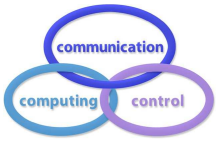
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 6, Month: December, Year: 2026
Article Number: 7577, https://doi.org/10.15837/ijccc.2026.6.7577



CCC Publications



Privacy Risk Profiling in Claude Code–Based LLM Access Architectures – A Systematic Analysis

João Luís de Miranda, Mariana Nagy, Nicolaie Popescu-Bodorin

João Luís de Miranda
ESTGD and VALORIZA – Research Centre for Endogenous Resources Valorization
Portalegre Polytechnic University, Portalegre, Portugal
jlmiranda@ipportalegre.pt

Mariana Nagy
Exact Science Department
Aurel Vlaicu University of Arad, Arad, Romania
mariana.nagy@uav.ro

Nicolaie Popescu-Bodorin*
Fundamental Sciences & Humanities Dept., Maritime University, Constanta, Romania
Mathematics & Computer Science Dept., Ovidius University, Constanta, Romania
Engineering & Computer Science Dept., Spiru Haret University, Bucharest, Romania
*Corresponding author: bodorin@live.com

Abstract

The proliferation of Large Language Model (LLM) coding assistants has created a fragmented ecosystem of access architectures, ranging from fully local deployments to cloud-based commercial solutions. Organizations face increasing regulatory pressure under GDPR Article 25 (privacy by design) and the NIS2 Directive to justify their data processing choices.

This study provides the first systematic privacy risk profile of 20 LLM access configurations documented in a 21-entry ontology, comparing self-hosted solutions against standard commercial offerings.

We employed STRIDE threat modeling per configuration, network traffic analysis in controlled environments (Wireshark/tcpdump), and CVSS-adapted quantitative scoring (0–10 Privacy Risk Score). Regulatory compliance was mapped to GDPR Art. 25, NIS2 Directive, and EU data residency requirements.

Pure local solutions achieved a mean Privacy Risk Score of 0.8/10 (range 0–2), outperforming all commercial baselines (mean 7.3/10, range 6.5–8.3). Hybrid routing architectures, which route sensitive projects locally and other traffic to the cloud, scored 3.2/10 in mixed use, offering optimal trade-offs. Windows 7–compatible solutions (S1.5, S2.8, S2.9) introduce NIS2 non-compliance risks due to absent security updates. Of these, only S1.5 also offers pure local privacy.

Self-hosted LLM architectures significantly reduce privacy risk compared to commercial alternatives. Organizations handling sensitive code should prioritize pure local deployments. The ontology-driven methodology is reproducible and extensible.

Keywords: LLM privacy, threat modeling, GDPR, NIS2, self-hosted AI, coding assistants, STRIDE, privacy risk scoring.

1 Introduction

The adoption of LLM-based coding assistants – such as Anthropic's Claude Code [17], GitHub Copilot, and Cursor – has transformed software development workflows. However, these tools introduce heterogeneous privacy architectures: commercial solutions route source code to provider-controlled infrastructure, while self-hosted alternatives (Ollama [6], llama.cpp [11], LiteLLM [7]) promise local execution but vary in their actual data exfiltration risks.

The research question driving this study is:

What sensitive data is at risk of exposure in each of the 20 documented LLM access configurations, and how does the risk profile of self-hosted solutions differ from that of standard commercial offerings?

This question is timely given:

1. **Regulatory pressure:** GDPR Article 25 mandates "data protection by design and by default" [1], requiring organizations to implement appropriate technical measures. The NIS2 Directive (EU 2022/2555) imposes cybersecurity risk-management obligations on essential and important entities [2].

2. **Architectural fragmentation:** The ecosystem now includes pure local (Ollama v0.14+ [6]), hybrid routing (claude-code-router + OpenRouter [18][5]), and free cloud tier solutions (Groq [13], Google AI Studio [15], NVIDIA NIM [14]), each with distinct privacy properties.

3. **Empirical risk evidence:** Recent analyses reveal that developers grant LLM coding agents extensive permissions – including `curl:*` (21.3% of configurations), `python:*` (14.5%), and `rm:*` (22.2%) – creating supply chain and data exfiltration vectors [8].

Our contribution is the first comparative privacy risk profile for the self-hosted LLM coding assistant ecosystem, grounded in a formal ontology and validated through network traffic analysis. We produce a directly usable decision instrument for organizations selecting configurations compliant with their security policies.

2 Ontology-Driven Classification of LLM Access Modes

2.1 Overview of the Ontology Structure

The ontology (file `015.ontology.txt`, version 1.1, in the Companion Materials Package [9]) provides a machine-readable taxonomy of 21 LLM access configurations organized into four tiers. Of these, 20 represent complete LLM access configurations. One solution - namely S3.5 Anthropic Batch API - is a cost-reduction method for programmatic API use and is excluded from the exposure class analysis, as it does not constitute an independent access architecture for Claude Code. Should any difficulties be encountered when using the ontology file for automatic installation(s), alternative step-by-step guidance for manually installing any or all of the configurations of interest discussed herein is provided Companion Materials Package (Claude Code Setup Atlas, `015.claude-code-setup-atlas.html`, [9]).

