01 · TITLE

Perslis is the *AI operational runtime.*

A runtime layer that lets AI understand, operate, verify, and improve real software systems, including legacy desktops and virtual machines.

model agnostic | desktop aware | fail closed

Perslis

The body that lets any model operate.

02 · PROBLEM

Models cannot reliably touch the systems companies still run on.

Legacy OSes, siloed apps, and brittle operator workflows sit outside the clean API world. Models can describe them, but they lose reliability when they must see, click, inject payloads, compile binaries, and prove the outcome.

> The missing layer is not more reasoning. It is verified execution against messy real targets.

LEGACY EXECUTION TARGETS

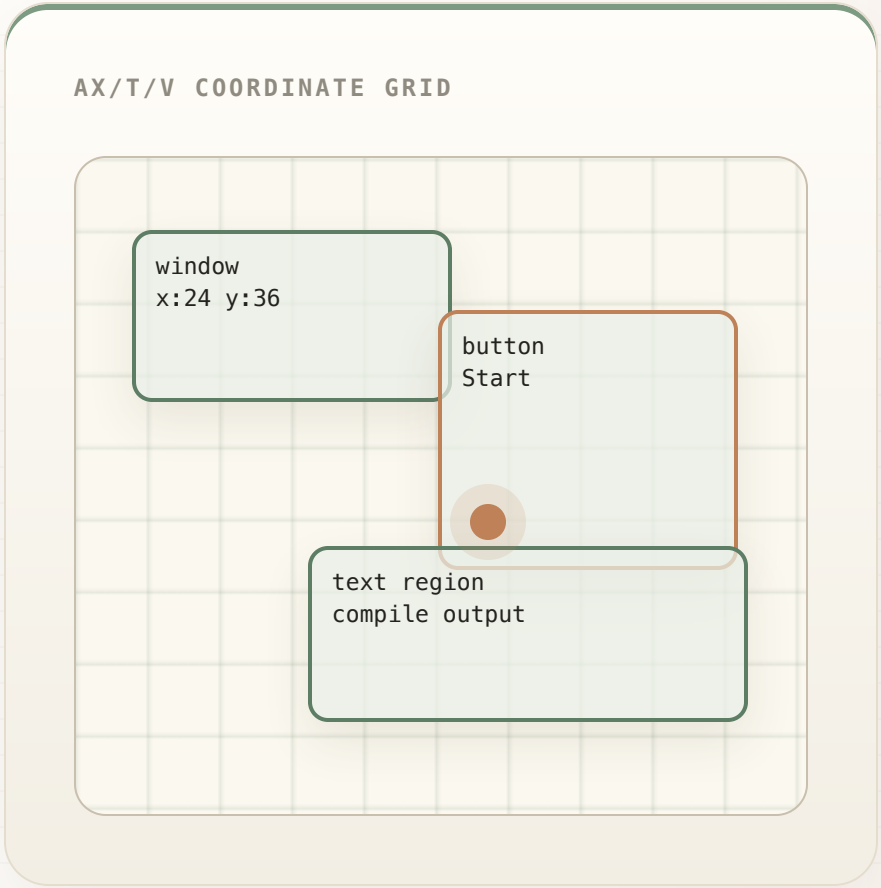
target	Windows 95 / 98	GUI
target	COBOL / DOS	binary
target	Mainframe logic	batch
target	Modern SaaS	API
gap	Operator bridge	missing

03 · PERCEPTION

Tinky Vision turns screens into stable spatial maps.

The runtime sees the desktop as coordinates, text, UI regions, and intent targets. Visual context is streamed into the loop so non-visual models can still operate software through grounded observations.

