RESOLVE

Case Study #001:

Investigating RPA Realities
and the Hype of Hyper-Automation

Introduction

In recent years, the concept of hyper-automation has garnered significant attention, often being hailed as the next frontier in business process optimization. At the heart of this hype is the assumption that Robotic Process Automation (RPA) is the foundational platform for achieving comprehensive automation and orchestration.

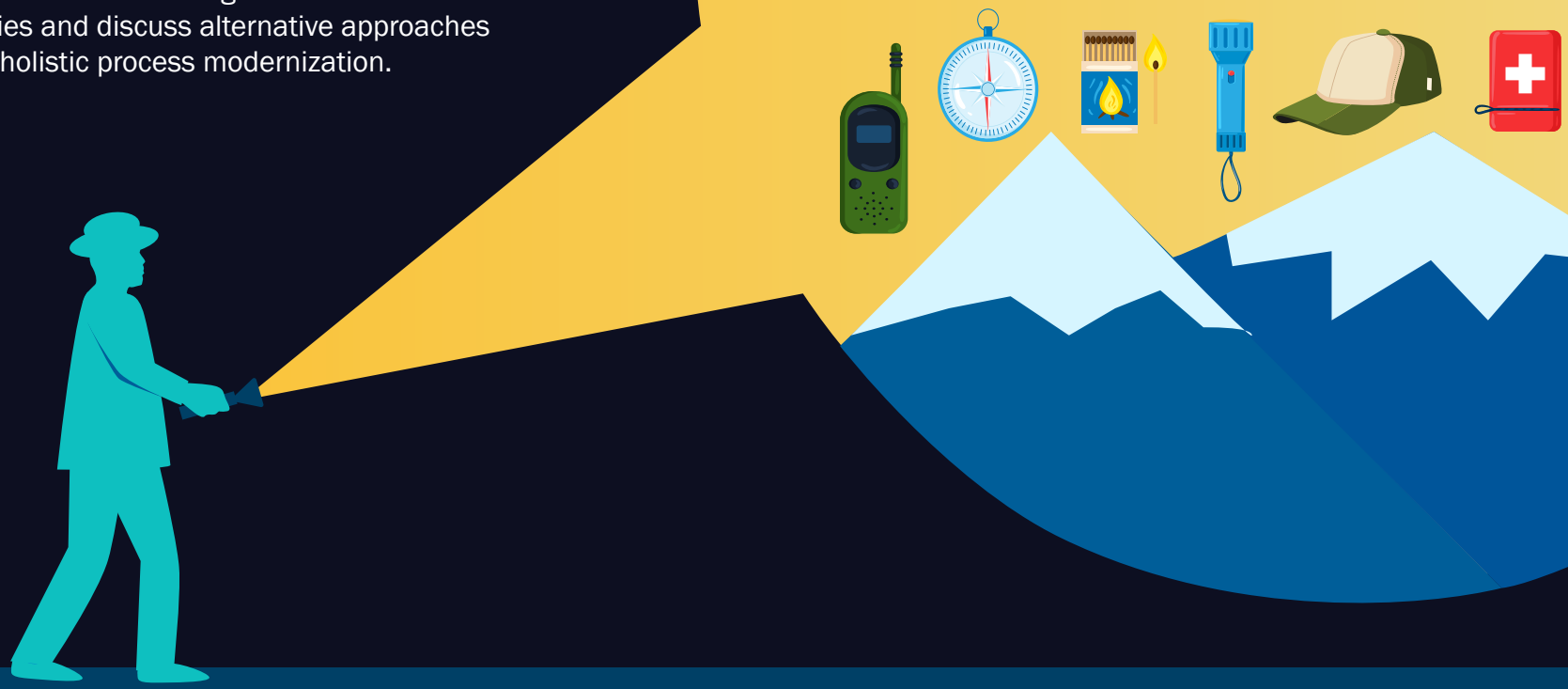
Despite its initial promise, RPA has struggled to handle the complexities inherent in modern business environments, rendering it insufficient for end-to-end process transformation.

