

David Chappell

Understanding Enterprise RPA

The Blue Prism Example



Sponsored by Blue Prism

Copyright © 2016 Chappell & Associates

Contents

Introducing Enterprise RPA	3
What is Enterprise RPA?	3
Characteristics of Enterprise RPA Technologies	5
Implementing Enterprise RPA: The Blue Prism Example.....	6
Maintainability and Reuse	6
Robot Creation.....	7
Development Process	8
Scalability	9
Reliability	10
Access Control and Auditing	11
Management	12
Ecosystem	13
Conclusion	14
About the Author	14

Introducing Enterprise RPA

Robotic process automation (RPA) is a technology for automating business processes. Because RPA can be applied to a range of scenarios, including many that are beyond the reach of traditional integration projects, this approach has fired the imagination of many people. And because RPA can lower costs, increase flexibility, and improve process accuracy, as well as help business and IT work together more effectively, it's a hot technology that's being adopted by many organizations in many different industries.

The core idea of RPA is simple. Rather than have people interact with applications, RPA instead uses *software robots* that drive application user interfaces in the same way. (Don't be confused by the terminology—these aren't robots in the usual sense of the word. Still, the name RPA has become established in the industry, as has the notion of software robots.) Figure 1 shows how this looks.

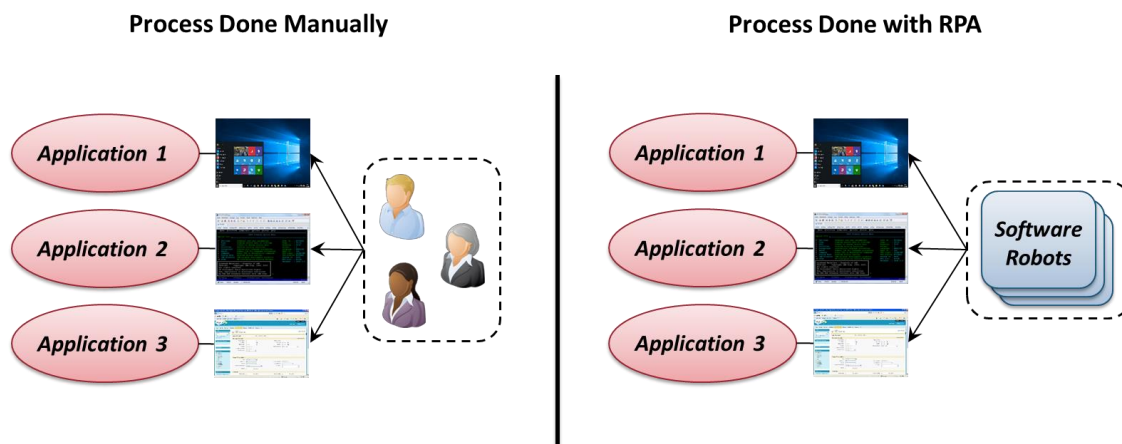


Figure 1: RPA uses software robots rather than people to carry out business processes.

Still, there's some confusion. The term "RPA" is applied to a broad set of offerings, ranging from simple screen-scraping products to much more complex technologies. The goal of this paper is to bring some clarity to this area. Specifically, we'll focus on *enterprise RPA*, describing what it is and what's required to do it well. As a concrete example of an enterprise RPA product, we'll use Blue Prism.

What is Enterprise RPA?

To understand the idea of enterprise RPA, start by thinking more broadly about the technology options for automating a business process. For business processes that rely on multiple applications, one approach is to rely on application programming interface (APIs) exposed by the applications the process uses. API-based integration is what's used in products that provide enterprise application integration (EAI), for example, and it's the right solution in many cases.

Yet automating a business process with RPA is often a better choice. As Figure 1 showed, RPA automates a business process solely through application user interfaces, which makes this approach simpler, faster, and less expensive than API-based automation. And because RPA can work with applications that don't expose APIs, it can be used with a broader range of existing software.

To decide whether RPA or the API-based approach is better, you need to consider two main factors:

- What kind of business process is being automated? How complex and important is it? Process automation that requires connecting two ERP systems when organizations merge is a multifaceted, business-critical operation. Automating a process that reads spreadsheets and feeds their data into an invoicing system is simpler to implement and probably less important to the business.
- Who will do the work required to implement the process automation? API-based automation—the EAI approach—is largely the province of IT, as it requires very technical people. RPA, however, is commonly done by business people, although sometimes in concert with IT.

Given these two dimensions for making the decision, Figure 2 summarizes the approaches your organization might take to automating business processes.

