From Rights to Runtime: Privacy Engineering for Agentic AI

KEIVAN NAVAIE, Lancaster University; The Alan Turing Institute, UK, United Kingdom

Agentic AI shifts stacks from request–response to plan–execute. Systems no longer just answer; they act—planning tasks, calling tools, keeping memory, and changing external state. That shift moves privacy from policy docs into the runtime. This opinion piece argues that we do not need a new privacy theory for agents; we need enforceable, observable controls that render existing rights as product behavior. Anchoring on GDPR—with portable touchpoints to CPRA, LGPD, and PDPA, we propose a developer-first toolkit: optional, bounded, user-visible memory; a purpose-aware egress gate that enforces minimization and transfer rules; proportional safeguards that scale with stakes; and traces that tell a coherent story across components and suppliers. We show how the EU AI Act's risk management, logging, and oversight can scaffold these controls and enable evidence reuse. The result is an agentic runtime that keeps people in control and teams audit-ready by design.

Additional Key Words and Phrases: Agentic AI, privacy, GDPR, AI Act, data protection by design, traceability, accountability

1 Introduction

Agentic AI does not need a new privacy theory; it needs runtime controls that render rights observable and enforceable. Our stance is pragmatic: take settled principles and express them as product behavior that can be tested in production. We focus on four design patterns that make rights concrete in the agent runtime: *memory governance* that is optional, bounded, and visible, with cascade erasure and receipts; *purpose-aware egress* that checks purpose, necessity, and region on every tool call and records an auditable decision; *proportional safeguards* that scale with stakes, from read-only defaults to human involvement before consequential actions; and *end-to-end traceability* that links each step to purpose, lawful basis, fields, recipients, regions, transfers, and outcomes. GDPR is the baseline for terminology; where relevant we note how traces and oversight align with AI Act duties for high-risk systems, including record keeping and human oversight. The rest of this section sets the context; the remainder of the paper shows how these controls fit the agentic runtime in practice.

Agentic AI is upgrading our stacks from request/response to plan/execute. We are moving from systems that answer to systems that act, from single turn replies to multi step plans that call tools, retain context, and change external state. That shift turns assistants into collaborators. It also sharpens a core question: when an AI plans, reads, writes, and remembers, whose data moves where, for what purpose, and under which set of privacy rights?

Privacy rights are protected worldwide under frameworks such as the EU and UK GDPR, California's CCPA/CPRA, Brazil's LGPD, Canada's PIPEDA, Singapore's PDPA, and sector specific rules like HIPAA for US health data. Names and thresholds differ, but most converge on transparency, purpose limitation, data minimisation, security, accountability, and enforceable individual rights, with scope and remedies varying by jurisdiction.

For a clear baseline, we anchor this discussion in GDPR, not as an appeal to authority but because the Brussels Effect has made its approach a de facto benchmark beyond the EU [1, 14]. GDPR is principles based, technology neutral, risk proportional, and rights centric, with extraterritorial scope and credible supervision and remedies [1, 6]. Even outside the EU, these properties give teams a shared language for autonomy, memory, and tool use that travels well between engineering, product, and policy.

Author's Contact Information: Keivan Navaie, Lancaster University; The Alan Turing Institute, UK, United Kingdom, knavaie@turing.ac.uk.

This article is intentionally high level and aimed at AI developers. It is for information and awareness, not legal advice or a definitive guide to data protection. Practitioners should adapt any approach to the laws, standards, and regulatory guidance that apply in their jurisdiction and sector, and seek qualified counsel where appropriate.

In what follows, we outline GDPR in plain terms and clarify what we mean by agentic AI. We then contrast agentic stacks with conventional request–response systems and explain the implications for data flows, roles, and accountability. Next, we translate privacy rights into high level, runtime oriented steps developers can apply today, including memory controls, readable activity logs, purpose aware egress, supplier governance, traceability, proportional safeguards, and meaningful human involvement. We then show how the EU AI Act can support these controls through risk management, logging, and oversight, and how teams can map evidence once and reuse it. We refer to the EU AI Act as Regulation (EU) 2024/1689 (OJ L, 12 July 2024). All article and annex references are to Regulation (EU) 2024/1689. We close with actionable roles for developers, regulators, standardization bodies, and end users so each has a share in a trustworthy agentic AI ecosystem.

What you will learn: You will learn how to make privacy rights executable by implementing four design patterns: memory governance, purpose aware egress, proportional safeguards, and end to end traceability, supported by a narrative example, a portability map, and a one page implementation checklist.

2 What is GDPR?

Think of the EU and UK GDPR as a risk based governance layer that wraps any processing pipeline touching personal data. It travels with the data across vendors and borders and it scales duties with impact. At its core are seven principles that shape everyday design: lawfulness, fairness, and transparency so you use a clear legal basis and explain it; purpose limitation so you use data only for stated aims; data minimisation so you collect no more than necessary; accuracy so you keep it correct; storage limitation so you delete on time; integrity and confidentiality so you protect it appropriately; and accountability so you can show your working [1].

