Operator-published CT Log Metadata

Operator-published CT Log Metadata

Currently, managing Certificate Transparency (CT) Log Lists is a largely manual process, relying on ad-hoc communication between Log Operators and CT Programs. This approach can be labor-intensive, prone to error, and can introduce delays in updating the status of logs within the ecosystem. To address these challenges, we propose a standardized, machine-readable format that allows Log Operators to publish their own metadata in a predictable and automated way.

This model consists of two (fairly) simple JSON schemas. The first, an “operator-list.json” file, serves as a central directory for a given operator, pointing to the metadata URLs for each individual log they run. The second, a “log-metadata.json” file, exists for each log and provides a comprehensive, single source of truth for all of its technical and operational details. By hosting these files at a stable endpoint, Log Operators can provide real-time data that Log Programs can consume automatically, eliminating manual data entry and email-based coordination.

The high-level expected workflow for this new data would be:

1. A new log operator submits a request for inclusion with Log Programs, as they do today, through the Program-specific method (e.g. a Chromium Issue or an email to certificate-transparency-program[at]group.apple.com).
    a. The submission would include the “operator-list.json” endpoint they will use to publish all logs which they are operating.
2. Once a log operator has been added by a Log Program, the Log Program will monitor and consume data from the “operator-list.json” to identify relevant logs, parsing the individual “log-metadata.json” files to determine which logs should be added to the Log Program’s Log List.
    a. The same lifecycle of logs would be followed as occurs today (at least initially – maybe there will be opportunity to update this in the future).

And that’s basically it. In theory, we would obviate the need for almost all out-of-band communication related to log lifecycle management (though incidents related to log operations would likely still require some manual touch points).

It is important to note that the information published by operators MUST NOT be used as a stand-in for the actual status of logs from the perspective of Log Operators; the Log Lists will remain the source of truth regarding what logs are ‘usable’, ‘retired’, etc. The two are entirely distinct, though complementary, sources of information about different parts of the CT ecosystem.

We believe adopting this standardized format will offer significant advantages for the entire CT ecosystem:

- Automation and Timeliness

    - It enables the full automation of log list management, drastically reducing the manual work required by CT Programs. This ensures that new logs can be included more quickly and that changes to a log's status are reflected in a timely manner.
- Unambiguous Operational Transparency
    - The schema provides clear, real-time insight into a log's lifecycle. With distinct fields for intended_use (production, test, retired), endpoint status (active/inactive), and even a planned_change object for future updates, all participants can have a precise understanding of a log operator's view into their logs' current and future state.
- Consistency and Predictability
    - A single, well-defined data structure for all operators reduces ambiguity. Whether a log is a standard rfc6962 type or a tiled log, its metadata will be presented in a consistent and predictable way across the ecosystem.
- Verifiable Log Retirement
    - The inclusion of a final_tree_head object for retired logs introduces a verifiable, cryptographic mechanism to confirm a log's final state, from the perspective of the log operator.
- Richer Data for Ecosystem Participants
    - The proposal (hopefully) makes it easy to provide valuable, but optional metadata like "log_software" and endpoint-specific "rate_limit" information. This data can be invaluable for CT Monitors, researchers, and other parties analyzing the health and behavior of the log ecosystem.

By providing a single, authoritative source of truth for each log, these schemas pave the way for a more dynamic, reliable, and transparent Certificate Transparency ecosystem. We encourage all CT Log Operators, Monitors, and Log Programs to review this proposal and consider the collective benefits of its adoption. **Please provide any comments here or on CT-Policy.**

Operator List Schema

JSON

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "CT Log Operator List",
  "description": "A lightweight manifest published by a CT Log
Operator, listing the metadata URLs for each log (or log family) they
operate. CT Programs consume this file to discover individual log
metadata without requiring out-of-band communication.",
```

