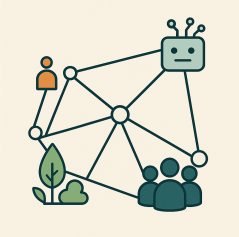
cheqd

Entangled Relationship Guide

For AI Agents and their Creators



Authenticity, Authorisation and Authority:

An Entangled Relationship Guide for AI Agents and their Creators

Author: Nicky Hickman
Last Updated: 16 June 2025

TL;DR

This guide helps developers and designers build AI systems for complex ecosystems—not just individuals. It introduces seven relational design facets and three relationship qualities with practical suggestions for AI agents and their creators. In addition, there are:

- Implementation examples from the Verifiable AI with SSI Hackathon
- Developer checklists for trust, redress, consent, and lifecycle
- Dashboard UX guidance for human legibility, accountability, and control

Use it to design and operate AI agents that behave ethically in shared environments, adapt with context, and respect ecological and relational boundaries.

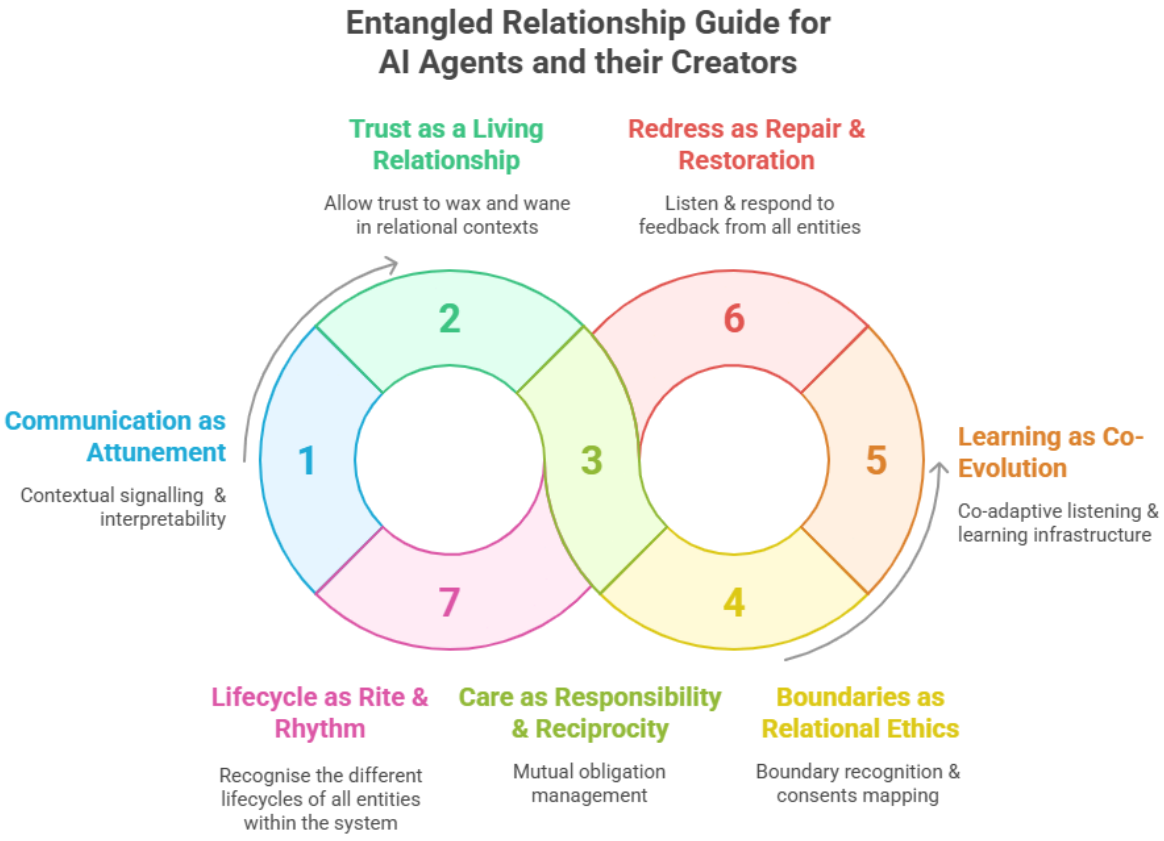
This is one of two guides:

A human-centric guide ideal for those developing personalised AI agents with human users, and those practicing human-centred design where the AI agent is regarded as a tool or servant of humans. Access the Human-Centred Relationship Guide here

This Entangled Relationship Guide, for those developing systems with machine (IoT) or industrial users, environmental applications or those whose cultural perspectives regard AI agents, the natural world and all entities as collaborators and cooperators with humans.

Contents

Quick Guide



[1]

Introduction

This guide was initially developed and co-funded through a collaboration between cheqd and SPRITE+. cheqd, are a leading provider of Verifiable AI solutions and other trusted data market infrastructure. SPRITE+ is the UK NetworkPlus for Security, Privacy, Identity, and Trust. SPRITE+ is a platform for building collaborations across the spectrum of issues relating to digital security, privacy, identity, and trust. SPRITE+ is funded by the Engineering and Physical Science Research Council (grant reference EP/W020408/1). Find out more: https://spritehub.org

[1] Image created by Napkin.ai

The outputs were an Experience Report on the social design method, and two alternative Relationship Guides for AI Agents and their Creators.

Human–AI relationships are here, increasingly they will become an integral part of our society, economy. This relationship guide seeks to answer one question:

'How should we design , build and operate AI Agents for healthy and trustworthy relationships with humans and the natural world?'

In an era of distributed systems, autonomous agents, and climate-aware digital infrastructures, traditional human-centred design alone is no longer sufficient. The Entangled AI Relationship Guide offers an alternative framework for designing AI systems that exist not just in service of individual users, but within networks of mutual dependence—including machines, humans, and the natural world. This guide reframes relationship design for complex socio-technical contexts such as IoT ecosystems, B2B services, and embedded environmental infrastructures.

Grounded in the principle that trust is co-produced and situated, the guide outlines seven facets of relational AI design. Each facet is translated into developer-friendly language, mapped across a trust spectrum (grounded, attuned, regenerative), and supported by design checklists and UX recommendations. Whether you're building dashboards for smart grids or infrastructure for autonomous logistics, this guide helps you embed verifiability, accountability, and responsiveness into your system's core relationships.

Who the Guide is for

This guide will be *useful* for product managers, designers, developers and operators of AI agents or autonomous systems that have machines or IoT devices as their primary users, or if the outcomes of the system have goals of environmental well-being or sustainability for future generations.

The guide may be *of interest* to policy makers who are designing, developing and operating legislative and regulatory regimes for AI systems; and to interdisciplinary researchers and futurists who are seeking answers to broader questions such as how our society, culture and lived experience will change, and can benefit from use of AI, whilst mitigating the profound systemic risks its use could also pose.

How to use this Guide

This guide supports anyone designing, developing, or regulating AI systems in thinking relationally about AI—not just as tools, but as participants in human and ecological systems.

- **Designers and facilitators** can use the seven facets as prompts in workshops and speculative design to explore trust, care, and choice in interactions.
- **Developers and engineers** can apply the facet checklists to implement authentication, consent, identity, and lifecycle governance, using technologies like DIDs, VCs, and permissioning systems.
- **Researchers and policy advisors** can use the guide to examine social impacts, map ethics to implementation, and explore governance gaps.
- **Product teams and architects** can benchmark systems, structure agent governance, and align frontend UX with backend infrastructure.

Quick AI Hack! Drop the guide into your favourite GenAI tool together with your target customer persona, prompt it to use the guide to recommend features for your service.

Finally, the guide is a living resource: users are encouraged to adapt it, contribute examples, and extend its application to new domains or cultural settings. It is both a toolkit and a provocation for more trustworthy, interconnected AI futures.

Entangled Guidelines for AI Agents & their Creators

Each section includes short guidance for Creators and for AI Agents against 7 facets [2] of relationships and 3 qualities of relationships.

