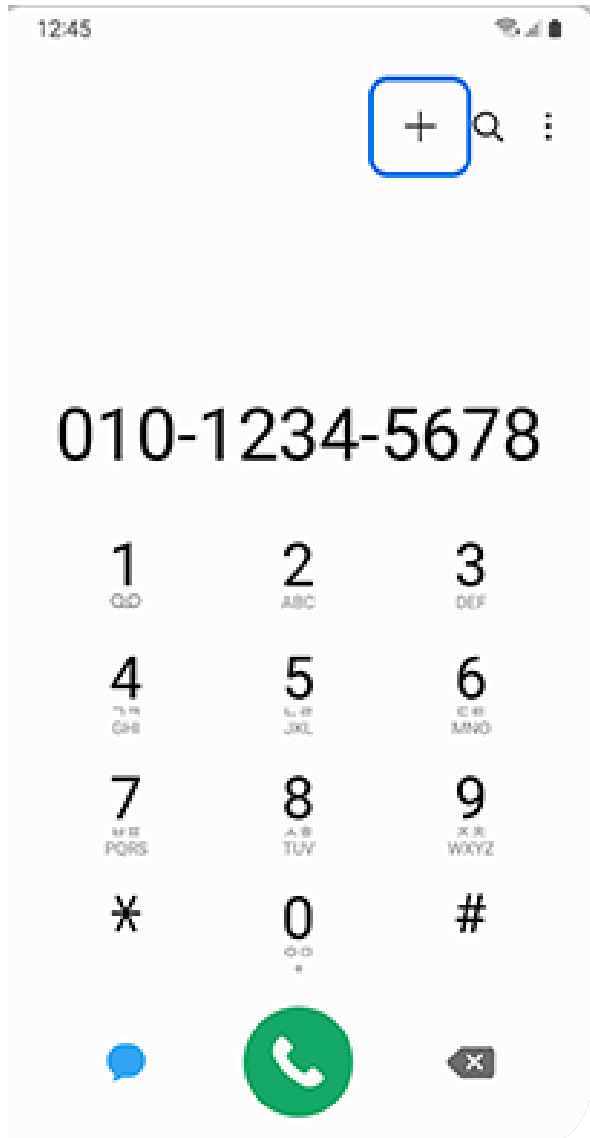


# DOMAIN NAME SYSTEM

By CommentLee

# WHY DOMAIN NAME SYSTEM



# 왜 DNS가 필요한가?

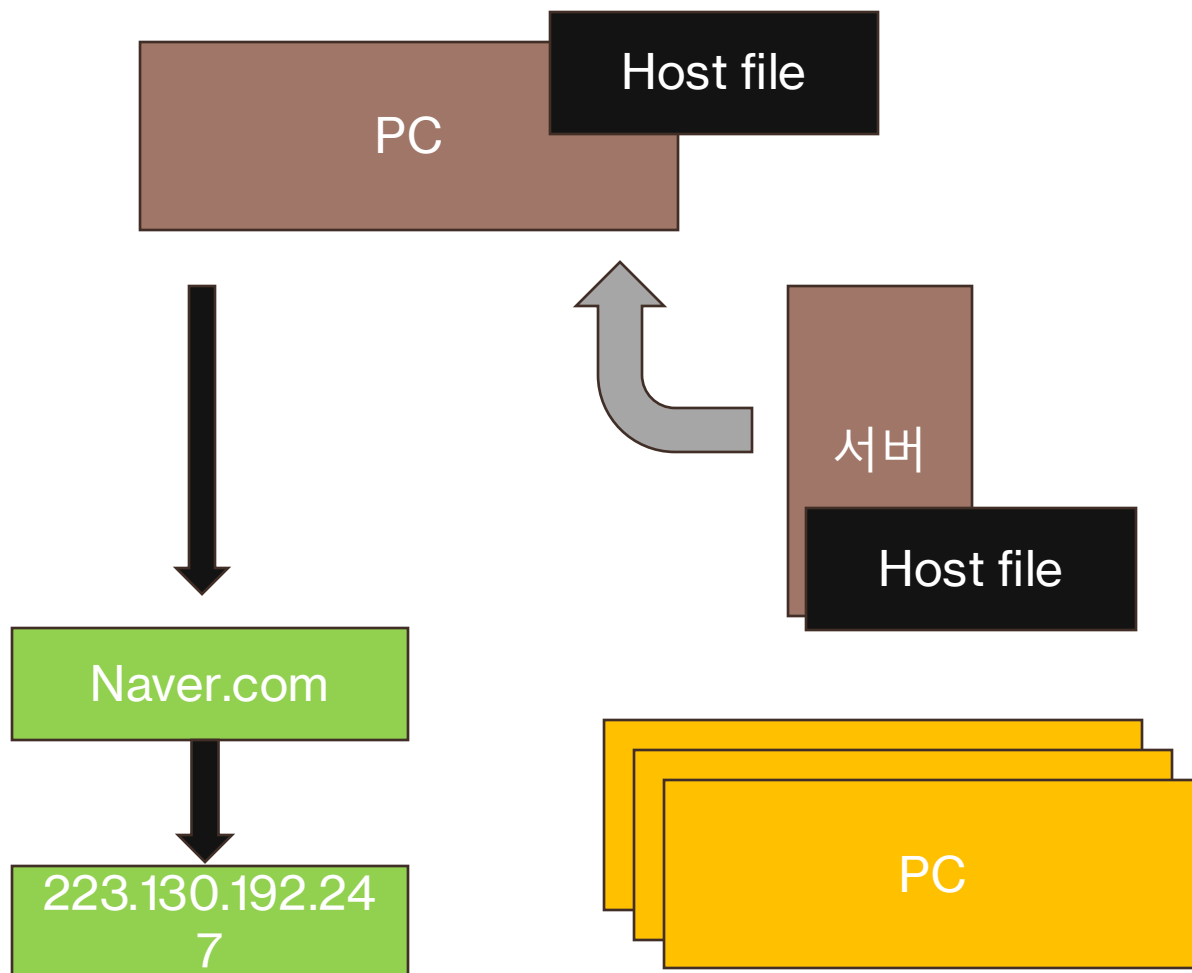
**OSI 7계층 복습: 저 멀리 누군가와 통신하려면 IP주소가 필요함**

**문제점: 142.250.76.142 , 223.130.192.247 ... 혹시 전화번호도 다 기억하고 전화거시는분?**

**대부분의 사람은 숫자보다는 이름을 잘 기억함**

**142.250.76.142 -> google.com 으로 변환하는 무언가가 필요하다**

# 왜 DNS가 필요한가?



## 방법1: host 파일

- 전화번호부처럼 저장하자
- 파일 하나를 만들고

Naver는 123... Google은 456... 저장

- 문제점: 전부저장가능? 업데이트?

