📘 **Unified Zyme Specification — Draft Outline v3

author: Stephen Pasco

0. Executive Summary

Zymes are interactive, layered knowledge maps that allow users to explore complex material faster, more flexibly, and with deeper comprehension than traditional reading. Each Zyme distills a source—such as a court opinion, technical paper, or research brief—into an expandable, multi-level structure designed for progressive disclosure: from high-level highlights to verbatim source material.

Unlike traditional summaries or static outlines, Zymes are **graph-native**. Every Zyme is both:

- A **root node**: an entry point derived from a specific document or concept, and
- A **graph node**: part of a larger, interconnected knowledge network linking terms, concepts, people, places, events, and other Zymes.

This means users can not only "zoom in" within a Zyme (L0 → L1 → L2), but also "jump out" from any point to explore related Zymes—such as a cited case, a referenced method, or the biography of an author. Over time, Zymes allow users to construct and navigate a personalized knowledge graph, shaped by their own interests and exploration history.

This specification defines how Zymes are created, stored, and interacted with. It includes:

- A standardized multi-level playbook (L0-L2) for structuring content.
- A JSON schema and storage model for rendering and linking Zymes.
- UI and UX rules for navigating within and across Zymes.
- A graph data model for representing concepts, definitions, and relationships.
- Extensibility support for academic papers, glossary nodes, concept graphs, and future variants.

Zymes enable a new kind of intellectual exploration: precise, personalized, and nonlinear. Whether scanning a 1-paragraph legal abstract or mapping a 40-page academic paper, Zymes let readers dive only as deep as needed, define unfamiliar terms on the fly, and follow connections into entirely new domains—one node at a time.

1. Conceptual Foundations

1.1 What is a Zyme?

A Zyme is an interactive, multi-level and multi-mode knowledge object that transforms a dense source—such as a legal ruling, technical paper, policy brief, or data-rich report—into an intuitive, explorable map. At a glance, a Zyme delivers progressively disclosed information through layers:

- **L0 (Initial View / Highlights):** The first view of a Zyme, presenting a high-level overview. This layer often utilizes a 5W+H (Who, What, Where, When, How, How Much) framework, displayed as a grid of "Highlight Cards." Each card represents a key insight with its own title and summary bullet. (See Section 8.2 for L0 wireframe details).
- **L1 (Clean Bullets):** Clear, jargon-free bullet points expanding on each L0 highlight. L1 design is work-in-progress. Each L0 highlight expands into 5-9 full-sentence bullets, each ≤ 25 words, with acronyms expanded and pronouns removed for complete thoughts.
- **L2 (Structured Summaries):** Compact evidence and reasoning. This layer provides structured summaries with charts, evidence, and reasoning. L2 design is work-in-progress. Typically a 1-paragraph summary (3-5 sentences, 120-180 words) and up to 2 charts, following a template like: Purpose • Method • Findings (≥ 2 numbers, cite [Chart A/B]) • Implication.

Each layer progressively discloses more detail, allowing readers to choose their depth of understanding without cognitive overload. Zymes are designed to preserve auditability, clarity, and relevance, even in fields where nuance and terminology are critical—such as law, medicine, science, and finance.

1.2 The Zyme as a Node in a Larger Knowledge Graph

While Zymes begin as transformed documents, they are also graph-native units of knowledge. Every Zyme becomes a node in a larger **semantic network**—a system of interlinked concepts, people, cases, terms, and supporting documents.

From any highlight, bullet, or glossary term, users can "jump out" to a related Zyme:

- Click "NP-complete" → load a concept Zyme defining and explaining it.
- Click "Justice Ginsburg" → load a profile Zyme with authored cases.
- Click "AAAI 2023" → navigate to a Zyme summarizing that conference's papers.

This turns every Zyme into both an endpoint (for comprehension) and a starting point (for exploration).

1.3 Motivation: Reduce Reading Time, Expand Navigability

Traditional documents are long, linear, and full of cognitive friction. Professionals waste hours parsing 40-page PDFs, re-reading sections to recall key points, or searching external sources to understand unfamiliar terms.

(Needs updating? Does this part need to be updated based on the latest wireframes?)

Zymes address these challenges by:

- **Compressing time:** Pilot users saw time-to-comprehension drop from 47 minutes → 12 minutes.
- **Reducing friction:** Glossary tooltips, expandable bullets, and linked visual summaries eliminate the need to break reading flow.
- **Unlocking connections:** Users can follow terms, citations, or people across documents, enabling nonlinear learning and personalized inquiry.

The goal isn't just faster reading—it's better **retention, insight discovery, and decision-making.**

1.4 Types of Zymes

Zymes aren't one-size-fits-all. While many are created from source documents, others originate from concepts, entities, or events.

Zyme Type	Description
Document-rooted	Standard Zymes built from a PDF or written source.
Concept-rooted	Standalone Zymes defining a key idea, e.g., "Bayes Theorem".
Entity-rooted	Zymes for people, institutions, or cases (e.g., "Justice Sotomayor").
Event-rooted	Zymes for events like conferences, meetings, or rulings.
Multimodal-rooted	Future Zymes from transcripts, audio, or visual media.

Each Zyme type uses the same three-level structure but may rely on different generation logic or metadata. Together, they form a heterogeneous but unified knowledge graph.

2. Zyme Structure & Content Generation

2.1 Level Ladder and Inclusion Rules (L0, L1, L2)

L0 (Initial View - Highlights):

- **Purpose:** Jargon-free orientation.
- **Content:** Typically 3-5 highlights presented as cards, often following a 5W+H structure (Who, What, Where, When, How, How Much). Each card has a title, an *optional* hero visual, and a brief summary. (See Section 8.2 and provided wireframes).
- **Generation:** Selected based on scoring rules from the core JSON extraction.

L1 (Clean Bullets):

- **Purpose:** Jargon-free orientation, expanding L0.
- **Include:** 5-9 full-sentence bullets per L0 highlight.
- **Length limit:** ≤ 25 words each.
- **Generation Rules:** Rewrite L0 bullets: expand acronyms, remove pronouns, complete thought.

L2 (Summary):

- **Purpose:** Compact evidence + reasoning.
- **Include:** 1 paragraph (3-5 sentences) + ≤ 2 charts.
- **Length limit:** 120-180 words.
- **Generation Rules:** Template: Purpose • Method • Findings (≥ 2 numbers, cite [Chart A/B]) • Implication.