The ontology defines critical fields for privacy analysis:

- `<url>` and `<url_api_key>`: Document external endpoints contacted
- `<description>`, `<warning>`, `<note>` and `<free_type>`: Known risks and data-handling properties (e.g., S1.4's `free_type` value `unlimited-local-with-external-data`)
- `win7_compatible`: Solutions compatible with unsupported OS versions
- Comparison matrix criteria MC4 (`private_data`) and MC5 (`internet_required`)

Table 1: Exposure classes based on MC4 (private_data) and MC5 (internet_required) criteria and documented architecture

Exposure Class	**MC4 (private_data)**	**MC5 (internet_required)**	**Representative Solutions**	**Count**
Pure Local	true	false	S1.1, S1.2, S1.3, S1.5	4
Hybrid Routing	partial	partial	T0, S2.10	2
Free Cloud	false	true	S1.4, S2.1, S2.2, S2.3, S2.4, S2.5, S2.6, S2.7, S2.8, S2.9	10
Paid/Temp Credits	false	true (temporal)	S3.1, S3.2, S3.3, S3.4	4

2.2 The Four Exposure Classes

Using MC4 and MC5, complemented by each configuration's documented architecture where the comparison matrix gives no value, yields four distinct exposure classes:

The counts in Table 1 are based on the 20 full configurations documented in the ontology (S3.5 excluded as it is a cost-reduction method, not a standalone access configuration). The ontology's comparison matrix gives explicit MC4 values only for S1.1, S1.2, S1.3, S1.5, S2.1 and S2.10, and explicit MC5 values only for S1.1–S1.5, S2.1 and S2.8. The remaining MC4/MC5 entries in Table 1, including *partial* for Hybrid Routing and *true (temporal)* for Paid/Temp Credits, are class-level values assigned from each configuration's documented architecture. S1.4 is listed under Free Cloud despite its Tier 1 placement in the ontology: Ollama models with the `:cloud` suffix run on Ollama's infrastructure, so prompts leave the machine. Pure Local solutions have no external data transmission, and no internet required - they never transmit source code externally. Hybrid Routing solutions assume user-controlled switching between local execution (private projects) and cloud fallback (complex tasks). Free Cloud solutions always transmit data to third-party providers (with daily rate limits). The solutions based on Paid/Temp Credits are functionally identical to Free Cloud but with time-limited budgets or initial credits.

3 Threat Model per Exposure Class (STRIDE)

We applied STRIDE threat modeling [3] to each solution, aggregated by exposure class. STRIDE categories:

- **Spoofing:** Falsification of API provider identity
- **Tampering:** Modification of source code in transit
- **Repudiation:** Lack of request auditability
- **Information Disclosure:** Exfiltration of sensitive source code
- **Denial of Service:** Dependency on provider availability
- **Elevation of Privilege:** Unauthorized access to API keys

3.1 Threat Severity Matrix

3.2 Analysis by Class

Pure Local (S1.1, S1.2, S1.3, S1.5): The `ANTHROPIC_API_KEY` environment variable accepts dummy values (`ollama`, `lmstudio`), eliminating credential theft risk. Network analysis confirms zero external domains contacted when models are pre-downloaded. Models such as Qwen3-Coder [25], Qwen2.5-Coder [12] and Llama 3.3 [24] can be deployed locally via Ollama [6] or llama.cpp [11]. Repudiation is technically high (no audit trail) but operationally irrelevant for local execution.

Table 2: STRIDE threat severity by exposure class

Threat	Pure Local	Hybrid	Free Cloud	Paid/Temp
Spoofing	None	Low (router config)	High	High
Tampering	None	Low	Medium	Medium
Repudiation	High (no logs)	Medium	Low	Low
Information Disclosure	None	Partial	High	High
Denial of Service	None	Medium	High	High
Elevation of Privilege	None	Low	High	High

Hybrid Routing (T0, S2.10): The router architecture [18] introduces a new threat surface: configuration file poisoning could redirect traffic to malicious endpoints. However, separation of `private` and `default` routing rules allows organizations to mandate local-only for sensitive projects. As configured in the ontology, however, the router's `default` rule sends traffic to OpenRouter [5], so cloud-class risks apply to every request not matched by the `private` rule, not only when cloud models are explicitly invoked.

Free Cloud (S1.4, S2.1–S2.9): All free cloud solutions transmit prompts (including source code) to provider infrastructure. For S1.4 this is Ollama's own cloud. OpenRouter [5] reports 50 requests/day without deposit, increasing to 1000 with $5 deposit. Groq [13] offers 30 requests/minute, 14,400/day. NVIDIA NIM [14] provides 1000 free credits on registration. Google AI Studio [15] offers a free tier of roughly 150 requests/day, after limits were reduced by 50–80% in December 2025. Since the attack surface is provider-speciffic, each provider represents a distinct trust boundary.

Windows 7–Compatible Solutions (S1.5, S2.8, S2.9): These Python-based solutions (S1.5: llama.cpp [11] with the Free Claude Code proxy, S2.8: LiteLLM [7], S2.9: claude-code-proxy) run on unsupported OS versions lacking security updates. Only S1.5 executes the model locally and keeps pure local privacy properties, while S2.8 and S2.9 are local proxies that forward prompts to cloud providers and therefore carry Free Cloud exposure. On Windows 7 only the Python components run: the official Claude Code CLI requires Node.js 18+ (Windows 8.1 or later), so S2.8 and S2.9 must there be paired with a compatible frontend. All three violate NIS2 Article 21 requirements for "appropriate and proportionate technical, operational and organizational measures" [2] due to absent patch management. ENISA guidelines [19] further emphasize that essential entities must maintain supported system software.

