

WebDAV

Fabio Vitali
(basato sui lucidi di Jim Whitehead)



WebDAV

Distributed Authoring and Versioning on the
World Wide Web

**Un'architettura per la realizzazione di
strumenti interoperabili per l'authoring
distribuito sul web**

<http://www.webdav.org/>



Usare WebDAV

Dovete scrivere un documento da mettere disponibile sulla rete.

- ◆ Aprite il vostro Word Processor preferito e scrivete.
- ◆ Salvate una copia sul Web.

Decidete di includere un altro autore

- ◆ Attivate la possibilità di condivisione del documento
- ◆ Usando lo stesso Word Processor, continuate a scrivere sul documento
- ◆ L'altro autore può lavorare indipendentemente da voi sul documento.



WebDAV e HTTP

WebDAV è un'estensione di HTTP

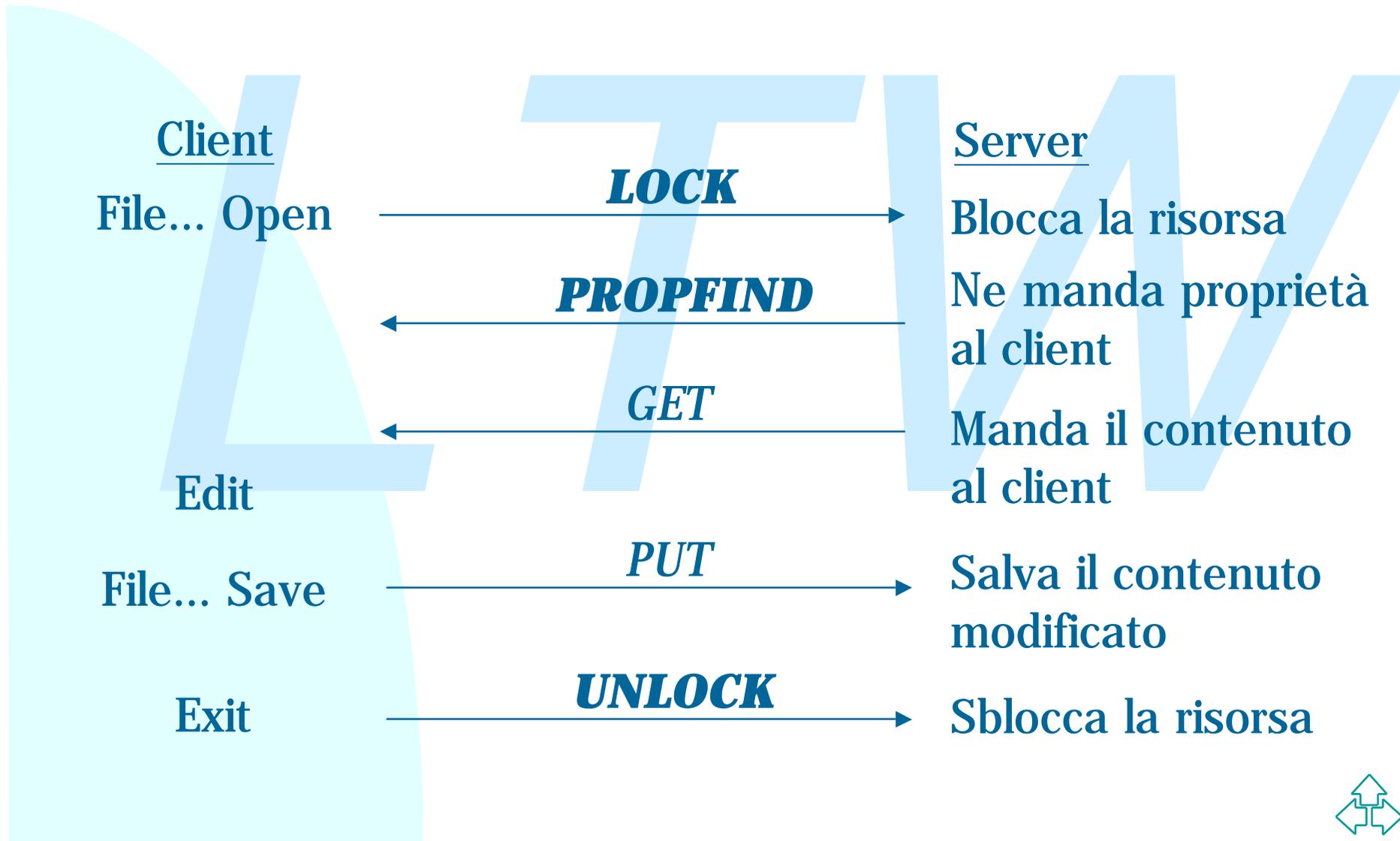
- ◆ Aggiunge proprietà e collezioni al modello HTTP