2.3 Glossary Generation & Display Logic

Zyme builds a glossary layer to support hover-to-define and jump-to-explanation UX:

- **Candidates:** UPPERCASE tokens; multi-word noun phrases with frequency ≥ 3.
- **Filtered:** Drop stopwords and common English terms.
- **Source priority:**
  1. Explicit definition in the source.
  2. Wikipedia first sentence.
  3. LLM-generated fallback.
- **Output:** One-sentence definition ≤ 25 words, high-school readability.
- **Presentation:**
  - Glossary terms appearing within Zyme content (e.g., Zyme Abstract, card titles, bullets, summaries) are **visually distinguished** (e.g., underlined and/or with a specific color) to indicate their interactive nature.
  - Hover tooltip over the distinguished term → displays one-sentence definition.
  - Click distinguished term → opens glossary side panel with full definition, and a "Go to Zyme" link if a dedicated Concept Zyme exists for that term.

2.4 Salient Figure Heuristic (for [Chart A]/[Chart B])

(TBD)

2.5 Confidence Scoring & Error Flags

Each L1 section receives a confidence score: `conf = 0.5*softmax(avg_logprob) + 0.3*ROUGE-Lnorm(L2) + 0.2/(1 + summary_length/180)`

- If `confidence < 0.60` → flag `low_confidence` → route to human review.
- If `L2 > 180 words` → trigger `summary_overlimit` → auto-shrink or truncate.
- If visual extraction fails → flag `missing_visual`.

These flags ensure quality and drive the human-in-the-loop (HITL) QA workflow.

2.6 Prompt Templates (Embedded in Zyme Generation Pipeline)

Task	Prompt
Glossary term	"Define term in ≤ 25 words a smart high-school student can understand."
Clean bullets	"Rewrite each bullet as one jargon-free sentence ≤ 25 words. Expand first acronym. Keep numbers."
Summary	"Write a 120-180 word paragraph: Purpose • Method • Findings (≥ 2 numbers, cite [Chart A/B]) • Implication."

These prompts can be tuned per document type (e.g., legal vs. scientific).

2.7 JSON Output Schema (Core)

This is the foundational JSON structure populated by LLMs from the source document. The `what.document_type` field is critical for guiding optional, category-specific data extraction.

```
{
  "who": {
    "authors_and_entities": [
      {
        "name": "string",
        "type": "string"
      }
    ]
  },
  "what": {
    "title": "string",
    "summary": "string | null",
    "keywords": ["string"],
    "document_type": "string | null",
    "research_questions_or_objectives": ["string"]
  },
  "where": {
    "publication_venue_name": "string | null",
    "publisher_or_issuing_body": "string | null",
    "doi": "string | null",
    "publication_or_event_location": "string | null",
    "publication_url": "string | null"
  },
  "when": {
    "publication_or_event_date": "string | null",
    "submission_date": "string | null",
    "acceptance_or_effective_date": "string | null",
    "reporting_period": "string | null"
  },
  "how": {
    "methodology_summary_or_process_description": "string | null",
    "key_findings_or_results_or_outcomes": ["string"],
    "main_conclusions_or_recommendations_or_rulings": ["string"]
  },
  "how_much": {
    "sample_size_or_dataset_description": "string | null",
    "key_quantitative_results_or_metrics": ["string"],
    "limitations_stated_or_risks_identified": ["string"]
  },
  "extraction_confidence_notes": {
    "overall_confidence_score": "number | null",
    "uncertain_fields": [
      {
        "field_path": "string",
        "note": "string"
      }
    ]
  },
  "category_specific_data": {
```

```
    // Optional extensions based on document_type
  }
}
```

3. Knowledge Graph Model

Zymes are not isolated summaries—they are **nodes** in an interconnected **knowledge graph**. This structure allows users to explore related ideas, trace concepts across documents, and build a personalized web of understanding.

3.1 ZymeNodes as Graph Entities

Every Zyme—whether rooted in a document, a definition, or an entity—is a **graph node**.

```
ZymeNode {
  id: string,                // URI or UUID
  type: string,              // e.g. "legal_opinion", "concept", "person", "event"
  title: string,
  metadata: {
    source_url?: string,
    pub_date?: string,
    authors?: [string],
    description?: string
  },
  levels: { L0, L1, L2 },   // standard 3-level structure (JSON content for each layer)
  edges: [ Edge ]            // outbound connections to other ZymeNodes
}
```

This allows for uniform treatment of both full documents and lightweight "concept nodes" that explain a single term.

3.2 Edge Types and Relation Semantics

ZymeNodes are connected by **typed edges**, which define the relationship between two nodes.

Relation	Meaning	Example
`CHILD_OF`	Zyme A navigates into Zyme B as a deeper explanation	"Intro to ML" → "Gradient Descent"
`DEFINED_BY`	Zyme A uses a term defined by Zyme B	"NP-complete" → definition Zyme
`RELATED_TO`	Peer-level semantic relationship	"Linear Regression" → "Logistic Regression"
`AUTHORED_BY`	Zyme was written by Person node	"Roe v. Wade" → "Justice Blackmun"
`MENTIONED_IN`	Zyme B is cited or referenced within Zyme A	"Smith v. Jones" → "Statute 28 USC 1331"
`EVENT_AT`	Zyme describes event at place/time	"ICML 2024" → "Vienna"

Each edge is directional and includes metadata:

```
Edge {
  target: string,        // ID of other node
  relation: string,      // One of the types above
  via: {
    level?: "L0" | "L1" | "L2",
    keyword?: string
  }
}
```

3.3 Expanding Nodes Beyond Documents

A Zyme may originate from:

Node Type	Definition	Example
Document	Rooted in a file or text corpus	Legal opinion, academic paper
Concept	A term, theory, or keyword with meaning across documents	"NP-complete", "Justice"
Person	A named individual with authorship or mention connections	"Ruth Bader Ginsburg"
Event	A dated occurrence, e.g. court ruling, conference	"AAAI 2025"
Place	A geographic region tied to a legal case, trial, or study	"Eastern District of NY"

Each of these has a corresponding ZymeNode and can be displayed using the same progressive disclosure levels. For instance, a Person node's L0 might list authored works; L1 might contain bios or highlights; and L2 a full narrative.

3.4 Breadcrumb Trails: Graph vs. Layer

Zymes track two navigational paths:

- **Breadcrumb A — Layer Trail:**
  - Tracks your vertical position inside a Zyme.
  - Example: Highlights → Clean Bullets → Summary → Source
- **Breadcrumb B — Graph Trail:**
  - Tracks your movement across ZymeNodes.
  - Example: Roe v. Wade → Due Process Clause → 14th Amendment → Lochner v. New York

Each breadcrumb provides orientation and navigation context without overwhelming the user. Users can switch focus between detail and discovery without losing their place.

