
Alexandria (ALX) Project

Admin	1
Name	2
Purpose	2
History and Status	2
Data Collected	3
Look:	3
Feel/emotions and physical:	3
What you think:	3
What you do: Personal and Work activities:	3
What goes on outside of you:	3
Data Collection Processes	4
Technology	4
Process	5
Immutable, Permanent and Trustless Data Storage	5
Privacy and Sharing	5
Networking, Collaborating and Leverage Expertise With And For Others	6
Roles & Structure	6
Buckets for Structuring My ALX system	6
Layer 1 — Governance / Authority	6
Layer 2 — Intent Layer	7
Layer 3 — Memory Layer/Structure	7
Layer 4 — Execution Layer/Modes	7
Chat	7
Code	7
Workflows/Autonomous agents	7
Agentic Commerce via Machine Payments Protocol	7
Layer 5 — Evaluation / Feedback	7
How This Is Evolving	8

Admin

Google doc URL for this document:

<https://docs.google.com/document/d/1Hhy9Aluf6O2RWMaQv3-tgOpyFmCHVfZVG5Uu4qw-viw/edit?usp=sharing>

This document lives at <https://eddiesoehnel.com/alexandria-project/>

Name

The Great Library of Alexandria in Alexandria, Egypt, was one of the largest and most significant libraries of the ancient world. Alexandria came to be regarded as the capital of knowledge and learning, in part because of the Great Library.

Purpose

ALX is a project to create an AI personal assistant that does the following:

1. find information, products, know-how, expertise and connects us to people automatically based on who we are, what we know, what we've done, where we live, what we do, who we do it with, what we like, what we own, and what we need;
2. makes us massively more productive by acting in place of us to to interact with others (planning/scheduling, helping others with basic things);
3. recalls the things we've done, the knowledge we've attained, the skills we've learned, the tools we've built (models, diagrams, processes, etc) so that we can use them in the future without forgetting they exist (and make it all searchable anonymously so others can benefit as well);
4. brainstorms with us because it knows us and knows what we know, how we think, what we like;
5. surfaces patterns in our lives that we cannot see but it can to help us improve (like health data);
6. clones our likeness so that when we die, our loved ones can still interact with our likeness and ask us for advice;
7. archives our knowledge and experience so that when we die, what we learned and know can live on to help others;
8. we bond with it emotionally and can talk to it as we would a loved-one or friend, and it responds because it knows us.

To get there, we need to identify the data that the AI needs on us, the data collection processes and the technologies we need to deploy the AI.

The AI will only be as good as the data we give it to train it and work on our behalf.

I update this document to reflect new information.

History and Status

I started thinking about this project in 2019 and began collecting the data profiled in this document.

Tony Stark in the Marvel superhero movies had Jarvis. I have named my version of Jarvis, Alexandria, after the Library of Alexandria in ancient Egypt.

Fall 2025 I began building the scaffolding and coding to structure almost 30 years of digital content for ingestion by AI. As I move further along, I will likely post the tools and code on Github so that anyone can build their own version of Jarvis.

Data Collected

Think very broadly about information and data that is being generated (not all of it necessarily collected but could be collected):

- How you look
- How you feel/emotions and physical
- What you think
- What you do
- What goes on outside of you, your loved ones, co-workers, friends, your house, street, community, state, country, region, the world; the weather

Format: visual, audible, smell, taste, feel

Look:

- Self pics daily, weekly, or at some periodic interval or which us auto-captured through images and video others take of you or where you are included

Feel/emotions and physical:

- Personal journal
- Health collection devices

What you think:

- Personal journal

What you do: Personal and Work activities:

- Emails, meetings, calls
- Tools you create for personal activities: methodologies, processes, tools, spreadsheets, diagrams, resource lists, research, data
- Places you go, things you do, people you do it with: location data, images, video, health collection devices, social graphs, experience graphs, ownership graphs of things you do stuff with.
- What you learn: reading, watching, doing, experiencing

What goes on outside of you:

- People are generating/collecting their own data similar to you
- Objects are collecting data via IoT
- News is happening and being reported

Data Collection Processes

A lot of information is already collected and stored automatically in digital format. Think of calendars and emails. The goal is to make sure this data is saved and accessible to give to an AI in the future.

The goal is to also be proactive to collect and store other information digitally that an AI would need. What is stuff that might not be collected automatically that which you need to be proactive:

1. Keep a personal journal - audio and video is probably best because an AI can learn how you speak and your facial expressions to replicate.
2. Wear personal health collection devices like smart watches
3. Get out of your head all your knowledge, know-how, expertise and experience. Some of this may already be captured, especially in context of work through emails, meetings, calls, file repositories and databases where we store methodologies, processes, tools, spreadsheets, diagrams, resource lists, research, data. But you have to be proactive in this area because many things, like your experiences and what they teach you, may not be digitally collected, so you need to be proactive in doing that.
4. What did you read, watch, do, experience, who did you meet? Record as much as you can or at least include links/sources to where the information is so that it can be scraped in the future.
5. Add IoT or electronic devices that create persistent data collection.
6. In the future hopefully we can add the entire contents of books/articles we read to our knowledgebase so the AI can archive that as part of our memory.

I use a Daily Log Template as a guide for what to collect daily so I do not forget

Technology

We already use a ton of technology to gather data. Think about new technologies coming along that collect more data, better and make data collection and storage easier.

This especially applies to personal health data collection devices.

