Umbreeze AI Systems Division

presents

Rona AI Infrastructure & Learning Manual

A comprehensive guide to Rona’s AI architecture, infrastructure, and conceptual design

Prepared by cheerllydevil — 2025

Executive Summary

Rona v6 blends a desktop GUI with Retrieval■Augmented Generation (RAG), a local vector database, optional web enrichment, and a local LLM via Ollama. This manual explains the architecture visually and conceptually, so developers can grasp the ideas behind the code.

AI Infrastructure Diagram

How Rona’s pieces collaborate at runtime.

Core Modules & Components

Config

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	—		Function `__init__` — see usage in code
load_config	—		Function `load_config` — see usage in code

WebUIConfig

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	base_dir		Function `__init__` — see usage in code

LoadBalancer

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	—		Function `__init__` — see usage in code
monitor_temperatures	—		Function `monitor_temperatures` — see usage in code

get_current_temps	—		Function `get_current_temps` — see usage in code
adjust_load_balancing	—		Function `adjust_load_balancing` — see usage in code
get_optimal_gpu_layers	—		Function `get_optimal_gpu_layers` — see usage in code

SimpleOllama

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	model		Function `__init__` — see usage in code
invoke	prompt		Function `invoke` — see usage in code

FileProcessor

Bases: object
Docstring: Multi-format loader with high-quality text normalization and RAG-friendly chunking.

Method	Args	Doc (first line)	Explanation
__init__	—		Function `__init__` — see usage in code

_normalize_text	text		Function `_normalize_text` — see usage in code
_split_docs	raw_text, source_meta	Split into RAG-friendly chunks using config; works with/without LangChain.	Function `_split_docs` — see usage in code
_to_documents	texts, meta		Function `_to_documents` — see usage in code
process_file	file_path		Load + chunk file for vectorization
_process_text	p		Function `_process_text` — see usage in code
_process_pdf	p		Function `_process_pdf` — see usage in code
_process_xml	p		Function `_process_xml` — see usage in code
_process_json	p		Function `_process_json` — see usage in code

_process_csv	p	CSV RAG-friendly documents with:	Function `_process_csv` — see usage in code
_process_html	p	Extract visible text from HTML; strip scripts/styles, keep title & meta description.	Function `_process_html` — see usage in code
_process_image	p		Function `_process_image` — see usage in code

ArabicProcessor

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	—		Function `__init__` — see usage in code
is_arabic	text		Function `is_arabic` — see usage in code
process	text		Function `process` — see usage in code

ImageCreator

Bases: object

Method	Args	Doc (first line)	Explanation

__init__	—		Function `__init__` — see usage in code
_parse_size	size_str		Function `_parse_size` — see usage in code
_draw_gradient	img, top, bottom		Function `_draw_gradient` — see usage in code
_wrap_text	text, font, max_width, draw		Function `_wrap_text` — see usage in code
create_image_from_text	prompt, size_hint=None		Function `create_image_from_text` — see usage in code
_extract_exif	path		Function `_extract_exif` — see usage in code
analyze_and_register	src		Function `analyze_and_register` — see usage in code

CNNImageProcessor

Bases: object
Docstring: CNN-based image processing for classification and object recognition.

Method	Args	Doc (first line)	Explanation

__init__	—		Function `__init__` — see usage in code
get_recommended_models	—		Function `get_recommended_models` — see usage in code
get_implementation_notes	—		Function `get_implementation_notes` — see usage in code

DatabaseManager

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	—		Function `__init__` — see usage in code
_initialize_database	—		Function `_initialize_database` — see usage in code
add_documents	docs		Insert batch into vector store
similarity_search	query, k=5		Function `similarity_search` — see usage in code

SQLiteManager

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	db_path		Function `__init__` — see usage in code
_connect	—		Function `_connect` — see usage in code
_ensure_tables	—		Function `_ensure_tables` — see usage in code
insert_document	table, filename, path, content, metadata, created_at		Insert record into SQLite with metadata
search	table, query, limit=5		Function `search` — see usage in code

