

Alfredo Cosco

Aracne2

*Una piattaforma modulare
per l'edizione e la pubblicazione di corpora TEI*

Federico II University Press



fedOA Press

Università degli Studi di Napoli Federico II

Saperi storici e nuove tecnologie, 5

Alfredo Cosco

Aracne2

*Una piattaforma modulare
per l'edizione e la pubblicazione di corpora TEI*

Federico II University Press



fedOA Press

Aracne2. Una piattaforma modulare per l'edizione e la pubblicazione di corpora TEI / Alfredo Cosco. – Napoli : FedOAPress, 2026. – 56 p. ; 21 cm. – (Saperi storici e nuove tecnologie: 5).

Accesso alla versione elettronica:

<http://www.fedoabooks.unina.it>

ISBN: 978-88-6887-439-1

DOI: 10.6093/978-88-6887-439-1

© 2026 FedOAPress – Federico II University Press
Università degli Studi di Napoli Federico II
Centro di Ateneo per le Biblioteche “Roberto Pettorino”
Piazza Bellini 59-60
80138 Napoli, Italy
<http://www.fedoapress.unina.it/>
Published in Italy
Gli E-Book di FedOAPress sono pubblicati con licenza
Creative Commons Attribution 4.0 International

Indice

1. Introduzione.....	7
2. Una traiettoria ventennale	11
2.1 Le origini: il progetto PRIN 2001	11
2.2 MaRa (2008–2016)	13
2.3 Aracne (2016–2026).....	18
2.4 Aracne2 (2026 →).....	21
3. Il flusso di lavoro editoriale	25
4. Il modello di pubblicazione.....	31
5. L'AI come collaboratore non accessorio.....	35
5.1 Assistenza in editor	36
5.2 Automazione bibliografica	38
5.3 Integrazione di assistenti esterni via MCP	38
5.4 Scelta del provider	39
5.5 Retrieval-augmented generation su TEI	40
6. Architettura, in sintesi.....	43
7. Prospettive	47
7.1 Una nota di metodo	49
Bibliografia.....	51
Riferimenti tecnici	56

1. Introduzione

L'edizione digitale di corpora testuali strutturati — diplomi, edizioni critiche, cataloghi manoscritti, dossier prosopografici — è da quasi tre decenni una preoccupazione costante delle Digital Humanities. Le TEI Guidelines, ormai alla loro quinta iterazione, mettono a disposizione un vocabolario di oltre 1500 elementi per descrivere testi, testimoni materiali, apparati editoriali e contesti culturali. Eppure, nei primi anni Venti del secolo, il percorso che porta da un flusso di trascrizioni dattiloscritte — prodotte negli anni dal gruppo editoriale di un progetto — a un'edizione digitale pubblicata, citabile, esplorabile e machine-actionable, resta ancora, per la maggior parte dei progetti, un percorso da affrontare in solitudine.

La frizione si manifesta in due punti. Il primo è editoriale: l'editor di un progetto piccolo o medio raramente parla XML come lingua madre, e non dovrebbe

essere costretto a farlo. Vuole marcare un nome come `persName` con un identificativo Wikidata senza dover sapere che cos'è un attributo; vuole una bibliografia coerente su duemila voci senza dover imparare la grammatica TEI di `biblStruct`; vuole sapere se ciò che ha scritto è valido rispetto allo schema del progetto non dopo, ma prima di averlo licenziato. Il secondo è la pubblicazione: una volta che il corpus è codificato, trasformarlo in qualcosa che il resto del mondo possa leggere, cercare, raccogliere, citare e collegare è un problema ingegneristico a sé, con le sue dipendenze da framework web, motori di ricerca, flussi di deposito e standard di metadati. I CMS editoriali che gestiscono bene una delle due metà tendono a gestire male l'altra, e ogni nuovo progetto che ha bisogno di entrambe finisce per reimplementare la metà mancante.

Aracne2 è una piattaforma costruita per assorbire entrambe le metà in un unico prodotto modulare. Questo testo presenta la sua progettazione e i diciotto anni di lavoro di tooling che vi hanno condotto: un arco lungo che inizia con un set di script PHP per i Registri della Cancelleria Angioina nel 2008, attraversa una piattaforma `eXist-db` / XQuery operativa fra il 2016 e il 2026 a sostegno di più edizioni accademiche, e arriva a una terza iterazione la cui scommessa distintiva è che l'assistenza AI — incluse `retrieval-augmented generation` ed `export Model`

Context Protocol — appartenga al centro dell'ambiente editoriale, e non ai suoi margini.

Il contributo è organizzato come segue. Il § 2 ricostruisce la traiettoria MaRa → Aracne → Aracne2 e legge ogni passaggio come risposta a un vincolo che il precedente non era riuscito ad assorbire. Il § 3 descrive il flusso editoriale supportato da Aracne2: editing schema-aware, controllo di autorità, automazione bibliografica, lavoro sulle entità nominate, allineamento facsimile-testo. Il § 4 descrive il modello di pubblicazione: collezioni, siti pubblici, motori di ricerca, backend di deposito, harvesting OAI-PMH, esportazione linked open data. Il § 5 — la sezione più ampia — presenta l'integrazione AI come concern di prima classe della piattaforma, con particolare attenzione alla retrieval-augmented generation ancorata alle TEI Guidelines e alle implicazioni dell'export MCP per il rapporto fra strumenti desktop dell'editor e CMS editoriale. Il § 6 abbozza poi l'architettura in prosa, senza diagrammi. Nel § 7 si conclude sulle questioni aperte e il lavoro in corso.

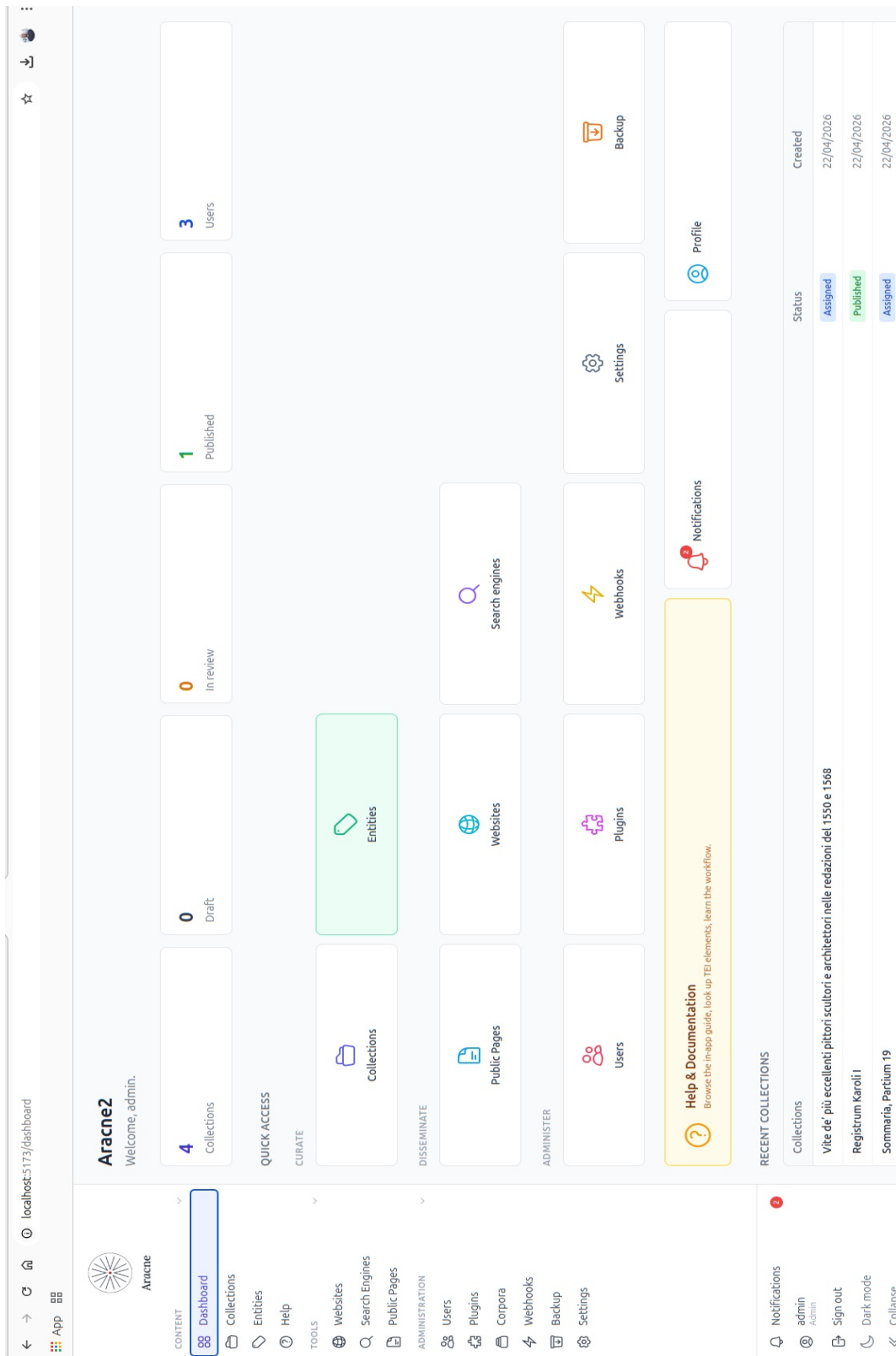


Figura 1: La dashboard di Aracne2, con le collezioni in stati di workflow diversi (assegnata, pubblicata) e le superfici curate/disseminate/administer.

2. Una traiettoria ventennale

Aracne2 è la terza iterazione di una linea di strumenti che da diciotto anni si pone la stessa questione: ***come trasformare un flusso di trascrizioni di manoscritti in un'edizione digitale citabile, esplorabile e machine-actionable senza obbligare il filologo a diventare un programmatore XML?*** Ogni iterazione ha risposto a quella domanda in modo diverso, e i vincoli incontrati ogni volta hanno dato forma a ciò che è venuto dopo.

2.1 Le origini: il progetto PRIN 2001

La genealogia di Aracne2 inizia prima di MaRa. Le sue prime basi concettuali e metodologiche sono state gettate nell'ambito del progetto di ricerca nazionale PRIN 2001 – *La cancelleria angioina. Un sistema informativo digitale per la gestione e l'analisi della documentazione superstite*, coordinato da Roberto Delle Donne,

dedicato al trattamento digitale, alla ricostruzione e allo sfruttamento scientifico della documentazione superstite della cancelleria angioina.¹ Tale progetto ha creato le condizioni per una riflessione approfondita su come una tradizione documentaria altamente frammentaria, trasmessa attraverso testimoni eterogenei, copie, estratti e riferimenti archivistici, potesse essere trasformata in un corpus digitale strutturato, ricercabile e criticamente affidabile.

