

# ConfRadar Implementation Plan

## Milestone 1: Literature Review & Design (Month 1-2)

- **Conduct literature review and requirements gathering**
- **Dependencies:** None.
- **Effort:** S (~5 hours).
- **Acceptance Criteria:** Summary document of related systems and data sources completed; initial list of candidate conference data sources compiled; high-level system requirements defined and reviewed.
- **Design system architecture and data schema**
- **Dependencies:** Literature review completed; requirements defined.
- **Effort:** M (~2 days).
- **Acceptance Criteria:** Architecture diagram created (showing modules for retrieval, extraction, storage, etc.); conference data schema finalized (fields like name, dates, location, etc.) <sup>1</sup> ; design walkthrough approved by team.
- **Define evaluation metrics and success criteria**
- **Dependencies:** System design completed.
- **Effort:** S (~4 hours).
- **Acceptance Criteria:** Documented list of MVP success metrics (e.g., extraction accuracy, recall of conferences, change detection speed); baseline targets set for each metric; team alignment on what constitutes MVP success.

## Milestone 2-3: Data Source Integration & Web Crawling

### Web Retrieval

- **Finalize list of core conference data sources (MVP)**
- **Dependencies:** Requirements/design phase completed.
- **Effort:** S (~2 hours).
- **Acceptance Criteria:** Confirmed list of sources to crawl is documented (including AI Deadlines, ACL events, ELRA events, ChairingTool, etc.); sources are accessible for testing; list approved by stakeholders.
- **Implement seeded web crawler for known sites**

- **Dependencies:** Data schema ready; seed URL list from core sources finalized.
  - **Effort:** M (~1 week).
- **Acceptance Criteria:** Crawler script/agent that periodically fetches pages from the seed list of conference websites <sup>2</sup>; successfully retrieves content (HTML or PDF) from at least 5 known conference sites; handles politeness (respects robots.txt, rate limiting) and logs fetch status; retrieved content is saved for extraction.
- **Implement search-based CFP discovery**
- **Dependencies:** Web crawler in place; search API access configured (if needed).
  - **Effort:** M (~3 days).
- **Acceptance Criteria:** Agent can perform targeted web searches (e.g. via API) for new conference calls not in the seed list <sup>3</sup>; identifies at least 3 new conference CFP pages (e.g., via keywords like "NLP conference 2025 deadline"); automatically fetches those pages; new conferences are added to the system with no manual input.
- **Develop scraper for AI Deadlines aggregator (NLP conferences)**
- **Dependencies:** Web crawling framework ready.
  - **Effort:** S (~1 day).
- **Acceptance Criteria:** Able to retrieve all listed NLP conference deadlines from the AI Deadlines source (e.g., Papers With Code's **AI Deadlines** site); parses each listed conference's name, dates, and deadlines; information is stored in the knowledge base.
- **Develop scraper for ACL Sponsored Events listing**
- **Dependencies:** Web crawling framework ready.
  - **Effort:** S (~1 day).
- **Acceptance Criteria:** The ACL events page is parsed successfully; for each listed event (conference or workshop), the name, date/location (if available), and link are extracted; at least 5 events are captured from the page; details for each are saved to the knowledge base.
- **Develop scraper or integration for ChairingTool platform conferences**
- **Dependencies:** None (can be done in parallel with other source integrations).
  - **Effort:** M (~2 days).
- **Acceptance Criteria:** Conference information is retrieved from ChairingTool (via available API or by scraping known conference URLs on the platform); at least one test conference's important dates are fetched (submission deadline, etc.); data is correctly parsed and stored in the knowledge base.
- **Develop scraper for ELRA conference listings (e.g., LREC)**
- **Dependencies:** None.

- **Effort:** S (~1 day).
- **Acceptance Criteria:** The ELRA website or event listings are parsed for upcoming conferences (such as LREC); relevant details (conference dates and any submission deadlines available) are extracted for at least one event; the information is saved to the knowledge base.

## Knowledge Base Setup

- **Set up conference knowledge base (database)**
- **Dependencies:** Data schema finalized.
- **Effort:** S (~1 day).
- **Acceptance Criteria:** Database instance initialized (SQL or graph DB as decided) with tables/collections for conference events and series <sup>1</sup>; schema fields match the defined data model (e.g., name, acronym, deadlines, etc.); a test record can be inserted and retrieved successfully.
- **Implement data ingestion pipeline to DB**
- **Dependencies:** Database setup; initial scraper outputs available.
- **Effort:** M (~2 days).
- **Acceptance Criteria:** Code module developed for taking raw extracted data and inserting or updating records in the knowledge base; ensures no duplicate entries for the same conference event (checks by conference series/year, etc.); each record is timestamped for last update; test by ingesting sample data (from crawled pages) and verifying records in DB.

