

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 27 September 2026

Michael S.P. Miller
SubThought Corporation
27 March 2026

The Locus URI Scheme: A 256-bit Interplanetary Addressing and Resolution Framework

draft-subthought-locus-uri-scheme-00

Abstract

This document defines the "locus" URI scheme, a hierarchical interplanetary addressing system that encodes both astronomical location and network routing information within a fixed 256-bit structure. The scheme enables deterministic resolution of named endpoints into IPv6-compatible network addresses and supports future interplanetary networking without reliance on centralized lookup systems.

The Locus scheme introduces a unified address format spanning five tiers: star system, orbital position, planetary body, network domain, and host identifier. The upper 128 bits encode astronomical location; the lower 128 bits encode IPv6-compatible network identity. For present Earth-based deployments, the scheme resolves to standard HTTPS URLs via DNS. Future extensions accommodate delay-tolerant networking across Lunar, Martian, and interstellar distances.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 27 September 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
---------------------------	---

1.1. Motivation	3
1.2. Gap in Existing Standards	4
2. Terminology	5
3. Locus Address Structure	5
3.1. Tier Definitions	6
3.2. Astronomical Mapping	6
4. Locus URI Scheme	8
4.1. Syntax	8
4.2. Authority Components	8
4.3. Examples	9
5. Encoding and Determinism	9
5.1. Named to Binary Mapping	9
5.2. IPv6 Compatibility	10
6. Resolution Model	10
6.1. Earth-based Resolution	10
6.2. Localhost Special Case	11
6.3. Non-Earth Addresses	11
7. Error Handling	11
7.1. Invalid Format	11
7.2. Unknown Endpoint	11
8. Interplanetary Extension Model	12
9. Security Considerations	12
10. IANA Considerations	13
10.1. URI Scheme Registration	13
11. Implementation Notes	13
12. Example Use Cases	14
13. References	15
13.1. Normative References	15
13.2. Informative References	15
14. Acknowledgments	16
Appendix A. Binary Encoding Example	16
Appendix B. Future Work	16
Author's Address	17

1. Introduction

1.1. Motivation

The Internet's current addressing model was designed for a single-planet deployment environment. IPv4 and IPv6 address network interfaces; DNS maps human-readable names to those addresses. Neither scheme incorporates the concept of physical location in the astronomical sense, nor do they provide a mechanism for addressing endpoints whose network identity is inseparable from their physical position in the solar system.

Several forces make a location-aware addressing scheme necessary:

- o Autonomous agency networks. Distributed artificial intelligence systems, autonomous robotic networks, and multi-agent cognitive architectures increasingly require persistent, stable endpoint identifiers that survive across deployments spanning planetary distances. An AI agency operating on Earth, the Moon, and Mars simultaneously needs a unified address space that encodes not just "where on the network" but "where in the solar system."
- o Speed-of-light latency. Communication between Earth and the Moon carries a one-way latency of approximately 1.3 seconds. Communication between Earth and Mars carries a one-way latency of 3 to 22 minutes depending on orbital positions. Addressing schemes that assume low-latency DNS resolution are architecturally unsuitable for interplanetary deployments. A scheme that encodes location deterministically -- without requiring a round-trip to a central registry -- is a prerequisite for practical interplanetary networking.

- o Deterministic resolution. Existing URI schemes rely on DNS for name resolution, which requires connectivity to a functioning DNS infrastructure. In disconnected or high-latency interplanetary environments, a deterministic mapping from human-readable names to binary addresses -- requiring no external lookup -- is essential.
- o Unified address space. Future infrastructure deployments on the Moon, Mars, and beyond will require a coherent addressing scheme that unifies Earth-based network identities with astronomical location. Bolt-on extensions to existing schemes are insufficient; a first-principles design is needed.

