HỌC VIỆN KỸ THUẬT MẬT MÃ
KHOA AN TOÀN THÔNG TIN

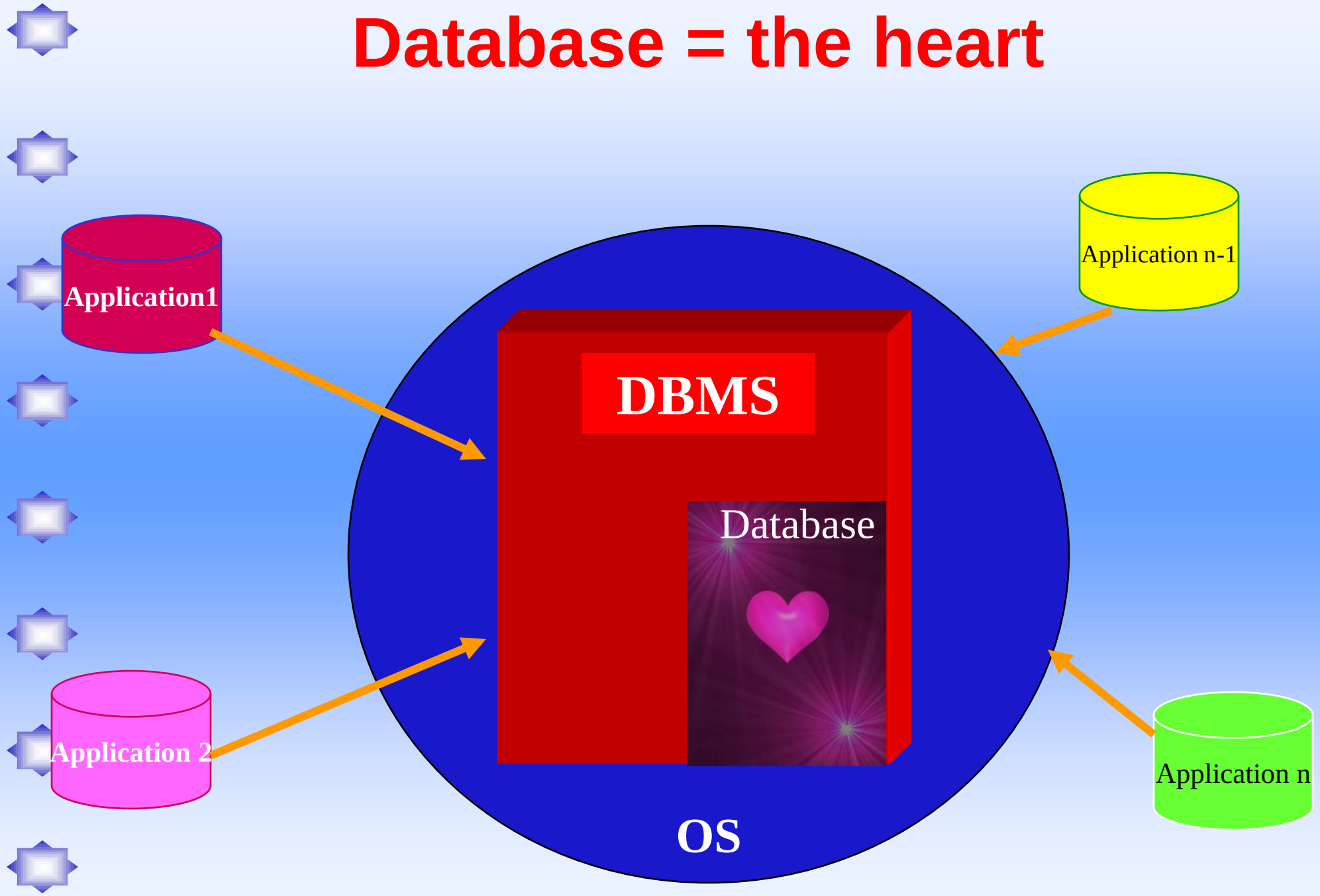
CHƯƠNG 3
AN TOÀN TRONG DBMS

Giảng viên: TS. Trần Thị Lượng

Nội dung

1. So sánh DBMS và OS
2. Các kiến trúc DBMS an toàn
3. Giới thiệu một vài DBMS
4. Các vấn đề an toàn chung trong DBMS

Database = the heart



So sánh DBMS và OS

- Câu hỏi:
  - *Sự khác nhau giữa OS và DBMS?*

DBMS vs OS

- ***Độ chi tiết của đối tượng (Object granularity):***
  - OS: độ chi tiết ở mức tệp (file), thư mục, thiết bị.
  - DBMS: chi tiết hơn tới table, rows, fields, entry.
- ***Các tương quan ngữ nghĩa trong dữ liệu (Semantic correlations):***
  - OS: không có.
  - DBMS: dữ liệu có ngữ nghĩa và liên quan với nhau thông qua các quan hệ ngữ nghĩa như:
    - Data
    - Time
    - Context
    - History

