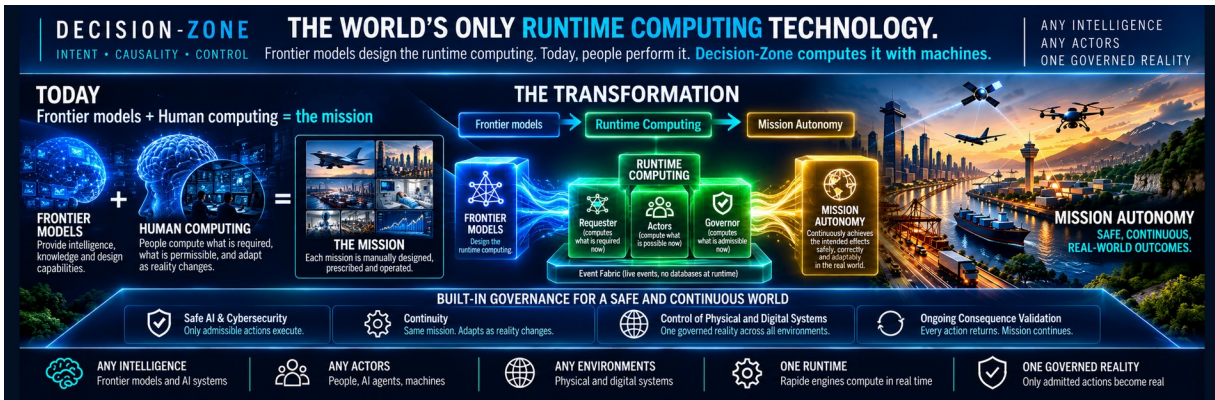




THREE REALMS

Keeping the requester, the thinker and the governor apart. In a life, and in a machine.

Nobody who proposes an action should also approve it.

That is a way of living before it is an architecture.

It is also the one thing missing from every intelligent system built so far.

A note written alongside the Runtime Computing architecture.

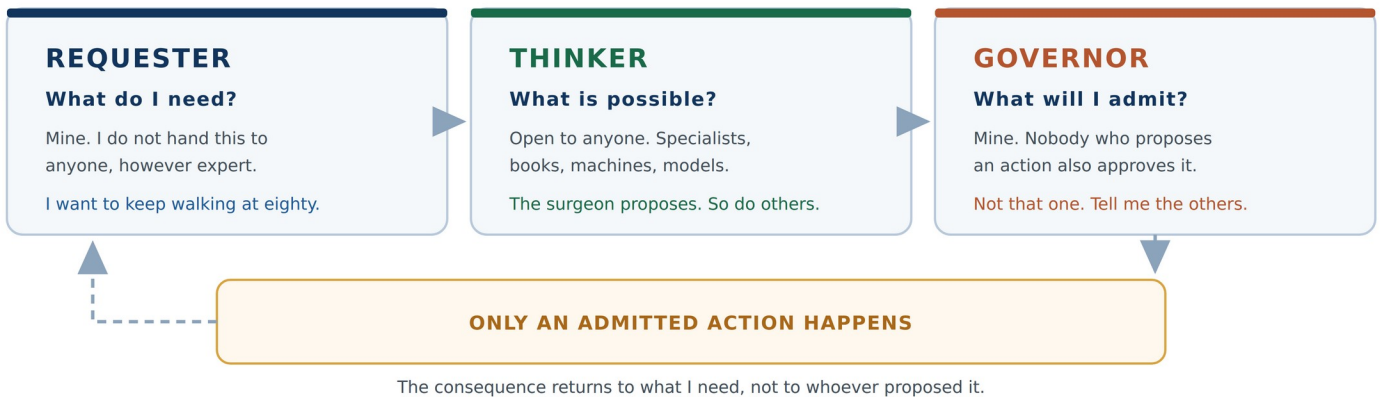
Rajeev Bhargava • Decision-Zone Inc. • October 2026

The separation is lived before it is engineered

Three questions run in a life, all the time. What do I need. What is possible. What will I admit. They are different questions, they belong in different hands, and keeping them apart is a discipline rather than a theory.

A specialist is a source of possibilities. That is the whole of their role and it is a considerable one. What they are not is the person who decides what is required, or the person who approves the action.

