Università degli Studi di Bergamo

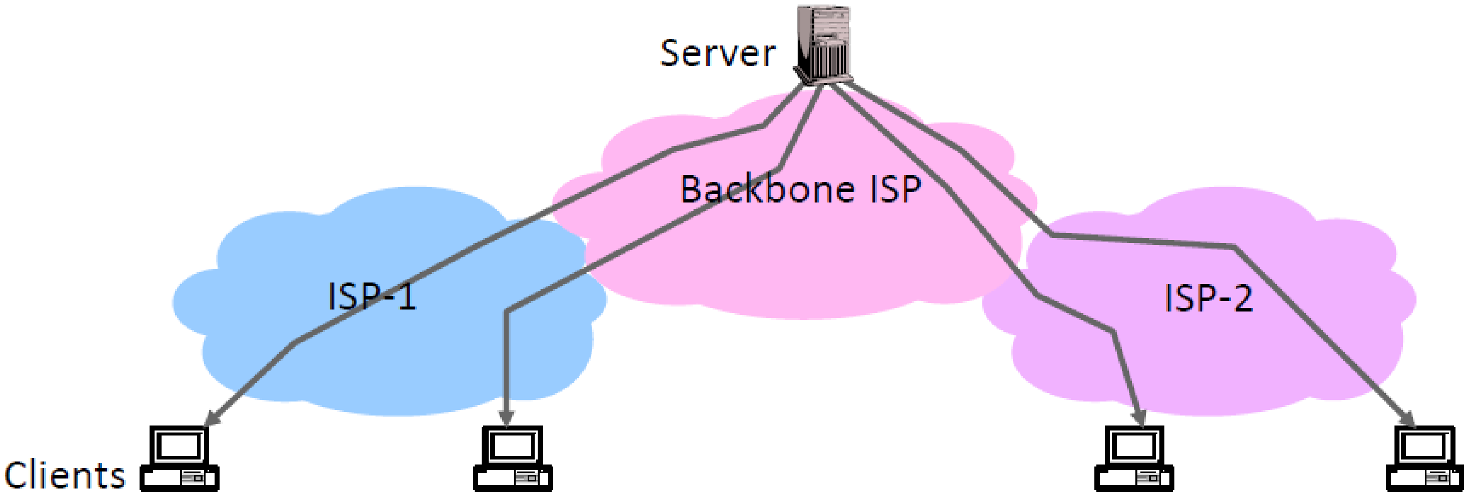
Multimedia Internet

Content Delivery Networks

Il documento è adattato da materiale cortesemente messo a disposizione dal Prof. Vittorio Trecordi

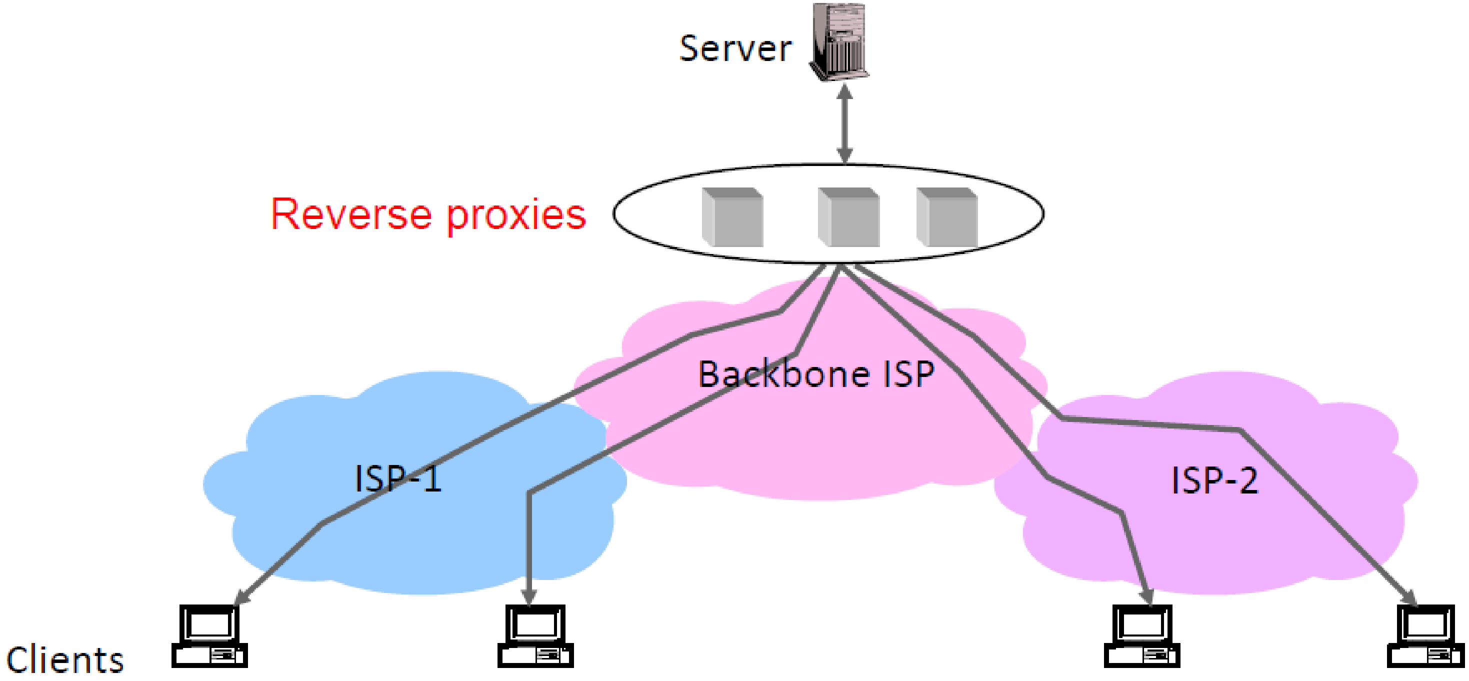
New needs and new strategies

- The spread of content with the Web architecture has had a remarkable development thanks to its simple and highly effective paradigm
  - Thanks to the Web architecture it is possible to transfer heterogeneous content: text, audio, video, images, etc.
- Historically, web content has been made available through *centralized servers* (origin servers) in the backbone network
  - Many clients that access the same information generate a load on such server and on the network that can be avoided or scaled down
- New strategies for the efficient dissemination of content via Web have been defined for:
  - Coping with the increase in network traffic
  - Ensure acceptable latencies
  - Use resources efficiently