1.2. Gap in Existing Standards

No existing IETF standard addresses the problem of interplanetary endpoint addressing. The following related standards are relevant but insufficient:

- o RFC 3986 [RFC3986] (URI Generic Syntax) defines the general structure of URIs but provides no mechanism for encoding physical location or astronomical context.
- o RFC 8200 [RFC8200] (IPv6) provides a 128-bit address space adequate for Earth-scale networking but contains no provisions for astronomical location tiers above the network layer.
- o RFC 4838 [RFC4838] (Delay-Tolerant Networking Architecture) addresses communication protocols for high-latency and disconnected networks, including deep space, but does not define a URI scheme or address format for interplanetary endpoint identification.
- o The Bundle Protocol (RFC 9171) [RFC9171] provides a transport mechanism for delay-tolerant networking but defines endpoint identifiers (EIDs) at a protocol level rather than providing a general-purpose URI scheme suitable for application-layer addressing.
- o Existing custom URI schemes (tel:, urn:, geo:, etc.) each address a narrow domain. The geo: URI scheme (RFC 5870) [RFC5870] encodes geographic coordinates on Earth but does not extend to astronomical bodies or network identity.

The Locus URI scheme fills this gap by providing a unified, hierarchical, 256-bit address structure that encodes both astronomical location and IPv6-compatible network identity in a single, human-readable URI form, with deterministic binary encoding requiring no external registry.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Locus: A 256-bit structured address identifying an agency endpoint in both astronomical and network space.

Agency: A networked computational entity -- a service, device, autonomous agent, or distributed system -- addressable via the Locus scheme.

Syndicate: A cluster of agencies sharing a common Locus domain.

Tier: A named segment of the Locus address representing one hierarchical dimension of location or identity.

Authority: The five-tier hierarchical identifier appearing in a Locus URI between the scheme delimiter and the path.

Orb: The planet or primary body at a given orbital position. Body code 0 always denotes the orb itself; higher body codes denote moons or satellites in order of proximity.

Star: A stellar body serving as the gravitational center of an orbital system. Sol (our Sun) is Star code 1.

3. Locus Address Structure

A Locus is a 256-bit value composed of five tiers arranged in descending order of astronomical scale:

Bits 255-192	Star	64 bits	Star system identifier
Bits 191-160	Orbit	32 bits	Orbital index from star
Bits 159-128	Body	32 bits	Planetary body or satellite
Bits 127-64	Domain	64 bits	IPv6 network prefix
Bits 63-0	Host	64 bits	IPv6 interface identifier

The upper 128 bits (Star + Orbit + Body) encode astronomical location: where in the universe the endpoint resides.

The lower 128 bits (Domain + Host) encode network identity: where on the network the endpoint is reachable. Together, Domain and Host form a complete 128-bit IPv6-compatible address.

3.1. Tier Definitions

Star (64 bits, bits 255-192)

Identifies the star system. Star codes are assigned by a registry in order of distance from Sol. Sol is assigned Star code 1. Proxima Centauri is assigned Star code 2. Star codes for unnamed or unregistered systems MAY be derived by applying a 64-bit FNV-1a hash to the system's canonical name string.

Orbit (32 bits, bits 191-160)

Identifies the orbital position around the star, assigned by distance from the star in ascending order. Within the Sol system: Mercury=1, Venus=2, Earth=3, Mars=4, Jupiter=5, Saturn=6, Uranus=7, Neptune=8.

Body (32 bits, bits 159-128)

Identifies the specific body at the orbital position. Body code 0 (Orb) always identifies the planet itself. Moons and satellites are assigned codes in ascending order of proximity to the Orb. For Earth orbit: Orb=0, Luna=1. For Mars orbit: Orb=0, Phobos=1, Deimos=2.

Domain (64 bits, bits 127-64)

Encodes the IPv6 network prefix (global routing prefix and subnet ID). In named URI form, Domain is a human-readable string (e.g., "com", "net", "ai") that resolves via standard DNS. In binary form, Domain is the 64-bit FNV-1a hash of

the lowercase Domain string.

