

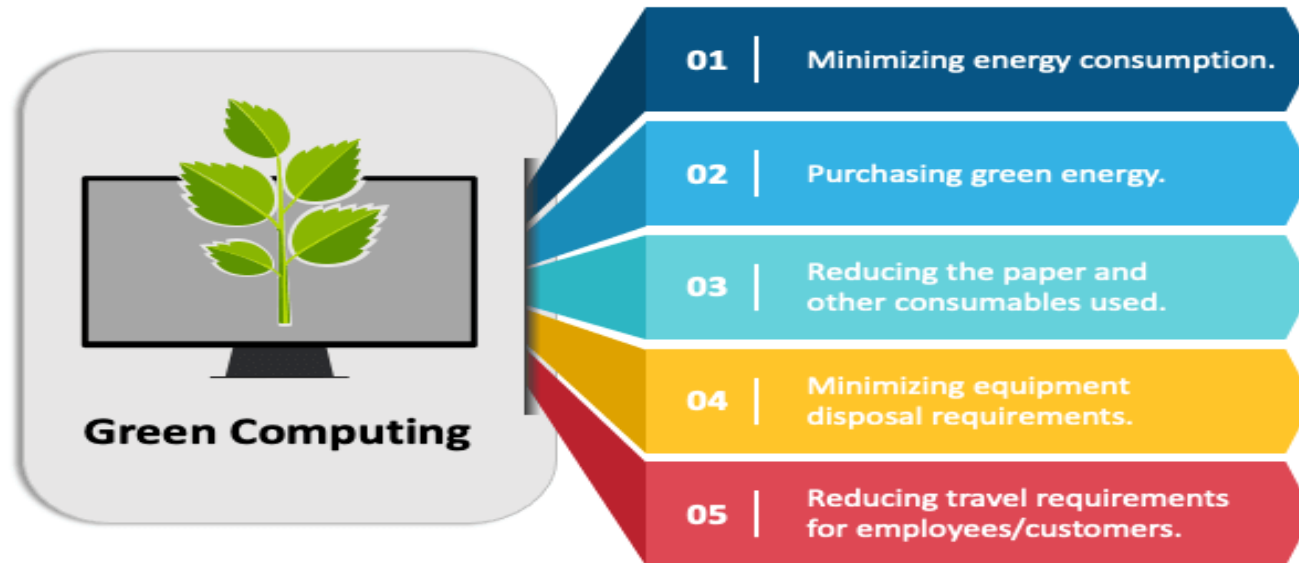
# GREEN CLOUD COMPUTING

# GREEN CLOUD

- As energy costs are increasing ,there is a need to shift focus from optimizing data center resources management for pure performance alone to optimizing for energy efficiency while maintaining high service level performance.
- We propose Green Cloud computing model that achieves not only efficient processing and utilization of computing infrastructure, but also minimize energy consumption.

