
University of
Southampton

University of Southampton

Web Caching, Proxies and CDNs

COMP3227 Web Architecture & Hypertext Technologies

Dr Heather Packer – hp3@ecs.soton.ac.uk

University of Southampton

Caching

- Caching stores the result of an operation so that future operations return faster
  - Computation is slow
  - Computation will run multiple times
  - When the output is the same for a particular input

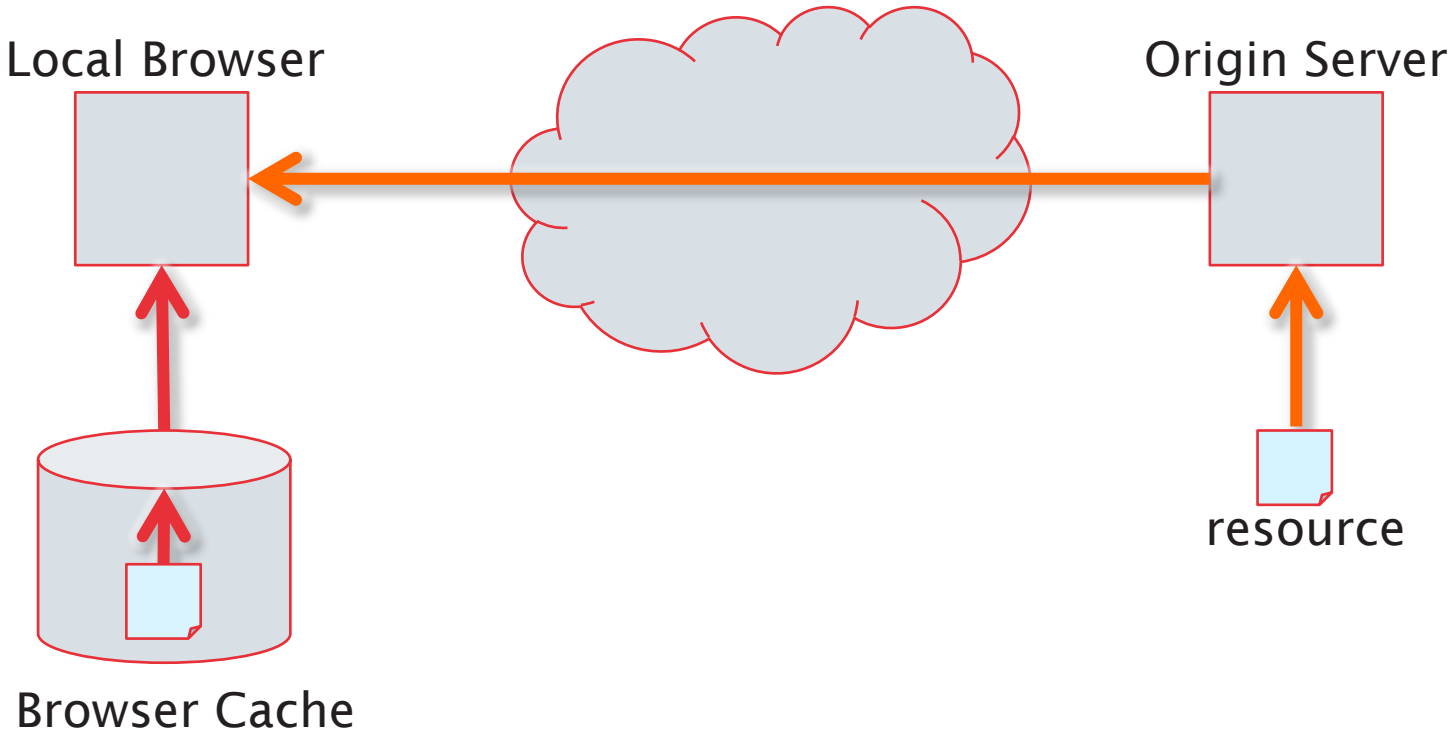
University of Southampton

Web Caching

- The temporary storage (caching) of frequently accessed data for rapid access
- Typically caches store “static assets”
  - HTML pages
  - Images
  - Stylesheets, Javascript
- Caches can be located at various points in a network
  - Reduces access time/latency for clients
  - Reduces bandwidth usage across slower links
  - Reduces load on a server

Browser Cache

- Browsers maintain a small cache
- Stored locally
- Cache for a single user or application
- Browser sets a caching policy, deciding what data to cache
  - User specific content
  - Expensive content



University of Southampton

What can be Cached?

- Cache friendly
  - Logos and brand images
  - Style sheets
  - Javascript files, site and library
  - Fonts
  - Downloadable content
  - Media files
- Be careful caching:
  - Data
  - HTML pages
  - Frequently modified Javascript and CSS
  - Content requested with authentication cookies
- Never cache
  - Sensitive data
  - User-specific data that frequently changes

University of Southampton

Caching using Conditional requests

- Last-Modified: Tue 17 Nov 2020 08:00:20 GMT
  - GET Requests: If-Modified-Since:
  - HTTP Status code: 304 Not Modified
- Etag: “0123456789ABCDEF…”
  - GET Request: If-None-Match:
  - HTTP Status code: 304 Not Modified

Controlling Caches with HTTP: Last-Modified Header

GET

```
GET / HTTP/1.1
Host: comp3220.ecs.soton.ac.uk
Accept: */*

HTTP/1.1 200 OK
Date: Wed 18 Nov 2020 17:43:20 GMT
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Content-Length: 4003
Last-Modified: Tue 17 Nov 2020 08:00:20 GMT
```

Conditional GET

```
GET / HTTP/1.1
Host: comp3220.ecs.soton.ac.uk
Accept: */*
If-Modified-Since: Tue 17 Nov 2020 08:00:20 GMT

HTTP/1.1 304 Not Modified
Date: Wed 15 Nov 2017 07:55:10 GMT
Connection: keep-alive
Last-Modified: Tue 17 Nov 2020 08:00:20 GMT
```

University of Southampton

Caching Headers

- HTTP Response header Cache-Control:
- Flags
  - no-store
  - no-cache
  - max-age=
  - must-revalidate
  - public
  - private
- Use of Cache-Control headers to be determined by website architect/designer

HTTP with Cache-Control Header

GET

```
GET / HTTP/1.1
Host: comp3220.ecs.soton.ac.uk
Accept: */*

HTTP/1.1 200 OK
Date: Wed 18 Nov 2017 17:43:20 GMT
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Content-Length: 4003
Last-Modified: Tue 17 Nov 2020 08:00:20 GMT
Cache-Control: max-age=86400
```

