ChatGPT

SGraph.AI – Business Plan

Executive Summary

SGraph.AI is an open-source driven company focused on helping organizations transition their data into **semantic knowledge graphs**. By converting unstructured and siloed data into a connected graph format, SGraph.AI enables businesses to unlock new insights, improve data interoperability, and future-proof their information management. The company's strategy centers on an open-standards platform (built on proven technology from the founder's prior ventures) offered alongside commercial services – including a cloud-based **Semantic Text-to-Graph API**, on-premises deployments, and expert consulting – to guide enterprises through this transformation. Initial efforts will focus on a highly scalable **semantic text transformation service** (turning documents and content into knowledge graphs) as a core product, supplemented by professional services for integration and customization. SGraph.AI's open-core model will build community trust and adoption quickly, creating a user base that can be monetized via value-added services and support [1]. The company will pursue an organic go-to-market strategy: releasing the core technology openly to drive adoption, engaging early customers through pilot projects and consulting, and scaling up to a sustainable SaaS and enterprise licensing model. External seed investment (offering roughly 10–15% equity) is sought to accelerate development, establish initial customer traction, and position SGraph.AI as a pioneer in the emerging semantic graph market. *(Pricing and exact market segmentation will be refined through early customer feedback.)*

Market Opportunity and Problem Statement

Today's organizations are awash in data, but much of it is **trapped in formats and silos** that limit its value. Critical information resides in spreadsheets, documents, legacy databases, and various proprietary systems that don't "talk" to each other. This fragmentation means companies struggle to get a unified view of their knowledge. Valuable insights remain hidden because data is unstructured or scattered. Moreover, as enterprises adopt AI and large language models, they are discovering that feeding **unstructured text blobs** into AI yields shallow insights – context is lost and explainability is poor [2] [3]. There is a growing recognition that **semantic knowledge graphs** – which represent information as connected entities and relationships – can vastly improve data integration, search, and AI understanding.

However, transitioning to a semantic graph-based infrastructure poses challenges: - **Lack of Expertise:** Most companies don't have in-house knowledge graph or ontology experts. Defining the right ontology/taxonomy for their domain and mapping existing data to it is daunting. - **Tooling Gaps:** Existing solutions for building knowledge graphs are either proprietary (causing vendor lock-in) or technically complex. Organizations fear that their data could become locked in yet another system. - **Data Volume and Fragmentation:** Enterprises have massive amounts of documents, emails, logs, etc., and need automated ways to convert this content into structured knowledge. Manual curation doesn't scale. - **Explainability and Compliance Needs:** In regulated industries and corporate governance, any AI-driven analysis must be explainable and traceable. Companies need solutions that can show provenance of insights (e.g. which source document a conclusion came from). Traditional AI approaches lack this transparency.

These pain points are evident in sectors like finance, cybersecurity, healthcare, and government, where data is plentiful but underutilized. For example, professionals face **information overload** (e.g. a CISO sifting through threat reports or an analyst monitoring industry news) and miss important insights because there's no personalized, concept-aware filtering [4] [5]. Tools that merely do keyword matching produce too much noise. This represents a **significant market opportunity**: organizations are seeking ways to harness all their data in a meaningful, structured way to drive decisions, while avoiding platform lock-in and maintaining transparency.

SGraph.AI's Opportunity – By providing an open, standards-based semantic graph platform, SGraph.AI addresses these needs head-on. We enable companies to **liberate their data from silos** and convert it into a **unified knowledge graph** that is machine-interpretable and rich in context. This not only improves internal data integration and analytics, but also enhances the effectiveness of AI/LLM solutions (since structured knowledge graphs can be used for more precise reasoning and recommendations [6] [3]). Importantly, SGraph's use of open formats and open-source technology means **no vendor lock-in** – a key selling point for governments and enterprises with long-term data ownership concerns. In fact, organizations with open data mandates or compliance requirements can particularly benefit from SGraph.AI's approach, as it promotes data transparency and portability.

Solution Overview: SGraph.AI Platform and Services

SGraph.AI offers a platform that transforms raw data (text and other media) into semantic knowledge graphs using advanced AI and open standards. At its core, SGraph.AI is built on a **Semantic Text-to-Graph Service** – a highly scalable engine that ingests unstructured text and outputs a structured representation of knowledge (entities, attributes, and relationships) in graph form. This core service is accessible via API and can be deployed in various modes (cloud, on-premises, or at the edge), providing flexibility in how organizations adopt the technology.

Key characteristics of the SGraph.AI solution: - **Open-Source Core & Open Standards:** All fundamental components of the platform are open-source and built on open standards for data interchange (e.g. JSON-LD or RDF for graph data). This means the customer's data and ontologies remain in standardized formats that they control. The open-core approach instills trust – clients and third-party developers can inspect the code and verify security/integrity [7]. It also accelerates integration, as others can build on the tools freely, fostering an ecosystem around SGraph.AI [7]. - **Highly Scalable Architecture:** The platform is designed as cloud-native microservices (primarily FastAPI-based) that can run anywhere (public cloud, private cloud, or even on a developer's laptop). It leverages containerization and serverless deployments for elasticity. In shared cloud mode, it can autoscale to handle bursts of documents, but it can just as easily be packaged for offline or on-prem use. This "runs everywhere" design is backed by a unified CI/CD pipeline that the founder developed, which packages applications to run on any environment with minimal overhead [8]. This ensures we can deploy SGraph with equal ease in an AWS Lambda, a Kubernetes cluster, or an on-site server farm. - **Semantic Conversion Pipeline:** SGraph's core workflow for data processing is **LETS** – *Load, Extract, Transform, Save* – inspired by traditional ETL pipelines. Every piece of data goes through a controlled sequence: 1. **Load:** Ingest the data source (e.g. text documents, PDFs, HTML pages, etc., or fetch from an external system via connectors). 2. **Extract:** Parse and clean the raw data. For text, this means extracting relevant sections, metadata, and preparing it for analysis. *(In the case of MyFeeds.ai, for example, articles from RSS feeds are fetched and parsed – capturing the raw text, author, date, and even HTML structure – and saved in a content store)* [9]. 3. **Transform:** Apply AI (large language models and custom NLP) to interpret the content and convert it into a **semantic graph structure**. This involves extracting key **entities, concepts, and relationships** and representing them in a graph format [10]. Essentially, the unstructured input is turned into a structured knowledge representation. *For example, an article about a new ransomware attack can be transformed into a graph with nodes like "Ransomware X", "Healthcare Industry", "Data*

