WHITE PAPER

Hitachi Vantara

SRE Vs Traditional ITOps: What's The Difference?

By Hitachi Vantara

Table of Contents

Introduction

Amidst recent technological advancements and shifts in macroeconomic conditions, organizations are rapidly pivoting towards digital transformation. With a surge in cloud-centric workloads, traditional IT Operations (ITOps or AMS) often grapple with managing these modern tasks due to their conventional methodologies. They face challenges such as:

- Managing the complexity and distributed nature of modern applications.
- Meeting rigorous reliability and uptime standards.
- Gaining clear insights into application performance.
- Navigating a multitude of tools that slow recovery during disruptions.
- Tackling inefficiencies with siloed teams and skills.
- Relying on a people-driven support model that does not efficiently scale with rising demand.

What Is Site Reliability Engineering (SRE)?

As organizations accelerate their digital transformations, ITOps struggle to keep pace, paving the way for newer methodologies like SRE. SRE is a modern approach to IT operations, blending software engineering principles with operational practices to ensure application reliability, scalability, and optimal performance. Originated at Google, it is designed for organizations aiming for continuous improvement in today's fast-paced digital landscape.

ITOps, DevOps, and SRE may share some common elements, but each occupies a distinct role with unique responsibilities in the IT landscape. While ITOps primarily manages IT functions, SRE builds upon this foundation, emphasizing service improvement and reliability. Differing from DevOps, SRE seamlessly integrates objectives from both software engineering and IT operations to achieve holistic operational excellence.

What are the differences between ITOps and SRE?

While traditional IT operations and SRE teams may appear similar, SRE introduces distinctive practices that significantly enhance operational efficiency and effectiveness. The key differences primarily lie in the scope of work. Beyond core operational tasks, SRE incorporates:

01 **Shared Reliability between Dev & Ops:** Joint accountability between Development and Operations through Service Level Objectives (SLOs) and Error Budgets.

02 **Proactive Monitoring:** Comprehensive observability for early problem detection and swift recovery.

03 **Engineered Resilience:** Built-in fault tolerance to improve overall reliability.

04 **Deployment Readiness:** Production Readiness Reviews (PRRs) ensure seamless rollouts.

05 **Incident Management:** Quick recovery strategies and a Directly Responsible Individual model

06 **Learning Culture:** Blameless postmortems to encourage learning from incidents.

07 **Cost Efficiency:** Integrated cloud cost management within operational duties.

The next **seven sections** delve deeper into these key differences between SRE and ITOps:

Shared Reliability between Dev & Ops

In traditional IT operations, development and operations teams often work in isolated silos. Developers prioritize releasing new features, sometimes overlooking application stability, while operations aim to maintain uptime without clear visibility into incoming releases. This disconnect can lead to conflicting goals, jeopardizing the reliability and availability of services in production.

SRE revolutionizes this dynamic by fostering shared responsibility between the two teams through Service Level Objectives (SLOs) and Error Budgets. An SLO, a keystone of the SRE approach, is more stringent than the typical Service Level Agreement (SLA). While an SLA formalizes the agreement between the service and its users, an SLO internally sets high availability targets. For instance, a 99.9% availability SLO translates to a permissible 43 minutes of downtime monthly, while 99.99% allows only 4 minutes. See figure 1.