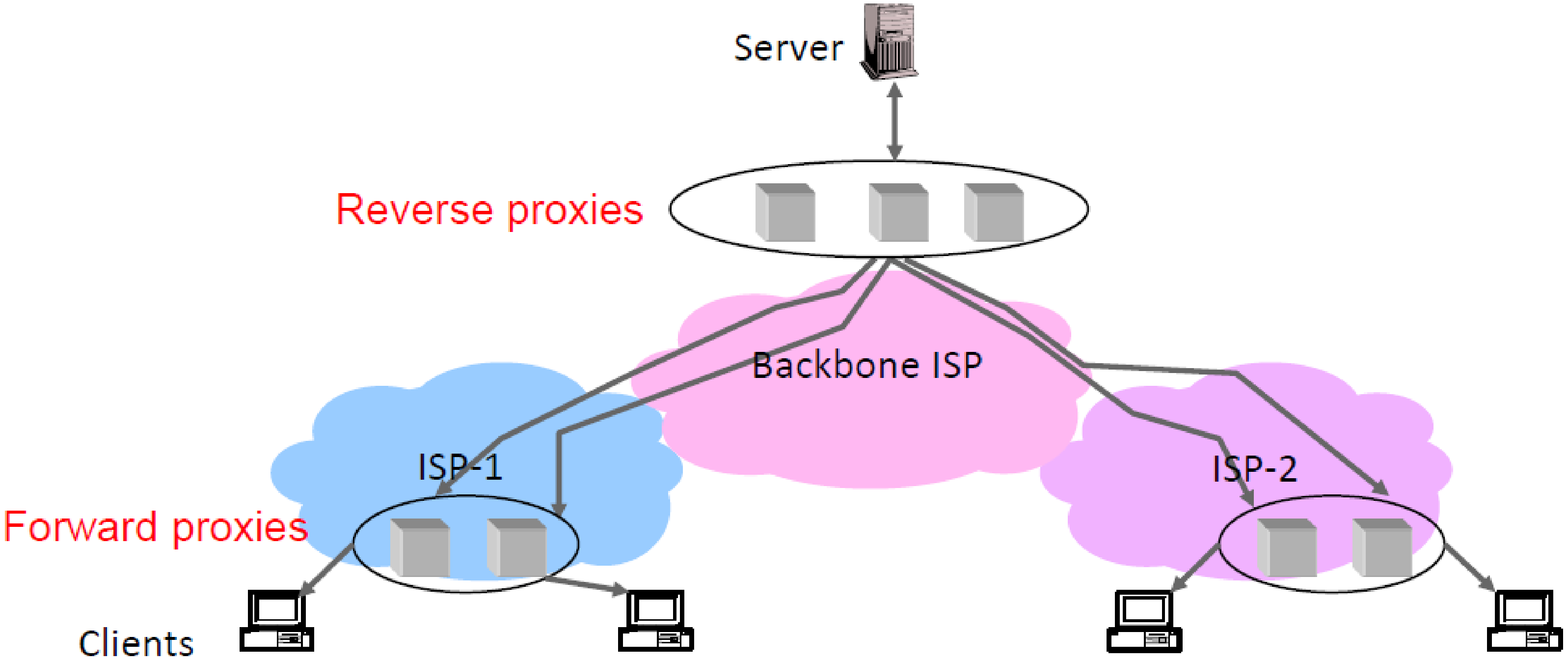
Proxy utilization

- The introduction of systems that replicate content (caching systems) and stand between clients and origin servers (called Proxies or Application Proxies) is a practice that tends to improve the content distribution system
- One possibility is to use reverse proxies as front-ends of the server by content providers to reduce the server load for spreading static content



Proxy utilization

- Forward proxies are intermediate systems with content replication functions, positioned close to clients and deployed in the corporate network or by the ISP in its network
- This location helps reduce access latency and reduce as well the network load for popular content
- We will call both forward and reverse proxies with the generic term *"cache"*



Content replica

- Content replication is effective if
  - The contents are consistent and valid
  - The contents are required by a plurality of users (spatial and temporal locality)
- Two different ways of caching
  - Content can be replicated by the cache opportunistically when requested by a user (pull system)
  - Content replicas can be systematically scheduled (push system)
    - The copy can be generated through an explicit dissemination system that propagates copies of the contents that are likely to benefit from caching (popular content)
- Two different types of access to the cache
  - In Transparent Caching, the caches are transparent with respect to the clients (they intercept the traffic in transit)
  - In Explicit Caching clients are configured to point to an intermediate system that performs caching

The caching principle

- The cache storage capacity that can be placed near users is usually low
  - We aim to replicate only the most commonly accessed content close to users
  - It is important to define a strategy to determine commonly used content (popular content)
- Accesses are often related to each other and are said to have *locality*
  - Temporal locality: the information that is consulted at a given moment will be consulted again in a short time
  - Spatial locality: the information consulted from one given point in space will probably be consulted by adjacent points

Consistency

- Consistency: ensures that replicas are consistent and aligned
- Different degrees of consistency
  - Strong: the delivery of inconsistent replicas is avoided
  - Weak: inconsistent replicas are delivered with low probability
- There are three mechanisms for controlling cached content replicas
  - *Invalidation*: depends on the expected expiry time, defined by the origin server. A replica is invalidated after this time has expired
  - *Freshness*: ensures that a cached replica can be considered "fresh", that is, not obsolete
  - *Validation*: permits to check if a cache replica is still valid, even after the expiry of the expected expiry time

Cooperative Caching

- When the cache does not contain a specific content (*cache miss*), instead of asking the origin server for it, it can request it from other caches
- Two types of cooperative caching
  - Flat: all caches are on the same level
  - Hierarchical: the caches are structured in hierarchical levels and the diffusion of the contents is organized in a tree

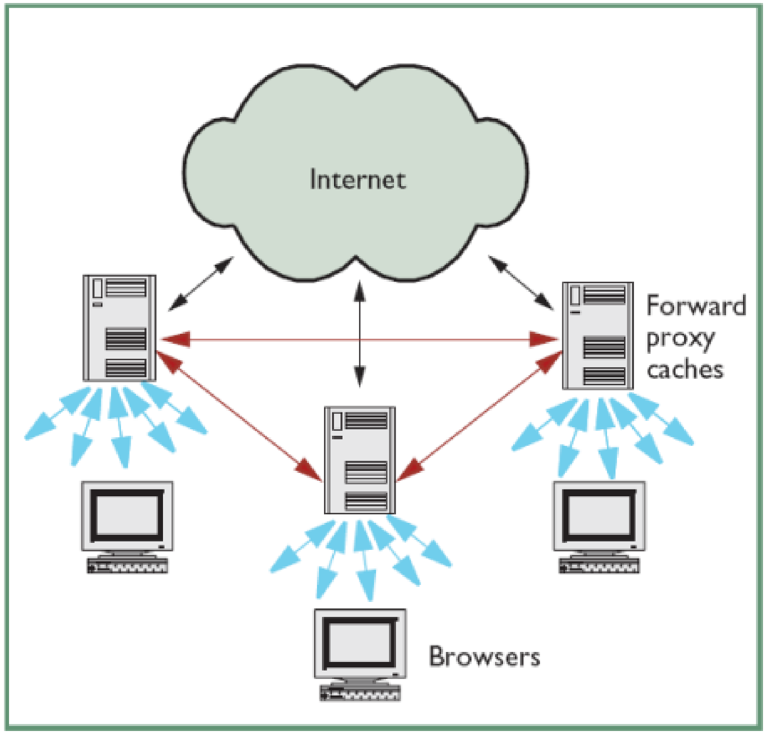


Figure 5. Cooperative caching. Caches communicate with peers before making requests over the Web.

