

Instructions

Introduction to Databases and Networking

Databases and Interfaces

Matthew Pike & Yuan Yao

University of Nottingham Ningbo China (UNNC)

Overview

- Introduction to databases - definition, uses, problem to solve
- Database management systems - improved centralised data management
- DBMS architecture - ANSI/SPARC model
- Networking basics - TCP/IP, IP addresses, DNS, URLs and HTTP

Databases

- *“A collection of data arranged for ease and speed of search and retrieval.”*
 - American Heritage Science Dictionary
- *“A structured set of data held in computer storage.”*
 - Oxford English Dictionary
- *“One or more large structured sets of persistent data, usually associated with software to update and query the data.”*
 - Free Online Dictionary of Computing

Databases are (virtually) everywhere!

- Library catalogues
- Medical records
- Bank accounts
- Stock market data
- Personnel systems
- Product catalogues
- Telephone directories
- Train timetables
- Airline bookings
- Credit card details
- Student records
- Customer histories
- Stock market prices
- Pretty much every website you ever use!
- This list is not exhaustive

"We want to store our data in a format that allows us to easily query, update, insert and delete without affecting the integrity of the data."

- **Example:** Student Records
 - **Insert:** New Students
 - **Update:** Students details change
 - **Delete:** Student leaves
 - **Query:** How many students study CS?



Figure 1: How would you find your student record in this pile of papers?

- Applications store their data in files.
- Each file has its own format defined by the application itself.
- Program has to know format.
- Any other program using the file has to know format.

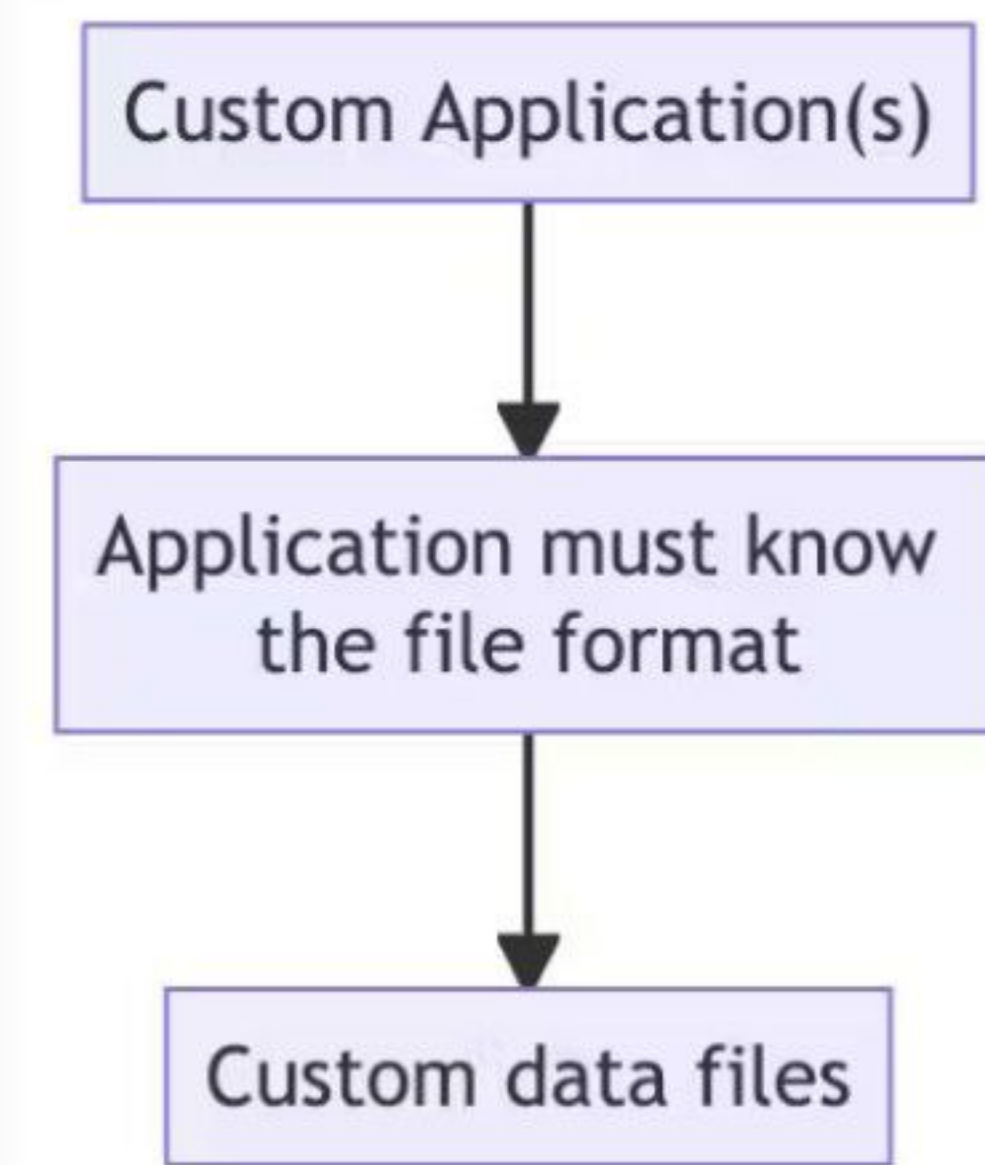


Figure 2: An example of an application using a bespoke data storage format.

