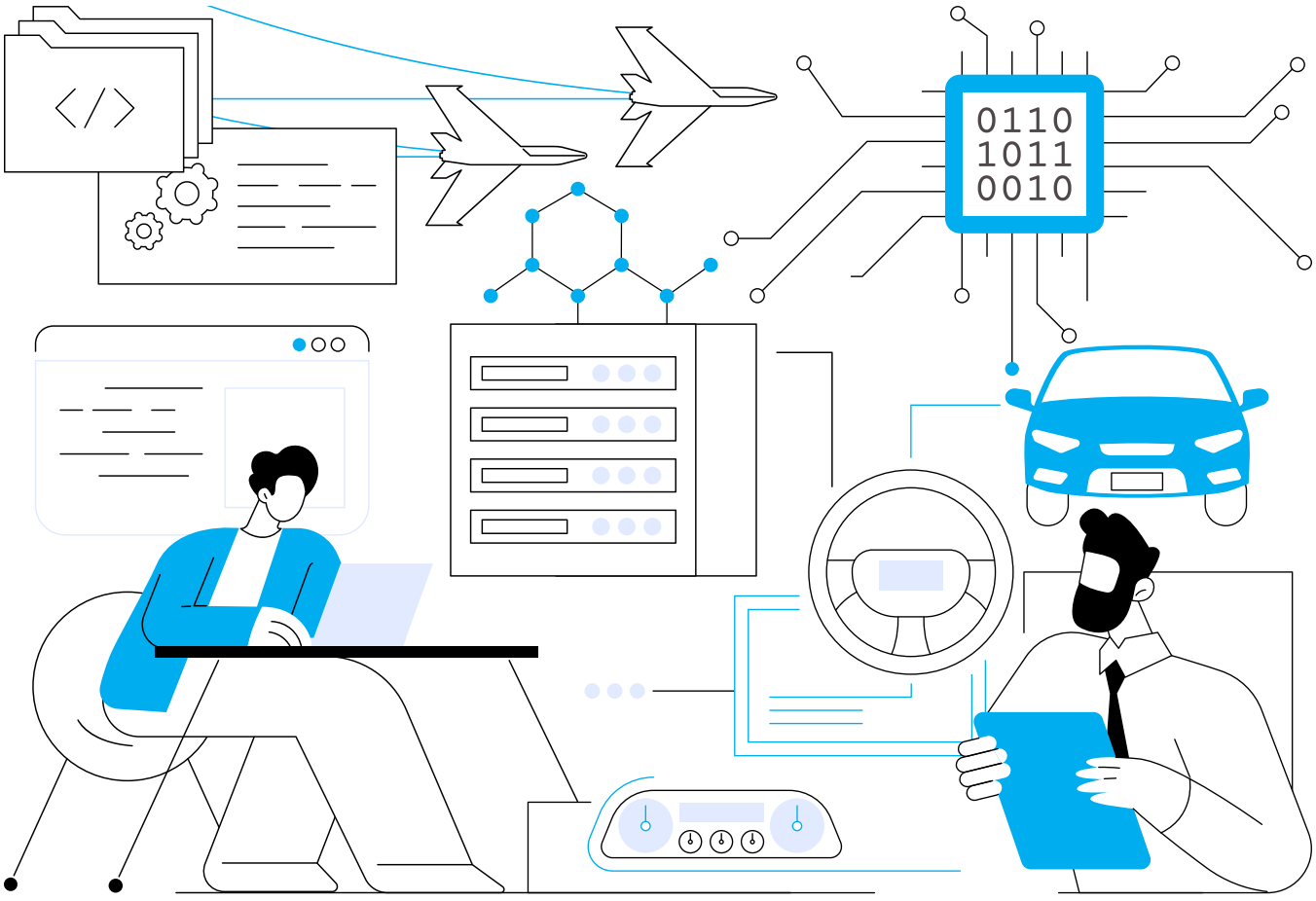
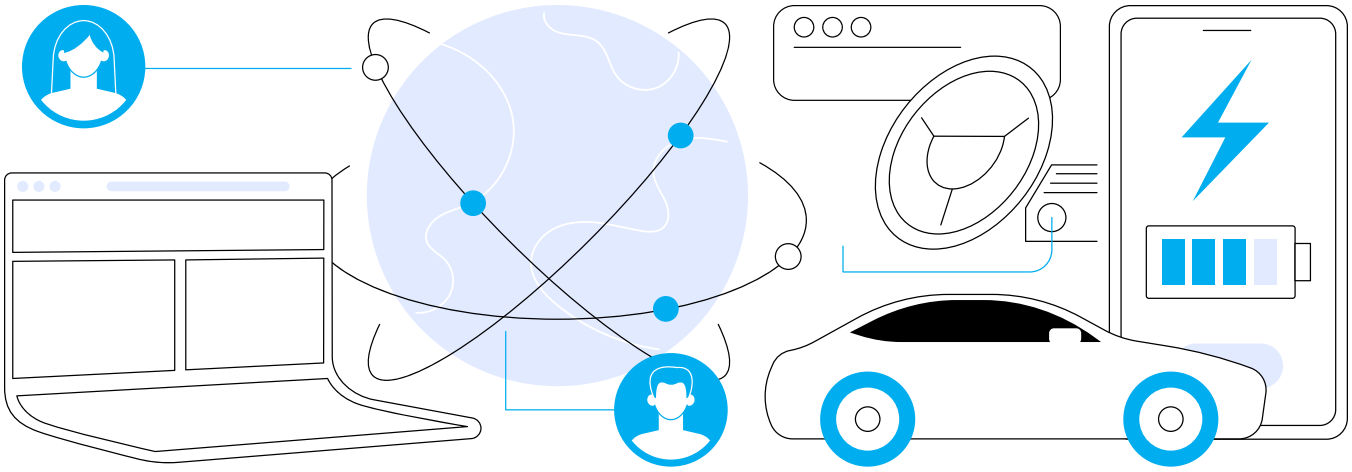
PERFORCE

