World Transcription Format (WTF) Extension for vCon

A standardized approach to transcription data in conversational containers

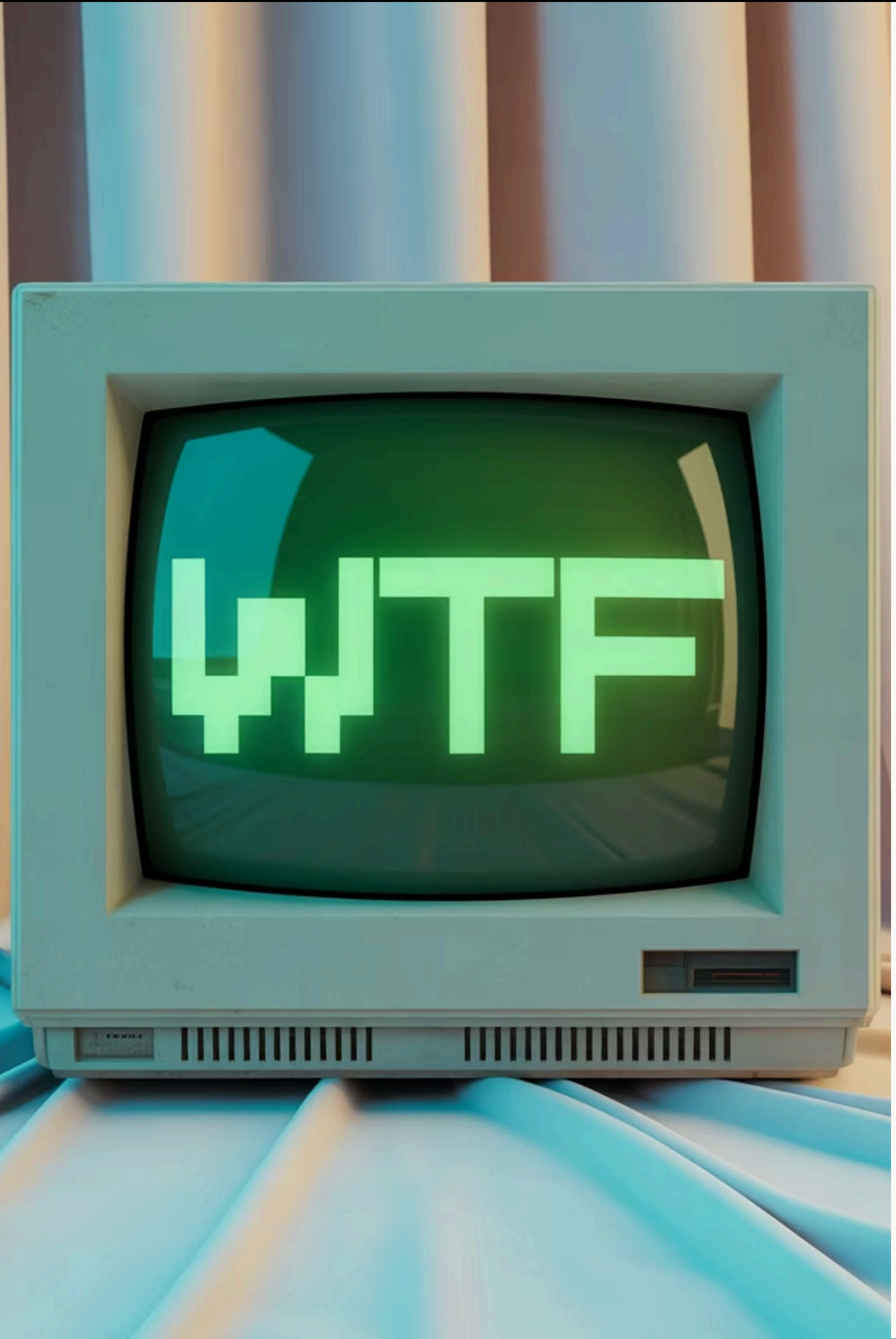
https://datatracker.ietf.org/doc/draft-howe-vcon-wtf-extension/

Thomas Howe, VCONIC

The Transcription Standardization Problem

Each transcription provider delivers data in unique, proprietary formats with incompatible scores and inconsistent timestamps. This makes integration complex, migrations and operations costly, and fair comparison impossible.

Fortunately, these formats aren't very very different.



World Transcription Format

WTF operates as a VCON **Compatible Extension** that introduces a new analysis type called "wtf-transcription" without modifying existing vCon semantics, ensures backward compatibility while enabling powerful new capabilities.

Integration Point

Sits inside vCon's analysis array

Normalization Layer

Provider data normalization occurs at conversion time, with original output optionally retained in extensions field

Backward Compatible

Existing vCon implementations safely ignore WTF data while continuing normal processing—no breaking changes

Wide Compatibility

Currently functionally supported providers include Whisper, Deepgram, AssemblyAI, Google Cloud Speech, Amazon Transcribe, Azure Speech Services, Rev.ai, Speechmatics, Wav2Vec2, and Parakeet.