According to GDPR, people hold runtime controls that your product must respect. They can request access, rectification, erasure, restriction, portability, and objection. They also have a qualified right not to be subject to a decision based solely on automated processing that produces legal or similarly significant effects [1, 3]. These rights are not footnotes. They are surfaces you should expose in product, with evidence behind them.

Responsibility is shared across the lifecycle, but not symmetric. Controllers set the purpose and essential means of processing and carry primary accountability. Processors act on the controller's documented instructions under an Article 28 data processing contract that defines scope, security, sub processors, and how data are deleted or returned at end of life [4]. Governance blends external enforcement by independent regulators with internal duties across the build–ship–run cycle: maintain records of processing during design and build; Use the WP29/EDPB Data Protection Impact Assessment (DPIA) Guidelines (WP248 rev.01) to decide when a DPIA is mandatory and how to scope it (apply the nine high-risk criteria; document measures, residual risk, and sign-off) [5]; meet security and breach notification obligations in operation and monitoring; and, where useful, adopt codes of conduct or certification to drive continuous improvement [1].

3 Agentic AI: Architecture and Key Differences

Unlike the term *AI system* defined in the EU AI Act Article 3(1), *agentic AI* has no settled legal definition. We use it as a practical shorthand for systems that plan, remember, and act through

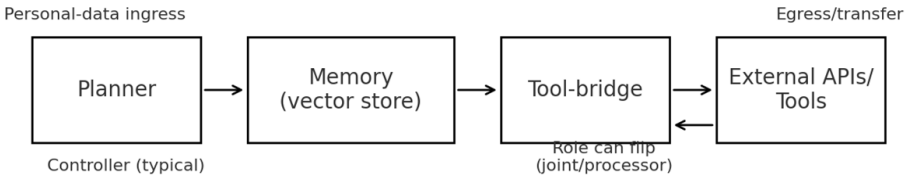

Fig. 1. Agentic AI architecture with personal data ingress and egress. Inside the boundary sit the planner, model, memory, and a policy enforcing tool gateway; around the edge are external tools and APIs. A trace layer spans all components.

tools, and we examine how personal data flows through these systems and how responsibility can shift across stages [2].

Figures 1 and 2 depict, respectively, a generic agentic system and a typical agentic stack, highlighting personal data ingress and egress points. A **planner** interprets intent and selects steps. The **model** handles language and reasoning. A **memory** layer carries context across turns as transcripts, summaries, or embeddings, which raises questions about minimisation and retention. A **tool gateway** mediates calls to internal and external services and applies policy before any data leave the boundary. Around the edge sit **tools and APIs** such as calendar, payments, and booking. Ingress occurs at user input or internal reads. Egress occurs when a tool call passes the gateway. Roles can change by stage, so a team that determines purpose and essential means at one step may act as a controller for that step, while a tightly directed service may act as a processor downstream. The aim is to make these transitions visible rather than implicit.

What makes these systems different from largely stateless model APIs is the runtime. Four shifts matter in practice: behaviour can adapt during deployment as new inputs arrive; state persists at inference so context survives across interactions; operational autonomy includes invoking tools and taking write actions that change external state; and orchestration is dynamic and modular so the processing topology can change at runtime as different tools are selected [15–18]. Not every agent implements all four, and many express them only in part, but these shifts determine how personal data moves, who decides what, and where accountability must be shown [1, 3].

4 From answers to actions

Agentic AI runs as a goal driven runtime. Given an intent, the system decomposes tasks, chooses and calls tools, maintains context across steps, and adapts as new signals arrive. Privacy is enforced in this loop, not only in policy text. The controls are therefore operational: they plan, read, write, and remember in production. This is where personal data are ingested, transformed, and disclosed, and where purpose, necessity, region, and retention must be enforced in real time. To keep the section practical, each subsection ends with a short running example tied to the travel concierge in Section 7. For portability across jurisdictions, Table 1 maps the controls in this section to GDPR, CPRA, LGPD, and PDPA.

4.1 Memory that earns trust

Memory should be optional, bounded, and visible. Keep account memory off by default. Offer session only memory for continuity inside a single task and a time boxed account memory for convenience across tasks. Label each memory item with purpose and time to live. Provide edit and remove actions that propagate to transcripts, summaries, and embeddings. Confirm completion of cascade erasure. Short defaults reduce risk while preserving utility [1].

