Solution Architect

Understanding the RPA Solution Architect role

Key responsibilities

- RPA Solution Architect definition
- Governance of the end-to-end performance of the agreed solution
- Automation process optimization
- Effort estimation
- Code review
- Workflow component and reusable definition
- PDD and DSD sign-off
- Number of robots used, config file, asset, queue, and schedule definition
- Logging and Reporting - Dashboards

TRAINING

Background and skillset

- Minimum 5 years programming experience in .NET (C#, C++or VB), Java
- Minimum 2 - 5 years experience in the service industry or business setup
- Infrastructure knowledge, including servers, storage, firewalls, load balancers, routers, etc.
- Strong conceptual and analytical skills, results orientation
- Ability to develop solution architecture designs
- Team player with leadership and cross-team collaboration experience

Stages of an RPA Project

01 Enable

Infrastructure Setup

- Designing the server architecture
- Installing and configuring the architecture
- Setting up dev, test & production environments

02 Preparation

Project Governance

- Agreeing on the project development approach
- Reviewing the RPA best practices

03 Design

Workflow Design

- Filling in the Process Design Document (PDD)
- Creating test cases and data
- Designing the solution

04 Build

Workflow Development

- Building the workflows
- Performing Unit and Functional testing
- Creating the Development Specification Document (DSD.)

05 Test

Quality Assurance

- Executing the test cases
- Reporting the results
- Making the Go/NoGo decision

06 Sustain

Hypercare

- Performing workflow support
- Managing Changes and Improvements

Responsibilities and ownership of the SA

- Infra Initial Setup
- Orchestrator deployment options
- RPA Deployment Environments

- Best Practices agreement
- Coding standards
- Robotic Enterprise Framework
- Collaboration with BA on feasibility and optimization
- Development effort estimation
- Technical meetings planning

- PDD sign-off
- Overall solution
- Component splitting
- Developer appointment
- Reusable component identification
- Configurations, queues, schedules
- Outstanding challenges documentation

- Extra requirement list after PDD sign-off
- Constant mentoring of Devs
- Troubleshooting and debugging
- Logging and reporting
- Source control solution owner

- Code review and audit
- DSD sign-off
- Functional testing sign-off

- Change documentation
- Monitoring
- Support

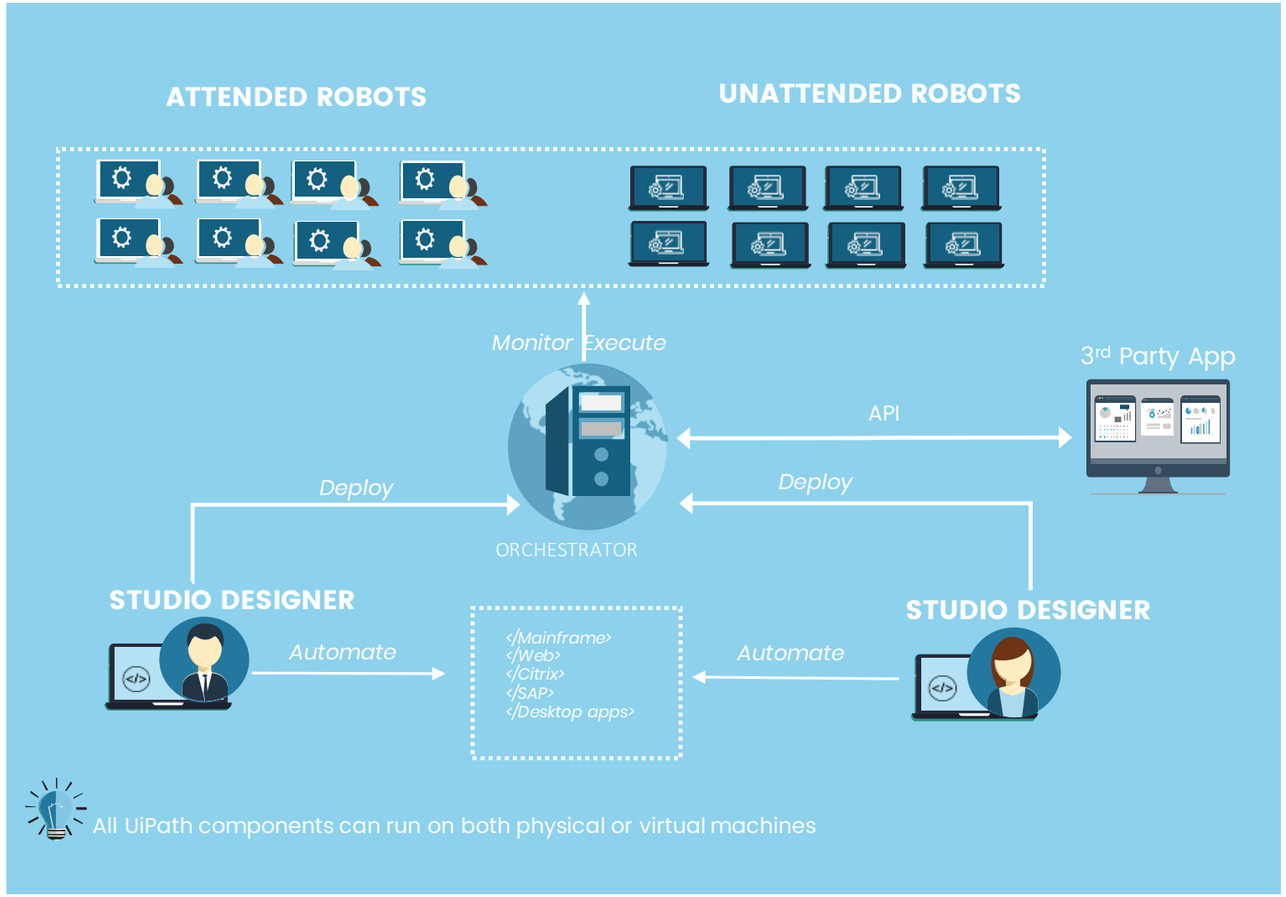

UiPath
Thank You!
academy.uipath.com

Solution Architect

Product Architecture

UiPath High Level Architecture

The UiPath Orchestrator manages robots specifically developed for customer front and back office processes.



