

4SO Workspace

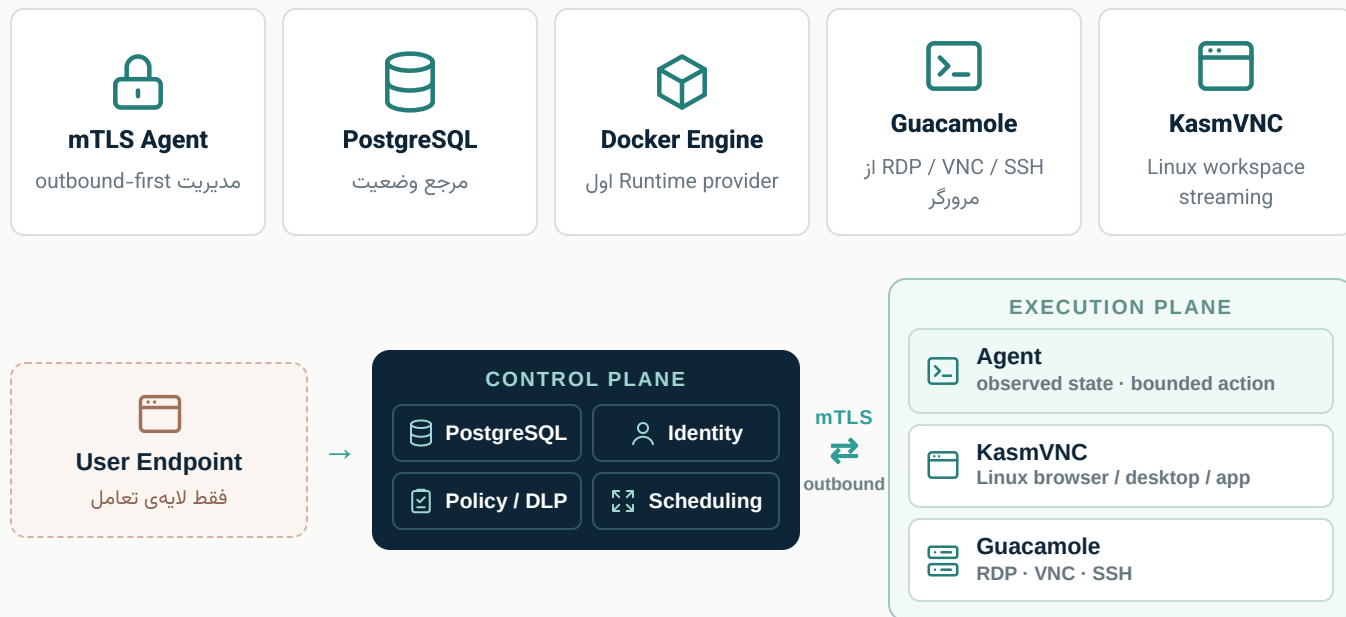
بروشور فنی

Secure Browser Workspaces with Separated Control & Execution Planes

RDP/VNC/SSH لینوکسی و دسترسی Application و Browser، Desktop

از یک تجربه‌ی وب؛ با Identity، Policy، Scheduling و DLP، و چرخه‌ی عمر

پایدار Session، و اجرای محافظت‌شده جدا از دستگاه کاربر.



Single-node درجه‌یک است

Control و Execution می‌توانند روی یک Host باشند؛ Kubernetes گزینه‌ی استقرار است، نه پیش‌شرط تجربه.

Endpoint محل اعتماد نیست

Session و اجرای workload در محیط کنترل‌شده باقی می‌مانند و Endpoint فقط لایه‌ی تعامل است.

محیط کار را از Endpoint جدا کنید؛ تجربه‌ی کاربر را ساده نگه دارید.



CAPABILITY SNAPSHOT

Capability Snapshot

تجربه‌ی کاربر ساده، زیرساخت پشت آن جدی

Workspace قابلیت‌های User Experience و Infrastructure Operations را در یک محصول کنار هم نگه می‌دارد.