## 방법2: 그 파일을 중앙 서버가 들고있으면?

- 내 host에 없다면 중앙 서버에 물어보면 될 것 같은데?
- 문제점: scalability, 안정성

DOMAIN NAME  
SPACE

# NAME SPACE

## Flat Name Space

각 주소에 유일한 이름 (unique name)을 할당 -> 중복 많다

## Hierarchical Name Space

각 이름을 여러 파트로 나누어짐

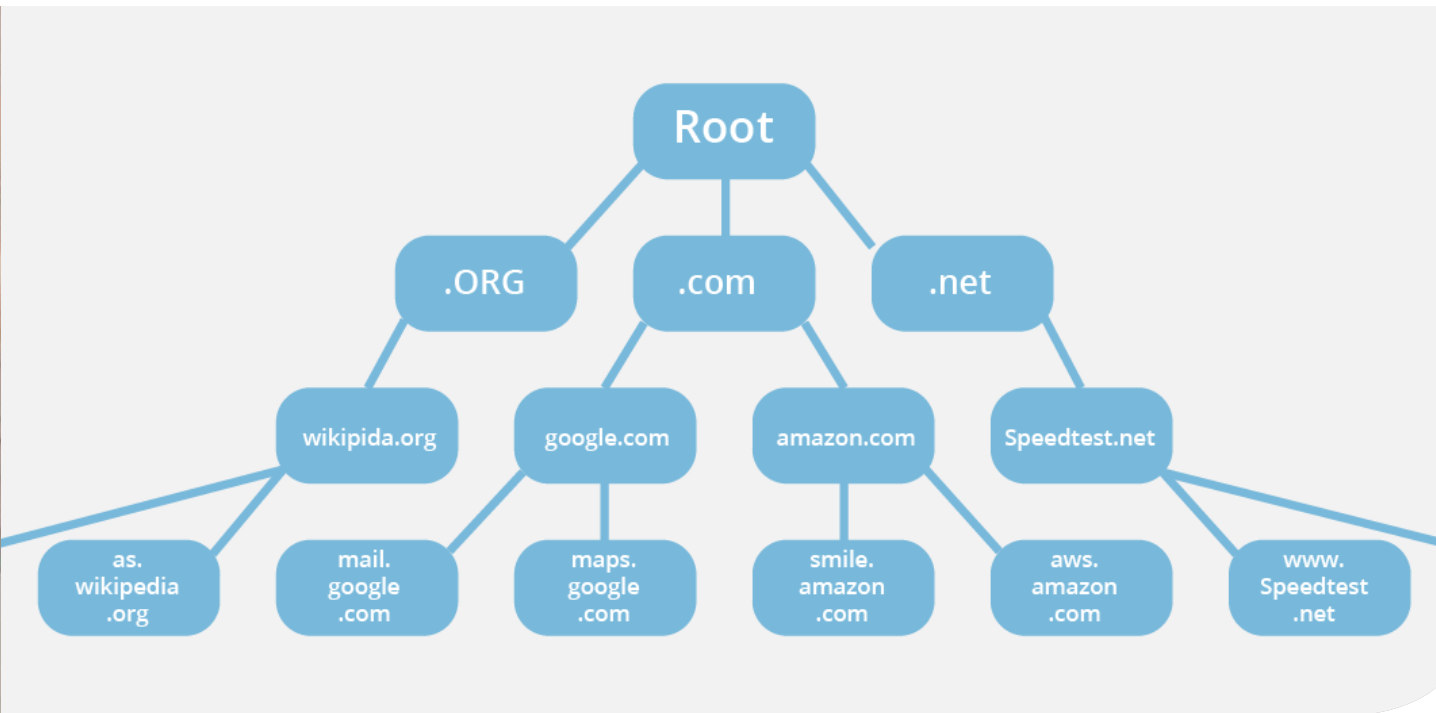
