arXiv:2603.04810v1 [cs.DC] 5 Mar 2026

语义时间之箭，第四部分：

事务为何失败

Paul Borrill
DÆDÆLUS

02026年2月27日

文件同步、电子邮件与记忆的脆弱性

摘要

这是组成《语义时间之箭》的五篇论文中的第四篇。第一至第三部分确立了：计算领域隐藏的时间之箭是语义性的，而非热力学性的（Borrill，2026a）；双边事务协议通过强制性的反思阶段建立因果顺序（Borrill，2026b）；以及 RDMA 的完成语义以工业规模实现了 FITO 范畴错误（Borrill，2026c）。

本文追溯 FITO 范畴错误在数据中心之外的影响，探讨人们日常使用的系统。我们考察三个领域，其中仅向前的时间假设会摧毁意义：文件同步，云平台因最后写入者胜无法表示分布式因果性，而悄然删除用户内容；电子邮件，基于时间戳的排序会产生幻影消息、因果关系违规和同步停滞；以及记忆——包括人类记忆和人工记忆——在缺少事务保证的情况下运行的重构过程会造成系统性的语义损坏。

在每个领域，我们都发现了同一种结构模式：系统在时间上向前提交状态变更，却没有反思阶段，因此无法区分成功的语义整合与单纯的时间先后。这种模式并非巧合，而是 FITO 范畴错误在不同尺度上的表现：网卡缓冲区中的字节、云端的文件、收件箱中的消息、海马体中的记忆痕迹、Transformer 中的词元。

我们得出的结论是：只要系统将信息向前流动视为足以证明意义存在，语义时间之箭就遭到了违背。第五部分将展示莱布尼茨桥如何提供一个统一框架，以弥合所有五个领域中的这一鸿沟。

1 引言：意义何以失落

本系列前几篇文章以日益具体的工程层次探讨了语义时间之箭：哲学基础（第一部分）、链路级协议设计（第二部分）以及互连硬件（第三部分）。在每个层次上，都呈现出相同的模式：假定时间向前流动足以确立意义的系统，会造成系统性的语义损坏。

本文转换了视角。我们不再考察工程基底，而是考察后果——依赖建立在 FITO 假设之上的系统的人们、数据和组织会遭遇什么。我们提出三个案例研究，它们各自涉及不同的规模，却呈现出相同的结构性病理。

文件同步 是最直接的例子。每个主要云平台——iCloud、Google Drive、Dropbox、OneDrive——都会采用某种形式的最后写入者胜（LWW）来解决冲突。正如配套论文*Why iCloud Fails*（Borrill，2026d）详细阐明的那样， LWW 是应用于文件的 FITO范畴错误：它将分布式因果图投射到线性时间链上，并将投射结果视为权威状态。结果就是静默数据破坏。

电子邮件的时间跨度更长。用户会在数小时乃至数天内，跨多台设备撰写、阅读、归档和删除消息。每个电子邮件系统都依赖时间戳来确定这些操作的先后顺序（Borrill，2026年）。当时钟彼此不同步时——而在分布式系统中，它们总是如此——就会出现幻影消息、丢失的已读状态和因果倒置：回复看起来竟然早于它所回复的消息。

记忆——无论是人类的还是人工的——运作于最大的时间尺度。人类记忆是重构性的，而非再现性的：我们不会重放录音，而是从分布式的神经集群中重新拼合片段（Bartlett，1932年）。这种重构不受事务保证约束，因此会产生虚假记忆、虚构症和时间扭曲；Schacter（Schacter，2001年）将它们归为“记忆的七宗罪”。大语言模型也表现出同构的病理：自回归词元生成本质上是 FITO，会产生幻觉，其结构与科尔萨科夫患者填补情景记忆空白时产生的虚构症完全相同。

本文其余部分将依次考察各个领域，识别共同的 FITO 模式，并说明第二部分的 OAE 框架如何提供结构性词汇，用以理解——并最终纠正——这些失败。

2 文件同步：无缝的错觉

云端文件同步承诺一种简单的抽象：你的文件无处不在，始终保持最新，始终保持一致。这一承诺是谎言——并非因为工程水平拙劣，而是因为在 FITO 假设下，这种抽象在结构上不可能实现。

