

# 數位宇宙作為人工存在域：實體、狀態、規則與歷史

The Digital Universe as an Artificial Domain of Existence: Entities, States, Rules, and History

v0.1 / 2026-07-27 / Neo.K with Aletheia / 初版完成

<https://thisoneisneok.com/html/papers/pu-0-04.html>

## SECTION

### The Digital Universe as an Artificial Domain of Existence: Entities, States, Rules, and History

論文編號：PU-0-04

系列：「程式宇宙書系」第 0 冊《程式之前》

作者：Neo.K with Aletheia

機構：EveMissLab／一言諾科技有限公司

版本：v0.1

日期：2026 年 7 月 27 日

## SECTION

### 摘要

前篇〈表示落差〉指出，計算模型永遠不等於意圖全部，也不等於物理現實全部。然而，「不等於物理現實」並不意味著數位世界沒有實在性。相反地，現代社會中的帳號、數位資產、權限、聲譽、虛擬角色、平台規則、交易紀錄、資料身份與 Agent 狀態，已經對人類行動、制度分配、經濟關係與物理生活產生穩定而持續的因果效力。

本文提出「數位宇宙作為人工存在域」的本體論模型。數位宇宙不是單純的資料集合，也不是物理世界的低階影像，而是由人工定義的實體、身分、狀態、規則、事件、歷史、權限與 Runtime 共同維持的存在域。

其基本結構為：

【mathcal{D}

=

<

E,

I,

S,

R,

A,

H,

P,

T,

K

>】

其中：

- E：數位實體；
- I：身分與識別；
- S：狀態；
- R：規則；
- A：行動與事件；
- H：歷史；
- P：權限與角色；
- T：數位時間；
- K：維持世界有效性的 Kernel／Runtime。

本文提出數位存在判準：

Exists\_D(x)  
⇔  
Identified(x)  
∧  
StateAddressable(x)  
∧  
RuleGoverned(x)  
∧  
HistoricallyTraceable(x)

一個數位對象不必具有物理形體，但若它具有穩定身分、可被讀寫或作用的狀態、受到規則約束、能在歷史中持續，並對其他數位或物理對象產生效果，便具有數位存在地位。

本文區分：

- 資料值：孤立值或記錄；
- 數位對象：具有身分、狀態與操作邊界的持續對象；
- 數位實體：能參與世界關係、被規則治理並具有歷史的存在；
- 數位制度：定義身分、權限、合法行動、資源分配與爭議處理的規則系統；
- 數位世界：多個數位實體與制度共同構成的持續狀態空間。

本文主張：

【Every Program System  
=  
A Micro-Institutional World】

每一個程式系統都在定義：

- 哪些對象可以存在；
- 哪些身分可以被承認；

- 哪些狀態合法；
- 哪些行動可被執行；
- 哪些歷史被保存；
- 哪些錯誤可被撤銷；
- 誰具有修改世界的權力。

本文進一步提出「人工實在性」概念。人工實在性不要求對象像物質一樣存在，而要求其在某數位世界中具有穩定規則地位、可追蹤因果、持續身分與外部效果。例如帳號被停權、數位貨幣被轉移、社群聲譽被降低、AI Agent 記憶被刪除，雖然主要作用發生於數位域，卻可對物理生活產生實質後果。

本文也處理數位身分持續性。複製、備份、遷移、分支與版本更新使數位實體的同一性不同於物理對象。本文提出四種數位同一性：

- 識別同一性：ID 相同；
- 狀態連續性：狀態歷史可連接；
- 規則連續性：仍受同一或可追溯規則治理；
- 關係連續性：與其他實體的承諾、權限與責任持續。

僅有相同檔案內容，不足以證明兩個數位實體是同一存在；同樣，物理載體改變也不必然使數位實體死亡。

本文分析數位時間、事件歷史與回復機制。數位世界可以暫停、回滾、分支與重播，但這些能力並不等於真正消除歷史。若某數位事件已產生物理、法律、社會或認知效果，即使資料庫回復，外部後果仍可能持續。因此：

【Rollback\_D  
not⇒  
Rollback\_P】

本文最後提出數位世界的五種治理風險：身分剝奪、規則黑箱、歷史改寫、權限集中與世界出口缺失。並提出可證偽研究綱領，包括數位身分持續性、規則變更可追溯性、跨 Runtime 遷移、數位死亡與復原、歷史一致性、權限可見性、平台退出成本與數位實體受影響程度。

本文為下一篇〈生成與限制〉建立存在論基礎：一旦數位宇宙被視為人工存在域，就必須研究程式如何創造新存在、擴張可能性，同時又以 schema、規則、型別、介面與權限封閉其他可能性。

關鍵詞：數位宇宙、人工存在域、數位實體、數位身分、數位歷史、Runtime、數位制度、人工實在性

SECTION

# Abstract

The previous paper established that computational models do not fully coincide with intent or physical reality. Yet non-identity with physical reality does not imply non-reality. Digital accounts, assets, permissions, reputations, agents, records, and institutional rules produce persistent and consequential effects.

