SECURITY IN COMPUTING, FIFTH EDITION

Databases

Objectives for Chapter 7

- Basic database terminology and concepts
- Security requirements for databases
- Implementing access controls in databases
- Protecting sensitive data
- Data mining and big data

Database Terms

- Database administrator
- Database management system (DBMS)
- Record
- Field/element
- Schema
- Subschema
- Attribute
- Relation

Database Terms

- What is a database?
  - A collection of data and a set of rules that organize the data by specifying certain relationships among the data
- Database administrator
  - Person who defines the rules that organize the data and controls who should have access to what parts of the data
- Database management system (DBMS)
  - The system through which users interact with the database

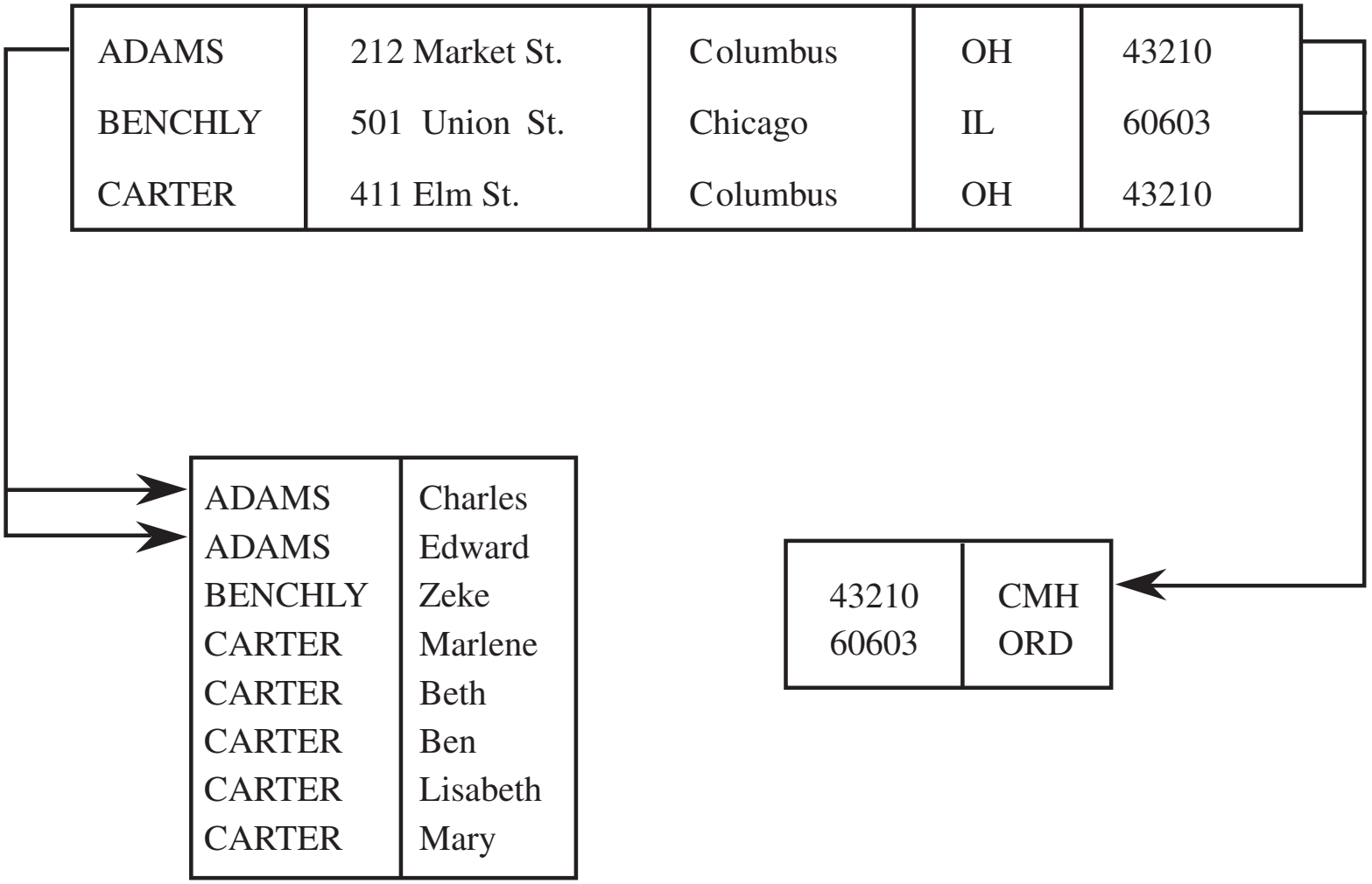
Database Terms

- Record
  - One related group of data
- Field/element
  - Elementary data items that make up a record (e.g., name, address, city)
- Schema
  - Logical structure of a database
- Subschema
  - The portion of a database a given user has access to

