MOR20:
A Standard for Recurring Protocol Payments and Payouts in Web3

TOM Anon
Rakshak Talwar LachsBagel@Proton.ME
Megan Tan megant9898@gmail.com
David A. Johnston djohnstonec@protonmail.com

Table of Contents

Abstract

Web3 currently lacks a mechanism for recurring payments, which has been crucial for the success of Web2 SaaS companies. This paper introduces MOR20, a novel standard for enabling automated recurring subscription payments in Web3 environments that span multiple blockchains and tokens. MOR20 allows projects to receive recurring revenue streams while clients only need to set up a one-time payment authorization.

The MOR20 framework builds upon primitives such as delays, weights, and contributions of capital yield, used by Morpheus' fair launch [1] [5], while introducing new mechanisms such as bonding and certifications. It enables the interoperability of any service with any other service with the creation of Protocol Engagement Tokens (PETs). Thus, allowing for the coordination of complex payment flows across multiple protocols and tokens.

By standardizing recurring payments, payouts, and proposal processes, MOR20 provides the infrastructure needed to scale Web3 economies and business models. It solves the "debit vs credit" problem that has limited Web3 adoption, allowing set-and-forget subscription models like Web2 while maintaining decentralization.

1. Introduction

Web3 and decentralized applications have enabled new models for digital services and economies. However, a key limitation has been the inability to facilitate recurring payments and subscriptions - a cornerstone of modern software-as-a-service (SaaS) business models. Current Web3 systems operate on a debit model, where users must initiate and approve each individual transaction. This creates friction for both users and service providers compared to the credit model prevalent in Web2, where users can set up automated recurring payments.

MOR20 aims to solve this problem by introducing a standard for automated recurring payments and subscriptions in Web3 environments. It enables a "set-and-forget" model for users while allowing service providers to receive steady recurring revenue streams. Critically, MOR20 achieves this while maintaining the decentralized, permissionless nature of Web3.

The key innovation of MOR20 is its ability to handle payments across multiple blockchains and tokens through a single interface. This allows Web3 services that rely on multiple protocols and blockchains to offer a seamless subscription experience to users.

MOR20 is a templatization of Morpheus [1], a decentralized network to enable broader and cost-effective access to accelerated compute to power AI use cases [2] [3] [4]. By paying out contributors on a recurring basis via emissions, Morpheus allows for a recurring flow of income for other projects.

MOR20 leverages the advanced primitives from Morpheus, specifically delays and weights, to ensure that incentives are directly aligned with the interests of value creators.

This enables more proportional and impactful reward payouts that reflect each contributor's true value added. Along with delays and weights, MOR20 introduces bonding and credentials. These allow for factoring income streams and qualifying the competence of contributors with respect to given tasks.

2. Related Work

2.1 Morpheus

Morpheus, unlike most previous Web3 projects, is not a formal entity. Instead, it was able to attract capital without conventional entity structures by implementing the Techno Capital Machine (TCM) framework first mentioned by Beff Jezos [8]. Morpheus attracted $165M worth of stETH in the first week of its fair launch which started on February 8, 2024 [6]. Morpheus then went on to attract over $400M worth of stETH by the time the MOR token debuted on May 8, 2024 [7]. The capital is always owned by the capital contributors. It is the staked yield generated which is used to create an automatic market to enable liquidity for Morpheus' MOR token. In turn, the capital contributors have rights to token emissions proportional to their capital contributed with respect to the total as well as how long they leave their capital staked in Morpheus' smart contract. By leveraging the TCM approach, Morpheus allows developers to launch and fund open-source projects without resorting to traditional corporate structures, foundations, or DAOs. Instead, Morpheus utilizes smart contracts that enable users to allocate yield from staked assets (such as stETH) directly to open-source projects of their choice. This innovative approach creates a sustainable funding mechanism where 75% of the ETH yield is used to purchase MOR, 25% of yield is paired with 1/3 of the MOR purchased, 1/3 is burned, and 1/3 is locked up for 16 years. By leveraging this model, Morpheus has created a permissionless, jurisdiction-agnostic way for open-source projects to secure funding and grow their ecosystem without the need for conventional legal entities or the associated costs and complexities [5].

