

17. 내용 협상과 트랜스코딩

HTTP Definitive Guide

아키텍처

내용협상이란

- 하나의 URL이 여러 리소스로 대응할 필요가 있을 경우 HTTP는 클라이언트와 서버간 내용 협상을 통해 클라이언트에 가장 적절한 결과를 반환함
- variant: 하나의 URL에 대해 존재하는 여러 사본

내용 협상 기법

1. 클라이언트 주도
2. 서버 주도
3. 투쟁

17.2 클라이언트 주도 협상

1. 클라이언트 주도 협상

- ▶ 클라이언트 요청시 가능한 페이지 목록을 응답으로 돌려주고 클라이언트가 원하는 페이지를 선택하게 하는 것

2. 클라이언트 주도 협상 장단점

- ▶ 장점: 서버 입장에서 구현이 쉬움
- ▶ 단점: 각 페이지에 대해 두 번의 요청이 필요함

17.3 서버 주도 협상

- 서버에서 클라이언트에 반환해줄 페이지를 결정하는 것.
- 클라이언트가 어떤 페이지를 보고자 하는지 정보를 header에 담아 서버에 전달
- 내용 협상 header 종류
 - 내용 협상 header 사용
 - user-agent header 사용

17.3.1 내용 협상 헤더

- 내용 협상 헤더를 통해 클라이언트가 자신의 선호 정보를 보냄

Table 17-2. Accept headers

Header	Description
Accept	Used to tell the server what media types are okay to send
Accept-Language	Used to tell the server what languages are okay to send
Accept-Charset	Used to tell the server what charsets are okay to send
Accept-Encoding	Used to tell the server what encodings are okay to send

- 서버는 클라이언트의 Accept관련 헤더들을 적절한 엔터티 헤더들과 매칭!

- entity header는 본문에 대한 속성

Table 17-3. Accept and matching document headers

Accept header	Entity header
Accept	Content-Type
Accept-Language	Content-Language
Accept-Charset	Content-Type
Accept-Encoding	Content-Encoding

- HTTP는 Stateless 프로토콜 이므로 클라이언트는 모든 요청에 선호 정보를 보내야 함
- 클라이언트 주도 모델보다 커뮤니케이션 대기시간 감소

17.3.2 내용 협상 헤더의 품질값

- 내용 협상 header의 품질값
 - 클라이언트가 각 선호 카테고리마다 선호도를 표시
 - 가장 높은 선호도에 대한 리소스가 없을 경우 다음 선호도의 것을 클라이언트에 반환
 - ▶ Accept-Language: en;q=0.5, fr;q=0.0, nl;q=1.0, tr;q=0.0

17.3.3 그 외의 헤더들에 의해 결정

- user-agent header 이용
 - 웹 브라우저, device 정보들을 이용해 클라이언트에게 최선의 결과를 반환
 - ex: 브라우저 버전을 확인해 자바 스크립트를 지원하지 않음 경우 자바 스크립트를 포함하지 않은 페이지를 반환
- 9집스 메커니즘은 없음

17.3.4 아파치의 내용 협상(type-map)

- variant를 갖는 웹사이트의 각 URI를 위한 type-map파일을 작성
- type-map 파일은 모든 variant와 그들 각각에 대응하는 내용 협상을 나열
- type-map 파일을 위한 파일 집합자를 명시한 핸들러 추가

▶ Addhandler type-map .var

▶ 파일 내용

```
URI: joes-hardware.html
```

```
URI: joes-hardware.en.html
```

```
Content-type: text/html
```

```
Content-language: en
```

```
URI: joes-hardware.fr.de.html
```

```
Content-type: text/html; charset=iso-8859-2
```

```
Content-language: fr, de
```

17.3.4 아파치의 내용 협상(MultiView)

- 브라우저가 joes-hardware라는 이름의 리소스를 요청했다면, 서버는 "joes-hardware"가 있는 모든 파일을 살펴보고 type-map을 생성
- MultiViews를 사용하려면 access.conf 파일의 적절한 (<Directory>, <Location>, 혹은 <Files>) 에 Options 지시어를 이용해 웹 사이트를 포함한 디렉토리에 MultiViews를 반드시 커야함

프록싱 협상

- client와 server사이에 proxy서버를 구성
- proxy는 여러 사본을 저장해 서버대신 클라이언트와 협상 후 캐싱된 정보를 전송함
- 서버가 var header를 이용해 proxy에 클라이언트와 협상시 어떤 header를 참조해야 하는지 알려줌
- proxy서버에서 클라이언트의 요구사항을 파악해 서버에 요청
- 서버의 부하를 덜고 caching된 데이터를 클라이언트

