**What Is Copado?**

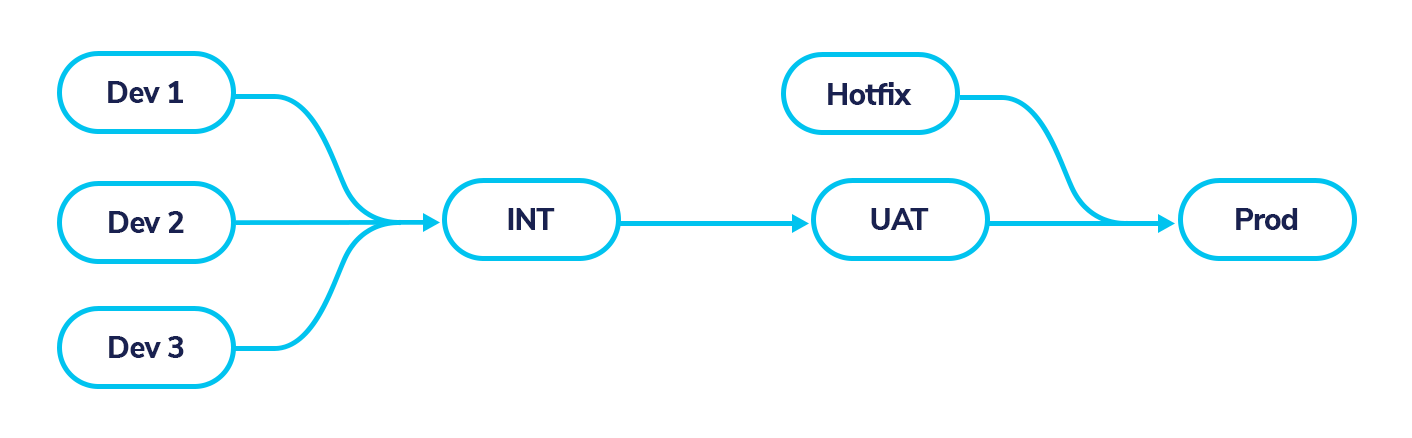
Copado is a scalable Salesforce-native application that you can download and install in an org. It provides you with a complete system to plan, track and carry out your DevOps activities. It also adds value in every stage of the Salesforce DevOps process.

Here are some of the exciting features:

* **Agile Implementation**: Copado helps you leverage Agile methodologies to compose your releases and plan your sprints.
* **Quality Check**: You can use the inbuilt set of quality and compliance features to implement quality checks right from the start of the development lifecycle.
* **Integration with Git**: You can use Copado to implement seamless and effortless integration with Git.
* **Easily Customisable**: You can customize Copado effortlessly as it is a native Salesforce app. Here are some of the actions you can perform on Copado to personalize it for your specific requirements:
  + Add custom fields.
  + Add validation rules.
  + Implement approval processes.
  + Modify fields and sections to customize the page layouts to enhance the user experience.

**How Copado Addresses the Development Lifecycle Challenges**

**Pipeline Provides a High-Level Overview**



With pipelines, Copado comes to set order out of the chaos. A pipeline is a simple, graphical representation of the path that changes will follow through the release process.

Copado uses a user story centric model. A **user story** is an element used in agile software development that contains the description of a requirement from an end-user perspective and relevant information about the changes to be implemented by developers. In Copado, it also holds the metadata related to those changes.

User stories will typically start in the lowest environments and get promoted to the next environments all the way until production, following the path described in the pipeline. The **Pipeline** enables you to orchestrate your user story deployments and verify that your quality standards are met.

In order to build your continuous delivery pipeline, you need to create pipeline connections and connect to your Salesforce environments. Copado uses for this OAuth 2.

Copado will provide a set of templates to help you choose a starting point for your pipeline. If none of the templates provided works for you, you can build your own template to suit your requirements as pipelines are completely flexible.

Defining a pipeline will help you later enforce best practices through the release process. All the changes in your pipeline will go through each stage, guaranteeing that they are properly integrated and tested so that no environment is skipped.



**Integration with Version Control for Everyone**

Forget about change sets! Copado incorporates a **Version Control System** - a single source of truth - into the development lifecycle. With Git, all changes are tracked.