Breach", "Country Y", and an "Impact $$$" node, with relationships indicating that Ransomware X caused a data breach in the healthcare sector in Country Y [10] *.* The result is a machine-interpretable summary of the content, stored as a graph [11] . 4. **Save:** Store the results in both raw and processed forms. SGraph uses a custom **GraphFS/MemoryFS** layer to save data uniformly either as files or as graph nodes in a database [12] . The output graph can be stored in a graph database or serialized to JSON-LD. The raw input and intermediate metadata can also be saved for audit trails. This structured pipeline improves transparency and provenance, since each step is logged and traceable. It's critical for debugging AI decisions and maintaining trust in the output.

- **Ontology and Taxonomy Management:** A core part of the platform is helping clients define or discover their **ontologies** (the concepts and entity types relevant to their business) and **taxonomies** (how those concepts are organized). SGraph can auto-suggest an ontology by analyzing a sample of documents (extracting recurring entities/topics). Alternatively, if a client already has a data model or industry schema, SGraph's transformation service can take a **"bring your own ontology"** approach – mapping their unstructured data into their predefined schema. This flexibility means SGraph supports both greenfield ontology creation and integration with existing knowledge models.
- **Multi-Ontology Support:** Uniquely, SGraph.AI recognizes that within a large organization, different departments or datasets may require different ontologies (or different "altitudes" of abstraction). The platform is designed to allow multiple semantic schemas to coexist and interlink. For instance, an HR dataset and an IT dataset might have distinct schemas, but SGraph can link common entities (like a person or location) between them. This addresses the real-world scenario that one size does *not* fit all for knowledge graphs – SGraph helps **harmonize data at the right level of meaning**.
- **Knowledge Graph Storage and Querying:** Once data is transformed, the resulting knowledge graph can be stored in a graph database or as serialized files (JSON-LD/OWL, etc.). Thanks to the GraphFS abstraction, data can be **queried and visualized seamlessly** – users can traverse relationships and run complex queries across the graph (for example, finding connections between a user's interests and an article's topics). SGraph doesn't lock customers into a particular database; the data can be exported or moved as needed since it remains in standard formats. We anticipate integrating with popular graph databases (Neo4j, Amazon Neptune, etc.) as backends, or clients can use the built-in lightweight GraphFS store for smaller-scale needs.
- **LLM-Powered Insights and Explainability:** SGraph.AI leverages large language models in the transformation step, but unlike black-box AI solutions, it produces **explainable outputs**. The knowledge graph structure inherently provides a form of explanation: users can trace *why* a piece of information was categorized a certain way or *how* two pieces of data are related. This is a major benefit – for example, in the MyFeeds application built on this technology, the system could explain why it recommended a given article by showing that "this post was suggested because it discusses **Topic X** and **Topic Y**, which are in your profile" [13] . Graph representations enable this kind of transparency. In enterprise use, this means any AI-driven analysis (recommendations, alerts, summaries) can be traced back to source data points on the graph, facilitating compliance and trust. SGraph essentially acts as the "semantic brain" that not only powers smarter automation but can *show its work* when needed.

In summary, SGraph.AI's solution is a **combination of product and platform**: a robust semantic graph **transformation engine** (productized via API and apps) underpinned by an open, extensible knowledge graph **platform** that enterprises can adopt and customize. Everything is built with enterprise-grade considerations – security, access control, scalability, and interoperability – to ensure it can be introduced into complex corporate IT environments.

Product Offerings and Services

SGraph.AI will deliver value to customers through a range of offerings, from self-service APIs to fully managed enterprise solutions. The **company's revenue model** is anchored in providing services on top of the open-source core technology. Key offerings include:

- **1. Semantic Text-to-Graph API (Cloud Service):** This is the flagship service – a cloud-based API where users can submit textual data (documents, articles, reports, etc.) and receive as output a structured semantic graph (likely as JSON or RDF). This service can be used on-demand (pay-per-use or subscription tiers). It lowers the barrier for any developer or company to start converting their data to a knowledge graph. For instance, a customer could send a policy document to the API and get back the key entities (e.g., regulations, departments, dates) and their relationships in JSON format, along with an auto-generated ontology. The **Semantic Text Service** thus serves as an entry point for companies beginning their semantic journey. We may extend similar APIs to other data types (e.g., **Semantic PDF Service**, **Semantic Spreadsheet Service**, etc. for specialized processing), but all ultimately feed into the same knowledge graph format. The focus will be on simplicity, scalability, and quality of the transformation. Pricing for this service will likely be usage-based (per document or per quantity of text processed), possibly with a freemium tier for developers to experiment.