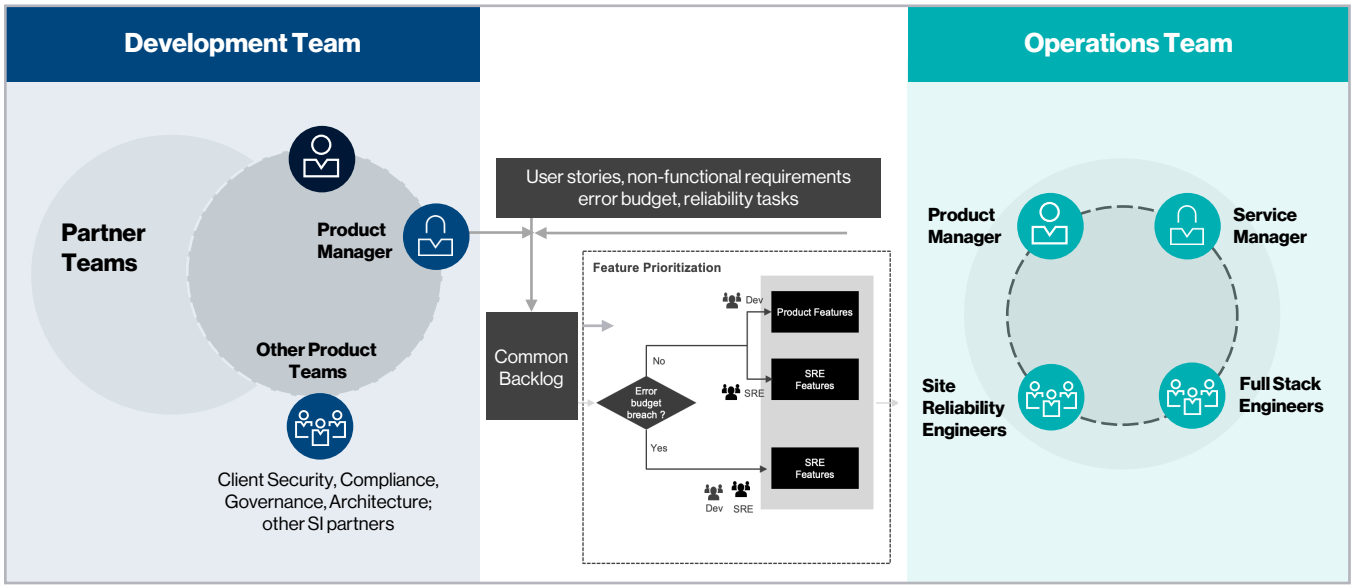

Figure 1: Shared Reliability between Dev & Ops

In simple terms, an Error Budget is like a time allowance for when things can go wrong without impacting the customer. It gives your development team a set 'budget' of allowable hiccups. They can use this 'budget' to try out new features or make updates. If things go wrong too often and you use up this budget, all new updates are paused. The development and operations teams then work together to fix the issues and make the service more dependable. This approach is a win-win: it keeps your tech teams (dev and ops) accountable while also giving them room to innovate.

Proactive Monitoring

Monitoring and Observability are one of the core elements that SREs are responsible for. In traditional IT Operations, we often look at a set of pre-defined data to spot and fix issues. The SRE approach, however, goes a step further to enhance the user experience. SREs aim to give a full picture of system health and performance in real-time, alerting you immediately to problems that need attention.

- Complete visibility into IT system/application performance with unified dashboards
- Configured metrics, logs, and events for prompt issue detection and swift recovery.
- Insightful alerts and notifications highlight immediate system concerns.

By 2024, 30% of enterprises implementing distributed system architectures will have adopted observability techniques to improve digital business service performance, up from less than 10% in 2020

- Gartner.

To address the objectives, SREs often work towards a solution that brings all relevant monitoring metrics, logs, and events from disparate set of tools into a common repository, often a timeseries database to enrich the data, correlate and gain better insights into IT systems behaviors as follows. See figure 2.