Host (64 bits, bits 63-0)

Encodes the IPv6 interface identifier. In named URI form, Host is a human-readable string (e.g., "example") that resolves via standard DNS. In binary form, Host is the 64-bit FNV-1a hash of the lowercase Host string.

3.2. Astronomical Mapping

Orbit and Body codes are derived from physical distance, ensuring codes are stable and independently computable without a registry for known bodies. The following table documents the Sol system mapping for well-known bodies:

Distance(AU)	Orbit	Body	Name
0.39	1	0	Mercury
0.72	2	0	Venus
1.00	3	0	Earth (Orb)
1.00	3	1	Luna
1.52	4	0	Mars (Orb)
1.52	4	1	Phobos
1.52	4	2	Deimos
5.20	5	0	Jupiter (Orb)
5.20	5	1	Io
5.20	5	2	Europa
5.20	5	3	Ganymede
5.20	5	4	Callisto
9.58	6	0	Saturn (Orb)
9.58	6	1	Mimas
9.58	6	2	Enceladus
9.58	6	3	Tethys
9.58	6	4	Dione
9.58	6	5	Rhea
9.58	6	6	Titan
9.58	6	7	Iapetus
19.2	7	0	Uranus (Orb)
19.2	7	1	Miranda
19.2	7	2	Ariel
19.2	7	3	Umbriel
19.2	7	4	Titania
19.2	7	5	Oberon
30.1	8	0	Neptune (Orb)
30.1	8	1	Triton
30.1	8	2	Nereid

The ten nearest star systems to Sol, assigned Star codes in order of distance:

Star Code	Star System	Distance (ly)
1	Sol	0.00
2	Proxima Centauri	4.24
3	Alpha Centauri AB	4.37
4	Barnard's Star	5.96
5	Wolf 359	7.79
6	Lalande 21185	8.29
7	Sirius AB	8.58
8	Luyten 726-8 AB	8.73
9	Ross 154	9.68
10	Ross 248	10.32

4. Locus URI Scheme

4.1. Syntax

A Locus URI conforms to the generic URI syntax defined in RFC 3986 [RFC3986]. The scheme-specific syntax is:

```
locus-URI  = "locus://" authority path-abempty
authority  = star "." orbit "." body "." domain "." host
star       = segment
orbit      = segment
body       = segment
domain     = segment
host       = segment
segment    = 1*( unreserved / pct-encoded )
```

The authority MUST contain exactly five dot-separated components. A Locus URI with fewer or more than five authority components is malformed and MUST be rejected.

4.2. Authority Components

Component	Description
star	Star system name (e.g., "sol")
orbit	Orbital body name (e.g., "earth", "mars")
body	Specific body name (e.g., "orb", "luna")
domain	Network domain string (e.g., "com", "net")
host	Host identifier string (e.g., "example")

All components are case-insensitive. Implementations MUST normalize components to lowercase before processing.

4.3. Examples

In the following examples, "example" and "example.com" are used as placeholder host and domain values per RFC 2606 [RFC2606].

Standard Earth-based endpoint:

```
locus://sol.earth.orb.com.example/concierge
```

Resolves to:

```
https://example.com/concierge
```

Localhost special case (development):

```
locus://sol.earth.orb.local.host/concierge
```

Resolves to:

```
https://localhost/concierge
```

Future Mars deployment (currently unresolvable):

```
locus://sol.mars.orb.com.example/concierge
```

Future Luna deployment (currently unresolvable):

```
locus://sol.earth.luna.com.example/concierge
```

Multi-agency within same syndicate:

```
locus://sol.earth.orb.com.example/concierge
locus://sol.earth.orb.com.example/envoy
locus://sol.earth.orb.com.example/liaison
```

locus://sol.earth.orb.com.example/messenger

5. Encoding and Determinism

5.1. Named to Binary Mapping

Locus URIs exist in two forms: named (human-readable) and binary (machine-compact). The mapping between forms is deterministic and requires no external registry for well-known bodies.

