

Project Computer Networks Application Architectures

Presented by:
Retaj Al-Rasheed
Balqees Al-Arjan
Raghad Al-Suhaim

Project Supervisor : Dr.Turki Alghamdi

TABLE OF CONTENT

01 Introduction

02 History

03 Application
Architecture

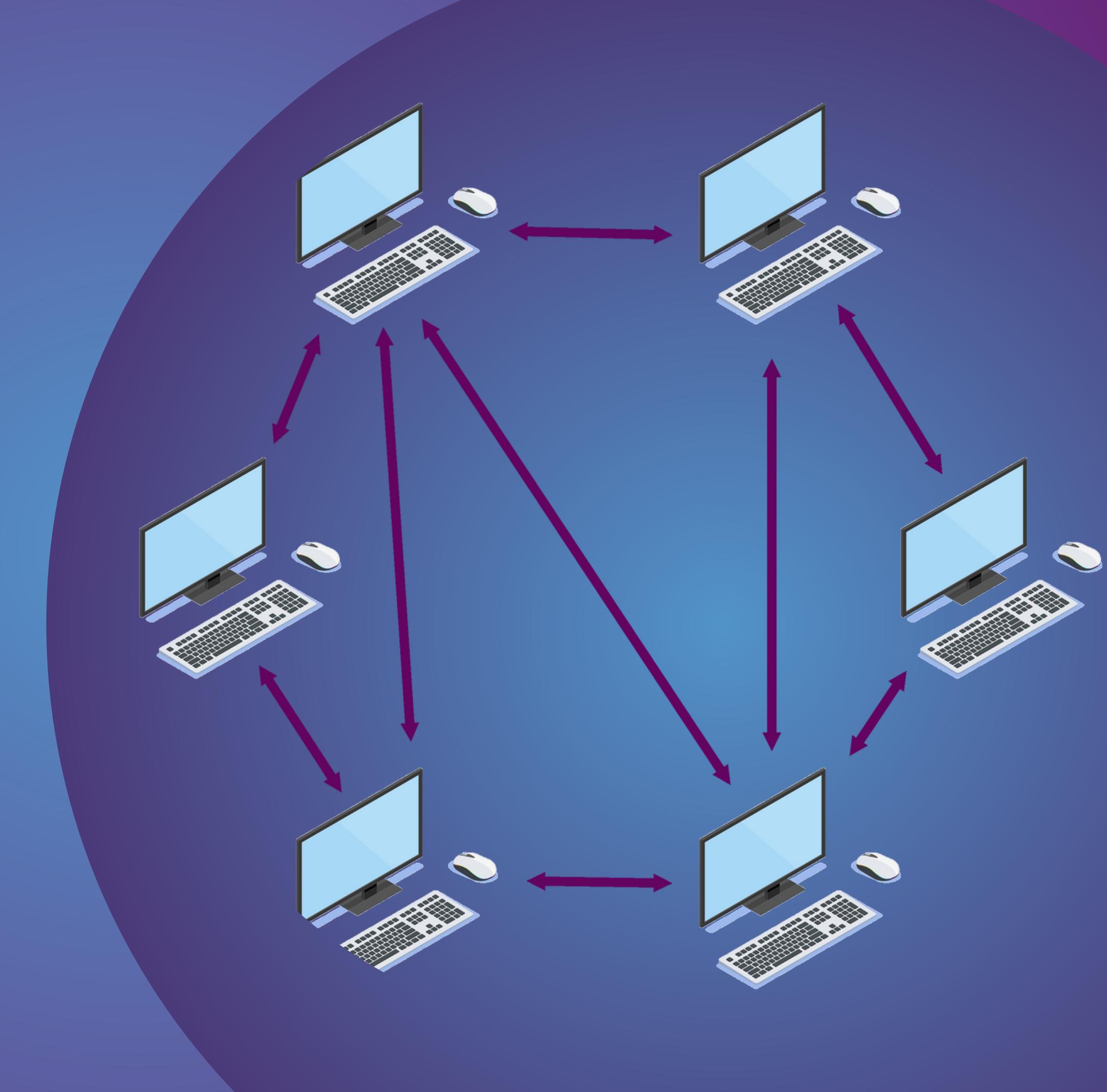
04 Problem
Description

05 Solutions

06 Conclusion

Introduction

Application architecture involves the high-level structure and design principles that govern the organization and interaction of software components within an application. It includes the selection of architectural patterns, frameworks, and protocols that determine how modules and components are organized and communicate with each other.