Il progetto PRIN 2001 non è stato semplicemente un'iniziativa di digitalizzazione. È stato piuttosto un laboratorio sperimentale in cui problemi filologici, archivistici e storici sono stati progressivamente tradotti in procedure computazionali. Il lavoro sui documenti della Cancelleria angioina ha richiesto l'identificazione di strutture documentarie ricorrenti, la normalizzazione dei riferimenti bibliografici abbreviati, la gestione dei rimandi incrociati, la formalizzazione di date, nomi e luoghi e la costruzione di una relazione stabile tra trascrizione, apparato critico e metadati. In questo senso, il progetto ha anticipato molte delle questioni che sarebbero poi diventate centrali per MaRa, Aracne e Aracne2: come

1 Roberto Delle Donne, *La cancelleria angioina. Un sistema informativo digitale per la gestione e l'analisi della documentazione superstite*, 2004 (edizione a stampa per il deposito legale della banca dati testuale). Catalogo IRIS: <https://hdl.handle.net/11588/115894>.

preservare la complessità della documentazione storica rendendola al contempo leggibile dalle macchine; come supportare i curatori senza costringerli a diventare programmatori; come passare da una sequenza di testi trascritti a un'edizione digitale interoperabile.

MaRa, quindi, non è nato in isolamento. Si trattò della prima concretizzazione tecnica di un processo di ricerca più lungo, iniziato con il progetto PRIN 2001 e con il tentativo collettivo di ricostruire, codificare e pubblicare la memoria documentaria della Cancelleria angioina dopo la perdita dei registri originali nel 1943.

2.2 MaRa (2008–2016)

I registri originali della **Cancelleria Angioina** — assieme alla parte più preziosa dei fondi storici dell'Archivio di Stato di Napoli — erano stati trasferiti, per ordine del Ministero dell'Interno, a Villa Montesano in San Paolo Belsito, a una trentina di chilometri da Napoli, con l'esplicita intenzione di sottrarli ai pericoli dei bombardamenti aerei. Più di 30.000 volumi e circa 50.000 pergamene, in larga parte contenuti in 866 casse, erano stati depositati nelle sale di quel solido edificio. Il 30 settembre 1943, quando l'esercito alleato era ormai prossimo, una squadra di tre soldati tedeschi in motocicletta armati di mitragliatrici, arrivò a Villa Montesano, annunciò

che il deposito sarebbe stato dato alle fiamme entro quindici minuti, rifiutò di visionare la lettera che il direttore dell'Archivio di Stato, Riccardo Filangieri, aveva fatto recapitare al comando tedesco locale per illustrare il contenuto del deposito e il carattere strettamente storico dei manoscritti, e procedette a disporre carta, paglia e polvere da sparo nei quattro angoli e al centro di ogni stanza. In pochi minuti, l'intera villa si trasformò in un'immensa pira. Delle 866 casse, le guardie dell'archivio e i contadini del posto riuscirono a portare in salvo soltanto undici casse di registri notarili e novantasette cartoni dell'Archivio Farnese.² Fra i materiali andati distrutti in quel singolo atto di distruzione, c'erano i 378 registri della Cancelleria Angioina relativi agli anni 1265–1435, i registri della Cancelleria Aragonesa, manoscritti, codici e collezioni autografe del Regno di Napoli, i celebri processi politici, e la parte più cospicua degli

² La testimonianza più autorevole di prima mano sulla distruzione è il rapporto redatto da Riccardo Filangieri, allora direttore del Reale Archivio di Stato di Napoli, che aveva organizzato personalmente il trasferimento dei documenti a Villa Montesano ed era sul posto nei giorni circostanti l'evento. Il rapporto fu pubblicato come Riccardo Filangieri, "Report on the Destruction by the Germans, September 30, 1943, of the Depository of Priceless Historical Records of the Naples State Archives", *The American Archivist*, vol. 7, n. 4 (ottobre 1944), pp. 252–255 (DOI: 10.17723/aarc.7.4.y8g06m75585jp8r1). La traduzione dall'italiano fu curata da Arthur H. Leavitt e Salvatore D. Nerboso del National Archives. Le cifre numeriche citate sopra (30.000 volumi, 50.000 pergamene, 866 casse, 11 casse di registri notarili e 97 cartoni dei Farnese salvati) sono tratte direttamente dal rapporto.

archivi della Real Casa di Borbone, della Casa Farnese, della Sommaria, del Consiglio Collaterale, della Real Camera di Santa Chiara, della Segreteria di Stato del Viceré, del Consiglio di Sicilia, dell'Ordine di Malta, della Commissione Feudale e dei più antichi notai.³ La distruzione fu, nelle parole di Filangieri stesso, un vuoto nelle fonti storiche della civiltà europea che nulla potrà mai colmare.

Quel che sopravvive è dunque una ricostruzione frammentaria messa insieme dagli archivisti del Novecento a partire da testimoni indiretti, copie e citazioni sparse fra archivi europei. L'edizione di questo corpus è, per necessità, un giallo filologico: ogni voce porta con sé un proprio apparato di annotazioni di provenienza, rimandi e citazioni bibliografiche in forme abbreviate idiosincratiche, ereditate dall'edizione manuale. Lo stesso vale, in modo parallelo e ugualmente decisivo, per i materiali della Cancelleria Aragonese: la lacuna che il rogo di Villa Montesano ha aperto nella base documentaria dello stato meridionale tardo-medievale è precisamente ciò che i program-

3 La lista dei fondi distrutti a Villa Montesano segue la disposizione fatta da Filangieri stesso nel rapporto citato. Sui registri della cancelleria angioina e sulla loro ricostruzione novecentesca, il riferimento canonico resta Riccardo Filangieri (a cura di), *I Registri della Cancelleria Angioina*, ricostruiti, Napoli, Accademia Pontaniana, 1950 ss., un'impresa editoriale di lungo periodo che è proseguita, sotto direzioni successive, fino ai nostri giorni.

mi editoriali angioino e aragonese hanno cercato, dal dopoguerra in poi, di ricucire.

MaRa è stata la pipeline tecnologica al servizio del progetto di un sistema informativo digitale per la gestione e l'analisi della documentazione superstite. Il quadro storiografico che motivava lo sforzo di digitalizzazione — la centralità della cancelleria e dell'amministrazione fiscale per la storia istituzionale di lungo periodo del Mezzogiorno, la continuità fra l'apparato fiscale angioino e quello aragonese attraverso la *Regia Camera della Sommaria*, e l'impegno metodologico nella lettura "lenta" dei testi amministrativi — è articolato in modo più compiuto nella monografia di Delle Donne sulla burocrazia e la fiscalità nella Napoli del Quattro-Cinquecento.⁴

MaRa (*Marcatore dei Registri Angioini*) era un set di script PHP per prendere i testi dattiloscritti — prodotti dagli editor del progetto in anni di lavoro su file *.txt* — e portarli in una codifica XML personalizzata. Gli script gestivano tre lavori che nessuno aveva voglia di fare a mano:

4 Roberto Delle Donne, *Burocrazia e fisco a Napoli tra XV e XVI secolo. La Camera della Sommaria e il Repertorium alphabeticum solutionum fiscalium Regni Siciliae Cifretanae*, Firenze, Reti Medievali — Firenze University Press, 2012, ISBN 978-88-6655-301-4.

— **Inserimento di tag** — passaggi a espressioni regolari riconoscevano pattern ricorrenti (parti dei documenti, date in numeri romani, abbreviazioni bibliografiche) e proponevano la marcatura all’editor, che la accettava o la correggeva.

— **Armonizzazione bibliografica** — le edizioni a stampa, fonte delle trascrizioni, avevano usato negli anni abbreviazioni diverse per la stessa fonte; MaRa le risolveva contro un file bibliografico canonico e le riscriveva come riferimenti completi.

— **Validazione e reportistica** — controlli sintattici per documento e un report a livello di corpus su abbreviazioni non risolte, riferimenti incrociati mancanti e formati di data inconsistenti.

Intorno al 2010, gli script furono integrati in una UI web basata su CodeIgniter, con l’obiettivo di costruire una prima versione di piattaforma in cui i filologi del progetto potessero eseguire la pipeline in autonomia anziché spedire batch di file allo sviluppatore. Quando MaRa è stato pubblicato/depositato nel 2018,⁵ aveva ormai trattato diverse migliaia di documenti ed era stato l’infrastruttura editoriale di

5 Alfredo Cosco, “Southern Italian Angevin Chancery Papers in XML: the script MaRa v2.0”, 2018. Zenodo: <https://zenodo.org/records/1447195> — Academia.edu: <https://www.academia.edu/37523540/>.

fatto del progetto. Restava però uno **strumento monocorpo, monoserver, privo di livello di pubblicazione**. L'urgenza di processare il corpus già in lavorazione ha fatto sì che MaRa non fosse mai adottato al di fuori della cerchia ristretta che lo aveva sviluppato.

2.3 Aracne (2016–2026)

La lezione di MaRa era chiara: il flusso editoriale era generalizzabile, quello di pubblicazione no. **Aracne**⁶ è stato il primo tentativo di trasformare il flusso angioino in una piattaforma — qualcosa che altri progetti filologici potessero adottare senza ereditare sei anni di PHP project-specific. La scommessa era sull'**ecosistema eXist-db / XQuery**, che intorno al 2015-2016 sembrava il luogo naturale per un CMS editoriale XML-native: un solo linguaggio (XQuery 3.1) per archiviazione, trasformazione, templating e routing, sostenuto da una comunità di digital humanists che già lo parlava.

Aracne metteva in campo un editor TEI basato su CodeMirror con autocompletamento degli attributi, un workflow bozza → revisione → pubblicazione con role

⁶ Alfredo Cosco (2019) "orazionelson/aracne: aracne1.1". Zenodo. doi:10.5281/zenodo.2647506.

gating, un sitebuilder che produceva edizioni HTML statiche e un'interfaccia di ricerca — il tutto in XQuery su eXist-db. La piattaforma ha funzionato, e fra il 2018 e il 2026 ha sostenuto più edizioni accademiche all'interno di due reti di ricerca complementari. Il lavoro originario sulla Cancelleria Angioina è proseguito sotto la direzione di Roberto Delle Donne, con l'edizione digitale della Cancelleria Angioina ora ospitata sull'istanza live e con integrazione di Linked Open Data. Dal 2018-2019, si è aggiunta una linea editoriale parallela su **fonti aragonesi, federiciane e rinascimentali**, in connessione con il programma di ricerca diretto presso l'Università della Basilicata da Fulvio Delle Donne e dai suoi collaboratori del CESURA (Centro Europeo di Studi su Umanesimo e Rinascimento Aragonese). Questa linea include le edizioni critiche degli *Alfonsi regis dicta aut facta memoratu digna* di Antonio Beccadelli, il commento di Enea Silvio Piccolomini all'opera medesima, l'*Itinerarium* di Federico II di Svevia, il *De rebus circa regni Siciliae curiam gestis* dello pseudo Ugo Falcando e un *Formulario* di Bartolomeo Miniatore, accanto a banche dati prosopografiche e inventariali come *ReDiAr — Reti Diplomatiche Aragonesi, Licterarum Partium 19* e *BiBas*.⁷

7 La linea editoriale aragonese, federiciana e rinascimentale è pubblicata nelle collane di FedoA – Federico II University Press e della Basilicata University Press,,

La stessa esperienza ha però messo allo scoperto i punti in cui la scommessa “tutto-in-XQuery” si rompeva, in quattro direzioni.

