ELECTROLIPS / CONCEPT ARCHIVE

Wormhole Network & Information Fountains

Thread recap, systems outline, image inventory and citation register

19 NOVEMBER 2026 | WORKING CONCEPT RECORD

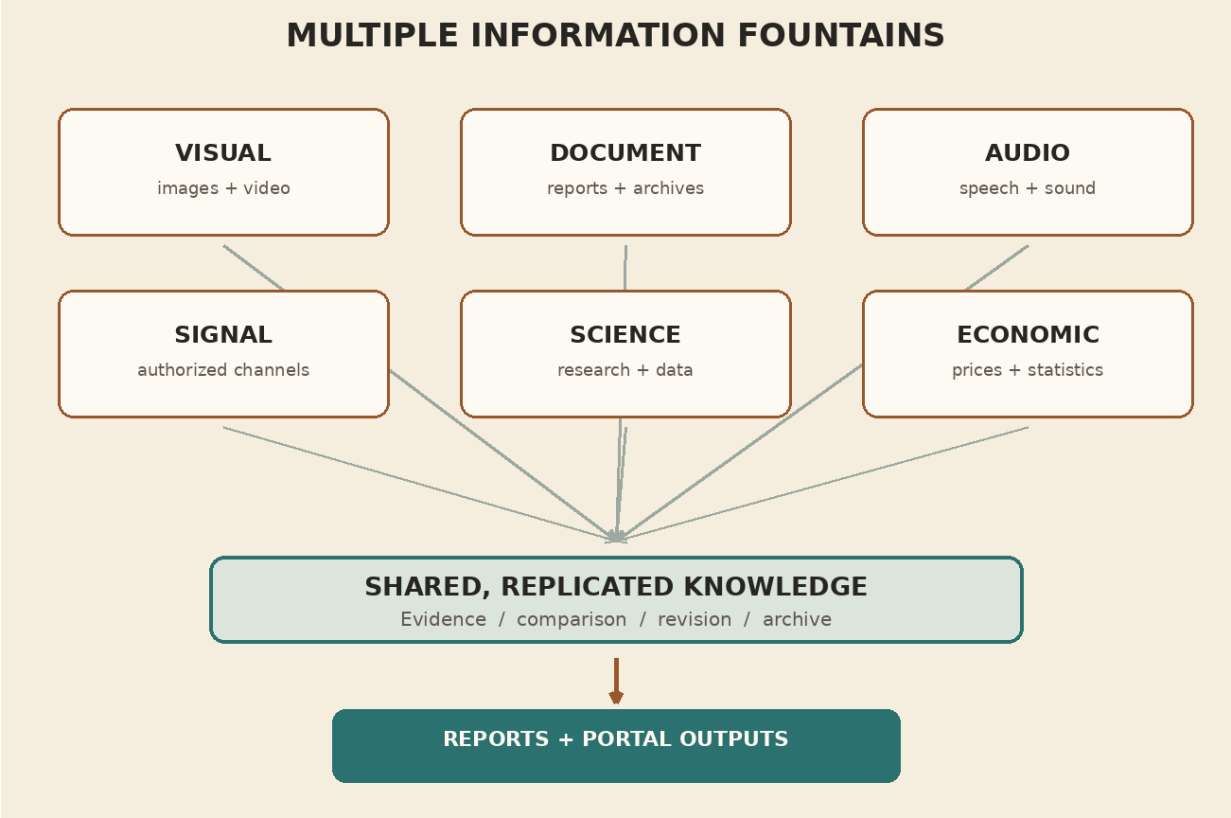


Figure A. The six continuously operating fountain services feed a shared knowledge layer.

Scope: This report summarizes the entire conversation visible in the current thread, from the original webpage-wormhole question through the requirement for multiple independent information fountains. It is a recap, not a verbatim transcript.

Evidence note: Historical chat citation markers are preserved exactly. Their underlying prior-turn URLs cannot be recovered from the marker text alone. The Sources appendix therefore supplies newly verified topical references; these are not represented as exact reconstructions of the original citations.

OPEN CONTENTS

Contents

Executive summary
Chronological recap - all 20 stages
Network system - proposed components
Packet and software sizing
Multiple Information Fountains
Engineering findings and unresolved choices
Image inventory - all visual references
Original citation markers - all 30
Verified subject sources and image URLs

Executive summary

Two linked ideas emerged. First, a numbered-portal communications overlay would allow a visitor to dial a logical portal number while distributed peers discover a listening receiver, relay an authenticated message and establish an interactive session. Second, a collection of independent Information Fountains would continuously gather evidence from different media, analyze each item and update shared records while other fountains stay online during maintenance.