# Supercharge Embedded Development with Modern Data Management

## Five reasons to ditch legacy data management tools for a future-proof solution





# Introduction

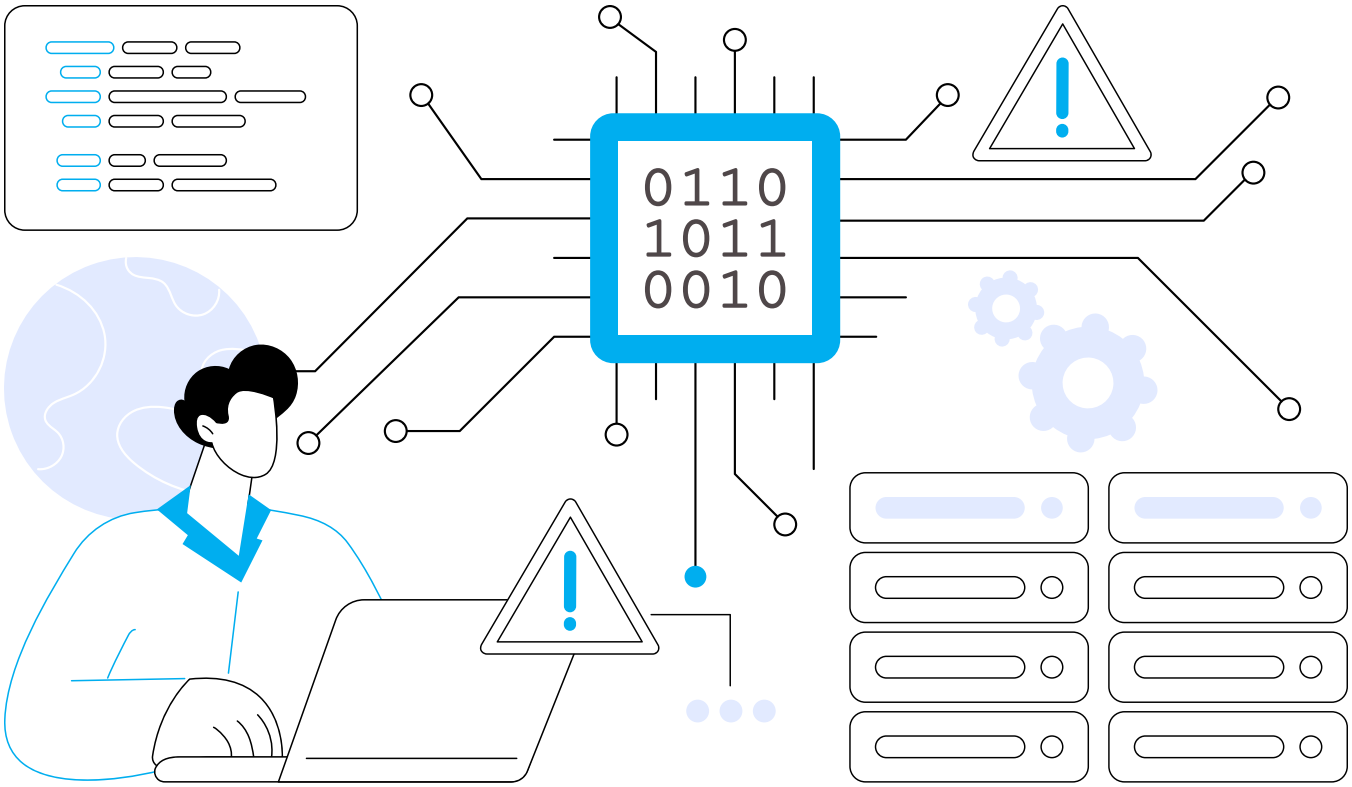
## For organizations developing complex embedded software, there's never been a better time – or a more urgent need – to ditch legacy data management tools in favor of a modern solution.

With ever-increasing file sizes, more dispersed global teams, and the exponential growth of embedded system complexity, teams are finding that yesterday's data management (DM) solutions cannot keep up with their needs.

From automotive OEMs and suppliers racing to crack the code that will enable more efficient Battery Management System (BMS) architectures for electric vehicles; to industrial manufacturing teams designing predictive maintenance systems to maximize production output; to medical device developers who must meet strict safety regulations while trying to beat their competition to market; and aerospace and defense companies developing mission-critical systems with stringent security requirements; the need for better, faster, more robust data management is felt across all high-performing embedded development teams.

However, it's crucial to ensure you select the right data management solution – one that you'll never outgrow. The solution you choose must meet the needs of a globally distributed workforce, offering limitless scale and enterprise-grade security.

In this eBook, we explore five reasons why now is the time to ditch your legacy DM tool, what to look for in a new, modern DM solution, and how switching to Perforce Helix Core's robust, federated, multisite data management solution can supercharge your embedded development at global scale.




# Legacy Tools Are Holding Back Your Innovation

Many legacy data management tools once stood out as cutting-edge technology, but lack of continuous development and insufficient scalability have made them ineffective against the challenges of today's embedded software development.

Two examples are Subversion (commonly abbreviated as SVN) and ClearCase (also known as IBM ClearCase or Rational ClearCase). SVN and ClearCase are both data management (DM) tools, also commonly known as version control, source control, or revision control tools.

Let's take a look at the main drawbacks of these legacy DM tools before we dive into five reasons to migrate to a modern solution.

## Subversion

SVN is an open-source, community-driven project which has seen minor and infrequent updates over the years. There is no set timetable for its next release.

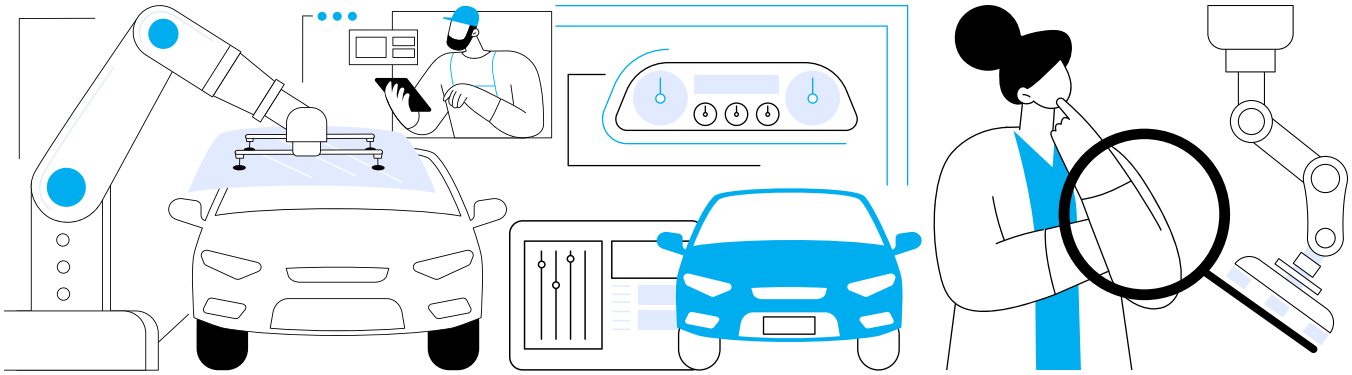
The most common issue teams have with SVN is that once your team grows to more than five or six people, merging branches becomes an extremely complex process. Because this process may take several hours (or even several days) to complete, teams using SVN merge branches less often. This infrequent merging leads to bugs and quality issues down the road, ultimately extending project timelines.

## ClearCase

Founded in 1990, ClearCase was designed in an era where projects consisted of only a few thousand files, yearly release cycles, and file sharing over a LAN. While embedded development has radically transformed since then, ClearCase has not. The tool has not seen any major innovation in decades.

