Computer Associates®

Java and the Wild Wild Web
Crash Course No.2

Maria Sarikos
maria.sarikos@ca.com

Agenda

- Web Services
- XML
- General Performance Topics
- “New Universal Driver” DB2 V8

Agenda

- Web Services
- XML
- General Performance Topics
- “New Universal Driver” DB2 V8

Web Services

- **Why the need for Web Services?**
  - **Integrate existing systems**
    - Implement IT support for business processes that cover the entire business cycle
  - **Demand for technologies which have the following:**
    - Support the connection and/or sharing of resources & data
    - Support must be flexible and standardised
  - **Need to structure large applications into building blocks**
    - Ability to reuse well-defined components within different business processes
  - **Therefore a Service Orientated architecture must focus on *HOW* services are described and organised to support automatic dynamic discovery and use**

Web Services – Business Examples

- **Business information** *(eg. Hotel URL links to city map website)*
  - Sharing information with consumers other businesses
  - Web services can be expanded to reach other services such as integrated travel planning, weather reports, news streams etc

- **Business Integration** *(eg. eBay, entertainment booking system, internet travel agencies)*
  - Provide transactional, fee based services for customers
  - Global network of suppliers can be created
  - Web services can be implemented in auctions, e-marketplaces, etc

- **Business Process Externalisation** *(eg. Buying travel insurance when booking a flight)*
  - Dynamically integrate processes to a new solution or to other e-businesses
  - This is achieved by dynamically linking internal applications to partners/suppliers to either offer their services or complement their services with yours

Computer Associates®

Web Services – What’s involved in a Service-Orientated Architecture?

1. **Interoperability between diverse systems and programming languages**
   - Communication protocol

2. **Fully understandable and unambiguous description language**
   - Ability to access the provider system
   - Syntax of service interface must be platform independent and clearly defined

3. **Service Retrieval**
   - Services are put into categories depending on what they do and how they are invoked
   - These categories called Taxonomies are hierarchical

Computer Associates®

Web Services – What's involved in a Service-Orientated Architecture?

4. **Security is paramount**
   - Services as well as the data passed to and received from a service must be protected
   - Level of security depends on the participants and services
   - Service usage monitoring and security incident action triggers must be in place
   - BALANCE is key

Web Services
Architecture Characteristics

- **A Service orientated structure has a loose coupling between its ‘participants’**
  - this is what gives its flexibility

- **The client is not coupled to a server but to a <u>service</u>**
  - Integration to a server is outside the scope of client applications

- **Existing and new functional blocks (apps) are encapsulated into service components**

- **Functional components and their interfaces are separated**

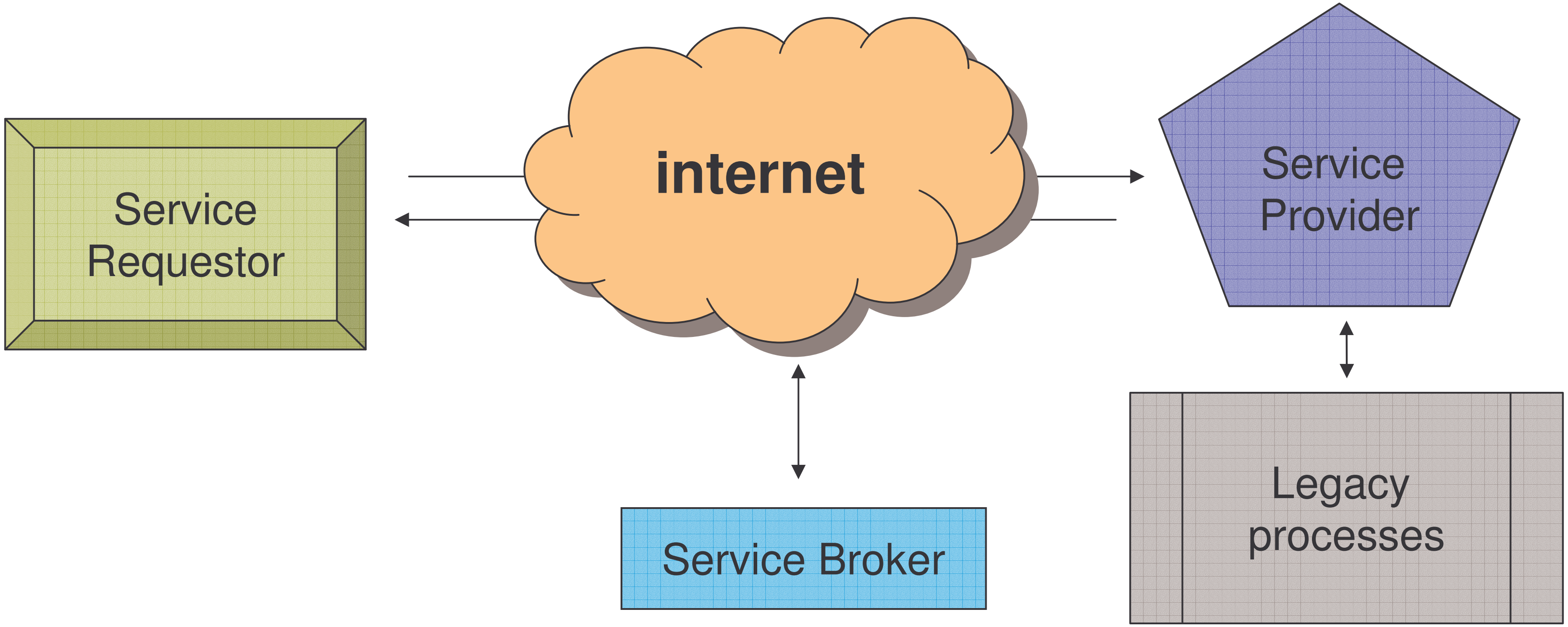
Web Services Architecture Characteristics

- **In complex applications, the control of processes can be easily isolated**
  - A business rule engine can be incorporated to control the workflow of a defined process
  - Depending on the workflow the engine calls the next appropriate service

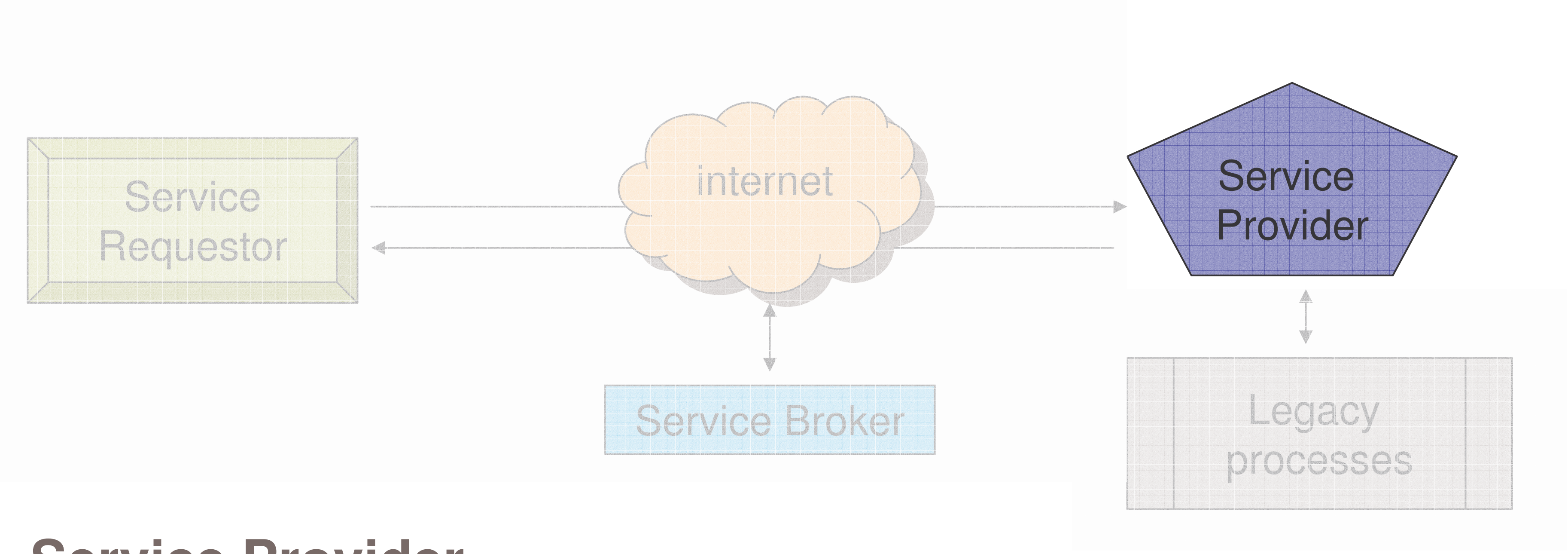
- **Services can be incorporated dynamically at runtime**

- **Bindings are specified using configuration files and are hence able to easily adapt to new requirements**

Web Services Architecture



Web Services Architecture

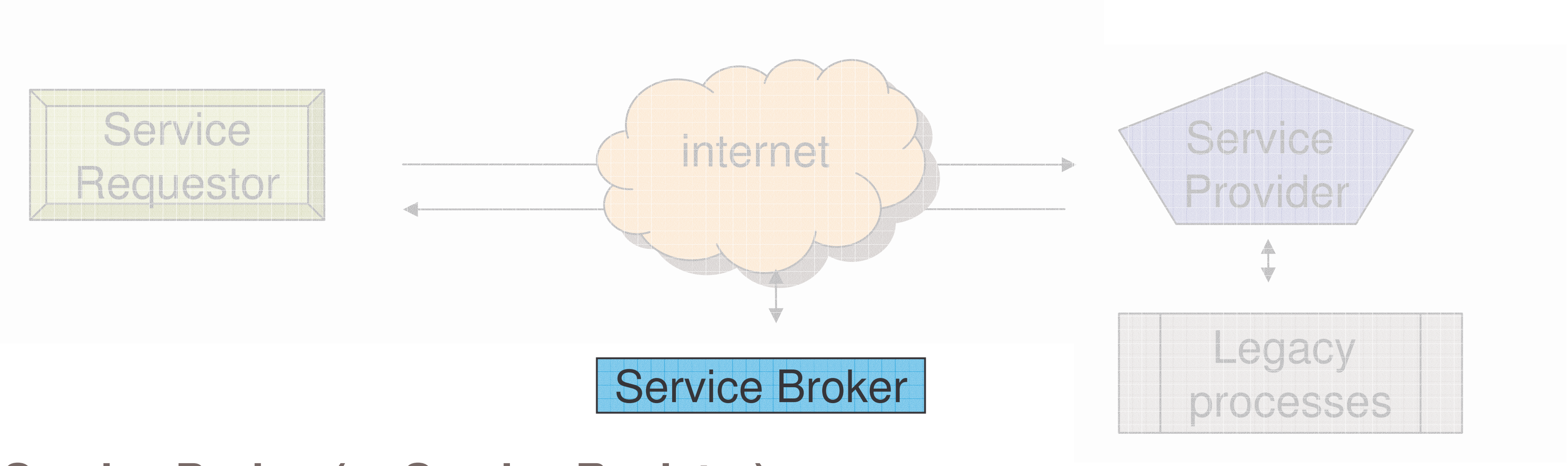


Service Provider

- Creates a web service
- Could publish its interface and access information to the service registry
- Each provider has to decide :
  - Which services to expose (trade-off between security and availability) ?
  - Which category the services should be listed in for a given broker service?
  - What type of agreements are required to use the service?

Computer Associates®

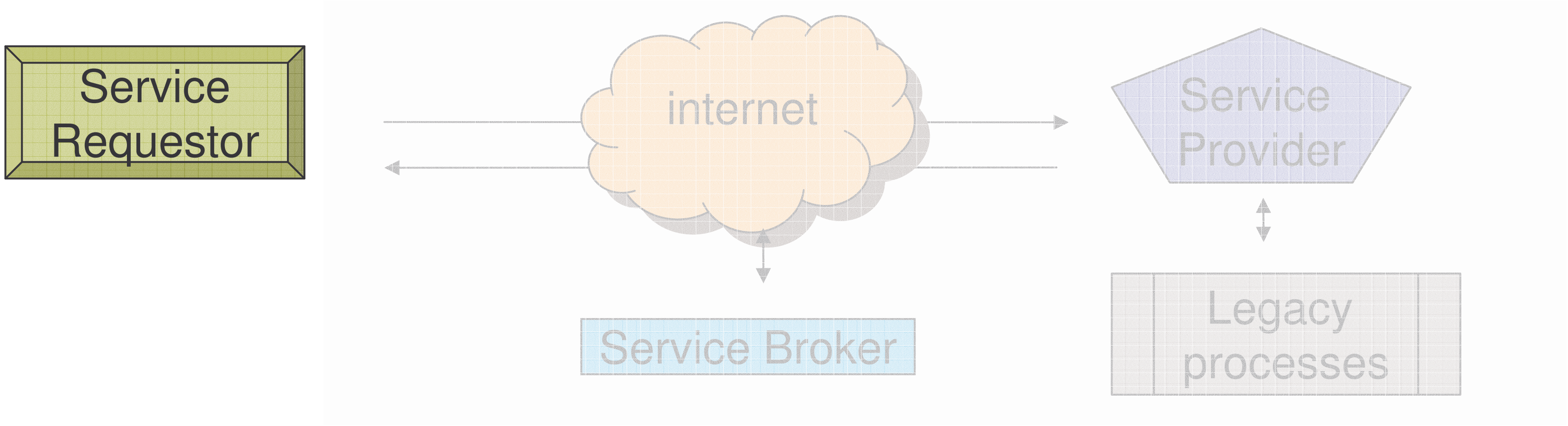
Web Services Architecture



Service Broker (or Service Registry)

- Responsible for making the web service interface and implementation access information available to potential service requestors
- Could publish its interface and access information to the service registry
- Public Broker – available to all over the internet
- Private Broker – available to a limited audience eg. company intranet
- Some brokers specialise in a wide variety of listings, others offer very secure listed services
- There are also brokers that catalog other brokers

Web Services Architecture



Service Requestor

- Locates entries in the broker registry using a variety of find operations
- Binds to the service provider and invokes one of its Web services
- Dynamic choice of services opens up a whole range of issues
  - How to choose the best service provider
  - How to access quality of service
  - How the service user can assess the risk of exposure to service supplier failures