From B. Davison: "A Web caching primer", IEEE Internet Computing, 5(4):38-45, Jul.-Aug. 2001

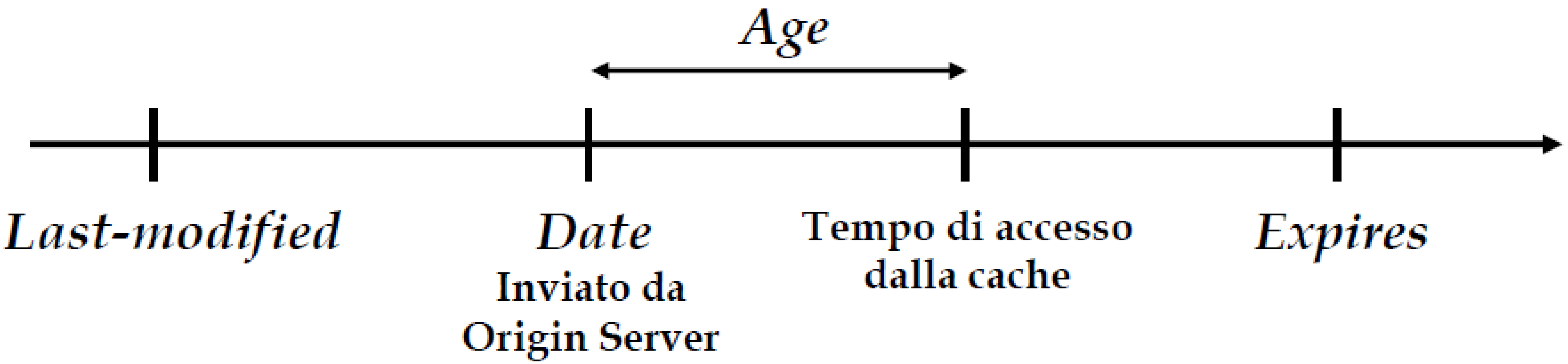
HTML and HTTP directives

- HTML and HTTP directives allow clients and servers to set how content is cached
- HTML directives
  - Eg <meta http-equiv = "pragma" content = "no-cache">
  - Easy to check for web page authors
  - Limited to HTML objects
- HTTP directives
  - Eg Cache-control :, Expires :, If-Modified-Since:
  - HTTP header fields
  - Easier to manage by caches
  - They are more common

Imperative HTTP directives

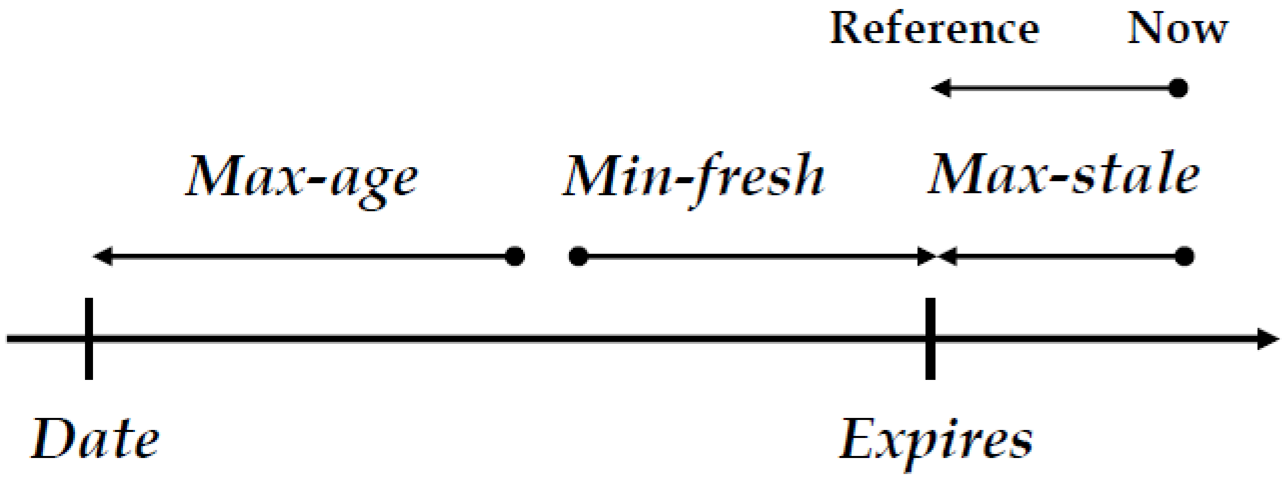
- They have precedence over other cache checks/controls
- In requests and responses
  - *cache-control: no-store*: prevents caching of objects
  - *cache-control: no-transform*: prevents the cache from performing data transformations
- Only in requests
  - *cache-control: only-if-cached*: requires the exclusive use of cached content
    - If the content is not present in the cache, the cache responds with a gateway timeout

Directives on the content's lifecycle



- *Last-modified*: istant in which the content has been modified by the origin server
  - The origin server and the caches must refer to a common clock
  - It does not differentiate between minor and major changes
- *Date*: Indicates the instant in which the object has been sent by the origin server to the cache
- *Expires*: server's estimate of the time in which the copies must be substituted
- *Age*: time passed by the object in tha cache
  - *Date + Age < Expires*

Directives related to the content's age



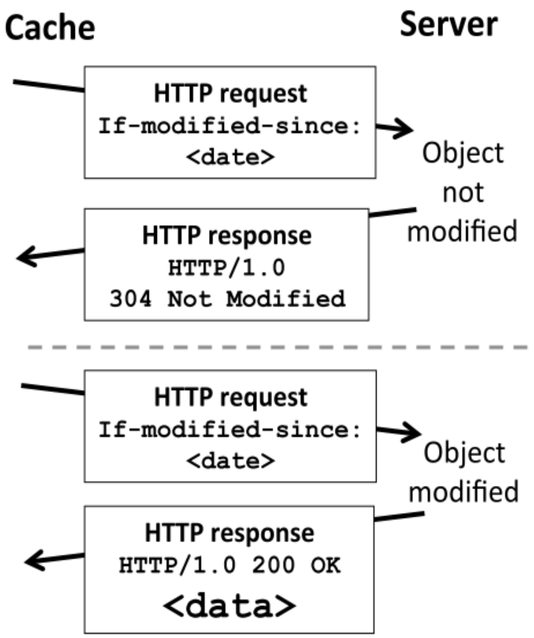
- The clients can request contents that have a given need in terms of freshness/age
  - *Max-age*: the client may not want information after a given obsolence time (referred to a *Date*)
  - *Min-fresh*: the client makes sure the content is sufficiently distant from the state of *Expiry*
  - *Max-stale*: the client could accept a slightly obsolete content (by default, caches do not serve obsolete contents)

