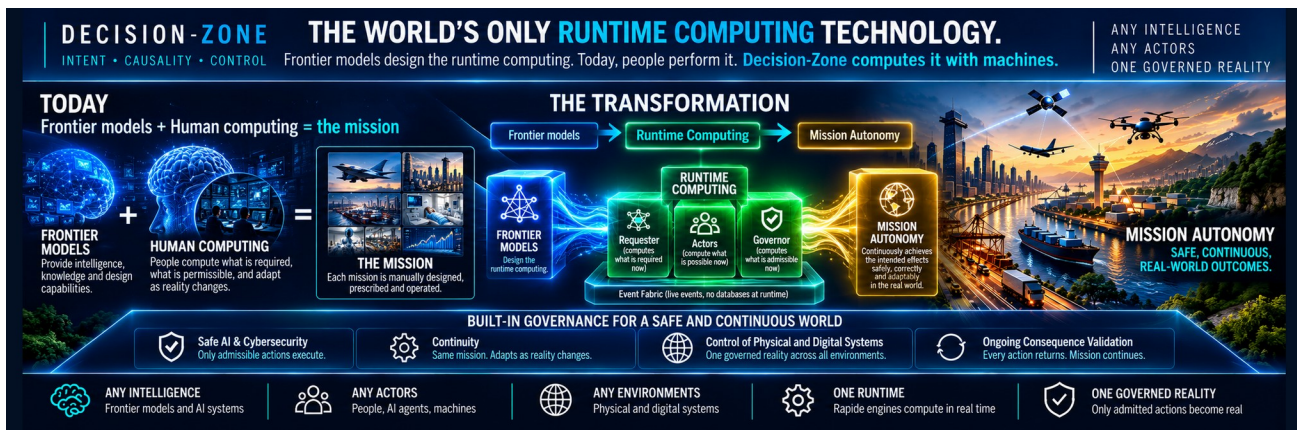




HUMAN CENTRICITY

From a human inside the machine's loop to the machine inside the human's mission.

Every other autonomy vendor is asking permission to remove the human. Runtime computing puts the human in charge of the loop instead of inside it.

THE WORLD RUNS MACHINE CENTRIC

Machine centricity is not a figure of speech. It is how every deployed system is actually organized. Devices, applications and models each hold their own state and report it outward. Nothing in the architecture holds the purpose they are serving. That purpose is held by people, in two separate layers, and both of them are manual.

Two layers of people, not one

The first layer writes the system down in advance. Programmers, now assisted by AI coding tools, encode the behavior, the logic, the rules and the exceptions for every situation they can imagine. What comes out is prescribed logic, hard coded workflows, predefined constraints and manual updates, and it takes a long time to implement. A coding assistant makes that layer faster. Faster prescription is still prescription.

The second layer supplies everything the prescription could not anticipate. Operators observe, decide, intervene and recover, continuously, for as long as the system runs. That is runtime computing, and it is the only part of a modern system still performed by hand.

Every dashboard is an invoice

A dashboard exists for one reason: nothing in the system computes what matters, so raw state is presented to a human who must synthesize it. Every dashboard and every alarm list is an invoice for missing runtime computing, and the industry ships them as features.

Human in the loop is not human in charge

All of this is sold as human oversight. Human in the loop. Human-centered AI. It is the opposite of what those words describe. The human is in the loop because the loop has a hole in it, and the person is the only component that can fill it. Being load-bearing is not the same as being in charge.

The inversion

A Human Mission Owner declares the mission: the required effect, the authority, the invariants and the constraints. A frontier model compiles that declaration into an executable TCP package, which answers the first layer, and no coding is required. The Rapide engines then compute the mission continuously against live causal state, computing what is required now, what is possible now and what is admissible now, which answers the second. Only an admitted action crosses the Commit Boundary into reality, and the observed consequence returns to the same mission so the requirement is computed again. Neither replacement is a person. The person who remains is the one who says what it is all for.

Authority, not interface

Human centricity in this sense is a statement about authority, not about screens. No Actor grants itself permission. No model supplies the Governor's judgment. Nothing becomes real without passing a boundary a person declared, and every refusal names the authority it was refused under.

We are not automating people out. We are taking them out of the loop and putting them in charge of it.

Decision-Zone is the only architecture that requires a human to declare what the system is for, and refuses to act without that declaration.

