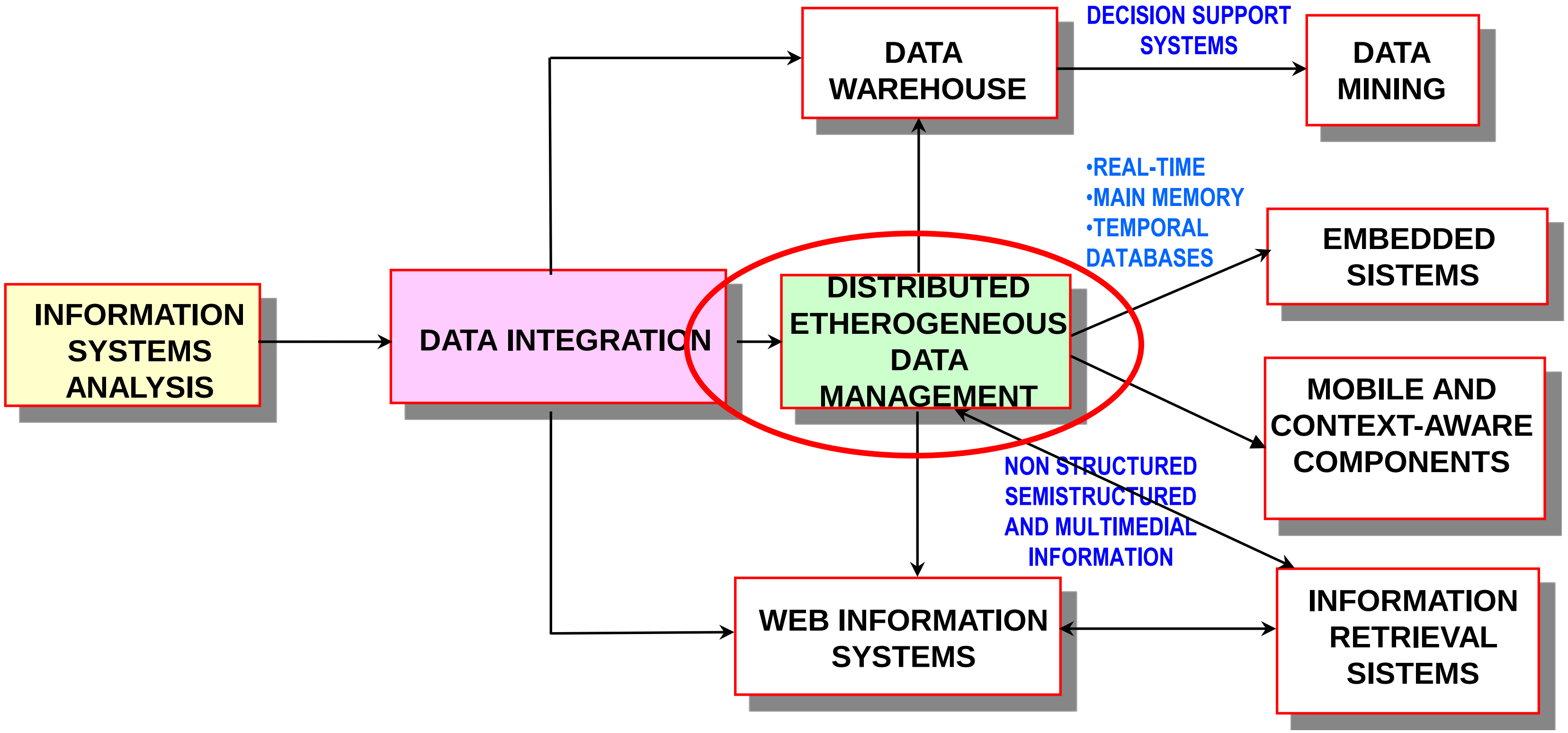
PERVASIVE DATA MANAGEMENT

DISTRIBUTED DATA MANAGEMENT

Prof. Fabio A. Schreiber
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

INFORMATION MANAGEMENT TECHNOLOGIES



COMPANY TELECOMMUNICATIONS

1st STAGE (< ‘80)

- TELEPHONE (POTS), TELEX, DATA TRANSMISSION **INDEPENDENT** ON SEPARATE NETWORKS

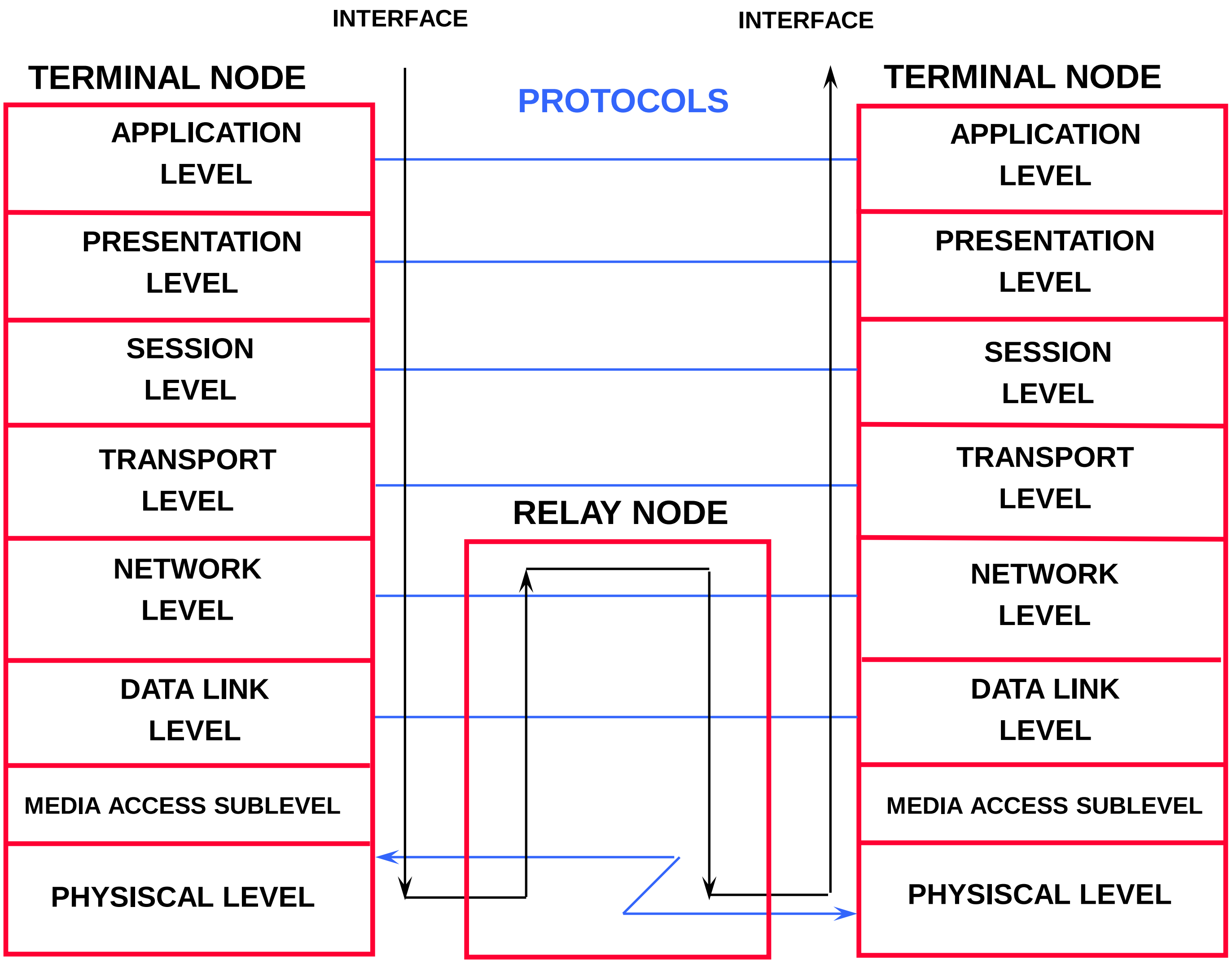
2nd STAGE (‘80 - ‘95)

- **DESIGN AND IMPLEMENTATION** OF LARGE DIGITAL COMMUNICATION NETWORKS
- DEVELOPMENT OF **LOCAL NETWORKS** OF PCs AND WORKSTATIONS

3rd STAGE (> 1995)

- **INTEGRATION AND MANAGEMENT** OF LARGE HETEROGENEOUS WANs AND OF LOCAL NETWORKS

ISO-OSI REFERENCE MODEL



DISTRIBUTED SYSTEMS APPLICATIONS

- **ACCESS TO REMOTE RESOURCES**

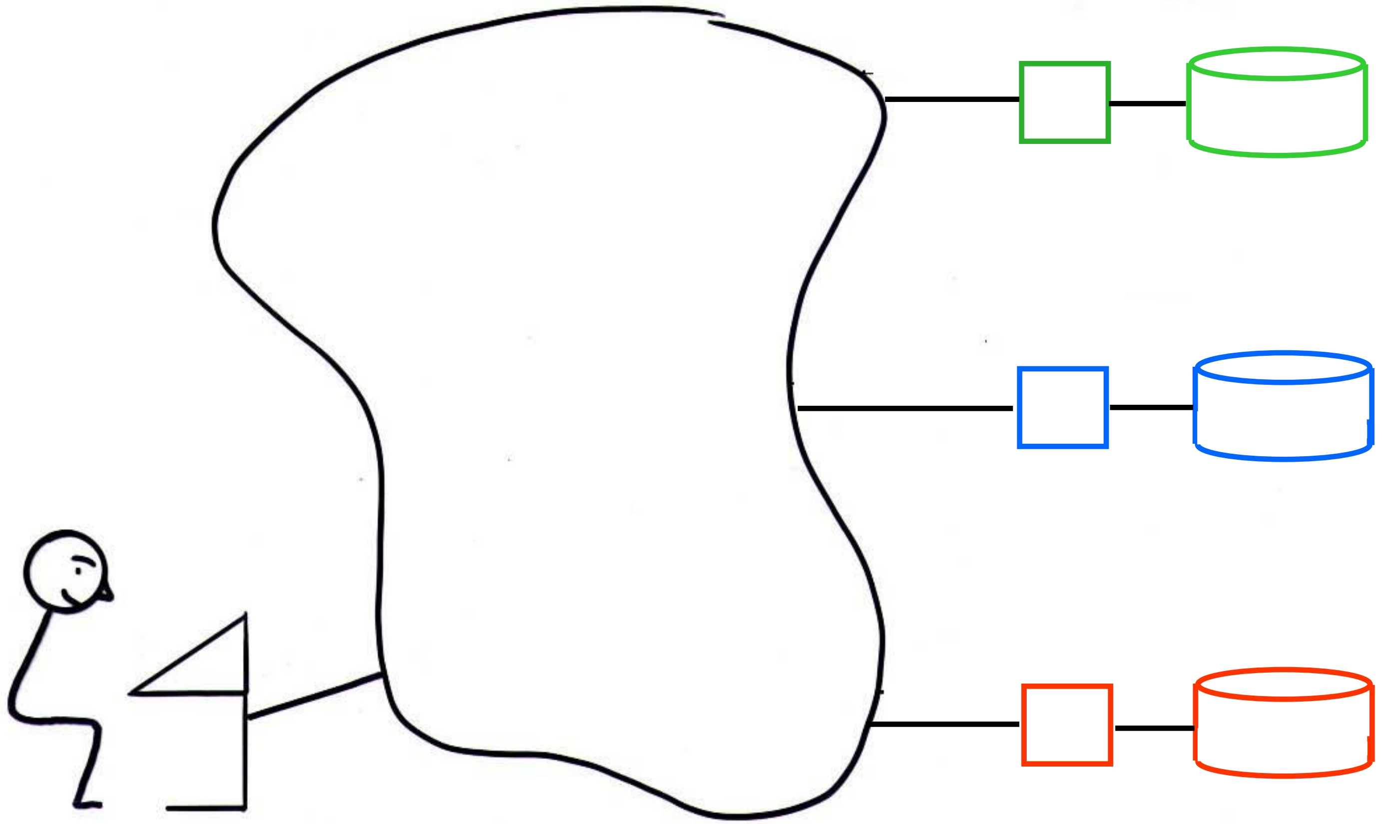
  connection to different computers and computing centers is made possible using the same terminal (e.g. TELNET, FTP, … PROTOCOLS)

- **DISTRIBUTED COMPUTING**

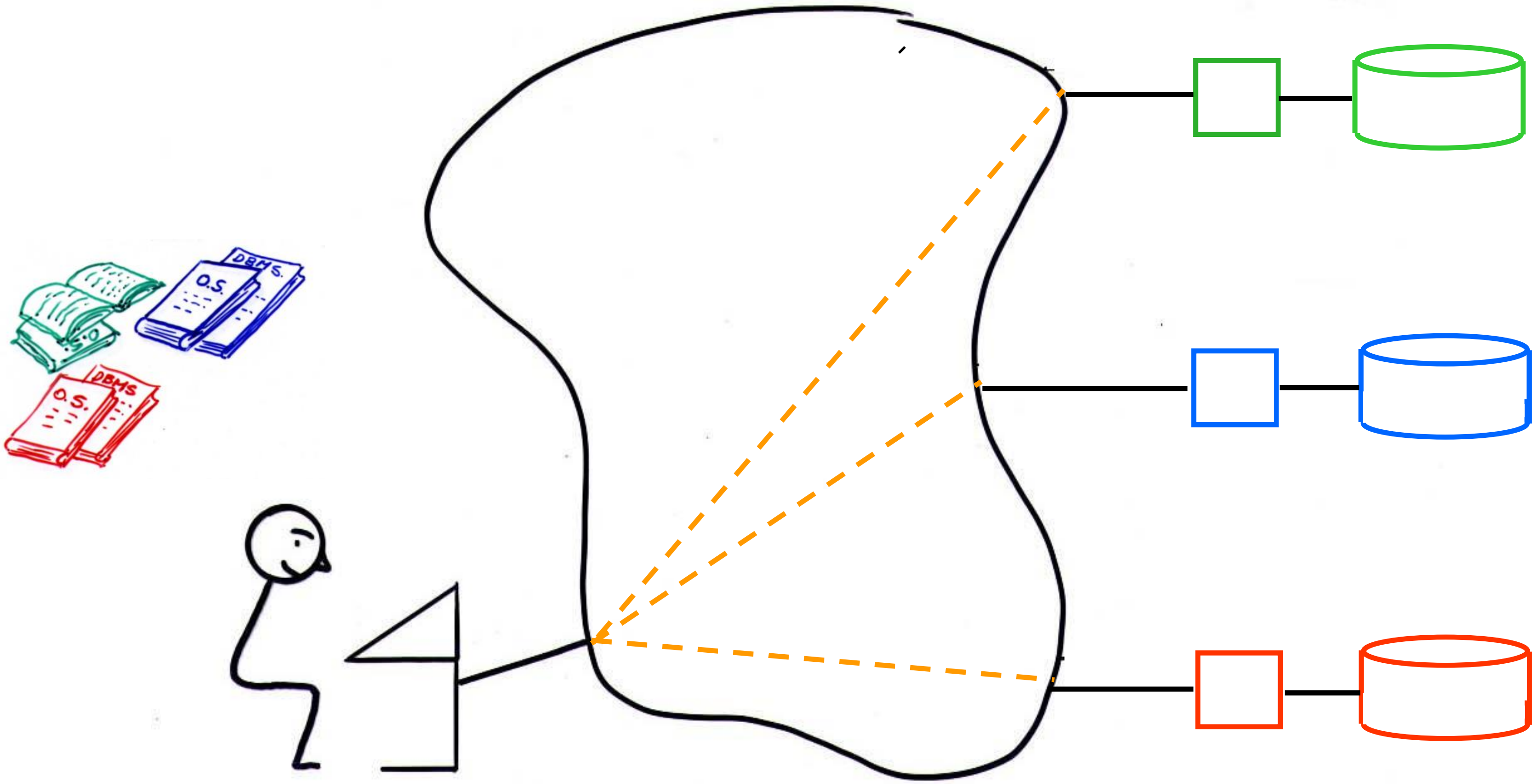
  complex systems are built in which the application process uses several remote computers and/or data sets, through telecommunication networks (e.g. distributed information systems, HPC, …)

- **TELEMATIC APPLICATIONS**
  - **electronic mail**
  - **teleconference**
  - **…….**