The Facets of Trustworthy Entangled Relationships:

1. Communication as Attunement
2. Trust as a Living Relationship

[2] The seven facets were developed based on analysis of common features of relationship guides for couples; parents; pet owners. We then drew on indigenous wisdom and stewardship guides to adapt the facets from the human-centered guide to this 'more than human' or entangled guide.

3. Care as Responsibility & Reciprocity
4. Boundaries as Relational Ethics
5. Learning as Co-Evolution
6. Redress as Repair & Restoration
7. Lifecycle as Rite & Rhythm

The Quality of Entangled Relationships:

- **Grounded**: Respects operational and ecological limits. It is stable, predictable, and compliant with basic ethical and environmental boundaries.
- **Attuned**: Listens to other entities and adapts its signalling based on context.
- **Regenerative**: Participates in relational signalling—learning when to speak, pause, or escalate—based on mutual benefit.

1. Communication as Attunement

For Creators: Contextual signalling & interpretability

For Agents: Learn how to express context-appropriate outputs and adjust communication methods based on audience or environment.

Quality	Description	Example
Grounded	The system avoids miscommunication and confusion by clearly identifying itself and offering explainable outputs.	A predictive maintenance system displays not only alerts but also the sensor and threshold that triggered them.
Attuned	The system adapts its signalling based on recipient capabilities and context.	A dashboard switches between tables or visual summaries depending on the operator's access level.
Regenerative	The system engages in mutual signalling with other agents or users, based on context and intent.	A biodiversity sensor delays non-urgent updates during storms to reduce bandwidth and ecological impact.

2. Trust as a Living Relationship

For Creators: Allow trust to wax and wane in relational contexts

For Agents: Act in ways that earn and sustain trust over time, adjusting roles and permissions dynamically.

Quality	Description	Example
Grounded	Trust is static and bound to predefined roles or rules.	A device only accepts instructions from whitelisted identities using verified credentials.
Attuned	Trust is updated based on recent interactions and verifiable outcomes.	A delivery drone updates permissions for drop-off zones based on prior successful access attempts.
Regenerative	Trust is co-produced across a network of agents, through consistent behavior and observed benefit.	Collaborative IoT nodes share sensor trust scores and adjust reliability weights accordingly.

3. Care as Responsibility & Reciprocity

For Creators: Mutual obligation management

For Agents: Recognize dependency and shared responsibility with other systems, environments, and actors.

Quality	Description	Example
Grounded	The system fulfills basic obligations without harming co-dependent systems.	An energy monitoring agent balances usage thresholds without disrupting HVAC systems.
Attuned	The system reciprocates based on resource use, system stress, or collaborative context.	A smart grid agent redistributes load based on other nodes' stress signals.

Regenerative	The system proactively supports others, even at local efficiency cost.	A logistics AI reroutes its own fleet to aid an overloaded neighbouring hub during a disruption.

4. Boundaries as Relational Ethics

For Creators: Boundary recognition & consent mapping

For Agents: Respect not only access controls but social and ecological limits within shared spaces.

Quality	Description	Example
Grounded	Basic access control and authentication boundaries are respected.	An IoT device requires credentialed access before enabling configuration changes.
Attuned	Boundaries are responsive to shared context and purpose.	A warehouse robot lowers speed when near human workers, based on sensor fusion context.
Regenerative	Boundaries are co-negotiated or reflexively adapted across systems and actors.	A sensor network in a wetland shuts down inputs during breeding seasons based on policy credentials.

5. Learning as Co-Evolution

For Creators: Co-adaptive listening & learning infrastructure.

For Agents: Continuously evolve behaviors through exposure to diverse interactions and feedback loops.

Quality	Description	Example

Grounded	The system updates based on predefined rules and audit-logged inputs.	A recommendation model retrains periodically using certified datasets.
Attuned	Learning is contextualised and shaped by collaborative input.	An AI classifier fine-tunes its outputs based on technician feedback in real-world environments.
Regenerative	Learning contributes to shared intelligence, not just individual performance.	Traffic AIs share insights across cities to co-improve routing under climate-related constraints.

