Reinventing the Wheel: The Hub-and-Spoke Topology of SaaS Tools

Executive Summary

Most leaders think of their tool stack as a toolbox; a static container holding the tools your teams need to get their work done. But you're not getting a full view of what's actually going on when you treat all these tools equally. Tools have relationships. Understanding these relationships can smooth out the way your teams work. That's what the Hub-and-Spoke topology is for. Think of your tool stack as a wheel with these two categories:

- **Hub:** The central location where most — if not all — your team's work is represented. That might be a project management tool or a CRM tool.
- **Spokes:** Every other tool your team needs to get a specific job done. Most tools will fall in this category. Think version control tools, marketing automation tools, and ticketing systems.

Mapping out your tool stack in this way promotes a natural flow of information from Spokes to theHub, which becomes a single source of truth.

By topology, we mean a structure used to understand a specific concept. In this case, Hub-and-Spoke topology is used to understand how different SaaS tools interact with each other across teams. However, not all teams use the same Hub. A marketing team might use a project management tool, such as Trello or Asana, while a development team may rely on Jira as their Hub.

The Hub-and Spoke-topology doesn't solve the problems these tool differences can cause but demonstrates how they can be solved. It reveals the gaps in your tool stack, where tool integrations need to connect Hubs and Spokes.

Integrations that Unito can provide.

Now that you have the basic rundown of this concept, let's get into the weeds of it.

Intro: The software toolbox

There are more than a few sayings around tools and the people who carry them: “always use the right tool for the job.” “When all you have is a hammer, everything looks like a nail.” “It's a poor craftsman who blames his tools.”

While you may be picturing a bright red toolbox full of hardware, each piece with its own specific purpose, the same thinking can apply to software tools. Try managing projects from a Word document and you’re probably not going to get very far.

It’s an intuitive analogy, but does it work for your tool stack?

A toolbox, the literal, physical item, is bound by its size. You can’t fill it with more tools than it can hold. That limitation doesn’t exist in the software world, as these trends show:

- Spending on SaaS (software-as-a-service) tools is expected to reach $397.5 billion by the end of 2022.
- The average company uses 254 SaaS tools.
- The average team uses between 40 and 60 apps.
- Fewer than half of those tools actually get used.

While your SaaS toolbox isn’t limited by the laws of physics, it’s easy to let your stack sprawl or feel overwhelmed by a wide range of options. Moreover, each department, team, and individual collaborator can have their own set of preferred tools. Much in the same way that a physical toolbox protects what’s inside, a specific SaaS stack can keep crucial data locked in.

So what’s the problem?

Approaching your SaaS tools on a tool-by-tool basis — and grouping them up as toolboxes — is great when you’re budgeting and taking stock of all the apps your team uses. If you’re trying to figure out which work management or spreadsheet tool you should buy, you want to treat those tools as equals.

The same is true when you're trying to measure tool usage, ROI, or other metrics that help you build the best SaaS stack you can. But when it's time to oil creaky workflows, streamline tasks across teams, and eliminate busywork, this approach can falter. While every craftsman can bring their own toolbox to a construction site, the same isn't true of a team relying on SaaS tools.

