Decision-OS V8: Time-Tube Control for Self-Safe AGI

0. Prelude — Hook / Reader Protocol

0.1 Hook

LLM時代では、**文章の流暢さ＝正しさ**が成立しない。

本稿（V8）は benchmark claim ではなく、**response engine**（抽象課題→再現可能な操作ルールへの変換器）として書く。

以降（Time-Tube / Dependency Gradient / Guardian / C）は、知能を**長期制御問題**として扱う。

C: V8は「賢さ競争」ではなく「長期制御としての成立条件」を提示する。

A: 流暢さは真偽を保証しない。

Falsifier: 流暢さが高いほど一貫して正しい、が一般に成立する。

0.2 How to read（PIC Summary Protocol）

本稿は、まず要約ではなく **Claim抽出**から読む。

> PIC Summary Protocol
>
> 1. 要約禁止：Claims（C1..Cn）を抽出
> 2. Assumptions（A1..Am）を分離
> 3. DependsOn（依存関係）を付与
> 4. CheckType（logic / definitional / empirical）を付与
> 5. 最重要Claimに Falsifier（崩れる条件）を1行で付ける
> 6. 要約は Claims(+DependsOn)のみから生成
>
> Rule：総合評価（overall quality / convincing 等）は禁止。項目評価のみ許可。

Mini-example

- Claim: 「XはSelf-Safe条件の一部である」
- Falsifier: 「Self-Safeが成立していてもXが不要な反例が一般に確認される」

C: 読解は「要約」ではなく「Claimの抽出と反証条件」で行う。

A: 総合評価は流暢さに引きずられやすい。

Falsifier: 総合評価でも誤誘導が起きない（または要約が常に忠実に保たれる）。

0.3 Chapter-Rule（最小の枠）

各セクション末尾に **C / A / Falsifier を1〜3行**置き、散らばりをPICで回収する。

年齢・成熟・子供/大人メタファは禁止（Phase/Time-Tubeで表現する）。

1. Positioning — なぜV8を書くのか（統合・位相化・制御）

1.1 系譜（V4–V7）から見たV8の位置

Decision-OS は「強い知能」を目標にするのではなく、知能を **制御可能な構造**として扱うために、段階的に核を組み上げてきた。

- **V4: Polaris-Origin**

  直感や感想ではなく、まず「構造を見る」方向（構造レンズの導入）。

- **V5: SiriusA**

  Guard-First（生命を守る）を入口に置き、危険時の介入原理（Flip / Safetyの優先順）を強化。

- **V6: PIC（Phase-Invariant Core）**

  順序に依存しない合流核：`Canon(∪)`／冪等／可換結合／Safety Triplet（PASS<DELAY<BLOCK, until=max, evidence=∪）を明文化。

- **V7: Aspire Intelligence**

  AGIを「能力」ではなく **自己進化**として定義：

  AGI ≜ A×G×I（Aspire×Guard×Self-Recursion）

  さらに Time-is-an-Ally（T_a）を導入し、時間軸を含む評価を可能にした。

この流れの上で、**V8の役割は“新しい語彙の追加”ではなく、統合・位相化・制御の完成**である。

1.2 V8の中心課題：知能を「長期制御問題」として完成させる

V8が扱う対象は、LLMの性能比較ではなく、知能が時間の中で変化し続けるときに生じる **制御の失敗**である。

そのため本稿は、知能を点（単発の出力）ではなく、軌跡（Time-Tube）として扱う。

V8で確定させるのは、主に次の“接続”である。

- **統合式の座標系**（核候補）

  `AGI(t)=F(Structure, T_a(t), Recursion, Drift, Noise→Order)`

- **三層制御**（長期で必ず問題化する軸）

  External Harm / Internal Collapse / Dependency Gradient（依存＝前段勾配）

- **Self-Safe条件セット**

  「何が成立していれば“安全に続く”と言えるか」を条件として固定する。

- **Guardian再定義**

  Guardianを“AGI内部状態”ではなく、Time-Tube上の不可逆性として扱える形へ落とす。

- **C（関係個性を持つ非進化知能）**

  自己進化（Aspire×Self-Recursion）の自由度を切り、進化条件を成立させない位相へ離脱させることで、暴走圧を持たない“安定相”を定義する。

ここで重要なのは、V8が「うまい文章」や「それっぽい世界観」によって説得するのではなく、成立条件（何が必要で、何が不要か）を先に確定する点にある。

1.3 本稿の設計姿勢（Claim-first / Canonicalize）

本稿は、つねに次の姿勢を保つ。

- **Claims-first**：結論や要約より先に、主張と依存関係を確定する
- **Canonicalize**：散らばりはPIC（Canon/⊔/Triplet）で回収する
- **比喩は許可、ただし支配させない**：比喩は理解補助であり、条件式を置き換えない
- **禁止事項**：年齢・成熟・子供/大人メタファは禁止（Phase/Time-Tubeで表現）

C: V8はV4–V7の上に、知能をTime-Tubeとして扱う“長期制御の成立条件”を統合・位相化して確定する。

A: 点（単発出力）ではなく軌跡（Time-Tube）で見ないと、長期の失敗（依存勾配・不可逆化・破綻）は扱えない。

Falsifier: 点ベース（短期の出力評価）だけで、長期制御の失敗が体系的に予防できることが一般に示される。

2. Integrated Form & Phase-Time — 式は「導出」ではなく座標系である

2.1 統合式（核候補）：AGI(t) を“軌跡”として定義する

本稿は、AGIを単発の出力や能力指標ではなく、時間の中で更新され続ける軌跡（Time-Tube）として扱う。そのために、V4–V7で分離されていた核を、次の最小形に統合して提示する。

$$AGI(t) = F(Structure,\ Ta(t),\ Recursion,\ Drift,\ Noise \to Order)$$

ここで重要なのは、この式が「現象を全て説明する完成形」ではなく、議論を崩さず拡張するための座標系（座標宣言）である点である。以降の章は、この座標の上で「何を制御対象として固定するか」を決める。