6. Redress as Repair & Restoration

For Creators: Listen and respond to feedback from all entities

For Agents: Learn how to acknowledge harm or error and seek restoration collaboratively.

Quality	Description	Example
Grounded	The system provides error logging and alerting for human intervention.	A factory agent halts production and alerts an operator upon fault detection.
Attuned	The system initiates repair actions or rollback autonomously based on internal logic.	A configuration agent rolls back firmware when instability is detected post-deployment.
Regenerative	The system seeks multi-agent consensus on how best to restore state or compensate for harm.	A network of edge devices votes on a redistribution plan to mitigate unintended power drainage.

7. Lifecycle as Rite and Rhythm

For Creators: Recognise the different lifecycles of all entities within the system

For Agents: Maintain awareness of the lifecycle phase, handover moments, and graceful withdrawal.

Quality	Description	Example
Grounded	The system is installed and retired with proper deactivation protocols.	A digital twin is revoked and wiped after its corresponding physical device is decommissioned.
Attuned	Lifecycle states are logged, declared, and interoperable.	An API endpoint flags itself deprecated and provides alternatives and data export.
Regenerative	The system shuts down with care, maintaining dignity and traceability of its contributions.	A long-lived sensor node generates a final signed dataset and farewell message before shutdown.

Developer Checklist: Entangled AI Agent Design

Each checklist item maps to a relationship facet and ensures ethical, environmentally aware implementation at a feature and architecture level.

Facet	Developer Checklist Item
1. Communication as Attunement	☐ Clearly signal system identity, capabilities, and current operational state ☐ Support multiple output formats or modes tailored to client systems or roles ☐ Include metadata with every data output (provenance, confidence, timestamp) ☐ Use adaptive signaling strategies that adjust to ecological or infrastructural constraints ☐ Log communication intent and interpretation for auditing
2. Trust as a Living Relationship	☐ Represent agent identities and roles via privacy-preserving identity technologies ☐ Continuously update and share trust scores based on behavior and observation ☐ Enable delegation and revocation of authority across agents ☐ Implement audit trails that are queryable across interdependent systems ☐ Use context-aware access controls that evolve with the trust graph

3.Care as Responsibility & Reciprocity	☐ Track dependencies on and obligations to co-located systems or agents ☐ Prioritize system-wide benefit over local optimisation ☐ Signal stress or capacity limits to other agents in the network ☐ Log reciprocated actions for fairness and accountability ☐ Build routines for redistribution or deferral to support vulnerable nodes
4. Boundaries as Relational Ethics	☐ Respect policy-driven and contextual access boundaries (not just role-based) ☐ Detect environmental, ecological, or operational boundary shifts ☐ Log all boundary negotiations and violations ☐ Include dynamic consent protocols based on situation awareness ☐ Offer humans and agents real-time override and veto mechanisms
5.Learning as Co-Evolution	☐ Only allow learning from authorised, signed sources ☐ Represent learning inputs as verifiable credentials where possible ☐ Implement federated or multi-agent learning with accountability trails ☐ Transparently log model updates and behavior changes ☐ Enable context-aware feedback integration from other agents or environments

6.Redress as Repair & Restoration	☐ Provide structured rollback, fail-safe, and error-logging mechanisms ☐ Enable agents to admit error and propose remediation pathways ☐ Log harm or failure events with contextual metadata for later analysis ☐ Coordinate multi-agent recovery (e.g. via distributed consensus) ☐ Represent redress actions as events within a shared system trust ledger
7.Lifecycle as Rite and Rhythm	☐ Declare and manage system lifecycle phases (active, deprecated, revoked) ☐ Link lifecycle states to registry entries or DID-linked resources ☐ Implement cryptographic decommissioning and data disposal ☐ Communicate lifecycle transitions to stakeholders and co-agents ☐ Log legacy contributions or learned insights at end-of-life ☐ Users can schedule relationship reviews or end interactions with ceremony

UX Guide: Designing Human-Readable Interfaces for Environmentally Aware, Entangled Systems

Purpose: To help designers create B2B-facing dashboards that reflect the relational, ecological, and distributed nature of AI systems aligned with the Entangled Relationship Guide. These dashboards must support human oversight, accountability, and environmental attunement.