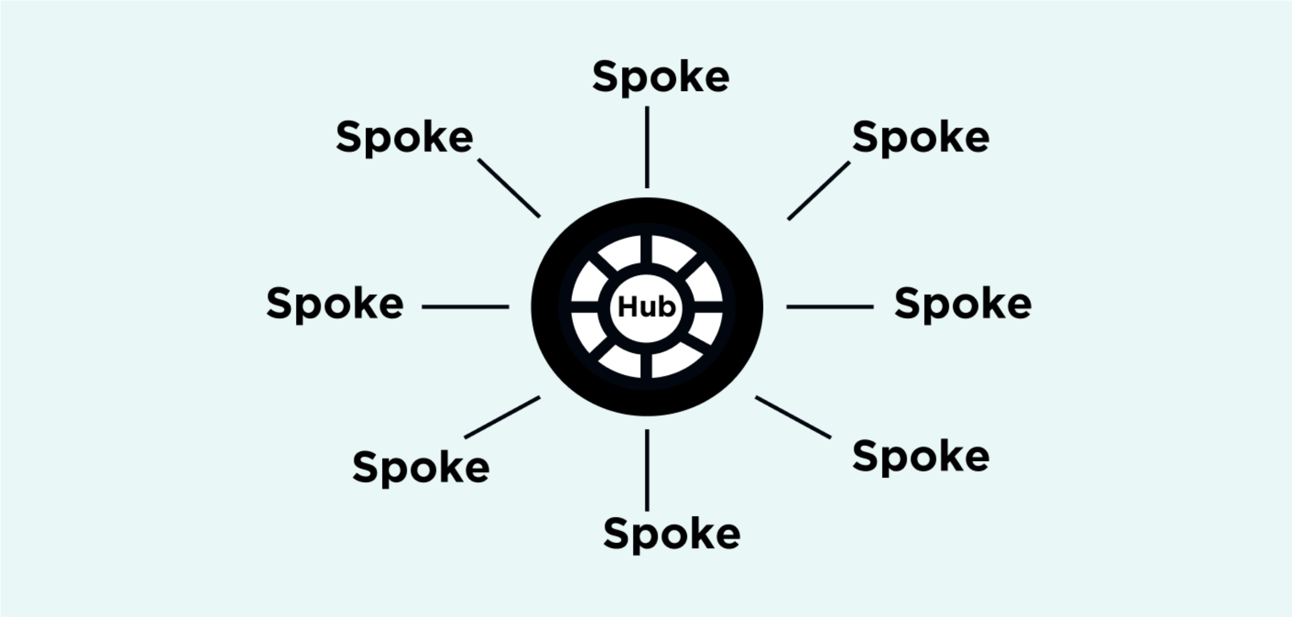
That's why you should be using Hub-and-Spoke topology instead.

What is the Hub-and-Spoke topology?

The "toolbox" approach to SaaS puts all your tools at the same level. Hammers, screwdrivers, and wrenches all serve unique purposes, but they swap in and out depending on the job. Although you could try using a wrench as a hammer, or a screwdriver as a pick, we don't recommend it.

But treating all these tools as equal doesn't accurately represent the way they work together.

That's why the Hub-and-Spoke topology turns this flat structure into a hierarchy.



The Hub?

The central location where most — if not all — your team's work is represented. That might be a project management tool or a CRM tool.

It might not necessarily be where they spend most of their day, but it's the first tool you think of when asked "where can I see what everyone is up to?" It's where you plan future work, manage tasks, and track deadlines. It's where conversations that would be pretty casual in a chat app turn into serious "let's get this done" talks.

Your team's Hub is the core of everything they do. The business at large can have a single Hub too, depending on its size. Enterprise-level organizations — especially multinationals — are more likely to have multiple Hubs.

Examples of Hubs include project management tools, e.g., Asana, Trello, or Jira. A team's Hub might also be a CRM platform, an email client, or a note-taking app, depending on their role.

The Spokes

Every other tool your team needs to get **a specific job** done. Most tools will fall in this category. Think version control tools, marketing automation tools, and ticketing systems. These specific tools can vary wildly depending on the team, individual contributors, and other factors. Nevertheless, they can be just as — if not more — essential — to your daily work as the Hub.

A Spoke is generally dedicated to a specific purpose. While your team's Hub is generally a project management tool, Spokes can be any kind of tool, so long as they accomplish a specific job and they're part of your stack. Here are a few examples of common Spokes for different teams:

All teams: Spreadsheet tools, word processors, password managers, chat apps, video conferencing, time management, note-taking, presentation tool, cloud storage, data management, screen sharing, email client, calendar.

Marketing team: Content management, email marketing, social media management, editorial calendar, SEO, photo editing software, content scheduling.

Software teams: Web hosting, code editing, version control, software integration, security enforcement, monitoring.

