carahsoft

Red Hat

Q&A: Building Platforms That Scale Without Slowing Innovation

Thank you for your interest in exploring this content.

Carahsoft is the **Trusted Government IT Solutions Provider**® supporting a broad portfolio of industry-leading technologies through GSA, NASA SEWP V, ITES-SW2 and a wide range of other contract vehicles.

As the **Master Government Aggregator**®, Carahsoft connects government agencies, industry partners, and technology providers to deliver innovative, mission-focused solutions.

In partnership with Red Hat, we provide technology solutions that drive modernization, strengthen operations, and ensure compliance with evolving government standards.

To learn more about how Carahsoft can support your technology needs, please visit carahsoft.com

Explore More Resources:
carah.io/RedHatResources

Join Events & Webinars:
carah.io/RedHatEvents

Discover Technology Solutions:
carah.io/redhat

Learn About Procurement:
carah.io/RedHatContracts

Connect With Our Team:
RedHat@carahsoft.com
877-RHAT-GOV

Q&A:

Building platforms that scale without slowing innovation

Red Hat

As agencies modernize their IT environments, many are discovering that adding more tools does not always lead to better outcomes. At enterprise scale, complexity can quickly outpace teams' ability to support, secure and evolve their systems.

Michael Hardee, chief architect for law enforcement and justice at Red Hat, works with public sector organizations navigating that challenge. Here, Hardee explains why platforms are essential for sustainable operations, how abstraction reduces cognitive load and what effective governance looks like when it is built in.

How do you define the difference between a collection of tools and a true enterprise platform, and what risks do organizations face when they optimize individual tools instead of the system as a whole?

Hardee: There's no silver bullet for technology, so integrations are necessary at some level. What matters is identifying where consolidation puts an organization in the best position, particularly around security, quality of service and support.

When tooling isn't consolidated and teams are working across too many disparate toolsets and systems, the overhead becomes immense. You end up with multiple support situations and growing dependence on vendors that may not have certified integrations or shared support models. At that point, what ends up happening is customers essentially become their own support organization, maintaining integrations that vendors may not fully stand behind.

That introduces a lot of toil. Maintaining a collection of tools pulls cycles away from innovation, and it often leads to shadow engineering. When environments sprawl, nonstandard practices start to emerge naturally — one too many integrations, one too many configuration points. While autonomy is important for operational and development teams, organizations still need guardrails to prevent things from running off the rails.

Why is abstraction essential for operating at enterprise scale, and how does it change the cognitive load placed on developers and operators across teams with different maturity levels?

Hardee: Most of our customers are in the cloud in some capacity, and when you go to a cloud provider, they tell you not to worry about what's behind the curtain. That same experience should exist for your end users.

Developers should be focused on building and driving innovation, not worrying about tooling, resources, or processes. They should be able to log into a platform interface and stay focused on the task at hand. because all of the unnecessary cognitive load has been offloaded to the platform.

For administrators, abstraction helps limit blast radius. Not everyone needs access to every interface, because that introduces serious security risks. When everyone has admin access everywhere, configuration drift becomes inevitable. Leveraging approved workflows and enterprise engineering practices help prevent backdoor engineering, favors between teams and inter-program politics from driving technical decisions.

When you pair abstraction with automation you can make advanced actions available to a broader user community. It empowers users to innovate while freeing senior engineers from constant operational toil. The technology exists to build self-reliant organizations through automation, [now] it's about making that investment.

Why does integrating capabilities often unlock more value than the sum of individual components — the "1 + 1 = 3" effect?

Hardee: It's all about being deliberate. Too often, a new trend emerges and organizations rush to adopt it without asking whether there's an ctual requirement.

Instead of piecemealing together solutions, organizations need to plan their work and work the plan. Strong integrations accelerate outcomes because the vendor ecosystem exists for a reason — cloud providers, network providers and storage providers are designed to work together.

Being extensible is a requirement. Developer's and engineer's taste in tools change over time. I encourage organizations to work with vendors that lead with their partner ecosystem. That's how you build long lived solutions and platforms.

What does effective, "invisible" governance look like in a well-designed platform, and why do guardrails fail when they're bolted on instead of built in?

Hardee: It starts with requirements, organizationally we need to be operating on the same information. That question needs to be asked across stakeholders and everyone needs to be at the table to understand each other.

Successful governance shows up in adoption. When people want to use your platform, it's a sign you've built effective onboarding and clear processes. That requires compromise, meaning nobody gets everything they want, but everyone agrees.

Enterprise platforms have to serve a diverse set of programs with different needs. The tooling exists to support off-ramps for nonstandard requirements while still maintaining guardrails. That means being able to adjust enterprise posture without putting the broader organization at risk.

If governance isn't in service to both the enterprise and the user community, users will find their own way around it, which ultimately weakens your security posture.

At enterprise scale, why is operational consistency often more valuable than local optimization? How do platforms help organizations move from heroic operations to repeatable outcomes?

Hardee: Organizations that prioritize abstracting micro-level complexity into higher-order platform capabilities position themselves for scalable, repeatable success. Operational consistency is only possible when standards have been defined, which is what a platform is at its core!

Those standards are going to empower you to create a repeatable organization. At the end of the day, I want a platform that sets every facet of my organization up for success.Weekend-long incident calls usually happen when something was built that was too exciting and outside the team's support capacity. Where technology stands today, there's no reason to rely on heroism if you build a sustainable, well-designed enterprise.

Beyond adoption or usage metrics, how should leaders evaluate platform success — particularly outcomes the platform enables but does not directly own?

Hardee: Adoption is the metric. When users come back and say they like the platform we have built together and have ideas about more things they want to do with it, that's success.

Beyond adoption leaders need to look at performance and not just that of their internal team but the end user. Technology's purpose is to abstract and enable. Are my teams able to service the platform and my user community with ease? Is my end user able to deliver mission capabilities at a velocity of their choosing? Is my platform serving as a hurdle or a highway?

The goal is to reduce overhead so the organization isn't constantly holding things together through extraordinary effort.

Technology is almost certainly guaranteed to keep changing and evolving. What should agencies invest in today to avoid long-term architectural regret?

Hardee: Pivoting away from the technology and first looking at the organization itself. Leaders need to understand their current team composition, what they can support today, and what they want their organization to look like five to 10 years from now.

There's a lot to consider there, including funding, mission changes and evolving requirements. What capabilities are we lacking today that we need tomorrow? And work with industry to figure out the most effective way to get there.

Organizations should engage vendors early, understand their road maps, hold them accountable and talk with industry experts about what's coming to see what future technology directions make the most sense with their mission needs.

Learn how **Red Hat can help your organization build secure, flexible platforms designed for long-term adaptability.**