Independent Strategic Architecture Brief

From Sovereign Compute to Sovereign Execution

A Control-Plane Architecture for Saudi Arabia's National AI Infrastructure

يمكن للذكاء أن يكون عالميًا، أما التنفيذ فيجب أن يبقى سياديًا.

Intelligence can be global. Execution must be sovereign.

An independent reference architecture defining the control-plane pathway for governed autonomous AI across Saudi AI cloud operations, digital government, smart cities, regulated sectors, and AI software factories.

Dr. Jun He
The OpenKedge Initiative

Strategic White Paper · May 2026

openkedge.io

Contents

Executive Brief: The Kingdom's Next AI Control Layer

Core Thesis

Saudi Arabia is rapidly building the sovereign AI stack: data centers, cloud platforms, national data infrastructure, Arabic and domain-specific models, digital government systems, and smart-city operating environments. The next strategic layer is sovereign execution: the ability to govern what autonomous AI systems are allowed to do, under which policy, using which identity, with what evidence, and with what replayable accountability.

Saudi Arabia[1] is moving from AI strategy into AI infrastructure: data centers, cloud platforms, national data systems, Arabic models, digital government, and smart-city operating environments. As those foundations mature, the governance question moves up the stack. The issue is not only where AI runs, but how AI-driven actions are admitted, authorized, bounded, executed, recorded, and replayed.

A model response is advisory. A cloud change, infrastructure deployment, data access, workflow approval, or smart-city operation is consequential. Compute, data localization, and domestic models remain necessary foundations; sovereign execution adds the runtime control layer that binds high-impact action to local policy, bounded identity, execution contracts, evidence, and replay.

Practical Governance Questions

- Who authorized this action?
- Which policy allowed it?
- What data or context did the model see?
- What identity executed the action?
- Was the action bounded?
- Can the decision be audited and replayed?
- Can a regulator, operator, or ministry prove why the action was allowed?

Executive Decision Frame

For senior decision-makers, platform leaders, and procurement teams, the architecture helps answer five practical decisions:

- which autonomous workflows are ready for governed pilots;
- which actions require human approval, policy checks, or operational limits;
- how domestic and global models can be used without granting execution authority to the model itself;
- what evidence vendors and integrators can produce before, during, and after execution;
- how generated code, workflows, and infrastructure changes can be admitted before production.

Definition: Sovereign Execution

Sovereign execution is the ability of a nation, public institution, or regulated enterprise to govern what autonomous AI systems are allowed to do, under which policy, using which identity, with what evidence, and with what replayable accountability.

[1] This brief is independent: it describes a reference architecture, not an adoption claim by any Saudi institution or vendor. Its starting point is Saudi Arabia's visible AI infrastructure momentum and the practical question of how that momentum becomes governed autonomous capability.

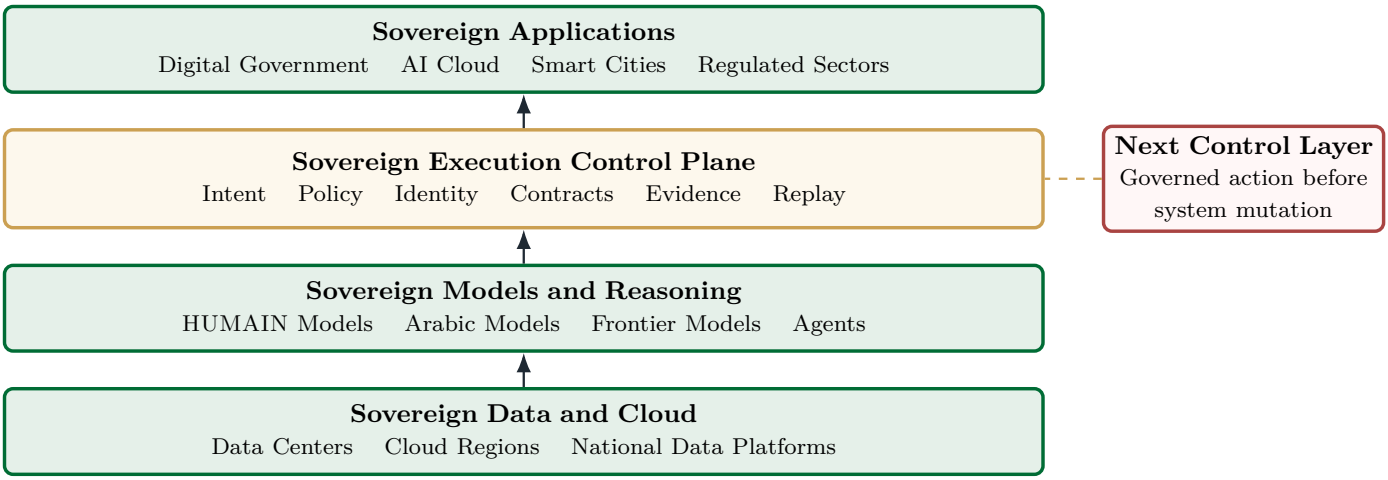

Figure 1: *From sovereign compute to sovereign execution. Sovereign data, cloud, and models are necessary foundations; high-impact autonomous actions are best governed through a dedicated execution control plane that manages intent, policy, identity, contracts, evidence, and replay.*

Sovereign Execution Principle

Intelligence can be global. Execution must be sovereign.

The Strategic Context

The Saudi AI landscape is moving from strategy to live infrastructure. HUMAIN is positioned as a full-stack AI ecosystem spanning data centers, cloud, models, and applications [9]. SDAIA's National Data Lake is positioned as a national data foundation for integrated data access and governance [10]. DGA drives digital government, and NEOM provides a prominent smart-city and digital operating-environment context [2, 8].

As compute and data capacity grow, a shared execution-governance layer becomes practical. It provides a single way to govern AI-driven actions across ministries, cloud platforms, and regulated sectors.

The Next Control Layer

Much AI governance still focuses on models: safety alignment, prompt filtering, benchmark performance, content moderation, and responsible AI review. These disciplines matter, and agentic systems add an execution question.

Autonomous agents may request cloud configuration changes, modify infrastructure-as-code, approve workflows, route citizen cases, access sensitive data, trigger smart-city operations, or deploy generated software. Traditional security is often designed around deterministic human or service callers. Agentic AI can be probabilistic, multi-step, delegated, recursive, or produced by a chain of models and tools. Static permissions and passive logs can leave governance gaps in that operating model.

Execution Boundary

AI agents should submit structured intent, not receive direct write access.

The Proposed Architecture

Together, the paper's five layers define a closed loop: Reasoning → Structured Intent → Context Binding → Policy Evaluation → Risk / Approval → Execution Contract → Ephemeral Identity → Controlled Execution → Evidence Chain → Replay / Audit → Governance Feedback.

Operating Rule

Models may reason. Sovereign control planes execute.

Table 1: *Sovereign execution architecture layers.*

Layer	Role	Decision-maker relevance
SAL	Separates reasoning from execution.	Allows global or domestic reasoning without granting execution authority.
ASCP	Provides the macro control-plane architecture.	Routes autonomous actions through policy, identity, contracts, evidence, and replay.
OpenKedge	Provides the open intent-governance protocol.	Standardizes the execution boundary across vendors, models, and platforms.
VAI	Provides evidence, identity, audit, and replay.	Supports audit, incident response, regulator review, and operational learning.
PDD	Governs generated artifacts before deployment.	Applies protocol admissibility to AI-generated code, workflows, and infrastructure-as-code.

Why This Matters for KSA Decision-Makers

Table 2: *Decision-maker relevance for sovereign execution.*

Institutional Context	Sovereign Execution Relevance
HUMAIN	Can govern autonomous AI cloud operations, GPU infrastructure, model-serving platforms, infrastructure-as-code, and agent marketplaces.
SDAIA	Can provide a reusable control model for national data and AI governance, including policy-bound access, minimized context, cross-agency evidence, and replayable accountability.
DGA	Can support autonomous public administration patterns: citizen-service workflows, permit routing, document verification, inter-agency orchestration, and government service automation.
NEOM and smart-city operators	Can define the boundary between digital-twin reasoning and real-world execution.
Regulated sectors	Can offer a common execution-governance layer with sector-specific policy packs for healthcare, finance, energy, and logistics.
Saudi AI software factories	Can govern AI-generated code, workflows, and infrastructure configurations before they enter production.

The architectural opportunity is not one governed AI application. It is a reusable execution-governance layer for the Kingdom's AI economy.

Strategic Value

Sovereign execution can accelerate autonomous AI deployment while preserving control. It reduces lock-in through open protocol boundaries, replaces permanent agent privilege with short-lived authority, and gives local integrators and Saudi startups a clear protocol target.

Strategic Recommendation

This paper recommends treating sovereign execution as a national AI infrastructure layer, alongside sovereign compute, sovereign data, cybersecurity, identity, and digital government platforms.

The next global AI leader will not simply be the nation with the largest models or the most GPUs. It will be the nation that can safely allow AI to act. Saudi Arabia has the opportunity to define that blueprint.

1 Saudi Arabia's AI Infrastructure Moment

Chapter Thesis

Saudi Arabia is moving from AI ambition to AI infrastructure. Data centers, cloud platforms, national data systems, domestic models, digital government programs, and smart-city operating environments are becoming the foundation of the Kingdom's AI economy. The strategic question is no longer only where AI runs, but how autonomous AI actions will be governed once they begin operating across that infrastructure.

Saudi Arabia is moving from AI strategy toward operational capacity: data centers, cloud regions, national data systems, models, and smart-city backbones. Designating 2026 as the Year of Artificial Intelligence signals this shift [12]. Under Vision 2030, AI is treated as core infrastructure for the local economy.

That infrastructure creates the foundation for autonomous AI across cloud platforms, government workflows, regulated sectors, and physical environments. As the foundation matures, the control question moves up the stack: not only where AI runs, but how AI-driven actions are admitted, authorized, bounded, executed, recorded, and replayed.

In other words: sovereign compute makes national AI capacity possible. Sovereign execution makes national AI action governable.

1.1 HUMAIN and the Full-Stack AI Ecosystem

HUMAIN serves as a primary national infrastructure anchor: a full-stack AI ecosystem bridging data centers, cloud, and models [9]. A full-stack environment is not just a place to host applications; it is an operational surface where autonomous systems may affect cluster scaling, model serving, network posture, deployment workflows, and service continuity.

A full-stack AI ecosystem is well served by a full-stack execution-governance layer: one that routes high-impact action through intent intake, policy evaluation, contract-bound credentials, evidence chains, and replay.

KSA relevance: HUMAIN

HUMAIN represents the move from AI aspiration to AI infrastructure. Sovereign execution gives such infrastructure a governance path for autonomous operations: intent intake, policy evaluation, contract-bound credentials, evidence chains, and replayable audit.

1.2 AWS/HUMAIN AI Zone and Hyperscaler Partnerships

Hyperscaler partnerships, such as the AWS/HUMAIN AI Zone, show the scale of Saudi Arabia's infrastructure ambition [1]. They also make interoperability a design requirement. A vendor-agnostic control layer allows domestic models, hyperscaler services, and future agent platforms to share one governance boundary.

Sovereign execution lets Saudi institutions benefit from hyperscaler platforms while keeping high-impact action governance under local policy, approval paths, bounded execution identity, and evidence Saudi operators can inspect.

1.3 SDAIA and the National Data Lake

SDAIA's National Data Lake is positioned as a national data foundation for integrated data access and governance [10, 11]. At that scale, governance extends beyond access. If an AI system can reason over approved data and then trigger a workflow, modify a record, or pass context to another agency, the downstream action also benefits from governance.

Data governance can evolve into data-plus-execution governance through context minimization, policy-filtered task views, and separation between what a model may see and what a sovereign system may execute.

1.4 DGA and Digital Government Transformation

DGA is the digital government anchor for autonomous public administration. Candidate workflows include citizen-service routing, permits, document verification, inter-agency orchestration, case management, and eligibility workflows [2].