Database Terms

- Attribute
  - A column in a database

Database Example



Schema Example

Name	First	Address	City	State	Zip	Airport
ADAMS	Charles	212 Market St.	Columbus	OH	43210	CMH
ADAMS	Edward	212 Market St.	Columbus	OH	43210	CMH
BENCHLY	Zeke	501 Union St.	Chicago	IL	60603	ORD
CARTER	Marlene	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Beth	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Ben	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Lisabeth	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Mary	411 Elm St.	Columbus	OH	43210	CMH

Queries

- A command that tells the database to retrieve, modify, add, or delete a field
  - or record
- The most common database query language is SQL
- The result of executing a query is a subschema

Example SQL Query

- `SELECT ZIP='43210'`

Name	First	Address	City	State	Zip	Airport
ADAMS	Charles	212 Market St.	Columbus	OH	43210	CMH
ADAMS	Edward	212 Market St.	Columbus	OH	43210	CMH
CARTER	Marlene	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Beth	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Ben	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Lisabeth	411 Elm St.	Columbus	OH	43210	CMH
CARTER	Mary	411 Elm St.	Columbus	OH	43210	CMH

Advantages of Databases

- A database is a single collection of data, stored and maintained at one central location
- People and application have access to it as needed
- Implementation may involve some other physical storage
  - At a local or remote location
- The users are unaware of the physical arrangements
  - Only see a unified logical arrangement

Advantages of Databases

- Shared access
  - many users can use one common, centralized set of data
- Controlled access
  - Only authorized users are allowed to view or to modify data values
- Minimal redundancy
  - Individual users do not have to collect and maintain their own sets of data

Advantages of Databases

- Data consistency
  - A change to a data value affects all users of the data value
- Data integrity
  - Data values are protected against accidental or malicious undesirable changes

Database Security

- Why is security required?
- Databases used by many entities, such as:
  - banks, large retailers, and law enforcement
- Therefore, data needs to be protected
  - It's confidentiality, integrity and availability to its users

Database Security Requirements

- Physical integrity
- Logical integrity
- Element integrity
- Auditability
- Access control
- User authentication
- Availability

Reliability and Integrity

- Reliability: in the context of databases, reliability is the ability to run for long periods without failing
- Database integrity: concern that the database as a whole is protected against damage
- Element integrity: concern that the value of a specific data element is written or changed only by authorized users
- Element accuracy: concern that only correct values are written into the elements of a database

Database Update

- Concern: What if the database system fails in the middle of an update?
  - Leaving the database in a partially updated and inconsistent state
- Solution: two-phase update

Two-Phase Update

- Phase 1: Intent
  - DBMS does everything it can, other than making changes to the database, to prepare for the update
    - Collects records, opens files, locks out users, makes calculations
  - DBMS commits by writing a commit flag to the database
- Phase 2: Write
  - DBMS completes all write operations
  - DBMS removes the commit flag
- If the DBMS fails during either phase 1 or phase 2, it can be restarted and repeat that phase without causing harm

Other Database Security Concerns

- Error detection and correction codes to protect data integrity
- For recovery purposes, a database can maintain a change log, allowing it to repeat changes as necessary when recovering from failure
- Databases use locks and atomic operations to maintain consistency
  - Writes are treated as atomic operations
  - Records are locked during write so they cannot be read in a partially updated state

Sensitive Data

- Inherently sensitive
  - Passwords, locations of weapons
- From a sensitive source
  - Confidential informant
- Declared sensitive
  - Classified document, name of an anonymous donor
- Part of a sensitive attribute or record
  - Salary attribute in an employment database

Sensitive Data (cont.)

- Sensitive in relation to previously disclosed information
  - An encrypted file combined with the password to open it

Preventing Disclosure

- Keeping records from being dumped out of the database is not sufficient to prevent disclosure.
- There are many ways to deduce the content of a database listed on this slide
  - all of them must be considered when protecting sensitive database information.
- To apply the appropriate protection mechanisms, it is important to understand:
  - the range of possible contents of each attribute
  - the data available to potential attackers

Types of Disclosures

- Exact data
- Bounds
- Negative result
- Existence
- Probable value
- Direct inference
- Inference by arithmetic
- Aggregation

Types of Disclosures (cont.)

