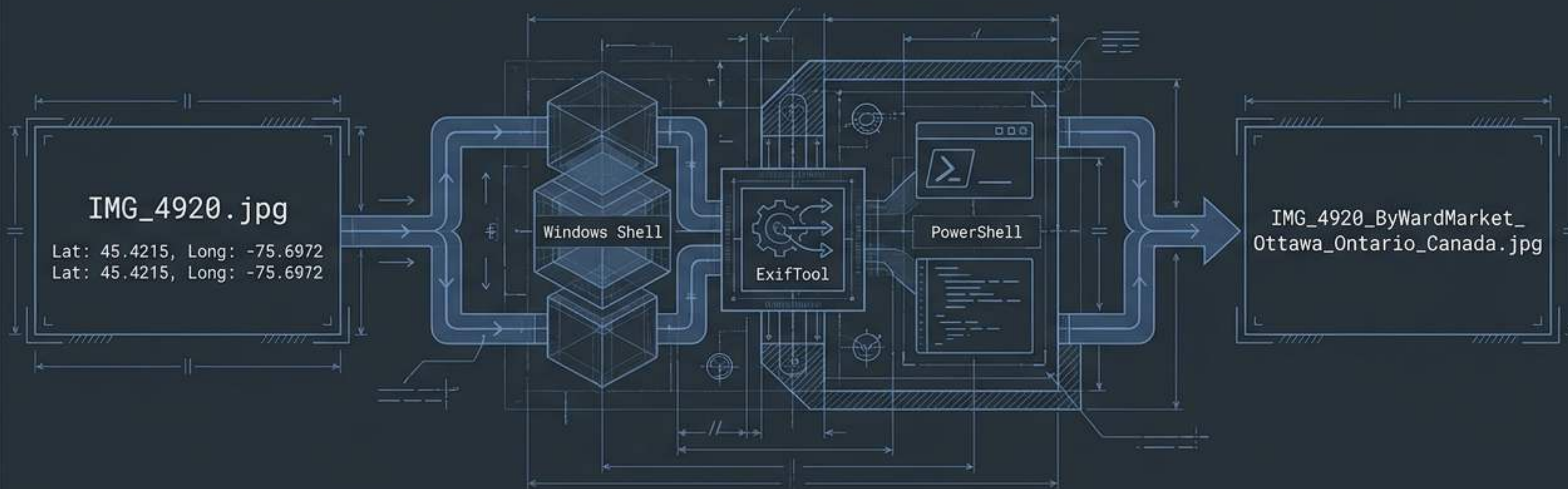
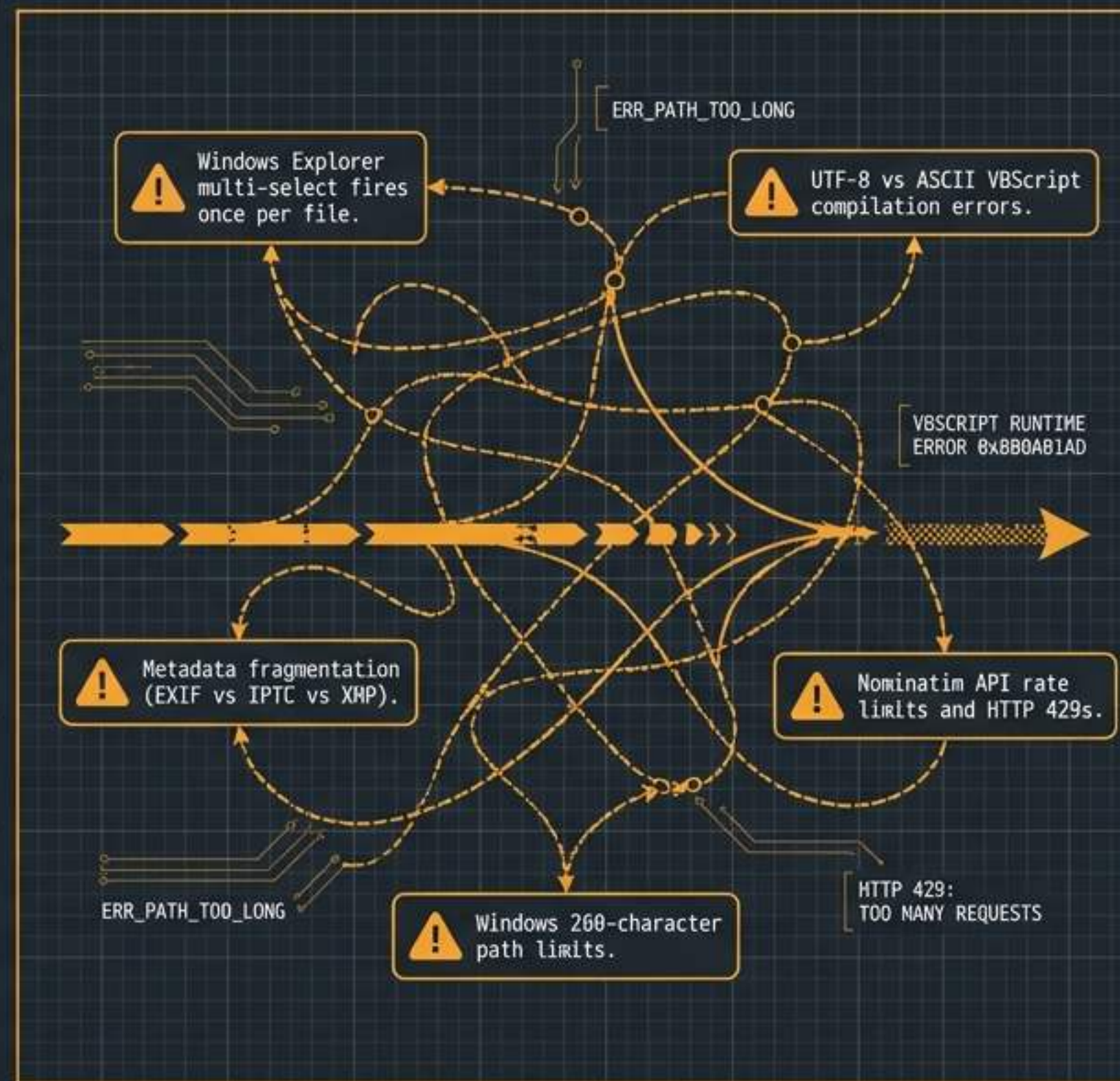