Computer Associates®

Core Technologies used for Web Services

XML (eXtensible Markup Language)

- Underlies most of the specifications used for Web Services
- Generic language that can be used to describe any kind of content in a structure manner separated from its presentation to a specified device

SOAP (Simple Object Access Protocol)

- Similar to JDBC
- Network, transport, programming language
- Platform neutral protocol that allows a client to call a remote service
- Message format is XML

Core Technologies used for Web Services

UDDI (Universal Description, Discovery and Integration)

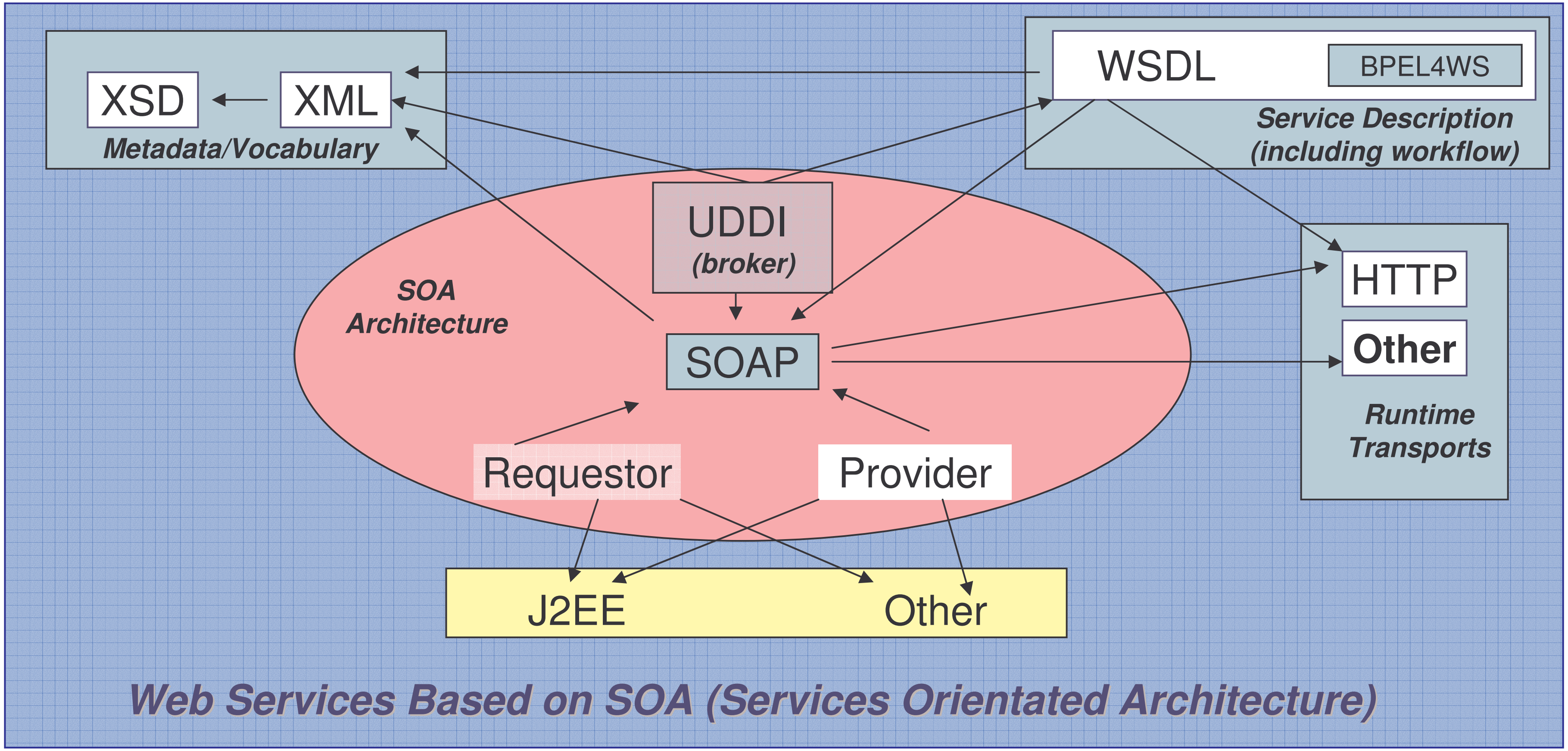
- Both a client side API and SOAP based server implementation
- Used to store and retrieve information on service providers and Web Services

WSDL (Web Services Description Language)

- Has an XML based interface and is an implementation description language
- Service provider uses a WSDL document to specify
  - The operations a Web service provides
  - Parameters and data types for these operations
- A WSDL document also contains service access information

Web Services based on SOA

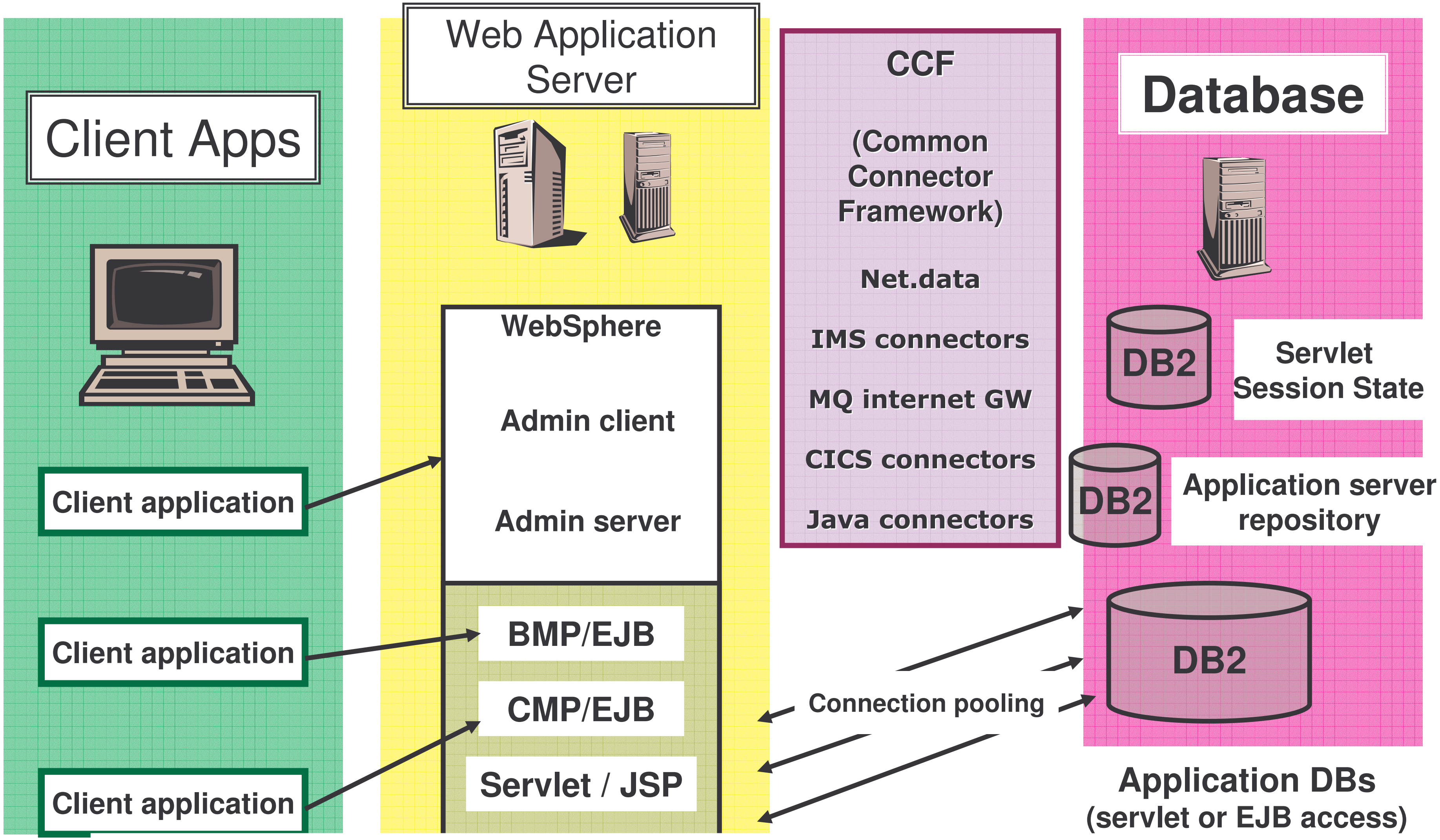
SOA (services orientated architecture)



WebSphere

- **Application Server**
  - Sits in the Middle-tier
    - Communicates with the back-end systems e.g. DBMS (DB2, IMS, Oracle etc) TP Monitors (CICS, IMS, Encina)
      - Many DBMS cannot understand commands written in HTML the WAS (web application server) acts as a translator
    - Communicates with front-end clients (e.g. web browsers)
    - Provides a runtime environment for business logic
  - Has naming service and uses JDNI (Java Naming Directory Interface)
  - Provides security
    - Controls access to web resources e.g. HTML pages, JSPs, EJBs etc.
  - Transactional
  - Work Load management
  - Implemented on J2EE standards

WebSphere Application Server - Overview



Client Apps

Client application

Client application

Client application

Web Application Server

WebSphere

Admin client

Admin server

BMP/EJB

CMP/EJB

Servlet / JSP

CCF

(Common Connector Framework)

Net.data

IMS connectors

MQ internet GW

CICS connectors

Java connectors

Connection pooling

Database

DB2

Servlet Session State

DB2

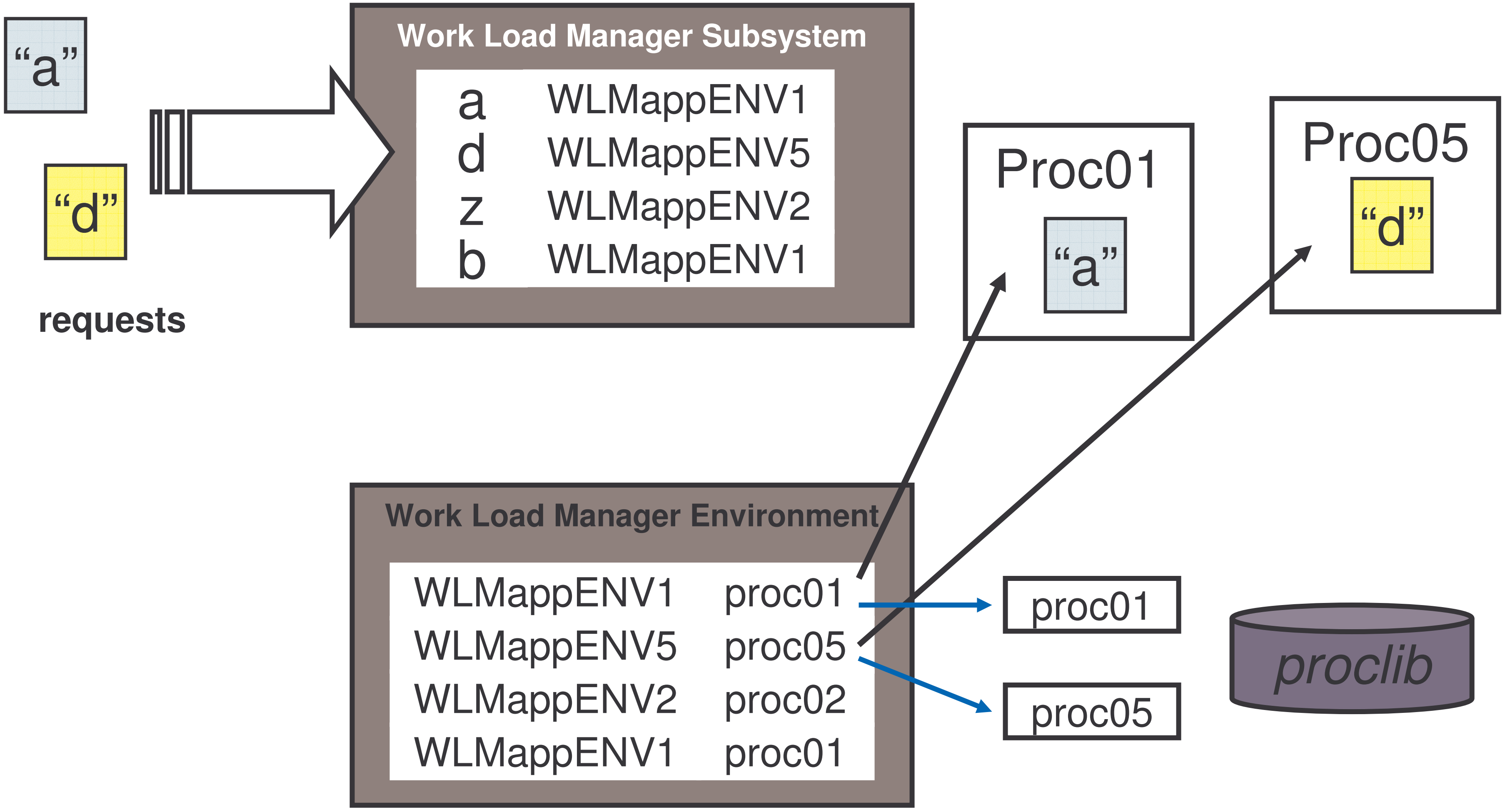
Application server repository

DB2

Application DBs (servlet or EJB access)

Computer Associates®

z/OS Workload Manager Applications Environments



Work Load Manager Subsystem

a	WLMappENV1
d	WLMappENV5
z	WLMappENV2
b	WLMappENV1

Work Load Manager Environment

WLMappENV1	proc01
WLMappENV5	proc05
WLMappENV2	proc02
WLMappENV1	proc01

Agenda

- Web Services
- XML
- General Performance Topics
- “New Universal Driver” DB2 V8

XML (eXtensible Markup Language)

- **XML is an open standard protocol**
  - Provides a mechanism to create metalanguage that can define other markup languages
  - Almost any type of data can be easily defined in XML

- **Two major advantages to using XML**
  - It is written in plain text format which allows it to be compatible with existing computing environments
  - Developers can create their own markup tags or elements to best represent the structure and nature of the data. – When you define XML documents you are actually defining a language to suit your application needs

