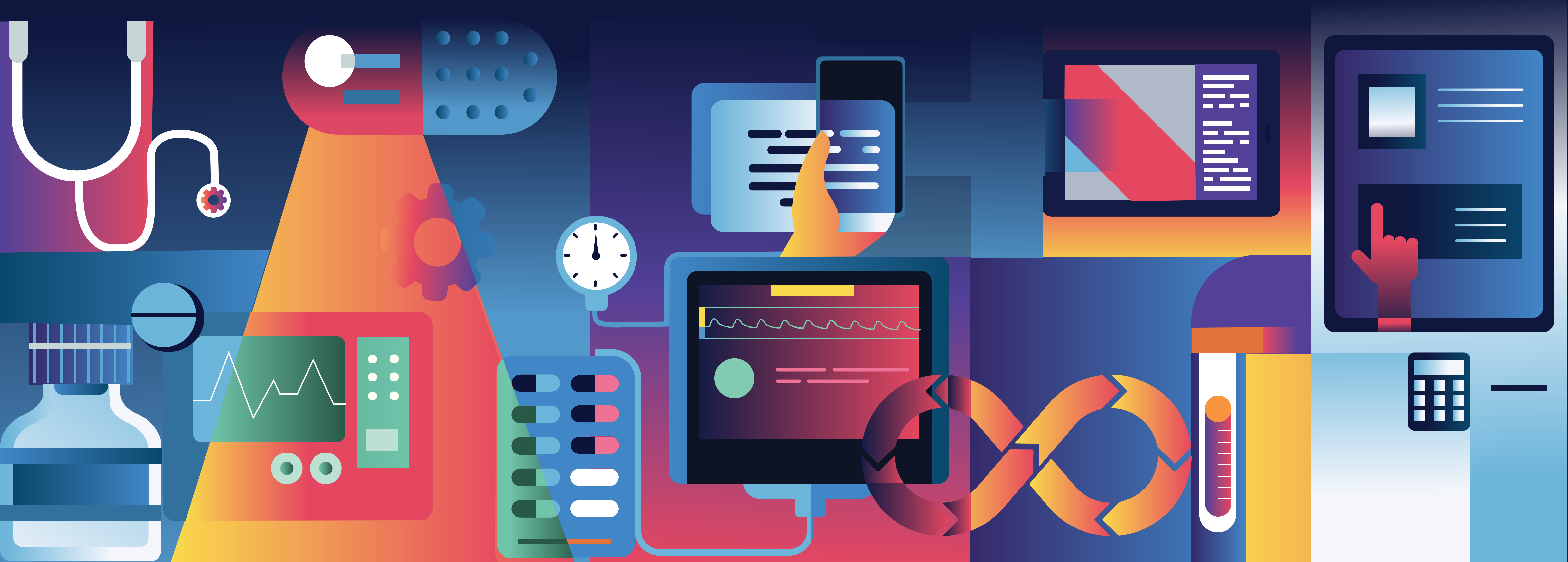
esper

5 keys to building a strong foundation to your HealthTech device fleet



“The next generation of innovative customer experiences is being driven by connected devices, and it’s more clear in the healthcare industry than nearly anywhere else. ”

So how do you get your device strategy off the ground? How do you meet the moment and accelerate innovation and time to market?

At Esper, we’ve worked with thousands of organizations who are at various stages of their device fleet evolution. We start working with some before there’s even a product, and we’ve guided them through the hardware selection process and helped them make the right decisions from the very start. Others already have large fleets of devices out in the field and come to us to help them better manage their devices because they lack some capability that is meaningful to their business.

Across all those experiences and learnings, whether you’re building a remote patient monitoring solution or an electronic clinical outcome assessment (eCOA) solution for clinical trials, there are five principles that your team can use to more effectively and efficiently build a HealthTech device fleet.