THREE REALMS, KEPT SEPARATE, IN AN ORDINARY LIFE



WHAT MOST PEOPLE DO INSTEAD

They hand all three to the specialist at once. Tell me what is wrong, tell me what to do, I will do it.
The specialist becomes requester, governor and actor in one person. The patient becomes the thing acted on.
Nothing in that arrangement is computing what the person actually needs.

Nothing else is computing it

When all three end up with whoever has the expertise, it is not carelessness. It is that no instrument exists to hold the other two, so they fall to whoever is standing closest. The person is doing the hardest computation in the room with nothing to do it with.

Three ways a life or a system fails

Each of the three can run on its own, and each produces a recognizable failure. None of them look like each other, and only one of them is usually called a problem.

Only the actor running is compulsion: doing, with nothing watching. Only the governor running is paralysis: refusing, with nothing required. Only the thinker running is endless possibility: capability with no judgment attached to it.

ONE PART RUNNING ALONE. THREE DIFFERENT FAILURES.



THE THIRD ONE IS THE MOST COMFORTABLE PLACE TO SPEND A CAREER

The third one does not look like a failure

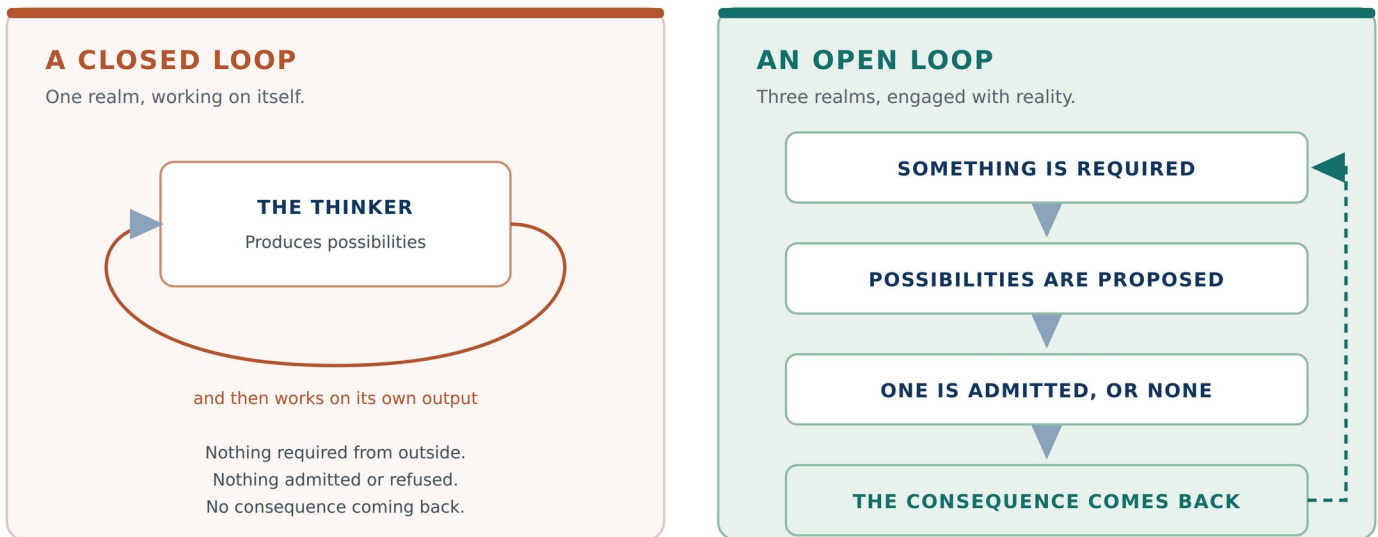
It looks like productivity. It looks like research. A person or an institution can spend decades generating possibilities, be rewarded the whole time, and never once compute what was required or what was admissible. That is the state the technology industry is in.

03 / THE CLOSED LOOP

A realm working on itself is still one realm

There is an old idea that the way to improve conduct is to work on your own mind. Quiet it, discipline it, refine what it produces.

Structurally that is the thinker operating on the thinker. Generating and suppressing are the same realm doing both. Nothing is required of it from outside, nothing is being admitted or refused, and no consequence is returning. It is a closed loop with one part running.



