

All-in-one solutions are a trap

# Why *specialists* are the right choice for your infrastructure

It's hard to create a multiplayer game. But these days, it's almost a necessity to keep your game alive. So you're going to need specialists.



## The gaming industry has a problem

# What we'll cover

We've noticed that many publishers and studios turn to all-in-one solutions for their backend services. But this approach can lead to major issues down the line, **like skyrocketing prices or downtime.**

But if it's a problem, why do studios do it? What can go wrong? And how can you avoid hitting the same pitfalls?

1.

Why do developers often struggle with infrastructure?

2.

What's the problem with an all-in-one?

3.

How can you avoid the pitfalls?

Why is there a problem?

# Studios are struggling with infrastructure

It's hard to create a multiplayer game. But these days, it's almost a necessity. **In fact, Unity found that about 60% of studios are focused on creating a multiplayer game.**

Yet studios don't have the in-house experience to develop and manage the infrastructure they need. Creating a game has very different challenges to running a network. It's just a distinct problem space.

This is why some studios end up turning to an all-on-one solution. Generally, that's a service that includes their matchmaking, their orchestration, and live service systems like user authentication and management. All the backend tools that a studio can't create themselves.

It's the simple answer to a complex issue. **An answer that locks the studio into an ecosystem that tries to do everything,** but is rife with compromise and limitations. This makes it very difficult to leave, if the studio wants to change.

The reason there's a problem

# Why do studios choose all-in-one?

Typically there are three reasons studios turn to an all-in-one solution:

- **It's appears simpler.** But that simplicity comes at a cost: *expertise*.
- **It's seems to be easier.** Because studios lack the skills around backend services, it's difficult to compare and research the best tool.
- **It looks like it saves time.** Getting everything from one provider sounds like it'll save time setting it up. But in the long run, studios find that it takes far longer to migrate away when they realise their provider isn't right for them.

From our experience, the smaller the studio the more likely they'll turn to an all-in-one provider. They assume it's the safer, easier option. But the reality is that it ends up **either costing more for server space or to migrate away.**

### Three core issues

# What are the main problems with all-in-one?

While it might seem simpler to use an all-in-one solution, it can lead to some large problems down the line.

**Your costs can rack up**

**You create a single point of failure**

**You have to deal with generalists**

## Problem one

# Your costs can rack up

Once you're locked into a single provider for everything, you're stuck with their pricing model. For example, many all-in-one providers claim you'll be able to scale up using the cloud. **But they won't offer a fixed price.** If your game is successful, you can suddenly be hit with a huge bill that you weren't expecting.

Another issue is that all-in-one providers can offer one service for incredibly low prices to entice you in, but another service will be incredibly expensive or the prices can suddenly increase after a year. **These hidden costs can catch studios out.**

Don't fall for that enticing upfront offer. Instead, look for how the rates could increase after the deal.

### **Palworld ended up paying \$475,000 in a month**

After becoming a viral hit in 2024, **Palworld found itself hammered by huge server costs.** Likely because they needed to scale into the cloud. This meant they had to scramble to find ways to lower their costs.

## Problem two

# You create a single point of failure

According to research from Ponemon Institute in 2016, the average cost of server downtime, regardless of industry, is \$8,500. **And a single outage can cost around \$740,000.** That includes the direct cost to fix the problem, the indirect cost for the time to fix it, and the opportunity cost.

These outages can happen to any company, not just gaming. Power can fail, cyber crime can strike, and people make mistakes.

So if you rely on a single provider, it's almost inevitable that you'll have problems. With no backup plan, you leave yourself vulnerable.

### Helldivers 2 had to cap their players

In March 2024, **Helldivers 2** capped their players to **450,000** to 'improve server stability.' They've raised the cap a few times, but still struggled. They even went on an emergency hiring spree as the team was "completely exhausted."

### Doborog's matchmaker got discontinued

Our client, **Doborog**, was using a matchmaker quite happily for a while. But then their provider switched how everything worked. Doborog realised that this was a good opportunity to create their own, simpler matchmaker.

### Problem three

# You have to deal with generalists

An all-in-one provider simply can't specialise in everything. They might have an excellent matchmaker, but that doesn't mean their user authentication is the best. You end up with a generalist, where everything is *okay*, but not exactly what you need.

For example, studios can find themselves needing to adapt their game to suit the technology. If you wanted to have players drop into a skirmish between matches or use a different matchmaking ranking algorithm, you're relying on your matchmaker having the capability. **When you're stuck with a specific provider, you're stuck to their set of rules.** Rules that might not be suitable for your game.

And if there's a problem, you won't have experts there to fix it. The longer it takes to fix, the more the bad reviews pile up. With such a competitive market, those bad reviews can be a death sentence for a new game.

## **Payday 3's matchmaker was faulty, causing the game to go down**

Straight out the gate, **Payday 3** struggled with their matchmaker. Months down the line, and the problems still continued. In one case, a faulty update from the matchmaker caused Payday 3 to grind to a halt.