GET

```
* No request sent *
```

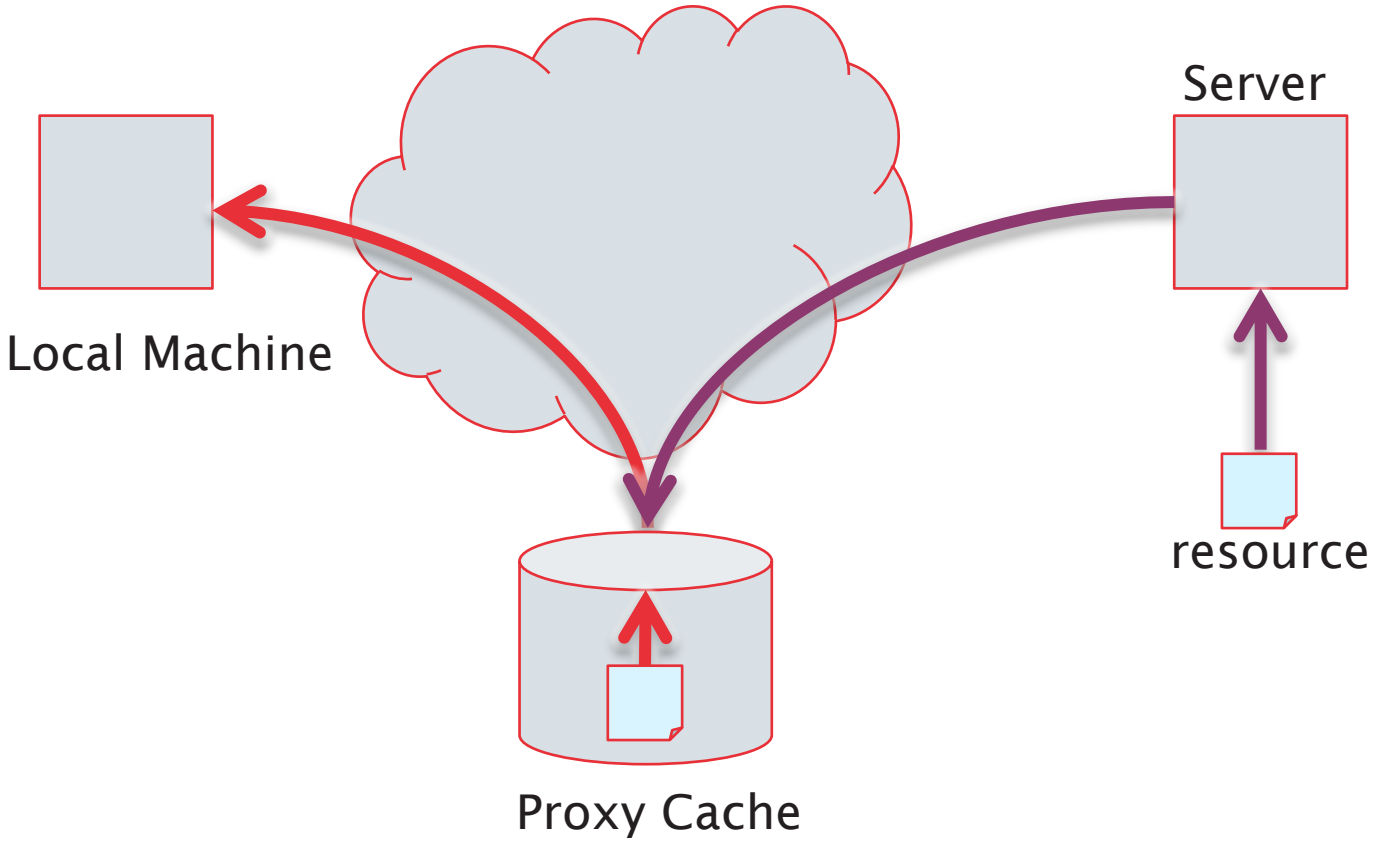
University of Southampton

Different Web Caching Solutions

- Caches can be located at various points in a network
  - Browser Cache
    - Embedded in the browser
  - Proxy Cache
  - Reverse Proxy Cache

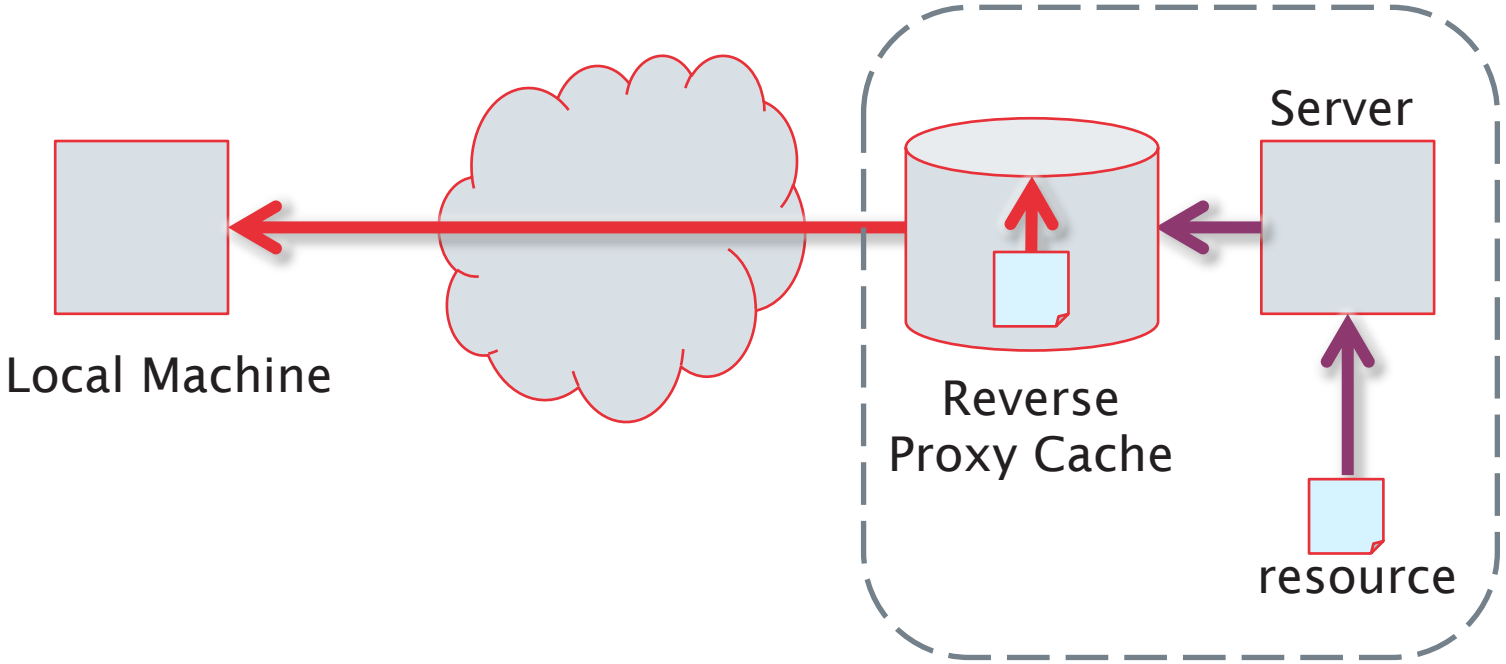
Proxy Cache

- Cache located close to the clients (hosted by University or Internet Service Provider)
  - Decrease bandwidth usage
  - Decreases network latency
- Scale provides the main advantage: many users within the ISP may all be asking for the same web pages
- ISPs use this approach to decrease bandwidth across their networks



Reverse Proxy Cache

- Cache proxy located closer to the origin web server
- Usually deployed by a Web host
- Decreases load on the Web service (e.g. database)
- Several reverse proxy caches implemented together can for a Content Delivery Network