**mail**.naver.com

**mail**.google.com

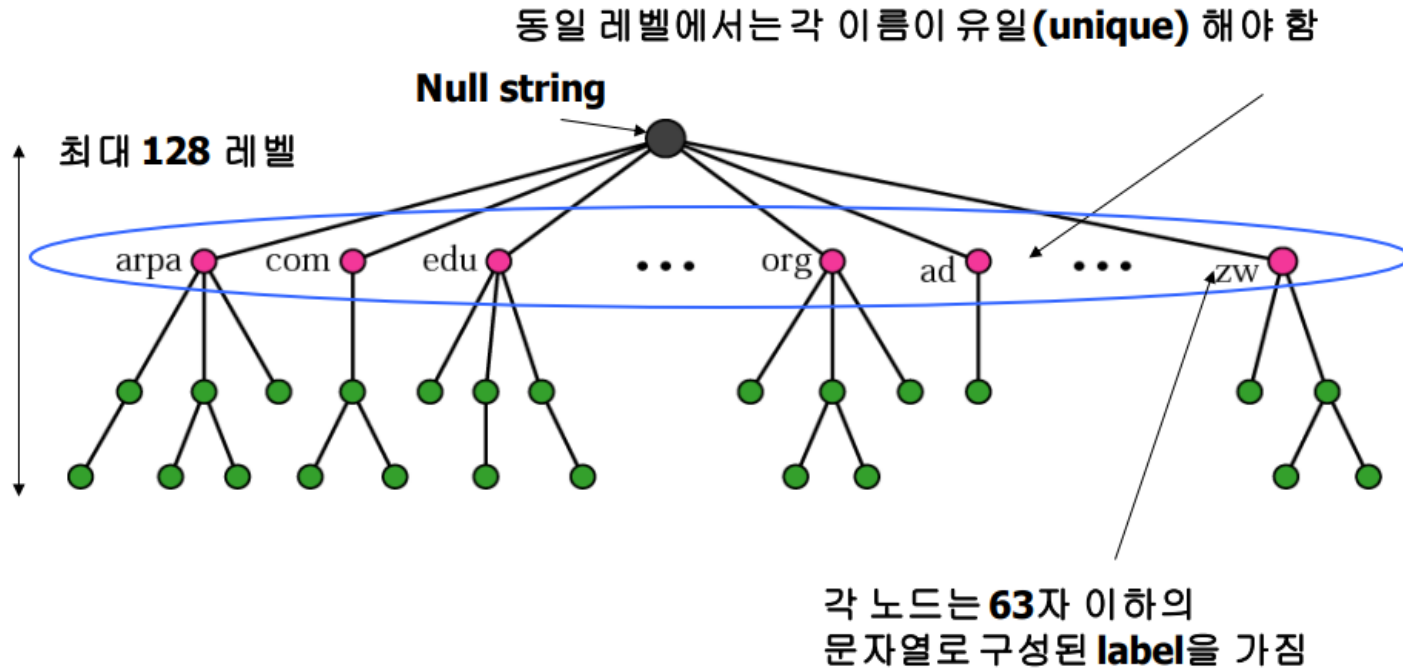
같은 이름이어도 ok.

회사=com, 단체=org, 한국=kr ...

관리하는 주체 분산가능



# DOMAIN NAME SPACE



## Hierarchical Name Space

를 구현한게 DOMAIN NAME SPACE

읽을때는 아래서부터 루트방향으로 읽기  
우리가 늘 쓰고 있어서 당연하게  
읽을수있다.