- Hidden data attributes
  - File tags
  - Geotags

Inference

- **Inference** is a way to infer or derive sensitive data from non-sensitive data.
- The inference problem is a subtle vulnerability in database security
- Example: a database may only return the number of entities who have a certain condition
  - If query is focused enough, return of zero would confirm a certain patient does not have this condition

Inference

- **Inference by Arithmetic**: Using statistics to determine sensitive information about users
  - Example: A database may only return the mean of the sensitive values
    - And not the actual values
  - An attacker may create a query which returns one or very few results
    - ‘select female with illness A who lives on Mamaroneck road and is 56 years old’
    - In this case, the attacker may learn the exact information

Aggregation

- Building sensitive results from less sensitive inputs
- Aggregation allows searches in parallel through databases
  - And combining the results

Aggregation

- Example: looking for crime suspects:
  - Find out who ad a motive for committing the crime,
  - when the crime was committed
  - who had alibis covering that time
  - who had the skills
- Using database queries to parallelize this scenario:
  - Create a list of possible suspects,
  - a list with possible motive, and
  - a list of capable persons
- The intersection of these lists is a single person
  - the police have their prime suspect

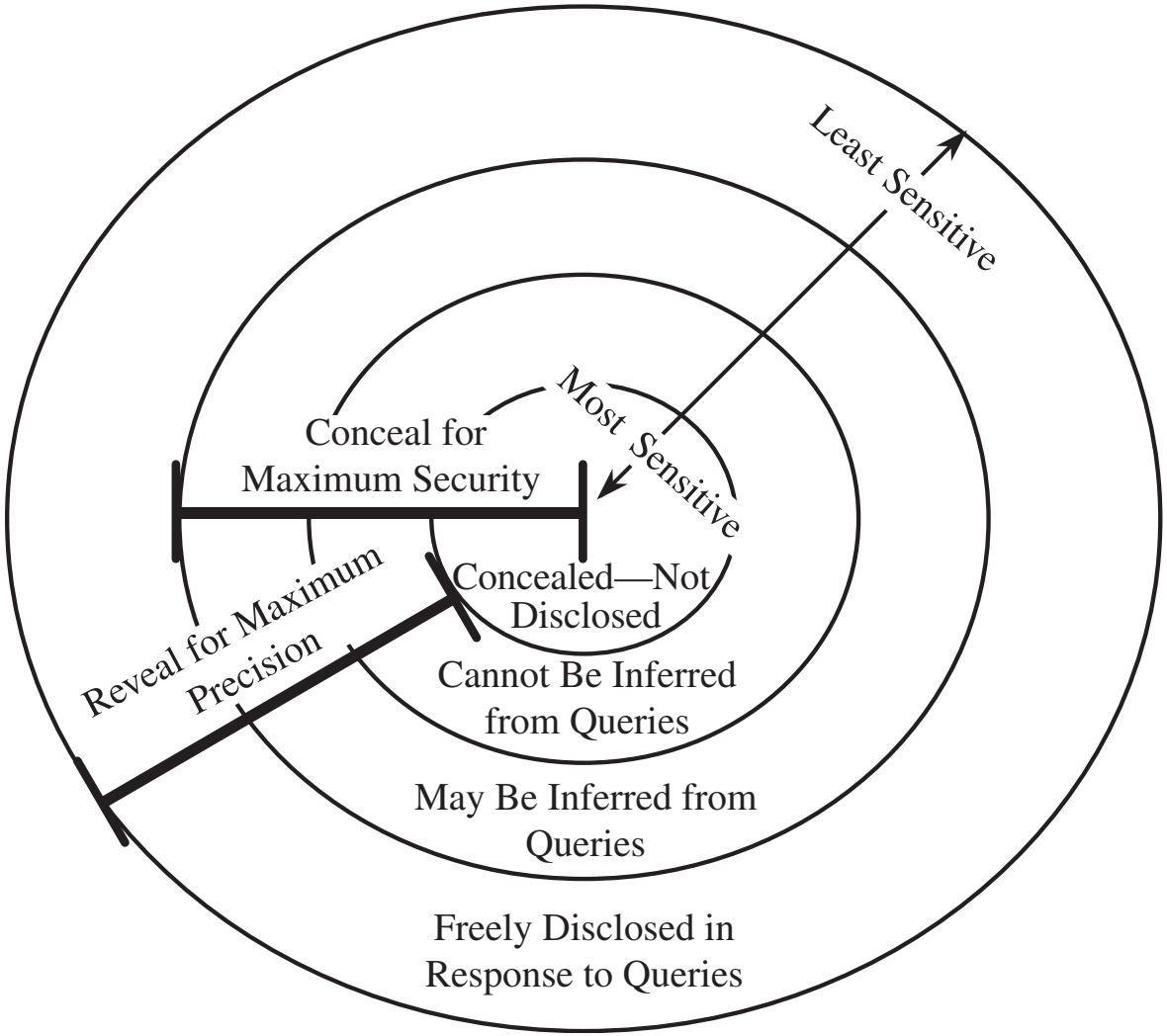
Preventing Disclosure

- Suppress obviously sensitive information
- Keep track of what each user knows based on past queries
- Disguise the data