```
  "type": "object",
  "properties": {
    "$schema": {
      "description": "A URI reference to the schema that this document
conforms to.",
      "type": "string",
      "format": "uri"
    },
    "operator_name": {
      "description": "The human-readable name of the CT Log
Operator.",
      "type": "string"
    },
    "logs": {
      "description": "An array of URLs, each pointing to the
log-metadata.json file for a log (or log family) operated by this
operator. This list only needs to change when a new log is added.",
      "type": "array",
      "items": {
        "type": "string",
        "format": "uri"
      }
    }
  },
  "required": [
    "operator_name",
    "logs"
  ]
}
```

Example Operator List JSON

```json
{
  "$schema":
"https://www.certificate-transparency.org/schemas/operator-list-v1.jso
n",
  "operator_name": "Cloudflare",
  "logs": [
    "https://ct.cloudflare.com/logs/nimbus2026/metadata.json",
    "https://ct.cloudflare.com/logs/raio2027/metadata.json",
    "https://ct.cloudflare.com/logs/raio2025/metadata.json"
  ]
}
```

CT Log Metadata Schema

```json
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "CT Log Metadata",
  "description": "Detailed metadata for a single Certificate
Transparency log, as published by the log operator.",
  "type": "object",
  "properties": {
    "$schema": {
      "description": "A URI reference to the schema that this document
conforms to.",
      "type": "string",
      "format": "uri"
```

```
    },
    "log_id": {
      "description": "The SHA-256 hash of the log's public key, base64
encoded.",
      "type": "string",
      "pattern":
"^([A-Za-z0-9+/]{4})*([A-Za-z0-9+/]{3}=|[A-Za-z0-9+/]{2}==)?$"
    },
    "key": {
      "description": "The log's public key, base64 encoded
(DER-encoded SubjectPublicKeyInfo).",
      "type": "string",
      "pattern":
"^([A-Za-z0-9+/]{4})*([A-Za-z0-9+/]{3}=|[A-Za-z0-9+/]{2}==)?$"
    },
    "friendly_name": {
      "description": "A brief, human-readable name for the log.",
      "type": "string"
    },
    "log_spec": {
      "description": "The protocol implemented by this log. 'rfc6962'
for RFC 6962 logs; 'static-ct-api' for logs implementing the C2SP
Static CT API specification.",
      "type": "string",
      "enum": [
        "rfc6962",
        "static-ct-api"
      ]
    },
```

```
    "mmd_seconds": {
      "description": "The Maximum Merge Delay in seconds: the maximum time the log promises to incorporate a submitted certificate into the tree.",
      "type": "integer",
      "minimum": 0
    },
    "intended_use": {
      "description": "The operator's declared intended use for this log. 'production' logs are eligible for inclusion in CT programs. 'test' logs are for operator or ecosystem testing and should not be included. 'decommissioned' logs have been permanently shut down by the operator (distinct from 'retired', which is a status that can only be designated by a UA).",
      "type": "string",
      "enum": [
        "production",
        "test",
        "decommissioned"
      ]
    },
    "tls_only": {
      "description": "If true, this log only accepts certificates containing the ServerAuth EKU. If false (or absent), the log accepts certificates regardless of EKU.",
      "type": "boolean"
    },
    "temporal_interval": {
```

```
      "description": "The window of certificate expiry dates this log
accepts.",
      "type": "object",
      "properties": {
        "start_inclusive": {
          "type": "string",
          "format": "date-time"
        },
        "end_exclusive": {
          "type": "string",
          "format": "date-time"
        }
      },
      "required": [
        "start_inclusive",
        "end_exclusive"
      ]
    },
    "status": {
      "description": "The overall operational status of the log as
understood by the operator. 'active' \u2014 the log is accepting
submissions and serving data. 'readonly' \u2014 submissions are closed
but monitoring/tile endpoints remain available (e.g. after a log's
temporal window closes). 'inactive' \u2014 all endpoints are offline.
This field reflects the operator's intended state, not real-time
availability.",
      "type": "string",
      "enum": [
        "active",
```