- **XML is great as a format for describing data in a way that can be shared by many applications on many platforms**
  - Humans and computers can understand the data because the author can describe data by defining each tag and how it relates to the structure

Computer Associates®

XML (eXtensible Markup Language)

- **XML can be used as a universal data format and for exchanging info between intranets or internets using Web browsers and Java**

- **XML is portable and self defining**
  - This includes the code page used, which makes it easy for other users (businesses) to understand the tags
  - XML tags represent the logical structure of the data

XML (eXtensible Markup Language)

```
<?xml version="1.0" encoding="UTF-8" ?>
<Recipe TimeToPrepare="25" CookMethod="Bake"
Difficulty="Easy for beginners "
Serves="1" Category="Tea time Treat">
  <Title>Moist Chocolate Cake</Title>
  <Ingredients>
     <Ingredient Name="Flour" Amount="550” Unit="ml” /ingredient>
     <Ingredient Name="Sugar" Amount="350” Unit="ml" /ingredient>
     <Ingredient Name="Baking Powder" Amount="20” Unit="ml" /ingredient>
     <Ingredient Name="Cocoa" Amount="60” Unit="ml" /ingredient>
     <Ingredient Name="Salt" Amount="1” Unit="ml" /ingredient>
     <Ingredient Name="Egg Yolks" Amount="4” Unit="n/a" /ingredient>
     <Ingredient Name="Oil" Amount="275” Unit="ml" /ingredient>
     <Ingredient Name="Vanilla Essence" Amount="5” Unit="ml" /ingredient>
     <Ingredient Name="Boiling Water" Amount="275” Unit="ml" /ingredient>
     <Ingredient Name="Egg Whites" Amount="4” Unit="n/a" /ingredient>
  </Ingredients>
  <Preparation>
    <Step>Preheat oven to 180&#176. Grease and line two 200mm cake tins with wax paper </Step>
    <!--&#176; is the degree symbol -->
    <Step>Sift dry ingredients together </Step>
    <Step>Beat egg yolks, oil, vanilla well together </Step>
    <Step>Add to dry ingredients with boiling water and mix well </Step>
    <Step>Whip egg whites until stiff with an electric beater and fold into mixture </Step>
    <Step>Pour batter into prepared tins and bake for 25-30 mins </Step>
  </Preparation>
  <Comment>Warning -Very Addictive</Comment>
  <Nutrition>
    <Calories>1000</Calories>
    <FatGrams>45</FatGrams>
    <CarboGrams>40</CarboGrams>
    <ProteinGrams>15</ProteinGrams>
   </Nutrition>
</Recipe>
```

XML (eXtensible Markup Language)

XML Declaration is located at start of document specifies the version of the XML used and char encoding used

Elements are the most common form of mark-up

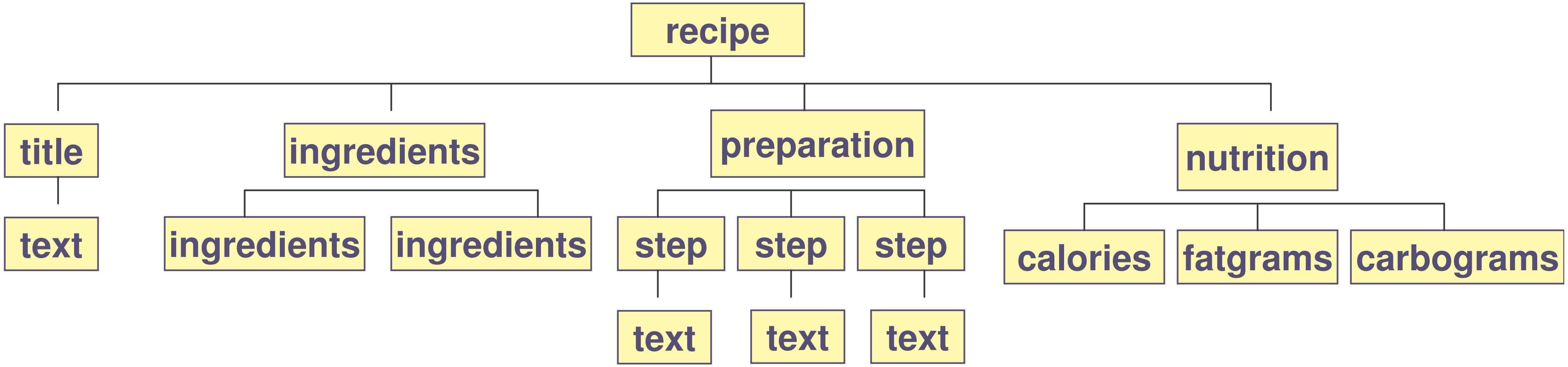
- They have start end tags
- They are delimited by angle brackets
- Element tags begin with a "Tag Name"
- They may contain nested elements, text, or other sub-structures
- An element start tag may also contain attributes

Attributes are name-value pairs that may occur within start tags

- Attributes may contain only text
  No other sub-structure is allowed
- Many attributes are allowed, but only a single instance of an attribute name is allowed within a single element start tag i.e. can't have two names for an Ingredient

```
<?xml version="1.0" encoding="UTF-8" ?>
<Recipe TimeToPrepare="25" CookMethod="Bake"
        Difficulty="Easy for beginners "
        Serves="1" Category="Tea time Treat">
 <Title>Moist Chocolate Cake</Title>
 <Ingredients>
   <Ingredient Name="Flour" Amount="550” Unit="ml"  /ingredient>
   <Ingredient Name="Sugar" Amount="350” Unit="ml" /ingredient>
   <Ingredient Name="Baking Powder" Amount="20” Unit="ml" /ingredient>
   <Ingredient Name="Cocoa" Amount="60” Unit="ml" /ingredient>
   <Ingredient Name="Salt" Amount="1” Unit="ml" /ingredient>
   <Ingredient Name="Egg Yolks" Amount="4” Unit="n/a" /ingredient>
   <Ingredient Name="Oil" Amount="275” Unit="ml" /ingredient>
   <Ingredient Name="Vanilla Essence" Amount="5” Unit="ml" /ingredient>
   <Ingredient Name="Boiling Water" Amount="275” Unit="ml" /ingredient>
   <Ingredient Name="Egg Whites" Amount="4” Unit="n/a" /ingredient>
 </Ingredients>
 <Preparation>
  <Step>Preheat oven to 180&#176. Grease and line two 200mm cake tins with wax paper </Step>
  <!--&#176; is the degree symbol -->
  <Step> Sift dry ingredients together </Step>
  <Step> Beat egg yolks, oil, vanilla well together </Step>
  <Step> Add to dry ingredients with boiling water and mix well </Step>
  <Step> Whip egg whites until stiff with an electric beater and fold into mixture </Step>
  <Step> Pour batter into prepared tins and bake for 25-30 mins </Step>
 </Preparation>
 <Comment>Warning -Very Addictive</Comment>
 <Nutrition>
  <Calories>1000</Calories>
  <FatGrams>45</FatGrams>
  <CarboGrams>40</CarboGrams>
  <ProteinGrams>15</ProteinGrams>
 </Nutrition>
</Recipe>
```

XML (eXtensible Markup Language)



- Elements may be nested that why they take on a hierarchical structure
- Elements may also have lists of children
- Note that the order of attributes is not significant
- XML documents can’t contain binary data

XML (eXtensible Markup Language)

- **Applications use parsers to access data**

- **SAX** (**S**imple **A**PI for **X**ML)
  - A SAX parser fires events as it parses and it calls routines in the program to handle ‘events’
  - No buffering is required
  - SAX parsers are quite fast
  - However what if you want to navigate the document?

startDocument

startElement

startElement

endElement

Computer Associates®

XML (eXtensible Markup Language)

- **DOM (D**ocument **O**bject **M**odel)
  - Reads the document and represents in memory as a hierarchical tree
  - Applications call methods to ‘traverse’ the tree and extract the data
  - DOM parsers use more resources. This is because they need :
    - CPU to build the tree
    - Memory to store the tree
  - Provide a higher level of functionality
    - Navigational ability ie. ability to move forward, backward, up, down etc
    - Ability to update/create new documents

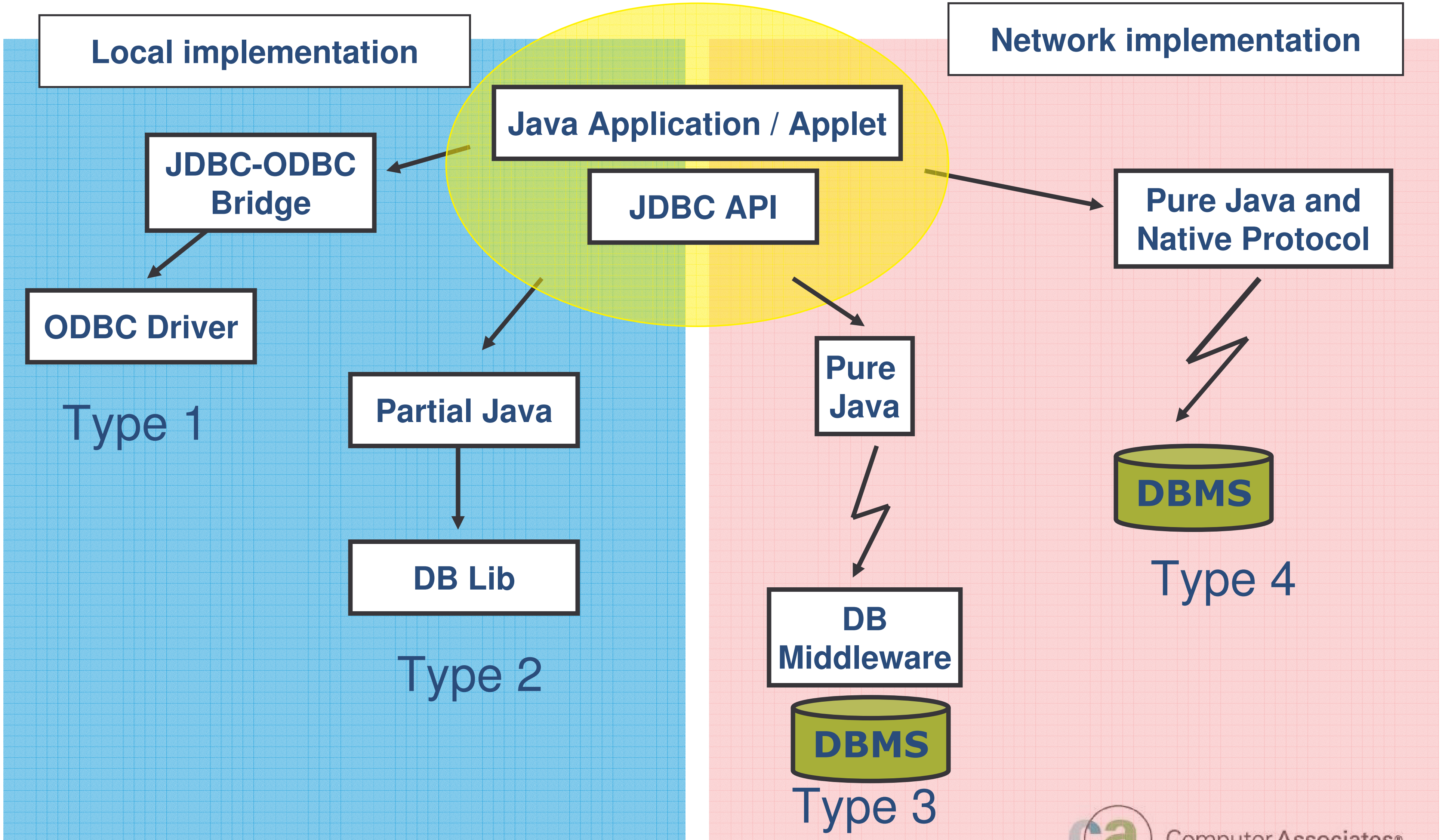
```
node.getFirstChild()
node.getAttributes()
node.nextsibling()
```

Computer Associates®

Agenda

- Web Services
- XML
- General Performance Topics
- “New Universal Driver” DB2 V8

JDBC – Driver Types



Computer Associates®

Ensure you have well tuned Dynamic SQL

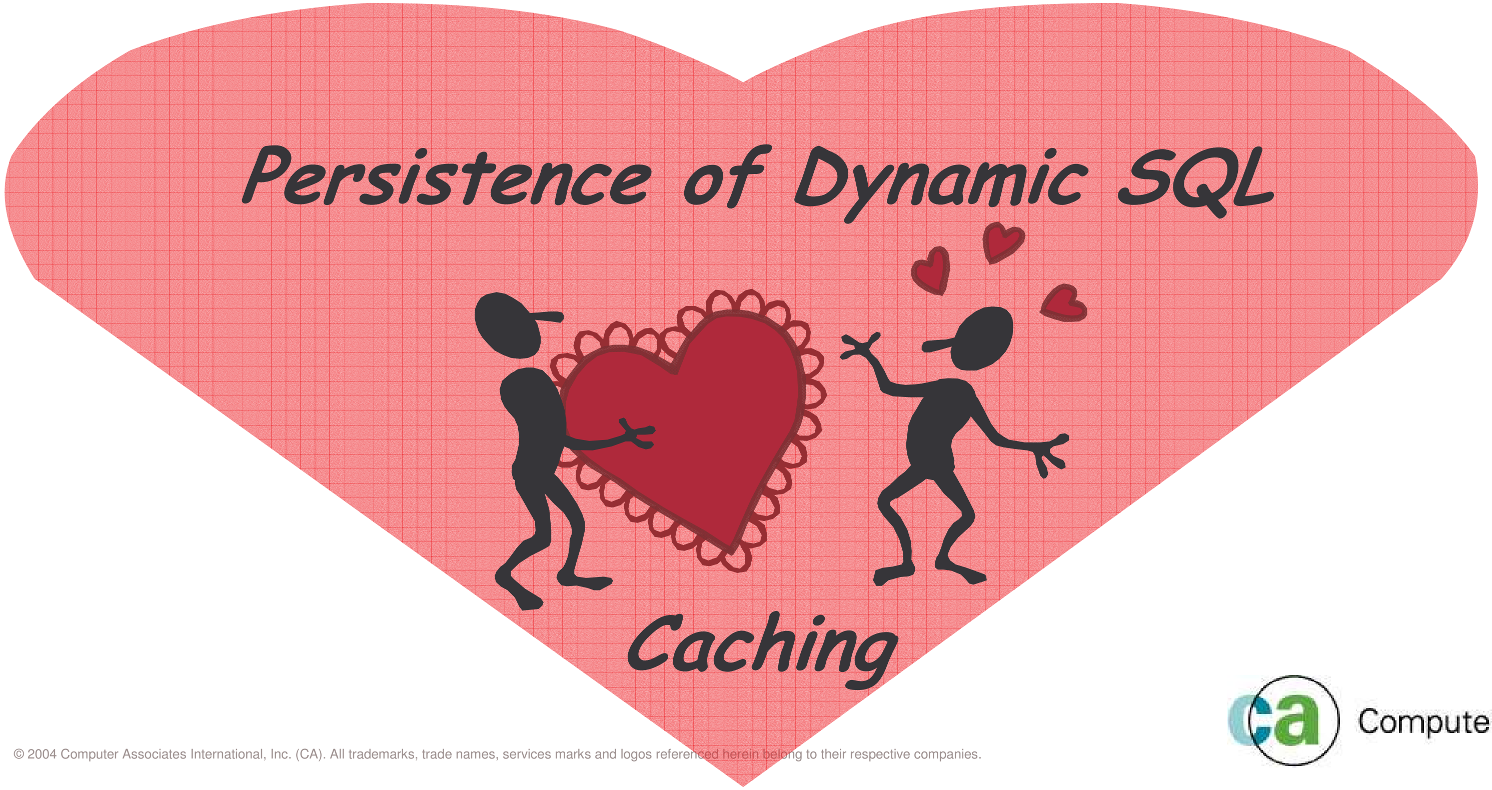
- **Determining Access Paths**
  - Catalog statistics
  - Capturing the SQL

- **Programming Dependencies**
  - What SQL is out there?
  - What if I change my DB2 Objects?
  - What is the most popular column/table?
  - What indexes can give me better performance?

