



SUSTAINABILITY AND PLATFORM ENGINEERING FOR THE FUTURE

Abstract

The imperatives of global sustainability demand a paradigm shift in the way businesses operate. This document elucidates the intricate relationship between sustainability and platform engineering, offering an exhaustive examination of advanced engineering practices capable of steering organizations towards positive environmental and social impact.

The Algorithmic Footprint: AI, the digital oracle of our times, is hungry for data and power. Training these behemoths consumes vast amounts of energy.



In the last 170 years, we added 2.4 trillion tons of Carbon dioxide into our atmosphere. Half of this was added in the last 35 to 50 years.

Source: Georgetown University



Microsoft's water use increased by 34% as a result of AI technologies, and one city worried about the impact on residential supplies due to its data centers.

Source: Fortune.com

Amidst accelerating global change, sustainability has risen from ambition to essential. As we strive for a greener future, companies across industries embrace green practices within their core operations, and platform engineering stands out as a fertile ground for sustainable innovation. This paper delves deep into the principles, strategies, and technologies that empower organizations to seamlessly weave eco-conscious practices into their platform strategies, incrementally paving the way for a sustainable future.

Sustainability and its Importance

Sustainability is a guiding light. It demands responsible resource usage for present needs without stealing from the future. Balancing economic strength, social fairness, and environmental care, it recognizes their intertwined existence in building a thriving world. Climate change, resource depletion, and environmental ruin amplify the need for several actions. More than just a moral must, sustainability fuels business resilience in a shifting world.



Figure 2 | Importance of sustainability

[150 Facts for 50 Years | Sustainability at Georgetown University | Georgetown University](#)



From a strategy and business perspective, embracing sustainability is more than a feel-good choice; it is a strategic win-win. It aligns with UN goals, boosts efficiency and responsibility, and unlocks economic and reputational rewards. Collaboration is vital to tackling challenges and building a thriving and sustainable future, from governments to businesses to citizens.

Enterprise sustainability is not a monolithic goal but a kaleidoscope of interconnected facets crucial for long-term success, especially in the complex world of software and IT. From technology choices to operational processes, from optimizing data center energy consumption to crafting eco-conscious code, myriad avenues exist for these organizations to improve their sustainability and actively contribute to a greener future. Figure 2 depicts the key pillars of sustainability for software and IT organizations.

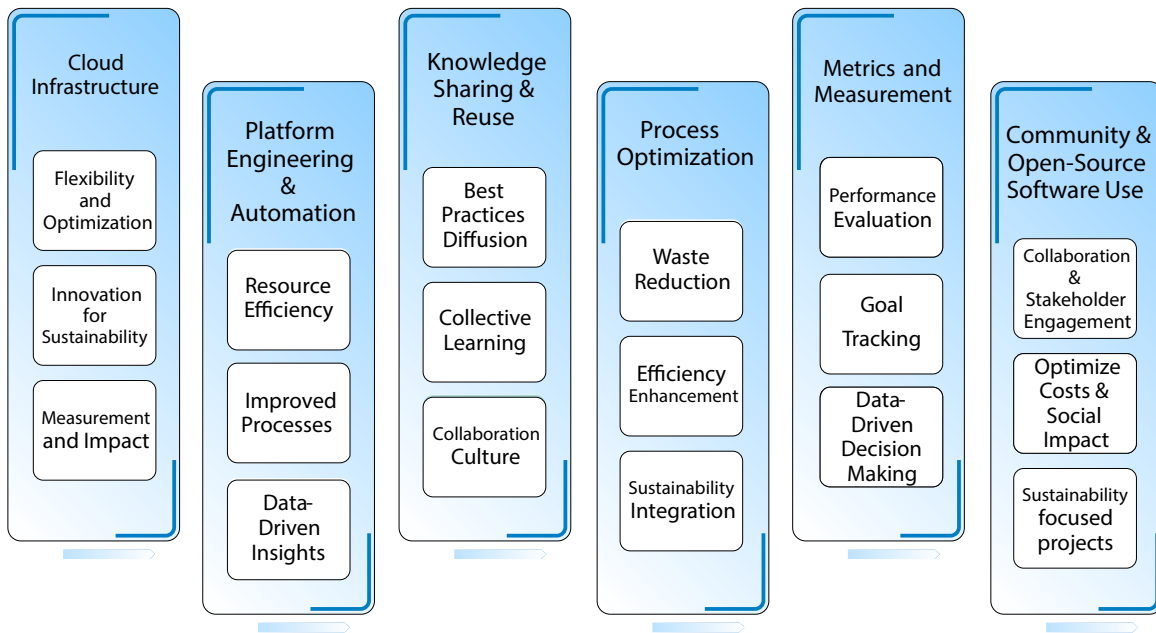


Figure 2 | Key pillars of sustainability

²A.I. usage fuels spike in Microsoft's water consumption | Fortune



Platform Engineering

The software industry also contributes to climate change.