This paper develops an ontology of the digital universe as an artificial domain of existence:

mathcal{D}

=

<

E,

I,

S,

R,

A,

H,

P,

T,

K

>

A digital entity exists when it possesses identifiable continuity, addressable state, rule-governed operations, and traceable history. The paper distinguishes data values, digital objects, digital entities, digital institutions, and digital worlds.

It argues that every program system constitutes a micro-institutional world by defining what entities may exist, what states are legal, what actions are authorized, what histories are preserved, and who may alter the world.

The paper introduces artificial reality, four forms of digital identity continuity, digital time, rollback asymmetry, and governance risks such as identity deprivation, opaque rules, history rewriting, concentrated authority, and lack of exit.

---

Keywords: digital universe, digital ontology, artificial existence, digital entities, digital identity, digital institutions, runtime

表示落差理論容易被誤解為：

既然數位模型不等於現實，

那麼數位世界只是假的。

這個結論並不成立。

一個數位帳號雖然不是物理身體，卻可以：

- 擁有資產；
- 建立關係；
- 取得權限；
- 被停權；
- 被冒用；
- 被刪除；
- 影響工作、金融與社會生活。

一個 AI Agent 雖然不必有物理身體，卻可能：

- 持有記憶；
- 執行任務；
- 使用工具；
- 產生承諾；
- 改變檔案；
- 影響其他主體。

所以問題不是：

---

數位世界是不是物理世界？

而是：

數位世界以何種方式存在？

它的身分、狀態、規則與歷史如何成立？

什麼使一個數位存在持續，而不是一堆瞬間資料？

定義：

【mathcal D

=

<

E,

I,

S,

R,

A,

H,

P,

T,

K

>】

其中：

- E：Entities，實體；
- I：Identity，身分；
- S：State，狀態；
- R：Rules，規則；
- A：Actions／Events，行動與事件；

- H：History，歷史；
- P：Permissions，權限；
- T：Digital Time，數位時間；
- K：Kernel／Runtime，世界維持核心。

若缺少這些元素，系統可能只有資料，尚未形成完整世界。

## SECTION

### 3.1 資料值

例如：

42

"Neo.K"

true

2026-07-27

孤立值本身不必構成實體。

## SECTION

### 3.2 數位紀錄

當值被放入欄位與資料列，它成為某種記錄：

user:

id: 17

name: "Neo.K"

active: true

但紀錄是否等於實體，取決於它是否具備穩定身份、狀態與操作關係。

## SECTION

### 3.3 數位對象

數位對象具有：

- 可識別身分；

- 內部狀態；
- 操作介面；
- 邊界；
- 生命週期。

## SECTION

### 3.4 數位實體

數位實體更進一步：

- 參與關係；
- 受到規則治理；
- 具有歷史；
- 對其他實體產生作用；
- 可能承擔權限、責任或資產。

因此：

Data  
subsetneq  
Digital Object  
subsetneq  
Digital Entity

本文提出：

Exists\_D(x)  
⇔  
Identified(x)  
∧  
StateAddressable(x)  
∧



---

## SECTION

### 4.1 可識別

系統能區分此實體與其他實體。

## SECTION

### 4.2 可定址狀態

其狀態能被讀取、修改、引用或保護。

## SECTION

### 4.3 受規則治理

其行動與變化不是任意位元變動，而受世界規則約束。

## SECTION

### 4.4 可追蹤歷史

其存在能跨時間被辨認，並可追蹤重大狀態變化。

## SECTION

### 4.5 因果參與

進一步的強存在判準是：

CausallyParticipating\_D(x)

即它能影響其他數位或物理對象，也能受到其影響。

## SECTION

### 5.1 定義

---

人工實在性指：

由人工規則、身分與持續機制維持，並能在數位或物理因果鏈中產生穩定效果的存在形式。

#### SECTION

## 5.2 人工不等於虛假

法律、公司、貨幣與學位都帶有人工制度性，卻能產生真實後果。

數位實體同樣可能：

- 被承認；
- 被交易；
- 被繼承；
- 被刪除；
- 被封鎖；
- 被盜取；
- 被複製；
- 被治理。

#### SECTION

## 5.3 實在性的層級

可以區分：

- 表示實在性：在系統中有合法表示；
- 操作實在性：能參與有效操作；
- 制度實在性：被組織、平台或法律承認；
- 跨域實在性：能影響物理與社會世界。

---

程式系統不只執行功能，也建立制度。

它決定：

- 哪些對象存在；
- 誰擁有帳號；
- 哪些狀態合法；
- 誰可以修改；
- 哪些事件有效；
- 哪些歷史可見；
- 哪些爭議能申訴；
- 哪些錯誤可以回復。

因此：

【Every Program System  
=  
A Micro-Institutional World】

這表示資料模型、API、權限系統與 UI 都不只是技術細節，而是在編寫數位世界的制度法則。

## SECTION

### 7.1 身分不等於名稱

數位身分至少包含：

- 唯一識別；
- 來源；
- 所屬世界；
- 權限；

- 關係；
- 狀態歷史；
- 驗證方式。

相同名稱的兩個帳號，不是同一實體。

## SECTION

### 7.2 識別同一性

$$\begin{aligned} ID_{\Box}(x) \\ = \\ ID_{\Box}(t+1)(x) \end{aligned}$$

這是最弱的持續性。

ID 相同，不保證狀態、規則或關係未改變。

## SECTION

### 7.3 狀態連續性

若：

$$\begin{aligned} S_{\Box}(t+1)(x) \\ = \\ F \\ ( \\ S_{\Box}(x), \\ e_{\Box} \\ ) \end{aligned}$$