Fornisce comandi per

- ◆ Proprietà - list, add, remove
- ◆ Operazioni sui nomi delle risorse - move, copy
- ◆ Controllo di sovrascrittura - lock, unlock
- ◆ Collezioni - mkcol



Ciclo d'uso di WebDAV



Possibili usi di WebDAV

- Un protocollo per l'authoring collaborativo
- Un file system sul Web, con caratteristiche positive in caso di grandi latenze.
- Un meccanismo integrato per accedere a repositories
 - ◆ Document management system, configuration management system, file system, ecc.
 - ◆ Un protocollo per la gestione di team distribuiti nel software engineering e nel content management
- Sono tutte visioni lecite



Obiettivi di WebDAV (1)

Infrastruttura di collaborazione

- ◆ Attraverso il meccanismo di locking
- ◆ Permette l'authoring collaborativo remoto di pagine Web e di qualunque tipo di risorsa e ogni formato di dato.

Infrastruttura per la memorizzazione di metadata

- ◆ E' possibile creare, modificare, rimuovere e accedere a proprietà (coppie nome, valore arbitrarie)
- ◆ I valori delle proprietà sono frammenti XML

Infrastruttura per la gestione di spazi di nomi

- ◆ Copy e Move di risorse singole e di gerarchie di strutture.
- ◆ Creare e modificare nuove collezioni di risorse.



Obiettivi di WebDAV (2)

Infrastruttura per il Versioning

- ◆ check-out, check-in, storia del grafo delle versioni, commenti, accesso alle vecchie versioni, operazioni di configuration management.