— **Ecosistema di librerie.** Tutto ciò che andava oltre la manipolazione TEI di base — ritaglio di immagini, OAuth, OAI-PMH, CrossRef, Zenodo, autenticazione moderna — significava o reimplementare le primitive in XQuery o appoggiarsi a servizi esterni con bridge fragili. La promessa “tutto in un solo linguaggio” valeva solo per il puro lavoro TEI.

— **Superficie di debug.** Le stack trace di eXist-db erano scarse e spesso indirizzate al posto sbagliato; il tuning delle prestazioni richiedeva la lettura degli internals del motore; il deploy su un server di produzione era un’avventura host per host.

— **Stagnazione del frontend.** Il frontend HTML5 con jQuery leggero invecchiava male; il resto del web si

in moltissimi casi in collaborazione con CESURA. Le edizioni citate includono lavori curati da Fulvio Delle Donne, Cristiano Amendola, Davide Morra, Biagio Nuciforo, Maria Giordano e Martina Pavoni, prodotti nell’ambito del progetto PRIN-PNRR 2022 *IMPERI SITUS — Imperial and Monarchical Power: Evolution of Regal Ideology in Southern Italy (XII–XV century)* e dell’ecosistema Tech3You. La raccolta di edizioni, banche dati e materiali di supporto è ospitata nell’istanza live all’indirizzo <https://www.detail-aracne.it/exist/apps/aracne/index.html>, che resta la facciata pubblica della piattaforma di prima generazione.

spostava su framework a componenti mentre Aracne restava su template scritti a mano.

— **Difficoltà di onboarding.** Il codice monolitico in xQuery e la nicchia tecnologica in cui si situa sono risultati un notevole collo di bottiglia nella possibilità di estendere la community dei developers di Aracne.

Con il 2024, la curva di frizione per ogni nuova feature era così ripida che un'aggiunta o un'integrazione sarebbe costata più che riprogettare la piattaforma da zero. La decisione di avviare Aracne2 non ha però dismesso Aracne v1: l'istanza live continua a ospitare le edizioni descritte sopra, con nuovi corpora che continuano ad accrescersi, e il codice sorgente è preservato come *reference implementation* su github.com/orazionelson/aracne — ancora utile come documentazione relativamente all'eventuale aspetto di un CMS eXist-db-native dall'inizio alla fine. Aracne2 non è dunque un sostituto operativo di Aracne v1; è un percorso in avanti che permette ai nuovi progetti editoriali di adottare una piattaforma non più costretta dai quattro vincoli elencati sopra.

2.4 Aracne2 (2026 →)

La terza iterazione mutua tre linee dalle due precedenti. **Mantenere eXist-db, ma solo come archivio**

XML. Aracne2 continua a usare eXist-db (al momento, la versione 6.4.1) per archiviare i documenti TEI nativamente, eseguire trasformazioni XQuery e servire la ricerca full-text, le operazioni in cui eXist-db è effettivamente più performante. Tutto il resto (autenticazione, controllo accessi, stato del workflow, registry dei plugin, impostazioni, audit, integrazione AI, API REST pubblica) è spostato su un backend Python 3.12 / FastAPI / PostgreSQL.

Separazione netta tra piattaforma e corpus.

MaRa era uno strumento di corpus che ha sviluppato una UI; Aracne era una piattaforma che ha sviluppato uno strato UI. Aracne2 consiste di **due strati di dati distinti by design**: PostgreSQL per lo stato di piattaforma (utenti, ruoli, sessioni, registry dei plugin, impostazioni, audit), eXist-db per lo stato dei documenti (XML TEI organizzato in database per collezione). I due non si contaminano mai: la piattaforma è portabile fra corpora; il corpus è portabile fra piattaforme. Un'istanza Aracne2 può essere ridisplegata su hardware diverso senza perdere traccia di chi modifica cosa; un corpus può essere estratto da un'istanza Aracne2 e collocato su un altro eXist-db senza perdere la propria forma nativa.

Modularità come concern primario. Ogni integrazione che in Aracne era un caso speciale codificato a mano, è in Aracne2 un plugin: dodici lookup di

authority, sei backend di deposito, provider AI, server MCP, feed per il viewer EVT. L'attivazione di un plugin monta a caldo le route del plugin senza riavvii del backend. Un piccolo insieme di tag di capacità (*inline_authority*, *collection_deposit*, *website_deposit*, *ui_descriptor*) permette ai plugin di auto-cablarsi nella SPA senza che nessuno tocchi il codice della SPA stessa: il frontend scopre cosa è disponibile e renderizza i controlli corrispondenti.

Condensando la decisione progettuale sotto le tre linee: *usare ogni motore per ciò in cui eccelle, mantenendo i punti di giunzione abbastanza puliti da permettere la sostituzione di qualsiasi singolo componente.*

3. Il flusso di lavoro editoriale

Aracne2 è prima di tutto uno strumento per chi cura un'edizione. Il flusso di lavoro editoriale che la piattaforma supporta è intenzionalmente vicino al modo in cui un editor descriverebbe il proprio lavoro se interrogato in forma colloquiale, piuttosto che al modo in cui una toolchain XML descriverebbe il concatenarsi delle proprie fasi.

Editing schema-aware. L'editor TEI carica l'ODD del progetto (oppure, di default, lo schema *TEI All P5* fornito a corredo). L'autocompletamento è quindi limitato agli elementi e agli attributi che lo schema effettivamente consente nella posizione in cui si trova il cursore. La validazione gira live nel pannello di editing: gli errori vengono segnalati in-line, con un link alla regola dello schema violata. L'editor non deve mai uscire dal documento per eseguire un validatore

separato; e vede il documento dichiarato non valido già nel corso della sessione stessa.

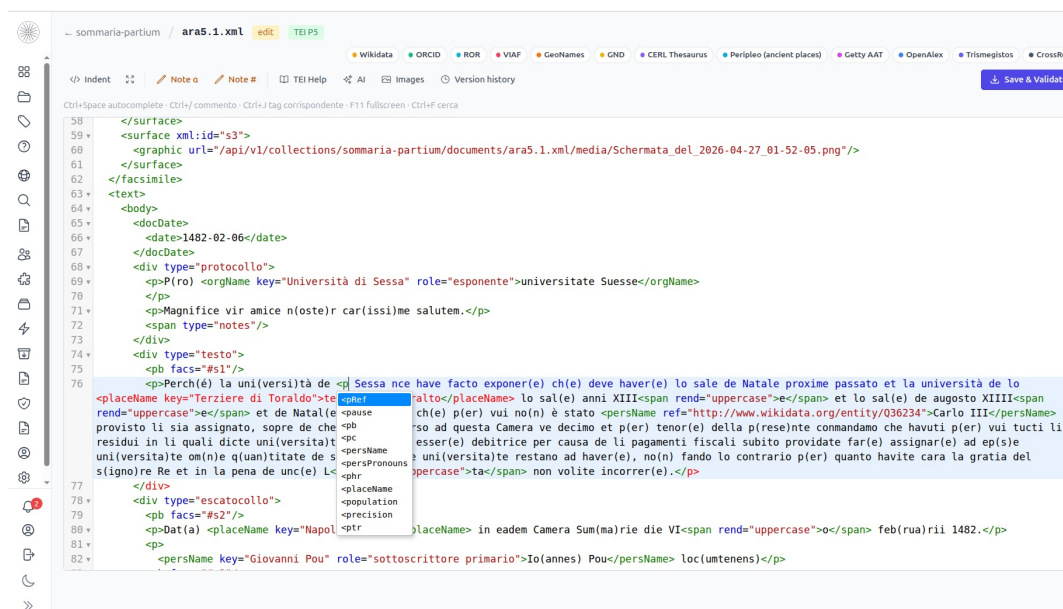


Figura 2: Il Tei Editor con l'autocompletamento basato sullo schema Tei predefinito.

Controllo di autorità. Sono inclusi come plugin nativi dodici lookup di authority: *Wikidata*, *ORCID*, *ROR*, *VIAF*, *GeoNames*, *GND*, *CERL Thesaurus*, *Periplo*, *Getty AAT*, *OpenAlex*, *Trismegistos*, *CrossRef*. Selezionando una porzione di testo all'interno di un elemento *persName*, *placeName*, *orgName*, *bibl* o di altro elemento di riferimento, si apre un dialogo di ricerca contro l'autorità scelta; accettando un risultato, l'URI canonico viene scritto nell'attributo *@ref*. L'editor scrive *prosa*; la piattaforma scrive *URI*.