1. Let customer needs drive hardware and OS decisions (not your tech)
2. Think about scale early
3. Build continuous improvement into your device DNA
4. Measure everything...and manage by exception
5. Integrate security into every step

Let customer needs drive hardware and OS decisions *(not your tech)*

Devices in the healthcare space are typically used by people deeply removed from the device builders, such as patients using medical monitoring devices in their homes or hospital staff using tablets to improve day-to-day efficiency.

These users may not be tech savvy or simply do not have the time or training to fiddle with device and app settings. Not only is it about ensuring delightful and convenient experiences for your end users, but requiring them — or even allowing them — to troubleshoot devices can lead to security and compliance issues.

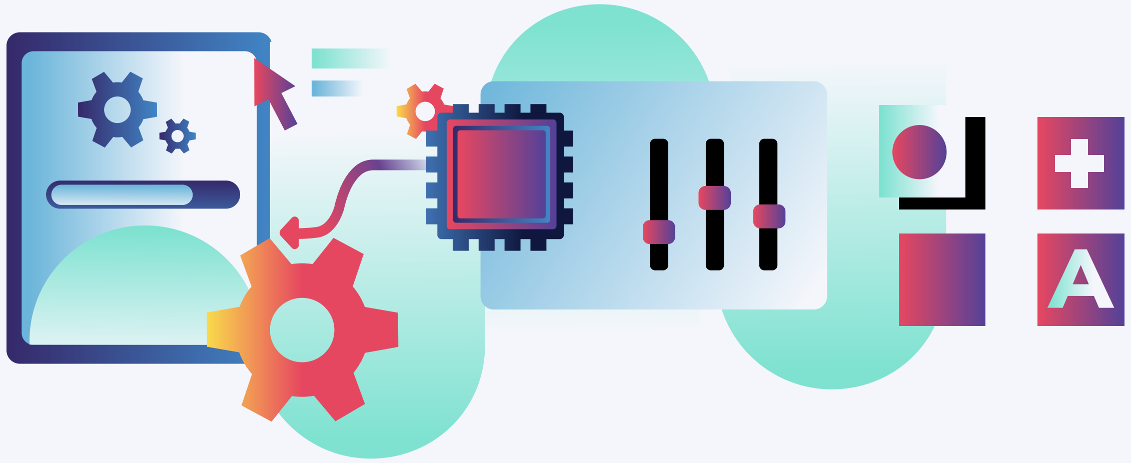
Having end users troubleshoot technical issues can compromise the safety of the device or the validity of the trial. The bar for healthcare solutions is clear:

"The devices need to just work."

Every solution journey starts with an idea to improve either the patient or practitioner experience. Sometimes that can even mean letting end users choose to use their own device (BYOD). Other times — often for risk and regulatory, consistency, or logistical factors — you'll need to provide a vendor-provisioned device. In this instance, you have the choice of hardware and operating system, and the tech to manage it. Too often, companies get the order wrong and let the tail wag the dog. They start with the management software, perhaps because it's already being used within the organization for something else, and that ends up dictating decisions. This limits decisions in addition to the core device, like hardware peripherals, because hardware-software compatibility is complex.

When putting the customer first, the first decision is about balancing form factor, familiarity, and cost. And that can dictate whether you choose something off-the-shelf or build something custom. When it comes to doing the best thing for patients, don't be afraid of custom builds. Today, there are numerous ODMs who you can work with to meet your performance, form factor, and cost needs.

This can even lead you to a mixed fleet:



You start with something off the shelf to get a pilot running, and then layer on customized configurations or even custom hardware for a more specialized, complex, or long-term use case. After you've chosen the right hardware solution to meet your customers' needs, you can then pair the right device software and management tools

***Pro-tip:** Choose something flexible enough to meet potential future needs, too. If you get locked into specific hardware with the choices you make today, you may lose out on great opportunities down the road.*

spirehealth

We can look at Spire Health to see a customer-centric hardware-software combination in action.

