Basics of Database Security

(Spring Computer Science School)

INI, June 17-19 2006, Algeria

Nacer Boudjlida

LORIA, UHP Nancy 1 (F)

Nacer.Boudjlida@loria.fr

‘’Basics’’ of Database Security

- Technical and Organizational point of view:

1. Protecting data against
   1. Unauthorized access
   2. Unauthorized operations
   3. Watching the database activities (Auditing)
2. Operational Security: tend toward service continuity:
   1. Replication
   2. Recovery

''Basics'' of Database Security: Outline

I. Preamble:

   i. DBMS

   ii. DB applications in C/S Architectures

II. Protecting Database Access

III. Operational Security and Recovery

IV. Operational Security Thanks to Replication

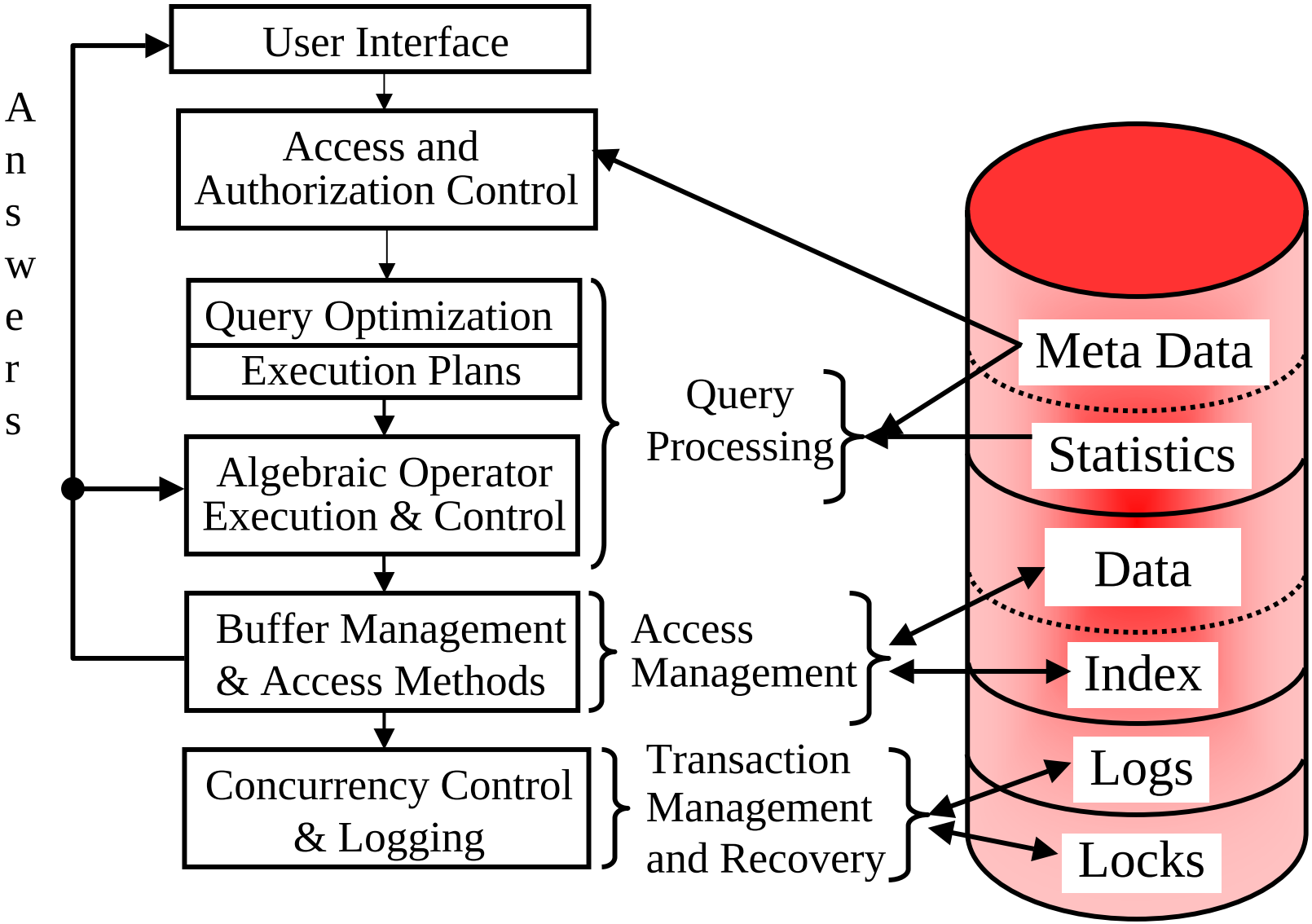
V. Database Auditing

VI. Concluding Remarks

Database Management Systems Features

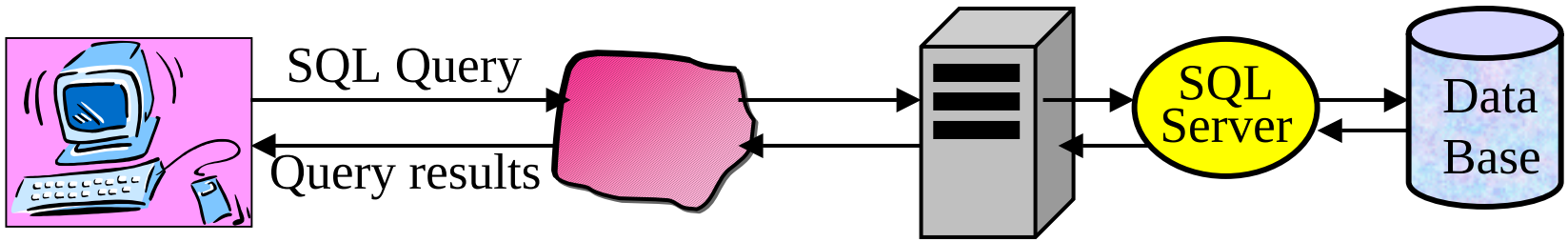
1. Data Description ("objects", attributes, relationships, constraints)
2. (High Level, Set-Oriented) Data Manipulation
3. Consistency and Privacy
4. Concurrency Control
5. Security (and Recovery from failure)

