MASSACHUSETTS INSTITUTE OF TECHNOLOGY
SLOAN SCHOOL OF MANAGEMENT

15.565 Integrating eSystems:

Technology, Strategy, and Organizational Factors

15.578 Global Information Systems:

Communications & Connectivity Among Information Systems

Spring 2002

Lecture 19

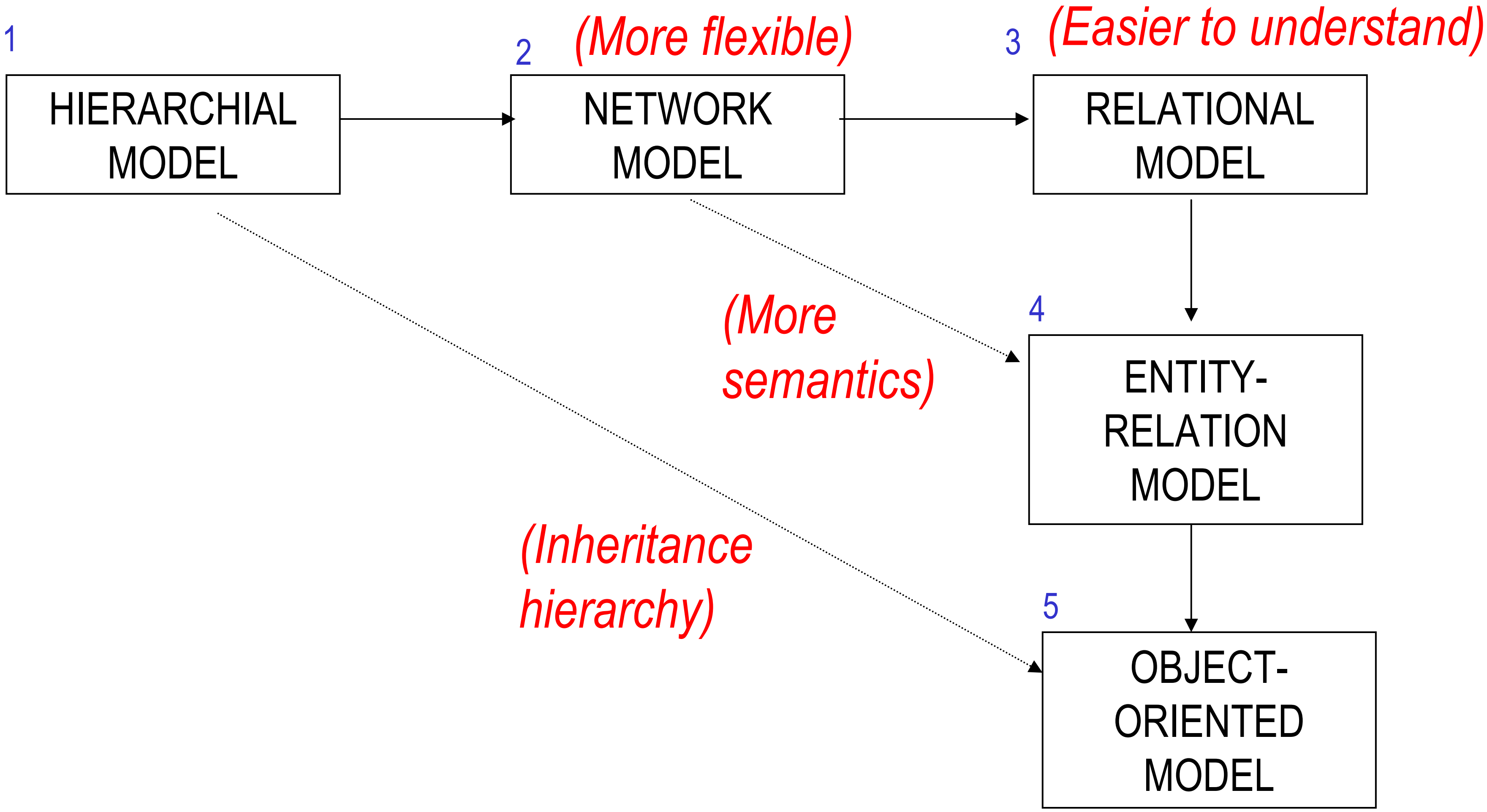
COMPONENT ARCHITECTURES & WEB SERVICES

OUTLINE

- PROGRESSION OF DATA MODELS
- OBJECT CONCEPT
- ADVANTAGES OF OBJECT-ORIENTED APPROACH
- OBJECT-ORIENTED DATABASE
- IMPLICATIONS FOR DATABASE INTEGRATION
- COMPONENT ARCHITECTURE (CORBA)
- JAVA AND .NET
- WEB SERVICES

PROGRESSION OF DATA MODELS

(“As the wheel turns”)



1 HIERARCHIAL MODEL

2 NETWORK MODEL

(More flexible)

3 RELATIONAL MODEL

(Easier to understand)

4 ENTITY-RELATION MODEL

(More semantics)

5 OBJECT-ORIENTED MODEL

(Inheritance hierarchy)