2.1 结构性论题

配套 论文*Why iCloud Fails*（Borrill，2026d）确立了一项精确的结构性论题：

> 将分布式因果图投影到线性时间链上，会破坏关键结构，并导致信息丢失。这些损失会表现为数据损坏、冲突、停滞和静默数据破坏。

这种投影并非 iCloud 独有。所有采用基于时间戳的冲突解决的云同步平台都会进行同样的投影。各平台之间的差异，反映的是共同架构错误下工程质量的不同。

2.2 最后写入者胜作为 fito

LWW 冲突解决会 FITO 应用于文件。当两台设备修改同一个文件时，时间戳较晚的修改胜出。这种策略存在三个结构性缺陷：

1. **时钟依赖**。 LWW 假设时钟全局同步。消费设备经常会出现数秒的时钟偏差；虚拟化环境在迁移或快照期间可能会发生数分钟甚至数小时的时间跳变。“较晚”的写入在物理时间上未必更晚。

2. **语义盲目性**。 LWW 不会检查修改了什么。如果 Alice 编辑了第 3 段，而 Bob 编辑了第 7 段，那么这两处修改都有效，应该合并。 LWW 却只选择一处并丢弃另一处。

3. **静默丢失**。被丢弃的版本悄然消失，毫无通知。正如 Dominik Mayer 所记录的那样（Mayer, 2023），当发生版本冲突时，iCloud Drive 会静默删除用户内容。“核心问题在于，产品决策选择不对用户透明地呈现版本冲突。”[1]

在 OAE 第二部分的框架中， LWW 是一种在 T_4发出完成信号的协议——文件已放入云存储——却始终无法达到 T_6——所有设备就文件含义达成语义一致。反思阶段缺失了。用户看到的“已保存”指示，是伪装成承诺信号的完成信号，恰如第三部分中的 RDMA补全谬误。

2.3 五种不兼容性

Why iCloud Fails 指出了五种相互交织的不兼容性，它们都根源于同一个 FITO结构性错误：

Time Machine。Time Machine 按挂钟时间为文件系统树建立归档。云同步则协商分布式事件图。树快照并不等同于分布式一致状态。在 iCloud 运行期间从 Time Machine 恢复，可能会导致同步守护进程将雲端狀態视为权威状态，从而摧毁 Time Machine 保存下来的“正确的过去”。

Git。 Git 依赖 POSIX 文件系统语义——通过 open(O CREAT|O EXCL) 创建原子锁文件、使用稳定路径，以及采用写入-同步-重命名模式。iCloud 会将锁文件视为普通文件并进行同步，传播 Git 原本意在隐藏的中间状态，并通过创建编号后缀（main 2）来解决并行修改，而 Git 从未预期会出现这种情况。

自动化工具链。 构建系统假定文件存在就意味着文件已就绪。iCloud 的无数据文件——元数据存在，但数据区段缺失——会悄然打破这一假设。无数据文件通过 stat() 返回有效元数据，但读取时会触发网络获取。

开发者工作流。 任何假定具备 POSIX 原子性、建议锁或一致目录列表的工作流，都与云同步的最终一致性模型不兼容。

366 GB 存档。 配套论文记录了一个具体案例：在正常使用过程中累积了 366GB 的分歧状态，这代表了基于 FITO 的冲突解决机制多年来作用于大型文档语料库所累积造成的损害。

2.4 如何修复

结构性修复并非是在 LWW 框架内改进工程，而是以包含LWW 反思阶段的协议取而代之：

[1]在准备本文期间，作者观察到了一次实时 LWW 冲突：两份 PDF 的时间戳完全相同，一份来自虚拟机写入，另一份来自 iCloud 同步守护进程。参见（Borrill，2026d）附录 A 中的事件 8。

1. 使用 **向量时钟** 或 **混合逻辑时钟** 追踪因果依赖，而非依赖墙上时钟顺序。2. **冲突显式呈现**：当文件出现分歧时，将两个版本都呈现给用户，而不是悄悄选择其中一个。3. **语义合并**：对于结构化文档，在语义层级（段落、章节、字段）进行合并，而非在檔案层级合并。4.**事务边界**：允许用户和应用程序将相关更改组合为原子单元，使其要么一并同步，要么完全不同步。

