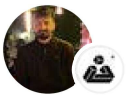
Navigating Token Engineering: The Blueprint for Designing Sustainable Decentralized Economies

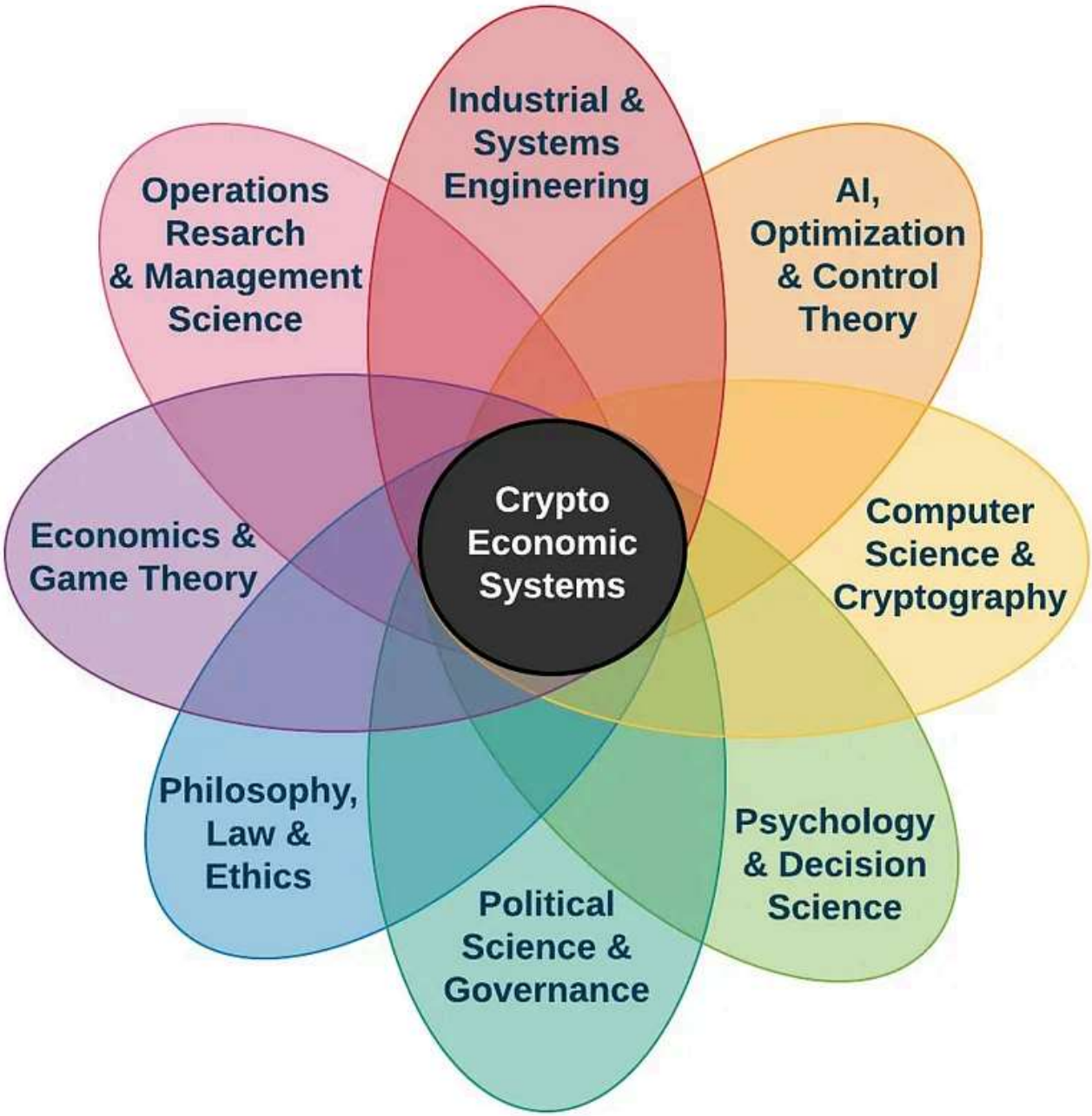


Idrees

Published in BlockApex · 11 min read · Sep 11, 2023

Abstract

This comprehensive guide serves as an essential resource for navigating the complex and interdisciplinary field of token engineering, grounded in systems design and complex systems engineering. This article outlines the critical phases — System Requirements Analysis, System Analysis, and System Design — of developing decentralized crypto-economic systems. The guide also aligns these phases with established frameworks in both systems engineering and token-based ecosystem creation. Additionally, it provides a detailed breakdown of tasks, skills, and deliverables for each phase. Readers will gain not only a deep understanding of why token engineering is vital for sustainable digital economies but also practical insights into the process, tools, and experts shaping this dynamic field.