Then, how is the flow users follow to move their changes while working with Git integration?

1. After they have configured the components in a dev environment, they will just need to access the **User Story** record and through a user-friendly interface, select the components to commit. For ease of use, they can filter within the list of available components using keywords like metadata type or name. Users are no longer overwhelmed having to scroll across a ton of pages!
2. The commit process takes place in the Copado UI, and Copado takes care of the Git operation, automatically creating a feature branch in the Git repository for each user story, where only the components committed will be included.
3. As users decide to release their changes to a higher environment, they create a promotion and pick the user stories they want to deploy. Once this promotion is stable for the deployment to be executed, Copado builds the package XML for you and deploys the changes to the next environment of the pipeline. If the deployment is successful, the last step is to merge the changes into the destination branch.

With all this, we ensure that all the changes are tracked in user stories, and whatever you have in an environment at Salesforce level is the same you have in that environment branch at Git level.

Committing Nested Components: Parallel releases and work in progress could complicate your releases, even more when hotfixes are required.

How do you easily “cherry-pick” your changes on a profile or just that urgent new field on the huge account object? Copado gives you the flexibility to isolate your changes and commit them from a user story.

In the metadata grid, you will be able to select nested components such as custom fields, on an object, changes in validation rules, or list views.

Copado will retrieve, commit and deploy only the selected nested components, giving you the possibility to seamlessly deploy from development to production without carrying work in progress.

Committing Destructive Changes: The change process is not always about building. It is important to have a clean and healthy organization.

In order to achieve that, you may sometimes need to delete some of the changes. With Copado’s **Destructive Changes Git** operation, you can easily remove any unwanted changes from your orgs.

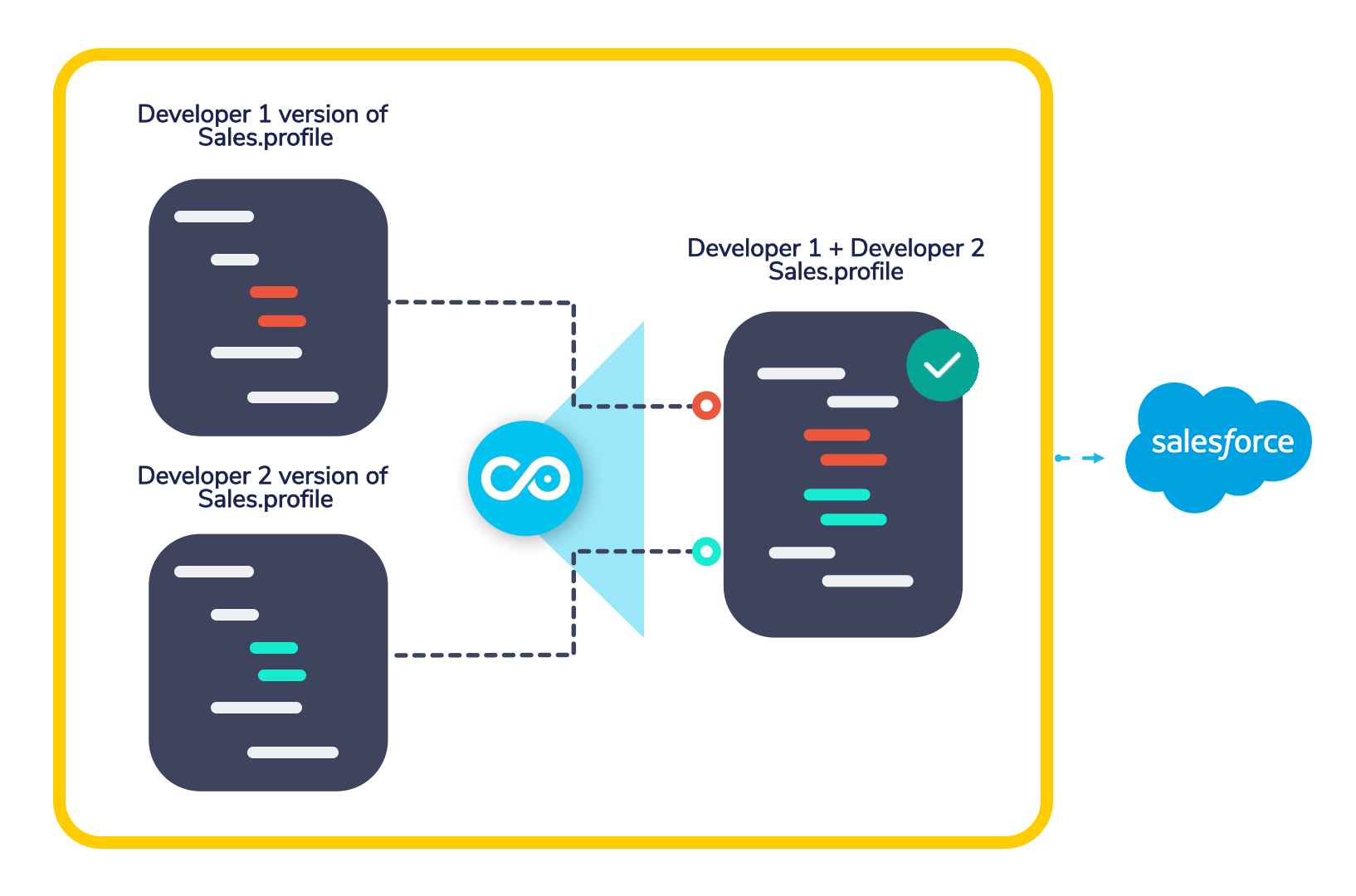
Think of deleting an unused field in your Salesforce organization. Is the main object the only component affected? The truth is behind the scenes, Salesforce will remove the field from all layouts, but also from profiles and permission sets. This wouldn’t be an issue if you were not leveraging version control. As your repository is the source of truth, you need to make sure all the changes in your Salesforce environment are also committed to Git.