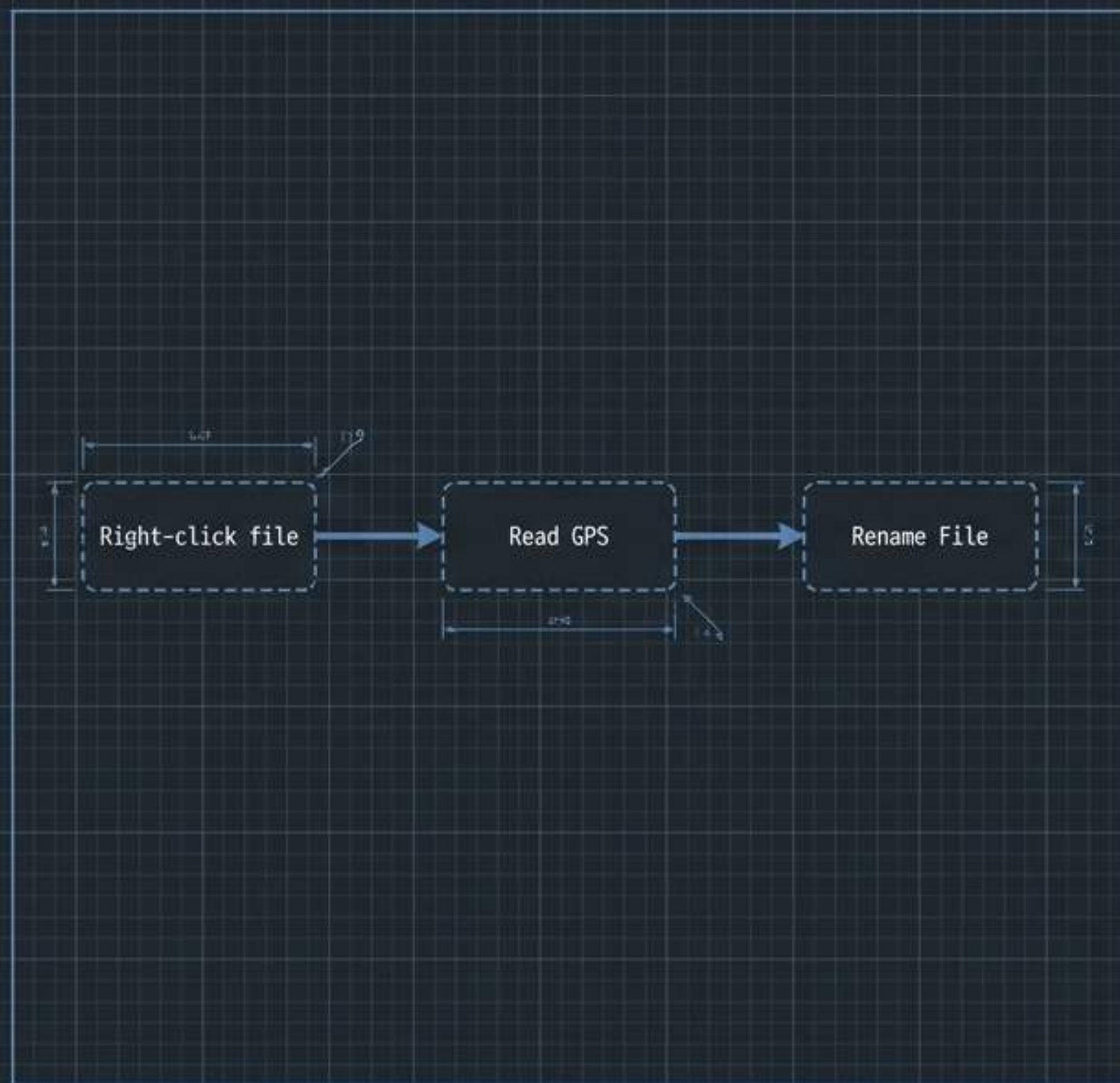


