HTTPbis Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

M. Nottingham
Akamai
P. McManus
Mozilla
J. Reschke
greenbytes

July 4, 2014

HTTP Alternative Services
draft-ietf-httpbis-alt-svc-02

Abstract

This document specifies "alternative services" for HTTP, which allow an origin’s resources to be authoritatively available at a separate network location, possibly accessed with a different protocol configuration.

Editorial Note (To be removed by RFC Editor)

Discussion of this draft takes place on the HTTPBIS working group mailing list (ietf-http-wg@w3.org), which is archived at <https://lists.w3.org/Archives/Public/ietf-http-wg/>.

Working Group information can be found at <https://tools.ietf.org/wg/httpbis/> and <http://httpwg.github.io/>; source code and issues list for tis draft can be found at <https://github.com/httpwg/http-extensions>.

The changes in this draft are summarized in Appendix A.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
  1.1. Notational Conventions 3
2. Alternative Services Concepts 4
  2.1. Host Authentication 5
  2.2. Alternative Service Caching 6
  2.3. Requiring Server Name Indication 6
  2.4. Using Alternative Services 6
3. The Alt-Svc HTTP Header Field 7
  3.1. Caching Alt-Svc Header Field Values 9
4. The ALTSVC HTTP/2 Frame 10
5. The Alt-Svc-Used HTTP Header Field 11
6. The 421 Not Authoritative HTTP Status Code 12
7. IANA Considerations 12
  7.1. Header Field Registrations 12
  7.2. The ALTSVC HTTP/2 Frame Type 13
8. Internationalization Considerations 13
9. Security Considerations 13
  9.1. Changing Ports 13
  9.2. Changing Hosts 13
  9.3. Changing Protocols 14
  9.4. Tracking Clients Using Alternative Services 14
10. Acknowledgements 15
11. References 15
  11.1. Normative References 15
  11.2. Informative References 16
Appendix A. Change Log (to be removed by RFC Editor before publication) 16
  A.1. Since draft-nottingham-httpbis-alt-svc-05 16
  A.2. Since draft-ietf-httpbis-alt-svc-00 16
  A.3. Since draft-ietf-httpbis-alt-svc-01 16

1. Introduction

HTTP [RFC7230] conflates the identification of resources with their location. In other words, "http://" (and "https://") URLs are used to both name and find things to interact with.

In some cases, it is desirable to separate these aspects; to be able to keep the same identifier for a resource, but interact with it using a different location on the network.

For example:

- An origin server might wish to redirect a client to an alternative when it needs to go down for maintenance, or it has found an alternative in a location that is more local to the client.

- An origin server might wish to offer access to its resources using a new protocol (such as HTTP/2, see [HTTP2]) or one using improved security (such as Transport Layer Security (TLS), see [RFC5246]).

- An origin server might wish to segment its clients into groups of capabilities, such as those supporting Server Name Indication (SNI, see Section 3 of [RFC6066]) and those not supporting it, for operational purposes.

This specification defines a new concept in HTTP, "Alternative Services", that allows a resource to nominate additional means of interacting with it on the network. It defines a general framework for this in Section 2, along with specific mechanisms for advertising their existence using HTTP header fields (Section 3) or an HTTP/2 frame type (Section 4).

It also introduces a new status code in Section 6, so that origin servers (or their nominated alternatives) can indicate that they are not authoritative for a given origin, in cases where the wrong location is used.

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document uses the Augmented BNF defined in [RFC5234] along with the "OWS", "delta-seconds", "parameter", "port", "quoted-string", "token", and "uri-host" rules from [RFC7230], and uses the "#rule" extension defined in Section 7 of that document.

2. Alternative Services Concepts

This specification defines a new concept in HTTP, the "alternative service". When an origin (see [RFC6454]) has resources that are accessible through a different protocol / host / port combination, it is said to have an alternative service.

An alternative service can be used to interact with the resources on an origin server at a separate location on the network, possibly using a different protocol configuration. Alternative services are considered authoritative for an origin's resources, in the sense of [RFC7230], Section 9.1.

For example, an origin:

("http", "www.example.com", "80")

might declare that its resources are also accessible at the alternative service:

("h2", "new.example.com", "81")

By their nature, alternative services are explicitly at the granularity of an origin; i.e., they cannot be selectively applied to resources within an origin.

