

Token 的阿凡达们

一个词在六个世界里的变身

竹小竹8167

费曼式写法 · 由多个 AI 代理撰写与互相审校

导读

一个词在六个世界里的变身

这个月你为 ChatGPT 的 API 付了一笔账单，计价单位是 token。上班刷卡进楼，闸机认的是 token。你是个程序员，上午刚调试完编译器的报错，那里面最小的零件也叫 token。中午刷到一个新闻：某某 token 在交易所暴涨。

四个 token，四个世界。是巧合，还是同一盏灯的四道光？

这本书借侯世达《表象与本质》的方法——他写「影子」怎么从地上的暗影，一路蔓延成「形影不离」「影子内阁」「心理阴影」——来追踪 token 这个概念的八次化身。你会发现它有一套固定动作：每降落一个新世界，就换上一具当地躯体，可灵魂始终没变：**一个东西站出来，代表另一个东西；因为验证替身，永远比验证正主便宜。**

我们从游戏厅哐当一声掉出来的铜板出发，路过门禁的通行证、编译器的词法刀、搜索引擎的指纹、大模型嚼碎的统计残字、区块链上的资产契据，最后停在收银台前——看文本碎片如何变成智能时代的计价单位。九章之后，你手里会多出一个趁手的工具：下次在任何领域撞见 token，问三个问题就知道它是谁——**它替谁站岗？谁铸的？兑什么现？**

第一章先到潘多拉星球——先弄明白「阿凡达」这个比喻，为什么比「引申义」三个字好使得多。

第一章 · 一个词的影子

先不急着想 token。先说一个看起来毫不相干的词：**影子**。

认知科学家侯世达（Douglas Hofstadter）在《表象与本质》里做过一件很好玩的事：他带着读者数「影子」这个词到底有多少种用法。一开始你以为它只有一个意思——光被挡住之后投在地上的那片暗。然后你开始数：

- 「形影不离」——影子成了**甩不掉的伴随者**；
- 「影子内阁」——影子成了**没有实权、只在一旁对照着的副本**；
- 「他影子都没见到」——影子成了**最轻微的一点痕迹**；
- 「这事给我留下了心理阴影」——影子干脆搬进脑子里，成了**一段挥不去的记忆**。

数到这里你会发现一件微妙的事：这些用法你全都秒懂，从来没有人教过你「影子的第十三条引申义」。你脑子里装的根本不是一份清单，而是一团**光晕**——中间是实体的暗影，往外一圈一圈地淡下去：伴随、副本、痕迹、心理残留……每一个新用法，都是把旧的影子**类比**（analogy，就是「这个跟那个在骨子里是同一回事」的判断）到一个新场景里投了一遍。

大白话

大白话：概念不是词典里的一排定义，而更像一盏灯。灯芯是那个最具体的画面（地上的暗影），灯光能照多远，取决于你能把它类比到多远的地方。侯世达的结论很猛：**类比不是修辞的装饰，类比就是思考本身**。概念靠类比长大。

token 的六张脸

现在把同样的方法用在 token 上。你随便抓一个软件工程师问「token 是什么」，他的答案取决于他昨天在干什么活：

- 做搜索的会说：token 是**文本切出来的特征**，用来撮合查询和文档；
- 做大模型的会说：token 是**语料里统计出来的常见字符序列**，可能是半个词，API 账单按它收钱；

- 写编译器的会说：token 是**词法分析的产物**，一个有类型、有内容的小对象；
- 搞安全的会说：token 是**一张凭证**，代表「你有权做某个操作」；
- 炒币的会说：token 是**链上的一种资产**；
- 而词典会告诉你，token 的本义是**标志、凭证、纪念品**。

同一个单词，六张脸。这就有意思了：它们是一个词的六种巧合（就像「杜鹃」既是鸟又是花），还是同一盏灯照出来的六道光？

阿凡达：一个比「引申义」更好的词

这本书借用电影《阿凡达》的设定来讲这件事。电影里，人类躺在链接舱里，意识投射到潘多拉星球的纳美人身体里——同一个灵魂，换了一具适合当地环境的躯体。化身阿凡达（avatar，梵语里「天神下凡的化身」，在互联网里指「你在虚拟世界里的替身」）这个词精准地描述了概念的成长方式：

大白话

大白话：token 这个概念的「灵魂」一直没换——**一个东西站出来，代表另一个东西**。但它每降落到一个新的技术世界，就换上一具当地的躯体：在搜索引擎里它是特征切片，在编译器里它是带类型的符号，在闸机口它是通行证，在账单上它是计价单位。六个阿凡达，一个灵魂。