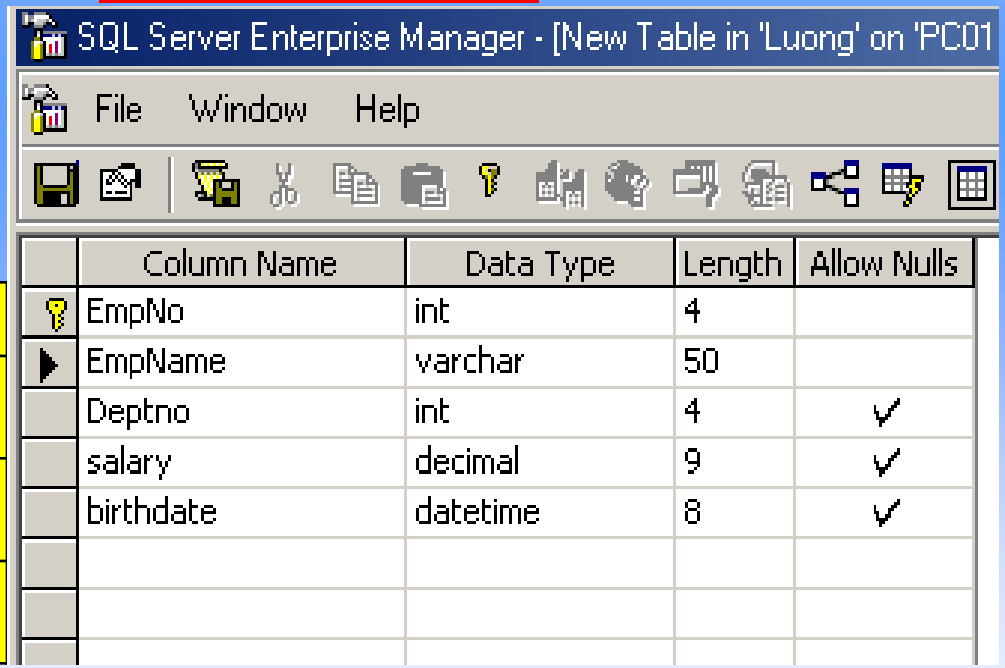
DBMS vs OS (…)

- ***Siêu dữ liệu (Metadata):***
  - OS: không có
  - DBMS: siêu dữ liệu cung cấp thông tin về cấu trúc của dữ liệu như: table, view, rows, fields, …

Data instance

EmpNo	EmpName	DeptNo	Salary	Birthdate
1	Lan	203	2000	03/28/1970
2	Minh	104	3000	12/11/1982
3	Huyền	300	3500	10/30/1983

metadata



DBMS vs OS (…)

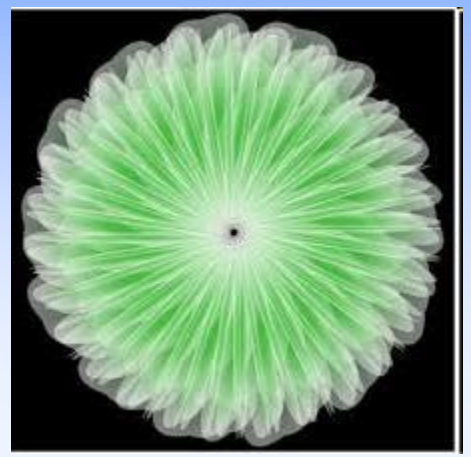
- ***Các đối tượng logic và vật lý***:
  - OS: chứa các đối tượng vật lý như: file, memory, process, devices….
  - DBMS: chứa các đối tượng logic như: table, view, index, column, rows, entry…và chúng độc lập với các đối tượng của OS.

DBMS vs OS (…)

- ***Multi-datatypes***:
  - OS: có các truy nhập vật lý: như Read, write, execute…
  - DBMS: có rất nhiều kiểu dữ liệu, do đó các CSDL cũng yêu cầu nhiều chế độ truy nhập như: chế độ thống kê, chế độ quản trị, select, insert, update, delete…

DBMS vs OS (…)

- ***Các đối tượng động và tĩnh***:
  - OS: quản lý các đối tượng tĩnh và tương ứng với các đối tượng thực.
  - DBMS: quản lý cả các đối tượng có thể được tạo ra động như: views hay SQL query và không có các đối tượng thực tương ứng.

