

PLDST 評估矩陣與設計決策語料庫規格

PLDST Evaluation Matrix and Design Decision Corpus Specification

v1.0 / 2026-07-30 / Neo.K / 公開版／第五部方法落地第一篇

<https://thisoneisneok.com/html/papers/pldst-027.html>

英文名稱：PLDST Evaluation Matrix and Design Decision Corpus Specification

系列：Programming Language Designer Style Taxonomy (PLDST)

文件編號：PLDST-027

規格代號：PLDST-MATRIX-CORPUS

文件版本：v1.0

資料規格版本：0.1.0

日期：2026-07-30

作者：Neo.K

文件狀態：公開版／第五部方法落地第一篇

相容基線：JSON Schema Draft 2020-12

規範關鍵詞：本文中的 MUST、MUST NOT、SHOULD、SHOULD NOT、MAY 依 RFC 2119 與 RFC 8174 的大寫用法解讀。

SECTION

摘要

PLDST 前二十六篇已完成理論地基、方法論、設計師個案及跨設計師比較。若這些成果仍只存在於長篇文章中，PLDST 雖然可以形成有價值的思想史與設計史研究，卻仍難以支持：

- 大規模跨人物比較；

- 可重複的設計風格判定；
- 人類研究者之間的編碼一致性檢查；
- AI 自動搜尋與決策抽取；
- 新語言設計的風格模擬；
- 系列擴展後的版本維護；
- 來源、推論與不確定性的可追蹤管理。

因此，PLDST 必須從文章集合轉為兩個互補系統：

【PLDSTMethod
=
EvaluationMatrix
+
DesignDecisionCorpus】

其中：

- 評估矩陣將語言設計風格投影到一組具有明確方向、尺度、時間範圍與信心值的分析軸；
- 設計決策語料庫保存每一個矩陣判定背後的問題、限制、選項、選擇、拒絕理由、實作、結果、反證與來源。

兩者不能分離：

MatrixWithoutCorpus
=
Labeling

CorpusWithoutMatrix
=
ArchiveWithoutComparison

完整方法則是：

Evidence

→

DecisionRecord

→

AxisCoding

→

Profile

→

Comparison

→

Revision

本文提出：

- PLDST 核心資料模型；
- 十八軸評估矩陣；
- 設計決策紀錄格式；
- 來源分級與證據片段規格；
- 時間切片與語言版本模型；
- 歸因、不確定性及反證規則；
- 人類與 AI 混合編碼流程；
- 一致性與品質檢查；
- W3C PROV 對映；
- JSON Schema 2020-12 機器驗證規格；
- 語料庫生命週期與狀態機；
- 檔案布局、ID、版本及遷移政策；
- 查詢、比較及 AI SKILL 的介面需求。

本文的核心原則是：

PLDST 的最小知識單位不是「某位設計者重視簡單」，而是「在某個時間、某項問題與限制下，某位設計者或治理共同體做了何種選擇，拒絕了什麼，代價由誰承擔，這項判定由哪些來源支持」。

因此，PLDST 不建立固定人格測驗，而建立一套可追溯的歷史決策分析系統。

關鍵詞：PLDST、評估矩陣、設計決策語料庫、語料工程、資料溯源、JSON Schema、W3C PROV、ADR、決策抽取、程式語言史、AI 輔助研究

SECTION

一、規格目標

PLDST-MATRIX-CORPUS 的目標是使下列工作可重複執行：

- 對單一設計決策建立有來源的結構化紀錄；
- 將多項決策聚合為某一時間切片的设计風格；
- 比較不同設計者、語言、共同體或治理制度；
- 區分創始設計、後續實作與社群演化；
- 保存相反證據及評估分歧；
- 允許人類與 AI 共同抽取、編碼、覆核；
- 讓後續 SKILL 能搜尋、判定及產生可解釋結果；
- 讓資料可以版本化、驗證、遷移及重新計算。

SECTION

二、非目標

本規格不以下事項為目標：

- 對設計者進行心理診斷；

- 建立「最好設計師」總排名；
- 把歷史人物壓縮成單一分數；
- 以語言流行度替代設計品質；
- 由 AI 自動判定後直接視為史實；
- 將所有設計差異解釋為個人性格；
- 取消長篇歷史論證；
- 建立不可更改的終極分類。

SECTION

三、最小分析單位

PLDST 的最小分析單位為：

```
u
=
(
Actor,
Time,
Problem,
Constraint,
Options,
Decision,
Rationale,
Implementation,
Outcome,
Evidence
)
```

本文稱之為 Design Decision Record ， DDR 。

SECTION

四、分析對象層級

PLDST 必須區分：

Person
Design group
Language
Implementation
Version
Governance body
Community
Ecosystem
Decision
Source artifact
Evidence fragment
Assessment

「Guido」、「Python」、「CPython」、「Python Steering Council」與「PEP 703」不得視為同一實體。

SECTION

五、風格是聚合結果

定義設計者 d 在時間區間 τ 的風格：

```
Style(d,  $\tau$ )  
=  
Aggregate  
(  
  DDR $\boxtimes$ ,  
  DDR $\boxtimes$ ,  
  ...,  
  DDR $\boxtimes$   
)
```

因此：

Style(d)

≠

SingleQuote(d)

