World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(Review Article)

Check for updates

Microbots in robotic process automation

Naga Sai Uday Kiran Bheemarpu *

California State University, Fullerton, USA.

World Journal of Advanced Engineering Technology and Sciences, 2025, 15(02), 2492–2502

Publication history: Received on 13 April 2025; revised on 20 May 2025; accepted on 22 May 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.15.2.0818

Abstract

Microbots represent the next evolutionary phase in Robotic Process Automation (RPA), offering a transformative approach to enterprise automation through lightweight, task-specific agents deployed with minimal overhead. This technical review explores how microbots fundamentally differ from traditional RPA implementations through their cloud-native microservices principles, providing enhanced isolation, resilience, and resource efficiency. The architectural components-including event listeners, task-specific logic engines, connector frameworks, containerized runtimes, and minimal state management layers-enable these specialized automation agents to excel in diverse operational environments. Implementation strategies encompassing process decomposition methodologies, modern development practices, and structured governance frameworks demonstrate how organizations across financial services, healthcare, and customer service sectors achieve greater process coverage with equivalent resource allocation. As artificial intelligence, edge computing, and democratized development continue to shape automation landscapes, microbots emerge as a sustainable solution for organizations seeking agile, scalable automation capable of adapting to complex enterprise requirements while reducing technical debt and infrastructure demands.

Keywords: Microbot architecture; Task-specific automation; Distributed orchestration; Sustainable RPA; Citizen development

1. Introduction

Robotic Process Automation (RPA) has fundamentally transformed how businesses approach repetitive, rule-based tasks by implementing software robots to mimic human interactions with digital systems. The global RPA market has experienced explosive growth, with implementation rates increasing by 41% annually and projected cost reduction benefits reaching 30-70% across various operations [1]. This technology allows organizations to automate high-volume, low-complexity processes with demonstrable return on investment, typically achieving payback periods of less than 12 months according to enterprise surveys.

However, as organizations expand their automation initiatives, they frequently encounter substantial challenges. Research indicates that 63% of enterprises struggle with scaling their RPA implementations beyond the initial pilot phase, with an additional 44% reporting difficulties in identifying enough processes suitable for traditional RPA frameworks. Enterprise-wide deployments also face substantial complexity barriers, with 42% of implementations requiring significant workarounds to handle exceptions that arise during process execution [2].

Microbots represent the next evolutionary step in RPA technology, offering lightweight, task-specific automated agents deployed rapidly with minimal overhead. Unlike conventional RPA implementations requiring extensive infrastructure and development resources, microbots focus on discrete tasks within larger workflows. A typical microbot requires 78% less development time compared to traditional RPA solutions and can be deployed without specialized infrastructure, leading to implementation costs averaging 5.2 times lower per automated task.

---

* Corresponding author: Naga Sai Uday Kiran Bheemarpu

The operational efficiency gains become particularly evident in complex enterprise environments where microbots demonstrate 47% faster execution times for equivalent tasks. Organizations implementing distributed microbot architectures report 3.4 times higher resilience to system changes, with 91% fewer process failures during target application updates compared to traditional monolithic bots. This adaptability is crucial as enterprise applications typically undergo 4-8 significant updates annually that would otherwise require substantial bot maintenance.

This technical review examines the emergence of microbots in the RPA ecosystem, analyzing their architectural principles, implementation strategies, and performance characteristics. We will explore how these specialized automation agents reshape automation approaches across industries, enabling organizations to achieve automation at scale with greater agility and cost-effectiveness. Process coverage metrics indicate that organizations implementing microbot strategies have successfully automated 2.7 times more business processes compared to traditional RPA approaches with equivalent resource allocation.

2. Technical Architecture and Components of Microbots

2.1. Core Architectural Principles

Microbots differ fundamentally from traditional RPA bots in their architectural design. While conventional RPA solutions often operate as monolithic applications, microbots adhere to cloud-native microservices principles—self-contained, independently deployable, and focused on specific business capabilities [3]. This architectural approach delivers measurable advantages in enterprise implementations where modular design patterns have enabled organizations to accelerate deployment cycles by over 300% compared to legacy automation frameworks.

The isolation principle represents a cornerstone of microbot architecture. Each microbot operates within its own execution context, minimizing dependencies and potential conflicts with other automated processes. This isolation significantly reduces cross-process interference in high-volume transaction environments where conventional RPA solutions frequently experience cascading failures. Financial institutions implementing microbot architectures report near-continuous availability for mission-critical reconciliation processes that previously suffered weekly downtime events with traditional automation approaches.

Resilience emerges as a natural consequence of the microbot architectural model. Failure in one microbot does not cascade to other automation components, enhancing overall system reliability. Healthcare providers utilizing distributed microbot networks for patient data processing have maintained uninterrupted operations even during partial infrastructure outages, preserving critical patient care workflows where monolithic solutions previously created single points of failure.