Star, Orbit, Body:

For well-known bodies, codes are assigned by the registry defined in Section 3.2. For unknown bodies, implementations MAY derive codes by applying FNV-1a 64-bit hashing to the lowercase component string, truncated to the field width:

Star code (64-bit): FNV-1a hash of star name string
Orbit code (32-bit): FNV-1a hash of orbit name, low 32 bits
Body code (32-bit): FNV-1a hash of body name, low 32 bits

Domain, Host:

Binary encoding applies 64-bit FNV-1a hashing to the lowercase component string:

Domain code (64-bit): FNV-1a hash of domain string
Host code (64-bit): FNV-1a hash of host string

FNV-1a 64-bit hash parameters:

Prime: 1099511628211
Offset: 14695981039346656037

This encoding ensures:

- o Deterministic binary representation from named form
- o No dependency on external registries for well-known bodies
- o Graceful extension to unknown bodies via hash fallback

5.2. IPv6 Compatibility

The Domain and Host fields together form a complete 128-bit IPv6-compatible address [RFC8200]:

IPv6 Address = Domain (bits 127-64) || Host (bits 63-0)

When Domain and Host encode IPv6 network prefix and interface identifier respectively, the binary Locus address maps directly to a full IPv6 address for the endpoint.

For named URI forms, Domain and Host are resolved via standard DNS using the convention:

DNS hostname = host + "." + domain

Example: domain="com", host="example" resolves to "example.com" via standard DNS.

6. Resolution Model

6.1. Earth-based Resolution

For Locus URIs where star="sol", orbit="earth", body="orb",

resolution proceeds via standard DNS:

```
locus://sol.earth.orb.<domain>.<host>/<path>
--> https://<host>.<domain>/<path>
```

Example:

```
locus://sol.earth.orb.com.example/concierge
--> https://example.com/concierge
```

Implementations MUST support this resolution path. All other resolution paths are OPTIONAL and subject to future extension.

6.2. Localhost Special Case

The domain "local" combined with host "host" is a reserved special case denoting the local machine:

```
locus://sol.earth.orb.local.host/<path>
--> https://localhost/<path>
```

This special case MUST be recognized and resolved without DNS lookup. It is intended for development and local testing environments.

6.3. Non-Earth Addresses

Locus URIs where star != "sol", orbit != "earth", or body != "orb" (excluding the localhost special case defined in Section 6.2) cannot be resolved using current infrastructure.

Implementations encountering such addresses:

- o MUST NOT attempt DNS resolution
- o MUST raise a resolution failure condition
- o MAY log the unresolvable address for diagnostic purposes
- o MAY defer resolution pending future interplanetary routing registry support

7. Error Handling

7.1. Invalid Format

A Locus URI is malformed if:

- o The scheme is not "locus" (case-insensitive)
- o The authority does not contain exactly five dot-separated components
- o Any authority component is empty

Implementations MUST reject malformed Locus URIs and MUST NOT attempt resolution of a malformed URI.

7.2. Unknown Endpoint

A Locus URI is well-formed but unresolvable if:

- o star is not "sol"
- o orbit is not "earth" (when star is "sol")
- o body is not "orb" and the URI is not the localhost special case (Section 6.2)

Implementations encountering an unresolvable Locus URI SHOULD produce a diagnostic message identifying which tier failed resolution and why.

8. Interplanetary Extension Model

The Locus scheme is designed for forward compatibility with interplanetary deployment. Future extensions MAY include:

- o Distributed interplanetary routing registries, analogous to DNS root servers, operated at planetary facilities and synchronized via delay-tolerant protocols.
- o Integration with the Bundle Protocol [RFC9171] and Delay-Tolerant Networking [RFC4838] for store-and-forward routing across disconnected interplanetary links.
- o Non-DNS resolution mechanisms for environments where DNS infrastructure is unavailable, such as early Martian deployments.
- o Standardized star registry governance, defining the process by which new star systems receive assigned Star codes.
- o Orbital computation formalization, providing a mathematical basis for deriving Orbit and Body codes from physical ephemeris data.

