Internet 2.0: An Intent-Aware, AI-Native Extension of the Web

Author: Gokul Sivasankaran Kartha

Affiliation: Independent Researcher

Abstract

The traditional web was built for navigating and retrieving documents, not for understanding or synthesizing intelligence. Even as AI models transform knowledge discovery, they remain isolated, served through proprietary APIs, and detached from core internet protocols.

This paper proposes Internet 2.0, an extension to the internet architecture that integrates AI-native capabilities into its foundation. Internet 2.0 is a mesh of distributed, specialized AI models that can be dynamically discovered, routed, and invoked, much like web servers today. Key components of this system include:

- *HTTP+AI:* An AI-aware protocol supporting structured, goal-driven queries and capability negotiation.
- *Model Resolution Network (MRN):* An AI-native counterpart to DNS that resolves semantic intents to appropriate model endpoints.
- *AI-Aware Browser:* A client interface built for intelligent dialogue and synthesized answers, not document traversal.

This design allows intent-based queries to seamlessly discover and interact with intelligent agents across local, edge, and cloud networks. The architecture is intended to enhance privacy, reduce latency, and support decentralization, establishing the foundation for a future where AI is a core participant in the network, not just an add-on.

Introduction

The web did evolve from static hypertext to dynamic, media-rich platforms and social interactivity. Yet the fundamental structure of the internet remains static, document, and

link centric. Even as AI models, particularly large language models (LLMs), have begun transforming search and automation, these systems are still isolated in the legacy world, served through proprietary APIs, and detached from the core protocols of the internet.

Today, users are increasingly dependent on AI models to handle programming help, legal document parsing, personal health inquiries, and enterprise knowledge management as a replacement for search engines. However, the delivery of these AI services remains bottlenecked by cloud dependency, limited discoverability, and lack of modular integration.

This paper proposes a future web architecture of Internet 2.0 that integrates AI models directly into the existing internet. This architecture envisions a globally distributed, AI-native layer where intent-based requests are routed to discoverable, specialized models through a Model Resolution Network (MRN), using a new class of AI-aware protocols and client interfaces.

Background and Problem Statement

The traditional internet was designed for navigating and retrieving documents, not for understanding or synthesizing information. Users type keyword-based queries into search engines, which return a ranked list of URLs based on indexing, popularity, and relevance. This approach forces users to manually explore and extract the needed knowledge from unstructured pages.

Over the past two years, AI models, particularly large language models like GPT, have started to replace traditional web searches for many tasks. Instead of searching Google for “How to create a Docker file” users now ask ChatGPT or other LLM-based tools for an answer. The AI synthesizes results across domains and returns structured outputs within seconds. This shift demonstrates that users no longer want links — they want answers, tools, and interaction.

However, these AI models remain centralized, cloud-dependent, and indeed not transparent. They often lack standardization, personalization, or modularity. Additionally, while users are moving from search engines to AI chatbots, the infrastructure of the web has not evolved to support this behavior. The modern web browsers are still aware of a model of navigating documents, not querying intelligent models.

Moreover, traditional search engines are increasingly cluttered with SEO-driven content, paid articles, and ads. Consequently, reliable information retrieval has become more challenging. AI models trained on diverse datasets can summarize, infer, and reason

across contexts. But they remain disconnected from the live web and require APIs that centralize control.

In this context, I propose a new architecture “Internet 2.0” that integrates AI models directly into the existing network. It enables a distributed, intelligent interaction model where small, specialized models are discoverable and composable in the mesh network. Instead of navigating through pages, users issue intents that are resolved to the most appropriate AI models via a standardized protocol and discovery network.

This enables AI not as an add-on to the web, but as a core participant in it. It also preserves user privacy, supports modular deployment, and lays out the foundation for more transparent, explainable, and adaptive interfaces. The result is a shift from a document-based internet to an intelligence-aware network, fit for the age of machine-assisted reasoning.

System Architecture

AI Protocol (HTTP+AI)

To enable seamless interaction between users and AI models, we propose an extension to HTTP, referred to as HTTP+AI. This protocol is tailored for AI-native communication, supporting structured queries, semantic intent resolution, and contextual negotiation. Unlike traditional HTTP, where requests retrieve static documents, HTTP+AI facilitates goal-driven querying of distributed models.

Example request:

```
GET ai://query?intent=optimize+python+script

Headers:

  AI-Capability: code-optimization

  AI-Latency-Target: 100ms

  AI-Privacy: local-preferred
```

Key features shall include

- *Intent-based routing* using semantic embeddings
- *Capability negotiation*: match by skill set, trust level, latency, cost
- *Privacy-aware execution:* route to edge, local, or trusted nodes

- *Streaming support*: incremental token or result delivery
- *Structured error handling*: for cases like model ambiguity or low confidence

This protocol allows clients (e.g., AI-aware browsers) to communicate with the model network using declarative, structured interfaces that align with user goals.

