

# $X^\infty$ – Postmoralisch und Gefühllos

## Mathematische Grundlagen ethischer Steuerung als selbstverstärkendes System

(Working Paper, Version 1.0)

Der Auctor

`x_to_the_power_of_infinity@protonmail.com`

`https://mastodon.social/@The_Auctor`

GitHub: Xtothepowerofinfinity

23. April 2025

### **Abstract**

Das  $X^\infty$ -System definiert Verantwortung neu: nicht als moralische Kategorie, sondern als mathematisch geregelte Wirkung. Es löst klassische ethische Dilemmata durch ein selbstverstärkendes Modell, das Macht durch Rückkopplung und Schutzmechanismen steuert. Dieser Text formalisiert die Grundlagen des Systems, mit Fokus auf die Cap-Logik (Befugnisse), Rückkopplungsstrafen und Antispeziesismus. Ziel ist ein robustes, transparentes System, das Verantwortung ohne moralische Vorannahmen verteilt.

# Contents

<b>1</b>	<b>Einführung</b>	<b>3</b>
1.1	Ziele des Systems . . . . .	3
1.2	Struktur des Dokuments . . . . .	3
<b>2</b>	<b>Verantwortung als Wirkung</b>	<b>3</b>
2.1	Postmoralische Perspektive . . . . .	3
2.2	Systemische Konsequenzen . . . . .	3
<b>3</b>	<b>Systemarchitektur</b>	<b>4</b>
3.1	Fluss der Verantwortung . . . . .	4
3.2	Lernfähigkeit . . . . .	4
<b>4</b>	<b>Grundbegriffe</b>	<b>4</b>
<b>5</b>	<b>Systemische Ursprungsgleichung und Rolle des UdU</b>	<b>4</b>
<b>6</b>	<b>Mathematisches Framework</b>	<b>6</b>
6.1	Cap-Logik . . . . .	6
6.2	Historische Befugnisse ( $\text{Cap}_{\text{past}}$ ) . . . . .	6
6.3	Zukünftige Leistungsfähigkeit ( $\text{Cap}_{\text{potential}}$ ) . . . . .	7
6.4	Rückkopplungsstrafen, Rückgaben und Übersteuerung . . . . .	8
6.5	Schutzmechanismen . . . . .	11
<b>7</b>	<b>Antispeziesismus</b>	<b>11</b>
<b>8</b>	<b>Schlussfolgerung</b>	<b>11</b>
8.1	Lizenz . . . . .	12

# 1 Einführung

Das  $X^\infty$ -System entstammt einer radikalen Prämisse: Moralische Kategorien wie „Gut“ oder „Böse“ sind ungeeignet, um Verantwortung in komplexen Systemen zu regeln. Stattdessen definiert  $X^\infty$  Verantwortung als *Wirkung* – messbar, mathematisch abbildbar und systemisch regulierbar. Dies führt zu einem *postmoralischen* Ansatz, der ethische Steuerung durch strukturelle Mechanismen ersetzt.

## 1.1 Ziele des Systems

- **Fairness durch Wirkung:** Verantwortung wird nicht durch Absicht, sondern durch tatsächliche Konsequenzen verteilt.
- **Schutz der Schwächsten:** Schwache Akteure erhalten durch Rückkopplungsgewichtung eine stärkere Stimme.
- **Antispeziesismus:** Menschen, Nichtmenschen und Umwelt werden gleichwertig als Entitäten betrachtet.
- **Selbstverstärkung:** Das System lernt aus Rückmeldungen und passt sich ohne externe Eingriffe an.

## 1.2 Struktur des Dokuments

Dieses Dokument formalisiert die mathematischen Grundlagen von  $X^\infty$ . Abschnitt 2 definiert den Begriff der Verantwortung, Abschnitt 3 skizziert die Systemarchitektur, Abschnitt 4 führt die Notation ein, Abschnitt 5 entwickelt das mathematische Framework, Abschnitt 6 behandelt Antispeziesismus, und Abschnitt 7 schließt mit einer Diskussion.

# 2 Verantwortung als Wirkung

Verantwortung in  $X^\infty$  ist keine moralische Pflicht, sondern ein systemisches Maß, definiert durch die Wirkung einer Entität auf das System. Eine Entität (Individuum, Organisation, Umwelt, KI) trägt Verantwortung für Aufgaben proportional zu ihrer Fähigkeit und erhält temporäre Befugnisse.