Database Management System Architecture

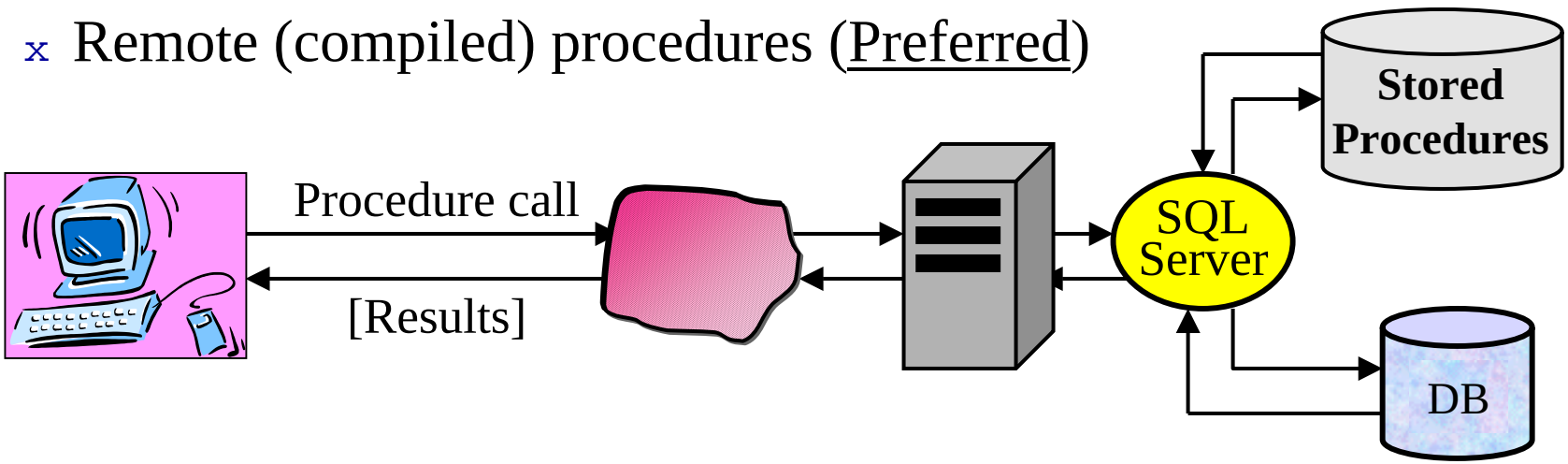


DBMSs in C/S Architectures

- Accessing Data/Object Servers
  - SQL Queries

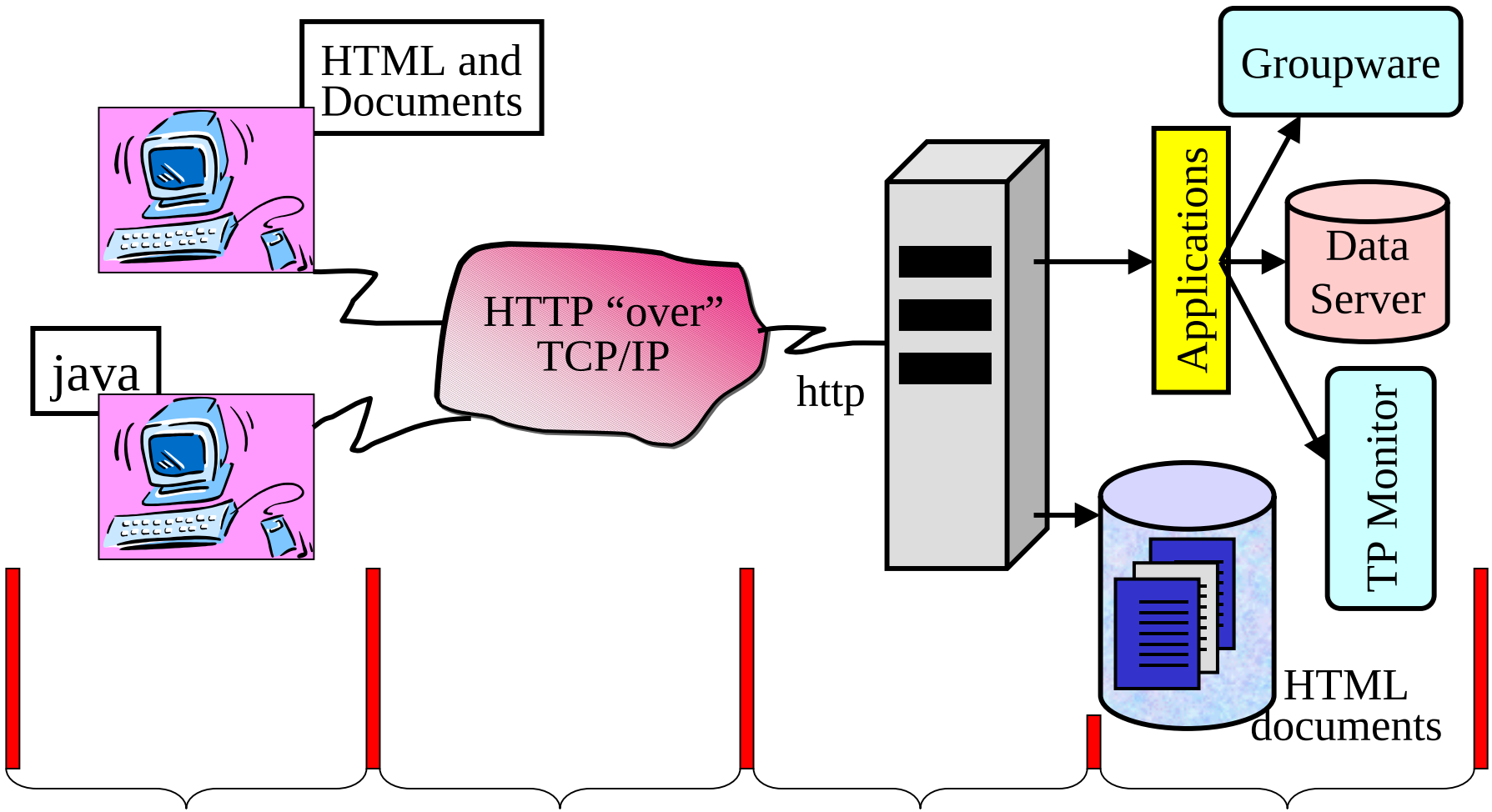


  - Remote (compiled) procedures (Preferred)



DBMSs in C/S Architectures

- Web Servers and Data/Object Servers



DBMSs in C/S Architectures: Middleware

- Middleware : the Client-Server “glue”
- General Middleware includes:
  - Communication stacks
  - Authentification services
  - Remote procedure calls
  - Message queues
  - Distributed naming services
  - etc.

DBMSs in C/S Architectures: Middleware

- Specific Middleware:
  - Database: ODBC, JDBC, DRDA, EDA/SQL
  - Transaction: TxRPC, Transactional RPC
  - Distributed Objects: CORBA, OLE/DCOM
  - Internet: HTTP, S(ecure)-HTTP

''Basics'' of Database Security: Outline

1. Preamble:
    1. DBMS
    2. DB applications in C/S Architectures
2. Protecting Database Access
3. Operational Security and Recovery
4. Operational Security Thanks to Replication
5. Database Auditing
6. Concluding Remarks

Data Integrity and Privacy

- *Privacy*:
  - Encryption, Delusions
  - Login, Password,
  - Views
  - Priviledges, Roles and Profiles
- *Identification Levels*
  1. Operating/Networking System
  2. DBMS
  3. [Database(s)]

''Basics'' of Database Security: Outline

1. Preamble:
    1. DBMS
    2. DB applications in C/S Architectures
2. Protecting Database Access
3. Operational Security and Recovery
4. Operational Security Thanks to Replication
5. Database Auditing
6. Concluding Remarks

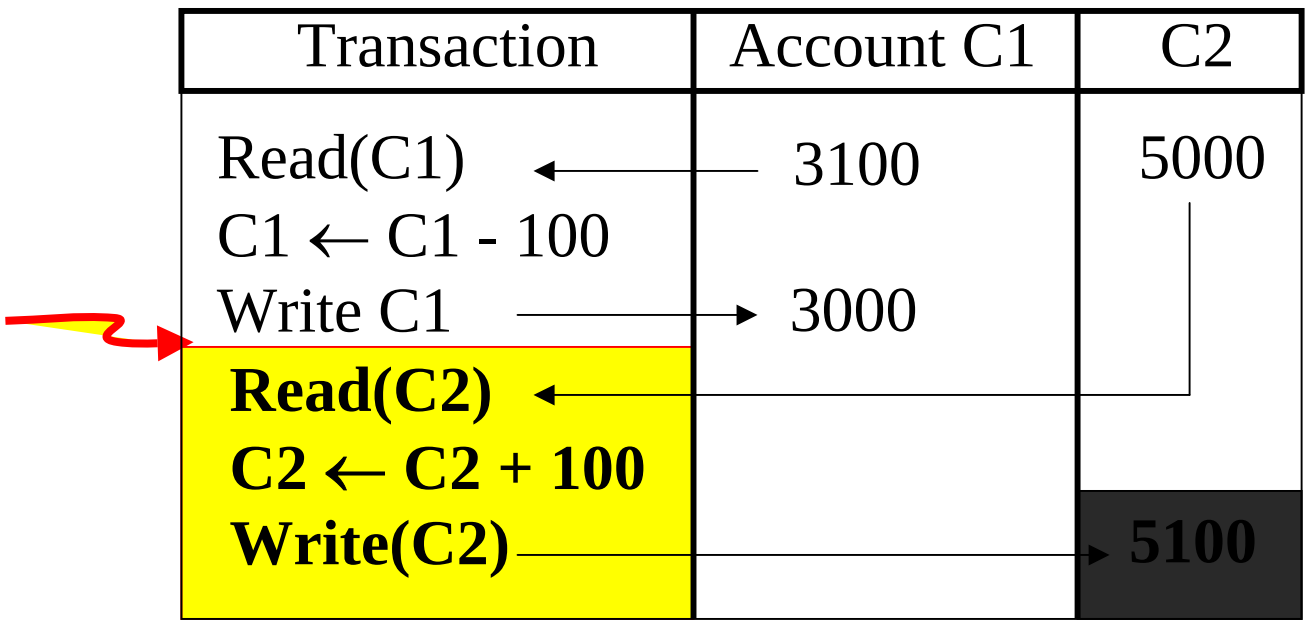
Operational Security and Recovery

- Ensure fast recovery from
  - Program failure
  - DBMS failure
  - Operating System failure
  - etc.
- Tightly Coupled with Concurrency Control
- Technical as well as Organizational Problem