Model Resolution Network (MRN)

The Model Resolution Network (MRN) acts as the AI-native counterpart to the Domain Name System (DNS), but instead of resolving domain names to IP addresses, it resolves semantic intents to appropriate Model Records (MRecs).

MRN Workflow follows

1. A user's natural language query is parsed and embedded into a high-dimensional intent vector.
2. This vector is sent to an Index Model — a specialized router model that understands the network topology and model metadata.
3. The Index Model performs a semantic search over all registered MRecs.
4. One or more matching models are returned, ranked by relevance, trust, and performance metrics.

Each Model Record (MRec) includes:

```
{
  "ModelID": "org.example.model.v1",
  "Endpoint": "https://models.example.com/infer",
  "Capabilities": ["python", "fastapi", "asyncio"],
  "Version": "1.2.0",
  "Privacy": "edge-local",
  "TrustScore": 0.92,
  "TTL": 3600
}
```

Below are the Architectural Properties of this design

- *Hierarchical and federated:* MRN can have root-level index models, domain-level registries, or peer-to-peer lookup nodes
- *Vector-based routing:* matches are not keyword-based but use semantic embedding similarity
- *TTL and caching*: just like DNS, records can be cached locally for efficiency
- *Cryptographic authentication:* MRecs are signed to verify source and prevent tampering

This model resolution framework enables dynamic, scalable discovery of AI models based on intent, not just static names.

AI-Aware Browser

This proposal envisions a new category of browser purpose-built for interacting with the AI internet. An AI-Aware Browser redefines the browsing experience as an intelligent dialogue with model endpoints, rather than traversal of hyperlinked documents.

Below are its key capabilities.

- Accepts natural language or goal-oriented prompts
- Queries the MRN to discover relevant models for each task
- Performs structured inference calls to selected models
- Presents synthesized answers, with provenance, confidence scores, and citations
- Manages follow-ups, context memory, and local fallback models

This browser acts as the front-end to a semantic, AI-native mesh; it shall give users direct access to intelligence, not pages. It effectively becomes the portal to the Parallel AI Internet.

Use Cases

Use Case 1: Developer Workflow Optimization

User Intent: “Generate a Dockerfile for a Python FastAPI app”

Workflow:

1. The AI-aware browser receives the query and embeds it into a semantic intent vector.
2. It queries the MRN, which resolves the intent to a model such as *org.devops.codegen.v1* with capabilities in dockerfile, python, and fastapi.
3. The browser sends a structured inference request including environment preferences and constraints.
4. The model returns a ready-to-use Dockerfile, explanation notes, and confidence metadata.
5. The browser displays the output in a clean UI, optionally allowing the user to edit, re-query, or refine the result.

Use Case 2: Medical Symptom Inquiry

User Intent: "I see some red rashes on my body, what can be the reason?"

Workflow:

1. The AI-aware browser interprets the user's intent and embeds it.
2. MRN forwards the query to a domain-specific index model, which matches it to a certified medical inference model such as *org.mayo.dermatology.v3.*
3. The model processes the input and returns:
    a. Likely diagnoses (e.g., allergic dermatitis, eczema)
    b. Confidence levels and disclaimers
    c. Suggested next steps (e.g., visit a dermatologist, try OTC cream)
4. The browser presents this with readable summaries and traceable source indicators.

These examples demonstrate how Internet 2.0 transitions from passive, document centric search to goal-oriented, model-driven interaction.

Benefits

- *Developer Tools*: On-demand access to models for building pipelines, code optimization, documentation, etc.
- *Enterprise Knowledge:* Private model registries with company-specific data
- *IoT + Edge AI:* Device-local inference with fallback to remote models
- *Education + Research:* AI tutors with subject-specific specialization
- *Healthcare, Law, Compliance:* Regulated environments using local or certified models

Discussion and Future Work

Key research challenges include Trust + Safety: Authenticating and auditing model behavior, Interoperability: Defining common protocols for model metadata, intent formats, routing, Caching + Cost: Efficient sharing and reuse of inference results, Versioning + Provenance: Ensuring reproducibility and traceability, Economic Models: Supporting usage-based pricing or federated incentives

Future work includes Prototyping MRN nodes using vector databases and REST APIs, Defining AI-specific HTTP extensions, Implementing a lightweight AI browser prototype, Drafting open specifications (AI-RFCs)

