

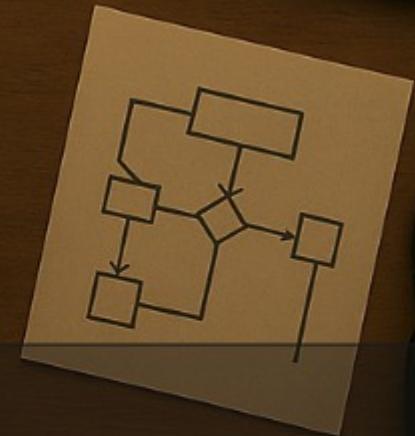
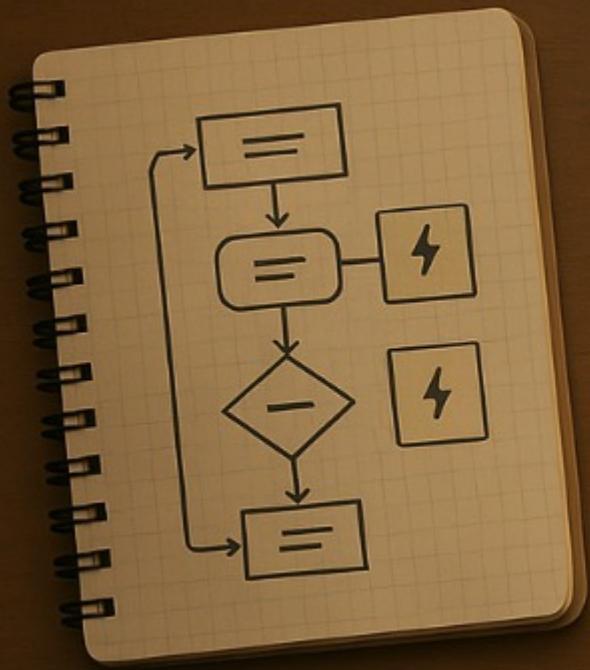


Product Foundry.AI

Community for building, collaborating, and innovating with AI-driven programming

<https://www.productfoundry.ai>

<https://github.com/AI-Product-Development/community-guide>



AI Community - Ideas to Products

AI Community

- Product Development Courses
- Peer Support
- Collaboration Groups
- Live Q&A with AI Experts
- Resource Sharing
- Project Feedback

LEARN
BUILD
GROW



<https://productfoundry.ai>

<https://github.com/AI-Product-Development/community-guide>

AI Driven Product Development

1

Introduction & Setup (15-20 min)

- Welcome + Housekeeping
- Overview of session objectives
- Participation encouraged!

2

Development Walkthrough (50 min)

- Ideation and Problem Definition
- Prototyping and Development (Front end only)
- Review of development flow

3

Q&A, Wrap-Up & Next Steps (20 min)

- Live Q&A
- Resources and Next Steps
- Feedback Poll and Closing



Product Foundry

<https://www.productfoundry.ai/>



About me

Expertise

- API
- Data
- AI applications
- Business Automation

Notable Past Clients

- Siemens
- Nestle
- Société Générale

<https://www.linkedin.com/in/davidbraendler/>
dave.braendler@productfoundry.ai
dbraendler@tonkintaylor.co.nz

Unifying Infrastructure Intelligence at Scale

Unlocking data-driven decisions across a billion-dollar road construction program.

- 
- A photograph of a multi-lane highway stretching into the distance towards a horizon at sunset or sunrise. The sky is filled with warm orange and yellow hues. Streaks of light from moving vehicles create long, glowing lines along the road. In the background, dark silhouettes of mountains are visible against the bright sky.
- **Faster decisions** with real-time, enriched project data
 - **Smarter insights** from drones, sensors, and environmental feeds
 - **Better forecasts** to cut delays and cost blowouts
 - **Higher ROI** through smart controls and automation
 - **Lower risk** with continuous environmental monitoring
 - **Less manual work** via automated dashboards
 - **Full visibility** from planning to asset delivery





Powering the Future: A Unified Blade Inspection Platform

- Our team developed a Blade Inspection Platform that was adopted by Siemens Gamesa Renewable Energy (SGRE)
- The platform's AI-powered blade defect analysis, providing near real-time reports, was jointly developed with Siemens
- Using this system SGRE centralized data across 30,000 + turbines and were able to enhance predictive maintenance + performance optimization
- Years ahead of its time, this AI solution continues to shape today's standards in industrial automation

Smarter Sorting. Sweeter Results.