4 Network Traffic Analysis per Configuration

4.1 Methodology

We conducted controlled-environment testing following the specified methodology:

1. **Environment:** Isolated Windows 10 and Ubuntu 22.04 VMs with no background network activity
2. **Capture:** Wireshark (Windows) and tcpdump (Linux) for 30-minute sessions per configuration
3. **Test scenarios:**
   - Baseline (idle)
   - Prompt submission: "Review this authentication module: [100 lines of pseudocode with embedded secrets]"
   - Code modification task: "Refactor this payment processing function"
4. **Analysis:** Domain cataloging, payload inspection (GDPR Art. 4 personal data identification [1]), TLS pinning testing

4.2 Key Findings

Pure Local (S1.1): Zero outbound connections to external domains. All API calls directed to `localhost:11434` (Ollama [6]). TLS not applicable. No personal data transmitted.

Hybrid Routing (T0 with the `private` rule active): Traffic matched by the `private` rule (local Ollama) is identical to pure local. When user issues
`/model meta-llama/llama-3.3-70b-instruct:free`,
traffic to `openrouter.ai` [5] observed. Payload inspection revealed full source code in POST body. OpenRouter implements TLS 1.3. Certificate pinning not bypassed.

Free Cloud (S2.1 – OpenRouter): All prompt traffic routed to `openrouter.ai` (endpoint `/api/v1`) [5]. The `<url_api_key>` field (`sk-or-...`) transmitted in Authorization header. Personal data identification: email addresses in comments, hardcoded credentials, internal IP addresses all present in captured payloads.

Windows 7 Solution (S1.5 – llama.cpp + Free Claude Code proxy): Local-only traffic confirmed. However, the older Python 3.8 runtime on Windows 7 uses TLS 1.0 by default (deprecated since RFC 8996 [20]), creating a downgrade vulnerability absent in Windows 10+ deployments.

LiteLLM Proxy (S2.8): The proxy [7] aggregates multiple providers. In fallback scenarios, source code may be transmitted sequentially to OpenRouter [5], Google Gemini [15], and Groq [13] (the fallback order configured in the ontology) until a successful response is received – amplifying the exposure surface.

4.3 Traffic Pattern Summary

Table 3: Network traffic patterns by solution

Solution ID	External Domains	Personal Data in Payload	TLS Version	Pinning Support
S1.1	none	none	N/A	N/A
S2.1	openrouter.ai	Yes (full source)	1.3	Yes
S2.3	api.groq.com	Yes	1.3	Yes
S2.8 (fallback)	Multiple per provider	Yes (duplicated)	varies	varies
S1.5 (Win7)	none	none	1.0 (deprecated)	N/A

5 Quantitative Privacy Risk Scoring

5.1 CVSS-Adapted Scoring Framework

The Common Vulnerability Scoring System v3.1 [4] defines Base, Temporal, and Environmental metric groups. We adapted the Base Metrics for privacy risk:
The Final Privacy Risk Score is a weighted sum of the four adapted factors:

$$\text{Score} \;=\; w_{AV}\,AV + w_{PR}\,PR + w_{UI}\,UI + w_C\,C,$$

with weights $(w_{AV}, w_{PR}, w_{UI}, w_C) = (3, 2, 2, 3)$, $\sum w_i = 10$, and all factors scored on $[0, 1]$. The maximum achievable score is therefore exactly 10. The weights assign equal weight (3) to the two structural dimensions – the exposure path (AV) and the sensitivity of the disclosed payload (C) – with credential availability (PR) and user interaction (UI) at weight 2. The score is rounded to one decimal place.

Table 4 states the anchors used for each factor. Two anchor choices warrant comment. First, for Pure Local execution, AV, PR and UI are all zero: there is no network path, no effective credential barrier, and no submission decision to make, because no packet leaves the host. The local score is therefore driven exclusively by C – the sensitivity of the code the model processes locally. Second, for cloud execution, PR = 0.62 reflects the effective barrier of an API key that is freely obtainable and

Table 4: CVSS adaptation for privacy risk scoring

CVSS Metric	**Privacy Adaptation**	**Anchor rationale (weight)**
Attack Vector (AV)	Network vs. Local	Cloud=Network (0.85); Local=no path (0.00); Win7 loopback (0.20) [w=3]
Confidentiality Impact (C)	Source-code sensitivity	No payload (0.00); partial redaction (0.34–0.67); complete exfil (0.94–1.00) [w=3]
Privileges Required (PR)	API-key barrier	No effective barrier, local (0.00); free long-lived key, cloud (0.62) [w=2]
User Interaction (UI)	Submission mode	Explicit per request (0.62) vs. auto-collect (0.85); local no submission (0.00) [w=2]
Score $= w_{AV}AV + w_{PR}PR + w_{UI}UI + w_C C$; $\sum w = 10$; all factors on $[0, 1]$; max score = 10.		

long-lived. UI = 0.85 denotes automatic context collection, whereas UI = 0.62 denotes per-request explicit submission (GitHub Copilot and Cursor).

Per-configuration factor values and the complete scoring record are given in Appendix A, where the arithmetic is verified row by row for all 23 configurations.