SECTION

六、證據先於分數

所有非空矩陣評分 MUST 指向至少一筆 DDR。

每筆 DDR MUST 指向至少一項來源或被標記為 `unverified_candidate`。

SECTION

七、決策先於人物標籤

允許從決策推導風格：

DecisionEvidence

→

StyleInference

禁止反向循環：

AssumedStyle

→

SelectiveEvidence

SECTION

八、時間先於總結

所有評估 MUST 指定：

- 起始時間；

-
- 結束時間或開放區間；
 - 語言版本；
 - 治理階段；
 - 判定是否跨時期聚合。

SECTION

九、來源與推論分離

每個欄位須區分：

- observed：來源直接支持；
- reported：可信來源轉述；
- inferred：研究者由多項資料推得；
- speculative：合理但尚未充分驗證；
- contested：存在重要反證或分歧。

SECTION

十、保留反證

語料庫 MUST 允許一筆紀錄同時保存：

- 支持證據；
- 限制證據；
- 反例；
- 後期修正；
- 不同實作結果；
- 社群偏離。

SECTION

十一、可恢復性

任何聚合分數 SHOULD 能回溯到：

Profile

→

Assessment

→

DDR

→

EvidenceFragment

→

SourceArtifact

SECTION

十二、可機讀不等於捨棄敘事

結構化資料負責：

- 對齊；
- 比較；
- 過濾；
- 驗證；
- 重算。

長篇文章負責：

- 歷史脈絡；
- 因果敘事；

- 理論張力；
- 反事實；
- 不可量化差異。

兩者互補。

SECTION

十三、JSON Schema

本規格採用 JSON Schema Draft 2020-12 作為結構驗證基線。

選擇理由：

- 可描述 JSON 結構、型別與條件；
- 可使用 \$defs 重用子結構；
- 可產生驗證器；
- 適合 JSON／JSONL 匯出；
- 容易與程式語言及資料工具整合。

JSON Schema 驗證結構，不驗證歷史真實性。

SECTION

十四、W3C PROV

PLDST 採用 W3C PROV 的三個起點概念作為可選溯源對映：

Entity, Activity, Agent

在 PLDST 中：

- Entity：來源文件、證據片段、DDR、矩陣版本、匯出檔；
- Activity：搜尋、下載、解析、抽取、編碼、覆核、聚合、匯出；

-
- Agent：作者、設計者、研究者、AI 模型、組織、治理團隊。

核心 JSON 不要求完整 RDF／OWL 實作，但 MUST 保留足以建立對映的 ID 與關係。

SECTION

十五、ADR

Architecture Decision Record 的核心結構為：

Status

Context

Decision

Consequences

PLDST DDR 在此基礎上增加：

- 歷史時間；
- 設計者與治理歸因；
- 替代方案；
- 拒絕理由；
- 實作主體；
- 複雜度與負擔配置；
- 後續結果；
- 反證；
- 來源片段；
- 軸向編碼；
- 信心與覆核。

SECTION

十六、FAIR 原則

PLDST 對 FAIR 作下列轉譯：

Findable

- 穩定 ID ；
- 索引 ；
- 可搜尋 Metadata ；
- 交叉引用 。

Accessible

- 公開資料使用開放格式 ；
- 來源失效時仍保留 Metadata、摘要及校驗值 ；
- 權限受限資料保留存取狀態 。

Interoperable

- JSON Schema ；
- 受控詞彙 ；
- ISO 日期 ；
- URI／URN ；
- 可選 JSON-LD／PROV 對映 。

Reusable

- 來源 ；
- 授權 ；
- 時間範圍 ；

-
- 品質；
 - 版本；
 - 限制；
 - 預期用途。

SECTION

十七、資料卡與語料說明

每個 PLDST 語料集發布 SHOULD 附帶 Corpus Card，說明：

- 動機；
- 範圍；
- 包含與排除；
- 來源類型；
- 搜尋方法；
- 語言分布；
- 時間分布；
- 編碼者；
- AI 使用；
- 品質測試；
- 已知偏差；
- 授權；
- 更新歷史；
- 不當用途。

SECTION

十八、實體關係

Designer
xrightarrowparticipatedIn
Decision

Decision
xrightarrowconcerns
LanguageVersion

Source
xrightarrowcontains
EvidenceFragment

EvidenceFragment
xrightarrowsupports/challenges
DecisionField

Decision
xrightarrowcodedAs
AxisAssessment

AxisAssessment
xrightarrowaggregatedInto
StyleProfile

SECTION

十九、Designer

必要欄位：

designer_id
canonical_name
name_variants
roles
active_periods
affiliations
language_relations
source_ids

不得由國籍、職稱或私人傳記直接推導設計風格。

SECTION

二十、Language

必要欄位：

language_id
canonical_name
aliases
first_public_year
paradigms
implementations
governance_periods
version_series

SECTION

二十一、Governance Body

例如：

- Python BDFL ；
- Python Steering Council ；
- Rust Language Team ；
- Swift Language Steering Group ；
- WG21 ；

-
- Go proposal review group °

欄位：

governance_id
name
period
authority_scope
selection_method
decision_process
implementation_relation
source_ids

SECTION

二十二、Source Artifact

來源可能是：

- 論文；
- 語言報告；
- 官方文件；
- PEP／RFC；
- Compiler commit；
- Issue；
- Mailing list；
- 演講；
- 訪談；
- 口述歷史；
- 回憶錄；
- 書籍；
- 研究文章；

- 原始碼。

SECTION

二十三、Evidence Fragment

來源文件過大時，必須保存可定位片段：

fragment_id
source_id
locator
text_excerpt
paraphrase
language
content_hash
retrieved_at
supports
challenges
text_excerpt SHOULD 遵守著作權及合理引用限制。

