Automation Stack Selection Guide

Microsoft vs Open-Source vs Hybrid: Choosing the Right Platform for BI Automation

Published by MBIC | 2025

Executive Summary

Organizations implementing Power BI automation face a critical decision: which automation platform to use. Microsoft Power Automate, open-source tools like n8n, cloud platforms like Make, or a hybrid approach?

This guide provides a comprehensive comparison to help you make an informed decision based on your specific requirements.

Key Decision Factors:

- Total cost of ownership
- Technical capabilities and limitations
- Integration complexity
- Security and compliance
- Scalability requirements
- Internal technical capabilities
- Vendor lock-in concerns

Bottom Line: There's no universal "best" choice. The right platform depends on your organization's specific needs, constraints, and long-term strategy.

Table of Contents

6. Migration Considerations
7. Hybrid Architecture Patterns
8. Implementation Examples
9. Vendor Analysis
10. Recommendations by Organization Type

---

1. Platform Overview

Microsoft Power Automate

What it is: Microsoft's low-code workflow automation platform, tightly integrated with the Microsoft 365 ecosystem and Azure services.

Key Strengths:

- Deep integration with Power BI and Microsoft ecosystem
- Enterprise security and compliance (SOC 2, HIPAA, etc.)
- Extensive pre-built connectors (400+)
- Low-code interface accessible to non-developers
- Microsoft support and SLAs

Key Limitations:

- Premium connectors can be expensive
- Limited complex logic capabilities
- Proprietary platform (vendor lock-in)
- Per-flow and per-user pricing

Best For:

- Microsoft-centric organizations
- Enterprise compliance requirements
- Teams with limited coding expertise
- Organizations already using Microsoft 365

Open-Source: n8n

What it is: Fair-code workflow automation tool (source-available with commercial restrictions) that can be self-hosted or cloud-hosted.

Key Strengths:

- Self-hosted option (full control)
- Unlimited executions
- Extensive customization
- Active community
- No per-execution costs
- Visual workflow builder
- Supports custom code (JavaScript)

Key Limitations:

- Requires infrastructure management
- Limited enterprise support
- Smaller connector library than commercial tools
- Security/compliance responsibility on you
- Fair-code license (not fully open-source)

Best For:

- Cost-conscious organizations
- High-volume automation (thousands of executions)
- Teams with DevOps capabilities
- Organizations needing full control
- Complex custom logic requirements

Cloud Platforms: Make (formerly Integromat) & Zapier

What they are: Cloud-based integration platforms with visual builders and extensive app connectors.

Key Strengths:

- Quick to start (no infrastructure)
- Large connector ecosystems
- Visual workflow design
- Good for simple integrations
- Regular updates and new connectors

Key Limitations:

- Can become expensive at scale
- Limited control over execution environment
- Data privacy concerns (all data flows through vendor)
- Execution limits on lower tiers
- Vendor lock-in

Best For:

- Quick prototypes
- Small to medium automation volumes
- Teams wanting simplicity
- Multi-SaaS integrations
- Organizations without DevOps resources

Custom Development (Python/Node.js)

What it is: Writing automation logic in general-purpose programming languages like Python or Node.js, hosted on Azure Functions, AWS Lambda, or your own servers.

Key Strengths:

- Ultimate flexibility
- No platform limitations
- Full control over logic
- Can optimize for performance
- No licensing costs for platforms

Key Limitations:

- Requires skilled developers
- Higher initial development time
- Ongoing maintenance burden
- No visual workflow builder
- Building connectors from scratch

Best For:

- Highly custom requirements
- Organizations with strong dev teams
- Performance-critical applications
- Unique integration needs
- Long-term strategic platforms

---

2. Detailed Platform Comparison

Feature Comparison Matrix

Feature	Power Automate	n8n	Make	Zapier	Custom Code
Ease of Setup	★★★★★	★★	★★★★★	★★★★★	★
Power BI Integration	★★★★★	★★★★	★★★	★★★	★★★★★
Cost at Scale	★★	★★★★★	★★★	★★	★★★★
Flexibility	★★★	★★★★	★★★	★★★	★★★★★
Enterprise Security	★★★★★	★★★	★★★★	★★★★	★★★★★
Support Quality	★★★★★	★★	★★★★	★★★★	★
Scalability	★★★★	★★★★	★★★★	★★★	★★★★★
Development Speed	★★★★	★★★	★★★★	★★★★	★★
Maintenance Burden	★★★★★	★★★	★★★★★	★★★★★	★★
Vendor Lock-in Risk	★★	★★★★	★★	★★	★★★★★