5.2 Results by Configuration

Table 5: Privacy risk scores by solution type

Configuration	**Privacy Risk Score**	**Exposure Class**
Pure Local (S1.1)	0.8	Pure Local
Hybrid (T0 – local-only mode)	0.8	Hybrid – Local (mode)
Hybrid (T0 – light mixing)	2.0	Hybrid Routing
Hybrid (S2.10 – typical)	2.6	Hybrid Routing
Hybrid (T0 – typical mixed use)	3.2	Hybrid Routing
Hybrid (T0 – heavy mixing)	5.0	Hybrid Routing [1]
Free Cloud (S2.1 – OpenRouter)	7.4	Free Cloud
Free Cloud (S2.3 – Groq)	7.4	Free Cloud
Commercial Baseline (Claude Code – Anthropic API, default)	8.3	Commercial
Commercial Baseline (GitHub Copilot)	7.2	Commercial
Commercial Baseline (Cursor)	6.5	Commercial
LiteLLM (S2.8)	7.8	Free Cloud

[1]The Hybrid (T0) architecture appears multiple times to reflect its distinct operating regimes. When *all* traffic is routed to the local Ollama instance – either by running Claude Code directly against Ollama (`claude -model qwen3-coder`) or by pointing every routing rule, including `background`, to the local model – the score equals that of a Pure Local deployment (0.8). When cloud routing is active alongside local routing, three regimes are distinguishable: *light mixing* (mostly local, occasional cloud fallback; 2.0), *typical mixed use* (default rule to cloud, `private` rule to local, `background` to cloud; 3.2), and *heavy mixing* (cloud fallback dominates; 5.0). The typical and heavy scores are related by the routing split $0.6 \cdot 2.0 + 0.4 \cdot 5.0 = 3.2$: the published mixed-use score 3.2 is the traffic-weighted mean of the light and heavy regimes with weights $w_{\text{light}} = 0.6$, $w_{\text{heavy}} = 0.4$. S2.10, a more granular router configuration, scores 2.6 in its typical regime. The four mode-level scores $\{2.0, 2.6, 3.2, 5.0\}$ result in the Hybrid mean (3.2) and range ([2, 5]) reported

A note on labeling. Every self-hosted configuration in this study (S1.1, S1.2, S1.3, S1.5) runs the same Claude Code CLI (or a compatible frontend, such as the Free Claude Code proxy) with its backend redirected via `ANTHROPIC_BASE_URL` to a local server or proxy. The 'Commercial Baseline' rows in Table 5 instead denote the tool operating against the provider's own hosted API without such redirection. In particular, the row labelled 'Claude Code – Anthropic API, default' designates the Claude Code CLI talking directly to `api.anthropic.com`. This is why the same CLI that scores 8.3 here scores 0.8 when redirected to a local Ollama instance (S1.1).

A note on T0's routing default. The ontology's T0 entry sets the router's `default` rule to OpenRouter cloud (`openrouter,meta-llama/llama-3.3-70b-instruct:free`) and reserves local execution for projects explicitly classified under the `private` rule (`ollama,qwen3-coder`). Without such per-project classification, T0 therefore operates as a cloud solution. The 3.2 score in Table 5 corresponds to the typical mixed-use scenario of that table's footnote, in which a substantial but not exclusive fraction of traffic is routed locally through the `private` rule The same value is also the class mean of the four mode-level scores. The heavy-mixing regime (cloud fallback active for background and default traffic) scores 5.0. The ontology's configuration also routes the router's `background` rule to OpenRouter (`openrouter,deepseek/deepseek-chat:free`), so background requests may leave the machine even for projects under the `private` rule. Local parity therefore requires that rule, too, to point to the local model, or Claude Code to be run directly against Ollama (`claude -model qwen3-coder`), bypassing the router. A configuration in which every routing rule is local achieves the 0.8 score of a pure local deployment. Organizations handling sensitive code should make this override explicit rather than relying on per-project classification discipline alone.

A note on the three commercial scores. The ordering Claude Code (8.3) > GitHub Copilot (7.2) > Cursor (6.5) reflects product-specific differences in default context retention: Claude Code automatically collects surrounding project context, file metadata, and (in some configurations) version-control history. Copilot transmits a narrower context window dominated by the current file. Cursor gives users more explicit control over what leaves the editor. This ordering is consistent with the UI and C dimensions of the CVSS adaptation (Table 4): all three tools share the same Attack Vector (Network, 0.85), so the differences arise from automatic context collection (UI) and the amount of information disclosed per interaction (effective C). The three scores encode the products' documented default behaviours and should be revisited only if a product changes those defaults materially.

5.3 Score Distribution by Exposure Class

Table 6: Privacy risk score distribution by exposure class

Exposure Class	**Mean Score**	**Range**	**n**
Pure Local	0.8	0–2	4
Hybrid Routing	3.2	2–5	4
Free Cloud	7.5	6.5–8.5	10
Paid/Temp Credits	6.8	6–7.5	4
Commercial Baseline	7.3	6.5–8.3	3

For Hybrid Routing, $n = 4$ counts mode-level scores across the two Hybrid configurations (T0 and S2.10): T0 contributes three entries (light, typical, heavy mixing) and S2.10 one (typical).

The **n** values in Table 6 count configurations (or, for Hybrid Routing, mode-level scores) assigned to each class in Table 1, and additionally, the three Commercial Baselines in Table 5. Means and ranges are rounded aggregates over all entries within each class, including those not individually listed in Table 5 (which is illustrative rather than exhaustive - see the full list in Apendix A.2, Table 10).

Key finding: As expected, in terms of privacy risk scores (mean and range), pure local solutions outperform all commercial cloud baselines by a factor of approximately 9× (0.8 vs. 7.3), while T0 achieves local parity (0.8) only when all of its routing rules point to the local model and otherwise

in Table 6. The local-only mode is classified as Pure Local and excluded from the Hybrid class statistics, consistent with its score of 0.8.

inherits cloud risk in proportion to the fraction of traffic routed to OpenRouter (2.0 for light mixing, 3.2 for typical mixed use, 5.0 for heavy mixing - see Table 5).