- **2. Semantic Graph Platform (Managed Deployment):** For customers with higher security or customization needs, SGraph.AI will offer dedicated deployments of the platform. This could range from a private instance running in our cloud (single-tenant SaaS) to an **on-premises package** that runs in the customer's environment. Because the core is open-source and containerized, we can deploy it within a client's AWS/Azure account, on their servers, or even entirely offline if needed. We will provide **Managed Hosting** options where we handle all maintenance, updates, and scaling on behalf of the client (with appropriate SLAs). We anticipate offering tiers of isolation:

- *Multi-tenant Cloud:* Customer data is processed in a shared environment (with strong logical separation and encryption). Lowest cost option, ideal for lighter workloads or less sensitive data.
- *Dedicated Cloud Instance:* A separate instance/environment for the client (e.g., their own set of containers or VMs), providing greater data isolation and customizability (at a higher price point).

- *On-Premises or VPC Deployment:* The SGraph platform deployed in the client's own infrastructure. This would come with an annual license or support contract. We will **"productize" the on-prem deployment via container images or AWS AMIs** that are easy to install – effectively making the adoption as seamless as possible, even though it's running in the customer's environment. Part of our revenue here comes from **maintaining these deployments** for the client: we would provide updates, patches, and even custom integrations through a support subscription. Because we maintain the core codebase, it's more efficient for us to manage these updates – the client gets improvements without dedicating internal teams to it. Our value proposition: it will be **cheaper and easier to pay SGraph.AI to run and maintain the semantic graph infrastructure than to do it themselves**, even though they have full access to the source [1].

- **3. Custom Integration & Ontology Consulting:** Many enterprises will need help identifying how to fit SGraph into their ecosystem. SGraph.AI will have a **professional services arm** that works closely with clients on:

- *Ontology/Taxonomy Development:* Helping define the semantic models for the client's domain. This could involve workshops with their domain experts and using our tools to auto-generate initial ontologies from sample data.
- *Data Source Integration (Connectors):* Building connectors to the client's existing data sources (databases, document repositories, SaaS tools like Confluence/Jira, etc.). For example, a connector might pull all documents from a SharePoint library or all records from a legacy CRM, so they can be semantically processed. While we will develop an open library of connectors, enterprises may pay us to develop **new connectors or customize existing ones** for their specific needs. This "connector ecosystem" can drive revenue: it's often more cost-effective for a client to fund us to build a connector (which we can likely do faster with our expertise) than to use their own developers. Over time, a rich set of connectors also increases the usefulness of the platform for all users.
- *Workflow Automation:* Customizing how the semantic pipeline fits into the client's workflows. For instance, setting up a pipeline where any new document added to a certain folder automatically triggers the SGraph transformation and the result is posted into a knowledge base. These kinds of automations can greatly increase the value the client gets, and we can charge for implementing them.
- *UI and Application Layer:* While SGraph focuses on the backend service, clients might want custom user interfaces (e.g., an internal dashboard to query the knowledge graph, or integration into their existing tools). SGraph.AI and its partners can develop front-end applications or visualizations on top of the client's knowledge graph. We envision an ecosystem where third-party developers might build specialized apps on the SGraph data, but initially we'll provide services to jump-start this for clients.

This consulting business not only generates revenue but also deepens our relationship with customers (leading to retention and upsell of software services). It also ensures successful adoption – which in turn creates reference cases for further sales. Notably, even leading AI companies today are finding that having a strong services component is key to helping customers implement AI solutions effectively; SGraph.AI will follow this model in the early stages.

- **4. Support & Maintenance Subscriptions:** For open-source users and enterprise deployments, we will offer paid support plans. Enterprises adopting the open-core on their own can subscribe for **professional support**, which includes timely patches, security updates, and helpdesk assistance. Additionally, for on-prem deployments, we can offer a managed update service. As mentioned, one innovative approach is providing clients with a **private GitHub repository** that contains their customized version of the SGraph platform (including any client-specific configurations or extensions). We maintain this repo for them – merging in upstream improvements, running automated tests (with synthetic data that simulates their environment) to ensure updates don't break their specific setup, and then helping them deploy updates safely. This effectively outsources the headache of maintenance to us. Clients pay an annual fee for this peace of mind. It's akin to the Red Hat model: even though the software is free, the **reliability and support** we offer is worth the subscription cost.

- **5. Data Hosting and Value-Added Services:** As an optional add-on, SGraph.AI can provide hosting for the resulting knowledge graphs and original data in a secure data store. For example, a client might use the API just to convert documents and handle the graphs themselves, or they might ask SGraph to **store and manage the knowledge graph** in the cloud for them. We could charge for **storage and database** provisioning, effectively acting as their knowledge graph repository. This includes features like:

- *Encrypted Data Vault:* We store their parsed data and graphs with strong encryption, and offer role-based access control for their team.
- *Disaster Recovery Backups:* A service where we keep redundant backups of the ingested data and graphs. This ties into a compelling side benefit of our approach – by creating an offline copy of their raw data during processing, we effectively create a backup that can protect against ransomware or accidental loss in their primary systems. (If their original data repository goes down or gets corrupted, they have the data preserved in an accessible form in the knowledge graph store.) SGraph can formalize this into a **Business Continuity offering**, ensuring that as we ingest and transform data, the originals (or cleaned versions) are archived in an immutable storage. This can be a selling point for customers concerned about data resilience and compliance.
- *Real-Time Sync:** For certain connectors, we can set up continuous sync so that changes in the source system update the knowledge graph in near-real-time. This keeps the graph up-to-date. Such advanced features would be premium services.