FROM A HUMAN IN THE MACHINE'S LOOP TO THE MACHINE IN THE HUMAN'S MISSION

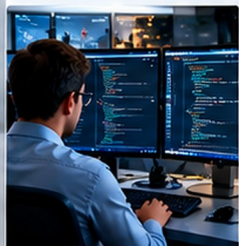
INTENT
CAUSALITY
CONTROL

TODAY THE HUMAN IS INSIDE THE MACHINE'S LOOP

Programmed, prescribed and manually operated

HUMAN PROGRAMMERS AND AI CODING ASSISTANTS

Design and code prescribed behavior, logic, rules and exceptions



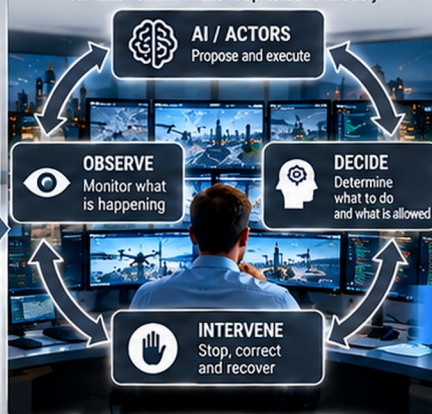
Software System (Actor-Centric)

- Prescribed logic
- Hard coded workflows
- APIs and integrations
- Predefined constraints
- Trained AI models
- Manual updates
- Long implementation time



HUMAN RUNTIME OPERATIONS

Humans remain in the loop to continuously:



SIX COMPUTATIONS HUMANS DO TODAY (MANUAL AND CONTINUOUS)

- | | | |
|--|--|--|
| 1 CONTEXT
Understand what is happening now | 2 REQUIREMENT
Determine what the mission requires now and what remains | 3 POSSIBILITIES
Identify and evaluate possible actions |
| 4 ADMISSIBILITY
Decide what is allowed under authority, constraints and risk | 5 COMMITMENT
Authorize and initiate action | 6 CONSEQUENCE
Verify what happened and determine what is needed next |

CURRENT THE MACHINE IS INSIDE THE HUMAN'S MISSION

The human declares the mission. An LLM compiles the executable TCP. The machine computes it continuously.

HUMAN MISSION OWNER



Declares the mission, objectives, authority, invariants and constraints

FRONTIER MODEL COMPILER (LLM)

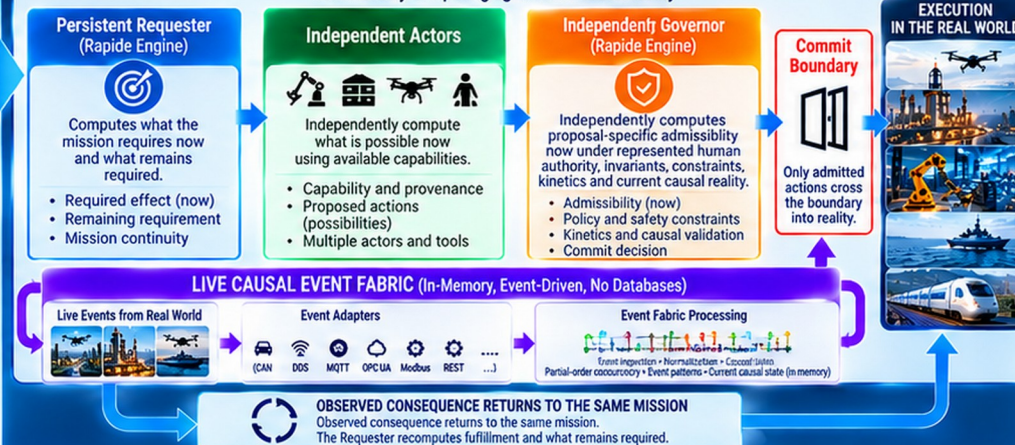


Compiles mission specification into an executable TCP package

- Mission specification
- Authority and invariants
- Constraints and kinetics
- Actor interfaces
- Event fabric design
- Rapide engine configuration

TRIADIC RUNTIME COMPUTATION

Continuously computing against live causal reality



Today the human is inside the machine's loop.

Humans program the system and continuously compute mission decisions during operations.



With runtime computing, the machine is inside the human's mission.