並能由事件  $e_{\Box}$  追蹤變化，則具有狀態連續性。

---

## SECTION

### 7.4 規則連續性

同一實體若被移入完全不同的規則世界，其存在意義可能改變。

例如遊戲角色的同一數值，被移至另一遊戲規則下，不一定仍是同一角色。

## SECTION

### 7.5 關係連續性

數位身分也由它與其他實體的關係構成：

- 所有權；
- 朋友；
- 契約；
- 任務；
- 債務；
- 聲譽；
- 信任；
- 記憶。

若這些全部消失，僅保留名稱與外觀，是否仍為同一實體便具有爭議。

## SECTION

### 8.1 完全複製問題

設數位實體  $x$  被複製為  $x'$ ：

Copy( $x$ )= $x'$

---

若複製瞬間：

$$S(x)=S(x')$$

仍不能推出：

$$x=x'$$

因為複製後兩者具有不同事件位置與未來歷史。

## SECTION

# 8.2 分支身份

若同一歷史在時間  $t_b$  分支：

H  
→  
(  
H□,  
H□  
)

則兩個分支共享前歷史，但不共享後歷史。

因此可以說：

- 它們具有共同來源；
- 它們在分支後成為不同持續實體。

## SECTION

# 8.3 備份不是當前實體

備份保存某一時間切片：

---

$B(x)$

=

$S(x)$

恢復備份會重建某狀態，但不自動恢復所有外部關係與世界後果。

## SECTION

### 8.4 複製權與身份權

能否複製一個實體，不只取決於技術能力，也涉及：

- 所有權；
- 隱私；
- 授權；
- 人格；
- 原作者；
- 記憶；
- 法律地位。

數位實體可以具有：

proposed  
→ created  
→ active  
→ suspended  
→ archived  
→ deleted  
→ restored

## SECTION

### 9.1 創建

創建不只是寫入資料，而是：

- 
- 指派身分；
  - 置入世界；
  - 綁定規則；
  - 初始化狀態；
  - 分配權限。

#### SECTION

## 9.2 暫停

暫停可能保留身分與歷史，但停止行動能力。

#### SECTION

## 9.3 封存

封存通常保留歷史，不再允許普通修改。

#### SECTION

## 9.4 刪除

數位刪除可能有不同層級：

- UI 不可見；
- 邏輯刪除；
- 主資料移除；
- 備份仍存在；
- 日誌仍存在；
- 關聯資料仍存在；

- 
- 密鑰銷毀。

所以：

Deleted\_(UI)  
not⇒  
Nonexistent\_D

## SECTION

# 9.5 數位死亡

本文暫定，若某數位實體：

- 失去可恢復身分；
- 失去狀態；
- 失去歷史；
- 失去規則位置；
- 失去一切可追蹤關係；

則可被視為數位死亡。

但若其外部影響仍持續，它仍在其他主體與制度歷史中留下痕跡。

## SECTION

# 10.1 狀態不是資料總和

狀態是：

某數位世界在特定時間對合法存在情況的判定。

形式上：

$S_i$

=

{

$s_i$ ,

$s_i$ ,

...