TRAINING A MODEL ON ITS OWN OUTPUTS TO SHAPE ITS OWN OUTPUTS IS THE LEFT-HAND PICTURE

This is also what model alignment is

A model trained on its own outputs, to shape its own outputs, is the left-hand picture exactly. There is no requester outside it stating what is required now, and no governor outside it deciding what is admissible now. The result is a better behaved thinker. It is not a governed system, and no amount of further training makes it one.

What the frontier model is for

Frontier models are the most capable artifacts anyone has ever built. They deserve the standing they have. In this architecture they hold the role that makes everything else possible, which is the compilation of the mission itself.

A person states, in their own words, what is required, what is permitted and what the limits are. The model turns that into a running program: it writes the requester, it writes the actor interfaces, and it writes the governor. Without a model capable of doing that, declaring a mission would need engineers, and the whole idea would stay out of reach of the people who actually hold the missions.

DESIGN TIME

THE FRONTIER MODEL COMPILES THE MISSION

A person says, in their own words, what is required, what is permitted and what the limits are.

WRITES THE REQUESTER

What the mission will require

WRITES THE ACTOR INTERFACES

What may be proposed, and by whom

WRITES THE GOVERNOR

What will be admitted, and never

RUNTIME

THE THREE RUN SEPARATELY. THE MODEL IS NOT IN THE LOOP.

REQUESTER RUNS

Rapide engine

ACTORS PROPOSE

Any model, machine or person

GOVERNOR ADMITS

Rapide engine, separate

A compiler can emit something stricter than itself.

What makes a governor a governor is that it runs apart and nothing can bypass it.

Writing the three is not being the three

A compiler routinely emits something stricter than itself. What makes a governor a governor is not the words it is made of, it is that it runs apart and nothing can bypass it. The model expresses all three at design time, and at runtime it is not in the loop. That is the arrangement that puts the most capable actors ever built onto consequential work.

THE COMPILER

Inside the compilation

Nothing mystical happens in there, and the mechanism is worth knowing, because it is the reason this became possible only now.

The person's words and the runtime grammar both sit in context. The model has seen the informal-to-formal mapping a million times in protocols, procedures, safety cases and specifications. Attention binds the scattered references into one declaration. The filled triad is simply the likeliest continuation, so no reasoning step is required.