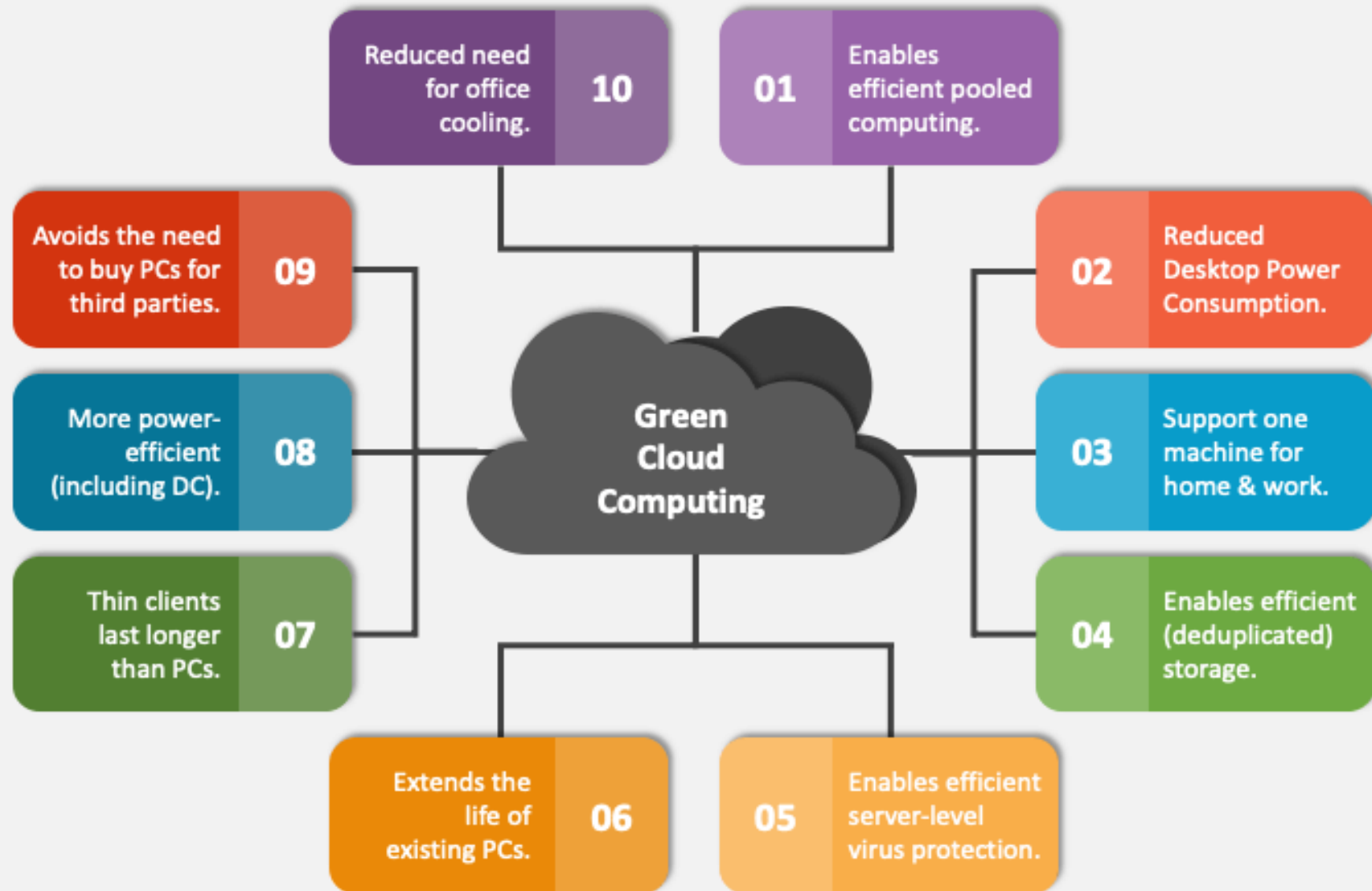
## GREEN COMPUTING

Why Green Computing is Important?



