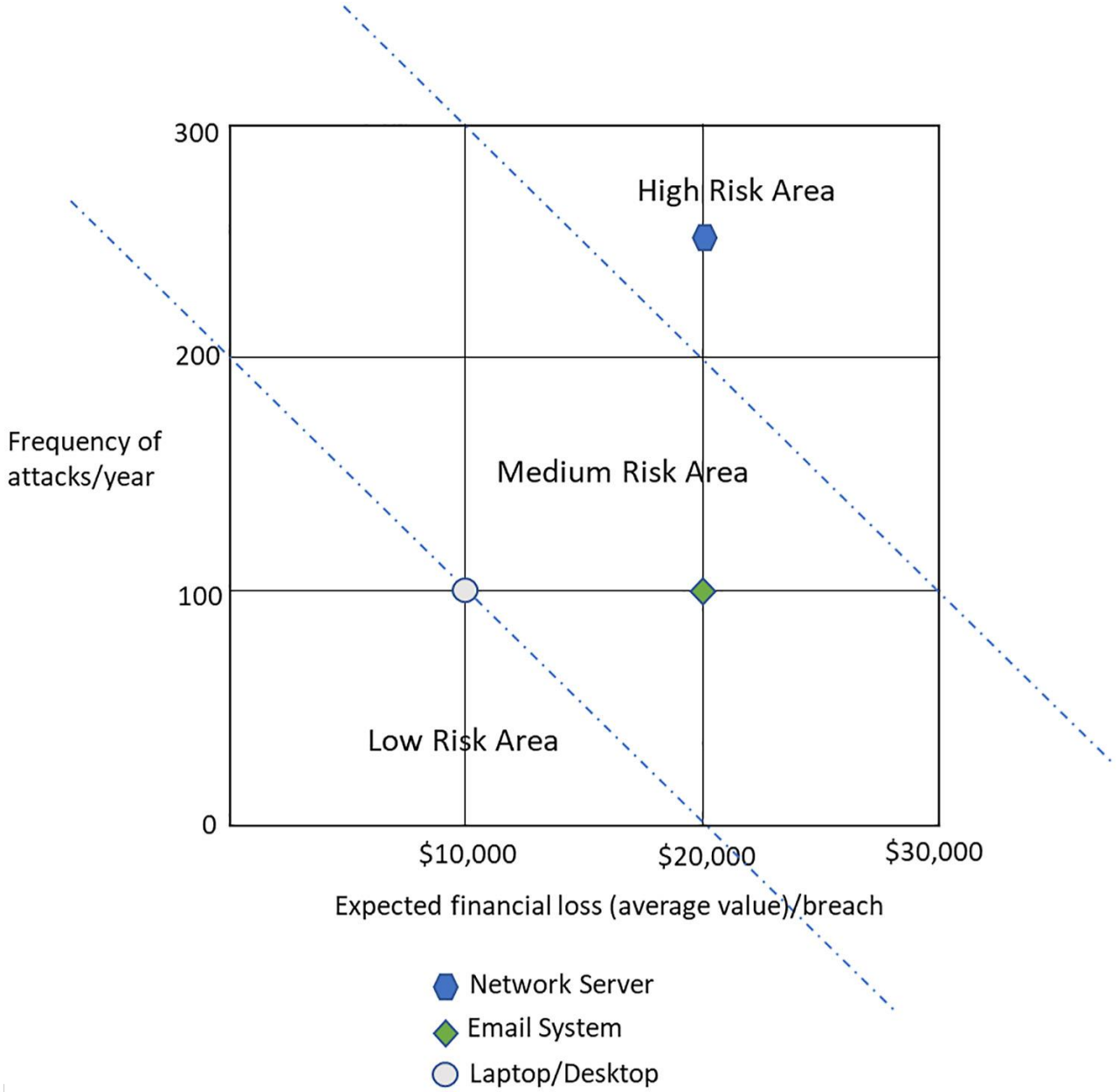
Lecture 8: Database Security

Dr Anesu Nyabadza

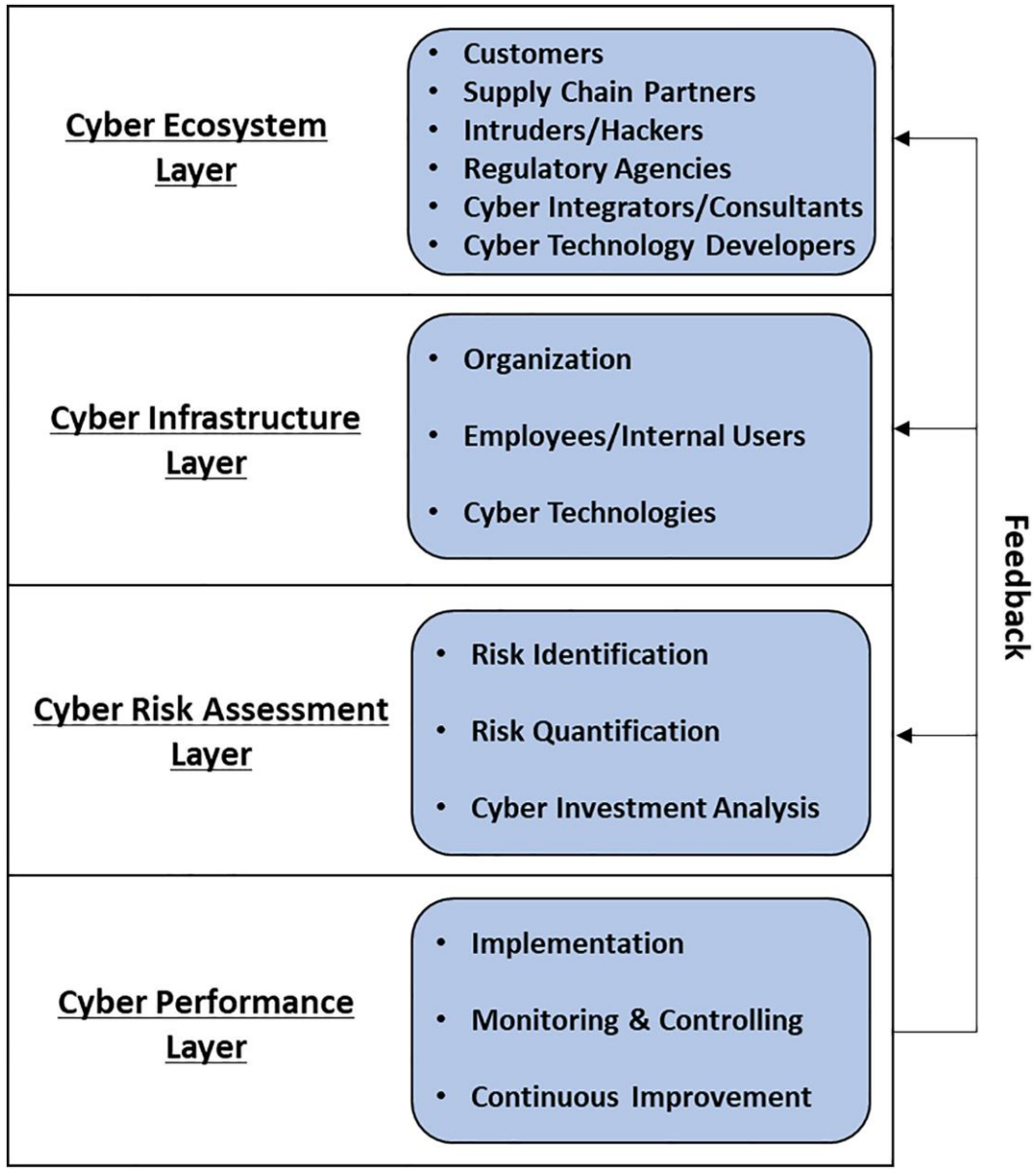

DATABASE
SECURITY

•Average number of security breaches increased by 11%, from 130 in 2017 to 145 in 2018 per organization (Accenture, 2019).
•Annual cybersecurity spending worldwide grew by 64%, reaching $124 billion in 2020, with a projected $133.8 billion in 2022 (Statista, 2020; Business Wire, 2019).
•Top security and risk trends for 2020 include creating risk appetite statements, implementing Security Operations Centers (SOCs), and prioritizing cloud security competency (Gartner, 2020).

Risk identification



Risk identification- The cybersecurity layers



1. Cyberecosystem layer

•The **cyberecosystem layer** is the top tier of the cyber risk management framework, encompassing largely independent or interdependent stakeholders with potentially conflicting interests and goals.

•Understanding the interactions of specific stakeholders, such as **supply chain partners, customers, intruders/hackers**, regulatory agencies, technology developers, and integrators/consultants, with IT assets like applications, networks, and data is vital for developing defense strategies against cyberattacks.

•Organizations must continuously monitor and assess the cyberecosystem, communicating detected changes to other layers of the cyber risk management framework.

•The cyberecosystem facilitates **collaborative and competitive engagements** with stakeholders to support cybersecurity activities within an organization.

•Adversaries, including **intruders and hackers**, are part of the cyberecosystem, necessitating the identification and analysis of their methods, motivations, and techniques to breach IT systems and commit cyberattacks.