Whenever you commit a deletion, Copado also makes sure all the other components that are modified as a result of the main deletion are cleaned up. This action happens automatically, so you only need to select the main component that has been deleted.

But wait, there’s still a piece missing. What happens with the Git conflicts that may arise? For that, Copado has a conflict resolution engine that auto resolves conflicts in components such as Profiles, Permission Sets and CustomObjects.

Copado’s deep understanding of Salesforce metadata allows it to analyze the conflicting files and to try to resolve the conflicts by merging the contributions of all into a new version of the component.

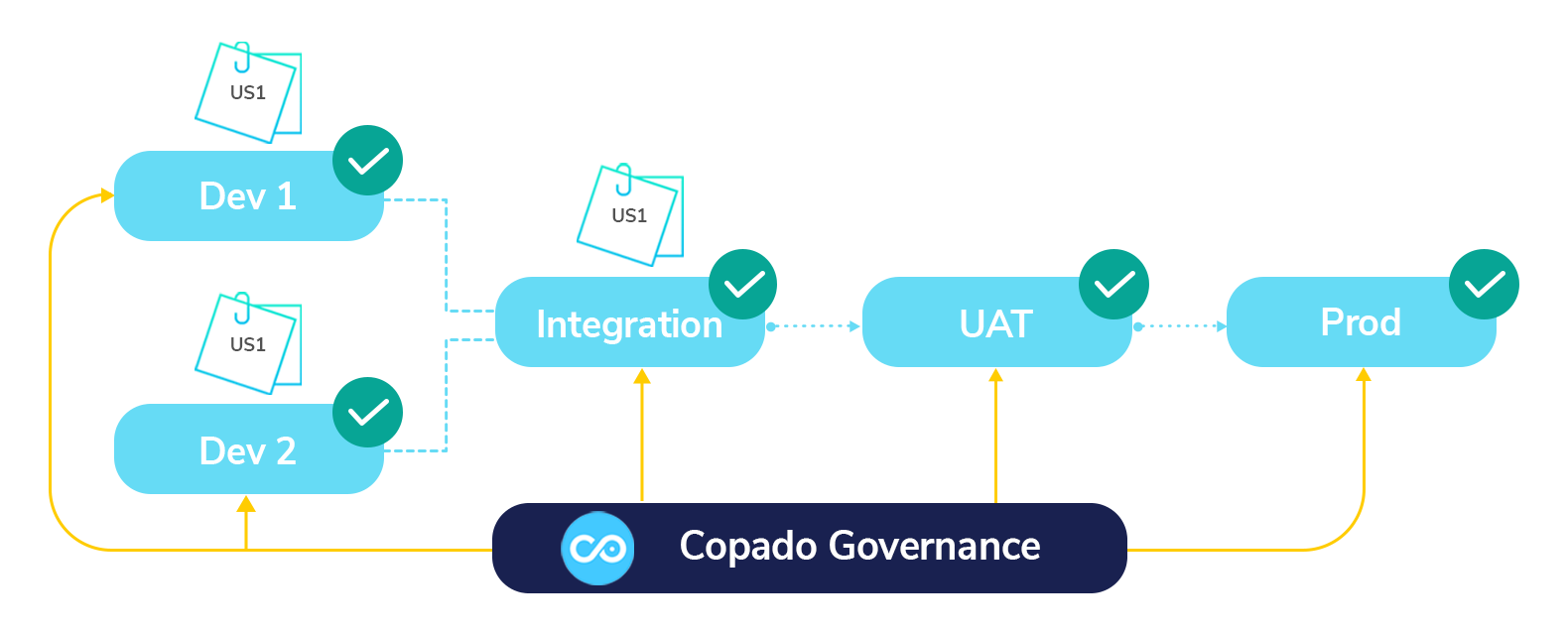
Nonetheless, when a Git merge is not possible, Copado offers several options to handle these conflicts, like **Online Conflict Resolution**, where users can visualize and resolve conflicts directly from the interface.