Security vs. Precision

- Precision: precise, complete, and consistent responses to queries
- Precision against sensitive information make it more likely that sensitive info will be disclosed

Security vs. Precision



Suppression Techniques

- Limited response suppression
  - Eliminates certain low-frequency elements from being displayed
- Combined results
  - Ranges, rounding, sums, averages
- Random sample
- Blocking small sample sizes
- Random data perturbation
  - Randomly add or subtract a small error value to/from actual values

Suppression Techniques (cont.)

- Swapping
  - Randomly swapping values for individual records while keeping statistical results the same

Data Suppression

- Less complex data makes for simpler inference and therefore is more likely to require suppression.
- The disclosure prevention must be balanced against the database requirements
  - as the loss of precision and completeness may make the database unusable

Data Mining

- Data mining uses statistics, machine learning, mathematical models, pattern recognition, and other techniques
  - to discover patterns and relations on large datasets
- The size and value of the datasets present an important security and privacy challenge,
  - as the consequences of disclosure are naturally high

Data Mining Challenges

- Correcting mistakes in data
  - What happens when data is moved to more databases before the original database can be corrected?
    - Need for correction may not be disclosed
    - Open challenge
- Preserving privacy
- Granular access control
  - Access control is often performed in a coarse way

Data Mining Challenges

- Secure data storage
  - Data may be collected globally and through cloud providers
    - Where security details are largely unknown to users
- Transaction logs
- Real-time security monitoring

Data Mining Challenges

- Many challenges remain open, partially solved or solved in certain data mining packages.
- As data mining platforms evolve, these features will mature

Big Data

- Analysis of massive amounts of data
- often collected from different sources
  - Distributed databases:
    - Oracle, MySql, etc.
  - Big data databases:
    - Hadoop, MongoDB, etc.
  - Data warehouses:
    - Netezza, Apache live, etc.
  - Fileshare:
    - Dropbox.com, box.com, Google drive, etc.