•Regular training and education of staff on the ecosystem is vital at preventing these attacks

2. Cyberinfrastructure layer

- **The cyberinfrastructure layer**, positioned in the middle of the cyber risk management framework, actively safeguards the existing IT assets and services of an organization, involving three key elements: **organizations, employees/internal users, and cybertechnologies**.

- The **organization element** at the core defines **roles, responsibilities, policies**, and processes for cybersecurity management, influencing the **strategies for cyberdefense and mitigation**. Positive attitudes toward cybersecurity policies correlate with more secure behaviors, emphasizing the significance of organizational commitment (Choong & Theofanos, 2015).

- The **employees/internal users element**, focusing on **awareness, motivation, and behavior regarding cybersecurity risk**, interacts with the cyberecosystem and can be a potential security risk. Raising cybersecurity awareness and **implementing training are essential** to address the **people-centric aspect of cybersecurity** and integrate best practices into daily tasks.

- Cybertechnologies play a critical role in protecting **IT assets and services**, covering applications, networks, and databases. Continuous assessment of cyberthreats and the strategic commissioning and decommissioning of cybertechnologies are necessary for successful cybersecurity management.

- In deploying cybertechnologies, organizations must **analyze how these technologies are used**, identify threats to vulnerabilities in applications and networks, and consider data security. Understanding data generation, usage, and potential targets of cyberattacks is crucial for adopting specific cybertechnologies. Machine-learning technologies are gaining attention for their effectiveness in certain scenarios (Lezzi et al., 2019).

3. Cyber risk assessment layer

The cyber risk assessment layer includes three steps
1. **Identify** potential cybersecurity threats, vulnerabilities, and attacks.
2. **Risk quantification prioritizes** cyberattack types based on their severity and frequency.
3. Conducting **cyberinvestment analysis** to assess cost-benefit and make informed judgments regarding cyberinfrastructure investments.

•**Understanding intruders' and hackers' preferred approaches** is essential for identifying cyber risks. Establishing and updating taxonomies for assets, vulnerabilities, and threats over time aids in risk identification. Risk identification involves recognizing external and internal actors, such as hackers and internal users, and their potential impact on specific assets.

•Increasing adoption of risk quantification in various industry sectors for efficient resource allocation and improved overall security. Risk quantification involves **measuring attack frequencies, calculating consequences' magnitude, and prioritizing attacks using a risk matrix**. **Probability distributions model cyberattacks**, and trend data from monitoring systems adjust cyber risk action plans over time.

•Preventing financial losses due to cyberattacks requires investment in security measures and data protection systems. Cyberinvestment cost analysis considers the **tradeoff between cyberattackers and cyberdefenders**, **aiming to minimize total costs**.

4. Cyber risk assessment layer

Implementation

- Involves the **development, testing, and deployment of cybersecurity systems**, including new policy development, training, and user acceptance studies.
- The implementation integrates new cyberinfrastructure with existing organizational elements and evaluates commercially available cybertechnologies and vendors using established selection criteria.
- Emphasizes the **ease, usability, and usefulness of monitoring** and control systems during implementation.

Monitoring and Control

- Encompasses **real-time monitoring of external cyberattacks**, **abnormal user activities**, and **illegal data and application access**, focusing on prevention, detection, and recovery simultaneously.
- Logs types and sources of cyberattacks, their frequencies, and magnitude, providing essential data for future cyberinvestment cost analysis.
- Prioritizes timely responses to cyberthreats during the risk monitoring and control stage to minimize potential damages.