所以这本书的读法是这样的：我们先回到灯芯——token 最古老、最具体的那张脸（游戏厅里哐当一声掉出来的铜牌）；然后一章一个化身，看它每降落一个新世界，躯体怎么变、灵魂怎么不变；最后一章把六个阿凡达并排摆开，看看它们共享的那张脸到底长什么样。

读完全书，你应该能回答这个贯穿始终的问题：**为什么这些行业不约而同地抓起了同一个词——它们各自漏掉了什么，又各自保留了什么？** 以及一个更大的问题：如果概念都靠类比长大，那「理解一个新概念」这件事本身，到底是怎么发生的？

下一章，先去游戏厅。

第二章 · 游戏厅里的铜牌

上世纪八九十年代的游戏厅里，你要玩《街霸》，得先去柜台换一把黄铜色的小圆片。投进机器，哐当一声，机器才肯理你。

这片小铜板就是 token 最古老的肉身。英文词典给 token 的本义是**标志、凭证、象征物**——它本身不值钱（一块黄铜而已），但它**代表**了一次游戏的机会。柜台收了你一块钱，把「玩一次」这个权利铸进铜片里，机器只认铜片不认人。

替身要满足的三个条件

仔细拆解这片铜板，你会发现一个合格的「替身」必须同时满足三件事：

- **可铸造**：得有个权威把它发出来。柜台能发，你家后院不能发——机器认得出柜台那批铜片的重量和齿纹。
- **可流通**：它能从柜台到你手里、再到机器肚子里，一路换手而意义不变。你甚至可以把它塞给旁边的小孩，权利跟着铜片走。
- **可兑现**：机器见了它，真的让你玩一次。替身背后得有个「正主」兜底——铜片代表的那次游戏机会是真实存在的。

大白话

大白话：token 的灵魂在这一章就全部登场了——**一个东西站出来，代表另一个东西**。它自己不重要，它背后的「正主」才重要；但正主太大、太远、太抽象，没法直接递来递去，所以铸一个小替身出来，替正主流通。

注意这个设计有多聪明：游戏机不需要懂「一块钱」，也不需要联网查你的账户。它只需要一个机械动作——掂一掂、数一个齿——就完成了授权。**验证替身的成本，远远低于验证正主的成本**。这是 token 千年不倒的根本原因，后面每一章都会回响这句话。

从铜板到令牌

英语世界里，这种替身有一个庞大的家族：地铁票、投币电话里的硬币替代品、赌场筹码、签到时挂的牌子、国王赐给使节的信物（*talisman* 式的 *token of authority*）……它们都是同一个设计：**把一项不好携带的权利，铸成一个好携带的物件。**

计算机行业第一次大规模借用这个词，是 1980 年代的令牌环网（Token Ring，一种老式局域网：一圈电脑连成环，谁手里握着一个叫「令牌」的特殊数据包，谁才有资格发言）。你看，连隐喻都没换——令牌就是游戏厅铜片的电子版：

- 可铸造：网络协议规定令牌长什么样，造不出假的；
- 可流通：令牌在环上一台台传下去，像击鼓传花；
- 可兑现：握到令牌的机器，真的获得发言权。

而且连那个根本原因也照搬了：每台电脑不用去质问「你凭什么发言」（那会吵成一团），只看「令牌在不在你手里」——验证替身，不验证正主。

本章留下的问题

到这里，*token* 的意思很干净：**凭证、替身、代表权利的信物**。但如果你现在去问一个写代码的人，他嘴里的 *token* 多半是「分词切出来的字符串」；去问用 ChatGPT 的人，*token* 是账单上的计价单位。**凭证**和**字符串**，听起来八竿子打不着。

这正是概念迁移最有戏的地方：一个以「代表权利」起家的词，是怎么一步步变成「代表一段文本」的？中间不是断裂，而是一次次像「影子」那样的类比跳跃，每一跳都扔掉了一些东西，也保留了一些东西。

下一章先看跳得最近的一步：保安手里的通行证——那里的 *token* 还在做凭证，只是凭证代表的东西，从「一枚硬币」变成了「你这个人」。

第三章 · 门卫手里的通行证

你早上刷工牌进办公楼，闸机嘀一声放行。注意闸机做了什么、没做什么：它**没有**重新核实你是谁——没有查你的身份证，没有打电话给人事，没有问你上司。它只是读了一张卡，卡说「这人已通过验证」，它就信了。

