

# What are Internal Developer Platforms?

## What is an Internal Developer Platform (IDP)

**A self-service platform created and maintained by platform engineering teams to simplify and standardise the software development process for your application developers and operations teams.**

## Internal Developer Platforms Explained

Often built on top of Kubernetes, an IDP provides application developers with secure, scalable workflows that reduce the complexity of managing infrastructure. This allows application developers to focus on coding, deployments and delivering value without deep Kubernetes or underlying infrastructure expertise. Meanwhile, platform engineering and operations teams retain control over policies, governance, and infrastructure standards, ensuring the platform remains efficient, compliant, and maintainable while supporting rapid software delivery.

An Internal Developer Platform (IDP) will comprise of these five properties:

- 1. Unified Tooling:** Brings together development tools into a cohesive interface, simplifying workflows.
- 2. Automated Workflows:** Automates tasks like provisioning and deployment, speeding up development.
- 3. Self-Service Capabilities:** Empowers developers to manage resources independently, reducing reliance on IT or DevOps teams.
- 4. Infrastructure as Code (IaC):** Ensures consistent, scalable, and automated infrastructure management.
- 5. Policy as Code (PaC):** Automates security and compliance enforcement, reducing errors and improving governance.

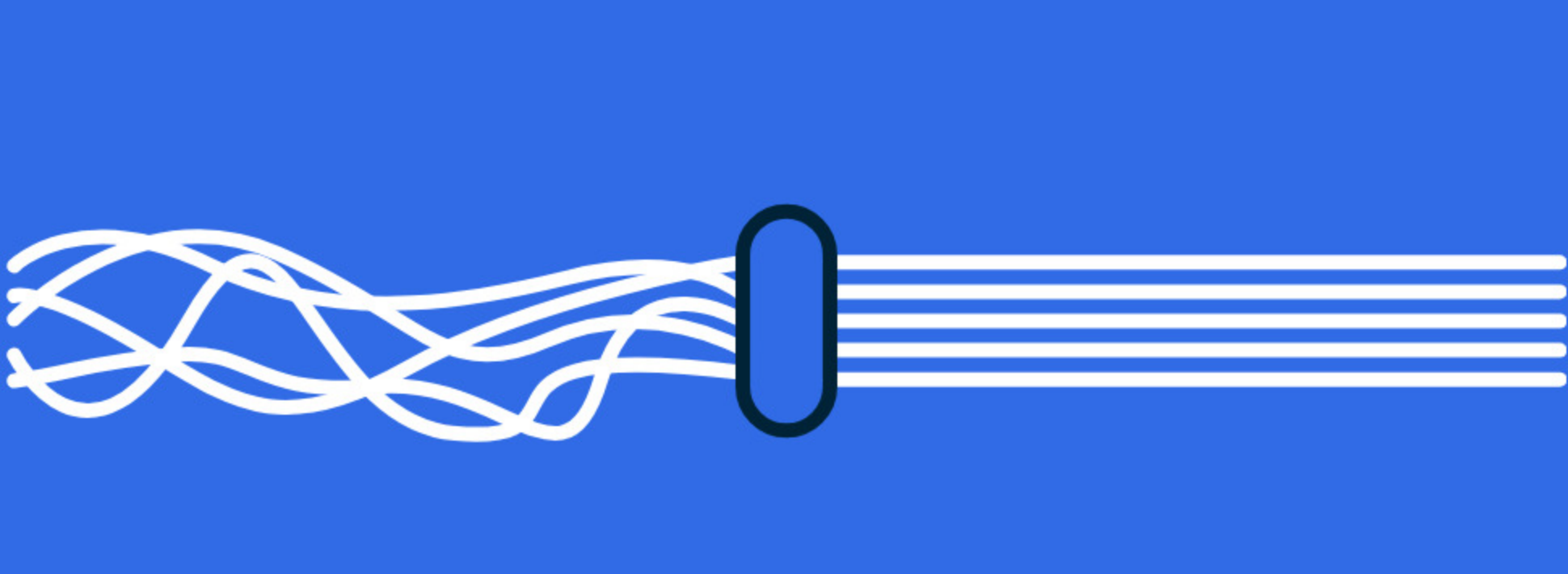
## Organisational Benefits of IDPs



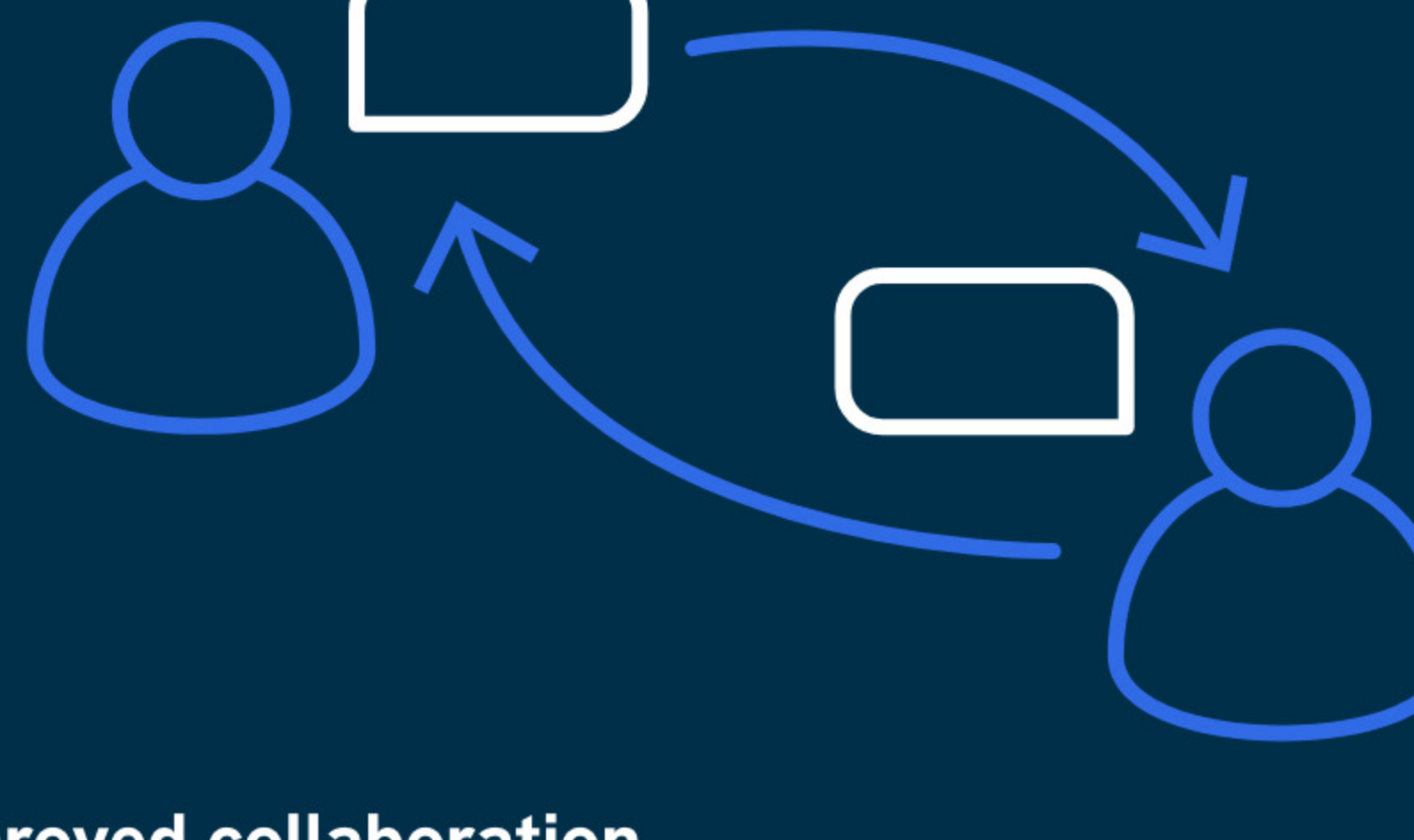
**Increased developer autonomy**  
Application developers can self-serve infrastructure and tools, reducing dependency on operations teams.



