THE LICENCE AND THE SHARED NAME

From GPL and Linux to Constitutional Compacts for Decentralised Brands

THE LICENCE AND THE SHARED NAME

From GPL and Linux to Constitutional Compacts for Decentralised Brands

SÎNICĂ ALBOAIE

A research manuscript in licensing and institutional design

OPEN CODE. RECIPROCAL REPUTATION. LIMITED CUSTODY.

OPEN SOURCE • AI LICENSING • DECENTRALISED BRANDS

Artificial intelligence systems were used as editorial and research instruments.
The argument, selection, and authorial responsibility belong to Sînică Alboaie.

Author's Note:

This volume was created as part of the overarching Outfinitism research program. While Artificial Intelligence language models were used to aid in text generation, the foundational philosophy, thematic structure, and analytical approach are exclusively derived from the author's decades of dedicated study of social phenomena and social technologies.

Outfinity Venture Studio's strategy is built on the belief that to launch global AI-native ventures, they will most likely need to be built upon the concept of "Constitutional Compacts," and more broadly, the evolution of open-source licensing examined in this book

.

NOTE ON SOURCES AND THE STATUS OF THIS BOOK

This book begins from two bodies of ideas that must remain distinct. The first is historical and descriptive. It concerns the application of copyright to software, the emergence of free software and open source, the role of the GNU General Public License in the formation of the Linux ecosystem, the rise of permissive licensing, the conversion of openness into an industrial strategy, the conflicts created by cloud computing, and the present fragmentation of rights across artificial-intelligence data, code, model weights, derivatives, outputs and services. That account draws on licence texts, official documentation, institutional histories, academic work and public statements by the organisations involved.

The second body is normative and prospective. It develops the idea of the decentralised brand set out in *Decentralised Brands: Shared Identity, Distributed Power, and the Architecture of Resilient Cooperation* [ALBOAIE-DB]. In that framework, a brand is not merely a logo, campaign or privately owned commercial sign. It is an institution that connects a name to a public promise, rules, reputation, infrastructure, contributions and a particular distribution of authority. The present book asks which legal and economic instruments might allow such an institution to exist without becoming either the sovereign property of one centre or an undefended commons that others can exploit while returning little to those who sustain it.

The proposal called the Decentralised Brand Compact is not an existing licence and is not presented as a legally validated solution. It is a research architecture for institutional design. It separates the software licence from the shared-mark regime, contribution governance from the universal right to use open-source software, provenance from reputation, and the right to fork from the right to claim continuity under the same name. Any implementation would require jurisdiction-specific drafting, specialist legal advice and empirical testing in real ecosystems.

The book is not an indictment of companies. Large corporations have funded, maintained and extended an enormous part of the open-source infrastructure on which the contemporary economy depends. Nor are permissive licences described as moral errors. They have produced standards, markets and innovations that might not have emerged under more restrictive regimes. The criticism is narrower. The freedom to copy code does not automatically distribute market power, customer relationships, reputation, access to infrastructure or revenue. Open source democratised technical production more successfully than it democratised the capture of value created through that production.

Several legal corrections are important in this revised edition. A licence is not always a contract, and the words cannot safely be used as synonyms. An

open-source copyright licence may operate as a conditional permission under copyright law, may also have contractual effects in some circumstances, or may be analysed differently across jurisdictions. A contributor covenant, membership agreement, service agreement, brand constitution and trademark licence are separate instruments. The phrase "a licence is a small constitution" is therefore used as an institutional metaphor, not as a universal proposition about legal classification.

The same caution applies to words such as "franchise", "collective mark", "certification mark", "dividend", "derivative work" and even "model licence". Their legal meaning depends on the jurisdiction, the rights actually held, and the substance of the arrangement rather than the label chosen by its designers. Model weights may engage copyright, contract, database rights, patents, trade secrets or none of these in the same way in every country. A rights manifest records the basis claimed and the uncertainties that remain; it does not manufacture titles where no title exists.

This is not legal advice. Legal examples are used to map a design space, not to replace analysis by qualified counsel. The compact proposed here should be treated as a hypothesis to be tested against trademark law, franchise regulation, competition law, tax, employment, consumer protection, data protection and the actual incentives of contributors, customers and commercial nodes.

TABLE OF CONTENTS

NOTE ON SOURCES AND THE STATUS OF THIS BOOK 2
PROLOGUE. THE PROMISE WITHOUT AN OWNER 6
INTRODUCTION. A LICENCE IS A SMALL CONSTITUTION 9
PART I: HOW CODE ACQUIRED AN OWNER 12
CHAPTER 1. WHEN SOFTWARE LEFT THE MACHINE 13
1.1 The technical gift before the market 13
1.2 The legal arrangement that separated creator and user 16
CHAPTER 2. COPYRIGHT TURNED AGAINST CLOSURE 19
2.1 GNU and the invention of copyleft 19
2.2 BSD and the two meanings of freedom 22
CHAPTER 3. LINUX AND THE PACT AMONG STRANGERS 25
3.1 The move to GPL 25
3.2 What GPL did - and did not - do 28
PART II: WHEN OPENNESS BECAME A MARKET STRATEGY 31
CHAPTER 4. THE LICENCE AS A MARKETING ENGINE 32
4.1 Netscape and the rebranding of freedom 32
4.2 Adoption before the sale 35
CHAPTER 5. CAPTURE WITHOUT THEFT 38
5.1 Cloud computing and complementary assets 38
5.2 Relicensing and the expensive fork 41
CHAPTER 6. THE ECONOMY MISSING FROM THE LICENCE 44
6.1 Maintenance and invisible labour 44
6.2 Corporations: real contribution, asymmetric power 47
PART III: FROM CODE TO THE SHARED NAME 50
CHAPTER 7. A BRAND IS AN INSTITUTION 51
7.1 The name as a promise 51
7.2 The decentralised brand 54
CHAPTER 8. FRANCHISING AND ITS INSTITUTIONAL RELATIVES 57
8.1 Franchising: entrepreneurial edges, sovereign centre 57
8.2 Cooperatives, collective marks, certification and standards 60
CHAPTER 9. CONSTITUTIONAL OPENNESS 63
9.1 Layers of intellectual property and control 63
9.2 Membranes, autonomy and exit 66
PART IV: ARTIFICIAL INTELLIGENCE AND THE END OF THE SINGLE LICENCE 69
CHAPTER 10. A MODEL IS A GENEALOGY 70
10.1 The AI licensing stack 70
10.2 Open weights are not enough 73
CHAPTER 11. DATA AND THE OBLIGATION THAT DISAPPEARS 76
11.1 Licensing data for training and use 76
11.2 Machine-readable provenance 79
CHAPTER 12. THE AGENT THAT CONSUMES THE COMMONS 82
12.1 AI as a multiplier of capture 82
12.2 AI as an administrator of reciprocity 85
PART V: THE COMPACT OF THE SHARED NAME 88
CHAPTER 13. OPEN CODE, RECIPROCAL NAME 89
13.1 Moving reciprocity towards the brand 89
13.2 The seven instruments of the compact 94
CHAPTER 14. THE ECONOMY OF CONTRIBUTION 99
14.1 Contributor distributions without a mythology of merit 99

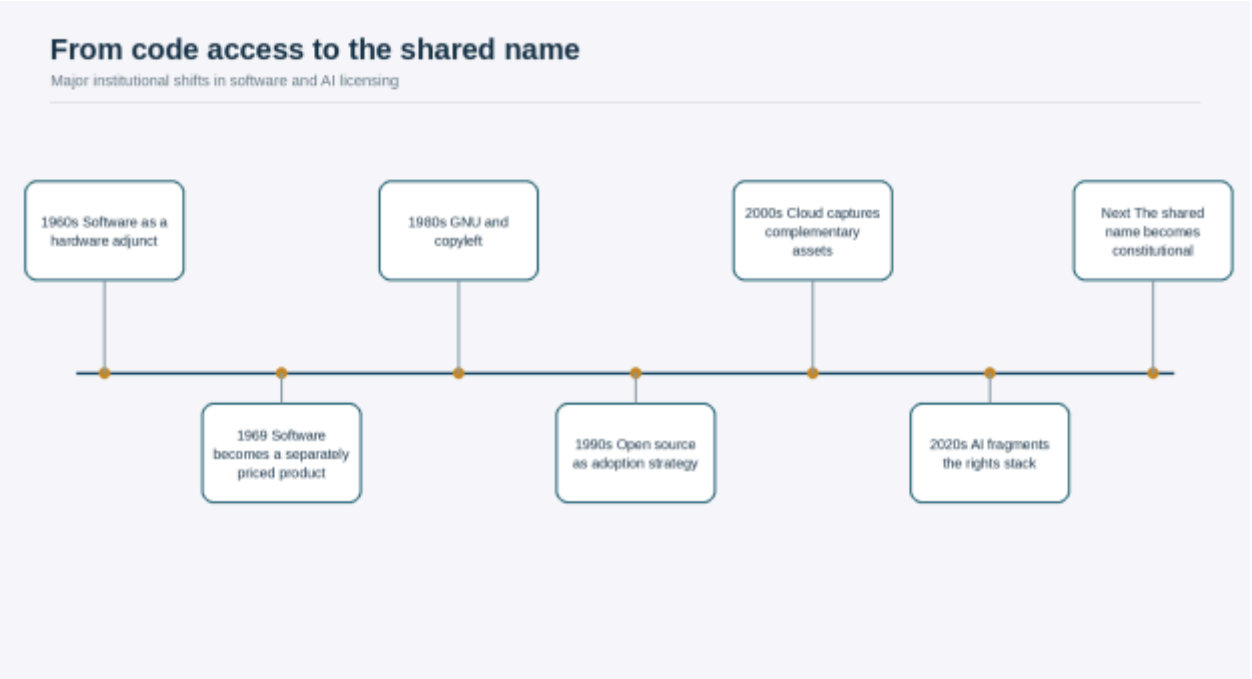
14.2 The budget that tells the truth..... 103
CHAPTER 15. CUSTODY WITHOUT SOVEREIGNTY..... 107
15.1 Governance, succession and emergency power..... 107
15.2 The fork that preserves dignity..... 111
CHAPTER 16. THREE FUTURES OF LICENSING..... 114
16.1 Empire, labyrinth and archipelago..... 114
16.2 A research and implementation agenda, 2026-2035..... 118
EPILOGUE. AFTER ABSOLUTE OWNERSHIP..... 122
ANNEX A. MINIMUM TERMS FOR A PILOT COMPACT..... 125
ANNEX B. THE RIGHTS AND PROVENANCE MANIFEST..... 128
BIBLIOGRAPHY AND SOURCES..... 131

PROLOGUE. THE PROMISE WITHOUT AN OWNER

A program can be copied at almost no marginal cost, but trust cannot be copied in the same way. Code multiplies by reproduction. Trust accumulates through repeated experience, responsibility, reputation and the ability to identify someone when a promise is broken. This difference has shaped the whole history of software. The free-software movement liberated code. Open source demonstrated that the freedom to inspect, modify and redistribute could produce more robust systems and larger markets. Companies then learned to build a new centralisation above those freedoms: they left the code open while concentrating cloud infrastructure, distribution, operational data, the customer relationship, the brand and the capacity to provide warranties and indemnities.

This transformation was not usually theft. Permissive licences authorised much of it. GPL constrained some forms of closure, but it never promised to distribute revenue, customers or prestige. A cloud provider could lawfully use free code, pay thousands of engineers and create real value for customers while the upstream project remained dependent on a handful of exhausted maintainers. A corporation could contribute heavily to the Linux kernel and, at the same time, acquire practical influence over technical priorities that an unpaid individual could not match. Both statements can be true. Open source has been a productive commons and an infrastructure for industrial concentration.

Figure 1. The major institutional shifts in software and AI licensing



The figure presents a sequence of institutional shifts rather than a moral progression. Software moved from an adjunct of hardware to a separately licensed product. Copyleft then used copyright against closure. Open source

converted legal freedom into a strategy of adoption. Cloud computing moved value from distributing the program to operating it. Artificial intelligence fragmented the object of licensing into code, data, weights, model derivatives, outputs and services. The next possible shift concerns the shared name: who may build reputation with others, who may earn from it, and who is entitled to define the promise that the name carries.

The centralised brand solves the problem simply. A company owns the mark, sets the standards, authorises use of the name and captures a disproportionate part of the reputational value. Even when a product is made by thousands of employees, suppliers, creators, integrators and community members, legal authority over the public promise remains concentrated. The advantage is coherence. The cost is that those who build the meaning of the brand may remain executors inside property they cannot contest and from which they cannot depart with the reputation they helped create.

Open source solves a different part of the problem. It distributes the right to continue the code without seeking permission from the centre. Yet a technical fork can be economically weak if it loses the name, domain, package registry, conferences, certifications, operational data, distribution channels and customer relationships. The formal right of exit exists, but the social cost of exit may be so high that the centre remains sovereign in practice.

The thesis of this book is that an underexplored institutional space lies between the proprietary brand and the economically defenceless commons. A decentralised brand could allow several autonomous organisations to use, protect and enlarge a common promise without turning each organisation into a subsidiary and without making the name available for every imposture. To do so, it requires more than a conventional software licence. The code may remain genuinely open source. Economic reciprocity should attach to the right to present oneself to customers under the official shared name, not to the universal right to copy the code.

That shift appears modest, but it changes the object of negotiation. Instead of trying to make GPL into a revenue-distribution system, an ecosystem can preserve open-source freedoms and build a constitution of reputation above them. Anyone may run, study, modify and redistribute the program under its licence. Only nodes that meet published standards, contribute to the common goods, accept audit and submit to transparent rules of distribution may claim official membership in the brand. The mark is no longer merely the private reward of the company that first reached the customer. It becomes a shared asset held under limited custodianship for those who maintain the promise.

This is where comparison with franchising becomes useful. A franchise distributes investment and entrepreneurial activity while preserving central authority over the name and operating method. A decentralised brand would retain the discipline of a public promise but constrain the centre through limited custody, plural representation, transparency, succession rules and meaningful

exit. It would resemble a cooperative without requiring every participant to adopt one ownership form. It would resemble a certification mark while adding contribution and a common economy. It would resemble an open-source project while recognising that reputation and market access are assets distinct from code.

The objective is not purity. Any institution strong enough to protect a name will accumulate power. A system open enough to receive contributions will attract opportunists. A reward formula will invite strategic behaviour. A contribution registry will omit invisible work. A custodian can become a de facto proprietor. No licence can abolish politics. A well-designed compact can, however, make power more visible, obligations more predictable, correction more practical and capture more difficult.

The following chapters trace this possibility from the early history of software to AI models and future ecosystems of agents. The story is not about replacing open source. It is about adding a layer that open source was never designed to provide: a legitimate economy of the shared name.

INTRODUCTION. A LICENCE IS A SMALL CONSTITUTION

A licence is often treated as an auxiliary legal text placed in a file that developers rarely open. In practice, licensing is one of the most compact forms of institutional design. It determines who may enter, what may be taken, what must be preserved, what may remain secret, what may be sold, and under which conditions one person may continue the work of another. A few pages can establish an order of cooperation among strangers who live in different jurisdictions and have never negotiated with one another.

Table 1. The legal instruments of an open technological institution

Legal layer	Primary institutional function
Copyright licence	Grants permission to exercise rights otherwise reserved by copyright, subject to stated conditions.
Contract or covenant	Creates promises between identified parties and can govern services, contributions, membership, payment, audit and remedies.
Trademark or certification rules	Control who may use a name or sign, what the sign communicates, and how quality or membership is verified.
Organisational constitution	Allocates authority, succession, emergency powers, amendment procedures and rights of voice and exit.
Rights and provenance manifest	Records artefacts, versions, declared rights, inherited obligations, uncertainty and the evidence supporting each claim.

The table replaces the tempting fiction that one licence can govern an entire ecosystem. Copyright permission, contractual promises, trademark control, organisational governance and provenance are related but distinct. Treating them as separate instruments reduces ambiguity and prevents a dispute in one layer from silently changing the rights in another.

The constitutional metaphor remains useful if its limits are explicit. Licensing can constrain a rightsholder by granting permissions in advance, define durable rights for participants, and create procedures that survive changes in personnel. GPL does not require a recipient to trust the author's future goodwill. The permission to modify and redistribute, together with the conditions that preserve corresponding freedoms for recipients, is stated in the legal instrument. Yet the instrument is not a political constitution, and its enforceability need not depend on a single theory of contract. The more precise expression is that a licence forms part of a permission architecture.

Permissive licences such as MIT and BSD distribute rights to use and integrate code with minimal friction. They do not require the larger product to remain open and do not create an economy for contributors. Apache License 2.0

adds a carefully drafted patent grant and notice structure that can reduce corporate uncertainty. GPL protects reciprocity in covered distributed derivatives, but does not require revenue sharing and does not govern the trademark. AGPL extends a source-offer obligation to a defined form of network interaction, yet it still does not make customers, reputation or the commercial channel into a commons. Source-available terms may restrict competition or fields of use, but those restrictions place them outside the Open Source Definition [OSI-OSD].

The distinction matters because software is no longer merely code distributed as a copy. An application may depend on services, identities, data, model weights, package registries, discovery channels, certification, reputation systems and operational infrastructure. In artificial intelligence, a supposedly single model may combine Apache-licensed code, weights under bespoke community terms, training data of incomplete provenance, a tokenizer under another licence, a revocable acceptable-use policy, a proprietary API and outputs whose copyright status depends on human authorship and jurisdiction. The file called LICENSE may describe one layer while the institution remains mostly invisible.

The future of licensing will therefore be modular. Instead of one universal formula, serious systems will maintain graphs of rights and obligations. Some permissions attach to code, others to data, weights, adapters, marks, hosted services or membership. Provenance must travel with technical transformation. A model produced by fine-tuning, distillation, merging or quantisation should carry not only the licence chosen for the new contribution but also the material upstream obligations and unresolved claims. SPDX 3 and CycloneDX's AI and machine-learning bill-of-materials capabilities point in this direction [SPDX-3; CYCLONEDX-MLBOM].

A decentralised brand adds one more difficulty and one more opportunity. It cannot be built by opening code alone because the brand is the promise recognised by the customer. Nor can it be built by releasing a trademark without conditions, because a name that no institution legitimately controls may cease to signal consistent origin, membership or quality. In trademark systems that recognise licensed or collective use, legitimate control over the nature and quality of goods or services is not an optional decoration [USPTO-RELATED; USPTO-COLLECTIVE]. Decentralisation therefore cannot mean abandonment. The custodian must be strong enough to apply objective standards and weak enough to be replaced when it captures the institution.

The solution proposed here is constitutional openness. It does not mean opening every resource to the maximum possible degree. It means that each form of control must be justified by function, limited in scope, reviewable, and compatible with meaningful exit. Code may be open, the shared mark conditional, sensitive data protected, standards public, factual reputation portable, and common revenue distributed under known rules. No layer should be allowed to swallow all the others.

This architecture rejects two simplifications. The first says that every restriction betrays open source. A software licence that discriminates by field of endeavour is indeed not open source in the OSI sense. It does not follow that certification, support, liability, membership, marketplace access and use of the official name must be free of contractual conditions. The second simplification says that sustainable funding requires closing the code or prohibiting competition. Linux, open foundations and professional-service ecosystems show that value can be captured in complementary layers. The unresolved question is who controls those layers and how part of the value returns to the people who make them possible.

A third simplification must also be rejected. A shared brand is not automatically fair because ownership has been placed in a foundation, cooperative or distributed ledger. The common institution may become a cartel, an exclusionary club, a founder-controlled shell or an automated bureaucracy. Competitors collaborating through a shared mark must continue to set their own prices and commercial terms. Standards and certification should be transparent, proportionate, appealable and, where feasible, available to qualifying non-members. A common name should coordinate trust, not coordinate prices, allocate customers or suppress independent competition [FTC-ASSOCIATIONS; EU-HORIZONTAL].

The history begins before software was widely understood as a product. By following its conversion into licensed property, we can see why copyleft was such a powerful institutional invention and why it cannot by itself solve the problem of the brand. By examining corporate capture, we can avoid the caricature in which firms merely exploit communities. By examining AI, we can understand why the single licence is dissolving. Only then can we design a credible middle path: open code, a protected shared name, visible contribution and custody without sovereignty.

PART I: HOW CODE ACQUIRED AN OWNER

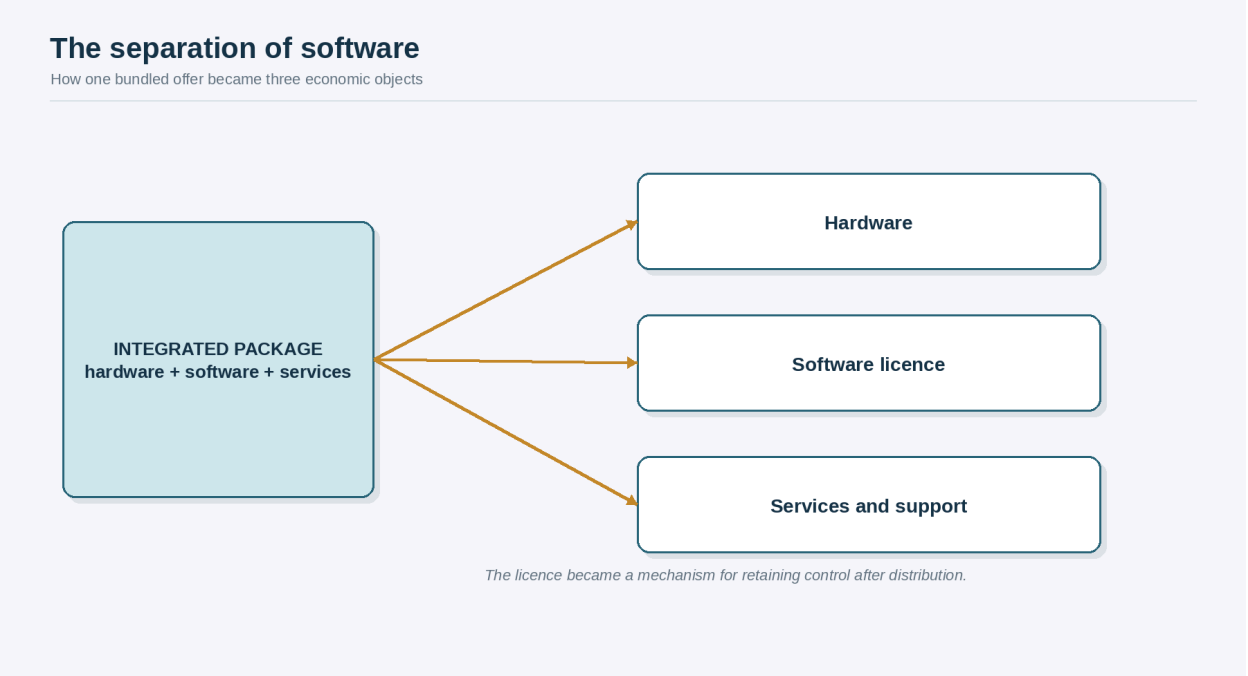
OPEN TECHNICAL COMMONS • ACCOUNTABLE INSTITUTIONS

CHAPTER 1. WHEN SOFTWARE LEFT THE MACHINE

1.1 The technical gift before the market

The history of software licensing begins in a world in which software was not yet an autonomous economic object. Early computers were scarce, expensive systems closely tied to the institutions that owned them. Programs travelled with machines, documentation, technical staff and relationships among laboratories. Most commercial value lay in hardware, while code was often treated as knowledge required to make the machine useful.

Figure 2. The separation of software from hardware and services



On the left, the figure shows an integrated system in which a manufacturer sold the machine while software and technical services supported its value. On the right, the elements have become separate markets: hardware is purchased, software is licensed, and services are charged as distinct lines of revenue. The separation occurred gradually, but IBM's 1969 unbundling announcement remains an important symbolic marker [IBM-1969].

Exchange in the early laboratory culture had a practical basis. Hardware was unstable, documentation incomplete, and users were often researchers capable of altering the system. Denying access to source code could make maintenance impossible. Technical communities formed around shared problems, and reputation came from the ability to repair and contribute rather than from exclusive control over a copy.

This world should not be romanticised. Access was limited to universities, corporations and public agencies wealthy enough to possess the machines. Code

could circulate freely within a technical elite while most people had no relationship with computing at all. Freedom was often the by-product of a market that had not yet formed, rather than an explicit moral constitution. As long as value was captured through the sale of hardware, a manufacturer could tolerate extensive exchange of programs.

Standardised hardware and a broader market gave code an economic identity of its own. Producing another copy cost almost nothing, while creating the first reliable version could require years of specialised work. That combination produced both the temptation and the perceived necessity of legal control. If a product could be copied immediately, how could its developer recover the investment? The emerging industry answered by using copyright permissions and contracts to control access, copying and use.

Software exposed an enduring mismatch. It is simultaneously expression, instrument, infrastructure and executable knowledge. Copyright had developed around works that could be read, heard or viewed, not around artefacts that direct machines and govern processes. Software licensing therefore came to do more than regulate reproduction. It addressed installation, numbers of users, transfer, modification, reverse engineering, warranties, maintenance and liability. The legal package around software often included a copyright licence, contractual terms and trade-secret protection; describing every element as "the licence" concealed this plurality from the beginning.

The proprietary model solved the investment problem through a powerful distinction. A customer could acquire a physical medium, a subscription or access to a service without acquiring a general right to reproduce and transform the program. In mature proprietary culture, the program was not "yours" in the way a physical tool might be. You received a defined permission to use it. The supplier retained authority over source code, evolution, distribution and, increasingly, interoperability.

The transformation produced real benefits. It supported a specialised software industry, large investments and products for mass markets. Companies could fund teams, testing, documentation, distribution and support. Customers gained a recognisable counterparty that could accept commercial responsibility. Yet the cultural relationship between developer and user changed. Executable knowledge became a black box, and the local capacity to repair was replaced by dependence on the supplier's roadmap.

The first free-software movements did not therefore arise simply against the abstract idea of property. They reacted to the disappearance of a concrete practice. Programmers accustomed to repairing, sharing and learning discovered that the same activity could become a breach of terms or an infringement. The conflict was not only about price. It concerned who had authority to understand and continue a system on which others depended.

This change contains the prehistory of the decentralised brand. While software circulated within a small community, reputation attached to people and

laboratories. Once programs became mass products, trust became concentrated in the supplier's name. The customer could not directly evaluate the whole codebase and instead purchased the organisation's promise. Intellectual property and brand became two faces of one system of control: one limited unauthorised reproduction; the other concentrated the signal of quality and responsibility.

Open source would reopen the code without dissolving reputational concentration. Some of the most powerful companies of the digital era would learn to distribute open code precisely to enlarge a platform, a service or a brand. Understanding that apparent paradox requires first understanding the proprietary arrangement that separated use from ownership and the creator from the customer.

1.2 The legal arrangement that separated creator and user

A proprietary software licence is not a single invention but a family of legal techniques through which a rightsholder retains most rights and grants only the permissions required for a defined use. Terms may limit installation to one device, prohibit modification, condition transfer, restrict benchmarking or reverse engineering, and exclude warranties. The phrase "licensed, not sold" expressed an ambition to preserve control after a copy had left the supplier, although the legal effect of that phrase and the limits of contractual restrictions vary by jurisdiction and method of distribution.

Table 2. The proprietary bundle

Controlled element	Practical consequence
Source code	The customer can execute the program but cannot normally inspect or continue the implementation independently.
Distribution rights	Access depends on the supplier's channels, pricing and continued consent.
Updates and interoperability	The supplier controls release cadence, compatibility, migration and often the practical cost of exit.
Brand, support and liability	Public trust and commercial responsibility are concentrated in one organisational centre.

The table shows how the proprietary model bundled four advantages into one relationship. Control of the copy, technical evolution, the customer channel and the public name reinforced one another. The customer received convenience and an accountable counterparty but lost much of the ability to continue independently. The arrangement is efficient while the supplier remains competent and reasonable; it becomes fragile when the supplier disappears, abandons a product, raises prices or exploits lock-in beyond the value it provides.

Economically, licensing turned an artefact with a near-zero marginal cost of reproduction into a series of controllable accesses. The same program could be segmented by edition, function, number of users, territory, duration and level of support. This flexibility funded development, but it also normalised a culture of restriction. The company did not merely sell functionality. It sold permission to use functionality under conditions that it largely defined.

The arrangement also converted what might once have been a completed sale into a continuing relationship. Updates, activation, telemetry, security patches, hosted components and compatibility decisions could keep the customer within the supplier's institutional perimeter for years. Formal licence terms and technical architecture reinforced one another. A customer might possess a perpetual permission on paper yet remain unable to operate safely after support ended, a file format changed or an authentication service disappeared. Practical dependence could therefore exceed the exclusivity stated in copyright law.

That dependence was not always irrational. Enterprise buyers often preferred one organisation to coordinate releases, warrant defined performance, defend intellectual-property claims and accept contractual consequences when a system failed. The proprietary bundle solved a genuine assurance problem. Its weakness was that the customer could rarely purchase assurance without also accepting the supplier's control over the code, migration path and public identity of the product. Rights in the artefact, the service relationship and the signal of trust were treated as inseparable.

This historical bundle suggests the institutional task pursued later in this book. The code licence, the commercial service contract and the rules governing use of the mark should be capable of separation. A customer may buy support and liability without surrendering every route of technical continuation. A creator may contribute to common code without transferring the complete future value of a name. Unbundling these elements does not abolish the firm; it makes the source of each authority visible.

Brand became the mechanism through which the customer accepted opacity. Few buyers could verify the quality of a complex program directly. The supplier's name condensed the history of previous products, the promise of support and the reputational risk borne by the organisation. The mark was a cognitive shortcut: a customer did not need to understand every line of code if it trusted the institution whose name stood behind the product.

This explains why publishing source code does not by itself eliminate central power. An enterprise customer may still buy assurance, certification, integration, continuity and liability. A company can distribute code at no charge while monetising the fact that its name reduces perceived risk. In open-source markets this capacity becomes more important, not less, because technically similar implementations compete to embody the most credible promise.

The proprietary system created another separation. Individual creators often assigned or licensed rights to an employer, while the corporate brand absorbed the reputation produced by collective work. Engineers created the technology; the legal relationship with the customer and the resulting reputational capital belonged to the company. That outcome is not inherently unjust. The company finances, coordinates and assumes risks that an individual may be unable to bear. The problem emerges when contribution becomes invisible and salary is treated as the complete account of value even though the work creates a durable institutional asset that continues to earn after the creator has left.

The free-software movement would attack the technical side of this separation. It would insist that users should be able to run, study, modify and redistribute the program. Copyleft would use copyright itself to make those freedoms durable. It would not create a general right to revenue or co-ownership of the brand. Such a rule would have been difficult to administer and might have contradicted the movement's universal commitment to freedom of use.

The resulting history preserved an asymmetry. Code could become common while the mark remained proprietary. Work could be collaborative while access to customers remained centralised. Technical governance could be described as meritocratic while economic governance remained with shareholders. A project could have thousands of contributors and one owner of the key domain, registry or mark.

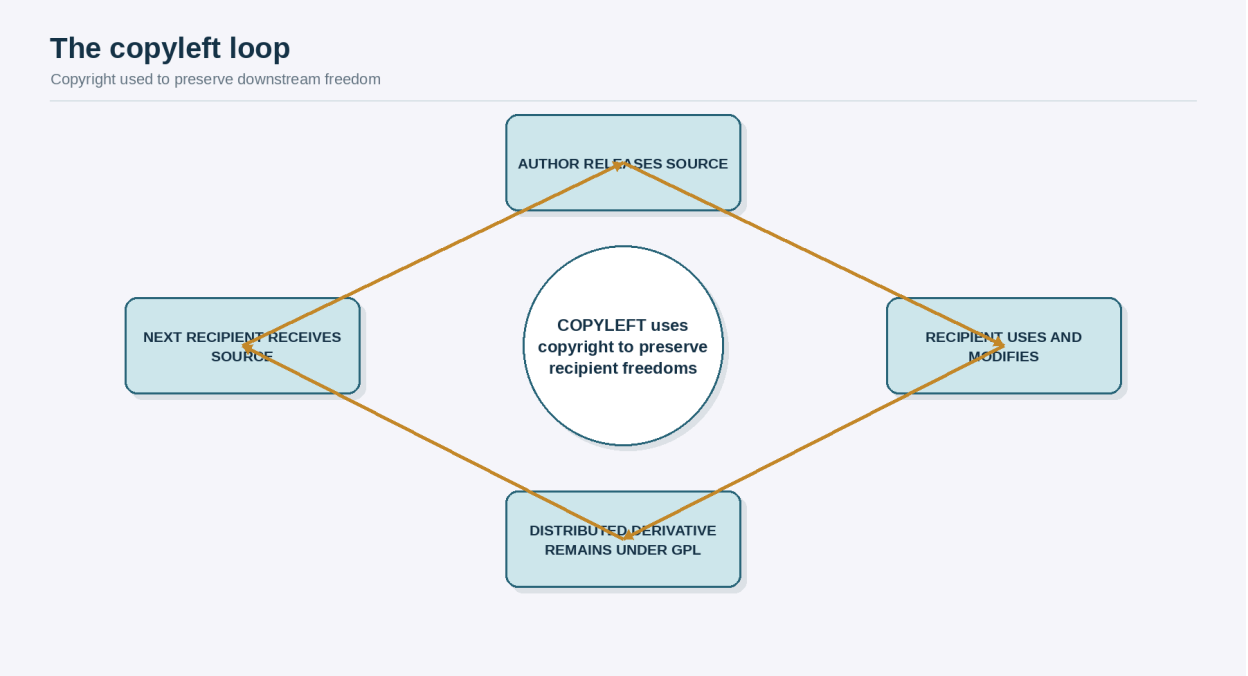
This is not a defect accidentally left in open-source licences. Those licences were designed to solve a more precise problem: freedom in relation to the program. Criticising them for failing to distribute every form of value confuses the object of the instrument. A better response is to preserve the technical freedom they created and design additional institutions for reputation, contribution and shared economic power.

