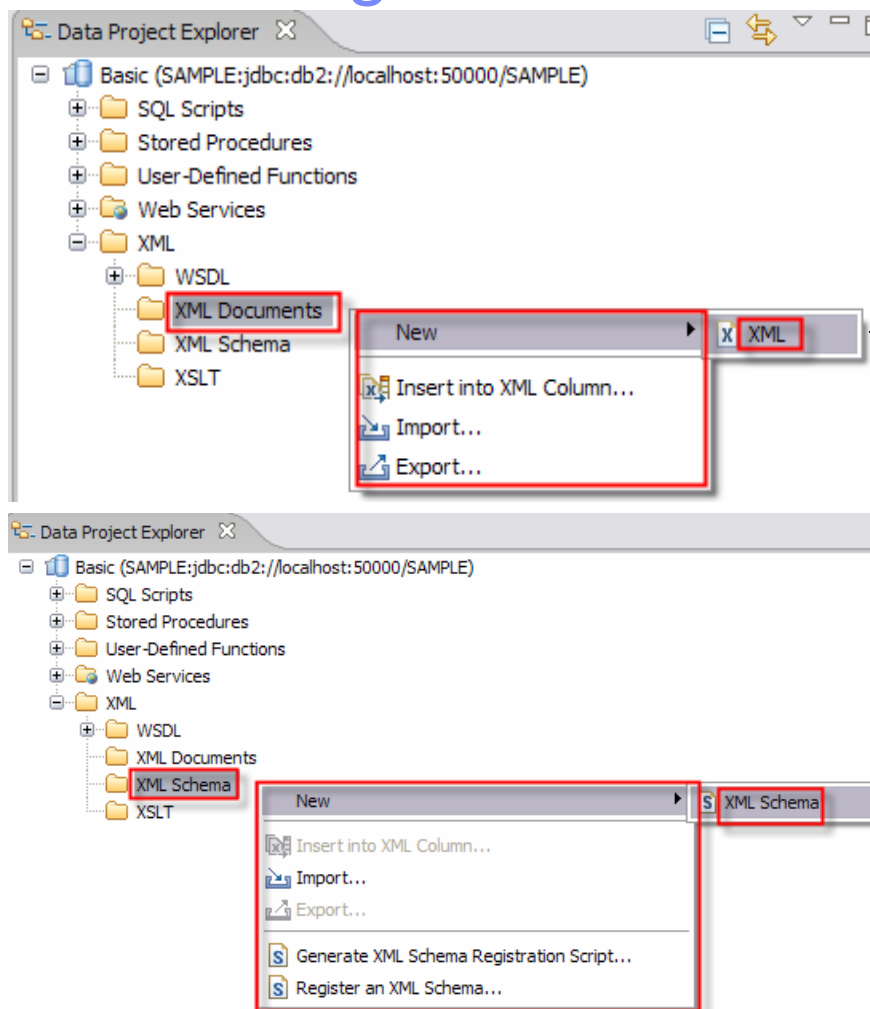
- **Easily perform all your XML needs**

  – Create XML documents and schemas

  – Import and export XML documents and schemas

  – Insert XML documents into XML columns

  – Generate XML Schema registration scripts

  – Register XML Schema

Extensive XML support and capabilities

DB2 Application Development
Programming paradigms

Java

- The IBM Data Server Driver for JDBC and SQLJ is available to support the most demanding of Java applications.

  - Driver is compliant with the JDBC 3.1 and JDBC 4.0 specifications
  - Supports type 4 and type 2 modes
  - Can be used for either client or server-side Java DB2 applications

.NET

- The IBM Data Server Driver for ODBC, CLI, and .NET is available as a small footprint runtime for deploying .NET applications.

- IBM Database add-ins for Visual Studio
  - Extends Visual Studio 2005 and 2008 IDE for DB2 application development

PHP, PERL, Python, Ruby

- **PHP**
  - Native DB2 (ibm_db2) and PDO drivers are available and maintained by IBM
    - binaries included in IBM Data Server clients
    - driver source code is available on PECL
  - Zend Core for IBM is an integrated stack
- **Perl**
  - IBM develops and maintains the DBD::DB2 driver
    - supports DBI standard and available on CPAN
- **Python**
  - IBM develops and maintains the Python DB API 2.0 driver
- **Ruby on Rails**
  - IBM develops and maintains the Ruby driver and Rails adapter
    - Binaries included in IBM Data Server client

Interaction between DBA and App Developer
How does Data Studio and pureQuery Help?

