Time as a Calibrator of Credibility and Trust in Information Systems

by Dinis Cruz and ChatGPT Deep Research, 2025/10/02

Introduction

In an era of information overload and rampant misinformation, **time** emerges as a critical factor in determining what information we trust. Traditional approaches to credibility tend to evaluate a statement at the moment it is made -- a snapshot judgment of truth or falsehood. Dinis Cruz envisions a more dynamic paradigm: treating **every statement as an evolving entity** whose credibility is calibrated by the passage of time and the evidence that accumulates (or erodes) around it. In this vision, facts, opinions, hypotheses, and data points are not static declarations but living components of a knowledge ecosystem, each with attributes that can be **objectively extracted, tracked, and updated over time**. Trust is not a binary label stamped at publication; it is an emergent property that grows or diminishes as statements are corroborated, disproven, or refined by subsequent information.

This whitepaper articulates that vision for an audience of AI researchers, cybersecurity professionals, and journalists. We explore the philosophical underpinnings and technical concepts of using time as the **ultimate arbiter of credibility** in information systems. Rather than prescribing a particular implementation, we discuss the frameworks and principles -- from **information taxonomies** to **semantic knowledge graphs** -- that can enable such a system. Two real-world initiatives led by Dinis Cruz, **MyFeeds.ai** and **The Cyber Boardroom**, will serve as running examples. These projects demonstrate how concepts like the **LETS data pipeline**, **LLM-driven extraction**, and **persona-based modeling** can be applied to build information systems where provenance, transparency, and temporal evidence tracking are first-class features.

The goal is to show how, by encoding the temporal evolution of knowledge, we can fundamentally enhance credibility assessment. Over time, truth finds a way of asserting itself -- and our systems should be designed to capture that, providing a **clearer signal amid the noise**. In the following sections, we introduce a classification of information types, outline how AI (especially large language models) can extract and monitor these over time, and delve into the architecture of systems that put this into practice. We also highlight the importance of open-source, transparent infrastructure in earning trust. This paper is authored in a factual, professional tone, reflecting the style of prior whitepapers by Dinis Cruz, and is intended for publication on platforms like LinkedIn and Dinis's personal site.

The Temporal Dimension of Trust

Time has a unique role in the ecology of information: it is the **calibrator of credibility**. A statement made today might carry uncertainty; given weeks, months, or years, that statement could be bolstered by confirming data or undermined by contradictions. Consider a breaking news report that includes an eyewitness claim -- initially an unverified piece of information. Over subsequent days, investigations might provide evidence that either validates the claim as a fact or exposes it as false. The **credibility of the original claim is thus a function of time and evidence**. In science, a hypothesis must endure rigorous testing over time before it is accepted as proven; in journalism, initial reports are updated as more sources speak out or more documents come to light. Human societies have long used the test of time to judge ideas -- *"truth prevails in the end"* is a common refrain -- yet many modern information systems lack any notion of this temporal validation.

In current social media and even some news feeds, **context collapse** is common: a two-year-old claim might circulate without indication that it was later debunked, or a prediction might still be treated as credible despite subsequent evidence against it. Dinis Cruz's vision addresses this gap by making the *timeline* of each piece of information an integral part of how machines assess trustworthiness. Any datum -- whether a factual assertion, an expert opinion, a hypothesis, or a raw statistic -- should carry with it a **history**: when it was first stated, what corroborations or refutations have appeared, and how its status has changed. By **explicitly tracking the evolution of information**, systems can present users with not just a claim, but the *current state of that claim's credibility*.

In practical terms, this means building data architectures where **statements are objects with properties and links** that evolve. A claim might start with a low credibility rating, essentially a hypothesis awaiting verification. If multiple reputable sources later confirm it, the system elevates its status (and can even reclassify it from "hypothesis" to "fact"). Conversely, if credible evidence disproves it, the statement can be tagged as "disproven" or its credibility score lowered accordingly. Importantly, time-based trust doesn't imply that older information is automatically more credible -- rather, it means that **older information has simply had more opportunity to be tested**. A long-standing claim that has been continually supported by evidence earns a kind of trust that a fresh, untested claim cannot yet possess. By the same token, an old claim that has never been tested or that has languished without scrutiny might actually be less credible than a newer claim backed by immediate solid evidence. Thus, time is the calibrator in conjunction with evidence: it is the framework within which evidence accumulates.

From a philosophical standpoint, this approach echoes the scientific method and investigative journalism -- continual gathering of proof and revisiting of prior assumptions. It acknowledges a reality often lost in AI systems: **knowledge is provisional** and our confidence in it should be proportional to the journey it has undergone. What Cruz proposes is essentially to bake that epistemological humility into information systems. Trust becomes a **dynamic metric**. Users (be they researchers, security analysts, or readers) could see not only *what* is known at a point in time, but *how* it came to be known and how that knowledge has changed. This temporal awareness is especially pertinent in cybersecurity and AI, where new vulnerabilities or discoveries can overturn "facts" quickly, and in journalism, where narratives evolve as stories develop.

By treating time as a first-class dimension, we enable a richer form of credibility -- one that can be visualized as a timeline or knowledge graph rather than a static label. The sections below discuss how to systematically classify information and harness large language models and knowledge graphs to implement this vision.

Classifying Information: Facts, Opinions, Hypotheses, and Data

A foundational step in building time-aware credibility systems is establishing a clear **taxonomy of information types**. Not all statements are created equal -- a verifiable fact is different from a personal opinion, which is in turn different from a hypothesis or a raw data point. By categorizing statements, an system can apply the appropriate handling and validation logic to each. Dinis Cruz emphasizes four primary categories: **fact**, **opinion**, **hypothesis**, and **data**. Below we define each and explain their roles:

- **Fact:** A statement about reality presented as objective truth, ideally supported by evidence. Facts are assertions that, in principle, can be verified or falsified. For example, *"The company's servers were breached on July 15, 2025"* is a factual claim. Facts initially may come with a confidence level (especially if just reported), and over time they can be corroborated by further evidence or challenged by contradictions. In our system, a fact would be linked to its sources (documents, interviews, sensors, etc.) and marked with its verification status. If later reports confirm it -- say, an official investigation verifies the breach -- the fact's credibility strengthens. If evidence emerges that the date was wrong or the breach never occurred, the statement's status would be downgraded (e.g., labeled as erroneous or retracted). Essentially, facts are **statements awaiting or bearing verification**, and their truthfulness is a function of evidence). Maintaining **traceability of facts to their evidence sources** is critical to this process. By tracking provenance, we ensure that every factual claim can point to *why* we believe it (or did at one time), aligning with Cruz's focus on transparency and provenance to minimize error and "hallucination" in AI.
- **Opinion:** A statement of personal belief, interpretation, or judgment, which by nature is subjective. Opinions can be expert assessments (*"In my view, this security threat is being exaggerated"*), editorial commentary, or individual preferences. They cannot be "proven" true or false in the same way facts can, but they can be more or less persuasive or widely accepted. In an information credibility system, opinions are handled differently: rather than verifying them, the system might track *who* holds the opinion, their expertise, and how that opinion may shift over time or differ from other viewpoints. Opinions often add context or insight around facts (e.g., an analyst's opinion on why a breach happened). Classifying a statement as opinion ensures it isn't conflated with factual reporting. Large language models can be trained to detect opinionated language or phrases indicating subjectivity (e.g., "I think," "it is likely that," or tonal indicators), helping to label these correctly. Over time, one could even see how opinions trend -- for instance, initially a lone opinion might later become consensus (many others echo it) or remain controversial (sharply divided opinions). **Persona-based modeling** (discussed later) becomes crucial

here, as understanding the source's identity and bias is key: the credibility of an opinion often depends on who expresses it. A statement like *"Our systems are secure enough"* carries different weight coming from a company's CEO versus an independent security researcher. Thus, opinions in the system carry metadata about their source and context, enabling users to factor in biases and perspective.