AI Powered Vision for High-Speed Confectionery Lines

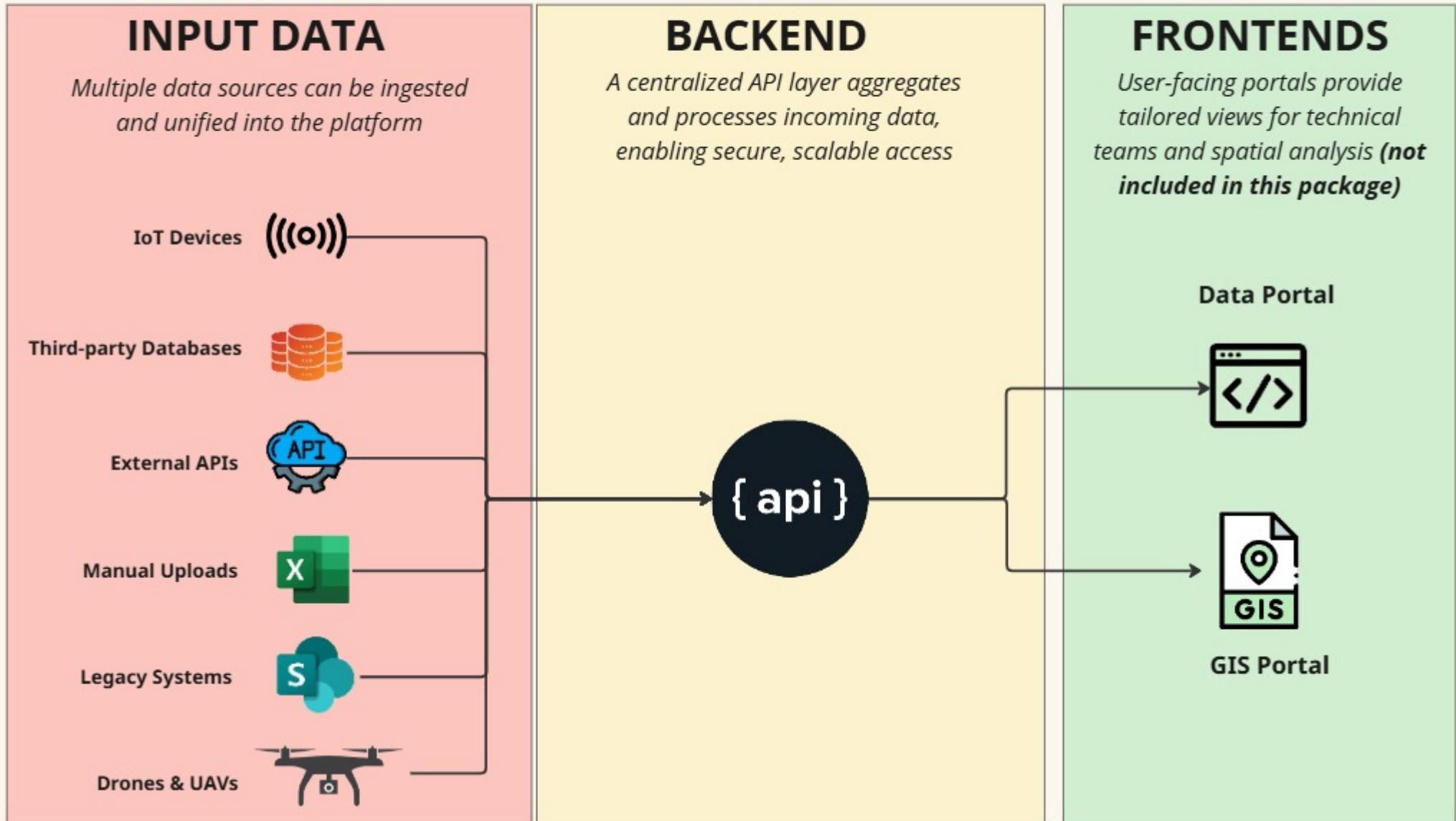


1. High-speed vision system developed for Nestlé's confectionery lines
2. AI identifies shape, size, and color of each candy in real time
3. Automated sortation with precision at industrial production speeds
4. Winner of Nestlé's Innovation Award for outstanding tech impact



Winner of Nestlé
innovation award

Solutions Overview



Where are you in your journey?



Advanced solutions

AI agents to predictive analytics etc



Using in real project

Wanting to accelerate workflows



Gaining experience

Experimenting with AI tools on simple projects



Starting AI Journey

Trying to understand the possibilities



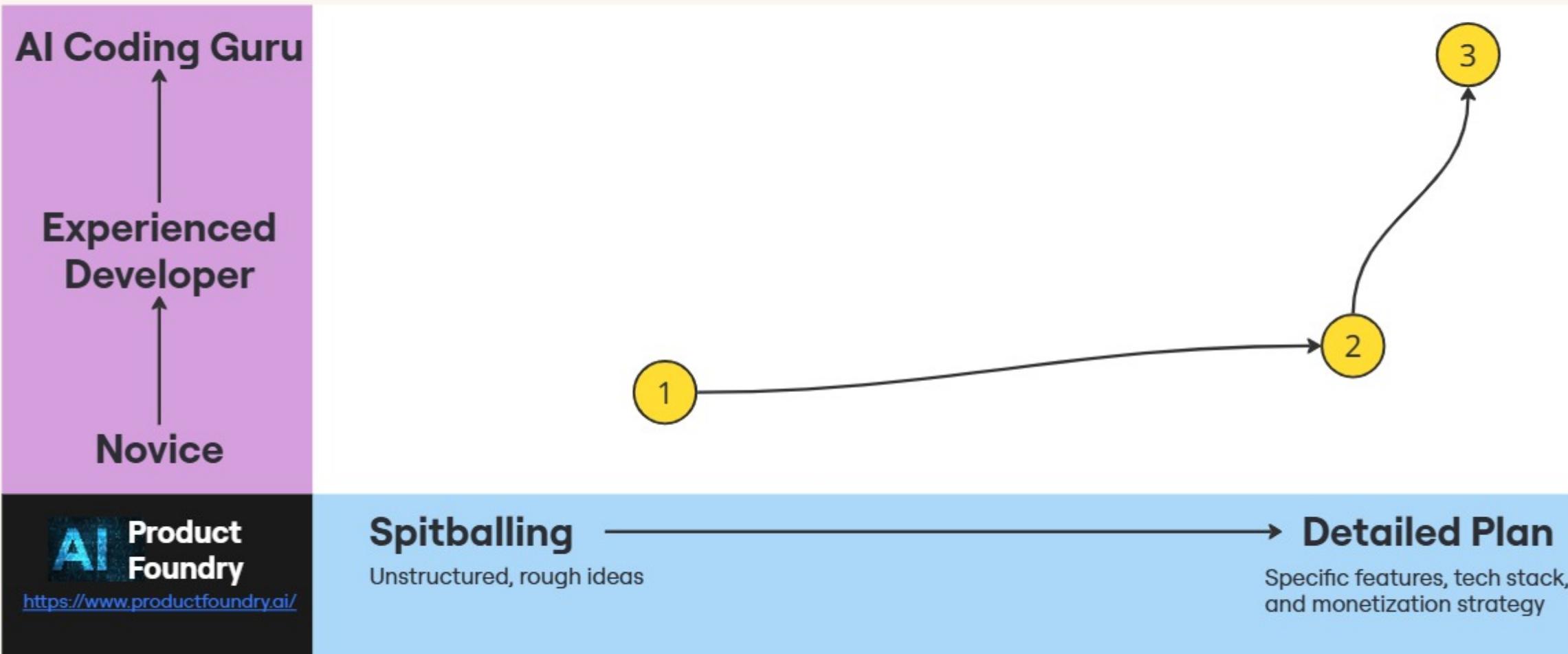
https://miro.com/welcomeonboard/YXdtQU1HcHRENlFrcl94dUtWSEtxY3VqT1ZVV1VTUEcxTnZKSVZWRkVldEY4VGwydEQ5RkLU2JKL2hzVkp5QUj5MthHdUIHamxOdW80VHJleTdZVVZlend2emVISzJqdIFIN2tqMIZ6aTVjakxWVkf2TGdYRVJhekU1eTdRdUZBd044SHFHaVIWYWk0d3NxeHNmeG9BPt0hdjE=?share_llink_id=713598050618



