The AI-Ready DITA Pipeline

MODELING, METADATA, AND MACHINE UNDERSTANDING

with Rob Hanna, CEO Precision Content

CIDM

ConVEx

April 13-15, 2026

Pittsburgh

precision content

shaping the future of content

Shaping the future of content

Experts in DITA and intelligent content delivery

We're a full-service, end-to-end technical communications consultancy, technology innovator, and systems integrator offering professional services, training, and tools.

Areas of Expertise

Precision Content is home to thought leaders and expertise in the areas of

- DITA/XML design and implementation
- structured authoring methods
- content lifecycle management
- information architecture
- content strategy,
- and structured content delivery.

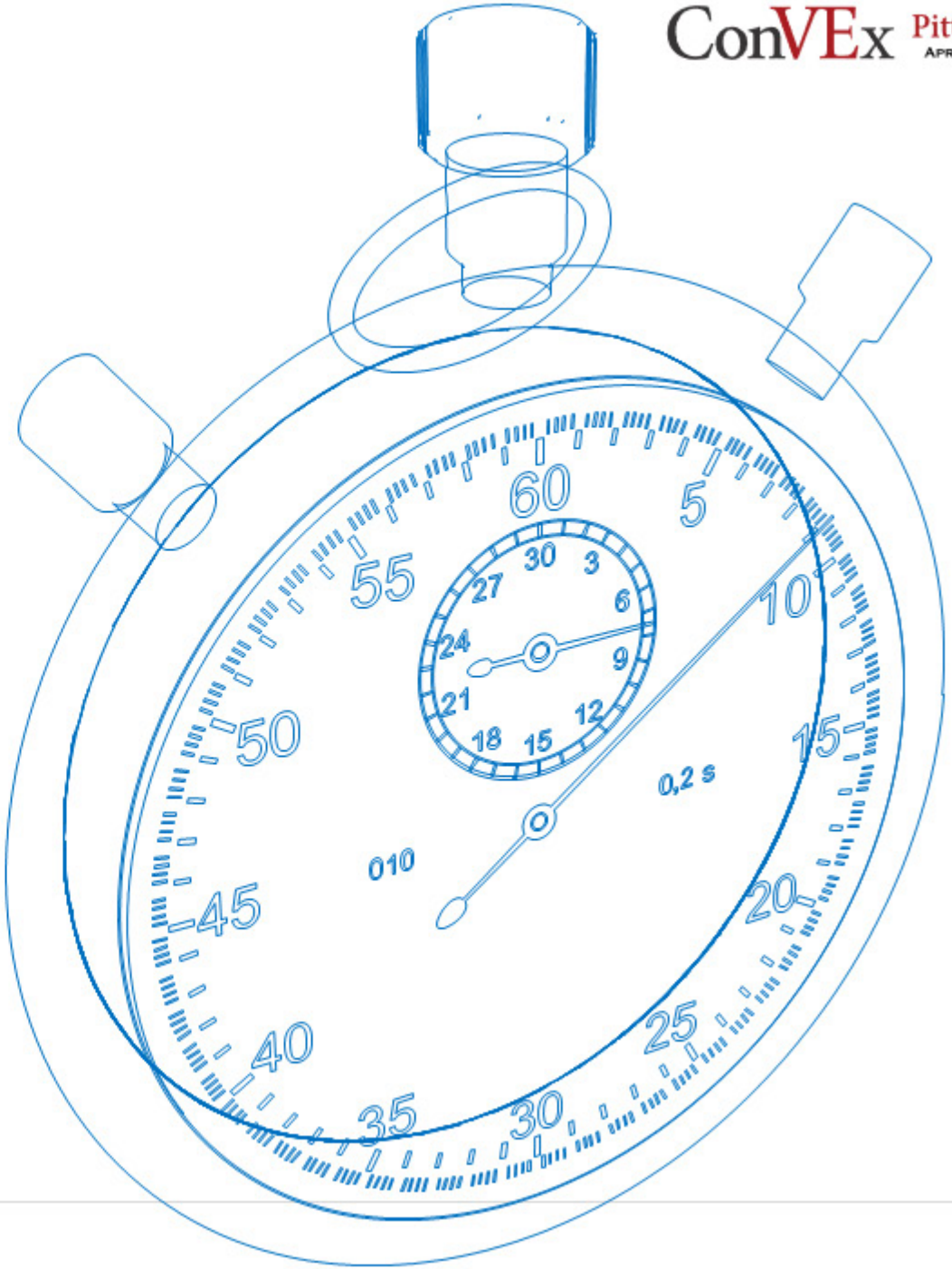
Best Workplaces™
Great Place To Work.
CANADA 2025

Presentation agenda

How to adapt DITA architectures, taxonomies, metadata strategies, and publishing streams for GenAI systems.

How to adjusting content granularity, semantic annotation, classification structures, and output transformations for

- retrieval accuracy
- hallucination resistance, and
- contextual generation quality.



What's changing the conversation?

The familiar **"garbage in, garbage out"** warning is still true.

What changed is that it no longer slows the program down.

Product velocity

- AI is accelerating product change, so support content has to scale safely.

Support economics

- Self-service and assistant channels are now core expectations, not experiments.

Executive pressure

- Organizations want visible AI outcomes sooner, even while content maturity lags.

The opportunity for content teams is to replace a defensive argument with a constructive one: **structured content is the fastest route to more reliable AI.**

From publishing pipeline to content supply chain

The content supply chain question is not how to push more text through the pipe. It is how to make the pipe scalable, reliable, and reusable across channels.

Returning reliable responses

Assuming well-developed sources, prompts, and agents:

Generative AI

- *For basic understanding based on large language model*
- Lowest cost $ / Highest propensity for hallucination ?????

RAG (Retrieval Augmented Generation)

- *For answers when a specific source must be cited*
- Slightly higher cost $$ / Lesser propensity for hallucination ???

Agentic Retrieval

- *For research that can supplement multiple cited sources with outside sources*
- Highest cost $$$$$ / Least propensity for hallucination ??

Chunking challenges with RAG solutions

- **Fixed-size chunking** is simple and fast, but can split ideas in awkward places
- **Sentence or paragraph-based chunking** keeps natural boundaries better
- **Structure-based chunking** works well for documents with clear headers and sections
- **Semantic chunking** tries to group content by meaning, not just by length

What does the solution look like?

How do we create a stable, scalable content supply chain?