1. Attended Robot: delivers low costs and higher performance with front office agent- supporting automation features.

2. Unattended Robot: these robots utilize unattended automation to run high back office transaction volumes in batch mode.

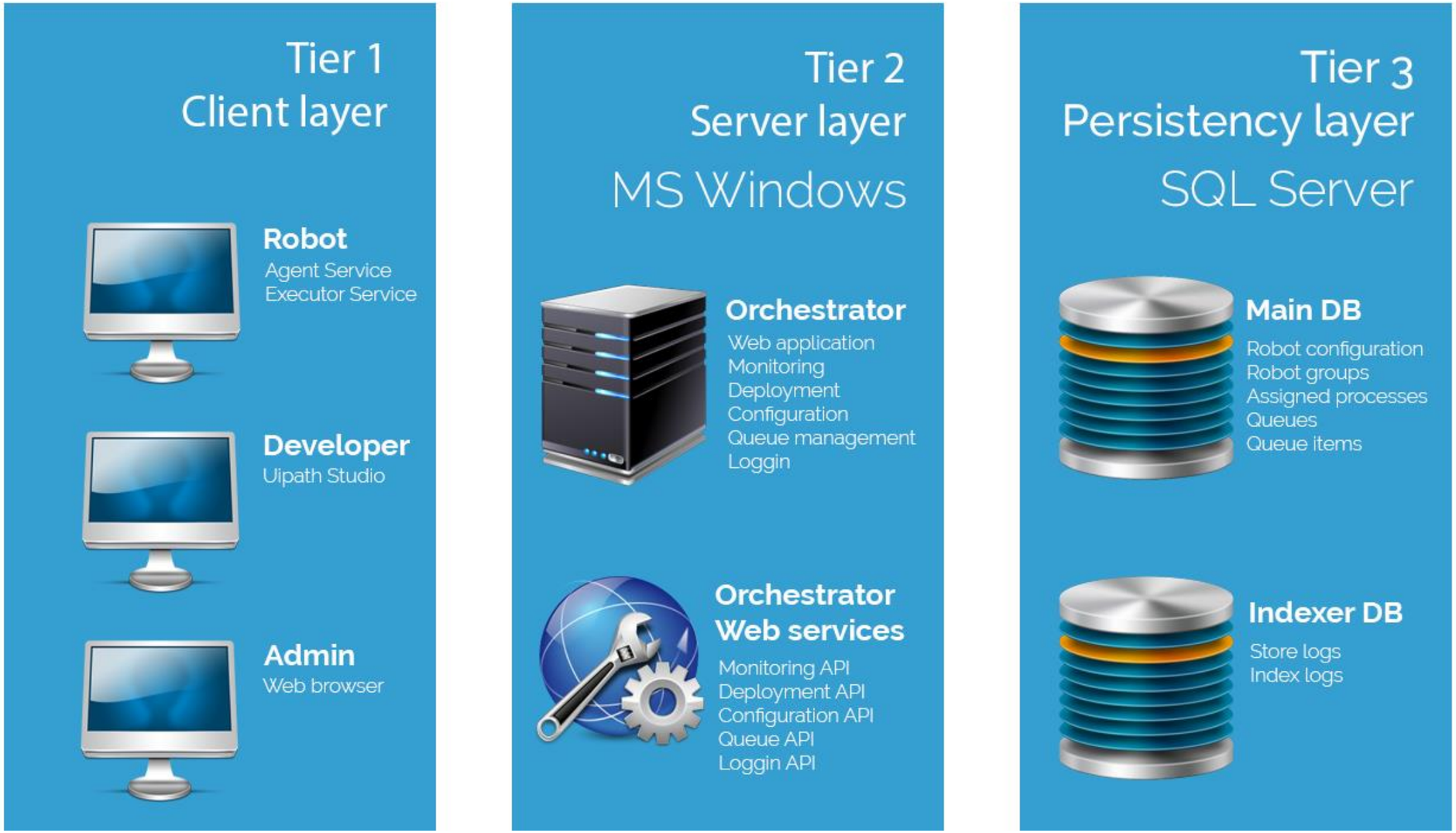
3. UiPath Orchestrator: Enterprise architecture server platform supporting: release management, centralized logging, reporting, auditing and monitoring tools, remote control, centralized scheduling, queue/robot workload management. assets management.

4. UiPath Studio: enables users to automate with highly intuitive tools (not code): process recorders; drag & drop widgets & best practices templates.