WTF Required Layers: Foundation Structure

WTF employs a hierarchical model with **3 required layers** and **7 optional layers** to balance standardization with flexibility.

Metadata

Provider identification, model version, timestamps, and audio properties

Transcript

Full text, language (BCP-47), duration, and overall confidence score

Segments

Time-aligned chunks with per-segment confidence and speaker attribution

MetaData Object

The metadata object captures processing and source information

```
"metadata": {
  "created_at": "string",       // ISO 8601 timestamp
  "processed_at": "string",      // ISO 8601 timestamp
  "provider": "string",          // Provider name (lowercase)
  "model": "string",             // Model/version identifier
  "processing_time": "number",   // Optional: Processing duration in seconds
  "audio": {
    "duration": "number",        // Source audio duration in seconds
    "sample_rate": "integer",    // Optional: Sample rate in Hz
    "channels": "integer",       // Optional: Number of channels
    "format": "string",          // Optional: Audio format
    "bitrate": "integer"         // Optional: Bitrate in kbps
  },
  "options": "object"            // Provider-specific options used
}
```

Transcript Object

The transcript object provides the high-level summary of the entire transcription:

```
"transcript": {
  "text": "string",       // Complete transcription text
  "language": "string",   // BCP-47 language code (e.g., "en-US")
  "duration": "number",   // Total audio duration in seconds
  "confidence": "number"  // Overall confidence score \[0.0-1.0\]
}
```

Segments Array

Segments represent logical chunks of transcribed content, typically sentence or phrase boundaries, but could refer to the entirety.

```
 "segments": [
 {
  "id": "integer",             // Sequential segment identifier
  "start": "number",           // Start time in seconds
  "end": "number",             // End time in seconds
  "text": "string",            // Segment text content
  "confidence": "number",      // Segment-level confidence \[0.0-1.0\]
  "speaker": "integer|string",  // Optional: Speaker identifier
  "words": ["integer"]         // Optional: Array of word indices
 }
]
```

WTF Optional Layers: Extended Capabilities

While WTF defines core required layers, its power truly expands with **7 optional layers**, offering enhanced capabilities and flexible integration for diverse transcription needs.

Optional Layers

Words

Word-level confidence and precise timing

Speakers

Diarization data for multi-party conversations

Alternatives

N-best hypothesis alternatives

Enrichments

Sentiment, named entities, analysis

Extensions

Provider-specific features preserved

Quality

Audio quality metrics and warnings

Streaming

Incremental delivery markers

Words Array

```
"words": [
 {
  "id": "integer",           // Sequential word identifier
  "start": "number",         // Word start time in seconds
  "end": "number",           // Word end time in seconds
  "text": "string",          // Word text
  "confidence": "number",    // Word-level confidence \[0.0-1.0\]
  "speaker": "integer|string",  // Optional: Speaker identifier
  "is_punctuation": "boolean"   // Optional: Punctuation marker
 }
```

Speakers Object

```
"speakers": {
  "speaker_id": {
    "id": "integer|string",       // Speaker identifier
    "label": "string",            // Human-readable speaker name
    "segments": ["integer"],      // Array of segment IDs for this speaker
    "total_time": "number",       // Total speaking time in seconds
    "confidence": "number"        // Diarization confidence \[0.0-1.0\]
  }
}
```

Quality Object

```
"quality": {
  "audio_quality": "string",      // high, medium, low
  "background_noise": "number",   // Noise level \[0.0-1.0\]
  "multiple_speakers": "boolean",
  "overlapping_speech": "boolean",
  "silence_ratio": "number",      // Percentage of silence
  "average_confidence": "number",
  "low_confidence_words": "integer",
  "processing_warnings": ["string"]
}
```

The Conversion Pipeline: Five Normalization Steps

Confidence Normalization

Convert all confidence scores to standardized [0.0, 1.0] range regardless of provider scale

