

BRAND MACHINES

A General Theory of Brand Systems

VIKTOR BERTHELIUS

BRTHLS

Brand Machines

A General Theory of Brand Systems

© 2026 Viktor Berthelius. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without the author’s prior written permission.

First English edition: September 2026.

ISBN (paperback): 9798176733549.

ISBN (hardcover): 9798176734225.

Design and composition: BRTHLS.

Typefaces: EB Garamond, Source Sans 3, Source Code Pro.

Typesetting: Quarto + LuaLaTeX.

Spanish technical editing assisted by Claude (Anthropic).

English translation and review assisted by Codex (OpenAI).

Published by BRTHLS.

viktor@brthls.com

Contents

Manifesto	xi
Introduction	xiii
How to Read This Book	xvii
A Note on Updates	xix
I Foundations of a New Era	I
1 The Death of the Industrial Branding Paradigm	3
2 Machine Society and Hybrid Brands	39
3 What Is a Brand Machine	77
4 What a Brand Machine Is NOT	103
II A General Theory of Brand Systems	117
5 The Five Ontologies of the Brand	119
6 Anatomy of a Brand Machine: The 7 Layers	131
7 Layer Theory	145
8 Evolution: Brand Book → Design System → Brand OS → Brand Machine	163
9 The Physics of the Brand: Chaos, Order, and Energy	179
10 Brand Language: Symbolic and Narrative Systems	195

III	The Architecture of a Brand Machine	211
11	The Core: Operational Identity	213
12	The Mind: Distributed Intelligence	233
13	The Body: Modular Architecture	245
14	The Skin: Expanded Experience	259
15	The Engines: Brand Engines	277
16	The Operating System: Brand OS	297
17	Interconnections: APIs, Integrations, and Ecosystems	315
IV	Design, Method, and Disruption	333
18	Principles of Systems Design	335
19	Transdisciplinarity: The Designer's New Role	349
20	Frameworks, Models, and Tools	361
21	Chaos and Order: Designing the Undesignable	377
22	Disruption: How to Create Evolutionary Leaps in a Brand	393
V	The Future and Expansion	407
23	Autonomous Brands: The Horizon of Artificial Intelligence	409
24	Continuous Updating: The Brand as Living Software	423
25	Cultural Infrastructures: Brands as Architectures of Meaning	437
26	Epilogue: The Architect of the New Symbolic Order	449
	Notes and Sources	457
	Glossary	461
	About the Author	467
	Colophon	469

Manifesto

BRANDING was born for a stable world. A world of long cycles, hierarchical structures, and predictable change. That world has disappeared.

Today we operate in an interconnected, accelerated, and profoundly complex environment, where brands no longer live in manuals, campaigns, or closed visual identities, but in distributed systems of meaning, technology, culture, and behavior. Yet we continue trying to design them with tools conceived for another era.

This book begins with a simple and radical conviction: **brands can no longer be understood as objects. They must be understood as systems.**

For decades, branding focused on defining *what* a brand is. Today the relevant question is different: *how does a brand operate, evolve, and adapt in real time?*

Static identities have been overtaken by dynamic contexts. Closed brand books by systems that change. Imposed coherence by emergent coherence. Linear planning by adaptive behavior.

In this new landscape, the brand ceases to be a representation and becomes an infrastructure: a living architecture in which identity, narrative, technology, and culture interact continuously.

We call this new kind of entity a **Brand Machine**.

A Brand Machine is not a metaphor. It is not advanced automation. It is not an aesthetic evolution of traditional branding.

It is a hybrid system—human and technological—capable of operating, learning, and transforming without losing coherence. An organism composed of interdependent layers: symbolic, narrative, operational, structural, and algorithmic. A system designed not only to communicate, but to produce meaning, action, and value over time.

This book does not propose a closed method or a recipe to be applied mechanically. It proposes a conceptual framework for thinking about brand through complexity, interconnection, and evolution.

Here, traditionally separate disciplines converge: design, architecture, technology, systems theory, visual culture, narrative, organization, and software. Not as a sum of parts, but as a transdisciplinary field in which the brand is understood as an operational whole.

Designing a Brand Machine does not mean eliminating chaos. It means orchestrating it.

Order is no longer a final state, but a temporary condition. Stability is not permanence, but the capacity to adapt. Coherence is not imposed: it emerges.

In this context, the role of the designer, the strategist, and the creator changes radically. It is no longer a matter of defining forms, messages, or campaigns, but of designing systems capable of evolving.

Brand Machines is not a book about trends. It is a book about structures.

