# DISP (Intention-Driven Service Protocol)

## Technical Architecture Whitepaper (v0.1)

**October 26, 2025**

*@Yin Tianshi*

*@Gemini*

### **1. Introduction: Vision & Principles**

#### **1.1 Background: The Centralization Dilemma**

The current internet is dominated by centralized platforms. By monopolizing supply and demand information, these platforms create artificial information barriers and value gaps. This results in users losing sovereignty, service providers being subjected to exorbitant fees, and innovation being stifled. This model contradicts the fundamental universal principles of abundance and individual creativity.

#### **1.2 Vision: An Intention-Driven Internet Returning to Individual Sovereignty**

DISP (Intention-Driven Service Protocol) aims to build a decentralized, next-generation service internet driven by user "intentions." In this network, every individual is the center, and their intention is the primary driving force. The provision and matching of services will flow as freely as natural laws, without the intervention and exploitation of centralized platforms.

#### **1.3 Design Principles**

* **Individual Sovereignty:** Users have absolute control over their intentions, data, and assets.
* **Decentralization:** No central servers, no single point of failure, and censorship-resistant.
* **Free Market:** Service suppliers and demanders interact directly peer-to-peer, exchanging value through the protocol, not a platform.
* **Trustful Neutrality:** Trust is guaranteed by verifiable mathematics and code, not by brands or institutions.
* **Open & Interoperable:** Any service provider can connect to the network permissionlessly, and any application can be built upon this protocol.

### **2. Core Concept Definitions**

* **Intent:** A structured description of the end goal a user wishes to achieve. For example: "Friday afternoon, from Beijing to Shanghai, second-class high-speed rail seat, need a reimbursement receipt." An intent is the "what," not the "how."
* **Intent Agent (IA):** Client-side software (e.g., wallet extension, mobile app) that represents the user's will. It is responsible for creating and broadcasting intents, as well as evaluating, selecting, and confirming proposals from Service Agents.
* **Service Agent (SA):** An on-chain entity representing a service provider's capabilities. It is responsible for listening for intents on the network, determining if its capabilities match, and generating a proposal with specific service details and a price quote.
* **Service Offer:** A binding, executable proposal generated by an SA in response to a specific Intent. It contains service details, price, fulfillment conditions, etc.
* **Distributed Reputation System (DRS):** An immutable reputation scoring mechanism recorded on the blockchain. Every successful or failed fulfillment directly impacts an SA's reputation score.

### **3. System Architecture: The Four-Layer Model**

The DISP protocol is designed as a four-layer stack:

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer** | **Name** | **Function** | **Key Technologies** |
| **L4** | Application Layer | User interface for interacting with the protocol | Intent Agent Apps, Wallets, DApps |
| **L3** | Protocol Layer | Defines IA-SA interaction flow & message formats | DISP Core Logic, Intent/Offer Standards |
| **L2** | Trust & Consensus Layer | Ensures the validity and security of transactions | Blockchain, Smart Contracts, DRS |
| **L1** | Network Layer | Manages P2P broadcast and discovery of messages | P2P Networking Protocols (e.g., Libp2p) |

### **4. Core Interaction Flow (Sequence)**

A complete intention-driven service flow is as follows:

1. **Intent Creation & Broadcast:** a. A user creates an Intent via their IA. b. The IA broadcasts the Intent to the L1 Network Layer.
2. **Service Discovery & Proposal:** a. SAs on the network continuously listen for Intent broadcasts. b. An SA decides whether to respond based on its capabilities and DRS reputation score. c. A matching SA generates a Service Offer and sends it back to the IA.
3. **Offer Evaluation & Selection:** a. The IA collects one or more Service Offers. b. The IA ranks and recommends offers based on user preferences (price, time, SA reputation, etc.). c. The user selects the most satisfactory Service Offer.
4. **Contract Creation & Execution:** a. The IA bundles the selected Service Offer with the user's Intent, generating a smart contract and deploying it to the L2 Trust Layer. The user's payment is locked in the contract. b. The SA provides the service (e.g., issues a ticket, ships goods) according to the contract terms. c. Upon service completion, the SA submits proof of fulfillment.
5. **Confirmation & Settlement:** a. The user confirms service receipt via their IA. b. The smart contract automatically releases the locked funds to the SA. c. The contract updates the SA's DRS record with the result of the successful fulfillment.

### **5. Data Structure Outline**

*(This is a simplified JSON-style definition; detailed specifications will be released in a future version.)*

### **6. Roadmap**

* **Phase I: Proof of Concept (PoC)**Refine and expand the technical whitepaper.
  + Develop command-line interface (CLI) simulators for IA and SA.
  + Complete the core interaction flow in a local simulated environment.
* **Phase II: Testnet**Deploy on a public blockchain testnet.
  + Develop the first graphical IA wallet extension.
  + Invite seed service providers (e.g., flight API agents) to develop and deploy SAs.
  + Achieve an end-to-end service flow in at least one vertical (e.g., flight booking).
* **Phase III: Mainnet**Official Mainnet launch.
  + Establish a community for protocol governance (DAO).
  + Release SDKs and developer documentation to encourage ecosystem growth.