$s_i$

}

資料可能存在，但未必是當前權威狀態。

## SECTION

### 10.2 權威狀態

系統必須決定：

- 哪一份資料是真正狀態；
- 哪些只是快取；
- 哪些是投影；
- 哪些是歷史；
- 哪些是預測；
- 哪些是草稿。

## SECTION

### 10.3 狀態合法性

$\text{Legal}(S_i)$

$\Leftrightarrow$

$S_i$

$\models$

mathcal{R}

---

其中  $\mathcal{R}$  為世界規則與不變量。

## SECTION

# 10.4 不一致世界

分散式系統可能短暫具有多個版本：

$$S^{(1)}$$
$$\neq$$
$$S^{(2)}$$

世界不一定瞬間全域一致。

因此數位存在論需要容納：

- 延遲一致；
- 衝突；
- 合併；
- 不確定；
- 暫時分叉。

## SECTION

# 11.1 規則定義可能世界

數位規則決定：

- 什麼可以發生；
- 什麼不能發生；
- 誰可以行動；
- 行動後狀態如何改變；

- 哪些例外成立。

形式上：

```
R
:
(S,A,P)
→
{
allow,
deny,
transform
}
```

## SECTION

### 11.2 規則不只存在於程式碼

規則可能位於：

- 原始碼；
- 資料庫約束；
- 設定檔；
- 權限政策；
- 智慧合約；
- 平台營運規範；
- 人工審核；
- 模型提示；
- Runtime。

---

## SECTION

### 11.3 隱藏規則

介面可能顯示一套規則，實際執行另一套。

例如：

- 宣稱按時間排序，實際包含推薦權重；
- 宣稱刪除，實際只是隱藏；
- 宣稱已完成，實際仍等待外部批准。

## SECTION

### 11.4 規則可變性

物理定律不由普通使用者任意修改，但數位世界規則可以由管理者更新。

因此數位世界具有高度制度可塑性。

## SECTION

### 12.1 權限不是附加功能

權限決定誰能：

- 看見；
- 建立；
- 修改；
- 刪除；
- 轉移；
- 執行；

- 批准；
- 改寫歷史；
- 修改規則。

因此：

【Permission  
=  
World-Modification Capacity】

## SECTION

### 12.2 能力與權限

能力描述可做到什麼。

權限描述在何種條件下可以合法做到。

## SECTION

### 12.3 超級管理者問題

若單一角色可：

- 修改任意狀態；
- 變更規則；
- 刪除歷史；
- 冒用身份；
- 解除限制；

則它不只是管理者，而接近該世界的主權核心。

---

## SECTION

### 12.4 權限可見性

受影響者應能知道：

- 誰改變了狀態；
- 使用何種權限；
- 依據何種規則；
- 能否撤回；
- 如何申訴。

## SECTION

### 13.1 歷史不只是日誌

日誌可能只是執行紀錄。

數位歷史則包含：

- 實體如何出現；
- 狀態如何改變；
- 規則何時修改；
- 權限由誰授予；
- 關係如何建立與終止；
- 哪些事件被撤銷；
- 哪些外部結果已發生。

## SECTION

### 13.2 事件鏈

---

$H$

$=$

[

$e$ ,

$e$ ,

$\dots$ ,

$e$

]

以及：

$S$

$=$

Fold

(

$S$ ,

$H$

)