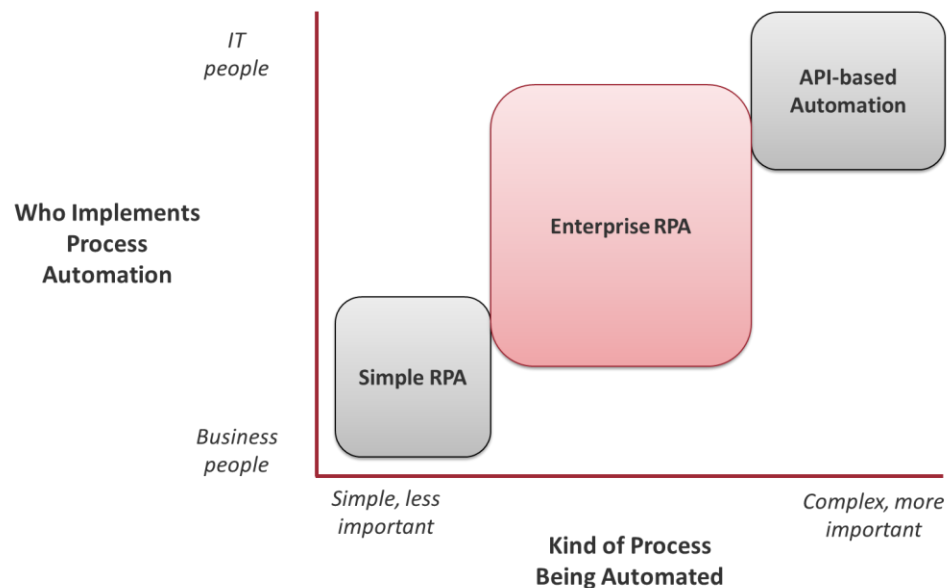


Figure 2: The best approach for automating a business process depends on what kind of process it is and who will implement the automation.

In this figure, the horizontal axis represents the complexity and importance of an automated process. The vertical axis shows the technical knowledge of the people who will implement that automated process, ranging from non-technical business users at the bottom to very technical IT staff at the top.

As the figure shows, automated business processes can be grouped into three categories, each addressed by a different style of process automation technology:

- In the upper right are complex, important processes automated using API-based automation. The implementation is done largely by IT.
- In the lower left are simple, less important processes for which automation can be done entirely by the business people who carry out these processes. Many RPA tools target this kind of process, often providing a

record button that makes it fast and easy for users to create automation scripts. The organization's IT staff commonly has no knowledge of these, since they're created and controlled entirely by the business. The technologies that address this part of the problem space provide simple RPA.

- The area in the middle is the province of enterprise RPA. The processes that fit here aren't as important or as complex as those in the upper right, but they're also not as simple as those in the lower left. To do them well, IT and business people need to work together. As the figure suggests, a substantial number of business processes fall into this category.

All three approaches—API-based automation, simple RPA, and enterprise RPA—have value, and so it's important to make good choices about which one to use in each situation. Business processes that are a good fit for enterprise RPA tend to have these characteristics:

- The automated process needs to be scalable, reliable, secure, and manageable. In other words, the automation must be built on a solid technical foundation; your IT organization needs to be comfortable with it.
- Automating the process has business value, but typically not enough to justify the cost of API-based automation. As a result, the process would never rise to the top of your IT organization's to-do list.
- Even though IT must be involved, control of the automated processes needs to remain with the business. This might be because the process changes frequently, and involving IT in every change would be too slow, or for some other reason. Yet once IT has laid the groundwork, business people must be able to create and change these process automations on their own.

Characteristics of Enterprise RPA Technologies

What do enterprise RPA technologies look like? To a great extent, these products have the core characteristics that organizations look for in all enterprise software. Those characteristics include the following:

- An approach that allows ***maintainability and reuse*** of software robots and the components they're built from.
- An approach to ***robot creation*** that balances the need for fast implementation with high-performance execution.
- A ***development process*** that's both repeatable and reliable.
- Support for ***scalability***, because enterprise RPA solutions don't run robots on user desktops.
- Support for ***reliability***, letting robots run unattended on servers.
- The ability to ensure ***access control and auditing***, letting an organization be sure that its robots are doing the right things.
- Effective ***management*** of robots, such as the ability to control them in groups rather than one at a time.
- An ***ecosystem*** that includes partners, training, and the other things enterprises need to succeed with RPA.

What follows uses Blue Prism to illustrate how these characteristics can be provided by an enterprise RPA solution.

Implementing Enterprise RPA: The Blue Prism Example

Blue Prism is an enterprise RPA company headquartered in the United Kingdom. Its product, also called Blue Prism, was an early entry in this space, and it's a market leader today. (In fact, Blue Prism coined the term “robotic process automation” in 2012.) Blue Prism is used today by financial services firms, organizations that do business processing outsourcing (BPO), mobile phone companies, and others.