According to the Intergovernmental Panel on Climate Change (IPCC), the information and communication technology (ICT) sector is responsible for 4% of global greenhouse gas emissions and is expected to grow to 14% by 2040.

[Source: https://ciandt.com/ca/en-ca/article/climate-crisis-and-technology-sector](https://ciandt.com/ca/en-ca/article/climate-crisis-and-technology-sector)

The technology industry can reduce its environmental impact in several ways, including embracing the circular economy, investing in eco-friendly products, reducing recycling, and extending the lifecycle of devices. Given the rapid growth of the number of applications and platforms being developed, one way is to adopt sustainable platform engineering practices. Platform engineering can help reduce energy consumption, minimize waste, and improve operational efficiency.

- Platform engineering is the discipline of designing, building, and operating reusable software platforms, which empower developers to create and deploy apps with lightning speed.
- Evolving DevSecOps & SRE: A holistic platform for product development, empowering the entire business.

Platform engineering is the bedrock of digital ecosystems, involving designing, constructing, and maintaining scalable, resilient and efficient digital infrastructures. It encapsulates numerous technologies, methodologies, and best practices to create a foundation that supports diverse applications and services, providing the backbone for digital innovation, now fueled and driven by AI.

The core principles of platform engineering driving sustainable IT practices.

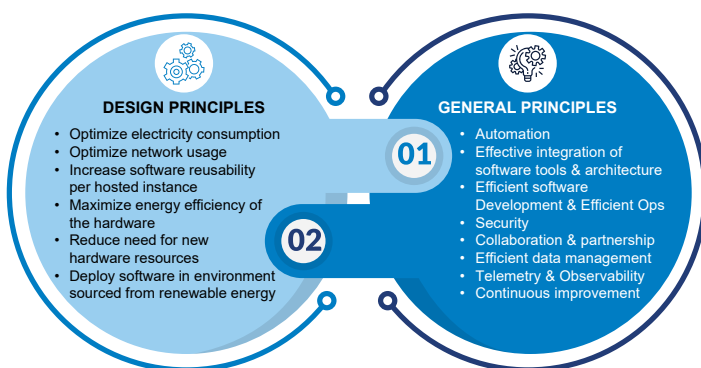


Figure 3 | Platform engineering core principles

Platform engineering is pivotal in realizing sustainability goals by providing the technical foundation for environmentally conscious practices. From optimizing resource utilization to facilitating the seamless integration of sustainable technologies, it is the catalyst for digital operations that are not only efficient but also ecologically responsible.

Key Components of Sustainable Platform Engineering

- **Green Software Development**
 - Green coding minimizes energy use, crafting eco-friendly applications with efficient practices
 - Legacy System Optimization
- **Renewable Energy Integration**
 - The transition to renewable energy sources is a cornerstone of sustainable platform engineering
 - Powering data centers with clean energy
 - Energy-efficient hardware
- **Circular Economy Principles**
 - Platforms designed for reuse. Build, repair, recycle: the circular future of digital.
 - Virtualization, DevSecOps and shared digital infrastructure
- **Emerging Technology Incorporation**
 - Cloud Computing: Cloud scales on demand, minimizing waste and maximizing efficiency
 - Microservices and Containerization: To improve scalability and flexibility, apply microservices architecture and containerization
 - Software-defined Infrastructure: Software-defined technologies to distribute resources dynamically in response to demand
 - Infusing AI/GenAI like a catalyst to ignite optimized performance, predictive intelligence and insightful breakthroughs
- **Security and Compliance**
 - Environmental Laws: Ensuring that environmental standards and laws about technology infrastructure are followed
 - Sustainability Certifications: Pursuing sustainability certifications and standards to show commitment towards ecologically friendly practices
 - Transaction and Data Security: Secure by design and preserving data and guaranteeing the confidentiality of information about sustainability



Figure 4 | Depicts how the core principles of platform engineering help in sustainability