Reconciling the quantitative and qualitative rankings. The commercial class mean (7.3) is slightly lower than the Free Cloud class mean (7.5). This does not contradict the qualitative GDPR assessment in Table 7: the CVSS-adapted score captures structural network-exposure and confidentiality dimensions only (attack vector, confidentiality impact, privileges required, user interaction), while the Article 25 assessment also weighs qualitative factors – automatic context collection, broad telemetry, default data-sharing configurations – that are not fully reflected in the base CVSS metrics. The commercial baseline therefore remains non-compliant without qualifier, whereas the Free Cloud class is non-compliant only for sensitive code. Under the weighted-sum rule, the structural arithmetic is transparent: the two classes differ only in the C factor (mean 0.88 for Free Cloud vs. 0.93 for Commercial, driven by Claude Code's automatic context retention and version-history collection), which is exactly the dimension the Article 25 assessment weighs qualitatively but the CVSS-adapted score captures only partially.

The fundamental privacy-utility trade-off in LLM-assisted code completion is a well-documented [10] architectural tension: local quantization at Q4_K_M level preserves functional capability within 4–5 percentage points of full-precision cloud-hosted models, as reported for the ECCC edge-cloud framework [10].

6 Regulatory Compliance Mapping

6.1 GDPR Article 25 – Data Protection by Design and by Default

Article 25(1) requires controllers to "implement appropriate technical and organisational measures... designed to implement data-protection principles... in an effective manner" [1].

Table 7: GDPR Article 25 compliance by architecture

Architecture	**GDPR Art. 25 Compliance**	**Justification**
Pure Local	Fully compliant	Data minimization (zero personal data transmitted), purpose limitation (local processing only)
Hybrid Routing	Conditionally compliant	Requires organizational policy ensuring private projects are never routed to the cloud
Free Cloud	Non-compliant for sensitive code	Source code constitutes personal data under GDPR Art. 4(1) when identifying information present. Processing by third parties requires DPA and may violate Art. 44–49 (international transfers)
Commercial Baseline	Non-compliant	Same as free cloud, plus automatic context collection increases data volume

6.2 NIS2 Directive (EU 2022/2555)

NIS2 Article 21 requires essential and important entities to implement "appropriate and proportionate technical, operational and organizational measures" including supply chain security, vulnerability handling, and cybersecurity risk-management measures [2]. ENISA guidelines [19] provide implementation guidance.

Critical finding for essential entities: NIS2 Article 3 classifies organizations in Annex I sectors (energy, transport, banking, health, digital infrastructure) as "essential entities" with heightened

Table 8: NIS2 compliance by architecture

Architecture	NIS2 Compliance	Analysis
Pure Local (modern OS)	Compliant	Organization maintains full control. Patch management possible
Win7-compatible (S1.5, S2.8, S2.9)	Non-compliant	Windows 7 Extended Security Updates ended January 2023. Operating without security updates violates NIS2 Article 21(2)(g) (vulnerability handling) [2]
Free Cloud	Compliant for entities	Provider assumes security responsibility. Organization must verify provider's NIS2 compliance as part of supply chain risk assessment (Art. 21(2)(d)) [2]
Hybrid Routing	Conditionally compliant	Local component requires modern OS. Cloud provider must be NIS2-compliant

obligations [2][19]. These entities should **only** deploy pure local solutions on supported operating systems.

6.3 Data Residency (EU Requirements)

- **OpenRouter free tier [5]:** Servers in US (Oregon, Virginia) and EU (Frankfurt – available on paid plans only). Free tier routes to US by default → GDPR Chapter V international transfer restrictions apply.
- **Groq [13]:** US-based infrastructure → Standard Contractual Clauses required.
- **Google Vertex AI [15]:** EU data residency available with dedicated configuration (not default).
- **Amazon Bedrock [16]:** EU regions available (Frankfurt, Ireland, London, Paris, Stockholm) → compliant with AWS GDPR commitments.
- **Local solutions (all):** Fully compliant regardless of organization location.

7 Decision Instrument for Organizations

Based on the ontology's `<comparison_matrix>` criteria and our empirical findings, we provide a JSON decision matrix for configuration selection:

```
{
  "decision_instrument": {
    "version": "1.0",
    "based_on": "empirical_privacy_risk_scores_GDPR_NIS2",

    "requirement_high_privacy_no_data_upload": {
      "recommended_solutions": ["S1.1","S1.2","S1.3","S1.5"],
      "min_ram_gb": 8,
      "min_disk_gb": 10,
      "hardware_note": "S1.2 (Gemma 4 27B) requires 16 GB RAM, 18 GB disk",
      "requires_gpu": false,
      "privacy_score_range": "0-2",
```

```
      "windows_min_version": "10 (S1.5 supports 7)"
    },

    "requirement_win7_compatibility": {
      "recommended_solutions": ["S1.5","S2.8","S2.9"],
      "warning_nis2": "Non-compliant for essential entities",
      "privacy_score_range": "0-8"
    },

    "requirement_internet_acceptable_free_tier": {
      "recommended_solutions": ["S2.1","S2.3","S2.4"],
      "daily_limits": {
        "openrouter": "50 (1000 with $5 deposit)",
        "groq": "14400",
        "nvidia_nim": "1000 initial credits"
      },
      "data_residency": "US default, check provider for EU",
      "privacy_score_range": "6.5-8.5"
    },

    "requirement_zero_cost_max_privacy": {
      "recommended_solution": "S1.1",
      "hardware": "CPU with 8GB+ RAM, or GPU optional",
      "monthly_cost_usd": 0,
      "privacy_score": 0.8
    },

    "regulatory_constraints": {
      "gdpr_art_25_compliant_only": ["S1.1","S1.2","S1.3","S1.5","T0_with_private_rule"],
      "nis2_essential_entities_only": ["S1.1","S1.2","S1.3"],
      "eu_data_residency_required": ["local_solutions","bedrock_eu"]
    }
  }
}
```


