

Diese Leseprobe haben Sie beim  
 edv-buchversand.de heruntergeladen.  
Das Buch können Sie online in unserem  
Shop bestellen.

[Hier zum Shop](#)

## Kapitel 3

# Einführung in SAP Intelligent Robotic Process Automation

*In diesem Kapitel lernen Sie das RPA-Tool SAP Intelligent Robotic Process Automation und dessen Hauptkomponenten kennen.*

Nachdem wir in den bisherigen Kapiteln detailliert auf die theoretischen und methodischen Grundlagen von RPA sowie die Abgrenzung zu verwandten Disziplinen eingegangen sind, dienen die folgenden Kapitel der Vorstellung der Möglichkeiten zur technischen Umsetzung am Beispiel der RPA-Lösung *SAP Intelligent Robotic Process Automation* (SAP Intelligent RPA). Wir beginnen in diesem Kapitel mit einer Einführung in die verschiedenen Komponenten dieser Lösung.

Zunächst liefert Ihnen Abschnitt 3.1 einen Überblick über die Hauptkomponenten des Tools. Daraufhin stellen wir in Abschnitt 3.2 das Desktop Studio, in Abschnitt 3.3 das Cloud Studio und in Abschnitt 3.4 den Desktop Agent vor. In Abschnitt 3.5 erfahren Sie mehr über die Cloud Factory, und Abschnitt 3.6 erläutert das Zusammenwirken der verschiedenen Komponenten.

Aufbau des Kapitels

### 3.1 Überblick

SAP Intelligent RPA umfasst vier Hauptkomponenten, die der Erstellung, Bereitstellung, Ausführung und Überwachung intelligenter Software-Bots dienen. Dieser Prozess wird in verschiedene Entwicklungsphasen aufgeteilt (siehe Abbildung 3.1):

Phasen der Bot-Entwicklung

1. Die Entwicklungsphase *Create* umfasst das Design und die Entwicklung der Bots.
2. Die Entwicklungsphase *Run* ist für den Betrieb der Bots zuständig.
3. Die Entwicklungsphase *Control* überwacht den Einsatz und bildet die Orchestrierung sowie das Monitoring ab.

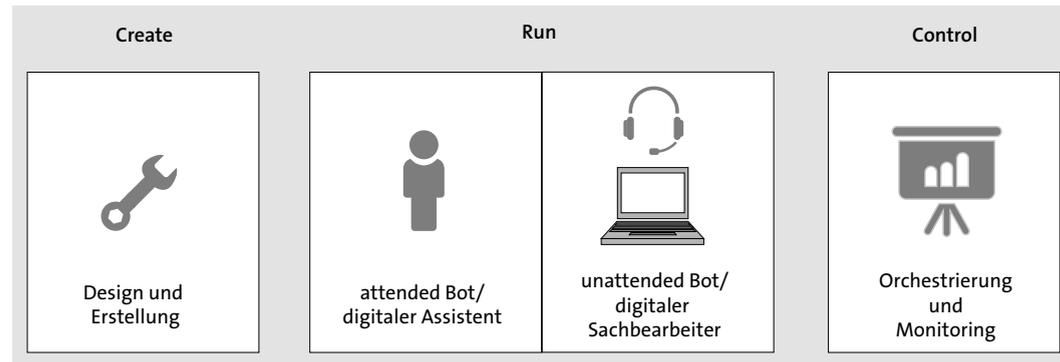


Abbildung 3.1 Die Entwicklungsphasen Create, Run und Control

Softwarekomponenten Abbildung 3.2 zeigt die Softwarekomponenten von SAP Intelligent RPA, die wir Ihnen im Folgenden näher vorstellen.

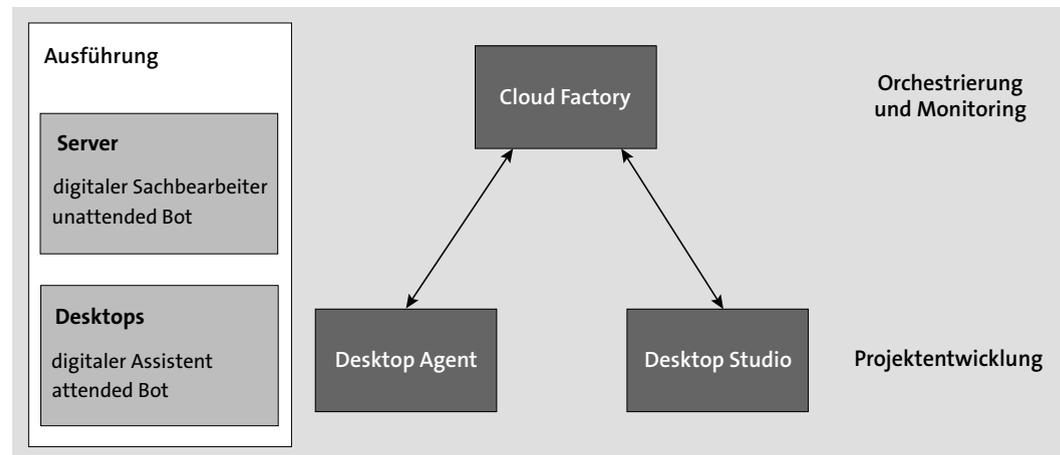


Abbildung 3.2 Softwarekomponenten von SAP Intelligent RPA

Die vier wichtigen Komponenten sind:

#### ■ Desktop Agent

Der Desktop Agent ist eine Softwarekomponente, die auf dem Rechner der Anwenderin oder des Anwenders installiert wird und die entsprechende Benutzer-Session nutzt. Der Desktop Agent erfordert ein *Automatisierungsszenario*, d. h. eine Reihe von Skripten, die beschreiben, welche Aktionen vom Bot erwartet werden. Diese Skripte werden mit dem Desktop Studio oder Cloud Studio erstellt und über die Cloud Factory bereitgestellt. Der Desktop Agent interagiert mit bereits vorhandenen Anwendungen, um die entsprechenden Aufgaben auszuführen. Diese Anwendungen werden durch die Replikation von Benutzeraktionen oder

durch die Ausführung regelmäßiger Ereignisse gesteuert. Um ordnungsgemäß zu funktionieren, erfordert der Desktop Agent eine Verbindung zur Cloud Factory.