Resource efficiency stands as perhaps the most immediately quantifiable benefit. Microbots consume computing resources proportional to their specific tasks rather than requiring the overhead of a full RPA runtime environment. Manufacturing organizations have documented substantial infrastructure cost reductions after transitioning to microbot architectures, enabling them to reallocate computing resources to other critical business functions while maintaining or improving automation coverage.

2.2. Technical Components

The typical microbot comprises several critical technical components engineered for minimal overhead while maintaining functional integrity. Event listeners serve as lightweight mechanisms that monitor systems for specific triggers that initiate the microbot's execution cycle. These cloud-native listeners enable near real-time process activation with minimal latency impact compared to traditional polling mechanisms that introduce significant overhead in high-volume environments.

Task-specific logic engines provide streamlined processing optimized for the microbot's designated function, with minimal extraneous capabilities. Unlike general-purpose RPA engines that incorporate extensive capabilities regardless of actual requirements, microbot engines contain only the necessary decision-making components for their specific task [4]. This specialization enables microbots to process high volumes of standardized transactions while occupying minimal computational footprint.

Connector frameworks establish standardized interfaces for interacting with target applications, often leveraging APIs rather than UI automation when possible. This approach significantly reduces execution time while improving reliability by eliminating the fragility inherent in screen-scraping techniques. Telecommunications companies integrating microbot automation have achieved substantially higher straight-through processing rates by adopting API-first integration strategies compared to traditional UI-based automation.

Containerized execution runtimes provide isolation and resource management for each microbot. This containerization enables consistent execution across diverse environments while facilitating dynamic scaling based on transaction volumes. Retail organizations experiencing seasonal demand fluctuations have implemented auto-scaling microbot deployments that adjust capacity without manual intervention, eliminating the overflow capacity planning required with traditional RPA implementations.

Minimal state management layers maintain execution context and handle recovery scenarios, storing only essential transaction data. This lightweight approach enables microbots to recover gracefully from interruptions while minimizing storage requirements and optimizing performance. Insurance providers processing high volumes of claims have achieved substantial improvements in recovery time objectives through distributed state management compared to centralized approaches used in conventional automation.

2.3. Orchestration and Coordination

Microbots typically operate within coordinated ecosystems rather than as isolated entities. Event-driven choreography enables them to communicate primarily through events, with each bot subscribing to relevant event types that trigger its execution. This loosely-coupled coordination model enables complex process orchestration without creating rigid dependencies between components, allowing individual microbots to evolve independently without disrupting overall workflows.

Lightweight orchestration layers maintain process context across multiple microbots without imposing heavy control structures. These orchestration services enable end-to-end process visibility while preserving the independence of individual automation components. Logistics companies have implemented distributed microbot networks coordinated through centralized orchestration to manage complex supply chain processes with higher resilience than previously achievable with monolithic automation solutions.

Distributed transaction management mechanisms maintain data consistency across microbot executions without tight coupling. This approach enables complex multi-step processes to maintain data integrity across distributed systems while supporting high transaction volumes. Financial services organizations have implemented microbot networks that maintain transactional consistency across global operations while supporting continuous processing across multiple time zones.

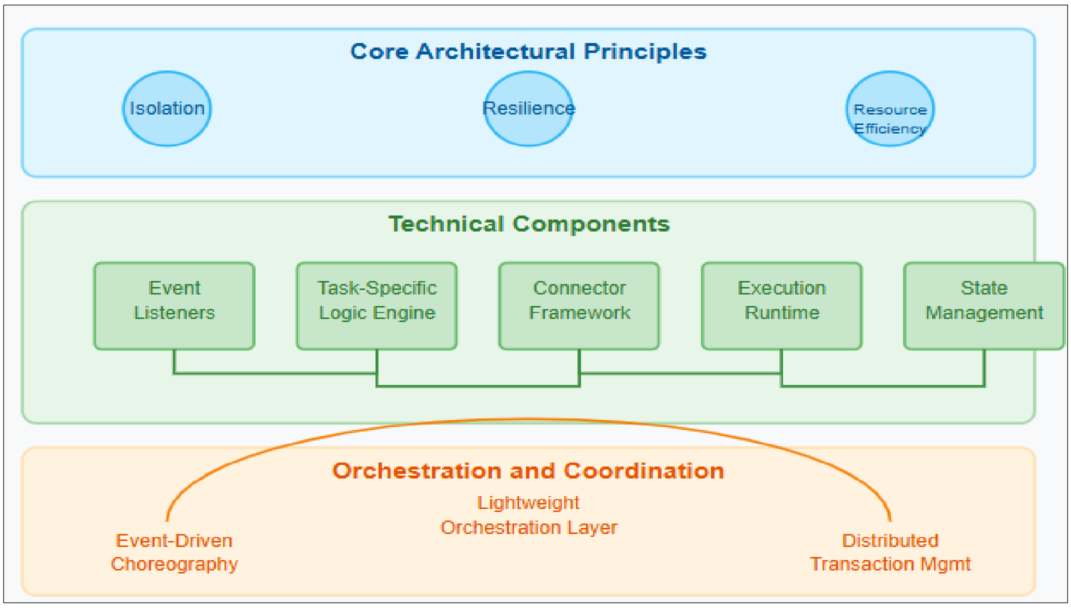

Figure 1 Microbot Architecture in RPA: Components and Interactions [3, 4]