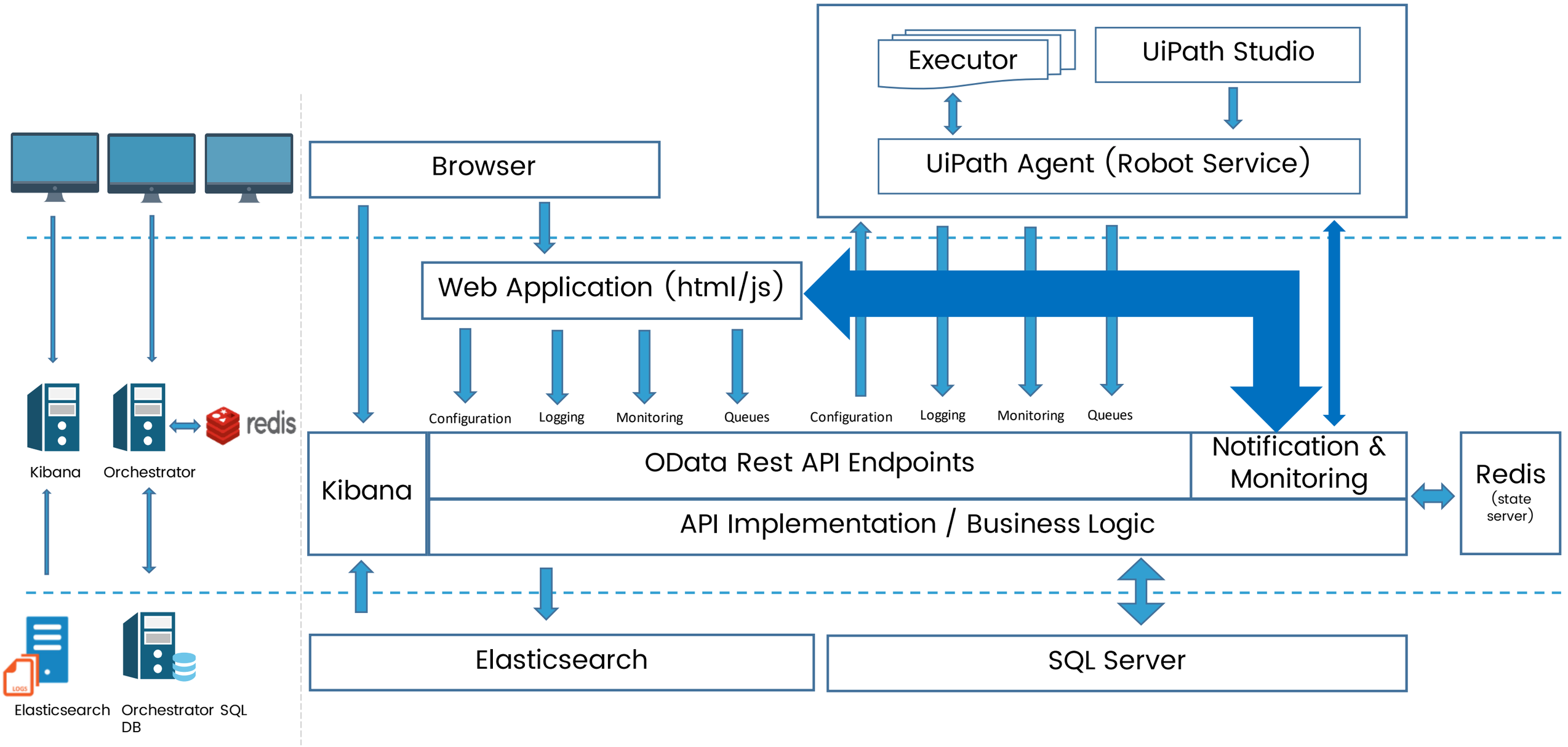
All UiPath products and features reside within a platform architecture designed to provide strong security, enterprise grade compliance & robust governance.

UiPath Architecture

High Level view of the Server Architecture



Component Diagram



Attended or Unattended Robot?

ATTENDED ROBOT

- **Definition:** A robot that works with human agents side by side, and assists them in completing processes automatically
- **General use**: in manual, repetitive, highly rule based activities containing decision points that require human intervention, due either to the necessity of pure judgmental calls or to the high complexity and volatility of process inputs
- **Best fit for:** Service Desks, Helpdesks and Call Centers
- **Communication with Server**: bi-directional
  - *Robot to Server:* execution logs, automated process upload
  - *Server to Robot:* automated process version deployment **only**
- **Features**:
  - release management (automatic update/rollback)
  - agent assisted mode
  - centralized logging, reporting and auditing tools
  - queue/robot workload management
  - asset management

Attended robots share the desktop with a human user. They can only be triggered manually through a human action on the local machine, and do not support remote running or scheduling.

UNATTENDED ROBOT

- **Definition:** A robot that works in an unattended manner, independently of any human action
- **General use:** in manual, repetitive, highly rule based back office activities not requiring any human intervention
- **Best fit for:** *a*ny type of back office activity prone to automation
- **Communication with Server**: bi-directional
  - *Robot to Server:* execution logs, automated process upload, robot status
  - *Server to Robot:* automated process version deployment, start or reset processes
- **Features**:
  - release management (automatic update/rollback);
  - centralized logging, reporting, auditing and monitoring tools
  - remote control
  - centralized scheduling
  - queue/robot workload management
  - asset management

Unlike Attended robots, Unattended robots can be triggered remotely, directly from the server.