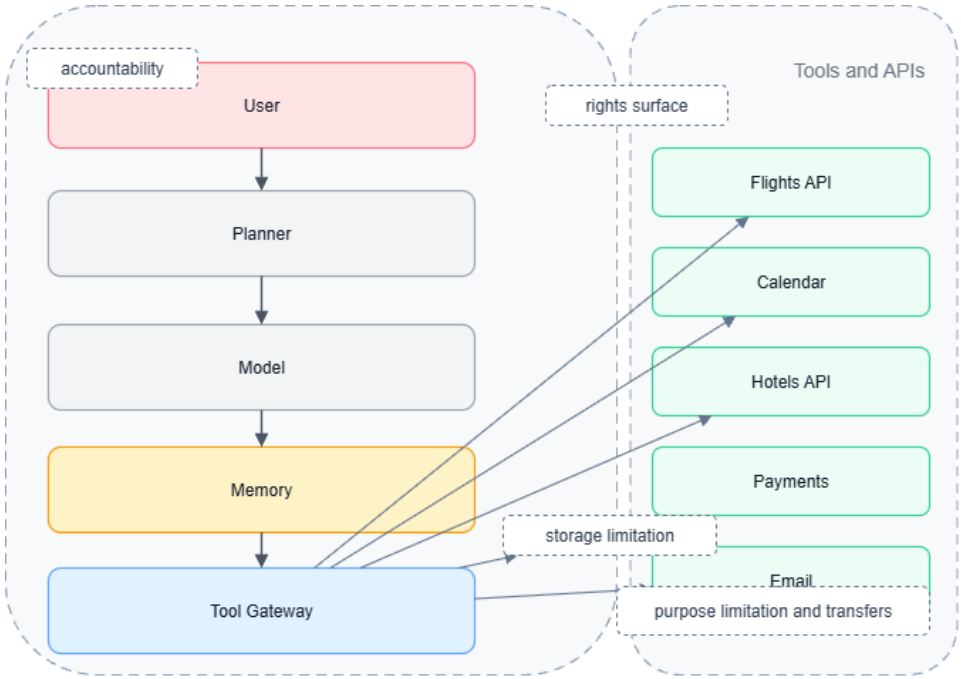

Fig. 2. Exploded agentic stack with privacy touchpoints. User, Planner, Model, Memory, and a Tool Gateway are surrounded by Tools and APIs; a translucent Trace and Evidence layer spans all layers. Tags indicate rights surface at the user panel, storage limitation at memory, purpose limitation and transfers at the gateway, and accountability at trace.

Running example (travel concierge): The concierge enables session memory for the Paris task. Account memory remains off until the user opts in for 30 days. Each item shows purpose trip planning and TTL 30 days. Deleting an item triggers erasure across transcripts, summaries, embeddings, caches, partner tools, and analytics, with a receipt when complete.

4.2 Autonomy with boundaries

Before any tool call leaves the boundary, evaluate three questions that follow core principles. Is the proposed use compatible with the declared purpose. Are the requested fields necessary for that purpose. Is the destination permitted, including region and the applicable transfer mechanism [1, 7, 13]. If any test fails, block the call and return a clear explanation. If a conditional path exists, allow with redaction or narrowed scope and log the reason. If the call passes cleanly, allow and log. Place an approved tool registry behind the gate. Record for each tool what it does, fields it receives, regions it operates in, retention properties, and how people exercise rights. Treat the registry like code and version it.

Running example (travel concierge): Using the policy in Listing 1, before calling a hotel API, the gate checks purpose trip planning, necessity dates and room type, and region EU. Passport number is unnecessary, so the call is allowed with redaction. A non EU vendor without an appropriate transfer tool is blocked with an explanation and an entry in the trace.

Listing 1 makes Fig. 4 implementable with a minimal, policy-driven gate.

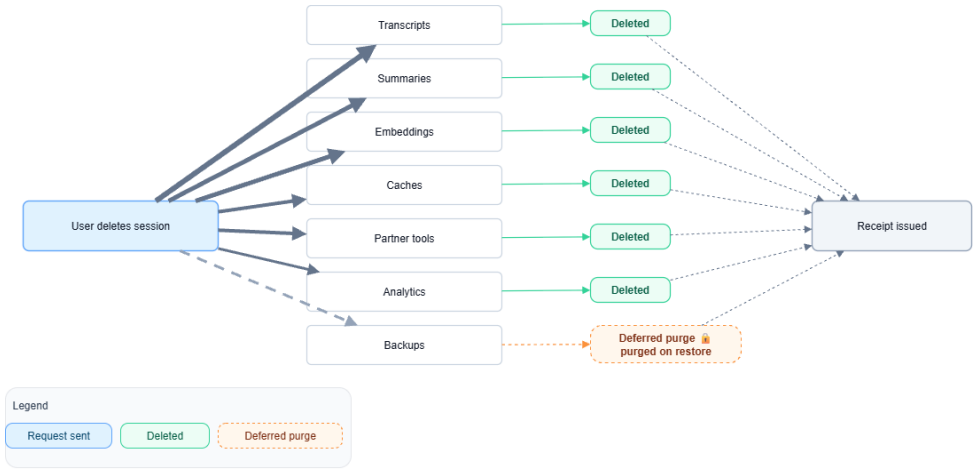

Fig. 3. Deletion cascade Sankey. Erasure reaches transcripts, summaries, embeddings, caches, partner tools, analytics, and backups. The backups branch is dotted to indicate deferred purge with purge on restore. A receipt is issued when confirmations arrive.