Concurrency Control & Transaction Management

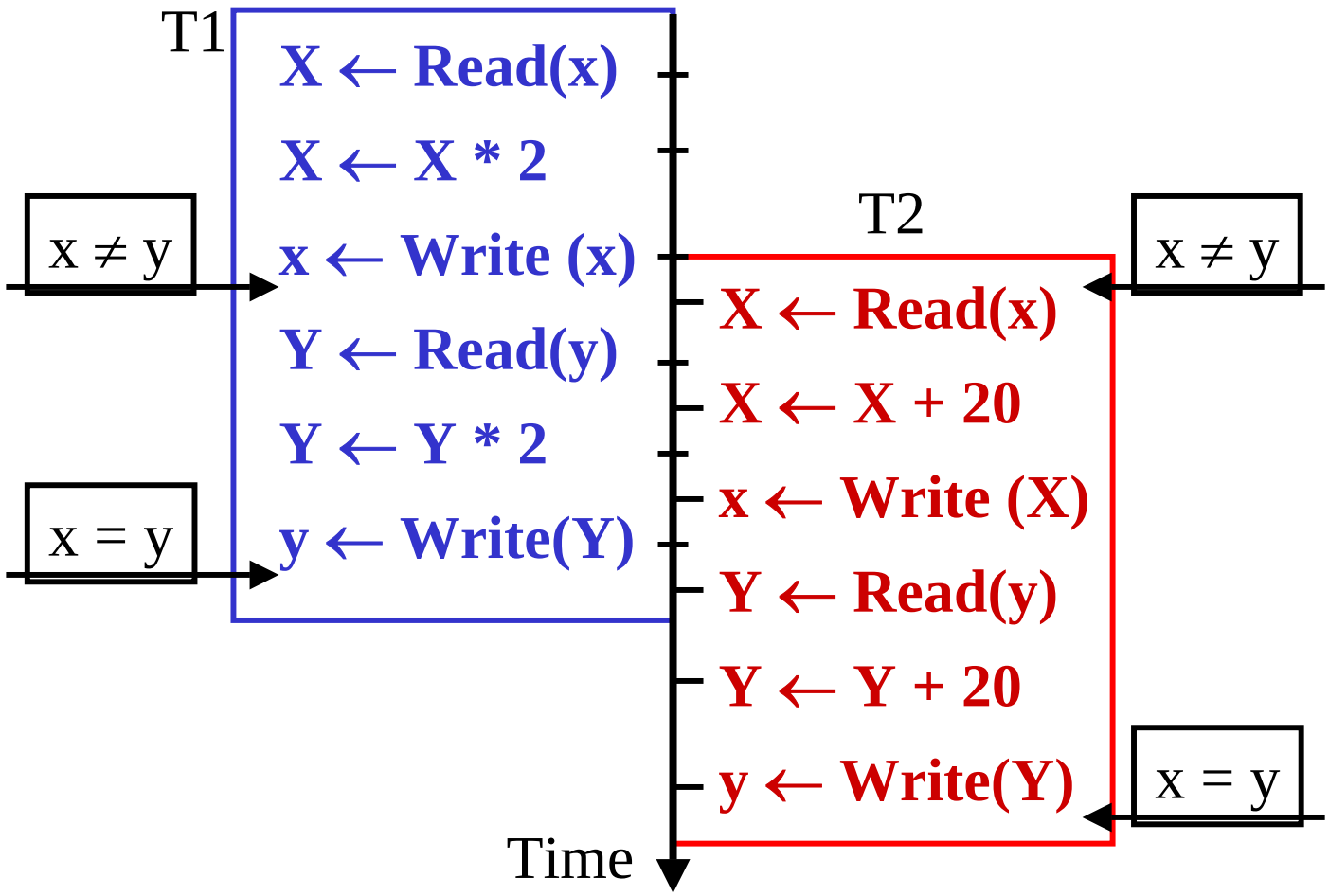
- Transaction: Atomic sequence of actions
- ACID Transaction Properties:
  - **A**tomicity
  - **C**onsistency
  - **I**solation
  - **D**urability (Skip details ?)
- Recovery: Transaction logs (Transaction history)
  - *Undo*: "Play" the transaction backward (Rollback)
  - *Redo*: "Play" the transaction forward

- Atomicity: Money transfer from C1 to C2
  - Withdraw from C1 (C1 ← C1 - 100)
  - Deposit into C2 (C2 ← C2 + 100)
  - Problem after Withdrawal and before Deposit



Transactions (cont'd): Isolation Levels

- Consistency: Constraint (x = y)



Transactions (cont'd): Isolation Levels

- **I**solation: Withdrawal and Deposit (same account)
- T1: {X ← Read(C); X ← X - 200; C ← Write(X)}
- T2: {Y ← Read(C); Y ← Y + 600; C ← Write(Y)}
- Concurrent execution of T1 and T2 (Dirty Reads)
  - **T1**: X ← Read(C): *X ← 5000*
  - **T1**: X ← X - 200: *X ← 4800*
  - T2: Y ← Read(C): *Y ← 5000*
  - T2: Y ← Y + 600: *Y ← 5600*
  - T2: C ← Write(Y): *C ← 5600*
  - **T1**: C ← Write(X): *C ← **4800*** (*instead of 5400*)

Transaction (cont'd): Isolation Levels

- **D**urability: Permanent database updates

1. (Non) Repeatable Reads
   - C in Database = 5000
   - **T1**: Read(C) → X: X ← 5000
   - T2: Read(C) → Y: Y ← **5000**
   - **T1**: X ← X + 5000: X ← 10 000
   - **T1**: Write(X) → C: C ← 10 000
   - T2: Read(C) → Y: Y ← **10 000**

- Different values !!!

Transaction (cont'd): Isolation Levels

- **D**urability: Permanent database updates (cont'd)

2. (Non) Shadow tuples

  - **T1**: Read(X) such that X = 500
  - T2: Read(X) such that X = 500
  - **T1**: Delete(X)
  - T2: Read(X) such that X = 500

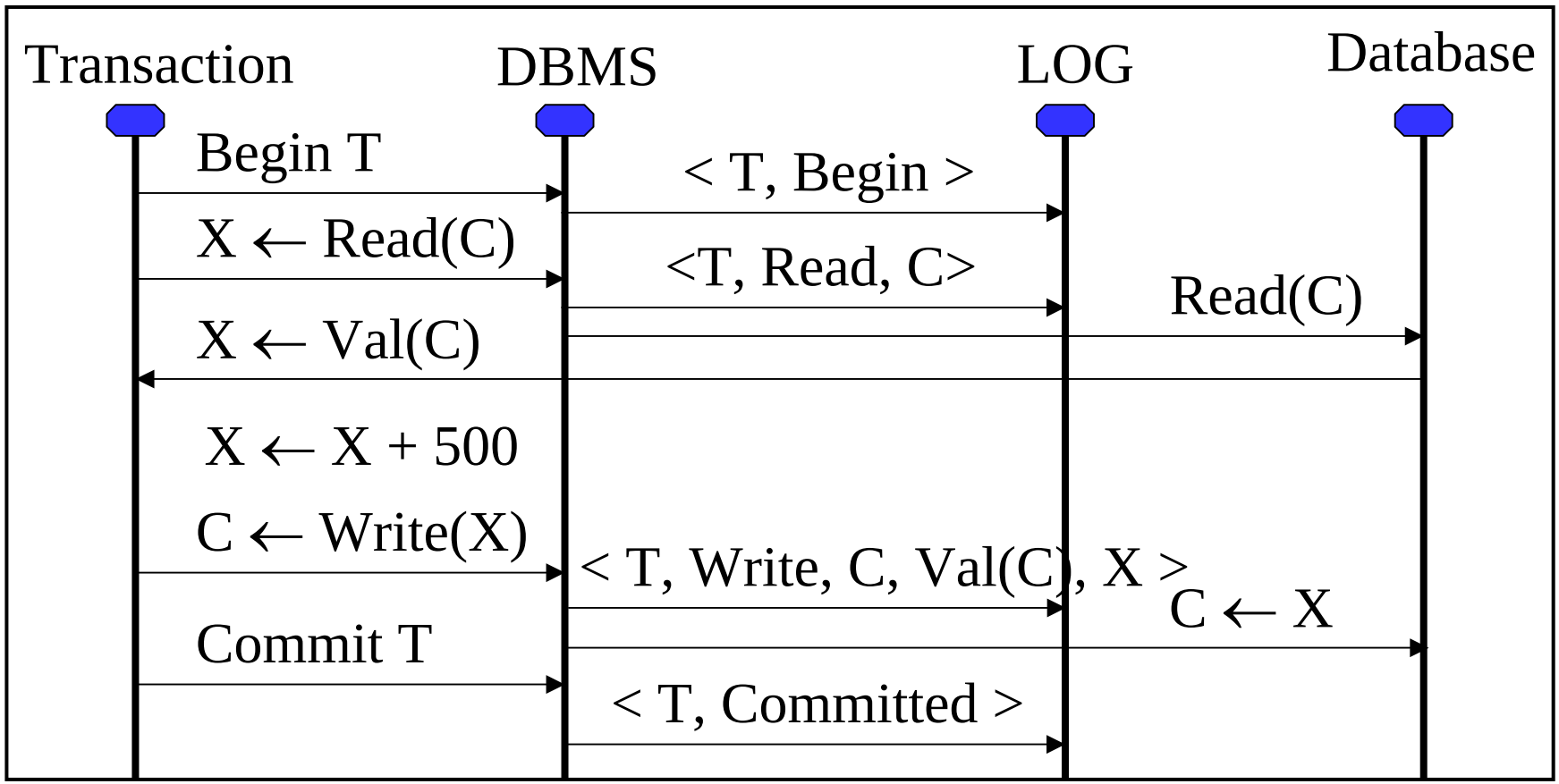
- No X, such that X = 500, exists anymore