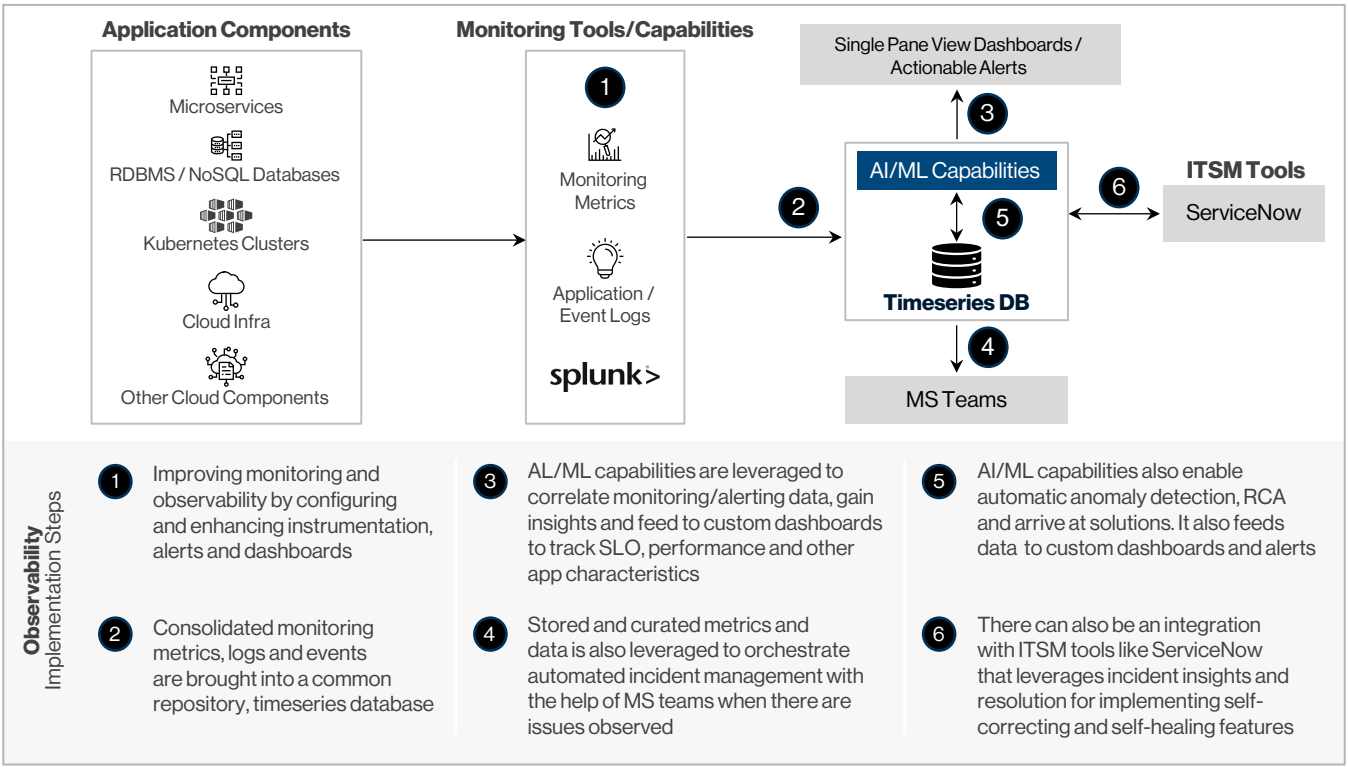

Figure 2: Proactive Monitoring

Instead of using multiple tools, SREs bring all the information into one place, helping the IT Team get a clear picture and make quick decisions. With the insights they gain:

- You get clear visuals and alerts about your system's health.
- Issues are quickly communicated through popular platforms like Microsoft Teams and Slack.
- These insights even talk to tools like ServiceNow, generating tickets that lead to fast solutions and built-in measures to prevent future problems.

What does this mean for your business? You get quicker warnings when something is not right (improved time to detection) and solve problems faster when they do occur (improved time to recovery). This initiative-taking approach lets you spend less time firefighting and more time driving the business forward.

Engineering Resilience

In today's fast-paced digital environment, think of our systems as a vast city grid, bustling with intersections and cross-traffic. SRE acts as our advanced traffic management system, ensuring that even during peak hours or unforeseen events, traffic flows smoothly, safely, and efficiently. One standout feature of SRE in comparison to traditional IT operations is how it deals with potential system hiccups, especially as we incorporate more complex structures, akin to introducing intricate junctions and multi-level highways into our city grid. See figure 3.

To ensure we do not just react to challenges but proactively prepare for them, SRE emphasizes 'Engineering Resilience'. Here is a simple breakdown:

1. **Planning for Challenges:** Like city planners anticipating high-traffic zones, we identify where our systems might falter and design them to be alert and adaptive.
2. **Building Robust Infrastructure:** Just as we would design roads to manage various vehicles and weather conditions, our systems are tailored to function optimally, regardless of potential IT challenges.
3. **Testing our Defenses:** Before launching, we run our systems through a series of scenarios — much like mock traffic situations — ensuring they're equipped to handle real-world challenges seamlessly.
4. **Constant Vigilance:** Our systems come with built-in monitors, like traffic cameras, ready to detect issues and initiate backup plans when necessary.