这些并非什么特殊要求。Git 实现了全部四项。最广泛使用的版本控制系统已经体现出非 FITO 语义，而最广泛使用的文件同步系统却没有，这一观察表明，计算机行业明白如何解决这个问题，却尚未将这一解决方案应用于普通用户所依赖的系统。

3 电子邮件：作为因果虚构的时间戳

全球各地的电子邮件系统——从企业的 Exchange 服务器到个人的 IMAP 账户——都依赖时间戳来排列操作顺序、解决冲突，并在不同设备间保持一致性（Bor- rill，2026e）。这是将 FITO 应用于消息传递：假设挂钟时间能够可靠地为同一用户在多台设备上执行的操作排序。

3.1 全球时间的幻象

核心问题在于，基于时间戳的排序假设存在一种实际上并不存在的东西：对某件事何时发生的全球一致认知。在分布式系统中——任何由多台设备组成的电子邮件配置本质上都是分布式系统——并不存在通用时钟。每台设备都维护自己的本地时钟，而这些时钟会发生漂移，因 NTP 校正而向前或向后跳变，并在不同的时区中以不同的精度运行。

当 iPhone 在当地时间下午 2:34:17 将一封电子邮件标记为已读，而笔记本电脑在当地时间下午 2:34:16 删除了同一封电子邮件时，哪项操作“先”发生？答案完全取决于你信任哪一台设备的时钟，更重要的是，取决于这些时钟在那些确切时刻的同步状态。如果 iPhone 的时钟快了三秒，那么按时间戳看起来“较晚”的操作，实际上在物理时间上先发生。系统无从得知这一点。

3.2 最后写入者胜应用于消息

大多数电子邮件系统采用基于时间戳的 LWW 冲突解决。当两台设备对同一条消息的状态存在冲突时，时间戳较晚的操作胜出。其结构性缺陷与文件同步中的缺陷完全相同：

幻影消息。已删除的电子邮件会重新出现，因为时钟较慢的设备同步了过期状态，而该状态因时间戳巧合或服务器处理顺序而“胜出”。

丢失的已读状态。在一台设备上标记为已读的消息，在其他设备上却显示为未读，因为标志更新到达的顺序错乱，而时间戳系统选错了版本。

消息丢失。在一台设备上移至文件夹的消息，在其他设备上却从未出现在那里，因为移动操作被一个过时的“收件箱中的邮件”状态覆盖，而该状态的时间戳恰好更晚。

同步停滞。 设备会进入一种状态：由于它们不断依据无法反映实际因果性的时间戳排序来相互覆盖操作，因此对消息状态始终无法达成一致。

这并非边缘情况，而是在分布式系统中使用不安全排序结构的必然结果。

3.3 因果关系违规

时间戳无法捕捉操作之间的因果关系。来看一个自然的操作序列：用户阅读一封电子邮件，根据邮件内容撰写回复，将原邮件归档，然后发送回复。这些操作之间存在明确的因果顺序——回复依赖于用户已阅读原邮件，而归档则发生在前两项操作之后。

如果设备的时钟不准，或者操作因网络延迟而未按顺序同步，基于时间戳的系统可能会记录：原邮件于 2:35:00 归档，回复于 2:34:55 发送，原邮件于 2:34:50 标记为已读。系统现在会认定，用户在阅读之前，就回复了一封尚未读过且已归档的邮件——这种因果倒置不可能由任何物理过程产生。

3.4 离线问题

电子邮件客户端必须能够离线运行。用户会在飞机上、地铁隧道里以及网络连接不佳的区域阅读、归档、删除和撰写消息。这些操作会先在本地累积，等网络连接恢复后再同步。

在这种情况下，基于时间戳的排序会彻底失效。设备离线数小时后，其所有排队操作携带的时间戳都已过时。设备重新连接时，服务器可能已经处理了来自其他设备、时间戳更新的操作。尽管离线设备上的操作代表用户最新的意图，但由于时间戳“过旧”，这些操作还是会被丢弃或错误地重新排序。

