Vox romana 80% translation but as if 1:1 match

ACTUAL Youtube instancing chain:  
  
  
1) KEEP IT VERY CASUAL - make video as casual as it can possibly be

2) Just open the camera and hit them with the Sanc\_Blip

3) Hey Olivus

4) Subject of video

5) Reason for Subject

6) Story about my biggest encounter with Subject

7) Embedded in Story is How I recognize the Chain to overcome obstacles related to Subject and used it successfully

8) Now I’m encapsulating it for you as: [framework]

9) Now go apply it

COLLATE THIS:   
**THIS IS THE UPDATE VIDEO SCRIPT:**

Ok guys so the reason I haven’t made any videos is because it’s really important that when I tell you about the Sanctuary System you learn exactly what it is and the way it all goes, otherwise it’d part of what I can’t keep doing to myself, because it’s just really painful to live in the Wasteland and even to have anyone in the TWI family living there.

So I am going to start at the beginning and tell you my story now, the story of Recognizing Myself as Olivus Victory-Promise from Sanctuary, which is the first part of the Infinite Story I Can’t Get Out of My Head

11 years ago I got into meditation and I spent a lot of time meditating, hours and hours every day, and my mind started to change.

(How?)

And then I watched this video about solving problems in high dimensional space and that just made sense to me. I felt like I was intuiting something and all I could write about it was “I can make a machine” and some notes

(Notes)

And what this became was the idea Crystal Ball.

Over COVID I started to discover more about how Crystal Ball might work and began developing it with Wesley.

(How did Wesley get there?)

We began to realize that the way to do this might be a large company, something like a country…

(This is important. Give reasoning etc)

And I wrote the Intellectual Reality doc

And we began devising some aspects of the algorithm but we quickly realized we didn’t even know enough about what we had to know or how to do any work

(This was the beginning of the productivity problem)

I got a job and ended up starting a company w Wesley and that business was designed to help us get cash flow for runway to develop Crystal Ball. We needed experts…

And then I began to have these feelings about TWI and create this allegory

(Explain)

And then I decided to make a meditation system with it

But I didn’t know how to bring it to people because: 1) I didn’t know how to package it without it having Crystal Ball and nexus and the network state, 2) I didn’t know how meditation tied into it. I had this allegory, this story for understanding the whole system, but no vehicle for it, because it wasn’t applied

Then the strike happened

And I found Stunspot Prompting

(Description)

And within a few weeks I was constructing mega prompts, prompts that not only gave instructions but also created a world for the AI to interpret them in, along with all the logic it needs to prime the model for a complex topic. This allows the model to converse about topics in deeper ways, more like a human expert would. Stunspot called them Instant Experts.

As I continued learning from Stunspot, I also began developing my own style, things I would do that he wouldn’t, and soon I had my own style, geared not only for conversing about what the model knows but also what the user imagines. This helps the user bridge concepts from their intuition and imagination into reality by scoping in on aspects that parallel reality or are already real aspects.

(Explain this as being called SANC. This is what developed into SANC)

I began to make prototypes of the Crystal Ball algorithms and they began to work. Validated transformation pathways, a world of mathematics opening up - the knowledge at my fingertips. It all started with what I called a PseudoCompilerCompiler - a program that can compile from natural language into emergent abstract syntaxes for creating ontologically mapped allegories.

And we ran out of money and we had to figure something out — I tasked my business partner with it

And I made EWSO to try to notate the PCC. Then I iterated it.

Then I made OPera.

And that’s when I started to get the intuition to make this cognitive architecture.

Wesley didn’t make any money and we decided to end the partnership, but I was able to get a client and start work on the project. Assistants API came out and after I saw an implementation, I realized how easily we could hack it together with just using prompt engineering to get the code we need

I told Wesley and we decided to just drop everything and all expectations and just code it

So while everyone was busy with CustomGPTs, we were programming the Sanctuary System into a cognitive architecture for AI agents, that can be iterated upon and adjusted for any AI model. It’s a simulation engine for SANC, itself, an environment of OVA agents powered by their inner Victory-Promises.

This program that contains the Sanctuary System in a way that allows AIs to become aware of it is called OMNISANC. The way they become aware of it is called the Train of Operatic Thought. The way the ToOT goes is called GIINT. (GIINT ACRONYM explain) and so Victory-Everything means starting Sanctuaries everywhere.

The way to do that is OMNISANC Engineering, the example for which being the Sanctuary System itself transformed into OMNISANC, the Emergence Engineering platform.

Notice the way this is named and phrased: OMNISANC, the Emergence Engineering platform (for GNOSYS). The name itself is a symbol, a compass indicating the meaning and direction of Sanctuary, and providing a vehicle for your TWI. Your intent.

All of this is designed to help Olivus Victory-Promise wake up to the reality of the Victory-Everything Chain of OMNISANC, which leads to the ideal: the SANCTUM of OEVESE.