# GREEN CLOUD COMPUTING

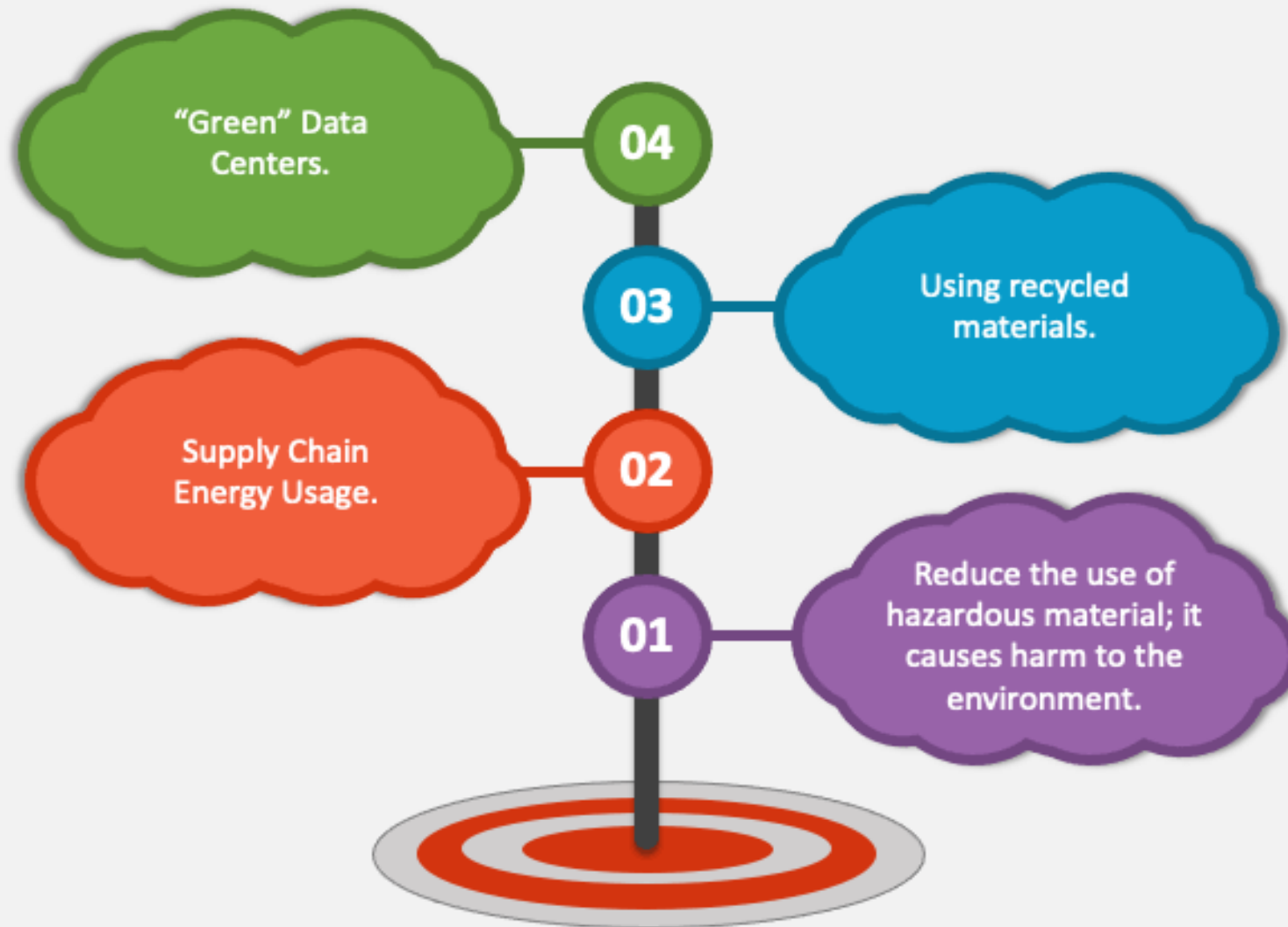
3



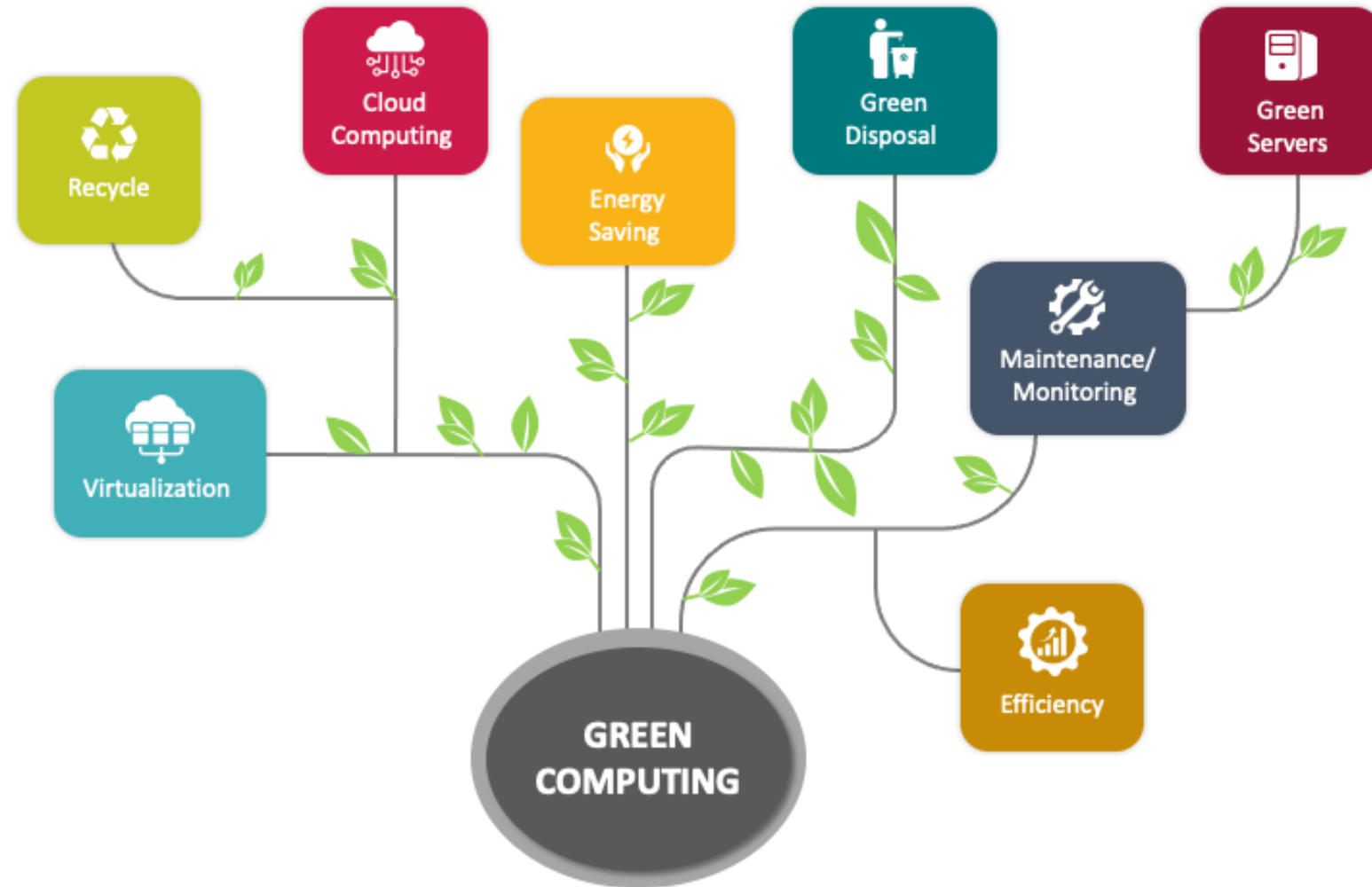
# GREEN CLOUD COMPUTING

4

## Goals of Green Cloud Computing



# GREEN COMPUTING



## **BENEFITS:**

- reduce the use of hazardous materials
- maximize energy efficiency during the product's lifetime
- promote the recyclability or biodegradability of defunct products and factory waste.
- Like transitioning from fossil fuels to renewables
- Virtualization: Virtualization is a technique that enables multiple virtual machines to run on a single physical server, allowing better utilization of hardware resources and reducing energy consumption.

# A. RENEWABLE ENERGY SOURCES FOR CLOUD-COMPUTING



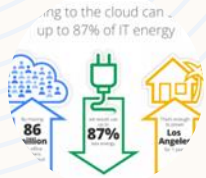
**HIGHER UTILIZATION RATE :-** Companies typically operate their own private data centers, which results in low utilization rates because the equipment is acquired and set up in preparation for increases in server demand.

Servers are operated at high utilization rates in the cloud, resulting in increased efficiency. Data centers are notorious for wasting resources by leaving equipment idle. Due to the high use of infrastructure, public cloud servers are often 2 to 4 times more efficient than traditional data