Copado will store this resolution strategy in your instance and will use it again in the future if the same conflict occurs.



**An Alternative to Sandbox Refresh**



Copado uses promotions to move changes forward and back-promotions to keep sandboxes in sync after deploying changes to higher environments. As you’ve seen, a promotion is basically a container of user stories.

Companies can minimize their differences across environments by just moving the changes they want, without affecting the work in progress of the sandboxes. This helps ensure that your team never suffers from downtime spent integrating or restoring their work manually.

From the pipeline, you can view at a glance all the environments that a user story has been deployed to, so you can save the overhead of checking whether the latest changes have been moved to every sandbox.



**Traceability**

With Git, developers can see all changes over time, and mechanisms like pull request verification allow them to let their peers know about their change, so that they can review them and give feedback before they get deployed.

Lastly, as each change is tied to a user story, you can establish a direct relationship between metadata and functional requirements.

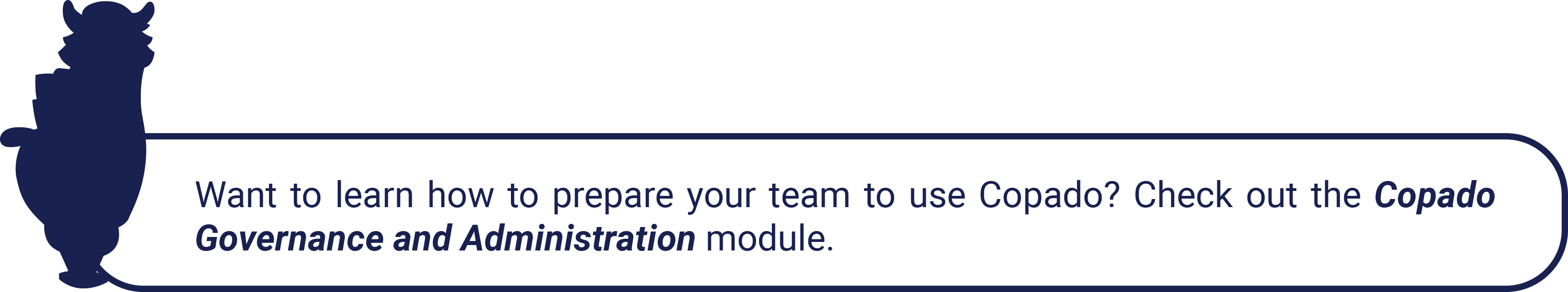
**Effective Collaboration**

Collaboration is key in DevOps. It is no secret that in order to achieve true continuous delivery, version control and agile development are essential.

