Payment-Coupled Carbon Accounting for LLM Inference: Governing Emissions Reporting Through Settlement Incentives

Carlo Segat and Thomas Franklin Cory
Technische Universität Berlin / Telekom Innovation Laboratories (T-Labs), Berlin, Germany

carlo.segat@tu-berlin.de
cory@tu-berlin.de

Abstract: Over a deployed model's lifetime, the energy and CO2-equivalent cost of large language model (LLM) inference now rivals the cost of training it. Yet reporting that cost is voluntary: inference providers self-disclose, most publish no per-inference figures at all, and each is commercially incentivised to understate the volume it serves. Existing remedies treat measurement and trust as separate problems. Per-token carbon estimators quantify emissions, while Distributed Ledger Technology (DLT) carbon markets tokenise verified offsets. Neither ever observes a served request, blindly trusting the provider's token counts and recording only the equivalent CO2, not the delivered service. Declaring fewer requests is therefore free. This paper's research question is whether carbon accounting can be made resistant to underreporting. To this end, the paper puts forward a design that cryptographically binds the accounted and paid party. A crucial element of this design is a settlement mechanism that forces paying for an inference to also record its emissions. The design is instantiated over the x402 micropayment standard on an EVM Ledger and evaluated against a baseline without carbon logic, measuring the marginal on-chain cost and the added end-to-end latency the coupling introduces. The coupling roughly doubles settlement gas while adding tolerable latency. The contribution is an economic anti-underreporting guarantee: the ledger acts as a tamper-evident record of served volume, from which a verifiable lower bound on a provider's emissions follows. The inherent limitation is that converting served volume into carbon still relies on declared inputs (e.g. region, model and grid intensity). The client can inspect those inputs before signing, though the prototype leaves such screening to future work. This paper is a first step towards joining sustainable-AI measurement and on-chain settlement, 2 fields that have so far developed in isolation. It reframes emissions reporting as an incentive problem, not a measurement problem. The effect is not only conceptual: users of LLM inference under carbon-disclosure duties can rely on a trustworthy record of their activities.

Keywords: Carbon attestation, LLM inference, Distributed ledger technology, x402 micropayments, Emission accounting, Verifiable sustainability

1. Introduction

The energy and carbon cost of large language model (LLM) inference, accumulated over a model's deployed lifetime, rivals that of training (Fu *et al*, 2024). Yet the carbon figures published by inference providers (IP) come from voluntary self-disclosure. In practice, most publish no per-inference emissions and underreporting is not disincentivised.

Existing research focuses on measurement. Per-request carbon estimators convert token counts into grams of equivalent CO2 (Fu *et al*, 2024; Pathania *et al*, 2026; Pizzini Cavagna *et al*, 2026), though they assume the reported inputs are honest. Another research direction tokenises and records CO2: Distributed Ledger Technology (DLT) based carbon markets issue 1 token per verified tonne of CO2 avoided or removed, then retire it once claimed so the same tonne cannot be counted twice (Zhou *et al*, 2023; Naik *et al*, 2026). Neither links the record to real service activity, so nothing stops a provider from understating how much it served.

This paper proposes a mechanism that binds on-chain carbon accounting to the settlement of an inference service. Paying for the inference on-chain derives the emissions and records them against the provider, while holding back a proportional slice of the payment priced at a carbon-removal rate. Recording less carbon means being paid for less service because a provider cannot draw the funds and leave its emissions unrecorded. Underreporting is therefore deterred economically.

The prototype realises this over the x402 micropayment standard on an EVM chain; Section 3 details the end-to-end settlement flow.

The contributions of this paper are:

- An economic anti-underreporting mechanism that couples payment and on-chain carbon accounting into a single atomic transaction.
- A concrete instantiation of the mechanism over the x402 micropayment standard on an EVM chain: a smart-contract system that converts served token counts into grams of CO2.

- An empirical evaluation of the marginal on-chain cost and added latency that coupling carbon attestation to payment introduces.