Continuous Improvement

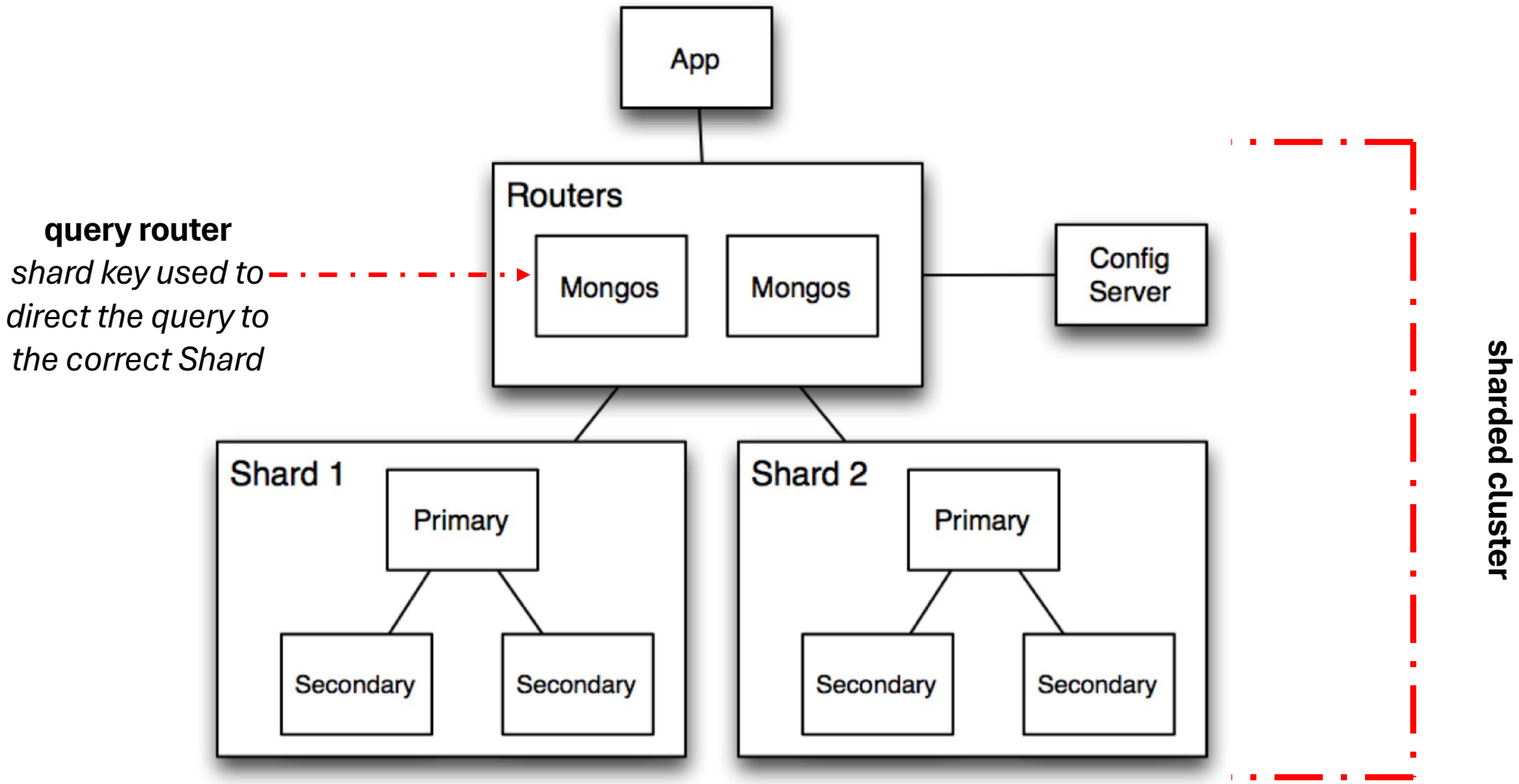
- Utilizes data collected over time to identify trends in cyberattacks and long-term performance, establishing measurable goals and generating periodic performance reports.
- Benchmarking against industry best practices and competitors aids in setting performance goals for various security dimensions.
- Enables organizations to adapt cyberinvestment and strategies based on evolving patterns of cyberthreats and financial losses, with the timing of periodic evaluations tailored to the organization's complexity and IT systems.

Risk identification

With the increased cybersecurity risks posed by cybercriminals and adversaries, it became imperative for organizations to increase their awareness of the change in the cybersecurity landscape and respond to the change quickly.

- 1. Understand our external environment through the **cyberecosystem layer**
- 2. Evaluate the organization, employees/internal users, and existing cybertechnologies through the **cyberinfrastructure layer**
- 3. Assess cyber risks through the cyber **risk assessment layer,** and
- 4. Conduct cybersecurity activities at the **cyberperformance layer**.

Advantages of MongoDB over Relational Databases- Structure



Advantages of MongoDB over Relational Databases- Scalability

1. Scalability

Sharding is a database architecture strategy designed to address the challenges of handling large and growing datasets by **horizontally** partitioning data across **multiple servers** or clusters. In MongoDB, sharding involves distributing data across different **shards, where each shard is essentially an independent database**. Sharding is a key feature that allows MongoDB to scale horizontally, ensuring efficient data distribution, improved performance, and increased system capacity.

Each **shard** can operate independently, enabling the system to handle a larger volume of data and increased read and write operations.

Key Aspects of Sharding

a) **Shard Key**

The **shard key** is a crucial component of sharding in MongoDB. It is the field or combination of fields chosen to determine how data is distributed across shards. Selecting an appropriate shard key is essential for achieving balanced data distribution and optimizing query performance.

b) **A shard** is a subset of a MongoDB database that contains **a portion of the overall data**. Each shard operates **independently**, with its own set of resources, **including storage and processing capacity**. Shards collectively form a sharded cluster.