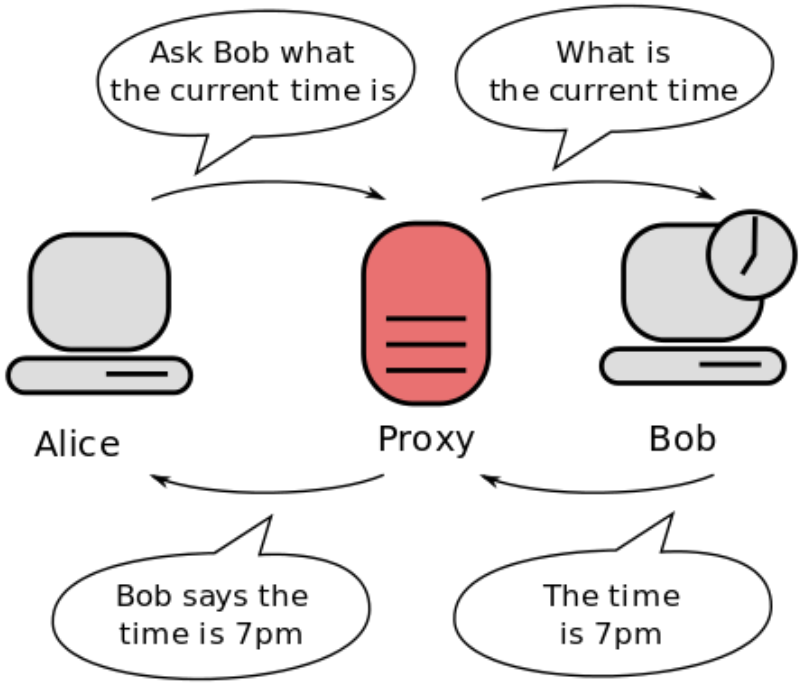


University of Southampton

Web Proxies

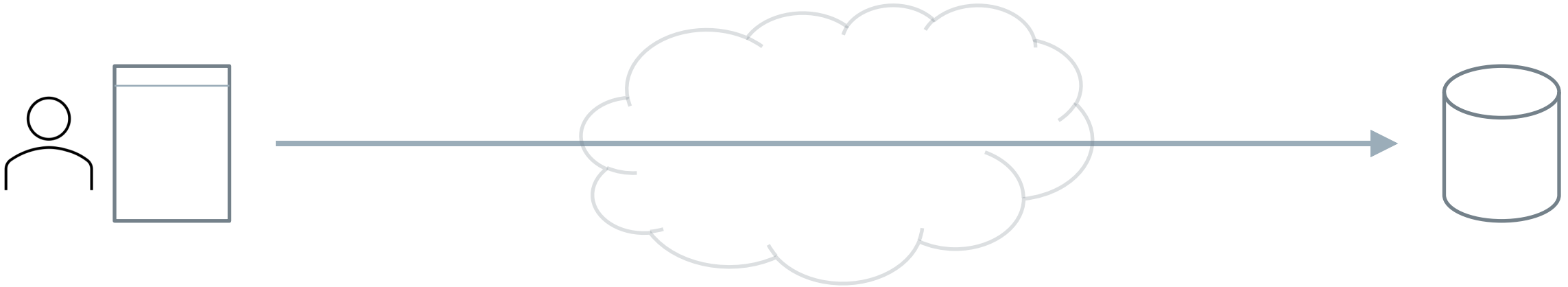
Web Proxy Architecture

- A web proxy is a network service
- They receive web requests from clients and make requests on their behalf to web servers
- A web proxies behaviour differs depending on their function