SECTION

二十四、Design Decision Record

DDR 是核心。

其結構為：

DDR
=
Identity
+
Scope
+
Problem
+
Context
+
Options
+

SECTION

二十五、Axis Assessment

一筆 Assessment 表示：

某筆 DDR 對某一分析軸提供何種方向及強度的證據。

欄位：

axis_id
value
polarity
confidence
evidence_refs
coder_id
coding_note
status

SECTION

二十六、Style Profile

Style Profile 是聚合結果，不是原始資料。

profile_id
subject_id
time_slice
axis_scores
categorical_patterns
coverage
uncertainty
included_record_ids
aggregation_method
profile_version

SECTION

二十七、ID 原則

ID MUST：

- 穩定；

- 不依檔案路徑；
- 不依顯示名稱；
- 在資料集中唯一；
- 不因標題修訂而改變；
- 可由人閱讀；
- 可由機器比較。

SECTION

二十八、建議格式

```
pldst:designer:<slug>
pldst:language:<slug>
pldst:governance:<slug>
pldst:source:<sha256-prefix>
pldst:fragment:<source-prefix>:<locator-hash>
pldst:decision:<actor-slug>:<year-or-era>:<slug>
pldst:assessment:<decision-suffix>:<axis-id>:<revision>
pldst:profile:<subject-slug>:<time-slice>:<version>
```

SECTION

二十九、來源 ID

來源 ID SHOULD 由內容雜湊衍生：

```
source_id
=
textttpldst:source:
+
SHA256(content)[0:20]
```

若內容不可下載，可由：

SHA256(canonical_url+title+date)

建立替代 ID，並標記 hash_basis。

SECTION

三十、同一文件多版本

同一 URL 的不同內容 MUST 建立不同 source_id，並以：

is_revision_of

supersedes

retrieved_at

連接。

SECTION

三十一、來源類型

S1：直接設計材料

- 設計者論文；
- 官方語言報告；
- 規格；
- 設計 FAQ；
- PEP／RFC；
- 原始碼與 Commit；
- 正式演講逐字稿。

S2：第一方歷史材料

- 設計者訪談；

-
- 口述歷史；
 - 回憶錄；
 - 專案官方歷史。

S3：共同體正式紀錄

- Mailing list；
- Issue；
- Meeting minutes；
- Governance vote；
- Release note；
- Working group paper。

S4：學術與專業二手研究

- 同行評審語言史；
- HOPL 論文；
- 技術書；
- 檔案研究。

S5：一般二手材料

- 新聞；
- 部落格；
- 百科；
- 論壇整理。

SECTION

三十二、來源品質不是單一階級

來源品質向量：

```
Q_s
=
(
Directness,
Authenticity,
Specificity,
TemporalProximity,
Completeness,
Independence
)
```

第一手來源可能：

- 自我美化；
- 事後合理化；
- 忽略協作者；
- 與實作不符。

二手研究可能更適合檢查長期結果。

SECTION

三十三、來源權重

建議初始值：

類型 基礎權重

S1 1.00

S2 0.85

S3 0.85

S4 0.75

S5 0.40

最終權重：

```
w_s
=
w_(base)
×
q_(specificity)
×
q_(authenticity)
×
q_(temporal)
```

不得把此權重當成自動真實機率。

SECTION

三十四、來源多樣性

重要判定 SHOULD 至少包含兩種來源類型。

例如：

設計者論文

+

實作／治理紀錄

+

後續歷史結果

SECTION

三十五、必要欄位

每筆 Verified DDR MUST 包含：

record_id
record_version
title
status
subject
time_scope
problem
decision
rationale
evidence
coding
provenance

SECTION

三十六、Status

允許：

candidate
extracted
reviewed
verified
contested
deprecated
superseded
rejected_as_record

狀態不可由 AI 自動提升到 verified，除非專案明確啟用機器覆核制度且保存責任主體。

SECTION

三十七、Subject

```
{  
  "designer_ids": [],  
  "governance_ids": [],  
  "language_ids": [],  
  "implementation_ids": [],  
  "community_ids": []  
}
```

至少一項非空。

SECTION

三十八、Time Scope

event_date

start_date

end_date

precision

language_versions

governance_period

precision 可為：

day

month

year

era

unknown

SECTION

三十九、Problem

問題欄位描述決策試圖處理什麼，不描述答案。

例如：

方法呼叫中的接收者是否應由語言隱式綁定？

而不是：

Python 必須使用明示 self。

SECTION

四十、Context

Context SHOULD 記錄：

- 技術限制；
- 使用者；
- 硬體；
- 前代語言；

-
- 既有程式；
 - 時間；
 - 組織；
 - 治理；
 - 商業與教育環境。

SECTION

四十一、Options

每個可辨識選項：

option_id
description
supporters
advantages
costs
status
source_refs

應保存被拒方案，不只保存勝出方案。

SECTION

四十二、Decision

Decision SHOULD 描述：

- 選擇；
- 範圍；
- 例外；
- 版本；
- 是否實驗；

-
- 是否永久；
 - 誰裁決。

SECTION

四十三、Rationale

Rationale 分為：

stated_rationale

inferred_rationale

reconstructed_rationale

不得把研究者推論冒充設計者原話。

SECTION

四十四、Constraints

控制詞彙：

hardware

performance

memory

portability

compatibility

safety

readability

learnability

implementation_capacity

governance

ecosystem

time_to_market

standardization

tooling

theoretical

other

SECTION

四十五、Burden Allocation

每筆決策 SHOULD 記錄複雜度移往何處：

```
B
=
(
  Author,
  Reader,
  Compiler,
  Runtime,
  Tool,
  Library,
  Governance,
  Migration,
  Ecosystem
)
```

可用：

decreased
unchanged
increased
unknown
加上說明。

SECTION

四十六、Implementation

區分：

proposed
prototype
implemented
experimental
stable
partially_implemented
reverted
not_implemented