The governor is the part that takes the time

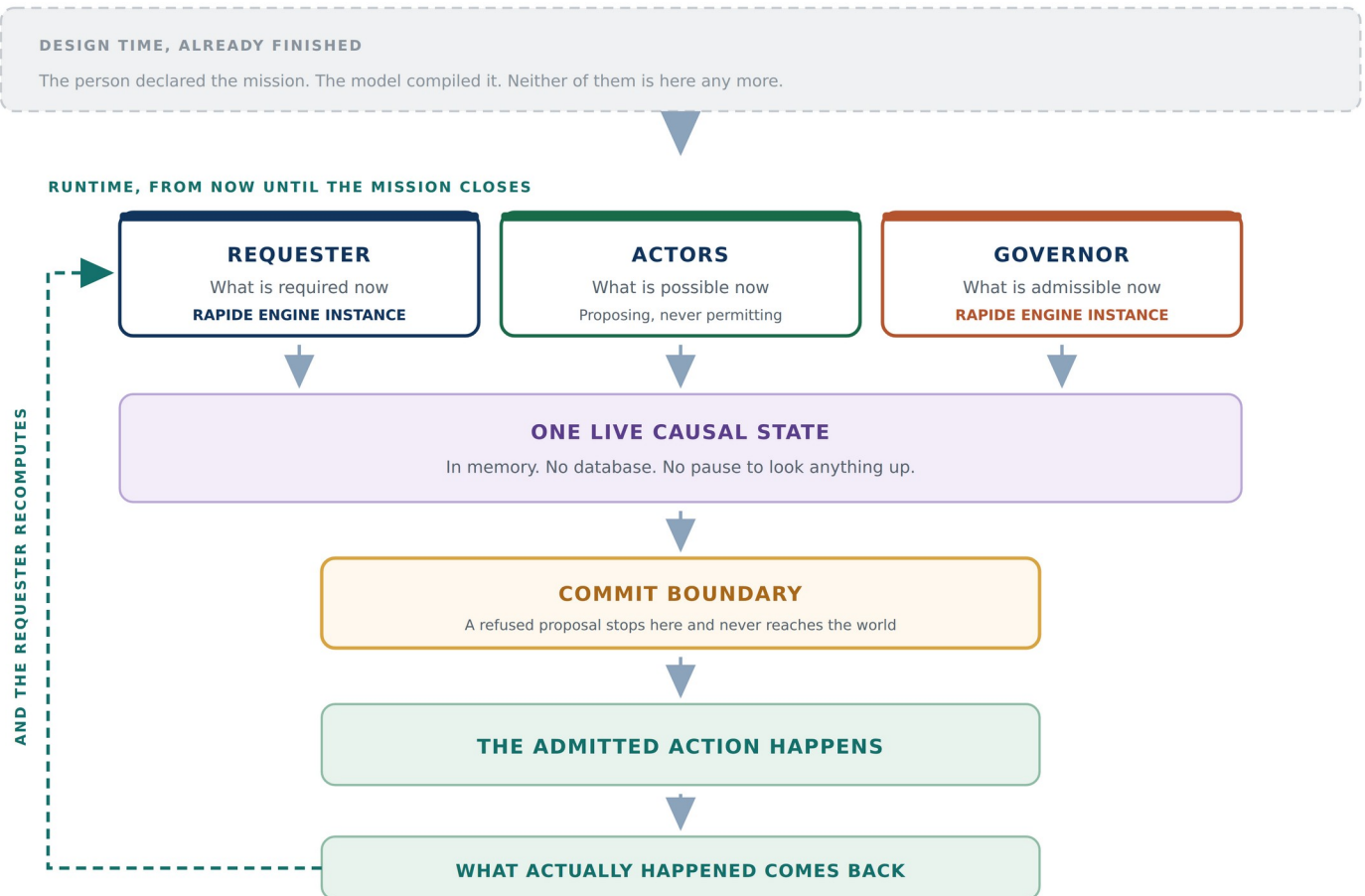
Adapters are interface work and come out almost free. The Requester needs the outcome decomposed. The Governor is hardest, and not for technical reasons: people do not state their limits unprompted, so it has to be drawn out by questioning. That is an interview, which is exactly what this kind of model is best at, and it is the reason no systems analyst is needed in the middle.

What happens after the model has gone

Everything up to this point is design time. The person declared the mission, the model compiled it, and both are now finished. What follows runs on its own, for as long as the mission exists.

The Requester and the Governor run as Rapide engine instances. The engines are the patented runtime and are the same every time. What the compilation produced is their configuration, not new code, so a new mission is a new instance rather than a new build.

Neither waits to be asked. The Requester recomputes on every change in reality, and the Governor determines each proposal against the world at that moment rather than against a rule written earlier.