Why did they came out with Application layer?

The application layer was invented to provide a standardized framework for network applications, simplifying development, ensuring interoperability, handling data formatting, providing application-specific functionality, supporting user interfaces, and implementing application security.

When did they came out with Application layer?

The application layer was included in the TCP/IP protocol suite from its initial development in the **1970s**. Specific protocols and standards associated with the TCP/IP application layer, like HTTP, SMTP, FTP, and DNS, were developed in the following years.

Application Architectures

P2P Architectures

here is minimal (or no) reliance on dedicated servers in data centers. Instead the application exploits direct communication between pairs of intermittently connected hosts, called peers.

peer-peer

- no always-on server
- directly communicate
- demands peers are intermittently connected
- IP addresses change

client-server Architectures

client-server architecture, there is an always-on host, called the server, which services requests from many other hosts, called clients.

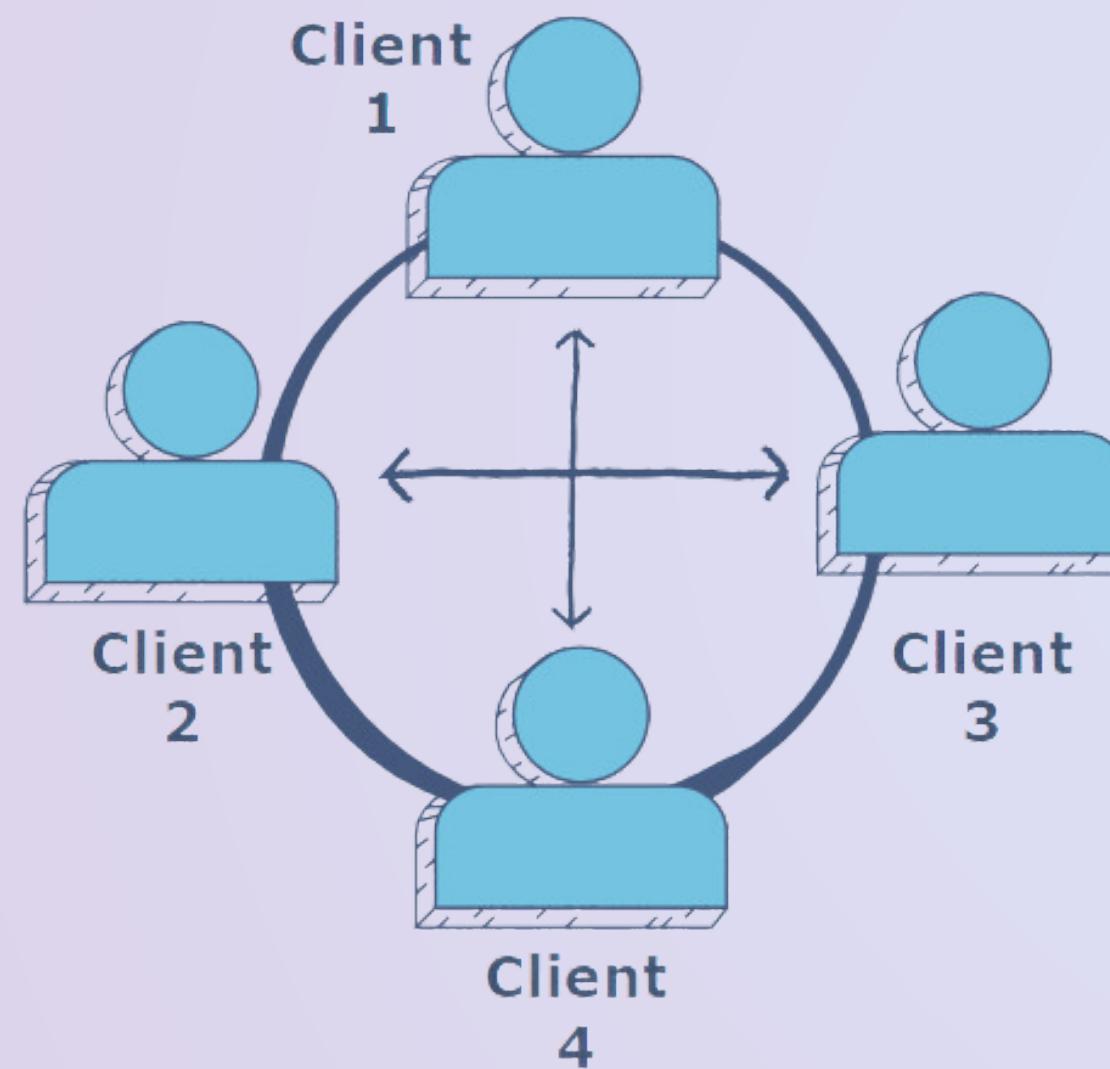
server

- always-on host
- permanent IP address
- data centers for scaling

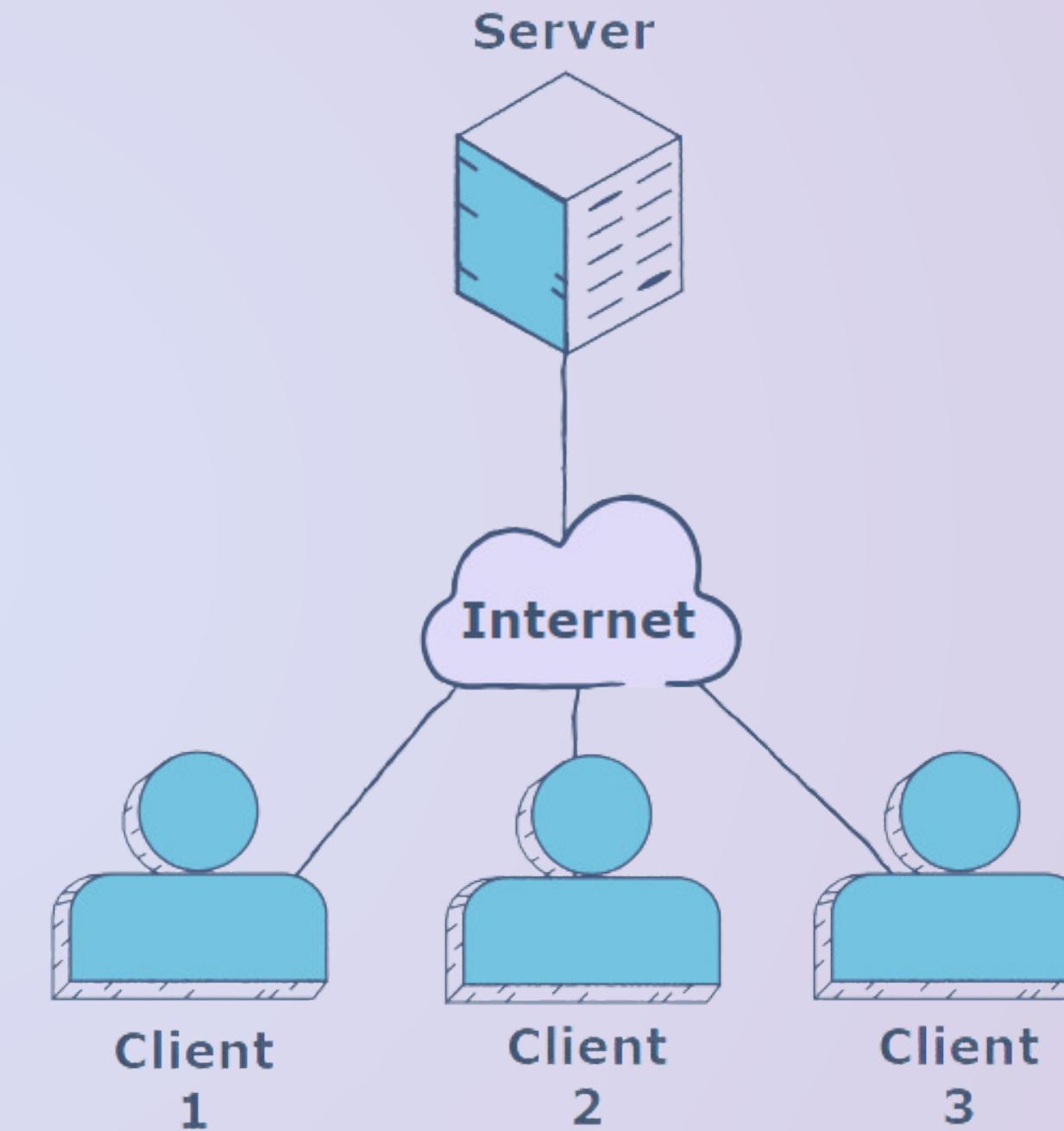
client

- communicate with server
- intermittently connected
- dynamic IP addresses
- do not communicate directly

Peer-Peer Model



Client-Server Model



Protocols in Application Layer

The application layer provides several protocols for manipulating the data which actually enables any type of user to access the Network and allows any software to easily send and receive information and present meaningful data to its users.

The most important protocols provided by the application layer include:

- **HyperText Transfer Protocol (HTTP)**: Web's application-layer protocol, is at the heart of the Web. provides for Web document request and transfer.
