# You’re here to fine-tune. But fine-tune **what*?

Before we turn any knobs, we need a clear mental model of the machine we’re working with. While tools like ChatGPT, Claude, and Gemini come from different companies, they share a core architectural DNA.

> This lesson decodes that DNA. You’ll learn the three families of language models, what each is for, and why one has become the undisputed foundation for modern AI engineering.

# It All Starts with the Transformer

All modern language models are built on the Transformer architecture, introduced by Google in the 2017 paper, *Attention Is All You Need*.

## Core Innovation:

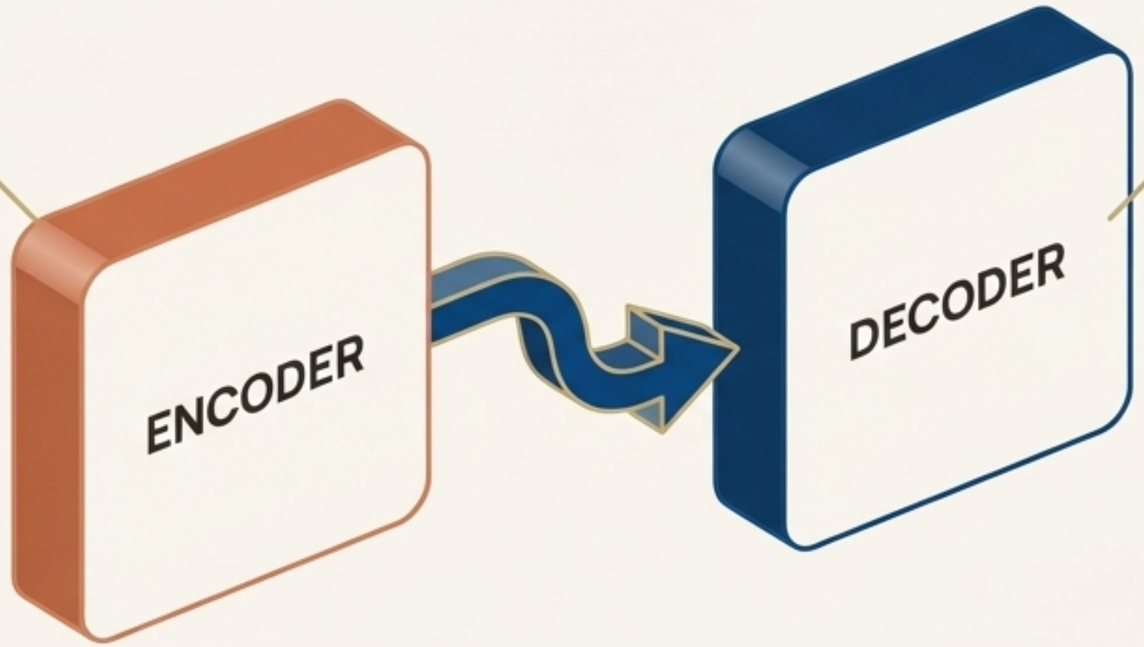
Its key breakthrough was the **self-attention mechanism**. This allowed for parallel processing of all tokens, breaking the sequential bottleneck of older models and enabling training at an unprecedented scale.

## Simple Example:

Self-attention is how a model knows that in the sentence, “The bank was steep,” the word *bank* relates to a river, not money.

# The Original Blueprint: Two Parts for One Job

Reads & understands the full input. It looks both forwards and backwards to build deep context.