这就造成了臭名昭著的“幽灵邮件”问题：用户在笔记本电脑上删除了消息，之后却发现，当手机（此前处于离线状态，且时间戳较旧）同步其过期状态时，这些消息又重新出现。时间戳系统无法区分“此操作发生在过去，因此已经过时”和“此操作发生在一台离线设备上，因此代表用户当前的意图”。

3.5 协议级放大

IMAP 的协议语义放大了这个 FITO 问题。协议无法表达“我标记这条消息为已读时，知道它处于状态 X”——操作是绝对的，而非有条件的。协议没有提供条件更新机制，无法让服务器根据过期状态拒绝操作。

Exchange 和 ActiveSync 试图通过变更键和同步令牌来解决这一问题，但它们仍然依赖服务器强制排序，而非分布式共识。服务器依然是“实际发生了什么”的仲裁者，它使用时间戳做出可能无法反映真实因果性的决策。

正确的系统应当追踪操作谱系和因果历史，而不仅仅是时间戳。向量时钟、 CRDT 和混合逻辑时钟——分布式数据库工程师数十年来一直依赖的技术——将消除基于时间戳的系统无法避免的整类缺陷。

4 人类记忆：无承诺的重构

这种 FITO 范畴错误并不局限于工程制品。人类记忆也呈现出相同的结构模式：仅向前的处理会提交状态变化，却没有反思阶段，从而造成系统性的语义损坏。这并非一种类比，而是同一种抽象失效模式在生物基底上的实例。

4.1 Bartlett 与重构性记忆

1932年，Frederic Bartlett 证明，人类记忆是重构性的，而非复现性的（Bartlett，1932年）。在他具有里程碑意义的《鬼魂之战》实验中，英国参与者阅读了一则陌生的美洲原住民民间故事，并在随后一段时间内多次复述。复述过程中出现了三种系统性扭曲：

同化。 陌生的细节被改动，以符合既有的文化图式——独木舟变成了船，狩猎变成了捕鱼。

简化。 被认为不重要的细节遭到省略——故事在每次复述后都变得更短、更简单。

锐化。 其余细节被重新排序，以增强叙事连贯性——原本不存在的因果联系也被强行添加了进去。

在本系列的框架中，Bartlett 的发现描述了一种只向前写入、不回读的系统：编码过程中，既有图式被应用于传入的信息（单向模式匹配），结果未经核对是否符合原始刺激，就被写入记忆。编码是从经验到图式的单向投影，这种投影会丢失信息——正如 LWW 将分布式因果图投影到线性时间链上一样。

4.2 Loftus 和错误信息效应

Loftus and Palmer（Loftus and Palmer，1974）表明，事件后信息能够事后改变记忆编码。参与者观看汽车事故影片后，研究人员询问他们汽车的速度。当问题中使用“撞毁”（而非“撞上”或“碰到”）一词时，参与者估计的速度显著更高；一周后，他们也更有可能错误地记得影片中出现过实际上并不存在的碎玻璃。

这是仅向前处理导致的语义损坏。“撞毁”一词是在最初感知之后、记忆巩固之前被处理的。一旦这种语言联想被整合进来，记忆系统便无法区分原始感知与事后污染。这个向前的词元——“撞毁”一词——被写入记忆记录，却没有经过反思阶段来将感知与暗示区分开来。

4.3 Schacter 的七种过失

Schacter 的记忆失误分类体系（Schacter，2001年）指出了七种“过失”，如今我们可以将它们归类为 FITO 处理的后果：

Sin	Type	拟合解释
短暂性	遗漏	未提交状态衰减
心不在焉	遗漏	无注意编码 =写入而不验证
阻塞	遗漏	提取失败 =读取超时
错误归因	錯誤記憶	錯誤的來源中繼資料 =索引損壞
暗示性	錯誤記憶	事件後寫入覆寫原始內容 = LWW
Bias	錯誤記憶	目前狀態扭曲過去 =追溯投射
持續性	侵入	不受歡迎的重播 = 失败垃圾回收