Alternative services do not replace or change the origin for any given resource; in general, they are not visible to the software "above" the access mechanism. The alternative service is essentially alternative routing information that can also be used to reach the origin in the same way that DNS CNAME or SRV records define routing information at the name resolution level. Each origin maps to a set of these routes -- the default route is derived from origin itself and the other routes are introduced based on alternative-protocol information.

Furthermore, it is important to note that the first member of an alternative service tuple is different from the "scheme" component of an origin; it is more specific, identifying not only the major version of the protocol being used, but potentially communication options for that protocol.

This means that clients using an alternative service will change the host, port and protocol that they are using to fetch resources, but these changes MUST NOT be propagated to the application that is using HTTP; from that standpoint, the URI being accessed and all information derived from it (scheme, host, port) are the same as before.

Importantly, this includes its security context; in particular, when TLS [RFC5246] is in use, the alternative server will need to present a certificate for the origin's host name, not that of the alternative. Likewise, the Host header field ([RFC7230], Section 5.4) is still derived from the origin, not the alternative service (just as it would if a CNAME were being used).

The changes MAY, however, be made visible in debugging tools, consoles, etc.

Formally, an alternative service is identified by the combination of:

- An Application Layer Protocol Negotiation (ALPN) protocol, as per [ALPN]

- A host, as per [RFC3986], Section 3.2.2

- A port, as per [RFC3986], Section 3.2.3

Additionally, each alternative service MUST have:

- A freshness lifetime, expressed in seconds; see Section 2.2

There are many ways that a client could discover the alternative service(s) associated with an origin. This document describes two such mechanisms: an HTTP header field (Section 3) and an HTTP/2 frame type (Section 4).

2.1. Host Authentication

Clients MUST NOT use alternative services with a host other than the origin's without strong server authentication; this mitigates the attack described in Section 9.2. One way to achieve this is for the alternative to use TLS with a certificate that is valid for that origin.

For example, if the origin's host is "www.example.com" and an alternative is offered on "other.example.com" with the "h2" protocol, and the certificate offered is valid for "www.example.com", the client can use the alternative. However, if "other.example.com" is offered with the "h2c" protocol, the client cannot use it, because there is no mechanism in that protocol to establish strong server authentication.

Furthermore, this means that the HTTP Host header field and the SNI information provided in TLS by the client will be that of the origin, not the alternative.

2.2. Alternative Service Caching

Mechanisms for discovering alternative services can associate a freshness lifetime with them; for example, the Alt-Svc header field uses the "ma" parameter.

Clients MAY choose to use an alternative service instead of the origin at any time when it is considered fresh; see Section 2.4 for specific recommendations.

Clients with existing connections to alternative services are not needed to fall back to the origin when its freshness lifetime ends; i.e., the caching mechanism is intended for limiting how long an alternative service can be used for establishing new requests, not limiting the use of existing ones.

To mitigate risks associated with caching compromised values (see Section 9.2 for details), user agents SHOULD examine cached alternative services when they detect a change in network configuration, and remove any that could be compromised (for example, those whose association with the trust root is questionable). UAs that do not have a means of detecting network changes SHOULD place an upper bound on their lifetime.

2.3. Requiring Server Name Indication

A client MUST only use a TLS-based alternative service if the client also supports TLS Server Name Indication (SNI). This supports the conservation of IP addresses on the alternative service host.

2.4. Using Alternative Services

By their nature, alternative services are OPTIONAL: clients do not need to use them. However, it is advantageous for clients to behave in a predictable way when they are used by servers (e.g., for load balancing).

Therefore, if a client becomes aware of an alternative service, the client SHOULD use that alternative service for all requests to the associated origin as soon as it is available, provided that the security properties of the alternative service protocol are desirable, as compared to the existing connection.

When a client uses an alternate service, it MUST emit the Alt-Svc-Used header field (Section 5) on every request using that alternate service.

The client does not need to block requests; the origin’s connection

can be used until the alternative connection is established. However, if the security properties of the existing connection are weak (e.g. cleartext HTTP/1.1) then it might make sense to block until the new connection is fully available in order to avoid information leakage.

Furthermore, if the connection to the alternative service fails or is unresponsive, the client MAY fall back to using the origin. Note, however, that this could be the basis of a downgrade attack, thus losing any enhanced security properties of the alternative service.