What potential issues and challenges arise from using a bespoke (custom made) data storage format?

70 responses

data leak

it's hard to find the file you want

Not a standard format and may cause other developer a problem(chance of misunderstanding)

double stored data and misplaced data

Easy to be attack (?) information exposure?

disorder

The files do not fit the program

do not know the personal format

There will be no storage standards, so cross-app/online data retrieving will be difficult

What potential issues and challenges arise from using a bespoke (custom made) data storage format?

70 responses

it's hard to share data between applications

not flexible for developers

privacy risk

difficult to dispise

Misread the file or data

Choosing the correct amount of data to store,
having enough protection from external attacks

security

terrorist attack

lost data due to system failure

What potential issues and challenges arise from using a bespoke (custom made) data storage format?

70 responses

Other applications cannot recognize them. If we update the data in other applications, data in other systems could still be outdated.

The data that being storage could have mutiple different formats.

Customs' privacy. The storage safety

it may error in another computer system

hard to be popular since it is bespoke without popularity

security

using wrong sets of data

The files can only be read in some specific operate systems.

maybe fail to compile on another device

What potential issues and challenges arise from using a bespoke (custom made) data storage format?

70 responses

flexie

information leakage

information leakage

There might be some problem when fitting others programs or systems

It's bespoke which means it cannot be easily used by others

flexible

hard to reuse. expensive

not covering essential information

Might not be compatible with other codes

What potential issues and challenges arise from using a bespoke (custom made) data storage format?

70 responses



The stoge safety

not uniform format

Cannot talk with other partner/no ability to cross databases, a translating layer is needed like an API

The database itself might be out of date but hard to upgrade

time creating one

Simultaneous

Might not be optimal and some legacy issues.

Not fit for all formats of files

What potential issues and challenges arise from using a bespoke (custom made) data storage format?

70 responses

GGMU !

low efficiency to find the data files.

don't have the same formats

unprocessed data update

summarising the information

Information lost Difficult to make a change

unstandard answer

hard to integrate

If the data has multiple formats, the database cannot store all of them

What potential issues and challenges arise from using a bespoke (custom made) data storage format?

70 responses

handly to find the data which you want to search

some elders may unable to use this thing

Refresh delay

some format may not fit

It is not available for translation to other languages

Very difficult to navigate and computationally heavy

security and privacy issues

Date leakage and the stability of using a bespoke data storage format

Complex operating system

What potential issues and challenges arise from using a bespoke (custom made) data storage format?

70 responses

stack overflow

Complicated data structures, for example stack, queue

Bespoke formats may not be compatible with standard data interchange formats, making it difficult to share data with external systems or tools.

stack overflow

too expansive to restore

security

Qqq

- No standards.
- Incompatible file formats.
- Data duplication.
- Data dependence.
- Fixed queries.
- Concurrency.
- Security.
- No theoretical foundations.