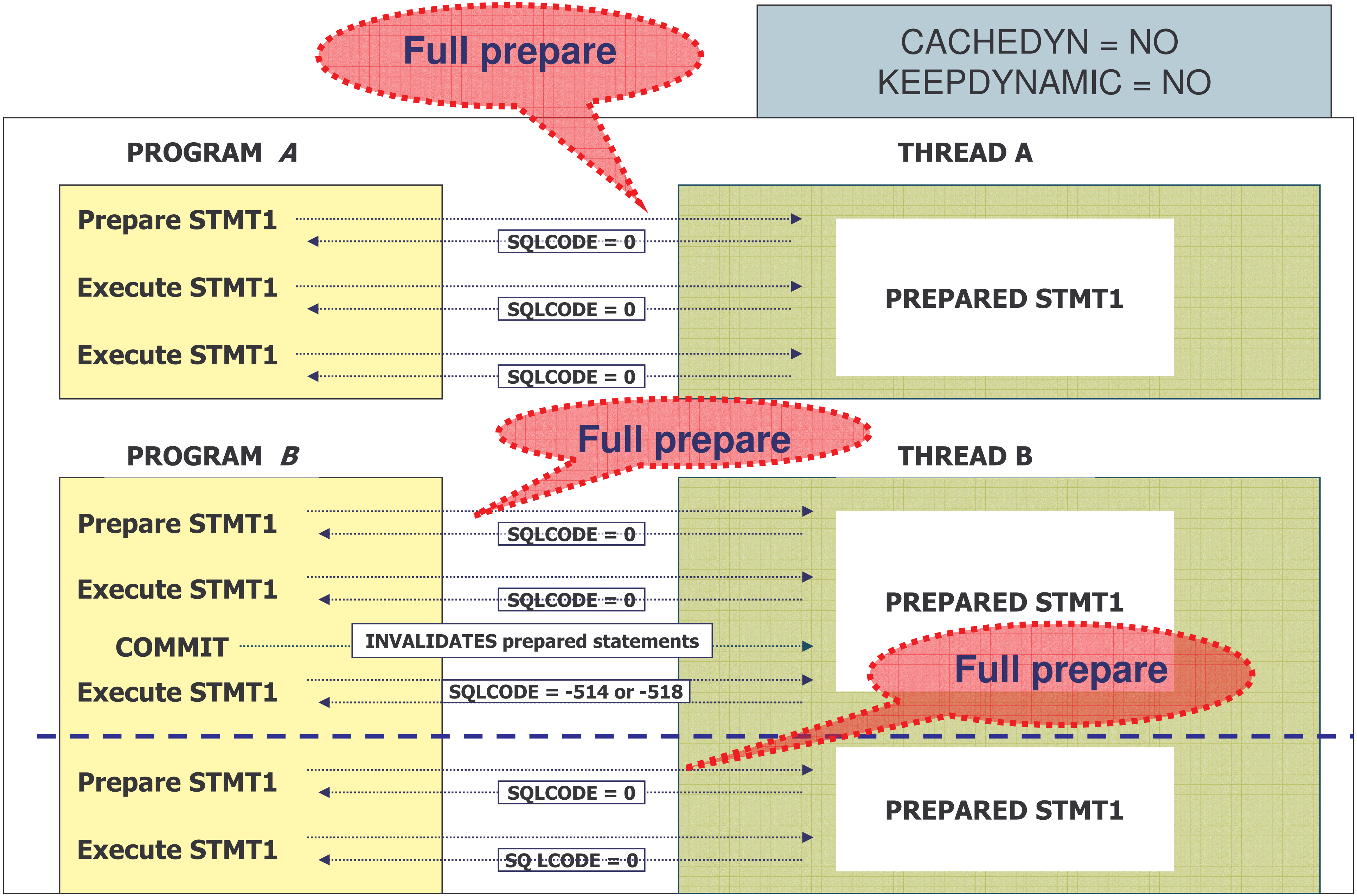
Ensure you have well tuned Dynamic SQL

- **Why is dynamic SQL so expensive?**
  - No SQL statement reuse!