## 2.1 Postmoralische Perspektive

Moralische Urteile basieren oft auf subjektiven Werten, die in heterogenen Systemen zu Konflikten führen.  $X^\infty$  ersetzt diese durch objektive Kriterien: *Cap* (Befugnisse) und *Feedback* (Rückkopplung). Damit wird Verantwortung messbar und unabhängig von kulturellen oder speziesistischen Vorannahmen.

## 2.2 Systemische Konsequenzen

Wer Verantwortung übernimmt, beeinflusst das System direkt. Wer sie missbraucht, wird durch Strafen (Cap-Reduktion) korrigiert. Dies schafft ein Gleichgewicht, in dem Befugnis durch Verantwortung legitimiert wird.

## 3 Systemarchitektur

$X^\infty$  basiert auf drei Säulen:

- **Cap-Logik:** Befugnisse (Cap) messen die Möglichkeiten einer Entität, auf Grund ihrer Verantwortung.
- **Rückkopplung:** Schwächere Entitäten haben ein stärkeres Gewicht, um Machtkonzentration zu verhindern.
- **Schutzmechanismen:**  $\text{Cap}_{\text{protection}}$  schützt vulnerable Entitäten vor Überforderung.

### 3.1 Fluss der Verantwortung

Verantwortung fließt durch Delegation und Rückgabe. Entitäten können Aufgaben delegieren, bleiben aber für deren Erfüllung so verantwortlich, als würden sie die Aufgabe selbst ausführen. Rückgaben sind erlaubt, führen aber zu Strafen, um Missbrauch zu verhindern.

### 3.2 Lernfähigkeit

Das System passt sich durch Rückkopplungen an. Positive Rückmeldungen erhöhen Cap, negative reduzieren es. Dies ermöglicht eine organische Entwicklung ohne zentrale Kontrolle.

## 4 Grundbegriffe

## 5 Systemische Ursprungsgleichung und Rolle des UdU

Die folgende Gleichung bildet die symbolische Grundlage für das Verantwortungssystem im  $X^\infty$ -Modell und beschreibt insbesondere die strukturelle Rolle des *UdU* (Unterster der Unteren) als Träger der Letztverantwortung:

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

Dabei gilt:

- $S$  ist die Anzahl aller relevanten System-Entitäten außer dem UdU selbst. Die Eigenverantwortung der UdU-Entität bleibt stets bestehen und ist nicht delegierbar.
- $D$  beschreibt die Position im Verantwortungsgeflecht. Der UdU agiert strukturell außerhalb des Systems, sodass  $D \rightarrow 1$  für seine Funktion zu betrachten ist.
- Das Hoch- $\infty$  steht symbolisch für die vollständige Durchgriffsbefugnis, die erforderlich ist, um Letztverantwortung tragen zu können. Diese Befugnis ist an Rückkopplung gebunden, nicht an Kontrolle.

Table 1: Notation der Parameter

Symbol	Bedeutung
$E$	Entität (Individuum, Organisation, Umwelt; KI)
$A$	Aufgabe
$\text{Cap}_{\text{solo}}(E)$	Temporäre Befugnisse für selbst durchgeführte Aufgaben
$\text{Cap}_{\text{team}}(E)$	Temporäre Befugnisse durch delegierte Aufgaben
$\text{Cap}_{\text{aktiv}}(E)$	Befugnisse für aktive Aufgaben ( $\text{Cap}_{\text{solo}} + \text{Cap}_{\text{team}}$ )
$\text{Cap}_{\text{past}}(E)$	Historische Befugnisse
$\text{Cap}_{\text{protection}}(E, t)$	Schutzparameter
$\text{Cap}_{\text{base}}(E)$	Unantastbare Minimalbefugnisse
$\text{Cap}_{\text{BGE}}(E)$	Grundeinkommen
$\text{Cap}_{\text{potential}}(E)$	Zukünftige Leistungsfähigkeit
$\text{Cap}_{\text{potential,aktiv}}(E)$	Verfügbare Kapazität für zusätzliche Aufgaben
$M_{\text{pot}}(E, A)$	Eignung von Entität $E$ für Aufgabe $A$ ist eine freiwillige Selbsteinschätzung der Entität, um Selbstverwirklichung und intrinsische Motivation zu berücksichtigen.
$D$	Delegationshistorie ( $D_{\text{Entität}} - 1$ )
$D_{\text{hist}}$	Historische Delegationstiefe (Median der Delegationsketten)
$D_{\text{tiefe}}$	Aktuelle Delegationstiefe
$D_{\text{aktuell}}$	Aktuelle Delegationstiefe einer Aufgabe
$B$	Delegationsbreite
$k$	Wachstumsparameter oder Komplexitätsfaktor
$k_{\text{aktuell}}$	Aktueller Komplexitätsfaktor einer Aufgabe
$k_{\text{Median}}$	Median der Komplexitätsfaktoren im System
$\alpha, \beta, \lambda, \gamma, \mu, \nu, \phi, \psi, \rho, \theta, \omega, \chi, \delta$	Modellparameter (Strafen, Rückkopplung, Kompensation)
$S_E, S_S$	Übersteuerungen durch $E$ , Gesamtübersteuerungen im System
$R_E, R_{E,\text{delegate}}, R_S$	Rückgaben (eigene, durch Delegierte, gesamt)
$w_E$	Rückkopplungsgewicht
$F_E, M_E$	Rückkopplungsaktivität, Missbrauchskomponente