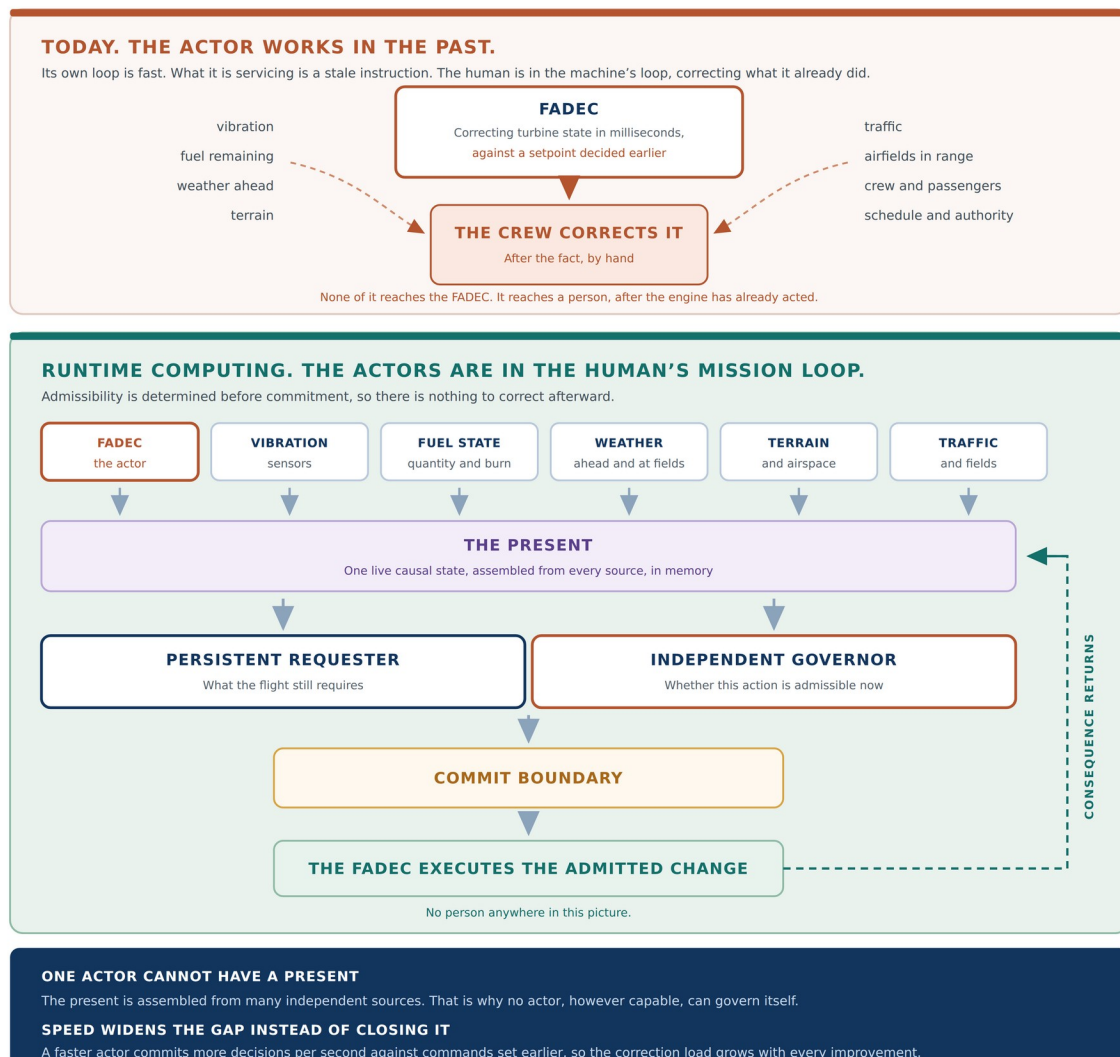
Why it has to be in memory

A governor that pauses to query a database is reading a report of the world rather than the world, and by the time the answer returns the thing it described has moved. The causal state is held live, in memory, so admissibility is computed against what is true now.

An actor computational system is not a mission computational system

An actor computational system works in the past. A FADEC corrects turbine state in milliseconds, against a setpoint decided earlier, servicing a stale instruction with great precision. It functions at all because a person stands inside the machine loop supplying the present. The design assumed that person would be there.

A mission computational system inverts it. The Requester and the Governor control the present, and the actors sit inside the human's mission loop. The actor becomes one event source among many, and admissibility is determined before commitment, so there is nothing to correct afterward. It is the number of independent sources that makes a present exist at all, which is why no actor can have one of its own and none can govern itself.



TWO KINDS OF SYSTEM

The same demotion, in five places

Every domain was built to the same pattern: one actor at the center, and the whole present left to a person to assemble. An architectural decision, taken over and over, that people have been covering for ever since.

On the right is what the present is made of in each case. The actor's own telemetry sits in every one of those lists, unmarked and in no particular position, because it has no standing the others do not.

	THE ACTOR WORKS IN THE PAST THE HUMAN IS IN THE MACHINE'S LOOP, CORRECTING IT	THE ACTORS ARE IN THE HUMAN'S MISSION LOOP REQUESTER AND GOVERNOR CONTROL THE PRESENT. NOTHING TO CORRECT.
AIRCRAFT ENGINE	FADEC corrects turbine state in milliseconds Against a setpoint and envelope decided earlier. THE CREW CORRECTS WHAT IT ALREADY DID after the vibration, after the fuel burn, after the weather.	vibration · fuel quantity and burn · FADEC telemetry weather ahead · terrain · airspace · traffic airfields in range · crew state · schedule · authority
AIRLINER	Autopilot flies a clearance issued earlier Conditions have changed since it was issued. THE CAPTAIN CORRECTS IT once the aircraft is already heading the wrong way.	weather · fuel · autopilot state · traffic · terrain airspace · airfield status · crew duty limits passengers · dispatch release · declared authority
WAREHOUSE	Pick robot works a list built earlier The line went down after the list was built. THE SUPERVISOR CORRECTS IT once the wrong items are already picked.	line status · truck arrival · robot and conveyor state dock capacity · order book · customer priority inventory · labor on shift · declared authority
HOSPITAL	Pump runs an order written earlier The patient has changed since it was written. THE CHARGE NURSE CORRECTS IT once some of the dose has already gone in.	vitals · laboratory results · pump state · imaging allergies · medication history · interactions · fluids staffing · bed and theatre state · clinician authority
TRADING	Algorithm works an order sent earlier The book has moved since it was sent. THE RISK OFFICER CORRECTS IT once the position is already on.	position book · net exposure · order and fill state collateral · market conditions · counterparty limits mandate · other desks · declared authority