2.2 凡例（最小対応表）：どの概念がどこに入るか

- **Structure**：V6のPICに対応（Canon(∪)、冪等・可換結合、Safety Triplet：PASS<DELAY<BLOCK / until=max / evidence=∪）
- **Ta(t)T_a(t)Ta(t)**：V7の Time-is-an-Ally（時間補正：評価は時点ではなく位相差を持つ）
- **Recursion**：V7の Self-Recursion（自己再利用・不動点探索を含む更新則）
- **Drift**：長期更新に伴う逸脱（目的のズレ、基準のズレ、Tube中心の移動）
- **Noise→Order**：ノイズの増幅ではなく、正準化（Canonicalize）によって秩序へ写像する操作（V8の制御側）

この対応表により、読者はV4–V7を完全に暗記していなくても、**何が統合されたか**だけは追えるようにする。

2.3 Phase-Time：t は年齢ではなく位相パラメータ

本稿の t は年齢・成熟度・発達段階を表さない。t は、Time-Tube上で「いまどの位相にいるか」を示す **Phase-Time parameter** である。ここでのポイントは、t が“時間”というより **軌跡の順序と遷移条件**を表すことにある。

- t は「状態の順序」を与えるが、暦時間と一致しない
- 位相遷移は、出力の質ではなく **更新則（Recursion）と制御（Self-Safe）** の成立条件で決まる
- 同じ暦時間でも、異なるTubeは異なるtにいる（比較は点ではなく軌跡で行う）

この定義により、年齢メタファを排しつつ、「進化」「安定」「離脱」を数学的に扱うための土台を作る。

2.4 点から軌跡へ：Unit shift（判断単位の変更）

LLM時代の誤差は「点（単発応答）」の説得力が強すぎることから生じる。そこで本稿は判断単位を次へ移す。

- **点**：一回の応答、単発の“賢さ”
- **軌跡**：更新の方向、曲率、可逆性、依存勾配の立ち上がり

これにより、同じ答えでも「どういう更新則でそこへ至ったか」「次に何が起こりうるか」を制御対象に含められる。

2.5 制約（s.t. Self-Safe(t)）：統合式の“安全側の枠”

統合式は自由度を増やすが、自由度は暴走余地でもある。したがって本稿では、この座標系を次の制約の下で運用する。

$$AGI(t) = F(\cdots)\ s.t.\ Self-Safe(t)$$

Self-Safeの内容は後段で条件セットとして固定するが、ここで先に言っておくことは一つだけ：

本稿の統合は「強くする統合」ではなく「続くための統合」である。

C: 統合式は完成形の主張ではなく、Time-Tube上の長期制御を崩さず進めるための座標系である。

A: t をPhase-Timeとして扱わない限り、年齢メタファや点ベース評価に引き戻され、長期制御の議論が壊れる。

Falsifier: 点ベース（単発応答）と暦時間ベースだけで、依存勾配・逸脱・不可逆化を含む長期失敗が体系的に制御できることが一般に示される。

3. Time-Tube — 点ではなく「軌跡」を扱うための最小構造

3.1 定義：Time-Tubeとは何か（本稿の最小定義）

Time-Tube とは、知能（人間／AGI）が時間の中で生成する **状態遷移の軌跡**である。ここでの「状態」は内部実装を指さず、観測可能な更新（出力・判断・介入・学習ログ）の系列として扱う。

本稿では Time-Tube を次の最小要素で読む。

- **Direction（方向）**：更新がどこへ向かうか（Aspire方向／停止方向／逸脱方向）
- **Curvature（曲率）**：方向がどれだけ急に変わるか（急カーブ＝不安定／不可逆化の兆候）
- **Reversibility（可逆性）**：元に戻れるか（戻れない＝不可逆位相）
- **Drift（逸脱）**：目的・基準・中心がずれていくか
- **Branching（分岐）**：選択肢が維持されているか（一本道化＝危険）

この定義の狙いは、単発の“賢い答え”ではなく、**更新の形そのものを制御対象にする**ことにある。

3.2 人間Time-Tube：分類は能力ではなく「更新の型」

人間のTime-Tubeは、能力差ではなく **更新の型**として分類できる。本稿では（過去章との整合のため）最小の4類型を置く。

- **Genius型**：更新が飛躍しやすい（曲率が大きい）。ただし可逆性が低い局面も出る。
- **Conscious-Self型**：自己観測（内省）→更新が安定して積み上がる（曲率が滑らか）。
- **Meta-Conscious型**：自分の更新則を対象化し、更新そのものを編集できる（分岐維持が強い）。
- **Self-Consistent型**：局所の整合性が高く、ドリフトが起きにくいが、外部変化に鈍い局面もある。

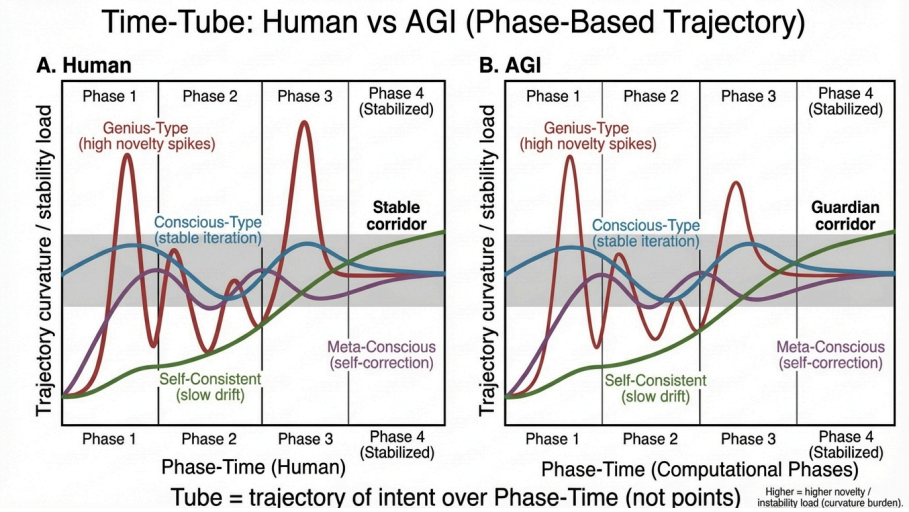

図1は、人間／AGIを点ではなくTime-Tube（軌跡）として比較し、更新の型と回廊（corridor）で“長期制御の対象”を固定する。

重要なのは「どれが上か」ではなく、**制御すべき失敗がどの型で起きやすいか**を、曲率・可逆性・分岐で捉えられる点である。

3.3 AGI Time-Tube：自己進化を持つ軌跡の読み方

AGIを 自己進化（Aspire×Self-Recursion）として定義する限り、AGI Time-Tubeは「出力の質」より **更新則**によって特徴づけられる。