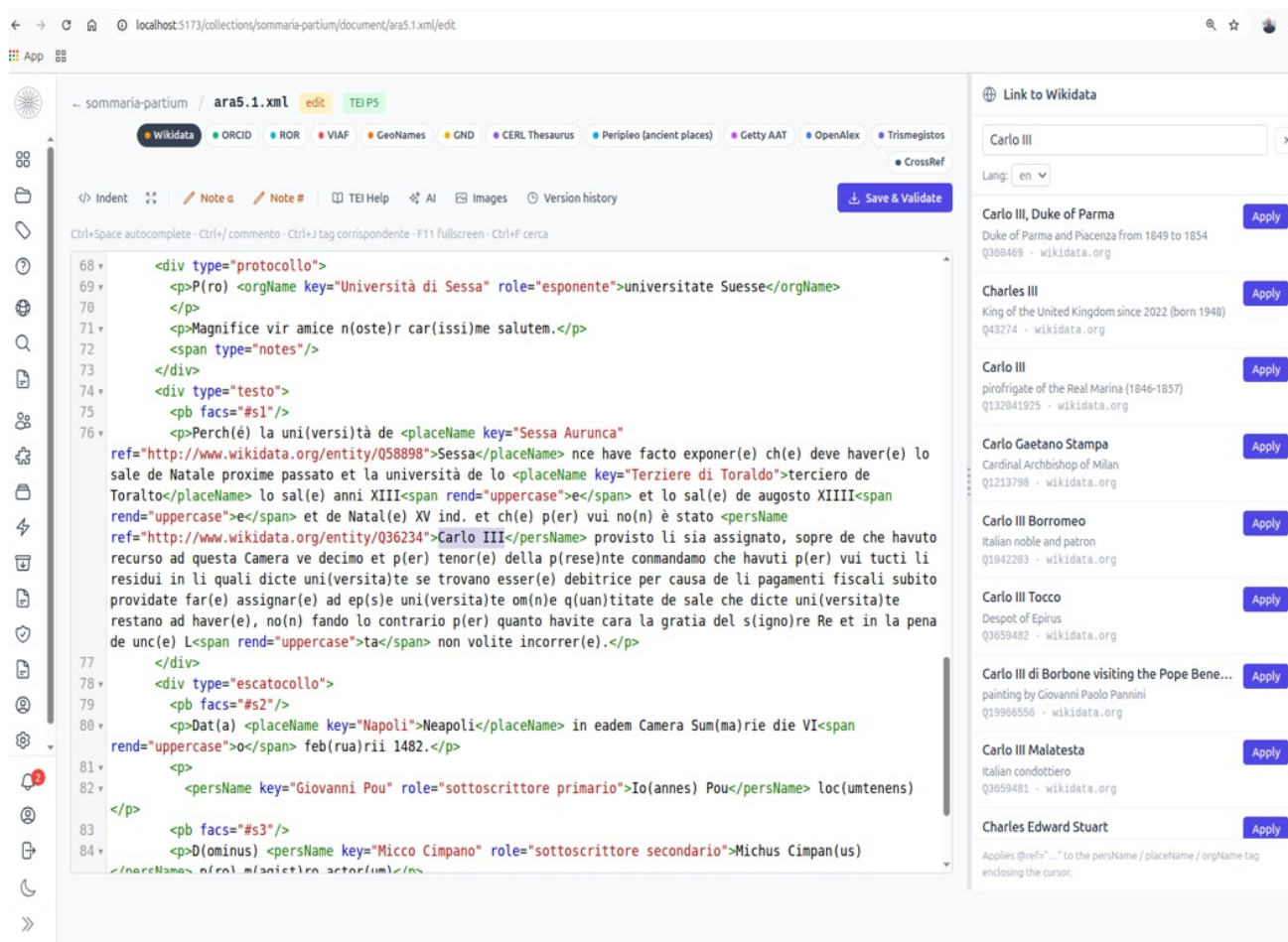


Figura 3: Nella parte alta dell'editor, i pulsanti per i servizi authority; a destra, il pannello di Wikidata.

Bibliografia. Un servizio *Bibliobuilder* ingerisce riferimenti forniti dagli autori — tipicamente un misto di forme estese, abbreviazioni e stili di citazione incoerenti — e ne propone la normalizzazione in voci *biblStruct*, con il supporto di un prompt AI dedicato al lavoro bibliografico. La risoluzione di DOI tramite CrossRef e l'importazione da gruppo Zotero sono percorsi di ingresso di prima classe; l'editor non passa mai per round-trip su CSV.

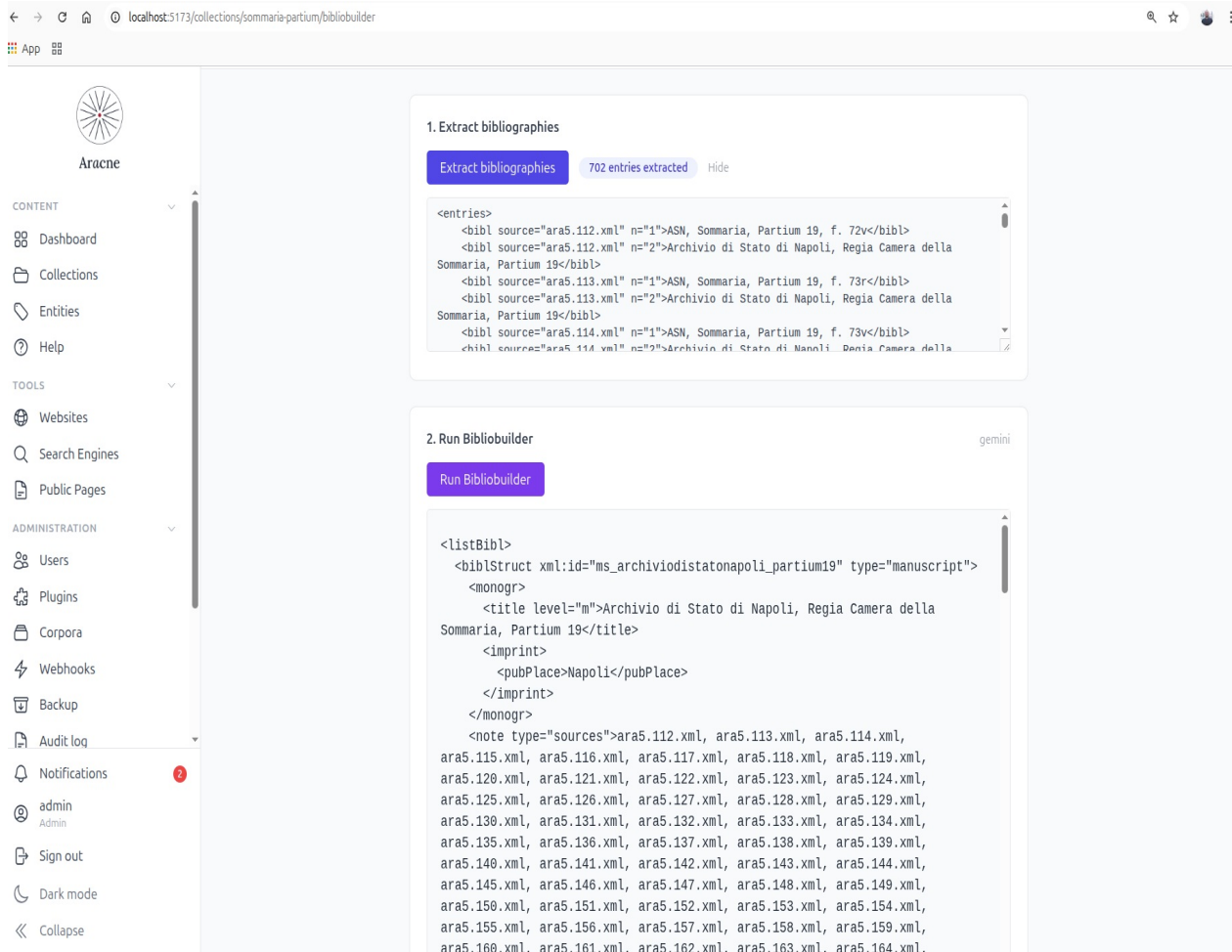


Figura 4: Il tool bibliobuilder

Facsimile. L'allineamento testo-immagine via TEI *zone* / *facs* è supportato attraverso un editor manuale: le zone vengono disegnate sul facsimile e legate alle porzioni corrispondenti di trascrizione.

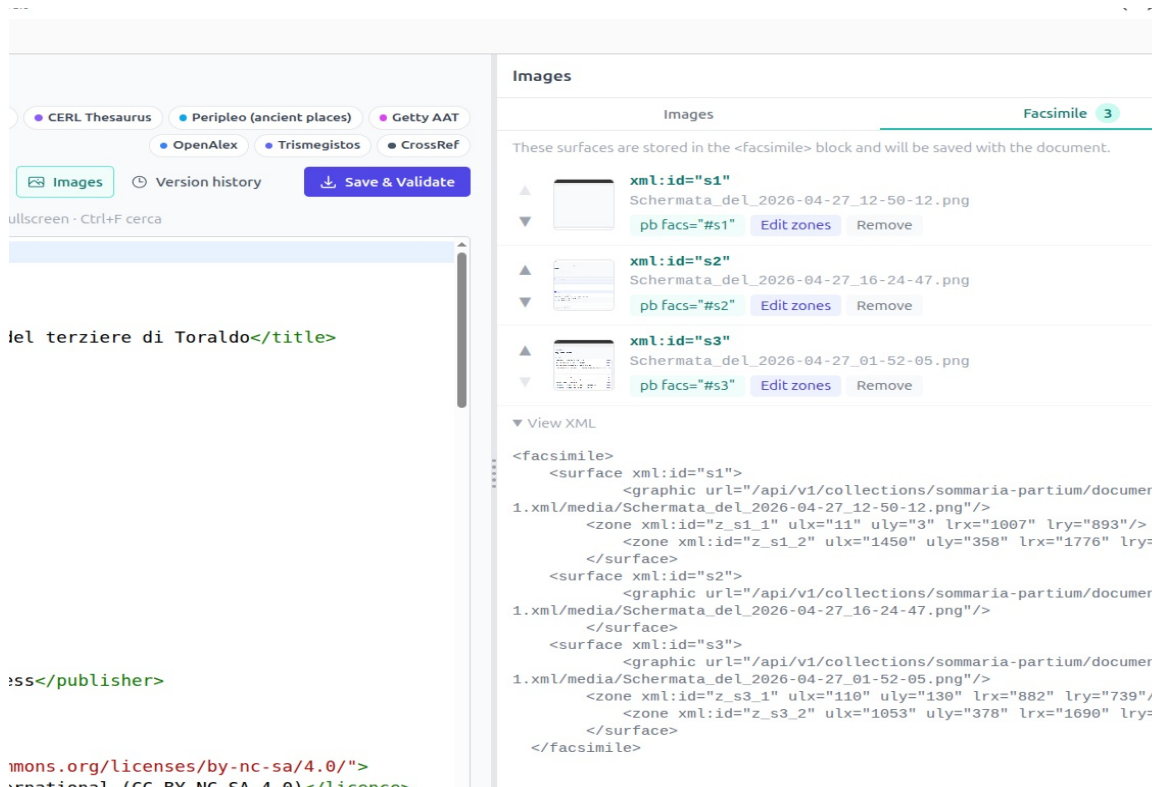


Figura 5: Associa immagine al testo come fac-simile e seleziona zone dell'immagine.

Stato del workflow. Ogni collezione si muove attraverso *bozza* → *revisione* → *pubblicazione*, con transizioni soggette al ruolo. I ruoli stessi separano gli ambiti laterali di editing e design: *Editor* e *Designer* stanno allo stesso livello, *EditorInChief* sta sopra entrambi, *Admin* sopra i ruoli editoriali, e *User* (sola lettura) sotto.

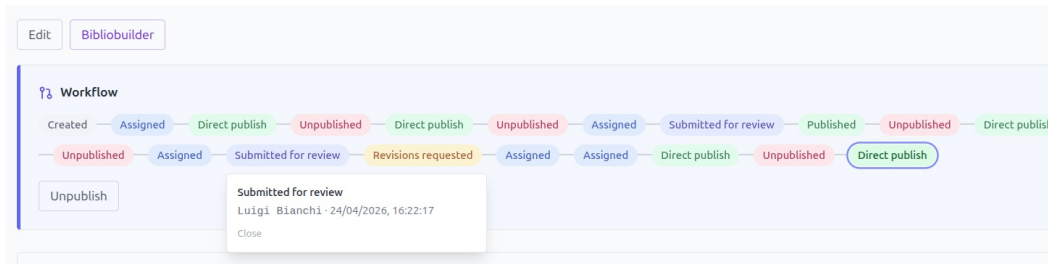


Figura 6: Il workflow editoriale visibile nel pannello di gestione della collezione.

