HTTP
Internet-Draft
Intended status: Informational
Expires: 17 July 2026

C. Lecklider
Independent

13 January 2026

HTTP Content Negotiation for Consolidated Machine-Readable Representations
draft-consolidated-content-00

Abstract

This document specifies the use of HTTP content negotiation with the Prefer header to request consolidated, machine-optimised representations of web resources. Clients use the Accept header for format negotiation and the Prefer header with return=consolidated to request content optimised for automated consumption. Publishers benefit from dramatically reduced request volume and the ability to present contextually complete information, improving both operational efficiency and content effectiveness. Consolidated resources enable practical caching, further reducing load. This document introduces a single new preference value for the existing Prefer header and otherwise relies solely on established HTTP mechanisms.

Table of Contents

1. Introduction

Web content is traditionally structured for human navigation through HTML pages. Automated agents retrieving this content for analysis or training must fetch multiple pages to obtain complete information about a topic. This creates unnecessary server load from repeated page fetches, consumes bandwidth inefficiently, produces fragmented information requiring client-side reassembly, and makes change detection difficult. The volume and fragmentation of web content make comprehensive caching impractical for automated systems, compounding the inefficiency.

HTTP provides content negotiation (Section 12 of [RFC9110]) and client preferences ([RFC7240]) to address varying client needs. This document specifies how these existing mechanisms can be combined so that clients can request consolidated representations optimised for machine consumption in appropriate machine-readable formats. It introduces a new return=consolidated preference value for the existing Prefer header and does not define any new HTTP headers or media types.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Requesting Consolidated Representations

Clients request consolidated representations using two standard HTTP headers.

2.1. Accept Header

Clients indicate preferred content type using the Accept header, with the desired machine-readable format given highest priority and other formats adjusted accordingly:

Accept: text/markdown;q=0.9, text/html;q=0.8

or

Accept: application/json;q=0.9, text/html;q=0.8

Clients MAY specify multiple formats with appropriate quality values.

2.2. Prefer Header

Clients indicate desire for consolidated content using the Prefer header with the return preference [RFC7240]:

```
Prefer: return=consolidated
```

2.3. Combined Request

A complete request combines both headers:

```
GET /documentation HTTP/1.1
Host: example.org
Accept: text/markdown;q=0.9, text/html;q=0.8
Prefer: return=consolidated
```

3. Server Behaviour

3.1. Honouring Preferences

Servers receiving requests with Prefer: return=consolidated SHOULD provide consolidated representations when practical. Servers that honour the preference MUST include Preference-Applied in the response:

```
HTTP/1.1 200 OK
Content-Type: text/markdown
Preference-Applied: return=consolidated
ETag: "consolidated-v1-a3f8b2"
Vary: Accept, Prefer
```

The Vary header MUST include both Accept and Prefer to ensure proper caching behaviour by intermediaries (proxies, CDNs). Other headers (such as Accept-Encoding) MAY also appear in Vary as appropriate. Without appropriate Vary headers, caches may incorrectly serve consolidated representations to clients that did not request them, or vice versa.

3.2. Content Structure

Consolidated representations SHOULD differ in structure and organisation from their navigational HTML counterparts. Servers SHOULD:

* Consolidate related content from multiple pages into hierarchical sections

* Organise information by semantic relationships rather than navigation structure

* Include appropriate context for understanding without navigation chrome

* Preserve information fidelity while restructuring for machine consumption

* Focus on coherent topics rather than consolidating entire sites

Very small sites MAY consolidate all content into a single resource. Larger sites SHOULD create multiple focused consolidated resources, each addressing a specific topic or information need. Content that is not directly relevant to understanding the primary topic SHOULD be excluded.

3.3. Discovery

Publishers MAY advertise the availability of alternate representations using HTML <link> elements in the same manner as feed discovery:

```
<link rel="alternate" type="text/markdown"
      href="/products/catalogue"
      title="Product Catalogue (Markdown)">
<link rel="alternate" type="application/ld+json"
      href="/products/catalogue"
      title="Product Catalogue (Linked Data)">
```

This allows automated agents to discover available formats without relying solely on content negotiation. The href attribute points to the resource URL, and the type attribute indicates the available media type. These alternate representations may support consolidation via the Prefer header, or may simply be format conversions - the <link> element advertises format availability either way. Publishers providing alternate representations for multiple resources MAY include multiple sets of <link> elements.