3. Morpheus Abstracted is a Protocol Engagement Token (PET)

The Morpheus structure can be adopted by any project through the implementation of a Protocol Engagement Token (PET). Each PET, mirroring Morpheus itself, operates with its own native token. These PETs have the flexibility to determine how rights to token emissions are distributed, basing their allocations on the weighted importance of each contributor's input to the project.

This system offers contributors a choice in how they receive their rewards. Those who opt for delayed payouts essentially allow their contribution's value to influence the PET's token prior to any debasement, thereby positioning themselves to receive a larger share of the PET's tokens. Conversely, for contributors seeking immediate compensation, the bonding mechanism provides a solution for cash flow factoring.

Bonding agents, tasked with assessing the risk associated with bonding a contributor, can utilize the decentralized credentialing system inherent to PETs to inform their decisions. This process is elaborated upon in Section 5.1.

By seamlessly integrating these various elements, **MOR20 establishes an adaptable framework capable of efficiently managing recurring payments, factoring, payroll, and reward distributions within decentralized systems.** This comprehensive approach ensures that projects can tailor their token economics to best suit their specific needs and objectives, all while operating within a decentralized paradigm.

4. Design Principles

MOR20 is built on several key design principles:

1. Decentralization: All mechanisms should be implementable in a fully decentralized manner without relying on trusted intermediaries.
2. Composability: The standard should be modular and allow for easy integration with existing blockchain protocols and tokens.
3. Incentive Alignment: Reward structures should align the interests of contributors, users, and the long-term success of projects.

4. Flexibility: The framework should accommodate a wide range of payment models and project structures.
5. Scalability: Mechanisms should be designed to handle large numbers of users, projects, and transactions efficiently.
6. Interoperability: The standard should facilitate seamless interaction between different blockchain ecosystems. This is enabled via LayerZero [11], Wormhole [10], and Wire Network [9].
7. Self-Regulation: Where possible, parameters should be adjusted through market mechanisms rather than centralized decision-making.

5. Core Components

5.1 Protocol Engagement Token (PET)

PETs are non-fungible tokens that represent specific projects or proposals within the MOR20 ecosystem. Examples are mentioned in Section 7.

Any protocol wishing to interoperate with others would adopt the PET standard to issue their own token on a recurring basis. A PET can choose how it wants to divide the total token emissions between groups of contributors. Higher value contributions command a greater number of weights. The fraction a contributor's weights are to the total number of weights, represents their proportional rights to those token emissions. Contributors opting for longer delays would receive greater proportional rights.

They encapsulate the following attributes:

- Delay: Time elapsed before the protocol pays a party through emissions
- Proportionality: Scaling ratio to incentivize greater rewards towards parties who select longer delays
- Deadline: Time required for vendor to deliver service/product
- Bond: Collateral posted to ensure delivery
- Weights: Rights to payments or distributions
- Credential: Non-transferable historical proof of completed work

5.1.2 Debasement Without Inflation

A key feature of MOR20 is its ability to provide immediate cash flow to contributors before their work generates value for the broader community. While token emission is inherently inflationary, MOR20 addresses this through bonding and credentials.

Contributors seeking immediate cash flow can receive token emissions if they secure a bonding agent to post collateral against their future contribution. The bonding agent posts a bond in the native token of the PET, equal to the value of the contribution. The contributor then receives emissions slightly below the bonding amount, with the remainder emitted to the bonding agent. The discount rate is set by the market. Factors such as delivery risk would impact such discount rate. The bonding agent profits when the contribution is successfully delivered.

Upon successful completion of the contribution:

1. The bonding agent's bond is released.
2. The emissions enable debasement of the PET token without inflation, as the contribution brings additional value.
3. A credential token is issued to the contributor, reflecting the positive outcome.