Introduction

In 2022, I started my journey in token engineering, with a background in mechatronics and robotics engineering. Over the past two years, I've gathered resources that I believe can guide others navigating this emerging domain. This research has been significantly influenced by several key figures in the field. The conceptual framework for understanding crypto economic systems draws heavily from Michael Zargham's groundbreaking

contributions, which define token engineering as both a Systems Design and a Complex Systems Engineering discipline. Angela Kreitenweis has played a key role in advancing the field of token engineering through TE Acadmey Platform and her initiatives, which include organizing research sessions, events like EthCC Barcamp, study groups, courses, and grants, while also assembling a global community of experts and enthusiasts in this field . Krzysztof Paruch, Trent McConaghy and Dr. Achim Struve are other key figures whose valuable research has played a pivotal role in defining and establishing this emerging domain by emphasizing the centrality of token engineering to innovation and development within web3 ecosystems. Recognizing the intricate and interdisciplinary nature of this domain and complex and diverse nature of topics, I've included a variety of resource links throughout the article to offer readers a deeper understanding of specific concepts.

What is Token Engineering?

Definition

Token engineering is design, verification, and optimization of token-based complex economic systems

Why Token Engineering Matters?

Token Engineering is not just about creating digital assets; it's a rigorous discipline that demands a comprehensive approach to designing

decentralized systems. Just as traditional systems require meticulous planning, analysis, and design, token ecosystems too demand a rigorous process for their conception and realization. Whether you're gathering requirements or deploying smart contracts, each phase plays a critical role in ensuring that the decentralized system is not only technically sound but also economically viable and socially impactful. As we delve into the nuances of each phase, you'll see that Token Engineering is not merely a technical endeavor but a comprehensive approach to creating sustainable digital economies.

In the contemporary landscape of blockchain-enabled decentralized and distributed economic systems, we'll delve into the field of token engineering as an important part of development of decentralized ecosystems, viewing it through the framework of systems design and situating it as a specialized subset of complex systems engineering.

Token Engineering Process

In this article, we'll delineate the scope of the token engineering process within the framework of the three standard phases of the product development lifecycle. This token engineering process aligns closely with established frameworks in systems engineering, and also with Outlier Ventures structured framework of token based ecosystems creation, which includes Discovery, Design, and Deployment phases. We'll outline the requisite skills, the expected deliverables, and the essential tools for each phase to offer a comprehensive understanding of the field.

1. System Requirements Analysis
2. System Analysis
3. System Design

System Requirements Analysis

System requirements analysis is the first phase where we document the system requirements. Here, we define what the system is supposed to achieve. Stakeholders generally agree on these requirements before moving forward. This sets the stage for subsequent analysis, design and development phase and serves as a point of reference for all stakeholders.

System Requirement Analysis phase is further divided in two following sub steps/phases:

1. System Requirements Gathering
2. Requirements Analysis

System Requirements Gathering

In this step where high-level needs and constraints are collected from stakeholders (documented / verbal). The focus is on understanding what the stakeholders expect the system to achieve. This could include both functional and non-functional requirements like features, performance, security, and compliance.

Process

Primarily focused on identifying and detailing the features and functionalities the system must have. This phase often involves stakeholder interviews, use-case definitions, and the documentation of both functional and non-functional requirements. It answers questions like “What should the system do?” and “What are the constraints?”

- Conduct stakeholder interviews to gather needs and expectations.
- Document user stories or use-cases.
- Identify functional and non-functional requirements (e.g., security, scalability).

Components

- Functional Requirements (Features the system must have)
- Non Functional requirements (Performance, security, compliance with legal regulations such as anti-money laundering (AML) laws, etc.)

- Performance Requirements
- System Technical Requirements
- Specifications
- Use-cases or user stories (Scenarios describing how the system will be used)

Requirements Analysis

After the initial gathering of system requirements, the second sub step/phases of requirements analysis, focuses specifically on examining and refining the project's requirements by dissecting, validating, and prioritizing these requirements while clearly documenting the functional and non-functional requirements of a system. The aim is to ensure that the requirements are specific, measurable, achievable, relevant, and time-bound (SMART). This involves clarification, prioritization, and validation activities.

