SCIRES it
SCIentific RESearch and Information Technology
Ricerca Scientifica e Tecnologie dell'Informazione
Vol 16, Issue 1 (2026), 147-164
e-ISSN 2239-4303, DOI 10.2423/i22394303v16n1p147

CASPUR-CIBER Publishing, http://www.sciresit.it

FROM REGEX TO TYPED AST: THE ANSELMUS COMBINATOR-BASED PARSER FOR THE IBISCRIPT DSL APPLIED TO MIDDLE PERSIAN AND PARTHIAN BILINGUAL INSCRIPTIONS

*Andrea Marruzzo**

*Sapienza University of Rome – Rome, Italy.

Abstract

This paper introduces ANSELMUS, a combinator-based parser for IbiScript, a new Domain-Specific Language (DSL) targeting epigraphic texts according to the Leiden Conventions. Implemented in Rust via the nom library, ANSELMUS produces a typed Abstract Syntax Tree (AST) that encodes the epigraphic grammar at the type level, enabling static validation and multi-format serialization—a departure from the regex-to-XML model of existing tools. The library compiles to WebAssembly for client-side deployment without server infrastructure, exposes a Command Line Interface (CLI) and Python bindings for integration with existing Digital Humanities (DH) pipelines, and is fully conformant with TEI P5 and EpiDoc guidelines. The system's validity is demonstrated through its application to a corpus of Middle Persian and Parthian bilingual inscriptions (3rd century CE), characterized by complex fragmentary states and diverse epigraphic notations.

Keywords

Digital Epigraphy, Leiden Conventions, Parser Combinators, Abstract Syntax Tree, TEI P5, Rust, WebAssembly

1. *The Leiden Tradition and the Foundations of Digital Epigraphy*

1.1 Leiden Conventions

The formalization of the Leiden Conventions following the 1931 meeting at the University of Leiden represented the first international effort to standardize the critical representation of ancient texts (Wilcken, 1932). However, as Schubart (1918) had observed in the preceding decades, *"die Editionsweise ist nicht überall gleich"* [the method of editing is not the same everywhere].

This lack of uniformity persisted well beyond 1932. Regional editorial schools and distinct linguistic traditions—particularly between Greek and Latin epigraphy—continued to diverge, producing a fragmented landscape of micro-traditions in which the same editorial problem might be handled differently depending on the corpus, the country, or even the individual editor. Even the *Supplementum Epigraphicum Graecum Online* (Chaniotis et al., 2026) acknowledges that editors differ in their deployment of brackets and sigla, confirming that the "Leiden System" has remained a living, if inconsistent, set of practices rather than a fixed universal standard.

Sterling Dow's (1969) systematic attempt to rationalize these conventions served as a critical precursor to the digital age. By attempting to reconcile papyrological and epigraphic usage into a unified logical set, Dow anticipated the structural requirements of later computational models. A parallel effort in the Italian epigraphic tradition is represented by Panciera (1982), whose contribution to the *Colloquio Internazionale AIEGL* proposed a systematic reformulation of the sigla for Latin inscriptions—an independent convergence toward the same formal clarity that the digital age would eventually demand.

Yet, the pre-TEI (Text Encoding Initiative) era was characterized by proprietary silos. Early projects such as the Packard Humanities Institute (PHI) and the *Thesaurus Linguae Graecae* (TLG) pioneered custom ASCII-based markup to facilitate machine processing, but highlighted a fundamental tension: the Leiden Conventions were designed as a visual shorthand for human readers, not as a formal language for computational parsing. As observed by those leading the redevelopment of the Duke Databank

of Documentary Papyri (cf. Sosin, 2012), the visual ambiguity of human-readable sigla—where a single bracket might represent multiple editorial states depending on context—presented a significant obstacle to data interoperability. As Panciera (2012) reflected, the absence of a shared formal standard meant that digital epigraphy initially mirrored the very diversity of editorial methods it sought to catalog, eventually necessitating the rigorous, XML-based semantic markup of the TEI and EpiDoc era.

1.2 The Rise of EpiDoc and Semantic Markup

The late 1980s witnessed the inception of the TEI as a strategic response to the fragmentation of project-specific encoding systems. The TEI sought to establish a universal, machine-readable framework for text representation, prioritizing long-term data sustainability and cross-institutional interoperability (Ide & Sperberg-McQueen, 1995; Burnard, 2014).

Nevertheless, the inherent generality of the TEI guidelines proved insufficient for the granular requirements of epigraphy and papyrology. The precise documentation of physical damage, lacunae, and multi-layered editorial interventions demanded a specialized formalism that standard prose-oriented tags could not provide.

Consequently, EpiDoc emerged as a specialized TEI customization designed to map the Leiden Conventions onto an XML-based hierarchical structure. By providing a common schema for major corpora—including the Duke Databank of Documentary Papyri (DDbDP), the Digital Corpus of Literary Papyri (DCLP), and the Epigraphische Datenbank Heidelberg (EDH)—EpiDoc facilitated the first sustained computational formalization of epigraphic logic (Cayless et al., 2009; Bodard, 2010).

Despite this achievement, the transition to XML introduced a significant "verbosity tax". The semantic richness of EpiDoc is frequently obscured by the syntactic overhead of nested XML elements, creating a disconnection between the scholar's shorthand mental model and the machine's requirement for well-formed markup. As Pichler (2021) has argued, the hierarchical nature of XML encoding is not a neutral representation of text but an interpretive act—one that necessarily privileges certain structural readings over others, embedding editorial choices at the infrastructural level.

1.3 The Monolithic Response and the "Friction of Verbosity"

The scholarly response to the inherent complexity of EpiDoc has largely resulted in "all-in-one" frameworks such as EFES (EpiDoc Front-End Services), TEI Publisher, Kiln (King's College London, 2024), and EVT (Edition Visualization Technology). While these platforms successfully lowered the entry barrier for digital publication, they reinforced a monolithic architecture that bundles philological logic within specific web-stack infrastructures. This systemic friction is best summarized by the "Tools Paradox" (Pape, Schöch, & Wegner, 2012): despite the maturity of TEI standards, the actual adoption of digital publishing frameworks remains low due to the daunting technical infrastructure—such as eXist-db—required to sustain them.

This challenge is compounded by a pedagogical hurdle. As Faghihi, Holford, and Jones (2022) observe, the TEI is frequently perceived as "intimidating" due to its complex hierarchical structure and the "profusion of angle brackets". This creates a situation where the "short but steep learning curve" prevents many practitioners from engaging with the standard outside of highly specialized, one-to-one teaching environments. Projects like TEICHI attempted to resolve this by embedding philological logic into Content Management Systems like Drupal, but as Faghihi et al. argue, teaching the TEI effectively still requires bridging the gap between legacy data and modern machine-readable standards.

The result, as Cummings (2018) and Materni (2020) have independently documented, is a twofold tension: a standard designed for data interoperability became functionally "trapped" within rigid, server-side silos, while its syntactic complexity—what we term here the "Friction of Verbosity"—alienated the very scholars it was meant to empower. Projects have attempted to bridge this gap via form-based interfaces, but such solutions often conceal the text to the point of losing the fluidity of traditional transcription. This combined burden of infrastructure obsolescence, pedagogical intimidation, and editorial friction necessitates a move toward a decoupled library approach.

2. Technical Evolution: From Regex to AST

For analytical purposes, the transition from early digital corpora to modern philological

pipelines can be characterized as a shift between two developmental phases. The first, which we term the "Heroic Age", is defined by experimental, project-specific systems that prioritized immediate philological output over architectural sustainability. The second, the "Industrial Age", is characterized by modular, high-performance engines designed for long-term interoperability and platform independence. This periodization is not merely a matter of improved hardware, but reflects a fundamental transformation in how scholarly notation is interpreted by the machine.

While the Heroic Age focused on the semantic storage of data (XML/TEI), the Industrial Age shifts the challenge to the computational parsing of the notation itself. By moving beyond the "visual shorthand" of the Leiden tradition toward formal DSLs and ASTs, the field can achieve the level of precision and platform-independence required for large-scale, cross-linguistic interoperability.