Spire Health provides patients with monitoring for chronic respiratory disease, and entered the HealthTech device market just as the demand for remote patient monitoring spiked with the onset of the pandemic.

In building the device, it was essential for Spire to work with a partner who understood that at-home tracking for sick and elderly patients requires a simple setup. The solution criteria was clear: The team needed to be able to work remotely with a non-tech-savvy patient population, be able remotely solve connectivity issues, and drive OpEx efficiencies with a nascent in-house IT team — all while ensuring HIPAA compliance and meeting healthcare security standards. Ultimately, they landed on wear-at-home trackers that connect to mobile phones using Bluetooth Low-Energy (BLE).

With device telemetry data, they are able to track biometric data and send patient status alerts to care providers much earlier than traditional monitoring.

Think about scale early

Cost is an important consideration in every device strategy. As you consider build and management costs, it's important to include both the hard and soft costs in your evaluation, as well as how they scale as your fleet scales.

At 10 devices, you can pretty much do everything manually, so hard costs like hardware costs and device management subscription costs are more apparent. Even if provisioning a device or updating device software takes 10 or 15 or 20 minutes per device, you can feasibly do it with negligible incremental resources.

You can touch every device, plug a USB stick in, and make an update. It would even be possible to send someone out in the field if that's where they are. But as you get to 100, 200 ... 1,000, 10,000 devices or more, all of that becomes prohibitively expensive from a person-hours perspective. Much more so than software, when it comes to physical devices in the real world, thinking about scale becomes really important much earlier than you think.

We worked with a customer with thousands of facilities across the country with dozens of devices in each, creating a pretty large fleet. Before they came to us, every time they wanted to add a new device to the fleet, someone had to follow an 80 page instruction document from unboxing to getting the device live. Of course, like many large enterprises, they outsourced it to a third-party staging company. But still, at the scale they were working at, that expense added up. Figuring out your provisioning mechanism early pays off. With more advanced device infrastructure where you can seamlessly and remotely provision new devices, you can reduce the vast majority of per device instructions down to essentially just the physical setup. Multiply that time savings by tens of thousands of devices, and that can easily add up to many millions of dollars saved.

Additionally, having the ability to reuse or repurpose your company-owned devices can save significant costs. For dedicated devices, this means the ability to reprovision devices with different configurations and applications, often remotely over the air. For example, a customer that runs clinical trials has the flexibility to reconfigure and retool devices between trials, so that the same device can be used across multiple studies. Especially as they scale, the cost savings from being able to easily and securely reuse devices becomes significant.

Build continuous improvement into your device DNA

Countless number of teams that we've worked with were previously scared to make any update, and as a result, they spent weeks testing it in fear that pushing the update broadly will have negative consequences.

To be fair, with devices in HealthTech, those consequences can be severe. If you brick an RPM device, for example, that might mean missing data for weeks as you need to ship out a replacement, and in that time there could be an undetected health issue.

But on the other hand, in order to make customer experiences better, you have to make updates. The more frequently you can make updates, the more frequently you can improve the customer experience.

If you're committed to digital transformation, you want the ability to make updates whenever you have business reasons to do it. You don't want to be limited or restricted by your technology stack. For example, if there are security implications that require you to change core device settings, like adopting a new policy that forbids any fleet device from using USB connectivity, you want to be able to roll that update out fleet wide as quickly as possible.

Measure everything...and manage by exception

Telemetry is important for cloud products, but it is especially important when it comes to dedicated device fleets, and even more important when those dedicated device fleets are mission- or even life-critical.

A connectivity error or low battery can result in serious health consequences. With more devices in the field than any one human being could hope to personally monitor, it's both critical and challenging to understand the state of your fleet while maintaining powerful visibility into individual devices. To achieve this at scale, you need tooling and automation. Conceptually, this means taking the approach of managing by exception. Rather than trying to keep track of the state of every device (which you do want to have access to), managing by exception means you only want to be alerted to take action when a device deviates from its expected state. This is the concept of drift.