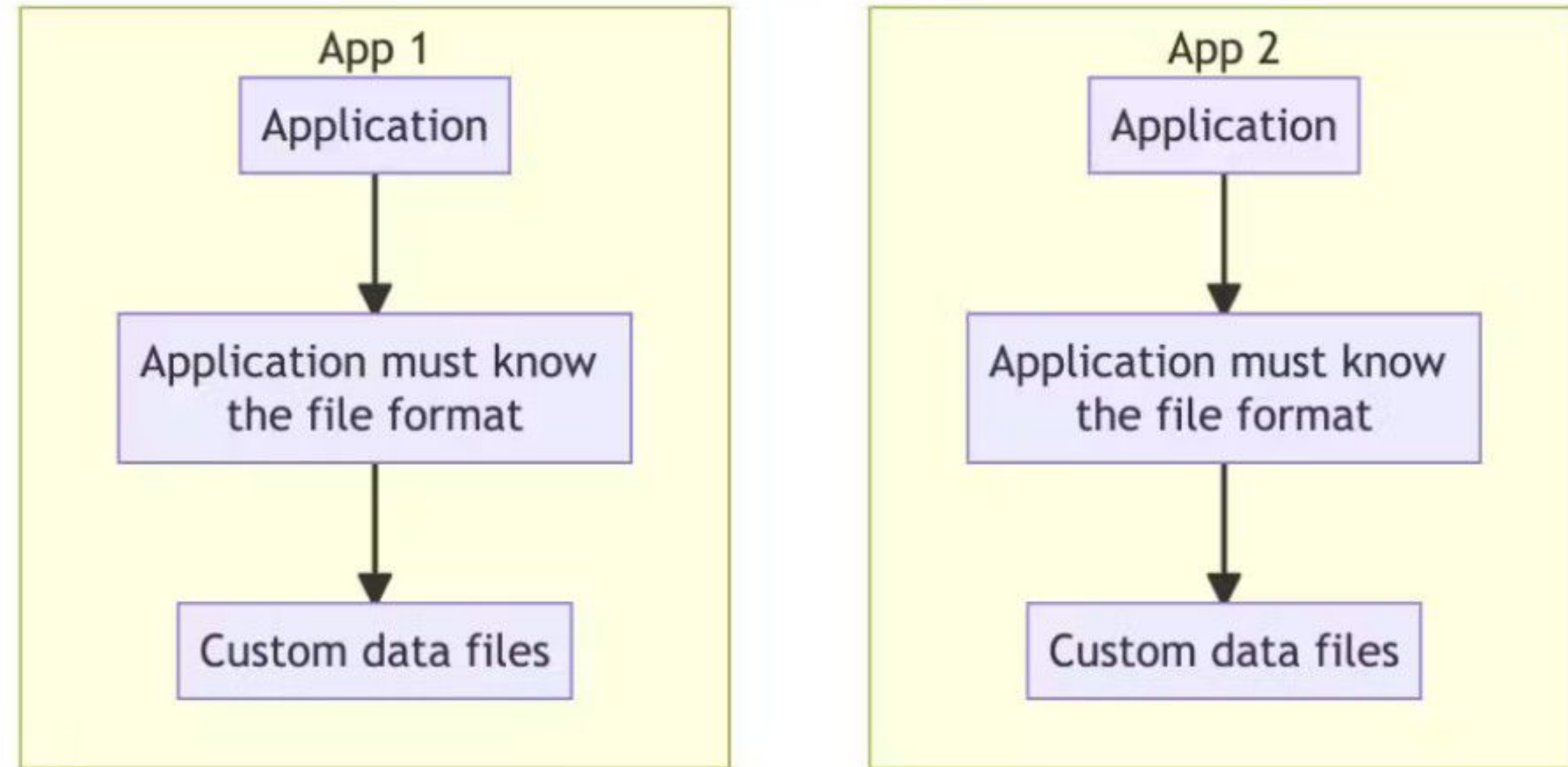


Figure 3: In this less-than-ideal approach, each application needs to maintain its own data storage format. This significantly adds to the Software Engineering load borne by the developers.

- A database management system (DBMS) is system software for creating and managing databases.
- It provides:
 - Data Definition Language (DDL) - defines data structures
 - Data Manipulation Language (DML) - inserts, updates, deletes data
 - Data Control Language (DCL) - controls access to data
 - Data Query Language (DQL) - queries and reports on data
- Structured Query Language (SQL) is used for data definition, manipulation and queries.
- A DBMS provides centralised data management and abstracts storage details.
- This improves data consistency, integrity, security, and access.

