(ρro)Metheus: Provenance Resolution Oracle for Covenant-Bound Artificial Intelligence

Ling Xiao. [1]
Metheus

Abstract

This technical document introduces the (ρro)Metheus Oracle, a protected computational environment whereby analysts mobilize:

1. audited artificial intelligence (AI) models,
2. validated information pipelines,
3. to compute *causal relations* from data.

Although (ρro)Metheus is a general purpose computational oracle, its initial deployment targets are the regulators, central/commercial banks, insurers, and other entities serving transnational trade in the Global South. Specifically, outputs of (ρro)Metheus will inform high-valued decisions pertaining to the disbursement of funds and claim resolutions. These are business-critical decisions that require both:

- regulator oversight *a priori* oracle deployment;
- and proper assignment of liabilities *a posteriori* oracle judgment.

This implies every step of the computation inside the (ρro)Metheus Oracle must be properly governed by state-level entities, so that the oracle satisfies the strong guarantee of:

- truthful or compliant computation runs to completion;
- while illegal or false computations fail outright.

This requirement is addressed by statistically-typed programming languages, where programs that do not terminate fail at compile time. We enforce this design pattern in (ρro)Metheus: at the heart of the oracle is a *domain-specific programming language* called: Typed ρ-Calculus for covenant-bound probabilistic causal reasoning. It is a two-tiered computational model whereby:

- *in the term universe* or the "analytic" level, (ρro)Metheus uses Pearl's Go-Calculus to evaluate causal relations. In particular, the oracle uses information from validated data sources and summaries generated by audited large language models (LLMs) to compute causal proofs.
- *at the type universe* or the compliance level rests a meta-language extended by state regulators. The type-system governs the quality or legality of computation, so that illegal and false computations terminate immediately. Thereby providing a static guarantee that (ρro)Metheus-bound AI agents are well-behaved in accordance to legal constraints.

When verifying programs for contractual compliance, the (ρro)Metheus compiler uses a legalistic analogue of Floyd–Hoare logic, or Loare-Logic to generate proofs witnessing the Covenant-Compliance of Typed ρ-Calculus code written by analysts. This automated proof system shifts the burden of writing legally-compliant programs from analysts to the (ρro)Metheus compiler, thereby extending the operation domain of AI agents into more sensitive and regulated settings.

In other words, if *code is law*, then one may conceptualize AI agents as entry-level employees within enterprises, so that the (ρro)Metheus sandbox is populated by a set of *law-abiding* junior employees. Meanwhile the (ρro)Metheus type-universe is the *constitution*, or the "system Loare." This Loare is the *lingua-franca* of data and AI validity in the domain of international trade, and shall sit in the public domain as a global public good. We brand this computational model distinguished by 1) validated data pipelines, 2) audited AI models, and 3) deep regulator oversight governing the computation graph of the (ρro)Metheus oracle: Covenant-Bound Artificial Intelligence.

[1] lingxiao@metheusai.xyz

Keywords and phrases Machine learning, statistics, artificial intelligence, formal verifications, deep AI governance, AI safety, programming language design, domain specific programming language, legalistic technology.

Contents

1 Introduction

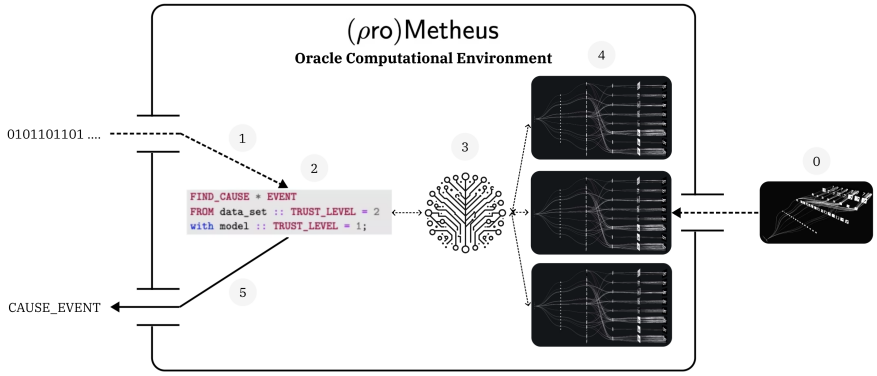
1.1 Technical Stakeholder Perspective

This is a system specification document for the (ρro)Metheus **Provenance Resolution Oracle** for Covenant-Bound Artificial Intelligence. It features:

- A *regulator-compliant* computational environment that enables analysts to run *audited* machine learning (ML) models on *validated* datasets. [2] We refer this set of ML models and datasets as *computational assets.*
- The (ρro)Metheus oracle is especially suitable for recovering casual relationships in datasets in a statistically rigorous manner.
- The theoretical underpinnings of the (ρro)Metheus Oracle is a domain specific programming language named: Typed ρ-Calculus for covenant-bound probabilistic causal reasoning. It combines decades of research and design in programming language theory, classical statistics, and recent advances in machine learning to produce a safer way of interacting with existing AI agents.

1.1.1 Computational Asset Lifecycle in (ρro)Metheus

In concrete terms, (ρro)Metheus is a user interface targeting industry professionals at banks, insurance companies, and regulators.



This user interface is the visible face of the (ρro)Metheus protected computational environment, whereby analysts interact with audited computational assets. The asset lifecycle proceeds as follows:

- (0,1): model and data are audited using stakeholder-defined rules of validation, and subsequently assigned a type, or more precisely a Covenant-Judgement. These legally-tagged computational assets then enter the (ρro)Metheus environment.
- (2) analysts query the data lake with Typed ρ-Calculus in lieu of an LLM-based chat assistant. In particular, they query data with the intention of computing the cause of some real-life **EVENT**.
- (3) (ρro)Metheus compiles the Typed ρ-Calculus language and mobilizes legally-compliant class of data and machine learning models to complete the task.
- (4) (ρro)Metheus draws on open-source and bespoke ML models to compute the likelihood of different causes of the **EVENT**.
- (5) (ρro)Metheus returns probable causes of this **EVENT** to the analyst.

[2] See definition 1 for more on computational environment.

In the figure above, the programming language Typed ρ-Calculus is the top-level user interface. Alternatively we may explore other user interfaces such as:

- A graphic user interface, which may be more appropriate in a consumer setting.
- A natural language interface using LLM's code-generation feature. In this case, we train an LLM to learn a mapping from natural language onto Typed ρ-Calculus.

The advantage of (ρro)Metheus computational environment is three-fold:

- All data and models are audited in accordance to stakeholder-defined laws of validity and security.
- These rule governing validity of information are then propagated along Typed ρ-Calculus's type system from data collection, to model training, fine tuning, and finally inference.
- Additionally, Typed ρ-Calculus has rigorous notions of causality that cannot be expressed in the "classical statistics" underlying machine learning models. Classically trained agents such as auto-regressive models (read: LLMs) can only express correlation, not causation.

The heart of (ρro)Metheus Oracle is the domain specific programming language: Typed ρ-Calculus for covenant-bound probabilistic causal reasoning. Unlike classical statistic language, Typed ρ-Calculus can express causation in addition to correlation.

▶ **Definition 1.** *[7] A **Computational Environment** involves the collection of computer machinery, data storage devices, work stations, software applications, and networks that support the processing and exchange of electronic information demanded by the software solution.*

Next we expand on two use cases of (ρro)Metheus in particular:

- one that targets who query audited data using audited ML models and validated datasets;
- one that targets regulators who define the criteria of valid data and models.

Although they appear to be two orthogonal deployment scenarios, they are in fact part of the same pipeline that takes regulator's legal requirements, and translate them into a compliant artificial intelligence computational environment.

1.1.2 An Example for Analysts

The analysts at banks and insurance companies interact with (ρro)Metheus as a simplified programming language. The language of interface is Typed ρ-Calculus. It is akin to `MySQL`, however instead of rudimentary operations that simply query the database with:

```
SELECT *
FROM employees;
```

The Typed ρ-Calculus under proposal enables analysts to interact with their database in a categorically more sophisticated manner, by seeking the cause of events with:

```
FIND_CAUSE * EVENT
FROM data_set :: TRUST_LEVEL = 2
with model :: TRUST_LEVEL = 1;
```

In this code block, we assume `data_set` and `model` are assets that are pre-loaded in the (ρro)Metheus computational environment. Some additional commentary follows:

1. (line 1): note instead of merely selecting data as is the case with `MySQL`, the analyst uses Typed ρ-Calculus to query the database for the probable cause of **EVENT**.
2. (line 2): the setting **TRUST_LEVEL**=2 in line 2 expresses the trustworthiness of the data used to determine the cost of **EVENT** in line 1. In this case **TRUST_LEVEL**=2 is a *regulator*-defined trust setting on the validity of the data under consideration. It is effectively a filter on the quality of data used to determine the final cause of **EVENT**.
3. (line 3): the statistical models are also typed with **TRUST_LEVEL**=1, which determines the quality of statistical model used to determine the final cause of **EVENT**. This is critical as the quality of statistical models is a function of:
   - the quality of statisticians training them;
   - the rigor of the assumptions underlying the models;
   - and most importantly, the quality of data the model is trained on.

