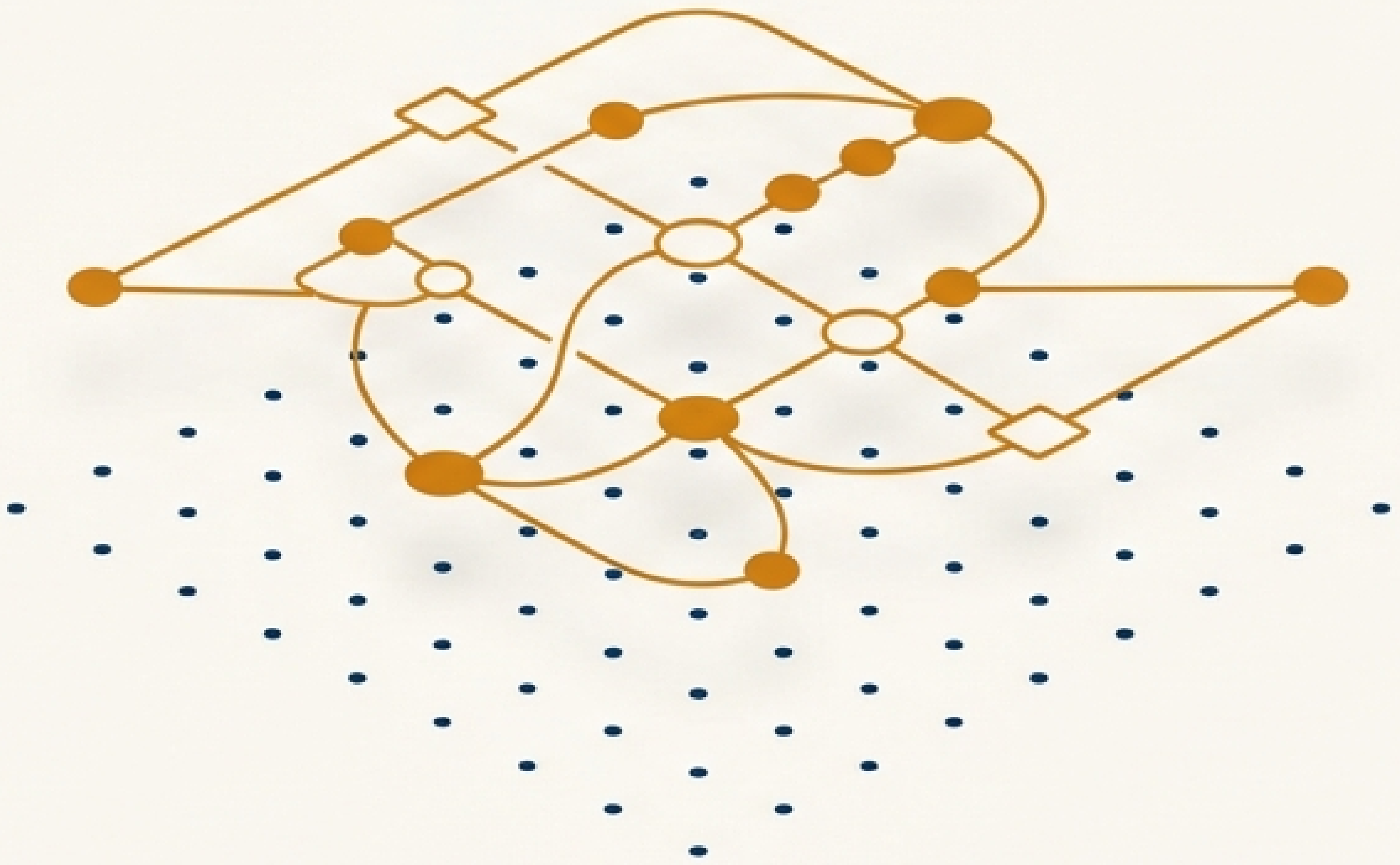
Beyond Static Models: Controlling Complex and Dynamic Processes with Meta-Petri-Nets

A visual exploration of a powerful modeling paradigm for reconfigurable systems



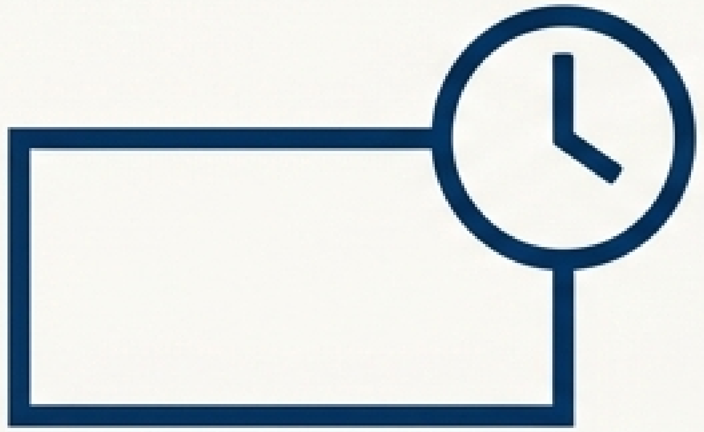
Based on the work of Vagan Terziyan (Kharkov State Technical University of Radioelectronics) and Vesa Savolainen (University of Jyvaskyla).

Petri Nets are a powerful tool for modeling complex processes.

Since their introduction, Petri nets have been a cornerstone for modeling, simulating, and controlling complex flows. Numerous powerful extensions have increased their flexibility and modeling power.

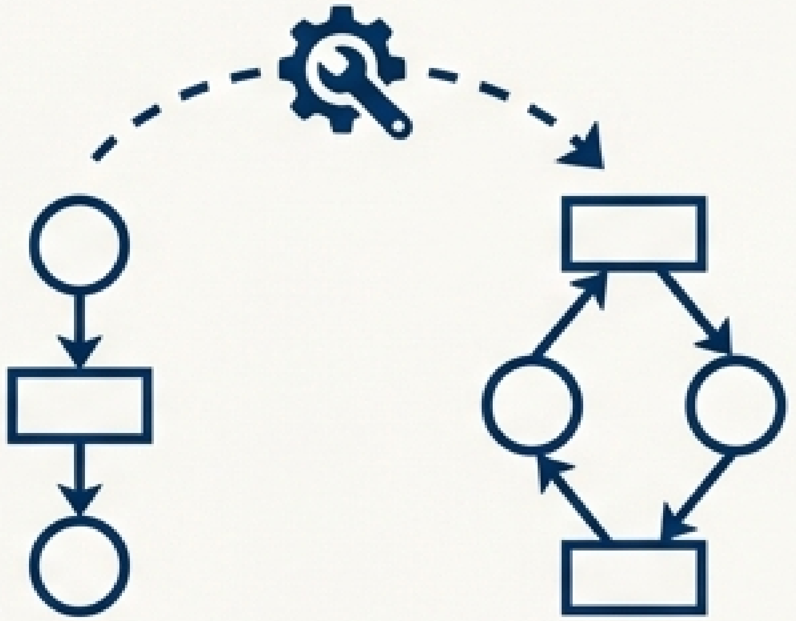
Colored Nets

Tokens have values or “colors,” enabling concise representation of systems with data and resources.