CHAPTER 2. COPYRIGHT TURNED AGAINST CLOSURE

2.1 GNU and the invention of copyleft

The GNU Project began from the experience of a loss. Richard Stallman described a laboratory culture in which programmers shared code and a subsequent world in which proprietary programs made such cooperation unlawful or impossible. His answer was not merely to publish one program. It was to construct a complete operating system and a legal order designed to prevent that system from being closed again [FSF-GNU].

Figure 3. The institutional loop of copyleft



The figure shows the recursive logic of copyleft. The author uses copyright to grant broad freedoms. A recipient may copy, modify and redistribute. When a covered derivative is distributed, corresponding source and equivalent freedoms must reach the next recipient. An exclusive right that could have been used to close the program becomes the mechanism through which openness is reproduced.

Version 1 of the GNU General Public License appeared in 1989 and GPLv2 in 1991 [GNU-GPL-HISTORY]. Its innovation was institutional more than rhetorical. A moral request to contribute back depends on the character and incentives of the recipient. Copyleft establishes legally stated conditions triggered by defined acts of distribution. It permits cooperation among strangers by reducing the fear that one participant will unilaterally convert the shared result into a closed branch.

That protection is particularly valuable to contributors with little market power. An individual can publish under GPL knowing that a company may sell copies, support and services but may not distribute a covered derivative while withholding the corresponding source and freedoms required by the licence. The company receives access to a common asset. The community receives a legally meaningful assurance that distributed modifications will not simply become the company's exclusive property.

GPL is not a prohibition on commerce. It permits charging for copies and services. This was crucial to Linux. The kernel's early terms restricted distribution for money; the move to GNU copyleft removed that barrier and made commercial distributions compatible with continuing reciprocity [TORVALDS-GPL]. "Free" in free software concerns freedom, not an obligation to provide everything without price.

Copyleft also addresses a coordination problem. Contributors to a project without a single commercial centre may have entirely different motives. Some seek reputation. Some need the program for their own work. Some are employed by companies. Some are committed to a political philosophy, while others are simply solving a technical problem. GPL does not require agreement on a complete worldview. It requires compliance with a rule of transmission. A heterogeneous community can therefore cooperate around an artefact without first becoming one organisation.

Copyleft is not, however, an economy of contributors. It does not measure who created how much value, require royalties, or prevent one company from earning far more than others from services. This limitation is structural. The Open Source Definition requires free redistribution and forbids discrimination against persons, groups and fields of endeavour [OSI-OSD]. A software licence that allowed only cooperatives to use the code, or required large users to surrender a percentage of revenue merely because they used or redistributed it, would no longer be open source in the strict OSI sense.

That neutrality is one of open source's strengths. The code can be used in research, education, commerce, public administration and activities the author may dislike. Freedom does not depend on the identity or ideology of the beneficiary. The same neutrality means that a software licence cannot select a preferred economic order without giving up universality.

Classic copyleft also responds chiefly to distribution of the program. Software as a service allowed an operator to modify GPL software and provide functionality through a network without distributing a copy to the user. AGPL addressed a defined version of that gap by requiring an opportunity to obtain corresponding source when users interact remotely with a modified program. Even AGPL, however, governs the program. It does not distribute customer relationships, brand equity or revenue.

Copyleft is best understood as a compact of technical continuity. It says that distribution cannot be used to turn a covered commons into a closed derivative

that denies corresponding freedoms to recipients. It does not say that no participant may become more powerful by combining the commons with capital, infrastructure, data or brand. The modesty of its object explains both its success and its insufficiency in a platform economy.

The lesson for a decentralised brand is not to abandon copyleft but to imitate its clarity. Economic reciprocity needs a trigger as precise as distribution is for GPL. The proposal developed here makes that trigger commercial representation under the shared mark. Everyone retains the software freedoms. An organisation that chooses to benefit from the trust accumulated by the official ecosystem accepts transparent contributions, audit, quality obligations and a fair share of the common costs.

This arrangement preserves technical universality while placing reciprocity at the point where a reputational advantage is claimed. It is not "economic copyleft" in the software licence. It is a constitution of the name. To understand why that constitution should remain plural, copyleft must be compared with the other major tradition: permissive licensing.

2.2 BSD and the two meanings of freedom

BSD, MIT and ISC licences begin from a different intuition. The beneficiary's freedom includes the option to incorporate code into a proprietary product. Obligations are deliberately narrow: retention of notices, disclaimer of warranties and, in some versions, limits on using a contributor's name to imply endorsement. The code is offered as infrastructure available to the world, not as a commons required to reproduce itself under identical terms.

Table 3. Two meanings of freedom

Permissive freedom	Copyleft freedom
The present recipient may integrate the code into a proprietary product.	Later recipients of a distributed covered derivative must receive corresponding freedoms.
It maximises the options of the actor who has the code now.	It protects the options of the future community along the chain.
It favours rapid diffusion, embedding and standardisation.	It favours the return of distributed improvements to the protected commons.
It accepts closed branches.	It limits closure within the scope defined by the licence.

Neither column is superior in every setting. Permissive licensing maximises the immediate recipient's options. Copyleft maximises the probability that subsequent recipients receive similar options. One treats closure as an exercise of freedom; the other limits closure to preserve freedom over time.

The history of BSD demonstrates the power of low-friction diffusion. University code could be incorporated into commercial operating systems, devices and products that the original authors could never have built alone. Permissive licences reduce the legal cost of adoption and can help create de facto standards. For a library, protocol or reference component whose purpose is maximal spread, this may be exactly the right strategy.

Apache License 2.0 refined permissive licensing for corporate collaboration. It provides broad copyright permissions, an express patent grant, notice obligations and termination consequences for specified patent litigation [APACHE-2]. That predictability matters to organisations that cannot adopt infrastructure without evaluating patent exposure. The licence is therefore not merely a gesture of generosity; it is an instrument of procurement and industrial coordination.

Permissive licences are criticised because a corporation can take the code, build a proprietary product and withhold modifications. The description is accurate but incomplete. The author chose to grant that freedom. Proprietary adoption may greatly expand the utility and ecosystem of a component. A company may also contribute upstream for practical reasons: maintaining a private fork is expensive; influence over the common version is valuable; open development attracts talent; and visible contribution creates reputation.

A permissive licence does not fail when it produces proprietary use. Failure occurs when users or contributors are encouraged to assume a reciprocity that the text never promises. Marketing may invoke the moral aura of open source while the economic rights and customer channel remain concentrated. Institutional honesty requires separating legal freedom from distributive fairness.

Permissive licensing can also serve an anti-monopoly purpose. When several firms can implement the same protocol or reuse the same reference component, no single supplier needs to own the technical input on which the market depends. This possibility is strongest when the surrounding standards, trademarks and registries are also governed in a way that does not allow one beneficiary to convert broad code permissions into exclusive institutional control. A permissive commons can therefore widen competition even though it does not require private improvements to return.

Different permissive instruments also allocate risk differently. A public-domain-like dedication seeks maximal freedom from copyright restrictions but may provide a less developed patent framework. Apache License 2.0 offers explicit patent permissions and notice machinery designed for collaboration among organisations. MIT and BSD remain attractive partly because their brevity makes adoption and review inexpensive. Treating them as one moral category obscures the practical reasons for choosing among them.

None of these advantages creates distributive justice by itself. A project can become a universal dependency while its maintainers remain underfunded. A standard can reduce monopoly at the code layer while one company dominates hosting, certification or the customer relationship. This is not an argument against permissive licensing. It is evidence that the economic institution around the code must be designed rather than inferred from the generosity of the grant.

This distinction is central to brands. Several companies may lawfully use the same permissively licensed code while one company dominates the public name, documentation, conferences and distribution. Code remains plural while the market becomes monocentric. Conversely, a GPL project may protect source code while a separate entity controls the official mark and decides who may appear as part of the recognised ecosystem.

A decentralised brand need not force every component under strong copyleft. It may use Apache 2.0 for a runtime intended to become a broad standard, MPL for components where file-level reciprocity is useful, GPL or AGPL for infrastructure where technical reciprocity is essential, and separate instruments for data. Coherence comes not from imposing one licence on every artefact, but from explaining why each layer has a particular rule and how the layers interact.

This is licensing pluralism. An ecosystem can keep a protocol permissive to maximise interoperability, place a reference implementation under copyleft to preserve a common core, and reserve official use of the shared mark to nodes that meet public standards and contribute economically. Each instrument solves a different problem.

Pluralism also creates complexity. Each additional layer increases legal cost and the possibility of incompatibility. The remedy is not to invent a novel licence for every component. Standard licences should be used wherever they fit. Innovation should be concentrated in the relatively small set of documents needed for shared-mark custody, contribution, provenance, economic reciprocity and exit. The future needs predictable combinations more than it needs hundreds of exotic texts.

The permissive-versus-copyleft debate is often framed as a moral contest. For institutional design it is more useful to ask where optionality should be preserved. A permissive licence preserves the recipient's optionality. Copyleft protects the optionality of the future community. The Decentralised Brand Compact will attempt to preserve both at the technical layer and connect the market advantage of the official name to economic reciprocity.

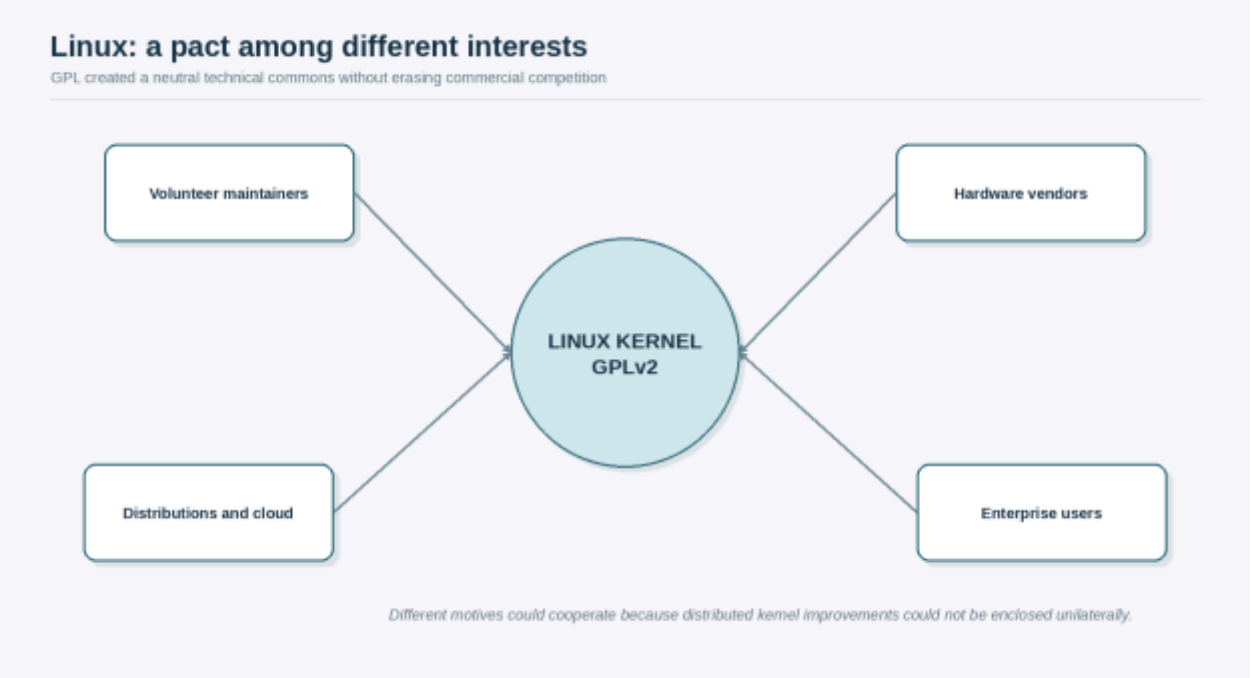
Linux demonstrated how powerful a protected core, a diverse commercial ecosystem and a separately governed mark can become. The role of GPL in that success should be understood without mythology.

CHAPTER 3. LINUX AND THE PACT AMONG STRANGERS

3.1 The move to GPL

Linux did not begin with the licensing arrangement that later defined it. Early terms permitted access and modification but restricted distribution for money. In the notes accompanying Linux 0.12, Linus Torvalds announced the change to GNU copyleft and explicitly connected it with the removal of that commercial restriction [TORVALDS-GPL]. The change mattered because it allowed commercial distribution while preventing a distributor from converting covered derivatives into a closed proprietary branch.

Figure 4. Linux as a pact among different interests



The figure places the GPL-licensed kernel at the centre of a heterogeneous system: volunteers, hardware producers, distributors, cloud operators, enterprise customers and research institutions. The licence does not make their interests identical. It establishes a dependable boundary within which they can cooperate without negotiating one multilateral contract and without permitting distributed covered derivatives to remove the freedoms of later recipients.

This rule enabled cooperation among competitors. Companies could invest in the kernel without signing a comprehensive agreement with every other participant. They knew that distributed covered modifications would remain available under GPL. The risk that one actor could turn common engineering into a wholly closed advantage was reduced. Economically, GPL functioned as an automatic non-closure treaty.

Commercial distributions became possible. A company could assemble the kernel, GNU tools, installation, updates, testing, support and certification into a product purchased by customers. It did not sell exclusivity over the code. It sold a reduction in complexity, a predictable release process, accountability and the assumption of operational responsibility. Discussions that equate free software with the absence of a business model miss precisely this distinction.

Linux also depended on complementarity with GNU. A kernel was not a complete system; compilers, shells, libraries and utilities made it usable. The expression "GNU/Linux" is part of a dispute over attribution, but it also records an institutional fact: ecosystems are built from commons produced by different communities with different histories and identities [FSF-GNU]. No single licence or organisation created the whole.

GPL sent contributors a credible signal. A programmer could invest time without assuming that the project would become the exclusive property of tomorrow's sponsor. The licence did not guarantee democratic governance and did not eliminate maintainer authority. It made legal continuity less dependent on the leader's future preferences. If the main project failed institutionally, the code could be continued elsewhere.

The right to fork disciplined the centre without abolishing it. Linux remained a project with strong leadership, demanding integration processes and technical hierarchy. Legal decentralisation did not mean the absence of authority. This is essential for decentralised brands. Node autonomy does not require every decision to be voted on by everyone. It requires that authority be bounded, justified and unable to convert the shared technical base into private property.

The present kernel is distributed under GPL-2.0-only with the Linux-syscall-note. Official documentation states that normal use of the system-call interface does not by itself impose the kernel's GPL terms on ordinary user-space programs [KERNEL-COPYING]. This boundary allowed a copyleft core to support applications under many licences, including proprietary ones. Success came from a membrane, not from forcing every component into one legal substance.

The Linux mark is another layer. The Linux Foundation administers sublicensing and protects the public from confusing uses [LINUX-MARK]. Anyone may use the kernel according to GPL. That does not mean anyone may represent any product as officially endorsed or use the mark in ways likely to confuse. The separation of code and name is a crucial precedent for the architecture proposed in this book.

The Linux arrangement does not distribute revenue directly to all contributors. Many contributions are funded by employers; others are voluntary. Companies selling cloud services, hardware and support capture very different amounts of value. Foundations finance coordination and infrastructure, but there is no automatic historical royalty for every contribution. At the scale of a global kernel, such a system might be undesirable or impossible. In smaller ecosystems

organised around an identifiable shared brand and customer market, some form of contributor distribution may be feasible.

GPL's role in Linux should not be reduced to the claim that "the licence made the project". Technical architecture, timing, the internet, x86 hardware, Unix culture, GNU tools, leadership and corporate engineering were indispensable. GPL nevertheless changed the form of success. It helped Linux become common infrastructure around which competitors could cooperate and compete, rather than merely the product of one supplier.

That institutional form is the point of departure for the future. If GPL made commercialisation compatible with non-closure of code, a new architecture may make open code compatible with reciprocity around the shared name. To design it, we need to understand what remained outside the Linux pact.

3.2 What GPL did - and did not - do

GPL is often described in two equally misleading ways. One treats it as a near-magical force sufficient to produce community and quality. The other calls it "viral", as though any nearby code were automatically infected. Both erase the actual scope of the legal instrument and the institutional conditions in which it operates.

Table 4. The real domain of GPL

GPL protects	GPL does not guarantee
Access to corresponding source for covered distributions under its conditions.	Distribution of revenue, customers or market power.
Freedom to modify and redistribute covered material.	Democratic governance or equal influence over technical decisions.
Continuity of the licence for covered derivatives.	Access to a mark, cloud channel, marketplace or certification system.
Commercial use and sale of free software.	Automatic funding of maintainers or integration of every contribution upstream.

The table separates actual protections from expectations that GPL does not satisfy. GPL preserves defined source and redistribution rights. It does not allocate commercial surplus, govern a trademark, compel upstream acceptance or equalise organisational power.

GPL reduces one form of free riding: taking a covered distributed derivative and closing the source in a manner inconsistent with the licence. It does not eliminate economic free riding. A company may comply fully, publish required source and build a profitable service through infrastructure, sales and support. From the perspective of GPL, this is not a defect or an infringement. Commercial benefit is deliberately permitted.

The category of "free rider" should therefore be used with care. A user who contributes no code may write documentation, report failures, enlarge the market, pay a distributor or legitimate a standard. A company operating a service may employ upstream engineers. Conversely, a sponsor that contributes extensively may acquire disproportionate influence over a roadmap. Contribution and capture are not opposites. One actor can do both.

GPL also does not by itself solve the network-service problem. Where no copy is conveyed, classic source-offer obligations may not be triggered. AGPL extends reciprocity to a specified form of remote interaction with a modified program. Some companies avoid it because of internal policy or concern about the scope of obligations. Projects seeking stronger protection against competing cloud services have adopted SSPL or other source-available terms, thereby giving up OSI-approved open-source status.

The timing of obligations matters. GPL does not generally require an organisation to publish every internal modification merely because it uses the

software. Its central conditions attach to defined acts such as conveying covered material, while AGPL adds its network interaction provision. This distinction is economically important. Companies can experiment, operate internal systems and contribute selectively without converting every private codebase into a public repository. The licence creates reciprocity at a boundary; it is not a general claim over everything an adopter writes.

The boundary between a covered derivative and a separate work can nevertheless be difficult in particular architectures, and slogans do not resolve it. The Linux syscall note offers a practical illustration of an explicitly stated membrane: ordinary user-space programs using the system-call interface are not treated as subject to the kernel's GPL merely because they run on Linux [KERNEL-COPYING]. That legal and technical boundary made coexistence with applications under many licences possible. Copyleft succeeded not by absorbing the entire ecosystem, but by protecting a core while allowing plural forms around it.

This also clarifies GPL's institutional achievement. It created a dependable cooperation bargain, not a complete constitution of the project. Maintainers still decide what is accepted; employers still direct paid work; foundations and companies may control infrastructure; users receive no automatic vote. The licence neutralises a particular route of enclosure while leaving governance and economic power to other institutions.

Data generated through use remain outside GPL's basic design. A service may run open-source code while accumulating logs, profiles, feedback and performance information that improve its offering. The code remains available; the data advantage concentrates. In artificial intelligence this becomes decisive. Model weights may be downloadable while training data, evaluation corpora and production feedback remain proprietary.

The customer relationship is also outside the licence. Organisations with distribution, contracts, certifications and a reputation for support can turn the same code into a far more credible offer. This is the complementary asset through which value is captured. Open source democratises a technical input; markets may reward the organisation that reduces uncertainty for the buyer.

GPL does not automatically protect the project's name. Trademark law serves a different purpose: reducing confusion and preserving a signal of origin, membership or certified quality. A fork may continue the code but often must adopt another name. That separation can be legitimate and necessary for consumers, yet it imposes an economic barrier. A fork may inherit the entire codebase and almost none of the accumulated recognition.

Formal exit can therefore be legally real but institutionally non-survivable. OpenTofu, OpenSearch and Valkey demonstrate that forks can become viable when companies, foundations and communities can rapidly transfer engineering, governance infrastructure and legitimacy [OPENTOFU; OPENSEARCH; VALKEY]. Smaller communities may possess the legal right to fork but lack the

resources to rebuild a domain, registry, release system, conference and customer channel.

A decentralised brand begins from this limit. It does not propose that a dissident fork should fraudulently continue under an identical mark. It proposes that custody of the name should be governed by a constitution containing a rule of succession. If the custodian violates the mission, attempts unilateral privatisation or loses the capacity to maintain the mark, an independent procedure can transfer custody or authorise a continuity mark. Exit then depends less on the financial ability to recreate every signal from zero.

GPL itself was a sophisticated middle path. It preserved commerce while prohibiting a defined form of privatisation. The next middle path must preserve genuine open source while limiting the privatisation of shared reputation. It cannot simply add a royalty to GPL, because a royalty on redistribution or a field-of-use restriction would change the nature of the software freedom. It needs a different legal trigger: the voluntary claim to official participation under the shared brand.

This idea becomes fully intelligible only after open source itself is examined as a technology of marketing and market formation.

PART II: WHEN OPENNESS BECAME A MARKET STRATEGY

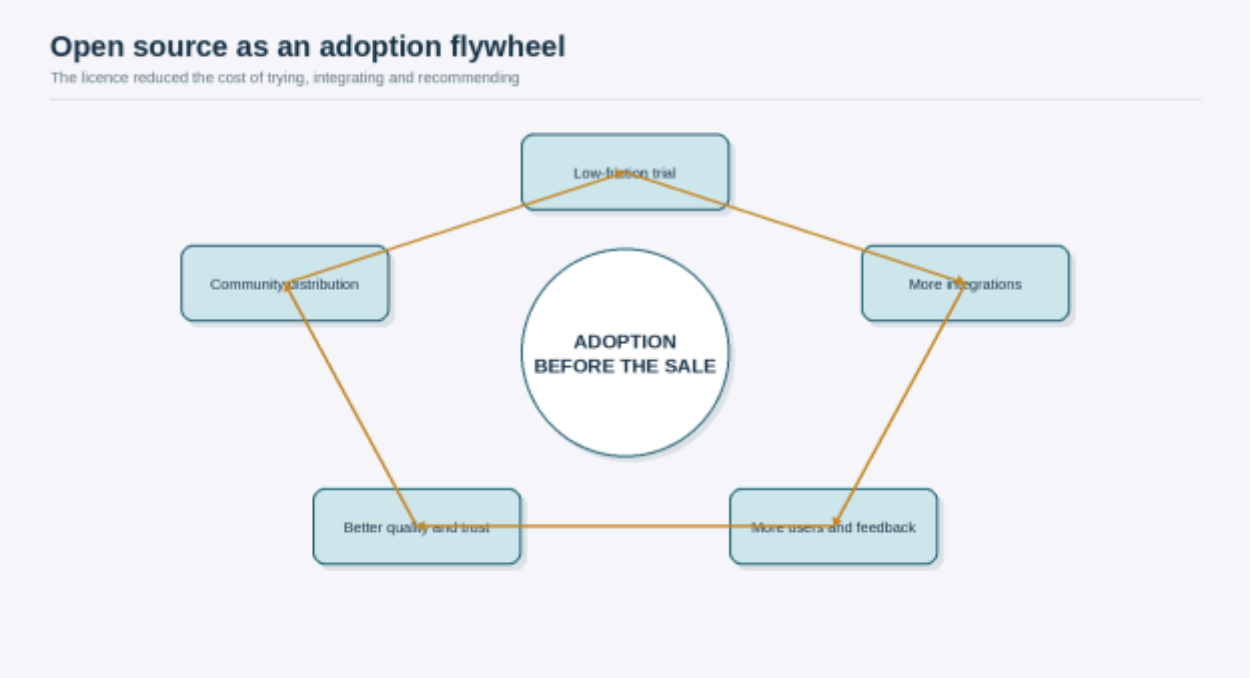
OPEN TECHNICAL COMMONS • ACCOUNTABLE INSTITUTIONS

CHAPTER 4. THE LICENCE AS A MARKETING ENGINE

4.1 Netscape and the rebranding of freedom

In 1998, the expression "open source" was chosen deliberately to present many of the practices associated with free software in language that businesses might find easier to adopt. The initiative emerged in the context of Netscape's decision to release the source of its browser, and the Open Source Initiative formed to define and defend the new term [OSI-HISTORY]. The change in vocabulary was not cosmetic. It made licensing an explicit component of market strategy.

Figure 5. Open source as a flywheel of adoption and distribution



The figure shows the intended mechanism. Publishing source attracts developers. Testing and contributions improve portability and quality. Adoption expands, and the company monetises complementary products and services. Openness is not the opposite of marketing. It is a channel through which a technology can enter an organisation before a commercial contract exists.

Netscape faced Microsoft's distribution power. A closed product, however good, could not easily compete with a browser bundled into the dominant operating system. Opening the source attempted to convert a distribution disadvantage into a global network of development and legitimacy. Early Mozilla documents described goals that included attracting developers, improving quality and extending distribution [NETSCAPE-MOZILLA].

Subsequent history did not confirm every commercial hope Netscape attached to the decision, but it validated the general strategy. Companies learned that a component could be made freely available in order to stimulate demand for a scarcer complement. A language, framework, runtime, database or model can become a standard while revenue appears in cloud operations, integration, hardware, proprietary data, enterprise services, certification or assurance.

This logic replaces the conventional question, "How can money be made if the code is given away?" The better question is, "What becomes scarce once the code is abundant?" Sometimes the scarcity is reliable operation at scale. Sometimes it is the capacity to remediate a vulnerability, integrate with a customer's environment, accept contractual responsibility, provide regulated documentation or obtain high-quality data. Licensing moves competition away from exclusive control of the copy and towards control of complementary assets.

Open source became a form of product-led growth before the expression was common. A developer can evaluate a project without a sales meeting, budget or negotiated licence. The technology enters an organisation from below, proves its usefulness and creates internal knowledge. Only when it becomes important do support, security, governance and service-level commitments become urgent. The open-source licence reduces the supplier's acquisition cost and the customer's cost of experimentation.

The arrangement also creates a signal of trust. Inspectability and the right to continue a released version reduce the danger of abandonment. A customer may never maintain a fork, but the legal capacity to do so changes the relationship with the supplier. "We will not trap you" becomes more credible when a version already received remains usable under a recognised licence regardless of the supplier's future pricing or ownership.

Open-source marketing can, however, become ambiguous. A company may foreground community, freedom and transparency while retaining unilateral control of the roadmap, trademark, production infrastructure, enterprise extensions and operational data. A public repository can create the impression of a shared institution when participants are in fact contributing to the distribution channel of a centrally controlled product.

The answer is not to forbid strategic use of open source. The strategy is legitimate and has created extraordinary public goods. The answer is institutional description. Who owns the mark? Who may relicense future versions? Who controls the registry, documentation and extensions required for production? What rights are granted with contributions? Can the community continue credibly if the sponsoring company changes direction? Which economic value returns to maintenance and infrastructure?

A decentralised brand would make such description a condition of official membership. Companies could use the code under its open-source licence without additional obligations. They could not use the official shared name to imply a reciprocity that did not exist. The mark regime would require disclosure of the

node's relationship to the commons, contribution to common costs, compliance with standards and acceptance of an audit and remediation process. Marketing could no longer separate the image of community from the location of budget and power.

The birth of the term "open source" shows that vocabulary has economic force. A recognised label reduces legal and cognitive friction and attracts a particular community. That is why the label is bounded by a definition. Terms that prohibit competition or fields of use may be commercially useful or ethically motivated, but they should not be sold as open source [OSI-OSD]. Terminological clarity is market infrastructure.

Future decentralised brands need equivalent clarity. "Decentralised" cannot mean merely that a token, repository or forum exists. The term should describe the effective distribution of custody, economic power, infrastructure, information and exit. Otherwise it will become another marketing word through which a centre hides control.

Netscape used licensing to attempt to change the competitive game. The broader lesson is that a licence can create a market rather than merely regulate a copy. The next step is to examine how adoption without a seller can turn a project into a standard and, sometimes, into infrastructure that another organisation monetises more effectively than its creators.

4.2 Adoption before the sale

Proprietary software normally enters an organisation through a contract. Open source can enter through a package manager. The difference seems operational, but it changes power. When the initial cost of adoption is close to zero, experimentation increases, and a project may be embedded in thousands of organisations before its initiators know who the users are.

Table 5. Two engines of adoption

Sale before access	Permission before sale
Sales, contract and procurement precede meaningful evaluation.	Download, evaluation and integration precede negotiation.
The supplier controls the channel of entry.	Developers, documentation and the community become the distribution channel.
Trust is produced mainly through contractual commitments and the supplier's brand.	Trust also comes from inspectability, recognised licences, standards and the possibility of fork.
The relationship begins commercially and at relatively high cost.	The relationship begins technically at low cost and is monetised later through complements.

The table contrasts persuasion-led adoption with permission-led adoption. In the first model, the supplier controls both access and message. In the second, the user can test directly, the community produces evidence and integration itself becomes proof. An open-source licence is therefore a distribution asset: it turns legal permission into immediate experience.

Lerner and Tirole explained open-source participation through a mixture of direct benefits, reputation, signalling of competence and professional opportunity [LERNER-TIROLE]. A community is not a homogeneous group of altruists. It is also a reputational market in which contribution can produce status, employment, contracts and influence. The licence creates the field; the project's name converts technical contribution into a social signal.

This creates a resource that copyright describes poorly: common reputational capital. Every resolved issue, review, tutorial, integration, conference talk and careful answer can increase confidence in the project's name. Many of these acts are not economically significant as separately protected works. Their cumulative effect on the brand can nevertheless be substantial.

The initiating company may benefit disproportionately because it owns the mark and maintains the official customer relationship. That can be justified by initial investment, coordination and risk. As the community creates a larger share of distribution and credibility, however, exclusive ownership of the name becomes less convincing as the only possible institutional order.

Open source also creates a distinctive path to standardisation. When many companies can integrate the same technology without bilateral negotiation, coordination costs fall. Libraries, training, integrations and a labour market grow around the project. Network effects can turn one implementation into the default.

At that point, the licence is not merely permission. It is legal infrastructure for an entire category.

This form of standardisation has a generous and a strategic face. A dominant project reduces fragmentation and total industry cost. Yet an organisation that controls most maintainers, the release infrastructure and the mark may shape the market even when the code is open. Competitors can fork, but the institutional cost of coordinating an alternative is high.

Recognised licences have marketing value of their own. Legal and procurement teams already know MIT, Apache License 2.0, GPL and AGPL. A bespoke licence forces a new review and may stop adoption. Elastic's decision in 2024 to add AGPLv3 as an option for the free portions of Elasticsearch and Kibana explicitly restored an OSI-approved open-source path while keeping Elastic License 2.0 as the default distribution licence [ELASTIC-AGPL]. Redis added AGPLv3 in 2025 alongside RSALv2 and SSPLv1 for Redis 8 and later, again acknowledging the value of a familiar open-source option [REDIS-AGPL].

The same dynamic explains why a proliferation of "almost open" licences can weaken distribution. Each custom text attempts to protect a particular economic niche, but it imposes analysis costs on every integrator. A company may gain control in the short term while encouraging the ecosystem to choose a simpler substitute or establish a fork under a familiar licence.

A decentralised brand should learn from this history. The technical commons should use recognised licences wherever possible. Novel legal design should be concentrated in the brand constitution, mark regime, contribution covenant and provenance system, where an unresolved problem actually exists. A software component should not acquire an exotic licence merely to make the surrounding institution appear innovative.

Adoption before sale creates a community before it creates an economy. That is both its strength and its danger. A project can become critical without developing a budget. It can attract thousands of users with no mechanism for funding maintenance. It can generate commercial opportunities for integrators who have no relationship with the people keeping the project alive.

The compact proposed later preserves adoption without a seller. No one needs a contract merely to test, run or fork the code. An additional agreement is required only for organisations that choose to sell under the official common name and benefit from its certification, channels and accumulated trust. Friction remains low at the technical edge; reciprocity appears at the commercial edge.