1.1 Scale of the Problem

Worldwide, data centres drew 485 TWh in 2025 and emit about 180 Mt CO2 a year (International Energy Agency, 2025, 2026). AI servers took 15% of that electricity in 2024 and their consumption grew 50% in 2025 (International Energy Agency, 2025, 2026). Global AI compute therefore ran on roughly 100 TWh and 40 Mt CO2 in 2025, at least half of it inference (Fu *et al*, 2024). Google alone processed 3.2 quadrillion tokens a month in May 2026, 7 times the year before (Pichai, 2026); ChatGPT served 2.5 billion prompts a day in mid-2025 (International Energy Agency, 2026). How much of these inference emissions is reported is a different question. An analysis that set out to quantify the gap between what hyperscalers report and what their data centres physically emit found the latter over 7 times the former (O'Brien, 2024). This shows how little of data-centre emissions is recorded, and since LLM inference is an ever larger share of those emissions, a means of tracking them is needed. At any removal price this is a liability in the billions of USD a year that no rule currently prices.

In Germany, for example, data centres drew 21.3 TWh in 2025 (Hintemann and Hinterholzer, 2025), 7.3 Mt CO2 at the national grid intensity (Umweltbundesamt, 2026) (reported yearly per facility (Bundesrepublik Deutschland, 2023)); settling inference through the mechanism would instead track each request's emissions at the live grid intensity against its provider, in a volume the provider cannot understate.

2. Related Work

The proposed mechanism joins 3 previously disjoint threads: measuring the carbon of LLM inference, using Distributed Ledger Technology (DLT) to account for and trade carbon, and binding cryptocurrency payments to agent services.

2.1 Measuring the Carbon Footprint of LLM Inference

Estimating the energy of LLM inference falls into 2 approaches. Direct measurement reads energy from the device: faithful but intrusive, and impossible to apply openly to closed-source models. Model-based estimation instead derives power consumption analytically (from proxies such as FLOPs) or empirically, by fitting a predictor to benchmarks. While more tractable, model-based estimation has less fidelity. Because closed-source serving forbids direct measurement, the literature, and this paper's own estimator, rely on the latter approach.

Analytical estimators derive energy from first principles (e.g. FLOPs per token, memory traffic per forward pass). For example, LLMCarbon (Faiz *et al*, 2024) is an end-to-end model spanning both dense and Mixture-of-Experts architectures and both operational and embodied carbon: it derives FLOPs from parameter and token counts, scales them by a modelled hardware efficiency, device peak power, PUE and grid intensity, and matches reported figures to within about 8 per cent, but at whole-workload rather than single-request granularity. In contrast, other models have request granularity, such as SweetSpot (Pizzini Cavagna *et al*, 2026), which derives Joules from the transformer's FLOP and memory-access complexity. Trained estimators fit predictors to measurements. LLMCO2 (Fu *et al*, 2024) trains a neural network over the transformer's kernel-level dataflow, separating the compute-bound prefill and memory-bound decode phases for per-request accuracy. Finally, researchers have shown that a predictor over detailed model and hardware attributes can estimate inference energy for unseen open architectures and hardware (Caravaca, Cuevas and Cuevas, 2025).

2.2 DLT for Carbon Accounting and Offset Markets

This research thread is mature in supply-chain and voluntary carbon market (VCM) research. Zhou *et al* (2023) articulate 4 DLT utilities for carbon markets (recording, verification, trading and dissemination) and present a case study on the Nori NFT-based offset project, whose on-chain issuance and retirement improve on the opacity and double-counting of legacy markets. Most of the field tokenises carbon. Naik *et al* (2026) give the canonical technical form: a Solidity ERC-721 contract in which each verified tonne of avoided or removed CO2 becomes a token that can be traded and retired once. Smart-contract emission-trading schemes automate the market itself, with reputation systems, e.g. from Khaqqi *et al* (2018), mitigating dishonest trading. Another parallel research thread records emissions as immutable, token-free on-chain state (Alotaibi *et al*, 2024), the closer relative of the ledger proposed here.

Research reveals the benefits of DLT for carbon accounting but it also highlights its limitations. As Sipthorpe *et al* (2022) conclude, DLT is no silver bullet: it guarantees an immutable record but not semantic correctness. Tokenising credits can make it worse: fungibility does nothing to verify a token stands for real abatement. A

dramatic example of this is the KlimaDAO market. Built on the Polygon chain, it attracted heavy speculation and then collapsed in price (Ballesteros-Rodríguez, De-Lucio and Sicilia, 2024). More fundamentally, trustworthy measurement is assumed (e.g. external sensors) and defence against under-reporting at the source is not provided.