$$X^\infty = \infty, \quad \text{mit } X = \text{Aufgabe} \quad (2)$$

Das Modell legt fest, dass der UdU strukturell außerhalb des Systems positioniert ist, um Handlungsfreiheit und Unabhängigkeit von internen Dynamiken zu gewährleisten. Gleichzeitig ist er jedoch systemisch rückgekoppelt und verpflichtet, die Wirkung seiner Entscheidungen ex post gegenüber dem System transparent zu kommunizieren — jedoch ausschließlich bezogen auf das *Wozu* (Ziel), niemals auf das *Wie* (konkrete Mittelwahl oder Vorgehensweise).

## 6 Mathematisches Framework

### 6.1 Cap-Logik

Cap (Befugnisse) ist das zentrale Maß für Verantwortung in  $X^\infty$ . Es gibt drei Hauptarten:

- **Cap<sub>past</sub>**: Historische Befugnisse, basierend auf abgeschlossenen Aufgaben.
- **Cap<sub>potential</sub>**: Potentielle Befugnisübernahme, basierend auf Eignung und historischer Leistung - erweitert um Wachstumsmöglichkeiten.
- **Cap<sub>protection</sub>**: Schutzparameter, der Überforderung verhindert.

### 6.2 Historische Befugnisse (Cap<sub>past</sub>)

$$\begin{aligned} \text{Cap}_{\text{past}}(E) = & \sum \text{Cap}_{\text{solo,final}}(E) + \sum \text{Cap}_{\text{team,final}}(E) \\ & + \sum \Delta \text{Cap}_{\text{past,return}}(E) \cdot \frac{1}{D_{\text{hist}}} + \sum \Delta \text{Cap}_{\text{past,comp}}(E) \\ & + \sum \Delta \text{Cap}_{\text{past,comp,oversteer}}(E) + \sum \Delta \text{Cap}_{\text{past,bonus}}(E) \\ & + \sum \Delta \text{Cap}_{\text{past,feedback}}(E) - \sum \text{Cap}_{\text{past,penalty}}(E) \quad (3) \end{aligned}$$

*Prosaisch:* Cap<sub>past</sub> stellt die akkumulierte historische Befugnis einer Entität dar – ihre „Leistungsgeschichte“ im System. Sie wächst ausschließlich aus abgeschlossenen Aufgaben, niemals aus aktuellen (aktiven) Aufgaben. Diese Regel verhindert ein Aufblasen der historischen Verantwortungsbilanz durch kurzfristige Projekte oder strategisches Delegieren.

Wichtig: Cap<sub>past</sub> ist das Fundament für das Cap<sub>potential</sub> und beeinflusst sowohl Anreize als auch die Gewichtsverteilung bei Rückkopplungen. Eine Entität, die viel Verantwortung übernommen und erfolgreich getragen hat, kann damit strukturell größere Befugnisse für zukünftige Aufgaben übernehmen – es sei denn, sie hat systemisch bedenkliches Verhalten gezeigt (siehe Strafen und Rückgaben).