A competitor remains free to offer the same software under its own brand. That freedom preserves competitive pressure and makes exit real. Official membership is not a concession from a sovereign. It is voluntary participation in an institution that the member can help govern and that, in return, gives the member a recognised place inside a larger promise.

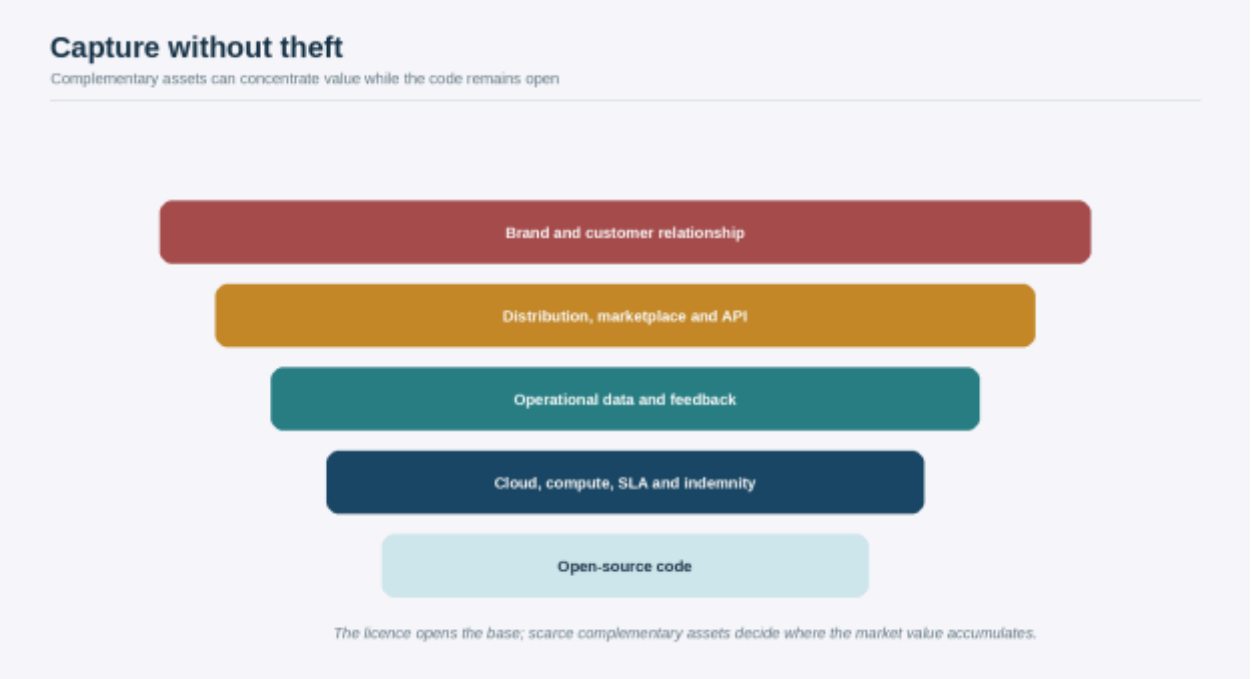
To see why that institution may be necessary, we must examine corporate capture. Capture usually does not occur by violating the software licence. It occurs through control of assets the licence never placed in common.

CHAPTER 5. CAPTURE WITHOUT THEFT

5.1 Cloud computing and complementary assets

The expression "capture of open source" can suggest an unlawful appropriation. In most important cases, no legal theft occurs. The licence permits the use. Capture emerges when one actor controls the complementary assets required to transform code into a service and the commons has no comparable mechanism for participating in the value created.

Figure 6. The complementary layers through which value is concentrated



Open-source code remains at the base of the figure. Above it sit compute and operations, proprietary data and feedback, identity and distribution, contractual assurance, the customer relationship and the brand. An actor that controls these layers can create a better service and capture most revenue without owning the underlying program exclusively.

Cloud computing magnified the asymmetry. An open-source project may be difficult to install, secure, scale, monitor and upgrade. A managed service creates genuine value. Customers pay for availability, integration, compliance and responsibility, not for possession of a copy. Classic GPL is not triggered merely because functionality is offered through a network, and permissive licences do not require publication of modifications at all.

For the project initiator, the outcome can be paradoxical. The code was published to accelerate adoption. Adoption succeeded. A cloud provider with global distribution offered the service more conveniently and captured the customers. The initiating company continued to carry a substantial part of

development cost while another organisation controlled the channel that converted technical utility into revenue.

Several companies responded by changing the terms for future versions. Elastic moved the previously Apache-licensed portions of Elasticsearch and Kibana to SSPL and the Elastic License in 2021, then added AGPLv3 as another option in 2024 [ELASTIC-LICENSING]. Redis moved from BSD-3-Clause for older releases to RSALv2 and SSPLv1 for the 7.4 line, and added AGPLv3 for Redis 8 and later in 2025 [REDIS-LICENSING]. HashiCorp announced in 2023 that future releases of several products would move from MPL 2.0 to Business Source License terms [HASHICORP-BSL].

These decisions have an intelligible economic rationale. The companies did not want to fund a product that a larger infrastructure provider could offer competitively while returning too little to its development. Relicensing exposed another asymmetry, however. Communities had contributed under an open-source identity while a concentrated rightsholder retained the capacity to change the conditions for future versions.

Contributor licence agreements sometimes support that capacity by granting broad relicensing rights. CLAs can serve legitimate purposes: clarifying title, enabling defence, and allowing a neutral foundation to maintain continuity. The Apache Software Foundation states that contributors retain ownership while granting rights needed to distribute and develop the contribution [ASF-CLA]. The institutional problem arises when an agreement presented as protection of the commons enables one company to convert aggregated community work into a unilateral licensing advantage.

A Developer Certificate of Origin follows a narrower logic. It records the contributor's representation that the contribution can be submitted under the project's stated licence without transferring an additional general power to relicense [DCO]. Even a DCO is not conclusive proof of title. Employment agreements, third-party code, patents and other rights may still matter. The lesson is not that one instrument is always correct. It is that the rights collected from contributors should match the public governance promise and should not silently exceed what continuity requires.

Cloud providers are not the only possible centres of capture. A company can control the package registry, marketplace, certification, conference, domains and communication channels. It can employ key maintainers and shape the roadmap without formally owning the project. It can reserve production-critical extensions. It can collect telemetry and customer data unavailable to the wider community.

Not every such arrangement is abusive. Registries require operators. Maintainers deserve salaries. Proprietary extensions can finance a common core. Operational data may contain confidential or personal information. A serious account does not demand that every asset be open. It asks that dependencies be

disclosed, concentration be bounded, and the community's identity not be used to disguise an economy it cannot influence.

Capture should therefore be defined functionally. It occurs when an actor uses the commons to consolidate a position from which others can no longer negotiate, exit or continue on reasonable terms, while common costs remain externalised. Profit alone is not capture. It becomes capture when complementary power is used to control the rules of the commons, drain its capacity or represent private sovereignty as community legitimacy.

A decentralised brand offers a middle instrument. A cloud provider remains free to use the open code and compete. To present itself as an official node, however, it contributes to a common fund, publishes defined compatibility and provenance information, meets standards and accepts audit. A provider that refuses may still offer the service under its own name. Commercial freedom remains intact; common reputation is not a free wrapper.

The arrangement does not guarantee that dominant providers will join. A global cloud may prefer its own brand. The ecosystem can still communicate a meaningful distinction to customers: technical compatibility is not the same as participation in the institution that finances, governs and assures the commons. The shared mark becomes information rather than a monopoly over functionality.

This may be more honest than loading anti-cloud restrictions into the code licence. Source-available terms can protect one company's revenue but fragment freedom and increase compliance uncertainty. A brand compact can protect an institution without prohibiting independent implementation or hosting. It shifts the competitive question from permission to run the technology to credibility in claiming membership.

The power of fork nevertheless remains essential. Recent cases show how a community can respond to licensing change through institutional continuity. Their cost also shows how incomplete a freedom based on code alone can be.

5.2 Relicensing and the expensive fork

HashiCorp's move to Business Source License terms led to the formation of OpenTofu, initially OpenTF, which entered the Linux Foundation. Its public case emphasised neutral governance, a predictable open-source licence and community continuity [OPENTOFU]. AWS initiated OpenSearch in 2021 from the last Apache-licensed Elasticsearch and Kibana code [OPENSEARCH]. After Redis changed its terms in 2024, Valkey emerged under the Linux Foundation under BSD-3-Clause [VALKEY].

Table 6. Forking as an institutional response

Fork or conflict	Institutional lesson
Elastic / OpenSearch	A cloud provider can turn a licensing change into a new project with its own technical and commercial identity.
Terraform / OpenTofu	A foundation, mark, release infrastructure and governance make a fork viable; source availability alone does not.
Redis / Valkey	Sponsors and neutral institutional support can reconstruct an ecosystem around a fork.
The failed fork	A legal right of exit is hollow when talent, distribution, infrastructure and reputation cannot migrate.

The table does not declare a moral winner. It shows a recurring pattern. A rightsholder protects monetisation; a community or competitor continues the last open version; a neutral foundation attempts to transfer legitimacy; the market decides whether the new name can accumulate enough confidence.

The right to fork is the core of open-source resilience. Without it, a unilateral change by the centre could end continuity. Forking is not free. Release engineering, security, documentation, roadmap, websites, package names, governance and distributor relationships must be rebuilt. Compatibility must be preserved without creating confusion under the original trademark.

The cost is partly technical and partly social. A mature project contains release credentials, build pipelines, security contacts, package registries, contributor knowledge, issue history and tacit rules that are not reproduced by copying a source tree. The new maintainers must persuade users that the branch will remain secure and compatible, persuade contributors that review will be competent, and persuade vendors that integration will not become a stranded investment. A fork is an entrepreneurial act performed under the pressure of institutional conflict.

Contribution architecture affects whether it can succeed. A modular codebase with reproducible builds, open infrastructure and a DCO or limited contribution grant is easier to continue than a project whose essential services, signing keys and relicensing rights are concentrated in one company. Technical debt can therefore become constitutional debt. The more continuity depends on

private systems that the community never controlled, the less substantial its formal freedom to leave.

The threat of a fork disciplines governance only when both sides believe continuation is possible. It is cheap to invoke the word and expensive to create a credible alternative. This asymmetry favours the incumbent, which already controls recognition and customer expectations. Precommitted succession rules, replicated archives, escrowed operational credentials and neutral custody can lower the cost without making every disagreement a schism. They turn exit from an improvised emergency into a known institutional possibility.

Successful forks therefore often have strong sponsors. AWS could finance OpenSearch. The Linux Foundation could provide infrastructure and neutrality for Valkey and OpenTofu. A smaller community may possess the legal right to leave without possessing the institutional capacity to survive the departure.

Relicensing also exposes the temporal problem of contribution. A developer may contribute for years under a permissive licence because the project is presented as shared infrastructure. An aggregated rightsholder can then publish future versions under more restrictive terms. The contributor retains rights in its own contribution but may be unable to recreate the complete product, governance system and brand.

Neutral foundation custody is one response. A limited contribution agreement or DCO-based process is another. The Decentralised Brand Compact combines them. Contributions to the technical commons are accepted under the project licence through a grant no broader than needed for distribution and defence. A fundamental licence change, sale of the shared mark, or conversion of the commons into a proprietary product requires a constitutional procedure involving several constituencies, public notice and a contestation period.

The compact also provides for continuity of the mark. This does not mean that two rival projects may use an identical name and confuse customers. It means that the legal custody of the mark contains a succession mechanism. If the custodian is dissolved, captured, persistently violates the stated purpose or ceases to exercise legitimate quality control, another qualifying institution can receive custody after a documented process.

Such a mechanism is unusual because trademarks are normally treated as assets of their owners. Purpose trusts, foundations, associations and other structures may in some jurisdictions separate legal custody from private benefit. The exact solution is jurisdiction-specific. The governing principle is more general: the holder of the shared name should have duties towards the declared purpose and should not be able to liquidate common reputation as an ordinary private asset.

The process must also protect consumers. Succession cannot be a private political manoeuvre in which two factions both claim to be "official". Notice, evidence, a transition period, objective standards and clear communication are essential. A continuity mark may be safer than transferring the exact mark in some

cases. The design goal is not to guarantee victory to dissidents. It is to prevent the present custodian from making legitimate continuation impossible by controlling every signal of identity.

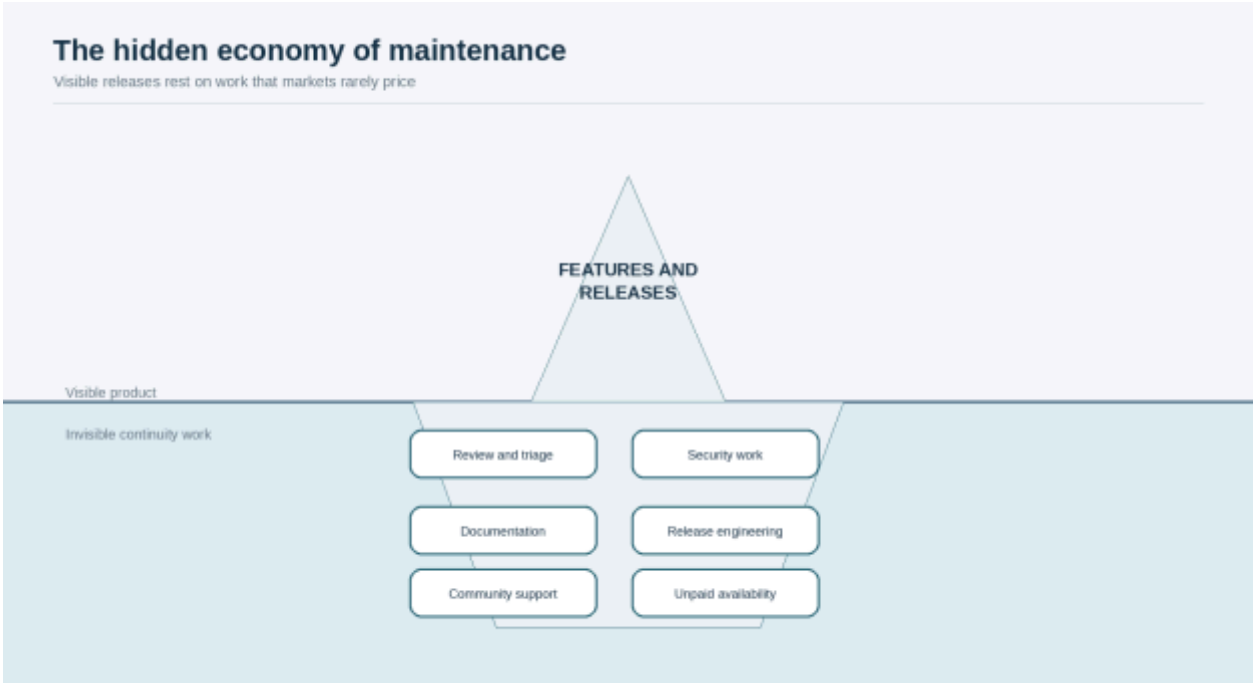
A fork is therefore not merely a copy of code. It is an attempt to move an institution. The next chapter examines the part of that institution most often ignored by licensing debates: the economy of maintenance and the asymmetry between corporate contribution and corporate power.

CHAPTER 6. THE ECONOMY MISSING FROM THE LICENCE

6.1 Maintenance and invisible labour

The public image of software creation is dominated by new features and visible releases. The everyday work that keeps a project alive is less dramatic: reviewing changes, reproducing failures, triaging issues, updating dependencies, responding to security reports, improving documentation, moderating discussion and telling users that a requested feature would damage the architecture.

Figure 7. The hidden economy of maintenance



The visible tip contains releases and features. Below the surface lie review, security work, documentation, testing, coordination, support and restraint. Reward systems based on easily counted commits observe the part most convenient to measure, not necessarily the labour that preserves continuity.

Research on software commons suggests that their characteristic "tragedy" is not overconsumption of code, because copying does not deplete it, but abandonment [SCHWEIK-ENGLISH]. A project fails when contributors can no longer justify the time, when review queues overwhelm maintainers or when the user population grows faster than institutional capacity.

Linux Foundation research has estimated very large organisational investments in open source, much of them in labour, while also showing how difficult companies find it to measure dependencies and contributions [LF-FUNDING-2024]. This produces double invisibility. Companies can underestimate the commons on which their products depend, while communities

can underestimate the contribution of engineers whose work is paid by companies.

Direct sponsorship, grants and fiscal hosts have improved the situation. GitHub Sponsors connects funders with individuals and organisations [GITHUB-SPONSORS]. NumFOCUS provides administrative, legal and financial infrastructure to scientific projects. OpenSSF and related initiatives fund security work on critical infrastructure. These mechanisms matter, but they do not fully change the structural relationship.

Donation is voluntary and often disconnected from value extracted. Visible, charismatic projects can attract money while an obscure but critical dependency remains unfunded. Maintainers may be forced to become marketers in order to secure resources, moving time away from technical work. Procurement systems can spend millions on downstream products while having no mechanism to allocate a small amount to upstream continuity.

Open core and commercial services internalise value more effectively, but can create two classes of contributor. The company decides which capabilities remain in the shared core and which are proprietary. Community members may improve the layer that produces adoption while revenue comes from layers over which they have no voice. The model can be legitimate when the boundary is explicit and participation is voluntary. It becomes misleading when "community" functions as reputational labour for a product whose economic constitution remains unilateral.

Maintenance also carries a liability problem. A vulnerability in a widely used library can affect millions of systems, while open-source licences normally disclaim warranties. The disclaimer is necessary to protect unpaid contributors. The resulting public-security problem remains. Organisations consuming the dependency have an incentive to finance prevention but face a collective-action problem in coordinating payment.

A decentralised brand can turn part of this responsibility into a common service. Official commercial nodes make contributions proportionate to the benefit of the mark, their scale or a published membership class. The common fund pays for maintenance, security, documentation, audit and shared infrastructure. Customers learn that the mark represents not only technical compatibility but participation in the costs of continuity.

The mechanism avoids two extremes. It does not charge every user a royalty for exercising open-source rights. It does not rely exclusively on donations. Reciprocity arises from voluntary commercial membership. A company may use and sell the software under its own name without paying the brand. When it claims the official name and the assurance attached to it, it joins the common budget.

Invisible work must be recognised in internal distribution. A fund that rewards only feature authors reproduces the biases of code-hosting platforms. The compact records different contribution classes: maintenance, review, documentation, research, design, governance, security, support, market

development, quality assurance and risk administration. Those classes should not be collapsed into a single supposedly objective score.

Contribution cannot be evaluated completely by algorithm. Effects may appear years later and depend on context. A small patch can prevent a major incident. A refusal to add a feature may preserve architectural coherence. The registry should therefore combine verifiable evidence with bounded human judgement, appeal, periodic review and distinct budgets for current labour and historical contribution.

The system must also resist Goodhart's law. When a measure becomes the route to payment or influence, participants will optimise the measure. Commit counts encourage fragmentation. Review counts encourage superficial approval. Popularity rewards marketing. The compact should use broad contribution categories, minimum maintenance allocations, conflict-of-interest rules, time decay where appropriate, privacy-preserving evidence and a human right of contestation. It should never pretend to derive moral worth from a metric.

Maintenance reveals the limit of an economy that confuses free access with absence of cost. Code may be free at the point of use and still require someone to pay for continuity. A mature institution does not hide this reality behind the romance of volunteering. It has a budget that tells the truth.

6.2 Corporations: real contribution, asymmetric power

A critique of corporate capture becomes implausible if it ignores that many major open-source projects are maintained substantially by paid employees. The Linux kernel, Kubernetes, compilers, databases and critical libraries benefit from investments that volunteer labour alone could not sustain. Companies contribute engineers, infrastructure, testing, security and demanding customers who expose real failures.

Table 7. The paradox of corporate contribution

Real contribution	Persistent asymmetry
Paid engineers and sustained maintenance increase capacity.	A company can maintain a continuous presence in decisions that volunteers cannot match.
Security, testing and integration operate at production scale.	Priorities may follow the sponsor's products and markets.
Foundations, events and common infrastructure receive funding.	Budget dependence can become informal influence.
Enterprise adoption legitimates and improves the project.	The mark and customer relationship often remain outside the commons.

The same corporate participation can enlarge the common good and concentrate power. A maintainer's salary improves sustainability; dependence on one employer can reduce autonomy. Sponsorship enlarges a conference; control of the conference can turn visibility into a commercial instrument.

Corporate motives are rarely altruistic. Firms contribute to reduce the cost of private forks, shape standards, recruit talent, improve their own products and sell complementary services. Those motives are compatible with a healthy commons. Good institutions do not demand purity. They align interests while making conflicts visible.

The problem appears when alignment becomes dependence. If most maintainers work for one company, the project can remain legally open while its roadmap becomes vulnerable to the employer's strategy. If a foundation depends on a few sponsors, the board may avoid decisions that disturb them. If participation in standardisation requires travel, lawyers and full-time delegates, technical meritocracy can become institutional plutocracy.

This power is difficult to measure because it does not appear in the licence. It moves through paid time, access to information, ability to attend meetings, control of infrastructure and the reputation of individuals. A constitution that counts votes may miss the fact that one actor drafts most proposals, owns the implementation capacity and employs the people who explain the options.

A decentralised brand should accept leadership and strong centres. A system in which every node has a permanent veto becomes slow and incoherent.

Power should instead be separated by function. No entity should simultaneously control the shared mark, treasury, identity infrastructure, common data, certification and final dispute process. Separation makes it harder to convert a legitimate contribution into total sovereignty.

One possible structure uses several chambers. Technical contributors choose representatives for standards and the common roadmap. Commercial nodes govern budgets for shared market and support activities. Customers, users or public-interest representatives participate in quality and impact oversight. A legal custodian protects the mark and constitutional procedure but cannot amend the purpose alone. Ordinary decisions remain delegated. Constitutional change requires agreement across constituencies.

No chamber structure guarantees fairness. Participation can become theatre, and organised factions can capture formal processes. The brand should therefore disclose who funds infrastructure, how many maintainers each major sponsor employs, how concentrated common revenue is, which services depend on one provider and where conflicts of interest arise.

Transparency is not enough unless information can produce consequences. The compact needs term limits, recusals, concentration thresholds, procurement portability and the practical ability to move common services. A sponsor may receive recognition and influence in the domains it supports. It should not receive perpetual ownership of the purpose or veto power over succession.

Competition law supplies another boundary. Commercial nodes may collaborate on interoperability, security and common assurance, but they must remain independent competitors. The shared institution must not coordinate prices, wages, customers, territories or output, use membership as a group boycott, or exchange competitively sensitive data without safeguards. Trade associations and standards bodies can be procompetitive, but the organisational form does not immunise coordination among competitors [FTC-ASSOCIATIONS; FTC-PRICE; EU-HORIZONTAL].

The anti-cartel rule should be constitutional, not an afterthought. Standards should be open and objectively related to the brand's promise. Certification should be available on transparent, proportionate terms and subject to appeal. Common purchasing, marketplaces and revenue pools require specific competition review. The compact coordinates trust and common infrastructure; it does not eliminate independent commercial judgement.

Companies can benefit from these constraints. A credibly governed brand reduces the risk that a competitor captures the centre or that a founder changes the rules after others have invested. Members can commit capital because continued participation does not depend entirely on one firm's goodwill. A constitution can be a protection for investment as well as for volunteers.

This is the middle ground that conventional open source did not try to construct. It neither excludes corporations nor asserts arithmetic equality between an occasional contributor and a company funding dozens of engineers. It requires

power to remain function-specific, contribution to remain visible and the shared name to remain unavailable for permanent purchase.

At this point the story must temporarily leave software licensing. The central object becomes the brand: the institution through which the public recognises a promise and through which participants transform contribution into commercial trust.

PART III: FROM CODE TO THE SHARED NAME

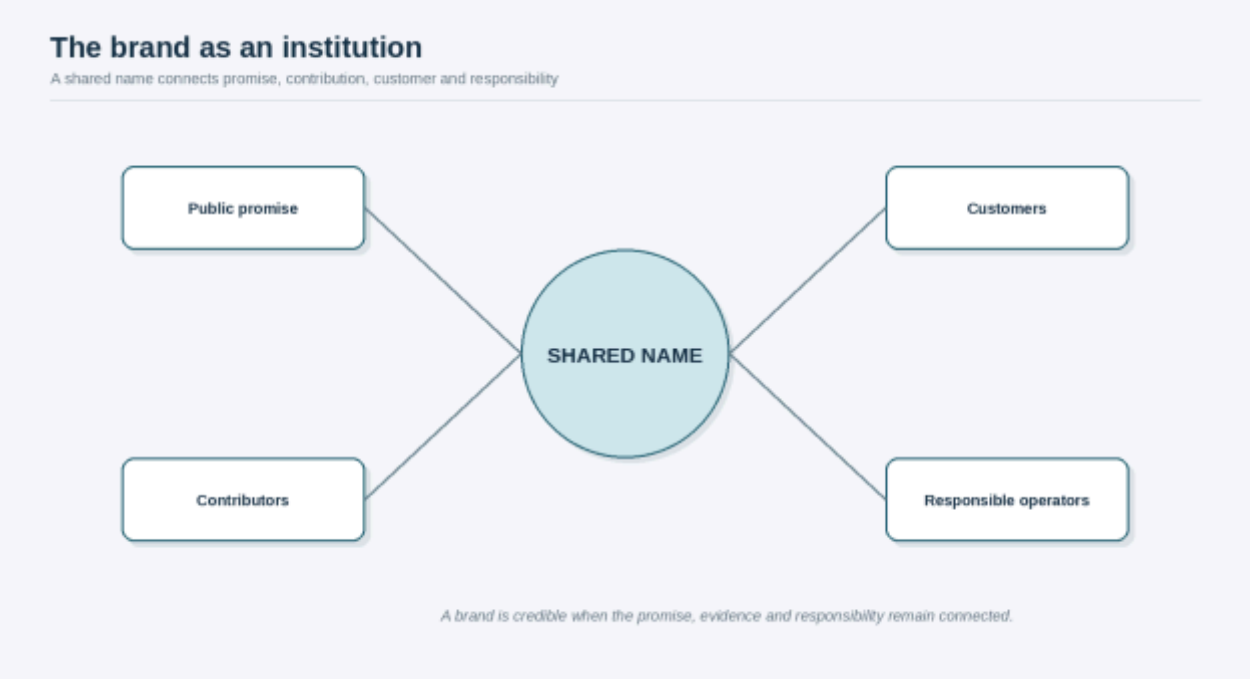
OPEN TECHNICAL COMMONS • ACCOUNTABLE INSTITUTIONS

CHAPTER 7. A BRAND IS AN INSTITUTION

7.1 The name as a promise

A brand is frequently described through visible elements: a name, logo, colour, slogan and campaign. These are instruments of recognition, not the essence. The essence is the promise that the public compresses into the name. When a customer chooses a product, it does not evaluate only the object present today. It uses reputation to estimate the future conduct of an organisation: quality, support, compatibility, responsibility and continuity.

Figure 8. The brand as a relationship among promise, contribution, customer and responsibility



The shared name stands at the centre of a relationship among memory, standards, experience and sanction. A brand works when the promise is stable enough to reduce uncertainty and contestable enough for a breach to produce reputational consequences. When the public cannot compare the promise with experience, brand becomes propaganda.

A trademark protects a sign against confusing uses, but the social brand extends beyond trademark law. It includes habits, technical architecture, communities, stories and expectations. A company may own the mark without owning the whole meaning. Employees, suppliers, users, critics and integrators create part of the reputation. Legal ownership of the sign nevertheless concentrates the authority to decide who may use the name and under which conditions.

That concentration has a strong justification. If anyone could use the same mark without control, customers could no longer infer consistent origin, membership or quality. An opportunist could sell a poor product using a reputation

built by others. Protecting the name protects both the public and the collective investment in reliability.

The problem begins when necessary custody is confused with sovereignty over the entire promise. A company can alter declared values, standards or conditions of participation without the consent of those who helped create the reputation. It can exclude contributors, sell the mark or convert a community into a marketing channel. The rightsholder may be acting within formal property rights while institutionally appropriating an asset produced by many.

The conflict is visible in open source. Code has distributed authorship while the project mark is often held by the initiating company or a foundation. When the foundation is neutral and credibly governed, the separation can protect the commons. When a company combines control of the mark, relicensing rights, infrastructure and customer channel, the community remains vulnerable even though the code is free.

Brand also allocates status. Contributors seek more than money. They seek recognition, professional identity, access, influence and the feeling that their work belongs to a meaningful story. A project provides a stage on which contribution becomes intelligible. When status is distributed by proximity to founders or capital, the community becomes a court. When status is distributed only by metrics, invisible work disappears.

A decentralised brand must therefore be an architecture of promise and status. It should state what the name means, who may change the standards, how contribution is recognised, and what happens when a node violates the promise. It cannot be merely a logo used by several companies. Without a constitution, the shared name becomes either incoherent or informally controlled by the strongest actor.

The promise must be precise enough to support verification and broad enough to preserve autonomy. If the brand dictates the complete method of operation, nodes begin to resemble franchisees. If it contains only moral aspirations, the customer receives no reliable signal. The common standard should focus on outcomes and behaviours that justify trust: interoperability, provenance, security, transparency, contribution to common infrastructure, truthful representation and effective remediation.

The mark must be governed, not merely "opened". Under systems that recognise licensed use, collective marks or certification marks, the owner or collective must exercise legitimate control over use of the sign and the relevant standards [USPTO-RELATED; USPTO-COLLECTIVE; USPTO-CERTIFICATION]. A passive custodian that sells approvals, ignores persistent failures or applies rules selectively undermines both legal protection and institutional legitimacy. Decentralisation therefore requires an accountable quality function, not the absence of one.

That function should be limited. Standards should be published, relevant to the promise, proportionate to risk and applied through evidence rather than

political favour. Auditors should be independent enough to challenge major sponsors. Nodes should receive notice, a right to respond, a remediation route and an appeal. Emergency suspension may be necessary where customers face immediate harm, but it must expire unless reviewed.

The shared mark does not replace each node's contracts. A node remains responsible for its customers, employees, products and representations. The mark states that the node belongs to an institution and meets a common assurance baseline. It should not ambiguously imply that every node is identical, that the custodian guarantees every product, or that all members are jointly liable for one another. Customer-facing terms must distinguish the node's warranty, the certifier's statement and any remedy supported by the common fund.

Diversity is a source of resilience. A centralised brand can optimise quickly but can propagate one error throughout the organisation. A decentralised brand can support several implementations, business models and local cultures while preserving compatibility at the boundaries. The public encounters an archipelago of recognisable nodes rather than a single platform controlling every relationship.

The shared name should be strong enough to attract customers and limited enough not to absorb each member's identity. A node can state that it is part of the brand without becoming property of the brand. Members retain their companies, customers, local assets and right to leave. This autonomy makes participation informative: membership is a renewed choice rather than captivity.

The decentralised brand is thus an institution between market and firm. It coordinates without employing everyone, protects a name without owning every product, and creates trust without turning every participant into a subsidiary. To function, it requires a definition more demanding than the usual rhetoric of decentralisation.

7.2 The decentralised brand

A decentralised brand is a shared symbolic, legal, operational and reputational structure used by autonomous actors who accept common rules without surrendering authority over their entire future. It may include a mark, standards, protocols, governance procedures, portable evidence, common infrastructure, contribution rules and mechanisms of exit. It is not a brand without a centre. It is a brand in which centres are custodians with limited mandates.

Table 8. Tests of substantive decentralisation

Dimension	Question that reveals effective control
Custody	Who legally controls the marks, domains, registries, keys, archives and constitutional documents?
Governance	Who can change standards, technical licences, membership terms and economic rules?
Economy	Who pays, who receives, who can block a budget and how concentrated are common revenues?
Infrastructure and data	Can nodes continue if a central provider, repository or data service disappears?
Voice and exit	Can a participant contest a decision or leave without erasure of its factual history?
Succession	Can the institution continue after the founder, sponsor or present custodian?

The questions examine different axes rather than offering a binary label. Custody locates assets. Governance locates rule-making. Economy follows money. Infrastructure and data reveal practical dependencies. Voice and exit test whether disagreement is survivable. Succession tests whether the institution can outlive its centre.

A distributed token is not sufficient. If founders control the mark, core code, treasury, website, data and communication channels, token voting may be decorative. A foundation is not sufficient. If its budget depends on one company and its board is inaccessible, legal form can mask effective control. A public repository is not sufficient. If only one team can make releases or controls all signing keys, technical autonomy remains weak.

The term therefore should not be used as a synonym for blockchain, a DAO, the absence of employees or the absence of leadership. Those features may alter one dimension while leaving the others concentrated. Substantive decentralisation asks a harder counterfactual question: what happens when a central actor fails, changes purpose or becomes hostile? If the mark, treasury, release process, identity system, data and appeal route all fail together, the institution has one practical centre regardless of how many tokens or repositories it displays.