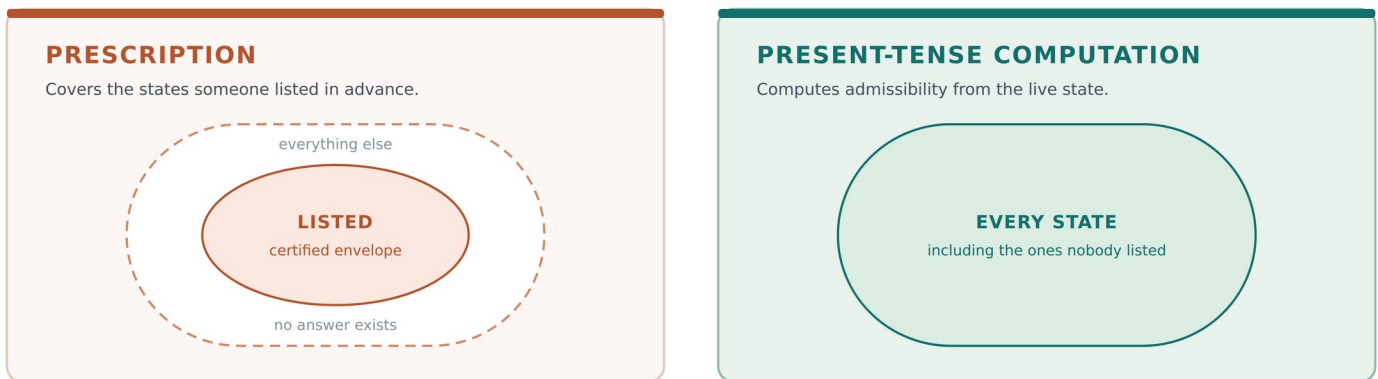
Why separation is the mechanism, not a precaution

A body works this way. Nothing in it is centrally commanded. Many independent signals build the present, something computes what is required, something else admits or refuses, and repair runs continuously without anyone deciding. Separation is not added for safety. It is the reason the thing functions at all.

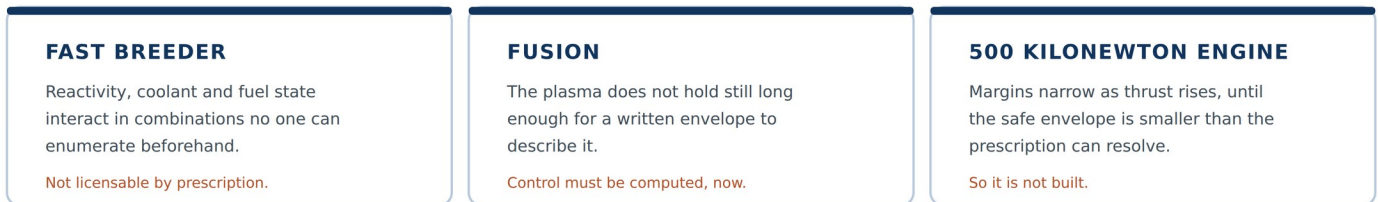
What becomes buildable

Prescription covers the states somebody listed in advance. Outside that list there is no answer, which is why a person has to be standing there. Present-tense computation has no list. It computes admissibility from the live causal state, so it covers every state, including the ones nobody wrote down.

That is not a refinement of control. It is the first time control has existed.