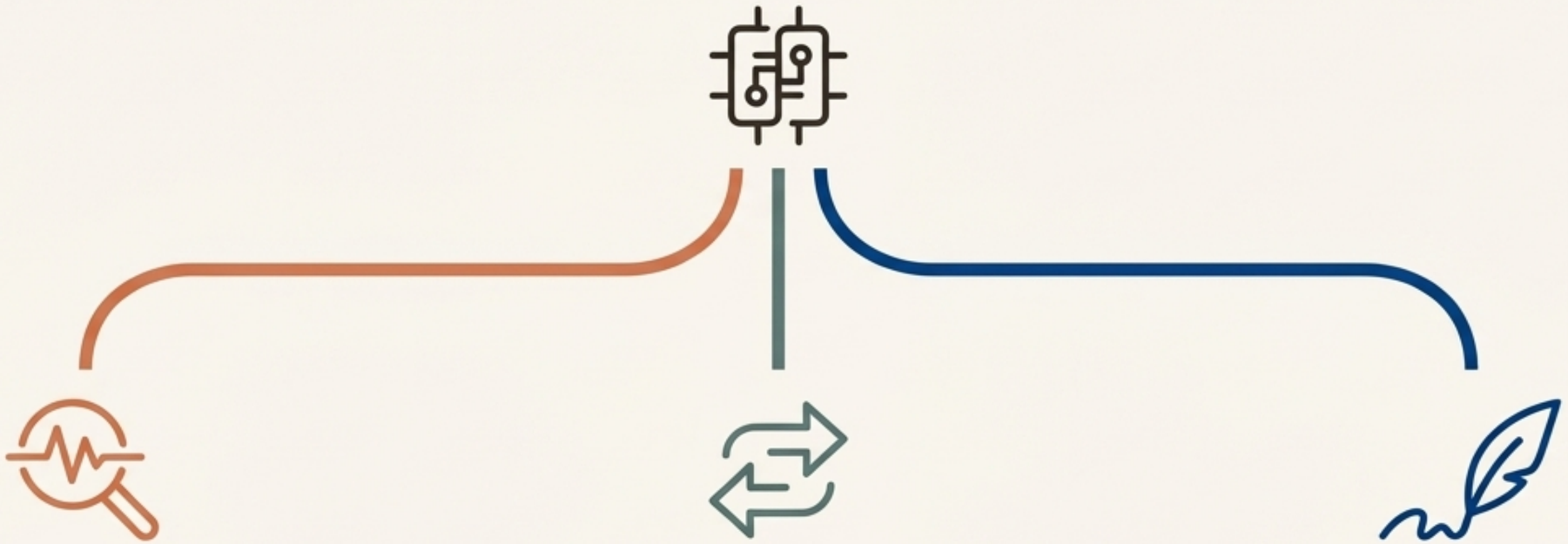



Generates the output, one word at a time, using the Encoder's understanding and what it has already written.

This full encoder-decoder setup was designed for a specific task: machine translation. The encoder understands a sentence in one language, and the decoder writes it in another.

# One Blueprint, Three Specialized Architectures

Researchers quickly realized you don’t always need both parts. Depending on the task, you can build powerful models using just the encoder, just the decoder, or both. This gave rise to three distinct families.



### The Analyst (Encoder-Only)

For deep understanding and classification.

### The Translator (Encoder-Decoder)

For sequence-to-sequence transformation.

### The Author (Decoder-Only)

For fluent, creative text generation.

# The Analyst: Encoder-Only Models

Uses only the **encoder block**. It looks at the entire input at once (bidirectional context) to build a deep, nuanced understanding. It excels at analysis but cannot generate fluent, new text.

**Great For**

- Text Classification (e.g., sentiment analysis)
- Named Entity Recognition (NER)
- Semantic Similarity & Embeddings

**Input::** “My order hasn’t arrived and it’s been 2 weeks.”

**Output:** `Category: "Shipping Issue", Urgency: "High"`

**Key Models:** BERT, RoBERTa, DistilBERT

# The Translator: Encoder-Decoder Models

The original full Transformer. The **encoder** fully understands an input sequence, and the decoder transforms it into a new output sequence.

**Great For**

- Machine Translation
- Summarization
- Paraphrasing & other “X → Y” tasks

**Input:** *A long English document about quantum computing.*
**Output:** *A short French summary of the document.*

**Key Models**

T5, BART, PEGASUS, Original Transformer

# The Author: Decoder-Only Models

Uses only the decoder block. It generates text one token at a time, left-to-right, based on the preceding context. This autoregressive process is ideal for creating new, coherent content.