2.3 Agent Payment Rails

The closest infrastructure to the proposed mechanism comes from agent commerce. Vaziry, Rodriguez Garzon and Küpper (2025) enhance the A2A agent protocol with ledger-anchored AgentCards, giving each agent a tamper-proof on-chain identity, and with x402 HTTP-402 micropayments for DLT-native settlement across organisational boundaries. This is the rail the mechanism presupposes: the identifier of the IP that is paid is the one emissions are recorded against, with settlement coupled to that recording. Li *et al* (2026) go one step further with A402, which makes payment finalise if and only if the paid-for service executed, using atomic service channels and TEE-assisted adaptor signatures. Where A402 binds payment to correct execution, the mechanism described in this paper binds it to emissions. Token payment for AI compute and training is already in production in decentralised-AI marketplaces such as Akash (compute) and Bittensor (training) (Lui *et al*, 2024), yet none binds any carbon accounting to that payment.

2.4 The gap

None of these threads meets the others. The nearest work couples an economic incentive to inference carbon directly (a 2-tier subscription that discounts the client's price when it accepts the lower-quality, higher-latency inference that emits less (Siris *et al*, 2026)), but it stays off-chain and acts on the client side, shaping which service the client picks rather than what the provider is paid, with no on-chain record and no attribution to the provider. A parallel trend brings DLT directly to LLMs (Musamih *et al*, 2026), but it targets data provenance, inference authentication, and audit trails, not emissions. Recording a per-request inference emission on-chain as an unavoidable by-product of payment has no academic predecessor.

3. Design

3.1 Setting and Threat Model

3 parties are considered.

- A client, which may be a human or an autonomous agent, purchases inference.
- An inference provider.
- A verifier, who can bound a provider's emissions from public data.

The recorded emission is derived from a request's metadata, which always includes token counts, what model was used, what hardware ran the model, in what region, the grid carbon intensity at that moment, and the inference duration. The provider is the rational, revenue-maximising adversary who would understate its total emissions. The client is assumed semi-honest: it does not collude with the provider and attests the request characteristics it observes. The threat model assumes the distributed ledger is live and its consensus sound, and that the oracles the system relies on (i.e. for energy and hardware) cannot be directly controlled by the provider.

3.2 Mechanism

Figure 1 shows one carbon-coupled settlement end to end. The provider serves the request and returns an HTTP 402 quote declaring the carbon-related request metadata (i.e. model, hardware, region, token counts, grid intensity, and inference duration). In step 3, the client attests the metadata and returns a single signed EIP-3009 authorisation (akin to signing a cheque). The provider verifies the authorisation and metadata (4) and submits it on-chain to be paid (5).