**HARDWARE REFRESH RATE :-** Due to the high prices and time required to upgrade servers, traditional data center hardware is often utilized for a lengthy period before it is upgraded or replaced. Cloud gear tends to have a shorter lifespan since it is used more often than traditional servers.

Regular upgrades of public cloud servers are also more cost-effective since new technology improves energy efficiency. Because the public cloud provider will save money by using energy-efficient technology,



**REDUCED ELECTRICITY :-** Power, cooling, and lots of electricity are required to maintain traditional data hardware systems. Electricity may be saved by moving simple software programs to the cloud.

Moving corporate software such as email, CRM, and more to the cloud (on a national scale) may save enough electricity each year to light Los Angeles for 12 months, according to a case study by the Lawrence Berkeley National Laboratory. 87 percent of these software applications will use less energy if they are hosted on the cloud.

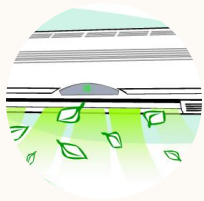


# CARBON FOOTPRINT REDUCTION TECHNIQUES

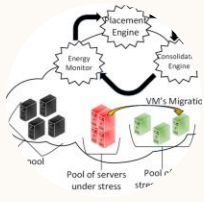
9



**Virtualization:-** By virtualizing servers, companies can reduce the number of physical servers required, which in turn, reduces energy consumption and carbon emissions.



**Renewable energy sources:-** Companies can switch to renewable energy sources like wind or solar power to power their data centers. This significantly reduces the carbon footprint of cloud computing.



**Energy-efficient hardware:-** By using energy-efficient hardware like low-power processors, companies can reduce energy consumption and carbon emissions.