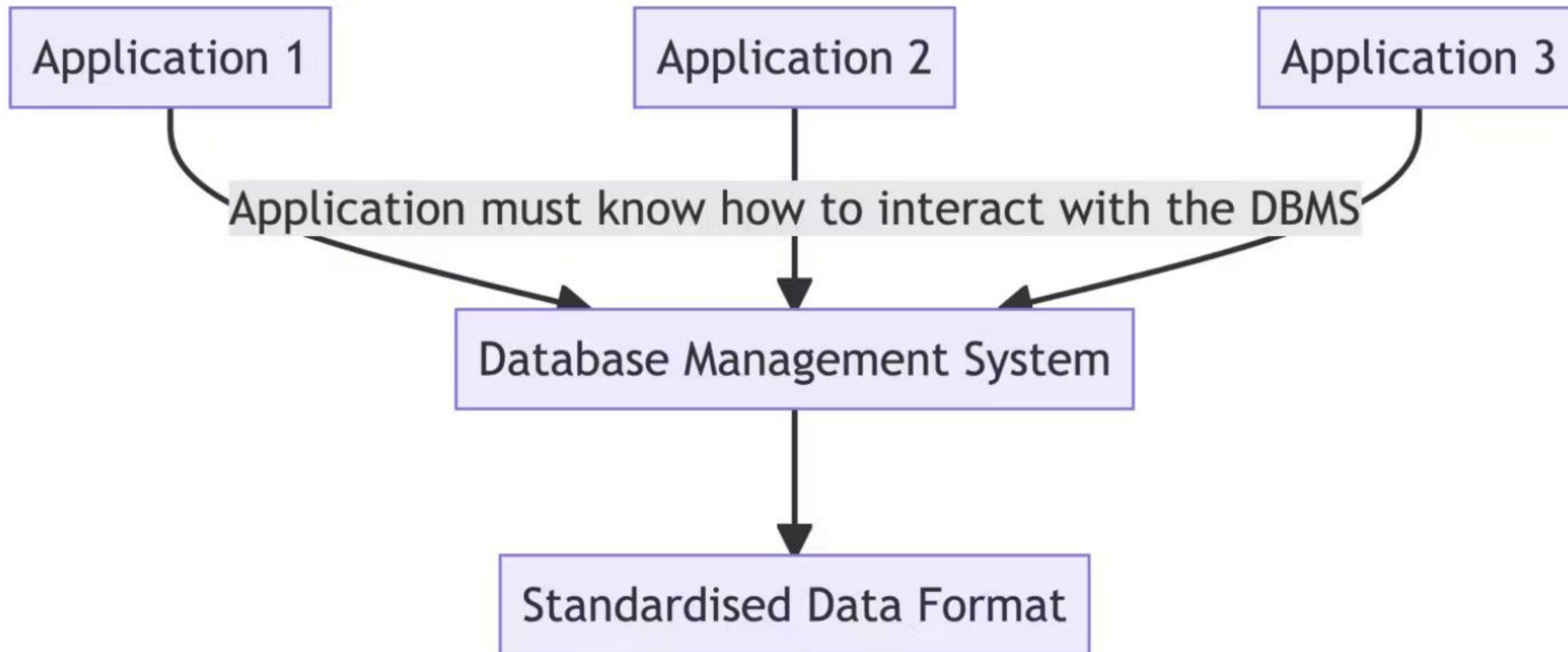
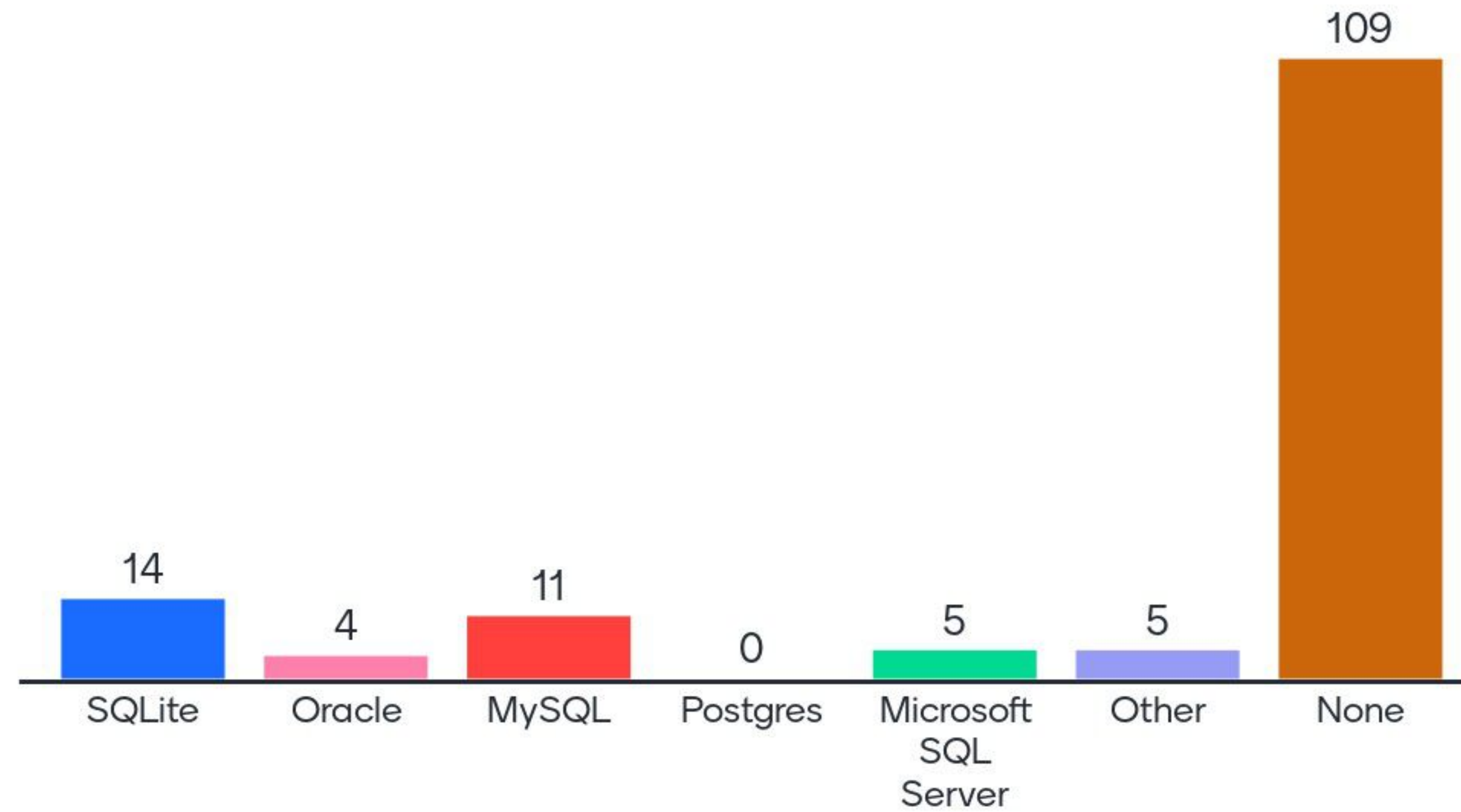


Figure 4: Instead of creating bespoke file formats, applications can now utilise a standard interface, delivered by the DBMS, for the Creation, Reading, Updating and Deletion of application data.

- Allow users to store, retrieve and update data.
 - Ensure either that all the updates corresponding to a given action are made or that none of them is made.
 - Ensure that DB is updated correctly when multiple users are updating it concurrently.
 - Recover the DB in the event it is damaged in any way.
 - Ensure that only authorised users can access the DB.
 - Be capable of integrating with other software.
- Examples of Modern DBMS
 - SQLite - <https://www.sqlite.org/index.html>
 - Oracle - <https://www.oracle.com/database/>
 - MySQL - <https://www.mysql.com/>
 - PostgreSQL - <https://www.postgresql.org/>
 - Microsoft SQL Server - <https://www.microsoft.com/en-us/sql-server/>

Have you used any of the following DBMSs already?



- Three tier/level architecture:
 - **External level:** This is the level that database users interact with. It presents data in a way end users can understand, abstracting unnecessary complexity.
 - **Conceptual level:** This part is primarily for database designers. It provides a holistic view of the entire database independent of the physical implementation, hiding details like storage locations and data pathways.
 - **Internal level:** This is meant for the system designers and deals with the physical storage of data. Working behind the scenes, it focuses on how data is stored and accessed, optimising for speed and efficiency.
- The goal is to separate the user-facing external views from the physical database implementation details.
 - This provides abstraction and independence between the levels, allowing changes to be made at one level without affecting the others.