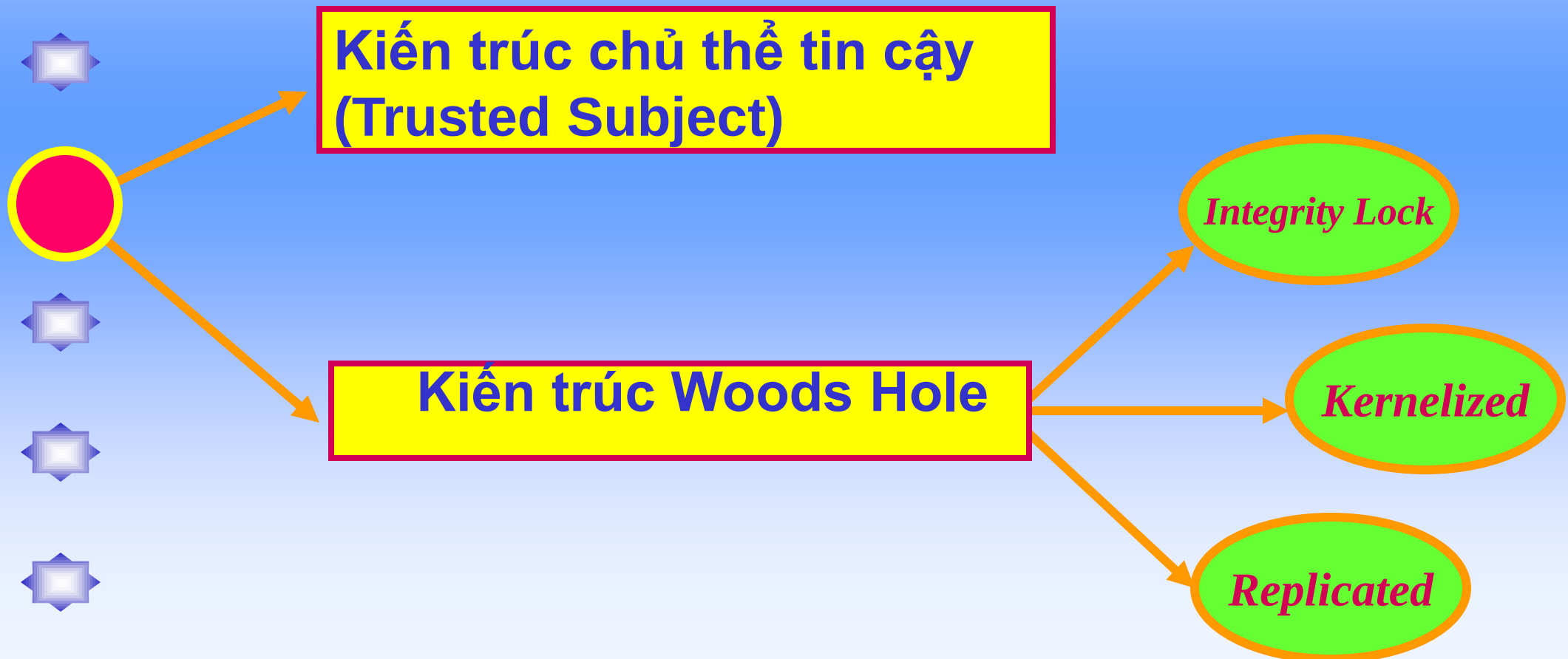


Nội dung

1. So sánh DBMS và OS
2. *Các kiến trúc DBMS an toàn*
3. Giới thiệu một vài DBMS
4. Các vấn đề an toàn chung trong DBMS

Các kiến trúc DBMS an toàn

- ***Hai kiến trúc cơ bản:***



Các kiến trúc của DBMS an toàn

Bảng 3.1 Các kiến trúc mẫu thử DBMS và các sản phẩm thương mại

Kiến trúc	*Các mẫu thử nghiên cứu*	*DBMS thương mại*
Integrity Lock	Mitre	TRUDATA
Kernelized	Sea View	Oracle
Replicated	NRL	--------
Trusted Subject	A1 Secure DBMS (ASD)	Sybase
		Informix
		Ingres
		Oracle
		DEC
		Rubix

Kiến trúc chủ thể tin cậy (Trusted Subject)

Đặc điểm:

- Giả thiết DBMS và một OS tin cậy.
- DBMS hoạt động như là một chủ thể tin cậy của OS
- DBMS có trách nhiệm trong việc bảo vệ đa mức (multilevel) các đối tượng của CSDL.
- Được sử dụng trong nhiều DBMS thương mại (Sybase, Informix, Ingres, Oracle, DEC, Rubix).

Trusted Subject

High user

Low user

Untrusted front end

Untrusted front end

Trusted DBMS

Phân tích, tối ưu câu truy vấn

Bảo vệ nhiều mức

Trusted OS

Database

User, Object được gắn nhãn

Trusted Subject (…)