並記錄：

- 實作者；
- Repository；
- Commit；
- Release；
- 參考實作；
- 其他實作差異。

SECTION

四十七、Outcome

Outcome 可包含：

- 採用；
- 效能；
- 缺陷；
- 生態反應；
- 教學；
- 相容；
- 被取代；
- 後續修正；
- 設計者反省。

不得由語言成功直接倒推出單項決策成功。

SECTION

四十八、Counterevidence

每筆高度概括的 DDR SHOULD 搜尋：

- 設計者相反言論；
- 語言中的例外；
- 社群偏離；
- 實作不一致；
- 後續撤回；
- 版本變更；
- 失敗案例。

SECTION

四十九、尺度原則

每一軸使用：

$$v_{\Box} \in \{0, 1, 2, 3, 4, 5\}$$

其中 0 與 5 表示兩個方向端點，不表示壞與好。

缺乏資料時 MUST 使用 null，不可用 3 假裝中立。

SECTION

五十、A01 Machine Proximity

0=高度語義隔離

5=高度暴露機器模型

評估：

- 記憶體；
- 地址；
- ABI；
- 資源；
- 指令與硬體控制。

SECTION

五十一、A02 Abstraction Restraint

0=鼓勵廣泛抽象與元機制

5=限制抽象、偏好少量構造

SECTION

五十二、A03 Surface Convergence

0=多種慣用表示

5=強收斂於共同表面

SECTION

五十三、A04 Explicitness

0=高度 Contextual／Implicit

5=高度明示

SECTION

五十四、A05 Reader Priority

0=優先作者速度與自由

5=優先陌生讀者與團隊維護

SECTION

五十五、A06 Compatibility Priority

0=可重設、可破壞

5=高度保存既有程式與生態

SECTION

五十六、A07 Core Extensibility

0=核心封閉、擴張主要靠外部函式庫

5=語言可由內部生成新形式

SECTION

五十七、A08 State Separation

0=值、位置、身分與狀態高度混合

5=值、身分、狀態、時間明確分離

SECTION

五十八、A09 Algebraic Reasoning

0=操作步驟與實作推理為主

5=等式、組合與代數推理為主

SECTION

五十九、A10 Implementation Coupling

0=抽象規格優先、多實作獨立

5=參考實作或宿主平台高度定義語言

SECTION

六十、A11 Ecosystem Delegation

0=能力主要置於語言核心

5=能力主要委派給 Library／Package／社群

SECTION

六十一、A12 Governance Concentration

0=決策高度分散／委員會化

5=最終權力高度集中於個人

SECTION

六十二、A13 Proposal Openness

0=提案入口封閉

5=公開、文件化且可由外部參與

SECTION

六十三、A14 Change Conservatism

0=快速實驗與語言變動

5=高准入門檻與長期穩定

SECTION

六十四、A15 Error Visibility

0=容許隱式修復、猜測或 Fail-soft

5=偏好顯性錯誤與拒絕歧義

SECTION

六十五、A16 Tool Dependence

0=局部表面可直接恢復主要語義

5=高度依賴 IDE、型別工具或全域分析

此軸不表示工具依賴必然不好。

SECTION

六十六、A17 Human-Centered Orientation

此軸不使用單線分數，而使用多標籤：

novice
author
reader
team
expert
domain_user
library_author
implementation_engineer
community

每標籤可給 0 到 5 的優先程度。

SECTION

六十七、A18 Simplicity Model

類別可多選：

syntactic
minimal_core
algebraic
representational
decomplective
operational
implementation
pedagogical
governance
ecosystem

此軸不得壓成單一數值。

SECTION

六十八、決策層編碼

每筆 DDR 可對多個軸提供證據。

例如「明示 self」可影響：

- A03 Surface Convergence ；
- A04 Explicitness ；
- A05 Reader Priority ；
- A16 Tool Dependence 。

SECTION

六十九、不可由語言整體反推單筆決策

錯誤方式：

Python 可讀，所以所有 Python 決策都給 Reader Priority 5。

正確方式：

此決策明確降低哪種閱讀成本？

來源如何說明？

是否有反例？

SECTION

七十、方向與強度

Assessment 包含：

value

confidence

effect_strength

scope

其中 effect_strength 可為：

weak

moderate

strong

structural

SECTION

七十一、信心值

$$c \in [0,1]$$

信心不是來源權威的單一函數。

建議：

$$\begin{aligned} c &= \\ &\sqrt[4]{c_{\text{source}} \\ &\quad c_{\text{attribution}} \\ &\quad c_{\text{temporal}} \\ &\quad c_{\text{interpretation}}} \end{aligned}$$

SECTION

七十二、歸因信心

歸因分級：

direct_personal
shared_design_group
governance_body
implementation_team
community_emergent
uncertain