- **How do I get dynamic SQL to use less resource?**



No Caching



CACHEDYN = NO
KEEPDYNAMIC = NO

Full prepare

PROGRAM *A*

THREAD A

Prepare STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

PREPARED STMT1

Full prepare

PROGRAM *B*

THREAD B

Prepare STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

PREPARED STMT1

COMMIT

INVALIDATES prepared statements

Full prepare

Execute STMT1

SQLCODE = -514 or -518

Prepare STMT1

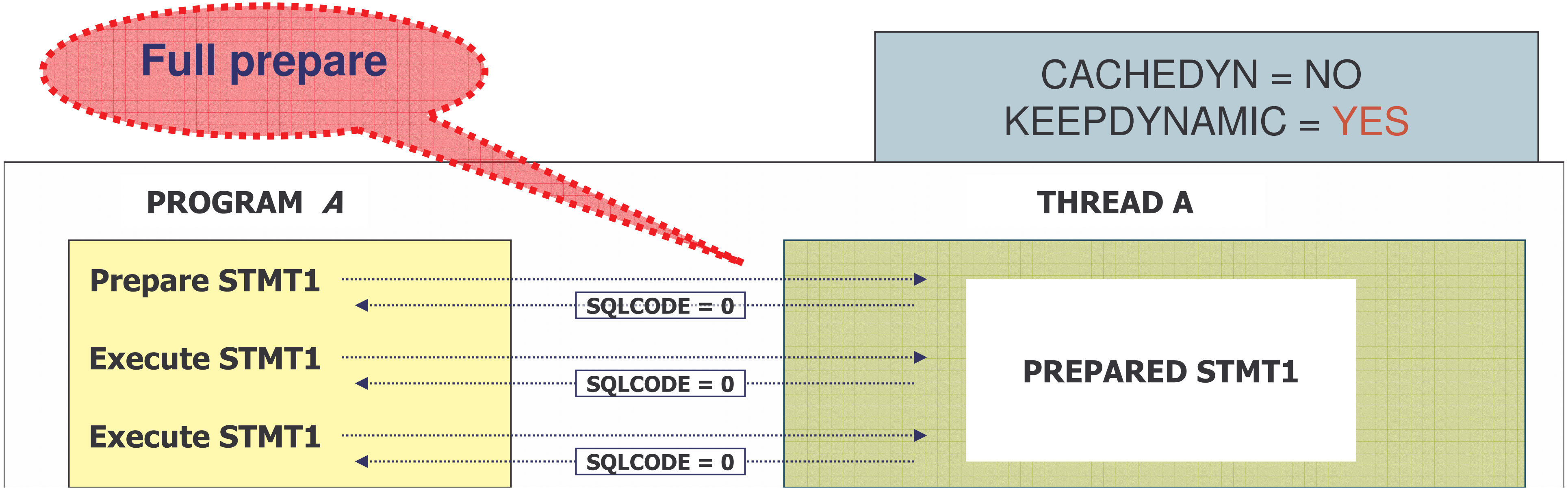
SQLCODE = 0

PREPARED STMT1

Execute STMT1

SQ LCODE = 0

Local Dynamic SQL caching



Full prepare

CACHEDYN = NO
KEEPDYNAMIC = YES

PROGRAM *A*

THREAD A

Prepare STMT1

SQLCODE = 0

Execute STMT1

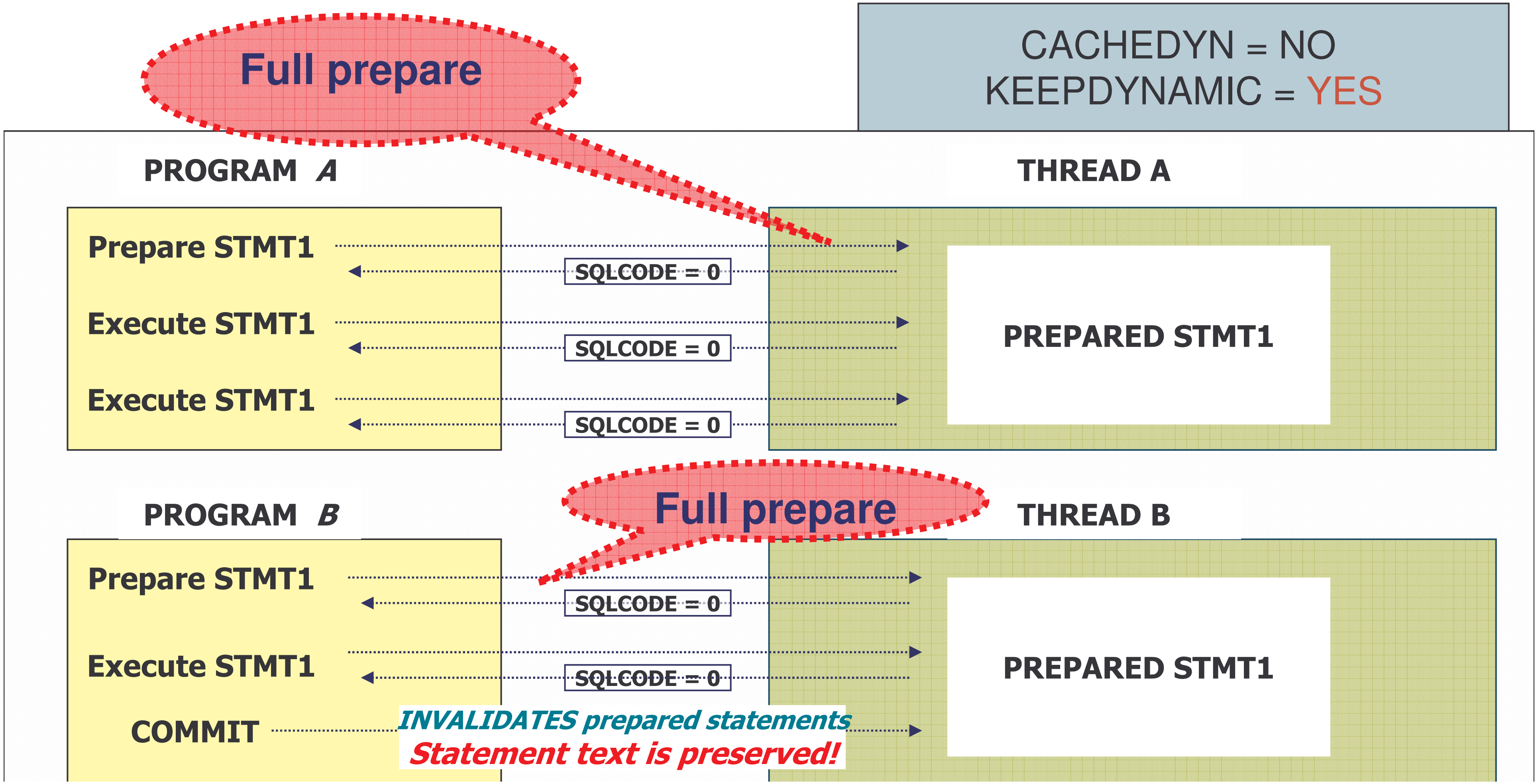
SQLCODE = 0

Execute STMT1

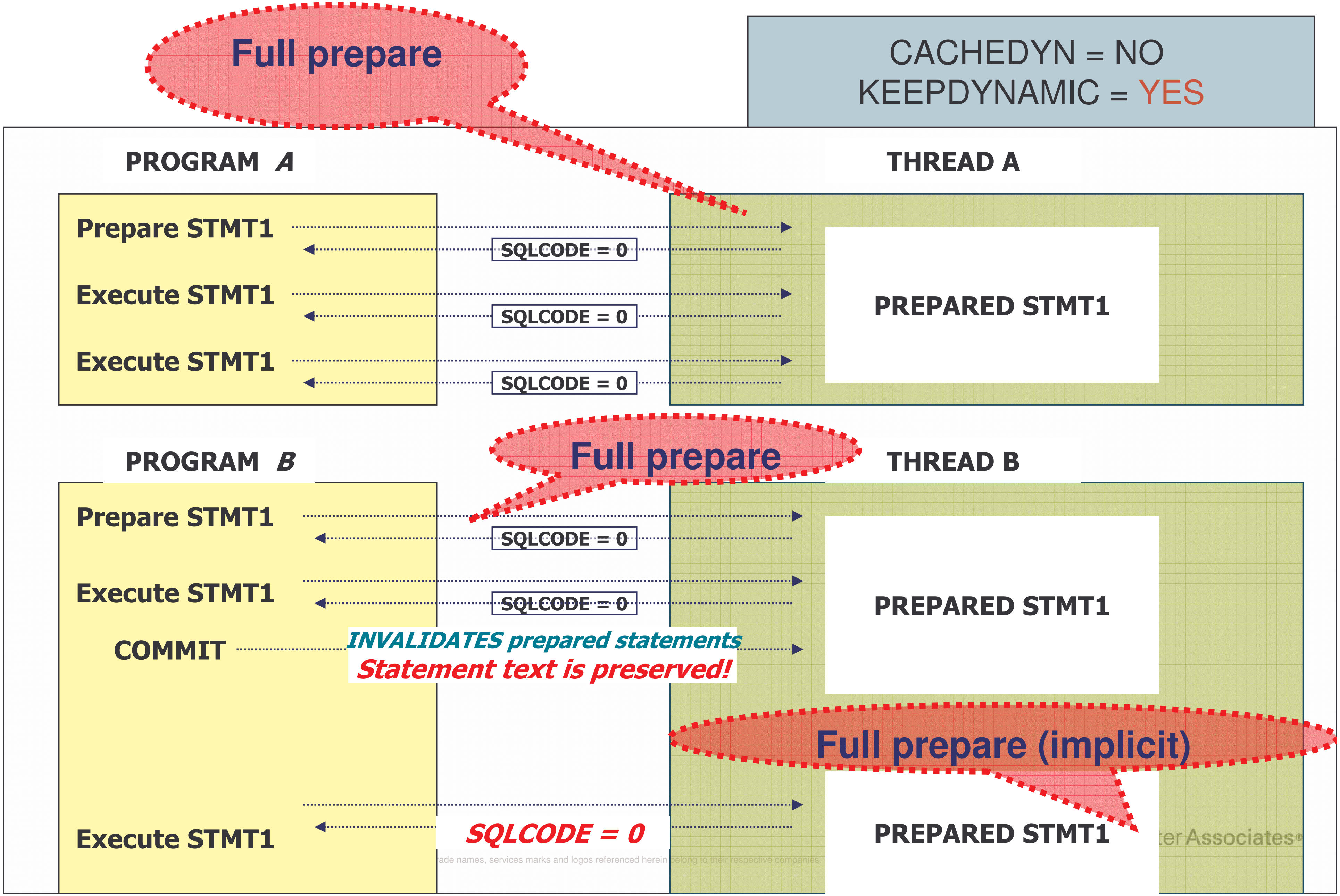
SQLCODE = 0

PREPARED STMT1

Local Dynamic SQL caching



Local Dynamic SQL caching



CACHEDYN = NO
KEEPDYNAMIC = YES

Full prepare

PROGRAM *A*

Prepare STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

THREAD A

PREPARED STMT1

Full prepare

PROGRAM *B*

Prepare STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

COMMIT

INVALIDATES prepared statements
Statement text is preserved!

THREAD B

PREPARED STMT1

Full prepare (implicit)