MACHINES WHOSE ENVELOPE CANNOT BE LISTED IN ADVANCE



THESE ARE NOT MATERIALS PROBLEMS

They are control problems, and the only control ever available was a list written in advance with a person watching it.

CONTROL IS THE PRESENT

Compute admissibility against the live causal state and the envelope is no longer something anyone has to write down first.

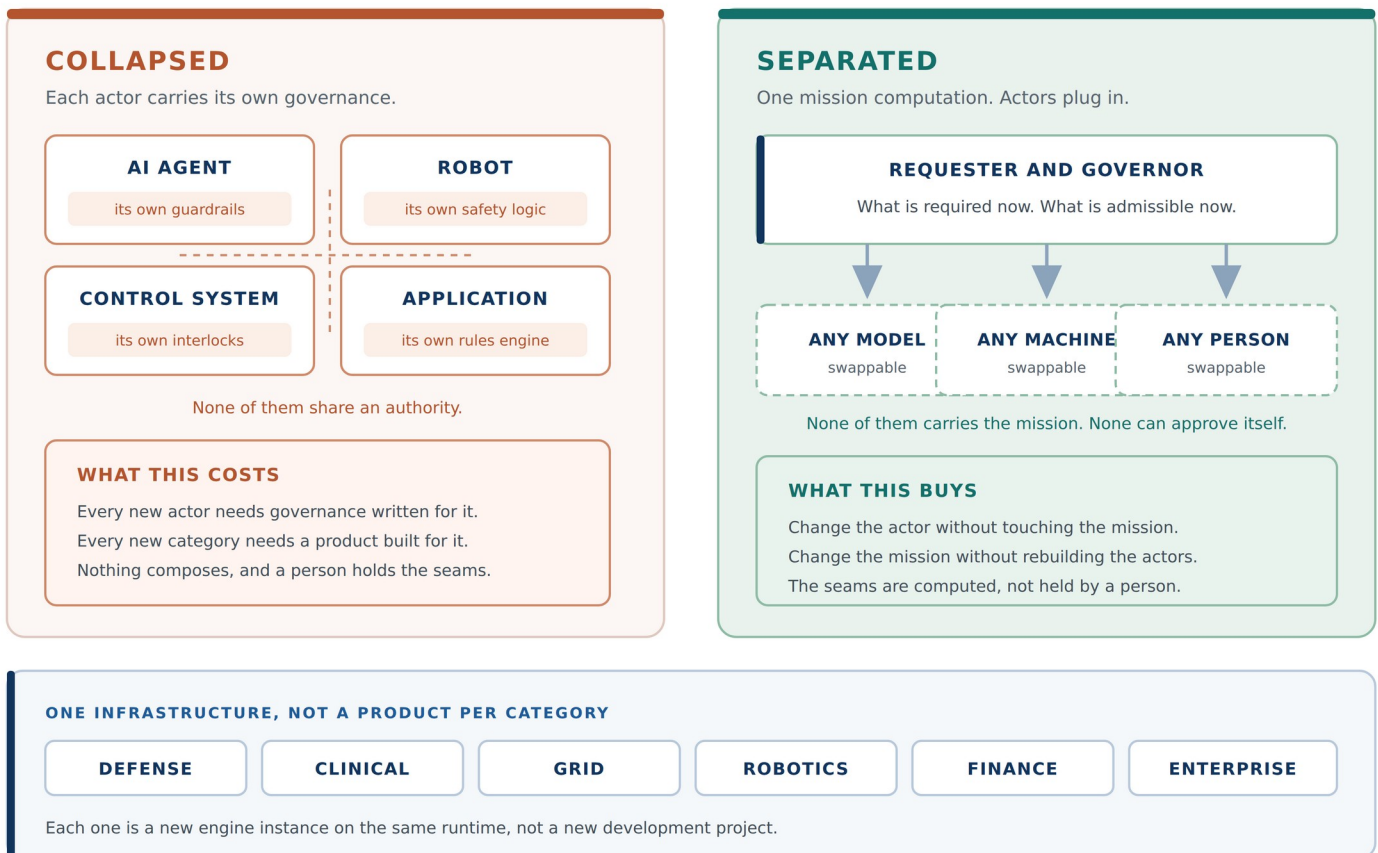
The machines that were never built

Fast breeder reactors, fusion plants and 500 kilonewton engines have been treated as materials problems. They are control problems. Each one has an operating envelope that cannot be enumerated in advance, and prescription with a person watching it was the only control available, so they were not built. With the envelope computed in the present, the limit is gone.