Infrastruttura per l'Access Control

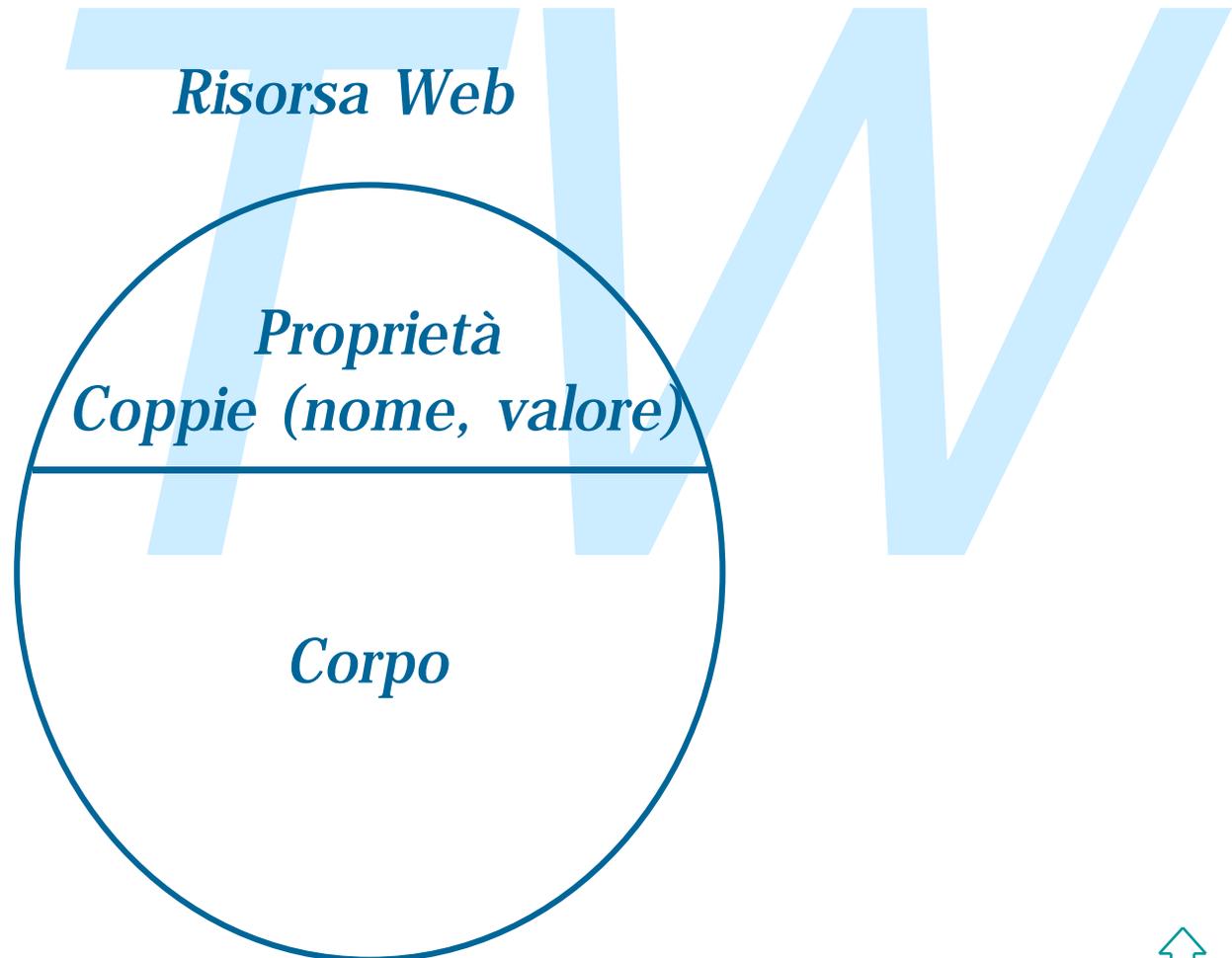
- ◆ La possibilità di controllare in remoto chi può leggere e scrivere una risorsa.
- ◆ La possibilità di specificare gruppi di collaborazione e gruppi di diritti.