University of Southampton

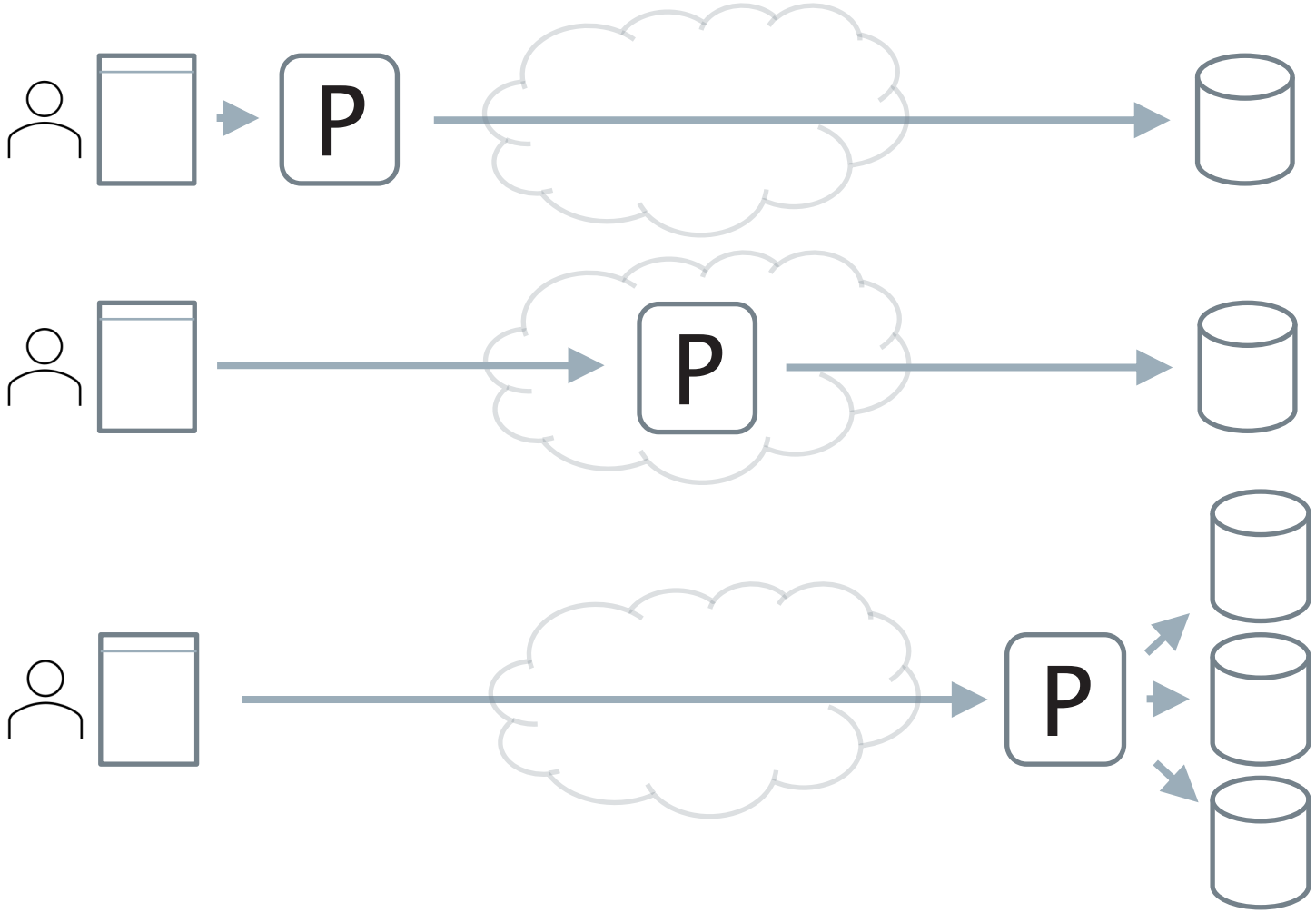
No Proxy



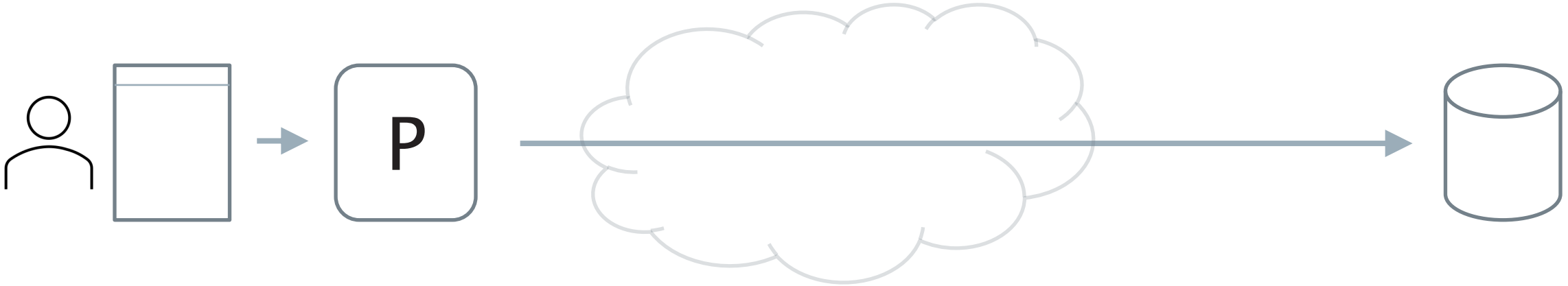
University of Southampton

Proxy Types

- Forward Proxy
- Open Proxy
- Reverse Proxy



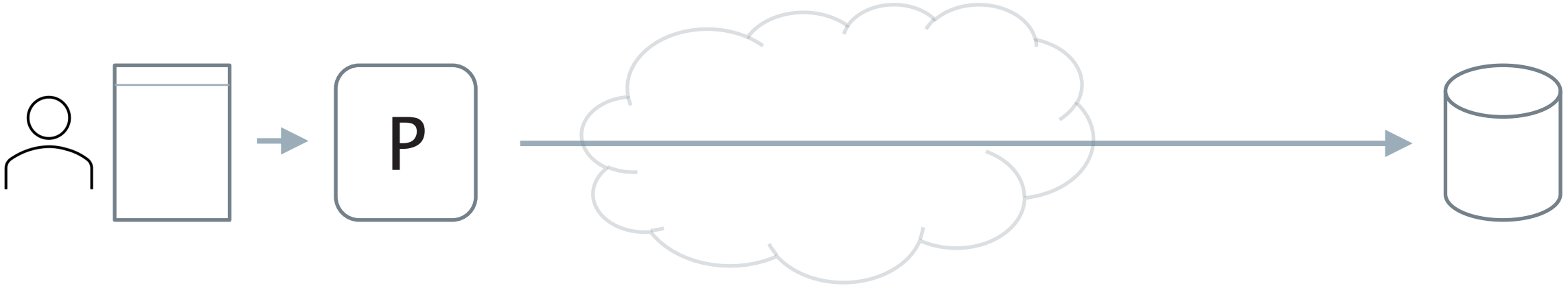
Forward Proxy – Content Filtering



Restricts requests

- Blocks blacklisted URLs
- Response can be scanned
  - Unwanted content
  - Malware

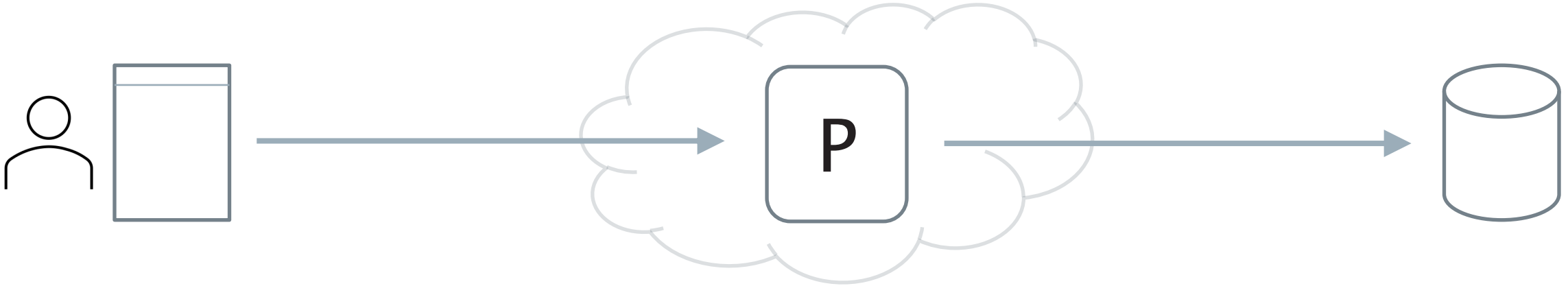
Forward Proxy – Content Translation



Transform responses

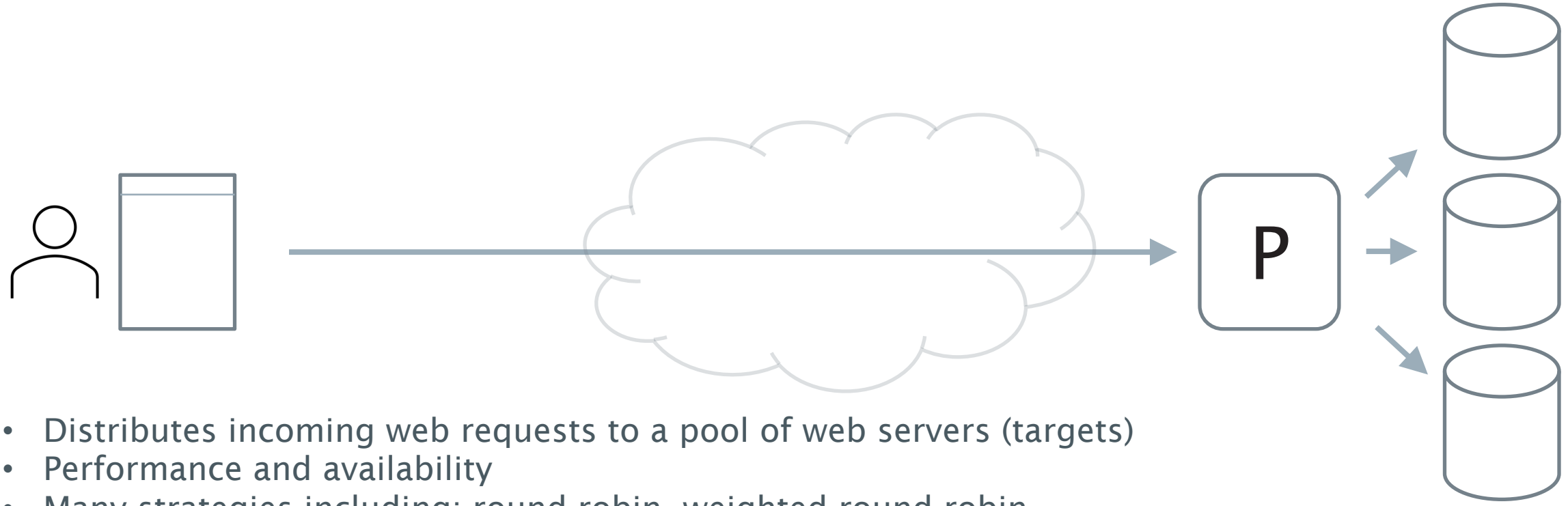
- Reduce bandwidth usage by client
  - Commonly used by mobile phone networks
  - Recompress images at lower resolution
  - Minimising HTML, CSS and JavaScript
- Inject content into web pages e.g. adverts

Open Forward Proxy - Access Services Anonymously



- Client accessing website via proxy:
  - Masks their IP address
  - Modifies their location (via GeoIP)
- Improve anonymity
- Defeat geo-blocking
- No filtering, encryption, or checks over content

