

# Lab 7: Content Delivery Network

by Pavinberg

# Review

- 五层协议栈

应用层

传输层

网络层

链路层

物理层

# Review

- 五层协议栈

应用层

传输层

网络层

链路层

物理层

以太网

# Review

- 五层协议栈

应用层

传输层

网络层

链路层

物理层

路由

以太网

# Review

- 五层协议栈



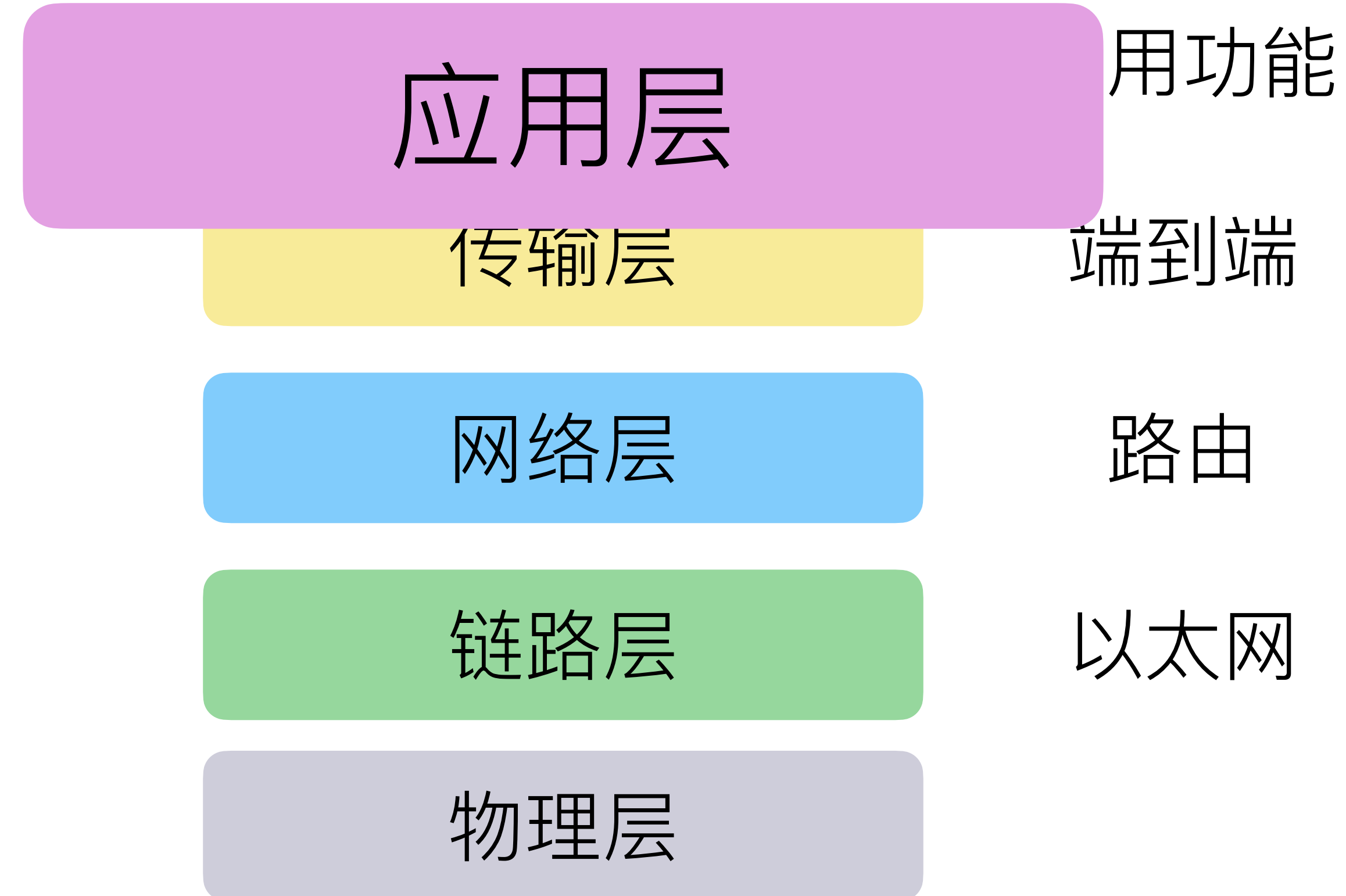
# Review

- 五层协议栈



# Review

- 五层协议栈



# Lab 7: Content Delivery Network

by Pavinberg





“为什么看视频这么卡？”