Anticipated future deployment targets:

```
sol.earth.luna    -- Lunar surface or orbital installations
sol.mars.orb      -- Martian surface installations
sol.mars.phobos   -- Phobos orbital installations
sol.mars.deimos   -- Deimos orbital installations
```

9. Security Considerations

Hash-based encoding (Section 5.1) avoids dependency on a centralized registry for well-known bodies, reducing the attack surface associated with registry compromise or spoofing.

Earth-based resolution (Section 6.1) inherits the security properties of HTTPS and the DNS infrastructure. Implementations SHOULD use HTTPS exclusively for resolved endpoints and SHOULD apply standard certificate validation.

Future interplanetary routing MUST consider:

- o High latency. Round-trip times of minutes to hours make interactive authentication protocols impractical. Pre-established cryptographic credentials and offline certificate validation are RECOMMENDED for interplanetary deployments.
- o Partition tolerance. Interplanetary links may be interrupted for extended periods. Implementations MUST NOT assume continuous connectivity for non-Earth endpoints.
- o Authentication across disconnected networks. Standard online certificate revocation mechanisms (OCSP, CRL) are unsuitable for high-latency interplanetary links. Future specifications SHOULD define offline-capable revocation mechanisms appropriate for interplanetary deployments.
- o Hash collision resistance. The FNV-1a hash used for binary encoding is not cryptographically secure. Binary Locus addresses MUST NOT be used as security identifiers. Named URI

form is the authoritative representation for security purposes.

10. IANA Considerations

10.1. URI Scheme Registration

This document requests registration of the following URI scheme in the "Uniform Resource Identifier (URI) Schemes" registry maintained by IANA:

Scheme name:	locus
Status:	Provisional
URI syntax:	Defined in Section 4.1 of this document
Semantics:	Hierarchical interplanetary addressing encoding astronomical location and IPv6-compatible network identity in a 256-bit structure
Encoding:	Defined in Section 5 of this document
Applications:	Distributed agency networks, autonomous robotic systems, interplanetary computing infrastructure
Contact:	SubThought Corp. <subthought@hotmail.com>, Michael Miller <3Sim.38@gmail.com>
References:	This document

11. Implementation Notes

A reference implementation of the Locus URI scheme has been developed as part of the Premise Abstract Machine (PAM) v3.3, published by SubThought Corporation.

Reference implementation behavior:

- o Named URI parsing splits the authority on dot separators and validates the five-tier structure before any resolution is attempted.
- o Binary encoding applies FNV-1a 64-bit hashing to Domain and Host strings. Star, Orbit, and Body codes use the registry table (Section 3.2) with hash fallback for unregistered bodies.
- o Resolution failure is exception-based. A well-formed but unresolvable URI raises a typed exception identifying the failing tier, rather than returning a null or error string.
- o The localhost special case (Section 6.2) is recognized before DNS resolution is attempted, requiring no network access for development environments.
- o Binary serialization produces a 64-character hexadecimal string in the format:

```
hex(Star:64) || hex(Orbit:32) || hex(Body:32) ||  
hex(Domain:64) || hex(Host:64)
```

yielding a 256-bit address serialized as 64 hexadecimal characters.

The reference implementation is available at:

<https://github.com/subthought/premise>

12. Example Use Cases

Distributed AI agency networks:

A cognitive architecture deploying agencies across Earth, Luna, and Mars uses Locus URIs as stable endpoint identifiers. The Earth-based directory agency (Concierge) registers all agency loci. Agencies communicate using Locus URIs in message headers, allowing routing to follow physical deployment without reconfiguration of application-layer addressing.

Autonomous robotic networks:

Robotic systems on the Martian surface use Locus URIs to identify control endpoints on Earth. The addressing scheme encodes the physical distance implicitly in the tier structure, enabling routing infrastructure to select appropriate delay-tolerant transport mechanisms based on the body tier.