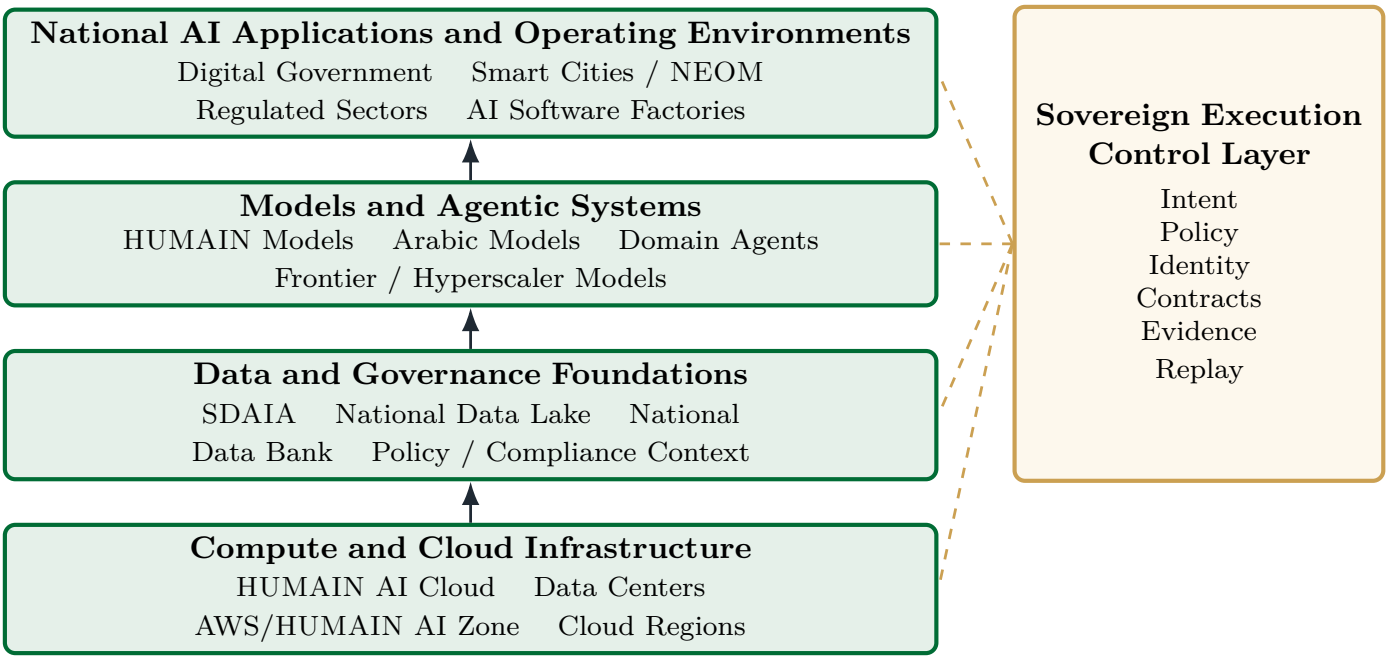

Figure 1.1: *Saudi Arabia's AI infrastructure moment. The Kingdom is assembling the foundations of national-scale AI across compute, cloud, data, models, applications, and operating environments. Sovereign execution is the control layer that allows autonomous AI systems to act safely across this stack.*

These workflows affect citizens, agencies, records, and public trust. The key question is not only whether an AI recommendation is accurate, but whether the resulting action is authorized, bounded, explainable, and replayable.

1.5 NEOM and Smart-City Operating Environments

NEOM and other smart-city or digital-twin environments represent high-consequence operating environments spanning mobility, energy, logistics, utilities, facilities, public services, and urban operations [8]. Digital twins often begin as simulation platforms; over time, their recommendations may influence operational decision layers.

Smart cities benefit from a governed path from digital-twin reasoning to physical-world action.

1.6 Regulated Sectors and National AI Adoption

Regulated sectors such as healthcare, finance, energy, logistics, and education will use AI agents for triage, compliance, optimization, and workflow automation. A common sovereign execution layer can support sector-specific policy packs while preserving a shared architecture: intent, policy, identity, contract, evidence, and replay.

Table 1.1: *KSA infrastructure anchors and the sovereign execution requirement.*

KSA anchor	Infrastructure role	Execution-governance requirement
HUMAIN	Full-stack AI ecosystem: data centers, cloud, models, applications, AI services.	Governance path for autonomous cloud operations, agent marketplaces, GPU/model-serving infrastructure, and AI-generated infrastructure changes.
AWS/HUMAIN AI Zone and hyperscaler partnerships	Large-scale AI infrastructure, services, and adoption capacity.	Preserve local execution control while enabling vendor/model interoperability.
SDAIA / National Data Lake	National data and AI governance foundation.	Policy-bound data access, minimized context, cross-agency evidence, and downstream AI action governance.
DGA / digital government	Public-sector service transformation and cross-agency workflows.	Support citizen-impacting AI workflows that are authorized, bounded, auditable, and replayable.
NEOM / smart cities	Digital-twin and smart-city operating environments.	Governed path from AI reasoning or simulation to physical-world action.
Regulated sectors	Healthcare, finance, energy, logistics, education, and other high-impact domains.	Sector-specific policy packs over a common intent, identity, evidence, and replay model.

1.7 The Next Control Layer

Saudi Arabia is building the AI foundation. The next question is not whether AI can be deployed, but whether autonomous AI can be allowed to act under policy, scoped identity, evidence, and replay.

The rest of this white paper defines that layer: a sovereign execution control plane for national-scale agentic AI.

2 The Strategic Gap: Sovereign Compute Is Not Sovereign Execution

Chapter Thesis

Sovereign compute, sovereign data, and domestic models are necessary foundations for national AI. As AI systems become agentic, sovereignty can extend to execution: who authorized an action, which policy allowed it, what identity performed it, whether it was bounded, and whether it can be audited and replayed.

KSA's infrastructure investments make autonomous AI possible at scale. Compute, data, and models define where AI can run and what it knows. They do not define what it is allowed to do.

An AI model can be hosted locally and still require controls before it triggers a network change. A global model can be useful when its reasoning is kept separate from execution. The strategic gap is no longer about model location. It is execution governance: the ability to constrain, execute, and audit AI actions across national systems.

2.1 Five Layers of Sovereignty

AI sovereignty is not a single checkbox. It operates across multiple layers. The runtime execution layer is where sovereignty translates into direct operational control.

Table 2.1: *From Infrastructure Sovereignty to Execution Sovereignty*

Sovereignty layer	What it controls	What remains outside its scope
Sovereign compute	Location, ownership, and operation of data centers, GPUs, cloud regions, and infrastructure.	Controls where workloads run, while autonomous action authority is handled at the execution layer.
Sovereign data	Data residency, access, exchange, governance, and localization.	Controls where data resides and how it is accessed, while downstream AI-initiated actions benefit from execution governance.
Sovereign models	Domestic model development, Arabic models, fine-tuned models, and model ownership.	A domestic model still benefits from policy-bound execution when it proposes state-changing actions.
Sovereign applications	User-facing AI services, workflows, copilots, digital government tools, and smart-city applications.	Application-level controls may not provide uniform policy, identity, evidence, and replay across vendors and sectors.
Sovereign execution	Intent, policy, identity, contracts, enforcement, evidence, replay, and override for autonomous actions.	This is the complementary runtime layer; it can serve as a national control-plane pattern.

2.2 Why Agentic AI Changes the Risk Model

Traditional IT security is built for predictable users, service accounts, and API callers. Agentic AI changes that operating model: agents can chain tools, delegate sub-tasks, write code, and modify configurations across multiple steps.

For high-impact systems, direct API access should be replaced by an execution-governance path for tasks such as:

- modifying cloud, GPU, network, identity, or infrastructure-as-code configurations;
- accessing or transforming sensitive data;
- routing citizen-service workflows;
- triggering smart-city or regulated-sector operational steps.

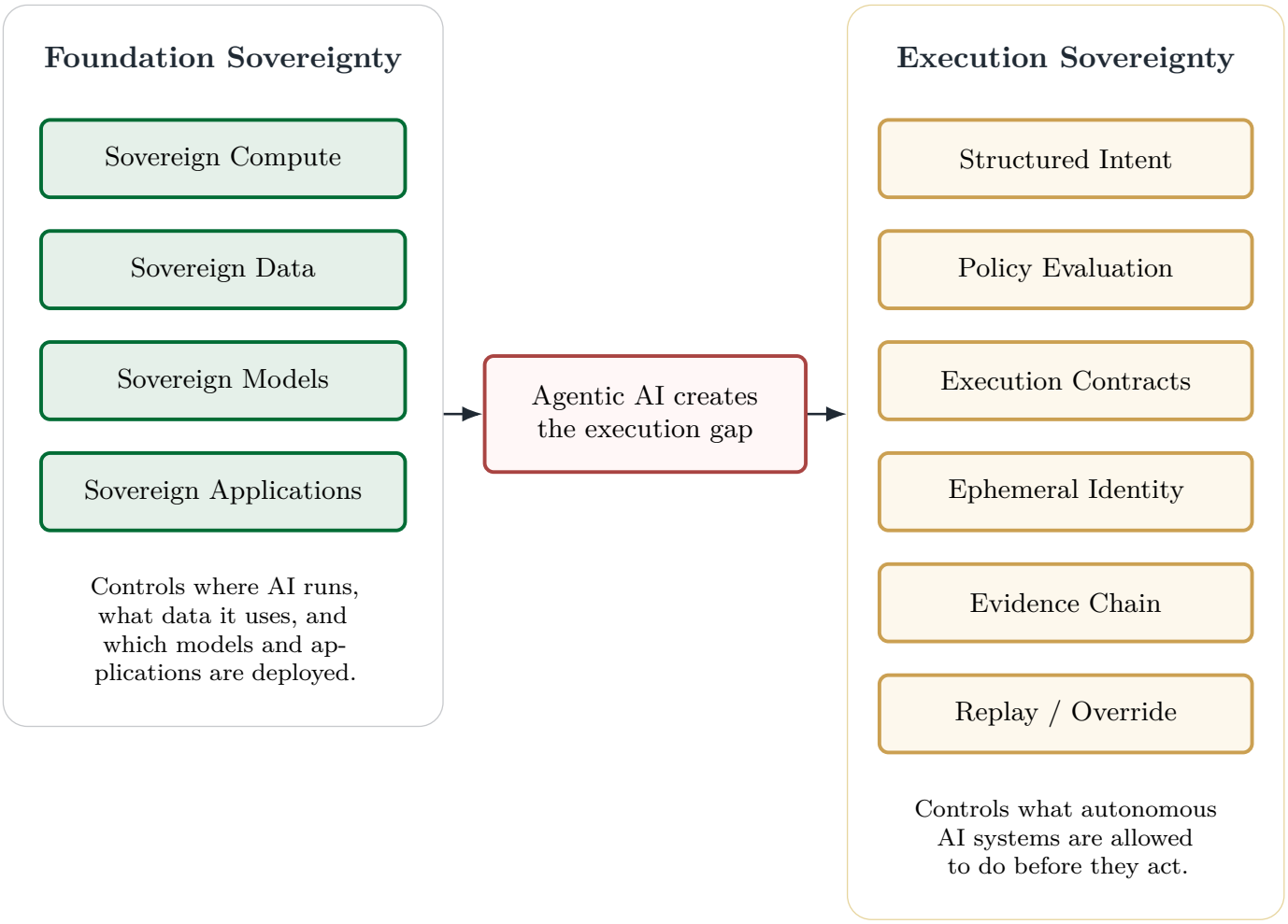

Figure 2.1: *Sovereign compute is foundational, but it does not by itself govern runtime action. Compute, data, models, and applications establish where AI runs and what it can use; sovereign execution closes the execution gap by governing what autonomous systems are allowed to do before they act.*

2.3 The Runtime Questions Execution Governance Addresses

Runtime Questions

- Who or what proposed the action?
- What task context was provided to the model?
- Was sensitive context minimized before reasoning?
- Which policy evaluated the intent?
- What was the expected effect of the action?
- What was the potential operational scope?
- Was human approval required?
- What execution identity was issued?
- What contract bounded the action?
- What evidence proves the action followed policy?
- Can the action be replayed or challenged?
- Can the workflow be suspended or overridden?

Model hosting platforms cannot answer these questions on their own. A runtime control plane can sit between reasoning and execution, evaluate intent, check policy, issue short-lived authority, and record evidence.

2.4 Definition: Sovereign Execution

Sovereign Execution

Sovereign execution is the capability of a nation, public institution, or regulated enterprise to constrain, authorize, execute, observe, and verify autonomous AI-initiated actions under local policy, scoped identity, enforceable contracts, and replayable evidence.

The implication is direct. Sovereign execution is model-agnostic and vendor-agnostic. It applies to domestic and foreign models. It governs actions rather than model outputs alone. It is well suited to high-impact autonomous AI in government, cloud, smart cities, and regulated sectors.

2.5 The Control Pattern

The pattern is not:

$$\text{AI Model} \rightarrow \text{Direct API / Write Access} \rightarrow \text{System Mutation} \rightarrow \text{Logs After the Fact}$$

The closed-loop pattern is:

$$\text{Reasoning} \rightarrow \text{Structured Intent} \rightarrow \text{Context Binding} \rightarrow \text{Policy Evaluation}$$
$$\rightarrow \text{Risk / Approval} \rightarrow \text{Execution Contract} \rightarrow \text{Ephemeral Identity} \rightarrow \text{Controlled Execution}$$
$$\rightarrow \text{Evidence Chain} \rightarrow \text{Replay / Audit} \rightarrow \text{Governance Feedback}$$

Execution Boundary

AI agents should submit structured intent, not receive direct write access.

Execution governance converts model output into a controlled sequence: structured intent, policy and context evaluation, execution contract, short-lived identity, controlled execution, evidence, and replay.

2.6 KSA Decision-Maker Implications

Table 2.2: *Decision-maker implications of sovereign execution.*

Institutional context	Execution implication
HUMAIN-style operations	Autonomous AI cloud operations benefit from operational-scope controls, contract-bound identity, and evidence-backed infrastructure mutation.
SDAIA-style platforms	National data platforms benefit from policy-bound data operations and downstream execution governance.
DGA-style services	Citizen-impacting workflows are well served by authorization, appealability, replay, and human escalation.
NEOM-style cities	Digital-twin reasoning can be separated from physical-world execution.
Regulated sectors	Healthcare, finance, energy, and logistics can use sector-specific policy packs over a shared execution model.

The next chapter introduces the principle that addresses this distinction: intelligence can be global, but execution must be sovereign. Sovereign Agentic Loops implement the principle by separating reasoning from execution before high-impact actions reach national infrastructure.