The primary drawback of ClearCase is that it is a slow, high-maintenance tool that requires heavy scripting to do pretty much anything. Developers regularly complain that it is overly complicated to set up, difficult to use, and time-consuming to maintain. Often there will only be one or two people in the organization who know how their custom scripts work or how to change them. Like SVN, ClearCase does not handle branching or merging well, which makes it nearly impossible to scale projects or teams.

Organizations that wish to lead their industry and drive innovation through next-gen embedded software cannot afford to be hampered by infrastructure scalability issues – not to mention security risks.



# 5 Reasons to Switch from Subversion and ClearCase

1 2 3 4 5

## Lack of Scalability

Embedded development teams work with a staggering number of files. And these files grow larger and more complex with each passing year. You need a data management solution that scales to fit the demands of your developers and keeps them working efficiently today and far into the future – not a system that lags, crashes, or otherwise struggles to keep up.

Subversion is a single-server system. This limits performance and scalability. As a result, SVN often can't handle large repositories and file sizes.

ClearCase also misses the mark when it comes to scalability, due to limited file views, additional installations, and lag times. ClearCase's Versioned Object Base (VOB) requires using either a dynamic view or a snapshot view to access your data.

Dynamic views require installation of the MultiVersion File System (MVFS). This means you must make modifications to the kernel of the operating system. Snapshot views do not depend on the new file system but are notoriously slow. ClearCase dynamic views also require a continuous network connection to the server(s), slowing down network operations.

If your legacy DM is struggling with scalability issues now, this problem will only get worse the longer you wait to modernize your infrastructure.

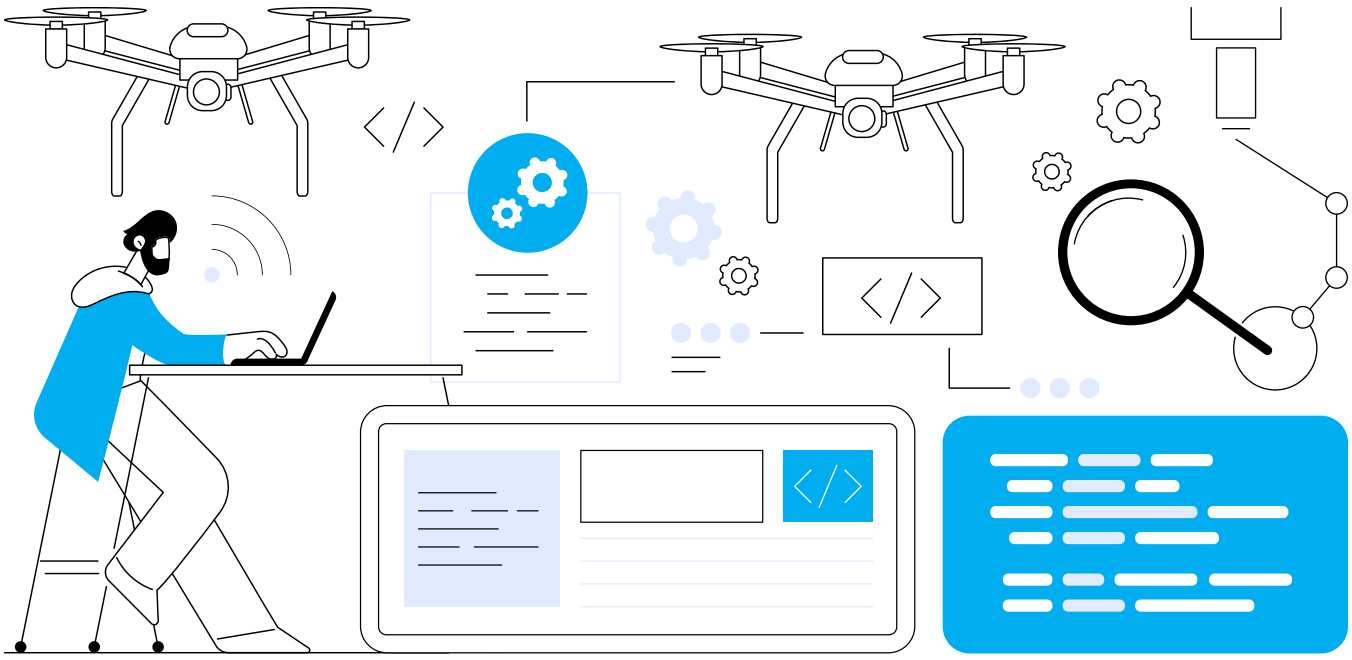
1 2 3 4 5

## Unsophisticated, Error-Prone Branching

Clear, intuitive branching is a key part of making sure changes are communicated across teams and file versions. Without modern branching capabilities, teams risk introducing errors into their workflows. As a result, they may fail crucial tests or require duplicate work. Reliable branching and merging capabilities are also essential to scale projects and teams and to encourage creativity and innovation.

Subversion users identify branches by naming conventions. If changes need to be applied across several development branches, merging often leads to errors. As mentioned above, slow, overly complex branching and merging is one of the core issues users have with Subversion. As a result of these branching and merging challenges, it's almost impossible for teams using Subversion to do parallel development and scale in any meaningful way.

ClearCase includes basic branching and merging capabilities using either base ClearCase or UCM (Unified Change Management). With base ClearCase, there is no true relationship between ClearCase branches and the main code line. The development process must be either written down or automated using external scripts – a time-consuming, outdated, error-prone process. Without implementing automation scripts, it's very difficult to know where to propagate changes from branch to branch.



1 2 3 4 5

## Risks to Your Data Security

Protecting your data and intellectual property (IP) is more important than ever in today's unpredictable geopolitical environment and constantly shifting security landscape. It's essential to invest in the right data management solution, rather than one that leaves your critical files and IP vulnerable to attack.

Subversion offers no out-of-the-box security. Instead, the tool provides only basic full-repo and path-based access control. And once it's in operation, it's labor-intensive and difficult to manage.

ClearCase lacks granular security permissions. This poses a risk, especially as knowing exactly who is viewing data and accessing files, as well as managing the geographic locations they're accessing files from, becomes increasingly vital.

ClearCase's security follows a similar model to a UNIX filesystem. Permissions can be applied at the VOB, directory, or file-level. However, since permissions are dependent on the operating system, it can be nearly impossible to synchronize permissions across different platforms (Linux, Windows, Mac) and network domains. This is a significant obstacle to managing file access.

1 2 3 4 5

## Lack of Ongoing Development or Support

Simply put, using Subversion or ClearCase means using an unsupported tool: These products are no longer being developed. When it comes to managing your key data and IP – your most valuable assets – do you really want to trust a tool that hasn't been actively developed in several years or, in the case of ClearCase, decades? What recourse do you have if your data management tool breaks, bringing your development to a halt?

This risk increases with each year that you continue to use these aging tools, as the underlying technology falls further and further behind.