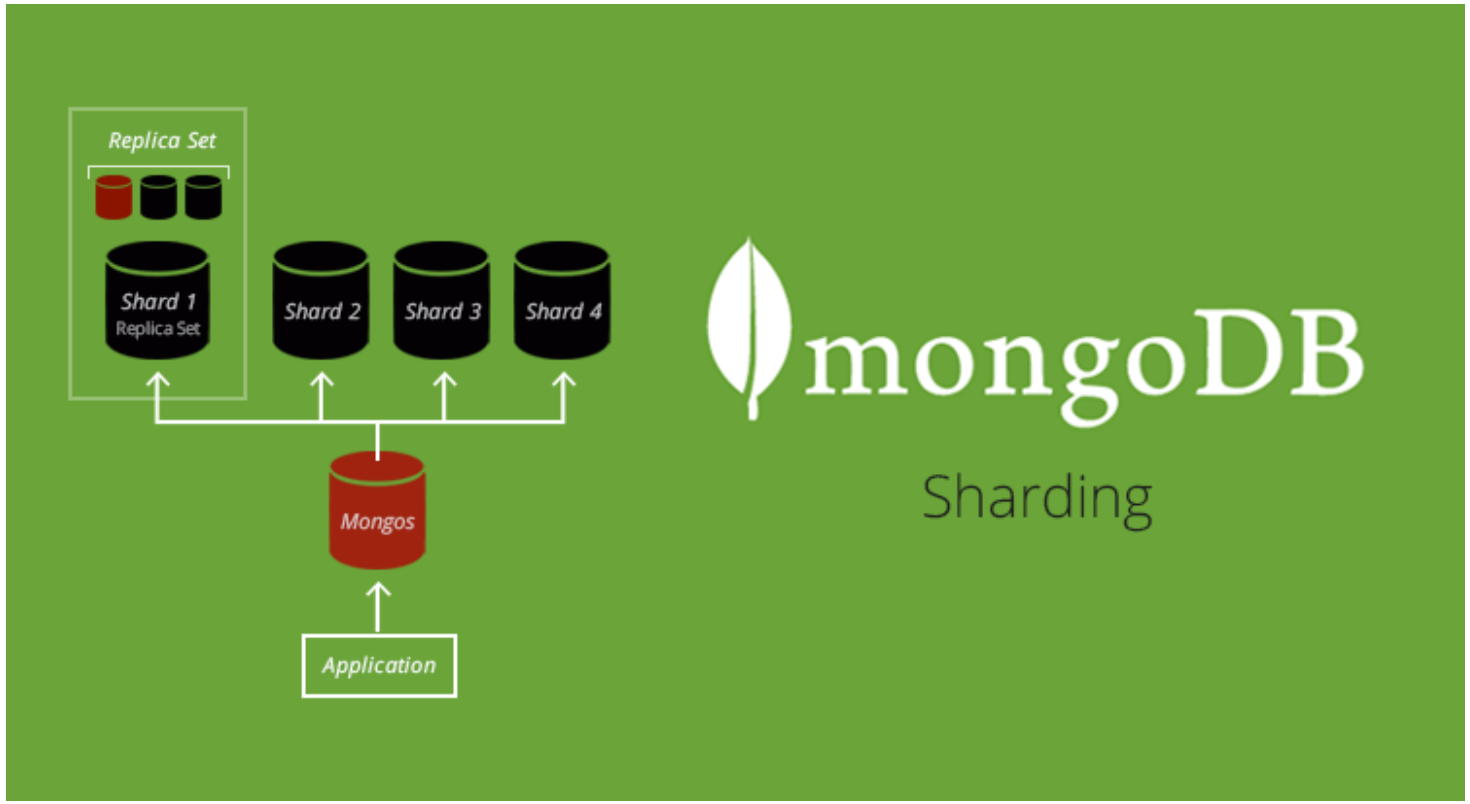
c) **sharded cluster** is the entire MongoDB deployment that incorporates **multiple shards**. It consists of three main components: the i) **query router** (mongos), ii) **config servers, and iii) the shards**. The query router directs client requests to the appropriate shard based on the shard key, and config servers store metadata about the sharded data.