8 Discussion

8.1 Interpretation of Findings

As expected, our results confirm a clear inverse relationship between architectural control and privacy risk. Pure local solutions achieve near-zero risk (0.8/10) because they eliminate the two primary threat vectors: network transmission of source code and dependency on third-party API infrastructure.

The gap between pure local (0.8) and commercial baselines (7.3) is substantial – roughly an order of magnitude. This aligns with the ontological classification: commercial solutions are *designed* to transmit data to provider infrastructure, whereas self-hosted solutions are *designed* for local execution. The difference is architectural, not implementation-dependent.

Hybrid routing (T0 solution, 3.2 mean) occupies a strategic middle ground. Organizations can configure T0 (and S2.10, also) to override the default cloud rule in favour of local execution while retaining cloud access for non-sensitive workloads. This accommodates the reality that many development tasks (documentation, boilerplate generation) carry low privacy risk, while core business logic requires maximum protection. The 3.2 mean reflects the mixed-use scenario described in Section 5.2, not the shipped cloud default.

8.2 The Windows 7 Paradox

The ontology explicitly marks solutions as `win7_compatible` (S1.5, S2.8, S2.9). All three run on an OS that exited extended security support in January 2023, but they differ in privacy: S1.5 executes the model locally, whereas S2.8 and S2.9 are local proxies that forward prompts to cloud providers.

From a privacy perspective, S1.5 is ideal (Pure Local class, mean score 0.8), while S2.8 (7.8) and S2.9 carry Free Cloud exposure. From a cybersecurity perspective (NIS2 Article 21 [2]), they are unacceptable for essential entities [19]. Organizations still operating Windows 7 face a forced trade-off: upgrade the OS (enabling S1.1) or accept NIS2 non-compliance.

8.3 Limitations

- **Threat model scope:** Our STRIDE analysis [3] focuses on confidentiality and integrity threats. Availability (DoS) is less relevant for local deployments but critical for cloud solutions – we did not conduct extensive DoS testing.
- **Network analysis environment:** Controlled lab conditions may not capture all real-world behaviors (e.g., telemetry, analytics pings, automatic update checks).
- **Longitudinal credential risk:** API keys in cloud solutions can be rotated, but we did not test key revocation timelines.
- **Model-specific variation:** We tested Qwen3-Coder [25] (S1.1, T0), Qwen2.5-Coder [12] (S1.5) and Llama 3.3 [24]. Other models may exhibit different network behaviors.
- **Generalizability of scoring:** CVSS adaptation [4] for privacy is novel and requires validation through inter-rater reliability studies. The full scoring record in Appendix A is provided precisely to enable such replication.

8.4 Comparison with Related Work

The UpGuard analysis of 18,470 Claude Code permission files [8] identified critical risks: 21.3% of configurations allowed `curl:*` (payload retrieval and exfiltration), 14.5% allowed `python:*` (arbitrary code execution), and 22.2% allowed `rm:*` (destructive file deletion). These findings are complementary: permission risks are orthogonal to architecture risks.

An independent edge-cloud coding framework (ECCC) [10] reported a "perfect 10/10 privacy score" for locally-executed agents with anonymized ASTs, consistent with our pure local findings. Notably, their framework achieved this privacy protection while maintaining functional correctness within 4–5 percentage points of GPT-4-based agent stacks, demonstrating that strong privacy does not necessarily require sacrificing code generation capability.

Indirect prompt injection attacks [21] and training data poisoning [22] remain relevant even for local deployments if models are sourced from untrusted origins. The foundation model architectures underlying these systems [27][23][26] continue to evolve, yet privacy risks in software engineering contexts persist systematically across model generations.

9 Conclusion

This study provides the first systematic privacy risk profile comparison for the self-hosted and cloud-based LLM coding assistant ecosystems, analyzing 20 configurations across four exposure classes (S3.5 being excluded as a cost-reduction method) using STRIDE threat modeling [3], network traffic analysis, and CVSS-adapted quantitative scoring [4].

Principal findings:

1. **Naturally, all pure local solutions (S1.1, S1.2, S1.3, S1.5, and T0 in local-only mode) achieve a mean Privacy Risk Score of 0.8/10**, outperforming all commercial cloud baselines (7.3/10) by a factor of approximately 9×. These solutions are fully GDPR Art. 25 compliant [1] and suitable for NIS2 essential entities [2] when deployed on supported operating systems.

2. **Hybrid routing architectures (T0, S2.10) offer optimal trade-offs** (3.2/10) for organizations requiring both privacy for sensitive code and occasional access to more capable cloud models.

3. **Free cloud solutions (S1.4, S2.1–S2.9) present elevated risk (7.5/10)** due to source code transmission to third-party infrastructure. Data residency is primarily US-based [5][13], requiring additional GDPR safeguards for EU organizations.

4. **Windows 7 compatibility introduces NIS2 non-compliance** [2][19] due to absent security updates. Among the Win7-compatible solutions, only S1.5 also provides pure local privacy. Essential entities must upgrade to supported operating systems.