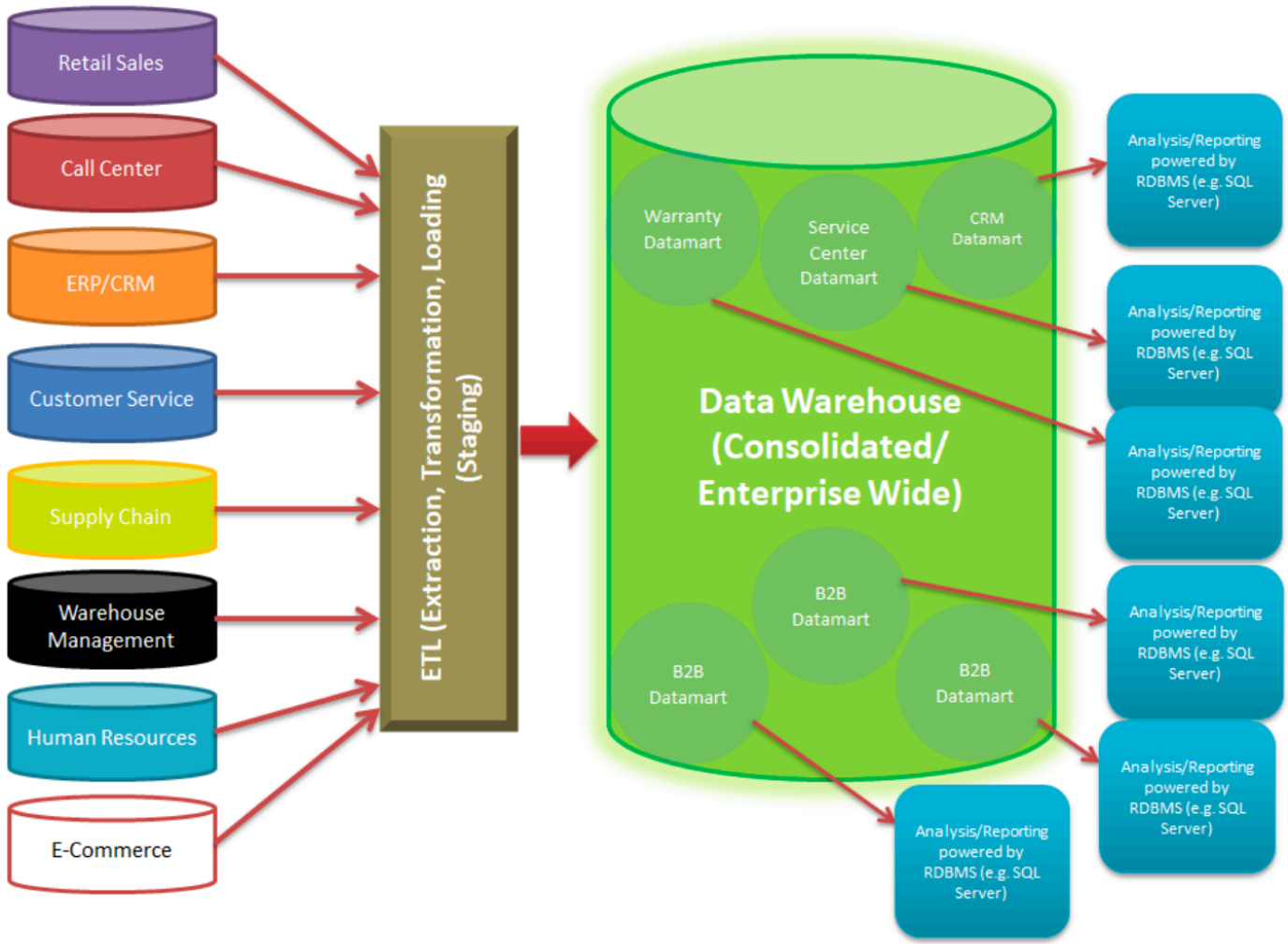
Data Warehouses

- First tier of Big Data repositories
- Fed information (via APIs or raw files) from different sources/business units (sales, customer service, etc)
- Information is cleansed and processed in a staging area before being fed into the Data Warehouse itself
- From this point on the data can be filtered and made available as different subsets (Datamarts)
  - usually pertaining to different lines of business.

Data Warehouses

- The data within each Datamart is housed within relational databases
  - A collection of tables containing related data that can be linked via unique keys
  - Records from each table can have one to one or one to many relationships with records in other tables
  - Relational Databases typically use SQL to extract and manipulate the data
    - have business rules and quality assurance checks in place to maintain the integrity of the data
    - There are different “engines” that power relational databases
      - Oracle, SQL Server and MySQL, etc.

Basic Data Warehouse Architecture



https://hickamsdictum.com/big-data-activation-data-warehouses-and-data-lakes-e0df2ae78c62

Data Lakes

- Similar to data Warehouses
- Main difference: data Lakes ingest and process data from both internal and external sources
  - Data ingested via APIs
    - In any format - structured and unstructured.
  - Data stored in its natural/raw format