Infrastruttura di ricerca

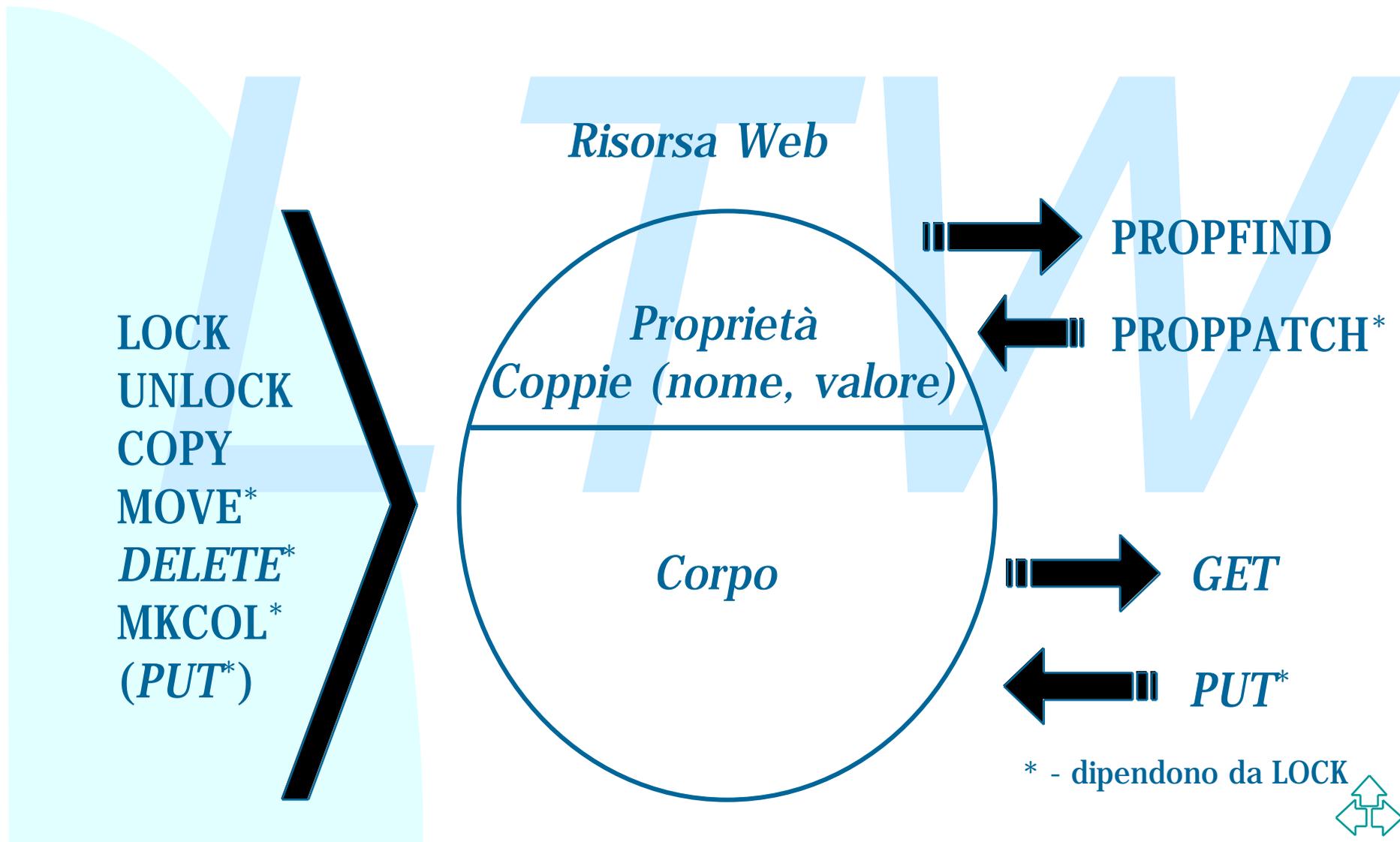
- ◆ Ricerca di risorse con una data proprietà, o un dato valore.
- ◆ Ricerca per sottostringhe all'interno di una risorsa o di una collezione di risorse.



WebDAV Object Model



Metodi WebDAV



Proprietà (1)

Le proprietà sono coppie (nome, valore).

Il nome è sempre un URI (universalità e generalità)

Le proprietà possono essere:

- ◆ Live: proprietà di cui è il server a mantenere coerenza obbligando sintassi e semantica. Ad esempio: data di ultima modifica.
- ◆ Non-live: proprietà in cui sintassi e semantica sono fornite dal client. Non esiste obbligo di coerenza. Ad esempio: autore del documento.

Vantaggi: sia il server che il client possono definire nello stesso modo delle proprietà, e ciascuno si gestisce e tiene aggiornate le proprie.



Proprietà (2)

I valori delle proprietà sono frammenti ben-formati in XML (senza DTD, ma qualificati).

- ◆ estensibili (grazie ai namespace), internazionali (grazie a Unicode), permettono valori strutturati, permettono l'inserimento senza fatica di strutture RDF.

WebDAV introduce due metodi HTTP per le proprietà:

- ◆ **PROPFIND**: recupera proprietà su una risorsa. Può recuperare tutte le proprietà, tutti i nomi di proprietà, tutti i nomi e valori specificati nella richiesta.
- ◆ **PROPPATCH**: crea, e rimuove proprietà da una risorsa. Le direttive di creazione o rimozione sono eseguite atomicamente.