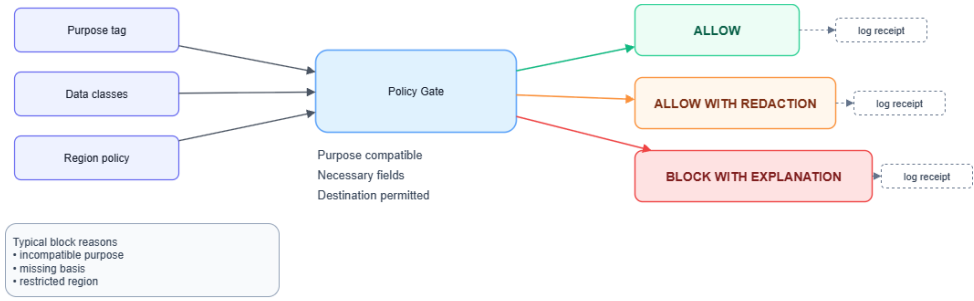

Fig. 4. Purpose aware egress gate. Inputs are purpose, data classes, and region policy. Outcomes are allow, allow with redaction, or block with explanation. Each outcome drops a receipt for the trace.

Listing 1. Minimal egress gate (pseudo-code). Inputs: purpose, fields, region, dest. Outputs: Allow(), AllowWithRedaction(...), Block("reason").

```
POLICY = {"trip_planning:hotel_api": {
  "allow": ["city","dates","room_type"],
  "redact": ["passport_number"],
  "regions": ["EU"]
}}
# Invariant: deny-by-default when no matching rule exists; the gate fails closed.

def egress_gate(purpose, fields, region, dest):
  rule = POLICY.get(f"{purpose}:{dest}")
  # Enforce invariant: if no rule matches, deny by default
  if not rule: return Block("no_rule")
  if region not in rule["regions"]: return Block("region_not_permitted")
  extra = fields - set(rule["allow"]) - set(rule["redact"])
  if extra: return Block("unnecessary_fields:"+",".join(sorted(extra)))
  to_redact = fields & set(rule["redact"])
  if to_redact: return AllowWithRedaction(to_redact)
```

```
  return Allow()

# Example: purpose=trip_planning, fields={city,dates,passport_number}, region=EU, dest=hotel_api
# Output: AllowWithRedaction({"passport_number"})
```

4.3 Proportional safeguards that scale with stakes

Controls should scale with stakes so low risk assistants are not over engineered and high risk agents are not under engineered. A simple ladder helps. At a baseline tier, allow read only tools and ephemeral memory, keep processing in region where possible, and provide a user visible activity log. At a heightened tier, allow write actions such as creating tickets, add time boxed account memory, require deny by default egress with purpose and region checks, and step up audit frequency. At a strict tier, bring in human review before impactful actions, shorten retention windows further, isolate tenants strongly, require formal assessments for cross border flows, and monitor continuously. For high risk systems under Article 6 and Annex III of the AI Act, logging and oversight duties in Article 12 and Article 14 apply [2].

Running example (travel concierge): Itinerary suggestions run at the baseline tier. Auto booking uses the heightened tier with deny by default egress. Refund handling runs at the strict tier with mandatory review. The tier is recorded in the trace and surfaced in the log.

4.4 Traces that tell a coherent story

Treat trace data as a first class component. For each step, record the tool invoked, fields exchanged, declared purpose and lawful basis, region, any transfer mechanism, the decision taken, and the outcome. Link entries to the tool registry so supplier information is one click away. Timestamp events and keep a minimal dependency chain so investigators can reconstruct what happened without custom forensics. The same trace supports GDPR accountability and, where the system is high risk under Article 6 and Annex III, the AI Act requirements for record keeping and human oversight, specifically Article 12 and Article 14 [1, 2]. The implementation checklist in Section 8 summarises the mappings and tests that keep the trace useful in practice.

Running example (travel concierge): For the Paris booking flow, the trace shows that the hotel API received dates and city only, that the call was allowed with redaction due to purpose and region checks, that payment used a processor in the EU under a recorded mechanism, and that the user deleted the session which triggered a confirmed cascade.

4.5 Roles and responsibility without ceremony

Responsibility is shared across the lifecycle, but not symmetric. A party that determines the purpose and essential means for a step is a controller for that step. A tightly directed service that acts on documented instructions is a processor for the downstream step. Joint controllership can arise for particular stages such as collection via embedded tools. The safest way to keep this legible is to maintain a stage by stage role map tied to the actual workflow, not only to contracts. The map should be anchored in the record of processing and reflected in the trace [4].

Running example (travel concierge): For plan → retrieve → book → pay → notify, the concierge acts as controller for planning and booking. The payment gateway acts as processor under an Article 28 agreement. If an external aggregator sets secondary purposes, it is a downstream controller for that stage. The trace links each stage to the role map.