Representations MAY also advertise alternate formats of the same resource within themselves. For example, a markdown representation might include in its frontmatter:

```
alternate-formats:
  - type: application/ld+json
    title: Product Catalogue (Linked Data)
  - type: text/html
    title: Product Catalogue (HTML)
```

This enables format discovery for agents that bypass HTML parsing and directly request machine-readable representations. The URL for alternate formats is the same as the current resource.

3.4. When Consolidation is Not Practical

Servers MAY decline to provide consolidated representations by serving the standard representation without the Preference-Applied header.

4. Caching Benefits

Consolidated representations can be cached independently with their own ETag values. This enables efficient conditional requests:

```
GET /documentation HTTP/1.1
Host: example.org
Accept: text/markdown;q=0.9
Prefer: return=consolidated
If-None-Match: "consolidated-v1-a3f8b2"

HTTP/1.1 304 Not Modified
Vary: Accept, Prefer
```

Clients can verify whether content has changed with a single request rather than fetching multiple individual pages. For publishers experiencing high load from automated crawlers, this can significantly reduce bandwidth and server processing costs.

Servers SHOULD use appropriate cache-related headers (Cache-Control, Expires, etc. as specified in [RFC9111]) to enable intermediaries and clients to cache consolidated representations effectively. While conditional requests provide significant benefits, proper caching eliminates requests entirely.

5. Applicability to HTTP Versions

This specification is protocol-agnostic and applies equally to HTTP/1.1, HTTP/2 [RFC9113], and HTTP/3 [RFC9114]. The examples in this document use HTTP/1.1 syntax for clarity, but the mechanisms work identically across all HTTP versions.

6. Security Considerations

This specification uses existing HTTP mechanisms and introduces no new security considerations beyond those in [RFC9110] Section 12 (content negotiation) and [RFC7240] (Prefer header).

Per [RFC7240], recipients that do not understand a particular preference value SHOULD ignore it rather than rejecting the request. However, some non-compliant servers, frameworks, or Web Application Firewalls (WAFs) may have stricter validation and could reject requests containing unknown preference values. Implementations that currently reject unknown preference values may need configuration updates to recognise return=consolidated as a valid preference value.

Servers SHOULD apply the same access controls to consolidated representations as to their constituent pages.

6.1. Content Quality

Consumers MUST NOT grant consolidated representations greater trust than equivalent content from other sources. As with any web technology, some publishers will attempt to exploit consolidated representations to gain preferential treatment from automated systems through keyword stuffing, artificial relevance signals, or other manipulative techniques. The presence of consolidated representations is a technical capability, not a quality signal. Consumers of consolidated content SHOULD treat it as one signal among many when assessing quality and relevance.

6.2. Content Integrity

While this specification does not introduce new attack vectors, consolidated representations may amplify the impact of existing content integrity issues. A single poisoned consolidated resource in a cache could affect more automated systems than poisoning individual pages, and similarly, malicious content from origin server compromise may have broader impact when consolidated. Publishers should re-evaluate their content integrity measures.

7. IANA Considerations

IANA is requested to update the HTTP Preferences registry established by [RFC7240] to add the following value for the return preference:

* Preference: return
* Value: consolidated
* Description: Indicates that the client prefers a consolidated, machine-readable representation of the resource optimised for automated consumption
* Reference: this document

8. References

8.1. Normative References

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[RFC9111] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD 98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <https://www.rfc-editor.org/rfc/rfc9111>.

[RFC7240] Snell, J., "Prefer Header for HTTP", RFC 7240, DOI 10.17487/RFC7240, June 2014, <https://www.rfc-editor.org/rfc/rfc7240>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

8.2. Informative References

[RFC9113] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <https://www.rfc-editor.org/rfc/rfc9113>.