- Manage capacity
- Manage performance
- Continuous monitoring

- Write fast code
- Easy to manage
- Focus on business layer and not data

- Develop Stored Procedures
- Deploy SQL from dynamic to static
- Use of explain plan

❖ Use same Data Studio tooling

❖ Increase productivity and better understanding of application and database

❖ Use same pureQuery tooling

❖ Increase productivity and better coordination

❖ Easy problem management

DBA

Application Developer

Application DBA

pureQuery

IBM Data Studio
Improved Java data access

pureQuery is a high-performance Java data access platform focused on simplifying the tasks of developing and managing applications that access data

Optimizing Java database development and deployment

– Productive set of integrated tools, APIs, and runtime for Java applications

– Unified programming interface to query databases and Java objects

– Unleash the power of SQL within Java applications

– Provides infrastructure to greatly improve monitoring and problem determination for Java applications that access DB2

Developing with pureQuery
Flexible programming styles

Tools
API
Runtime

pureQuery supports several programming styles

1. **Inline style – SQL in application**

   - Simplified direct data access via SQL

2. **Method Style – Encapsulate SQL in Java interfaces**

   - Annotated Method Style
     – Define SQL as Java annotations
   - Named query style – extension of Annotated Method Style
     – Define SQL in XML files

pureQuery - "Inline Style"

- **SQL statements coded directly in application**

- **SQL can be fully declared and embedded, or constructed at execution time**

- **All standard SQL supported including queries, updates, DDL, CALL statements**

```
public Customer {
    @Id public Integer custId;
    public String name;
    public String addressL1;
    public String city;
    @Column(name="REGION")
    public Integer storeRegion;
...
}
```

```java
import com.acme.Customer;

Connection con = DriverManager.getConnection(…);

Data db = DataFactory.getData(con);

Customer c;

int region = 123;

Iterator<Customer> customers =

        db.queryIterator("SELECT custId, name FROM Customer WHERE region=?1",
        Customer.class, region);

while (customers.hasNext()){

  c = customers.next();

  System.out.println(c.custId+" "+c.name);

}

((ResultIterator) customers).close();  // best practice
```

Java Persistence Technologies with pureQuery

```
┌──────────┐   ┌──────────┐   ┌──────────────┐   ┌──────────┐   ┌──────────┐
│  JDBC    │   │ JPA API  │   │ pureQuery API│   │  iBatis  │   │  Spring  │
│  SQLJ    │   └────┬─────┘   └──────┬───────┘   └────┬─────┘   └────┬─────┘
└────┬─────┘        ↓                │                │             │
     │     ┌────────────────┐        │                │             │
     │     │  JPA Runtime   │        │                │             │
     │     └───────┬────────┘        │                │             │
     │             ↓                 ↓                ↓             ↓
     │     ┌─────────────────────────────────────────────────────────────┐
     │     │                   pureQuery Runtime                          │
     │     └──┬──────────────┬──────────────┬──────────────┬────────┬────┘
     │        ↓              ↓              ↓              ↓        ↓
     │        ┌────────────────────────────────────────────────────────┐
     │        │                    High Speed API                       │
┌────────────────────────────────────────────────────────────────────────┐
│                          JDBC w/pureQuery                                │
└────────────────────────────────┬───────────────────────────────────────┘
                                  ↓
```