The Content Triforce

Standards for rapid, reliable deployment

Content model standards

- ✓ DITA

Metadata model standards

- ✓ iiRDS

Writing model standards

- ? Microcontent practices

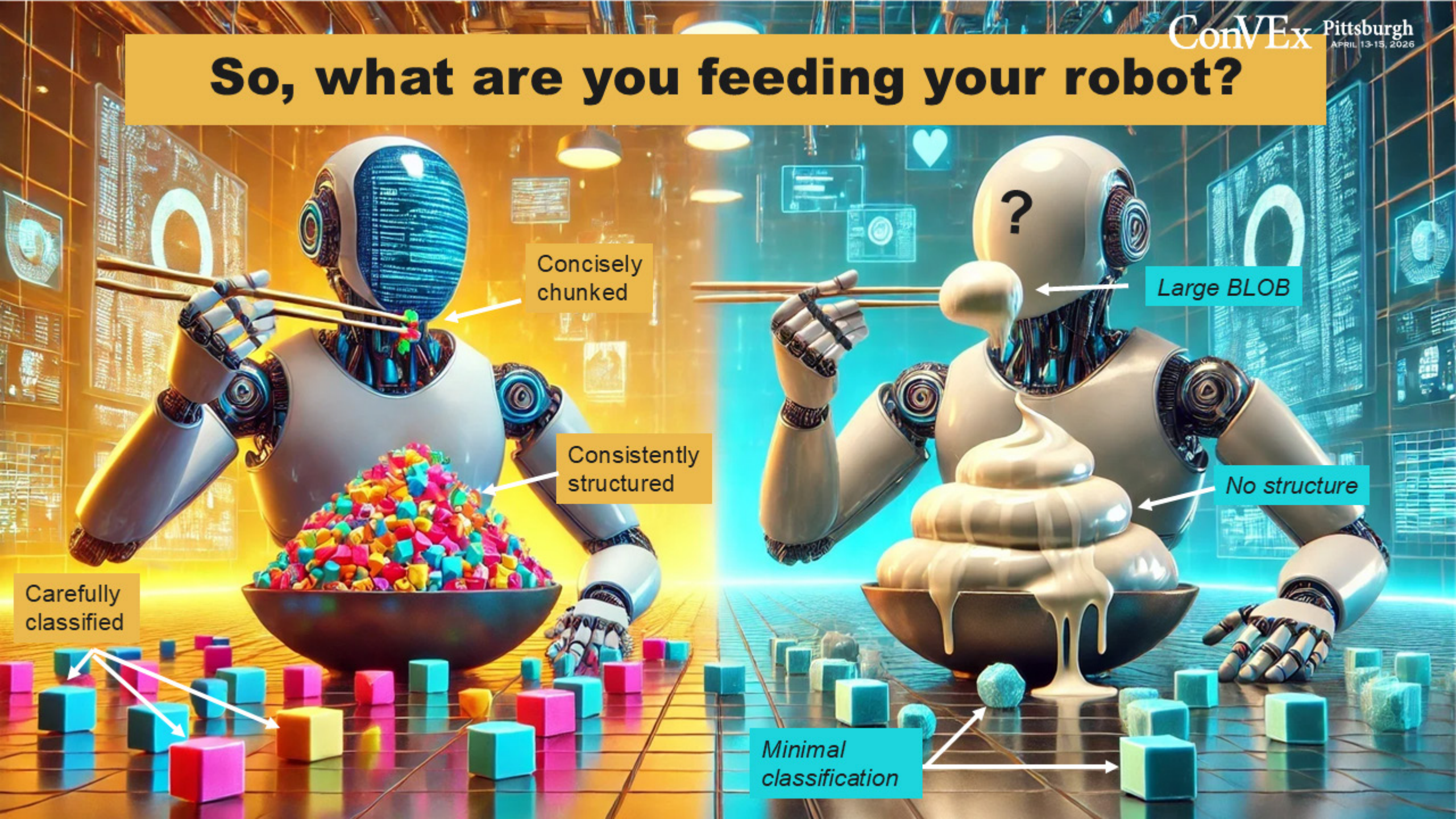
DITA is ideally suited to AI-ready content

Because it already encodes distinctions that retrieval systems need.

Topic orientation, typing, reuse, metadata, and extensible transforms all reduce ambiguity before generation begins.

That makes DITA not just publishable, but indexable, rankable, and citeable.

So, what are you feeding your robot?



Challenges of feeding robots raw DITA

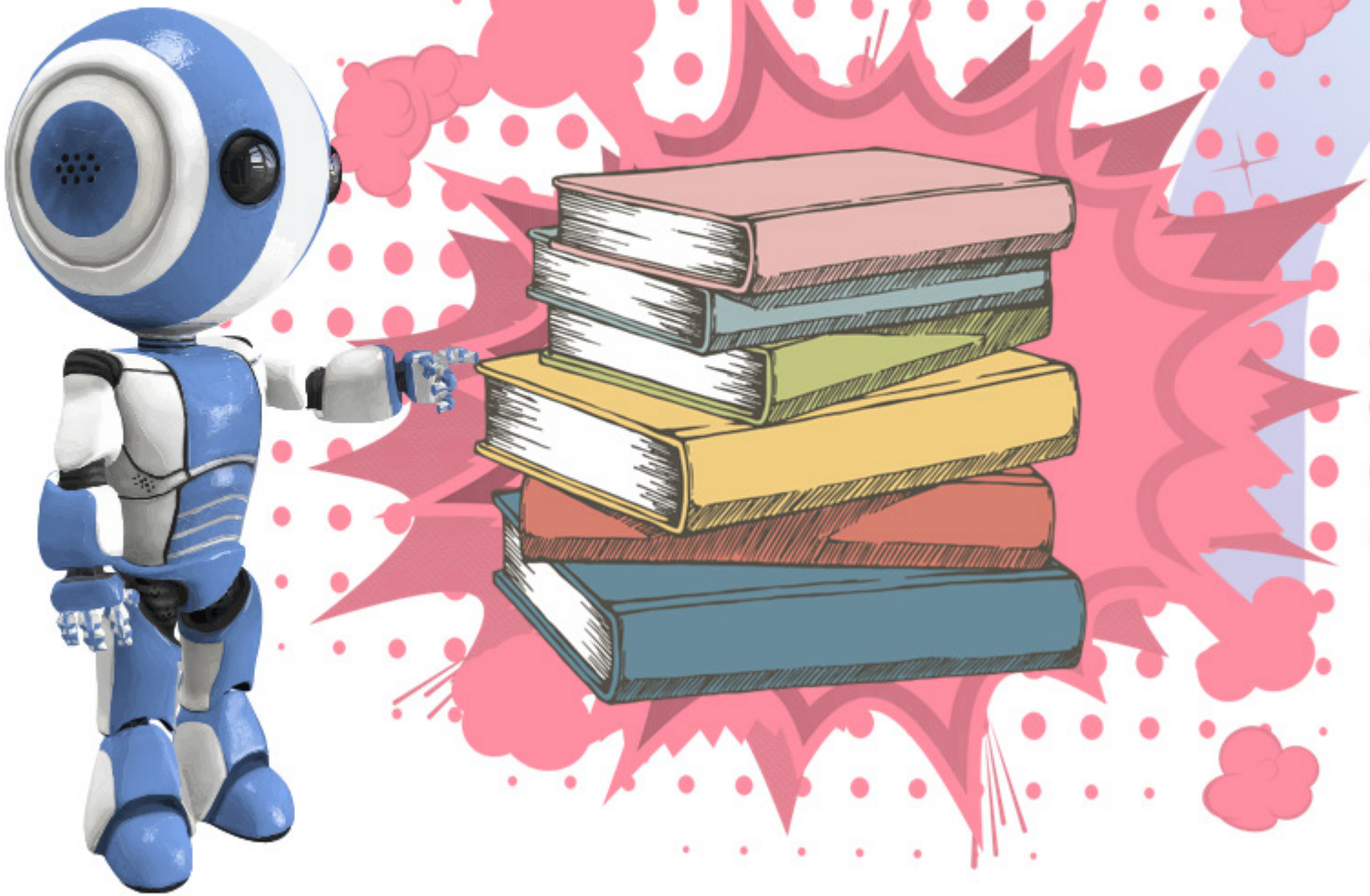
- **Granularity:** Topics can be long and not easily parsed
- **Reuse:** Can distort probabilistic responses
- **Metadata:** Not enough context or too much noise
- **Conditional Markup:** Confuses ingestion
- **Semantic Confusion:** Incorrect or irrelevant tagging

Best practices for feeding your robot

- Source Format
  - **Chunking = Topics/Maps**
    - DITA/XML
  - **Metadata = System, publication, map, topic, element-level**
    - DB/XML
- Response/Delivery Format
  - **Chunking = Microcontent**
    - Markdown
  - **Metadata = iiRDS**
    - JSON

Context from the Source

Traditional documents we write contain an abundance of metadata at the CCMS level, book level, map level, topic level, and element level. This metadata is naturally inherited from each higher level.