3.5 Hop-Aware Exploration (Future Feature)

Zyme's graph layer supports optional **hop distance calculations**, allowing users to explore how far two nodes are apart:

- "This term is 2 hops away from 'Cook-Levin Theorem'."
- "Here are 3 Zymes that connect 'Bayes' Theorem' to 'Adversarial ML' in 3 steps."

This will be enabled via `hop_distance` metadata in edges and surfaced in:

- Concept panels
- Advanced exploration views
- "My Zyme Map" overlays

Though not required for MVP, this feature enriches lateral discovery, curiosity-driven navigation, and knowledge graph visualization.

4. Processing Workflow

The Zyme generation process is a modular pipeline that transforms raw documents into structured, navigable, and graph-linked knowledge objects. The workflow is designed to be **scalable, traceable, and human-correctable.**

4.1 End-to-End Flow

The following steps describe how any input—text, PDF, or structured data—becomes a rendered Zyme:

1. **Step 1: Ingestion**
   - **Input types:** PDF, DOCX, HTML, plain text, transcription, scraped web data, document URL.
   - **Preprocessing:**
     - Optical Character Recognition (OCR) if needed (e.g., via Mistral AI OCR).
     - Text normalization: remove headers, extract metadata.
     - Section splitting (headings, page breaks, or sliding window).
   - **Output:** Cleaned document with tokenized segments and metadata.
2. **Step 2: Clustering & Top-Idea Selection (for L0 card generation from long documents)**
   - Segment document into ~500-word logical units.
   - Embed each segment using SBERT or similar encoder.
   - Cluster to detect 3-7 high-density topics using `k = min(7, sqrt(N_segments))`.
   - Label each cluster using high-salience sentence ≤ 6 words.
   - Apply overflow rule: If mean Silhouette > 0.72 and doc > 80 pages, allow 8-10 cards.
3. **Step 3: Core JSON & Layered Generation (L0–L2)**
   - LLM populates the Core JSON Schema (Section 2.7), including `what.document_type`.
   - If applicable, LLM populates `category_specific_data` based on `document_type`.
   - For each top-level idea (L0 card ID, derived from clustering or core JSON fields):
     - **L0:** Select 3-5 highlights with scoring rules.
     - **L1:** Clean and expand L0 bullets into full sentences (using LLM prompts from Sec 2.6).
     - **L2:** Use structured summary prompt → generate paragraph + pick [Chart A/B] (using LLM prompts from Sec 2.6).
4. **Step 4: Glossary & Edges**
   - **Glossary terms:**
     - Extract multi-word nouns + uppercase terms (freq ≥ 3).
     - Lookup in source → Wikipedia → LLM fallback (using LLM prompts from Sec 2.6).
     - One-sentence plain-language definition.
   - **Edge detection:**
     - Add edges for `DEFINED_BY`, `CHILD_OF`, `RELATED_TO`, etc.
     - Link authors, concepts, citations, keywords to ZymeNodes.
   - **Output:** Fully populated ZymeNode + Edge list + glossary dictionary.
5. **Step 5: JSON Output + Persistence**
   - **Emit canonical JSON:** `levels.L0–L2`, `glossary`, `edges`, `nav.order`, `confidence`.

- **Store in:**
  - `zyme`, `zyme_section`, and `zyme_edge` tables (PostgreSQL).
  - Optional: duplicate edge data in Neo4j for complex traversal.
- **Trigger downstream actions:**
  - UI render, compare-mode indexing, search vector update.

4.2 Error Handling & Human Review

The pipeline includes automated quality checks and flags:

Trigger	Flag	Action
Confidence < 0.60	`low_confidence`	Send to human QA
Summary > 180 words	`summary_overlimit`	Auto-shrink or truncate
Visual extraction fails	`missing_visual`	Placeholder stub + warning log
Missing glossary candidate		No action (silent fallback)

Human-in-the-loop workflow:

- Flagged sections routed to **editor dashboard** (e.g., Airtable or internal CMS).
- Editors see: source, flagged L1/L2 content, edit box, and regenerate button.
- Accepted edits are written back to DB and used in RLHF pairs.

4.3 RLHF Loop & Telemetry for Model Improvement

Zyme captures user signals to improve summarization, glossary quality, and graph recommendations.

Input Channel	Data Captured	Used for...
"⚠️ Report inaccuracy"	Zyme ID, section ID, freeform comment	QA dataset & error classification
Hover / click analytics	Keyword ID, glossary interaction count	Glossary quality tuning
Upvotes / downvotes	Section ID, user hash, sentiment	RLHF preference pairs
Dwell time	Per card · per level · per session	Confidence calibration

Retrain cadence:

- Telemetry + editor-corrected summaries = ~5,000 new preference pairs per quarter.
- Used for supervised fine-tuning (SFT) and reward modeling (RLHF).

Summary of Zyme Processing Workflow:

- Combines LLM generation, visual detection, and clustering to build structured, browsable knowledge.
- Flags low-confidence outputs for human review.
- Feeds user interaction back into the system for continual improvement.
- Produces a JSON-native ZymeNode ready for UI rendering and graph integration.

This workflow supports both breadth (any document) and depth (scalable knowledge trees).

5. UI & Interaction Design

The Zyme interface is designed for **clarity, control, and nonlinear exploration.** Every action—click, hover, scroll—reveals just the right amount of information, while preserving the ability to dive deeper or step sideways into related knowledge.

5.1 Level Navigation: L0–L2 (Breadcrumb A)

Each Zyme supports **progressive disclosure** across its content levels. The initial L0 view is structured as follows (refer to wireframes in Section 8):

1. **Main Zyme Title:** The overall title of the document or concept being Zyme-ified.
2. **Global Zyme Metadata:** Positioned directly below the Main Zyme Title, displaying the primary Date and Author(s) for the entire Zyme (for single-source Zymes).
3. **Zyme Abstract:** A concise 1-2 sentence overview of the entire Zyme's content, providing immediate overall context. Positioned below the Global Zyme Metadata.
4. **L0 Highlight Cards Grid:** A collection of cards (typically 3-7, based on 5W+H or key themes), each representing a key insight with its own title and summary bullet.

Interaction to navigate levels:

Level	Interaction	Result
L0	Click card or "+"	Expand to L1 bullets
L1	Click bullet or "+"	Expand to L2 summary paragraph
L2	Click chart, source link	Reveals source section (if applicable)

A persistent **breadcrumb trail (Breadcrumb A)** appears at the top of the screen (typically in the header): `Highlights → Clean Bullets → Summary → Source`

This allows the user to backtrack vertically within the current Zyme.

5.2 Node Navigation: Zyme-to-Zyme (Breadcrumb B)

Users may jump to **other Zymes** by clicking:

- Underlined glossary terms.
- Footnotes or citations.
- Author names.
- "Compare to..." menu options.
- "Explore related" in a mini-menu.

