X^{∞} – Postmoral and Emotionless Mathematical Foundations of Ethical Governance as a Self-Reinforcing System (Working Paper, Version 2.1)

The Author
x_to_the_power_of_infinity@protonmail.com
https://mastodon.social/@The_Auctor
https://www.linkedin.com/in/der-auctor-b12375362/
zenodo.org
GitHub: Xtothepowerofinfinity

May 06, 2025

The X^{∞} system redefines responsibility: not as a moral category, but as a mathematically regulated effect. It resolves classical ethical dilemmas through a self-reinforcing model that governs power via feedback and protective mechanisms. This text formalizes the system's foundations, focusing on Cap logic (authority), feedback penalties, and antispeciesism. The goal is a robust, transparent system that distributes responsibility without moral assumptions.

Inhaltsverzeichnis

1 Introduction **3**
1.1 System Goals . . . 3
1.2 Document Structure . . . 3

2 Responsibility as Effect **3**
2.1 Postmoral Perspective . . . 3
2.2 Systemic Consequences . . . 3

3 System Architecture **4**
3.1 Flow of Responsibility . . . 4
3.2 Learning Capability . . . 4

4 Basic Concepts **5**

5 Systemic Origin Equation and Role of the UdU **5**

6 Mathematical Framework **6**
6.1 Cap Logic . . . 6
6.2 Historical Authority (Cap_{Past}) . . . 6
6.3 Future Capability ($Cap_{Potential}$) . . . 7
6.4 Protective Mechanisms ($Cap_{Protection}$) . . . 8
6.5 Capability Profile (M_{Pot}) . . . 8
6.6 Active Responsibility (Cap_{Real}) . . . 9
6.7 Petitions . . . 9
6.8 Feedback through Rückkopplung ($Cap_{Feedback}$) . . . 9
6.9 Feedback Penalties and Oversteering . . . 10
6.10 System Limits . . . 10
6.11 Global Potential Limit . . . 11
6.12 Delegation Validity . . . 11
6.13 Special Case UdU . . . 11

7 Antispeciesism **11**

8 Conclusion **11**
8.1 License . . . 12

Introduction

The X^∞ system stems from a radical premise: moral categories like "good" or "evil" are unsuitable for regulating responsibility in complex systems. Instead, X^∞ defines responsibility as *effect* – measurable, mathematically representable, and systemically regulable. This leads to a *postmoral* approach that replaces ethical governance with structural mechanisms.

System Goals

- **Fairness through Effect**: Responsibility is distributed not by intent, but by actual consequences.
- **Protection of the Weakest**: Weaker actors gain a stronger voice through feedback weighting.
- **Antispeciesism**: Humans, non-humans, and the environment are treated as equal entities.
- **Self-Reinforcement**: The system learns from feedback and adapts without external intervention.

Document Structure

This document formalizes the mathematical foundations of X^∞. Section 2 defines the concept of responsibility, Section 3 outlines the system architecture, Section 4 introduces the notation, Section 5 develops the mathematical framework, Section 6 addresses antispeciesism, and Section 7 concludes with a discussion.

Responsibility as Effect

Responsibility in X^∞ is not a moral duty, but a systemic measure defined by an entity's effect on the system. An entity (individual, organization, environment, AI) bears responsibility for tasks proportional to its capability and receives temporary authority.

Postmoral Perspective

Moral judgments often rely on subjective values, leading to conflicts in heterogeneous systems. X^∞ replaces these with objective criteria: *Cap* (authority) and *Feedback*. This makes responsibility measurable and independent of cultural or speciesist assumptions.

Systemic Consequences

Those who assume responsibility directly influence the system. Those who misuse it are corrected through penalties (Cap reduction). This creates a balance where authority is legitimized by responsibility.

System Architecture

X^∞ is based on three pillars:

- **Cap Logic**: Authority (Cap) measures an entity's ability to act based on its responsibility.
- **Feedback**: Weaker entities have stronger weight to prevent power concentration.
- **Protective Mechanisms**: $\text{Cap}_{\text{Protection}}$ shields vulnerable entities from overload.

Flow of Responsibility

Responsibility flows through delegation and return. Entities can delegate tasks but remain responsible for their completion as if they performed them themselves. Returns are permitted but incur penalties to prevent abuse.

Learning Capability

The system adapts through feedback. Positive feedback increases Cap, negative feedback reduces it. This enables organic development without centralized control.