The Wandering Path to GeoNamer

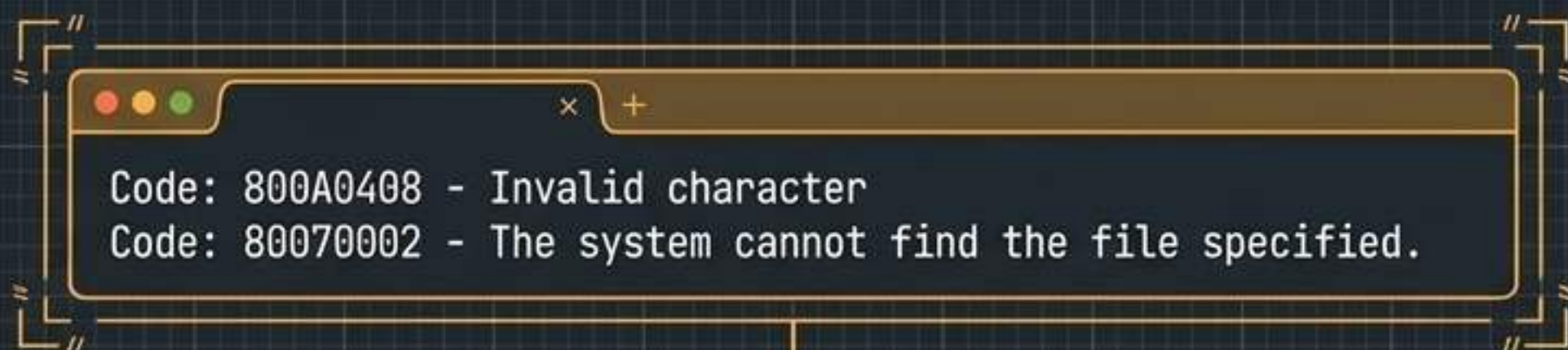
Navigating Metadata, Windows Shells, and the Art of Vibe Coding



A simple user objective hides a labyrinth of system edge cases



The journey begins with a fatal 80070002 script execution error



A hidden UTF-8 Byte Order Mark (BOM) corrupts the executable path and crashes the Windows Script Host parser.

```
shell.Run cmd, 0, False
```