3 Principle: Intelligence Can Be Global; Execution Must Be Sovereign

Governing Principle

Saudi Arabia's AI strategy does not have to force a false choice between using the world's strongest reasoning systems and preserving national control. The right architecture separates reasoning from execution: models may analyze, plan, and propose, while high-impact actions pass through sovereign policy, identity, approval, and evidence before they affect national systems.

Saudi Arabia will operate a mixed AI ecosystem: HUMAIN models, ALLAM, open-source models, specialized agents, and global frontier systems. This diversity is a strategic asset when the architecture separates systems that reason from systems that execute.

Sovereignty is not only model location. A domestic model can still create operational risk if granted excessive privilege; a global model can be useful if it never sees raw data and never holds execution authority. For high-impact AI, the architectural answer is separation. Models may reason. Sovereign control planes execute.

3.1 The False Choice: Capability Versus Control

AI sovereignty is often framed as a binary: use domestic models for control, or use global models for capability. At the architecture level, this is the wrong choice. The goal is not to ban global models or assume local models are automatically safe; the goal is to ensure no model holds direct execution authority.

Saudi Arabia can use global intelligence when raw sensitive context is minimized before reasoning and execution remains local, policy-bound, identity-scoped, and evidence-backed. A model can analyze a task, propose a plan, or draft a structured request without standing permission to mutate infrastructure, route citizen workflows, change smart-city operations, or deploy generated software.

The model is not the control plane.

SAL is therefore not a localization doctrine; it is an execution-authority doctrine.

3.2 Sovereign Agentic Loops

Definition

Sovereign Agentic Loops are an architectural pattern that separates AI reasoning from operational execution. A model or agent may receive minimized task context and produce a structured intent, but only a sovereign execution environment can evaluate, authorize, and enforce the resulting action.

SAL is the pattern that turns the principle into an operating architecture [6]. It allows the Kingdom to use domestic reasoning assets, specialized agents, and external models without granting those models direct authority over high-impact systems. The loop has four layers.

Reasoning Layer. The reasoning layer handles analysis, planning, and drafting across domestic models, specialized agents, and global frontier systems. Its output is only a proposal; final authority stays with the sovereign execution environment.

Sovereign Obfuscation Membrane. The membrane reduces what a reasoning system sees through context minimization, redaction, anonymization, and policy-filtered task views. Its role is to keep national context, regulated data, and operational state inside approved boundaries while preserving useful reasoning.

Structured Intent Boundary. Models emit declarative intent, not direct API calls. A structured intent can describe the requested action, target, expected effect, risk class, required authority, and constraints. This makes model output admissible for evaluation by a control plane rather than executable by default.

Sovereign Execution Environment. The execution environment binds live context, evaluates local policy, scores risk and operational impact, routes approvals, generates execution contracts, issues ephemeral identity, enforces execution, records evidence, supports replay, and preserves override. This is where sovereign authority becomes operational.

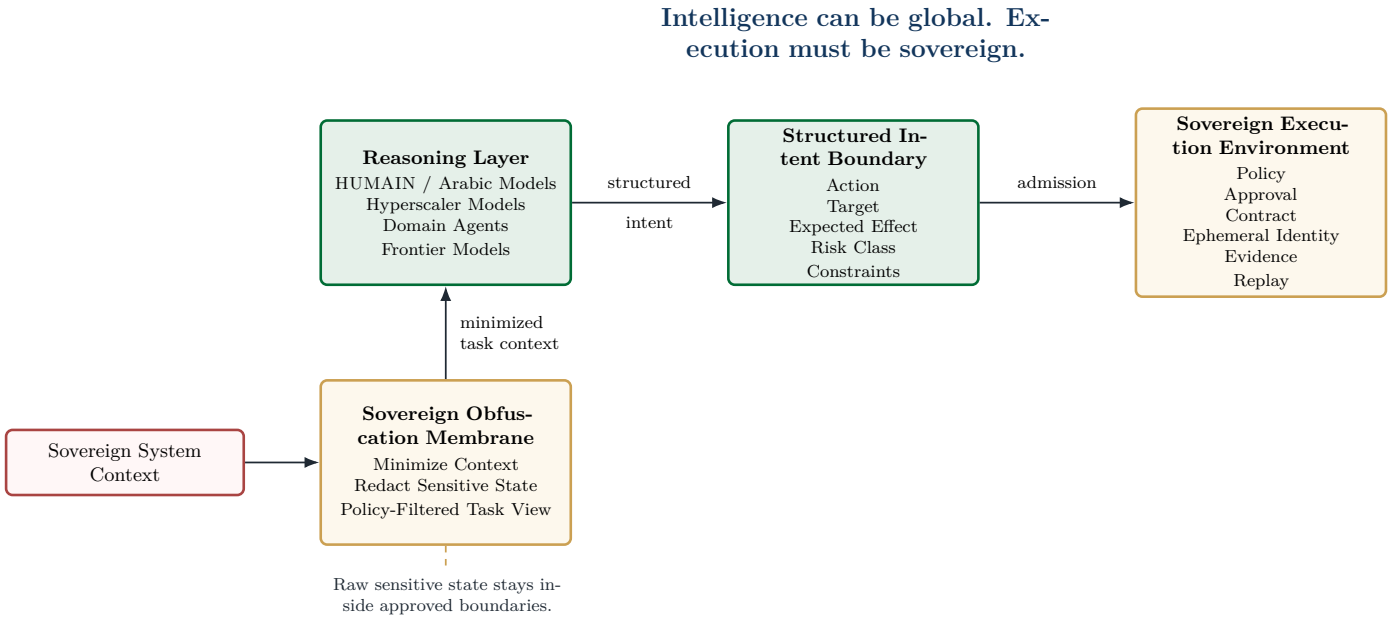

Figure 3.1: *Sovereign Agentic Loops separate reasoning from execution. Models may analyze minimized context and emit structured intent, but high-impact actions are evaluated, authorized, executed, and evidenced inside the sovereign execution environment.*

3.3 How SAL Works

A SAL pattern can be implemented as a repeatable lifecycle:

1. A national system or operator defines a task.
2. The obfuscation membrane converts raw context into minimized task context.
3. A reasoning model analyzes the task and proposes a structured intent.
4. The intent crosses into the sovereign execution control plane.
5. Local policy and live context determine admissibility.
6. Approved intent becomes a bounded execution contract.
7. Short-lived identity is issued only for the approved contract.
8. Execution is enforced through a controlled pathway.
9. Evidence is recorded for audit and replay.

The loop changes the governance surface. The institution no longer has to trust every model in every situation; it can ask whether a proposed action is admissible under sovereign policy, scoped identity, and replayable evidence.

3.4 Why This Matters for KSA

KSA relevance: SAL

SAL can give Saudi Arabia a practical way to benefit from global frontier intelligence and domestic AI assets at the same time. Capability can be imported or diversified; execution remains governed by sovereign policy, scoped identity, and evidence.

The pattern is repeatable: reasoning may be diversified, but authority remains local.

Table 3.1: *KSA implications of Sovereign Agentic Loops.*

KSA context	SAL implication
HUMAIN-style operations	Global or domestic models can support AI cloud operations without receiving persistent write authority over cloud infrastructure, GPU platforms, model-serving systems, or infrastructure-as-code.
SDAIA-style platforms	Analytical agents can reason over minimized, policy-filtered context rather than raw national data, while downstream data operations remain governed by local policy and evidence.
DGA-style services	AI copilots and workflow agents can propose citizen-service actions without becoming the authority that executes them. Approval, escalation, and replay remain part of the execution pathway.
NEOM-style cities	Smart-city and digital-twin reasoning can remain separated from physical-world operations, creating a governed path from simulation to action.
Regulated sectors	Healthcare, finance, energy, and logistics systems can use powerful reasoning models while preserving sector-specific execution controls over sensitive workflows.

3.5 Design Implications for Sovereign Reasoning

The SAL principle leads to a concrete design checklist for national-scale AI systems:

- Context minimization by default.
- No standing write access for reasoning models.
- Structured intent as the only crossing point from reasoning to execution.
- Local policy evaluation before execution.
- Short-lived execution identity.
- Evidence chain for every high-impact action.
- Replayability for audit and incident review.
- Vendor and model agnosticism.

These requirements create a common control pattern across models, clouds, government systems, and sector-specific applications.

3.6 What SAL Does Not Claim

SAL is strongest when its claims are kept precise. It does not claim that all reasoning requires domestic models. It does not claim that foreign models are inherently unsafe. It does not claim that obfuscation alone solves governance. It does not replace cybersecurity, data governance, identity, cloud controls, human oversight, or sector regulation.

Instead, SAL provides the architectural separation those controls use to govern AI-initiated execution. It gives each control a place to attach: data governance at the membrane, policy at the intent gate, identity at the execution contract, security at the controlled pathway, and audit at the evidence chain.

SAL establishes the principle. The next chapter turns that principle into a reference macro-architecture: the Autonomous Systems Control Plane for KSA.

4 Reference Architecture: Autonomous Systems Control Plane for KSA

Reference Architecture

The Autonomous Systems Control Plane extends cloud control-plane discipline into the agentic AI era. It treats AI-generated actions as proposals requiring admission, not as trusted API calls. For Saudi Arabia, this pattern provides a complementary execution layer between sovereign AI infrastructure and high-impact autonomous operations.

The previous chapter established the principle; ASCP turns it into architecture. Cloud control planes already govern resource creation, identity, configuration, and state transition. Agentic AI benefits from the same discipline for AI-initiated actions.

ASCP is not a model or dashboard. It is a runtime governance path that treats AI-generated actions as proposals requiring admission, not trusted API calls.

4.1 Architecture Overview

ASCP treats AI systems as sources of proposals, not holders of standing authority [3]. The architecture has four layers.

Reasoning and agent layer. Domestic models, hyperscaler agents, and software copilots analyze, plan, and propose. They do not touch production.

Intent boundary. Models and agents submit structured intent. The intent describes the requested action, target, expected effect, constraints, risk class, and required authority. This boundary is where model output becomes a control-plane input.

ASCP **core.** The core binds intent to live context, evaluates policy, routes approvals, generates execution contracts, issues short-lived identity, enforces execution, and records evidence.

National production fabrics. ASCP can govern the path into HUMAIN-style AI cloud operations, SDAIA-style data workflows, DGA-style digital government systems, NEOM-style digital-twin environments, regulated-sector platforms, and AI software factories.

4.2 ASCP Core Components

ASCP components can be deployed centrally, federated by domain, or embedded into platform teams. The key is the same control path: intent, context, policy, contract, identity, execution, evidence, replay.

Table 4.1: *ASCP Core Components and KSA Institutional Value*

Component	Role in ASCP	KSA institutional value
Intent intake	Receives structured proposals from models, agents, copilots, and pipelines.	Standard entry point across ministries, clouds, and vendors.
Context engine	Binds intent to system state, policy context, data class, and risk signals.	Decisions reflect live infrastructure, agency rules, and sector conditions.
Policy engine	Evaluates admissibility under national, sectoral, organizational, and workflow rules.	Makes policy enforceable at runtime.
Risk and operational-impact evaluator	Estimates potential impact before execution.	Routes high-impact changes to escalation.
Approval router	Sends sensitive actions to operators or supervisory workflows.	Preserves accountable human authority.
Execution contract generator	Converts approved intent into a bounded contract.	Limits execution to the approved action and constraints.
Ephemeral identity issuer	Issues short-lived credentials tied to the contract.	Reduces standing privilege and credential exposure.
Execution gateway	Enforces the contract against APIs, infrastructure, workflows, or pipelines.	Creates a controlled path from proposal to mutation.
Evidence recorder	Captures intent, context, decision, approval, contract, identity, execution, and result.	Provides audit and regulator-grade evidence.
Replay and audit console	Reconstructs the action path for review and incidents.	Enables dispute handling and continuous improvement.
Emergency stop / override	Suspends or blocks execution paths as conditions change.	Supports institutional control over autonomous workflows.