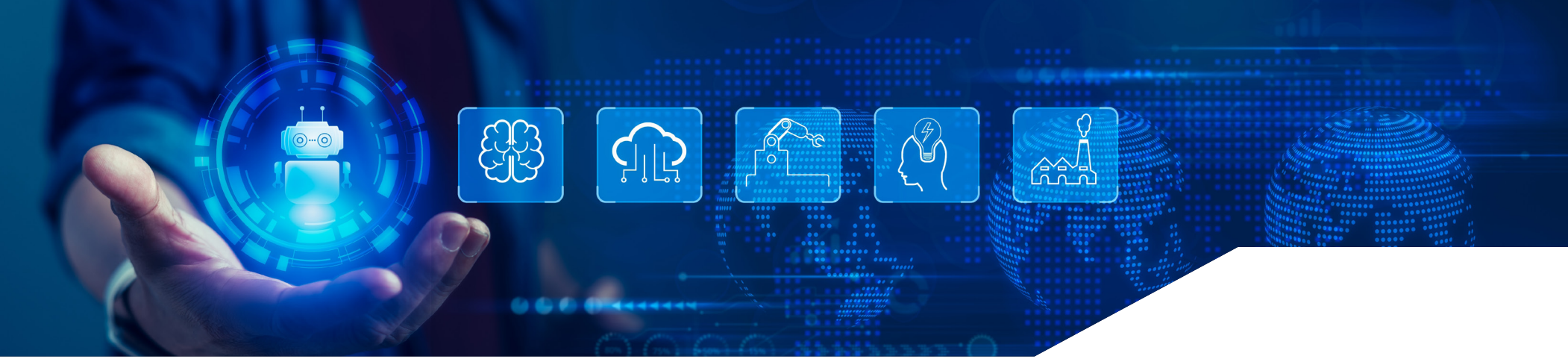
This eBook debunks the myth that hyper-automation efforts should be centered around RPA. Instead, we will explore why RPA's limitations prevent it from being the cornerstone of automation strategies and discuss alternative approaches that can truly drive holistic process modernization.

The Tools You Choose Matter

Imagine you're hiking in the mountains. The difficulty of the hike dictates the gear you bring—a GPS, extra batteries, proper footwear, and more.

You wouldn't set out on a multi-day trek armed with just a smartphone, would you?





Reality 1

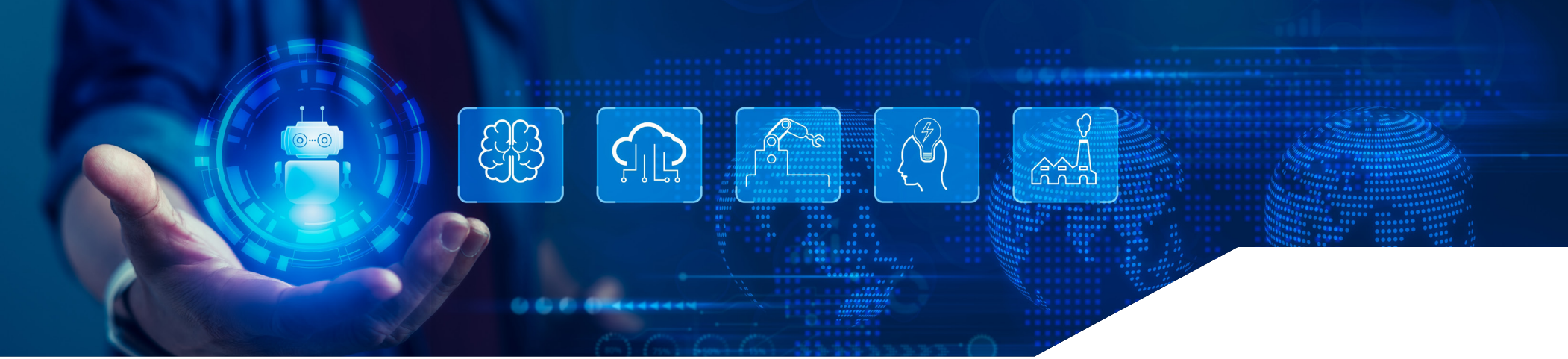
Only the Right Use Case = Success

One of the most critical factors in achieving success with Robotic Process Automation (RPA) is selecting the most appropriate use case. Conversely, choosing the wrong use case could set your RPA implementation up for failure, leading to wasted resources, unmet expectations, and a negative perception of automation within the organization.