Lack of ongoing support also means your legacy tool can't integrate into any modern tooling ecosystem. Neither SVN or ClearCase supports today's popular developer tools out of the box. And building home-grown integrations requires heavy scripting and third-party plug-ins, bringing additional risk and further complication to your tool suite.

1 2 3 4 5

## High Upfront or Maintenance Costs

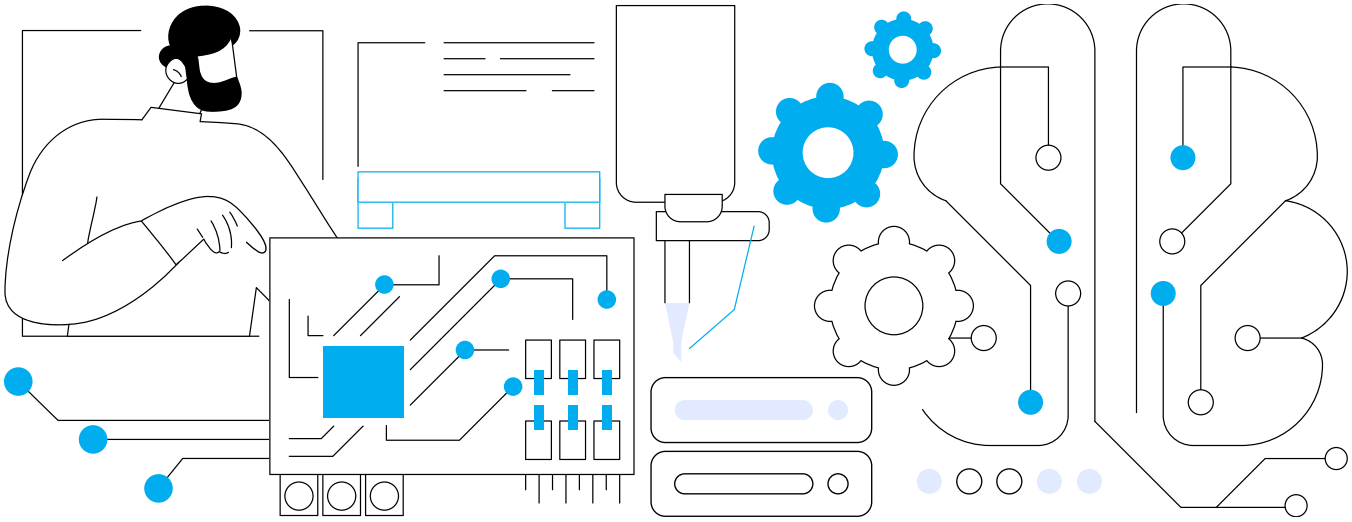
With legacy data management, you may be paying more for tools that perform less effectively than modern DM solutions. These legacy DM solutions also have fewer features and less support.

Subversion is "free" – a major factor that plays into organizations' decision to use the tool. Despite SVN standing out as a seemingly budget-friendly solution, it's essential to look beyond the surface and identify hidden associated costs.

Although you don't have to pay for licenses to use Subversion, you can quickly drain your company resources trying to make it work. Time is money, and wasting valuable developer and IT hours trying to make a sub-standard tool meet your needs is not something most organizations can afford.

Between code management issues, suboptimal workflows, and poor scalability, many Subversion issues can hamper productivity. In many instances, the net loss is more expensive than licensing a proprietary data management solution.

ClearCase users can expect costly licensing and maintenance. This is in part due to the tool's active-user floating license model, which is expensive to purchase and upkeep. You'll need extra tooling for global development, which adds to the overall price tag. The lack of integration and missed automation opportunities also contribute to driving up costs, as does administrative support. You typically need one administrator for every 25–50 users.



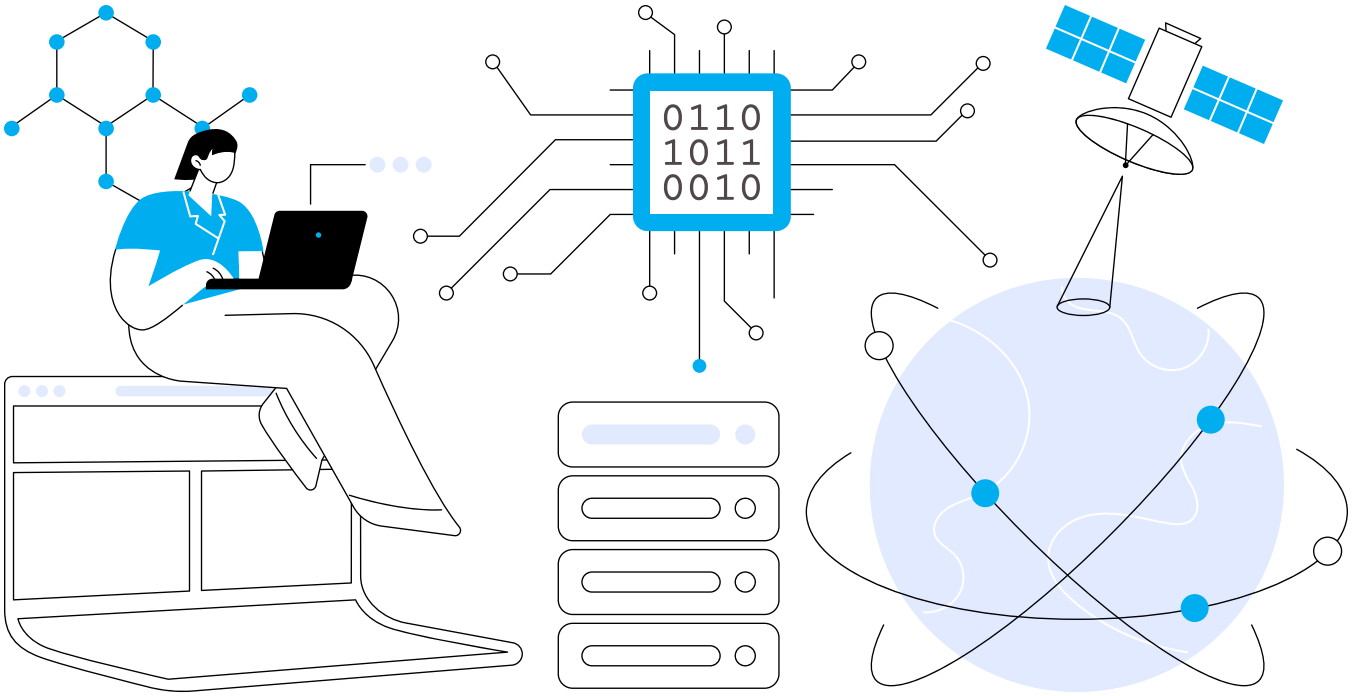
# Why Now is the Time to Migrate to a Modern DM Solution

Across every industry, organizations are seeing more intense competition to get new features and products to market, increasingly tighter deadlines, and demands for more sophisticated, complex development at an ever-accelerating pace. These factors raise the stakes and underscore the urgency for teams to drop the legacy tools that are hampering their ability to innovate.