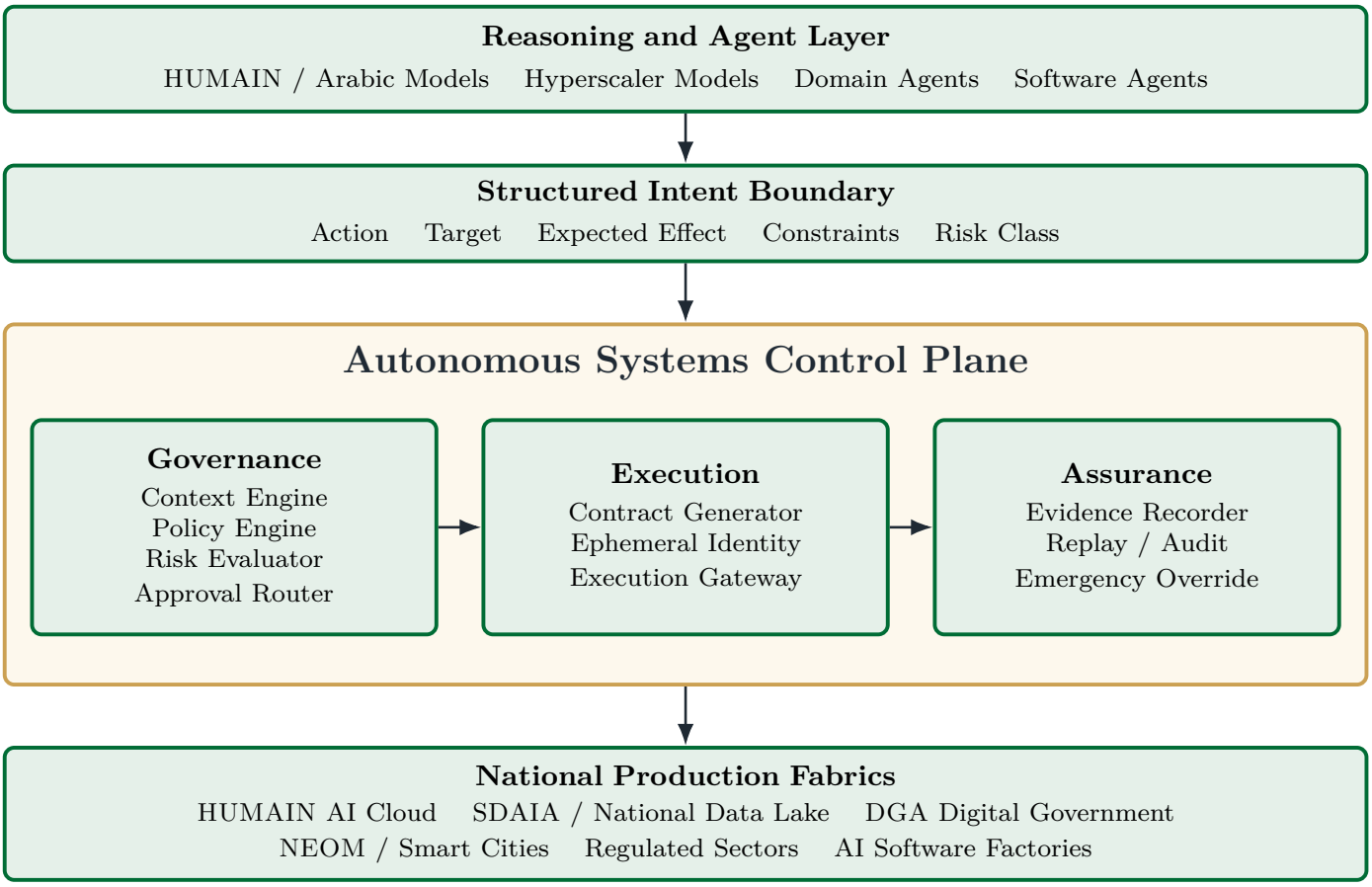

Figure 4.1: *Autonomous Systems Control Plane for KSA. ASCP sits between reasoning systems and national production environments, converting AI-generated intent into policy-bound, identity-scoped, evidence-backed execution.*

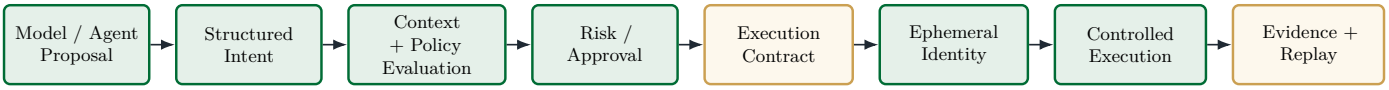

Figure 4.2: *Agentic action lifecycle. ASCP converts an AI-generated proposal into governed execution through structured intent, policy evaluation, execution contracts, short-lived identity, controlled execution, and evidence capture.*

4.3 The Agentic Action Lifecycle

ASCP converts an AI proposal into governed execution through a repeatable lifecycle:

1. Model or agent proposes an action as structured intent.
2. ASCP binds the intent to live context.
3. Policy and risk evaluation determine admissibility.
4. Approval routing escalates sensitive actions.
5. Approved intent becomes an execution contract.
6. Ephemeral identity is issued for that contract.
7. The execution gateway performs the bounded action.
8. Evidence and replay records close the loop.
9. Governance feedback updates policy, schemas, and runbooks.

4.4 What Makes ASCP Different from Traditional AI Governance

Traditional AI governance focuses on model behavior: content safety, prompt filtering, benchmark scores, and responsible AI review. ASCP adds runtime execution governance. The question shifts from whether a response is acceptable to whether a proposed action is admissible under policy, identity, contract, and evidence constraints.

Table 4.2: *Model Governance vs. Execution Governance*

Model governance	Execution governance
Reviews model outputs.	Governs system actions.
Focuses on prompts, content, and benchmarks.	Focuses on intent, policy, identity, contracts, and evidence.
Often occurs before deployment or at interaction time.	Occurs at runtime before execution.
Uses logs and monitoring after the fact.	Produces evidence before, during, and after execution.
Helps make models safer.	Makes autonomous actions governable.
Model-centric.	Control-plane-centric.

4.5 KSA Deployment Pattern

ASCP maps to HUMAIN-style AI cloud operations, SDAIA-style data workflows, DGA-style digital government, NEOM-style digital twins, regulated-sector platforms, and AI software factories. The same pattern governs cloud scaling, model-serving changes, data workflows, citizen-service routing, simulation-to-action paths, sector workflows, and generated deployment artifacts.

KSA relevance: ASCP and Workforce

ASCP offers a repeatable execution-governance architecture across AI cloud, national data systems, digital government, smart cities, and regulated sectors. It also points to a workforce path: from manual operation toward AI governance, protocol engineering, evidence review, and autonomous-operations oversight.

4.6 Design Principles

Table 4.3: *ASCP design principles.*

Principle	Meaning
Intent before execution	Autonomous systems submit proposed actions before any system mutation occurs.
Policy before privilege	Permission is evaluated before credentials are issued.
Context before authorization	Decisions are bound to live system state, data classification, workflow rules, and risk signals.
Contracts before credentials	Approved actions become bounded execution contracts before identity is created.
Evidence before trust	Evidence chains are the audit primitive for autonomous AI.
Replay before finality	Operators and auditors can reconstruct why the action was allowed and what happened.
Human authority for high-impact actions	Sensitive actions route to accountable people or supervisory workflows.
Model and vendor agnosticism	The control plane governs actions from domestic, open-source, hyperscaler, and frontier models.
Emergency override by design	Institutions retain the ability to suspend or block autonomous pathways.
Open protocol boundaries	Integration can occur through clear intent, policy, contract, identity, and evidence interfaces.

4.7 Boundary of the Architecture

ASCP does not replace cybersecurity programs, model safety, responsible AI processes, or platform ownership. It is a reference architecture for governing high-impact AI-initiated actions across heterogeneous systems.

ASCP defines the macro-architecture. The next chapter defines the protocol surface that can make this architecture operational: OpenKedge, the intent-governance protocol for converting AI proposals into policy-bound execution.

5 OpenKedge: National Intent Governance Protocol

Protocol Surface

OpenKedge provides the protocol surface for sovereign execution. It defines how AI-generated proposals become policy-bound, identity-scoped, evidence-backed actions. In the KSA context, OpenKedge can serve as an open, vendor-neutral intent-governance protocol across AI cloud operations, national data workflows, digital government, smart-city systems, regulated sectors, and AI-generated software pipelines.

ASCP provides the architecture; OpenKedge provides the protocol surface [4]. It is not a dashboard or model. It is an open intent-governance protocol that makes agentic AI legible as a governed proposer rather than a direct actor.

5.1 Why ASCP Needs a Protocol Surface

A national control-plane pattern scales more easily when vendors and platforms integrate through a common protocol boundary. OpenKedge provides that intent boundary so ministries, clouds, agents, and regulated systems can ask for governed execution through the same protocol instead of one-off pathways.

5.2 The OpenKedge Lifecycle

1. **Agent submits structured intent.** The agent expresses the proposed action as a control-plane object, not as a direct API call. The institutional value is a standard entry point for autonomous actions across models, vendors, and workflows.
2. **Control plane binds live context.** The intent is evaluated against current system state, policy context, data sensitivity, actor role, and operational conditions. This helps avoid static approval being reused in a changed environment.
3. **Policy engine evaluates admissibility.** National, sectoral, organizational, and workflow-specific rules determine whether the action is allowed, blocked, or escalated. Policy before privilege becomes a runtime practice rather than a governance slogan.
4. **Approved intent becomes an execution contract.** Approval does not create broad authority. It creates a bounded, machine-enforceable contract describing what may happen, where, when, and under which constraints.
5. **Short-lived execution identity is issued.** The execution identity is scoped to the approved contract and expires after completion or timeout. This reduces standing administrative privilege for autonomous agents.
6. **Execution gateway enforces the contract.** The gateway is the controlled path into target systems, APIs, infrastructure, workflows, or deployment pipelines. It enforces the contract rather than trusting the agent to self-limit.
7. **Evidence chain records the full path.** The protocol records intent, context, policy decision, approval path, contract, identity, execution, and result. Evidence before trust makes autonomous execution reviewable.
8. **Replay and audit reconstruct the event.** Operators, auditors, regulators, and incident teams can examine why an action was allowed and what occurred. Replay supports dispute handling, assurance, and continuous improvement.

5.3 Structured Intent

Structured intent is the core unit of governance: not a prompt, not a raw API call, but a machine-readable control-plane object. It declares the actor, target system, expected effect, requested authority, constraints, time window, and evidence requirements.

Intent is the point where probabilistic reasoning becomes governable infrastructure.

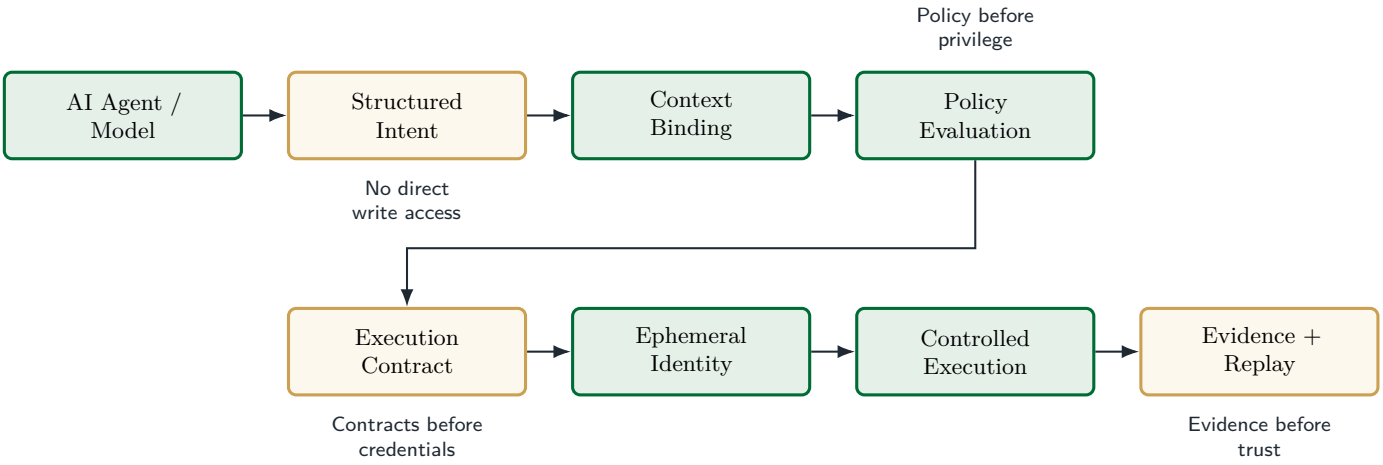

Figure 5.1: *OpenKedge protocol flow. AI systems submit structured intent rather than receiving direct write access. The protocol binds intent to context, evaluates policy, generates an execution contract, issues short-lived identity, enforces execution, and records evidence for audit and replay.*

This is the practical boundary between reasoning and execution: OpenKedge converts model proposals into explicit requests that policy engines, identity systems, gateways, and auditors can understand.

5.4 Context-Bound Policy Evaluation

Policy evaluation is strongest when it binds intent to live context: system state, data sensitivity, actor role, target criticality, dependency topology, operational scope, and approval status. A request that is safe in one context may require escalation in another: scaling a test GPU cluster, changing a production network policy, reading aggregated analytics, accessing identifiable records, simulating a smart-city change, or applying a physical-world action.

5.5 Execution Contracts

An approved intent becomes an execution contract, not broad privilege. The contract states what may happen, where, under which constraints, during which time window, with what rollback expectations, and with what evidence obligations.

Contracts before credentials.

5.6 Ephemeral Execution Identity

AI agents do not need permanent administrative credentials. Once an execution contract is approved, the system issues short-lived, task-scoped identity bound to the approved contract and expiring after completion or timeout. Called-via or control-plane-mediated execution can be used as a vendor-neutral pattern: target systems see that the action arrived through the approved pathway.

Policy before privilege.

5.7 Evidence Chain

Every governed action can produce evidence binding intent, context, policy decision, approval path, execution contract, identity, system mutation, observed result, and replay metadata. This records that execution followed the approved path and bridges protocol governance to the trust fabric in the next chapter.

Evidence before trust.

5.8 KSA Institutional Mapping

The protocol becomes most useful when mapped to concrete institutional contexts.