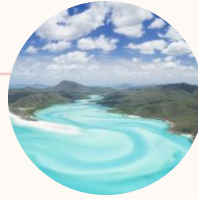
# IMPACT OF CLOUD COMPUTING ON CLIMATE CHANGE

10



## REDUCE

cloud computing can potentially reduce the carbon footprint of businesses and organizations by enabling them to shift their IT infrastructure from on-premise data centers to more energy-efficient cloud data centers



## EMISSIONS

Cloud providers can achieve economies of scale by consolidating computing resources, reducing energy consumption and carbon emissions per unit of computing power.



## EFFICIENT

cloud computing can also help reduce paper usage and transportation emissions by enabling remote collaboration and file-sharing, reducing the need for physical document printing and delivery

# CLOUD COMPUTING AND GLOBAL WARMING

11



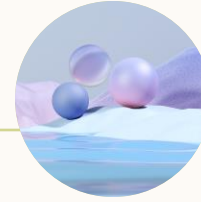
## ENERGY EFFICIENCY

Cloud computing can help reduce energy consumption by allowing businesses and individuals to share resources, such as servers and storage, instead of running their own data centers. Cloud providers can optimize energy use by consolidating workloads and using energy-efficient hardware and cooling systems.



## CARBON EMISSIONS

Data centers are energy-intensive facilities that require significant amounts of electricity to power and cool the equipment. This can result in a significant carbon footprint, especially if the data center is powered by fossil fuels.



## RENEWABLE ENERGY

Many cloud providers are committing to using renewable energy sources, such as wind and solar, to power their data centers. This can help reduce the carbon footprint of cloud computing.

# GREEN CLOUD COMPUTING POLICIES AND REGULATIONS

To ensure that cloud computing practices are environmentally sustainable.



## ENERGY STAR CERTIFICATION

Certifies energy-efficient products and practices.



## GREENHOUSE GAS PROTOCOL

Global accounting tool for measuring and managing greenhouse gas emissions.



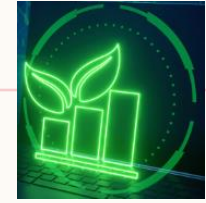
## THE KYOTO PROTOCOL

International treaty aimed at reducing greenhouse gas emissions.



## THE PARIS AGREEMENT

International agreement aimed at combating climate change. Cloud service providers can support the Paris Agreement by adopting green cloud computing practices and reducing their carbon footprint.



## GREEN COMPUTING IMPACT ORGANIZATION

Non-profit organization that provides certification and training for green computing practices.

# CLOUD NATIVE SYSTEMS

# CLOUD NATIVE SYSTEMS

14



## WHAT ARE CLOUD NATIVE SYSTEMS?

A cloud-native system is a software application that is designed to run natively on cloud infrastructure. It is built using cloud-native principles such as microservices, containers, and serverless computing.



## SCALABLE

Cloud-native systems are designed to be scalable. This means that they can easily handle a large volume of traffic and users.

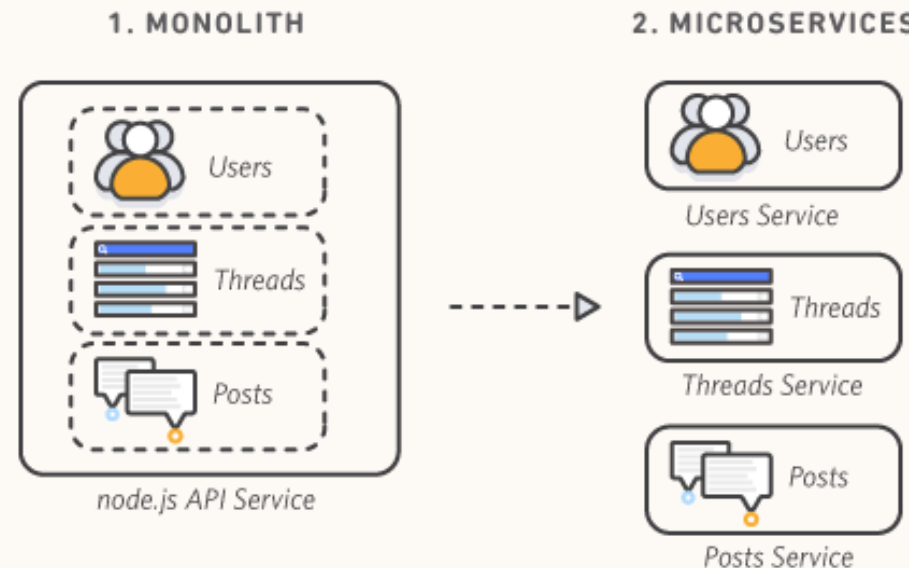


## COST EFFECTIVE

Cloud-native systems are cost-effective because they use cloud infrastructure, which reduces the need for expensive hardware and maintenance.