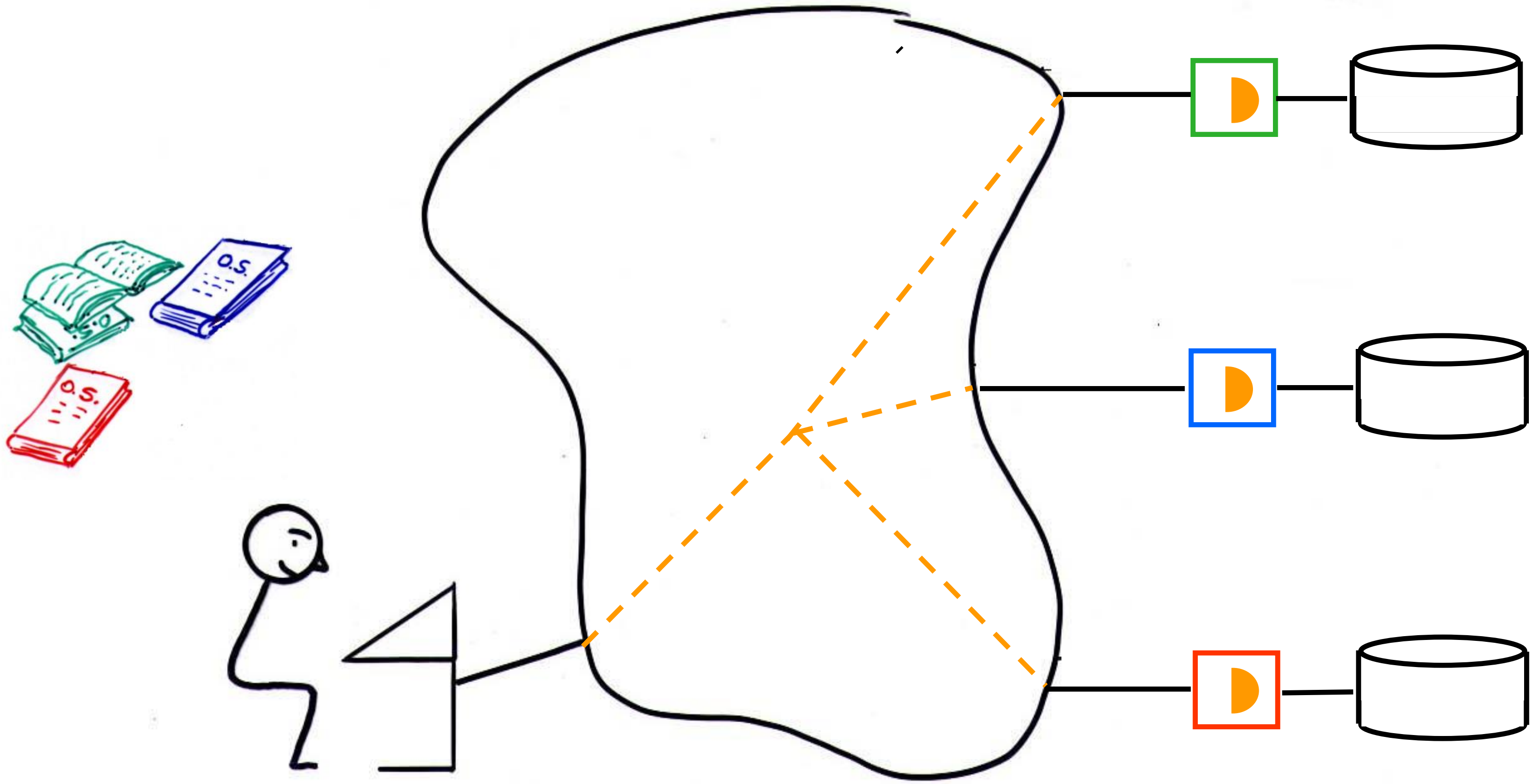
DATA MANAGEMENT ON THE NETWORK



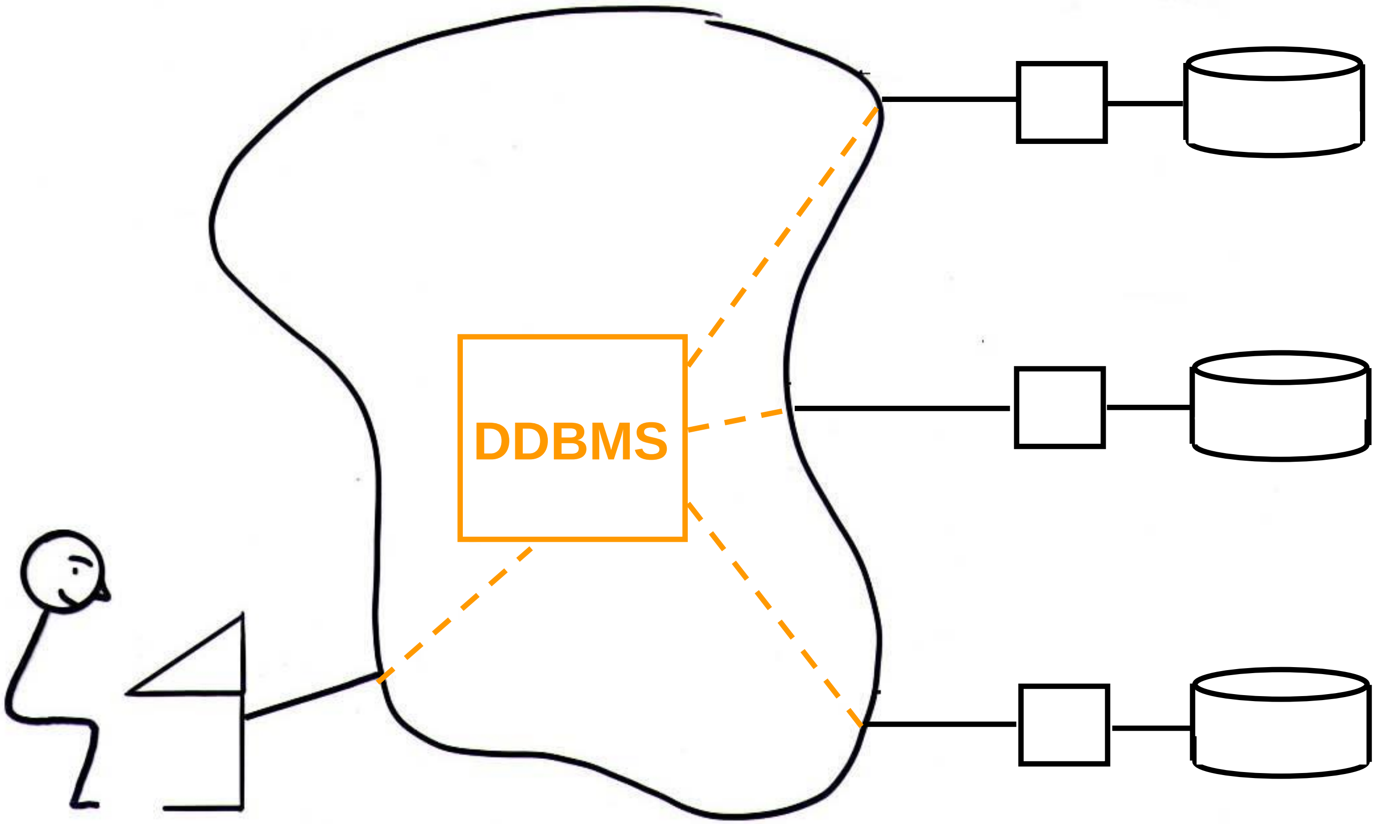
DATA MANAGEMENT ON THE NETWORK : INDEPENDENT DATA BASES



DATA MANAGEMENT ON THE NETWORK : COMMON COMMAND LANGUAGE



DATA MANAGEMENT ON THE NETWORK : DISTRIBUTED DATA BASE

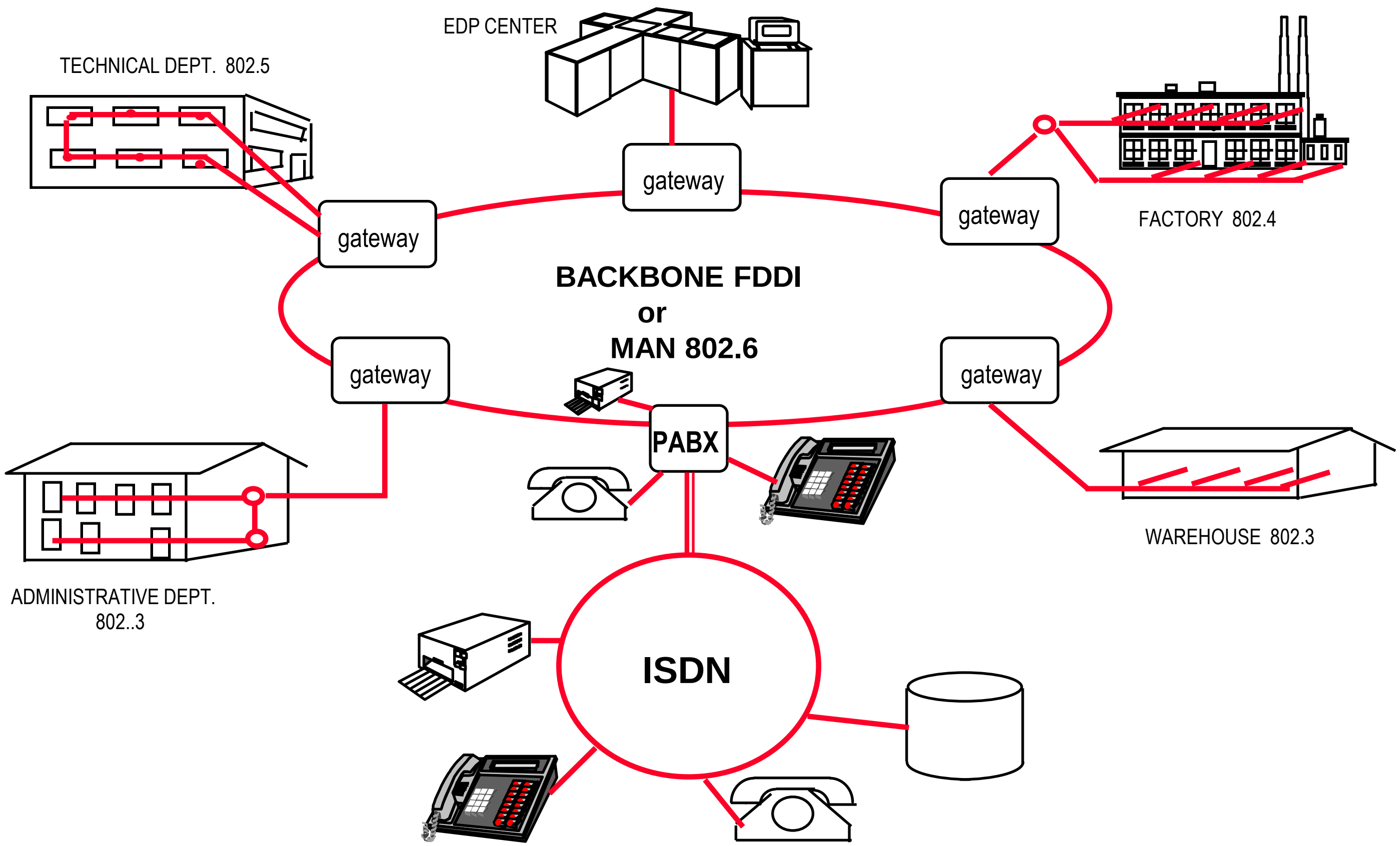


MODERN INFORMATION SYSTEMS

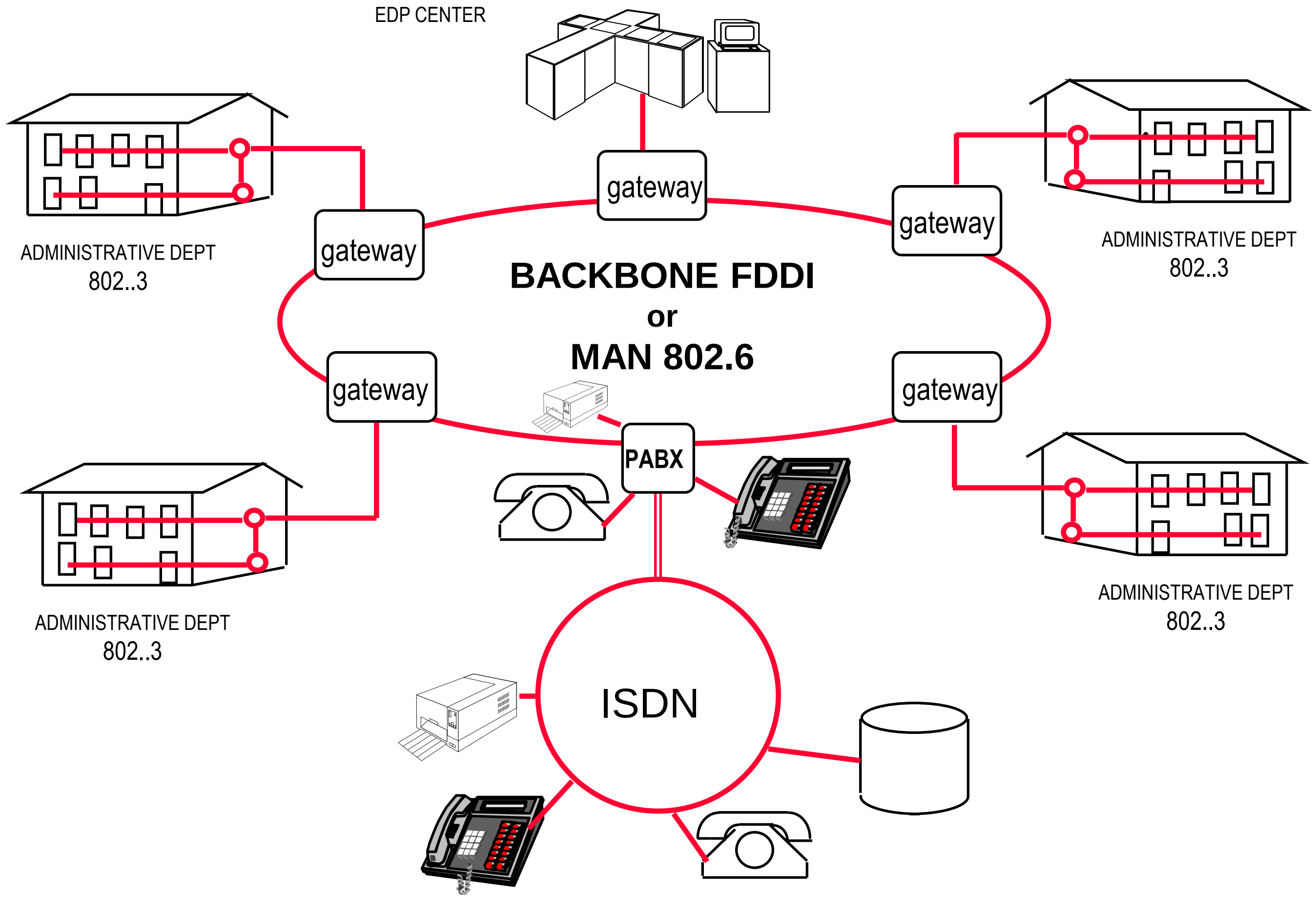
TELECOMMUNICATION NETWORKS BECOME AN ESSENTIAL COMPONENT FOR A GOOD ECONOMICAL AND FUNCTIONAL OPERATION OF THE ORGANIZATION

THE AVAILABILITY OF EFFECTIVE TELECOMMUNICATION SYSTEMS ALLOWS THE DEVELOPMENT OF NEW BUSINESS TYPES

VERTICAL (FUNCTION) PARTITIONING OF AN I.S.



HORIZONTAL (LOCATION) PARTITIONING OF AN I.S.



DATA MANAGEMENT

1st STAGE (< ‘70)

- SPARSE FILES

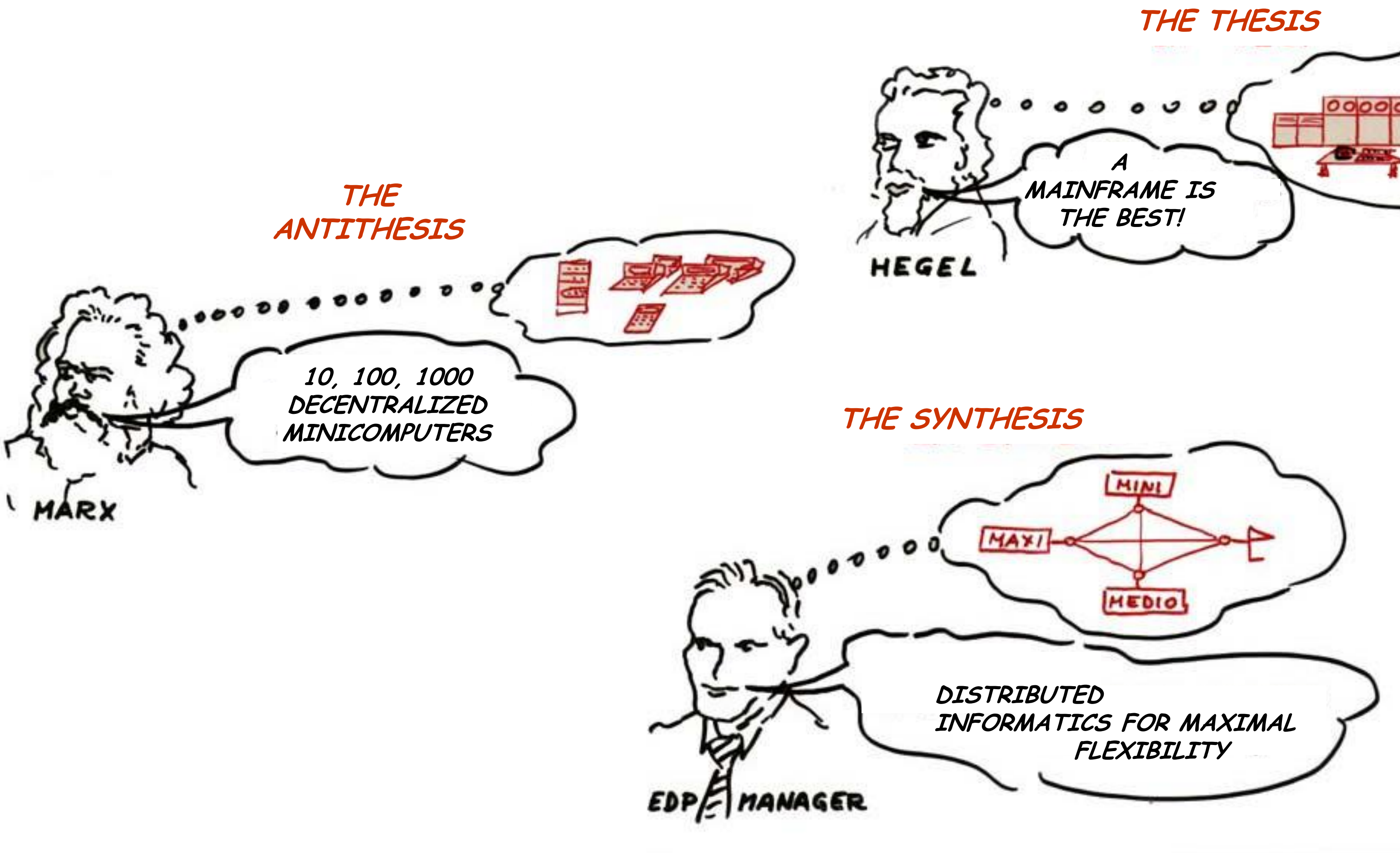
2nd STAGE (‘70 - ‘90)

- LARGE CENTRALIZED DATA BASES

3rd STAGE (> 1990)

- DISTRIBUTED DATA MANAGEMENT

SYSTEM ARCHITECTURE: A DIALECTIC PROCESS



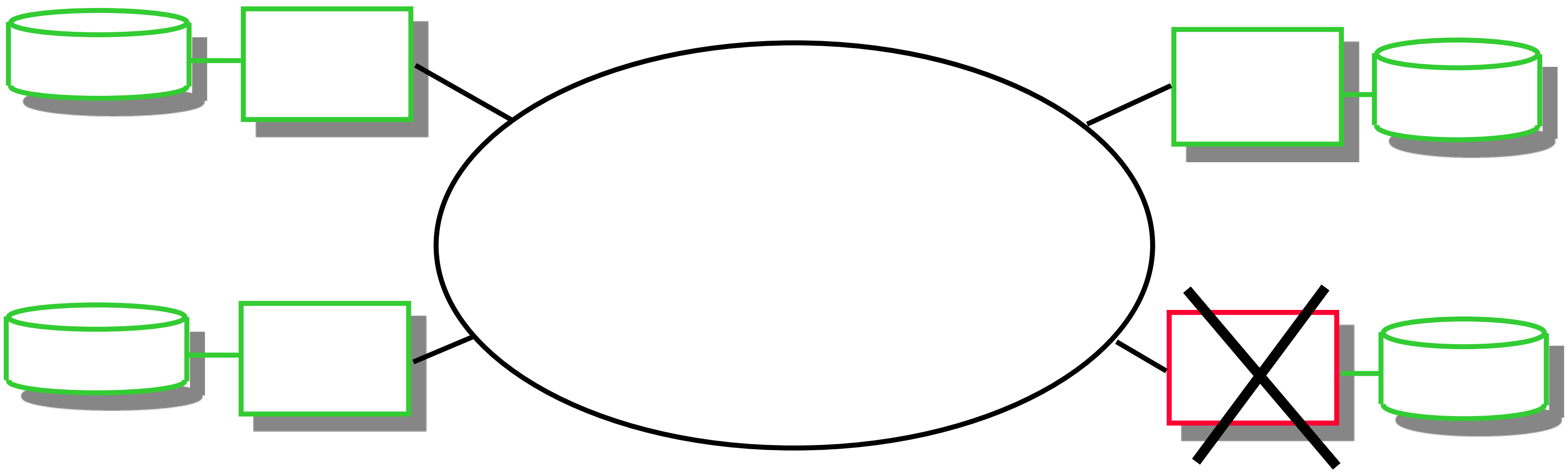
DISTRIBUTED DATA MANAGEMENT: FUNCTIONAL GOALS

- **AVAILABILITY**
- **LOAD SHARING**
- **RESOURCE SHARING**
- **QUALITY OF SERVICE TO THE USER**

FUNCTIONAL GOALS: AVAILABILITY

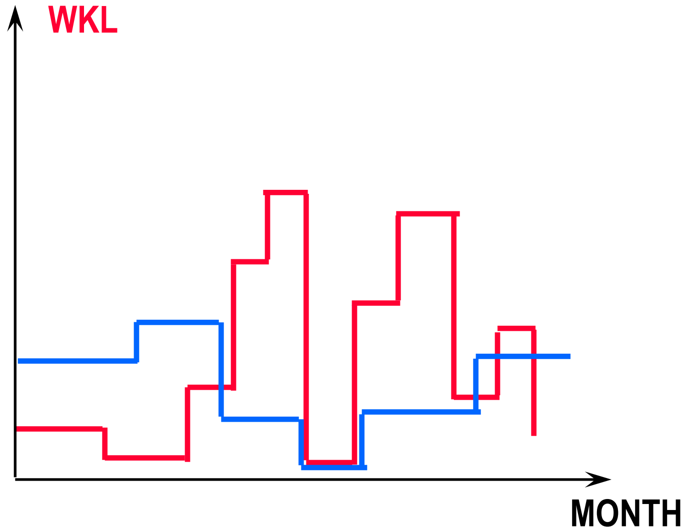
- REDUNDANT HW/SW RESOURCES CAN BE USED TO OBTAIN AN OVERALL SYSTEM HIGHER AVAILABILITY

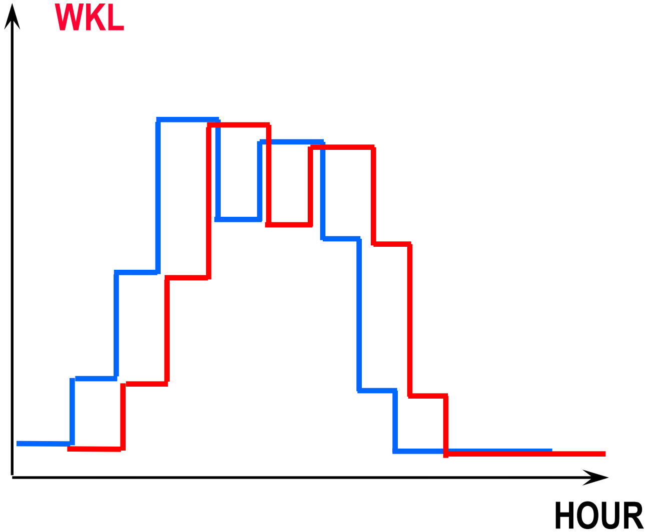
  - FAULT TOLERANT SYSTEMS
  - SOFT DEGRADATION SYSTEMS



FUNCTIONAL GOALS: LOAD SHARING

IT ALLOWS A BALANCED RESOURCES DEVELOPMENT





DISTRIBUTED DATA MANAGEMENT: FUNCTIONAL GOALS

- **RESOURCE SHARING**

  SPECIALIZED OR UNIQUE RESOURCES CAN BE SHARED AT WHICHEVER NODE

- **QUALITY OF SERVICE IMPROVEMENT**
  - **LOCAL PROCESSING CAPABILITIES**
  - **RESPONSE TIME REDUCTION**
  - **USER FRIENDLY INTERFACE**

DISTRIBUTED DATA MANAGEMENT: ECONOMICAL AND ORGANIZATIONAL FACTORS

- **THE PROS**
  - LOCAL SYSTEMS EFFECTIVENESS ALLOWS A TIGHTER CONNECTION BETWEEN USERS AND SYSTEM
  - DISTRIBUTION OF THE "POWER" (ORGANIZATIONAL, PSYCHOLOGICAL, POLITICAL, …) ASSOCIATED TO INFORMATION OWNING
  - ORGANIZATIONAL ACTIVITIES INTEGRATION IN GEOGRAPHICALLY DISTRIBUTED COMPANIES
  - COST/BENEFIT ???