#### ■ Cloud Factory

*SAP Intelligent RPA Cloud Factory* (im Folgenden kurz Cloud Factory genannt) ist eine Softwarekomponente, die auf der SAP Business Technology Platform (SAP BTP) läuft und gehostet wird. Sie verwaltet alle gemeinsamen Ressourcen im Zusammenhang mit SAP Intelligent RPA an zentraler Stelle. Das heißt, sie ist zuständig für die Verwaltung von Umgebungen und Bot-Hierarchien, das Deployment von Projekten durch Pakete sowie die Konfigurationen und Steuerung von Jobs, Agenten und Triggern. Die Cloud Factory stellt die Projekte, Szenarien oder auch Jobs (sowohl attended als auch unattended) auf dem Desktop Agent bereit.

#### ■ Desktop Studio

Das Desktop Studio ist eine Softwarekomponente, die auf dem Rechner einer RPA-Entwicklerin oder eines -Entwicklers installiert wird. Dieses Werkzeug dient der Erstellung von Bots unter Berücksichtigung aller Aspekte der Prozessautomatisierung, einschließlich der Anwendungserfassung und -deklaration, der Definition von Szenarien und Workflows sowie dem Testen und Debugging. Das Desktop Studio benötigt den Desktop Agent und eine Verbindung zur Cloud Factory, um ordnungsgemäß zu funktionieren.

#### ■ Cloud Studio

Das *SAP Intelligent RPA Cloud Studio* (im Folgenden kurz Cloud Studio genannt) ist eine Cloud-Softwarekomponente, die nicht mehr auf dem Rechner der RPA-Entwicklerin oder des -Entwicklers installiert werden muss, sondern über den Webbrowser aufgerufen wird. Das Cloud Studio ersetzt strategisch das Desktop Studio im Sinne der Cloud-First-Strategie von SAP. Genau wie sein Pendant, das Desktop Studio, dient das Cloud Studio der Erstellung von Bots unter Berücksichtigung aller Aspekte der Prozessautomatisierung einschließlich Anwendungserfassung und -erkennung, der Definition von Szenarien und Workflows sowie dem Testen und Debugging. Es benötigt eine Verbindung zur Cloud Factory, um die auf Basis abgeschlossener Design-Projekte erstellten Pakete zum Deployment zur Verfügung stellen zu können.

SAP Intelligent RPA ist eine integrierte, Cloud-basierte Lösung. Die hiermit implementierten Bots können sowohl in SAP- als auch in Nicht-SAP-Systemen eingesetzt werden, unabhängig davon, ob diese Systeme on-premise oder in der Cloud betrieben werden. Es handelt sich bei SAP Intelligent RPA um eine End-to-End-Lösung zur Automatisierung aller Arbeitsschritte.

Automatisierbare  
Systeme

Makros und VBA-Skripte sind beliebte Mittel, um bestimmte Aufgaben in Microsoft-Anwendungen wie Excel zu automatisieren. Sie können jedoch nur innerhalb der einzelnen Anwendung funktionieren und nicht für andere, Nicht-Microsoft-Anwendungen eingesetzt werden. SAP Intelligent RPA überwindet diese Systembrüche. Die Lösung kann sowohl mit Nicht-SAP-Systemen und Legacy-Anwendungen als auch mit SAP-Lösungen wie *SAP ERP*, *SAPS/4HANA*, *SAP Ariba*, *SAP Concur* und vielen anderen interagieren. Abbildung 3.3 zeigt einige Beispiele für Anwendungen, die mit SAP Intelligent RPA erfasst und automatisiert werden können.

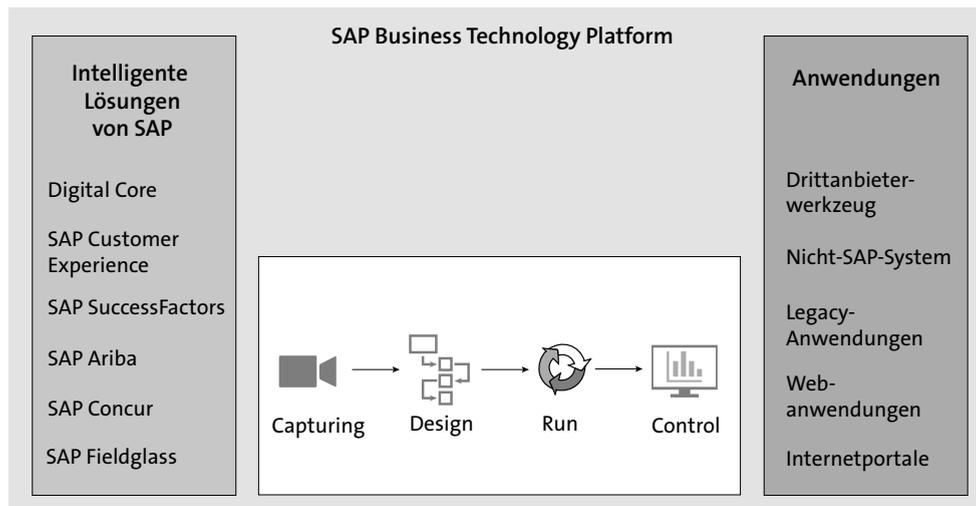


Abbildung 3.3 Anwendungen, die mit SAP Intelligent RPA automatisiert werden können

In den folgenden Abschnitten gehen wir auf die einzelnen Hauptkomponenten von SAP Intelligent RPA ein.