Table 5.1: *OpenKedge Institutional Mapping for KSA*

KSA context	OpenKedge role	Example governed action
HUMAIN-style AI Cloud	Standard protocol for AI cloud operations and autonomous infrastructure changes.	Agent proposes GPU cluster scaling or model-serving configuration change.
SDAIA-style Data Platforms	Protocol for policy-bound data operations and minimized-context workflows.	Agent proposes analytical task execution over approved data context.
DGA-style Digital Government	Protocol for citizen-service workflow automation and cross-agency orchestration.	Agent proposes permit routing, document verification, or case escalation.
NEOM-style Smart Cities	Protocol boundary between digital-twin reasoning and operational action.	Agent proposes mobility, energy, logistics, or facility optimization.
Regulated Sectors	Shared governance protocol with sector-specific policy packs.	Agent proposes healthcare, finance, energy, or logistics workflow step.
Saudi AI Software Factories	Protocol interface for AI-generated code, workflow, and infrastructure deployment.	Agent proposes infrastructure-as-code change or generated workflow release.

5.9 Interoperability and Ecosystem Value

OpenKedge is framed as open and vendor-neutral. It can support domestic models, global models, local startups, hyperscalers, ministries, system integrators, and regulated enterprises. The aim is not one implementation; it is a standardized governance boundary that reduces lock-in and creates opportunity for Saudi integrators and AI companies.

The protocol boundary is where sovereignty becomes operational.

5.10 Boundary of the Protocol

OpenKedge does not replace cybersecurity, network policy, or data governance. It does not demand a monolithic centralized product.

OpenKedge is presented here as a reference protocol surface, not as a claim of national standardization.

Its role is narrow: it provides the protocol layer that ties those existing systems into sovereign execution governance. It gives agents and infrastructure a shared vocabulary for intent, contracts, and evidence.

OpenKedge defines the protocol path from proposed action to governed execution. The next chapter defines the trust fabric that makes those actions verifiable: evidence chains, execution identity, audit, and replay.

6 Verifiable Agentic Infrastructure: Evidence as the New Audit Layer

Trust Layer

Autonomous AI changes the audit problem. Traditional logs can show that an event happened, but they often do not prove why the action was authorized, which policy admitted it, which contract bounded it, which short-lived identity executed it, or whether the decision path can be replayed. Verifiable Agentic Infrastructure positions evidence, alongside logs, as the core audit primitive for sovereign execution.

A protocol path is most useful when it produces verifiable evidence. Traditional logs are necessary, but they are often passive, fragmented, and written after the fact. Autonomous operations require evidence that captures the decision before it executes.

VAI provides the trust layer for ASCP and OpenKedge [7]. Logs describe events; evidence chains prove governed execution.

6.1 Why Evidence Complements Logs

Logs are valuable for troubleshooting, but they often capture events after the fact and across disconnected systems. For autonomous AI, the question is not only "what happened?" but "who authorized this, under which policy, and within what contract?" Passive logging cannot govern high-impact AI on its own.

Table 6.1: *Logs vs. Evidence Chains*

Traditional logs	Evidence chains
Record events after they occur.	Bind decision and execution before, during, and after action.
Often system-specific.	Cross-system and control-plane anchored.
Show what a component did.	Show why the action was allowed.
May omit model intent and policy context.	Include intent, context, policy, approval, and contract.
Support troubleshooting.	Support audit, replay, dispute, and accountability.
Passive record.	Active governance artifact.
Difficult to correlate across vendors.	Designed for cross-vendor evidence continuity.

6.2 The Evidence Chain

Definition

An evidence chain is a tamper-evident record that links the full path from AI-generated intent to controlled execution and observed result.

The evidence chain is not a log stream. It is a control-plane record proving that an execution followed its approved path.

- agent or model identity;
- submitted structured intent;
- minimized context snapshot;
- policy version and decision;
- approval path;
- execution contract;
- ephemeral identity reference;

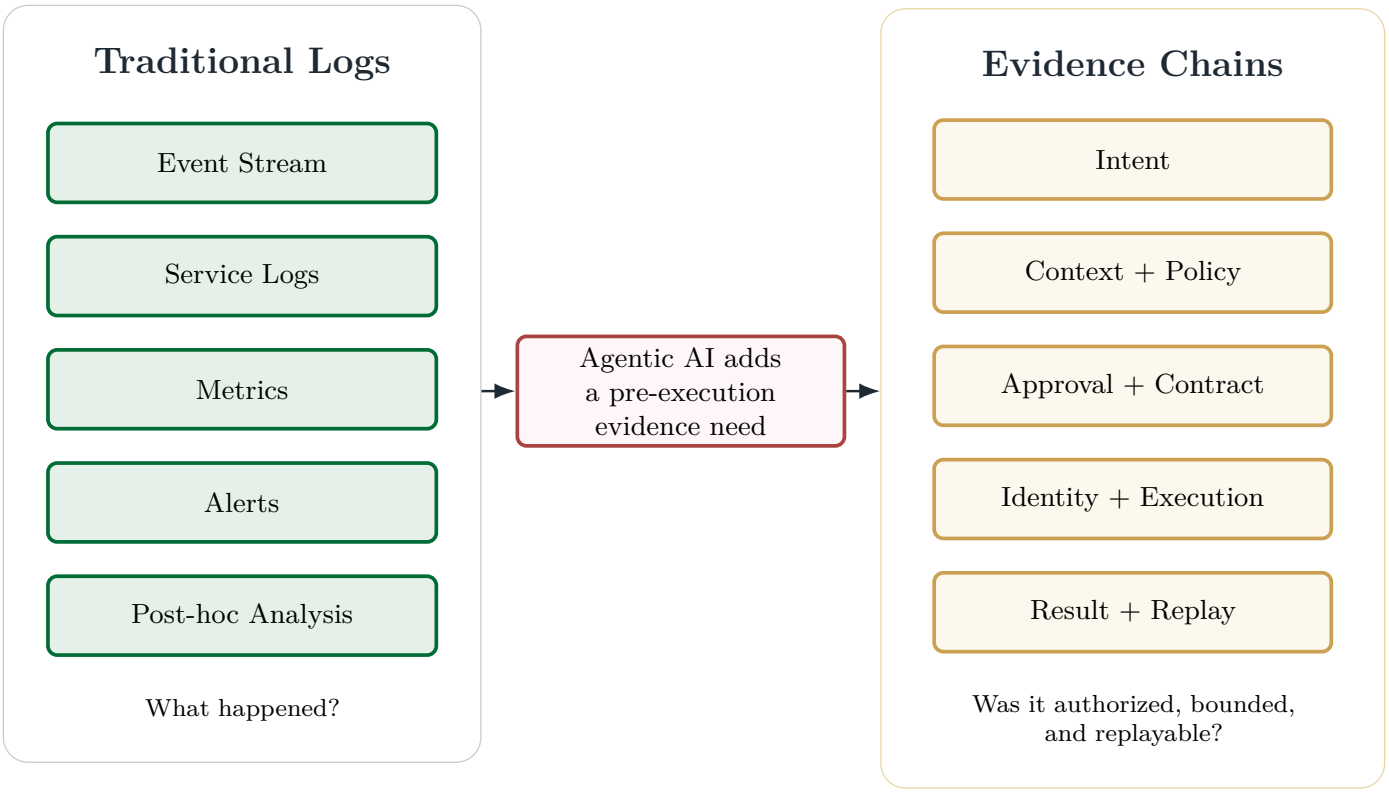

Figure 6.1: *Traditional logs versus evidence chains. Logs often describe what happened after execution; evidence chains bind intent, context, policy, approval, contract, identity, execution, and result into a replayable accountability record.*

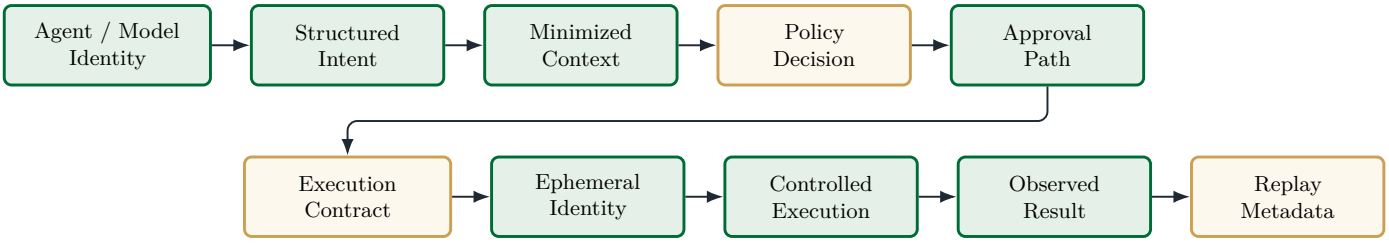

Figure 6.2: *Agentic evidence chain. A replayable evidence record links the submitted intent, minimized context, policy decision, approval, execution contract, ephemeral identity, controlled execution, observed result, and replay metadata.*

- target system and operation;
- observed result;
- replay metadata.

6.3 Ephemeral Execution Identity

Evidence is strongest when execution identity is short-lived. The control plane issues task-scoped identity only after approving an execution contract, and the evidence chain records the identity reference without exposing secrets. Auditors can see exactly which contract granted authority.

Identity can be computed from approval, not pre-granted to the agent.

6.4 Replayable Accountability

Replay reconstructs the decision path: what the agent proposed, which policy allowed it, who approved it, which identity executed it, and what changed. It supports audit, incident response, and regulator review without re-running the action in production.

Every high-impact autonomous action should be designed to be replayable.

6.5 KSA Institutional Mapping

Table 6.2: *Evidence Requirements Across KSA Environments*

KSA environment	Why evidence matters	Evidence focus
HUMAIN-style AI Cloud	Autonomous cloud operations benefit from incident reconstruction and infrastructure accountability.	Cluster changes, model-serving updates, identity grants, infrastructure-as-code execution, rollback records.
SDAIA-style Data Platforms	Cross-agency data operations benefit from policy-bound access and accountability.	Data-access intent, minimized context, policy decision, approval path, downstream action record.
DGA-style Digital Government	Citizen-impacting workflows benefit from appealability and public-sector accountability.	Workflow intent, eligibility or routing decision, human approval, execution identity, case outcome.
NEOM-style Smart Cities	Smart-city operations may affect physical systems and public services.	Simulation result, operational intent, safety threshold, approval, execution record, observed effect.
Regulated Sectors	Healthcare, finance, energy, and logistics benefit from regulator-grade audit.	Sector policy version, decision rationale, approval, execution identity, compliance report.
Saudi AI Software Factories	AI-generated code and infrastructure changes benefit from provenance and admissibility evidence.	Generated artifact, invariant checks, review outcome, deployment contract, runtime result.

6.6 From Compliance Reporting to Operational Trust

Traditional compliance often produces reports after the fact. Autonomous infrastructure benefits from evidence generated as part of execution itself, giving ministries, regulators, AI cloud operators, and smart-city operators a common evidence language.

For executive leadership, evidence chains convert autonomy from a trust assumption into an inspectable operating model.

6.7 Boundary of VAI

VAI does not replace cybersecurity, IAM, monitoring, observability, or human judgment. It provides the evidence layer that lets operators and regulators verify whether autonomous execution followed policy.

VAI helps make sovereign execution verifiable. The next chapter extends the same governance principle earlier in the lifecycle: before AI-generated code, workflows, or infrastructure configurations enter production, they can satisfy protocol-level admissibility. That is the role of Protocol-Driven Development.

7 Protocol-Driven Development: Governing AI-Generated Software

Software Supply-Chain Layer

AI-generated software changes the software supply-chain problem. When models can rapidly generate code, infrastructure-as-code, workflow logic, policies, and configurations, the strategic bottleneck is no longer producing candidate implementations. The bottleneck is deciding which generated artifacts are admissible into national or regulated production systems.

Earlier chapters focused on runtime governance. AI also changes how software is written: KSA platforms will use generative AI to write code, workflows, policies, adapters, and infrastructure templates. Generated implementation is abundant; admissibility is scarce. PDD provides the admission-control layer that governs whether artifacts enter production [5].

7.1 Why AI-Generated Software Changes the Control Point

Traditional governance assumes code is expensive to write. AI shifts the bottleneck from creation to admission: which candidate is safe, compliant, operable, and compatible with national infrastructure constraints? Examples include infrastructure-as-code, ministry workflow automation, data transformations, smart-city rules, compliance logic, generated adapters, remediation scripts, and deployment manifests.

7.2 The PDD Principle

Definition

Protocol-Driven Development treats the protocol as the primary software artifact. Implementations are replaceable candidates that are admitted only if they satisfy the protocol's structural, behavioral, and operational invariants.

Compilation and model provenance are useful signals, but they are not sufficient admission criteria. An artifact is admitted only when it satisfies the protocol, turning AI-assisted software generation into a more governable acceleration path.

The protocol becomes the control boundary.

7.3 Three Classes of Invariants

Table 7.1: *PDD Invariant Classes*

Invariant class	What it governs	KSA example
Structural invariants	Interfaces, schemas, resource boundaries, dependency shape, required fields, allowed integration patterns.	Generated government workflow can use approved identity, data, and service interfaces.
Behavioral invariants	Allowed state transitions, authorization logic, safety checks, fallback behavior, escalation conditions.	A citizen-service workflow routes high-impact approval through required policy and human review.
Operational invariants	Observability, rollback, rate limits, operational-impact limits, deployment constraints, evidence requirements.	AI-generated infrastructure-as-code includes rollback path, evidence hooks, and production-scope limits.