3. Implementation Strategies and Best Practices

3.1. Process Decomposition Methodology

Successful microbot implementation begins with effective process decomposition—the systematic breakdown of complex workflows into discrete, automatable tasks. Research on sustainable automation practices indicates that organizations adopting structured decomposition methodologies achieve higher automation success rates while reducing resource consumption by integrating environmental sustainability considerations into the decomposition process [5]. Task Atomicity Analysis forms the foundation of effective microbot design by identifying independent work units that function as standalone automation targets. Financial services implementations reveal that processes decomposed with sustainability metrics in mind not only yield optimal performance but also minimize energy consumption during runtime execution.

Dependency Mapping provides essential context for microbot orchestration by documenting relationships between decomposed tasks. Enterprises implementing comprehensive dependency mapping demonstrate fewer integration failures and reduced carbon footprints through optimized execution patterns. The telecommunications sector has particularly benefited from mapping approaches that prioritize resource-efficient pathways through complex process networks. Reusability Assessment enables identification of common patterns that can be implemented as templates while simultaneously reducing development resource requirements. Insurance sector implementations show that systematic reusability assessment achieves remarkable efficiency improvements while supporting broader organizational sustainability goals through reduced computational requirements.

3.2. Development and Deployment Approaches

Microbots benefit substantially from modern software development practices aligned with digital sustainability principles. Low-Code Development Platforms have emerged as critical enablers for resource-efficient microbot creation, with organizations leveraging visual environments reporting faster implementation cycles with lower energy requirements. Healthcare providers have successfully deployed microbots in short timeframes using these platforms while simultaneously decreasing their technology-related environmental impact through optimized development lifecycles.

Continuous Integration/Continuous Deployment practices have transformed microbot reliability while supporting sustainable IT operations. Organizations implementing automated testing and deployment pipelines report fewer production issues and reduced resource waste from failed deployments. Manufacturing implementations demonstrate how intelligent test optimization reduces unnecessary computational overhead during deployment cycles. DevOps Integration facilitates rapid iteration through collaboration that considers both operational efficiency and sustainability metrics. Recent research on cyber-physical systems integration with RPA indicates that DevOps-enabled microbot architectures achieve improved responsiveness while significantly reducing the overall carbon footprint of automation solutions [6]. Container-Based Deployment represents a transformative practice that achieves consistent deployment with simplified scaling while enabling more precise resource allocation, substantially reducing idle computing capacity compared to traditional virtualized environments.

3.3. Governance and Management Framework

Scaling microbots requires structured governance frameworks that incorporate sustainability metrics alongside traditional performance indicators. Organizations with governance models that include energy efficiency targets deploy more microbots while consuming fewer computing resources. Centralized Registry implementation has emerged as a foundational practice, with comprehensive inventories enabling both improved operational visibility and resource optimization. Healthcare providers utilizing registries that track sustainability metrics alongside functional attributes report more efficient resource allocation across their automation portfolios.

Performance Monitoring enables tracking of both business outcomes and resource consumption patterns. Telecommunications implementations incorporating power utilization effectiveness metrics within their monitoring frameworks have remediated energy-inefficient automation components proactively. Version Control practices ensure proper microbot lifecycle management while supporting sustainability through efficient code reuse patterns. Retail sector experiences demonstrate how version management strategies that prioritize component efficiency result in reduced infrastructure requirements. Security Controls implementation addresses authentication and data handling requirements while balancing protection with optimization. Financial implementations show that appropriately scoped security frameworks reduce computational overhead while maintaining robust protection for sensitive operations.

Cross-industry analysis confirms that microbots designed with sustainability principles require fewer resources throughout their operational lifecycle while delivering equivalent business value.