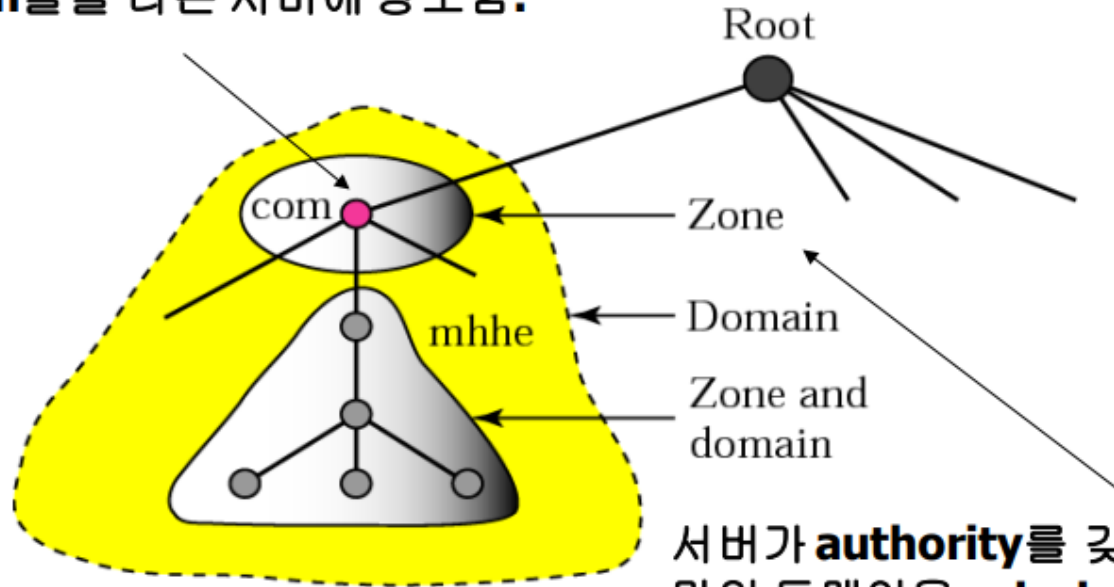
www.naver.com

계층으로 잘 분리하였다

-> 이제 처음 말했던 중앙 서버를 이대로  
분산시키자

# 네임 서버 분산

이 서버는 **com** 도메인에 **authority**를 가지며,  
**Sub-domain**들을 다른 서버에 양도함.



각 서버는 본인의 도메인에 대해 권한을  
갖는다

- 루트서버 13개(복사 수백개)
- 최상위 도메인 (TLD, Top-Level-Domain)
- 서브 도메인

내가 응답이 불가능하다면 권한이 있는 하위  
도메인으로 가라한다.