SQLiteManagerSingleton

Bases: object

Attribute	Notes
_instance	class-level variable

Method	Args	Doc (first line)	Explanation
get	cls		Function `get` — see usage in code

PsychoStore

Bases: object

Method	Args	Doc (first line)	Explanation

__init__	path		Function `__init__` — see usage in code
_load	—		Function `_load` — see usage in code
_save	entries		Function `_save` — see usage in code
add_entry	title, date_iso, details, mood_0_10		Function `add_entry` — see usage in code
list_entries	—		Function `list_entries` — see usage in code
export_text_chunks	max_items=50	RAG-friendly short chunks for LLM context.	Function `export_text_chunks` — see usage in code
emotion_summary	—		Function `emotion_summary` — see usage in code

FlaskServerController

Bases: object

Method	Args	Doc (first line)	Explanation

__init__	app, host='127.0.0.1', port=8765		Function `__init__` — see usage in code
start	—		Function `start` — see usage in code
shutdown	—		Function `shutdown` — see usage in code

DatabaseManagerSingleton

Bases: object

Attribute	Notes
_instance	class-level variable

Method	Args	Doc (first line)	Explanation
get	cls		Function `get` — see usage in code

DeepSearchEngine

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	—		Function `__init__` — see usage in code
google_cse_search	query, num=5		Function `google_cse_search` — see usage in code

search_zoomeye	query		Query ZoomEye API for hosts/web
search_zoomeye_pages	keyword, start_page, end_page, question		Query ZoomEye API for hosts/web
local_db_search	query, k=5		Semantic search over vector DB
convo_search	query, conversation_history		Search recent conversation memory

QueryProxy

Bases: object
Docstring: Query Proxy + Augmenter

Method	Args	Doc (first line)	Explanation
__init__	search_engine, file_processor, max_fetch=6		Function `__init__` — see usage in code
_needs_augmentation	query		Function `_needs_augmentation` — see usage in code
_is_blocked_query	query		Function `_is_blocked_query` — see usage in code

ResponseFormatter

Bases: object

Method	Args	Doc (first line)	Explanation

format	text		Function `format` — see usage in code

TestSuite

Bases: object

Method	Args	Doc (first line)	Explanation
__init__	—		Function `__init__` — see usage in code
run_all_tests	—		Function `run_all_tests` — see usage in code

RonaAppEnhanced

Bases: ctk

Method	Args	Doc (first line)	Explanation
__init__	—		Function `__init__` — see usage in code
_show_dragon_splash	path, duration_ms=2000, on_done=None	Show a centered animated splash GIF, then close and call on_done().	Function `_show_dragon_splash` — see usage in code
_gif_delays_ms	path	Return list of per-frame delays (ms) for a GIF.	Function `_gif_delays_ms` — see usage in code

_animate_gif_tk	label, path, delays	Animate a GIF on a tk.Label using Tk's native GIF frames.	Function `_animate_gif_tk` — see usage in code
_load_gif_frames_with_durations	path, size=None	Return (frames, durations_ms) coalesced to full RGBA frames.	Function `_load_gif_frames_with_durations` — see usage in code
_load_gif_frames	path, size		Function `_load_gif_frames` — see usage in code
_show_four_dragons	duration_ms=5000	Show 4 animated GIFs (left/right/top/bottom). Auto-close after duration.	Function `_show_four_dragons` — see usage in code
_destroy_dragons	—	Cancel timer and destroy all dragon windows; clear refs.	Function `_destroy_dragons` — see usage in code
_normalize_time_terms	text		Function `_normalize_time_terms` — see usage in code
_start_entry_pulse	base='#9d2c2c', peak='#ae0c88', period_ms=900	Soft pulse between base and peak on entry halo.	Function `_start_entry_pulse` — see usage in code