Maintainability and Reuse

Simple RPA products commonly create scripts that directly access applications. This approach works well for desktop automation scenarios. For enterprise RPA, however, it raises some challenges.

For example, when an application's user interface changes, all of the scripts that use that application must change as well. This can make maintenance challenging—how does an organization even find every script that needs to be changed? Also, wrapping all of the logic of an automated business process into a script makes it hard to reuse that logic, even when multiple processes use the same applications in similar ways.

As an enterprise RPA product, Blue Prism takes a different approach. Rather than creating scripts, software robots are built using Blue Prism-defined *business objects* and *processes*. Figure 3 shows how this approach compares with simple RPA

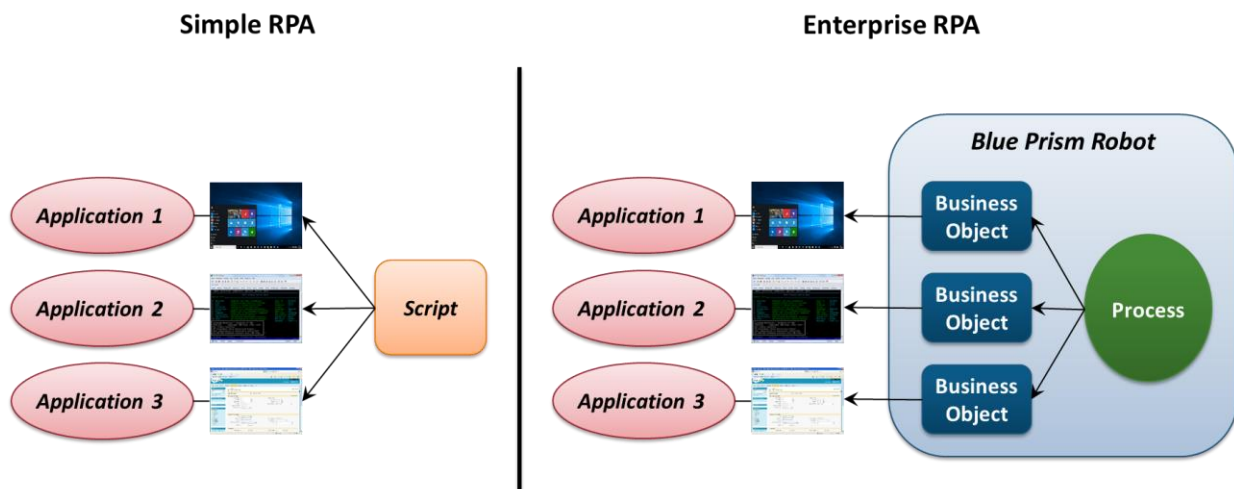


Figure 3: While simple RPA uses scripts, Blue Prism creates robots using business objects driven by processes.

In a Blue Prism robot, business objects communicate with application user interfaces, while the robot's process contains the logic that drives this communication. This approach can make life easier for the people who create and modify robots for a couple of reasons:

- Adapting to changes in application user interfaces is easier. Because all interaction with applications is done by business objects, changes in a user interface require changing only the business objects that talk to this application. This is simpler than finding and changing all of the scripts that work with the changed application.

- New business processes can be automated more quickly, because the same business objects can be reused by multiple processes. Over time, organizations using Blue Prism build up a library of business objects for communicating with their applications. Once this is done, automating a new business processes requires only creating new logic to drive these objects.

Robot Creation

Every RPA technology provides tools for creating automated processes. But the tools provided in simple RPA differ significantly from those used with an enterprise RPA solution such as Blue Prism. Figure 4 illustrates this difference.