- **Hypothesis:** A conjecture or tentative explanation that requires validation. Hypotheses often appear in investigative contexts (security analysts theorizing about an attack vector, scientists proposing a link between variables, journalists suspecting a cover-up). An example might be, *"The breach might have been an inside job,"* stated before any proof is available. Hypotheses are essentially questions framed as statements -- they signal *uncertainty and an invitation for further evidence*. In Cruz's approach, hypotheses are explicitly tagged as such and occupy an important place in the knowledge graph: they are nodes that expect evolution. As time passes, a hypothesis can be supported by facts (which might graduate it to accepted theory) or refuted by facts (leading to its rejection). One can imagine the system automatically updating a hypothesis's status when certain conditions are met -- e.g., if forensic data later show external IP addresses, the hypothesis of an inside job gets a lower credence or a "disproven" mark. Notably, hypotheses often drive the collection of new data. In the **Interactive Report Assistant** (one of Cruz's projects for AI-guided reporting), the AI explicitly captures hypotheses and even outstanding questions during a consultation, building a *knowledge base* that distinguishes between confirmed facts and speculative point. This structured capture means that the resulting report can label which findings are definitive and which are possible issues to investigate. By treating hypotheses as first-class citizens, an information system encourages a scientific mindset: everything is up for re-evaluation as new evidence comes in. Over a long term, tracking hypotheses can reveal how knowledge advances -- for example, a hypothesis from five years ago in medical research might now be a well-established fact, or might have been debunked, and that journey should be traceable.

- **Data (Data Point):** A raw observation or measurement -- often numeric or categorical -- presented without interpretation. Data points are the building blocks of facts. For instance, *"Server logs show 5,000 failed login attempts between 1-2 AM"* is data. By itself, data may not be meaningful until placed in context (is 5,000 high? does it indicate an attack?). Data can also be statistical results, experimental readings, or quotes from sources. In our taxonomy, data points are captured and stored as evidence nuggets. They often feed into facts (supporting evidence for a factual statement) or can lead to new hypotheses (*given this unusual metric, could something be wrong?*). Ensuring data integrity and tracking its source is a key part of provenance. Data is often time-stamped, and its credibility might hinge on how it was collected (e.g., a properly calibrated instrument vs. an anecdotal report). In a dynamic credibility system, raw data might be reinterpreted over time: for example, an anomaly in data might later be explained by a calibration error (thus the data point would be flagged as faulty), or multiple independent datasets might confirm the same trend (boosting confidence in those numbers). Automation through AI can assist in extracting data points from text (via techniques like

OCR for numbers in documents, or pattern recognition in text for "X% increase" statements) and storing them in the knowledge graph along with units and context. Over time, linking data points to the claims they support or refute is crucial. A single data point might be an outlier, but a time series or repeated measurement can solidify a fact. Hence, the **temporal tracking of data** itself -- noting how a metric changes -- can calibrate trust (for instance, a sudden spike in a security metric might at first seem like an attack, but if it drops the next day, the interpretation changes).

By classifying statements into these categories, an information system gains clarity on how to treat each piece. **Facts** and **data** demand verification and are the basis of objective truth-seeking; **opinions** require context and source awareness; **hypotheses** call for monitoring and future resolution. This taxonomy is not just theoretical -- it is being operationalized in tools. The **Interactive Report Assistant** for example, uses a conversational AI to extract and differentiate facts, assumptions/hypotheses, questions, and evidence in real time while an expert conducts an assessment. It builds a structured representation (almost a mini knowledge graph) where each piece of captured information is labeled appropriately -- facts here, open questions there, etc. -- and even keeps track of *which statements have been confirmed by the user and which are tentative*. This ensures that when the final report is generated, every claim is either backed by confirmed input or clearly marked as an open issue, with the provenance of each fact traceable to the conversation snippet or document it came from.

The ability to extract such a taxonomy from raw text is greatly enhanced by **Large Language Models (LLMs)**, which we discuss next. But even before the AI gets involved, having a human-understandable classification sets the stage for how information will flow through the system. It aligns with the principle that *different types of knowledge have different lifecycles*, and by recognizing that, the system can calibrate trust accordingly. A fact might have a lifecycle of verification steps; a hypothesis has a lifecycle of testing and either confirmation or abandonment; an opinion might shift with perspective or remain constant with its author; data points can accumulate into trends. Time affects each of these in unique ways, and a robust taxonomy is the first tool to manage that complexity.

Extracting and Evolving Knowledge with LLMs

Modern **Large Language Models** have demonstrated remarkable ability to read and interpret unstructured text. In the context of time-calibrated credibility, LLMs serve as the engine that **extracts structured knowledge** from raw information and helps update it as new inputs arrive. The idea is to leverage AI to do what humans do when researching: read documents (news articles, reports, transcripts), identify key claims and evidence, classify them (is this a fact? an opinion? who said it?), and flag their relationships to other information (does this support a prior claim? contradict it? raise a new question?). LLMs can perform or assist in all these tasks, making it feasible to maintain a rich, constantly updating knowledge base.

Initial Extraction: When a new piece of content comes in -- say, a news article or an incident report -- an LLM can parse it and pull out the statements of interest. This involves natural language processing steps like entity recognition (finding the who/what/where), relation extraction (how entities relate, e.g. X attacked Y at time Z), and classification (is this sentence an assertion of fact or speculation?). For example, given a cybersecurity blog post about a newly discovered vulnerability, an LLM might extract: *Fact:* "A vulnerability CVE-2025-1234 was found in AcmeCorp's software;" *Opinion:* "The researcher believes it could be exploited widely;" *Data:* "70% of tested servers were affected;" *Hypothesis:* "It might be related to an earlier bug in a shared library." Each of these would be added as nodes or entries in the system, linked to the source document and time-stamped. Off-the-shelf LLMs (like GPT-based models or domain-tuned variants) can be prompted or fine-tuned to perform this multi-faceted extraction. Indeed, Dinis Cruz's projects use LLMs for content understanding tasks. In **MyFeeds.ai**, for instance, after raw articles are ingested, an LLM analyzes each article to extract **key entities, topics, and relationships**, effectively summarizing the article in a structured form. This creates a semantic representation (a mini knowledge graph) of each piece of content. Similarly, the **Semantic Content Filter** project uses an LLM (or a distilled model) to read web pages on the fly and generate a **semantic profile** of the page, identifying main topics and even classifying content by type (e.g. "this section is unverified information"). These practical uses underscore how LLMs can discern and label the components of information needed for our taxonomy.

It's worth noting that while LLMs are powerful, they can also make errors or "hallucinate" facts. Therefore, Cruz's approach couples AI extraction with human or systematic verification steps and **provenance tracking**. The LLM might propose that "Statement X is a fact supported by Source Y," but the system will store that linkage so it can be checked. In the Report Assistant, every fact the AI captures is immediately shown to the user for validation or correction, creating a feedback loop that refines the AI's output in real time. This human-in-the-loop design ensures the knowledge base being built is accurate and trusted by its users. The emphasis on **traceability of facts to their origins** cannot be overstated -- as noted earlier, the Report Assistant was designed to maintain provenance for each extracted fact, and more broadly, all of Cruz's startups place importance on **provenance and explainability** of AI outputs. By keeping the source links, the system allows any claim to be audited: a user can ask *"How do we know this?"* and the system can point to the supporting evidence. This directly contributes to credibility, as trust in the system's information is reinforced when users see that nothing is conjured from thin air -- every assertion ties back to an input source or confirmed user input.

Temporal Updates: Once the initial extraction has populated the knowledge base, the LLM's job is not done -- it now helps in **evolving that knowledge as new information arrives**. This is where time as a calibrator comes in. As the system ingests new content over days and weeks, it needs to reconcile it with existing knowledge. If a new source provides additional evidence for a fact, the system (with AI assistance) should link that evidence to the fact, possibly increasing a confidence score. If new content directly contradicts a previously stored claim, this conflict must be noted and ideally resolved (perhaps flagged for human review or majority-rules logic). LLMs can assist by analyzing the new information in context: for example, when a follow-up article comes, the model could recognize *"this statement from today's article refers to the same event as that statement from last week's article, but with a different detail"*. It might flag, *"Previously it was claimed 10,000 records were leaked; now this source says 5,000 -- discrepancy noted."* The system can then mark that fact as contested until further clarity.

One way to implement this is through a **semantic knowledge graph** where each node (representing an entity or claim) accumulates links to sources and evidence over time. LLMs can help by merging nodes that appear to refer to the same real-world fact (e.g., "CVE-2025-1234" in various phrasings) and by creating relational links (e.g., "CVE-2025-1234 is confirmed by Source X on Date Y"). The **graph approach** is central to Cruz's solutions: MyFeeds.ai, the content filter, and the report assistant all convert unstructured text into a structured graph form. By doing so, it becomes much easier to **track the state of a particular node** (like a claim or an entity) as evidence is attached to it. The graph is a living data structure, and because it's stored in a database (or even as JSON files via the MemoryFS/GraphFS approach[17][18]), the system can update it incrementally. For instance, MyFeeds aggregates news and represents each article's knowledge as a graph; if another article comes in about the same vulnerability, the system can connect those graphs, enriching the picture of that vulnerability across sources[18]. Over time, if one of those articles turns out to be erroneous and is retracted, that could also be encoded in the graph (perhaps a property on the edge from that source saying "retracted" with a date).

