THE RETURN PROTOCOL

RETURN.

A MARKET MECHANISM FOR NON-SCARCE GOODS

Use is free. Return is priced. The market is the mechanism.

ABSTRACT

Classical markets price scarcity. Open systems produce value without scarcity, which leaves the producers unpriced. No market mechanism has existed to bring these producers income without recreating artificial scarcity.

This protocol specifies one. *A deterministic pricing function that converts commercial use into contribution — scaled progressively to revenue, discounted by in-kind return, modulated by historical behaviour. No discretion. No token. A fixed formula executed by a charter-bound minimal institution that cannot change the rules it administers.*

The claim is that **non-scarce goods can sustain their producers through markets**, *if the market is priced on realised value rather than on denied access.*

THOMAS SCHÜLLER · VIENNA, AUSTRIA
ORCID 0009-0003-9799-6747 · FORSCHUNG@TSCHUELLER.COM

§ 1

The Problem Statement

Markets price what is scarce. Open systems produce what is not scarce. No market mechanism has existed to price the latter — which is why the producers of open systems have never had a systemic income source.

This is not a licensing failure. It is a pricing failure. Classical licenses — GPL, AGPL, Apache, CC-BY-SA — solve access. They do not solve pricing. The production side of open systems has been sustained by volunteer labour, corporate subsidy, grant funding, and sporadic donation. None of these is a market mechanism. All of them depend on the goodwill of participants or the largesse of outside funders.

Meanwhile the consumption side — commercial actors building trillion-dollar industries on open foundations — faces no market signal that would return any portion of value to the producers. The result, well documented over three decades: systematic extraction, chronic underfunding of maintenance, burn-out of contributors, and the repeated pattern of successful open projects either re-licensing under restrictive terms or being captured by single corporate sponsors.

Previous attempts to address this have been:

- **Scarcity reconstruction** (patents, SSPL, BSL) — effective at producing income, but they destroy the open character of the system. The cure is the disease.
- **Voluntary mechanisms** (OpenCollective, GitHub Sponsors, Patreon) — honourable, but statistically negligible. Open Source Surveys consistently report that under 5% of commercial users contribute financially.
- **Governance tokens and DAOs** — introduce speculative markets around the commons, which collapse into plutocracies within 2-4 years (observed across every major deployment).
- **Grant-funded foundations** (Apache, Linux, Mozilla) — work at institutional scale but require sustained major-donor attention. They are not self-sustaining market mechanisms; they are ongoing fundraising operations.

Each of these either destroys openness, fails to produce real income, or depends on outside subsidy. None is a market mechanism operating inside an open system.

> THE QUESTION THIS PROTOCOL ADDRESSES
>
> **Can an open system — one with no artificial scarcity — generate market-driven income for its producers, priced on observable commercial value, distributed by a deterministic formula, without requiring governance, discretion, or outside subsidy?**
>
> *The answer this protocol proposes: yes, if the price is coupled to realised commercial use and the distribution is algorithmic.*

§ 2

Axioms

Five axioms. Each is a constraint. Each eliminates a failure mode observed in prior attempts. Together they define the design space.

AXIOM 1 — ACCESS IS FREE

The Work is available to every person and entity without financial, geographic, institutional, or procedural gate.

This is non-negotiable. The moment access has a price, the system is scarcity-based, not commons-based. Violations of this axiom move the system out of the design space entirely.

AXIOM 2 — COMMERCIAL USE CREATES A PRICED OBLIGATION

Productive application of the Work in a commercial context generates a contribution obligation, calculated by a published formula, visible to all.

Not a moral obligation. Not a suggestion. An obligation with a determinable price, expressible in numbers, checkable by anyone against public evidence.

AXIOM 3 — THE FORMULA IS FIXED

The pricing formula is specified in advance, published publicly, and can be changed only by forking the protocol under a different name.

No committee, no board, no vote adjusts the formula. If people disagree with the formula, they fork. The Executor that operates the protocol is bound by its own charter to apply the formula without modification; attempts to amend it would breach the charter and be unenforceable. Capture becomes pointless — there is no discretion to capture.