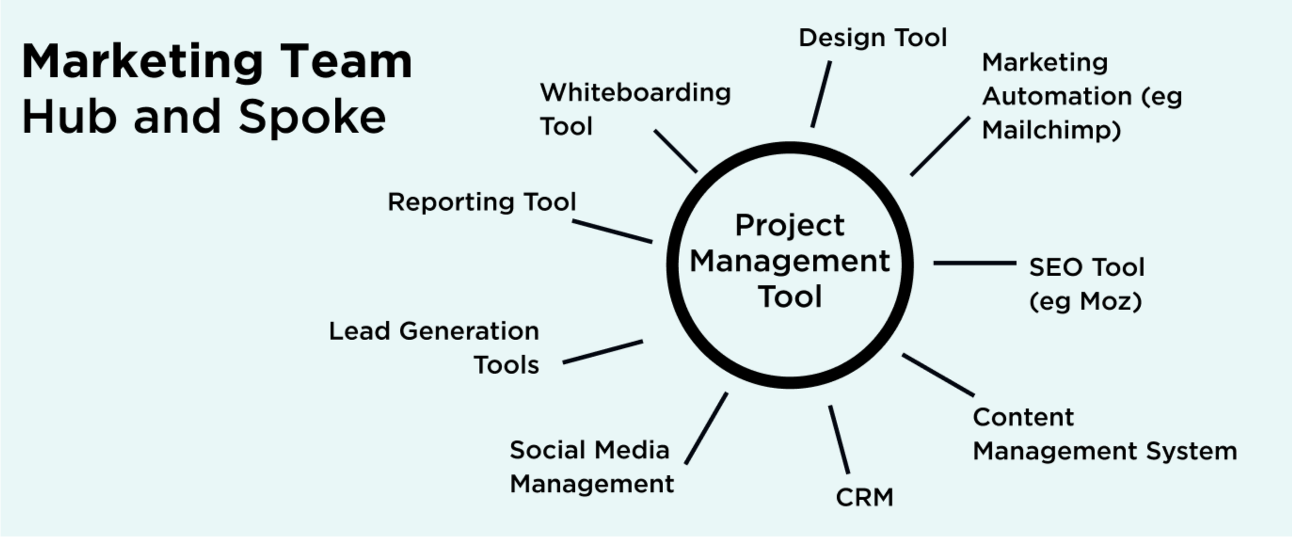
Customer-support teams: Live chat tools, meeting management, help center manager.

Sales teams: sales automation, enrichment, sales enablement, live demo tool.

BizOps teams: Employee benefit tools, payment services, employee feedback platforms.

Some teams can have tool stacks with dozens of Spokes, some of which might only be used a few times each month, or even once a quarter. While each one might be essential to your work, they aren't used with the same frequency as your Hub.

When you map out your tool stack according to the Hub-and-Spoke topology, you get something that looks like this:



Why are all the spokes connected to your Hub? Because when they're not, your team suffers. Almost a third of marketing leaders surveyed in Airtable's Marketing Trends Report said that gaps between their tools cause stress for their teams.

Why the Hub-and-Spoke topology matters

Thinking of your tool stack like a toolbox is straightforward and intuitive, but it doesn't really fit the way your tools work together. Here's why the Hub-and-Spoke topology is a better way to think about your tool stack.

The relationships between tools matter

With the toolbox topology, the relationships between your tools are a lot less clear. Some tools fit naturally together, like a drill and bits — or a CRM and a contact management tool. But, generally, there's no easy, clear way to determine how they work together. That might be clear to you when you're talking about your tools, but how can you quickly explain that to someone from another team? Or even another department?

The Hub-and-Spoke topology naturally lends itself to mapping out these relationships. Once you've established your Hub, you can relegate other tools to the Spoke role. Note that you don't have to keep these tools separate from each other. You can drill down in more detail to draw out the relationships between them, and even create "mini-hubs" to represent smaller networks within your primary tool stack.

Generally, we tend to approach our stack on a tool-by-tool basis. We'll determine which ones we need — a spreadsheet tool, a work management tool — and try to plug the holes with the best tools we can get. We'll think about our budget, our team's needs, and how each tool solves those problems.

But once the tools are acquired, we hold to that approach. Each tool is a block that goes into a specific slot of our stack, and that's it. That might work when you're acquiring tools, but not so much when it's time to smooth out your workflows.