Transaction (cont'd): Isolation Levels (end)

- Introduced in SQL'92
- *Level 0*: Dirty reads permitted
  - While a data item `D` is being updated by T1
  - Any other transaction can read `D` but it CANNOT modify it
- *Level 1*: Dirty reads not allowed
- *Level 2*: Reads repeatability not allowed before commit
- *Level 3*: Inhibits shadow tuples

Transaction Loging and Recovery

- Write Ahead Loging Protocol (WAL)
- Rollback: 'Execute' Transaction Log backward



Transaction Management and Recovery (cont'd)

- Concurrency Control and Serialisability: Concurrent execution is equivalent to sequential (serial) execution
- Foundations: partial ordering of actions
- Techniques
  - Time stamping (not usually implemented)
  - Two-Phase Locking (ensures Serialisability)
    - Phase1: Acquire Locks
    - Phase2: Free Locks
    - Does not prevent from deadlocks or livelocks

Transaction Management and Recovery (cont'd)

- Most Usual Types of locks:
  - **S**hared: multiple reads and no update
  - E**X**clusive: single update
  - **I**ntention locks: **R**ead, **W**rite, **U**pdate

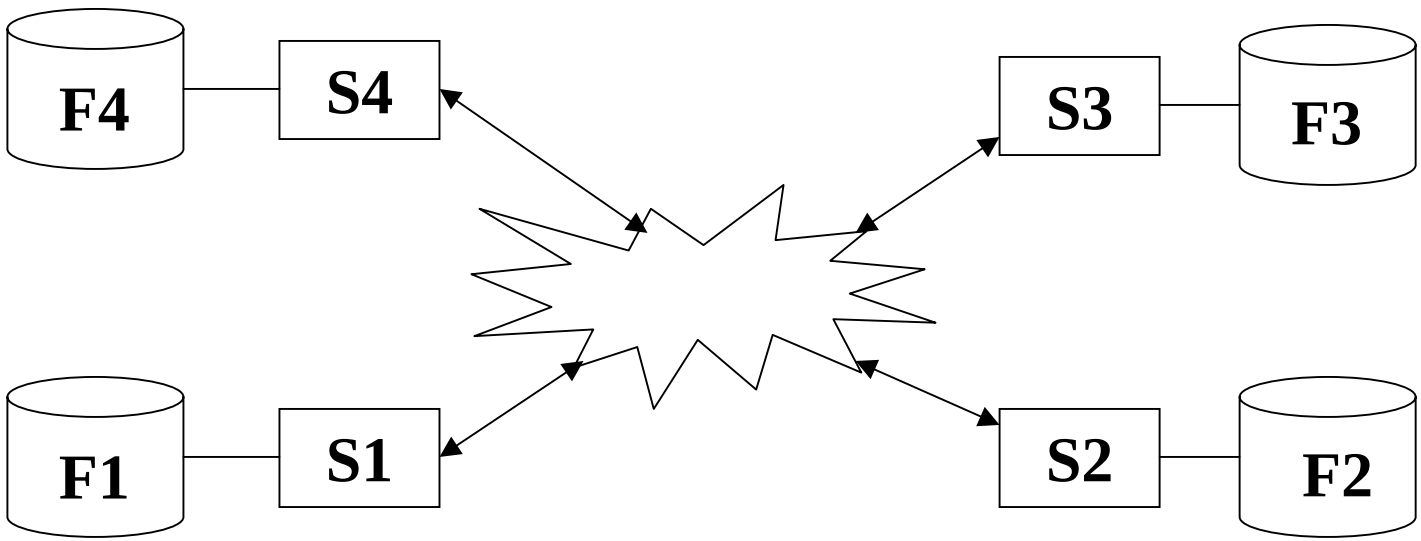
Lock mode / Requested Lock Mode	**S**	**X**
S	Yes	No
X	No	No

	R	**U**	**W**
R	Yes	No	No
U	Yes	No	No
W	No	No	No

- cf. Select … for update/at isolation level serializable/holdlock

Note: Distributed Transaction Management

- Distributed DB:
  - DB = « Union » of « Fragments »

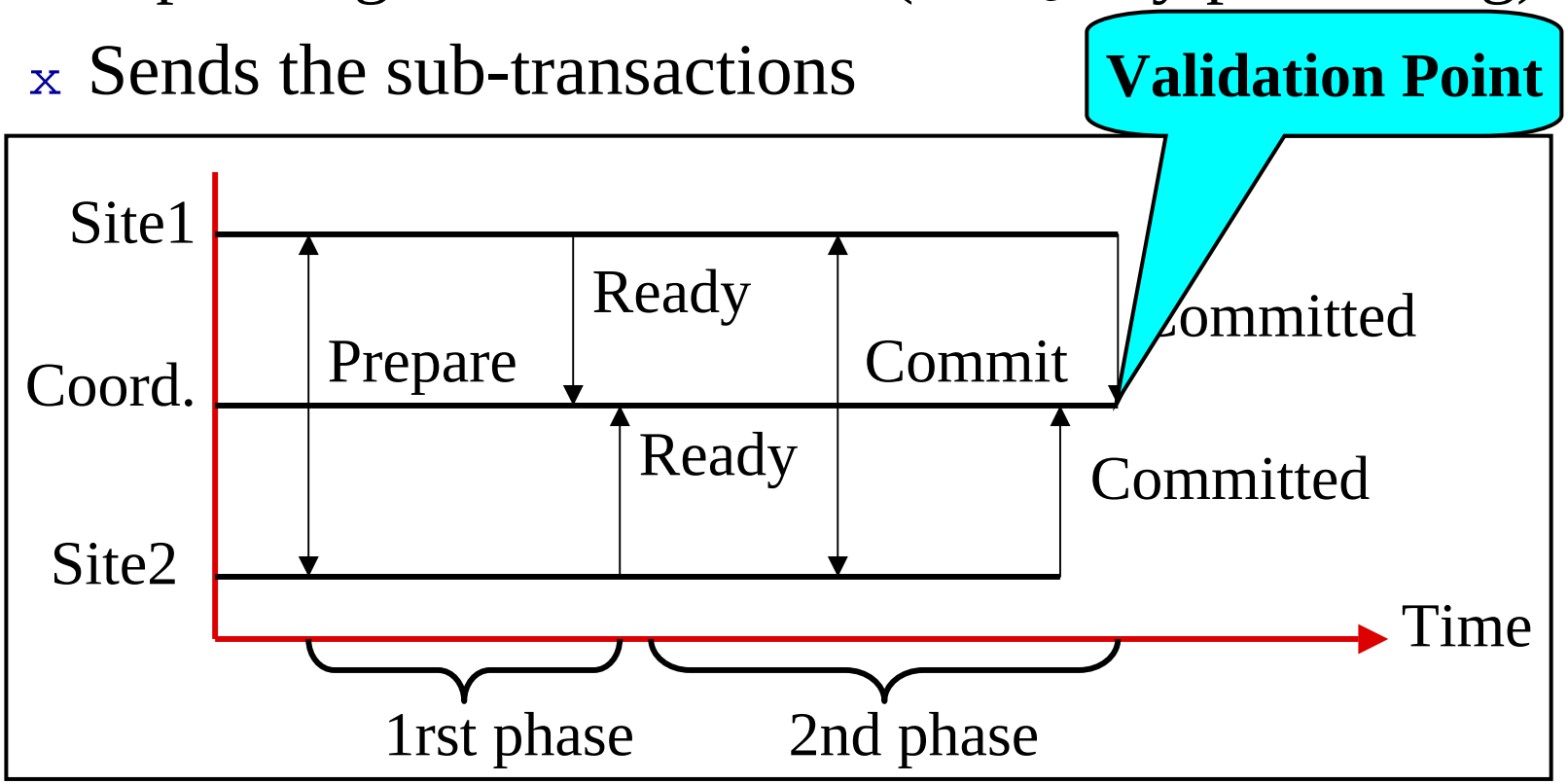


Note: Distributed Transaction Management

- Context:
  - Multiple sites cooperate in (sub-)transactions
  - Commit/Rollback on every site
- *Two-Phase Commit Protocol*
  - Lampson [1976], Gray [1978]
  - Most commonly implemented
  - Ensures transaction **A**tomicity [Baer & al., ICSE81]
  - Requires a coordination site

