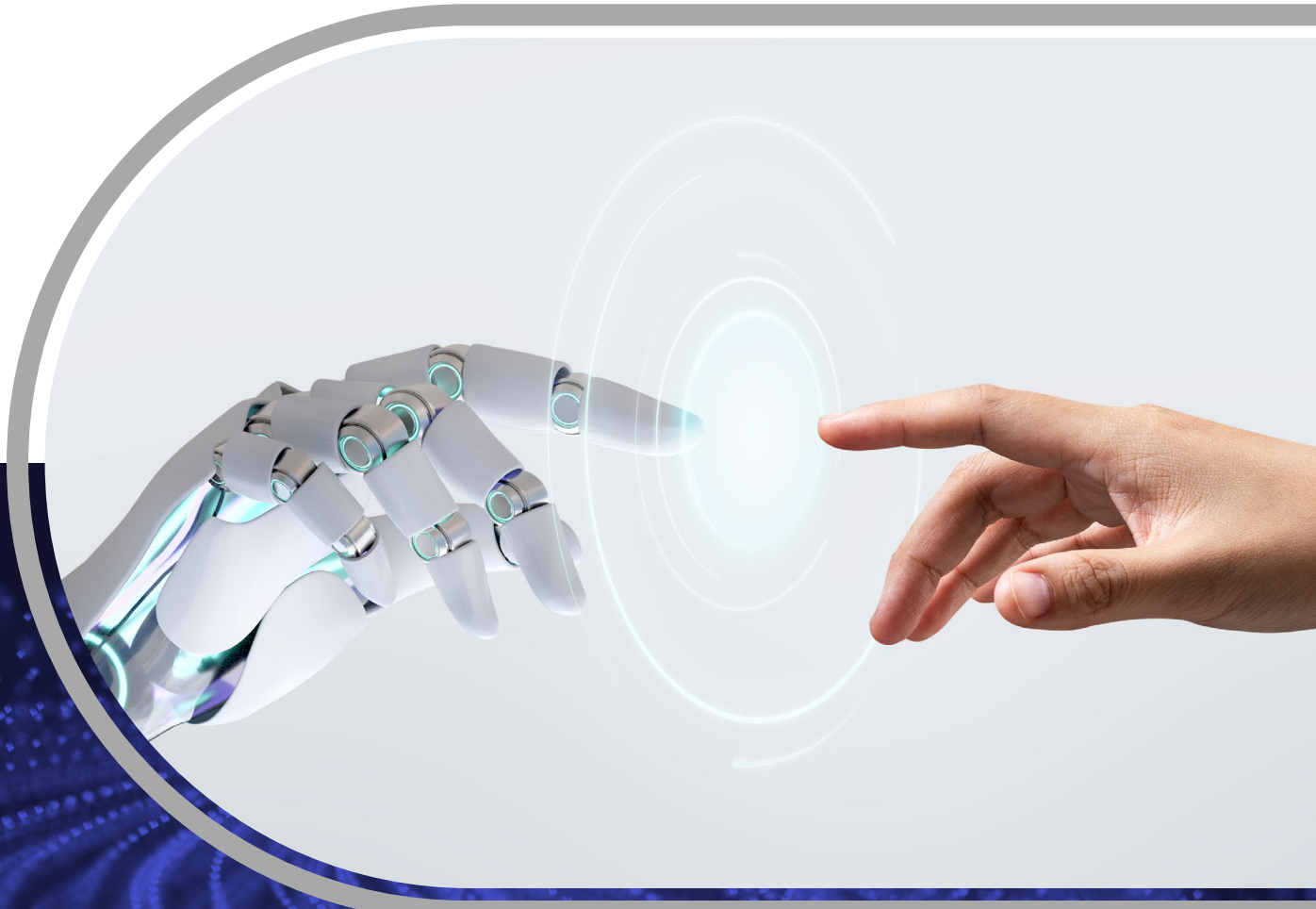
INNOVATIVE INTEGRATION, INC.
IT Solutions at Work



A PRACTICAL GUIDE TO MODERN AI METHODS

V 1.0

Contents

Introduction

This is a **practical, enterprise oriented explanation of the major AI methods in use today**, written for business and IT leaders. **It explains what each method is, what it is best used for**, and **how Microsoft Copilot capabilities (including Notebook, Excel, agent creation, and MCP)** fit together into a coherent strategy.

If you are searching for ROI from your AI then I hope to demonstrate that new techniques which give AI an “Identity” will allow you to deploy Identity based control policies that make deploying powerful/ Autonomous AI practical. This document will show you how your organization is just the right size to purse ROI from AI in a managed framework

Executive summary for CFOs and CIOs at midmarket firms ($10M – $250M in revenue)

AI is moving from "answers" to "action." For midmarket companies, the next competitive advantage won't come from another chatbot—it will come from **Production Agents:** fully autonomous AI services that run continuously, respond to real business events, and execute workflows across systems (think "always-on digital operators," like Crawbot-style automation).

This paper is written for business and technology leaders evaluating real deployment—not experimentation. It outlines why production agents are now feasible in the midmarket, what it takes to govern them, and how to capture measurable ROI. It is also a practical blueprint for engaging our team to implement **Microsoft AI Identity Controls** so your agents can operate safely in production.

The differentiator is **identity and access control for agents:** how they authenticate, what permissions they receive, how approvals are enforced, and how every action is logged. With Microsoft identity governance and privileged access controls, production agents can be run with least privilege, separation of duties, conditional access, and auditable workflows—so autonomy is constrained to business intent. That makes outcomes safe enough for production and governed enough for finance, enabling defensible ROI.

Where midmarket ROI typically shows up first: (1) faster order-to-cash (quote, order entry, billing, collections, dispute resolution), (2) automated reporting, reconciliation, and exception handling in finance, (3) procurement and vendor operations (PO matching, approvals, price/contract compliance), and (4) customer operations execution (case routing, status updates, renewals, and follow-ups) with consistent audit trails.

Recommended path: Select 1–2 high-value workflows tied to CFO-level metrics (cycle time, working capital, margin leakage, close speed), then deploy them as production agents with Microsoft identity controls from day one. From there, standardize an "agent operating model" (identity, approvals, monitoring, and change control) so you can scale automation across functions without scaling risk.

CFOs and CIOs should assume autonomous agents will enter the business—through packaged software, copilots, vendors, and internal teams. The practical challenge is maintaining visibility and control: which agents exist, what data they can access, which actions they can execute, and how to enforce approvals and auditability before autonomy becomes financial, operational, or compliance risk.