Bursting documents into fragments of microcontent and wrapping each piece in metadata provides for rich context to serve it up dynamically.

Sample PCDITA Chunker output

PCDITA Reference Topic

```
<?xml version="1.0" encoding="UTF-8"?>
<reference class="- topic/topic reference/reference
" id="reference_9589">
  <title class="- topic/title ">Employee referral
  bonus pay schedule </title>
  <primaryblock class="+ topic/abstract
  prec-d/primaryblock ">
    <shortdesc class="- topic/shortdesc ">Employees
    will be paid 50% of the bonus at the time of
    the new hire and the other 50% once the new
    hire has successfully completed their probation
    period.</shortdesc>
  </primaryblock>
</reference>
```

JSON Chunk


```
{
  "context_title": "Employee referral policy",
  "context_id": "reference_9671",
  "context_description": "",
  "text": "[Reference: Employee referral bonus pay schedule]
  Employees will be paid 50% of the bonus at the time of the
  new hire and the other 50% once the new hire has
  successfully completed their probation period.",
  "context": "reference",
  "metadata": {
    "context_title": "Employee referral policy",
    "context_id": "reference_9671",
    "context_description": "",
    "context": "reference",
    "this": "Reference"
  },
  "id": "reference_9589"
}
```


DITA OT Plug-in also produces markdown and graphRAG variants.

METADATA

standardizing how you

describe your content

Organizations struggle with metadata because it is

- not planned properly
- not clearly described, and
- inflexible or proprietary.

what is it?

what is it about?

who is it for?

The metadata “naming nightmare”

iiRDS standardized metadata

tekom iiRDS

The international standard for IIRDS (Intelligent Information Request and Delivery) is sponsored by Tekom in Germany in response to the growth of Industry 4.0 and the Industrial Internet of Things (IoT).

iiRDS is envisioned to serve the needs of highly-componentized manufacturing where many individual hardware and software components need to be brought online and work together.

what the iiRDS standard provides

- Common model
- Interoperability
- Guidance

Concepts, properties, and relationships

Common, structured format for data exchange across platforms

Rules and data formats, plus documentation

what the iiRDS standard provides

- Common model
- Interoperability
- Guidance

what are the benefits?

- Less development
- Easier movement
- Faster adoption

WHAT IF you had a metadata standard that allowed you to describe your content with **precision**?

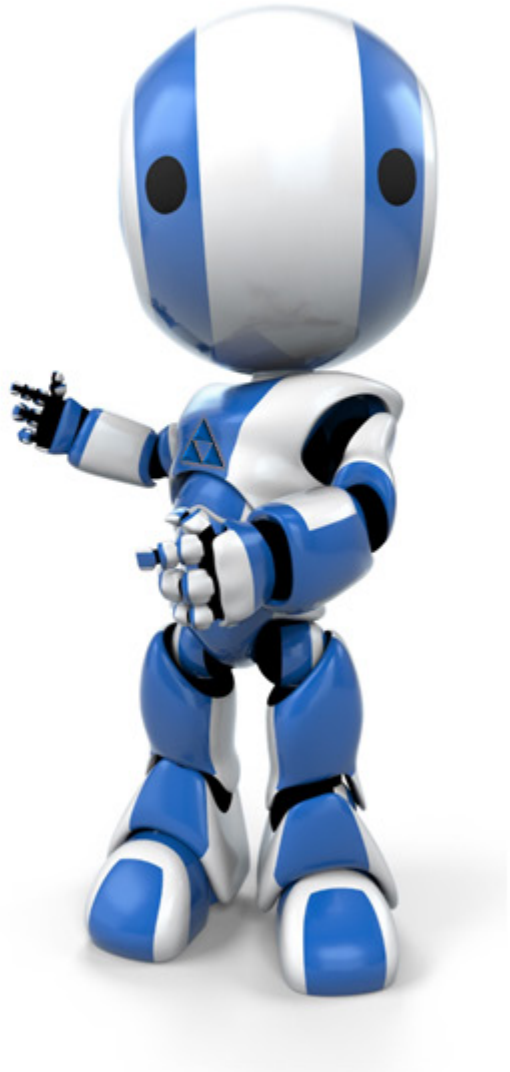
iiRDS applications

- Content delivery
- Variant management
- Knowledge graphs with generative AI

MICROCONTENT

standardizing how you

write your content



Users struggle with your content because it is

- unfocused
- overfilled with information, and
- not matched to their goals.

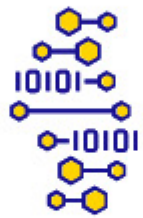
Four principle characteristics for microcontent