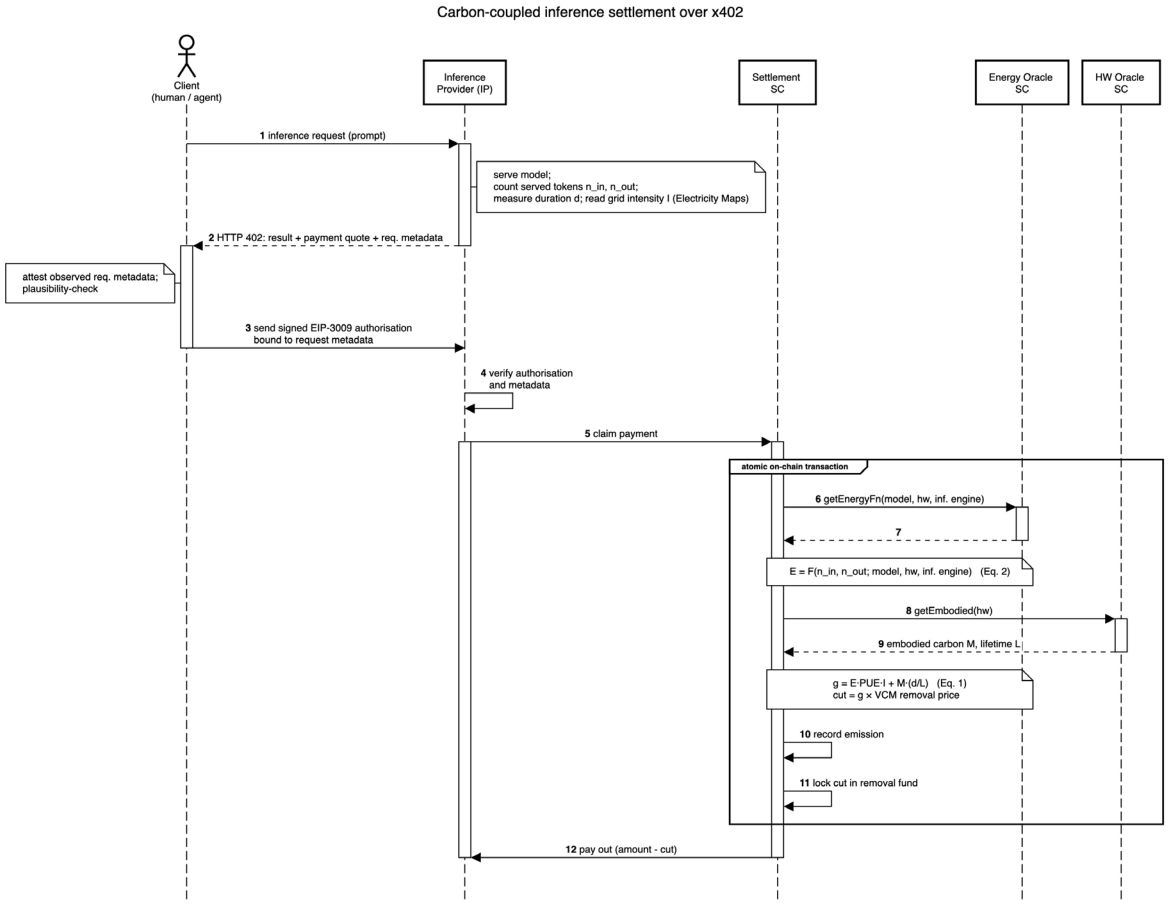

Figure 1: Carbon-coupled inference settlement over x402.

Settlement is one atomic transaction. To compute emissions, the contract reads 2 on-chain oracles: an energy oracle for the token-to-energy function (6–7); and a hardware oracle for the device's embodied carbon and lifetime (8–9).

The CO2-equivalent formula is:

$$g = E \cdot \mathrm{PUE} \cdot I + M \cdot \frac{d}{L} \qquad (1)$$

Where E is the energy in Joules consumed by the request, PUE is power usage effectiveness, I is the declared region's live grid intensity, and M is embodied carbon of the GPU amortised over the request's duration share (d) and GPU lifetime (L).

The energy-per-request formula is:

$$E = F(n_{\mathrm{in}}, n_{\mathrm{out}};\ \mathrm{model},\ \mathrm{hardware},\ \mathrm{inference\ engine}) \qquad (2)$$

No single formula fits all hardware, and hardware-agnostic proxies underestimate energy (Fernandez *et al*, 2025), so each (model, inference engine, GPU) tuple gets its own fitted function committed to the energy oracle. Section 4 fits one such function for the prototype.

The same transaction then records the emissions against the provider in contract state, with no token minted (10), locks a cut priced at the removal rate in a provider-governed removal fund (11), and pays the provider the remainder (12).

Because accounting and payment are one transaction, understating is self-defeating: recording less carbon means being paid for fewer tokens. A state channel, batching many inferences and settling only their net on-chain at closure, was evaluated to keep individual payments off-chain, but set aside for CO2 recording: closure would still need to retain every batched request's carbon inputs. Settling per inference instead keeps the record auditable and preserves the request-time pinning that limits temporal intensity gaming.

3.3 Security Property and its Limits

The mechanism gives a trustless, verifiable lower bound on a provider's served volume (and thus emissions), resting on chain liveness and the provider's desire to be paid, but it fixes volume underreporting only. A provider can still lower its emissions by declaring more efficient hardware or a greener region. Economically the cut is carved from the client's payment rather than charged to the provider, so it internalises carbon as a transaction cost but is repriceable and does not by itself push providers toward greener hardware or regions. Section 7 proposes closing the intensity side with a hyperscaler co-signature on the declared region's grid factor.

The mechanism is un-underreportable but not un-omittable: it resists omission only under the Section 3.1 no-collusion assumption. If the client and provider collude, they can settle the same inference over a separate carbon-blind channel.

Closing omission needs an anchor external to the protocol. The strongest is a demand-pull paradigm. If the buyer is itself under a carbon-disclosure duty, such as the EU CSRD, the US SEC climate rule, or GHG Protocol Scope 3 reporting on purchased AI compute, the on-chain record becomes compliance evidence, though this binds only regulated buyers.