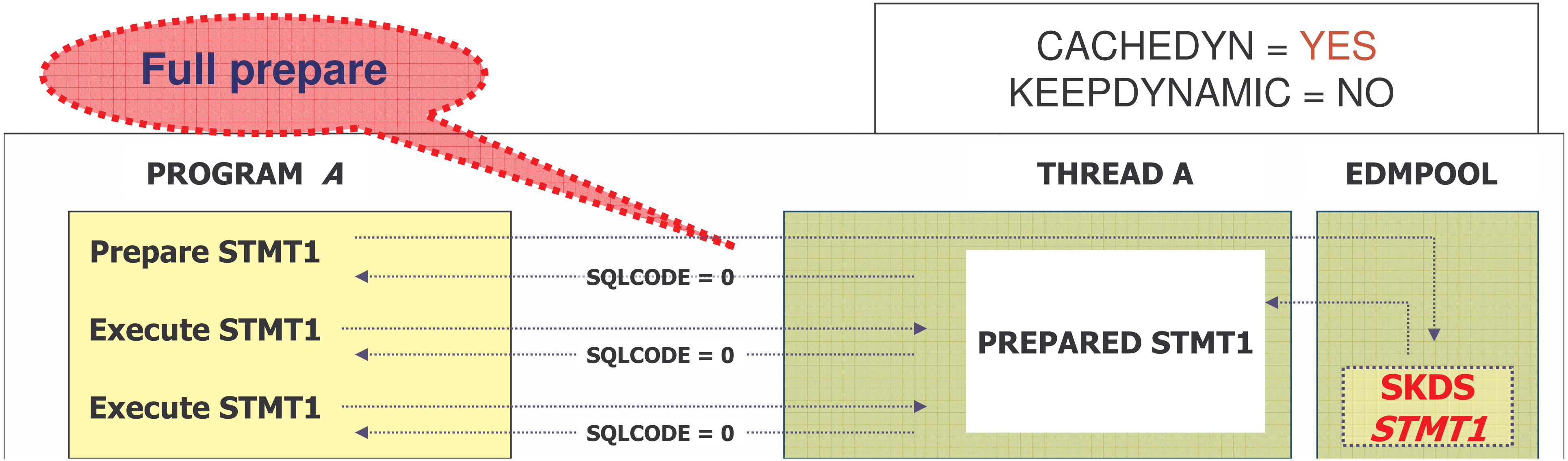
Execute STMT1

SQLCODE = 0

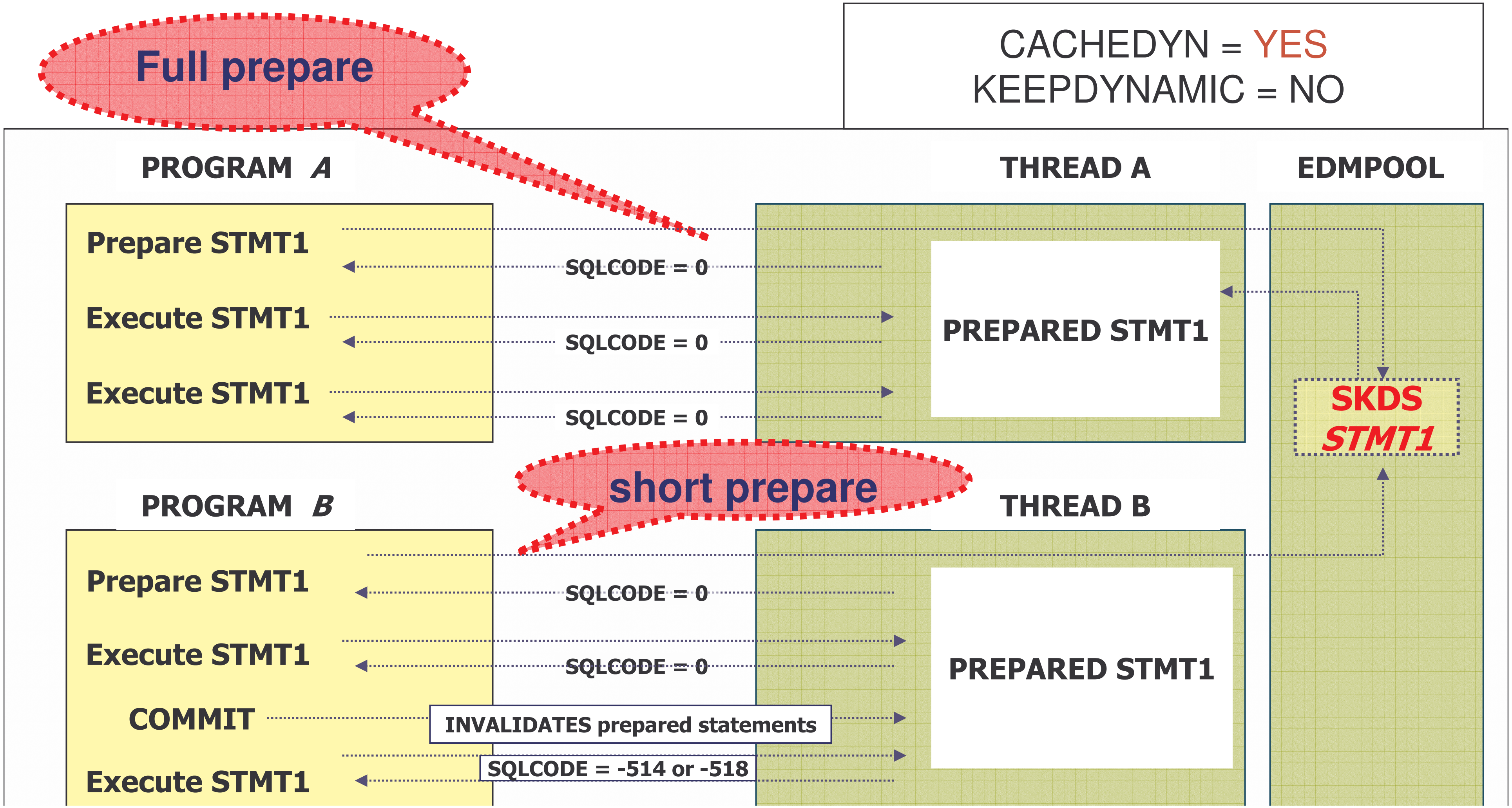
PREPARED STMT1

Global Dynamic SQL caching

CACHEDYN = YES
KEEPDYNAMIC = NO



Global Dynamic SQL caching



CACHEDYN = YES
KEEPDYNAMIC = NO

Full prepare

short prepare

PROGRAM *A*

Prepare STMT1

Execute STMT1

Execute STMT1

PROGRAM *B*

Prepare STMT1

Execute STMT1

COMMIT

Execute STMT1

THREAD A

PREPARED STMT1

THREAD B

PREPARED STMT1

EDMPOOL

SKDS
STMT1

SQLCODE = 0

SQLCODE = 0

SQLCODE = 0

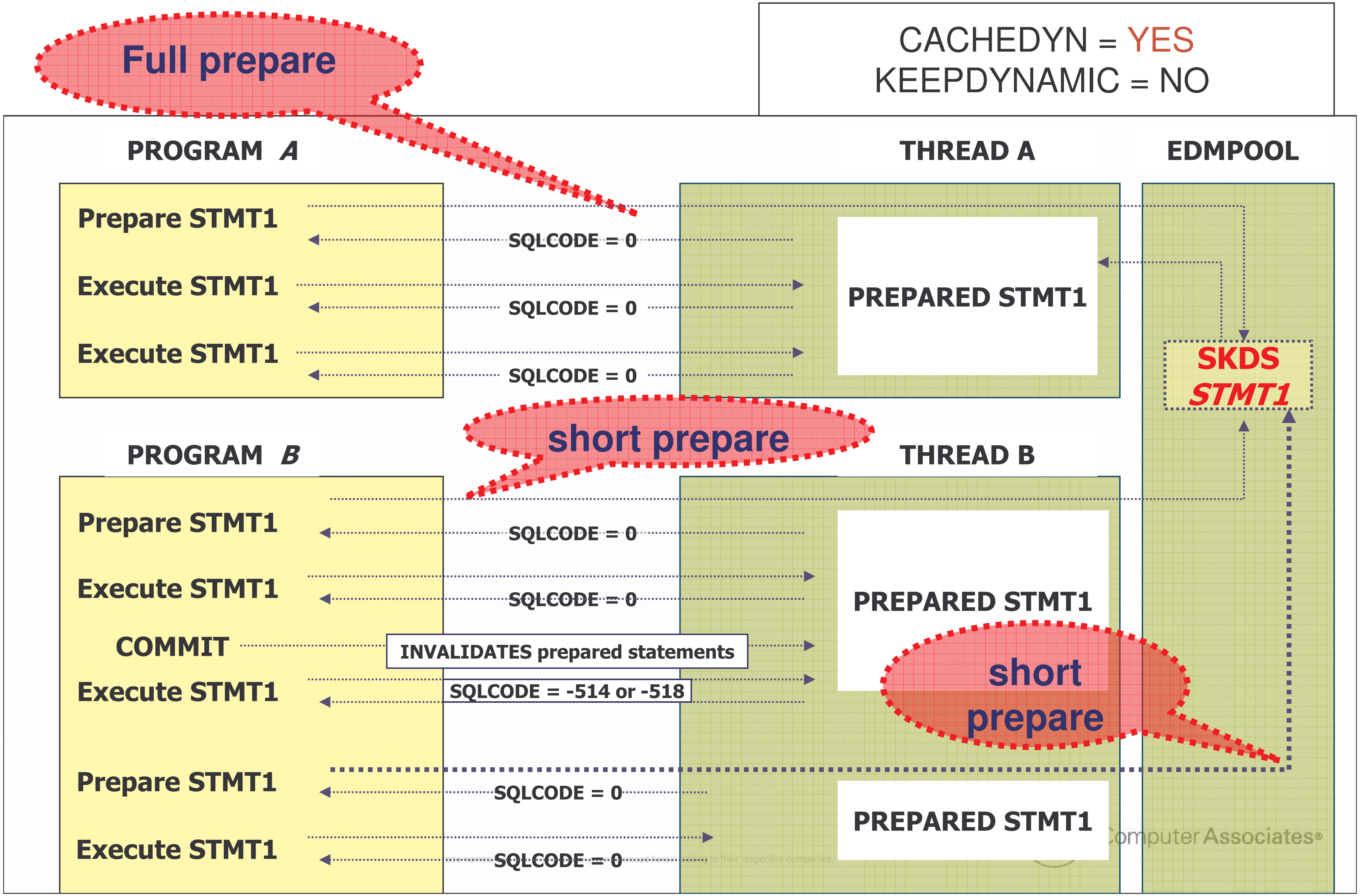
SQLCODE = 0

SQLCODE = 0

INVALIDATES prepared statements

SQLCODE = -514 or -518

Global Dynamic SQL caching



CACHEDYN = YES
KEEPDYNAMIC = NO

Full prepare

PROGRAM *A*

THREAD A

EDMPOOL

Prepare STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

PREPARED STMT1

SKDS
STMT1

short prepare

PROGRAM *B*

THREAD B

Prepare STMT1

SQLCODE = 0

Execute STMT1

SQLCODE = 0

COMMIT

INVALIDATES prepared statements

Execute STMT1

SQLCODE = -514 or -518

PREPARED STMT1

short prepare

Prepare STMT1

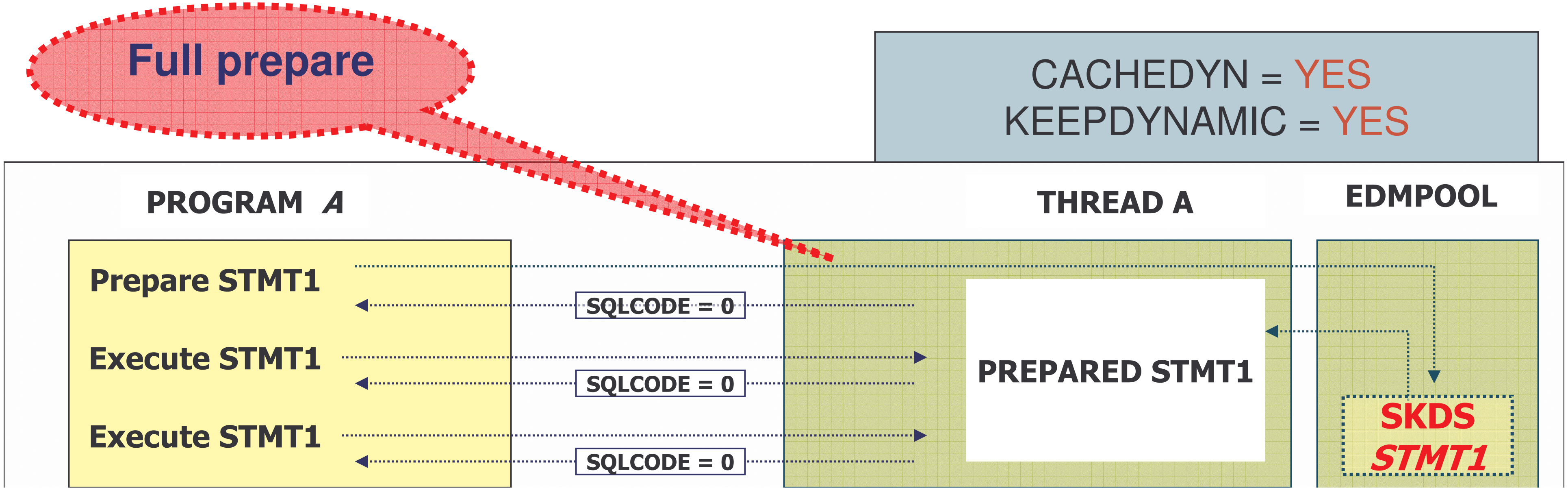
SQLCODE = 0

Execute STMT1

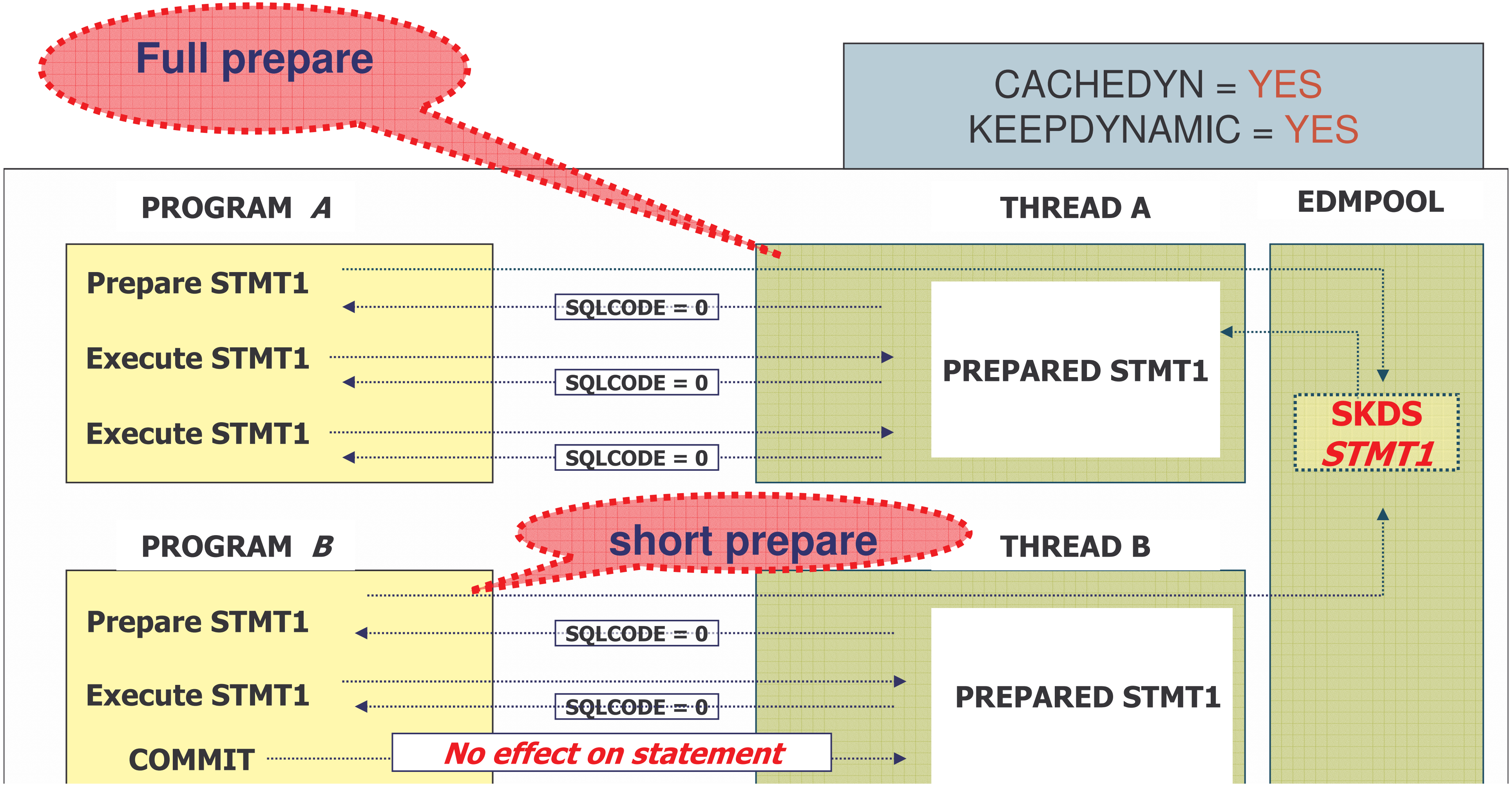
SQLCODE = 0

PREPARED STMT1

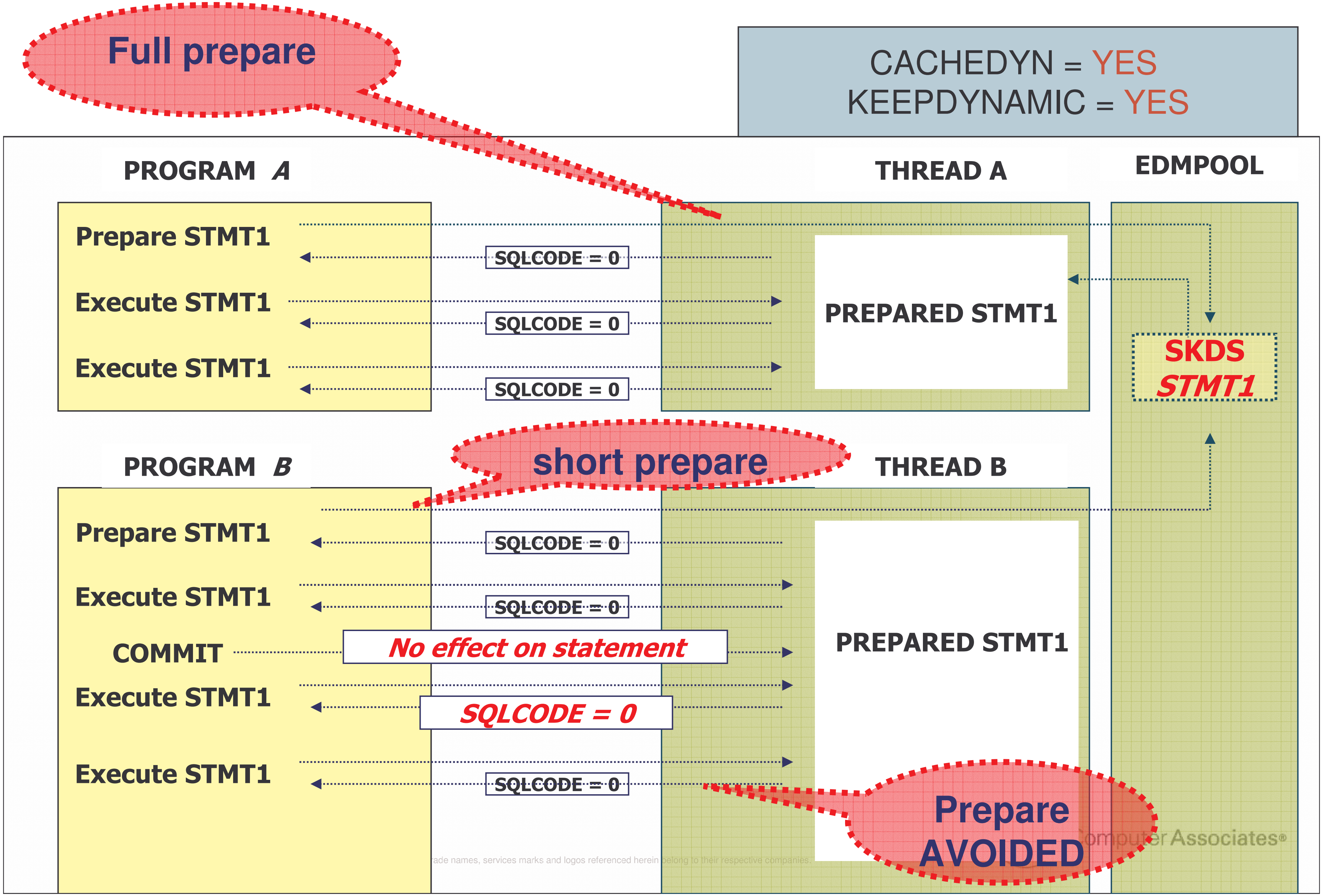
Full Dynamic SQL caching



Full Dynamic SQL caching



Full Dynamic SQL caching



CACHEDYN = YES
KEEPDYNAMIC = YES

Dynamic SQL Caching

For Statement REUSE to occur …

- **These must be 100% identical**
  - Same length
  - Trailing blanks
  - ATTRIBUTES
  - Authorisation ID
  - Plan or package bound with same values
  - Special registers
  - Declared cursor characteristics
  - Parser options
  - Parallelism

  - Use PARAMETER MARKERs ‘?’

Dynamic SQL Caching

DYNAMIC SQL statement invalidation

- **GLOBAL CACHE**
  - No free pages in the EDMPOOL
  - DROP, ALTER, REVOKE executed on anything plan is dependent on
  - RUNSTATS

- **LOCAL CACHE**
  - MAXKEEPD is exceeded
  - STATEMENT ID reused by a prepare
  - Thread deallocates
  - DROP, ALTER, REVOKE executed on anything plan is dependent on
  - ROLLBACK and re-signon
  - RUNSTATS

Dynamic SQL Caching

- **Increase size of EDM Pool until get the hit ratios you want**

- **Rule of Thumb → make it 10 times bigger than you largest DBD**
  - Easy way to see size → -DISPLAY DB(???)

- **DBD chunks**
  - Modify DBD to get into 32K chunks if DBD created before V6

- **Ensure that EDM Pool I/Os are few and fast**
  - Hit Ratio
    - Put SKDS in dataspaces
      (EDMSPAC > 0 & CACHEDYN = YES)
  - Minimise size of DBDs with MODIFY

Dynamic SQL Caching

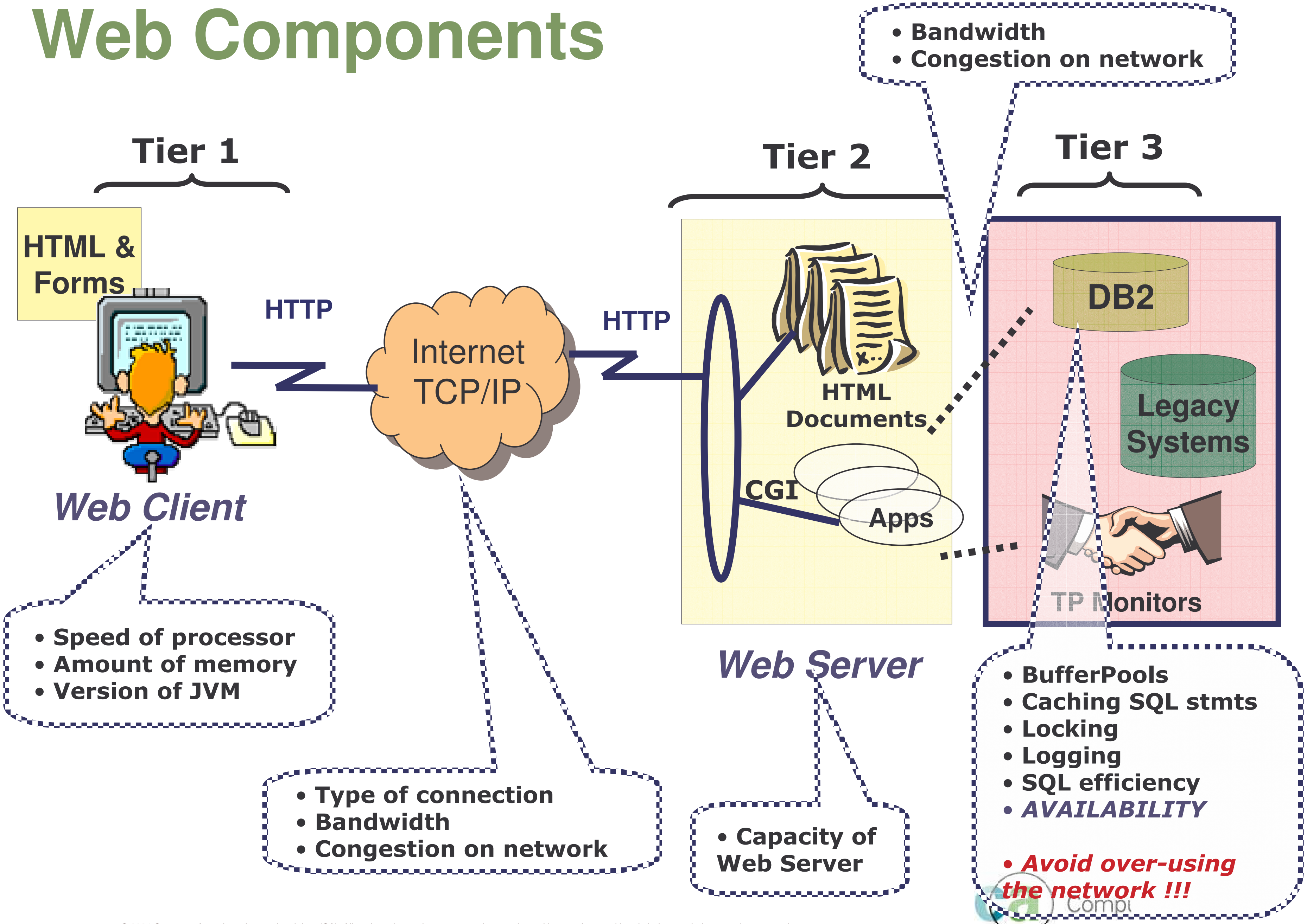
Too small an EDM Pool causes

- Fewer threads to run concurrently due to lack of space
- Increased response time due to loading SKCT / SKPT / DBD from DASD
- Performance degradation due to auth check for each stmt if the SKCT cannot remain in EDM Pool

