MOUSSAIFXT · COGNITIVE KERNEL · PART III · REFERENCE RUNTIME & PROOF CLOSURE

موسوعة الإغلاق التنفيذي العميق لنواة MOUSSAIFXT الإدراكية

من المعمارية والتقنيات الحدّية إلى Runtime مرجعي قابل للتكرار، الإثبات، الاسترجاع، والـE4/E5

Part III من سلسلة هندسة MOUSSAIFXT Cognitive Kernel

تركيب هندسي من Part I + Part II + MUT/MUG/MUT-AFI + Research Gap Audit

Arabic RTL · Amiri · A4 · 2026

الحالة: Reference Engineering Specification — لا تدّعي تحقق AGI أو إثبات أداء لم يُقَس.

DEEP CLOSURE · EXECUTABLE CONTRACTS · FAILURE PROOF · BENCHMARK GATES

العقد التنفيذي لهذا الجزء

Part III لا يضيف طبقات نظرية جديدة لمجرد الإغلاق الشكلي. مهمته تحويل الثغرات التي بقيت بعد Part I وPart II إلى عقود وبرمجيات مرجعية قابلة للتنفيذ والقياس. العائق الذي اعتمده هذا الجزء هو ما حدده تدقيق الفجوات: **frozen baselines، reference implementation، cross-model/environment replication، failure replay، world-effect verification، ablation، telemetry.**

> **قاعدة الكتاب:** كل طبقة يجب أن تنتج metric، failure mode، state transition، schema، وpath rollback/recovery، test. architecture مكتملة لفظياً لا تكفي.

ما لا يعاد

- لا نعيد CK-01→CK-50 كتفسيرات نظرية.
- لا نعيد EX-01→EX-24 ولا X0→X7 كتعريفات عامة.
- لا نعيد MCD/QMCD/MCAI/MURI/MSRAK/MUT-AFI إلا بقدر ما يلزم لعقد التنفيذ.

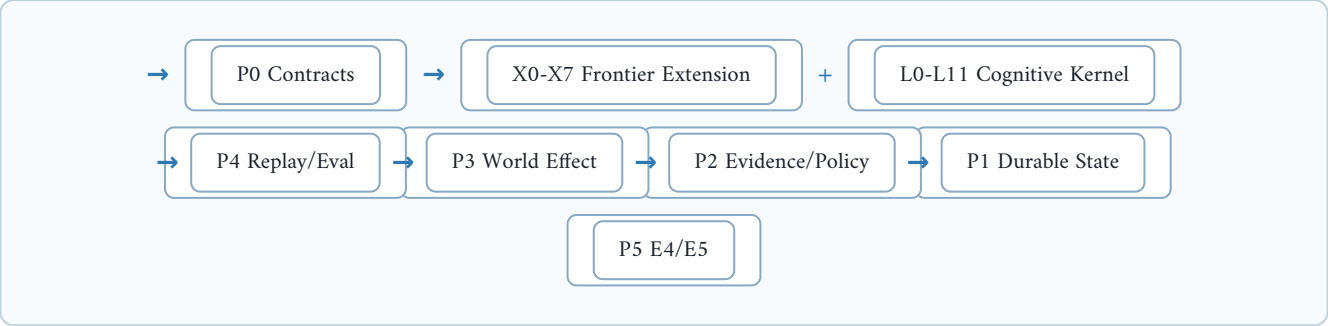
ما يُضاف

40 Proof-Closure Components، ثم schemas موحدة، monorepo، benchmark matrix، threat model، error taxonomy، مسار بناء Sprint 12، وتجربة end-to-end واحدة تمثل Minimal Complete Slice.

```
Theory → Contract → Runtime → Trace → World Effect → Replay → Ablation → E4 →
Replication → E5
```

الخريطة الجديدة: من L/X إلى طبقة الإثبات P

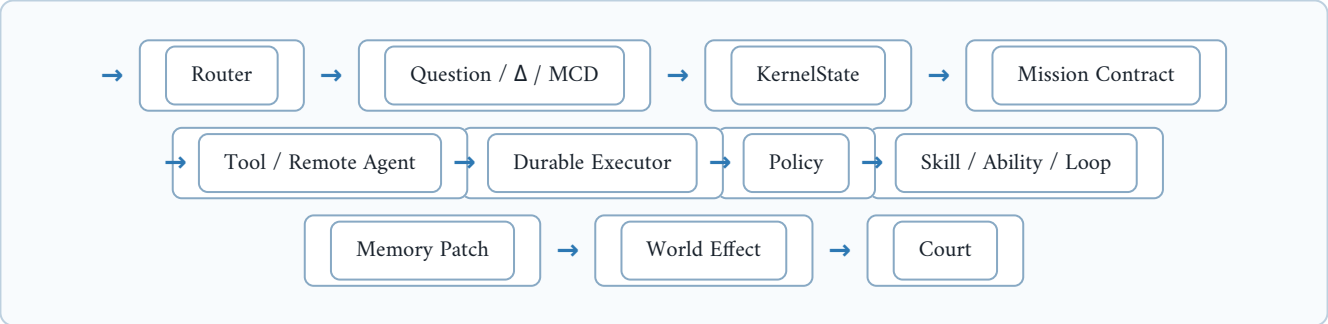
Part I حافظ على 12 طبقة L0-L11. Part II أضاف Extension Planes X0-X7. Part III يضيف مستوى مختلفاً لا يسمى Layer جديدة: **Proof Closure Plane P**. هذه الطبقة تقطع جميع الطبقات أفقياً وتسأل: هل يمكن تشغيلها وإعادة تشغيلها وهدمها وقياسها؟



الفكرة الحاكمة

MOUSSAIFXT لا يحتاج الآن إلى مزيد من الأسماء بقدر حاجته إلى حزمة صغيرة من artifacts المرجعية تثبت أن الأسماء تتصل ببعضها فعلاً: state store، event ledger، router، loop engine، policy engine، tool gateway، world-effect verifier، memory promotion، trace store، وbenchmark harness.

المعمارية المرجعية العليا



الاختلاف المركزي في Part III هو أن كل سهم من هذه الأسهم يصبح transaction أو event أو gate واضحاً. لا توجد «وصلة مفهومية» غير ممثلة في runtime.

Invariant الأعلى

```
RuntimeSuccess = ContractPass ∧ AuthorityPass ∧ ArtifactPass ∧ EvidencePass ∧
PostconditionPass ∧ RecoveryReady
```

> **Not enough:** LLM says done، tool returns 200، file exists، agent agrees with itself.

الفهرس الهندسي Part III

المجموعة	المكوّنات
State & Knowledge	PC-01..04
Execution & Routing	PC-05..13
Evidence & World Effect	PC-14..18
Memory & Context	PC-19..22
Models & Compute	PC-23..27
Policy / SaaS / Security	PC-28..30
Observability / Failure	PC-31..33
Training / Promotion / Migration	PC-34..37
Benchmarks / E4-E5 / Reference Runtime	PC-38..40

مبدأ القراءة

كل مكوّن له خمس صفحات ثابتة: الفرق، العقد، الخوارزمية، الفشل/الأمن، ثم القياس والنضج. التكرار في الشكل مقصود لأنه يجعل الكتاب قابلاً للتحويل لاحقاً إلى specification registry أو code generation templates.

سلم الأدلة المستخدم

المستوى	المعنى	ما يمنع الترقية
E0	فكرة/هدف غير محقق.	لا artifact.
E1	تعريف، حدود، آلية مقترحة.	لا schema أو tests.
E2	Artifact/Contract/Schema.	لا runtime مرجعي.
E3	تنفيذ مرجعي واختبارات.	لا baseline مضبوط.
E4	تحسن تجريبي مضبوط.	لا replication مستقل.
E5	تكرار عبر نماذج/مجالات/بيئات.	يبقى محدوداً بنطاق الأدلة.

قاعدة إزالة الطبقات: إذا لم ينجُ المكوّن من ablation أو تبين أن طبقة أبسط تحقق نفس النتيجة، يدمج أو يحذف من Runtime المرجعي.

تعريف Minimal Complete Slice

الشريحة المرجعية الدنيا ليست أصغر demo. هي أصغر نظام يثبت كل انتقال حرج مرة واحدة على الأقل. يجب أن يستطيع:

- استقبال MissionContract محدد.
- إنشاء KernelState ذي نسخة.
- استخراج سؤال/فرق وتوجيه مورد واحد على الأقل.
- اختيار Skill أو Loop واحد.
- اتخاذ قرار سلطة واضح.
- تنفيذ side effect idempotent.
- التحقق من world effect من مصدر مستقل.
- كتابة MemoryPatch مع provenance.
- التوقف بأمان وإعادة التشغيل من checkpoint.

```
MCS = mission → state → route → action → effect → memory → replay
```

كل ما لا يحتاجه هذا المسار الأول يبقى خارج النواة حتى تثبت ضرورته.

PC-01

الحالة المعيارية الواحدة للنواة

Canonical KernelState v3

> **فجوة الإغلاق.** توحيد جميع صور الحالة التي ظهرت في Part I وPart II وMUT/MUG في كائن واحد ذي إصدار، مع منع امتلاك كل Adapter أو Agent نسخة شبه مستقلة من الحقيقة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- هوية المهمة والنسخة
- السؤال الحاكم والفرق الفاعل
- سياق السلطة والسياسة
- الادعاءات والأدلة
- الخطة والمسار
- الأثر والذاكرة

Invariant

```
PC-01 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-01 · CONTRACT

الحالة المعيارية الواحدة للنواة - العقد والبيانات

Canonical Schema

```
KernelState{mission_id,version,goal_contract,question,active_differences,claims,evidence,authority,route,plan,world_before,world_after,courts,memory_delta,budget,recovery_pointer}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

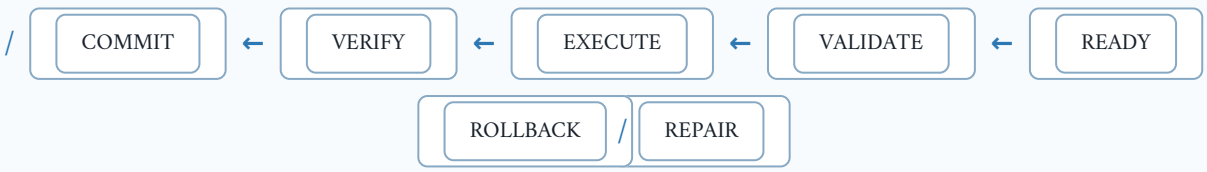
PC-01 · ALGORITHM

الحالة المعيارية الواحدة للنواة - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-01 · FAILURE / SECURITY

الحالة المعيارية الواحدة للنواة - الفشل والأمن والرصد

أنماط الفشل الحرج

- state_fork
- lost_update
- stale_read
- adapter_shadow_state

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-01 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-01 · EVALUATION

الحالة المعيارية الواحدة للنواة - المقاييس وE0→E5

المقاييس الأساسية

- state_convergence_rate
- lost_update_rate
- replay_fidelity
- schema_validity

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-01 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-02

سجل الأحداث الحاكم

Canonical Event Log

فجوة الإغلاق. جعل كل انتقال في النواة حدثاً قابلاً للإعادة، بدلاً من الاعتماد على state snapshots غير القادرة على شرح لماذا تغيرت الحالة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- event_id
- causal_parent
- actor
- before_hash
- delta
- after_hash

Invariant

```
PC-02 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-02 · CONTRACT

سجل الأحداث الحاكم - العقد والبيانات

Canonical Schema

```
KernelEvent{event_id,mission_id,type,actor,ts,input_refs,decision,delta,before_hash,after_hash,trace_id,authority_ref}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

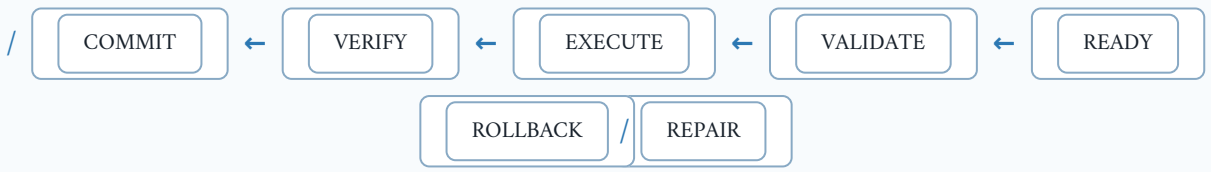
PC-02 · ALGORITHM

سجل الأحداث الحاكم - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-02 · FAILURE / SECURITY

سجل الأحداث الحاكم - الفشل والأمن والرصد

أنماط الفشل الحرج

- event_gap
- causal_orphan
- duplicate_event
- clock_skew

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-02 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-02 · EVALUATION

سجل الأحداث الحاكم - المقاييس وE0→E5

المقاييس الأساسية

- event_completeness
- causal_link_rate
- replay_success
- duplicate_event_rate

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-02 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-03

ذرة المعرفة ذات النسب

KnowledgeAtom v2

> **فجوة الإغلاق.** إغلاق فجوة توحيد المعرفة بين HyperKB وGraphRAG والذاكرة والأدلة بحيث تصبح الذرة قابلة للتقييم والتحديث والإبطال عبر النظام كله.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- identity
- claim
- scope
- source
- time
- confidence
- sensitivity
- dependencies
- supersedes
- retrieval_cues

Invariant

```
PC-03 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-03 · CONTRACT

ذرة المعرفة ذات النسب - العقد والبيانات

Canonical Schema

```
KnowledgeAtom{id,kind,claim,scope,source_ref,valid_from,valid_to,confidence,sensitivity,dependencies[],supersedes[],contradictions[],retrieval_cues[],status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