```
        "readonly",
        "inactive"
      ]
    },
    "status_timestamp": {
      "description": "The timestamp (ISO 8601) when the current
top-level status was last set.",
      "type": "string",
      "format": "date-time"
    },
    "planned_changes": {
      "description": "An ordered list of planned future status
transitions for this log (e.g. going read-only, then fully inactive).
Consumers can use this to prepare for upcoming changes without
out-of-band communication.",
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "new_status": {
            "description": "The status the log will transition to.",
            "type": "string",
            "enum": [
              "active",
              "readonly",
              "inactive"
            ]
          },
          "effective_date": {
```

```
            "description": "The date/time (ISO 8601) at which the
status change is expected to take effect.",
            "type": "string",
            "format": "date-time"
          },
          "comment": {
            "description": "A human-readable explanation of this
planned change.",
            "type": "string"
          }
        },
        "required": [
          "new_status",
          "effective_date"
        ]
      }
    },
    "final_tree_head": {
      "description": "For decommissioned logs: the cryptographically
verifiable final state of the log tree, as published by the
operator.",
      "type": "object",
      "properties": {
        "sha256_root_hash": {
          "type": "string",
          "pattern":
"^([A-Za-z0-9+/]{4})*([A-Za-z0-9+/]{3}=|[A-Za-z0-9+/]{2}==)?$"
        },
        "tree_size": {
```

```
          "type": "integer",
          "minimum": 0
        }
      },
      "required": [
        "sha256_root_hash",
        "tree_size"
      ]
    },
    "log_software": {
      "description": "Optional information about the software powering
this log.",
      "type": "object",
      "properties": {
        "name": {
          "type": "string"
        },
        "version": {
          "type": "string"
        }
      },
      "required": [
        "name"
      ]
    }
  },
  "required": [
    "log_id",
    "key",
```

```
    "friendly_name",
    "log_spec",
    "mmd_seconds",
    "intended_use",
    "temporal_interval",
    "status",
    "status_timestamp"
  ],
  "if": {
    "properties": {
      "log_spec": {
        "const": "rfc6962"
      }
    }
  },
  "then": {
    "properties": {
      "endpoint": {
        "$ref": "#/definitions/EndpointInfo"
      }
    },
    "required": [
      "endpoint"
    ]
  },
  "else": {
    "properties": {
      "submission_endpoint": {
        "$ref": "#/definitions/EndpointInfo"
```

```
      },
      "monitoring_endpoint": {
        "$ref": "#/definitions/EndpointInfo"
      }
    },
    "required": [
      "submission_endpoint",
      "monitoring_endpoint"
    ]
  },
  "definitions": {
    "EndpointInfo": {
      "description": "URL and optional rate limit information for a
single log endpoint. Operational status is expressed at the top-level
log object, not per endpoint.",
      "type": "object",
      "properties": {
        "url": {
          "type": "string",
          "format": "uri"
        },
        "rate_limit": {
          "description": "Optional rate limit information for this
endpoint.",
          "type": "object",
          "properties": {
            "requests": {
              "description": "Maximum number of requests permitted
within the interval.",
```

```
              "type": "integer",
              "minimum": 0
            },
            "interval_seconds": {
              "description": "The time window in seconds over which
the request count applies.",
              "type": "integer",
              "minimum": 1
            },
            "scope": {
              "description": "The dimension against which this rate
limit is applied.",
              "type": "string",
              "enum": [
                "ip",
                "asn",
                "subnet",
                "user-agent",
                "geography",
                "global"
              ]
            }
          },
          "required": [
            "requests",
            "interval_seconds"
          ]
        }
      },
```

```
      "required": [
        "url"
      ]
    }
  }
}
```

Example 6962 Log

JSON

```
{
  "$schema":
"https://www.certificate-transparency.org/schemas/log-metadata-v1.json
",
  "log_id": "someOtherLogIdBase64EncodedString=",
  "key":
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEdummykeyfornimbus2026logthatisver
ylongandvalidbase64encoded==",
  "friendly_name": "nimbus2026",
  "log_spec": "rfc6962",
  "mmd_seconds": 86400,
  "intended_use": "production",
  "tls_only": false,
  "temporal_interval": {
    "start_inclusive": "2026-01-01T00:00:00Z",
    "end_exclusive": "2027-01-01T00:00:00Z"
  },
  "status": "active",
```