The refined requirements then become the basis for system analysis and system design. This step ensures that the requirements are unambiguous, complete, and aligned with the project's goals. It ensures that all stakeholders have a mutual understanding of what the system is supposed to achieve.

> *Requirements Analysis might answer: "What kind of staking rewards will incentivize network participation?"*

Components

While system requirements analysis often starts with the gathering of initial requirements from stakeholders, the requirements analysis part dives deeper to scrutinize those requirements for clarity, completeness, feasibility, and relevance by:

- *Validation:* Ensuring that the gathered requirements are aligned with the project's goals and stakeholders' needs.
- *Prioritization*: Deciding which requirements are must-haves versus nice-to-haves.
- *Clarification:* Breaking down high-level requirements into more detailed specifications.
- *Conflict Resolution:* Identifying and resolving conflicting requirements from different stakeholders.

The output of the requirements analysis process is often an updated and more detailed version of the Requirements Specification Document, which now includes prioritized, clarified, and validated requirements.

Skills / Techniques

- *Requirements Engineering*
- *Use-case modeling, user story mapping, and feature prioritization*
- *Mechanism Design and Game Theory*
- *Communication and Stakeholder Management*
- *Basic understanding of blockchain and Web3 concepts*

> *In the context of token ecosystems or blockchain projects, phase could involve mechanism design and game theory. These methods help in designing incentive*

> *structures, governance models, and other features that are crucial for decentralized systems. The focus here is on what mechanisms should be in place for the system to function as desired.*

Output/Deliverable

- ***Requirements Specification Document:*** A detailed document outlining what the system should do, often including mockups or wireframes.

It includes both the initial requirements gathering and the more detailed Requirements Analysis. It outlines what the system is expected to do. The Requirements Specification Document focuses on the 'what' — what the system should achieve without detailing how it will do so.

Example

Let's assume we're building a Metaverse protocol that allows users to own, trade, and interact with digital assets in a virtual world. The requirements might include:

1. User-owned digital assets (NFTs).
2. A marketplace for trading assets.
3. Virtual spaces for social interaction.
4. Governance mechanisms for community decisions.
5. Interoperability with other Metaverse protocols.
6. Low latency and high performance.

In system requirements analysis, we document these features and criteria that the Metaverse protocol must fulfill and duly scrutinize and refine these

requirements. This stage sets the foundation and serves as a guideline for what the system is supposed to achieve

System Analysis

Once we know what the system is supposed to do, we proceed to analyze how it can be done and what challenges might arise. This step involves feasibility studies, risk management, stress testing, economic modeling, and sometimes, initial prototypes to validate key assumptions. The findings from this stage can refine or even alter the original requirements. System analysis also involves evaluating the existing similar systems and understanding various facets like user requirements, system limitations, and potential bottlenecks. It often includes analyzing the feasibility of the project in terms of technology and economics. It involves examining the implications of integrating with existing ecosystems or the potential for creating new systems.

This phase, especially in complex systems like token ecosystems, involves *mathematical specification, differential specifications, state space representation, agent-based modeling and system dynamics modelling* to validate the feasibility and robustness of the mechanisms designed in the requirements analysis phase to validate assumptions about system behavior, user incentives, and economic viability. These methods aim to foresee how the system will behave under different conditions.

> *System Analysis might answer: "Will the staking mechanism stand up to extreme market volatility, and how will agents behave in such conditions?"*

System analysis aims to understand how to best implement the requirements and technical challenges associated with implementing those features and functionalities. This phase involves simulations, modeling, and

other forms of validation to understand how the system can meet the defined requirements. It answers questions like “Is this technically feasible?” and “What are the potential risks?”

Tasks

- Feasibility studies: Technical, economic, and operational viability.
- Preliminary modeling to validate assumptions or to understand system behavior.
- Risk assessment: Identifying potential bottlenecks, limitations or security risks.
- Modeling and Simulations: For behavioral modelling and user incentives design, to validate assumptions and understand system behavior.