AXIOM 4 — CONTRIBUTION SUBSTITUTES FOR PAYMENT

In-kind return (code, documentation, data, research, reviews) reduces the financial obligation by the market value of what was returned.

This is the mechanism by which contributors participate without ever paying money. A small team that returns 100% of its obligation as code pays nothing. A large corporation that returns nothing pays the full amount. Both are compliant.

AXIOM 5 — HISTORY IS PRICED

Past behaviour modulates present obligation. Sustained over-contribution earns a permanent discount; sustained under-contribution earns a permanent premium.

This creates an economic incentive for long-term participation. It cannot be gamed by short-term behaviour because it uses cumulative, time-decaying aggregates. It rewards patience and penalizes extraction.

§ 3

The Core Formula

One equation. Four inputs. Deterministic output. This is the whole protocol.

```
AnnualObligation(entity, year) =
  max(0,
    AttributableRevenue
    × BaseRate(AttributableRevenue)
    × ReturnDiscount(ExpectedContribution, InKindValue)
    × HistoryMultiplier(Karma)
  )
```

Four components. Each specified below. Each computable from public data. Each deterministic.

3.1 AttributableRevenue

The portion of the entity's revenue that derives from productive use of the Work. Defined operationally as the lesser of:

- **Direct attribution:** revenue from a product or service where the Work is a load-bearing component (the product would not exist, or not function meaningfully, without it)
- **Proportional fallback:** if direct attribution is ambiguous, 0.1 × total entity revenue per Work dependency is the default ceiling

The entity declares which applies. The declaration is public. Competitors, analysts, and auditors can challenge implausible declarations — not through a governance body, but by publishing counter-analysis in the Register. The market (investors, customers, journalists) adjudicates plausibility.

3.2 BaseRate — the progressive curve

A smooth, bounded function of revenue. Small revenues pay near nothing. Large revenues approach — but never exceed — the asymptotic maximum.

```
BaseRate(U) =
  MaxRate × (1 - e^(-k × (U - U₀) / U_scale))   if U > U₀
  0   otherwise

Parameters (fixed at v1.0 release):
  U₀ = 100,000 EUR   // floor below which no obligation
  U_scale = 10,000,000 EUR   // characteristic scale
  MaxRate = 0.015   // asymptotic ceiling: 1.5%
  k = 1.0   // curve steepness
```

The table shows the resulting rates. Note the smoothness — no cliff at the threshold, no shock when a firm crosses into obligation territory.

ATTRIBUTABLE REVENUE	BASERATE	OBLIGATION (PRE-DISCOUNT)
50,000 €	0.00%	0 €
100,000 €	0.00%	0 €
250,000 €	0.02%	54 €
500,000 €	0.06%	293 €
1,000,000 €	0.13%	1,285 €
5,000,000 €	0.57%	28,600 €
10,000,000 €	0.95%	94,900 €
50,000,000 €	1.49%	745,000 €
100,000,000 €	1.50%	1,498,000 €
1,000,000,000 €	1.50%	14,999,000 €

Two properties worth noting:

- **No step function.** Crossing from 99,999 € to 100,001 € produces zero additional obligation. The curve is continuous.
- **No punishment for success.** No matter how large the firm grows, the rate asymptotes at 1.5%. There is no level at which RETURN becomes unaffordable relative to commercial reality.

3.3 ReturnDiscount — the contribution pathway

The mechanism by which in-kind return substitutes for financial payment. This is the critical innovation: **labour and capital become substitutable** in satisfying the obligation.

```
ReturnDiscount(E, B) = max(0, 1 - B / E)

Where:
  E = expected obligation (AttributableRevenue × BaseRate)
  B = market value of in-kind contributions for the period
```

CONTRIBUTION RATIO (B/E)	DISCOUNT	CASH PAYABLE
0% (no contributions)	0%	100% of E
50%	50%	50% of E
100% (parity)	100%	0 €
150% (over-contribution)	100%	0 € + Karma credit

In-kind value is not discretionary. The protocol specifies a market-price table, fixed at v1.0 release, that maps contribution types to values. A senior developer-hour is valued at a standard figure; documentation work at another. Tables are denominated in EUR with annual inflation-adjustment via a published index (e.g. Eurostat HICP).