IBM
Database

pureQuery Metadata,
Manageability
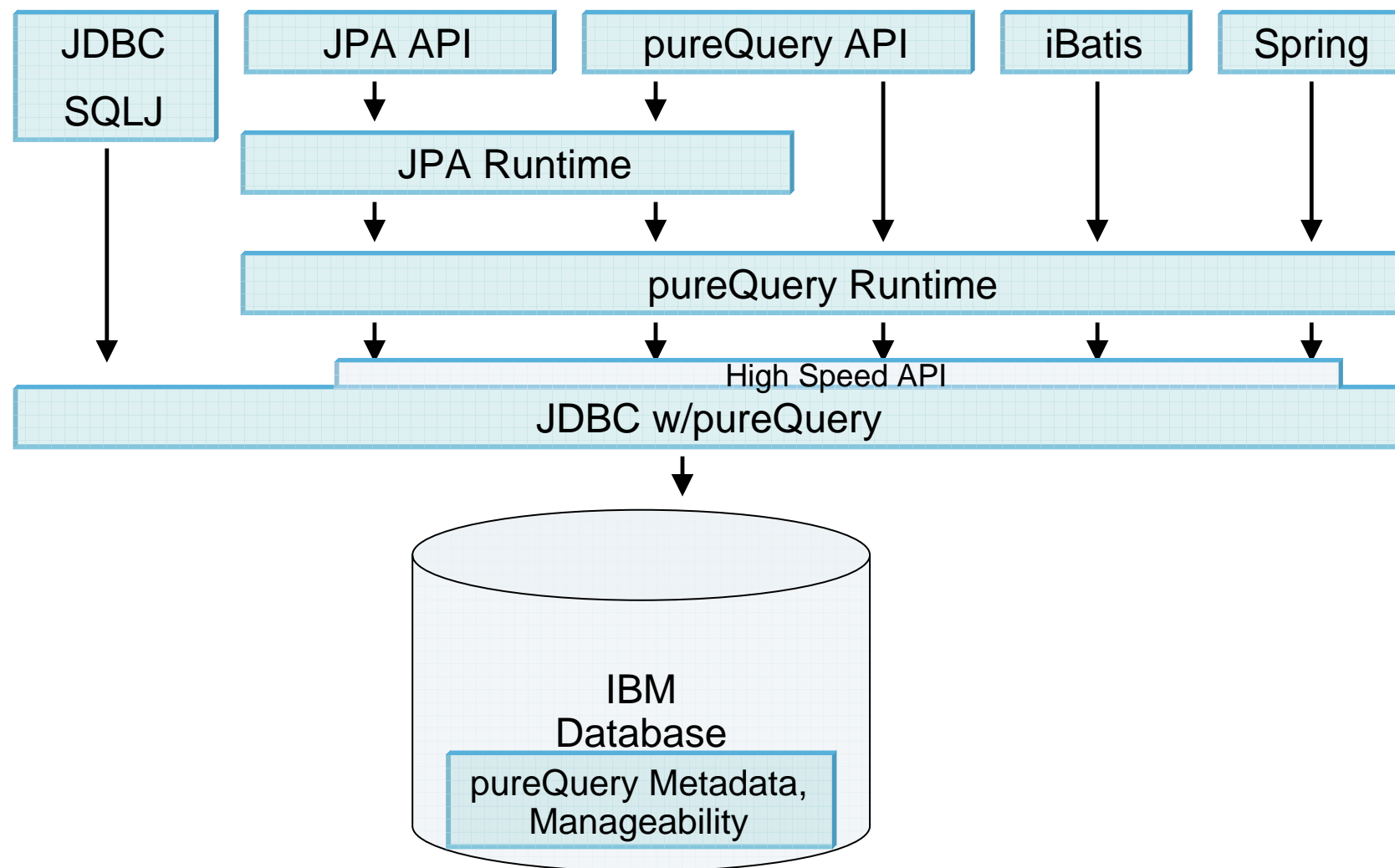
Toughest issue for Web applications – Problem diagnosis and resolution