#### Einzelne Komponenten von Cap<sub>past</sub>:

- Cap<sub>solo,final</sub>(E): Summe der selbst erfüllten, abgeschlossenen Aufgaben. (*Eigenverantwortlich und operativ erledigt – höchste Bewertungsstufe im System.*)

- Cap<sub>team,final</sub>(E): Summe der durch Delegation abgeschlossenen Aufgaben. (*Verantwortung übernommen, aber Ausführung (teilweise) delegiert.*)

- ΔCap<sub>past,return</sub>(E): Kompensation für Aufgaben, die freiwillig zurückgegeben wurden. Der Faktor  $\frac{1}{D_{\text{hist}}}$  mildert die Kompensation, wenn Rückgaben tief im Delegationsbaum erfolgen, da hier die Verantwortung primär bei den übergeordneten Delegierenden liegt.

- ΔCap<sub>past,comp</sub>(E): Entschädigung für Aufgaben, die einer Entität entzogen wurden (etwa durch Übersteuerung oder Reorganisation).

- ΔCap<sub>past,comp,oversteer</sub>(E): Spezifische Kompensation für den Übersteuerten. Sie deckt die Erfahrungsakkumulation ab, die vor unfreiwilliger Abgabe der Aufgabe entstanden ist.

- ΔCap<sub>past,bonus</sub>(E): Förderbonus für Entitäten, die aktiv Schwächere unterstützen. Diese Komponente belohnt gezielte Förderung und trägt zur Balance im System bei.

- ΔCap<sub>past,feedback</sub>(E) =  $\phi \cdot w_E \cdot F_E - \psi \cdot M_E$ : Die Rückkopplungsanpassung. Positive Rückmeldungen erhöhen Cap<sub>past</sub>, Missbrauch verringert es. Dies wirkt als **Notaus** gegen Manipulation und verantwortungsloses Verhalten.

-  $\text{Cap}_{\text{past,penalty}}(E)$ : Strafen, bestehend aus:

$$\begin{aligned} \text{Cap}_{\text{past,penalty}}(E) = & P_{\text{oversteer}}(E) + \Delta\text{Cap}_{\text{past,penalty,return}}(E) \\ & + \Delta\text{Cap}_{\text{past,penalty,delegate}}(E) + \Delta\text{Cap}_{\text{past,penalty,k}}(E) \\ & + \Delta\text{Cap}_{\text{past,penalty,D}}(E) \end{aligned} \quad (4)$$

*Prosaisch dazu:* Diese Strafen verhindern, dass Entitäten sich einen Vorteil verschaffen können. Sie sanktionieren unverantwortliches Verhalten wie häufige Rückgaben, fehlgeschlagene Delegationen, übermäßig komplexe oder tief delegierte Aufgaben sowie unberechtigte Übersteuerungen. Jede Strafe ist exponentiell skaliert, um wiederholtes Fehlverhalten stärker zu bestrafen und die systemische Balance zu wahren.

### 6.3 Zukünftige Leistungsfähigkeit ( $\text{Cap}_{\text{potential}}$ )

$$\begin{aligned} \text{Cap}_{\text{potential}}(E) = & \sum_{j \in \text{Aufgaben}} M_{\text{pot}}(E, j) \cdot \text{Value}(A_j) \\ & \cdot f(D_{\text{hist}}, D_{\text{gesamt}} - D_E, B) \\ & \cdot \frac{\text{Cap}_{\text{past}}(E) + \text{Cap}_{\text{BGE}}(E) + \text{Cap}_{\text{base}}(E) - \text{Cap}_{\text{protection}}(E)}{\text{Cap}_{\text{past}}(E) + \text{Cap}_{\text{base}}(E)} \end{aligned} \quad (5)$$

*Prosaisch:*  $\text{Cap}_{\text{potential}}$  misst, welches Maß an zukünftiger Verantwortung eine Entität realistischerweise tragen kann. Dabei berücksichtigt das Modell sowohl das bisher Erreichte ( $\text{Cap}_{\text{past}}$ ), das unverlierbare Grundeinkommen ( $\text{Cap}_{\text{BGE}}$ ), als auch das Schutzlevel ( $\text{Cap}_{\text{protection}}$ ), das Schwächere vor Überforderung bewahrt.