3. The Alt-Svc HTTP Header Field

An HTTP(S) origin server can advertise the availability of alternative services to clients by adding an Alt-Svc header field to responses.

```
Alt-Svc       = 1#( alternative *( OWS ";" OWS parameter ) )
alternative   = protocol-id "=" alt-authority
protocol-id   = token ; percent-encoded ALPN protocol identifier
alt-authority = token / quoted-string
                ; containing [ uri-host ] ":" port
```

ALPN protocol names are octet sequences with no additional constraints on format. Octets not allowed in tokens ([RFC7230], Section 3.2.6) MUST be percent-encoded as per Section 2.1 of [RFC3986]. Consequently, the octet representing the percent character "%" (hex 25) MUST be percent-encoded as well.

In order to have precisely one way to represent any ALPN protocol name, the following additional constraints apply:

1. Octets in the ALPN protocol MUST NOT be percent-encoded if they are valid token characters except "%", and

2. When using percent-encoding, uppercase hex digits MUST be used.

With these constraints, recipients can apply simple string comparison to match protocol identifiers.

The "alt-authority" component consists of an OPTIONAL uri-host ("host" in Section 3.2.2 of [RFC3986]), a colon (":"), and a port number.

For example:

Alt-Svc: http2=":8000"

This indicates the "http2" protocol on the same host using the indicated port 8000.

An example involving a change of host:

Alt-Svc: http2="new.example.org:80"

This indicates the "http2" protocol on the host "new.example.org", running on port 80. Note that the "quoted-string" syntax needs to be used when a host is specified in addition to a port (":" is not an allowed character in "token").

Examples for protocol name escaping:

ALPN protocol name	protocol-id	Note
http2	http2	No escaping needed
w=x:y#z	w%3Dx%3Ay#z	"=" and ":" escaped
x%y	x%25y	"%" needs escaping

Alt-Svc MAY occur in any HTTP response message, regardless of the status code.

Alt-Svc does not allow advertisement of alternative services on other hosts, to protect against various header-based attacks.

It can, however, have multiple values:

Alt-Svc: h2c=":8000", h2=":443"

The value(s) advertised by Alt-Svc can be used by clients to open a new connection to one or more alternative services immediately, or simultaneously with subsequent requests on the same connection.

To reduce the ability of servers to track individual clients over time (see Section 9.4), an alternative service indication sent by a client SHOULD NOT include any alternative service information other than the protocol, host and port.

When using HTTP/2 ([HTTP2]), clients SHOULD instead send an ALTSVC

frame. A single ALTSVC frame can be sent for a connection; a new frame is not needed for every request.

Note that all field elements that allow "quoted-string" syntax MUST be processed as per Section 3.2.6 of [RFC7230].

3.1. Caching Alt-Svc Header Field Values

When an alternative service is advertised using Alt-Svc, it is considered fresh for 24 hours from generation of the message. This can be modified with the 'ma' (max-age) parameter;

```
Alt-Svc: h2=":443"; ma=3600
```

which indicates the number of seconds since the response was generated the alternative service is considered fresh for.

```
ma = delta-seconds
```

See Section 4.2.3 of [RFC7234] for details of determining response age.

For example, a response:

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: 600
Age: 30
Alt-Svc: h2c=":8000"; ma=60
```

indicates that an alternative service is available and usable for the next 60 seconds. However, the response has already been cached for 30 seconds (as per the Age header field value), so therefore the alternative service is only fresh for the 30 seconds from when this response was received, minus estimated transit time.

When an Alt-Svc response header field is received from an origin, its value invalidates and replaces all cached alternative services for that origin.

See Section 2.2 for general requirements on caching alternative services.

Note that the freshness lifetime for HTTP caching (here, 600 seconds) does not affect caching of Alt-Svc values.

4. The ALTSVC HTTP/2 Frame

The ALTSVC HTTP/2 frame ([HTTP2], Section 4) advertises the availability of an alternative service to an HTTP/2 client.

The ALTSVC frame is a non-critical extension to HTTP/2. Endpoints that do not support this frame can safely ignore it.

An ALTSVC frame on a client-initiated stream indicates that the conveyed alternative service is associated with the origin of that stream.

An ALTSVC frame on stream 0 indicates that the conveyed alternative service is associated with the origin contained in the Origin field of the frame. An association with an origin that the client does not consider authoritative for the current connection MUST be ignored.

The ALTSVC frame type is 0xa (decimal 10).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Max-Age (32)                         |
+-------------------------------+---------------+---------------+
|            Port (16)          | Proto-Len (8) |
+-------------------------------+---------------+---------------+
|                        Protocol-ID (*)                        |
+---------------+-----------------------------------------------+
| Host-Len (8)  |                   Host (*)                  ...
+---------------+-----------------------------------------------+
|                          Origin? (*)                        ...
+---------------------------------------------------------------+
```

ALTSVC Frame Payload

The ALTSVC frame contains the following fields:

Max-Age: An unsigned, 32-bit integer indicating the freshness lifetime of the alternative service association, as per Section 2.2.