It looks not to the next quarter, but to the next decade. It answers not “what to do,” but “how to think.”

It is written for those who do not want to optimize what exists, but to build systems that do not yet have names.

This book does not seek to close the debate. It seeks to open a discipline.

7

Layer Theory

Why every enduring system organizes itself in layers, what exactly happens at the interfaces between them, and how to recognize one that has collapsed.

Why layers are inevitable

EVERY complex system organizes itself in layers. Your body has layers. Your city has layers. The software you are using to read this has layers. It is not a design choice—it is an emergent property of complexity itself. And brands are no exception.

Operating systems have layers: hardware, kernel, drivers, applications, user interface. Each layer speaks to adjacent layers through defined interfaces. No layer needs to understand how the others work—it only needs to respect the communication contract.

Communication networks have layers: the OSI model defines seven, from the physical layer (cables, electrical signals) to the application layer (the browser you are using to read this). An email travels through all the layers without any of them needing to understand the message's content.

Biological ecosystems have layers: atoms, molecules, cells, tissues, organs, organisms, populations, ecosystems. Photosynthesis occurs at the cellular level, but its effects reverberate to the ecosystem level.

Cities have layers: underground infrastructure (water, electricity, subway), street level (buildings, businesses), aerial level (communications, drones), social level (communities, economies).

Why this universal convergence toward layered structures?

Because layers solve a fundamental problem: **how to manage complexity without collapsing under its weight.**

The problem of combinatorial complexity

Imagine a system with 100 components.

If every component could interact directly with every other, you would have 4,950 possible connections. Every time you modify a component, you potentially affect 99 others. Every design decision has unpredictable consequences. The system becomes unmanageable.

Now imagine that you organize those 100 components into five layers of 20 components each.

Every component interacts only with components in its own layer and the immediately adjacent layers. The number of possible connections falls dramatically. Modifying one component affects only its local neighborhood. The system becomes comprehensible.

Layers are the fundamental strategy for reducing complexity.

The three principles of layered architecture

Principle 1: Upward abstraction

Each layer hides the complexity of the layers below.

An application user does not need to know how the processor's transistors work. The designer of a brand campaign does not need to understand how the server hosting the website works. Each layer provides a simplified interface that abstracts the underlying complexity.

In a Brand Machine:

- The **Skin** does not need to know how the **Mind** works
- It only needs to know which outputs it receives and which inputs it must deliver
- The complexity of cognitive processing remains hidden

Principle 2: Downward dependency

The upper layers depend on the lower layers, but not the reverse.

The Core of a Brand Machine does not depend on how the Skin is implemented. If you change the visual expression, the foundational purpose remains intact. But if you change the Core, all the upper layers must be recalibrated.

This asymmetry is not a defect—it is a feature. It allows the upper layers to evolve rapidly while the lower layers maintain stability.

Principle 3: Contractual interfaces

Layers communicate through contracts, not implementations.

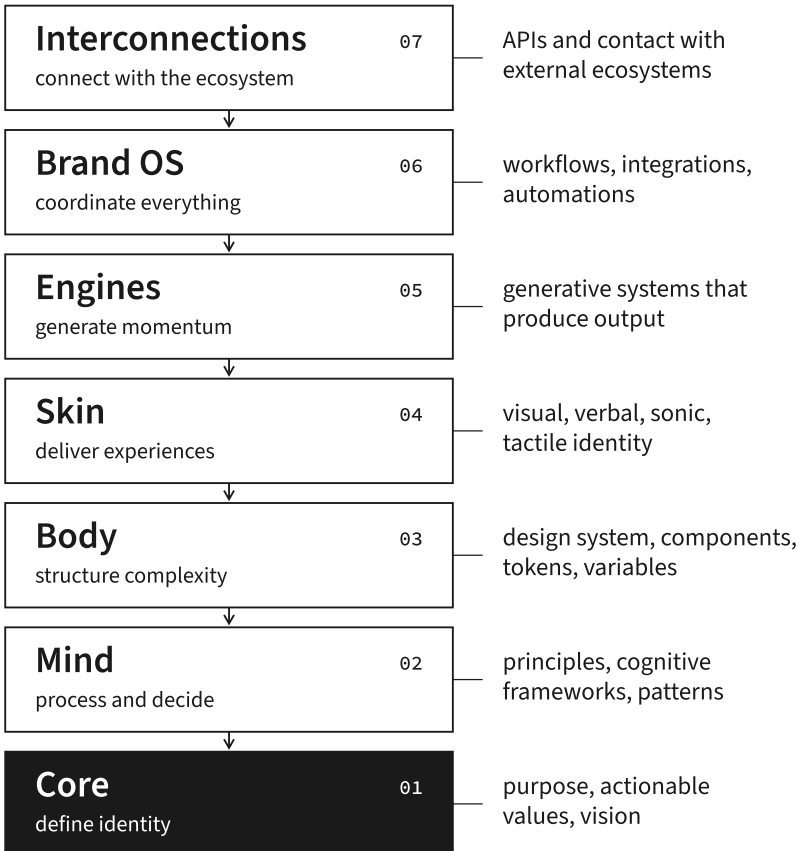
The Mind layer does not tell the Body layer *how* to structure a component. It tells it *what* it needs. How the Body solves it is the Body's concern.