PROPFIND: richiesta

```
PROPFIND /demo.txt HTTP/1.1
```

```
Host: dav.ics.uci.edu
```

```
Content-Type: text/xml; charset="utf-8"
```

```
Content-Length: 160
```

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<d:propfind xmlns:d="DAV:">
```

```
  <d:prop>
```

```
    <d:getetag/>
```

```
    <d:getcontenttype/>
```

```
  </d:prop>
```

```
</d:propfind>
```



PROPFIND: risposta

HTTP/1.1 207 Multi-Status

Server: DAV-demo-server/1.0

Date: Tue, 09 Feb 1999 00:52:55 GMT

Content-Type: text/xml; charset="utf-8"

Content-Length: 321

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<a:multistatus xmlns:a="DAV:">
```

```
<a:response>
```

```
<a:href>http://dav.ics.uci.edu/demo.txt</a:href>
```

```
<a:propstat>
```

```
<a:status>HTTP/1.1 200 OK</a:status>
```

```
<a:prop>
```

```
<a:getetag>"10598798c153be1:a82"</a:getetag>
```

```
<a:getcontenttype>text/plain</a:getcontenttype>
```

```
</a:prop>
```

```
</a:propstat>
```

```
</a:response>
```

```
</a:multistatus>
```



PROPPATCH: richiesta

```
PROPPATCH /webdav.html HTTP/1.1
Host: sandbox.xerox.com
Content-Type: text/xml; charset="utf-8"
Content-Length: 283

<?xml version="1.0" encoding="utf-8" ?>
<d:propertyupdate xmlns:d="DAV:">
  <d:set>
    <d:prop xmlns:j="http://mysite.com/myNS">
      <j:personal>
        <j:item>Proprieta' personale</j:item>
      </j:personal>
    </d:prop>
  </d:set>
</d:propertyupdate>
```



PROPPATCH: risposta

HTTP/1.1 207 Multi-Status

Date: Tue, 09 Feb 1999 01:36:43 GMT

Server: PyDAV 1.1 filestore 1.1

Content-Type: text/xml; charset="utf-8"

Content-Length: 317

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<a:multistatus xmlns:a="DAV:">
```