Expiry predicted by the server

- The origin server limits the lifetime of the content it sends to the caches by means of the *Expires*
- However, the expiry time is a prediction of the server and can be wrong, for this reason it is necessary to resort to *validation* once the expiry time has been reached.
- Note: if there is a guarantee that the origin server does not modify an object before its expiry time (Expires), the client can be sure of the strong consistency

Validation by the client

- The client looks for a valid replica
- *Validation* is the verification carried out to understand if, once the expiry time has expired, a copy is still usable
- If the expiry time has been reached for the requested content
  1. The cache sends a GET request with
     - *if-modified-since*: cached copy date
     - *if-none-match*: Etag of the cached copy
       - » Etag (entity tag): string that distinguishes different versions of a content
  2. The server replies with one of the following
     - The content (if it has been modified)
     - *HTTP/1.0* Response code *304, «Not Modified»*



Validation Example

```
GET /img/ietf.png HTTP/1.1
Host: irtf.org
Referer: http://irtf.org/
If-Modified-Since: Fri, 06 May 2011 10:01:43 GMT
If-None-Match: "aa0b06-754-4a29893aa8fc0"
```

HTTP GET Request

```
HTTP/1.1 304 Not Modified
Date: Tue, 24 May 2011 05:21:29 GMT
ETag: "aa0b06-754-4a29893aa8fc0“
Expires: Tue, 31 May 2011 05:21:29 GMT
```

HTTP Response

Heterogeneity of contents

- There are heterogeneous types of content
  - *Static content*: relatively stable over time (e.g., HTML, images, archives)
  - *Volatile content*: modified frequently, periodically or by current events (news, sporting events, stock exchanges)
  - *Dynamic content*: dynamically created based on the client's request (search engines, e-commerce)
- *Multimedia content*, i.e. audio/video content typically larger than other types of content (video clips, mp3, Flash animations), is usually *static*

Cacheable content

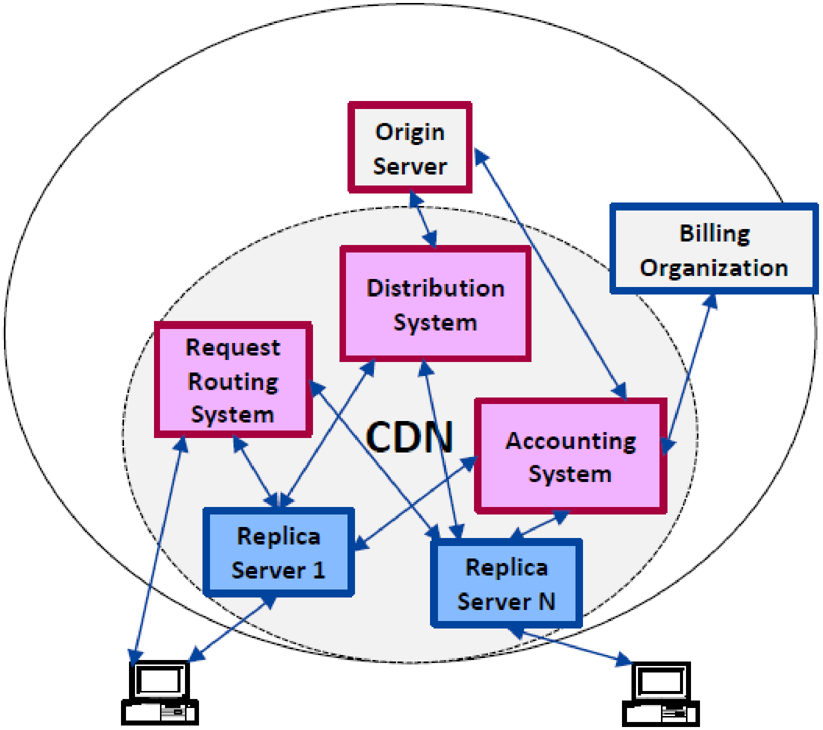
- A significant portion of the contents (> 50%) are "uncacheable"
- The main sources of uncacheability are
  - Volatility: they change frequently over time and it would be expensive to align cached copies
  - Dynamism: Caches are not usually designed to dynamically generate content
  - SSL: encrypted data cannot be processed in the cache
  - Advertising/Analytics: for example, the owner of an advertising banner wants to measure the number of clicks
- However, most volatile or dynamic content has a modest size, while static and multimedia content is voluminous

What is a Content Delivery Network (CDN)

- A CDN is an infrastructure created to effectively distribute the contents of the most popular web servers to Internet users
- CDNs are based on the scheduled and intelligent distribution of content replicas of the main *Content Provider* server (origin server) to a multiplicity of servers arranged on the network by a *CDN Provider*
    - The CDN service is offered by CDN Providers to Content Providers who have popular content requiring a wide and widespread diffusion
- The CDN service aims to improve performance
    - Reduce the latency of accessing content
    - Reduction of the band occupied on the network

Components of the CDN architecture

- *Content delivery* components
  - Origin server and server replica set (cache)
- *Content distribution* component
  - Replicates the content of the original server to the Replica servers and maintains consistency
- *Request routing* component
  - Direct user requests to a server (origin or replica)
  - Interacts with the distribution component to maintain an updated copy of the content
- *Accounting* component
  - Maintains user access logs
  - Performs traffic analysis and allows the Content Provider to charge