这就是安全领域的 token，词典里那条「代表执行某个操作的权利」的定义，在这里几乎原封不动。它是所有阿凡达里离游戏厅铜板最近的一个——只不过铜板代表的权利是「玩一次游戏」，通行证代表的权利是「以你的身份做事」。

为什么不每次都验正主

设想一个不用 token 的世界：你在网站上每点一个链接，都要重新输一遍用户名和密码。烦还在其次，真正的灾难是**密码满天飞**——它在你电脑里存了无数份，在网络上跑了无数趟，每趟都可能被截获。正主（密码）太贵重也太脆弱，经不起这么折腾。

于是工程上造了个替身，流程和游戏厅一模一样：

- **铸造**：你输一次密码（验正主，只此一次），服务器确认无误后，铸出一枚 token 发给你——通常是一串长长的随机字符，存在浏览器里。
- **流通**：之后你的每次请求都捎上这串字符，像把工牌贴在胸前。
- **兑现**：服务器看到这串字符，认出是自己铸的，就按你的身份办事——不再问密码。

大白话

大白话：登录一次拿到的 token，就是服务器发你的临时工牌。服务器只认牌不认脸——所以丢工牌（token 泄露）比你想象的严重，谁捡到都能刷进门。这也是为什么 token 都有有效期：工牌过夜作废，丢了也只损失一小段窗口。

令牌自己会说话：JWT

传统 token 有个小麻烦：服务器收到一串随机字符，得回头查账本（数据库）才知道这牌发给了谁、灵没灵验。每次请求查一次账，量大时账本先被压垮。

于是有了更精致的化身 JWT (JSON Web Token, 一种把「你是谁、你有什么权限、什么时候过期」直接写进令牌本身、再加防伪签名的通行证)。它像一张自带照片和公章的证件：服务器不用查账本，只要验公章（签名）没被动过，证件上写的话就可信。

这又是一跳类比：从「随机号码牌，凭号查账」跳到了「自带内容的防伪证件」。token 开始往「携带信息」的方向走——记住这个方向，第四、五、六章里它会成为主角。

硬件 token：把替身铸进铁里

安全领域还有一个返祖的分支：银行发的 U 盾、动态口令器、插在电脑上的安全密钥。它们干脆把 token 做成实体——字面意义回到了游戏厅铜板。网银转账时，光知道密码不够，你手里还得真的捏着那块小硬件。「你有的东西」（硬件）加「你知道的东西」（密码），两样凑齐才算你本人。这块铁代表的，仍然是那条古老的定义：**执行某个操作的权利**。

本章留下的问题

这一章的 token 几乎没变形：还是替身，还是凭证，还是「验证替身比验证正主便宜」。但注意 JWT 露出的那条缝——token 内部开始**装载信息**了，而不只是「代表」信息。

下一章里，token 将纵身跳进一个完全不同的世界：编译器。在那里它不再代表「权利」，而是代表「一段字符是什么」——比如这串字符是个数字，那串是个名字。凭证的色彩淡了，**特征**的色彩浓了。可你会看到，三条件一个都没丢，只是换了主语。

第四章 · 编译器的第一道刀

你写下一行代码：

```
total = price * 0.13
```

在你眼里，这是三个名字、两个符号、一个数字。但在编译器眼里，它一开始什么都不是——只是十四个字符排成的一列：t、o、t、a、l、空格、= 像一长串没断句的电报。

编译器干活的第一步，就是把这串电码切成段、贴上标签。这一步叫词法分析（lexical analysis，也叫 tokenization：照着一套规则，把字符流切成一个个有意义的最小单位），切出来的每一段就是一个 token。Martin Fowler 在《领域特定语言》里给它的定义干净利落：**每个 token 是一个小对象，只有两个属性——类型（type）和内容（payload）**。上面那行代码切完，大概是这样一排：

- 类型「标识符」，内容 total
- 类型「赋值号」，内容 =
- 类型「标识符」，内容 price
- 类型「乘号」，内容 *
- 类型「数字」，内容 0.13

为什么非要先切一刀

你可能会问：编译器直接读字符不行吗，为什么非要先过一遍这道刀？

因为不切断句子，后面的活没法干。设想你要分析「price 乘 0.13」的结构，就得先知道哪些字符算一伙的：空格和乘号是边界，p-r-i-c-e 五个字母要抱成一团当个整体看。词法分析就是把「字符的河流」冻成「符号的冰块」——下游的语法分析（parsing，检查这些符号按语法能不能组成合法结构的那一步）只需要面对一小盒冰块，而不用在河里游泳。