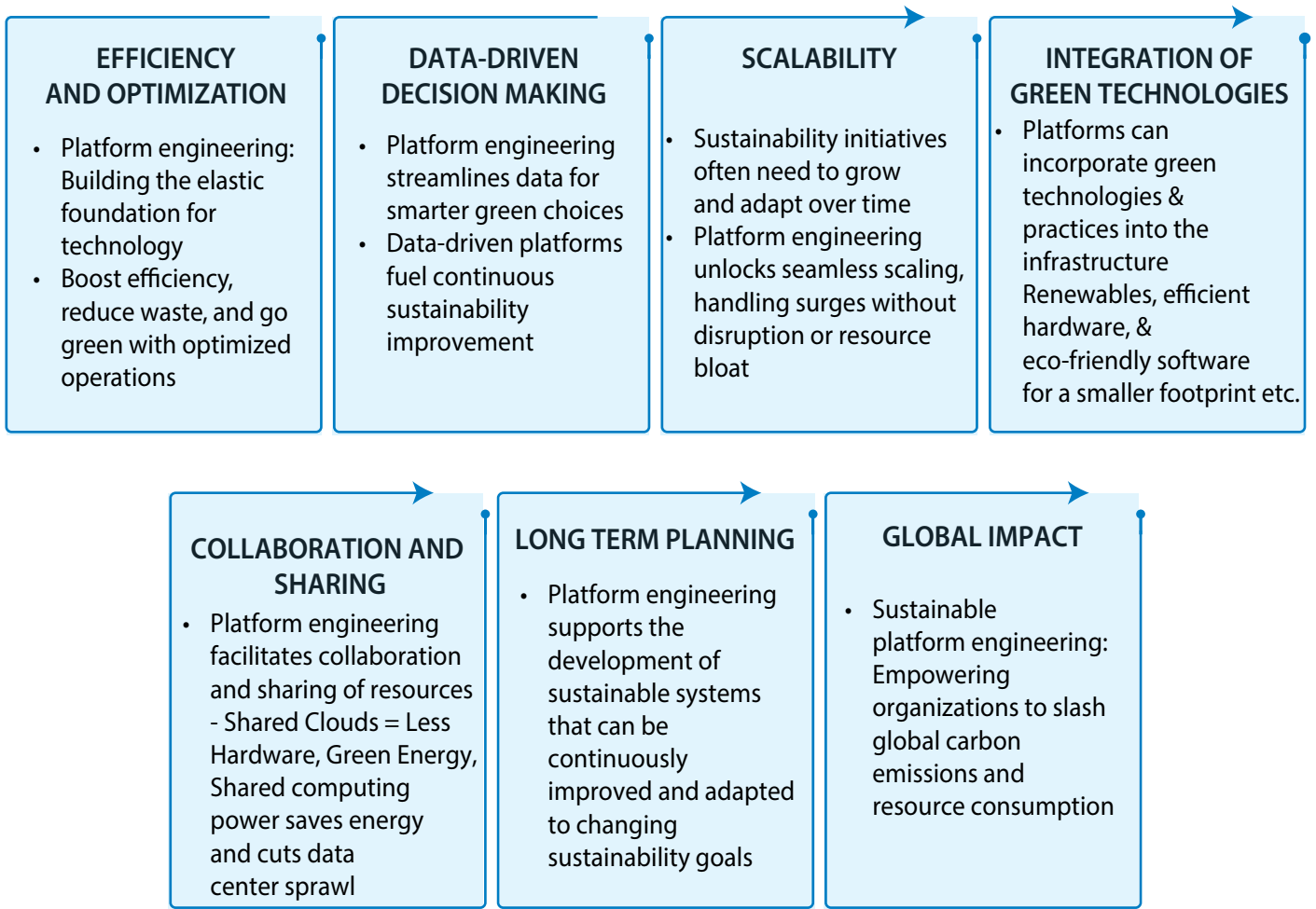


Figure 5 | Platform engineering driving sustainability

Shared Digital Infrastructure

Shared digital infrastructure is a key enabler of sustainable platform engineering. It refers to using common IT resources, such as data centers, networks, storage, tools, software, and platforms, by multiple organizations.

