# **Interactive and Non-Interactive Applications**

**In Microsoft Intune, applications fall into two broad categories:**

* **Interactive** installs run in the context of the signed-in user and display progress bars or prompts as they execute. These are typically delivered through the Company Portal, allowing an end user to initiate or monitor the install.
* **Non-interactive** (silent) installs execute entirely in the system context via the Intune Management Extension (IME) with no UI. They’re ideal for required Line-of-Business (LOB) or Win32 applications you need to enforce across all devices without user interruption.

**Required and Available App Assignments**

* **Required** assignments push an app automatically to every targeted device or user; there’s no action required on their part.
* **Available** assignments simply make the app visible in the Company Portal; users choose to install it when—and if—they need it.
* You can even allow certain mobile apps to appear as “Available” on unenrolled devices for users with a valid Intune license.

**Groups, Dynamic Queries, Users**

1. **Static groups** with manually added members.
2. **Dynamic groups** driven by rules (e.g., “OS version greater than or equal to Windows 10”). Membership updates automatically as devices or users meet or fall out of the criteria.  
    All assignments—Required or Available—resolve against these groups whenever a device checks in.

**Process Flow for a Win32 App on Windows via the IME Service**

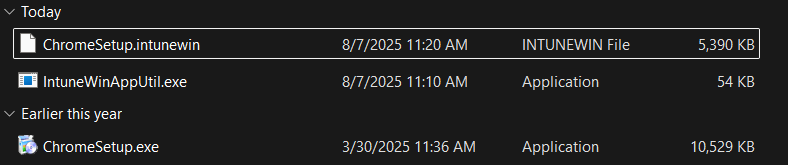
1. **Polling**

* Every hour (approx.), the Intune Management Extension on each client polls the Intune service.

1. **Detection**

* IME checks your predefined detection rules (registry keys, files, etc.) to see if the app is already installed.

1. **Download & Install**



* If the app isn’t detected and it’s Required, IME downloads the .intunewin package and silently runs your install command.