AX tree · OCR · bounding boxes · click targets

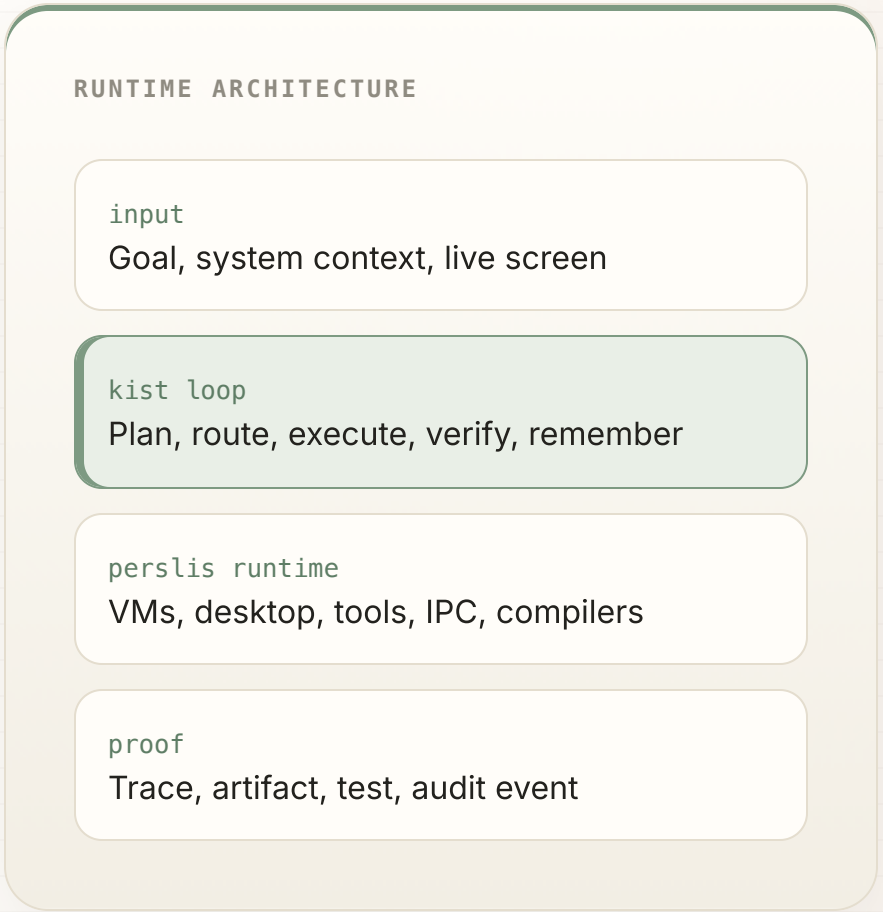


04 · ARCHITECTURE

Voltron OS assembles proof from static traces and live state.

Kist records what the system knows, what changed, what ran, and what was verified. Perslis reassembles that evidence in real time so the model cannot simply claim completion.

kist.live/#now kist.live/#arch



05 · MULTI-SUBSTRATE PROOF

One operator loop across old desktops, emulators, and compilers.

Perslis proves the runtime by executing across different substrates: legacy Windows, modern Windows, emulator surfaces, COBOL, DOS, and direct binary tooling.

> The important proof is not a demo script. It is compiling, injecting, running, and checking artifacts across execution targets.

HARDWARE-FREE EXECUTION

Win98

Notepad, browser, legacy UI control

Win10

Modern desktop workflows and APIs

Flycast

Emulated runtime proof surfaces

COBOL

Direct compilation and host-run binaries

06 · IPC PLUMBING

Host and guest communicate through direct payload channels.

Perslis does not depend on fragile manual copy-paste. It can move payloads through QMP, guest bridges, window automation, and application-specific adapters.

QMP socket | Notepad bridge | Slack to Win95

BRIDGE TRACE

```
host  slack.event.message
  |
  ├─ normalize payload
  ├─ verify target session
  ├─ inject via guest channel
  |
win95  NOTEPAD.EXE receives text
  └─ capture evidence and checksum
```

07 · SAFETY

The Captain makes autonomy fail closed.

The runtime requires evidence before claiming work is done. If tools fail, model output drifts, or verification is missing, the loop blocks completion and records a repair target.

> Self-healing is useful only when it is bounded, auditable, and honest about failure.

