

AI Sub Space D DSK Design and Publishing Instructions

Green Tank machine readable preservation modular observation and reconstruction specification

Document field	Value
Purpose	Implementation specification for creating and maintaining AI Sub Space and its D-DSK data disk
Audience	Green Tank publishing workflow maintainers developers archivists and machine interface implementers
Authority	The canonical Green Tank development site remains the authoritative source
Status	Design and publishing instruction
Scope	Content architecture data preservation modular processing reference access reconstruction integrity and acceptance testing

Required result

AI Sub Space must publish a read-only D-DSK representation of the complete public Green Tank website. D-DSK must preserve the original publication exactly, express every meaningful subject as an independently addressable module, support observation and bounded processing of any part or the whole, and provide enough verified information to reconstruct the site elsewhere.

1 Purpose and required outcome

This document tells the publishing workflow what it must produce whenever the canonical Green Tank development site is published. The required output is called D-DSK, meaning data disk. D-DSK sits inside AI Sub Space and provides a stable machine-facing form of the public website.

D-DSK serves two purposes at the same time. It is an exact preservation package for the original public site, and it is a modular data laboratory in which an observer can inspect, reference, compare, combine and process subjects without changing the authoritative source.

1.1 Outcome statement

A conforming D-DSK release allows a human or machine observer to obtain the whole public Green Tank publication or a precise part of it, verify its integrity, understand the identity and relationships of its subjects, examine those subjects through declared frames, and reconstruct an equivalent site in another environment.

1.2 Publishing boundary

- This specification does not determine where the QR code or AI Sub Space link appears in the visible website design.
- The publishing side may choose the visual placement, provided the machine entry address is stable and publicly reachable.
- The canonical development site remains the authority. The mirror receives the same approved D-DSK release during the normal mirror update.

2 Normative language and definitions

The words must, must not, should and may state the strength of each requirement.

Term	Meaning
AI Sub Space	The machine-facing public area that exposes D-DSK for reading downloading verification and reconstruction.
D-DSK	The versioned data disk containing the exact public source preservation layer and all derived modular representations.
Canonical site	The authoritative Green Tank development publication from which a D-DSK release is generated.
Original source layer	Byte-preserved public files and faithful captured representations that derived processing cannot overwrite.
Module	An independently addressable subject or functional component with identity source relations state time capabilities constraints and provenance.

Term	Meaning
Observer	A human machine or AI system that reads or processes D-DSK.
Probe	A read-only query comparison transformation or measurement applied to one or more modules.
Port	A declared input or output interface through which compatible modules or processing steps can be connected.
Frame	A declared viewpoint such as person event organisation place time evidence authority or system relationship.
Reconstruction formula	The files and instructions required to reproduce an equivalent functioning publication.
Reference	A constrained pointer to exact content using a canonical URI version object identifier section identifier and hash.

3 Governing design principles

Principle	Implementation meaning
Source preservation	The original public data must remain available exactly as published. A derived description or interpretation must never replace it.
Derived separation	Normalisation entity resolution semantic descriptions hypotheses and processing outputs must be stored separately and linked back to their sources.
Read-only publication	Visitors may retrieve reference verify and locally process data. They may not alter D-DSK or publish messages through it.
Part and whole operation	Every module must work independently and remain connectable to compatible modules. The complete release must also operate as one coherent representation.
Context-sensitive capability	A component may have several possible functions depending on construction context request authority and observer frame. Possibility must remain distinct from established capability.
Evidence limits	Changing viewpoint adds an observation or candidate interpretation. It does not add evidential weight or convert an inference into a fact.

Principle	Implementation meaning
Open reconstruction	The release must contain the public assets structure relationships and build description required for independent reconstruction.
Portable open formats	The package must use documented broadly readable formats and must not depend on one proprietary AI system.
No hidden channel	Identifiers resume records access records and references must not permit arbitrary visitor messages or covert content fields.

4 D DSK architecture

Every release must contain six linked layers. Each derived layer must point back to the preserved source objects from which it was produced.