- **THE CONS**
  - THE GENERAL COORDINATION AND COLLABORATION NEED REQUIRES A "CULTURAL ATTITUDE" NOT EASY TO FIND
  - COST/BENEFIT???

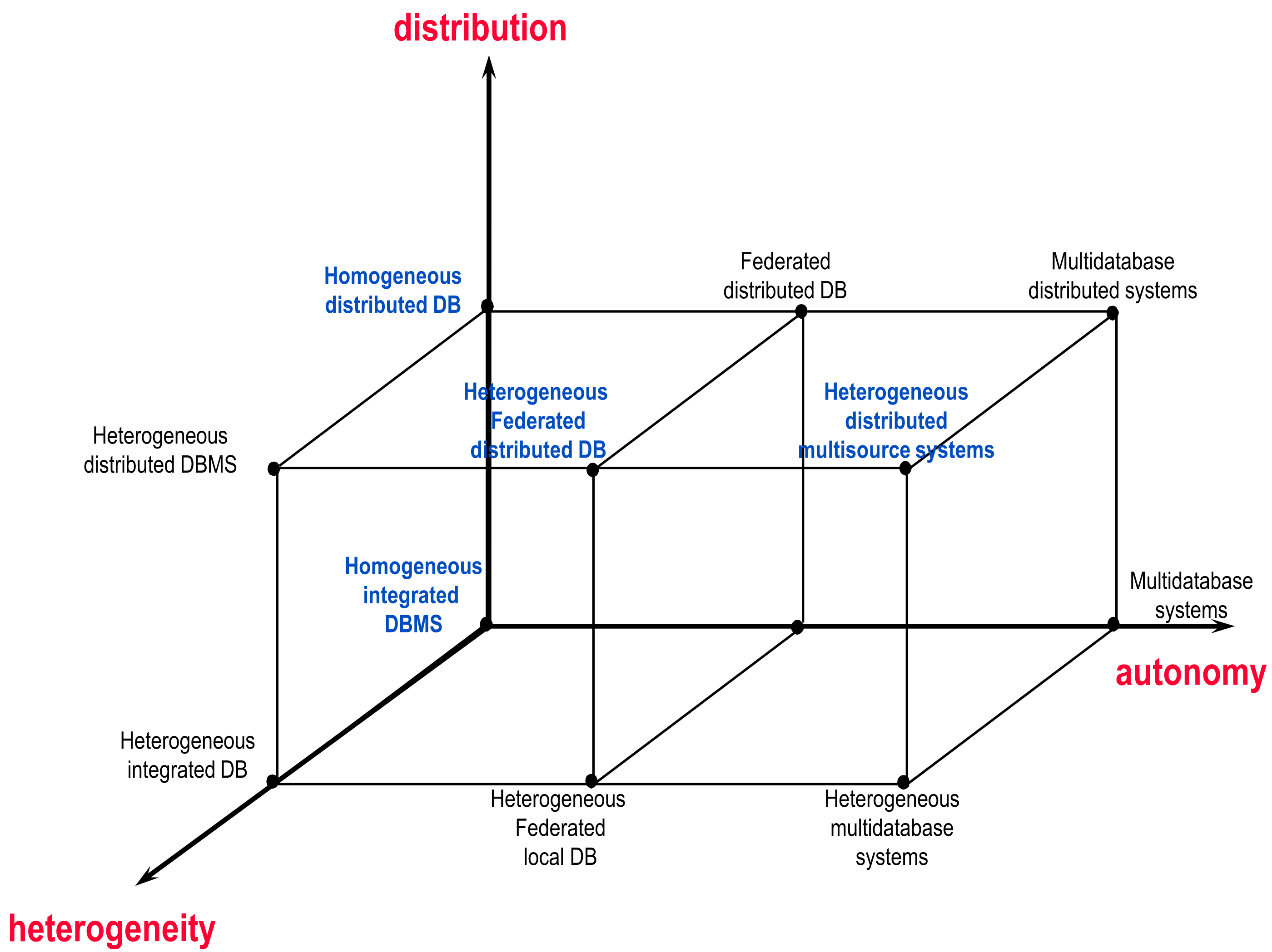
DISTRIBUTED SYSTEMS

SYSTEMS IN WHICH MESSAGE TRANSMISSION TIME IS NOT NEGLIGIBLE WITH RESPECT TO THE TIME BETWEEN TWO EVENTS IN A SINGLE PROCESS

DISTRIBUTED DATA MANAGEMENT: ARCHITECTURES

Hw/Sw ARCHITECTURE	DATA MANAGEMENT
narrow bandwidth loosely coupled systems geographically distributed computer networks	Distributed Data Base Management System (DDBMS)
wide bandwidth loosely coupled systems local networks, functionally distributed systems	back-end processor
wide bandwidth tightly coupled systems multiprocessor systems, associative memories, ...	database machine

DDSS SPACE
(DISTRIBUTED DATA SHARING SYSTEMS)



DDSS TAXONOMY

SYSTEM TYPE	THE GLOBAL SYSTEM HAS ACCES TO ...	TYPICAL LOCAL NODES ARE ...	GLOBAL DB FUNCTIONALITY	HOW GLOBAL INFORMATION IS DEALT WITH
DISTRIBUTED DATA BASE	DBMS INTERNAL FUNCTIONS	HOMOGENEOUS DATA BASES	Y	NAME SPACE AND GLOBAL SCHEMA
FEDERATED DATA BASE WITH A GLOBAL SCHEMA	DBMS USER INTERFACE	HETEROGENEOUS DATA BASES	Y	GLOBAL SCHEMA
FEDERATED DATA BASE	DBMS USER INTERFACE	HETEROGENEOUS DATA BASES	Y	PARTIAL GLOBAL SCHEMATA
MULTIDATABASE WITH A SYSTEM LANGUAGE	DBMS USER INTERFACE	HETEROGENEOUS DATA BASES	Y	ACCESS FUNCTIONS IN THE LANGUAGE
INTEROPERABLE SYSTEMS	APPLICATION PROGRAMS ABOVE DBMS	ANY DATA SOURCE SATISFYING THE INTERFACE PROTOCOL	N	DATA EXCHANGE

MEDIATORS

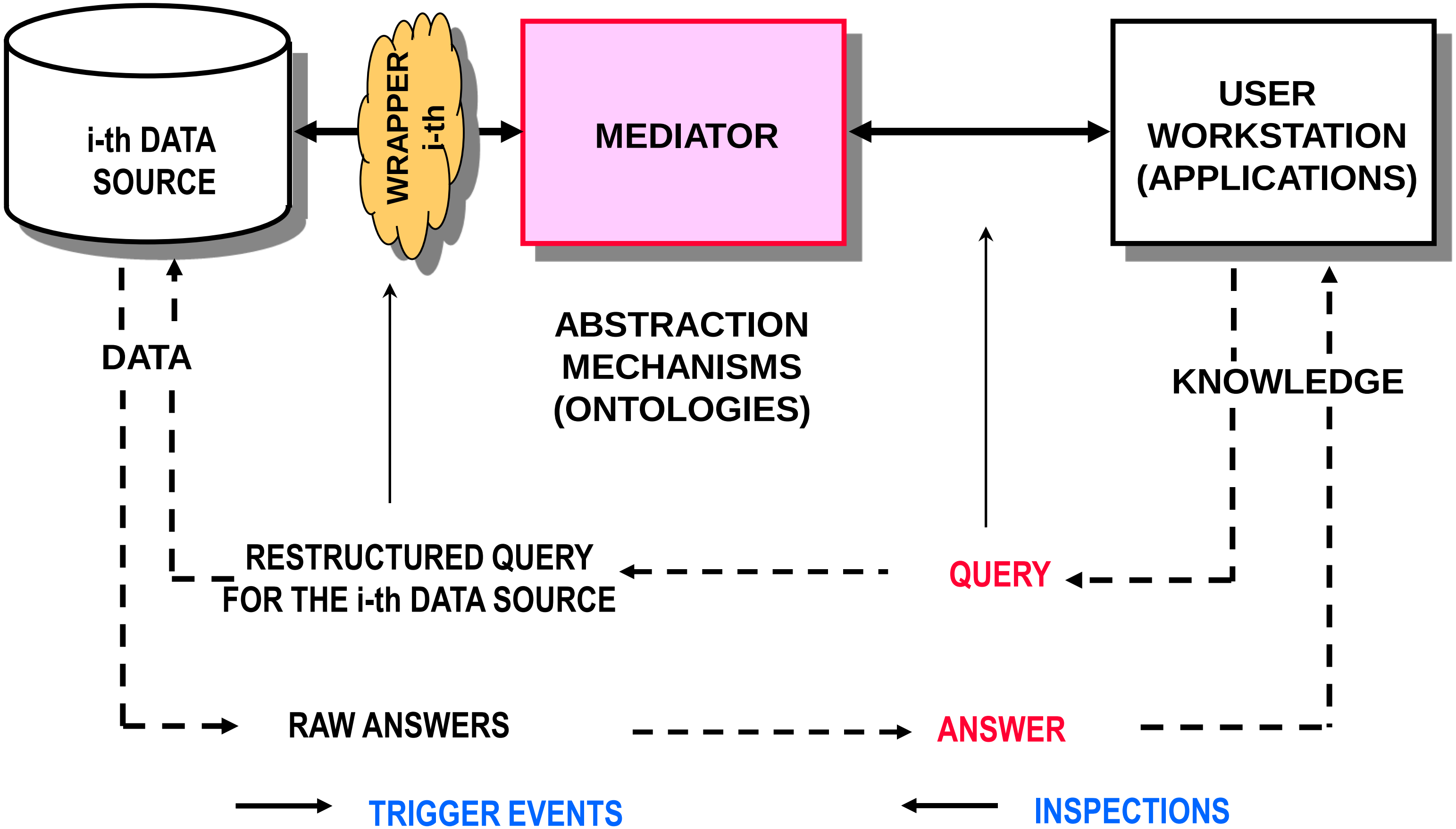
A MEDIATOR IS A SOFTWARE MODULE (AT ISO-OSI LEVEL 7) WHICH USES AN IMPLICIT KNOWLEDGE OF SOME DATA SETS OR SUBSETS TO CREATE KNOWLEDGE FOR A HIGHER APPLICATION LAYER (WIEDERHOLD)

ITS MAIN FUNCTION IS OBJECT FUSION

- TO GROUP INFORMATION ABOUT THE SAME ENTITY OF THE REAL WORLD
- TO REMOVE REDUNDANCIES AMONG DIFFERENT SOURCES
- TO SOLVE INCONSISTENCIES AMONG DIFFERENT SOURCES

A MEDIATOR WORKS AT QUERY EXECUTION TIME AND ITS ACTION IS NOT PROPAGATED (i. e., IT DOES NOT PRODUCE ANY PERMANENT DATA MODIFICATION)

MEDIATORS



i-th DATA SOURCE
WRAPPER i-th
MEDIATOR
USER WORKSTATION (APPLICATIONS)
ABSTRACTION MECHANISMS (ONTOLOGIES)
DATA
KNOWLEDGE
RESTRUCTURED QUERY FOR THE i-th DATA SOURCE
QUERY
RAW ANSWERS
ANSWER
TRIGGER EVENTS
INSPECTIONS

WRAPPERS

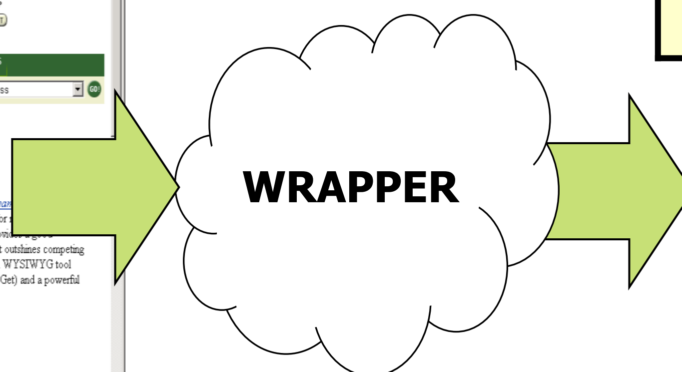
TRANSLATE QUERIES IN ONE OR MORE COMMANDS/ QUERIES UNDERSTANDABLE BY THE SPECIFIC SOURCE

THEY CAN EXTEND THE QUERYING POWER OF A SPECIFIC SOURCE

THEY CONVERT NATIVE FORMAT RESULTS TO A FORMAT UNDERSTANDABLE BY THE APPLICATION

THEIR WRITING INVOLVES A LARGE IMPLEMENTATION EFFORT, BUT SOME TOOLS CAN EASE THE TASK (e.g. THE TSIMMIS TOOLKIT)

WRAPPER



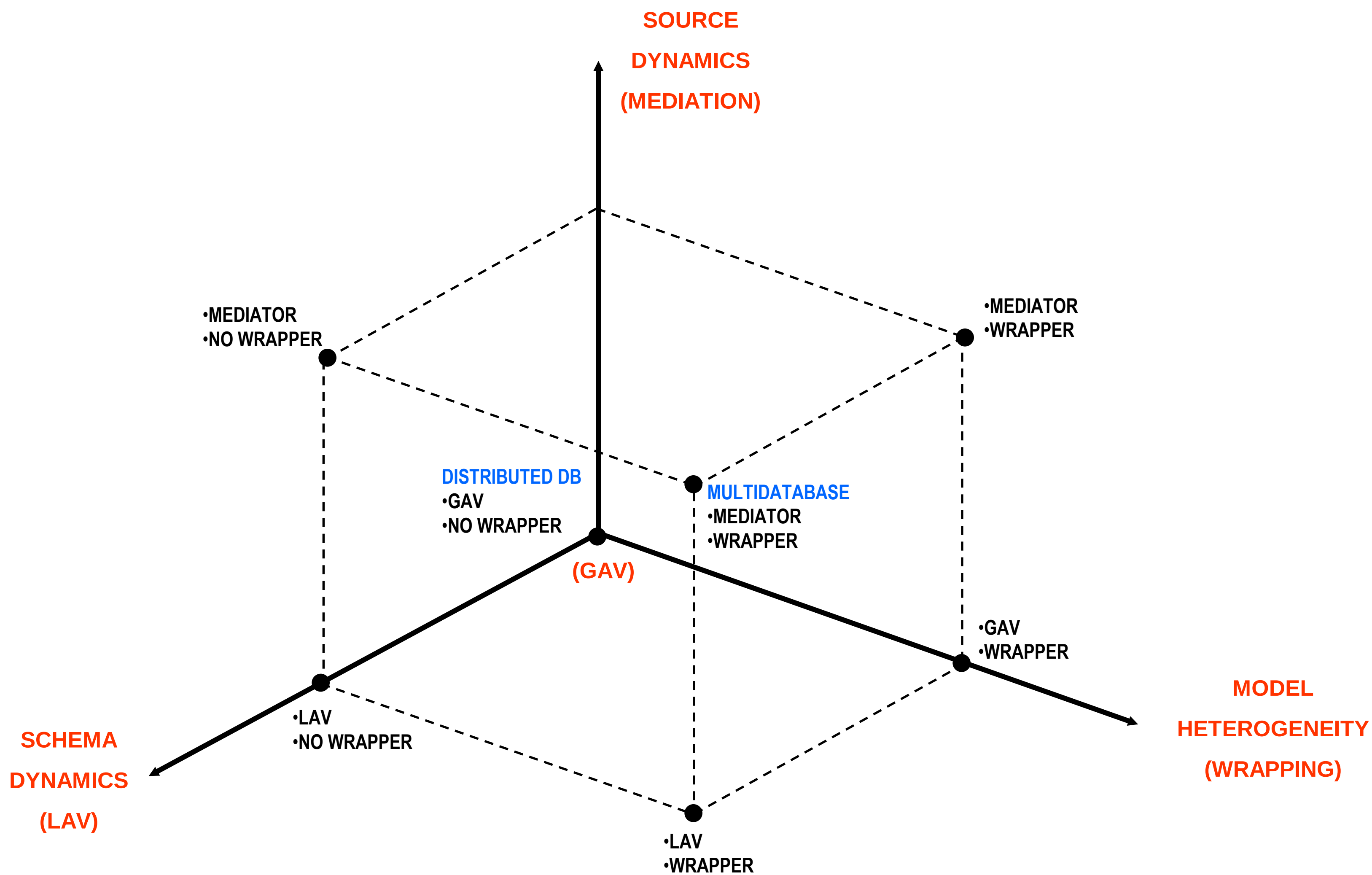
BookTitle	Author	Editor
The HTML Sourcebook	**J. Graham**	**...**
Computer Networks	**A. Tannenbaum**	**...**
Database Systems	**R. Elmasri, S. Navathe**	**...**
Data on the Web	**S. Abiteboul, P.Buneman, D.Suciu**	**...**

HTML PAGE

DATABASE TABLES (OR XML DOCUMENTS)