Naturally models trained on **TRUST_LEVEL**=2 data can do no better than attaining a **TRUST_LEVEL**=2 itself, as the adage: "garbage in garbage out" attests. [3] The proper propagation of **type** assignment from datasets to models trained on said datasets is a core feature of Typed ρ-Calculus. Now observe that in the Typed ρ-Calculus query above, the quality of dataset is lower than the quality of the model. Then intuitively the final **CAUSE** of **EVENT** determined by this short program should carry **TRUST_LEVEL**=2.

The interaction of datasets and models at inference time is an example of *program composition*. In programming language (PL) parlance, the *inferred type* of the final recommendation correspond to the confidence of the final recommendation in statistical terms. The intellectual core of Typed ρ-Calculus pivots upon this correspondence between type in the PL sense, and a promise or *covenant* of compliance in the legal sense. We term this category of legally-compliant and statically guaranteed artificial intelligence: Covenant-Bound Artificial Intelligence. This correspondence is explored in table (1).

The intellectual core of Typed ρ-Calculus pivots upon the one-to-one correspondence of *covenant* in the legal sense, with inferred the *type* of data in the programming language sense.

1.1.3 An Example for Regulators

In the previous example, we assumed the criteria that determined which computational asset satisfies any particular **TRUST_LEVEL** was a given. In reality, this criteria is determined by regulators. This

[3] For the sake of simplicity we assume **TRUST_LEVEL**=2 is less trustworthy than **TRUST_LEVEL**=1.

section provides an example of how regulators would interact with (ρro)Metheus to define the vectors of quality along which computational assets are audited against. Since a regulatory environment with broad stakeholder buy-in is seldomly unilaterally defined, finding consensus on the *criteria to audit* must be an interactive process. Now referencing figure (1), (ρro)Metheus presents a user interface whereby:

1. Multiple parties enter a common covenant proposal environment, so as to submit candidate criteria for community judgement.
2. What follows is an iterative process, whereby the stakeholders debate and elect the most sensible set of criteria to judge data validity and AI model quality.
3. Finally this criteria is reified in the (ρro)Metheus computational environment as a type or Covenant-Judgement.

One example of such a Covenant-Judgement is `TRUST_LEVEL` that will be used by the analysts in the example above. Finally observe that depending on the quality of the dataset or AI model, it may only witness some of the criteria defined in Covenant-Judgement. This fine-grained satisfaction is what give rise to different levels of `TRUST_LEVEL`, i.e. `TRUST_LEVEL` = 1 etc.

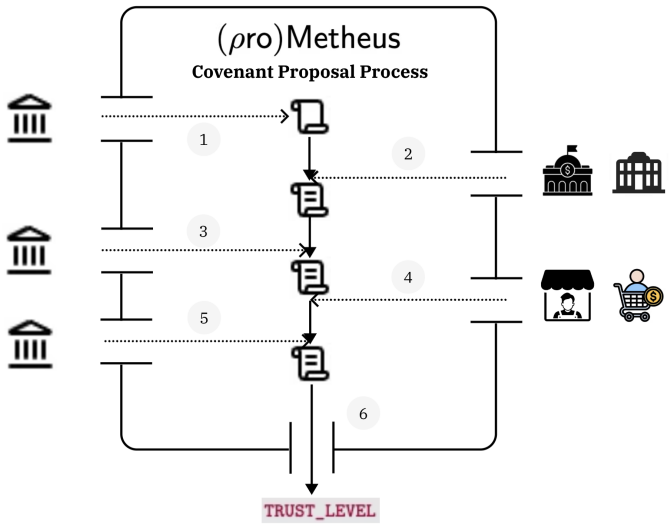


Figure 1 The (ρro)Metheus covenant proposal process is an iterative procedure where the regulator proposes criteria of audit in step (1). In step (2), other stakeholders such as financial institutions amend the proposal. This is sent to the regulator for review in step (3). Then more stakeholders are invited to the roundtable for review in step (4). In step (5), regulators give final approval. And in step (6), this law is translated into code as part of the (ρro)Metheus Typed ρ-Calculus Covenant-Judgement system. This community-defined type or Covenant-Judgement can now be used to audit AI models and datasets for stakeholder compliance.