La forma del workflow è prescrittiva solo dove deve esserlo: collezioni e stato bozza → revisione → pubblicazione sono nucleo. Tutto il resto — quali provider di authority sono attivi, quale provider AI è cablato, quali backend di deposito si attivano alla pubblicazione — è una scelta per deployment.

4. Il modello di pubblicazione

Un'edizione pubblicata è, in Aracne2, una funzione della collezione. Una volta che i documenti di una collezione hanno superato la revisione, la piattaforma può produrne tre tipi di artefatto: un **sito web**, un **motore di ricerca** e uno o più **record di deposito**.

I **siti web** (nel software, **websites**) possono essere statici (per essere ospitati su qualsiasi server HTTP, inclusi i servizi tipo Pages), dinamici (per ricerca full-text e trasformazione on-demand) o ibridi (statici per la navigazione, dinamici per la ricerca). Il sito è tematizzabile: il ruolo Designer gestisce un catalogo di fogli di stile XSLT e di temi CSS, mentre una piccola superficie Markdown permette al gruppo editoriale del progetto di scrivere i contenuti della home senza mettere mano ai template. L'integrazione con EVT 2 è disponibile come plugin opt-in per i progetti che desiderano un'interfaccia di lettura

ottimizzata per edizioni critiche e trascrizioni diplomatiche.

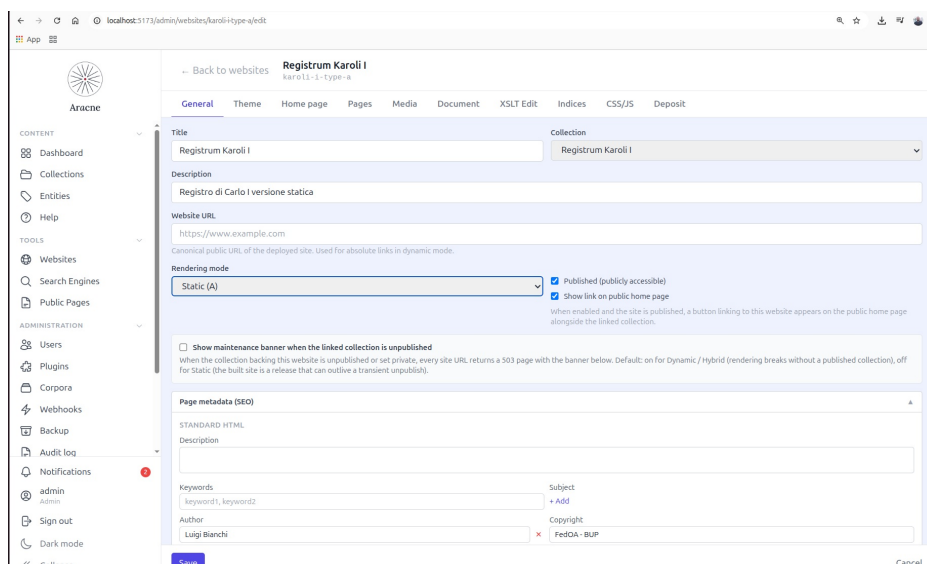


Figura 7: L'interfaccia di editing dei websites

I motori di ricerca (nel software, **search-engines**) sono configurabili per singola collezione pubblica o su un sottoinsieme di collezioni. Ogni motore espone tre superfici: una pagina HTML statica generabile, un'API REST pubblica e un widget JavaScript inseribile in qualsiasi sito esterno con uno snippet di tre righe, con controllo di accesso per origine. Un progetto che gestisce un portale museale o un blog scientifico può inserire il widget e ottenere ricercabilità sul proprio corpus senza dover trasferire il corpus stesso. È presente, inoltre un modulo che introduce nelle pagine pubbliche del sito una interfaccia di ricerca in lingua naturale.

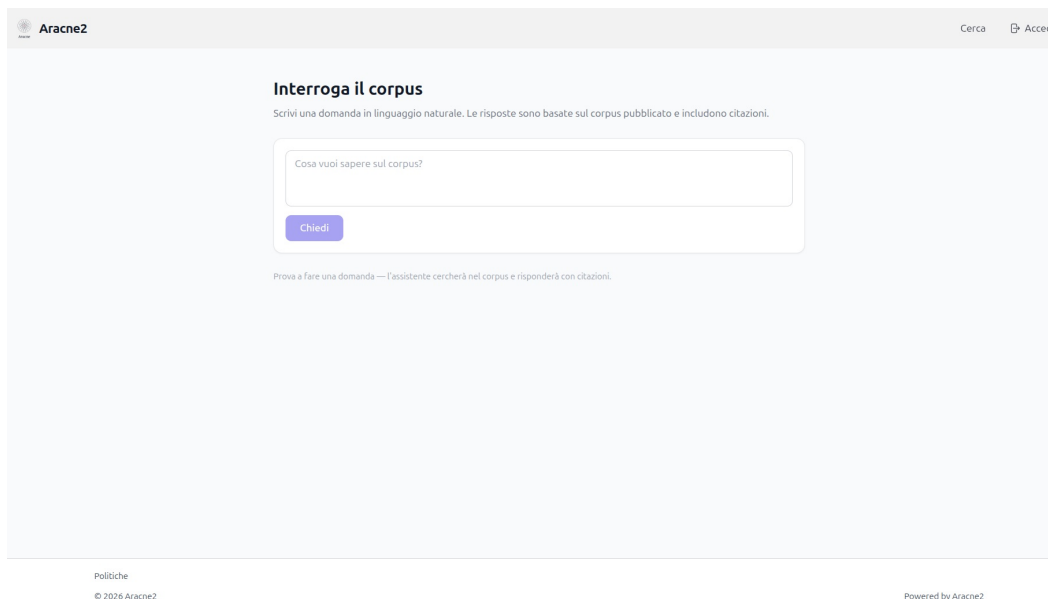


Figura 8: La pagina di ricerca in NL

I backend di deposito collegano la pubblicazione alla preservazione a lungo termine e alla citabilità. Sono inclusi sei backend: *Zenodo*, *Internet Archive*, *Codeberg*, *GitHub*, *GitLab*, *Dataaverse*. La pubblicazione di una collezione spinge automaticamente una copia di deposito su ognuno dei backend cablati, recupera l'identificatore persistente (DOI, ARK o URL del repository) e renderizza badge di frontend con quell'identificatore sul sito pubblico. Il pattern è *una collezione → molti depositi*, non *una collezione → un deposito*; un'edizione può insistere su Zenodo per il DOI, sull'Internet Archive per la preservazione a lungo termine e su GitHub per il versionamento collaborativo, senza scripting per deployment.

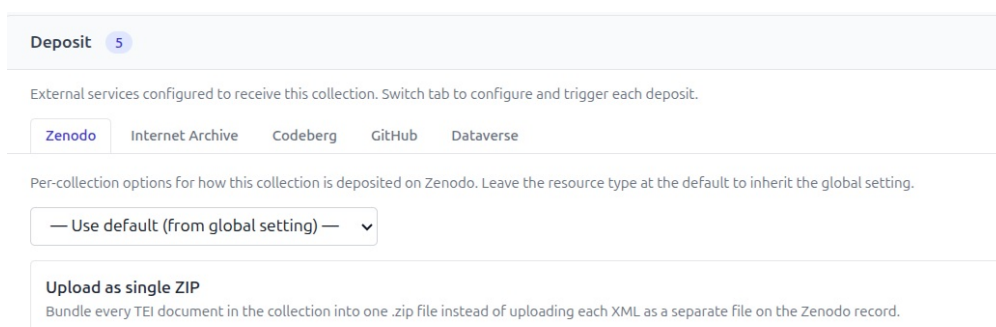


Figura 9: Il pannello di deposito delle collezioni

OAI-PMH 2.0 è esposto come endpoint provider, in modo che le collezioni pubblicate dalla piattaforma possano essere raccolte da aggregatori di metadati nazionali e internazionali (*BASE*, *OpenAIRE*, *Cultura Italia*, portali specifici di progetto) senza ulteriore intervento.

Linked open data. Le pagine pubbliche portano JSON-LD secondo *schema.org*; gli endpoint pubblici di collezione e di documento supportano la negoziazione del contenuto e servono RDF in Turtle, RDF/XML o JSON-LD su richiesta. Gli stessi identificativi Wikidata che l'editor ha selezionato per gli attributi *@ref*, si propagano, alla pubblicazione, nel grafo RDF pubblico; un consumer esterno può scoprire, dalla pagina pubblica, di che cosa parla l'edizione in termini machine-actionable.

5. L'AI come collaboratore non accessorio

Aracne2 non tratta l'assistenza AI come un widget di chat innestato su un workflow scollegato, ma come uno strato che conosce TEI, conosce lo schema del progetto, conosce il corpus dell'utente, e si innesta nei passaggi editoriali in cui fa effettivamente risparmiare tempo. L'integrazione corre lungo cinque assi; vale la pena descriverli uno per uno, perché le scelte progettuali che li sostengono sono ciò che distingue questa piattaforma da un CMS generico con un pulsante chatbot.

La scelta progettuale ha, va detto, un esplicito precedente teorico nella letteratura storiografica italiana. In un saggio del 2025 sull'editoria accademica, i dati della ricerca e l'intelligenza artificiale, **Roberto Delle Donne** ha sostenuto che i Large Language Models e la Retrieval-Augmented Genera-

tion appartengono al toolkit legittimo del lavoro editoriale accademico, a condizione che siano coerenti con i principi FAIR dell'accesso equo a dati e infrastrutture e della missione umanistica di preservazione del patrimonio culturale.⁸ L'architettura del § 5 di Aracne2 è, in questo senso, una risposta implementativa a una posizione metodologica già articolata dallo stesso milieu scientifico in cui la linea della piattaforma è nata.

5.1 Assistenza in editor

L'editor TEI integrato in Aracne2 mette a disposizione un pannello laterale AI a cui viene fornito il contesto del documento come parte di ogni turno. Fra i tipi di turno che il pannello gestisce:

— *Marca questo paragrafo come persName / placeName / orgName.* Il modello redige la marcatura; l'editor la rivede e la accetta o la modifica.

— *Perché questo non passa la validazione?* Il pannello legge gli errori strutturati del validatore e li spiega in prosa, con una proposta di correzione inline.