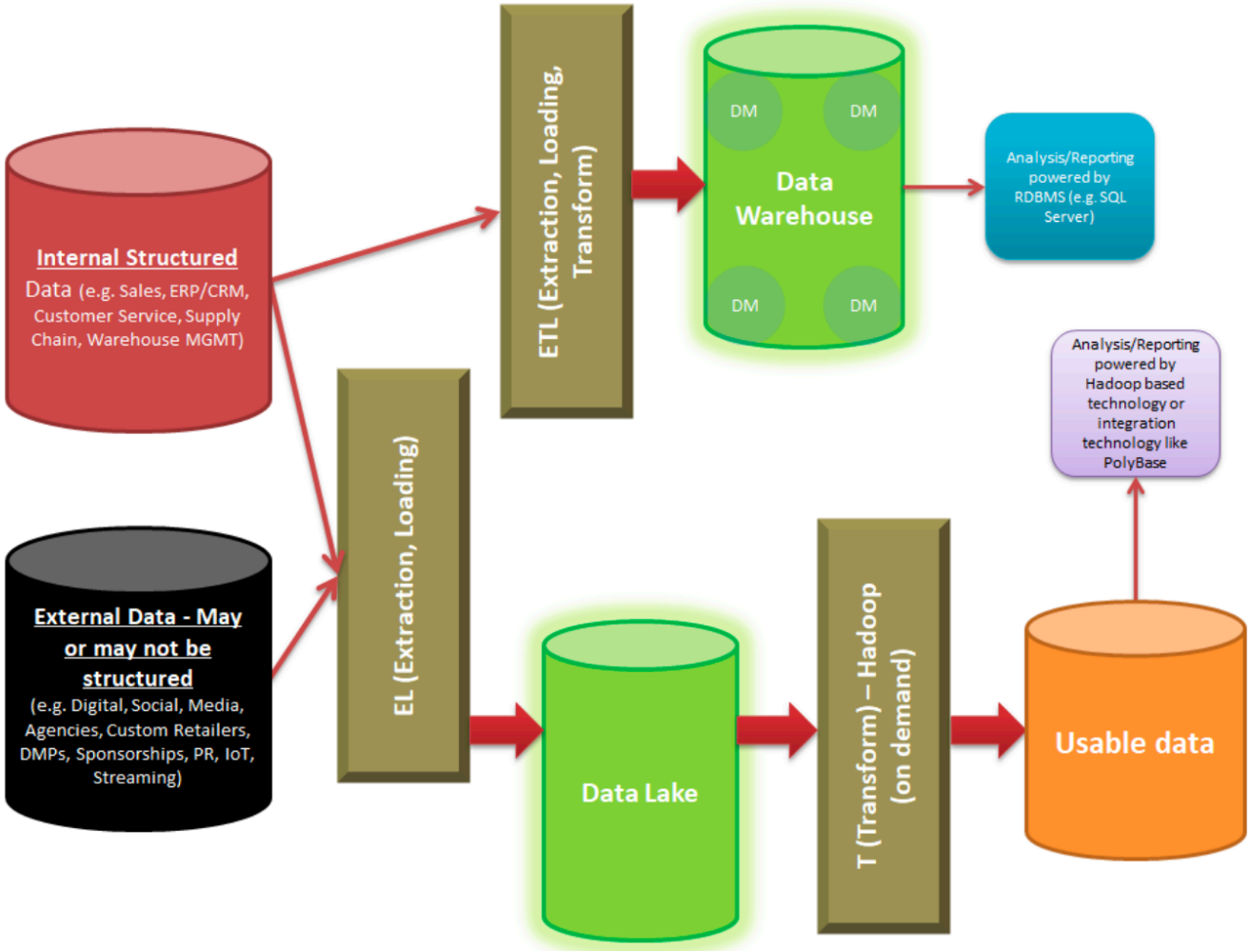
Data Lakes

- Example: allow a large corporation to have one central storage and processing location
  - internal items like sales and service data
  - external data like cross channel digital marketing performance metrics from external agencies, digital audience profiles (from DMPs), etc.

Data Lakes

- Traditionally, Data Lakes were (and still are) powered via Hadoop frameworks
- Data Lakes typically ingest data via an ELT process:
  - Data is extracted from each source and loaded into the Hadoop framework
  - Data is then processed directly from the Datalake and not in a staging area
    - in addition to their ability to handle Big Data, Hadoop clusters can also process data (Transform) faster than most ETL tools.
      - Therefore, the processing can happen within the Data Lake itself as needed

Data Lakes


Internal Structured Data (e.g. Sales, ERP/CRM, Customer Service, Supply Chain, Warehouse MGMT)
External Data - May or may not be structured (e.g. Digital, Social, Media, Agencies, Custom Retailers, DMPs, Sponsorships, PR, IoT, Streaming)
ETL (Extraction, Loading, Transform)
EL (Extraction, Loading)
DM
DM
Data Warehouse
DM
DM
Data Lake
T (Transform) – Hadoop (on demand)
Usable data
Analysis/Reporting powered by RDBMS (e.g. SQL Server)
Analysis/Reporting powered by Hadoop based technology or integration technology like PolyBase

APACHE HADOOP APPLICATION FRAMEWORK

Conventional Architecture

- Conventional architecture scales by adding more, bigger, or faster components
- To expand such a system, storage can be increased by addition of more disks
- However, there is an implicit limit to how far storage can grow
  - without suffering serious performance delays: At some point the biggest device on the market fills up and a
- Different architecture is needed.