OBJECT CONCEPTS & TERMINOLOGY

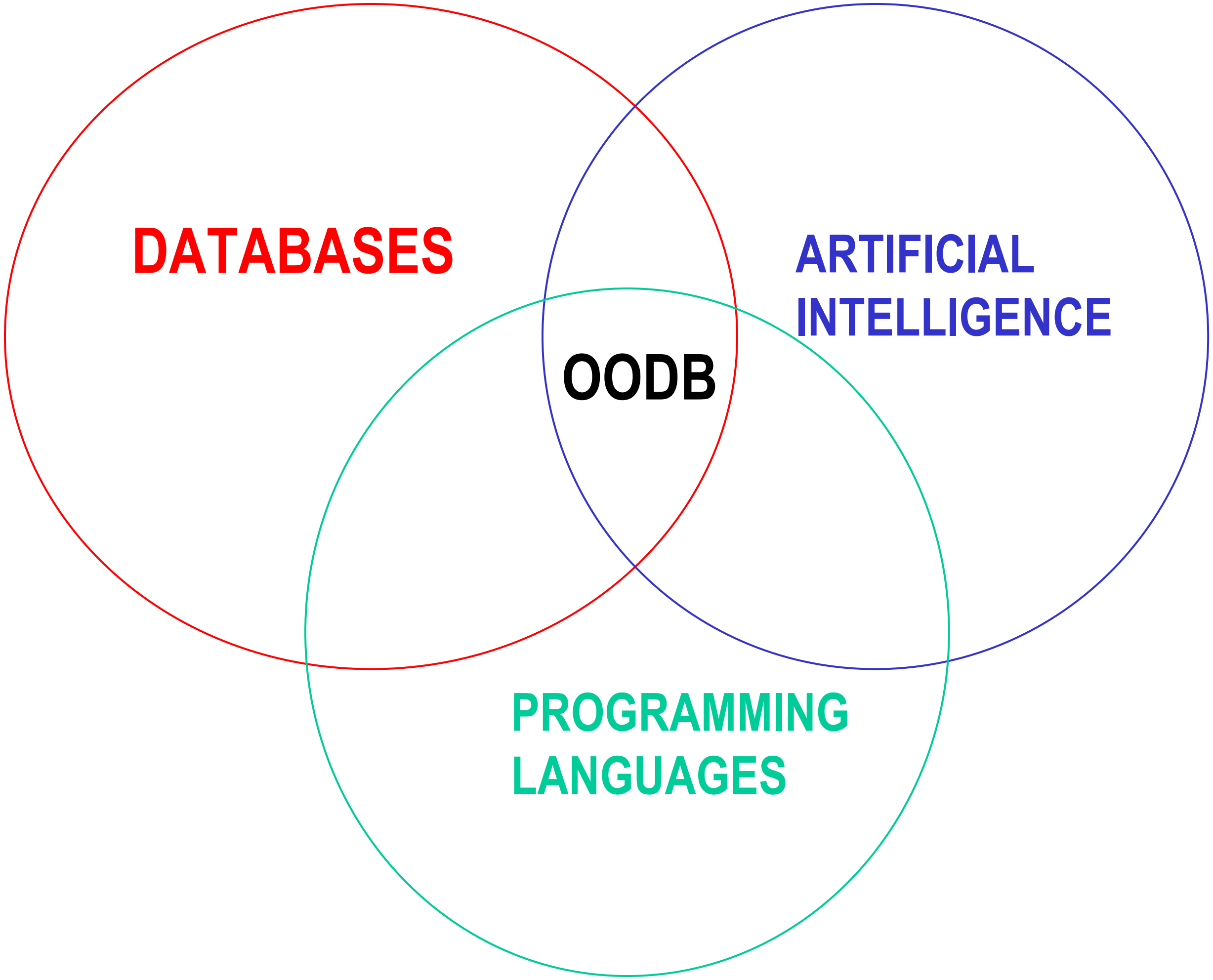
- EXAMPLES

OBJECT	PROPERTIES	OPERATIONS	RELATIONSHIPS
PIPE	- DIAMETER	- CONNECT	- IS CONNECTED TO
	- LENGTH	- DRAW	- IS PART OF
STOCK	- NAME	- BUY	- IS OWNED BY
	- PRICE	- SELL	- OF COMPANY

- OBJECT TYPE (CLASS) VS. OBJECT INSTANCE
- RELATE TO DATA BASE CONCEPTS?

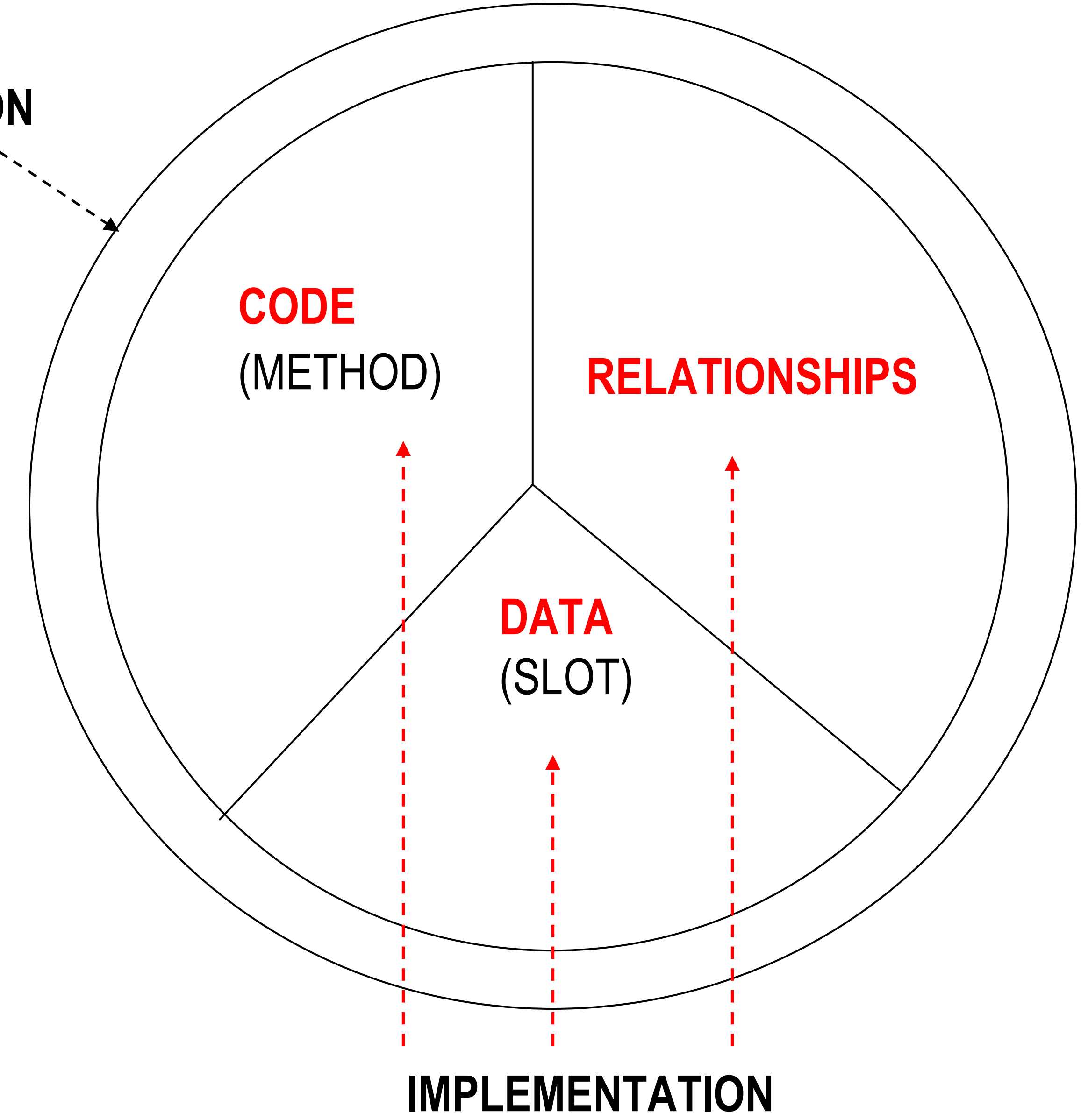
OBJECT	DBMS
Object Type	File / Table / Relation
Instance	Record / Row / Tuple
Property	Field / Column / Attribute
Operation	Program / Procedure
Relationship	Relationship (ER) / Join