- *Analytics & Dashboard:* We might offer a basic web dashboard for customers to explore their knowledge graph, run queries, and see analytics (like how many documents processed, trend of certain entity occurrences, etc.). This could be bundled with higher tiers or as a separate module.

- **6. Future Products:** Once the core platform gains traction, SGraph.AI can spawn specialized products. For example:

- *Semantic Analytics:* Tools to perform analytics directly on the knowledge graph (answering complex business questions by traversing relationships, something traditional BI struggles with).
- *Domain-Specific Ontology Packs:* Pre-built ontologies and models for specific industries (e.g., a healthcare pack, a cybersecurity pack). These could be sold or open-sourced to spur adoption, and then consulting services provided for tailoring them.
- *Graph-enhanced LLMs:* As large language models improve, SGraph could offer an AI assistant that leverages the company's *own* knowledge graph to answer questions or generate reports. This would differentiate from generic AI (which can hallucinate or miss company-specific context) by grounding responses in the firm's verified knowledge graph. This is a longer-term play that sits on top of the data foundation we help build.
- *Marketplace:* If adoption grows, there could be a marketplace for connectors, plugins, or even data/ontology sets, where SGraph or third parties offer extensions. This is speculative at this stage but aligns with an open ecosystem approach.

Summary of Monetization: All the above offerings mean SGraph.AI will have multiple revenue streams: recurring SaaS/API revenue, annual enterprise licenses, consulting fees, and support contracts. This diversified model ensures we can start generating income early (through consulting and support) even as we build out the scalable product revenue (cloud API, enterprise licenses) over time. It also aligns our incentives with customer success – the more data they transform and the more use cases they find, the more value we can provide and charge for.

Importantly, **our open-source foundation is a feature, not a bug, in the business model**: by **giving away the core tech**, we aim to become the standard in semantic transformation, achieving widespread adoption. A wide user base increases our chances of landing big accounts and also justifies value-added offerings. Many successful companies (database companies, AI platforms, etc.) have shown that having a popular open-source project can drive enterprise sales – investors recognize metrics like community adoption, GitHub stars, and download counts. SGraph.AI will track and promote such metrics to demonstrate traction. High adoption could even lead to strategic partnerships or acquisitions down the line, but our focus for now is on sustainable monetization via services. In essence, **we commoditize the**

data transformation layer (making it cheaply and widely available), and then **commercialize the layers above it** (deployment, customization, support) – a classic open-core play [1] .

Go-to-Market Strategy

Our go-to-market will be **organic and community-driven in the early phase**, transitioning into a more traditional enterprise sales approach as we mature. Here's how we plan to execute:

- **Phase 1: Community & Developer Adoption (Months 0-6)**
  In the first phase, we will focus on releasing the core technology and attracting early adopters (developers, data scientists, innovation teams in companies). This involves:
- **Open-Source Launch:** We will release the SGraph.AI core (the semantic transformation engine, connectors framework, etc.) on platforms like GitHub under a permissive license. Accompanying this will be thorough documentation, quick-start guides, and maybe a tutorial video or demo. The aim is to lower the barrier for technically inclined users to try it on their own data. By demonstrating the platform's capabilities openly, we build credibility.
- **Tech Community Engagement:** We'll engage on forums (Stack Overflow, Reddit data engineering threads, etc.), write blog posts on the benefits of knowledge graphs, and possibly present at meetups or webinars on semantic AI. Dinis Cruz (the founder) already has credibility as an open-source advocate, which we will leverage to get attention. We'll share how SGraph solves real problems encountered in his other ventures (for instance, curating personalized feeds via MyFeeds). These real use cases make the technology tangible.
- **Leverage Existing Projects as Showcases:** MyFeeds.ai, an existing prototype from our ecosystem, is a great showcase of the tech – it **uses semantic graphs and LLMs to deliver personalized news feeds**, solving info overload [14] . We will highlight this case study: e.g., "Look how using SGraph technology, MyFeeds was able to match content to user interests with concept-level precision." One example: a CISO profile concerned with supply chain attacks got an article about an NPM package vulnerability recommended – even though the article never said "supply chain", the knowledge graph inferred the connection [15] . Such stories demonstrate the power of our approach in action. We will prepare a few case studies (cybersecurity news, compliance document analysis, etc.) to illustrate ROI.
- **Initial Cloud Service Preview:** In this phase, we may run a limited beta of the cloud API. Interested users (who discovered us via GitHub or blogs) can sign up to get an API key and process a limited number of documents for free. This helps us gather usage data and feedback on the service (e.g., accuracy of extraction, performance).
- **Identify Early Evangelists:** Often, early adopters of open-source projects become evangelists if they find value. We will nurture relationships with a handful of such users – for instance, a data scientist at a mid-size company who tries SGraph and loves it. Their testimonials or word-of-mouth can drive further adoption, especially if they publish about how they used SGraph to solve a problem.

The goal of Phase 1 is to **establish credibility and prove demand**. Success looks like: a growing GitHub repository with stars/contributions, a few hundred developers signed up to the beta API, perhaps one or two community-contributed connectors or integrations, and inbound inquiries from organizations who see potential.

- **Phase 2: Pilot Projects & Professional Services (Months 3-12)**
  In parallel with the community push, we will actively seek out a few **pilot customers** for deeper engagements. These are likely to be companies (or government agencies) that have an acute need for what we offer – perhaps they've expressed interest in building a knowledge graph or they have a specific project (e.g., consolidating research documents, or enabling AI on their

internal knowledge base). Through the founder's network (20+ years in security and enterprise consulting) we already have some leads. For example, security firms interested in better organizing threat intelligence, or enterprises wanting to apply GenAI to their document archives in an explainable way.