Generated implementation is abundant; admissibility is scarce.

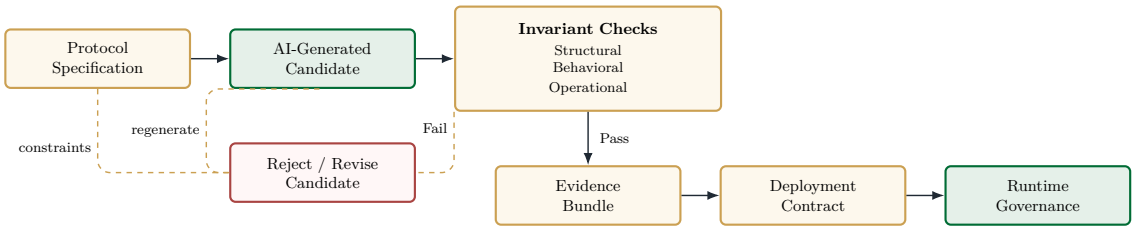

Figure 7.1: *Protocol-Driven Development admission loop. A protocol specification guides AI-generated candidate implementations. Candidates are admitted only if they satisfy structural, behavioral, and operational invariants. Failed candidates are rejected or revised and returned to the generation loop; passing candidates produce an evidence bundle, deployment contract, and runtime governance linkage.*

7.4 The Admission Pipeline

1. **Protocol specification.** The protocol defines required interfaces, invariants, policies, evidence obligations, and operational constraints.
2. **Generated candidate artifact.** A model or agent generates code, IaC, workflow logic, configuration, policy rules, or integration adapters.
3. **Static and structural validation.** The artifact is checked for schemas, interfaces, dependencies, approved resources, and integration boundaries.
4. **Behavioral validation.** The artifact is evaluated against allowed state transitions, authorization rules, escalation paths, and safety constraints.
5. **Operational validation.** The artifact is checked for observability, rollback, rate limits, deployment scope, and evidence hooks.
6. **Evidence bundle.** The admission process produces evidence describing the protocol version, checks performed, results, approvals, and deployment constraints.
7. **Deployment contract.** If admitted, the artifact enters deployment under a bounded contract linked to runtime governance.

PDD governs what enters the system; ASCP governs what acts inside it.

7.5 How PDD Links to ASCP, OpenKedge, and VAI

PDD is pre-deployment governance; ASCP and OpenKedge govern runtime actions; VAI records evidence before, during, and after execution. A generated artifact admitted by PDD can carry evidence into the runtime layer, where later actions are governed and replayed.

Generated Artifact → Protocol Admission → Deployment Contract
→ Runtime Intent Governance → Evidence Chain

7.6 KSA Institutional Mapping

Table 7.2: *PDD Institutional Mapping for KSA*

KSA environment	Generated artifacts	PDD governance requirement
HUMAIN-style AI Cloud	Infrastructure-as-code, cluster automation, model-serving configuration, SRE scripts.	Admit artifacts with rollback, operational-impact limits, evidence hooks, and approved cloud interfaces.
DGA-style Digital Government	Citizen-service workflows, permit routing logic, document verification flows, agency integration adapters.	Apply policy, approval, privacy, escalation, and replay invariants before deployment.
SDAIA-style Data Platforms	Data pipelines, access logic, analytical workflows, transformation scripts.	Apply data classification, context minimization, authorized access, and evidence requirements.
NEOM-style Smart Cities	Digital-twin automation rules, mobility/energy/logistics optimization logic, facility scripts.	Use simulation validation, safety thresholds, operational rollback, and physical-action constraints.
Regulated Sectors	Healthcare triage workflows, financial compliance logic, energy operations scripts, logistics optimizers.	Apply sector-specific invariants and regulator-facing evidence before production.
Saudi AI Software Factories	Code, tests, deployment manifests, integrations, policies, and operational workflows.	Use protocol admissibility as the standard entry point for generated software delivery.

7.7 What PDD Does Not Replace

PDD does not replace human engineering. It does not catch every bug. It does not force you to use AI for all code.

It simply creates a firm boundary: demonstrate that the artifact satisfies the protocol before it enters production.

7.8 Strategic Value for Saudi Arabia

PDD can help KSA scale AI-assisted software delivery across government software factories, AI cloud operations, smart-city platforms, and regulated sectors. It gives procurement teams a way to request admissibility evidence and gives local integrators clear protocol targets.

Saudi Arabia can accelerate software delivery without making implementation generation the trust boundary.

PDD governs what enters the system. ASCP and OpenKedge govern what acts inside it. The next chapter brings these layers together through concrete KSA deployment playbooks: HUMAIN-style AI cloud operations, SDAIA-style data governance, DGA-style public administration, NEOM-style smart-city digital twins, regulated sectors, and Saudi AI software factories.

8 KSA Deployment Playbooks

Deployment thesis

The same sovereign execution pattern can serve multiple KSA institutional contexts. HUMAIN, SDAIA, DGA, NEOM, regulated sectors, and Saudi AI software factories each use different policies and operating models, but they share the same architectural opportunity: autonomous AI actions can be proposed as intent, evaluated against context and policy, executed under bounded identity, and recorded as replayable evidence.

Earlier chapters established the core architecture: SAL separates reasoning from execution; ASCP provides the control plane; OpenKedge provides the protocol; VAI provides evidence; and PDD governs generated code [6, 3, 4, 7, 5]. The playbooks below translate that architecture into operating models for AI cloud, national data, digital government, smart cities, regulated sectors, and software factories.

The goal is not a single governed AI application. The goal is a repeatable execution layer for the Kingdom's AI economy. Policy packs and risk thresholds differ, but the control grammar remains stable: intent, policy, identity, contracts, evidence, replay.

This is the practical meaning of sovereign execution for institutions: common control plane, domain-specific policy. Models may reason. Sovereign control planes execute.

8.1 Playbook 1: HUMAIN AI Cloud Operations

Operational problem. In a reference HUMAIN-style AI cloud environment, autonomous operations can involve high-velocity work across GPU clusters, model-serving infrastructure, AI cloud environments, capacity planning, networking, identity, cost controls, incident response, and infrastructure-as-code. AI agents can help operate this environment without holding standing administrative authority [9].

Control-plane answer. ASCP governs AI cloud operations by requiring operational agents to submit structured intent. OpenKedge evaluates the intent against live context and policy. Approved actions become execution contracts. Ephemeral identity is issued only for the approved action. VAI records evidence. PDD governs generated IaC or automation scripts before deployment.

Required controls.

- Operational-impact scoring and production/staging distinction.
- Contract-bound credentials.
- Approval escalation for production or network/security changes.
- Rollback requirements and infrastructure evidence chain.
- PDD checks for generated IaC and SRE scripts.

Candidate pilot. A governed AI cloud operations sandbox for non-critical cluster scaling, model-serving configuration, or generated IaC admission.

Success indicators.

- Autonomous operations routed through intent governance.
- Reduced standing agent privileges.
- Replayable operational changes and rollback evidence.
- Faster approval for low-risk actions.

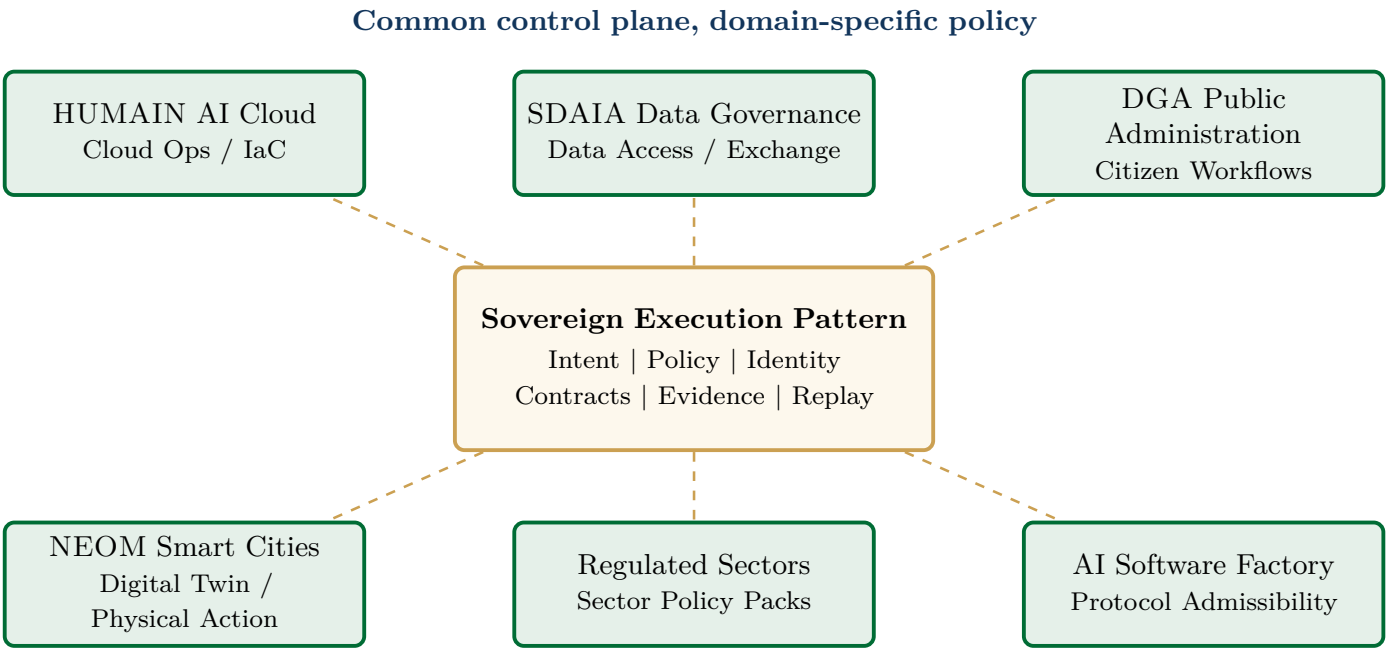

Figure 8.1: *KSA deployment playbook matrix. Different KSA environments use different policy packs, while the common sovereign execution pattern remains the same: intent, policy, identity, contracts, evidence, and replay.*

8.2 Playbook 2: SDAIA-Style National Data and AI Governance

Operational problem. National data platforms require more than just access control. When AI agents reason over centralized data, governance must track what those agents do next. The problem expands from data access to downstream accountability [10].

Control-plane answer. SAL minimizes raw context before reasoning. Agents submit structured data-operation intent. ASCP and OpenKedge evaluate the action against policy, data classification, agency authority, and purpose. VAI records evidence for audit and replay.

Required controls.

- Context minimization and data classification binding.
- Purpose-bound access.
- Approval routing for sensitive data operations.
- Downstream execution tracking.
- Cross-agency evidence and replayable data-use records.

Candidate pilot. A governed analytical workflow where an AI agent proposes a data operation over minimized approved context, without persistent access to underlying national data stores.

Success indicators.

- Complete evidence for data operations.
- Reduced raw data exposure to reasoning agents.
- Traceable policy decisions.
- Replayable cross-agency data-access workflows.

8.3 Playbook 3: DGA-Style Autonomous Public Administration

Operational problem. Digital government workflows are natural candidates for AI acceleration: citizen-service routing, permit workflows, document verification, case management, benefits or eligibility workflows, and inter-agency orchestration. These workflows carry public-sector accountability because they affect citizens, agencies, records, and trust in public services [2].

Control-plane answer. AI agents propose workflow actions. ASCP evaluates authorization, policy, citizen impact, risk, and required escalation. Approved actions execute through contract-bound identity and produce evidence for appeal, audit, and replay.

Required controls.

- Citizen-impact classification.
- Human approval for high-impact actions.
- Policy-bound workflow routing.
- Identity-scoped execution.
- Appeal, replay, and audit support.

Candidate pilot. A non-critical citizen-service routing workflow where the AI proposes routing or document-verification actions and high-impact decisions remain subject to approval.

Success indicators.

- Faster low-risk workflow routing.
- Documented approval paths and replayable evidence.
- Reduced manual triage load.
- Preserved human authority for high-impact cases.

8.4 Playbook 4: NEOM-Style Smart-City Digital Twins

Operational problem. Smart cities merge digital simulation with physical operations. As AI recommendations start influencing real-world mobility, energy, and utilities, the boundary between simulation and execution becomes an important governance boundary [8].

Control-plane answer. SAL separates digital-twin reasoning from real-world execution. ASCP routes proposed actions through policy, simulation status, safety thresholds, risk scoring, approval, contract-bound execution, and evidence capture.

Required controls.

- Simulation-before-execution.
- Physical-action risk classification.
- Safety threshold validation.
- Approval escalation and operational override.
- Rollback planning and evidence for physical-system changes.

Candidate pilot. A smart-city simulation-to-action workflow where AI proposes a low-risk mobility, facility, or energy optimization, but execution remains gated and evidence-backed.

Success indicators.

- Clear separation of simulation from execution.
- Documented safety checks.
- Replayable operational decisions.
- Human override and evidence completeness.

8.5 Playbook 5: Regulated Sectors

Operational problem. Healthcare, finance, energy, logistics, education, and other regulated sectors will use AI agents for triage, compliance, optimization, operations, document processing, fraud review, resource planning, and workflow automation. Each sector has distinct rules and risk thresholds.