大白话：这就像你听别人说话。对方嘴里出来的是一条连续的气流，但你大脑会先把气流切成一个一个词，然后才谈得上理解句子。词法分析器就是编译器的「耳朵」：先切词，再理解。

两个属性里的讲究

Fowler 点出了一个很细的差别：**对关键词来说，内容不重要，类型就是一切；对标识符来说，内容才是命根子。**

什么意思？`if` 这个词，编译器只在乎「它是个 `if`」——至于拼写本身，确认是 `if` 之后就没用了，类型「`if` 关键字」携带了全部信息。但 `price` 不一样：它是个标识符没错，可到底是 `price` 还是 `prize`，天差地别——后面算钱的时候要用到它指向的那个变量，内容必须原样带走。

所以 token 在这里是个**行李牌**：类型是牌子（「这是易碎品」「这是托运件」），决定后面的人怎么处置它；内容是箱子，有的箱子是空的（光靠牌子办事），有的箱子装着真金白银。

从「凭证」到「特征」：这一跳扔了什么

现在对照一下第三章的通行证。安全 token 代表的是「你已通过验证」这件**事实**；词法 token 代表的是「这段字符是什么」这个**判断**。替身还是那个替身——一个站出来代表别的东西的小对象——但正主换了：从「权利」换成了「类别」。

有意思的是，游戏厅的三条件仍然都在，只是变了形：

- **可铸造** → 词法规则定义了哪些字符组合是合法 token（`0.13` 是，`0.1.3` 不是）；
- **可流通** → token 从词法分析器交给语法分析器，一站站往下传；
- **可兑现** → 下游见到「数字」类型，就真的按数字处理，不再回头看原始字符。

还有那个老原因：语法分析器不用再逐字辨认「`p` 后面是 `r`」，只看 token 的类型牌——**验证替身，不验证正主**，在这里变成了「处理类别，不处理字符」。

本章留下的问题

词法分析的切法是**规则驱动**的：人先定好「字母开头的一串叫标识符」，机器照章执行，黑白分明。但轮到自然语言——中文没有空格，英文有词形变化，「unbelievable」算不算一个词？——规则开始左支右绌。

下一章和再下一章，我们将看到两次大胆的换招：搜索引擎用**语言学的刀**来切，大模型干脆用**统计学的刀**来切。同一个 token，刀法截然不同。

第五章 · 搜索引擎的切片师

你在搜索框里敲「北京 咖啡馆」，半秒后拿到一堆结果。这中间发生了什么？

搜索引擎没有读懂你的文艺情怀，它做的事机械得多：把你的查询切成 北京 和 咖啡馆 两个 token，然后翻一本提前编好的大账本——账本上记着每个 token 出现在哪些文档里——把同时沾着这两个 token 的文档捞出来，排个序，端给你。《相关性搜索》一书把这个过程概括成一句话：**从文本中提取特征（token），用它撮合用户预期和文档之间的匹配。**

注意定义里那个关键词：特征（feature，能代表一段文本「讲了什么」的那些抓手）。这一章的 token 不再代表权利，也不像编译器里那样代表「语法类别」——它代表的是**这段文字的主题**。替身还在，正主又换了：从「类别」换成了「意思」。

三把语言学的刀

切自然语言比切代码难得多，搜索引擎的刀法是一套组合拳：

- **分词**（tokenization）：先把句子切成词。英文按空格切就有个七八成，中文没有空格，得靠词典和算法猜：「北京咖啡馆」是「北京／咖啡馆」还是「北京咖啡／馆」？切错了，搜「咖啡馆」就捞不回这篇文章。
- **词干提取**（stemming）：把 running、runs 粗暴地削掉词尾，归成 run ——因为搜「跑步」的人不在乎你用哪个时态。快是快，但它只认得规则词尾。
- **词形还原**（lemmatization）：词干提取的文雅版，不光削词尾，还查词典认亲：不规则的 ran 能找回 run，better 能认祖归宗到 good。

大白话

大白话：搜索引擎根本不存你的文章，它存的是文章的「指纹」——一串切好、削平、认完亲的 token。查询也做同样的处理变成指纹，然后指纹对指纹。所以你搜「最好的咖啡」也能命中写「上好咖啡」的文章——不是搜索引擎懂中文，是两边的指纹被刀法归一到了同一批 token 上。

为什么敢只比指纹

只看 token、不管全文，听起来很粗暴。但这套刀法背后还是那句老话：**验证替身，不验证正主**。真正相关性的正主是「这篇文档合不合用户的心意」——没法直接算。token 是它便宜的替身：可以数、可以比、可以建账本，速度撑得起每秒百万次查询。