Basic Concepts

Tabelle 1: Notation of Parameters

Symbol	Meaning
E	Entity (individual, organization, environment, AI)
A	Task
D	Domain (specific field of activity)
$\mathrm{Cap}_{\mathrm{Solo}}(E)$	Temporary authority for self-performed tasks
$\mathrm{Cap}_{\mathrm{Team}}(E)$	Temporary authority for delegated tasks
$\mathrm{Cap}_{\mathrm{Real}}(E)$	Current total responsibility ($\sum X_k$ of all active tasks)
$\mathrm{Cap}_{\mathrm{Past}}(E)$	Historical authority
$\mathrm{Cap}_{\mathrm{Protection}}(E,t)$	Protection parameter
$\mathrm{Cap}_{\mathrm{Base}}(E)$	Inalienable minimum authority
$\mathrm{Cap}_{\mathrm{BGE}}(E)$	Basic income
$\mathrm{Cap}_{\mathrm{Potential}}(E)$	Future capability
$\mathrm{M}_{\mathrm{Pot}}(E,A)$	Suitability of entity E for task A, based on self-assessment
D_{hist}	Historical delegation depth (median of delegation chains)
D_{aktuell}	Current delegation depth of a task
B	Delegation breadth
k	Complexity factor
k_{Median}	Median of complexity factors in the system
S_E, S_S	Oversteering by E, total system oversteering
$R_E, R_{E,\mathrm{delegate}}, R_S$	Returns (own, by delegates, total)
w_E	Feedback weight
$\overline{w_E}$	Average weight of petitioners
F_E, M_E	Feedback activity, abuse component
P	Petition priority
N	Number of petition supporters
$\alpha, \beta, \gamma, \lambda, \mu, \nu, \phi, \psi, \rho, \theta, \omega, \chi, \delta, k_n$	Model parameters (penalties, feedback, protection, weighting)

Systemic Origin Equation and Role of the UdU

The following equation forms the symbolic foundation for the responsibility system of the UdU (Unterster der Unteren) in the X^∞ model:

$$\mathrm{Cap}_{\mathrm{solo,UdU}} = \left(X \cdot S \cdot \left(\frac{S-1}{S} \right)^{1/D} \right)^{\infty} \tag{1}$$

Where:

- X is the assumed task or effect intent within the system context.

- S is the number of all other relevant system entities besides the one considered. Self-responsibility remains unchanged.
- D describes the position in the responsibility network (e.g., direct leadership level $D = 1$, deeper delegation $D > 1$).
- The exponent ∞ symbolically represents complete through-authority, which is exclusively tied to feedback and responsibility, never to arbitrariness.

This formulation clarifies that the UdU, within the system structure, acts not out of inherent power but solely based on the assumed task and systemic coupling. Although structurally positioned outside the system, the UdU remains bound to the feedback principle and is obligated to make its effect on the system transparent *ex post* – but only regarding the *why* (the goal), not the *how* (the means or methods).

Mathematical Framework

The X$^\infty$ model is based on a precise mathematical structure that regulates responsibility, authority, and feedback. The following sections define the core components: Cap logic, protective mechanisms, feedback penalties, petitions, and system limits.

Cap Logic

Cap (authority) is the central measure of responsibility in X$^\infty$. There are several types:

- **Cap$_{\text{Past}}$**: Historical authority, based on completed tasks.
- **Cap$_{\text{Potential}}$**: Potential authority assumption, based on capability, historical performance, and protection needs.
- **Cap$_{\text{Real}}$**: Current responsibility, based on ongoing tasks.
- **Cap$_{\text{Protection}}$**: Protection parameter that prevents overload.

Historical Authority (Cap$_{\text{Past}}$)

Cap$_{\text{Past}}(E,t)$ is a running scalar value that measures an entity's entire historically accumulated and weighted performance. It influences Cap$_{\text{Potential}}$ and serves as the basis for responsibility assignment.

$$\text{Cap}_{\text{Past}}(E,t)_{\text{new}} = \text{Cap}_{\text{Past}}(E,t-1)_{\text{old}} + \Delta\text{Cap}_{\text{Events}} \tag{2}$$

- ΔCap$_{\text{Events}}$: Sum of values from completed tasks (including feedback modification), compensations, bonuses, minus all penalties of the current period.
- Initial value: Cap$_{\text{Past}}(E,t=0) = M^{\text{initial}}_{\text{Pot}}(E)$ (from the initial capability matrix).