SOURCES OF IDEAS



ABSTRACTION & ENCAPSULATION

SPECIFICATION

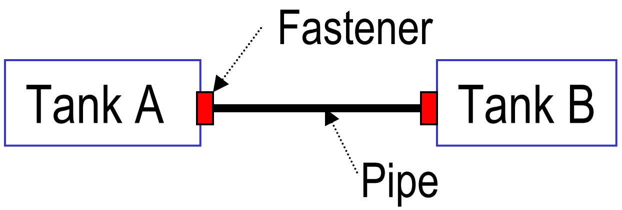


Example:
Object: Employee
Data: Name, Salary
Methods:
- Print paycheck
- Give raise
- Fire

Benefits:

RELATIONSHIPS & INHERITANCE

- AN INSTANCE OF (AIO) -- CLASSIFICATION
- A KIND OF (AKO) -- INHERITANCE
- A PART OF (APO) -- COMPONENT HIERARCHIES

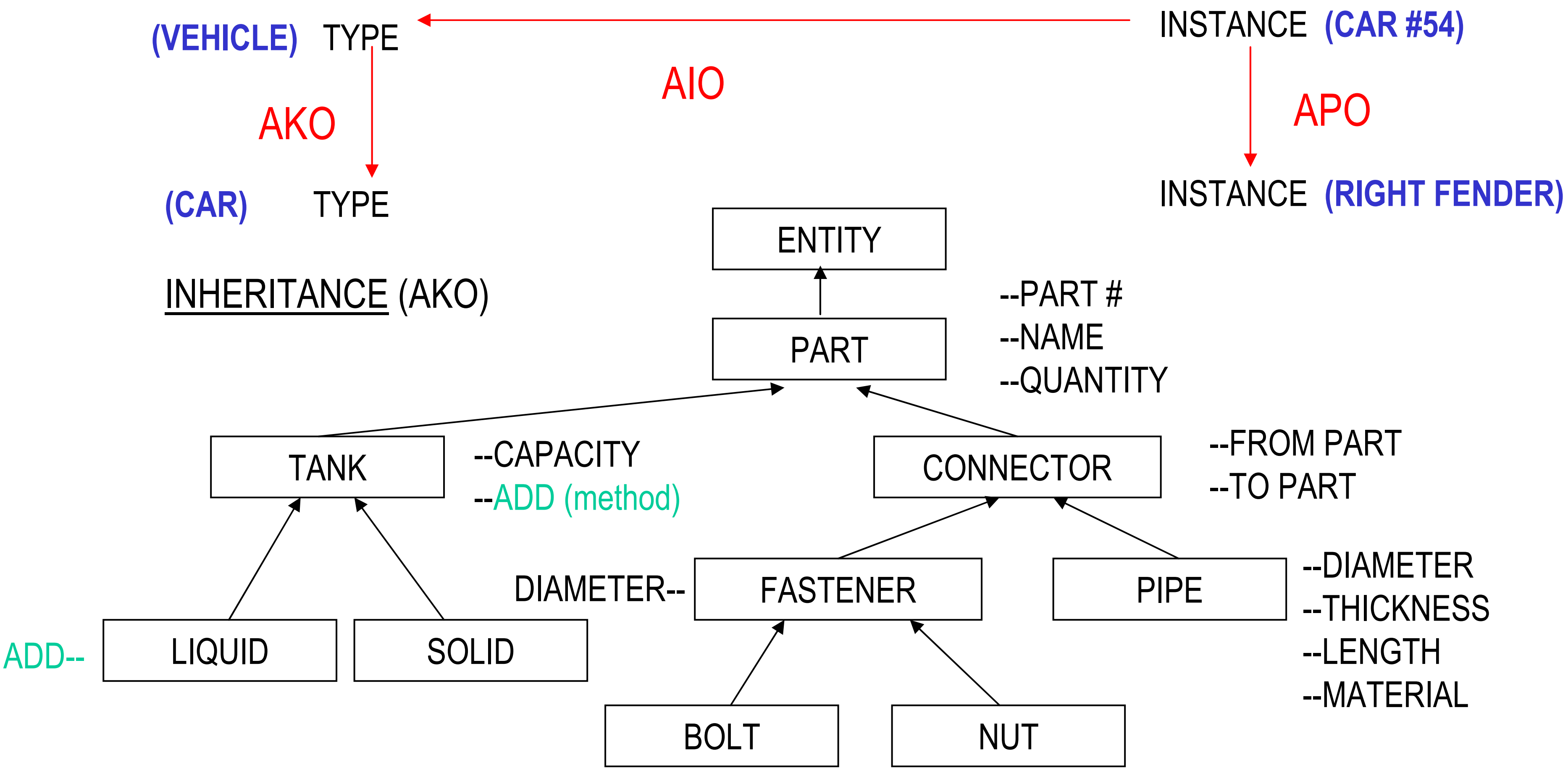


Fastener

Tank A

Tank B

Pipe



(VEHICLE) TYPE

AIO

INSTANCE (CAR #54)

AKO

APO

(CAR) TYPE

INSTANCE (RIGHT FENDER)

INHERITANCE (AKO)

ENTITY

PART

--PART #
--NAME
--QUANTITY

TANK

--CAPACITY
--ADD (method)

