

The New Secret Creataur.io Master Plan

A Reasonable and Exciting Ten Year Plan

Background: for the past ten years I have been working as a programmer for small software companies. Like many people, I plan to reach financial freedom so that I can work on my own ambitious enterprises. Eventually I want to head a technology company. Ideally my projects will have Singularity University level impact at some point, meaning they positively impact the lives of 1,000,000,000+ people. Unfortunately, my formative years were a bust and my social network (or lack thereof) does not allow me to find a team to fund a company at the moment. On paper, the world should already be my oyster - considering I am a white, male programmer with roughly 25k dollars available for the cause.

This master plan is inspired by [the Tesla one](#) - and [the other](#) - but don't tell that to anyone.

Basically I think there should be a semantic layer under the public internet by now. According to my thinking, this is a missed multi-trillion dollar opportunity. I am unable to create and launch such a project, though. Basically I have an idea of what such a project could look like but the sheer scope means I can't create a MVP on my own. I did some thought experiments of what a Designaur.io project could look like. It would allow me to create SaaS platforms a lot faster by abstracting and streamlining the code generation.

The project would be very hard to market directly though, because the vision is not trivial to understand. This is why I am going for a high level project that builds on a minimal feature set of the semantic web layer that I envision. That project is even more ambitious, but it can be broken down into chunks and, more importantly, more easily marketable parts and can be tackled in several different ways with countless opportunities to pivot, because, at its core, it is constructed around the business idea that you should create a pipeline when others carry buckets of water. It is the idea of managerial leverage - maximized as much as reasonably conceivable. In the first iteration it will be applied to the department of digital asset creation.

Obvious topics that I would like to eventually cover are:

- Audio
- 2D / 3D Resources (Textures, Models etc.)
- Design (Architectural, Fashion, Textual, Industrial, Logos, Calligraphy, Covers etc.)
- Game Assets
- Test & Demonstration Data
- Simulations

The basic idea behind the semantic layer aspect of the project is that currently people reinvent the wheel when they create APIs. Understanding and using an API is relatively hard as a consequence. The technology also is not as robust as a battle hardened standardized solution could be. Different services and products have their own proprietary ways to handle metadata. The only way to parse the content on the internet is using natural language processing which will create infinite pain in the near future.

Having a standardized technology for metadata with adoption comparable to Apache Web Servers would no doubt eliminate billions of dollars per month in trouble that software and consulting companies have. It would also provide new ways for creative solutions around the areas that become more relevant in the modern web world. Aggregation, migrations, remixing etc.

The simple pitch for the solution would be creating query and definition / manipulation languages for metadata, pretty much the way that relational databases currently work. No other technology would implement a different solution to store and fetch metadata. There would be a unified data model that describes the world which is constantly being updated. Tags and categories would be used to make sure software can fetch only the metadata that is relevant. Versioning would provide a way to prevent version conflicts when the data model gets updated.

The idea behind Computer Aided Procedural Engineering is that rulebooks which are built around the metadata in those systems can create data that follows the metadata description and has a random character while it is created in a repeatable manner. This is a well established concept. Procedural Content Creation is the term that is used to describe a way to use rulebooks that can create entire worlds, so that there is no need to store the entire world directly at any point.

Basically that means something is created from nothing. A sequence of numbers is sent to a program, and the rule engine basically makes creative decisions that an artist would make. The end product is a digital asset.

Such a system can be monetized directly or indirectly. Imagine a customer wants a WordPress Theme and all you have to do is check a few boxes that determine what the result should look like. The service creates 30 results that conform to the parameters and you can pitch your favorites to a customer. Of course the tricky part is creating rulebooks that can create

WordPress Themes which are sophisticated enough to compete with the results that a web designer would create.

This sounds like a downside, but actually it means that the business would be hard to compete with. Reproducing the results is not trivial. Warren Buffett calls this a moat. It increases the intrinsic value of a business significantly.

Also, the idea is not creating asset factories that are set in stone and hard to change. The idea is creating an authoring tool for rulebooks. Artists and domain specialists should be able to define how they want their assets to be created, then the service must be able to mass-produce assets at a rate that no human could. The hierarchical nature also means that, once many basic rulebooks exist, emergent features are likely to manifest, because high level rulebooks can build on the work that has already been done.

To create such a service, it needs to be broken down into manageable chunks. The principle of keeping it simple must not be violated. Getting the basic technological concept right is crucial. A look at the functional concept reveals that there are several hard design challenges.