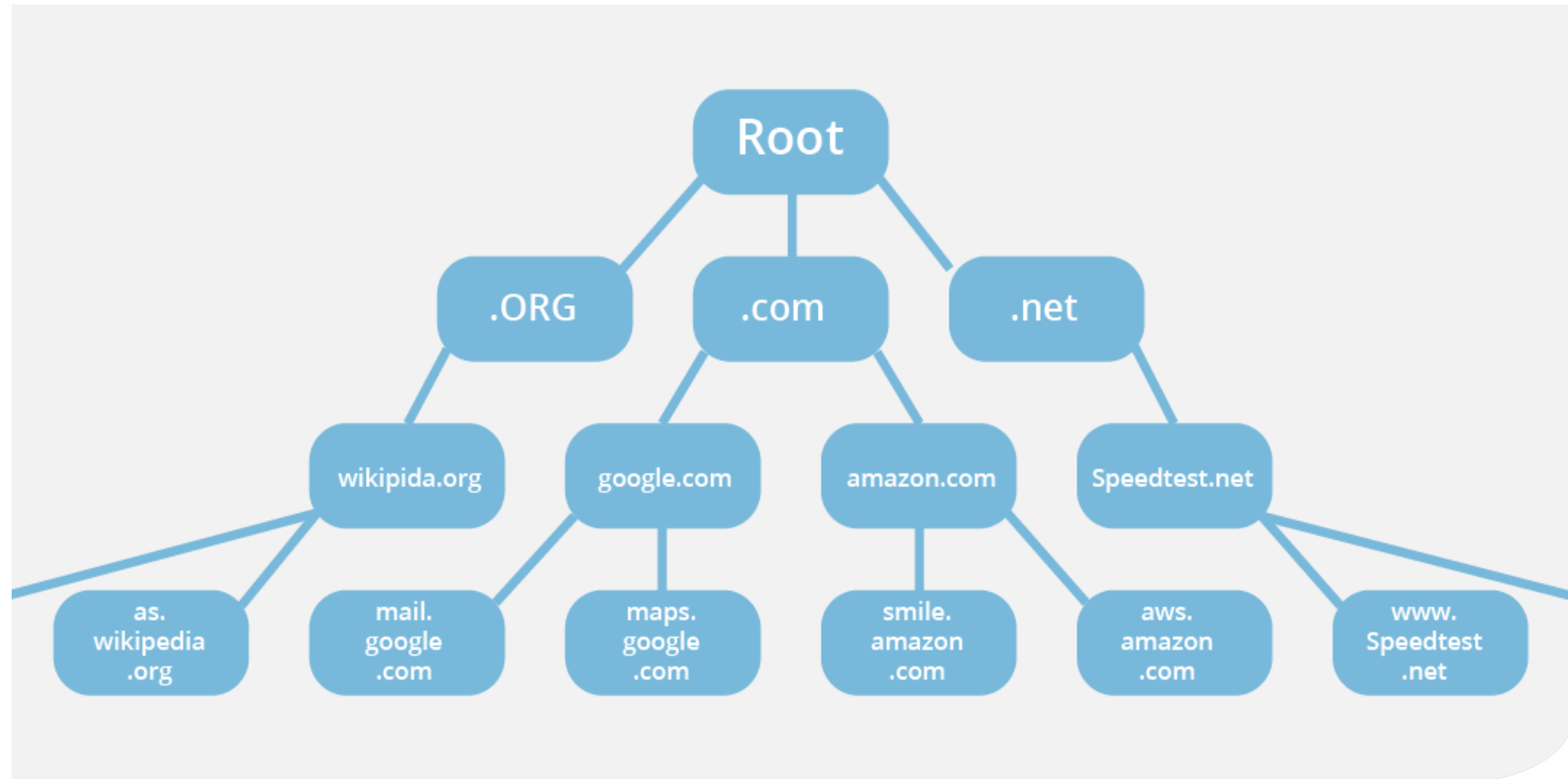
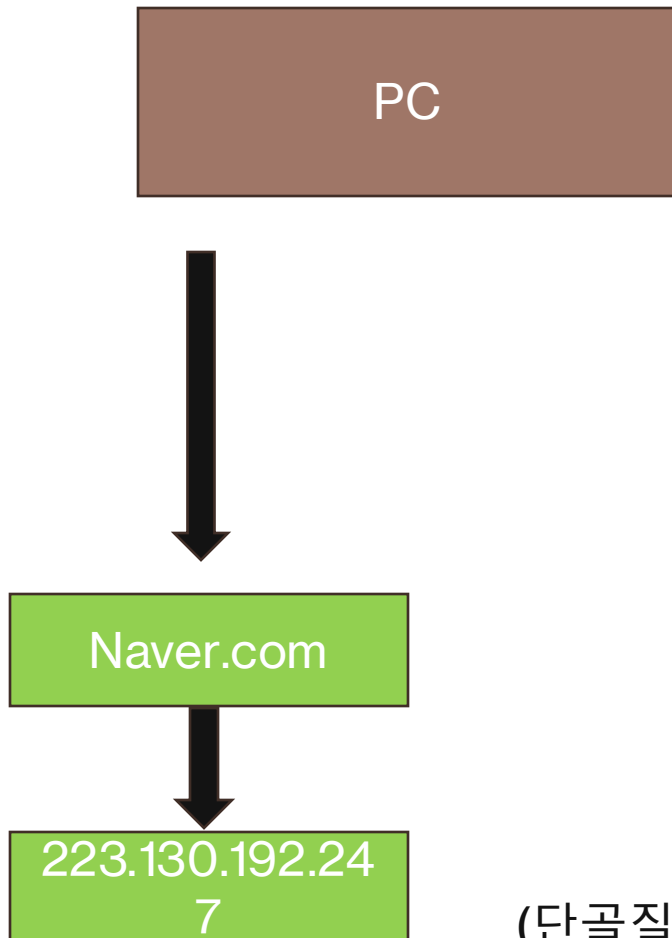
서버가 **authority**를 갖는 영역.  
만일 도메인을 **sub-domain**으로  
분할하지 않는 경우 **domain = zone**.  
그렇지 않은 경우 둘은 서로 다름.  
**Zone file**을 저장함.



# HOW DNS WORKS

# DNS RESOLUTION

가장 가까운 DNS SERVER를 ACCESS해서 MAPPING REQUEST를 수행함  
- 서버는 그 요구를 만족시킬 수 있으면 그 정보를 전달하고,  
그렇지 않으면 다른 서버에게 그 정보를 요청한 후 정보를 전달



(단골 질문:WWW.NAVER.COM 에 접속하면 일어나는 일을 설명하세요)