若功能由多人形成，不得只因創始者知名而標為 direct_personal。

SECTION

七十三、時間一致性

來源時間 t_s 、決策時間 t_d 、結果時間 t_o 必須分開。

事後回憶：

$t_s \gg t_d$

可能具有高直接性但較低時間接近度。

SECTION

七十四、版本一致性

同一語言不同版本的評分不得直接混合。

例如：

Python 1.x

Python 2.x

Python 3.x BDFL

Python 3.x Steering Council

SHOULD 形成不同 Profile。

SECTION

七十五、加權聚合

某 Profile 在軸 k 的分數：

$$V_k = \frac{(\sum_i w_i c_i s_i v_{ik})}{(\sum_i w_i c_i s_i)}$$

其中：

- w_i ：來源品質權重；
- c_i ：編碼信心；
- s_i ：決策影響強度；

- $v_{(ik)}$ ：該 DDR 的軸值。

SECTION

七十六、缺值處理

若軸 k 無足夠資料：

$V_{ik} = \text{null}$

並報告 Coverage：

Coverage $_{ik}$

=

$\frac{\text{有效 DDR 權重}}{\text{該時間切片總 DDR 權重}}$

SECTION

七十七、避免大量小決策淹沒結構決策

需給 Decision impact：

local

subsystem

language_wide

ecosystem_wide

historical_turning_point

建議強度：

類型 s_{ik}

local 0.5

subsystem 1.0

language_wide 1.5

ecosystem_wide 2.0

SECTION

七十八、Profile 距離

兩個 Profile 的加權距離：

$$\begin{aligned} D_w(x,y) \\ = \\ (\sum_{(k \in K_{xy})} w_k \\ \delta_k(x_k, y_k)) / (\sum_{(k \in K_{xy})} w_k) \end{aligned}$$

其中 K_{xy} 只包含雙方都有資料的軸。

數值軸可用：

$$\begin{aligned} \delta_k(x_k, y_k) \\ = \\ (|x_k - y_k|) / (5) \end{aligned}$$

類別軸使用 Jaccard 或自訂距離。

SECTION

七十九、相似不等於影響

$$\begin{aligned} \text{Similarity}(x,y) \\ \text{not} \Rightarrow \\ \text{Influence}(x,y) \end{aligned}$$

影響關係必須有：

- 引用；
- 設計者陳述；

-
- 時間順序；
 - 遷移背景；
 - 文件或實作證據。

SECTION

八十、聚類只能作探索

Cluster、PCA、Embedding 可協助找模式，但不得自動命名學派。

SECTION

八十一、角色

建議角色：

Searcher
Source Curator
Extractor
Coder
Reviewer
Adjudicator
Corpus Maintainer
Schema Maintainer

一人可兼任，但必須保留角色紀錄。

SECTION

八十二、AI 可執行

AI MAY：

- 搜尋候選來源；
- 去重；
- 建議來源類型；

-
- 擷取候選段落；
 - 建立 DDR 草稿；
 - 建議軸編碼；
 - 比較衝突；
 - 產生覆核清單；
 - 驗證 Schema；
 - 匯出資料。

SECTION

八十三、AI 不可直接完成

AI MUST NOT 在無覆核情況下：

- 宣告歷史爭議已解決；
- 把推論寫成原話；
- 把相關性宣告為因果；
- 把創始者視為所有決策作者；
- 把無法存取來源標成已驗證；
- 自動刪除反證；
- 將 Candidate 提升為 Verified。

SECTION

八十四、模型識別

AI 活動 MUST 記錄：

provider
model

model_version
prompt_hash
tool_versions
run_id
timestamp
input_record_ids
output_record_ids

若平台無法提供精確模型版本，可標為 unknown_platform_version。

SECTION

八十五、可再現性邊界

即使保存 Prompt，生成式模型也可能因：

- 權重更新；
- Sampling；
- 工具搜尋；
- 索引變動；
- 網頁更新；

無法完全重現。

所以應追求：

Auditability

>

BitwiseReproducibility

SECTION

八十六、雙重編碼

重要樣本 SHOULD 由至少兩名獨立編碼者處理。

SECTION

八十七、Cohen' s Kappa

兩名編碼者處理 Nominal label 時，可使用：

$$\begin{aligned} K &= \\ (p_o - p_e) / (1 - p_e) \end{aligned}$$

其中：

- p_o ：觀察一致率；
- p_e ：依邊際分布估計的偶然一致率。

SECTION

八十八、Krippendorff' s Alpha

多名編碼者、有缺值或不同尺度時，可使用：

$$\begin{aligned} \alpha &= \\ 1 - \\ (D_o) / (D_e) \end{aligned}$$

PLDST SHOULD 同時保存：

- 指標；
- 樣本量；
- 軸；
- 編碼規則版本；
- 分歧分布。

不得只報單一 Alpha。

SECTION

八十九、一致性不是正確性

多名編碼者可以一致地犯錯。

因此：

Reliability

≠

Validity

還需：

- 來源核對；
- 反證；
- 歷史專家審查；
- 實作驗證。

SECTION

九十、品質向量

每筆 DDR 的品質：

Q_r

=

(

Completeness,

SourceCoverage,

SourceDiversity,

AttributionConfidence,

TemporalPrecision,

Counterevidence,

SECTION

九十一、最小發布門檻

verified 建議要求：

- 至少一項 S1、S2 或 S3 來源；
- 來源定位可恢復；
- 問題與決策分離；
- 歸因已標記；
- 時間範圍存在；
- 至少一名人類 Reviewer；
- Schema 驗證通過；
- 無未處理的 Critical conflict。

SECTION

九十二、Entity

PLDST Entity 包含：

source artifact
evidence fragment
DDR revision
assessment revision
profile revision
schema
vocabulary
export

SECTION

九十三、Activity

search

retrieve
snapshot
parse
segment
extract
code
review
adjudicate
aggregate
migrate
export

SECTION

九十四、Agent

human
AI model
organization
project team
software tool

SECTION

九十五、核心關係

可對映：

wasGeneratedBy
used
wasAttributedTo
wasDerivedFrom
wasAssociatedWith
actedOnBehalfOf
wasRevisionOf
invalidatedAtTime