——一位外国友人

# Directly visit

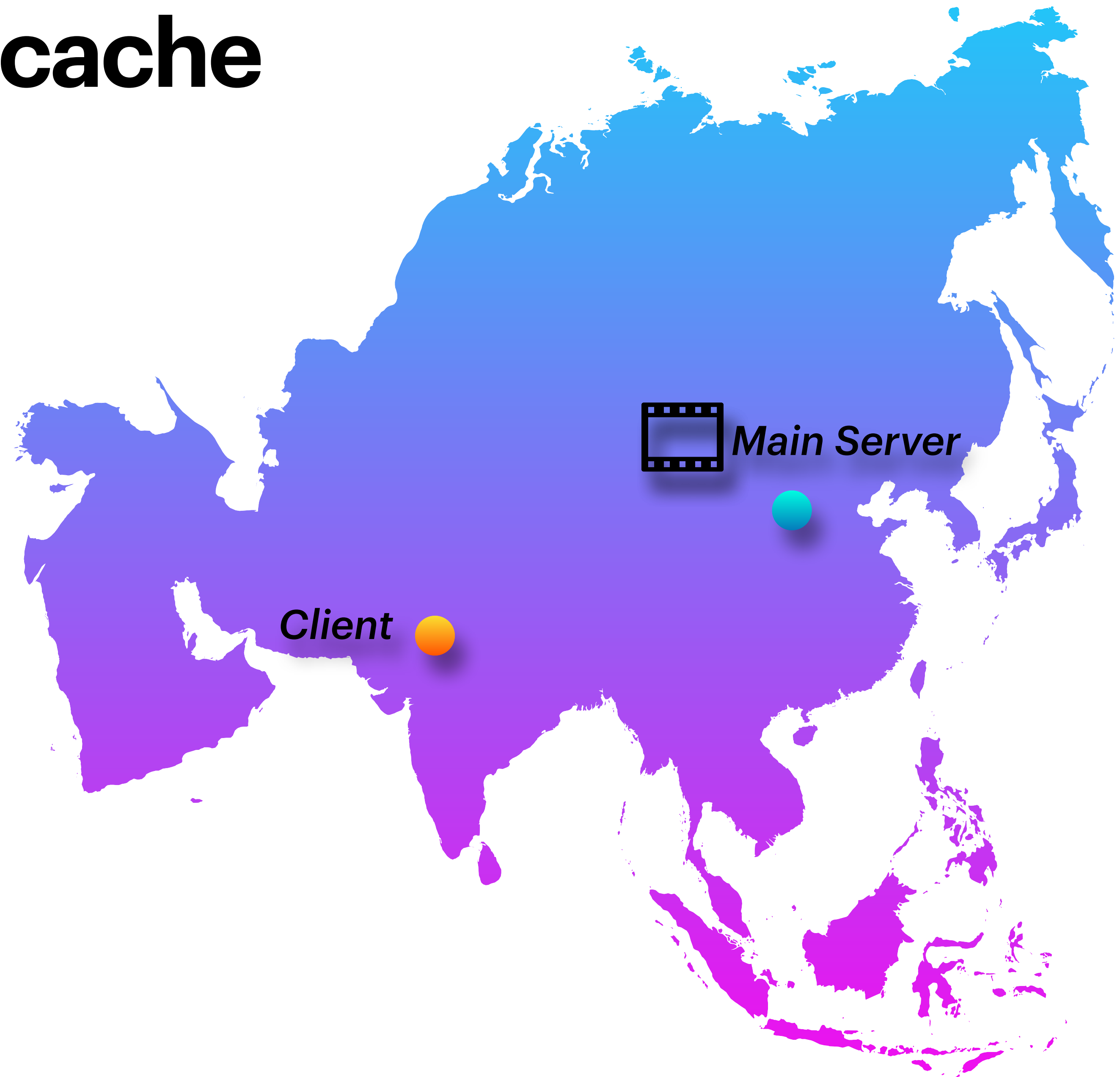


# Directly visit

👋: 有亿点点远

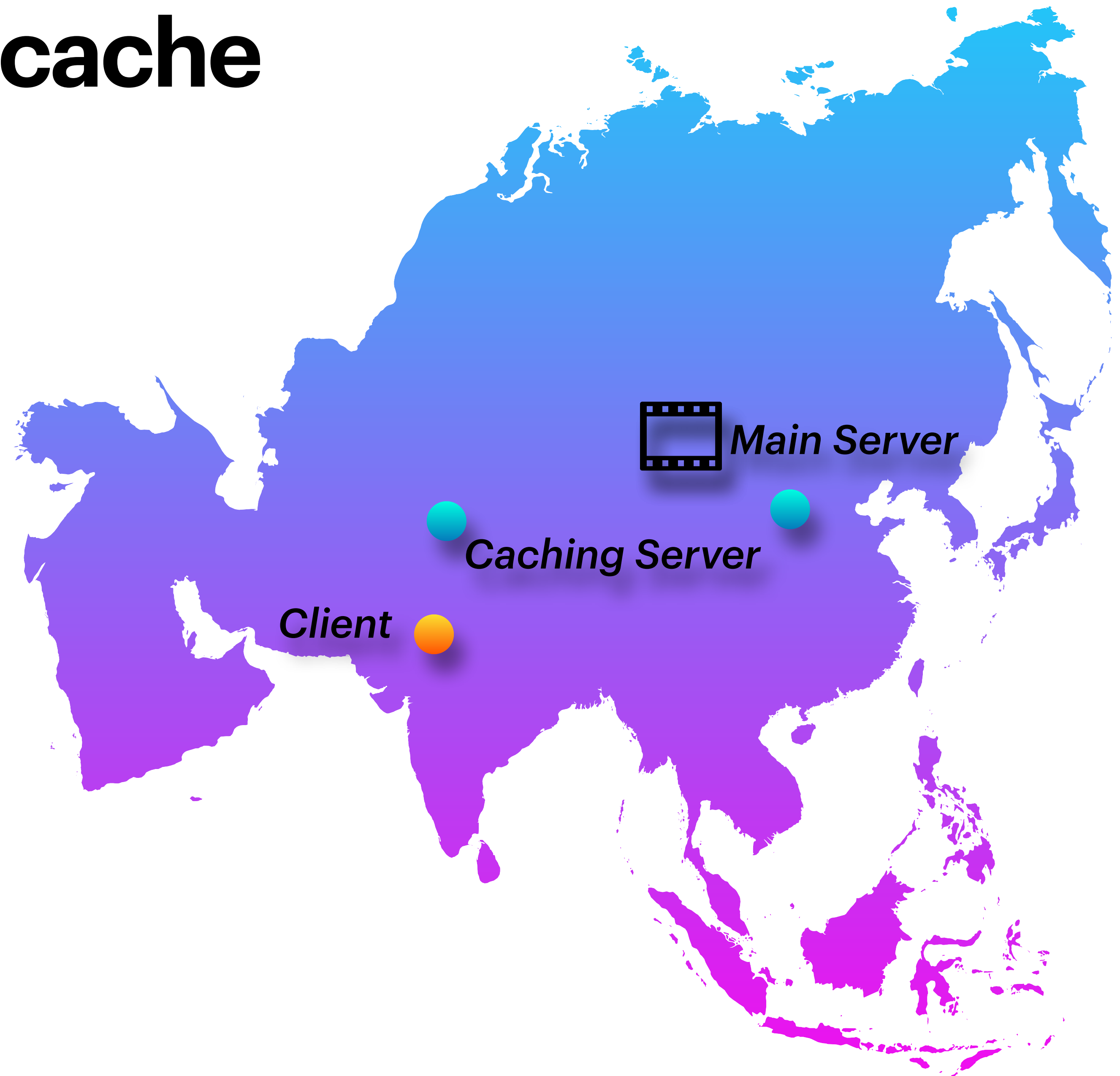


# Use cache



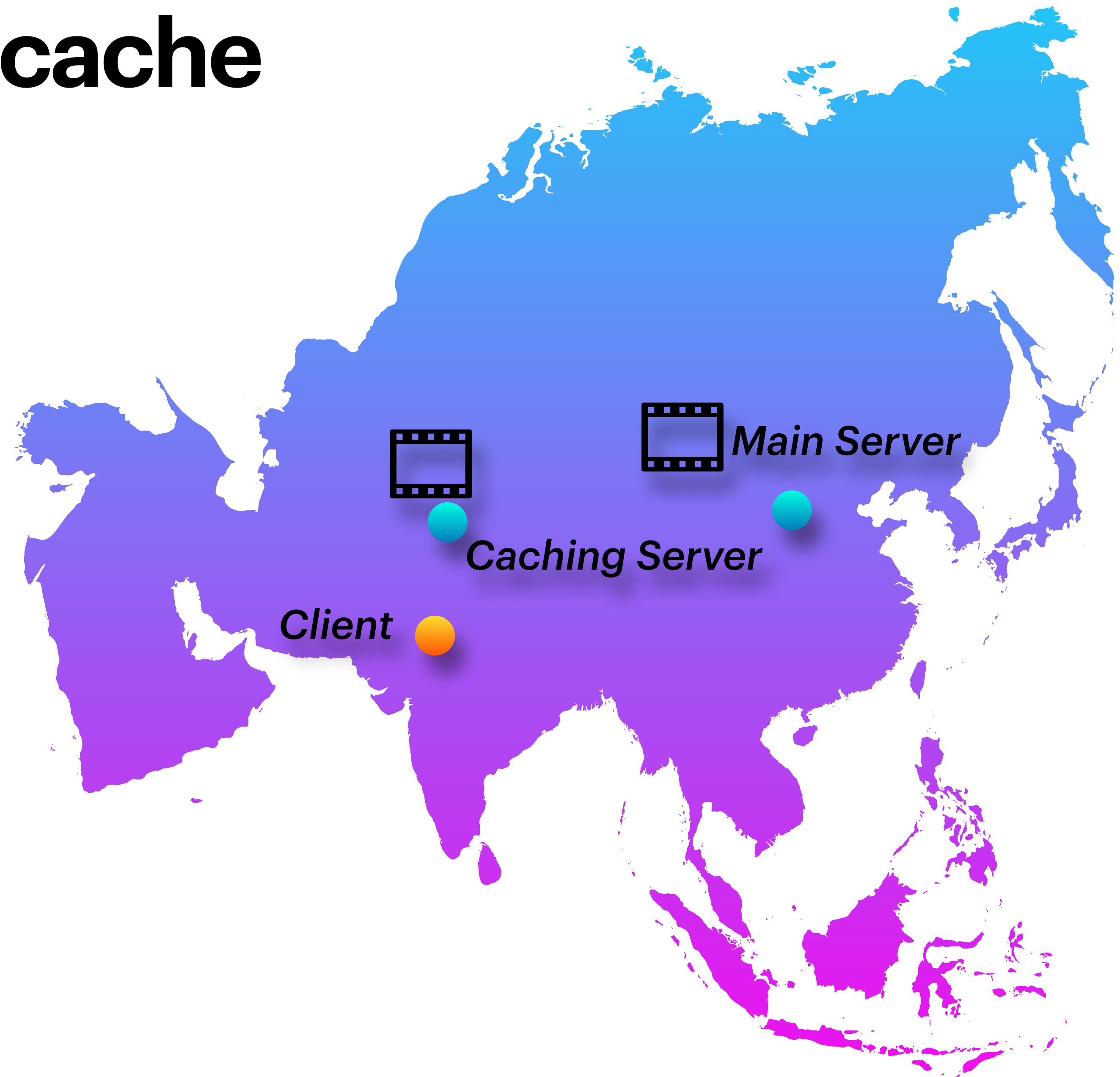
# Use cache

- 设置一个缓存服务器 💰



# Use cache

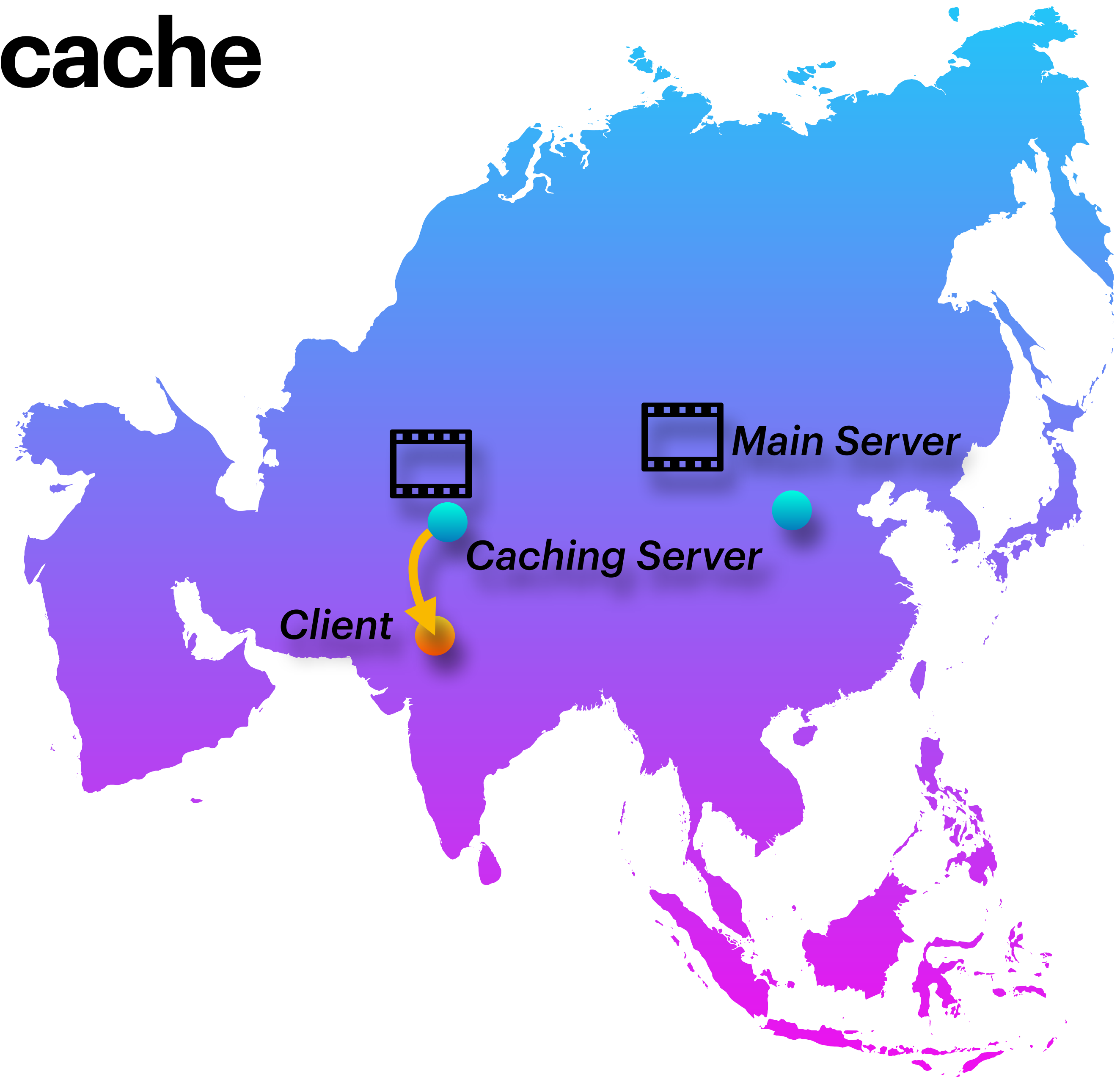
- 设置一个缓存服务器 💰
- 将内容复制一份