UiPath
Thank You!
academy.uipath.com

Solution Architect

Enable – Prepare RPA

academy.uipath.com

RPA Operations COE

Security Design	Operations Design	Implementation Design	Platform Design	Governance Modeling	Support Modeling
• Existing Client Enterprise Security Architecture compliance • Existing Client Data Securiy definitions • Existing Client Infrastructure Security definitions • Client Applications Vulnerability and Penetration definitions and Standards • Existing User Management and Access Management Architecture • Application Credentials and Access Management guidelines for underlying subsystems • Risk Management strategy definition	• Existing Operations SOP • Existing Operations Execution design (how operation schedules are defined, how BCP is designed, process criticality definitions etc) • Operations Roles and Responsibilities definition • Existing accepted Operations APT definitions • Existing Operations Execution State and Stage Management definitions • Existing Operations Transcription validation guidelines	• Pilot Use Case Definition • Application Feasibility Analysis and documentation • Implementation Solution Architecture • Implementation Approach and Atomicity definition • Implementation design methodology • Code Repository and Version Control standard definition • Code Migration/Deployment strategies • Testing Methodology definition	• Infrastructure definition (VM/VDI, W in Workstation vs W in Server OS etc) • Infrastructure deployment and management guideline definition • Infrastructure Scalability and Availability design definition • Infrastructure Access Management and User Control definitions • DR and BCP design • Redundancy model • Load Balancing strategy definition • Failover strategy definition	• Compliance definition • RACI definition • Approval Matrix definition • Process Analytics definitions • Process Monitoring and Control definitions • Performance Monitoring and Improvement Cycle definition	• Change Management process definition • Support SLA definition • Change and Release Management strategy definition • Communication Matrix definition

Enterprise Integration Considerations

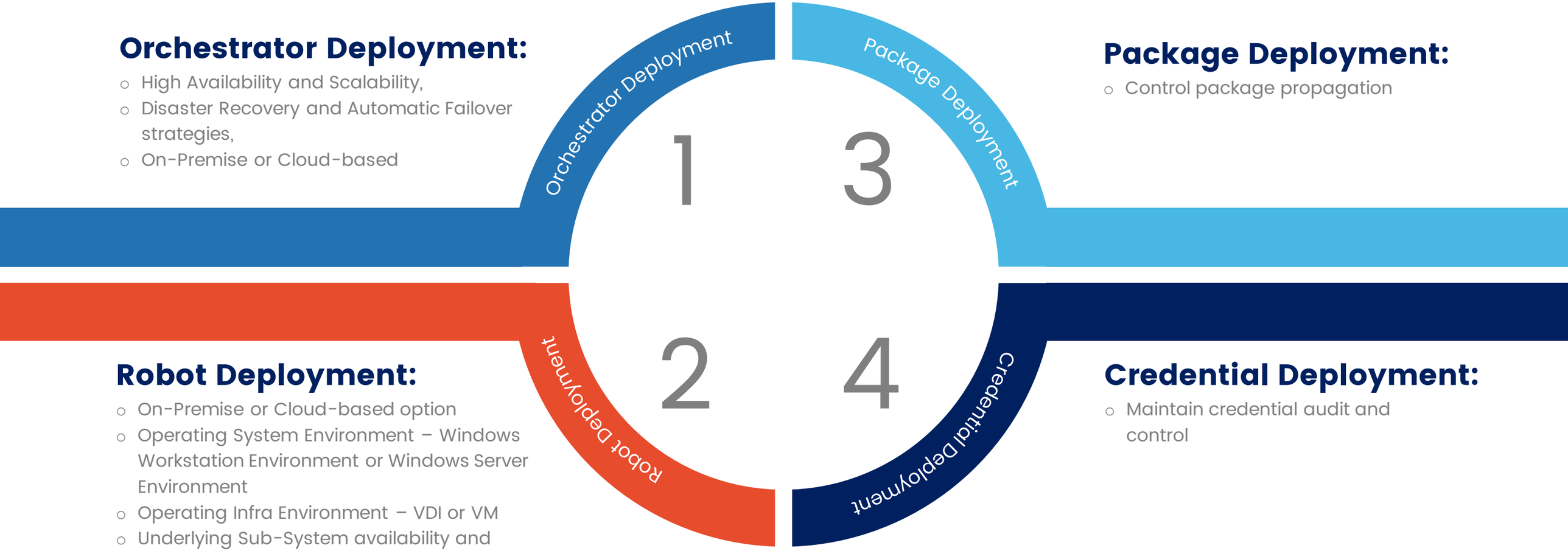
RPA Solution Design breakdown

RPA Deployment considerations

From a deployment standpoint, there are four major components:

Orchestrator Deployment:

- High Availability and Scalability,
- Disaster Recovery and Automatic Failover strategies,
- On-Premise or Cloud-based



Package Deployment:

- Control package propagation

Robot Deployment:

- On-Premise or Cloud-based option
- Operating System Environment – Windows Workstation Environment or Windows Server Environment
- Operating Infra Environment – VDI or VM
- Underlying Sub-System availability and integration
- Ease of upscaling during peak loads and off-peak downscaling

Credential Deployment:

- Maintain credential audit and control


UiPath
Thank You!
academy.uipath.com

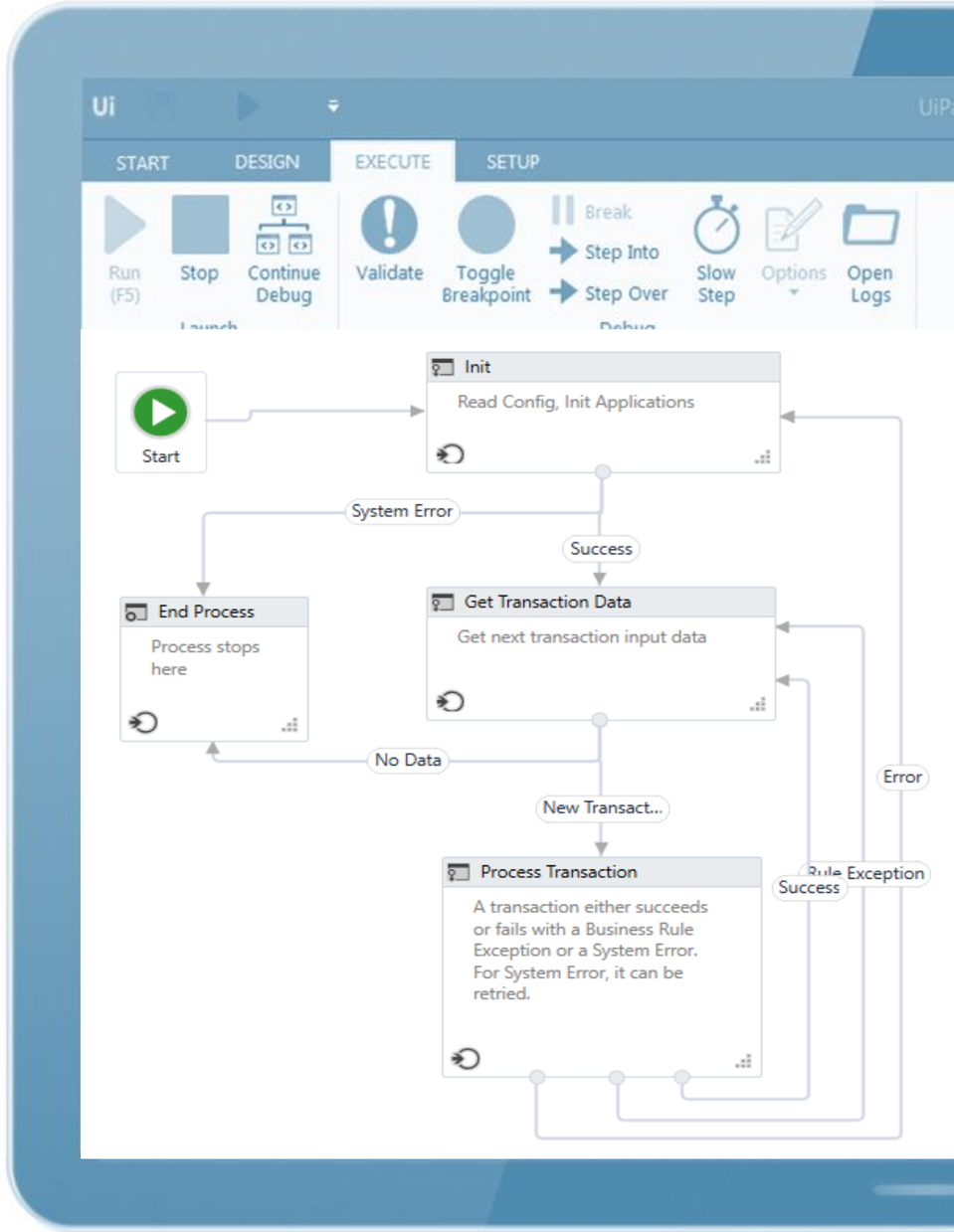
Solution Architect

Preparation – Project Governance

RPA Development Best Practices

The RPA SA team agrees on the Best Practices for RPA Projects Coordinated by the Implementation Lead

- REFrameWork
- Workflow layout (Sequence, Flowchart)
- Naming strategy
- Variable scope
- Comment and Annotation strategy
- UI Automation - Input and Output
- UI Automation – UI Synchronization
- Selectors
- Containers
- Error Handling
- Clean workflows
- Confidential data usage