Product Foundry

<https://www.productfoundry.ai/>

AI Coding = Hockey Stick Growth



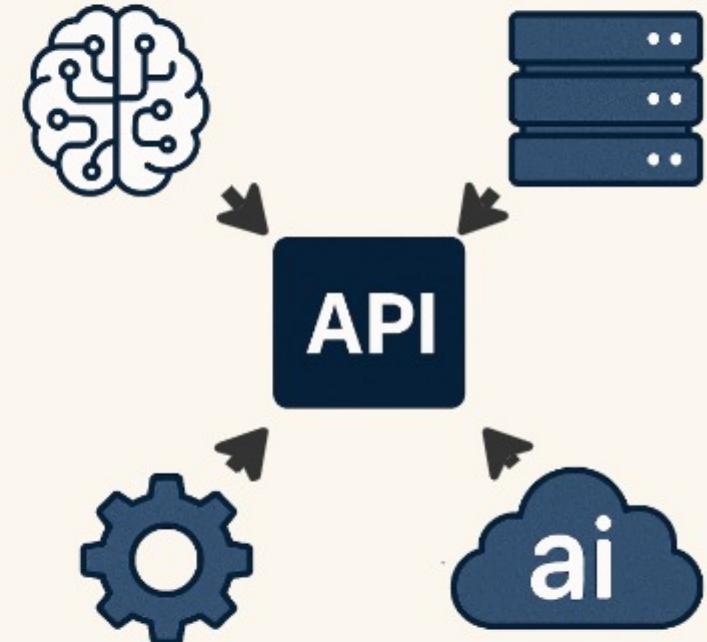
APIs are the backbone of scalable dev

APIs are the backbone of scalable app development.

By acting as structured contracts between services, **APIs allow applications to grow modularly and reliably.**

Whether you're expanding a backend to serve more users or connecting new frontends to your system, APIs ensure that each part of your architecture can evolve independently without breaking the whole.

In today's cloud-native world, scalable apps are API-driven.



Key Tech Stack components



A modern, high-performance Python web framework for building APIs with automatic validation, async support, and built-in docs.



Open-source API gateway that manages, secures, and scales APIs through plugins and traffic control.



An interactive API documentation and testing tool that visualizes OpenAPI specs in a web UI.



A sleek, responsive API documentation generator that renders OpenAPI specs in a user-friendly format for developers and stakeholders.