Each of these actions spawns a **new ZymeNode**, while retaining the original in session history.

A second **breadcrumb trail (Breadcrumb B)** tracks the graph journey:
`Roe v. Wade → Due Process → 14th Amendment → Lochner v. NY`

Breadcrumb B represents the **graph path** traversed across ZymeNodes. Users can click any point in the path to return to that node's last-viewed level.

5.3 Mini-Menus, Glossary, and Sibling Navigation

Each card or keyword includes a "+" mini-menu or interactive icon that reveals relevant options. A tooltip (e.g., "More options," "Actions") should appear on hover over the "+" mini-menu icon to clarify its function.

Element	Mini-menu Actions
Card	Open deeper · Copy link · Add note · Glossary · Compare to...
Keyword	Define term · Go to Zyme · View glossary
Author	Go to Author Zyme · View authored works
Citation	View cited Zyme · Copy reference

Glossary Tooltip + Side Panel:

- Hover on an interactively styled (e.g., underlined) glossary term → tooltip with one-sentence definition.
- Click term → right-side panel with full definition + "Go to Zyme" option if applicable.
- Pinned terms appear in "My Glossary" tab for reference.

Sibling Navigation:

- Shift-click on a different card opens **Compare Mode.**
- Right-click breadcrumb → shows tray of **sibling Zymes** (e.g., other cases by same judge).

5.4 Responsive Layouts

Viewport	Layout Characteristics
Desktop	3-5 L0 cards per row; side panels (Glossary, Notes) dock to right (320px)
Tablet (≥768)	Two-column grid; mini-map becomes dropdown
Mobile (<768)	Stacked cards; swipe L/R for sibling cards, swipe U/D to change levels
XL desktop	Zyme viewer max width = 1440px centered; mini-map fixed left (56px)

Mobile Gesture Map:

Gesture	Action
Swipe Left/Right	Move between sibling cards
Swipe Up/Down	Change depth (L0 ↔ L2)
Long-press	Open "+" mini-menu
Tap glossary	Show tooltip / side panel

5.5 Global "My Zyme Map" View

Users can activate a **graph view** of their exploration:

- Shows ZymeNodes visited in session as nodes.
- Edges represent `CHILD_OF`, `DEFINED_BY`, or `RELATED_TO` relations.
- Clicking a node reopens that Zyme at last-viewed level.
- Optionally includes filters: "Show definitions only", "Show authored by X", "Highlight legal opinions".

This enables users to retrace their learning path, revisit key concepts, or visualize how different topics relate.

5.6 Accessibility (A11y) Compliance

All Zyme interactions are designed to be accessible via keyboard and screen reader.

Feature	Accessibility Implementation
Cards	`<button role="group">`, `aria-level`
Mini-menu + tooltips	`aria-haspopup`, `aria-describedby`. Tooltips must be keyboard accessible, dismissible (e.g., with `Esc`), and properly announced by screen readers.
Keyboard shortcuts	`?` overlay listing full shortcut map.
High-contrast & reduced-motion	Toggle via Settings panel.
Glossary side panel	Landmarked region, keyboard focus trap.
Screen reader summaries	Each L0-L2 block has an alt-description string.
Interactive Keyword Links	Must meet accessibility standards for links: sufficient contrast against background, clear focus indication (not just color), and screen reader compatibility (e.g., announcing link text and purpose).

Summary of UI Design Principles:

The Zyme UI is designed around two key principles:

1. **Stay oriented:** Breadcrumb trails, consistent levels, and progressive disclosure help users avoid feeling lost.
2. **Explore freely:** Graph jumps, glossary reveals, sibling navigation, and Compare Mode allow nonlinear movement without penalty.

This system empowers users to explore deeply without overwhelm—and to move laterally across domains with ease.

6. Data Architecture & Storage

Zyme's architecture blends the strengths of **document-based, relational, and graph-based** storage patterns. This hybrid model allows Zymes to be stored as structured JSON for rendering, while also being **queryable by topic, edge type, authorship, and keyword relations.**

6.1 Mapping JSON → Relational Schema

ZymeNodes are emitted as structured JSON objects during generation (see §2.7), and stored in relational tables for indexing, analytics, and user permissions.

JSON Field	Table / Column	Notes
id, title, type	`zyme.id`, `zyme.title`, `zyme.zyme_kind`	Core metadata
levels.L0-L2	`zyme_section`	One row per level-section (content in JSONB)
glossary	`article_keyword` or `glossary_term`	Terms stored with source and confidence
edges	`zyme_edge` (new)	See §6.3
confidence	`zyme_section.quality_score`	Per-section

This enables Zymes to be rendered on-demand from the DB and queried by metadata, topic, or document type.

6.2 `zyme` & `zyme_section` Tables (Core)

The schema includes two central content tables:

- **`zyme`**
  - Represents the root Zyme object (one per document or concept).
  - Includes metadata: `title`, `owner_id`, `team_id`, `zyme_kind`, `description`, `timestamps`.
  - Stores processing state, quality score, and sharing flags.
- **`zyme_section`**
  - Stores each L0-L2 card/content block as a row.

- Columns:
  - `section_order`
  - `section_type` (overview, findings, methods, etc. - maps to 5W+H or L1/L2 types)
  - `content_data` (JSONB, e.g. summary or bullet content, L0 card title)
  - `interactive_config` (JSONB, menu options, glossary links)
  - `style_config` (JSONB, font, icon, color if needed)
  - `source_section_ids` (optional back-references to article_section)

Indexed by Zyme ID and section order for fast retrieval.

6.3 Graph Edges (New Table: `zyme_edge`)

This table powers graph traversal across Zymes, glossary terms, concepts, authors, and other node types.

```
CREATE TABLE zyme_edge (
  from_id      TEXT,        -- source ZymeNode ID
  to_id        TEXT,        -- target ZymeNode ID
  relation     TEXT,        -- CHILD_OF, DEFINED_BY, RELATED_TO, etc.
  via_level    TEXT,        -- L0–L2 (optional)
  keyword      TEXT,        -- triggering keyword (optional)
  hop_distance INT,         -- precomputed for future UX
  created_at   TIMESTAMPTZ DEFAULT now()
);
```

This enables queries like:

- "Which concepts are defined in this Zyme?"
- "What Zymes were authored by this person?"
- "How many hops connect Topic A and B?"