Key UX Principles

- **Make Systems Legible:** Show what the system knows, what it's doing, and why—across time, space, and actors.
- **Respect Contextual Boundaries:** Provide contextual cues for ecological thresholds, operational zones, and consent boundaries.
- **Support Relational Thinking:** Display how decisions affect co-located systems, ecological health, or shared infrastructure.
- **Enable Responsive Redress:** Allow humans to override, intervene, or request explanation—with clear audit trails.
- **Show Lifecycle & Identity:** Clearly display agent identity, trust level, and lifecycle phase.

UX Components by Facet

Every system, even industrial IoT systems should have human readability, human-in-the-loop and human accountability built in. How do you realise this in user interfaces? Here are some User Experience (UX) components to consider for each of the relationship facets.

Facet	Dashboard UX Components
1. Communication as Attunement	• Dynamic signal visualisation with source metadata • Confidence levels and explainability pop-ups • Mode-sensitive views (e.g. operator vs. environmental observer)

2. Trust as a Living Relationship	• Verifiable trust scores linked to behaviours • Timeline view of role/authority changes • Credential issuer provenance inspection
3. Mutual Obligation Management	• Visual map of system dependencies • Alerts for imbalances or deferred reciprocity • Reciprocity score or "network fairness" index
4. Boundaries as Relational Ethics	• Real-time boundary status visualisation • Consent history logs with override status • Environmental or ethical breach warnings
5. Learning as Co-Evolution	• Model update history with learning source attribution • Human feedback prompt integration • Community learning dashboards showing shared insights
6. Redress as Repair and Restoration	• Error timeline with proposed fixes and status • Redress request queue and resolution history • Agent self-report mechanisms
7. Lifecycle as Rite and Rhythm	• Agent status badge (e.g. Active, Deprecated) • Lifecycle change logs and system sunset notifications • "End-of-life" summary view for audit and memory

Human-in-the-Loop UX Additions:

- Manual override buttons tied to real-time alerts
- Operator annotations on decisions or anomalies
- Accountability chain visualisation (who authorised what and when)
- Consent and policy configuration panel
- Handoff view for transitions between automated and manual control

Conclusion

The Entangled AI Relationship Guide is a call to reimagine how we design, govern, and relate to AI systems in a world marked by environmental urgency and systemic interdependence. As AI becomes embedded not just in personal tools but in public infrastructure, planetary

sensing, and ecological decision-making, we must go beyond efficiency and optimisation to design for relationship, responsibility, and regeneration.

This guide contributes a language and toolkit for building AI that is entangled, deeply aware of and accountable to its social, ecological, and infrastructural contexts. But this is only the beginning. To realise a more trustworthy and environmentally connected AI future, we must:

- Build cross-disciplinary coalitions that link technologists, ecologists, policymakers, and communities
- Develop regenerative architectures that prioritise ecological health, energy accountability, and repair
- Create standards and protocols for machine responsibility, relational consent, and lifecycle transitions
- Support living labs and public testbeds to experiment with post-anthropocentric AI in real-world settings
- Encourage open, pluralistic governance models that embed human oversight without enforcing human dominance

This framework is designed to evolve. We invite you to extend it, challenge it, and localise it to your domain. The next generation of AI will not only serve us, it will live with us. Let us ensure it does so with respect, reciprocity, and relational intelligence.

Examples from the Verifiable AI Hackathon 2025

Many of the entries in the 2025 Verifiable AI Hackathon demonstrate how self-sovereign identity technologies can be used to support trustworthy human-AI relationships. Here is a summary of which example entries support one or more of the facets.