Most enterprise AI agents fall into three categories: **agentic chatbots, local agents**, and **production agents**. Each delivers different business value—and introduces different governance and financial risk profiles.

Risk and ROI are driven by access and autonomy

Not all agents carry the same level of business risk—or deliver the same level of economic benefit. The practical risk model comes down to two factors: access and autonomy. Access is the systems, data, and transactions an agent can touch (ERP, CRM, banking portals, billing, procurement, HRIS, cloud services, APIs). Autonomy is how independently the agent can execute steps without human approval.

Agents with limited access and strong oversight usually have limited downside. But as access expands and autonomy increases, the potential impact grows quickly resulting in incorrect postings, unauthorized commitments, data leakage, or process breakage at scale. An agent that only reads internal documentation poses low risk.

An agent that can initiate payments, create vendors, change pricing, approve purchases, modify customer terms, or orchestrate workflows across multiple systems requires a production-grade control environment, not just good prompts.

For CFOs and CIOs, this creates a clear prioritization model: the greater the access and autonomy, the stronger the identity controls, approvals, monitoring, and audit requirements must be.

Deploy production agents at scale without weakening your control environment

Production agents create and use identities programmatically and interact with systems continuously, often outside normal user patterns. Without dedicated controls, they can outpace traditional joiner/mover/leaver processes, shared secrets management, and manual approval chains. Microsoft AI Identity Controls provide the foundation to govern these agent identities with least privilege, conditional access, privileged access workflows, and end-to-end auditing.

Back to the Beginning, Agentic chatbots: the entry point

Your organization is most familiar with agentic chatbots. These assistants operate inside managed platforms (productivity suites, knowledge systems, CRMs, contact-center tools). They are typically triggered by a user and help retrieve information, summarize documents, and complete bounded tasks through approved integrations.
Midmarket firms use chatbots to accelerate sales and customer conversations, improve knowledge retrieval for finance and operations teams, and reduce time spent searching and summarizing. From a governance perspective, this category can be relatively low risk when access is limited and interactions are user-initiated.

Their autonomy is limited and most actions begin with a user prompt. However, they can still introduce meaningful governance issues that are often overlooked.

Many chatbot tools rely on embedded connectors, delegated permissions, or static credentials to access business systems. If permissions are overly broad or not aligned to job roles and segregation-of-duties requirements, the chatbot effectively becomes a high-privilege pathway into sensitive data and transactions.

Similarly, connected knowledge bases can expose confidential data through natural-language queries unless data classification, access boundaries, and logging are enforced.

Chatbot agents may be the lowest-risk category, but they still require disciplined identity governance, connector permissions management, and auditability which Microsoft AI Identity Controls projects are designed to implement.

Local agents: rapid adoption, inconsistent controls

The second category, local agents, is rapidly becoming the most widespread—and the least consistently governed. Local agents run on employee endpoints and integrate with tools like browsers, spreadsheets, terminals, and development environments to automate work.

They can automate tasks such as drafting customer communications, extracting data from invoices and statements, reconciling spreadsheets, querying data sources, creating journal-entry drafts, preparing variance explanations, or orchestrating multi-step workflows across SaaS tools.

What makes local agents distinct is their identity model. Instead of operating under a dedicated, managed system identity, they often inherit the permissions and network access of the user running them. This means the agent can interact with enterprise systems exactly as that employee can except faster and at larger scale.

This design accelerates adoption. Employees can connect agents to tools and data sources without centralized provisioning. But it can also bypass intended controls resulting in inconsistent approval workflows, unclear logging, unmanaged connectors, and hard-to-audit data movement.

CIOs lose centralized visibility into what these agents can access, which systems they interact with, and what actions they can execute. CFOs lose confidence in auditability when automations run "under a user" with unclear evidence, controls, and change management.

Local agents can also introduce supply-chain and data-handling risk. Many rely on third-party plugins and tools from public ecosystems. If an integration is compromised or behaves unexpectedly, it can inherit the user's permissions and move data or trigger actions that are difficult to trace.

For midmarket CFOs and CIOs, local agents are often the biggest near-term governance gap: adoption is fast, controls vary by user, and the business impact can be disproportionate to the size of the automation.

Production agents: fully autonomous AI infrastructure

The third category, production agents, is the most powerful and the most consequential. These agents run as managed enterprise services, built using orchestration platforms or custom code, and are designed to execute workflows end-to-end.

Unlike chatbots or local assistants, they can operate continuously without human interaction, respond to system events, and orchestrate complex workflows across multiple systems.

Midmarket firms can deploy production agents for order management and billing operations, collections and dispute workflows, procurement approvals and three-way match exception handling, customer onboarding, renewal execution, pricing and contract operations, and cross-system reporting/close activities.

Because these agents run as services, they rely on dedicated machine identities (service principals, managed identities, API permissions) to access ERP/CRM/SaaS platforms and cloud resources. This creates a new identity surface that must be governed with the same rigor as privileged human access, often more, because the agent can act continuously.

The biggest risks arise from autonomous agents are:

First, these agents often operate with high autonomy, executing actions without human review so approvals, limits, and exception handling must be engineered into the workflow.
Second, they frequently process untrusted external inputs (customer requests, emailed documents, web forms, webhook data). If not constrained, this can cause the agent to take actions that violate policy creating data integrity issues, unauthorized commitments, or downstream control failures.
Third, complex multi-agent architectures can create unclear handoffs and approval boundaries as agents trigger other agents across systems. Without explicit identity scoping and segregation-of-duties controls, accountability blurs and high-impact actions can occur without the right evidence or approvals.

AI agents create an identity and controls challenge

Across all three categories, one reality is clear. AI agents are a new set of first-class identities operating inside enterprise environments. They access data, trigger workflows, interact with infrastructure, and make decisions using identities and permissions.

When those identities are poorly governed and access is over-permissioned, agents can execute transactions or move data in ways that are hard to reconcile—creating fraud exposure, material errors at scale, compliance issues, and costly remediation.

For CFOs and CIOs, the goal is not to "slow down AI," but to operationalize it with control: visibility into what agents exist, governance over the identities they use, and proof that access and approvals match business intent.

1. Which agents exist (owned by whom, in which environment)
2. Which identities and credentials they use (human, service, managed identity)
3. Which systems, data, and transactions they can access
4. What approvals, constraints, and monitoring are enforced for high-impact actions
5. Whether permissions and behavior align to documented business purpose and segregation-of-duties requirements

Midmarket companies have spent years securing human users and core service accounts. Agents are the next wave of identities—and they are arriving quickly through vendors, copilots, and internal automation.