1. Focus

Limit microcontent to a discrete answer to a question

2. Function

Assign microcontent to a specific intended user response



3. Structure

Write microcontent using predictable patterns and language

4. Context

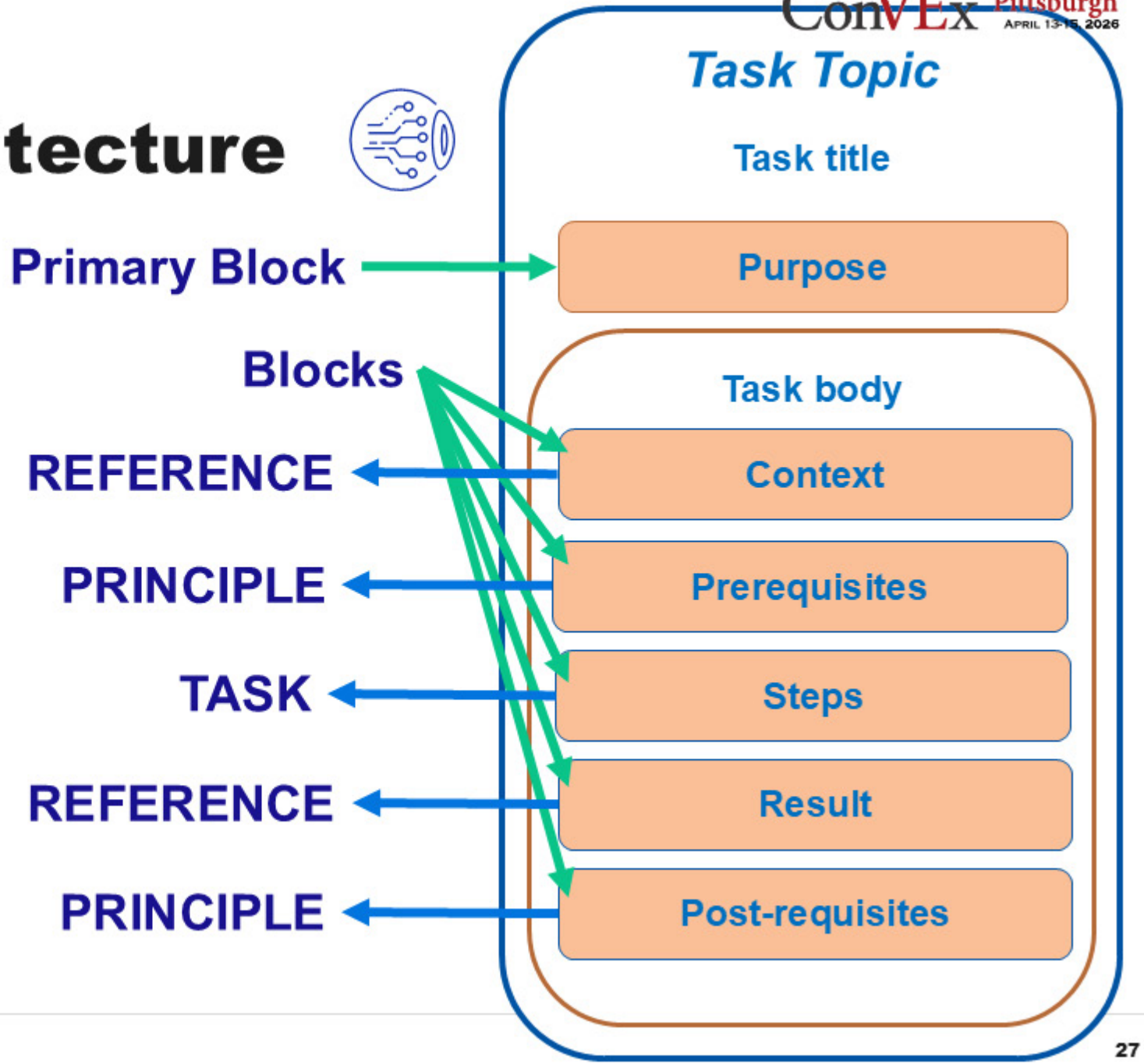
Classify microcontent to make it easily relatable to other content

Topic block architecture

Consider what happens if we focuses writing at the block-level within topics

The short description supports the title of the topic as a block

Every block is an information type supporting the topic



Precision Content® Information types

Reference

DESCRIBES things the reader needs to **KNOW**

Task

INSTRUCTS the reader **HOW TO DO** things

Concept

DEFINES things the reader needs to **UNDERSTAND**

Process

EXPLAINS to the reader how things **WORK**, and

Principle

ADVISES the reader about what they need **TO DO or NOT DO and WHEN.**

Before

Time-based synchronization, also known as distributed clock synchronization, is characterized by the use of an external timing source such as GPS, 1588, or an external IRIG generator. The system timing module uses the external time reference to determine the present time and create a clock that is locked to the external source. The individual clocks of each module and device in the system are synchronized to the same external source, ensuring synchronization between nodes no matter how far apart they are. Devices act on timing signals originating from a local clock that is synchronized to the other clocks in the system, so instead of sharing timing signals directly, the devices periodically adjust their local timing sources to match the chosen external time reference.

Using the time-based synchronization method, you can perform the following actions:

- Create future time events that execute at a specific board time to control clock and trigger signals.
- Write and read timestamps to measure clock skew, record the start time of data acquisition, and troubleshoot timing issues.
- Create timed loops that run at a specific time of the day.
- Discipline the backplane clock to an external time reference.
- Return the current data and time, or the date and time when a measurement was taken.
- Generate a sample clock that starts and stops at a specific board time.

What frequency sources do most time-based timing and synchronization devices use?

Synchronizing distributed clocks requires constant adjustment. A clock is essentially a two-part device that consists of a frequency source and an accumulator. In theory, if you set two clocks identically and their frequency sources run at the exact same rate, they are synchronized indefinitely. In practice, however, clocks are set with limited precision, frequency sources run at slightly different rates, and the rate of a frequency source changes over time and temperature. Most time-based TimeCo timing and synchronization devices use an over-controlled crystal oscillator (OCXO) or a temperature-controlled crystal oscillator (TCXO) as a frequency source, but even these highly accurate frequency sources vary due to initial manufacturing tolerance, temperature and pressure changes, and aging.

Microcontent

Is content that is

- about one primary idea, fact, or concept
- easily scannable
- labelled for clear identification and meaning, and
- appropriately written and formatted for use anywhere and any time it is needed.

After

What frequency sources do most time-based timing and synchronization devices use?

How time-based synchronization works

The individual clocks of each module and device in the system synchronize to the same external source. Devices act on timing signals originating from a local clock that synchronizes to the other clocks in the system. Instead of sharing timing signals directly, the devices periodically adjust their local timing sources to match the chosen external time reference.

Why synchronization is required

A clock is consists of a frequency source and an accumulator. Synchronization is required as follows:

In theory …	In practice …
you set two clocks identically	you can set clocks with limited precision
their frequency sources run at the exact same rate	frequency sources • run at slightly different rates, and • change rate over time and temperature
they are synchronized indefinitely	distributed clocks must be synchronized continually in frequency and phase

Time-based TimeCo device frequency sources

Most time-based TimeCo timing and synchronization devices use one of the following as a frequency source:

- an over-controlled crystal oscillator (OCXO), or
- a temperature-controlled crystal oscillator (TCXO).

Sources of variation

Even these highly accurate frequency sources vary due to

- initial manufacturing tolerance
- temperature and pressure changes, and
- aging.

Advantages and disadvantages of time-based synchronization

There are advantages and disadvantages to time-based synchronization.