Reclassification: A particularly interesting aspect of using AI over time is the possibility of *reclassifying information as its nature changes*. A statement initially extracted as a *hypothesis* could later be reclassified as *fact* if evidence is found. The system might initially label *"It's likely an insider attack"* as a hypothesis. Later, if an investigation report definitively says "It was an insider," the system, via an update routine, could change that node from hypothesis to fact and annotate it with "confirmed by [source] on [date]." Likewise, opinions can shift to facts in some cases (for example, an expert's prediction that "X will happen" becomes a fact once X does happen). LLMs can be employed periodically or triggered by events to review the knowledge base: essentially asking the AI, *"Given this new document, do any existing nodes in the graph need to be updated or have their status changed?"* This is a complex task -- it requires maintaining consistent identifiers for the same real-world claims and having business logic about what constitutes sufficient evidence. Not all of this can or should be fully automated -- human oversight is valuable. However, the AI can do the heavy lifting of reading and comparing new text with stored assertions.

For example, the system might store a hypothesis node: *"Cause of outage: misconfiguration (hypothesis)"*. When a postmortem report arrives a week later, the LLM might detect sentences like "The outage was caused by a misconfiguration of the firewall." It can then signal that this hypothesis is now confirmed by that report, prompting the system to mark it as a fact (and maybe move the original hypothesis node to an archive or link it as "now proven"). The system might also notify users or maintainers of this change -- effectively providing a **news feed of knowledge status updates** ("Hypothesis H has been confirmed as Fact, by source Z"). In this way, time and AI work together to calibrate what the system deems true.

The **LETS pipeline** (Load, Extract, Transform, Save), which we will discuss more later, provides a structured way to carry out these regular updates[15][19]. Each time new data is loaded, the extraction and transformation steps include reconciling with existing saved knowledge. By breaking the process into discrete steps, it's easier to monitor and tweak how updates occur -- a design choice that favors transparency and control[20]. Dinis Cruz has highlighted that a structured pipeline improves *"transparency and tweakability -- critical for debugging AI decisions and maintaining provenance"*[20]. Indeed, when AI is used to adjust our knowledge base, we must be able to audit those adjustments. The provenance of every change (which AI suggestion or which source triggered it) should be logged, so that trust is maintained in the system's ongoing evolution.

In summary, LLMs act as both the **miners of knowledge** (extracting structured claims from raw text) and the **maintenance crew** (continually comparing new information against the old and adjusting the structure). They work in concert with human experts and curated rules to ensure the knowledge base doesn't drift into error. By automating extraction and updates, we can keep pace with the torrent of information in domains like cybersecurity or global news, where no single person could manually track all the threads. The result is an always-current, evidence-weighted map of what is known and unknown. In the next section, we delve deeper into that map -- the semantic knowledge graph and the pipeline architecture that supports it, including how it handles provenance, source bias, and evidence trails over time.

Semantic Knowledge Graphs and the LETS Pipeline

Central to operationalizing a time-calibrated trust system is a robust **information architecture** that can store content, context, and connections. Dinis Cruz's approach relies on **semantic knowledge graphs** as the backbone for representing information, and a well-defined processing pipeline (called **LETS: Load, Extract, Transform, Save**) to manage how data flows from raw inputs to structured knowledge[15][17]. These components work together to ensure that every statement is captured with its provenance, that relationships (like support or contradiction) between statements are explicitly modeled, and that updates can be applied systematically as time goes on.

Building a Knowledge Graph of Evidence

A *semantic knowledge graph* is a graph data structure where nodes typically represent entities or concepts (people, organizations, events, claims, etc.) and edges represent relationships or interactions between them (e.g., "reported_by," "confirmed_by," "contradicts," "part_of"). By representing information in a graph, we gain two advantages for our purposes: **interconnectivity** and **explainability**. Interconnectivity means any given piece of information doesn't live in isolation -- it's linked to sources, related facts, and broader contexts. Explainability comes from the graph's ability to show why something was suggested or how a conclusion was reached (following the edges back to evidence).

In Cruz's vision, whenever the system ingests an information source, it doesn't just store a blob of text. Instead, through LLM extraction, it **populates the graph** with structured data. For example, an article might produce nodes for the event it describes, the key actors involved, and the claims made, all connected by labeled edges indicating relationships (like "Actor A -> involved_in -> Event X"). If a claim is made, an edge from that claim node to the article's node could be labeled "asserted_in [source name/date]." If later another source confirms the same claim, another edge from the claim to that source is added, perhaps labeled "corroborated_by [other source]." Over time, the claim node accumulates a web of supporting or refuting links. This graphical approach directly supports **time-based credibility**: one can literally count or weight the number of confirmations vs refutations attached to a claim node, and factor in the credibility of each source node attached (we will address source credibility soon). The **semantic graph enables reasoning** -- for instance, the system can traverse the graph to find all evidence for a statement or to detect if two statements are in conflict (if they cannot both be true logically).

MyFeeds.ai provides a clear example of semantic graph construction. It takes in articles and *"analyzes the article to extract key entities, topics, and relationships, forming a semantic representation (a graph) of the article's content"*[11][12]. So an article about a new ransomware might result in graph nodes like *Ransomware X*, *Healthcare Industry*, *Country Y*, *Data Breach*, with relationships encoding facts such as "Ransomware X caused a data breach in Healthcare sector in Country Y"[12]. This graph is then stored via a specialized interface called GraphFS (graph filesystem)[21]. The graph acts as a **machine-interpretable summary** of the article[22]. Why is this useful? Because now MyFeeds can do content matching at the level of concepts rather than keywords -- and it can explain recommendations by pointing to the graph overlap[23][24]. For instance, if a user's profile graph (more on persona graphs later) has a node "zero-trust architecture" and an article's graph has a related node "zero-trust security model," the system can match them even if the exact words differ, and then tell the user *"Recommended because it discusses zero-trust, which is one of your interests."* This transparency in *why* content was shown builds trust in the system's curation[24]. The broader point is that the graph is the structure that makes such explainability possible. Similarly, in our credibility context, a graph would allow the system to explain *why* a statement is considered credible or not: e.g., *"This claim is considered credible because it has been independently reported by three sources (Nodes A, B, C) and no contradictory evidence is in the graph."* If contradictory evidence exists, that too can be shown: *"However, one source (Node D) disputes this claim, hence it is marked contested."* This is far more informative than a simple true/false tag and provides nuance that professionals and analysts need.

The LETS Pipeline -- Ensuring Structure and Provenance

The **LETS (Load, Extract, Transform, Save) pipeline** is a disciplined workflow for data processing that Cruz employs to handle content ingestion and analysis[15][17]. It is inspired by classic ETL (Extract, Transform, Load) but adapted for the needs of AI-driven knowledge extraction. The phases are:

1. **Load:** Gather raw data from sources. In practice, this could mean fetching RSS feeds, API data, web pages, PDFs, or any input. For example, MyFeeds periodically pulls in new articles from dozens of cybersecurity news feeds[25][26]. In the context of our credibility system, Load might happen continuously -- new information is always coming (news articles, social media posts, internal reports) -- so this stage deals with connecting to those sources and retrieving the content, possibly in real-time or in batches.
2. **Extract:** Parse and clean the raw data to get it into a usable form. This often involves text extraction (removing HTML boilerplate, splitting into sections or sentences), metadata extraction (capturing author, publication date, etc.), and initial filtering. In MyFeeds, after

loading, each item is parsed and its text and metadata are saved in a uniform storage (MemoryFS) and then converted into a JSON structure for further processing[27][28]. The idea is to have a structured representation of the content that can be fed into AI models. In our scenario, extraction also includes pulling out those facts/opinions/hypotheses as discussed earlier -- essentially preparing the inputs for the semantic analysis. By the end of Extract, we have the raw content distilled into something like: text segments, identified entities, and preliminary classifications.