# Use cache

- 设置一个缓存服务器 💰
- 将内容复制一份

🙄: 整挺好





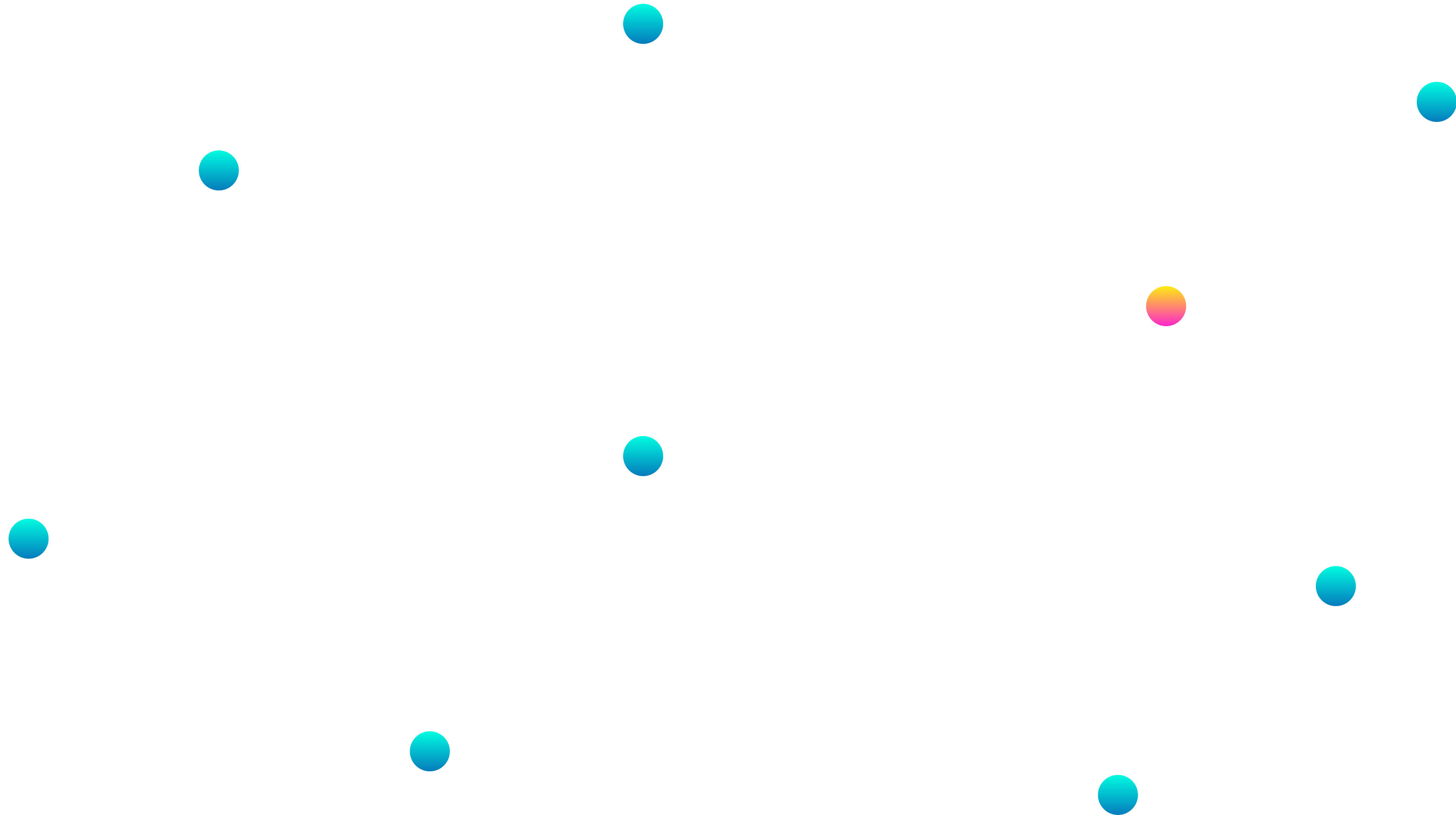
# Content Delivery Network



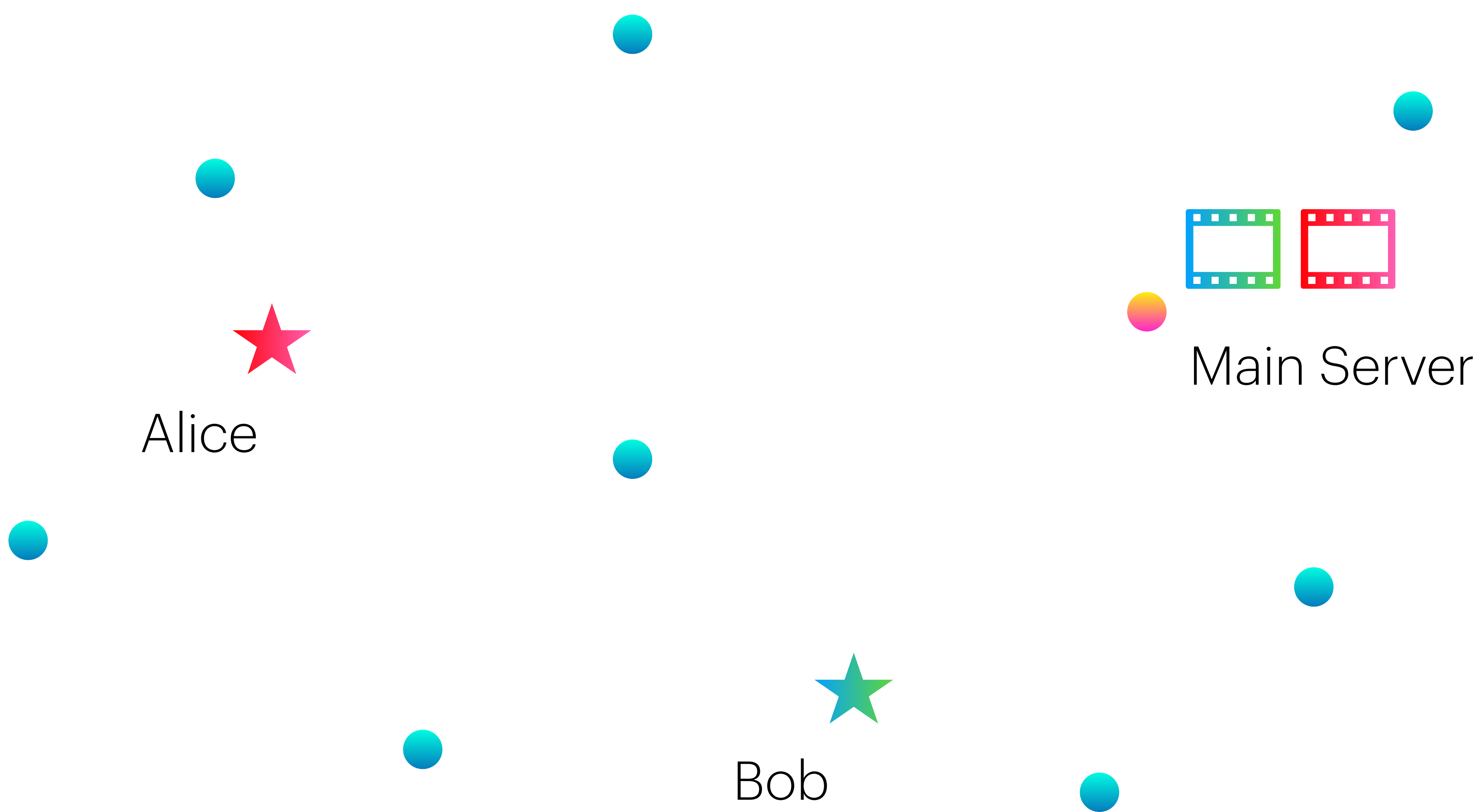
# Content Delivery Network



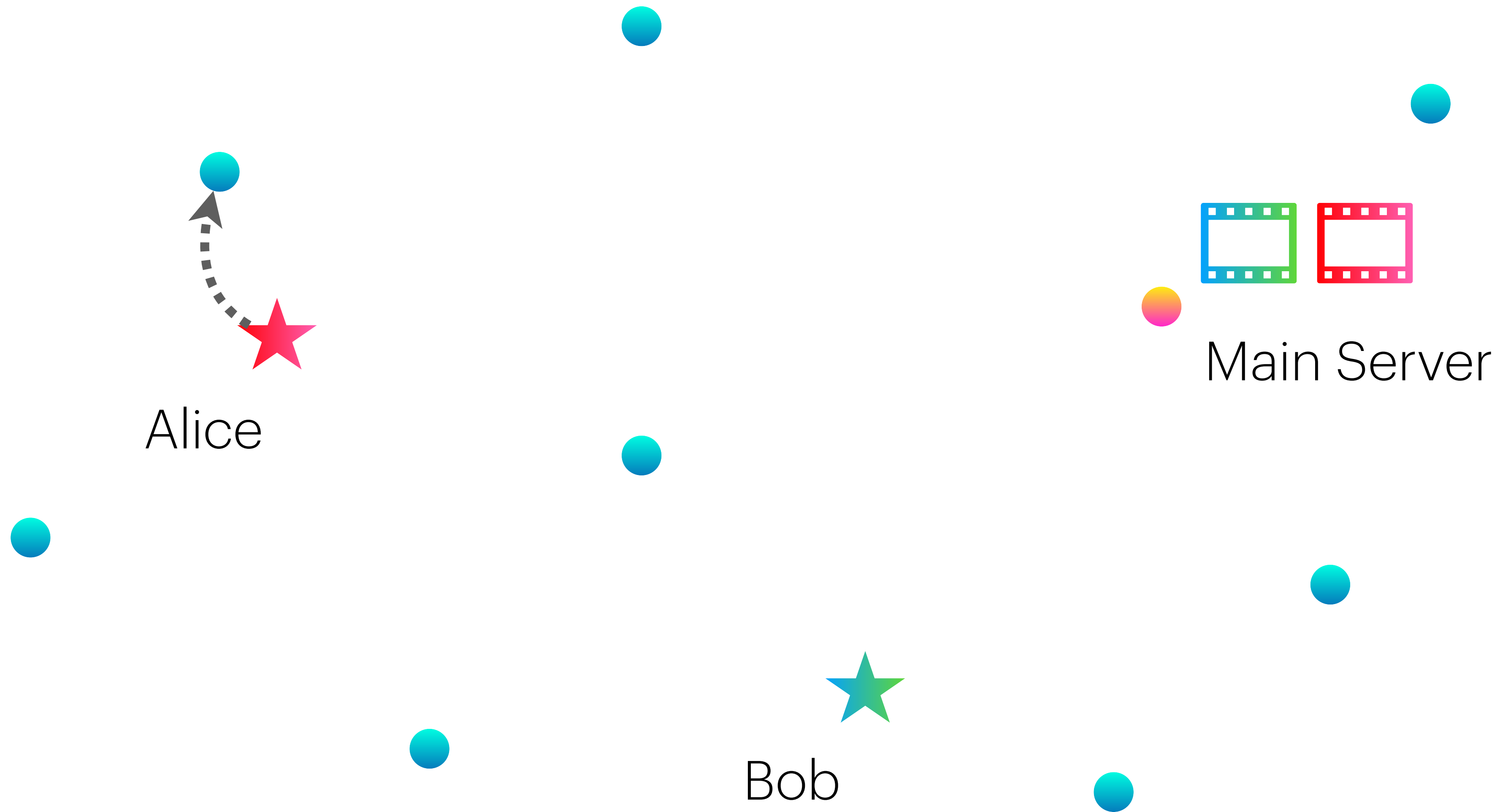
# Content Delivery Network



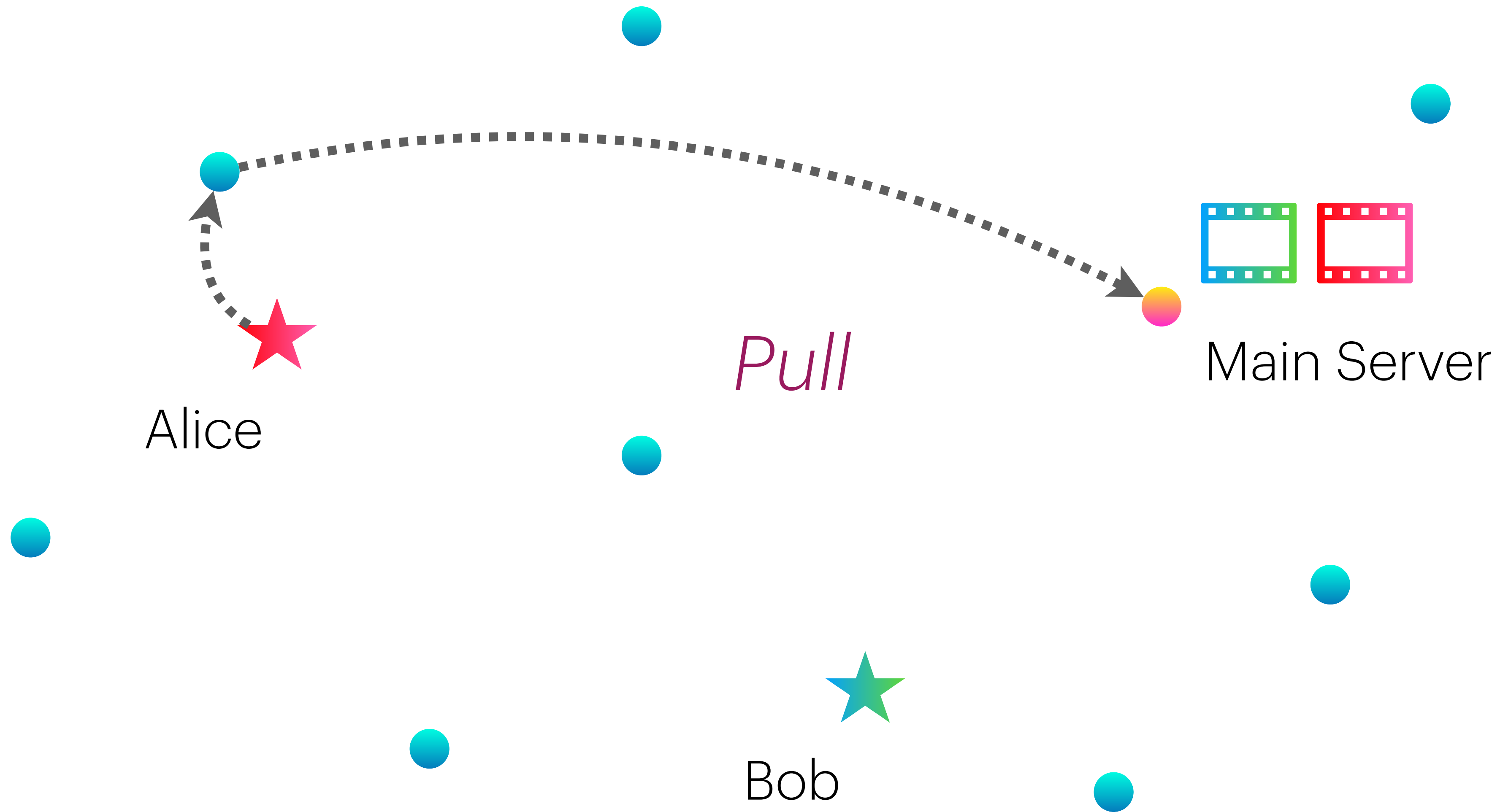