- **File Transfer (FTP)**: provides for the transfer of files between two end systems.
- **SMTP**: is at the heart of Internet electronic mail. transfers messages from senders mail servers to the recipients' mail servers.
- **domain name system (DNS)**: to translate user-supplied hostnames to IP addresses.

Problem

Solution

Standardisation

Lack of consistency and interoperability between applications developed by different vendors.

Each application had its own proprietary communication methods and protocols, making it difficult to communicate and exchange data seamlessly.

Standardised protocols and services at the application layer enable effective communication and data exchange between applications from different vendors.

This promotes compatibility, simplifies development, and ensures reliable communication across platforms and networks.

Problem

Solution

Encapsulation and Data Formatting

Applications generate data in different formats and structures, leading to challenges in accurate information exchange and data interpretation.

The application layer encapsulates data into standardized message formats, including headers, metadata, and formatting according to protocol specifications.

This ensures proper interpretation and understanding of data by promoting interoperability and reliable data exchange across different formats and structures.

Problem

Establishing effective and seamless client-server communication

This problem refers to the challenge of ensuring smooth and efficient communication between clients (such as web browsers or mobile apps) and servers (where the application logic and data reside).

Establishing reliable and responsive communication channels is crucial for successful data transfer, request handling, and overall application performance.

Solution

- Implementing well-defined communication protocols such as HTTP.
- Utilizing message queues or event-driven architectures to decouple clients and servers.