2.1 Heroic Age: Regex and Interpreted Languages

Because manual XML encoding is a complex and error-prone task, various systems have been developed to assist scholars in producing critical editions within a user-friendly environment. These tools allow researchers to work directly on the text using DSLs, bypassing the need for a specialized XML editor.

Leiden+ was created between 2008 and 2010 as part of the SoSOL/Papyri.info redevelopment project (Papyri/Sosol, 2010-2025) and remains the official input syntax for both DdbDP and DCLP contributions. This notation is then processed by a monolithic pipeline (Ruby on Rails + XSLT + XQuery + eXist-db), rendering it inseparable from its original infrastructure (Papyri.info Collaborative, 2013).

Crucially, the translation from the shorthand notation to the underlying XML is largely performed via complex regular expression (regex) patterns. This reliance on string-matching, while sufficient for the "Heroic Age" of digital papyrology, introduces a "flat" processing model that struggles with the nested, recursive, and often context-sensitive nature of the sigla. To contribute even a single bracketed correction, the user must not only adopt a specific Ruby/eXist-db stack but also trust a parser that lacks a formal grammar or a verifiable AST.

The Proteus project (Williams et al., 2015) advanced this lineage by introducing CSYN (Critical Syntax), a shorthand notation inspired by both the Leiden Conventions and the simplicity of Markdown. By developing a custom parser in an interpreted language like Python, Proteus demonstrated that scholars could bypass the rigidity of XSLT to receive real-time validation of their transcriptions. While revolutionary, these tools eventually encountered a maintenance paradox: by coupling philological logic to specific web-framework versions and relying on flat regex patterns for CSYN-to-XML conversion, these systems lacked the structural recursion necessary to validate complex, overlapping editorial interventions.

This reliance on string-matching introduces a fundamental theoretical failure. Regular expressions are mathematically limited to the class of regular languages, a constraint rooted in their foundation as Finite-State Automata (FSA). As formal language theory dictates, "A is regular [if and only if] there is a DFA M such that L(M)=A" (Sipser, 2013). Consequently, any notation requiring nested, hierarchical, or recursive structures—such as the overlapping brackets of the Leiden Conventions—necessarily exceeds the expressive power of regex. When an editorial intervention involves an expansion within a restoration—such as [...(abc)...]—a "flat" regex-based processing model lacks the "memory" (the stack) required to track stateful recursion. This makes such parsers inherently brittle, prone to catastrophic backtracking or silent misinterpretation of nested sigla.

These theoretical limitations are exacerbated by the practical failures of regex regarding Unicode handling. Regular expressions vary significantly across implementations and lack stateful grapheme boundary detection, multi-code-point lookahead, and script-specific tailoring—properties essential for the precise Unicode manipulation required by non-standardized historical scripts (Davis, 2025). In high-stakes philological contexts such as Book Pahlavi—a writing system not yet fully supported in Unicode due to its extreme ligatures—this architectural vulnerability leaves the digital scholarly record susceptible to silent data corruption and infrastructure obsolescence.

2.2 Industrial Age: AST, Compiled Languages, and WebAssembly

The transition to the "Industrial Age" is defined by a shift from heuristic string-matching to Formal Language Theory, where scholarly notation is

treated as a structured language governed by a formal grammar. This shift is not merely technical but pedagogical; as Mugelli, Re, and Taddei (2020) argue, digital tools must move toward a "user-centered" approach to overcome the "incompatibility" often perceived between new technologies and ancient languages.

The "Industrial" path was paved by several artisanal precursors that attempted to move beyond flat regex toward DSLs. Early efforts within the EpiDoc community, such as the use of RelaxNG and Schematron, successfully formalized the XML "semantic container" but failed to provide an engine for the "editorial input" (Leiden) itself (Cayless et al., 2009; Bodard, 2010; Burnard & Baumann, 2022). Subsequent grammar-based strategies remained largely experimental. Prototypes ranging from a PyParsing-based Leiden+ translator (Baumann et al., 2011) to a full BNF grammar defined in ANTLR at Oxford (ca. 2014–2016) demonstrated that treating the Leiden Conventions as a recursive, generative language was technically feasible (Aho et al., 2006). The Epigraphische Datenbank Heidelberg conducted similar experiments (ca. 2018–2020). None of these projects matured into a reusable library: all remained unpublished or were abandoned during institutional transitions.

A significant milestone in this transition is the Euporia system (Mugelli et al., 2020), developed in collaboration with CoPhiLab (CNR-Pisa). By allowing scholars to define their own marking conventions—frequently utilizing "natural" markers such as hashtags—Euporia demonstrated that DSLs could lower the "Friction of Verbosity" without sacrificing philological rigor.

The transition to a formalized industrial paradigm has been recently codified as DSL-based Digital Scholarly Editing (DSL-based DSE). As proposed by Del Grosso et al. (2024), this methodology seeks to bridge the gap between traditional philological notation and digital outputs by leveraging DSLs as the primary medium of scholarly work. This approach is currently being implemented in the GreekSchools project (ERC-885222) via the CoPhiEditor, a collaborative platform that uses ANTLR-generated grammars to manage the dense complexity of Herculaneum papyri. Their work marks a definitive shift away from the legacy "regex-to-XML" model toward a workflow where the DSL is not just a shorthand, but a mathematically verifiable foundation for the entire life-cycle of the digital edition.

However, even within this sophisticated framework, two "industrial" bottlenecks persist. First, the technical stack remains primarily coupled to the Java Virtual Machine (JVM), which—while robust—introduces significant overhead and dependency management issues for lightweight or browser-native environments. Second, there remains a tendency to treat the generated AST as a transient intermediate state, which is ultimately serialized back into XML/TEI "middle grounds" for storage and interchange. As Schmidt (2014) has argued, the pursuit of interoperability through XML serialization frequently sacrifices the structural precision of the source representation to maintain compatibility with legacy infrastructure.

These architectural choices carry a long-term cost: monolithic, infrastructure-dependent systems are inherently susceptible to the "software rot" that progressively undermines DH projects as their underlying technology stacks decay (Fenlon, 2020; Barats, Schafer, & Fickers, 2020; Martinelli, 2025).

A vital touchstone for this landscape within the Middle Iranian domain is the Zoroastrian Middle Persian: Digital Corpus and Dictionary project (MPCD), a long-term initiative operating across Bochum, Berlin, and Cologne (Rezania, Cantera, Eide, & Neuefeind, 2021–2030). While MPCD successfully utilizes comprehensive, multi-layered TEI-XML schemas to manage expansive annotations and syntax mapping across the Pahlavi literary tradition, its systemic scale highlights a profound opportunity for cross-field collaboration.

Bridging the gap between expansive, macro-level lexicographical text databases and lightweight, notation-driven parsing architectures offers a direct avenue for future integration—specifically in adapting AST-driven engines to serve as real-time validation and ingestion tools for diverse Middle Iranian sub-corpora.

ANSELMUS addresses these systemic and domain-specific bottlenecks by proposing an AST-first architecture where the parsed representation—not the XML file—becomes the primary source of truth.

The following section details this approach.

3. *ANSELMUS: Bridging Epigraphic Convention and Formal Grammar*

ANSELMUS is a Rust library implementing a combinator-based parser for a new epigraphic DSL named IbiScript, which constitutes the notation language in which the scholar writes. Unlike previous approaches based on regular expressions or XSLT transformations, ANSELMUS produces a typed AST that encodes the epigraphic grammar at the type level, enabling static validation and multi-format serialization from a single parsed representation. Because the parsing logic is strictly idempotent, it guarantees deterministic, reproducible outputs across automated workflows, completely eliminating the state-tracking errors common to legacy text-replacement pipelines. The notation and its parser synergize allowing scholars to focus on the critical study and edition of the text while BEDA—the overarching digital platform designed for both technologies—concentrates on metadata and facsimile.

3.1 Combinator Parsing as a First-Class Epigraphic Tool

The choice of parsing strategy is not merely a technical detail but a direct consequence of the epigraphic domain's requirements. Two dominant paradigms exist for implementing DSLs: generative grammar tools such as ANTLR, and combinator-based parsing libraries. While both approaches produce parsers capable of recognizing structured input, their architectural implications differ substantially.