Reverse Proxy – Load-balancing



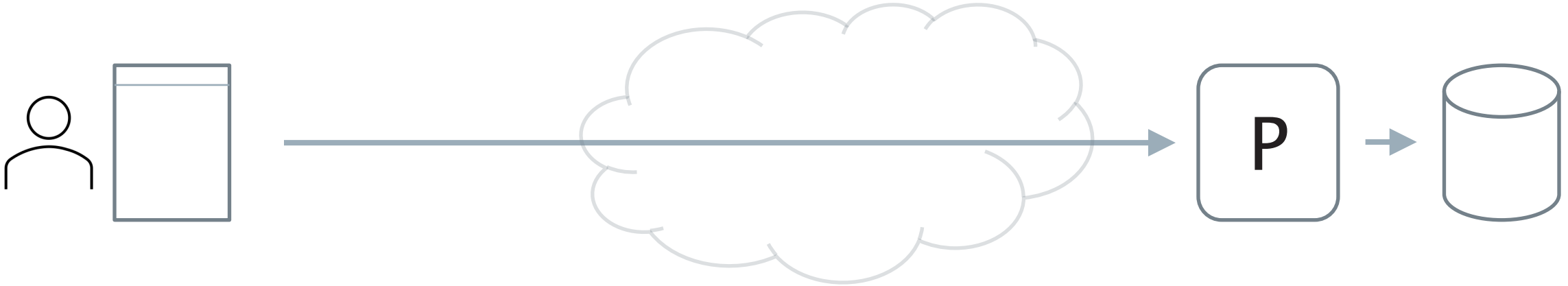
- Distributes incoming web requests to a pool of web servers (targets)
- Performance and availability
- Many strategies including: round robin, weighted round robin
- Health checks ensure resilience

Reverse Proxy – Content Switching

Examine incoming request and direct traffic to specific web servers
Can use:

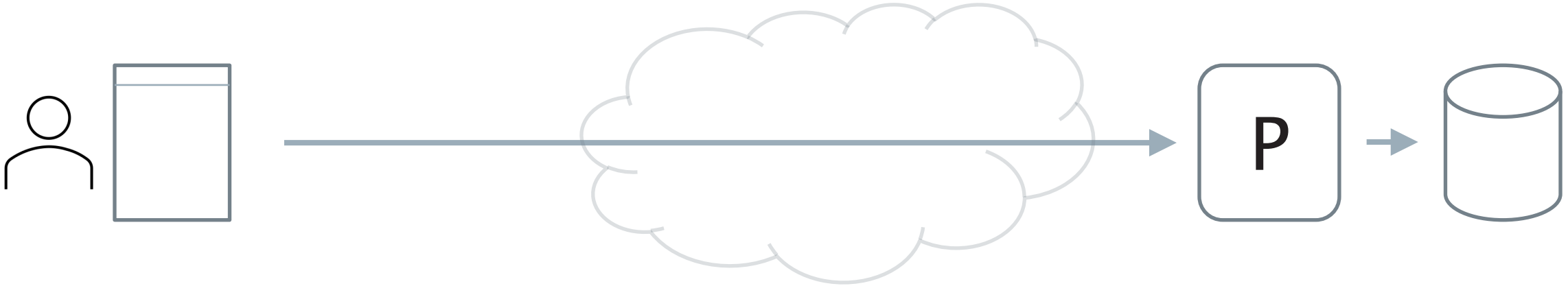
- Host or path based e.g. https://example.com/static -> static webserver
- IP address, cookie or user-agent

Reverse Proxy – Protocol Translation



- HTTPS SSL/TLS off-loading (OSI Layer 6)
  - Incoming web requests over https, internal communications via http
- HTTP/2 to HTTP/1.x (OSI Layer 7)
- IPv6 <-> IPv4 (OSI Layer 3)

Reverse Proxy – Monitoring and filtering



- Monitor incoming requests
  - Access logs / statistics
- Filter incoming requests
  - Check security credentials (eg valid API tokens)
  - Rate limit requests
  - Intrusion detection and handling DDoS attacks
  - Applying security policies (WAF)

University of Southampton

Content Delivery Networks

Motivation Scenario

Stream video content to 100,000+ simultaneous users

- You could use a single large “mega-server”
  - Single point of failure
  - Point of network congestion
  - Long path to distant clients
  - Multiple copies of video sent over outgoing link
- This solution **doesn’t** work in practice

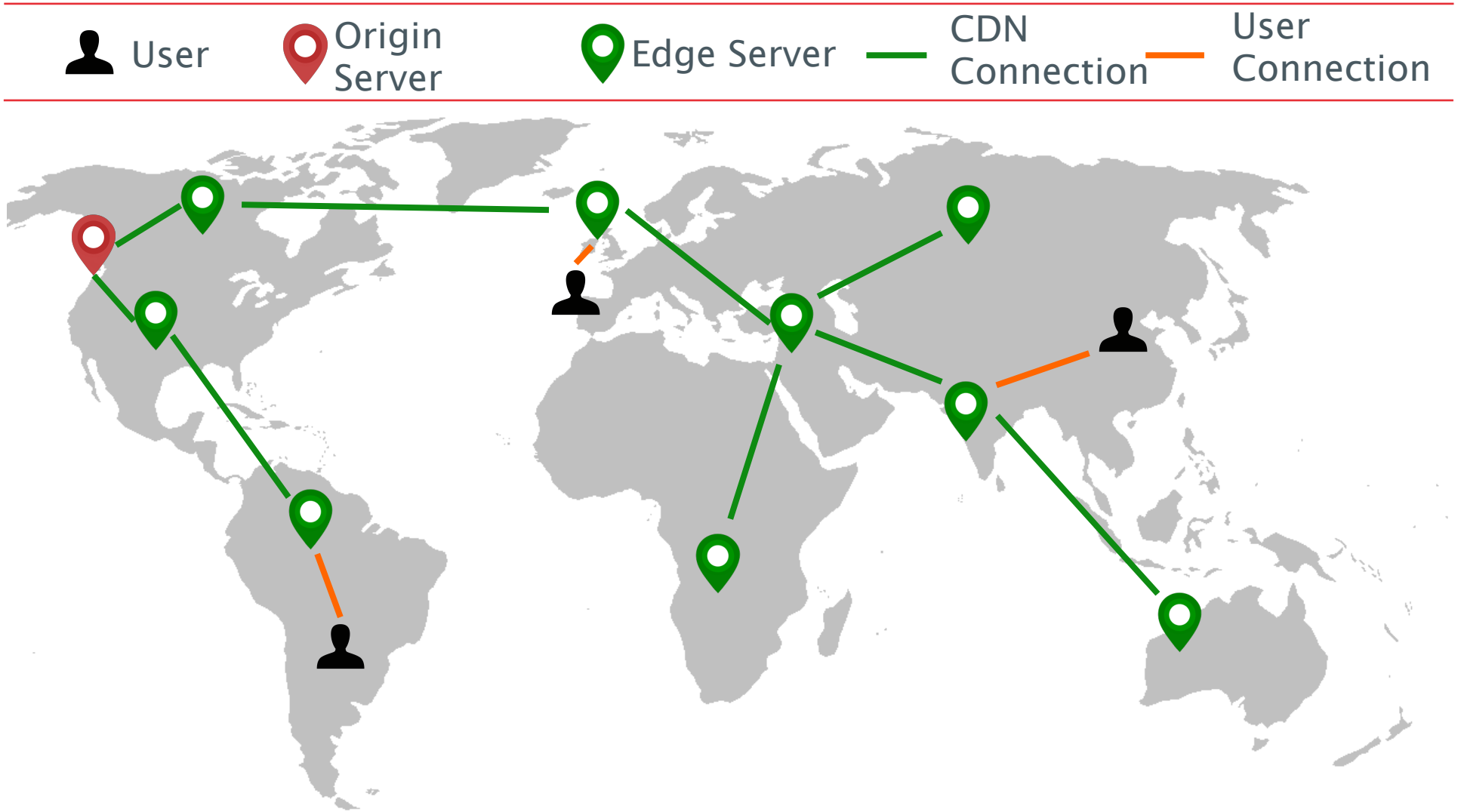
University of Southampton

Content Delivery Network

- CDN a geographically distributed network of proxy servers (edge nodes)
- Hosts static content (such as images, videos, CSS and JS)
- Data travels to user via the shortest path (reduced latency)

University of Southampton

CDN



CDN

- **Services that use CDNs**
  - Netflix
  - Amazon
  - Reddit
  - Twitch
  - gov.uk
  - PayPal
  - Shopify
  - BBC.com
  - Vimeo
  - YouTube
  - Hulu
  - Wikipedia
  - CNN
  - New York Times
  - The Guardian
  - Stack Overflow
  - GitHub
  - Stripe
  - Quora

- Video streaming
- Software downloads
- Web and mobile content acceleration
- Payment services
- E-commerce
- News

University of Southampton

Commercial CDNs

- Limelight Networks
- Level 3 Communications
- Akamai Technologies
- Amazon CloudFront (cloudping.info)
- CloudFlare

Motivational Scenario

Streaming video to 100,000+ simultaneous users

- Working Web solution: store/serve many copies of video at multiple geographically distributed sites (CDN)
- Two strategies:

1. **Push CDN servers deep into many access networks**

   Better latency and better network performance.