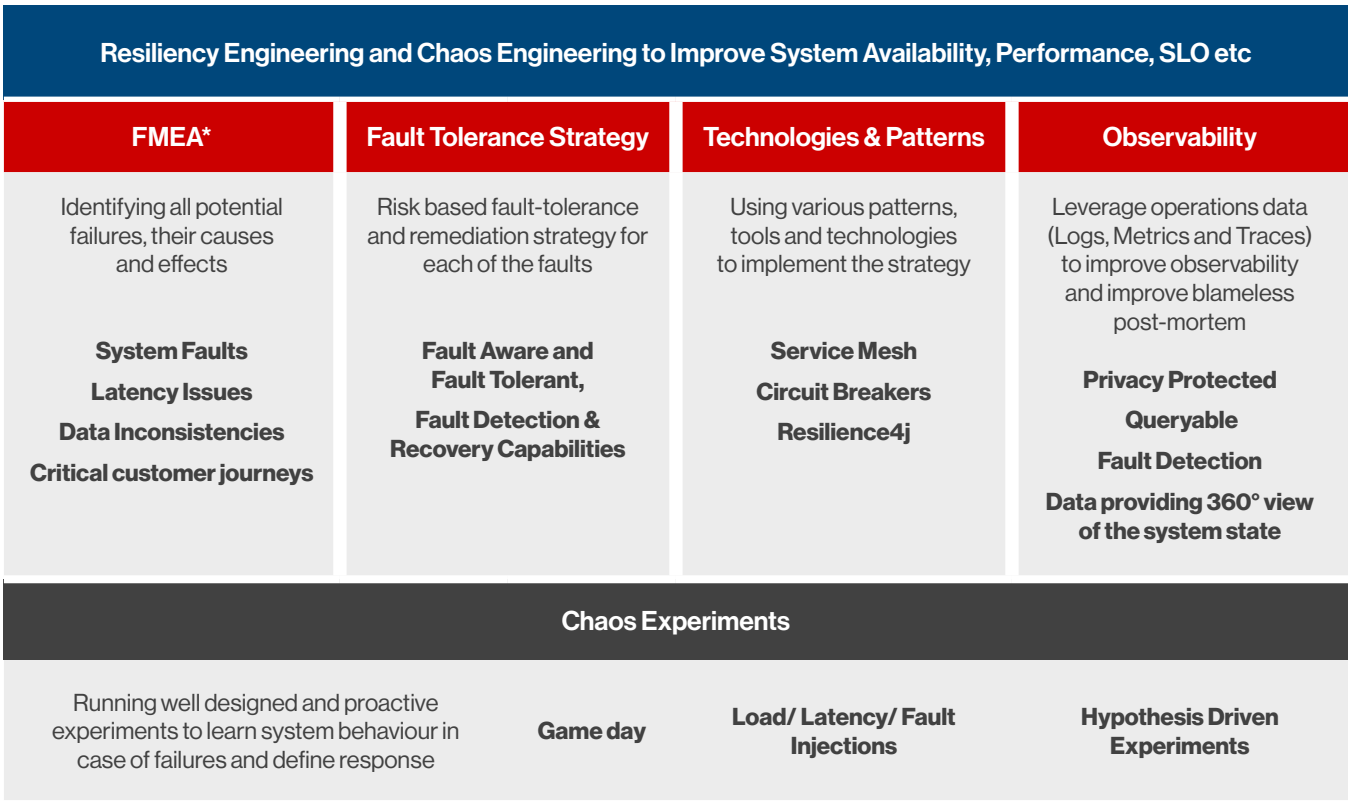

Figure 3: Engineering Resilience

*FMEA – Failure Mode Effects and Analysis

Deployment Readiness

Imagine you are about to launch a brand-new, state-of-the-art car model. You would not just assemble the car and send it off to the dealerships. First, you had run a series of thorough tests: safety features, fuel efficiency, ease of repair, and so on. That is like what SRE does through its 'Production Readiness Review' (PRR) before any software or update goes live.

So, what is PRR? Think of it as a 'final quality check' that looks at various aspects to ensure smooth operation once the application is up and running. Here is what it covers:

1. **Blueprint Review:** Just as you would scrutinize car blueprints for any structural issues, SRE ensures the software's architecture is robust and resilient, prepared for any challenges it might face.
2. **System Checkup:** Before a car hits the road, you should check its various indicators and sensors. Likewise, PRR ensures that we have proper monitoring tools in place so we can catch and fix issues before they impact the users.
3. **Emergency Protocols:** Much like having a spare tire and a toolkit in a car, PRR makes sure there are set procedures and resolute teams.
4. **Resource Allocation:** Think of this as ensuring that the car has enough fuel and horsepower for the journey ahead. SRE verifies that we have enough computational power to manage expected user demand.
5. **Smooth Transitions:** Before any car model goes to market, you would check how easily it can be modified or upgraded. Similarly, PRR ensures that any future changes to the software can be rolled out seamlessly, with minimal impact on users.
6. **Performance Tuning:** Finally, you would want to test the car under various road conditions and speeds. SRE does this by putting the software through rigorous performance tests to ensure it meets all quality benchmarks.