Generative grammar tools operate by compiling a formal grammar specification into a parser at build time. The resulting parser is a black box: correct in its behavior but opaque in its internals. For a DSL as semantically dense as epigraphic notation, this opacity is a liability. The generated code is neither readable nor modifiable by domain experts, error messages are generic and divorced from philological meaning, and the AST produced is a generic parse tree that must be post-processed and cleaned before it can carry domain-specific semantics. Furthermore, ANTLR's runtime dependency on the JVM reintroduces the same infrastructure overhead discussed above, making it ill-suited for lightweight, browser-native, or embedded deployments.

Combinator parsing inverts this relationship. In a combinator-based library such as nom, parsers are ordinary functions that compose through higher-order combinators. Each philological phenomenon—unclear signs, supplied text, lacunae of known extent—is implemented as a discrete, named, independently testable parser function. The composition of these functions directly mirrors the formal grammar of the notation, making the parser itself a readable, auditable specification of the language. There is no build step, no generated code, and no runtime dependency beyond the library itself.

This architecture has three direct consequences for ANSELMUS. First, the AST is not a generic parse tree but a domain-typed Rust enum hierarchy—*IbiUnit*—where the type system statically enforces the constraints of the Leiden Conventions. Invalid combinations, such as an omitted sign inside an erasure, are rejected at the parser level rather than deferred to a post-processing validation step. Second, the parser compiles to WebAssembly without modification, enabling real-time client-side parsing in browser environments without server infrastructure. Third, a CLI and Python bindings via PyO3 allow integration with existing DH pipelines without requiring Rust expertise from the end user.

The result is a parser that is simultaneously a formal specification, a validation layer, and a portable execution kernel—properties that are structurally inaccessible to generative grammar approaches operating within the JVM ecosystem.

3.2 IbiScript: the Epigraphic Notation Language

Within IbiScript, a deliberate separation exists between input notation and rendered output. In the Leiden Conventions and in systems such as Leiden+, unclear signs are rendered typographically as dotted letters—ḥ, ṣ, ṭ—a visual convention originating in print publication. Accepting such characters as input syntax would introduce a fundamental ambiguity: in the transliteration of Middle Iranian languages dotted letters are independent phonological characters, not editorial markers. The string "ḥ" is therefore genuinely ambiguous if accepted as input: it could represent an unclear "h" or a plain Iranian phonological character, and no parser can resolve this without external context. IbiScript resolves this by enforcing a strict boundary: unclear characters are marked exclusively with asterisk delimiters "*h*", producing *Unclear*([*Plain*("h")], *cert*: Medium), while "ḥ" is always parsed as

Plain("ḥ"), preserving its transliteration value unmodified.

This design principle extends to all notation: IbiScript prioritizes ergonomics alongside unambiguity. For phenomena with established visual conventions, multiple equivalent input forms are accepted—gaps of known extent, for instance, can be expressed as "[------]", "[.5.]", or "[2-2]"—allowing scholars to write in whichever form feels most natural without sacrificing parse correctness. Similarly, the parser applies syntactic laxism to whitespace: "[MLKAn]" produces *Supplied*([*Plain*("MLKAn")], *cert*: Medium) regardless of the number of spaces or newlines between the square bracket delimiters. This tolerance for minor formatting variation eliminates a common source of friction in text-based workflows, where invisible whitespace differences would otherwise cause silent parse failures. The result is a notation that is non-ambiguous in its semantics but permissive in its surface form—designed for quick yet rigorous scholarly transcription.

Two syntactic conventions govern the notation as a whole. The delimiter "$" acts as a universal metadata marker, disambiguated by the token immediately following it: "->" or "<-" encode a text direction change, "M:content" anchors a critical note to the M marker, "vacatN" marks a willingly unwritten space spanning N lines, and so on.

The escape character "\" suppresses the semantic meaning of the following character, rendering it as literal text: "[abc\?]" produces *Supplied*(*content*: [*Plain*("abc?")], *cert*: Medium) rather than *Supplied*(*content*: [*Plain*("abc")], *cert*: Low).

Tab. 1: Core epigraphic phenomena in IbiScript. Mapping of notation to AST representations and Unicode output, exemplified by the Pahlavi-inscribed ashlars of the Paikuli inscription (Cereti & Terribili, 2014).

Epigraphic Source	IbiScript	AST	Unicode Output
PK21 – D2, 1	[LN]*E*	*Supplied*([*Plain*("LN")], *cert*: Medium), *Unclear*([*Plain*("E")], *cert*: Medium)	[LN]𐭩𐭥𐭩
PK13 – C2, 4	!]AMT	*Fracture*(Left), *Plain*("AMT")	⊣AMT
PK32 – E8, 6	[]	*Gap*(Unknown, *precision*: Medium)	[...]
PK21 – D2, 6	[.5.]	*Gap*(Known(5), *precision*: Medium)	[−−−−−]
PK87 – C9, 3	$2:*W?*$	*ApparatusMark*(*marker*: "2", *content*: [*Unclear*([*Plain*("W")], *cert*: Low)])	⋘𐭩𐭦𐭩⋙²
PK36 – F18, 3	[%h|\u{right-half-ring}%mw]	*Supplied*([*PlausibilityAssessment*(*PlausibilitySeq*(*lectiones*: [*Plain*("h"), *Plain*("’")]), *links*: [*PlausibilityLink*(*kind*: Alternative, *degree*: None)]), *Plain*("mk")], *cert*: Medium)	[{h ‖ ’}mw]

The notation is designed to be writable in any plain text editor without special input methods or Unicode entry tools. All delimiters are drawn exclusively from characters accessible on standard European keyboards, ensuring that the full expressive power of the system remains available to scholars working in ordinary transcription environments—without dependency on specialized software or input configurations.

The last example of Tab. 1 shows how the scholar can easily insert the right half ring character "’", fundamental for the transliteration of Iranian languages, by using an intuitive notation which revolves around a valid ISO 15924 writing system code (not needed for Latin and common characters like the right half ring) and ergonomic glyph name aliases derived from the official Unicode name of the glyph (here "MODIFIER LETTER RIGHT HALF RING"), minus Unicode working prefixes.

Inserting Inscriptional Pahlavi characters is equally effortless: if the scholar were to specify the diplomatic form of the word "AMT", heterogram previously seen in Tab. 1, he could have written as a *LexicalUnit* the following

```
&AMT;
  dipl = \u{phli:aleph, m, taw}
&
```

populating the word metadata with the Inscriptional Pahlavi text 𐭠𐭬𐭮, difficult to both type and render in a document.

IbiScript's glyph insertion notation supports all historical scripts and the majority of symbols required by related linguistic disciplines; any

remaining characters can be directly inserted using their respective Unicode codepoints.

The final example in Table 1 highlights the expressiveness of IbiScript, demonstrating its capacity to deterministically encode a scholar's interpretive doubt within a critical edition. To understand the structural mechanics of this "plausibility chain", an architectural metaphor is instructive.

Each relationship between two adjacent *lectiones* represents one element of a path that always proceeds from left to right, in a descending direction: the leftmost *lectio* is always the most plausible. The ">" operator introduces a descending step, whose steepness is determined by the degree modifier: ">?" indicates a gentle step, ">" a step of medium steepness, and ">!" an abrupt step. The "|" operator, by contrast, introduces a landing: the two *lectiones* stand at the same level, with no preference expressed by the editor.

Consequently, IbiScript's plausibility chain forms an architecture of steps and landings that, taken as a whole, always tends downwards. Even when the path alternates between horizontal stretches and descents of varying steepness, it constitutes a highly effective notation—offering both intuitive simplicity for the scholar and a deterministic, machine-readable output for the parser.

Although ANSELMUS and IbiScript were developed primarily for Eastern epigraphy, their architecture is designed to meet the rigorous demands of classical Greco-Latin studies. The system's AST renderers are highly configurable, allowing for output that reflects the specific typographical conventions of different epigraphic branches (e.g., *RenderConfig*::*leiden_classical*() produces standard Leiden-compliant output for classical studies).