A maintainer of a RETURN-licensed project signs attestations for received contributions. The maintainer chooses only whether to accept; the price follows the table. This limits collusion: a maintainer cannot inflate contribution values at will.

3.4 HistoryMultiplier — the Karma system

The longest-term signal. Rewards entities that contributed more than required across years; penalizes entities that consistently contributed less.

```
Karma(entity, t) = Σ (overperformance - underperformance) ×
decay(years_ago)

HistoryMultiplier(K) = 1 - 0.5 × tanh(K / KarmaScale)

Where:
  overperformance = max(0, B - E)
  underperformance = max(0, obligation_unpaid)
  decay(n) = 0.5^(n/3)  // 3-year half-life
  KarmaScale = 3 × average_annual_obligation

  Result: HistoryMultiplier in [0.5, 1.5]
```

KARMA STATE	MULTIPLIER	PRACTICAL MEANING
Strongly positive	0.50	Long-term over-contributor. Pays half rate.
Mildly positive	0.80	Reliable contributor. 20% permanent discount.
Neutral (new or balanced)	1.00	Standard rate.
Mildly negative	1.20	One past underperformance. 20% surcharge, decays.
Strongly negative	1.50	Repeated non-compliance. 50% surcharge.

The karma state is public, computable from the Register, and cannot be forged. New legal entities start at zero karma. Attempting to game by creating shell entities triggers the threshold problem: they are below U_0 and thus owe nothing, which defeats the purpose of using them.

3.5 Full worked example

A hypothetical software firm with 8M EUR attributable revenue, contributing 20k EUR of developer-hours, with moderately positive karma:

```
Input:
  AttributableRevenue = 8,000,000 EUR
  InKindValue = 20,000 EUR  // signed attestations
  Karma = +0.6 × KarmaScale

Step 1: BaseRate(8M) = 0.015 × (1 - e^-0.79) = 0.82%
Step 2: Expected = 8M × 0.0082 = 65,600 EUR
Step 3: ReturnDiscount = 1 - 20,000/65,600 = 0.695
Step 4: HistoryMultiplier = 1 - 0.5 × tanh(0.6) = 0.732
Step 5: FinalObligation = 65,600 × 0.695 × 0.732 = 33,370 EUR
```

The firm pays 33,370 EUR cash plus 20,000 EUR in-kind. Total return: 53,370 EUR on 8M revenue, or 0.67%. A firm with the same revenue but no in-kind contribution and negative karma would pay 65,600 × 1.0 × 1.5 = 98,400 EUR — three times as much. The spread is the participation incentive.

§ 4

Properties of the Mechanism

Five properties follow mathematically from the formula. Each addresses a failure mode of prior systems.

4.1 Capital and labour are substitutes

An entity can satisfy its full obligation through in-kind contribution alone, through cash alone, or through any combination. A small skilled team pays zero cash because their work satisfies the obligation. A large firm without technical capacity pays cash. Both are equally compliant. This dissolves the usual distinction between "users" and "contributors."

4.2 Long-term participation is rationally preferable

Because karma has a 3-year half-life and responds non-linearly (tanh), the optimal long-term strategy is consistent over-contribution. A firm that contributes 110% of its obligation year after year reaches an equilibrium karma giving a permanent 30-40% discount. A firm that under-contributes 90% reaches the opposite equilibrium, paying 30-40% more indefinitely.

4.3 Non-contributing extractors pay the most

A firm that takes maximum use and returns nothing pays the full BaseRate × 1.5 (negative karma surcharge). For a billion-EUR firm with zero contribution and negative karma: 15M EUR × 1.5 = 22.5M EUR per year per Work. The rational response: convert some cash to in-kind. The formula makes in-kind strictly cheaper than cash because in-kind is valued at market price while improving the infrastructure the firm depends on.