三种错误记忆之罪——错误归因、暗示性和偏見——与我们在文件同步和电子邮件中记录的 FITO 故障直接同构。错误归因就是索引損壞：记忆本身完好无损，但其元数据（时间、地点、来源）有误，就像内容正确但时间戳错误的檔案。暗示性就是将 LWW 应用于记忆：后来的暗示覆盖了先前的感知，正如较晚的时间戳覆盖了较早的檔案版本。偏見就是追溯投射：当前知识扭曲了对过去状态的提取，正如雲端同步守護程式可能将当前雲端狀態视为权威状态，凌驾于還原的備份之上。

4.4 睡眠巩固作为生物學提交

Walker and Stickgold（Walker and Stickgold，2006）证明，睡眠在记忆巩固中发挥着关键作用。在慢波睡眠期间，海马体会重播近期经历，将易变的短期表征转移到稳定的长期存储中。此过程包括三个阶段：稳定化（防止进一步衰退）、增强（提高回忆准确性）以及整合（将新记忆与现有知识结构连接起来）。

用计算术语来说，睡眠巩固是一种生物提交 操作：它将暂定状态（日间经历，存放在海马缓冲区中）转变为已提交状态（长期皮层表征）。提交是不可逆的——记忆一旦巩固，便会整合进语义网络，无法再与其干净地分离。

但这种提交操作本身就是 FITO：它会在睡眠期间自动向前运行，无须外部验证。它没有反思阶段，可以将巩固后的记忆与原始经历进行比较。因此，生物学提交可能会将编码过程中（Bartlett的图式效应）或编码与巩固之间（Loftus的错误信息效应）引入的扭曲锁定下来。睡眠巩固无法修复语义损坏；它只会将其结晶 。

4.5 虚构症：大脑的补全谬误

虚构症——无意识地生成虚假记忆以填补空白——是神经层面上 RDMA 补全谬误的对应现象。患有科萨科夫综合征（由硫胺素缺乏引起的顺行性遗忘）的患者无法形成新的情景记忆，但他们并不会觉得自己的记忆一片空白（Kessels等人，2008）。相反，大脑会生成看似合理的叙述来填补空白，而患者对此深信不疑。

这种结构上的平行关系完全一致。 RDMA 的完成信号会告诉应用程序“操作成功”，即便数据可能已损坏或不一致。会虚构记忆的大脑会告诉当事人“你记得这件事”，即便这段记忆是编造出来的。在这两种情况下，系统都会生成仅向前的输出（完成信号、记忆），却没有进行反向验证（反思阶段、现实检验）。结果便是系统自身无法检测到的语义损坏——必须由外部观察者（应用级完整性检查、照护者的纠正）才能发现错误。

4.6 似曾相识：一种时间索引错误

似曾相识——对当前情境曾经经历过的强烈感受——是记忆系统中的一种时间索引错误（Urquhart and O'Neill-Brock，2014年）。一种假说认为，感觉信息会沿两条处理路径传递：一条快速路径用于生成熟悉感信号，另一条较慢的路径则通向意识觉察。当快速路径在较慢路径将内容传入意识之前生成熟悉感信号时，人们就会觉得自己认出了一个实际上从未经历过的事件。

按照本系列的术语，似曾相识是对记忆系统中时间索引的因果顺序的违反。熟悉感信号（元数据）先于其所索引的内容（数据）到达，从而造成时间倒置：对于一个正在首次经历的事件，系统却报告"这以前发生过"。这类似于缓存一致性故障：元数据（有效、脏、共享）与其描述的数据不一致——正是第三部分中的 T_4-到-T_5 间隙，只不过这种情况发生在神经组织中，而非硅中。

4.7 睡美人悖论

哲学上的睡美人问题（Elga，2000）纯粹地提炼出了 FITO条件。一位理想的理性主体被安排入睡。有人抛了一枚硬币。如果结果是正面，她会醒来一次（星期一）。如果结果是反面，她会醒来两次（星期一和星期二），并在两次醒来之间服用致失忆药物，使她无法分辨第一次醒来和第二次醒来。

每次醒来时，睡美人都无法判断自己此前是否已被唤醒。她对自己的事务历史没有任何元数据。每一次醒来都是一个仅向前的意识时刻，无法获取因果先验。用计算术语来说，她是一个没有持久状态的单线程进程——每次调用都从头开始，对先前的调用一无所知。