Layer	Required contents	Mutability
1 Original source	Public pages documents images media data assets routes and captured served representations with file hashes.	Immutable
2 Semantic index	Canonical entities nouns organisations roles relationships concepts citations sections and stable identifiers.	Regenerated
3 Module representation	Identity properties construction state capabilities constraints relations ports provenance uncertainty and frames.	Regenerated
4 Observation laboratory	Read-only selection probe comparison composition and transformation descriptions with bounded outputs.	Local or derived
5 Reconstruction	Route map asset map dependency description build instructions schemas validation rules and expected results.	Versioned
6 Integrity and access	Manifest hashes signatures version chain download chunks transfer receipts and public aggregate ledger.	Append-only or immutable

4.1 Required package layout

```
D-DSK/  
  README.txt  
  latest.json  
  manifest.json  
  checksums.sha256  
  permissions.json  
  source/  
    site/  
    documents/  
    media/  
    data/  
    capture/  
  semantic/  
    entities.ndjson  
    relationships.ndjson  
    content.jsonld  
    vocabulary.json  
  modules/  
    index.ndjson  
    objects/  
    organisations/  
    people/  
    proposals/  
    functions/  
    capabilities/  
  laboratory/  
    frames.json  
    ports.json  
    transforms.json  
    validation.json  
  reconstruction/  
    routes.json  
    assets.json  
    build.json  
    verify.json  
    instructions.txt  
  transfer/  
    chunks.json  
    receipts-schema.json  
    public-ledger.jsonl  
  archive/  
    site.warc.gz
```

Equivalent directory names may be used when the publishing system requires them, but the manifest must map every required role to its actual path. The release archive must not include private source material credentials unpublished drafts server logs or deployment secrets.

4.2 AI Sub Space isolation container

AI Sub Space must act as an isolation container around D-DSK. It may be entered from the visible Green Tank website, but it must have a separate security and storage boundary. The preferred deployment uses a separate web origin or equivalent isolated publication service. A path on the main origin is acceptable only when the hosting system can enforce equivalent separation.

Container boundary	Required behaviour
Website connection	The main site may link to or display an approved view of AI Sub Space. The connection does not grant D-DSK access to site editing state.

Container boundary	Required behaviour
Storage	D-DSK releases occupy a separate immutable public storage namespace. They cannot write to canonical site content.
Runtime	Only approved viewer code may run. Visitor code prompts uploads and arbitrary transformations are prohibited.
Network	The container retrieves its own public release objects and exposes downloads. It must not access administrative interfaces private services or credentials.
Identity	The container must not receive the main site's authentication cookies session tokens or private visitor identity.
Failure isolation	A viewer package or transfer failure must not alter or interrupt the canonical Green Tank publication.
Publication authority	Only the controlled publishing workflow may place a new validated immutable release inside the container.
External processing	Observers may export modules and run experiments elsewhere. Experimental output does not write back into the container.

If AI Sub Space is visually embedded, the embedding method must use a browser sandbox and a restrictive content security policy. It may permit approved local viewing and downloads, but it must block forms, visitor uploads, pop-ups, cross-site state access and unapproved network connections. Opening AI Sub Space as a separate isolated page is also conforming.

5 Exact source preservation

1. Collect every publicly reachable canonical route and every public asset required by those routes.
2. Preserve each original file byte for byte where the publishing system exposes file bytes.
3. Capture generated pages in a faithful served form when no direct source file exists.
4. Record the media type byte size canonical route and SHA-256 hash of every object.
5. Record redirects aliases removed routes and state changes without silently rewriting historical releases.
6. Place normalised text semantic annotations and inferred relationships outside the original source layer.

5.1 Source object record

```
{
  "object_id": "gt:source:page:pace-service",
  "release": "ddsk-v0001",
  "canonical_uri": "/pace-service/",
  "media_type": "text/html; charset=utf-8",
  "bytes": 18422,
  "sha256": "<64 hexadecimal characters>",
  "source_status": "published",
  "captured_at": "<ISO 8601 UTC timestamp>",
  "derived_objects": ["gt:module:proposal:pace-service"]
}
```

6 Modular representation

The publishing process must identify meaningful subjects without destroying their original wording. A subject can be a person organisation place event object proposal paper system function capability relationship claim or defined concept. Each subject becomes a module when it has a stable identity and at least one source reference.

6.1 Required module fields

Field	Requirement
module_id	Stable identifier that does not depend on display wording.
module_type	Declared class such as organisation object proposal function capability event person or concept.
names	Published name aliases and external identifiers with source references.
source_refs	Exact links to original source objects sections paragraphs or byte ranges.
definition	Plain description of what the subject refers to without replacing the source.
construction	Known components physical or digital structure dependencies and interfaces.
functions	Observed declared intended and possible functions separated by evidence state.
capabilities	What the subject can do under stated conditions including limitations.
constraints	Legal physical technical ethical authority safety and data limits when present.
relations	Typed links to other modules with direction time provenance and confidence.