4.4 No discretion exists to be captured

The formula has no parameters that adapt over time. No committee votes on the BaseRate. No jury adjudicates contribution values (the price table is fixed). No board decides who owes what. Attestation is the only human judgement, constrained to accept/reject — not to price. This eliminates the primary capture vector. Change requires a fork, which creates a competing protocol rather than a corrupted one.

4.5 The protocol is self-verifying

Any third party — auditor, journalist, competitor, researcher — can compute any entity's obligation from public data: declared revenue (publicly visible in SEC/EU filings), BaseRate (mechanical), in-kind contributions (public attestations), karma (computable from history). The obligation is a computed quantity, not a contested claim. Disputes are about input honesty, not formula interpretation.

§ 5

Legal Form

The protocol above is a mechanism. A license makes it legally operative. The license is a thin wrapper around the formula.

The legal operation requires:

1. **A copyright license** granting free use conditional on compliance — identical in structure to AGPL or CERN-OHL-S, with a single substantive clause: *"The Licensee's annual obligation, if any, is computed by the Return Protocol v0.2 formula as published at return-license.org/protocol/v0.2."*
2. **Automatic termination** of rights for non-compliance, with standard 90-day cure period
3. **Enforcement standing** for any contributor to the Work, not only a central Steward
4. **Severability** ensuring that if the formula is held unenforceable in a jurisdiction, the license reverts to AGPL-equivalent terms for that jurisdiction

The license text is dramatically shorter than traditional drafts — approximately 700 words — because most of what a license governs is replaced by the formula.

§ 6

The Register

A public, permanent, tamper-evident record of declarations, attestations, and obligations. No governance layer. Only data.

The Register is the sole infrastructure the protocol requires. It serves four functions:

- **Publication of declarations.** Annual filings of revenue, in-kind contributions, and computed obligation. Public.
- **Publication of attestations.** Maintainers sign attestations for received contributions, specifying contribution, price-table value, and contributor. Public.
- **Computation of karma.** From the two feeds, any observer computes any entity's karma. No authoritative source needed.
- **Computation of obligations.** From declared revenue + attestations + karma, any observer computes any obligation.

Implementation can be on-chain (Ethereum Attestation Service or equivalent), off-chain (content-addressable storage with signed manifests), or hybrid. What matters: tamper-evidence, public access, permanent retention, open schema.

Multiple independent mirrors may operate. If the canonical Register is compromised, the community migrates to a mirror.

§ 7

Governance: Minimal Executor, No Discretion

The formula does the thinking. A minimal institution does the plumbing. It has no discretion over rules — only over operational execution of rules it cannot change.

An earlier draft of this document claimed "no governance, no institution." That was incomplete. A formula alone cannot receive a payment, operate a Register, or file a lawsuit. These acts require a legal person. The practical question is not whether an institution exists, but how much it is allowed to decide.

The answer specified here: **as little as possible.** The Executor performs mechanical operations the formula prescribes. It has no rule-setting power, no interpretive discretion, no authority to amend the protocol.

7.1 What the Executor does

- **Receives incoming contributions.** Funds arrive at the Executor's account.
- **Distributes funds by published formula.** Outbound distribution follows a public split table computed from attestations.
- **Operates the canonical Register instance.** One of potentially many mirrors.
- **Pursues non-compliance through courts.** Standing under the license. Quantum is formula-derived (§ 8.2), not argued.
- **Publishes the price table and inflation adjustment.** Mechanical only.

7.2 What the Executor cannot do

- **Cannot change the formula.** Not the rates, not the thresholds, not the karma decay, not the price table structure. Ever.
- **Cannot refuse a formally valid declaration.** Challenging the content is open to any third party through civil action; the Executor does not adjudicate.
- **Cannot set contribution values discretionarily.** Price table is fixed. Maintainer accepts or rejects; Executor applies the published price.