Our approach to pilots: - **"Free" Consulting to Land Logo:** For 1-3 strategic logos, we might initially offer our services at cost or even pro bono (with a mutual understanding that if value is shown, they will be a reference and potential paying customer). This can jumpstart our learning and demonstrate real-world value. We will work closely with their team to implement SGraph for a specific use case, essentially acting as a specialized consultant. While this doesn't bring immediate profit, it's essentially R&D funded by customer problems – ensuring our product evolves in the right direction. It's also a foot in the door for future deals. - **Tailored Solutions:** Each pilot will likely focus on a particular domain or problem (e.g., creating a knowledge graph of all company policies to enable better Q&A for employees, or integrating SGraph into a cybersecurity product to add semantic analysis features). We will tailor our platform to these needs – perhaps building new connectors or features as required. This not only solves the client's problem but enriches our platform for broader use. *(One potential B2B2B angle: partnering with a cybersecurity software vendor to embed SGraph's capabilities into their product, so they can offer semantic analysis of security logs or alerts. We do the heavy lifting behind the scenes.)* - **Show Quick Wins:** We aim to deliver visible results within a short time (say a 4-8 week pilot). For example, after a pilot, the client should have a small but working semantic graph of one of their data sources, and a simple query or application demonstrating its value (like "find all documents related to X across our silos" or "get an alert when two related entities appear in separate data feeds"). Quick wins will generate enthusiasm and help convert the pilot into a paid phase 2 (or expand to more datasets). - **Capture Metrics:** During pilots, we'll capture data that will be gold for sales/marketing later. E.g., "After using SGraph, **Company X reduced document search time by 80%**" or "Our system linked 5,000 customer records with policy documents, revealing 15% previously unknown connections." Such quantitative benefits will strengthen our market proposition.

Meanwhile, we will continue to offer paid consulting for any inbound requests that are not just pilots. If, say, a company directly asks us to help implement an internal knowledge graph and is willing to pay for our team's time, we will engage (selectively, ensuring alignment with our product roadmap). This brings early revenue and validates willingness to pay.

- **Phase 3: Productization & Scale (Months 12-24)**
  After gathering feedback and solidifying the platform through pilots, we'll double down on the **scalable product offerings**:
- **General Availability of Cloud Platform:** We will officially launch the SGraph.AI cloud service (API and perhaps a simple web app interface). Marketing efforts will ramp up around this time. We'll emphasize how easy it is – "Just feed us your documents, and get a knowledge graph." We may introduce pricing at this stage (with free tier and paid tiers).
- **Sales Focus for Enterprise Deals:** With case studies and reference customers from Phase 2, we will approach larger organizations for enterprise subscriptions. Likely targets: Fortune 500 companies in data-rich industries (finance, pharma, tech) and government agencies focused on open data or intelligence analysis. Our sales strategy will highlight key differentiators:
  - Open-source transparency (no lock-in, can self-host if desired).
  - Proven technology (with references and metrics from pilots or the MyFeeds case).
  - Flexible deployment (meeting strict privacy/security needs).
  - The combination of product + experts (we're not just tossing software over the fence, we partner in success).
- **Partnerships:** We will seek partnerships with cloud providers and consulting firms. For example, partnering with a major cloud (AWS/Azure) to have SGraph available in their marketplace or even

integrated with their solutions (imagine SGraph powering a “knowledge graph” add-on to cloud data lakes). Also, data science consulting firms could become implementers of SGraph – we provide the tech, they configure it for their clients (similar to how some firms specialize in deploying Elastic or Neo4j for others). This can expand our reach without a large direct salesforce.

- **Marketing & Thought Leadership:** We’ll produce whitepapers and perhaps host workshops/webinars on topics like “Enterprise Transition to Semantic Knowledge Graphs” or “How to Make Your Data AI-Ready with Knowledge Graphs”. These will educate the market (and subtly market our offering as the solution). We’ll use content marketing to highlight issues of data lock-in, the importance of open standards, etc., positioning SGraph.AI as a visionary in this space.
- **Targeted Verticals:** If we see particular traction in a vertical (say, cybersecurity because of our background, or legal because of heavy text documents), we might create targeted messaging or slightly tailored solutions for those. For instance, a “SGraph for Compliance” package that includes pre-built ontologies for regulations, and connectors to common systems like SharePoint or document management systems used by legal teams.
- **Expand Connector Library:** By this stage, we should invest in building out connectors to popular enterprise systems (databases, SaaS apps). The easier it is for customers to plug SGraph into their existing workflows, the faster the adoption. We might open-source many connectors to encourage community contributions as well.

The *go-to-market trajectory* is designed to start small (low-cost, low-friction adoption) and gradually move upmarket: - Early developer adoption leads to grassroots usage (and some viral growth within circles that care about knowledge management). - Success stories from that feed into convincing management at larger organizations. - We convert interest into pilot projects, which then convert into full deployments and recurring licenses. - Over time, as the product matures, sales cycles should shorten for new customers because the value proposition will be proven and the platform more plug-and-play.

Crucially, our strategy of **open-source core + value-added services** helps drive an **organic pipeline**: some users will start with the free/community version, and as their needs grow (or as they convince their bosses of the value), they will reach out for enterprise features or support. This acts as a self-feeding funnel. It also means marketing spend can be more efficient (the product partly markets itself through community channels and word-of-mouth).