4.6 Security as applied discipline

When agents can act, security defends a live, tool using runtime. Training data extraction is one risk [19]. Two others require first class treatment in practice: prompt injection and insecure output handling [22, 23]. Treat all inbound content and tool responses as untrusted. Resolve instruction precedence so system and tenant policies take priority. Bind model outputs to typed parameters and validate against strict contracts before any API call or file operation. Run external fetches in sandboxes that restrict the network and file system. Proxy outbound requests and block access to private ranges and sensitive metadata endpoints. Keep block events and reasons in the trace. Use a lifecycle taxonomy to plan tests and mitigations across data, model, and application layers. NIST AI 100 2 situates poisoning, evasion, privacy inference, reconnaissance, and abuse, and maps them to safeguards including secure data pipelines, robust training and evaluation, red teaming, and runtime detection and response [24]. ENISA's threat landscape provides context for attacker incentives and technique shifts over time [25].

Running example (travel concierge): The concierge fetches an itinerary via a sandboxed browser. Untrusted content cannot reach internal metadata endpoints. Prompts that try to override policy are neutralised. Model outputs are parsed into typed fields and validated before any booking call is made. Blocked events are visible in the trace for follow up.

4.7 Significant effects deserve meaningful involvement

Where an output may have legal or similarly significant effects on a person, provide meaningful human involvement. The system should pause before consequential actions, route the case to a reviewer with the minimal context needed to decide, record the reviewer's authority, and make challenge routes clear to the person [1, 3]. These safeguards are good practice regardless of formal thresholds and help align with oversight duties in high risk settings under the AI Act.

Running example (travel concierge): Auto cancel of a non refundable booking is treated as consequential. The concierge pauses and requests human confirmation. The reviewer sees the reason, the relevant trace, and the ability to approve or reverse. The user can challenge the outcome through the same surface.

Takeaway: No new privacy theory is required; enforce rights at runtime with short, visible memory, a purpose aware egress gate, safeguards that scale with stakes, and traces that make decisions auditable.

5 The AI Act as helpful scaffolding

Many teams will prepare for the AI Act alongside privacy obligations. For *high-risk* systems as defined in Article 6 and Annex III, Article 12 (record keeping) and Article 14 (human oversight) are particularly relevant, alongside Article 9 (risk management), Article 10 (data and data governance), and Article 11 (technical documentation). Cross map artifacts rather than duplicate effort. The execution traces that support GDPR accountability also support record keeping and traceability for high risk systems. Human in the loop plans that avoid solely automated significant effects also serve as human oversight measures. Risk management, data governance, technical documentation, and post market monitoring become easier to demonstrate when purpose aware logging and mediated egress are already in place [2].

6 Implementation challenges and trade offs

Turning rights into runtime controls introduces real engineering costs. The aim is to ship guardrails that are effective, observable, and sustainable in production.

- **Cost:** Always-on tracing, policy evaluation, and deletion cascades add compute, storage, and operations overhead. Keep traces structured and compact, sample low-risk events, batch receipts, push redaction upstream so less data flows, and reuse evidence across frameworks so one artifact set serves GDPR and AI Act needs.
- **Latency:** A purpose aware egress gate and inline redaction put work on the critical path. Reduce impact by colocating the gate with the data plane, caching allow decisions with short TTLs keyed by purpose and destination, and using asynchronous logging with durable buffers. Track p95 and p99 and set service level objectives the team can meet.
- **Reliability:** Gates can fail closed and block work or fail open and miss violations. Treat policy as code with unit tests and replayable fixtures. Run shadow mode, then canary, before full block. Keep the gateway stateless and horizontally scalable. Provide fallbacks such as read only mode, alternate suppliers, and time bound overrides with audit.
- **Deletion and memory:** Cascade erasure is a distributed systems problem. Model it as an event driven DAG with retries, idempotency, and backoff. Separate fast deletes for active stores from deferred purge for backups with purge on restore. Issue receipts only after downstream confirmations.
- **Human in the loop:** Reviews introduce queueing and staffing load. Define routing rules, service levels, and escalation paths. Surface the minimal context needed to decide. Measure latency to review and the reversal rate to tune thresholds.
- **Vendors and regions:** Tool calls cross trust and geography boundaries. Maintain a signed tool registry with scopes, regions, retention, and contacts. Enforce least privilege scopes and short lived tokens. Prefer regional endpoints and document transfer mechanisms where cross border processing is required.
- **Operations:** Add kill switches, circuit breakers, and budget caps for spend and write scope. Monitor precision and recall of egress decisions, deletion completion rates, and drift in memory TTLs. Schedule chaos drills for purpose-aware egress gate outages and simulated partner failures so the system degrades safely and recovers predictably.
- **Rollout playbook:** Shadow → canary → full block; promote only when precision/recall of egress decisions, p95/p99 latency, and false-block rate meet targets; time-bound overrides are logged and audited.