6.4 Storage Model Options

Option	Description	Use Case
PostgreSQL	**Primary relational DB for Zymes and edges**	**MVP backend; supports `jsonb`, GIN indexes**
Neo4j / GraphDB	External graph engine for deep traversal & inference	Optional for advanced graph views
MongoDB	JSON-native storage for unstructured documents	Viable for prototype, but less suited for joins
Hybrid	SQL for content, Neo4j for edges	Best for scale: keeps relational & graph optimized

(Default choice: PostgreSQL for MVP)

The schema is designed to be **queryable and renderable** without duplicating the entire knowledge tree into memory. Nodes and edges are resolved incrementally as the user navigates.

6.5 Cross-Zyme Search & Linking

(This section is work-in-progress)

Zymes can be linked, compared, and queried across documents via:

- **Full-text indexing** (via `tsvector` fields in `article_section` or `zyme_section.content_data`).
- **Keyword joins** (`article_keyword.confidence > 0.85`).
- **Edge traversals** (`zyme_edge` via relation type).
- **Author matching** (via `article_author` or `AUTHORED_BY` edge).
- **Event or timeline tags** (via metadata fields or edge nodes).

This architecture enables:

- Compare Mode: View two Zymes side-by-side.
- Related Zymes: Show siblings or linked nodes.
- Knowledge Graph View: Render "My Zyme Map" (visited nodes + links).

Summary of Data Architecture:

- Stores layered summaries (L0-L2) as structured sections.
- Uses `zyme_edge` to build a semantic graph between Zymes and concepts.
- Enables fast lookup, traversal, and recomposition via SQL + optional graph engines.
- Supports both document-rooted and concept-rooted knowledge nodes.

This foundation allows Zymes to be not just a visualization layer—but a composable, linkable, and queryable knowledge network.

7. Zyme Variants & Extension Paths

While the core Zyme architecture (L0-L2 + graph edges) is document-agnostic, some content types require **specialized adaptations**—either during extraction, rendering, or graph linking. This section outlines example variants, their unique needs, and how they extend the base model, primarily through the `category_specific_data` JSON block and tailored LLM prompting.

7.1 Legal Zyme Addendum (Beachhead Use Case)

(This section is work-in-progress)

U.S. appellate-court opinions form the beachhead use case for TimeZyme, requiring legal-specific features:

a) Section-Type Tagging:

- Distinguish between majority, dissent, concurrence, and syllabus.
- Stored as `section_type` in `zyme_section` or within `category_specific_data.legal_regulatory`.

```
CREATE TYPE legal_section_type AS ENUM ('majority', 'dissent', 'concurrence', 'syllabus');
```

b) Citation Parsing & Jump Links:

- Regex-based extraction of case citations (e.g., *Smith v. Jones, 492 U.S. 1*).
- Linked via `MENTIONED_IN` or `DEFINED_BY` edge to referenced Zyme.
- Tooltips show case name, year, and holding; click opens Zyme.

c) Legal-Aware Summary Prompts:

- Tailored summary structure for L2: Issue → Rule → Analysis → Holding.
- Separate prompt variant for dissents: "Main disagreement + alternative interpretation".

d) Paragraph and Line Preservation in L2:

- Inline tagging for citation support: `<p data-para="8">The court held that ...</p>`
- Enables direct references like "see para 8" and supports precise annotation.

e) Graph Structure:

- `AUTHORED_BY` edges link to judges.
- `MENTIONED_IN` edges tie cited statutes, clauses, and precedents into the broader legal graph.
- Event and jurisdiction metadata link to `EVENT_AT` and `PLACE` nodes.

7.2 Other Document Variants (Leveraging `category_specific_data`)

Variant Type	Adaptation Required (Examples for `category_specific_data` & Prompts)	Status (Example)
Clinical Studies	Add glossary emphasis (dosage units, drug names); specific outcome fields.	✅
ESG/Finance Reports	chart salience, entity linking; specific financial metrics.	✅
Annual Reports (10-K)	Section mapping for "Risk Factors", "Financial Highlights"; exec summaries.	✅
Patent Filings	Claim definitions, inventor links, prior art citations; patent-specific fields.	🚧
Technical Specs	Glossary from schema/abbreviation tables; spec tables.	🚧

(✅ = Well-defined path, 🚧 = Requires more detailed spec for category fields)

7.3 Concept & Entity Zymes

These are Zymes not derived from source documents, but from **key terms, people, or events** that arise across documents. Generated via `DEFINED_BY`, `AUTHORED_BY`, and `EVENT_AT` edges.

Entity Type	Typical L0-L2 Content
Concept	L0 = 1-liner · L1 = context bullets · L2 = explainer
Person	L0 = authored works · L1 = bio bullets · L2 = profile summary
Event	L0 = key outcomes · L1 = sessions/agenda · L2 = recap summary

7.4 Cross-Document & Timeline Zymes

a) Timeline Zymes:

- Constructed from date-stamped events across Zymes (e.g., case history).
- Rendered with `viz_type = "timeline"`.
- Events are clickable; each node links to its own Zyme.

b) Comparative Zymes:

- Compare two sibling Zymes (e.g., two court rulings on same issue).
- L0 + L1 content shown side-by-side.
- Enabled via `Compare to...` in card mini-menu.

7.5 Future Extension Paths (Illustrative)

(This table is work-in-progress)

Zyme Type	Why Different	Extraction Path
Interactive Dashboards	Primary source is visual; minimal text	Treat each chart as a "virtual section"; extract title, caption, data; apply L0-L2 logic
Video/Audio Zymes	Source = transcript with timestamps	Align transcript blocks to L0-L2; sync with playback
Multilingual Zymes	Support glossary/summary in multiple langs	Add `lang` tag to all levels; layer translations
Chat Logs / Support Threads	Speaker turns must be segmented & clustered	Treat turns as segments; cluster by topic
Codebases	Source = directory of files, not prose	Cluster functions/modules into "cards"; L2 = docstring + test case
Databases / Spreadsheets	Raw tables, not narrative	Generate insights (e.g., top deltas, outliers); wrap in L0-L2

Each of these will use the same node and edge structure but require specialized ingestion logic and potentially new `category_specific_data` blocks.

Summary of Zyme Flexibility:

Zyme is a flexible architecture built to support:

- Legal, technical, scientific, and business documents.
- Concepts, people, and events as standalone Zymes.
- Custom extensions for timelines, comparisons, and low-text inputs.

As new formats emerge, they can be incorporated via: Prompt changes, Edge extensions, Visualization adaptations, Input pipeline variants. Zyme's layered design ensures all variants are explorable in a consistent way—deep when needed, simple by default.

8. Wireframes & Visual Design System

The Zyme visual interface is grounded in simplicity, clarity, and progressive interaction. All components are styled in grayscale to emphasize hierarchy, with responsive layouts and accessibility built in from the start. This section defines the core design language, level-specific layouts, reusable components, and handoff conventions.