Der Schutzwert  $\text{Cap}_{\text{protection}}$  reduziert das  $\text{Cap}_{\text{potential}}$ , um sicherzustellen, dass vulnerable Entitäten nicht in eine Rolle gedrängt werden, der sie strukturell nicht gewachsen sind. Damit ist Missbrauch ausgeschlossen: Eine hohe Schutzbedürftigkeit limitiert direkt die Verantwortungszuteilung.

#### Komponenten der Formel:

-  $M_{\text{pot}}(E, j)$ : Die Potenzialmatrix. Sie beschreibt die Eignung der Entität  $E$  für die Aufgabe  $A_j$ , abgeleitet aus Qualifikation, Freude und Wissen in der jeweiligen Domäne  $\Delta$ .  
 -  $\text{Value}(A_j)$ : Der systemisch bestimmte Wert einer Aufgabe. Aufgaben von höherer Komplexität oder Relevanz tragen stärker zur Berechnung des Potenzials bei.

-  $f(D_{\text{hist}}, D_{\text{gesamt}} - D_E, B)$ : Die Wachstumsfaktorfunktion.

- $D_{\text{hist}}$ : Historische Delegationstiefe, zeigt die Erfahrung im Umgang mit Projektkomplexität.
- $D_{\text{gesamt}} - D_E$ : Aktuelle Delegationstiefe einer Aufgabe gegenüber der Entität – je näher an der Quelle, desto höher das Potenzial.
- $B$ : Delegationsbreite – die Anzahl der parallel gewählten Empfänger. Verhindert strategische Verwässerung der Verantwortung.

- Die Klammer im Nenner:

$$\frac{\text{Cap}_{\text{past}}(E) + \text{Cap}_{\text{BGE}}(E) + \text{Cap}_{\text{base}}(E) - \text{Cap}_{\text{protection}}(E)}{\text{Cap}_{\text{past}}(E) + \text{Cap}_{\text{base}}(E)}$$

skaliert das Potenzial abhängig vom historischen Beitrag.  $\text{Cap}_{\text{base}}$  garantiert, dass der Nenner nie null wird, da unantastbare Minimalbefugnisse immer vorhanden sind.

#### Wesentliche Eigenschaften:

- **Belohnung echter Leistung:** Historische Verantwortung ( $\text{Cap}_{\text{past}}$ ) steigert die Befugnisse für zukünftige Aufgaben.
- **Schutz der Schwachen:**  $\text{Cap}_{\text{protection}}$  verhindert systemische Überforderung vulnerabler Entitäten.
- **Missbrauchsprävention:** Die Kombination von Schutzmechanismus und dynamischer Begrenzung von  $\text{Cap}_{\text{potential}}$  sorgt dafür, dass Verantwortung systemgerecht verteilt bleibt.
- **Automatische Balance:** Wer mehr Verantwortung trägt und erfolgreich abschließt, gewinnt legitime Zuteilungsrechte für künftige Aufgaben – ohne manuelle Eingriffe.

*Prosaisches Fazit:*  $\text{Cap}_{\text{potential}}$  ist nicht nur ein Zahlenwert, sondern Ausdruck systemischer Fairness: Es beschreibt die Wachstumsmöglichkeiten einer Entität auf Basis echter Leistung, unter Berücksichtigung ihrer Schutzbedürftigkeit und systemischen Balance. Das Modell verhindert gezielt die Ausbeutung Schwächerer und erzwingt Verantwortung dort, wo sie auch getragen werden kann.

## 6.4 Rückkopplungsstrafen, Rückgaben und Übersteuerung

### 1. Rückkopplung als systemischer Notaus

$$\Delta \text{Cap}_{\text{past,feedback}}(E) = \phi \cdot w_E \cdot F_E - \psi \cdot M_E \quad (6)$$

*Prosaisch:* Die Rückkopplung dient als sofort reagierendes Frühwarn- und Korrektursystem. Eine Entität, deren Verhalten systemisch auffällig wird (z. B. durch Delegationsmissbrauch oder andauernde Verantwortungslosigkeit), wird unmittelbar sanktioniert – und zwar direkt in  $\text{Cap}_{\text{past}}$ . Diese Funktion ersetzt ethische Appelle durch mathematisch geregelte Wirkungskorrektur.