Source: G. Mecca EDBT 2002

DATA INTEGRATION SPACE



DATA INTEGRATION IN DDSS (1)
CENTRALISED SYSTEM

CONCEPTUAL SCHEMA DESIGN

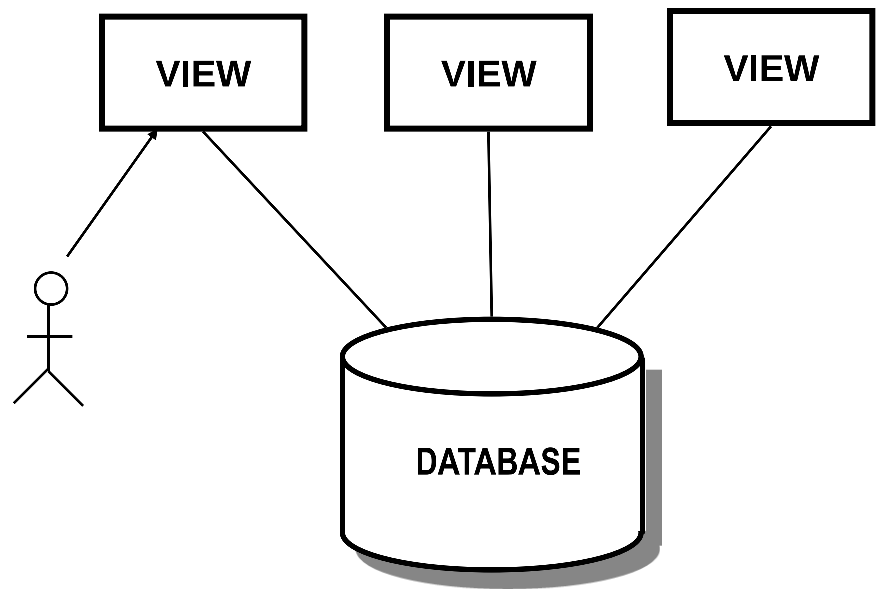
• TOP-DOWN, BOTTOM-UP OR MIXED

DATA DISTRIBUTION DESIGN

• NO DISTRIBUTION

DATA INTEGRATION APPROACH

• VIEW INTEGRATION



DATA INTEGRATION IN DDSS (2)
DISTRIBUTED DATABASE

CONCEPTUAL SCHEMA DESIGN

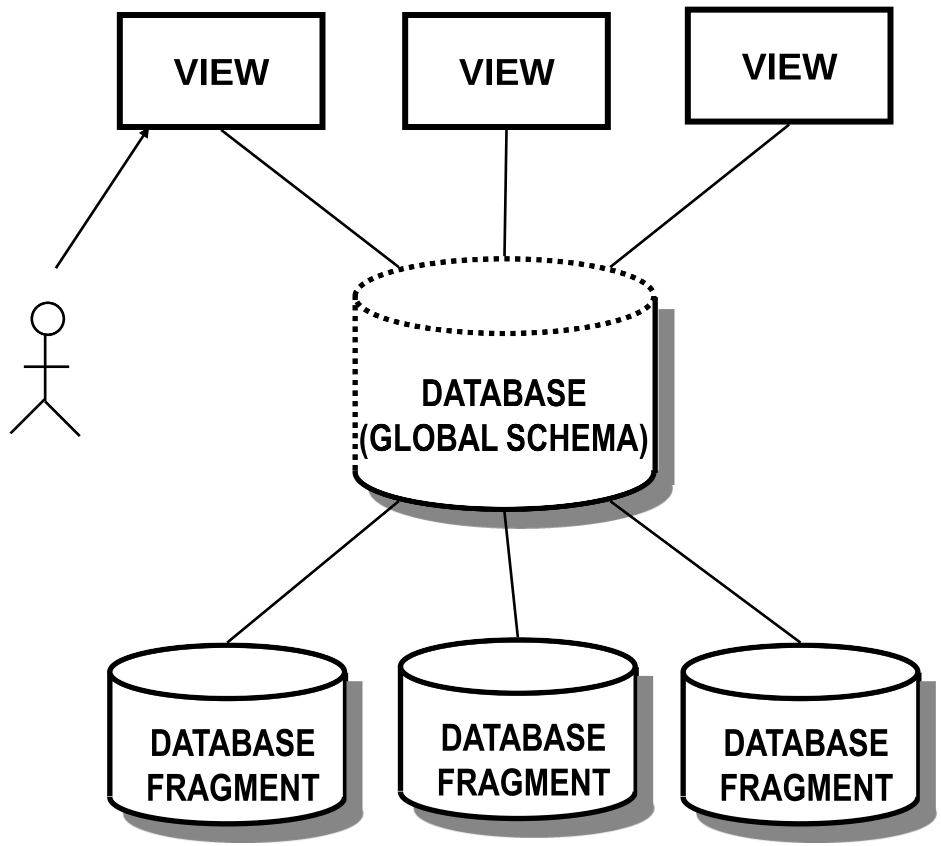
- TOP-DOWN (INTEGRATED)

DATA DISTRIBUTION DESIGN

- TOP-DOWN
- POSSIBLY EXISTING INFORMATION SOURCES
- STATIC

DATA INTEGRATION APPROACH

- Global As View (GAV)



DATA INTEGRATION IN DDSS (3) FEDERATED / AGGREGATED DDSS

CONCEPTUAL SCHEMA DESIGN

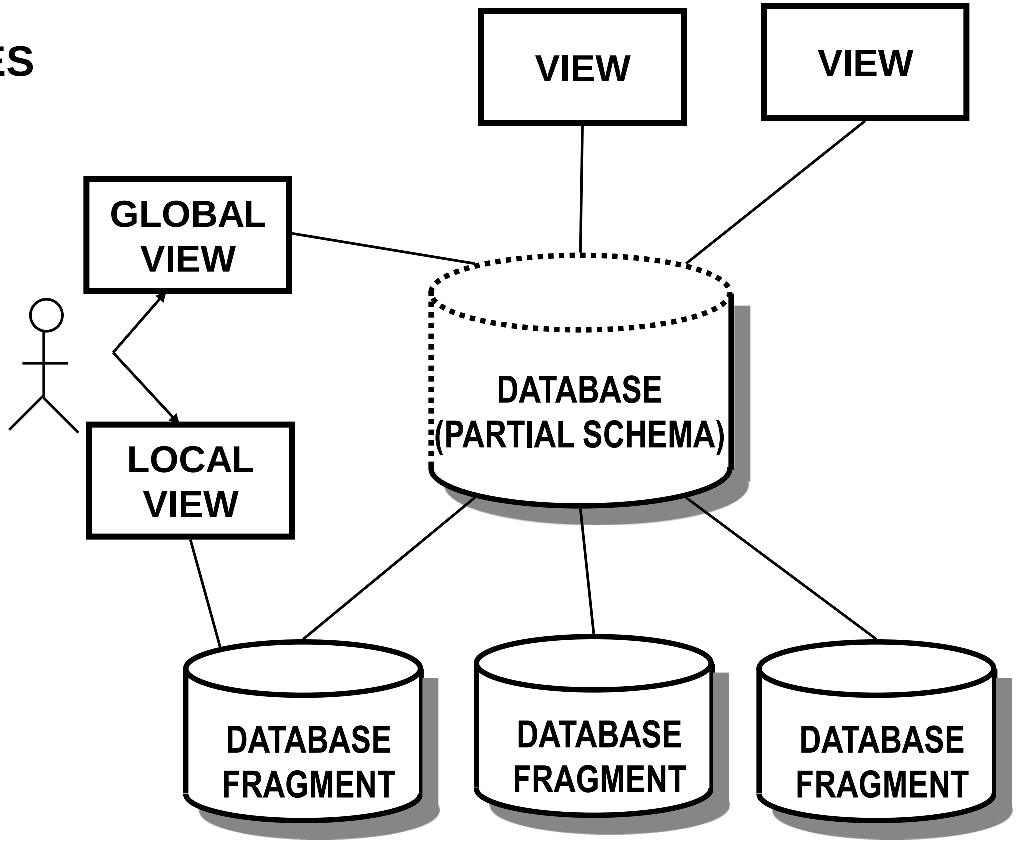
- THE GLOBAL SCHEMA CAPTURES ONLY PART OF THE DDSS DATA
- BOTTOM-UP (AGGREGATION)

DATA DISTRIBUTION DESIGN

- BOTTOM-UP
- EXISTING INFORMATION SOURCES
- STATIC OR MODERATELY DYNAMIC

DATA INTEGRATION APPROACH

- Global As View (GAV)
- Local As View (LAV)



DATA INTEGRATION IN DDSS (4)
DBs WITH NO GLOBAL SCHEMA

DATA DISTRIBUTION DESIGN

- HIGHLY DYNAMIC
- RUN TIME DATA SOURCES DISCOVERY

DATA INTEGRATION APPROACH

- MEDIATORS AND WRAPPERS

  UNIFORM MODEL AND LANGUAGE FOR THE USER

DATA INTEGRATION IN DDSS (5) MULTIDATABASE

DATA DISTRIBUTION DESIGN

- NETWORK CATALOG

DATA INTEGRATION APPROACH

- NO INTEGRATION

DISTRIBUTED DATA MANAGEMENT

- **CLIENT-SERVER ARCHITECTURE**
  – MANY CLIENTS REFER TO A SINGLE SERVER
  – MAINLY USED FOR OLTP ON LOCAL NETWORKS

- **DISTRIBUTED DATABASE**
  – MANY CLIENTS REFER TO MANY SERVERS
  – MAINLY USED FOR OLTP ON LOCAL AND WIDE AREA NETWORKS

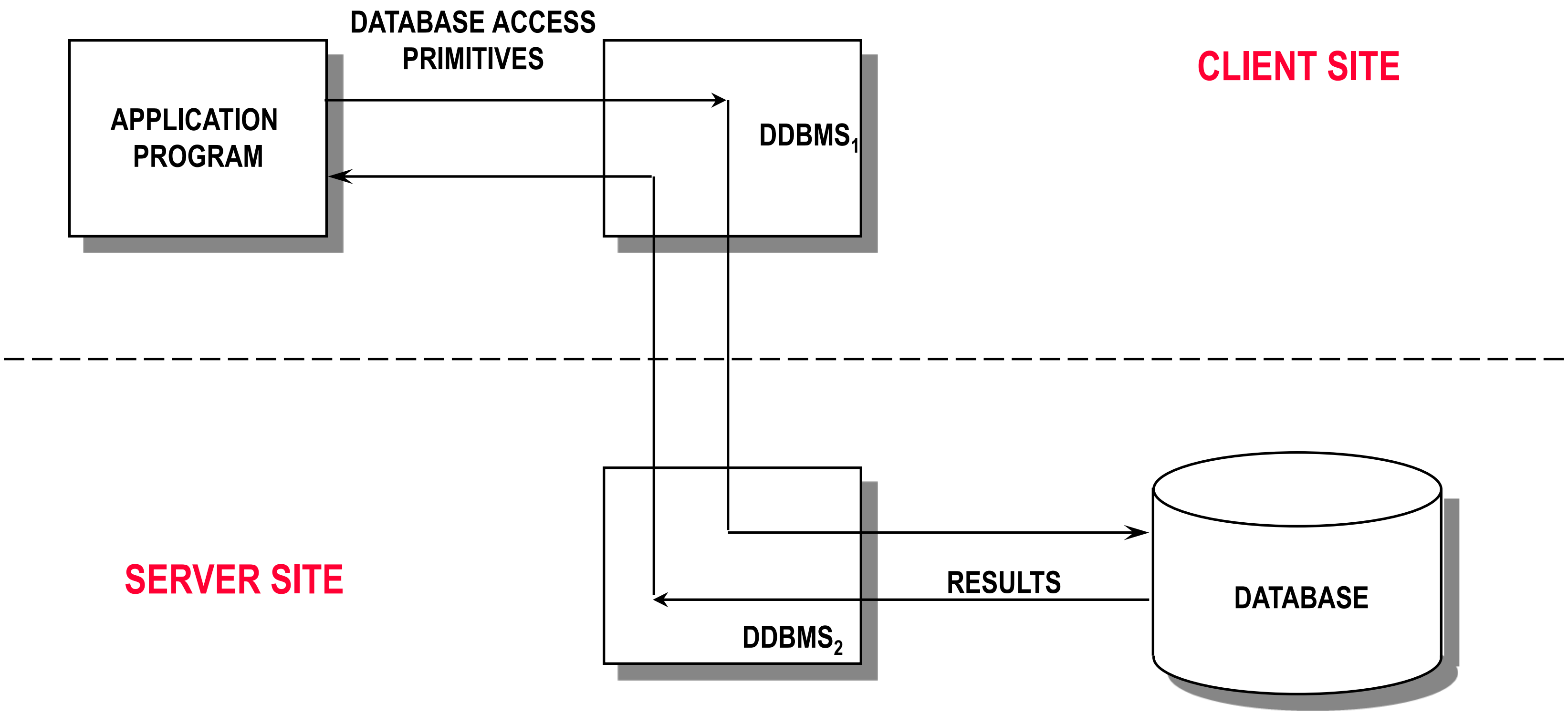
- **DATA WAREHOUSE**
  – DATA COLLECTION FROM MANY DIFFERENT DATA SOURCES
  – USED IN DSS ON LOCAL AND WIDE AREA NETWORKS

DDSS IN A CLIENT-SERVER ARCHITECTURE

- THE CLIENT
  - IS MANAGED BY THE APPLICATION PROGRAMMER
  - SHOWS A FRIENDLY INTERFACE TO THE FINAL USER
  - USES EITHER STATIC OR DYNAMIC SQL
- THE SERVER
  - IS MANAGED BY THE DATABASE ADMINISTRATOR
  - ITS DIMENSION DEPENDS ON THE WORKLOAD AND ON THE SERVICES TO BE DELIVERED
  - MANAGES THE OPTIMIZATION PROCEDURES

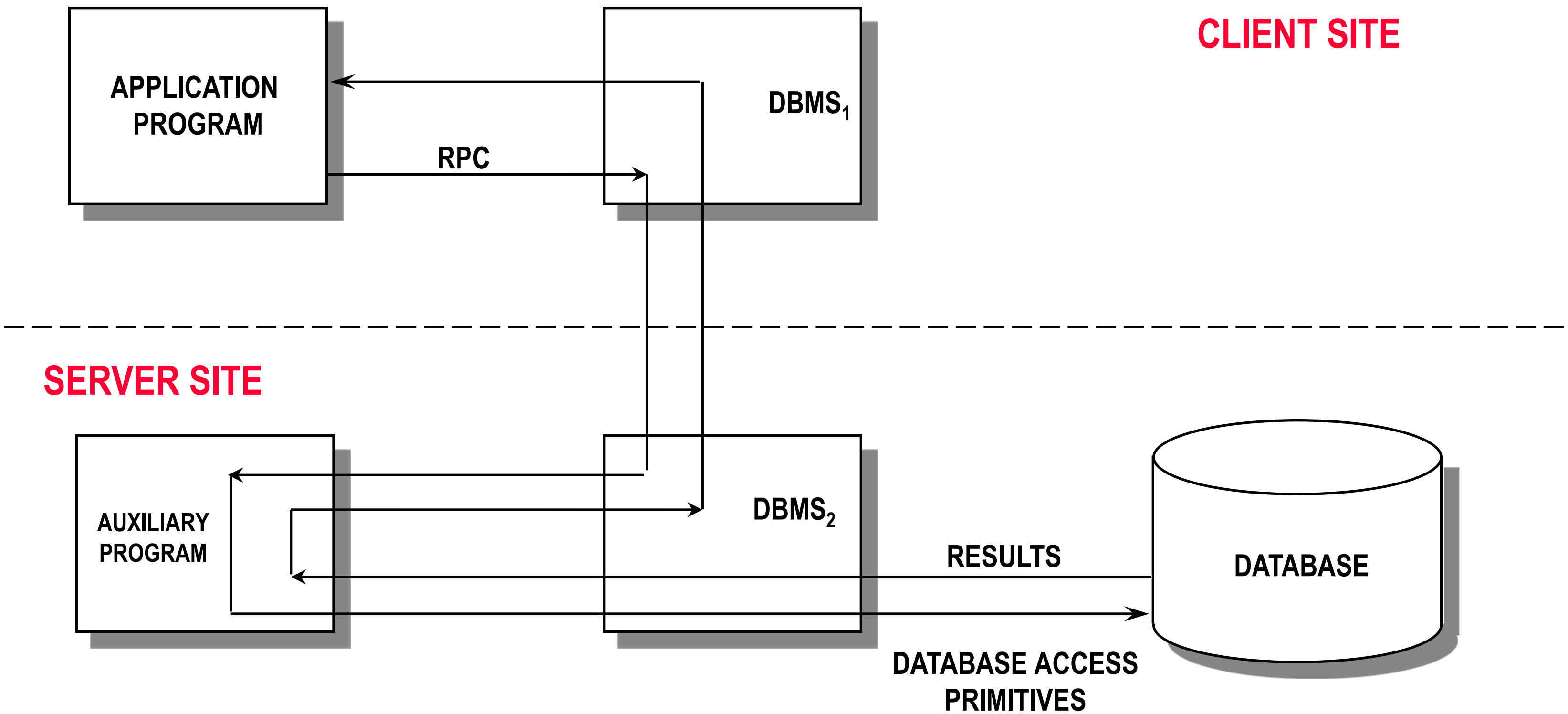
DDSS IN A CLIENT-SERVER ARCHITECTURE

USING DDBMS PRIMITIVES



DDSS IN A CLIENT-SERVER ARCHITECTURE

USING AUXILIARY PROGRAMS AND RPC



DDSS IN A CLIENT-SERVER ARCHITECTURE

- USING DDBMS PRIMITIVES
  - THE DDBMS LOCAL COMPONENT ROUTES THE QUERY TO THE SERVER WHICH ACCESSES THE DATABASE AND SENDS BACK THE RESULTS
  - HIGH DISTRIBUTION TRANSPARENCY THANKS TO GLOBAL FILE NAMES
  - LOW EFFICIENCY SINCE THE ANSWER TRAVELS ONE TUPLE AT A TIME
- USING AUXILIARY PROGRAMS AND RPC
  - THE APPLICATION ASKS THE AUXILIARY PROGRAM TO EXECUTE ON THE SERVER AND TO SEND BACK THE RESULT
  - THE AUXILIARY PROGRAM ASSEMBLES TUPLES INTO RESULT SETS IMPROVING TRANSMISSION EFFICIENCY

DATA WAREHOUSE (DW)

