**A blue logo with blue text

Description automatically generated**

**CASE STUDY**

**ON**

**RPA TECHNICAL ARCHITECTURE PROGRAM**

**Submitted By: Netlink Software Group America Inc.**

**UiPath Case Study: On-Premises Setup for 6 Automation Processes**

**Client Requirements:**

* 6 Automation processes:
  + **2 Complex** Processes
  + **2 Medium** Processes
  + **2 Simple** Processes
* **On-prem environment** setup:
  + 2 Unattended Bots
  + 2 Studio licenses
  + 1 Orchestrator (Version 2022.4.3)

**Overview of the Case Study:**

This case study will walk through an on-premises UiPath environment capable of handling six automation processes across different complexity levels. It also covers how the environment is designed to optimize for performance, security, and scalability while addressing the client's needs for two unattended bots and two development studio licenses.

**1. Understanding Automation Process Complexity**

Before diving into the architecture, it's essential to break down the complexity of the automation processes:

* **Complex Processes**: These typically involve multiple applications, require interaction with different systems (APIs, databases), and have intricate workflows with decision-making logic and exception handling.
* **Medium Processes**: These involve moderately complex workflows, such as integrating a few systems with straightforward logic.
* **Simple Processes**: These are straightforward automations, such as data entry or report generation, with minimal decision-making.

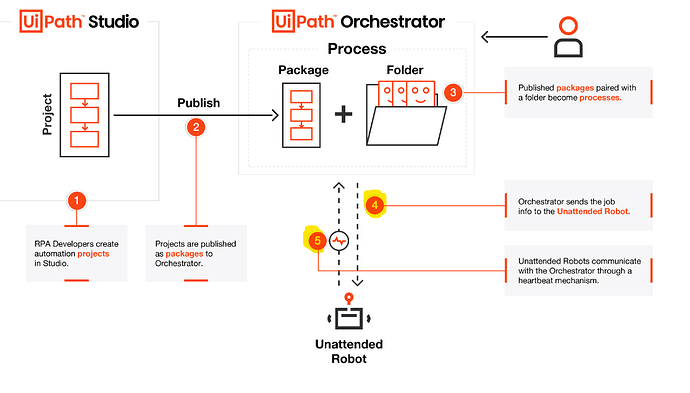
**2. Architecture Layout for On-Prem Environment**

Below is a high-level architecture that accommodates the client's requirements for on-premises deployment.

**Architecture Components:**

1. **Orchestrator (Version 2022.4.3)**:
   * Central control hub that manages the automation lifecycle.
   * Responsible for scheduling, managing, and monitoring the bots and processes.
2. **2 Unattended Bots**:
   * Run automation processes without human intervention, enabling 24/7 operations.
3. **2 Studio Licenses**:
   * Development environment where the automation workflows are designed and tested before being deployed to production.

**High-Level Architecture Diagram:**



**3. Breakdown of Architecture Components**

**1. Orchestrator (Version 2022.4.3)**

* **Purpose**: Orchestrator will handle all scheduling, monitoring, and management tasks for the bots running the automation processes.
* **High Availability**: For an on-premise setup, it's important to have failover options (e.g., backup Orchestrator setup or replication in case of failure).
* **Features Used**:
  + **Process Scheduling**: The two unattended bots will handle the automation processes based on a scheduled time or on-demand triggers.
  + **Queues**: For complex processes, job requests will be managed in **Queues**. Queues ensure better load balancing for Unattended Bots, allowing processes to be distributed efficiently.
  + **Logs & Monitoring**: Orchestrator stores logs and offers real-time monitoring of the bots. Error handling and troubleshooting are streamlined through centralized logging.