## 3.2 Desktop Studio

Entwicklungs-  
umgebung

Das Desktop Studio ist eine integrierte Entwicklungsumgebung (IDE) zur Entwicklung und zum Testen von RPA-Bots. Es nutzt ein *Software Development Kit* (SDK), das ein Projekt-Framework und gemeinsame Ressourcen bereitstellt und Plug-ins von Drittanbietern für Aktionen wie das *Application Capturing*, also die Erfassung der Screens einer zu automatisierenden Anwendung, unterstützt. Das Desktop Studio fasst alle relevanten Informationen eines Bots zusammen und liefert als Ergebnis ein sogenanntes *Paket* (engl. *Package*). Dieses Paket muss in die Cloud Factory importiert werden, um den Bot bereitzustellen und verwalten zu können (siehe Abbildung 3.4).

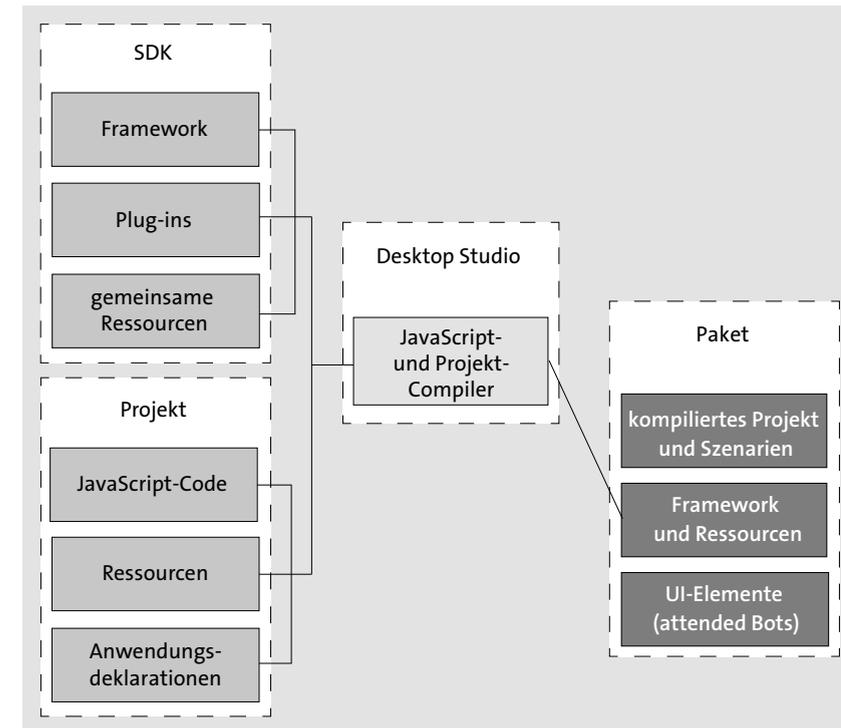


Abbildung 3.4 Komponenten des Desktop Studio und eines Pakets

Sobald ein Paket in die Cloud Factory importiert wird, wird ein Projekt erstellt. Typischerweise besteht dieses aus Parametern, die z. B. beschreiben, ob das Paket im Attended- oder im Unattended-Modus laufen soll, in welchen Zeitfenstern das Paket zur Ausführung zur Verfügung steht oder wann ein Bot ausgeführt werden soll. Ein Paket ist dabei stets an eine Umgebung gebunden, z. B. die Entwicklungs- oder Testumgebung. Auf das Konzept der Umgebungen gehen wir in Abschnitt 3.5, »Cloud Factory«, noch spezifischer ein.

Als integrierte Entwicklungsumgebung bietet Ihnen das Desktop Studio im Wesentlichen die folgenden Funktionen:

- Application Capture Tool zur Anwendungserfassung
- Workflow Designer
- Code Editor und Debugger

Wir stellen Ihnen diese einzelnen Komponenten im Folgenden näher vor.

Mit der Anwendungserfassung werden Anwendungen und Benutzeroberflächenelemente identifiziert und deklariert, die bei der Automatisierung des Geschäftsprozesses durch den Bot verwendet werden sollen. Das Desk-

Konfiguration des  
Projekts

Komponenten des  
Desktop Studio

Anwendungs-  
erfassung

top Studio bietet als Application Capture Tool *Inspektoren* für die Elemente der zu erfassenden grafischen Benutzeroberflächenelemente und einen *Validierungsassistenten* (Validation Wizard). Bei diesen Funktionen handelt es sich um spezifische Tools für die Erfassungs- und Deklarationsphase der RPA-Entwicklung. In Kapitel 4, »Capturing und Deklaration von Anwendungen«, zeigen wir Ihnen, wie Sie in der Erfassungs- und Deklarationsphase vorgehen.

**Workflow Designer** Der *Workflow Designer* wird zum Entwerfen eines Workflows verwendet. Sobald die zu automatisierenden Anwendungen und ihre Benutzeroberflächenelemente im Anwendungserfassungswerkzeug erfasst wurden, stehen sie im Workflow Designer zur Verfügung, wo Sie sie zu einem Prozess orchestrieren können. Die Modellierung erfolgt dabei grafisch: Die einzelnen Schritte des Workflows können per Drag-and-drop aus einer Palette ausgewählt und dem Workflow hinzugefügt werden (siehe Abbildung 3.5).