Policy و DLP

فایل، Clipboard، Peripheral و Network Path



هویت سازمانی

Authentication، Authorization و دامنه



RDP / VNC / SSH

دسترسی از مرورگر بدون کلاینت پراکنده



App و Browser، Desktop

محیط‌های Linux موقت و کنترل‌شده



Observability

Audit و Health، Capacity دید



Recording

مشاهده، همکاری و Audit در سطح Session



Multi-zone

گسترش Execution بدون جابه‌جایی Control



Capacity و Scheduling

انتخاب نود اجرایی بر اساس وضعیت

اصول عملیاتی



مدیریت نود outbound-first و با mTLS



Browser UI فقط از Product API عبور می‌کند



وضعیت Session از Runtime مشاهده می‌شود



Kubernetes اختیاری است، نه پیش‌شرط

مرز مالکیت محصول



PostgreSQL — Intent، Session، Policy، Node Capacity



Control ≠ Execution — دو نقش مستقل



Agent — وضعیت مشاهده‌شده، نه Authority



Single-node — مدل استقرار درجه یک

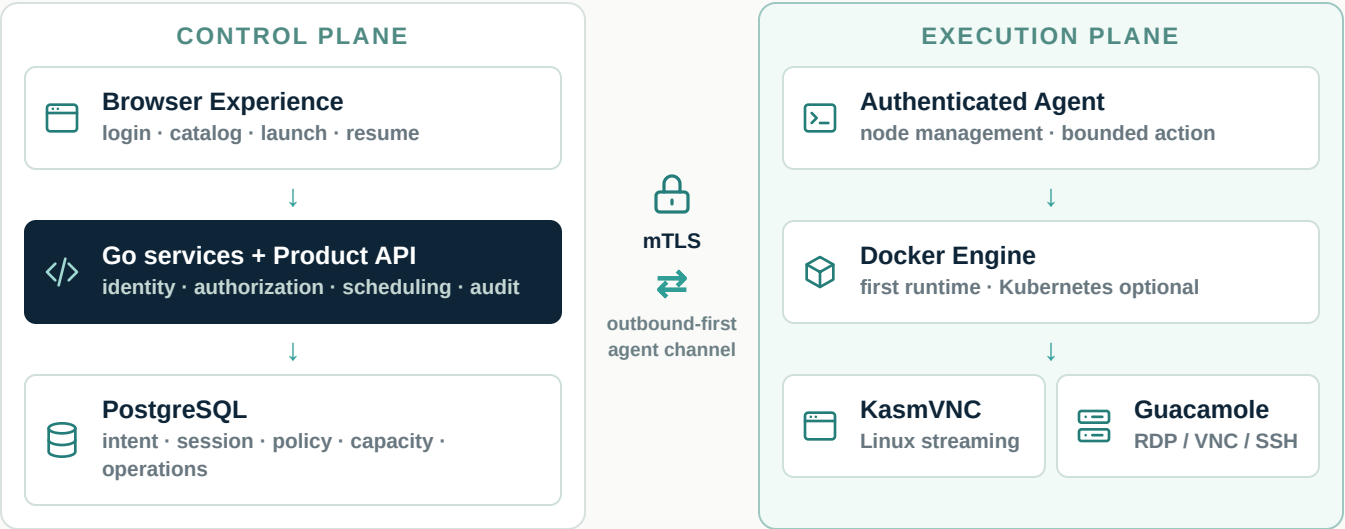
Core contract: Session = Intent + Durable Operation + Runtime Observation — هیچ مرحله‌ای به‌تنهایی موفقیت نیست.



Architecture Overview

Control Plane و Execution Plane عمداً یکی نیستند

Intent و Policy در Control Plane می‌مانند؛ Agent‌ها مدیریت را انجام می‌دهند و Session واقعی در Execution Plane اجرا می‌شود.