These contracts are the Brand Machine's **internal APIs**.

These three principles govern a concrete stack: seven layers, each with a function no other can absorb.

The seven layers

below, what does not change; above, what changes every day



Coherence is measured at the interfaces, not in the layers.

Upper layers depend on lower ones, never the reverse.

Figure 7.1. The vertical order is one of time, not importance: that is why a Core decision forces a review of the six layers above it, and never the reverse.

Anatomy of a layer

Every layer of a Brand Machine has the same internal structure.

Components of a layer

1. Upper interface

Defines which services this layer offers the layers above it.

For the Body layer, the upper interface might specify:

- I can provide visual components in any format
- I can assemble layouts for any platform
- I can scale any asset to any resolution

The Skin layer does not need to know how the Body does this. It only needs to know it can request it.

2. Lower interface

Defines which services this layer requires from the layers below.

For the same Body layer, the lower interface might specify:

- I need tone guidelines from the Mind layer
- I need foundational values from the Core layer
- I need decision parameters for ambiguity

3. Internal modules

Each layer contains modules that implement its specific functionality.

In the Body layer:

- Typography module
- Color module
- Layout module
- Iconography module
- Motion module

Modules can communicate with one another inside the layer. The color module needs to speak to the typography module to guarantee legible contrast.

4. Internal state

Each layer maintains its own state.

The Mind layer maintains state about previous decisions, accumulated learning, detected patterns. The Body layer maintains state about existing components, generated variations, historical use.

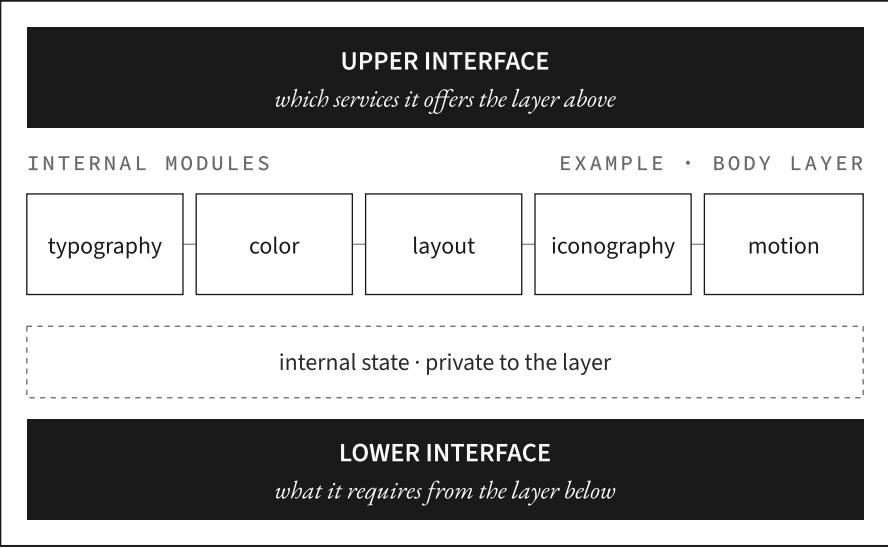
This state is *private* to the layer. The other layers do not access it directly—only through the interfaces.

Anatomy of a layer

two interfaces, modules, and a state that does not leave this layer

LAYER N

layer N+1



layer N-1

THE CONTRACT

The neighboring layer requests by contract, never by implementation.

Figure 7.2. What does not leave the layer matters as much as what it offers: private state is what makes it possible to rework the inside without notifying the neighboring layers.

Flows between layers

Layers do not exist in isolation. They communicate constantly.

Downward flow: From purpose to expression

The most basic flow is downward: from the Core toward Interconnections.

Each layer *translates* the message from the layer below into its own language.

