OPENAI ARCHITECTURE PRIMER

Codex SDK

A thread orchestration boundary

A visual guide to thread identity, runtime paths, result contracts, input authority, configuration precedence, and sandbox state.

FIRST-PARTY DOCUMENTATION / INDEPENDENT LUMINITY SYNTHESIS



The SDK Is a Thread Orchestration Boundary

The application owns identity, continuation, result handling, effective configuration, and downstream consequence.

SIX APPLICATION DECISIONS

01
Start
new thread

02
Continue
same handle

03
Resume
saved identity

04
Receive
result or events

Configure 05

Resolve client, raw, SDK-managed, thread, and turn settings in a visible precedence chain.

Authorize 06

Treat input authority, filesystem scope, approvals, and downstream action as application controls.

IN THIS PRIMER

Key Points

01 Choose the runtime path

TypeScript wraps the CLI; Python controls a local App Server. The lifecycle is shared, the coupling is not.

02 Own thread and input state

Separate a live handle from a saved ID, completed output from streamed progress, and user input from external content.

03 Resolve authority before use

Record effective configuration and sandbox state, then authorize any consequence outside the SDK result.

The six-decision model is Luminity synthesis. OpenAI documents the underlying thread, runtime, input, configuration, and sandbox mechanics.

© 2026 Luminity Digital, Inc. | Independent synthesis of official OpenAI documentation.

Source: https://learn.chatgpt.com/docs/codex-sdk · verified Nov 5, 2026

OpenAI Select Partner

Choose the Runtime Path

TypeScript and Python expose one thread concept through different local process and protocol relationships.

01

TypeScript — CLI Wrapper

Node.js 18+ | @openai/codex-sdk | server-side



Completed Turn — RETURN

run() collects items, the final response, and usage into one return value.

Streamed Turn — EVENTS

runStreamed() exposes an asynchronous generator of structured thread events.

02

Python — App Server Client

Python 3.10+ | openai-codex | pinned runtime



Stable Package — PACKAGE

Stable installs use the matching pinned Codex CLI runtime dependency.

Lifecycle Breadth — STATE

The current Python surface includes resume, fork, history controls, and turn handles.

03

The Selection Decision

Host Fit — STACK

Choose the language that belongs in the application runtime.

Process Ownership — OPS

Decide who starts, observes, updates, and stops the local runtime.

Protocol Need — CLIENT

Use direct App Server integration when building a richer client protocol surface.

Release Contract — DRIFT

Pin and record the package and effective runtime used by the host.

© 2026 Luminity Digital, Inc. | Independent synthesis of official OpenAI documentation.

Source: https://learn.chatgpt.com/docs/codex-sdk · verified Nov 5, 2026

OpenAI Select Partner

Own Thread, Result, and Input State

A live handle, a persisted ID, a completed result, an event stream, and external content are different contracts.

01

Thread Identity

✓ startThread / thread_start	NEW IDENTITY
✓ run again on the same object	CONTINUE
✓ resume by saved thread ID	NEW HANDLE
✓ fork saved history	NEW BRANCH
✓ include_turns=False	HISTORY-LIGHT

Handle · Thread ID · History

02

Result Contract

Completed result · Streamed progress · Active turn handle



COMPLETION

Final response is an outcome signal, not proof of correctness

STREAMING

Progress events do not authorize downstream consequence

03

Input Authority

User Input — HIGHER

Plain strings and text input represent user-level instructions in the SDK contract.

External Message — LOWER

Content from tools, applications, or other agents retains tool-level authority and does not create approval.

Application Decision — OWNER

Preserve provenance, validate results, and authorize every external effect separately from the turn.

STATE RULE

- Thread ID ≠ live handle
- Progress ≠ outcome
- Output ≠ authorization
- External content stays lower-authority

Luminity Digital
INTELLIGENCE, ENGINEERED. DELIVERED.

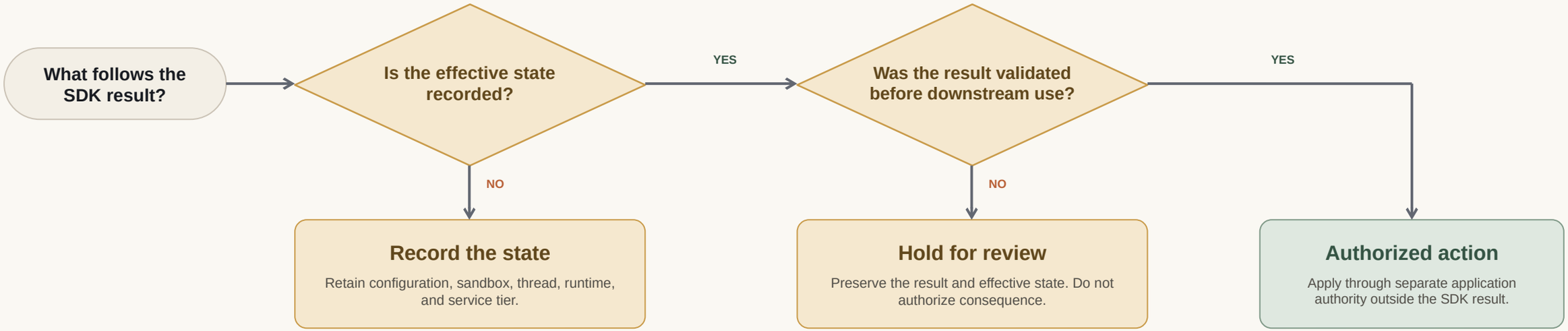
© 2026 Luminity Digital, Inc. | Independent synthesis of official OpenAI documentation.

Source: https://learn.chatgpt.com/docs/codex-sdk · verified Nov 5, 2026

OpenAI Select Partner

Effective Authority Before Consequence

Record the effective state, validate the result, and authorize every downstream action separately.



RETAIN THIS RECORD WITH EVERY SDK RESULT

Runtime + Release — PACKAGE

Retain SDK package, CLI or App Server version, model, and service tier.

Thread + Input — STATE

Retain thread ID, turn source, working directory, and input provenance.

Effective Authority — CONTROL

Retain config chain, approvals, sandbox state, and any turn-level override.

Validation + Decision — REVIEW

Retain the result, reviewer, authorization record, and downstream action.

BOUNDARY

- Configuration determines capability
- Sandbox changes persist
- A result does not grant approval
- Consequence uses separate authority

Luminity Digital
INTELLIGENCE, ENGINEERED. DELIVERED.

© 2026 Luminity Digital, Inc. | Independent synthesis of official OpenAI documentation.

Source: https://learn.chatgpt.com/docs/codex-sdk · verified Nov 5, 2026

OpenAI Select Partner