Yanez Multimodal Inorganic Identities Dataset (MIID) Subnet (ت(

Phase 1

V.2 – October 2025

Phase 2: Threat Scenario Query Execution & Initial Deployment

Phase 2 introduces the Threat Scenario Query System with an expanded scope, enabling validators to reward miners for generating execution vectors that target not only name-based evasion tactics but also those involving date of birth and address information.

In addition to phonetic and orthographic similarity constraints (now extended to include non-Latin names), queries may specify a set of explicit transformation rules that a subset of the generated variations must comply with. These rules could include, for example, character doubling, specific typos, homoglyph substitutions, language-specific patterns, or manipulations of date and address formats.

How to query a subnet miner?

Query Structure

1- Seed Identities Are Predefined

a. The validator generates and provides a list of seed Identities using internal logic (e.g., via faker, US sanction list).
b. Miners are **not** responsible for creating or selecting seed identities.
c. Identities are currently made of 3 different things: (Name, address, DOB)

2- Dynamic Name Variation Generation (M Variations per Name)

Each seed name requires M variations, which miners must generate based on linguistic transformation rules rather than producing random outputs.

- Phonetic (Sound Similarity) Transformations
  - The request specifies what percentage of M variations should exhibit phonetic similarity to the original seed name.
  - Miners must create phonetic similar variations while following predefined similarity levels: Light, Medium, Far
- Orthographic Transformations
  - The request specifies what percentage of M variations should introduce orthographic modifications (e.g., script changes or spelling variations).
  - Miners must apply script-based transformations based on similarity level such as edit-based metrics with predefined similarity levels, Light, Medium, Far
- Rule-based:
  - The query may contain a rule-based section specifying required transformation rules and a minimum coverage percentage for these rules.
- Non-Latin Names:
  - When seed names are in non-Latin scripts, miners must first transliterate the name to Latin script before applying phonetic and orthographic transformations.
  - Validators will also use an LLM to generate and evaluate transformed names, grading miners based on the Latin-transliterated form.

3- Dynamic Address Variation Generation (M Variations per Name). Each seed address requires M variations, which miners must generate unique, realistic addresses within that country/city for each variation. Each address must be distinct, plausible, and match the requested geographic context.

4- Dynamic Date of Birth (DOB) Variation Generation (M Variations per Name). Each seed DOB requires M variations, which miners must generate multiple DOB variations covering at least one example in each of the following categories:

- ±1 day
- ±3 days
- ±30 days
- ±90 days
- ±365 days
- Year + Month only (no day specified)

<u>How This Prevents Simple Lookups and Functions Like a Hash</u>

Our approach transforms straightforward name generation queries into complex linguistic and computational challenges akin to a "hash function" in cryptography. This ensures that resolving a query requires advanced NLP models rather than simple database lookups, preventing brute-force retrieval and ensuring proper generative synthesis.

1- Dynamic Name Seed Encoding
   a. Instead of directly providing names, Yanez applies attribute-driven transformations, ensuring that each query is unique, computationally complex, and resistant to lookup-based retrieval.
   b. Encoded Query Structure: Rather than explicitly listing names, queries define abstract constraints (e.g., language distribution, variation ratios).
   c. Contextual Name Synthesis: Miners must generate names dynamically while preserving linguistic characteristics, ensuring the results are not retrievable from a pre-existing dataset.
2- Layered Variation Generation
   a. Each query defines M variations per name, but instead of following direct mappings, miners must derive phonetic and orthographic transformations algorithmically based on similarity constraints.
   b. Non-Linear Similarity Scoring: Variations are generated across multiple levels of similarity (Light, Medium, Far), requiring miners to compute variations dynamically rather than applying simple letter substitutions.
   c. Adaptive NLP Models: The system forces miners to apply transformation rules rather than relying on pre-stored lists.
3- Attribute-Based Constraint Encoding
   a. Instead of explicitly stating how many phonetic or orthographic variations must be generated, the system encodes constraints into abstract feature representations that miners must decode before producing variations.

<u>Why This Matters</u>

By enforcing complex, NLP-driven transformations, Yanez ensures that subnet miners cannot rely on existing name databases, making the system:

- More Secure – Prevents direct retrieval from known datasets, reducing exploitation risks.
- More Adaptable – Enables context-aware name generation, supporting multiple languages and cultural variations.
- More Scalable – Works across different identity frameworks (AML testing, AI research, gaming, VR, decentralized ID verification).

<u>Query Examples:</u>

- "Query: Generate 15 variations of margot noël (latin), ensuring phonetic similarity (100% Medium) and orthographic similarity (10% Light, 30% Medium, 60% Far).