Skills / Techniques

- *Mathematical Modelling (Mathematical Specifications, Differential Specifications)*
- *State Space Representations*
- *Statistical Modeling*
- *System Dynamics and Agent Based Simulations*

Output/Deliverable

- Feasibility Report: Documenting the viability and risks associated with the project
- Updated Requirements: Modifications or refinements to the initial requirements based on analysis
- Mathematical and Differential Specifications of Mechanisms and System
- Mathematical (State Space Representations)and Statistical Models

- Simulations (Agent Based and System Dynamics)

Example

In this stage, we assess the feasibility and implications of the requirements. For our Metaverse example, this could include:

1. Feasibility: Can current blockchain technology support the level of performance we need?
2. Economic Model: How will the marketplace function economically? Are there any game-theoretical considerations for trading?
3. User Behavior: How are users likely to interact with assets and each other?
4. Stress Testing: How the agents will have behave in extreme conditions?
5. Boundary Conditions: What worst can happen and how to avoid it?
6. Security: What are the potential vulnerabilities that actors can exploit?

System Design

After understanding the requirements and completing the analysis, we move on to designing the architecture and components of the system. This stage results in a blueprint for building the actual system, including technology stack decisions, data models, and workflows.

In the context of a decentralized system or a token ecosystem, System Design serves as the architectural blueprint that dictates how the system will be built and how its components will interact with each other. This phase comes after System Requirements Analysis and System Analysis have established what the system should do and verified that it's feasible.

Tasks

- Architectural design: Laying out the system's high-level structure.
- Component design: Detailed design of each component, like smart contracts or APIs, Integrations.
- Data modeling: Deciding how data will be stored, accessed, and managed.
- Token Models, Strategies, KPIs,

Skills Required

- Software Architecture
- Smart Contract Development
- Systems Engineering
- Verification and Optimization
- User Experience (UX) Design

Output/Deliverable

- System Design Document: A comprehensive blueprint of the system architecture, components, data models, and interaction flows.

> *The System Specification Document focuses on the 'how' — providing a blueprint for building the system.*

This document is more technical and is usually the output of the System Design phase. It outlines how the system will meet the requirements set forth in the Requirements Specification Document. The System Specification Document may include:

Example

After requirements are set and analyzed, we move on to system design to create the architecture that will satisfy these requirements. For our Metaverse protocol, this could involve:

1. Blockchain Layer: Choosing between a layer 1 or layer 2 solution based on scalability needs.
2. Smart Contracts: Designing the contracts that will handle asset ownership, trading, and governance.
3. Tokenomics: Defining the role of tokens in incentivizing participation and governance.

If we've determined in the Requirements Analysis that our token ecosystem needs a governance token, and the System Analysis has confirmed that such a mechanism is feasible, the System Design phase will specify:

- The smart contract code for minting and distributing governance tokens.
- Governance Structure, Quorum, Quorum Threshold, Voting Power etc.

Summary

These steps of product development are sequential but not strictly linear; they are often iterative and may loop back on each other. For example:

- Feedback Loops: During system analysis, you might discover that some requirements are not feasible or could be optimized, leading to a revision in the system requirements.
- Iterative Refinement: As the design progresses, new insights could necessitate revisiting the analysis or requirements stages.

- Agile Development: In agile frameworks, these steps might happen in smaller, iterative cycles rather than one large sequence, allowing for continual refinement.

These phases are iterative and often loop back for refinements. For instance, during system design, you might realize a need to update requirements or re-analyze certain aspects, triggering a return to earlier phases. Each phase has its own set of specialized skills, but all contribute to the project's ultimate goal: building a functional and effective Web3 system.

Individuals to Keep an Eye On: Leading Figures in the Field of Token Engineering

- Michael Zargham
- Angela Kreitenweis
- Trent McConaghy
- Dr. Achim Struve
- Tarun Chitra
- Krzysztof Paruch
- Guillaume Lambert
- Dan Robinson
- Rohan Mehta
- Curiousrabbit Eth
- Lisa JY Tan
- Tim Roughgarden
- Roderick McKinley, CFA

Companies to Watch

- BlockScience
- Gauntlet
- TE Academy
- TE Labs
- Outlier Ventures
- TE Commons
- BlockApex Labs
- Tokenomia Pro
- Tokenomics DAO
- Economics Design
- CryptoEconLab

Here is a list of companies indirectly contributing to the field of token engineering through valuable research:

- ConsenSys
- a16z Crypto
- Ocean Protocol
- Paradigm
- ChainLink

Online Communities

- TE Academy Discord

- TE Labs Discord
- TE Academy Research Sessions
- TE Academy Study Groups
- Outlier Venture's Discord
- Commons Stack Discord
- cadCAD Discord
- Twitter List
- Bonding Curve Research Group

Tools

The following is a list of tools commonly used in this domain, complete with links for further learning and understanding. Notable individuals working on these tools are also highlighted.