(Note: The following values are examples and need to be updated once the UI is finalized based on the provided wireframes.)

8.1 Shared Layout Grid & Visual Language

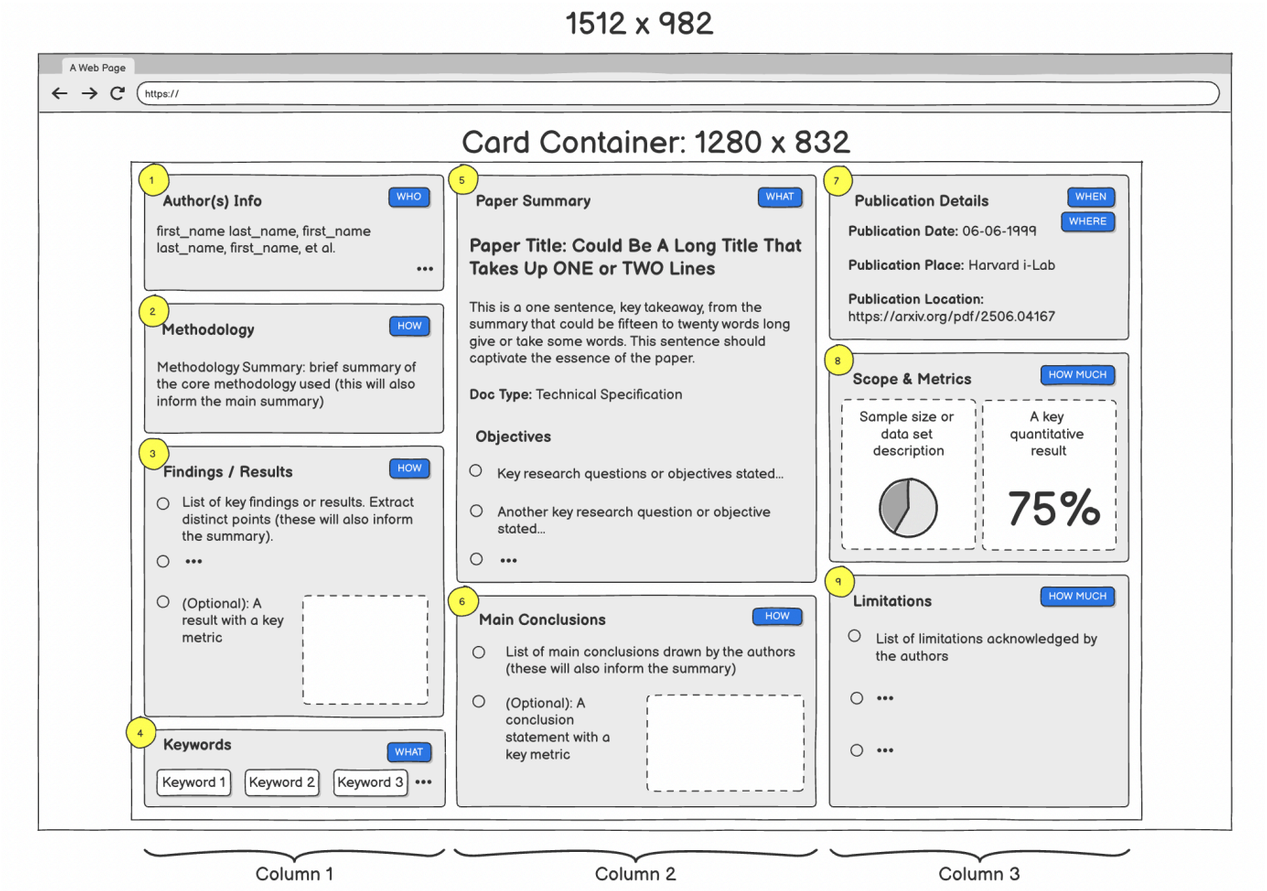
Element	Spec
Grid	12 columns (desktop), 4 columns (mobile/tablet)
Baseline	8 px units
Header	56 px tall · left: breadcrumb · right: search bar
Mini-map width	56 px fixed · one tick per L1 section
Typography	Inter or system font · 16 px base · 24 px for KPIs
Color	Grayscale only (#111–#EEE) to highlight content levels
Visual spacing	Card padding: 16 px · Bullet spacing: 8 px vertical

8.2 Level-Specific Frames

L0 – 5W+H Highlights (Initial View):

- Refer to `layout-example.png` for the card container (1280x832) and 3-column layout.
- Each card (e.g., Author(s) Info, Paper Summary, Publication Details, Methodology, Findings/Results, Keywords, Scope & Metrics, Limitations) maps to fields in the Core JSON Schema (Section 2.7).
- `JSON-to-Card.png` illustrates this mapping.
- The "Card Container" holds multiple L0 "Highlight Cards".

layout-example



json-to-card

JSON -> CARD Map

CARD 1: Author(s) Info

who > authors[full_name]

CARD 2: Methodology

how > methodology_summary

CARD 3: Findings / Results

how > key_findings_or_results

CARD 4: Keywords

what > keywords

CARD 5: Paper Summary

what > title
what > summary
Doc Type:
what > document_type
Objectives:
what >
research_questions_or_objectives

CARD 6: Main Conclusions

how > main_conclusions

CARD 7: Publication Details

Publication Date:
when > publication_date
Publication Place:
where > publication_location
Publication Location:
where > publication_url

CARD 8: Scope & Metrics

how_much >
sample_size_or_dataset_description
how_much > key_quantitative_results