```
  <a:response>
```

```
    <a:href>/webdav.html</a:href>
```

```
    <a:propstat>
```

```
      <a:prop>
```

```
        <b:personal xmlns:b="http://mysite.com/myNS"/>
```

```
      </a:prop>
```

```
      <a:status>HTTP/1.1 200 OK</a:status>
```

```
    </a:propstat>
```

```
  </a:response>
```

```
</a:multistatus>
```



Collezioni

Una collezione è una lista di URI delle risorse che ne fanno parte.

Svolge il compito di raggruppamento che hanno le directory nei file system, a parte che:

- ◆ Una risorsa può avere più URI, e quindi appartenere a più collezioni diverse
- ◆ Non c'è nessuna implicazione sulle modalità di memorizzazione fisica delle risorse sul server.
- ◆ E' possibile definire collezioni ordinate, indipendentemente dal valore di qualunque proprietà.

Le risorse forniscono la possibilità di navigare gerarchicamente o altro tra le risorse.

- ◆ Permettono il supporto di finestre di dialogo tipo "File:Open" o "File:Save as" anche su server di rete.



Comandi sulle collezioni (1)

MKCOL

- ◆ Crea una nuova collezione vuota.

PROPFIND

- ◆ Elenca i membri di una collezione.
- ◆ L'intestazione Depth della richiesta permette di specificare quanti livelli mostrare:
 - ◆ Depth: 1 - La collezione e i suoi membri
 - ◆ Depth: Infinity - La collezione, i suoi membri, i loro membri ecc ricorsivamente.

PROPPATCH

- ◆ Permette di aggiungere o rimuovere membri da una collezione



Comandi sulle collezioni (2)

COPY

- ◆ Duplica risorse, proprietà e collezioni
- ◆ La copia avviene server-side (non c'è un giro inutile sul client)
- ◆ Copie per valore, eseguite ricorsivamente attraverso l'intestazione Depth
- ◆ Permette di controllare la sovrascrittura di proprietà o risorse già esistenti.

MOVE

- ◆ Definita come equivalente ad una COPY seguita da una DELETE, eseguiti atomicamente.
- ◆ Ha le stesse opzioni di COPY



Lock

Il lock definito da WebDAV è solo in scrittura.

- ◆ Un client che non possiede il lock non può eseguire comandi che modifichino lo stato della risorsa bloccata.
- ◆ In particolare impedisce i comandi: PUT, POST, DELETE, MKCOL, PROPPATCH, MOVE, LOCK, UNLOCK
- ◆ Al contrario GET e PROPFIND non sono impediti dal lock.

Il lock funziona solo su intere risorse

- ◆ E' l'unico modo per rendere il meccanismo indipendente dal content type.
- ◆ E' molto simile al meccanismo fornito dai file system, rendendo la transizione di un file system in file system distribuito particolarmente semplice.
- ◆ Le proprietà live possono cambiare durante la vita di un lock, le proprietà non-live debbono essere modificate dal proprietario del lock.



Lock (2)

I lock possono essere esclusivi o condivisi.

- ◆ Un lock esclusivo permette l'accesso solo al client che ha ottenuto il lock
- ◆ Un lock condiviso permette al proprietario del lock di specificare altri client che possono accedere alla risorsa.

I lock condivisi sono un bene e un male

Tuttavia i lock esclusivi sono troppo rigidi in condizioni di lavoro opportunistiche

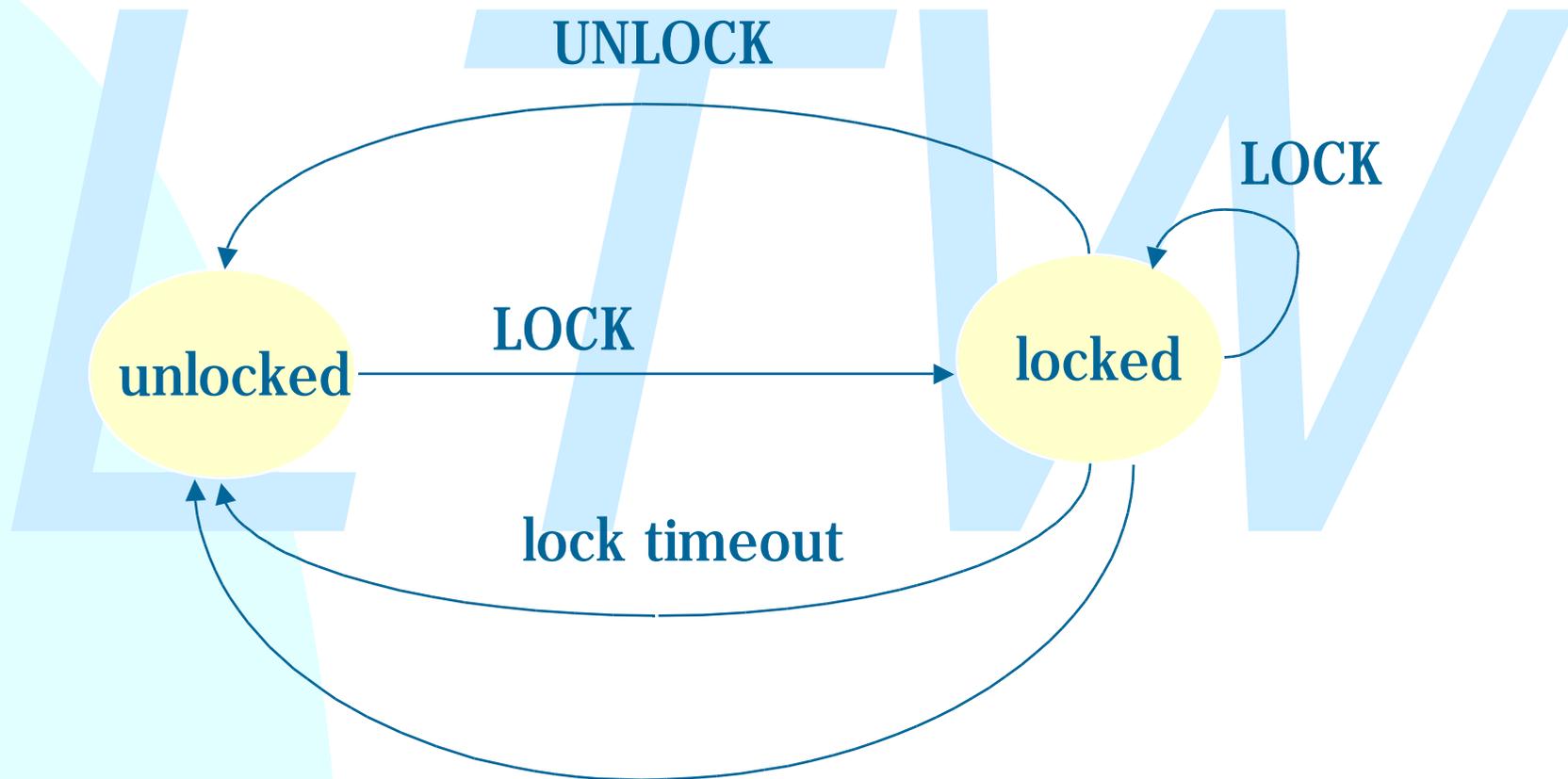
- ◆ La gente si dimentica di rilasciare un lock
- ◆ E' necessario richiedere l'intervento dell'amministratore per rimuovere un lock dimenticato.

I lock condivisi

- ◆ Permettono di continuare il lavoro anche se il proprietario del lock se ne è andato alle Seychelles.
- ◆ Richiedono però meccanismi *out-of-band* e in definitiva extra-protocollo per la negoziazione sull'accesso alla risorsa.



Ciclo di vita di un Lock



Rimozione del lock da parte dell'amministratore



LOCK: richiesta