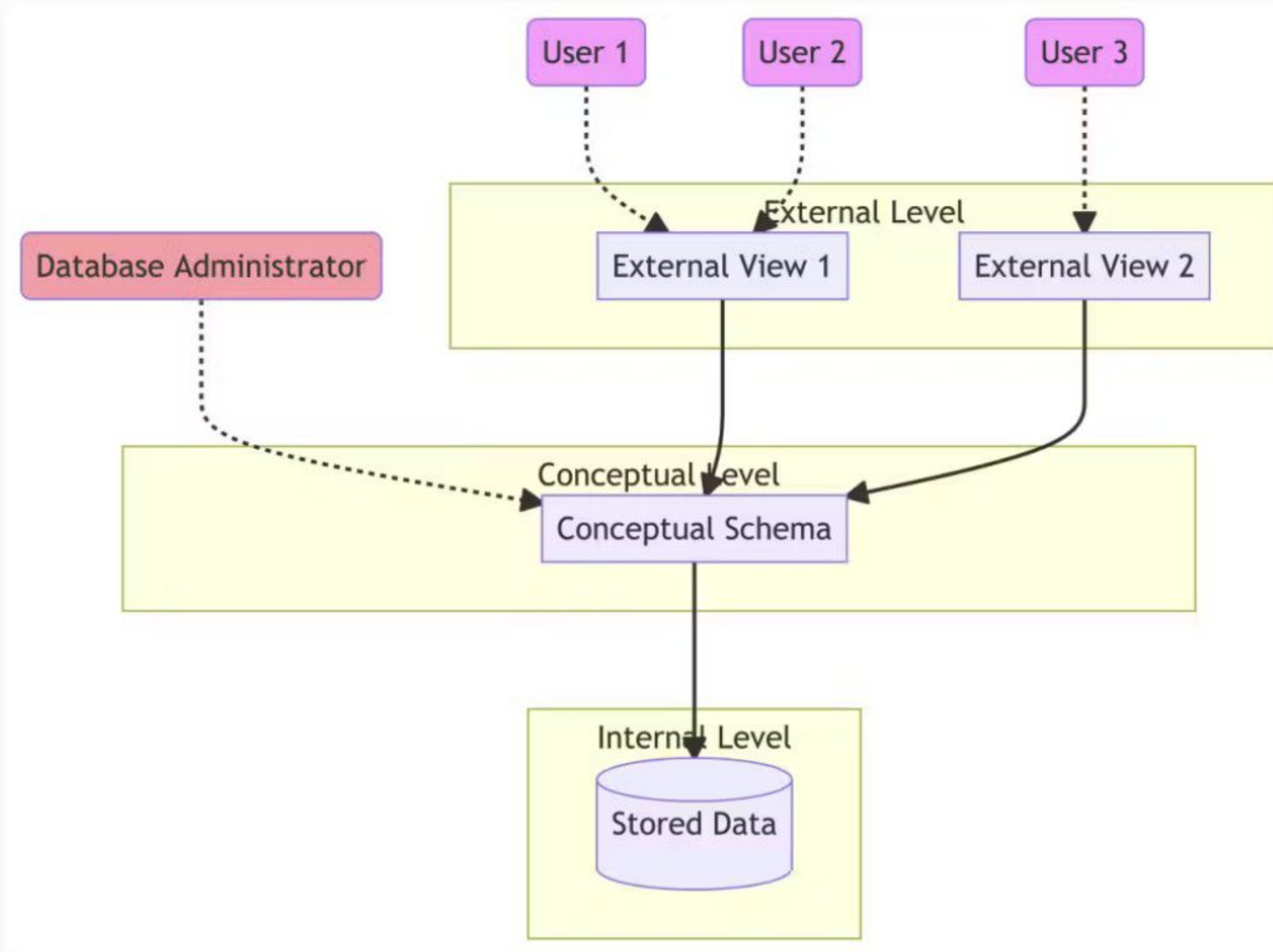


Figure 5: The ANSI/SPARC Architecture.

Networking

We will not be covering the OSI model in this module. It will be covered in the SYS module.

- Transmission Control Protocol/Internet Protocol (TCP/IP) allows devices on the Internet to communicate with each other.
- Internet Protocol (IP) addresses uniquely identify devices connected to the Internet.
 - IP addresses are four-part numbers (e.g. 192.168.1.1)
- This module will primarily operate at the Application Layer of the OSI model.