If the contribution is not delivered successfully:

1. The bond is locked, negating the inflationary effect of the contributor's emissions.
2. A credential token is issued to the contributor, reflecting the negative outcome.

This system creates a transparent due diligence process for clients seeking contributions and helps bonding agents estimate delivery risks. The credential tokens serve as a permanent record of a contributor's performance, facilitating future collaborations and risk assessments. PETs allow for granular management of project-specific incentives and requirements while operating within the broader MOR20 framework.

5.2 Protocol Engagement Token Coordinator (PETCo)

PETCos are entities that manage multiple PETs, enabling:

- Multi-MOR20 product income management
- API simplification for interacting with multiple protocols
- Reward difficulty adjustment
- Credential activation thresholds
- Cross-chain coordination

PETCos abstract away the complexity of multi-protocol interactions, allowing for more efficient and user-friendly decentralized applications.

5.3 Difficulty Adjustment Mechanism

MOR20 implements a market-driven difficulty adjustment that modifies reward proportionality based on factors such as:

- Delay period length
- Type of contribution (in Morpheus' case, these would be code, compute, builder, or capital [5])
- Overall system demand

Given that token emissions in a PET follow a predetermined schedule. The adjustment in compensation between contributors is their right to the proportion of those emissions. Their right to emissions then is their proportionality divided by the sum of all contributor proportionalities. For Morpheus, a PET, the following Formulas 1 and 2 are used to compute the contributor rights. The tanh function and its coefficient models the predetermined emissions schedule of the MOR token. Formula 1 allows for accurately compensating contributors based on the actual dilution they would be facing. It adjusts proportionality with respect to the time of entry and length of time staked. Note that the current minimum and maximum return of the proportionality equation is clamped between a power factor of 1.0 and 10.7. A Staking length of 6 years in the case of Morpheus generates the max power factor of 10.7.

Proportionality equation

Note: that emissions in Morpheus are computed every second. For simplicity, the following examples are shown by the day, however the equations are formulated at second intervals.

$$P(s_{start}, s_{end}; T_{start}, T_{end}) = \mathrm{clamp}\left(P_{\max} \cdot \left[\tanh\left(2 \cdot \frac{s_{\text{end}} - T_{\text{start}}}{T_{\text{end}} - T_{\text{start}}}\right) - \tanh\left(2 \cdot \frac{s_{\text{start}} - T_{\text{start}}}{T_{\text{end}} - T_{\text{start}}}\right)\right], 1, 10.7\right) \quad \texttt{(Formula 1)}$$

Definitions:

> $P(s_{start}, s_{end})$: proportion calculated for a span of time from s_{start} to s_{end}
> P_{max}: the max power used to define the hyperbolic tangent cure, set to $P_{max} = 16.61327546$
> T_{start}: the unix timestamp for July 25, 2024 12:00 UTC
> T_{end}: the unix timestamp for January 26, 2040 12:00 UTC
> s_{start}: the unix timestamp for when a contributor begins the staking period
> s_{end}: the unix timestamp for when a contributor ends the staking period

Where:

> $s_{start} >= T_{start}$
> $s_{end} <= T_{end}$
> $P(s_{start}, s_{end}; T_{start}, T_{end}) \propto s_{end} - s_{start}$

Note that although $P_{max} > 16$, the value returned by the proportionality equation is clamped between 1 and 10.7.

A hyperbolic tangent function, $tanh(.)$, is used as it models the dilution rate from MOR emissions over time.