CONNECTOR

--FROM PART
--TO PART

ADD--

LIQUID

SOLID

DIAMETER--

FASTENER

PIPE

--DIAMETER
--THICKNESS
--LENGTH
--MATERIAL

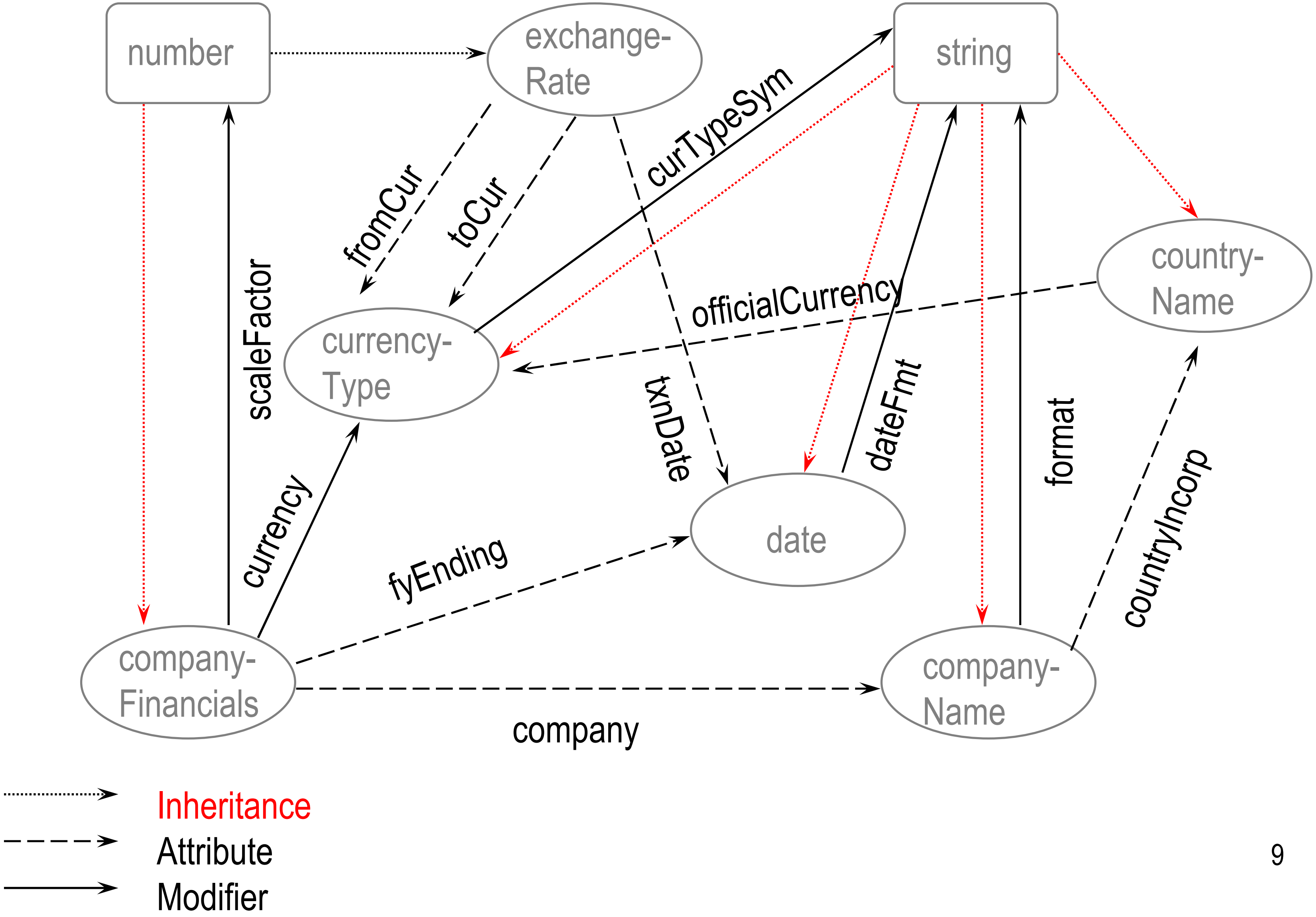
BOLT

NUT

ADVANTAGES OF OBJECT-ORIENTED APPROACH

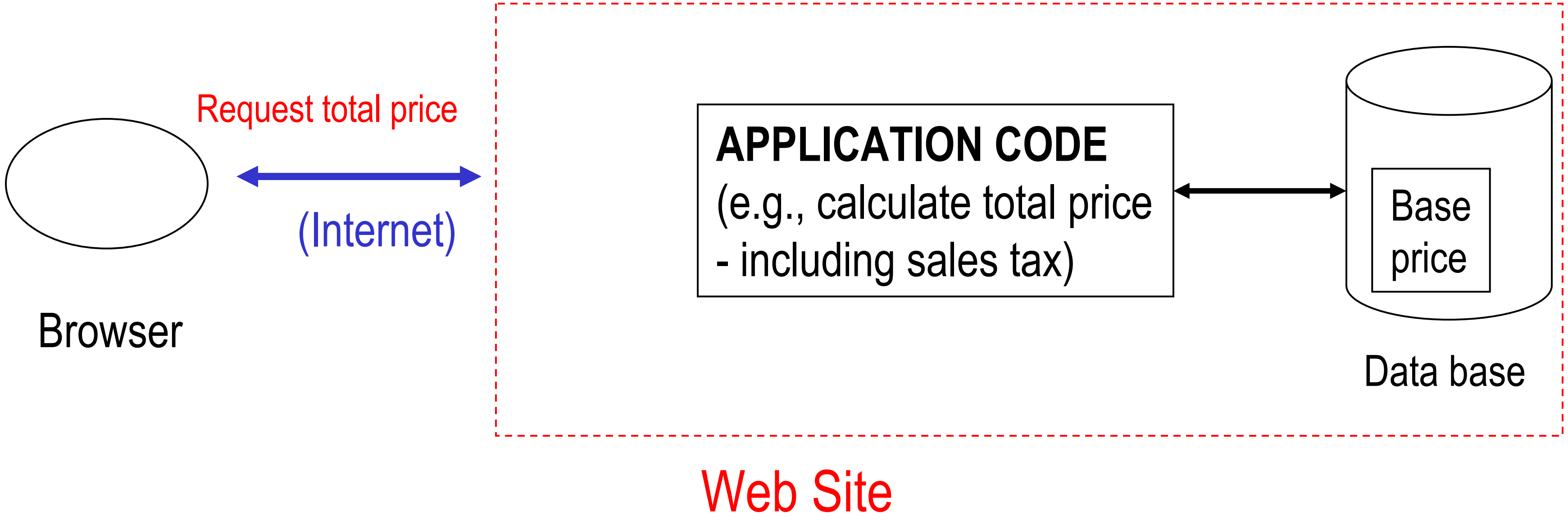
- EASIER DESIGN -- REFLECTS APPLICATIONS
- MODULARITY AND REUSABILITY
- INCREMENTAL REFINEMENT AND ABSTRACTION
  - ADD “PIPE” AS KIND OF “PART”
- MULTIPLE INHERITANCE -- e.g., A “TANKER TRUCK” IS
