

# Extended Accessibility Metadata Protocol (EAMP) v1 — Full Protocol

Status: Draft • Category: Standards Track • Editors: SHAFT.Finance Accessibility WG (proposed)

## Abstract

EAMP v1 defines a two-layer accessibility metadata model that preserves WCAG 2.2 / ADA compliance (Layer 1) while introducing a structured, machine-readable layer (Layer 2) that enables AI agents, assistive technologies, and Web2/Web3 apps to retrieve rich, context-heavy descriptions, data tables, and time-coded annotations on demand. This document contains normative requirements (MUST/SHOULD/MAY), discovery and transport rules, JSON schemas, OpenAPI fragments, conformance profiles, and comprehensive examples.

## 1. Conformance & Terminology

Conformance keywords (MUST, MUST NOT, SHOULD, SHOULD NOT, MAY) are to be interpreted as described in RFC 2119. Layer 1 refers to baseline WCAG/ADA-compliant artifacts (e.g., short alternatives, captions, labels, transcripts). Layer 2 refers to extended metadata delivered in JSON (HTTP(S) or decentralized URIs) that augments, but never replaces, Layer 1. A User Agent (UA) is any browser, native app, assistive technology, or AI agent consuming metadata.

## 2. Design Goals

G1 Compatibility: Never reduce accessibility for existing assistive tech. G2 Progressive Enrichment: Provide concise, human-first alternatives by default, and expose verbose metadata upon request. G3 Broad Needs: Address visual, auditory, cognitive, and motor impairments. G4 Simplicity: Use pragmatic hooks that are easy to author and integrate. G5 Interop: Work across Web2, mobile, and Web3 ecosystems.

## 3. Layer 1 — Baseline (Normative)

L1-1 Authors MUST provide a short text alternative for non-text content (e.g., alt, aria-label, native platform equivalent). L1-2 Authors SHOULD target ~100–250 characters for screen-reader efficiency; however, when cognitive/learning accessibility requires additional clarity, authors MAY extend to 500 characters. L1-3 For complex visuals (infographics, maps, charts), Authors MUST provide a linked long description via aria-details, <details> sections, or an explicit metadata link. L1-4 Time-based media MUST include captions; descriptive transcripts SHOULD be supplied where applicable. L1-5 Controls MUST have accessible names, roles, and states; disabled/enabled conditions MUST be programmatically determinable.

### 3.1 Persona-Oriented Length Guidance

Persona	Typical L1 Length	Notes
Screen reader primary	100–250 chars	Concise summary for rapid navigation.
Cognitive/learning	250–500 chars	Plain-language, simplified phrasing allowed.
Low-vision/magnification	100–250 chars	Pair with high-contrast cues &lt;details&gt;.
Motor	100–250 chars	Emphasize control purpose/state and shortcuts.

### 3.2 Layer 1 HTML Examples

```
<!-- Complex chart with long description hook -->


<!-- Long description container (discoverable) -->
<section id="desc-sales-2024" hidden>
```

```

    <h2>Extended description: Sales 2024</h2>
    <p>Sales increase each quarter; Q4 peaks at $2.3M. Targets met in Q3-Q4.</p>
    <p>For full dataset and annotations, see metadata link.</p>
</section>

<!-- Control with explicit state -->
<button aria-pressed="false" aria-label="Play video">Play</button>
<!-- Toggleable verbose summary for cognitive support (opt-in UI) -->
<details>
  <summary>Plain-language summary</summary>
  <p>This chart shows sales going up each quarter in 2024. The biggest growth is in the last quarter.
</details>

```

### 3.3 Native Mobile Examples (iOS/Android)

```

// iOS (Swift)
imageView.isAccessibilityElement = true
imageView.accessibilityLabel = "Quarterly sales chart for 2024"
// Link to extended description via help or deep link
imageView.accessibilityHint = "Double-tap for full description and data"

// Android (Kotlin)
imageView.contentDescription = "Quarterly sales chart for 2024"
// Use talkback hint pattern in UI or link to extended description screen

```

## 4. Layer 2 — Extended Metadata (Normative)

L2-1 A JSON representation of extended metadata MUST be available at a dereferenceable URI (HTTP(S) or decentralized). L2-2 Responses MUST include: id, type, shortAlt, extendedDescription. L2-3 Authors MAY include tags, dataPoints, transcript, scenes, accessibilityFeatures, sourceAttribution, metadataURI. L2-4 The shortAlt value in Layer 2 MUST NOT exceed 250 characters. L2-5 The extendedDescription MUST NOT be arbitrarily truncated; pagination MAY be used for very large media.

