What is claimed is:

1. A computer-implemented method for executing semantic object-resident computation in a distributed computing system, the method comprising:

   instantiating a persistent executable object stored in non-transitory memory, the persistent executable object comprising an intent field encoding a machine-readable execution descriptor, a context block encoding execution-relevant metadata, and a memory field encoding prior execution state;

   propagating the persistent executable object among a plurality of execution nodes of the distributed computing system;

   at each execution node that receives the persistent executable object, performing an execution evaluation cycle comprising:

   parsing the intent field to identify an execution operation expressed by the machine-readable execution descriptor;

   evaluating the context block against locally applicable execution policy without reliance on centralized coordination;

   reading the memory field to retrieve one or more prior execution records stored by a previous execution evaluation cycle; and

   selecting, based solely on the parsed intent field, the evaluated context block, and the retrieved prior execution records, an execution action selected from the group consisting of execution, mutation, delegation, dormancy, reentry, and termination;

   executing the selected execution action; and

   recording an execution outcome corresponding to the selected execution action by appending a new execution record to the memory field of the persistent executable object,

   wherein execution continuity across multiple execution lifecycles is maintained by the memory field of the persistent executable object.

2. The method of claim 1, wherein the persistent executable object persists across asynchronous execution intervals and resumes execution without re-instantiation based on one or more reentry conditions encoded in the memory field.

3. The method of claim 1, wherein the persistent executable object transitions into a dormant state when execution conditions encoded in the context block or memory field

are unmet and subsequently transitions to a reentry state upon satisfaction of a reentry condition recorded in the memory field.

4. The method of claim 1, wherein mutation comprises modifying at least one of the intent field, the context block, or the memory field of the persistent executable object in response to an execution outcome.
5. The method of claim 4, wherein each mutation is recorded as a distinct memory record in the memory field, thereby preserving execution lineage of the persistent executable object.
6. The method of claim 1, wherein delegation comprises instantiating one or more subordinate executable objects derived from the persistent executable object.
7. The method of claim 6, wherein execution outcomes generated by one or more subordinate execution objects are aggregated into the memory field of the persistent execution object.
8. The method of claim 1, further including forming a distributed execution graph defined by object-resident lineage references stored in respective memory fields from a plurality of persistent executable objects.
9. The method of claim 1, wherein evaluation of locally applicable execution policy is performed using one or more policy references encoded within the persistent executable object.
10. The method of claim 9, wherein different execution nodes select different execution actions for the same persistent executable object based on locally applicable execution policy while preserving execution continuity through the memory field.
11. The method of claim 1, further including reusing a previously executed persistent executable object to fulfill a new execution operation when execution history stored in the memory field and metadata stored in the context block satisfy local evaluation criteria.
12. The method of claim 1, wherein at least one execution node comprises a probabilistic inference engine configured to generate a recommended execution action for the persistent executable object, wherein the recommended execution action is recorded as an execution outcome in the memory field without computing a performance state or controlling access to a system capability.

13. The method of claim 1, wherein multiple persistent executable objects operate concurrently and coordinate execution through object-resident delegation, mutation, and lineage tracking without centralized control.
14. The method of claim 1, wherein the persistent executable object self-terminates upon satisfaction of a terminal condition recorded in the memory field.
15. The method of claim 1, wherein the execution node does not store execution progress, execution eligibility, or execution history for the persistent executable object outside the memory field of the persistent executable object.
16. The method of claim 1, wherein the persistent executable object is serialized for propagation between execution nodes and deserialized prior to each execution evaluation cycle, such that execution continuity is preserved independently of execution node identity.
17. The method of claim 1, wherein the memory field is append-only and prior execution records are not overwritten during mutation, delegation, or termination.
18. The method of claim 1, wherein execution outcomes include partial execution results and negative capability signals recorded in the memory field, and further including using the execution outcomes to govern subsequent execution behavior.
19. A non-transitory computer-readable medium storing instructions that, when executed by one or more processors, cause the processors to perform the method of claim 1.
20. A distributed computing system, comprising:

a plurality of execution nodes; and

a persistent execution object configured to propagate among the execution nodes, the persistent execution object comprising:

an intent field encoding a machine-readable execution descriptor;

a context block encoding execution-relevant metadata; and

a memory field storing an append-only execution history,

wherein each execution node is configured to:

parse the intent field, evaluate the context block, and read the memory field;

select an execution action as a function of data stored within the intent field, the context block, and the memory field without centralized coordination; and append an execution outcome corresponding to the selected execution action to the memory field,

such that execution continuity of the persistent executable object is preserved by an object-resident execution state.