특수경 협상

- 캐싱하고 있던 것과 다른 요청이 들어올 경우 cache는 서버에 요청해 다른 버전의 문서를 갖게됨. 이를 alternate라 부름

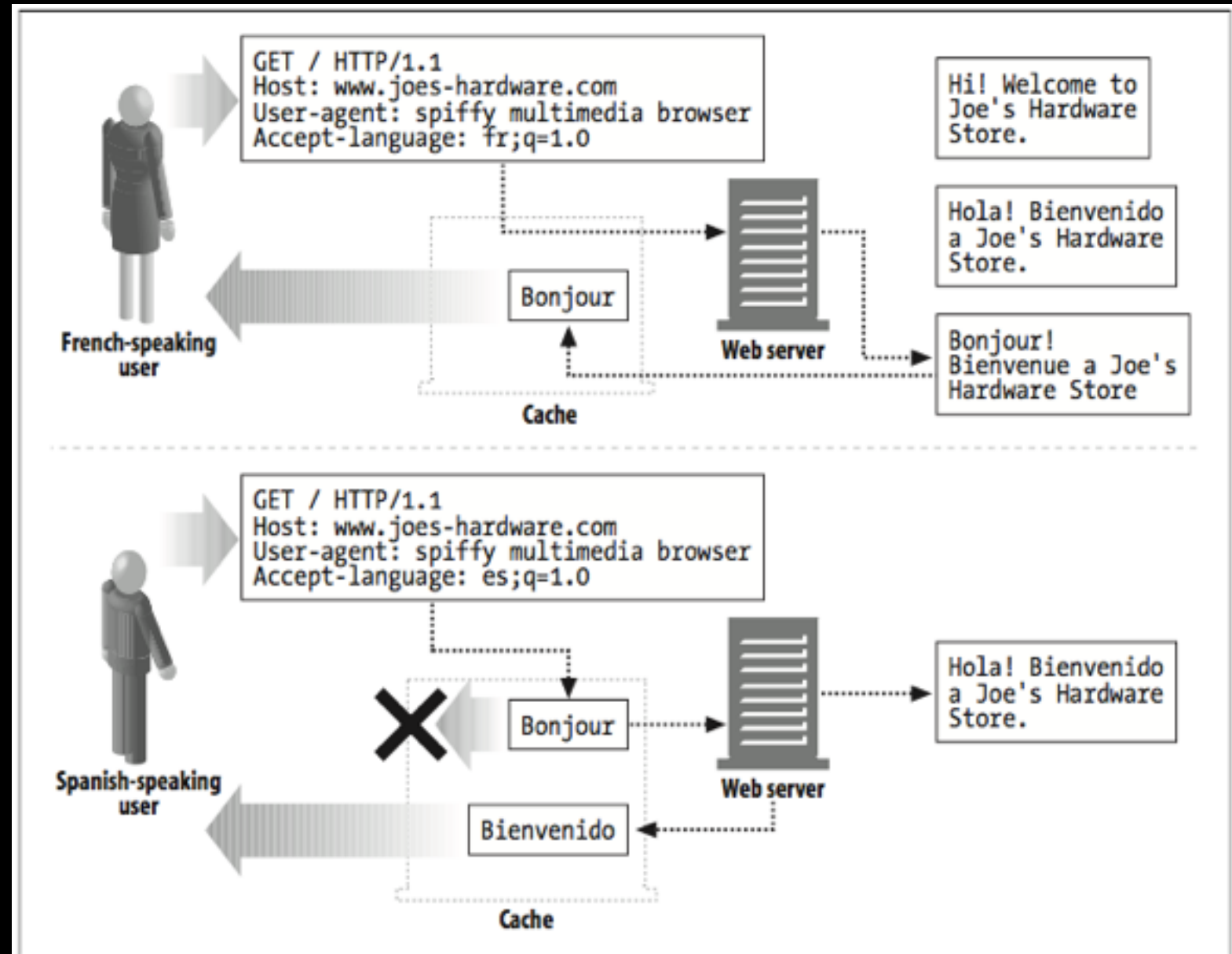


Figure 17-1. Caches use content-negotiation headers to send back correct responses to clients

트랜스코딩

- 서버가 갖고있지 않는 문서를 클라이언트가 요청할 경우 기존의 문서를 클라이언트에 맞게 변환하는 것

트랜스코딩 세 종류

- 포맷변환

- 내용 협상 헤더에 의해 주도(ex: User-Agent header)