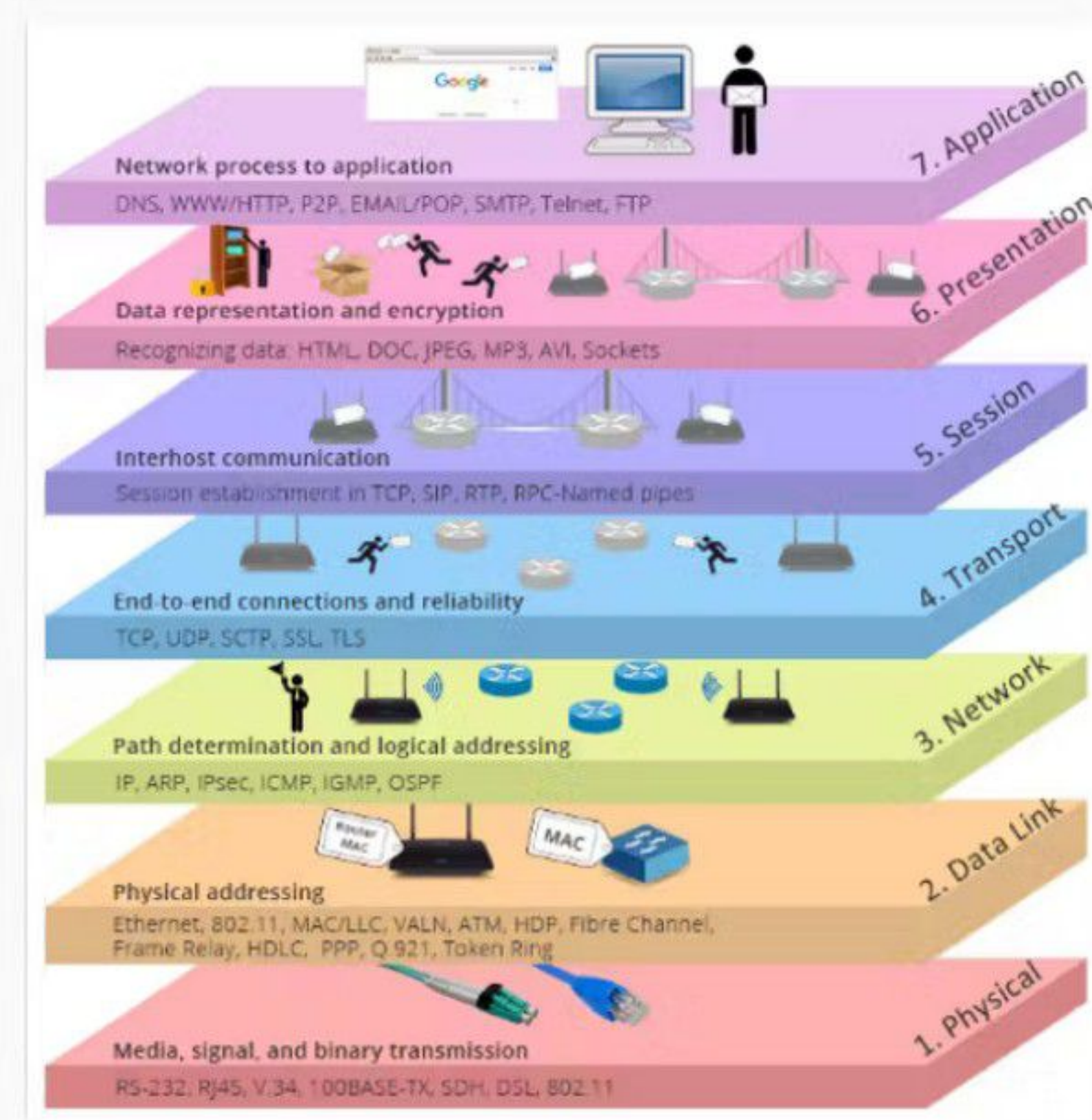


Figure 6: Hierarchy of the OSI model.

- Domain names are meaningful to humans, but not to machines.
- Domain names are translated into IP addresses by the Domain Name System (DNS), which are meaningful to machines.
 - DNS is a distributed database that maps domain names to IP addresses.
- Begins with the name of the host machine, followed by progressively larger collections of machines i.e., subdomains.