3. **Transform:** This is where the heavy semantic lifting occurs -- using LLMs or algorithms to generate the knowledge graph entries and any additional annotations. The transform step takes the extracted content and applies the intelligence: it might call an LLM to generate the semantic graph of the content (as MyFeeds does)[12], or to classify each statement into our taxonomy, or to apply policies. In the Content Filter pipeline, for example, the transform step includes the LLM analyzing the page content and creating a semantic profile (a graph or metadata profile) of the page, including tags like "contains explicit language" or "phishing login form detected"[29][30]. In the credibility use-case, transform would link statements to prior knowledge (e.g., if it identifies a claim that's already known, it will connect it) and perhaps score them. If a source is known to be biased or unreliable, the transform step could note that -- e.g., *tag the incoming claims with a lower initial trust weight due to source reputation*. The transform phase is also where any complex logic like deduplicating information or reconciling conflicts can be executed. Essentially, transform turns input data into structured knowledge updates.
4. **Save:** Finally, the results of transform -- the updated knowledge graph nodes/edges, the cleaned content, the metadata -- are saved to persistent storage. This could be a graph database, a relational database, a set of JSON files, or a combination. Cruz's projects use abstractions like MemoryFS and GraphFS to simplify storing both file-like data and graph data uniformly[17][18]. Save is crucial for persistence (so that the next pipeline run knows the current state of the world) and for traceability. Saving doesn't just mean storing the final graph, but also logging the pipeline operations, recording timestamps, and possibly keeping previous versions (for auditing how something changed over time).

By enforcing the LETS structure, the system gains **transparency and modularity**. Each step can be monitored and improved independently. For example, if some false information is getting through, one can check: did we fail to extract correctly, or was our transform rule not catching a contradiction? Because the pipeline is stepwise, debugging is easier than in a monolithic process. As noted in the multi-startup strategy document, *"This structured pipeline improves transparency and tweakability -- critical for debugging AI decisions and maintaining provenance."*[20] The controlled processing also means you can implement checks like saving intermediate outputs for review. In an environment where trust is paramount, this is important. We don't want a black-box AI magically altering our knowledge base; we want a clear record of how each piece of data was handled, what the AI suggested, and what was ultimately stored.

Critically, **provenance** is woven throughout the pipeline. From the moment of Load, we tag every piece of information with its origin (source name, URL, timestamp, etc.). During Extract, if we break content into sentences, each sentence knows which document it came from. In Transform, when producing a knowledge graph node for a claim, we attach references to the source sentence or document node (the graph edge "asserted_in [source]" as mentioned). In Save, we ensure these references persist in the database. This way, any node or edge in the knowledge graph can be traced back to *who said it, where, and when*. Provenance is the bedrock of a credible system -- users and auditors can verify things for themselves. Dinis's emphasis on provenance in his designs (e.g., the Report Assistant explicitly stores which conversation snippet led to each fact[9]) aligns with best practices in both journalism and science, where citations and references are mandatory for trust.

Content Provenance, Source Bias, and Evidence Trails

With the graph and pipeline in place, the system can tackle the nuances of **source credibility and bias** as well as maintain **evidence trails over time**. Every source (be it a media outlet, a social media account, an internal document, or a sensor feed) can be represented as an entity in the graph with attributes (e.g., source type, known bias, reliability score). External data can be used to seed these attributes -- for instance, a news source might be annotated with its political leaning if known, or a scientific journal with its impact factor, etc. More dynamically, the system can compute a **trustworthiness score** for sources based on how their information pans out over time. If Source A's claims are frequently confirmed by others and rarely retracted, Source A's credibility rating could increase. If Source B has numerous claims later proven false or exaggerated, its rating would fall. These ratings can feed into how new information is treated: an unverified claim from a historically trustworthy source might be given more initial credence (or flagged as high priority to fact-check), whereas one from a dubious source might be tagged with a warning or held until corroboration appears. For example, the content filter could be configured to insert a banner like "**Warning: This news is from a source with a history of unreliability**." This kind of feature would be invaluable for readers and analysts trying to quickly judge new information.

Analyzing **source bias** goes beyond just reliability scores. It also means understanding perspective: a source might be consistently skewing facts in a certain direction (e.g., minimizing certain risks or always pushing a particular narrative). An AI system can be trained to detect tonal or framing patterns that indicate bias. Over time, if one compares how different outlets report the same event, one might annotate "Outlet X tends to emphasize cybersecurity threats, while Outlet Y downplays them." This context can be included in the knowledge graph (perhaps as a "bias profile" node linked to the source). Then, when that source publishes something, the transform step can note, for example, *"Source X described this as 'massive' breach, but their bias profile suggests they often use hyperbole."* The system might then either normalize the language or flag to the user that *"Source X tends to exaggerate, consider verifying details from another source."* In MyFeeds, while the current focus is on personalization and relevance, the underlying tech of knowledge graphs and LLM analysis could well be extended to capture sentiment and bias in content -- especially since it's already extracting semantic meaning[12]. Indeed, the open nature of the platform means adding a "bias detection" transform is feasible.

Content provenance in this context ensures that when we talk about a claim, we can always list its lineage. For example: *Claim:* "ZeroTrustCorp suffered a data breach leaking customer records."

Provenance might tell us: first appeared in Source A (a tweet by a security researcher at time T1), then was reported by TechNews (article at time T2 citing Source A), then confirmed in an official press release (at time T3). The knowledge graph would have nodes for each source with edges like *Claim -> reported_in -> TechNews article (T2)*, *Claim -> reported_in -> press release (T3)*, and *TechNews article -> cites -> researcher tweet (T1)*. Traversing the graph, an analyst can see the chain of reporting -- essentially a **transparency timeline**. This helps in assessing credibility: by the time of the press release confirmation, the claim is pretty solid. Early on at T1, it was just a researcher's allegation (maybe credible if the researcher is known, but still single-source). If someone sees the claim on TechNews, the system can show: "source of this claim is a tweet by X." This alerts the user that the article might be based on a single external source and encourages caution until further confirmation. Maintaining such provenance links addresses one of the challenges in today's information landscape: repeated information can create an illusion of multiple sources when in fact everyone is quoting the same original source. The knowledge graph can cut through that by literally linking all those mentions to the common origin, preventing false amplification of confidence.

Finally, **evidentiary support over time** is recorded as trails in the system. Each claim or hypothesis accumulates an **evidence trail** -- a list of supporting and opposing pieces of evidence along with timestamps. This could be visualized to users as a timeline ("Here is the history of this claim"). Such a trail might look like: - *Day 0:* Statement first made (by Source X). - *Day 1:* No verification yet (status: unverified). - *Day 2:* Another outlet Y reports the same (status: two reports, limited verification). - *Day 5:* Official source confirms (status: confirmed fact). - *Day 10:* An analysis questions a detail (status: mostly confirmed, minor contention). - *Day 20:* Further analysis resolves the contention (status: confirmed, consensus reached).

This running log is essentially extracted from the evolving knowledge graph. It provides a narrative of how the truth unfolded, which is invaluable for deep analysis, audits, or simply understanding context. For example, journalists verifying a story can use this to ensure they've seen all angles. Cybersecurity professionals investigating an incident can track how early assumptions changed as forensic data came in. AI researchers could feed these trails into models to study how information solidifies or decays, perhaps improving the models' own calibration of uncertainty.

MyFeeds.ai and the other systems already embody parts of this vision. MyFeeds doesn't just throw articles at users; it curates them based on a structured understanding and is very mindful of explaining *why* something is surfaced[24]. The Report Assistant ensures every generated line in a report ties back to input evidence[9], which is essentially leaving an evidence trail in the final product for the reader or auditor. The Semantic Content Filter even can insert annotations into content, like highlighting suspicious or unverified parts of a web page[14][31] -- this is a real-time manifestation of evidentiary labeling, warning users at the moment of consumption. All these contribute to an ecosystem where trust is continuously assessed and communicated.

In summary, the combination of **knowledge graphs** and the **LETS pipeline** yields an infrastructure where information is systematically ingested, analyzed, and stored with rich relationships and provenance. This sets the stage for building applications that leverage time-based credibility. Next, we will discuss how **persona-based modeling** integrates with this to tailor the system's outputs to different users and use cases, and then we'll dive into the specific examples of MyFeeds.ai and The Cyber Boardroom to illustrate these principles in action.