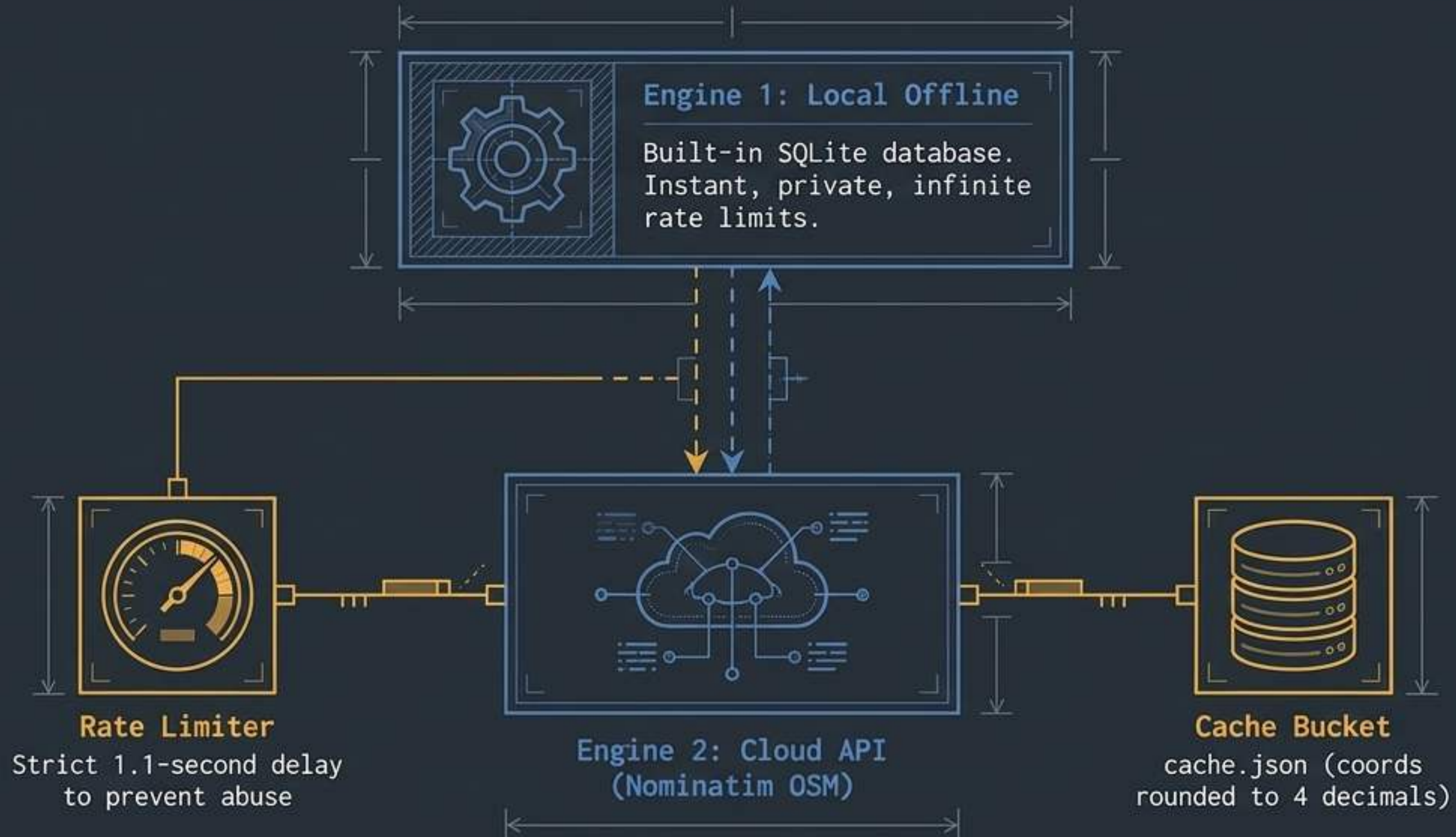
Fix: Force plain ASCII encoding, strip BOM, and hide the console.

"Location" is scattered across decades of fragmented metadata standards

EXIF	The Camera Standard. Good for raw GPS coordinates; terrible for structured location text (often crammed into ImageDescription or XPCComment).
IPTC	The Legacy Press Standard. Contains structured fields (City, Province) but is outdated and often conflicts with modern tags.
XMP	The Modern Standard. Extensible, highly structured, but frequently lives completely outside the image in .xmp sidecar files
QuickTime/MP4	The Wild West. Highly variable formatting depending on the camera manufacturer.

GeoNamer must intelligently parse ALL of these simultaneously, prioritizing the most accurate text source before falling back to GPS.