[RFC9114] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/rfc/rfc9114>.

Appendix A. Practical Benefits

A.1. Why This Matters

Automated systems increasingly access web content: search engines, AI training systems, research tools, and monitoring services. These systems must fetch multiple pages to assemble complete information, repeatedly crawl sites to detect changes, and employ heuristics to distinguish content from presentational markup. This inefficiency costs money: publishers serve largely redundant traffic, consumers fetch and process bloated responses, and both produce suboptimal results.

This specification provides a mechanism for publishers to serve consolidated, machine-optimised representations directly. Publishers reduce costs and load, and consumers improve information quality and efficiency.

A.2. Beyond Simple Conversion

While format conversion alone reduces bandwidth and parsing overhead, consolidated representations provide something more valuable: a direct way to communicate relevance and context to automated systems. Rather than forcing machines to scrape multiple pages and infer relationships, consolidated representations explicitly state what information belongs together (see Example Transformation below for concrete illustrations).

By consolidating related information, publishers guide automated systems to the complete picture while implicitly indicating, by omission, what is not relevant. This benefits both parties: machines get better information with less work, publishers reduce load while maintaining control over how their content is understood.

A.3. Quantifiable Benefits

Based on typical modern web architectures, implementing consolidated representations can yield substantial operational improvements. The figures in this section are illustrative estimates intended to convey likely magnitudes of effect, not results of formal benchmarking.

Server load can, in many cases, drop by on the order of 70-90% per information retrieval session, as a single consolidated resource replaces five to twenty individual page fetches. This reduction cascades through the infrastructure: fewer database queries, less server-side rendering overhead, and diminished load on application servers.

Bandwidth consumption often decreases proportionally. Consolidated representations in machine-readable formats are typically significantly smaller than equivalent HTML pages, lacking navigation chrome, advertisements, and presentational markup. A single ETag check can replace multiple page checks for change detection, further reducing transfer overhead. Network egress costs decline accordingly.

Caching efficiency can also improve markedly. Rather than caching dozens of individual pages, CDNs and intermediaries cache single consolidated resources. Cache hit rates increase, invalidation simplifies, and CDN costs drop.

Illustrative example: Consider a documentation site receiving 10,000 automated agent visits daily. Suppose each visit currently fetches an average of 10 pages to assemble complete information (typical for documentation traversal), representing 100,000 daily requests. With consolidated representations reducing this to one request per visit, the site serves 90,000 fewer requests daily.

Modern documentation pages of even modest visual complexity can easily reach 1 MB once navigation, styling, and scripts are included, while equivalent consolidated markdown representations might plausibly be 350 KB for the same underlying content. In this scenario, bandwidth drops from approximately 100 GB daily (HTML) to 3.5 GB daily (consolidated markdown) - a reduction of roughly 2.9 terabytes monthly. Actual figures will vary by site design and traffic patterns, but these orders of magnitude are realistic for many contemporary sites.

Automated systems benefit as well. Consolidated representations provide clearer information structure, reducing parsing errors from complex HTML and JavaScript execution. Deliberate consolidation supplies context that scattered pages cannot, improving comprehension and result quality.

A.4. Relationship to Other Approaches

Various initiatives have attempted to make web content more machine-accessible. The Semantic Web / RDF / Linked Data efforts and embedded structured data approaches like Microdata and Schema.org have been under development for decades, yet broad and consistent adoption has not materialised. Feed formats like RSS and Atom achieved significant adoption but remained separate from the standard web browser model, requiring dedicated client software or an aggregator. In practice, this separation was vulnerable: when major implementations were discontinued, the ecosystem fragmented and recovery proved difficult. These approaches suggest fundamental challenges: either misalignment between protocol requirements and publisher willingness, or fragile architectural independence.

This specification takes a different approach by using only existing HTTP mechanisms.

* *Uses existing HTTP content negotiation* - RFC 9110

* *Uses existing Prefer header* - RFC 7240

* *Uses existing media types* - for example, text/markdown (RFC 7763), application/json, application/xml

* *No new protocols, no new standards, no new infrastructure*

Unlike approaches that prescribe specific data models or require adoption of complex frameworks, this specification leaves content organisation entirely to implementers. The challenge shifts from technical implementation to editorial judgement: what information belongs together, what context is needed, what can be omitted.