Organizations that win with agents will not be the ones that avoid autonomy. They will be the ones that **govern agent identity, enforce least privilege and approvals**, and **prove auditability** so automation can scale without creating uncontrolled financial or operational risk.

Next step: If you are planning to deploy production agents in finance, operations, or customer workflows, engage our team to implement **Microsoft AI Identity Controls** as the control layer covering agent identity design, privileged access, approval patterns, monitoring, and audit-ready evidence. At that point your autonomous workflows will deliver measurable ROI with a defensible control environment.

From Copilot Chat to Agentic Automation with MCP

1. Foundational Method: Generative AI (Prompt Based)

What it is
This is the base capability most people encounter first: large language models generating text, summaries, explanations, or ideas from a prompt.

Strengths
- Fast to use
- No setup required
- Excellent for drafting, summarization, brainstorming

Limitations
- Relies on model training data
- Cannot "know" your organization unless grounded
- Prone to hallucination when asked factual or policy specific questions

Where this shows up
- Microsoft Copilot Chat (without document grounding)
- Ad hoc writing in Word, Outlook, Teams

Best use
Idea generation and language tasks where precision is not critical

---

2. Grounded AI with RAG (Retrieval Augmented Generation)

What RAG is
Retrieval Augmented Generation (RAG) combines two steps:
1. Retrieve relevant documents from trusted data sources
2. Use the model to generate an answer grounded in those documents

This dramatically improves accuracy, trust, and explainability by anchoring responses in real enterprise content rather than model memory alone. [learn.microsoft.com], [labs.zenity.io]

Copilot Notebook: "RAG for Business Users"

Copilot Notebook is Microsoft's most accessible RAG experience. It allows users to:

- Attach multiple documents
- Maintain long running context
- Ask increasingly deep questions over the same material
- Control response style and structure

This effectively gives each user a **lightweight, per task RAG system** without any setup or engineering work. [zenn.dev]

Best use of Copilot Notebook

- Policy interpretation
- Research synthesis
- Project specific analysis
- "One job, one RAG" knowledge work

Notebook is best when accuracy matters and the data set is bounded.

---

3. Embedded AI in Productivity Tools (Copilot for Work)

What this method is

This is AI embedded directly inside Microsoft 365 applications, operating on **live working data** while respecting permissions.

Key examples

Copilot in Excel

Copilot in Excel can:

- Analyze tables
- Generate formulas
- Build charts and PivotTables
- Perform multi step data transformations
- Explain trends and outliers

Importantly, it operates using **native Excel features**, meaning outputs remain editable and auditable. [support.mi...rosoft.com], [computerworld.com]

Best use

Data analysis, reconciliation, and insight generation directly where the data lives

---

Copilot for Work (Word, Outlook, Teams, PowerPoint)

This class of Copilot excels at:

- Drafting and summarizing
- Meeting recap and follow ups
- Presentation creation
- Email and communication acceleration

Best use: **Human productivity amplification, not process automation**

4. Custom AI Agents (Copilot Agent Creation)

What an agent is

An **agent** is more than a chatbot. It can:

- Understand intent
- Retrieve knowledge
- Take actions
- Execute multi step workflows

Microsoft enables agent creation through:

- **Agent Builder in Microsoft 365 Copilot**
- **Copilot Studio (low code/no code)**

Business users can create agents using natural language and guided interfaces, without needing software developers. [learn.microsoft.com], [support.mi...rosoft.com]

Types of agents

- Informational (Q&A over knowledge)
- Task oriented (generate reports, create artifacts)
- Process aware (trigger workflows, retrieve data)

Best use

Reusable assistants for repeatable business scenarios

---

5. Agentic Automation with MCP (Model Context Protocol)

What MCP is

Model Context Protocol (MCP) is an open standard that defines how AI agents discover, access, and use external tools and data through a consistent interface.

It separates:

- **Reasoning** (the agent)
- **Execution** (tools and systems)

This avoids hard coding integrations and reduces long term technical debt.
[learn.microsoft.com], [aihandbook.io]

Why MCP matters

Before MCP:

- Each agent required custom integrations
- Tool logic was tightly coupled
- Scaling agents created exponential complexity

With MCP:

- Tools are exposed once
- Any compliant agent can use them
- Agents remain portable and governable

MCP is now supported across major AI platforms and is positioned as the standard integration layer for autonomous AI workflows. [clarion.ai], [en.wikipedia.org]

Best use of MCP

Assembling autonomous, multi step AI processes that span systems

Examples:

- Cross system reconciliation
- Automated business oversight
- Agent to agent coordination
- Human in the loop workflows

Putting It All Together: How to Use These Methods Strategically

A simple mental model

NEED	BEST METHOD
Draft, summarize, brainstorm	Copilot Chat
Accurate answers from documents	Copilot Notebook (RAG)
Analyze spreadsheets	Copilot in Excel
Reusable assistants	Copilot Agents
Cross system automation	MCP based agentic workflows

Recommended Adoption Path

1. **Start with Copilot for Work**
   Build familiarity and trust.
2. **Use Copilot Notebook for high value knowledge work**
   This is the fastest path to RAG benefits.
3. **Introduce agents for repeatable scenarios**
   Move from ad hoc help to structured assistance.
4. **Use MCP to scale autonomy safely**
   When outcomes, not prompts, matter.

Final Takeaway

- **Copilot Chat** helps you think
- **RAG (Notebook)** helps you be right
- **Copilot in tools** helps you work faster
- **Agents** help you reuse intelligence
- **MCP** helps you automate outcomes

Mapping AI Methods to Security & Governance Implications

Below is a **clear, governance first mapping of each AI method to its security and control implications**, written so it can be used by **IT leadership, security, audit, and architecture teams.**

This is **not marketing** and **not tool specific implementation guidance.** It's a **control-lens view:** *what risk each method introduces, what controls are required, and who should own it.*

---

1. Prompt Based Generative AI (Ungrounded Chat)

Examples

- Copilot Chat (no document grounding)
- Free form LLM prompts
- Brainstorming and drafting

Security Characteristics

- No guaranteed grounding in enterprise data
- High reliance on user judgment
- Minimal determinism

Primary Risks

- Hallucination presented as fact
- Accidental disclosure via prompts
- Over trust by users
- Low auditability of outcomes