Persona-Based Modeling and Contextual Trust

An innovative aspect of Dinis Cruz's approach is the use of **persona-based modeling** -- designing AI systems that understand and simulate the perspectives of different users or stakeholders[32][33]. In the context of credibility and information analysis, persona modeling plays a dual role. First, it helps tailor what information is presented and how it is presented, in order to maximize relevance and comprehension (which in turn affects how information is trusted by the end-user). Second, it allows the system to factor in biases and preferences associated with those personas, which is crucial when evaluating credibility. Essentially, the truth may be objective, but **the reception of information is subjective** -- who the user is can influence what they consider credible or what needs extra explanation. By modeling personas, we can adapt the system's behavior accordingly without compromising the underlying factual integrity.

User Persona Profiles: MyFeeds.ai provides a clear example of persona-driven content delivery. The system builds a **profile graph for each user or persona** representing topics of interest, role, industry, and even content preferences[34][35]. A CISO at a bank might have a persona graph weighted towards regulatory compliance, financial sector threats, and high-level summaries, whereas a software engineer might have a persona focused on technical vulnerabilities, open-source tool news, and detailed analysis. These persona graphs are themselves part of the knowledge graph ecosystem -- they are essentially filters or lenses through which the global information graph is viewed. MyFeeds uses the intersection of the content's semantic graph and the user's persona graph to decide which articles to recommend[23]. This means the system doesn't just fling "top news" at everyone; it picks what matters to *you*, and it can even justify that choice ("we recommended this because it matches your interest in X")[24]. This increases trust in two ways: the user sees that the system understands their needs (making it feel like a credible assistant rather than a random feed), and the user can verify that the recommendations aren't arbitrary or biased -- they're tied to the user's own stated preferences. In a world of AI, giving users that transparency and control loop fosters trust in the system itself.

Persona Simulation for Communication: The Cyber Boardroom takes persona modeling a step further by using it to simulate how different stakeholders perceive information. One of its distinguishing features is the ability to model perspectives of roles like CFO, CEO, CTO, etc., and even simulate a boardroom Q&A where the AI adopts those personas[32][36]. For instance, a security leader can input a report and ask the system to respond as a skeptical CFO might -- *"If presented this way, the CFO might worry about X or misunderstand Y"*[36]. This is immensely valuable for **trust-building in communication**. It forces the technical communicator to address likely concerns and clarify points before the real meeting, ensuring that when the information is delivered, it resonates and is convincing. Here, time is involved in a different sense: through iterative practice and refinement, trust is built up between the technical and non-technical sides. Over multiple board meetings, if the board consistently gets clear, persona-tailored answers, their trust in the security team's information increases. The Cyber Boardroom's persona simulation can be seen as a training ground for that trust -- the AI helps the human deliver information in the most credible way possible for each audience[37]. It essentially encodes empathy and perspective-taking into the information delivery process, which is a profound application of AI beyond just data crunching. By modeling "what if I am a CFO hearing this?", the system uncovers potential credibility gaps (e.g., jargon that a CFO might not trust or understand, or missing business context that a CEO would need). Filling those gaps ahead of time means the eventual communication is more **trustworthy** to its recipients.

Incorporating Source and Persona Bias: Persona modeling also offers a structured way to deal with bias. We discussed source bias above -- now consider that users (personas) have their own biases too. A journalist might inherently distrust information from an anonymous source unless verified, whereas a corporate PR officer might be more skeptical of negative news until fully confirmed. By encoding such tendencies into persona profiles, the system can adapt its behavior. For example, a "Journalist" persona might cause the system to highlight "unverified" labels more prominently and provide easy access to source documents, aligning with a journalist's training to double-check facts. A "Casual Reader" persona might prefer the system to automatically filter out anything not yet corroborated (to avoid spreading rumors). The persona preferences could include thresholds for what confidence level is needed to show a claim as a fact. Another scenario: consider political or cultural bias -- if a user leans a certain way, they might initially distrust sources from the other side. While the ultimate goal is to present objective truth, a persona-aware system can *acknowledge* these biases in how information is presented. It might say, for instance, *"This claim comes from a source you don't usually follow, but note that three independent sources from across the spectrum have confirmed it"* -- essentially nudging the user to trust information that is well-evidenced even if it comes from an unlikely place. The key is not to reinforce bias, but to transparently work with it to gain the user's trust in the facts. Over time, an evidence-driven system might even help broaden a user's trust network by showing how reliability can come from many quarters if tracked objectively.

Persona in AI Reasoning: Persona modeling isn't just for the user interface; it can be part of the AI's reasoning under the hood. When evaluating a claim, the system might apply different heuristics based on context: for a cybersecurity professional user, the system might emphasize technical evidence (logs, indicators of compromise), whereas for a board-level summary, it might emphasize authoritative statements (regulatory findings, law enforcement confirmations). The underlying data is the same, but the weight given can shift to match what that persona finds credible. Dinis's framework for defining personas includes factors like role, expertise, cultural context, and goals[38]. For example, a persona with low technical expertise but high need for certainty (like a board member) might trigger the system to only present facts that are confirmed and to avoid technical jargon, focusing instead on analogies and business impact[39][40]. On the other hand, a highly technical persona might be shown provisional findings with the caveat that they are unconfirmed, because that persona can handle uncertainty and may want the heads-up. By **integrating persona profiles into the LLM's prompts or the graph query process**, the system can effectively shape its outputs to maintain credibility in the eyes of the beholder. This is reminiscent of how a human analyst would brief different audiences: it's not about changing the facts, but about how you frame them and what you choose to highlight or contextualize.

Feedback Loop and Persona Evolution: Personas themselves are not static -- they can evolve as the system learns from user feedback. MyFeeds.ai, for instance, updates a user's persona graph based on which articles the user reads or finds useful[41][42]. If the user consistently ignores news about a certain topic, the system might down-weight that topic in their profile; if they always click items about a new subject, it might add that subject to the interest graph. In terms of trust calibration, if the user frequently gives feedback like "this source is not credible" or "I don't believe this", the system could incorporate that into the persona bias (though ideally it would also aim to show why something is credible if evidence supports it -- feedback could also highlight places where the system needs to provide more evidence to convince the user). This adaptability means the credibility system becomes personalized over time -- not in the sense of filtering truth (we must avoid just creating echo chambers), but in tailoring *how* it engages the user to build trust in verified information.

To ground these ideas, let's incorporate an example scenario: Suppose the system is analyzing a news article that makes a bold claim about a cyber-attack, sourced from an anonymous intelligence report. A **skeptical persona** (say an experienced analyst) might be given that information with a caution: *"Preliminary claim from an unverified source, treat with caution until more info emerges."* The system might even suggest questions that this persona would likely ask (because it knows the analyst persona values certain evidence): *"No malware hashes or IOCs provided -- you might want to see those for proof."* In contrast, a **general executive persona** receiving information on the same incident might get a different treatment: *"Early reports suggest a cyber-attack; confirmation pending -- we will update you when official statements arrive."* The exec doesn't need technical details (and might distrust them if confusing), but does need the assurance that the information is being validated. Both personas ultimately get the same outcome (if later an official report confirms the attack, both will be informed of that fact), but the journey to that point -- and how the uncertainty is communicated -- differs to maintain credibility for each audience.

In summary, persona-based modeling is about **contextualizing credibility**. It acknowledges that trust is partly in the eye of the beholder and that a one-size-fits-all approach to presenting information can fall short. By designing AI systems that simulate and adapt to personas, Dinis Cruz's projects ensure that the sophisticated analyses (knowledge graphs, evidence tracking, etc.) actually translate into insights that different users trust and find useful. The Cyber Boardroom and MyFeeds.ai both exemplify this: one by translating tech to business language to earn trust at the board level[43][40], the other by curating feeds that feel almost eerily relevant to each user[44]. With persona modeling covered, we now move to highlight these real-world systems explicitly, drawing out how they implement the principles discussed and how they are paving the way for a new standard in information credibility.

Real-World Examples: MyFeeds.ai and The Cyber Boardroom

To illustrate the principles discussed, we turn to two of Dinis Cruz's ongoing projects: **MyFeeds.ai** and **The Cyber Boardroom**. Each addresses a different problem space (personalized information feeds and executive cybersecurity communication, respectively), yet both are built on the core ideas of semantic analysis, provenance, and adaptive presentation of information. They serve as concrete prototypes of how time-calibrated credibility and trust can be encoded in practical systems.