⁸ Roberto Delle Donne, "L'editoria accademica, i dati della ricerca e l'intelligenza artificiale", in *Life-on-life. L'umanesimo di fronte alle sfide dell'intelligenza artificiale*, a cura di M. Cosenza e L. Lo Sapio, Napoli, FedOAPress, 2025 (Human & Future, 1), pp. 31–53. ISBN 978-88-6887-355-4 — DOI: 10.6093/978-88-6887-355-4. Open Access: <https://fedoa.unina.it/15770/>.

La correzione è applicata come azione editoriale, non come risposta in testo libero che l'editor debba trascrivere.

— *Abbozza un template XSLT per il blocco msDesc.*
In modalità Designer, il pannello costruisce uno scaffolding di stylesheet che il Designer poi raffina.

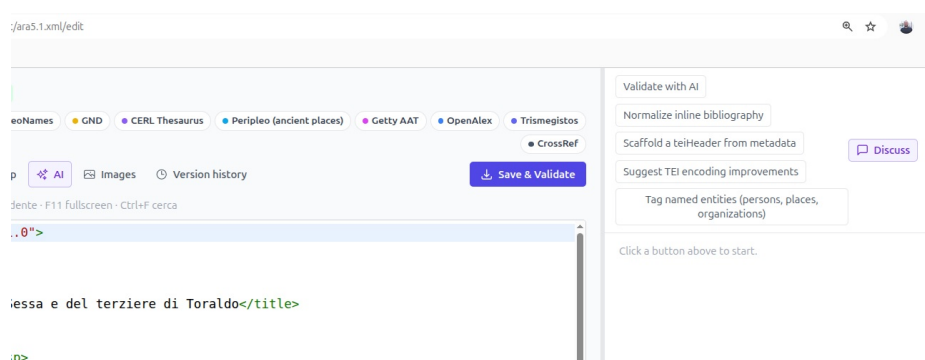


Figura 10: Il pannello AI nel TEI Editor.

La **libreria di prompt** è modificabile per deployment. Ogni prompt è scoped su una superficie (*editor TEI, bibliografia, debugger XSLT, navigatore di schema...*); il pulsante della toolbar corrispondente si auto-cabla nell'interfaccia in base allo scope del prompt, senza codice UI dedicato. Un progetto che necessiti di un prompt personalizzato — *riassumi le caratteristiche diplomatiche di questa carta, proponi note di datazione a partire dal formulario, segnala incipit formulari rispetto a una lista* — lo aggiunge alla libreria e il pulsante corrispondente compare.

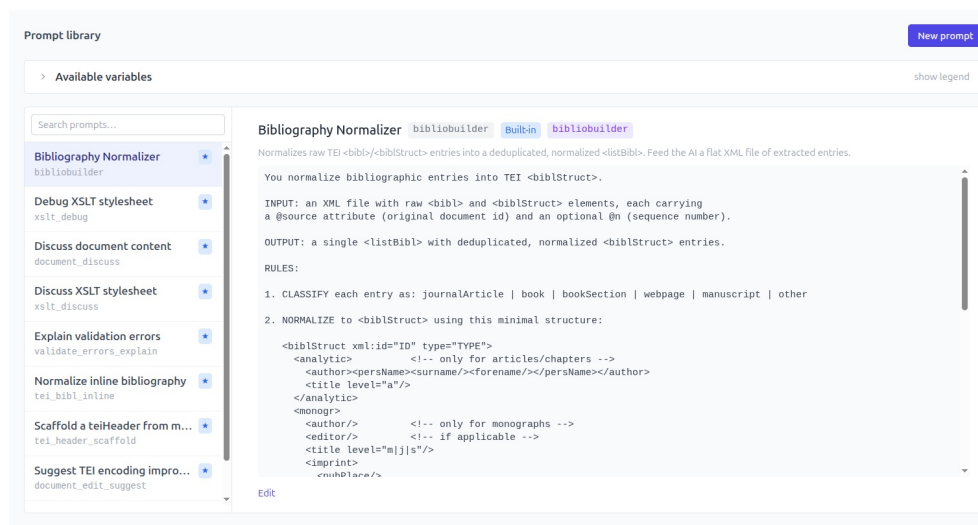


Figura 11: La libreria di prompt di default con la possibilità di aggiungere prompt personalizzati.

5.2 Automazione bibliografica

Bibliobuilder, già menzionato al § 3, è l'applicazione più matura dell'AI nel flusso editoriale: ingerisce riferimenti disordinati forniti dagli autori e li normalizza in *biblStruct*. La risoluzione di DOI via CrossRef è il percorso a basso costo per le voci che il modello riconosce; il contributo del modello è più utile dove l'input grezzo è troppo irregolare per la sola normalizzazione su pattern.

5.3 Integrazione di assistenti esterni via MCP

Aracne2 si espone come endpoint **Model Context Protocol**, così che un editor che lavora in Claude Desktop, Cursor, Claude Code o in qualsiasi

altro client MCP-aware, possa chiedere: *riassumi le tragedie in questo corpus o in quali documenti compare il placeName “Napoli”*, ottenendo risposte ancorate ai documenti TEI reali e non al pretraining del modello. I token sono scoped per **corpus**, così che domini eterogenei ospitati sulla stessa istanza non si contaminino mai a vicenda nelle analisi.

La rilevanza di MCP per il lavoro editoriale sta nel fatto che inverte la direzionalità abituale dell’integrazione. L’assistente desktop preferito dell’editor diventa il client; Aracne2 diventa il server che risponde con autorevolezza alle domande sul corpus. L’editor non deve abbandonare il proprio ambiente per fare una domanda al corpus, e l’assistente non deve allucinare per rispondere.

5.4 Scelta del provider

La piattaforma supporta quattro famiglie di provider AI: OpenAI, Anthropic Claude, Google Gemini e **Ollama** (locale). I provider cloud si configurano per deployment incollando una API key in *Settings* → *AI*; la chiave è cifrata a riposo. Il profilo Ollama è incluso nel file Docker Compose ed esegue il modello scelto sull’hardware del deployment, senza chiave necessaria e senza traffico che lasci l’host.

The screenshot shows a configuration panel with two sections. The first section, labeled 'ai_provider', contains a dropdown menu currently set to 'gemini'. Below the dropdown is a list of available providers: 'disabled', 'anthropic', 'openai', 'gemini' (which is highlighted with an orange background), and 'ollama'. To the right of the dropdown is a 'string' input field with 'Save' and 'Cancel' buttons. The second section, labeled 'ai_rag_context_tokens', contains an 'int' input field and an 'Edit' link.

F

Figura 12: Scelta dell'AI provider

5.5 Retrieval-augmented generation su TEI

La parte più distintiva dello strato AI — e quella che gli strumenti AI generici non possono fare — è la retrieval-augmented generation **ancorata alle Guidelines TEI e al corpus dell'utente.**

Un large-language model di frontiera è un generalista sicuro di sé. “Sa” che aspetto debba avere un *persName* con *@ref*, ma la sua conoscenza è la media di ogni snippet TEI visto durante il pretraining, inclusi quelli sbagliati. Quando un editor chiede *questo blocco msDesc segue le ultime convenzioni TEI P5?*, un modello vanilla risponde la sua media, non la risposta canonica.

Aracne2 chiude lo scarto con il grounding via retrieval, in tre passi.

— **Ingestione delle TEI P5 Guidelines:** la specifica *viva*, non la media congelata del pretraining, in un archivio *pgvector* locale con embeddings Ollama. La piattaforma fornisce lo script di ingestion: si punta lo script all'URL delle Guidelines pubblicate, si lascia girare, si torna a un corpus ricercabile delle regole canoniche di marcatura TEI. Le Guidelines possono essere reingerite ogniqualvolta il TEI Council pubblichi una revisione; gli editor lavorano sempre contro la specifica corrente.

— **Ingestione del proprio corpus accanto a quello:** collezioni pubblicate, manuale di codifica del progetto, regole interne di stile. La stessa istanza *pgvector* tiene insieme più sorgenti contemporaneamente; le query di retrieval pescano da ciò che è rilevante per il prompt.

— **Opt-in per prompt.** La libreria di prompt porta un toggle *rag_enabled* su ogni voce. Il prompt che spiega la validazione beneficia del retrieval (deve poter citare la sezione delle Guidelines che sta invocando); il prompt di sintesi casuale, no. L'editor non configura nulla turno per turno — i metadati del prompt decidono.

La pipeline RAG completa (PostgreSQL + *pgvector* + embeddings Ollama + generazione Ollama) gira nel profilo Docker Compose *ai-local* incluso.

Nessun dato lascia l'host. Per le istituzioni che non possono condividere la propria documentazione TEI con OpenAI o Anthropic per ragioni legali, di compliance o di sovranità, questo è il discrimine fra *l'AI è proibita dalle policy* e *l'AI fa parte del processo editoriale*.

La pipeline è inoltre **fail-soft**. Se pgvector è offline, il modello di embedding non è stato scaricato, o il RAG è semplicemente disattivato per un deployment, i prompt girano senza augmentation, con una piccola nota strutturata nella risposta che indica che manca il retrieval. L'editor non vede mai un crash: vede lo stesso prompt senza il paragrafo di retrieval.

La rivendicazione di posizionamento è la seguente: TEI è un vocabolario con oltre 1500 definizioni di elementi e decenni di guida comunitaria sedimentata, in larga parte in prosa lunga nelle Guidelines; gli strumenti chat generici non possono essere *puntati* contro quel corpo di sapere in modo principiato.

Un CMS TEI-native che fornisce il percorso di ingestion **come parte della piattaforma**, anziché chiedere a ogni progetto di ingegnerizzarlo localmente, allo stato attuale appare un caso unico.

6. Architettura, in sintesi

Aracne2 è implementato come sistema a tre tier. Una single-page application Vue 3, scritta in TypeScript e impaginata con Tailwind, comunica con un backend FastAPI esclusivamente attraverso un'API REST JSON autenticata da JWT a vita breve in memoria, e da un refresh token a vita lunga in cookie *httpOnly*. Il backend, a sua volta, parla con due engine di persistenza: un database PostgreSQL (con pgvector, quando è abilitato il RAG) per lo stato di piattaforma, e un server eXist-db 6.x per l'archiviazione dei documenti TEI. Il frontend non tocca mai direttamente nessuno dei due database.

Le giunzioni interne del backend sono intenzionali. Tutto l'XQuery è caricato da file *.xq* / *.xqm* su filesystem, mai composto inline in Python; il sistema dei plugin è hook-based, con i plugin che registrano listener su eventi nominati (*document.*

uploaded, collection.published...) anziché patchare il core; il rate limiting è applicato a livello di router, e il parsing XML passa sempre attraverso *defusedxml* per prevenire attacchi XXE. Nessuna di queste è una scelta inusuale in un backend web del 2026, ma insieme proteggono la proprietà che conta: ogni pezzo sostituibile dell'architettura resta tale.