Power BI Integration Comparison

Integration Aspect	Power Automate	n8n	Make	Custom Code
Native Triggers	Yes (alerts, refreshes)	Via API	Via API	Via API
Dataset Refresh	Built-in action	HTTP node	HTTP module	REST API
Report Export	Built-in action	HTTP node	HTTP module	REST API
Authentication	Managed	Manual OAuth setup	Manual setup	Full control
Error Handling	Built-in	Manual	Manual	Full control
RLS Support	Yes	Via API	Via API	Full control

Power Automate Example:

yaml

```yaml
# Trigger: When Power BI alert fires
# (Native, no configuration needed)

When data driven alert triggered
  Alert: Revenue Below Target

# Action: Refresh dataset
# (One-click configuration)

Refresh dataset
  Workspace: Sales
  Dataset: Revenue Dashboard
```

n8n Example:

json

```
{
 "nodes": [
  {
   "name": "Power BI Webhook",
   "type": "n8n-nodes-base.webhook",
   "webhookId": "powerbi-alert"
  },
  {
   "name": "Get Access Token",
   "type": "n8n-nodes-base.httpRequest",
   "parameters": {
    "url": "https://login.microsoftonline.com/{{$env.TENANT_ID}}/oauth2/v2.0/token",
    "method": "POST",
    "authentication": "genericCredentialType",
    "sendBody": true,
    "bodyParameters": {
     "grant_type": "client_credentials",
     "client_id": "={{$env.CLIENT_ID}}",
     "client_secret": "={{$env.CLIENT_SECRET}}",
     "scope": "https://analysis.windows.net/powerbi/api/.default"
    }
   }
  },
  {
   "name": "Refresh Dataset",
   "type": "n8n-nodes-base.httpRequest",
   "parameters": {
    "url": "https://api.powerbi.com/v1.0/myorg/groups/{{$node['Webhook'].json['workspaceId']}}/datasets/{{$node['Webho
    "method": "POST",
    "authentication": "genericCredentialType",
    "sendHeaders": true,
    "headerParameters": {
     "Authorization": "Bearer {{$node['Get Access Token'].json['access_token']}}"
    }
   }
  }
 ]
}
```

Custom Code Example:

```
typescript
```

```
// Azure Function (TypeScript)
import { AzureFunction, Context, HttpRequest } from "@azure/functions";
import { ClientSecretCredential } from "@azure/identity";
import axios from "axios";

const httpTrigger: AzureFunction = async (
  context: Context,
  req: HttpRequest
): Promise<void> => {
  const { datasetId, workspaceId } = req.body;

  // Get access token
  const credential = new ClientSecretCredential(
    process.env.TENANT_ID,
    process.env.CLIENT_ID,
    process.env.CLIENT_SECRET
  );

  const token = await credential.getToken(
    "https://analysis.windows.net/powerbi/api/.default"
  );

  // Refresh dataset
  try {
    await axios.post(
      `https://api.powerbi.com/v1.0/myorg/groups/${workspaceId}/datasets/${datasetId}/refreshes`,
      {},
      {
        headers: {
          Authorization: `Bearer ${token.token}`,
        },
      }
    );

    context.res = {
      status: 200,
      body: { success: true, message: "Dataset refresh triggered" },
    };
  } catch (error) {
    context.res = {
      status: 500,
      body: { success: false, error: error.message },
    };
```

```
  }
};

export default httpTrigger;
```

Analysis:

- **Power Automate:** Fastest to implement, requires minimal technical knowledge
- **n8n:** Middle ground - more complex than Power Automate but more flexible
- **Custom Code:** Maximum flexibility but highest development time

---

3. Total Cost of Ownership Analysis

Scenario: Mid-Size Organization

Assumptions:

- 100 automated workflows
- 50,000 workflow executions per month
- 25 users creating/managing workflows
- 3-year timeframe

Power Automate Costs

```
Licensing:
- Power Automate Premium: $15/user/month × 25 users = $375/month
- Premium connectors included
- Unlimited executions for attended flows

OR

- Power Automate Process: $150/month per flow × 100 flows = $15,000/month
  (For unattended automation)

Realistic Hybrid Approach:
- 10 Power Automate Process licenses (critical flows): $1,500/month
- 25 Power Automate Premium (user-triggered): $375/month
- Total: $1,875/month = $22,500/year

Additional Costs:
```