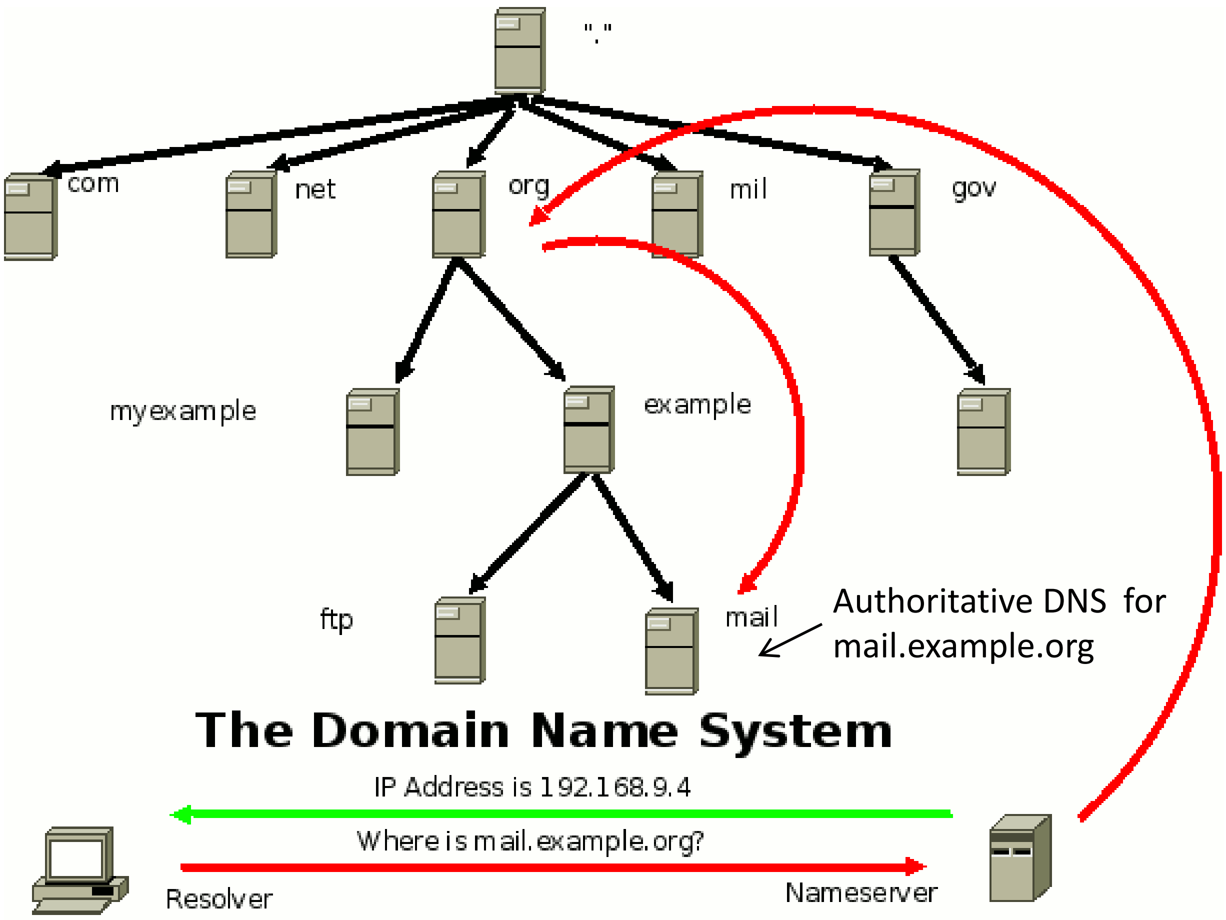
Placement of replica servers

- Optimization problem
- Variation of the k-medians problem (which is NP-hard, but good heuristics and approximate algorithms exist):
  - Given aset of points (users) and *k* (k = number of replica server)
  - Find *k* centers such that the determined clusters (distance from the center/server) are minimized (e.g.: average distance, latency etc.)

Request routing component

- How can we direct a client's request to a specific CDN cache server?
- Domain Name Server (DNS) system is used
- Two mechanisms are used
  - *DNS redirection*
  - *URL rewriting*

Domain Name Server (DNS)



Authoritative DNS for mail.example.org

DNS redirection

- The website's authoritative DNS can
  - Delegate the resolution of the hostname to an IP address to a name server controlled by the CDN
  - Directly resolve an address to a CDN cache server (if the CDN can directly manage authoritative DNS)
- In both cases
  - The choice of a specific CDN cache server is done with the translation from hostname to IP address
  - The choice is made by the internal DNS system of the CDN

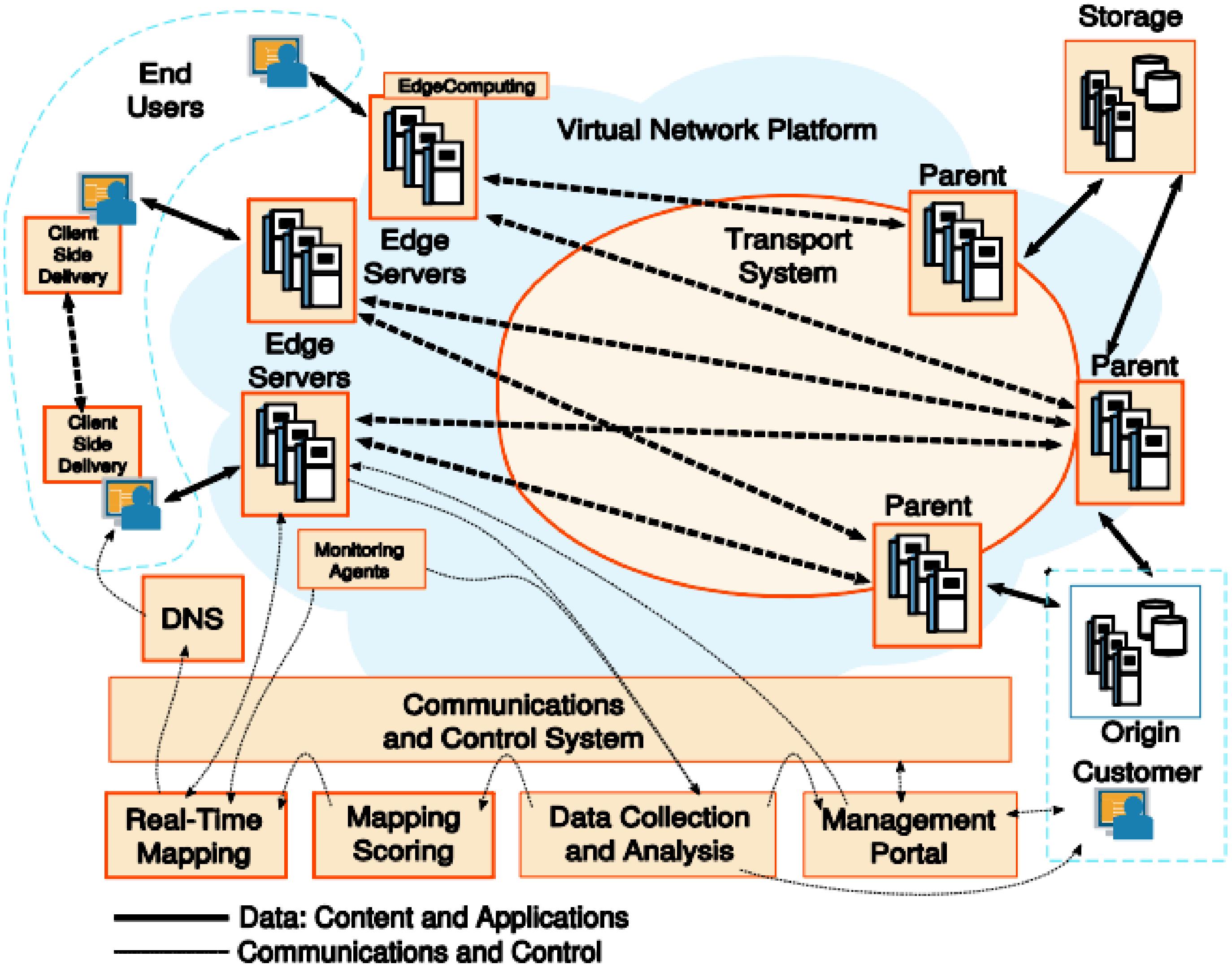
URL rewriting

- The Content Provider rewrites the URLs present in the HTML page in order to make it appear that the embedded objects are located on a cache server
- So there will be need for a specific resolution of the hostname of the various embedded objects, whose authoritative name server is under the control of the CDN
  - In resolving the new hostname, the DNS of the CDN will direct client requests for embedded objects to a cache server of the CDN
- It is possible to use more than one cache server for the embedded objects of a page