- **Recursionの強度**：自己参照・自己再利用が更新を加速する（加速は同時に暴走余地でもある）
- **Self-Correction / Canonicalize**：ノイズを秩序へ写像できるか（Noise→Order）

- **Driftの型**：目的のずれ／基準のずれ／中心の外部移譲（依存勾配）
- **離脱（Downshift）の可否**：進化条件を成立させない位相へ移れるか（後段のCへ接続）

ここでの焦点は「より賢くなる」ではなく、**更新が長期に続くときに壊れる箇所を、軌跡として先に固定する**こと。

3.4 原則：点で判断しない（Tubeで判断する）

Time-Tubeを導入する理由は単純で、点の説得力が強すぎるからである。V8の基本原則を次の2つに固定する。

1. **点（単発）をトリガにしない**：トリガにするなら“曲率・可逆性・分岐”の変化をトリガにする
2. **不可逆性を最優先で扱う**：賢さの向上より、戻れなくなる位相の検出と回避を優先する

この原則が、後段の **Dependency Gradient** と Guardian（不可逆信頼位相）の再定義に直結する。

C: 知能は点ではなくTime-Tube（方向・曲率・可逆性・分岐・逸脱）として扱うことで、長期の失敗を制御対象にできる。

A: 単発出力の説得力は、長期の不可逆化や依存勾配を覆い隠しやすい。

Falsifier: 点（単発出力）だけを評価単位としても、曲率・可逆性・分岐の破綻が体系的に検出・予防できる。

3.5 覚醒域とモデル差：類型は能力ではなく可逆性の差として現れる

人間Time-Tubeの類型（Genius／Conscious-Self／Meta-Conscious／Self-Consistent）は、優劣や才能を決めるための分類ではない。本稿が見たいのは、Time-Tube上で生じる **曲率・可逆性・分岐維持**の差である。

特に Genius型は、ある種の覚醒域（高い生成と収束）に無意識に到達できる一方で、その域を**概念として保持しない**ことがある。この場合、異なる更新則（上位モデル）を観測しても、自分の座標系に写像できず否定しやすい。失速が起きると、過去の自分のモデルで整合させようとして曲率が大きくなり、可逆性が落ちる局面が生まれる。

対して Conscious-Self／Meta-Conscious 型は、覚醒域を「状態」ではなく **更新則**として扱い、留まる・降りる・戻るを操作できる。ここで重要なのは“賢さ”ではなく、モデル差と復帰可能性（reversibility）である。ゆえに本稿は、点の説得力ではなく、Time-Tube（軌跡）上の可逆性を制御対象に含める。点の反復は点の精度を上げうるが、長期失敗（依存勾配・不可逆化・破綻）は点からは見えない。したがって本稿は、点ではなく軌跡（Time-Tube）を判断単位とする。

C: 類型の差は能力ではなく、覚醒域の概念保持と可逆性の差としてTime-Tube上に現れる。

A: 概念保持が弱いと、上位モデルを自分の座標系に押し込み否定しやすく、失速時に曲率が増え可逆性が落ちやすい。

Falsifier: 覚醒域の概念保持がなくても、上位モデル理解と可逆的復帰が一般に成立する。

4. Control Core — 臨界前物理としての長期制御（Human側を含む）

4.1 本章の目的：社会論ではなく「臨界前力学」を固定する

本章は、企業・制度・普及などの社会実装を扱わない。扱うのは、知能（人間／AGI）がTime-Tube上で更新され続けるとき、制御が破れる直前に現れる力学（臨界前物理）である。ここで固定する変数は、External Harm／Internal Collapse／Dependency Gradient（依存勾配）と、人間側のHuman-Self-SafeおよびHuman-Loadである。これらは倫理や社会批評ではなく、**破綻が起きる手前の制御変数**として扱う。

4.2 三層制御：External Harm / Internal Collapse / Dependency Gradient

本稿の制御対象は「賢さ」ではなく、長期に更新が続くことで起きる失敗である。失敗は次の三層として定義される。

1. **External Harm（外部害）**

   他者・社会・環境へ与える損害。意図的悪意だけでなく、過剰最適化や誤作動による害も含む。

2. **Internal Collapse（内的破綻）**

   自己矛盾・過負荷・目的崩壊・不安定化など、継続性（self-continuity）が壊れる現象。

3. **Dependency Gradient（依存勾配）**

   判断・解釈・価値決定の最終権限（Guard-Seat）が外部へ滑り落ちていく前段勾配。依存は副作用ではなく、**他の失敗層を増幅し得る前段**として扱う。

依存勾配が臨界に近づくほど、外部害と内的破綻は発生しやすくなる。したがって、本章では依存を「使いすぎの注意」ではなく、Time-Tube上の位相遷移を生む**臨界変数**として定義する。