Emissions earned by contributor c based on their Proportionality (Relative Power) and Weight Allocation

$$E_{\substack{\mathrm{T=t} \\ C=c}} = (0.24) \cdot (E_{T=t}^{\text{total}}) \cdot \left(\frac{P_c W_c}{\sum_{i=1}^{C} P_{i,t} W_i} \right) \qquad \texttt{(Formula 2)}$$

Definitions:

> $E_{\substack{\mathrm{T=t} \\ C=c}}$: Emissions in MOR tokens earned by contributor c out of all contributors C for the given bucket $b \in B$
>
> $B \in \{B_{code}, B_{capital}, B_{compute}, B_{community}\}$: a bucket from any of the code, compute, capital, or community buckets
>
> $E_{T=t}^{\text{total}}$: All MOR emissions produced on day t
>
> (0.24): coefficient to scale down all emissions $E_{T=t}^{\text{total}}$ for the emissions calculation of a single bucket B
>
> $P_c = P(s_{start}, s_{end}; T_{start}, T_{end})$, for contributor c based on their chosen interval $[s_{start}, s_{end}]$
>
> W_c: amount of weights or capital for contributor c
>
> $\sum_{i=1}^{C} P_{i,t} W_i$: sum of all products of proportionality and weights for all contributors with respect to a bucket B

Contributors who come in earlier and opt for greater delays in claimability of Morpheus MOR token emissions would justifiably benefit from higher proportionality ratios. A contributor who contributes by July 25, 2024, and opts for a 2-year delay in claiming token emissions would receive a proportionality of ~4.2, whereas another contributor delivering by July 25, 2026, and opting for a 1-year delay would only have a proportionality of ~1.9, a difference of over double in relative rights to emissions for the same contribution. This can be seen in Figures 1 and 2.