4. Implementation

The mechanism is demonstrated with a prototype. Solidity contracts built on OpenZeppelin realise the design: a settlement contract that verifies the client's authorisation, computes and records the request's emissions, and locks the carbon cut, and 2 oracles supplying the energy-function coefficients and the hardware embodied-carbon values. Payment uses the x402 flow of Section 3.2: the provider replies with an HTTP 402 whose body carries the **`accepts`** quote (an **`exact-carbon`** scheme with the declared carbon metadata), and the client returns its signed EIP-3009 authorisation in the **`X-PAYMENT`** header for the provider to submit on-chain. The settlement asset is a mocked stablecoin standing in for USDT, a 6-decimal ERC-20 whose EIP-3009 **`receiveWithAuthorization`** entry point redeems the client's signed authorisation and whose **`mint`** is left unrestricted so tests can fund any account. The same call that releases payment computes and records the emissions and withholds the cut.

The conversion to CO2 runs on-chain in the same call. The contract reads the energy-function coefficients for the declared model, engine, and hardware, evaluates the served token counts to Joules, applies a fixed power-usage-effectiveness factor (PUE = 1.54, the weighted-average annual data-centre figure reported by the Uptime Institute 2025 survey (Uptime Institute, 2025)) and the declared grid intensity, and adds embodied hardware carbon amortised over the request's share of the device lifetime; it then prices the total at a fixed VCM removal rate to size the locked cut. The PUE and the removal price are constants. The only off-chain inputs are the grid intensity, taken from the Electricity Maps feed (Electricity Maps, 2026) and declared at request time, and the offline fitting of the energy function, which follows the accessible-attribute approach of Section 2.1 (Pathania *et al*, 2026) and so needs no privileged access to the provider's hardware. The hardware oracle's embodied-carbon figure comes from the Boavizta API's GPU model (Boavizta, 2026), parametrised for the measured device from its die area and memory, as no vendor product-carbon-footprint is published for this card.

To further demonstrate how a real IP would obtain the energy-estimation function it commits to the oracle, this paper fits one offline. The committed function is second-order in the token counts, $E = \beta_0 + \beta_{\text{in}} n_{\text{in}} + \beta_{\text{out}} n_{\text{out}} + \beta_{\text{in,in}} n_{\text{in}}^2 + \beta_{\text{in,out}} n_{\text{in}} n_{\text{out}} + \beta_{\text{out,out}} n_{\text{out}}^2$, fitted by ordinary least squares (HC3 heteroscedasticity-robust errors, 5-fold cross-validation) on GPU energy measured for 3 models, phi-4 (14B), Gemma-3 (27B) and DeepSeek-R1 (32B), served through Ollama on the llama.cpp inference engine. Energy is read from the GPU's hardware counter via the Zeus monitor (You, Chung and Chowdhury, 2023) over the exact NVML generation window rather than modelled from FLOPs, so the fit absorbs the engine's behaviour and a different engine is captured by refitting. Input and output token counts are the only variables: lengths are swept (inputs ≈20–410, outputs 16–16,384 tokens) on a single NVIDIA RTX A6000 (48 GB, 300 W) with every other factor pinned (locked clocks, 1 request in flight, greedy decoding, fixed engine parameters, a KV-cache-evicting warm-up, the device isolated and idle); the full configuration and the prototype are in the paper's public repository (https://github.com/CarloSegat/carbon-accounting-llm-dlt). A purely linear fit is inadequate: it predicts negative energy for short requests, which the oracle cannot emit, and errs by over 50 per cent on individual requests. Adding the squared token-count terms removes the negatives and brings error within a few per cent. The output-length term dominates, because each generated token attends over all preceding ones, so energy grows faster than linearly in output length (Pizzini Cavagna *et al*, 2026).

5. Evaluation

The prototype couples a small-model inference provider to the on-chain settlement path. It measures 2 quantities: the marginal on-chain cost of coupling carbon accounting to payment, and the added end-to-end latency. Each is measured against a non-carbon baseline that uses a direct EIP-3009 transfer for settlement rather than the smart contract system. Validating attributed emissions against metered GPU energy, and implementing the removal-fund governance, are left to future work (Section 7).

5.1 Marginal on-chain Cost

The on-chain cost is the gas the carbon coupling adds to settlement. The baseline settles directly on the stablecoin contract with no carbon logic. The carbon-coupled settlement is the contract described in Section 4. To make the reported number robust, every touched storage slot is already non-zero, and each figure is the median over repeated calls. The one-off cost of onboarding a new provider (the first writes to its emissions and balance slots) is excluded, as it amortises to zero over the provider's lifetime.