_add_button_whoosh	btn, min_w=80, max_w=92, step=2, interval=18		Function `_add_button_whoosh` — see usage in code
_detect_lang	text		Function `_detect_lang` — see usage in code
detect_lang	text		Function `detect_lang` — see usage in code
_years_in_text	t		Function `_years_in_text` — see usage in code
_on_close	—		Function `_on_close` — see usage in code
_create_modern_ui	—		Function `_create_modern_ui` — see usage in code
_is_metadata	content		Function `_is_metadata` — see usage in code
_assess_answer_confidence	answer		Function `_assess_answer_confidence` — see usage in code

_mood_face	v		Function `_mood_face` — see usage in code
open_psycho_panel	—		Function `open_psycho_panel` — see usage in code
_psycho_add_entry	—		Function `_psycho_add_entry` — see usage in code
_psycho_open_list	—		Function `_psycho_open_list` — see usage in code
_psycho_export_json	parent		Function `_psycho_export_json` — see usage in code
_create_corner_icons	—		Function `_create_corner_icons` — see usage in code
_attach_context_menu_to_textbox	textbox		Function `_attach_context_menu_to_textbox` — see usage in code

_attach_context_menu_to_entry	entry		Function `_attach_context_menu_to_entry` — see usage in code
_initialize_agent	—		Function `_initialize_agent` — see usage in code
send_message	event=None		Function `send_message` — see usage in code
_append_conversation	role, text		Function `_append_conversation` — see usage in code
_reply_assistant	text		Function `_reply_assistant` — see usage in code
update_status	msg		Function `update_status` — see usage in code
_maybe_arabic	text		Function `_maybe_arabic` — see usage in code
grammar_correct	text		Function `grammar_correct` — see usage in code

_append_terminal	text		Function `_append_terminal` — see usage in code
confirm_and_run_shell	cmd	Ask the user to confirm before running a shell command,	Function `confirm_and_run_shell` — see usage in code
_ensure_ollama_model	name		Function `_ensure_ollama_model` — see usage in code
_run_lovely_summary	query		Function `_run_lovely_summary` — see usage in code
_run_shell_worker	cmd	Run the command using bash -lc and stream output live (unbuffered).	Function `_run_shell_worker` — see usage in code
_handle_command	raw		Function `_handle_command` — see usage in code
_process_with_agent	message		Function `_process_with_agent` — see usage in code

_fallback_unified_search_and_reply	message		Function `_fallback_unified_search_and_reply` — see usage in code
_is_refusal	text		Function `_is_refusal` — see usage in code
_answer_with_consolidated_methodology	query, context		Function `_answer_with_consolidated_methodology` — see usage in code
_answer_with_web_results_only	query, context		Function `_answer_with_web_results_only` — see usage in code
deep_search_dialog	—		Function `deep_search_dialog` — see usage in code
_deep_search_and_reply	query		Function `_deep_search_and_reply` — see usage in code
open_file_dialog	—		Function `open_file_dialog` — see usage in code

open_image_dialog	—		Function `open_image_dialog` — see usage in code
create_image_dialog	—		Function `create_image_dialog` — see usage in code
run_tests	—		Function `run_tests` — see usage in code
open_settings	—		Function `open_settings` — see usage in code
clear_chat	—		Function `clear_chat` — see usage in code
_run_hunt_command	target		Function `_run_hunt_command` — see usage in code
_run_lovely_mode	folder_path		Function `_run_lovely_mode` — see usage in code
_show_memory_summary	—		Function `_show_memory_summary` — see usage in code

_get_scan_recommendations	target		Function `_get_scan_recommendations` — see usage in code