# Content Delivery Network



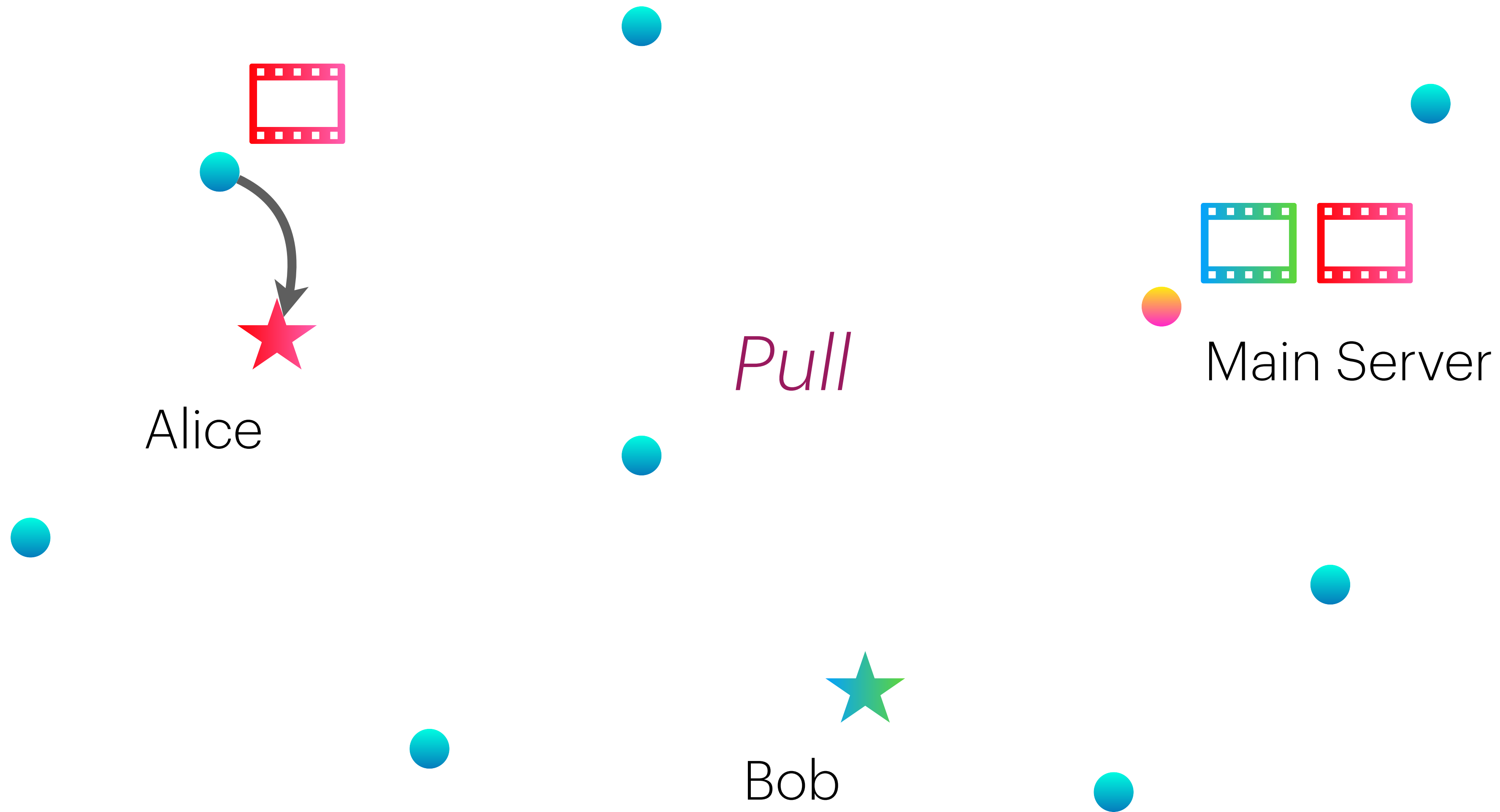
# Content Delivery Network



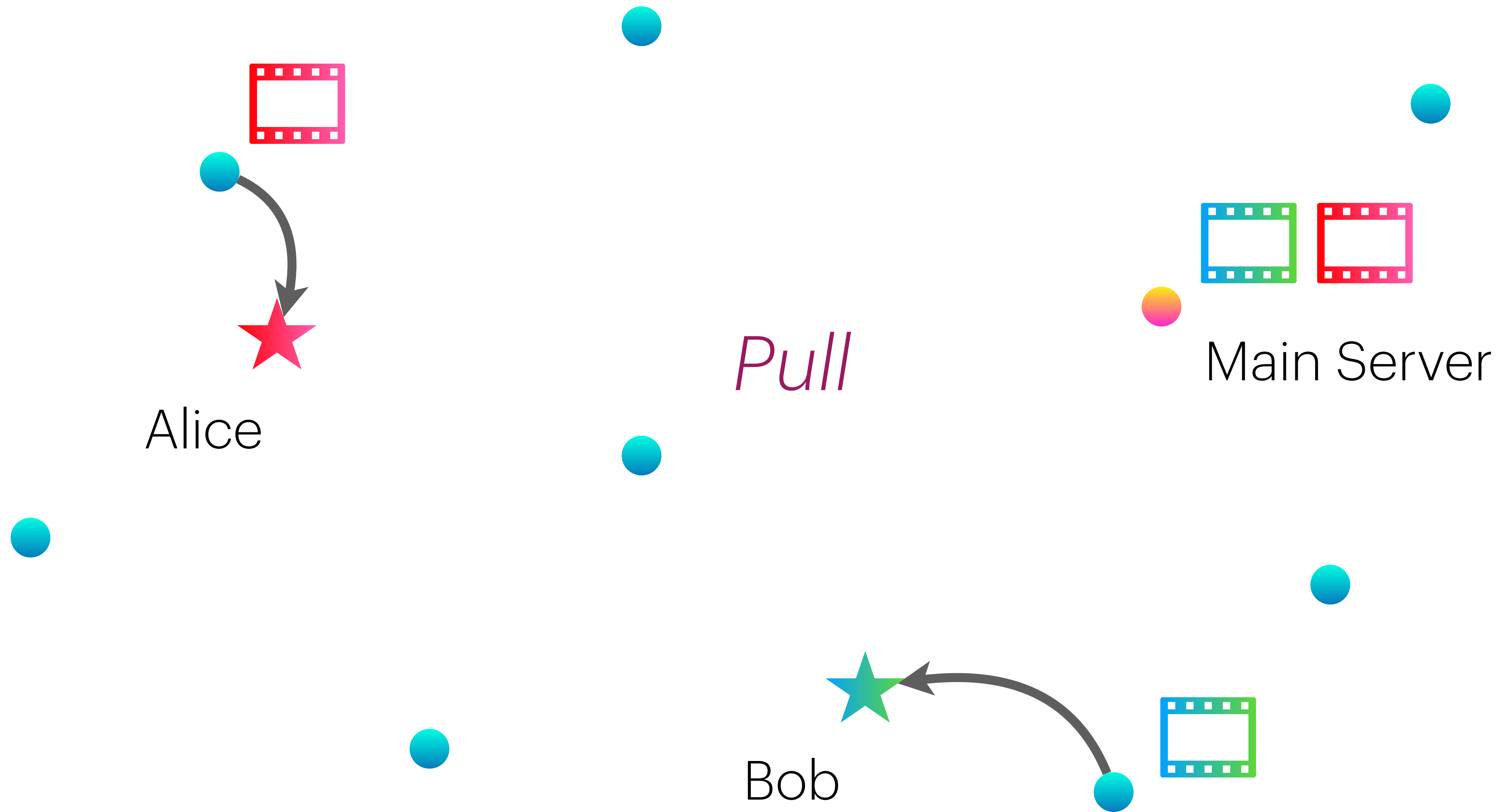
# Content Delivery Network



# Content Delivery Network



# Content Delivery Network





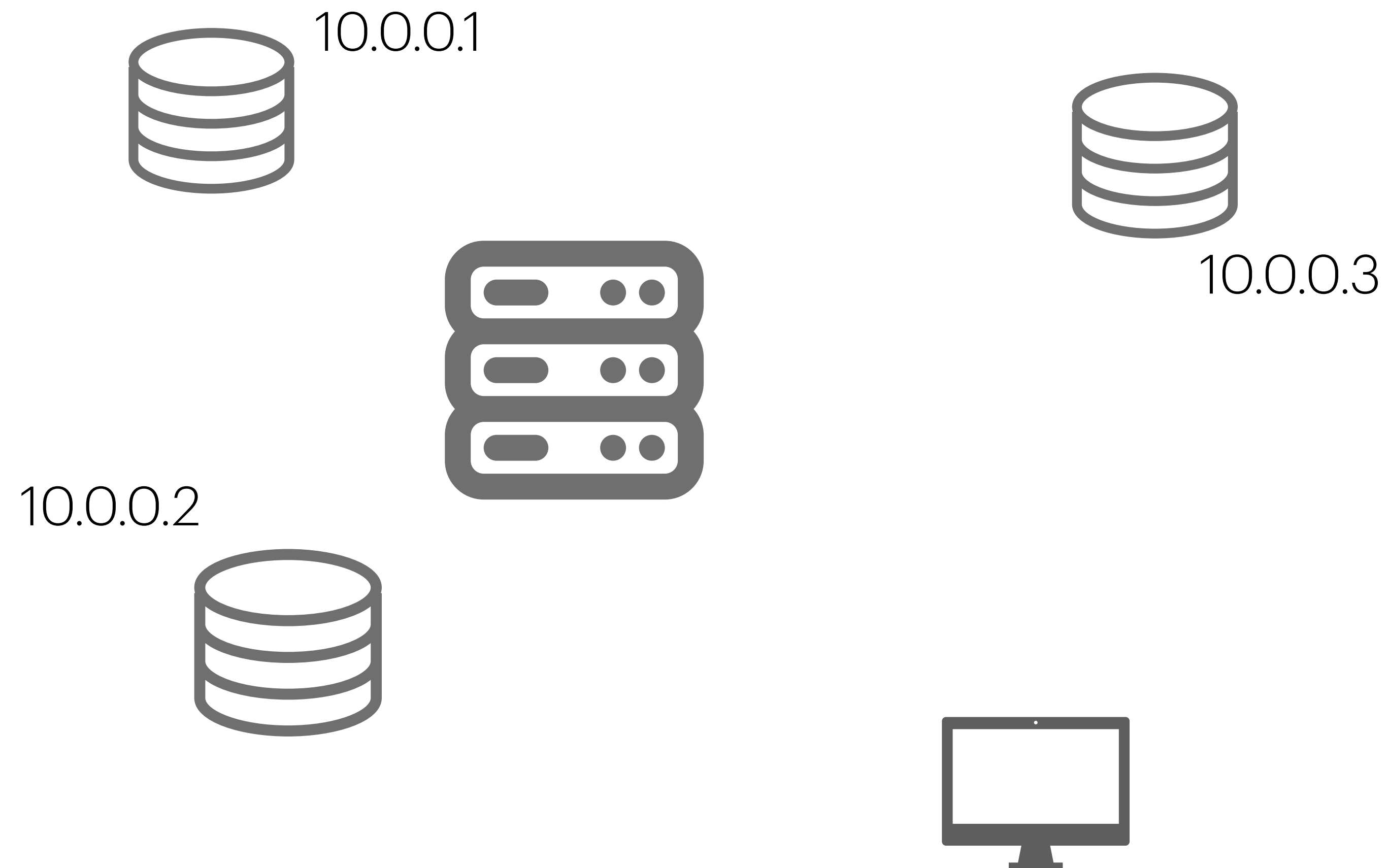


“我如何找到最合适的缓存服务器？”