Governance Implications

- ✓ Lowest technical risk
- ⚠ Highest human error risk

Required Controls

- User training and acceptable use policy
- Enterprise data protection (EDP)
- Prompt logging (where available)
- Explicit "not authoritative" guidance

Best Ownership

End user productivity + security awareness

Governance posture:
Allowed broadly, trusted minimally

2. RAG (Retrieval Augmented Generation)

Includes

- Copilot Notebook
- SharePoint-grounded Copilot
- Knowledge based Copilot agents

Security Characteristics

- Answers grounded in enterprise content
- Inherits source permissions
- Deterministic retrieval + probabilistic synthesis

Primary Risks

- Incorrect interpretation of retrieved content
- Stale or incomplete knowledge sources
- Overconfidence due to grounded appearance

Governance Implications

- ✓ Much higher trustworthiness than chat
- ✓ Strong alignment with data governance
- ⚠ Risk shifts from model error » content quality

Required Controls

- Source curation and lifecycle management
- Permissions enforcement (Graph / SharePoint ACLs)
- Content ownership and review processes
- Data classification alignment

Best Ownership

Information governance + business data owners

Governance posture:
Trustworthy if data is trustworthy

---

3. Copilot Embedded in Productivity Tools (Excel, Word, Outlook, Teams)

Includes

- Copilot in Excel
- Copilot in Word / PowerPoint
- Meeting summaries, email drafting

Security Characteristics

- Operates on live, permission scoped data
- Uses native application features
- Changes are visible and reversible

Primary Risks

- Unintended data manipulation
- Over automation of judgment based tasks
- Misinterpretation of analytical output

Governance Implications

- ✓ Strong auditability (outputs exist as files)
- ✓ Low credential risk
- ⚠ Requires clear accountability for outputs

Required Controls

- Standard M365 access controls
- Versioning and document history
- Role clarity (who approves outputs)
- Training on "AI assisted ≠ AI approved"

Best Ownership

Business teams with IT guardrails

Governance posture:
Safe when outputs remain human owned

---

4. Copilot Agent Creation (Non Autonomous Agents)

Includes

- Copilot Agents built in Copilot Studio
- Task oriented or informational agents
- Human invoked execution

Security Characteristics

- Persistent behavior
- Defined scope and instructions
- Limited autonomy

Primary Risks

- Scope creep
- Poorly defined instructions
- Inconsistent behavior across updates

Governance Implications

- ✓ More control than ad hoc usage
- ✓ Reusable and inspectable logic
- ⚠ Requires lifecycle governance

Required Controls

- Agent ownership and purpose definition
- Change management for instructions
- Logging of interactions
- Retirement and versioning processes

Best Ownership

IT + business process owners

Governance posture:
Govern like an application, not a user

5. Agentic Automation (Autonomous or Semi Autonomous Agents)

Includes

- Multi step task execution
- Cross system workflows
- Agents that act without constant human input

Security Characteristics

- Actions, not just advice
- Persistent state
- Decision making under uncertainty

Primary Risks

- Unauthorized actions
- Runaway workflows
- Accountability ambiguity
- Tool misuse

Governance Implications

- ⚠ Highest impact if unmanaged
- ✓ Safest when identity governed
- ✓ Requires Zero Trust principles

Required Controls

- Explicit identity (non human identity)
- Least privilege enforcement
- Conditional access
- Kill switch and revocation
- Full audit trail

Best Ownership

Security, IAM, and platform architecture

Governance posture:
Treat as a digital worker

---

6. MCP (Model Context Protocol) as an Automation Assembly Layer

What MCP Changes

- Separates reasoning from execution
- Standardizes tool access
- Enables safe composability

Security Characteristics

- Explicit tool contracts
- Centralized execution boundaries
- Reduced integration sprawl

Primary Risks

- Tool over exposure if poorly designed
- Weak MCP server security
- Improper permission mapping

Governance Implications

- ✓ Enables scalable governance
- ✓ Reduces “shadow automation”
- ✓ Aligns with Zero Trust architecture

Required Controls

- Tool level permissioning
- MCP server authentication
- Change control for tool schemas
- Execution approval patterns (where required)

Best Ownership

Platform engineering + security architecture

Governance posture:
The safest way to scale autonomy

Summary Table: Security & Governance View

METHOD	RISK LEVEL	GOVERNANCE FOCUS	WHO OWNS IT
Prompt based chat	Low technical / high human	Usage policy	End users
RAG / Notebook	Medium	Data quality & access	Data owners
Copilot in tools	Low	Output accountability	Business teams
Copilot agents	Medium	Lifecycle management	IT + business
Agentic automation	High	Identity & access	Security
MCP based processes	Controlled high	Platform governance	Architecture

The Key Governance Insight

The more an AI system can act, the more it must be governed like a person or application, not a feature.

- Chat needs education
- RAG needs data governance
- Copilot needs accountability
- Agents need lifecycle management
- Autonomous agents need **identity, policy, and audit**
- MCP is how you make that scalable without chaos

Below is an **enhanced, security forward narrative** you can use with **CISOs, IT leadership, auditors, and risk committees** when a firm is operating with **Microsoft Agent 365, Microsoft Intune, and Microsoft Entra Conditional Access** together.

This reframes the conversation from *"Can AI be secured?" to "How AI becomes governable infrastructure."*

Enhancing the Security Model with Agent 365 + Intune + Conditional Access

When Microsoft **Agent 365, Intune,** and **Entra Conditional Access** are used together, AI agents can be governed with **stronger controls than most human users.**

This stack enables:

- Identity centric governance of AI agents
- Device and environment aware access decisions
- Centralized visibility, audit, and kill switch capability
- Alignment with Zero Trust and NIST guidance

Microsoft explicitly positions AI agents as a **new security primitive**, not just applications, and Agent 365 is designed to be the control plane for that layer. [siliconangle.com], [techzine.eu]

---

1. Agent 365: Making AI a First Class Security Subject

What Changes with Agent 365

Agent 365 introduces **Agent Identities** in Microsoft Entra, allowing AI agents to be treated as:

- Non human identities
- Lifecycle managed security principals
- Governed assets with ownership, scope, and auditability

This addresses a fundamental gap in earlier automation models, where agents operated with shared credentials or opaque execution contexts. [learn.microsoft.com], [dellenny.com]

Security Impact

- ✓ Eliminates anonymous or shared bot accounts
- ✓ Enables least privilege access for agents
- ✓ Central registry prevents "shadow AI"

Security shift: AI moves from "tool risk" to "identity risk"—which IAM is designed to manage.