**Great For**

- Chat Assistants & Instruction-Following
- Code Generation
- Email Drafting & Creative Writing

**Input:** “Write a polite email declining a meeting request.”

**Output:** “Dear [Name], Thank you for the invitation... [polite decline message]”

**Key Models:** GPT series, Claude series, LLaMA series, Mistral, Gemini

# Architecture Showdown: Understanding vs. Generation

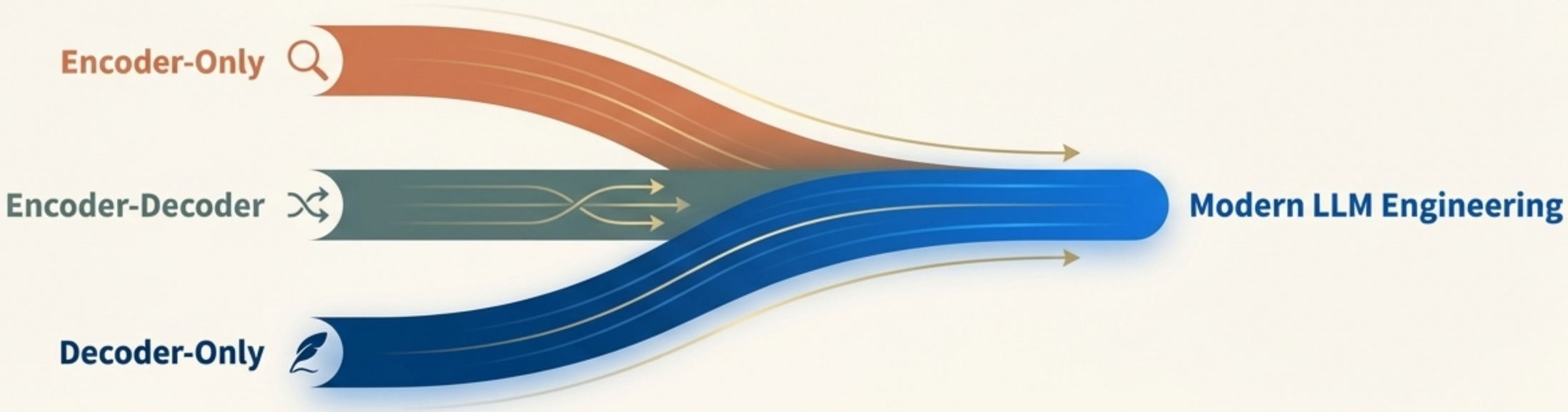
| | The Analyst | The Translator | The Author |
|---|---|---|---|
| Text Processing | Bidirectional (Sees full context) | Hybrid (Encoder bidirectional, Decoder unidirectional) | Unidirectional (Left-to-right) |
| Primary Strength | Deep Contextual Understanding | Sequence Transformation | Fluent Text Generation |
| Can It “Chat”? (Generate Fluent Text) | ✕ NO<br>Cannot generate new, fluent text. | ! LIMITED<br>Yes, but focused on transforming inputs. | ✓ YES<br>Yes, its core function. |
| This Program’s Focus | Not Covered | Not Covered | ✓ Primary Focus |

# The Great Convergence: Why *One* Architecture Dominates

The modern LLM landscape has decisively shifted. While all three architectures are powerful, one has emerged as the clear foundation for nearly every major AI product and open-source project today.

**When most people say “LLM” today,
they mean a Decoder-Only model.**

The next three slides break down the key reasons for this dominance:
Versatility, Scalability, and a mature Ecosystem.



# Reason 1: Unmatched Versatility

A single, well-trained **Decoder-Only** model can act as a chat assistant, a code generator, a translator, a summarizer, and even a reasoning engine.