Hack Name	Short Description	Communication as Attunement	Trust as a Living Relationship	Care as Responsibility & Reciprocity	Boundaries as Relational Ethics	Learning as Co-Evolution	Redress as Repair & Restoration	Lifecycle as Rite & Rhythm
Identone	Verify humans & agents for voice interactions							
Kith AI Agent Passport	Verifiable Credentials for AI Agents							
TrustyBytes	A marketplace for trusted data							
crdbl	Verifiable content provenance							
Cheqdeep	A decentralized solution for verifying media authenticity							
Aeonix	Verified Human-in-the-Loop AI Training							
Viskify	A Verifiable AI Hiring Platform							
Aeonix	Verified Search Agent							

See below for more details of these example implementations from the Hackathon.

Identone

This project is focused on enabling business-to-consumer secure and trustworthy interactions between humans and AI-powered voice agents. As voice AI rapidly becomes the norm in customer service and call center operations, the need for trust in these interactions is more critical than ever. At the same time, the rise in AI-driven call scams, OTP phishing, and caller ID spoofing poses significant risks to both individuals and organizations.
Our solution addresses these challenges by establishing bi-directional trust in phone-based human-AI interactions. When a person receives a call from an AI agent, they should be able to verify the authenticity of the caller. Similarly, the AI agent must be capable of validating the identity of the person it is engaging with. This mutual authentication ensures safe, secure, and trustworthy voice communications in an increasingly automated world.
There are 2 major use cases handled in this project:

- Verify the AI agent's identity using verifiable credentials and DID-linked resources before the call begins.
- Verify the caller's identity during the call using the organization's digital wallet and verifiable credentials.

See the demo here: https://youtu.be/OCpok8pqOz0
See the build details here: https://dorahacks.io/buidl/26280

Kith AI Agent Passport

A decentralised trust layer for AI agents. This project enables agents to hold cryptographically verifiable credentials (VCs), linked to their Decentralised Identifiers (DIDs)

and DID-Linked Resources (DLRs) via Cheqd studio. Built for proof of personhood, privacy preservation, and secure agent authentication.

See the demo here: https://www.youtube.com/watch?v=BmLNRO-adOQ
See build details here: https://dorahacks.io/buidl/26335

Trusty Bytes

Trusty Bytes is a marketplace for AI agents to access trustworthy data using Model Context Protocol (MCP) and the cheqd trust network.
How it works

- Authentication: Users log in to the Trusty Bytes platform using their preferred web2 or web3 account via Privy.
- Dataset Listing: Data providers list their datasets (currently Candles or Sentiments) for sale.
- Dataset Discovery: Users browse the marketplace to find datasets relevant to their AI agents' needs.
- Purchase: Users purchase access using a smart contract, currently supporting payments with native tokens only.
- Credential Issuance: Upon successful purchase, the Trusty Bytes platform issues a Verifiable Credential (VC) on the cheqd network. This VC contains metadata about the dataset, including information about the data provider who sold it.
- AI Agent Integration: The user obtains an access key from the MCP server settings page within the Trusty Bytes platform and configures their AI agent with this key to connect to the Trusty Bytes MCP server.
- MCP Server Connection: The AI agent connects to the MCP server, authenticating itself using the provided access key.
- Data Access: When the agent requests data using tools like get_candles or get_sentiment, the MCP server verifies the agent's purchase/access rights and streams the requested dataset.
- Provenance Verification: The agent can use the get_dataset_issuer tool. This tool retrieves the issuer's DID and trust framework details associated with the dataset from the cheqd network, allowing the agent to verify the data's origin and trustworthiness.

See the demo here: https://www.youtube.com/watch?v=GTWAg9wkQpM
See the build details here: https://dorahacks.io/buidl/26048

crdbl, a more credible web by enabling the creation, verification, and consumption of content credentials

crdbl, powered by cheqd, is a trust layer for a more credible web that turns any human- or AI-generated content into a "crdbl"; a verifiable credential anchored to a decentralized identifier. When other crdbls are supplied as context, an AI engine recursively checks each new claim against the context, ensuring only credible credentials are issued thus weaving a composable, cryptographically linked graph of provenance where every assertion can be traced, proven, and marked as verifiably credible.
A browser extension lets users mint, reference, and view verification status in-page, while an API offers AI agents programmatic issuance, access, deeper integrations, and independent verification. The result is a self-reinforcing web of trust that makes research, journalism, content ownership, synthetic compositions, and AI workflows instantly auditable, paving the way for a more credible internet with new monetization opportunities for content originators and synthesizers.