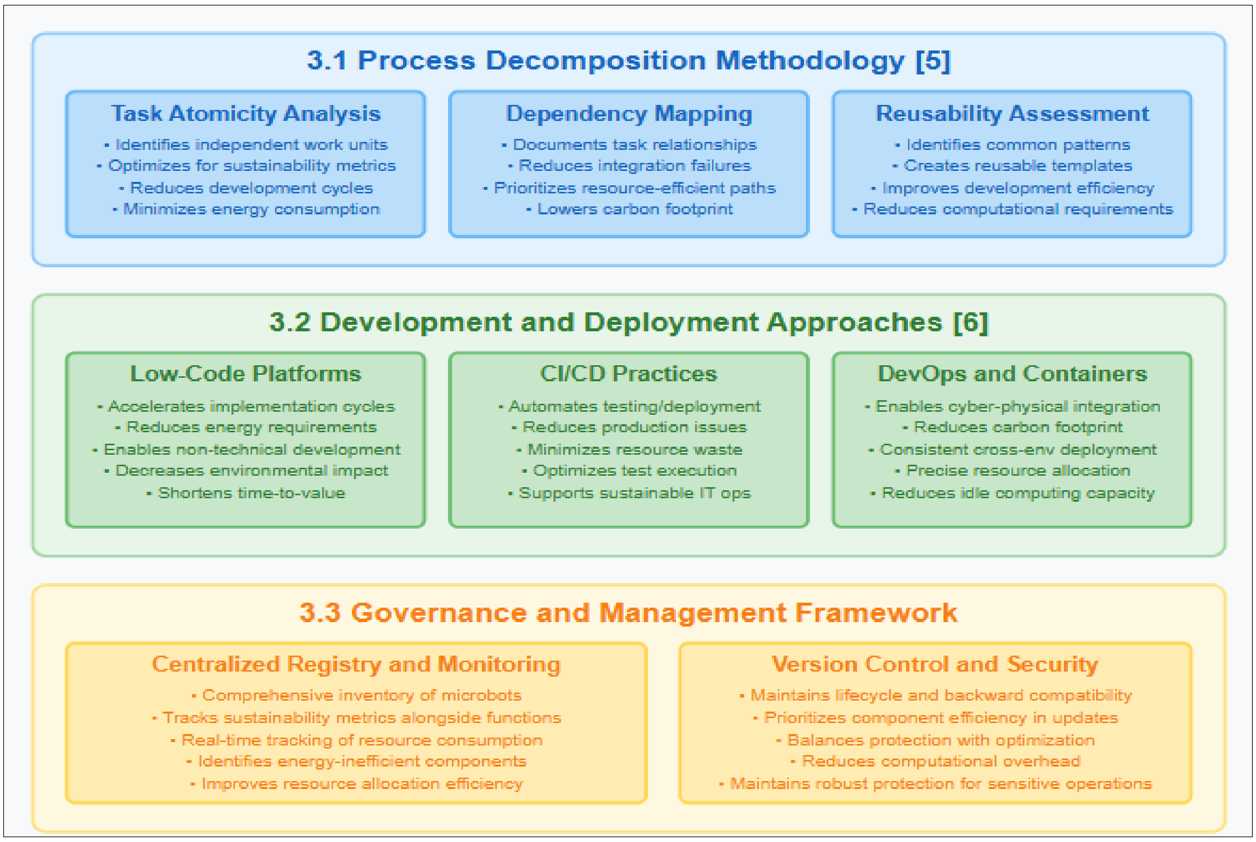

Figure 2 Sustainable Microbot Implementation: Strategies and Best Practices [5, 6]

4. Industry Applications and Case Studies

4.1. Financial Services

The financial sector has emerged as an early adopter of microbot technology for specific operational challenges. Research indicates microbot architectures provide enhanced security and process efficiency compared to traditional RPA implementations in banking environments [7]. The fragmented nature of financial systems, with numerous legacy platforms alongside modern applications, creates an ideal environment for specialized interface microbots.

Transaction Reconciliation represents a primary application where microbots match transactions across multiple systems with superior accuracy. Specialized reconciliation microbots excel at handling specific account types or transaction categories that previously required extensive manual intervention, resulting in shortened reconciliation cycles and improved match rates.

Regulatory Reporting benefits from coordinated microbot networks for compliance requirements across frameworks like FATCA and anti-money laundering regulations. These implementations reduce compliance effort while improving report accuracy in an increasingly scrutinized regulatory environment.

Case Study: An investment bank implemented 75 microbots for post-trade processing, reducing execution time by 43% and achieving 99.8% uptime compared to 96.5% with their previous monolithic solution. The distributed architecture proved particularly valuable during market volatility periods when processing demands spiked.

4.2. Healthcare and Life Sciences

Healthcare organizations have embraced microbot architecture to overcome interoperability challenges while maintaining strict compliance with patient privacy regulations [8]. Distributed automation approaches align particularly well with healthcare's decentralized operational model across facilities.

Claims Processing has emerged as a dominant application, with microbots handling different insurance providers and verification steps. The healthcare revenue cycle involves multiple payer systems with distinct requirements, making it ideal for specialized microbot deployment. Organizations report faster reimbursement and decreased denial rates through payer-specific verification sequences.

Clinical Documentation benefits from microbots extracting specific data elements from electronic records based on medical specialties. Specialized microbots optimized for cardiology or emergency medicine workflows demonstrate higher extraction accuracy than general-purpose solutions while preserving specialized document handling capabilities.