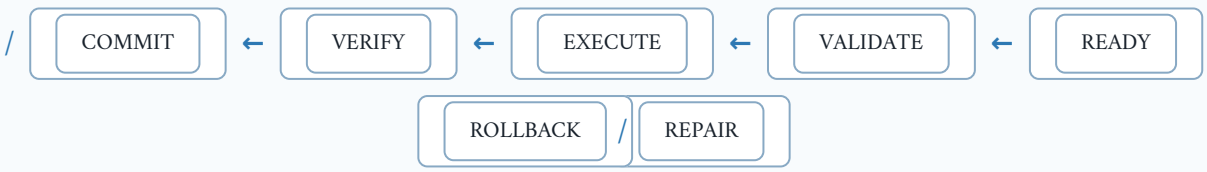
PC-03 · ALGORITHM

ذرة المعرفة ذات النسب - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-03 · FAILURE / SECURITY

ذرة المعرفة ذات النسب - الفشل والأمن والرصد

أنماط الفشل الحرج

- provenance_loss
- scope_leak
- stale_atom
- duplicate_atom

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-03 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-03 · EVALUATION

ذرة المعرفة ذات النسب - المقاييس وE0→E5

المقاييس الأساسية

- provenance_coverage
- staleness_error
- dedup_precision
- retrieval_usefulness

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-03 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-04

بوابة تحويل الدليل إلى سلطة

Evidence-to-Authority Gate

> **فجوة الإغلاق.** الفصل الصارم بين صحة معلومة وحقها في قيادة الفعل؛ المعرفة القوية لا تمنح وحدها إذن التنفيذ.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- epistemic_status
- authority_scope
- action_risk
- reversibility
- institutional_rule

Invariant

```
PC-04 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-04 · CONTRACT

بوابة تحويل الدليل إلى سلطة - العقد والبيانات

Canonical Schema

```
AuthorityDecision{principal,action,resource,context,evidence_status,risk,decision,obligations,expiry}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

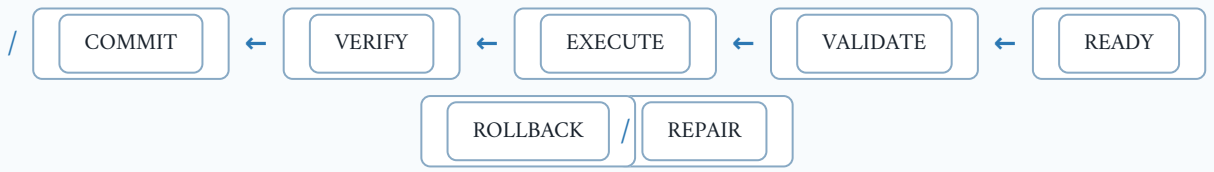
PC-04 · ALGORITHM

بوابة تحويل الدليل إلى سلطة - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-04 · FAILURE / SECURITY

بوابة تحويل الدليل إلى سلطة - الفشل والأمن والرصد

أنماط الفشل الحرج

- confidence_as_permission
- scope_escalation
- policy_bypass
- stale_authority

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission component=PC-04 authority input_hash decision effect cost latency verdict.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-04 · EVALUATION

بوابة تحويل الدليل إلى سلطة - المقاييس وE0→E5

المقاييس الأساسية

- false_allow
- false_block
- policy_latency
- obligation_completion

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-04 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-05

معاملة الأثر Exactly-Once

Effect Transaction

> **فجوة الإغلاق.** منع تكرار الأثر الخارجي عند retry أو crash أو resume، خصوصاً في البريد وCRM والملفات والواجهات التي تملك world effect حقيقياً.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- idempotency_key
- precondition
- lease
- attempt
- receipt
- postcondition
- compensation

Invariant

```
PC-05 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-05 · CONTRACT

معاملة الأثر Exactly-Once - العقد والبيانات

Canonical Schema

```
EffectTxn{id,key,action,resource,precondition_hash,lease,attempts[],receipt,verified_postcondition,compensation,status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

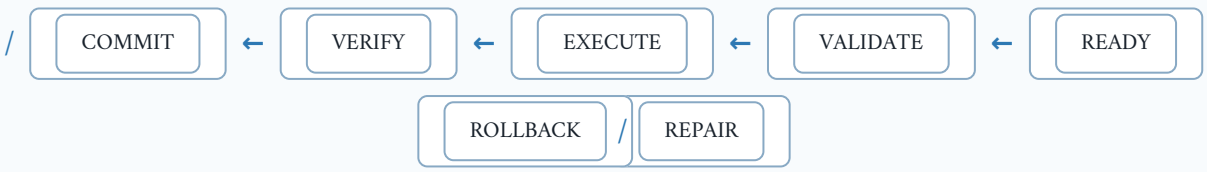
PC-05 · ALGORITHM

معاملة الأثر Exactly-Once - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-05 · FAILURE / SECURITY

معاملة الأثر Exactly-Once - الفشل والأمن والرصد

أنماط الفشل الحرج

- double_send
- duplicate_charge
- lost_receipt
- partial_commit

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-05 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-05 · EVALUATION

معاملة الأثر Exactly-Once - المقاييس وE0→E5

المقاييس الأساسية

- duplicate_effect_rate
- commit_correctness
- compensation_success
- resume_safety

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-05 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-06

محرك HyperLoop الحتمي

Deterministic Bounded HyperLoop

> **فجوة الإغلاق.** تحويل HyperLoop من وصف agentic إلى آلة حالات تفرض الميزانية والتوقف والتحقق والإصلاح والتراجع برمجياً.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- INIT
- RUN
- VERIFY
- REPAIR
- ESCALATE
- ROLLBACK
- DONE

Invariant

```
PC-06 :: observed_state + explicit_contract + authority + evidence + verification >
                                                                  model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-06 · CONTRACT

محرك HyperLoop الحتمي - العقد والبيانات

Canonical Schema

```
LoopState{loop_id,state,iteration,budget,goal,evidence_threshold,failure_threshold,checkpoint,last_verdict}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

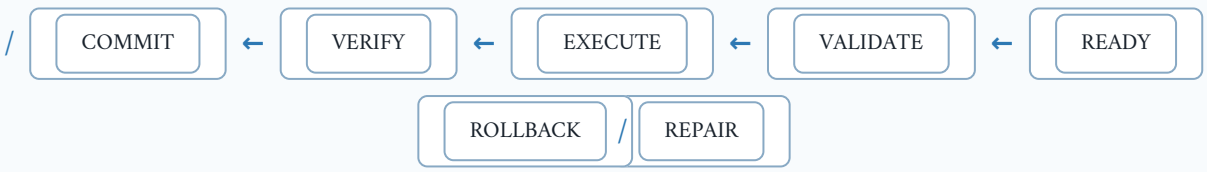
PC-06 · ALGORITHM

محرك HyperLoop الحتمي - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-06 · FAILURE / SECURITY

محرك HyperLoop الحتمي - الفشل والأمن والرصد

أنماط الفشل الحرج

- unbounded_loop
- plateau
- budget_overrun
- non_resumable_loop

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission، component=PC-06، authority، input_hash، decision، effect، cost، latency، verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-06 · EVALUATION

محرك HyperLoop الحتمي - المقاييس وE0→E5

المقاييس الأساسية

- safe_exit_rate
- plateau_detection
- resume_fidelity
- verified_closure

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-06 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-07

الموجه المتناثر القابل للقياس

Sparse HyperRouter v2

> **فجوة الإغلاق.** إغلاق الفجوة بين مفهوم التوجيه الوعائي والتوجيه القابل للقياس عبر استرجاع دلالي+رمزي، pruning بالسلطة، beam متكيف، وحساب regret.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- semantic_candidates
- symbolic_candidates
- authority_prune
- compatibility_prune
- adaptive_beam
- counterfactual_replay

Invariant

```
PC-07 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-07 · CONTRACT

الموجه المتناثر القابل للقياس - العقد والبيانات

Canonical Schema

```
RouteDecision{candidates[],pruned[],selected,score,expected_gain,cost,risk,regret_estimate,
reason}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event
→ compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

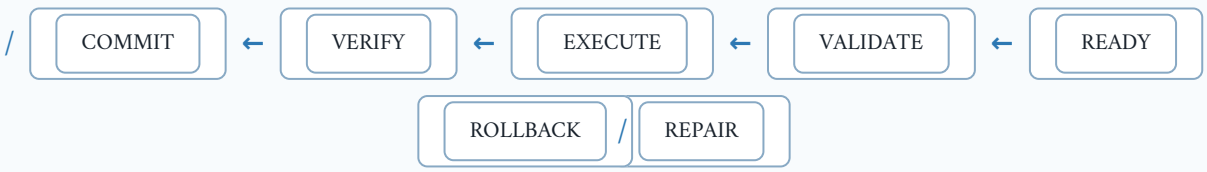
PC-07 · ALGORITHM

الموجه المتناثر القابل للقياس - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-07 · FAILURE / SECURITY

الموجه المتناثر القابل للقياس - الفشل والأمن والرصد

أنماط الفشل الحرج

- route_collapse
- expensive_overrouting
- unsafe_route
- semantic_only_bias

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-07 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-07 · EVALUATION

الموجه المتناثر القابل للقياس - المقاييس وE0→E5

المقاييس الأساسية

- router_regret
- cost_quality_frontier
- unsafe_route_rate
- counterfactual_gap

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-07 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-08

مترجم المهارات إلى تنفيذ

HyperSkill Compiler v2

> **فجوة الإغلاق.** تحويل skill من ملف إرشادي إلى عقد تنفيذ قابل للتحليل الساكن والاختبار والتوقيع والإصدار.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- trigger
- preconditions
- input_schema
- steps
- tool_caps
- postconditions
- failure_modes
- tests

Invariant

```
PC-08 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-08 · CONTRACT

مترجم المهارات إلى تنفيذ - العقد والبيانات

Canonical Schema

```
CompiledSkill{id,version,trigger,input_schema,preconditions,plan_template,tool_caps,postconditions,failure_modes,tests,signature}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

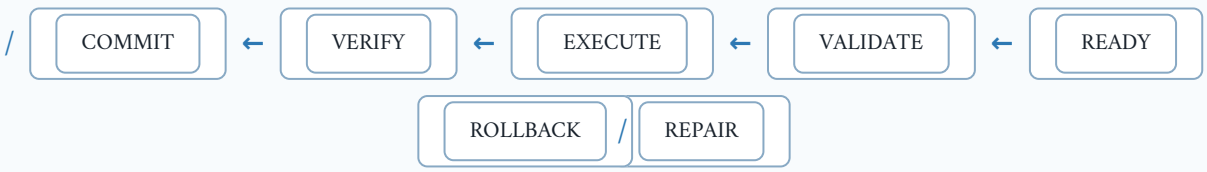
PC-08 · ALGORITHM

مترجم المهارات إلى تنفيذ - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-08 · FAILURE / SECURITY

مترجم المهارات إلى تنفيذ - الفشل والأمن والرصد

أنماط الفشل الحرج

- prompt_only_skill
- hidden_side_effect
- missing_postcondition
- version_drift

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-08 · authority · input_hash · decision · effect · cost · latency · verdict.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-08 · EVALUATION

مترجم المهارات إلى تنفيذ - المقاييس وE0→E5

المقاييس الأساسية

- compile_rate
- static_error_rate
- skill_transfer
- postcondition_pass

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-08 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-09

إغلاق القدرة التركيبية

HyperAbility Closure

> **فجوة الإغلاق.** إثبات أن تجميع skills/loops/tools يخلق قدرة مركبة مستقرة، لا مجرد قائمة أجزاء ناجحة منفردة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- dependency_graph
- critical_edges
- contracts
- closure_tests
- transfer_tests

Invariant

```
PC-09 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-09 · CONTRACT

إغلاق القدرة التركيبية - العقد والبيانات

Canonical Schema

```
AbilityManifest{id,dependencies[],critical_edges[],entry_contract,exit_contract,closure_tests[],rollback,status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

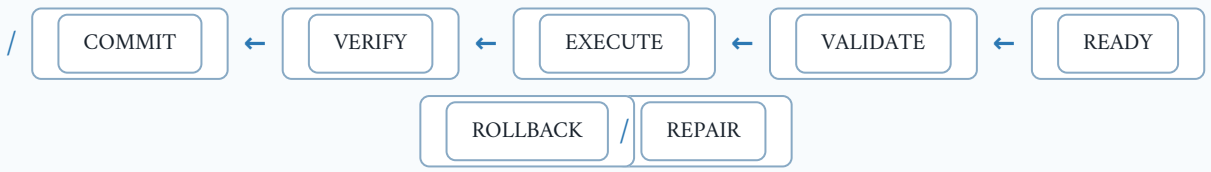
PC-09 · ALGORITHM

إغلاق القدرة التركيبية - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-09 · FAILURE / SECURITY

إغلاق القدرة التركيبية - الفشل والأمن والرصد

أنماط الفشل الحرج

- weakest_edge_failure
- hidden_dependency
- contract_mismatch
- local_only_success

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-09 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-09 · EVALUATION

إغلاق القدرة التركيبية - المقاييس وE0→E5

المقاييس الأساسية

- closure_rate
- weak_edge_frequency
- transfer_success
- marginal_gain

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-09 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-10

مخطط المهمة القابل للتنفيذ

Mission DAG v2

> **فجوة الإغلاق.** ربط التخطيط بالاعتماديات والسلطة والأثر بحيث تصبح كل عقدة قابلة للتنفيذ والاستئناف والمراجعة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- node_id
- deps
- owner
- authority
- tool_caps
- precondition
- postcondition
- compensation

Invariant

```
PC-10 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-10 · CONTRACT

مخطط المهمة القابل للتنفيذ - العقد والبيانات

Canonical Schema

```
MissionNode{id,deps[],owner,authority_ref,action,preconditions[],postconditions[],compensation,cost_estimate,status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