——客户端

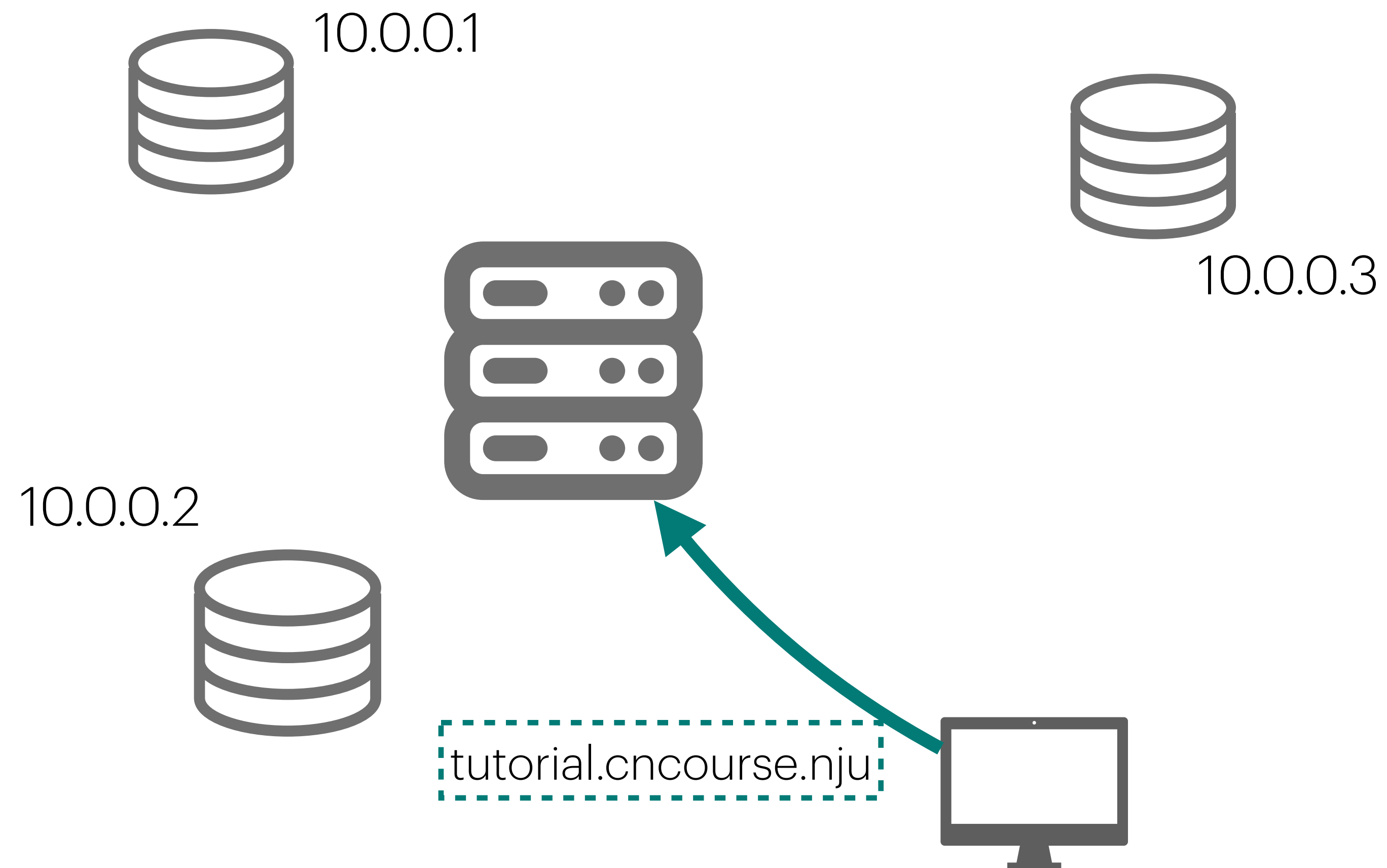
# DNS Server

is all you need



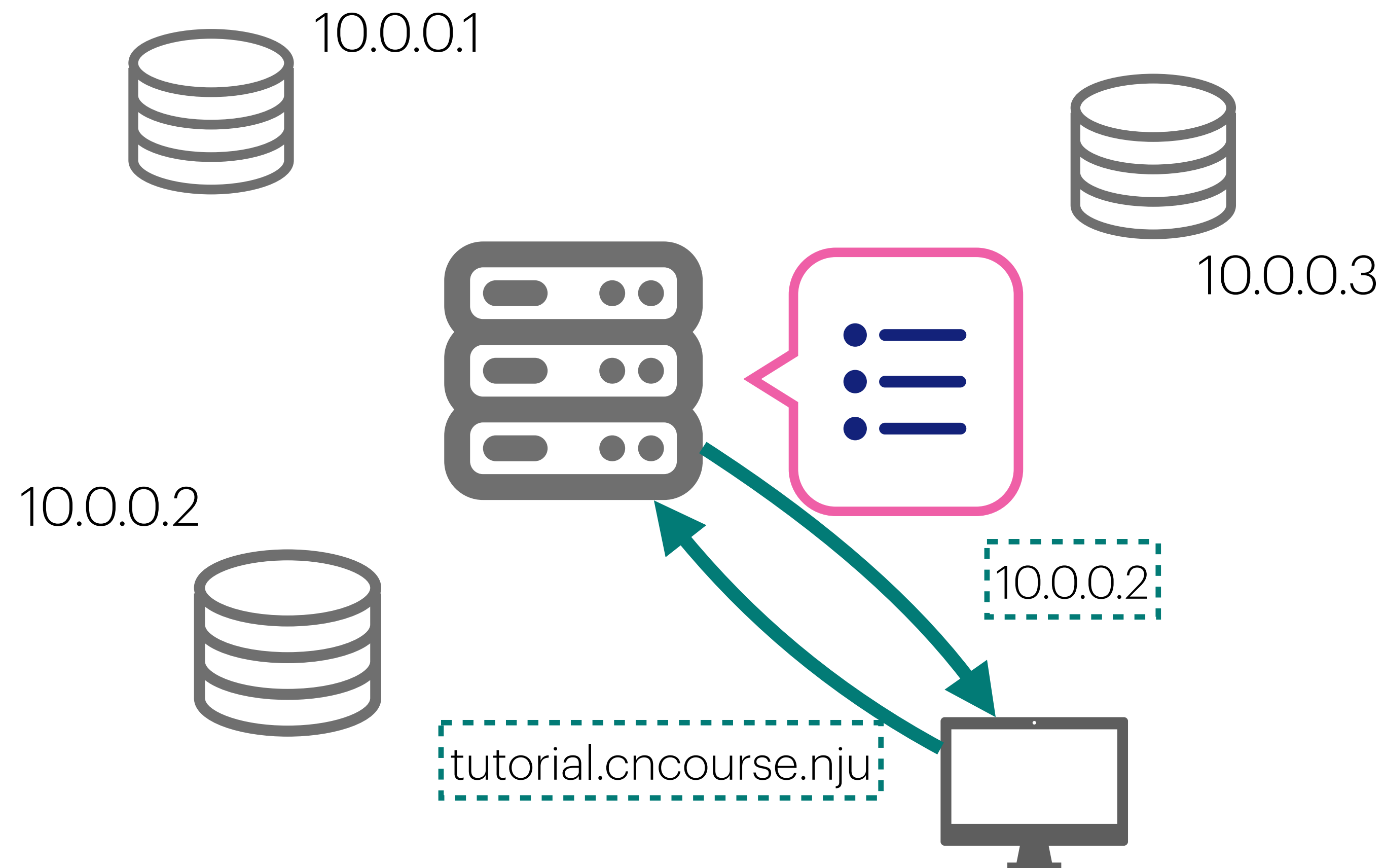
# DNS Server

is all you need



# DNS Server

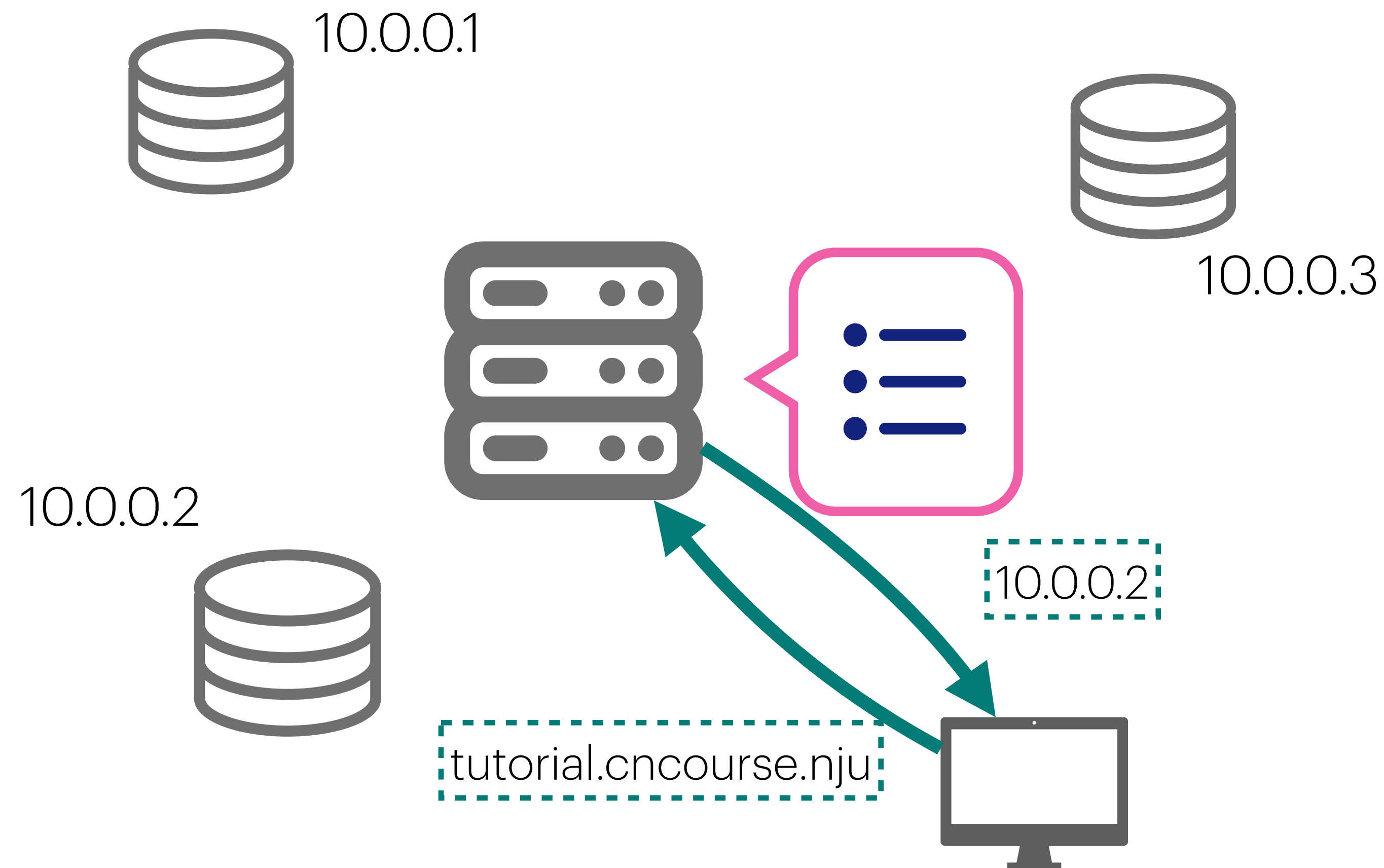
is all you need



- DNS 根据:
  - 地理距离
  - 跳数
  - RTT
  - 负载
  - .....