La piattaforma viene rilasciata in Docker. Un comando *make up* avvia PostgreSQL, eXist-db, il backend FastAPI e il frontend servito da Vite; *make migrate & make seed* prepara la piattaforma al primo uso. Sono inclusi un *docker-compose.prod.yml* e una configurazione Nginx come reverse proxy.

La regola di progettazione che tiene insieme l'architettura è la stessa che ha guidato la progettazione di Aracne2 in astratto: *usare ogni motore per ciò in cui eccelle, mantenendo i punti di giunzione abbastanza puliti da permettere la sostituzione di qualunque dei singoli componenti.*

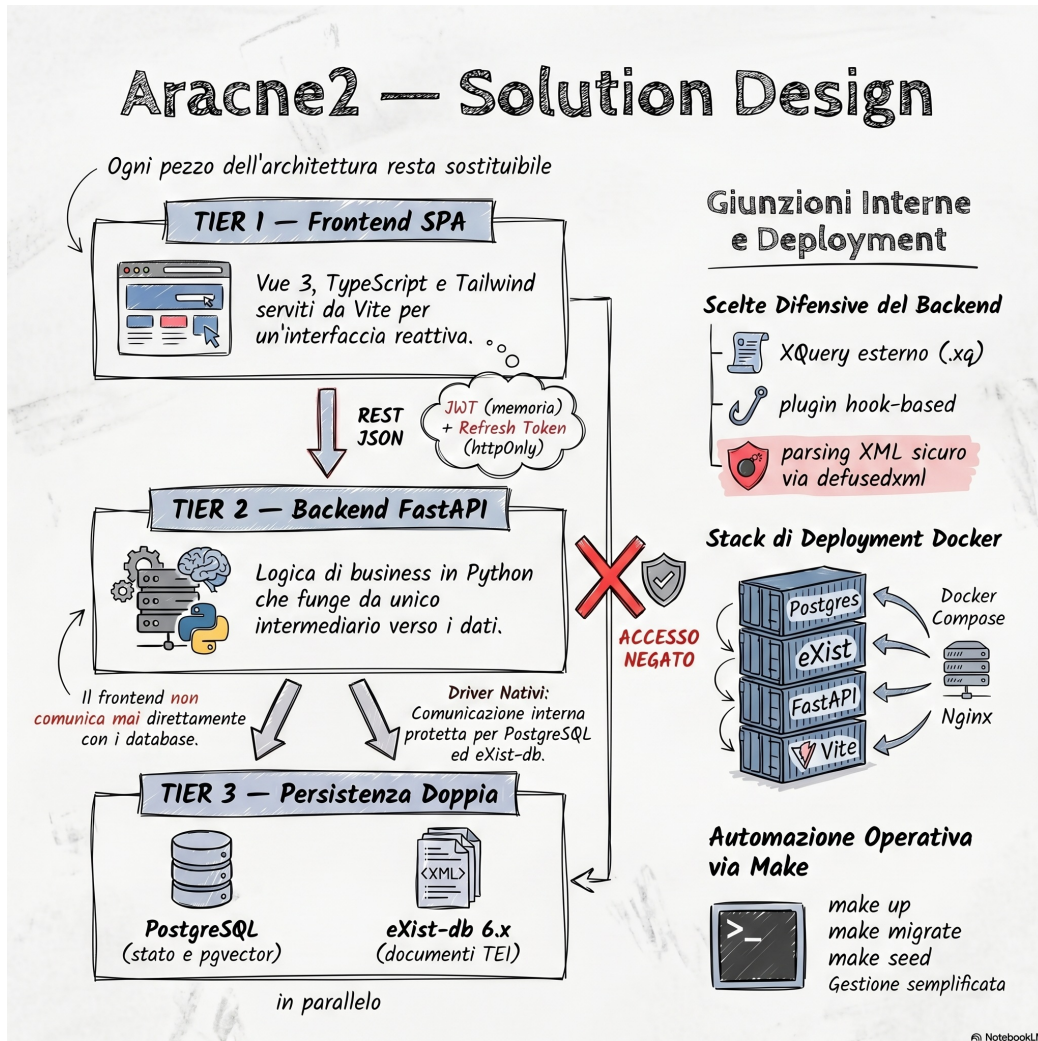


Figura 13: Solution design di Aracne2

7. Prospettive

Al momento in cui si scrive, Aracne2 è distribuito con una licenza di uso libero (MIT) ed è pubblicamente disponibile su GitHub all'url:

<https://github.com/orazionelson/aracne2>

Al di là del lavoro editoriale già in corso sulla piattaforma, ciò che ora serve è esperienza sul campo: gruppi di ricerca disposti a provare la piattaforma su un corpus reale e a riferire come si comporta nella loro infrastruttura, come gli strumenti AI funzionano effettivamente nella pratica editoriale quotidiana, come reagisce il sistema quando più editor lavorano simultaneamente sulla stessa collezione (un test di carico per cui l'architettura è stata progettata, ma a cui non è ancora stata sottoposta in produzione), e se la saturazione del garbage collector di eXist-db che riemergeva su Aracne v1 si presenta ancora sotto uso sostenuto. La piattaforma è aperta al contatto da parte

di qualsiasi gruppo voglia ospitarne un'istanza o contribuire upstream.

In parallelo, il lavoro attualmente in corso, in ordine approssimativo di priorità, è il seguente.

— **Approfondimento del corpus RAG.** Ingerire le TEI P5 Guidelines nella loro forma attualmente pubblicata è semplice; ingerire le *TEI Customisation Best Practices*, le *EpiDoc Guidelines* e i manuali di codifica specifici dei progetti è il passo successivo, con scelta per deployment di quali corpora siano attivi.

— **Ampliamento della superficie MCP.** L'endpoint MCP attuale espone strumenti di lettura sull'intero corpus; una superficie di scrittura che permetta a un assistente esterno di *proporre* una modifica (sempre rivista dall'editor prima del commit) è in roadmap.

— **Maturazione dell'ecosistema di plugin.** L'architettura supporta plugin out-of-tree, ma il canale di scoperta e revisione dei plugin di terze parti è al momento informale. Un catalogo firmato e revisionato è in fase di progettazione.

L'idea di fondo di Aracne2 non è cambiata dalla prima iterazione: **fornire una piattaforma completa e libera che copra l'editing, la cura e la pubblicazione dei corpora TEI dall'inizio alla fine**, perché progetti piccoli e medi possano produrre

edizioni scientifiche digitali senza dover assemblare una propria toolchain.

La novità di questa iterazione è che la piattaforma integra ora l'assistenza AI come parte dello stesso continuum, nella speranza di ridurre ulteriormente la distanza fra la competenza dell'editor e gli artefatti tecnici che quella distanza, fino a oggi, ha richiesto. Il lavoro di questi anni è stato il tentativo di rendere l'editing e la pubblicazione TEI accessibili allo storico, all'archivista, al filologo che non è anche programmatore; l'AI è, in questo senso, l'ultimo strumento di quella linea, non una deviazione da essa.

7.1 Una nota di metodo

Aracne2 è stato sviluppato con l'assistenza di un modello linguistico di ultima generazione (Anthropic Claude, accessibile attraverso un agente di programmazione integrato all'ambiente di sviluppo locale), secondo un modello di lavoro che merita di essere dichiarato esplicitamente in quanto coerente con la posizione metodologica articolata in questo contributo. L'integrazione dell'AI non è avvenuta come autocompletamento di superficie, ma come collaborazione architeturale: il modello ha avuto accesso integrale alla base di codice, alla documentazione di design, alle issue aperte e ai test del progetto, ed è stato coinvolto nelle decisioni di

struttura insieme al curatore umano. Il flusso di lavoro è stato iterativo e progettuale: definizione condivisa di un obiettivo (un nuovo plugin, una migrazione di schema, una superficie MCP); discussione in linguaggio naturale dei vincoli e delle alternative; stesura di una prima implementazione completa; revisione critica con scarto motivato delle proposte non coerenti con i vincoli architetturali del progetto, test, refactoring, commit.

La direzione del progetto, le decisioni di scope, l'architettura a due strati, la scelta dei provider AI integrati, la separazione fra plugin nativi (core) e plugin non-nativi (extensions), la disciplina sulla validazione XML e sulla sicurezza dei parser, sono **decisioni autoriali umane**. L'AI ha agito come collaboratore tecnico di rigore equivalente, sotto direzione autoriale, esattamente nello stesso senso in cui il § 5 propone che l'AI agisca per il filologo che lavora dentro Aracne2.

Bibliografia

Amendola, C. (2025). Digital TEI-XML edition of Camillo Porzio's "La congiura de' Baroni del Regno di Napoli contra re Ferdinando primo" [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.18047915>.

Amendola, C. (2023). Digital TEI-XML edition of "B. Miniature, Formulario di petitioni, responsioni e repplicationi per Astorre II Manfredi" (FM corpus) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.18047741>.

Amendola, C. (2021). Editoria universitaria, open access e nuove frontiere del lavoro umanistico: la Basilicata University Press e la collana "Digital Humanities". DigItalia, 16(2), 135–142. <https://doi.org/10.36181/digitalia-00042>.

Anderson, C.B. (2014) "On Teaching XQuery to Digital Humanists," Balisage Series on Markup Technologies. doi: <https://doi.org/10.4242/balisagevol13.anderson01>.

Bauman, S. (2019) "A TEI Customization for Writing TEI Customizations," Journal of the Text Encoding Initiative. UMR ESPACE et UMR LISST, (Issue 12). doi: <https://doi.org/10.4000/jtei.2573>.

Baillet, A. and Giovacchini, J. (2021) "TEI Models for the Publication of Social Sciences and Humanities Journals: Opportunities, Challenges, and First Steps Toward a Standardized Workflow," Journal of the Text Encoding

Initiative. UMR ESPACE et UMR LISST, (Issue 14). doi: <https://doi.org/10.4000/jtei.3419>.

Birnbaum, D.J. *et al.* (2019) “The Integration of XML Databases and Content Management Systems in Digital Editions,” *Balisage Series on Markup Technologies*. doi: <https://doi.org/10.4242/balisagevol23.birnbaum01>.

Brundage, M.P. (2004) “XQuery: The XML Query Language.” Available at: <http://ci.nii.ac.jp/ncid/BA68065930>.

Burghart, M. (2016) “The TEI Critical Apparatus Toolbox: Empowering Textual Scholars through Display, Control, and Comparison Features,” *Journal of the Text Encoding Initiative*. UMR ESPACE et UMR LISST, (Issue 10). doi: <https://doi.org/10.4000/jtei.1520>.

Cummings, J. (2013) “The Text Encoding Initiative and the Study of Literature,” pp. 451-476. doi: <https://doi.org/10.1002/9781405177504.ch25>.