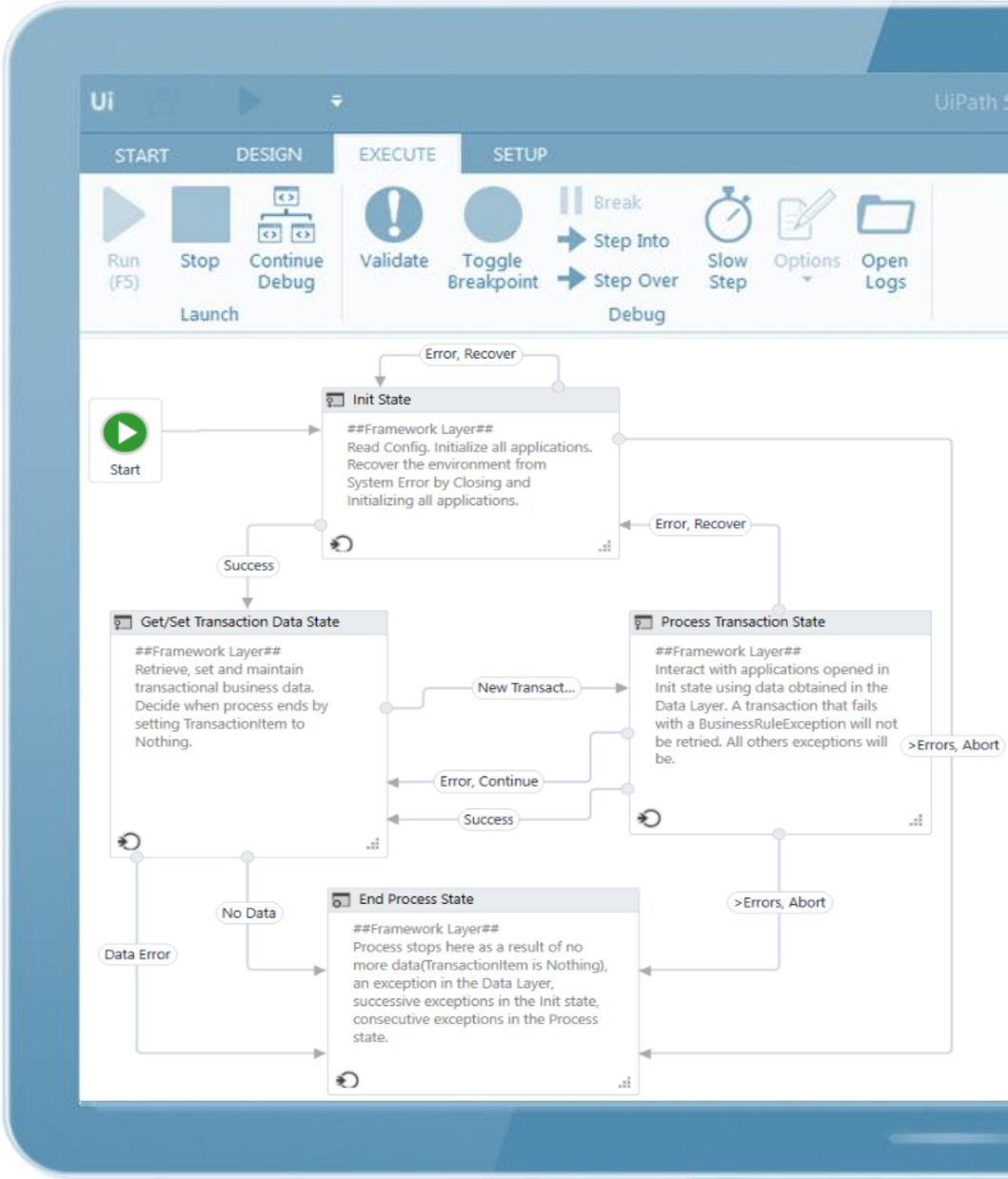


Enhanced REFramework

Extra features over the REFramework

- Principled layer design
- Clear separation between data and process layers
- Fixed “system reserved” components and activities
- InitState Retry
- Abort on max errors
- Workblock concept
- Enhanced hierarchical logging

  and audit tracing
- Configurable logging suppressor and exception handling
- Enhanced automatic testing
- More configurations - added the JSON Config option
- Unified Dispatcher, Performer, and Hybrid
- State machine task as a service template
- Easy migration from REFramework