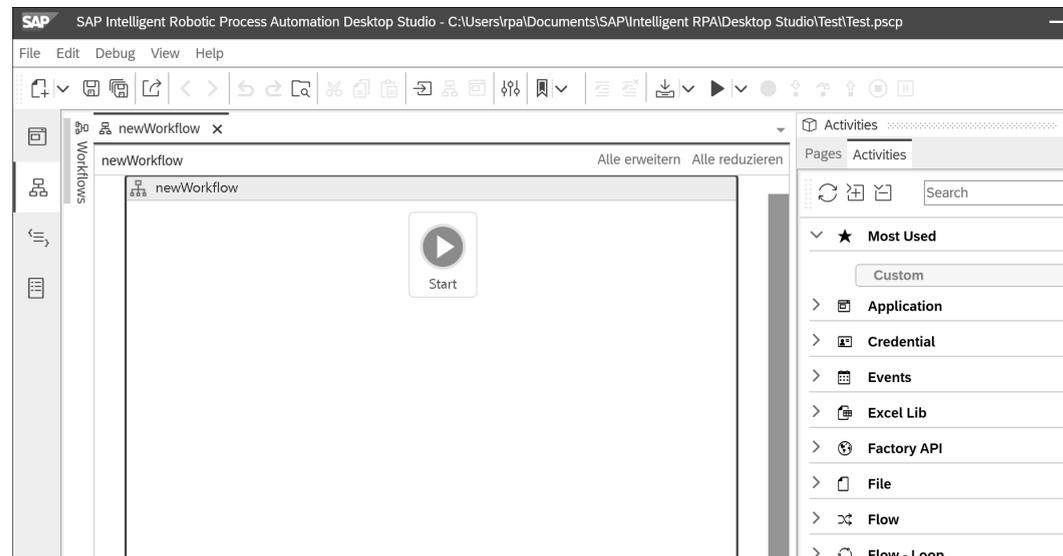


Abbildung 3.5 Workflow Designer im Desktop Studio

Jeder Flowgraph, also jede grafische Darstellung eines Workflows, beginnt mit einem Startsymbol. Durch Hinzufügen der verschiedenen erfassten Elemente wird ein Prozessablauf erstellt. Die Aktionen, die im Rahmen der Automatisierung auf den Anwendungsbildschirmen durchgeführt werden sollen, sind als sogenannte *Aktivitäten* im Workflow Designer verfügbar. Sie können aus dem Fenster **Activities** in den Flowgraph des Workflows gezogen werden. Wie das geht, erfahren Sie neben weiteren Aspekten der Workflow-Erstellung in Kapitel 5, »Workflows designen«.

Nach Abschluss des Design-Prozesses kann mit dem Speichern der Java-Script-Code des Workflows generiert werden. Diesen Code können Sie anschließend im Code Editor bearbeiten.

Der *Code Editor* ermöglicht es Ihnen, eigenen Code hinzuzufügen oder den generierten Code zu optimieren, sobald der Code im Workflow Designer generiert wurde. Sie haben hier Zugriff auf das von SAP bereitgestellte SDK und können darüber hinaus eigene JavaScript-Bibliotheken erstellen und einbinden. Es ist jederzeit möglich, zwischen der grafischen Darstellung des Workflows und dem Code zu wechseln. Den Umgang mit dem Code Editor zeigen wir Ihnen in Kapitel 6, »Workflows erweitern«.

Ein vollständiger *Debugger* ist ebenfalls Teil des Desktop Studio. Er bietet Funktionen wie eine schrittweise Ausführung, eine Variablenüberwachung oder Breakpoints. Daneben sind Tools zum Testen Ihrer Automatisierungen im Desktop Studio enthalten. Sie können damit jederzeit Ad-hoc-Tests mit einem vollständigen JavaScript-Debugger und einer aktiven Konsole durchführen. *Remote-Tests* werden durch Aufzeichnungs- und Wiedergabefunktionen unterstützt, die das Testen mit Remote-Desktop-Agenten ermöglichen. Aspekte des Testens und des Debuggings behandeln wir in Kapitel 8, »Testen und Debugging«.

### 3.3 Cloud Studio

Die Webanwendung SAP Intelligent RPA Cloud Studio unterstützt den Low-Code- und den No-Code-Ansatz für den Entwurf von Bots zur Anwendungs- und Prozessintegration.

Mit dem *Low-Code-Ansatz* können sowohl RPA-Entwicklerinnen und -Entwickler als auch Mitarbeitende aus den Fachabteilungen, die über weniger technisches Verständnis verfügen, Softwareroboter erstellen und ausführen. Dazu werden die Automatisierungs-Workflows als grafische Modelle aus vordefinierten Komponenten zusammengestellt. Abbildung 3.6 zeigt einen Workflow in der Modellierungsumgebung des Cloud Studio. Bei Bedarf können die vordefinierten Schritte ebenso wie im Desktop Studio mit benutzerdefiniertem Code erweitert werden.

Bei dem *No-Code-Ansatz* werden die Anwenderinnen und Anwender durch eine intuitive Oberfläche geführt, die ihnen ein einfaches Workflow-Design ermöglicht. Die einfach zu bedienenden Bot-Entwicklungsfunktionen, die im Cloud Studio angeboten werden, ermöglichen es sowohl erfahrenen Entwicklerinnen und Entwicklern als auch Fachanwenderinnen und -anwendern, Projekte zu erstellen und zu bearbeiten, ohne selbst eine Zeile Code schreiben zu müssen.