Case Study: A healthcare network deployed microbots to automate patient eligibility verification across 12 insurance providers. Each microbot specialized in a specific payer's portal, reducing verification time from 8 minutes to 45 seconds while improving accuracy from 92% to 99.3%. Pre-visit verification significantly improved both patient experience and revenue capture.

4.3. Customer Service Operations

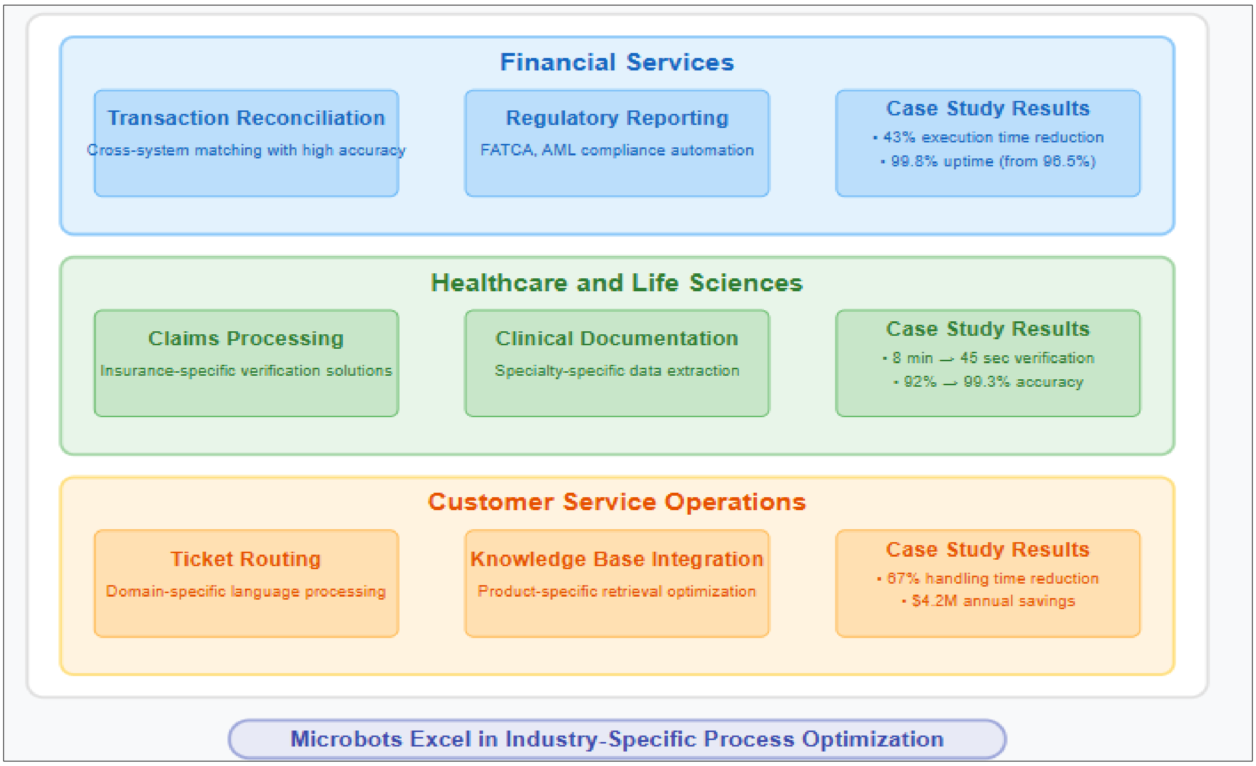

Figure 3 Microbot Applications Across Industries: Case Studies and Implementations [7, 8]

Contact centers benefit from microbot implementations that handle specific interaction types with precision. The distributed nature allows gradual implementation, with organizations adding specialized microbots as processes are optimized rather than requiring comprehensive transformation.

Ticket Routing microbots analyze incoming service requests using domain-specific language processing tailored to particular product lines, significantly improving routing accuracy and reducing handle times through appropriate initial assignment.

Knowledge Base Integration provides efficiency through specialized bots retrieving information based on specific inquiries. The granular nature allows knowledge retrieval to be optimized for particular product areas, enabling more precise responses than general solutions that struggle with ambiguous queries.

Case Study: A telecommunications provider implemented 50+ microbots for customer service processes, achieving 67% reduction in handling time for billing inquiries and 82% decrease in manual processing for service changes, generating $4.2M annual savings while significantly improving satisfaction metrics.

5. Challenges of Microbot Implementation

Despite their substantial benefits, microbots present organizations with several implementation challenges that must be effectively addressed to realize their full potential. These challenges span technical, organizational, and strategic dimensions that can significantly impact adoption success.

5.1. Technical Complexity and Integration Challenges

While individual microbots are lightweight, managing distributed networks of specialized automation agents introduces architectural complexity that organizations frequently underestimate. The technical debt associated with managing numerous specialized components can potentially offset efficiency gains without proper architectural governance.