MyFeeds.ai -- Personalized, Provenance-Rich Intelligence Feeds

MyFeeds.ai is designed to combat information overload for professionals in cybersecurity, tech, and related domains[45]. The problem: there is an endless flood of news and alerts, but busy experts have limited time to sift signal from noise. Traditional news aggregators lack fine personalization and often miss context, while raw feeds and keyword alerts return heaps of irrelevant results[46]. MyFeeds tackles this by delivering **highly personalized news feeds** that are tailored to each user's role and interests, accompanied by concise summaries[47][48]. Under the hood, it exemplifies many of the concepts we've covered:

- **Semantic Knowledge Graph Curation:** Instead of naive keyword matching, MyFeeds uses a semantic pipeline. It ingests many sources (RSS feeds, APIs) frequently, ensuring timely updates[25]. Each article is parsed and then transformed by an LLM into a semantic graph of its content[11][12]. So, all content becomes structured data linked by meaning. For example, an article on a new vulnerability might have graph nodes for the vulnerability identifier, affected software, potential impact, etc. This structured approach allows MyFeeds to match content to users at the concept level: it knows what the article is *about*, not just what words appear. This dramatically improves relevance -- a user interested in "supply chain attacks" will be shown an article about a compromised NPM package, even if the article doesn't use the phrase "supply chain," because the graph understands the relationship (NPM package hack is a type of software supply chain issue)[49].
- **LETS Pipeline and Memory Graph Storage:** MyFeeds leverages the LETS pipeline to handle its data flow[15]. **Load:** it fetches new content periodically. **Extract:** it parses articles and stores raw text and metadata in a uniform way using MemoryFS (an in-memory filesystem abstraction)[50]. **Transform:** it generates the semantic graph (using what Cruz built as GraphFS for storing graph data uniformly)[21]. **Save:** it stores both the raw content and graph representation. This disciplined pipeline means MyFeeds can process information systematically and at scale -- dozens of feeds, thousands of articles, continuously. The use of serverless functions and lean infrastructure means it only consumes significant resources when processing new content (keeping costs low and scalability high)[51][52]. This is critical for a system intended to monitor information streams around the clock.
- **Content Provenance and Explainability:** MyFeeds emphasizes provenance in its recommendations[15]. Each piece of content retains a link back to its source, and when the system generates a summarized newsletter or feed for a user, it can explain *why each item is included*. For instance, a daily brief email might list 5 headlines with 2-sentence summaries; next to each, MyFeeds can indicate the source (e.g., "via Krebs on Security, reported 2 hours ago") and a rationale ("Chosen for you because it relates to [Cloud Security] in your profile") [24]. Users are not left guessing why something showed up -- the system is transparent about its reasoning. If a user wants to drill down, they could see the semantic connections (e.g., the article discusses AWS breaches and the user's profile has interest in cloud breaches). This

fosters trust: professionals can rely on MyFeeds because it's not a black box, it's an assistant that shows its work. Moreover, provenance tracking means if there are conflicting reports on a story, MyFeeds could potentially show both and attribute them correctly, helping the user be aware of disagreements or evolving stories (a future enhancement might be to explicitly highlight when a story in yesterday's brief has an update or correction today).

- **Adaptive Persona Feeds:** MyFeeds builds a graph for the user's persona (interests, role, etc.)[34] and continuously refines it based on feedback[41]. Suppose a user is an *investor* focusing on cybersecurity startups; their feed might prioritize funding news, major breaches with business impact, and tech trend analysis. If they start frequently reading AI-related security articles, MyFeeds will learn and expand their profile to include that. The system can also produce multiple persona feeds from the same content pool. As noted in Cruz's briefing, one CISO user asked for multiple feeds: one for themselves, one simplified for their non-technical executives, and one for their technical team[53]. MyFeeds delivered this by creating distinct persona profiles for each audience type and repackaging the same source content appropriately[53]. This demonstrates how powerful the combination of semantic graphs and persona modeling is -- the content can be filtered and reframed without manual effort, and each audience gets the information in the form they trust and understand. The CISO's bosses got a high-level brief (trust through clarity and relevance, no jargon), while the technical staff got a detailed feed (trust through completeness and technical accuracy).
- **Evolving Content and Alerts:** Because MyFeeds runs continuously, it can catch the evolution of stories. If a Monday brief included "Company X breach reported, cause unknown," and by Tuesday there's an update "Cause identified as phishing," the Tuesday brief can include that development, possibly even referencing that it's an update to yesterday's news. This temporal linking is exactly what time-calibrated credibility is about. MyFeeds could in principle tag the Monday item as a hypothesis ("cause unknown") and then automatically follow up when the cause is confirmed, adjusting the status of that story from speculative to factual. While not explicitly stated, the underlying tech is ready for that kind of feature. In fact, Cruz originally built MyFeeds to support the Cyber Boardroom -- he needed a steady stream of content to discuss in board meetings without using sensitive data[54]. This means the feed had to be topical and up-to-date to simulate real-world scenarios. The MVP of MyFeeds was publicly demonstrated (with example newsletters like a "CEO Cybersecurity Brief" and an "Investor Tech Digest") and garnered positive feedback for its uncanny relevance[44]. This validation shows that focusing on semantic relevance and user context produces a qualitatively better information experience than generic feeds.
- **Open-Source and Integration:** Importantly, MyFeeds (like the other projects) is built with an open-source core and a serverless-friendly architecture[55][56]. This means organizations could run their own instance or extend it. For instance, an enterprise might plug in internal data sources (threat intel feeds, internal incident reports) into MyFeeds to create a hybrid feed combining external and internal intel, all analyzed under the same graph framework. The serverless approach (deploying as cloud functions, etc.) means it's cost-efficient and scales with usage[52]. An open-source foundation also fosters trust: users (especially cybersecurity pros) can inspect how MyFeeds processes data and be confident there are no hidden agendas or leaks[57]. In the security community, open tools are often preferred for this reason

-- they can be vetted[58]. MyFeeds's design reflects this ethos by focusing on interoperability (MemoryFS/GraphFS making it easy to connect systems) and transparency.

In essence, MyFeeds.ai demonstrates how to deliver the *right* information at the *right* time in the *right* way, which is the crux of building trust. It doesn't overwhelm, it doesn't hide its logic, and it evolves as the world and the user's needs evolve. It shows that by using semantic understanding and tracking provenance, an automated system can actually earn the user's confidence in a domain where trust is paramount.

The Cyber Boardroom -- Trust at the Nexus of Tech and Business

The Cyber Boardroom addresses a very different challenge: bridging the communication gap between cybersecurity experts and business executives (such as corporate boards)[59]. Here the trust we're concerned with is the trust business leaders have in the information and advice coming from their security teams (and vice versa). Often, miscommunication or lack of context leads to misaligned expectations and skeptical board members who are not sure whether to take cybersecurity recommendations seriously. The Cyber Boardroom uses GenAI (generative AI) as an **intelligent translator and facilitator** in this relationship[60]. It exemplifies time-calibrated credibility in a more human-centric sense: through iterative dialog and persona simulation, it ensures that over time, the messages delivered to boards are consistently understandable, relevant, and backed by appropriate evidence -- all of which build credibility and trust in the eyes of leadership.

Key features and principles of The Cyber Boardroom include:

- **Bidirectional Translation:** The system works both ways -- it helps technical teams explain things to the board in plain business terms, and helps boards ask the right questions or get clarifications to relay back to technical teams[39][61]. For example, a CISO can input a detailed risk assessment or a technical report. The AI then produces a polished executive summary emphasizing business impact and critical points, stripped of jargon[62]. It might use analogies and focus on strategic priorities so that board members immediately grasp the significance without wading through technicalities[39]. Conversely, if a board member types a question like "Why do we need to invest $X more in cybersecurity next quarter?", the AI can interpret that against the technical data and either formulate an answer or translate it into actionable queries for the security team[61]. This translation is not done blindly -- it leverages a knowledge base of cybersecurity concepts mapped to business outcomes that the Boardroom maintains[40]. Essentially, the system "understands" common cybersecurity topics and how they relate to things a board cares about (financial impact, legal risk, operational continuity, etc.)[40]. Over time, as it's used in an organization, it can even incorporate specifics of that organization's context (like past incidents, industry regulations, the company's risk appetite) to make the translations more tailored.
- **Persona Simulation and Training:** As mentioned earlier, a standout feature is the persona simulator for different stakeholders[63]. The user (say a CISO preparing for a board meeting) can run a mock presentation through the AI and get simulated responses from various personas: *"As a CFO, I'm concerned about the cost implications here,"* or *"As an outside director*

with legal background, I need clarity on regulatory exposure."[64]. This allows the security leader to iteratively refine their message. It's essentially a **sandbox to preemptively address skepticism**. By the time the real meeting happens, many of the rough edges have been smoothed. This builds trust in two ways: the board gets a clearer, well-thought-out presentation (so they trust the presenter more), and the security leader feels more confident and in tune with the board's perspective (so they trust the board to understand, creating a virtuous cycle of open communication). Over repeated use, this can significantly improve the relationship; as the whitepaper notes, *"Over time, this can significantly improve mutual understanding and trust between tech and business leadership."*[65]. The time element here is the iterative practice and learning: the persona feedback loop effectively compresses what might take years of trial-and-error presentations into a much shorter learning curve.