Tab. 2: Comparative analysis of IbiScript Unicode output and legacy rendering system limitations.

Tradition	Notation	Output	AST	Notes
Iranian studies	*abç?*	(abç?)	*Unclear*([*Plain*("abç")], *cert*: Low)	Overloads expansion syntax; "?" is indistinguishable from literal text.
Greco-Latin Leiden	*abç?*	ạ̣ç (?)	*Unclear*([*Plain*("abç")], *cert*: Low)	Underdots obscured by diacritics; visually identical to row 5.
IbiScript universal	*abç?*	⦗abç⍰⦘	*Unclear*([*Plain*("abç")], *cert*: Low)	Preserves diacritics while providing unambiguous markers.
Iranian studies	*ab**ç?*	(ab)(ç?)	*Unclear*([*Plain*("ab")], *cert*: Medium), *Unclear*([*Plain*("ç")], *cert*: Low)	Fragmented syntax causes visual stutter and persistent ambiguity.
Greco-Latin Leiden	*ab**ç?*	ạ̣ç (?)	*Unclear*(*Plain*(["ab"]), *cert*: Medium), *Unclear*([*Plain*("ç")], *cert*: Low)	Cumulative dots increase visual noise and obscure uncertainty levels.
IbiScript universal	*ab**ç?*	⦗ab⦘⦗ç⍰⦘	*Unclear*([*Plain*("ab")], *cert*: Medium), *Unclear*([*Plain*("ç")], *cert*: Low)	Distinctly maps the boundaries of separate editorial phenomena.

The most significant innovation proposed by IbiScript concerns the representation of uncertain signs, represented in the AST as *Unclear*. The classical Leiden tradition prescribes underdots for uncertain signs; however, as noted previously, this convention is impractical for Iranian scripts, as for Semitic and Indic scripts. In these contexts, the dot is a critical phonetic indicator (diacritic) rather than an editorial marker of uncertainty. Consequently, scholars like Skjærvø & Humbach (1978-1983) or Cereti & Terribili (2014) often resort to parenthetical notation—e.g., "(abc)"—despite the fact that this conflicts with the Leiden use of parentheses for expanding abbreviations.

To resolve this ambiguity, IbiScript proposes a universal solution: the use of the mathematical brackets "⦗⦘". These characters are visually similar to standard parentheses—maintaining a familiar "vibe" of uncertainty—yet they remain harmoniously distinct from other established

mathematical brackets, such as the "⟦⟧" used for erasures (*Erasure* in our AST). Moreover, IbiScript utilizes the special character "⸮" rather than the common question mark to express low certainty. This further disambiguates the output, preventing confusion in cases where a question mark might actually be part of the original disputed text.

Tab. 2 illustrates the practical advantages of this innovation, demonstrating how it resolves edge cases that typically fail in traditional rendering systems. For the sake of usability, ANSELMUS applies these innovations by default, ensuring that the scholar is not required to manually insert specialized characters while still benefiting from a high-precision digital edition.

3.3 ANSELMUS: The Combinator Parser

The ANSELMUS parser is implemented as a hierarchy of combinator functions built on the nom library (Couprie, 2026). The AST is rooted in the *IbiUnit* enum, which defines all the nodes that ANSELMUS can parse to. Each node has its own payload and often auxiliary types are involved. Most epigraphic phenomena, like *Ligature*, rely on the type *Cert*, expressing certainty, while *Gap* and *Vacat* rely on the type *Precision*.

Concerning the domain of certainty and precision, and so in general of reliability, ANSELMUS solves the absence of an explicit indicator of certainty or precision with the automatic assignment of the neutral value "Medium". As a consequence, every node in the notation that supports these attributes therefore always has a defined value (be it "Low", "Medium" or "High").

This architecture responds to a specific critique of EpiDoc/TEI practice. In these standards, the omission of the certainty attribute "@cert" generates a semantic aporia: does the absence of marking imply a "certain" fact or an "undeclared" fact? ANSELMUS resolves this ambiguity on the basis of two postulates:

- *Interpretation as an epistemological act*: every notational choice (for example, the distinction between *Erasure* and *Gap*) is the outcome of a deliberate hermeneutic process. There is no such thing as a "neutral" transcription: every mark made by the editor serves as a scientific thesis on the reality of the artefact.
- *Informational saturation*: since the act of transcribing implies an assumption of scientific responsibility, the formulated reading inherently possesses a degree of reliability. The "Medium" value formalizes the "heuristic contract" between scholar and reader, defining the standard of reliability of the editorial proposal in the absence of further specifications.

Crucially, IbiUnit is recursive: an *Unclear* node may contain a *Supplied* node, which may in turn contain a *Gap*—mirroring the TEI P5 content model, where *<unclear>* legitimately contains *<supplied>* and vice versa, as verified against the Guidelines (TEI Consortium, 2026).

The current version of the *ibiscript-formats* Rust crate supports the following output targets:

- *IbiScript*: the serializer is capable of emitting valid IbiScript directly from the AST, exploiting the idempotency of the parser. Depending on configuration, this functions either as a formatter—normalizing whitespace and delimiter style to a canonical, human-readable form—or as a minifier, producing the most compact valid representation for storage or transmission. Round-tripping through the parser and this emitter is therefore a lossless operation by construction.
- *Unicode*: plain text expressing its formatting exclusively through pure Unicode symbols, intended as a fast audit format especially in editor contexts. Being plain text, it is the only format directly diffable with standard versioning tools such as *git diff*, making it ideal for philological version control workflows.
- *EpiDoc*: XML conforming to the latest EpiDoc guidelines (version 9.8; Bodard et al., 2026) and thus TEI P5. As anticipated, the serialization is epistemically opinionated regarding reliability attributes such as "@cert", and features minimal inline comments referencing the IbiScript source wherever EpiDoc lacks the syntax to express a feature—most notably multi-steepness plausibility chains. These comments function as a bidirectional bridge: a philologist receiving only the XML can trace back to the original notation without requiring access to the authoring system.
- *HTML*: a critical digital edition featuring the epigraphic text and the apparatus at the bottom, when present. Each element is hoverable, providing a clear tooltip and explicit phenomenon boundaries. Crucially, notation delimiters are excluded from the selectable text, so that copying the edited text

yields pure transcription without critical markup—while the full notation remains accessible via a dedicated button. The CSS stylesheet is the only artifact modified by project-level configuration, leaving the HTML markup itself agnostic and stable for long-term archival. Advanced modes are available, including an analytical view that exposes the metadata of each *LexicalUnit* inline.

- *Typst*: a highly modular and customizable script ready to be compiled to PDF. Each phenomenon, rather than being rendered inline according to a hardcoded convention, is expressed as a named Typst function. All function definitions are collected in a rich, commented header placed at the beginning of the exported file, alongside a reference list of delimiters. Project-level configuration modifies exclusively this header, leaving the document body entirely agnostic: redefining a single function globally changes how the corresponding epigraphic phenomenon appears in the compiled output, without touching the transcription itself.
- *PDF*: a critical digital edition typeset as a printable document, compiled directly from the Typst intermediate step. Because the PDF is derived deterministically from the same IbiScript source through a verifiable intermediate, it is fully reproducible at any point in time—a property rarely guaranteed by traditional digital edition workflows.

Fig. 1 illustrates the complete processing pipeline, from plain-text IbiScript notation to the output targets described above. The modular nature of the ANSELMUS parser and the clear separation of concerns in the *ibiscript-formats* crate mean that adding a new serialization format requires no modifications to the core parser—only a new concrete implementation of the relevant Rust traits.