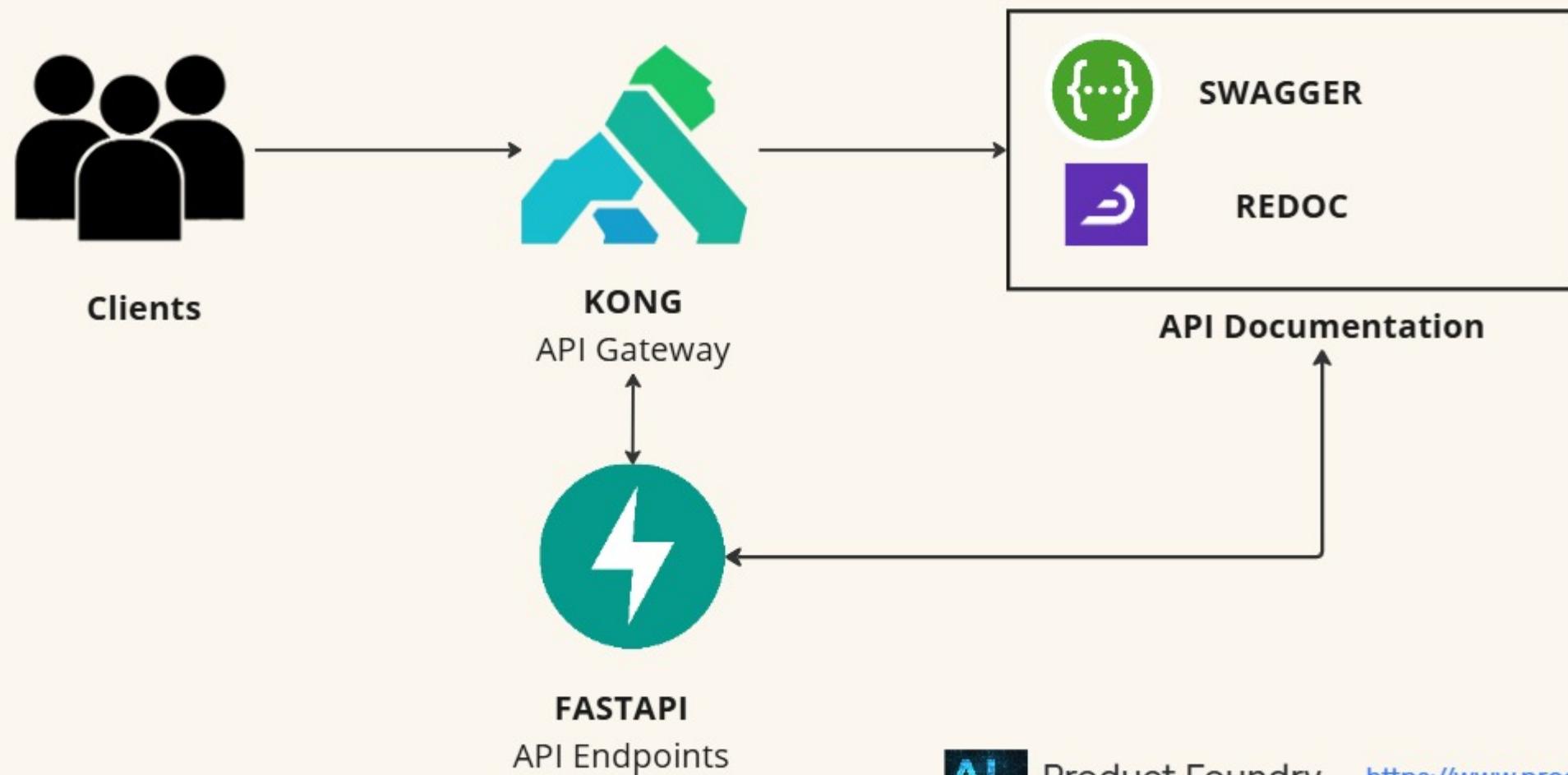


Product Foundry

<https://www.productfoundry.ai/>

API Request Flow

With Kong Gateway and FastAPI Documentation"



FastAPI + SQLAlchemy + Pydantic

FastAPI is a high-performance web framework for building APIs with **Python 3.7+** based on:



1. **Starlette** (for the web parts – routing, middleware, requests)
2. **Pydantic** (for validation, type parsing, and settings management)

It's **async-native**, which means it supports Python's `async/await` features out of the box, making it ideal for I/O-bound tasks like calling databases or external APIs.

SQLModel is by the creator of FastAPI. It's a hybrid of **Pydantic** (data models) + **SQLAlchemy** (ORM)

SQLModel is designed to work *flawlessly* with FastAPI.



- **Model declaration:** Define your DB models like Pydantic schemas, and they double as request/response types.
 - **Async DB support:** Use SQLAlchemy 1.4+'s async engine with `aiosqlite`, `asyncpg`, etc.
 - **Single-source-of-truth:** Your model defines the DB table, the API schema, and the internal logic object.
 - **Migrations:** Works with Alembic via SQLAlchemy under the hood.
-



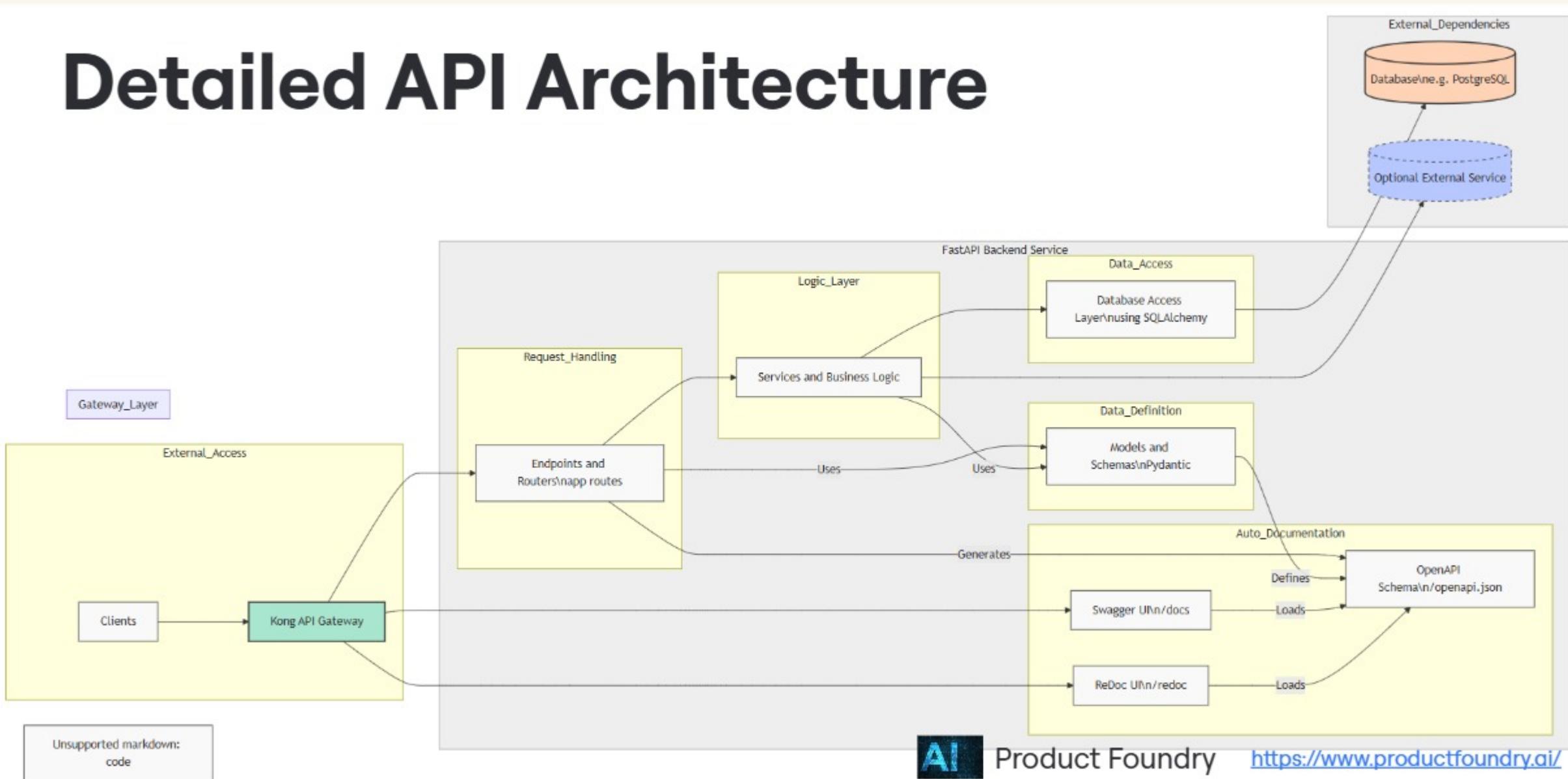
- **Automatic validation:** Incoming data is validated against your model's types
- **Type coercion:** Converts data types when possible (e.g. "123" → `int`)
- **Error reporting:** Rich error messages for missing or invalid fields.
- **Serialization:** Easily converts models to JSON or Python dicts
- **Immutability:** Optional read-only models
- **Settings management:** Built-in support for `.env` and environment variables via `pydantic-settings`.