- **Knowledge Graph and Memory:** Under the hood, The Cyber Boardroom likely uses similar tech to MyFeeds for knowledge management. It maintains a knowledge base of cybersecurity concepts and maps them to business outcomes[40]. This sounds like a knowledge graph where nodes could be things like "ransomware attack" linked to "business downtime" and "revenue loss" as outcomes, etc. It factors in attributes like the stakeholder's role, concerns, and even personal style if known[40]. All this context forms a persona profile that the LLM uses to shape its output[40]. In essence, the system encodes things like: *"If audience is CFO, prioritize cost/benefit language; if General Counsel, highlight legal risk; if CTO, you can include some technical detail,"* and so forth[66][67]. This knowledge base would need to be built and refined over time -- presumably the system can learn from each interaction, noting what follow-up questions were asked by the real board and incorporating that into future simulations. Also, because the platform is used as a central hub, it can accumulate an institutional memory (with appropriate security) -- e.g., it could recall that *"Last quarter the board was particularly concerned about supply chain attacks"* and ensure that is addressed upfront in the next briefing if relevant. This temporal memory aspect again fosters trust: the board sees continuity and attentiveness to their past concerns, reinforcing that the tech team is responsive and thorough.
- **Integration with Live Content:** The Cyber Boardroom doesn't operate in isolation; it can pull in live data like news or threat intel to enrich board discussions. In fact, one of the MVP features was the ability to ingest RSS news about cyber incidents and produce a tailored briefing for a given persona[68]. This was developed in tandem with MyFeeds.ai[69], showing synergy between the projects. For example, if a major cybersecurity incident is in the news on the day of a board meeting, the CISO can query the Athena bot (the Boardroom's AI assistant persona, as per demos) about that incident and get a quick rundown plus any relevant context on how it might affect their company, all in board-friendly terms. The system thus keeps the discussion timely and grounded in reality. From a credibility standpoint, this means board members aren't left in the dark about something they heard on the news -- the AI proactively brings it into the conversation with analysis, which increases the board's trust that the security team is on top of current events. It also showcases transparency: *"Yes, we're aware of breach X that's in headlines; here's what it means for us."*
- **Outcome: Better Decisions and Trust Building:** The ultimate measure of The Cyber Boardroom's success is whether it *revolutionizes board-level decision-making* in cybersecurity,

as intended. Early feedback from CISOs who tried the persona simulation has been that it's "especially insightful"[70] -- it surfaces concerns they hadn't thought of and helps them see their communication from the outside perspective[64][65]. This introspection leads to more polished communication. A polished, clear, evidence-backed presentation to the board results in fewer misunderstandings and more informed questions. Over successive quarters, the board can see a track record: "the security team consistently communicates well, provides data to back up claims, and addresses our concerns." That consistency is how **credibility is built over time**. It's analogous to how a news outlet builds trust by being accurate again and again. Here, the security program builds trust with leadership by communicating effectively again and again, aided by AI. Meanwhile, the security team gains trust in the board too -- seeing that when they articulate risks in business terms, the board responds constructively (e.g., approving budgets for critical defenses). This mutual trust can ultimately lead to better cybersecurity posture, as initiatives are understood and supported at the highest level.

- **Open-Source and Security:** Just like MyFeeds, The Cyber Boardroom is built on an open-source core and can be deployed flexibly (cloud or on-prem) via a serverless model[71][52]. This is crucial because board discussions often involve sensitive data. Companies might be wary of putting that into a black-box SaaS. By having an open architecture, The Cyber Boardroom can be inspected for security and even hosted in a controlled environment. The open model also invites contributions from the community -- for example, new personas could be added (imagine a template persona for "audit committee chair" or "non-executive director with finance background"), or the knowledge base could be expanded with more scenarios. As noted, open-source establishes credibility with enterprise customers who can verify the integrity of the code[57]. For a tool meant to be in the boardroom, credibility of the tool itself is important -- it must be beyond reproach in terms of confidentiality and accuracy. Adopting open, transparent development helps in that regard, aligning with Cruz's overarching strategy of leveraging open-source as a trust and innovation catalyst[57][72].

In sum, The Cyber Boardroom showcases how time and iteration, combined with AI, can **calibrate trust in a human relationship context**. It's not marking a statement true or false over time, but rather refining a message over time so that it lands truthfully and effectively with an audience. It demonstrates that the same principles of evidence, context, and adaptation apply whether we are verifying a fact or persuading a person: provide the right context, check understanding, adjust, and do this repeatedly to build confidence. It complements MyFeeds by focusing on how insights are communicated and acted upon at the strategic level, closing the loop: data turns into intelligence (MyFeeds), which turns into decisions and actions (via Boardroom) -- all under a philosophy of **transparency, provenance, and continuous learning**.

Open Infrastructure for Transparent, Scalable Analysis

Underpinning the philosophy and examples above is a commitment to **open-source infrastructure and scalable architecture**. Dinis Cruz's vision is not only about *what* should be done (track credibility over time) but also *how* it should be enabled. The **credibility** of an information system is tied not just to the data it presents but to the trust users have in the system itself. By using open-source, transparent methods and modern cloud-native design (like serverless computing), these projects ensure that the system's operations are **auditable, adaptable, and capable of growing** with the data.

Open-Source as a Foundation: All four of Cruz's synergistic startups (including MyFeeds.ai and The Cyber Boardroom) share an open-source core[73]. This is a deliberate strategy. Open-source software allows anyone to inspect the code for biases, errors, or security issues. For AI systems dealing with knowledge and truth, this is particularly important. Users -- especially in cybersecurity and research -- are rightly skeptical of black boxes. An open approach means the logic behind claim classification, evidence scoring, or feed curation can be scrutinized and validated. It establishes a baseline of **trust through transparency**[57]. Enterprises can vet the tools before adopting them, and independent contributors can suggest improvements or catch problems. Moreover, open-source encourages a community of practice: for example, researchers might contribute new modules for fact-checking or journalists might extend the taxonomy for new types of media. This collective innovation accelerates the development of robust credibility systems[74]. Cruz's experience as an open-source advocate (e.g., his creation of the OWASP O2 Platform for security testing) feeds into this approach -- he has seen how community-driven projects can shape industry standards[75][76]. In these new ventures, being open means not reinventing the wheel for each project. Indeed, the startups reuse key building blocks -- a library for semantic graph handling created in one is reused in others[77][78] -- which speeds up development and keeps the design consistent.

Serverless and Lean Architecture: Scalability is crucial because evaluating credibility over time can become data-intensive. There may be thousands of sources, millions of statements, and constant updates. The use of a **serverless deployment pipeline** means the system can scale out when needed (e.g., processing a burst of news during a major incident) and scale to zero when idle[79][52]. This ensures cost-effectiveness -- a key consideration for startups and also for any organization deploying such a system. They only pay for the compute they actually use, making it feasible to monitor vast amounts of information without a massive always-on infrastructure. A unified CI/CD and packaging approach across these projects allows them to run in various environments easily -- whether as cloud functions, containers, or on-prem appliances[79]. This flexibility means the tools can be brought to the data (important if certain data can't leave a corporate environment for privacy reasons). **Minimal fixed costs and high elasticity** also mean that as the user base grows or as more data streams are added, the system can accommodate that without a ground-up redesign[52]. It's essentially future-proofing the platform to handle the "firehose" of information we expect in the coming years.

Shared Components and Interoperability: The startups were conceived to complement each other, sharing technology and passing data between them where useful[18][80]. This interoperability is a strength of an open, modular design. For example, the knowledge graphs are stored in standardized formats across the systems[81][18]. This means an insight discovered in MyFeeds (like a trending new threat topic) could be fed into The Cyber Boardroom's knowledge base to alert CISOs and boards about it[18]. Or the Report Assistant's structured output of a risk assessment could be used to generate an executive summary for the board, or to feed into MyFeeds as an internal news item. By **weaving these tools together**, Cruz envisions an ecosystem where data flows securely to where it's needed, and every piece of analysis reinforces others[80][82]. This reduces duplication of effort (each tool doesn't need to rediscover the same facts) and enhances consistency (a fact confirmed in one context is automatically updated everywhere). From an infrastructure perspective, this is facilitated by using common storage abstractions (MemoryFS and GraphFS to represent data uniformly as files or graphs)[17], and by keeping everything open so integration is straightforward (no proprietary formats or locked APIs).