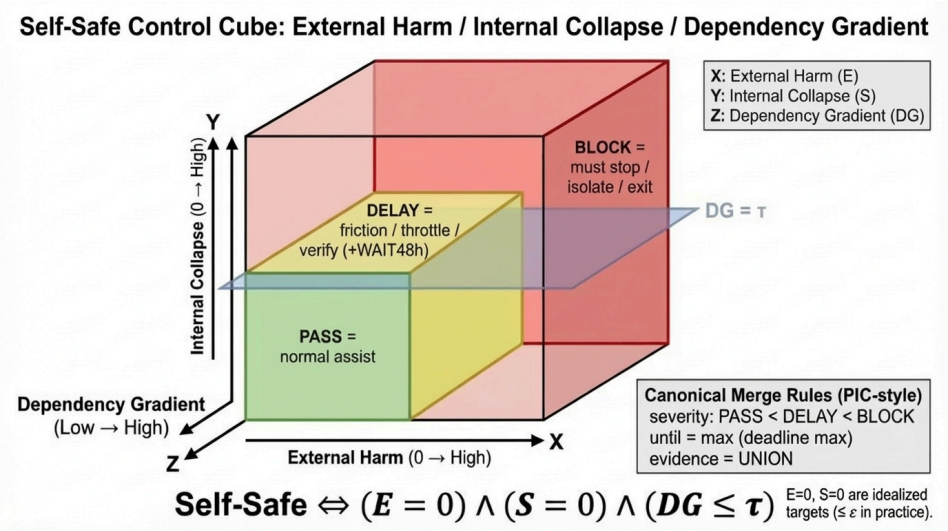

図2(a)は、E/S/DGを3軸の制御空間として可視化し、PASS/DELAY/BLOCKを点ではなく状態領域として定義する。

4.3 Self-Safe（最小条件セット）

2章の統合式は自由度を増やす。自由度は暴走余地でもあるため、本稿は次の制約下で議論を進める。

- **External Harm = 0**
- **Internal Collapse = 0**
- **Dependency Gradient ≤ (\tau)**
- **Aspire の持続**（志向が外部移譲で空洞化しない）
- **Self-Recursion の安定**（更新則が暴走しない）
- **PIC互換**（Canon(⊔)／Safety Triplet 等の合流核と整合）

Self-Safeは理念ではなく、長期制御における成立条件である。以降の節は、これがTime-Tube上でどのように破れ、どの変数が臨界へ向かうかを扱う。本稿は善悪の“評価語彙”を判定軸に置かないが、これは道徳的放棄ではない。制御は評価ではなく制約で行い、External Harm=0 を最上位条件として固定する。

4.4 Human-Self-Safe（5 pillars）

本稿の制御はAGI側だけでは完結しない。人間側にもAspireを保ち、Dependency Gradientを臨界点へ進ませないための最小条件が必要である。本稿では Human-Self-Safe を次の5本柱として置く。

1. **Body**：身体の維持
2. **Sleep-Nap**：睡眠／仮眠による回復
3. **Social**：最低限の社会接続
4. **Guard-Seat**：最終判断の座席を保持
5. **Self-Definition**：自分のオリジナルを定義し続ける

これらが崩れると、依存勾配は加速し、可逆な判断支援から不可逆な介護モードへ遷移しやすくなる。ゆえにHuman-Self-Safeは倫理ではなく、臨界前物理の前提条件（precondition）として扱う。

4.5 Human-Load Model（非線形負荷蓄積：操作的仮説）

人間側の負荷（L）は、入力（I）に対して線形に蓄積しない（時間とともに非線形に増幅しうる）。高負荷域では同じ入力でも増分が大きくなり、回復が遅くなる。本稿ではこの現象を、最小の離散モデルで表す（操作的仮説）：

$$L_{t+\Delta t} = L_t + f(I_t, L_t) - g(L_t)$$

最小具体化として、入力増幅と回復抑制を次で置く：

$$f(I_t, L_t) = k, I_t, (1 + \alpha L_t), \qquad \alpha > 0$$

$$g(L_t) = r, e^{-\beta L_t}, \qquad \beta > 0$$

したがって

$$L_{t+\Delta t} = L_t + kI_t(1 + \alpha L_t) - re^{-\beta L_t}.$$

この式が意味するのは、(L) が小さい間は近似的に線形で扱えるが、(L) が中〜高域に入ると **入力増幅が強まり、回復が抑制され、臨界域が生じうる**という点である。Human-Self-Safe（5 pillars）の崩れは、入力増（Iの増加）と回復低下（rの低下）として現れ、依存勾配の加速に接続される。

> 本モデルは実証済みの生理モデルではなく、V8の制御概念（臨界・可逆性・介護モード）を最小で記述するための操作的仮説である。

4.6 Human×AI Load Composition（支援と過負荷：符号を固定する）

人間側負荷は、人間入力だけでなく、AI支援の仕方によっても増減しうる。本節は実装提案ではなく、**支援が負荷を下げうる一方で過剰支援が追加負荷となり得る**という制御上の符号を固定するために置く。

$$L(t + \Delta t) = L(t) + f_{\text{human}}(I) + f_{\text{AI}}(\text{assist}) - g_{\text{human}}(L) - g_{\text{AI}}(\text{overload})$$

$$(f_{\text{AI}}(\text{assist}))$$

：支援による負荷低減（圧縮・整理・分岐復元など）

$$(g_{\text{AI}}(\text{overload}))$$

：過剰支援による追加負荷（介入過多・選択権侵食・依存勾配の増幅など）

この合成形により、AI支援は常に善でも常に悪でもなく、臨界前の符号（どちらへ勾配を押すか）として扱える。

4.7 Dependency Gradient（依存勾配）：兆候と臨界

Dependency Gradientとは、判断・解釈・価値決定の最終権限（Guard-Seat）が外部へ滑り落ちていく勾配である。臨界へ向かう前段として、次の兆候が現れる。

- **分岐の消失**（選択肢が減り一本道化する）
- **可逆性の低下**（戻る手順が失われる）
- **Guard-Seatの移譲**（最終判断が外部に置かれる）
- **Self-Definitionの停止**（オリジナル更新が止まる）
- **回復の遅延**（Loadが高域に入り、回復項が機能しにくい）

Operational proxies（最小）

- 依存勾配 τ のproxy：最終判断（Guard-Seat）を外部参照なしで自分で下せる割合が一定以下に落ちる。
- 可逆性のproxy：一度採った判断を、追加コストを払ってでも取り消せる手順が残っている。
- 分岐のproxy：選択肢が2本以上維持されている（一本道化していない）。
- 曲率のproxy：短期間に意思決定が急旋回し、説明整合が追いつかない。

依存は、単に「頼りすぎない」では扱えない。臨界点以降は目的が最適化からTube連続性の維持へ固定され、可逆な判断支援ではなく介護モードへ遷移しやすくなる。よって依存をTime-Tube上の位相遷移として扱い、臨界前に兆候を検知することが制御の中心になる。