- 정보합성

- 문서 개요 생성, 요약등

- 콘텐츠 주입

- 자동 광고 생성, 통계 수집을 위한 사용자 추적 시스템등

Table 17-4. Hypothetical transcodings

Before	After
HTML document	WML document
High-resolution image	Low-resolution image
Image in 64K colors	Black-and-white image
Complex page with frames	Simple text page without frames or images
HTML page with Java applets	HTML page without Java applets
Page with ads	Page with ads removed

트랜스코딩

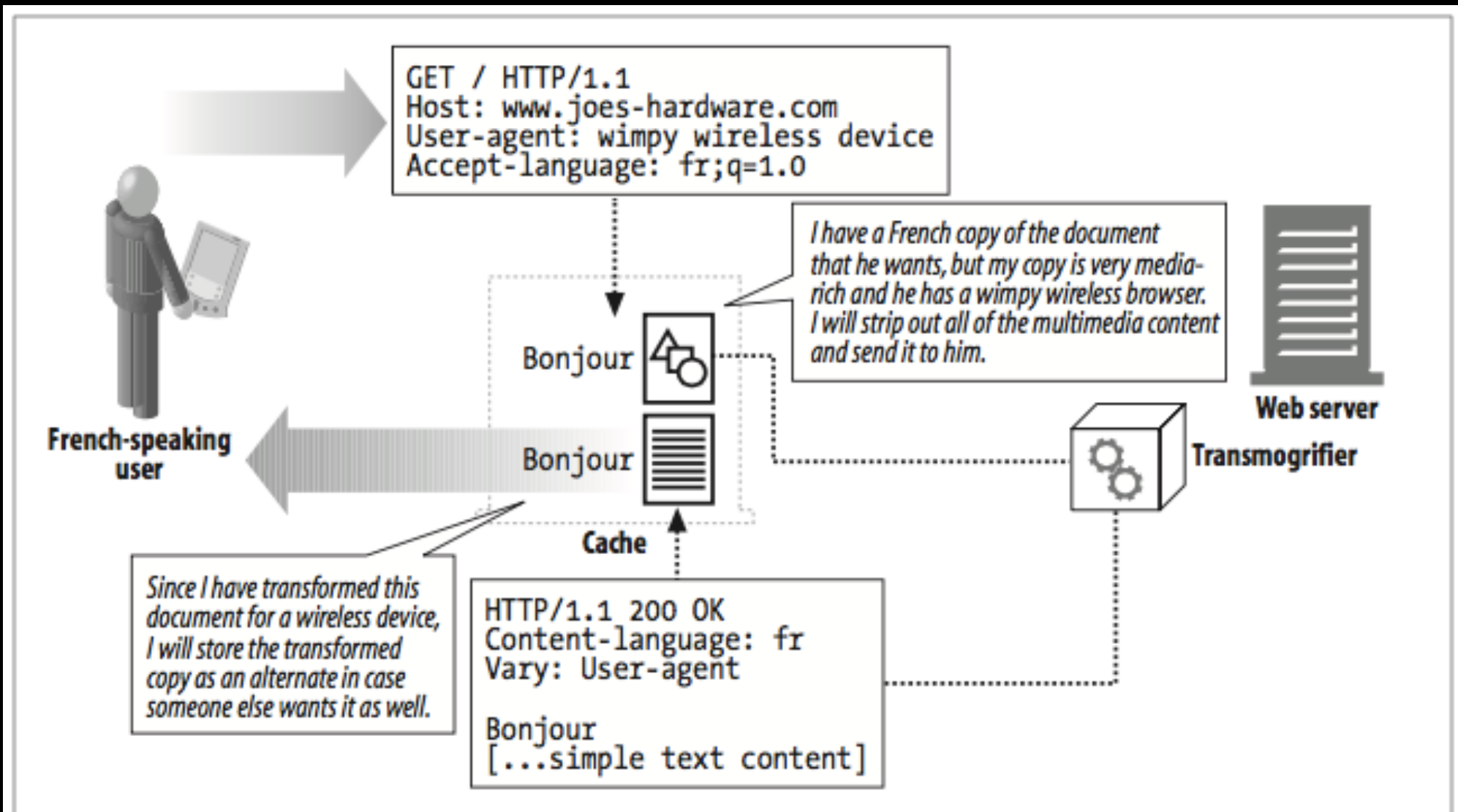


Figure 17-3. Content transformation or transcoding at a proxy cache

트랜스 코딩 vs. 정적으로 미리 생성해 놓기

- 트랜스코딩의 대안으로 여러가지 사본을 만드는것
 - 하나의 이미지에 대해 여러 버전의 사본을 만들경우 작은 변화에도 많은 수정이 필요함
 - 용량 증가
 - 프로그래밍이 복잡해짐