- **Cannot grant waivers or exemptions.** Hardship cases pay less automatically (revenue below threshold) or owe what the formula says.
- **Cannot interpret ambiguous clauses.** Ambiguity is resolved by courts, not by the Executor.

7.3 How the Executor is constrained to execution only

1. **The charter encodes the formula.** The legal document establishing the Executor (statutes of foundation, articles of association) contains the Return Protocol formula as a substantive clause. Changing it requires regulatory approval and can be locked via perpetuity clauses (Ewigkeitsklauseln).
2. **The charter prohibits Core amendment.** A specific article reads: "The formula in Article X shall not be amended. If changes become necessary, a new entity must be established under a different name." Under Austrian and German foundation law, such a prohibition is legally binding on the Executor's own board.
3. **Multi-signature treasury requirement.** Any outbound payment above routine thresholds requires 5-of-9 multi-sig from three independent signer groups (community-elected, board-appointed, external auditor).
4. **Redundant Register mirrors.** If the Executor's Register goes offline or is corrupted, the community migrates to a mirror.

7.4 The fork path

If the formula proves wrong, the response is not governance. It is a fork.

1. Someone proposes a revised formula publicly.
2. A new protocol is published (Return-1.1, Return-Hardware-1.0, any distinct name).
3. A new Executor is established. The old Executor continues with its charter unchanged.
4. New Works choose which protocol to license under. Existing Licensees continue on the protocol they adopted.
5. Bridge tools may allow voluntary migration. Migration is always opt-in.

This is the Bitcoin model. Capture becomes structurally pointless — capturing the Executor gives operational control of existing Licensees, but no power to alter the protocol.

> **DESIGN CLAIM**
>
> *The Executor is not a traditional foundation with a governance mandate. It is an operational body bound by its own charter to execute mechanical rules it cannot change. This reduces its attack surface to near zero: there is nothing to capture except the plumbing.*

§ 7A

Legal Form of the Executor

Five options, each viable. Choice depends on phase, jurisdiction, and appetite for setup complexity.

OPTION 1 — HOSTED BY AN EXISTING STEWARD

SFC, FSFE, Apache Software Foundation, Open Source Initiative — any can act as fiscal host with a charter addendum specifying the Return Protocol formula. Cost: near zero. Time to stand up: weeks. Risk: dependence on hosting organization.

This is the recommended starting form for Phase 0-2.

OPTION 2 — DUTCH STICHTING

Foundation under Netherlands law. Widely used by international open-source projects. Light regulatory oversight, no board-member tax liability, recognized across the EU. Cost: EUR 5-15k setup plus EUR 3-8k annual. Time: 2-4 months.

OPTION 3 — SWISS VEREIN

The most lightweight formal legal person in Europe. Flexible statutes, minimal reporting, internationally recognized. Cost: EUR 5-10k setup plus EUR 2-5k annual. Time: 2-3 months.

OPTION 4 — AUSTRIAN PRIVATSTIFTUNG

Foundation under Austrian law with strong perpetuity clauses and strict regulatory oversight. Minimum capital EUR 70,000. Well-suited to locking in a formula via genuinely hard-to-alter charter clauses. Cost: EUR 15-30k setup plus EUR 10-20k annual, plus minimum capital.

OPTION 5 — AUSTRIAN OR GERMAN VEREIN

Registered association. Seven founding members, flexible statutes, nonprofit tax status available. Cost: below EUR 2k setup plus EUR 1-3k annual. Risk: requires very carefully drafted perpetuity clauses to prevent member-assembly amendment.

RECOMMENDED PATH

Start with **Option 1** (hosted) for the first 2-3 years. Once adoption justifies independence, transition to **Option 2** (Dutch Stichting) or **Option 3** (Swiss Verein), chosen by legal counsel. Option 4 only if revenue justifies ongoing cost.