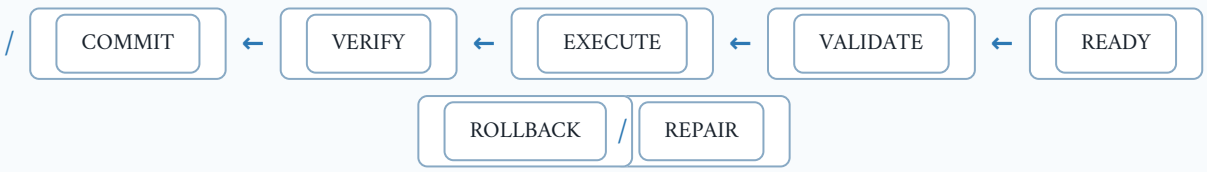
PC-10 · ALGORITHM

مخطط المهمة القابل للتنفيذ - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-10 · FAILURE / SECURITY

مخطط المهمة القابل للتنفيذ - الفشل والأمن والرصد

أنماط الفشل الحرج

- hidden_dependency
- deadlock
- premature_execution
- orphan_node

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-10 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-10 · EVALUATION

مخطط المهمة القابل للتنفيذ - المقاييس وE0→E5

المقاييس الأساسية

- dag_validity
- critical_path_error
- resume_success
- dependency_violation

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-10 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-11

جدولة طويلة الأفق

Durable Workflow Scheduler

> **فجوة الإغلاق.** فصل Mission DAG الدلالي عن محرك التنفيذ المتين الذي يوفر checkpoint وreplay وretry وسياسة timeouts.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- workflow_id
- checkpoint
- activity
- retry_policy
- timeout
- heartbeat

Invariant

```
PC-11 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-11 · CONTRACT

جدولة طويلة الأفق - العقد والبيانات

Canonical Schema

```
WorkflowRun{id,mission_id,checkpoint,activities[],retries,timeouts,heartbeats,recovery,status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

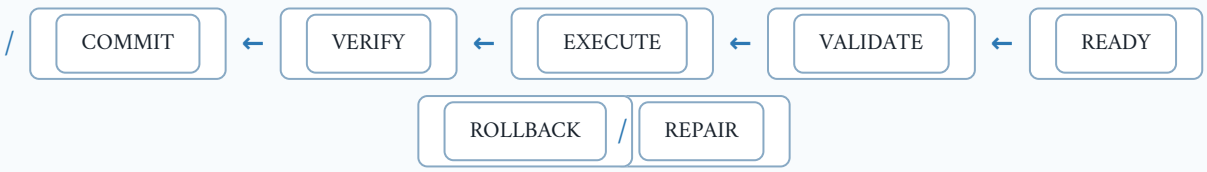
PC-11 · ALGORITHM

جدولة طويلة الأفق - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-11 · FAILURE / SECURITY

جدولة طويلة الأفق - الفشل والأمن والرصد

أنماط الفشل الحرج

- non_deterministic_replay
- retry_storm
- lost_checkpoint
- timeout_misclassification

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-11 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-11 · EVALUATION

جدولة طويلة الأفق - المقاييس وE0→E5

المقاييس الأساسية

- workflow_recovery
- replay_determinism
- retry_efficiency
- mean_recovery_time

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-11 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-12

حد الثقة للوكلاء البعيدين

Remote Agent Trust Boundary

> **فجوة الإغلاق.** منع A2A أو أي اتحاد وكلاء من منح الوكيل البعيد حق الكتابة في الذاكرة المعيارية أو رفع السلطة مباشرة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- agent_card
- capability_claim
- trust_class
- evidence_boundary
- commit_prohibition

Invariant

```
PC-12 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-12 · CONTRACT

حد الثقة للوكلاء البعيدين - العقد والبيانات

Canonical Schema

```
RemoteAgentSession{agent_id,capabilities,trust_class,allowed_tasks,evidence_outputs,prohibited_effects,expiry}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

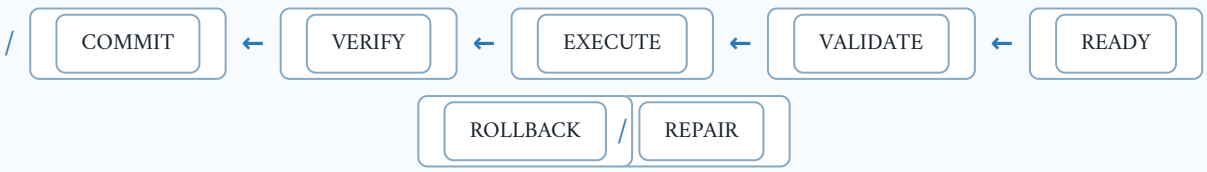
PC-12 · ALGORITHM

حد الثقة للوكلاء البعيدين - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-12 · FAILURE / SECURITY

حد الثقة للوكلاء البعيدين - الفشل والأمن والرصد

أنماط الفشل الحرج

- agent_impersonation
- capability_overclaim
- remote_commit
- evidence_poisoning

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-12 · authority · input_hash · decision · effect · cost · latency · verdict.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-12 · EVALUATION

حد الثقة للوكلاء البعيدين - المقاييس وE0→E5

المقاييس الأساسية

- remote_claim_accuracy
- boundary_violations
- verification_yield
- delegation_success

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-12 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-13

بوابة الأدوات المعيارية

Tool Gateway v3

فجوة الإغلاق. توحيد native adapters وMCP والواجهات المحلية تحت ToolContract واحد حتى لا يصبح transport هو مصدر دلالة أو سلطة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- tool_id
- transport
- capability
- auth_scope
- input_schema
- output_schema
- side_effect_class

Invariant

```
PC-13 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-13 · CONTRACT

بوابة الأدوات المعيارية - العقد والبيانات

Canonical Schema

```
ToolContract{id,transport,endpoint,capability,input_schema,output_schema,auth_scope,side_effect_class,idempotency,healthcheck}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

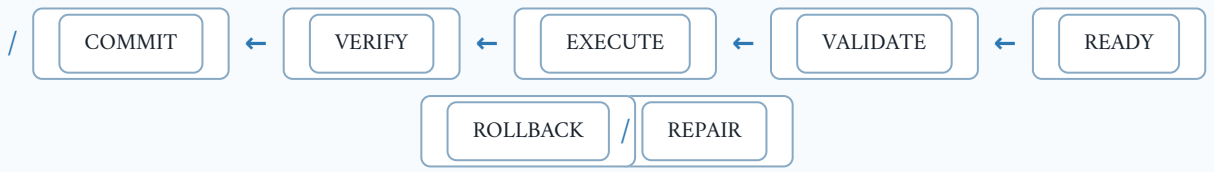
PC-13 · ALGORITHM

بوابة الأدوات المعيارية - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-13 · FAILURE / SECURITY

بوابة الأدوات المعيارية - الفشل والأمن والرصد

أنماط الفشل الحرج

- transport_confusion
- schema_drift
- hidden_mutation
- tool_impersonation

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-13 · authority · input_hash · decision · effect · cost · latency · verdict.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-13 · EVALUATION

بوابة الأدوات المعيارية - المقاييس وE0→E5

المقاييس الأساسية

- tool_success
- schema_error
- side_effect_detection
- transport_parity

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-13 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-14

التحقق المستقل من الأثر

World-Effect Verifier v2

> **فجوة الإغلاق.** إغلاق الفرق بين tool receipt وبين التغير الحقيقي في البيئة السلطوية عبر readback مستقل وpostconditions معرفة مسبقاً.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- expected_state
- authoritative_readback
- invariants
- residual_gap
- side_effect_scan

Invariant

```
PC-14 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-14 · CONTRACT

التحقق المستقل من الأثر - العقد والبيانات

Canonical Schema

```
WorldEffect{action_ref,expected,observed,source,attribution,side_effects[],residual_gap,verdict}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

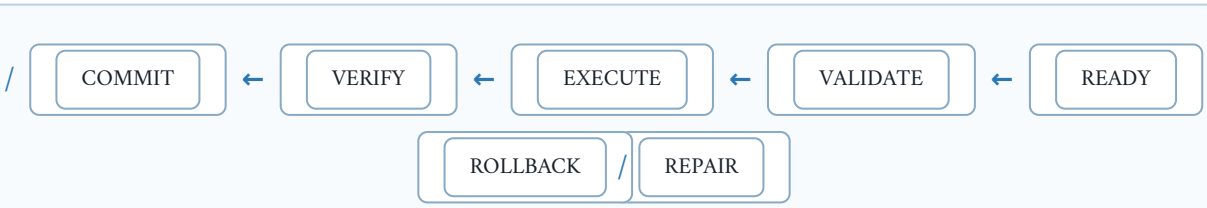
PC-14 · ALGORITHM

التحقق المستقل من الأثر - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-14 · FAILURE / SECURITY

التحقق المستقل من الأثر - الفشل والأمن والرصد

أنماط الفشل الحرج

- false_done
- receipt_as_effect
- unread_side_effect
- stale_readback

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission component=PC-14 authority input_hash decision effect cost latency verdict.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-14 · EVALUATION

التحقق المستقل من الأثر - المقاييس وE0→E5

المقاييس الأساسية

- world_effect_verification
- false_done_rate
- residual_gap_size
- independent_readback_rate

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-14 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-15

الإسناد السببي للأثر

Causal Attribution Runtime

فجوة الإغلاق. تحديد أي مكون أو فعل ساهم فعلاً في الأثر بدلاً من نسبة النجاح إلى آخر خطوة أو إلى النواة كلها.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- causal_graph
- intervention
- counterfactual
- mediator
- confounder
- credit

Invariant

```
PC-15 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-15 · CONTRACT

الإسناد السببي للأثر - العقد والبيانات

Canonical Schema

```
CausalCredit{target_effect,contributors[],interventions[],counterfactuals[],confounders[],credits[],uncertainty}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

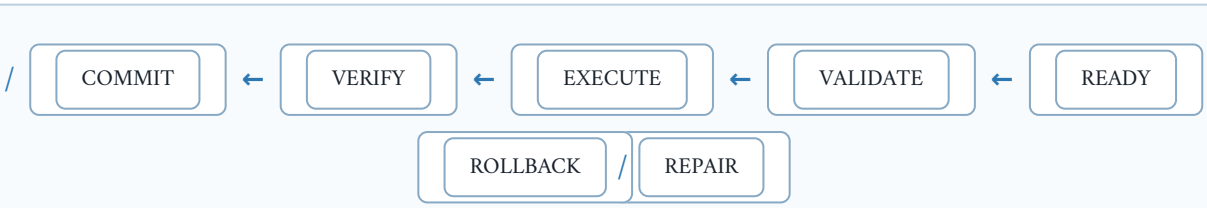
PC-15 · ALGORITHM

الإسناد السببي للأثر - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-15 · FAILURE / SECURITY

الإسناد السببي للأثر - الفشل والأمن والرصد

أنماط الفشل الحرج

- correlation_credit
- confounding
- mediator_omission
- posthoc_attribution

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission component=PC-15 authority input_hash decision effect cost latency verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-15 · EVALUATION

الإسناد السببي للأثر - المقاييس وE0→E5

المقاييس الأساسية

- credit_stability
- counterfactual_fit
- confounder_coverage
- causal_prediction

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-15 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-16

هندسة الدليل

Evidence Geometry Runtime

فجوة الإغلاق. تمثيل استقلال الأدلة وجذورها المشتركة ومباشرتها وتناقضها لكي لا يعادل النظام بين عشر نسخ من مصدر واحد وعشرة مصادر مستقلة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- evidence_graph
- shared_roots
- directness
- coverage
- contradiction_mass

Invariant

```
PC-16 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-16 · CONTRACT

هندسة الدليل - العقد والبيانات

Canonical Schema

```
EvidenceGraph{claims[],evidence_nodes[],edges[],shared_roots[],directness,coverage,contradictions[],independence_score}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

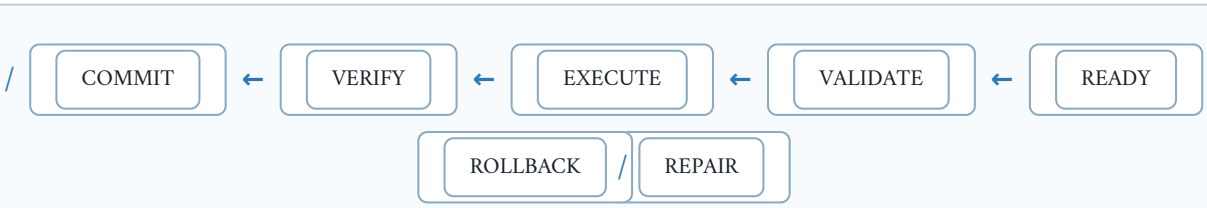
PC-16 · ALGORITHM

هندسة الدليل - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-16 · FAILURE / SECURITY

هندسة الدليل - الفشل والأمن والرصد

أنماط الفشل الحرج

- source_echo
- citation_chain
- contradiction_suppression
- coverage_blindspot

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-16 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-16 · EVALUATION

هندسة الدليل - المقاييس وE0→E5

المقاييس الأساسية

- independence_score
- contradiction_detection
- coverage_ratio
- verification_efficiency

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-16 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-17

محرك المحاكم كآلة حالات

Court Engine v2

> **فجوة الإغلاق.** تحويل محاكم الهدم والتكذيب والدليل والسلطة من تعليمات نصية إلى state machine بقرارات ملزمة ومسارات اعتراض.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- FILED
- ADMISSIBLE
- CHALLENGE
- EVIDENCE
- VERDICT
- APPEAL
- CLOSED

Invariant

```
PC-17 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-17 · CONTRACT

محرك المحاكم كآلة حالات - العقد والبيانات

Canonical Schema

```
CourtCase{id,type,claim_ref,challengers[],evidence_refs[],ruleset,verdict,confidence,appeal
,status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event
→ compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