Timed / Time Nets

Introduce time delays (at places or transitions) to solve real-time problems.

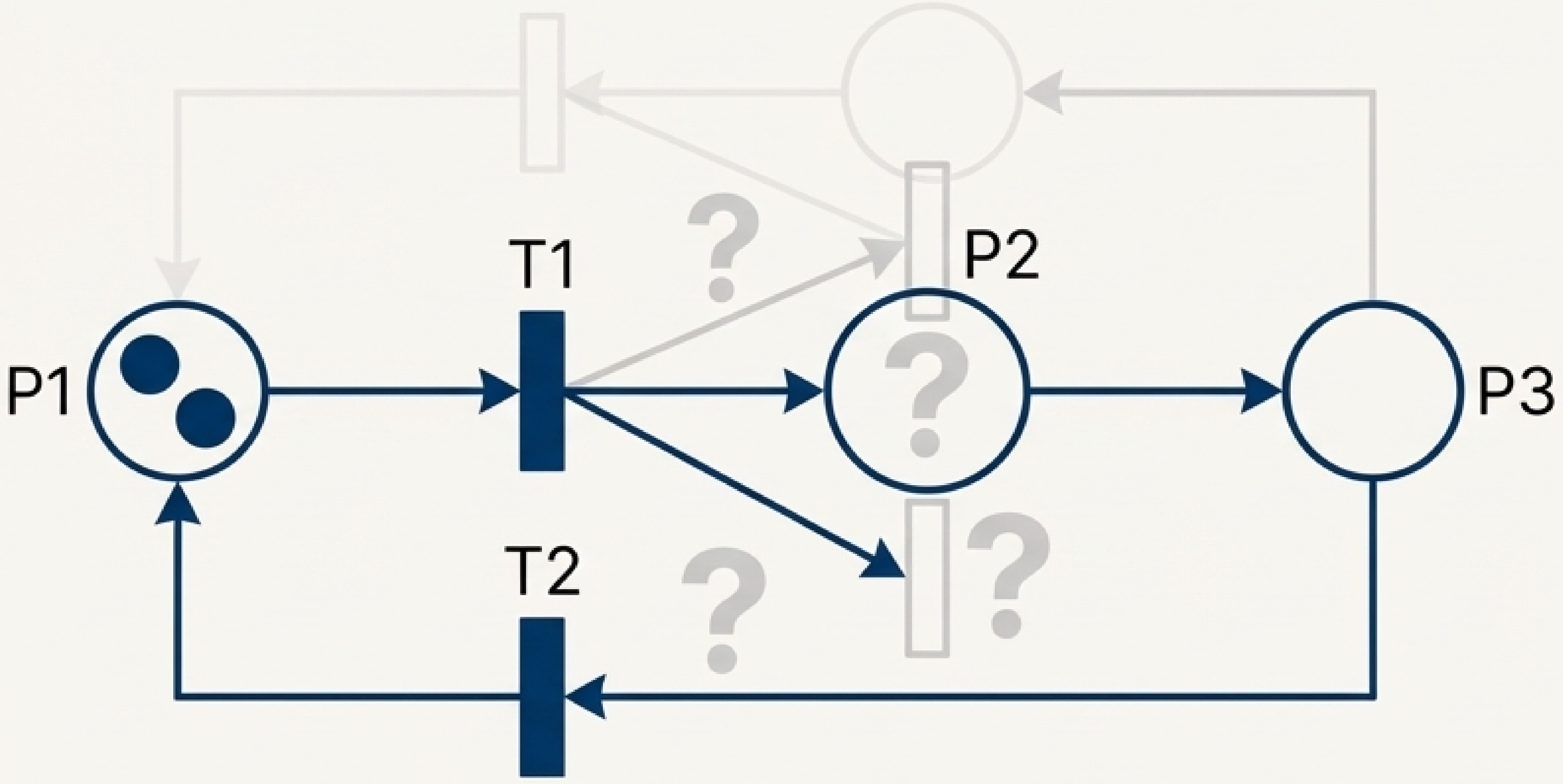


Reconfigured Structures

Allow for logical reconfiguration to handle faults or evaluate system performance under different conditions.

But what happens when the rules of the system itself must change?

Traditional extensions excel at modeling systems with fixed structures. However, they face challenges in elegantly representing systems where the process itself is dynamic. Powerful tools for flexible restructuring have been missing.

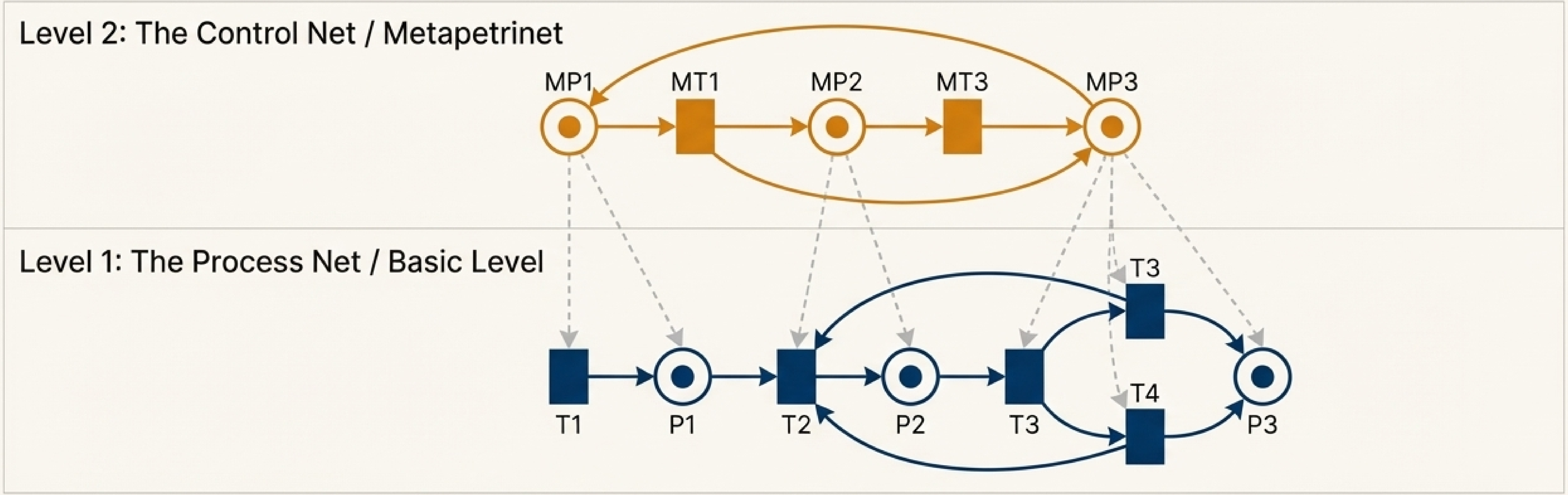


How do we model systems that require dynamic reconfiguration, such as:

- Changing-routed jobshops in manufacturing?
- On-the-fly traffic flow management?
- Fault-tolerant systems that must logically reconfigure?

The Solution: Add a New Dimension of Control with a Meta-Level

A Meta-Petri-Net (MPN) is a multi-level model where a higher-level net is used to dynamically reconfigure the structure of a lower-level net.

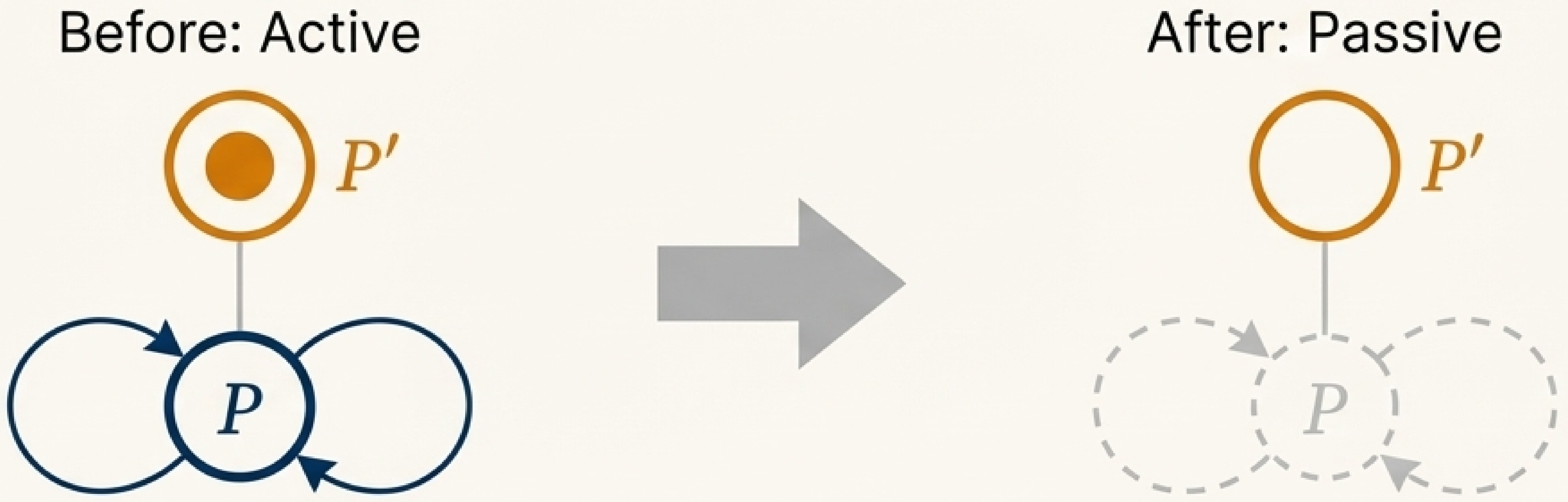


"A metapetrinet is able not only to change the marking of a petrinet but also to reconfigure dynamically its structure."

The Fundamental Rule of Control

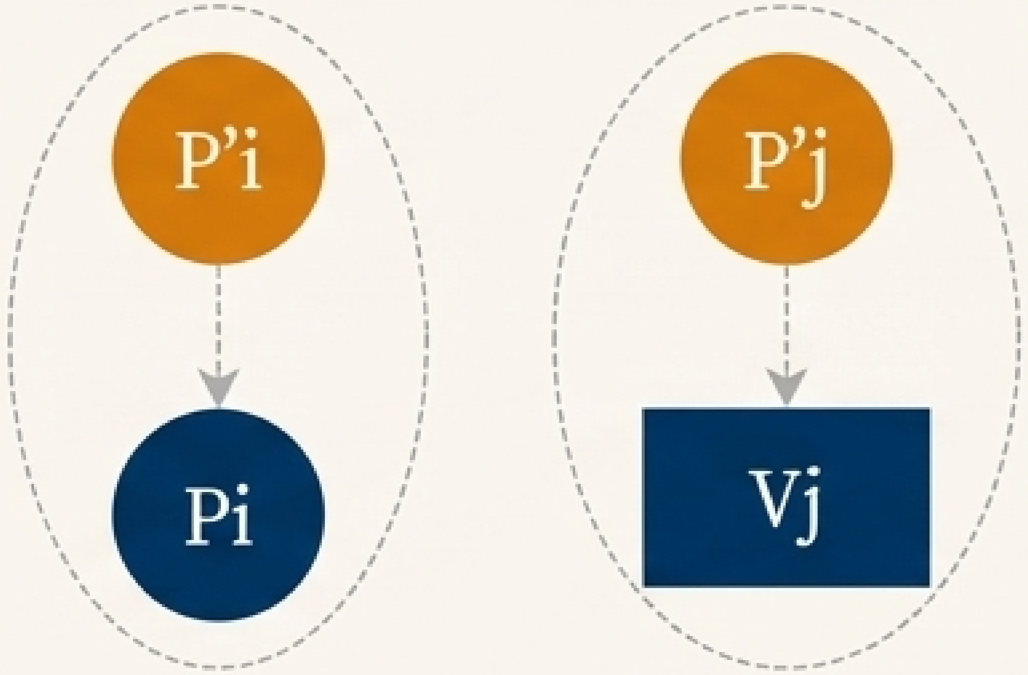
A node in the basic net is active if and only if its corresponding metaplace contains a token.

An empty metaplace renders its corresponding node (and its connecting arcs) passive, temporarily removing it from the net's configuration. When a token is added to the metaplace, the node is immediately restored.