**Enhanced productivity**  
Simplifies the development process, allowing application developers to focus on coding rather than managing environments.



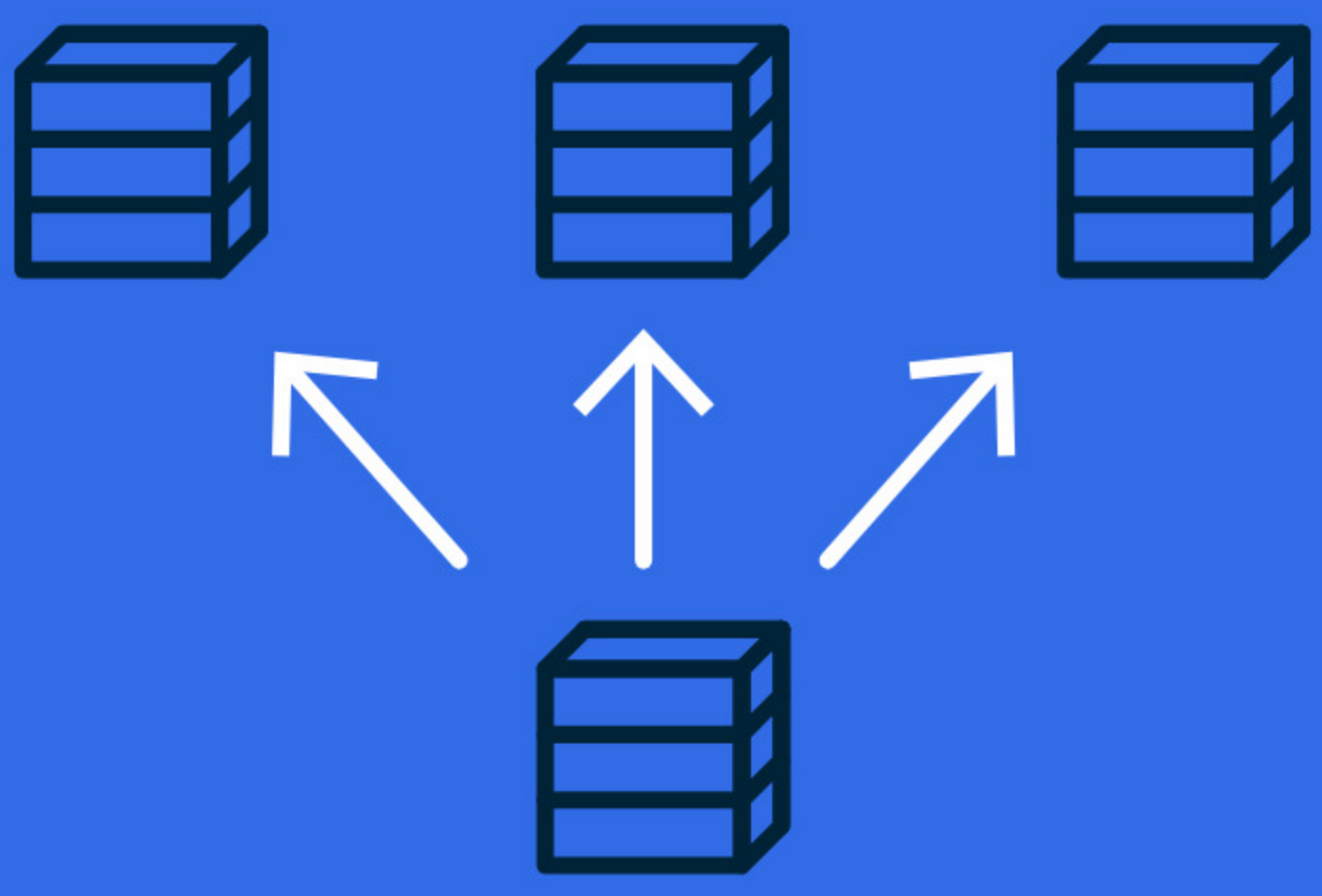
**Automated streamlined operations**  
Standardises workflows and automates repetitive tasks, reducing errors and improving efficiency.