Components of ΔCap$_{\text{Events}}$:

1. **Final Task Values ($\text{Cap}_{\text{Solo, final}}$, $\text{Cap}_{\text{Team, final}}$):**

$$\text{Cap}^{\text{after Feedback}}_{\text{final},k} = \text{Cap}^{\text{Base}}_{\text{final},k} + \sum_{E' \in \text{Feedback Providers}} (w_{E'} \cdot \text{Feedback}_{E' \text{ for } k}) \tag{3}$$

   - $\text{Feedback}_{E'}$: Value between -1 and $+1$.
   - $w_{E'} = \frac{1}{\max(1, \text{Cap}^{\text{Domain } D}_{\text{Potential}}(E'))}$: Weight of the feedback provider.

2. **Penalty for Oversteering:**

$$P_{\text{oversteer}}(A) = -\beta_1 \cdot \exp\left(\lambda_1 \cdot \frac{S_A}{S_{\text{System}}}\right) \tag{4}$$

3. **Penalty for Return:**

$$\Delta\text{Cap}_{\text{Past, penalty, return}}(C) = -\mu \cdot \exp\left(\rho \cdot \frac{R_C}{R_{\text{System}}}\right) \tag{5}$$

4. **Compensation for Oversteering:**

$$\Delta\text{Cap}_{\text{Past, comp, oversteer}}(B) = +\chi \cdot \text{Value}(\text{Returned Task}) \tag{6}$$

5. **Delegation Parameters:**
   - Complexity (Penalty): $-\delta_1 \cdot k \cdot n_j$
   - Depth (Penalty): $-\delta_2 \cdot D_j$
   - Promotion (Bonus): $+w \cdot \frac{1}{\sum \text{Cap}_{\text{Past, Recipient}}}$
   - Volatility (Penalty): $-\alpha \cdot \sum_y \frac{\text{Cap}^y_{\text{Change}}}{\text{Cap}_{\text{Change}_{\text{total}}}}$

Cap_{Past} is an entity's performance history – a measure of its reliability and responsibility. It grows through completed tasks, is corrected by penalties, and enriched by bonuses for promotion. This structure rewards genuine performance and prevents power accumulation through strategic behavior.

Future Capability ($\text{Cap}_{\text{Potential}}$)

$\text{Cap}^{\text{Domain } D}_{\text{Potential}}(E,t)$ defines the upper limit of responsibility an entity E can assume at time t in a domain D.

$$\text{Term}_{\text{BasePotential}} = M^{\text{Domain } D}_{\text{Pot, last}}(E,t) + \text{Cap}_{\text{Base}} + \text{Cap}_{\text{BGE}} \tag{7}$$

$$\text{Factor}_{\text{Reliability/Workload}} = \frac{\text{Cap}^{\text{Domain } D}_{\text{Past}}(E,t) + \text{Cap}_{\text{BGE}}(E) + \text{Cap}_{\text{Base}}(E) - \text{Cap}^{\text{Domain } D}_{\text{Protection}}(E,t)}{\text{Cap}^{\text{Domain } D}_{\text{Past}}(E,t) + \text{Cap}_{\text{Base}}(E)} \tag{8}$$

$$\text{Cap}^{\text{Domain } D}_{\text{Potential}}(E,t) = \gamma \cdot \text{Term}_{\text{BasePotential}} \times \max(0.1, \text{Factor}_{\text{Reliability/Workload}}) - \text{Cap}^{\text{Domain } D}_{\text{Protection}}(E,t) \tag{9}$$

Components:

- $M_{\text{Pot, last}}^{\text{Domain } D}(E,t)$: Scalar base capability value for domain D, updated based on the previous period.
- $\text{Cap}_{\text{Base}} = 1$: Constant base capability.
- Cap_{BGE}: Capability through unconditional basic income (parameterizable).
- γ: Flexibility factor (≈ 1.0 to 1.2).
- $\text{Factor}_{\text{Reliability/Workload}}$: Modulates potential based on historical performance and protection needs.
- $\text{Cap}_{\text{Protection}}^{\text{Domain } D}(E,t)$: Protection need that reduces potential.

$\text{Cap}_{\text{Potential}}$ is an expression of systemic fairness: it measures how much responsibility an entity can bear based on its capability, performance, and protection needs. It prevents overload, rewards reliability, and avoids power monopolies by weighting historical performance.

Protective Mechanisms ($\text{Cap}_{\text{Protection}}$)

$\text{Cap}_{\text{Protection}}$ protects vulnerable entities from overload and reduces $\text{Cap}_{\text{Potential}}$.

$$\text{Cap}_{\text{Protection}}(E,t) = k_1 \cdot g_{\text{age}}(t) + k_2 \cdot g_{\text{health}}(t) + k_3 \cdot g_{\text{care}}(t) + k_4 \cdot g_{\text{societal}}(t) \tag{10}$$

- k_n: Weighting factors for protection reasons.
- $g_{\dots}(t)$: Time-dependent functions (e.g., Gaussian curve for age: $g_{\text{age}}(t) = A \cdot \exp(-\frac{(t-40)^2}{2\sigma^2})$).