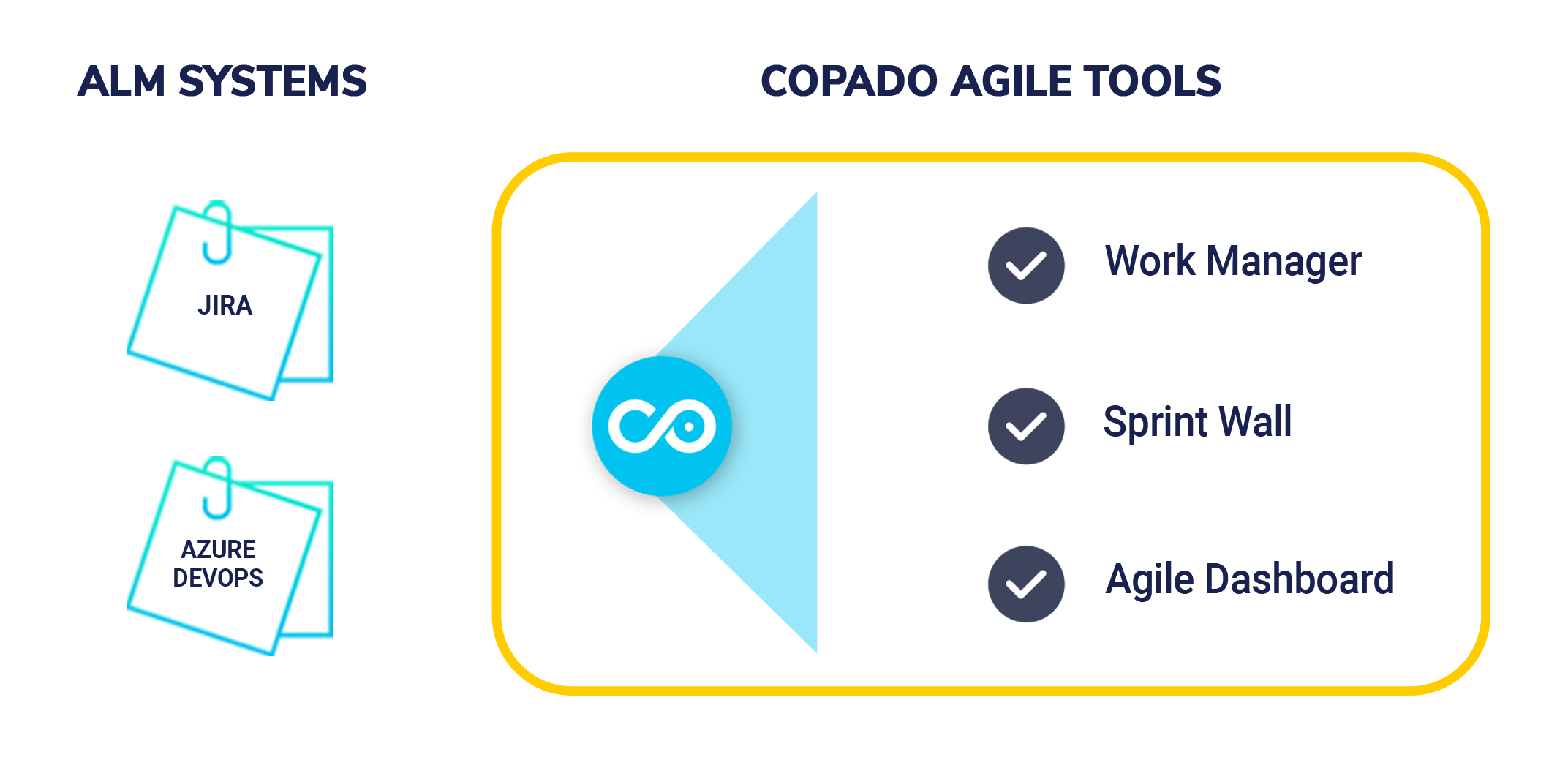
DevOps processes include stakeholders with different technical skills. One of the questions Copado helps you answer is how to introduce all stakeholders to version control so that they can all follow the same process.

First, Copado has an easy-to-use interface that facilitates technical users and users with a little knowledge of Git sharing a common release process.