> **COMMON FEATURE ACROSS ALL OPTIONS**
>
> *Whichever form is chosen, the charter must contain: (a) the Return Protocol formula as a substantive clause; (b) an express prohibition on amending the formula, with fork as the only permitted evolution path; (c) a 5-of-9 multi-sig requirement for treasury operations; (d) mandatory board rotation (no member serves more than four consecutive years); (e) a mandate to operate the Register and pursue enforcement; (f) no other operational scope.*

§ 8

Sanctions

Sanctions flow from the same determinism. No human discretion; no severity judgement; only the mechanical consequences of non-compliance.

8.1 The cascade

Non-compliance — failure to file an annual declaration, or filing an implausible declaration — triggers a fixed sequence:

1. **Day 0:** Missing or implausible declaration detected. Karma reduction begins accruing.
2. **Day 30:** Automatic notification (any party with standing may send). No cost beyond eventual compliance.

3. **Day 120:** Cure period expires. Karma reduction applied permanently. Rights under the license automatically terminate.
4. **Day 120+:** Continued use is copyright infringement. Any contributor has standing for civil remedies.

8.2 Damages are formula-derived

No separately specified liquidated-damages clause. Damages are:

```
Damages = unpaid_obligation × (1 + years_elapsed × 0.1) +
reasonable_costs

// obligation accrues 10% per year of non-compliance
```

A firm that evaded 100k EUR over 3 years owes 130k plus costs. Plainly enforceable. No court needs to reason about punitive damages.

8.3 No private sanctions

Because all sanction effects are formula-derived from public data, no actor can impose private sanctions. The protocol cannot be weaponized.

§ 9

What Remains Unsolved

Honest enumeration of weaknesses no formula eliminates.

9.1 Attributable revenue is contestable

The protocol specifies how to calculate the obligation once AttributableRevenue is known. It cannot prevent underdeclaration. Detection depends on external evidence — competitor counter-analysis, journalistic investigation, whistleblower reports. Same weakness every existing license has.

9.2 Attestation collusion is possible

A maintainer and Licensee could collude on fraudulent attestations. Defense: market-price table bounds per-hour value. Counter-mechanism: maintainers whose attestations are systematically anomalous lose signing authority through community attention — not through committee.

9.3 Initial parameter choices may be wrong

U_0 = 100k, U_scale = 10M, MaxRate = 1.5%, decay = 3 years — all judgement calls at v1.0. Response: not adjustment within v1.0 (prohibited by Axiom 3), but release v1.1 and let users migrate.

9.4 Enforcement still depends on courts

Determinism makes litigation simpler — quantum is not in dispute — but does not eliminate the need.

9.5 Legal validation takes years

Whether the formula is enforceable as a contract in specific jurisdictions requires court decisions over time. Same timeline as any novel license. Early adopters accept this risk knowingly.

§ 10

Why This Might Work

Four reasons, stated with appropriate uncertainty.

First: The formula removes the moral dimension. Prior open-source licensing asked commercial actors to do the right thing. This protocol tells them what they owe. Converting a moral question into a pricing question aligns the protocol with how commercial actors already make decisions — by reading numbers.

Second: The contribution pathway gives the technically-capable a zero-cash option. Most firms with capacity to contribute would rather contribute than pay, because contribution improves the infrastructure they depend on. The formula makes this preference economically rational.

Third: The karma system creates a long-term reputation economy. Firms that commit early and consistently accumulate durable discounts. Firms that try to extract and migrate accumulate durable premiums. Structural incentives for patience.

Fourth: The absence of discretionary governance eliminates the attack surface that has destroyed comparable systems. The Executor exists, but has no authority to change the formula. No board to bribe, no committee with interpretive power, no token to pump. As fixed as Bitcoin's 21 million cap.

The market is the mechanism.
The formula is the market.

This protocol is not a call to change the world. It is a specification. If it works, it will work quietly — as any functioning market mechanism does. If it does not work, it will fail quietly — forks will migrate to other designs. Either outcome is acceptable. What is unacceptable is a system that claims to be open while depending on the goodwill of its extractors. That system is the current one, and this protocol exists because that system is exhausted.

Thomas Schüller · Vienna, Austria · ORCID 0009-0003-9799-6747
forschung@tschueller.com · return-license.org