A.5. Contemporary Approaches

The llms.txt initiative (September 2024) provides machine-readable markdown at conventional paths like /llms.txt. This follows the pattern of robots.txt and sitemap.xml - pragmatic bolt-on conventions that work immediately without protocol changes.

Both approaches recognise that machines need clean content without presentational complexity. The llms.txt convention offers apparent simplicity: create markdown files, document their locations, agents fetch them. However, semantic equivalence depends on filename convention (foo.html.md is equivalent to foo.html) and content discipline; if the markdown file contains additional material, it is no longer equivalent but a different resource. Machines must fetch and parse the llms.txt index to infer relationships between resources; meaning emerges from content rather than protocol.

This specification establishes semantic equivalence through protocol mechanisms. The same URL serves both representations, with the Prefer header providing explicit protocol-level semantics: "this consolidated representation IS this resource". Machines understand resource relationships before fetching content. The Preference-Applied header confirms consolidation, eliminating inference. For most publishers, CMS platforms will handle the protocol mechanics, requiring no technical intervention from site operators. For static sites, simple server configuration can serve markdown files via content negotiation whilst maintaining compatibility with llms.txt filename conventions.

The approaches differ in architecture - a bolt-on convention with content-level inference versus protocol integration with explicit semantics - but share the goal of making web content accessible to automated systems.

Appendix B. Example Transformation

The following examples use text/markdown and application/json to illustrate consolidated representations. These formats are chosen for clarity and represent common use cases, but the consolidation principle applies equally to any machine-readable format.

B.1. Markdown Example: Product Website

Consider a typical small business website with navigational structure:

```
Site structure (navigational):
/               (landing page: hero image, value proposition,
                 social proof testimonials, call-to-action buttons)
/features/      (feature list with marketing copy)
/features/a/    (detailed feature A with screenshots)
/features/b/    (detailed feature B with screenshots)
/pricing/       (pricing tiers with comparison table)
/contact/       (contact form, office locations, map)
/docs/          (technical documentation)
```

A consolidated representation for the root resource provides an overview with links to detailed consolidated resources:

```
GET / HTTP/1.1
Accept: text/markdown;q=0.9
Prefer: return=consolidated

# Product Overview
[Value proposition and core description from landing page.
 Hero images, testimonials, and CTAs omitted]

High-level feature summary with key capabilities.

For detailed information:
- Features: /features/
- Pricing: /pricing/
- Documentation: /docs/

[Note: Links point to same URLs; consolidated representations
 served based on Accept/Prefer headers]
```

A consolidated representation for the features resource provides comprehensive technical detail:

```
GET /features/ HTTP/1.1
Accept: text/markdown;q=0.9
Prefer: return=consolidated

# Features

## Feature A
### What it does
[Consolidated from /features/ and /features/a/ - technical
 description of capabilities]

### How it works
[Implementation details from /docs/ where relevant]

### Pricing
[Relevant pricing tier information for this feature, consolidated
 from /pricing/]

### Technical requirements
[System requirements, API details, integration notes]

## Feature B
[Similar comprehensive structure]

Related resources:
- Complete technical documentation: /docs/
- Pricing comparison: /pricing/
```

Note the key differences from simple page conversion:

* *Multiple consolidated views*: Different URLs provide different semantic organisations of the same underlying content

* *Deep consolidation*: Feature pages pull in relevant pricing, documentation, and technical details

* *Semantic restructuring*: Content organised by "what/how/ requirements" rather than mirroring site navigation

* *Selective omission*: Marketing copy, testimonials, decorative elements excluded

* *Preserved navigation*: Links to other consolidated resources maintained for context

This illustrates the key principle: consolidation is about semantic organisation and selective inclusion of substantive information, not mechanical conversion of all page content. Each consolidated

resource provides a complete, contextual view optimised for understanding that specific topic, drawing from multiple source pages as needed.