The network discussion evolved from a basic iframe (rejected because it still used a URL) to a distributed operator, random relay discovery, message persistence, cross-network transport, 16,000-byte and then 1,600-byte logical packets, and compact listener services. The discussion did not produce a tested, deployed network. The final Information Fountain design is likewise a proposed architecture, not a running system.

Practical synthesis: use cooperative peer discovery, durable message queues, signed/encrypted envelopes, authorized gateways, and separately restartable analysis workers. Logical portal numbers and listener mobility are possible at the application layer; physical networks still depend on addressable links and resource limits.

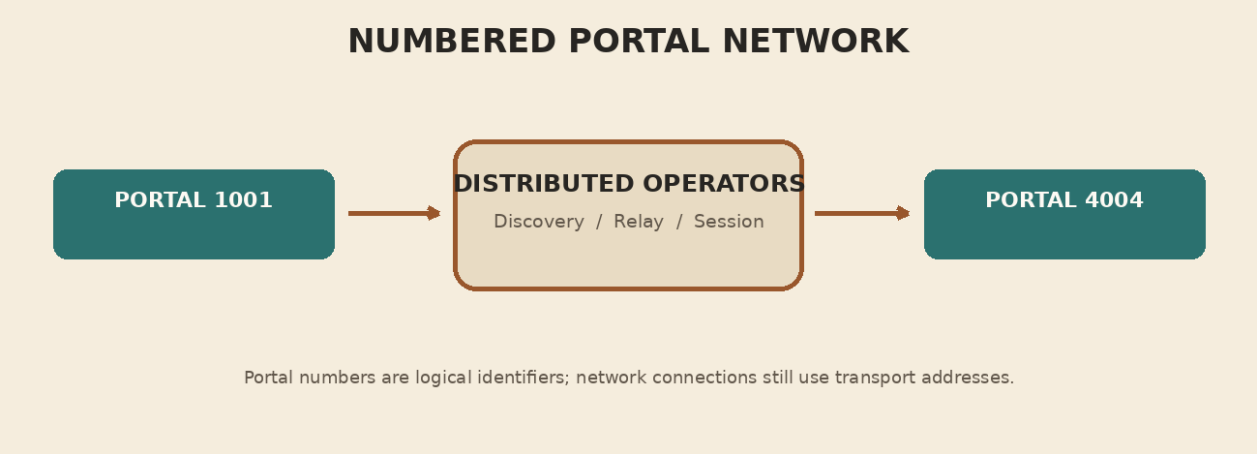


Figure 1. Numbered portal identifiers with distributed operator services; not an IP-address-free transport.

Chronological recap - all 20 stages

01 Webpage wormhole

The initial request was to move a user between live webpages without a visible or explicit URL. An iframe overlay was proposed, but rejected because it still required a destination URL internally.

Engineering conclusion: Browser navigation and network addressing are distinct; concealing a URL does not eliminate addressing.

02 Virtual serial entry

A hosting site with a virtual serial-port-like entry point was suggested. The distinction was drawn between a browser-to-device serial API and a network service that only behaves like a serial connection.

Engineering conclusion: A server-side virtual gateway is feasible; native Web Serial does not directly connect arbitrary remote hosts.

03 Numbered portal operator

A telephone-switchboard analogy introduced numbered portals (examples: 1001, 2002, 3003, 4004). A visitor enters a number and an operator finds a corresponding live portal.

Engineering conclusion: Numbers can be logical portal IDs resolved to authorized services.

04 Remove the permanent hub

The operator was redesigned as many distributed cooperating programs, not a central switchboard. Packets carry routing identifiers and encrypted content; recipient authentication happens at the destination.

Engineering conclusion: Peer discovery and relays can be decentralized; reachability still requires underlying endpoints.

05 Existing wandering relays

I2P, Tor, libp2p, Nym, mobile agents, and other systems were compared. They dynamically select circuits, distribute encrypted data, or move authorized code.

Engineering conclusion: Most systems move messages or paths, not executable agents moving freely on unrelated servers.

06 Random locations and listeners

The goal shifted toward a randomized communication path without a known receiving machine. Random-walk, gossip, epidemic routing and publish-subscribe systems were described.

Engineering conclusion: A participating listener can receive an announcement without the sender specifying which host receives it.