## Milestone 3–4: Information Extraction Pipeline

- **Implement rule-based parsing for important dates**
- **Dependencies:** Sample conference pages fetched and available for parsing.
- **Effort:** M (~3 days).
- **Acceptance Criteria:** Parser functions identify structured "Important Dates" sections in HTML pages <sup>4</sup>; key dates (submission deadline, abstract deadline, notification date, conference start/end) are correctly extracted for at least 3 different conference pages with known date formats; extracted data matches the source content exactly.
- **Integrate LLM-based extraction for unstructured text**
- **Dependencies:** Basic rule-based parser in place; access to LLM API.
- **Effort:** M (~3 days).
- **Acceptance Criteria:** An LLM-powered extraction component is implemented (e.g., prompt the LLM to output a JSON of fields given the page text) <sup>5</sup>; for at least 2 pages where rules fail (e.g., narrative CFP announcements), the LLM successfully pulls out the required fields (conference name, deadlines, location, etc.) in the correct format; output validated against manual extraction for accuracy.
- **Combine rule-based and LLM extraction**
- **Dependencies:** Rule-based and LLM extraction methods completed.

- **Effort:** S (~1 day).
- **Acceptance Criteria:** A unified pipeline is created where the system applies rule-based parsing first and falls back to LLM extraction for fields not found by rules; pipeline returns a complete structured record for a given raw page input <sup>6</sup>; tested on a sample input where some fields are filled by rules and others by the LLM (ensuring no field is missing in the final output).
- **Validate extraction accuracy on sample data**
- **Dependencies:** End-to-end extraction pipeline ready; sample conference pages available.
- **Effort:** S (~1 day).
- **Acceptance Criteria:** Run the extraction pipeline on a set of ~5 known conference pages; compare each extracted field to ground truth from those pages; report compiled showing the extraction accuracy (e.g., number of correct vs. missed dates, correct names, etc.); if accuracy is below acceptable threshold, identify the errors for further improvement in parsing.

## Milestone 4–5: Clustering & Alias Resolution

- **Create alias dictionary for conference series**
- **Dependencies:** Initial conference data collected (names and acronyms).
- **Effort:** S (~1 day).
- **Acceptance Criteria:** A list or map of known aliases compiled (e.g., "NeurIPS" ↔ "NIPS", "IJCAI" ↔ full name) <sup>7</sup>; at least 10 major conference series and their common synonyms/abbreviations are included; this dictionary is accessible by the system for use in resolution logic.
- **Implement conference series clustering logic**
- **Dependencies:** Knowledge base populated with multiple conference entries; alias dictionary available.
- **Effort:** M (~3 days).
- **Acceptance Criteria:** Algorithm developed to group conference events by series identity (using title similarity, acronyms, year) <sup>7</sup>; each event is assigned a series ID or linked to a series entity in the DB; events with matching series or known aliases are clustered together; verify on examples (e.g., "IJCAI 2025" and "International Joint Conference on AI 2025" share a series ID, avoiding duplicate listings).
- **Link workshops to parent conferences**
- **Dependencies:** Clustering logic in place.
- **Effort:** S (~1 day).
- **Acceptance Criteria:** Workshops are identified (e.g., by category or name patterns) and their `parent_event` field is set to the main conference event in the knowledge base <sup>8</sup>; for a test case (e.g., a workshop at ACL 2025), the system correctly links it to the ACL 2025 event; parent-child relationship stored and can be queried.

- **Test alias resolution & clustering on sample data**

- **Dependencies:** Alias resolution and clustering implemented.
- **Effort:** S (~1 day).
- **Acceptance Criteria:** Use a sample dataset of known conferences (including some with aliases and workshop relationships) to evaluate clustering; confirm that no duplicate conference series appear in the final output (all aliases unified) <sup>9</sup>; measure basic clustering metrics (e.g., manual inspection for any mis-grouped events) and document results.

## Milestone 5–6: Agent Integration & Orchestration

- **Integrate all modules into ConfRadar agent**

- **Dependencies:** Retrieval, extraction, storage, and clustering modules all functional.
- **Effort:** M (~3 days).
- **Acceptance Criteria:** A central orchestrator (agent controller) is implemented to coordinate the pipeline <sup>10</sup>; the agent sequentially executes: crawl new data -> extract fields -> update knowledge base -> run clustering/alias resolution -> trigger output update; confirmed by running a one-click/one-command update that goes through all steps and produces an updated conference list.