(ρro)Metheus replace the notion of "code is law" with "`type is law`." Here the subjects under legal audit are computer programs, not people.

The most important aspect of this Covenant-Judgement construction process is that there is a one-to-one correspondence between drafting and enforcing regulations, and designing types and type-checking code. This is (ρro)Metheus' interpretation of the adage "code is law," or alternatively the *Covenant-Type Correspondence*. This correspondence is outlined in table 1.

Law	**Type**
Assign Legal Status	Type Judgment or Covenant-Judgement
Drafting Legislation	Type Design
Core Constitution	Type Primitives
Legislation Amendment	Type Extension
Legislation Enforcement	Type Checking by Compiler
Upholding Covenant	Does Type check
Violating Covenant	Does not Type Check
Relationship before the Covenant	Data/Model Composition before the Compiler
Proving Action is Legal under Terms of Covenant	Proving Typed ρ-Calculus Maintain Logical Invariants

Table 1 The correspondences between legal covenants and type system in Typed ρ-Calculus is the theoretical underpinnings that give a *computational interpretation* to the notion that *compiling code is the same as applying the law*. Under this correspondence, when the (ρro)Metheus compiler type checks the Typed ρ-Calculus code, it is also auditing the AI agents and datasets for legal compliance.

1.2 Broad Stakeholder Perspective

The technical perspective may be sufficient for those with a background in both machine learning and programming language theory, however it is too abstract for the non-technical audience. This section pulls the lens back and recontextualize (ρro)Metheus within the technical and political milieu of the present moment. We answer the question: why is (ρro)Metheus necessary when a diverse set of tools exists to characterize the confidence, validity, and scope of statistical models and datasets?

Why (ρro)Metheus

The demand arises with the latest tranche of breakthroughs in AI around large language models (LLMs). Nominally LLMs are trained via regression on time series data, therefore they are not categorically different from i.e., deep convolution neural networks (CNNs) trained on non-time-series data, or a soft-object manipulations algorithm sampled via reinforcement learning. However the *user-behavior* that have spawned around LLMs applications differ significantly from earlier machine learning tools. Presently users are not only using LLMs as general purpose information retrieval systems, but they are also prompting them as "intelligent reasoning" engines. This user behavior spans across many settings:

- *Financial setting*: banks and insurance companies are using LLMs in high-valued contexts such as producing analyst reports to inform buy/sell decisions. They have never used CNNs in this way. Moreover, the prior generation of financial models used to price assets are either simple, i.e. linear regression on a few factors, or reasonably explainable models based on heat-diffusion equations. In contrast, LLMs are complex and opaque.
- *Medical setting*: doctors are using LLMs to summarize patients' past to inform an opinion, while patients are using LLMs to diagnose their illness. This has far reaching consequences extending past the hospital setting, including the pricing of insurance and other ancillary services.

- *Consumer setting*: people are treating LLMs as friends, teachers, and personal psychologists. This use case may appear the most trivial, however social media has already demonstrated its capacity to silo citizens into misinformation echo chambers and throw elections. Moreover, whereas the harm due to social media is bounded by the number of people who posts, LLM-based social media tools have no such upper bound. This presents a long-term challenge to the health of any society.

In summary, unlike prior deep learning models, LLMs function as "artificial people," or more to the point: "entry level employees." Now given the variety and sensitivity of LLMs' application domains, we assert LLM-based applications require stricter oversight, perhaps even legal oversight, from regulators beyond rudimentary internet guidelines.