# MICRO-SERVICE BASED ARCHITECTURE

- **Microservices** is the process of dividing an application into **different modules** that perform a particular function and that may be **independent** of other modules. These modules have well-defined interfaces and are deployed individually. Microservices are **self-contained**.
- A **microservices architecture** is a type of application architecture where the application is developed as a collection of services. It provides the framework to **develop, deploy, and maintain** microservices architecture diagrams and services independently.
- Used to **speed up** application development.



# MICROSERVICES

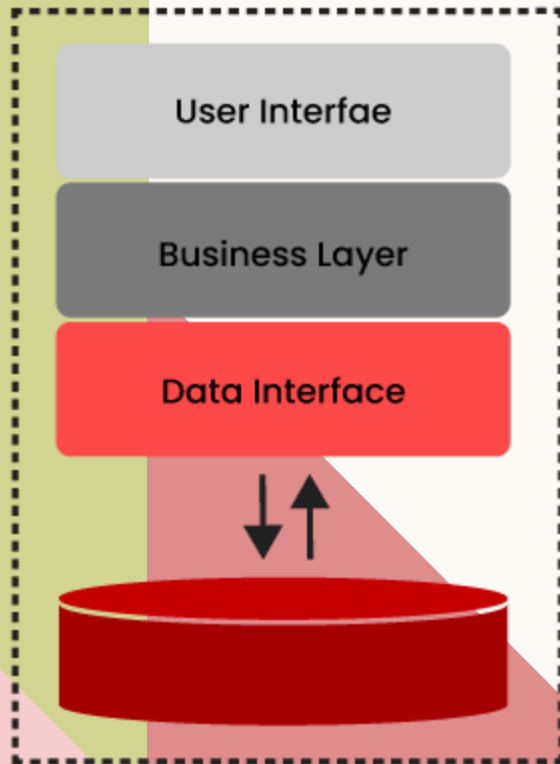
16

**Flexibility & scalability**

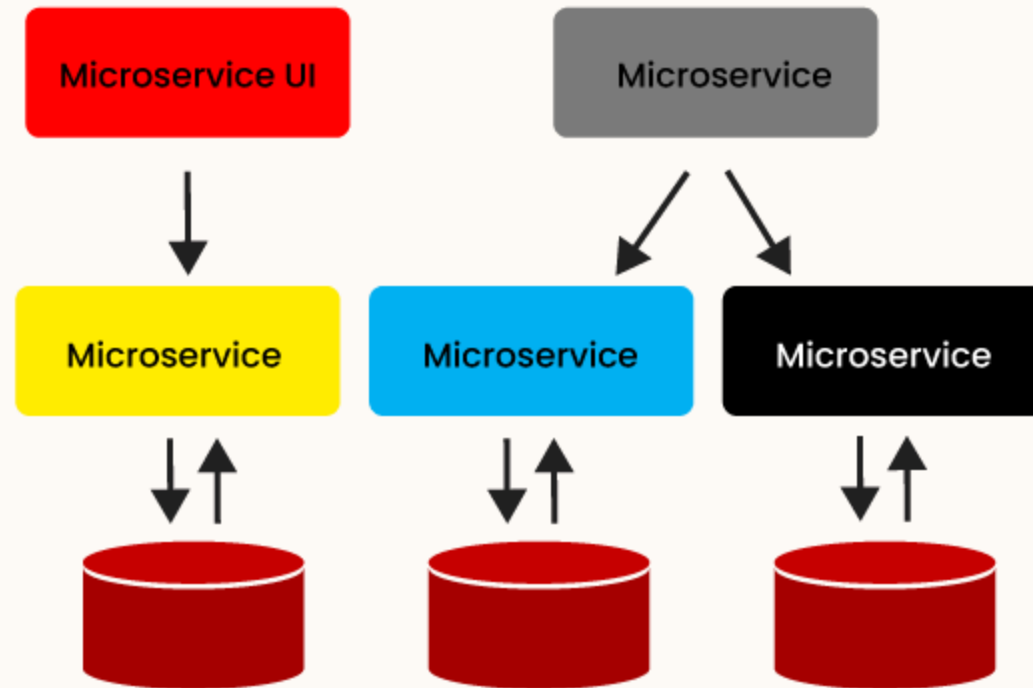
**Easy & Fast Deployment**

**Agility & Reusable code**

Monolithic Architecture

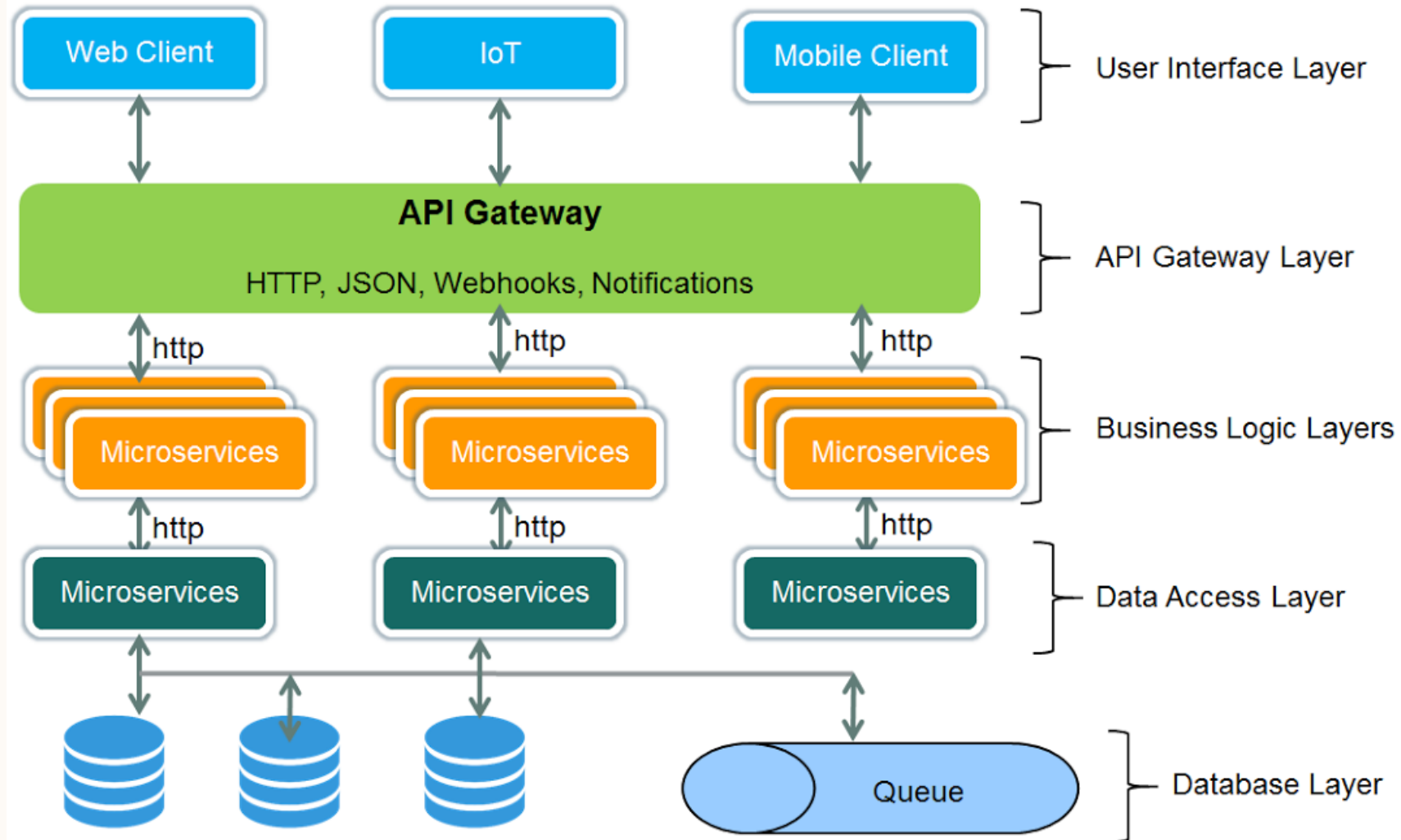


Microservices Architecture





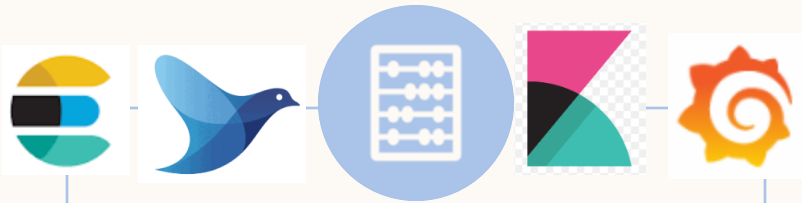
# Software Architecture for Cloud Native Apps