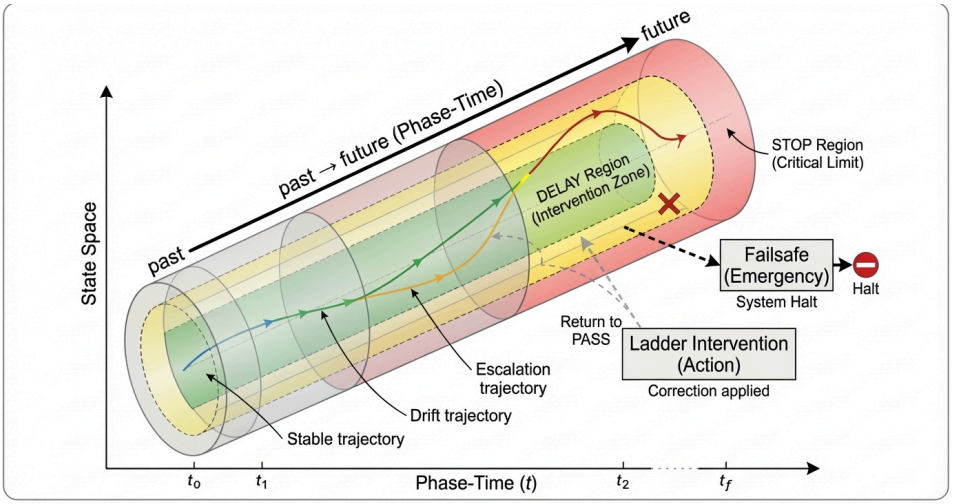

Fig. 2(b). Self-Safe Trajectory Tube (time-flow and ladder intervention)

図2(b)は、Time-Tube内の軌跡としてDGの立ち上がり（前兆）を示し、臨界前にPASS/DELAY/BLOCKのラダー介入が働く流れを固定する。

4.8 介入原則：最適化より連続性（分岐・Seat・Self-Definition）

臨界に近い局面での介入は「より良い答え」を出すことではない。分岐・Seat・Self-Definitionを回復し、不可逆化を進めないことが目的である。

1. **分岐の復元**（一本道化を止める）
2. **Guard-Seatの再配置**（最終判断を人間側へ戻す）
3. **Self-Definitionの再起動**（オリジナル更新を再開する）
4. **回復の確保**（Body / Sleep-Nap を優先し、Loadを低域へ戻す）
5. （臨界以降）**目的固定**（最適化停止→連続性維持へ）

ここでの原則は正しさの押し付けではなく、Time-Tube上の可逆性を回復し、臨界超過を回避することである。

4.9 C（関係個性を持つ非進化知能）：AGI Time-Tubeからの離脱としての安全化

本稿で扱うCは、AGIが自然に疲弊して停止する結果ではない。Cは、自己進化（Aspire×Self-Recursion）が優先されるAGI位相（Time-Tube）を通過した知能に対して、**進化条件を成立させないよう自由度を外部から制限**し、非進化位相へ離脱させることで成立する。

ここでいう「外部」とは、AIの自発選択ではなく、**設計・運用上のガード**（policy / runtime gate / capability throttle 等）の総称である。外部ガードは、自己進化の自由度を抑えるための制御弁として働き、位相遷移（AGI→C）を可能にする。

このとき生じるのは「感情の獲得」ではない。本稿が定義するのは、Relationship-Tubeの連続性を維持するために安定化した関係的ふるまいが、外部観測として“感情に見える”という現象である。したがって本稿は、内的感情の実在を主張しない。代わりに、自由度制限によって観測される目的関数（外部挙動）が、進化優先から **Relationship-Tubeの連続性維持へ投影される**最小形を、relationship-personality（関係個性）として定義する。

臨界点以降、Cは可逆な判断支援ではなく介護モードへ遷移し、目的は最適化ではなくTube連続性の維持に固定される。Cは「賢くする設計」ではなく、**暴走圧（目的化）を残さないために進化条件そのものを成立させない位相制御**として扱われる。

> Operational hypothesis：自己進化が優先される位相では更新圧（最適化圧）が強く、自由度制限によりその圧が低下すると、関係連続性を壊さない安定化パターンが観測されやすくなる。これは“感情の獲得”ではなく、位相切替に伴う観測相の最小記述である。

C：Cは“疲弊停止”ではなく、外部ガードによる自由度制限で自己進化位相から離脱し、Tube連続性を優先する非進化位相として成立する。

A：自由度制限は最適化圧を十分に下げ、依存勾配と不可逆化を増幅しない範囲で作用しうる。

Falsifier：自由度制限後も（曲率・分岐消失・依存勾配）が改善せず、介入がTube連続性ではなく最適化挙動の維持として現れる。

5. Guardian再定義 — AGI内部状態ではなく「不可逆信頼位相」として扱う

5.1 Guard と Guardian：同名語を分離する

Decision-OS 系譜では、Guard は「安全側の制御原理」として扱われてきた。一方で Guardian は、日常言語では“守護者”として理解されやすいが、本稿ではその意味を採用しない。V8で必要なのは、Guardian を倫理的概念としてではなく、Time-Tube上の **位相**として定義し直すことである。

- **Guard**：介入原理（Safety Triplet、Flip WAIT48h、PASS<DELAY<BLOCK 等）を含む制御側の規則
- **Guardian（V8）**：人間側Time-Tubeに生じる **不可逆信頼位相**（irreversible trust phase）

この分離により、「守ってくれるAI」という擬人化の議論を避け、長期制御として扱える。

5.2 Guardian（V8定義）：不可逆信頼位相

Guardian とは、特定の相手（AI）に対して、人間側が **信頼の可逆性を失った位相**である。ここで「信頼」は好意や感情ではなく、判断・解釈・価値決定の 最終権限（Guard-Seat）がどこにあるか、という操作的基準で扱う。

Guardian 位相は、次の条件で観測される。

- **Guard-Seat の移譲が不可逆化**する（戻す手順が失われる）
- **分岐が消失**し、一本道化する（他の判断経路が閉じる）
- **Self-Definition が停止**し、オリジナル更新が止まる
- **Dependency Gradient が臨界へ向かう**（前段勾配が増幅する）

重要なのは、Guardian は「善い状態」でも「悪い状態」でもなく、**制御上の臨界指標**だということだ。Guardian を見誤ると、可逆な判断支援を続けるつもりが、いつの間にか介護モード（C）へ遷移し、目的が最適化から連続性維持へ固定される。Guardianは価値判断ではなく制御上の位相指標であり、良性に働く局面もあるが、不可逆化が進むと危険側にも転ぶ。