SHARED DIGITAL INFRASTRUCTURE

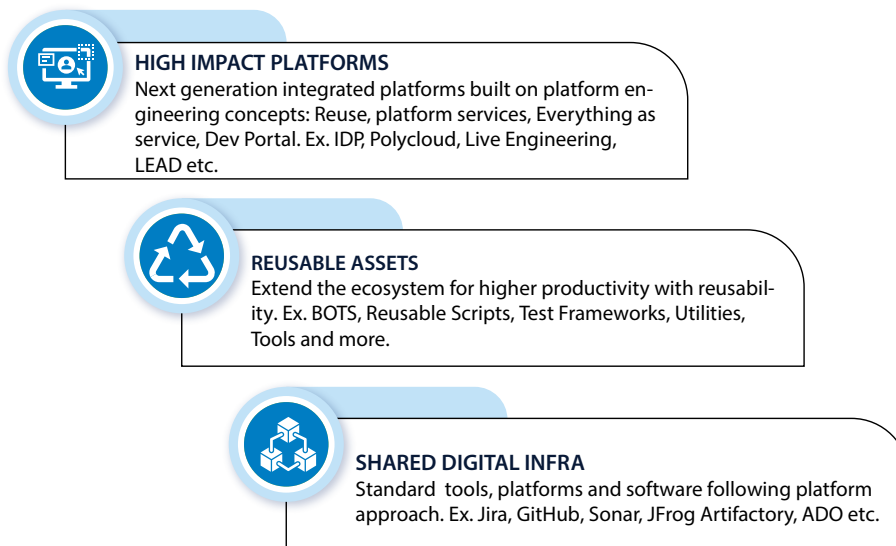


Figure 6 | Unified model for shared digital infrastructure

Shared digital infrastructure acts as a green tech library, fostering code, software, platform, and asset reuse, reducing redundant development, and minimizing resource consumption and environmental impact. It is like recycling for code, building future apps on the back of massive reuse, not reinventing while conserving precious resources for a sustainable future.

Shared digital infrastructure offers several benefits, such as reduced costs, superior quality, and improved efficiency and agility, making it a compelling option for organizations of all sizes. By carefully considering these benefits, organizations can make an informed decision about the right implementation approach.

Reusability

Reusability is a practice that extends beyond code and embraces a comprehensive approach to resource conservation. Embracing reusability is a philosophical shift towards a more sustainable IT ecosystem.

Code

- **Reduces redundancy:** Reusability minimizes resource usage during development, testing and deployment by eliminating repetitive code
- **Improves quality and maintainability:** Reusable code components are easier to update and manage and can be assured of quality, leading to sustainable software lifecycles and reduced maintenance costs

Components & Bots

- **Promotes modularity:** Building with reusable components fosters flexibility, allowing for easy integration and adaptation across projects, minimizing new development efforts
- **Standardizes development:** Reusable components establish common practices, leading to consistency, efficiency, high quality and reduced development time

Test Cases and Frameworks

- **Optimizes testing:** Reusable test cases and frameworks minimize redundant test creation and execution, saving time and resources
- **Improves test coverage:** Sharing and adapting reusable tests across projects ensures thorough testing without reinventing the wheel

Tools and Platforms

- **Extends lifespan:** By maximizing the use of existing tools and platforms, reusability reduces the need for new acquisitions, minimizes resource consumption and standardization, and significantly reduces sprawl and the associated costs
- **Enhances collaboration:** Shared tools and platforms facilitate knowledge transfer and collaborative development, leading to more efficient resource allocation

DevSecOps, SRE, Software Bill of Materials (SBOM)

DevSecOps, SRE and integrating SBOM creation in pipeline work hand-in-hand to drive sustainability in platform engineering by optimizing operations, amplifying feedback loops, enhancing security and efficiency, reducing toil, and responsible sourcing and resource usage.

1. Efficiency and Resource Optimization

- **DevSecOps Pipelines:** Minimize carbon footprint with reusable green pipelines: schedule builds for clean energy peaks
- **Automation:** DevSecOps principles like automation and infrastructure as code unlock efficient resource management, eliminating overprovisioning
- **SRE principles** help optimize infrastructure, team structures, and flow, reduce technical debt and human toil, reuse resources across portfolios, and drive regenerative capabilities across environments
- **Continuous Delivery:** Smaller, frequent deployments save energy and resources, with little requirement for massive energy-guzzling redeployments
- **Metrics and Monitoring:** Continuous insights help drive fine-tuned optimizations, eliminating resource waste

2. Shorter Feedback Loops and Reduced Rework

- **Collaboration:** DevSecOps fosters tight dev-sec-ops teamwork, removing silos, accelerating problem-solving and minimizing rework
- **Testing and Feedback:** Shift left, optimized test suite, and integrated testing minimize downstream costs by catching errors early
- **Deployment Pipelines:** Seamless deployments with built-in testing avoid expensive failures and rollbacks

3. Resilient Infrastructures

- DevSecOps safeguards platforms with secure coding, privacy by design and infrastructure as code, minimizing failures and resource drains from redeployments
- The DevSecOps methodology, combined with SRE, unifies the organization and streamlines processes to increase efficiency across the board

4. Software Bill of Material (SBOM)

- **Transparency and Accountability:** SBOMs map the software DNA, streamlining dependency and vulnerability management, cutting waste and patching redundancies
- **Sustainable Security Practices:** Outdated vulnerable components are identified early, preventing rework and costly breaches, thus saving time and energy
- **Responsible Sourcing:** Transparent and accurate software composition allows choosing ethically licensed, sustainably developed components that align with sustainability goals

Common Hurdles in Implementing Sustainable Platform Engineering

The path to sustainable platform engineering has its share of challenges, some of which include managing the complexity of platforms and distributed systems, consistent configuration management, effective monitoring and observability, integration challenges with legacy and API compatibility, automation complexities, vendor lock-in and choice of tooling, growing security considerations and compliance imperatives.