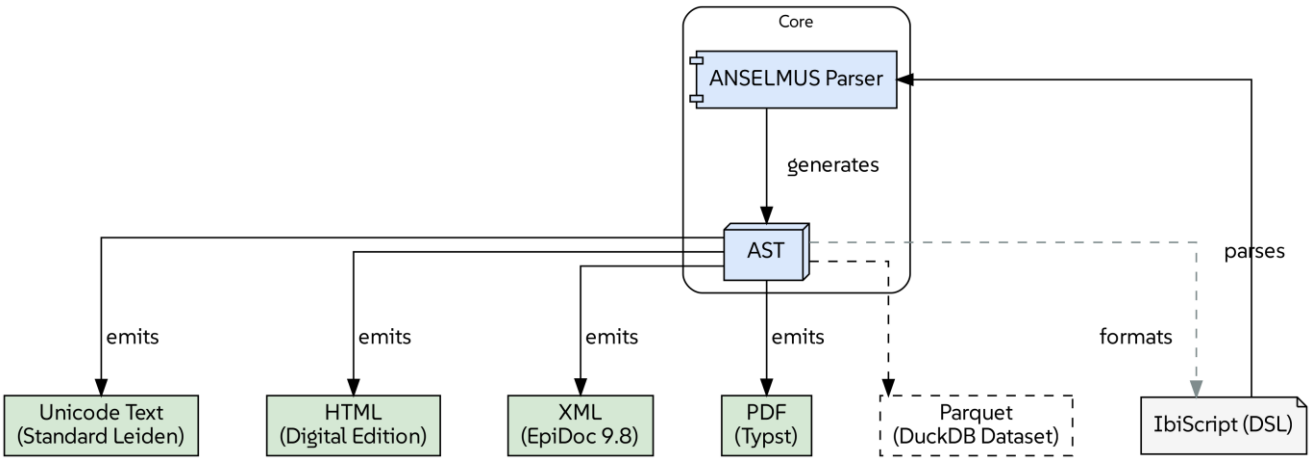

Fig. 1: IbiScript processing pipeline. High-level data flow from the IbiScript DSL through combinatorial parsing to several output formats, including a regenerated DSL representation, which effectively acts as a formatter due to IbiScript's idempotency.

3.4 Example-Driven Validation and Living Documentation

With the core parsing logic fully stabilized, the system architecture prioritizes rigid regression testing and cross-format consistency over generative test prototypes. Rather than relying on synthetic test generation, ANSELMUS evaluates correctness through an extensive, curated suite of test cases covering every epigraphic phenomenon. Because IbiScript theoretically supports infinite recursion, these test cases explicitly validate complex, deeply nested structures up to a depth of two, directly corresponding to real philological demands.

This comprehensive matrix of examples serves as the single source of truth for both parser validation and rich documentation. A dedicated Python orchestration pipeline utilizes the ANSELMUS parser and IbiScript serialization tools to process these test cases, executing a multi-tiered compilation and deployment workflow:

- *Structural validation*: every test case is compiled into compliant EpiDoc XML, validating database schema compatibility and standard epigraphic interoperability.

- *Print documentation*: the examples are natively integrated into a comprehensive reference manual engineered in Typst, aimed at philologists and designed for physical publication.
- *Digital documentation*: the exact same Typst assets undergo a custom transpilation pass into Markdown and HTML, which are then injected directly into a mdBook instance for online publication.

By tightly coupling compiler validation with documentation generation, the pipeline guarantees absolute synchronicity across all physical and digital media. A feature cannot be documented without its underlying parser implementation passing regression testing, ensuring that print manuals, digital portals, and the Rust core remain perfectly aligned.

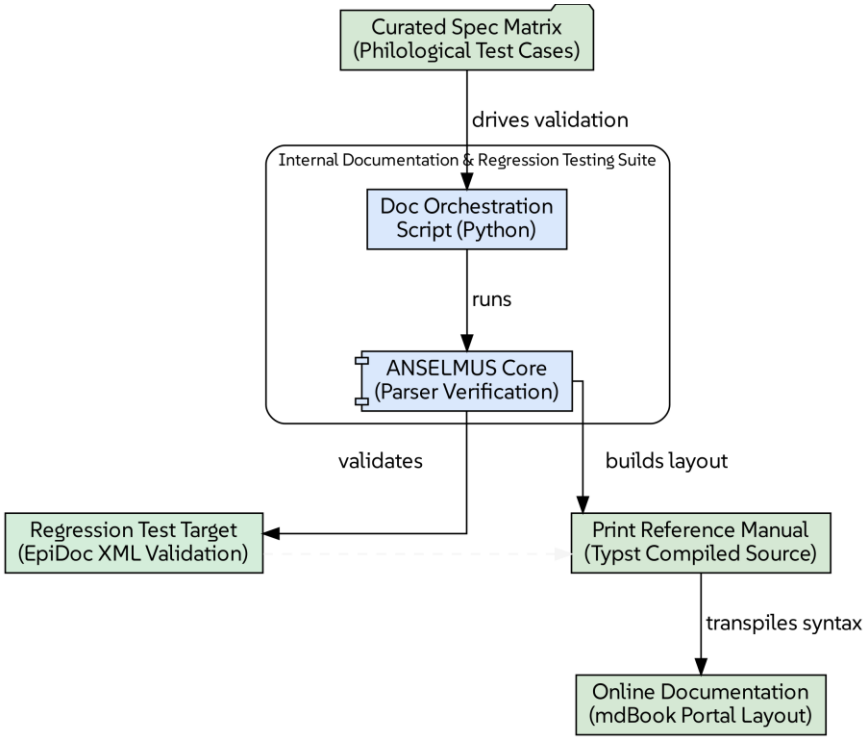

Fig. 2: The automated validation and multi-tiered compilation engine of the ANSELMUS/IbiScript system. A curated philological test matrix serves as the single source of truth, simultaneously driving interoperability validation and documentation generation via a strict regression gate that blocks manual deployment upon compilation failure.

3.5 Case Study: The Bilingual Monument of Paikuli

The development of ANSELMUS/IbiScript is framed within the Project of Excellence of the Department of Ancient World Studies (DiSA) at Sapienza University of Rome, titled "Persistence of the Ancient" (Persistenza dell'antico). This institutional context motivated the development of new digital tools capable of handling the fragmentary and complex nature of Iranian epigraphy.

The practical validity of the system is demonstrated through its application to the Paikuli inscription, a monumental Middle Persian and Parthian bilingual text dating to the late 3rd century CE. Historically, the study of this monument has relied on the foundational work of Herzfeld (1924) and the comprehensive edition by Skjærvø & Humbach (1978-1983). More recently, the corpus has been re-examined and expanded by Cereti & Terribili (2014, 2022), whose research forms the philological backbone of the present digital edition.

The development of ANSELMUS and IbiScript was preceded by an active prototyping phase that built directly upon the methodological and architectural conclusions of the author's master's thesis regarding digital frameworks and

algorithmic methods for Western Middle Iranian epigraphy (Marruzzo, 2024). This prototyping was executed within the Paikuli Digital Epigraphy (PDE) project—a digital edition initiative developed by the author within the institutional framework of DiSA, in close collaboration with the project's principal investigators, Prof. Cereti and Prof. Terribili. The initial operational paradigms of the BEDA web platform, alongside its foundational typographic and multi-period analytic strategies, were subsequently presented and evaluated within the wider context of regional digital heritage initiatives (Marruzzo, 2025).

The constraints encountered in that context—strict Unicode handling for both the Iranian scripts and the diacritically rich characters of their scholarly transliteration, recursive notation structures, and the need for deterministic output across multiple formats—proved incompatible with regex-based approaches and drove the architectural decisions documented in this paper. ANSELMUS and IbiScript are therefore not a theoretical proposal but the direct outcome of iterative, corpus-driven development against real philological data.

A primary test case is block E17\18 of the inscription (Cereti & Terribili, 2014, p. 352).

This ashlar is representative of the philological complexity typical of the corpus: it contains unclear signs, supplied text, lacunae, fractures and critical apparatus entries reflecting ongoing scholarly disagreement.