Integration with legacy systems presents particular challenges as these systems often lack modern APIs necessary for effective microbot interaction. Organizations without comprehensive API strategies find themselves creating complex workarounds that undermine the efficiency benefits of microbot architectures.

Data consistency across distributed automation components requires sophisticated orchestration mechanisms that many organizations lack. These challenges necessitate implementation of distributed transaction patterns that add complexity to microbot ecosystems.

5.2. Governance and Management Overhead

Proliferation of specialized microbots can lead to governance challenges without comprehensive management frameworks. This governance gap manifests in tracking, version control, and dependency management issues that compound as microbot deployments scale.

Maintaining visibility across numerous specialized automation components presents monitoring challenges that traditional RPA platforms do not encounter. This overhead necessitates investment in specialized monitoring solutions with distributed tracing capabilities.

Security considerations become more complex with distributed automation components operating across system boundaries. These security gaps create potential vulnerabilities that require sophisticated identity management and access control frameworks to mitigate effectively.

5.3. Organizational and Skills Challenges

Transitioning from traditional RPA to microbot architectures requires specialized skills that many organizations lack. This gap necessitates significant investment in training and recruitment to build teams capable of designing and managing distributed automation solutions.

Organizational resistance remains a significant barrier, with automation teams accustomed to monolithic approaches often resisting distributed architectures. This resistance often manifests as preference for familiar monolithic solutions despite their recognized limitations.

Cultural shifts toward service-oriented thinking present substantial challenges for organizations with process-oriented automation teams. Organizations failing to address these cultural dimensions typically experience suboptimal outcomes despite investing in appropriate technical solutions.

6. Future Directions and Emerging Trends

6.1. AI and Machine Learning Integration

The next generation of microbots is increasingly leveraging AI capabilities, creating sophisticated autonomous systems that can adapt to complex industrial environments. Recent industrial research demonstrates that microbots integrating multiple AI techniques achieve substantially higher task completion rates in manufacturing settings [9]. This convergence represents a significant evolution from simple rule-based automation toward truly intelligent operational support systems.

Intelligent Task Selection emerges as a critical capability where algorithms dynamically assign tasks based on real-time conditions and historical results. Manufacturing implementations show that intelligent selection mechanisms substantially reduce material waste while improving throughput in assembly operations. These systems continuously assess numerous operational variables to optimize task distribution, allowing for rapid adaptation to changing production requirements without manual intervention.

Natural Language Processing (NLP) Enhanced Microbots interpret and prioritize unstructured communication within industrial contexts. Factory implementations demonstrate these specialized agents process maintenance requests, quality notes, and operator feedback with remarkable accuracy, enabling seamless integration between human workers and automated systems. The focused nature of microbots allows for specialized language processing optimized for specific industrial terminology rather than generic comprehension.

Predictive Analytics Integration enables preemptive responses to developing issues before they affect operations. Production lines utilizing predictive microbots maintain higher sustained output by addressing equipment variations and material inconsistencies proactively. These systems process continuous data streams from industrial sensors to forecast potential disruptions, shifting from reactive to proactive operational models with measurable efficiency improvements.

Self-Optimizing Microbots leverage reinforcement learning to refine execution patterns without reprogramming. Industrial implementations demonstrate they continuously improve material handling efficiency and energy consumption patterns through iterative learning cycles. This autonomous optimization extends beyond initial programming limitations, allowing systems to discover novel operational efficiencies unreachable through traditional automation approaches.

6.2. Edge Computing and Distributed Execution

Advanced manufacturing environments increasingly deploy computation capabilities directly alongside operational technology, enabling real-time decision making without cloud dependencies. Research indicates that edge-deployed microbot architectures deliver critical latency reductions necessary for high-precision industrial applications [10]. This distributed approach ensures operational continuity even during network disruptions while maintaining strict data sovereignty requirements.

Edge-Deployed Microbots operate at the machinery interface level, processing sensor data and executing control functions with minimal response times. Production environments implementing edge microbots demonstrate dramatically reduced cycle times with consistent performance regardless of network conditions. These implementations distribute specialized control functions across multiple edge devices throughout production facilities, ensuring localized responsiveness while maintaining coordination with broader systems.

Cross-System Orchestration provides seamless integration between operational technology and information systems across multiple environments. Factory implementations show that distributed orchestration significantly improves material flow coordination across disconnected production cells. The orchestration layer dynamically optimizes workload distribution based on real-time capacity and throughput metrics, ensuring consistent production flow despite equipment variations.

Event Mesh Integration creates resilient communication frameworks connecting distributed automation components across complex industrial environments. Production facilities implementing mesh architectures demonstrate exceptional operational resilience with continuous production capability despite localized system failures. This decoupled communication approach enables flexible reconfiguration of production flows without system-wide modifications, supporting agile manufacturing strategies.