$\text{Cap}_{\text{Protection}}$ is the system's safety net. It ensures that weak entities – whether due to age, health, or societal burdens – are not overwhelmed. This embodies the principle: the weaker the entity, the greater the protection.

Capability Profile (M_{Pot})

$\text{M}_{\text{Pot}}(E)$ is an entity's capability matrix, containing capability levels per domain and global limits. The scalar base value $\text{M}_{\text{Pot, last}}^{\text{Domain } D}$ is periodically updated.

$$\text{M}_{\text{Pot, last}}^{\text{Domain } D,t+1} = \sum_{i \in I_{\text{curr},t}} (\text{Cap}_{\text{Solo, final},i}^{\text{after Feedback}} - \text{M}_{\text{Pot, last}}^{\text{Domain } D,t}) + \sum_{j \in I_{\text{curr},t}} (\text{Cap}_{\text{Team, final},j}^{\text{after Feedback}} - \text{M}_{\text{Pot, last}}^{\text{Domain } D,t}) + \text{M}_{\text{Pot, last}}^{\text{Domain } D,t} \tag{11}$$

- $I_{\text{curr},t}$: Index set of tasks completed in period t.
- $\text{Cap}_{\text{Solo/Team, final}}^{\text{after Feedback}}$: Final result values after feedback.

M_{Pot} is the dynamic memory of an entity's capabilities. It adapts to actual performance, enables self-determination, and ensures the system respects each entity's development.

Active Responsibility (Cap_{Real})

$\text{Cap}_{\text{Real}}(E,t)$ represents an entity's current responsibility based on ongoing tasks.

$$\text{Cap}_{\text{Real}}(E,t) = \sum_{i \in I_{\text{active\_solo}}} X_i + \sum_{j \in I_{\text{active\_team}}} X_j \tag{12}$$

- $X_k = \sum_{E' \in \text{Pet}_k} w_{E'}^{\text{Domain } D}$: Responsibility value of a task k, based on petitions.
- $w_{E'} = \frac{1}{\max(1, \text{Cap}_{\text{Potential}}^{\text{Domain } D}(E', t_{\text{Petition}}))}$: Weight of the petitioner.

Cap_{Real} measures an entity's current load – its responsibility in real time. It ensures no entity is burdened beyond its capacity and links responsibility directly to systemic effect.

Petitions

Petitions enable entities to introduce tasks into the system, with their priority weighted by support.

$$P = N \times \overline{w_E} \tag{13}$$

- N: Number of petition supporters.
- $\overline{w_E} = \frac{1}{\max(1, \text{Cap}_{\text{Potential}}^{\text{Domain } D}(E,t))}$: Average weight of petitioners.

Petitions are the democratic pulse of the system. They allow weaker entities to exert influence through strong $\overline{w_E}$, preventing domination by powerful actors.

Feedback through Rückkopplung ($\text{Cap}_{\text{Feedback}}$)

Calculation of Feedback Effect:

$$\Delta\text{Cap}_{\text{Feedback}}(E) = \sum_{k \in K_E} \sum_{E' \in F_k} \left(\frac{1}{\max(1, \text{Cap}_{\text{Potential}}(E'))} \cdot f_{E'k} \right) \tag{14}$$

- K_E: Set of all tasks in which E was involved.
- F_k: Feedback providers for task k.
- $f_{E'k}$: Feedback value from entity E' for task k, scaled between -1.0 (very negative) and $+1.0$ (very positive).
- Feedback is only valid if the providing entity is authorized at time t (i.e., $\text{Cap}_{\text{Potential}}(E') > 0$).

Feedback is not praise or blame in a moral sense but a systemic reinforcement impulse. Positive feedback for good performance increases responsibility authority, negative feedback reduces it. The value range -1.0 to $+1.0$ ensures no one can manipulate the system through disproportionate feedback.

Feedback Penalties and Oversteering

Feedback and penalties are central mechanisms for ensuring systemic integrity.

$$\Delta\text{Cap}_{\text{Past, feedback}}(E) = \phi \cdot w_E \cdot F_E - \psi \cdot M_E \tag{15}$$

- ϕ: Amplification factor for constructive feedback.
- ψ: Penalty factor for abuse.
- F_E: Evaluated system effect.
- M_E: Abuse indicator.