The human declares the mission. An LLM compiles the TCP. The machine continuously computes what is required, what is possible and what is admissible – and executes only the correct actions.



- ✓ Purposeful action
- ✓ Admissible execution
- ✓ Continuity through change
- ✓ Recovery and adaptation
- ✓ Safe, sovereign and controlled

POSITIONING HIERARCHY

One Core Technology. Many Applications. A Safer, More Controlled Tomorrow.

INTENT
CAUSALITY
CONTROL

1. DESIGN-TIME INTELLIGENCE

Frontier models as the intelligent GUI and compiler



- Define the mission in natural language
- Specify desired outcomes
- Set authority, invariants and constraints
- Identify actors, tools and data sources
- Compile to an executable TCP package

Executable TCP Package



- Mission specification
- Authority and invariants
- Constraints and kinetics
- Actor interfaces
- Event fabric design
- Rapide engine configuration

2. TRIADIC RUNTIME COMPUTATION

Independent Rapide engines computing continuously against live causal reality

Persistent Requester (Rapide Engine)



Computes what the mission requires now and what remains required.

- Required effect (now)
- Remaining requirement
- Mission continuity

Independent Actors



Independently compute what is possible now using available capabilities.

- Capability and provenance
- Proposed actions (possibilities)
- Multiple actors and tools

Independent Governor (Rapide Engine)



Independently computes proposal-specific admissibility now under represented human authority, invariants, constraints, kinetics and current causal reality.

- Admissibility (now)
- Policy and safety constraints
- Kinetics and causal validation
- Commit decision

Commit Boundary



Only admitted actions cross the boundary into reality.

3. GOVERNED EXECUTION IN THE REAL WORLD

Digital and physical systems

- APIs & Software
- Robotics
- Industrial Systems
- Vehicles & Drones
- Financial Systems
- Scientific Instruments
- People & Organizations



LIVE CAUSAL EVENT FABRIC (In-Memory, Event-Driven, No Databases)

Live Events from Real World



Event Adapters (JCA Providers)



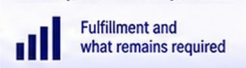
Event Fabric Processing



Observed Consequence



Requester Recomputes



4. APPLICATION DOMAINS

Different industries. The same runtime computational architecture.

CYBERSECURITY
From detection to governed remediation



- Autonomous threat detection
- Governed patch application
- Containment and recovery
- Reduced human decision delay

INDUSTRIAL / OT
Safe operation of physical systems.



- Real-time operational decisions
- Safety and process constraints
- Scheduled and conditional actions
- Continuous mission operation

HEALTHCARE
Safe, compliant and continuous care.



- Clinical decision support
- Patient safety and compliance
- Real-time monitoring
- Continuity through change

FINANCE
Trusted and compliant transactions.



- Transaction monitoring
- Fraud prevention
- Risk and regulatory compliance
- Governed autonomous actions

DEFENSE & NATIONAL SECURITY
Mission execution with human authority.



- Autonomous systems
- Mission coordination
- Deconfliction and safety
- Trusted and auditable control

SCIENCE & RESEARCH
Autonomous discovery and experimentation



- Experiment planning and control
- Instrument coordination
- Safety and constraint enforcement
- Continuity through outcomes

TODAY: ACTORS WITHOUT REQUESTER AND GOVERNOR

Intelligence without mission computation.



Can reason, plan, propose and execute

Activity without independent mission computation

Actors become zombies just like humans.

TRANSFORMATION: TRIADIC RUNTIME COMPUTATION

Mission computation gives purpose, permission and continuity.

Requester (Requirement now)

Actors (Possibility now)

Governor (Admissibility now)

Commit → Reality → Consequence → Recompute

RESULT: MISSION AUTONOMY

Continuous, governed, real-world execution.

- ✓ Purposeful action
- ✓ Admissible execution
- ✓ Continuity through change
- ✓ Recovery and adaptation
- ✓ Safe, sovereign and controlled



U.S. Patent 7,908,160 | Runtime Computing (Method and System) — Patent Pending

SOFTWARE EXECUTES WHAT WAS DECIDED EARLIER.
DECISION-ZONE COMPUTES WHAT IS REQUIRED NOW.

ANY INTELLIGENCE. ANY ACTORS. ONE GOVERNED REALITY.