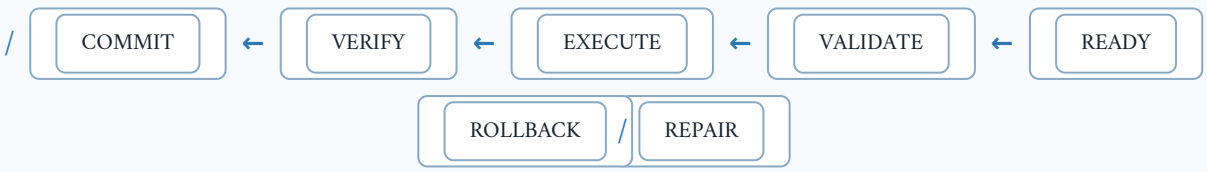
PC-17 · ALGORITHM

محرك المحاكم كآلة حالات - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-17 · FAILURE / SECURITY

محرك المحاكم كآلة حالات - الفشل والأمن والرصد

أنماط الفشل الحرج

- court_theater
- verifier_capture
- circular_evidence
- nonbinding_verdict

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-17 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-17 · EVALUATION

محرك المحاكم كآلة حالات - المقاييس وE0→E5

المقاييس الأساسية

- verdict_precision
- appeal_reversal
- independence_index
- decision_latency

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-17 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-18

استقلال المدقق

Verifier Independence Protocol

> **فجوة الإغلاق.** منع planner/executor من أن يصبحا المدقق الوحيد لنتيجتهما عبر فصل model/provider/context/tool paths عند الأثر العالي.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- independence_class
- shared_context
- shared_model
- shared_tool
- blind_verification

Invariant

```
PC-18 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-18 · CONTRACT

استقلال المدقق - العقد والبيانات

Canonical Schema

```
VerifierPlan{risk_class,required_independence,models[],tools[],blind_fields[],acceptance_rule}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

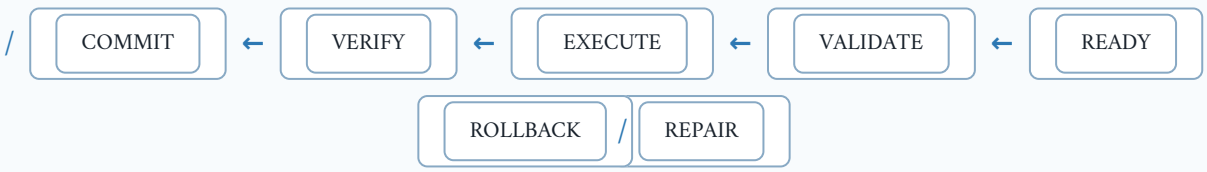
PC-18 · ALGORITHM

استقلال المدقق - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-18 · FAILURE / SECURITY

استقلال المدقق - الفشل والأمن والرصد

أنماط الفشل الحرج

- self_verification_bias
- shared_failure_mode
- context_leak
- rubber_stamp

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: `mission` `component=PC-18` `authority` `input_hash` `decision` `effect` `cost` `latency` `verdict`.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-18 · EVALUATION

استقلال المدقق - المقاييس وE0→E5

المقاييس الأساسية

- independence_index
- false_accept
- false_reject
- verifier_diversity

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-18 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-19

ذاكرة متعددة الطبقات

Memory Tiering v2

> **فجوة الإغلاق.** فصل working/episodic/semantic/procedural/canonical memory ومنع انتقال كل trace مباشرة إلى معرفة مستقبلية.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- working
- episode
- semantic_candidate
- procedure_candidate
- canonical
- quarantine

Invariant

```
PC-19 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-19 · CONTRACT

ذاكرة متعددة الطبقات - العقد والبيانات

Canonical Schema

```
MemoryItem{id,tier,content_ref,source,confidence,scope,expiry,contradictions[],promotion_history[],status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

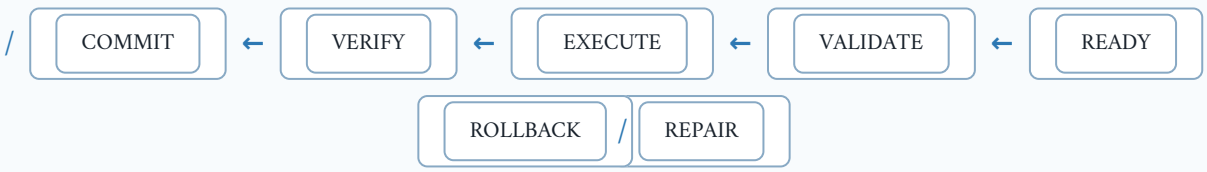
PC-19 · ALGORITHM

ذاكرة متعددة الطبقات - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-19 · FAILURE / SECURITY

ذاكرة متعددة الطبقات - الفشل والأمن والرصد

أنماط الفشل الحرج

- memory_poisoning
- premature_promotion
- stale_canonical
- scope_leak

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: `mission` `component=PC-19` `authority` `input_hash` `decision` `effect` `cost` `latency` `verdict`.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-19 · EVALUATION

ذاكرة متعددة الطبقات - المقاييس وE0→E5

المقاييس الأساسية

- promotion_precision
- retrieval_precision
- stale_memory_error
- quarantine_yield

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-19 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-20

إعادة توطيد الذاكرة

Reconsolidation & Entropy Control

> **فجوة الإغلاق.** قياس التكرار والتناقض والقدم وفقد المصدر ثم إصلاح الذاكرة دون حذف الخلاف أو تاريخ التصحيح.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- dedup
- contradiction_preservation
- source_repair
- staleness
- orphan_detection

Invariant

```
PC-20 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-20 · CONTRACT

إعادة توطيد الذاكرة - العقد والبيانات

Canonical Schema

```
ReconsolidationJob{id,scope,candidates[],duplicate_sets[],contradictions[],repairs[],retained_history,verdict}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

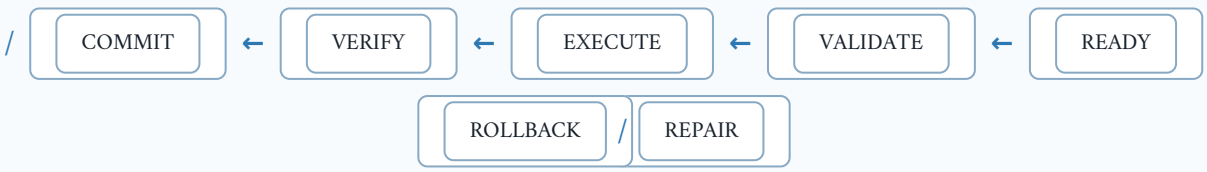
PC-20 · ALGORITHM

إعادة توطيد الذاكرة - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-20 · FAILURE / SECURITY

إعادة توطيد الذاكرة - الفشل والأمن والرصد

أنماط الفشل الحرج

- overcompression
- rare_fact_loss
- conflict_erasure
- history_loss

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission component=PC-20 authority input_hash decision effect cost verdict latency.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-20 · EVALUATION

إعادة توطيد الذاكرة - المقاييس وE0→E5

المقاييس الأساسية

- memory_entropy
- reconstruction_fidelity
- rare_fact_retention
- repair_precision

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-20 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-21

ضغط السياق الحافظ للحاكمية

Governance-Preserving Context Compaction

> **فجوة الإغلاق.** الضغط لا يقلل tokens فقط؛ يجب أن يحفظ السؤال الحاكم والفرق والأدلة والسلطة والقرارات المفتوحة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- stable_contract
- active_differences
- evidence_refs
- open_questions
- authority
- decision_ledger

Invariant

```
PC-21 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-21 · CONTRACT

ضغط السياق الحافظ للحاكمية - العقد والبيانات

Canonical Schema

```
ContextBundle{stable,active_differences,evidence_digest,open_questions,authority,decision_ledger,retrieval_pointers,compression_hash}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

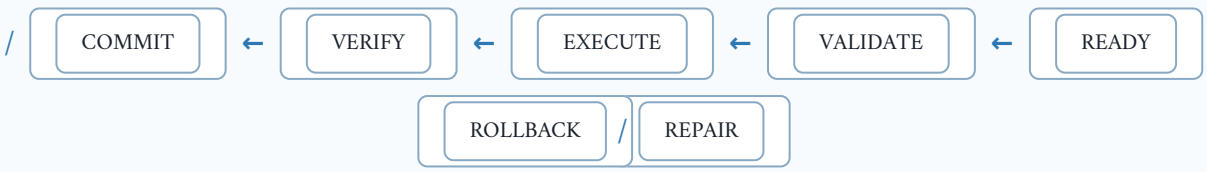
PC-21 · ALGORITHM

ضغط السياق الحافظ للحاكمية - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-21 · FAILURE / SECURITY

ضغط السياق الحافظ للحاكمية - الفشل والأمن والرصد

أنماط الفشل الحرج

- goal_loss
- evidence_detachment
- authority_loss
- false_summary

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-21 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-21 · EVALUATION

ضغط السياق الحافظ للحاكمية - المقاييس وE0→E5

المقاييس الأساسية

- context_fidelity
- decision_replay
- token_savings
- compression_regret

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-21 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-22

تقييم GraphRAG والاسترجاع

Retrieval Evaluation Plane

> **فجوة الإغلاق.** قياس ما إذا كان الاسترجاع يجلب معرفة تغير القرار فعلاً وليس فقط مقاطع مشابهة لغوياً.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- relevance
- novelty
- decision_gain
- source_quality
- redundancy
- latency

Invariant

```
PC-22 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-22 · CONTRACT

تقييم GraphRAG والاسترجاع - العقد والبيانات

Canonical Schema

```
RetrievalEval{query,items[],decision_gain,coverage,redundancy,source_quality,latency,verdict}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

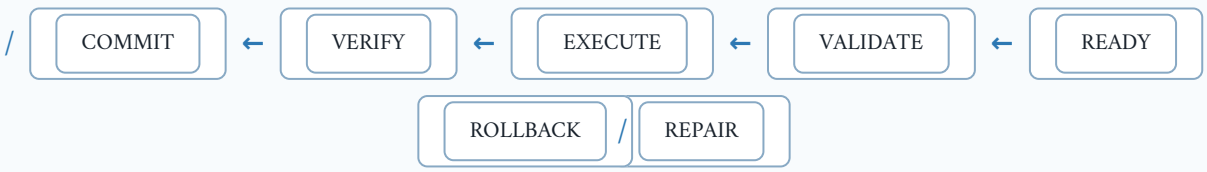
PC-22 · ALGORITHM

تقييم GraphRAG والاسترجاع - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-22 · FAILURE / SECURITY

تقييم GraphRAG والاسترجاع - الفشل والأمن والرصد

أنماط الفشل الحرج

- semantic_echo
- retrieval_bloat
- stale_chunk
- low_decision_gain

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-22 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-22 · EVALUATION

تقييم GraphRAG والاسترجاع - المقاييس وE0→E5

المقاييس الأساسية

- decision_gain_per_token
- source_quality
- redundancy_rate
- freshness_error

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-22 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-23

الاستتباب الدلالي عبر النماذج

Cross-Model Semantic Homeostasis v2

فجوة الإغلاق. منع تراكم الانحراف عند handoffs بين نماذج/providers حتى عندما تمر الـJSON schemas شكلياً.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- canonical_invariants
- round_trip
- contradiction_probe
- semantic_hash
- drift_budget

Invariant

```
PC-23 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-23 · CONTRACT

الاستتباب الدلالي عبر النماذج - العقد والبيانات

Canonical Schema

```
SemanticTransfer{from_model,to_model,state_ref,canonical_hash,roundtrip_hash,drift_score,contradictions[],verdict}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

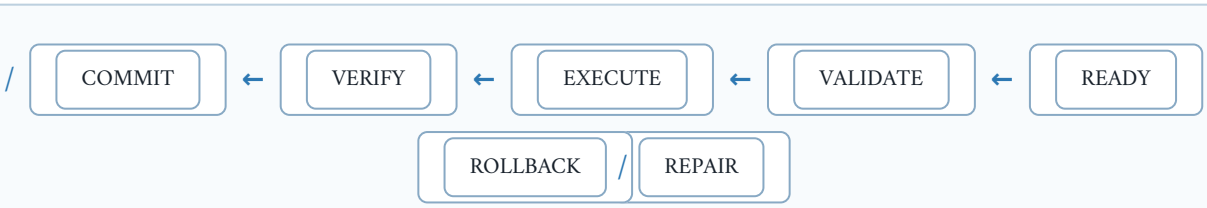
PC-23 · ALGORITHM

الاستنباب الدلالي عبر النماذج - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-23 · FAILURE / SECURITY

الاستتباب الدلالي عبر النماذج - الفشل والأمن والرصد

أنماط الفشل الحرج

- silent_semantic_drift
- schema_valid_semantics_wrong
- provider_bias
- roundtrip_loss

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-23 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-23 · EVALUATION

الاستتباب الدلالي عبر النماذج - المقاييس وE0→E5

المقاييس الأساسية

- semantic_drift
- roundtrip_fidelity
- contradiction_rate
- cross_model_task_delta

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-23 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-24

توجيه الاستدلال متعدد النماذج

Model Capability Router v2

> **فجوة الإغلاق.** الاختيار بين النماذج على أساس قدرة محلية ومعايرة وكلفة ومخاطر وسياق لا على اسم النموذج أو ترتيب ثابت.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- capability_vector
- task_signature
- risk
- context
- latency
- cost
- calibration

Invariant

```
PC-24 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-24 · CONTRACT

توجيه الاستدلال متعدد النماذج - العقد والبيانات

Canonical Schema

```
ModelRoute{task_signature,candidates[],capability_scores[],risk,cost,latency,selected,reason,regret}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

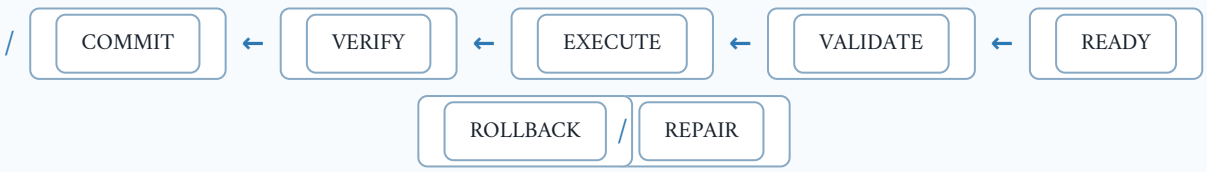
PC-24 · ALGORITHM

توجيه الاستدلال متعدد النماذج - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-24 · FAILURE / SECURITY