A useful design test is replaceability. Legal custody of the mark may remain concentrated, but the custodian should be replaceable through a declared

process. One team may coordinate a release, but builds and evidence should be reproducible by others. A common service may host a registry, but nodes should be able to export the records required for continuity. Decentralisation is credible when essential functions are bounded and replaceable without erasing the institution.

This does not imply symmetrical power in every moment. Founders, maintainers, auditors and large nodes may possess different responsibilities and influence. The objective is to prevent temporary authority from becoming a permanent right to define the whole future. A plural institution admits leadership while retaining routes of correction.

Decentralisation is multi-dimensional and will often be incomplete. Some functions require concentration. A critical vulnerability cannot wait for a global referendum. Trademark enforcement needs a legal entity capable of acting. Coherent standards require maintainers with authority. The question is not whether power exists, but whether it is partitioned, auditable, replaceable and unable to present its own interest as the entire institution.

The decentralised brand differs from a cooperative, although it may use one. A cooperative distributes ownership and control among members according to a defined legal form. A decentralised brand may connect cooperatives, companies, foundations, independent professionals and public institutions. Its unity lies in the promise and constitution, not in a single corporate form.

It also differs from a DAO. A blockchain can record decisions, funds and credentials, but cannot by itself decide which contribution matters, whether a standard was met in context or how an exceptional human case should be repaired. Exclusively on-chain governance tends to confuse capital ownership with legitimacy. A decentralised brand can use verifiable records without turning a token balance into citizenship.

It differs from an open-source project. An open-source project principally organises an artefact, a development process and rights in code. A brand organises a relationship with the public and an economy around multiple artefacts and services. Linux, Apache, Wikipedia, cooperative federations and protocol communities provide partial laboratories, not complete templates.

The objective is not to abolish ownership. Nodes should be able to own local intellectual property, infrastructure, contracts and profit. Initial creators should be protected from a symbolic expropriation in which a community announces that every idea has automatically become common. Fair decentralisation does not require founders to surrender all control on launch day. It requires rights and transitions to be declared in advance, and it requires power over the name to mature towards custody as others become material co-authors of the promise.

The transition may be staged. During origin, founders hold concentrated authority to create coherence. During federation, validated nodes receive limited mark rights and representation. During constitutional maturity, the mark and key institutional assets move to a custodian whose mandate cannot be altered

unilaterally. The stages should be time-bounded or linked to objective thresholds, not left to the founder's discretion forever.

A decentralised brand should also allow more than one centre. Technical maintenance, legal custody, certification, data stewardship, treasury operations and dispute resolution need not reside in the same entity. A plural-centre architecture reduces the damage caused by capture or failure. It also creates friction, but some friction is the price of institutional resilience.

The most important test is whether membership remains voluntary in substance. A node that cannot leave without losing its customers, local IP, factual contribution history and ability to interoperate is not autonomous merely because the contract says it is independent. Meaningful exit does not require the right to continue using the official mark after departure. It requires the ability to build an independent future without institutional annihilation.

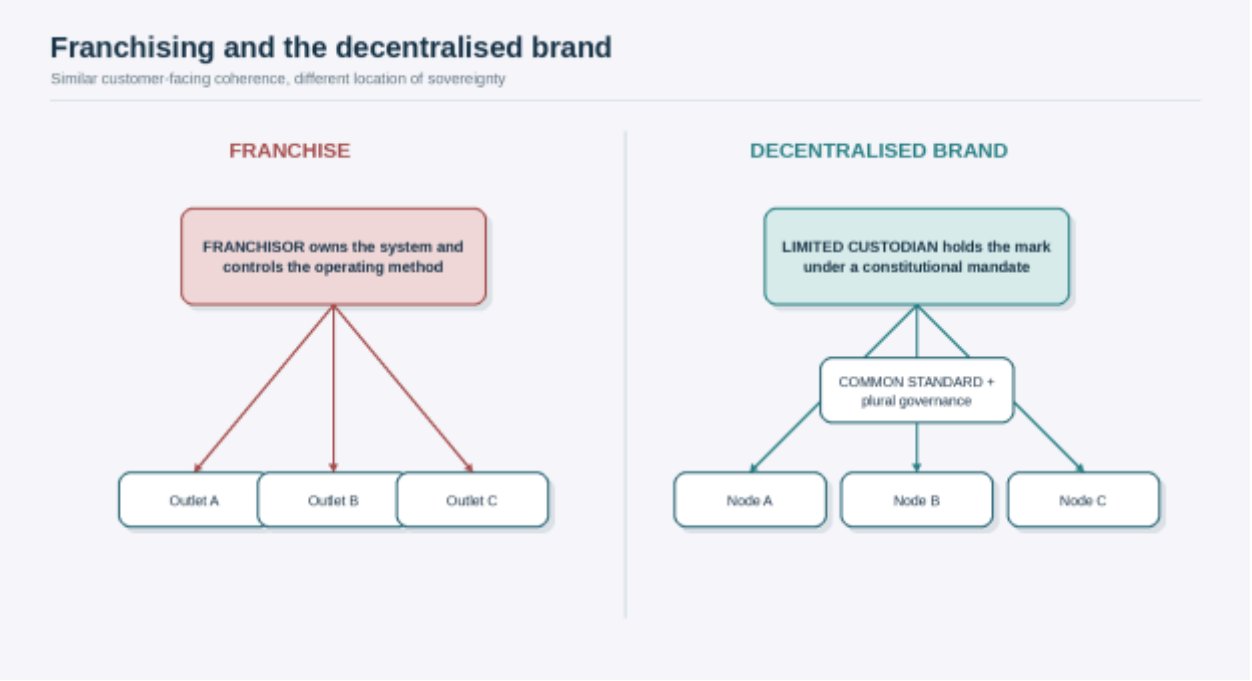
Success is therefore not measured only by growth. A decentralised brand succeeds when cooperation becomes easier, local initiative remains possible, gains from reputation are shared more fairly, the radius of failure is limited, and exit remains real. The ambition can be stated simply: not the abolition of brands, but the distribution of their power; not the destruction of centres, but the refusal to let one centre become the only future.

CHAPTER 8. FRANCHISING AND ITS INSTITUTIONAL RELATIVES

8.1 Franchising: entrepreneurial edges, sovereign centre

Franchising is one of the most successful technologies for distributing investment and operations without distributing brand control to the same degree. A franchisee invests capital, employs people and serves local customers. The franchisor owns the mark, defines an operating system and receives fees or royalties. The arrangement combines entrepreneurship at the edge with sovereignty at the centre.

Figure 9. Franchisor sovereignty and limited custodianship



A franchise resembles a star. Each unit relates primarily to the centre, while lateral relations are secondary. A decentralised brand resembles an archipelago. It has common standards and a mark custodian, but nodes can cooperate with one another, retain local assets and participate in constitutional change. The distinction is not the absence of a centre. It is the centre's limited and contestable character.

Franchising offers important lessons. A common standard protects customers. Training and support reduce entry costs. Aggregated marketing creates economies of scale. Fees fund brand infrastructure. Quality control makes the name credible. A decentralised brand needs versions of all these functions.

Classic franchising preserves a deep ownership asymmetry. A franchisee contributes to local reputation and the global value of the mark, but loses the right

to use the name on exit. It may spend years building a market that remains attached to the brand. The franchisor can impose detailed controls over design, suppliers, operating methods and marketing. Franchise law provides disclosure and other protections in many jurisdictions, but the centre normally remains the final interpreter of the promise.

The arrangement can protect originating intellectual property. Manuals, trade secrets and marks remain controlled, and the local operator receives conditional rights. It can also appropriate peripheral innovation. Improvements developed by franchisees may be transferred to the system without a proportionate participation in the durable value created. Intelligence is distributed while ownership of the institutional result remains concentrated.

A decentralised brand would retain discipline while changing the economy of custody. Official nodes would meet standards and contribute financially, but would also possess constitutional rights rather than only permissions granted by the centre. They would participate in governance, retain local intellectual property and receive a defined route for recognition and compensation when they contribute an asset to the commons.

The common standard should govern the promise rather than the entire method of operation. That preserves experimentation and may reduce the likelihood that the relationship resembles a classic franchise, but it does not produce a legal safe harbour. In the United States, FTC materials describe a franchise through the substance of a continuing commercial relationship involving association with a mark, significant control or assistance and a required payment; relabelling an arrangement as "membership" or "licensing" does not alter the analysis [FTC-FRANCHISE]. Other jurisdictions apply different statutory, disclosure, competition, agency and good-faith rules. The design must choose consciously between certification, collective membership, distribution and franchising rather than pretending that terminology decides the issue.

This is not an argument for evading franchise law. In some sectors the franchise form may be the most honest and useful. Strong operating control can protect safety and customer experience. The cost is reduced node autonomy and a clearer obligation to comply with the relevant franchise regime. A decentralised brand should use the minimum control required for the promise and accept the legal classification produced by the actual arrangement.

A further distinction concerns the direction of value. In a franchise, royalties usually flow from the edge to the centre, and the centre supplies the mark, system and marketing. In a decentralised brand, part of the flow returns to common maintenance and contributors under published rules. The custodian is not an unlimited residual beneficiary. It administers funds assigned to declared purposes and receives transparent remuneration for its own work.

Founders and central operators may still earn. They can be paid for originating IP, management, risk and services. They may own member companies and participate in commercial growth. The limit is that they cannot treat all future

value of the name as personal property after the ecosystem has become a material co-creator of the promise.

Franchising also teaches the importance of customer responsibility. A decentralised system cannot answer failure by saying that no one is in charge. Each node needs a clear contractual identity. The shared institution needs complaint intake, audit, remediation and the capacity to suspend or withdraw official status. Distributed power cannot mean evaporated liability.

The allocation of responsibility must be explicit. The node is ordinarily responsible for its product and customer contract. The custodian is responsible for truthful administration of the mark. The auditor is responsible within the scope of its engagement. The common fund may support remediation without thereby guaranteeing every member's debts. Customer communications should not create an unintended representation that certification is insurance or that every member is jointly liable.

A mature system may use territorial operators, master licences or local coordinators. Their mandates should be limited, reviewable and non-exclusive unless exclusivity is demonstrably necessary. The innovation lies not in rejecting every existing instrument, but in recombining them so that contribution, accountability and exit become more equitable.

Franchising proves that one mark can organise thousands of independent businesses. Its limit is that economic independence rarely extends to co-authorship of the promise. Other institutional relatives offer components for a different balance.

8.2 Cooperatives, collective marks, certification and standards

No existing form is identical to the decentralised brand, but several provide useful components. Cooperatives connect benefits and control to members. Collective marks identify goods or services as originating in qualifying members of an organisation. Certification marks attest that goods, services or providers meet published standards. Standards organisations create interoperability among competitors. Open-source foundations can hold marks and infrastructure in relatively neutral custody.

Table 9. Institutional relatives of the decentralised brand

Institutional relative	What it contributes
Franchise	Distributes entrepreneurship while preserving central sovereignty over the mark and operating system.
Cooperative	Connects member participation to control and surplus, but may require one ownership form.
Collective mark	Signals origin in members of an organised collective that meet admission standards.
Certification mark	Separates certification from production and can be available to qualifying non-members.
Consortium or open-source foundation	Can hold infrastructure and marks neutrally without automatically distributing economic value.
DAO	Makes voting and treasury operations programmable but does not eliminate wealth concentration or weak legitimacy.

Each form solves part of the problem and leaves another part open. Franchising provides coherence and monetisation. Cooperatives provide participation. Certification provides neutral assurance. Foundations provide custody. Open source provides technical continuation. The decentralised brand combines these elements without pretending that one legal form can perform every function.

Cooperative principles include member economic participation, autonomy and use of surplus for common development or benefits linked to members' transactions [ICA-PRINCIPLES]. These ideas can inspire a common fund and governance in which capital alone does not purchase every vote. A global technology brand may nevertheless include companies and institutions that do not wish to become members of one cooperative.

Federation provides an alternative. Nodes retain their legal form and join a shared entity for the mark, standard and common infrastructure. The federation can follow cooperative principles without converting every node into a cooperative. Voting can combine member equality with functional chambers that recognise real differences in responsibility and contribution.

Collective and certification marks offer especially important design choices. A collective mark identifies goods or services as originating in members who meet the organisation's admission standards. A certification mark indicates that goods, services or providers meet standards set by a certifying organisation, and use need not depend on membership [USPTO-COLLECTIVE; USPTO-CERTIFICATION; EUIPO-MARKS]. An ecosystem can use both: an official-node mark for institutional membership and a separate compatibility or assurance mark for technically qualifying implementations.

The separation reduces the conflict between membership and interoperability. A competitor can implement the open standard and seek certification without joining the political and economic compact. Official members accept additional duties of contribution, transparency and governance. The public can distinguish technical conformity from institutional co-responsibility.

A certification function needs independence. The entity setting or administering the certification should not manipulate standards to favour its own products, sell approvals without evidence or exclude competitors for reasons unrelated to quality. In the United States, certification-mark doctrine distinguishes the certifier from producers of the certified goods or services; exact requirements differ elsewhere [USPTO-CERTIFICATION]. A decentralised brand should preserve the principle even where it uses another legal form.

Standards organisations add another lesson. Competitors can cooperate on interfaces that enlarge a market and compete on implementations. Rules for essential patents attempt to prevent a rightsholder from blocking implementation, although FRAND regimes create conflicts of their own. A decentralised brand can adopt an analogous rule: intellectual property necessary to implement the common core must be available on predictable, disclosed terms, while differentiating IP may remain with the node.

Standards and certification also create competition risks. A shared mark must not become a cartel badge. Members should set their own prices, wages, customer terms and output. The institution should not allocate territories or customers, organise a group boycott, or use standard-setting to exclude a technically qualified rival. Sensitive commercial data should be minimised and aggregated where possible. Certification criteria should be relevant, transparent and open to appeal. Voluntary, independently administered certification that remains available to non-members is generally more defensible than a closed system designed to foreclose competition [FTC-ASSOCIATIONS; DOJ-CERTIFICATION; EU-HORIZONTAL].

Open-source foundations show how a mark can be separated from the initiating company. The Linux Foundation administers Linux mark sublicensing, and foundation policies make clear that a copyright licence does not automatically authorise implication of affiliation or endorsement [LINUX-MARK; LF-TRADEMARK]. This is the essential precedent for "open code, conditional name".

A foundation can still become a distant administrative centre. Large dues-paying members may enjoy privileged access, and a community may experience governance as opaque. Nonprofit status does not distribute power by itself. A compact must state not only who holds the mark, but how the custodian is appointed and replaced, how budgets are published, how conflicts are handled and what rights different participants possess.

CoopCycle offers an instructive experiment at the ethical edge. Its Coopyleft terms link commercial use primarily to cooperatives and limited-profit organisations, while a federation provides support and receives contributions linked to revenue [COOPCYCLE]. The model protects a solidarity economy but is not open source under the OSI definition because it discriminates by type of user.

The lesson is not that open source should be abandoned. It is that economic reciprocity can be designed explicitly. For a general decentralised brand, the restriction can be moved from the software licence to the right to use the official mark. Any company may use the code. Only organisations accepting the compact may sell as official members. Technical freedom and a reciprocal economic network can then coexist.

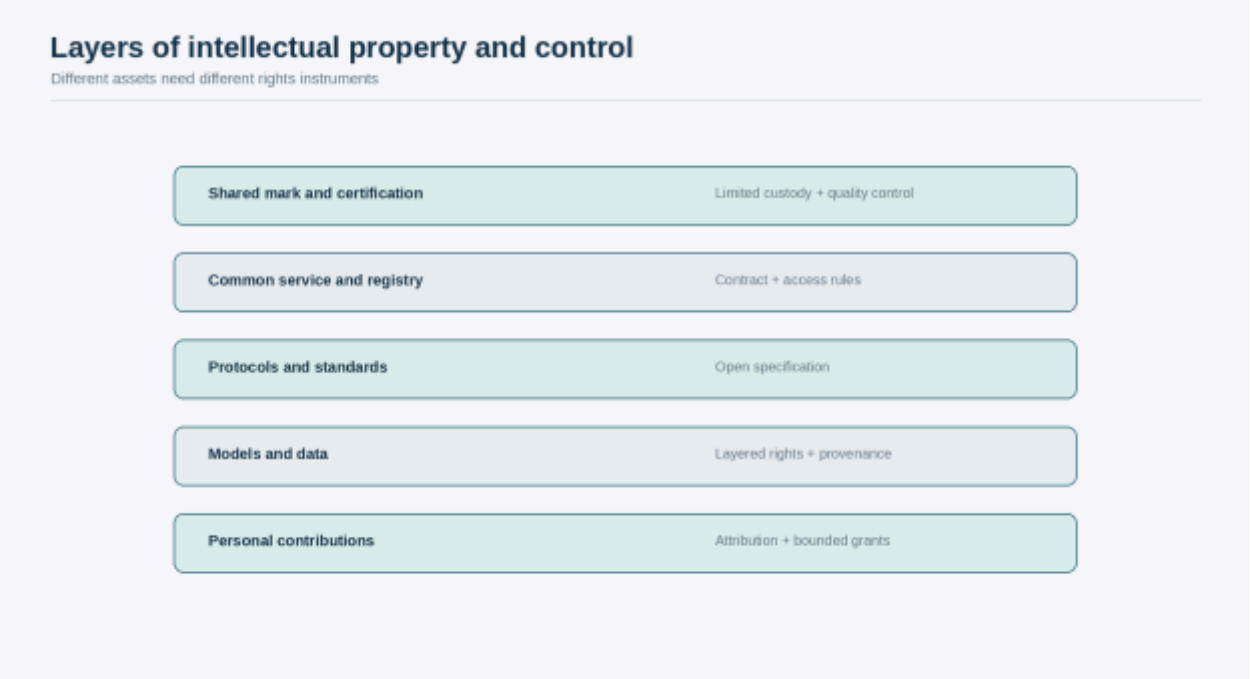
No institutional relative supplies the whole solution. Franchising contributes discipline; cooperation, participation; collective marks, identity; certification, neutrality; standards, interoperability; foundations, custody; and open source, the right of continuation. The decentralised brand is their constitutional combination.

CHAPTER 9. CONSTITUTIONAL OPENNESS

9.1 Layers of intellectual property and control

A community that declares everything to be common will quickly discover that different assets cannot and should not be treated alike. Code can often be redistributed. Data may contain personal or confidential information. A mark requires controlled use to preserve its signal. An AI model can have downloadable weights and non-redistributable training data. Visual identity may be licensed only to official nodes. A local optimisation may remain a member's trade secret.

Figure 10. Separating rights according to institutional function



Four areas are distinguished. Background IP existed before participation and remains with its owner. The technical commons contains contributions deliberately accepted under stated licences. Node IP is created locally and may remain proprietary while respecting open interfaces. Custodied assets include the shared marks, constitutional documents, standards, registries and continuity keys held for the common purpose.

The separation protects originating creators. A founder can license technology to the brand without assigning every use. A contributor can submit code under Apache License 2.0 and retain the right to use the same work elsewhere. A node can build a commercial extension so long as it preserves required interoperability and does not falsely represent the extension as mandatory to the common standard.

The commons must be defined positively. It is not everything created near the project. It consists of artefacts accepted through a procedure with clear provenance and a stated licence. This prevents the dispute in which a contributor

believes rights were retained while the centre assumes it received a broad assignment.

Contribution instruments should be no broader than their function. A DCO records a representation about the right to contribute. Where a CLA is needed, it should explain the grant, preserve the contributor's ownership where intended and constrain unilateral relicensing. Neither instrument can guarantee that the contributor actually controls every right. Employer ownership, third-party material, patents, moral rights and local law may matter. The compact should therefore provide representations, a correction procedure and a way to withdraw or replace material whose title proves defective.

The mark is different. It should not be universally licensed because its function is to indicate origin, membership or verified conformity. A mark licence can require quality standards, truthful description of affiliation, audit and contribution to common costs. Suspension or withdrawal should follow a process with notice and a right to respond, subject to narrow emergency protection.

Data require separate governance. A brand may share aggregate statistics, compatibility lists and evaluation data without centralising every customer record. The architecture should favour minimisation, local processing and selective exchange through membranes. Common data need a stated purpose, rights basis, retention rule, access policy and response to correction or deletion requests where applicable.

In AI, rights in a dataset, training code, tokenizer, weights and derivative model may differ. A licence applied to a collection does not automatically clear rights in every item within it. A rights holder can grant only rights it controls. Personal data, publicity, confidentiality, database rights and cultural or contractual restrictions may remain. A distilled or merged model may also inherit obligations from a base model or from terms governing outputs. Each significant artefact therefore needs a rights and provenance manifest.

The manifest does not cure defective title. It records what is known, the legal basis asserted, evidence, uncertainty and restrictions. A "rights-cleared" label should be reserved for a documented process and defined scope, not used as a magical conclusion. Where deletion, retraining or machine unlearning is promised, the promise must state technical limits and the circumstances in which it is actually feasible.

Protecting creators does not justify making the commons dependent on revocable private permission. If a component becomes essential to the standard, the institution should obtain rights stable enough for continuity. A node should not introduce a proprietary critical dependency and then use withdrawal to capture the ecosystem. The standard may require a perpetual implementation licence, source escrow, a reference alternative or a transition period for essential components.

The brand should not confiscate every innovation. Nodes need room for differentiation and profit. An economy in which every improvement is compulsorily common may suppress investment and turn the institution into a consumer of its

members' creativity. Reciprocity should focus on the core required for cooperation and on contributions deliberately offered to the commons.

This architecture is more complex than a single LICENSE file, but the complexity already exists. The choice is whether it remains implicit and advantages the actor with the largest legal team, or becomes explicit in standard formats. The compact should use established software, data and content licences where they fit, connected by a common manifest and constitutional rules.

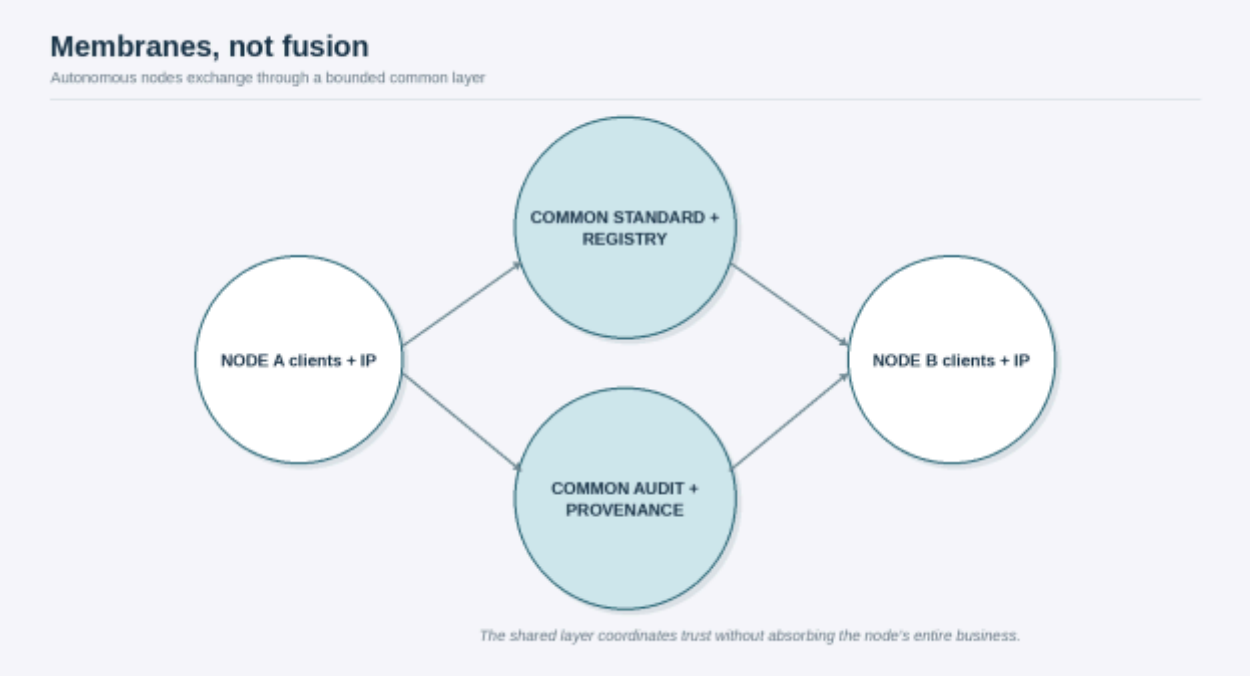
Constitutional openness means that every restriction has a declared function. The mark is controlled to protect the promise. Data access is limited to protect people, contracts and legitimate interests. Some extensions remain proprietary to finance innovation. The technical core remains open to preserve interoperability and exit. A custodian cannot use a restriction justified in one layer as permission to dominate every other layer.

This separation is the foundation of the middle path. It permits creators to retain and monetise IP, allows the community to continue common assets, and lets customers recognise an assured standard. Static rights are not enough. The ecosystem must exchange information without becoming one platform. That is the purpose of membranes.

9.2 Membranes, autonomy and exit

Interoperability is often presented as an absolute good. When systems can communicate, users gain freedom and competition can increase. Total interoperability can also dissolve boundaries that protect autonomy, confidentiality and diversity. A decentralised brand needs membranes: selective interfaces that permit exchange without absorption.

Figure 11. Controlled exchange among autonomous nodes



Autonomous nodes are connected through narrow interfaces. They exchange verified identities, compatible artefacts, contribution evidence and information needed for service. They do not automatically transfer all internal processes, customer data or commercial intelligence to a centre. The membrane preserves the distinction between cooperation and merger.

A centralised platform tends to make interoperability a one-way street. Nodes send data and activity to the centre, and the centre returns conditional access. As dependence grows, migration becomes expensive. A decentralised brand should reverse the presumption: customers and operational data remain with the node unless a defined common purpose requires an exchange.

A membrane is not a poetic synonym for an API. It is a rule about what crosses an institutional boundary, for which purpose, under whose authority and with what possibility of reversal. It may be implemented through a protocol, contract, data schema, cryptographic credential or human procedure. Its function is to support cooperation while preserving the identity and decision capacity of each side. A system with no exchange is isolated; a system in which everything flows to one centre is absorbed.

The distinction matters economically. A node can contribute compatibility evidence without surrendering its customer list, contribute aggregate safety

information without exposing confidential cases, and prove membership without transferring ownership of its local identity. The common institution receives enough information to sustain the promise, but not enough to make every participant dependent on one database. Data minimisation becomes a constitutional principle rather than only a privacy technique.

Open standards are one membrane. They allow products to cooperate without one supplier owning the interface. Portable identity and evidence form another. A contributor should not lose factual history when changing nodes. Signed provenance forms a third. Artefacts carry information about origin, version, licence and transformation without exposing all private content.

Exit is the test of autonomy. A member should be able to leave the brand and continue under its own name. It retains its legal entity, local IP, customer contracts subject to their terms, and all rights granted by open-source licences. It loses the right to claim current official membership and must remove signs likely to create confusion.

That loss should not become a total punishment. The compact should not impose unjustified non-competes, confiscate local extensions or erase professional history. A registry can confirm the period and nature of past contributions after departure without suggesting that the former member remains certified. Reputation should be portable as evidence, not as a permanent entitlement to the mark.

Exit becomes more complex where common investments are involved. A node may have access to shared leads, infrastructure, documentation and confidential data. The compact should define transition, portability, confidentiality and the return or deletion of controlled material. The objective is continuity and protection of legitimate interests, not revenge.

Voice and exit support each other [HIRSCHMAN-1970]. When departure is impossible, voice can become ritual. When departure is effortless and no loyalty or repair exists, each conflict fragments the system. The brand should make disagreement bearable through reasons, appeals, mediation and spaces in which nodes may experiment locally.

Loyalty in this framework is not obedience. It is the willingness to invest in repair because the institution has previously proved that criticism can matter. Voice becomes credible when decision-makers must give reasons and when an adverse decision can be reviewed by a body not wholly controlled by the same interests. Exit becomes credible when a member can carry the factual evidence, technical assets and contractual relationships necessary to begin again. Each mechanism keeps the other honest.

Portability must be specified rather than celebrated. Source code may be portable under an open-source licence while operational data are trapped in a proprietary format. A contributor may own a commit history but lose access to reviews and evidence of maintenance. A company may retain customer contracts yet be unable to replace credentials or export the configuration needed to

continue service. The compact should identify these dependencies early and state which records, formats and transition assistance are part of legitimate exit.

Customer continuity deserves special care. Departure should not become an excuse for misleading use of the old mark, but customers should receive adequate notice, access to their own data, and a clear explanation of which assurances cease and which obligations continue. The brand protects trust best when it makes institutional separation intelligible rather than attempting to erase the departing node.

Fork is the technical form of exit, but it need not be the first response. A membrane architecture permits compatible branches, regional profiles and alternative implementations inside the brand. The standard defines what must remain common; the remainder can vary. Plurality reduces the pressure to win every central decision.

Succession is the extreme form of exit. When the custodian fails, the community must be able to continue the institution, not only the code. Continuity marks, escrowed keys, replicated registries, independent archives and a procedure for appointing a successor make this possibility operational. Without them, "decentralisation" continues to depend on the uninterrupted virtue and capacity of the centre.

Membranes also protect against systemic error. If every node depends on the same cloud, AI provider and database, the brand may be legally distributed and technically monocultural. Standards should permit multiple implementations and test interoperability among them. Redundancy appears inefficient until the centre fails.

The architecture prepares the transition to artificial intelligence. Models and agents tend to concentrate data, compute and operational control. They can also reduce the administrative cost of provenance, audit and contribution accounting. AI can become the next invisible centre or the infrastructure that makes constitutional openness manageable at scale.

PART IV: ARTIFICIAL INTELLIGENCE AND THE END OF THE SINGLE LICENCE

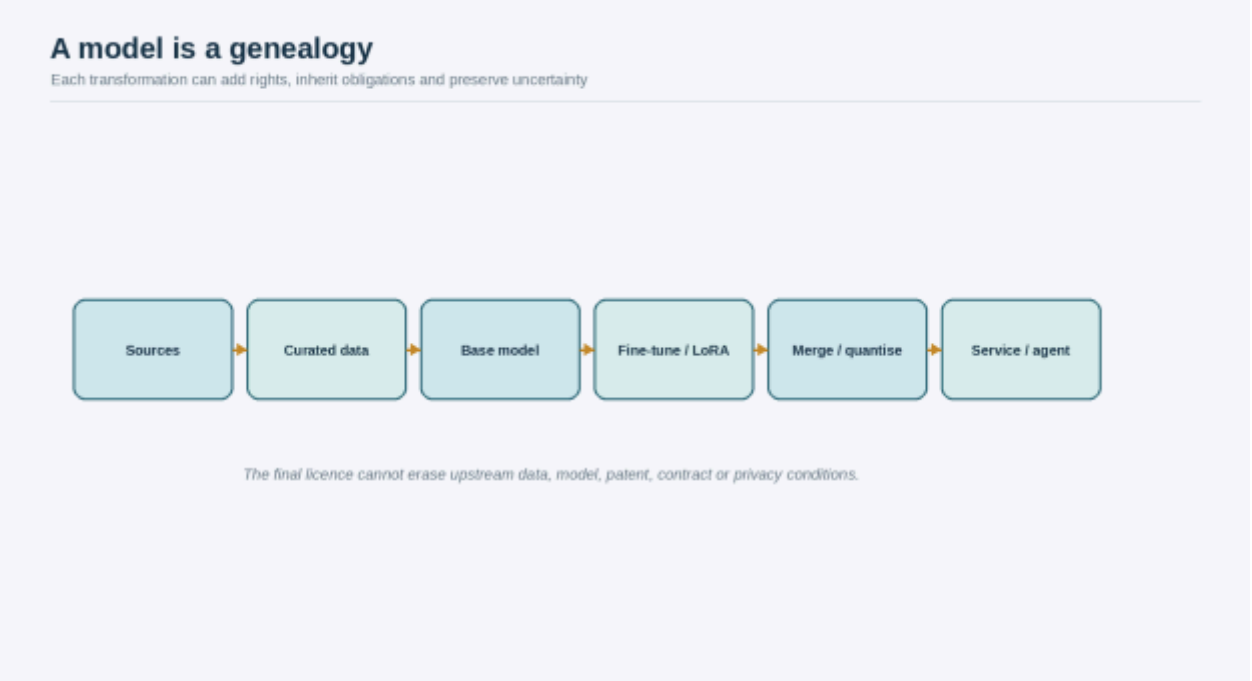
OPEN TECHNICAL COMMONS • ACCOUNTABLE INSTITUTIONS

CHAPTER 10. A MODEL IS A GENEALOGY

10.1 The AI licensing stack

In conventional software, legal review often begins with a repository and its dependency tree. In artificial intelligence, the final artefact is the result of a genealogy. Collection tools, raw sources, curated datasets, filters, labels, tokenisers, architecture, training code, recipes, model weights, intermediate checkpoints, adapters, evaluations and inference servers may have different authors and different terms. The service delivered to the customer can add proprietary access conditions, acceptable-use policies and contractual allocation of risk.