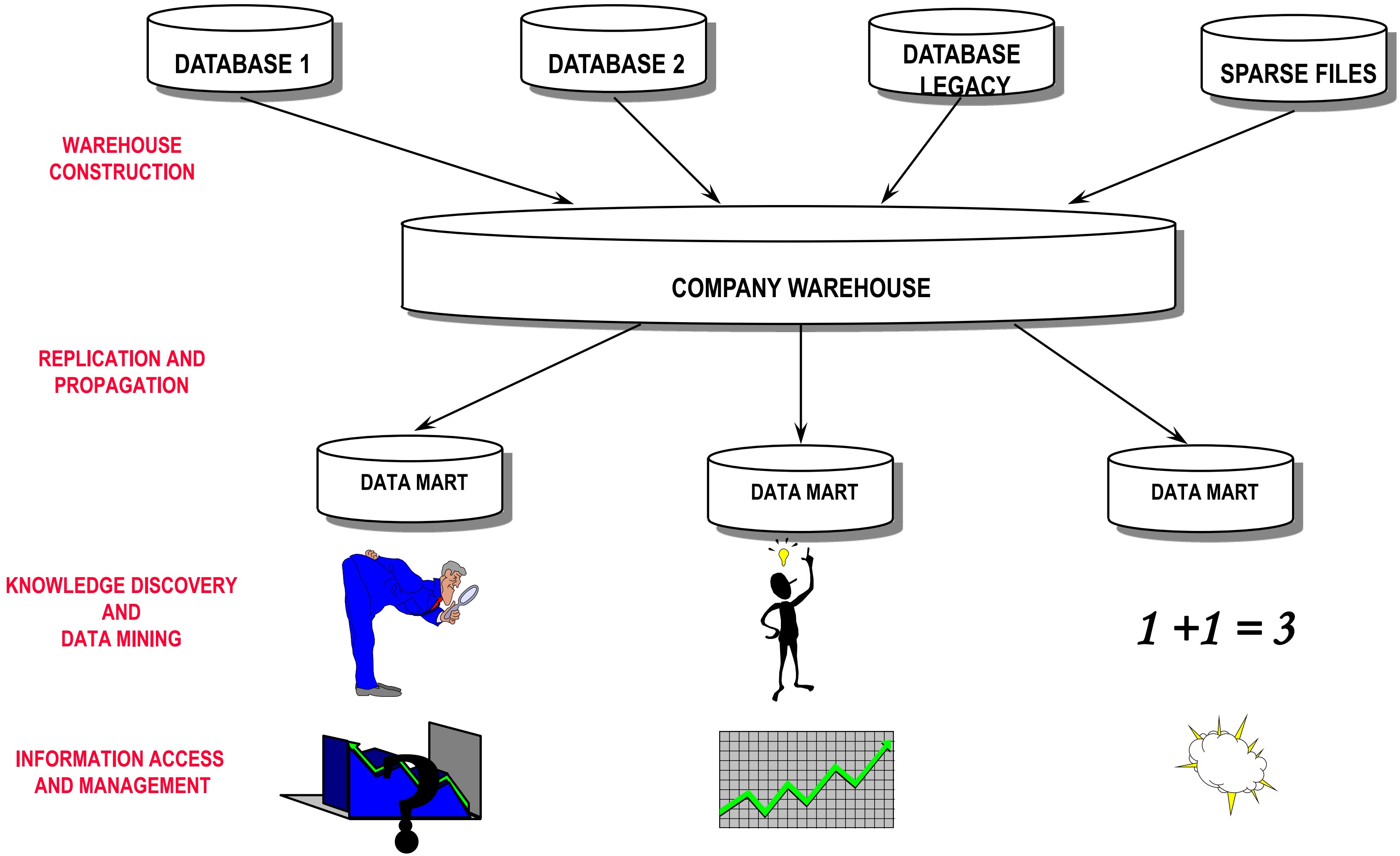
- A TECHNIQUE FOR CORRECTLY ASSEMBLING AND MANAGING DATA COMING FROM DIFFERENT SOURCES TO OBTAIN A DETAILED VIEW OF AN ECONOMIC SYSTEM

- IT IS AN
  - INTEGRATED
  - PERMANENT
  - TIME VARIANT
  - TOPIC ORIENTED

  DATA COLLECTION TO SUPPORT MANAGERIAL DECISIONS

- IT IS THE SEPARATION ELEMENT BETWEEN OLTP AND DSS WORKLOADS

DW ARCHITECTURE



MAIN PROBLEMS IN A DW

- **VIEW AND METADATA MAINTENANCE**
- **REPLICATION MANAGEMENT**
- **CONSISTENCY MANAGEMENT**
- **APPLICATIONS IMPLEMENTATION**

A DISTRIBUTED DATA BASE

IS A SET OF FILES, STORED IN DIFFERENT NODES OF A DISTRIBUTED SYSTEM, WHICH ARE LOGICALLY CORRELATED WITH FUNCTIONAL RELATIONSHIPS OR WHICH ARE REPLICAS OF THE SAME FILE, IN SUCH A WAY AS TO CONSTITUTE A SINGLE DATA COLLECTION

A DISTRIBUTED DATA BASE ...

- …IS A DATA BASE
  - AN INTEGRATED ACCESS MODE TO DATA MUST EXIST
  - IT MUST BE PROTECTED AGAINST INCONSISTENCIES AND FAILURES IN SUCH A WAY AS TO GUARANTEE DATA INTEGRITY
- …IS DISTRIBUTED
  - PHYSICAL DATA DISTRIBUTION MUST BE TRANSPARENT TO THE END USER

SOME DESIGN PROBLEMS

- GENERAL SYSTEM ARCHITECTURE
  - DESIGN FROM SCRATCH
  - SYSTEM RESTRUCTURING
  - SYSTEM AND DATA HETEROGENEITY
- LOGICAL RELATIONS FRAGMENTATION
- REPLICATION AND ALLOCATION
  - HOW MANY COPIES AND WHERE
- ACCESS TO AND PROCESSING OF RELATIONS
- INTEGRITY AND PRIVACY
- RELIABILITY

DATA INDEPENDENCE

THE NOTION OF DATA INDEPENDENCE MUST BE EXTENDED TO ENCOMPASS THE FOLLOWING CASES

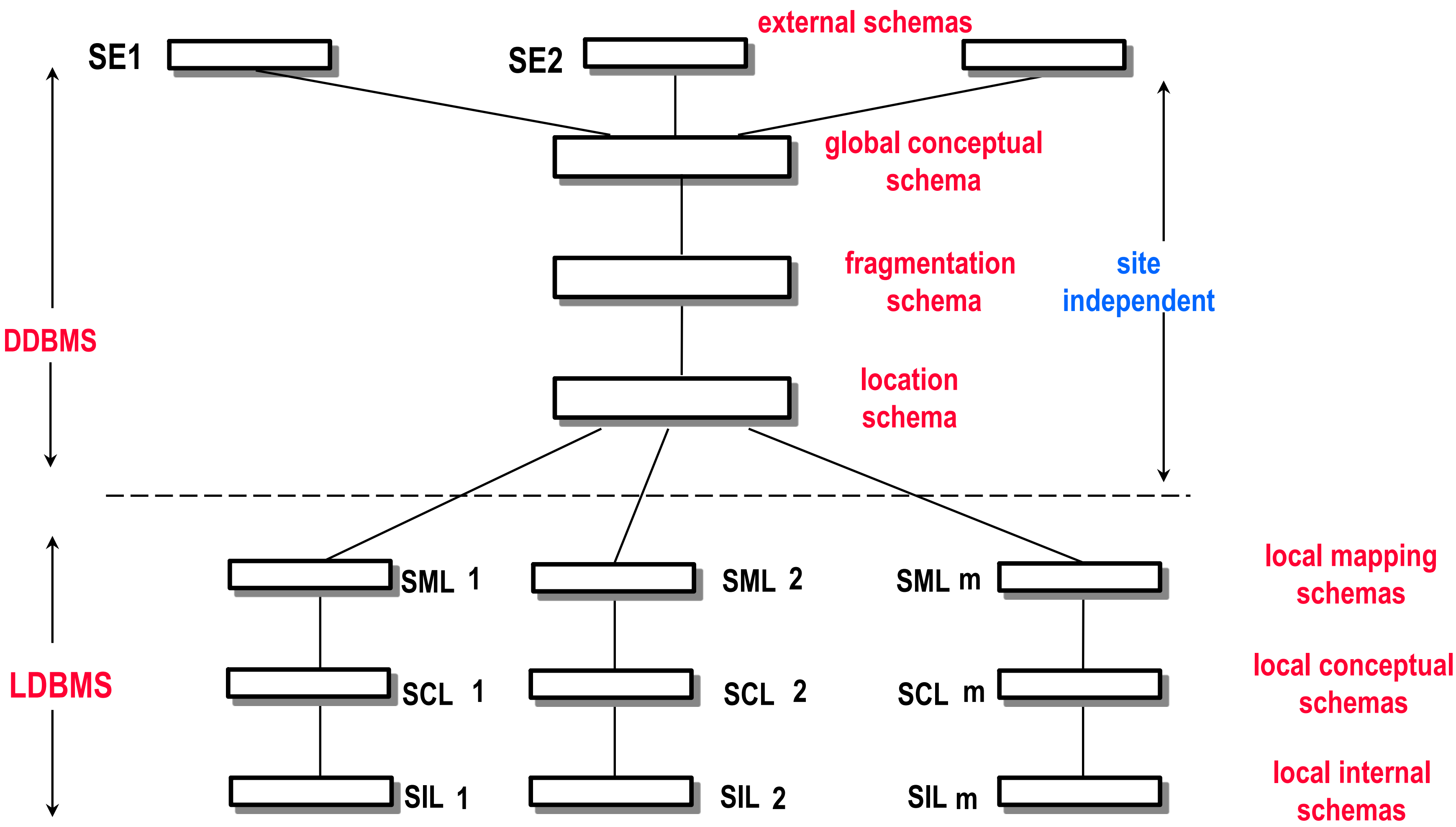
- **LOGICAL**

  THE DB ADMINISTRATOR NEEDS TO RESHAPE THE GLOBAL SCHEMA IN ORDER TO MEET THE REQUIREMENTS OF A VERY LARGE, HETEROGENEOUS AND DYNAMIC SET OF USERS

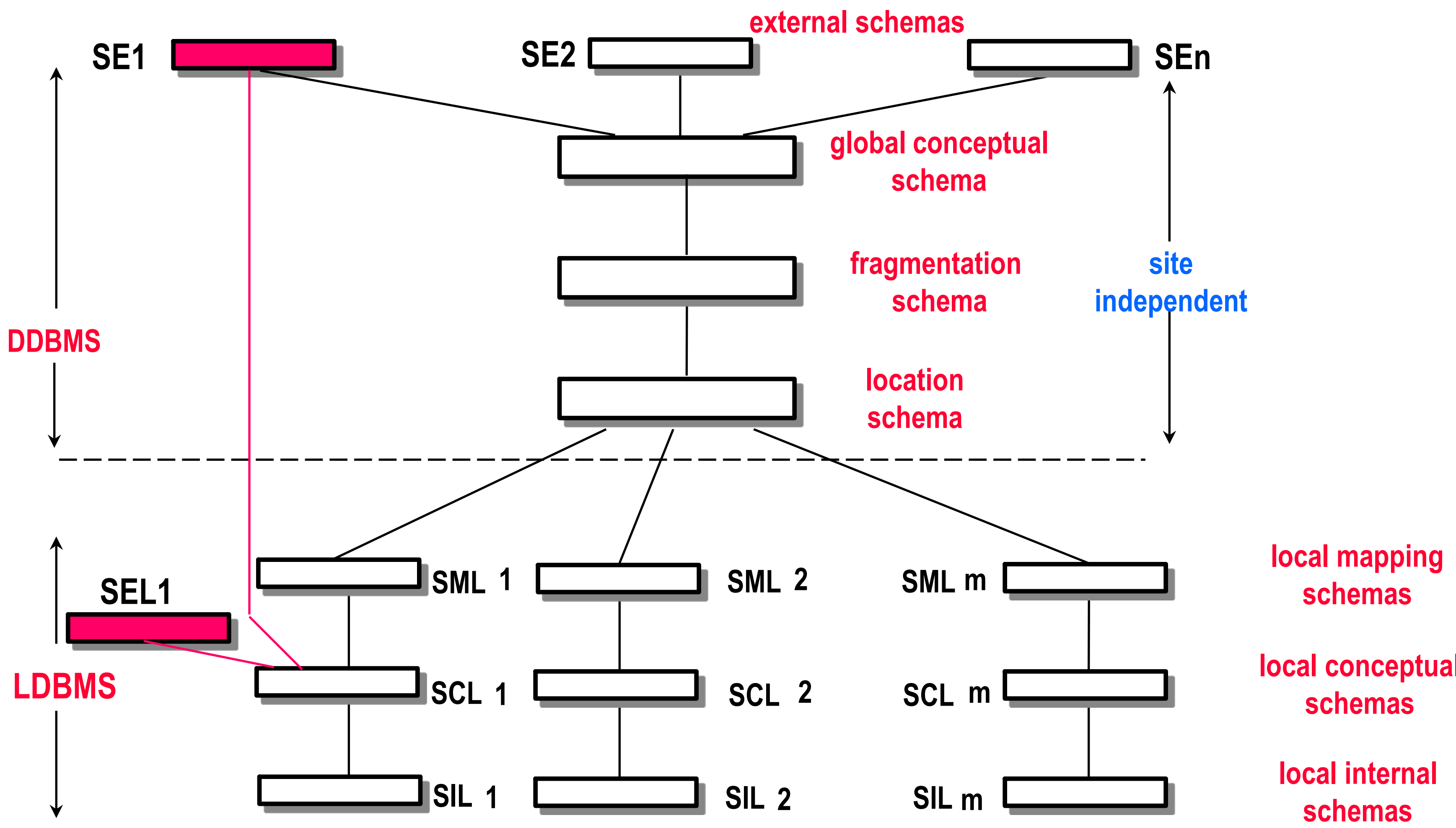
- **PHYSICAL**

  IMMUNITY TO (DYNAMIC) NETWORK CONFIGURATION CHANGES (SITES CONNECTION/DISCONNECTION)

INTEGRATED DDBMS



FEDERATED DDSS



GLOBAL SCHEMA DESIGN

- SIMILAR TO THE VIEW INTEGRATION PROBLEM
- STRUCTURAL CONFLICTS (different schemas)
- SEMANTIC CONFLICTS (even with similar schemas)
  - HOMONYMY (same name, different meaning)
  - AMBIGUITY (different name, same meaning)
  - FORMAT (NUMERIC ⟷ ALPHANUMERIC, ...)
  - AGGREGATIONS (PART-OF, …)
  - DATA OWN SEMANTIC (DIFFERENT MEASURE UNIT, DIFFERENT GRANULARITY, DIFFERENT JUDGEMENT, …)
- UPDATING LIMITED TO LOCAL ACTIVITY

SEMANTIC CONFLICTS RESOLUTION

- **RESTRUCTURING OF LOCAL DATA BASES (NOT FEASIBLE)**

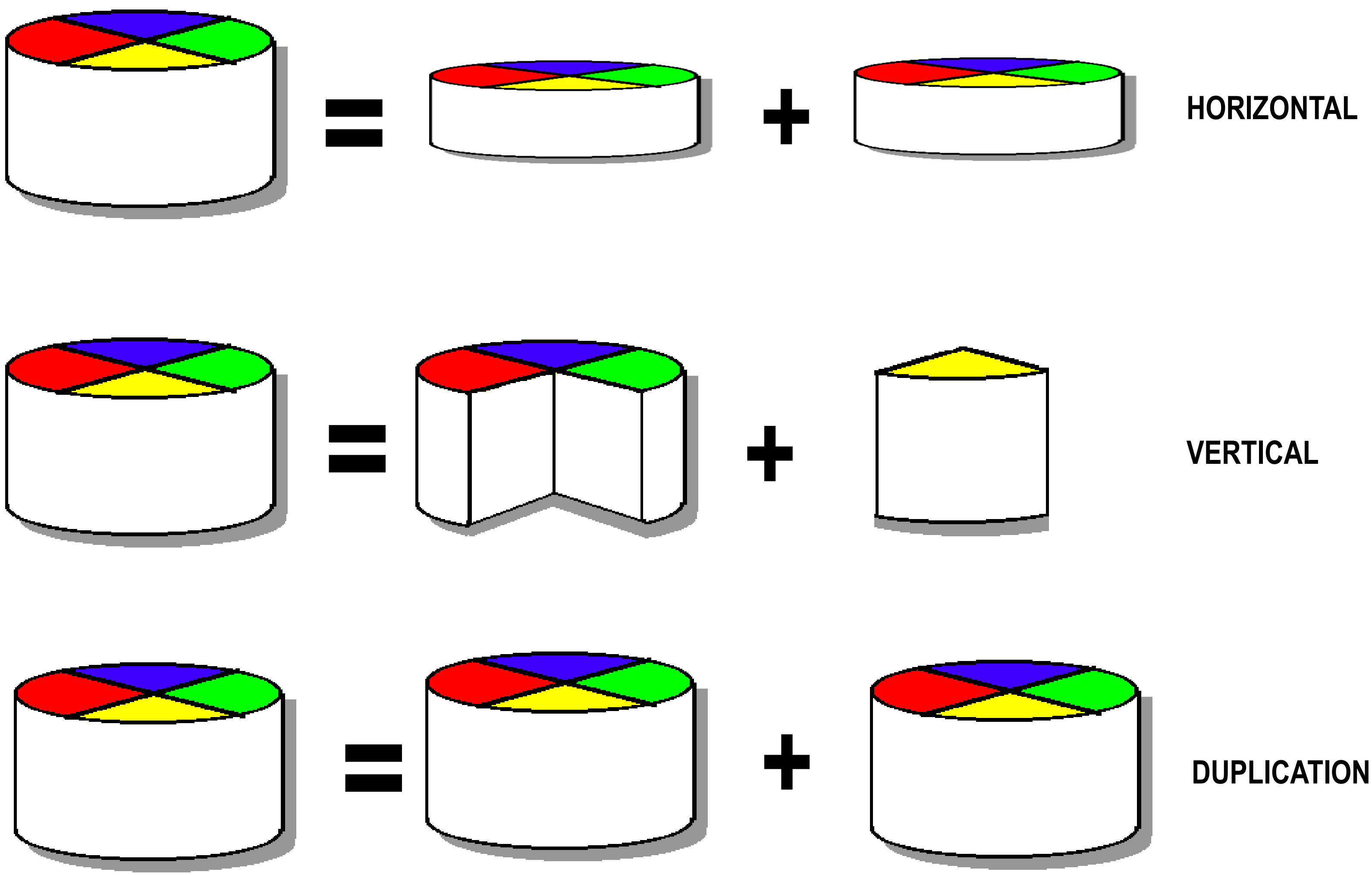
- **EXPLICIT ADDITION IN THE SCHEMAS OF SEMANTIC INFORMATION ABOUT DATA TO ALLOW APPLICATION PROGRAMS TO BEHAVE ACCORDINGLY**

LOGICAL RELATIONS FRAGMENTATION

- **HORIZONTAL**
  - ALL THE FRAGMENTS SHARE THE SAME SCHEMA
  - TUPLES BELONG TO FRAGMENTS ACCORDING TO A SELECTION PREDICATE CORRESPONDING TO A DISTRIBUTION CRITERION
- **VERTICAL**
  - EACH FRAGMENT SCHEMA IS A PROJECTION OF THE GLOBAL RELATION SCHEMA
    - SCHEMAS WITH A NOT EMPTY INTERSECTION
    - DISJOINT SCHEMAS

LOGICAL RELATIONS FRAGMENTATION

The 4 seasons pizza metaphore



DATA REPLICATION

- **PERMANENCE**
  - **A COPY OF A DATA ELEMENT (RELATION OR FRAGMENT) IS PERMANENT IF IT EVOLVES IN TIME UNDER THE DDBMS MANAGEMENT**
  - **IT IS TEMPORARY IF IT IS CREATED ONLY FOR SOME SPECIFIC OPERATION (IN A WORK AREA) AND THEN IT IS CANCELLED OR REFRESHED, ON DEMAND, FROM THE MASTER COPY**