Additionally, we will explore **public sector opportunities**. Given the emphasis on open data and avoiding vendor lock-in, SGraph.AI is well positioned for government projects that require making data accessible and linked. We will seek grants or innovation programs in the EU, UK, and US that fund data integration or AI projects. Early engagement with such programs (even if not huge revenue) could both fund development and lead to high-visibility case studies (e.g., “City Government uses SGraph to link and open up public datasets”).

By the end of 24 months, we aim to have a healthy mix of: - Active SaaS/API subscribers (from small teams up to mid-sized companies). - A handful of large enterprise customers on annual licenses or support contracts. - A vibrant open-source community contributing or at least advocating for our tool. - Perhaps partnerships that embed SGraph’s tech in other products (the B2B2B model, e.g., a cybersecurity product that includes our semantic engine under the hood – and pays us per use or a licensing fee).

This sets the stage for rapid scaling in subsequent years, as we could then justify a larger Series A investment to hire a bigger sales team, further R&D, etc. (For now, we remain lean and focused on proving the model.)

Competitive Landscape and Differentiation

While our primary focus is executing on the opportunity, we recognize there are adjacent solutions in the market: - **Traditional Graph Databases (Neo4j, etc.):** They provide the graph storage and query engine, but they do **not** provide an easy way to transform raw text into a graph. In fact, SGraph can be seen as complementary – we could feed data *into* those databases. Our differentiator: we handle the *semantic extraction* and integration layer, which databases alone do not address. - **ETL / Data Integration Tools:** Big players (Informatica, Talend, etc.) help move and transform data between systems. However, they often focus on structured data and don't leverage AI for semantic understanding. They also tend to be proprietary and expensive. SGraph's AI-powered pipeline (with LLMs extracting meaning) is a new capability that traditional ETL lacks. Plus, we're open-source and flexible. - **Niche Semantic Tools and Startups:** There are some startups doing things like AI-based ontologies or knowledge management. Many are point solutions (for example, only for legal contracts analysis, or only for customer support FAQ mining). Our approach is more platform-like and agnostic to domain, which means we can apply to many use cases. Our open approach also differentiates from any closed SaaS offering – customers can own their solution. - **In-house Efforts:** Some large enterprises (or tech giants like Google) have built their own knowledge graphs. But most companies cannot afford the talent or time. SGraph offers a shortcut with out-of-the-box capabilities. Additionally, even those who built in-house might adopt SGraph if it's superior or more cost-effective than maintaining their own custom code – especially since it's open-source, they can contribute rather than start from scratch.

Overall, our **competitive edge** lies in the combination of **open, flexible technology** with a **focus on semantic transformation** using AI: - We are not just a graph database; we are the *engine that creates and maintains the graph* from messy data. - We are not just an NLP tool; we deliver structured outputs that plug into knowledge management processes. - We are not just consulting; we have a product foundation that makes solutions reproducible and scalable. - We alleviate **lock-in concerns** by being open, which many CIOs will appreciate (in contrast, they may be wary of getting stuck with a proprietary "AI insight" platform that they can't extend or move off of).

By continuously nurturing our open-source community and staying ahead in integrating the latest NLP/LLM advances (for accuracy of extraction), we intend to remain the go-to solution in this niche.

Traction and Milestones

Although SGraph.AI is a new venture, it builds on a strong foundation of R&D and even some initial usage via related projects: - The underlying technology components have been **developed and tested** in the context of the founder's other startups. For example, the semantic parsing and graph generation pipeline is already functional in MyFeeds.ai's MVP, which has delivered uncannily relevant personalized news briefings to test users [16] [17]. In one early pilot, a CISO received multiple tailored news feeds using this tech, and the results were well-received [18]. This validates both the technology and the appetite for semantic-driven content. - Another component, the **MemoryFS & GraphFS** storage system, has been proven to handle large volumes of data by storing it uniformly as graphs [12]. The LETS pipeline was used to process continuous streams of RSS feeds and demonstrated the ability to maintain performance in a serverless environment [19] (as seen in the semantic web content filter prototype). - We have one *co-design partnership* underway (hypothetical example): working with a cybersecurity solution provider to integrate SGraph for analyzing their incident reports. This is not a public customer yet, but it's a collaboration that helps shape our features (e.g., ensuring our graph outputs can easily link to Mitre ATT&CK threat ontology, which is a requirement in that domain). - Community interest: even prior to formal launch, the ideas behind SGraph (open semantic graphs, LLM integration) have attracted interest on social media and from participants in knowledge management forums where Dinis

has shared insights. We anticipate at launch having a few hundred followers or watchers keen to try the beta.