Think of networks of nanosensors and “smart dust” —sensors the size of a match head that have onboard power supplies, computation abilities and wireless connectivity.

Think of using data storage systems that are not proprietary and have easy API capabilities to plug into an AI system.

Think of the holes in your data collection process and what new tools could help plug those holes.

Make a list of where all data is stored:

-
- Email
 - Calendars
 - Contacts
 - Files (images, video, documents, databases with structured data, etc)
 - Health/exercise data
 - Social and CRM data
 - Purchase history
 - Browser history
 - Ownership graphs - what you own
 - Location data
 - Communication logs and content

Think of the lowest common denominator of file types: txt, doc, csv, mp3, ma4, mp4, pdf

Process

The challenge in all this is collecting the data so that it is really easy and fast. To make it comprehensive, a combination of using apps/devices to collect the data, then automatically send it to watched folders, and using scripts/smart agents that detects the content in the folders, normalizes it, structures as needed, sends to a database, which will then vectorize the content for AI. Very doable with today's technology.

Ideal is to capture everything - but that could be very time consuming and inefficient. Is there a middle ground to make info capture fast and not onerous, where it fits naturally within the day?

Use personal audio devices that record audio like a journal and reference other deeper content that may be captured.

Immutable, Permanent and Trustless Data Storage

I use cloud services but backup all my data to my home computing environment. As open source AI models proliferate and gain inference efficiencies, it will be possible to locally host all data as well as the AI model. This is the direction I will go.

Privacy and Sharing

What are the rules around which data my AI can access if it interacts with the public for me?

The control plane that has emerged for me is a CSV file that stores all the metadata to my content and which contains the rules for how that data is accessed and used.

No one trusts companies with their data and deservedly so. Fully homomorphic encryption could solve this problem and is becoming cheaper to use

Networking, Collaborating and Leverage Expertise With And For Others

See more at [Decentralized AI-Native Professional Networking](#)

Roles & Structure

I think of my personal AI across several different roles.

(h) Eddie: is the human

(d) Eddie: is my but in digital format via an AI that is me based on all the data I give it.

I could have (d) Eddie act in place of (h) Eddie as needed, like answering emails. I would restrict (d) Eddie from accessing personal journal content, but it could have access to all my prior work content

Alexandria would be an agent - like Jarvis from the Marvel Universe. It not only has my complete content history that I give it, but I could add other knowledgebases so it becomes more than just me. I am thinking of access to other major frontier AI models and the world knowledge they have accumulated. From this context, Alexandria knows what I know and alot more. But knowing me allows it to respond to me in context of me.

If Alx is my main super agent, could other agents roles emerge to do things? We are moving in the direction of having discrete agents that do limited tasks which they are very good at. Could Alx interact with Rhodes, another agent I create to manage my edge data center and all the compute infrastructure? Maybe Rhodes is more focused, with access to all my digital operations. SOPs and document, while connecting to a knowledgebase for architecture, engineering, sys admin and coding. Alx could query Rhodes as needed.

Buckets for Structuring My ALX system

Layer 1 — Governance / Authority

What is allowed to act, where, and with what permissions? The control plane lives here:

- When emails can be automatically sent
- Which topics require human review
- What commitments the system cannot make
- Disclosure requirements
- Data access permission

Layer 2 — Intent Layer

Defines what the system is optimizing for.

- Example intents
- Preserve relationships
- Save Eddie time
- Provide accurate information
- Avoid speculation
- Escalate uncertainty

Intent acts as the decision-making compass guiding system behavior.

Layer 3 — Memory Layer/Structure

What is the data, from where (see above we already discussed) How is it structured (JSON, JSON chunks, with deep metadata and ontology)

Layer 4 — Execution Layer/Modes

I think of AI across three main layers. Alx works in all layers, but I like to think about which layer it is working in because that defines the context it needs.

Chat

Brainstorming, Q&A, back and forth

Code

Coding in terminal or via apps that interact directly with my repos and virtual machines where apps are hosted.

Workflows/Autonomous agents

Agents built for specific workflows or tasks that get spun up as needed.

Agentic Commerce via Machine Payments Protocol

I am adding this as a 4th execution layer that may change. This is taking content and knowledge and encoding it so that other agents can access it and pay for it.

Layer 5 — Evaluation / Feedback

How do we detect quality, drift, errors, hallucination, regression?

Ensures the system improves and remains aligned.

Evaluation methods may include:

- response audits
- escalation tracking
- hallucination monitoring
- confidence thresholds
- human feedback loops
- periodic system reviews

Evaluation prevents behavioral drift and maintains reliability. Would an entirely new agent be created just to do audits of Alx that has access to specific knowledgebases and tools to help it do its job?

How This Is Evolving

Check out Gbrain as a memory system. Amazing project.

A process for designing agents has emerged and best practices are surfacing. Here is a high-level overview of my interpretation for best practices around agent development. As I get further along in using it to build my own agents I will add more about it.

1. Define objective → Design layer
2. Choose model → Harness
3. Define tools → Tools (deterministic layer)
4. Define data sources → Tools + Resolvers
5. Write prompt → Skills (procedural cognition)
6. Build harness loop → Harness
7. Add memory → Harness + Tools + Skills
8. Add context control → Resolvers + Harness
9. Define output → Skills + Tools
10. Set trigger (cron/API) → Harness

11. Add logging → Harness

12. Add constraints → Harness + Skills