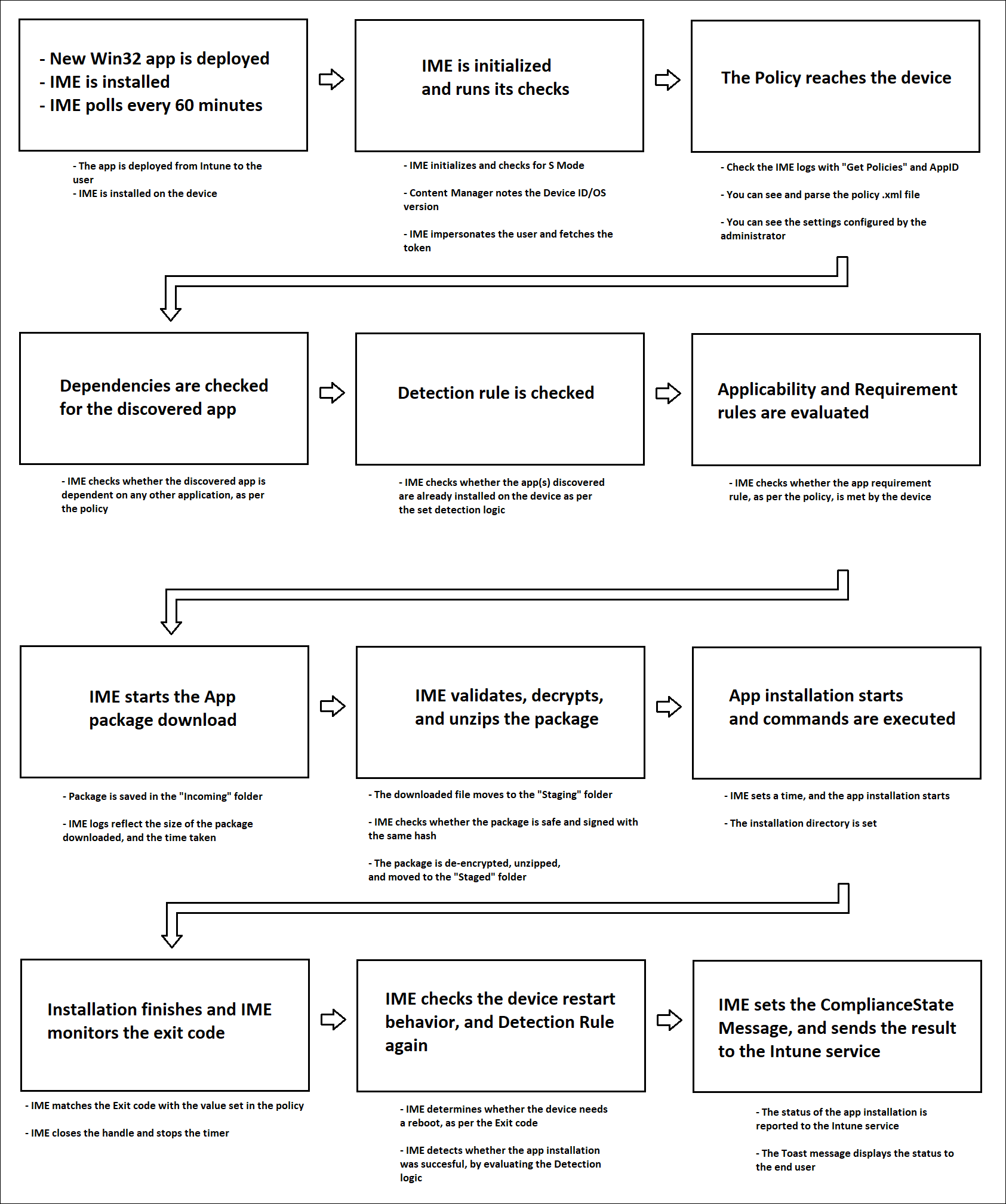
1. **Post-install Detection**

* IME re-evaluates the same detection rules to confirm a successful install.

**Toast Notification**

* Windows displays a small “success” or “failure” toast to the user (if enabled in policy).

Process Flow Diagram Here:



**Registries with Respect to LOB and Win32 Apps**

* **Win32 apps** managed by IME write their policy payloads under:

HKLM\SOFTWARE\Microsoft\IntuneManagementExtension\Policies\{App-GUID}

* Detection state lives here:

HKLM\SOFTWARE\Microsoft\IntuneManagementExtension\DetectionState\{App-GUID}

* **LOB apps** (stand-alone MSIs or line-of-business packages) use the classic MDM registry keys:

HKLM\SOFTWARE\Microsoft\EnterpriseManagement

**Specific Registries with Application GUID Status**

Every app you deploy gets a unique GUID:

* Under **Policies**, you’ll find the installation instructions.
* Under **DetectionState**, the value indicates install status:
  + 0 = succeeded
  + 1 = pending
  + 2 = failed

You can script checks against these keys to automate health-monitoring tasks.

**Log File Locations**

* **Intune Management Extension (Win32 & scripts):** C:\ProgramData\Microsoft\IntuneManagementExtension\Logs\IntuneManagementExtension.log
* **MDM Agent (policy delivery):**  
   C:\ProgramData\Microsoft\Intune\Logs
* **Company Portal (user actions):** %LOCALAPPDATA%\Packages\Microsoft.CompanyPortal\_8wekyb3d8bbwe\LocalState\Logs

**Company Portal**

The Company Portal is the user-facing Windows/UWP app where enrolled users can:

* Browse and install “Available” apps.
* Check device compliance status.
* Trigger a manual **Sync** to pull down new policies or apps.
* View their device’s enrollment and inventory details.

**How to Sync Once App Assignments Are Done**

* **Server-side:** In the Intune admin center, select a device → **Sync**.
* **Client-side:** User opens Company Portal → clicks the ellipsis (…)> **Sync**.

**Breakdown of Events in Log Files**

* **EnrollmentService.log:** Azure AD join and MDM enrollment events.
* **IntuneManagementExtension.log:** Poll cycles, download/install attempts, and detection checks for Win32 and script deployments.
* **MDM CSP logs:** Delivery and processing of built-in configuration profiles.
* **CompanyPortal.log:** User-initiated syncs, app launches, and error codes.