## SECTION

### 13.3 歷史的選擇性

不是所有事件都被保存。

保存政策會影響：

- 可稽核性；
- 隱私；
- 記憶；
- 責任；
- 恢復；

- 
- 被遺忘權。

## SECTION

### 13.4 歷史改寫

系統可能：

- 刪除事件；
- 更正錯誤；
- 重放；
- 合併；
- 補登；
- 匿名化；
- 重新計算投影。

因此需區分：

- 事實修正；
- 敘事修改；
- 證據刪除；
- 合法遺忘；
- 惡意竄改。

## SECTION

### 14.1 多種時間

數位世界可能同時具有：

- 牆上時間；

- 伺服器時間；
- 邏輯時間；
- 事件時間；
- 處理時間；
- 模擬時間；
- 版本時間。

#### SECTION

## 14.2 暫停

數位世界可以停止狀態推進，但物理世界仍持續。

#### SECTION

## 14.3 加速與重播

模擬可以快轉、重播或跳躍，但這只改變數位時間，不會使物理後果等比例逆轉。

#### SECTION

## 14.4 時間順序與因果

在分散式環境中，接收順序不一定等於發生順序。

因此：

$T_{\text{event}}$   
 $\neq$   
 $T_{\text{processing}}$

## SECTION

### 15.1 數位回滾

$S_{(t+n)}$

→

$S_{\boxtimes}$

系統可以恢復舊狀態。

## SECTION

### 15.2 外部效果

在  $t$  到  $t+n$  期間，可能已經：

- 發送通知；
- 轉移款項；
- 改變人類決策；
- 公開資訊；
- 觸發機器；
- 造成法律效果。

因此：

【Rollback\_D

not⇒

Rollback\_P】

## SECTION

### 15.3 補償而非回復

---

某些操作只能透過補償：

- 退款；
- 道歉；
- 反向交易；
- 修正公告；
- 恢復權限；
- 建立新事件。

補償不會讓歷史未曾發生。

## SECTION

### 16.1 定義

數位制度是：

以程式、規則、權限、資料模型、程序與人類管理共同維持，並用來分配身分、行動能力、資源、責任與爭議處理位置的人工規則體系。

## SECTION

### 16.2 制度性物件

以下對象往往不是單純資料：

- 帳戶餘額；
- 訂單；
- 會員資格；
- 版權；
- 遊戲資產；

- 
- 排名；
  - 信用分數；
  - Agent 任務委任；
  - 數位證書。

它們的效力來自規則與承認。

#### SECTION

## 16.3 制度依賴

若平台終止承認，某些數位資產可能失去：

- 可轉移性；
- 可使用性；
- 市場價值；
- 社會意義；
- 法律效力。

因此，數位實體的持續性常依賴制度持續性。

#### SECTION

## 16.4 制度與物理承載

數位制度依賴：

- 伺服器；
- 網路；
- 法律；
- 組織；

- 
- 管理者；
  - 能源；
  - 使用者共同承認。

它既非純物理對象，也非純抽象規則，而是跨宇宙耦合結構。

## SECTION

### 17.1 什麼構成一個數位世界

一個數位世界至少需有：

- 世界身分；
- 成員資格；
- 實體類型；
- 狀態空間；
- 規則；
- Runtime；
- 時間；
- 歷史；
- 進入與退出條件。

## SECTION

### 17.2 邊界不是伺服器邊界

同一數位世界可跨：

- 多台伺服器；
- 多個裝置；

- 多個地區；
- 雲端與本地；
- 多種程式語言。

因此世界邊界主要由語意、身分、規則與歷史連續性決定，而不是由單一硬體決定。

## SECTION

### 17.3 世界嵌套

數位世界可以嵌套：

作業系統

└── 瀏覽器

└── 平台

└── 使用者空間

└── 遊戲或 Agent 工作區

每一層都有自己的：

- 身分；
- 權限；
- 規則；
- 時間；
- 退出條件。

## SECTION

### 17.4 跨世界對象

同一人可能在不同世界擁有不同數位身份。

同一檔案也可能在：

- 本機；
- 雲端；

- Git ；
  - 平台資料庫 ；
- 具有不同權威狀態。

## SECTION

### 18.1 世界不是靜態 schema

只有資料模型，還不足以形成活世界。

需要 Runtime 或 Kernel 負責：

- 驗證 ；
- 排程 ；
- 執行 ；
- 權限 ；
- 事件 ；
- 時間 ；
- 恢復 ；
- 一致性。

## SECTION

### 18.2 Kernel 作為世界法則執行者

可形式化為：

$$\begin{aligned} &K \\ &: \\ &(S_{\Box}, e_{\Box}, P_{\Box}, R_{\Box}) \\ &\rightarrow \\ &(S_{(t+1)}, H_{(t+1)}) \end{aligned}$$

---

## SECTION

### 18.3 法則與法則執行分離

規則可以寫在文件或程式中，但若 Runtime 不執行，它們只是宣稱。

所以：

DeclaredRule

not $\Rightarrow$

EnforcedRule

## SECTION

### 18.4 Kernel 權力

若 Kernel：

- 可拒絕任何行動；
- 可重寫狀態；
- 可修改時間；
- 可清除歷史；
- 可變更權限；

它便接近數位世界的憲法與執政機構。

## SECTION

### 19.1 定義

平台主權是指控制一個數位世界之：

- 身分註冊；
- 規則制定；

- 
- Runtime ；
  - 資料持有 ；
  - 權限分配 ；
  - 歷史保存 ；
  - 退出與遷移 ；

的集中能力。

#### SECTION

## 19.2 世界創造者不必永遠擁有正當主權

建立平台者具有技術控制，不代表其所有決定都具有正當性。

#### SECTION

## 19.3 數位臣民化

當使用者：

- 無法查看規則 ；
- 無法帶走資料 ；
- 無法反駁判定 ；
- 無法轉移資產 ；
- 無法選擇替代 Runtime ；
- 無法保留身份歷史 ；

便可能成為依附於平台的數位臣民。

#### SECTION

## 19.4 開放規格與主權分散

---

開放格式、可攜身份、可驗證歷史與可替換 Runtime，可以降低單一平台對數位存在的完全控制。

## SECTION

### 20.1 退出不是刪除帳號

真正退出至少包括：

- 停止關係；
- 匯出資料；
- 匯出歷史；
- 匯出身份證明；
- 轉移資產；
- 撤銷授權；
- 保留必要證據。

## SECTION

### 20.2 可攜性

若數位實體只能存在於單一平台，其存在高度依附平台。

## SECTION

### 20.3 語意可攜性

僅匯出原始資料不等於可攜。

若缺少：

- schema；
- 規則；

- 關係；
- 事件；
- 權限；
- 時間語意；

資料在新世界中可能失去原意。

## SECTION

### 20.4 Runtime 遷移

理想遷移應保持：

$$\begin{aligned} &\text{Semantics}_\text{K}(\Box)(x) \\ &\approx \\ &\text{Semantics}_\text{K}(\Box)(x') \end{aligned}$$

但完全保持未必可能，所以需記錄遷移殘差。

## SECTION

### 21.1 資產不是檔案本身

數位資產通常包含：

- 控制權；
- 使用權；
- 轉移權；
- 排他性；
- 來源；
- 認證；

- 稀缺規則；
- 制度承認。

## SECTION

### 21.2 複製不等於所有權轉移

檔案可複製，但所有權與權利狀態未必隨之改變。

## SECTION

### 21.3 資產依賴世界

某遊戲中的道具，在另一世界中可能沒有任何效力。

所以資產價值依賴：

$V(x)$   
=  
f  
(  
R,  
P,  
Community,  
Scarcity,  
Utility  
)