Codegenerierung

Code Editor und Debugger

Low-Code-Ansatz

No-Code-Ansatz

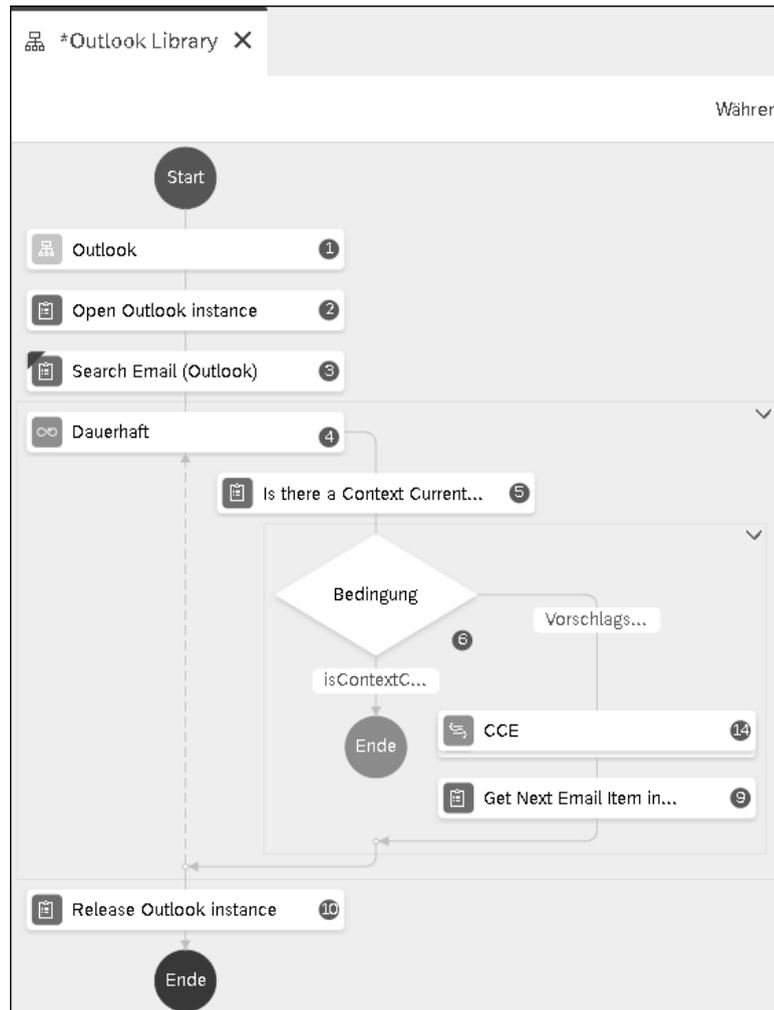


Abbildung 3.6 Beispielhafter Workflow-Überblick im Cloud Studio



**Wiederverwendung von Desktop-Studio-Paketen**

Das Cloud Studio ermöglicht auch die Wiederverwendung von Szenarien und Datentypen, die im Desktop Studio erstellt wurden. Diese Szenarien werden dazu kompiliert und in Form von Paketen exportiert, die in das Cloud Studio importiert werden können.

**3.4 Desktop Agent**

Die Komponente *Desktop Agent* wird von einer Administratorin oder einem Administrator auf den Desktop-Computern der Endanwenderinnen und -anwender, auf einem Server oder einem virtuellen Server installiert. Es handelt sich bei dem Desktop Agent um ein JavaScript-basiertes Framework.

Abbildung 3.7 stellt die Architektur dieses Frameworks schematisch dar. Der Desktop Agent verfügt über eine Reihe von Treibern zur Interaktion mit Anwendungen wie SAP-Systemen, Webseiten, benutzerdefinierten Anwendungen u. v. m. Der Desktop Agent kann sich darüber hinaus mit Application Programming Interfaces (APIs), Dateisystemen und Datenbankschnittstellen verbinden, um die definierten Aufgaben auszuführen und Ergebnisse abzurufen. Es ist auch möglich, eine Verbindung mit intelligenten Cloud-Services von SAP herzustellen, z. B. mit *SAP Conversational AI* und *SAP Internet of Things (SAP IoT)*.