لایه	Component	قرارداد
State Authority	PostgreSQL	UI و Agent منبع وضعیت مستقل نمی‌سازند
Control Plane	Go services + Product API	Execution، Identity، Scheduling و audit؛ جدا از Execution
Agent Plane	Outbound-first mTLS agents	Authority از Product API و RBAC عبور می‌کند
Linux Streaming	KasmVNC	اجرای محافظت‌شده روی Execution Plane می‌ماند
Remote Access	Apache Guacamole	پروتکل راه دور پشت همان Identity و Policy



Observed, not assumed

وضعیت Session از Runtime خوانده می‌شود



No direct mutation

Browser UI تغییر مستقیم Runtime ندارد



Deterministic release

API، Controller، Edge، CLI و migration‌ها در یک artifact تحویل می‌شوند

Deployment & Install Contract

از تک سرور تا Execution Plane چند ناحیه‌ای

پروفایل	Control Plane	Execution	کاربرد
تک نود	همان Server	همان Server	شروع ساده و محیط کوچک
Control + نود اجرایی	مستقل	چند نود اجرایی	تفکیک مدیریت از اجرای Workspace
Multi-zone	مرکزی	Zone در چند Execution	نزدیکی به کاربر و توزیع ظرفیت
HA Control Plane	چند نودی	مستقل Execution Pool	افزونگی Control Plane

ARTIFACT-BOUND INSTALL



Safe resume

و resume_required
owner از last_safe_resume_point
boundary

Unsafe cache rejected

Hard-link و permission نامن رد می‌شوند

Loopback TLS edge

PostgreSQL و Control API روی Host منتشر
نمی‌شوند

No Kubernetes tax: Runtime provider جزئیات پیاده‌سازی زیر Product lifecycle است؛ تجربه‌ی عادی Workspace به Kubernetes وابسته نیست.



Action & Workflow Contract

Session یک درخواست لحظه‌ای نیست

و وضعیت Session جدا از وضعیت درخواست مشاهده می‌شود.
Launch، Stop، Restart و Recovery از Product API به Operation و Outbox می‌روند



Failure / Recovery	Success	Execution / Lock	Admission	Action
recovery یا retry event؛ UI نتیجه را حدس نمی‌زند	Session ID و Runtime مشاهده شده	Idempotency-Key → Operation / Outbox → Scheduler / Agent	مجوز، Actor/Project و Catalog، Policy Capacity	Launch
پیش از lease قابل cancel؛ بعد از آن Reconcile	Session دیگر running نیست	Operation مستقل به / Agent Runtime	Session موجود و مجوز معتبر	Stop
Correlation مبدأ و جانشین حفظ می‌شود	Successor مشاهده شده و قابل استفاده	Restart attempt + successor binding	Source session معتبر	Restart
تعداد و زمان تلاش‌ها در History	Runtime دوباره Ready	RECOVER_RUNTIME صریح، نه عمومی restart	وضعیت / failure recovery	Recover Runtime
بدون اعلام موفقیت جانشین کاذب	منابع یتیم پاک و همگرا	RECOVER_CLEANUP روی همان Session	Runtime نیمه‌کاری قابل اثبات	Recover Cleanup
Session فعال بی‌صدا یتیم نمی‌شود	Placement از وضعیت مشاهده شده	Scheduling هماهنگ با چرخه عمر نود	Health و ظرفیت پیش از Add / Drain / Remove	Node Capacity