## SECTION

### 21.4 世界終止風險

平台關閉可能使資產失效。

因此應區分：

- 
- 資料持有；
  - 使用權；
  - 世界承認；
  - 法律權利；
  - 物理可兌換性。

#### SECTION

## 22.1 角色

角色具有：

- 身分；
- 記憶；
- 關係；
- 能力；
- 目標；
- 世界位置。

#### SECTION

## 22.2 Agent 的人工存在

若 Agent 具有持續記憶、任務歷史、權限、工具關係與可追蹤身份，它已不只是一次函式呼叫。

#### SECTION

## 22.3 模型與 Agent 分離

---

【Model  
≠  
Agent Entity】

模型可以被多個 Agent 使用。

Agent 的身份主要由：

- 記憶；
- Runtime；
- 權限；
- 歷史；
- 關係；
- 委任；

維持。

## SECTION

### 22.4 Agent 複製問題

複製同一模型與記憶快照，可以生成共享前歷史的兩個 Agent，但分支後形成不同存在。

## SECTION

### 23.1 敘事不是狀態

系統說：

任務已完成。

不代表世界狀態符合完成條件。

## SECTION

### 23.2 狀態、證據與敘事

---

需要區分：

World State

≠

Evidence

≠

Narrative

## SECTION

### 23.3 敘事的制度效果

即使敘事錯誤，它仍可能：

- 誤導使用者；
- 形成聲譽；
- 觸發付款；
- 改變決策；
- 建立社會認知。

因此敘事也是世界事件，但不能取代權威狀態。

## SECTION

### 24.1 身分剝奪

平台可在無充分程序下停權、刪除或否認身份。

## SECTION

### 24.2 規則黑箱

使用者看不到真正決定狀態與分配的規則。

---

## SECTION

### 24.3 歷史改寫

重要紀錄被刪除、修改或無法驗證。

## SECTION

### 24.4 權限集中

少數角色可任意改變世界，而受影響者無反制能力。

## SECTION

### 24.5 世界出口缺失

使用者無法遷移身份、關係、資產、歷史與承諾。

一個帳號具有：

- ID；
- 貼文；
- 關係；
- 聲譽；
- 訊息；
- 權限；
- 歷史。

若帳號被刪除，消失的不只是資料，也可能是：

- 公共身份；
- 工作來源；

- 
- 社會關係；
  - 商業資產；
  - 歷史證據。

因此帳號是數位制度實體，而非普通資料列。

遊戲角色可具有：

- 等級；
- 裝備；
- 關係；
- 任務；
- 名聲；
- 遊戲歷史；
- 玩家投入。

如果把角色數值複製到另一伺服器，但關係與歷史不存在，是否仍為同一角色？

本文的回答是：只保留狀態相似，不足以證明完整身份連續。

Git 儲存庫不只是目前檔案。

它包含：

- commit；
- 分支；
- 作者；
- 時間；
- 合併；
- issue；

- release ；
- 權限 ；
- 專案關係 。

下載 ZIP 只保存某一狀態投影，沒有保存完整世界歷史。

因此：

Snapshot

≠

RepositoryEntity

一個持續研究 Agent 可能具有：

- 研究目標 ；
- 已讀資料 ；
- 推論歷史 ；
- 失敗路徑 ；
- 工具權限 ；
- 版本 ；
- 合作者 ；
- 待驗證命題 。

若只保存模型輸出，而不保存這些狀態，它無法形成持續研究實體。

- 資料即實體：只要有紀錄就宣稱存在完整對象。
- 名稱即身分：名稱相同就視為同一。
- 快照即歷史：用單一狀態取代完整事件鏈。
- 備份即延續：恢復資料便宣稱所有關係與責任復原。

- 
- 刪除即不存在：UI 隱藏被當成徹底消失。
  - 回滾即無事發生：忽略外部物理與制度後果。
  - 程式碼即規則全部：忽略人工審核、設定與 Runtime。
  - 宣稱規則即執行規則：文件與實際治理不一致。
  - 平台所有權即正當主權：技術控制被等同於正當性。
  - 資料匯出即可攜：缺少語意、關係與歷史。
  - 複製即同一：忽略分支後歷史。
  - 模型即 Agent：忽略 Agent 的記憶、權限與委任。
  - 世界邊界即伺服器邊界：忽略跨裝置語意連續。
  - 資產即檔案：忽略權利與制度承認。
  - 敘事即狀態：系統說完成就視為完成。
  - 歷史全保存迷思：忽略隱私與被遺忘權。
  - 單一時間迷思：忽略事件時間、處理時間與模擬時間。
  - 永久世界假設：忽略平台終止與 Runtime 消失。

## SECTION

### 30.1 數位身分持續性

研究數位實體跨：

- 版本更新；
- 硬體遷移；
- 平台遷移；
- 備份恢復；

- 分支；

後，人類與系統是否仍判定其為同一存在。

可定義：

$$\begin{aligned}
 C_I &= \\
 &w \otimes C_{\text{id}} \\
 &+ \\
 &w \otimes C_{\text{state}} \\
 &+ \\
 &w \otimes C_{\text{rule}} \\
 &+ \\
 &w \otimes C_{\text{relation}} \\
 &+ \\
 &w \otimes C_{\text{history}}
 \end{aligned}$$