Selecting the right use case for RPA isn't just about identifying tasks that are repetitive or time consuming, either. Automation leaders must thoroughly evaluate their use cases against various key parameters to ensure they select the most suitable candidates for RPA implementation. This evaluation process is essential because not all processes are equally suitable for automation, and overlooking critical factors can lead to significant challenges down the road.

8 Reasons RPA Might Not Be the Best Choice

1	Long workflows [greater than 10 steps and 7 if-else blocks]	✗
2	Complex business logic	✗
3	Orchestration / API connections [Greater than 3-5]	✗
4	Subjective decision making	✗
5	Unstable application [upgrades or user interface changes planned]	✗
6	Bring your own code? {to orchestrate existing automations]	✗
7	Comprehensive exception handling	—
8	Scale - go beyond 1M+ runs per month	✗



Reality 2

UI-based Automation = Tech Debt

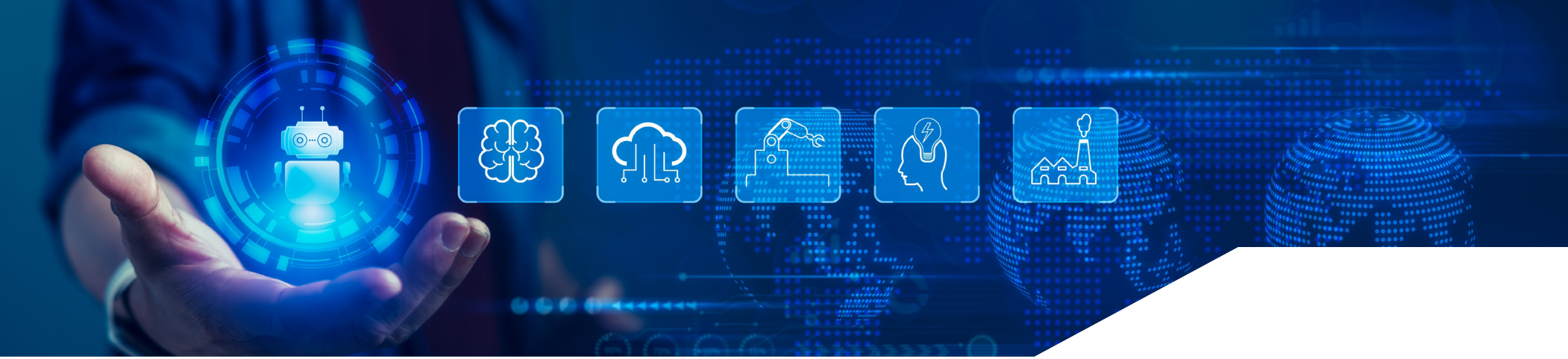
Tech debt refers to the future costs of rework when a quick, simple solution is chosen over a more robust, long-term approach. In the context of RPA, while it promises quick ROI, it can also quickly accumulate technical debt.

Automating at the User Interface (UI) level is often an easy and fast way to implement RPA. However, this approach is fragile—small changes in the UI can break the automation, leading to constant maintenance and rework. Over time, this creates technical debt that undermines the initial efficiency gains.

To avoid this, it's crucial to prioritize more sustainable automation strategies that minimize future rework and maintenance.

4 Ways UI-based Automation Creates More Problems Than It Solves

1	RPA requires constant maintenance and reconfiguration of the bots, leading to increased downtime and resource allocation for support.
2	Costs can quickly add up since pricing is based on number of bots. Scaling requires more bots, each with its own dedicated runtime environment.
3	Error handling is not robust. It's a paradox between a platform that can't handle complex decision logic and the need to interpret and respond to unexpected changes.
4	Interaction with the UI often inadvertently exposes and/or mishandles confidential information.



Reality 3
API or RPA?

Integration is the process of making independently designed systems work well together—a task that becomes increasingly complex as business processes evolve and expand.

Modern organizations must integrate diverse applications, data, events, devices, and people, both internally and externally. This expanding integration landscape presents unique challenges, particularly when RPA is used improperly.

Two major risks associated with RPA usage can quickly build up technical debt and drive up costs:

RISK 1: Using RPA for everything—even when robust APIs are available. While RPA is versatile, it's not always the best tool for integrating systems, especially when APIs are readily accessible. Over-relying on RPA in these cases can lead to fragile automations that require constant maintenance.