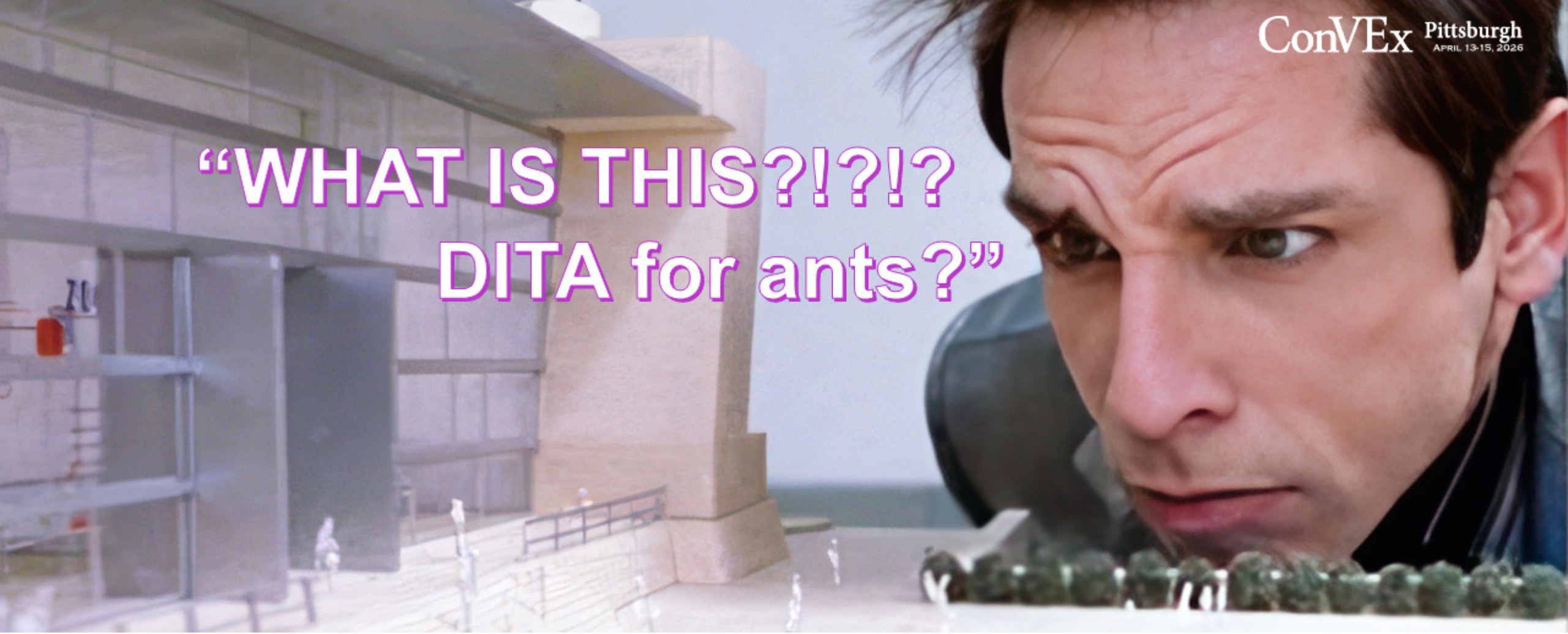
Disadvantage

A time-based system is generally not as accurate as a signal-based system.

Advantages

Time-based synchronization enables you to

- synchronize complex systems with many different nodes distributed over a large area with no loss of accuracy, even when the nodes are moving, and
- measure the location, speed, and altitude of a node when using the GPS timing protocol.



It’s not microcontent just because it’s small

What a microcontent standard provides

- Common model
- Interoperability
- Guidance

Writing structures, grammar, and syntax

A medium for exchange that is format-free, platform-independent

Principles for identifying the audience, purpose, and intended audience response

What a microcontent standard provides

- Common model
- Interoperability
- Guidance

What are the benefits?

- Consistency
- Universality
- Clarity

Microcontent as a medium for exchange

Microcontent is not strictly an input nor an output format. Instead, microcontent is a medium for exchanging information across different platforms and formats.

Units of microcontent need to contain

- piece of standalone content, and
- metadata records.

Content and metadata need to be automatically extracted at publishing time.

Facilitated usability lab

Both before and after samples are presented to a user in a controlled environment.

The facilitator will

- observe the behaviour of the user
- record the time-to-answer
- assess the confidence level, and
- evaluate the answer.

Test scenario

Your boss has just called you at your desk to ask a question. You've placed them on hold while you look up the answer.

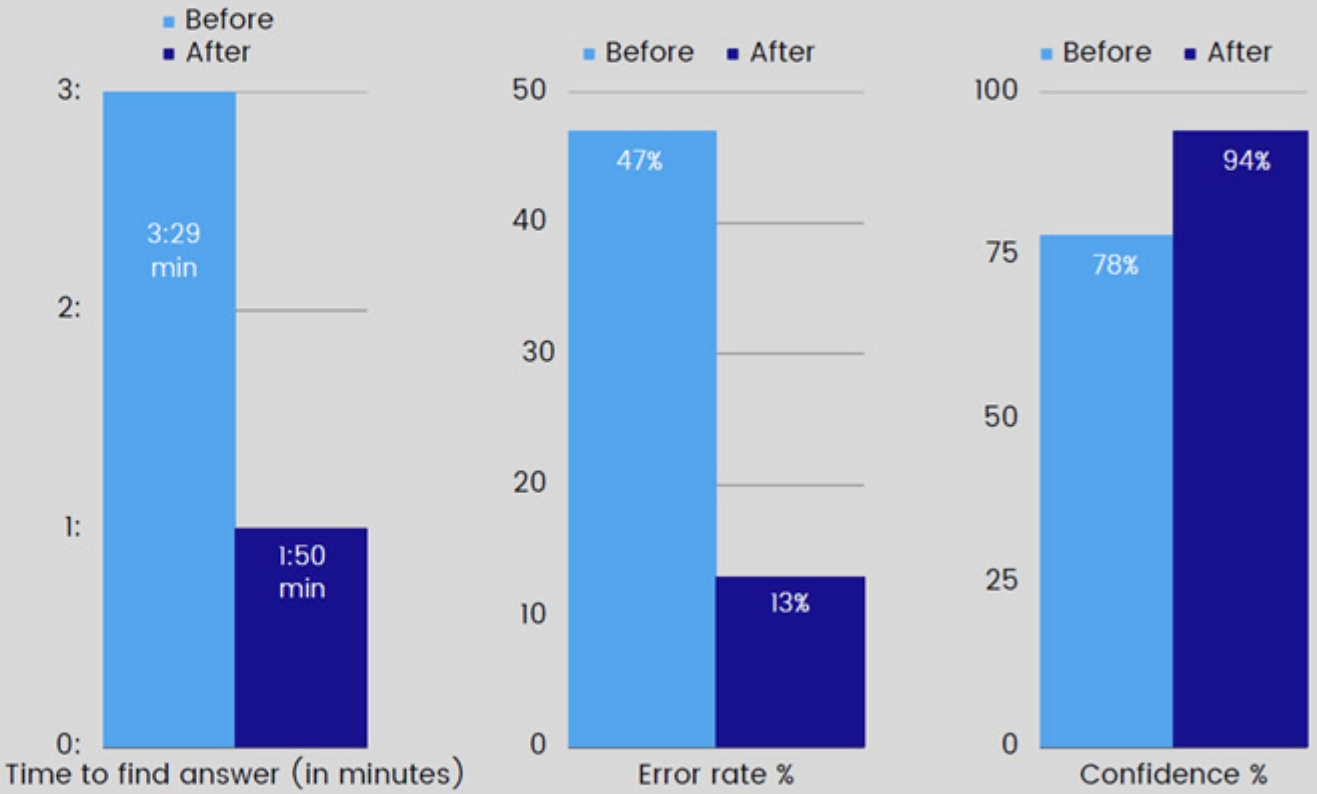
Scan the procedure document and find the answer.

Once you have found the answer, pick up the phone and relay the answer to your boss.