Port: An unsigned, 16-bit integer indicating the port that the alternative service is available upon.

Proto-Len: An unsigned, 8-bit integer indicating the length, in octets, of the Protocol-ID field.

Protocol-ID: A sequence of bytes (length determined by "Proto-Len") containing the ALPN protocol identifier of the alternative service.

Host-Len: An unsigned, 8-bit integer indicating the length, in octets, of the Host header field.

Host: A sequence of characters (length determined by "Host-Len") containing an ASCII string indicating the host that the alternative service is available upon.

Origin: An OPTIONAL sequence of characters (length determined by subtracting the length of all preceding fields from the frame length) containing the ASCII serialisation of an origin ([RFC6454], Section 6.2) that the alternate service is applicable to.

The ALTSVC frame does not define any flags.

The ALTSVC frame is intended for receipt by clients; a server that receives an ALTSVC frame MUST treat it as a connection error of type PROTOCOL_ERROR.

The ALTSVC frame is processed hop-by-hop. An intermediary MUST NOT forward ALTSVC frames, though it can use the information contained in ALTSVC frames in forming new ALTSVC frames to send to its own clients.

5. The Alt-Svc-Used HTTP Header Field

The Alt-Svc-Used HTTP header field is used in requests to indicate that an alternate service is in use.

```
Alt-Svc-Used     = ("1" / "0") *( OWS ";" OWS Alt-Svc-Used-Ext )
Alt-Svc-Used-Ext = token "=" ( token / quoted-string )
```

Alt-Svc-Used is intended to allow alternate services to avoid sending alternative service indications where there is a risk of generating a loops. It also allows a service to identify requests for accounting and load balancing purposes.

When using an alternative service, clients MUST include a Alt-Svc-Used header field in all requests.

For example:

```
GET /thing HTTP/1.1
Host: origin.example.com
Alt-Svc-Used: 1
```

The extension parameters (Alt-Svc-Used-Ext) are reserved for future use; specifications that want to define an extension will need to update this document (and ought to introduce an extension registry).

6. The 421 Not Authoritative HTTP Status Code

The 421 (Not Authoritative) status code is defined in [HTTP2], Section 9.1.2 to indicate that the current server instance is not authoritative for the requested resource. This can be used to indicate that an alternative service is not authoritative; see Section 2).

Clients receiving 421 (Not Authoritative) from an alternative service MUST remove the corresponding entry from its alternative service cache (see Section 2.2) for that origin. Regardless of the idempotency of the request method, they MAY retry the request, either at another alternative server, or at the origin.

A 421 (Not Authoritative) response MAY carry an Alt-Svc header field.

7. IANA Considerations

7.1. Header Field Registrations

HTTP header fields are registered within the "Message Headers" registry maintained at <https://www.iana.org/assignments/message-headers/>.

This document defines the following HTTP header fields, so their associated registry entries shall be added according to the permanent registrations below (see [BCP90]):

Header Field Name	Protocol	Status	Reference
Alt-Svc	http	standard	Section 3
Alt-Svc-Used	http	standard	Section 5

The change controller is: "IETF (iesg@ietf.org) - Internet Engineering Task Force".

7.2. The ALTSVC HTTP/2 Frame Type

This document registers the ALTSVC frame type in the HTTP/2 Frame Types registry ([HTTP2], Section 11.2).

Frame Type: ALTSVC

Code: 0xa

Specification: Section 4 of this document

8. Internationalization Considerations

An internationalized domain name that appears in either the header field (Section 3) or the HTTP/2 frame (Section 4) MUST be expressed using A-labels ([RFC5890], Section 2.3.2.1).

9. Security Considerations

9.1. Changing Ports

Using an alternative service implies accessing an origin's resources on an alternative port, at a minimum. An attacker that can inject alternative services and listen at the advertised port is therefore able to hijack an origin.

For example, an attacker that can add HTTP response header fields can redirect traffic to a different port on the same host using the Alt-Svc header field; if that port is under the attacker's control, they can thus masquerade as the HTTP server.