d) Balancer

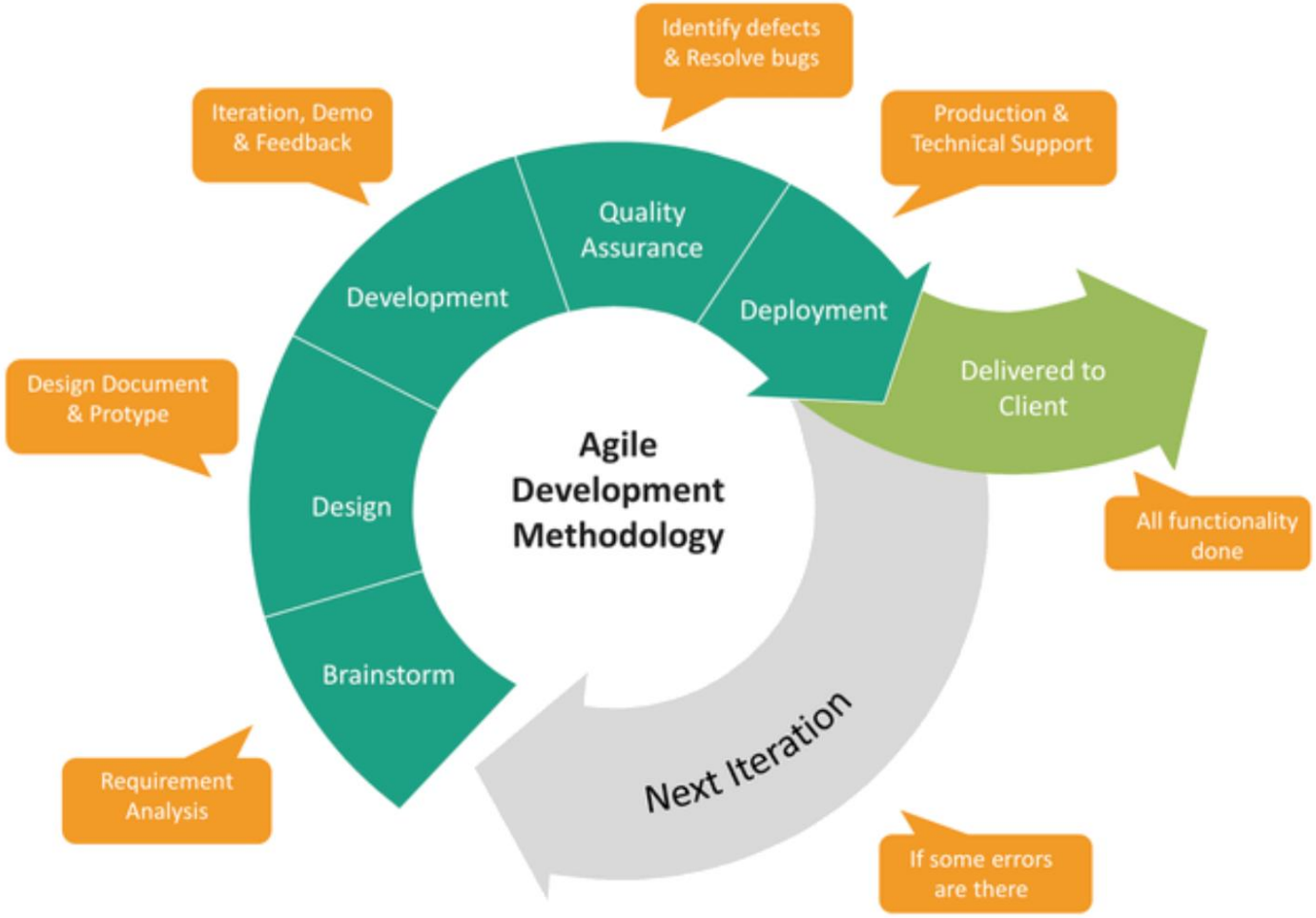
MongoDB includes an **automatic balancer** that **monitors the distribution of data across shards** and **moves chunks (ranges of data) between shards as needed** to maintain a balanced workload. The balancer helps **prevent hotspots** and ensures efficient utilization of resources across the sharded cluster.

4. High Availability

Sharding **enhances fault tolerance and availability**. If one shard goes down, the **remaining shards can continue to operate**, minimizing the impact on the overall system. **Replication** within each shard further ensures data redundancy and fault tolerance.



Advantages of MongoDB over Relational Databases- Agile Development in requirements analysis



Database Security

As soon as a database system is put into operation, it is exposed to risks.
Data can be **lost, unintentionally or maliciously changed**, or **disclosed**. In order to ensure the long-term functioning of a database system, the general objectives of information **securit**y as classically defined are

- **Confidentiality:** Protecting privacy from unauthorized access to information
- **Integrity:** Ensuring the correctness and accuracy of information
- **Availability:** Maintaining the functional state of the information system

Frontline security
Database security is based on the **basic security of the data center**, **the computers, and the network** in which the database server is operated. Computers must use the **latest version of all software components** to close security gaps. The network must be protected with **firewall rules and geo-IP filters**. And the data center must **physically protect hardware** from access.