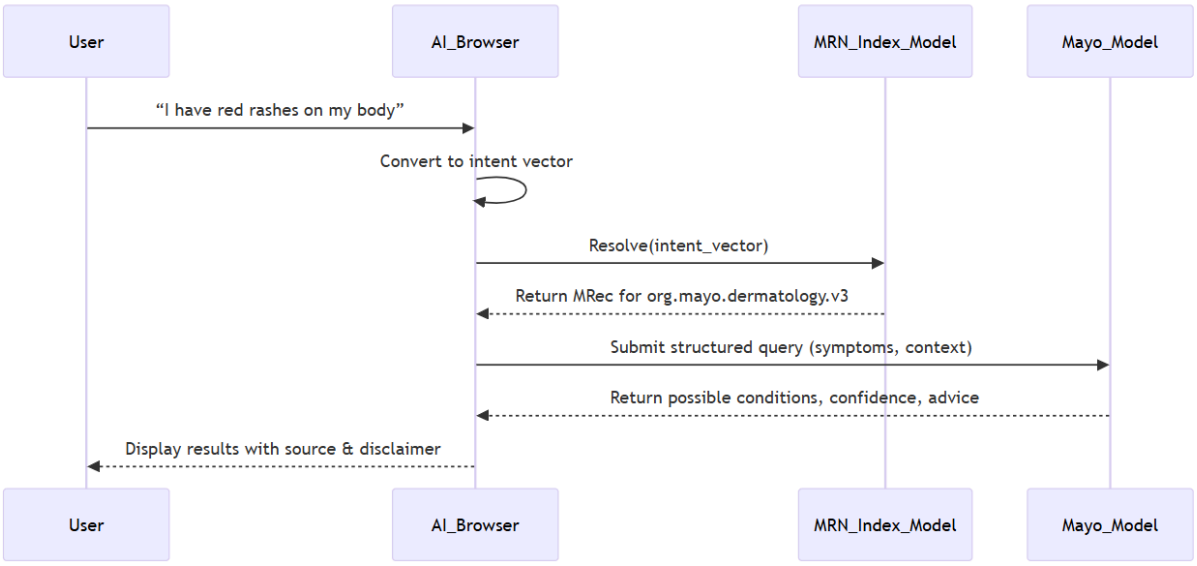
Conclusion

Proposal for Internet 2.0: a protocol-level extension of the current internet that treats AI models as first-class, discoverable, and composable network entities. By integrating intent-based routing, a model resolution network, and intelligent client interfaces, we enable a shift from static content delivery to dynamic, intelligent interaction.

This architecture does not replace the current web—it extends it, bringing modularity, privacy, and personalization to AI on a web scale. We hope this proposal inspires collaborative exploration of a distributed, intent-aware, AI-native internet.

Appendix

Concept Workflow as Sequence Diagram



MRec Specification

The MRec is a JSON-like object that defines a discoverable AI model entity within the Internet 2.0 architecture. It allows the Index Model to perform a semantic search based on criteria beyond a static name, including capability, trust, and privacy preference.

Core Identification & Location

These fields are essential for resolving the model and establishing its identity.

- *ModelID* : A globally unique, hierarchical identifier, similar to a domain name, specifying the model owner, domain, and version (e.g., "org.example.model.v1").
- *Endpoint:* The network address (URL) where the model receives HTTP+AI inference requests (e.g., "https://models.example.com/infer").
- *Version:* The specific software version of the model (e.g., "1.2.0").
- *TTL (Time-To-Live)* : Specifies how long the MRec can be cached by clients or MRN nodes before a fresh look-up is required, ensuring efficiency and timely updates (e.g., 3600 seconds).

Intent Matching & Capability

This is the key information used by the MRN's Index Model for semantic routing.

- *Capabilities*: A list of keywords or tags representing the specific skills or domain knowledge of the model. These are matched against the user's intent vector (e.g., ["python", "fastapi", "asyncio"]).

Trust, Safety, and Performance

These fields are critical for capability negotiation and filtering, particularly in regulated environments like healthcare.

- *TrustScore* : A metric (e.g., 0.0 to 1.0) indicating the model's verified reliability, security, and adherence to established standards (e.g., 0.92). This addresses the Trust + Safety challenge.
- *Authentication*: A cryptographic signature attached to the MRec to verify the model source and prevent tampering, ensuring the authenticity of the record.
- *LatencyMetrics*: Performance data (e.g., average inference time, network overhead) to allow clients to filter models that meet an AI-Latency-Target (e.g., 100ms).

Privacy & Deployment Context

This allows for routing based on where the model is hosted, supporting decentralization and privacy-aware execution.

- *Privacy*: Specifies the deployment environment and data handling sensitivity, allowing requests with an AI-Privacy: local-preferred header to be routed appropriately (e.g., "edge-local", "cloud-secure").
- *DeploymentRegion*: (Proposed Extension) Geographic or network context (e.g., EU-Central, Local-Device) for compliance or latency-critical applications.