Notation. The block is encoded in IbiScript as follows. The "!]" and "[!" combinations denote respectively a fracture to the left and to the right of the support, not necessarily interrupting the text; "//" and "/*…*/" denote, respectively, line and block comments, which are never serialized to any output format; a single "/" delimiter marks line breaks within the text, ending with explicit line numbering (when applicable); "-" at line boundaries signals word interruption, in the context of the Paikuli inscription a continuation from or into an adjacent block; "$M:content$" anchors a critical note; "^M:text^" provides the corresponding apparatus entry; "#key:locus=value;…#" defines loci that can be referenced elsewhere and where the key may correspond to the identifier of another critical edition (created with the BEDA platform, which uses IbiScript). The example has been formatted by IbiScript's built-in formatter, which optimizes the layout for readability in monospaced fonts.

```
// Text
$<-$ /* This means the text is from right to left */ $lang = pal-Phli$
!]*\u{s-with-caron}*p[] HWE /1

$1:[\u{right-half-ring}whrm]*z*d*y ZY* wr*\u{right-half-ring}*[c]$ /2

*w*[]t*y?* nwky *klp*[kyhy?] /3

!]*W*N n*rs*[hy ZY] bgy ZY $2:[-5-?]$ /4

$3:-y$[] W \u{s-with-caron-capital}*M* []*kd?* *n*[! /5

!]$4:[YKT?]L*WNn*$ *W*m AD*Y*[N] /6

// Apparatus

^1:The anthroponym \i{Ohrmazd \u{i-with-macron} War\u{a-with-macron}z} is attested in the Paikuli inscription in
#
 Paikuli a7: row = a; block = 7; ln = 6 /
 Paikuli C14: row = C; block = 14; ln = 4
#.^
 ^2:The faint surviving signs seem to support a reading &n*rs*[hy]&, though other interpretations are certainly possible.^
```

```
^3:We have considered this to be the final letter of MP &*\u{s-with-caron}p\u{s-with-caron}y*l[\u{right-half-ring}l]y&, possibly found at the end of
#
 Paikuli E16\17: row = E; block = 16\17; ln = 5
#. Should this be so, it would represent the only surviving link between the two blocks.^
^4:To be reconstructed as &[YKT]LWNn& “I shall kill”, corresponding to &kw\u{s-with-caron}t& found in Parthian
#
 Paikuli e4: row = e; block = 4; ln = 3
#.^
```

Unicode output. The IbiScript Unicode serializer with default configuration produces the following output directly from the AST, without any post-processing. Since the Unicode export is plain text, the italic formatting marked with “\i{…}” as well as the metadata are lost, preserved only in richer formats (like HTML in Fig. 3 and EpiDoc in Fig. 4).

```
1    ¬◖š◗p[···] HWE /
2    ≪[’whrm]◖z◗d◖y ZY◗ wr◖’◗[c]≫[1] /
3    ◖w◗[···]t◖y◈◗ nwky ◖klp◗[kyhy◈] /
4    ¬◖W◗N n◖rs◗[hy ZY] bgy ZY ≪[·5◈·]≫[2] /
5    ≪-y≫[3][···] W Š◖M◗ [···]◖kd◈◗ ◖n◗⊢ /
6    ¬≪[YKT◈]L◖WNn◗≫[4] ◖W◗m AD◖Y◗[N] /

[1]The anthroponym Ohrmazd ī Warāz is attested in the Paikuli inscription in [Paikuli a7: row a, block 7, ln 6 | Paikuli C14: row C, block 14, ln 4].
[2]The faint surviving signs seem to support a reading n◖rs◗[hy], though other interpretations are certainly possible.
[3]We have considered this to be the final letter of MP ◖špšy◗l[’l]y, possibly found at the end of [Paikuli E16\17: row E, block 16\17, ln 5]. Should this be so, it would represent the only surviving link between the two blocks.
[4]To be reconstructed as [YKT]LWNn “I shall kill”, corresponding to kwšt found in Parthian [Paikuli e4: row e, block 4, ln 3].
```

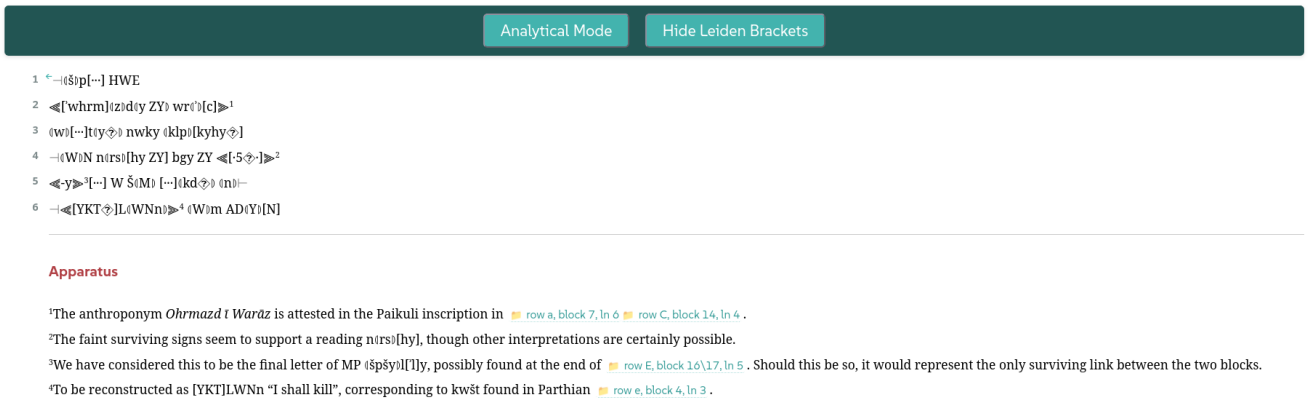

Fig. 3: Web application rendering of the serialized IbiScript text. The user interface maintains a clean default reading view while offering interactive elements that dynamically display structural markers upon mouse hover.

```
<text>
    <body>
        <div type="edition">
            <ab xml:lang="pal-Phli">
                <lb n="1" style="text-direction:r-to-l"/>
                <milestone unit="fracture" rend="left"/>
                <unclear cert="medium">š</unclear>p
                <gap reason="lost" extent="unknown" unit="character" precision="medium"/>
                HWE
                <lb n="2" style="text-direction:r-to-l"/>
                <seg xml:id="app-1">
                    <supplied reason="lost" cert="medium">'whrm</supplied>
                    <unclear cert="medium">z</unclear>d<unclear cert="medium">y ZY</unclear> wr
                    <unclear cert="medium">'</unclear>
                    <supplied reason="lost" cert="medium">c</supplied>
                </seg>
                <lb n="3" style="text-direction:r-to-l"/>
                <unclear cert="medium">w</unclear>
                <gap reason="lost" extent="unknown" unit="character" precision="medium"/>t<unclear cert="low">y
            </unclear> nwky
                <unclear cert="medium">klp</unclear>
                <supplied reason="lost" cert="low">kyhy</supplied>
                <lb n="4" style="text-direction:r-to-l"/>
                <milestone unit="fracture" rend="left"/>
                <unclear cert="medium">W</unclear>N n
                <unclear cert="medium">rs</unclear>
                <supplied reason="lost" cert="medium">hy ZY</supplied>
                bgy ZY
                <seg xml:id="app-2">
                    <gap reason="lost" quantity="5" unit="character" precision="low"/>
                </seg>
                <lb n="5" style="text-direction:r-to-l"/>
```

Fig. 4: EpiDoc serialization. A portion of the interoperable EpiDoc XML export generated from the reference text sample, demonstrating the serializer's native TEI P5 output capabilities.

Discussion. This example demonstrates three core advantages of the IbiScript system within a formal philological workflow. First, complex phenomena that frequently co-occur in Iranian epigraphy—such as unclear signs adjacent to supplied text or apparatus anchors wrapping multi-phenomenon sequences—are handled through recursive AST nesting rather than arbitrary string concatenation. This approach preserves the structural integrity of every editorial decision. Second, the apparatus is encoded inline with its base text, while the note content is sequestered into a distinct block; this architecture mirrors the layout of printed critical editions while ensuring the data remains fully machine-readable. Third, the Unicode output is derived deterministically from the AST. Consequently, a specific input consistently produces the same output, which can be regenerated at any time from the stored notation without the loss of philological nuance.

Line 6 of block E17\18 warrants particular attention regarding the requirements of modern digital critical editions. Cereti & Terribili (2014) propose the verbal form "[YKT]LWNn" at the beginning of the line, yet they restrict this reconstruction to an apparatus note, leaving a cautious "[…]" in the main critical text. To prioritize interoperability and searchability, we have opted to make the most widely accepted philological reconstruction transparent. We encoded the sequence as "[YKT?]L*WNn*" (using *Supplied*([*Plain*("YKT")], *cert*: Low) rather than "[.3.]L*WNn*" (using *Gap*(*Known*(3), *precision*: Medium)). While this continues to signal significant uncertainty, it provides a version of the text that aligns more closely with the authors' scholarly intent and is significantly easier for computational tools to index and process.