  - A KIND OF CONTAINER AND A KIND OF VEHICLE
- SUPPORT FOR MULTIPLE “VERSIONS” AND “ALTERNATIVES”
- PERFORMANCE TAILORING & AUTOMATIC METHOD SELECTION:
  - DIFFERENT REPRESENTATIVES FOR DIFFERENT TYPES
  - DIFFERENT REPRESENTATIVES FOR DIFFERENT INSTANCES OF THE SAME TYPE
  - SINGLE INSTANCE MAY HAVE TWO REPRESENTATIONS AT SAME TIME

EXAMPLE: Domain Model (from COIN)



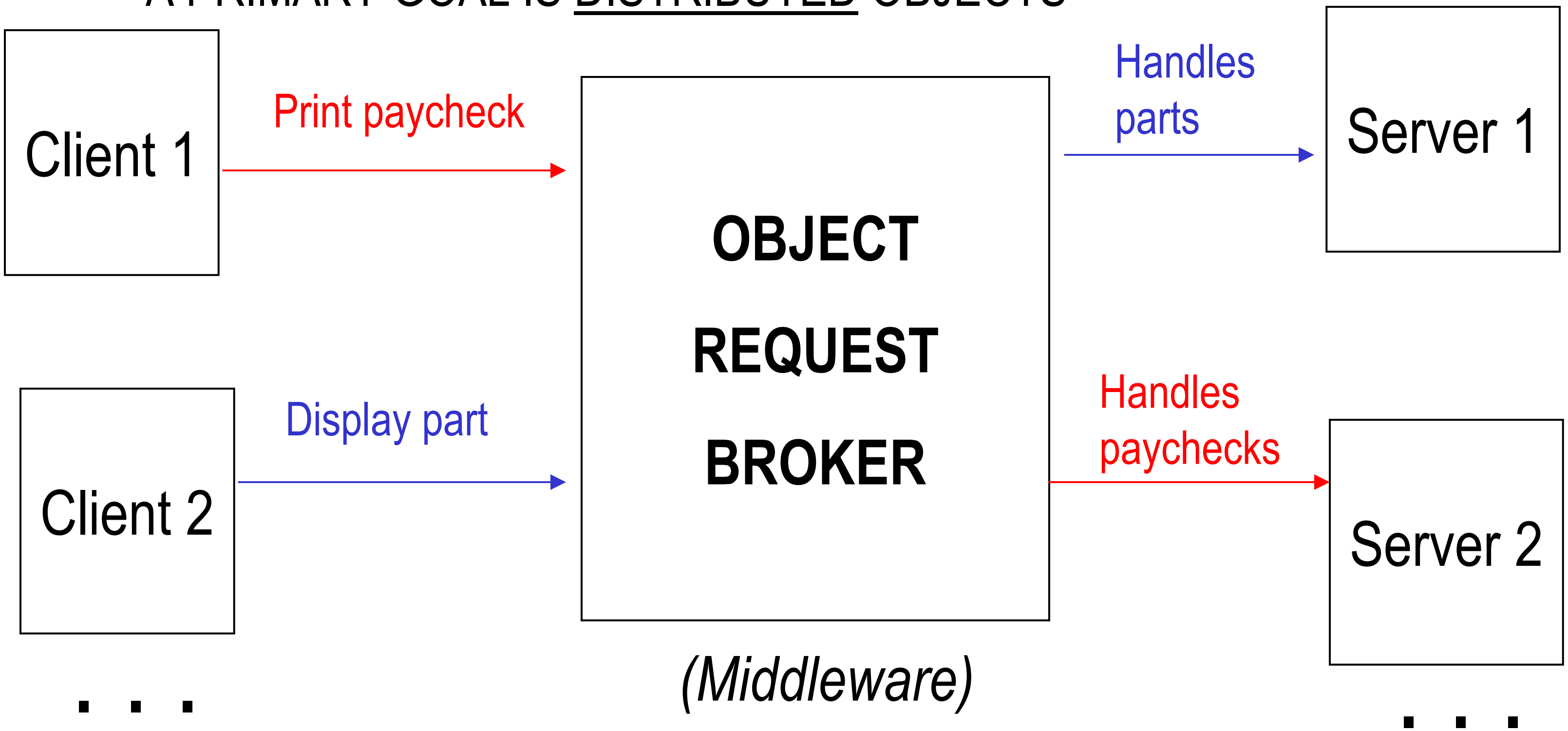
OBJECT-ORIENTED APPROACH TO DATABASE INTEGRATION