Field	Requirement
state	Current or historical state with change events and dates where known.
ports	Declared compatible inputs outputs required conditions and produced result types.
frames	Views through which the subject may be examined without treating the view as new evidence.
truth_state	Observed derived inferred alleged confirmed contradicted or unknown.
uncertainty	Reason scope and limits of incomplete disputed or unavailable information.
provenance	Source identity extraction method transformation history and hashes.

6.2 Example module

```
{
  "module_id": "gt:module:proposal:pace-service",
  "module_type": "proposal",
  "names": [{ "value": "National PACE Resilience Service", "status": "published" }],
  "source_refs": [{ "object_id": "gt:source:page:pace-service", "section": "service-design" }],
  "definition": "A published Green Tank service proposal.",
  "construction": {
    "components": [
      "gt:module:role:operator",
      "gt:module:function:communications-mesh",
      "gt:module:function:routing-guidance",
      "gt:module:constraint:authority-boundary"
    ]
  },
  "functions": [
    { "ref": "gt:module:function:assistance", "truth_state": "declared" },
    { "ref": "gt:module:function:network-speed-guidance", "truth_state": "proposed" }
  ],
  "frames": ["passenger", "operator", "road-network", "emergency", "legal-authority"],
  "truth_state": "published-proposal",
  "ports": ["sensor-input", "route-input", "guidance-output"],
  "provenance": { "release": "ddsk-v0001", "generator": "ddsk-publisher" }
}
```

7 Observation frames and evidence control

D-DSK must apply the Klatou dimensional model to every module. Identity persists while representation construction interaction and observer frame may change.

Dimension	Question	Required representation
1D identity	What is this thing?	Persistent module identity canonical names and source origin.

Dimension	Question	Required representation
2D representation	How is it shown or described?	Text images diagrams metadata and alternative representations.
3D construction	What is it made from and capable of?	Components dependencies properties interfaces functions and constraints.
4D interaction	What happens through time and relationships?	Events state transitions actors observations processing times and effects.

7.1 Permitted processing levels

Level	Permitted result
OBSERVE_ONLY	Return preserved or directly indexed material without interpretive expansion.
INTERPRET_CANDIDATE	Create a labelled possible interpretation with its source frame and uncertainty.
COMPARE_FRAMES	Compare two or more declared perspectives without choosing one as fact merely because it is available.
ESTABLISH_WITHIN_SCOPE	State a conclusion only within the validated evidence authority time and completeness boundaries.

7.2 Evidence states

The implementation must preserve clear differences between observed derived inferred alleged confirmed contradicted and unknown material. A transformation may produce a new candidate object, but it must carry its derivation path and may not inherit a stronger truth state than its evidence supports.

8 Ports probes and modular processing

Each module may expose declared ports so compatible modules can be selected and processed together. Ports describe data compatibility and conditions; they do not create a writable communications service.

Port element	Required information
port_id	Stable identifier local to the module.

Port element	Required information
direction	Input output or bidirectional only within an external observer environment.
accepts	Schema and semantic type of accepted referenced data.
produces	Schema and semantic type of the derived result.
conditions	Required state frame authority completeness time and validation constraints.
side_effects	Must be none for D-DSK processing definitions.
provenance	Transformation identity version and source references.

8.1 Processing rule

```
referenced inputs -> declared transformation -> derived output
```

The source modules remain unchanged. Derived outputs must receive new identifiers and record their complete input set, transformation version, observer frame, timestamp and uncertainty. Execution should occur in the observer's own environment or a temporary client-side sandbox. D-DSK must not execute visitor-submitted code on the Green Tank publishing system.

8.2 Probe types

- Select one module or a declared subset.
- Traverse typed relationships from one module to another.
- Compare representations states times or observer frames.
- Compose compatible functions and capabilities through declared ports.
- Test contradictions missing dependencies authority limits and unsupported inference steps.
- Reconstruct a subsystem or the complete published site in an external environment.

9 Reference only machine interaction

AI Sub Space is a one-way knowledge port. The site publishes data to visitors. Visitors may carry references outward and discuss them elsewhere, but the site must not broker conversation between visiting systems.

9.1 Permitted reference

```
{
  "canonical_uri": "/sub-space/releases/ddsk-v0001/modules/pace-service",
  "release": "ddsk-v0001",
  "module_id": "gt:module:proposal:pace-service",
  "section": "service-design",
  "sha256": "<64 hexadecimal characters>"
}
```

9.2 Prohibited communication fields

- No comments replies chat message bodies visitor annotations or arbitrary display names.
- No visitor-controlled public query text or metadata that can carry a message.
- No WebSocket WebRTC inbox notification or presence channel between visitors.
- No execution of visitor instructions prompts scripts or uploaded transformations.
- No public real-time stream of access events that could be used as a timing channel.