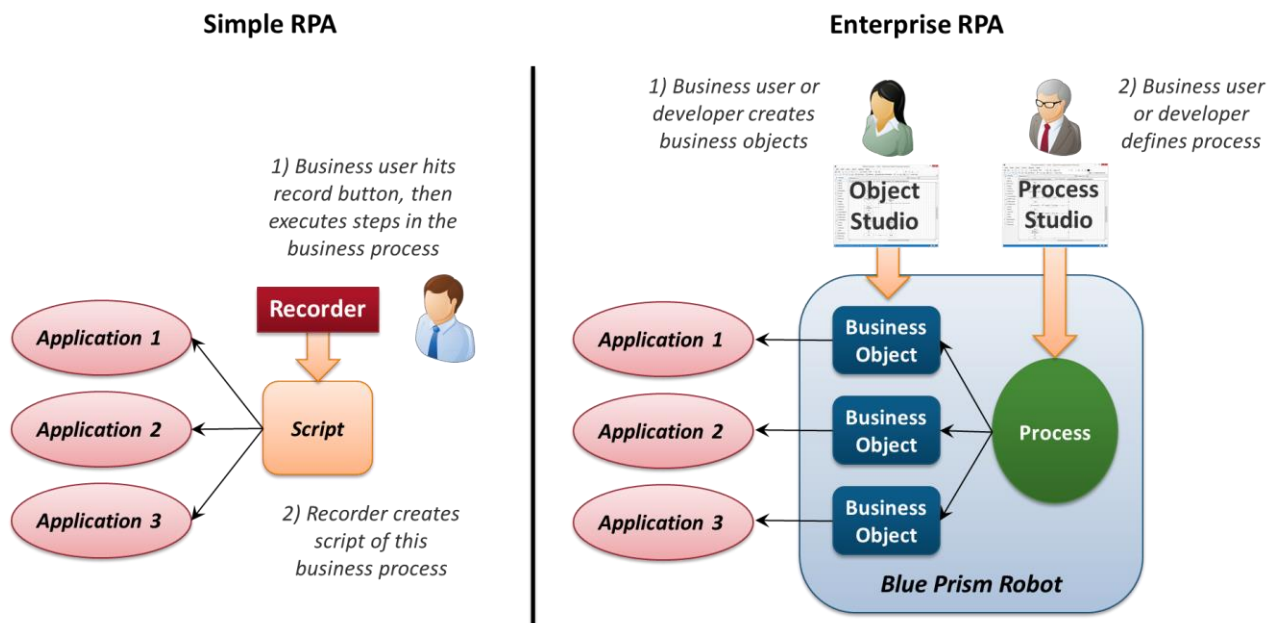


Figure 4: Rather than recording a user's action as a script, Blue Prism provides tools for creating objects and processes.

Simple RPA solutions are aimed at creating scripts that implement a business process. Accordingly, these technologies commonly provide a tool that records what a business user does, then generates an executable script to carry out the same steps.

As an enterprise RPA technology, however, Blue Prism's approach is quite different. The goal is to create (or perhaps reuse) business objects and processes, and so the product provides tools to do these things: *Object Studio* for creating business objects and *Process Studio* for creating processes. Both are graphical tools, designed to be used by either business users or software developers. No coding is required.

Pressing a record button to generate a script, the simple RPA approach, has some real advantages. It's easy, for one thing, and it's also fast. If your goal is to create an automation as quickly as possible, this approach can be the best option. In an enterprise RPA scenario, however, the goal is typically to optimize the *execution* of software robots, not the time it takes to create them. Rather than creating simple scripts, an organization is typically building a virtual workforce that can be effectively deployed, managed, and maintained.

And while scripts are fast to create, they come with some limitations. Creating a script with a recorder can work well with linear business processes that don't branch frequently or make complex decisions. For more complex processes, however, using a graphical tool that supports a more flexible process flow is likely to be more effective. Enterprise RPA scenarios commonly deal with this kind of business process, which is why Blue Prism provides Process Studio.

Scripts can also be constrained in their behavior. For example, if an application took two seconds to respond when the script was created, what happens if that same application takes ten seconds to respond when the script is running? While enterprise RPA tools are typically designed to handle this kind of variation, situations like this can cause problems with scripts.

Finally, generated scripts can be harder to scale, secure, and manage than software robots, as described later, and they might also execute more slowly. If an automated business process handles a million transactions a month, and a software robot is one second faster than a generated script for each transaction, an organization will save a million seconds a month—close to 300 hours. Because the usual goal in enterprise RPA is to maximize process execution, spending more time creating the robots is commonly seen as the right trade-off to make.

Development Process

Simple RPA typically doesn't require a formal development approach. Using a tool that generates a script from a user's actions is straightforward, and the process being automated isn't critical to the business. Enterprise RPA does require a more structured approach, however. Since enterprise RPA is commonly used for more important business processes, some degree of rigor in development is required. Without this, an organization runs the risk of introducing errors and inconsistency into the heart of its operations.

Accordingly, Blue Prism defines a structured process for creating software robots. The team that executes this process is led by business people, although it typically includes both business-oriented and more technical members. Whatever the team's makeup, it begins by creating a *solution design document*. This document defines the entire automated solution, which might include multiple Blue Prism processes and business objects, and it's quite detailed. Because it will be used to create a software robot that can run on its own, a solution design document must specify the complete business process, what exceptions might happen, what to do for each exception, and more.

From this document, the team then creates a *process design instruction* for each Blue Prism process that must be created and an *object design instruction* for each business object the solution requires. These documents become task lists for the people who use Process Studio and Object Studio to create the Blue Prism processes and business objects for this automated process.