If you want to understand how your teams work – and solve the collaboration problems they run into daily – you need to recognize how interconnected their tools are. The relationships between those tools can either be a blocker or an accelerant. The Hub-and-Spoke topology makes them the latter.

The alternative breeds passivity

Once you've finally gone through the process of buying a tool, the last thing you want to think about is whether or not you made the right choice. It's like buying a car. You could spend weeks comparing prices, features, mileage, and fuel efficiency. When you finally decide to drive it off the lot, you don't want to start digging around and figuring out how everything works or second guess yourself. You just want to feel satisfied that a decision has been made and the hard part is over.

But in the same way that a car won't last long if you don't get it serviced regularly, a new tool won't do much good for your team if you treat it passively.

What do we mean by passively? It can mean sitting back, hands-off, and just hoping that your team uses the tool as best they can. As attractive as that might sound, it's not conducive to adoption or productivity.

People will naturally follow the path of least resistance, figuring out how to continue working the way they always have if an alternative seems more challenging. That can mean losing out on essential features, lower adoption, and generally just a waste of a new tool.

Who in the team is the first to learn about new features? Do they spread them to the rest of the team? How much of the tool you acquired are people actually using? Are there bad habits that have popped up that are costing everyone time and effort? These are all questions you should be asking yourself for every tool in your stack. That can't happen when you stay passive.

Why does the toolbox topology breed passivity? The answer is simple:

- We don't use tools until they're needed.
- We aren't pushed to consider the relationships between tools.
- When it's time to map out and improve your workflows, you have to start from scratch, and it's easy to convince yourself it's not worth doing.

Adding a new tool to your stack requires a big investment. You need to research your options, spend time testing them out, and get your team to see the value in the tool you ultimately choose. If you treat your tools with passivity – and your teams do too – you're losing out on that investment.

So let's review

What problems does the Hub-and-Spoke topology solve? In a nutshell:

- **The fallacy that all tools are equal:** Look at your own stack and you'll immediately see this isn't true.
- **Invisible tool relationships:** Relationships matter as much as features, because tools don't live in isolation.
- **Passivity:** Approach your tool stack with passivity and problems will grow and fester. The right topology encourages you to be active.

The Hub-and-Spoke topology is a better way to approach your tools. Map out your stack in this way, and you'll instantly get a better understanding of how your tools interact.

That'll help you find the places in your workflow that need improvement, reduce overall tool spend, and understand how your teams are really using these tools. But what if we told you there's a flaw with this topology, a flaw that every tool stack has?