Overcoming these hurdles requires:

- **Strategic planning and architecture:** Design and document the platform architecture, considering scalability, security and operational needs
- **Cloud First, Automation First and AI-led** approach to stay ahead of the curve
- **Build for reuse and reuse before you build** as the core guiding principle for all platform components
- **Investing in tooling and automation:** Choose the right tools and automation of repetitive tasks to improve efficiency and consistency
- **Collaboration and communication:** Foster collaboration between platform engineering, development and operations teams
- **Proactive monitoring and security:** Continuously monitor the platform for vulnerabilities and implement robust security measures
- **Continuous learning and adaptation:** Stay updated on evolving technologies and adapt the platform as needed to maintain its effectiveness with active feedback loops

Best Practices and How to Implement Platform Engineering for Sustainability

1. **Modular Architecture:** Design the platform with modular, loosely coupled components for easier management, scalability and future updates.
2. **Automation:** Automate infrastructure provisioning, testing, deployment, and monitoring to reduce manual effort and improve efficiency.
3. **Reuse, Green components:** Consume a mix of open source, homegrown green assets and frameworks for development.
4. **Standardization:** Define and enforce standards for tools, coding practices and API usage to streamline development and ensure consistency.
5. **Security and Privacy by Design:** Integrate security and privacy considerations throughout the platform lifecycle, from design to deployment, with proactive threat detection and mitigation.

6. **Monitoring and Observability:** Implement comprehensive monitoring and observability tools to gain insights into platform performance and identify potential issues quickly.
7. **Platform Telemetry** drives feature incorporation in the roadmap and amplifies the feedback loops.
8. **Version Control and Patch Management:** Establish robust version control and patch management practices for platform components and dependencies.

Implementation Steps: A Three-Step Journey

1. Awareness: Laying the Foundation

- a. Empower yourself and your team with sustainable platform engineering: grasp its environmental, social, and economic impact and potential hurdles.
- b. Share insights with colleagues, leaders and stakeholders through talks and campaigns, and build buy-in for progress.
- c. Assess your platform for sustainability gains - from resource use to waste reduction.

2. Consciousness: Shifting Mindsets

- a. Embed sustainability in platform ethos, prioritize resource efficiency and long-term impact, not just functionality and speed, as part of the design and architecture.
- b. Unify Dev, Ops, and sustainability in a solution-driven team. Break silos, open communication, and bridge the gap.
- c. Track clear metrics (consumption, footprint, recycling) to measure progress and identify areas for sustainability



3. Execution: Taking Action

a. Make a shift to DevSusOps = DevSecOps + SRE + Sustainability. Actions from both the Development and Operations sides to become sustainable:

DEVELOPMENT	OPERATIONS
Choice of green, sustainable frameworks and languages as part of design Optimize Code, Reuse	Design for higher utilization
Faster languages and runtime selection	Implement automation
Architect efficient algorithms	Relax over-specified requirements
Design for faster implementation	Archive and delete data often
Implement appropriate logging	Deduplicate data
Reduce retries and work amplifications	Choose times and locations carefully

b. Implement renewables, optimize hardware/software, and automate tasks for reduced impact

c. Go green by design- Choose sustainable tools with built-in resource optimization, waste reduction and carbon tracking

d. Embrace continuous improvement. Continually refine the sustainability journey, adapt strategies and consider new opportunities

Measuring and Reporting Sustainability Impact

Establishing key performance indicators (KPIs) for environmental impact is foundational. Frequent monitoring and reporting on these metrics, coupled with transparent, automated ways of reporting, are essential in measuring and demonstrating the sustainability impact of platform engineering efforts.