Every device has a stated ideal configuration. This could be the application version, admin settings, online or offline state, geolocation, etc. If the device's actual state, for whatever reason, does not match the ideal configuration, it has drifted and you want to be notified so you can take remediation actions. We have a customer in the healthcare industry and they have devices all around the world collecting data to be collected and analyzed by doctors. However, they have a few reasons why they need to have devices with intentionally unique configurations — sometimes referred to as snowflake devices. For example, to serve users in different regions, they need to apply different device configurations for language and regulatory reasons.

This requires them to be able to understand the difference between unplanned drift and managed drift. They want to be able to track and monitor all those devices with irregular configurations while it is necessary — differentiate them from devices that may be suffering from unplanned drift— and then be able to reconfigure them back to standard once they're done. Doing this at their global scale requires strong observability practices.

Integrate security into every step

While great for delivering rich customer experiences, device fleets, especially ones that involve highly sensitive data, are also a rich target for bad actors.

The most successful security teams approach security in ways that are appropriately flexible and adaptive: continuous improvements, rapid deployments, and intelligent automation strategies. This is DevOps, and it is transforming how product teams approach cybersecurity. Many call this emerging area “DevSecOps” — and it is essentially taking the same philosophies to knock down the wall between development, operations, and now security teams. This dynamic is most prevalent in highly regulated industries like finance or healthcare, but it really should be practiced everywhere.

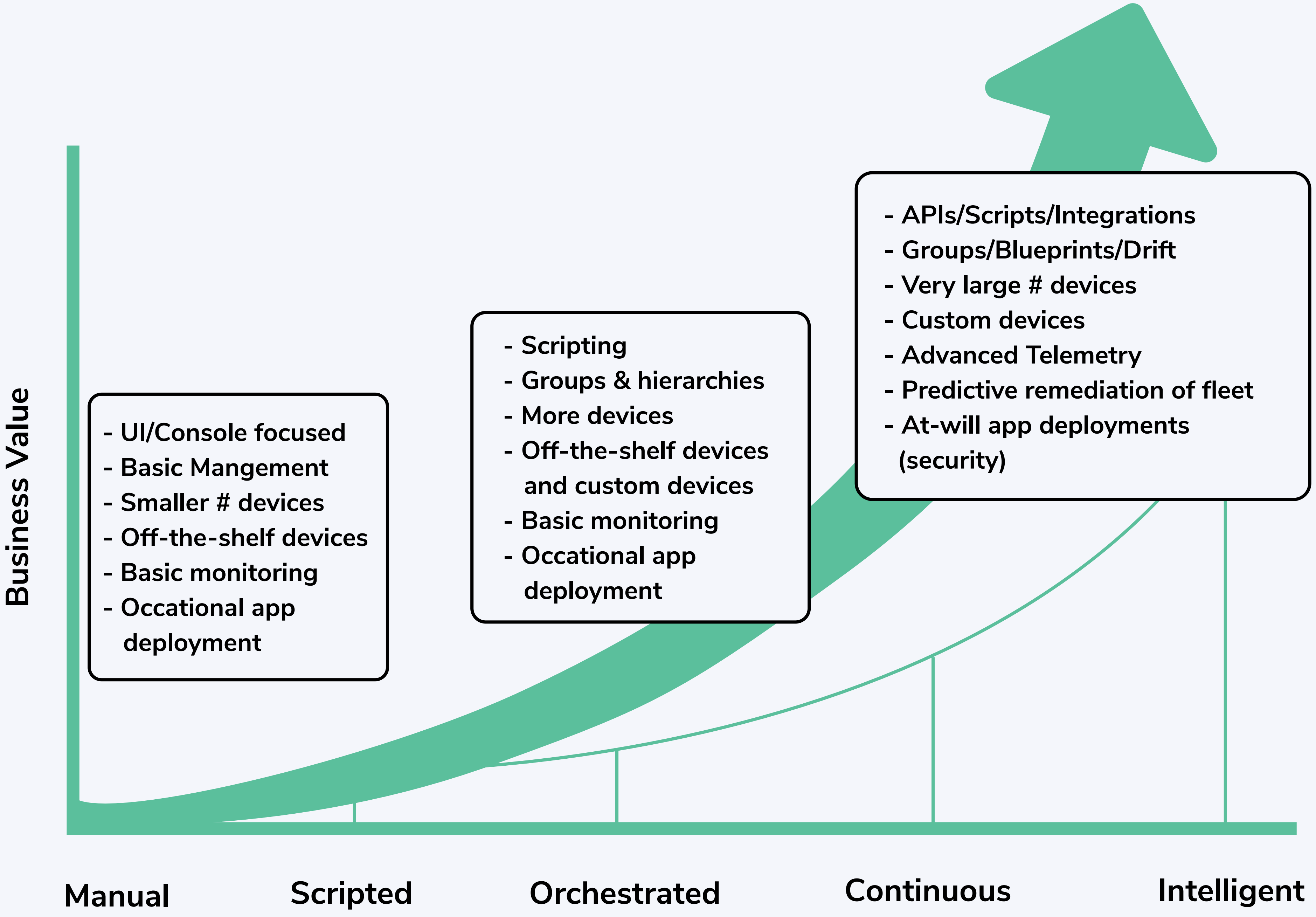
Some aspects of device security are quite straightforward. Several of our customers, such as healthcare facilities, have devices like tablets that are not fixed to a specific location. Whether intentionally or accidentally, if the device leaves the room, hall, or facility’s perimeter, there’s potential security vulnerability. With advanced device management tooling, in the device configuration, customers can set rules that lock the device if it leaves a geo-fenced certain area, ensuring the device is only used for its specific purpose.