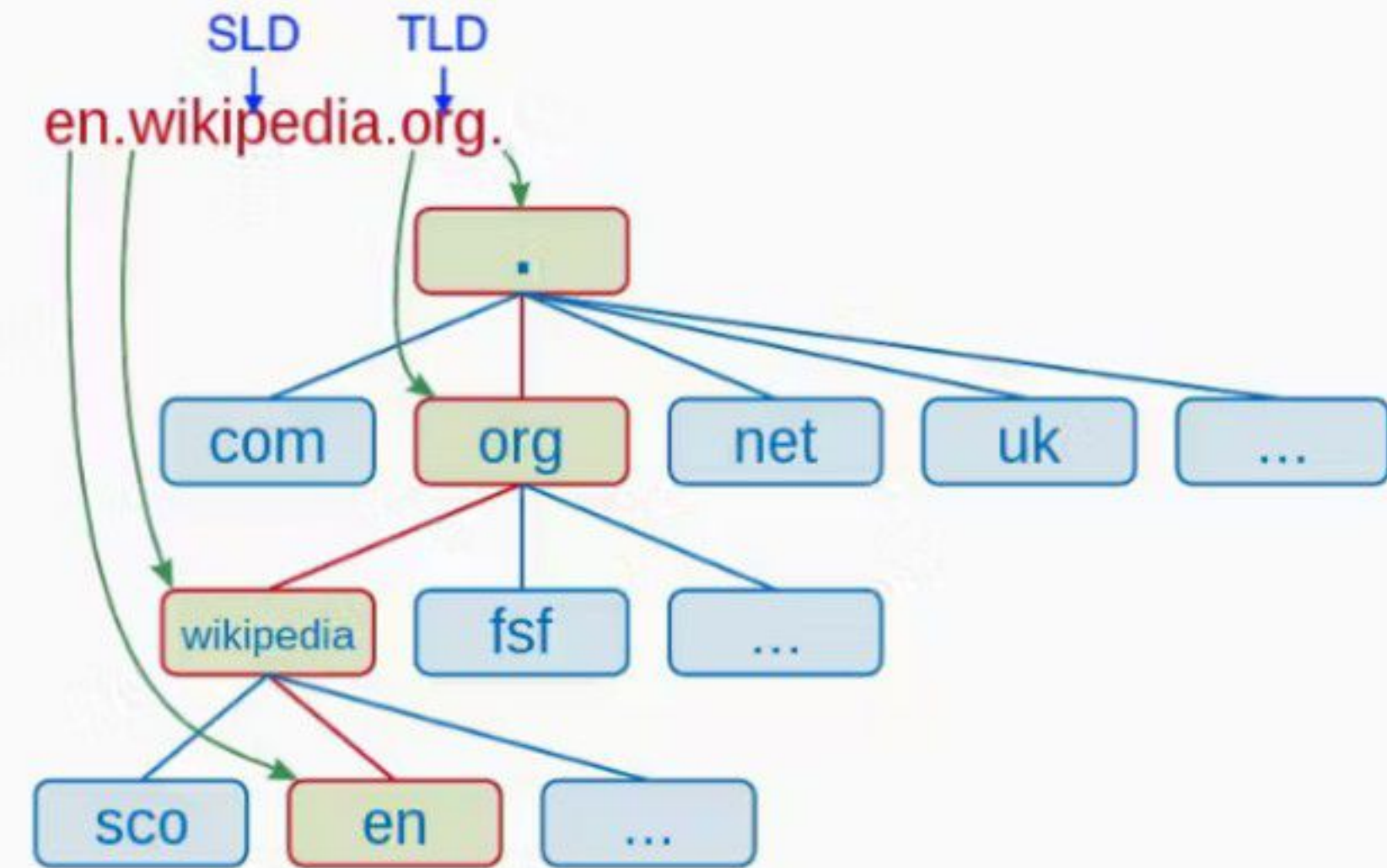


Figure 7: Components of a Domain Name.

- A URL is a web address that specifies the location of a resource on the internet. It points to the unique location of a file, page, or other web resource.
- The structure of a URL typically begins with the protocol (https://), then the domain name, path to the resource, and optional parameters.
 - The protocol is often the HTTP protocol, but can also be FTP, HTTPS, etc.

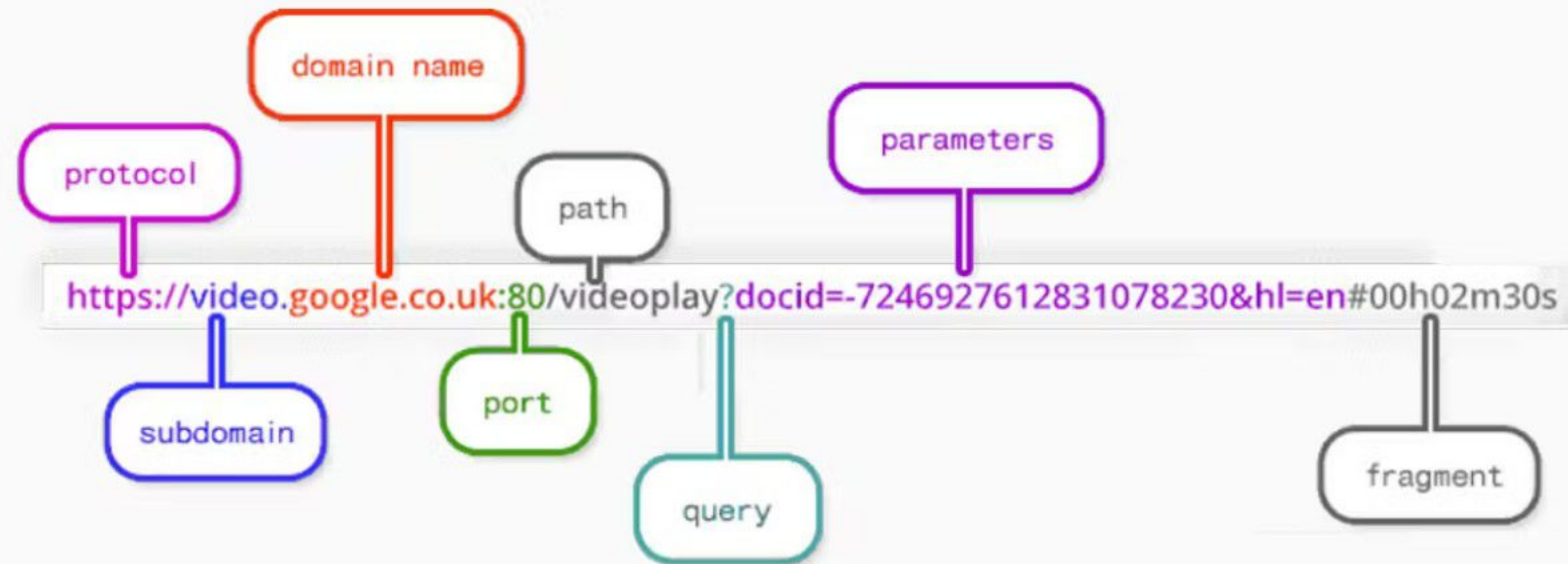


Figure 8: An example of a valid URL with annotated components.