Architektur des Desktop Agent

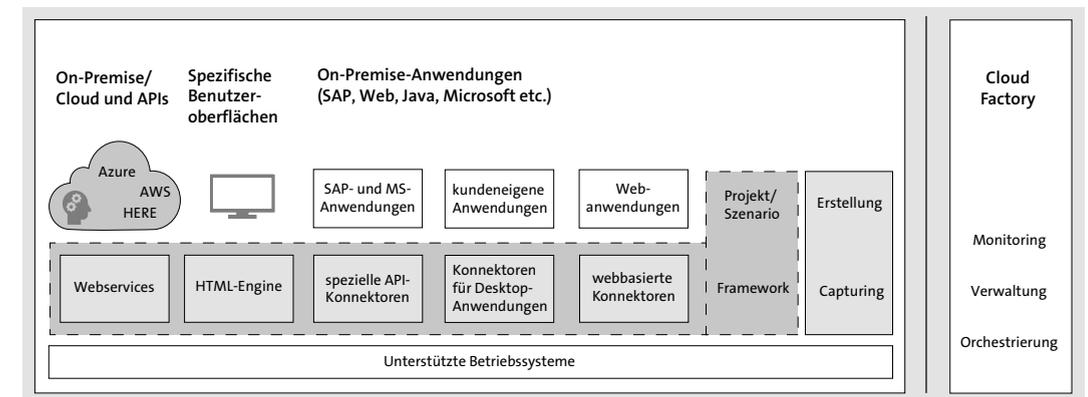


Abbildung 3.7 Aufbau des Desktop Agent

Operationen wie das Anklicken von Benutzeroberflächenelementen, das Navigieren und das Manipulieren von Daten können vom Desktop Agent ausgeführt werden. Das *Desktop-Framework* bietet Ihnen die Möglichkeit, eine vereinfachte Benutzeroberfläche zu erstellen, die über der eigentlichen Benutzeroberfläche Ihrer Geschäftsanwendung auf dem Bildschirm zu sehen ist. Dies ist beispielsweise dann hilfreich, wenn die eigentliche Eingabemaske sehr komplex ist und eine vereinfachte Bedienung ermöglicht werden soll. Dies erhöht die Systemakzeptanz und reduziert Einarbeitungszeiten und Erfassungsfehler. Der Desktop Agent ist in der Lage, Elemente der Benutzeroberfläche wie Schaltflächen oder Labels in die automatisierten Geschäftsanwendungen zu integrieren, ohne deren Code zu verändern.

Overlay-Benutzeroberfläche

Wie Sie Ihr Automatisierungsprojekt mit dem Desktop Agent ausführen, beschreiben wir ausführlicher im gleichnamigen Abschnitt 9.2.3.

### 3.5 Cloud Factory

#### Orchestrierungs- und Überwachungsschicht

Die *Cloud Factory* ist eine Cloud-basierte Orchestrierungs- und Überwachungsschicht für eingesetzte Bots. Die Cloud Factory wird auf der SAP BTP betrieben. Sie ist die zentrale Komponente für die Überwachung und Steuerung der Jobs und Agenten innerhalb einer Landschaft. Im Rahmen von SAP Intelligent RPA hat sie folgende Aufgaben:

- Koordination der Agenten zur Ausführung vollautomatischer Prozesse
- Remote-Auslösen von Prozessen auf Desktops, Servern oder virtuellen Maschinen
- Planen und Überwachen der Bot-Ausführung

#### Organisation der Cloud Factory

Die Cloud Factory ist in Umgebungen organisiert. Dabei werden typischerweise eine Entwicklungs- (DEV), eine Test- (TEST) und eine Produktivumgebung (PROD) eingesetzt. Benutzerinnen und Benutzer sowie Maschinen werden in Hierarchien geordnet, um festzulegen, welche Personen und welche Maschinen welchen Bot ausführen dürfen. Sie können einen Bot während seiner Ausführung überwachen, um eine erfolgreiche Automatisierung des Geschäftsprozesses entsprechend Ihren Vorgaben sicherzustellen.

#### Verwaltete Elemente

Im Kern befasst sich die Cloud Factory mit der Verwaltung von vier wesentlichen Elementen, die hierarchisch miteinander zusammenhängen:

##### ■ Verwaltung von Agentenhierarchien

Eine strukturierte Menge von Maschinen und Agenten, die für den Betrieb der Bots zuständig sind, werden in der Cloud Factory in sogenannten *Hierarchien* definiert.

Mit Hierarchien können Sie den Bots Zeitfenster zuweisen (siehe Abbildung 3.8), in denen sie im Attended- oder Unattended-Modus laufen sollen. Dies kann je Agent unterschiedlich erfolgen. Sie können bestimmte Bots beispielsweise so planen, dass sie während der Nacht oder außerhalb von Stoßzeiten ausgeführt werden, um die Arbeitslastverteilung Ihrer Organisation zu berücksichtigen.

Darüber hinaus ermöglichen Agentenhierarchien die hierarchische Organisation von Computern und Anwenderinnen und Anwendern, indem Anforderungen über Eltern-Kind-Beziehungen definiert werden.

Dies dient zur Steuerung der Zugriffskontrollen. Zugriffskontrollen, die auf einer übergeordneten Ebene definiert wurden, werden nach unten vererbt und gelten für alle Kindelemente. Ein *Hierarchieknoten* kann in mehreren Umgebungen verwendet werden.

The screenshot shows a form titled 'Bearbeiten Auslöser hinzufügen' (Edit Trigger). The form contains the following fields:

- \*Name: Bearbeitet\_01
- Beschreibung: z.B. Bearbeiteter Auslöser
- \*Desktop-Paket: PoC\_Order\_Process\_Legacy
- \*Datumsbereich: 13. Nov. 2020 to 20. Nov. 2020
- \*Zeitzone: (UTC+01:00) Amsterdam, Berlin, Bern, Rom, Stockhc

Below the form is a calendar grid showing execution time windows for each day of the week. The grid has columns for hours from 12 AM to 12 PM. The 'Anlegen' (Create) button is highlighted.

Abbildung 3.8 Zuweisung von Ausführungszeitfenstern über Hierarchien

##### ■ Verwaltung von Umgebungen

Umgebungen (engl. *Environments*) ermöglichen Ihnen die Definition und Verwaltung von Entwicklungs-, Test- und Produktionsumgebung. Es gibt keine Begrenzung für die Anzahl oder Typen der Umgebungen, die definiert werden. Umgebungen unterstützen benutzerdefinierte Variablen und sind mit den Agentenhierarchien verknüpft, um eine granulare Steuerung der Agentenbereitstellung zu ermöglichen. Eine Umgebung kann mehrere Projekte und Pakete enthalten.

### ■ Verwaltung von Paketen

Aus dem Desktop Studio exportierte Pakete werden zur Konfiguration, Ausführung und Überwachung in die Cloud Factory importiert. Jedes Paket kann mehrere Konfigurationen unterstützen, die den Ausführungsmodus (attended oder unattended), die Agentenumgebung für die Ausführung sowie Datums- und Zeitbereiche festlegen, in denen die Ausführung erfolgen kann. Ein Paket kann mehrere Konfigurationen enthalten.