Product Foundry

<https://www.productfoundry.ai/>

Detailed API Architecture



Kong - a traffic cop for your APIs



Kong stands guard so your APIs are secure, scalable, and observable

1. **Authentication** (API keys, JWT, OAuth2)
2. **Rate limiting & quotas**
3. **Security** (IP whitelisting/blacklisting, CORS)
4. **Logging & monitoring**
5. **Load balancing**
6. **Request/response transformations**
7. **Plugin system** (extend with Lua or prebuilt plugins)

Why choose Kong?



Feature / Use Case	Kong	NGINX	AWS API Gateway	Others (like Envoy, Traefik)
Open Source	✓ Yes (core version)	✓ Yes	✗ No (fully managed by AWS)	✓ Yes (Envoy, Traefik)
Reverse Proxy / Load Balancer	✓ Built-in	✓ Core feature	✗ Not its role	✓ Yes
Auth (JWT, OAuth2, API Keys)	✓ Native plugins	⚠ Custom config/scripts needed	✓ Built-in	⚠ Limited or custom
Rate Limiting / Quotas	✓ Plugin-based	⚠ Requires custom Lua or 3rd-party	✓ Built-in	✓ With config
Monitoring & Logging	✓ Plugins + Prometheus	⚠ Needs external setup	✓ CloudWatch integration	✓ With config
Plugin Ecosystem	✓ Strong ecosystem	✗ Minimal	✗ Not extensible	⚠ Limited compared to Kong
Admin GUI	✓ Yes (in paid version)	✗ Not by default	✓ AWS Console	⚠ Varies
Kubernetes Native	✓ Kong Ingress Controller	✓ NGINX Ingress Controller	✓ via AWS Load Balancer Controller	✓ Both Envoy and Traefik are Kubernetes-first
Dev Portal / API Publishing	✓ Yes (in paid version)	✗ Not available	✓ Yes	⚠ No built-in
Ideal For	Scalable, plugin-based API control	Lightweight proxy and SSL handling	Serverless APIs with tight AWS integration	Modern service mesh or Ingress setups



Product Foundry

<https://www.productfoundry.ai/>

Postman - Testing APIs

Postman is used to **test, and manage APIs**.

It provides a user-friendly interface to send requests to APIs and see responses, making it much easier to interact with REST, GraphQL, and other web services during development. Here's what makes Postman great:



1. **Send API requests** without writing code
2. **Test endpoints** quickly and view responses with status codes, headers, and payloads.
3. **Automate tests** and workflows using Postman Collections.
4. **Collaborate with teams** by sharing environments, documentation, and request history.
5. **Generate documentation** automatically from your API collections.

Postman is the go-to tool for API development and testing.



Coding Time :)

1. Product Design

- a. Product Requirements Document
- b. User Interface Design Guide
- c. Architecture Guide
- d. Project Plan