**2. Unattended Robots (2 Bots)**

* **Unattended Robot 1**:
  + Assigned to handle **simple** and **medium complexity** processes. This bot will run automated workflows like routine data entry tasks or medium complexity API-based interactions.
* **Unattended Robot 2**:
  + Will handle the **complex processes** as well as medium complexity workflows. These processes will likely involve integration with multiple systems, more decision logic, and exception handling.
* **Scalability**:
  + Even with 2 unattended bots, the architecture is designed to scale. More bots can be added to the environment in the future to support additional automation processes, allowing for horizontal scaling.

**3. UiPath Studio (2 Licenses)**

* **Studio 1**:
  + Primarily used by one RPA developer for designing and testing the simpler and medium complexity workflows.
* **Studio 2**:
  + Used for developing more **complex workflows**, involving API calls, advanced error handling, and integration with external systems like databases or ERP systems.
* **Git/Version Control**:
  + Both studios are connected to a version control system (e.g., Git) to ensure collaboration and manage changes in workflows.

**4. Environment Setup for Automation Process Execution**

**Unattended Bot Execution Strategy:**

* **Bot 1** will handle the simpler tasks such as:
  + Automating data entry into a CRM or ERP system.
  + Running scheduled reports for business units.
  + Updating records in a database from Excel sheets.
* **Bot 2** will focus on more complex jobs:
  + Integration tasks involving multiple APIs.
  + Automating multi-step workflows that involve ERP, CRM, and internal applications with exception handling.
* Both bots will also be configured to handle **medium complexity** tasks, such as periodic API calls to fetch reports, updating databases, or working with legacy systems through screen scraping.

**Handling Complex Processes:**

* For complex processes, bots will be scheduled to run during non-peak hours to ensure the systems they interact with are not overloaded.
* Exception handling mechanisms will be built into these workflows to ensure bots can handle errors and recover gracefully. The logs will be stored in Orchestrator for easy troubleshooting.
* These processes may also be set up to utilize **UiPath Queues** for optimized task handling and better load balancing between the two bots.

**Handling Medium Processes:**

* Medium complexity processes, such as data fetching and processing from web APIs or database interactions, will be run periodically.
* These processes will require moderate exception handling and will typically be more straightforward than the complex processes but still involve some system integrations.

**Handling Simple Processes:**

* The simple processes will likely involve repetitive tasks like data entry, report generation, or email automation, and can be run during the day when the load on the bots is lower.

**5. Scalability and Future Expansion**

The current architecture is designed to be **scalable**. As the automation program grows, additional unattended bots can be easily added to the environment by provisioning new machines or virtual machines.

* **Add Unattended Bots**: As new automation processes are identified or the existing processes become more complex, the architecture can be expanded by adding more robots.
* **Increase in Studio Licenses**: As more developers join the team, additional UiPath Studio licenses can be acquired to allow parallel development.
* **Multi-Orchestrator Setup**: For larger enterprises, the Orchestrator can be set up in an **active-active high-availability configuration** with a load balancer.

**6. Security Considerations**

**Access Control:**

* Implement **Role-Based Access Control (RBAC)** in UiPath Orchestrator to ensure only authorized users can access, modify, or execute automation processes.

**Data Encryption:**

* **SSL/TLS encryption** for all communications between the Orchestrator, robots, and Studio machines to ensure data security.

**Audit Logs:**

* Enable detailed audit logs within Orchestrator to track actions taken by users, developers, and robots. This is crucial for compliance and troubleshooting.

**7. Monitoring and Maintenance**

* **Orchestrator Monitoring**:
  + Use built-in monitoring tools within Orchestrator to track robot performance, job execution statuses, and errors in real time.
* **Bot Health Checks**:
  + Implement automatic health checks for robots to ensure they are online and operational at all times.
* **Logging & Alerts**:
  + Enable alerts within Orchestrator for error handling, so if any job fails, the team is immediately notified for resolution.

**Conclusion**

This proposed architecture is designed to provide a stable, scalable, and secure on-premises environment that meets the client’s requirements for automating 6 processes of varying complexity with two unattended bots, two development studios, and one Orchestrator. This structure ensures effective resource utilization, robust error handling, and secure operations while providing flexibility for future growth.