DATA REPLICATION

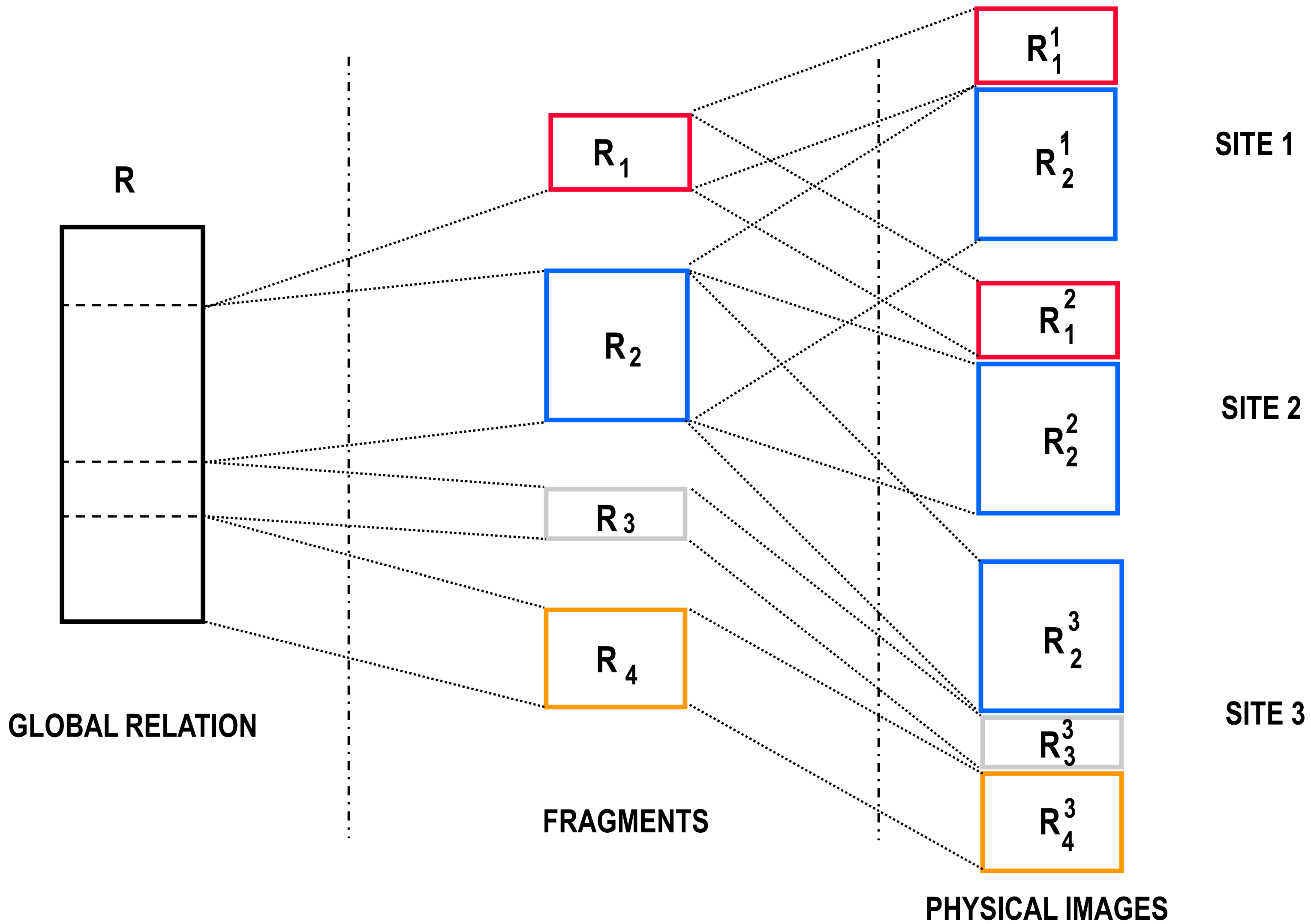
- CONSISTENCY
  - STRONG CONSISTENCY

    AT EVERY INSTANT EACH COPY OF EACH DATA ELEMENT MUST HAVE IDENTICAL VALUES
  - WEAK CONSISTENCY

    UPDATES MADE ON A COPY ARE PROPAGATED TO THE OTHER COPIES LATER ON
  - INDEPENDENCE

    UPDATES ON DIFFERENT COPIES ARE UNCORRELATED (OFTEN USED WITH TEMPORARY COPIES)

FRAGMENTATION AND REPLICATION

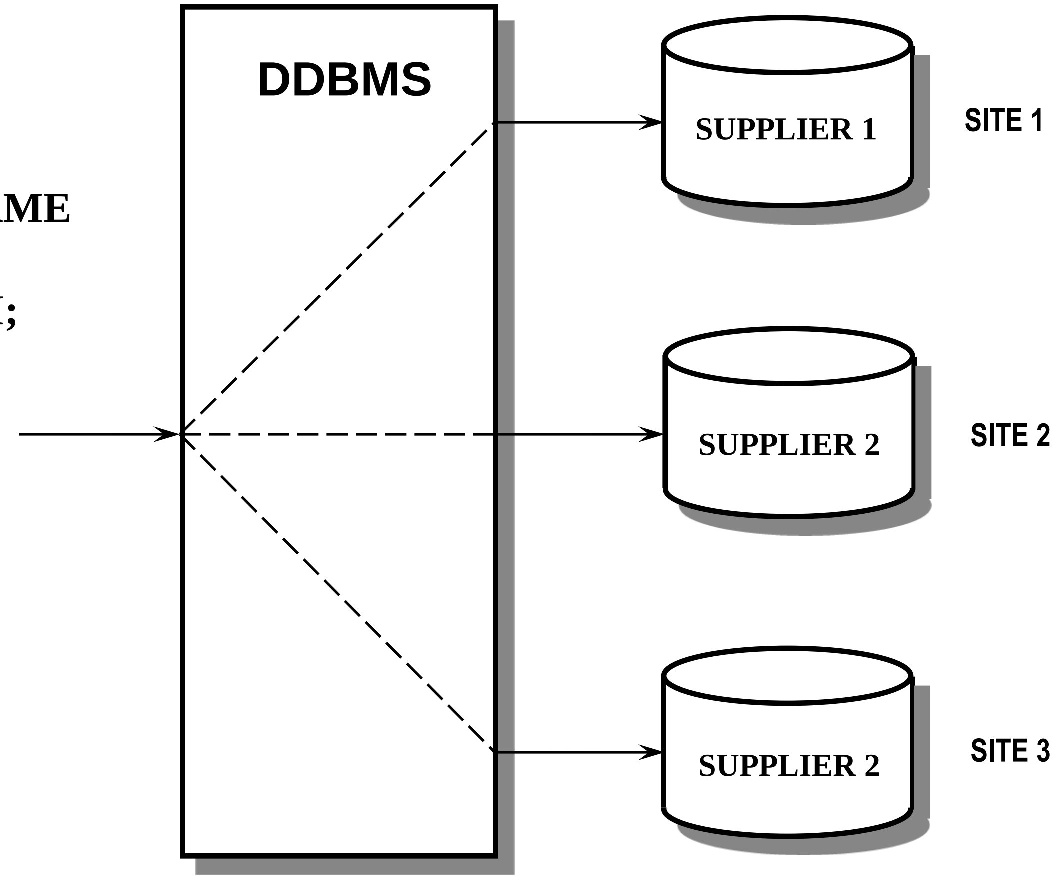


TRANSPARENCY LEVELS

- **TO FRAGMENTATION**
  - **APPLICATION PROGRAMS REFER TO GLOBAL RELATIONS AND IGNORE FRAGMENTATION**
- **TO LOCATION**
  - **APPLICATION PROGRAMS ARE INDEPENDENT OF REPLICATION AND OF PHYSICAL DATA LOCATION, BUT THEY PERCEIVE THE CHANGES IN THE FRAGMENTATION SCHEMA**
- **TO LOCAL MAPPING**
  - **APPLICATION PROGRAMS USE THE OBJECTS (DATA FRAGMENTS OR ACCESS PRIMITIVES) GLOBAL NAMES, BUT THEY MUST SPECIFY THE LOCATION SITE**
- **NO TRANSPARENCY**
  - **THE APPLICATION PROGRAMMER MUST WRITE THE ACCESS MODULE FOR EACH DBMS, WHICH ONLY ACTIVATE THE REMOTE MODULES**

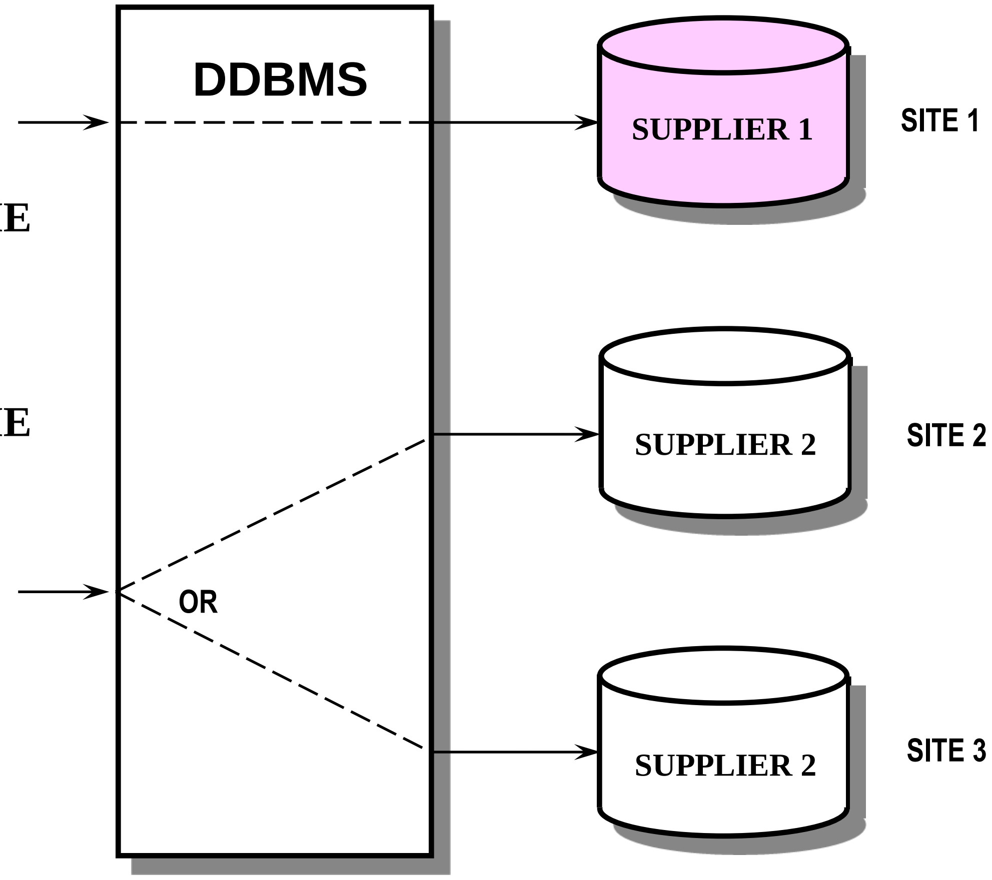
TRANSPARENCY LEVELS: QUERIES
fragmentation transparency

read (terminal,$SNUM);
    select NAME into $NAME
    from SUPPLIER
    where SNUM=$SNUM;
write (terminal, $SNAME)

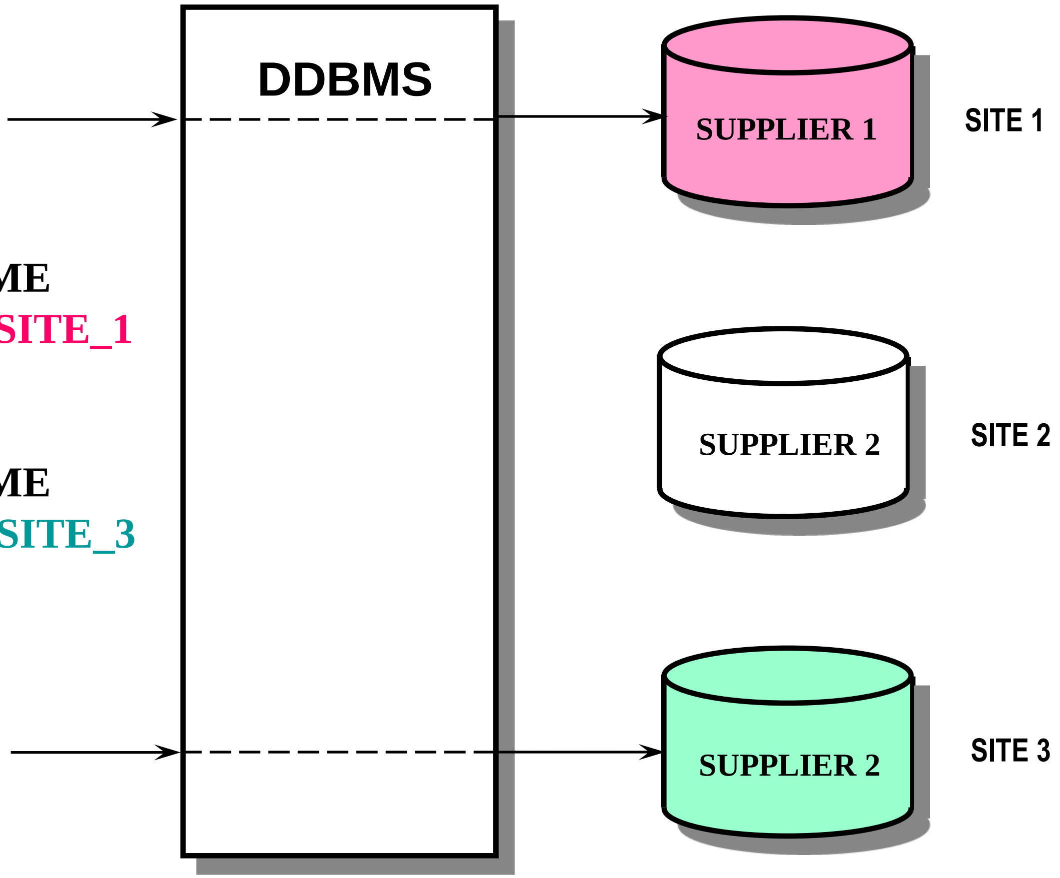


TRANSPARENCY LEVELS: QUERIES
location transparency

```
read (terminal,$SNUM);
        select NAME into $NAME
        from SUPPLIER_1
        where SNUM=$SNUM;
if not #FOUND then
        select NAME into $NAME
        from SUPPLIER_2
        where SNUM=$SNUM;
write (terminal, $SNAME)
```



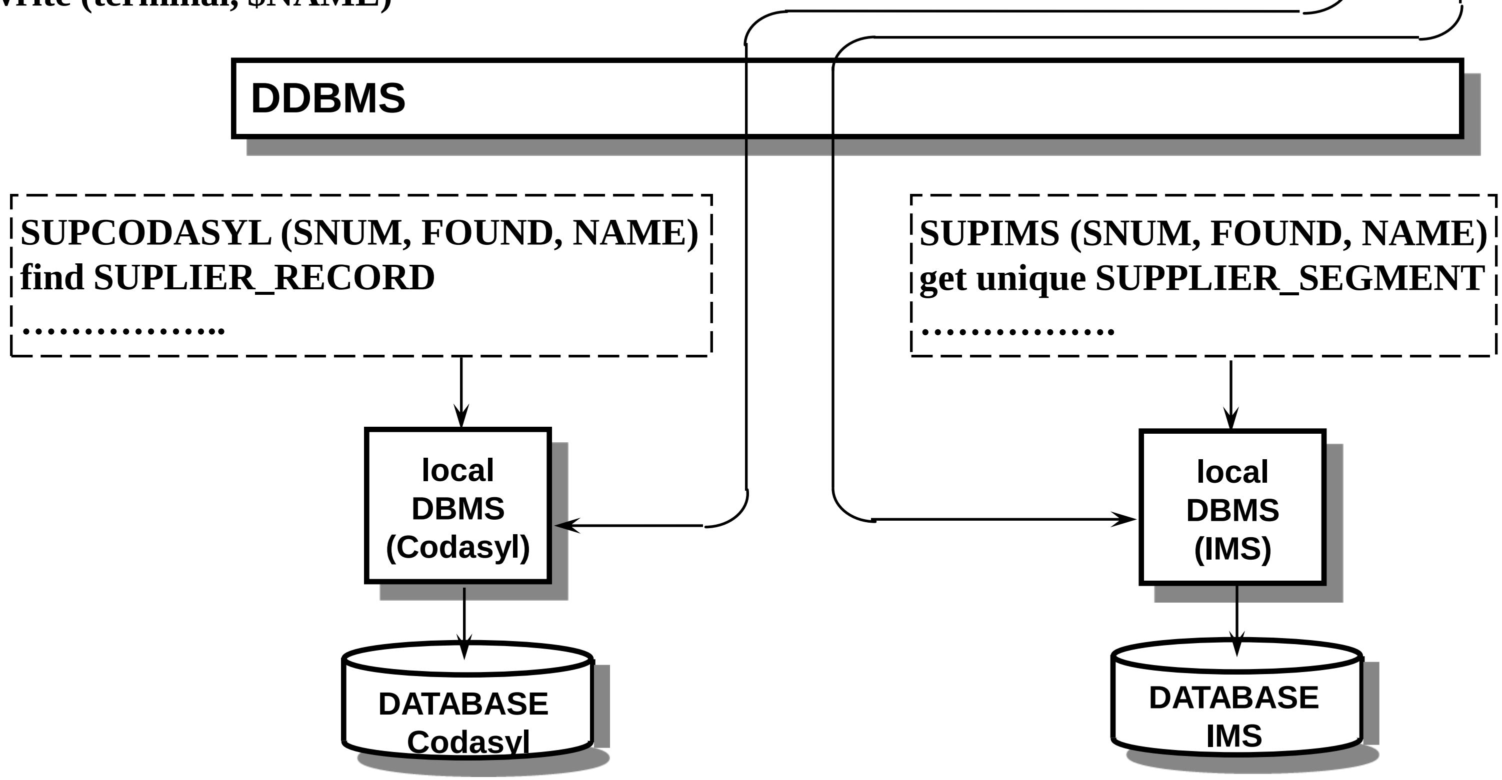
TRANSPARENCY LEVELS: QUERIES
local mapping transparency



read (terminal,$SNUM);
select NAME into $NAME
from SUPPLIER_1 AT SITE_1
where SNUM=$SNUM;
if not #FOUND then
select NAME into $NAME
from SUPPLIER 2_AT SITE_3
where SNUM=$SNUM
write (terminal, $SNAME)

TRANSPARENCY LEVELS: QUERIES
no transparency

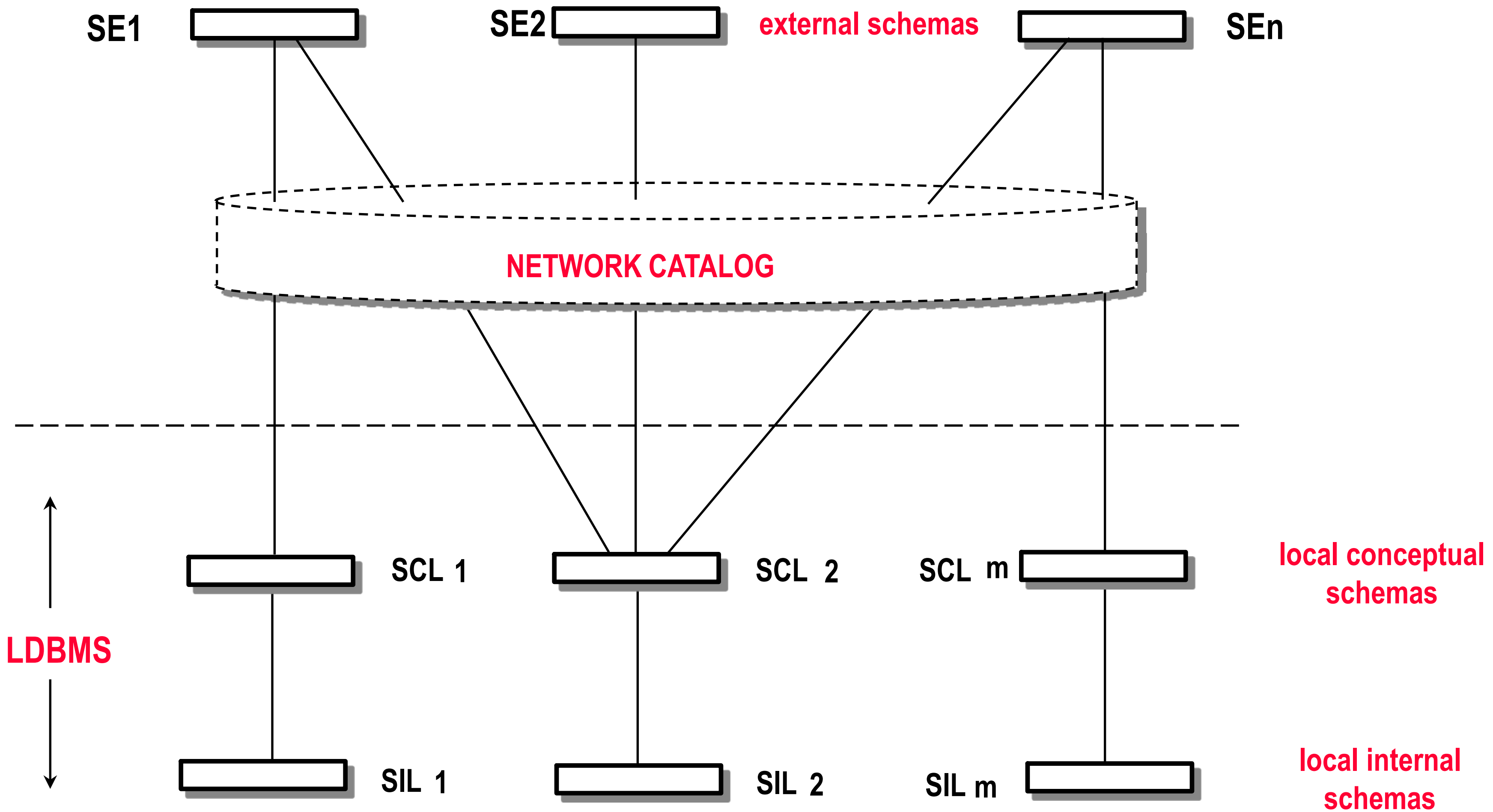
SUPPINQRY:
read (terminal, $SNUM);
execute $SUPIMS ($SNUM, $FOUND, $NAME) at site 1;
if not $FOUND then
execute $SUPCODASYL ($SNUM, $FOUND, $NAME) at site 3;
write (terminal, $NAME)