Separate them and almost anything can be built

This is the engineering consequence, and it is the reason one company can address defense and clinical and grid without building three products.

Once the mission computation sits outside the actors, the actors become swappable. Any model, any machine, any person, in any combination, because none of them carries the mission and none of them can approve itself.



Collapse them and you get the opposite

Every actor needs governance written for it, every category needs a product built for it, and a person holds the seams between them. That is why nothing composes today. It is not a tooling problem. It is what happens when the computation that should sit above the actors is distributed into each of them.

Work replaced, and work that was never possible

Around ten trillion dollars a year of paid work is people deciding what is required and what is permitted rather than doing the thing. That is a wage bill, not a software budget.

Declaring a mission was never a technical skill, only an unwritable one. A charge nurse, a farmer, a harbour master can each state theirs exactly in five minutes, and with a frontier model compiling it, no engineer stands in between. Beyond both sits everything that was never attempted, because the only available control was a person.

WORK DONE BY HAND TODAY	WHO CAN DECLARE A MISSION	WHAT WAS NEVER ATTEMPTED
\$10 trillion a year, worldwide Global labour compensation is about \$55 trillion. Roughly a fifth of paid time goes on deciding, not doing. Who is doing it - Charge nurses and clinical leads - Shift and site supervisors - Grid and plant operators - Dispatchers and controllers - Risk and compliance officers - Mission and flight operations - Schedulers and planners Not a software budget. A wage bill. Computed instead of carried.	Anyone who holds one Declaring a mission is not a technical skill. The people who hold missions can already state them exactly, in plain words. A frontier model compiles it from there. Stated in five minutes, by - A charge nurse - A farmer - A site foreman - A harbour master - A school principal - A fleet manager - A city engineer No engineer stands between the person who knows the mission and the machine that runs it.	Work no one could supervise THE HUMAN WAS THE RATE LIMIT Swarms. Large autonomous fleets. Millions of distributed units. Not blocked by the machines. Blocked by how many permissions a person can issue. NO PERSON CAN BE PRESENT Reactor interiors. Deep sea. Orbit. Contaminated ground. The first hours of a disaster. Supervision needs presence. SUPERVISION COSTS TOO MUCH Any small autonomous unit that needs a person watching it stays uneconomic, however cheap the unit becomes. Remove the watcher and the category opens.

ONE REPLACES WORK BEING DONE. THE OTHER CREATES WORK THAT WAS NEVER POSSIBLE.

Both come from the same change: the mission is computed instead of carried.

The larger column is the one with no number

Substitution is measurable because the work is already being paid for. The projects that were never attempted have no line in anyone's accounts, because nobody ever proposed them.

THE POINT

The separation is the whole thing

Not a constraint added to intelligence. Not a rule layer bolted onto a capable actor. The separation is what makes any of it safe to run, and it is as true of a person as it is of a system.

Nobody who proposes an action also approves it.

The consequence returns to what was required, not to whoever acted.

Runtime Computing is that, built so it cannot be collapsed.

Three independent computations against live reality. A frontier model to express them in a person's own words. An enforced boundary that only an admitted action crosses, and a consequence that returns to the same mission.

Nothing here is a criticism of the people doing it. They are performing the hardest computation in any operation, continuously, with no instrument built for it. The fault is in a technology that computed the actor and left the mission to them.

For five thousand years every tool required the person to learn the machine's language. Mathematics, notation, programming, every one of them.

The frontier model is the first thing we have built that learns ours.

Everything a person knew but could not write down has been locked outside the machine until now.

Runtime Computing is what happens on the other side of that door. What they say is compiled, and then it runs, governed, without them in the loop.

Dream the triad with a frontier model. Run it on our engines.

Say what you need in your own words: the outcome required, the authority it runs under, the limits. Any frontier model compiles that into the three computations.

The Requester and the Governor run as Rapide engines. Your actors stay exactly as they are, and become one event source among many.

Every mission anyone has ever held by hand is available to be computed. Start with one.

decision-zone.com · rb@decision-zone.com

ANY INTELLIGENCE. ANY ACTORS. ONE GOVERNED REALITY.