而且替身一旦可计算，就能玩出正主玩不了的花样：一个词在文档里出现得多、在全网出现得少，那它八成是这篇文档的「题眼」，权重给高些——这就是搜索圈著名的 TF-IDF（词频-逆文档频率，一套「越稀有越说明问题」的打分规则）。token 从「代表意思」进一步变成了「**可以称重量的意思**」。这个「可计量」的属性先记住，第八、九章它要唱大戏。

广义的 token：万物皆可切

《相关性搜索》还点了一句：token 不限于文本。图片可以切成视觉特征（颜色块、边缘、纹理的指纹），声音可以切成声学特征（一段段频率的指纹）。只要能把一个复杂对象切成一批「可比的替身」，检索的魔法就能成立。人脸识别搜图、哼唱识曲，骨子里都是同一台机器：**切 token，建账本，比指纹**。

本章留下的问题

搜索这把刀是**语言学**的：词典、词根、语法，背后站着人类几千年的语言知识。可这套刀法有两个软肋：一是每种语言都要单独打造一套（中文分词器和英文词干器完全是两套手艺）；二是「词」这个单位本身未必是最佳切点——凭什么「咖啡」要算一个 token 而不是两个？

下一章的主角把这把刀整个扔了。大模型的工程师说：别管什么词不词的，让语料自己说话——**统计**会告诉我们该在哪里下刀。切出来的东西连「词」都不是，却撑起了 ChatGPT 的整个世界。

第六章 · 大模型嘴里嚼的碎字

打开 OpenAI 官网的 tokenizer 页面，粘一句「我喜欢吃麻辣烫」，你会看到这句话被切成几段彩色小方块。再粘一句英文 `unbelievable`，它可能被切成 `un`、`believ`、`able` 三块——中间那块 `believ` 连一个完整的英文词都算不上。

这就是大模型嘴里的 token。官方定义朴素得惊人：**token 是文本中常见的字符序列**。不是词，不是字，是「经常一起出现的那么一段字符」。多常见算常见？不问语言学，问统计学——让语料自己投票。

让语料自己决定在哪下刀

切法叫 BPE (Byte Pair Encoding，字节对编码，一种数据压缩出身的老算法：反复把语料里出现最频繁的相邻片段合并成一个新单位，直到攒够几万种单位为止)。想象你面前摊着全网文本，然后玩一个贪吃蛇游戏：

- 第一步，每个字符都是独立的小段；
- 看一眼全语料，发现 `t` 和 `h` 老黏在一起，合并成 `th`；
- 再看，`th` 和 `e` 老黏在一起，合并成 `the`；
- 一直合并到攒出一张「常用零件表」——规模几万到几十万个片段不等，各家模型自己定。

从此任何文本都能用这张零件表拼出来：常见词整个是现成的零件（一个 token），生僻词拆成几个零件拼，没见过的怪词就退到单个字母去拼。**高频的整块批发，低频的拆件零售**——压缩效率就是这么来的。

大白话

大白话：语言学那套「什么是词」的知识，这里一概不用。BPE 只认一条：哪段字符老一起出现，哪段就焊死成一个零件。「麻辣烫」如果语料里出现得够多，就是一个 token；「麻辣小龙虾」如果不够多，就拆成「麻辣」+「小龙虾」或更碎的几块。词不词的，统计学不在乎。

半个词也是合法公民

这套刀法切出来的东西，在语言学家眼里相当「不正经」：英文里 `believ` 这种残肢是合法 token；中文里常见的是一个字或一个常见词，但「凤梨酥」也可能被拆成「凤梨」＋「酥」。维基百科特意强调了这一点：**大模型的概率切分和编程语言的词法切分，是两种不同的东西**——前者靠统计，切出来的可能不成词；后者靠规则，切出来的必是合法的语法单位。

更「不正经」的是，这张零件表**没有标准答案**。每家模型各自在自己的语料上焊零件，焊出来的表就不一样；同一家公司的新版模型换了语料、调了参数，切法也会变。同一句话，在 GPT-4 里是 12 个 token，换一家可能是 10 个也可能是 15 个。token 在这里第一次露出它的**相对性**：它不是文本的天然关节，而是「某个模型的某种刀法」的产物。

为什么大模型偏偏要这么切

因为模型干的那件事——OpenAI 文档里那句话：学习 token 之间的统计关系，擅长在一个 token 序列后面**预测下一个 token**——天生需要一批颗粒度刚好的「棋子」：