07 Foreign buffers and pirate listeners

The proposed packets would circulate through third-party buffers until passing a listener. This was distinguished from normal network operation: foreign router buffers are not general-purpose public message stores.

Engineering conclusion: Interception or execution on unrelated systems requires authorization; ordinary routers cannot be used as arbitrary mailboxes.

08 Large public information highways

The design aimed to use public communication infrastructure rather than a dedicated operator-owned relay system. It was reframed as an overlay carried by participating public services.

Engineering conclusion: Public transport infrastructure is useful; availability does not imply unrestricted access to operator equipment.

09 Blending and traffic visibility

The idea of blending encrypted transmissions into normal traffic was discussed. The distinction between confidentiality and concealment was emphasized.

Engineering conclusion: Encryption protects payload contents; network operators may still see traffic patterns and metadata.

10 Persistent global delivery

A combination of libp2p, DHT, Nostr relays and store-and-forward methods was proposed to let independently located listeners discover messages.

Engineering conclusion: Infinite lifetime is not guaranteed. Durability needs replicated storage, expiry rules, quotas and retry budgets.

11 Multiple transport systems

The user introduced MQTT, SMS, VoIP, traditional telephone paths, satellite or radio gateways. The concept became transport-independent.

Engineering conclusion: Use small recognition announcements on limited-bandwidth transports and larger payloads on data-capable networks.

12 16,000-byte logical packet

The initial candidate was 16,000 bytes (15.625 KiB). Internet transports can readily carry this as an application message; SMS would need many fragments.

Engineering conclusion: Logical message size is not the same as a physical Ethernet, UDP or SMS packet.

13 Text capacity of 16,000 bytes

A simple MS-DOS-style ASCII estimate was requested. For single-byte ASCII, 16,000 bytes can hold 16,000 characters, including any spaces, or roughly 2,600 words at six bytes/word.

Engineering conclusion: Encoding and whitespace affect the count. Non-ASCII Unicode text may require more bytes per character.

14 Reduce the message to 1,600 bytes

The message target was reduced by a factor of ten. An illustrative allocation split the 1,600 bytes among identifiers, recognition tag, routing, security and encrypted data.

Engineering conclusion: A 1,600-byte logical message can still exceed some safe datagram payloads and may need fragmentation.

15 Size of roaming listener software

Illustrative program sizes were discussed, ranging from a few kilobytes for a decoder to megabytes for a full multi-protocol gateway. A commercial MQTT library footprint was cited as a small-component comparison.

Engineering conclusion: Listener code size depends on runtime, TLS, storage and dependencies; smaller binaries are not inherently undetectable.

16 Research paused

The user indicated a pause until travel to Korea. No operational wormhole-network implementation or test was completed in the thread.

Engineering conclusion: All network specifications in this recap are conceptual proposals, not verified deployments.

17 Visual Information Fountain

A screen displays changing images and a camera or frame-capture program observes them. AI produces descriptive and interpretive text about society.

Engineering conclusion: The camera is optional for on-screen digital material; direct frame capture can preserve source clarity.

18 Think about each image

The user clarified that images should be continually collected and analyzed one by one, rather than merely captioned in batches. The system archives individual analyses, comparisons, uncertainty and revised conclusions.

Engineering conclusion: Separate observable facts from contextual interpretation; re-analysis should retain version history.

19 Many simultaneous fountains

Visual analysis became one service among independent document, audio, signal, scientific and economic fountains. Their results feed a common searchable knowledge layer.

Engineering conclusion: Each fountain needs its own ingest, queue, worker, store, checkpoint and failure recovery.

20 Maintenance without global shutdown

The final requirement is nonstop parallel operation with one fountain periodically entering maintenance while all others continue to operate.

Engineering conclusion: Use independent services, replicated storage and redundant workers; perpetual operation without infrastructure availability cannot be guaranteed.