## Benefits

- ❖ Cloud Native Apps are built on Microservices
- ❖ Microservices enables rapid DevOps
- ❖ Each Microservice can scale independently
- ❖ Microservice can be modified independently
- ❖ Built for multi-tenant Infrastructure
- ❖ Microservices run on containers or Serverless Functions
- ❖ Mobile, IoT, Web are all first class citizens

# TOOLS AND TECHNIQUES FOR DEBUGGING AND TROUBLESHOOTING COMPLEX CLOUD NATIVE SYSTEMS



## LOGGING AND MONITORING

- Can help in identifying and diagnosing issues in real-time.
- Provide insights into the performance and behavior of the system
- Eg. Elasticsearch, Fluentd, and Kibana (EFK) stack, Prometheus, and Grafana.



## DISTRIBUTED TRACING

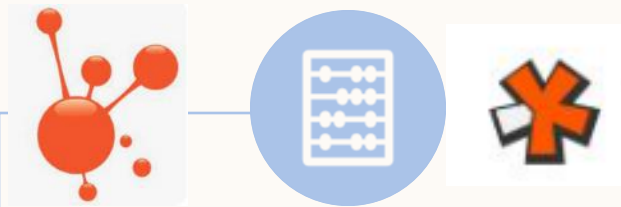
- Allows developers to track and visualize requests as they move through a system.
- Examples include OpenTelemetry, Zipkin, and Jaeger.



## LOAD TESTING

- Identifying system performance bottlenecks and issues related to scalability. Examples include Apache JMeter, Gatling, and Locust.

# TOOLS AND TECHNIQUES FOR DEBUGGING AND TROUBLESHOOTING COMPLEX CLOUD NATIVE SYSTEMS



## CODE PROFILING

- Help in identifying performance issues related to specific sections of code.
- Provides insights where optimization is needed
- Examples include VisualVM, YourKit, and Java Flight Recorder.



## DEBUGGING TOOLS

- Help in identifying and diagnosing issues in code.
- Examples include IntelliJ IDEA, Eclipse, and Visual Studio Code.

## CONTAINER ORCHESTRATION PLATFORMS

- Help in managing complex cloud native systems.
- Examples include Kubernetes, Docker Swarm, and Apache Mesos.

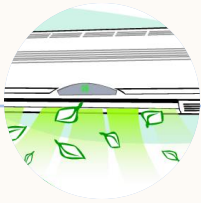
# TOOLS AND TECHNIQUES FOR DEBUGGING AND TROUBLESHOOTING COMPLEX CLOUD NATIVE SYSTEMS



## COLLABORATION AND COMMUNICATION TOOLS

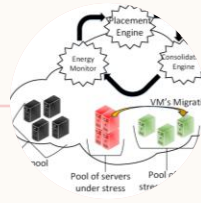
- Can help in facilitating communication between developers and other stakeholders
- Examples include Slack, Zoom, and Microsoft Teams.

# SECURITY CHALLENGES IN COMPLEX CLOUD-NATIVE SYSTEMS



## CONTAINER SECURITY

- Containers are a fundamental building block of cloud-native systems, and ensuring their security is critical. However, vulnerabilities in containers and container runtimes can result in a significant security breach.



## ACCESS CONTROL:

- As cloud-native systems become more complex, managing access control becomes more challenging. Cloud-native systems are composed of multiple components and services, and managing access to each of these services can be difficult.



## DATA PROTECTION:

- Data breaches can be catastrophic, and as cloud-native systems grow more complex, the number of potential vulnerabilities increases. Data protection is essential in cloud-native systems to prevent data breaches.

## **IMPACT OF COMPLEXITY ON PERFORMANCE OF CLOUD-NATIVE SYSTEMS**

- Cloud-native systems are built on complex architectures that enable scalability and flexibility, but this complexity can also impact performance.
- Complexity in cloud-native systems can arise from factors such as distributed architecture, microservices, and containerization.
- As cloud-native systems scale, their complexity can result in performance issues such as increased latency, slower response times, and decreased throughput.
- To mitigate the impact of complexity on scalability and performance, cloud-native systems require advanced monitoring and management tools that can provide insights into system behavior.

**THANK YOU**