Reverse-geocoding requires a respectful, offline-first fallback architecture



Initial prompts dictate the what, but frequently fail the contract

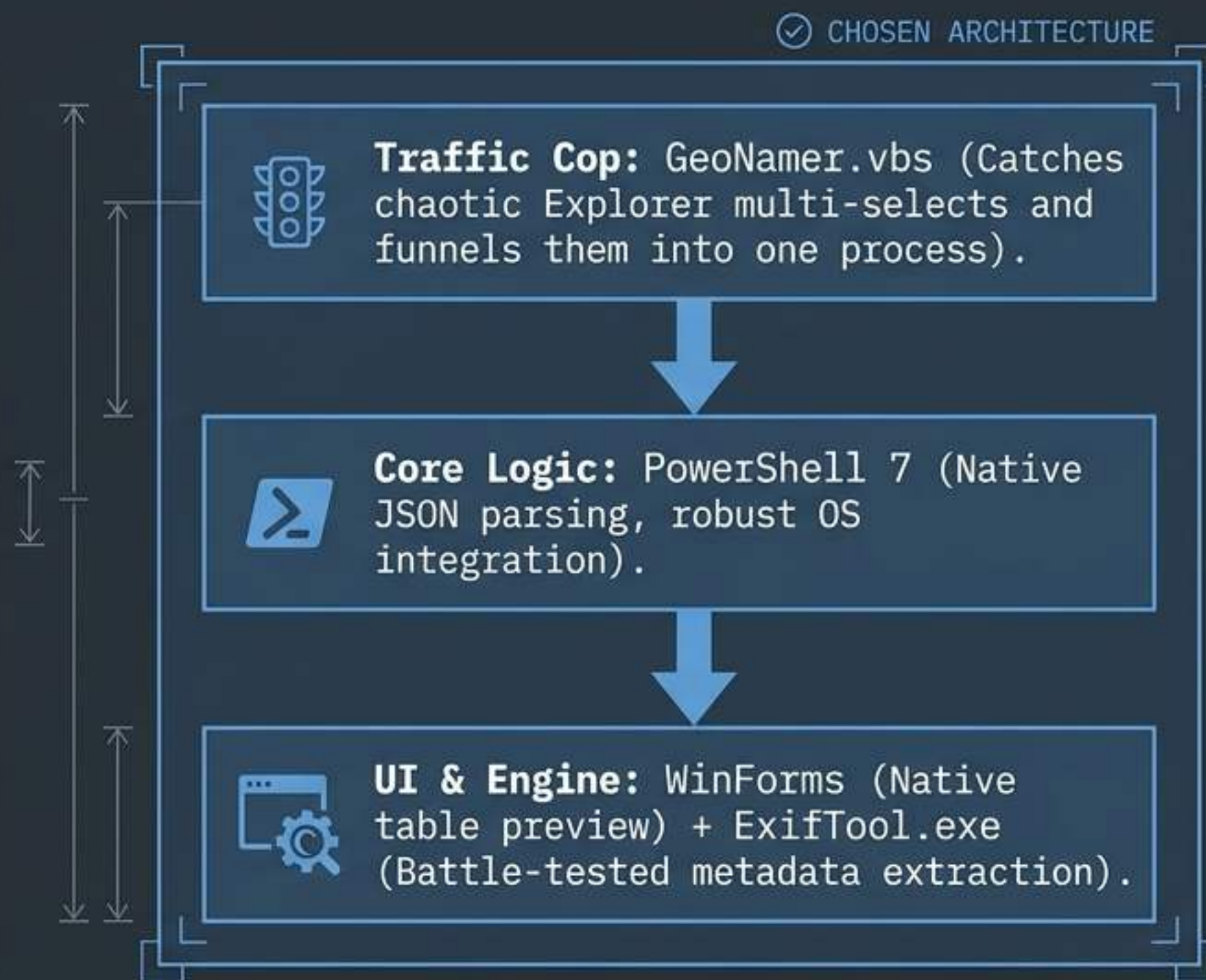
Make a windows program...
~~in C using tinyCC...~~
~~add~~ fields separated by
underscore...
~~work as a menu item~~
in File Explorer.