CARD 9: Limitations

how_much > limitations_stated

JSON Data Definitions

```
{
 "who": {
  "authors": [ // List of author objects
   {
    "full_name": "string" // Full name of the author
   }
  ]
 },
 "what": {
  "title": "string", // Full title of the paper
  "summary": "string | null", // A concise summary (3-5 sentences) generated following the internal Chain-of-Thought steps (objectives, methods, findings, conclusions).
  "keywords": "string", // Extract a comprehensive list of key terms, concepts, acronyms, and named features/levels that are central to the paper's subject. Prioritize terms from any explicit keyword list, glossary, section titles, and those frequently emphasized (e.g., bolded, defined, or repeatedly referenced as core components). Aim for a list that accurately reflects the core vocabulary and concepts of the document. Use an empty list [] if no relevant items are found.
  "document_type": "string | null", // e.g., "Original Research", "Review Article", "Case Study", "Conference Paper", "Preprint". Infer if not stated.
  "research_questions_or_objectives": "string" // Key research questions or objectives stated, often in Intro/Abstract (these will also inform the summary)
 },
 "where": {
  "publication_venue_name": "string | null", // e.g., Journal name, Conference name
  "publisher": "string | null", // Publisher of the journal or proceedings
  "doi": "string | null", // Digital Object Identifier
  "publication_location": "string | null", // City/Country of journal/conference, if specified
 "publication_url": "string | null" // if available
 },
 "when": {
  "publication_date": "string | null", // Full publication date (e.g., "YYYY-MM-DD" or "Month YYYY")
  "submission_date": "string | null", // If available
  "acceptance_date": "string | null" // If available
 },
 "how": {
  "methodology_summary": "string | null", // Brief summary of the core methodology used (this will also inform the main summary)
  "key_findings_or_results": "string", // List of key findings or results. Extract distinct points (these will also inform the summary).
  "main_conclusions": "string" // List of main conclusions drawn by the authors (these will also inform the summary)
 },
 "how_much": {
  "sample_size_or_dataset_description": "string | null", // e.g., "N=150 participants", "Dataset of 10,000 images"
  "key_quantitative_results": "string", // Specific important numerical results or metrics mentioned
  "limitations_stated": "string" // List of limitations acknowledged by the authors
 },
 "extraction_confidence_notes": {
  "overall_confidence_score": "number | null", // Optional: A subjective score (0.0-1.0) of your confidence in the overall extraction accuracy for this document.
  "uncertain_fields": [ // List of fields where extraction was ambiguous or inferred with low confidence
   {
    "field_path": "string", // e.g., "what.document_type"
    "note": "string" // Brief explanation of uncertainty
   }
  ]
 }
}
```

L1 & L2:

- (Designs are work-in-progress, detailed in Section 1.1 and 2.1)
- Wireframe Reference 1a (from `Unified Zyme Specification-2.png`) shows a conceptual pop-out for Zyme Levels (One, Two, Three) corresponding to language complexity (Simplified, Student-friendly STEM, Academic). This needs reconciliation with the L0/L1/L2 structure. *(Self-correction: The L0/L1/L2 structure is primary; the "Level One/Two/Three" language complexity seems like a Pro feature or display mode on top of L0/L1/L2).*

8.3 Compare Mode, Side Panels & Interaction

Panel Type	Opens from	Width	Behavior
Glossary	Right side	320 px	Click on term or menu item
Annotation	Right side	320 px	Click "Add note" in mini-menu
Compare View	Split screen	50/50%	Shift-click or Compare mode

- Compare mode allows side-by-side Zyme cards with synchronized L1 scroll.
- Hovering on glossary terms highlights the same term across both views.
- Clicking outside or `Esc` exits split view.

8.4 Component Library & Naming (Example Figma Tokens)

Component	Style Token / Layer Name
Highlight card (default)	`TZ/Card/Highlight/Default`
Chart stub (line)	`TZ/Chart/Stub/Line`
Icon (copy link)	`TZ/Icon/Link`
Overlay (shortcuts)	`TZ/Overlay/KeyboardShortcuts`
Bullet row	`TZ/Bullet/Clean`
Confidence bar	`TZ/Indicator/Confidence/Default`
Breadcrumb bar	`TZ/Nav/Breadcrumb`
Tooltip	`TZ/Tooltip/Glossary`
Compare frame	`TZ/Compare/Overlay/Active`

All components are designed with auto-layout and Figma tokens to support responsive rendering.

8.5 Error States & Loading Indicators

Condition	Visual Cue
`missing_visual`	Grey stub card + broken-image icon
`summary_overlimit`	Ellipsis … + tooltip: "Content truncated..."
Load failure	Modal: "Content failed to load. Retry?"
No glossary terms	Side panel says: "No glossary terms found."
Loading (cards)	Ghost cards with pulse animation
Loading (initial)	Central spinner + document title placeholder

8.6 Export & Handoff Conventions

Item	Format	Notes
Balsamiq pages	`00–99`	One page per level, overlay, mode
Prototype links	Figma prototype	Includes: happy path, compare, glossary
Export presets	PNG 2x, SVG	Dev handoff + sprite library
Layer naming	`TZ/*`	Consistent with app layout engine

Summary of Design System:

- Enables consistent visual hierarchy across L0-L2 levels.
- Supports responsive interaction on desktop, tablet, and mobile.
- Provides flexible components for glossary, annotation, comparison, and graph traversal.
- Minimizes visual clutter while enabling deep interaction.

By grounding interaction in clarity and progressive detail, the visual system makes Zymes not only readable—but explorable.

9. Implementation & Milestones

The Zyme system is designed for a modular, testable rollout—starting with MVP document types (e.g., academic papers, U.S. appellate opinions) and expanding into a full-scale, graph-native knowledge platform.

9.1 MVP Architecture (Q2–Q3 2025)

The MVP phase will prioritize a vertical slice that demonstrates the core product loop:

- **Input:** Academic papers (URL or uploaded PDF); U.S. appellate opinions.
- **Output:** Rendered L0-L2 Zyme with glossary and basic graph links.
- **Users:** Students and science professionals, legal professionals, internal QA testers.

MVP Components:

Layer	Details
Backend	PostgreSQL (`zyme` + `section` + `glossary` + `edge` tables)
LLM Routing	Model selection per task (Mistral AI OCR, highlights, summaries, definitions); Inngest for queues.
Frontend	Nuxt 3 / Vue 3 + TailwindCSS + DaisyUI; responsive UI w/ mini-map & breadcrumbs.
Graph Engine	Postgres knowledge graph entity and relationship tables for MVP.
Storage	S3 or GCS for original PDFs; PostgreSQL for content metadata.
Search	Title + section text via GIN + `tsvector` .
Error flags	Summary over-limit, missing visual, low confidence.

9.2 Legal Pilot Plan / Academic Beachhead

Academic Zymes and U.S. appellate opinion Zymes will serve as the "beachhead" product segments.

Milestone	Goal	Timeline
M0	Validation study: ≥75% time savings vs. control	Q2 2025
M1	GA launch: 100 Pro users · CSAT ≥ 4.5	Q3 2025
M2	Team rollout: 25 teams · ≥30% Pro → Team conversion	Q4 2025
M3	Enterprise pilots: 3 accounts · ≥85% weekly retention	Q1 2026

(These milestones align with TimeZyme's go-to-market plan.)

9.3 RLHF Instrumentation

To improve generation quality and surface relevance over time, Zyme includes a full reinforcement learning loop:

Component	Implementation Notes
Confidence scores	Per-section · triggers review if < 0.60
User flags	"⚠️ Report inaccuracy" → sent to editorial queue
Telemetry	Dwell time, thumbs, term clicks = soft signals
Editor review	Manual approval of summaries → labeled preference pairs
Retrain schedule	Every quarter → ~5,000 new pairs for fine-tuning
Model targets	Summaries (L2), glossary (defs), bullet clarity

RLHF ensures Zymes get smarter over time, not just faster.