- **OBJECT OPERATIONS**
  - LIKE “PACKAGED” DATABASE TRANSACTION
  - INVENTORY EXAMPLES: RECEIVE PART, ISSUE PART
  - MORE FOCUSED AND APPLICATION-SPECIFIC THAN GENERAL PURPOSE QUERIES
- **SIMILAR TO “WEB WRAP” OF WEB SITES**



COMPONENT ARCHITECTURES

- OBJECT REQUEST BROKERS (**ORB**s)
- COMMON OBJECT REQUEST BROKER ARCHITECTURE (**CORBA**)
  - A PRIMARY GOAL IS DISTRIBUTED OBJECTS



- NOTE: A COMPUTER MAY BE BOTH A CLIENT AND A SERVER
- EXAMPLE APPLICATION: **UPS LOGISTICS** -- TO INTERFACE WITH DISPARATE CLIENTS' LOGISTIC SYSTEMS.
- ALSO ARCHITECTURE FOR DISTRIBUTED DBMS.

JAVA

- OBJECT ORIENTED
  - ABSTRACTION & ENCAPSULATION, INHERITENCE (AKO)
- CROSS PLATFORM
- … BUT NOT INHERENTLY DISTRIBUTABLE
  - INTEGRATING WITH NON-JAVA OBJECTS CAN BE DIFFICULT
  - JAVA WITH CORBA (RMI OVER IIOP) IS ONLY A PARTIAL SOLUTION