RISK 2: Using RPA as an API connection tool. RPA has evolved from its early days as a screen-scraping tool but using it to handle API-based workflows can be inefficient. Dedicated API integration platforms offer more stability and are better suited for these tasks, while RPA-based API automations often lead to increased complexity and technical debt.

While RPA platforms can be extended to support API-based automation, they are not inherently designed for this purpose. Developers might face challenges related to performance, scalability, integration complexity, and maintenance.

API Challenges in RPA: 5 Red Flags You Need to Consider

Attribute	Support	Comment
API Support and Features		Lacks advanced features such as asynchronous calls, handling large payloads, or complex authentication mechanisms.
Complex API Workflows		Struggles with API workflows that require conditional logic, looping through large datasets, or handling multi-step API interactions.
Concurrency		Not optimized for handling concurrent API calls, leading to issues with scalability.
Integration Implementation		Integrating data from APIs can require complex data transformation and mapping which is cumbersome to implement and a nightmare to maintain with RPA.
Tech-debt		Lack of native support for API versioning and testing will be a maintenance burden and people-intensive.
Security and Compliance		Secure API interactions, for e.g., managing tokens, encrypting data, is less straightforward in an RPA platform.

Reality 4

Total Cost of Ownership

When you consider automating with RPA, there are two types of costs you need to consider:

1. **Cost of bots**
2. **The cost of everything else needed to support the bots**

Cost of Bots

Licenses account for approximately 30–40% of the total implementation costs. This total cost will depend on the following:

Type of bots: unattended or attended

Number of bots

Frequency of usage

Other premium fuctionalities (AI Builders, IDP etc.)

Most RPA initiatives continue to be small and piecemealed; scaled RPA deployments are rare due to the licensing costs (not including the resources that are needed to support it).

Remember, each bot that is executing the automation needs to be paid for.

Cost of Everything Else

Each RPA bot requires supporting resources. For example, scaling RPA typically requires additional servers, storage, and network resources to host and manage the increased number of bots. This infrastructure investment can be substantial.

Further, let's say your RPA bot is interacting with an SAP instance—Logging into SAP and getting some key data to advance automation. You would also need to factor in the SAP license needed for the bot to be able to log in and perform its task.

RPA Realities Amidst the Hype of Hyper-automation

Hyper-automation refers to the use of advanced technologies, including artificial intelligence (AI) and machine learning (ML), in conjunction with RPA, to rapidly and efficiently identify, vet, and automate as many business processes as possible.

The goal is to transcend traditional automation by integrating multiple tools and technologies to automate complex workflows comprehensively.

While RPA offers significant benefits in automating repetitive, rule-based tasks, it's crucial to recognize the realities that temper the excitement surrounding hyper-automation.

Beware!

The allure of hyper-automation—seamlessly integrating advanced technologies like AI, ML, and RPA to automate end-to-end business processes—may be compelling. However, relying too heavily on RPA as the cornerstone for such complex initiatives can be risky.

RPA is inherently fragile due to its dependence on UI interactions, making it susceptible to breaking whenever underlying applications change. Adding AI and ML into the mix only increases the complexity, making maintenance even more challenging and costly.

The brittle nature of RPA bots means they require constant monitoring, frequent updates, and ongoing debugging, all of which can quickly erode the anticipated ROI. This fragility is a critical reason why centering hyper-automation efforts solely around RPA can lead to significant technical debt and operational inefficiencies.

Service Orchestration and Automation as the Way Forward

Just as relying solely on a smartphone for a multi-day hike would be unwise, depending entirely on RPA for comprehensive automation is short sighted. While RPA is effective for quick, rule-based tasks, it's not designed to handle the complexities of dynamic, end-to-end process automation. Service Orchestration and Automation, which is purpose-built to orchestrate complex workflows across different systems using API calls, CLIs etc., offers a more robust, scalable solution.

This approach allows organizations to automate more complex workflows while minimizing the risk of technical debt and ensuring long-term sustainability. Just as a detailed map and compass are better suited for a long, challenging journey, Service Orchestration and Automation provides the dependable, long-lasting tools needed to navigate the complexities of modern business environments successfully.

Ready to learn more?