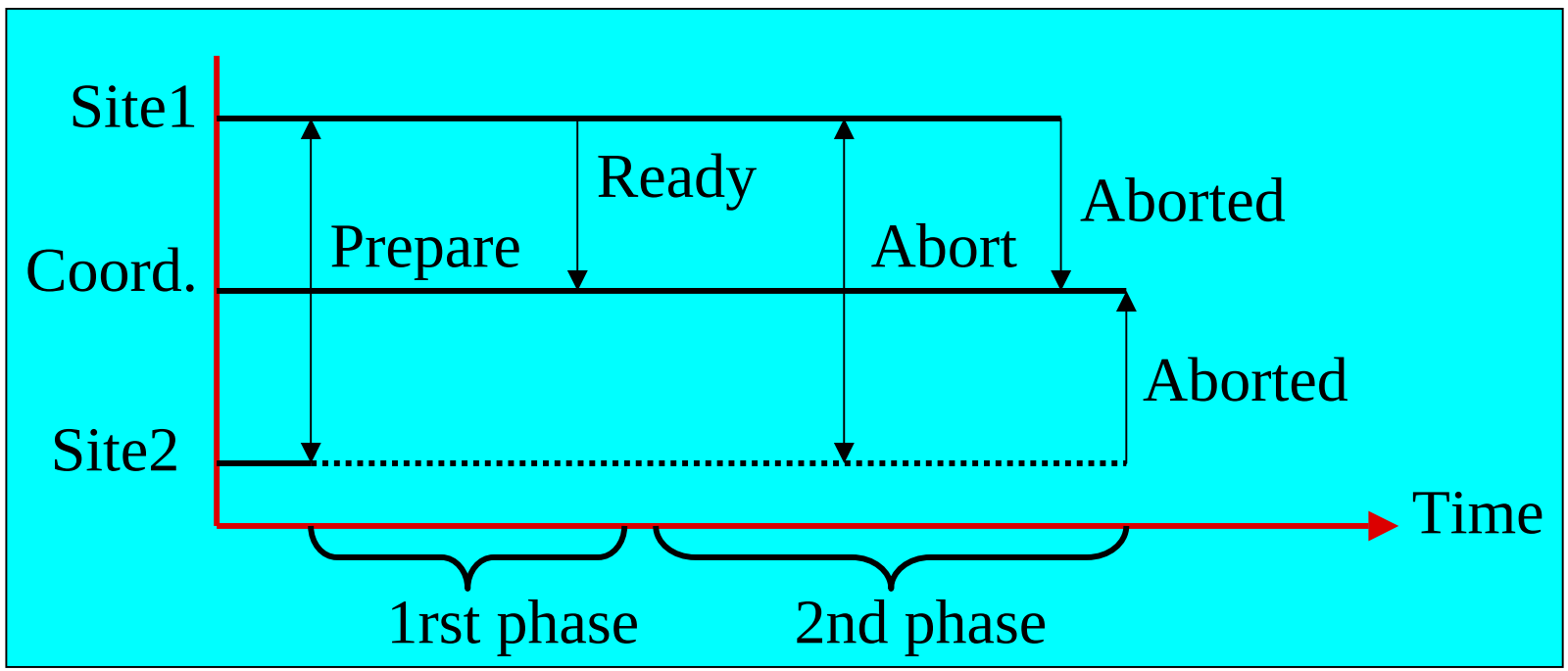
Distributed Transaction Management (cont'd)

- Coordination site:
  - Decomposes a transaction into sub-transactions depending on data location (cf. Query processing)
  - Sends the sub-transactions



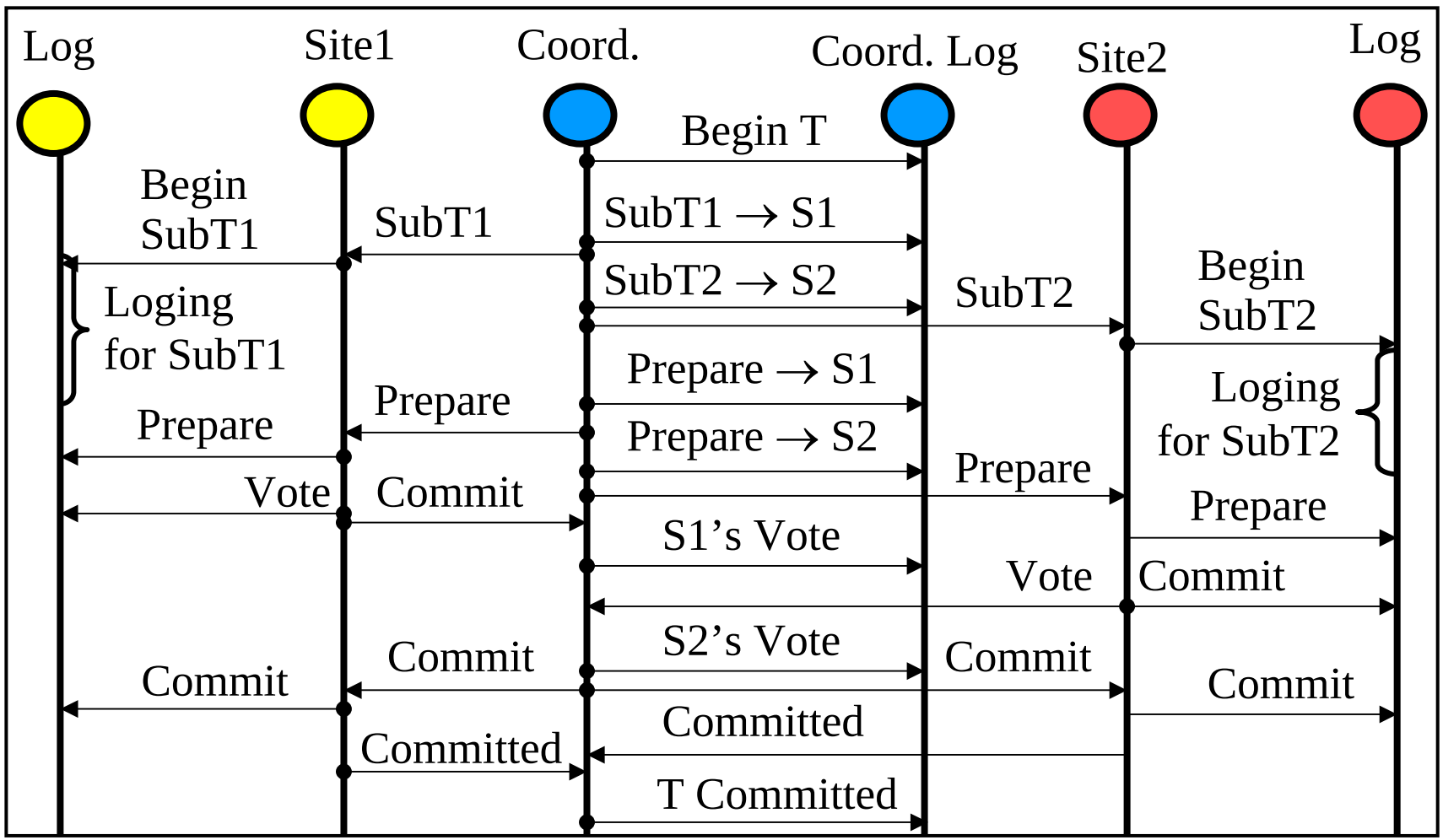
Distributed Transaction Management (cont'd)

- Transaction failure:
  - One or more sites cannot commit
  - One or more answers to the prepare message are missing...