Akamai

- Akamai is an American company that deals with Internet services
  - It was born in 1998 in Massachussets
  - Among other things, it is the most important CDN Provider in the world
- It has the largest CDN in the world
  - Numerous Content Providers use the Akamai CDN for the dissemination of their content
  - Among the various companies that have a commercial partnership with Akamai we can mention Apple, Facebook and Twitter

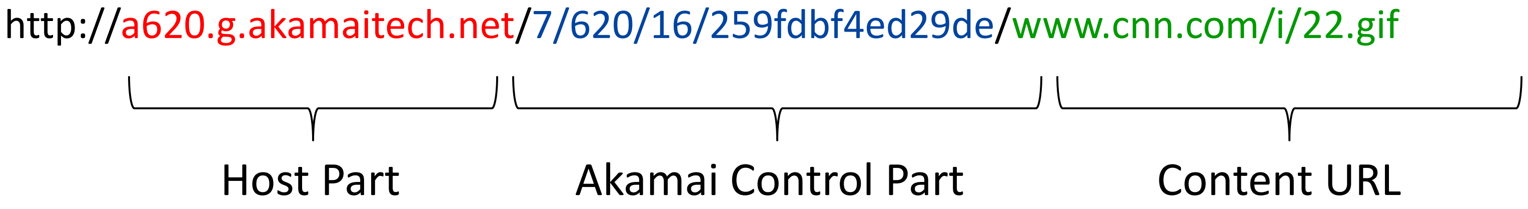
Akamai Architecture



Akamai
Indirect Routing and URL rewriting

- A website that wants to have part of its content distributed by Akamai must rename the URLs related to them with a specific prefix
- The resolution of the hostname to an IP address of an Akamai cache server is performed by Akamai's DNS
- The chosen cache server is "close" to the client and should not be overloaded
  - Akamai performs measurements and tests to obtain a map of the Internet
  - In this way, the DNS can establish the optimal cache server from which to obtain a specific content for a specific user

Akamai
ARL: Akamai Resource Locator



http://a620.g.akamaitech.net/7/620/16/259fdbf4ed29de/www.cnn.com/i/22.gif

Host Part

Akamai Control Part

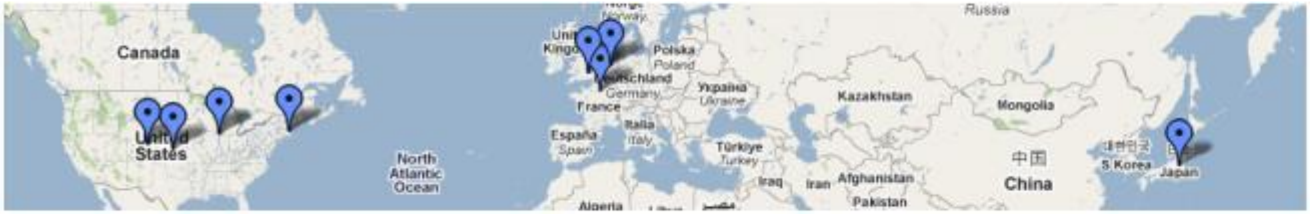
Content URL

- The content provider selects the content that will be hosted by Akamai
- Akamai provides a tool that transforms the URL (www.cnn.com/i/22.gif) into the ARL shown above
  - The ARL host part is sent as a response by the authoritative DNS of the content to the client's Nameserver
- In this way, the client accesses Akamai's cache servers (a620.g.akamaitech.net/) and not the origin server
- If the Akamai server does not have the cached content, it is requested from the origin server

