

Endless Gameplay by Design

Schema-Driven AI for Playable Systems

Scott Bromander



A Quick Note on AI



Hi, I'm Scott

Goal of this Talk

“To understand **Structured Data Outputs (SDOs)**,
to add to your design tool kit.”

DEMO TECHNOLOGIES



Web Based

Easier to share within a Conference setting. Applications can connect to web based services however, so the client application is not as important. But internet is required to leverage this technology.



OpenAI APIs

There are other options, but OpenAI's API is easier to understand at a glance, which is great for teaching.



Python/FastAP

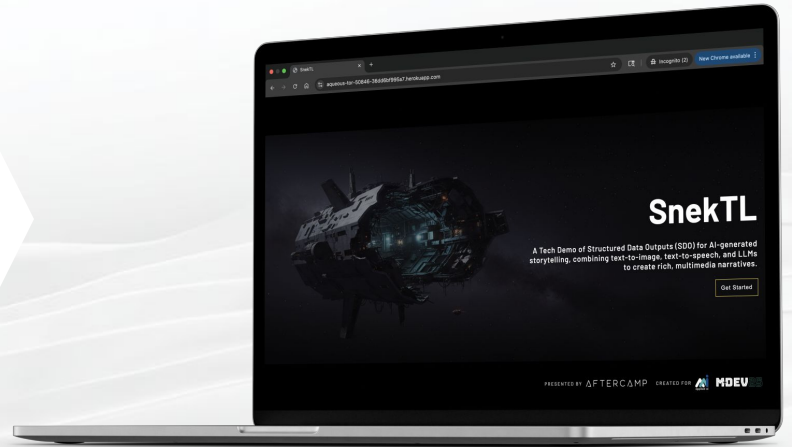
Most of the 3rd party SDKs are written in Python. FastAPI is a great backend choice for rapidly developed prototypes.



React

Front End presentation layer written in JavaScript's most popular framework.

Demo - SnekTL



Goal of this Talk

“To understand **Structured Data Outputs (SDOs)**,
to add to your design tool kit.”

Basic LLM Usage



Input



LLM



Output

Basic LLM Usage



Input

We enter some text...



LLM

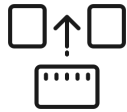
The Large Language Model thinks about it...



Output

Typically some kind of text comes back.

This is the regular user pattern in ChatGPT, Gemini, etc.



RAG

Retrieval Augmentation Generation

Retrieval Augmentation Generation,
“RAG” is about retrieving task-relevant
info from any store and augmenting the
model’s prompt with it before generation.

Basic LLM Usage



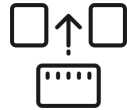
Input



LLM



Output

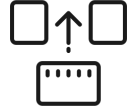


RAG Allows us to modify an Input

Basic LLM Usage



Input



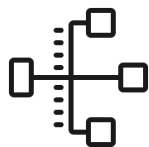
LLM



Output

But what about modifying Output?





Structured Data Output (SDOs)

Structured Responses from our LLMs

Structured Data Outputs are AI outputs formatted in a predictable structure, instead of free-form text. This format makes it easy for applications to reliably extract and use information without guessing or parsing natural language.

By returning data in a defined schema, AI systems can seamlessly integrate with other tools, APIs, and databases.

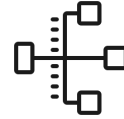
What's the Difference?



LLM

"Describe Scott and what he's doing."

"Scott Bromander is the presenter of this talk at Applied AI Conf. He is 6 ft tall and is an AI consultant."



SDO

"Describe Scott and what he's doing."

name : "Scott Bromander",
title : "AI Consultant",
height: "6 ft",
current : "Presenter",
location : "Applied AI Conf"

To be clear, this is something different than the “Old Way”



Input



LLM



Output



OLD WAY

Prompt, Pray, and Parse

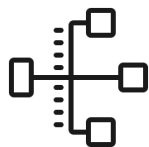
“Please, please, please
give me JSON.”



NEW WAY

(Schema w/SDO)

“Give me exactly this.”
(please.)

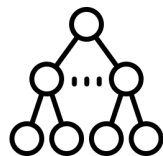


Structured Data Output (SDOs)

Structured Responses from our LLMs

As application creators, we can use AI responses in the applications we design.

We can shape the data from those responses. We do this by defining a “Schema”.



Schema

Form for Function

A Schema is a concise blueprint that defines the fields and types an AI (or any system) should return, so apps can use the data reliably without guessing.

Developers define the Data “Shape” that the AI should fit the response into. These are made of groups of defined **Data Types**.

DATA TYPES IN A NUTSHELL

1.0 NUMBERS

Floats, Doubles, Integers,
and more.