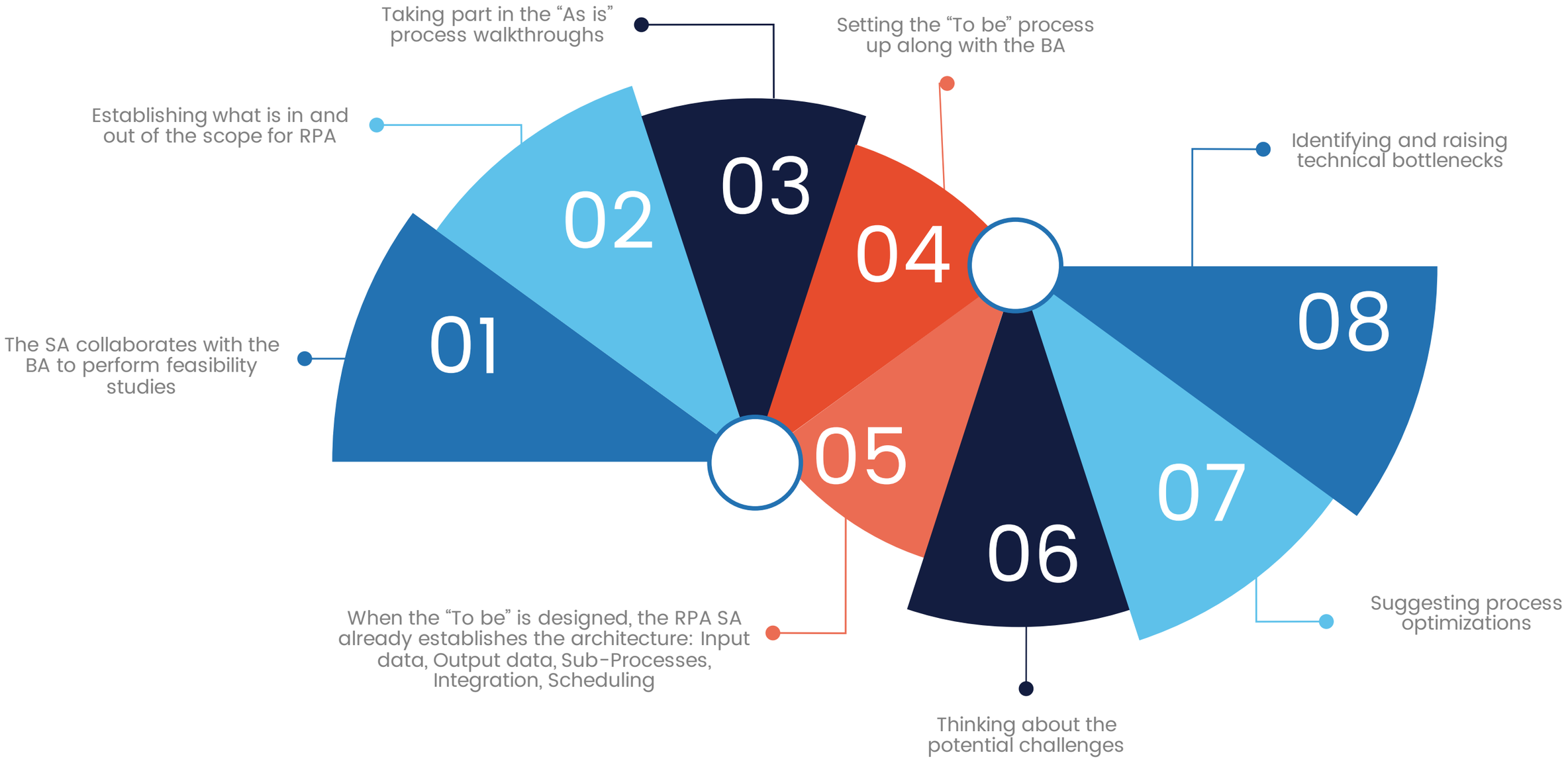



UiPath
Thank You!
academy.uipath.com

Solution Architect

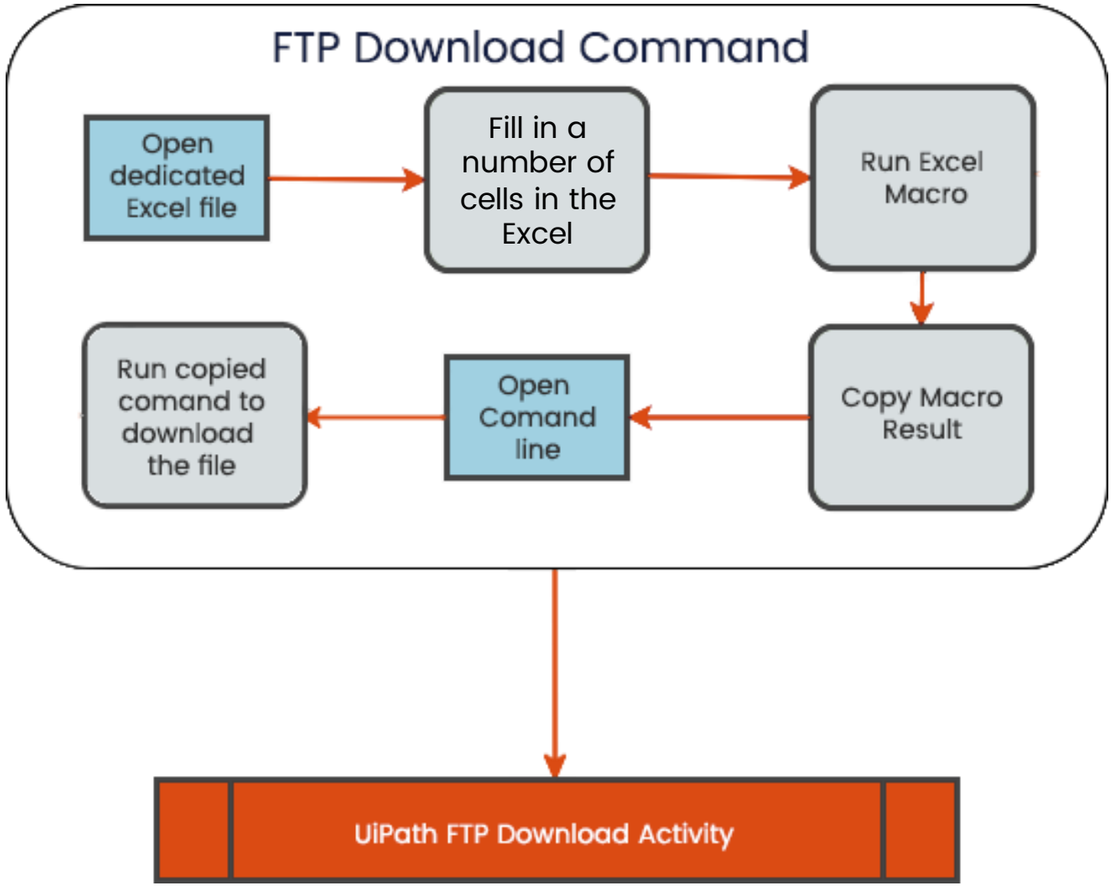
Preparation – Project Governance

RPA Feasibility study and Process optimization



Process optimization - FTP Download Example

- **As is Process:**
  - Open an Excel file used for generating the command
  - Fill the local and remote file path, the username, and the password. Some inputs are hardcoded in the file
  - Run Excel Macro
  - Copy the generated result on the clipboard
  - Open CMD
  - Run command to download the file
- **To be Process (optimized):**
  - UiPath FTP Download Activity