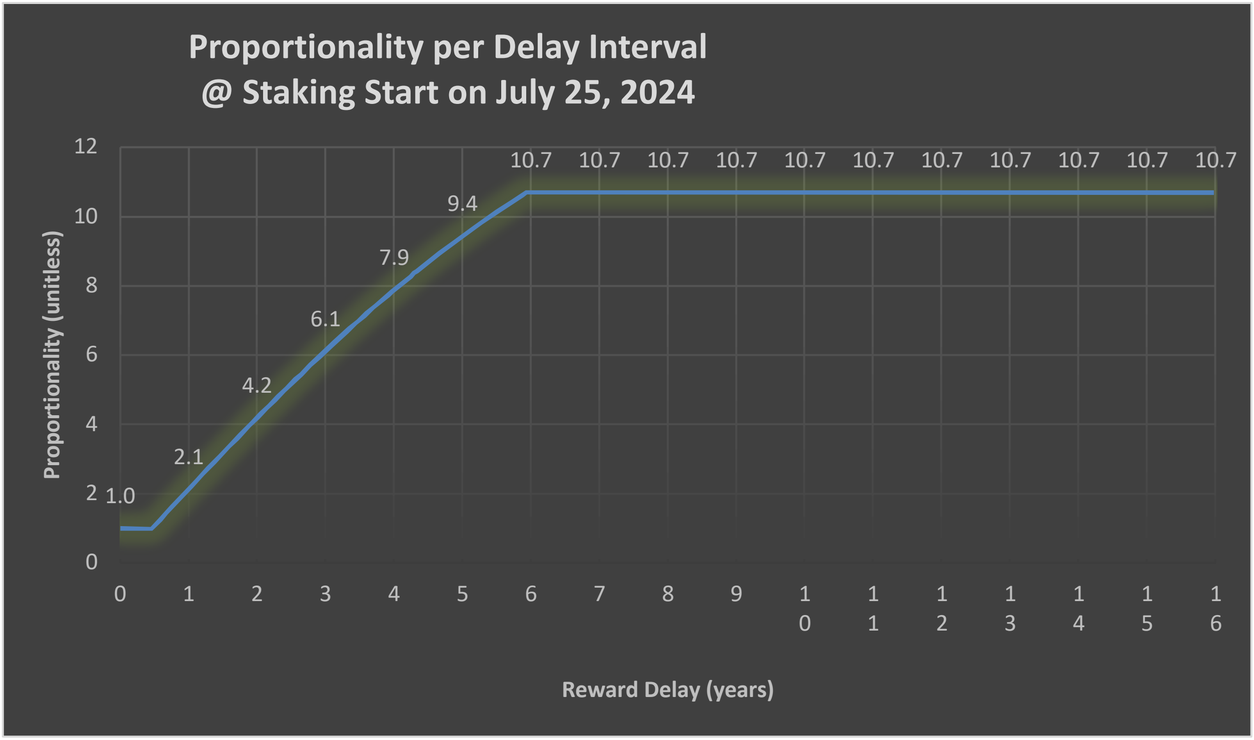

Figure 1: Proportionality obtained given contribution is delivered by July 25, 2024, with respect to the contributor's chosen delay period.

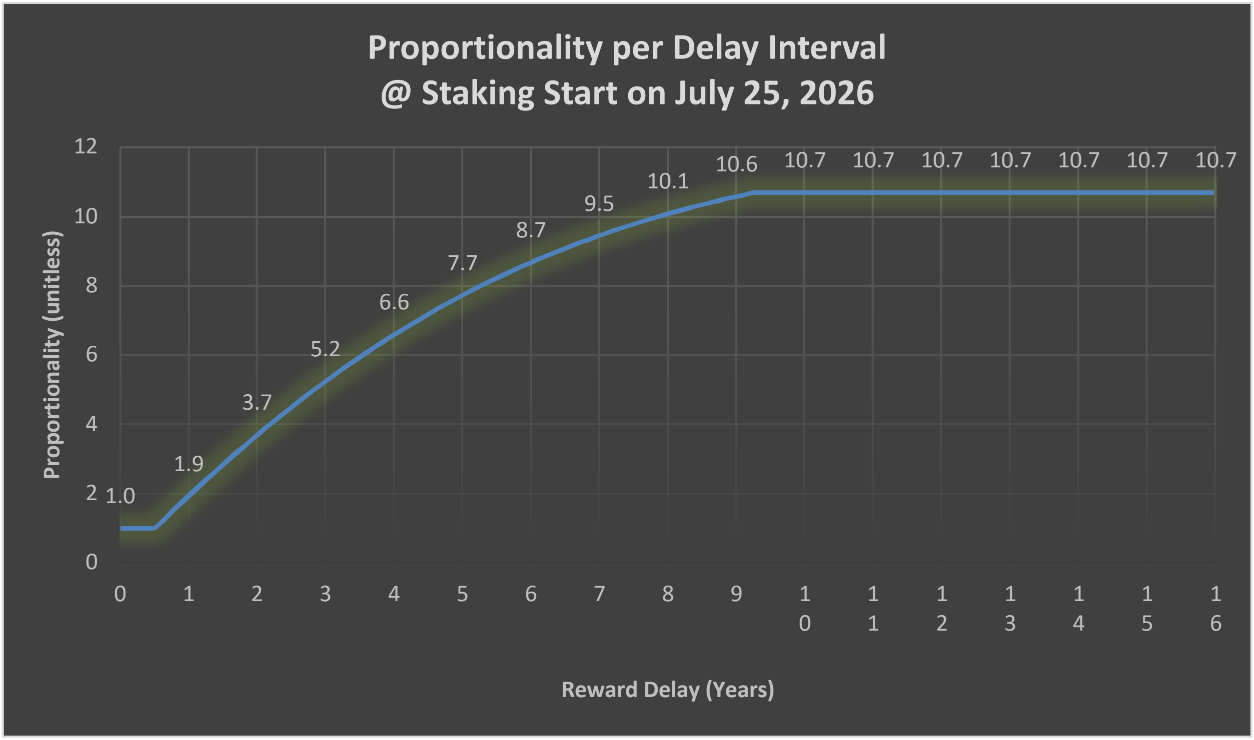

Figure 2: Proportionality obtained given contribution is delivered by July 25, 2026, with respect to the contributor's chosen delay period.