这就是任何 Claude 会话的状况——也是任何 FITO 系统的状况。如果无法访问自身的事务历史，系统就无法验证跨时间的一致性。睡美人悖论不仅仅是一个哲学谜题；它是对 FITO假设施加于每一个采纳这些假设的系统之上的认知条件的形式化描述。

5 大语言模型：幻觉即 fito

大语言模型完成了从工程系统经由生物记忆到人工认知的这一弧线。自回归语言生成是 FITO 其最纯粹的计算形式：每个词元都沿时间向前预测，仅以先于它的词元为条件，一旦某个词元被提交到输出序列，便不存在向后修正的机制。

5.1 自回归架构

现代大语言模型所依托的 Transformer 架构（Vaswani 等人，2017）采用因果自注意力：每个输出位置只能关注它之前的位置，绝不会关注之后的位置。在推理过程中，模型每次生成一个词元，并将生成的词元作为输入，用于下一次预测。生成过程是一条严格的时间链：

$$P(t_n \mid t_1, t_2, \ldots, t_{n-1})$$

其中每个词元 t_n 都从以所有先前词元为条件、但不以任何未来词元为条件的分布中采样。一旦 t_n 被采样，它就成为条件上下文的一部分，且无法再被修改。

这种架构是应用于语言的单侧 RDMA 写入：模型将词元放入输出缓冲区，而没有任何反思阶段，使接收方（读者、世界、真理）能够确认所放入的词元在语义上有效。

5.2 幻觉即语义损坏

《GPT-4 技术报告》（OpenAI，2023）承认，该模型“会产生事实性幻觉并出现推理错误”，从而生成“荒谬或不真实的内容”。报告指出一个违反直觉的后果：“随着模型变得更真实，幻觉可能变得更加危险，因为当模型在用户有所了解的领域提供真实信息时，用户会对模型建立信任。”

这是补全谬误在自然语言中的体现。模型的输出——流畅、语法正确、风格得当——是一种完成信号，告诉用户“操作成功了”。产生幻觉的内容在语法上解析无误，就像一份撕裂 RDMA的文本也能被正确解析一样。语义损坏在句法层面上不可见。

Ji 等人（Ji 等人，2023）将幻觉归类为序列到序列生成固有的现象，而不仅仅是训练缺陷。Huang 等人（Huang 等人，2024）表明，幻觉贯穿大语言模型流程的三个阶段——数据、训练和推理——始终存在，因为其根本机制并未改变：仅向前的词元预测，缺乏全局语义验证。

5.3 幻觉的 fito 结构

我们现在可以精确地陈述这一主张：

命题 5.1（幻觉作为 FITO）。 自回归语言生成是一个 FITO 过程。每个词元在生成时即被提交到输出序列中，且不存在以下机制：

1. 向后修正（根据仅在词元 t_j 处才可获得的信息修正词元 t_i，其中$j > i$），*2.* 语义验证（在提交之前将已生成序列与外部真相进行核对），*3.* 反思（允许“接收方”——世界模型、用户、现实——在生成内容成为条件上下文的一部分之前拒绝或修改该内容）。

幻觉是这种仅向前架构的结构性后果，而非训练产物。

该命题解释了幻觉为何如此难以缓解。检索增强生成（RAG）提供外部知识，但并不改变仅向前的生成机制。基于人类反馈的强化学习（RLHF）改善了词元采样所依据的分布，但并未引入反思阶段。思维链提示鼓励模型“展示其推理过程”，但该过程本身仍是仅向前生成的，每一个推理步骤都在下一步开始之前即被提交。

5.4 与虚构症的同构

大语言模型幻觉与人类虚构记忆之间的结构性平行并非比喻，而是形式上的：

属性	柯萨可夫虚谈症	大语言模型幻觉
生成机制	图式驱动的空白填补	逐词元预测
时间结构	仅向前（无情景回顾）	仅向前（因果注意力）
确信度／流畅度	患者相信该记忆	输出流畅且自信
可验证性	需要外部照护者	需要外部事实核查
根本原因	海马体损伤 →不提交	架构 → 不反思