- **Set up scheduling and monitoring (Airflow/Dagster)**

- **Dependencies:** Agent integration complete; decision made on Airflow vs. Dagster for orchestration.
- **Effort:** M (~2 days).
- **Acceptance Criteria:** Workflow orchestration in place using Apache Airflow or Dagster for periodic execution and monitoring of ConfRadar tasks; the pipeline is configured as a DAG/pipeline with tasks for each component (crawl, extract, store, etc.) <sup>11</sup>; scheduling is enabled (e.g., a daily run) and verified by observing automated runs in the orchestrator's UI; any failures are logged and visible for debugging.

- **Implement change detection and versioning**

- **Dependencies:** Knowledge base with initial data; ability to re-run extraction for existing entries.
- **Effort:** M (~2 days).
- **Acceptance Criteria:** On repeated runs, the system compares newly extracted data with the existing record in the DB <sup>12</sup>; if any field has changed (e.g., a deadline date), the change is captured: the old value is preserved or logged alongside the new value <sup>13</sup>; the conference record is updated to the latest info; testing confirms detection by simulating a known deadline change and seeing the change flagged in logs or data.

- **Provide basic update notifications**

- **Dependencies:** Change detection implemented.
- **Effort:** S (~1 day).
- **Acceptance Criteria:** When a change is detected for a conference, the system reflects this in the user-facing output or logs (for MVP, this could mean adding an "*updated*" note next to the

conference entry in the output document) <sup>14</sup> ; ensure that a user reviewing the output can identify what changed (e.g., “deadline extended by 7 days” noted in text or change log).

## Milestone 6: Evaluation & Iteration (Month 6)

- **Evaluate end-to-end system performance**
- **Dependencies:** Full MVP pipeline implemented and running.
- **Effort:** M (~3 days).
- **Acceptance Criteria:** Define a test set of conferences (e.g., 10 upcoming events with known ground-truth data); run ConfRadar end-to-end on this set; measure extraction accuracy (compare extracted dates to official info), recall (did we find all 10 events?), and change detection correctness; compile an evaluation report with metrics and findings for each core feature.
- **Gather user feedback and review results**
- **Dependencies:** System deployed in test mode; sample output available (e.g., on Google Doc/Notion).
- **Effort:** S (~1 day).
- **Acceptance Criteria:** Present the MVP output (conference list document) to a small group of target users or stakeholders; collect feedback on data accuracy, completeness, and usefulness; document any suggestions or discovered issues (e.g., missing conferences, incorrect dates); feedback is reviewed and prioritized for action.
- **Refine extraction and alias rules based on feedback**
- **Dependencies:** Evaluation and user feedback completed.
- **Effort:** M (~2 days).
- **Acceptance Criteria:** Improvements are made to address identified issues – e.g., update parsing rules or LLM prompts for cases where data was missed or mis-extracted, expand the alias dictionary for any new variants encountered; verify that after refinements, running the pipeline on the test set shows improved accuracy (e.g., previously missed dates are now captured, duplicate entries resolved).
- **Finalize MVP implementation**
- **Dependencies:** All critical fixes from evaluation are done.
- **Effort:** S (~4 hours).
- **Acceptance Criteria:** All core MVP features (automated retrieval, NLP extraction, knowledge base, alias clustering, change detection, basic output) are verified in an integrated test run with no critical bugs; code is cleaned up for stability; the team signs off that the MVP meets the requirements and is ready for deployment.

## Milestone 7: Deployment & User-Facing Integration (Month 7)

- **Deploy ConfRadar system to production environment**
- **Dependencies:** MVP fully tested; orchestration in place.

- **Effort:** M (~2 days).
- **Acceptance Criteria:** The ConfRadar agent is deployed on a server or cloud platform for continuous operation <sup>15</sup>; scheduling (from Airflow/Dagster) is running in the production environment; environment variables/credentials (for any APIs, databases, or output integration) are securely configured; a full pipeline run executes successfully in the production setting.
- **Integrate with Google Docs/Notion for output**
- **Dependencies:** Stable data pipeline; decision on output format.
- **Effort:** M (~2 days).
- **Acceptance Criteria:** A basic user-facing output is implemented by syncing data to a document or page <sup>16</sup> – e.g., the system updates a Google Sheet or Notion page with a table of upcoming conferences and deadlines; after each scheduled run, the output document reflects the latest data (new conferences added, changed deadlines highlighted); manual verification that the document is correctly updated after an agent run.
- **Ensure error handling and monitoring in prod**
- **Dependencies:** Deployment and output integration complete.
- **Effort:** S (~1 day).
- **Acceptance Criteria:** Logging and monitoring are active – the system logs key actions and errors at each step; any failures in the pipeline trigger alerts (e.g., email or Slack notification, or at least visible in Airflow/Dagster UI); a test failure (simulated by, say, disconnecting the network for a task) is caught and an alert is received by the team, ensuring issues will be noticed in production.
- **Complete documentation and project wrap-up**
- **Dependencies:** All implementation tasks finished.
- **Effort:** S (~1–2 days).
- **Acceptance Criteria:** Codebase is fully documented (README and inline comments for how to run, configure, and extend the system); a short user guide is written explaining how to view the conference deadline document and interpret updates; final project report (if required) is drafted, and knowledge transfer session held with relevant stakeholders or maintainers.