Moving forward, our key milestones over the next 12-18 months include: - **MVP Release (Q1 YYYY)** – Complete the initial version of the cloud API and open-source toolkit. This includes core text-to-graph conversion, a basic set of connectors (file upload, RSS feed loader, etc.), and documentation. Milestone success: at least 2 pilot clients using the MVP on their data, and ~100 developers signed up for API keys. - **Pilot Project Completions (Q2-Q3 YYYY)** – Successfully deliver at least 2-3 pilot use cases, demonstrating the platform in different contexts (e.g., one in cybersecurity, one in enterprise knowledge management). From these, secure at least one letter of intent or convert one into a paying customer. Milestone success: e.g., a paid contract for an on-prem deployment with a design partner client. - **Community Growth (ongoing)** – By mid next year, aim for the open-source project to have, say, 500+ stars on GitHub, 5+ external contributors, and an active chat/forum with user discussions. This will indicate healthy interest. - **Enterprise-Ready Features (Q3 YYYY)** – Add features needed for bigger customers: user management and access controls in the platform, more robust security reviews of the code (possibly an external audit, given our target clients), and scalability tests (process millions of documents in a batch to prove throughput). Achieve a stable v1.0 release. - **Revenue Generation (Q3-Q4 YYYY)** – Start converting usage into revenue. Target: $X in ARR (annual recurring revenue) by the end of the first year post-launch. This could come from a mix of sources (a couple of enterprise license deals, a dozen smaller SaaS subscriptions, and consulting engagements). The exact number is to be determined, but any revenue at this stage proves the business model viability. - **Team Growth and Next Round (Q4 YYYY)** – With technical validation and initial customers in hand, prepare for a larger fundraise (Series A). At that point, we'd highlight our user base, the open-source traction, and the initial revenues to justify the infusion to scale sales and marketing. An ideal milestone: raise a Series A at a strong valuation, enabling hiring of more engineers (for advanced features) and sales personnel to expand globally.

The above timeline is aggressive but achievable given our head start with existing tech. We will continuously iterate based on user feedback – agility is key. The company's structure (multiple parallel startups sharing tech) actually provides resilience: improvements in SGraph can flow into other products and vice versa [20], meaning we effectively have an **innovation feedback loop** across different use cases. This shared R&D accelerates reaching our milestones.

Team and Company Background

SGraph.AI is founded by **Dinis Cruz**, a seasoned technologist and entrepreneur with over two decades of experience at the intersection of software engineering, cybersecurity, and AI. Dinis has a strong track record of open-source contributions and has previously built innovative tools that gained industry adoption. Notably, he has been driving a multi-startup strategy focusing on GenAI-powered solutions for enterprise problems (board communication, content curation, etc.), all using a common open-source core [20]. SGraph.AI emerges from this ecosystem as the commercial catalyst to bring the semantic graph technology to a wider market.

The broader team (current and planned hires) includes experts in natural language processing, data engineering, and knowledge representation. We also leverage a **distributed network of contributors** and contractors: for instance, we have relationships with developers worldwide who have contributed to our open-source projects – a talent pool we can draw on for rapid project-based scaling (e.g., need to build a new connector or implement a client's custom request). This flexible resourcing strategy means SGraph.AI can punch above its weight, executing multiple projects in parallel by assembling ad-hoc teams (with lower overhead than maintaining a huge in-house staff early on).

Our ethos – "**open by design**" – not only applies to our code but also to how we work with customers. We aim to be transparent and collaborative. When engaging a client, we often co-develop solutions in the open (sanitizing any sensitive data) which benefits the community and the client alike. This approach has earned trust in prior projects and we will continue it under SGraph.AI.

Advisors: We are bringing on a few advisors familiar with scaling open-source startups and enterprise sales in the data space. They will guide our go-to-market refinement and introductions to potential clients/investors.

Culture: Internally, we emphasize agility, innovation, and community. Every engineer is encouraged to interact with users on GitHub or forums – this tight loop helps prioritize what features or fixes matter most. Given our multiple-venture background, the team is used to context-switching and knowledge sharing, which will serve us well as we juggle product development and custom projects.

Financial Plan & Investment Proposal

Financial Model: SGraph.AI plans to maintain a lean operation in the early stages, benefiting from the open-source, serverless approach which keeps infrastructure costs low [21] . Our primary expenses in the first year will be personnel (a small core team of engineers, plus some contractors for specific projects), cloud costs for the SaaS platform (scaling with usage, and largely passed on to customers in pricing), and modest marketing (mostly content and community events, rather than large ad spends). Thanks to open source, we incur **no licensing costs** for core technology, and our cloud architecture ensures we "pay for what is used" – which means if we onboard a big customer, the revenue from that customer covers the additional compute, keeping margins healthy.

We project that services (consulting) will constitute a significant portion of Year 1 revenue, helping fund development. By Year 2, subscription revenues (API and enterprise licenses) will grow as the product gains traction and more customers convert from pilots to full deployments.

Use of Funds: We are seeking an initial **seed investment** to accelerate our roadmap. The investment will primarily be used for: - Expanding the Engineering Team – hiring 2-3 additional developers with expertise in NLP/LLMs and cloud infrastructure to speed up feature development (e.g., building the user management portal, more connectors, optimizing the LLM inference costs). - Building the Sales/BD Foundation – possibly hiring one business development lead who can start reaching out to enterprise contacts and handling contracts, plus budget for attending one or two key industry conferences to showcase SGraph. - Launch Marketing – producing high-quality case study materials, videos, and developer documentation. Also running pilot programs (and absorbing their costs) as needed to secure referenceable success stories. - Cloud Infrastructure – scaling the cloud service for broader beta usage, which might involve a larger cloud hosting budget (though again, this will correlate with user activity). - General runway – ensuring we have ~18 months of runway to comfortably iterate product-market fit without revenue pressure forcing bad short-term decisions.

Investment Terms (Proposal): We are open to raising approximately **$X (for example, $1–2 million)** in seed funding, in exchange for an equity stake in the range of **10–15%** of the company. This valuation is justified by: - The advanced state of our technology (beyond prototype, with components in use and open-sourced). - The early interest and pilot engagements signaling market validation. - The founder's prior experience and the synergy with other initiatives (which de-risks execution). - The huge addressable problem of enterprise data integration for AI (positioning us in a high-growth segment).