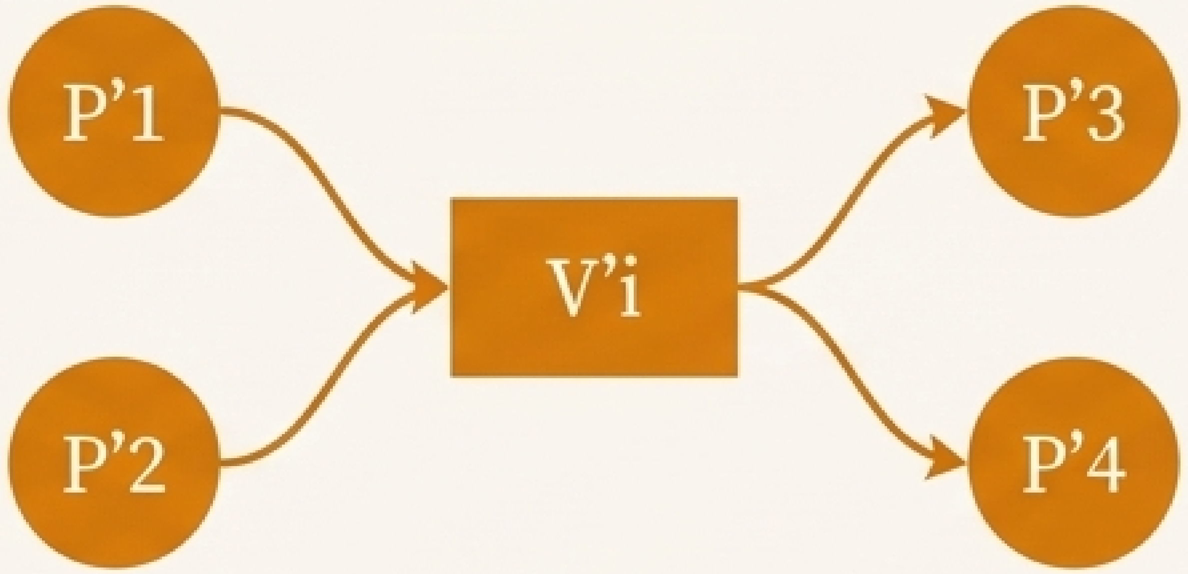
The Building Blocks of the Meta-Level

Metaplaces



Metaplaces are nodes in the control net. Each can be assigned to a specific place or transition in the basic net to govern its active state.

Metatransitions



Metatransitions function like ordinary transitions but operate on the meta-level. Their firing changes the marking of metaplaces, thereby triggering reconfigurations in the basic net below.

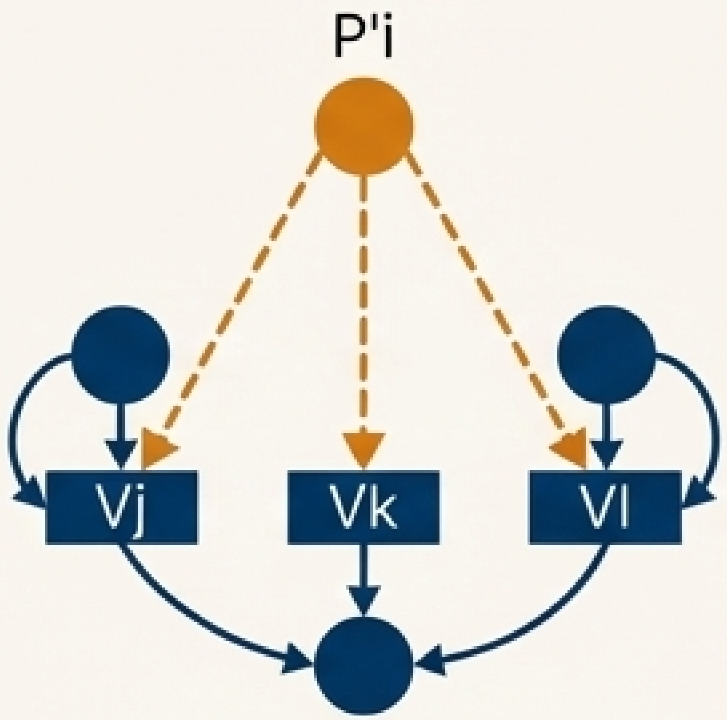
A Formal Definition for a Two-Level Metapetrinet

A two-level MPN is defined by the following vector, extending the classical Petri net definition with components for the meta-level and its interaction with the basic level.

$$N = \langle P, V, F, H, M_0, P', V', F', H', M_0', R', Q', k' \rangle$$

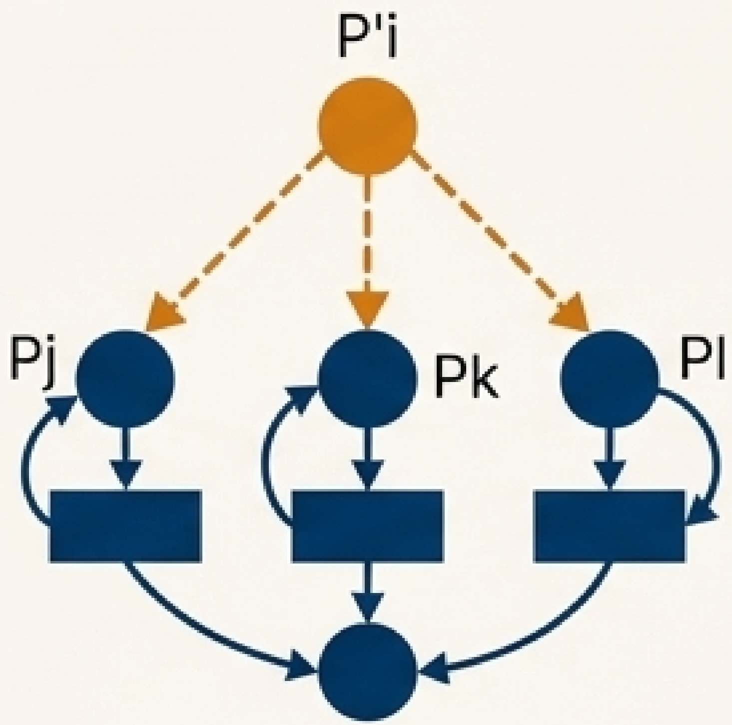
Component	Description
P', V', F', H', M_0'	The places, transitions, incidence functions, and initial marking of the **meta-level net**.
R'	Incidence function mapping **metaplaces to places** in the basic net.
Q'	Incidence function mapping **metaplaces to transitions** in the basic net.
k'	Defines the time delay of metatransitions relative to the basic net's "beat" ($\tau' = k' \cdot \tau$).