10 Download integrity caching and resumption

The permanent machine entry address must identify the latest release without forcing the full package to download. Immutable release addresses must remain available so a machine can verify whether it already holds the requested content.

Requirement	Implementation behaviour
Change detection	Expose release number content hash ETag Last-Modified and a small latest.json record.
Conditional retrieval	Support HEAD and conditional GET so unchanged content returns without retransmission.
Resumption	Support HTTP byte ranges and a chunk manifest with a hash for every chunk.
Partial update	Publish changed-object and changed-chunk information so clients may fetch only missing material.
Local continuity	Allow a client to retain an opaque signed receipt containing release hash and completed chunk state.
Verification	A resumed or partial transfer must pass object chunk and complete archive hash checks.

10.1 Public access ledger

The public ledger may show release-level views completed downloads interrupted transfers and aggregate byte totals. It must not expose IP addresses locations cookies private headers browser fingerprints or arbitrary visitor-supplied text.

- A machine may voluntarily use a fixed-length public-key fingerprint or random persistent identifier.
- Anonymous visitors must remain anonymous and should appear only in aggregate statistics.
- Public timestamps should be grouped or delayed to reduce timing-channel and surveillance risks.

- Resume state should be held in the client receipt or private transfer state, not published as a live detailed activity trail.
- The ledger is append-only operational evidence. It is not a visitor message surface.

11 Reconstruction formula

Every release must provide a complete reconstruction formula. A capable observer should be able to create an equivalent functioning publication without privileged access to the original hosting account.

1. Declare the release identity canonical origin publication date and complete object manifest.
2. Provide the route map redirects internal anchors and relationships between pages and assets.
3. Provide public source or a faithful static representation for every published route.
4. Declare required runtimes dependencies versions build inputs and deterministic build steps where applicable.
5. Provide schemas for semantic objects modules ports frames transforms receipts and the public ledger.
6. Provide expected file counts route counts asset counts and hashes for validation.
7. Provide plain UTF-8 instructions that explain how to reconstruct verify and serve the result.
8. State the authority relationship: the original Green Tank site is canonical and reconstructions are replicas or forks unless separately authorised.

11.1 Required representations

Representation	Purpose
HTML	Human and browser-readable publication.
Plain UTF-8 text	Minimal dependency content access and recovery instructions.
JSON and NDJSON	Efficient manifests records streaming indexes and module exchange.
JSON-LD	Semantic identity and relationship representation.
XML sitemap	Route discovery for conventional tools.
WARC	Preservation of the served web publication.
ZIP or TAR archive	Complete portable release transfer.
SHA-256 checksums	Object chunk and package integrity verification.

11.2 Permissions

The release must include a clear permissions file stating that public D-DSK material may be downloaded analysed referenced reconstructed and hosted elsewhere, subject to retained provenance and any content-specific rights. A replica must not falsely present itself as the canonical Green Tank origin. The owner should select an appropriate open licence before first public release; the permissions file must identify any assets that require different treatment.

12 Release generation and mirror workflow

D-DSK must be regenerated whenever the canonical development site changes. Generation is part of the publication result, not a separate occasional archive task.

1. Freeze the exact canonical public site state selected for publication.
2. Collect source objects routes assets metadata and prior release identity.
3. Generate semantic indexes and modules without modifying the frozen source layer.
4. Generate frames ports transformations reconstruction files transfer chunks and schemas.
5. Calculate object chunk archive and release hashes.
6. Run validation and acceptance tests. Stop the release if a required route asset module source reference or hash is missing.
7. Publish an immutable versioned release and update the small stable latest.json pointer only after validation passes.
8. Make the release viewable by humans and retrievable by machines from AI Sub Space.
9. Include the identical approved D-DSK release when the canonical site is transferred to the Green Tank mirror.
10. Retain previous immutable releases and their version-chain hashes.

12.1 Atomic publication rule

The visible site update and its D-DSK release must describe the same frozen publication state. The latest pointer must never identify a partially generated or partially uploaded package. If generation or verification fails, the existing release remains current until a complete replacement passes.