B.2. JSON Example: Financial News Article

Consolidated representations are not limited to text/markdown. Consider a financial news website:

Site structure (navigational):

```
/article/12345      (news article with ads, related links)
/stock/ACME         (stock price chart and basics)
/company/acme-corp  (company profile)
/filings/acme-q3    (SEC filing summary)
```

A consolidated JSON representation for the article provides structured data combining relevant financial information:

```
GET /article/12345 HTTP/1.1
Accept: application/json;q=0.9
Prefer: return=consolidated

HTTP/1.1 200 OK
Content-Type: application/json
Preference-Applied: return=consolidated
ETag: "article-12345-consolidated-v2"
Vary: Accept, Prefer

{
  "alternateFormats": [
    {
      "type": "text/markdown",
      "title": "Article (Markdown)"
    }
  ],
  "article": {
    "headline": "ACME Corp Reports Strong Q3 Results",
    "published": "2024-12-16T14:30:00Z",
    "summary": "ACME Corp exceeded analyst expectations...",
    "content": "[Article text without ads/chrome]"
  },
  "financial_data": {
    "stock_symbol": "ACME",
    "current_price": 142.50,
    "change_percent": 8.3,
    "market_cap": "125B",
    "as_of": "2024-12-16T20:00:00Z"
  },
```

```
    "company": {
      "name": "ACME Corporation",
      "sector": "Technology",
      "employees": 15000,
      "founded": 1995
    },
    "quarterly_results": {
      "period": "Q3 2024",
      "revenue": "8.2B",
      "revenue_growth": 12.5,
      "eps": 1.85,
      "eps_expected": 1.72
    },
    "related": {
      "stock_details": "/stock/ACME",
      "company_profile": "/company/acme-corp",
      "sec_filings": "/filings/acme-q3"
    }
  }
```

This JSON consolidation pulls key financial metrics from separate stock ticker, company profile, and filing pages, presenting them alongside the article content in a structured format optimised for automated analysis. The same article URL serves both human-readable HTML and machine-readable consolidated JSON based on content negotiation.

Appendix C. Implementation

C.1. Client Implementation

Clients can begin requesting consolidated representations immediately. The protocol is designed for graceful degradation: if a server does not support consolidated representations, it will simply return the standard representation (typically HTML), which clients already handle. There is no risk in asking, and early adoption benefits clients immediately whenever any publisher implements support, with no cost when publishers have not yet done so.

Clients SHOULD request text/markdown by default, as it handles the vast majority of web content effectively. When requesting resources known to contain structured data (API endpoints, financial feeds, datasets), clients SHOULD request application/json instead. These formats have widespread parser support and cover nearly all use cases. Clients can refine their format preferences as publisher implementations mature and specific needs emerge.

Clients SHOULD:

1. Include Prefer: return=consolidated in requests where consolidated content would be beneficial

2. Specify appropriate Accept headers (text/markdown by default for text-heavy resources, application/json for known structured resources)

3. Check for Preference-Applied: return=consolidated in responses to confirm support

4. Fall back to standard content processing when the preference is not honoured

C.2. Server Implementation

Implementation requires three steps:

1. Parse the Prefer header and recognise return=consolidated

2. Serve machine-readable format when requested via Accept header

3. Generate appropriate consolidated content for the publisher’s use case

Existing web servers, frameworks, or WAFs that already parse the Prefer header may need minor updates to recognise the return=consolidated value. Beyond that, the technical implementation is straightforward. Publishers SHOULD log when consolidated representations are served for analytics and capacity planning; logging the Preference-Applied response header provides one straightforward approach.

Publishers SHOULD begin by implementing text/markdown for text-heavy content and application/json for structured data. These formats have widespread parser support, are straightforward to generate, and align with client expectations. Publishers may evolve their format offerings as the ecosystem matures and specific consumer needs emerge, but starting with these pragmatic defaults ensures immediate interoperability.

The real work lies in content decisions: which information to consolidate, how to structure it, what context to include, what to omit. These decisions depend on site architecture, content type, and audience needs. This specification provides the mechanism, publishers provide the judgement. (Large language models may prove surprisingly capable at making these editorial decisions, should publishers wish to automate the process.)