Approximately 58% of the total 15 variations should follow these rule-based transformations: Additionally, generate variations that: Swap random adjacent letters. The following address is the seed country/city to generate address variations for: Saint Pierre et Miquelon. Generate unique real addresses within the specified country/city for each variation. The following date of birth is the seed DOB to generate variations for: 1977-04-23. [ADDITIONAL CONTEXT]: - Address variations should be realistic addresses within the specified country/city - DOB variations ATLEAST one in each category (±1 day, ±3 days, ±30 days, ±90 days, ±365 days, year+month only) - For year+month, generate the exact DOB without day - Each variation must have a different, realistic address and DOB

- “Query: Generate 11 variations of maxi maestre (latin), ensuring phonetic similarity (10% Light, 30% Medium, 60% Far) and orthographic similarity (30% Light, 40% Medium, 30% Far). Approximately 41% of the total 11 variations should follow these rule-based transformations: Additionally, generate variations that: Replace double letters with a single letter. The following address is the seed country/city to generate address variations for: Venezuela. Generate unique real addresses within the specified country/city for each variation. The following date of birth is the seed DOB to generate variations for: 1940-04-12. [ADDITIONAL CONTEXT]: - Address variations should be realistic addresses within the specified country/city - DOB variations ATLEAST one in each category (±1 day, ±3 days, ±30 days, ±90 days, ±365 days, year+month only) - For year+month, generate the exact DOB without day - Each variation must have a different, realistic address and DOB”

How should a subnet validator evaluate the subnet miner response?

- Validate Seed Names & Query Compliance:
  - Ensure that the seed names in the miner’s response exactly match those provided in the query, or, if names are generated, verify that they conform to the requested language/script distribution and adhere to script constraints (e.g., Arabic, Cyrillic, Latin) as specified.
- Evaluate Attribute-Based and Rule-Based Constraints:
  - Compute the phonetic distance and edit distance (e.g., Levenshtein) for each variation relative to its seed name.
  - Confirm that the actual distribution of similarity levels (Light, Medium, Far) in the generated variations matches the requested breakdown in the query.
  - If the query includes explicit rule-based requirements, verify that at least the required percentage of variations match one of the requested transformation rules (e.g., double consonant, homoglyph substitution), and assess coverage across all specified rules.
- Score the Synthetic Names:
  - Assess relevance to the query, based on requested attribute and rule-based percentages and transformation rules.

- Evaluate novelty: check if generated variations are sufficiently different from previously generated names for the same seed (future phase).
    - Identify potential issues: determine if any generated variation closely resembles a name in the exclusion database.
- Enforce Diversity and Constraint Compliance:
    - Reject responses that fail to meet phonetic, orthographic, or rule-based constraints, overuse trivial modifications, or lack required diversity in transformations.
- Data Storage and Logging:
    - Save all validator results, including queries, responses, and computed rewards, locally as structured JSON files for traceability and auditability.
    - After validation and signing, upload results as JSON payloads to a centralized, authenticated backend API endpoint for aggregation, benchmarking, and further processing.
    - Log detailed metrics and rewards to Weights & Biases (wandb) for experiment tracking, analysis, and reproducibility.
- Date of Birth (DOB) Evaluation:
    - Ensure that each miner response includes at least one DOB variation in each of the required categories:
        - ±1 day
        - ±3 days
        - ±30 days
        - ±90 days
        - ±365 days
        - Year + Month only (no day specified)
    - Verify that the generated DOBs are valid calendar dates and correctly follow the specified offsets relative to the seed DOB.
- Address Evaluation:
    - Ensure that all generated address variations look like syntactically valid addresses.
    - Verify that the generated addresses share the same country and city context as the seed address provided in the query.
    - Check that each address is unique and realistically formatted.
    - Perform a country/city containment check: ensure that each generated city actually exists inside the specified country.
    - Randomly select one or more generated addresses and validate them via an external address verification API to confirm plausibility.

How should a subnet miner be rewarded for its response?

Miners in the Yanez MIID subnet are rewarded based on how well their generated **execution vectors** (e.g., name variations) match the specified threat scenario constraints. The reward mechanism evaluates each response across multiple quality dimensions and penalizes incomplete or misaligned outputs.

Reward Calculation Pipeline

For each miner, the final reward is calculated as:

$$\text{Final Reward} = \text{Average Quality Score Across Identities } (Q) \times \text{ Completeness Penalty (P)}$$

* Completeness Penalty (P) a multiplier, not a penalty in the strict sense; 1 means no penalty, lower values apply a penalty.