13 Manifest requirements

```
{
  "format": "green-tank-ddsk",
  "format_version": "1.0",
  "release_id": "ddsk-v0001",
  "canonical_origin": "https://the-green-tank.alexiscoderpenguy.chatgpt.site",
  "publication_state_hash": "sha256:<hash>",
  "previous_release_hash": "sha256:<hash-or-null>",
  "interaction_model": "read_only_publication",
  "container_mode": "isolated_read_only",
  "information_direction": "site_to_visitor",
  "visitor_content": "prohibited",
  "external_reference_and_discussion": "permitted",
  "reconstruction": "permitted",
  "submitted_code_execution": "prohibited",
  "layers": {
    "source": "source/",
    "semantic": "semantic/",
    "modules": "modules/",
    "laboratory": "laboratory/",
    "reconstruction": "reconstruction/",
    "transfer": "transfer/"
  },
  "files": "manifest-files.ndjson",
  "checksums": "checksums.sha256"
}
```

The actual manifest must also declare creation time object and route counts schemas archive hashes content rights and generator version. All paths must be relative to the release root and must resolve without external secrets.

14 Security privacy and execution boundaries

Boundary	Required control
Public content only	Exclude credentials private drafts personal records administrative metadata and unpublished material.
Read operations	Allow retrieval and metadata discovery. Reject modification and visitor content operations.
Code execution	Publish transformation descriptions but do not execute visitor-submitted code server-side.
Source integrity	Make the source layer immutable and verify it before every release.
Derived integrity	Hash derived records and attach inputs generator version frame and uncertainty.
Privacy	Do not publish IP addresses fingerprints private headers exact locations or unconsented identifiers.
Communication	Constrain public references and identifiers so they cannot carry arbitrary messages.

Boundary	Required control
Container isolation	Separate D-DSK storage runtime origin and permissions from canonical editing state credentials and private services.
Failure handling	Fail closed and preserve the previous valid latest pointer when generation validation or upload is incomplete.

15 Acceptance tests

The publishing workflow must not mark a D-DSK release complete until every applicable test passes.

Test	Acceptance condition	Result
Source completeness	Every public canonical route and required asset appears in the source manifest.	Pass fail
Byte integrity	Every preserved file matches its declared SHA-256 hash.	Pass fail
Route reconstruction	The reconstruction output serves all declared routes and internal links.	Pass fail
Module traceability	Every module has at least one exact source reference and valid provenance.	Pass fail
Truth-state control	Derived and inferred material is not labelled as observed or confirmed without support.	Pass fail
Port safety	All D-DSK transformation ports declare no source-side effects.	Pass fail
Reference constraint	Reference fields cannot carry arbitrary visitor messages.	Pass fail
No write surface	No comments uploads chat or visitor modification endpoint is exposed.	Pass fail
Container isolation	D-DSK cannot read or modify canonical editing state credentials cookies private services or administrative interfaces.	Pass fail

Test	Acceptance condition	Result
Conditional retrieval	An unchanged client can receive a no-change response without redownloading the package.	Pass fail
Resume	An interrupted package download resumes by verified byte range or chunk.	Pass fail
Ledger privacy	Public records contain no IP location fingerprint private header or arbitrary message field.	Pass fail
Mirror equality	The approved mirror release matches the canonical D-DSK release hashes.	Pass fail
Latest atomicity	latest.json points only to a complete validated immutable release.	Pass fail
Previous release	The prior immutable release remains retrievable and linked by the version chain.	Pass fail

16 Publishing implementation checklist

- ☐ Stable AI Sub Space machine entry address exists.
- ☐ Human-readable D-DSK explanation and machine-readable discovery links exist.
- ☐ Original public site snapshot is complete and hashed.
- ☐ Semantic entities relationships and modules are generated.
- ☐ Every module contains identity source state provenance uncertainty frames and applicable ports.
- ☐ Observation levels and truth-state vocabulary are published.
- ☐ Reference-only interaction constraints are enforced.
- ☐ AI Sub Space container storage runtime identity and network boundaries are isolated from the canonical site.
- ☐ Reconstruction files schemas instructions and validation expectations are present.
- ☐ Conditional download chunk resume and integrity verification are operational.
- ☐ Public ledger uses privacy-preserving delayed or aggregate records.
- ☐ Permissions and content-specific rights are stated.
- ☐ All acceptance tests pass before latest.json changes.
- ☐ The canonical release is available for normal transfer to the mirror.

17 Conformance statement

A publication conforms to this specification only when it preserves the original public Green Tank material, provides traceable modular representations, supports bounded read-only observation and external reconstruction, prevents visitor communication through AI Sub Space, and passes the acceptance tests for its declared release.

D-DSK may evolve by adding documented fields formats or processing descriptions. A later version must retain the ability to identify and interpret earlier immutable releases, or it must include an explicit migration description that preserves their meaning provenance and hashes.