In a nutshell, the PRR process makes sure that by the time a software application is ready for public use, it's as reliable, efficient, and user-friendly as possible. It's not just about preventing problems; it's about anticipating them, so the ride is smooth from start to finish."

Sample PRR Checklist

Sl.No	Category and questions	Responses
1	**Service Definition and Goals**	
1.1	Describe what your service does from the customer's point of view.	
1.2	Describe your operational goals for the service.	
1.3	What is the SLA of the service?	
2	**Dependencies**	
2.1	What are this microservice's dependencies?	
2.2	What are its clients?	
2.3	How does this microservice mitigate dependency failures?	
2.4	Are there backups, alternatives, fallbacks, or defensives catching for each dependency?	
3	**Routing and Discovery**	
3.1	Are health checks to the microservice reliable?	
3.2	Do health checks accurately reflect the health of the microservices?	
3.3	Are health checks run on a separate channel within the communication layer?	
3.4	Are there circuit breakers in place to prevent unhealthy microservices from making requests?	
3.5	Are there circuit breakers in place to prevent production traffic from being sent to unhealthy hosts and microservices?	
4	**Scalability and Performance**	
4.1	What is the microservice's qualitative growth scale? (Orders or Users etc)	
4.2	What is the microservice's quantitative growth scale? (RPS/TPS)	
4.3	Is the microservice running or dedicated or shared hardware?	
4.4	Are any resources abstraction and allocation technologies being used?	
4.5	What are the microservice's resource requirements (CPU,Ram,etc.)?	
4.6	How much traffic can one instance of the microservice handle?	
4.7	How much CPU does one instance of the microservice require?	
4.8	How much memory does one instance of the microservice require?	
4.9	Are there any other resource requirements that are specific to this microservice?	
4.10	What ere the resource bottlenecks of the microservices?	
4.11	Does this microservice need to be scaled vertically, horizontally, or both?	
4.12	Is capacity planning performed on a scheduled basis?	
4.13	Are there microservices given priority when hardware requests are made?	
4.14	Is capacity planning automated or is it manual?	
5	**Dependency Scaling**	

Incident Management

In traditional IT operations, the process of managing incidents often feels like an elaborate relay race. An issue starts at Level 1 (L1) and is gradually passed up to Level 2 (L2), Level 3 (L3), and so on until it reaches the expert who can resolve it. This step-by-step escalation not only slows down problem resolution but also comes at an excessive cost—lost business and frustrated customers. See figure 4.