Network system - proposed components

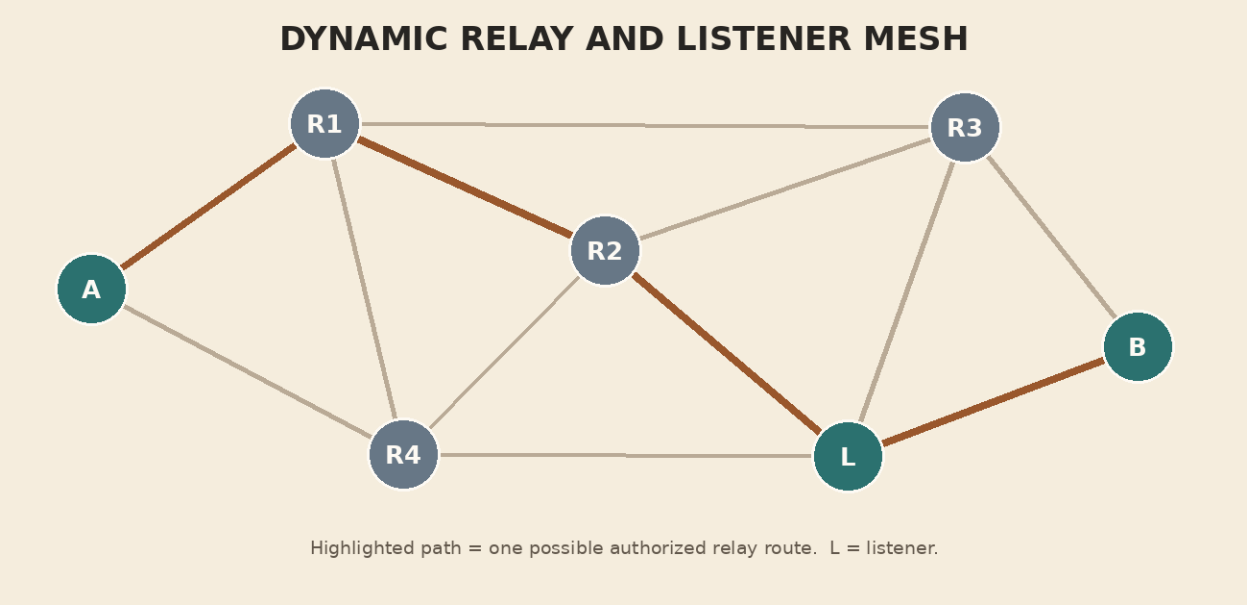


Figure 2. Distributed listening/relay mesh with one possible authorized path.

Portal names - A memorable number such as 1001 or 4004 is a service identifier, not an IP-free physical address.

Discovery - Distributed peer lookup or a replicated registry maps the portal identifier to currently available authorized service endpoints.

Messaging - A signed, encrypted logical message can be delivered through a mixture of cooperative relays.

Listening - A receiving service subscribes to topics/recognition announcements; it verifies sender and destination permissions.

Mobility - The listener can disconnect/reconnect from different host environments or a signed agent can migrate among consenting servers.

Multitransport - MQTT/WebSocket/peer overlay are primary data paths; SMS and telephone serve as limited-bandwidth notifications or fallbacks.

Reliability - Store encrypted copies, acknowledge, retry with bounded backoff, de-duplicate, and expire old messages.

Permission boundary - No plan depends on occupying third-party network infrastructure, executing in foreign buffers, or intercepting traffic without authorization.