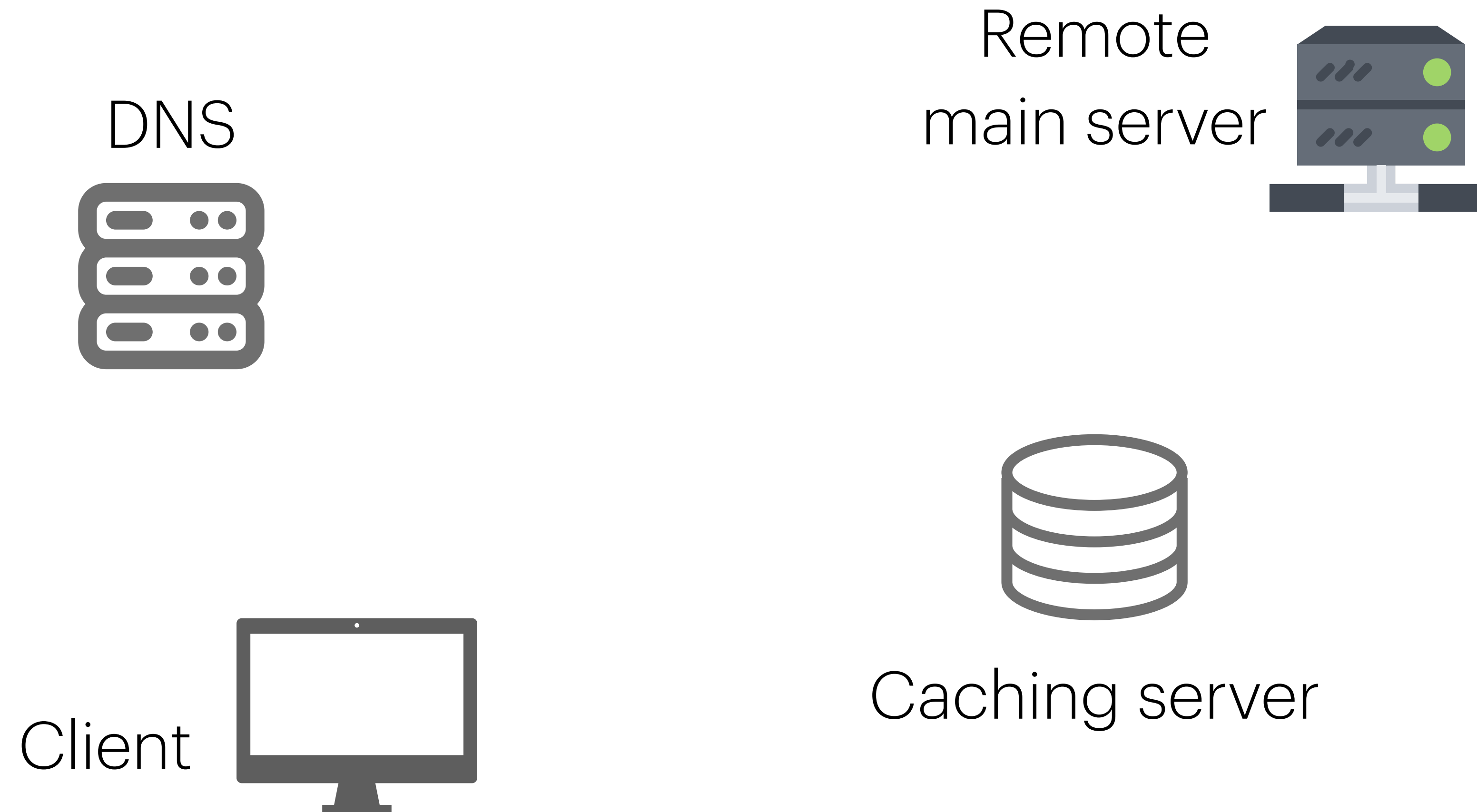
# DNS Server

is all you need

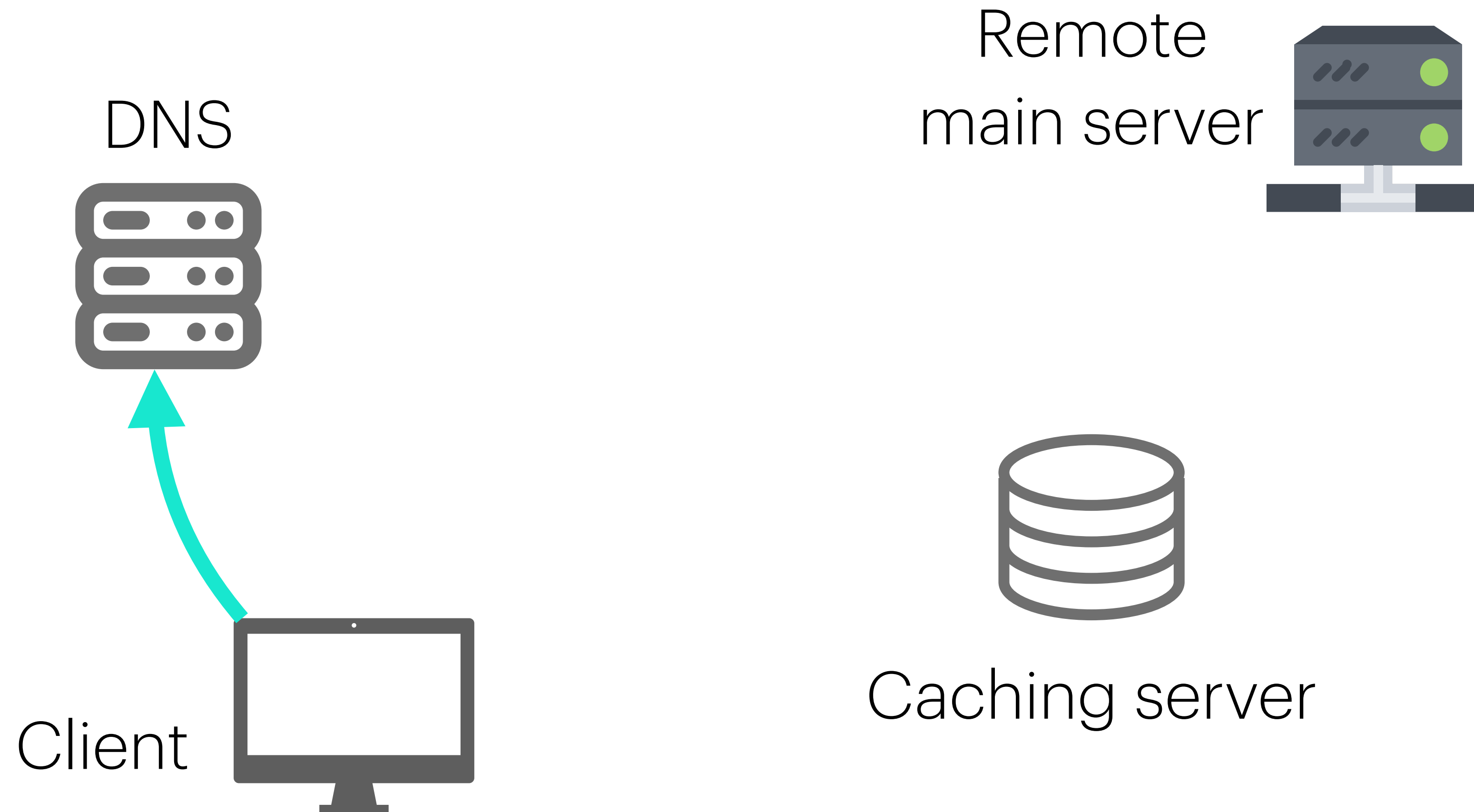


- DNS 根据:
  - 地理距离
  - 跳数
  - RTT
  - 负载
  - .....
- 也有利用 BGP + Anycast 做多路选择

# Lab overview

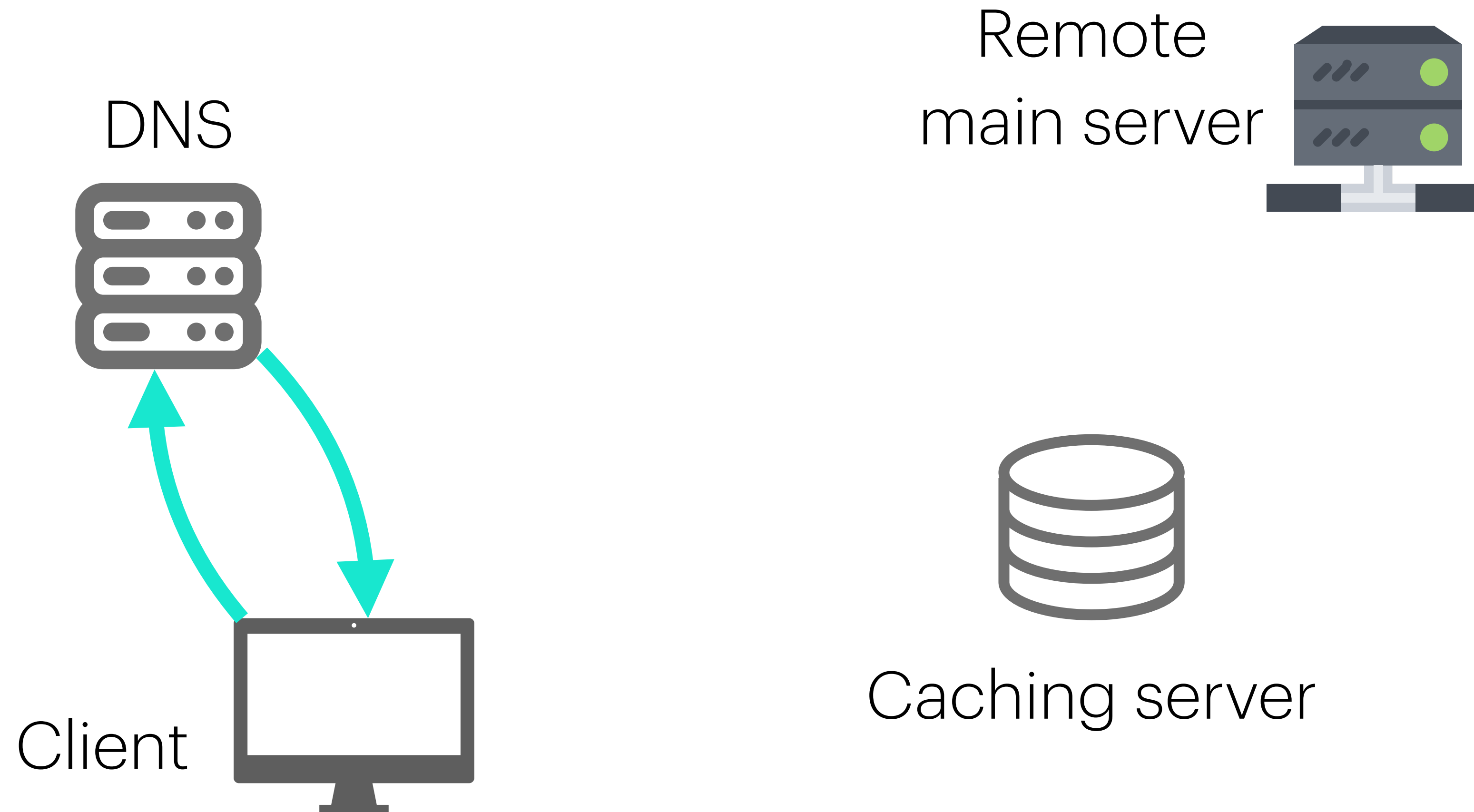


# Lab overview

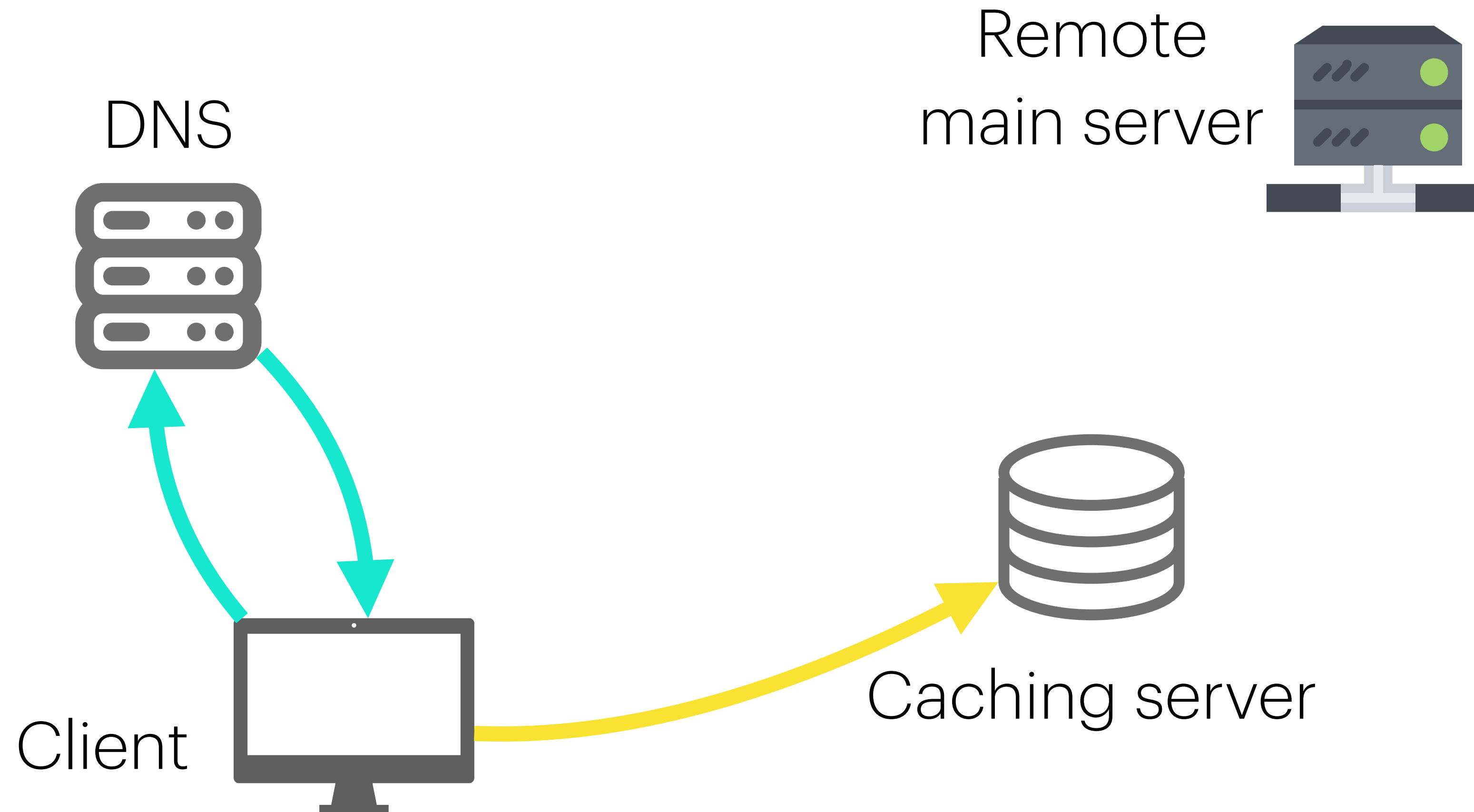




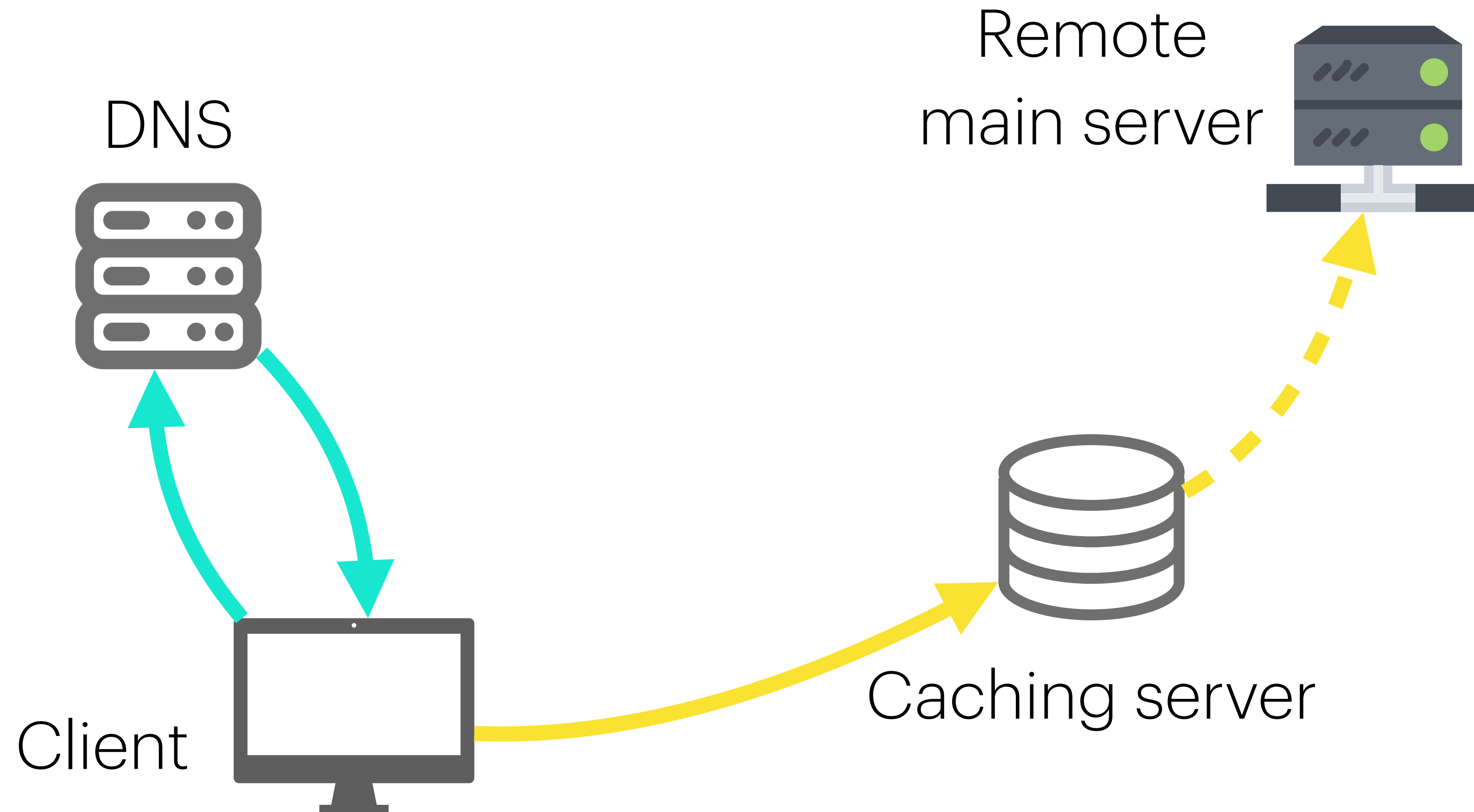
# Lab overview



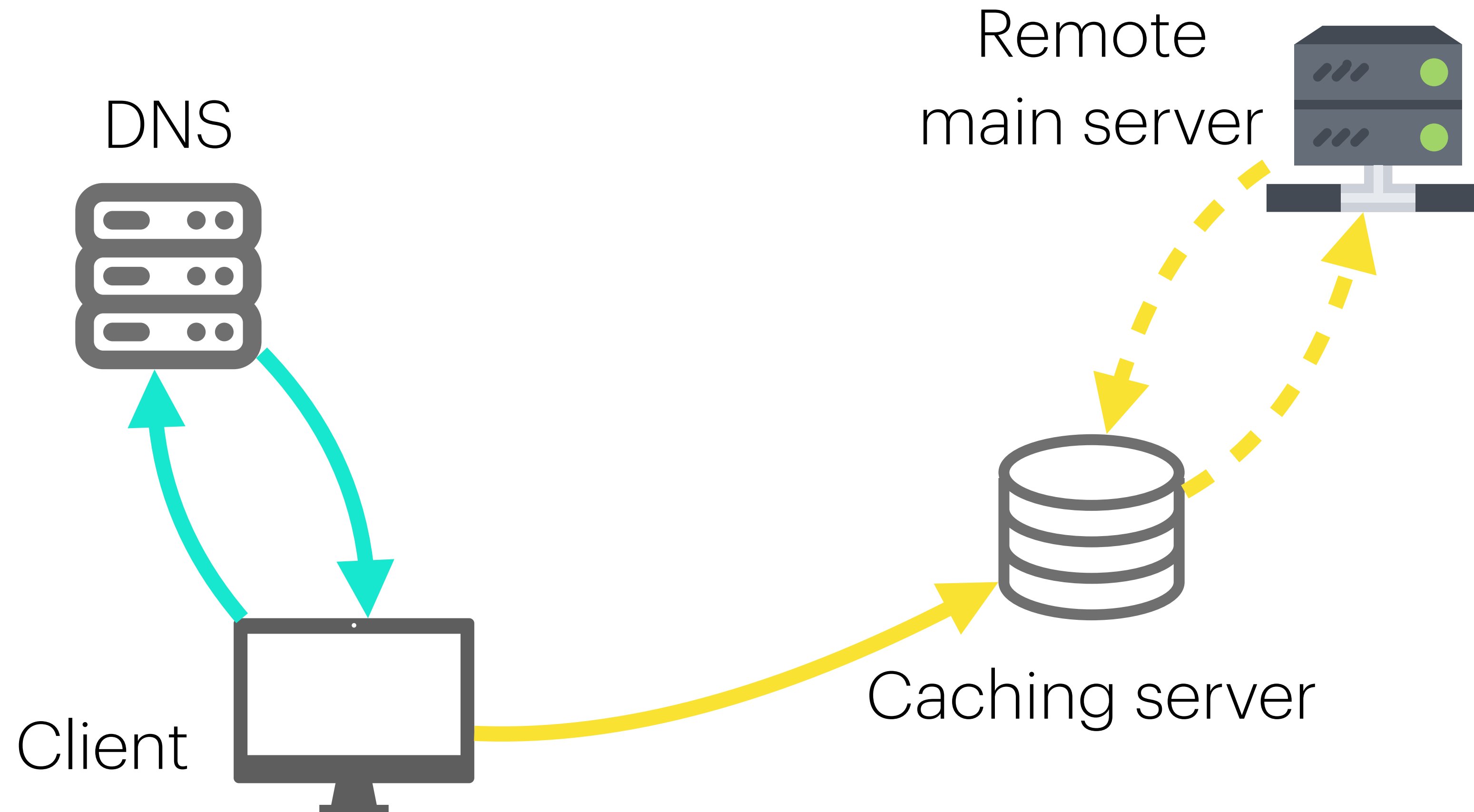
# Lab overview



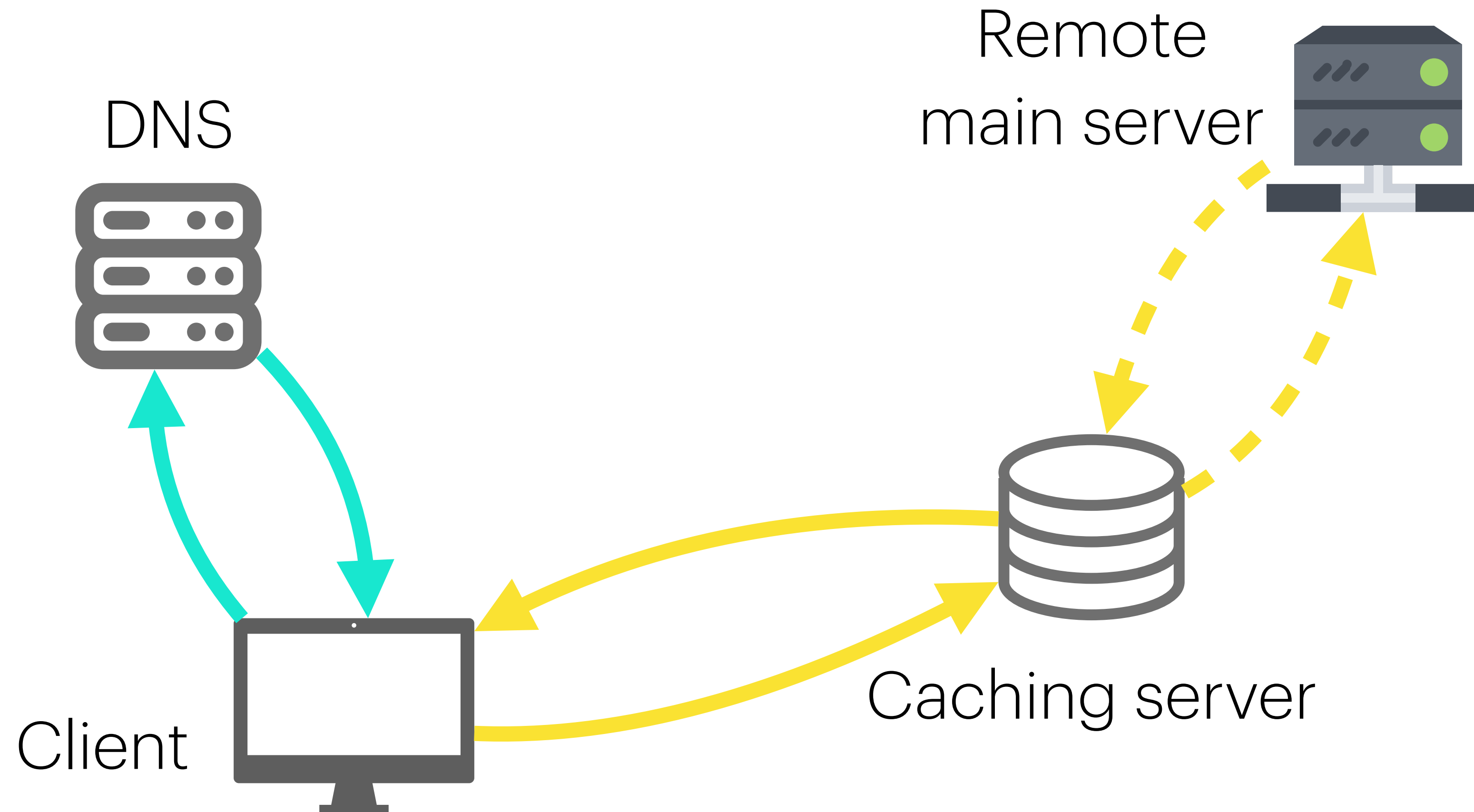
# Lab overview



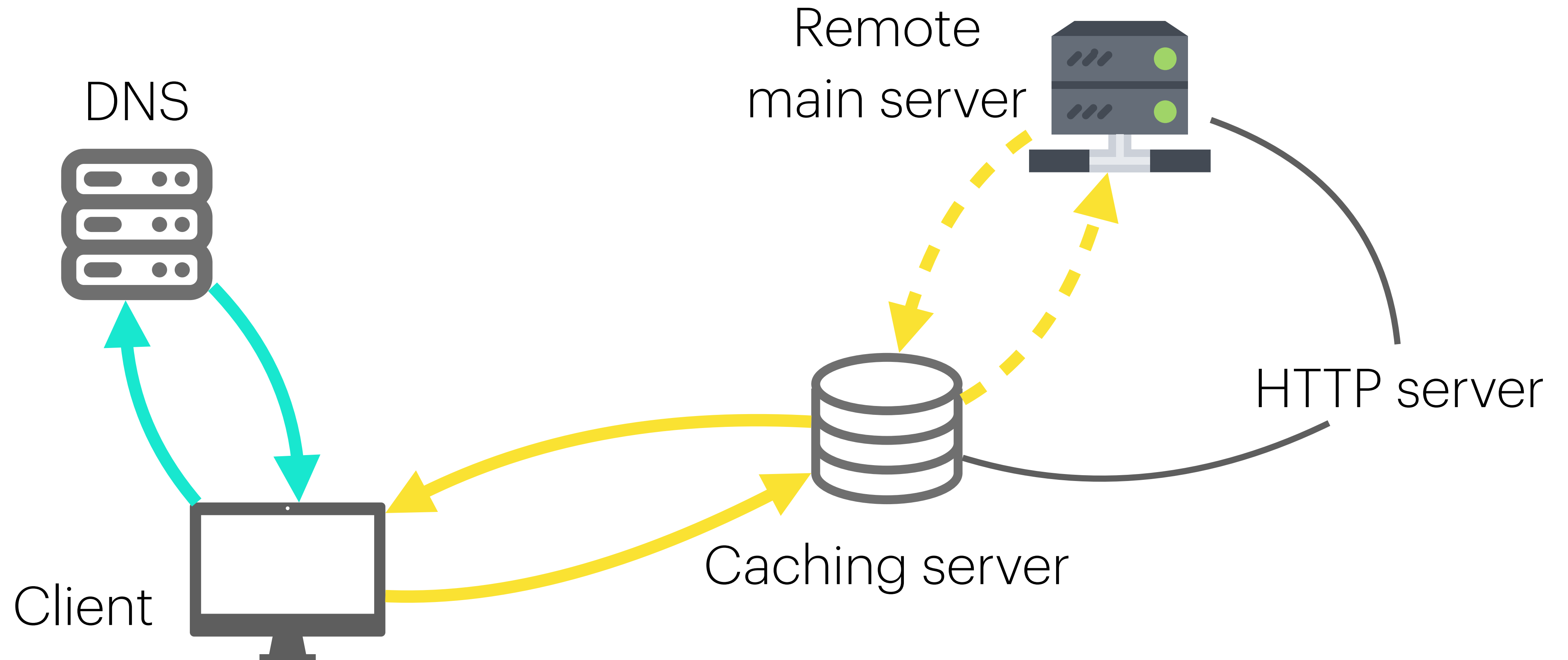
# Lab overview



# Lab overview

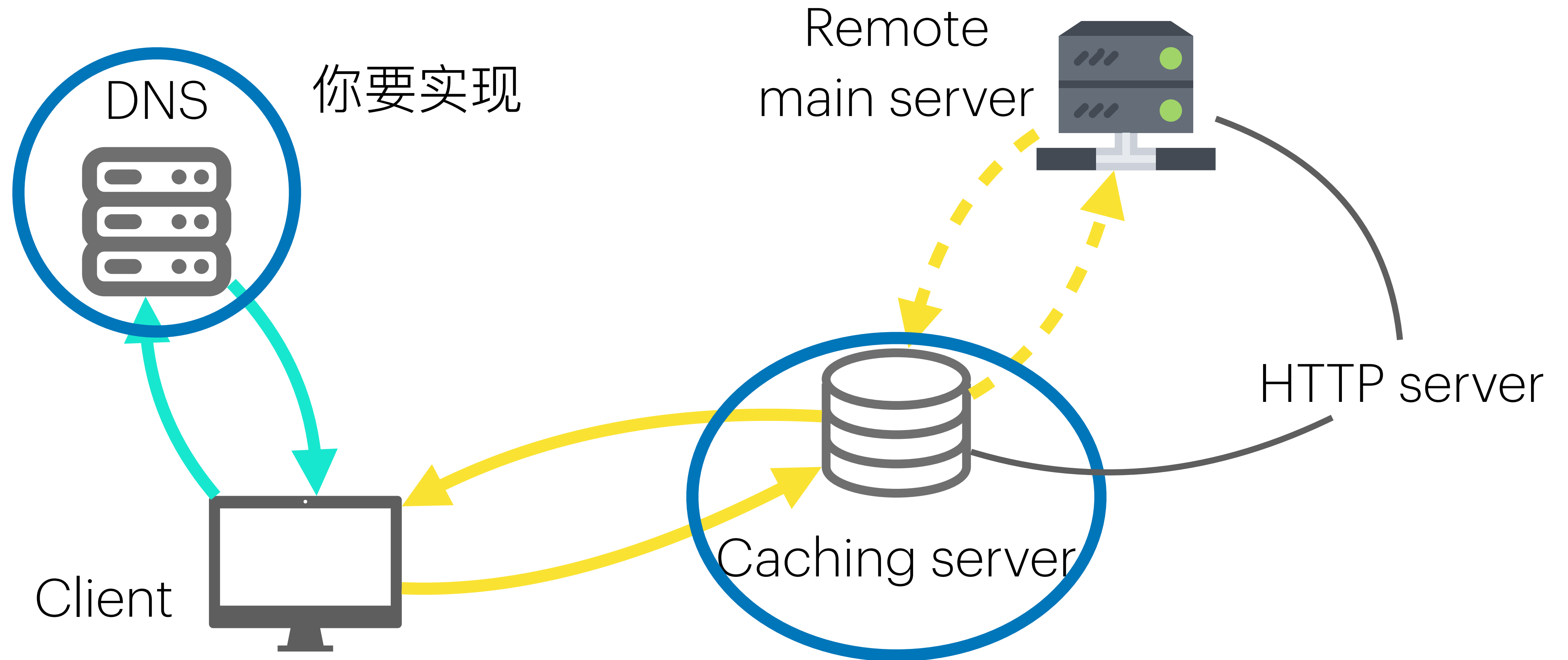


# Lab overview



# Lab overview

或许是你写的第一个分布式系统



# Details

Things that changed



# Details

Things that changed

应用层：你不再需要关注数据传输细节

# Details

Things that changed

应用层：你不再需要关注数据传输细节

不再需要 Switchyard

# Details

Things that changed

应用层：你不再需要关注数据传输细节

不再需要 Switchyard

使用 Python 的 HTTP 库 + Socket

# Details

Things that changed

应用层：你不再需要关注数据传输细节

不再需要 Switchyard

使用 Python 的 HTTP 库 + Socket

可在 Linux / macOS / Windows 完成

# Details

Things that changed