- Microsoft 365 E3 licenses (if not already): $0 (assume existing)
- Azure infrastructure: $200/month for Azure Functions, Key Vault, etc.
- Training: $5,000 year 1, $1,000/year ongoing
- Support: Included with Microsoft

3-Year TCO:
Year 1: $22,500 + $2,400 + $5,000 = $29,900
Year 2-3: $24,900/year each
Total: $79,700

n8n Costs

Self-Hosted Option:

Infrastructure:
- Azure VM (4 cores, 16GB RAM): $300/month
- Azure SQL Database: $100/month
- Azure Blob Storage: $50/month
- Load balancer: $50/month
- Total: $500/month = $6,000/year

OR n8n Cloud:
- Pro plan: $50/month (20K executions)
- Need 3 Pro instances for 50K executions: $150/month = $1,800/year

Licensing (self-hosted enterprise features):
- n8n Enterprise: $500/month = $6,000/year (optional)
  OR
- Community edition: $0

Personnel:
- DevOps maintenance: 10 hours/month × $100/hour = $1,000/month = $12,000/year
- Development/customization: $10,000 year 1, $3,000/year ongoing

Training:
- Team training: $3,000 year 1

3-Year TCO (Self-Hosted Community):
Year 1: $6,000 + $12,000 + $10,000 + $3,000 = $31,000
Year 2-3: $18,000/year each
Total: $67,000

3-Year TCO (n8n Cloud Pro):

Year 1: $1,800 + $10,000 + $3,000 = $14,800
Year 2-3: $4,800/year each
Total: $24,400

(Note: Cloud assumes minimal DevOps, but limits customization)

Make Costs

Pricing (as of 2025):
- Pro: $29/month per user (10K operations)
- Teams: $99/month (unlimited users, 100K operations)
- Enterprise: Custom pricing

For 50K operations/month:
- Teams plan: $99/month = $1,188/year
  OR
- Enterprise (for higher volume): ~$500/month = $6,000/year

3-Year TCO:
Year 1: $6,000 + $2,000 (training) = $8,000
Year 2-3: $6,000/year each
Total: $20,000

Custom Code (Azure Functions)

Infrastructure:
- Azure Functions Premium: $200/month
- Azure SQL Database: $100/month
- Application Insights: $50/month
- Key Vault: $10/month
- Total: $360/month = $4,320/year

Development:
- Initial development: 200 hours × $150/hour = $30,000
- Ongoing maintenance: 10 hours/month × $150/hour = $18,000/year

3-Year TCO:
Year 1: $4,320 + $30,000 = $34,320
Year 2-3: $22,320/year each
Total: $78,960

Cost Comparison Summary

Platform	Year 1	Year 2	Year 3	3-Year Total	Per Execution
Power Automate	$29,900	$24,900	$24,900	**$79,700**	$0.044
n8n Self-Hosted	$31,000	$18,000	$18,000	**$67,000**	$0.037
n8n Cloud	$14,800	$4,800	$4,800	**$24,400**	$0.014
Make	$8,000	$6,000	$6,000	**$20,000**	$0.011
Custom Code	$34,320	$22,320	$22,320	**$78,960**	$0.044

Key Insights:

1. **Make is cheapest** for this volume but has limitations
2. **n8n Cloud** excellent cost/value for medium scale
3. **Power Automate & Custom Code** similar costs but different value propositions
4. **n8n Self-Hosted** middle ground on cost, maximum control

Volume Sensitivity: At 500K executions/month:

- Make: $50,000/year (becomes expensive)
- n8n Cloud: $5,000/year (still cheap)
- Power Automate: $22,500/year (flat rate)
- n8n Self-Hosted: $18,000/year (scales well)

---

4. Use Case Scenarios

Scenario 1: Simple Alert-Based Workflows

Requirement:

- 20 Power BI alerts trigger emails/Teams messages
- Low complexity
- 1,000 executions/month

Best Choice: Power Automate

Why:

- Quick to set up (hours, not days)

- Native Power BI integration
- Pre-built Teams connector
- No infrastructure to manage
- Cost: ~$150/month (minimal)

When to Reconsider:

- If you expect to add 100+ more workflows
- If you need custom logic beyond email/Teams
- If budget is extremely tight

