5.1 What's Actually Shipping on Intel NPUs

The most useful thing a book like this can do, in its closing chapter, is be honest about what is *really* deployed on NPU hardware today versus what is announced, planned, or aspirational. The gap matters. If you build your roadmap on press releases, you'll discover too late that the workload you assumed worked doesn't. This section surveys publicly documented NPU deployments, calls out what's measured versus marketed, and identifies the patterns that recur across them.

Microsoft Copilot+ PC: The Reference Deployment

The most public NPU case study is Microsoft's **Copilot+ PC** program. Certification requires ≥40 TOPS NPU, 16 GB RAM, 256 GB SSD, and Windows 11 24H2 or newer. On Intel silicon, **only Core Ultra 200V (Lunar Lake) qualifies** — Meteor Lake and Arrow Lake-S do not. This is the first hard truth about NPU agent deployment: marketing covers many SKUs, but feature certification covers very few.

Features confirmed to run on the NPU on Copilot+ PCs include Windows Recall, Live Captions with Translation (40+ languages), Studio Effects (Background Blur, Eye Contact, Auto-framing, Voice Focus, Portrait Light), Cocreator in Paint, Restyle in Photos, Phi Silica, and Click to Do (which is hybrid NPU+cloud). Note what's *not* on this list: most third-party agents.

Phi Silica: The Gold Standard

Phi Silica is the published reference for NPU-resident agentic LLM inference. Microsoft's December 2024 Windows Experience Blog post is one of the most technically detailed NPU agent writeups available. The summary:

- Based on a derivative of Phi-3.5-mini (3.3B parameters)
- 4-bit weight quantization
- Prompt processing fully on NPU at **650 tokens/sec prefill, ~1.5 W**
- Decode at ~27 tokens/sec on **CPU**, reusing the NPU's KV cache (hybrid execution)
- Long prompts decomposed into 64-token chunks
- Speculative decoding with a smaller draft model
- Tokenizer/embedding/LM head on CPU, transformer block on NPU
- "650 tokens/second prefill, ~1.5W power" — *Microsoft Windows Experience Blog*

Crucially, **all published Phi Silica numbers are from Snapdragon X Elite hardware**, not Intel. Microsoft has not published Intel-NPU-specific Phi Silica figures. Phi Silica reached Intel Copilot+ PCs through Windows Updates (KB5079266, KB5084176, KB5089866) during 2025, but the comparative performance data isn't in the public record.

The Phi Silica architecture is the template the rest of the chapter draws on. Three patterns from it generalize directly:

1. **Hybrid NPU+CPU execution** for transformer LLMs (prefill on NPU, decode on CPU)
2. **Tokenizer/embedding/LM head on CPU** while the transformer block runs on NPU
3. **Speculative decoding with a smaller draft model** to amplify NPU throughput

If you remember nothing else from this chapter, remember that this is what production NPU LLM deployment looks like in 2025–2026: not "all on NPU," but a careful partition with the NPU doing what it's best at.

Quote Worth Internalizing

Microsoft's Phi Silica post contains the most concise statement of why NPU agents matter:

> "NPUs can sustain AI workloads that exhibit emergent behavior (3 to 7B parameter SLMs) in a semi-continuous loop, allowing users to make limitless low-latency queries to the model… we now have the ability to run powerful reasoning agents as part of background operating system services."

This is the architectural shift the whole book has been pointing at. Not "AI in the cloud, called through a network." Not "AI on the GPU, blocking the user's foreground work." Something genuinely new: agents that live in the OS, available continuously, at a power budget the user doesn't notice. The NPU is what makes that economic.

Adobe: Documented, Limited

Adobe Premiere Pro's Audio Category Tagger is the only Adobe feature jointly confirmed by Adobe and Microsoft to run on Intel NPU (via DirectML, announced November 2024). Other Adobe AI features run differently:

- **Enhance Speech, Scene Edit Detection**: GPU via DirectML, not NPU
- **Photoshop Generative Fill**: cloud
- **Lightroom AI Denoise on Apple Silicon NPU**: enabled, then suspended due to artifacts

That last one is the cautionary tale. A shipped NPU feature was *withdrawn* because users noticed visual artifacts that didn't appear in the GPU/CPU path. This is exactly the failure mode Chapter 4's pre-flight validation is meant to catch.