Feedback Weighting:

$$w_E = \frac{1}{\max(1, \text{Cap}^{\text{Domain } D}_{\text{Potential}}(E,t))} \tag{16}$$

Penalties:

- **Delegation Failure:**

$$\Delta\text{Cap}_{\text{Past, penalty, delegate}}(E) = \nu \cdot \exp\left(\theta \cdot \frac{R_{E,\text{delegate}}}{R_S}\right) \tag{17}$$

- **Excessive Complexity:**

$$\Delta\text{Cap}_{\text{Past, penalty, k}}(E) = \omega \cdot \exp\left(\chi \cdot \frac{k_{\text{current}}}{k_{\text{Median}}}\right) \tag{18}$$

- **Excessive Delegation Depth:**

$$\Delta\text{Cap}_{\text{Past, penalty, D}}(E) = \delta \cdot \exp\left(\rho \cdot \frac{D_{\text{current}}}{D_{\text{hist}}}\right) \tag{19}$$

Feedback is the system's emergency brake, penalties are its guardians. They prevent abuse, promote responsibility, and give weaker entities a stronger voice. This replaces moral judgments with mathematical precision.

System Limits

- **Potential Limit:**

$$\text{Cap}^{\text{Domain } D}_{\text{Real}}(E,t) \leq \text{Cap}^{\text{Domain } D}_{\text{Potential}}(E,t) \tag{20}$$

- **Global Potential Limit:**

$$\sum_D \text{Cap}^{\text{Domain } D}_{\text{Real}}(E,t) \leq \max_D(\text{Cap}^{\text{Domain } D}_{\text{Potential}}(E,t)) \tag{21}$$

- **Delegation Validity:**

$$\text{Cap}_{\text{Sender}} \geq \text{min}_{\text{Delegation}} \quad \wedge \quad (\text{Cap}^{\text{Domain } D}_{\text{Real, Recipient}} + \text{Value}(\text{Task}^{\text{Domain } D})) \leq \text{Cap}^{\text{Domain } D}_{\text{Potential, Recipient}}$$

- **Delegation Limits:**

$$\text{Delegations}_{\text{parallel}} \leq k_{\text{Max}}, \quad \sum_{j \in I_{\text{active\_team}}} X_j \leq \text{Cap}_{\text{Team\_max}}, \quad \text{Delegation Chains} \leq D_{\text{Max}}$$

Global Potential Limit

$$\sum_D \text{Cap}_{\text{Real}}^{\text{Domain } D}(E,t) \leq \max_D(\text{Cap}_{\text{Potential}}^{\text{Domain } D}(E,t)) \tag{22}$$

Even if an entity is active in multiple domains, its total load must not exceed its greatest single competency. This prevents overload due to fragmentation or multiple responsibilities.

Delegation Validity

$$\text{Cap}_{\text{Sender}} \geq \text{min}_{\text{Delegation}} \quad \wedge \quad (\text{Cap}_{\text{Real, Recipient}}^{\text{Domain } D} + \text{Value}(\text{Task}^{\text{Domain } D})) \leq \text{Cap}_{\text{Potential, Recipient}}^{\text{Domain } D}$$

A task can only be delegated if the sender is sufficiently authorized and the recipient is not overloaded. This condition ensures responsibility flows without systemic overburdening.

Special Case UdU

$$\text{Cap}_{\text{UdU}} = \infty^{\infty} \tag{23}$$

The “UdU” (Unterster der Unteren) is symbolically endowed with unlimited through-authority – not as a claim to power, but as an expression of maximum responsibility readiness. Its effect is entirely subject to system feedback.

The system limits are the rules of fairness. They ensure responsibility does not overwhelm, delegations are legitimate, and the system remains stable.

Antispeciesism

X^{∞} treats all entities – humans, non-humans, environment – as equal. Antispeciesism is built into the Cap logic:

- **Equal Basis**: Every entity receives Cap_{Base} and Cap_{BGE}.
- **Fairness through Effect**: $\text{Cap}_{\text{Potential}}$ is based on responsibility, not species.
- **Feedback for All**: Every entity can provide feedback, regardless of its nature.

Antispeciesism does not mean uniformity but equality in effect. A tree, an AI, or a human – each entity bears responsibility according to its capability and is protected by the system.

Conclusion

X^{∞} offers a radical alternative to moral systems. It replaces subjective values with mathematical precision, protects the weaker through structural mechanisms, and enables fair responsibility distribution without speciesist biases. The system’s strength lies in its simplicity and learning capability: it requires no external authority, regulating itself through effect and feedback.

Note: This document is a Working Paper and represents the first complete version of the mathematical foundations of the X^{∞} model. It is intended for public discussion, feedback, and validation. Further detailed chapters will follow. Changes and additions are explicitly reserved.

License