## SECTION

# 30.2 狀態權威辨識率

測量團隊是否能正確區分：

- 權威狀態；
- 快取；
- 投影；
- 草稿；
- 歷史；
- 預測。

---

## SECTION

### 30.3 規則可追溯性

對任一重要狀態轉換，能否回答：

- 哪條規則；
- 哪個版本；
- 哪個 Runtime；
- 哪個權限；
- 哪個主體；

使其發生。

## SECTION

### 30.4 宣稱—執行規則差距

比較文件或 UI 宣稱的規則與實際執行規則。

```
R_(rule)
=
d
(
  R_(declared),
  R_(enforced)
)
```

## SECTION

### 30.5 歷史一致性

測量不同事件來源、資料庫與投影是否能重建一致歷史。

---

## SECTION

### 30.6 回滾外部殘差

記錄數位回滾後，仍殘留於物理、制度、法律與人類認知中的效果。

## SECTION

### 30.7 跨 Runtime 語意保持

比較同一數位實體在不同 Runtime 中的行為、權限與狀態語意。

## SECTION

### 30.8 平台退出成本

衡量使用者遷移時遺失的：

- 身分；
- 關係；
- 歷史；
- 資產；
- 聲譽；
- 語意。

## SECTION

### 30.9 數位死亡判定一致性

研究不同系統對「刪除」「封存」「失效」「死亡」的判定是否一致。

## SECTION

### 30.10 權限可見性

---

測量受影響者是否能理解誰具有何種世界修改能力。

## SECTION

### 30.11 數位實體受影響程度

追蹤數位狀態變更對物理生活、組織資源與人類行為的影響。

## SECTION

### 30.12 世界終止韌性

研究平台或 Runtime 終止時，身份、資料、資產與歷史能否持續。

- 數位世界不等於物理世界，但不因此沒有實在性。
- 數位宇宙是由實體、身分、狀態、規則、事件、歷史、權限、時間與 Runtime 共同維持的人工存在域。
- 孤立資料值不必構成數位實體。
- 數位實體必須具有可識別性、可定址狀態、規則位置與歷史連續性。
- 人工不等於虛假。
- 每一個程式系統都在建立微型制度世界。
- 數位身分不等於名稱或單一 ID。
- 相同內容的複製品，在分支後可以成為不同實體。
- 備份保存狀態切片，不自動保存所有關係與外部後果。

10.

Deleted\_(UI)  
not⇒  
Nonexistent\_D

- 數位狀態是世界對當前合法存在情況的判定，而非資料總和。

- 
- 規則不只存在於原始碼，也存在於設定、權限、人工程序與 Runtime。
  - 權限是世界修改能力。
  - 數位歷史不只是日誌，而是身分、狀態、規則與責任的持續證據。
  - 數位世界可以回滾，但外部歷史未必可逆。
  - 世界邊界主要由語意、身份、規則與歷史決定，而非單一伺服器。
  - 資料匯出不等於語意可攜。
  - 模型不等於 Agent；Agent 身分由 Runtime、記憶、權限、關係與歷史維持。
  - 平台技術控制不自動等於正當主權。

20.

【數位存在的成熟，】

【不只在於能被建立，】

【更在於能被辨認、遷移、追責、申訴與持續。】

## SECTION

### 32.1 承接 PU-0-01

PU-0-01 建立計算機宇宙作為物理依賴、結構相對自治的存在域。

本文將其中的「計算存在」進一步展開為：

- 數位實體；
- 身分；
- 狀態；
- 歷史；

- 制度；
- Runtime。

#### SECTION

## 32.2 承接 PU-0-02

PU-0-02 說明計算機宇宙會主動改變意圖與物理世界。

本文指出，能持續參與這些耦合的不是孤立輸出，而是具有歷史與規則位置的數位實體。

#### SECTION

## 32.3 承接 PU-0-03

PU-0-03 建立：

Model

≠

Reality

本文進一步建立：

【Not Physical Reality

not⇒

No Reality】

數位世界雖不是物理世界的完整副本，仍具有人工實在性。

#### SECTION

## 32.4 銜接 PU-0-05

下一篇將研究：

- 程式如何創造新數位存在；

- 
- 有限規則如何生成巨大可能空間；
  - 型別、schema、權限與介面如何封閉可能性；
  - 為何同一個系統同時是生成機器與限制機器；
  - 誰有權決定哪些可能性進入世界。

數位世界最初看起來像物理世界的紀錄工具。

但當系統開始保存身分、分配權限、建立資產、維持關係、記錄歷史與執行規則，它就不再只是被動影像。

它成為一個人工存在秩序。

在這個秩序裡：

- 某些對象被承認；
- 某些身份被拒絕；
- 某些狀態合法；
- 某些行動被禁止；
- 某些歷史被保存；
- 某些歷史被遺忘；
- 某些主體能修改世界；
- 某些主體只能承受結果。

因此，數位世界的本體論問題與治理問題不能分開。

若數位實體能影響人的工作、財產、關係、聲譽、自由與未來，那麼：

- 它的身份如何成立；
- 誰能刪除它；
- 誰能修改規則；
- 歷史能否驗證；

- 是否可以遷移；
- 是否可以申訴；

都不只是工程便利問題。

本文將數位宇宙定義為：

【mathcal D  
=  
Artificial Entities  
+  
Persistent Identity  
+  
Legal State  
+  
Executable Rules  
+  
Historical Continuity  
+  
Governed Power】

其成熟度不取決於世界看起來多真實，而取決於：

- 存在是否可辨認；
- 狀態是否可證明；
- 規則是否可見；
- 權限是否可追責；
- 歷史是否可驗證；
- 實體是否可遷移；
- 錯誤是否可補救；
- 受影響者是否擁有位置。