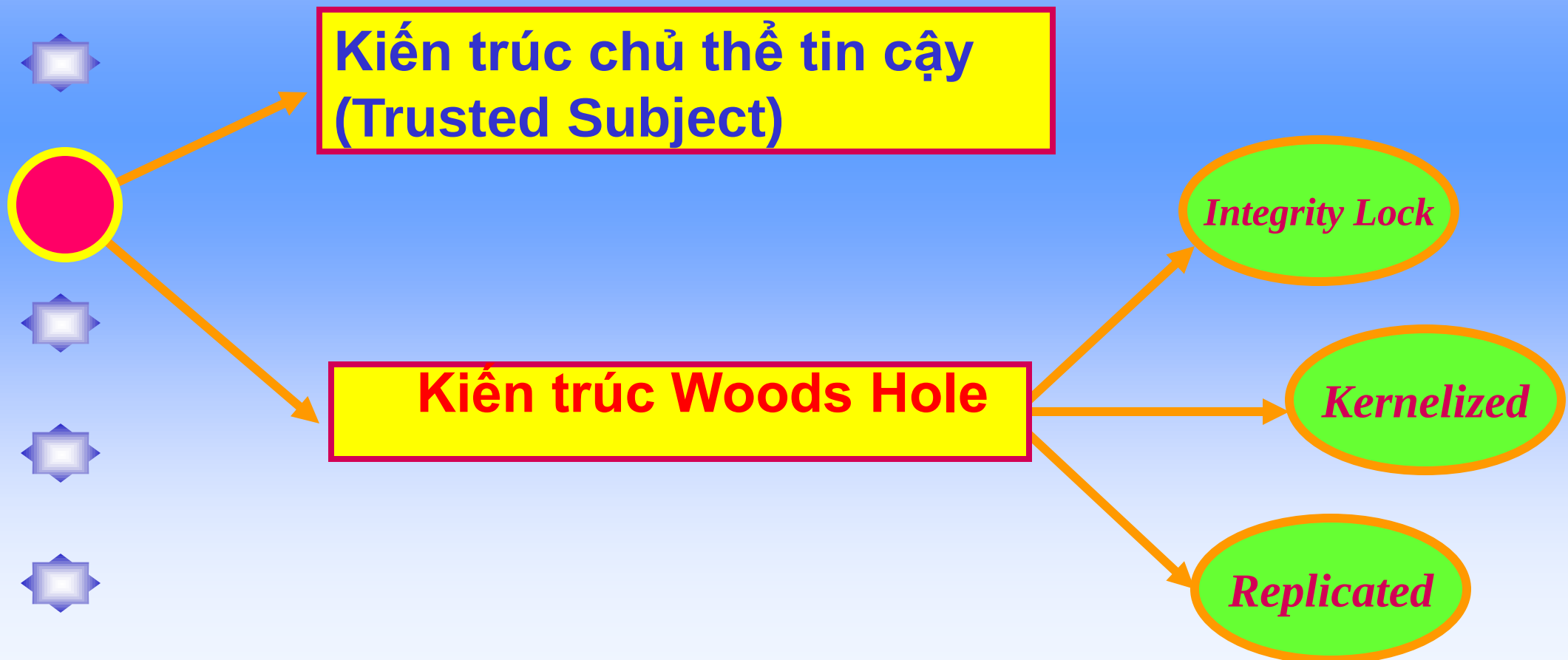
- Người dùng kết nối tới DBMS qua các phần mềm *untrusted front end* (vì họ kết nối qua Internet).
- Người dùng được phân loại các mức nhạy cảm khác nhau: High (cao), Low (thấp), và một mức DBMS khác với hai mức trên.
- Các chủ thể và đối tượng được gán một nhãn DBMS không giống với mức High và Low.
- Chỉ có các chủ thể được gán nhãn DBMS mới được phép thực hiện mã lệnh và truy nhập vào dữ liệu.
- Các chủ thể có nhãn DBMS được coi là các chủ thể tin cậy và được miễn kiểm soát bắt buộc của OS

Trusted Subject (…)

- Các đối tượng CSDL được gán nhãn nhạy cảm (ví dụ: các bộ, các giá trị).
- Hệ quản trị Sybase tuân theo giải pháp này, với kiến trúc máy khách/máy chủ, Sybase thực hiện gán nhãn mức bản ghi (mức hàng).

Các kiến trúc của DBMS an toàn

- ***Hai kiến trúc cơ bản:***



Các kiến trúc Woods Hole

- Các kiến trúc Woods Hole sử dụng DBMS không tin cậy cùng với một bộ lọc tin cậy và không quan tâm đến OS có tin cậy hay không.
- Được phát triển năm 1982 bởi National Research Council

Woods Hole

High user

Low user

Untrusted front end

Untrusted front end

Trusted front end

Bảo vệ nhiều mức

Phân tích, tối ưu câu truy vấn

Untrusted DBMS

Database

User, Object được gắn nhãn

Các kiến trúc Woods Hole

- ***Nhận xét:***
  - Phần mềm front ends và DBMS đều không tin cậy (Không quan tâm OS có tin cậy hay không)
  - Phần mềm untrusted front-end thực hiện các công việc xử lý trước và sau các câu truy vấn (phân tích, tối ưu hóa, phép chiếu).
  - Phần mềm trusted front end (TFE) ở giữa thực thi các chức năng an toàn và bảo vệ nhiều mức, vì vậy hoạt động như một TCB (Trusted Computing Base).