1.2.1 The (ρro)Metheus Oracle Desiderata: a High Level Attempt

The (ρro)Metheus computational environment is a protected sandbox with strictly defined and enforced rules. In particular, the *type-system* of the (ρro)Metheus Typed ρ-Calculus is crafted to satisfy the following desiderata:

1. *Flexibility*: regulators from across jurisdictions may define laws governing the behavior of LLMs in a way that suit their local conditions. This is important since language models summarize the internet and with it: culture. There cannot be one set of universal "LLM-laws," as different cultures carry different standards of "good behavior." Therefore the defined boundary conditions must be different.
2. *Composability*: it would be prohibitively expensive, not to mention tedious, to define laws governing every invocation of an AI agent. Thus Typed ρ-Calculus language is defined so that regulators can specify simple rules, and when composed they form a rich system spanning the behavior of an AI agent.
3. *End-to-end oversight*: once the boundary conditions are defined, regulator-privilege extends deep into any AI-based system over the course of its application lifecycle. This means regulators may define high-level laws governing what is good AI behavior. Then the laws would propagate from data collation, to model-training, refinement, and finally inference as well as post-inference data-collection steps. If one conceptualizes an AI agent as a living being, then (ρro)Metheus regulates agent-behavior from cradle to grave.

In conclusion, the (ρro)Metheus Typed ρ-Calculus computational framework operationalizes the adage "code is law." In this case the subjects under legal audit are not people but computer programs. In more technical terms:

> The (ρro)Metheus Typed ρ-Calculus language is a new way of realizing *code as law.* It does so with a novel application of type systems common in statically-typed programming languages. Whereby AI agents and data constitutes the term universe, while the regulator legal framework defines the type universe. This type system, or legal framework, governs:
>
> - how AI learns;
> - once learned, how AI interacts with people, data, and each other.

1.2.2 The (ρro)Metheus Stakeholder Stack

The fact that (ρro)Metheus interacts with regulators at the sovereign level is a critical differentiator and unique go-to-market channel. Presently both financial and medical institutions are inundated with "AI offerings" from major firms and startups alike.

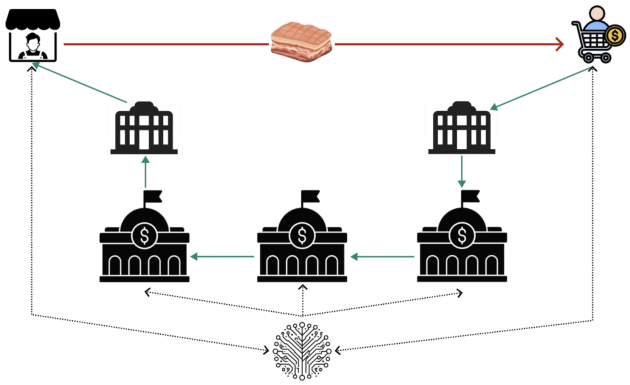


Figure 2 The (ρro)Metheus oracle applied to supply chain orchestration, whereby vendors sell foodstuff to buyers across national borders. The (ρro)Metheus oracle informs trade across stakeholders along this transnational network servicing: buyers, sellers, commercial/central banks, and insurers.

- The startups sell GPT wrappers: that is new way of selling the same old enterprise workflows. This game is stale and soon they shall be cannibalized.
- Major firms such as Microsoft and Google use "AI-agents" as cheese in the trap to sell what they have always sold: API calls and data egress. Ambitious managers use these projects to pad their promo-package. And once promoted, the projects are typically orphaned and whither away.

We escape this red ocean completely and appraise the stakeholder stack as follows:

- *Sovereign regulators*: (ρro)Metheus sell protected computational environments, whereby all code execution is compliant with respect to local laws and norms. The importance of security cannot be overstressed, especially as we enter a multipolar era whereby all things on the chessboard enter a state of motion. Since laws are public goods that rightly belong in the public domain, we engage with regulators through the nonprofit foundation FairCovenant. See the whitepaper at http://bit.ly/44EiQV0 for details on how we engage with regulators [9].
- *Enterprises*: (ρro)Metheus sells compliance as a service, so that the downsides of said institutions using AI are protected. This is critical: if AI is to make high valued decisions, then someone must be liable - (ρro)Metheus answers this question in a principled manner.
- *Enterprises Analysts*: (ρro)Metheus offers a high level interactive environment whereby analysts may query the data for causal relations. This dovetails how people are using LLMs anyways. However the statistical language describing this class of auto regressive models *cannot express causation*. The (ρro)Metheus Typed ρ-Calculus language fills this gap by expressing causality via a principled mathematical language.