- HTTP is the underlying protocol used by the World Wide Web to define how messages are formatted and transmitted between web browsers and servers. HTTP works on a request-response model (Shown in Figure 9).
- There are various HTTP request methods that specify the type of action to be performed on a resource.
 - Examples include: GET, POST, PUT, DELETE.
 - We'll look at GET and POST in more detail later in the module.
- HTTP messages consist of a header and a body. The header contains metadata like request method, URLs, status codes. The body contains the resource content.
- HTTP is stateless, meaning the server does not retain any state about previous requests.

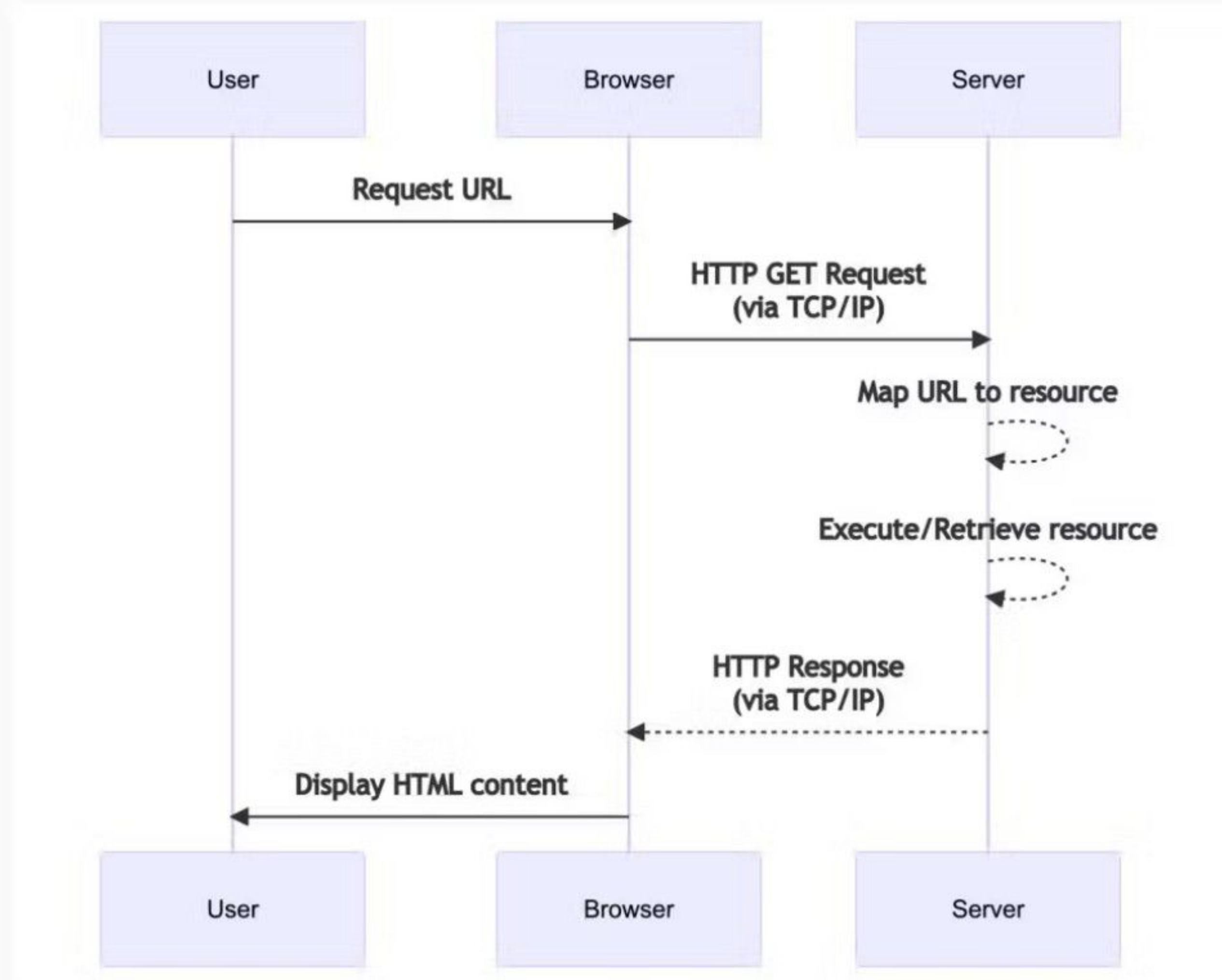


Figure 9: Example demonstrating the stages of a HTTP request/response cycle.

- Devices connected to the Internet rely on the TCP/IP protocol suite to communicate.
- IP addresses are four-part numbers (e.g. 192.168.1.1) that uniquely identify devices on a network.
- DNS (Domain Name System) translates domain names that are meaningful to humans into the numerical IP addresses.
- URLs (Uniform Resource Locators) specify the location of resources on the web (e.g. a web page, image, video file).
- HTTP (Hypertext Transfer Protocol) is the fundamental protocol used for communication over the web.
- The most common HTTP request methods are GET for retrieving data, and POST for submitting data.

Any Questions?

1 question
0 upvotes

Questions?