Performance vs Memory Conservation – Trade-off

Computer Associates®

Web Components



- Bandwidth
- Congestion on network

- Speed of processor
- Amount of memory
- Version of JVM

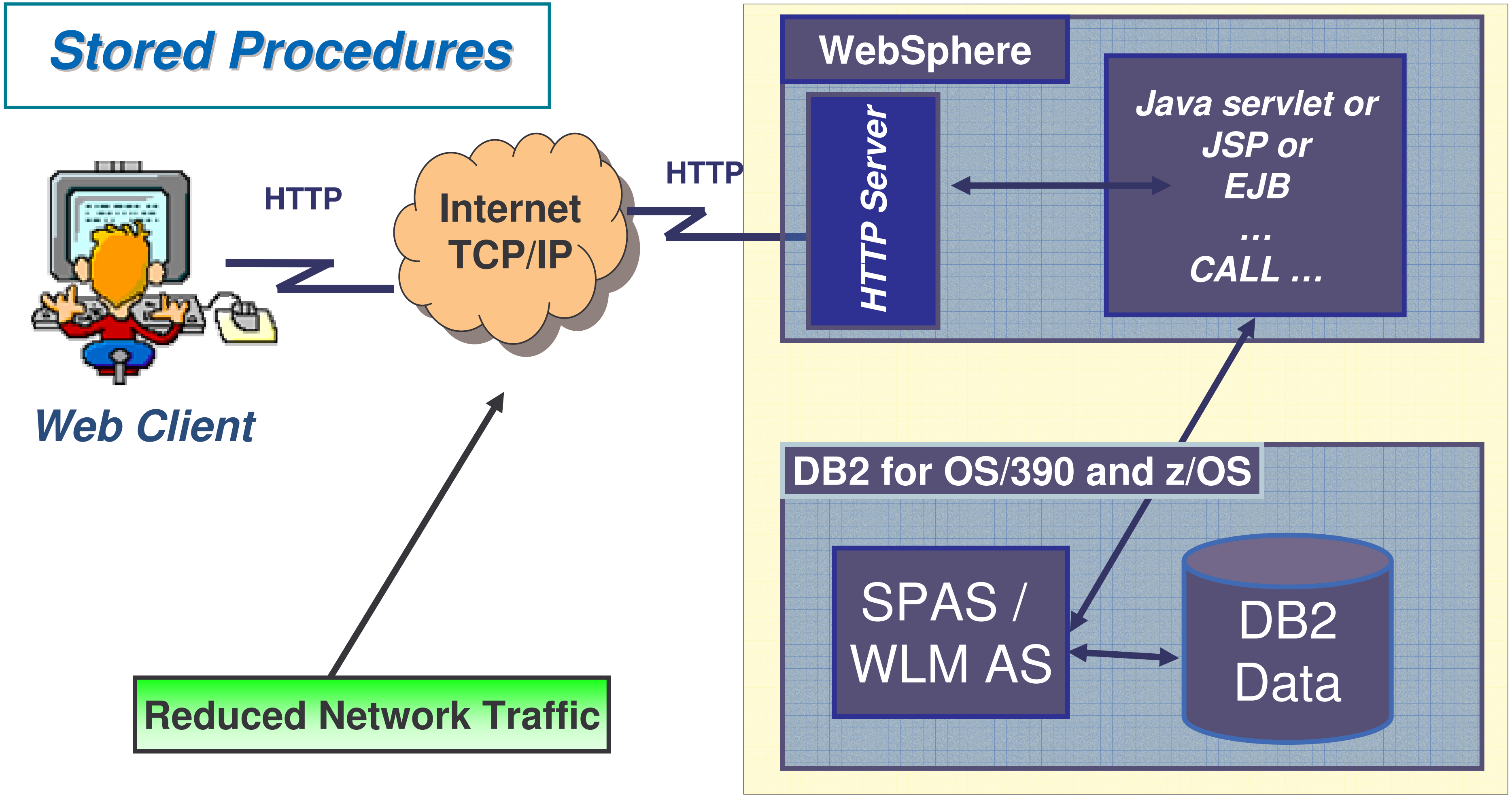
- Type of connection
- Bandwidth
- Congestion on network

- Capacity of Web Server

- BufferPools
- Caching SQL stmts
- Locking
- Logging
- SQL efficiency
- ***AVAILABILITY***

- ***Avoid over-using the network !!!***

Separate excessive SQL from heavily used Java programs



SQL Performance Recommendations

- ***Turn autocommit off***
  - By default when you open a database connection via the DriverManager class
    - It has autocommit property set to true by default

- ***Only retrieve/update columns required***
  - Column processing is one of the major CPU eaters
    - Strings for character string cols must be converted between Unicode (Java) & EBCDIC/ASCII (DB2)
    - A Java object is created per column per row !!! … for those data types that are not primitive data types in Java eg. Char string cols

SQL Performance Recommendations

- ***Store numbers as numeric types***
  - Consider storing tel no’s as INTEGER instead of character
  - Saves the overhead of creating an object (if col is declared NOT NULL) and EBCDIC/Unicode conversion

- ***Use matching data types – non matching getxxx causes overhead***
  - Use the recommended mappings of DB2 to Java data types
  - Non-matching data type may result in a poor access path
  - While is syntactically OK to retrieve a TIMESTAMP column into a String variable, you should not do so
  - String is less efficient because the SQLJ runtime has to format TIMESTAMP column into String format
  - Using a java.sql.Timestamp variable allows to control the format of the timestamp

SQL Performance Recommendations

- ***Release Resources***
  - Close & release resources when they are no longer used
  - JDBC driver maintains its own links to resources – which are only released when closed
  - **Close ResultSets** – if not done the JVM garbage collection cannot reclaim the objects
  - **Close PreparedStatements** – if not done the underlying cursor is held for the life of the PreparedStatement
  - **Close CallableSections** – else the application may run out of callable sections
  - Release resources in the case of failure
    - **Java try** / **finally construct** is well suited for this purpose
    - **SQLJ automatically generates the code to release statements** However you still have to close the iterators yourself

SQL Performance Recommendations

- ***Use DB2 built-in functions***
    - DB2 has many useful built-in functions that are more efficient than their Java counterparts
    - Example
        - When retrieving a fixed-width character data column, you may want to get rid of the trailing blanks that DB2 appends whenever the value is shorter than the columns length.
        - Java → String.trim() method
        - DB2 → RTRIM
            - equivalent is easier and more efficient to use, because you can incorporate the TRIM function within the SQL stmt

SQL Performance Recommendations

- ***Customise with Online Checking enabled***
  - Very important when you have predicates using host variables
  - Host variables should match the corresponding columns in data type and size
  - For a predicate to use a matching index scan
    - Definition in the Java package must match the definition in the DB2 catalog (ie. data type & length)
    - String objects do not have a concept of length – this info can only be obtained from the DB2 catalog
    - If online checking is disabled → could get a TS scan instead of a index access
    - Character columns are not the only ones affected – Numerics are also affected
    - If a host variable type of long is used to match to an INTEGER data type, the optimizer will choose a non-matching index scan because the predicate has to be evaluated at Stage 2 instead of Stage 1

Computer Associates®

SQL Performance Recommendations

- ***Use JDBC DataSource Connection Pooling***
  - Re-using the DB2 connection thread

DataSource definition

(this below would be executed only once by DBA)

```
ds = new com.ibm.db2.jcc.DB2DataSource();
ds.setDatabaseName("TESTDB");
```

A Pooled Connection within an Application

```
 //get connection from pool
Connection Conn1 = ds.getConnection("user","password");
// Turn off auto commit default
Conn1.setAutoCommit(false);
....
Conn1.close();
```

Computer Associates®

System Level Performance Tuning

- ***Tune the JVM Heap size***
  - In Java when accessing relational data, a lot of objects are created & then destroyed
  - JVM heap size plays an important role in the overall Java performance
  - DEFAULT initial heap size = 1MB
  - DEFAULT max heap size = 8MB
  - The default sizes are insufficient and cause poor performance
  - Good idea to set the
    - initial heap size = max. heap size = large value
    - 300-400MB heap sizes are not uncommon
  - Therefore garbage collection is not triggered so often
    - Reduces the repeated scanning of long living objects

System Level Performance Tuning

- ***Get the latest code & maintenance***
  - Keep current with the upgrades to the JDBC driver
  - There is constant improvement being made to CPU performance (column processing overheads)

- ***Turn on Dynamic Statement Caching***
  - CACHEDYN=YES in DSNZPARM
  - Dynamically prepared SQL stmts are cached across transaction boundaries

- ***Consider DB Server Queuing - Review MAXDBAT***
  - Number of remote concurrent database threads
  - Too few, could result in queuing for available threads

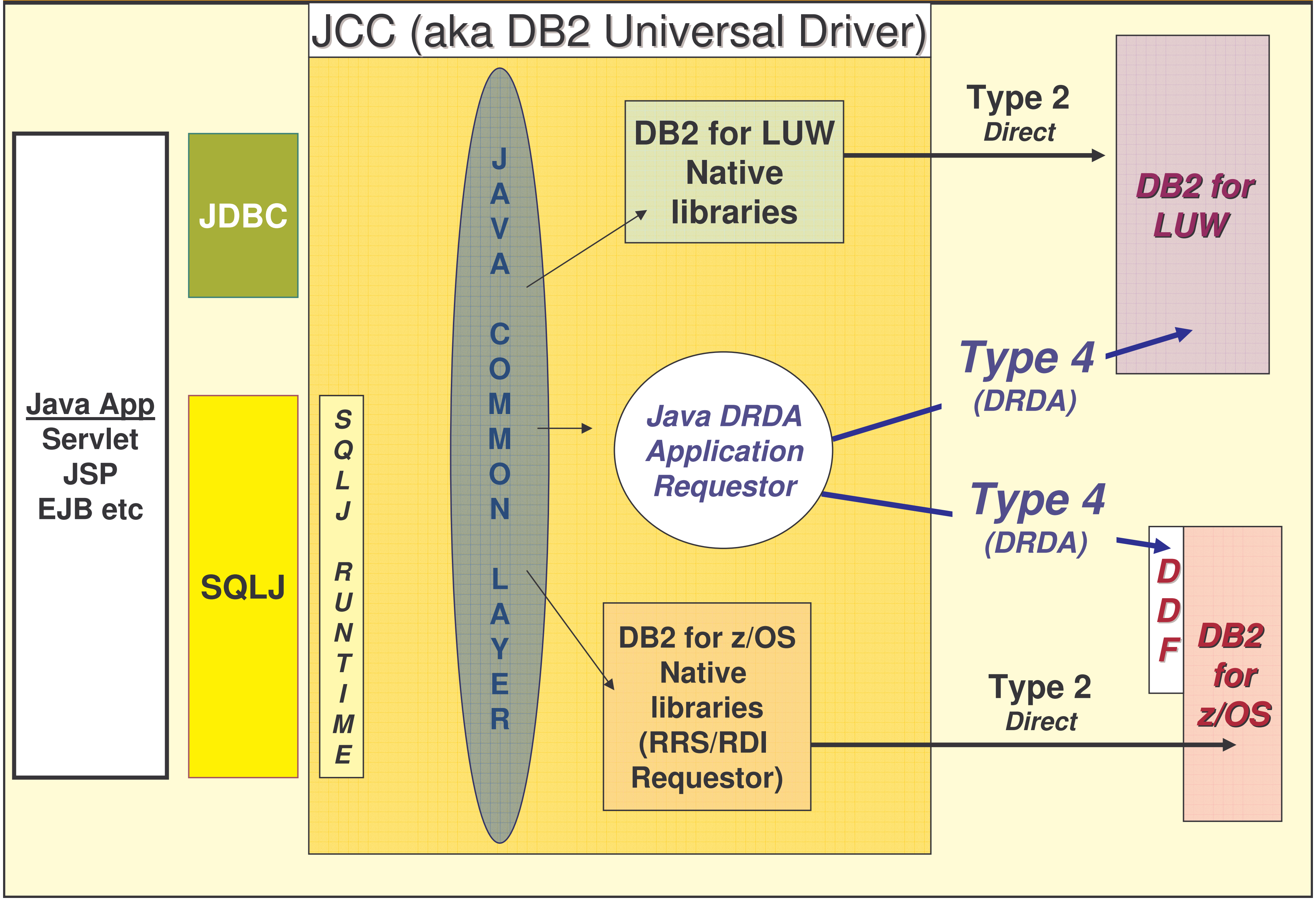
Agenda

- Web Services
- XML
- General Performance Topics
- “New Universal Driver” DB2 V8