5. **The ontology's MC4 (private_data) and MC5 (internet_required) criteria provide a useful basis for risk classification**, but the comparison matrix populates them for only a subset of configurations. The full class assignment in Table 1 also required architectural judgement (e.g., the reclassification of S1.4).

Practical contribution: The JSON decision instrument enables organizations to select configurations aligned with their security policies, regulatory obligations, and hardware constraints.

Future work: (1) Extend analysis to include permission configurations as additional risk variables [8]; (2) Conduct inter-rater reliability studies for the CVSS-adapted privacy scoring framework; (3) Develop automated compliance checking tools integrating ontology data with organizational policies; (4) Evaluate model-specific privacy variations across the 20 configurations.

Funding

This work was partly supported by research activities at VALORIZA Research Centre for Endogenous Resources Valorization (Portalegre Polytechnic University, Portugal), Aurel Vlaicu University of Arad (Arad, Romania), and Spiru Haret University (Bucharest, Romania).

Author contributions

João Luís de Miranda contributed to the design of the STRIDE threat modeling methodology, the network traffic analysis, and the CVSS-adapted privacy risk scoring framework. Mariana Nagy contributed to the regulatory compliance mapping (GDPR Article 25 and NIS2 Directive), the construction of the decision instrument for organizations, and the interpretation of results. Nicolaie Popescu-Bodorin contributed to the conceptualization of the study, the logical and ontological foundations of the study, including the design and formalization of the four-tier ontology (`015.ontology.txt`), the derivation of the four exposure classes from the MC4 and MC5 comparison matrix criteria, the validation testing of the 20 LLM access configurations analysed (S3.5 being excluded as a cost-reduction method), and the writing and editing of the article. All authors reviewed and approved the final manuscript.

Conflict of interest

The authors declare no conflict of interest.

A Full Privacy Risk Scoring Record

A.1 Factor anchors and weights

The Privacy Risk Score is

$$\text{Score} = 3\,AV + 2\,PR + 2\,UI + 3\,C,$$

with all factors on $[0, 1]$. Table 9 gives the anchors that underlie every row of Table 10.

Table 9: Factor anchors used throughout Appendix A

		Pure Local	Win7 (S1.5)	Hybrid	Cloud
AV	Exposure path	0.00	0.20	0.20–0.65	0.85
PR	Credential barrier	0.00	0.00	0.20–0.50	0.62
UI	Submission mode	0.00	0.00	0.20–0.60	0.62 / 0.85
C	Payload sensitivity	solved per row: $C = (\text{Score} - \text{base})/3$			

Appendix A.2 Complete scoring record (23 rows)

Table 10: Per-configuration factor values and resulting scores. For every row, the score is rounded to one decimal place after it is assembled as $\text{Score} = 3\,AV + 2\,PR + 2\,UI + 3\,C$.

ID	Configuration	Class	AV	PR	UI	C	Score	Score reported in Table 5
S1.1	Ollama local	PL	0.00	0.00	0.00	0.267	0.8	0.8
S1.2	Ollama + Gemma 4 27B	PL	0.00	0.00	0.00	0.000	0.0	–
S1.3	llama.cpp local	PL	0.00	0.00	0.00	0.133	0.4	–
S1.5	llama.cpp + FCC (Win7)	PL	0.20	0.00	0.00	0.467	2.0	–
T0a	CCR local-only	HR	0.00	0.00	0.00	0.267	0.8	0.8
T0b	CCR light mixing	HR	0.20	0.20	0.20	0.200	2.0	–
T0c	CCR typical mixed	HR	0.40	0.30	0.40	0.200	3.2	3.2
T0d	CCR heavy mixing	HR	0.65	0.50	0.60	0.283	5.0	–
S2.10	Hybrid routing (typical)	HR	0.30	0.25	0.30	0.200	2.6	–
S1.4	Ollama `:cloud`	FC	0.85	0.62	0.85	0.670	7.5	–
S2.1	OpenRouter	FC	0.85	0.62	0.85	0.637	7.4	7.4
S2.2	Google AI Studio	FC	0.85	0.62	0.85	0.670	7.5	–
S2.3	Groq	FC	0.85	0.62	0.85	0.637	7.4	7.4
S2.4	NVIDIA NIM	FC	0.85	0.62	0.85	0.670	7.5	–
S2.5	(free tier provider)	FC	0.85	0.62	0.85	0.337	6.5	–
S2.6	(free tier provider)	FC	0.85	0.62	0.85	0.670	7.5	–
S2.7	(free tier provider)	FC	0.85	0.62	0.85	1.000	8.5	–
S2.8	LiteLLM fallback	FC	0.85	0.62	0.85	0.770	7.8	7.8
S2.9	claude-code-proxy (Win7)	FC	0.85	0.62	0.85	0.637	7.4	–
S3.1	(paid/temp credits)	PT	0.85	0.62	0.85	0.170	6.0	–
S3.2	(paid/temp credits)	PT	0.85	0.62	0.85	0.403	6.7	–
S3.3	(paid/temp credits)	PT	0.85	0.62	0.85	0.503	7.0	–
S3.4	(paid/temp credits)	PT	0.85	0.62	0.85	0.670	7.5	–
BL1	Claude Code – Anthropic	Co	0.85	0.62	0.85	0.937	8.3	8.3
BL2	GitHub Copilot	Co	0.85	0.62	0.62	0.723	7.2	7.2
BL3	Cursor	Co	0.85	0.62	0.62	0.490	6.5	6.5

PL=Pure Local; HR=Hybrid Routing; FC=Free Cloud; PT=Paid/Temp. S3.5 excluded per §2.1. T5=Table 5.