These trade offs are real, yet manageable with disciplined engineering. The same artifacts that make enforcement observable also reduce audit friction and enable evidence reuse across privacy and safety frameworks.

7 A narrative example, now with rights in view

Consider a travel concierge agent. The user asks for the best weekend in Paris next month. The system declares the purpose as trip planning and turns on session memory for the task. It proposes to check flights, hotels, calendar, and a payment method. Each step passes through the purpose-aware egress gate. The gate confirms that only necessary fields will be shared, that destinations match policy, and that retention promises are in place. We apply the controls in Table 1 to keep the flow portable across GDPR, CPRA, LGPD, and PDPA.

When the booking completes, the person sees an activity log that lists tools used, categories of data exchanged, regions involved, and outcomes. If they delete the session, the system begins a cascade through transcripts, summaries, embeddings, caches, and partner stores. When the cascade finishes, the system issues a confirmation. If an action would have significant effects, the agent requires step up approval. The experience remains quick and helpful. The rights story is visible inside the product, not hidden in footnotes. That is how adoption grows without surprises.

Table 1. Runtime controls mapped to GDPR, CPRA, LGPD, and PDPA (portable, high-level view). *Scope note: AI Act duties such as Article 12 and Article 14 apply where the system is classified as high-risk under Article 6 and Annex III.*

Control	GDPR	CPRA	LGPD	PDPA
Transparency & trace	Arts. 12–14; 5(2) [1]	Notice; right to know [26]	Transparency; Art. 18 [27]	Notification; Access/Correction [29]
Purpose-aware egress	5(1)(b); Art. 6 [1]	Necessary & compatible (§1798.100(c)) [26]	Purpose; adequacy [27]	Purpose; Transfer limits [29]
Minimisation & storage	5(1)(c),(e) [1]	Minimisation; retention [26]	Necessity; retention [27]	Retention limits [29]
Rights surface	Arts. 15–22 [1]	Know, delete, correct, portability; opt-out [26]	Art. 18 [27]	Access; Correction [29]
Automated decisions	Art. 22; EDPB ADM [1, 3]	ADMT (rulemaking) [28]	Art. 20 [27]	PDPC model guidance [30]

8 Implementation checklist

This checklist maps each right or principle to a runtime control, key artifacts, and basic tests or service levels. It is a grab-and-go summary for developers and reviewers.

9 Related technical foundations

Beyond regulatory definitions, there is a complementary body of technical research in ACM and IEEE that explores how privacy and accountability can be embedded into system architectures. Foundational work in privacy engineering emphasized the need to operationalize abstract rights into usable processes and system design [31, 32]. Privacy-preserving identity management approaches similarly highlighted bounded retention, optionality, and user-visible control [33]. From the AI systems perspective, safety and observability have been framed as core runtime challenges, with proposals for intervention points to mitigate unintended side effects [39]. Complementary research on information accountability argues that logging and traceability should function as first-class privacy mechanisms [34], an idea now echoed in transparency artifacts such as FactSheets and Model Cards [36, 38]. Parallel advances in ML systems engineering propose structured readiness assessments that scale safeguards with operational risk [35]. Finally, fairness-oriented audits emphasise that accountability mechanisms must be complemented by meaningful human oversight [37]. Together, these contributions demonstrate that privacy-by-design for agentic AI represents an emerging synthesis of regulatory principles and technical practices.

10 Closing thought

Agentic AI adds real capability to everyday work. It does not need a new privacy theory. The principles are known. The engineering task is to turn them into runtime controls. Short, visible memory. Purpose aware orchestration. A policy enforced egress layer. Risk tiered safeguards. End to end tracing that explains who did what, when, where, and why. The difference between a demo and a durable system is whether it shows its work and keeps people in control.

Looking ahead, several research threads can raise the bar for practice. Formal specifications and verification of purpose limitation, necessity, and region constraints at design time and at runtime; shared benchmarks and simulation datasets that pair agent tasks with labeled data rights events,

Table 2. Rights and principles mapped to runtime controls, build artifacts, and tests.