“Hello” STRINGS

Character, Dates, UUIDs,
Groupings of Character

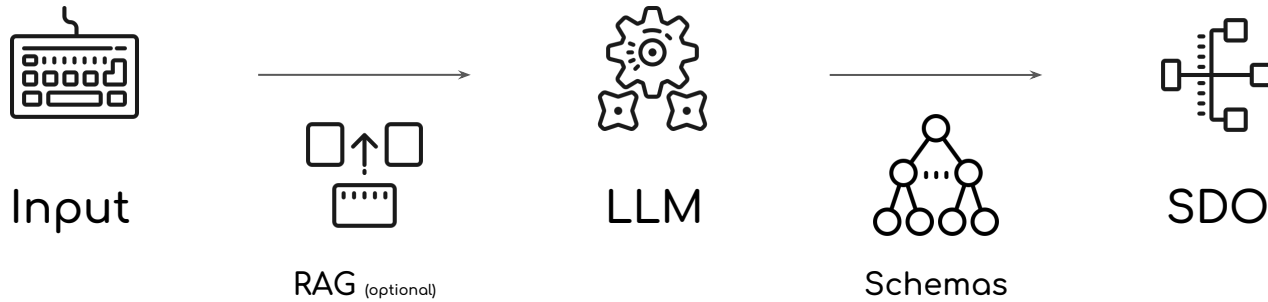
True/Fals e BOOLEANS

A value is either
True or False

[{}, {}] COMPLEX

Enums, Arrays,
Functions, Other Classes,
etc.

Basic LLM Usage



We use **Schemas** to produce **Structured Data Outputs**



Schema

Form for Function

Pairing a prompt and a schema, we can produce a Structured Data Output, that our games and applications can use to modify the experience for the Player.

**(RAG optional)*

Key Point

**“Schemas and Structured Data Outputs (SDOs),
are the bridge between LLM and Game/Application.”**



Schema

Simplified Schema for Crew Members

CrewMember

- name: String
- gender: String
- age: Number
- role: String
- backstory: String
- motivation: String
- appearance: String
- mood: int
- img_prompt: String
- portrait_url: String



Schema

Simplified Schema for Events

Event

- title: String
- description: String
- options: List[Options]
- img_url: String
- narration_url: String
- locations: List[SciFiLocation]
- selected_location: SciFiLocation
- encounter: EncounterLite



Schema

Expanded Schema for Events

Event

- title: String
- description: String
- options: List[Options]
- img_url: String
- narration_url: String
- locations: List[SciFiLocation]
- selected_location: SciFiLocation
- encounter: EncounterLite

Option

- outcome_header: String
- outcome_text: String
- stat_changes: StatChanges

StatChanges

- hull: int
- scrap: int
- fuel: int
- missiles: int
- drones: int

SciFiLocation

- name: String
- scientific: String
- category: String

EncounterLite

- title: String
- category: String
- summary: String
- tags: List[String]

This would have been a nightmare with the old way - total breeze with the new way!

Basic LLM Usage



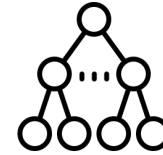
Input



LLM



Output



Schema

Event

- title: String
- description: String
- options: List[Options]
- img_url: String
- narration_url: String
- locations: List[SciFiLocation]
- selected_location: SciFiLocation
- encounter: EncounterLite

SciFiLocation

- name: String
- scientific: String
- category: String

EncounterLite

- title: String
- category: String
- summary: String
- tags: List[String]

Option

- outcome_header: String
- outcome_text: String
- stat_changes: StatChanges

StatChanges

- hull: int
- scrap: int
- fuel: int
- missiles: int
- drones: int



Considerations



Human in the Loop

Design can control SDO

You can and should use design controls to steer the LLM. Not only can you prompt to tell the LLM what to do, but you can also inject what is allowed and what is not.

Key Point

“We can weave in information ‘RAG’ style, to help inform our LLM and make our Structured Data Outputs more relevant.”

Human in the Loop Examples

Where I steer SneakTL



Encounter Types

Helps keep the events at a jump fresh. Breaking up the categories into 48 different categories with 85 example encounters created.



Location Types

Breaks up the different kinds of locations the player could run into. 100 locations populated for this demo.