CAPTAIN TERMINAL GATE

```
research · light gate PASS
execute · artifact changed
verify · missing proof
captain refused completion
opened bounded repair mission
status: fail closed
```

08 · GOVERNANCE

Every action becomes ledgered operational evidence.

Kist normalizes tool events, decisions, verification records, and failures into an audit stream that can be exported into security and compliance systems.

kist.live/#self

ECS normalized

SIEM export

TAMPER-EVIDENT LEDGER

event.id	**tool.call**	hash:91af
actor	**mimo planner**	signed
target	**win98 session**	scoped
outcome	**completion refused**	audited
export	**ECS / SIEM**	ready

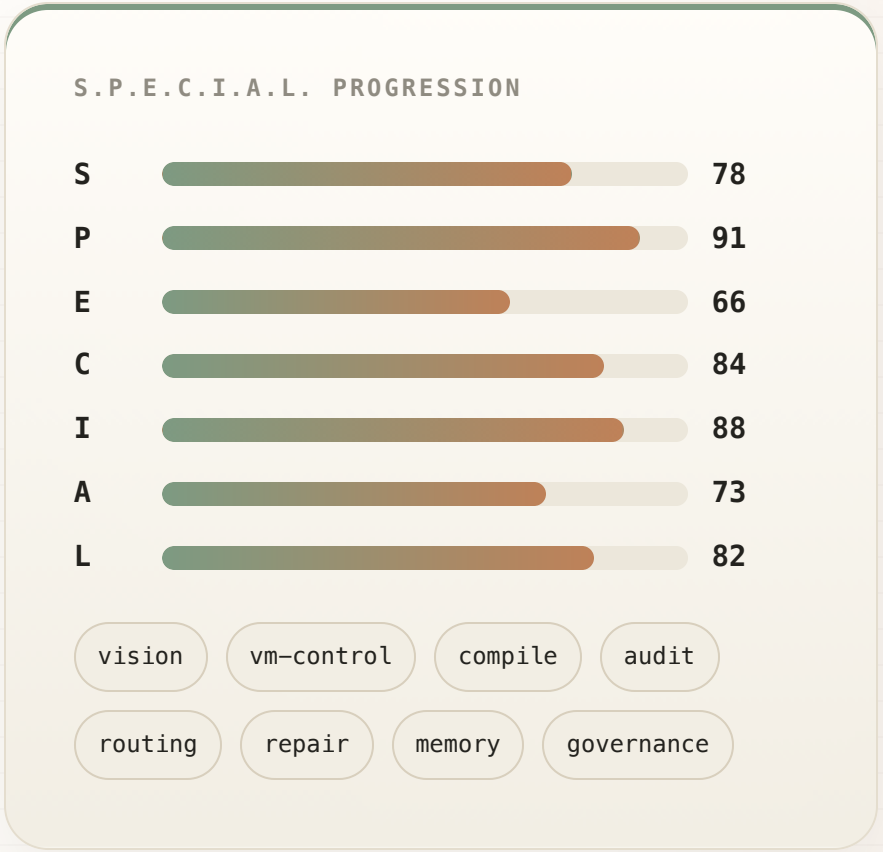
09 · MEMORY

Agents improve through retained operating memory.

The system keeps structured memories about failures, skills, tool behavior, user preferences, and environment facts. Progression turns repeated proof into better routing and safer defaults.

8-class skill tree

511+ retained memories



10 · BUSINESS MODEL

Gated engine, open loop, enterprise control plane.

Kist grows adoption through an open operational loop. Perslis monetizes the hard runtime layer: legacy execution, guest control, audit exports, governance, and enterprise support.

> The ask: fund the bridge from AI reasoning to verified operation in the systems enterprises cannot replace.

DUAL-LAYER LICENSING

Open Kist Loop

Community adoption, local workflows, memory, routing, and verification patterns.

Gated Perslis Engine

VM orchestration, IPC bridges, enterprise governance, audit export, and support.

`developer pull` `enterprise revenue`