Other customers, especially in the healthcare space where there are numerous data compliance regulations, have more complex security scenarios. One company that had a 60 day security review after code complete to ensure it met strict data standards. Of course, at the pace of innovation today, that’s not great. By integrating security reviews into the development cycle — making those elements smaller but more frequent — they’re able to accomplish the same level of compliance. And now instead of 60 days after code complete, it’s a lot fewer, even sometimes as few as zero incremental days.

The DevOps for Devices Maturity Model

Putting the five keys described above into practice is a journey, not a static destination. Device innovation is never complete. Hardening security is never complete. And there are always ways to optimize and get more efficient, or continue to build and scale your fleet. As we've guided and partnered with customers through their journey, we've found it helpful to map it out on a maturity curve, so it's easy to see where they currently are in the journey, and it's clear what the next steps are.

The maturity curve comprises the different stages and typical behaviors and capabilities that teams experience as they progress. As companies grow along the curve through the stages, business value increases.



- UI/Console focused
- Basic Mangement
- Smaller # devices
- Off-the-shelf devices
- Basic monitoring
- Occational app deployment

- Scripting
- Groups & hierarchies
- More devices
- Off-the-shelf devices and custom devices
- Basic monitoring
- Occational app deployment

- APIs/Scripts/Integrations
- Groups/Blueprints/Drift
- Very large # devices
- Custom devices
- Advanced Telemetry
- Predictive remediation of fleet
- At-will app deployments (security)

Early Stage:

- *Primarily UI or console-driven*
- *Siloed delivery teams*
- *Manual build, testing, and deployment*
- *Small number of off-the-shelf devices*
- *Basic management and basic monitoring*

Many customers that we talk to are here. They want to build out a device fleet, but are in the early phases, such as working on a proof of concept and probably with off-the-shelf devices. At that stage, it's still feasible to manually accomplish device management tasks. Often, this is done out of a web console.