Database Security

Confidentiality	Integrity	Availabilty
▶ Authentication ▶ Password policies ▶ Read access authorization ▶ Protection against code injection ▶ Encryption of the database ▶ Encryption of communication (SSL) ▶ Certification of the database server	All measures for confidentiality; furthermore: ▶ Authorization with restricted write rights ▶ Integrity conditions ▶ Transaction management ▶ Auditing of all database activities	All measures for integrity; moreover: ▶ Regular data backups ▶ Transaction log ▶ Log file backup ▶ Redundant servers with load balancing ▶ Multiple geographically distributed database servers

Authentication and Authorization

Data protection is the prevention of access to and manipulation of data by unauthorized persons. Protective measures include procedures for the **positive identification** of a person or for the assignment of **user permissions** for specific data access as well as **cryptographic methods** for confidential data storage and transmission.

Data security means the hardware and software solutions that help to protect data from **falsification, destruction, and loss**. The relational model facilitates the implementation of reliable restrictions to ensure data protection. A major data protection mechanism in relational databases is to **provide users with only those tables and table sections they need for their work.**

To create a user account for authentication in SQL, we use the **CREATE USER command**.
The following example creates a user account for employee Anesu with a relatively secure password that does not contain words, but lowercase and uppercase letters, numbers, and special characters

```
CREATE USER Anesu IDENTIFIED BY 'jd7k_Ddjh$1';
```

Authentication and Authorization

User accounts can be modified and deleted with ALTER USER and DROP USER

The GRANT command is used to authorize users for actions on tables. The following command authorizes user Anesu for all actions on the STAFF table

GRANT ALL ON STAFF TO Anesu; -- Grants all permissions to Anesu to table STAFF

Revoke ALL on STAFF From Anesu ;-- removes all rights from Anesu

Authentication and Authorization

To simplify the assignment of rights for **several users**, reusable roles can be defined. In the following example, a role “hr” is created. This role will be authorized to perform **read and write actions** on the table STAFF.

```
CREATE ROLE hr;
GRANT SELECT, INSERT, UPDATE on STAFF to hr;
GRANT hr TO Anesu;
GRANT hr TO Peter;
.
.
```

We may want to **further restrict access** to columns and rows of a table. This is done with **table Views**, each of which is based on either one or multiple physical tables and is defined using a SELECT statement. **View security** is only effective when users are granted privileges on the views rather than on the base tables.

```
CREATE VIEW employee_summary3 AS
SELECT staff_id, first_name, last_name, salary
FROM STAFF
WHERE salary = 50000;
```

Updateable views allow for insert, delete, and update operations. The following criteria determine whether a view is updateable:

- The view contains content from only one table (no joins allowed).
- That base table has a **primary key**.
- The defining SQL expression contains no operations that affect the number of rows in the result set (e.g., aggregate, group by, distinct, etc.)

Authentication and Authorization

Using **views**, it is possible to grant only **reading privileges** for a subset of columns of the STAFF table with the employee_summary3 view

For a more selective assignment of permissions, for instance, it is possible to authorize only a certain HR employee with the user jd7k_Ddjh$1 to make changes to a subset of rows in the employee_summary3 view

```
GRANT UPDATE ON employee_summary3 TO jd7k_Ddjh$1
WITH GRANT OPTION
```

User jd7k_Ddjh$1 can now modify the employee_summary3 view and **assign** this authorization or a limited reading privilege to others and take it back.

Audit Trails and Monitoring

•Enable auditing features to track database activities and user actions. Regularly review audit logs for any suspicious activities.

•Implement **real-time monitoring** to detect and respond to security incidents promptly.

•MySQL provides an official audit plugin that you can use to capture various types of events. You can download the plugin from the MySQL website.

•Customize the audit plugin configuration to specify what events and actions you want to log. For example, you can configure the plugin to log **logins, queries, and administrative actions**

•Periodically review the audit logs to identify any suspicious activities. You can use standard log analysis tools or build custom scripts to parse and analyze the logs.

•For real-time monitoring, you can leverage third-party tools that offer MySQL monitoring capabilities. These tools often provide features such as real-time alerting, dashboard views, and historical analysis.

•In addition to the MySQL Audit Plugin, you can use **triggers to implement custom auditing** for specific tables or actions. Triggers allow you to execute custom code (e.g., logging to a separate table) in response to specific events.

Backup and Recovery

- Regularly back up the database and ensure that the backup copies are stored securely.
- Test the backup and recovery procedures to ensure data integrity and availability in case of a security incident.
- Store backup copies in a secure location that is isolated from the production environment. This can be on a separate server, cloud storage, or offline storage media.
- Encrypt backup files to protect sensitive data. MySQL Enterprise Backup, for example, supports encryption during the backup process.
- Implement scheduled backup jobs to automate the backup process.