   Harder to maintain because there are many more servers in the CDN.

2. **Place larger clusters at key points in the network near internet exchanges**

   Higher latency and lower performance for the end user

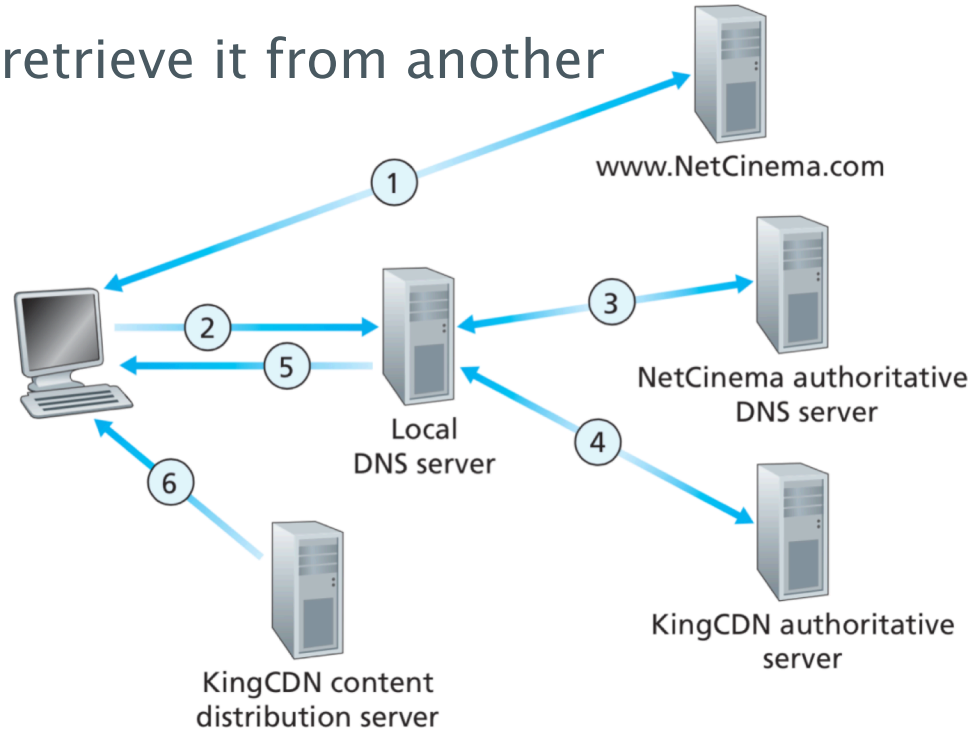
   Easier to manage but with higher latency and lower performance for the end user.

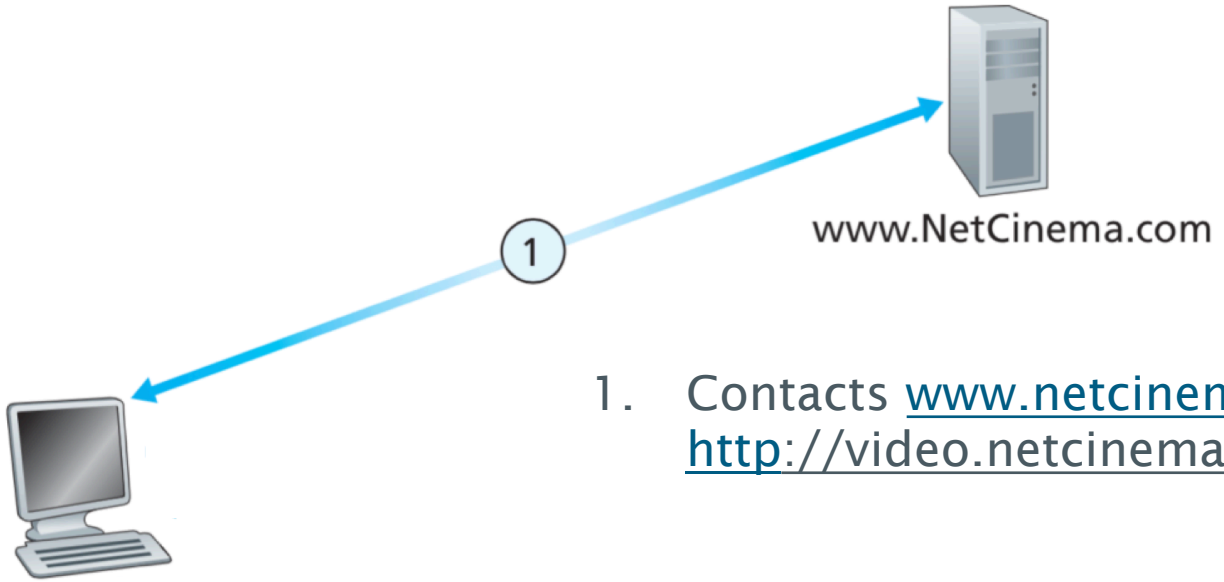
CDN: Simple content access scenario

- A CDN has to be able to tell clients where to find resources
- A client will request a file, with one URL but retrieve it from another