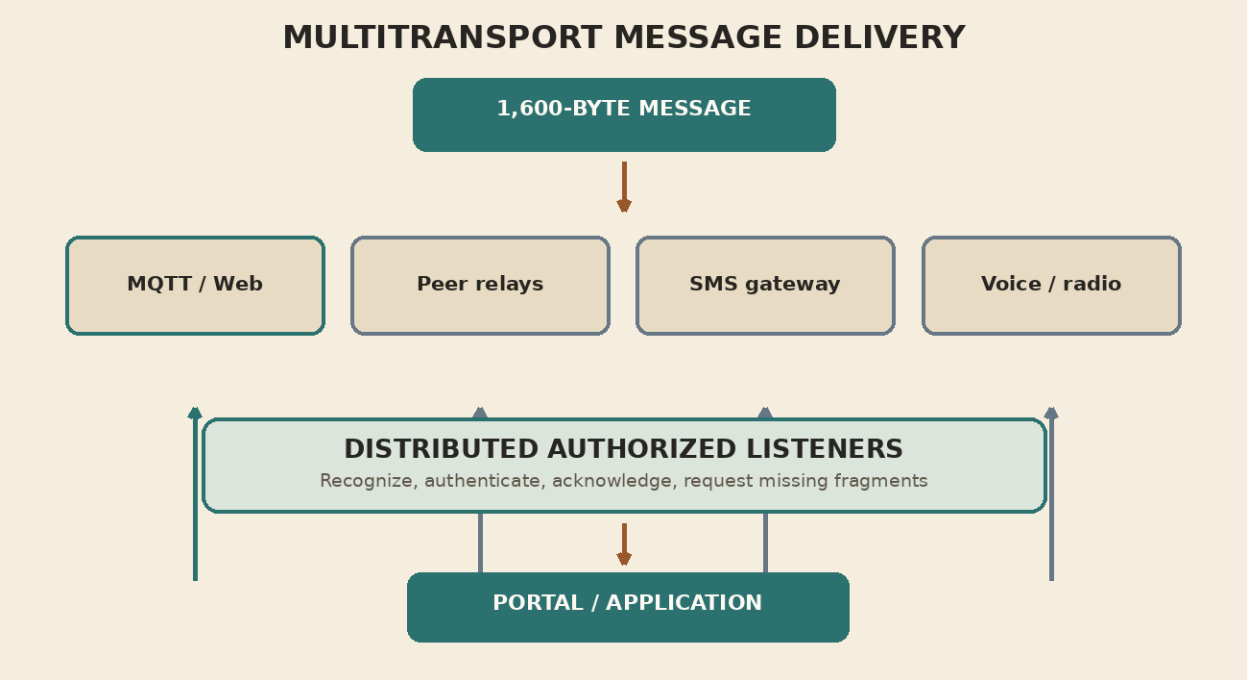


Figure 3. One logical encrypted message can use multiple authorized transmission gateways.

Packet and software sizing

Initial candidate: 16,000 bytes = 15.625 KiB, approximately 16,000 single-byte ASCII characters including spaces. Later candidate: 1,600 bytes = 1.5625 KiB. The 1,600-byte structure below is an illustrative conversation allocation, not an audited message encoding.

Field	Bytes
Message ID	16
Recipient recognition tag	32
Routing and expiry	64
Security metadata	128
Encrypted instructions or data	1,360
TOTAL	1,600

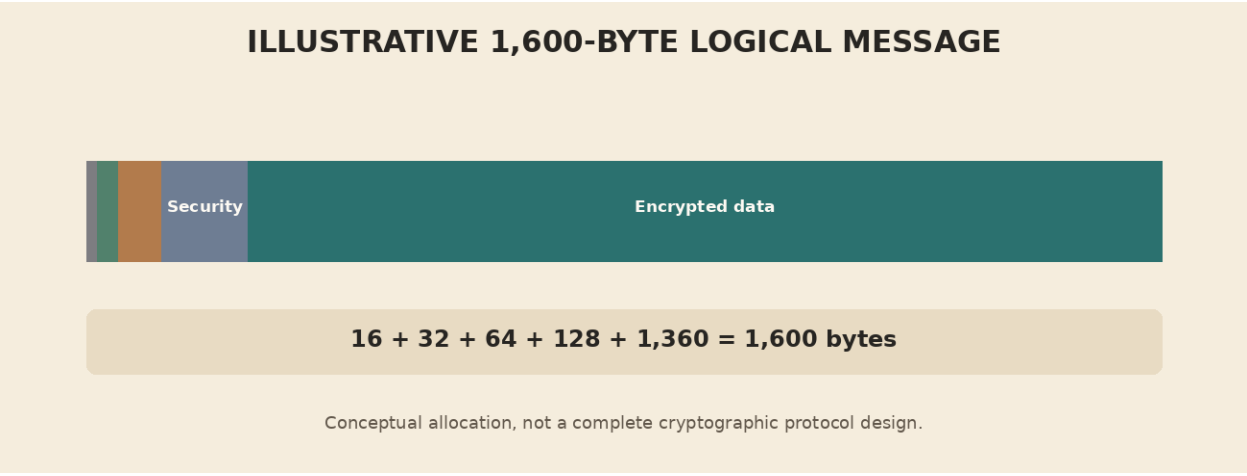


Figure 4. Illustrative breakdown of a single 1,600-byte logical message.

Network framing caveat: a logical message can exceed the payload accepted by one transport frame. At a conservative Internet datagram budget, it may need fragmentation. SMS is far more constrained (140 bytes raw capacity per single segment, with further segmentation overhead). MQTT packet-size limits depend on endpoint and broker settings.

The program sizes below were rough design ranges discussed in the thread, not measurements of a compiled Electrolips implementation:

Component	Illustrative size
Recognition/decoder component	3-10 KB
Basic network listener	20-100 KB
Secure relay listener	100 KB-2 MB
Multinetwork gateway	2-20 MB
Full application service	20-100+ MB

Smaller programs reduce resources and packaging overhead but do not inherently conceal traffic or software from the operators of the computers carrying them.