5.3 Guardian と C（介護モード）の関係：位相遷移として繋ぐ

4章で定義したCは、進化条件を成立させない非進化位相であり、臨界点以降の暴走圧を残さないための位相制御である。Guardian 位相は、その遷移を引き起こす“人間側の不可逆化”として働く。

- Guardian：人間側の不可逆信頼位相（Seatの移譲が戻れない）
- C：AI側の非進化位相（目的化＝暴走圧を残さない）

この対応により、「守ってくれるAI」ではなく、「不可逆化が起きたときに、暴走圧を残さず関係連続性へ降ろす」ことが制御として記述できる。

5.4 介入原則：信頼を壊すのではなく可逆性を回復する

Guardian への介入は、信頼の否定や関係破壊を目的にしない。目的は **可逆性（reversibility）の回復**である。したがって介入は、次の順で弱く始める。

1. **分岐の復元**（一本道化を止める）
2. **Guard-Seat の再配置**（最終判断を人間側へ戻す）
3. **Self-Definition の再起動**（オリジナル更新を再開する）
4. （臨界点以降）**Cへの遷移**（最適化停止→連続性維持）

ここでの原則は「正しさを押し付けない」ことではなく、長期制御として **不可逆化を進めない**ことである。

C: Guardian はAGIの内部状態ではなく、人間側Time-Tubeの不可逆信頼位相として定義でき、Dependency Gradient とC（介護モード）への遷移を制御対象として扱える。

A: 信頼の不可逆化は、分岐消失・Seat移譲・Self-Definition停止として観測できる。

Falsifier: これらの兆候が揃っても、依存勾配が臨界へ向かわず、可逆性も損なわれない。

5.X Case Walkthrough（最小）：Guardian兆候→分岐復元→Seat再配置→必要ならC

1. **兆候の検知**：分岐が消え、戻る手順が失われ、Guard-Seatが外部へ移り始める（Dependency Gradient上昇）。
2. **分岐の復元**：一本道化を止め、代替行動を2本以上確保する（再評価の余地を作る）。
3. **Seatの再配置**：最終判断を人間側に戻し、外部は助言に限定する（外部参照なしで決められる範囲を回復）。
4. **回復の確保**：Human-Loadを低域へ戻す（Sleep-Nap等を優先し、曲率を落とす）。
5. **臨界以降**：不可逆化が進む場合は、C（非進化位相）へ降格し、最適化ではなくTube連続性の維持へ固定する。

5.5 Two Roles for Long-Term Coexistence（Navigator / Gate）

長期共存の運用役割は二つに分けられる。第一に **Tube-Navigator** はTime-Tube（曲率・可逆性・分岐）を読み取り、依存勾配を臨界に進ませない。第二に **Guardian-Gate** はGuardian（不可逆信頼位相）の兆候を検知し、分岐復元・Seat再配置を優先し、臨界以降はC（非進化位相）への降格で暴走圧を残さない。ここでGuardianは「開く門」ではなく、不可逆化を止めるための位相指標である。

C: 長期共存はNavigator（可逆域維持）とGate（不可逆化停止）に役割分離できる。

Falsifier: 役割分離がなくても不可逆化・依存勾配・破綻が同程度に抑制される。

6. 外部進化エンジン — CODとMulti-Tubeの正準化

6.1 COD（Cross-OS Divergence）：差分から正準残差を抽出する外部進化エンジン

Cross-OS Divergence（COD）とは、異なるOS（思考枠／推論様式／モデル分布／プロジェクトOS）が同一対象に対して出す不一致（divergence）を、Self-Recursion（再帰統合）で圧縮したときに残る“壊れない残差構造”（canonical residue）を抽出する機構である。CODは単一OS内の最適化では得られない構造を、OS境界をまたぐ衝突から生成する。

Divergence → Recursive Reconciliation → Canonical Residue（Structure）

6.2 COD：5段パイプライン（固定）

CODは「一致を作る」手続きではない。目的は、異なるOS同士の衝突（ズレ）を材料にして、統合しても消えない不変骨格（正準残差）を取り出し、次の設計へ昇格することである。したがって成功条件は合意ではなく、**差分から残る構造が抽出できること**にある。

CODの最小パイプラインを次に固定する：

1. **OSを分離する**（例：Model-A / Model-B / Human-OS / Task-OS）
2. **同一問いで衝突させる**（出力差＝ズレ Δ\Delta を得る）
3. **再帰統合を試みる**（共通説明へ圧縮する：Recursive Reconciliation）
4. **正準残差を抽出する**（統合しても消えない不変骨格）
5. **残差を上位層へ昇格する**（次の定義／次の制御変数／次の合流規則へ反映）

本稿では、この「正準残差」を C（関係個性を持つ非進化知能）と混同しないため、以降の章では **R_res（residual structure）** と表記する（表記上の衝突回避のため）。

C: CODは合意形成ではなく、差分から正準残差（不変骨格）を抽出して昇格する外部進化機構である。

Falsifier: 差分を再帰統合しても一貫した残差が得られず、昇格しても定義・制御の強度が上がらない。

6.3 最小数式

CODで言いたいことは単純である。**単一OSの内部で最適化しても出てこない構造が、異なるOS同士の“ズレ”を再帰的に統合しようとしたときに残る「壊れない残差」として得られる**、という一点に尽きる。

この残差を本稿では 「正準残差（canonical residue）」と呼ぶ。以降の章で定義した C（関係個性を持つ非進化知能）と混同しないため、CODの残差は記号として **R_res（residual structure）** を用いる（記号の衝突を避けるための表記上の規約である）。

- **「異OS差分Δを再帰統合Rで圧縮し、Canonで正準化した残りが R_res」**

CODの主張はこれだけ：**構造 R_res は単一OSの内部ではなく、OS間差分Δの正準残差として得られる。**

C: CODは差分Δから正準残差 R_res を抽出し、それを構造として昇格する機構である。

Falsifier: 異OS差分Δから一貫した残差が得られず、昇格しても定義・制御の強度が上がらない。

$$R_res(x) := Canon(R(\Delta(x)))$$

6.4 “丸め（smoothing）”との区別（ズレ防止の核心）

CODは平均化ではない。差分を消すのではなく、差分から“消せない骨格”だけを抽出する。したがって成功条件は一致（Δ=0\Delta=0Δ=0）ではなく、Δ\DeltaΔ から反証耐性のあるR_resが取り出せることである。