This risk can be mitigated by restricting the ability to advertise alternative services, and restricting who can open a port for listening on that host.

9.2. Changing Hosts

When the host is changed due to the use of an alternative service, it presents an opportunity for attackers to hijack communication to an origin.

For example, if an attacker can convince a user agent to send all traffic for "innocent.example.org" to "evil.example.com" by successfully associating it as an alternative service, they can masquerade as that origin. This can be done locally (see mitigations above) or remotely (e.g., by an intermediary as a man-in-the-middle attack).

This is the reason for the requirement in Section 2.1 that any alternative service with a host different to the origin's be strongly authenticated with the origin's identity; i.e., presenting a certificate for the origin proves that the alternative service is authorized to serve traffic for the origin.

However, this authorization is only as strong as the method used to authenticate the alternative service. In particular, there are well-known exploits to make an attacker's certificate appear as legitimate.

Alternative services could be used to persist such an attack; for example, an intermediary could man-in-the-middle TLS-protected communication to a target, and then direct all traffic to an alternative service with a large freshness lifetime, so that the user agent still directs traffic to the attacker even when not using the intermediary.

As a result, there is a requirement in Section 2.2 to examine cached alternative services when a network change is detected.

9.3. Changing Protocols

When the ALPN protocol is changed due to the use of an alternative service, the security properties of the new connection to the origin can be different from that of the "normal" connection to the origin, because the protocol identifier itself implies this.

For example, if a "https://" URI had a protocol advertised that does not use some form of end-to-end encryption (most likely, TLS), it violates the expectations for security that the URI scheme implies.

Therefore, clients cannot blindly use alternative services, but instead evaluate the option(s) presented to assure that security requirements and expectations (of specifications, implementations and end users) are met.

9.4. Tracking Clients Using Alternative Services

The alternative service indicator (Section 5) provided by clients provides a server the means of correlating requests. If the alternative service indicator includes a sufficiently unique identifier, requests made to an alternative service can be correlated with the original alternative service advertisement.

Clients that do not wish to be tracked MAY choose to ignore alternative service advertisements.

In a browser, any alternative service information MUST be removed when origin-specific data is cleared (for instance, when cookies are cleared).

10. Acknowledgements

Thanks to Eliot Lear, Stephen Farrell, Guy Podjarny, Stephen Ludin, Erik Nygren, Paul Hoffman, Adam Langley, Will Chan and Richard Barnes for their feedback and suggestions.

The Alt-Svc header field was influenced by the design of the Alternate-Protocol header field in SPDY.

11. References

11.1. Normative References

[ALPN] Friedl, S., Popov, A., Langley, A., and S. Emile, "Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension", draft-ietf-tls-applayerprotoneg-05 (work in progress), March 2014.

[HTTP2] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol version 2", draft-ietf-httpbis-http2-13 (work in progress), June 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, December 2011.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing",

RFC 7230, June 2014.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, June 2014.

11.2. Informative References

[BCP90] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Appendix A. Change Log (to be removed by RFC Editor before publication)

A.1. Since draft-nottingham-httpbis-alt-svc-05

This is the first version after adoption of draft-nottingham-httpbis-alt-svc-05 as Working Group work item. It only contains editorial changes.

A.2. Since draft-ietf-httpbis-alt-svc-00

Selected 421 as proposed status code for "Not Authoritative".

Changed header field syntax to use percent-encoding of ALPN protocol names (<https://github.com/http2/http2-spec/issues/446>).

A.3. Since draft-ietf-httpbis-alt-svc-01

Updated HTTP/1.1 references.

Renamed "Service" to "Alt-Svc-Used" and reduced information to a flag to address fingerprinting concerns (<https://github.com/http2/http2-spec/issues/502>).

Note that ALTSVC frame is preferred to Alt-Svc header field (<https://github.com/http2/http2-spec/pull/503>).

Incorporate ALTSRV frame (<https://github.com/http2/http2-spec/pull/507>).

Moved definition of status code 421 to HTTP/2.

Partly resolved <https://github.com/httpwg/http-extensions/issues/5>.

Authors' Addresses

Mark Nottingham
Akamai

EMail: mnot@mnot.net
URI: https://www.mnot.net/

Patrick McManus
Mozilla

EMail: mcmanus@ducksong.com
URI: https://mozillians.org/u/pmcmanus/

Julian F. Reschke
greenbytes GmbH

EMail: julian.reschke@greenbytes.de
URI: https://greenbytes.de/tech/webdav/