### ■ Verwaltung und Überwachung von Jobs

Die Cloud Factory ermöglicht die Überwachung der Job-Ausführung, u. a. des Fertigstellungsstatus, der Ausführungszeit, der verbundenen Agenten und deren Status. Außerdem können Sie hier eine Historie der ausgeführten Jobs auswerten.

Eine Konfiguration kann im Laufe der Zeit eine beliebige Anzahl von Jobs haben.

Wie Sie Pakete in die Cloud Factory importieren und Ihre Automatisierungsprojekte verwalten, beschreiben wir ausführlicher in Kapitel 9, »Deployment und Ausführung von Bots«.



#### Vorkonfigurierte Bots und Pakete aus dem Store

Der *SAP Intelligent RPA Store* ist global verfügbar und direkt von der Cloud Factory aus für alle registrierten Nutzerinnen und Nutzer zugänglich. In einer von SAP bereitgestellten Bibliothek haben Sie hier umfassenden Zugriff auf vordefinierten und konfigurierten Content für die Automatisierung Ihrer Projekte. Dies kann die Produktivität von Projektautomatisierungskonfigurationen deutlich erhöhen. Eine Suchfunktion, verschiedene Filter und Inhaltsbeschreibungen helfen Ihnen bei verschiedenen Problemlösungen.

## 3.6 Zusammenwirken der Komponenten

Phasen eines Bot-Entwicklungsprojekts

Automatisierungsprojekte in SAP Intelligent RPA werden unter Verwendung der vier Hauptkomponenten von SAP Intelligent RPA für die Bot-Entwicklung, das Deployment und die Konfiguration sowie die Ausführung und Überwachung in einer definierten Reihenfolge entwickelt, ausgeliefert und überwacht (siehe Abbildung 3.9). In diesem Abschnitt werden die Phasen der Bot-Projektlieferung entlang dieser Komponenten erläutert.

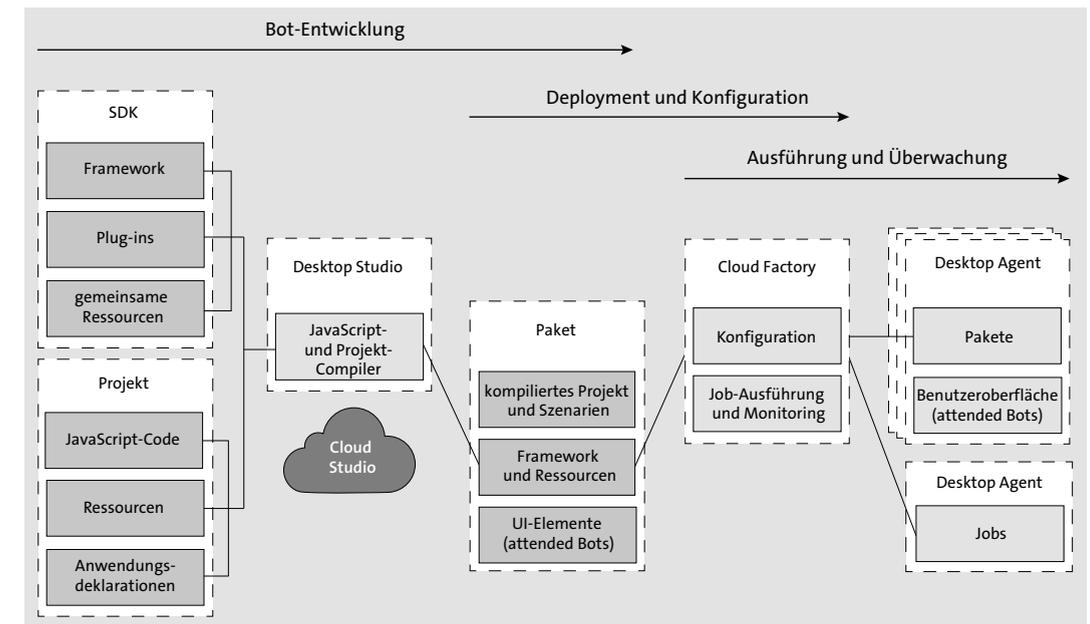


Abbildung 3.9 Hauptkomponenten von SAP Intelligent RPA

### 3.6.1 Bot-Entwicklung

Während der Bot-Entwicklungsphase verwendet das RPA-Entwicklerteam das Desktop Studio oder das Cloud Studio (siehe Abschnitt 3.2 und Abschnitt 3.3), um die zu automatisierenden Anwendungen zu erfassen und zu konfigurieren, Workflows zu definieren und Skripte zu ändern. Das Desktop Studio nutzt den Desktop Agent, um Projekte zu testen und zu debuggen. Ziel des Desktop Studio ist auch, die Bot-Entwicklung im Sinne von Low-Code oder No-Code einfacher zu gestalten. Zukünftig soll das Desktop Studio durch das Cloud Studio abgelöst werden, das heißt, die Entwicklungsumgebung wird vollständig in der Cloud betrieben und über einen Webbrowser aufgerufen.

Desktop Studio oder Cloud Studio

Die Bot-Entwicklung umfasst im Wesentlichen folgende Arbeitsschritte:

Schritte der Bot-Entwicklung

#### 1. Application Capturing

Erfassung der zu automatisierenden Anwendungen sowie deren Benutzeroberflächenelemente (Schaltflächen usw.), die bei der Automatisierung des Geschäftsprozesses durch den Bot verwendet werden sollen.

#### 2. Workflow-Design

Sobald die Anwendungen erfasst sind, können Workflows erstellt werden, die die sequenzielle Interaktion des Bots mit den Anwendungen beschreiben, einschließlich Schritten wie der Anmeldung bei der Anwen-

derung, der Auswahl von Elementen auf der Benutzeroberfläche, der Erfassung und Verwendung von Variablen usw.