6.3. Democratization and Citizen Development

The accessibility of industrial automation is expanding beyond specialized engineers to include production specialists with domain expertise but limited programming skills. Manufacturing organizations empowering shop floor experts to develop microbots achieve more practical automation solutions with higher adoption rates than those relying solely on engineering-led development.

No-Code Microbot Platforms enable production specialists to create specialized automation solutions addressing specific operational challenges. Factory implementations show production teams using these platforms develop effective solutions for material handling, quality inspection, and equipment monitoring within days rather than months. These platforms incorporate industrial-specific components covering common manufacturing workflows, enabling rapid deployment without traditional programming requirements.

Bot Marketplaces facilitate sharing of specialized industrial automation solutions across manufacturing sectors. Organizations utilizing these marketplaces implement standardized automation solutions for common processes with minimal customization requirements. This approach accelerates digital transformation initiatives by leveraging proven solutions adaptable to specific manufacturing environments.

Community-Driven Development creates collaborative ecosystems where engineers and production specialists jointly develop industrial automation solutions. Factory implementations demonstrate these collaborative approaches result in automation solutions that better reflect operational realities while meeting technical requirements. This cross-functional approach ensures automation addresses actual production challenges rather than theoretical efficiency improvements.

6.4. Microbots and the Future of AI Integration

The convergence of microbot architecture with advanced AI capabilities represents a significant frontier in enterprise automation, with profound implications for how organizations implement intelligent systems. Microbot architectures provide optimal frameworks for deploying domain-specific AI models that would be impractical in monolithic automation solutions. This architectural alignment enables enterprises to leverage specialized intelligence across diverse operational domains.

Federated Learning approaches are emerging as powerful enablers for microbots to collectively improve while maintaining operational independence. Distributed microbot networks implementing federated learning achieve significantly faster optimization rates compared to centralized learning approaches. This distributed learning architecture allows specialized automation agents to benefit from collective experiences while maintaining task-specific optimization.

Explainable AI integration addresses critical transparency requirements in regulated industries through componentized automation that provides clearer decision trails. Financial services implementations leveraging explainable microbots demonstrate superior compliance outcomes compared to "black box" automation solutions. The granular nature of microbot architecture allows precise attribution of decision factors, enabling effective audit trails and regulatory reporting.

Human-AI Collaborative Frameworks are evolving to create seamless interactions between knowledge workers and specialized microbots. Organizations implementing microbot-based cognitive assistants achieve higher productivity gains compared to those deploying generalized AI assistants. These collaborative frameworks dynamically allocate tasks between humans and automation components based on comparative advantages, creating truly symbiotic work environments.