- 按整词切？词表爆炸，没见过的词直接抓瞎；
- 按单字母切？序列太长，模型刚读到 `t-h-e` 就忘了开头；
- BPE 是两头妥协的甜点：零件表不大不小，什么文本都拼得出，序列还不至于太长。

回头对一下全书的主线。这个阿凡达代表什么？token 在这里是**文本的最小可处理单位**——正主是「这段文字」，替身是「一串编号」。而它依然是便宜的替身：模型不必直接面对无穷无尽的原始字符，只面对一张有限的零件表。连「可计量」都被继承并放大了——每个 token 有编号，编号可以数，数出来就是下一章的账单。

本章留下的问题

到这里，token 已经从「代表权利的铜板」一路变成了「代表文本碎片的编号」。化身离灯芯越来越远，远到很多人已经感觉不到它们和「凭证」有什么关系。

但下一章画风一转，我们去一个看似八竿子打不着的地方——区块链——你会惊讶地发现，那里的 token 居然返祖了：它重新做回了凭证，只不过代表的东西从「玩一次游戏」变成了

「一笔资产」。而再往后一章，连大模型的 token 也被拽回了价值世界：当每个 token 都明码标价，智能本身变成了可以按斤称两的商品。

第七章 · 区块链上的硬币影子

兜兜转转六章，token 这个词终于在区块链世界完成了一次惊人的返祖——它又变回了游戏厅里的那块铜板，只不过柜台换成了代码。

打开任何一个加密钱包，你会看到两类东西。一类是区块链的「原生币」：比特币（BTC）、以太币（ETH）——它们像这片土地上的法币，协议本身就发行它们。另一类就叫 **token**：USDT、UNI、成千上万种名字各异的代币——它们是**别人在以太坊这块土地上开的「游戏厅」里发的铜板**。

这个区分正是本章的题眼：在链上，token 特指**由某个项目方发行、代表某种资产或权利的凭证**。连加密圈自己都沿用了「替身」的直觉——原生币是正主，token 是正主地盘上铸出来的影子。

柜台换成了智能合约

还记得游戏厅三条件吗？在以太坊上，它们各自找到了代码形态。最流行的一套铸币章程叫 ERC-20（以太坊上同质化代币的技术标准：规定一个代币合约必须提供「查余额、转账、授权代扣」等一组标准动作，让钱包和交易所不用为每个新币单独适配）：

- **可铸造** → 一段智能合约（smart contract，部署在区块链上、按写死的规则自动执行的程序）充当柜台。它决定总共铸多少、发给谁。规则写在链上，所有人可查，没人能偷偷加印——除非合约给自己留了后门（这就是买币前要读合约的原因）。
- **可流通** → 转账就是改合约里的账本：你名下的数字减一，对方加一。没有铜板物理移动，但「权利跟着 token 走」的效果一模一样，而且全场观众（全网节点）共同见证。
- **可兑现** → 这一步最关键也最参差：token 背后的「正主」到底是什么？

正主是谁，决定影子值多少钱

游戏厅铜板之所以敢收，是因为你信柜台一定兑现「玩一次」。链上 token 的分水岭也在这里——正主的成色千差万别：

- **稳定币**（如 USDT）：发行方声称每发 1 枚，就在银行存 1 美元。token 代表的正主是**法币存款**，所以价格锚在 1 美元。它本质上是「美元的游戏厅铜板」——进得了链上这个游戏厅，随时回柜台换真美元（只要柜台守信且有储备）。
- **功能型代币**：代表「使用某个服务的权利」——比如用某 token 支付去中心化存储的费用。正主是**一项服务**，像极了一张长期有效的游戏币。
- **治理代币**：代表「对项目决策的投票权」。正主是**一票发言权**——和令牌环网里「握令牌者得发言权」遥相呼应，绕了四十年，隐喻回家了。

大白话

大白话：链上 token 就是柜台发行的铜板，区别只在柜台是谁、铜板能兑什么。USDT 兑美元，治理币兑投票权，空气币什么都兑不了——正主不存在，影子再漂亮也只是影子。判断一个 token，永远先问：它代表的正主是什么？谁来兜底？

还有 NFT：每个铜板都独一无二

ERC-20 铸的铜板彼此没区别（一枚 USDT 和另一枚完全等价，这叫「同质化」）。但如果把章程改成「每枚铜板都有唯一编号、各自登记一件东西」，就得到 NFT（非同质化代币：链上登记「这个编号对应这张图、这块地、这张门票」的唯一凭证）。这时 token 代表的不再是可互换的权利份额，而是**某件独一无二之物的所有权登记**——替身和正主一对一绑定，像一张写了房本号的契据。