### 4.1 Canonical JSON Schema (Excerpt)

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "EAMPMetadata",
  "type": "object",
  "required": ["id", "type", "shortAlt", "extendedDescription"],
  "properties": {
    "id": {"type": "string", "minLength": 1},
    "type": {"type": "string", "enum": ["image", "video", "audio", "ui-element", "document"]},
    "shortAlt": {"type": "string", "maxLength": 250},
    "extendedDescription": {"type": "string", "minLength": 1},
    "tags": {"type": "array", "items": {"type": "string"}, "uniqueItems": true},
    "dataPoints": {
      "type": "array",
      "items": {
        "type": "object",
        "required": ["label", "value"],
        "properties": {
          "label": {"type": "string"},
          "value": {"type": ["string", "number", "boolean"]},
          "unit": {"type": "string"}
        }
      }
    },
    "transcript": {"type": "string"},
    "scenes": {
      "type": "array",
      "items": {
        "type": "object",
        "required": ["time", "description"],
        "properties": {
          "time": {"type": "string"},
          "description": {"type": "string"}
        }
      }
    },
    "accessibilityFeatures": {"type": "array", "items": {"type": "string"}},
    "sourceAttribution": {"type": "string"},
    "metadataURI": {"type": "string"}
  }
}
```

### 4.2 Representative Payloads

```
// IMAGE
{
  "id": "sales-2024.png",
  "type": "image",
  "shortAlt": "Quarterly sales chart for 2024",
  "extendedDescription": "Bar chart of Q1-Q4 2024 sales: Q1 $1.2M, Q2 $1.5M, Q3 $1.9M, Q4 $2.3M. Blue",
  "tags": ["finance", "sales", "2024"],
  "dataPoints": [
    {"label": "Q1", "value": 1200000, "unit": "USD"},
    {"label": "Q2", "value": 1500000, "unit": "USD"},
    {"label": "Q3", "value": 1900000, "unit": "USD"},
  ]
}
```

```

    {"label": "Q4", "value": 2300000, "unit": "USD"}
  ]
}

// VIDEO
{
  "id": "onboarding-promo",
  "type": "video",
  "shortAlt": "Onboarding promo video with captions",
  "extendedDescription": "A promotional video demonstrating accessible onboarding.",
  "transcript": "Narrator: Welcome... [speaker IDs, tone]",
  "scenes": [
    {"time": "0:00-0:08", "description": "Logo animation with high-contrast colors"},
    {"time": "0:09-0:20", "description": "User navigates app with large touch targets"}
  ],
  "accessibilityFeatures": ["captions", "high-contrast"]
}

// UI ELEMENT
{
  "id": "delete-button-42",
  "type": "ui-element",
  "shortAlt": "Delete item",
  "extendedDescription": "Deletes the selected record. Disabled until the form is complete. Undo avail
}

```

## 5. Discovery, Transport, and Negotiation

D1 HTML authors SHOULD expose a discoverable pointer via one or more of: data-metadata-link, rel link, or aria-details. D2 Servers MAY advertise metadata via HTTP Link headers. D3 Clients MAY request JSON using Accept: application/eamp+json. D4 For decentralized URIs, a metadataURI field SHOULD provide an ipfs:// or arweave:// locator.

### 5.1 Discovery Examples

```
<!-- HTML link element -->
<link rel="accessibilityMetadata" href="https://ally.example.com/metadata/sales-2024" />

<!-- data-* hook on the element itself -->


# HTTP response header
Link: <https://ally.example.com/metadata/sales-2024>; rel="accessibilityMetadata"; type="application/

