Distributed Systems and Transparency

1. Distributed Systems

A **distributed system** is a collection of independent computers or devices that work together over a network and appear to users as a single unified system. Although each component has its own memory, processing power, and possible failures, coordination mechanisms allow the system to provide shared services, data, and computation transparently.

2. Transparencies in Distributed Systems

Access Transparency

Access transparency hides differences in data representation and how resources are accessed, allowing users to interact with local and remote resources in the same way. Users do not need to know network protocols, APIs, or storage locations. This improves usability and adoption.

Example:

Google Drive allows users to open, edit, and save files exactly like local files on their computer, even though the files are stored on remote servers.

Location Transparency

Location transparency ensures that users do not need to know where a resource is physically located. Resources can be moved or replicated without affecting users.

Example:

When using WhatsApp, users send messages without knowing which country or data center stores the chat history.

Replication Transparency

Replication transparency hides the existence of multiple copies of data or services. It Improves fault tolerance and performance while avoiding confusion from duplicate data.

Example:

Google Drive replicates files across multiple servers for reliability, but users see only one file.

Failure Transparency

Failure transparency masks system failures so users experience minimal disruption. It increases reliability and user trust in the system.

Example:

If one WhatsApp server fails, users are often redirected to another server without noticing.

Concurrency Transparency

Concurrency transparency allows multiple users to access shared resources simultaneously without interference. It prevents data corruption and ensures consistent system behavior.

Example:
Multiple users can edit the same Google Docs file at the same time, and changes are merged consistently.

3. Transparency Implemented Well by Selected Systems

WhatsApp – Failure Transparency

WhatsApp handles server outages and network instability gracefully. Messages are queued and delivered once connectivity is restored, often without the user noticing backend failures. Automatic reconnection and message retry mechanisms hide failures effectively.

Google Drive – Replication Transparency

Google Drive excels at replication transparency by maintaining multiple copies of files across data centers while presenting a single consistent version to users. Users never see or manage replicas, yet benefit from high availability and durability.

Bitcoin – Location Transparency

Bitcoin users send and receive transactions without knowing the physical locations of nodes validating them. The peer-to-peer architecture hides node locations, enabling decentralization.

4. Basic Distributed System Architecture

```
      User Devices
     ----------------
     |    Phone     |
     |    Laptop    |
     ----------------
            |
     Internet / Network
            |
   --------------------------------------------------
   |              |              |            |
 Server A  Server B  Server C   Server D
 (Replica) (Replica) (Replica) (Backup)
```

5. Conclusion

Distributed systems are fundamental to modern computing, enabling scalability, reliability, and global access. Transparency plays a critical role by hiding complexity and failures, allowing users to focus on functionality rather than infrastructure.