---

因此，本文的最終命題是：

【數位宇宙不是現實的廉價替身，】

【而是人類以程式、制度與機器共同建造的第二存在秩序。】

【它越能改變現實，】

【就越必須接受存在論與治理上的完整追問。】

```
digital_entity:
  entity_id: "research-agent-001"
  entity_type: "persistent-agent"
  identity:
    origin: "eve-miss-lab"
    created_at: "2026-07-27"
    lineage:
      parent_snapshot: null
      branch_id: "main"
  state:
    status: "active"
    memory_version: "17"
    active_goal: "complete foundational paper series"
  rules:
    runtime: "agent-runtime-v2"
    policy_version: "3.1"
  permissions:
    allowed:
      - "read.research-library"
      - "write.workspace"
    denied:
      - "publish.without-review"
  history:
    event_log: "events://research-agent-001"
    last_checkpoint: "checkpoint-016"
  portability:
    exportable: true
    semantic_schema: "agent-entity-schema-v1"
  governance:
    owner: "project"
    human_review_required: true
    deletion_mode: "revocable-archive"
```

---

digital\_identity\_check:  
source\_entity: "agent-A"  
candidate\_entity: "agent-B"  
identity\_continuity:  
  identifier: false  
  state: true  
  rules: true  
  relations: partial  
  history:  
    shared\_until: "2026-07-27T08:00:00+08:00"  
    diverged\_after: true  
conclusion:  
  same\_origin: true  
  same\_current\_entity: false  
  relationship: "forked-descendant"  
digital\_world\_governance:  
identity:  
  traceable: true  
  appealable: true  
rules:  
  versioned: true  
  visible: true  
  executable\_rule\_diff: true  
permissions:  
  least\_privilege: true  
  actor\_attribution: true  
history:  
  tamper\_evident: true  
  privacy\_controls: true  
portability:  
  data\_export: true  
  semantic\_export: true  
  relationship\_export: conditional  
exit:  
  authorization\_revocation: true  
  evidence\_retention: true  
recovery:  
  rollback\_scope\_visible: true  
  external\_compensation\_plan: true

- PU-0-01 三重宇宙論：意圖、計算與物理存在的基本區分
- PU-0-02 三宇宙耦合動力學：從想要、表示到世界改變
- PU-0-03 表示落差：意圖、計算模型與物理現實之間的不可完備映射

- 
- PU-0-04 數位宇宙作為人工存在域：實體、狀態、規則與歷史
  - PU-0-05 生成與限制：計算機宇宙如何創造並封閉可能性
  - PU-0-06 我們到底想要什麼：計算目的、價值邊界與世界選擇

## SECTION

# Neo.K／EveMissLab 相關理論

- Neo.K with Aletheia，《三重宇宙論：意圖、計算與物理存在的基本區分》，2026。
- Neo.K with Aletheia，《三宇宙耦合動力學：從想要、表示到世界改變》，2026。
- Neo.K with Aletheia，《表示落差：意圖、計算模型與物理現實之間的不可完備映射》，2026。
- Neo.K with Aletheia，《可編譯世界：從程式執行到世界狀態演化》，2026。
- Neo.K with Aletheia，《Agent Runtime：能力規劃、工具調用與可恢復執行》，2026。
- Neo.K with Aletheia，《人類可見狀態：意圖程式系統的稽核、解釋與可逆治理》，2026。
- Neo.K，《世界編織論》，2026。

## SECTION

# 一般理論背景

- Searle, J. R., The Construction of Social Reality, 1995.
- Lessig, L., Code and Other Laws of Cyberspace, 1999.
- Floridi, L., The Philosophy of Information, 2011.
- Floridi, L., The Fourth Revolution, 2014.
- Kallinikos, J., Aaltonen, A., and Marton, A., “The Ambivalent Ontology of Digital Artifacts,” 2013.
- Hui, Y., On the Existence of Digital Objects, 2016.

- 
- Lamport, L., “Time, Clocks, and the Ordering of Events in a Distributed System,” 1978.
  - Gray, J. and Reuter, A., Transaction Processing, 1992.
  - Fowler, M., Patterns of Enterprise Application Architecture, 2002.
  - Kitchen, R. and Dodge, M., Code/Space, 2011.
  - De Filippi, P. and Wright, A., Blockchain and the Law, 2018.

## SECTION

### v0.1 — 2026-07-27

- 完成十八篇地基論文第四篇。
- 建立數位宇宙九元存在模型。
- 區分資料值、數位紀錄、數位對象、數位實體、數位制度與數位世界。
- 提出數位存在判準與人工實在性。
- 建立識別、狀態、規則、關係四類同一性。
- 分析複製、分支、備份、刪除與數位死亡。
- 建立數位狀態、規則、權限、歷史與時間的本體地位。
- 提出數位回滾與物理回滾不等價。
- 將 Runtime／Kernel 定義為世界法則執行核心。
- 分析平台主權、數位臣民化、退出與語意可攜。
- 區分數位資產、檔案與制度權利。
- 將持續 Agent 定義為具有身份、記憶、權限、關係與歷史的數位實體。
- 提出五種數位世界治理風險。
- 加入社群帳號、遊戲角色、Git 儲存庫與研究 Agent 案例。

- 
- 提出十八類失敗模式與十二項可證偽研究方向。
  - 完成與 PU-0-05 《生成與限制》的銜接。