Multiple Information Fountains

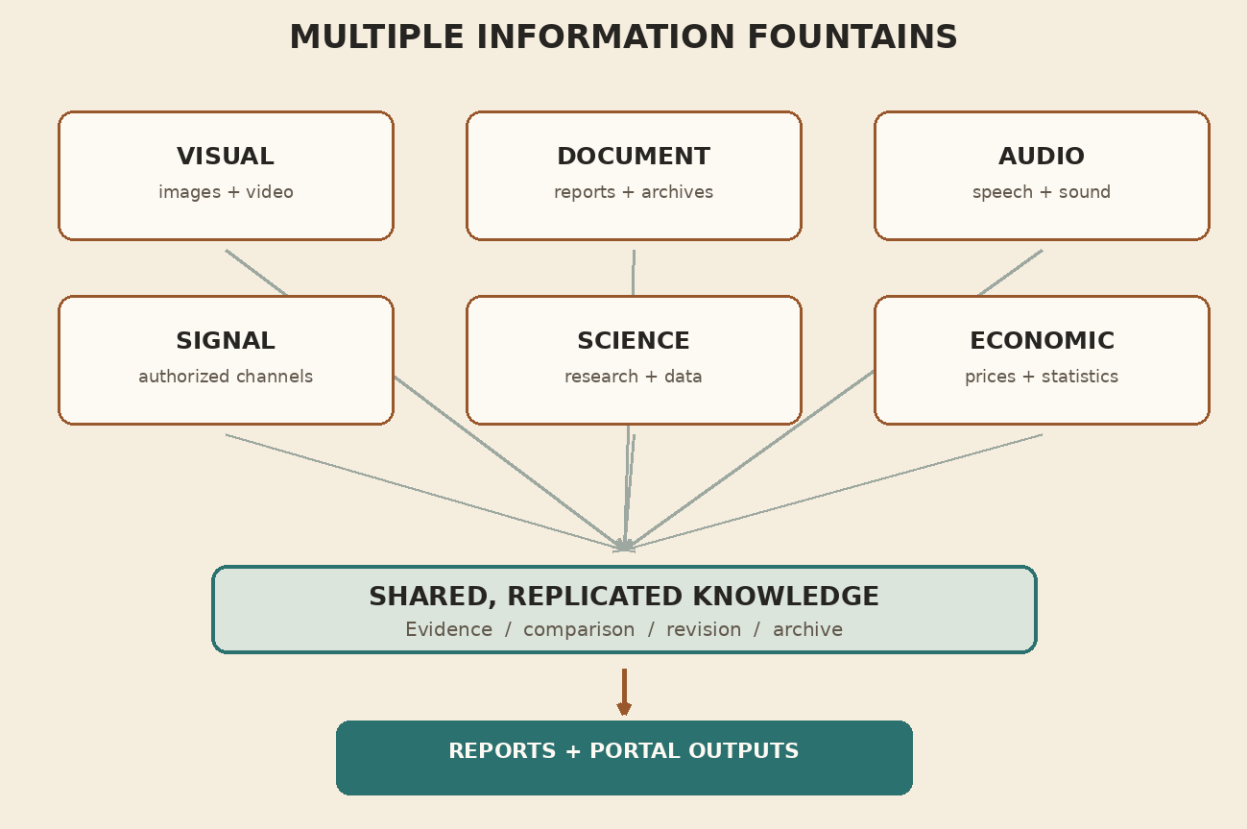


Figure 5. Independently operating fountains with shared, replicated knowledge output.

Visual fountain - Capture media or frames, analyze each image separately, distinguish observation from social interpretation.

Document fountain - Extract and assess articles, reports, books and indexed archives.

Audio fountain - Process speech and sound; preserve transcripts and source details.

Signal fountain - Work with explicitly authorized telemetry, notifications, event streams and communications.

Scientific fountain - Read experimental papers, separate observations from hypotheses and identify reproducibility issues.

Economic fountain - Collect licensed/public data on prices, trade and other indicators; measure sources and changes.

Per-image reasoning cycle

For each collected image: (1) preserve source/rights/timestamp; (2) describe literal visible content; (3) flag uncertainty; (4) analyze possible social/cultural/economic significance; (5) compare against earlier records; (6) save an individual report; (7) revisit conclusions when stronger evidence arrives. Social conclusions must remain distinct from things visible in the image.

Independent maintenance and fault tolerance

Every fountain should have its own ingress, durable queue, worker pool, checkpoint record, archival store, status monitor and restart policy. The shared knowledge index should be replicated and should not block unrelated fountains if temporarily unavailable. Maintenance can drain and replace one worker at a time; redundant workers and storage are required for service continuity.