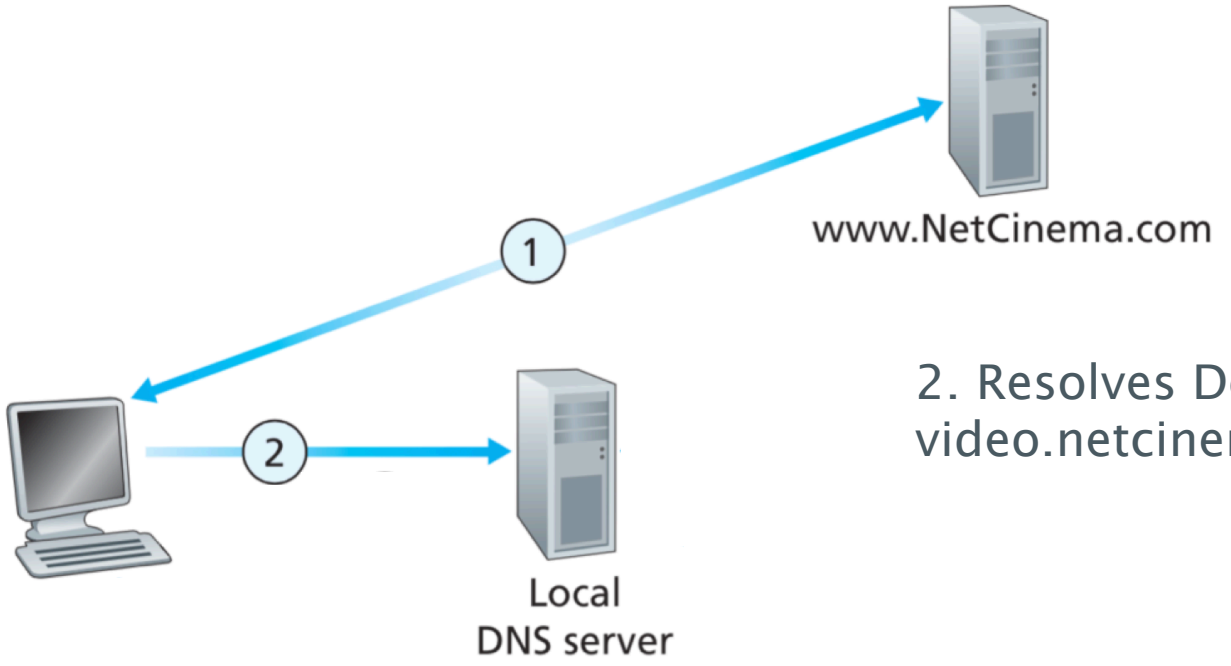
http://video.netcinema.com/6Y7B23V

http://KingCDN.com/NetC6Y7B23V

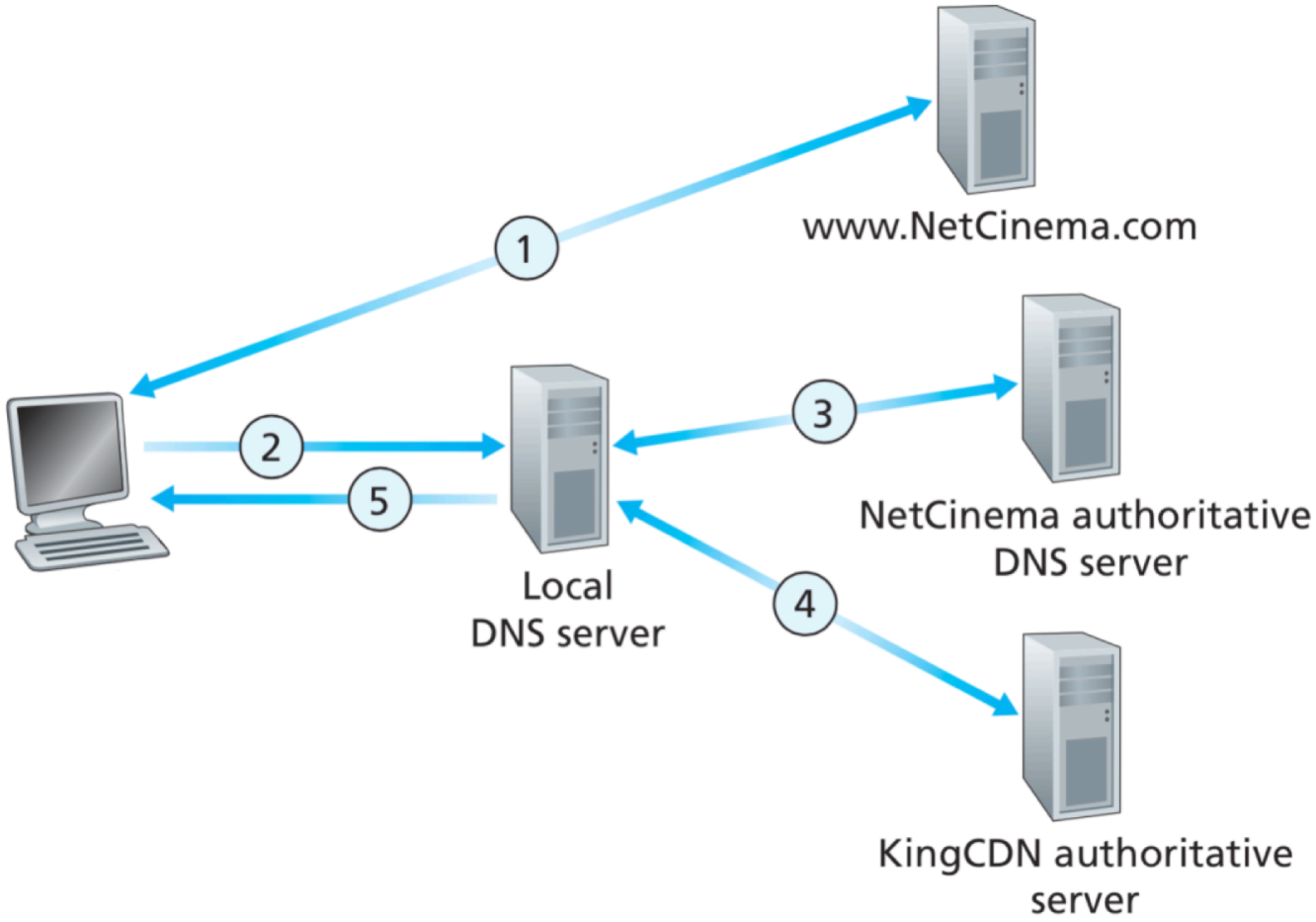




1. Contacts www.netcinema.com and receives a link to a video http://video.netcinema.com/6Y7B23V



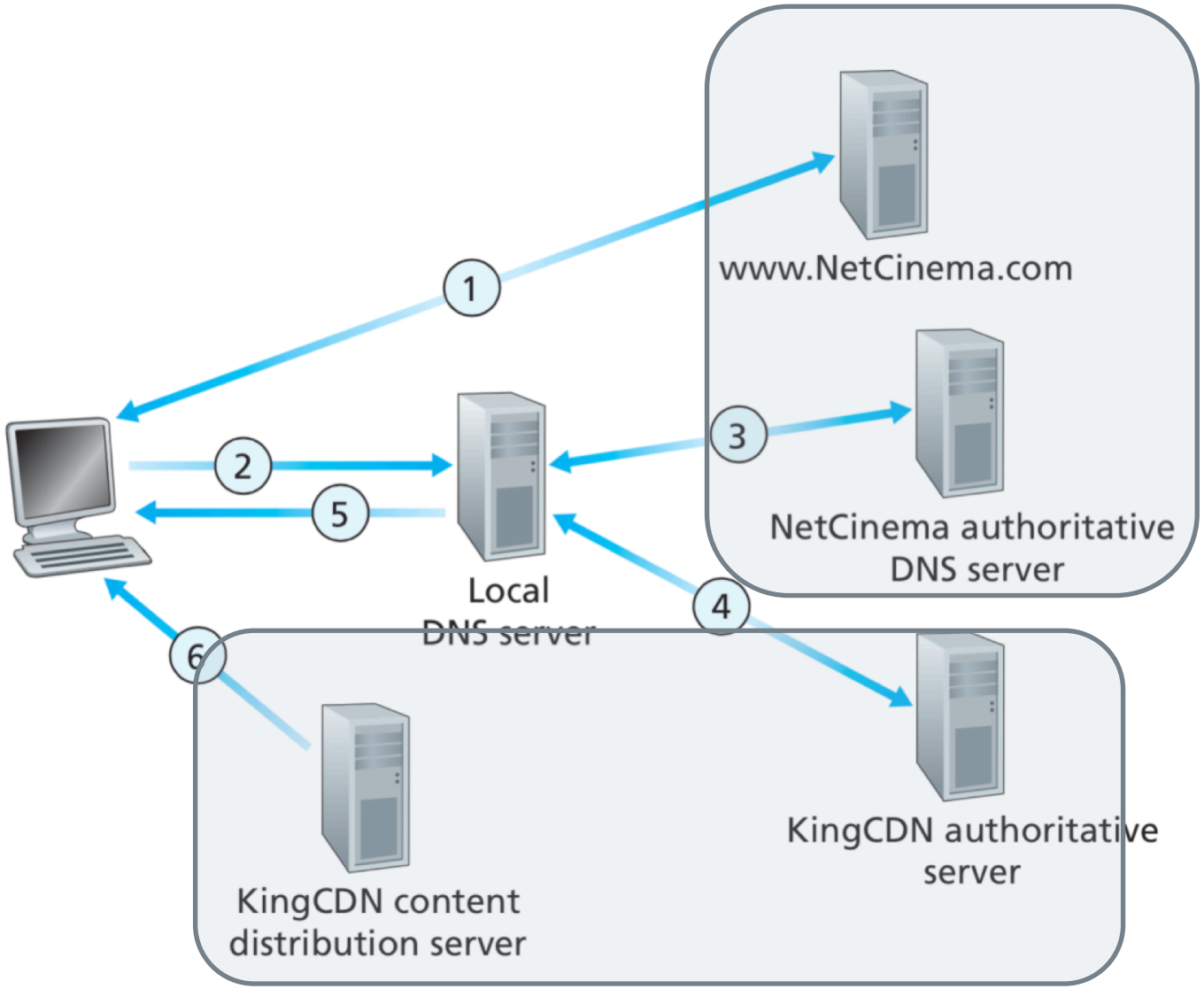
2. Resolves Domain name via local DNS
video.netcinema.com