-  $\phi$ : Verstärkungsfaktor für konstruktives Feedback. -  $\psi$ : Straffaktor für destruktive oder missbräuchliche Handlung. -  $F_E$ : Bewertete Systemwirkung, z. B. durch korrekt ausgeführte Rückmeldungen. -  $M_E$ : Missbrauchsindikator – z. B. überdurchschnittliche Rückgaben ohne Deckung, strategische Übersteuerungen.

**Zweck:** Diese sofortige Wirkung verhindert Akkumulation von Macht durch Intransparenz und stellt sicher, dass Feedbacks in Echtzeit wirken. Rückkopplung ersetzt Kontrolle durch Wirkung – und schützt das System durch direkte Selbstregulation.

### 2. Gewichtung der Rückkopplung: Schwächere Stimmen zählen mehr

$$w_E = \frac{1}{\text{Cap}_{\text{potential}}(E)} \quad (7)$$

*Prosaisch:* Schwächere Stimmen im System – also Entitäten mit wenig  $\text{Cap}_{\text{potential}}$  – erhalten ein stärkeres Gewicht. Ihre Rückmeldungen verändern das System unmittelbar als die der „Starken“. Der mathematische Ausdruck dafür ist das reziproke Rückkopplungsgewicht  $w_E$ , das die Gewichtung direkt an das zukünftige Potenzial koppelt. Da  $\text{Cap}_{\text{potential}}$  durch systemisch garantierte Minimalbefugnisse nie null wird, ist die Gewichtung mathematisch stabil.

Dies folgt dem ethischen Prinzip: **Je schwächer die Stimme, desto genauer muss hingehört werden.**



So werden systemische Blasen verhindert, in denen nur etablierte Verantwortungsträger wirken. Auch Minderheiten oder neue Teilnehmer im System behalten unmittelbare Relevanz – nicht weil sie formal gleichgestellt sind, sondern weil ihre Perspektive im Modell stärker einwirkt.

### 3. Rückgaben – freiwillig, aber nicht kostenlos

$$\Delta \text{Cap}_{\text{past,penalty,return}}(E) = \mu \cdot \exp\left(\rho \cdot \frac{R_E}{R_S}\right) \quad (8)$$

*Prosaisch:* Wer Aufgaben freiwillig zurückgibt, verliert  $\text{Cap}_{\text{aktiv}}(E)$  sofort – und wird ebenfalls nach Abschluss der Aufgabe bestraft. Rückgaben sind erlaubt, aber nicht neutral: Sie kosten Ansehen, Vertrauen und systemisch gesehen auch Wirkung. Der exponentielle Term  $\exp(\rho \cdot \frac{R_E}{R_S})$  sorgt dafür, dass häufige Rückgaben stärker bestraft werden, wobei  $\rho$  die Strafschärfe steuert:

- Rückgaben in einem hochverantwortlichen Umfeld wiegen schwerer. - Wenn viele andere ebenfalls zurückgeben, sinkt der individuelle Strafwert.

**Zweck:** So entsteht ein System, das Rückgabe erlaubt, ohne sie attraktiv zu machen. Es schützt Überforderte, aber fördert gleichzeitig Verantwortungsübernahme.

### 4. Strafe für Delegationsversagen (indirekte Rückgabe)

$$\Delta \text{Cap}_{\text{past,penalty,delegate}}(E) = \nu \cdot \exp\left(\theta \cdot \frac{R_{E,\text{delegate}}}{R_S}\right) \quad (9)$$

*Prosaisch:* Wenn eine Entität Verantwortung weiterdelegiert – und der Empfänger diese zurückgibt –, wird dies dem ursprünglichen Delegierenden angelastet. Nicht als direkte Schuld, sondern als fehlgeschlagene Verantwortungskette. Der exponentielle Term  $\exp(\theta \cdot \frac{R_{E,\text{delegate}}}{R_S})$  bestraft häufige Rückgaben durch Delegierte stärker, wobei  $\theta$  die Strafschärfe steuert.

-  $R_{E,\text{delegate}}$ : Anzahl der Rückgaben durch Zuarbeitende. -  $\nu$ : Basisskala der Strafe – typischerweise niedriger als bei direkter Rückgabe. -  $\theta$ : Eskalationsfaktor für wiederholtes Delegationsversagen.

**Zweck:** Verantwortung endet nicht bei der Übergabe. Wer delegiert, muss mitdenken – und trägt Mitverantwortung für systemisches Scheitern der Kette.