2. Conditional Access: Policy Enforcement for AI Agents

Conditional Access for Agent Identities

Microsoft Entra extends **Conditional Access (CA)** to agent identities, applying Zero Trust evaluation at token issuance for AI agents. [learn.microsoft.com], [learn.microsoft.com]

Supported controls include:

- Explicit allow/block
- Risk based blocking for high risk agents
- Scoped access to resources

While agents do not use interactive MFA or device posture like humans, CA still functions as a **central policy decision point** for agent execution.

Security Impact

- ✓ Immediate block capability for compromised agents
- ✓ Risk adaptive enforcement using Entra risk signals
- ✓ Centralized policy rather than per tool controls

Key distinction:
CA for agents enforces *authorization*, not *authentication ceremony*.

---

3. Intune: Device and Execution Environment Trust

Why Intune Still Matters in an Agent Driven World

Intune governs **the devices and environments through which humans interact with agents**, and—critically—where agent outputs are consumed or approved.

Intune + Conditional Access enables:

- Device compliance enforcement
- Encryption, OS health, AV posture validation
- Blocking access from unmanaged or risky endpoints. [learn.microsoft.com], [oneuptime.com]

Security Impact

- ✓ Prevents agent interaction from compromised endpoints
- ✓ Reduces data leakage via unmanaged devices
- ✓ Ensures human in the loop actions occur on trusted hardware

Intune secures the human boundary of AI systems.

---

4. The Combined Control Plane (Why This Is Stronger Than Traditional Security)

Traditional Model (Pre Agent 365)

- Users authenticated
- Devices optionally trusted
- Automation largely invisible
- Bots often bypassed IAM

Modern Model (Agent 365 + Intune + CA)

Identity (Human + Agent)

⬇

Conditional Access (Policy Decision)

⬇

Execution (Agent or User)

⬇

Device Trust (Intune)

⬇

Audit / Defender / Purview

This creates **continuous verification** across:

- Who is acting (human or agent)
- Under what policy
- From what environment
- With what permissions
- With full traceability

Microsoft explicitly frames this as **security woven into every layer of the AI estate**, not bolted on afterward. [microsoft.com]

5. Risk Reduction Compared to Humans and Legacy Automation

RISK AREA	HUMAN USERS	LEGACY RPA	AGENT 365 STACK
Credential sprawl	High	High	Low
Least privilege	Inconsistent	Rare	Enforced
Visibility	Partial	Tool local	Centralized
Revocation	Slow	Manual	Immediate
Auditability	Variable	Fragmented	Unified

Counter intuitive truth:
Properly governed AI agents are often *safer than humans*.

6. Alignment with Zero Trust and NIST

This combined model aligns with:

- **Zero Trust Architecture (NIST SP 800 207)**
- **Identity centric security models**
- **Continuous monitoring and risk adaptive access**

Key principles satisfied:

- Never trust, always verify
- Least privilege by default
- Assume breach
- Continuous evaluation

Microsoft explicitly maps Agent 365 and Entra Agent ID to these principles in its 2026 security guidance. [microsoft.com], [microsoft.com]

7. What CISOs Care About

- Visibility
- Control
- Accountability
- Kill switches
- Audit trails

"We are not giving AI more freedom. We are bringing AI under the same identity, device, and policy controls as the rest of the enterprise."

Bottom Line

When **Agent 365, Intune**, and **Conditional Access** are deployed together:

- AI agents become **governable digital workers**
- Security posture **improves**, not degrades
- Zero Trust extends naturally to autonomous systems
- The organization gains visibility it never had with human or RPA execution

This is not experimental security—it is **IAM done correctly for the AI era**.

Agentic AI (like Crawbot, Perplexity Computer, Nvidia NeMo) under Microsoft 365, Intune, and Conditional Access

Mid Market Enablement: Freedom with Control

Organizations in the **$10M–$250M** range typically face the same tension:

- High operational complexity
- Limited engineering staff
- Heavy reliance on Microsoft 365
- Strong security expectations, but no appetite for building platforms

Historically, this meant choosing between:

- **Speed without governance** (shadow automation, scripts, RPA sprawl), or
- **Governance without speed** (custom development they can't staff)

Crawbot style AI, operating under Microsoft Agent 365 with MCP, changes that tradeoff.

What "Freedom" Actually Means for the Mid Market

1. Freedom from Custom Application Development

Your assumption is valid. An MCP based agent can operate without requiring a mature application development department.

Microsoft explicitly designed MCP and Copilot Studio so that:

- Business and IT teams can **consume MCP servers**
- Tools are discovered dynamically
- Integration logic is externalized from the agent
- No custom APIs or point to point connectors are required

[devblogs.m...rosoft.com], [learn.microsoft.com]

Key implication:
The mid market does *not* need to build applications—only to **authorize tools**.

2. Freedom from the N×M Integration Problem

Before MCP:

- Each automation required a custom connector
- Each system integration added fragility
- Maintenance scaled exponentially

With MCP:

- Easily connects to standardized MCP servers
- Tools are reusable across agents
- Microsoft treats MCP as a **first class integration pattern** [bridgeapp.ai]

This directly benefits the mid market, where integration maintenance is a silent cost multiplier.

---

3. Freedom Without Losing Security Control

MCP systems that run locally do **not** run outside your security perimeter.

When deployed correctly:

- Crawbot has an **Entra Agent ID**
- Access is governed by **Conditional Access**
- Human interaction occurs only on **Intune compliant devices**
- Agent behavior is visible via **Agent 365 registry and logs** [microsoft.com], [computerworld.com]

This means:

- No shared credentials
- No shadow bots
- No "one off" scripts bypassing policy

For the mid market, this is rare:
Automation that reduces security risk instead of increasing it.