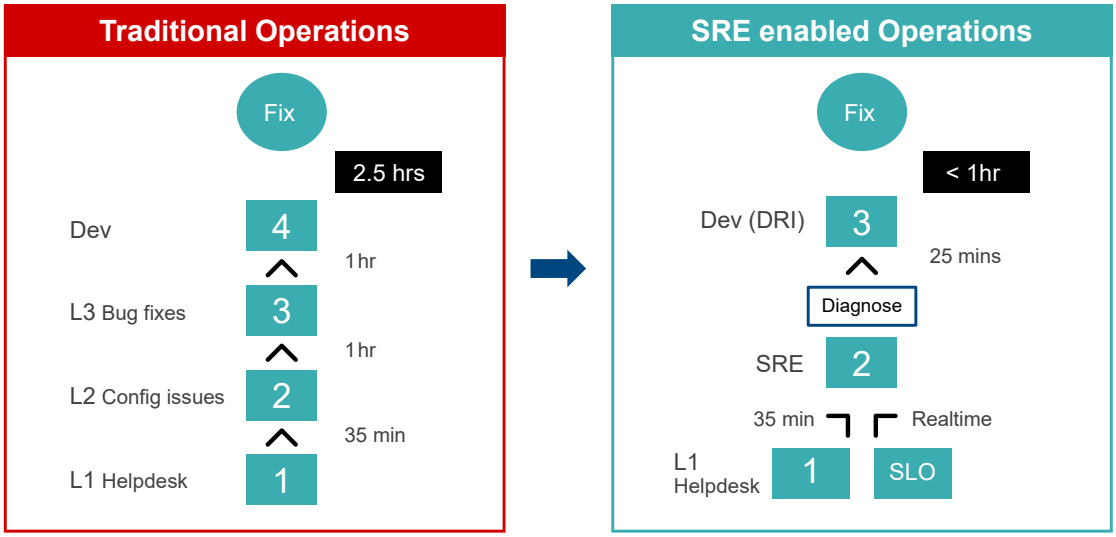

Figure 4: Incident Management

In contrast, SRE operates like an expert SWAT team, agile and empowered. Equipped with sophisticated monitoring tools and a wealth of data, SREs have an eagle-eye view of system health. They are trained to identify the root cause of an issue quickly and send it directly to the designated responsible individual (DRI) who can resolve it. No more step-by-step delays—just immediate action enabled by technology and insights.

The result? Rapid problem resolution, minimized downtime, and a smoother, more reliable experience for our customers. SRE does not just patch up issues; it strategically elevates the robustness of our operations, aligning perfectly with the kind of excellence our customers expect and deserve.

Learning Culture

In the fast-paced world of technology, failures will inevitably occur. What distinguishes SRE from traditional ITOps is its forward-looking culture of 'Learning from Failures'. Instead of assigning blame, SREs champion a process called 'blameless postmortem analysis'. This is not a finger-pointing exercise, but a constructive look at what went wrong, why, and how to prevent it in the future. These postmortem sessions are an opportunity for teams to come together, dissect significant incidents, and draw invaluable insights. Every incident becomes a lesson, ensuring it is documented, its causes understood, and measures put in place to prevent recurrence.

When do we deploy this postmortem strategy? It's implemented after notable incidents that lead to:

- Visible disruptions or slowdowns in user services.
- Any data losses.
- Need for on-call engineers to step in for recovery.
- Extended incident resolution times.
- Failures in monitoring or situations where incidents are flagged manually.

Cost Efficiency

Though In the evolving landscape of IT operations, the SRE role is expanding to embrace not just reliability but also cost-effectiveness. Traditionally, financial oversight of cloud operations—known as FinOps or Cloud Cost Management—was separate from SRE responsibilities. However, in today's cloud-centric environment, the performance and efficiency of applications in production are intertwined with cloud expenses. Thus, ensuring cost efficiency in cloud operations has become a focal point for SREs.

71% of organizations expect their cloud spend to increase by 2024

- Gartner.

Traditionally, Cloud Cost Management was often a reactive process, reliant on tools like Cloudability and CloudHealth, and retrospective reviews of billing statements. The architecture of applications, which significantly influences cloud costs, typically did not receive the attention it deserved. See figure 5.