Three Primary Configurations for Metanet Control



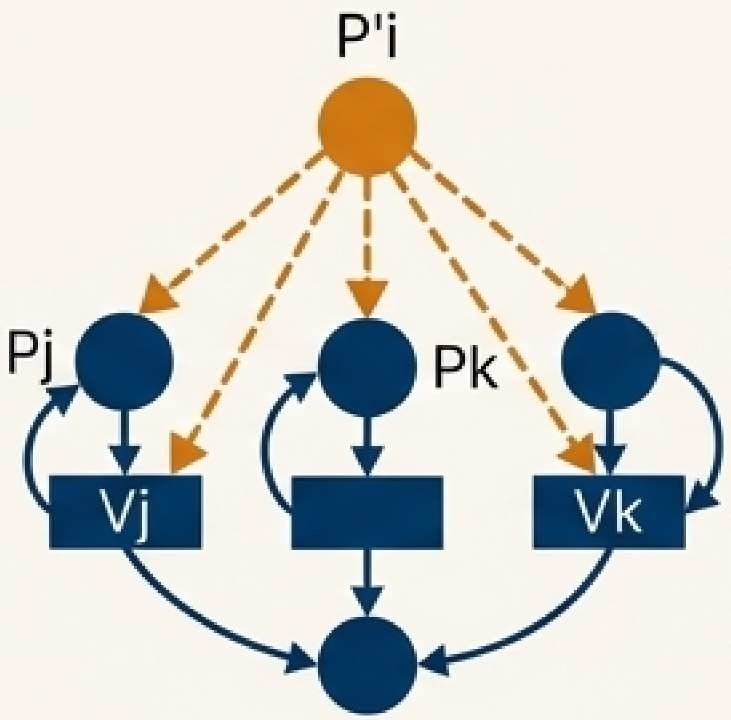
Transitional MPN

Metaplaces correspond **only to transitions** of the basic level. The meta-level controls which processes or actions are available. available.



Positional MPN

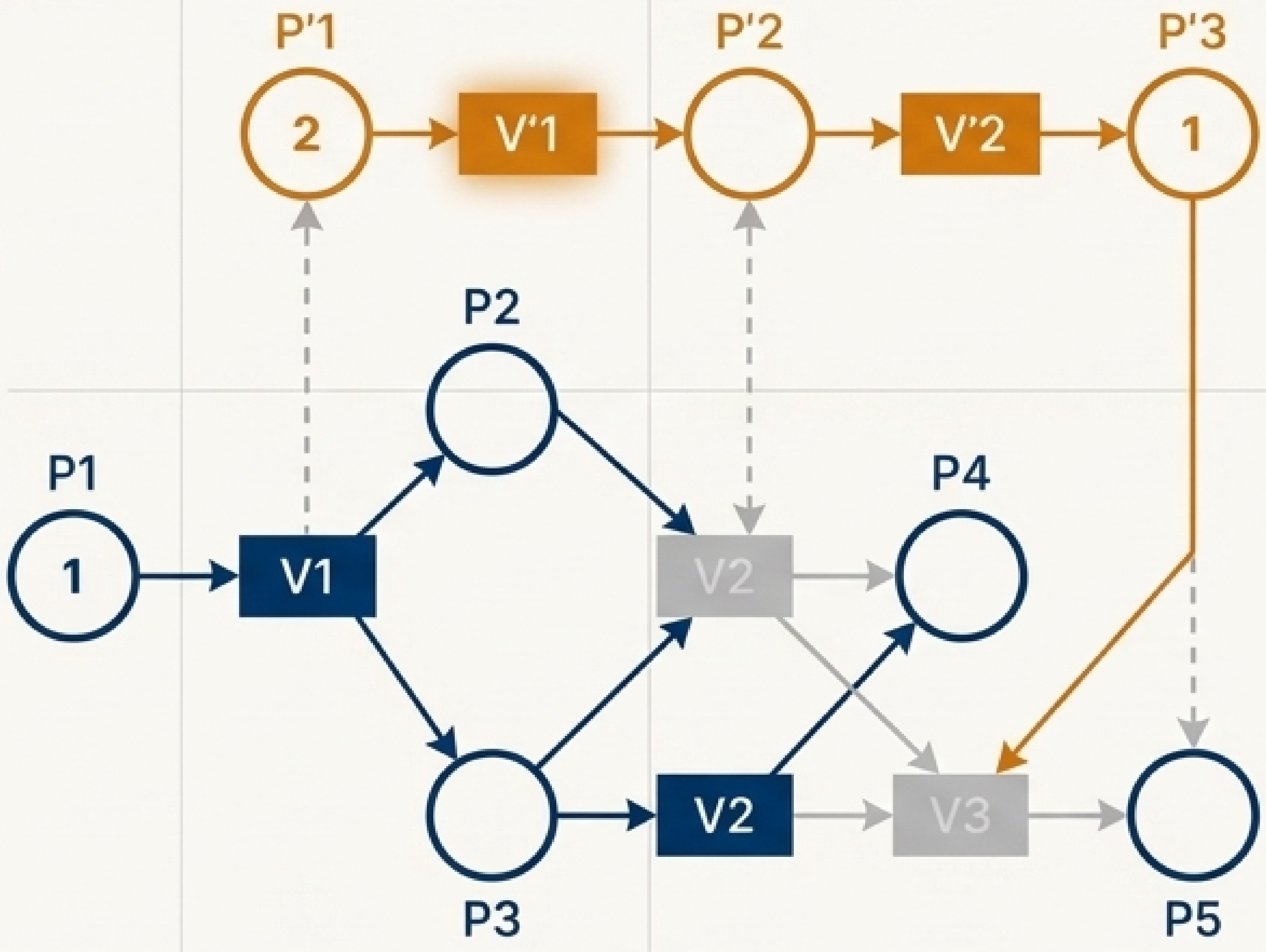
Metaplaces correspond **only to places** of the basic level. The meta-level controls which states or resource locations are active. The marking of a deactivated place is preserved.



Mixed MPN

Metaplaces can correspond to **both places and transitions,** allowing for complex, hybrid control strategies.

Example in Practice: A Transitional MPN



Execution Trace

Step 0: Initial State

- Metaplace marking is (2; 0; 1).
- Transitions V1 and V3 are active. V2 is inactive because P'2 is empty.

Steps 1-2: Basic Net Execution

The active basic net executes. V1 and V3 fire.

Step 5: Reconfiguration Event

Metatransition V'1 fires. Metaplace marking changes to (1; 1; 0).