*Blindly renaming
files is destructive.
We need an
interactive user
preview.*

*C/TinyCC is a terrible
fit for building GUIs
and parsing JSON.*

*Explorer fires the
registry command
once per selected file.
This naive approach
will spawn 50
concurrent windows.*

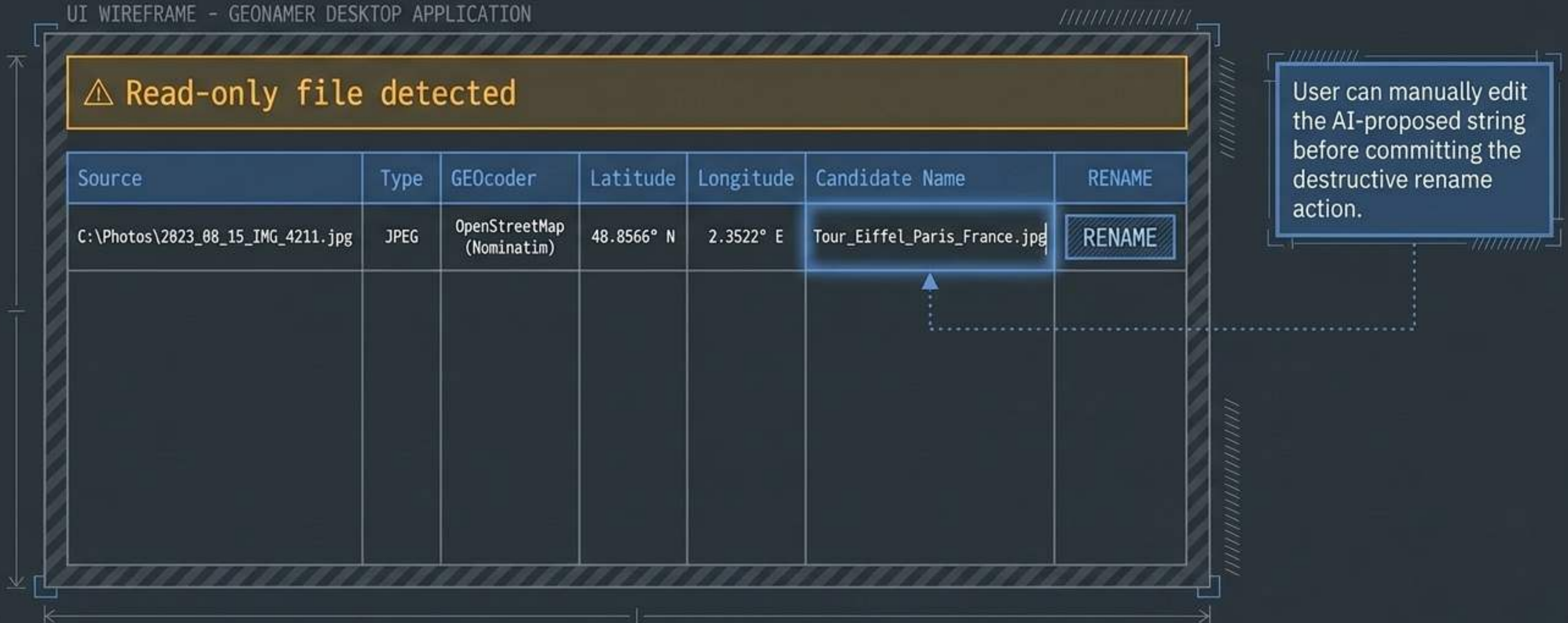
A reliable Windows utility requires aligning the stack with OS realities



A structured algorithmic hierarchy guarantees the most accurate filename



The interface must guarantee safe, transparent, and user-editable previews



Meticulous safeguards bridge the gap between a raw script and robust software

Path Limits

Truncating base filenames to stay strictly under **240 characters** (avoiding the fatal Windows 260-character path crash).

⚠️ C:\...\[241+ chars] ❌

Character Sanitization

Replacing illegal OS characters **/ \ : * ? < > |** with safe dashes or omitting them entirely.

/ \ : * ? < > | → Safe-Filename_V2.ext

Duplicate Prevention

Appending **_2**, **_3** to identical names to prevent accidental data overwriting.

[Photo.jpg] → [Photo.jpg]

[Photo_2.jpg] → [Photo_3.jpg]