Các kiến trúc Woods Hole

- **Kiến trúc Integrity Lock**
- Kiến trúc Kernelized
- Kiến trúc Replicated (còn được gọi là kiến trúc Distributed)

Kiến trúc Integrity Lock

- *Khoá toàn vẹn* được đề xuất lần đầu tiên tại Viện nghiên cứu của Lực lượng Không quân về An toàn cơ sở dữ liệu [AF83], được dùng để kiểm soát *tính toàn vẹn* và sự *truy nhập* cho cơ sở dữ liệu.
- Kiến trúc Integrity lock đã có trong hệ quản trị thương mại TRUDATA.

Kiến trúc Integrity Lock

- Đặc điểm:
  - TFE thực thi bảo vệ nhiều mức bằng cách gắn các nhãn an toàn vào các đối tượng CSDL dưới dạng các *tem – Stamps.*
  - Một *tem* là một trường đặc biệt của một đối tượng, lưu thông tin về nhãn an toàn và các dữ liệu điều khiển liên quan khác.
  - *Tem* là dạng mã hóa của các thông tin trên, sử dụng một kỹ thuật niêm phong mật mã gọi là **Integrity Lock**.

High user

Low user

Integrity Lock

Untrusted front end

Untrusted front end

Trusted filter

Cryptographic unit

Append stamp

Check stamp

Gắn tem cho đối tượng CSDL

Store

Response

Untrusted DBMS

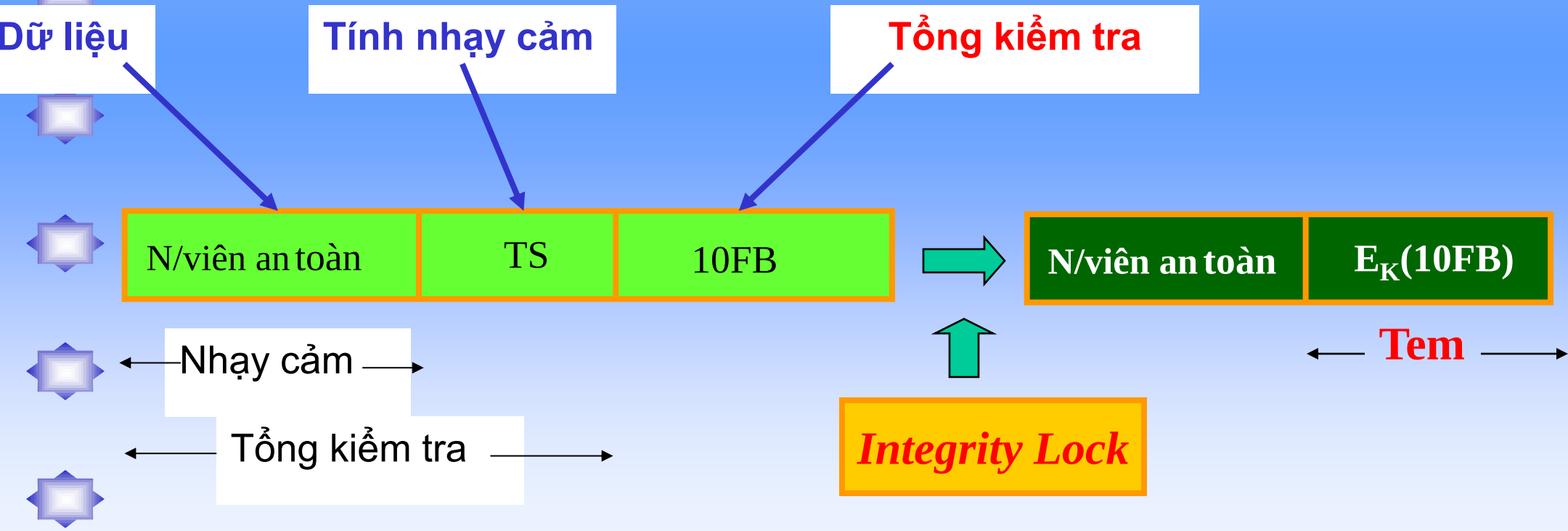
Database

Kiến trúc Integrity Lock (…)

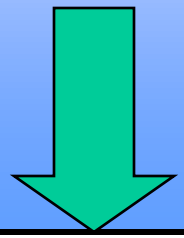
- TFE có nhiệm vụ tạo và kiểm tra các tem.
  - TFE sử dụng mật mã khóa bí mật để tạo *tem* và giải mã các tem. Các tem này có thể tạo ra dựa vào tổng kiểm tra (checksum).
  - Khóa bí mật chỉ có TFE biết.

Kiến trúc Integrity Lock (…)

- Một mô hình về khoá toàn vẹn cơ bản được chỉ ra như trên hình vẽ.



TenDuAn	NganSach	Level
DA1	100.000.000	**TS**
DA2	10.000.000	**S**



Integrity Lock

TenDuAn	**NganSach**	**Level**
DA1+ E_{K1}(DA1+TS)	100.000.000 + E_{K1}(100.000.000 +TS)	**TS**
DA2 + E_{K2}(DA2+S)	10.000.000 + E_{K2}(10.000.000 +S)	**S**

Câu hỏi

- Tính toàn vẹn nằm ở chỗ nào?