Control-plane answer. Use a common execution-governance architecture with sector-specific policy packs. Agents submit structured intent. Policy packs determine admissibility. Execution is bounded by contract and identity. Evidence supports regulator review.

Required controls.

- Sector-specific intent schemas and policy versioning.
- Approval escalation.
- Privacy and data minimization.
- Execution identity binding.
- Regulator-facing evidence, replay, and dispute support.

Candidate pilot. A regulated workflow assistant in a low-to-medium risk process, such as non-clinical healthcare operations, finance compliance triage, energy maintenance planning, or logistics optimization.

Success indicators.

- Policy-pack reuse across workflows.
- Complete evidence records.
- Reduced manual review time for low-risk actions.
- Regulator readiness and reliable escalation.

8.6 Playbook 6: Saudi AI Software Factory

Operational problem. Saudi AI software factories will use generative AI to write code, deployment manifests, and integration adapters at scale. This accelerates delivery, but it requires a much stronger admission boundary than basic compilation or shallow tests.

Control-plane answer. PDD governs what enters the system. Generated artifacts can satisfy protocol-level structural, behavioral, and operational invariants. If admitted, they produce an evidence bundle and deployment contract. Later runtime actions are governed by ASCP and OpenKedge and recorded by VAI.

Required controls.

- Structural, behavioral, and operational invariant checks.
- Generated-artifact provenance.
- Evidence bundle.
- Deployment contract.
- Runtime link to ASCP and OpenKedge.

Candidate pilot. An AI-generated IaC or government workflow artifact pipeline where generated candidates pass PDD admission before deployment.

Success indicators.

- Generated artifacts passing or failing invariant checks.
- Reduced manual review burden for low-risk artifacts.
- Evidence completeness and rollback readiness.
- Runtime governance linkage.

Table 8.1: *Common Sovereign Execution Pattern Across KSA Deployment Contexts*

Control element	Purpose	Applies across
Structured intent	Converts model output into a governable action proposal.	AI cloud, data workflows, government services, smart cities, sectors, software factories.
Context-aware policy	Evaluates action against live state, data sensitivity, sector rules, and organizational authority.	All deployment playbooks.
Risk and operational-impact scoring	Distinguishes low-risk automation from high-impact actions requiring escalation.	Cloud operations, public administration, smart cities, regulated sectors.
Approval routing	Preserves human authority for sensitive or high-impact actions.	Government, smart cities, healthcare, finance, infrastructure.
Execution contracts	Bounds approved actions before credentials are issued.	Runtime automation and deployment pipelines.
Ephemeral identity	Eliminates standing privilege for agents.	Cloud, data, workflow, sector, and software environments.
Evidence chains	Creates replayable accountability.	Auditors, regulators, operators, incident response.
Protocol admissibility	Governs generated artifacts before deployment.	AI software factories, IaC, workflows, integrations.

8.7 Common Pattern Across Deployment Contexts

Across all playbooks, the operating model is the same: the model proposes, the control plane evaluates, the contract bounds, the identity expires, and the evidence chain records what happened. The policy content changes by domain, but the execution-governance structure remains stable.

8.8 Recommended Pilot Selection Criteria

Good first pilots are high value but bounded, measurable, connected to real workflows, isolated from irreversible high-impact actions, evidence-capable, generalizable, operationally acceptable, and compatible with existing identity and audit systems.

Avoid first pilots that involve irreversible citizen-impacting decisions, direct mutation of critical physical systems, broad cross-agency integration on day one, unverified data quality, unclear policy ownership, or weak evidence capture.

The strongest first deployment is useful enough to matter, bounded enough to operate safely, and instrumented enough to teach the next wave of adoption.

Best first pilot: AI-generated infrastructure-as-code admission or a non-critical AI cloud operations sandbox.

The playbooks make sovereign execution concrete across the Kingdom's AI economy. The adoption model that follows moves from sovereign sandbox to bounded production rollout, multi-domain expansion, and national execution fabric.

9 Strategic Adoption Model for KSA

Adoption thesis

Sovereign execution can begin as a bounded adoption path rather than a nationwide big-bang deployment. It can mature through bounded pilots, controlled production rollouts, multi-domain expansion, and national protocol standardization. Each phase can produce reusable policy packs, evidence schemas, operating procedures, and procurement lessons that help the Kingdom scale autonomous AI safely.

The roadmap is incremental: start bounded, prove the loop, build reusable policy, and scale across domains. The first goal is not broad autonomy. The first goal is trusted, replayable autonomy inside a strict operating boundary.

9.1 Phase 1: Sovereign Sandbox

Objective. Validate the sovereign execution loop in an isolated, low-risk environment. Phase 1 proves the loop.

Candidate pilots. AI cloud staging cluster; generated IaC admission pipeline; non-critical digital government workflow; smart-city simulation-only workflow; internal data analysis over minimized context.

Deliverables. Initial intent schema, policy integration, context binding, execution contract format, evidence-chain prototype, replay dashboard, PDD checks, and operator approval workflow.

Success indicators. All pilot actions route through structured intent; policy decisions are recorded; pilot agents hold no standing administrative credentials; evidence chains are complete; pilot actions can be replayed; operators accept the workflow.

Exit criteria. The pilot can show that an AI-generated proposal can be transformed into governed execution with policy decision, contract-bound identity, evidence, and replay.

9.2 Phase 2: Bounded Production Rollout

Objective. Move from sandbox to selected low-to-medium-risk production workflows while preserving approval, rollback, and evidence controls. Phase 2 proves controlled production value.

Candidate workflows. Low-risk AI cloud scaling; non-critical citizen-service routing; data workflow proposals over approved/minimized context; smart-city simulation-to-advisory workflows; generated code or IaC deployment to non-critical environments.

Deliverables. Hardened execution gateway, ephemeral identity integration, approval routing, rollback procedures, replay/audit console, policy versioning, operator runbooks, and incident review.

Success indicators. Low-risk workflows take less manual time; agents have no direct write access; high-impact actions escalate correctly; evidence is accepted by operators and compliance reviewers; rollback and override paths are tested successfully.

Guardrails. No irreversible citizen-impacting decisions without human approval; no direct physical-system mutation without simulation and escalation; no production infrastructure mutation without operational-impact classification; no sensitive data access without purpose-bound policy and evidence.

9.3 Phase 3: Multi-Domain Expansion

Objective. Extend sovereign execution across multiple agencies, clouds, sectors, or operating environments using reusable policy packs and evidence schemas. Phase 3 proves repeatability.

Expansion pattern. Start from the proven pilot domain, generalize intent schemas, create domain-specific policy packs, standardize evidence records, build adapters, train operators and auditors, and use procurement requirements to align vendors.

Deliverables. Reusable intent schemas, sector or ministry policy packs, shared evidence schema, vendor adapter model, regulator and auditor reporting views, common approval patterns, and a cross-domain incident review process.

Success indicators. Policy packs are reused across multiple workflows; evidence is interoperable across systems; new agents and vendors cost less to integrate; cross-domain audit review succeeds; standing privileges are reduced; escalation behavior is consistent.

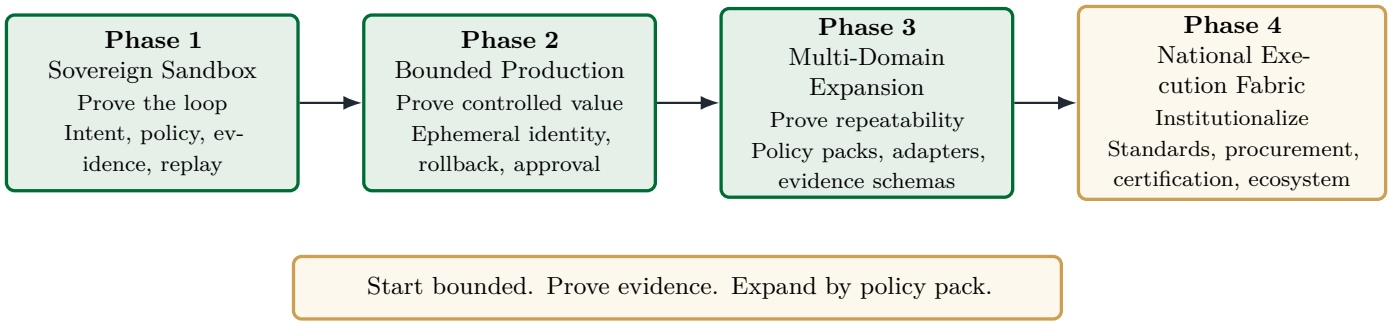

Figure 9.1: *Strategic adoption model for sovereign execution. The recommended path begins with a bounded sovereign sandbox, expands to controlled production workflows, scales across domains through reusable policy and evidence models, and matures into a national sovereign execution fabric.*

9.4 Phase 4: National Sovereign Execution Fabric

Objective. Mature sovereign execution into a reusable national layer for autonomous AI adoption. Phase 4 turns sovereign execution into national infrastructure.

What becomes standardized. Intent schemas, policy-pack structure, execution contract format, evidence-chain schema, replay requirements, identity-scoping patterns, procurement requirements, certification expectations, and reference adapters.

Institutional outcomes. Ministries gain a common governance boundary; HUMAIN-style cloud operators can expose governed agentic operations; regulators can review replayable evidence; local integrators can build against a known protocol surface; vendors can demonstrate compliance; Saudi AI software factories can ship faster under protocol admissibility.

Strategic value. Saudi Arabia is well positioned to become a reference model for governed autonomous AI infrastructure by defining how high-impact AI actions are authorized, bounded, executed, and audited.

9.5 Pilot Selection Framework

Pilot selection should favor workflows that are valuable, bounded, measurable, and evidence-capable.

Table 9.1: *Pilot Selection Criteria*

Criterion	Good first pilot	Poor first pilot
Risk level	Low-to-medium operational risk with clear rollback.	Irreversible citizen-impacting or physical-world action.
Policy clarity	Clear policy owner and approval path.	Ambiguous authority or contested policy rules.
Evidence feasibility	Actions can produce complete intent, policy, contract, identity, and execution records.	Systems lack usable evidence outputs.
Integration scope	One or two target systems with clear APIs.	Broad cross-agency integration on day one.
Measurability	Clear metrics for time, safety, evidence completeness, and operator acceptance.	Vague innovation demo without operational metrics.
Generalizability	Pilot teaches patterns reusable across domains.	One-off demo that does not become a policy pack or adapter.
Human oversight	Escalation path is clear and available.	No responsible operator or review authority.

9.6 Metrics for Executive Oversight

Executive oversight can track whether sovereign execution increases operational velocity while preserving control: actions routed through structured intent; actions blocked by policy; actions escalated to human approval; evidence-chain completeness; approval time for low-risk actions; standing credentials removed; generated artifacts admitted or rejected by PDD; replay success; rollback and override test success; reusable policy packs; and vendor systems integrated through the common protocol boundary.

9.7 Operating Model

Technology is only half the adoption model. The organization needs clear owners for policy, security, data, and incident response.

Who writes the rules? Who approves the high-risk actions? Who reads the evidence when an agent makes a mistake? A successful rollout answers these questions early.

The roadmap shows how sovereign execution can move from architecture to operating reality. The next chapter converts the same principles into a procurement checklist: recommended requirements KSA procurement teams can use with autonomous AI vendors, platforms, and system integrators.

10 Procurement Checklist for KSA AI Decision-Makers

Autonomous AI Procurement

Autonomous AI procurement can evaluate more than model capability alone. For high-impact environments, KSA procurement teams can ask for proof that AI systems operate through structured intent, local policy, scoped identity, execution contracts, tamper-evident evidence, replayable audit, and protocol admissibility for generated artifacts.

Procurement is where architecture becomes enforceable. Alongside "How capable is the model?", KSA procurement teams can ask, "How governable is its execution path?" The checklist applies to autonomous AI platforms, agent frameworks, AI cloud services, government workflow tools, smart-city automation, regulated-sector AI, and AI-generated software pipelines.

10.1 Recommended Minimum Requirements

Minimum Requirements for Autonomous AI Procurement

1. **Can your system submit structured intents instead of directly executing actions?** High-impact AI systems can expose a governable intent boundary rather than requiring direct shell, API, or database write access.
2. **Can execution be blocked by local policy?** Procurement teams can require enforcement of national, sectoral, organizational, and workflow-specific policies before execution.
3. **Can sensitive context be minimized before model reasoning?** The system can support data minimization, redaction, anonymization, and policy-filtered task context.
4. **Can actions be bound to short-lived credentials?** Autonomous agents do not need permanent administrative privilege.
5. **Can high-risk actions require human approval?** Citizen-impacting, infrastructure-impacting, physical-world, or regulated actions can support escalation.
6. **Can every action produce tamper-evident evidence?** Evidence can bind intent, context, policy, approval, contract, identity, execution, and result.
7. **Can auditors replay the decision path?** Operators and auditors can reconstruct why an action was allowed and what happened.
8. **Can the system operate across domestic and foreign models?** Execution governance can be model-agnostic and vendor-agnostic.
9. **Can generated code, workflows, and IaC be checked against invariants?** AI-generated artifacts can pass structural, behavioral, and operational admissibility checks before deployment.
10. **Can the platform operate inside Saudi data, cloud, and compliance boundaries?** Deployment can respect local infrastructure, data, policy, and operational requirements.
11. **Can all mutative actions be routed through a verified control-plane pathway?** High-impact changes can be routed through the execution-governance layer.
12. **Can the institution independently verify execution evidence?** Evidence can support institutional audit, regulator review, incident response, and dispute handling.