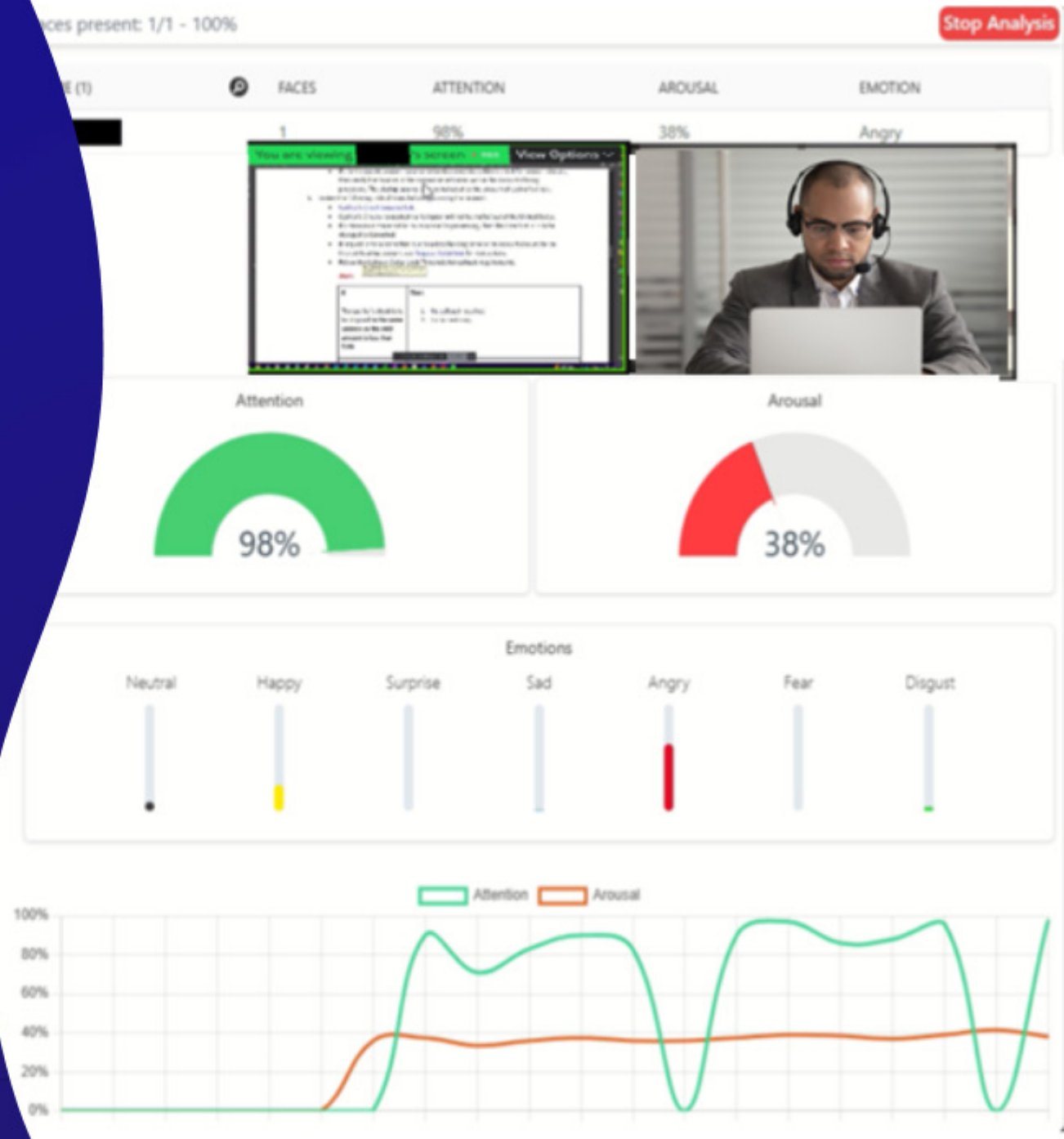
Before and after stats

Usability Test Results

High level statistics from the Usability Lab data

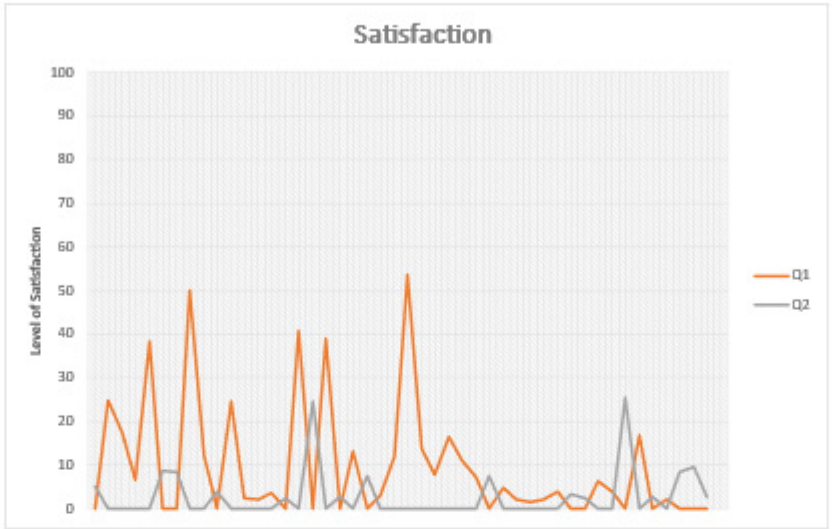


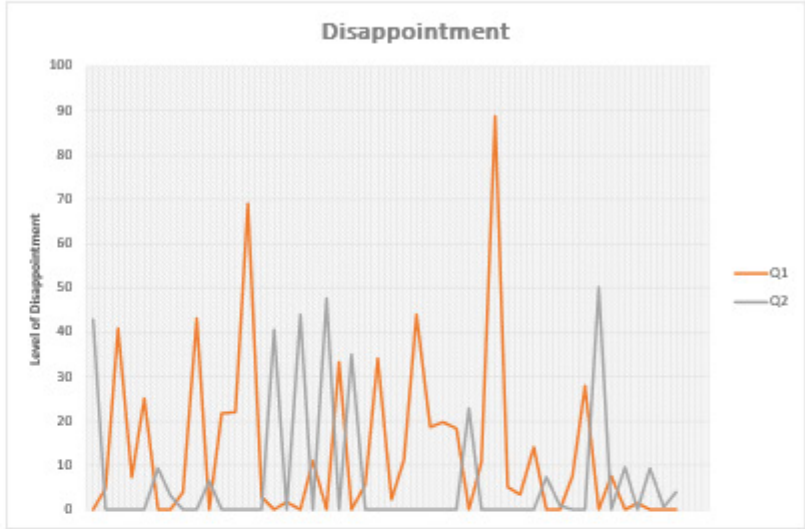
Real-time facial analysis

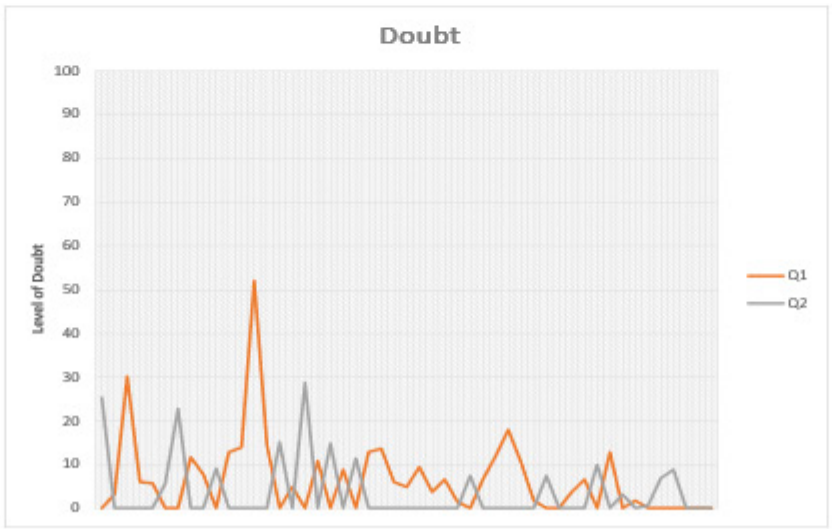


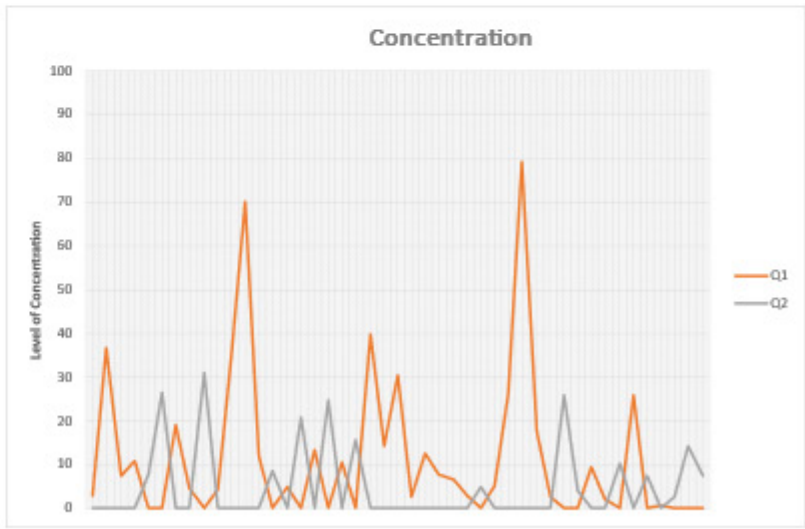
MorphCast for Zoom

Voice Prosody Results User 1









Hume AI

OT chunking experiment

We tested the effectiveness of a basic RAG-based response over 100 sample questions for metadata-enriched semantic chunks compared to

- semantic chunks with no metadata enrichment
- topic-level chunks, and
- entire publication with no semantic chunking.

Semantic chunking experiment

- A RAG benchmark inside an AI-ready DITA pipeline
  - **Goal:** test whether DITA-aware chunking improves RAG answers over generic chunking
  - **Corpus:** Precision Content style standards + employee guide in DITA
  - **Core question:** if the source already has semantic structure, should chunking preserve it?

Four chunking strategies same corpus

Test Sets

A. Non-enriched markup-based
Subtopic units only, no metadata enrichment

B. Semantic
Same subtopic units but enriched with context and metadata from the surrounding topic

C. Topic-level
Each whole topic rendered as Markdown

D. PDF-based
Upload whole PDFs and let the RAG platform chunk them

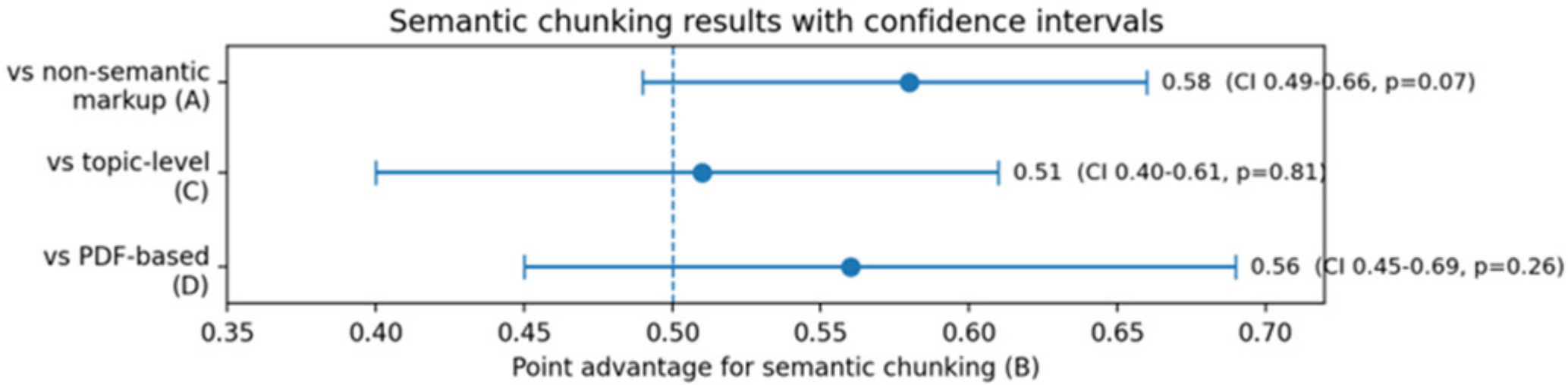
Test

- A vs B
- B vs C
- B vs D

Evaluation

- LLM-generated 100 test questions + reference answers
- Same questions run against four AnythingLLM workspaces
- Pairwise LLM judge scored wins, ties, and point advantage

Semantic chunking test results



Interpretation: Semantic chunking had a

- clear edge over non-enriched markup-based chunks
- slight edge over topic-level chunks, and
- clear edge over platform-generated PDF fixed chunking.

Take aways: Promising signals

- While promising, results are not definitive on this small sample
  - The study was intentionally lightweight and underpowered
  - Confidence intervals are wide, so this is evidence of promise, not proof
  - Best interpretation:
    - metadata enrichment helps smaller chunks stay interpretable
    - topic-sized chunks remain a strong baseline
    - generic PDF chunking is weaker than author-side semantic preparation
  - Pipeline implication: OT transforms and metadata strategy are part of the AI system, not just publishing

Lean RAG response layer

- Chunk-level where-used metadata harvested from CCMS, publication, map, topic, element and filtered based on feature applicability at delivery end-point
- No reused blocks are duplicated in the RAG system – they are referenced
- Only microcontent deltas sent to the response layer
- Variants managed by core-exception rules, the same block applies in all contexts except where it doesn't