The current high level concept could be:

Horizontal Modules:

- Microservice: Metadata Management
- Microservice: Data Storage
- Microservice: Data Timeline
- Microservice: Data Archives
- Microservice: Project Management
- Microservice: Orchestrator
- GUI: Main UI
- GUI: Admin UI
- Monolithic Solution: Data Creation (accessed via Microservice API)
- Monolithic Solution: Asset Creation (accessed via Microservice API)

Data Verticals:

- Texts & Numbers
- Audio
- 2d Scenes
- 3d Scenes

Feature Verticals:

- Load / Import
- Save / Export

- Tenancy
- Versioning

One design challenge here is deciding which parts should be generic and which need to be fixed. The conflict: on the one hand you want everything to be generic, so that the artists are not limited in their creativity. On the other hand the GUI should be able to represent data in a way that the user understands intuitively. That means the GUI needs a way to interpret the RAW data that the data creation produces.

The initial plan is creating first class data model elements that the custom data models can use. The GUI can then be relatively generic by being able to infer properties of the elementary data model parts.

Another design decision that needs to be explained is the separation of data creation and asset creation. The intuitive way to implement a PCG (Procedural Content Generation) asset creator to output assets directly. A layer of abstraction offers countless benefits, though.

Most data creation rulebooks will have several hierarchical layers. Suppose a planet is created, then the continents on that planet and the creatures, landscapes and animals on those continents.

The creation of the landscapes will depend on the properties of the continents that have been created. This is a lot easier with well structured raw data than with a hard to parse 3d model that can be viewed in Blender.

It will also be possible to create raw data that allows for assets in different levels of details. The properties that the asset creation service receives will determine the level, then. This allows for more flexibility in the asset creation pipeline.

There are countless potential use cases where computer aided procedural engineering could provide immense value. Fields that are currently on my radar are:

Music Generation, Calligraphy, Typesetting / Layouting, Story Arcs, Textures, Landscapes, Universes, Creatures, People, Vehicles, Buildings, Interiors, Cities, Furniture, Machinery, Ornaments, Covers, 2d/3d Tilesets, Sculptures, Fonts, Paintings, Drawings, Spritesets / Character Sheets, Gif Loops, Audio Loops, Speech, Sensor Data, Infographics, Clothes, Game Levels, Lyrics, Poses, Expressions, Body Movement, Traffic, Battles, Ecosystems, Family Trees, Peoples, Shows, Dialogues, Languages, Accents, Alphabets, Companies, Weather, Color Palettes, Logos, Style/Design Guides, Sewing Patterns, WordPress Themes

The value of the generic aspects of the metadata hopefully becomes apparent when we take a closer look at the following use cases:

- I. Test and Demonstration Data: Imagine a software company wants to test how well its software deals with massive data streams from millions of sensors. A CAPE cloud SaaS - for example productized as the creataur.io cloud - could create repeatable datastreams

with a random character and all the reporting that management might want to see. The data model has to be defined and a rulebook has to be written. The asset creation might not even play a role here. The data creation could be used by another service that feeds the software to be tested.

- II. Music Generator: Here the separation between raw data and asset creation could lead to great results. The raw data could be stored the way MIDI does it plus additional data like lyrics or comments that the asset creators can choose to ignore or incorporate in their result. Different services could allow the creation of a MIDI file or an MP3 file with kinks, effects and emphasis that allows the result to resemble a human performance of the song.

Identifying the best initial use cases that reveal blind spots and flaws in the intuitive technical concept is crucial. Another constraint on the use case is scope. It should be possible to create a MVP in a reasonable timespan. Creating convincing speech based on raw text is definitely too ambitious. Calligraphy is challenging but could offer just the right kind of value to the market. I also have a personal interest in random Piano Sheet Music that I could perform or use in projects. The most obvious use case is the creation of 3D models of creatures and people that can be used in games or short animation projects.

To me it seems that the value skew is so obvious that there is no need to validate the CAPE potential. It will become important to evaluate the value of a certain kind of asset later, but investing time in a solid foundation can hardly be an issue.

The plan is selling assets at some point in 2022. Opening the creataur.io cloud as an authoring platform for rulebooks to studios and artists is planned for 2025. From 2025 through 2030 the plan is working on proprietary rulebooks with an ever increasing sophistication and over time cross-financing other projects that create a technology company that is well positioned to positively influence the lives of 1,000,000,000+ people in the 10 years that follow.

Don't tell anyone.