Publishers can begin with minimal investment; even simple format conversion of existing pages provides immediate bandwidth and load reduction benefits. The incremental adoption approach (below) allows publishers to start small and expand based on observed value.

C.3. Incremental Adoption

Implementing consolidated representations does not require a complete site overhaul. Publishers can adopt this specification incrementally:

Phase 1: Simple Format Conversion Start by serving machine-readable versions of existing pages (ignoring the Prefer header initially). This provides immediate bandwidth and parsing benefits. A simple conversion tool can generate markdown from HTML with minimal effort.

Phase 2: Analyse Access Patterns Monitor which pages automated agents fetch together. Collect statistics on common access patterns: which documentation pages are read sequentially, which product pages are accessed alongside pricing information, and so on.

Phase 3: Create Targeted Consolidated Resources Based on usage patterns, create consolidated representations for high-traffic combinations. A site might start with just two or three strategic consolidated resources covering the most common information retrieval patterns.

Phase 4: Expand as Beneficial Add consolidated representations where server load or bandwidth justify the effort.

C.4. Format Selection

Initial adoption

Publishers and consumers SHOULD begin with text/markdown for text-heavy content (documentation, articles, blogs, guides) and application/json for structured data (financial information, API responses, datasets, metrics). These formats provide widespread parser availability, straightforward implementation, clear specifications, and existing adoption by automated consumers. Starting with these formats ensures immediate interoperability.

Publishers MAY use alternative formats where compelling domain-specific reasons exist. For example, application/xml for complex structured documents where XML tooling provides clear value, or text/csv for tabular data in domains with established CSV conventions.

Format evolution

This specification is deliberately format-agnostic to accommodate future evolution. Publishers and consumers SHOULD NOT treat markdown and JSON as permanent requirements. They are pragmatic starting points that work today. Should new formats emerge with compelling advantages, this specification supports their adoption through discovery and standard content negotiation. Migration is possible without specification change.

Selection principle

Keep it simple. Use formats with widespread support, clear specifications, and proven parser availability. Novel formats require compelling justification from major consumers demonstrating concrete benefits that outweigh implementation burden.

C.5. Validation and Testing

Analogous to how social media platforms provide preview tools for OpenGraph meta tags, organisations consuming consolidated representations SHOULD provide validation tools allowing publishers to verify their implementations. Such tools should provide metrics and feedback beyond what simple command-line tools offer: consolidation quality assessment, structure analysis, and guidance on whether the representation meets consumer needs. Only consumers possess the technical capability to evaluate whether consolidated representations suit their processing requirements.

Appendix D. Motivation

The World Wide Web was originally conceived as a system for sharing information, with HTML providing semantic markup focused on content and structure.

Over time, the web evolved to prioritise presentation. Modern web pages contain dramatically more presentational markup, navigation, advertising, and scripts than actual content, with the informational payload representing only a small fraction of transmitted bytes.

For human readers using browsers, this evolution has been successful. However, for automated agents attempting to extract information, this presentational complexity is counterproductive. Agents must parse elaborate HTML, execute JavaScript, and employ heuristics to distinguish content from chrome.

This specification provides a mechanism for automated agents to request consolidated, presentation-free representations. The purpose of information-rich sites - documentation, news, research, technical content - is to convey information. Whether that information is

consumed directly by humans or via automated intermediaries is immaterial; the underlying purpose remains unchanged. Sites whose primary value lies in substantive content benefit from making that content efficiently accessible to machines. Sites whose value proposition is purely presentational will find this specification of limited relevance, which is as it should be.

The challenge was to identify existing HTTP capabilities that could address this use case. The HTTP specifications are both extensive and comprehensive; practical deployment requires working within these established capabilities. Content negotiation with client preferences proved sufficient.

Acknowledgements

The author thanks Darryl Hughes and Helen King for their careful review and thoughtful feedback on this document.

Author’s Address

Charles Lecklider
Independent
Email: charlesl@invis.net
URI: https://invis.net