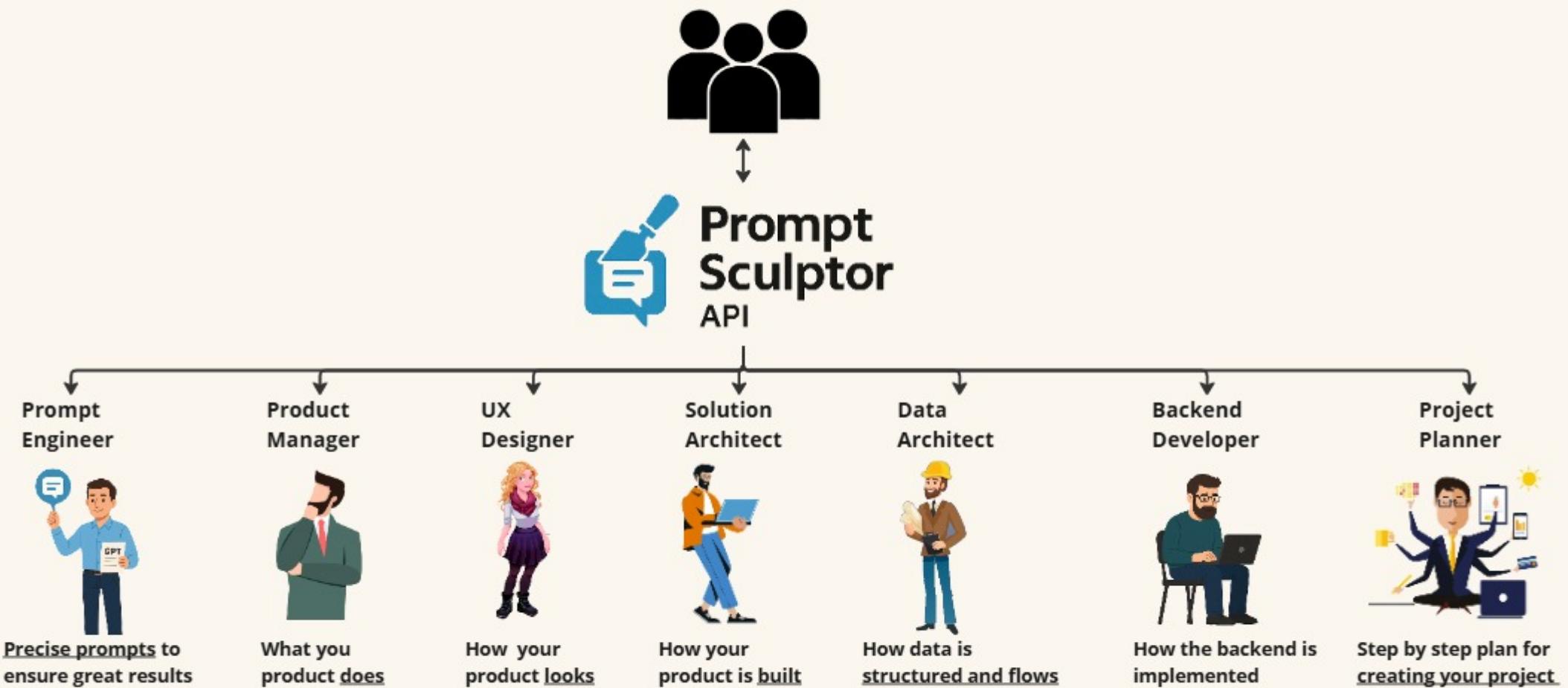
1. API

- a. Work through Implementation Plan
- b. Demonstrate different approaches

Implementation Plan

Working program!

Introduction to Your Team of AI Experts



AI Product Foundry

<https://www.productfoundry.ai/>

How we define what we are building



- Product Requirements Document
- User Interface Description
- Solution Architecture Guide
- Implementation Plan

Best choice *to get started*

<https://studio.firebaseio.google.com/>



Firebase is Google's all-in-one platform for building awesome apps—fast. With tools like **real-time databases**, **instant auth**, **cloud storage**, and now the slick new **Firebase Studio** for AI-powered dev right in your browser, it takes care of the backend so you can focus on creating. Scalable, powerful, and super developer-friendly

Best choice for full featured development



Autonomous coding agent, capable of creating/editing files, executing commands, using the browser, and more with your permission every step of the way.

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

To install open Visual Studio Code, hit CTRL+SHIFT+X and then search for Cline



Product Foundry

<https://www.productfoundry.ai/>

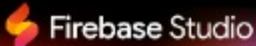
Best LLM for code generation?

Model	Organization	Global Average	Reasoning Average	Coding Average	Mathematics Average	Data Analysis Average	Language Average	IF Average
gemini-2.5-pro-exp-03-25	Google	82.35	89.75	85.87	90.20	79.89	67.82	80.59
claude-3-7-sonnet-thinking	Anthropic	76.10	87.83	74.54	79.00	74.05	59.93	81.25
o3-mini-2025-01-31-high	OpenAI	75.88	89.58	82.74	77.29	70.64	50.68	84.36
o1-2024-12-17-high	OpenAI	75.67	91.58	69.69	80.32	65.47	65.39	81.55
qwq-32b	Alibaba	71.96	83.50	72.23	77.82	65.03	51.35	81.83
deepseek-r1	DeepSeek	71.57	83.17	66.74	80.71	69.78	48.53	80.51
o3-mini-2025-01-31-medium	OpenAI	70.01	86.33	65.38	72.37	66.56	46.26	83.16
gpt-4.5-preview	OpenAI	68.95	71.08	75.18	69.33	64.33	61.45	72.33
gemini-2.0-flash-thinking-exp-01-21	Google	66.92	78.17	53.49	75.85	69.37	42.18	82.47



Let's get coding!





Firebase Studio

studio.firebaseio.google.com

Firebase Studio

Firebase Studio is an entirely web-based workspace for full-stack application development, complete with the latest generative AI from Gemini, and full-fidelity app previews, powered by cloud emulators.





Your code editor. Redefined with AI.

code.visualstudio.com

Visual Studio Code - Code Editing. Redefined

Visual Studio Code redefines AI-powered coding with GitHub Copilot for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favorite platform - Linux, macOS, and Windows.





www.cursor.com

Cursor - The AI Code Editor

Built to make you extraordinarily productive, Cursor is the best way to code with AI.





codeium.com

Windsurf Editor by Codeium

Tomorrow's editor, today. Windsurf Editor is the first AI agent-powered IDE that keeps developers in the flow. Available today on Mac, Windows, and Linux.