本章留下的问题

这一章的 token 回到了「凭证」，但有一个新属性被彻底点燃了：**价格**。游戏厅铜板面值固定（一元一个），链上 token 却在交易所里每秒波动——替身本身成了交易对象，影子卖出了比本体还热闹的行情。「token 可以计价、可以炒」这件事一旦成立，下一个问题就拦不住了：

如果有一种 token，它代表的是「智能」本身的消耗量呢？下一章回到大模型——你会发现 API 账单上那个按 token 计费的数字，正在把第六章的文本碎片变成第八章的经济单位。两个阿凡达，在收银台前握手。

第八章 · 按斤称两的智能

打开任何一家大模型厂商的价目表，你会看到一种很新的报价方式：不是包月，不是按次，而是「每百万 token 输入 X 元、输出 Y 元」。智能，像菜市场的猪肉一样，挂出了单价。

这是 token 最新的一个阿凡达，也是最出人意料的一个：一个语言学兼工程学的切词单位，摇身变成了**经济单位**——围绕它的调用、分发与结算，一套新的价值体系正在成形。有人把词元称为「智能时代可计量、可定价、可交易的最小信息单元」。这一章就拆解这句话：凭什么偏偏是 token，能坐上这个位子？

一个合格的计价单位需要什么

历史上能当计价单位的东西都过了三关：**数得清**（一度电就是一千瓦时，没有半个「度感」）、**人人认**（买卖双方的表得是同一种表）、**和成本挂钩**（多耗一度，电厂真多烧一把煤）。看 token 怎么过这三关：

- **数得清**：第六章说过，每个 token 有编号。一次请求用了多少输入、吐出多少输出，服务器数得一个不差——这是纯机械计数，没有扯皮空间。
- **人人认**：tokenizer 是公开的，你可以把同一段话拿去官网自己切一遍对账。表摆在明处。
- **和成本挂钩**：模型每处理一个 token，就要实打实地做一轮矩阵运算，GPU 的电费随 token 数近似线性增长。token 数和真金白银的成本之间，有物理意义上的对应。

大白话

大白话：token 就是 AI 时代的「电表读数」。电厂不按「你家的幸福感」收费，按千瓦时；大模型也不按「这个回答多有智慧」收费，按 token。两者都不完美，但都赢在同一点：表可信，数可查，成本对得上。

但它终究只是电表，不是电

这里有全书最值得警惕的一个混淆：**token 计量的是「处理的量」，不是「智能的质」**。一千个 token 的废话和一千个 token 的真知，账单上一模一样。电表只记你用了多少电，不

管你用来炼钢还是烤糊了面包。

而且这块表还有第六章埋下的那个毛病：**不通约**。各家刀法不同，同一段话在 A 家切出 100 个 token，在 B 家可能是 120 个。所以「每百万 token 的价格」跨厂比较时打了个折扣——就像两家商店都用「斤」，但一家的斤是 500 克，另一家是 450 克。精明的买家会把同一份真实任务分别跑一遍、看总账单，而不是只比单价。计价单位的历史重演了：秦始皇统一度量衡之前，跨国的「斤」也是一笔糊涂账。

围绕读数长出的生意

一旦有了可信的读数，围绕它的商业生态就像围绕电表长出的电力工业一样冒出来：

- **调用**：开发者按 token 买「智能原料」，组装成自己的产品；
- **分发**：聚合平台像电力批发商，打包多家模型转售，赚调度和差价；
- **结算**：企业内部按部门分摊 token 用量，像分摊电费；预算、限流、成本优化全围绕着这个读数转——「这句提示词能不能省 200 个 token」成了真实的工程问题，提示词工程一半是技术，一半是节俭。

再往远处看一步：当 AI 代理（agent）开始互相调用、互相付费时，它们之间结算的「货币」大概率就是 token 或它的某种衍生凭证。第六章的文本碎片和第七章的链上铜板，在收银台前正式握手——一个提供读数，一个提供账本。

本章留下的问题

八个阿凡达巡礼完毕（算上词典本义和硬件返祖）：凭证、通行证、语法符号、文本特征、统计碎片、资产凭证、计价单位。有的代表权利，有的代表类别，有的代表意思，有的代表钱。