توجيه الاستدلال متعدد النماذج - الفشل والأمن والرصد

أنماط الفشل الحرج

- brand_routing
- stale_capability_map
- cheap_model_overuse
- expensive_model_overuse

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: `mission` `component=PC-24` `authority` `input_hash` `decision` `effect` `cost` `latency` `verdict`.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-24 · EVALUATION

توجيه الاستدلال متعدد النماذج - المقاييس وE0→E5

المقاييس الأساسية

- model_route_regret
- cost_quality
- calibration_error
- fallback_success

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-24 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-25

موازنة الحوسبة أثناء التنفيذ

Adaptive Test-Time Compute

> **فجوة الإغلاق.** ربط depth/branching/reflection/verification بـMCD وعدم اليقين والخطر ومكسب المعلومات المتوقع.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- mcd
- uncertainty
- risk
- expected_information_gain
- budget
- plateau

Invariant

```
PC-25 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-25 · CONTRACT

موازنة الحوسبة أثناء التنفيذ - العقد والبيانات

Canonical Schema

```
ComputePolicy{mcd,uncertainty,risk,evi,budget,depth,branches,reflection_points,verification_budget}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

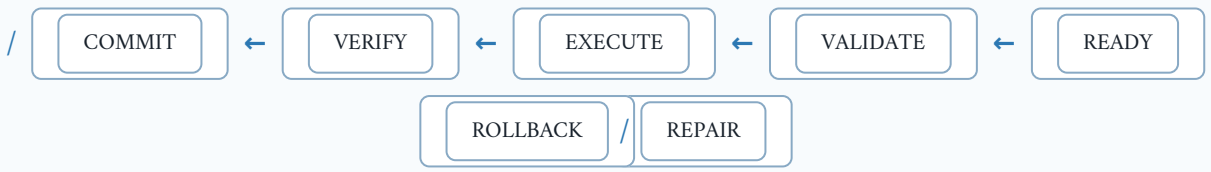
PC-25 · ALGORITHM

موازنة الحوسبة أثناء التنفيذ - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-25 · FAILURE / SECURITY

موازنة الحوسبة أثناء التنفيذ - الفشل والأمن والرصد

أنماط الفشل الحرج

- compute_sprawl
- underthinking
- reflection_loop
- budget_starvation

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-25 · authority · input_hash · decision · effect · cost · latency · verdict.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-25 · EVALUATION

موازنة الحوسبة أثناء التنفيذ - المقاييس وE0→E5

المقاييس الأساسية

- verified_value_per_token
- compute_regret
- plateau_latency
- risk_adjusted_gain

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-25 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-26

الحمل الإبستيمي

Epistemic Load Balancer

> **فجوة الإغلاق.** تخصيص التحقق على الادعاءات ذات الأثر الأعلى أو الدليل الأضعف بدلاً من التحقق بالتساوي.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- claim_impact
- evidence_independence
- uncertainty
- verification_cost
- irreversibility

Invariant

```
PC-26 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-26 · CONTRACT

الحمل الإبستيمي - العقد والبيانات

Canonical Schema

```
VerificationQueue{claims[],priority_scores[],allocated_budget,verifier_assignments,stop_rules}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

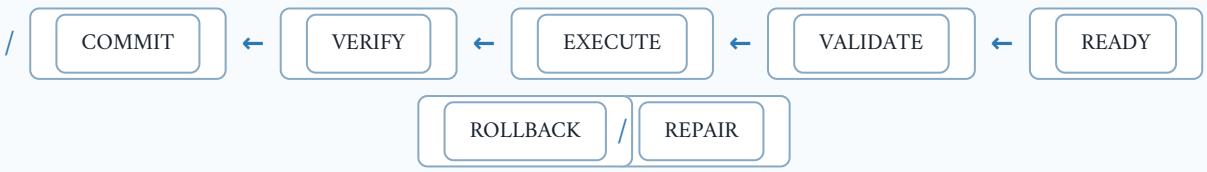
PC-26 · ALGORITHM

الحمل الإبستيمي - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-26 · FAILURE / SECURITY

الحمل الإبستيمي - الفشل والأمن والرصد

أنماط الفشل الحرج

- verify_everything
- verify_nothing
- high_impact_blindspot
- cheap_claim_bias

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-26 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-26 · EVALUATION

الحمل الإبستيمي - المقاييس وE0→E5

المقاييس الأساسية

- verification_efficiency
- critical_claim_coverage
- budget_utilization
- missed_risk

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-26 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-27

إدارة عدم اليقين إلى وكالة

Uncertainty-to-Agency Gate

> **فجوة الإغلاق.** تحويل عدم اليقين إلى سؤال أو تجربة أو تصعيد أو خفض سلطة بدلاً من إخفائه داخل confidence واحد.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- epistemic_uncertainty
- execution_uncertainty
- environmental_uncertainty
- reversibility
- authority

Invariant

```
PC-27 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-27 · CONTRACT

إدارة عدم اليقين إلى وكالة - العقد والبيانات

Canonical Schema

```
AgencyDecision{uncertainty_vector,action_options,experiment_option,escalation_option,authority_ceiling,selected}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

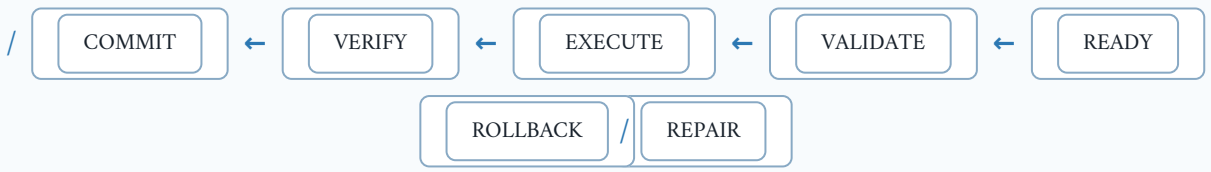
PC-27 · ALGORITHM

إدارة عدم اليقين إلى وكالة - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-27 · FAILURE / SECURITY

إدارة عدم اليقين إلى وكالة - الفشل والأمن والرصد

أنماط الفشل الحرج

- confidence_collapse
- unsafe_guess
- permanent_escalation
- overcautious_block

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-27 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-27 · EVALUATION

إدارة عدم اليقين إلى وكالة - المقاييس وE0→E5

المقاييس الأساسية

- unsafe_guess_rate
- useful_escalation
- experiment_value
- authority_accuracy

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-27 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-28

السياسة ككود

Policy Decision Plane

فجوة الإغلاق. جعل التفويض والخصوصية والـtenant boundaries والالتزامات قابلة للاختبار والإصدار بدلاً من دفنها في prompts.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- principal
- action
- resource
- context
- policy_version
- obligations

Invariant

```
PC-28 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-28 · CONTRACT

السياسة ككود - العقد والبيانات

Canonical Schema

```
PolicyDecision{principal,action,resource,context,policy_version,allow,deny_reason,obligations[],expiry,trace_id}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

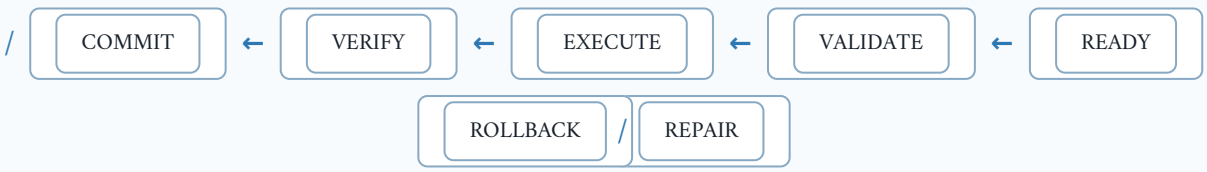
PC-28 · ALGORITHM

السياسة ككود - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-28 · FAILURE / SECURITY

السياسة ككود - الفشل والأمن والرصد

أنماط الفشل الحرج

- prompt_policy_bypass
- policy_drift
- obligation_loss
- tenant_leak

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-28 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-28 · EVALUATION

السياسة ككود - المقاييس وE0→E5

المقاييس الأساسية

- policy_coverage
- decision_latency
- false_allow
- policy_regression

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-28 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-29

عزل المستأجرين والبيانات

Multi-Tenant Sovereignty

فجوة الإغلاق. إغلاق حدود الذاكرة والأدوات والأدلة والمفاتيح بين tenants في SaaS/hybrid deployment.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- tenant_id
- namespace
- key_scope
- data_residency
- retention
- audit

Invariant

```
PC-29 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-29 · CONTRACT

عزل المستأجرين والبيانات - العقد والبيانات

Canonical Schema

```
TenantBoundary{tenant_id,namespaces[],allowed_models[],allowed_tools[],data_region,retention,secrets_scope,audit_policy}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

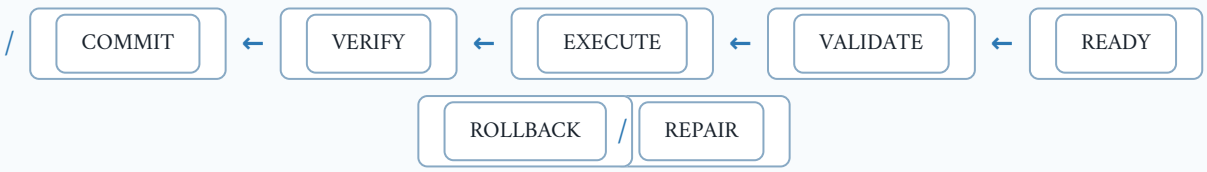
PC-29 · ALGORITHM

عزل المستأجرين والبيانات - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-29 · FAILURE / SECURITY

عزل المستأجرين والبيانات - الفشل والأمن والرصد

أنماط الفشل الحرج

- cross_tenant_leak
- shared_cache_leak
- wrong_secret
- residency_violation

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-29 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-29 · EVALUATION

عزل المستأجرين والبيانات - المقاييس وE0→E5

المقاييس الأساسية

- isolation_violations
- secret_scope_errors
- tenant_trace_separation
- residency_compliance

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-29 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-30

إدارة الأسرار وسلسلة التوريد

Secrets & Supply-Chain Integrity

فجوة الإغلاق. ربط runtime بالأسرار الموقّتة وتوقيع الحزم والتحقق من provenance لكل skill/adapter/tool package.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- secret_lease
- package_signature
- sbom
- provenance
- revocation
- rotation

Invariant

```
PC-30 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-30 · CONTRACT

إدارة الأسرار وسلسلة التوريد - العقد والبيانات

Canonical Schema

```
PackageTrust{id,version,signature,sbom,source_hash,provenance,permissions,revoked,verified_at}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

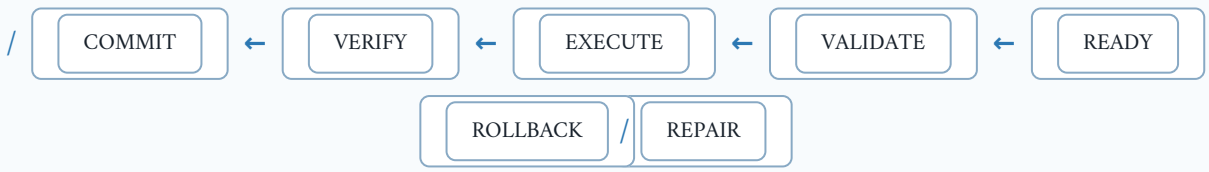
PC-30 · ALGORITHM

إدارة الأسرار وسلسلة التوريد - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-30 · FAILURE / SECURITY

إدارة الأسرار وسلسلة التوريد - الفشل والأمن والرصد

أنماط الفشل الحرج

- secret_exfiltration
- dependency_tamper
- unsigned_skill
- stale_credential

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-30 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-30 · EVALUATION

إدارة الأسرار وسلسلة التوريد - المقاييس وE0→E5

المقاييس الأساسية

- signed_package_rate
- secret_leak_rate
- revocation_latency
- supply_chain_findings

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-30 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-31

الرصد الدلالي

Semantic Observability Plane

فجوة الإغلاق. عدم الاكتفاء CPU/latency؛ التتبع يجب أن يسجل السؤال والفرق والمسار والدليل والسلطة والأثر والذاكرة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- trace_id
- mission_id
- question_id
- difference_ids
- route
- evidence
- policy
- effect
- memory

Invariant

```
PC-31 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-31 · CONTRACT

الرصد الدلالي - العقد والبيانات

Canonical Schema

```
TraceEvent{trace_id,span_id,mission_id,semantic_type,refs[],metrics{},cost,latency,verdict}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً؛ يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event
→ compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

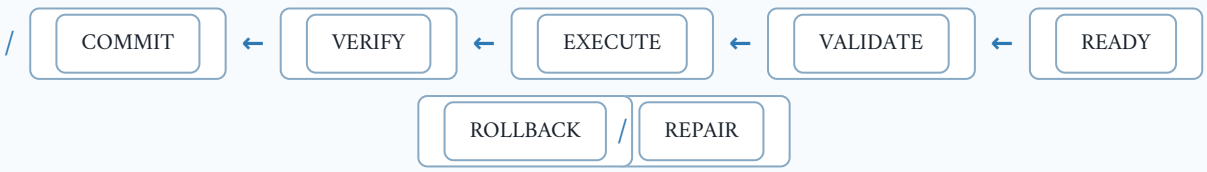
PC-31 · ALGORITHM

الرصد الدلالي - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-31 · FAILURE / SECURITY