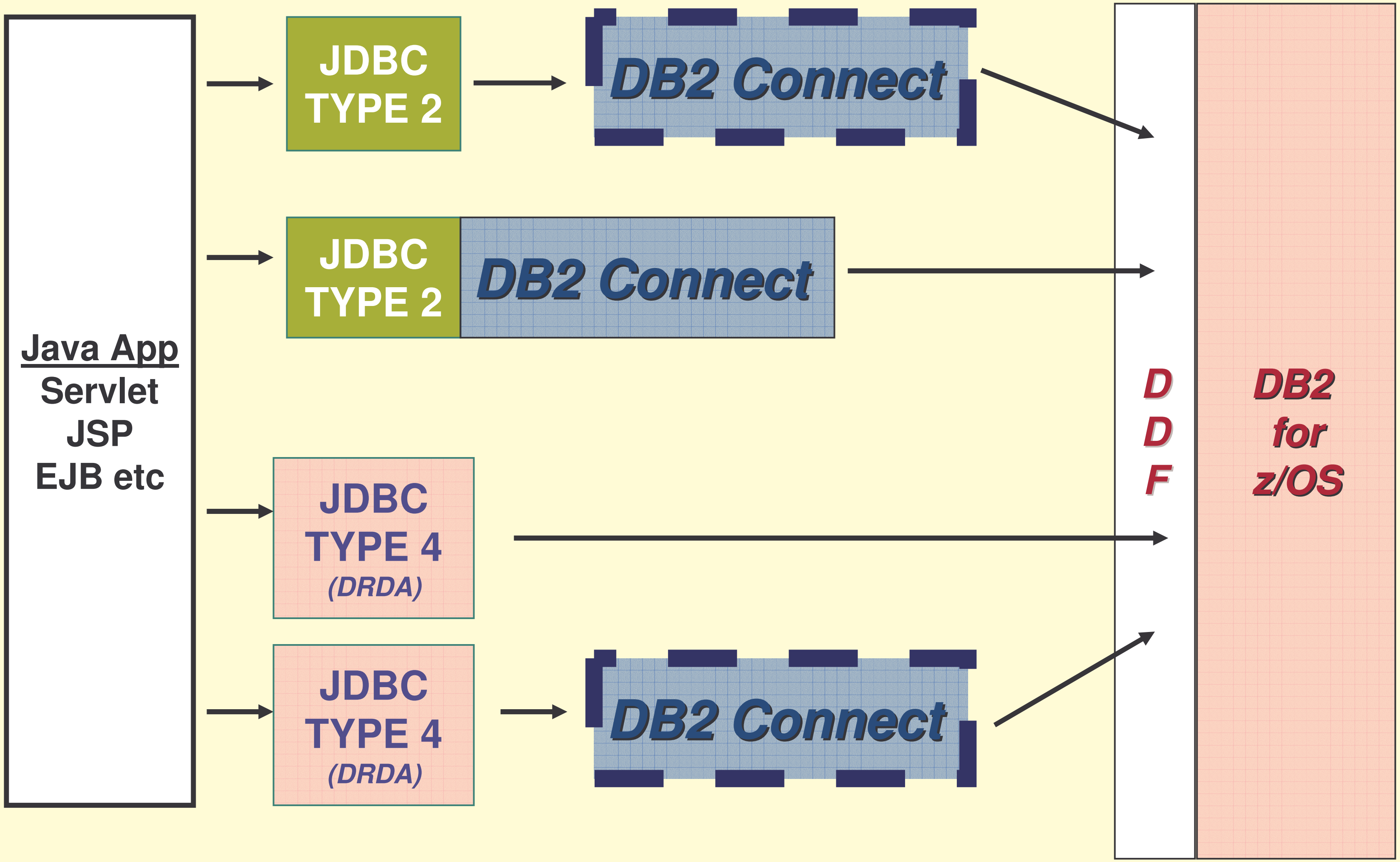
Universal Driver for SQLJ and JDBC

- **Functionality Enhancements for Type 2 and Type 4 drivers**
- **Fully Compliant with JDBC 3.0 standard**
- **Functionality for DB2 LUW and z/OS is EXACTLY the same!**
- **Why use the Universal Driver?**
  - Reduce porting errors
  - Common code for Type 2 and Type 4 driver
  - Reduce the client footprint
  - Provide full Java application development process for SQLJ
  - Improve tracing capabilities

JCC - (DB2 Universal Driver for Java Common Connectivity)



Connectivity to DB2 *from a non-z/OS* platform



Universal Driver for SQLJ and JDBC

Benefits for DB2 for z/OS V8

- Improves family compatibility
- Better DRDA performance – Private Protocols eliminated for DB2 LUW
- Easy installation and deployment
  - No DLL or runtime dependencies
  - Installation = copy of a .jar and .zip file

Universal Driver for SQLJ and JDBC

- **Universal Driver will eventually replace existing legacy implementations of JDBC**

- **Subtle behavioural differences from legacy drivers can be expected**
  - Existing drivers will not have the exact same behaviour
  - Impossible to match JDBC behaviour on LUW & OS390 in all cases

- **Migration should be done gradually and in a controlled manner**

- **db2sqljupgrade utility for DB2 for z/OS for serialized profiles**

Universal Driver for SQLJ and JDBC

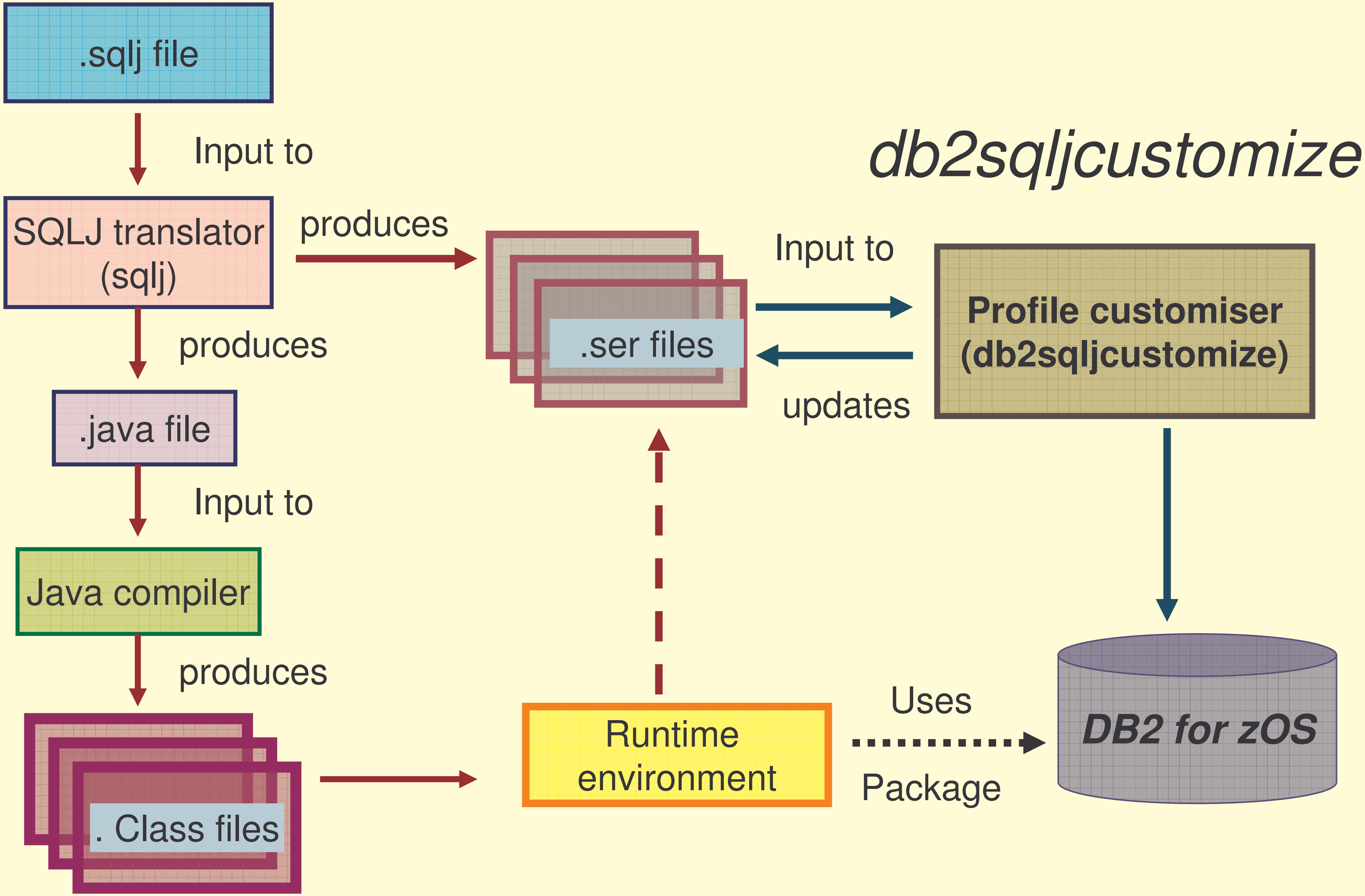
- **Licensing !!!**
  - Technically speaking you do not need a DB2 Connect to use the JCC T4 driver to connect to DB2 for zOS
  - But you still have to get a DB2 Connect license!!!

Universal Driver for SQLJ and JDBC

Java API enhancements

- Scrollable cursor support
- Batched updates support
- Improved security for DB2 authentication
- Improved Java SQL error information
- Java API for set client information (SQLSETI)
- Native DB2 server SQL error messages
- Multiple open Stored Procedure results sets
- SAVEPOINT support
- Auto-generated keys
- Enhanced LOB support

NEW SQLJ preparation process



Tracing

- **Useful feature in the new Universal Driver**
  - Tracing can be turned on
    - in program
    - externally by setting up properties on a DataSource definition
    - JDBC connection URL
  - Tracing can now be turned on even for a running application (for which there is no source code)
    - IF the connection URL is specified externally to the program

Tracing

- **To turn on tracing programmatically**
  - Use setJccLogWriter() method of class DB2Connection.
  - First argument is a PrintWriter (where output is sent)
  - Second argument (optional) specifies the trace level
  - Constants representing these levels are declared in class com.ibm.db2.jcc.DB2BaseDataSource
  - The individual levels can can be combined using bitwise OR
  - If one-argument setJccLogWriter() method is used
    - TRACE_ALL is assumed

- **To turn on tracing using connection properties**
  - Tracing can be turned on outside the program IF the JDBC URL is not hard coded in the program

Tracing

To trace for example :

- Statement calls
- Result set meta data
- Parameter meta data

TRACE_STATEMENT_CALLS / TRACE_RESULT_SET_META_DATA / TRACE_PARAMETER_META_DATA

= 0x0002 / 0x0080 / 0X0100

= 0X0182

= 386

The URL is :

Jdbc:db2://your.server.name:port/SSID:traceFile=jcctrace.log:tracelevel=386

When the trace file contains colons : you have to enclose it in double quotes “

Jdbc:db2://your.server.name:port/SSID:traceFile=“jcctrace.log”:tracelevel=386

Tracing

Constant name	*Value*
TRACE_NONE	0x0000
TRACE_CONNECTION_CALLS	0x0001
TRACE_STATEMENT_CALLS	0x0002
TRACE_RESULT_SET_CALLS	0x0004
TRACE_DRIVER_CONFIGURATION	0x0010
TRACE_CONNECTS	0x0020
TRACE_DRDA_FLOWS	0x0040
TRACE_RESULT_SET_META_DATA	0x0080
TRACE_PARAMETER_META_DATA	0x0100
TRACE_DIAGNOSTICS	0x0200
TRACE_SQLJ	0x0400
TRACE_XA_CALLS	0x0800
TRACE_ALL	0xFFFF

Conclusion

Conclusion

- **If you're using JDBC heavily - then Dynamic SQL caching is essential**

- **Use SQLJ first** - if is not suitable ***THEN use JDBC***
  - Not the other way around !!!

- ***Java programmers need to be convinced to use SQLJ !!!***
  Its good for everyone … DB2, DBAs, Programmers

Conclusion

- **The key point is to minimise network traffic**
  - The more work the net has to do the more your performance will suffer

- **If you are truly internet-enabled, you have NO CONTROL over the network**

- Remember the **7 second rule**?

Bibliograpghy

- http://java.sun.com/
- http://java.sun.com/docs/books/tutorial/
- http://java.sun.com/products/ejb/
- http://www.ibm.com/developerworks/xml/newto/
- http://alphaworks.ibm.com/xml

- http://www.ibm.com/developerworks/xml/
- http://www.xml.org/
- http://www.xml.com/
- www.w3.org/TR/SOAP
- www.uddi.org

- WebSphere V5.1 Application Developer 5.1.1 Web Services Handbook (SG24-6891-01
- Squeezing the most out of dynamic SQL (SG24-6418-00)
- DB2 UDB e-business Guide (SG24-6539-00)
- DB2 for z/OS and OS/390: Ready for Java (SG24-6435-00)
- Design and Implement Servlets, JSPs and EJBs for IBM WebSphere Application Server (SG24-5754-00)
- Enterprise JavaBeans for z/OS and OS/390 WebSphere Application Server V4.0 (SG24-6283-00)
- Client/Server Survival Guide – 3rd Edition by Orfali, Harkey, Edwards
- Using XML on z/OS and OS/390 for Application Integration (SG24-6285-00)
- DB2 UDB for z/OS and OS/390 V7 Administration Guide
- DB2 UDB for z/OS and OS/390 V8 Administration Guide
- DB2 UDB for z/OS and OS/390 V7 Application Programming and Reference for Java
- DB2 UDB for z/OS and OS/390 V8 Application Programming and Reference for Java
- DB2 UDB for z/OS and OS/390 V7 Application Programming and SQL Guide
- DB2 UDB for z/OS and OS/390 V8 Application Programming and SQL Guide
- DB2 UDB for z/OS and OS/390 V7 XML Extender and Administration Programming

Computer Associates®

Java and the Wild Wild Web
Crash Course No. 2

Maria Sarikos
maria.sarikos@ca.com