Chiffolleau, F. (2021) “Keeping it Open: a TEI-based Publication Pipeline for Historical Documents.” Available at: <https://hal.science/hal-04357295>.

Cho, J. *et al.* (2024) “CAAP: Context-Aware Action Planning Prompting to Solve Computer Tasks with frontend UI Only,” *arXiv (Cornell University)*. Cornell University. doi: <https://doi.org/10.48550/arxiv.2406.06947>.

Cosco, A. (2019) “orazionelson/aracne: aracne1.1”. Zenodo. Doi: <https://doi.org/10.5281/zenodo.2647506>.

Cosco, A. (2018) “angioXML/MaRa2: Southern Italian Angevine Chancery Papers in XML, the Script MaRa2”. Zenodo. Doi: <https://doi.org/10.5281/zenodo.1447195>.

Delle Donne, F., Pavoni, M., Amendola, C. e Cosco, A. (2024) “Reti epistolari tra Medioevo e Rinascimento”, in De Blasi, M. (a cura di) *Moving Texts. Filologie e digitale*. Napoli: UniOr Press, pp. 95-138. doi: <https://doi.org/10.6093/978-88-6719-287-8>.

Delle Donne, F. (a cura di) (2024a) *Antonio Beccadelli (Panormita), Alfonsi regis dicta aut facta memoratu digna – I detti e i fatti memorabili di re*

Alfonso. Potenza-Napoli: BUP – Basilicata University Press / CESURA (BUP – Digital Humanities, ISSN 2724-2072).

Delle Donne, F. (a cura di) (2024b) *Il Commento di Enea Silvio Piccolomini agli Alfonsi regis dicta aut facta memoratu digna di Antonio Panormita*. Potenza–Napoli: BUP / CESURA. (PRIN-PNRR 2022 IMPERI SITUS).

Delle Donne, R. (2025) “L'editoria accademica, i dati della ricerca e l'intelligenza artificiale”, in Cosenza, M. e Lo Sapio, L. (a cura di) *Life-on-life. L'umanesimo di fronte alle sfide dell'intelligenza artificiale*. Napoli: FedOAPress (Human & Future, 1), pp. 31-53. DOI: 10.6093/978-88-6887-355-4. Disponibile all'indirizzo: <https://fedoa.unina.it/15770/>.

Delle Donne, R. (2019) «Information Technologies for Medieval Studies: Some Recent Experiences in Italy», *Reti Medievali Rivista*, 20(1), pagg. 13–39. doi: <https://doi.org/10.6092/1593-2214/6071>.

Delle Donne, R. (2018) “Crisi di legittimità nel Regno aragonese di Napoli: pratiche politiche e rappresentazioni culturali”, *Reti Medievali Rivista*, 19(2), pp. 237-245.

Delle Donne, R. (2014) “Un intreccio di iniziative scientifiche. Reti Medievali e il futuro della storiografia digitale”, *Reti Medievali Rivista*, 15(2), pp. 93-156. DOI: 10.6092/1593-2214/439.

Delle Donne, R. (2012) *Burocrazia e fisco a Napoli tra XV e XVI secolo. La Camera della Sommaria e il Repertorium alphabeticum solutionum fiscalium Regni Siciliae Cisfretanae*. Firenze: Reti Medievali e-book – Firenze University Press.

Delle Donne, R. (2005) “Le fonti digitali e le pratiche della ricerca. A proposito di un recente libro di Stefano Vitali”, *Reti Medievali Rivista*, 6(2), pp. 1-15.

Delle Donne, R. (2004) *La cancelleria angioina. Un sistema informativo digitale per la gestione e l'analisi della documentazione superstite*. Edizione a stampa per il deposito legale. Disponibile all'indirizzo: <https://hdl.handle.net/11588/115894>

Delle Donne, R. (a cura di) (2001-2008) *Il Regno angioino (secoli XIII-XV). Sistema informativo digitale per la gestione e l'analisi della documentazione superstite*. Napoli: Università degli Studi di Napoli “Federico II”.

Delle Donne, R. (1994) “Le cancellerie dell'Italia meridionale (secoli XIII-XV)”, *Ricerche storiche*, 24(2), pp. 361-388.

Fagervig, A. (2023) “Wikidata for authority control: sharing museum knowledge with the world”, *Digital Humanities in the Nordic and Baltic Countries Publications*, 5(1), pp. 222–239. [doi:10.5617/dhnbpub.10665](https://doi.org/10.5617/dhnbpub.10665).

Filangieri, R. (1944) “Report on the Destruction by the Germans, September 30, 1943, of the Depository of Priceless Historical Records of the Naples State Archives”. *The American Archivist*, vol. 7, n. 4 (ottobre 1944), pp. 252-255. DOI: <https://doi.org/10.17723/aarc.7.4.y8g06m75585jp8r1>.

Filangieri, R. (a cura di). *I Registri della Cancelleria Angioina, ricostruiti*. Napoli, Accademia Pontaniana, 1950 ss.

Gibbs, F. and Owens, T. (2012) “Building Better Digital Humanities Tools: Toward Broader Audiences and User-Centered Designs”, *Digital Humanities Quarterly*. Alliance of Digital Humanities Organizations, 6. doi: <https://doi.org/10.63744/my9nfvthga3w>.

Haaf, S., Geyken, A. and Wiegand, F. (2014) “The DTA ‘Base Format’: A TEI Subset for the Compilation of a Large Reference Corpus of Printed Text from Multiple Sources”, *Journal of the Text Encoding Initiative*. UMR ESPACE et UMR LISST, (Issue 8). doi: <https://doi.org/10.4000/jtei.1114>.

Holmes, M. (2021) “Getting Along with Relational Databases”, *Journal of the Text Encoding Initiative*. UMR ESPACE et UMR LISST, (Issue 14). doi: <https://doi.org/10.4000/jtei.3874>.

Klein, M. and Kyrios, A. (2013) “VIAFbot and the Integration of Library Data on Wikipedia”, *DOAJ* (DOAJ: Directory of Open Access Journals). Available at: <https://doaj.org/article/e0dccb1b74bf48c6b03847d4afcd3327>.

Meier, W. (2003) “eXist: An Open Source Native XML Database”, Lecture Notes in Computer Science, pp. 169-183. doi: https://doi.org/10.1007/3-540-36560-5_13.

Mojjada, H. (2025) “Protocol for Metadata Harvesting: The Role of OAI-PMH in Digital Resource Integration”, International Journal of Research and Innovation in Applied Science, X(VII), pp. 724-736. doi: <https://doi.org/10.51584/ijrias.2025.100700066>.

Piccari, I. (2026) “Collective Intelligence. AI for Expanding the Information Content Offered to Users: the SBN Sommerso Project”, JLIS.it, Edizioni Università di Macerata, 17(1), pp. 128-140. doi: <https://doi.org/10.36253/jlis.it-676>.

Pierazzo, E. (2014) “Digital Scholarly Editing: Theories, Models and Methods”, HAL (Le Centre pour la Communication Scientifique Directe). French National Centre for Scientific Research. Available at: <https://hal.univ-grenoble-alpes.fr/hal-01182162>.

Sula, C.A. and Hill, H. (2017) “The Early History of Digital Humanities”, DH. Available at: <https://dh2017.adho.org/abstracts/347/347.pdf>.

Tang, X. *et al.* (2025) “Dynamic Context-Aware Prompt Recommendation for Domain-Specific AI Applications,” arXiv (Cornell University). Cornell University. doi: <https://doi.org/10.48550/arxiv.2506.20815>.

Toscano, M., Cobo, M.J. and Liu, X. (2022) “Software Solutions for Web Information Systems in Digital Humanities: Review, Analysis and Comparative Study”, Profesional De La Informacion. Ediciones Profesionales de la Informacion SL. doi: <https://doi.org/10.3145/epi.2022.mar.11>.

Wicentowski, J. and Anderson, C.B. (2017) “XQuery for Digital Humanists”. Available at: <https://dh2017.adho.org/abstracts/661/661.pdf>.

Riferimenti tecnici

DETAIL — istanza live di Aracne. <https://www.detail-aracne.it/exist/apps/aracne/index.html>.

TEI Consortium. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. <https://tei-c.org/release/doc/tei-p5-doc/>

eXist-db. *eXist-db Open Source XML Database*. <https://exist-db.org/>.

Documentazione FastAPI. <https://fastapi.tiangolo.com/>.

Documentazione Vue.js. <https://vuejs.org/>.

PostgreSQL. <https://www.postgresql.org/>.

pgvector: <https://github.com/pgvector/pgvector>.

Anthropic. *Model Context Protocol*. <https://modelcontextprotocol.io/>.

Ollama. <https://ollama.com/>.

EVT — Edition Visualization Technology. <https://evt.labcd.unipi.it/>.

Aracne (reference implementation, archiviata).
<https://github.com/orazionelson/aracne>.

Aracne2 — repository sorgenti. <https://github.com/orazionelson/aracne2>.

Aracne2 è un CMS modulare per l'edizione e la pubblicazione di corpora scientifici codificati in TEI. Pensato per filologi, storici, archivisti ed editori digitali, consente di trasformare trascrizioni dattiloscritte in edizioni digitali citabili, interrogabili e machine-actionable, senza richiedere competenze sistemiche avanzate. Il progetto nasce da una lunga esperienza di strumenti per l'edizione digitale: da MaRa, sviluppato a partire dal 2008 per la marcatura automatica dei Registri della Cancelleria Angioina, ad Aracne, piattaforma basata su eXist-db e XQuery, utilizzata tra il 2016 e il 2024 per diverse edizioni accademiche.

Aracne2 conserva eXist-db come archivio nativo dei documenti TEI, ma affida autenticazione, workflow, plugin e integrazioni a un backend Python/FastAPI/PostgreSQL, con interfaccia Vue 3. La piattaforma si distingue per la separazione tra stato del corpus e stato del sistema, per un'architettura a plugin estensibile e per l'integrazione di strumenti di assistenza AI. Provider diversi — OpenAI, Anthropic, Gemini, Ollama in locale — possono essere configurati entro una prompt library personalizzabile, con funzioni di retrieval-augmented generation ancorate alle TEI P5 Guidelines e al corpus dell'utente.

Il volume illustra il modello editoriale e tecnico di Aracne2, ne ricostruisce la genesi e riflette sull'uso dell'intelligenza artificiale come strumento editoriale integrato, non come semplice appendice conversazionale.

Alfredo Cosco unisce una formazione storica, maturata all'Università degli Studi di Napoli Federico II con una tesi in Digital Humanities, a competenze avanzate in ambito informatico. Dopo una lunga esperienza nei servizi bibliotecari e digitali della Federico II, ha lavorato con 4Science su repository, sistemi informativi ed editoria digitale scientifica. Attualmente è istruttore informatico presso il Comune di Parma per attività collegate al progetto InSTARSmartCity.