Web Browser Users

Web Server

Application Server

DB2 Server

Business Logic

EJB Query Language

Data Access Logic

Persistence Layer

DB2 Java Driver

JDBC Package
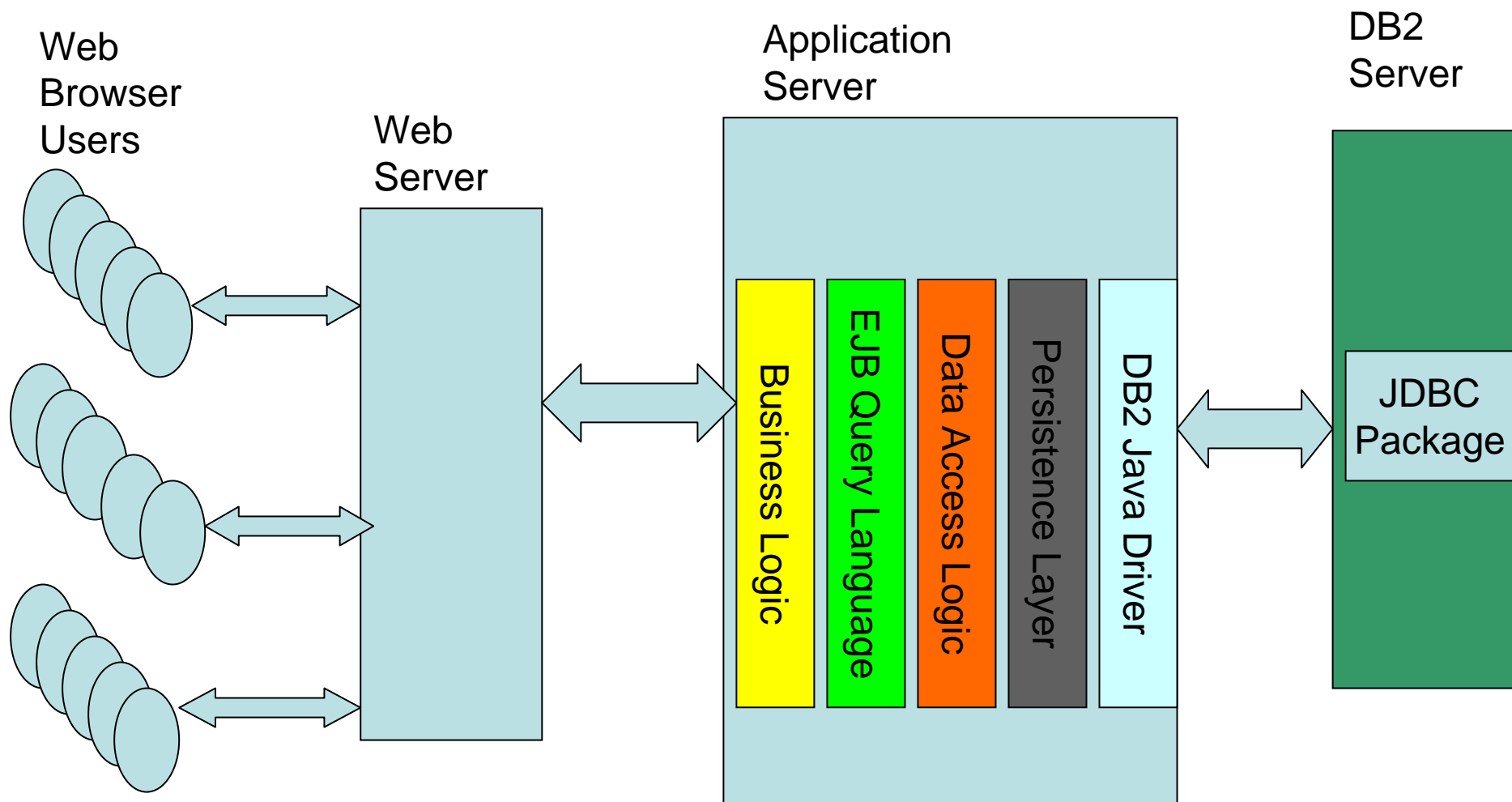
Customer Job Roles – A Barrier to a "Holistic View"