Table 1: Marginal on-chain gas of the carbon-coupled payment contract against a direct EIP-3009 settlement baseline

Scenario	Gas
Baseline (direct EIP-3009 pay, no carbon)	65,512
Carbon-coupled (PaymentSC.cash())	129,955
Marginal overhead	+64,443 (+98.4%)

The coupling roughly doubles settlement gas, adding 64,443 gas (+98.4%) over the baseline (Table 1).

To attribute this overhead, a `gasleft()`-instrumented twin of the settlement function (`CashProfiler`) records the gas remaining at each internal phase of a single `cash()` call and emits the per-phase deltas. Table 2 gives the per-phase split, showing that pricing the carbon is the most expensive operation.

Table 2: Per-phase attribution of the 64,443 gas marginal overhead (CashProfiler, pre-warmed storage; the shared payment pull is excluded)

Component	Gas	% of marginal
Carbon pricing (2 oracle reads + math)	31,870	49.5%
Emissions ledger (totalCO2 + lockedFund SSTOREs)	10,311	16.0%
Metadata + payment plumbing (calldata, nonce, event, 2nd transfer, dispatch)	22,262	34.5%
Marginal overhead (total)	+64,443	100%

5.2 Added Latency of Carbon Coupling

Latency is measured by comparing the 2 payment flows described in section 5.1 (baseline and carbon-coupled) end-to-end and timing each request from dispatch to on-chain settlement. The mock inference step (i.e. the provider calling the inference engine) is identical in both flows, and both settle on the same local Hardhat node, so loopback RPC round-trips and block production time are common to both. Any fixed cost shared by both flows cancels in the paired difference, so the reported figure is the median additional latency the carbon coupling introduces across paired requests.

Table 3: Median end-to-end latency of the carbon-coupled flow against a bare x402 baseline, over 30 paired requests per flow with warm-up iterations discarded

Scenario	Latency (ms)
Baseline (bare x402 flow, no carbon)	274.3
Full flow (carbon-coupled)	283.2
Marginal added latency	+9.0 (+3.3%)

Table 3 shows the coupling adds 9.0 ms, a 3.3% increase over the 274.3 ms baseline; the on-chain carbon accounting adds only single-digit-millisecond overhead to end-to-end settlement.

6. Limitations

On-chain attestation is only as reliable as the oracles used and the provider's declared request metadata. Five challenges undermine its completeness and accuracy.

Sensor accuracy. On-chip energy counters such as RAPL and NVML systematically deviate from ground truth: against an external power meter, the average error of the dynamic energy they report reaches 73 per cent, the worst case 300 per cent (Fahad *et al*, 2019). Once oracle governance exists (Section 7), other providers and independent parties can challenge a committed energy function against their own metered references.

Idle power. Energy consumption in networking and AI inference alike is dominated by load-independent idle power (Barroso and Hölzle, 2007; Fernandez *et al*, 2025), roughly half the budget even under heavy utilisation. An energy function counting only dynamic power above that baseline lets the provider under-report cheaply.

Batching. The committed energy function fixes batch size, but production serving does not. Mature engines (vLLM, TensorRT-LLM) use continuous batching (Yu *et al*, 2022), folding arriving requests into the forward pass in flight, so weight-streaming cost is amortised over a varying number of requests. The same request then draws less energy under load and more when the server is quiet.

Oracle trust. The conversion inputs enter through 2 oracles, an energy oracle and a hardware oracle, that the attestation trusts without verifying. The prototype's oracle values are owner-set and unchecked on-chain, so an oracle controller can write arbitrary data, relocating trust rather than removing it. The oracle governance planned for spending the locked funds (Section 7) would also be effective here. Multiple parties proposing oracle values under majority or incentive rules, plus the challenge process discussed previously under sensor accuracy, would bound what any single controller can record.

Client screening. The design relies on the client screening the declared metadata and refusing implausible values, but the prototype does not implement it. A plausibility filter (e.g. bounds on duration per token for the declared model and hardware) and a channel to flag or dispute a quote before signing are left to future work.