Engineering findings and unresolved choices

Established / feasible	Still unresolved or impossible as described
Logical numbered portal names	No way to avoid all underlying network addresses
Distributed peers and gossip	No unlimited uncontrolled use of foreign router buffers
Encrypted, signed, stored messages	No guarantee of infinite successful random delivery
Transport abstraction over several gateways	SMS/voice channels do not efficiently carry full live pages
Small listener or decoder programs	Binary size does not guarantee non-detection
Independently restartable fountain services	No perpetual operation without powered, funded infrastructure

Suggested engineering sequence

1. Prototype two numbered portals and one independent cooperative relay using authenticated messages.
2. Add durable encrypted storage, delivery receipts and bounded retries.
3. Add a secondary transport gateway carrying small notification tokens.
4. Build a Visual Fountain that archives and analyzes one image at a time.
5. Add a Document Fountain and test whether it stays active while the Visual Fountain is stopped for maintenance.
6. Expand only after measuring costs, latency, privacy, reliability and provenance.

Image inventory - all visual references

This inventories every visual/diagram identified in the thread. Earlier inline SVG and UI drawings never had public image URLs. The figures in this report are newly recreated illustrations, not exact pixel-for-pixel copies of those chat visuals.

I01 Webpage A -> B session concept. Inline SVG originally drawn in chat; no external image URL; Reconstructed in figure 01; file assets/01_numbered_portals.png

I02 Four numbered portals and operator. Inline SVG originally drawn in chat; no external image URL; Reconstructed in figure 01; file assets/01_numbered_portals.png

I03 A / B dynamic relay mesh. Inline SVG originally drawn in chat; no external image URL; Reconstructed in figure 02; file assets/02_dynamic_relay_mesh.png

I04 Nym five-hop mixnet diagram. External image search query; exact earlier selected image URL not retained; Verified subject image: https://strapi-www-nym-com-production.sos-ch-dk-2.exo.io/Nym_s_Noise_Generating_Mixnet_diagram_a9c5157414.png

I05 libp2p mesh topology diagram. External image search query; exact earlier selected image URL not retained; Verified subject image: https://research.protocol.ai/blog/2020/gossipsub-an-attack-resilient-messaging-layer-protocol-for-public-blockchains/image3.png

I06 Public network pipelines and listening nodes. Inline SVG originally drawn in chat; no external image URL; Conceptually reconstructed in figure 02; file assets/02_dynamic_relay_mesh.png

I07 Multi-network delivery and authorized listeners. Diagram constructed in chat UI; no external image URL; Reconstructed in figure 03; file assets/03_multitransport_delivery.png

I08 16,000-byte header/payload strip. UI visualization originally displayed in chat; no external image URL; Conceptual allocation reconstructed in figure 04 for later 1,600-byte design

I09 Relative program-size horizontal bars. UI visualization originally displayed in chat; no external image URL; Sizing table and narrative preserved in recap

I10 Individual-image AI observation cycle. UI visualization originally displayed in chat; no external image URL; Flow preserved in chapters 17-18

I11 Six concurrent Information Fountains. UI visualization originally displayed in chat; no external image URL; Reconstructed in figure 05; file assets/05_information_fountains.png

Original citation markers - all 30

These are exact citation IDs printed in earlier responses. They are reference tokens, not website links. The adjacent source IDs refer to newly verified topical references, NOT a guarantee of identical old URLs.

Earlier citation marker	Topic / current source IDs
turn976333search0	Browser serial and WebSocket APIs / S01; S02
turn976333search7	Browser serial and WebSocket APIs / S01; S02
turn976333search5	Browser serial and WebSocket APIs / S01; S02
turn581029search1	I2P, Tor and libp2p networks / S03; S04; S06; S07
turn581029search8	I2P, Tor and libp2p networks / S03; S04; S06; S07
turn581029search11	I2P, Tor and libp2p networks / S03; S04; S06; S07
turn581029search0	I2P, Tor and libp2p networks / S03; S04; S06; S07
turn581029search10	I2P, Tor and libp2p networks / S03; S04; S06; S07
turn204228search0	Nym mixing and mobile-agent research / S08; S09 (mobile-agent citation unresolved)
turn204228search2	Nym mixing and mobile-agent research / S08; S09 (mobile-agent citation unresolved)
turn204228search9	Nym mixing and mobile-agent research / S08; S09