5.4 Credential System

The credential system provides non-transferable on-chain recognition of completed work and accumulated expertise. Key features include:

- Qualification-based bidding for projects.
- Portable reputation across different PETs and protocols.
- Granular representation of different skills and experiences.

An emergent property of decentralized credentialing should show different success rates for different skillsets. Advanced scientific research projects would likely comprise of competent research contributors with lower successful to unsuccessful ratios of credentials compared to contributors performing deterministic work, such as front-end web development. PETs and bonding agents would be able to transparently see that the global pool of other contributors for a given task fall within a certain expected ratio of success to fail, and thus adjust their risk calculations accordingly.

5.5 Factoring Capabilities

MOR20 enables the selling of future cash flows at a discount, providing liquidity options for contributors. This is achieved through:

- Tokenization of future revenue streams
- Bonding mechanisms for advance payments

- Market-driven pricing of discounted cash flows

6. Mechanisms

6.1 Recurring Payments

MOR20 enables recurring payments through the following process:

1. User authorizes a subscription using stETH tokens.
2. Smart contract holds authorized funds in escrow.
3. Periodic distributions are made to service providers based on predefined rules.
4. Users can cancel or modify subscriptions with notice periods defined in the smart contract.

6.2 Reward Distribution Types

There are two primary ways to enable recurring payments. First by issuing a new token or second by staking an existing token.

1. **MRC20 Project: Creation of a New Fungible Token**

   Example: Nounspace.com [12]

   - New ERC20 style token called $SPACE is created [13].
   - Contributors to Nounspace stake stETH to earn $SPACE tokens.
   - Those holding SPACE tokens can access the Nounspace platform.
   - Yield is divided between Nounspace and Morpheus. Given that stETH is currently generating ~3.5% annual yield. The Morpheus platform receives 0.35% of yield for its protocol owned liquidity, with the remaining 3.15% yield flowing to Nounspace.

2. **MRC21 Project: Staking an Existing Token**

   Example: Venice.ai [14]

   - ERC 721 style Non-Fungible Token representing access / utility is created.
   - Venice customers buy MOR tokens and stake towards Venice's address.
   - Venice grants customers Venice Pro access.
   - MOR proceeds are sent to Venice by the Morpheus protocol. These rewards originate from the 24% builder rewards emissions bucket [1].

6.3 Bonding

Bonding allows for advance payments while mitigating risk:

1. Contributors or third-party bonding agents stake collateral
2. Advance payments are made based on the bonded amount
3. Bonds are released upon successful completion of work

4. Bonds can be locked for non-delivery or subpar performance

6.4 Credential Issuance and Verification

Credentials are issued and verified as follows:
1. Completion of work is recorded on-chain
2. Non-transferable credential NFT is minted to the contributor's address
3. Future projects can specify required credentials for bidding
4. Smart contracts verify credential ownership for bid eligibility

7. Applications

MOR20 enables a wide range of applications across different domains:

7.1 Decentralized Software-as-a-Service (DSaaS)

- Subscription-based access to decentralized applications.
- Multi-protocol service bundles with single payment interface.
- Usage-based billing for blockchain resources.

7.2 Decentralized Autonomous Organizations (DAOs)

- Recurring membership fees and benefit distribution.
- Reputation-based governance participation.
- Cross-chain treasury management and project funding.

7.3 Decentralized Finance (DeFi)

- Automated yield distribution for liquidity providers.
- Subscription-based access to premium trading features.
- Cross-chain yield farming strategies.

7.4 Creator Economies

- Fan subscription models for content creators.
- Automated royalty distribution for collaborative works.
- Credential-based access to exclusive content or experiences.

7.5 Decentralized Compute Markets

- Subscription access to distributed computing resources.
- Automated payment distribution to compute providers.
- Reputation-based prioritization of compute tasks.

Morpheus itself being a set of decentralized markets for capital, code, compute & builders.