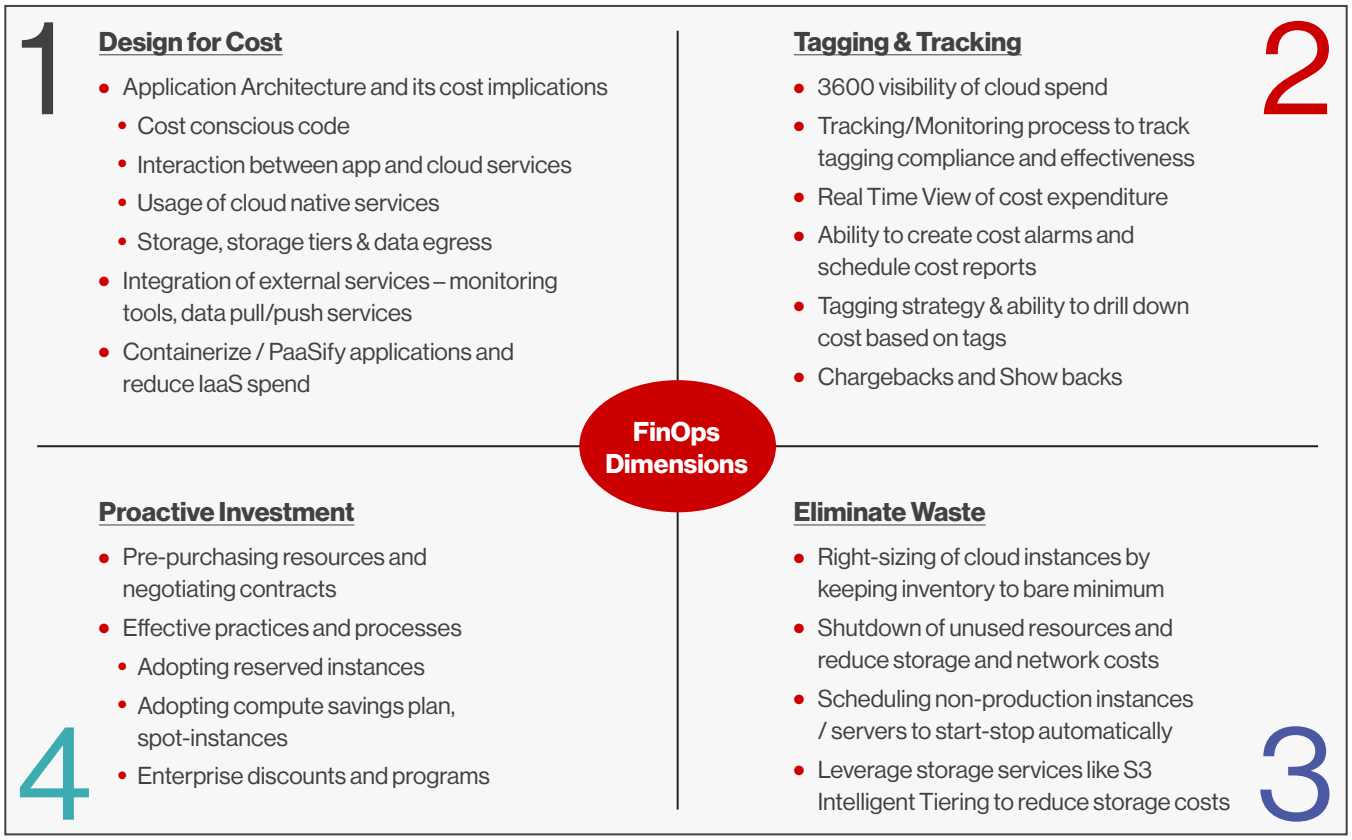

Figure 5: Cost Efficiency

SRE brings a proactive approach to this domain, focusing on four pivotal dimensions:

- **Designing for Cost:** It is not just about how an application works, but also how cost-effective its operation is. By understanding the financial implications of architecture choices, SRE ensures the most cost-efficient cloud services are harnessed. This includes minimizing costs related to data transfers and outbound data.
- **Cloud Cost Visibility:** Knowledge is power. With heightened observability and meticulous tracking, SREs provide a transparent view of cloud consumption and costs. They facilitate chargebacks and show backs, creating clear accountability for expenses.
- **Eliminating Waste:** Unused or unnecessarily large cloud resources are akin to leaky faucets. SREs ensure the right number of resources are deployed—no more, no less—eliminating wastage and ensuring optimal provisioning.
- **Proactive Investment:** Think of this as buying in bulk. By purchasing cloud resources in advance or consolidating accounts, significant discounts can be secured, leading to substantial savings.

32% of a cloud spend went to waste in 2022, an increase by 2% from 2021

- Gartner.

Summary

In the digital era, where agility and reliability are paramount, traditional IT Operations often lag the demands of contemporary, cloud-based systems. Enter SRE, a paradigm that marries software engineering principles with operational best practices. While traditional IT focuses on maintaining the status quo, SRE prioritizes resilience, rapid incident resolution, and cost efficiency. This whitepaper delves into how SRE emerges as a strategic choice for modern enterprises, enhancing agility, robustness, and financial sustainability compared to the legacy operational frameworks.

About Hitachi Vantara

Hitachi Vantara, a wholly owned subsidiary of Hitachi Ltd., delivers intelligent data platforms, infrastructure systems, and digital expertise that supports more than 80% of the fortune 100. To learn how Hitachi Vantara turns businesses from data-rich to data-driven through agile digital processes, products, and experiences, visit www.hitachivantara.com.

WE ARE HITACHI VANTARA

Hitachi Vantara solves digital challenges by guiding you from what's now to what's next. Our unmatched industrial and digital capabilities benefit both business and society.

Learn More →

Take the next step and create a Center of Excellence for a holistic view of your entire IT environment. Explore the solutions and services Hitachi Application Reliability Centers delivers.

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA
hitachivantara.com | community.hitachivantara.com

Contact Information
USA: 1-800-446-0744
Global: 1-858-547-4526
hitachivantara.com/contact