Oftentimes, teams continue using legacy data management tools simply because it's what they are used to. In the fast-paced world of embedded development, learning a new tool and getting large teams up to speed can seem overwhelming. Stakeholders who could (and should!) advocate for more modern technology might worry that introducing a new solution could impact development velocity.

However, continuing to use outdated tools will have steeper, hidden costs – financially, from a security perspective, and potentially by losing the competitive advantage that comes with being first to market.

Consider this: if you're still battling DM infrastructure issues while your competition is cruising ahead with a high-speed, infinitely scalable DM under its hood, where will you be a few years from now?




# High-Performance Data Management to Accelerate Embedded Development at Global Scale

There's one more reason why now is the best time to ditch your legacy DM tool: Migrating to Perforce Helix Core now means you'll start reaping the benefits of a modern, limitless, high-performance data management solution that much faster.

Helix Core gives embedded development teams the foundation they need to accelerate innovation. It enables global teams to collaborate better, and versions in the background so you can focus on your work, not your tools.

Helix Core is trusted by numerous Fortune 500 companies, including top automotive, medical device, and aerospace and defense companies. It's the standard in data management for embedded systems leaders like Nvidia, Samsung, McKesson, BioMerieux, Honda, Porsche, and Boeing.

That's because Helix Core is the only data management solution that scales endlessly without sacrificing performance. It facilitates even the most complex development by allowing distributed teams to quickly and securely access the files they need, improving development velocity and team collaboration.

Plus, Helix Core provides endless flexibility and superior branching and integration capabilities so you can develop your way, while boosting productivity and establishing a single source of truth across your organization.

| How do teams work? | SVN | ClearCase | Helix Core |
|---|---|---|---|
| Centralized (check files in and out from a central repo) | ✓ | ✓ | ✓ |
| Distributed (push and pull between repos) | ✗ | ✗ | ✓ Helix Core offers DVCS functionality, giving your teams added flexibility. |

| Does it scale to manage…? | SVN | ClearCase | Helix Core |
|---|---|---|---|
| Large repos | ✗ Performance suffers at scale, delaying teams. | ✗ ClearCase slows down as the codebase and history grows. | ✓ Helix Core is known for its high performance at scale, no matter how big your projects and repos grow. |
| Large files (including binaries) | ✓ Depends on the file type. | ✗ | ✓ Helix Core stores all your digital assets — including large binary files — and more. |

| How does it protect your assets? | SVN | ClearCase | Helix Core |
|---|---|---|---|
| Secures valuable IP | ✗ Nothing available out-of-the-box. | ✓ ClearCase offers a file/directory level permission structure, similar to a filesystem. It's complex to administer. | ✓ Helix Core protects IP down to the individual file and supports fine-grained security rules. |

| How does it manage files and branching? | SVN | ClearCase | Helix Core |
|---|---|---|---|
| File locking | ✓ | ✓ | ✓ Helix Core takes file locking a step further offering exclusive checkouts to avoid duplicating efforts. |
| Visualization into how code flows | ✗ | ✗ | ✓ Perforce Streams helps teams see how code should flow. Developers can quickly tell what branches have changes available. |
| Tracks relationships between branches | ✗ | ✗ | ✓ Streams guides developers as they branch and merge. |
| Built-in code review | ✗ | ✗ Requires 3rd party options. | ✓ Helix Swarm is the code review solution for Helix Core (and it's free). |
| Offers integrations with tools teams use | ✗ | ✗ ClearCase is no longer actively developed. | ✓ Helix Core has plugins and integrations to the tools teams already use. Plus you can use Helix Sync to version any digital asset. |
| Works with Git | ✗ | ✗ | ✓ Helix4Git allows you to solve development challenges that can happen as Git teams grow, and it offers faster builds. |

| How does it support teams as they scale? | SVN | ClearCase | Helix Core |
|---|---|---|---|
| Can be hosted in the cloud | ✓ Learn more about SVN hosting. | ✗ | ✓ Helix Core can be hosted in your preferred cloud provider. Get Perforce on the cloud. |
| Global replication | ✗ Only works if you install costly 3rd party add-ons. | ✓ ClearCase MultiSite is required to replicatte, but also expensive to purchase and maintain. | ✓ Perforce Distributed Architecture's advanced replication technology delivers files around the globe, fast. |
| Offers world-class support | ✗ | ✗ | ✓ Support is available around the clock to ensure your teams always have the access they need. |

While legacy tools introduce user pain points in key areas like scalability, branching, and data security, Helix Core excels in these areas.

## Scalability

Helix Core performs at any scale and grows with your team as your needs and projects evolve. It serves as a single repository for everything – including binaries, IPs, and even Git source code.

Organizations can manage and version all embedded software development files, plus scale infrastructure to support globally distributed teams — fast.

It's the only DM solution you'll never outgrow, as it was designed to handle:

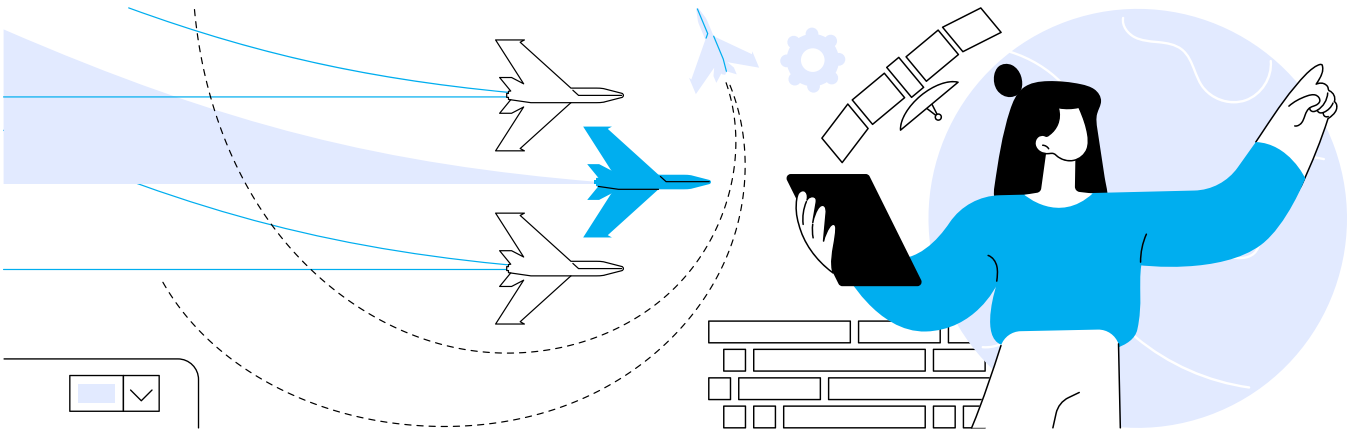
**10s of thousands of users.**

**10s of millions of daily transactions.**

**Petabytes of data.**

**Global, multi-site topologies.**

Helix Core securely supports both distributed teams and remote contributors. With a powerful caching and replication architecture that is unmatched in the industry, access for engineers and designers is seamless and nearly instantaneous, no matter where they are in the world.