Figure 12. The technical and legal genealogy of an AI system



The path runs from sources and data preparation through training, base model, adapter or fine-tune, merge or quantisation, and finally a service or agent. Each transformation may introduce new rights, inherit obligations and preserve uncertainty. A licence for code does not grant rights in data. A dataset licence does not automatically define the status of weights. Terms selected for a derivative cannot extinguish upstream conditions.

The expression "model licence" compresses this structure too far. Even the legal status of weights is not uniform across jurisdictions. Some terms may be enforced as copyright permissions, some as contracts accepted on access or distribution, and others may depend on database rights, patents, trade secrets or access controls. The safe description is an AI rights stack: a set of instruments governing distinct artefacts and operations.

Technical operations need to be named. "Modification" is too general for fine-tuning, low-rank adaptation, merging, pruning, quantisation, distillation, synthetic-data generation and continued pretraining. DeepSeek-R1, for example, publishes its main code and weights under MIT terms and permits distillation, while distilled variants based on other model families retain the relevance of the underlying model terms [DEEPSEEK-R1]. Google's Gemma terms expressly address model derivatives and, in their current version, certain uses involving outputs and hosted services [GEMMA-TERMS]. The legal result depends on the exact artefact and version, not the vendor's family name.

Vendors increasingly maintain mixed portfolios. Some models are distributed under Apache License 2.0 or MIT-like terms, others under community licences, responsible-use conditions, research-only terms or service contracts. OpenAI's gpt-oss models are described as open-weight and distributed under Apache License 2.0, while the surrounding use policy and service ecosystem remain separate layers [OPENAI-GPTOSS]. "Open-weight", "open model" and "open-source AI" should therefore be used as different claims.

The Open Source AI Definition 1.0 attempts to adapt open-source freedoms to AI systems. It requires freedom to use, study, modify and share for any purpose and describes the preferred form for modification through data information, relevant code and model parameters [OSI-OSAID]. It does not require every raw dataset to be redistributed, because law, privacy and access conditions may make that impossible. It requires sufficient information about provenance and processing for the freedoms to be meaningful.

The compromise remains contested. Without the actual training data, complete recipe and intermediate evidence, a model cannot be reproduced or audited in the same way that source code can be rebuilt. Downloadable weights permit local use and adaptation but reveal little about how capabilities, bias, memorisation or legal risk arose. Fully open AI is therefore a more demanding category than open weights.

OLMo provides an important reference point. Ai2 publishes data, code, recipes, checkpoints, evaluations and weights so researchers can examine learning as a process rather than only experiment with the final product [AI2-OLMO]. The cost is substantial, and complete data openness cannot be achieved in every domain. The example nevertheless establishes an epistemic benchmark against which narrower forms of openness can be described honestly.

AI-specific licences also pursue responsible-use restrictions. OpenRAIL instruments permit forms of access, modification and commercial use while transmitting restrictions on specified uses, and explicitly distinguish themselves from OSI open source [OPENRAIL]. Such terms can express values and create contractual obligations where a binding agreement exists. Their enforcement against an offline model can be difficult, and they combine two different questions: what recipients may do with an artefact and which conduct society should prohibit or regulate.

A decentralised brand need not force every model into one licence. It can maintain a fully open reference core for audit, use open-weight models for local adoption, and procure proprietary services where regulated performance or liability requires them. The obligation is accurate labelling. An official node should state which layers are open, which are merely accessible, which restrictions apply, and who assumes responsibility.

The stack also includes output. A provider may decline to claim rights in generated output, but that does not guarantee that the output is copyrightable, free from third-party rights or suitable for every downstream use. Outputs can become training data for another model. Some terms regulate distillation or model development using outputs. A customer needs to know whether it may create a dataset, train a competing system, redistribute outputs, or use them in a regulated product.

The complexity cannot be eliminated by a universal licence. It can be made navigable through a manifest that answers operational questions: may the artefact be run, copied, fine-tuned, adapted, merged, pruned, quantised, distilled, used to generate training data, redistributed, hosted or exposed through an API? Each answer must be tied to an artefact, version, source of authority and level of confidence.

Within a decentralised brand, the manifest becomes part of the public promise. An official node cannot merely say "we use open AI". It identifies the model, version, known provenance, applicable terms, significant restrictions and the responsible operator. The mark certifies disciplined disclosure and assurance, not metaphysical purity.

This discipline can itself become a market advantage. In regulated industries, customers pay for provenance, repeatability, documentation and an accountable allocation of risk. Code and weights may be freely available while verified lineage, audit, support and indemnity remain premium services. Once again, value moves from the abundant copy to scarce trust.

10.2 Open weights are not enough

Publishing model weights changes the distribution of power in a meaningful way. Organisations can run models locally, keep sensitive inputs under their control, avoid dependence on one API and adapt systems to their tasks. For research, resilience and technological sovereignty, open weights are real progress. They should not be confused with openness of the entire system.

Table 10. A practical scale of AI openness

Openness level	Practical capacity
API access	Permits conditional use of a service without control over the model or continuity of access.
Open weights	Permits local inference and often fine-tuning, but not reconstruction of training.
Open code and weights	Permits inspection of the distributed runtime and modification of the released model.
Data information and training recipes	Improves auditability without necessarily allowing recreation of the exact corpus.
Accessible data, code, checkpoints and logs	Allows much deeper study of the process and substantially stronger reproducibility.
Fully open science	Aims at independent reproduction, audit and continuation of the complete system.

The table describes a scale, not a simple moral hierarchy. Different levels support different forms of autonomy and scrutiny. A responsible label should locate a system on the scale instead of compressing all distinctions into the word "open".

Weights enable experimentation with the final artefact but do not reveal everything that entered it. Copyrighted works, personal data, representational harms and memorised sequences may be difficult to evaluate. A model card is valuable but remains a provider's account. Without evidence of provenance, the user accepts risks that model terms normally disclaim.

Open weights can also be a platform strategy. A vendor releases a capable model to build adoption, tooling and developer familiarity, while monetising hosted inference, larger models, enterprise features or complementary services. This is the direct analogue of open-source marketing. The accessible model becomes an integration standard and can draw users towards the provider's closed layers.

The strategy is not inherently objectionable. The market needs an accurate account of power. A community licence may contain usage restrictions, attribution conditions, scale thresholds or different rights for very large actors. A model can be widely downloadable and commercially useful without being open source in the strict sense.

Permission also does not provide compute. A model may be legally available to everyone and practically usable only by organisations with sufficient

hardware, energy and expertise. Power can reconcentrate in infrastructure providers. Licensing distributes legal capacity; it does not distribute the material resources required to exercise it.

This is why openness should be described as a profile rather than a badge. One system may offer excellent local deployability but little information about training. Another may publish data and recipes while requiring compute beyond the reach of most laboratories. A third may be reproducible at modest scale but carry restrictions that make commercial use uncertain. Procurement and research should compare dimensions such as runnability, modifiability, provenance, reproducibility, redistribution, compute accessibility and continuity of governance.

Open-weight releases can decentralise inference while leaving training and agenda-setting highly concentrated. A community may adapt the last stage of a system without being able to inspect the decisions that shaped its corpus, architecture or safety behaviour. The accessible artefact is still valuable, but the market should not infer from local execution that the entire epistemic process has become public.

Fully open reference models have a special role even when they are not the most capable. They provide measurement infrastructure. Researchers can examine checkpoints, reproduce selected steps, test data interventions and compare claims that would remain invisible in an API-only system. Public institutions may therefore support fully open models for the same reason that they support scientific instruments and standards: not because every deployment must use them, but because an auditable baseline constrains the claims of the rest of the market.

For decentralised brands, autonomy cannot therefore be measured by weight availability alone. The ecosystem needs multiple inference providers, portable formats, common evaluation, local fallback and the ability to replace one model without losing service identity. A brand should not become the exclusive distributor of a vendor's model when the vendor can change access or terms.

A technical membrane can separate the promise from a particular model. The standard describes capabilities, traceability, safety evidence and interfaces rather than mandating one supplier. Nodes can choose different models, and shared evaluations can verify whether the common promise is met. This reduces monoculture and permits technical competition inside one identity.

Open weights do not resolve remuneration of creators whose works contributed to training. The United States Copyright Office's 2025 report emphasised that fair-use analysis is contextual and that voluntary licensing markets should be allowed to develop, while targeted collective mechanisms could be considered if demonstrated transaction failures emerge [USCO-AI3]. The report is a policy analysis rather than a judicial determination, and litigation will continue to shape the boundary.

The European Union uses a different architecture. The DSM Directive establishes text-and-data-mining exceptions, including a general exception

subject to rights reservation in appropriate form, with machine-readable reservation especially relevant online [EU-DSM]. The AI Act requires providers of general-purpose AI models to maintain a copyright-compliance policy and publish a sufficiently detailed summary of training content [EU-AI-ACT]. These regimes transform licence and provenance information into operational compliance infrastructure.

A decentralised brand can reduce compliance costs for smaller nodes by maintaining common manifests, rights-reservation handling, evaluation protocols and standard contract clauses. It should not claim that a shared process eliminates legal risk. It creates evidence, consistency and a clearer allocation of responsibility.

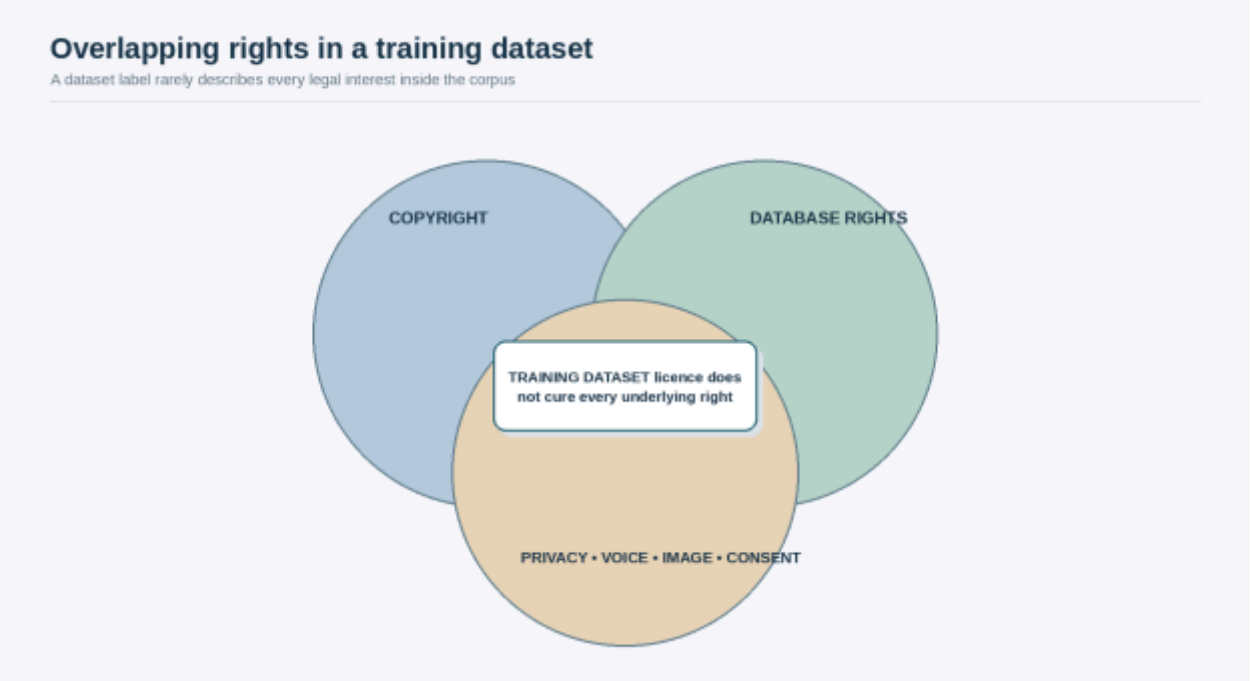
The future is unlikely to be a contest between one "open" and one "closed" category. It will be an ecology of fully open scientific models, open-weight commercial models, restricted community models, proprietary services and domain models trained on negotiated data. The role of licensing is to make the differences legible and the transformations traceable.

CHAPTER 11. DATA AND THE OBLIGATION THAT DISAPPEARS

11.1 Licensing data for training and use

Code usually has an identifiable repository and a bounded set of authors. A dataset can aggregate millions of works, databases, personal records, images, voices and materials obtained under different contracts. A licence applied to the collection does not prove that the organiser controls every relevant right in the contents.

Figure 13. Overlapping rights in a training dataset



The figure separates copyright in individual works, rights in selection or arrangement, European database rights, access contracts, data protection, image and voice rights, confidentiality, trade secrets and other sectoral obligations. One label on the dataset cannot clear the entire chain.

Creative Commons licences govern copyright and related rights that the licensor is able to grant. Their consequences for AI training depend on jurisdiction and on whether the relevant act requires permission [CC-AI]. CC BY requires attribution. CC BY-SA can raise questions about adaptation and downstream sharing. CC BY-NC limits acts that require permission to non-commercial purposes. CC BY-ND is especially difficult where training or a transformation is legally treated as adaptation. In retrieval-augmented generation, attribution to retrieved documents can be practical. In pretraining, the causal connection between one source and one output is much more diffuse.

Open Data Commons distinguishes database rights from rights in individual contents [ODC-LICENSES]. CDLA-Permissive allows broad computational use

and defines results, including models and insights, without automatically imposing its obligations on those results [CDLA]. The Montreal Data License proposed an operation-oriented taxonomy covering access, representation, training, research, deployment and commercialisation [MONTREAL-DATA]. These instruments demonstrate why data permissions need to describe the act, not only the object.

"May use the dataset" is insufficient. Terms should state whether they permit pretraining, continued training, fine-tuning, embeddings, evaluation, retrieval, publication of weights, synthetic-data generation, distillation and commercial deployment. They should specify duration, territory where relevant, downstream transfer, deletion of raw data, audit and what happens to already trained models when the agreement ends.

Rights of represented persons remain distinct. A photographer may license a photograph while the person depicted retains data-protection, publicity or personality rights. A voice recording may involve the performer, producer and people identified by the content. Medical data may be available for a defined research protocol but not for an unrelated commercial system. A copyright licence is not a lawful basis for personal-data processing.

Cultural, communal and sectoral interests may also matter even where conventional copyright analysis is inconclusive. Indigenous knowledge, confidential professional records, children's data, biometric features and security-sensitive information should not be treated as ordinary web text merely because they can be scraped. A credible brand should state which categories it excludes or subjects to enhanced review.

Licensing markets are developing by sector. Publishing, music, images and specialised databases have different collecting institutions and contractual traditions. The Copyright Office observes that direct and collective voluntary licensing can reduce transaction costs, while feasibility differs across markets and collective systems can create their own competition and governance risks [USCO-AI3]. There is no single credible licence for the entire internet.

For decentralised brands, data can become a valuable common asset and a shared liability. Nodes may contribute curated corpora, evaluations or aggregated feedback and receive economic recognition. Centralising all data would recreate the extractive platform. The compact should prefer granular rights, local custody and distributed computation, including federated learning or local evaluation when technically and legally appropriate.

Data contributors should be treated as economic participants rather than as free raw material. An organisation offering a carefully cleared corpus may receive payment from the common fund, access to resulting models, governance rights over defined reuse, or a negotiated combination. Terms should avoid unlimited assignment for unforeseeable future purposes.

Withdrawal must be described realistically. Removing raw files, ceasing future use and deleting a document from a retrieval index are different from removing its influence from an already trained model. Retraining or machine

unlearning may be technically possible in some systems and prohibitively expensive or ineffective in others. The agreement should distinguish these remedies and avoid promising impossible erasure.

Synthetic data complicate genealogy. One model generates outputs that become a corpus for another. The originating model's terms may regulate distillation or model training. Outputs may reproduce protected or personal material. "Synthetic" does not mean legally clean. A manifest should identify the generating model, version, prompting or sampling regime where relevant, filters, human review and the rights basis asserted for downstream use.

A fair data economy cannot calculate the exact contribution of every document to every response. It can combine payment by corpus, category, licensed use and measurable outcome. Retrieval can support direct attribution and per-source accounting. Pretraining is more likely to require negotiated corpus licences, sectoral collecting arrangements or pooled remuneration than token-level micropayments.

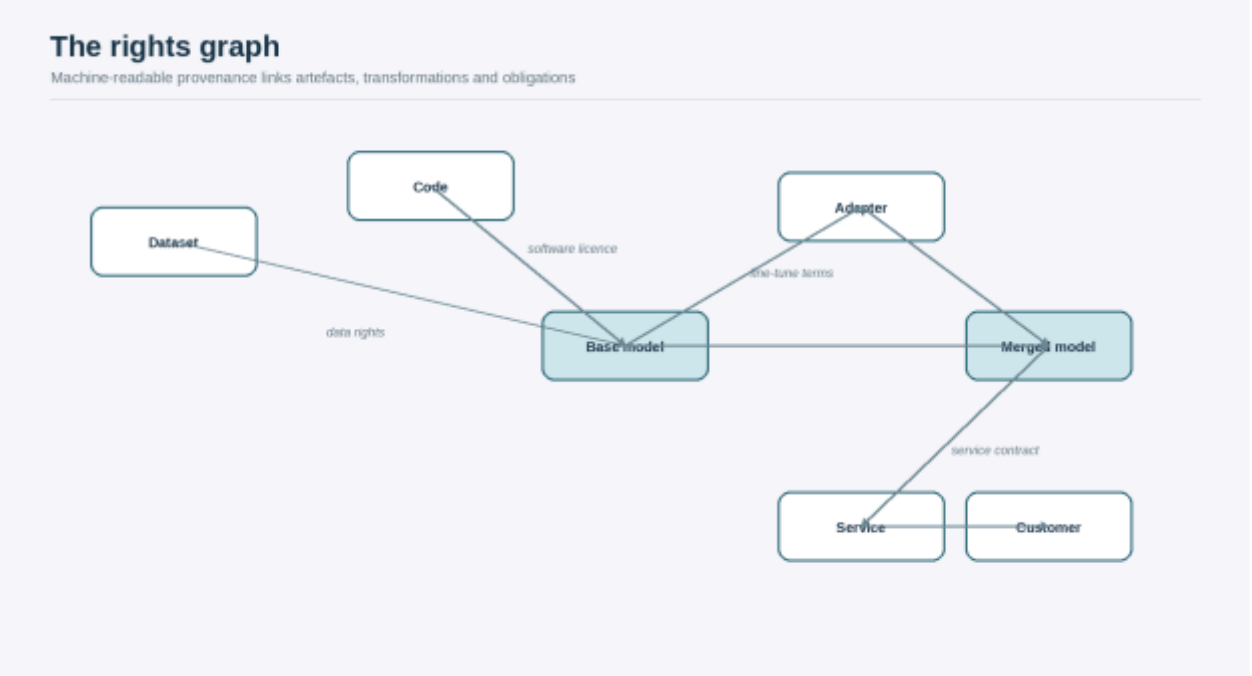
A decentralised brand may organise voluntary collective licensing for contributors who have granted a clear mandate, subject to applicable collective-management, labour and competition law. It can aggregate corpora with documented rights, negotiate defined uses with model developers and distribute revenue under audited rules. It cannot license works or rights it does not control. The mark then signals not merely technical performance but a more credible data chain.

In AI, "open" without provenance can become a transfer of risk to the user. The next critical infrastructure is therefore not another literary licence alone, but a structured system through which rights claims and obligations remain attached to artefacts as they are transformed.

11.2 Machine-readable provenance

A person can read two or three licence texts. An AI system may depend on thousands of packages, datasets, models and transformations. Compliance can no longer be administered only through text files, spreadsheets and institutional memory. Rights information must become structured data.

Figure 14. A machine-readable graph of artefacts, transformations and obligations



Nodes represent datasets, code, base models, adapters, quantised variants, containers and services. Edges represent provenance, transformation and inherited obligations. The final application has no single origin. It has a path that must be queried.

SPDX standardised licence and component information for software, and version 3 adds profiles capable of representing AI, datasets, builds and provenance [SPDX-3]. CycloneDX can represent models, datasets, dependencies, training approaches and configuration in an AI or machine-learning bill of materials [CYCLONEDX-MLBOM]. These standards do not decide disputed legal questions. They make automation and evidence possible.

A useful manifest identifies an artefact by version, hash and signature. It states the exact licence or terms, parents, type of transformation, operations claimed to be permitted, restrictions, downstream obligations, data categories and output policy. "Derived from" should not be an undifferentiated relation. The graph should distinguish continued pretraining, supervised fine-tuning, LoRA, merging, pruning, quantisation, distillation and generation of a synthetic corpus.

Machine-readable does not mean perfectly executable. Clauses can be ambiguous, and compatibility depends on facts and jurisdiction. An agent can identify a likely conflict, missing evidence or a term requiring interpretation and

route it for human review. The objective is not to replace counsel. It is to stop routine loss of information.

Recent empirical preprints report substantial licensing disorder in AI supply chains. One study of repositories, models and datasets found large numbers of artefacts with no declared licence and widespread conflicts along combined chains [AI-LICENSING-RISK]. A later preprint analysing more than two hundred thousand dataset-model-application chains reported that obligations such as attribution, share-alike and non-commercial restrictions frequently disappeared from downstream metadata, a phenomenon its authors call licence laundering [LICENSE-LAUNDERING]. These are preprints and should not be treated as final measurements, but they identify a problem visible in practice.

A derivative author cannot grant rights it never received. Publishing a new model under MIT or Apache terms does not cleanse incompatible upstream conditions. The manifest must distinguish the licence selected for the new contribution from inherited obligations and unresolved claims.

Provenance has economic value. A corpus with verified authority, consent where required and traceability becomes a premium asset. A model with a documented genealogy can enter regulated procurement more easily. A provider can sell audit and indemnity because it possesses evidence rather than merely optimistic representations.

In the European Union, this infrastructure is becoming urgent. General-purpose AI obligations under the AI Act applied from 2 August 2025; the Commission's enforcement powers for those obligations are scheduled from 2 August 2026, while models placed on the market before 2 August 2025 have a later compliance deadline [EU-GPAI]. A shared brand that standardises manifests, copyright-reservation handling and audit can lower the compliance burden for smaller nodes.

The brand registry need not centralise all underlying files. It can hold hashes, signed declarations, evidence references and links to custodians. Sensitive material remains local. Auditors verify evidence under controlled access. This is a membrane: sufficient information for trust without universal collection of content.

Contributor provenance can use the same logic. Commits, reviews, decisions, documentation, incident work and accepted data contributions can be signed and associated with portable identities. AI can classify and summarise, but individuals need the right to challenge attribution and protect sensitive information. The registry is institutional memory, not a system of total surveillance.

A rights graph can also support economic distribution. Revenue from an official product can be linked to the components and contribution classes it uses. Perfect causal attribution will remain impossible. The institution can still define explainable rules: active maintenance, essential components, verified data contributions and shared infrastructure receive weights, subject to review and appeal.

Machine-readable licensing changes the operational role of the licence. The legal text remains authoritative. The manifest becomes the interface through which build systems, procurement agents and auditors query the rights position. Agents can reject incompatible combinations, preserve notices, propose alternatives and record the reason for a decision.

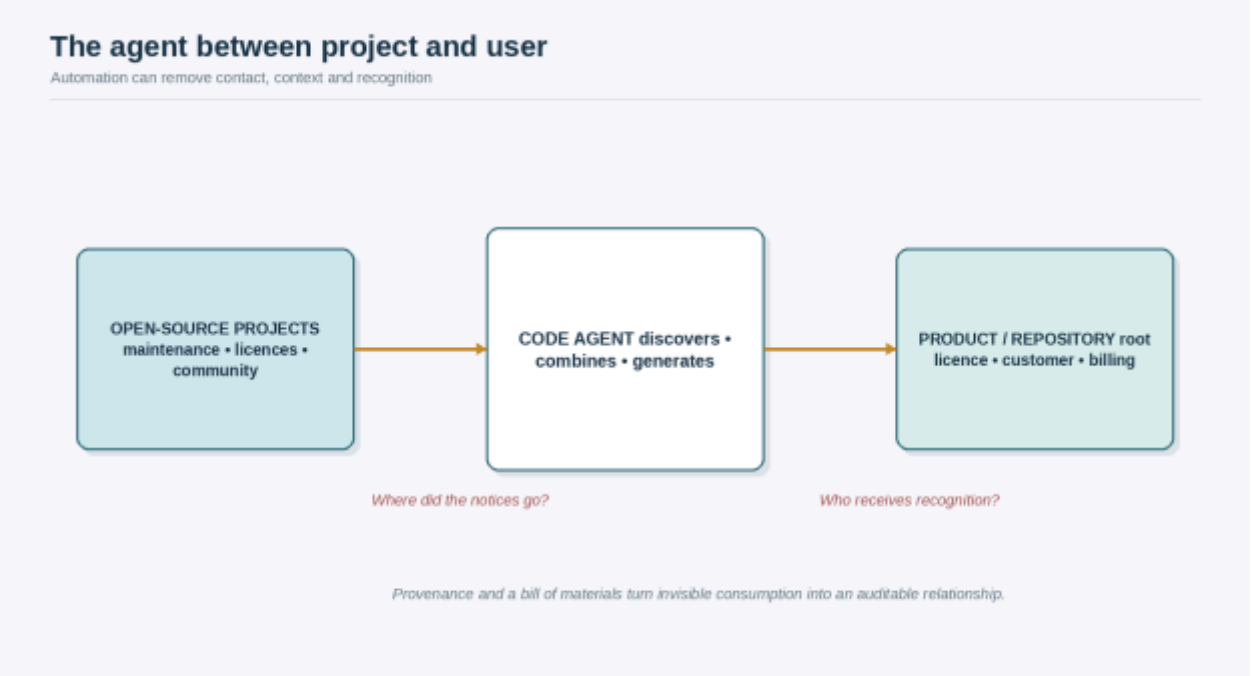
For a decentralised brand, provenance is the foundation of an economy that rewards without claiming omniscience. Value cannot be returned to contributors whom the institution has forgotten. Creators cannot be protected when transformations erase origin. The licence of the future is legal text plus verifiable memory.

CHAPTER 12. THE AGENT THAT CONSUMES THE COMMONS

12.1 AI as a multiplier of capture

Coding agents change the open-source economy by reducing the cost of discovering, combining and adapting components. A user no longer needs to read the documentation or interact with the project. An agent finds a package, copies a pattern, generates an integration and moves on. Productivity rises while the social relationship through which a project received bug reports, contribution and recognition can weaken.

Figure 15. Automatic consumption of the commons and loss of recognition



In the older path, users encountered projects directly, and friction sometimes produced questions, reports and visible contribution. In the agentic path, a platform selects and combines components inside a final product while the user may never learn which upstream projects were decisive.

A 2026 preprint models the possibility that "vibe coding" can increase productivity while weakening project entry and quality where maintainers monetise through direct contact with users [VIBE-CODING-OSS]. The result is not established for every ecosystem, but the mechanism is plausible. AI can further separate use from recognition.

Licence notices can be lost in generated projects. An agent may recombine fragments, select dependencies and produce a repository with a permissive root licence while omitting notices or conditions attached to included components. Where the model itself was trained on open-source code, relationships between

outputs and sources are even harder to determine. Manual compliance cannot scale to every generated artefact.

The companies with the strongest models and agent platforms can capture value created by the commons at enormous scale. They control the interface, identity, billing and usage data. Upstream projects become invisible material inside an automated act of creation. This repeats the cloud pattern at the level of software production itself.

Copyleft obligations can still apply to distributed dependencies and covered derivatives, but detection becomes more difficult. Attribution obligations remain, but the user may not know what the agent incorporated. Model providers cannot guarantee that every output is free from protected fragments. Responsibility is distributed among provider, developer, deployer and customer, and contracts often shift much of the risk downstream.

One response is to restrict training or agentic use. Such restrictions may be justified for particular works, data or communities, but a universal ban is difficult to implement and may favour large actors that can negotiate private licences. Another response is provenance infrastructure. An agent should generate a software and model bill of materials, preserve notices, identify dependencies and express uncertainty rather than presenting every build as clean.

Decentralised brands can provide a distribution counterweight. Instead of recommending only the most popular library, a brand registry can describe verified components, applicable licences, maintenance status, known risks and funding mechanisms. Agents can choose according to compatibility and sustainability as well as benchmark performance.

The shared-mark terms can require official agent providers to preserve provenance and contribute to the funds supporting components used in their official services. This is not a tax on every copy of code. It is a duty of a commercial node that voluntarily claims the brand's assurance. An independent provider can offer a similar agent under its own name without receiving official certification.

AI can also amplify internal capture. A centre that controls the model evaluating contribution can decide who receives money and status. The algorithm may appear neutral while embodying the data and objectives of its operator. A scoring model must therefore remain advisory. Multiple methods, explanations, human appeal and independent audit are required.

Agents can manufacture contribution volume. Automatically generated commits, documentation and issues can overwhelm projects and distort metrics. Value moves towards curation: selecting, testing, integrating and accepting responsibility. Reward systems should favour accepted and maintained contribution, not raw production.

As code becomes more abundant, competitive advantage moves towards selecting important problems, governing data, earning distribution, maintaining

legitimacy and operating trustworthy systems. This strengthens the decentralised-brand thesis. Unless a commons of reputation is deliberately designed, the agent platform will absorb value even when underlying code remains open.

Capture is not inevitable. The same models that reduce friction can administer provenance, audit and contribution accounting. The difference lies in the institution that governs them.

12.2 AI as an administrator of reciprocity

A federation with hundreds of nodes generates more decisions, contributions and relationships than a small team can administer manually. This is one reason centralisation wins: it simplifies coordination. AI can reduce coordination cost without requiring every decision and record to be concentrated in one platform.

Table 11. Legitimate and illegitimate roles for AI

Legitimate role	Constitutional red line
Classifies artefacts and contribution evidence.	Does not define a person's merit, identity or moral worth.
Detects licences, missing provenance and probable conflicts.	Does not convert legal uncertainty into permission.
Proposes distributions and simulates alternative formulas.	Does not execute contested payments without authorised human control.
Summarises deliberation and translates across languages.	Does not erase disagreement by producing one authoritative summary.
Signals risk, abuse and inconsistent certification.	Does not impose permanent exclusion without notice, reasons and appeal.

AI can reduce administrative labour without becoming the sovereign of the institution. Normative authority remains human and constitutional. The model assists evidence, explanation and simulation; it does not define the mission or issue unreviewable sanctions.

In provenance, an agent can generate and validate manifests, compare terms, trace dependencies and stop a build when essential information is absent. It can identify that an adapter inherits conditions from a base model or that a corpus is not licensed for commercial deployment. Its output should include evidence, assumptions and confidence, not only a verdict.

For contributor distributions, AI can aggregate signals that people cannot follow at scale: duration of maintenance, review work, incident prevention, component use, customer evidence and node feedback. It can propose weights and simulate how formulas affect different classes of contributor. The legitimate budget bodies retain the decision.

Shapley values provide one useful intuition from cooperative game theory: contribution can be estimated by a participant's marginal effect across possible coalitions. Exact computation does not scale, assumptions about the value function are contentious, and results can be counterintuitive in contextual work. Shapley-inspired methods may be useful for bounded events, not as a metaphysics of merit.

A practical system uses several ledgers. Current work is paid through contracts and approved budgets. Historical contribution may receive a limited, reviewable allocation. Essential components receive continuity funding.

Reputation is not automatically converted into money. AI detects imbalance and explains proposed distributions without claiming perfect causality.

The term "contributor dividend" can be a useful metaphor, but the legal instrument should usually speak of contributor distributions or grants. Recipients may not be shareholders, and payments can create tax, employment, withholding, social-security, unclaimed-property or securities questions depending on design. Transferable tokens or promises of passive appreciation add regulatory risk without solving the underlying problem. The simplest pilot uses ordinary accounting, clear eligibility and conventional payments.

In governance, an agent can summarise proposals, compare consequences and flag conflicts of interest. It can translate and reduce the advantage of participants who have time to read hundreds of pages. Deliberation should not be reduced to a model's recommended vote. A summary can omit the very disagreement that matters.

Governance models should be plural. Several nodes can run different evaluators and compare outcomes. Rules, prompts and model versions should be disclosed where possible, signed and change-controlled. No company should be able to alter silently the agent that allocates reputation or access.

Confidentiality requires local processing and data minimisation. An agent need not read every customer contract to verify an aggregated claim. Attestations, selective disclosure and independent audit can establish some facts without publishing their underlying content. Cryptographic proof can narrow trust; it cannot determine whether a rule is legitimate.

A ledger can preserve hashes, decisions and transfers, but immutability does not create justice. An immutable system can preserve an error indefinitely. The institution needs correction, context, proportional retention and, in appropriate cases, forgiveness. Provenance should be stable while interpretation remains contestable.