The last point bears repetition. Recall Typed ρ-Calculus is not just for regulator-compliance, but also for *probabilistic causal reasoning*. We did not stress this aspect of (ρro)Metheus in the introduction, but it shall be specified in the main body shortly.

1.3 The Greater Metheus Ecosystem

Although (ρro)Metheus is a general purpose computational oracle, it is initially deployed for stakeholders along specific supply chains to support transnational trade organizations (see figure 2). Additionally, (ρro)Metheus will activate the larger open-source AI community by drawing both from

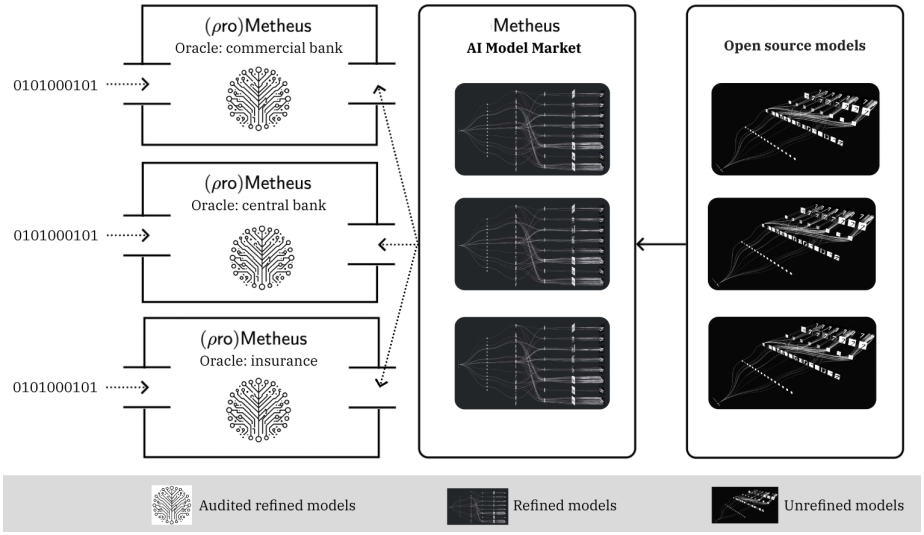


Figure 3 The (ρro)Metheus Oracle connects enterprises with the broader AI foundational model ecosystem by embedding refined AI models within a rigorous legal framework.

freely available models, as well as incentivizing the community to fine-tune or train new models. The process is outlined in figure 3:

- *left column*: there will be separate (ρro)Metheus oracle environment for each industry stakeholder. This is appropriate as each player has different standards of data validity, and AI model security. These separate requirements find congruence in the Typed ρ-Calculus type system.
- *center column*: (ρro)Metheus is connected to an open source AI model market, whereby foundational models are refined by distributed teams of ML engineers. [4]
- *right column*: this moment in technology is unique because of the gluttony of open source foundational models. We funnel this set of commodified assets into the privileged environment of (ρro)Metheus oracle.

Observe that (ρro)Metheus's moat is the regulator and stakeholder-defined laws of data validity and model security. That is say: **their wall is our moat.**

In (ρro)Metheus, the regulator's wall is our moat.

---

[4] See monograph at http://bit.ly/4lFn8Sf for details on AI model market [8].

References

1 *Covenant*. Cornell Law, 2025.

2 *Covenant that runs with the land*. Cornell Law, 2025.

3 A. Zaldivar P. Barnes L. Vasserman B. Hutchinson E. Spitzer I. D. Raji T. Gebru M. Mitchell, S. Wu. Model cards for model reporting. *FAT*, (220-229), 2019. doi:10.1145/3287560.3287596.

4 Oddur Kjartansson Mahima Pushkarna, Andrew Zaldivar. Data cards: Purposeful and transparent dataset documentation for responsible ai. *FAccT*, 2022. doi:10.1145/3531146.3533231.

5 Benjamin C. Pierce. *Types and Programming Languages*. The MIT Press, 2002.

6 Benjamin C. Pierce. *Software Foundations, Volume 2: Programming Language Foundations*. The MIT Press, 2017.

7 Richard F. Schmidt. *Software Engineering Architecture-driven Software Development*. Elsevier Inc, 2013.

8 Ling Xiao. *Metheus: Fair Accountable Transparent Machine Intelligence Audit*. 2025.

9 Ling Xiao and Warwick Powell. *FairCovenant: the Operating System of Trade, Talent, & Technology for the Global South*. 2025.