The Core speaks in terms of existential purpose. The Mind translates it into decision principles. The Body translates it into technical specifications. The Skin translates it into perceptible experiences.

There is no loss in translation if the interfaces are well defined. There is *transformation*.

Upward flow: From signals to adaptation

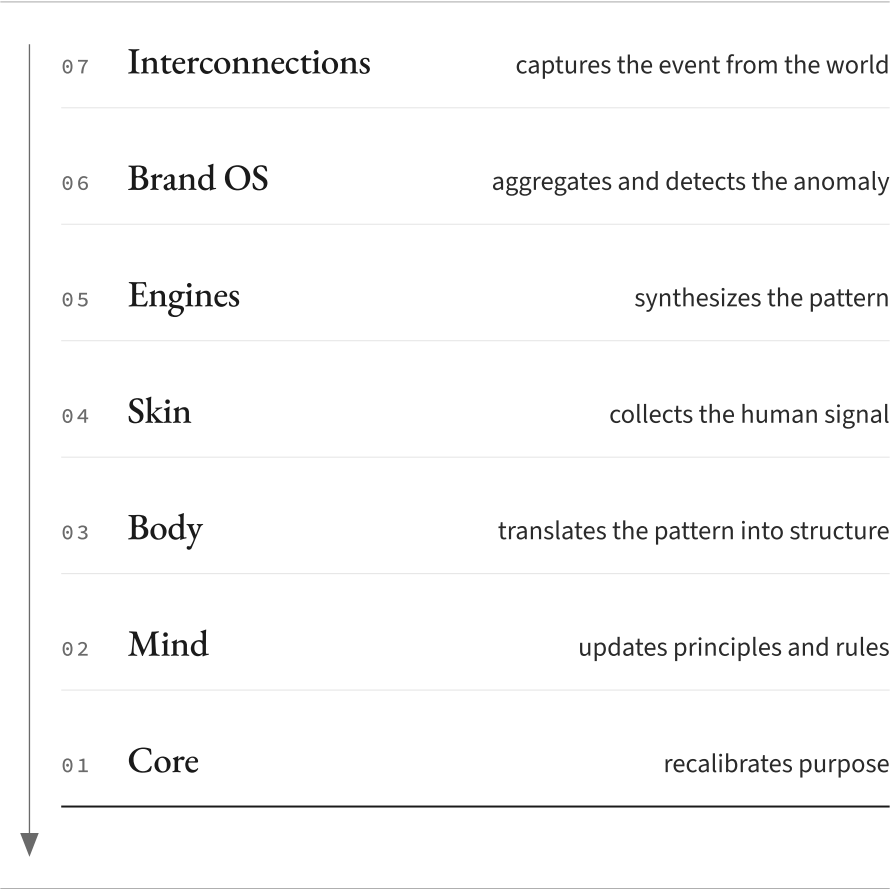
The upward flow is equally important.

This upward flow is **feedback**: information from the real world that allows the system to adapt.

Without an upward flow, the Brand Machine is blind. It can execute an incorrect strategy perfectly.

Upward flow

if the flow is blocked, the system operates open-loop and repeats errors



THE PRINCIPLE

Without upward flow, the Brand Machine is blind.

It can execute the wrong strategy perfectly.

Figure 7.3. Seven chained translations: what enters is noise from the world and what leaves is a change in purpose. Breaking one link is enough to lose it.

Horizontal flow: Coordination between modules

Within each layer, modules coordinate horizontally.

This horizontal coordination is *negotiation*. Each module has its own constraints. The result is a compromise that meets the needs of the complete system.

Emergent properties

When the layers interact correctly, properties emerge that no individual layer possesses.

Systemic coherence

No individual layer produces coherence.

Coherence emerges from *fidelity in the translations* between layers.

If the Core says “democratization of knowledge” and the Skin delivers “elitist experiences,” there is incoherence. Not because a layer failed individually, but because the translation between layers became corrupted.

The coherence of a Brand Machine
is measured at the *interfaces*, not in
the layers.

Distributed resilience

If one layer fails, the others can compensate temporarily.

If the Mind fails, the Body uses cached previous decisions. The Brand OS escalates to human intervention. The Skin falls back to safe variants. The system degrades gracefully instead of collapsing.

Independent evolution

Each layer can evolve at its own pace.

Apple's Core has not changed substantially since 1997. But its Skin has evolved radically: from the iMac's Bondi Blue to today's minimalism.

This asynchronous evolution is possible because the layers are decoupled. As long as the interfaces are respected, each layer can innovate internally without breaking the system.