### 3. Testen und Debugging

Die Entwicklungsumgebung verwendet den Desktop Agent, um Automatisierungen lokal oder remote zu testen und den Bot zu debuggen. Dies ermöglicht es Ihnen, die Automatisierung zu validieren, bevor sie paketiert und in der Cloud Factory bereitgestellt wird.

### 4. Paket-Export

Ein Automatisierungsprojekt im Desktop Studio enthält die Anwendungsdeklaration, das Workflow-Design, die Framework-Komponenten, die gemeinsamen Ressourcen und vieles mehr. Sobald die Entwicklung abgeschlossen ist, werden alle notwendigen Bestandteile zu einem Paket zusammenfasst. Das Paket kann nun in eine Archivdatei exportiert werden, die anschließend in die Cloud Factory importiert werden kann.

## 3.6.2 Deployment und Konfiguration

Sobald das Paket für die Bereitstellung in der Cloud Factory erstellt wurde, konfiguriert das RPA-Entwicklerteam die Eigenschaften des Pakets, z. B. die Versionsnummer. Im nächsten Schritt wird das Paket aus dem Desktop Studio exportiert, in die Cloud Factory importiert und dort weiter konfiguriert.

Deployment und Konfiguration umfassen im Wesentlichen folgende Arbeitsschritte:

#### 1. Umgebungen und Agentenhierarchie

Umgebungen und Agentenhierarchien müssen für jede Umgebung definiert werden, in der die Desktop-Agenten Automatisierungsschritte ausführen sollen. Diese müssen nur bei Bedarf angepasst werden, beispielsweise bei Änderungen in der IT-Landschaft, die sich auf die Hierarchien oder Umgebungen auswirken.

#### 2. Paket-Import

Das aus dem Desktop Studio exportierte Paket wird in die Cloud Factory importiert. Während des Imports wird die Integrität des Pakets überprüft.

#### 3. Paket-Konfiguration

Nach dem Import muss mindestens eine Konfiguration für das Paket erstellt werden, die unter anderem Anmeldeinformationen wie Benutzernamen und Passwörter sowie die Zeitplanung enthält.

Schritte für  
Deployment und  
Konfiguration

## 3.6.3 Ausführung und Überwachung

Nach der Konfiguration ist das Paket bereit zur Ausführung. Paket-Konfigurationen können sowohl für beaufsichtigte (attended) als auch für unbeaufsichtigte (unattended) Automatisierungsszenarien definiert werden:

- Attended Szenarien erfordern eine Benutzerinteraktion: Sobald das Paket an einen Agenten in einer Umgebung verteilt wird, wird ein Desktop-Menü angezeigt, über das die Benutzerinnen und Benutzer mit dem Agenten interagieren oder beispielsweise einen Workflow starten können.
- Unattended Szenarien werden automatisch auf Basis des in der Konfiguration festgelegten Zeitplans ausgeführt, können allerdings auch manuell gestartet werden.

Die Ausführung und Überwachung einer Automatisierungslösung umfasst im Wesentlichen die folgenden Arbeitsschritte:

#### 1. Job-Ausführung

Konfigurierte Pakete werden zur Ausführung an Instanzen des Desktop Agent verteilt. Je nach Konfiguration kann ein Job automatisch, z. B. per Zeitschaltuhr, ausgeführt werden oder darauf warten, dass eine Benutzerin oder ein Benutzer ihn auslöst.

#### 2. Job-Überwachung

Der Desktop Agent führt den Auftrag aus. Daraufhin werden der Ausführungsstatus und die Ergebnisse an die Cloud Factory zurückübermittelt, wo sie auch überwacht werden können. Jobs über alle bereitgestellten Agenten hinweg können im Monitoring-Bereich der Cloud Factory überwacht werden, beispielsweise der Job-Status oder aufgetretene Fehler.

#### 3. Agentenüberwachung

Der Status aller eingesetzten Agenten kann in der Cloud Factory transparent und auf einen Blick dargestellt werden. Autorisierte Benutzerinnen und Benutzer können Agenten bei Bedarf direkt abschalten.

Wie Sie sehen, orientiert sich das Zusammenspiel der drei Hauptkomponenten von SAP Intelligent RPA am Prozess zur Entwicklung, Bereitstellung und Überwachung von Softwarerobotern. Jede der drei vorgestellten Komponenten hat einen klar abgegrenzten Einsatzzweck. Das integrierende Element zwischen allen drei Komponenten stellt das Paket dar, in dem alle für das jeweilige Prozessautomatisierungs-Szenario erforderlichen Informationen gebündelt sind.

Schritte für  
Ausführung und  
Überwachung

### 3.7 Zusammenfassung

In diesem Kapitel haben Sie das RPA-Tool SAP Intelligent Robotic Process Automation kennengelernt. Wir haben die Hauptkomponenten des Tools vorgestellt und ihre jeweiligen Funktionen und Bestandteile erläutert. Die wichtigsten Hauptkomponenten sind das Desktop Studio, das Cloud Studio, der Desktop Agent und die Cloud Factory. Sie müssen nicht nur einzeln funktionieren, sondern auch miteinander interagieren, um eine lauffähige Automatisierung mit SAP Intelligent RPA zu erstellen.

Nachdem Sie nun einen spezifischen Überblick über SAP Intelligent RPA gewonnen haben, widmet sich das folgende Kapitel 4, »Capturing und Deklaration von Anwendungen«, das gleichzeitig auch den Beginn von Teil II des Buchs darstellt, Schritt für Schritt der konkreten Entwicklung eines Bots.