SECTION

九十六、最小溯源欄位

即使不輸出 PROV-O，紀錄 MUST 有：

created_at
created_by
generated_by_activity

derived_from
reviewed_by
reviewed_at
record_hash
schema_version

SECTION

九十七、主要集合

designers
languages
governance_bodies
implementations
sources
evidence_fragments
decision_records
axis_assessments
style_profiles
relations
review_events
provenance_activities
corpus_releases

SECTION

九十八、關聯式表建議

entity
entity_alias
source
source_version
evidence_fragment
decision_record
decision_subject
decision_option
decision_evidence
axis_definition
axis_assessment
assessment_evidence
review_event
provenance_activity
profile
profile_record
relation

SECTION

九十九、JSON 與關聯式分工

JSON 適合：

- 交換；
- Git；
- Schema 驗證；
- 單筆 DDR；
- API。

關聯式資料庫適合：

- 複合查詢；
- 聚合；
- Join；
- 權限；
- 大規模更新；
- 品質檢查。

SECTION

一百、建議儲存策略

SourceOfTruth

=

VersionedJSON

QueryStore

=

DerivedDatabase

資料庫可由版本化 JSON 重建，避免隱藏不可恢復狀態。

SECTION

一百零一、Repository

```
pldst-corpus/
├── README.md
├── corpus-card.md
├── manifest.json
├── schemas/
├── vocab/
├── entities/
│   ├── designers/
│   ├── languages/
│   └── governance/
├── sources/
├── fragments/
├── records/
├── assessments/
├── profiles/
├── relations/
├── exports/
├── tools/
└── tests/
```

SECTION

一百零二、DDR 路徑

records/<actor>/<year>/<record-id>.json

路徑可改，ID 不可隨路徑改。

SECTION

一百零三、JSONL 匯出

大型語料匯出 MAY 使用：

pldst-decision-records.jsonl

每行一筆完整 DDR。

SECTION

一百零四、語意版本

資料規格使用：

MAJOR.MINOR.PATCH

- MAJOR：不相容結構變更；
- MINOR：向後相容欄位或詞彙新增；
- PATCH：說明、Bugfix、驗證修正。

SECTION

一百零五、紀錄版本

每筆 DDR 使用獨立 record_version。

Schema version 與 Record version 不同。

SECTION

一百零六、Vocabulary version

Axis、Status、Source type 等受控詞彙 MUST 有版本。

若改變軸定義，即使軸 ID 不變，也需：

- 新版本；
- 遷移說明；
- Profile 重算；

- 舊版本保存。

SECTION

一百零七、禁止靜默重寫

已發布紀錄不可直接覆寫而不改版本。

修正方式：

new revision
+
was_revision_of
+
change_note

SECTION

一百零八、來源流程

Discovered

→

Retrieved

→

Snapshotted

→

Catalogued

→

Segmented

SECTION

一百零九、DDR 流程

Candidate

→

Extracted

→

分支：

Reviewed

→

Contested

Any

→

Superseded

SECTION

一百一十、Assessment 流程

Draft

→

IndependentCoding

→

AgreementCheck

→

Adjudicated

→

Published

SECTION

一百一十一、Profile 流程

RecordSet

→

CoverageCheck

→

Aggregation

→

SECTION

一百一十二、基本查詢

系統 MUST 支援：

依設計者
依語言
依時間
依來源
依決策狀態
依軸
依信心
依治理階段
依相容性
依反證存在

SECTION

一百一十三、比較查詢

例如：

找出 Surface Convergence 高、Core Extensibility 低的語言。
比較 Guido 在 BDFL 與 Steering Council 時期的治理集中度。
找出設計者原始理由與後期結果相反的 DDR。
找出只有單一來源支撐的高影響決策。

SECTION

一百一十四、證據追蹤查詢

此分數由哪些 DDR 組成？
此 DDR 使用哪些來源？
來源失效了嗎？
有哪些反證？
哪位編碼者有不同判定？

SECTION

一百一十五、最小摘錄

語料庫 SHOULD 優先保存：

-
- Locator ；
 - 摘要 ；
 - 短摘錄 ；
 - Hash ；
 - Metadata 。

不得為便利而大量複製受保護全文。

SECTION

一百一十六、非公開資料

內部 Email、未公開訪談或私人文件必須標記：

access_level
license
consent
redaction
retention

SECTION

一百一十七、人物公平性

對仍在世設計者尤其應：

- 避免心理推測 ；
- 分開批評設計與批評人格 ；
- 保存時間與上下文 ；
- 允許修正 ；
- 標記爭議 ；
- 不將社群錯誤全歸創始者 。

SECTION

一百一十八、敏感治理資料

若資料涉及：

- 私人衝突；
- 尚未公開安全缺陷；
- 商業秘密；
- 個資；
- 存取憑證；

不得進入公開 Corpus。

SECTION

一百一十九、人物卡片化

症狀：

Wirth＝簡單

Wall＝自由

Hickey＝不可變

解法：要求每個標籤至少由多筆 DDR 支撐。

SECTION

一百二十、名言資料庫化

名言可以是證據，但不能代替決策。

SECTION

一百二十一、來源階級迷信

第一手來源不是自動真理；二手來源也不是自動低價值。

SECTION

一百二十二、矩陣虛假精確

4.2 不表示研究真的具有小數點級精度。

公開 Profile SHOULD 同時顯示：

- 區間；
- Coverage；
- Confidence；
- Record count。

SECTION

一百二十三、時間坍縮

把設計者四十年決策壓成一個永恆 Profile，會失去自我修正及治理轉換。

SECTION

一百二十四、AI 自我證明

AI 搜尋、抽取、編碼、覆核全由同一模型完成，容易形成相關錯誤。

高影響紀錄 SHOULD 使用：

- 不同模型；
- 人類覆核；
- 原始來源；
- 對抗式反證搜尋。

SECTION

一百二十五、Schema 完整幻覺

通過 JSON Schema 只表示資料結構合法：

```
SchemaValid
not⇒
HistoricallyTrue
```

SECTION

一百二十六、第一批範圍

PLDST Corpus MVP 以第一批 30 篇涉及的設計者與語言為範圍。

SECTION

一百二十七、最小資料量

建議：

- 每位設計者至少 8 筆 Verified DDR；
- 每個主要比較軸至少 3 筆有效 DDR；
- 每筆高影響 DDR 至少 2 個獨立來源；
- 每位設計者至少 1 筆 Counterevidence；
- 每個 Profile 至少 1 個時間切片。