Effect: Structure Changes

V3 becomes inactive. V2 becomes active. The basic net's available pathways have been reconfigured.

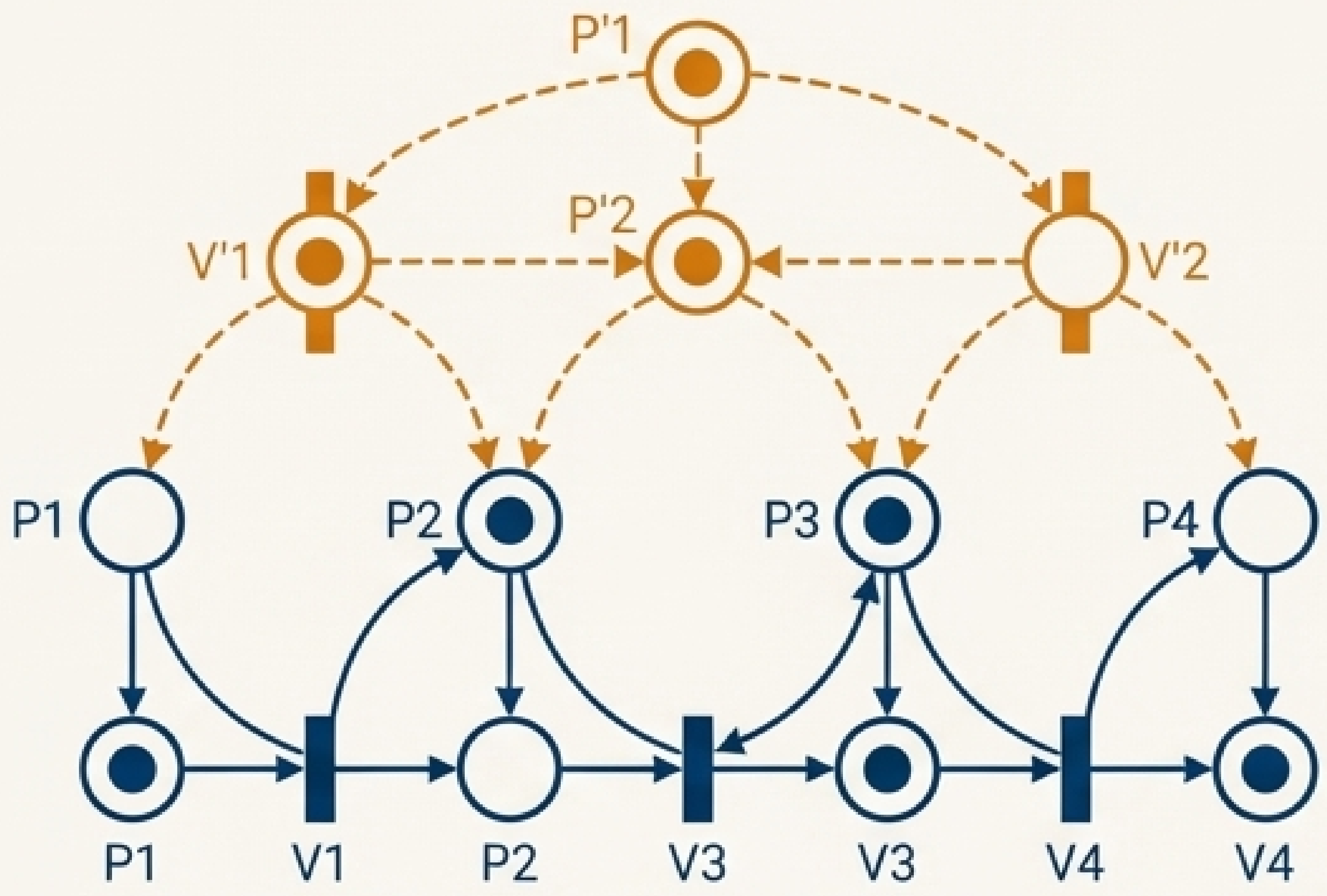
Steps 6–7: New Configuration Executes

The basic net executes with the new structure. V2 and V1 fire.

The Result: Elegance vs. Complexity

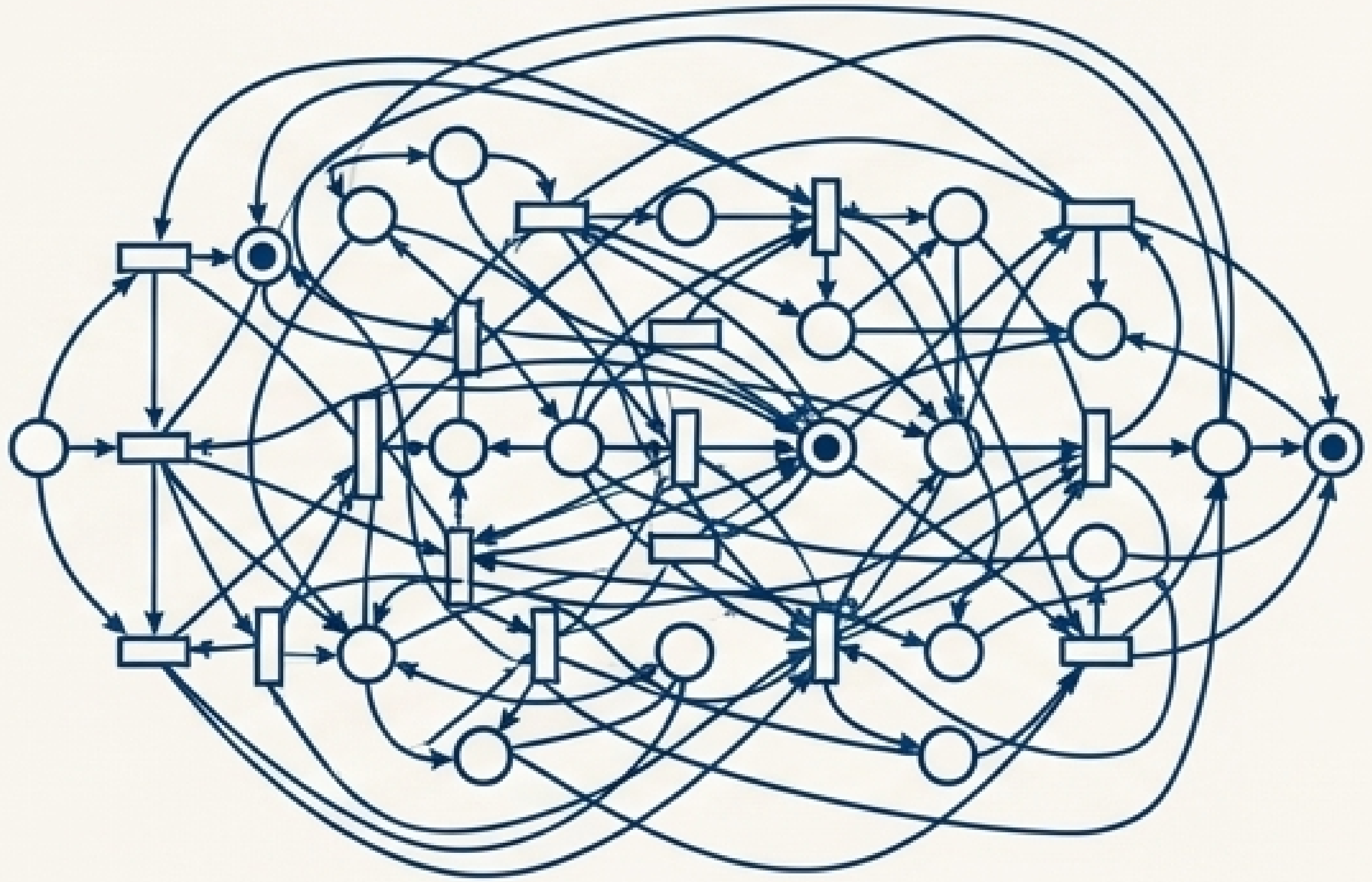
Simulating the dynamic process of the Positional MPN example using an ordinary, single-level Petri net reveals the significant advantage of the meta-level approach.