Application Developer

Application Server

System Programmer

DB Server

DBA

Business Logic

1

3

2

4

5

WebSphere Connection Pool

EJB Query Language

Data Access Logic

Persistence Layer

DB Java Driver

JDBC Package
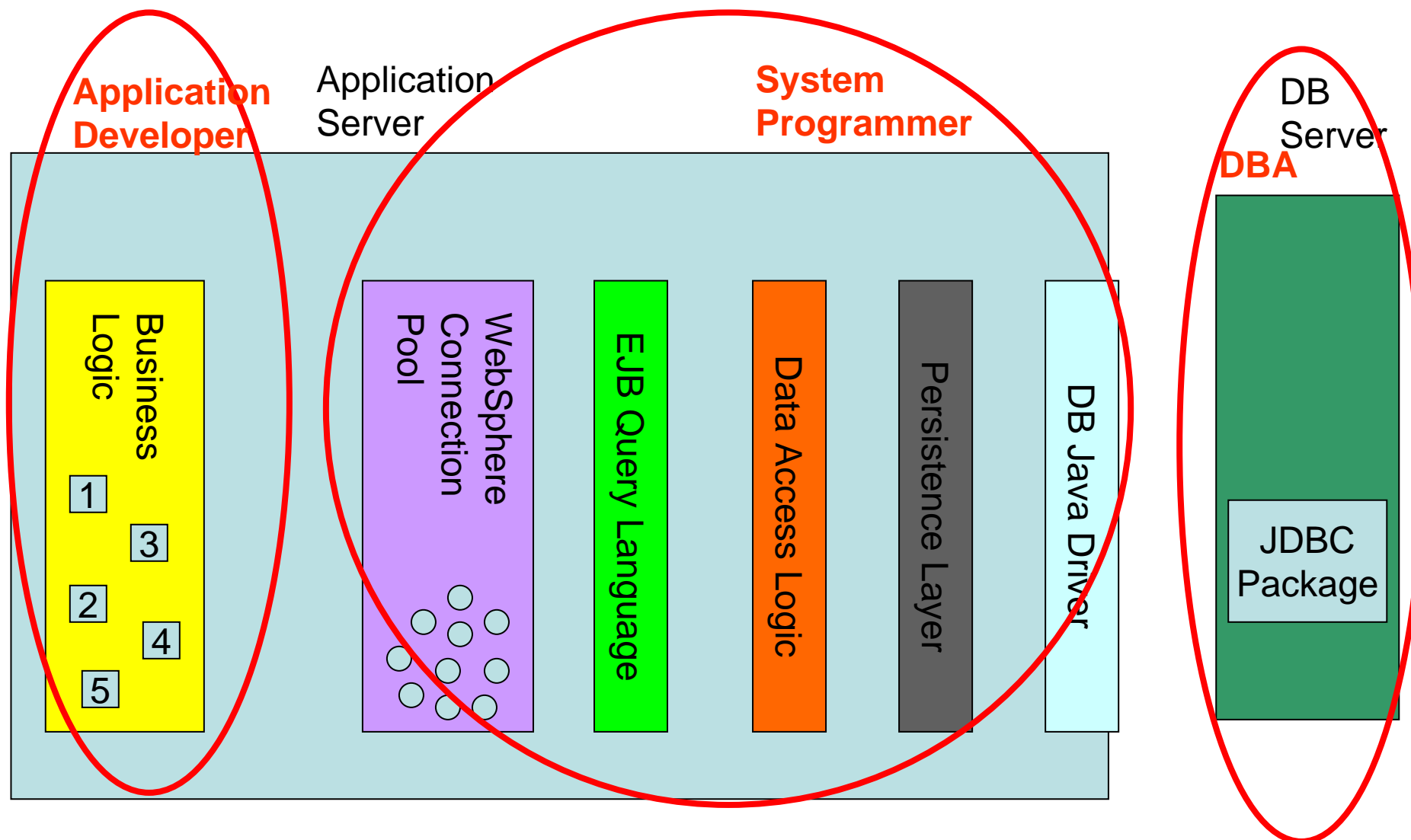
Typical Customer Pain Points

•DBA has no control or visibility into how the WebSphere connection pool is configured