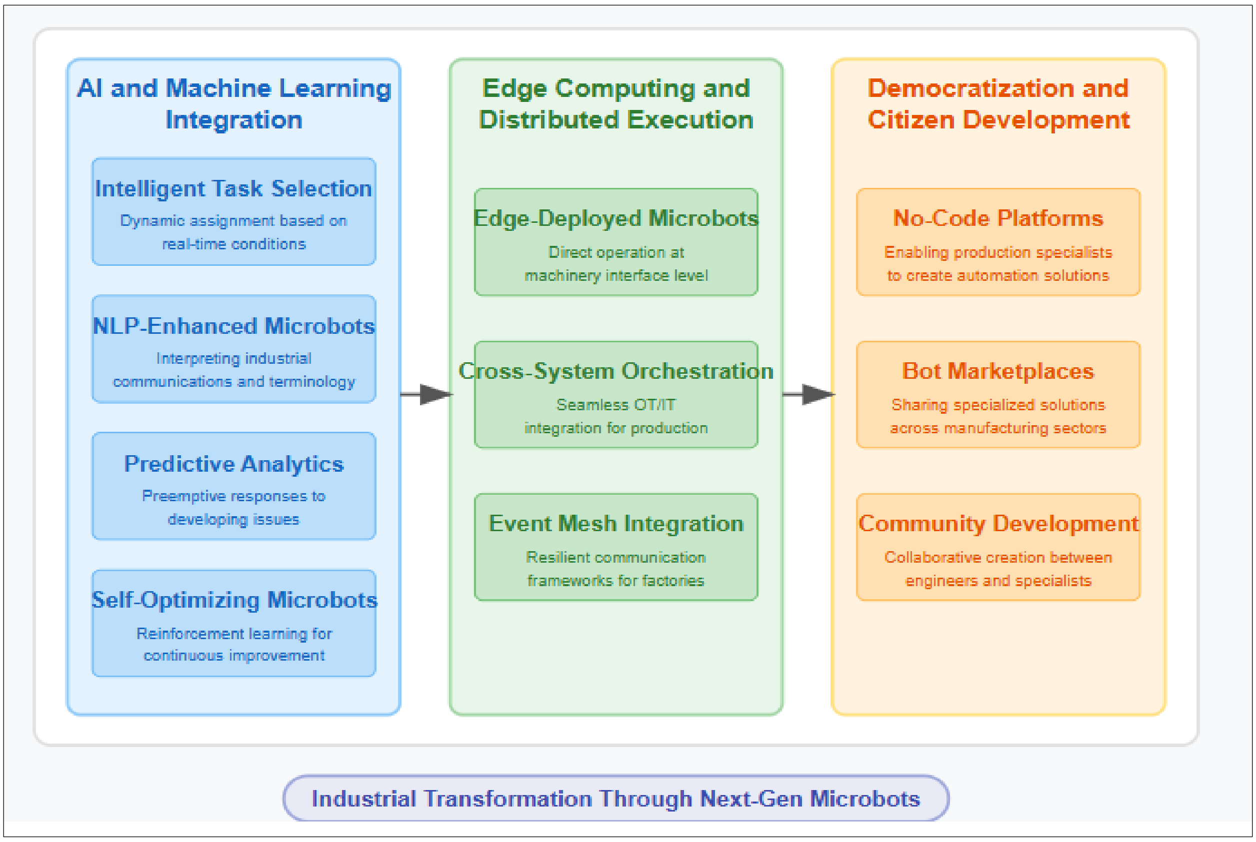

Figure 3 Future Directions in Industrial Microbot Technology [9, 10]

7. Conclusion

Microbots signify a profound advancement in the automation landscape, offering organizations a pathway to achieve scalable, agile process automation without the constraints inherent in traditional RPA implementations. By decomposing complex workflows into discrete, independently deployable components, microbots enable enterprises to extend automation coverage while simultaneously reducing technical overhead and resource consumption. The architectural principles underpinning microbots-isolation, resilience, and resource efficiency-align perfectly with broader technology trends toward microservices and event-driven systems, positioning these specialized agents for continued evolution alongside advancements in cloud computing and artificial intelligence. Across diverse sectors from financial services to healthcare and customer operations, microbots demonstrate compelling benefits through accelerated implementation cycles, enhanced adaptability to system changes, and improved operational performance. The emergence of AI-enhanced capabilities, edge deployment patterns, and democratized development platforms further expands the potential impact of microbots, creating opportunities for comprehensive automation that adapts dynamically to changing business requirements. For organizations pursuing digital transformation initiatives, microbots represent a sustainable approach to building adaptable digital workforces—balancing transformative potential with practical considerations of resource constraints and organizational agility while delivering tangible operational excellence across enterprise environments.

References

[1] Shailza Kaushal, "An Ultimate Guide To RPA (Robotic Process Automation)," Signity, 2025. [Online]. Available: https://www.signitysolutions.com/blog/robotic-process-automation-guide

[2] David Wright, Dupe Witherick and Marina Gordeeva, "The robots are ready. Are you? | Untapped advantage in your digital workforce," Deloitte, 2018. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/process-and-operations/us-cons-global-rpa-survey.pdf

[3] Steve Shah, "RPA Platform Cloud Architecture: 5 Must-haves and 3 to Avoid," Automation Anywhere, 2020. [Online]. Available: https://www.automationanywhere.com/company/blog/product-insights/rpa-platform-cloud-architecture-5-must-haves-and-3-to-avoid

[4] Ravi Sharma, "RPA vs Traditional Automation – What's the Actual Difference?," Hurix Digital, 2025. [Online]. Available: https://www.hurix.com/blogs/rpa-vs-traditional-automation-whats-the-actual-difference/

[5] Leonel Patrício, et al., "Sustainable Implementation of Robotic Process Automation Based on a Multi-Objective Mathematical Model," Sustainability, 2023. [Online]. Available: https://www.mdpi.com/2071-1050/15/20/15045

[6] Jean-Pierre Busch, Lennart Reiher and Lutz Eckstein, "Enabling the Deployment of Any-Scale Robotic Applications in Microservice Architectures through Automated Containerization," IEEE Xplore, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10611586

[7] Sudhir Pant and Manjari Agarwal, "A Study of Robotics in Banking and Financial Services," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/374982628_A_Study_of_Robotics_in_Banking_and_Financial_Services

[8] Vadim Struk, "Robotic Process Automation in Healthcare: an Exploration," Relevant Software, 2024. [Online]. Available: https://relevant.software/blog/robotic-process-automation-in-healthcare/

[9] Vugar Hacimahmud Abdullayev, et al., "Integration of Artificial Intelligence and Robotics into the industrial sector," ResearchGate, 2025. [Online]. Available: https://www.researchgate.net/publication/388840757_Integration_of_Artificial_Intelligence_and_Robotics_into_the_industrial_sector

[10] Joshua Boluwatife Adelusi, Adelusi Ajisola and Saheed Olabanji, "Edge Computing Architectures for Real-Time Robotic Control and Decision-Making," ResearchGate, 2025. [Online]. Available: https://www.researchgate.net/publication/390897926_Edge_Computing_Architectures_for_Real-Time_Robotic_Control_and_Decision-Making