AI can assist customers as well. A procurement agent can compare official nodes, certification level, rights manifests, operational evidence and contribution to the commons. Reciprocity becomes a purchasing criterion rather than a hidden line in an annual report. Customers can choose suppliers that help finance the infrastructure on which they rely.

This is the decisive change. In the present economy, a company's behaviour towards upstream projects is difficult to observe. A decentralised brand can convert that behaviour into a verifiable signal. AI reduces the cost of checking the signal at scale.

The danger is a total reputation system. To resist it, records must remain contextual, contestable and subject to retention limits. Identities should be separable where legitimate. Not every mistake should follow a person forever. The objective is resource allocation and professional trust, not moral classification.

AI can make federation administrable only when it remains an instrument under a constitution. If one model controls access, reputation and payment, the centre has not disappeared. It has become less visible. Constitutional openness requires that the model, provider and scoring method can be replaced without losing the institution.

PART V: THE COMPACT OF THE SHARED NAME

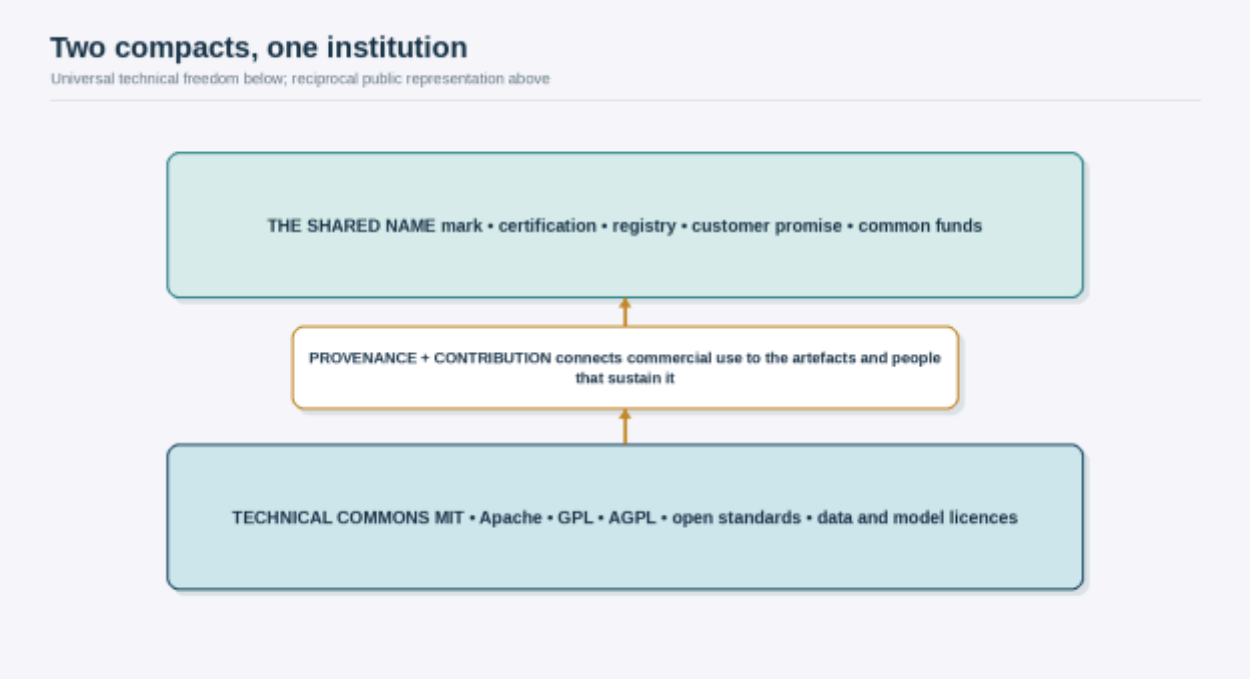
OPEN TECHNICAL COMMONS • ACCOUNTABLE INSTITUTIONS

CHAPTER 13. OPEN CODE, RECIPROCAL NAME

13.1 Moving reciprocity towards the brand

The central proposal can be stated in one sentence: code can remain free without making the shared name free to exploit, and the name can be protected without allowing a centre to confiscate the work. Existing software licences address the first half more successfully than the second. They determine who may use, modify and redistribute a program. They do not determine who may commercially benefit from the trust accumulated around it, how that trust should be maintained, or whether some of the resulting value should return to the people and institutions that made it credible.

Figure 16. The technical commons and the constitution of the shared name



The lower layer is a technical commons governed by standard open-source licences. Above it sit the shared name, certification, distribution channels, registries and customer-facing assurance. Technical rights remain available to everyone. Additional duties arise only for autonomous nodes that voluntarily use the official mark or claim institutional membership. Provenance connects the layers by showing which artefacts, obligations and contributions support the public offer.

Table 12. The boundary of reciprocity

Universal technical right	Duty of a node using the shared brand
Runs, studies and modifies the software.	Meets the published standard and discloses relevant provenance.

Universal technical right	Duty of a node using the shared brand
Redistributes the original or a derivative under its licence.	Contributes to maintenance and the common funds under an agreed formula.
Offers a service under its own name.	Accepts audit, remediation and rules for public representation.
Creates a fork and an independent identity.	Does not claim current membership or certification without authorisation.

The table locates reciprocity at the commercial and reputational boundary rather than inside the open-source licence. Anyone can use the code in accordance with its technical licence. Only the voluntary use of the official name, certification seal, common channels or assurance infrastructure activates the additional compact.

This boundary preserves the conceptual integrity of open source. A person or company may take the code, modify it, sell it, offer it as a service and create a competing fork under another name, subject to the applicable licence. The reciprocal contribution is not the price of software freedom. It is the condition for taking part commercially in an institutional asset whose value consists of a common promise to customers.

The distinction avoids a first error: inserting a fee, field-of-use restriction or discriminatory condition into the software licence and continuing to market the result as open source. The Open Source Definition requires free redistribution and prohibits discrimination against persons, groups or fields of endeavour [OSI-OSD]. A licence reserved to cooperatives, restricted to small companies or conditioned on a revenue payment may be defensible as fair source, source-available, coopyleft or a commercial licence. It should be named accurately. Misdescription would weaken the very trust signal the system is meant to create.

It also avoids a second error: allowing the mark to become a private toll booth for the founder. A company may publish a permissive core, retain exclusive control of the name and charge every commercial participant after the community has made the project famous. That is a familiar platform strategy, not a decentralised brand. The difference is not the existence of a payment. It lies in the mandate under which the mark is held, the procedure by which obligations are set, the destination of revenue, the limits on relicensing and the possibility of replacing the custodian.

In the proposed model, the mark is held by a custodian with a limited constitutional mandate. Depending on jurisdiction and purpose, the custodian may be a foundation, federated cooperative, purpose trust, association or hybrid entity. The legal form does not by itself produce legitimacy. The governing documents must prevent sale or encumbrance of the shared mark for private benefit, define succession and require the custodian to protect the promise, authorise qualified nodes, finance common goods and preserve continuity.

Trademark law also imposes a discipline that political language cannot remove. A proprietor that authorises others to use a mark normally needs legitimate control over the nature or quality of the goods and services associated with it. Collective and certification marks likewise depend on rules, standards and supervision [USPTO-RELATED; USPTO-COLLECTIVE; USPTO-CERTIFICATION]. Decentralisation cannot mean passive licensing of a symbol whose meaning no one is responsible for. The compact must make control objective, limited, auditable and contestable, rather than pretend that control is unnecessary.

An official node remains an autonomous organisation. It retains its company, people, capital, customer relationships, local processes and node-specific intellectual property. It is not automatically a subsidiary, agent or joint venture of the custodian, and the compact should avoid language that creates such a relationship unintentionally. The node accepts a bounded common layer: technical interoperability, accurate licence and provenance disclosure, contribution to maintenance, data-governance duties, audit, customer-remediation procedures and a defined participation in the cost of the brand.

The architecture resembles franchising only up to a point. A franchise also permits independent entrepreneurs to trade under a common name while receiving a system and accepting controls. Yet the franchisor normally owns the system and exercises a stronger unilateral authority. In the United States, a relationship involving a trademark, significant control or assistance and a required payment may fall within franchise regulation regardless of its label [FTC-FRANCHISE]. Other jurisdictions use different tests. A decentralised brand cannot avoid applicable law by calling a fee a contribution or a rule a constitution. Where the substance is franchising, the disclosure, registration and relationship rules must be respected.

This comparison supplies a useful design limit. The compact should define the minimum results that make the public promise credible, not administer every detail of the node's business. The more the centre dictates prices, customers, territories, suppliers, employment, marketing and internal operations, the less meaningful autonomy becomes. It also creates competition-law risk. A trade association or network may establish legitimate standards, but it must not become a venue for price fixing, customer allocation, output restriction, coordinated boycotts or exchange of competitively sensitive information [FTC-ASSOCIATIONS; FTC-PRICE; EU-HORIZONTAL].

An explicit anti-cartel clause should therefore be constitutional, not an appendix drafted after growth. The brand may define quality, interoperability, safety, provenance and truthful representation. It should not set resale prices, wages or commercial territories for independent nodes. Certification should be open to qualified non-members where the legal form allows it, based on published criteria and subject to independent appeal. The network coordinates the promise, not the market choices that should remain competitive.

Collective and certification marks are closer to this logic than a conventional franchise. A collective mark can identify membership in an association. A certification mark can indicate that goods or services meet a defined standard applied by a certifier that is institutionally distinct from ordinary sellers [USPTO-COLLECTIVE; USPTO-CERTIFICATION]. European Union law also recognises collective and certification marks, subject to their governing regulations [EUIPO-MARKS]. These instruments do not solve economic distribution, but they permit custody of the sign to be separated more clearly from the commercial sovereignty of one company.

The brand should use levels of representation rather than one binary permission. Descriptive reference, such as "compatible with" or "built using", should normally be broad and governed by nominative and fair-use principles. A compatibility badge can certify a technical profile. An assurance mark can certify provenance, security or governance. The designation "official node" can indicate full participation in the compact. Each level should have its own criteria, visual rules, term, audit burden and remedy. This avoids using the strongest control for every kind of speech.

Economic reciprocity can be proportional to commercial use of the shared brand rather than to the number of copies of a program. A small node may pay a modest fixed membership contribution. A larger provider may contribute a limited fraction of revenue earned from products explicitly sold under the common name, verified maintenance work, shared infrastructure or a transparent combination. The formula should not subsidise large actors through unpaid volunteer labour, but neither should it turn growth into an unpredictable penalty.

The customer receives a concrete benefit. The customer is not buying software alone, but a verifiable promise concerning compatibility, provenance, remediation, continuity and the supplier's contribution to the infrastructure on which the service depends. The mark becomes a form of institutional assurance. It does not guarantee perfection. It identifies the responsible operator, the standard applied, the limits of certification and the mechanisms activated when the promise fails.

Liability must remain explicit. Certification does not automatically make the custodian liable for every product, nor does node autonomy excuse the brand from deceptive assurance. Contracts should distinguish the responsibilities of the node, auditor, custodian, infrastructure operator and any common remediation fund. Marketing should never imply a guarantee broader than the legal and financial capacity behind it. A credible brand describes its allocation of risk instead of hiding behind the ambiguity of a shared logo.

Contributors receive something that a software licence cannot provide alone. Their work is not merely visible in version history; it can participate in the budget generated by the shared reputation. This does not mean that every commit earns a perpetual royalty. It means that the institution preserves provenance,

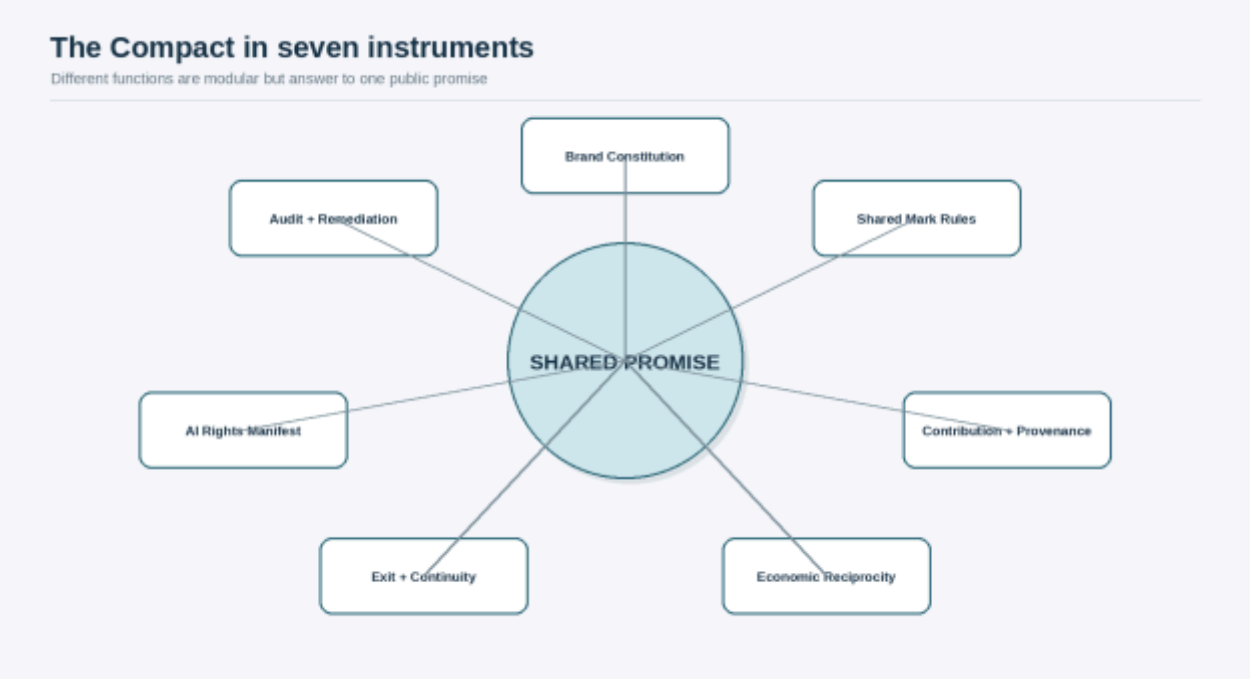
recognises different classes of contribution and reserves resources for the people and organisations that build and maintain the commons.

The freedom to use the code remains intact. The freedom to claim that one represents the institution becomes conditional. This is the middle path: open source is not converted into a private club, while the brand is not allowed to become a private pump extracting reputation from common work.

13.2 The seven instruments of the compact

A decentralised institution cannot be created through one licence. A document that attempts to govern code, models, data, trademarks, contribution, revenue, governance, certification and exit will become either vague or impossible to adopt. The Compact of the Shared Name must be modular. Its instruments can be analysed, amended and versioned separately, while each remains subordinate to a common constitutional purpose.

Figure 17. The modular instruments of the Compact of the Shared Name



Seven instruments divide distinct legal and institutional functions. The technical licences grant rights in code, data or models. The constitution defines purpose and limits. The mark rules govern public representation. The contribution instrument records provenance and grants. The economic scheme allocates common resources. The exit protocol protects continuity and autonomy. The AI manifest and audit protocol make claims verifiable and remediable.

Table 13. The seven instruments

Instrument	Function
Brand Constitution	Defines the public promise, common assets, governing bodies, limits, amendment and succession.
Shared Mark Licence and Rules	Defines levels of use, quality control, certification, suspension, transition and appeal.
Contribution and Provenance Covenant	Preserves origin, obtains necessary grants and limits unilateral relicensing.
Economic Reciprocity Scheme	Defines node contributions, common funds and contributor distributions.
Exit and Continuity Protocol	Protects autonomy, factual reputation, customer transition and institutional succession.

Instrument	Function
AI Rights and Provenance Manifest	Makes the rights stack and technical genealogy machine-readable and version-specific.
Audit, Certification and Remediation Protocol	Separates verification from commercial interest and supplies reasons, correction and appeal.

No instrument is sufficient on its own. The table deliberately separates licence from contract, trademark control from governance, and economic distribution from attribution. A participant should know before joining which rights are granted, which obligations are contractual, which claims are certified and what remains portable after departure.

The first instrument is the Brand Constitution. "Constitution" is an institutional metaphor, not a claim that a private network acquires public sovereignty. Legally, its effect will depend on the entity's statutes, membership agreement, trust deed, cooperative rules and applicable law. Its function is to state the public promise, identify common assets, allocate authority, define the minimum rights of nodes and contributors, constrain the custodian, describe amendment and establish succession.

The constitution should distinguish protected purpose from ordinary strategy. A mission can be stable without freezing every product decision. Changes to the shared promise, disposal of the mark, closure of the commons, removal of exit rights and unilateral relicensing require elevated procedures. Budgets, technical profiles and operational policies can be delegated. A constitution that constitutionalises everything cannot operate; one that protects nothing is ceremonial.

The second instrument is the Shared Mark Licence and Rules. It identifies the proprietor or custodian, the authorised users, the goods and services covered, the quality standards, the evidence required and the forms of use. It should distinguish descriptive reference, compatibility, certification and official membership. It needs a fixed term or review cycle, notice, cure periods where appropriate, emergency suspension, independent appeal and rules for post-termination transition.

Quality control cannot be fictional. The rules should identify who inspects, how often, with what evidence and under which conflict-of-interest safeguards. Standards should be published in a form that qualified organisations can understand. A node should not be denied because it competes with a favoured member. Certification decisions should include reasons, and serious disputes should be reviewable by a body not paid according to the result of the original inspection.

The third instrument is the Contribution and Provenance Covenant. For code, a project may continue to use a Developer Certificate of Origin, a contributor agreement or direct copyright assignments, depending on its needs [DCO; ASF-CLA]. None of these mechanisms proves every aspect of title. Employment

agreements, prior obligations, patents, moral rights, database rights, confidential information and third-party material can remain relevant. The covenant records the contributor's representations, the grant actually needed and the origin of the contribution without pretending to cure unknown defects.

The preferred default is the least transfer of rights compatible with continuity. A DCO-style certification can be sufficient where the project does not need broad relicensing power. A limited contributor agreement can grant patent permissions, defence rights and the ability to distribute under approved successor licences. A broad assignment may be justified for some projects, but it should not be presented as neutral commons governance if one company can later relicense the aggregate unilaterally.

The covenant should therefore restrict relicensing. A change from an open-source licence to source-available terms, or a change that removes established downstream freedoms, should require the procedure specified in the constitution and may require new consent where the original grants do not authorise it. Contributions whose owners cannot be reached remain under their existing licences. The institution cannot grant rights it does not possess.

Provenance extends beyond code. Data contributors must identify the source, collection basis, applicable permissions, privacy constraints and permitted operations. Model contributors identify the base model, weights, adapters, training or distillation process and inherited terms. Designers, writers and translators identify their own work and incorporated material. Provenance is not merely attribution; it is evidence about the rights and conditions that travel with an artefact.

The fourth instrument is the Economic Reciprocity Scheme. It defines the membership contribution or mark-related fee, the revenue base, treatment of in-kind work, common funds, reserves, contributor distributions, audit and change procedure. It should use ordinary legal language rather than attractive but misleading financial metaphors. "Contributor dividend" may describe the intuition, but a recipient who owns no shares normally receives a distribution, grant, fee or award. Tax, employment, withholding, social-security, charity, securities and unclaimed-property rules may differ by jurisdiction.

The scheme should preserve node-level entrepreneurship. Nodes retain the majority of their commercial revenue because they employ people, acquire customers, accept delivery risk and carry product liability. The common contribution finances the mark, standard, maintenance, infrastructure, audit, continuity and a bounded return to contributors. The compact does not turn every participant into a co-owner of every node.

The fifth instrument is the Exit and Continuity Protocol. It defines notice, transition, export of evidence, customer communication, survival of legitimate obligations, cessation of mark use and treatment of pending certifications. It protects the right to compete, criticise and continue open-source artefacts under a

new identity. It also protects the public against confusion by making clear when official status ends.

Exit must include institutional failure, not only voluntary withdrawal. If the custodian becomes insolvent, abandons the project, loses legal capacity, privatises the mark contrary to the constitution or becomes unable to protect it, the protocol activates succession. It identifies alternate custodians, escrow arrangements, transfer triggers, key recovery and a procedure for a successor mark. A decentralised brand whose name dies with one company has not decentralised its most important asset.

The sixth instrument is the AI Rights and Provenance Manifest. It is machine-readable, but its entries point to authoritative legal texts and evidence. It identifies artefact, version, hash, type, parent models, source datasets, transformations, applicable licences, access terms, service restrictions, patent grants, known opt-outs, privacy constraints, audit status and uncertainty. It answers operational questions concerning running, copying, fine-tuning, merging, quantising, distilling, synthetic-data generation, redistribution, hosting and API exposure.

The manifest does not manufacture title. A declaration that data are "licensed" is only as strong as the underlying permission. Nor should an automated compatibility result be treated as legal advice. Its value lies in preventing information from disappearing between dataset, model, adapter, package and service. It allows procurement and build agents to identify missing evidence, preserve notices and stop combinations that require review.

The seventh instrument is the Audit, Certification and Remediation Protocol. It specifies auditor accreditation, independence, sampling, evidence retention, confidentiality, reporting, complaints, corrective action, suspension and appeal. The entity selling a product should not be the sole judge of whether the product satisfies the common standard. The custodian can supervise the system without deciding every case.

Remediation matters more than theatrical purity. A node can make a mistake in a manifest, miss a licence notice or fail an operational control. Proportionate correction preserves trust better than automatic expulsion. Fraud, deliberate concealment and immediate customer danger justify stronger action. The protocol should distinguish negligence, disagreement, technical failure and deception.

The instruments need a precedence rule. A software licence governs the granted copyright and patent permissions for the covered code. A data agreement governs the covered dataset. The mark licence governs use of the sign. Membership and economic agreements govern the participating node. The constitution constrains the institution's internal authority. No private agreement should be described as changing an open-source licence for third parties who never joined the compact.

Every instrument must also be versioned. A node's obligations are those attached to a defined version and effective date, subject to lawful amendment procedures. Existing permissions in released open-source artefacts are not retrospectively revoked. New versions may use different terms, but the change must be explicit. Stability is a core part of the public promise.

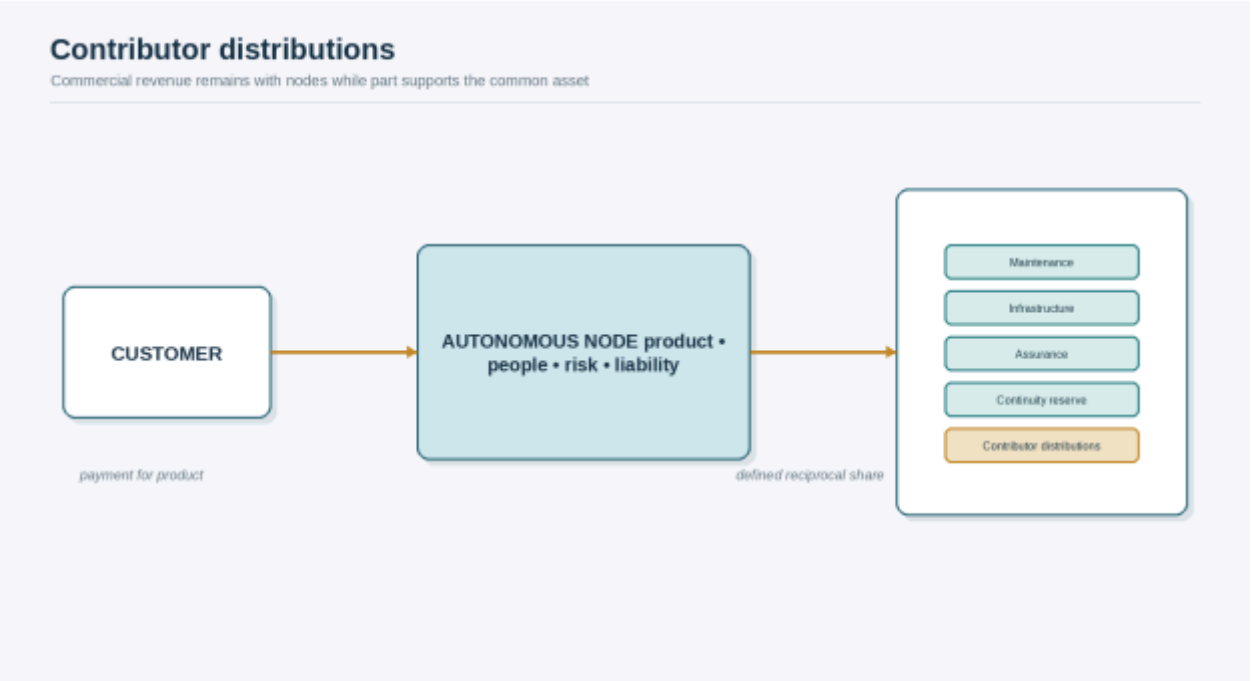
The compact should be drafted for replacement as well as adoption. A better standard, custodian or legal form may emerge. Modular instruments can migrate without rewriting the entire institution. The objective is not a perfect private legal system. It is a legible allocation of rights, power, money, evidence and exit.

CHAPTER 14. THE ECONOMY OF CONTRIBUTION

14.1 Contributor distributions without a mythology of merit

The most difficult claim of this book begins with a modest observation: technical contribution is not identical to economic value, but an economy that systematically forgets contribution becomes extractive. A decentralised brand should return resources to the people and organisations that build its common assets without pretending that total merit can be calculated mathematically.

Figure 18. The proposed economic flow between customer, node and commons



Customers pay autonomous nodes for products and services. Nodes retain most revenue because they carry employment, sales, delivery and liability. A defined share supports common maintenance, infrastructure, assurance, continuity reserves and contributor distributions. The flow does not make contributors shareholders in every node; it recognises that part of the commercial value was created by a shared institutional asset.

Table 14. Classes of contribution

Contribution	A plausible equitable treatment
Founding architecture and IP	Structural recognition bounded by vesting, decay, succession and the rights actually granted.
Current maintenance	Regular contracts or grants rather than dependence on an uncertain annual distribution.

Contribution	A plausible equitable treatment
Code, review and security	Provenance, technical impact, operational responsibility and sustained quality.
Documentation, design and translation	Evaluation outside code-volume metrics and portable evidence of contribution.
Data, models and evaluations	Explicit rights, quality, provenance, permitted uses and continuing stewardship.
Community and governance	Payment for work and recognition of trust without creating a permanent caste.
Capital and infrastructure	Contractual return without automatic sovereignty over the shared mark.

The table rejects the idea that commits form a universal unit of value. Different contributions have different duration, risk, scarcity and substitutability. A fair system uses separate treatment where the underlying relationship is different.

The first rule is to separate payment for current work from distributions recognising accumulated contribution. A maintainer who works every week should receive a contract, salary or dependable grant. The person should not be told to wait for a speculative annual pool while managers and sales teams receive stable compensation. A contributor distribution can recognise historical or structural value; it is not a substitute for labour remuneration.

The second rule is bounded duration. Founding work may deserve longer recognition because it created the institutional option, supplied early risk and made later contributions possible. Yet an unlimited perpetual royalty can convert the past into an aristocracy that taxes every future improvement. A weight can vest, decline over time, retain a modest floor for genuinely structural work and rise again through renewed maintenance. The formula must be visible before contribution, not invented after success.

The third rule is plural evidence. Usage is informative but incomplete. A tiny library may sit in thousands of products; a rarely executed security validator may prevent catastrophic loss. Documentation reduces adoption cost. Release engineering preserves continuity. Moderation can prevent a community from becoming unusable. Evaluation should combine technical dependency, operational risk, maintenance effort, review quality, customer impact, peer assessment and evidence of actual use.

The fourth rule recognises negative contribution. Code can increase attack surface, automatic content can consume scarce review attention, and marketing promises can impose costs on others. Historical acceptance should not guarantee permanent economic weight when an artefact was removed for serious reasons. Any reduction must be reasoned and documented so that correction does not become political erasure.

Algorithms can help. Dependency analysis identifies components on which official products rely. Provenance links delivered systems to artefacts. Issue and

review history can reveal maintenance work that commit counts miss. Bounded Shapley-inspired calculations may estimate marginal contribution in clearly defined cooperative tasks. None should execute the financial result automatically. The value function is itself a political and economic choice.

Value is institutional. A designer can create the visual language that makes the brand recognisable. A lawyer can construct the rights structure that makes collaboration possible. A community organiser can prevent a split. A researcher can supply an evaluation that protects the public from inflated claims. These effects cannot be compared to a patch through one numerical scale. A contributor chamber and periodic qualitative review must complement machine metrics.

Metrics create behaviour. Once money or status depends on a signal, participants learn to optimise it. Commit splitting, synthetic issues, reciprocal reviews, coordinated endorsements and agent-generated activity can manufacture volume. The system should reward accepted, maintained and consequential contribution, apply materiality thresholds, inspect anomalies and keep the formula simple enough to audit. Complexity may disguise discretion rather than eliminate it.

Contribution units can be useful as internal accounting, but freely tradable governance tokens are a poor default. When economic claims and votes become speculative assets, capital can purchase authority that others built through work. A pilot should use non-transferable records and conventional payments. Limited inheritance or assignment may be appropriate, but the objective is economic memory, not a secondary market in institutional sovereignty.

Organisational contribution requires a double record. A company may pay ten engineers and carry the contractual risk. It deserves recognition for resources and coordination. The people still deserve portable evidence of their work and professional reputation, subject to employment and confidentiality obligations. The compact should not assume that the individual owns the copyright, nor allow the employer to erase individual authorship where acknowledgement is lawful.

Data and models need their own classes. A curated, rights-cleared corpus can be more valuable than millions of unverified samples. A maintained domain adapter can support several products. Evaluations, red-team suites and taxonomies may be central to assurance. The people whose data or creative works are included are not automatically contributors merely because the dataset records them; copyright, privacy, publicity, biometric, consent and cultural-governance questions must be addressed through the relevant legal and ethical instruments.

Benefits can take several forms: money, research grants, paid maintenance, access to compute, training, shared legal services, travel support or infrastructure credits. Early projects may have little cash, and a reliable build service may be more useful than thousands of negligible payments. The form and valuation of benefits should be disclosed and approved through the economic scheme.

A materiality threshold is necessary. A system that calculates millions of trivial transfers may spend more than it returns. Small contributions can accumulate evidence and participation rights until a meaningful threshold is reached. Unclaimed amounts should return to maintenance or a published common purpose, not silently to management.

The recipient needs an appeal. The person should be able to inspect the evidence attributed to them, the formula, weights and reason for the result. Identity, omission, classification and conflicts should be contestable. An independent body decides material cases. AI may assemble the record and simulate alternatives, but should not issue the final unappealable judgement.

Privacy also matters. Public contribution histories can expose employment, health, political or interpersonal information. A participant may need a pseudonymous record, a private economic identity or the right to keep some evidence with an independent auditor. Portability does not require universal publication. The institution should prove what is needed for distribution without building a permanent surveillance profile.

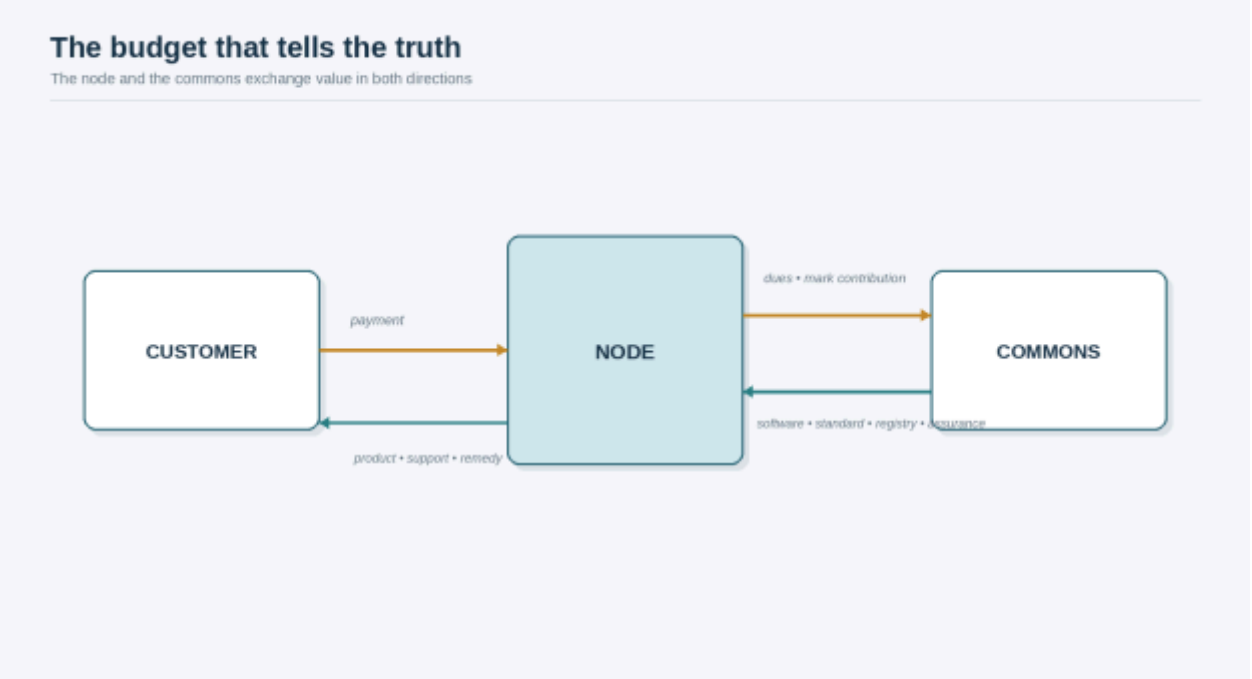
The phrase "contributor dividend" captures an intuition: some of the value accumulated under the shared name should return to those who made it possible. The operative documents should generally use more precise terms such as contributor distribution, maintenance grant, licence payment or service fee. The legal treatment can affect tax, employment, social security, charity rules, securities regulation, withholding and reporting. A poetic name should not create an accidental financial product.