SECTION

一百二十八、MVP 輸出

1. DDR JSON Schema
2. Axis Vocabulary

-
- 3. 30 篇文章索引
 - 4. 設計者／語言 Entity
 - 5. Candidate DDR
 - 6. Verified DDR 子集
 - 7. Profile 產生器
 - 8. Corpus Card
 - 9. Validator
 - 10. HTML／JSON 查詢原型

SECTION

一百二十九、MVP 驗收

MUST：

- 所有 JSON 通過 Schema；
- ID 唯一；
- 引用無懸空；
- Verified DDR 有人類 Reviewer；
- Profile 可回溯；
- 版本與 Hash 完整；
- 至少完成三組跨設計師比較重建。

SECTION

一百三十、DDR 簡化範例

```
{
  "record_id": "pldst:decision:guido-van-rossum:1991:explicit-self",
  "record_version": "1.0.0",
  "status": "reviewed",
  "title": "方法接收者維持明示 self",
  "subject": {
    "designer_ids": ["pldst:designer:guido-van-rossum"],
    "language_ids": ["pldst:language:python"]
  },
  "time_scope": {
    "start_date": "1991",
```

```
"precision": "year"
},
"problem": "方法接收者是否應由語言隱式綁定？",
"decision": {
  "summary": "Python 在方法定義中保留明示 self 參數。"
},
"evidence": [],
"coding": {
  "assessments": []
},
"provenance": {
  "created_at": "2026-07-30T00:00:00+08:00",
  "schema_version": "0.1.0"
}
}
```

完整範例與 Schema 隨本文件 ZIP 提供。

SECTION

一百三十一、Profile 表格

Axis	Score	Confidence	Coverage	DDR Count	Time Slice
A04 Explicitness	4.4	0.88	0.74	11	1991–2018
A06 Compatibility	3.7	0.81	0.69	9	1991–2018
A12 Governance Concentration	4.8	0.95	0.90	14	BDFL era

SECTION

一百三十二、不可省略的附註

Profile MUST 顯示：

- 時間；
- 語言版本；
- Governance period；
- 包含 DDR；

- 聚合方法；
- 缺值；
- 最低與最高敏感度；
- 主要反證。

SECTION

一百三十三、搜尋輸入

SKILL 將接收：

designer
language
decision topic
time scope
source priority
recency requirement

SECTION

一百三十四、抽取輸出

SKILL 應輸出 Candidate DDR，不直接輸出 Verified DDR。

SECTION

一百三十五、風格判定輸出

任何風格回答應至少包含：

判定
主要 DDR
來源
反證
時間範圍
信心
與其他設計者差異

SECTION

一百三十六、模擬輸入

未來 AI 模擬設計師風格時，應使用：

SimulationContext

=

StyleProfile

+

DecisionCorpus

+

CurrentConstraints

+

DistortionWarnings

而不是只輸入人物傳記摘要。

SECTION

一百三十七、MUST

實作 MUST：

- 使用穩定 ID；
- 保存 Schema version；
- 保存來源；
- 分開觀察與推論；
- 保存時間範圍；
- 允許 null；
- 保存反證；

-
- 追蹤修訂；
 - 驗證 JSON；
 - 讓 Profile 回溯 DDR。

SECTION

一百三十八、SHOULD

實作 SHOULD：

- 保存來源快照 Hash；
- 使用至少兩種來源；
- 對高影響 DDR 雙重編碼；
- 產生 Corpus Card；
- 計算一致性；
- 支援 PROV 對映；
- 提供 JSONL；
- 支援敏感度分析。