Kiến trúc Integrity Lock (…)

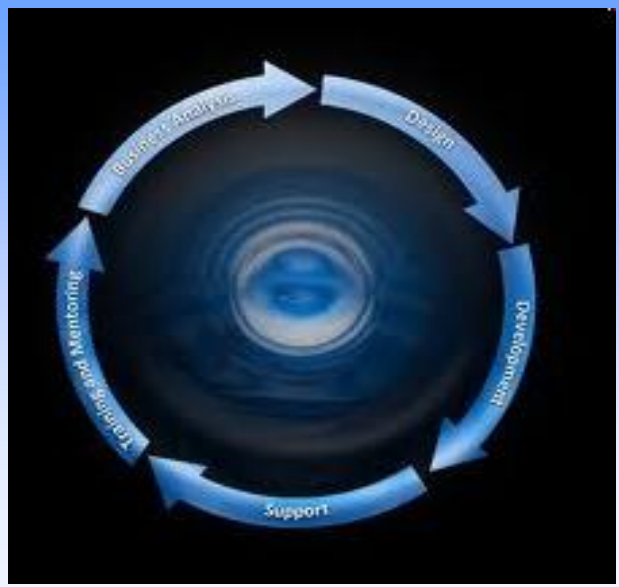
- ***Insert dữ liệu**:* khi người dùng muốn insert một mục dữ liệu, TFE sẽ tính:
  - *Tổng kiểm tra = mức nhạy cảm dữ liệu + dữ liệu*.
  - *Mã hoá* tổng kiểm tra này bằng một khoá bí mật K, tạo ra *tem*, và lưu vào trong CSDL cùng với mục dữ liệu đó (gắn với mục dữ liệu).

Kiến trúc Integrity Lock (…)

- ***Đưa ra dữ liệu:*** Khi đưa ra dữ liệu trả cho người dùng, TFE nhận được dữ liệu từ DBMS không tin cậy, nó sẽ kiểm tra tem gắn với mục dữ liệu xem có chính xác không:
  - Giải mã tem gắn với dữ liệu.
  - So sánh dữ liệu nhận được với dữ liệu sau khi giải mã tem. Nếu không khớp chứng tỏ dữ liệu đã bị sửa đổi.
  - ***Lưu ý:*** nếu dùng hàm băm để tạo tem, thì sau khi DBMS nhận được dữ liệu và tem tương ứng, nó sẽ băm dữ liệu này ra và so sánh với tem nhận được xem có trung nhau không.

Các kiến trúc Woods Hole (…)

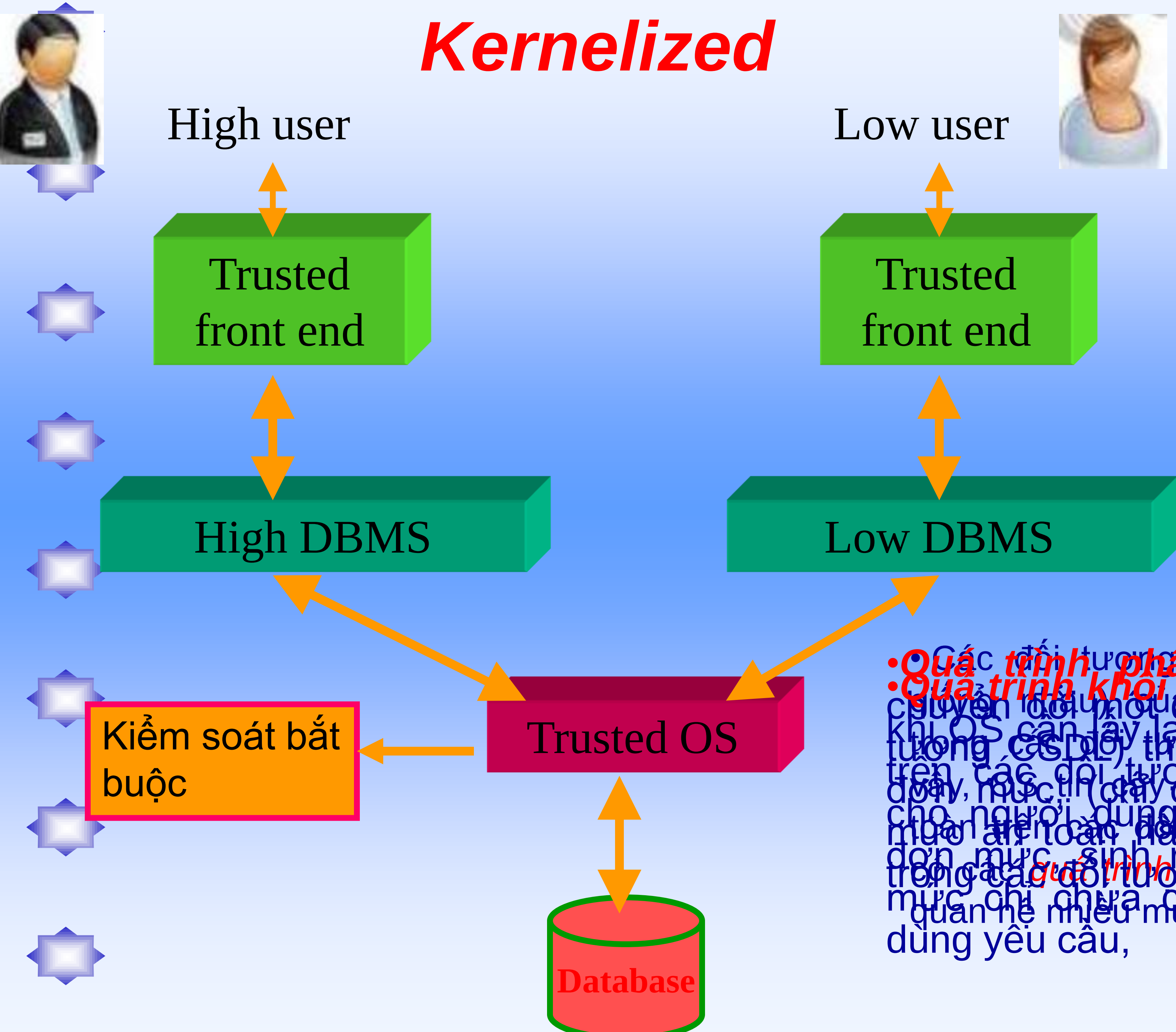
- Kiến trúc Integrity Lock
- **Kiến trúc Kernelized**
- Kiến trúc Replicated (còn được gọi là kiến trúc Distributed)