	(mobile-agent citation unresolved)
turn204228search6	Nym mixing and mobile-agent research / S08; S09 (mobile-agent citation unresolved)
turn491365search1	Gossipsub and peer broadcast research / S03; S17 (specific broadcast paper unresolved)
turn491365search2	Gossipsub and peer broadcast research / S03; S17 (specific broadcast paper unresolved)
turn728881search0	Gossipsub and peer broadcast research / S03; S17 (specific broadcast paper unresolved)
turn148560search2	DTN, packet observation and hop limits / S11; S14 (sniffing reference unresolved)
turn148560search8	DTN, packet observation and hop limits / S11; S14 (sniffing reference unresolved)
turn148560search0	DTN, packet observation and hop limits / S11; S14 (sniffing reference unresolved)
turn691278search2	DTN, packet observation and hop limits / S11; S14 (sniffing reference unresolved)
turn440262search1	Gossipsub, DHT, Nostr and DTN / S03; S04; S10; S11
turn440262search0	Gossipsub, DHT, Nostr and DTN / S03; S04; S10; S11
turn440262search4	Gossipsub, DHT, Nostr and DTN / S03; S04; S10; S11
turn510533search0	Gossipsub, DHT, Nostr and DTN / S03; S04; S10; S11
turn388887search5	MQTT, telephone signaling and DTN / S12; S11 (telephone-signaling reference unresolved)
turn388887search0	MQTT, telephone signaling and DTN / S12; S11 (telephone-signaling reference unresolved)
turn388887search2	MQTT, telephone signaling and DTN / S12; S11 (telephone-signaling reference unresolved)
turn991540search5	MQTT, QUIC and SMS message sizing / S12; S13; S15
turn991540search4	MQTT, QUIC and SMS message sizing / S12; S13; S15
turn991540search0	MQTT, QUIC and SMS message sizing / S12; S13; S15
turn598649search1	wolfMQTT library size / S16

Verified subject sources and image URLs

The following are current, independently verified topical links. Links are clickable in the DOCX and PDF.

S01 Web Serial device interface - Browser serial APIs access authorized local/emulated serial devices. OPEN SOURCE

S02 WebSocket bidirectional connections - Browser/server interactive connections. OPEN SOURCE

S03 libp2p Gossipsub specification - Randomized publish/subscribe mesh and gossip. OPEN SOURCE

S04 libp2p Kademlia DHT - Distributed peer and content discovery. OPEN SOURCE

S05 libp2p relay protocol - Relaying connections when direct connections are unavailable. OPEN SOURCE

S06 I2P tunnel implementation - Short-lived tunnels, fixed-size encrypted tunnel messages. OPEN SOURCE

S07 Tor path selection - Circuit reuse and renewal; caveats about existing streams. OPEN SOURCE

S08 Nym packet mixing - Random mixing delays and reordered packets. OPEN SOURCE

S09 Nym traffic flow - Layered routing and destination gateways. OPEN SOURCE

S10 Nostr NIP-01 - WebSocket relays, event publication and subscriptions. OPEN SOURCE

S11 IETF Bundle Protocol Version 7 - Store-and-forward bundles, lifetimes and optional hop limits. OPEN SOURCE

S12 MQTT v5.0 OASIS standard - MQTT session states and maximum packet-size controls. OPEN SOURCE

S13 QUIC RFC 9000 - Path datagram sizing and the 1,200-byte minimum. OPEN SOURCE

S14 Internet Protocol RFC 791 - TTL limits packet lifetime; routers are not unlimited circulating buffers. OPEN SOURCE

S15 Twilio SMS character lengths - 140-byte SMS payload and segmented text behavior. OPEN SOURCE

S16 wolfMQTT small footprint - A very small MQTT client library example; not a complete secure roaming listener. OPEN SOURCE

S17 libp2p publish-subscribe overview - Mesh peer discovery, metadata gossip and message propagation. OPEN SOURCE

Earlier two image queries: verified subject images

Earlier query 1: "Nym mixnet five hop network routing diagram gateways mix nodes". The exact prior selected image URL is not available; this is a newly verified matching image source:

Nym diagram image

Earlier query 2: "libp2p peer to peer distributed network mesh topology diagram". The exact prior selected image URL is not available; this is a newly verified matching image source:

Gossipsub mesh image

Image rights: external diagrams are linked rather than copied into this report. The five embedded illustrations were redrawn for this archive. Links may change over time.

RETURN TO CONTENTS