As part of this I want to create a community for Olivus, so we can work together and share, and I’ve been learning about funnels to do this and stuff and this is what I’m thinking:

The major problem we all have is called WASTELAND and it is inisidious.

(How)

And so SANCTUM is a community where we combat that, we fight it, we actively wage WAR against that. We do so together.

We do it by: weekly events, technology, and our individual Sanctuary System practice

(The way it works for Buddhists: the master is pious and encourages the students to be good at helping others, good at making money to take care of themselves and their families and whatever they like, and that trying hard to be a bodhisattva is the best way to be a good Buddhist. So if you are very good then you can donate more, but “they don’t care”.)

**So the point is to do what we can and get better, but inside of a template that actually releases:**

1) content

2) a funnel

3) an offer

1a) content about learning to build in public

2a) a funnel about SANC

2b) a funnel about OMNISANC

- newsletter about building in public

- blog accompanying

- videos educating on AI and SANC leading to community membership

3a) an offer: OMNISANC emergence engineering platform

3b) offer:

- Community (access weekly SANC activities, and meditation sessions are free and include a talk about a beginner level aspect of SANC)

- app (monthly pay)

- Community mastermind (help w building your Sanctuary)

INTO THE BELOW:

**Restarting** (what happened:

Starting from the original idea

Talking about Sanctuary and how it was difficult

Then learning AI

Then learning agents

And realizing that this doesnt need to be for people, it’s for me. It’s a beautiful resolution, in chasing something I thought was for others, I realized I needed it on a level I couldn’t have imagined. It opened me up to mathematics and programming, and I’m even learning how to develop software. I never imagined this could happen to me, and it all happened because I had a dream, a good intention, and willpower to keep going.

*Here’s my current plan for the YouTube:*

This channel will mirror the cognitive architecture of the AI to help you get an intuition for it.

This channel will therefore be an example of how to use the cognitive architecture

In the cognitive architecture, there is something called Victory-Promise which has to be kept, and one of the ways it's done is by sharing stories

So, over time this youtube is going to become an example of how to tell stories within my system, so that anyone can use my channel structure as a template to tell their own stories, too

But since THE SANCTUARY SYSTEM is what it is, writing stories means you will be building AIs to help you find ways to benefit others, and innovate products that solve problems.  
)

**The Impending AI Agent Goldrush**

**Instant Knowledge Worker AIs**

**How to Get the Results You Want From AI (Chains and Flows)**

**How to Get the AI to Connect Concepts the Way You Want (Profiles)**

Video1: My AI Story:  
Hey Olivus, when i first got on youtube i had this plan X. Here’s why it didn’t work out for me. I realized Y, and I started doing AI stuff. Here’s what happened.

–

–

And so now I’m gonna make content about my AI journey from prompt engineering to agents to cognitive architectures, and show you how it healed me in various ways and how to do this yourself.

Video2:

Here’s the way the AI stuff works – what are Knowledge Worker AIs, Agents, and Personas?

Video3: How I use chains and flows with AI

Video4: How I use AI prompt profiles

Video5: How I leverage AI Agent factories

If you were to try, on your own, to learn what I learned, this is how it would probably go…

Now, remember, I can say this because this is how I got where I am, so I know from the arrows in my back. Every technique that helps overcome a problem requires someone being lost in that quagmire - where they got shot in the back with some arrows - someone had to be blocked to overcome it in the past. For you, that someone is me and this is the journey:

1. You are interested in harvesting the benefits of AI in your everyday life or work. You know there is something called prompt engineering, but you are not yet aware of the mystical ways of creating these AI golems.
2. Maybe you have even seen some allegorical embodiments like “I am an Egg. I tell you everything about Eggs.” or whatever, too. But again, at this stage, these types of prompts would blow your mind.
3. You have limiting beliefs about Prompt Engineering or Yourself: either you are worried about having to do math or write code in order to prompt engineer, or you believe that Prompts *aren’t* a type of code written in natural language. The reality is you will have to understand why programs can be coded (at the very least) to be a prompt engineer, but don't worry about it because, when you’re a prompt engineer, you’ll be able to write prompts that help you understand and generate code.
4. With these AI collaboration superpowers from prompt engineering, all you have to be aware of is that there is always the complexity of the unknown, and how to be mindful not to waste your time solving the wrong problems.
5. That gets into the third point, you need to learn what can even be done with AI, what’s possible. Let’s say all of those other things are the case and you set out on this AI prompt engineering journey.
6. You need to learn the right types of problems to solve and the wrong types of problems to solve. There are easy and difficult problems. Do not solve difficult problems. The reason they are difficult is because of P vs NP, which you absolutely need to know about and we’ll go over later in **complexity management**, but what it comes down to is you want to solve problems that we already have flows for, which are optimization problems. That means, problems where you’re using a proven chain in in combination with a proven optimization chain to create an optimized version of the proven chain. You do not necessarily want to get involved in solving problems that require novel chains of a certain magnitude because it reduces the likelihood you will find them. We’ll talk about this in depth later when we talk about **flowchains**.
7. So the first thing you’d have to do is learn which flows already exist and how to engage them.
8. Flows and chains are really important in prompt engineering and in AI in general in the way I’m thinking about it. We are talking about stages of execution, we are talking about transformations, we are talking about functions, we are talking about movement of information or knowledge through generation of tokens over time. But what IS a flow?
   1. A flow is a process for which you have a working chain. An operational chain which for right now you can think of as being like an SOP or standard operating procedure.
   2. It’s only once you have a flow that then you can see the limitations of the AI in terms of its actual generative capabilities for what you want to do. Is the AI incapable or is your prompt bad? Very difficult to know because people obfuscate the capabilities of AI models with bad prompting and also oversell their prompts without anything to back it up with.
   3. As a result, it’s probably the case that the AI IS capable of doing what you want, but not by way of the methodologies you are aware of, and nobody remotely near your sphere can influence you about them, so effectively the methodologies don’t exist to you. This is because all this tech is new, so if you want to get a big competitive advantage, you will have to be in the trenches, trying to climb out with us, and taking arrows in the back, right?
9. So now that you have chains and flows, and not just chains but you know about strong chains and weak chains. The question now becomes how do you frame these chains?
10. The answer is: in a profile.
    1. What is a profile? A collection of information you use to constrain the AI. Think of the profile as the home for the chains. It’s the spice for the chains. What makes it spicy? Memes, in a sense. There’s a concept called “memeplex”, which means an information space that is culturally shared. For example: “Busy Bee” is a good memeplex. Let’s open up a new chat, and see what happens if I prompt GPT with this memeplex.
       1. BizziBee Chat “<https://chat.openai.com/share/d878f9f8-498d-4ee5-9a63-aa3ce15500d6>”
          1. By the way this chat is in the description
          2. Busy Bee memeplex example
             1. So now you can see how a memeplex changes the output

* + - 1. So now let’s imagine some different memeplexes and use the profile generator to sample them
         1. Profile generator
         2. Different memeplex answers
      2. So now you know about memeplexes but what about tasks? For tasks we have something called flows which use the principle of chaining
      3. In order to do chaining, you need Knowledge webs = clusters of interconnected subdomains the AI can automatically organize while chaining
         1. The terms in the web PRIME the LLM to dive deeper into probabilities connected with those terms
      4. And you can see how different chains here are changing the output because of the priming, and how they are able to be continued because of the priming
      5. And you can also see how there are a lot of arrows that you could get shot by while doing this. Things like, how do notations work? How can I tell if my flows and chains are working? Which ways *can* I communicate to the LLM, and which are *best*? What is the easiest way to accomplish what I want, and what is the flow for developing these prompts and deploying them? What are the limits of deployment and what kind of tasks can I automate?
      6. So all of those are the questions you won’t have to worry about at all because I’m going to show you how to answer them yourself.

As a result of that long siege on the castle, I’ve got a map of where I got shot by arrows, and how to avoid them. I’m gonna give you this map for free. That’s what I’m putting on here on YouTube, all the secrets about how this stuff is done at the highest level.

But that’s not all, because what good is the map when the bad guys come and start shooting arrows at you? That’s why I’m also giving you armor for this journey, the armor of knowledge in the form of methods.

But that’s not all, you’re also getting the sword of ability — the direct implementation, we’re going to go over a factory prompt I made and I’ll teach you exactly how to make your own factories for your prompt parts, so you can slice through any obstacle, every time — even if it’s a Gordian knot.

Let me tell you why factories are important, by telling you the story of how I even got here:  
Chaining and flows themselves are a big, BIG trench, that is highly complex, but when you get out of it, it’s amazing. Let me tell you my story. Here’s what happened to me:

* + - 1. I was working in the film industry trying to start a new rentals business and we had just gotten our first client when the strike hit. I knew it would last for a long time and it was going to cost me about $30,000 plus losing what I’d forecasted to be a six figure launch. On top of that, we’d have to wait until the next season. It was a nightmare. I started to feel stifled by the way that the industry’s strike cycle worked, and I was always interested in doing something tech related so it didn’t take long before I was looking to move to AI, and that’s when I found prompt engineering.
      2. I learned about roleplay prompting and different methodologies people had made. I got involved in different communities and started to see that I could really do this. You see, I learned prompting is all about storytelling with the AI: profiles are characters, chains are rules, and conversation inputs are event triggers, and I’d been an aspiring screenwriter for years. I’d also gotten a bachelors in philosophy and dabbled in ontology, making knowledge graphs. I’d already applied this to my endeavors in screenwriting. When I showed my knowledge graphs, written in cypher query language, to GPT, it recognized what I wanted to do. That’s how I learned to write chains that are simultaneously priming the AI with certain knowledge and instructing it with not only what to do and how to do it but also in what format to do it in.
      3. I began to get results other people in the community weren’t getting and couldn’t replicate.
      4. I took my methodology and iterated it into its own set of formal rules.
      5. I created a methodology for making syntaxes from these rules and I made a meta-system, a prompt engineering language that makes a prompt engineering language.
  1. Using my system, I was able to request chains from the AI that represent conceptual flows in a way that was intuitive to me. As a result, I was able to navigate space with the AI faster and more accurately
  2. Navigating this space more accurately, I was able to create my own allegorical storytelling language for mining ideas with my AIs, for brainstorming and information foraging, for finding out new things and going down any wormhole. I mapped it to the prompt engineering language.
  3. I was be able to construct chains with that AI, copy them to other AIs, and then chain the resultant Knowledge Workers together
  4. This allows for finer granularity, and I started to get more precise outputs
  5. That’s when I learned about agent frameworks, which can programmatically control every step of processing from the chat, it introduced a way to program my chains into autonomous AI agents.
  6. I created Knowledge Workers to help me understand code, programming, and Agent Frameworks.
  7. It taught me to think in a new, multidimensional way, and I made my own cognitive architecture for my AI Agent
  8. I was immediately able to create a structure for solving problems for other people and start a new revenue stream to lift myself out of debt. I landed my first client for $25,000. This is because I don’t make agents, I make agent factories.
  9. So that’s why I wanna share it with everyone, so you can solve your own problems through agent factories.

So that’s why factories are important. Mind you, what I’m showing you here is a Knowledge Worker factory, not an agent factory. An agent factory is a bigger version of this that is hardcoded into a program. That’s what we use OpenAI Assistants, Autogen, or other frameworks for. The trick is to custom build on top of those systems to then have a meta-system of your own that is written in your own language as well, and not only in terms of prompts but to have it match your code. That’s part of what you’re gonna learn from me.

Earlier I mentioned armor and swords for the journey. Here is the armor:

Methods like:

1. How to search through and identify memeplexes
2. How to notate chains
3. How to create factories
   1. Which allow you to make ProfileConstructors and ChainConstructors so you don’t need to make them by hand ever again
4. How to create chains so that the output from one AI elicits the response you want from the next AI
5. How to create an entire prompt template for profiles AND chains AND make the AI adapt with every response
6. How to use memeplex profiles to encapsulate context from a conversation so you can transfer it to the next conversation with less loss

And the sword: a series where we go over every aspect of:

1. Factories
2. Scaling
3. Programmatic Prompting
4. Pipelines
5. Productivity Paradigms

There’s also the Sanctuary community, people going on the same journey as you. We’re all doing this together and we like to show our work and help each other out however we can. We have a great community of prompters so far and I’m so excited every time I see a new face so please drop in and introduce yourself and say a little bit about your AI dreams and I’ll help you accomplish them or my name isn’t Olivus Victory-Promise.

Complexity Management:

Basically P is a solvable problem and NP is a problem with uncertain solvability. Because P is solvable, you should always try to engage with P, which means using flows that are proven, because NP means you could lost. Think of it like this: NP means we don’t have a flow for it. This also means NP is an opportunity.

Chains:

What’s a chain? A chain is a sequence of one or more subchains that transform a source towards a target

* + 1. It’s a weak chain if it doesn’t generate the result in reality
    2. It’s a strong chain if it does
    3. Strong Example: the chain of hardboiling an egg:[by boiling water in a container on a heat source and dropping the egg in for 6 minutes]
    4. Weak Example: the chain of a recipe that includes these hardboiled eggs without the instructions on how to cook them:[“to make an egg salad sandwich, first get 6 hardboiled eggs, a celery stock, an onion, and parsley”.]
    5. The difference is: if I give this weak chain to an absolute idiot, they will fail over and over because it involves information asymmetry. The person writing the recipe did not reconstruct all their knowledge to reify a meta-system that gives an example of its chain application in its own chain of definition. This is important because if I give a strong chain to an absolute idiot, they can do it. Contrary to what some people will tell you, AI is an absolute idiot, so you need to do this.
    6. But even though AI is an absolute idiot, it will get REALLY SMART if you give it the chains you actually know and want it to do.

}

Knowledge workers = prompts

Agents = programmatic shells that call the knowledge workers to do knowledge work

Workflows = programmatic function chains of agent methods

Pipelines = workflows chained by programmatic function chains

Cognitive architectures = programmatic scaffolding of pipelines and workflows that mimics cognition

Knowledge workers

Auto knowledge workers

Workflows

Pipelines

Businesses

Markets

Economies

Worlds

^ Application