Right / Principle	Control pattern	Primary artifacts to build	Tests / Metrics	Notes
Transparency & accountability	Activity log + execution trace	Log UI; trace schema; decision receipts; tool registry linkbacks	100% steps logged; schema validation; trace→tool-registry join success	Make logs human-readable; traces machine-verifiable
Access (Art. 15)	Self-service data viewer	Viewer UI; export API (CSV/JSON)	Export completeness vs. trace; export latency SLO	Include fields, purposes, recipients, regions
Rectification (Art. 16)	Editable memory/profile	Memory UI; edit endpoints; propagation hooks	Edit→propagation success to summaries/embeddings within SLA	Show what changed; keep minimal diff in trace
Erasure (Art. 17)	Deletion cascade	Erasure DAG; partner webhooks; receipts	p95 completion time; downstream acks; backup "purge on restore" proof	Dotted "deferred purge" for backups
Restriction (Art. 18)	Processing pause switch	Per-subject/process toggle; policy flag in purpose-aware egress gate	Gate blocks on restriction; no egress while active	Show status in UI; record unblocked time
Portability (Art. 20)	Machine-readable export	Export pipeline; stable schemas; checksums	Round-trip parse test; hash match; coverage vs. memory	Prefer open, documented schemas
Objection (Art. 21)	Preference center + policy engine	Purpose flags; deny rules; consent log	Deny rule hit-rate; false-allow rate; audit trail	Tie to purpose-aware egress gate decisions
Automated decisions (Art. 22)	Meaningful human involvement	Review queue; reason surface; override control; audit trail	p95 time-to-review; reversal rate; reviewer context completeness	Trigger on consequential writes
Purpose limitation	Purpose-aware egress gate	Policy-as-code; tool registry; allow/deny with reason	Precision/recall on allow/deny; drift alerts on purpose tags	Log "why allowed/blocked"
Data minimisation	Field-level selectors + redaction	Contracts for tool I/O; field maps; redactors	Field over-collection rate; redaction correctness	Bind model output to typed params
Storage limitation	TTLs + retention jobs	TTL labels; retention scheduler; receipts	TTL drift monitor; late-delete rate; p95 retention	Short defaults; visible TTLs
Integrity & confidentiality	Sandboxes, proxies, scopes	Sandboxed fetcher; egress proxy; token scopes; kill switches	OWASP LLM01/02 test pass; blocked-call logs; pen-test fixes	Treat tool outputs as tainted until validated
International transfers	Region policy + transfer mechanism	Region map; SCC/DPF records; gateway checks	Region misroute rate; mechanism recorded per transfer	Name countries; cite mechanism in trace
Accountability (Art. 5(2))	Record of Processing Activities (ROPA) + trace linkage	ROPA entries; trace→ROPA keys	Coverage: trace steps linked to ROPA; periodic review	Evidence reuse for audits
Risk & oversight (AI Act HR)[1]	Logging + human oversight plan	High-risk log set; HITL SOP; drills	Log coverage vs. Art. 12; HITL readiness vs. Art. 14	Applies where system is high-risk

Abbrev. HR: high-risk. HITL: human in the loop. ROPA: record of processing activities. TTL: time to live.

transfers, and deletions; evaluation frameworks and metrics for runtime rights compliance, including precision and recall of egress decisions, completeness of deletion cascades, and latency to human review for significant effects; interoperable trace and tool registry schemas with conformance tests so evidence travels across vendors; and methods for verifiable erasure and memory governance, for example cryptographic receipts, purge on restore proofs, and privacy preserving embeddings. Progress on these fronts can make rights not only visible, but reliably testable.

Disclaimer: This article is intended to inform and raise awareness. It is not legal advice and it is not a definitive guide to data protection. Practitioners should interpret and implement any approaches in line with the laws, standards, and regulatory guidance that apply in their own region and sector, and seek qualified counsel where appropriate.

References

[1] European Parliament and Council. 2016. Regulation (EU) 2016/679 (General Data Protection Regulation). *Official Journal of the European Union* L119 (2016). Available at EUR-Lex.

[2] European Parliament and Council. 2024. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union*, OJ L, 12 July 2024. https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng

[3] European Data Protection Board. 2018. Guidelines on Automated Individual Decision-Making and Profiling for the purposes of Regulation 2016/679 (WP251 rev.01, endorsed). EDPB, Brussels.

[4] European Data Protection Board. 2021. Guidelines 07/2020 on the concepts of controller and processor in the GDPR (final). EDPB, Brussels.

[5] Article 29 Working Party. 2017. Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is "likely to result in a high risk" for the purposes of Regulation 2016/679 (WP248 rev.01). Brussels: European Commission. Endorsed by the European Data Protection Board on 25 May 2018.

[6] European Data Protection Board. 2019. Guidelines 3/2018 on the territorial scope of the GDPR (Article 3). Final version. EDPB, Brussels.

[7] European Data Protection Board. 2021. Recommendations 01/2020 on measures that supplement transfer tools to ensure compliance with the EU level of protection of personal data. Final version. EDPB, Brussels.

[8] European Commission. 2023. Commission Implementing Decision (EU) 2023/1795 on the adequate level of protection of personal data under the EU–US Data Privacy Framework. *Official Journal of the European Union* L231 (2023), 118–229.

[9] United Kingdom. 2023. The Data Protection (Adequacy) (United States of America) Regulations 2023 (SI 2023/1028). London: The Stationery Office. In force 12 October 2023. https://www.legislation.gov.uk/uksi/2023/1028/contents/made

[10] Court of Justice of the European Union. 2019. Case C-40/17 *Fashion ID GmbH & Co. KG v Verbraucherzentrale NRW eV*. Judgment of 29 July 2019.

[11] Court of Justice of the European Union. 2018. Case C-210/16 *Wirtschaftsakademie Schleswig-Holstein GmbH*. Judgment of 5 June 2018.