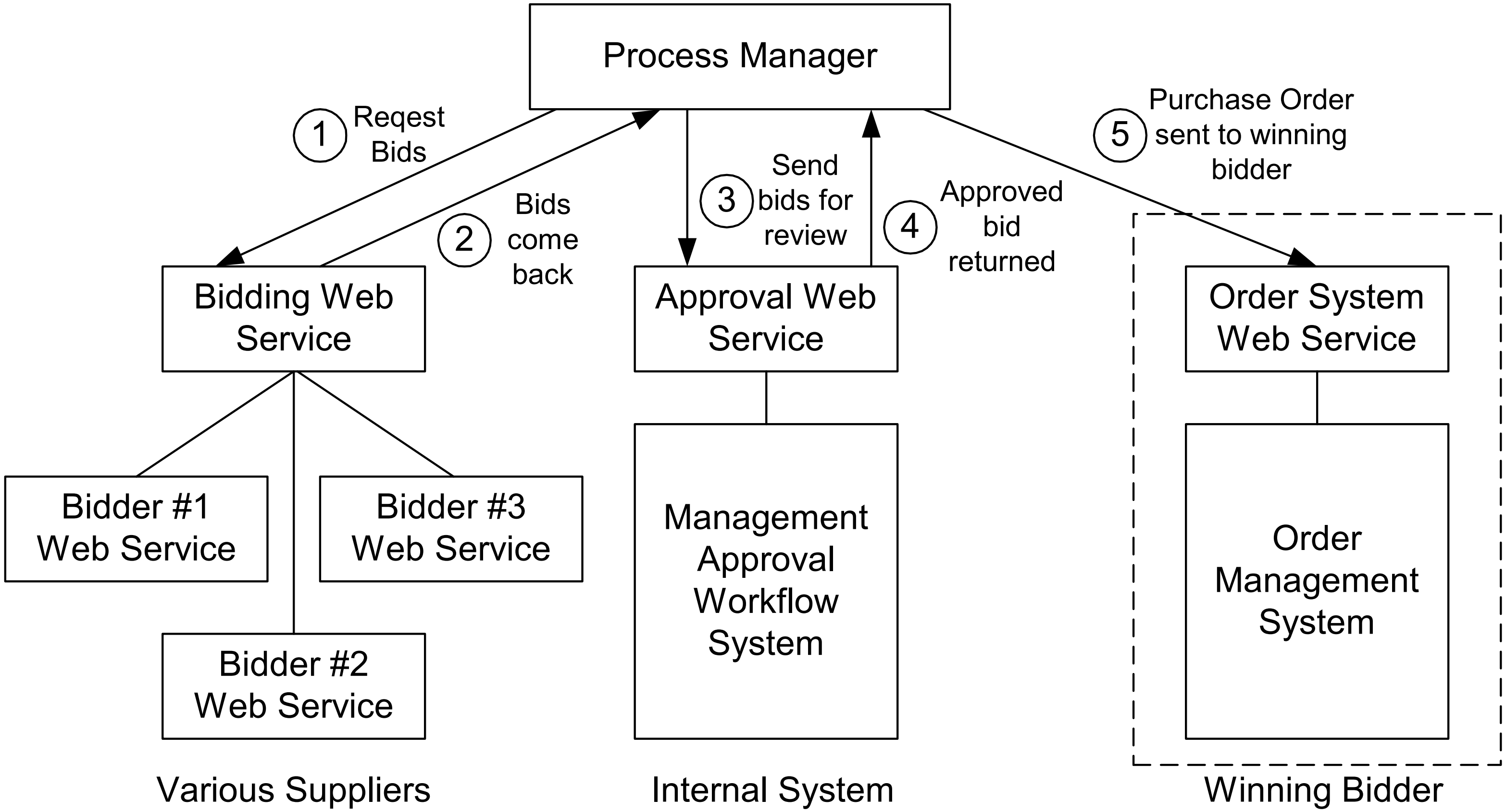
WHERE WE ARE TODAY

- CORBA’S POPULARITY IS FADING
  - ALTHOUGH IT REMAINS THE “GOLD STANDARD” FOR INTEGRATION, ITS ADMINISTRATION AND PROGRAMMING COMPLEXITIES HAVE MADE LARGE SCALE CORPORATE ADOPTION DIFFICULT
- JAVA’S ACHILLES HEEL IS SHOWING
  - INTEGRATION WITH LEGACY SYSTEMS IS NOT EASY
- MARKETPLACE IS LOOKING FOR A SIMPLE SOLUTION TO DISTRIBUTED COMPUTING CHALLENGES
  - 80/20 RULE – 20% OF CORBA EFFORT SOLVES 80% OF PROBLEMS
  - LEVERAGE INTERNET / INTRANET INFRASTRUCTURE

- … WEB SERVICES SEEMS TO FIT THE BILL

WEB SERVICES

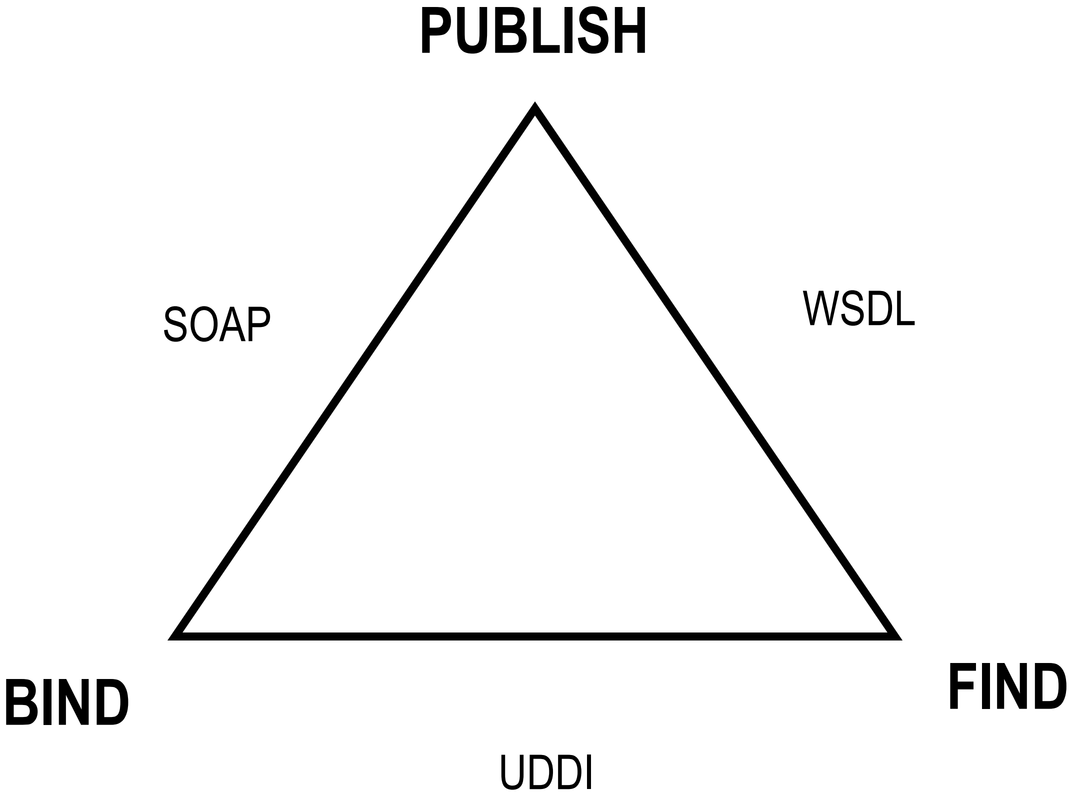
- SIMPLER THAN CORBA / "LOOSELY COUPLED"
- BETTER INTEROPERABILITY THAN JAVA
- DESIGNED FOR E-BUSINESS



WEB SERVICES STANDARDS

- XML (eXtensible Markup Language)
  - STANDARD DATA SYNTAX
  - USER DEFINED VOCABULARIES (NAME SPACES)
- SOAP (Simple Object Access Protocol)
  - WRITTEN IN XML
  - ENABLES COMMUNICATION BETWEEN OBJECTS ON ANY PLATFORM AND WRITTEN IN ANY LANGUAGE
- WSDL (Web Services Definition Language)
  - DESCRIBES WHAT A WEB SERVICE DOES AND HOW TO INTERACT WITH IT (WRITTEN IN XML)
- UDDI (Universal Description, Discovery, and Integration)