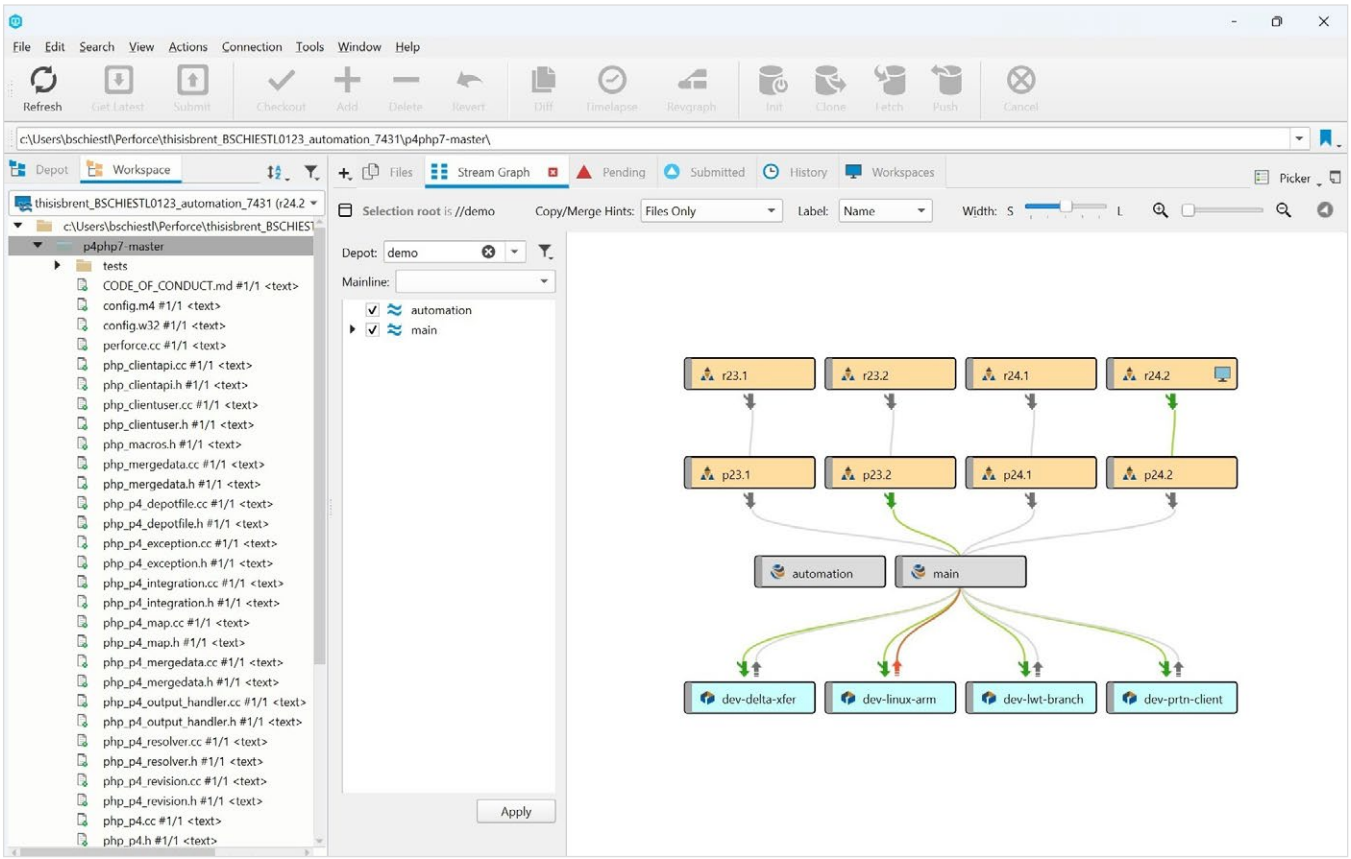
# Streamlined Branching & Merging

Nearly all modern data management solutions have branching capabilities, but overly complicated branching can restrict your developers and sow confusion across your teams. For most teams, their branching structure is either hidden in scripts that only a handful of admins have access to, or it might be drawn out on a whiteboard in an office somewhere and instantly out of date the moment it's created.

That's why Helix Core simplifies branching and merging with Perforce Streams.

Built into Helix Core, Streams guides teams through branching and merging, with no external tools or scripts required. It allows developers to visualize the branching structure, so they can easily see which code and changes are moving into which branch – even if a project has a lot of dependencies, files, binaries, contributors, and changes.

And it's always current because it's a "living," dynamic diagram of your development flow. Everyone can see what work has been completed, and the changes that still need to be integrated. This makes branching and merging easier to manage, reduces errors, and increases developer productivity.



## Enterprise-Grade Data Security

Helix Core protects your most valuable data and intellectual property. With its fine-grained permissions, you get granular access control – down to the individual file level, including based on the IP address the user is trying to access the file content from – giving team members, contractors, and third-party teams access to only the files they need. Plus, you can review your full audit history to see what was accessed, what was changed, when, and by whom.

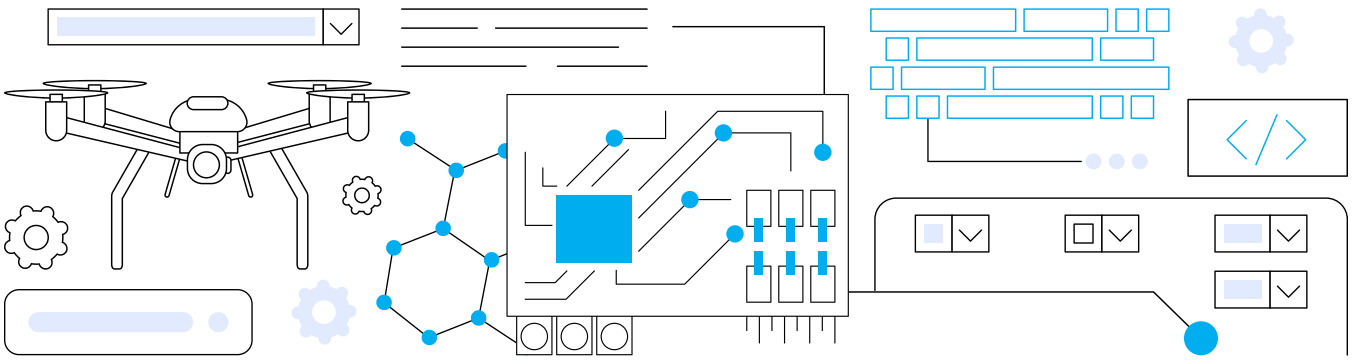
To further enhance security with Helix Core, organizations can use enterprise identity and access management (IAM) infrastructure with SSO (SAML or OIDC) or AD/LDAP.