Audacity and OBS Studio: Open Source NPU Agents

The cleanest open-source NPU case study is `intel/openvino-plugins-ai-audacity`. It's a plugin suite for Audacity exposing:

- DeepFilterNet noise suppression
- Demucs music source separation
- MusicGen audio continuation
- Whisper transcription
- Audio super-resolution

It includes a runtime device selector for CPU / GPU / NPU. The plugin docs explicitly warn users: "**10 to 30 seconds the first time** you run this effect, then on-disk caching kicks in." This is the right user-facing communication pattern — it sets expectations and shows you trust the user to understand cold-start.

OBS Studio has a similar plugin set (`intel/openvino-plugins-for-obs-studio`) for smart-framing and face-mesh effects on NPU.

These are good case studies precisely because they're open source. You can read the device-selection code, the fallback paths, and the user-facing communication patterns. They are also useful negative examples: notice how much code is required just to expose NPU as an option in a desktop app.

Gaps in the Public Record

Several Intel partners have been announced but have *no public Intel-NPU latency or quality numbers*:

- **DaVinci Resolve**: forum threads as of 2025 confirm Resolve does not engage the NPU even on supported hardware
- **Topaz Photo AI / Video AI**: no NPU support
- **CyberLink PowerDirector, Skylum Luminar Neo, BUFFERZONE, McAfee Deepfake Detector, Rewind, Deep Render**: announced Intel partners, no public numbers
- **Zoom, Webex, Teams**: use Windows Studio Effects (NPU) when present, but no published quality comparisons

Dell and Intel co-marketing claims **38% more battery life on a Zoom call** with NPU engaged (Dell KB 000223944). This is one of the few quantified third-party numbers in circulation, but it's a power claim, not a quality claim.

Intel-Published Benchmarks Worth Citing

For NPU LLM throughput, Intel's OpenVINO Model Hub gives the most reliable numbers:

- **DeepSeek-R1-Distill-Llama-8B INT4 on Core Ultra 7 NPU**: **6.10 tokens/sec, 163.10 ms/token**
- Same model on iGPU: 19.80 tokens/sec
- Same model on Arc B-Series dGPU: 75.75 tokens/sec

That single data point captures the entire economic argument for NPU agents: a 7-watt NPU delivers a third the throughput of a 15-watt iGPU. Use the NPU for sustained low-power workloads; use the iGPU when latency matters and you have the power budget.

For Lunar Lake specifically, Intel disclosed Llama 3.2 3B numbers: TTFT 28.5 ms for 32 input tokens, 31.4 ms for 1024 input tokens, throughput 32–35 tokens/sec. Intel did not disclose whether this is NPU, iGPU, or hybrid — which itself is a tell about what they're willing to commit to.

No Intel-published numbers exist for M2M-100, NLLB, MarianMT, or any other encoder-decoder NMT model on the NPU. If you're using the book's M2M-100 example, you will be measuring on your own hardware — there is no public baseline to defer to.

What This Section Bought You

You now have an honest map of the NPU agent landscape:

- **Copilot+ PCs and Phi Silica** are the reference deployment — only Lunar Lake on Intel side
- **Hybrid NPU+CPU execution** with tokenizer/embedding/LM head on CPU is the production pattern
- **Adobe, Audacity, OBS Studio** are the documented third-party deployments
- **Many announced partners haven't shipped measurable NPU features** — be skeptical of roadmaps
- **Intel's OpenVINO Model Hub** is the best public benchmark source — though biased toward LLMs
- **No public seq2seq translation benchmarks exist** on Intel NPU — you'll measure your own

The next section is where the rubber meets the road: building an end-to-end agentic translation assistant using the lessons of the book. We'll see how the abstract patterns turn into actual code, actual numbers (where they exist), and actual trade-offs.

Next: *5.2 A Worked Agentic Translation Assistant*

Revision #2
Created 2026-05-12 17:28:42 UTC by Admin
Updated 2026-05-12 18:59:55 UTC by Admin