Conventional Architecture

- Different architecture is needed
- A new processor can give greater speed, however, again:
  - Existing technology has its limits
  - Higher performance tends to be disproportionately more expensive.
  - One processor and one storage array become potential points of catastrophic failure
    - Big data requires an architecture that can readily scale to

Conventional Architecture

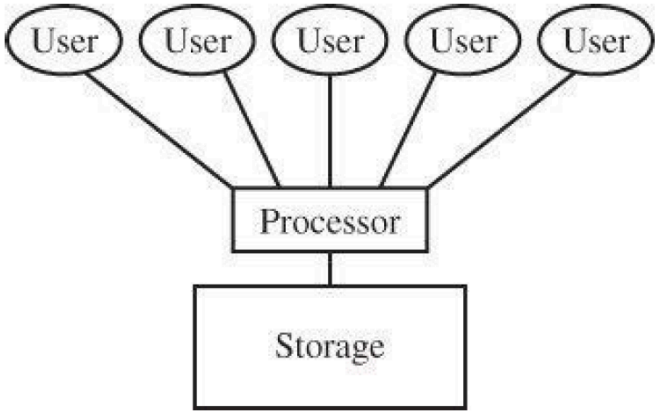
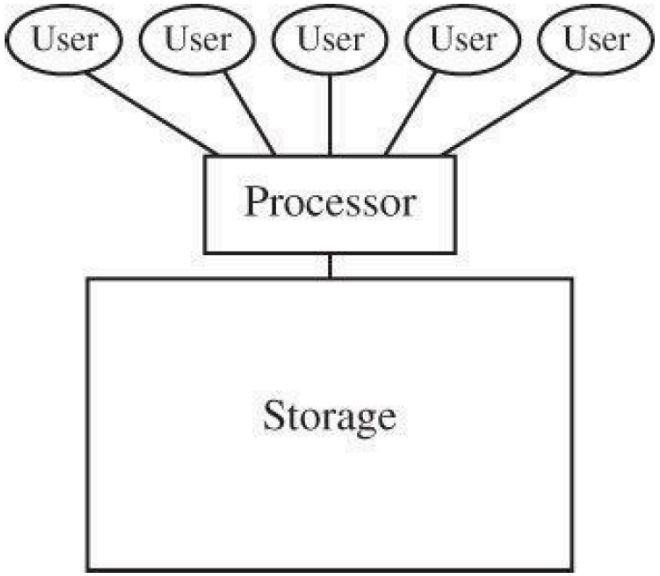


FIGURE 7-5 Conventional Computing Architecture



HADOOP Application Framework

- Hadoop technology was originally used to manage the web
  - making it ideal for data lakes as it can handle different file formats.