# Recursive vs Iterative

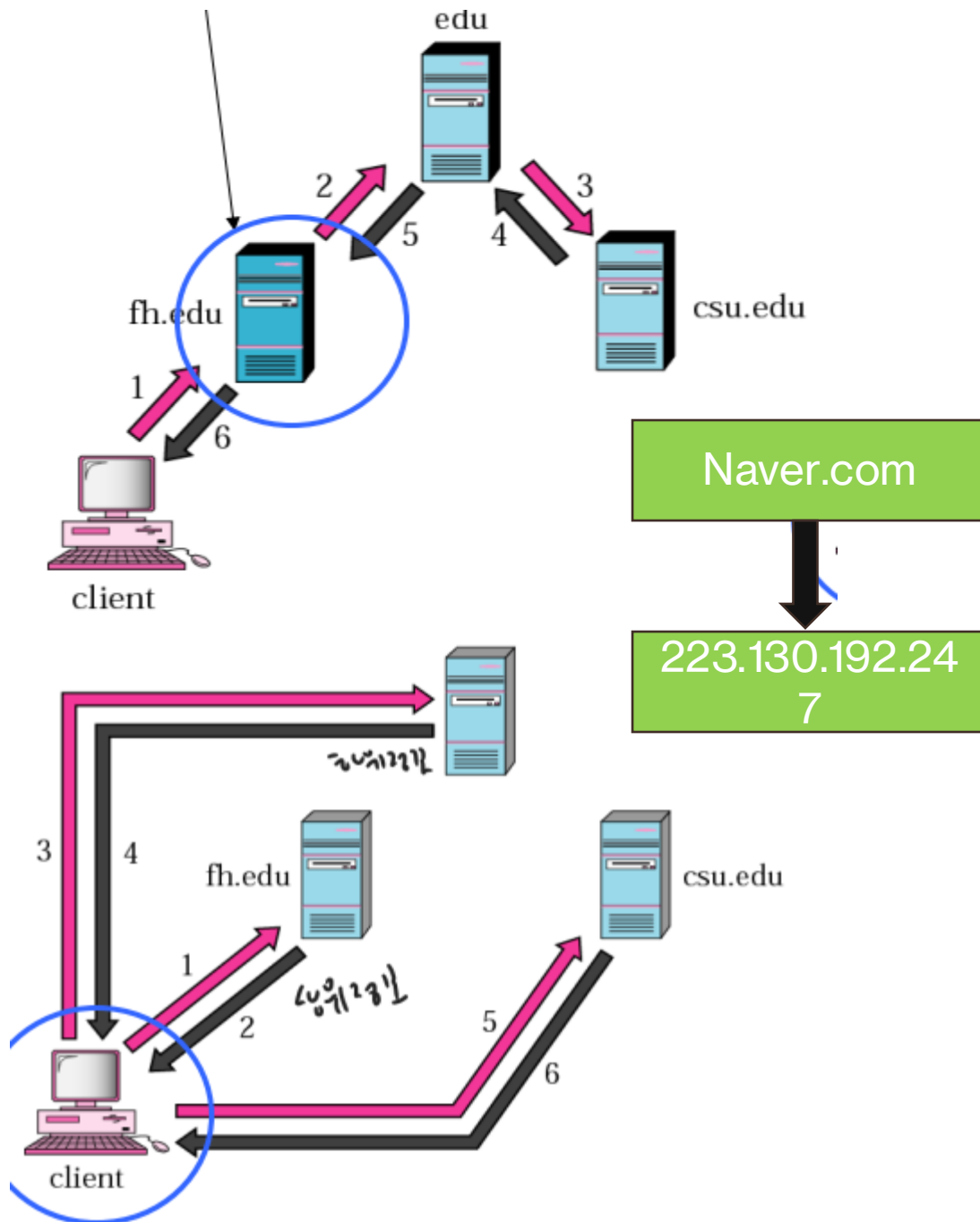
## Recursive DNS Resolution

재귀함수처럼 우리는 한번만 호출하고 기다림

서버가 다른 서버를 호출,, 또 그서버가 다른 서버를 호출,,