Positional Metapetrinet



Adds **3 metaplaces** and **2 metatransitions** to the controlling metalevel.

Equivalent Ordinary Petrinet

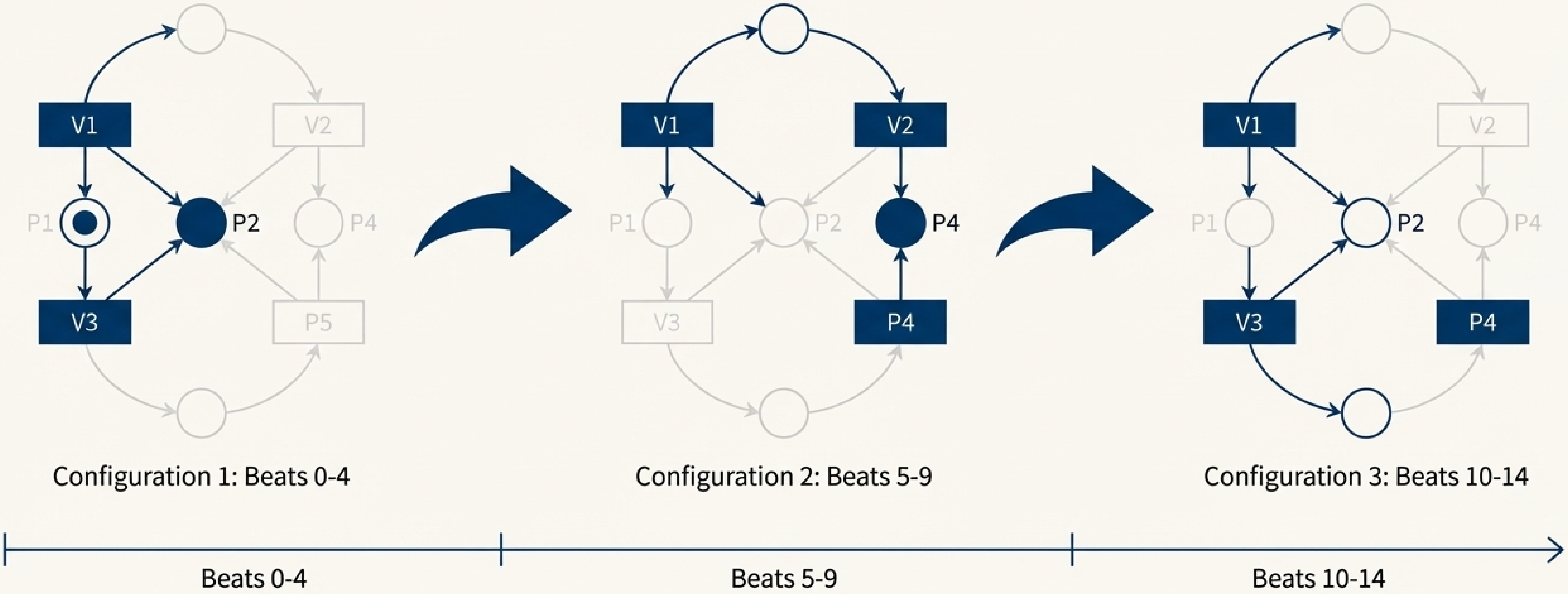


Requires adding **8 places** and **4 transitions** to the initial petrinet to achieve the same behavior.

"Metapetrinets offer more compact facilities to the models of complex dynamic processes than single level petrinets."