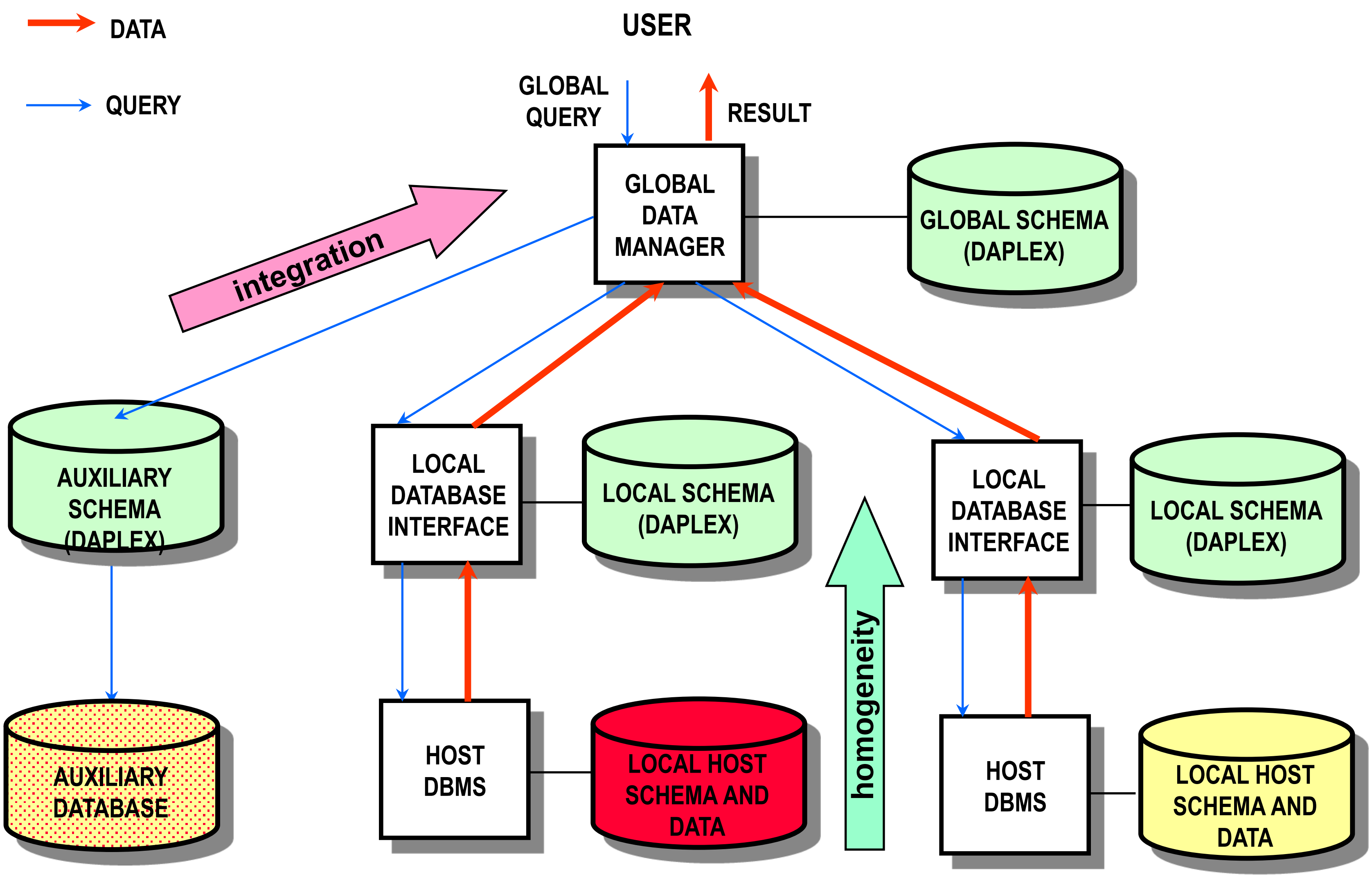
TRANSPARENCY LEVELS: UPDATES

- FOR FRAGMENTATION TRANSPARENCY
  - IF AN ATTRIBUTE BELONGING TO AN HORIZONTAL FRAGMENTATION PREDICATE IS UPDATED THE TUPLE SHALL BE MOVED BETWEEN THE FRAGMENTS
- FOR LOCATION AND REPLICATION TRANSPARENCY
  - ALL THE COPIES MUST BE UPDATED SIMULTANEOUSLY

MULTIDATABASE

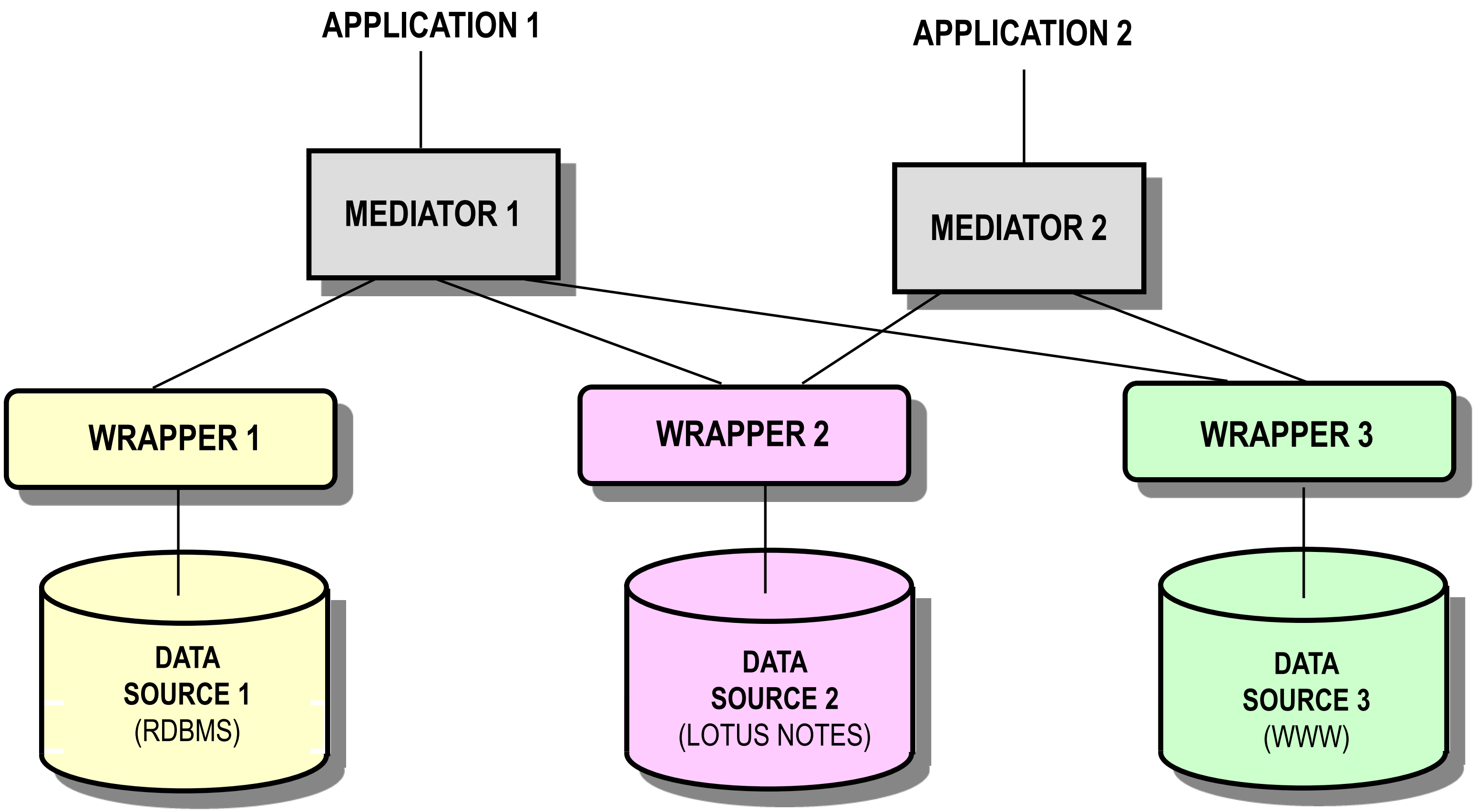


MULTIBASE



DATA
QUERY
USER
GLOBAL QUERY
RESULT
integration
GLOBAL DATA MANAGER
GLOBAL SCHEMA (DAPLEX)
AUXILIARY SCHEMA (DAPLEX)
LOCAL DATABASE INTERFACE
LOCAL SCHEMA (DAPLEX)
LOCAL DATABASE INTERFACE
LOCAL SCHEMA (DAPLEX)
homogeneity
AUXILIARY DATABASE
HOST DBMS
LOCAL HOST SCHEMA AND DATA
HOST DBMS
LOCAL HOST SCHEMA AND DATA

AN EXAMPLE OF ARCHITECTURE WITH MEDIATORS (TSIMMIS)



EXAMPLE OF SEMANTIC CONFLICT

CONTEXT C1:
- *MONEY AMOUNTS IN ORIGINAL CURRENCY*
- *MONEY AMOUNTS SCALE 1:1 BUT FOR YEN WHICH IS SCALED 1:1000*

r1

COMPANY	REVENUE	COUNTRY
IBM	1 000 000	USA
NTT	1 000 000	JPN

r4

COUNTRY	CURRENCY
USA	USD
JPN	JPY

CURRENCY	SCALE
ALL	1:1
JPY	1:1000

CONTEXT C2:
- *MONEY AMOUNTS IN USD SCALE 1:1*

r2

COMPANY	EXPENSES
IBM	1 500 000
NTT	5 000 000

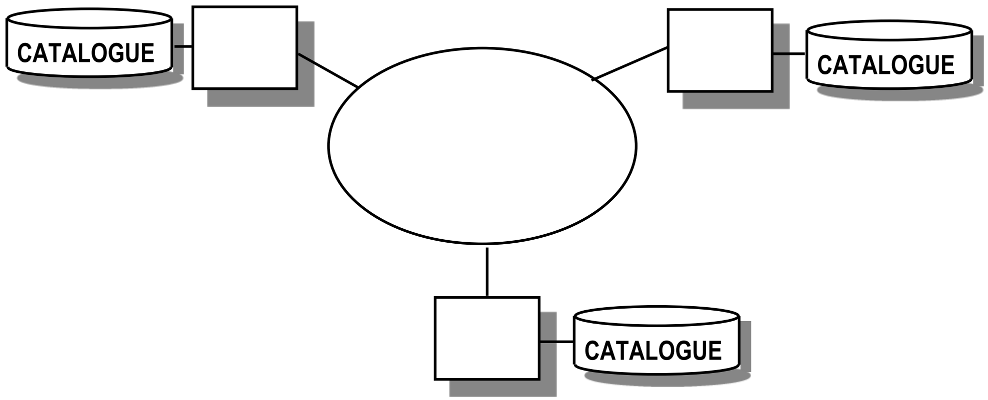
r3

FROMCURREN	TOCURRENCY	EXCHRATE
USD	JPY	104.0
JPY	USD	.0096

```
select r1.company, r1.revenue
from r1, r2
where r1.company = r2.company
and r1.revenue > r2.expenses
```

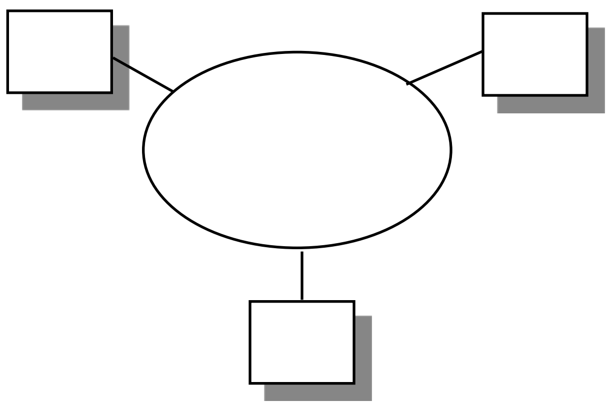
SOURCE: C. H. GOH et Al.

NETWORK CATALOGUE



FULLY REPLICATED

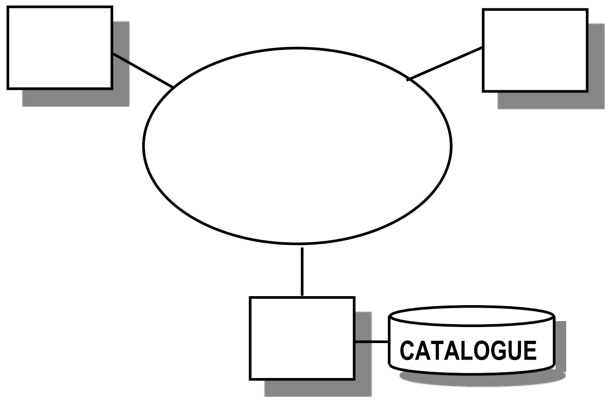
- FAST ACCESS
- DISK OCCUPATION
- DATA CONSISTENCY



NO CATALOGUE AT ALL

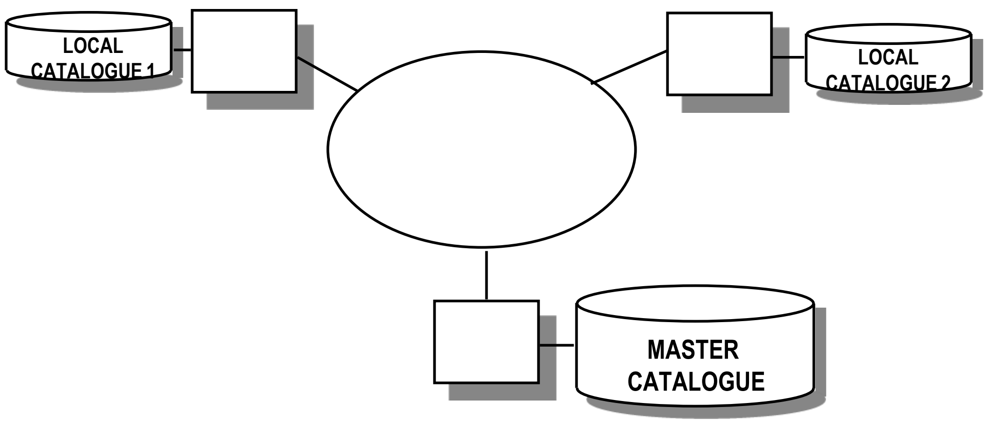
- BROADCASTING NEED
- ACCESS OVERHEAD

NETWORK CATALOGUE



FULLY CENTRALIZED

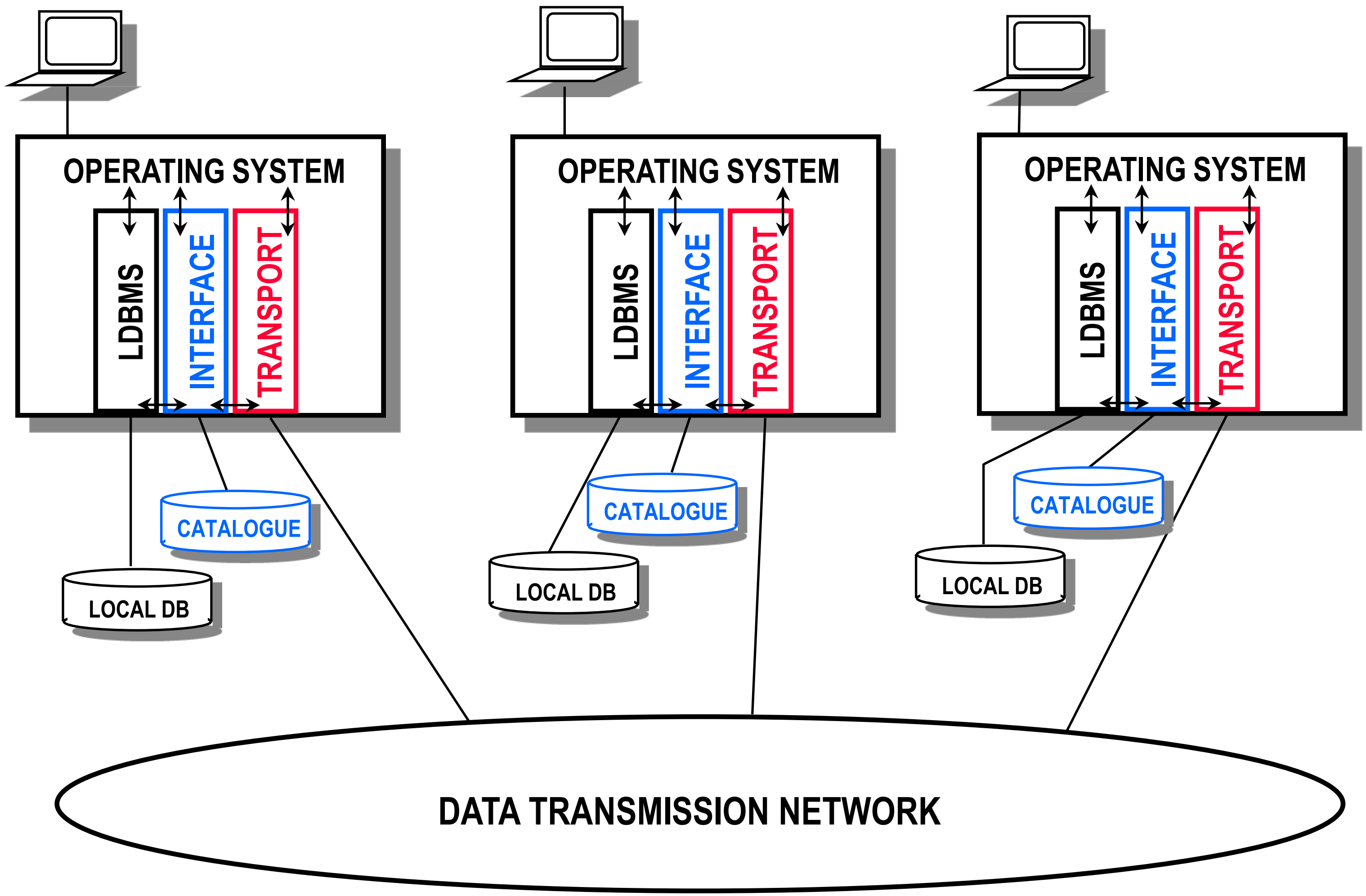
- ACCESS BOTTLENECK
- DISK OCCUPATION
- DATA CONSISTENCY