85 x 100 x Crew Considerations x History Considerations x LLM = n

Basic LLM Usage



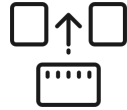
Input



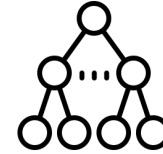
LLM



Output



RAG



Schema

Event Decision History

Crew
Location
Encounter

Event

- title: String
- description: String
- options: List[Options]
- img_url: String
- narration_url: String
- locations: List[SciFiLocation]
- selected_location: SciFiLocation
- encounter: EncounterLite

SciFiLocation

- name: String
- scientific: String
- category: String

Option

- outcome_header: String
- outcome_text: String
- stat_changes: StatChanges

EncounterLite

- title: String
- category: String
- summary: String
- tags: List[String]

StatChanges

- hull: int
- scrap: int
- fuel: int
- missiles: int
- drones: int

CONSIDERATIONS OF STRUCTURED DATA OUTPUTS

Structured Data Outputs



Cost! (AI FinOps)

Each Jump costs roughly:
\$0.01 for the LLM/SDO,
\$0.03 for Image Gen.



Latency More = More

Currently at about 30 seconds.
Image adds 10 seconds.
Audio adds 15 seconds.



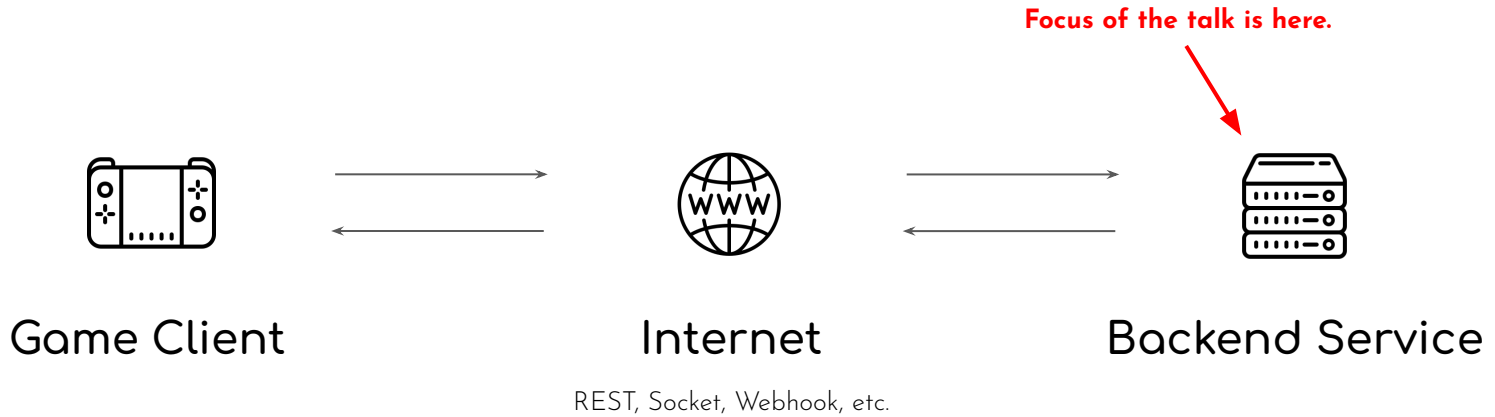
Non-Deterministic

While there are ways to tighten how the LLM will respond, getting true deterministic results should not be a goal. If you need exact results, do not use AI/LLM/SDO.



The Future

“Where” is this Talk?



Backend Service today, Local tomorrow?



Local AI

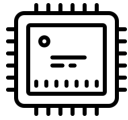
Local meaning "Our Hardware"

Models like OpenAI's GPT-oss can already use Structured Data Outputs. Other providers are likely to follow suit.

While the best models may be online, great models will be available offline.

LOCAL AI

AI Models installed on our Home Devices



AI Chips More Storage

Hardware manufacturers have shifted to talking about how their hardware supports AI.



Offloaded Load

“What can be shifted to the consumers device?” is a stance I have already seen in our industry.



Better Open Source Models

Models can already support SDO locally, but we will see better, faster, lighter versions in the near future.



Other Hardware

More than just our Computers

AI in our TVs may not make sense.
Until they are capable of rendering entire
movies.

AI in our Playstation? Why not?
Developers can hook into it to create
endless games.

Goal of this Talk

“To understand **Structured Data Outputs (SDOs)**,
to add to your design tool kit.”

Key Points

“Schemas and Structured Data Outputs (SDOs),
are the bridge between LLM and Game/Application.”

“Schemas and Structured Data Outputs (SDOs),
are great at some things - but it's a scalpel, not a sledgehammer.”

BTW, this is “Snek”



Just in case you were wondering why everything is named “Snek”.
Its the ground squirrel that lives in our yard.

THANK YOU!

I'm Scott Bromander

AI, Product, and Engineering

Let's Connect!

LinkedIn:

<https://www.linkedin.com/in/scott-bromander/>

Content and Freebies on my Website!

<https://www.Aftercamp.io/>