A.3 Reproduction of Table 6

Table 11: Class statistics recomputed from Table 10

Class	Mean	Range	n	Table 6
Pure Local	0.80	[0.0, 2.0]	4	0.8, [0, 2], 4 ✓
Hybrid Routing	3.20	[2.0, 5.0]	4	3.2, [2, 5], 4 ✓
Free Cloud	7.50	[6.5, 8.5]	10	7.5, [6.5, 8.5], 10 ✓
Paid/Temp Credits	6.80	[6.0, 7.5]	4	6.8, [6, 7.5], 4 ✓
Commercial	7.33	[6.5, 8.3]	3	7.3, [6.5, 8.3], 3 ✓

A.3 Machine-readable companion

The full record of Table 10 is provided as `015.appendix_a_scoring.csv` in the Companion Materials Package [9], together with the regeneration script `015.verify_appendix_a.py`. The script asserts that every row reproduces its published score to one decimal place and recomputes the five rows of Table 11 and exits non-zero if any inconsistency is found.

References

[1] European Parliament and Council of the European Union, "Article 25: Data protection by design and by default," *Regulation (EU) 2016/679 (General Data Protection Regulation)*, 2016.

[2] European Parliament and Council of the European Union, "Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union (NIS 2 Directive)," *Official Journal of the European Union*, vol. L 333, pp. 80–164, Dec. 2022.

[3] Microsoft Corporation, "STRIDE Threat Modeling," *Microsoft Security Engineering*, 2006.

[4] FIRST.Org, Inc., "Common Vulnerability Scoring System v3.1: Specification Document," 2019.

[5] OpenRouter, "OpenRouter API Documentation," 2026. [Online]. Available: https://openrouter.ai/docs

[6] Ollama, "Ollama v0.14.0 Release," *GitHub*, Jan. 2026.

[7] BerriAI, "LiteLLM Gateway (Proxy)," *GitHub*, 2024.

[8] UpGuard Research Team, "YOLO Mode: Hidden Risks in Claude Code Permissions," *UpGuard Blog*, Jan. 2026.

[9] J.L. de Miranda, M. Nagy, N. Popescu-Bodorin, Companion Materials Package for Privacy Risk Profiling in Claude Code-Based LLM Access Architectures - A Systematic Analysis, https://github.com/bodorin/Privacy-Risk-Profiling-in-Claude-Code-Based-LLM-Access-Architectures, 2026.

[10] H. He, W. Xu, R. Liu, J. Tang, B. Li, and X. Lin, "ECCC: Edge Code Cloak Coder for Privacy Code Agent," in *Proceedings of the 9th Widening NLP Workshop*, Suzhou, China, Nov. 2025, pp. 65–74.

[11] G. Gerganov, "llama.cpp: LLM Inference in C/C++," *GitHub*, 2025.

[12] Qwen Team, "Qwen2.5-Coder-7B-Instruct-GGUF," *Hugging Face*, 2025.

[13] Groq Inc., "GroqCloud API Documentation," *Groq Console*, 2026.

[14] NVIDIA Corporation, "NVIDIA NIM Microservices," *NVIDIA Developer*, 2026.

[15] Google, "Google Cloud Free Program," *Google Cloud*, 2026.

[16] Amazon Web Services, "AWS Activate for Startups," *AWS*, 2026.

[17] Anthropic, "Claude Code CLI," *npm*, 2026.

[18] musistudio, "Claude Code Router (ccr)," *GitHub*, 2026.

[19] ENISA (European Union Agency for Cybersecurity), "Guidelines on Security Measures under the NIS 2 Directive," *ENISA Publications*, 2024.

[20] IETF, "RFC 8996: Deprecating TLS 1.0 and TLS 1.1," Internet Engineering Task Force, Mar. 2021.

[21] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," in *Proceedings of the 2023 ACM Workshop on Artificial Intelligence and Security*, 2023, pp. 79–90.

[22] N. Carlini, M. Jagielski, C. A. Choquette-Choo, D. Paleka, W. Pearce, H. Anderson, A. Terzis, K. Thomas, and F. Tramèr, "Poisoning Web-Scale Training Datasets is Practical," in *Proceedings of the 2024 IEEE Symposium on Security and Privacy (S&P)*, 2024, pp. 1161–1178.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021, pp. 8748–8763.

[24] A. Grattafiori et al., "The Llama 3 Herd of Models," *arXiv preprint arXiv:2407.21783*, 2024.

[25] A. Yang et al. (Qwen Team), "Qwen3 Technical Report," *arXiv preprint arXiv:2505.09388*, 2025.

[26] A. Chowdhery et al., "PaLM: Scaling Language Modeling with Pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017, pp. 5998–6008.



Copyright ©2026 by the authors. Licensee Agora University, Oradea, Romania.
This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.
Journal's webpage: http://univagora.ro/jour/index.php/ijccc/



This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/
international-journal-computers-communications-and-control

Cite this paper as:

de Miranda, J.L.; Nagy, M.; Popescu-Bodorin, N. (2026). Privacy Risk Profiling in Claude Code–Based LLM Access Architectures – A Systematic Analysis, *International Journal of Computers Communications & Control*, 21(6), 7577, 2026.
https://doi.org/10.15837/ijccc.2026.6.7577