Keep yourself covered

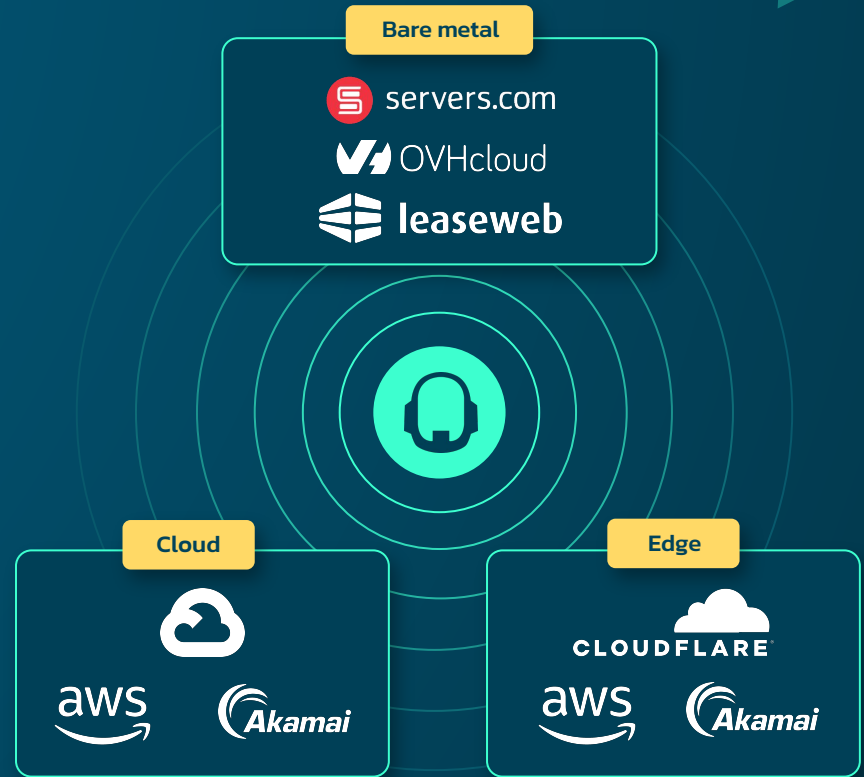
How can you avoid  
these pitfalls?

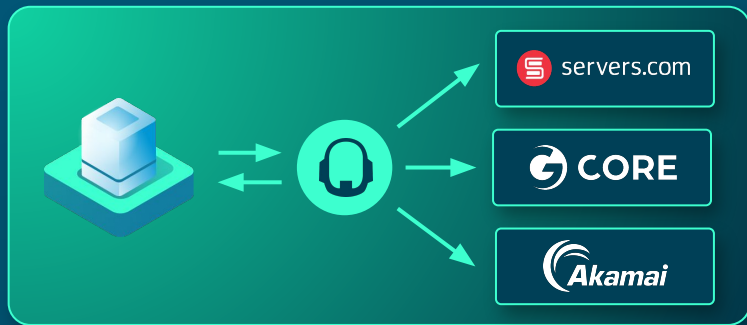
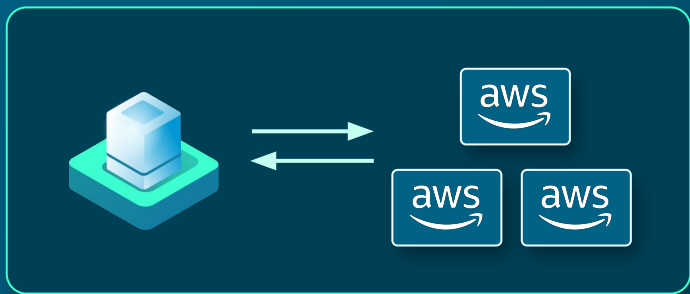
## Branch out

# Look for agnostic solutions

**Your backend systems don't need to be tied together.** You can look for services that are agnostic and can integrate with any other system. This way, you always get the best technology available.

Take server providers. If you only rely on one server provider, you're in trouble if they go down. If you combine multiple providers together into your network, you have redundancies in place to make sure your game is always running.





## Change your mind

# It's always possible to switch

**Most studios (62%) decide on their infrastructure before production begins.** And it's always best to try and make these decisions early. Migrating later takes a lot of engineering effort and might result in downtime while you switch.

But that doesn't mean you're stuck. There are migration services out there to help you move away from an all-in-one provider.

**The biggest trap studios fall into is the sunk cost fallacy** – they've invested so much it feels like a waste to move now. But that money is already spent. Once you've recognised the problem, you should only look to the future.

Stay aware

# Think about *cost per user*

Multiplayer and liveops are crucial now to keeping your game alive. But not many liveop games have perfected their monetization model from day one.

Once you've figured out your monetization model, you'll need to recalculate your server costs. But if you're locked into an all-in-one provider, it's harder to leave and adapt. Similarly, what if they unexpectedly increase your prices? You'll need to do the same, which can anger players.

**We find it useful to think about the Average Cost Per User and compare it to the Average Revenue Per User.** Your Cost Per User should look at how long a player typically spends in your game – how many years will they keep playing? And how much will it cost you to keep your servers and systems running for that entire time?

**Average  
Cost Per User**

**Versus**

**Average  
Revenue Per User**

# What questions should you ask a potential partner?

- What are the resource limits for cloud services, and how do you manage them during periods of high demand?
- How do you make sure the physical servers are available? And what options do you offer if demand becomes higher than the available inventory?
- Are you locked into one single provider or can we make changes at a later date?
- What happens if we find a cheaper provider elsewhere?
- How can we incorporate your infrastructure into the network?
- How do you optimize server performance and resource utilization for gaming workloads?
- Do you provide tools or support for performance monitoring and optimization?
- What level of support do you offer for server orchestration and infrastructure management?
- Do you have dedicated experts available to assist with troubleshooting and optimization?

Get in touch

# Gameye can help with your orchestration

**We're trying to free people to design more interesting games.** We want to enable as many types of games as possible. And we believe in making sure you're not locked into any particular model of play.

So our orchestration platform allows you to scale between multiple server providers, automatically. You can always use the cheapest provider available. And, if one provider has problems, you can move over to another automatically.

**Cheaper hosting. No downtime. Full control.**



Let your creativity drive your game, not the technology.

Get in touch