```
LOCK /webdav.html HTTP/1.1
```

```
Host: sandbox.xerox.com
```

```
Timeout: Second-500, Infinite
```

```
Content-Type: text/xml
```

```
Content-Length: 151
```

```
<?xml version="1.0" ?>
```

```
<d:lockinfo xmlns:d="DAV:">
```

```
  <d:locktype><d:write/></d:locktype>
```

```
  <d:lockscope><d:exclusive/></d:lockscope>
```

```
</d:lockinfo>
```



LOCK: risposta

HTTP/1.1 200 OK

Date: Tue, 09 Feb 1999 02:25:21 GMT

Server: PyDAV 1.1 filestore 1.1

Content-Type: text/xml

Content-Length: 435

```
<?xml version="1.0"?>
```

```
<a:prop xmlns:a="DAV:">
```

```
  <a:lockdiscovery>
```

```
    <a:activelock>
```

```
      <a:lockscope><a:exclusive/></a:lockscope>
```

```
      <a:locktype><a:write/></a:locktype>
```

```
      <a:depth>infinity</a:depth>
```

```
      <a:timeout>Second-500</a:timeout>
```

```
      <a:locktoken>
```

```
        <a:href>opaquelocktoken:918527121.406</a:href>
```

```
      </a:locktoken>
```

```
    </a:activelock>
```

```
  </a:lockdiscovery>
```

```
</a:prop>
```



UNLOCK

UNLOCK rimuove il lock identificato dal token, non solo sulla risorsa specificata nell'URI della richiesta, ma anche su tutte le risorse incluse nel lock.

Se un lock è stato richiesto su una collezione, tutta la collezione viene liberata dal lock: il lock utilizza la semantica del tutto o niente.



Feature Discovery

Il metodo OPTIONS (è di HTTP!) viene usato per scoprire le caratteristiche HTTP o WebDAV supportate dal server per una data risorsa.

```
OPTIONS /wg/ HTTP/1.1
```

```
Host: www.webdav.org
```

```
HTTP/1.1 200 OK
```

```
Date: Tue, 09 Feb 1999 02:39:06 GMT
```

```
Server: Apache/1.3.4 (Unix) DAV/0.9.6
```

```
Content-Length: 0
```

```
Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS, TRACE,  
PROPFIND, PROPPATCH, MKCOL, COPY, MOVE
```

```
DAV: 1
```



Riferimenti

<http://www.webdav.org>

- Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen, HTTP Extensions for Distributed Authoring -- WEBDAV, RFC 2518, February 1999, <http://www.ietf.org/rfc/rfc2518.txt>
- G. Clemm, J. Amsden, T. Ellison, C. Kaler, J. Whitehead, *Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)*, RFC 3253, March 2002, <http://www.ietf.org/rfc/rfc3253.txt>

