

The Composable vs Headless Decision Worksheet

The question is not which term wins. It is how far down the stack you actually need to go. This is how to tell.

The distinction in one table

Headless is one split. Composable is splits all the way down. Read across before you decide how far to go.

	Headless	Composable
What it separates	Front end from back end	Every function into swappable pieces
Scope	One boundary	The whole stack
Back end	Can be one unified platform	Best-of-breed pieces you connect
Contains the other?	No	Yes, it is headless plus more
Cost / complexity	Smaller move	Much bigger commitment
Fits when	You need front-end / experience freedom	You need flexibility across every function

The one line to remember: all composable is headless, but not all headless is composable. You can have a custom front end on a single unified back end, that is headless, not composable.

The scope question that decides it

Not which term sounds advanced. How far do you actually need to go?

Do you need front-end / experience freedom?

A custom or multi-channel storefront, while your commerce engine is fine. Headless may be enough.

1	2	3	4	5
---	---	---	---	---

Do you need flexibility across every function?

Different best-of-breed needs per function, swapping components over time. This is composable.

1	2	3	4	5
---	---	---	---	---

Is your current commerce engine actually a problem?

If not, full composable may be replacing something that works with a lot you now have to run.

1	2	3	4	5
---	---	---	---	---

The trap: going composable because it sounds most advanced, when headless would have met your need at a fraction of the cost. Composable is powerful and a lot to run, many pieces, many integrations, many vendors, all yours to connect. Match the change to the need, not the fashion.

Where are you now, and where do you want to be?

Today we are: monolithic / headless / composable, and the specific reason we would move

If you cannot name a specific need the next level solves, you may already be where you should be.

—

—