Problem

Dealing with increasing complexity and number of client requests

As applications grow and gain popularity, they face a higher volume of client requests, which can become complex and diverse. This problem involves handling a wide range of client requests efficiently, managing their complexity, and ensuring that the application can effectively process and respond to them in a timely manner.

Solution

- Applying modular design principles to break down the application into smaller, manageable components.
- Utilizing the microservices architecture to isolate different functionalities and handle client requests independently.

Implementing load balancing techniques to distribute client requests across multiple servers.

Problem

Solution

Ensuring smooth interaction between clients and servers

This problem relates to providing a seamless experience for users interacting with the application. It involves designing efficient interfaces, managing user interactions, and handling data transfers between clients and servers without interruptions, delays, or inconsistencies.

- Implementing caching mechanisms to store frequently accessed data and reduce server load.
- Optimizing database queries and utilizing indexing techniques to improve database performance.

Conclusion

In conclusion, Application architecture involves the high-level structure and design principles that govern the organization and interaction of software components within an application. It includes the selection of architectural patterns, frameworks, and protocols that determine how modules and components are organized and communicate with each other. Thank you for your attention, and I hope this presentation has provided valuable insights into the significance of application architecture.

References

Ø1 High-Performance Browser Networking-by Ilya Grigorik

Ø2 Computer Networking: A Top-Down Approach" by James F. Kurose and
Keith W. Ross.

THANK YOU!