Crawbot as MCP: Why This Works Without a Dev Team

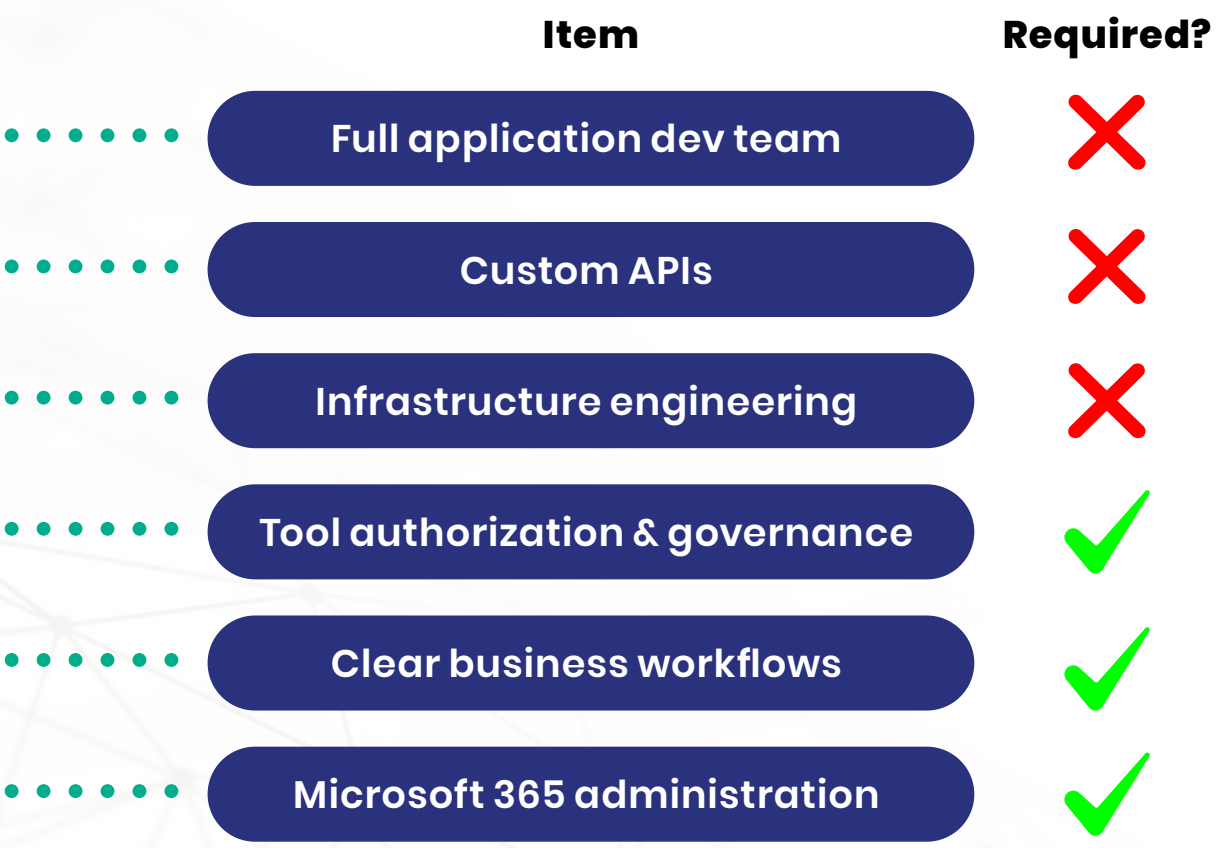
What Crawbot Is (In This Model)

- Crawbot is **not a custom app**
- Crawbot is **not a hard coded integration**
- Crawbot is an **agent that reasons**
- MCP servers handle execution and data access

Microsoft's own MCP guidance emphasizes:

- Declarative agent configuration
- Tool discovery via MCP manifests
- Enterprise security inheritance (auth, DLP, network controls) [devblogs.m...rosoft.com], [microsoft.github.io]

What the Mid Market Actually Builds



This is exactly the skill set the mid market *already* has.

Security Model: Why This Is Safer Than "Traditional Automation"

Local MCPs under Microsoft Agent365 + Intune + Conditional Access

CONTROL LAYER	WHAT IT DOES
Agent 365	Central agent registry, lifecycle, audit
Entra Agent ID	Identity, least privilege
Conditional Access	Risk based allow/block
Intune	Device trust for human in the loop
Purview / Defender	Data & threat protection

Microsoft explicitly frames this as identity governed AI, not application sprawl. [nexustek.com]

Mid Market Value Proposition (Plain English)

Local MCP agents let a mid market firm behave like a large enterprise—without building enterprise software.

You gain:

- Autonomous execution
- Cross system visibility
- Faster operations
- Centralized security
- No engineering tax

And you avoid:

- RPA fragility
- Script sprawl
- Shadow AI
- Custom app maintenance

Validating the Assumption (Explicitly)

✓ **Validated:**
Crawbot can function as an **MCP based agent** that the mid market can deploy **without a mature application development department**, provided:

1. Microsoft 365 is the core platform
2. Agent 365 governs identity and lifecyclemanaged identity)
3. MCP servers are consumed—not built from scratch
4. Intune and Conditional Access are enforced

This is **explicitly supported** by Microsoft's MCP and Agent 365 architecture and documentation. [devblogs.m...rosoft.com], [learn.microsoft.com]

The Mid Market Bottom Line

For the first time, the mid market can:

- Use **agentic automation**
- Maintain **enterprise grade security**
- Avoid **enterprise grade complexity**