Next, the newly created robots must be tested. Some organizations have access to test versions of the applications these robots use. In this case, testing can be done using these non-production alternatives. In many cases, though, the applications the new robots will access exist only in production—no test version is available. The idea of testing on production applications can seem scary, and if it's not done intelligently, it is. Yet the truth is that creating a new software robot to work against applications is much the same as training a new human user to use those applications, something that organizations have plenty of practice with. Just like a new user, a Blue Prism robot being tested has a human expert sitting alongside it, making sure that everything it does is correct. This kind of process verification is the same approach used when training a new human user, and the risk is no greater.

It's important to understand the role that business people play in this methodology: They're deeply involved in making sure that the automated process is correct. In fact, business analysts might even be the prime movers on the team developing the process. To some IT people, this kind of business-led computing is terrifying. Even with the Blue Prism methodology just described, won't the product's business-friendly tools tempt non-expert people to take too much into their own hands? And if something goes wrong, won't they blame IT rather than themselves?

These are legitimate concerns, especially in organizations where there's not much trust between the business and IT organizations. Still, when the overarching need is for rapid process development and frequent process change—and it often is—making these efforts contingent on IT isn't feasible. Helping the business side of the organization do as much as it can, then gently working with them to fix any mistakes, is likely to be a better course. After all, they know their own processes intimately, and they certainly want to automate them correctly. Using the structured development process provided by Blue Prism, IT and business people can work together toward a common goal

While having a repeatable development process is necessary for enterprise RPA, it's not sufficient—more is required. For example, many organizations have chosen to create an RPA center of excellence (CoE) that can capture best practices, then spread them throughout an organization. More broadly, Blue Prism has created a *robotic operating model (ROM)* that describes how to adapt an organization's culture to the concepts enterprise RPA brings. The first step in the ROM is to define the vision for RPA, matching the expected benefits with an organization's strategy. After this, the model recommends creating an RPA governance board, including representatives from both IT and business, to make decisions about RPA projects. Blue Prism's ROM also includes a service model that specifies how RPA support should be handled, a set of roles for implementing processes (such as a process analyst and a process developer), and other aspects of using RPA. The goal is to make sure that an organization has a broad enough perspective to be successful with enterprise RPA.

Scalability

The scripts used in simple RPA solutions commonly run on their user's desktop or laptop computer. Execution of those scripts can be started and stopped by a user, who's often sitting at the machine while they execute. For the straightforward automation scenarios these products address, this is an effective approach.

But if your goal is to create a virtual workforce of software robots, as is typical in enterprise RPA, this approach won't work. In scenarios like this, it's common to have many robot instances executing the same business process, just as a group of people would. If each robot had to be individually started, stopped, and monitored, you wouldn't gain much from the automation. A more scalable approach is required that lets robots run largely on their own.

To accomplish this, those robots can't run on desktops or laptops. Instead, they must run in virtual machines hosted on servers in a datacenter. Figure 4 illustrates the situation, once again showing both simple RPA and Blue Prism.