Team Scale:

- *UI/Console and script-driven*
- *DevOps teams*
- *Automated and scaled operations through CI/CD*
- *Larger number of off-the-shelf devices*
- *Advanced monitoring*

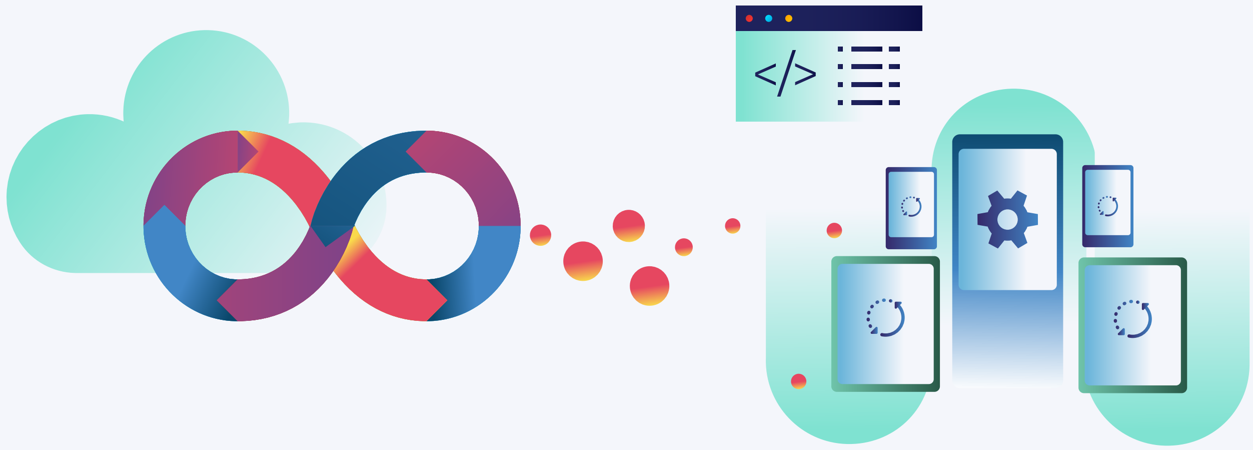
We also see a lot of customers who come to us here. These customers have devices already in the field and want to grow, but their manual processes and perhaps their existing MDM tools are slowing down their growth. They want to figure out how to automate more, script more, and get more advanced with things like monitoring and alerting.

Enterprise Scale:

- *APIs, scripts, and integration-driven*
- *Scaled DevOps teams*
- *Intelligent automation and canary (staged) deployments*
- *Very large number of custom devices (or mixed off-the-shelf and custom fleet)*
- *Advanced telemetry and anomaly detection*

This is where most customers ultimately want to get to. At enterprise scale, tools need to be integrated using APIs. You're pulling data into centralized data warehouses where you can do more advanced analytics, maybe even get predictive about where issues may occur. As your device volume grows, you probably also have a mixed fleet. Often at this scale it makes more sense to build custom devices so that you have total control from end-to-end.