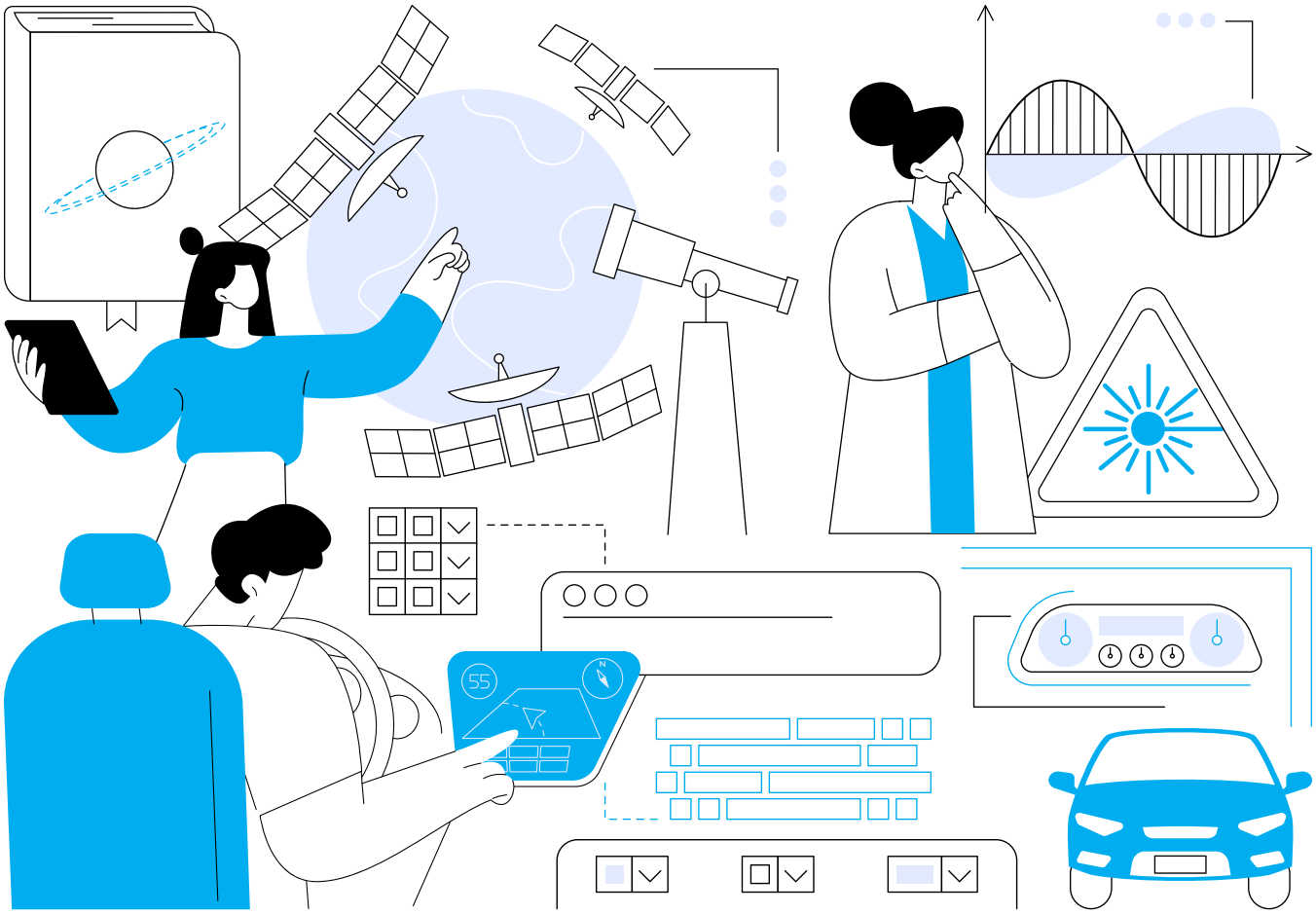
## World-Class Support and Development

As the industry standard for data management and version control, Helix Core benefits from continuous development, unparalleled support, and a relentless focus on helping organizations tackle their most complex development challenges. Perforce is constantly improving and adding new features and functionality to Helix Core to ensure it exceeds the requirements of the world's most high-performing development teams.

Helix Core is the only data management solution you'll never outgrow. And it easily integrates with modern development ecosystems, so everyone in your organization can use their preferred tools, while your organization benefits from a single source of truth across teams, limitless scalability, enhanced security, and superior performance.

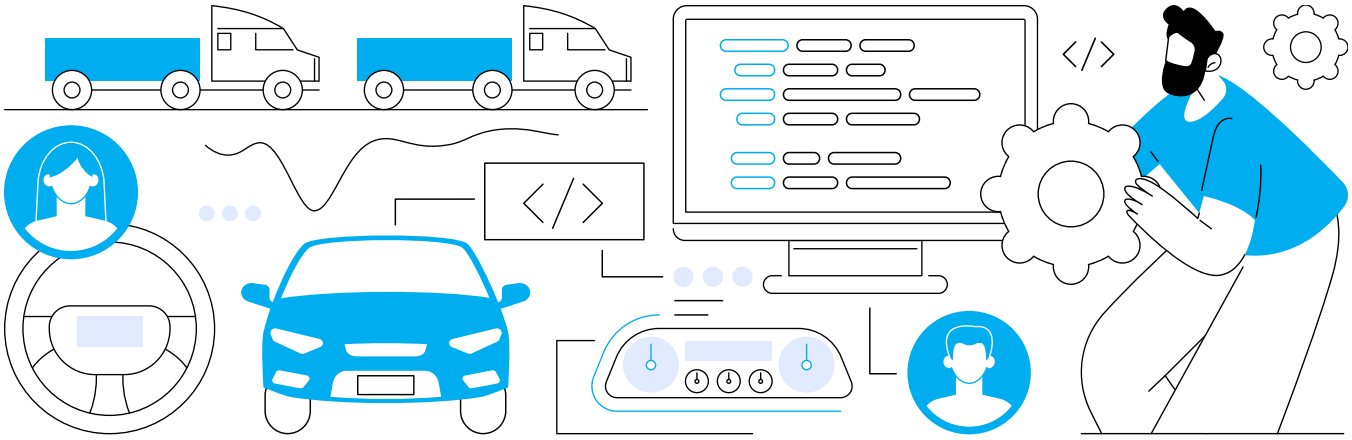
Perforce's global support team is ready to help you every step of the way. You can leverage our proven processes for helping teams like yours migrate from existing tools (such as SVN or ClearCase) to Helix Core. With deep knowledge and extensive experience in the data management space, our expert consultants can help with complex projects and provide guidance throughout the migration process to ensure your success – today and well into the future.