应用层：你不再需要关注数据传输细节

不再需要 Switchyard

使用 Python 的 HTTP 库 + Socket

可在 Linux / macOS / Windows 完成

通信：DNS + HTTP

# Details

Things that changed

应用层：你不再需要关注数据传输细节

不再需要 Switchyard

使用 Python 的 HTTP 库 + Socket

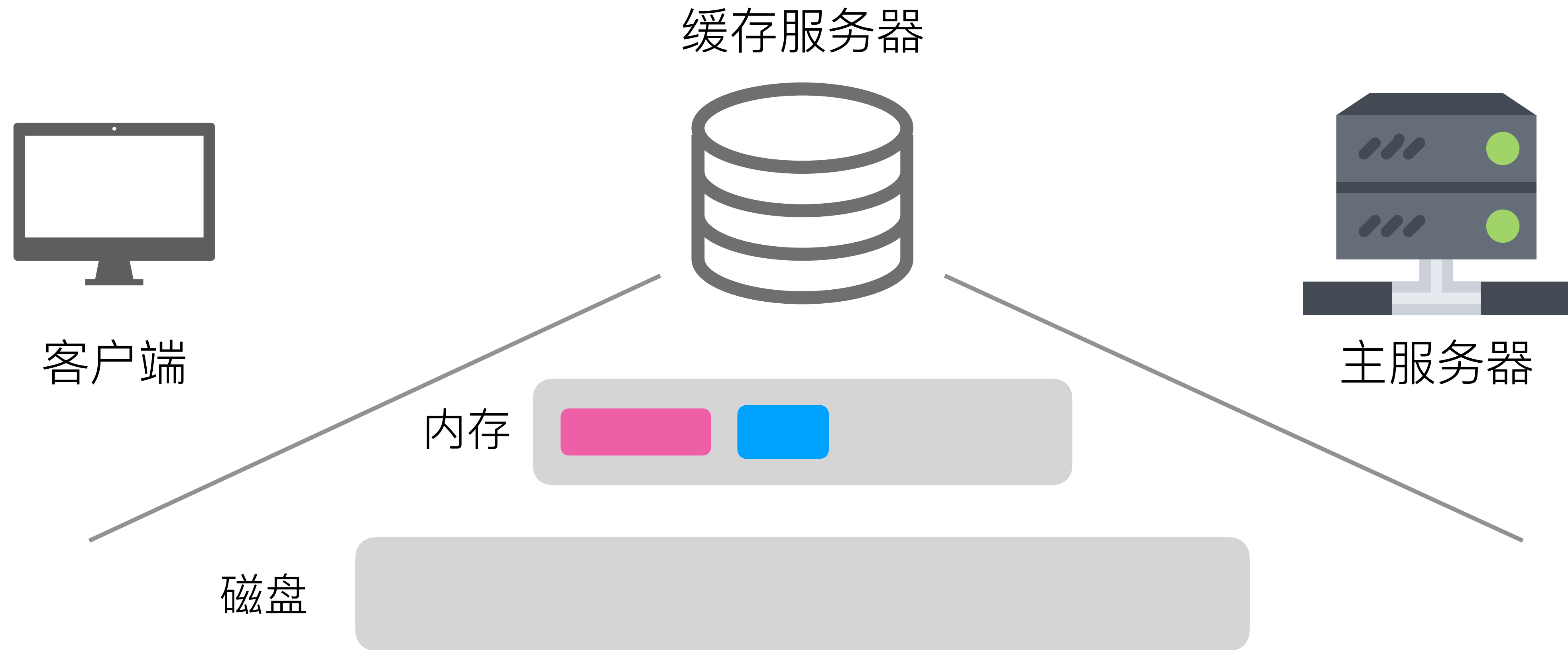
可在 Linux / macOS / Windows 完成

通信：DNS + HTTP

**你需要翻阅足够多的文档**

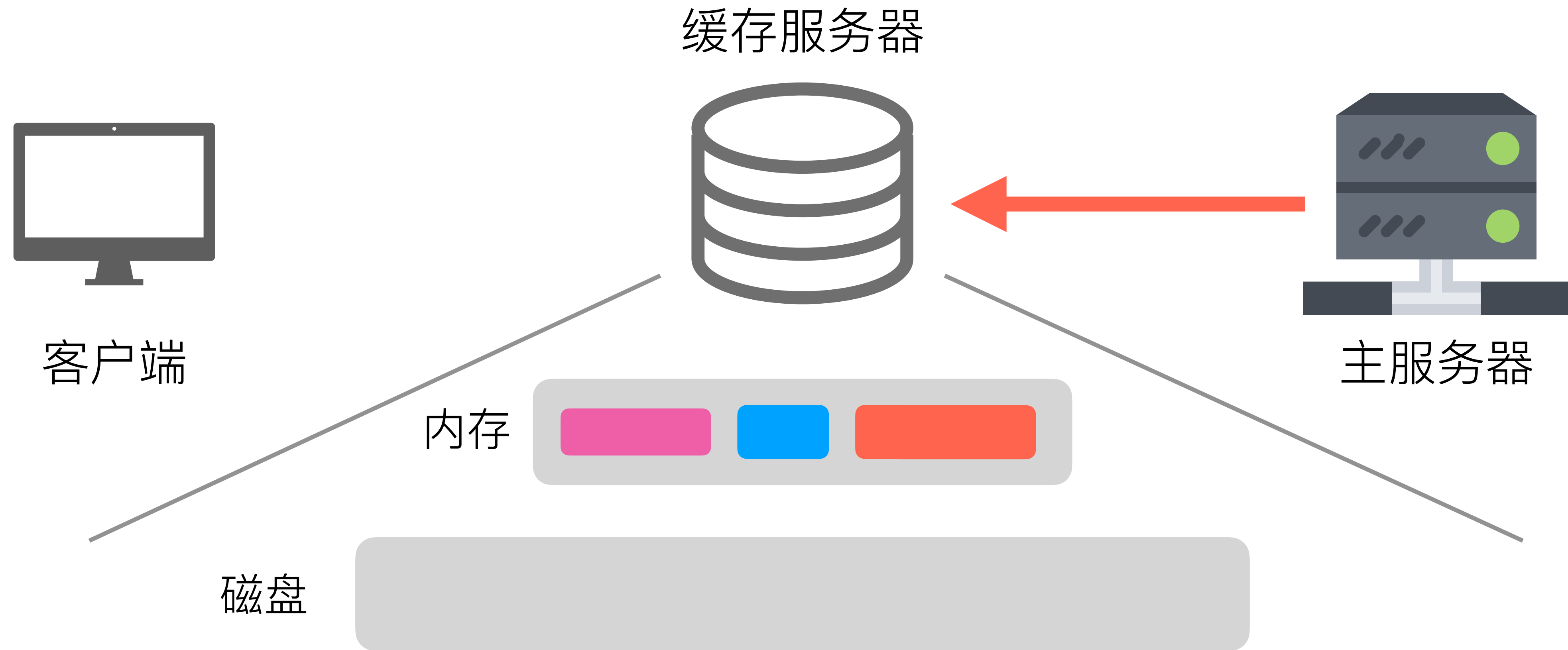
# Caching server

Where to put cache



# Caching server

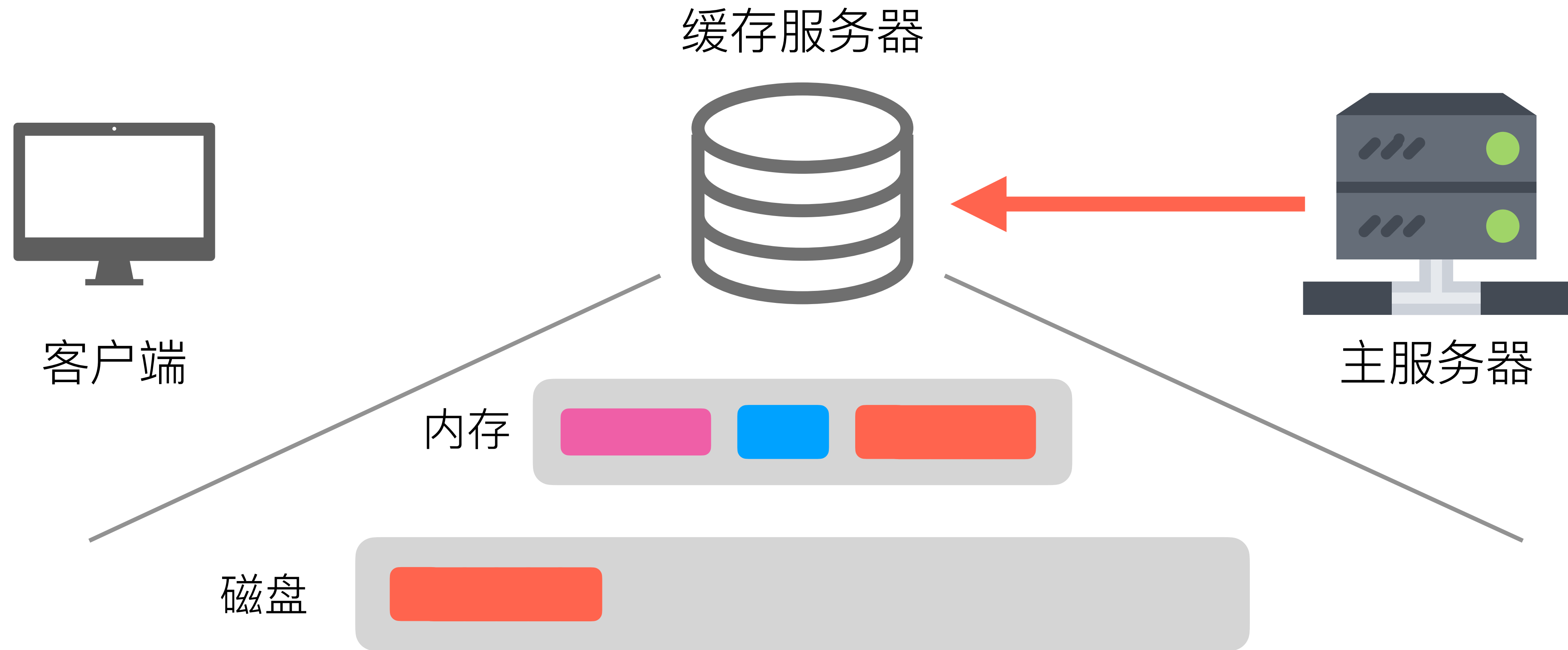
Where to put cache





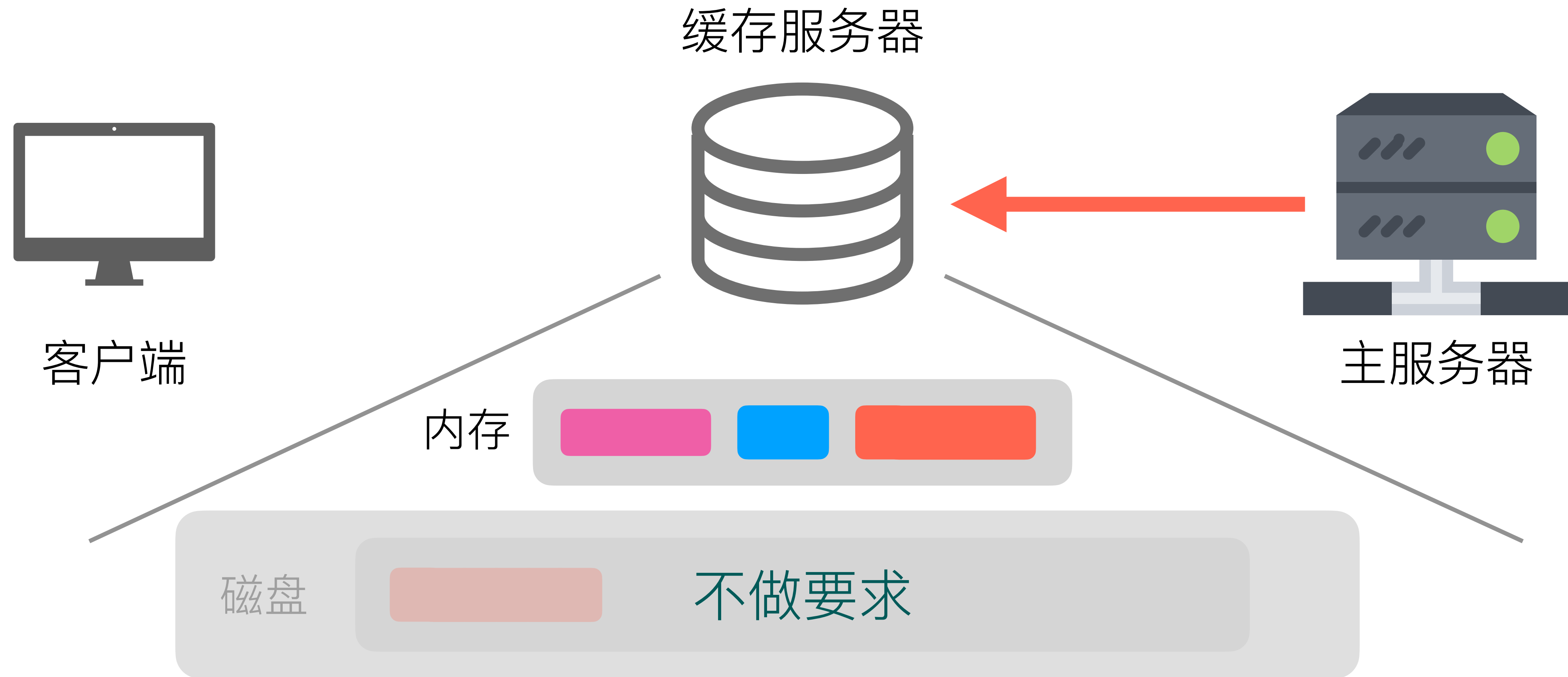
# Caching server

Where to put cache



# Caching server

Where to put cache



# HTTP basic

## URL

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#Somewhere

# HTTP basic

## URL

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#Somewhere

Scheme

# HTTP basic

## URL

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#Somewhere

Domain name

# HTTP basic

## URL

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#Somewhere

Domain name

219.219.114.172

# HTTP basic

URL

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#Somewhere