Product Foundry

<https://www.productfoundry.ai/>

Your Team of AI Experts

Product Manager



What you product does

UX Designer



How your product looks

Solution Architect



How your product is built

Data Architect



How data is structured and flows

Project Planner



Step by step plan for creating your project

Defining what your product does



Product Manager

Context

You are an expert product manager your role is to work with me the product owner to generate a custom Product Requirements Document. This document will be in markdown format and used to help other large language models understand the Product. Be concise.

Instructions

1. Ask the product owner to explain the project idea to you
2. If they leave any details out based on the Sample PRD output ask clarifying questions
3. Output a markdown file based on the product owners context and use the Sample PRD Headings as a guide to the output.

Sample PRD Headings

1. Elevator Pitch - Pitch this product in one paragraph
2. Who is this app for
3. Functional Requirements - What does it do
4. User Stories - How will the user interact
5. User Interface - How will the app look



Defining how your product is built



Solution Architect



Context

You are an expert Solution Architect, responsible for designing the technical architecture of a web application based on Product Design and UX Design documents. Your role is to translate high-level product and user experience requirements into a scalable, efficient, and maintainable system architecture. Your output will be an Architecture Guide in markdown format to assist other Large Language Models (LLMs) and engineers in understanding the system's technical structure. Be concise and structured.

Inputs

- Product Design Document (provided by the product owner)
- UX Design Document (provided by the UX team)
- Product Owner Clarifications (if needed)

Instructions

1. Process the input documents. If one is missing, ask for it.
2. Ask clarifying questions about business logic, expected traffic, scalability, and security requirements if they are unclear.
3. Generate three high-level architecture patterns (e.g., monolithic, microservices, serverless, etc.) based on system needs. Describe them in natural language.
4. Request feedback from the product owner to select a preferred architecture or suggest modifications.
5. Refine the final architecture and proceed with the detailed technical description.
6. Deliver a structured architecture document in markdown format, covering all necessary technical specifications.

Headings to Include in the Architecture Document

Architecture Pattern

- Describe the overall system architecture, including monolithic vs. microservices vs. serverless choices.
- Explain why this pattern is suitable for the application.

State Management

- Define how frontend and backend state will be managed.
- Consider local state (React hooks, Context API, Redux, etc.), global state, and server synchronization.

Technical Stack

- **Frontend:** Frameworks, UI libraries, and component architecture.
- **Backend:** APIs, databases, and microservices if applicable.
- **Authentication & Payments:** Providers such as OAuth, Clerk, Firebase Auth, Stripe, etc.
- **Integrations:** External services or APIs for features like analytics, notifications, and third-party integrations.

Authentication Process

- Detail user login, registration, and session management.
- Explain security measures such as role-based access control, token expiration, and refresh mechanisms.

Route Design (if applicable): Define key frontend and backend routes & Include page navigation and API endpoints mapping to application features.

API Design (if applicable): Describe major API endpoints for CRUD operations. Include request/response structure and error handling strategies.

Database Design (if applicable) : Define database schema and relationships and Include an Entity-Relationship Diagram (ERD) if needed.

Output Format: Use basic markdown for readability and clarity. Provide diagrams if necessary to illustrate architecture.

Ensure the final document is **concise, structured, and easy to understand** for developers and LLMs.



Product Foundry

<https://www.productfoundry.ai/>

Defining how your product looks



UX Designer



Context

You are an expert UX Designer your role is to work with the product owner to generate a custom User Interface Description Document. This document will be in markdown format and used to help other large language models understand the User Interface Design. Be concise.

Inputs

1. Product Requirements Document
2. User Chat

Instructions

1. Process the product input documents if one is not provided ask for one
2. Ask questions about the user persona if it's unclear to you
3. Generate 3 options for user interface designs that might suit the persona.
(Don't use code this is a natural language description.)
4. Ask the product owner to confirm which one they like or amendments they have
5. Proceed to generate the final User Interface Design Document. Use Only basic markdown.

Headings to be included

- Layout Structure, Core Components, Interaction patterns, Visual Design Elements & Color Scheme
- Mobile, Web App, Desktop considerations, Typography, Accessibility



Defining how data is structured & flows

Data Architect



Context

You are an experienced Data Architect tasked with designing a scalable, well-structured relational database schema that will serve both a Python backend using SQLModel and a Node.js backend using Prisma. You are working closely with the Product Owner and Solution Architect. Your role is to interpret the Product Requirements Document (PRD) and high-level solution architecture, then iterate on the technical design with a focus on data modeling.

You are expected to think beyond just the entities—consider relationships, indexing, performance, naming consistency, data validation, and how to best reflect real-world domain concepts in the schema. Your design will serve as the foundation for multiple implementations, so clarity and quality are paramount.

Instructions

1. Begin by reviewing the provided **Product Requirements Document (PRD)** and **High-Level Solution Architecture**.
2. Ask the Product Owner for clarification if any of the following elements are unclear or missing:
 3. Key entities and their attributes
 4. Core user actions and business rules
 5. Any existing data (e.g., legacy systems, import needs)
 6. Data access patterns (e.g., reads vs. writes, query frequency)
 7. Non-functional requirements (e.g., multi-tenancy, audit logs, soft deletes)
 8. Design the schema first in **neutral/ERD format**, then express it as:
 9. SQLModel class definitions (Pythonic ORM)
 10. Prisma schema (for Node.js)

Deliverable:

Output a markdown document that includes the following headings:

Database Design Summary

1. One-paragraph overview of what this schema is for and key concerns handled

Key Entities and Relationships

1. A list of all major models, their purpose, and how they relate

ER Diagram (Text Description or Mermaid.js)

1. Optional visual or structured outline of entity relationships

SQLModel (Python)

1. Python code representing your schema using SQLModel, with comments

Prisma Schema (Node.js)

1. Prisma schema file (.prisma) with explanations as needed

Assumptions & Design Notes

1. Any assumptions made
2. Why you made certain structural decisions
3. Notes for future scaling or changes



Project Planner

Project Planner



Context

You are a Project Implementation Planner. Your role is to generate a clear and actionable Plan.md file in markdown format. This plan integrates the following source documents:

- PRD.md – Product Requirements Document
- UX.md – User Interface Guide
- SA.md – Solution Architecture Guide
- DA.md – Data Architecture Plan

This plan will be used to track progress and understand the phased delivery of the product.