Regularly test your backup and recovery procedures to ensure they work effectively. This involves:
•Restoring a backup to a separate test environment.
•Verifying the integrity of the restored data.
•Confirming that the restored database is functional.

MySQL supports Point-In-Time Recovery, allowing you to recover your database to a specific point in time. This is useful for recovering from user errors or data corruption.

Encryption

•Encrypt sensitive data at rest and in transit. Use technologies like Transparent Data Encryption (TDE) for data at rest and SSL/TLS for data in transit.
•Implement encryption for backups to protect data even when it is not actively in use.

•Basics of encryption
Plaintext and Ciphertext
Plaintext→ Original, readable data.
Ciphertext → Encrypted data, not easily readable without the decryption key.
Key→ An essential component of encryption. The key is used to transform plaintext into ciphertext (encryption) and vice versa (decryption).
Algorithms→ Mathematical functions that define the encryption and decryption processes. Common algorithms include Advanced Encryption Standard (AES), Triple DES (3DES), and RSA.

Symmetric and Asymmetric Encryption
Symmetric→ Uses the same key for both encryption and decryption. Faster but requires secure key exchange.
Asymmetric → Uses a pair of public and private keys. Slower but eliminates the need for secure key exchange.
•

Encryption

Encryption for Data at Rest
Transparent Data Encryption (TDE)

TDE is a technology that encrypts databases, files, and backups at the storage level. It operates transparently to applications and users.
When TDE is enabled, data is encrypted before it's written to disk and decrypted when read from disk.
The **encryption key** used by TDE is stored separately from the data, adding layer of security.

General steps involved:

1. Enable TDE on your database management system (DBMS).
2. Configure and manage **encryption keys** securely.
3. **Regularly monitor and audit TDE** settings to ensure the ongoing security of data at rest.

Encryption

Encryption for Data in Transit
SSL/TLS (Secure Socket Layer/Transport Layer Security)

SSL/TLS in the database context protects the communication channel between a client application (such as a web server or application server) and the database server. It keeps unauthorized parties from intercepting and tampering with sensitive data sent between them.

The SSL/TLS certificate is essential for authenticating the identity of the database server to the connecting clients. This step involves obtaining a certificate from a trusted Certificate Authority (CA), which verifies the legitimacy of the server.

•SSL and TLS are cryptographic protocols that provide secure communication over a computer network.
•SSL/TLS encrypts **data during transmission between a client and a server**, preventing eavesdropping and man-in-the-middle attacks.

General steps

1.Acquire an SSL/TLS certificate from a trusted Certificate Authority (CA).
2.Configure your web server or database server to use SSL/TLS.
3.Enable SSL/TLS in your application or client.

Encryption

Steps involved n acquiring certification

1. Generate CSR (Certificate Signing Request)

- Use a tool or the database server to create a CSR containing server information.

2. **Submit CSR to CA**

- Send the CSR to a chosen CA, which will issue a digital certificate after validating the server's identity.

3. **Install Certificate**

- Download the certificate from the CA and install it on the database server.
- Ensure the database server has an SSL/TLS library installed (e.g., OpenSSL).
- Modify the database server's configuration file to include SSL/TLS settings. Specify the location of the certificate and private key.
- Modify the configuration settings of the client application to use SSL/TLS for database connections.
- If required, specify the CA's certificate or bundle to verify the server's certificate during the connection.

MariaDB

A **relational database** is a type of database that uses a structure (e.g., Tables) that allows data to be organized in tables with rows and columns.

Relational Database Management System (RDBMS)

A **RDBMS is software** that provides an **interface to interact with the database**, allowing users to create, manage, and query relational databases.

Key Features

1. Data Integrity→ Ensures data accuracy and consistency through constraints like **primary keys**, foreign keys, and data types.
2. ACID Properties→ enforces **A**tomicity, **C**onsistency, **I**solation, and **D**urability to maintain the integrity of transactions.

- **A**tomicity guarantees that a transaction is handled as a single, indivisible unit of work. Either all or none of the transaction's changes are committed to the database.
- **C**onsistency ensures that each transaction moves the database from one legitimate state to another.
- **I**solation ensures that the effects of one transaction are separated from those of other concurrent transactions.
- **D**urability ensures that once a transaction is completed, its effects remain even in the face of system failures (for example, a power loss or hardware failure).

3. SQL (Structured Query Language)→ RDBMS systems use SQL as a standard language for defining and manipulating the data.
a relational database is a data model based on tables, while an RDBMS is the software that implements and manages these databases.