The exact amount and terms can be adjusted based on investor discussions, but we envision this seed round carrying us through the product launch and initial revenue milestones, after which a larger Series A (at a higher valuation) could be raised once we demonstrate traction.

Our financial projections (conservative case) show that by the end of the seed runway, we could reach on the order of $500k–$1M in annualized revenue, stemming from a mix of 5-10 enterprise customers and hundreds of smaller SaaS users. Given the scalability of the SaaS model and the continued demand for data solutions, there is potential to grow to tens of millions in revenue within a few years thereafter, especially if we capture even a fraction of enterprises transitioning to knowledge graphs.

One noteworthy point for investors: the **capital efficiency** of our model is very high. We don't require massive upfront capital to build datacenters or develop proprietary algorithms from scratch – we stand on open source and commodity hardware. Each new customer or user increases our value with relatively marginal cost increase (especially for the cloud service). This means investment dollars go primarily into acquiring users and building features that widen our lead. In early tests, we have been able to support initial users on "cheap cloud functions" with costs easily covered by what even a small subscription would be [21]. This efficiency, combined with the open-source adoption engine, gives SGraph.AI a favorable burn rate and the ability to potentially reach breakeven with less capital than a typical enterprise software startup.

We will keep the pricing model flexible as we learn (perhaps experimenting with per-document vs. per-seat or flat license fees for on-prem). The **goal is to optimize lifetime value while keeping customer acquisition friction low**.

Risks and Mitigations

No business plan is complete without acknowledging risks: - **Technical risk:** Can our AI reliably extract correct semantic graphs at scale? Mitigation: we use state-of-the-art LLMs and will invest in fine-tuning them on domain data as needed. We also keep a human-in-the-loop option for high accuracy needs (e.g., a review interface for a client's experts to validate the ontology or extracted facts, which then feeds back to improve the model). Our focus on provenance (keeping raw data and steps) helps debug and improve quality continuously. - **Open-source dynamic:** There's a risk that competitors (or big cloud providers) use our open-source tech and offer it as a service, undercutting us. Mitigation: We will build a strong brand and first-mover advantage in this niche. Our service quality and expertise will outshine a pure code copy. Additionally, we could consider an open-core model where certain advanced features are only in our enterprise edition, though for now the plan is to open-source nearly everything and compete on service. - **Market adoption risk:** Enterprises might be slow to embrace knowledge graphs or see it as a "nice-to-have." Mitigation: We tie our solution to pressing needs (AI initiatives, compliance, data governance). We will educate the market with clear ROI examples. Also, we aren't solely reliant on big enterprise adoption – the API model allows even small firms or individual developers to use SGraph for their own innovative applications, giving us multiple shots on goal. - **Talent and execution:** The fields we operate in (AI, data engineering) are competitive for talent. Mitigation: Our open-source stance actually attracts talent (engineers often want to work on things that have community impact). We also can tap global contributors rather than hire only Silicon Valley salaries. The founder's ability to inspire through vision (already demonstrated by multiple projects) is a key asset to bring great people on board.

Conclusion

SGraph.AI is poised to become a **key enabler of the semantic data revolution** that enterprises must undergo to fully leverage AI. By focusing on open standards and knowledge graphs, we position ourselves not just as a product vendor but as a **trusted guide** in our customers' data transformation journey. Our business plan builds on a solid technological base, multiple clear monetization streams, and a go-to-market that reduces risk by starting with real user needs and scaling up organically.

In the next 5 years, we envision SGraph.AI powering the knowledge layer in many organizations – akin to how databases or data lakes are standard today, a **semantic graph layer** will become essential tomorrow, and SGraph will be the open yet enterprise-ready choice to implement it. Executives will gain unprecedented visibility into their organization's knowledge, AI systems will perform better with a rich context to draw on, and even day-to-day operations (like finding information or ensuring consistency across documents) will be improved. All of this is achievable while **keeping the customer in control of their data**, thanks to our open approach.

We invite partners and investors to join us in this journey. With the seed investment and our team's drive, SGraph.AI will capture this opportunity to not only build a successful company but also foster an ecosystem where data truly becomes knowledge. The transition to semantic graphs is coming – our mission is to make it accessible, scalable, and beneficial to all parties involved.

By executing this plan, SGraph.AI will transform scattered information into actionable intelligence for businesses worldwide, turning the dream of interoperable, intelligent data into reality.

Sources:

- Cruz, D. *Multi-Startup Strategy: Open-Source Innovation Across Four Synergistic Ventures* – Technical overview of shared AI/graph technology across Dinis Cruz's startups [7]. Contains descriptions of the semantic pipeline (LETS) and benefits of open-source approach for enterprise adoption [1].
- *MyFeeds.ai Investor Brief* – Describes the use of semantic knowledge graphs for personalized content feeds, validating SGraph's core concept. Notably, it highlights how graph-based recommendations improve relevance and explainability [15] [13] and details the pipeline of ingesting and transforming data into a graph format [9] [10].
- Voice Memo (Founder's Strategy Discussion) – Outlines SGraph.AI's business strategy in the founder's own words, including the emphasis on open standards, potential revenue streams (cloud service, on-prem licenses, consulting), and the importance of helping customers discover their ontologies. It also introduced the concept of using the transformation pipeline as a form of backup/disaster recovery, and suggested an incremental go-to-market approach focusing on early pilots and community building (content from this memo has been integrated throughout the plan).

---

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] Dinis Cruz's Multi-Startup Strategy_ Open-Source Innovation Across Four Synergistic Ventures.pdf
file://file_00000000a86c71f49f64dfb7ff917724