Also, with the **Overlap Awareness** feature, if any potential conflict is found while multiple developers are making changes in shared components, Copado will alert them so that they can be aware of the risk, track the conflict status and open a pull request in order to work together towards a resolution.



**Work Based on Agile Methodology**



Copado supports agile integrations to external application lifecycle management (ALM) systems to retrieve your user stories into Copado, and if you’re using Jira or Azure DevOps the synchronization is bidirectional.

Moreover, Copado includes built-in agile tools to manage your releases:

* **Work Manager**: Intended to streamline agile development at every phase of the project.
* **Sprint Wall**: You can manage your sprints in a more friendly and agile way with a view that provides useful information regarding the status of the sprint and a whole picture of its content.
* **Agile Dashboards**: Out-of-the-box dashboards to better monitor the work of your team.

With license-based user rights and out-of-the-box permission sets, Copado gives you the ability to define release team roles (Developer, Admin, Release Manager…) and the **Copado Permissioner** also allows you to have more granular control over permission set assignments to users.



**Full Range of Quality Checking Options**

Out of the box, you will find in Copado the possibility to execute unit tests related to the changes being committed. By running unit tests directly from the user story, you will be able to determine if code coverage is acceptable for your organization.

In addition to that, Copado will provide a list of the methods executed and the result of the execution, enabling an instant detection of failing methods.

Copado offers capabilities far beyond just running Apex tests. You can automate the execution of Apex tests and customize the minimum Apex code coverage for each environment to be above the 75% required by Salesforce.

Since Copado is built on top of Salesforce, you can easily tailor your release process with Salesforce standard features like approvals or validation rules.



**Static Code Analysis** is a type of quality check that allows you to analyze your code to find inefficiencies and detect common errors in the code. Code analysis can be performed using PMD, CodeScan, or SonarQube.

Code scans are quite flexible. They can be performed on-demand or automated to be run at a scheduled time. The analysis can inspect all the code of a certain environment or only the content added to a user story.

Code review findings are stored as records, hence you can access the results and assess the violations straight from Salesforce without having to move through different applications.

You can use code analysis to automatically monitor developers’ code and enforce implementation best practices. With Copado’s static code analysis you can create and prioritize rules depending on your business needs as well as decide whether code can be moved to an upper environment or not.

Copado is packed with other quality analysis tools such as automated regression tests with **Selenium**. Copado also comes with a Selenium step recorder extension known as **Copado Recorder**, which translates simple point-and-click activity into effective UI tests without requiring your team to know Selenium or write lines of code.

The recorder gathers detailed information about the elements being interacted with, screenshots of the area, feedback of commands, and much more.

You can leverage Copado’s **validation deployments** for each user story individually and also for an entire release. While validating changes outside of Copado, you can assert that changes from the source environment can be deployed to the target environment. However, this process is not considering components as they are in the source environment and how they would be once they get merged.

Validating changes with Copado will ensure that the validation deployment is made out of the merged components thus avoiding unpleasant surprises.