The current implementation covers all phenomena attested in the Paikuli corpus. All other features—including surplus text, additions, lacunae, apparatus, text partitioning—are fully operational and sufficient for a complete critical edition of the inscription.

3.6 ANSELMUS/IbiScript as Reusable Components

ANSELMUS and IbiScript are designed from the outset as standalone libraries rather than a monolithic application. This architectural choice has direct consequences for its deployment profile and its potential for adoption beyond the immediate research context in which it was conceived.

The library currently serves as the text engine of BEDA, a forthcoming epigraphic digital edition platform built entirely in Rust and compiled to WebAssembly. Within this framework, ANSELMUS occupies the role of a low-level kernel: it receives IbiScript notation as input and returns a typed AST that the higher-level application layers use for rendering, querying, and export.

This separation of concerns means that the platform's presentation logic is entirely decoupled from the parsing logic—a change in

the output format requires no modification to the parser, and a change in the notation requires no modification to the application.

The most significant deployment advantage of a Rust-based kernel is its compilation target flexibility. The libraries compile to WebAssembly without modification, enabling real-time client-side parsing directly in the browser without server infrastructure. This is a qualitative departure from the current generation of digital edition platforms—EFES, TEI Publisher, eXist-db—which require a running server process to perform any parsing or transformation operation.

A WASM-compiled ANSELMUS/IbiScript instance can be embedded in a static web page, opening the possibility of fully offline-capable epigraphic editors without server-side dependencies. Python bindings via PyO3 and maturin expose the full parsing API to Python consumers, allowing integration with existing DH pipelines without requiring Rust expertise from the end user. This interoperability is particularly relevant for the DH community, where Python remains the dominant language for corpus processing and machine learning workflows. For the most experienced users, the *ibis* CLI enables to access easily to the full parsing and serialization pipeline, with support for batch processing.

Broadly, ANSELMUS adheres to a UNIX-style design philosophy centered on the AST as a universal intermediate representation. Its primary responsibility is to transform idiosyncratic epigraphic notation into a structured, typed AST. While the library includes high-performance renderers to transform this AST into various formats, these are implemented as pure functions that remain entirely decoupled from the parsing logic. This ensures the library remains agnostic regarding the calling environment: it can be embedded in a desktop application, compiled to a command-line tool, served as a WebAssembly module, or called from Python—serving as a stable, side-effect-free "epigraphic kernel" for any DH pipeline.

Looking forward, the development roadmap prioritizes a first-class, seamless integration with the broader BEDA ecosystem. Rather than relying on the fragmented configuration of general-purpose text editors, future iterations will focus on constructing a dedicated, domain-specific toolchain.

This includes the implementation of a custom Language Server Protocol (LSP) to handle real-time epigraphic validation and diagnostics, alongside tailored syntax highlighting. Ultimately, these components will culminate in a full, opinionated, and highly focused Rust-native IbiScript editor, establishing a definitive, official environment designed specifically for the rigorous demands of digital epigraphy.

4. Future Roadmap

The core architectures of ANSELMUS and IbiScript continue to expand into adjacent domains, transitioning from a structured transcription system into a high-performance data pipeline for epigraphic analysis and machine learning.

With the deterministic relationship between IbiScript notation and the AST fully established, the architecture naturally lends itself to large-scale data serialization. Rather than relying on volatile text streams, a high-performance Parquet serialization layer is being introduced. This encapsulates parsed inscriptions into an immutable, columnar data format that retains the full structural fidelity of the AST.

This layout transforms the archive into an instantly queryable database via analytical engines like DuckDB, bypassing the need for heavyweight relational database migrations. Furthermore, because Parquet is natively integrated into the Hugging Face datasets ecosystem, this architecture provides a direct, zero-overhead pipeline for feeding Domain-Specific Language Models (DSLM) with highly structured ancient textual data.

This Parquet-backed data layer integrates directly into the final phase of the project: an optical character recognition (OCR) post-processing and validation engine. When the OCR system processes inscription photographs or synthetic glyph renders, candidate transcriptions along with their spatial and confidence metadata can be emitted straight into the Parquet data stream.

ANSELMUS then acts as a deterministic validation gate, running the candidate text noted in IbiScript through its parser to normalize readings, flag epigraphic structural anomalies, and cross-reference variants. By storing both the OCR hypotheses and the normalized IbiScript outputs in the same Parquet architecture,

researchers can utilize DuckDB to perform immediate, vectorized accuracy auditing, completing the workflow from raw rock-face digitization to critical edition within a single, unified framework.

Finally, to ensure maximum academic reproducibility, long-term software sustainability, and seamless integration within adjacent DH pipelines, the core technical infrastructure developed throughout this research cycle will be transitioned into the public domain. Upon the conclusion of the currently funded project, the full software suite will be released under an open-source distribution model.

The reference implementation of the parsing compiler engine, ANSELMUS (Marruzzo, 2026a), the core DSL and serialization framework, IbiScript (Marruzzo, 2026c), and the comprehensive backend storage archive ecosystem, BEDA (Marruzzo, 2026b), will be made publicly available across official open-access registries. This ensures that the technical outputs of the Paikuli digital workspace remain an immutable, community-driven resource, effectively mitigating the software decay and institutional isolation that frequently challenge DH projects.

5. Conclusions

ANSELMUS and its DSL IbiScript represent a departure from the two dominant paradigms in epigraphic digital editing: the regex-to-XML model of first-generation systems such as Leiden+, and the generative grammar approach of more recent DSL-based frameworks such as CoPhiEditor. By treating the AST as the primary source of truth rather than a transient intermediate state, IbiScript decouples the act of scholarly notation from the act of format production—a separation that has architectural, philological, and practical consequences.

Architecturally, the combinator-based parser is a readable, auditable specification of the epigraphic grammar, independently testable at the level of each philological phenomenon. Each atomic parser enforces the respective constraints of the Leiden Conventions at compile time, making a class of structural errors impossible rather than merely detectable. The separation between ANSELMUS and IbiScript themselves allows new serialization formats to be added without modifying the parser or DSL.

Philologically, the strict separation between input notation and rendered output eliminates the class of ambiguities that affect output-as-input systems. The notation is designed to be writable in a plain text editor, accessible to scholars without specialized software, and unambiguous by construction—equivalent surface forms resolve to identical AST representations.

Practically, the Rust implementation compiles to WebAssembly, enabling client-side deployment without server infrastructure, and exposes a CLI and Python bindings for integration with existing DH pipelines. The system is already operational as the text engine of the BEDA ecosystem and has been validated against the full attested phenomenon inventory of the Paikuli inscriptional corpus.

The directions outlined in the roadmap—machine learning dataset generation and philological OCR post-processing—are not extensions that require architectural revision. They are natural consequences of maintaining the AST as a typed, queryable, persistent object rather than discarding it at the point of serialization. This is, ultimately, the central claim of ANSELMUS: that the grammar of a critical edition is too valuable to be used once and thrown away.

References

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). *Compilers: Principles, techniques, and tools* (Pearson new international edition, 2nd ed.). Essex, United Kingdom: Pearson.

Barats, C., Schafer, V., & Fickers, A. (2020). Fading Away... The challenge of sustainability in digital studies. *DHQ: Digital Humanities Quarterly*, *14*(3). https://dhq-static.digitalhumanities.org/pdf/000484.pdf

Baumann, R., Bodard, G., Cayless, H., Sosin, J., & Viglianti, R. (2011). Integrating Digital Papyrology. In *Big Tent Digital Humanities*. Presented at the ADHO, Stanford, CA.

Bodard, G. (2010). EpiDoc: Epigraphic Documents in XML for Publication and Interchange. *Latin On Stone: Epigraphic Research and Electronic Archives*, 101–118.