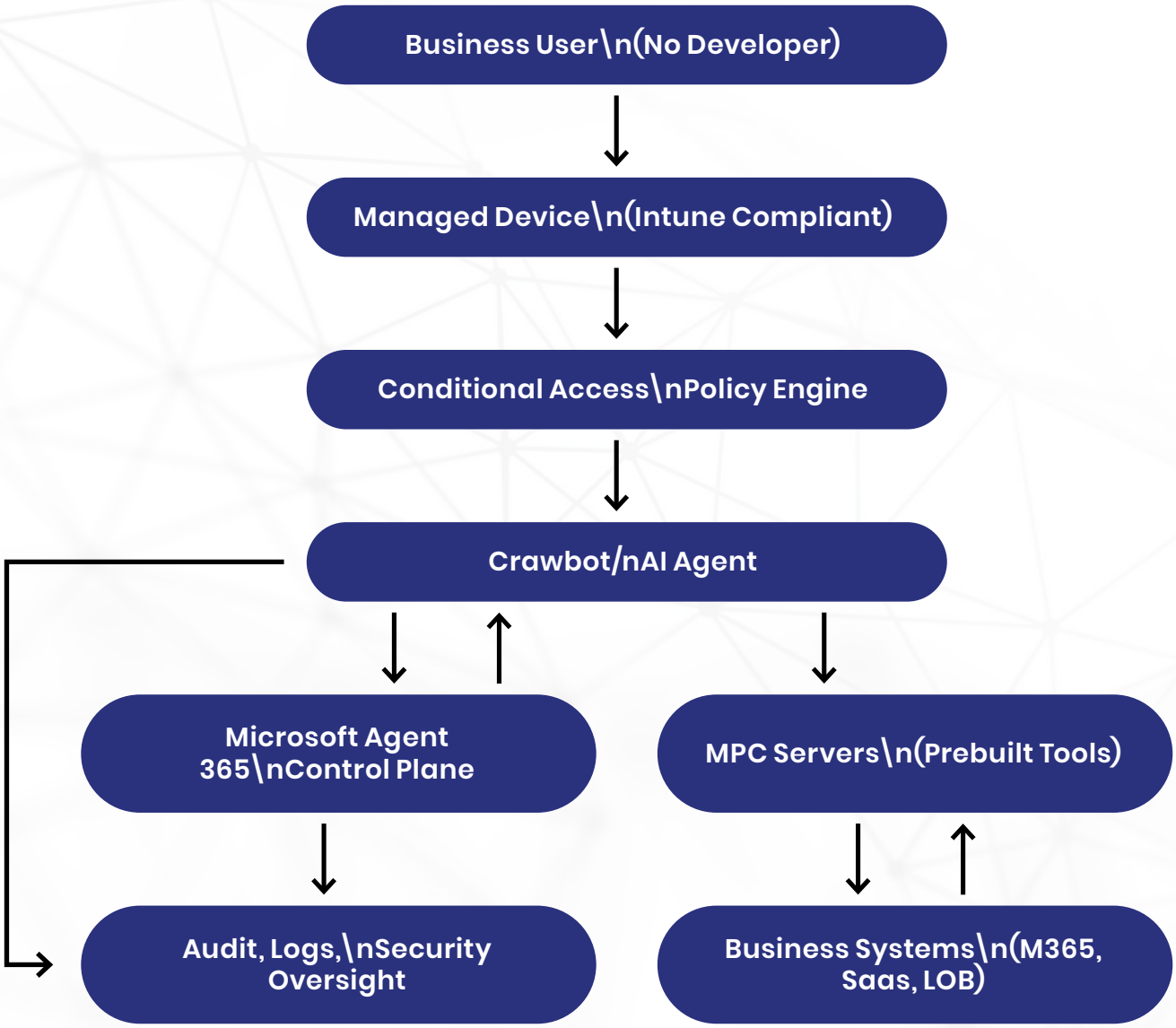
A MCP driven agent under Microsoft 365, is not "AI experimentation." It is **controlled freedom**.

Why This Works Without Developers diagram

1. "Why This Works Without Developers" — Architecture Diagram

Core Idea

Reasoning stays with the agent. Execution stays with MCP tools. Governance stays with Microsoft 365 and Agent 365.



2. How to Read The Diagram (Non Technical Explanation)

Business User 

- No coding
- No scripts
- No APIs
- Interacts through Microsoft 365 (Teams, Copilot, etc.)

✓ **No developer dependency**

Managed Device (Intune) 

- Ensures access only from compliant endpoints
- Protects the *human boundary* of AI

✓ **Security without engineering**

Conditional Access 

- Decides *if* Crawbot can act
- Enforces Zero Trust (risk, identity, policy)

✓ **Policy replaces custom logic**

Microsoft Agent 365 

- Registers Crawbot as a non human identity
- Applies lifecycle, access, visibility, kill switch
- Central place IT already knows

✓ **Governance replaces app management**

MCP Servers (Prebuilt Tools) 

- Expose actions like:
  - Read data
  - Update records
  - Trigger workflows
- Standardized, reusable, secured

✓ **Execution without custom development**

Business Systems 

- Microsoft 365
- SaaS tools
- Line of business apps
- Accessed only via MCP permissions

✓ **No direct API coding**

Audit & Oversight 

- Every action is logged
- Every agent is visible
- Security teams retain control

✓ **Enterprise grade accountability**

---

3. This works without developers because it never integrates directly with systems—MCP provides the tools, Microsoft provides the security, and your AI only provides reasoning.

---

4. Why This Is Perfect for the Mid Market

Traditional Model (Why Devs Were Required)

- Custom integrations
- Script maintenance
- Fragile automation
- Security exceptions

Local Agent + MCP + Microsoft 365 Model

- **Declarative configuration**
- **Prebuilt tool interfaces**
- **Identity based governance**
- **No integration sprawl**