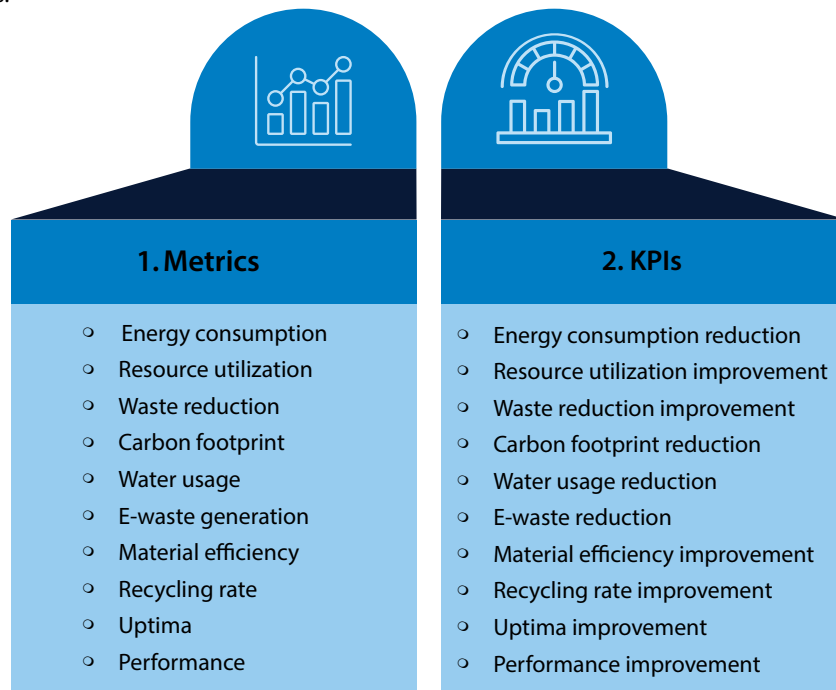


Figure 7 | Key metrics and KPIs

Infosys Story: Orchestrating Multiple Platforms with Green Practices

Infosys has become carbon neutral for FY2020, 30 years ahead of the timeline set by the Paris Agreement. In 2019, Infosys received the prestigious United Nations Global Climate Action Award in the "Climate Neutral Now" category. Imagine a world where software development not only churns out innovative applications but also does it without compromising environmental responsibility. This is the vision of sustainable platform engineering, a movement that extends green practices across the entire Software Development Life Cycle (SDLC), from ideation to deployment. Our green data centers are designed for carbon neutrality and water usage effectiveness to zero, leverage renewable power, have low power usage and are certified according to various environmental standards. Taking the platform approach and following platform engineering principles and practices, Infosys has made available multiple next-generation platforms for application development, maintenance, DevSecOps, SRE, AI First Development, Learning & Development, Sales, and operations.