**The Big Shift**
Tasks that once required specialized models are now handled effectively by general-purpose “Authors.”

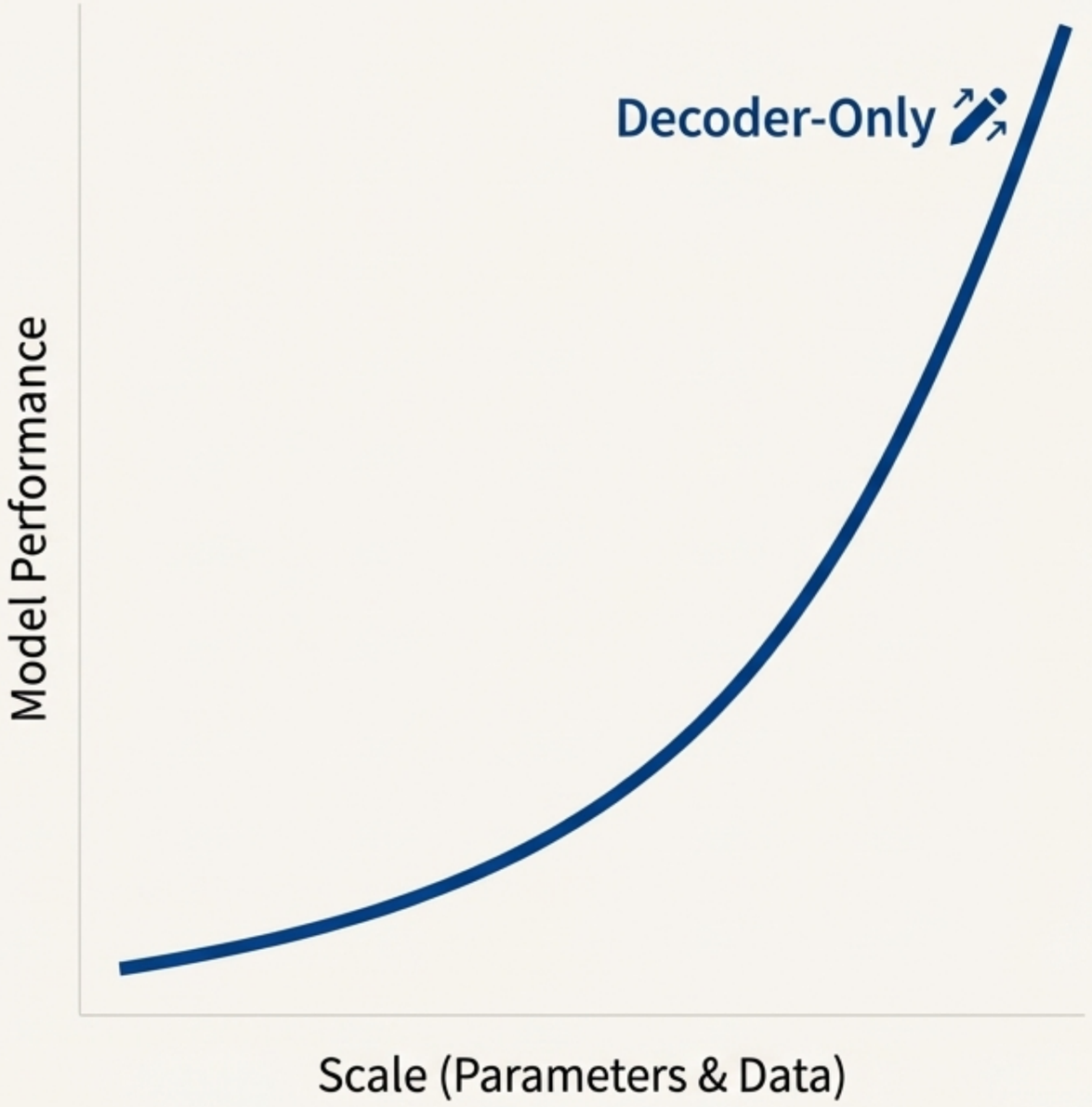
- Classification (formerly Encoder-Only)
- Translation & Summarization (formerly Encoder-Decoder)

**Business Impact**
This flexibility allows organizations to rely on a single model architecture across dozens of use cases, simplifying management and deployment.

# Reason 2: Predictable, Massive Scalability

Decoder-Only models scale extremely well. Performance continues to improve predictably as model size and training data (trillions of tokens) grow.

“This predictable scaling has made them the architecture of choice for both cutting-edge research and enterprise deployment.”



# Reason 3: A Mature, Thriving Ecosystem

The entire open-source and commercial ecosystem has consolidated around **Decoder-Only** models. When you fine-tune, evaluate, or deploy an LLM in the real world, this is the architecture you will be working working with.

PEFT

DeepSpeed

Axolotl

OpenAI

ANTHROPIC

Google

Meta

# The Model vs. The Product

When we say ChatGPT, Claude, or Gemini use a Decoder-Only architecture, we're talking about their **core text-generation engine**.


**Key takeaway:** This program focuses on the core language model itself: how to fine-tune and deploy it for your own use cases.

# This is the Architecture that Powers Modern AI

You now have a clear mental model of the three LLM families and a solid understanding of why the industry has converged on one. Our entire focus is on mastering this dominant architecture. You will learn to fine-tune these powerful Decoder-Only models for your own use cases—whether you're building a customer service assistant, a domain-specific agent, or something entirely new.

By the end of this program, you'll know how to make this technology your own.