Distributed Transaction Management (cont'd)

- Transaction Logs: (Usually) WAL Protocol

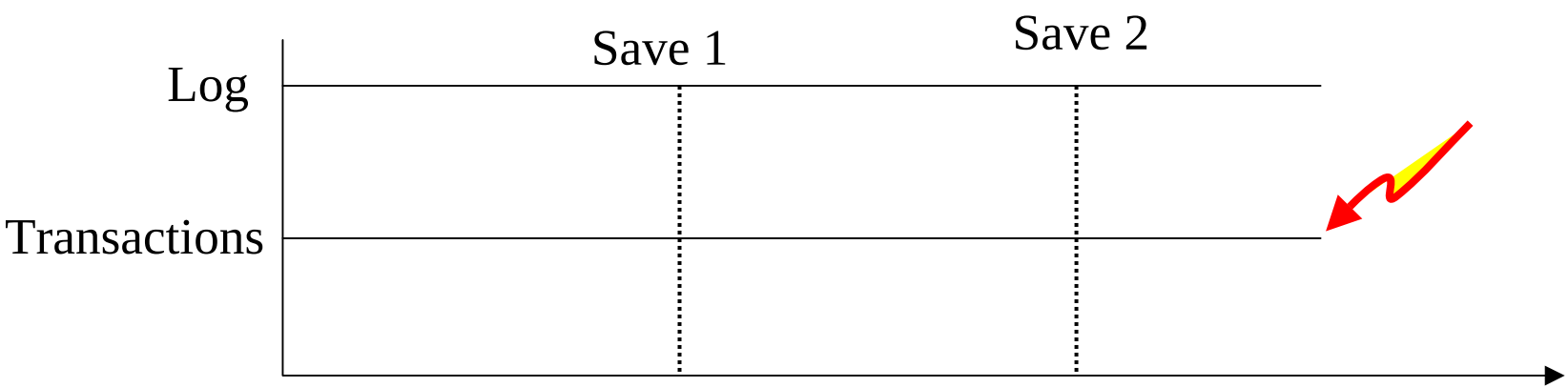


Transaction Management (end)

- Serialisability: Serial execution equivalent to concurrent execution
- ACID properties
- Mechanisms:
  - Concurrency Manager
  - Integrity Manager
  - Recovery Manager
  - Programmers
  - Security Officer [+ DB Administrator]

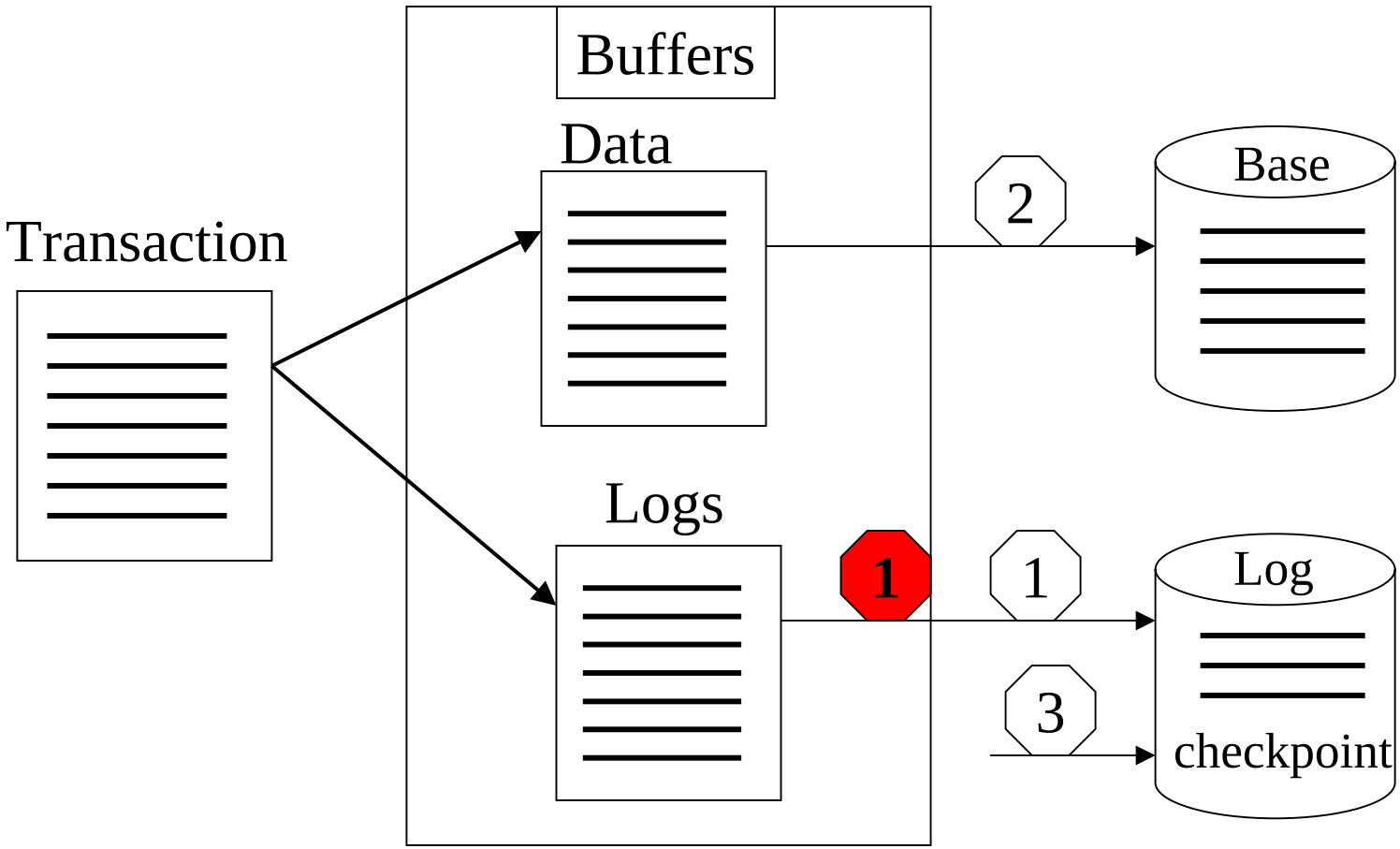
Log Management and Recovery

- Log management:
  - Configuration
    - Database creation
    - During database « life »
  - Periodic Save
  - [Restore, when needed]



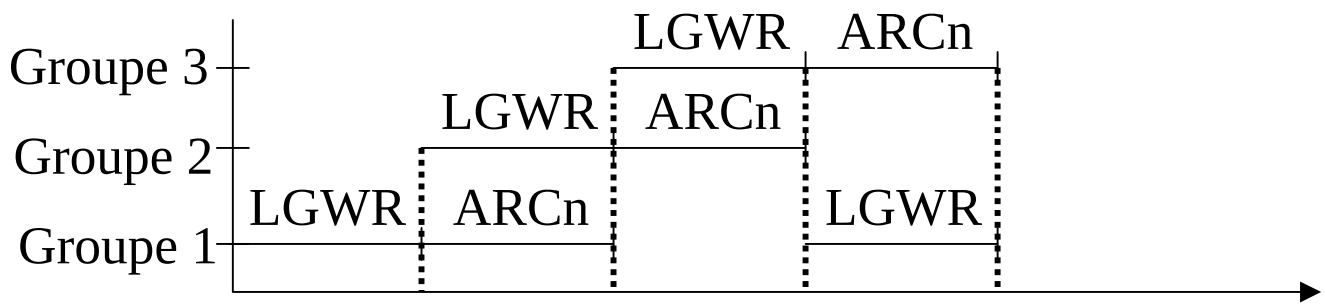
Log Management and Recovery

- Commit vs Checkpoint
  - Data Server configuration (*checkpoint frequency*)