Hybrid DNS and interplanetary systems:

Systems operating across Earth and lunar installations use Locus URIs uniformly. Earth endpoints resolve via DNS. Lunar endpoints queue via delay-tolerant routing. The same URI format and resolution interface is used throughout, with the resolution mechanism selected by the body tier.

Deployed satellite constellations:

Low-Earth orbit and geostationary satellite networks present a class of endpoint whose physical location changes continuously but whose Locus address remains stable. A satellite identified as `locus://sol.earth.sat-a87zq.com.example/telemetry` retains its address regardless of orbital position. Ground stations resolve the Locus URI to the current contact window endpoint via an orbital routing registry, which maps satellite body identifiers to reachable ground station HTTPS endpoints based on real-time ephemeris data. This decouples application-layer addressing from the physical dynamics of orbital mechanics, allowing software systems to reference a satellite by stable Locus URI without knowledge of its current ground contact geometry.

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2606] Eastlake, D. and A. Panitz, "Reserved Top Level DNS Names", BCP 32, RFC 2606, DOI 10.17487/RFC2606, June 1999, <<https://www.rfc-editor.org/info/rfc2606>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

13.2. Informative References

- [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, DOI 10.17487/RFC4838, April 2007, <<https://www.rfc-editor.org/info/rfc4838>>.
- [RFC5870] Mayrhofer, A. and C. Spanring, "A Uniform Resource Identifier for Geographic Locations ('geo' URI)", RFC 5870, DOI 10.17487/RFC5870, June 2010, <<https://www.rfc-editor.org/info/rfc5870>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC9171] Burleigh, S., Fall, K., and E. Birrane, "Bundle Protocol Version 7", RFC 9171, DOI 10.17487/RFC9171, January 2022, <<https://www.rfc-editor.org/info/rfc9171>>.

14. Acknowledgments

The Locus URI scheme was developed by Michael S.P. Miller of SubThought Corporation as part of the Premise Abstract Machine (PAM) architecture, a platform for distributed artificial intelligence and autonomous agency deployment.

Appendix A. Binary Encoding Example

The Locus URI:

`locus://sol.earth.orb.com.example/concierge`

encodes to the following binary tiers:

StarCode	(64-bit):	0x0000000000000001	(Sol = 1)
OrbitCode	(32-bit):	0x00000003	(Earth = 3)
BodyCode	(32-bit):	0x00000000	(Orb = 0)
DomainCode	(64-bit):	FNV-1a("com")	
HostCode	(64-bit):	FNV-1a("example")	

Serialized as a 64-character hexadecimal string:

`hex(Star) || hex(Orbit) || hex(Body) || hex(Domain) || hex(Host)`

The resulting 256-bit address uniquely identifies the endpoint operating on Earth's surface in the Sol system.

Appendix B. Future Work

The following items are deferred to future documents:

- o Standardized star registry: a formal governance process for assigning Star codes to stellar systems beyond the ten nearest currently registered.
- o Orbital computation formalization: a mathematical specification for deriving Orbit and Body codes from IAU ephemeris data, ensuring interoperability across independent implementations.
- o Cross-planet DNS equivalents: protocol specifications for name

resolution services operating on Martian and Lunar installations, analogous to terrestrial DNS.

- o Integration with Delay-Tolerant Networking: a specification for using Locus URIs as Bundle Protocol endpoint identifiers, enabling store-and-forward routing across interplanetary links.
- o Locus URI scheme for non-Sol star systems: extensions to support addressing of endpoints beyond the Sol system, including governance of star registry expansion.

Author's Address

Michael S. P. Miller
SubThought Corporation
311 N. Robertson Blvd #301
Beverly Hills, California
United States of America
Email: subthought@hotmail.com, 3sim.38@gmail.com
Phone: +1 310 925 5160
URI: <https://github.com/subthought/premise>