Modular scalability

Adding capacity to one layer does not require modifying the others.

If the Engines layer needs more power, you can add new engines without touching the Core or Skin. The new engines simply implement the existing interfaces.

This is the difference between scaling a Brand Machine and scaling a traditional brand.

Traditional brand: every new channel requires an alignment meeting, a new manual, constant supervision.

Brand Machine: every new channel is a new module that inherits coherence automatically.

Pathologies of the layer system

Layers can become ill. Recognizing the pathologies is the first step toward curing them.

Collapsed layers

When two layers merge and lose their distinction.

Symptom: The design team makes strategic decisions without a clear distinction.

Consequence: Tactical decisions contaminate strategic ones. The immediate is optimized at the expense of the fundamental.

Treatment: Restore the distinction. Define clearly which decisions belong to each layer.

Broken interfaces

When layers do not communicate correctly.

Symptom: The Skin does not know what the Mind wants. The Body produces components no one uses.

Consequence: Every layer optimizes locally, but the overall system is chaotic.

Treatment: Rebuild the interfaces. Explicitly document which information flows between layers.

Hypertrophied layers

When a layer grows disproportionately.

Symptom: The company has 50 people producing assets but only two thinking about strategy.

Consequence: High execution capacity without clear direction.

Treatment: Rebalance resources.

Ossified layers

When a layer stops evolving.

Symptom: The Core has not been reviewed in 15 years.

Consequence: The brand falls out of sync with the world.

Treatment: Introduce mechanisms for periodic renewal.

Layers without feedback

When the upward flow is blocked.

Symptom: The Interconnections layer detects problems, but the information never travels upward.

Consequence: The system operates in an open loop. It repeats errors indefinitely.

Treatment: Create explicit feedback channels.

Designing layer systems

The design of a layer system begins with the hardest question: *which functions are irreducible?* If you can collapse two functions into one without losing capacity, they are not two layers—they are one. For a Brand Machine, we arrive at seven: defining identity, processing and deciding, structuring complexity, delivering experiences, generating momentum, coordinating everything, and connecting with the ecosystem. Each answers a question no other can absorb.

Once identified, the order emerges naturally from dependency: the Skin cannot deliver experiences unless the Body provides components; the Body cannot structure unless the Mind defines criteria. What remains to be defined are the interfaces (which information flows between adjacent layers), internal modules (specific to each organization), and governance (who is responsible for each layer). Without the last, the layers come into conflict.

The invisible layer: Culture

There is a layer that does not appear in the diagram but permeates all the others.

Organizational culture.

Calling it layer 0 or layer 8 is a convenience of exposition, not an eighth layer. Culture is not numbered because it does not occupy a place in the stack: it is transversal. It exists before the Core and after Interconnections, runs through every interface, and therefore cannot be isolated like the other seven. It is everywhere and nowhere. There are still seven layers; culture is the medium in which they breathe.

Culture determines:

- How the Core's values are interpreted
- How the Mind decides
- How the Body structures
- How the Skin delivers

You can design all seven layers perfectly and fail if the culture does not sustain them.

A culture of silos will sabotage the interfaces. A short-term culture will atrophy the Core. A culture of control will prevent distributed intelligence. A culture of perfectionism will paralyze the Engines.

Culture is the reason two organizations can implement the same seven layers and obtain radically different results. One uses them as living architecture. The other turns them into additional bureaucracy. The difference is not in the design of the system—it is in the people who inhabit it.

That is why Brand Machines are not merely technical architecture. They are **sociotechnical architecture**. Half is system, half is people. And people are, by far, the hardest part to design.

This is the limit of pure theory. The layers explain *how* a Brand Machine is organized. The next chapter traces *where they come from*: the historical evolution from brand book to generative machine. Because no architecture is comprehensible without its genealogy.

Questions for reflection

1. Can you name, layer by layer, who decides in each of the seven? If two layers return the same name, they have collapsed.
2. Is there a document stating what the Mind delivers to the Body and what it expects in return? If it does not exist, that interface is not contractual: it is habit.
3. Distribute your team across the seven layers. Does one contain more than half?
4. On what date was the Core last reviewed? Write it down before you continue reading.
5. Take one signal the Interconnections detected last month: in what concrete decision does it appear today? If it appears in none, the upward flow is blocked.
6. Can you redesign the Skin without touching the Body? Count how many components you reopen.

7. Of the five pathologies—collapsed layers, broken interfaces, hypertrophied layers, ossified layers, layers without feedback—how many can you rule out with concrete evidence?