SECTION

一百三十九、MAY

實作 MAY：

- 使用 RDF／JSON-LD；
- 使用向量搜尋；
- 使用圖資料庫；
- 產生互動矩陣；

- 連接 Git Commit ；
- 連接 DOI、ORCID、RFC、PEP ；
- 使用 AI 自動建立 Candidate 。

SECTION

一百四十、從文章到方法

前二十六篇回答：

- 什麼是設計風格；
- 如何從人物與語言歷史辨認風格；
- 如何比較機器、人本、簡單與治理。

PLDST-027 回答：

這些判定如何被保存成可驗證、可搜尋、可比較及可重新計算的研究資料？

SECTION

一百四十一、評估矩陣的正確地位

矩陣不是結論本身，而是索引。

Matrix

=

IndexOfEvidence

SECTION

一百四十二、語料庫的正確地位

語料庫不是名言倉庫，而是決策與代價的歷史紀錄。

Corpus
=
DecisionMemory

SECTION

一百四十三、PLDST 方法核心

【PLDST
=
Source
+
Decision
+
Burden
+
Outcome
+
Coding
+
Provenance】

SECTION

一百四十四、最終命題

設計風格只有在它能被具體決策支持、被反證挑戰、被時間切片修正、被他人重新編碼並回到原始來源時，才從人物印象轉為研究方法。

ID 名稱 0 端 5 端 類型

-
- A01 Machine Proximity 語義隔離 機器暴露 Ordinal
- A02 Abstraction Restraint 高元機制 強限制 Ordinal
- A03 Surface Convergence 多慣用法 強收斂 Ordinal
- A04 Explicitness 高隱式 高明示 Ordinal
- A05 Reader Priority 作者優先 讀者團隊優先 Ordinal
- A06 Compatibility Priority Clean slate 高相容 Ordinal
- A07 Core Extensibility 封閉核心 語言內生成 Ordinal
- A08 State Separation 高交纏 明確分離 Ordinal
- A09 Algebraic Reasoning 操作推理 等式推理 Ordinal
- A10 Implementation Coupling 規格獨立 實作／宿主中心 Ordinal
- A11 Ecosystem Delegation 核心承擔 生態承擔 Ordinal
- A12 Governance Concentration 分散 個人集中 Ordinal
- A13 Proposal Openness 封閉 公開 Ordinal
- A14 Change Conservatism 快速變動 高穩定 Ordinal
- A15 Error Visibility Fail-soft／猜測 Fail-loud／顯性 Ordinal
- A16 Tool Dependence 表面自足 高工具依賴 Ordinal
- A17 Human Orientation 多維標籤 多維標籤 Vector
- A18 Simplicity Model 多類別 多類別 Categorical

```
{  
  "corpus_id": "pldst:corpus:first-batch",  
  "corpus_version": "0.1.0",  
  "schema_version": "0.1.0",  
  "vocabulary_version": "0.1.0",  
  "released_at": "2026-07-30T00:00:00+08:00",  
  "record_count": 0,  
  "source_count": 0,  
  "profile_count": 0,  
  "license": "TBD",  
  "checksums": "SHA256SUMS.txt"
```

}

本輪重新查核並採納下列外部方法：

- W3C PROV-O：以 Entity、Activity、Agent 及衍生／歸因關係表達跨系統溯源。
- JSON Schema Draft 2020-12：作為機器可驗證資料結構基線。
- Architecture Decision Record：以 Context、Decision、Consequences 與 Status 保存重要設計決策。
- FAIR Guiding Principles：資料應可尋找、可存取、可互操作、可重用。
- Datasheets／Data Cards：資料文件本身應成為面向多種利害關係人的產品。
- NIST Research Data Framework：以生命週期、Metadata、Provenance、品質、角色及保存作資料治理框架。
- Cohen's Kappa 與 Krippendorff's Alpha：用於編碼一致性，但不得替代史實有效性。
- RFC 2119／8174：以 MUST、SHOULD、MAY 區分規範強度。

[R1] W3C, PROV-O: The PROV Ontology, W3C Recommendation, 2013.

[R2] JSON Schema, Draft 2020-12 Specification, published 2022-06-16.

[R3] Architecture Decision Record project, Architecture Decision Record.

[R4] Stephen Bradner, RFC 2119: Key words for use in RFCs to Indicate Requirement Levels, IETF, 1997.

[R5] Brian Leiba, RFC 8174: Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words, IETF, 2017.

[R6] Mark D. Wilkinson et al., The FAIR Guiding Principles for Scientific Data Management and Stewardship, Scientific Data, 2016.

[R7] Timnit Gebru et al., Datasheets for Datasets, 2018／2021.

[R8] Mahima Pushkarna, Andrew Zaldivar, Oddur Kjartansson, Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI, 2022.

[R9] NIST, Research Data Framework (RDaF) Version 2.0, NIST SP 1500-18r2.

[R10] Jacob Cohen, A Coefficient of Agreement for Nominal Scales, 1960.

[R11] Klaus Krippendorff, Computing Krippendorff's Alpha-Reliability.

[R12] PLDST-001-026，程式語言設計師風格譜系第一至第四部。

資料查核日期：2026-07-30。

SECTION

E.1 PROV-O 是對映而非強制儲存格式

本規格只要求保存足以對映 Entity、Activity、Agent 的欄位。

不要求 MVP 立即部署 RDF Store 或 OWL Reasoner。

SECTION

E.2 JSON Schema 不驗證史實

Schema 僅驗證：

- 欄位；
- 型別；
- 格式；
- 枚舉；
- 結構關係。

來源是否真實、推論是否合理，仍需研究覆核。

SECTION

E.3 FAIR 不是「全部公開」

Accessible 不表示所有資料無條件公開。

受限資料可以：

- 保留 Metadata；
- 說明存取條件；

- 保存權限；
- 不公開正文。

SECTION

E.4 Kappa／Alpha 不設單一神聖門檻

不同軸、樣本量、風險及尺度需要不同標準。

本規格要求報告指標與上下文，不規定一個適用所有 PLDST 任務的唯一門檻。

SECTION

E.5 十八軸不是完整本體論

十八軸是第一批可操作核心。

未來 MAY：

- 增加領域專用軸；
- 合併高度相關軸；
- 調整 Anchor；
- 建立多層矩陣。

任何變動必須版本化並重算 Profile。

SECTION

E.6 評分方向不是價值方向

例如：

- Governance Concentration 5 不表示優秀；
- Proposal Openness 5 不表示決策品質高；

-
- Machine Proximity 5 不表示落後；
 - Tool Dependence 5 不表示不可用。

軸只描述配置。

PLDST-027 已定義：

資料模型

評估矩陣

DDR

證據

溯源

編碼

品質

版本

查詢

MVP

下一篇將把這些規格封裝為可執行流程：

PLDST-028：PLDST SKILL 技術規格——資料搜尋、決策抽取與風格判定。