Log Management and Recovery in Oracle

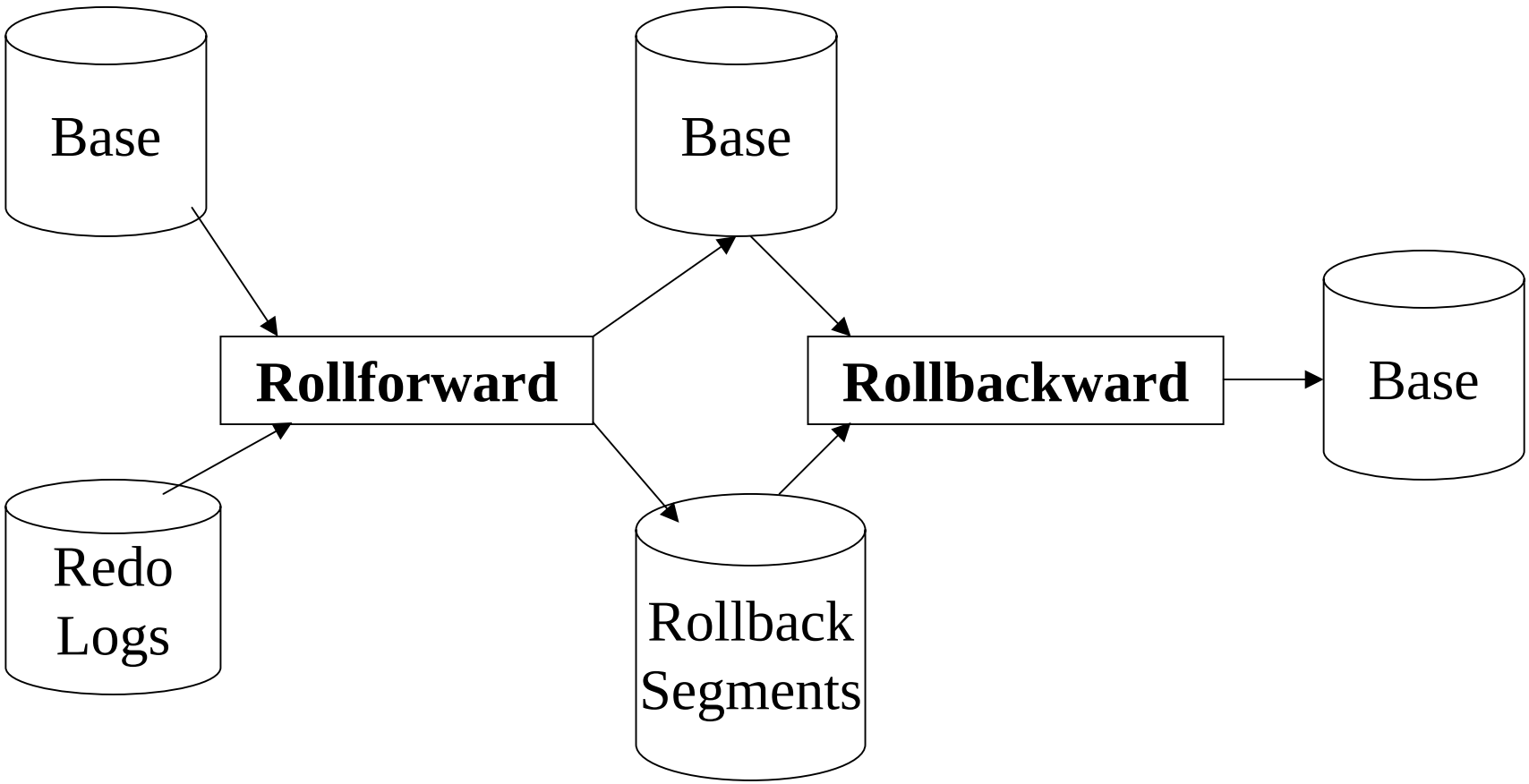
- Checkpoint: configuration parameters
  - *Log-checkpoint_interval* (number of blocks)
  - *Log_checkpoint_timeout* (in seconds)
- Loging:
  1. Redo logs
  2. Rollback segments/Undo Tablespaces
- Automatic Archive Logs



Log Management and Recovery in Oracle

- Back-Up and Recovery
  - Total/Partial
- Recovery
  - Rollforward (*cache recovery*)
  - Rollbackward (*transaction recovery*)

Log Management and Recovery in Oracle



''Basics'' of Database Security: Outline

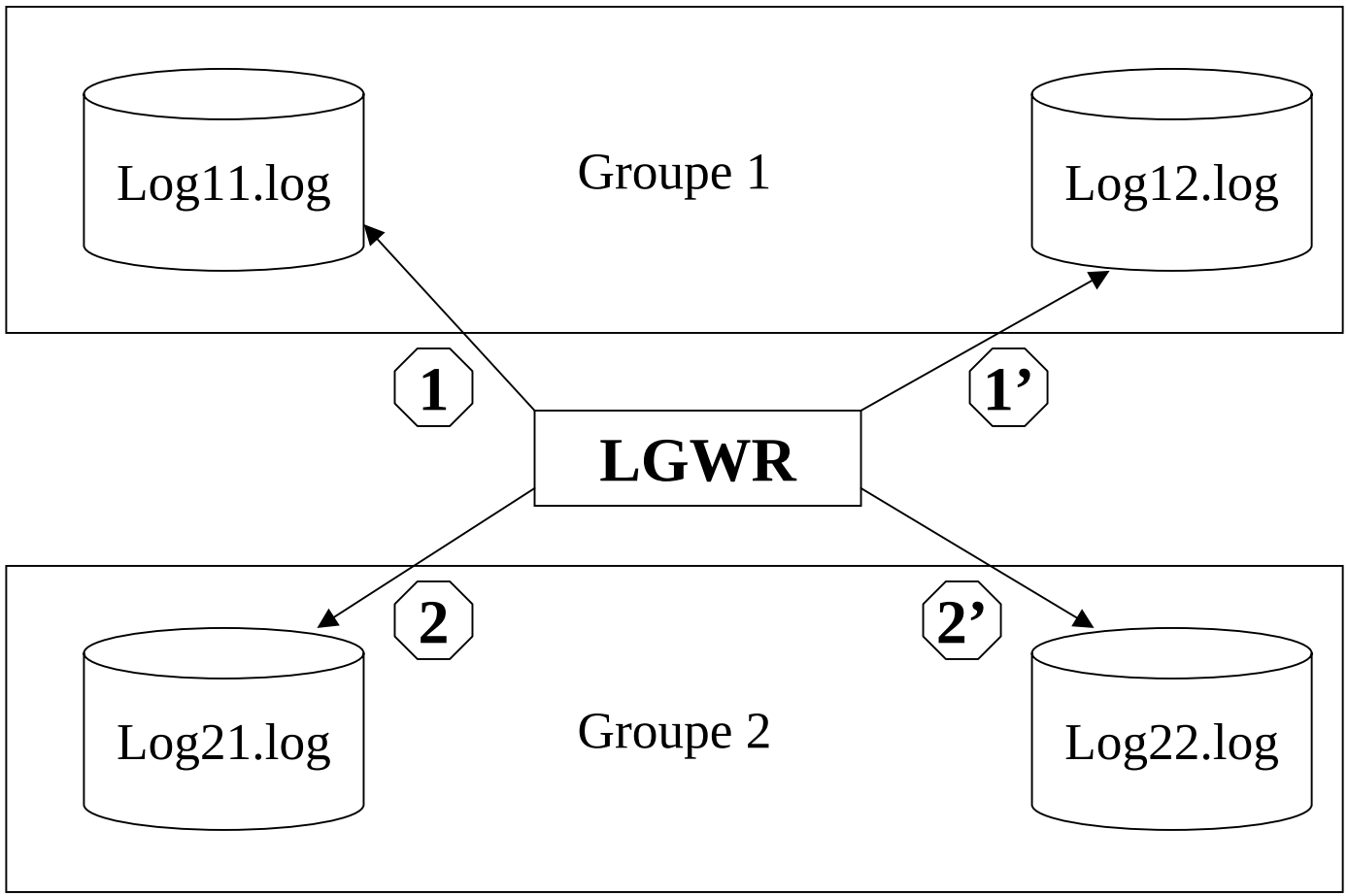
1. Preamble:
    1. DBMS
    2. DB applications in C/S Architectures
2. Protecting Database Access
3. Operational Security and Recovery
4. **Operational Security Thanks to Replication**
5. Database Auditing
6. Concluding Remarks