# Client content negotiation
GET /metadata/sales-2024
Accept: application/eamp+json
```

### 5.2 OpenAPI Fragment

```
openapi: 3.1.0
info: { title: EAMP Reference API, version: 1.0.0 }
paths:
  /metadata/{id}:
    get:
      summary: Get extended metadata
      parameters:
        - in: path
          name: id
          required: true
          schema: { type: string }
      responses:
        '200':
          description: OK
          content:
            application/eamp+json:
              schema:
                $ref: '#/components/schemas/EAMPMetadata'
        '404': { description: Not Found }
    components:
      schemas:
        EAMPMetadata:
          $ref: 'https://shaft.finance/schemas/eamp.metadata.schema.json'
```

### 5.3 Error Model & Versioning

```
// Error Envelope
{
  "error": {
    "code": "ValidationError",
    "message": "shortAlt exceeds 250 characters",
    "field": "shortAlt"
  }
}

// Versioning
- Add "eampVersion": "1.0.0" to payloads (recommended).
- Use SemVer; breaking changes require a new major.
```

## 6. Web3 Extensions (Informative)

W3-1 Smart contracts MAY expose an `accessibilityMetadataURI` for off-chain JSON. W3-2 Marketplaces SHOULD surface accessibility summaries to buyers. W3-3 DAOs SHOULD include plain-language and extended descriptions for proposals.

```
// Solidity interface example
interface IAccessible {
    function accessibilityMetadataURI() external view returns (string memory);
}

// Contract example
contract AccessibleNFT is IAccessible {
    string private _uri;
    constructor(string memory uri_) { _uri = uri_; }
    function accessibilityMetadataURI() external view returns (string memory) { return _uri; }
}

// Example IPFS JSON (application/ethereum+json)
{
    "id": "nft-001",
    "type": "image",
    "shortAlt": "Oil painting of a sunrise over mountains",
    "extendedDescription": "Warm orange and magenta hues illuminate a snow-capped ridge...",
    "tags": ["art", "landscape", "painting"],
    "sourceAttribution": "Artist: A. Example (2025)",
    "metadataURI": "ipfs://bafy..."
}
```

## 7. Localization & Internationalization

I18N-1 Payloads MAY include language tags (BCP 47) via Content-Language headers or per-field conventions (e.g., `shortAlt_en`, `shortAlt_es`). I18N-2 Authors SHOULD avoid machine-only jargon in short alternatives; technical detail belongs in `extendedDescription`. I18N-3 Multi-script content SHOULD indicate script variants if distinct.

```
// Multilingual example (per-field)
{
    "id": "chart-ml",
    "type": "image",
    "shortAlt_en": "Quarterly sales chart for 2024",
    "shortAlt_es": "Gráfico de ventas trimestrales de 2024",
    "extendedDescription_en": "Bar chart of Q1-Q4 2024 sales ...",
    "extendedDescription_es": "Gráfico de barras de las ventas Q1-Q4 de 2024 ..."
}
```

## 8. Performance, Caching & Security

P1 Large extendedDescription payloads SHOULD be cacheable with strong validators (ETag) and appropriate Cache-Control. P2 Servers SHOULD support range requests or pagination for very long transcripts. P3 Private metadata MUST require authorization; redact PII by default. P4 Integrity for decentralized objects SHOULD use content addressing (IPFS CID) and optional signatures.

```
# HTTP caching
ETag: "W/"ally-6f2""
Cache-Control: public, max-age=86400

# Pagination (example)
GET /metadata/onboarding-promo?section=scenes&offset=0&limit=100
```

## 9. WCAG Mapping (Informative)

WCAG SC	How EAMP Helps
1.1.1 Non-text Content	L1 shortAlt + L2 extendedDescription & dataPoints
1.2.x Time-based Media	L1 captions + L2 transcript & scenes
2.4.x Navigable	Clear labels; L2 context for complex widgets
3.1.x Readable	Plain-language L1; detailed L2
4.1.x Compatible	Programmatically determinable roles/states; JSON structure

## 10. Conformance Profiles

Profile A (Baseline): Implements Layer 1 fully (shortAlt, captions, labels, transcripts); exposes at least one discovery mechanism. Profile B (Extended): All of Profile A plus Layer 2 JSON for complex content. Profile C (Advanced): Profile B plus localization, caching, and Web3 integration.