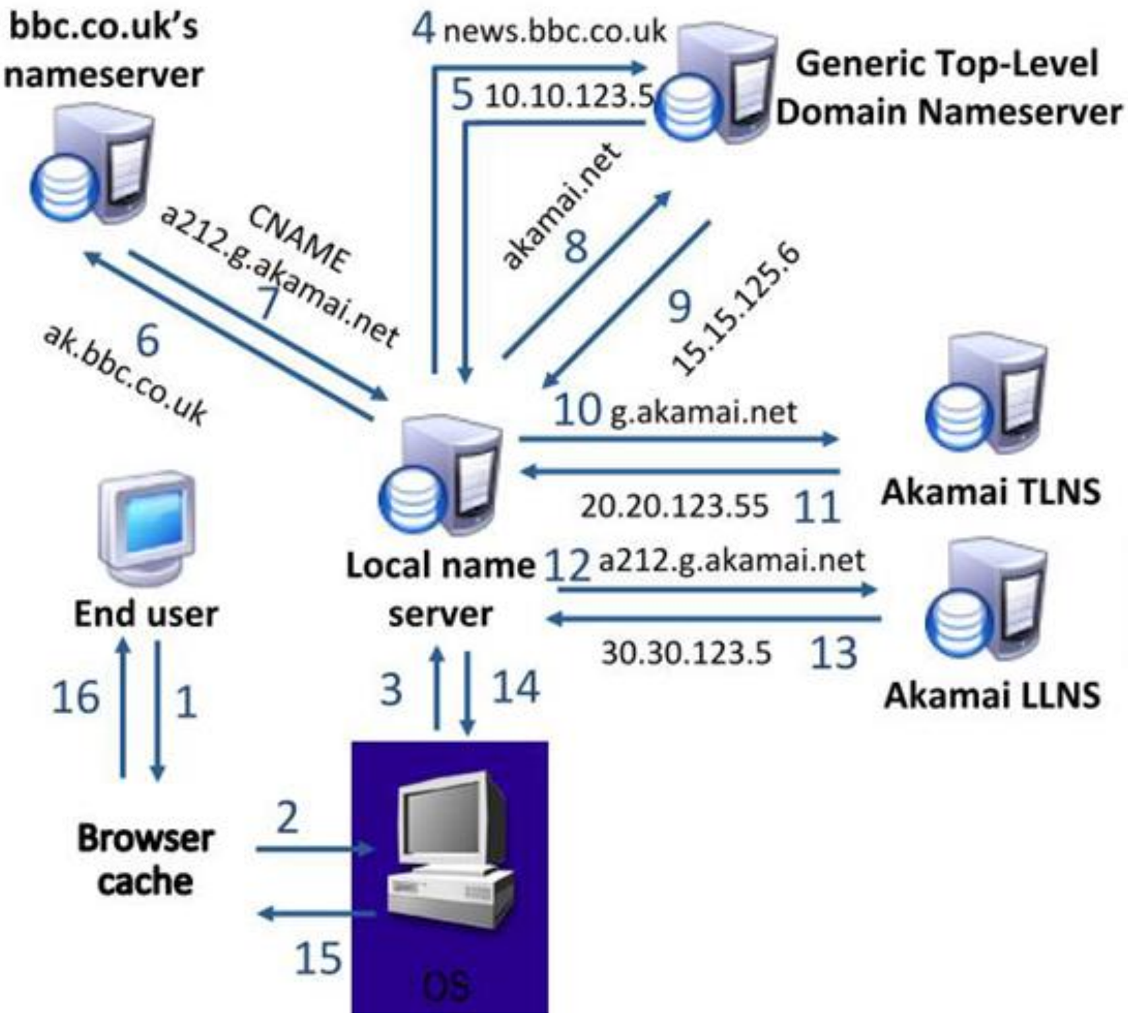
Akamai
Two-level DNS Redirection

- Akamai provides two levels of DNS redirection

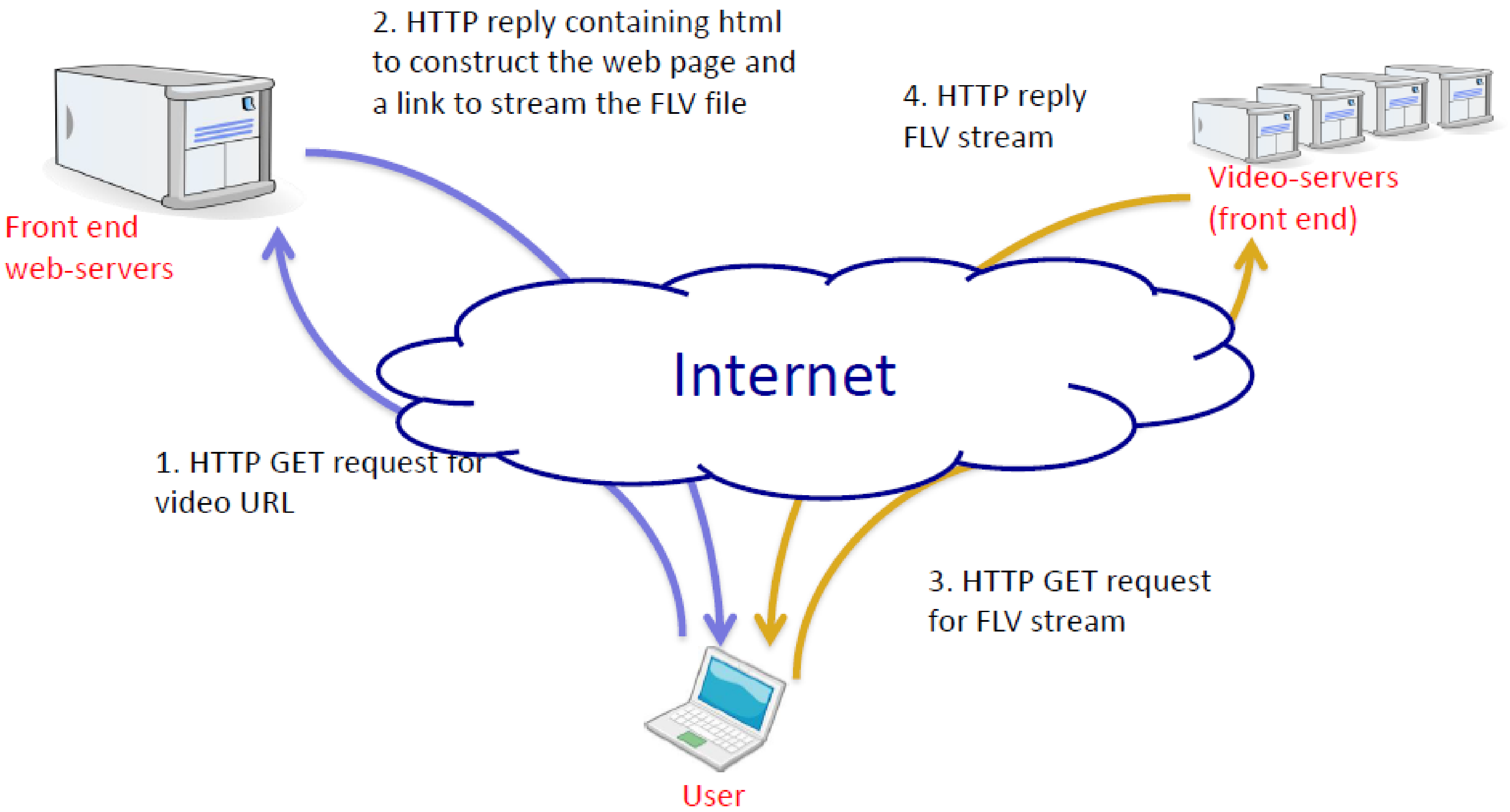
1. Akamai Top-Level Name Server (TLNS)
    - Localized: 4 in the US, 3 in the EU and 1 in Asia
    - The TLNS respond with an LLNS, taken from a set of 8 LLNS close to the user
2. Akamai Low-Level Name Server (LLNS)
    - They point to the Akamai Edge Servers that deliver the content
    - They perform traffic balancing