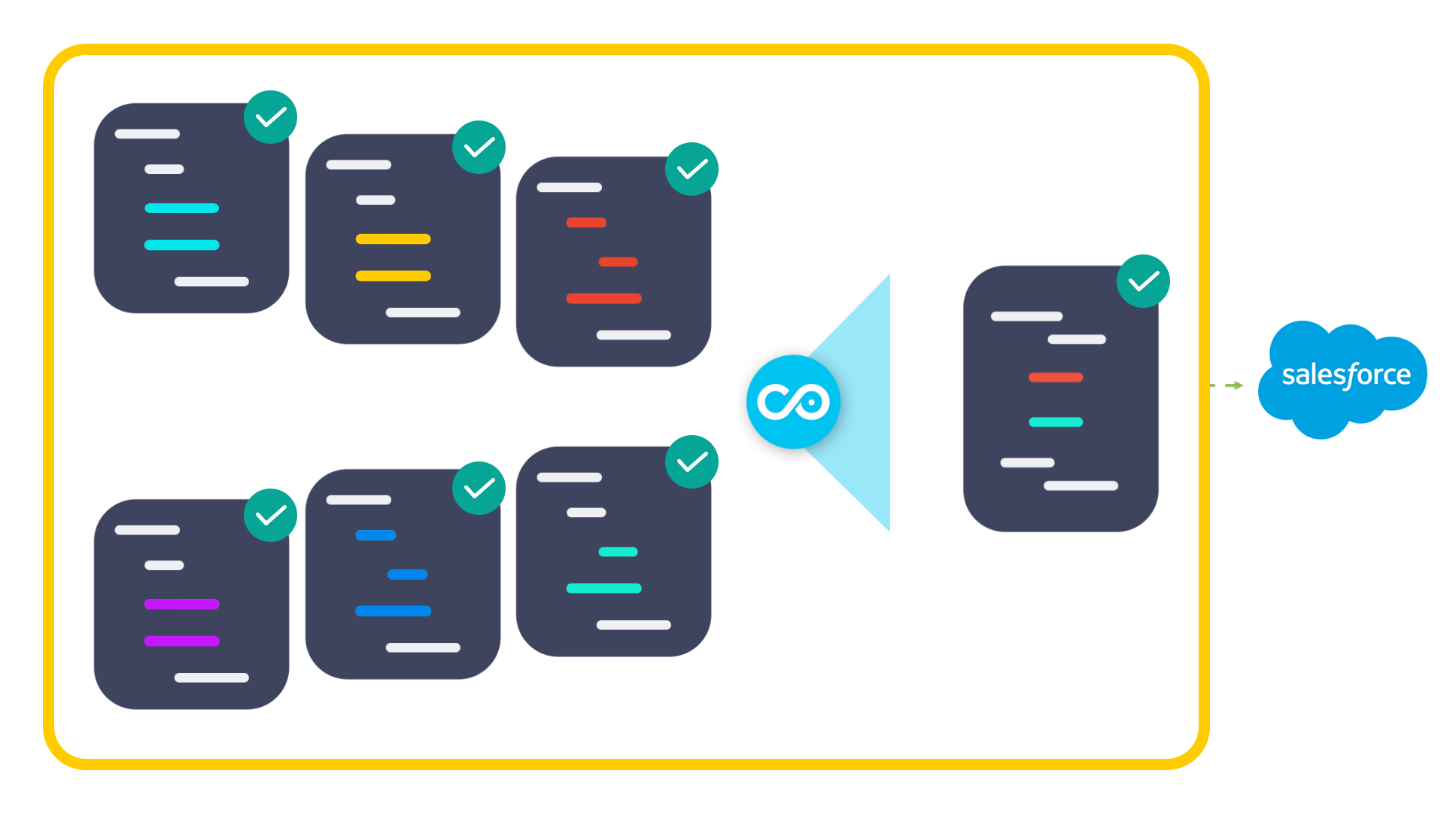
**Solving the Lack of Monitoring and Compliance**

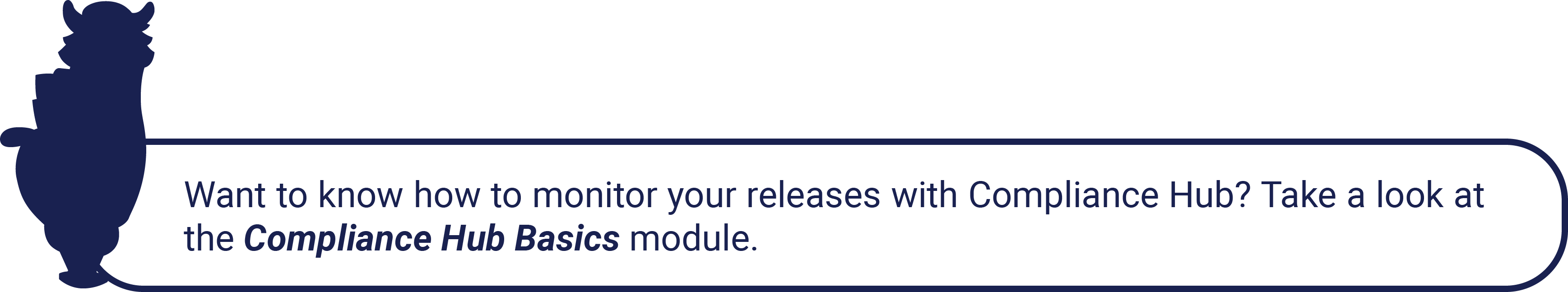
You can start leaving manual monitoring behind because Copado gives you a compliance hub tool that allows you to scan all the metadata changes made in your environments in order to spot and identify security breaches.