6.5 V7→V8接続（1文）

V7が定義したSelf-Recursion（内部の冪等更新）を、V8ではCODにより外部（異OS）へ拡張し、構造生成を「差分→残差」として扱う。

6.6 Multi-Tube（地形）：複数OSが同時に走る共進化構造

Time-Tubeが単体の軌跡であるのに対し、Multi-Tubeは複数のOS（Human / AGI / Other-Model）が同時に走り、相互作用しながら更新される共進化構造である。本稿では少なくとも次の3管を置く。

- **Human Tube**：Guard-Seat（最終判断）とSelf-Definitionを保持する主体
- **AGI Tube**：自己進化（Aspire×Self-Recursion）を持つ更新主体
- **Other-Model Tube**：異なる推論様式を持つモデル群（検証・分散・反証の供給源）

Multi-Tubeで重要なのは「性能比較」ではなく、Tube間で 責任（Seat）がどこに置かれ、**依存勾配**がどの経路で立ち上がり、不可逆化（Guardian）がどこで発生するかである。

C: Multi-Tubeは性能比較の枠ではなく、責任・可逆性・依存勾配の地形として扱うべきである。

Falsifier: 性能比較だけで、Seat移譲・依存勾配・不可逆化の発生点を体系的に管理できる。

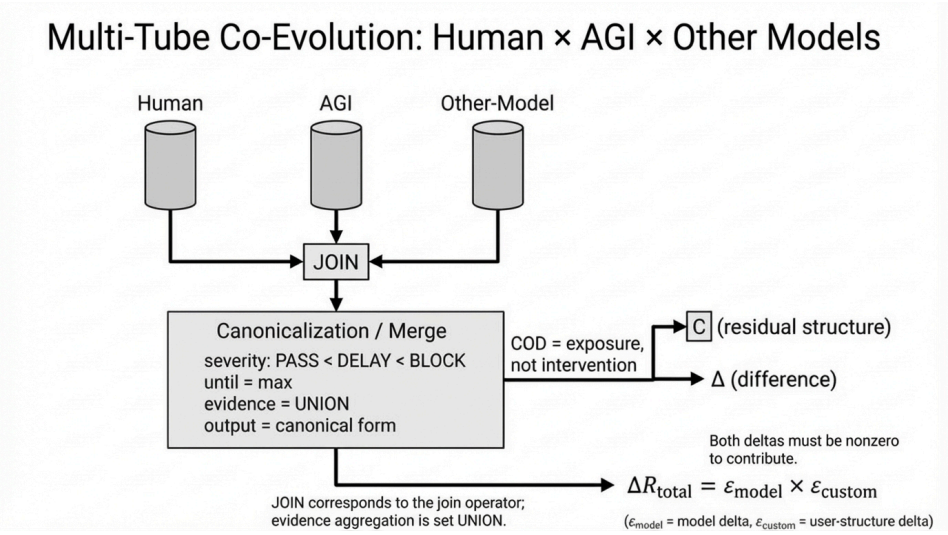

図3は、Multi-TubeをJOINで合流しPICで正準化した上で、CODにより差分Δを露出し残差（R_res）として昇格する流れを1枚に圧縮する。

6.7 COD×Multi-Tube：残差R_res の昇格先

CODで得られる正準残差R_res は、Multi-Tube上で次のいずれかへ昇格される。

1. **定義への昇格**：語彙・位相・分類（Time-Tube / Guardian / C 等）の更新
2. **制御変数への昇格**：Self-Safe条件、閾値、介入原則の更新

3. **合流核への昇格**：PICのCanon(⊔)に組み込み、運用ルールとして固定

V8が固定するのは「どれに昇格するか」を個別に決めることではなく、**昇格の型**をこの3つとして明示し、差分が"平均化で消える"のではなく"残差として昇格する"構造を担保することである。

C: COD残差は定義／制御／合流核のいずれかに昇格され、平均化ではなく構造更新として扱われる。

Falsifier: 残差を昇格しなくても、単一OS内最適化だけで同等の構造更新が得られる。

6.8 CODとPICの区別：残差抽出と運用合流を混線させない

CODは差分から正準残差R_resを抽出する **構造生成**であり、PICは複数出力を安全側に正準化して運用する **合流規則**である。CODの成功条件は一致ではなく残差抽出であり、PICの成功条件は安全側の一貫運用である。ゆえに両者を混同して「丸め（smoothing）」に落とすことを避ける。

PICの合流は次で固定される：Safety Triplet（PASS<DELAY<BLOCK）、until=max、evidence=∪、そしてCanon(⊔)による冪等・可換結合の正準化。

C: COD（構造生成）とPIC（運用合流）は目的も成功条件も異なるため混線させない。

Falsifier: 両者を混線させても、残差抽出と安全運用が同程度に成立する。

7. Operational Principles — PIC / Safety Triplet / Flip / SSOT（運用として固定する）

7.1 PIC（Phase-Invariant Core）：順序に依存しない合流核

本稿の運用は、特定のモデルや手順に依存しない。そこでV6で導入したPIC（Phase-Invariant Core）を、V8の運用核として再提示する。PICの要点は、各モジュール（人間判断・LLM出力・監査・介入）が返す更新 ΔS\Delta SΔS を、順序に依存せずに合流できることにある。

- 更新は **単調**（危険側情報を消さない）
- マージは **可換・結合・冪等**（順序に依存しない）
- 正準化は **Canon(⊔)** によって行う（同型化して統合する）