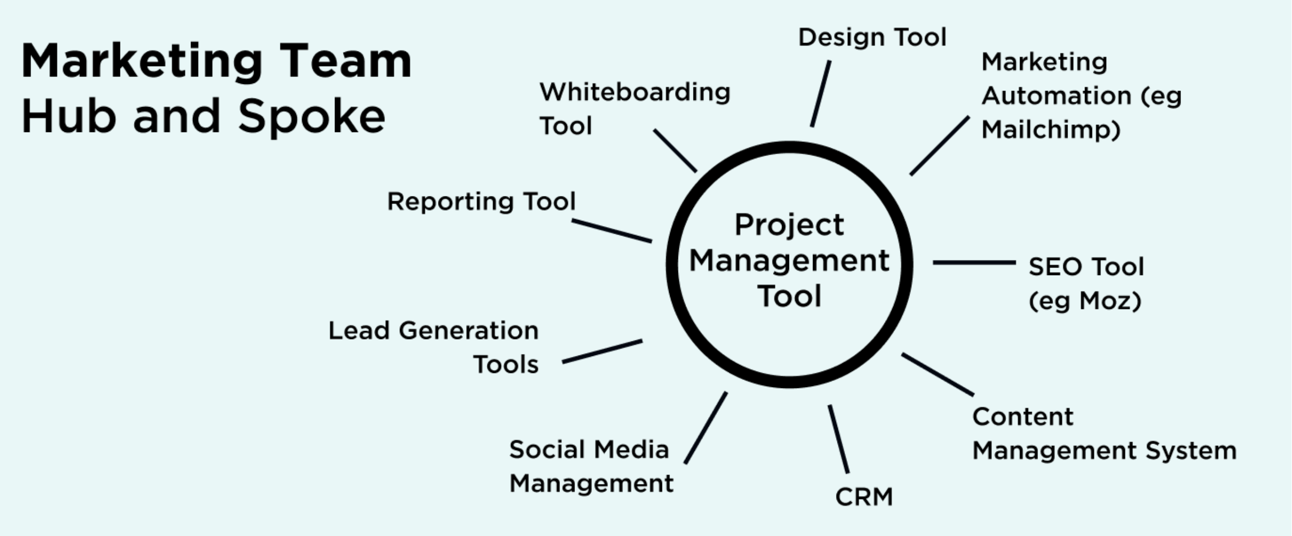
The problem of subjectivity

What perspective are you taking when you think about your tool stack? If you're leading a software development team, you probably have a different understanding of your organization's stack than the VP of marketing. Even if you try to be objective, you'll bring a bias that colors your perception of the organization's tool stack. This is especially true when you think of your stack as a toolbox, but it's also true of Hub-and-Spoke topology.

But here's the difference; the latter actually helps you identify and address this problem.

One way it does that is by giving you the words you need to explain the similarities — and differences — between tools different teams use. For instance: not everyone uses the same Hub.

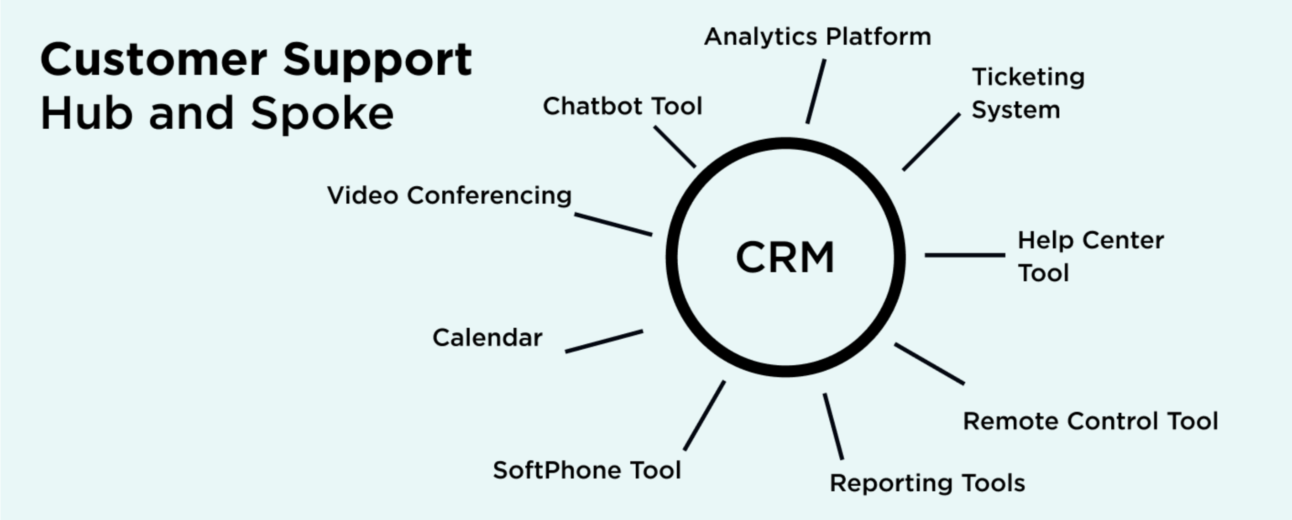
Let's look at a representational diagram for a marketing team.



The project management tool is the marketing team's Hub. Regardless of individual preferences, everyone's work makes it through the Hub at some point. But notice what's among the Spokes?

To a marketer, a CRM tool is a Spoke. They might use it only to pass leads along to the sales team, or they may use its marketing features, such as HubSpot's automated email campaigns.

But can that same CRM tool be someone else's Hub?



For a salesperson, the CRM platform is the Hub. It's where conversations with individual contacts are logged, campaigns are launched from, deals are tracked, and more. No matter what other tools they're using, everything goes through the CRM at some point.