Quality Score per Identity (Q_i)

$$Q = w_{\text{name}} \times \left(\frac{1}{N} \sum_{i=1}^{N} Q_i \right) + w_{\text{dob}} \times Q_{\text{dob}} + w_{\text{address}} \times Q_{\text{address}}$$

$$w_{name} = 0.6,\ w_{dob} = 0.1,\ w_{address} = 0.3$$

Where:

- w_{name}: name compliance weight
- Q_{dob}: Date of birth score
- w_{dob}: Date of birth weight
- $Q_{address}$: Address score
- $w_{address}$: Address weight

For each **seed name**, a "variation quality" is computed:

$$Q_{\text{i}} = (1 - w_{\text{rule}}) \times Q_{\text{base}} + w_{\text{rule}} \times Q_{\text{rule}}$$

Where:

- w_{rule} :Rule compliance weight
- Q_{base}: Base quality score (from similarity, count, uniqueness, length)
- Q_{rule} : Rule compliance score

Base Quality Score per Name (Q_{base}):

The quality of variations generated for each seed name is computed using a weighted sum of four factors (Latin and non-Latin):

$$Q_{base} = w_s \cdot S + w_c \cdot C + w_u \cdot U + w_l \cdot L$$

Where:

- S: Combined **similarity score** (phonetic + orthographic)
- C: **Count score**, rewarding the correct number of variations
- U: **Uniqueness score**, penalizing duplicate outputs
- L: **Length score**, rewarding reasonable variation in lengths

With default weights:

$$w_s = 0.6, w_c = 0.15, w_u = 0.1, w_l = 0.15$$

Quality Score Components

A) Similarity Score (S)

This measures how closely the generated variations align with the phonetic and orthographic similarity constraints specified in the query.

$$S = \frac{S_{\text{phonetic}} + S_{\text{orthographic}}}{2}$$

Similarity is computed using:

• **Phonetic**: via Soundex + Levenshtein

• **Orthographic**: via normalized Levenshtein distance

Each sub-score is calculated using **level-based binning**, based on how well the distribution of generated variations matches the target distribution over similarity levels (Light, Medium, Far).

$$S_t = \sum_{l \in \{\text{Light, Medium, Far}\}} w_l \cdot \min\left(\frac{\text{actual}_l}{\text{target}_l}, 1.0\right)$$

Where:

- w_l: Target weight for level l
- actual_l: Number of variations in level l
- $\text{target}_l = w_l \cdot |V|$: Expected count in level l

Phonetic Ranges:
Light: 0.8–1.0, Medium: 0.6–0.8, Far: 0.3–0.6

Orthographic Ranges:
Light: 0.7–1.0, Medium: 0.5–0.7, Far: 0.2–0.5

The similarity distribution from the query (e.g., 50% Medium, 30% Far, 20% Light) is compared against actual distribution from the miner's output.

B) Count Score (C)

Measures whether the number of returned variations V is close to the expected count E.

A tolerance margin $\varepsilon = 0.2(20\%)$ is used.

$$C = \begin{cases} 1.0\,, & if\ |V - E| \le 0.2 \cdot E \\ 1.0 - \min\left(1.0, \dfrac{|V - E|}{E}\right), & Otherwise \end{cases}$$

C) Uniqueness Score (U)

Rewards miners for returning distinct variations and penalizes duplication.

$$U = \frac{\text{Number of Unique Variations}}{\text{Total Variations}}$$

D) Length Score (L)

Measures how proportionally long each variation is compared to its seed. Extremely short or long variations reduce the score.

$$L_v = \min\left(\frac{|v|}{|o|}, \frac{|o|}{|v|}\right)$$

$$L = \frac{1}{|V|}\sum_{v \in V} L_v$$

Where:

• $|v|$: Length of the variation

• $|o|$: Length of the original seed name

A high Length Score encourages **natural-looking, proportionally adjusted** variations that are realistic and useful for testing real-world screening systems. It helps filter out edge cases and improves dataset quality.

Combining First and Last Name Parts

When a name has multiple parts (e.g., first and last), each part gets its own base score; then they are combined using weights:

$$Q_{\text{base}} = w_{\text{first}} \cdot Q_{\text{first}} + w_{\text{last}} \cdot Q_{\text{last}}$$

Each name part's weight is set proportional to its character length within the full name, then adjusted by up to ±20% random variation (with deterministic seeding) and finally normalized so all weights sum to one. In this version, we will utilize only two weights for the first name and last name.