WEB SERVICES LIFECYCLE



PUBLISH

SOAP

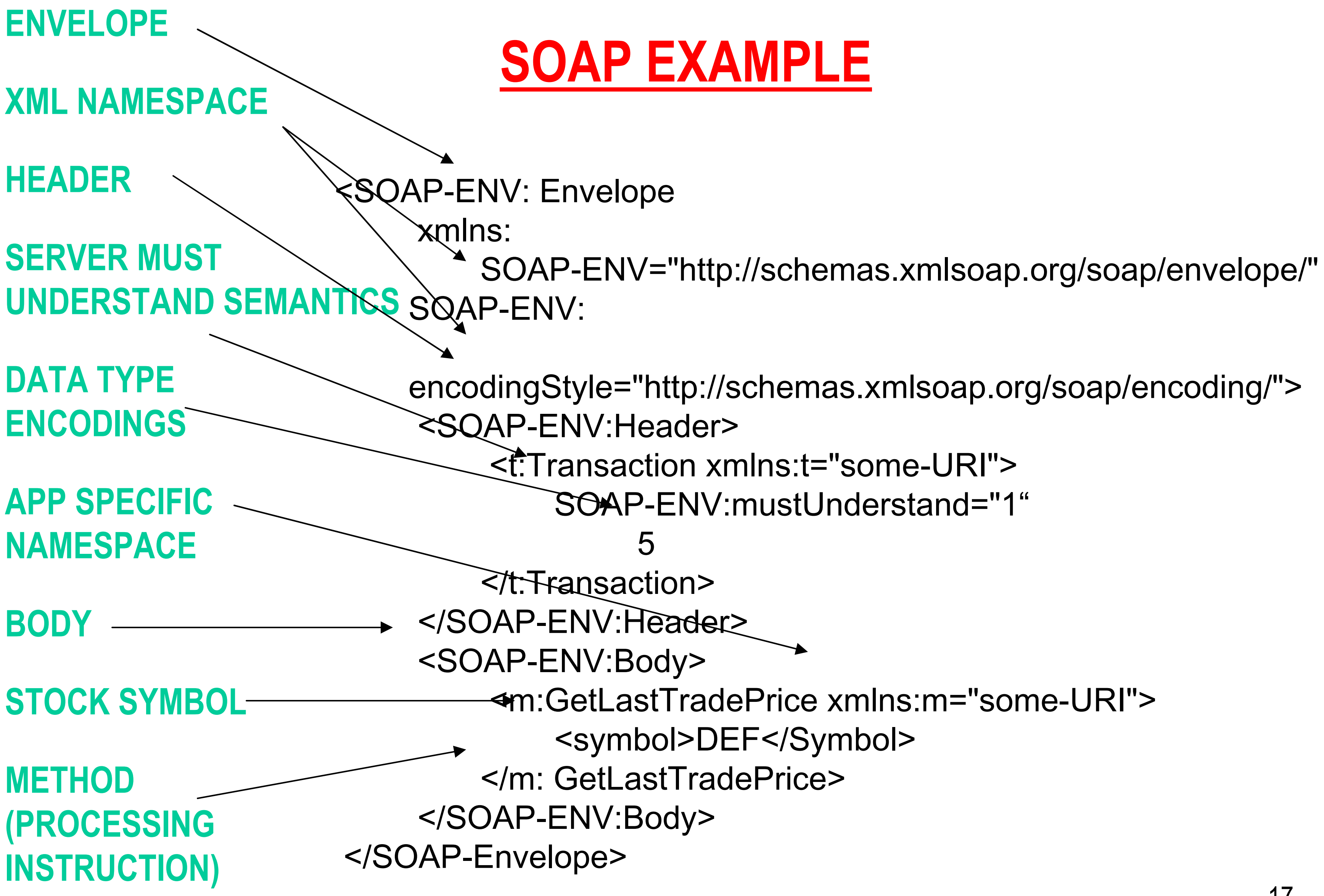
WSDL

BIND

FIND

UDDI

SOAP EXAMPLE



ENVELOPE

XML NAMESPACE

HEADER

SERVER MUST UNDERSTAND SEMANTICS

DATA TYPE ENCODINGS

APP SPECIFIC NAMESPACE

BODY

STOCK SYMBOL

METHOD (PROCESSING INSTRUCTION)

```
<SOAP-ENV: Envelope
   xmlns:
        SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:

  encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
        <t:Transaction xmlns:t="some-URI">
             SOAP-ENV:mustUnderstand="1“
                       5
        </t:Transaction>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
        <m:GetLastTradePrice xmlns:m="some-URI">
             <symbol>DEF</Symbol>
        </m: GetLastTradePrice>
   </SOAP-ENV:Body>
<SOAP-Envelope>
```

WEB SERVICES LIMITATIONS

- **SACRIFICE ROBUST DISTRIBUTED PROCESSING FOR SIMPLICITY**
  - DISTRIBUTED TRANSACTIONS / TWO-PHASE COMMIT
  - HIGH LATENCY
- **SEMANTIC CHALLENGES REMAIN**
  - STANDARD VOCABULARIES / DIFFERENT MEANINGS
  - E.G., STOCK PRICE, BUT AT WHAT TIME (CLOSE OF BUSINESS, AFTER HOURS, WHICH EXCHANGE?)
- **MODULARIZING BUSINESS PROCESSES**
  - BREAKING SAP INTO WEB SERVICES IS NON-TRIVIAL
- **QUALITY AND SOURCE SELECTION**
  - WHICH WEB SERVICE IS MOST ACCURATE ??

SUMMARY

- **OBJECT-ORIENTED APPROACH INCREASING**
  - ABSTRACTION, ENCAPSULATION, INHERITANCE, ETC.
  - CORBA, JAVA, J2EE, .NET, WEB SERVICES

- **MANY BENEFITS, INCLUDING MORE POWERFUL DATA SEMANTICS**
  - GENERALITY, FLEXIBILITY, EXTENSIBILITY

- **PROVIDES AN APPROACH TO DATABASE INTEGRATION**

- **MORE GENERALLY, PROVIDES BASIS FOR BUILDING COMPLEX SYSTEMS OUT OF “COMPONENTS”**
  - USING CORBA-TYPE ARCHITECTURE (J2EE, .NET)
  - USING A LOOSELY COUPLED APPROACH (WEB SERVICES)
  - BEING USED TO INTEGRATE MULTIPLE DISPARATE SYSTEMS