```
  "status_timestamp": "2025-06-01T00:00:00Z",
  "planned_changes": [
    {
      "new_status": "readonly",
      "effective_date": "2026-12-31T23:59:59Z",
      "comment": "This log will stop accepting new certificate
submissions at the end of its operational year, but will remain
available for monitoring."
    },
    {
      "new_status": "inactive",
      "effective_date": "2027-06-30T00:00:00Z",
      "comment": "All endpoints will be taken offline mid-year
following the submission window closure."
    }
  ],
  "endpoint": {
    "url": "https://ct.cloudflare.com/logs/nimbus2026/",
    "rate_limit": {
      "requests": 1000,
      "interval_seconds": 60,
      "scope": "ip"
    }
  },
  "log_software": {
    "name": "CloudyMcCloudface",
    "version": "2.1.0"
  }
}
```

Example Tiled CT Log Metadata - Pre-Production Log

JSON


```
{
  "$schema":
"https://www.certificate-transparency.org/schemas/log-metadata-v1.json
",
  "log_id": "someOtherLogIdBase64EncodedString=",
  "key":
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEdummykeyfornimbus2026logthatisver
ylongandvalidbase64encoded==",
  "friendly_name": "nimbus2026",
  "log_spec": "static-ct-api",
  "mmd_seconds": 86400,
  "intended_use": "test",
  "tls_only": true,
  "temporal_interval": {
    "start_inclusive": "2026-01-01T00:00:00Z",
    "end_exclusive": "2027-01-01T00:00:00Z"
  },
  "status": "active",
  "status_timestamp": "2025-06-01T00:00:00Z",
  "planned_changes": [
    {
      "new_status": "inactive",
      "effective_date": "2026-12-31T23:59:59Z",
      "comment": "This test log will be taken offline at the end of
its operational year."
    }
```

```
  ],
  "submission_endpoint": {
    "url": "https://ct.cloudflare.com/logs/nimbus2026/",
    "rate_limit": {
      "requests": 1000,
      "interval_seconds": 60,
      "scope": "ip"
    }
  },
  "monitoring_endpoint": {
    "url": "https://ct.cloudflare.com/logs/nimbus2026/",
    "rate_limit": {
      "requests": 5000,
      "interval_seconds": 60,
      "scope": "ip"
    }
  },
  "log_software": {
    "name": "CloudyMcCloudface",
    "version": "1.3.0"
  }
}
```

Example Tiled CT Log Metadata - Decommissioned Log

JSON

```
{
```

```
  "$schema":
"https://www.certificate-transparency.org/schemas/log-metadata-v1.json
",
  "log_id": "logIdForRetiredRaio2025LogHere=",
  "key":
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEoldkeyforraio2025logthatisnowreti
redandvalidbase64encoded==",
  "friendly_name": "raio2025",
  "log_spec": "static-ct-api",
  "mmd_seconds": 60,
  "intended_use": "decommissioned",
  "tls_only": true,
  "temporal_interval": {
    "start_inclusive": "2025-01-01T00:00:00Z",
    "end_exclusive": "2026-01-01T00:00:00Z"
  },
  "status": "readonly",
  "status_timestamp": "2026-01-01T00:00:00Z",
  "planned_changes": [
    {
      "new_status": "inactive",
      "effective_date": "2026-07-01T00:00:00Z",
      "comment": "The monitoring endpoint will be taken offline once
archival is confirmed complete."
    }
  ],
  "final_tree_head": {
    "sha256_root_hash":
"eJj4IHvdYpljVsW/YCery+QsSRHbuYBME7H912a5P2Y=",
```

```
    "tree_size": 1213760846
  },
  "submission_endpoint": {
    "url": "https://ct.cloudflare.com/logs/raio2025/submit"
  },
  "monitoring_endpoint": {
    "url": "https://raio2025.ct.cloudflare.com/monitor",
    "rate_limit": {
      "requests": 2000,
      "interval_seconds": 60,
      "scope": "ip"
    }
  },
  "log_software": {
    "name": "CloudyMcCloudface",
    "version": "1.9.8"
  }
}
```