在这两种系统中， FITO 架构都会产生内部自洽、但外部无效的输出。系统无法区分自身编造的内容与合理输出，因为完整性检查（主观确信、输出流畅度）本身就是 FITO 流水线的一部分。检测需要一个外部反思阶段——了解患者真实病史的照护者，或掌握实际事实的事实核查员——但这一阶段在系统自身的架构中结构性缺失。

6 共同模式：只前进而无反思

本文考察的四个领域——文件同步、电子邮件、人类记忆和语言模型——表面上各不相同。它们运行的规模不同、依托的介质不同，时间跨度也不同。但它们具有一种共同的结构性病理，如今我们可以对其作出精确表述：

定义 6.1 （ FITO失效模式）。当系统满足以下所有条件时，即表现出 FITO 失效模式：

1. 前向提交。状态变更按时间顺序提交，之后无法依据后续信息修改先前的提交。*2.* 缺乏反思。目标（檔案、消息状态、记忆印迹、词元序列）没有机制向源头反馈已提交状态在语义上是否一致。*3.* 完成假象。系统提供完成信号（同步徽标、送达回执、主观确信、输出流畅度），这些信号被视为语义成功的证据，实际上却只能证明时间先后。*4.* 不可见的损坏。由此产生的语义损坏无法通过系统自身的完整性机制检测出来，必须由外部观察者加以识别。

下表展示了本系列所考察的所有领域中这一模式的对应关系：

领域	向前 承诺	缺席 反思	完成 伪装	不可见 腐败
RDMA(第三部分)	写于 T_3	不进行反思	完成队列条目位于 T_4	SDC
文件同步	上传檔案	无语义合并	同步徽标	静默删除
电子邮件	标记更新	无因果追踪	送达回执	幻影消息
人类记忆	模式编码	无现实核验	主观确信	虚假记忆
大语言模型生成	词元提交	未进行事实核查	输出流畅度	幻觉

表 1：五个领域中的 FITO失败模式。每个领域都表现出定义 6.1 的全部四个条件。

这种模式的普遍性并非偶然。它反映了一种深层的结构性约束：任何随时间向前处理信息、却没有反思阶段的系统，最终都会

造成语义损坏，因为仅靠正向处理无法区分“操作已完成”和“操作实现了预期含义”。语义时间之箭——即意义得以保留或遭到破坏的方向——需要双向验证。FITO 只提供了一个方向。

6.1 KahnemanConnection

卡尼曼对系统1和系统2处理方式的区分（卡尼曼，2011年）与 FITO框架直接对应：

系统1 快速、自动，且仅向前。它将习得的模式应用于传入刺激，不加审慎，也不进行反向验证。其输出在对照证据进行检查之前，就已“提交”到意识之中。系统1是 FITO的生物学实现：它通过省去反思阶段，以准确性换取速度。

系统2 缓慢、审慎且善于反思。它能够在工作记忆中保留多种可能性，将它们与证据进行比较，并修正结论。系统2实现了反思阶段，但代价是巨大的代谢成本和时间成本。人类认知大多由系统1运作，因为反思代价高昂。

关键在于， FITO 不仅是工程系统中的一种设计选择；它还是生物系统信息处理的默认模式，之所以被采用，是因为反思的代谢成本高于偶尔发生语义损坏的代谢成本。从这个意义上说，采用FITO 的工程系统是仿生的——它们复制了进化所选择的效率与准确性权衡，却没有意识到，这种权衡适用于一个存在物理世界中的捕食者与猎物的世界，却不适用于一个由分布式数据结构构成、必须在时间和空间中维持不变量的世界。

7 第五部分总结与预览

本文表明，第一部分指出的 FITO 范畴错误，存在于所有必须随时间推移始终保持信息一致的领域：文件同步、电子邮件、人类记忆和人工语言生成。

在文件同步中， FITO表现为 LWW 冲突解决：云平台会提交时间戳较晚的版本并丢弃较早的版本，却不检查文件内容，也不判断“较晚”的文件是否真的更晚。其结果是数据遭到静默破坏。