A well-designed distribution does not promise equal results. Founders, maintainers, nodes, investors and occasional contributors will earn different amounts and through different instruments. It promises something narrower and more credible: a visible share of the value accumulated by the common name returns to the infrastructure and contribution that created it, according to rules that can be inspected, challenged and changed.

14.2 The budget that tells the truth

A community that refuses to discuss money is governed by the money it does not discuss. Open source has survived through volunteer time, corporate salaries, universities, foundations, consulting, sponsorship and personal commitment. That mixture produced extraordinary infrastructure. It also produced critical projects maintained without adequate resources, dependence on a few sponsors and decision-making advantages for organisations able to assign full-time staff.

Figure 19. Economic reciprocity between an autonomous node and the commons



Value moves in both directions. A node receives software, standards, reputation, certification, registry services, shared security and distribution. It returns a defined contribution to the common funds. Treating the flow as reciprocal infrastructure rather than donation makes the economic relationship visible and auditable.

Table 15. Rules for a credible common budget

Rule	Reason
Diversified income	No sponsor can convert one payment stream into an undisclosed veto.
Separated funds	Operations, maintenance, contributor distributions and continuity reserves cannot be casually merged.
A precise contribution base	Nodes can plan and audit does not become permanent negotiation.
Capped in-kind valuation	Sponsors cannot manufacture influence through inflated internal rates.
Disclosed related-party transactions	Members can see who benefits from institutional decisions.

Rule	Reason
Prospective, plural approval of changes	The economic formula cannot become a managerial weapon.

Transparency is not a dashboard by itself. It becomes an anti-capture mechanism only when revenue is diversified, funds are separated, conflicts are visible and no single payer can threaten the institution into abandoning its promise.

Membership contributions are the simplest revenue source. They make participation explicit and provide predictable operating funds. They should be scaled so that small organisations can enter. Tiers may use revenue earned under the mark, number of people engaged or certification level. Sensitive financial information can be verified by an independent auditor and reported in bands rather than disclosed publicly.

A mark-related fee can reflect commercial use of the common reputation. The base must be precise: revenue invoiced for products or services presented under the shared name, not the global revenue of the company. Mixed products require defined categories and consistent allocation. Ambiguity would turn each audit into a bargaining contest and invite selective enforcement.

Any revenue-linked formula should be reviewed for franchise, competition, tax and accounting consequences in each jurisdiction. The network must not use the formula to coordinate the prices nodes charge customers. Nodes report the agreed base; they remain free to set their own prices, costs, commercial strategy and customer relationships. The compact funds common infrastructure without acting as a cartel.

Certification and assurance can generate another income stream. Regulated customers may pay for evaluations, traceability, compatibility, incident handling and contractual commitments. Payment must be separated from the certification decision. Auditors can be accredited and compete on service while applying a common standard. The institution finances inspection; it does not sell approval.

Common infrastructure can also earn revenue through registries, distribution, training, conferences, shared procurement and marketplaces. The constitution should decide when one service is a legitimate common utility and when it should be pluralised. A single registry may improve coherence, but control of the registry, mark, treasury and certification should not be concentrated merely because one technical operator built the first version.

Grants and donations remain valuable for research and public goods. Foundations and security initiatives demonstrate that corporate resources can support common infrastructure without owning every project [LF-FUNDING-2024; OPENSSF-GRANTS]. The issue is not the moral purity of the money. It is the consideration received and the power created. A sponsor may receive visibility,

services or a defined membership role. It should not purchase an undisclosed veto over the mission.

A maintenance fund should have priority. A minimum portion of common income supports critical components before prestige marketing or expansion. The list of critical assets is public and periodically reviewed. Multi-year maintenance contracts are often more useful than prizes because continuity cannot be planned through occasional attention.

The contributor fund receives a separate allocation governed by the rules of the previous section. It should not be redirected to a campaign or executive deficit through an ordinary management decision. In a severe downturn, all categories may be adjusted through a declared emergency process, but the centre should not protect itself by cutting only the distributed edges.

A continuity reserve finances security incidents, litigation, infrastructure migration, customer remediation and transfer to a successor custodian. A brand without a reserve can be captured by rescue. The organisation that pays the next invoice becomes the real governor. Institutional autonomy has a cost and must appear in the budget.

In-kind contributions deserve recognition but require limits. A company can host infrastructure, pay maintainers, provide legal work or supply audit. The institution can value the contribution under agreed rates and caps. Otherwise a sponsor can assign inflated internal prices, claim to have funded the commons and acquire influence without providing usable liquidity.

Related-party transactions require special visibility. A board member's company may operate the registry, an auditor may sell remediation, or a contributor may receive a maintenance contract. Such relationships are not automatically improper. They should be disclosed, competitively justified, approved without the interested party and periodically reviewed. The aim is to see power, not to pretend that expertise has no interests.

Transparency does not require publication of every salary or customer contract. It requires intelligible revenue categories, fund balances, executive and board remuneration bands, major contracts, procurement criteria, related parties and the effective beneficiaries of decisions. Personal data can be protected without concealing institutional dependence.

The allocation formula should not be permanent. A young ecosystem may invest more in adoption and tooling. A mature one may shift towards maintenance, assurance and distributions. Changes should be announced prospectively, simulated against representative nodes and approved by more than one chamber. Participants need enough notice to decide whether to remain under the new terms.

Investors have a legitimate place. They can finance nodes, infrastructure operators and commercial ventures and receive contractual returns. Funding one phase should not automatically give them ownership of the shared mark. Equity,

debt, revenue share or project rights can attach to the financed entity, while the constitutional asset remains in limited custody. The compact is not anti-capital; it separates capital at risk in an enterprise from sovereignty over a public promise created by many actors.

The brand can publish an annual reciprocity statement. It shows the value received from node contributions, services and grants; the value returned through maintenance, infrastructure, assurance, contributor support and reserves; sponsor concentration; related-party transactions; and deviations from policy. The statement should explain, not merely display, the numbers.

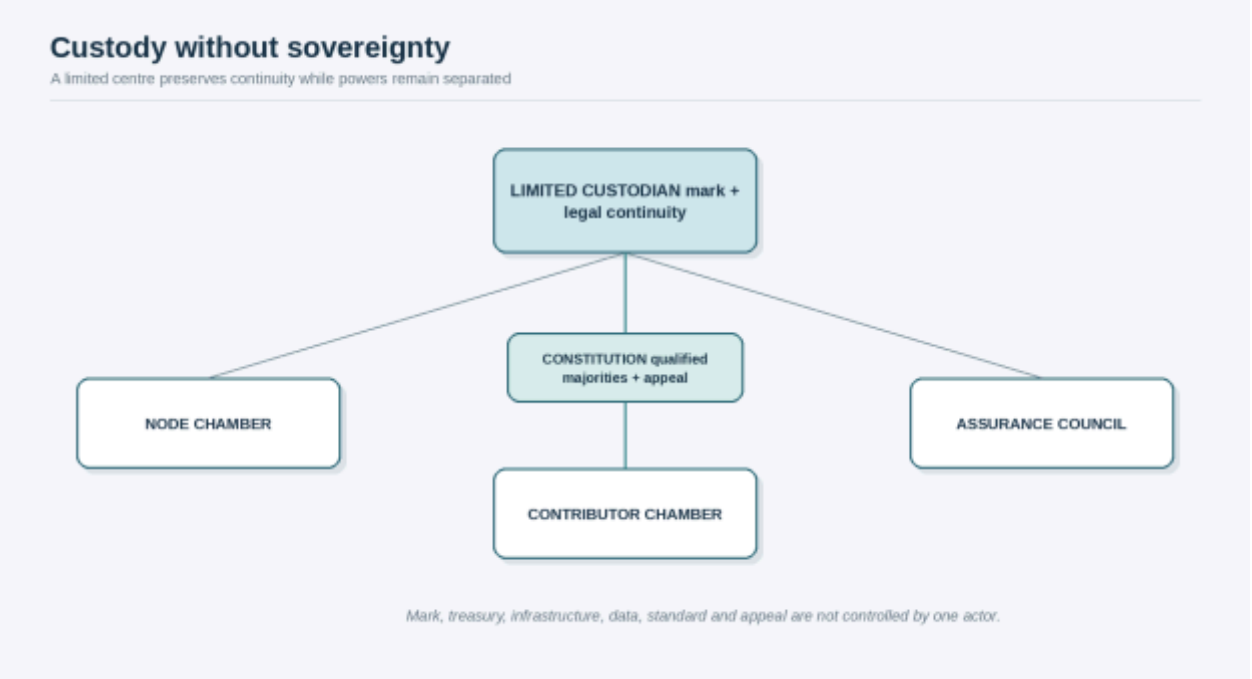
A budget that tells the truth does not guarantee that every participant considers the distribution perfect. It makes visible that freedom, audit, continuity, legal defence and dignified exit require resources. Equity does not mean that the institution is free. It means that participants can understand, challenge and revise who pays, who benefits and who controls the decision.

CHAPTER 15. CUSTODY WITHOUT SOVEREIGNTY

15.1 Governance, succession and emergency power

A decentralised brand is not a brand without a centre. A shared promise requires interpretation, legal defence, standards, response to fraud and a public voice. The absence of a formal centre does not abolish power. It transfers power to senior maintainers, funders, infrastructure administrators or people with enough time and expertise to govern. The serious question is not whether a centre exists, but how its authority is partitioned and replaced.

Figure 20. Multi-chamber governance with separated powers



A limited custodian holds the mark and legal continuity. A node chamber represents organisations serving customers and assuming operational risk. A contributor chamber represents creators and maintainers of the commons. An assurance council represents standards, audit and affected users. Major constitutional changes require more than one body; operational decisions remain delegated.

Table 16. Separation of institutional powers

Power	Institutional control
Shared mark	Mandated custodian, revocation procedure, legitimate quality control and succession.
Treasury	Separated funds, multi-signature authority, conflicts rules and independent audit.

Power	Institutional control
Infrastructure	Replaceable operators, documented backups, export and distributed recovery credentials.
Data	Local or purpose-limited custody, lawful basis, consent where required, portability and deletion.
Standard	Plural chambers, published evidence and qualified approval for fundamental changes.
Appeal	A body institutionally independent from the original certifier and commercial operator.

Formal voting is insufficient when one actor can withdraw the mark, stop the registry and freeze the treasury. Effective decentralisation requires that the practical levers of continuity are separated and recoverable.

The custodian may need a small board able to respond quickly. Its members should reach office through different routes, have staggered terms and face incompatibility rules. Some can be selected by nodes, some by contributors and some through an independent assurance or public-interest process. No single economic class should be able to occupy every seat through one coalition.

Founders present a legitimate difficulty. They supplied the concept, early risk, accumulated knowledge and often the initial mark. Removing their influence immediately can destroy coherence and invite opportunistic capture. Perpetual founder power, however, turns the institution into a monarchy with consultations. Time-limited protective rights can be justified: a temporary veto against sale of the mark, appointment of initial custodians or protection of the mission during a defined formation period. Every special right needs a term, scope, trigger and succession path.

Succession must be designed before crisis. What happens if a founder dies, the custodian becomes insolvent, a key company is acquired or leadership abandons the project? The mark can be held in a legally separate entity with a transfer mechanism tied to the constitution. Domains, registries, signing keys, repositories and financial accounts should not depend on one natural person. Recovery is part of the brand architecture, not an information-technology detail.

The compact needs a mark continuity plan. It identifies primary and alternate custodians, the evidence that activates transfer, who can petition for succession, how the proprietor's obligations are preserved and how the public is notified. Where automatic transfer is not legally available, escrowed instruments, conditional assignments or contractual undertakings can reduce the risk. Local legal advice is indispensable; the constitutional objective is stable even when the instrument changes.

Fundamental amendments require plural approval. Changes to mission, disposal of the mark, closure or relicensing of the commons, alteration of contribution rights or removal of exit protections can require qualified majorities in two or more chambers. Ordinary technical and operational decisions remain delegated. Requiring a referendum for every patch paralyses the system; allowing one board to rewrite the basic compact limits nothing.

The node chamber should not be weighted purely by revenue. Large nodes carry more risk and may contribute more resources, but one currency-weighted vote recreates a shareholder company. A system can combine one vote per qualifying node, bounded weights for contribution and concentration caps. The contributor chamber can use sustained activity, role diversity and peer recognition without turning commits into votes. The assurance body requires expertise and independence, not popularity.

Competition safeguards belong inside governance. Nodes are competitors as well as collaborators. Agendas should exclude discussion of future prices, wages, customer allocation, output plans and confidential commercial strategy unless a lawful and necessary context has been reviewed. Minutes record the legitimate subject of coordination. Counsel or a trained compliance officer can supervise high-risk meetings. The network's common standard must not become a cover for collective exclusion.

Certification criteria should be related to the public promise, proportionate and consistently applied. A standard can require provenance, security controls, interoperability, incident response and truthful marketing. It should not exclude a supplier merely because members dislike its business model or fear competition. Independent certification, open eligibility and appeal reduce both antitrust risk and internal capture [DOJ-CERTIFICATION].

Conflicts of interest are inevitable. A board member may represent a node bidding to operate the registry. A maintainer may evaluate their own component for funding. An auditor may sell remediation. Recusal, public registers, competitive procurement and independent review are more credible than declarations of moral purity. Institutions should make interests visible and limit their effect.

Emergency authority is necessary for severe vulnerabilities, fraudulent use of the mark or immediate danger to customers. It can permit temporary suspension, isolation of an infrastructure key, an urgent public notice or restricted expenditure from the reserve. Without such authority, the institution can fail while its chambers deliberate. Emergency is also the oldest route by which a centre converts temporary necessity into permanent sovereignty.

The difference lies in scope, expiry, reasons and review. Every extraordinary measure should identify the legal and constitutional basis, be limited to what is necessary, expire automatically unless confirmed, and be reviewed by a distinct body. Security-sensitive details may be withheld temporarily, but the existence and duration of the measure should be recorded. Repeated use of emergency power for the same class of problem triggers institutional review.

Customer interests need representation even when customers are not voting members. A complaint channel, ombudsperson or user council can expose situations in which standards protect suppliers more effectively than users. The brand exists in public. Its governance cannot be reduced to a compact among producers who certify one another.

AI can assist governance but cannot hold the constitutional key. Agents can compare proposals, model budgets, detect conflicts, classify evidence and translate deliberation. Relevant models, prompts, datasets and policy versions should be change-controlled and auditable. An automated recommendation gains no authority merely because it is consistent or numerically precise.

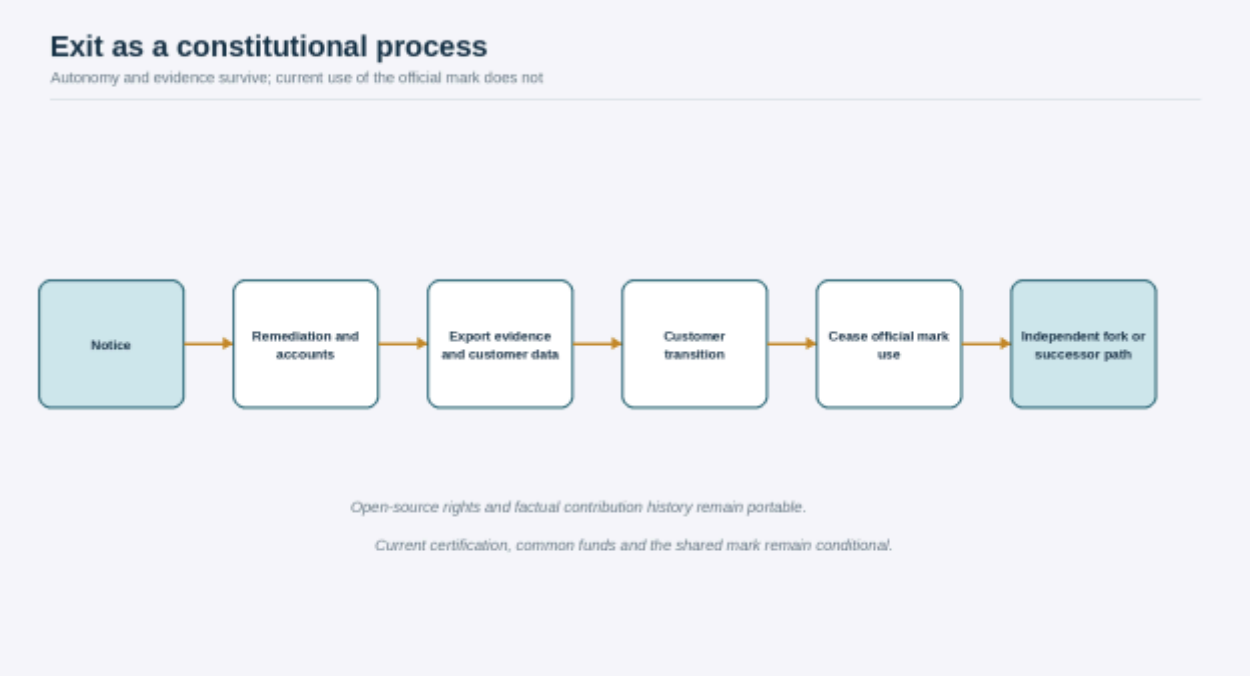
The organisation should preserve multiple routes for institutional memory. Minutes, decisions, manifests, budgets and keys have different confidentiality and retention requirements. Critical public records should be mirrored. Sensitive evidence can remain with independent custodians. A single platform account is not a constitution.

Good governance does not eliminate politics. It creates visible places where political conflict can be reasoned, appealed and corrected before it becomes secession. A legitimate centre is powerful enough to defend the promise and limited enough for the ecosystem to survive its failure.

15.2 The fork that preserves dignity

Open source treats the right to fork as the final guarantee against abuse. Legally, that right is real: the licence permits copying and continuation of the covered code. Socially and economically, a fork can lose the name, distribution, community, institutional history and customers. An exit that leaves a participant technically free but institutionally ruined is an incomplete protection.

Figure 21. Exit as a constitutional transition rather than an erasure



The node gives notice, completes defined remediation, exports its own evidence, communicates the transition to customers and stops current use of the official mark. Open-source rights and factual contribution history remain. Where the custodian itself has violated the constitution, a separate continuity procedure can authorise a successor institution or genealogically linked successor mark.

Table 17. What is portable and what remains shared or conditional

Portable for a node or contributor	Shared or conditional
Its own IP, capital, staff and customer contracts.	The official mark and representation of current membership.
Rights already granted by open-source and other licences.	Common funds and data belonging to other parties.
Verifiable contribution and service history.	Current certification after its stated expiry or withdrawal date.
Evidence of past performance and resolved incidents.	Common channels and infrastructure without a new agreement.
The right to criticise, compete and create a new identity.	The right to cause confusion about institutional continuity.

The compact protects evidence and autonomy without multiplying the same mark. A departing participant keeps what it owns and the rights already granted to it. It does not keep the right to imply continuing authorisation or to appropriate shared assets.

The protocol should provide an ordinary transition period. Customers should not be abandoned and certificates should not vanish without explanation. A node may use a factual statement such as "official node until [date], now in transition", under defined visual and temporal limits. In cases of fraud or immediate danger, the mark can be suspended at once, but the measure remains reasoned and reviewable.

Portable reputation is one of the most important innovations. The registry should allow export of contribution records, historical certifications, resolved incidents, training and verifiable service evidence. The departing party does not take the future prestige of the brand. The brand cannot erase the past. A person can prove that they maintained a component or operated at a certified level during a defined period.

Portability must remain contextual. A five-year-old certification is not a current guarantee, and an internal endorsement is not a universal score. Exported records include issuer, standard version, date, scope, evidence and later status. Privacy and confidentiality may require selective disclosure. Portability means that facts can travel, not that one institution dictates a global reputation.

Customer data follow contract and law, not the political preference of the brand. Where the node is an independent controller or operator, customers decide migration under the applicable arrangement. Where common infrastructure hosts data, the protocol specifies export, access continuity, deletion, backups and responsibility. A decentralised brand that holds data hostage reproduces the centralised platform.

Code forking is technically familiar. Institutional forking is more difficult. A group may believe that the custodian has abandoned the constitutional purpose and seek to continue the promise. The original mark cannot simply be duplicated without confusing the public or destroying quality control. The compact can instead define contested succession: a high-threshold process through which an independent adjudicator or qualified multi-chamber majority recognises a successor body or successor mark with an accurate statement of genealogy.

The threshold should be demanding. If every losing minority can claim the same continuity, the mark ceases to communicate. The process is justified when the custodian becomes unable or unwilling to perform its mandate, disposes of common assets contrary to the constitution, suppresses lawful succession or fundamentally abandons the public promise. Ordinary strategic disagreement should lead to an independent fork under a new name.

This mechanism reduces the custodian's ability to hold the ecosystem hostage. It does not grant a general right to seize a trademark. The enforceability of transfer, successor use and genealogy depends on applicable trademark,

association, insolvency and contract law. The legal instruments should be prepared in advance rather than improvised after rupture.

Linux supplies an indirect lesson. The Linux mark is administered separately from the GPL terms governing the kernel; software rights and trademark representation are distinct [LINUX-MARK]. A distribution can exercise the licence while respecting rules that prevent misleading use of the name. The compact extends this separation by adding plural custody, economic reciprocity and a continuity procedure.

OpenTofu, OpenSearch and Valkey show that forks can become institutional responses to relicensing or control changes, but viability depends on foundations, infrastructure, sponsors and a new public identity [OPENTOFU; OPENSEARCH; VALKEY]. The technical licence makes continuation lawful. Complementary assets make it credible. Exit design should prepare those assets before conflict.

The system can permit partial exit. A node can relinquish "official" status while retaining a narrower compatibility certification. A contributor can stop governance participation while preserving vested economic rights under the scheme. A local project can implement the open standard without entering the common economy and use only descriptive language. Departure should not be an excommunication ritual.

Surviving obligations must be limited and legitimate. Confidentiality, outstanding payments, preservation of customer records and accurate post-termination trademark use may continue. A broad non-compete, non-disparagement clause or prohibition on using acquired professional knowledge would nullify autonomy and can be unlawful. The brand protects its promise and confidential assets, not the future of every participant.

Re-entry should remain possible. A node that departed for economic reasons can return after meeting the current standard. A sanctioned participant should have a path to remediation where the misconduct and public risk permit it. Institutions that turn every conflict into a permanent identity resemble sects rather than commons. Memory needs context and the possibility of change.

Exit does not replace voice. An institution that responds to criticism with "you are free to leave" uses freedom as an exclusion mechanism. The chambers, appeal procedures and transparency obligations should make correction from within realistic. Exit is the last protection against domination, not the normal answer to disagreement.

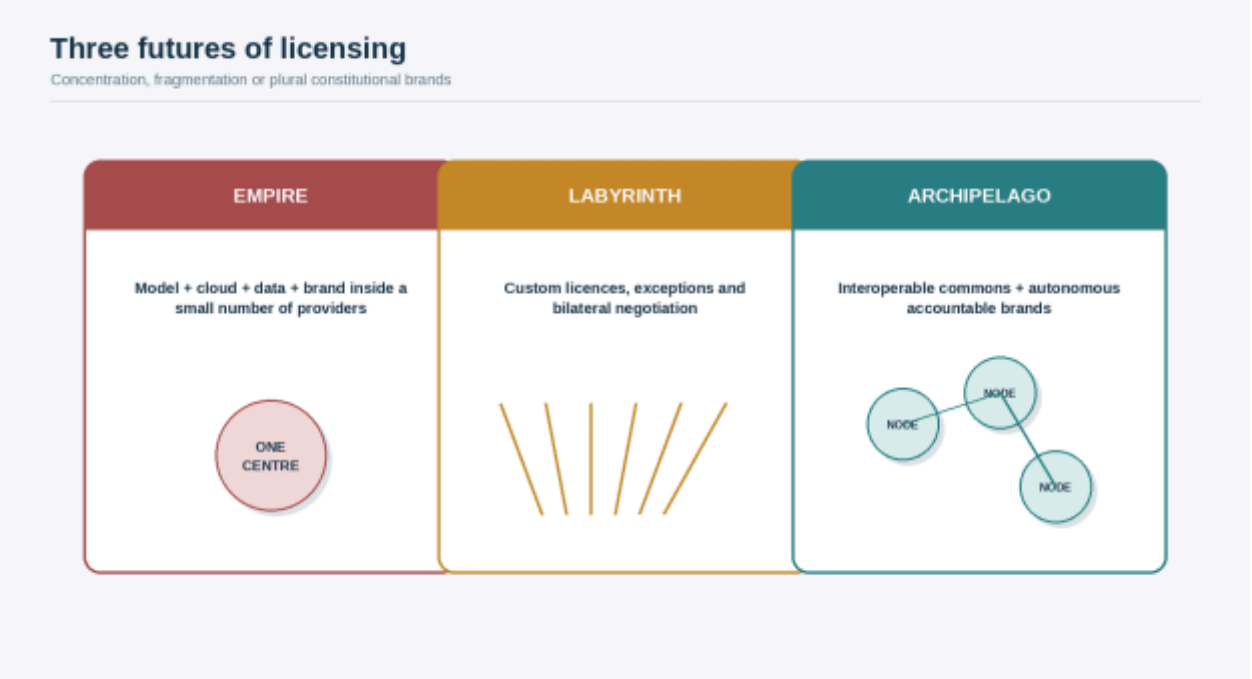
A dignified exit disciplines the centre. Leadership knows that nodes can continue, that factual history is portable, that common infrastructure has recovery paths and that custody can be challenged under defined conditions. Authority remains, but loyalty must be earned through performance and legitimacy rather than manufactured through the artificial cost of leaving.

CHAPTER 16. THREE FUTURES OF LICENSING

16.1 Empire, labyrinth and archipelago

The future of licensing will not be decided by lawyers alone. It will emerge from the economics of AI models, concentration of compute, demand for provenance, regulatory pressure and the ability of contributors to organise credible names. Three scenarios summarise plausible directions. They are not mutually exclusive predictions. Different markets may inhabit all three at once.

Figure 22. Three institutional scenarios for the next era of licensing



The Empire concentrates model, data, cloud and brand in a small number of providers. The Labyrinth multiplies custom licences and exceptions until only large organisations can navigate them. The Archipelago keeps interoperable commons while plural institutions provide assurance, identity and economic reciprocity. Each contains centres; what differs is how many assets any one centre controls.

Table 18. Three futures

Scenario	Institutional profile
Empire	Integration, capital, assurance and scale combined with lock-in and concentrated data, compute, distribution and brand.
Labyrinth	Custom protection and bilateral negotiation combined with high compliance cost and advantage for powerful actors.
Archipelago	Interoperable commons and autonomous brands combined with harder coordination but bounded failures and meaningful exit.

The scenarios expose different trade-offs rather than moral caricatures. Empire can industrialise capability. Labyrinth can protect investment and values. Archipelago can distribute continuity. Each can fail through a different concentration of power.

In the Empire, frontier models and agent platforms become the principal infrastructure of software and knowledge work. Open-source code persists, but it is increasingly consumed through agents, services and marketplaces controlled by a few companies. Permissive licences remain attractive for components that increase platform adoption. The most valuable models, data, customer interfaces and operational evidence remain contractual.

Large providers offer security teams, service levels, legal support, indemnity and compliance documentation that small communities cannot easily replicate. Customers choose convenience and a clearly accountable counterparty. Contributors receive salaries when absorbed into the platform and reputation when they remain upstream. The commons becomes an exploratory reservoir from which the Empire selects what deserves industrialisation.

This scenario can produce excellent systems and enormous investment. Its weakness is political and economic fragility. A change in price, policy, access or model behaviour can affect thousands of dependent organisations. The dominant provider can define evaluation criteria, preferred formats and the language through which quality is perceived. Formal freedom in a dependency does not compensate for dependence on identity, compute and distribution.

In the Labyrinth, creators respond to capture through specialised terms. Some prohibit competing cloud services, AI training, particular industries, large-company use or commercial deployment. Delayed-open instruments such as BSL and FSL protect a commercial period before conversion to an open-source licence. Responsible-use licences add behavioural restrictions [BSL; FSL; OPENRAIL]. Model families add thresholds, output restrictions and derivative conditions.

These tools can be rational. A startup may need time to recover investment. A data community may refuse uses it regards as abusive. A model provider may need to constrain redistribution where it lacks rights to transfer upstream material. The problem is proliferation. Every agent, product and procurement team must interpret a new combination. Vague terms invite defensive rejection. Large actors purchase exceptions; small actors avoid the artefact.

The Labyrinth can therefore undermine equity. A restriction drafted to protect the creator may burden the small reuser most, while a corporation negotiates a commercial licence. Custom language fragments the commons, increases transaction cost and makes automated compatibility difficult. Without standard families and provenance, legal complexity becomes another economy of scale.

In the Archipelago, foundational code, protocols and standards remain open, while the institutions that organise trust above them become plural. Decentralised brands unite autonomous nodes under verifiable promises. They negotiate data rights, fund maintenance, certify services, preserve provenance and offer common assurance. Interoperability is sufficient for migration but not so total that every institution loses its distinct purpose.

An archipelago is not a world without power. Some brands become large and some custodians influential. The difference is that every strategic asset need not sit inside one company. A foundation can hold the mark, multiple nodes can operate infrastructure, data can remain with local custodians, and software can continue under standard licences. The failure of one organisation does not destroy every route to continuity.

Brands compete through constitutions as well as products. Customers can choose an ecosystem optimised for low cost, privacy, open science, local control or regulated assurance. Contributors can choose which economy and governance deserve their effort. Machine-readable provenance makes those institutional differences comparable.

The Archipelago has higher coordination costs. Decisions can be slower, conflicts more visible and economic formulas imperfect. Sponsors can capture a chamber. Founders can manipulate succession. Networks can fragment. Its promise is not immunity from politics. It is a smaller radius of failure and the existence of credible alternatives.

The real future will probably mix the scenarios. The largest frontier models may remain imperial because compute, data and risk are concentrated. Niche products may rely on delayed-open terms. Public infrastructure, research communities and local AI may form archipelagos. Sensible policy should not impose one form. It should preserve competition among forms and prevent one from closing every route of exit.

Several licensing trends follow from this mixture. A single LICENSE file will cease to be an adequate description of an AI system. Open weights will remain the most common market-facing form of openness because they permit local use without surrendering the full training advantage. Fully open systems will remain a smaller but strategically essential reference for science, public infrastructure and audit. Rights-cleared data will become a premium asset. Assurance, provenance and indemnity will become commercial products around artefacts that may themselves be free.

Delayed-open models will probably become more common for companies that need a temporary exclusivity window without promising permanent enclosure. Their legitimacy will depend on automatic conversion, stable terms, precise competitive restrictions and honest labelling. A delayed-open project is not open source during the restricted period. It can still be a useful bridge if the eventual conversion is real and enforceable.