7. Future Work

Six directions follow from the design and its stated limits:

- Fund governance: a provider-governed mechanism to spend the locked fund on carbon-removal projects.
- Oracle governance: the energy and hardware oracles are fed from a single, unverified source; placing them under the same collective governance would reduce the risk of wrong or adversarial data.
- Corroborating attestation: a hyperscaler co-signature on the per-region grid factors it already publishes (Google Cloud, 2026) would make declared inputs more reliable.
- Stronger atomic binding: an inference taken without paying leaves no record; A402's atomic service channels and TEE-assisted adaptor signatures (Li *et al*, 2026) would bind payment to execution.
- Accuracy validation: metering the serving node at the wall plug to quantify the sensor error and idle-power share the token-based figure misses (Section 6).
- Mechanism footprint: the emissions of the ledger itself, across consensus models and layer-2 batching, which trades per-inference auditability for fewer transactions.

8. Conclusion

Carbon reporting for AI inference is today voluntary and unverifiable, and providers can underreport at will. This paper has proposed making underreporting self-defeating by fusing on-chain payment and carbon accounting into one atomic transaction, so that a provider is paid only when a smart contract simultaneously records the emissions and holds back a removal-priced share. The payment ledger thereby provides tamper-evident activity records from which a verifiable lower bound on emissions can be derived. A prototype shows the coupling roughly doubles settlement gas (+98.4%) while adding under 10 ms (+3.3%) of latency. The mechanism guarantees only that served volume cannot be understated; converting that volume into carbon still depends on the provider's declared region and model, which this paper identifies as the next trust assumptions to tighten. This paper represents a first step towards bridging sustainable-AI measurement and on-chain settlement, 2 fields that have so far developed in isolation.

Acknowledgements

This work was funded by the Federal Ministry of Education and Research (BMBF) in Germany under the grant number 16KIS2251 of the SUSTAINET_guarDian project.

Ethics Declaration: This research did not involve human participants, personal data, or animal subjects. It concerns the design and measurement of a software mechanism evaluated on the authors' own hardware. No ethical clearance was required.

AI Declaration: Generative AI tools were used to assist with brainstorming, summarising related work, and optimizing the language of this paper. All technical claims, the design, the implementation, and the evaluation were produced and verified by the authors, who take full responsibility for the content.

References

Alotaibi, E. et al (2024) "Blockchain-driven carbon accountability in supply chains", Sustainability, 16(24), p. 10872. Available at: https://doi.org/10.3390/su162410872.

Ballesteros-Rodríguez, A., De-Lucio, J. and Sicilia, M.-Á. (2024) "Tokenized carbon credits in voluntary carbon markets: The case of KlimaDAO", *Frontiers in Blockchain*, 7, p. 1474540. Available at: https://doi.org/10.3389/fbloc.2024.1474540.

Barroso, L.A. and Hölzle, U. (2007) "The case for energy-proportional computing", *Computer*, 40(12), pp. 33–37. Available at: https://doi.org/10.1109/MC.2007.443.

Boavizta (2026) "Boavizta API: Environmental impact data for digital equipment". Available at: https://doc.api.boavizta.org (Accessed: 24 June 2026).

Bundesrepublik Deutschland (2023) "Gesetz zur Steigerung der Energieeffizienz in Deutschland (Energieeffizienzgesetz, EnEfG)". Available at: https://www.gesetze-im-internet.de/enefg/ (Accessed: 15 September 2026).

Caravaca, F., Cuevas, Á. and Cuevas, R. (2025) "From prompts to power: Measuring the energy footprint of LLM inference", *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2511.05597.

Electricity Maps (2026) "Electricity Maps API: Live and historical grid carbon intensity". Available at: https://www.electricitymaps.com (Accessed: 24 June 2026).

Fahad, M. *et al* (2019) "A comparative study of methods for measurement of energy of computing", *Energies*, 12(11), p. 2204. Available at: https://doi.org/10.3390/en12112204.

Faiz, A. *et al* (2024) "LLMCarbon: Modeling the end-to-end carbon footprint of large language models", *International Conference on Learning Representations (ICLR 2024)*.

Fernandez, J. *et al* (2025) "Energy considerations of large language model inference and efficiency optimizations", *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2504.17674.

Fu, Z. *et al* (2024) "LLMCO2: Advancing accurate carbon footprint prediction for LLM inferences", *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2410.02950.

Google Cloud (2026) "Carbon free energy for Google Cloud regions". Available at: https://cloud.google.com/sustainability/region-carbon (Accessed: 10 July 2026).