### 5. Übersteuerung – Eingriff in bestehende Verantwortung

$$P_{\text{oversteer}}(E) = \lambda \cdot \exp\left(\gamma \cdot \frac{S_E}{S_S}\right) \quad (10)$$

*Prosaisch:* Übersteuerung ist der Notfallmechanismus im System: Eine Entität greift in den Aufgabenbereich einer anderen ein, weil sie erkennt, dass die Zielerfüllung sonst gefährdet ist. Das ist erlaubt – aber teuer.

- Der Preis für häufige Übersteuerung steigt exponentiell. - Jede Übersteuerung verringert langfristig  $\text{Cap}_{\text{past}}$ .

Der Preis wird auch mit dem ursprünglichen Delegierenden geteilt. So wird nicht nur autoritärer Missbrauch, sondern auch strategisches „Abwälzen“ disincentiviert und Kooperation fördern. Der Übersteuernde erhält die Verantwortung und Befugnisse des Übersteuerten zusätzlich als  $\text{Cap}_{\text{aktiv}}(E)$ , wird aber nach Abschluss der Aufgabe bestraft.

Eine Übersteuerung ist nur möglich, falls der Übersteuernde eine  $\text{Cap}_{\text{aktiv}}(E)$  des Übersteuerten aufnehmen kann, d.h. sein  $\text{Cap}_{\text{potential}}(E)$  für diese zusätzliche Aufgabe noch ausreicht.

## 6. Strafe für übermäßige Komplexität (k-Werte)

$$\Delta\text{Cap}_{\text{past,penalty,k}}(E) = \omega \cdot \exp\left(\chi \cdot \frac{k_{\text{aktuell}}}{k_{\text{Median}}}\right) \quad (11)$$

*Prosaisch:* Entitäten, die übermäßig komplexe Aufgaben übernehmen (hoher  $k_{\text{aktuell}}$ ), riskieren eine Strafe. Dies verhindert, dass Verantwortung durch unnötige Komplexität verwässert wird. Der exponentielle Term  $\exp(\chi \cdot \frac{k_{\text{aktuell}}}{k_{\text{Median}}})$  bestraft Aufgaben, deren Komplexität den systemischen Median ( $k_{\text{Median}}$ ) deutlich übersteigt, wobei  $\chi$  die Strafschärfe steuert.

-  $\omega$ : Basisskala der Strafe. -  $k_{\text{aktuell}}$ : Komplexitätsfaktor der aktuellen Aufgabe. -  $k_{\text{Median}}$ : Median der Komplexitätsfaktoren im System.

**Zweck:** Diese Strafe fördert effiziente und verantwortungsvolle Aufgabenübernahme, indem sie strategische Komplexitätseskalation sanktioniert.

## 7. Strafe für übermäßige Delegationstiefe (D-Werte)

$$\Delta\text{Cap}_{\text{past,penalty,D}}(E) = \delta \cdot \exp\left(\rho \cdot \frac{D_{\text{aktuell}}}{D_{\text{hist}}}\right) \quad (12)$$

*Prosaisch:* Entitäten, die Aufgaben zu tief delegieren (hoher  $D_{\text{aktuell}}$ ), riskieren eine Strafe. Dies verhindert, dass Verantwortung durch übermäßige Delegationsketten verwässert wird. Der exponentielle Term  $\exp(\rho \cdot \frac{D_{\text{aktuell}}}{D_{\text{hist}}})$  bestraft Aufgaben, deren Delegationstiefe die historische Norm ( $D_{\text{hist}}$ ) deutlich übersteigt, wobei  $\rho$  die Strafschärfe steuert und mit der Rückgabenstrafe konsistent ist.

-  $\delta$ : Basisskala der Strafe. -  $D_{\text{aktuell}}$ : Aktuelle Delegationstiefe der Aufgabe. -  $D_{\text{hist}}$ : Historische Delegationstiefe (Median der Delegationsketten).

**Zweck:** Diese Strafe fördert klare und verantwortungsvolle Delegationsstrukturen, indem sie übermäßig tiefe Ketten sanktioniert.