•DBA can't see which app issued a given SQL query

DB Server

•App developer and DBA can't track SQL to JPA Query Language transitions

Business Logic

Pool

JPA Query Language

Data Access Logic

Persistence Layer

DB Java Driver

JPAQL

SQL

JDBC Connections

1
3
2
5
4
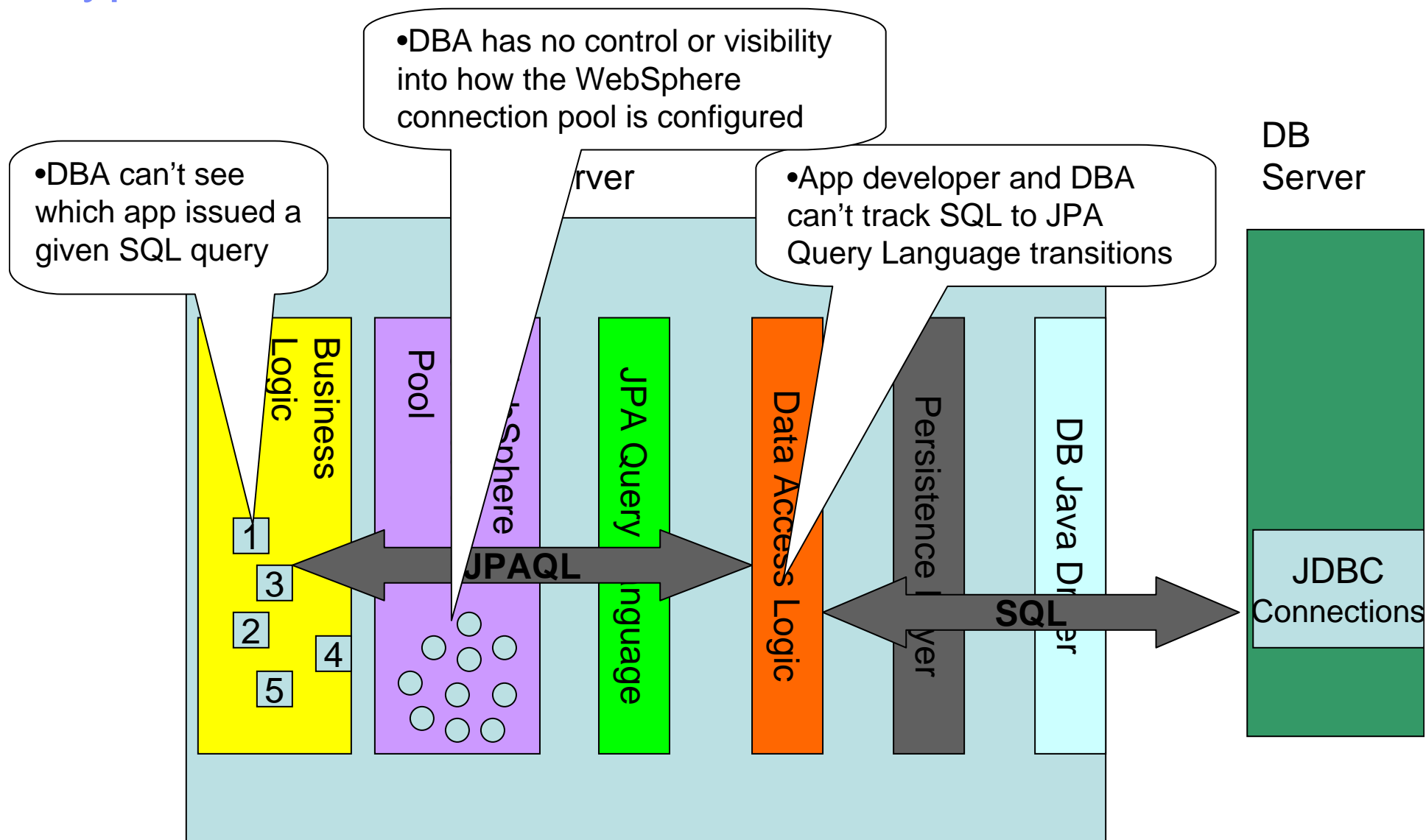
Simplifying Problem Determination Scenario

Java

Profiling

pureQuery

Application Developer

- **Available for each db access**
  - SQL text generated
  - Access path
  - Cost estimates
  - Estimated response time
  - Elapsed & CPU time
  - Data transfer (getpages)
  - Tuning advice

Database Administrator

- **Available for each SQL**
  - Application name
  - Java class name
  - Java method name
  - Java object name
  - Source code line number
  - Source code context
  - J-LinQ transaction name
  - Last compile timestamp

DRDA Extentions

pureQuery with IBM Runtime/Tooling

Application Server

EJB Query Language

Data Access Logic

Persistence Layer

DB2 Java Driver

A1
A4
A3
A2
A5
A6

DB2 or IDS

A1
A4
A5

Application
Meta data

A1 A2 A3

A4 A5 A6

Catalog data for SQL statements

A1 A2 A3 A4 A5 A6

Compile-time application details:
- Java class/line number for SQL
- Original query syntax
- Final SQL query syntax

Performance data by application:
- CPU and elapsed time
- Getpages, locks, etc.

App	CPU
A1	2.1
A4	8.3
A5	22.0

Where to get IBM Data Studio ?

- ## **IBM Data Studio**

  - www.ibm.com/software/data/studio

    - FAQs / Tutorials
    - Downloads
    - Forum / Blogs
    - Join the IBM Data Studio user community