1. *cadCAD by BlockScience* — Michael Zargham, Zanecstarr (Introduction to cadCAD by Micheal Zargham, Youtube Tutorials, Model Walkthrough, Model Structure Overview, Study Group, Modelling Process, Community Forum, cadCAD study Group on TE Academy discord)
2. *Machinations* — Curiousrabbit Eth
3. *radCAD by CADLabs*
4. *TokenSpice by Ocean Protocol Team* — *Trent McConaghy* (Verification of Token Based Systems, Tools of verification)
5. *QTM by Outlier Venture's* — *Dr. Achim Struve*
6. *QTM radCAD Integration (Ongoing)* — *Dr. Achim Struve*

7. Bonding curve Research(Ongoing) — Curiousrabbit Eth
8. AI-powered Token Engineering (Ongoing) — Rohan Mehta

GitHub

Beyond the GitHub repositories for tools and models cited in this article, the following additional GitHub repositories are worth exploring.

1. https://github.com/CADLabs
2. https://github.com/BlockScience
3. https://github.com/bonding-curves
4. https://github.com/Jeiwan/uniswapv3-code/tree/main
5. https://github.com/backstop-protocol/whitepaper/blob/master/Risk%20Analysis%20Framework.pdf
6. https://github.com/A-Hitchhiker-s/Guide-to-Token-Engineering/tree/master

Courses

1. TE Academy Course
2. Software Architecture & Design of Modern Large Scale Systems
3. cadCAD Masterclass: Ethereum Validator Economics

Additional Resources

In addition to the numerous resources referenced throughout this article, the following additional materials can further deepen your understanding of this complex field.

1. Token Engineering 101 — Compiled Notes
2. Economics and Math of Token Engineering and Defi
3. Tokenomics and blockchain tokens: A design-oriented morphological framework
4. Token Ecosystem Creation Framework by Outlier Ventures
5. Complex Systems Engineering by Micheal Zargham
6. Foundations of Cryptoeconomic Systems
7. cadCAD Community Forum by BlockScience
8. Conviction Voting Model By BlockScience
9. Technical Documentation
10. Tokenomics Audit
11. TE Barcamp Ethcc Talks
12. Bonding Curve Research Group
13. Demystifying Many names of Token Engineering
14. Token Classification Framework
15. The Future of Crypto Compliance
16. The Data Scientist
17. Incentive Design and Tooling for DAOs
18. Tokenomics 101

Written by Idrees

Edit profile

21 Followers · Writer for BlockApex

More from Idrees and BlockApex

Idrees in BlockApex

Market Making Mechanics and Strategies

Introduction

10 min read · Jul 13, 2023

27

BlockApex in BlockApex

Pickle Finance Hack Analysis & POC

Introduction

2 min read · Jun 11, 2022

1

BlockApex in BlockApex

Flashbots And MEVs: A Beginner's Guide

By BlockSense

7 min read · Nov 16, 2021

6

Idrees in BlockApex

Intelligent Liquidity Provisioning Framework

Reinforcement Learning Approach to Optimize Liquidity Provisioning on Uniswap...

11 min read · Oct 28, 2023

16

See all from Idrees

See all from BlockApex

Recommended from Medium

DWF Labs Research in Hindsight Series

Diving Deep into Decentralised Perpetual Futures: An Ecosystem...

Explore the evolution of perpetual DEXs, their complex challenges and potential innovatio...

18 min read · Oct 10, 2023

31 1

AlexEuler

Onchain computation proofs for EVM

Ethereum provides two main data/call types for external RPC users: current data from...

11 min read · Dec 13, 2023

16

Lists

Staff Picks
594 stories · 791 saves

Stories to Help You Level-Up at Work
19 stories · 505 saves

Self-Improvement 101
20 stories · 1436 saves

Productivity 101
20 stories · 1321 saves

Real World Asset Tokenization

Why Tokenize RWA With Chintai?

When deciding on a partner to work with to tokenize and subsequently trade RWA, ask...

4 min read · Nov 30, 2023

CoW Protocol

What is Backrunning?—MEV Attacks Explained

Overview

6 min read · Oct 25, 2023

200 1

21

DODO in DODO

ZK Rollups On DODO

One of the most exciting advancements for DODO is the recent integration of several...

3 min read · Oct 29, 2023

7

Modulus

Track 1: Upshot's ZKML Price Predictor

Debuting the largest AI application ever implemented on Ethereum—and yes, it's...

6 min read · Nov 21, 2023

113 4

See more recommendations