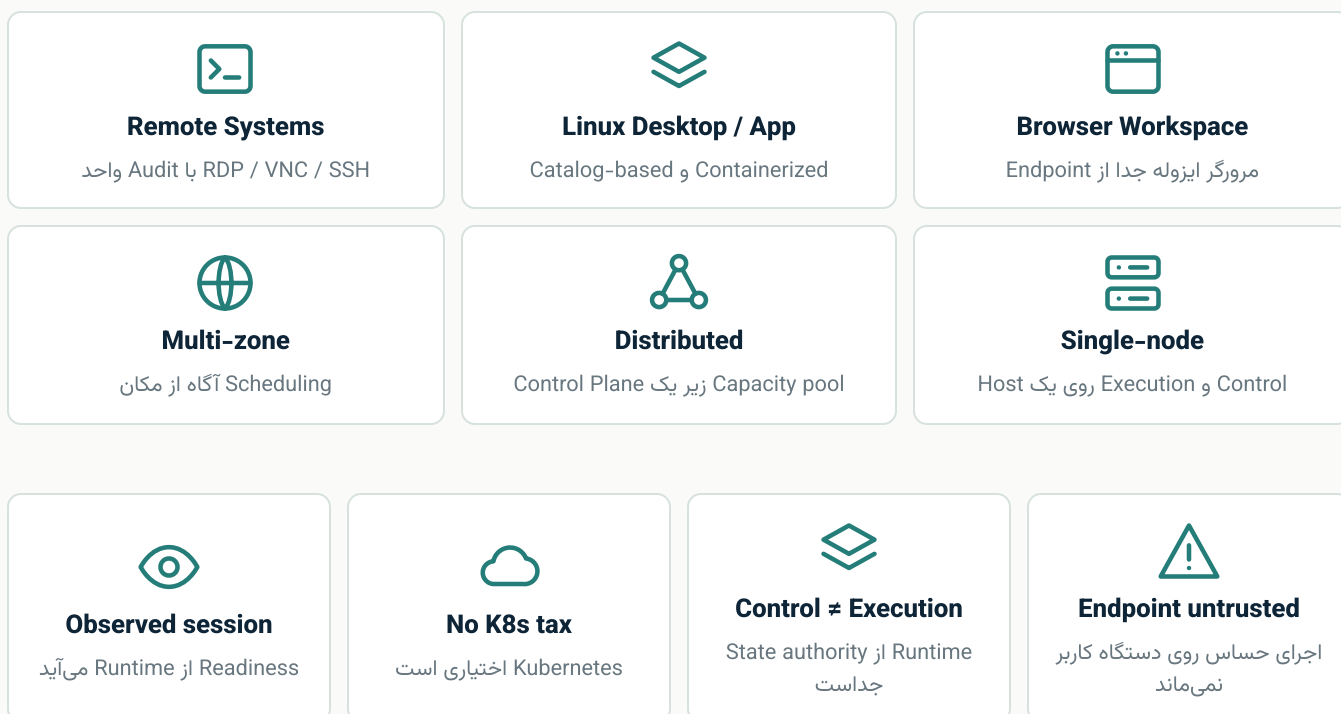
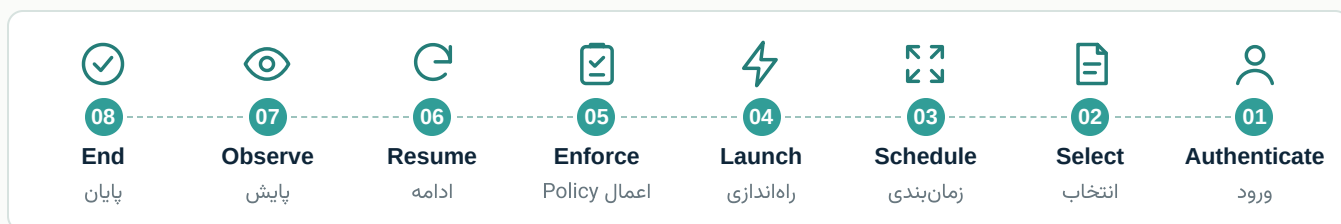
Safe cancel: فقط در REQUESTED یا ADMITTED، بدون lease owner و پیش از عبور از مرز side effect؛ History
تلاش‌ها و آخرین recovery را نگه می‌دارد.