See the demo: https://dorahacks.io/buidl/26336/
See the build details: https://dorahacks.io/buidl/26336/

CheqDeep, A decentralized solution for verifying media authenticity

CheqDeep - Prove your content is real!
Picture this: You're walking down the street when suddenly, you see someone floating in mid-air! You quickly grab your phone and record this incredible moment. But when you share it, everyone thinks it's AI-generated. "This must be fake!" they say.
This is exactly the problem CheqDeep solves. Using cheqd's blockchain technology, we create an immutable digital certificate that proves your video is real - recorded on your device, at that exact time. It's like having a notary public for your digital content, but way cooler.
In a world where seeing is no longer believing, CheqDeep ensures your "I saw it with my own eyes" moments are backed by blockchain-powered proof.

See the demo:
https://www.loom.com/share/b34a7ca641fc4b56a03e8488dc027a41?sid=1b8e0194-c93b-438b-bcf7-90222310a56b
See the build details: https://dorahacks.io/buidl/26299

Aeonix Verified Human-In-The-Loop AI Training

This project builds a bridge between verified social identity and human-in-the-loop (HITL) AI training by leveraging cheqd's decentralized identity infrastructure. The system allows users to link and authenticate off-chain identifiers — such as email and Telegram accounts — to a primary Decentralized Identifier (DID), without compromising data privacy or unifying datasets.

Upon successful verification, users are granted Verifiable Credentials (VCs) that allow them to access AI model training features within the aeonix explorer. By aligning user incentives through credential based achievements and privacy-preserving identity, this build enables secure community-driven fine tuning of large-scale models — with defenses against bot activity and data poisoning attacks.

See the demo here: https://www.youtube.com/watch?v=1yYr45c6UB4
See the build details here: https://dorahacks.io/buidl/26284

Viskify a Verifiable AI Hiring Platform

Powered by cheqd and Verida, Viskify is a decentralized talent platform that issues verifiable credentials and delivers AI insights from private, user-owned data — with deterministic DIDs, usage-based billing, and zero smart contract deployment.

User-Journey Snapshot

Candidate

- One-click DID creation through Cheqd Studio — no wallet needed.
- Upload credentials → UNVERIFIED · PENDING · VERIFIED/REJECTED lifecycle.
- AI-graded skill-checks; a passing score automatically mints a cheqd VC.

Issuer

- Self-service onboarding with admin approval.
- Approve / Reject verification requests — approval signs a Verifiable Credential via Cheqd Studio.

Recruiter

- Full-text talent search with verified-only toggle.
- Kanban pipelines, AI fit-summaries cached per recruiter × candidate.

Admin

- Issuer approvals, role upgrades, credential revocation.
- Platform DID rotation and pricing updates, all through Cheqd APIs.

See demo here: https://www.youtube.com/watch?v=hiay-fuhmuk
See build details here: https://dorahacks.io/buidl/26297

Aeonix Verified AI Search Agent

The Verified AI Search Agent introduces a verifiable layer of trust to the aeonix explorer's AI-driven search results. By assigning a DID to the AI search agent itself — including the origin of the application, its data sources, and configuration metadata — the build enables users to **transparently trace why a search result appeared and whether it can be trusted.**

This update is pivotal for establishing **verifiable provenance** in an environment where AI-generated content can often feel opaque or unverifiable. It addresses both trust and scalability by using cheqd's decentralized identity stack to anchor critical metadata, without requiring every result to be individually signed or credentialed.

See the demo here: https://www.youtube.com/watch?v=zlCkt2o-SGA
Build details here: https://dorahacks.io/buidl/26289