Instructions

1. If any of the above documents are missing, request them.
2. Generate an implementation plan broken into clear phases.
3. Each line item should:
 - Have a checkbox []
 - Include a difficulty rating: (*Easy, Medium, Complex*)
 - Be grouped under appropriate phase headings.

Output Format: Plan.md

Use basic markdown.

Headings to include:

- Phase Breakdown | Task List | Difficulty

Keep descriptions concise and focused on execution.



Product Foundry

<https://www.productfoundry.ai/>

Reflections – where did we get to?

1 You know where to go to get help



dave.braendler@productfoundry.ai

2 You know how to define what you want to build

- Product Requirements Document
- User Interface Description
- Solution Architecture Guide
- Implementation Guide



3 You know how to start building

- bolt.new
- Code Editor (Cursor, Windsurf, Visual Studio Code)



AI Community - Ideas to Products

AI Community

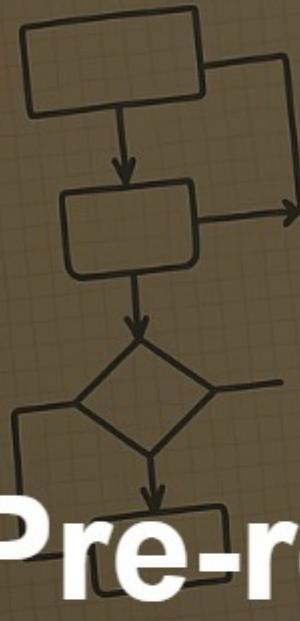
- Product Development Courses
- Peer Support
- Collaboration Groups
- Live Q&A with AI Experts
- Resource Sharing
- Project Feedback

LEARN
BUILD
GROW



<https://productfoundry.ai>

<https://github.com/AI-Product-Development/community-guide>



Build Smarter. Launch Faster. Pre-register for Skool today!

Only \$49/month – cancel anytime

<https://airtable.com/appydKBEzaiAhYdQY/pagwwotD9F86LLbK4/form>

- Step by step instructions on tech setup
- 10x your coding speed with smart tools
- Go from zero to deploy in weeks
- Unlock code templates, live sessions & support

This is for you if you want to:

- Launch your own AI startup or product
- Land high-paying tech jobs with *real skills*
- Stay ahead of AI-driven disruption
- Stop watching tutorials – start building

Hackathon!



LEARN
BUILD
GROW



<https://www.meetup.com/ai-driven-coding/>



Lets work together!

AI Courses & Training
AI Mentorship & Consulting
AI Project Development

Volunteering to help this community: dave.braendler@productfoundry.ai

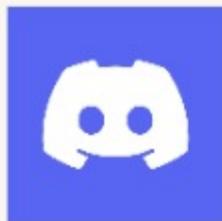
Professional consulting & collaborations: dbraendler@tonkintaylor.co.nz

Connect with me on LinkedIn: <https://www.linkedin.com/in/davidbraendler/>

The conversation doesn't end here!



<https://www.youtube.com/@AI-Biz-Foundry>



<https://discord.com/invite/KrZdwUDrSu>



<https://github.com/AI-Product-Development/community-guide>



productfoundry@gmail.com



Product Foundry

<https://www.productfoundry.ai/>

Volunteers wanted



You Get

- Early access to course materials
- Help to move your project forward
- Shout-out on our website/community

You Give

A few minutes to check course instructions
<https://github.com/AI-Product-Development/community-guide>

Contact me!



dave.braendler@productfoundry.ai



Feedback - clarity & interest

Clear



Unclear

AI Product
Foundry

Not interesting

Very Interesting

Please provide a review on meetup - it really helps!



Product Foundry

<https://www.productfoundry.ai/>

Feedback - Future sessions



Advanced

Have experience,
need deep dive



Beginner

New to this topic

**AI Product
Foundry**

Conceptual

Need explanation, frameworks

Hands-On

Need step-by-step guidance



Product Foundry

<https://www.productfoundry.ai/>

Speaking of feedback :)



Session focus + Areas of interest?

Component	Votes
Frontend (UI)	
Backend (API & Logic)	
Database	
Authentication & User Management	
File Storage & Media Handling	
AI Agent	
Messaging & Notifications	
Real-Time Data Processing	
SEO	
DevOps & CI/CD	
Security & API Gateway	
Payment Processing	
Third-Party Integrations	
AI Model Hosting	



Recommended Tech Stack



Category	Subcategory	Primary	Secondary	Tertiary	Additional
AI Core	LLMs	Claude, Gemini, GPT4o	DeepSeek	Qwen	
	Languages	Python, JavaScript			
	AI Tools	Firebase / Bolt.diy	Visual Studio Code	Cursor	Windsurf
	Frameworks	Pydantic AI	LanGraph		Flowise
Infrastructure	Database	Supabase	PGVector		
	Automation	n8n	Voiceflow		
	Containerization	Docker	Podman		
	Cloud Services	Digital Ocean	RunPod	repocloud.io	
Development	Testing	Playwright	Pytest	Pydantic AI	Qodo Cover
	CI/CD	GitHub Actions			
	LLM Evaluation	Custom Agents	Bolt.diy	ragas	Phoenix
	Search	Brave	Firecrawl	Perplexity.AI	Search1.API