Bodard, G., Mylonas, E., Elliott, T., Stoyanova, S., Tupman, C., & Vagionakis, I. (2026). EpiDoc Guidelines post 9.8 dev. Retrieved from https://epidoc.stoa.org/gl/latest/

Burnard, L. (2014). *What is the Text Encoding Initiative?: How to add intelligent markup to digital resources*. Marseille, France: OpenEdition Press. https://doi.org/10.4000/books.oep.426

Burnard, L., & Baumann, S. (2022). TEI P5: Guidelines for Electronic Text Encoding and Interchange. Retrieved from https://guidelines.teipublisher.com/

Cayless, H., Roueché, C., Elliott, T., & Bodard, G. (2009). Epigraphy in 2017. *DHQ: Digital Humanities Quarterly*, *3*(1).

Cereti, C. G., & Terribili, G. (2014). The Middle Persian and Parthian Inscriptions on the Paikuli Tower. *Iranica Antiqua*, 347–412.

Cereti, C. G., & Terribili, G. (2022). Epigraphic Findings at Paikuli (2018-2019). A Preliminary Study. *Vicino Oriente*, *26*, 53–75. https://doi.org/10.53131/VO2724-587X2022_4

Chaniotis, A., Corsten, T., Papazarkadas, N., & Tybout, R. A. (2026). Supplementum Epigraphicum Graecum Online (SEGO). Retrieved from https://scholarlyeditions.brill.com/sego/

Couprie, G. (2026). *nom: Rust parser combinator framework*. Retrieved from https://github.com/rust-bakery/nom?tab=readme-ov-file

Cummings, J. (2018). A world of difference: Myths and misconceptions about the TEI. *Digital Scholarship in the Humanities*. https://doi.org/10.1093/llc/fqy071

Davis, M. (2025). UTS #18: Unicode Regular Expressions. Retrieved from https://www.unicode.org/reports/tr18/?utm_source=copilot.com

Del Grosso, A. M., Zenzaro, S., Boschetti, F., & Ranocchia, G. (Eds.). (2024). Bridging Traditional and Digital Papyrology with Domain-Specific Languages. The GreekSchools Case Study. In *The Digital Critical Edition of Greek Papyri: Issues, Projects, and Perspectives*: *Vol. III*. Berlin, Germany: De Gruyter. https://doi.org/10.1515/9783111070162

Dow, S. (1969). *Conventions in Editing: A Suggested Reformulation of the Leiden System*. Durham, NC.

Faghihi, Y., Holford, M., & Jones, H. (2022). Teaching the Text Encoding Initiative: Context, Community and Collaboration. *Journal of Open Humanities Data*, *8*, 15. https://doi.org/10.5334/johd.72

Fenlon, K. S. (2020). Sustaining Digital Humanities Collections: Challenges and Community-Centred Strategies. *International Journal of Digital Curation*, *15*(1), 13. https://doi.org/10.2218/ijdc.v15i1.725

Herzfeld, E. (1924). *Paikuli: Monument and Inscription of the Early History of the Sasanian Empire* (D. Reimer; Ernst Vohsen, Vols. 1–2). Berlin, Germany: D. Reimer.

Ide, N. M., & Sperberg-McQueen, C. M. (1995). The TEI: History, Goals, and Future. *Computers and the Humanities*, *29*(1).

King's College London, Department of Digital Humanities. (2024). *Kiln: A framework for publishing XML and TEI content* [JavaScript]. [Legacy] Department of Digital Humanities, King's College London. Retrieved from https://github.com/kcl-ddh/kiln (Original work published 2011)

Marruzzo, A. (2024). *Middle Persian and Digital Innovation. New Web Implementations, Algorithmic Methods and Fonts* (Unpublished master's thesis). Roma, Italy.

Marruzzo, A. (2025). *Digitising Iranian Scripts: The Paikuli Web Platform and a New Pahlavi Typeface as Tools for Multi-Period Analysis*. Presented at the PolEmA - Polycentric Empires in Western Asia, Roma, Sapienza University of Rome.

Marruzzo, A. (2026a). *ANSELMUS* [Rust]. Retrieved from https://crates.io/crates/anselmus

Marruzzo, A. (2026b). *BEDA* [Rust]. Retrieved from https://crates.io/crates/beda

Marruzzo, A. (2026c). *IbiScript* [Rust]. Retrieved from https://crates.io/crates/ibiscript

Martinelli, N. (2025, December 18). Software rot: Saving science's digital legacy. Retrieved from Software Heritage website: https://www.softwareheritage.org/2025/12/18/software-rot-saving-sciences-digital-legacy/

Materni, M. (2020). Complessità della codifica ed ergonomia strumentale nel contesto XML-TEI: Dove siamo? (Bilancio a partire da un nuovo progetto di edizione digitale medievale). *Umanistica Digitale*, *No 8*, research in the age of Digital Humanities. https://doi.org/10.6092/ISSN.2532-8816/9976

Mugelli, G., Re, G., & Taddei, A. (2020). Annotazione digitale di testi antichi. Lingue antiche e Digital Humanities, tra ricerca e didattica. *Umanistica Digitale*, 35–60. https://doi.org/10.6092/ISSN.2532-8816/9962

Panciera, S. (Ed.). (1982). Epigrafia e ordine senatorio. *Atti Del Colloquio Internazionale AIEGL Di Roma*, *4–5*, IX–XIII. Rome, Italy: Edizioni di storia e letteratura.

Panciera, S. (2012). What Is an Inscription? Problems of Definition and Identity of an Historical Source. *Zeitschrift Für Papyrologie Und Epigraphik*, (183), 1–10.

Pape, S., Schöch, C., & Wegner, L. (2012). TEICHI and the Tools Paradox: Developing a Publishing Framework for Digital Editions. *Journal of the Text Encoding Initiative*, (Issue 2). https://doi.org/10.4000/jtei.432

Papyri.info Collaborative. (2013). Leiden+ Documentation. Retrieved from Papyri.info website: https://papyri.info/docs/leiden_plus

Papyri/SoSol. (2010–2025). SoSol: Son of Suda On-Line / Papyri.info collaborative editing platform. Retrieved from https://papyri.info/

Pichler, A. (2021). Hierarchical or Non-hierarchical? A Philosophical Approach to a Debate in Text Encoding. *DHQ: Digital Humanities Quarterly*, *15*(1).

Rezania, K., Cantera, A., Eide, Ø., & Neuefeind, C. (2021–2030). Zoroastrian Middle Persian: Digital Corpus and Dictionary (MPCD). Ruhr-Universität Bochum / Freie Universität Berlin / Universität zu Köln. Retrieved from https://www.mpcorpus.org/

Schmidt, D. (2014). Towards an Interoperable Digital Scholarly Edition. *Journal of the Text Encoding Initiative*, (Issue 7). https://doi.org/10.4000/jtei.979

Schubart, W. (1918). *Einführung in die Papyruskunde*. Berlin, Germany: Weidmannsche Buchhandlung.

Sipser, M. (2013). *Introduction to the Theory of Computation* (3rd ed., international edition). Boston, MA: Cengage Learning.

Skjærvø, P. O., & Humbach, H. (1978–1983). The Sassanian inscription of Paikuli (Vols. 1–3). Wiesbaden, Germany: Reichert.

Sosin, J. (2012). Digital Papyrology. In P. Schubert (Ed.), *Proceedings of the 26th International Congress of Papyrology* (pp. 767–772). Bibliothèque d'Etudes Papyrologiques.

TEI Consortium. (2026). TEI P5: Guidelines for Electronic Text Encoding and Interchange. Retrieved from https://www.tei-c.org/release/doc/tei-p5-doc/en/html/

Wilcken, U. (1932). *Das Leydener Klammersystem*. Leipzig, Germany: B.G. Teubner Verlagsgesellschaft.

Williams, A. C., Santarsiero, A., Meccariello, C., Verhasselt, G., Carroll, H. D., Wallin, J. F., … Brusuelas, J. H. (2015). Proteus: A platform for born digital critical editions of literary and subliterary papyri. *2015 Digital Heritage*, 453–456. Granada, Spain: IEEE. https://doi.org/10.1109/DigitalHeritage.2015.7419546