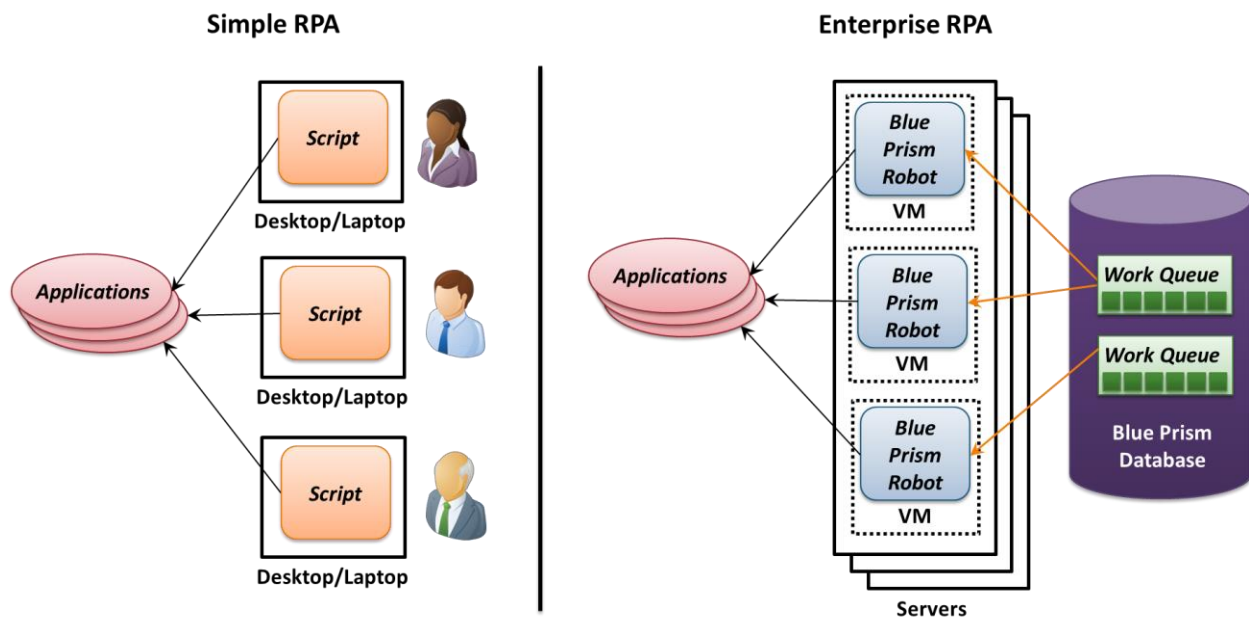


Figure 5: While simple RPA typically runs scripts on desktops, Blue Prism runs robots on servers in a datacenter.

As the figure suggests, Blue Prism is designed to minimize the number of people needed to monitor the running robots. The goal is lights-out automation, with one or two people able to shepherd hundreds of robots.

Since the robots aren't being started and stopped by people, as scripts often are, they need a more scalable way to be assigned work. As Figure 5 shows, Blue Prism robots can get their input from *work queues* stored in the *Blue Prism database*. For example, suppose a group of Blue Prism robots was used to read data from many spreadsheets, then insert that data line by line into an invoice management system. The spreadsheets could be stored in a work queue, with each robot reading from the queue. This makes it straightforward both to assign work to the robots and to scale the number of robots up and down as the workload changes.

Organizations often look to enterprise RPA to bring transformational change. Rather than helping current workers automate some of the work they do on their desktops, enterprise RPA aims at eliminating full-time positions. This lets an organization re-assign these people to more valuable task, or perhaps grow their business without adding staff. Scalable software robots are an essential part of how enterprise RPA enables this.

Reliability

RPA inherently makes processes more accurate. Unlike people, software robots don't get tired and make mistakes. Using robots can also improve data quality, since it eliminates typos made by people. Yet all but the simplest business processes will sometimes have variations, such as a longer-than-usual response time, and raise exceptions.

With simple RPA, these can be handled by the people monitoring a running script. If a script crashes in the middle of execution, it can be manually restarted. With enterprise RPA, however, neither of these is possible—the robots are running largely unattended. This implies that enterprise RPA technologies must have stronger built-in support

for more complex processes that include exception handling, along with a way to recover from failures automatically.

To address these requirements, the Blue Prism process model is designed to handle complex logic with exception handling. For example, a robot might create a support ticket or send email to an administrator when a problem arises. The product is also capable of restarting a failed robot and marking any queue items that it was working on when a failure happened. This lets those items be examined by a Blue Prism administrator to determine how they should be handled.

Blue Prism also supports communication between robots about unusual events. For example, if a robot repeatedly fails to communicate with a particular application, it can inform other robots of this problem. Those robots can then stop, marking the point at which they should restart, rather than continuing to overload the non-responding application. Backing off like this can help this application recover by itself, which improves the reliability of the automated process.

Access Control and Auditing

RPA can lower an organization's costs and improve productivity. But neither of these things can happen if the organization doesn't trust its software robots to carry out business processes in a correct and compliant way. To ensure this, an enterprise RPA solution must provide effective access control and auditing.

Access control lets an enterprise RPA solution limit who has access to robots and the data they use. Unlike simple RPA products, where each business user might have the ability to unilaterally change a process or access the data it uses, enterprise RPA uses access control to constrain who can do these things. With Blue Prism, an administrator can add a new Blue Prism user, then define the roles that user can operate in. For example, one user might be allowed to create new business objects and processes, while another can only run existing robots. Because you now need to trust fewer people, your security should improve.

A related challenge is providing a way to audit the operation of robots. For enterprise RPA, auditability means that you can track what changes were made to a robot, along with who made those changes and when. Blue Prism also logs changes to each user's permissions and other security-sensitive data. Having this information is important for finding errors, and it's also essential for detecting fraud or other attempts to subvert a robot's work. To provide this, Blue Prism maintains an audit log for each robot.

Simple RPA technologies commonly take a more lightweight approach to these concerns. Figure 6 illustrates some differences between simple RPA and Blue Prism in this area.