10.2 Procurement Scoring Model

The scorecard below converts the checklist into evaluation categories that procurement teams, platform leaders, and architecture review boards can use during RFPs, pilots, and vendor reviews.

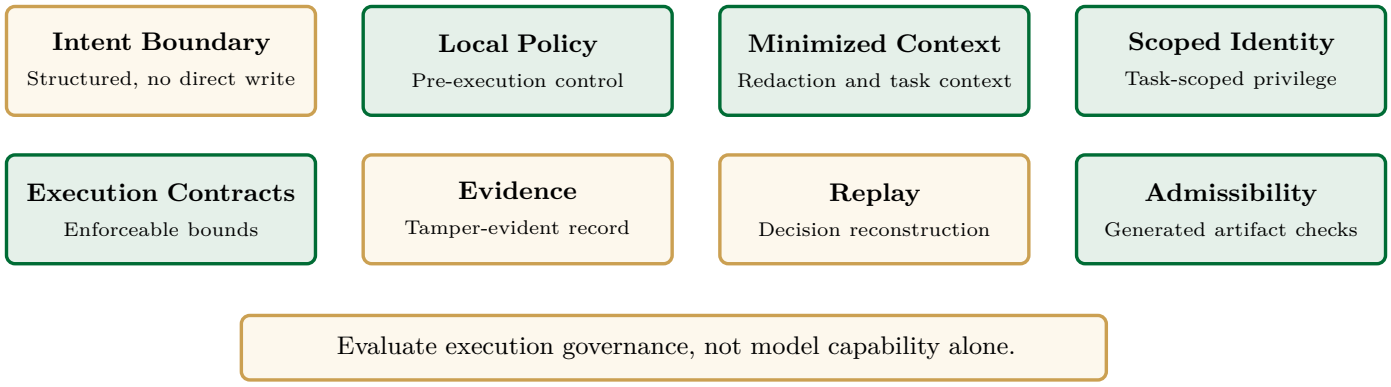

Figure 10.1: *Autonomous AI procurement scorecard.* KSA *procurement teams can evaluate vendors not only on model capability, but on whether autonomous actions are governed through intent, policy, identity, contracts, evidence, replay, and protocol admissibility.*

Table 10.1: *Autonomous AI Execution Governance Scorecard*

Capability Area	What to look for	Red Flag
Intent governance	Structured intent schemas and no required direct write access.	Vendor requires broad API, shell, or database permissions.
Policy control	Local policy evaluation before execution.	Policy is hardcoded, vendor-controlled, or post-hoc only.
Context minimization	Redaction, anonymization, minimized task context.	Vendor requires raw sensitive context for routine reasoning.
Identity & privilege	Short-lived, task-scoped credentials.	Persistent administrative service accounts for agents.
Execution contracts	Machine-enforceable bounds on approved actions.	Approval grants broad runtime authority.
Evidence & replay	Tamper-evident evidence and replayable decision path.	Logs only, no intent/policy/contract linkage.
Generated software	Invariant checks for code, workflow, IaC, and configuration.	Generated artifacts deploy after shallow tests only.
Vendor/model agnosticism	Works with domestic, hyperscaler, open-source, and domain models.	Governance only works inside one vendor's model stack.

10.3 How Procurement Teams Can Use This Checklist

Embed this checklist in RFPs, vendor evaluations, pilot exit criteria, architecture reviews, and compliance reviews. Ask vendors to demonstrate the full path from model proposal to governed execution, including sample evidence records and human escalation for high-impact workflows. Score execution governance alongside model accuracy, user experience, and benchmark performance.

> "Do not procure autonomous execution without evidence."

10.4 Conclusion

This checklist turns the white paper's architecture into procurement action. The final chapter summarizes the strategic recommendation: treating sovereign execution as a national AI infrastructure layer and building an open ecosystem around governed autonomous AI.

11 Strategic Recommendation: Make Sovereign Execution a National AI Layer

Strategic Opportunity

Saudi Arabia's next AI advantage can come from defining the world's clearest architecture for governed autonomy. Sovereign compute, sovereign data, and domestic models establish the foundation. Sovereign execution turns that foundation into safe operational capability by governing what AI systems are allowed to do, under which policy, using which identity, with what evidence, and with what replayable accountability.

Saudi Arabia is building the foundations of a national AI economy: compute, cloud, data, and models. The next strategic layer is sovereign execution: the ability to safely allow AI systems to act across national infrastructure.

Autonomous AI changes the control problem from model hosting to execution governance. Sovereign execution is the control layer between AI ambition and autonomous operation.

The architecture presented in this paper connects SAL, ASCP, OpenKedge, VAI, and PDD into one execution-governance stack for AI cloud, national data governance, public administration, smart-city operations, regulated sectors, and AI software factories.

11.1 Seven Strategic Recommendations

1. **Treat sovereign execution as national AI infrastructure.** Sovereign execution sits alongside sovereign compute, sovereign data, and cybersecurity. It is an infrastructure layer, not a vendor-specific add-on.

2. **Require reasoning/execution separation for high-impact AI.** Models may analyze, recommend, and plan, while high-impact execution can remain inside sovereign environments. SAL provides the architectural pattern: global or domestic reasoning over minimized context, followed by local policy, identity, approval, execution, and evidence.

3. **Adopt intent governance as the default pattern.** Autonomous AI systems can submit structured intent rather than receive direct write access. This creates a standard boundary for policy evaluation, risk scoring, execution contracts, approval routing, and audit.

4. **Use evidence chains as the audit primitive.** Traditional logs remain necessary, but evidence chains add the proof required for autonomous accountability by binding intent, policy, contracts, identity, results, and replay metadata.

5. **Govern AI-generated software through protocol admissibility.** As AI-generated code, infrastructure-as-code, workflows, policies, and configurations become common, generated artifacts can be admitted through structural, behavioral, and operational invariants. PDD governs what enters the system; ASCP governs what acts inside it.

6. **Start with bounded reference deployments.** The first deployments can be high-value but bounded: AI cloud operations sandbox, non-critical digital government workflow, minimized-context data workflow, smart-city simulation-to-action pipeline, or AI-generated IaC admission pipeline. They should prove intent, policy, identity, evidence, and replay before expanding.

7. **Build an open Saudi ecosystem around sovereign execution.** A common open protocol boundary can support ministries, HUMAIN-style AI cloud operators, SDAIA-style data governance, DGA-style workflows, NEOM-style smart-city systems, regulated sectors, local integrators, hyperscalers, and AI vendors. The strategic value lies in the ecosystem around that boundary.

11.2 The Final Architecture

Together, these layers define a sovereign execution stack: reasoning is separated, intent is governed, execution is bounded, identity is scoped, evidence is recorded, generated artifacts are admitted before production, and replay feeds governance improvement.

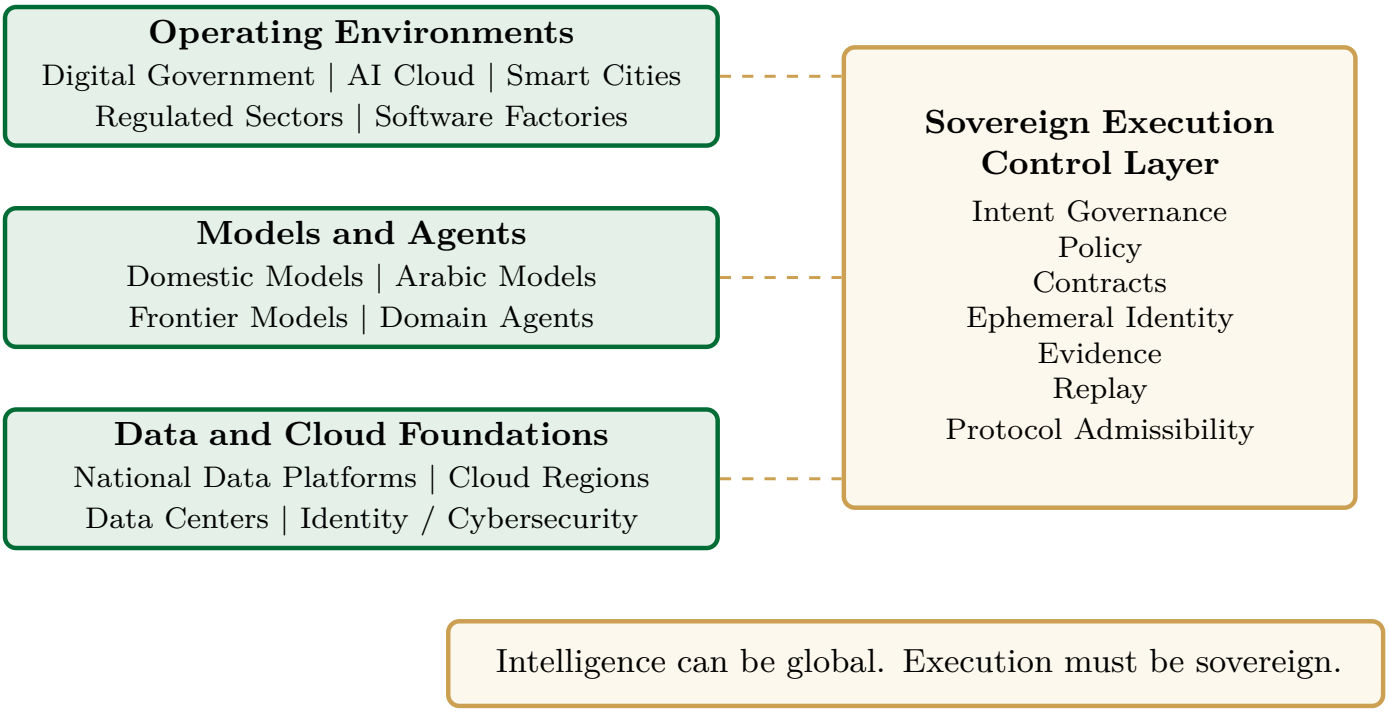

Figure 11.1: *Sovereign execution as a national AI layer. Compute, data, models, and applications establish the foundation; sovereign execution governs high-impact autonomous actions across ministries, cloud operators, smart-city environments, regulated sectors, and software factories.*

- SAL separates reasoning from execution.
- ASCP provides the macro control plane.
- OpenKedge provides the intent-governance protocol.
- VAI provides evidence, identity, audit, and replay.
- PDD governs generated software and infrastructure artifacts before deployment.

11.3 Closing Statement

This is the strategic role of sovereign execution: converting AI infrastructure into governed autonomous capability.

The next decade of AI leadership will not be determined by GPUs, model parameters, or benchmark scores alone. It will be determined by the ability to let AI act safely inside real institutions and critical infrastructure.

The next global AI leader will not simply be the nation with the largest models or the most GPUs. It will be the nation that can safely allow AI to act.

Saudi Arabia has the opportunity to define that blueprint.

Bibliography

[1] Amazon Web Services and HUMAIN. AWS and HUMAIN announce a more than $5B investment to accelerate AI adoption in Saudi Arabia and globally. https://www.aboutamazon.com/news/company-news/amazon-aws-humain-ai-investment-in-saudi-arabia, 2025. Accessed May 2026.

[2] Digital Government Authority. Digital Transformation. https://dga.gov.sa/en/digital-transformation, 2022. Accessed May 2026.

[3] Jun He. The Autonomous State Control Plane: A Reference Architecture for Sovereign AI Systems. White paper, The OpenKedge Initiative, 2026.

[4] Jun He and Deying Yu. OpenKedge: Governing Agentic Mutation with Execution-Bound Safety and Evidence Chains. *arXiv preprint arXiv:2604.08601*, 2026.

[5] Jun He and Deying Yu. Protocol-Driven Development: Governing Generated Software Through Invariants and Evidence. *arXiv preprint arXiv:2605.12981*, 2026.

[6] Jun He and Deying Yu. Sovereign Agentic Loops: Decoupling AI Reasoning from Execution in Real-World Systems. *arXiv preprint arXiv:2604.22136*, 2026.

[7] Jun He and Deying Yu. Verifiable Agentic Infrastructure: Execution Identity and Evidence Chains at Scale. *arXiv preprint arXiv:2605.15228*, 2026.

[8] NEOM. Technology and Digital. https://www.neom.com/en-us/our-business/sectors/technology-and-digital, 2026. Accessed May 2026.

[9] Public Investment Fund. HRH Crown Prince launches HUMAIN as global AI powerhouse. https://www.pif.gov.sa/en/news-and-insights/press-releases/2025/hrh-crown-prince-launches-humain-as-global-ai-powerhouse/, 2025. Accessed May 2026.

[10] Saudi Data and AI Authority. National Data Lake. https://lake.data.gov.sa/en, 2026. Accessed May 2026.

[11] Saudi Data and AI Authority. Our Strategies and Initiatives. https://sdaia.gov.sa/en/SDAIA/SdaiaStrategies/pages/default.aspx, 2026. Accessed May 2026.

[12] Saudi Press Agency. SDAIA Issues Year of Artificial Intelligence 2026's Guidelines to Unify National Efforts and Showcase Saudi Leadership in Advanced Technologies. https://spa.gov.sa/en/N2546742, 2026. Accessed May 2026.