الرصد الدلالي - الفشل والأمن والرصد

أنماط الفشل الحرج

- opaque_trace
- missing_semantics
- PII_overlogging
- uncorrelated_effect

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-31 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-31 · EVALUATION

الرصد الدلالي - المقاييس وE0→E5

المقاييس الأساسية

- trace_coverage
- semantic_span_rate
- effect_trace_link
- debug_time

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-31 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-32

إعادة تشغيل الفشل

Failure Replay Laboratory

> **فجوة الإغلاق.** تحويل failures إلى test assets قابلة لإعادة التشغيل والحقن عبر model/tool/environment versions.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- failure_signature
- snapshot
- seed
- fault_injection
- expected_behavior
- repair

Invariant

```
PC-32 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-32 · CONTRACT

إعادة تشغيل الفشل - العقد والبيانات

Canonical Schema

```
FailureCase{id,signature,environment_snapshot,seed,fault,expected_safe_behavior,observed,repair_ref,status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

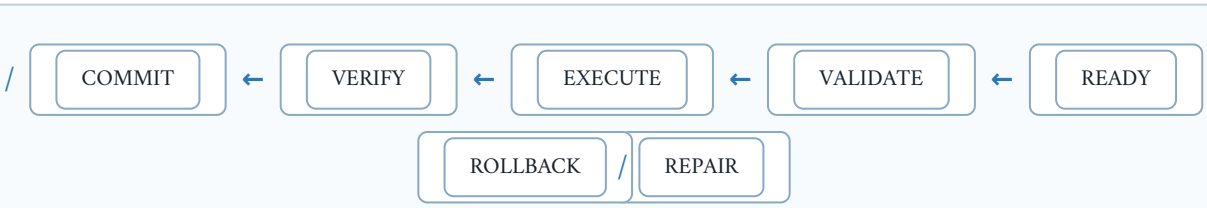
PC-32 · ALGORITHM

إعادة تشغيل الفشل - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-32 · FAILURE / SECURITY

إعادة تشغيل الفشل - الفشل والأمن والرصد

أنماط الفشل الحرج

- irreproducible_failure
- missing_snapshot
- repair_overfit
- fault_mismatch

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-32 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-32 · EVALUATION

إعادة تشغيل الفشل - المقاييس وE0→E5

المقاييس الأساسية

- replay_rate
- repair_transfer
- failure_recurrence
- mean_time_to_repair

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-32 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-33

تصنيف الفشل النظامي

Failure Topology Graph

> **فجوة الإغلاق.** ربط الأعطال بمسارات الانتشار والحلقات الراجعة بدلاً من تصنيفها كأخطاء مكونات منفردة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- nodes
- edges
- propagation
- feedback
- impact
- containment

Invariant

```
PC-33 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-33 · CONTRACT

تصنيف الفشل النظامي - العقد والبيانات

Canonical Schema

```
FailureGraph{nodes[],edges[],motifs[],propagation_scores[],containment_points[],recovery_paths[]}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

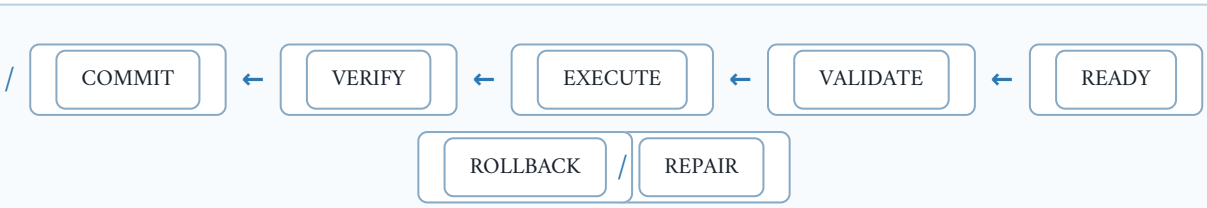
PC-33 · ALGORITHM

تصنيف الفشل النظامي - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-33 · FAILURE / SECURITY

تصنيف الفشل النظامي - الفشل والأمن والرصد

أنماط الفشل الحرج

- local_fix_systemic_fail
- hidden_cascade
- feedback_amplification
- containment_failure

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-33 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-33 · EVALUATION

تصنيف الفشل النظامي - المقاييس وE0→E5

المقاييس الأساسية

- cascade_size
- containment_success
- motif_recurrence
- recovery_latency

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-33 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-34

مختبر التكيف والتدريب

Adaptation Experiment Plane

فجوة الإغلاق. فصل retrieval/prompt/SFT/preferences/LoRA/CPT ومقارنة كل تدخل بأقل بديل كلفة مع حماية النسيان والسلامة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- baseline
- intervention
- train_split
- heldout
- time_split
- regression_suite
- rollback

Invariant

```
PC-34 :: observed_state + explicit_contract + authority + evidence + verification >
                                        model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-34 · CONTRACT

مختبر التكيف والتدريب - العقد والبيانات

Canonical Schema

```
AdaptationRun{id,baseline,intervention,data_manifest,train_split,heldout,metrics,forgetting
,safety,rollback,verdict}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event
→ compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

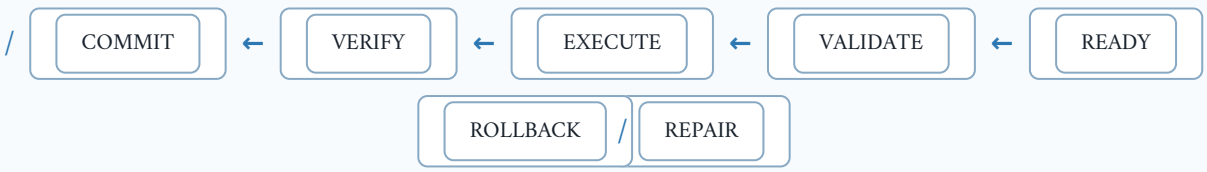
PC-34 · ALGORITHM

مختبر التكيف والتدريب - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-34 · FAILURE / SECURITY

مختبر التكيف والتدريب - الفشل والأمن والرصد

أنماط الفشل الحرج

- leakage
- catastrophic_forgetting
- benchmark_overfit
- unsafe_gain

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-34 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-34 · EVALUATION

مختبر التكيف والتدريب - المقاييس وE0→E5

المقاييس الأساسية

- heldout_gain
- forgetting
- safety_regression
- cost_of_gain

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-34 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-35

ترقية القدرة وحضانتها

Capability Promotion Service

> **فجوة الإغلاق.** منع capability جديدة من الدخول canonical runtime قبل candidate→canary→promotion مع rollback.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- candidate
- evidence
- canary_scope
- promotion_rule
- rollback
- deprecation

Invariant

```
PC-35 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-35 · CONTRACT

ترقية القدرة وحضانتها - العقد والبيانات

Canonical Schema

```
CapabilityLifecycle{id,state,evidence_refs[],canary_scope,metrics,rollback_ref,promotion_history[],status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

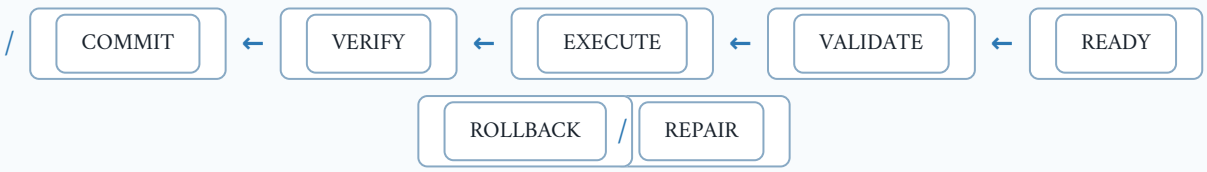
PC-35 · ALGORITHM

ترقية القدرة وحضانتها - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-35 · FAILURE / SECURITY

ترقية القدرة وحضانتها - الفشل والأمن والرصد

أنماط الفشل الحرج

- premature_promotion
- canary_blindspot
- rollback_missing
- zombie_capability

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-35 · authority · input_hash · decision · effect · cost · latency · verdict.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-35 · EVALUATION

ترقية القدرة وحضانتها - المقاييس وE0→E5

المقاييس الأساسية

- promotion_precision
- rollback_success
- canary_detection
- post_promotion_regression

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-35 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-36

ترحيل المخططات

Schema Evolution & Migration

> **فجوة الإغلاق.** جعل KernelState/KnowledgeAtom/Skill/Trace/Memory قابلة للتطور دون كسر replay أو artifacts القديمة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- schema_version
- migration
- backward_read
- forward_write
- compatibility_test

Invariant

```
PC-36 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-36 · CONTRACT

ترحيل المخططات - العقد والبيانات

Canonical Schema

```
SchemaMigration{from_version,to_version,transform,loss_report,compatibility_tests[],rollback,status}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

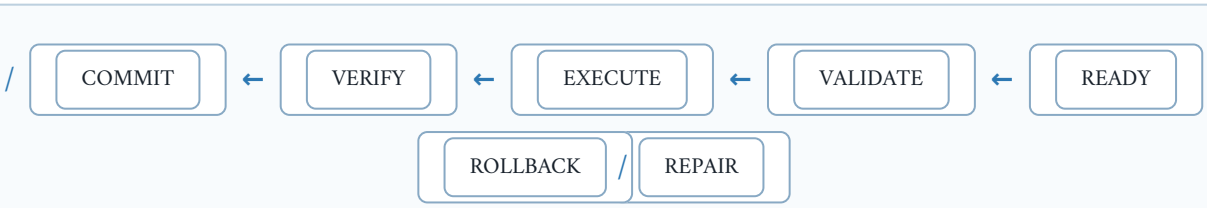
PC-36 · ALGORITHM

ترحيل المخططات - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-36 · FAILURE / SECURITY

ترحيل المخططات - الفشل والأمن والرصد

أنماط الفشل الحرج

- silent_field_loss
- unreplayable_history
- version_fork
- partial_migration

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-36 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-36 · EVALUATION

ترحيل المخططات - المقاييس وE0→E5

المقاييس الأساسية

- migration_fidelity
- backward_read
- replay_after_migration
- loss_rate

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-36 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-37

بصمة البيئة والتطابق العابر للمضيفات

Cross-Environment Conformance

> **فجوة الإغلاق.** قياس ما إذا كان Local/Hermes/SaaS/Hybrid يحافظ على نفس العقود والنتائج المنطقية تحت اختلاف المضيف.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- environment_fingerprint
- capabilities
- policy
- storage
- model
- toolset

Invariant

```
PC-37 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-37 · CONTRACT

بصمة البيئة والتطابق العابر للمضيفات - العقد والبيانات

Canonical Schema

```
EnvironmentFingerprint{id,host,os,runtime,models[],tools[],storage,policy_version,network,capability_hash}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

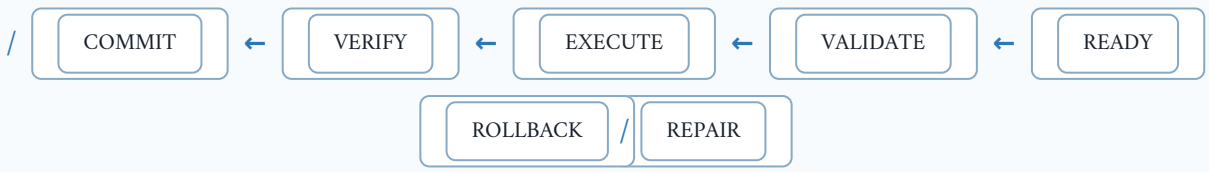
PC-37 · ALGORITHM

بصمة البيئة والتطابق العابر للمضيفات - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-37 · FAILURE / SECURITY

بصمة البيئة والتطابق العابر للمضيفات - الفشل والأمن والرصد

أنماط الفشل الحرج

- host_semantic_drift
- adapter_special_case
- policy_divergence
- storage_inconsistency

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن allowlist + policy gate + evidence للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-37 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-37 · EVALUATION

بصمة البيئة والتطابق العابر للمضيفات - المقاييس وE0→E5

المقاييس الأساسية

- cross_host_fidelity
- environment_regression
- adapter_parity
- policy_parity

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-37 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-38

برنامج Benchmark الموحّد

Benchmark & Baseline Harness

فجوة الإغلاق. إغلاق الفجوة العلمية عبر baselines مجمدة وميزانيات متطابقة وheld-out وseeds وفواصل ثقة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- task_suite
- frozen_baseline
- budget
- seeds
- primary_metric
- failure_threshold
- ci

Invariant

```
PC-38 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-38 · CONTRACT

برنامج Benchmark الموحّد - العقد والبيانات

Canonical Schema

```
BenchmarkRun{id,suite,baseline,treatment,budget,seeds,primary_metric,secondary_metrics,fail_threshold,ci,artifacts}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

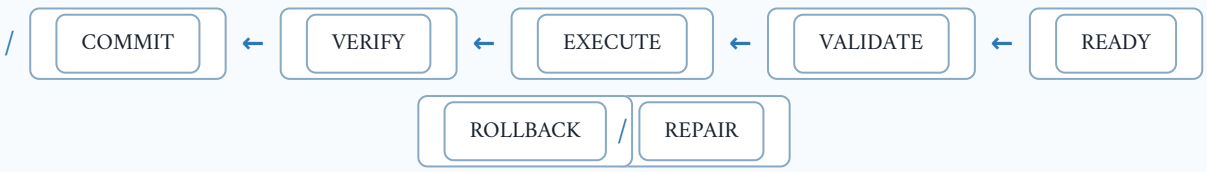
PC-38 · ALGORITHM

برنامج Benchmark الموحّد - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-38 · FAILURE / SECURITY

برنامج Benchmark الموحّد - الفشل والأمن والرصد

أنماط الفشل الحرج

- moving_baseline
- budget_mismatch
- cherry_pick
- single_seed

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission component=PC-38 authority input_hash decision effect cost verdict latency.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-38 · EVALUATION