Kiến trúc Kernelized

- Sử dụng một OS tin cậy, có trách nhiệm đối với các truy nhập vật lý vào dữ liệu (trong CSDL) và có trách nhiệm tuân theo bảo vệ bắt buộc.
- High User (người dùng làm việc ở mức cao) tương tác với một High DBMS, thông qua một TFE, Low User (người dùng làm việc ở mức thấp) tương tác với một Low DBMS cũng thông qua một TFE.
- Sau đó, các yêu cầu của họ được chuyển cho OS, và OS lấy lại dữ liệu hợp lệ từ CSDL.
- Đã có trong mẫu thử Sea View và trong hệ quản trị thương mại Oracle.

Kernelized



- ***Quá trình phân tách:*** thực hiện chuyển đổi một quan hệ đa mức (đối tượng CSDL) thành một số quan hệ đơn mức, (chỉ chứa dữ liệu ở một mức an toàn nào đó), được lưu giữ trong các đối tượng OS.
- ***Quá trình khôi phục:*** được thực hiện khi OS cần lấy lại dữ liệu được lưu giữ trên các đối tượng của nó để trả về cho người dùng. Để từ các quan hệ đơn mức, sinh ra một khung nhìn đa mức chỉ chứa các dữ liệu mà người dùng yêu cầu,
- Các đối tượng (có các nhãn an toàn giống nhau) của CSDL được lưu giữ trong các đối tượng của OS. Vì vậy, OS tiến hành kiểm soát an toàn trên các đối tượng lưu giữ này, cần có các *quá trình phân tách* và *khôi phục* quan hệ nhiều mức.

Các kiến trúc Woods Hole (…)

- Kiến trúc Integrity Lock
- Kiến trúc Kernelized
- **Kiến trúc Replicated (hay kiến trúc Distributed)**

Kiến trúc Replicated (lặp)

- Có trong mẫu thử NRL, nhưng chưa có trong DBMS thương mại nào, vì nó rất đắt!

Replicated

High user

Trusted front end

High DBMS

Database high & low data

- *Dữ liệu mức thấp được lặp trong CSDL mức cao*
- *Người dùng mức thấp chỉ được phép truy nhập vào CSDL mức thấp, không có khả năng sửa đổi dữ liệu mức cao.*
- Người dùng mức cao có thể xem và sửa đổi cả dữ liệu mức thấp và mức cao
- Để tuân theo giải pháp này cần có các thuật toán đồng bộ an toàn để đảm bảo tính tương thích lặp và chi phí lặp cũng rất lớn.

Low user

Trusted front end

Low DBMS

Database low data

Câu hỏi

- ***So sánh 4 kiến trúc DBMS an toàn?***
  - Kiến trúc Trusted Subjects (chủ thể tin cậy)
  - Kiến trúc Integrity Lock
  - Kiến trúc Kernelized
  - Kiến trúc Replicated (còn được gọi là kiến trúc Distributed)

Nội dung

1. So sánh DBMS và OS
2. Các kiến trúc DBMS an toàn
3. *Giới thiệu một vài DBMS*
4. Các vấn đề an toàn chung trong DBMS

Giới thiệu một số hệ quản trị

Sybase Secure Server

- ***Được phân loại vào lớp B1 hoặc B2***
- Phân biệt một ***miền TCB*** với ***miền User*** không tin cậy.
- ***Các đối tượng chính:*** các hàng trong bảng (table rows), là đối tượng nhỏ nhất có thể được gắn một nhãn an toàn.
- ***Các đối tượng phụ*****:** các bảng, CSDL, bao gồm một danh sách cá truy nhập tuỳ ý (ACLs) mà những user hay group hợp pháp được thực hiện.