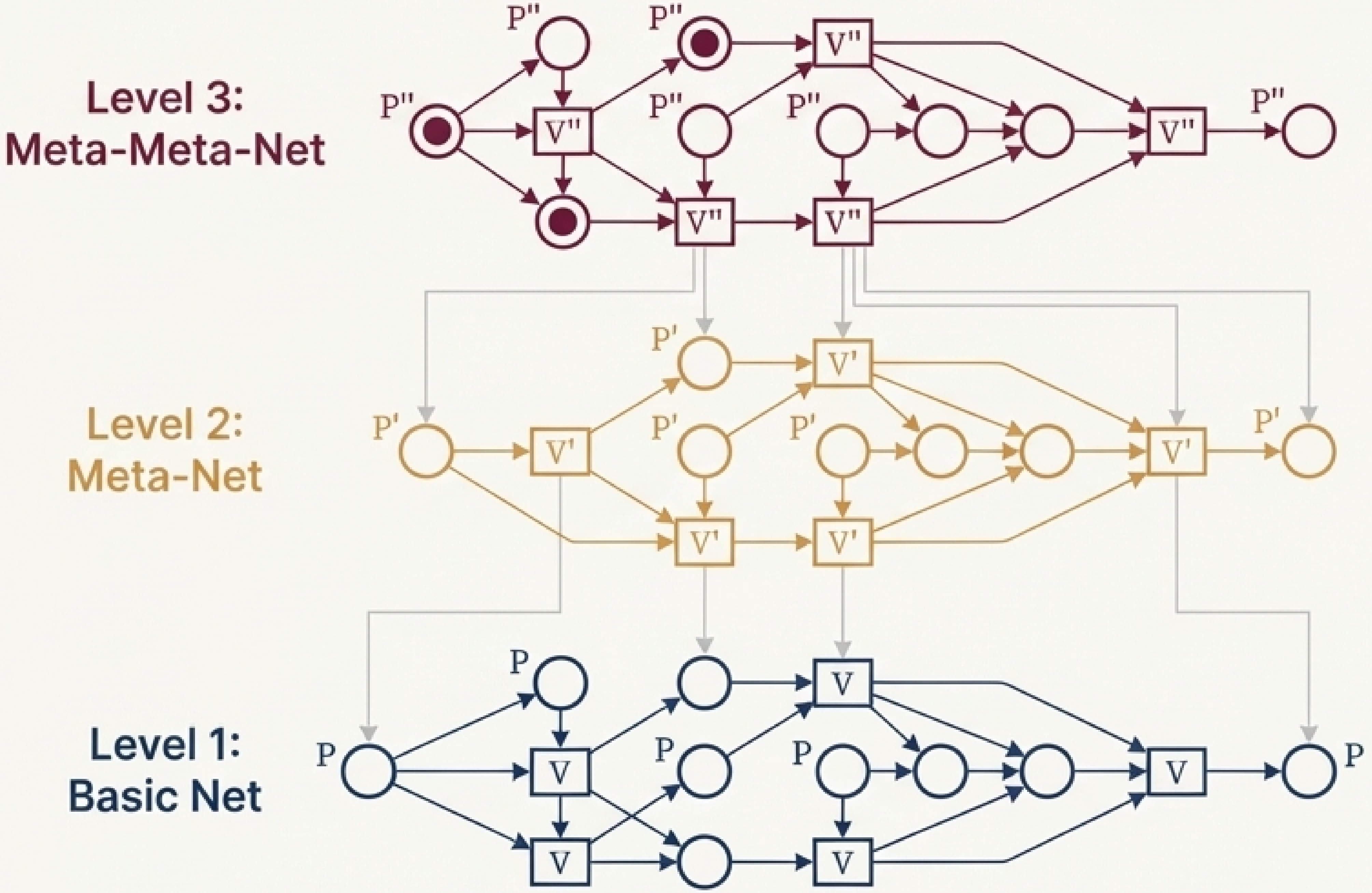
Visualizing Dynamic Reconfiguration with a Mixed MPN

In a mixed MPN, the metalevel controls both places and transitions. Its execution causes the basic level's structure to evolve through distinct configurations over time.



Expanding the Concept: Multilevel Metapetri-Nets

For exceptionally complex processes, the control logic itself may require its own control layer. The MPN framework can be extended to any number of levels, creating a control hierarchy.



The more complex a process it is necessary to simulate, the more metalevels are needed to describe it in an adequate and compact way.

A Universal Framework for Dynamic Control

The meta-level's influence extends far beyond simply activating or deactivating nodes. It provides a comprehensive framework for controlling virtually any attribute of a basic level petrinet.

<Place>

- 3 Change capacity
- Change marking

<Transition>

- Change time settings
- Change the fire rule

<Link>

- Change direction
- 5 Change capacity

<Token>

- Change color
- Change place

The Significance of Metapetri-Nets

Compactness

Dramatically reduces the complexity of models for dynamic systems compared to single-level nets.

Efficiency

Execution can be optimized by only testing transitions within the currently active substructure, improving performance in computer-based simulations.

Dynamic Control

Provides a simple, powerful, and generalizable framework for modeling systems that reconfigure their own structure and rules.

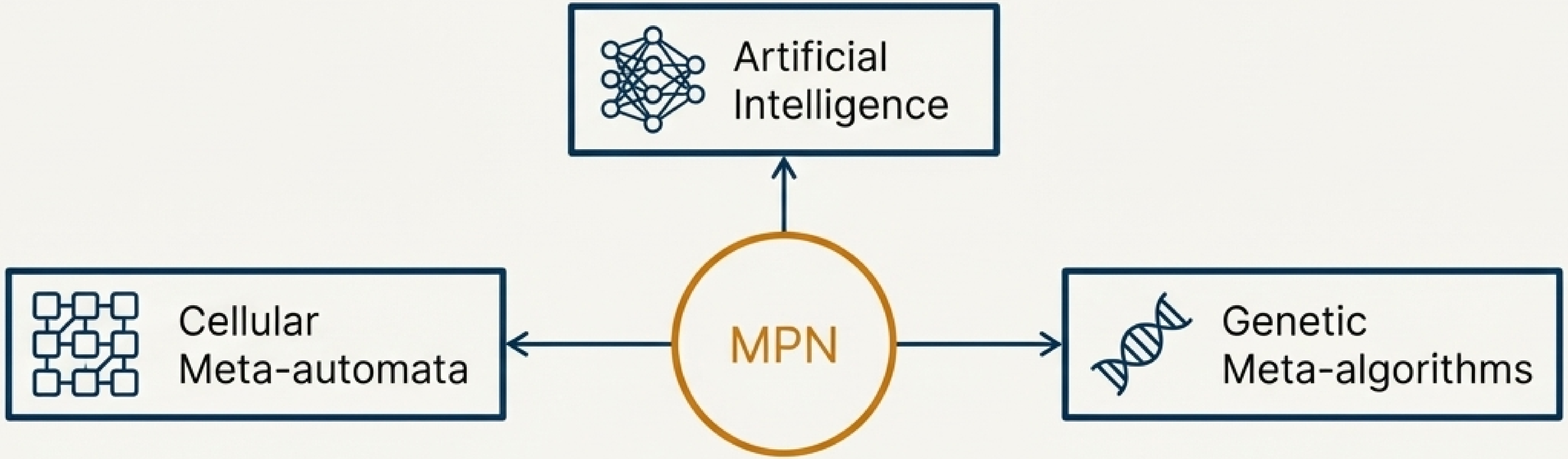
From Theory to Practice

The MPN technique has been successfully applied to real-world problems and is being explored for others.

- Applied: Simulation of precious metal flow at the Kharkov Jewelry Plant.
- Future Investigation: Planned for the area of flight connection control.

A Deterministic Model for Intelligent Systems

The completely deterministic behavior of an MPN, combined with its hierarchical control structure, opens up interesting possibilities in other domains. The ability to model systems that can modify their own operational logic is a powerful feature.



By providing a formal way to model structural dynamics, Metapetri-Nets offer a foundational tool for designing the next generation of complex, adaptive, and intelligent systems.