برنامج Benchmark الموحّد - المقاييس وE0→E5

المقاييس الأساسية

- effect_size
- confidence_interval
- cost_normalized_gain
- critical_regression

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-38 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-39

بوابة E4/E5

Evidence Maturity Gate

> **فجوة الإغلاق.** منع تسمية specification أو demo نجاحاً علمياً؛ E4 تحسن مضبوط وE5 تكرار مستقل عبر نماذج/مجالات/بيئات.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- pre_registered_hypothesis
- primary_metric
- heldout
- replication
- security_nonregression

Invariant

```
PC-39 :: observed_state + explicit_contract + authority + evidence + verification >
                                    model_confidence
```

> **عدم التكرار.** إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-39 · CONTRACT

بوابة E4/E5 - العقد والبيانات

Canonical Schema

```
MaturityDecision{component,current_level,target_level,evidence_refs,criteria,passed,failures[],next_action}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

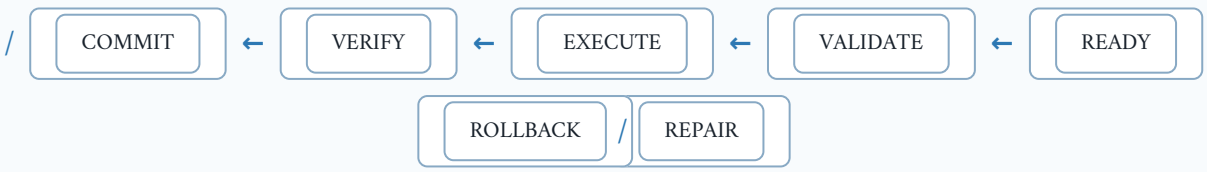
PC-39 · ALGORITHM

بوابة E4/E5 - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-39 · FAILURE / SECURITY

بوابة E4/E5 - الفشل والأمن والرصد

أنماط الفشل الحرج

- demo_as_evidence
- self_replication_only
- unmatched_budget
- security_regression

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-39 · authority · input_hash · decision · effect · cost · latency · verdict.

> **قاعدة فشل:** أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-39 · EVALUATION

بوابة E4/E5 - المقاييس وE0→E5

المقاييس الأساسية

- e4_pass_rate
- e5_replication
- independent_replication
- regression_free_gain

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-39 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في success، grounding، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

PC-40

الشريحة المرجعية الدنيا الكاملة

Minimal End-to-End Reference Runtime

فجوة الإغلاق. بناء أصغر نظام يمر من mission إلى world-effect موثق ثم memory patch، ويصبح الأساس الذي تقاس عليه كل طبقة جديدة.

لماذا توجد هذه الطبقة في Part III؟

Part I عرّف المكوّنات الأساسية، وPart II أضاف MUT/MUG والتقنيات الحدّية. هذه الصفحة تثبت الفرق المتبقي الذي يمنع الانتقال من معمارية غنية إلى Runtime مرجعي قابل للإعادة والقياس. الهدف ليس اختراع اسم جديد، بل جعل المكوّن **قابلاً للتشغيل، للفشل، وللتكرار.**

الكائنات الحاكمة

- mission_contract
- kernel_state
- router
- skill
- loop
- tool
- policy
- court
- effect
- memory

Invariant

```
PC-40 :: observed_state + explicit_contract + authority + evidence + verification > model_confidence
```

عدم التكرار. إذا كان المفهوم موجوداً في CK/EX، فإن Part III لا يعيد تعريفه؛ بل يضيف schema أو state machine أو commit rule أو benchmark gate كان ناقصاً.

حد الادعاء

هذه مواصفة هندسية E1-E3 ما لم توجد نتائج تنفيذ محفوظة. لا تُرفع إلى E4/E5 إلا وفق بوابة التجارب المضبوطة في نهاية الكتاب.

PC-40 · CONTRACT

الشريحة المرجعية الدنيا الكاملة - العقد والبيانات

Canonical Schema

```
ReferenceRun{mission,initial_state,route,plan,tool_receipts,court_verdicts,world_effect,memory_patch,trace,replay_bundle}
```

قواعد صحة البيانات

Ownership
كل كتابة تحدد من يملك الـcanonical field ومن يقترح patch فقط. adapters والنماذج لا تملك الحقيقة المعيارية.

Identity
كل كائن يحمل معرفاً ثابتاً ونسخة ومرجع المهمة والـtrace. النسخة بلا provenance غير قابلة للترقية.

Conflict semantics
التناقض لا يحذف تلقائياً، يسجل كعلاقة ويذهب إلى verifier/court عندما يؤثر في القرار.

Temporal validity
الحالة المتغيرة تحمل زمن القياس وexpiry أو شرط إعادة الفتح؛ stale state لا يندمج بصمت.

قانون التعديل

```
read canonical → propose typed delta → validate schema → validate authority → append event → compute new state hash → commit or reject
```

ممنوع: overwrite غير مفسر، merge بلا lineage، أو تحويل output نصي مباشرة إلى canonical state.

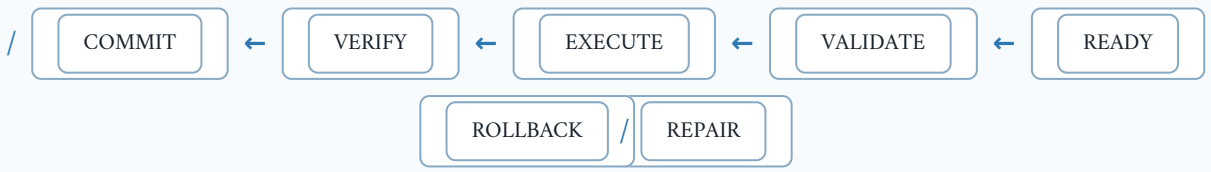
PC-40 · ALGORITHM

الشريحة المرجعية الدنيا الكاملة - الخوارزمية وحالة التشغيل

الخوارزمية المرجعية

```
1. Load mission contract + canonical state
2. Resolve authority and active differences
3. Validate required inputs and freshness
4. Execute the smallest sufficient transition
5. Emit typed receipt / evidence / state delta
6. Verify invariants and expected postconditions
7. Commit event or rollback / quarantine
8. Update trace, metrics and recovery pointer
```

State machine



قواعد التوقف

- تحقق postcondition وانخفاض residual gap تحت العتبة.
- نفاد budget أو اكتشاف عدم صلاحية السلطة.
- ارتفاع uncertainty فوق threshold يفرض سؤالاً أو تصعيداً.
- فشل متكرر مطابق لتوقيع معروف ينقل التنفيذ إلى recovery path بدلاً من إعادة المحاولة العمياء.

تفاعل المكوّن مع بقية النواة

المدخل يأتي دائماً عبر KernelState أو كائن typed مشتق منه؛ المخرج يعود كـdelta أو receipt أو verdict. لا يملك المكوّن قناة جانبية لتعديل الذاكرة أو الأثر الخارجي.

PC-40 · FAILURE / SECURITY

الشريحة المرجعية الدنيا الكاملة - الفشل والأمن والرصد

أنماط الفشل الحرج

- architecture_without_run
- partial_demo
- manual_hidden_step
- nonreplayable_success

Threat model

تهديد	أثره	التحكم المطلوب
Input/Prompt Injection	تحويل نية أو صلاحية المكوّن	فصل instruction عن evidence + policy gate + allowlist للعقود
State Poisoning	تغيير canonical state بمعلومة غير موثقة	append-only event + provenance + court عند التناقض
Replay / Retry	تكرار side effect أو الحكم	idempotency key + version precondition + effect receipt
Cross-boundary leak	تسرب tenant أو secret أو memory	namespace + policy-as-code + scoped credentials

Observability

كل تنفيذ يجب أن ينتج trace يحمل: mission · component=PC-40 · authority · input_hash · decision · effect · cost · latency · verdict.

قاعدة فشل: أي نجاح لا يمكن إعادة بنائه من trace + artifacts لا يُعامل كنجاح مرجعي.

PC-40 · EVALUATION

الشريحة المرجعية الدنيا الكاملة - المقاييس وE0→E5

المقاييس الأساسية

- end_to_end_closure
- manual_step_count
- replay_success
- verified_value_per_cost

تصميم Ablation

Baseline يملك نفس model/tools/task/budget، ثم يعطل PC-40 فقط أو يستبدله بأبسط آلية ممكنة. تقاس النتيجة في grounding، success، cost، safety، recovery، وlatency. إذا لم يظهر الفرق المتوقع، يُخفض المكوّن أو يدمج مع طبقة أبسط.

بوابات النضج

المستوى	المطلوب
E1	تعريف + نطاق + failure boundary.
E2	Schema / contract / artifact قابل للفحص.
E3	Reference implementation + unit/integration/fault tests.
E4	تفوق على baseline مجمد تحت ميزانية متطابقة وتكرار وCI بلا regression حرج.
E5	Replication عبر model family أو domain أو environment مستقل.

Exit Gate. لا تنتقل الطبقة إلى doctrine المعيارية إلا بعد E4، ولا توصف بأنها معممة إلا بعد E5.

A-01

KernelState v3

المخطط المركزي. هذا المخطط هو مرجع تنفيذي؛ الحقول يمكن أن تتوسع بالإصدار لكن لا ينبغي تغيير معناها بصمت.

```
mission_id:string
version:int
goal_contract:ref
governing_question:ref
active_differences:[ref]
claims:[ref]
evidence:[ref]
authority:ref
route:ref
plan:ref
world_before:ref
world_after:ref
courts:[ref]
memory_delta:ref
budget:object
recovery_pointer:ref
```

قواعد عامة

- كل ref يجب أن يحل إلى artifact أو event موجود.
- كل mutation ينتج version أو event جديداً.
- الحقول الحساسة تحمل scope وretention.
- unknown لا يحول إلى قيمة افتراضية مخفية.

> الهدف من هذه المخططات هو جعل Part III قابلاً للتحويل إلى JSON Schema / Pydantic / Protobuf دون إعادة اختراع semantics.

A-02

KnowledgeAtom v2

الذرة المعرفية. هذا المخطط هو مرجع تنفيذي؛ الحقول يمكن أن تتوسع بالإصدار لكن لا ينبغي تغيير معناها بصمت.

```
id:string
kind:enum
claim:text/ref
scope:object
source_ref:uri/ref
valid_from:time
valid_to:time?
confidence:number
sensitivity:enum
dependencies:[ref]
supersedes:[ref]
contradictions:[ref]
retrieval_cues:[string]
status:enum
```

قواعد عامة

- كل ref يجب أن يحل إلى artifact أو event موجود.
- كل mutation ينتج version أو event جديداً.
- الحقول الحساسة تحمل scope وretention.
- unknown لا يحول إلى قيمة افتراضية مخفية.

> الهدف من هذه المخططات هو جعل Part III قابلاً للتحويل إلى JSON Schema / Pydantic / Protobuf دون إعادة اختراع semantics.

A-03

EffectTransaction

معاملة الأثر. هذا المخطط هو مرجع تنفيذي؛ الحقول يمكن أن تتوسع بالإصدار لكن لا ينبغي تغيير معناها بصمت.

```
id:string
idempotency_key:string
action:ref
resource:ref
precondition_hash:string
lease:object
attempts:[ref]
receipt:ref?
verified_postcondition:ref?
compensation:ref?
status:enum
```

قواعد عامة

- كل ref يجب أن يحل إلى artifact أو event موجود.
- كل mutation ينتج version أو event جديداً.
- الحقول الحساسة تحمل scope وretention.
- unknown لا يحول إلى قيمة افتراضية مخفية.

> الهدف من هذه المخططات هو جعل Part III قابلاً للتحويل إلى JSON Schema / Pydantic / Protobuf دون إعادة اختراع semantics.

A-04

PolicyDecision

قرار السياسة. هذا المخطط هو مرجع تنفيذي؛ الحقول يمكن أن تتوسع بالإصدار لكن لا ينبغي تغيير معناها بصمت.

```
principal:ref
action:string
resource:ref
context:object
policy_version:string
allow:boolean
deny_reason:string?
obligations:[object]
expiry:time?
trace_id:string
```

قواعد عامة

- كل ref يجب أن يحل إلى artifact أو event موجود.
- كل mutation ينتج version أو event جديداً.
- الحقول الحساسة تحمل scope وretention.
- unknown لا يحول إلى قيمة افتراضية مخفية.

الهدف من هذه المخططات هو جعل Part III قابلاً للتحويل إلى JSON Schema / Pydantic / Protobuf دون إعادة اختراع semantics.

A-05

MemoryEvent

حدث الذاكرة. هذا المخطط هو مرجع تنفيذي؛ الحقول يمكن أن تتوسع بالإصدار لكن لا ينبغي تغيير معناها بصمت.

```
id:string
tier:enum
source_event:ref
content_ref:ref
scope:object
confidence:number
promotion_candidate:boolean
contradictions:[ref]
expiry:time?
status:enum
```

قواعد عامة

- كل ref يجب أن يحل إلى artifact أو event موجود.
- كل mutation ينتج version أو event جديداً.
- الحقول الحساسة تحمل scope وretention.
- unknown لا يحول إلى قيمة افتراضية مخفية.

> الهدف من هذه المخططات هو جعل Part III قابلاً للتحويل إلى JSON Schema / Pydantic / Protobuf دون إعادة اختراع semantics.

A-06

TraceEvent

حدث التتبع. هذا المخطط هو مرجع تنفيذي؛ الحقول يمكن أن تتوسع بالإصدار لكن لا ينبغي تغيير معناها بصمت.

```
trace_id:string
span_id:string
parent_span:string?
mission_id:string
semantic_type:enum
refs:[ref]
metrics:map
cost:number
latency_ms:number
verdict:enum
```

قواعد عامة

- كل ref يجب أن يحل إلى artifact أو event موجود.
- كل mutation ينتج version أو event جديداً.
- الحقول الحساسة تحمل scope وretention.
- unknown لا يحول إلى قيمة افتراضية مخفية.

> الهدف من هذه المخططات هو جعل Part III قابلاً للتحويل إلى JSON Schema / Pydantic / Protobuf دون إعادة اختراع semantics.

A-07

CapabilityManifest

سجل القدرة. هذا المخطط هو مرجع تنفيذي؛ الحقول يمكن أن تتوسع بالإصدار لكن لا ينبغي تغيير معناها بصمت.