Timestamp Standardization

Convert all timestamps to floating-point seconds from various precision levels

Language Code Mapping

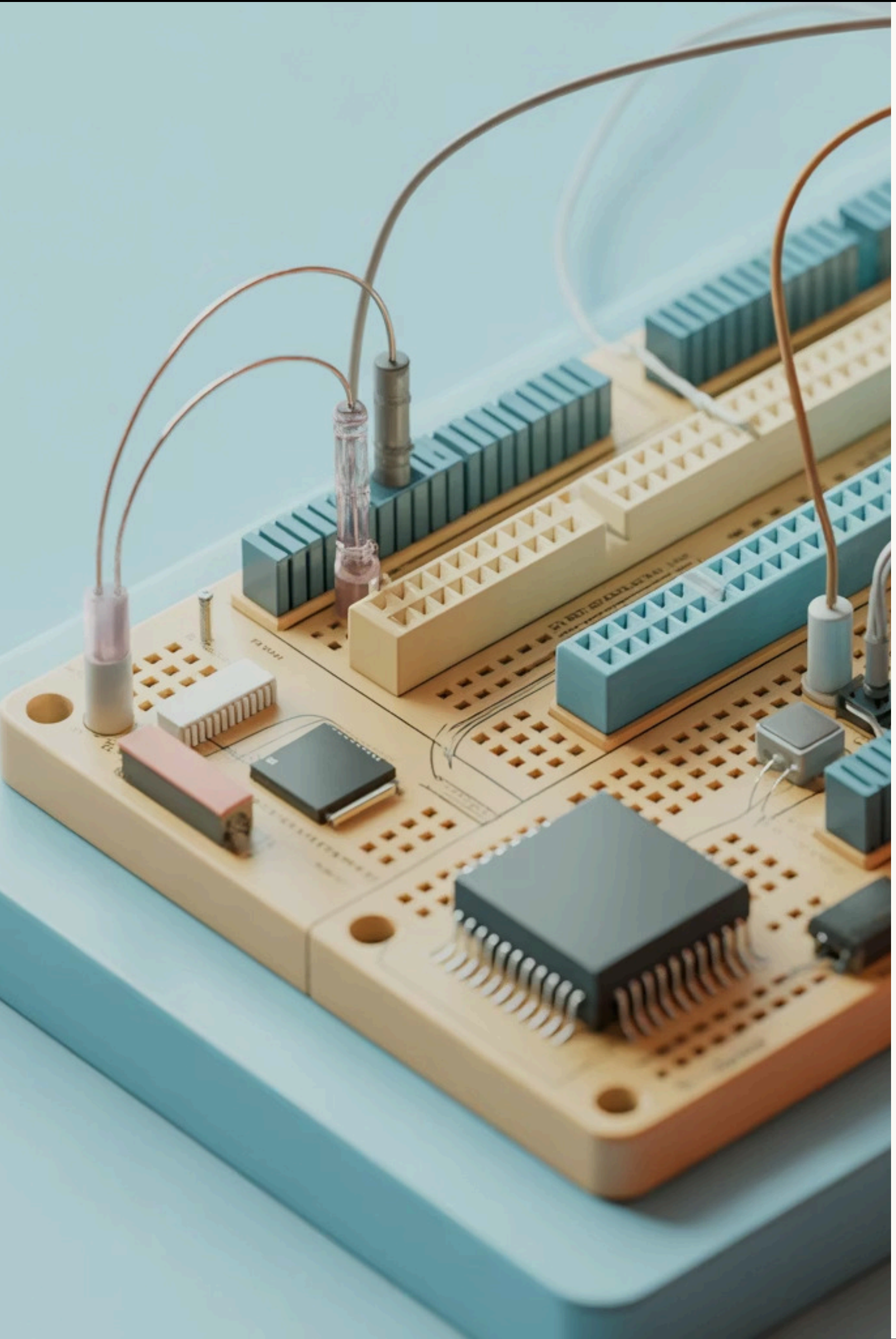
Standardize language identifiers to BCP-47 format

Boundary Alignment

Map segment and word boundaries to hierarchical structure

Feature Preservation

Retain provider-specific data in extensions field



Reference Implementation: wtf-transcript-converter

Bidirectional Conversion

Transformation between provider formats and WTF in both directions

Multi-Provider Support

Integrated support for 6 providers: Whisper, Deepgram, AssemblyAI, Rev.ai, Canary, and Parakeet

Schema Validation

Validation against WTF schema specifications for compliance

Testing Framework

Cross-provider testing for consistency, performance benchmarking, and quality metrics

CLI Interface

Command-line tools for batch processing and automation workflows

Extensible Architecture

Modular provider system simplifies adding new transcription services

Open Technical Questions for Working Group Discussion

As we progress toward formal adoption, we're seeking Working Group guidance on several key design decisions that will shape WTF's future direction and adoption patterns.

Confidence Score Representation

Is the [0, 1] range sufficient for all use cases, or should we support provider-specific ranges alongside normalized values?

Optional Enrichment Scope

Should sentiment analysis, named entity recognition, and other analysis remain in WTF scope, or be delegated to separate extensions?

Speaker Diarization

Is the current speakers object adequate for complex multi-party conversations, or do we need richer speaker metadata structures?

Streaming Support

Does the proposed streaming field adequately address incremental delivery requirements for real-time transcription scenarios?

IANA Provider Registry

Should we maintain a registry of supported providers as standardized values, or keep provider identification open-ended?

Read the specification at

https://datatracker.ietf.org/doc/draft-howe-vcon-wtf-extension/

Download your own converter at

https://github.com/vcon-dev/wtf-transcript-converter