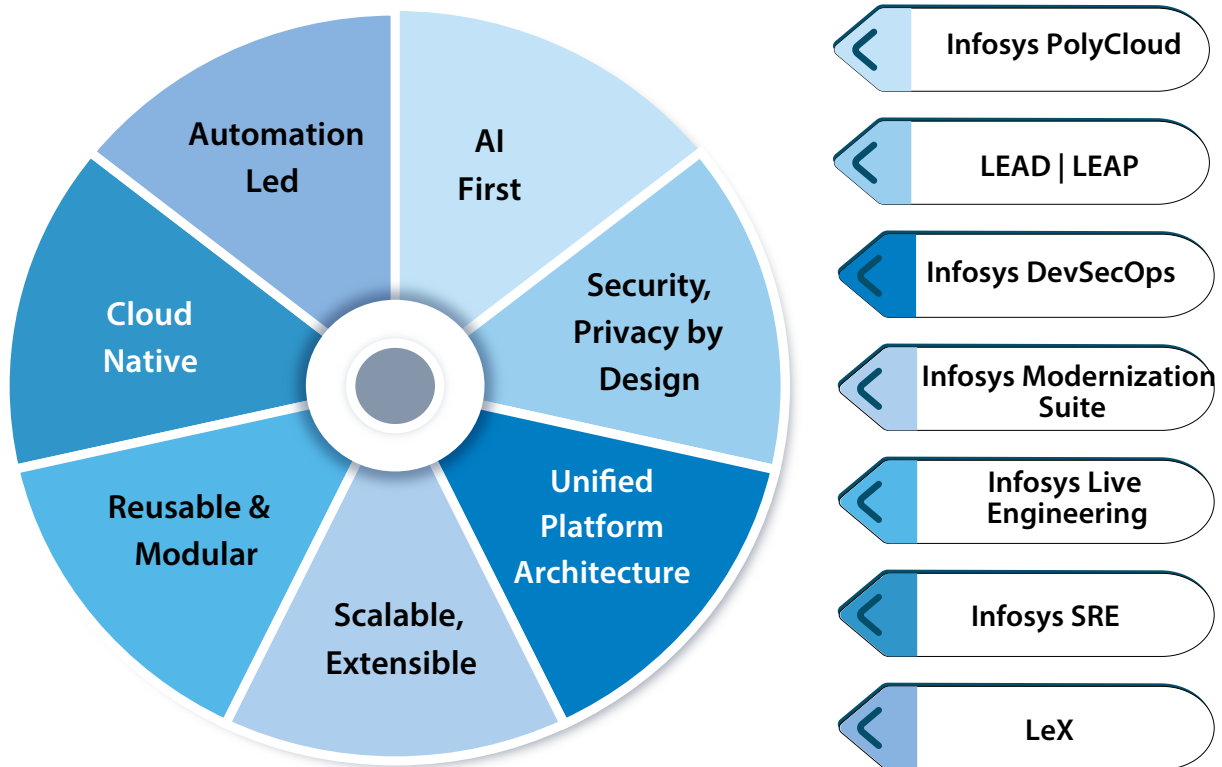


Figure 8 | Key platforms examples



Future Trends

The landscape of sustainable platform engineering is dynamic, with several anticipated trends shaping the future:

1. Green Infrastructure

- **Renewable energy:** Wind and solar power data centers: A sustainable shift led by platform engineers
- **Resource-efficient hardware:** Choosing energy-efficient servers, storage solutions, and network equipment is becoming a top priority to reduce carbon footprint and costs
- **Dynamic scaling:** Optimizing infrastructure to adapt to demand fluctuations, preventing unnecessary resource consumption during idle periods

2. Sustainable Development Practices

- **Green coding:** Code generation tools and static code analysis are used to write more efficient code, reducing memory usage and energy consumption
- **Green testing:** Using energy-efficient testing frameworks and minimizing unnecessary test runs lowers the environmental impact of software development
- **DevSecOps, SRE for sustainability:** Integrating sustainability into the entire DevSecOps lifecycle promotes continuous improvement and optimization
- **Reusing and refactoring code:** Prioritizing code reuse and refactoring to avoid unnecessary development and resource waste