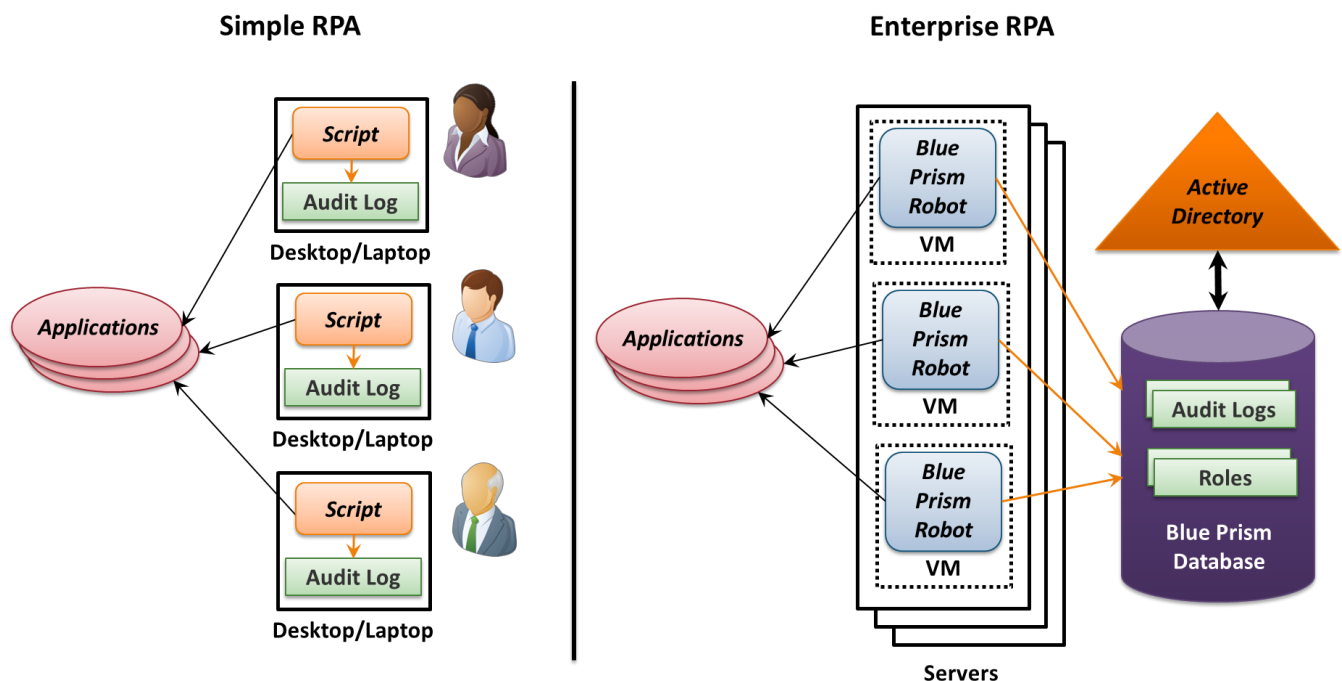


Figure 6: Unlike simple RPA, Blue Prism stores audit logs, role definitions, and other information in a central database.

As the figure shows, simple RPA solutions can maintain an audit log on each machine. This is quite different from Blue Prism, where audit logs and role definitions are kept in the centralized Blue Prism database. Using Blue Prism's audit log viewer, an administrator can examine the audit logs to find and fix problems or respond to audit requests.

As illustrated, Blue Prism commonly depends on Active Directory for user identities (although it provides its own encrypted credentials database that can be used if desired). Blue Prism can also use Active Directory to assign logins to the robots themselves. This is important, since robots need a secure and uniquely identifiable way to access the applications they use.

Management

In enterprise RPA, software robots run in VMs on servers in an organization's datacenter. Since those servers often don't have keyboards or monitors, enterprise RPA products must provide tools for managing robots remotely. Blue Prism provides two tools for doing this, each implemented as a tab in the Blue Prism client application. The *Control* tab lets an administrator work with robots, while the *System* tab lets an administrator add users and make other changes to the Blue Prism environment. Figure 7 illustrates this, along with how it compares to a typical simple RPA product.