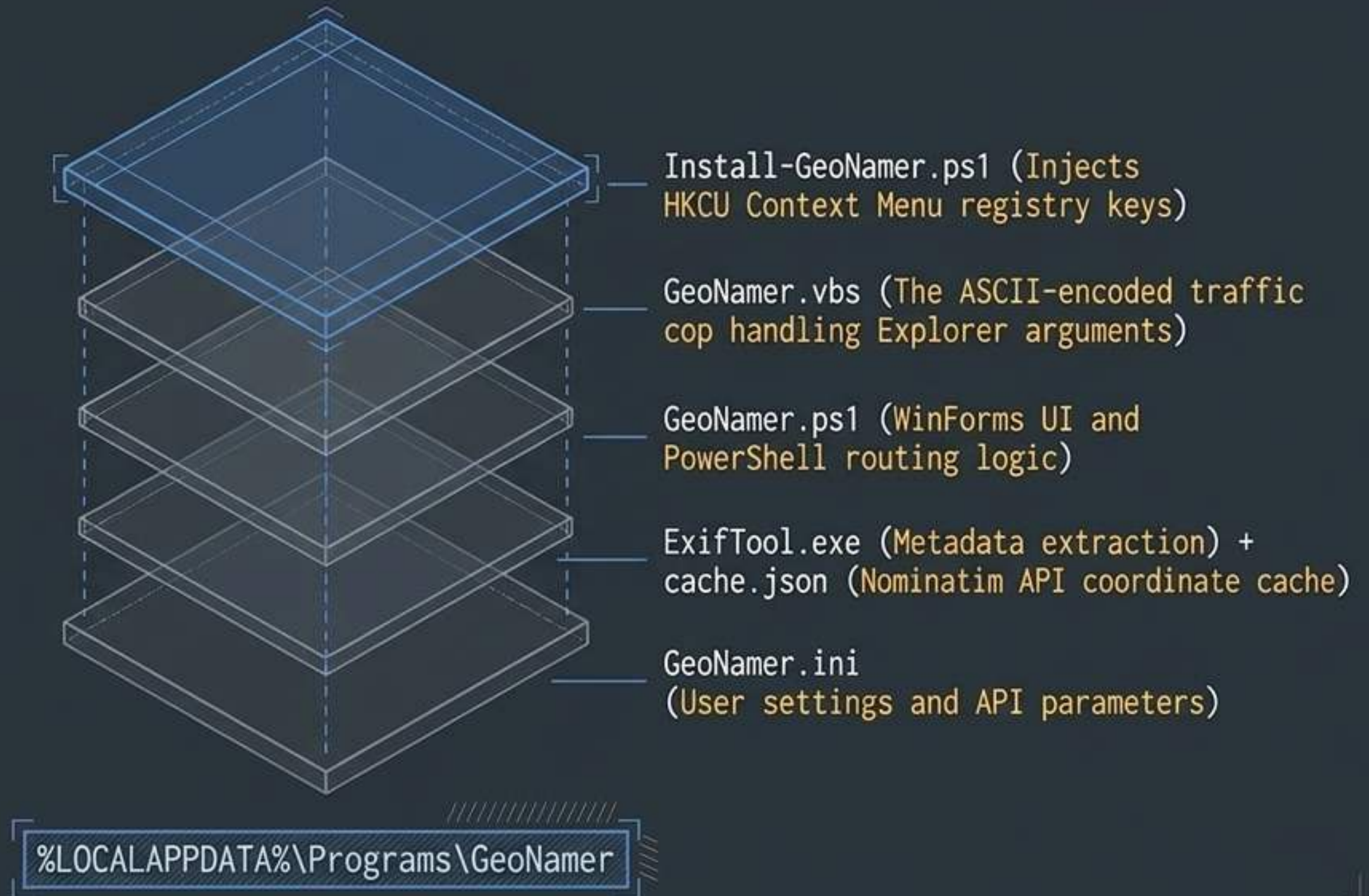
Sidecar Syncing

Ensuring that if IMG_4920.jpg is renamed, IMG_4920.xmp is automatically renamed to match.

[IMG_4920.jpg] + [IMG_4920.xmp] → [NewName_Location.jpg] + [NewName_Location.xmp]



The final GeoNamer package operates as a harmonious system of lightweight components



You are the Director; the AI is the Developer

Define the Intent:

Start with the user experience and the non-negotiable contract, not the exact code syntax.

Invite Critique:

Ask the AI to actively poke holes in your naive architecture (e.g., 'Will this break in File Explorer?').

The art of vibe coding is negotiating the reality of the system.

Harden the Edges:

Iterate relentlessly on silent errors, text encoding, rate limits, and destructive edge cases.

/ \ minopoint.jpg
⚠ exfhanax.jpg

Negotiate the Stack:

Let the AI suggest the path of least resistance (e.g., pivoting to PowerShell/WinForms over C/tinyCC).

/ \ ?!ttyocl | → GeoNamer_V2.ext
⚠ coonrma.json