You can choose which metadata types you want to be monitored and enforce compliance rules according to your business needs. You can set a severity for each rule based on how critical the rule is for your company and choose the action that should be taken if the scan finds the rule has been violated.

Depending on the action selected, Copado will abort the process on the fly, preventing the noncompliant change from moving to the next stage, alert the running user, or just document the findings.

You can use Copado Compliance Hub in real-time during commit or deployment, or you can schedule scans to run behind the scenes - so-called “passive scans”.





**Deliver Changes to Production in a Quick, Safe and Sustainable Way with Continuous Delivery**

Copado Continuous Delivery enables you to schedule promotions and back promotions and automate all the quality gates we have talked about before.

It is important to understand that continuous integration is all about developers being in sync. Most developers have ownership of this stage of the process since continuous integration lives in the development and integration sandboxes.

The continuous delivery stage starts after the continuous integration stage. The approach and the automations that apply to each stage may be different, and Copado makes it easy for you.

With Copado’s connection behaviors you can define your continuous integration process and set up automated or scheduled back promotions. In this way, Copado will be constantly merging qualified changes and moving them to the integration environment.

You can also create metadata groups to define specific quality gates that will be triggered if changes to the metadata items within the group are submitted, e.g. running a compliance scan whenever a profile or permission set is deployed.

**Deploying More Than Metadata**

Think for a moment about your last release. What were the actions you took before successfully completing the release? Did you create records? Was it necessary to assign permission sets? How did you manage CPQ? Did you require to deploy Vlocity configurations?

The truth is enterprise releases go beyond metadata. Copado was designed with that in mind, understanding that metadata is an important piece, but that release managers also struggle with manual tasks, deployment of managed packages configurations relying on records, JSON files, custom settings values, and many other things.

Copado has filled this gap by incorporating into its platform the possibility to deploy all of these changes in the same way you deploy metadata, thus significantly reducing the amount of manual effort required and eliminating the burden of trying to orchestrate different systems for the release to succeed.

So what can you deploy or do with Copado?

* Manual Pre/Post Deployment Tasks
* Relational Data
* Vlocity Configurations
* Custom Setting Values
* Apex Anonymous Scripts
* Users

With **Copado Data Deploy**, you can create point-and-click data templates with no technical knowledge in order to build relational queries.

Directly from the interface, you can create a data template for each object, connect these data templates to retrieve only the related records, and deploy data from multiple objects in one deployment just following the steps of the data deployment wizard.

To help you preserve the governance of your data and keep your organization compliant, Copado enables you to match the owners of the records from source to target. If you have sensitive or confidential information that you may not want to deploy, Copado gives you the possibility to mask or scramble the data on deployment.

Copado also provides a set of pre-built data templates that you can use to deploy configuration data for other packages such as nCino or CPQ. You can import these templates to your org and customize them according to your requirements.



**Get a 360 Vision**

With **DevOps 360** you can accurately measure the performance of your Salesforce implementation.

**Value Stream Maps** enables you to visualize your entire team as a flow and easily identify bottlenecks in your DevOps process.

**DevOps Analytics** increases the visibility into your DevOps process with additional metrics that help you to proactively spot where your performance is slowing down.

**Increase the Quality and Frequency of Your Releases**

With all this, you can build a high-performing DevOps process as described in the[***Pathfinder to Salesforce DevOps***](https://success.copado.com/s/learning-plan-detail-standard?ltui__urlRecordId=a970N000000Cde7QAC)module. This means you can reduce the time you spend on tedious activities like deployments and re-invest that time in developing and testing new features.

While speed and efficiency are important, ensuring quality is even more important. Having proper quality gates in place enables you to send the most critical updates more quickly, reduce the number of failures and recover faster.

Sounds interesting right? Now that you have an overview of the constraints that Copado helps you to overcome in order to succeed, continue with the next unit reading to know how you can establish the governance of your organization with Copado.

Done!