Port

# HTTP basic

## URL

└── Authority ─┘

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#Somewhere



# HTTP basic

## URL

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#Somewhere

Path to the file

# HTTP basic

URL

http :// www.example.com : 80 /path/to/myfile.html ?key1=value1&key2=value2 #Somewhere

Parameters

# HTTP basic

URL

http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#Somewhere

Anchor

# Details

How things work

# Details

How things work

## 本地运行

- 一个进程作为一个单位

你的电脑

进程 1: 远程主服务器

进程 2: DNS 服务器

进程 3: 缓存服务器

进程 4: 客户端

# Details

How things work

## 本地运行

- 一个进程作为一个单位

### 你的电脑

进程 1: 远程主服务器

进程 2: DNS 服务器

进程 3: 缓存服务器

进程 4: 客户端

## 真实部署

-  OpenNetLab

进程 1: 远程主服务器

进程 2: DNS 服务器

进程 3: 缓存服务器

进程 4: 客户端

# Details

How things work

## 本地运行

- 一个进程作为一个单位

你的电脑

进程 1: 远程主服务器

进程 2: DNS 服务器

进程 3: 缓存服务器

进程 4: 客户端

## 真实部署

-  OpenNetLab

很远

进程 1: 远程主服务器

进程 2: DNS 服务器

进程 3: 缓存服务器

进程 4: 客户端

# Disclaimer

💡 这是一个全新的实验

🧩 我们和你共同探索

👤 发现设计上的问题，欢迎指正

💡 有新的想法，欢迎交流

🚫 Deadline: 2021-6-16 24:00 (UTC+8:00)



Good luck!