3. NetCinema's DNS returns cdn1.KingCDN.com

4. Resolves Domain name cdn1.KingCDN.com

5. Returns IP of cdn1.KingCDN.com



6. Requests URL
http://cdn1.KingCDN.com/NetC6Y7B23V

University of Southampton

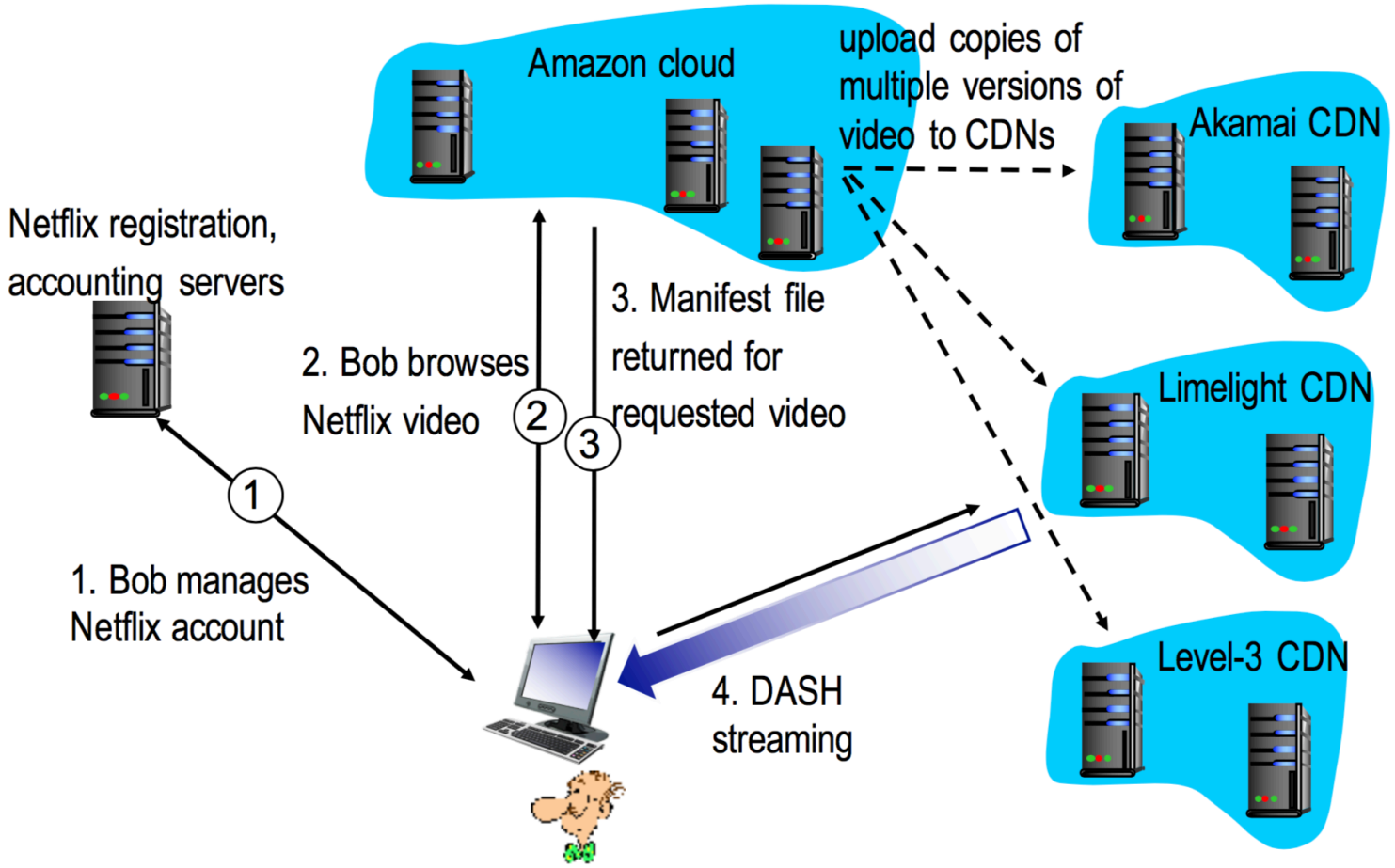
CDN Cluster Selection Strategy

- The CDN's DNS decides which edge server to use
  - Pick CDN node geographically closest to client
  - Pick CDN node with shortest delay (min hops) to client (CDN nodes periodically ping access ISPs, report results to CDN DNS)
- Or let the Client decide – give client a list of several CDN servers

Case Study: Netflix's first Approach

- Owned very little infrastructure, uses 3rd party services
  - Own registration, payment servers
  - Amazon (3rd party) cloud services
    - Netflix uploads studio master to Amazon cloud
    - Create multiple version of movie (different encodings) in cloud
    - Upload versions from cloud to CDNs
  - Three 3rd party CDNs host/stream Netflix content: Akamai, Limelight, Level-3

Case Study: Netflix



University of Southampton

DASH - Dynamic Adaptive Streaming over HTTP

- Server
  - Divides video files into multiple chucks
  - Each chunk stored encoded at different bit rates
  - Manifest file: provides URLs for different chunks
- Client
  - Periodically measures server-to-client bandwidth
  - Consulting manifest, requests one chunk at a time
  - Chooses maximum coding rate sustainable given current bandwidth
  - Can choose different coding rates at different points in time (depending on available bandwidth at time)
- The intelligence happens at the client level to make sure that there is no buffer starvation or overflow

University of Southampton

MPEG-DASH Adoption

- MPEG DASH is independent, open and international standard, which has broad support from the industry
- HTML5 Media Source Extensions and HbbTV are MPEG-DASH enabled
- Heavy plugins like Silverlight and Flash perform poorly and cause security issues

- Chrome dropped the Silverlight support
  - It was a problem for the majority of premium video providers
  - Video providers delivered their streams via Smoothstreaming and Playready DRM, which enforced Silverlight
  - These providers switch to using HTML5 with MPEG-DASH and MPEG-CENC based DRM

University of Southampton

Netflix OpenConnect

- Netflix wanted the absolute best streaming they could get, while lowering cost
- High optimised for delivery large files, still use Akamai for small assets
- Data centers around the world
  - There may be a data center with a couple of racks that contain the entire Netflix library
  - Others might only have 80% of the most popular content
- Unpopular material will have to travel further
- Client Intelligence
  - Calculates best edge server to use (based on bit rate and closeness)
  - Selects which edge server based on the required bit rate and latency
  - Continually probes the best way of receiving content

University of Southampton

Next: Web Advertising