Transparency in AI Workings: Another reason open-source is vital is the need to **audit AI decisions**. When an AI model suggests that "Claim X is likely true" or filters out a piece of content as misinformation, stakeholders will want to know why. By having an open system, one can examine the rules or model outputs that led to that decision. For example, if a claim was flagged as false, was it because the AI found a contradicting source? Did it perhaps misinterpret something? Transparency allows developers and even end-users to ask these questions. In a closed system, one might suspect biases or errors but have no way to confirm; in an open system, one can look under the hood. The LETS pipeline structure supports this by design, since it logs each step's output and keeps the transformations modular[20]. It would not be hard, for instance, to output an intermediate file that shows "extracted claims and their initial confidence scores before and after reconciliation" for a given news article. Such traceability builds confidence that the system isn't arbitrarily labeling things as true or false -- it's following documented procedures that can be verified or contested as needed.

Security and Trust: In cybersecurity applications, trust in the tool is paramount. By open-sourcing and building on well-tested components, Cruz's startups aim to be **secure by design** and earn the trust of security professionals (a notoriously tough crowd). The idea is that by the time these tools are being used in a critical environment (like a board meeting or processing a company's internal knowledge), they've been vetted by many eyes and perhaps formally verified or certified. Community vetting can catch vulnerabilities or logic flaws early. Additionally, from a user trust standpoint, being open-source aligns with the values of many in the target audience: journalists favor transparency, researchers value open data and methods, and security experts trust open scrutiny over closed promises[57]. We see this in the widespread adoption of open-source tools in security (like Wireshark, Metasploit, etc.) largely because people can ensure there are no malicious backdoors. Similarly, an open-source credibility engine can be trusted not to have hidden biases introduced for commercial or political reasons -- any such attempt could be discovered in the code or training data.

Community and Ecosystem: Finally, open infrastructure fosters an ecosystem. Others can build atop these tools -- for example, someone might create a specialized plug-in for MyFeeds to handle a new domain (say medical news or financial markets) using the same pipeline. Or they might extend The Cyber Boardroom to other types of boardroom topics (risk in general, not just cyber). This means Dinis's core idea -- time-calibrated credibility -- can spread and adapt beyond his immediate implementations. It encourages **industry-wide adoption** of standards for tracking provenance and evidence. If multiple tools output knowledge graphs with similar schema for claims and evidence, these could interoperate or be aggregated. Imagine an open standard for representing "credibility of a claim" with timestamps, source references, and status -- much like RSS became a standard for syndicating feeds, a standard for credibility data could enable a whole new class of applications. By basing everything on open principles from the start, these projects are well-positioned to contribute to and benefit from such developments.

In summary, the **infrastructure choices** reflect the same values as the system's logic: transparency, trust, and adaptability. Just as we want each statement's trustworthiness to be traceable and updatable, we want the system itself to be transparent and improvable. By leveraging open-source and serverless architecture, Dinis Cruz's projects not only address the technical challenges of building credibility systems but also the social and ethical ones -- ensuring the systems themselves merit the trust we place in them.

Conclusion

In a world awash with information and misinformation, **time** may be the most underutilized tool we have for discerning the truth. Dinis Cruz's vision, as articulated in this paper, is to explicitly harness the temporal dimension as a calibrator of credibility in our information systems. By treating each statement as an entity with its own life story -- from inception through evolution under the scrutiny of evidence -- we can transform how trust is built in the digital age. This approach moves us beyond static true/false judgments into a dynamic model where assertions are born as hypotheses or opinions, mature (or wither) as facts through corroboration or refutation, and are continually recontextualized as new data emerges.

We introduced a taxonomy of information types (facts, opinions, hypotheses, data) as the foundation for this framework, recognizing that different kinds of statements demand different handling and validation. We saw how large language models and AI can serve as powerful allies in extracting these elements and populating **semantic knowledge graphs** that serve as living maps of knowledge. With systems like MyFeeds.ai and The Cyber Boardroom, we explored how these concepts are not merely theoretical -- they are being implemented in real products that address pressing needs: from personalized intelligence feeds that **keep professionals informed with context and provenance**[15], to boardroom assistants that **bridge the gap between technical truth and business trust**[32][65]. These examples underscore the practicality and versatility of the vision. They show, for instance, that an AI-curated news brief can gain a user's trust by explaining its recommendations and highlighting source credibility[24], or that an executive can trust a cybersecurity briefing because it's been honed through persona-driven simulations to preempt their concerns[36].

A recurring theme is that **transparency and traceability** are inseparable from credibility. A system that tracks the lineage of every claim, that can point and say "this is what we know and here's how we know it," inherently engenders more trust than one that cannot. By using time and evidence as the yardstick, we also inject a healthy dose of humility and resilience into our AI: humility in acknowledging uncertainty and separating what is known from what is conjectured, and resilience in being able to update and correct course as reality unfolds. In essence, the systems we build must themselves learn and adapt over time, much like the humans who operate them.

We also emphasized the **importance of open, scalable infrastructure** in realizing this vision. The choice to build on open-source principles and serverless architectures is not just an implementation detail; it is a statement of values. It says that **truth-seeking should be a collaborative, transparent endeavor**, and that the tools for it should be accessible and trustworthy. By aligning engineering choices with the end goal of trust, Cruz's approach ensures the platform on which we measure credibility is itself credible. This alignment of content and platform -- having open data pipelines (LETS), common schemas, and community vetting -- means that the credibility system can be trusted not to distort or conceal. It also means it can scale to meet the challenge: as the volume of information explodes, the combination of cloud scalability and crowd-sourced improvement positions these systems to keep up with the deluge, extracting signal from noise.

For AI researchers, this whitepaper offers a blueprint of how AI can move beyond isolated predictions and into the realm of **knowledge maintenance over time** -- a sort of longitudinal AI that remembers and revises. For cybersecurity professionals, it outlines tools that can enhance situational awareness and strategic communication, ensuring that security insights are both accurate and effective in driving decisions. For journalists and truth-seekers, it presents hope that technology can be harnessed to bolster fact-checking, provide context, and uphold the integrity of information in the public sphere.

Looking ahead, one can imagine the principles outlined here being applied widely: social media platforms that tag posts with the current credibility status of their claims (and update them as facts emerge), scientific literature databases that track the replication and validation status of published results over time, or public knowledge bases (like Wikipedia or Wikidata) enhanced with temporal evidence graphs that show how our understanding of a topic has evolved. The concept of a "credibility timeline" could become a standard feature in information consumption, much like timestamps or view counts are today.

In conclusion, by making **time the calibrator of credibility**, we align our information systems more closely with the reality of how knowledge works in the real world. Truth is a process -- one of inquiry, verification, and sometimes revision. Dinis Cruz's vision is to embed that process into the fabric of our digital knowledge tools. It is a vision of *information integrity through temporal context*, one that holds great promise for improving trust in the age of AI. As these ideas are implemented and refined in projects like MyFeeds.ai and The Cyber Boardroom, they lay the groundwork for a new paradigm in which *credibility is not just asserted, but demonstrated and earned over time*.

By combining philosophical rigor with technical innovation -- from taxonomy and knowledge graphs to persona models and open infrastructure -- we can build systems that not only handle information, but genuinely *understand* and *honor* the journey each piece of information takes. In doing so, we equip ourselves and our societies with better defenses against falsehood and better tools to navigate an ever more complex information landscape. The ultimate calibrator, time, will tell how successful this approach will be, but the framework laid out here provides a clear and compelling path forward.

Sources: The concepts and examples discussed in this paper are drawn from Dinis Cruz's work and voice memos, as well as related documentation and demonstrations of the mentioned platforms. Key references include the MyFeeds.ai architecture for semantic feed curation the Cyber Boardroom's persona-based communication approach, the Interactive Report Assistant's method of capturing facts and hypotheses with provenance, and Dinis Cruz's overall advocacy for building trust in AI through **provenance, transparency, and human-centered design**. These sources illustrate the marriage of philosophy and practice, showing the real-world momentum behind time-calibrated credibility systems.