Lifecycle & Workspace Profiles

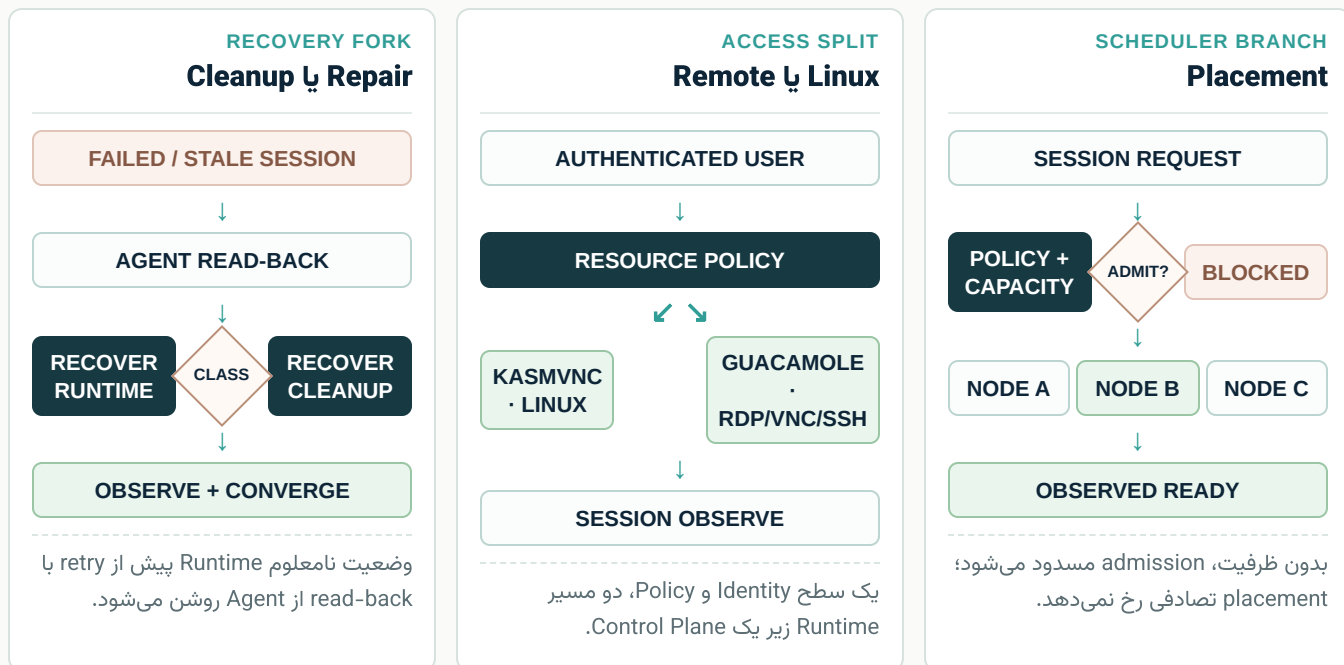
Session فقط Launch نیست

از احراز هویت تا Scheduling، اجرای Recording، Resume، Policy و پایان، همه بخشی از چرخه‌ی عمر کاربر و زیرساخت‌اند.



Failure & Recovery Semantics

Runtime و Browser، Control، Agent شکست‌های متفاوتی دارند



Recovery / Fence	Product Behavior	Detection	Failure Domain
Reattach بدون ساخت دوباره‌ی Authority	Runtime می‌تواند ادامه دهد	Telemetry کلاینت و Session	Browser disconnect
اتصال دوباره‌ی Agent و Reconcile	Mutation جدید مسدود یا degraded	Heartbeat و وضعیت مشاهده‌شده	Agent unavailable
Restart یا RECOVER_RUNTIME	Session نیازمند توجه یا بازبینی	وضعیت failure در Session	Runtime crash
RECOVER_CLEANUP	موفقیت تازه اعلام نمی‌شود	کشف منابع یتیم	Partial cleanup
Policy طبق Relaunch و Drain	Admission تغییر می‌کند	وضعیت Scheduler و نود	Capacity loss

Data protection: Session موقت با داده‌ی پایدار کاربر یکی نیست؛ جایگزینی نود نباید هویت Session را بدون Reconcile رها کند.



Action & Operations Catalog

در یک محصول Infrastructure Journey و User Journey



Session •

- recover-runtime
- restart
- stop
- recover-cleanup



Workspace •

- status
- launch
- catalog-list
- attach



Capacity •

- rebalance
- schedule
- observe



Node •

- remove
- drain
- preflight
- register
- replace



Diagnostics •

- operation history
- status-json
- doctor



Policy •

- access scope
- network policy
- DLP

Evidence و Observability |



Session در recovery و Status: attention و تلاش‌های



وضعیت نود و Capacity برای اپراتور



Unknown به جای Healthy حدسی



Identity و RBAC پیش از admission



mTLS برای اتصال Agent و اعتماد نود



Credential یک بار مصرف؛ بدون Secret در JSON

Action safety: UI و Agent همان Product API را مصرف می‌کنند؛ shell یا runtime bypass جایی در workflow عادی ندارد.



Workspace امن، بدون تبدیل Endpoint به رمز اعتماد.

4SO Workspace دسترسی امن را از یک اتصال پراکنده به یک محصول قابل اداره تبدیل می‌کند.