Module-Level Functions

Function	Args	Doc (first line)	Explanation
tokenize	text		Function `tokenize` — see usage in code
overlap_score	a, b		Function `overlap_score` — see usage in code
ensure_ollama_running	—		Function `ensure_ollama_running` — see usage in code
ensure_ollama_model	model_name		Function `ensure_ollama_model` — see usage in code
expand_query_variants	query		Function `expand_query_variants` — see usage in code
rank_with_local_priority	candidates		Function `rank_with_local_priority` — see usage in code
cluster_snippets	snippets, max_clusters=5		Function `cluster_snippets` — see usage in code

build_connected_reasoning	query, ranked		Function `build_connected_reasoning` — see usage in code
synthesize_answer_from_clusters	query, clusters, sources		Function `synthesize_answer_from_clusters` — see usage in code
process_message	message	Process message using NLTK and spaCy for NLP analysis	Function `process_message` — see usage in code
choose_port	preferred=8765	Try preferred; if busy, ask OS for a free ephemeral port.	Function `choose_port` — see usage in code
create_psycho_app	—		Function `create_psycho_app` — see usage in code
extract_zoomeye_directive	query	Returns (natural_question, keyword, start_page, end_page) or (query, None, None, None)	Function `extract_zoomeye_directive` — see usage in code
_relevance	item_text, question		Function `_relevance` — see usage in code

main	—		Function `main` — see usage in code

AI Concepts Used by Rona

Concept	Explanation
Retrieval■Augmented Generation (RAG)	Retrieve chunks from Vector DB and memory, then ask the LLM to answer using
Embeddings	Numerical vectors capturing semantics; enable similarity search.
Vector DB / Similarity	Stores embeddings; finds nearest neighbors by cosine distance.
Chunking & Overlap	Split large text with overlap to preserve context recall.
Prompt Engineering	Design instructions/format to guide the LLM.
Context Window	Max tokens the model can read; keep prompts compact.
Temperature	Controls randomness: low=precise, high=creative.

Configuration & Tuning Guide

Key	Meaning
model_name	LLM via Ollama (e.g., mistral:7b, llama3:8b).
gpu_layers	Transformer layers placed on GPU for speed.
chunk_size / chunk_overlap	Granularity of text chunks for embedding.
temperature	Creativity of responses.
max_results	Top■k retrieved chunks for the model.
max_conversation_context	How many previous messages to include.
deep_search	Enable web/proxy augmentation.
arabic_processing	Enable Arabic shaping/bidi normalization.

Libraries & Internal Usage

Library	Role
requests	HTTP calls for APIs (ZoomEye, Google/DDG).
aiohttp	Async HTTP client for concurrent fetches.
sqlite3 / SQLiteManager	Relational store for notes/assets.
chromadb / faiss	Vector DB for embedding search.
nltk / spacy	Optional NLP preprocessing.
customtkinter (ctk)	GUI toolkit.
LangChain	Optional retrievers/memory/chain framework.
Ollama / ChatOllama	Local LLM inference wrapper.
re, datetime, threading, asyncio	Parsing, timing, concurrency.

Learning Resources (clickable)

- **3Blue1Brown – Neural Networks**: https://www.youtube.com/watch?v=aircAruvnKk
- **Google ML Crash Course**: https://developers.google.com/machine-learning/crash-course
- **DeepLearning.AI – Prompt Engineering**: https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/
- **The Illustrated Transformer**: https://jalammar.github.io/illustrated-transformer/
- **NVIDIA – RAG Explained**: https://developer.nvidia.com/blog/retrieval-augmented-generation-explained/
- **Pinecone – Vector Embeddings**: https://www.pinecone.io/learn/vector-embeddings/
- **LangChain Docs**: https://python.langchain.com/docs/
- **Ollama Library**: https://ollama.ai/library
- **AsyncIO in Python – Real Python**: https://realpython.com/async-io-python/

Developer Roadmap

- Master RAG fundamentals and vector search.
- Practice with LangChain retrievers and memory buffers.
- Run local LLMs via Ollama; evaluate different models & temps.
- Add new API sources to the retrieval layer with strict filtering.
- Iterate on prompt templates and ranking strategies for reliability.

© Umbreeze Research — Prepared by cheerllydevil, 2025