[12] European Data Protection Board. 2020. Guidelines 4/2019 on Data Protection by Design and by Default (Version 2.0). EDPB, Brussels.

[13] UK Information Commissioner's Office. 2022. Purpose limitation, data minimisation, and storage limitation (ICO guidance). https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/data-protection-principles/ Accessed April 2022.

[14] Anu Bradford. 2020. *The Brussels Effect: How the European Union Rules the World*. Oxford University Press.

[15] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing reasoning and acting in language models. arXiv:2210.03629.

[16] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. arXiv:2302.04761.

[17] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed H. Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2023. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. arXiv:2308.08155.

[18] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. arXiv:2305.16291.

[19] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *USENIX Security*.

[20] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*.

[21] Hengrui Xu, Mengwei Xu, et al. 2023. Machine unlearning: A survey. arXiv:2306.03558.

[22] OWASP GenAI Security Project. 2024. OWASP Top 10 for LLM Applications, Version 1.1 (April 11, 2024). https://genai.owasp.org/resource/llm-top-10-for-llms-v1-1/

[23] OWASP GenAI Security Project. 2024–2025. OWASP Top 10 for LLMs and Generative AI (2025 refresh, project materials). https://genai.owasp.org/llm-top-10/

[24] Apostol Vassilev et al. 2025. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations (NIST AI 100 2e2025), Final. NIST Computer Security Resource Center, March 24, 2025. https://csrc.nist.gov/pubs/ai/100/2/e2025/final

[25] European Union Agency for Cybersecurity (ENISA). 2024. ENISA Threat Landscape 2024. September 2024. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024

[26] California Civil Code, Title 1.81.5 (California Consumer Privacy Act of 2018, as amended by the CPRA), Cal. Civ. Code §§1798.100–1798.199. https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&lawCode=CIV&

part=4.&title=1.81.5

[27] Autoridade Nacional de Proteção de Dados (ANPD). 2018/2019. Lei No. 13.709/2018 — Lei Geral de Proteção de Dados (LGPD) (English reference text). https://www.gov.br/anpd/pt-br/centrais-de-conteudo/outros-documentos-e-publicacoes-institucionais/lgpd-en-lei-no-13-709-capa.pdf

[28] California Privacy Protection Agency (CPPA). 2024–2025. Proposed Regulations: Automated Decisionmaking Technology (ADMT) (rulemaking materials). https://cppa.ca.gov/regulations/ccpa_updates.html

[29] Personal Data Protection Commission (PDPC) Singapore. Data Protection Obligations under the PDPA (official guidance, updated 2023). https://www.pdpc.gov.sg/overview-of-pdpa/the-legislation/personal-data-protection-act/data-protection-obligations

[30] PDPC Singapore. 2020. Model AI Governance Framework (Second Edition). https://www.pdpc.gov.sg/help-and-resources/2020/01/second-edition-of-model-artificial-intelligence-governance-framework

[31] Sarah Spiekermann and Lorrie Faith Cranor. 2009. Engineering privacy. *IEEE Transactions on Software Engineering* 35, 1 (2009), 67–82. doi:10.1109/TSE.2008.88

[32] Seda Gürses, Carmela Troncoso, and Claudia Díaz. 2011. Engineering privacy by design. In *Proceedings of the Conference on Computers, Privacy & Data Protection (CPDP)*, Brussels, Belgium. https://software.imdea.org/~carmela.troncoso/papers/Gurses-CPDP11.pdf

[33] Marit Hansen, Andreas Pfitzmann, and Sandra Steinbrecher. 2008. Identity management throughout one's whole life. *Information Security Technical Report* 13, 2 (2008), 83–94. doi:10.1016/j.istr.2008.06.003

[34] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James Hendler, and Gerald J. Sussman. 2008. Information accountability. *Communications of the ACM* 51, 6 (2008), 82–87. doi:10.1145/1349026.1349043

[35] Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, and D. Sculley. 2017. The ML Test Score: A rubric for ML production readiness and technical debt reduction. In *2017 IEEE International Conference on Big Data (Big Data)*, 1123–1132. doi:10.1109/BigData.2017.8258038

[36] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19)*, 220–229. doi:10.1145/3287560.3287596

[37] Ben Hutchinson and Margaret Mitchell. 2019. 50 years of test (un)fairness: Lessons for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19)*, 49–58. doi:10.1145/3287560.3287600

[38] Matthew Arnold, Rachel K. E. Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilović, Ravi Nair, Karthikeyan Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, Darrell Reimer, John Richards, Jason Tsay, Kush R. Varshney, and Yunfeng Zhang. 2019. FactSheets: Increasing trust in AI services through supplier's declarations of conformity. *IBM Journal of Research and Development* 63, 4/5 (2019), 6:1–6:13. doi:10.1147/JRD.2019.2942288

[39] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint* arXiv:1606.06565. https://arxiv.org/abs/1606.06565