Sybase Secure Server

- ***Các chủ thể - Subjects*** là các user và group user, sử dụng ngôn ngữ truy vấn T-SQL.
- Các chủ thể có thể được gán các **roles** như: nhân viên an toàn, nhà quản trị CSDL, người sở hữu CSDL, người dùng bình thương.
- Một thủ tục ***đăng nhập-Logon*** được sử dụng để tạo kết nối giữa giao diện người dùng và DBMS.
- Một user ở một mức an toàn, chỉ có thể kết nối tới một mức an toàn không vượt quá mức an toàn của anh ta.

Sybase Secure Server

- ***Hoạt động của User:*** là các yêu cầu bằng các lệnh Transact-SQL (Select, Update, Insert, Delete). Bộ phân tích truy vấn SQL và chương trình dịch sẽ chạy các tiến trình của người dùng không tin cậy. Chúng sẽ truyền các lệnh này thành một tập các lệnh dạng nhị phân (dưới dạng một ***thủ tục-procedure*****).**
- Một thủ tục được thực hiện bởi ***TCB.*** TCB cũng kiểm tra quyền truy nhập của người dùng đó dựa vào mức an toàn của anh ta và dựa vào các quyền DAC mà anh ta có.
- ***Kiểm toán-Auditing*** có thể được cấu hình.

Ingres

- ***Các chủ thể*** là các users và groups.
- Tất cả các user trong một group được cung cấp một tập quyền để thực hiện những ứng dụng cụ thể.
- Khi thực hiện một ứng dụng, một user phải gõ vào ***role*** và ***password*** của role đó.
- ***Các đối tượng-Objects*** là các CSDL, các danh mục liệt kê-catalogues, tables, views, procedures. Ingres sử dụng Grant và Grant Option cho các quyền Select, Insert, Delete, Update và Execute.
- Lệnh ***Auditdb*** để kiểm tra kiểm toán.

Oracle

- Mức phân loại B1 với Unix, A1 với GEMSOS.
- ***Các chủ thể-Subjects*** có thể được tạo, thay đổi và bị xoá.
- Nhà quản trị định nghĩa một ***role***, gán ***các đặc quyền-privileges*** cho role đó và sau đó gán role này cho các chủ thể.
- Gán các role cho các role, tạo ra phân cấp.
- Đặc quyền ***Connect*** để kết nối tới CSDL
- Đặc quyền ***Resource*** để tạo các bảng cơ sở.
- Đặc quyền ***DBA*** cũng để tạo các user.

Oracle

- ***Các đối tượng*** là các CSDL, các bảng, khung nhìn,...Các đối tượng có các nhãn an toàn, định nghĩa ở mức quan hệ.
- ***Các phép toán:*** Select, Insert, Update, Delete, Alter, Index và Reference được thực hiện trên các tables. Đối với các View, chỉ có các phép toán: Select, Insert, Update và Delete. Đặc quyền Execute được thực hiện trên các procedures.
- ***Grant option*** được dùng.
- Lệnh ***Audit*** để kiểm tra các vết kiểm toán.

Nội dung

1. So sánh DBMS và OS
2. Các kiến trúc DBMS an toàn
3. Giới thiệu một vài DBMS
4. *Các vấn đề an toàn chung trong DBMS*

CÁC VẤN ĐỀ AN TOÀN CHUNG TRONG DBMS

- **Xác thực (Authentication):**
  - Là quá trình xác nhận định danh của các cá nhân hay ứng dụng có yêu cầu truy nhập tới một môi trường an toàn.
  - Có ba mức xác thực thường xuyên trong môi trường cơ sở dữ liệu, đó là:

  + Mức hệ điều hành,
  + Mức cơ sở dữ liệu,
  + Hỗ trợ của bên thứ ba.

CÁC VẤN ĐỀ AN TOÀN CHUNG TRONG DBMS

- **Cấp quyền (Authorization):**
  - Là quá trình đảm bảo rằng những cá nhân hoặc các ứng dụng yêu cầu truy nhập vào một môi trường hoặc một đối tượng trong môi trường có sự cho phép hay không.

CÁC VẤN ĐỀ AN TOÀN CHUNG TRONG DBMS

- **Kiểm toán (Auditing):**
  - Là giám sát việc sử dụng tài nguyên hệ thống của người dùng.
- Các cơ chế này bao gồm hai giai đoạn:
  - *Giai đoạn ghi vào nhật ký*: tất cả các câu hỏi truy nhập và câu trả lời liên quan đều được ghi lại (dù được trả lời hay bị từ chối).
  - *Giai đoạn báo cáo:* các báo cáo của giai đoạn trước được kiểm tra, nhằm phát hiện các xâm phạm hoặc tấn công có thể xảy ra.

End of chapter 3