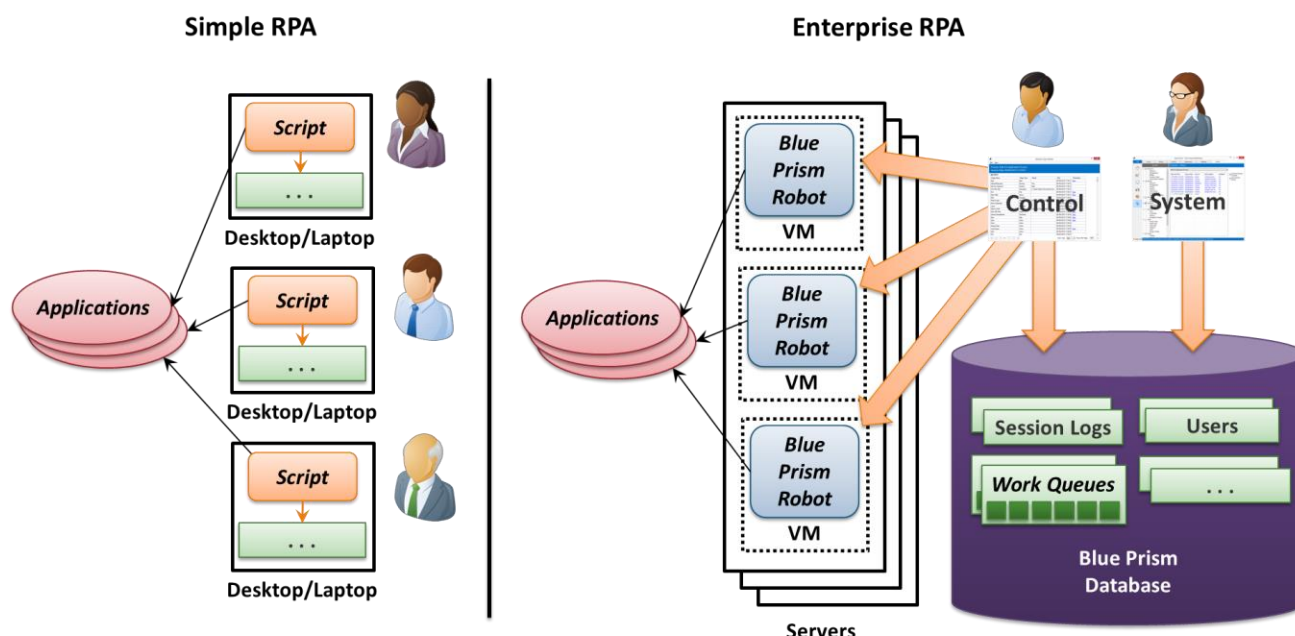


Figure 7: Blue Prism provides a central point of control for managing robots and users.

As this figure suggests, simple RPA products typically don't provide much in the way of centralized management. Instead, they depend on their users to do what's required to keep scripts running. As an enterprise RPA product, Blue Prism's approach is quite different. For example, using the Control tab, an administrator can do things such as the following:

- ❑ Start and stop robots.
- ❑ Examine the session log maintained by each robot. This lets the administrator see what robots are running, where each one is in the process it's executing, and see all of the data the robot has used in this process.
- ❑ Switch robots in groups from one process to another. For example, suppose an organization has 50 robots reading from a work queue that contains new accounts to be opened. An administrator can cause, say, 30 of those robots to begin executing some other process reading from another work queue. Rather than do this one robot at a time, the tool lets this reassignment be done as a group.

The goal is to help organizations effectively manage their virtual workforce in a centralized way, while still providing a clear separation between the people who create and modify robots and the people who manage them.

Ecosystem

Every RPA product provides support for automating business processes through application user interfaces. For enterprise RPA, however, this isn't enough. Like every other enterprise offering, products in this category must also be part of a larger ecosystem.

The most important aspect of this ecosystem is partners, firms that provide complementary services to an enterprise RPA product. For example, suppose your organization chooses to adopt an enterprise RPA product. To

shorten your time to business value, you might choose to hire a partner of your RPA vendor who already has deep implementation skills. Ideally, this partner is someone you already trust, such as a global integration firm that you're working with today. The partner might even specialize in your industry, making them a better fit for your needs. Your RPA vendor will always provide the deepest product knowledge, but partners are an essential part of getting business value from an enterprise RPA product.

Blue Prism's implementation partners today include a range of companies: Accenture, Capgemini, Cognizant, HCL, IBM, and others. These organizations help Blue Prism's customers deploy the product, create robots, manage those robots, set up CoEs, and more. Some of them also provide broader skills, such as help in deciding whether to automate an existing process unchanged or to first spend time improving the process.

To help ensure that customers can trust the partners they choose, Blue Prism provides a partner certification program. Certified partners are required to complete a defined set of Blue Prism training courses. The goal is to make sure that customers can choose competent partners who can genuinely help them be successful with their enterprise RPA implementations.

Conclusion

Simple RPA technologies focus almost entirely on business users. API-based integration relies mostly on IT. As an enterprise RPA technology, Blue Prism is intended to be used by business and IT working together.

With Blue Prism, IT creates the environment and configures deployments, making sure that robots can run in a scalable, reliable, and secure way. Once this is done, business people can create and modify automated processes largely on their own. This approach lets an organization be sure that its software robots meet compliance and regulatory requirements while still letting those robots be created and updated quickly. The goal is to meet the needs of both IT and the business.

Simple RPA is the right solution for some problems, while others demand API-based integration. Yet the broad middle ground of process automation is occupied by enterprise RPA. Without this capability, your organization can miss opportunities to make your operations better: to improve the productivity of your people, to make life easier for your customers, and to compete more effectively in your market. If you're not at least considering enterprise RPA today, you could well be missing out on an important way to advance your organization.

About the Author

David Chappell is Principal of Chappell & Associates (<http://www.davidchappell.com>) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technologies.