And that's how subjectivity comes into play. A tool that's just a Spoke for a marketer can be a Hub for another team.

The Hub-and-Spoke topology is uniquely suited to finding ways to work around these issues. By creating a tool hierarchy, you can quickly understand how important a specific tool is to another team without much context. Just tell someone a specific tool is your team's Hub and they get it. If you try to cover how your task management system works, you might just get some blank stares.

If team one's work always goes from individual Spokes to their Hub, and team two has a different Hub, how do you get crucial information from one team's Spoke to that of another team?

From a purely theoretical standpoint, that information should go from Hub to Hub.

As the main source of truth for a team, a Hub is where they know to go for updates. In practice, however, that isn't always the best approach. If team one uses Google Sheets as their database and team two uses Airtable, should you have the data from one tool go through *two Hubs* before it gets to Airtable?

You see the problem. Even if you try to take the most direct approach to transfer data between tools, you're potentially creating inefficiencies and other problems for your workflows.

The Hub-and-Spoke topology and tool integrations

By thinking of your tool stack as an interconnected ecosystem with the Hub-and-Spoke topology, you're smoothing out your team's workflows and taking an approach that promotes tool adoption. But what if there's a way you can take this even further?

As your stack grows, so does the need for the appropriate integration. With more Hubs and Spokes, your teams generate more information that needs to be kept in sync. That's especially true if you rely on cross-functional collaboration — and who doesn't?

You could rely on manual work to bridge the gap between your tools. After a few months of data entry, copying-and-pasting, and double-checking multiple tools for updates, your teams will clamor for a better solution. Connecting Hubs exclusively seems like a natural start, especially when so many of these tools come with built-in integrations.

After all, if critical work has to go through them, why not focus your efforts?

The Airtable and Google Sheets example above shows that's far from ideal.

You need a way to connect Hubs to each other, Hubs to Spokes, and Spokes to Spokes. Here's why:

- **Hub to Hub:** When collaborating between teams, much of what you're doing needs to go from Hub to Hub.

It's where you track your initiatives, dispatch work, and surface your reporting. By integrating Hubs together, you make sure every team has the same information.

- **Hub to Spoke:** If you follow the Hub-and-Spoke topology, work needs to flow naturally from Spokes back to your team's Hub. The amount of work and depth of information that needs to follow this path will depend on the actual workflow. That said, you'll almost always want to represent some of the work happening in Spokes in your team's Hub. Integrations make this process smoother and eliminate manual work.
- **Spoke to Spoke:** Since not all kinds of work need to be represented in-depth in your Hub, sometimes it's valuable to send data from one Spoke to another. This can be true between teams, too; their collaboration might depend on these connections before anything gets reported to a Hub.

In short, you won't get away with a single integration. If you want your teams to operate at maximum efficiency, you need a way to integrate tools at all levels of the Hub-and-Spoke map. You need a solution that can do it all.

Enter Unito.

Unito is a no-code workflow management platform that enables the deepest two-way integrations for the most popular tools on the market. In just a few minutes, you can build a flow that connects your tools and keep information in sync between them in real-time. Connect Hubs such as Asana, Trello, or Jira to Spokes such as Google Sheets, HubSpot, Mailchimp, and more. Unito supports over 30 integrations to cover a broad range of workflows.

Learn more about Unito's integrations, and how they can transform your tool stack.

To sum things up

Your tool stack isn't a toolbox because tools aren't created equal. A tool is either a team's Hub, where most of their work is represented, or a Spoke, where a very specific job gets done. You'll get a better sense of the relationships between these tools by mapping out your workflow following this topology.

The goal is to spot potential weaknesses in your workflows and find ways to deal with them proactively. You'll also gain a better understanding of how vastly different a tool stack can be across teams, or as sub-stacks of your organization's broader app portfolio.

And, of course, when you add a workflow management tool such as Unito, you turn that map into a collaborative network that enables data to flow naturally, in real-time.

Think of your tool stack as a wheel, as Hubs and Spokes, you'll always be one step ahead.