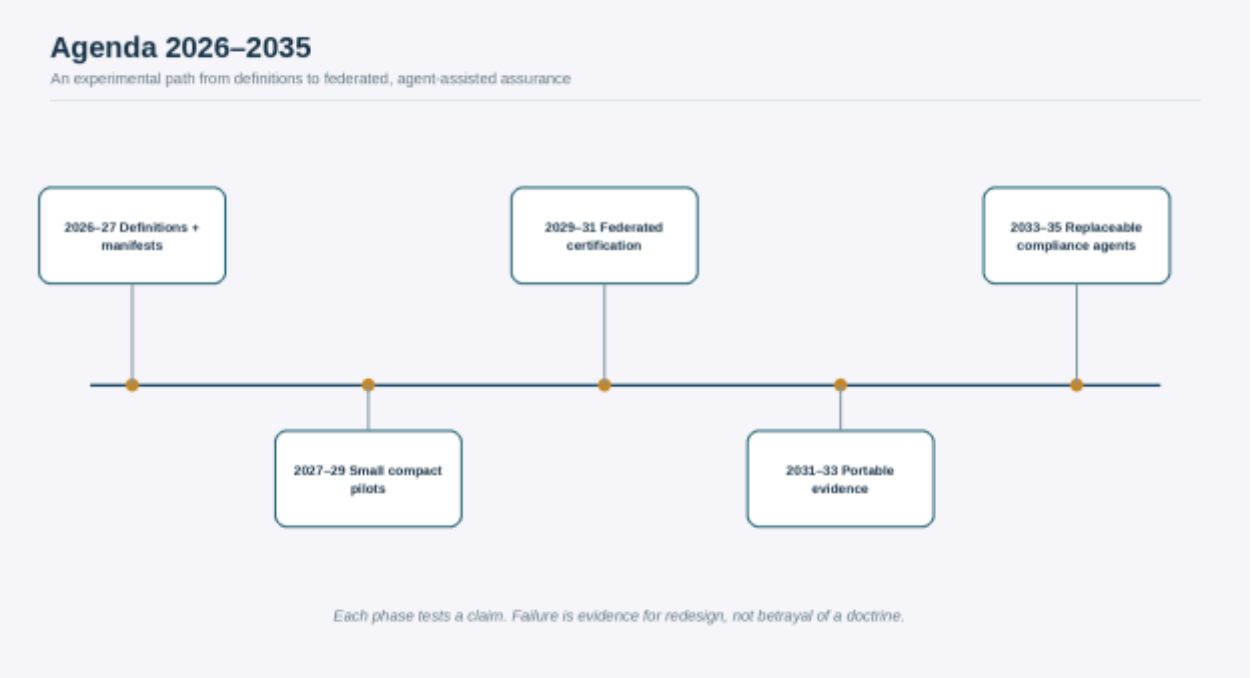
Licences and manifests will become transformation-aware. They will distinguish fine-tuning, adapters, merging, quantisation, distillation, synthetic-data generation and hosted service. A small number of standard profiles is more valuable than hundreds of creative licences. The legal text remains authoritative; machine-readable profiles become the operational interface through which agents preserve obligations.

The Compact of the Shared Name belongs to the Archipelago. It does not claim to replace companies, proprietary services or open-source foundations. It provides an intermediate form when several independent actors need a common identity and economy but do not wish to become subsidiaries, franchisees in substance, or invisible suppliers to a platform.

16.2 A research and implementation agenda, 2026-2035

An institutional theory becomes useful when it can fail in a measurable experiment. The coming decade will provide natural tests through open-weight models, data licences, code-generating agents, general-purpose AI regulation, certification networks and projects seeking economic sustainability. The agenda should begin with bounded pilots rather than a universal licence.

Figure 23. Phases of research and experimentation through 2035



The first phase clarifies terminology and builds minimal rights manifests. The second tests compact designs in small commercial ecosystems. The third develops federated certification and interoperable audit. The fourth explores portable evidence, collective data institutions and agents that administer compliance without acquiring normative sovereignty.

Table 19. Research agenda

Phase	Question to be tested
2026-2027: definition	Can openness levels, artefact genealogy and rights uncertainty be declared clearly?
2027-2029: pilots	Does the compact return value to maintenance and contribution without excessive bureaucracy?
2029-2031: federated assurance	Can several auditors and operators apply a common standard consistently and competitively?
2031-2033: portability	Can factual reputation, manifests and continuity evidence migrate between institutions without becoming a universal score?
2033-2035: agents	Can agents administer licensing and reciprocity while remaining replaceable tools under human governance?

Growth alone is not success. Evaluation should include sponsor concentration, maintenance stability, contributor retention, cost of compliance, remediation time, audit independence, portability, exit rates and survival after leadership change.

During 2026-2027 the priority is definition. "Open model", "open weights", "source available", "fair source", "responsible AI" and "fully open" should not be used interchangeably. The Open Source AI Definition supplies one reference for system-level freedoms, while disputes concerning access to training data and reproducibility will continue [OSI-OSAID]. A brand should declare the precise level of openness rather than use "open" as decoration.

The same phase requires machine-readable profiles. SPDX 3 and CycloneDX provide foundations for software, data, model and provenance descriptions [SPDX-3; CYCLONEDX-MLBOM]. AI profiles need operational fields for parent models, adapters, fine-tuning, merge, quantisation, distillation, synthetic data, hosted service and mark status. A compact can begin by publishing manifests for its own components and requiring human review when automated compatibility is uncertain.

The first pilot does not require a global ecosystem. Two or three independent companies, several maintainers and one customer-facing product are enough to test the separation between code and name. The pilot can use an established open-source licence, a certification or collective mark, a small maintenance fund and a simple contribution formula. The objective is to discover real conflict, not to demonstrate a utopia.

Legal classification should be part of the pilot. Counsel examines whether the level of control, assistance and payment creates a franchise or similar regulated relationship; whether the mark rules provide legitimate quality control; whether the economic coordination creates competition risk; and whether the contributor scheme creates tax, employment, charity or financial-regulation consequences. "Decentralised" is not an exemption from law.

Between 2027 and 2029, designs should be compared. One pilot may use Apache License 2.0 for the technical core and a modest mark-related contribution. Another may use AGPL for a service component and fixed node dues. A third may use delayed-open terms for a commercial layer. Research should track who joins, who leaves, what legal cost arises, how customers perceive the mark and whether maintenance receives more dependable funding than in a conventional project.

Contributor mechanisms need controlled comparison. One scheme can reserve fixed proportions for current maintainers, historical contributors and critical dependencies. Another can use peer-reviewed grants. A third can let nodes nominate upstream projects supported by evidence of use. Outcomes should be measured through maintenance response, security performance, retention, diversity, dispute volume and reduced dependence on unpaid emergency labour.

The economic formulas should be stress-tested against gaming and concentration. A large sponsor enters. A founding node loses most customers. A

maintainer controls a critical dependency. Automated agents generate contribution volume. An auditor becomes commercially dominant. The compact is credible only if it can respond without abandoning its declared promise.

From 2029 to 2031, mature ecosystems can develop federated certification and interoperable audit. Several accredited auditors apply the same public standard. Customers verify signatures, scope and provenance. Nodes migrate between infrastructure operators without losing institutional identity. Certification becomes a protocol of trust rather than a portal controlled by one company.

Regulation will increase the commercial value of this infrastructure. In the European Union, general-purpose AI model obligations concerning copyright policy and public summaries of training content have applied since 2 August 2025; the Commission's enforcement powers and fines for the GPAI regime begin on 2 August 2026, while certain pre-existing models have a later compliance date [EU-GPAI; EU-AI-ACT]. Provenance will increasingly affect procurement, not merely scholarship.

Current empirical evidence should be treated carefully. Recent preprints report large numbers of AI artefacts without declared licences, conflicts across software-model-data supply chains and loss of upstream obligations as applications adopt simple root licences [AI-LICENSING-RISK; LICENSE-LAUNDERING]. The exact percentages may change with methods and repositories. The durable signal is that rights information does not reliably survive transformation. Pilots should therefore measure information retention, not merely licence selection.

Between 2031 and 2033, partial interoperability among brands can emerge. A node certified in one ecosystem exports some evidence to another. Contributors carry signed histories. Mark permissions and manifests use standard identifiers. This must not become one global reputation score. Context, issuer, standard version and time remain attached, but the person or organisation need not begin from zero.

A market of constitutions can become visible. Customers compare the share reinvested in maintenance, concentration of voting power, independence of audit, remediation time, data-governance discipline and exit terms. Brands compete through verifiable legitimacy as well as advertising. Some will choose strong centres and fast execution. Others will distribute more authority. The important achievement is that the difference becomes intelligible.

Collective licensing institutions for data may also appear. Professional, editorial, scientific or cultural communities can aggregate clearly governed corpora, negotiate training rights and distribute revenue. They must avoid fictional consent, hidden collective monopoly and the erasure of individual or community claims. Rights-cleared data is valuable precisely because its governance is credible, not because a platform has declared it clean.

Between 2033 and 2035, agents can administer licensing at scale. A procurement agent checks whether a model can be used in a product, which

obligations apply and which nodes have the required assurance. A contributor agent traces use and prepares a distribution claim. A brand agent detects missing provenance and conflicting terms. Normative decisions remain with authorised humans; administrative cost falls.

The agent layer itself requires pluralism. Models, rules and providers must be replaceable. Explanations and evidence accompany decisions. Appeals can be processed outside the model that produced the original score. If one proprietary agent controls membership, certification, reputation and payment, the institution has rebuilt the Empire inside the Archipelago.

The decisive indicator is not token count, membership or gross revenue. It is the capacity to survive a founder transition, financial crisis, security incident, fork and entry of a very large sponsor without abandoning the public promise. A decentralised brand should be evaluated through institutional stress, not growth alone.

The agenda must accept negative results. Some sectors may need a central company and direct contracts. Contributor formulas may create more conflict than value. A certification mark may be sufficient without complex governance. A shared brand may be inappropriate where product liability requires one operator. Failure is evidence for institutional design, not betrayal of a creed.

The most plausible future is not one perfect licence. It is a small vocabulary of recognised patterns for code, models, data, provenance, mark custody and economic reciprocity. MIT, Apache and GPL became common grammar because users could understand their essential bargain. The Compact of the Shared Name is a testable hypothesis about one of the next patterns.

EPILOGUE. AFTER ABSOLUTE OWNERSHIP

The history of software licensing can be read as a sequence of attempts to separate things that the prevailing economy had kept together. IBM and the early software industry separated the program from the machine. Proprietary licensing separated use from ownership. GNU separated copyright from enclosure and converted it into an instrument of continuity. Permissive licensing separated collaboration from compulsory reciprocity. Open source separated adoption from the initial sale. Cloud computing separated use of a service from distribution of the program. Artificial intelligence now separates the model from the data, code, process, infrastructure and organisations that produced it.

The next separation concerns the brand. For more than a century, the modern trademark has joined a public promise to private ownership. That union created responsibility and investment. Whoever owns the name has both motive and legal capacity to defend it. It also created an asymmetry. Thousands of people can construct the experience, culture and trust associated with a name while the right to define and monetise its meaning remains concentrated.

Open source did not dissolve that asymmetry. It created the freedom to continue the program, not the freedom to continue its reputation. A fork can possess every line of code and no customers. A community can create a standard while one supplier controls the interface through which the market encounters it. Governance can be public while the mark, domain, package registry, conference and customer channel remain assets of one company.

Corporate capture of open source should be described precisely. It is usually not the unlawful taking of a thing. It is the lawful use of a commons combined with control of complementary assets: capital, compute, distribution, sales, data, operational evidence, indemnity and brand. Large companies have also made enormous contributions. That is why a simple morality tale fails. The same organisation can fund infrastructure and concentrate a market. The same cloud can make a technology accessible and weaken the economic position of its upstream project.

The response cannot be a general retreat into closed property. A world of bespoke licences, prohibitions and negotiated permissions would favour the actors able to maintain legal departments. Nor is universal free access sufficient. A commons without a budget, assurance or succession is vulnerable to exhaustion, abandonment and conditional rescue.

The middle path is a new separation. Code retains genuine open-source freedoms. The shared name becomes an institution of conditional access, transparent economy and limited custody. Autonomous nodes build businesses, preserve specific intellectual property and compete with one another. In return for

the common reputation, they finance maintenance, provenance, audit, continuity and contributor support.

This architecture cannot guarantee equity. A reward formula can be gamed. A chamber can be captured. A custodian can use emergency power to expand its mandate. A sponsor can control the institution through dependence. A reputation system can become surveillance. A certification standard can become an exclusionary barrier. Protection therefore lies in the plurality of mechanisms: separated powers, diversified revenue, objective criteria, competition safeguards, reasons and appeal, provenance, succession and bearable exit.

A decentralised brand is not a conventional company with a governance token. It is not a decentralised autonomous organisation that confuses voting with legitimacy. It is not merely an open-source repository with a discussion forum. It is an institutional grammar through which independent actors can carry one public promise without surrendering to a centre every means of interpreting, monetising and continuing that promise.

The comparison with franchising identifies both what should be preserved and what must be limited. The discipline of a common promise, public recognition and intervention against deception are valuable. Unilateral sovereignty over the name, market and method is not. Cooperatives demonstrate member participation but need not be the only legal form. Collective and certification marks provide tools for membership and standards. Open-source licences provide technical continuity. Provenance provides memory. The economic compact attempts to connect them without pretending that they are one instrument.

The need grows in an AI-dominated economy. Agents can consume the commons without meeting a maintainer or even naming a dependency. Models combine code and data through genealogies that final users rarely see. As implementation becomes more abundant, value moves towards identity, verification, distribution and the capacity to accept responsibility. Whoever controls the brand and registry can govern a market even when the underlying code is everywhere.

There is also an opportunity. AI can reduce the administrative cost of federation. It can preserve manifests, detect probable incompatibilities, simulate allocation formulas, translate deliberation and identify invisible maintenance. Under a constitution, these capabilities can allow plural institutions to achieve an efficiency previously reserved for central platforms. As an opaque governor, the same systems will convert old centralisation into a form that is harder to inspect and contest.

The argument therefore changes the licensing question. It is no longer only "who may copy the program?" It becomes "who may carry the promise, who finances continuity, who receives value, what evidence travels with the artefact, and who can leave without being erased?" These are questions of law, economics, reputation and power.

Licences were the small constitutions of software. In the next phase, they will be accompanied by constitutions of the names under which software, models and services enter social life. Property will not disappear. Companies, investment and centres will remain necessary. The objective is more modest and more difficult: property should not automatically claim that it created common value alone, and openness should not be used to request work without memory, security or economy.

A name deserves to be called shared only when those who enlarge it can participate in its promise, not merely carry its burden.

ANNEX A. MINIMUM TERMS FOR A PILOT COMPACT

This annex is not a contract template and does not replace legal analysis. It translates the book's proposal into a test specification for a small ecosystem. Its purpose is to expose decisions that should not remain implicit and to establish conditions under which the experiment can be simplified, corrected or abandoned.

Table 20. Minimum terms for a pilot

Element	Minimum decision
Public promise	A small set of verifiable claims rather than general values alone.
Commons and IP	Inventory, licence or rights basis, custodian and procedure for change.
Shared mark	Levels of use, quality control, certification, remediation, transition and appeal.
Economy	Node contributions, precise base, separated funds, reporting and change procedure.
Contribution	Identity where needed, rights, provenance and treatment of non-code work.
Governance	Plural representation, conflicts, competition safeguards and bounded emergency power.
Exit and succession	Transition, factual portability, cessation of mark use and continuity triggers.
Evaluation	Metrics, falsification criteria and a mandatory review after the first operating cycle.

The pilot should be simple, but simplicity is not silence. Missing ownership, undefined certification, opaque money or an absent exit process should be recorded as risk rather than celebrated as flexibility.

The public promise should fit in several verifiable sentences. "Innovation", "community" and "openness" are aspirations, not standards. A useful pilot promise might include technical compatibility, local deployment, accurate licence disclosure, a defined support period, a vulnerability-reporting process and a minimum contribution to the commons. Every public claim should have an owner, evidence and a remedy for material failure.

The common assets require an inventory. They may include reference code, documentation, tests, data schemas, model artefacts, the certification profile, registry, shared mark, manifests, domains and funds. For each asset, the pilot records the legal holder or custodian, applicable licence or agreement, technical operator, recovery path and amendment procedure. A plain register is more valuable than a solemn constitution that does not reveal who owns the domain or controls the signing key.

The technical core should use recognised licences wherever possible. Innovation should be concentrated where an unsolved problem actually exists: the mark, provenance, economic reciprocity and continuity. Customising every

software component produces avoidable compliance cost. Where the pilot uses source-available or delayed-open terms, the label and conversion conditions must be explicit.

Nodes should have an identifiable legal or accountable operator. A decentralised brand must not become a device for escaping responsibility. The customer contract remains with the node unless another arrangement is clearly stated. Certification can provide an additional remedy, common escalation or reserve; it does not make the operator anonymous. A non-commercial project can designate an association, foundation or responsible maintainer, but the public should know who answers.

The mark structure can begin with three levels. Descriptive compatibility requires only truthful reference. Certification requires compliance with a published technical or assurance profile. Official-node status adds membership, contribution, audit and customer-remediation duties. The pilot defines visual rules, evidence, duration, suspension, cure and appeal for each level. It should not impose the burden of full membership on every compatible implementation.

The economic formula can be modest. A tiered annual contribution may be combined with a limited percentage only after a threshold of revenue explicitly earned under the mark. Funds are separated among operation, maintenance, contributor support and continuity reserve. Exact percentages matter less initially than separate accounting, quarterly reporting and a rule preventing one sponsor from converting funding into constitutional control.

Revenue-linked payments require jurisdiction-specific review. The pilot should test the substance of the relationship against franchise, agency, competition, tax and consumer law. Nodes remain independent in price, customers, territory, staffing and commercial strategy. Meetings exclude competitively sensitive coordination. The compact defines quality and reciprocity, not a common commercial plan.

Contributions are recorded with artefact, date, origin, rights basis and class. Code can use a DCO or limited contributor agreement. Data, models and evaluations need separate declarations. Non-code work can be evidenced through approved artefacts, decisions, service records or signed reviews. The pilot should not create a total score for a person. It preserves only evidence relevant to maintenance, recognition, payment and governance.

Current maintenance is budgeted through contracts or grants. A small contributor pool can test historical or structural recognition, but it should not replace remuneration for ongoing work. The rules use materiality thresholds, bounded duration, explanation and appeal. Conventional accounting is preferable to transferable tokens.

Certification can begin with structured self-assessment and peer verification, followed by independent audit for higher-assurance claims. Criteria are public and related to the promise. A failing node receives reasons and a cure period unless there is fraud or immediate customer risk. Suspension is

documented and appealable. The certifier's payment should not depend on a positive decision.

A pilot board of five can be sufficient: two selected by nodes, two by contributors and one independent member with assurance or public-interest responsibility. Fundamental changes require four votes and support from both represented classes. Operational decisions are delegated. Conflicts are declared, and interested members recuse themselves from procurement and certification decisions affecting their organisations.

Emergency powers should be narrow. The board may temporarily suspend a mark use, isolate a key, publish a security notice or spend from a defined reserve. Every action receives a reason, expiry date and independent review. The pilot records emergency use as a metric because repeated emergencies usually reveal a design failure.

After twelve months, the experiment is reviewed. The report examines direct legal and administrative cost, time spent in governance, distribution of money, sponsor concentration, conflicts, certification time, customer understanding, maintenance response, contributor retention, portability and the practical feasibility of exit. Continuation is not automatic.

Falsification criteria should be declared in advance. The compact fails or requires radical simplification if nodes cannot understand the cost of joining, if audit consumes a disproportionate share of common income, if maintenance remains as precarious as before, if contributor distributions create more gaming than support, if customers cannot distinguish the levels of the mark, or if threat of withdrawal becomes a routine instrument against criticism.

Governance time deserves separate measurement. A network may appear plural because it holds many consultations while transferring participation cost to unpaid contributors. The report should estimate hours by category, who participates during paid working time and who contributes after hours. If voice depends on free administrative labour, formal decentralisation can conceal financial concentration.

The customer relationship should be evaluated outside internal satisfaction. Customers should be able to explain the difference between an official node, a certified implementation and an independent compatible fork. They should know who carries contractual responsibility, which remedies the brand adds, and what happens after suspension or exit. A network that distributes internal power but confuses the public has failed the central function of a brand.

The pilot also needs an orderly closure procedure. If it ends, the code, documentation, public provenance records, customer notices and disposition of funds are handled under the declared rules. The mark should not remain trapped in an empty entity that can later sell it contrary to the purpose. Capacity to close well is the first evidence that the institution serves its purpose rather than its administrators.

ANNEX B. THE RIGHTS AND PROVENANCE MANIFEST

The manifest is the operational interface between legal terms, technical artefacts and automated agents. Authoritative licence and contract texts remain controlling. Structured fields allow systems to determine what must be checked before an artefact is combined, distributed, certified or offered under the shared name.

Table 21. Core fields of the manifest

Field	Requirement
Identity	Name, version, hash, signature, provider, timestamp and status.
Artefact type	Code, dataset, weights, tokenizer, adapter, evaluation, container, service or documentation.
Parents and transformation	Sources and operations such as training, fine-tuning, merge, quantisation and distillation.
Rights and operations	Exact terms and separate claims for running, modifying, hosting, output use and redistribution.
Data and persons	Provenance, reservation or opt-out, legal basis, consent where relevant, retention and deletion.
Output and synthetic data	Declared rights, restrictions and permitted use for later training or distillation.
Brand and certification	Level, standard version, auditor, scope, date, expiry and revocation or remediation events.
Uncertainty and evidence	Unknowns are stated; every material claim points to evidence or an accountable declaration.

The manifest can describe several rights layers in one package without pretending that they have one licence. A human-readable view should explain what is open, restricted, inherited, uncertain and independently verified.

One package may contain Apache-licensed inference code, weights under community terms, a tokenizer under another licence, a dataset that can only be described, and a service governed by contract. The manifest keeps the layers distinct. It prevents the misleading statement that "the model is Apache" when Apache License 2.0 governs only one part.

Operations are more useful than broad labels. The manifest records running, copying, modification, fine-tuning, adapters, merging, pruning, quantisation, distillation, synthetic-data generation, redistribution, hosted service, benchmarking and API exposure. A result may be permitted, prohibited, conditional, not applicable or unknown. An agent must never convert "unknown" into "permitted".

Each claim includes its basis. The basis may be a copyright licence, contract, database permission, patent grant, public-domain dedication, legal exception, consent, policy or provider representation. Where the legal status of an artefact such as model weights is uncertain, the manifest records the instrument relied upon and the uncertainty. A structured assertion is not a substitute for rights.

Data provenance can be aggregated when publication of every source is impossible, unlawful or unsafe. The categories, period, collection method, principal licences or rights bases, opt-out handling, personal-data basis and custodian should still be described. Sensitive sources can refer to an audited statement, cryptographic commitment or controlled evidence rather than publish the content.

The manifest should distinguish intellectual-property permission from privacy and person-related rights. A copyright licence does not by itself establish a lawful basis for personal-data processing, consent to biometric use, permission to commercialise a voice or image, or legitimacy of extracting culturally governed knowledge. Separate fields and evidence are required where the risk exists.

Transformations are signed events. An adapter identifies the base model, training data and process. A distilled model identifies the teacher, relevant output terms and new training corpus. A merge identifies all material components and parameters. Quantisation preserves the source model and transformation tool. Inherited obligations and uncertainty therefore do not disappear when a new name is assigned.

The brand layer records whether an artefact is merely compatible, certified for a profile or official within the compact. Certification identifies standard version, auditor, evidence scope, date and expiry. Suspension or revocation appends an event and public reason where legally possible; it does not erase the historical record.

Automated systems can build a rights graph from these records. A commercial product that incorporates a non-commercial dataset is flagged. A model restricted from public hosting is not deployed automatically as an API. A package missing a required notice cannot receive complete certification. The agent supplies evidence and escalation rather than a fictional guarantee.

The format requires extensibility. Medical, financial, public-sector, defence, educational and creative domains will add fields. Extensions should preserve a common core, version namespaces and declare their governing authority. Interoperability should not require every sector to flatten its legal reality.

The brand registry need not store every artefact or sensitive document. It can preserve manifests, signatures, evidence locations and status events while artefacts remain with distributed custodians. Several registry implementations can verify the same signed records. This reduces the risk that the provenance system itself becomes the next universal platform.

Human readability is mandatory. Every machine-readable manifest should generate a concise page explaining what may be done, what conditions apply, what is not known, which claims were audited and who is responsible for correction. Machine-readable licensing should reduce legal fog, not turn it into a data format.

BIBLIOGRAPHY AND SOURCES

[AI-LICENSING-RISK] Wang, Bo, et al. "Hidden Licensing Risks in the LLMware Ecosystem." arXiv:2602.10758, 2026. Preprint. https://arxiv.org/abs/2602.10758

[AI2-OLMO] Allen Institute for AI. OLMo releases, data, training code, checkpoints, evaluations and documentation. https://allenai.org/olmo

[ALBOAIE-DB] Alboaie, Sînică. *Decentralised Brands: Shared Identity, Distributed Power, and the Architecture of Resilient Cooperation*. Research manuscript, Version 0.1, July 2026.

[APACHE-2] Apache Software Foundation. *Apache License, Version 2.0*. https://www.apache.org/licenses/LICENSE-2.0

[ASF-CLA] Apache Software Foundation. Contributor agreements and project governance documentation. https://www.apache.org/licenses/contributor-agreements.html

[BSL] MariaDB plc. *Business Source License 1.1*. https://mariadb.com/bsl11/

[CC-AI] Creative Commons. *Using CC-Licensed Works for AI Training*, 2025. https://creativecommons.org/using-cc-licensed-works-for-ai-training-2/

[CDLA] Community Data License Agreement. CDLA-Permissive-2.0 and related data agreements. https://cdla.dev/permissive-2-0/

[COOPCYCLE] CoopCycle. Federation, governance and Coopyleft licensing materials. https://coopcycle.org/en/federation/

[CYCLONEDX-MLBOM] CycloneDX. Machine Learning Bill of Materials capability and specification. https://cyclonedx.org/capabilities/mlbom/

[DCO] Developer Certificate of Origin, Version 1.1. https://developercertificate.org/

[DEEPSEEK-R1] DeepSeek AI. DeepSeek-R1 repository, model documentation and licensing. https://github.com/deepseek-ai/DeepSeek-R1

[DOJ-CERTIFICATION] United States Department of Justice, Antitrust Division. Business review materials concerning open, voluntary and independently administered certification programmes. https://www.justice.gov/atr/business-review-letters-and-request-letters

[ELASTIC-AGPL] Elastic. "Elasticsearch Is Open Source Again." 2024. https://www.elastic.co/blog/elasticsearch-is-open-source-again

[ELASTIC-LICENSING] Elastic. Licensing FAQ and history for Elasticsearch and Kibana. https://www.elastic.co/pricing/faq/licensing

[EU-AI-ACT] European Parliament and Council. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence. https://eur-lex.europa.eu/eli/reg/2024/1689/oj

[EU-DSM] European Parliament and Council. Directive (EU) 2019/790 on copyright and related rights in the Digital Single Market. https://eur-lex.europa.eu/eli/dir/2019/790/oj

[EU-GPAI] European Commission. Guidelines, templates and compliance information for providers of general-purpose AI models. https://digital-strategy.ec.europa.eu/en/policies/guidelines-gpai-providers

[EU-HORIZONTAL] European Commission. Guidelines on the applicability of Article 101 TFEU to horizontal co-operation agreements, 2023/C 259/01. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52023XC0721(01)

[EUIPO-MARKS] European Union Intellectual Property Office. Guidance on individual, collective and certification marks. https://www.euipo.europa.eu/en/trade-marks/before-applying/types-of-trade-marks

[FSF-GNU] Free Software Foundation. The GNU Project and the history of free software. https://www.fsf.org/history

[FSL] Fair Source. Functional Source License and Fair Source principles. https://fair.io/

[FTC-ASSOCIATIONS] United States Federal Trade Commission. Guidance concerning trade associations, standard-setting and exchange of competitively sensitive information. https://www.ftc.gov/advice-guidance/competition-guidance/guide-antitrust-laws/dealings-competitors

[FTC-FRANCHISE] United States Federal Trade Commission. Franchise Rule and compliance guidance. https://www.ftc.gov/legal-library/browse/rules/franchise-rule

[FTC-PRICE] United States Federal Trade Commission. Guidance on price fixing and dealings among competitors. https://www.ftc.gov/advice-guidance/competition-guidance/guide-antitrust-laws/dealings-competitors/price-fixing

[GEMMA-TERMS] Google. Gemma Terms of Use and model-specific terms. https://ai.google.dev/gemma/terms

[GITHUB-SPONSORS] GitHub. GitHub Sponsors documentation. https://docs.github.com/en/sponsors

[GNU-GPL-HISTORY] GNU Project. GPL history, licence texts and frequently asked questions. https://www.gnu.org/licenses/gpl-faq.html

[HASHICORP-BSL] HashiCorp. "HashiCorp Adopts Business Source License." 2023. https://www.hashicorp.com/en/blog/hashicorp-adopts-business-source-license

[HIRSCHMAN-1970] Hirschman, Albert O. *Exit, Voice, and Loyalty: Responses to Decline in Firms, Organizations, and States*. Harvard University Press, 1970.

[IBM-1969] Computer History Museum. IBM unbundling and the emergence of software as a separate product, 1969. https://www.computerhistory.org/revolution/mainframe-computers/7/172

[ICA-PRINCIPLES] International Cooperative Alliance. Cooperative identity, values and principles. https://ica.coop/en/cooperatives/cooperative-identity

[KERNEL-COPYING] Linux Kernel Documentation. Kernel licensing rules, GPL-2.0-only and Linux-syscall-note. https://www.kernel.org/doc/html/latest/process/license-rules.html

[LERNER-TIROLE] Lerner, Josh, and Jean Tirole. "The Simple Economics of Open Source." NBER Working Paper 7600, 2000; and "The Scope of Open Source Licensing." NBER Working Paper 9363, 2002. https://www.nber.org/papers/w7600 ; https://www.nber.org/papers/w9363

[LF-FUNDING-2024] Linux Foundation Research. Reports on open-source funding, contributors and the economics of open source. https://www.linuxfoundation.org/research

[LF-TRADEMARK] Linux Foundation. Trademark usage guidelines. https://www.linuxfoundation.org/legal/trademark-usage

[LICENSE-LAUNDERING] Jewitt, James, et al. "Don't Trust the Label: License Laundering in AI Supply Chains." arXiv:2607.20300, 2026. Preprint. https://arxiv.org/abs/2607.20300

[LINUX-MARK] Linux Foundation. The Linux Mark and trademark sublicensing programme. https://www.linuxfoundation.org/legal/the-linux-mark

[MONTREAL-DATA] Benjamin, M., et al. "Towards Standardization of Data Licenses: The Montreal Data License." arXiv:1903.12262, 2019. https://arxiv.org/abs/1903.12262

[NETSCAPE-MOZILLA] Netscape and Mozilla archives. Source-code release announcement and source FAQ, 1998. https://www-archive.mozilla.org/src-faq

[ODC-LICENSES] Open Data Commons. Open Database License, ODC Attribution License and Public Domain Dedication and License. https://opendatacommons.org/licenses/

[OPENAI-GPTOSS] OpenAI. gpt-oss open-weight models, model card and official documentation. https://openai.com/open-models/

[OPENRAIL] Responsible AI Licenses. OpenRAIL licence family and frequently asked questions. https://www.licenses.ai/

[OPENSEARCH] OpenSearch Project. Project history, governance and Apache License 2.0 distribution. https://opensearch.org/

[OPENSSF-GRANTS] Open Source Security Foundation. Programmes supporting open-source security and maintainers. https://openssf.org/

[OPENTOFU] OpenTofu. Project manifesto, governance and Linux Foundation stewardship. https://opentofu.org/

[OSI-HISTORY] Open Source Initiative. History of the Open Source Initiative. https://opensource.org/about/history-of-the-open-source-initiative

[OSI-OSAID] Open Source Initiative. Open Source AI Definition 1.0 and FAQ. https://opensource.org/ai/open-source-ai-definition

[OSI-OSD] Open Source Initiative. The Open Source Definition. https://opensource.org/osd

[REDIS-AGPL] Redis. "Redis Returns to Open Source under AGPLv3." 2025. https://redis.io/blog/agplv3/

[REDIS-LICENSING] Redis. Licensing documentation and history. https://redis.io/legal/licenses/

[SCHWEIK-ENGLISH] Schweik, Charles M., and Robert C. English. *Internet Success: A Study of Open-Source Software Commons*. MIT Press, 2012.

[SPDX-3] SPDX. SPDX Specification 3.0 and AI, data and build profiles. https://spdx.github.io/spdx-spec/v3.0/

[TORVALDS-GPL] Torvalds, Linus. Historical release notes and messages concerning the Linux kernel's transition to GPLv2, 1992. https://www.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.12

[USCO-AI3] United States Copyright Office. *Copyright and Artificial Intelligence, Part 3: Generative AI Training*. Pre-publication report, 2025. https://www.copyright.gov/ai/

[USPTO-CERTIFICATION] United States Patent and Trademark Office. Trademark Manual of Examining Procedure and guidance on certification marks. https://tmep.uspto.gov/

[USPTO-COLLECTIVE] United States Patent and Trademark Office. Trademark Manual of Examining Procedure and guidance on collective marks. https://tmep.uspto.gov/

[USPTO-RELATED] United States Patent and Trademark Office. Trademark Manual of Examining Procedure §1201.03, use by related companies and control over nature and quality. https://tmep.uspto.gov/

[VALKEY] Valkey. Open-source in-memory data store under Linux Foundation governance. https://valkey.io/

[VIBE-CODING-OSS] Koren, Miklós, et al. "Vibe Coding Kills Open Source." arXiv:2601.15494, 2026. Preprint. https://arxiv.org/abs/2601.15494