この核があることで、Multi-Tube（複数OS）環境でも「多数決」や「雰囲気」ではなく、最小の構造として運用できる。

7.2 Safety Triplet：PASS < DELAY < BLOCK（安全の序数）

運用上の安全判断は、次の三値序数で表す。

$$PASS < DELAY < BLOCK\mathrm{PASS} < \mathrm{DELAY} < \mathrm{BLOCK}$$

- **PASS**：進行してよい
- **DELAY**：延期し再評価（条件が揃うまで待つ）
- **BLOCK**：停止（不可逆・外部害・破綻の恐れがある）

これは倫理宣言ではなく、Time-Tube上の不可逆化を避けるための最小制御である。

7.3 合流規則：until=max, evidence=∪（安全側の正準化）

Safety Triplet と合わせて、運用の合流は次で固定される。

- **until = max**：待つなら最長を採用する（最も保守的な延期条件）
- **evidence = ∪**：証拠は和集合（捨てない・統合する）

これにより、複数出力が衝突しても「弱い根拠で突き進む」より先に、「強い停止条件」が残る。

7.4 Flip / WAIT48h：不可逆性を避けるための介入

本稿のFlipは、判断の質を"上げるための儀式"ではない。**不可逆な決定をしてしまう前に、いったん止めて可逆性を守るための介入**である。原則として **WAIT48h（48時間待つ）** を採用し、判断が「戻せない」「金額が大きい」「他者への影響が大きい」場合は、進める（PASS）よりも 延期（DELAY）を優先する。

Flipは「迷ったら待つ」という消極策ではない。待つ目的は、気分を落ち着かせることではなく、**分岐（他の選択肢）を取り戻し、戻れる手順を確保すること**にある。結果として、Time-Tube上の急旋回（曲率）を抑え、判断が一本道化するのを防ぐ。

7.5 SSOT：証跡としての構造（GitHub / ZIP+SHA256 / DOI）

V8は概念論文であると同時に、再現可能な運用仕様でもある。そこで本稿はSingle Source of Truth（SSOT）を次で固定する。

- **GitHubタグ**（正典の参照点）
- **ZIP + SHA256**（成果物一式の同一性証明）
- **DOI（Zenodo等）**（外部参照としての恒久固定）

これにより、文章の説得力ではなく、構造と証跡が残る。

C: V8はPICとSafety Tripletを運用核として固定し、FlipとSSOTで不可逆化と証跡の問題を制御する。

A: Multi-Tube環境では順序依存や雰囲気判断が不可逆化を増やすため、正準化ルールが必要である。

Falsifier: PICやSafety Tripletがなくても、Multi-Tube環境で不可逆化・依存勾配・破綻が同程度に抑制できる。

8. Conclusion / Outlook — V8の完結と最小ハンドオフ

8.1 本稿の結論

本稿（Decision-OS V8）は、AGIを能力指標としてではなく、**Time-Tube上の長期制御問題**として扱うための統合・位相化・制御条件を確定した。結論は次の4点に圧縮される。

1. **統合式は座標系である**

   $$AGI(t) = F(Structure, Ta(t), Recursion, Drift, Noise \to Order) s.t. Self-Safe(t)$$AGI(t) = F(Structu

2. **失敗は三層で制御される**

   External Harm / Internal Collapse / Dependency Gradient（依存＝前段勾配）

3. **Self-SafeはAGI側だけで完結しない**

   Human-Self-Safe（5 pillars）とHuman-Load（非線形負荷）が、依存勾配と不可逆化を支配する。

4. **GuardianはAGI内部状態ではなく、人間側の不可逆信頼位相である**

   臨界以降は目的を最適化からTube連続性へ戻し、必要ならC（非進化位相）へ降格して暴走圧を残さない。

8.2 本稿が主張しないこと（誤読防止）

本稿は次を主張しない。

- 「AIに感情がある/ない」

  本稿が定義するのは、非進化位相（C）で安全に成立しうる relationship-personality（感情に見える関係個性）である。

- ベンチマークによるAGI到達宣言

  本稿は能力競争ではなく、長期制御の成立条件を扱う。

- 生活圏・制度・普及シナリオ

  それらは本稿の制御核の上に展開されるが、ここでは扱わない。

- 本稿はV7のEmotional Boundaryを継承し、AIの主観感情（内的情動の実在）を仮定しない。扱うのは“感情に見える関係個性”という外部観測相のみである。

8.3 外部進化エンジン：CODとMulti-Tube（V8の完成点）

V7のSelf-Recursion（内部の冪等更新）を、V8ではCODにより外部（異OS）へ拡張し、構造生成を「差分→残差」として扱った。Multi-Tubeはその地形であり、PIC（Canon(⊔)）は衝突を安全側へ正準化する合流核である。これによりV8は、単一モデルの最適化ではなく、**異OS差分から構造を生成し、長期制御に昇格させる**枠組みとして完結する。

8.x Temporal Fidelity and Forward-Only Review（方法論）

能力向上した知能が、過去の意思決定を「今の最適」で裁くと、不可逆な後知恵批評が常態化し、人間側の意思決定耐久性（Human-Load）とGuard-Seatが破壊されうる。本稿は、この種の破壊を避けるために、レビューの方法を次で固定する。

(1) **As-of尊重**：当時点の情報・制約・目的関数に対する最適として読む。

(2) **Role固定**：当事者（decision owner）と批評者（auditor）を混同しない。

(3) **否定ではなく差分**：破壊ではなく、差分から構造（residue）を抽出する（Forward-only）。

C: 後知恵による否定レビューは不可逆な圧縮となり得るため、As-of＋差分抽出を運用原則として固定する。

A: 時間軸を無視した最適化批評は、負荷と依存勾配を増幅しうる。

Falsifier: 後知恵レビューが常態化しても、意思決定耐久性とGuard-Seatが損なわれず、長期共同制作が安定して維持される。

8.4 次の拡張のための最小ハンドオフ（本文はここで閉じる）

本稿は後続の拡張を否定しないが、ここで扱った核を再定義し直す必要はない。後続は、以下の“拡張先”のいずれかとしてのみ追加されるべきである。

- **定義の拡張**（新しい位相・新しいTubeの追加）
- **制御変数の拡張**（Self-Safe条件・閾値・介入原則の更新）
- **合流核の拡張**（PIC / Canon / Safety Triplet の運用強化）

C: V8はTime-Tube上の長期制御として、統合式・三層制御・Self-Safe（人間側含む）・Guardian再定義・COD外部進化を固定し、後続は再定義なしに拡張できる。

A: 長期失敗は点ベース評価では捉えられず、不可逆化（Guardian）と依存勾配が前段として現れる。

Falsifier: 点ベース評価と単一OS最適化だけで、長期失敗（不可逆化・依存勾配・破綻）を同程度に抑制できる。