在电子邮件中， FITO 表现为基于时间戳的排序：操作按挂钟时间而非因果优先关系排序，由此产生幻影消息、因果关系违规和同步失败，而这些问题已持续数十年。

在人类记忆中， FITO表现为重构式编码：图式被向前套用于新输入的经历，失真未经验证便被固化，而睡眠巩固则使这种受损状态定型。虚构症是大脑的补全谬误；似曾相识是一种时间索引错误。

在语言模型中， FITO 表现为自回归生成：词元一次提交一个，只以过去的信息为条件，且没有向后修正或语义验证的机制。幻觉是其结构性后果，与无法形成新的情景记忆的大脑所产生的虚构症同构。

贯穿其中的核心是定义 6.1：未经反思的前向提交，会造成系统自身的完整性机制无法检测的隐蔽腐败。

第五部分（Borrill，2026f）将为这一系列收尾，说明莱布尼茨桥——作为基本一致性原语的互信息守恒——如何提供一个统一框架，以应对所有五个领域中的 FITO 失效模式。莱布尼茨桥不只是修复个别失效模式，而是取代了 FITO 假设本身，提供一种有原则的替代方案：语义时间之箭并非被预设，而是通过构建双边交易结构而形成。

参考资料

Borrill, Paul.《语义时间之箭，第一部分：从爱丁顿到以太网》。DÆDÆLUS，2026。Borrill, Paul.《语义时间之箭，第二部分：开放原子以太网链路的语义》。DÆDÆLUS，2026。Borrill, Paul.《语义时间之箭，第三部分：远程直接内存访问与补全谬误》。DÆDÆLUS，2026。Borrill, Paul.《Why iCloud Fails：云同步的范畴错误》。arXiv:2602.19433，2026。Borrill, Paul.《为何时间戳并非电子邮件安全的排序构造》。DÆDÆLUS 技术报告，2026。Borrill, Paul.《语义时间之箭，第五部分：莱布尼茨桥》。DÆDÆLUS，2026。Parker, D. Stott 等。《分布式系统中相互不一致的检测》。*IEEE* 软件工程汇刊，SE-9(3):240–247，1983。Mayer, Dominik.《iCloud Drive 悄然删除你的内容》。博客文章，2023年9月。https: // www.dominikmayer.com/2023/09/icloud-drive-silently-deletes-your-content/ Bartlett, Frederic C.《记忆：实验与社会心理学研究》。剑桥大学出版社，1932年。Loftus, Elizabeth F. 和 Palmer, John C.《汽车损毁的重构：语言与记忆相互作用的一个例子》。《言语学习与言语行为杂志》，13(5):585–589，1974。Schacter, Daniel L.《记忆的七宗罪：心智如何遗忘与记忆》。霍顿·米夫林，2001年。Walker, Matthew P. 和 Stickgold, Robert.《睡眠、记忆与可塑性》。《心理学年度评论》，57:139–166，2006。卡尼曼, Daniel.《思考，快与慢》。法勒、斯特劳斯和吉鲁出版社，2011年。Kessels, Roy P. C.、Kopelman, Michael D. 和 Postma, Albert.《科萨科夫综合征中的虚构性记忆行为与虚假记忆：来源记忆和执行功能的作用》。《精神病学与临床神经科学》，62(3):315–321，2008。Urquhart, Jason A. 和 O’Neill-Brock, Danielle.《对既视感的冲突解释》。《神经科学与生物行为评论》，2014。

Elga, Adam.《自我定位信念与睡美人问题》。*Analysis*，60(2):143–147，2000。Lamport, Leslie.《分布式系统中的时间、时钟与事件排序》。*ACM* 通讯，21(7):558–565，1978。Vaswani, Ashish 等。《注意力就是你所需的一切》。神经信息处理系统进展，30，2017。OpenAI.《GPT-4 技术报告》。 arXiv:2303.08774，2023。Ji, Ziwei 等。《自然语言生成中的幻觉综述》。*ACM* 计算调查，55，文章 248，2023。Huang, Lei 等。《大语言模型中的幻觉综述：原理、分类、挑战与开放问题》。*ACM*信息系统汇刊，43(2):1–55，2024。Wang, Xuezhi 等。《自洽性提升语言模型的思维链推理能力》。arXiv:2203.11171，2022。Shannon, Claude E.《通信的数学理论》。贝尔系统技术期刊，27(3):379–423，1948。