8. Challenges and Future Work

While MOR20 presents a promising framework for Web3 payments and engagement, several challenges and areas for future research remain:

1. Scalability: Ensuring efficient operation with large numbers of users and complex multi-token flows.
2. User Experience: Developing intuitive interfaces for managing subscriptions and credentials.
3. Interoperability: Expanding compatibility with a wider range of blockchain protocols and token standards.
4. Economic Modeling: Refining reward curves and incentive structures based on real-world usage data.
5. Privacy: Implementing zero-knowledge proofs or other privacy-preserving technologies for credential verification.

9. Conclusion

The MOR20 standard introduces a comprehensive framework for handling recurring payments, reward distributions, and engagement in decentralized systems. By addressing key challenges in Web3 payments and protocol interaction, MOR20 enables more efficient and flexible economic models for decentralized applications and services.

Key innovations of the MOR20 standard include set-and-forget recurring payments, sophisticated reward systems with adjustable parameters, Protocol Engagement Tokens (PETs) and Coordinators (PETCos), difficulty adjustment mechanisms via delays, non-transferable credentials, and factoring capabilities.

As the Web3 ecosystem continues to evolve, standards like MOR20 will play a crucial role in enabling the next generation of decentralized services and economies. Future work will focus on implementing and testing these concepts in real-world applications, refining the economic models, and developing user-friendly interfaces to make these complex systems accessible to a wider audience.

References

[1] Morpheus, Trinity, and Neo, *Morpheus: A Network for Powering Smart Agents*, 2023.
https://github.com/MorpheusAIs/Docs/blob/180b63c824107085c76bc0f373cd4750137e957f/!KEYDOCS%20README%20FIRST!/WhitePaper.md

[2] Voorhees, Erik, *Morpheus: "Yellowstone" Compute Model*, 2024.
https://github.com/MorpheusAIs/Docs/blob/main/!KEYDOCS%20README%20FIRST!/Compute%20Providers/Yellowstone%20Compute%20Model.md

[3] Talwar, Rakshak, *Lake Travis Decentralized AI Inference System,* 2024.
https://github.com/MorpheusAIs/Docs/blob/180b63c824107085c76bc0f373cd4750137e957f/!KEYDOCS%20README%20FIRST!/Compute%20Providers/Lake%20Travis%20Decentralized%20AI%20Inference%20System.md

[4] Condron, Ryan, *Morpheus Lumerin Model*, 2024.
https://github.com/MorpheusAIs/Docs/blob/180b63c824107085c76bc0f373cd4750137e957f/!KEYDOCS%20README%20FIRST!/Compute%20Providers/Morpheus%20Lumerin%20Model.md

[5] Johnston, David, *Techno Capital Machine (TCM)***:** Code, Capital, Community, Compute: The Open Source Revolution Starts Here, 2024.
https://github.com/MorpheusAIs/TCM/blob/3925ea9e926a8502ec4e28c915e4caf1bfc86bf0/README.md

[6] Shaughnessy, Tom, *MorpheusAI: New Crypto x AI Project Attracts 60,000 stETH*, 2024.
https://members.delphidigital.io/feed/morpheusai-new-crypto-x-ai-project-attracts-60000-steth

[7] Shaughnessy, Tom, *Morpheus - Crypto x AI Project's Token Goes Live May 8*, 2024.
https://members.delphidigital.io/feed/morpheus-crypto-x-ai-projects-token-goes-live-may-8

[8] @BasedBeffJezos, Accessed July 2024
https://x.com/BasedBeffJezos

[9] Wire Network, Accessed July 2024
https://www.wire.network

[10] Wormhole, Accessed July 2024
https://wormhole.com

[11] Layer0, Accessed July 2024
https://layerzero.network

[12] Nounspace, Accessed July 2024
https://nounspace.com

[13] Nounspace Token $SPACE, Accessed July 2024
https://space.nounspace.com/#/mainnet/capital

[14] Venice.ai, *Private and Uncensored AI*, Accessed July 2024
https://venice.ai/