UiPath
Thank You!
academy.uipath.com

Solution Architect

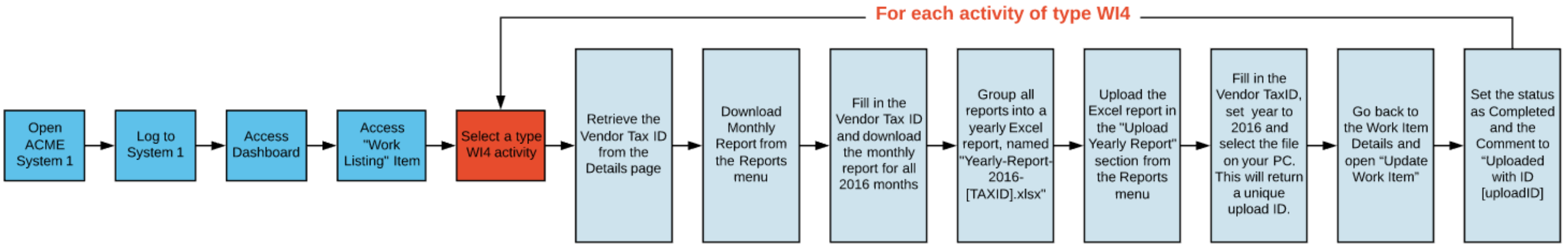
Preparation – Project Governance

Development effort estimation - Guidelines

- Effort estimation needs to be conducted in the analysis phase
- The RPA SA should thoroughly understands the process and collaborate with the BA and PM
- High-level process breakdown requires individual estimation
- The SA should identify the potential challenges.
- Preliminary integration should be tested - applications and particular screens with UiPath Studio
- The complexity of the applications and business rules should be taken into account when handling exceptions

- The level of your RPA developers should be taken into account
- Studio workflow creation, Orchestrator configurations, and dashboards should be included
- Unit and functional testing should be considered
- Additional change requests after the PDD sign off should not be considered. In case that happens, more time is included
- Diminishing returns
- RPA Projects are extremely hard to estimate, as many challenges arise during development. Additional time should be considered – typically 30% or more

Development effort estimation – Example 1



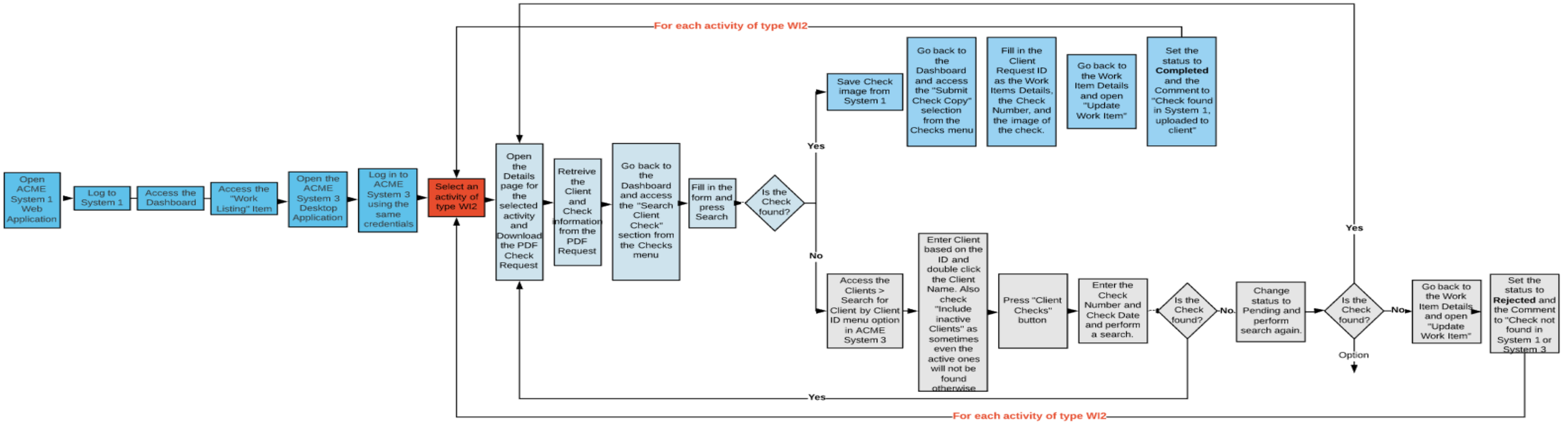
- Number of sub-processes: 2 (Dispatcher and Performer)
- Number of applications used: 2 (ACME System 1, Excel)
- Process complexity – Low (linear process, few rules)
- Some difficulty in downloading reports and dealing with temporary files
- Integration with ACME System1 App tested successfully
- Typical exception handling in the Performer

Estimation

Around 15 xaml files to be build and tested:

Sub-Process	Components	Estimation
Dispatcher	Login, Add to queue	2 days
Performer	Initialize	1 day
Performer	Report management	2 days
Performer	Navigation	1 day
Dispatcher / Performer	Integration, Functional Tests	3 days
Total Estimation	All + 30%	12 days

Development effort estimation – Example 2



- Number of sub-processes: 2 (Dispatcher and Performer)
- Number of applications used: 3 (ACME System 1, System 3, PDF Reader)
- Process complexity – Medium
- Integration with ACME System1 App tested successfully
- Integration with ACME System3 App tested successfully, with some challenges (dynamic UI elements)
- Reusable components required in both systems
- Strong exception handling mechanism. Robust solution required

Estimation

Around 30 xaml files to be build and tested:

Sub-Process	Components	Estimation
Dispatcher	Reuse	1 days
Performer	Reuse, Initialize	2 days
Performer	PDF Processing	3 days
Performer	Navigation	2 days
Performer	Check Search System 3	4 days
Dispatcher / Performer	Integration, Functional Tests	5 days
Total Estimation	All + 30%	22 days
Total Estimation	2 developers	24 days


UiPath
Thank You!
academy.uipath.com