## Future Enhancements (Post-MVP)

- **Full-fledged UI & notifications** – Build a dedicated web application for ConfRadar with user accounts, dashboards, and alerts (beyond MVP scope) <sup>17</sup>.
- **Dependencies:** ConfRadar MVP backend stable and delivering accurate data.
- **Effort:** L (several weeks of work).
- **Acceptance Criteria:** A user-friendly interface where users can log in to view and search conference deadlines; ability for users to configure email or push notifications for new conferences or deadline changes; live updating dashboard or calendar; this goes live to a test user group with positive feedback.

- **Recommendation & analytics engine** – Provide personalized conference recommendations and deadline analytics (trends, stats) <sup>18</sup> .
- **Dependencies:** Sufficient conference data accumulated; user preference data (optional).
- **Effort:** M (~1–2 weeks).
- **Acceptance Criteria:** The system can suggest conferences to a user based on their field/interests (e.g., using keywords or past selections); basic analytics such as number of conferences per month or average deadline lengths are available; accuracy of recommendations is validated with users (e.g., >70% of suggestions seen as relevant in testing).
- **Multi-domain and multi-language support** – Extend ConfRadar to track conferences in other research fields and languages <sup>19</sup> .
- **Dependencies:** Core pipeline generalized (not hard-coded to CS/AI terms).
- **Effort:** M (1 week per new domain or language, roughly).
- **Acceptance Criteria:** Ability to easily plug in a new domain (e.g., Medical conferences) by providing new seed sources; the system successfully tracks a sample of conferences in the new domain; for multi-language, the extraction pipeline can handle at least one non-English CFP (with appropriate adjustments or translation step), confirmed by accurately parsing an example in that language.
- **Robust PDF and email integration** – Add support for parsing PDF CFPs and monitoring mailing list announcements <sup>20</sup> .
- **Dependencies:** MVP handles HTML reliably; PDF parsing tools available.
- **Effort:** M (~1–2 weeks).
- **Acceptance Criteria:** PDF parsing module integrated (e.g., using OCR or PDF-text extraction) so that if a conference CFP is only a PDF, the system can extract dates from it (at least for standard formatted PDFs); integration with mailing list archives or an email inbox to catch announcements not posted on websites; tested by successfully extracting data from at least one PDF-only CFP and one mailing list email announcement.
- **Scaling & performance improvements** – Optimize the pipeline for hundreds of sources and frequent updates.
- **Dependencies:** Increased load or planned scale-out.
- **Effort:** M (~1 week for significant optimizations).
- **Acceptance Criteria:** Refactored crawling to be parallel or use asynchronous requests to handle more sites; database indexing and query optimization for faster change detection; system verified to handle ~10x more conferences than MVP (e.g., simulating a full global list) without major slowdowns or crashes; resource usage remains within acceptable limits under heavy load.
- **Continuous NLP model upgrades** – Enhance extraction with improved models or prompts as they become available.



- **Dependencies:** New NLP models or updated LLM APIs accessible.
- **Effort:** M (few days for integration and testing).
- **Acceptance Criteria:** The extraction component is updated to use a more advanced model or refined prompt that increases accuracy (for instance, fewer missed or incorrect dates) compared to the MVP's approach; run A/B comparison on a set of pages to confirm improved performance (e.g., 10–15% boost in accuracy of field extraction); changes do not significantly degrade performance or increase cost beyond acceptable range.

---

1 2 3 4 5 6 7 9 10 11 12 13 14 15 16 17 18 19 20 ConfRadar MVP – Product

Requirements Document.pdf

file:///file\_0000000090d461f59957d7861436fbe5

8 ConfRadar.md

file:///file\_000000006ccc61f7b8d9c9c00b7ef7bb