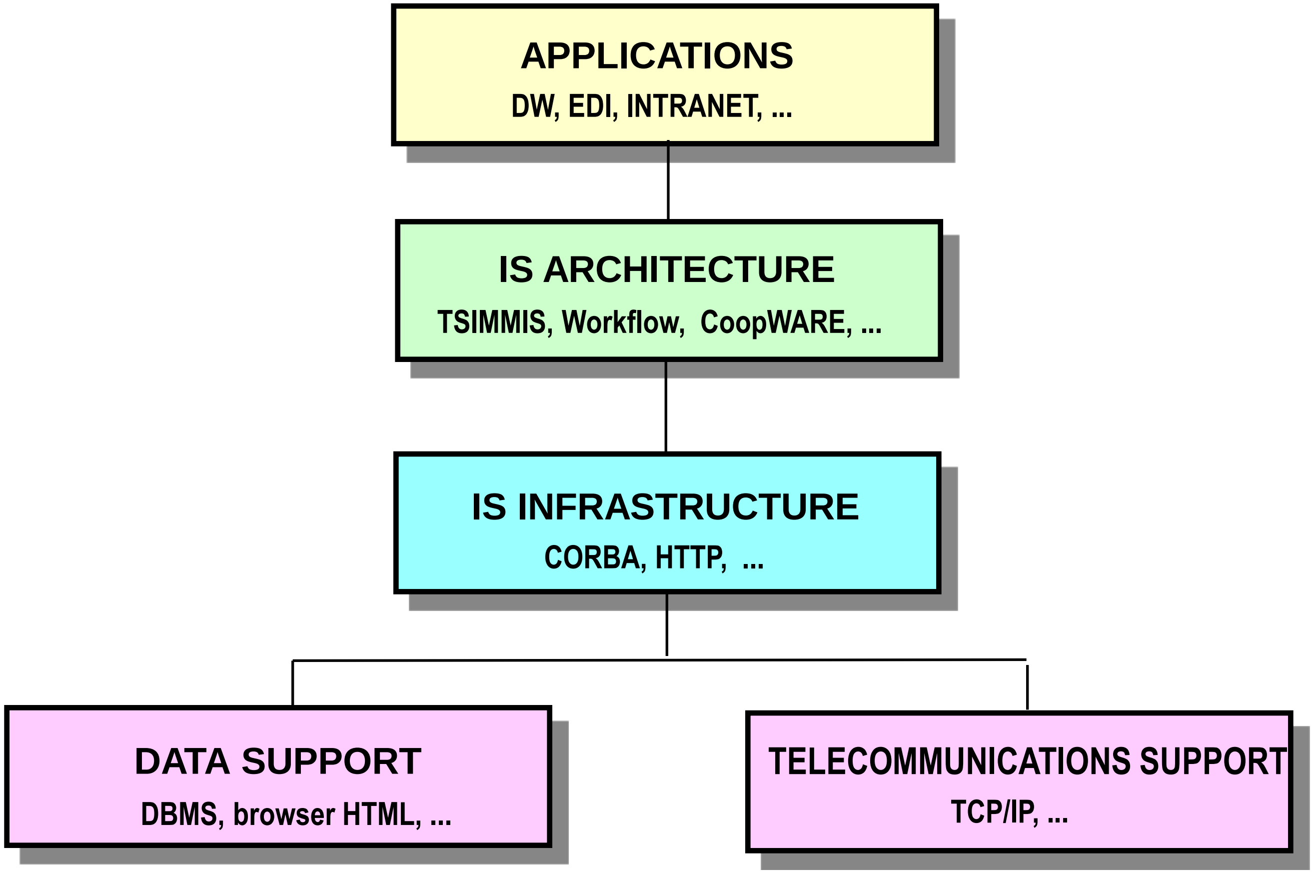
PARTIALLY REPLICATED

- A GOOD COMPROMISE BETWEEN OVERHEAD AND ACCESS EFFICIENCY

OVERALL ARCHITECTURE



A MODEL FOR DISTRIBUTED AND HETEROGENEOUS INFORMATION SERVICES



OTHER PARADIGMS

- **CONTENT DELIVERY NETWORKS (CDN)**
  - **A SET OF EDGE SERVERS CACHES REPLICATE INFORMATION OF ORIGINAL SOURCES**
    - LOAD BALANCING, BANDWITH SAVINGS, REDUCING ROUNDTRIP TIME
- **PEER-TO-PEER NETWORKS (P2P)**
  - **AD-HOC AGGREGATION OF RESOURCES TO FORM A DECENTRALIZED SYSTEM, REMOVING CENTRALIZED AUTHORITY**
    - SCALABILITY, RELIABILITY, RESOURCE SHARING, …
- **DATA GRIDS**
  - **PROVIDE SERVICES THAT HELP USERS DISCOVER, TRANSFER, AND MANIPULATE MASSIVE DATASETS**
- **DATA CLOUDS**
  - **PROVIDE ON-DEMAND SERVICES THROUGH THE INTERNET**

TECHNOLOGY TRENDS

MOST OF THESE SLIDES HAVE BEEN DRAWN SINCE THE ’80s OF LAST CENTURY; ARE THEY OBSOLETE?

NO!

- HOMOGENEOUS DDBs REMAIN A USEFUL CONCEPTUAL PARADIGM FOR DISCUSSING THE MAIN PROBLEMS
- FEDERATED- AND MULTI-DBs ARE STILL AT WORK, THEIR ROLE HAVING BEEN ENHANCED BY THE DIFFUSION OF THE INTERNET
- CLOUD COMPUTING BROUGHT TO A NEW LIFE THE OPTIMISATION ISSUES RELATED TO LOAD SHARING AND TO DATA PARTITIONING

GRID vs. CLOUD COMPUTING

	GRID	CLOUD
GOAL	Fosters collaboration among participating organizations to leverage existing resources	Provide a rather fixed (distributed) infrastructure to all kinds of users
TARGET	Large scientific computations and enterprise applications	On demand, reliable services over the Internet with easy access to virtually infinite computing, storage and network resources
DATA MANAGEMENT	File based. Global (distributed/replicated) directories accessible via Web Services (OGSA-DAI multidatabase)	Optimized versions of RDBMS (e.g. column-oriented, shared-nothing DB) MapReduce paradigm

CLOUD COMPUTING

CLOUD COMPUTING: DELIVERING HOSTED SERVICES THROUGH THE INTERNET

- **IaaS (Infrastructure as a Service)**

  Provides virtual server instances with unique IP address and storage on demand (e.g. Amazon Web Services)

- **PaaS (Platform as a Service)**

  Set of software and development tools hosted on the provider's infrastructure (e.g. GoogleApps)

- **SaaS (Software as a Service)**

  The service provider supplies both the application and the data. The user operates from a front-end portal

CLOUD COMPUTING

CLOUD STORAGE

DATA IS STORED ON MULTIPLE (THIRD PARTIES) VIRTUAL SERVERS

MAIN PROS

- DEVICE INDEPENDENCE
- RELIABILITY (RESOURCES REPLICATION)
- SCALABILITY (RESOURCES USED ON DEMAND)
- MAINTENANCE

MAIN CONS

- LOSS OF CONTROL
- PRIVACY LEAKS

CLOUD COMPUTING

NOT SUITED (AT THE MOMENT) FOR OLTP

- **DIFFICULTY TO GUARANTEE ACID PROPERTIES**
  - **COMPLEX DISTRIBUTED LOCKING AND COMMIT PROTOCOLS**
  - **INCREASED LATENCY OWING TO DATA SHIPPING OVER THE NETWORK**
  - ***GILBERT AND LYNCH* ARGUE THAT ONLY TWO OUT OF THREE AMONG {consistency, availability, tolerance to partitons} PROPERTIES CAN BE ASSURED IN A SHARED DATA SYSTEM**
  - **THEREFORE, IN LARGE NETWORKS, AVAILABILITY IS PREFERRED TO CONSISTENCY**
  - **DATA SECURITY AND PRIVACY AT RISK ON UNTRUSTED HOSTS**

CLOUD COMPUTING

BETTER SUITED FOR OLAP PROCESSING IN:

- MEDIUM-SIZED BUSINESSES (NO LARGE INITIAL INVESTMENT NEEDED)
- SUDDEN/SHORT-TERM PROJECTS (NO LARGE SET-UP TIMES)
- PUBLICILY DISPLAYED DW WINDOWS (NO PRIVACY NEEDED)

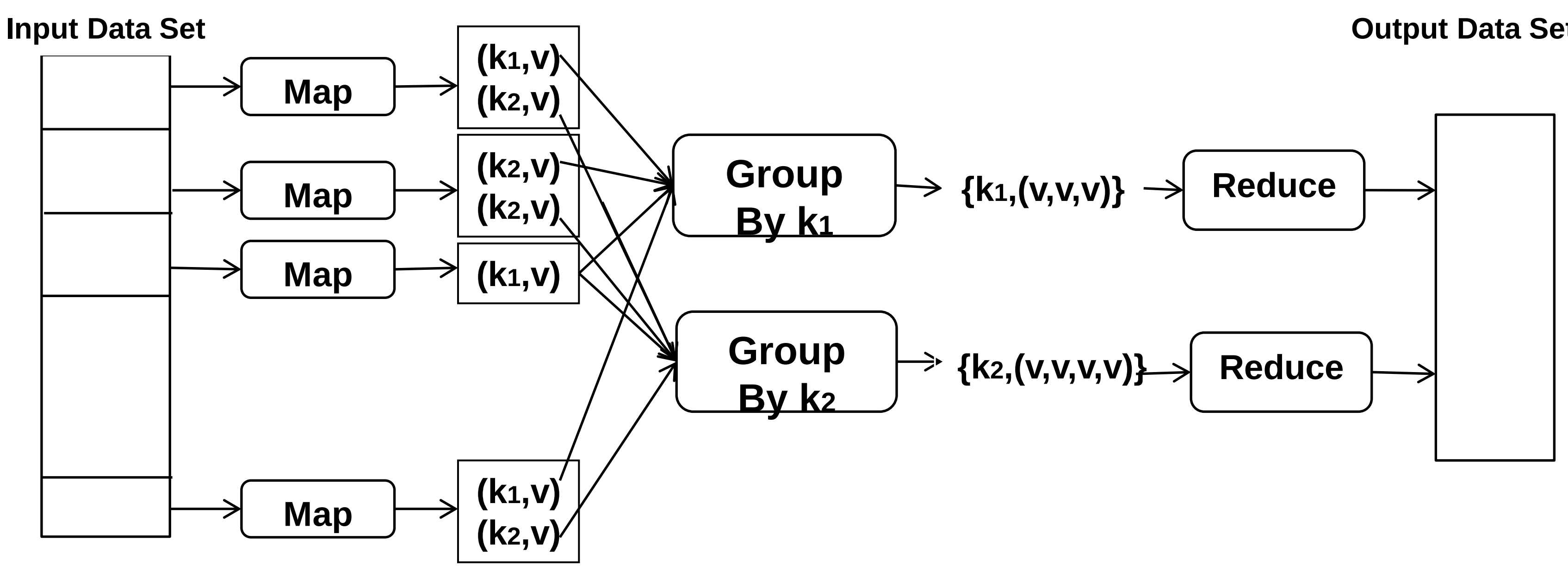
FEATURING

- VERY LARGE SCALE DATA SYSTEMS (PByte)
- MOSTLY READ-ONLY WITH OCCASIONAL BATCH INSERTS
- THEREFORE, ACID PROPERTIES NOT NEEDED
- SENSITIVE DATA CAN BE ANONYMIZED, ENCYPTED OR USED AT A LESS GRANULAR LEVEL

CLOUD COMPUTING: SW SOLUTIONS

MapReduce PARADIGM

- Map is applied to each record in the input dataset to compute one or more (key, value) pairs (like a `group-by` clause in SQL)
- Reduce is applied to all the values which share the same unique key in order to compute a combined result (like the SQL aggregate function computed over the rows with the same grouped attribute)



CLOUD COMPUTING: SW SOLUTIONS

SQL vs. MapReduce

RELATION	**EMP(ENAME,TITLE,CITY)**
QUERY	**FOR EACH CITY, RETURN THE NUMBER OF EMPLOYEES NAMED "Smith"**

- **SQL**

```
SELECT        CITY, COUNT(*)
FROM          EMP
WHERE         ENAME LIKE "%Smith"
GROUP BY CITY
```

- **MapReduce (pseudocode)**

  Map (Input (TID,emp), Output: (CITY,l))

  If emp.ENAME like " %Smith" return (CITY,l)

  Reduce (Input (CITY,list(l)), Output: (CITY,SUM(list(l)))

  Return (CITY,SUM(l*))

CLOUD COMPUTING: SW SOLUTIONS

MAP-REDUCE SOFTWARE {Google, Hadoop (Yahoo)}

- Automate the parallelization of large scale data analysis workloads
- Fault tolerance as a high priority; tasks assigned to a failed machine are transparently reassigned to a working one
- Tasks are replicated on different machines and the first replica which terminates ends the task
- Borne and fit for indexing large sets of web (non structured) documents
- Not SQL compliant → difficult interfacing with other business intelligence products
- Ability to operate on encrypted data only through application code
- Performance needs improvement and is application dependent

CLOUD COMPUTING: SW SOLUTIONS

SHARE-NOTHING PARALLEL DATABASES

{Teradata, Vertica, DBMS-X, …}

- **DISTRIBUTED DB PARTITION DATA FUNCTIONALLY AMONG THE SERVERS**
- **PARALLEL DB PARTITION DATA FOLLOWING ONLY PERFORMANCE ISSUES**
  - **Higher efficiency through built-in optimization structures in the local DBMSs (indexes, materialized views, compression, …)**
  - **Native ability to interface with business intelligence products**
  - **Unnecessary (read-only) query restarts after failures**
  - **Running in a heterogeneous environment can degrade the system's performance**
  - **Limited ability to directly operate on encrypted data**

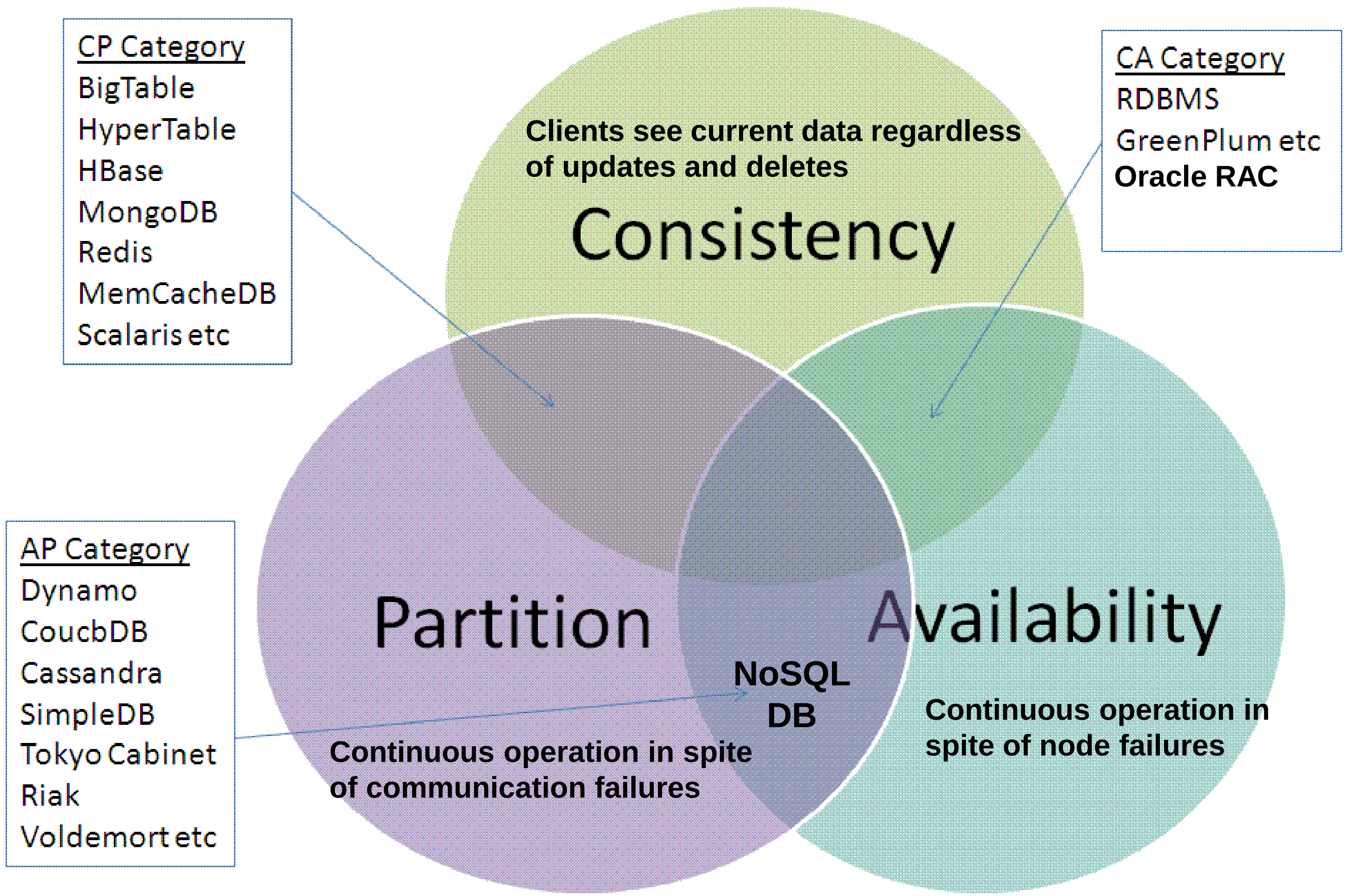
MOST RECENT ISSUES: *BIG DATA*

- **NoSQL DATABASES**
  - **SUPPORT FLEXIBLE SCHEMA**
  - **SCALE HORIZONTALLY**
  - **DO NOT SUPPORT ACID PROPERTIES**
    - UPDATES PERFORMED ASYNCHRONOUSLY
    - POTENTIAL DATA INCONSISTENCY RESOLVED BY READERS

- **THE CAP THEOREM**

  ANY NETWORKED SHARED-DATA SYSTEM CAN HAVE AT MOST TWO OUT OF THREE DESIRABLE PROPERTIES
  - **CONSISTENCY (C)**
  - **HIGH AVAILABILITY (A)**
  - **TOLERANCE TO NEWORK PARTITIONS (P)**
    - **PARTITION ≡ TIME BOUND ON COMMUNICATION LATENCY**

THE CAP THEOREM



Adapted from: amitpiplani.blogspot.com

REFERENCES

TEXTBOOK

Tamer Ötzsu M., Valduriez P. – Principles of *Distributed Database Systems: 3rd ed.* - Springer, 2011

OTHER REFERENCE MATERIAL

- Abadi Daniel J. - *Data Management in the Cloud: Limitations and Opportunities* - IEEE Data Engineering Bulletin, Vol. 32 No. 1, March 2009
  http://sites.computer.org/debull/A09mar/A09MAR-CD.pdf#page=5
- Dean J., Ghemawhat S. – *MapReduce: A Flexible Data Processing Tool* - CACM, Vol.53, n. 1, pp. 72-77, 2010
- Foster I., Yong Zhao, Raicu I., Lu S - *Cloud Computing and Grid Computing 360-Degree Compared* - Grid Computing Environments Workshop 2008**,** pp. 1-10, 2008
  http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4738445

REFERENCES

- Gilbert S., Lynch N. – *Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services* – SIGACT News, Vol. 33, n. 2, pp. 51-59, 2002
  http://dl.acm.org/citation.cfm?id=J697
- Pavlo A., et Al. – *A comparison of approaches to large-scale data analysis* – Proc. ACM-SIGMOD Int. Conf., Providence (RI), 2009
  http://database.cs.brown.edu/projects/mapreduce-vs-dbms/
- Stonebraker M., et Al. – *MapReduce and Parallel DBMSs: Friends or Foes?* – CACM, Vol.53, n. 1, pp. 64-71, 2010
- Venugopal S., Buyya R., Ramamohanarao K. - *A Taxonomy of Data Grids for Distributed Data Sharing, Management, and Processing* - ACM Computing Surveys, Vol. 38, pp. 1-53, March 2006
- AA.VV. – The Growing Impact of the CAP Theorem – IEEE Computer, February 2012