**Improved collaboration**  
Clearly defines roles between development, platform engineering and operations teams, fostering better teamwork.



**Ensures compliance**  
Maintains governance and security policies across environments, reducing risk.



**Environment replication**  
Promotes uniformity in deployment, reducing discrepancies between development, testing, and production.



**Faster software delivery**  
Accelerates release cycles by reducing infrastructure bottlenecks.



**Cost savings**  
Optimises resource usage and infrastructure management, lowering operational costs.

## Key Considerations

Gartner's [Software Engineering Leader's Guide](#) to Improving Developer Experience states the importance of 3 key considerations:

- 1. Improve application developer experience by building internal developer platforms (IDPs) to reduce cognitive load, developer toil and repetitive manual work.*
- 2. Platforms don't enforce a specific toolset or approach - it is about making it easy for developers to build and deliver software while not abstracting away useful and differentiated capabilities of the underlying core services.*
- 3. Platform engineering teams treat Platforms as a Product (PaP) and design the platform to be consumed in a self-service manner.*

Source: [Summary Translation: A Software Engineering Leader's Guide to Improving Developer Experience \(gartner.com\)](#)

## Overcoming IDP Challenges

Challenges when architecting and building IDPs are unavoidable, as they are complex. That being said, there are key steps and technical principles you can follow to set yourself up for success.

LiveWyer has ten architectural principles for building a [Kubernetes as a Service IDP](#). You can combine them into the following proven approach:

- 1. POC:** Start with a small team and a small scope. This allows you to start with the basics, enabling the platform to scale across your Enterprise organisation when ready.
- 2. Healthy environment & healthy feedback:** Feedback from application developer teams is vital to platform success. If the application developer does not receive value, future adoption will not occur. Focus on delivering value to application developers, operations, and security teams.
- 3. Continuous improvement:** By combining a healthy environment between your platform and development teams with a Continuous Improvement practice, the application developers using the IDP will quickly see the value of saving time, allowing them to focus on their application code.
- 4. Clear responsibilities:** IDPs often fail when the teams' roles and responsibilities are not enforced correctly. The Platform is for everyone, and having application developers, Operations teams, and SRE Teams input and even own specific components is vital for them to feel like they are a part of its development.
- 5. Product management:** A competent Product Manager is necessary to enforce the roles and responsibilities of all parties, resist application developer requirements that are outside the scope of the platform engineering team, and, most importantly, set clear expectations of what the platform engineering team is working on and their current and future priorities.
- 6. Senior stakeholders:** For the Platform to grow, senior buy-in is imperative. However, steps 1 - 4 will help you show how the Platform delivers value and is ready to scale to further teams with the proper backing.

## Getting Started with Internal Developer Platforms

When we [design and build an Internal Developer Platform \(IDP\)](#), we begin by adopting a product mindset and treating the application developers as your customers.

Start small, identify the specific needs of a single application developer team, and then begin creating a minimal version of your platform that delivers immediate value. You can iteratively build this foundation by adding core building blocks over time, holding each building block against your core platform principles.

The journey should be incremental rather than a "big bang" approach: focus on solving critical problems first, using Cloud Native tools where possible.

Make small iterations and additions that solve your organisation's unique challenges. Listening to and acting on feedback from application developers ensures they are keen to adopt and champion the platform as it scales to additional application developer teams and new business areas.

Once you have multiple application developer teams championing your platform, the buy-in of Senior Stakeholders will soon follow.

### Thank you for reading

Do you need help Internal Developer Platforms? Get in touch and let's work together.

Contact Us

At LiveWyer Labs we innovate through research and development, see what else we've been [working on](#) lately.

If you want to stay up to date and be notified when we post new and exciting content, make sure to follow our [Linkedin](#) and [Medium](#).