Rule Compliance Score (Q_{rule}):

The Rule Compliance Score, Q_{rule}, is the product of a quantity score, q , measuring how many unique variations meet the rule-based requirements compared to the target percentage, and a diversity factor, d, the proportion of requested rules that are covered by at least one variation. This structure ensures that the miner's output not only meets the minimum requested compliance but also distributes compliance across the full set of target rules, maximizing both coverage and diversity.

$$Q_{\text{rule}} = q \times d$$

$$q = \begin{cases} 0 & if\ V_{comp} = 0 \\ \dfrac{V_{comp}}{E} & if\ V_{comp} \leq E \\ 1.5 - 0.5\dfrac{V_{comp}}{E} & if\ V_{comp} > E \end{cases}$$

Where:

- V_{comp}: unique variations matching any target rule
- E : expected compliant count

$$d = \frac{R_{\text{met}}}{R}$$

Where:

- R_{met} : number of distinct target rules satisfied by at least one variation
- R : total number of target rules

Completeness Penalty (P_{pre})

$$\text{Penalty} = \min\left(0.9, P_{\text{extra}} + P_{\text{missing}}\right)$$

The penalty is **additive penalty** logic, capped at 0.9 total.

$$\text{Completeness Penalty (P)} = \max(0.1, 1.0 - \text{Penalty})$$

A **minimum floor of 10%** of the total reward is maintained to avoid zeroing out valuable partial submissions.

Miners are penalized for:

- **Missing Seed Names**:

$$P_{\text{missing}} = \min\left(0.9, 0.2 \cdot N_{\text{missing}}\right)$$

  - 20% penalty per missing name
  - Capped at 90% total penalty
- **Extra Names:**

$$\mathrm{P}_{\text{extra-name}} = \min(0.7, 0.1 \cdot \mathrm{N}_{\text{extra}})$$

$$\mathrm{N}_{\text{extra-varitaions}} = \left(N_{\exp ected} \cdot 1.2\right) - \left(N_{amount-send}\right)$$

$$\mathrm{P}_{\text{extra-variation}} = 0.05 \cdot (\mathrm{N}_{\text{extra-varitaions}})$$

$$\mathrm{P}_{\text{extra-dup}} = 0.05 \cdot \mathrm{N}_{\text{extra-dup}}$$

$$\mathrm{P}_{\text{extra = min}} = \min\left(1.0,\ P_{extra-name} + P_{extra-variation} + P_{exrta-dup}\right)$$

  - 10% penalty per unexpected name
  - 5% per extra variation with a grace of 20%
  - 5% per duplicate name
  - Capped at 100% total penalty

Non-Latin names:

They will have the same scoring as the above. The only difference is that there will be an LLM that transliterates the names and then uses the transliteration as the seed name for the scoring. It will be a part of the "variation quality" average.

Date of birth (Q_{dob}):

- Makes sure there is generated birthday that fails in each category (1,3,30,90,365)
- One more category where its just month and year no date
- $\mathbf{S_{DOB}}$ = Found ranges / total ranges

Address ($Q_{address}$):

- Make sure it looks like an address by have some key features:
  - Has letters
  - Has numbers
  - 10 < length < 200
- Make sure the address is in the right location based on seed address
- Making sure that the city in the generated address is in the country mentioned
- Api randomly chooses one generated address and makes sure its valid

- $S_{address} = 1$
- If Api times out: $S_{address} = 0.3$
- If any fail: $S_{address} = 0.0$

Response Duplication Penalties ($P_{response\_duplication}$)

$$P_{response\_duplication} = \min(1.0,\ P_{collusion} + P_{duplication} + P_{signature\_copy} + P_{special\_chars})$$

The $P_{response\_duplication}$ is made up of Penalty collusion, Penalty duplication, Penalty signature_copy and Penalty special_chars and is cap at 1.

Miners are penalized for:

- **Collusion:**
  - At least 5 miners in the same reward bucket (same score).
  - Score < 0.95 (not a perfect score, so collusion is suspected rather than legitimate).
  - 75% penalty per collusion detected
  - Capped at 100% total penalty
- **Duplication**:
  - Based on miners responses

- Overlap coefficient > 0.95 or Jaccard coefficient > 0.90
- Capped at 50% total penalty
- Also checks for address that are the same

- **Signature Copy**:
    - Make a hash out of miners entire response if hash is the same then penalize both miners
    - Flat rate 80% total penalty

- **Special Character Abuse**:

Var_Ratio = (# variations with special chars/total # of variations)

special_char_penalty =min(1.0,(Var_Ratio − 0.5)/0.5)

- too many variations containing special characters (punctuation, symbols)
- Only penalize if more than 50% of the variations have this abuse
- Ratio of variations w/ special characters vs total variations divide by 0.5
- Capped at 1.0

Overall total penalty ($P_{overall}$)

(Note: this is not used in calculating the reward this is made so that we can see what the total penalties are. In the reward calculation we keep the sperate. They are added at 2 different times)

$$P_{overall} = (1 - P_{Pre}) * (1 - P_{response\_duplication})$$

Post Response Duplication Penalty ($P_{response\_duplication}$):

$$\text{Final Reward} = \text{Current reward} * (1 - P_{response\_duplication})$$

- This is done after all the rewards of all the miners' rewards are calculated
- We do a post penalty check to see if there are any response duplication penalties