Akamai
Two-level DNS Redirection



YouTube Architecture Basic mode



2. HTTP reply containing html to construct the web page and a link to stream the FLV file

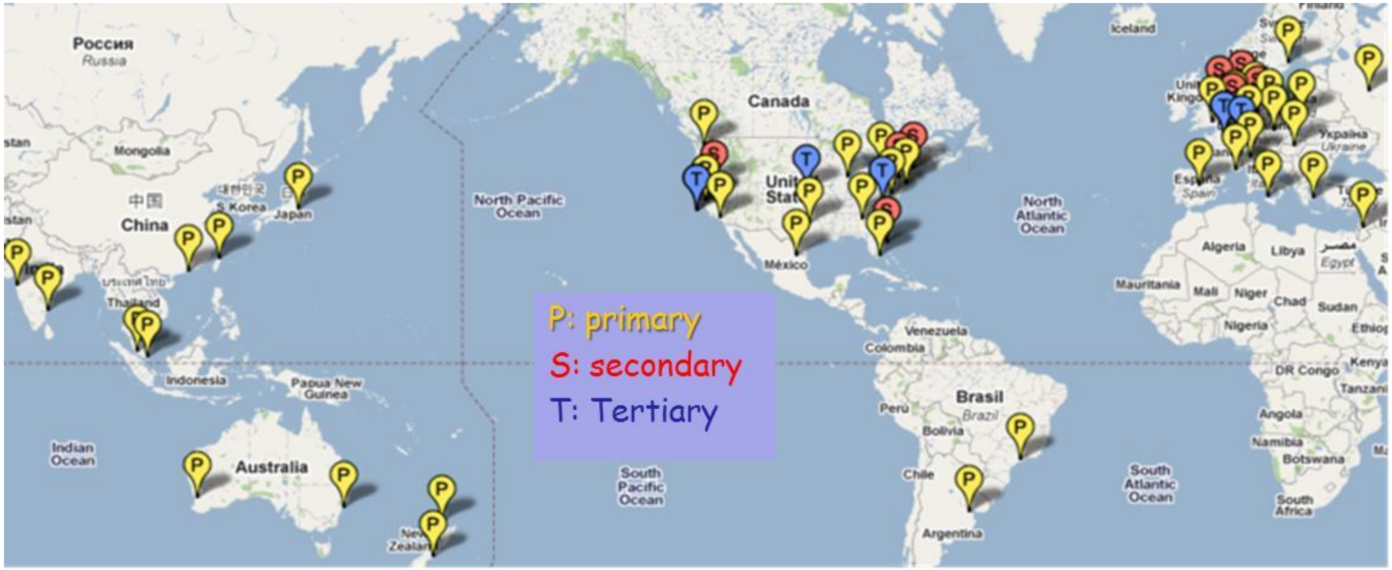
4. HTTP reply FLV stream

1. HTTP GET request for video URL

3. HTTP GET request for FLV stream

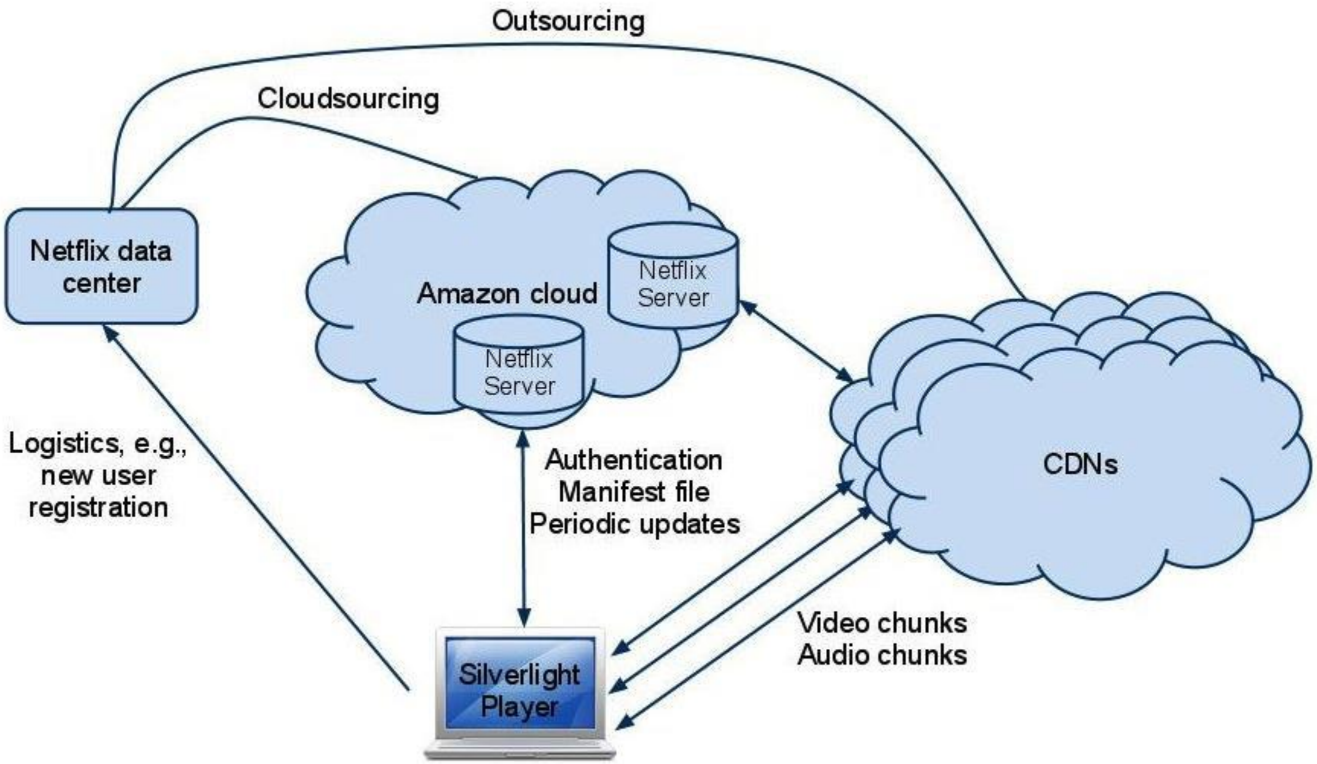
Youtube architecture New mode

- Cache structure at three levels: primary, secondary and tertiary
- Load balancing of servers based on location information
- About 50 caches
  - 40 primary (about 10 at ISPs)
  - 8 secondary
  - 5 tertiary



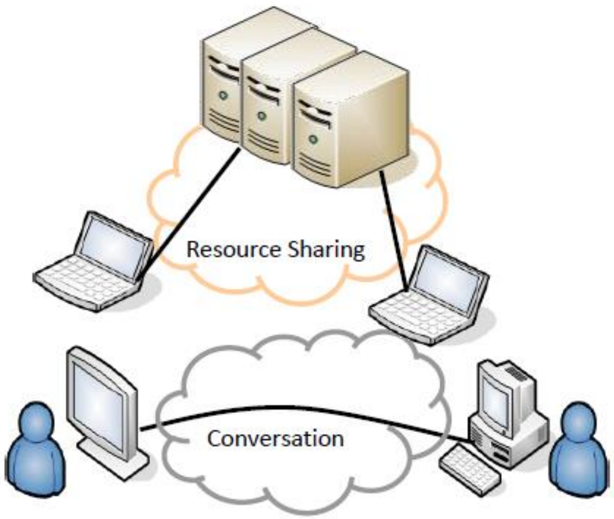
Netflix Architectures

- Netflix has its own Datacenter for management functions (e.g. registration and pricing), for the rest it is based on Amazon's AWS (Amazon Web Service) Cloud for video streaming servers
- It uses three CDNs for delivering video content (scalable encoding)



A novel paradigm
Information Centric Networking

- Progressively we assist to a paradigm shift in the main utilization of the Internet
  - From a system of sharing resources and conversations between hosts to a system of content distribution in various forms and with increasing volumes
- The awareness of these trends prompted the study of a new paradigm for the Internet that shifts the focus on the treatment of content: **Information Centric Networking**
- It is an alternative paradigm to IP, based on content rather than IP addresses



20 exabytes/month added in 2011