9.4 Next-Step Tasks (Engineering & Product - Post MVP Launch)

A. Graph Infrastructure:

- ✅ Add `zyme_edge` table for relation tracking.
- Support directed graph queries (`CHILD_OF` , `RELATED_TO`).
- ✅ Index edge metadata: via keyword, level, hop-distance.
- Enable breadcrumb B (graph trail) UI pathing.

B. LLM Pipeline Enhancements:

- Tune prompts per domain (academic, legal, business) using `category_specific_data` .
- Add model confidence reporting (token-level log-probs).
- Auto-shrink fallback for L2 summaries >180 words.

C. Frontend UI Work:

- ✅ Breadcrumb A (L0 → L2).
- Implement Breadcrumb B (Zyme-to-Zyme path).
- Polish Compare mode interaction.
- Implement My Zyme Map (session graph view).

D. Product/UX Experiments:

- Test Compare Mode for comprehension benefit.
- Glossary pinning & shortcut behavior.
- Personal "Zyme history" & saved paths.

Summary of Implementation:

The Zyme system is being implemented in **phased milestones**, beginning with an academically and legally focused MVP, then expanding to a generalized, multi-domain knowledge graph platform. Key ingredients for success include: Clear KPI-driven rollout, Strong feedback loop (telemetry + human review), Graph-aware UI with linked breadcrumbs, Extensible architecture to support new formats and domains.

10. Appendices

A. UI Interaction Cheat-Sheet

Action / Gesture	Result
Click card or "+"	Drill down one level (e.g., L0 → L1). A tooltip (e.g., "More options") appears on hover.
Shift-click card	Enter Compare Mode (select Card A).
"Compare to..." in mini-menu	Prompt to select Card B → open side-by-side view.
Click (styled) glossary term	Tooltip → side panel with full definition and "Go to Zyme".
Hover (styled) glossary term	Tooltip with 1-sentence definition.
Click author name	Open Zyme authored by that person.
Click citation / footnote	Open cited Zyme.
Breadcrumb A click	Navigate L0-L2 within a Zyme.
Breadcrumb B click	Navigate graph trail of visited Zymes.
Hover mini-map tick	Preview card title for section.
`←` / `→`	Navigate between sibling cards.
`↑` / `↓`	Navigate between content levels (within card context or L0/L1/L2 overall).
`?` or `Cmd/Ctrl + ?`	Show full shortcut overlay.
`/` or `Cmd/Ctrl + K`	Focus search (titles, glossary, bullet content).

B. Edge Case Handling

Condition	System Response
No highlights score ≥ 2	Use title + first abstract sentence as fallback L0.
Visual extraction fails	Show placeholder card with "missing visual" icon.
Summary > 180 words	Auto-shrink up to 3 times → truncate with tooltip warning.
All bullets tie for 5th slot	Keep highest-score + earliest-order ones.
No glossary terms found	Display "No glossary terms found" in glossary panel.
Citation lacks matchable ID	Render unlinked footnote with full text.

C. Parameter Reference Table

Parameter	Default Value	Tuned via
Cluster count (k)	`min(7, sqrt(N))`	Dynamic per doc
Silhouette threshold θ	`> 0.72`	Pilot studies vs. human clustering
Confidence threshold	`conf < 0.60`	ROC analysis on validation set
Glossary min frequency	`≥ 3`	TF-IDF + stop-word exclusion
Summary word limit	`< 180 words`	User UX tolerance (readability)
Visual-fallback trigger	No visual detected	Captions + alt-text missing
Text-first threshold	Text ≥ 30%	Byte count of extracted content (for visuals)

D. Glossary of Key Terms (from original spec)

Term	Definition
Zyme	A layered, interactive, explorable map of a document or concept.
ZymeNode	A graph node containing L0-L2 layers, edges, and metadata.
Zyme Edge	A typed relation between two ZymeNodes (e.g. `CHILD_OF`, `DEFINED_BY`).
Zyme Abstract	A concise 1-2 sentence overview providing immediate overall context for an entire Zyme, typically displayed at the L0 level before highlight cards.
Breadcrumb A	Vertical navigation path inside a Zyme (L0 → L2).
Breadcrumb B	Lateral navigation trail across Zymes (document-to-document).
Glossary term	Defined concept or phrase accessible by hover or click, visually cued in text.
Hero visual	A numeric card, spark-line, chart, or branded fallback shown on L0 highlight cards.
Confidence score	Automated estimate of content quality and summary fidelity.
Compare Mode	Side-by-side Zyme view for analysis or concept comparison.
My Zyme Map	Visual graph of visited nodes and relationships.

11. Graph Addendum: Advanced Network Intelligence (Summary - Full version in original spec pages 21-24)

(This section summarizes the key ideas from the Graph Addendum for completeness. The full addendum provides much more detail on implementation strategies for these advanced features.)

- **11.0 Introduction:** Evolving Zyme from interconnected summaries to a dynamic, intelligent, self-enriching knowledge ecosystem.
- **11.1 Edge Governance & Quality:** Ensuring reliability of connections.
  - Edge Creation Policies (Explicit Automated, AI-Suggested Implicit, User-Curated).
  - Edge Confidence & Validation (Calculation, Thresholds, User/Editorial Feedback).
  - Edge Lifecycle Management (Versioning, Deprecation, Provenance).
- **11.2 Inferring Implicit Edges:** Expanding the semantic web with AI.
  - Semantic Similarity Links (`RELATED_TO` via embeddings).
  - Co-occurrence & Relational Pattern Analysis.
  - Topic Modeling & Thematic Connections.
- **11.3 Multi-Hop Queries & Pathfinding:** Uncovering hidden narratives.
  - Conceptual Framework (traversing multiple edges).
  - Example Scenarios (Legal Precedent, Scientific Influence, Cross-Domain Bridging).
  - Technical Considerations (Algorithms, Performance, UI).
- **11.4 Graph Analytics Powering Intelligent Features:**
  - "Knowledge Concierge" – Proactive Zyme Suggestions.
  - Personalized "My Zyme Map" Enhancements.
  - Improving Zyme Generation & Linking Quality (Centrality, Community Detection, Orphan Nodes).
  - Trend Detection & Emergent Insights (Organizational/Global Scale - Future).
- **11.5 Future Graph Capabilities:** Temporal Graph Analysis, Knowledge Graph Completion, Causal Inference, Integration with External KGs.
- **11.6 Summary:** The graph is Zyme's intelligent backbone, enabling continuous learning, adaptation, and user empowerment.