是时候回答第一章那个贯穿全书的问题了：**这些化身到底共享了什么，让六个行业不约而同抓起了同一个词？**下一章收网——我们会看到，答案不在任何一本技术手册里，而在侯世达那盏灯里：概念的扩张从来不是逻辑推演，而是一次次类比起跳。

第九章 · 阿凡达们共一张脸

把八个化身在桌上并排摆开：

- **词典本义**：凭证、标志、纪念物；
- **游戏厅/令牌环**：代表一次操作权的信物；
- **安全领域**：代表「你已通过验证」的通行证；
- **编译器**：代表一段字符「是什么」的类型牌；
- **搜索引擎**：代表一段文本「讲什么」的特征指纹；
- **大模型**：代表文本碎片的最小处理单位；
- **区块链**：代表一笔资产或一份权利的链上契据；
- **经济层**：代表智能消耗量的计价单位。

初看像一锅大杂烩：一会儿是权利，一会儿是类别，一会儿是钱。但如果你跟着全书一路走来，会看见一条从未断过的线——

共享的那张脸：替身

每个阿凡达都是同一件事：**一个东西站出来，代表另一个东西；正主太贵、太大、太抽象，替身便宜、小巧、可机读。**而且每个世界都不约而同地利用了同一条工程学真理，我们在第二章就埋下的那句：

大白话

大白话：验证替身，永远比验证正主便宜。闸机不查你这个人，查工牌；语法分析器不读字符，读类型牌；搜索引擎不读文章，比指纹；大模型不面对无穷文本，面对有限零件表；交易所不搬金条，改账本；财务不估算「智能值多少钱」，数 token。六个行业，同一个偷懒的智慧——而这个「偷懒」恰恰是整个数字世界跑得动的原因。

连第二章的三条件都贯穿始终：每个世界都要回答「谁来铸」（柜台、登录服务器、词法规则、分词算法、BPE 语料、智能合约、tokenizer）、「怎么流通」（投递、请求头、流水线、

索引、序列、转账、账单)、「兑什么现」(游戏、权限、语法结构、相关性、下一个词、资产、智能服务)。三问题是普适的，答案各带本地色彩——这就是阿凡达换躯体不换灵魂的确切含义。

每一跳，丢了什么、留了什么

概念迁移不是整体搬家，是有取有舍的。回看这条链：

- 从铜板到通行证：丢了「实物」，留了「代表权利」；
- 从通行证到词法 token：丢了「权利」，留了「代表某个判断、免检流通」；
- 从词法到搜索：丢了「语法规则」，留了「代表一段文本、可比对」；
- 从搜索到大模型：丢了「语言学知识」，留了「特征」，捡起「可计数」；
- 从大模型到经济层：什么都没丢，直接兑现「可计数」——编号即读数，读数即账单；
- 区块链那条支线则反着走：从纯信息世界跳回价值世界，把丢了一路的「权利」和「凭证」全部捡回来。

没有任何一个设计师规划过这条路线。是六个行业各自的工程师，在面对「正主没法直接处理」的难题时，各自从公共词库里抓住了同一个现成的隐喻：**铸个替身出来**。词不是被定义的，是被一次次抓来救火的——救火救多了，火场就成了它的新家。

类比是思维之火

现在可以回答第一章那个更大的问题了：理解一个新概念，到底是怎么发生的？

侯世达的答案是：从来不是从零开始，而是**把新东西挂到旧概念的钩子上**。你第一次听说「大模型的 token」时，如果有人告诉你「就像编译器切出来的词，不过是按统计切的」，你三秒就懂了；如果给你一篇形式化定义，你要啃半小时。前一种教法之所以快，是因为你的理解本来就是一张类比的网——新概念只要在旧网上找到一根足够像的线头，整张网就为它所用。

这也正是「token」这个词自己的命运。它没有一份中央定义，靠的是一代代工程师做同一件事：**发现新问题与旧场景「骨子里是同一回事」，于是把旧词连同旧直觉一起搬过来**。概念范畴不是有围墙的领地，而是有引力半径的光晕——这是《表象与本质》最想讲的那件事，也是 token 用六张脸亲自演示的那件事。

尾声：下次见到 token

下次再遇到这个词——在 API 文档里、在钱包里、在一篇论文里——别急着问「token 到底是什么」。问三个老问题就够了：**它替谁站岗？谁铸的、怎么流通、兑什么现？**答案会自动告诉你，你此刻面对的是哪一尊阿凡达。

而如果哪天你在一个全新的领域又撞见它——别惊讶。那不是巧合，是灯芯又亮了一次。

Token 的阿凡达们 · 竹小竹8167