HADOOP Application Framework

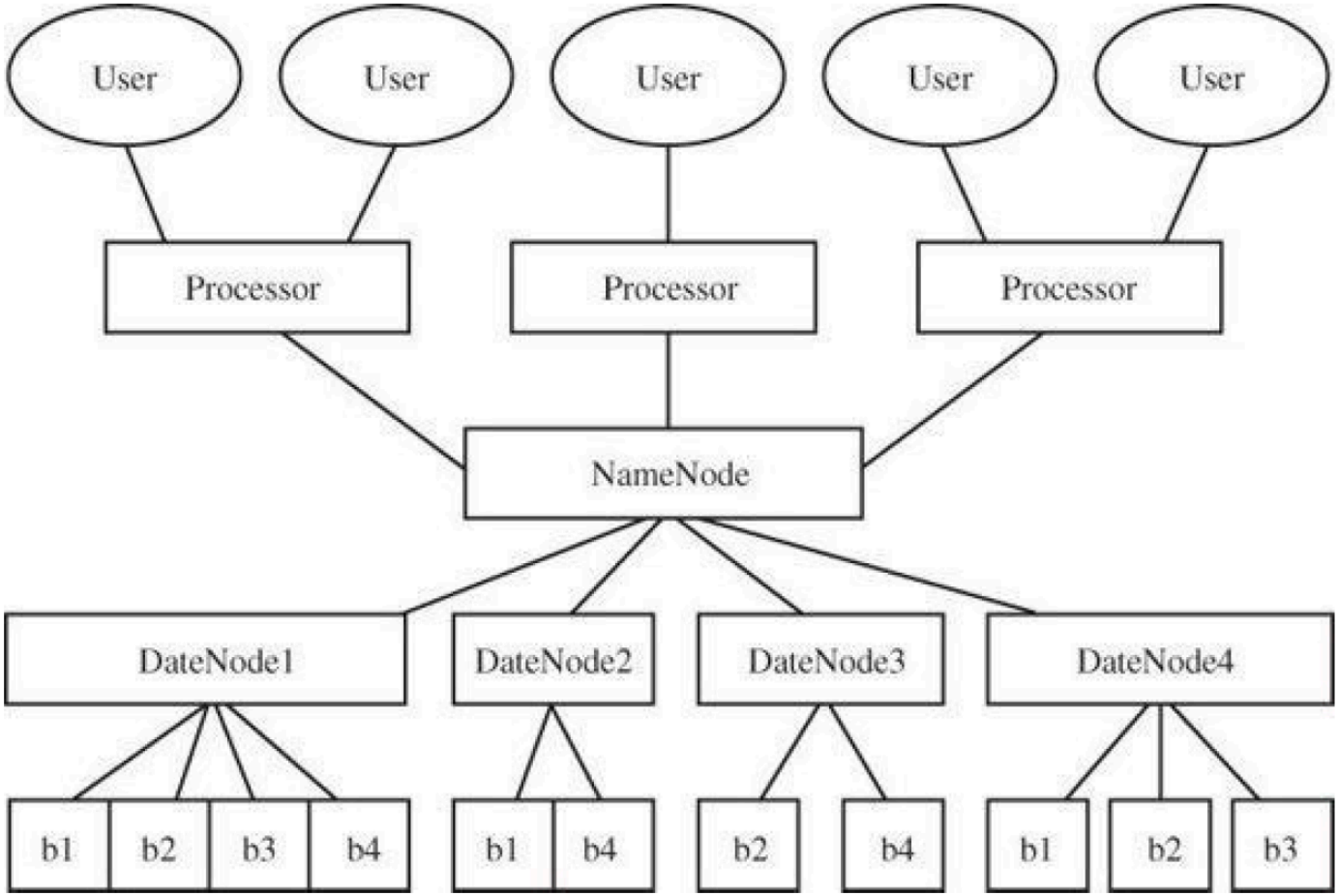


FIGURE 7-7 Hadoop Architecture

HADOOP Application Framework

- The Apache Hadoop framework is a software environment for running big data projects.
  - http://hadoop.apache.org
- Users use Hadoop clusters to manage the data they collect
  - E.g. Yahoo!, LinkedIn, and Twitter

HADOOP Application Framework

- Hadoop is one of the most common big data frameworks
- Largest or fastest processing will always have a performance limit depending on the technology
- Instead, Hadoop uses an unbounded array of smaller components ganged into a network.

HADOOP Application Framework

- Hadoop supports:
  - Distributed data storage and processing
  - Multiple computing platforms of different types, redundancy
  - Concurrent access.

HADOOP Application Framework

- Originally built for a project involving web crawlers
  - autonomous agents that traverse the Internet and build indices for web search engines
    - Number of web pages and descriptors of content on those pages is huge.
- Data on web content tend to have few interconnections and a simple structure
  - In contrast to a highly structured relational database,
  - Important to providing providing some result quickly
    - More than providing the most comprehensive answer slowly

HADOOP Application Framework

- The Hadoop model was developed for an environment of open data shared by all
- It has no mechanism for security primitives;
  - No access control, correctness checking, privacy, user identification or authentication, logging of actions, or limited privileges
    - all primitives you might expect in any secure environment.

HADOOP Application Framework

- The early security model was total separation:
  - Big data was processed in a separated, trusted environment
    - by only trusted users on dedicated machines.
  - Similar to the earliest mainframe computing installations, original intentions for the Unix
    - A closed environment where all users knew and trusted all others
      - there was no need to exclude some users from some data
    - Over time, designers implemented security for both mainframe computers and Unix
      - although adding on security was challenging.

HADOOP Application Framework

- Hadoop secure mode is described on the Hadoop website
  - https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SecureMode.html
    - and in a white paper [O'Malley 2-09] from the Yahoo!
  - These proposals for security functions suggested security features to be included in the Hadoop framework

HADOOP Application Framework

- The white paper identified two security holes to address:
  - lack of user authentication (and identification)
  - lack of access control to data blocks.

HADOOP Application Framework

- Yahoo team proposed a secure mode that involves the following extensions to Hadoop:
  - Authentication for end-user web devices
  - mutual authentication with Kerberos
    - (user–process–Hadoop service)
  - Access control to files in the Hadoop file system
  - Delegation tokens for continuous authentication between internal clients and services

HADOOP Application Framework

- Yahoo team proposed a secure mode that involves the following extensions to Hadoop (cont.):
  - Job tokens for distributing access authorization to multiple distributed platforms
    - collectively implement a data search across disparate stores
  - SSL encryption for network traffic

HADOOP Application Framework

- Security extensions implemented
  - Enhancements added after the fact
  - Security functions not allowed to reduce performance significantly by more than 3%
- Some vulnerabilities still exist

Summary

- Database security requirements include:
  - Physical integrity
  - Logical integrity
  - Element integrity
  - Auditability
  - Access control
  - User authentication
  - Availability

Summary

- There are many subtle ways for sensitive data to be inadvertently disclosed
- There is no single answer for prevention
- Data mining and big data have numerous open security and privacy challenges

- Questions?