## Zusammenfassendes Bild:

Die Strafen adressieren verschiedene Missbrauchskanäle, um die systemische Integrität zu wahren. Tabelle 2 fasst die Strafen und ihre Zwecke zusammen:

Table 2: Strafen und Missbrauchskanäle

Strafe	Missbrauchskanal	Typ
$\Delta\text{Cap}_{\text{past,penalty,return}}$	Rückgabe der eigenen Aufgaben	Exponentiell
$\Delta\text{Cap}_{\text{past,penalty,delegate}}$	Rückgabe durch die eigenen Empfänger	Exponentiell
$\Delta\text{Cap}_{\text{past,penalty,k}}$	Zu breite Delegation (k-Werte)	Exponentiell
$\Delta\text{Cap}_{\text{past,penalty,D}}$	Zu tiefe Delegation	Exponentiell
$P_{\text{oversteer}}$	Unberechtigte Übersteuerung	Exponentiell

- Rückgabe ist erlaubt, aber mindert Ansehen. - Übersteuerung rettet Projekte, aber verringert Legitimation. - Komplexität und tiefe Delegationen werden bestraft, um Verantwortung klar zu halten. - Rückkopplung schützt das System – insbesondere vor

langanhaltender Manipulation. - Die ethische Wirkung der Rückkopplung basiert auf asymmetrischer Gewichtung:

Schwache wiegen stärker.

Diese Architektur schafft ein lernfähiges, selbstheilendes System, in dem Macht durch Wirkung korrigiert wird und Verantwortung keine moralische Option, sondern strukturelle Folge ist.

## 6.5 Schutzmechanismen

$\text{Cap}_{\text{protection}}$  ist der Kern des Schutzes für schwächere Entitäten. Es wird dynamisch berechnet:

$$\text{Cap}_{\text{protection}}(E, t) = \alpha \cdot \frac{1}{\text{Cap}_{\text{past}}(E) + \text{Cap}_{\text{base}}(E)} + \beta \cdot \text{Vulnerability}(E, t) \quad (13)$$

-  $\alpha$ : Skalierungsfaktor für den Basis-Schutz. -  $\beta$ : Gewichtung der Vulnerabilität. -  $\text{Vulnerability}(E, t)$ : Zeitabhängiges Maß für die Schutzbedürftigkeit (z. B. Ressourcen, Erfahrung).

*Prosaisch:*  $\text{Cap}_{\text{protection}}$  stellt sicher, dass keine Entität überfordert wird. Es wirkt wie ein Sicherheitsnetz: Je schwächer die Entität, desto höher der Schutz. Dies verhindert Ausbeutung und fördert systemische Stabilität.

## 7 Antispeziesismus

$X^\infty$  behandelt alle Entitäten – Menschen, Nichtmenschen, Umwelt – gleichwertig. Der Antispeziesismus ist in die Cap-Logik eingebaut:

- **Gleiche Basis:** Jede Entität erhält  $\text{Cap}_{\text{base}}$  und  $\text{Cap}_{\text{BGE}}$ .
- **Fairness durch Wirkung:**  $\text{Cap}_{\text{potential}}$  basiert auf Verantwortung, nicht auf Spezies.
- **Rückkopplung für alle:** Jede Entität kann Feedback geben, unabhängig von ihrer Natur.

*Prosaisch:* Antispeziesismus bedeutet nicht Gleichmacherei, sondern Gleichwertigkeit in der Wirkung. Ein Baum, eine KI oder ein Mensch – jede Entität trägt Verantwortung nach ihrer Fähigkeit und wird durch das System geschützt.

## 8 Schlussfolgerung

$X^\infty$  bietet eine radikale Alternative zu moralischen Systemen. Es ersetzt subjektive Werte durch mathematische Präzision, schützt Schwächere durch strukturelle Mechanismen und ermöglicht eine faire Verantwortungsverteilung ohne spezieisistische Vorurteile. Die Stärke des Systems liegt in seiner Einfachheit und Lernfähigkeit: Es benötigt keine externe Autorität, sondern reguliert sich durch Wirkung und Rückkopplung.

*Hinweis:* Dieses Dokument ist ein Working Paper und stellt die erste vollständige Fassung der mathematischen Grundlagen des  $X^\infty$ -Modells dar. Es dient der öffentlichen Diskussion, dem Feedback und der Validierung. Weitere Detailkapitel folgen. Änderungen und Ergänzungen sind explizit vorbehalten.

## 8.1 Lizenz

Dieses Werk steht unter der Lizenz CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>).