Hintemann, R. and Hinterholzer, S. (2025) *Rechenzentren in Deutschland: Aktuelle Marktentwicklungen (Update 2025)*. Berlin: Bitkom / Borderstep Institut. Available at: https://www.bitkom.org/Bitkom/Publikationen/Studie-Rechenzentren-in-Deutschland (Accessed: 15 September 2026).

International Energy Agency (2025) *Energy and AI*. Paris: IEA. Available at: https://www.iea.org/reports/energy-and-ai (Accessed: 15 September 2026).

International Energy Agency (2026) *Key questions on energy and AI*. Paris: IEA. Available at: https://www.iea.org/reports/key-questions-on-energy-and-ai (Accessed: 15 September 2026).

Khaqqi, K.N. *et al* (2018) "Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application", *Applied Energy*, 209, pp. 8–19. Available at: https://doi.org/10.1016/j.apenergy.2017.10.070.

Li, Y. *et al* (2026) "A402: Binding cryptocurrency payments to service execution for agentic commerce", *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2603.01179.

Lui, E. *et al* (2024) "SoK: Blockchain-based decentralized AI (DeAI)", *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2411.17461.

Musamih, A. *et al* (2026) "Blockchain for large language models (LLMs): Applications, challenges, and framework implementation", *Expert Systems with Applications*, 308, p. 131100. Available at: https://doi.org/10.1016/j.eswa.2026.131100.

Naik, M. *et al* (2026) "Sustainable NFTs: Enabling green energy adoption through IoT-integrated blockchain tokenization of carbon offsets", *International Communications in Heat and Mass Transfer*, 175, p. 110939. Available at: https://doi.org/10.1016/j.icheatmasstransfer.2026.110939.

O'Brien, I. (2024) "Data center emissions probably 662% higher than big tech claims. Can it keep up the ruse?" The Guardian, 15 September. Available at: https://www.theguardian.com/technology/2024/sep/15/data-center-gas-emissions-tech (Accessed: 15 September 2026).

Pathania, P. *et al* (2026) "SEALing the gap: A reference framework for LLM inference carbon estimation via multi-benchmark driven embodiment", *IEEE/ACM International Conference on Software Engineering (ICSE 2026), NIER track*.

Pichai, S. (2026) “I/O 2026: Welcome to the agentic Gemini era”. Google, The Keyword. Available at: https://blog.google/innovation-and-ai/sundar-pichai-io-2026/ (Accessed: 15 September 2026).

Pizzini Cavagna, H. *et al* (2026) “SweetSpot: An analytical model for predicting energy efficiency of LLM inference”, *ACM/SPEC International Conference on Performance Engineering (ICPE 2026)*.

Sipthorpe, A. *et al* (2022) “Blockchain solutions for carbon markets are nearing maturity”, *One Earth*, 5(7), pp. 779–791. Available at: https://doi.org/10.1016/j.oneear.2022.06.004.

Siris, V.A. *et al* (2026) “Greening AI inference with accuracy and latency-aware user incentives”, *IEEE Internet Computing* [Preprint].

Umweltbundesamt (2026) “CO2-Emissionen pro Kilowattstunde Strom 2025 nur leicht gesunken”. Available at: https://www.umweltbundesamt.de/themen/co2-emissionen-pro-kilowattstunde-strom-2025-nur (Accessed: 15 September 2026).

Uptime Institute (2025) *Uptime Institute global data center survey 2025*. Uptime Institute. Available at: https://datacenter.uptimeinstitute.com/rs/711-RIA-145/images/2025.Annual.Survey.Report.pdf (Accessed: 10 July 2026).

Vaziry, A., Rodriguez Garzon, S. and Küpper, A. (2025) “Towards multi-agent economies: Enhancing the A2A protocol with ledger-anchored identities and x402 micropayments for AI agents”, *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2507.19550.

You, J., Chung, J.-W. and Chowdhury, M. (2023) “Zeus: Understanding and optimizing GPU energy consumption of DNN training”, *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2023)*.

Yu, G.-I. *et al* (2022) “Orca: A distributed serving system for Transformer-based generative models”, *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2022)*, pp. 521–538.

Zhou, C. *et al* (2023) “Harnessing Web3 on carbon offset market for sustainability: Framework and a case study”, *arXiv* [Preprint]. Available at: https://arxiv.org/abs/2308.02039.