Security thanks to Replication

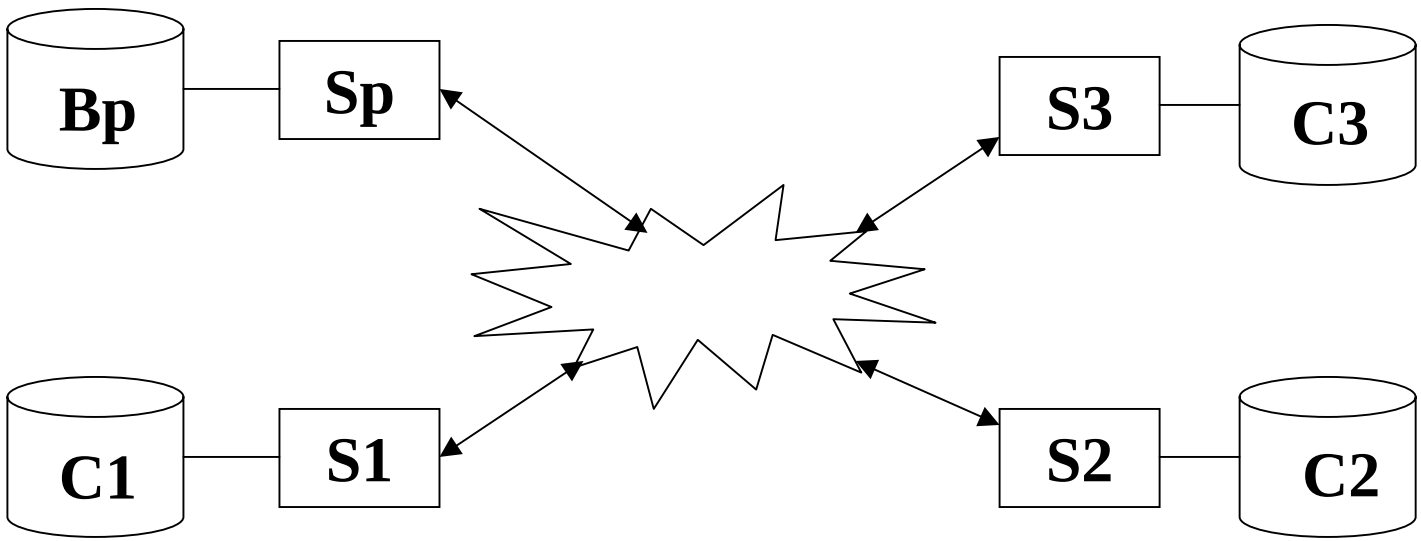
- Synchronize the content of a storage unit with its replicate
- Switch in case of failure
- Replicate
  - At least the logs
  - And/or sensitive data
- Requires more space
- Activity overhead (may conflict performance issues)

Security thanks to Replication

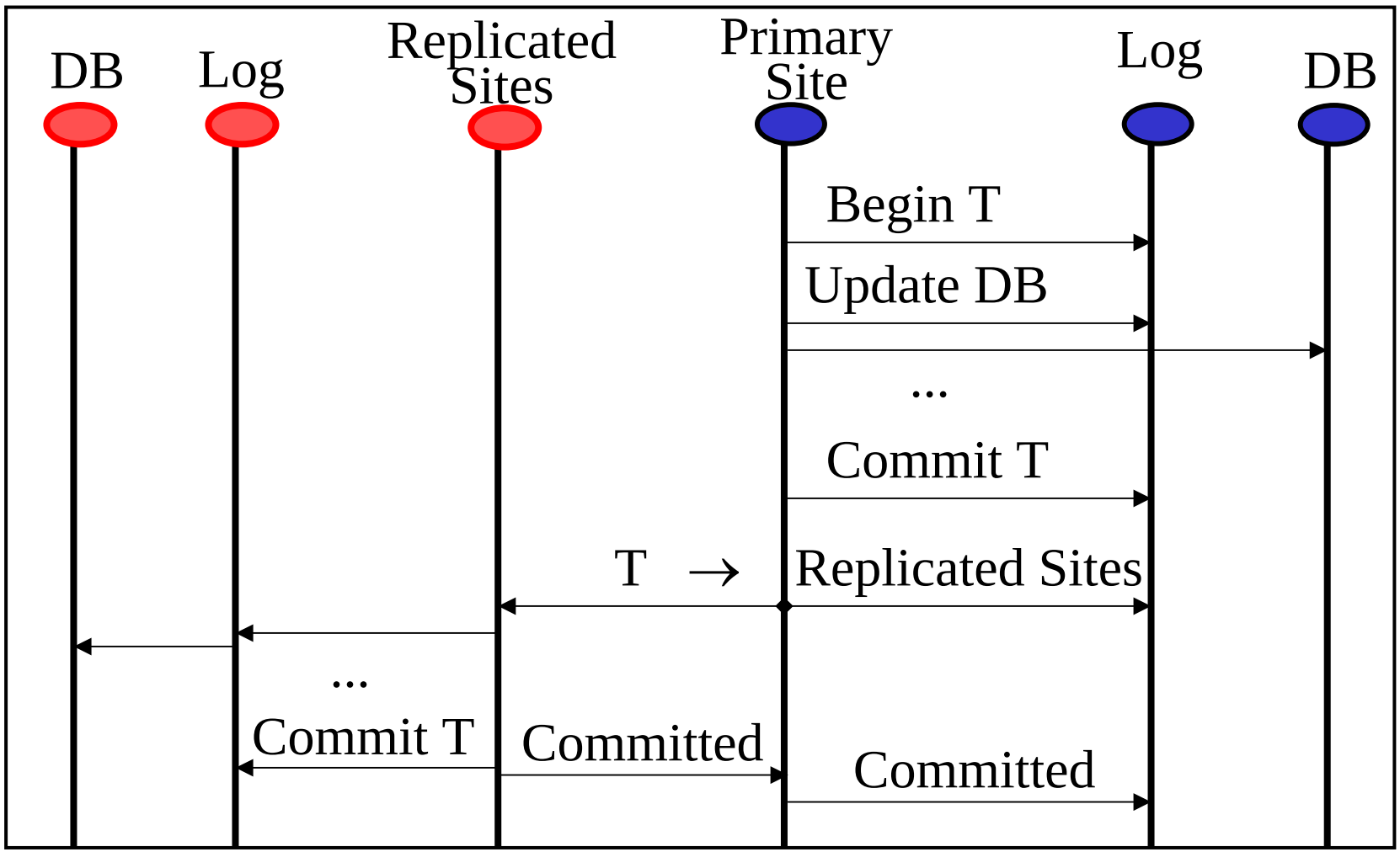
- Example: Oracle Multiplexing



- Case of Replication Servers with primary copy



- Case of Replication Servers with primary copy



- Case of Replication Servers with primary copy
  - Case of primary site failure
    - Elect an other one
  - Case of secondary site failure
    - Exclude it during the failure
    - Re-synchronization (Logs)

''Basics'' of Database Security: Outline

1. Preamble:
   1. DBMS
   2. DB applications in C/S Architectures
2. Protecting Database Access
3. Operational Security and Recovery
4. Operational Security Thanks to Replication
5. **Database Auditing**
6. Concluding Remarks

Auditing Services

- Record events (*audit options*) about
  - Database(s)
  - User(s)
  - Data Server(s)

- Process:
  1. Install or Configure the Audit Mechanism
  2. Define the audit options
  3. Manage the audit trail

Auditing Services in Oracle

1. *Install or Configure the Audit Mechanism*
   - *audit_trail* = true (init file)
   - sys.aud$ exists

2. *Define the audit options*
   1. Per session/per access
   2. Whenever successfull/not successfull
   3. Per type of SQL statements
   4. Per type of system priviledge commands
   5. Per type of action on database objects

Auditing Services in Oracle: Examples

- *audit* select on U1.MaTable whenever not successfull
- *audit* select table, update table by U1, U2
- *audit* role whenever successfull
- *audit* all privileges by U1, U2
- *audit* insert, delete, update on sys.aud$ per access
- *noaudit* select table by U1, U2

Auditing Services in Oracle

3. *Manage the audit trail*

   a) Fix the right size (create/alter table)

   b) Explore
      - Dictionary tables and views
      - *dba_stmt_opts*, *all_def_audit_opts*, etc.

   c) Save or purge

Concluding Remarks

- DB security:
  - Tools and mechanisms exist
  - But, it's not only a matter of DB technology
    - Platform
    - Network
    - Global enterprise security: policy and rules
- New challenges:
  - Ubiquituous/mobile computing
  - Replicated Mobile Data
  - Web-based computing

The (Very) Basics of Database Security

That's all Folks !

Thank you for paying attention.

Any questions ?