## Iterative DNS Resolution

클라이언트가 직접 요청을 여러 번 하러다닌다

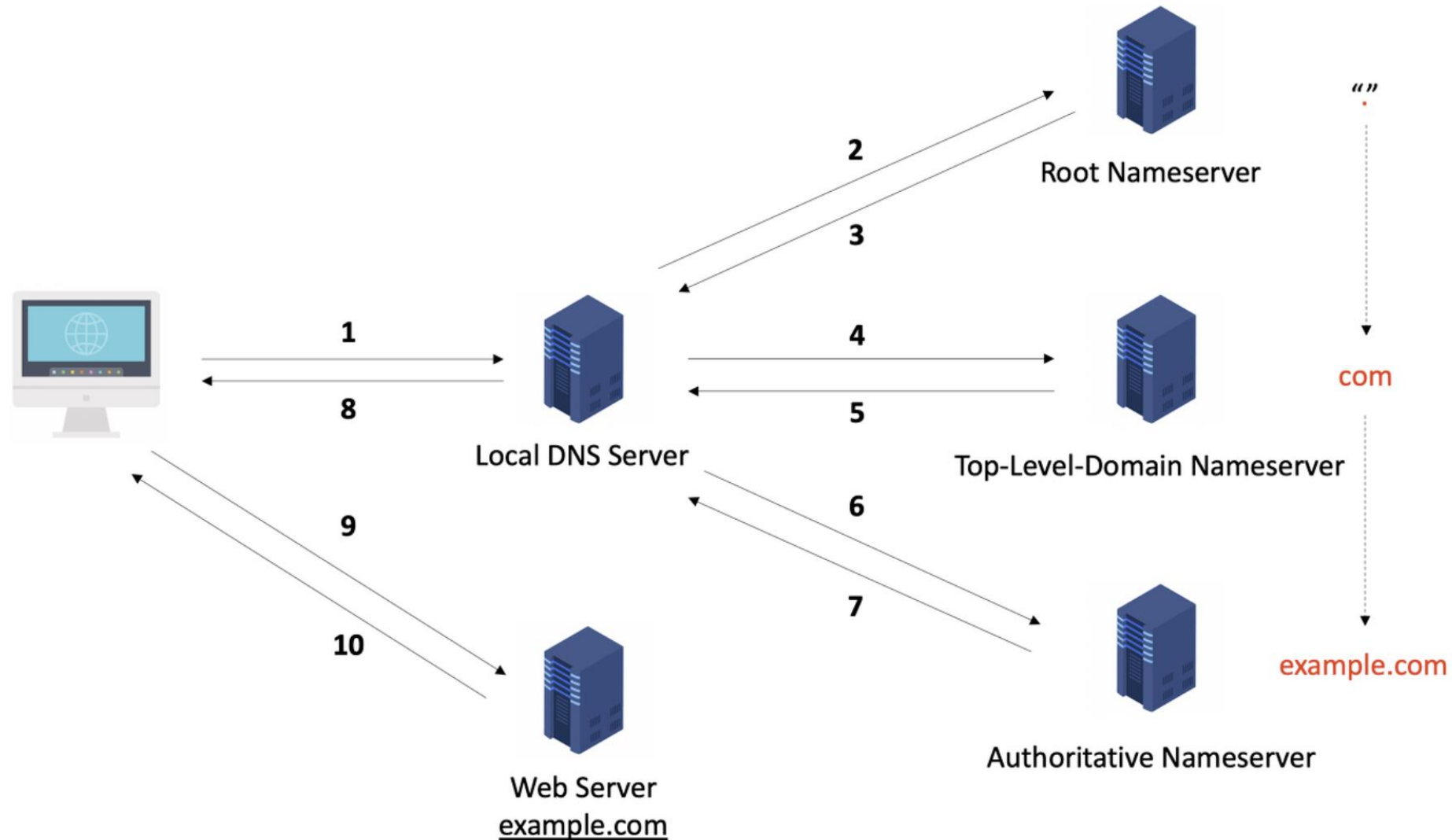


# 최종 동작

## DNS Caching

공유기, 로컬 DNS..  
여러곳에서 일어날 수 있음

TTL(Time-To-Live)로 캐시  
만료확인 가능



질문?