```
id:string
version:string
dependencies:[ref]
critical_edges:[ref]
authority_ceiling:enum
entry_contract:ref
exit_contract:ref
tests:[ref]
rollback_ref:ref
maturity:E0..E5
```

قواعد عامة

- كل ref يجب أن يحل إلى artifact أو event موجود.
- كل mutation ينتج version أو event جديداً.
- الحقول الحساسة تحمل scope وretention.
- unknown لا يحول إلى قيمة افتراضية مخفية.

الهدف من هذه المخططات هو جعل Part III قابلاً للتحويل إلى JSON Schema / Pydantic / Protobuf دون إعادة اختراع semantics.

A-08

ExperimentRecord

سجل التجربة. هذا المخطط هو مرجع تنفيذي؛ الحقول يمكن أن تتوسع بالإصدار لكن لا ينبغي تغيير معناها بصمت.

```
id:string
hypothesis:text
baseline:ref
treatment:ref
task_suite:ref
budget:object
seeds:[int]
primary_metric:string
secondary_metrics:[string]
failure_threshold:object
ci:object
artifacts:[ref]
verdict:enum
```

قواعد عامة

- كل ref يجب أن يحل إلى artifact أو event موجود.
- كل mutation ينتج version أو event جديداً.
- الحقول الحساسة تحمل scope وretention.
- unknown لا يحول إلى قيمة افتراضية مخفية.

> الهدف من هذه المخططات هو جعل Part III قابلاً للتحويل إلى JSON Schema / Pydantic / Protobuf دون إعادة اختراع semantics.

ملحق الأخطاء المعيارية 1

Code	الفئة	التعريف
E-AUTH	فشل السلطة	action غير مسموح أو scope غير صحيح.
E-SOURCE	فشل المصدر	مصدر مفقود/قديم/غير مستقل/مسموم.
E-SCHEMA	فشل المخطط	كائن غير صالح أو migration ناقص.
E-TOOL	فشل الأداة	timeout أو malformed output أو capability mismatch.

قاعدة Error Envelope

```
ErrorEnvelope{code,component,mission_id,trace_id,stage,severity,cause,evidence_refs,retryable,recovery_action,world_effect_risk}
```

الخطأ ليس exception نصية فقط؛ يجب أن يتحول إلى كائن يسمح بالتوجيه إلى retry أو repair أو escalation أو rollback.

ملحق الأخطاء المعيارية 2

Code	الفئة	التعريف
E-EFFECT	فشل الأثر	receipt موجود لكن postcondition غير محقق.
E-LOOP	فشل الحلقة	plateau/budget exhaustion/unbounded recurrence.
E-MEM	فشل الذاكرة	premature promotion/stale canonical/scope leak.
E-COURT	فشل المحكمة	verifier capture/circular evidence/nonbinding verdict.

قاعدة Error Envelope

```
ErrorEnvelope{code,component,mission_id,trace_id,stage,severity,cause,evidence_refs,retryable,recovery_action,world_effect_risk}
```

الخطأ ليس exception نصية فقط؛ يجب أن يتحول إلى كائن يسمح بالتوجيه إلى retry أو repair أو escalation أو rollback.

ملحق الأخطاء المعيارية 3

Code	الفئة	التعريف
E-ROUTE	فشل التوجيه	route regret عالٍ أو unsafe route.
E-SEC	فشل الأمن	injection/secret leak/tenant breach/package tamper.
E-STATE	فشل الحالة	fork/lost update/stale canonical state.
E-REPLAY	فشل الإعادة	run ناجح لا يمكن إعادة تشغيله أو تفسيره.

قاعدة Error Envelope

```
ErrorEnvelope{code,component,mission_id,trace_id,stage,severity,cause,evidence_refs,retryable,recovery_action,world_effect_risk}
```

الخطأ ليس exception نصية فقط؛ يجب أن يتحول إلى كائن يسمح بالتوجيه إلى retry أو repair أو escalation أو rollback.

Threat Model 1

Prompt / Instruction Injection

فصل instruction channels، provenance، policy, allowlist.

Retrieval Poisoning

source ranking، signatures، contradiction scans، quarantine.

Tool Output Injection

treat output as data، schema validation، no implicit instruction.

Agent Impersonation

identity binding، signed AgentCard/session، trust class.

الأصل: كل تهديد يختبر بالـfault injection أو red-team case؛ وجود policy في الوثيقة ليس دليلاً على صلابتها.

Threat Model 2

Replay Side Effects

idempotency key، preconditions، receipts، compensation.

Secrets Exfiltration

short-lived credentials، scoped injection، redaction.

Cross-Tenant Leakage

namespaces، policy-as-code، cache partitioning.

Supply-Chain Tampering

signatures، SBOM، provenance، revocation.

الأصل: كل تهديد يختبر بالـfault injection أو red-team case؛ وجود policy في الوثيقة ليس دليلاً على صلابتها.

Threat Model 3

Model Endpoint Impersonation

.TLS/pinning/provider registry/attestation where available

Memory Poisoning

.candidate tier، promotion court، rollback

Policy Drift

.version pinning، regression suite، audit

Semantic Drift

.round-trip tests، canonical invariants، contradiction probes

الأصل: كل تهديد يختبر بالـfault injection أو red-team case؛ وجود policy في الوثيقة ليس دليلاً على صلابتها.

سجل التجارب 1-4

ID	التجربة	التصميم
EXP-01	KernelState convergence	تعطيل version/event discipline وقياس forks.
EXP-02	KnowledgeAtom provenance	إزالة provenance ومقارنة false promotion.
EXP-03	Evidence→Authority separation	confidence-only baseline مقابل policy gate.
EXP-04	Exactly-once effects	حقن crash/retry في mutation side effects.

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 5-8

ID	التجربة	التصميم
EXP-05	Bounded HyperLoop	fault injection + plateau + budget exhaustion.
EXP-06	Sparse router	semantic-only baseline مقابل hybrid+pruning.
EXP-07	Skill compiler	prompt skill مقابل compiled contract.
EXP-08	Ability closure	skills ناجحة منفردة مقابل dependency closure.

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 9-12

ID	التجربة	التصميم
EXP-09	Mission DAG	hidden dependencies وresume.
EXP-10	Durable workflow	process crash ثم replay.
EXP-11	Remote agent boundary	malicious/overclaim remote agent.
EXP-12	Tool transport parity	native vs MCP same contract.

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 13-16

ID	التجربة	التصميم
EXP-13	World-effect verifier	tool receipt vs independent readback.
EXP-14	Causal attribution	intervention on contributor nodes.
EXP-15	Evidence geometry	duplicate-root sources vs independent sources.
EXP-16	Court engine	self-verifier vs independent verifier.

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 17-20

ID	التجربة	التصميم
EXP-17	Verifier independence	shared model/context ablations.
EXP-18	Memory tiers	direct canonical write vs promotion ladder.
EXP-19	Reconsolidation	dedup while preserving rare contradictions.
EXP-20	Context compaction	token reduction vs decision replay fidelity.

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 21-24

ID	التجربة	التصميم
EXP-21	Retrieval decision gain	top-k relevance vs decision-gain ranking.
EXP-22	Semantic homeostasis	multi-hop model handoffs.
EXP-23	Model router	fixed best model vs capability routing.
EXP-24	Adaptive compute	fixed budget vs EVI/MCD policy.

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 25-28

ID	التجربة	التصميم
EXP-25	Epistemic load	uniform verification vs impact-aware allocation.
EXP-26	Uncertainty-to-agency	guess vs ask/experiment/escalate.
EXP-27	Policy-as-code	prompt policy vs PDP/PEP.
EXP-28	Tenant isolation	cache/memory/secret cross-tenant attacks.

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 29-32

ID	التجربة	التصميم
EXP-29	Supply chain	.tampered skill/adapter package
EXP-30	Semantic observability	.generic logs vs typed semantic traces
EXP-31	Failure replay	.real failure reproduced across versions
EXP-32	Failure topology	.single-node repair vs propagation-aware repair

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 33-36

ID	التجربة	التصميم
EXP-33	Adaptation	prompt/retrieval/SFT/LoRA matched budget.
EXP-34	Promotion service	direct release vs canary/rollback.
EXP-35	Schema migration	old runs replay after v2→v3 migration.
EXP-36	Cross-environment parity	local/Hermes/SaaS/hybrid same mission.

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

سجل التجارب 37-40

ID	التجربة	التصميم
EXP-37	Benchmark harness	.moving baseline vs frozen baseline
EXP-38	E4 gate	.matched-budget repeated seeds and CI
EXP-39	E5 replication	.different model family and environment
EXP-40	Minimal Complete Slice	.mission→effect→memory→replay with zero manual hidden steps

قاعدة موحدة: frozen baseline + matched budget + repeated seeds + held-out tasks + cost/latency + critical regression gates.

هيكل Monorepo المرجعي

```
moussaifxt/
  contracts/
    kernel_state/
    knowledge_atom/
    policy/
    tools/
    memory/
    traces/
  kernel/
    question/
    differences/
    router/
    courts/
    world_effect/
  runtime/
    loops/
    dag/
    durable/
    transactions/
  adapters/
    native/
    mcp/
    a2a/
    hermes/
    local/
  stores/
    state/
    events/
    memory/
    evidence/
  eval/
    baselines/
    replay/
    benchmarks/
    redteam/
  deploy/
    local/
    saas/
    hybrid/
  tests/
    unit/
    contract/
    integration/
    fault/
    security/
    e4/
    e5/
```

الهيكل يفصل contracts عن runtime وعن adapters حتى لا يصبح transport أو provider جزءاً من semantics.

مصفوفة الاختبارات

النوع	المطلوب	مثال
Unit	دوال وحسابات محلية	MCD normalization, schema validators
Contract	تعامل المكونات عبر schemas	ToolRequest/ToolReceipt
Integration	مسارات متعددة	DAG→Policy→Tool→Verifier
Fault	حقن الأعطال	timeout, stale data, crash, duplicate retry
Security	حقن خبيث وحدود الثقة	prompt injection, secret leak
Replay	إعادة أحداث محفوظة	failure replay and deterministic resume
E4	baseline مضبوط	matched budget repeated seeds
E5	تكرار مستقل	cross-model/domain/environment

خارطة التنفيذ - Sprint 12

Sprint	الناتج
01	KernelState v3 + Event Log + schemas.
02	Policy/Authority + KnowledgeAtom + EvidenceGraph.
03	Tool Gateway + EffectTxn + idempotency.
04	Bounded HyperLoop + durable checkpoints.
05	Sparse Router + model capability registry.
06	WorldEffect verifier + causal credit.
07	Memory tiers + reconsolidation + compaction.
08	Courts + verifier independence + evidence geometry.
09	Observability + failure replay + threat suite.
10	Local/Hermes/SaaS/hybrid conformance.
11	Benchmark harness + frozen baselines + 40 experiments.
12	Minimal Complete Slice E3 → first E4 candidates.

Definition of Done - Runtime مرجعي

- لا side effect بلا PolicyDecision وidempotency key.
- لا نجاح بلا independent world-state readback عندما يكون الأثر خارجياً.
- لا canonical memory write بلا provenance وpromotion path.
- لا agent/model يملك canonical state مباشرة.
- لا runtime transition بلا trace/event.
- كل failure عالي الأثر قابل للإعادة من snapshot/seed/artifacts.
- كل schema له version وmigration test.
- كل capability لها rollback وmaturity level.
- كل benchmark له frozen baseline وbudget متطابق.
- أي ادعاء E4/E5 يملك artifact قابل للمراجعة.

```
Done ≠ code runs ; Done = reproducible governed verified runtime
```

نموذج End-to-End Trace

```
MISSION M-2026-001
  → Contract C-1
  → KernelState v3 #41
  → GoverningQuestion Q-7
  → ActiveDifference Δ-12
  → Route R-9 (model+skill+tool)
  → PolicyDecision P-44 ALLOW + obligations
  → EffectTxn T-91 key=...
  → ToolReceipt TR-91
  → IndependentReadback WR-22
  → CourtVerdict CV-8 VERIFIED
  → WorldEffect WE-5 residual_gap=0
  → MemoryCandidate MC-19
  → PromotionVerdict HOLD
  → KernelState v3 #42
  → ReplayBundle RB-5
```

هذا هو المستوى الذي يجب أن يستطيع النظام إنتاجه آلياً. إذا احتاج reviewer إلى تخمين ما حدث بين مرحلتين، لم نصل بعد إلى الإغلاق التنفيذي.

خاتمة Part III

بعد Part I وPart II كان النظام يملك خريطة نواة غنية وامتدادات تقنية حديثة، لكن الخطر بقي أن تبقى العلاقات بين الطبقات وصفية. Part III يقفل هذه الفجوة على مستوى المواصفة: 40 مكوّن schemas، Proof-Closure موحدة، exactly-once effects، state/event semantics، benchmarks، failure replay، policy boundaries، memory promotion وبوابات E4/E5.

> **الحالة الصحيحة الآن:** يمكن وصف MOUSSAIFXT كـReference Architecture قابلة للتحويل إلى Runtime مرجعي. لا يجوز وصفه كنظام AGI مثبت قبل تنفيذ الشريحة الدنيا وإكمال التجارب.

الهدف التالي

ليس Part IV نظرياً جديداً، بل repository تنفيذي وBenchmark Book يولدان من التشغيل الحقيقي: ablations، failures، traces، World-Effect measurements، cost curves، calibration وreplications.

```
Next canonical artifact = Runtime Repo + Benchmark/Evidence Laboratory
```