# Teams Who Have Switched from Legacy Tools

Enterprises across all industries have made the leap from legacy data management tools to Helix Core – resulting in heightened productivity, faster go-to-market, better collaboration, and improved security. Consider these two success stories to visualize how Helix Core can accelerate your embedded development.



Transurban

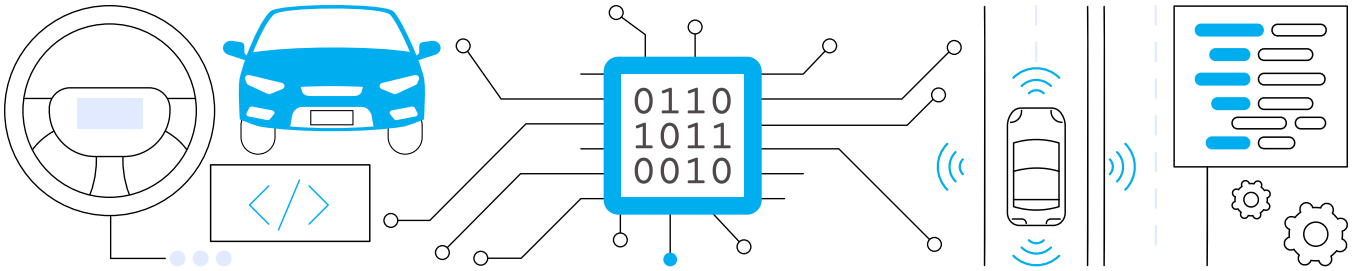
Transurban partners with governments in Australia and the United States to deliver embedded IoT technologies to toll road networks, such as smart-road sensor systems.

Transurban's development environment involves large deployments with complex files, multiple environments, technologies, components, and contributors. Before streamlining their deployment process using Helix Core, "all our builds were incremental and we had an ever-growing list of deployables that needed packages," said Matthew de Vanny, Technical Environment Specialist at Transurban.

With Helix Core, Transurban can release larger deployments faster and more reliably. Deployments that previously took eight hours now only take two.

Transurban witnessed these benefits firsthand when they brought the systems from their Queensland office on board. "We completed that in a weekend," said de Vanny. "With our previous methodologies, that wouldn't have been deployable in the given development window."

> "The performance is so fast that we take it for granted. Merges and integrations run like greased lightning. Even when we're doing 4,000-6,000 objects in an integration, it runs in moments."
>
> Matthew de Vanny, Technical Environment Specialist at Transurban




u-blox is a global leader in positioning modules and chips for the automotive, industrial, healthcare, and other IoT consumer markets.

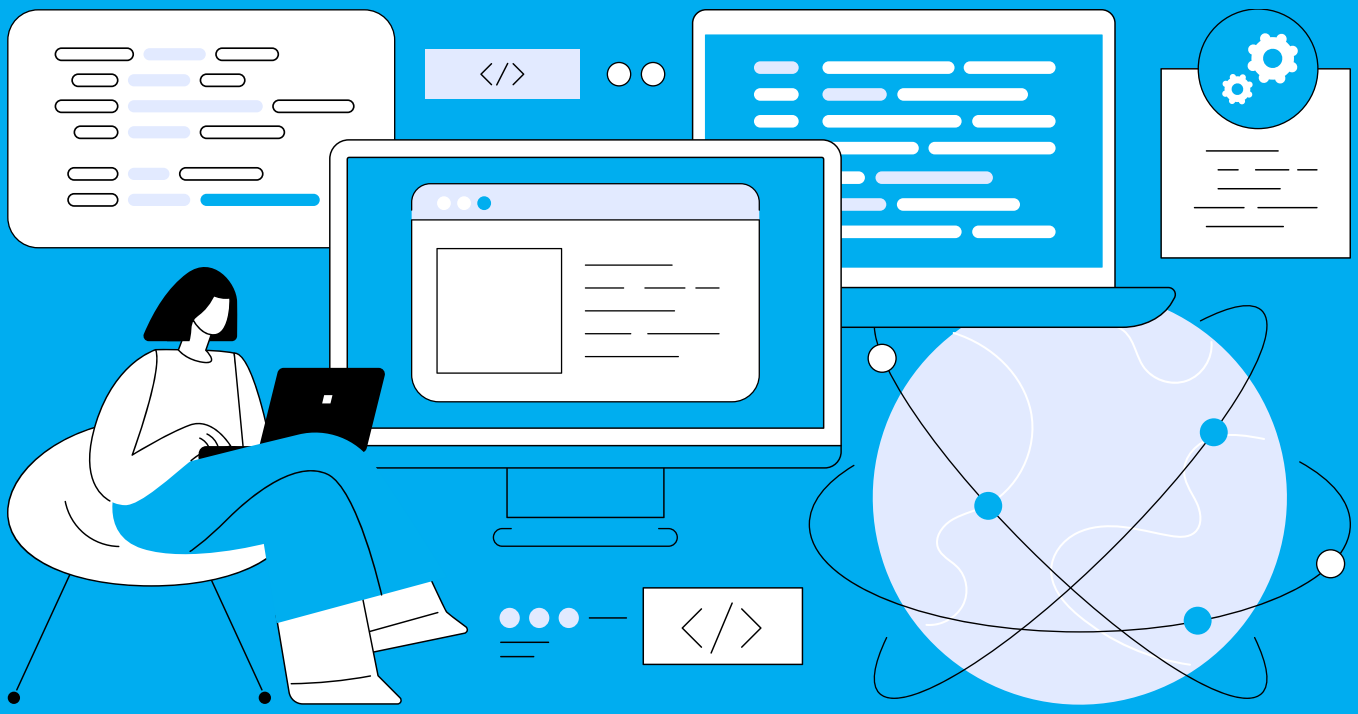
Using a legacy data management tool was hampering the company's ambitious project goals. u-blox engineers working across seven countries were experiencing performance lags with their outdated system.

By pivoting to Helix Core, u-blox was able to adopt a new development process to serve as a key project foundation – component-based development (CBD). CBD is important for u-blox because it allows their distributed developers to divide and conquer, said Stephan Uyttebroeck, Principal Software Engineer at u-blox.

Since they can store all their digital assets in Helix Core — regardless of file type or size — developers can quickly access components and assemble variants. Thanks to these capabilities, Helix Core allows for resource sharing across u-blox's development sites.

Helix Core's many integration capabilities, such as support for Jira and Jenkins, and SDKs for popular scripting languages, allow u-blox to integrate their data management with internally developed tools.

> "One team wanted to carry on using their existing tool, but when they saw that teams using Helix Core were working much faster, they made the move to Helix Core because of the stability and performance."
>
> Stephan Uyttebroeck, Principal Software Engineer, u-blox



# Supercharge Your Development with Helix Core

Take the jump from legacy tools and discover firsthand how Helix Core's modern data management capabilities can supercharge embedded software development and accelerate innovation. Talk to our data management experts today to learn more.

**TALK TO AN EXPERT**