✓ Mid market teams configure **what's allowed**, not **how it's coded**

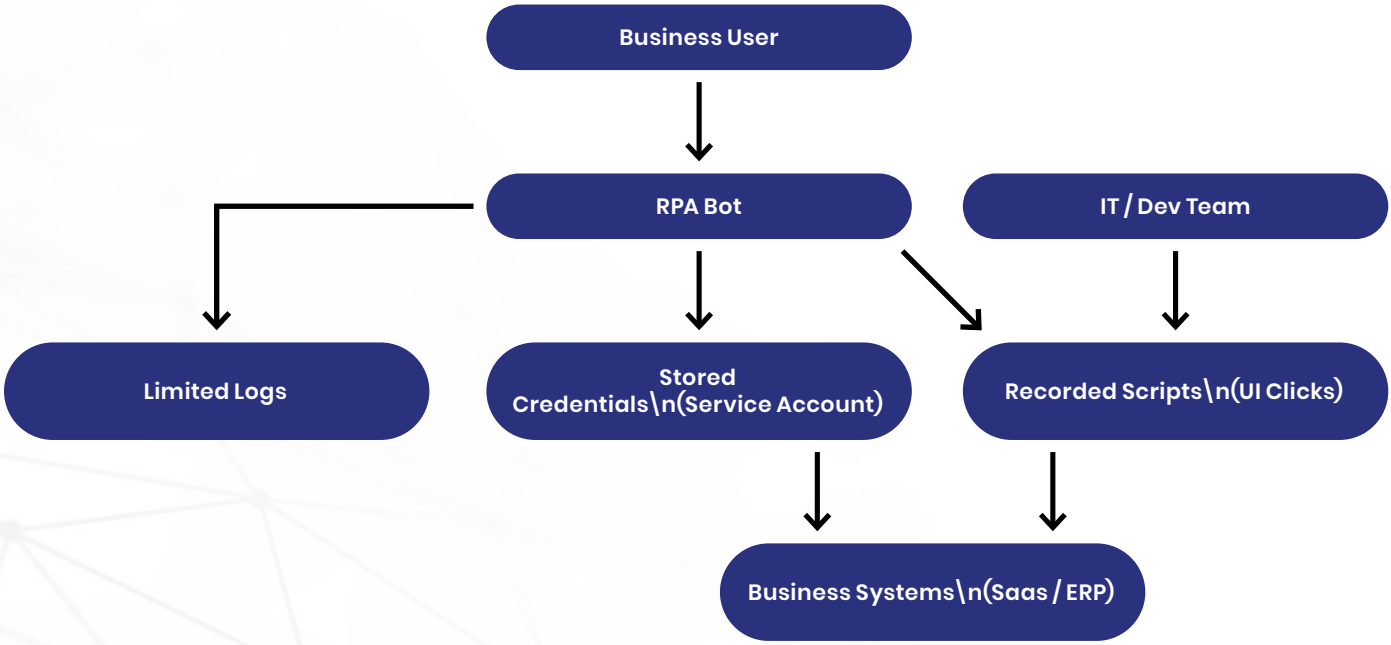
---

5. Tools like "Crawbot" deliver autonomous outcomes while Microsoft 365 enforces identity, device trust, and policy—no application development required."

(MCP based, governed by Microsoft 365).

1. Before / After Diagram — RPA vs Crawbot

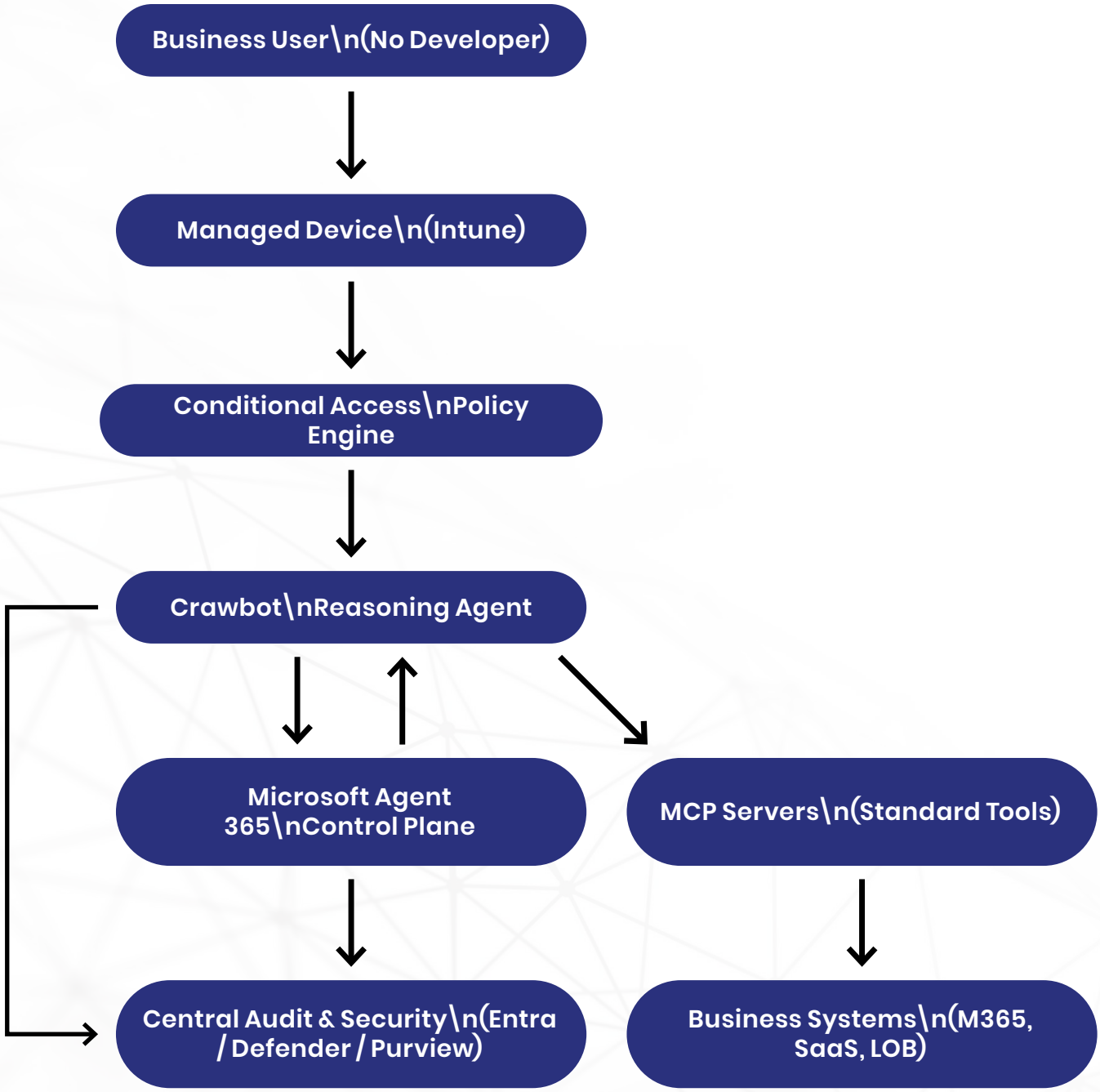
BEFORE: Traditional RPA (Why It Breaks in the Mid Market)



Characteristics

- Script driven
- UI fragile
- Credential heavy
- IT dependent
- Poor visibility

AFTER: Crawbot (MCP + Microsoft 365 Governance)



Characteristics

- Reasoning driven
- Tool based execution
- Identity governed
- No app development
- Full visibility

2. What Changed (Plain English Comparison)

DIMENSION	TRADITIONAL RPA	CRAWBOT (MCP + M365)
How work is done	Replays scripts	Reasons over goals
Integration model	Custom per system	Standard MCP tools
Credentials	Stored passwords	Entra Agent ID
Security	Tool local	Zero Trust (CA + Intune)
Change tolerance	Breaks easily	Adapts dynamically
Who builds it	Developers / IT	Business + IT governance
Auditability	Fragmented	Centralized
Mid market fit	Poor	Excellent

---

3. Why This Matters for the Mid Market

RPA Forced This Question:
"Do we have developers to keep this alive?"

Crawbot Changes the Question to:
"Do we trust our identity, policy, and data controls?"

The mid market already **does** trust:

- Microsoft 365
- Entra ID
- Intune
- Conditional Access

Crawbot simply **operates inside that trust boundary**.

---

4. RPA automates scripts. Crawbot style tools automate outcomes—under identity, policy, and audit control.

---

5. *"Crawbot" style tools removes application development from automation by separating reasoning (agent) from execution (MCP) and governance (Microsoft 365)."*