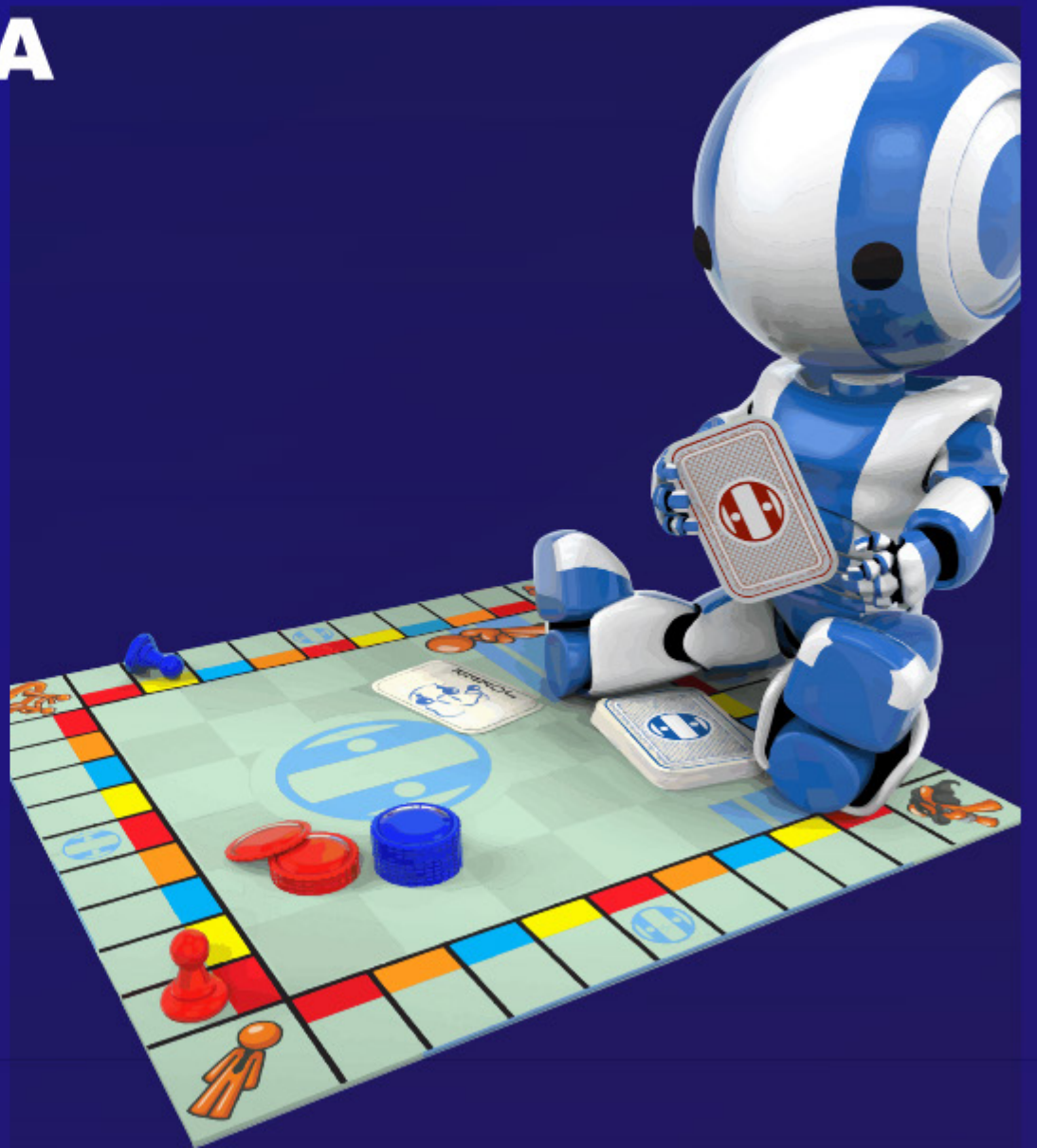
Leadership lens

Five questions leaders must ask

1. What is the governed source of truth behind the AI solution?
2. Are topics typed for user intent or flattened text?
3. Is metadata required and meaningful, or optional and uneven?
4. Can the pipeline emit AI-ready outputs without rewriting the source?
5. Can the response cite approved, version-aware content for the current user?

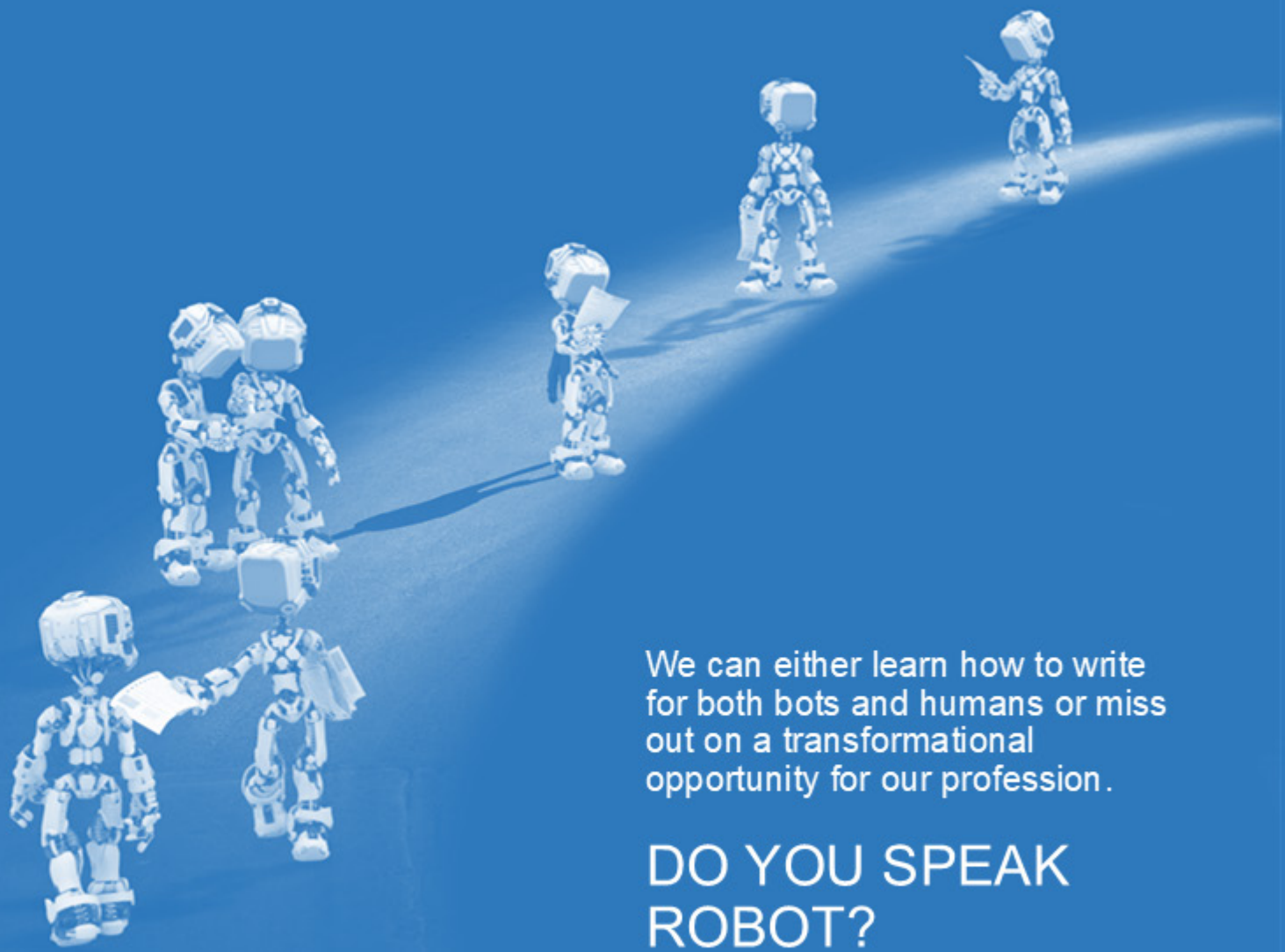
Double down on DITA

Double down on structured authoring.



ConVEx Pittsburgh
April 13-15, 2026

The robots are coming ...



We can either learn how to write for both bots and humans or miss out on a transformational opportunity for our profession.

DO YOU SPEAK ROBOT?