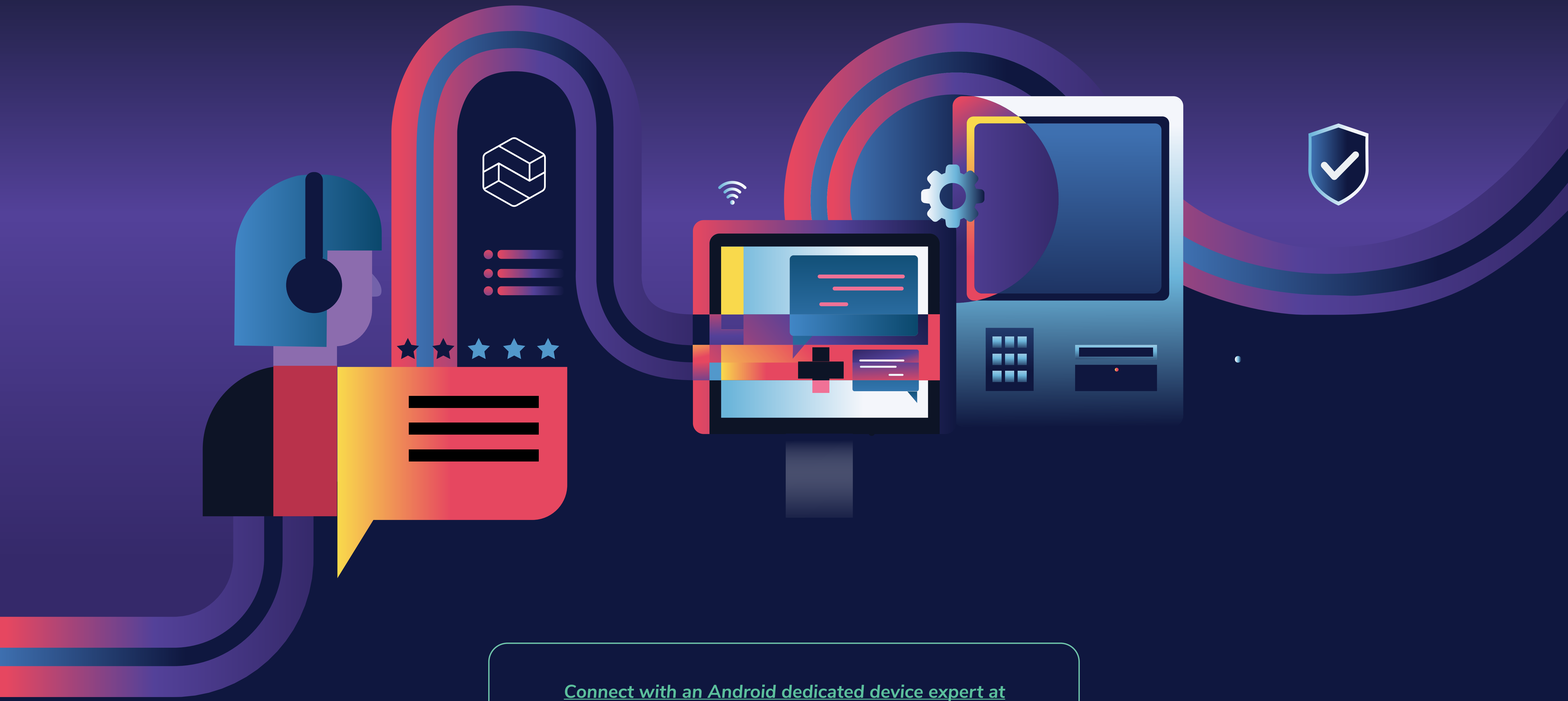
North Star:



The Intelligent stage is the last stage of the DevOps for Devices maturity model. When you are here, DevOps principles are embraced across the enterprise. You reduce incidents to very rare occurrences and can very quickly (in minutes) push fixes and updates to your systems. You can manage hundreds of thousands of devices seamlessly, you implement automated canary (staged) deployment methods, and you have advanced monitoring, forecasting, and anomaly detection processes. From app to hardware in the field, your infrastructure systems allow for frictionless updates and upgrades.

Most importantly, the time your teams save not doing manual coding, patching, or configuration is instead spent on customer thinking and innovation. And from your customers' perspective, little do they know how much work it took to scale up the DevOps program that made it possible to streamline everything from app development to management of your fleet of smart devices in the field; they just know that every interaction with your company is consistently great.

Getting here is a journey, not a destination. Every organization is at different stages along the DevOps for Devices maturity curve, and as you grow along the maturity model stages, your business value and your ability to deliver exceptional customer experiences increases.



[Connect with an Android dedicated device expert at Esper to see how we can help you build a strong foundation for your HealthTech device fleet.]

esper.io