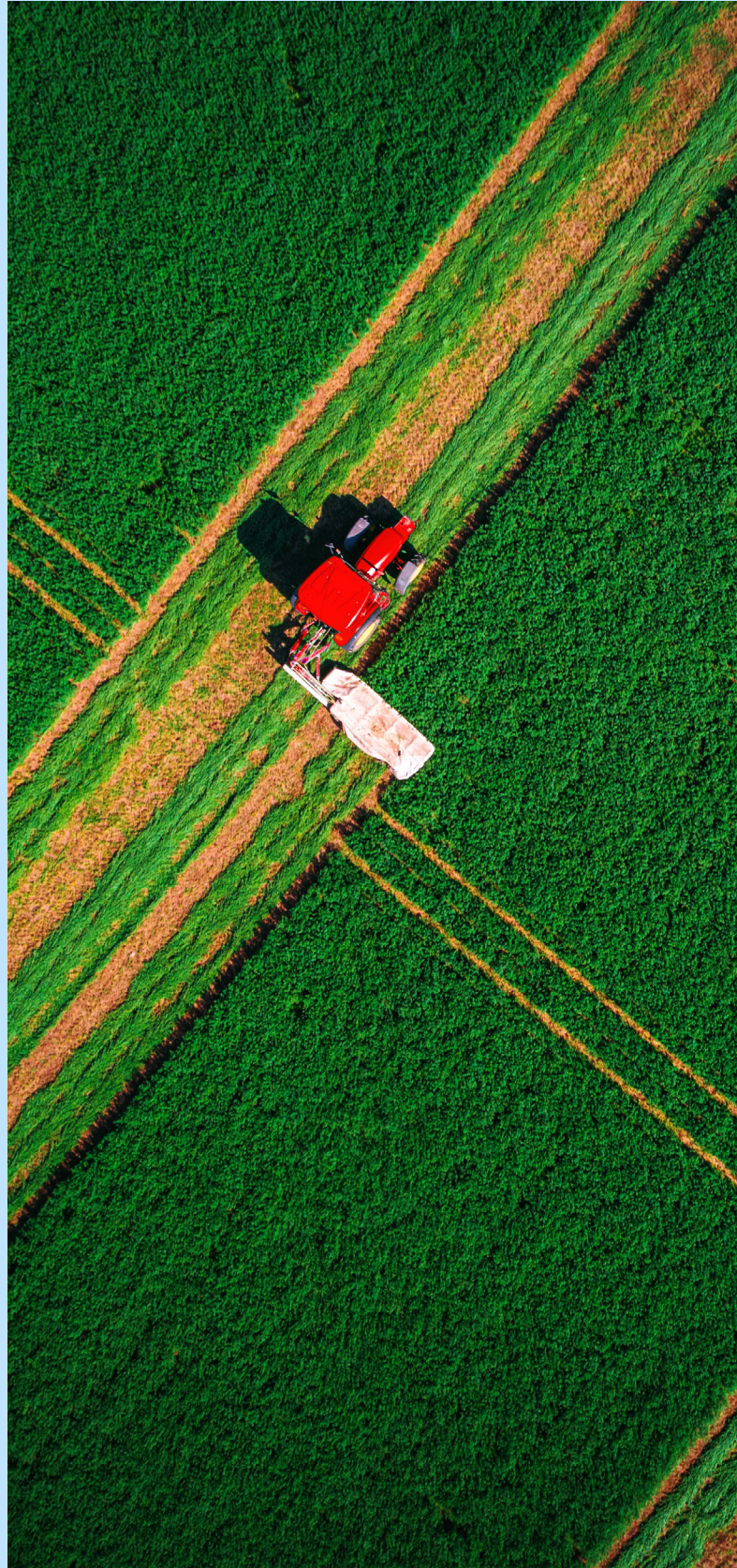
3. AI for Sustainable Platforms

- **AI-powered resource management:** Leveraging AI to optimize resource allocation, predict demand fluctuations, and automate energy-saving measures
- **Sustainability-aware algorithms:** Developing AI algorithms considering environmental impact during decision-making processes
- **Explainable AI for trust and transparency:** Ensuring AI-driven sustainability solutions are transparent and understandable to build trust and encourage further adoption
- **Green AI Development:** Optimizing AI models for efficiency, utilizing recycled data for training, and choosing energy-efficient hardware for AI training and deployment are becoming crucial considerations

4. The Rise of Green Platforms

- **Sustainability-first platforms:** New platforms are emerging that are specifically designed and built with sustainability in mind

- **Eco-conscious features:** Existing platforms add features like eco-mode or carbon footprint tracking to help users make sustainable choices
- **Partnerships for sustainability:** Platforms partner with sustainability-focused organizations to develop and implement innovative solutions



Conclusion

In conclusion, sustainable platform engineering is not merely a technological choice but a responsibility organization must embrace for the greater good. By aligning technological advancements with ecological considerations, businesses can contribute to a sustainable future while ensuring their resilience and competitiveness in an evolving digital landscape.

The journey towards sustainability is ongoing, marked by continuous innovation and a collective commitment to building a better, more sustainable world through platform engineering. As organizations navigate this path, integrating sustainable practices becomes not just a necessity but a defining characteristic of responsible and forward-thinking enterprises. The future of sustainable platform engineering is bright, and emerging technologies and approaches have the potential to make it easier and more efficient for organizations to build and operate sustainable software platforms. As these technologies and approaches continue to mature, we can expect their adoption to grow wider across organizations of all sizes. It's critical, though, to ensure that business priorities take center stage in driving the process changes and new ways of working to ensure sustainable practices, be it platform engineering-led or otherwise, are effective and serve the organizations in the long run.



About the Author

Naresh Choudhary

VP - Head – Enterprise Productivity and
Engineering Excellence, Infosys

Contributors

Mohammed Rafee Tarafdar, EVP - Chief Technology Officer, Infosys

Srinivasa Sujit Rao, Senior Project Manager, Infosys

Somasekhar Pamidi, Distinguished Technologist, Infosys

Mrinalini Dutt, Senior Consultant, Infosys

References

- <https://www.infosys.com/about/esg.html>
- <https://www.infosys.com/iki/perspectives/modernize-engineering-culture.html>
- <https://devops.com/how-platform-engineering-protects-profit-people-and-our-planet/>
- <https://learn.microsoft.com/en-us/training/modules/sustainable-software-engineering-overview/>
- <https://sustainability.georgetown.edu/50-facts-for-50-years>
- <https://fortune.com/2023/09/09/ai-chatgpt-usage-fuels-spike-in-microsoft-water-consumption/>
- <https://www.infoq.com/presentations/devsusops/>
- <https://illuminem.com/illuminemvoices/sustainable-it-the-why-what-and-how>
- <https://www.compareyourfootprint.com/measuring-key-kpis-sustainability-business/>
- <https://www.iea.org/>

For more information, contact askus@infosys.com

Infosys[®]
Navigate your next

© 2024 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/or any named intellectual property rights holders under this document.