Scenario 2: High-Volume Data Processing

Requirement:

- Process 100K+ rows of data daily
- Complex transformations
- Multiple system integrations
- Performance critical

Best Choice: Custom Code (Azure Functions)

Why:

- Can optimize for performance
- Parallel processing
- Efficient memory usage
- Full control over execution
- Cost-effective at volume

Implementation:

```typescript
```

```
// Azure Function with parallel processing
import { chunk } from "lodash";

export async function processLargeDataset(data: any[]) {
  // Process in batches of 1000
  const batches = chunk(data, 1000);

  // Process batches in parallel (10 at a time)
  const results = [];
  for (let i = 0; i < batches.length; i += 10) {
    const batchPromises = batches
      .slice(i, i + 10)
      .map((batch) => processBatch(batch));

    const batchResults = await Promise.all(batchPromises);
    results.push(...batchResults);
  }

  return results;
}

async function processBatch(batch: any[]) {
  // Transform data
  const transformed = batch.map((row) => ({
    ...row,
    calculated_field: complexCalculation(row),
  }));

  // Bulk insert to database
  await bulkInsert(transformed);

  return transformed.length;
}
```

When to Reconsider:

- If development team lacks expertise
- If time-to-market is critical
- If volume might decrease significantly

Scenario 3: Multi-SaaS Integration

Requirement:

- Connect Salesforce, HubSpot, Slack, Google Sheets, Power BI
- Standard transformations
- Medium complexity
- 10,000 executions/month

Best Choice: Make or n8n Cloud

Why:

- Pre-built connectors for all systems
- Visual workflow builder
- Quick to implement
- Reasonable cost at this scale

Make:

- Easier setup
- More polished UI
- Better for non-technical users

n8n Cloud:

- More cost-effective
- More customizable
- Better for teams with some technical skills

Scenario 4: Enterprise with Strict Compliance

Requirement:

- HIPAA/SOC 2 compliance mandatory
- All data must stay in company's Azure tenant
- Audit logging required

- 50 workflows

Best Choice: Power Automate or n8n Self-Hosted

Power Automate:

- Built-in compliance certifications
- Microsoft BAA available
- Enterprise audit logging
- Support for compliance questions

n8n Self-Hosted:

- Complete data control (never leaves your network)
- Self-managed compliance
- Custom audit implementation
- Lower cost but higher responsibility

Decision Factor:

- Choose Power Automate if you need vendor compliance documentation
- Choose n8n if you have strong internal compliance team

Scenario 5: Startup with Limited Resources

Requirement:

- Small team (5 people)
- Limited budget
- Need to move fast
- Uncertain future scale

Best Choice: Make or n8n Cloud

Why:

- Low upfront cost
- No infrastructure management

- Quick to start
- Can migrate later if needed

Start with Make if:

- Team is non-technical
- Need reliability over customization
- Want extensive pre-built connectors

Start with n8n Cloud if:

- Team has some technical skills
- Budget is very tight
- Want flexibility to self-host later

---

5. Decision Framework

Step 1: Assess Your Requirements

Technical Capabilities:

- [ ] Strong development team available
- [ ] DevOps resources available
- [ ] Comfort with infrastructure management
- [ ] Need for custom logic
- [ ] Performance requirements

Score:

- 0-1 checks: Low-code platform (Power Automate, Make)
- 2-3 checks: Flexible platform (n8n)
- 4-5 checks: Custom code or hybrid

Business Constraints:

- [ ] Strict budget limitations
- [ ] Compliance requirements (HIPAA, SOC 2)
- [ ] Already invested in Microsoft ecosystem
- [ ] Need for enterprise support

☐ Fast time-to-market required

Compliance-Driven:

- Yes to compliance + Microsoft ecosystem → Power Automate
- Yes to compliance + data sovereignty → n8n Self-Hosted
- No strict compliance → More options available

Budget-Driven:

- Very limited → Make or n8n Cloud
- Medium → n8n Self-Hosted or Make
- Large → Any option based on other factors

Step 2: Calculate Your Volume

```
Workflow Count: _____
Executions per Month: _____
Growth Rate Expected: _____%

If Executions > 100K/month:
  → n8n Self-Hosted or Custom Code

If Executions < 10K/month:
  → Any platform works

If 10K-100K:
  → Compare costs using TCO calculator
```

Step 3: Evaluate Integration Needs

Primary Integrations:

☐ Power BI (all platforms support)
☐ Microsoft 365 apps
☐ Salesforce/HubSpot
☐ Custom APIs
☐ Legacy systems

If mostly Microsoft: → Power Automate makes sense

If diverse SaaS: → Make or n8n

If custom/legacy: → Custom code or n8n (more flexibility)

Step 4: Assess Technical Debt Tolerance

Low tolerance (want to minimize): → Cloud platforms (Power Automate, Make, n8n Cloud)

Medium tolerance: → n8n Self-Hosted (some maintenance)

High tolerance: → Custom code (maximum flexibility, maximum maintenance)

Step 5: Consider Migration Path

Ask yourself: "If we outgrow this platform in 2 years, how hard is it to migrate?"

Migration Difficulty (1-5, 5 = hardest):

- Power Automate → Other platforms: **4/5** (proprietary)
- Make → Other platforms: **4/5** (proprietary)
- n8n → Other platforms: **3/5** (can export workflows)
- Custom Code → Other platforms: **2/5** (just APIs)

Recommendation: If you're uncertain about scale, start with n8n or ensure you can hybrid/migrate

---

6. Migration Considerations

Migrating FROM Power Automate

Challenge: Power Automate uses proprietary formats and connectors

Strategy:

1. **Document workflows** (screenshots, descriptions)
2. **Identify dependencies** (what triggers what)
3. **Rebuild in target platform** (no automated conversion)
4. **Run parallel** for 30 days (verify parity)
5. **Cutover** (disable old flows)

Effort Estimate:

- Simple flows: 2-4 hours each
- Complex flows: 8-16 hours each

Cost: For 50 flows averaging medium complexity:

- 50 flows × 6 hours × $150/hour = $45,000

Migrating TO Power Automate

When it makes sense:

- Acquired by Microsoft-centric company
- New compliance requirements
- Need enterprise support
- Budget less constrained

Strategy: Similar to above - manual rebuild required

Migrating BETWEEN n8n and Custom Code

Easier because:

- Both use standard HTTP/API calls
- Workflows are portable concepts
- Can reuse authentication logic

n8n → Custom Code:

- Export workflow JSON
- Convert to TypeScript/Python
- Reimplement logic (more control)

Custom Code → n8n:

- Map functions to n8n nodes
- Configure HTTP requests nodes
- Test thoroughly

---

7. Hybrid Architecture Patterns

Pattern 1: Power Automate + Azure Functions

Use Case: Simple workflows in Power Automate, complex logic in Azure Functions

```
Power BI Alert
  ↓
Power Automate (orchestration)
  ↓
Azure Function (complex processing)
  ↓
Power Automate (notifications)
```

Example:

```yaml
# Power Automate Flow
When Power BI alert triggered

# Call Azure Function for complex logic
HTTP
  Method: POST
  URI: https://yourfunction.azurewebsites.net/api/ProcessAlert
  Body: @{triggerBody()}

# Parse result
Parse JSON
  Content: @{body('HTTP')}

# Send notification based on result
Condition: @{body('Parse_JSON')?['severity']} = "high"
  THEN
    Send Teams message: High priority alert
  ELSE
    Send email: Standard alert
```

Benefits:

- Simple things stay simple (Power Automate)
- Complex things possible (Azure Functions)
- Best of both worlds

Drawbacks:

- Need to maintain two systems

- Authentication between systems
- Debugging across platforms

Pattern 2: n8n + Power Automate

Use Case: Use n8n for high-volume processing, Power Automate for Microsoft 365 integrations

```
n8n (data processing & business logic)
  ↓ (webhook)
Power Automate (Microsoft 365 actions)
```

Example:

javascript

```
// n8n workflow
{
 "nodes": [
  {
   "name": "Process Data",
   "type": "n8n-nodes-base.function",
   "parameters": {
    "functionCode": "// Process 1000s of records efficiently\nreturn items.map(item => ({...item, processed: true}));"
   }
  },
  {
   "name": "Trigger Power Automate",
   "type": "n8n-nodes-base.httpRequest",
   "parameters": {
    "url": "https://prod-123.eastus.logic.azure.com:443/workflows/abc/triggers/manual/paths/invoke",
    "method": "POST",
    "body": "={{$json}}"
   }
  }
 ]
}
```

Benefits:

- Cost-effective high-volume processing (n8n)
- Native Microsoft integration (Power Automate)

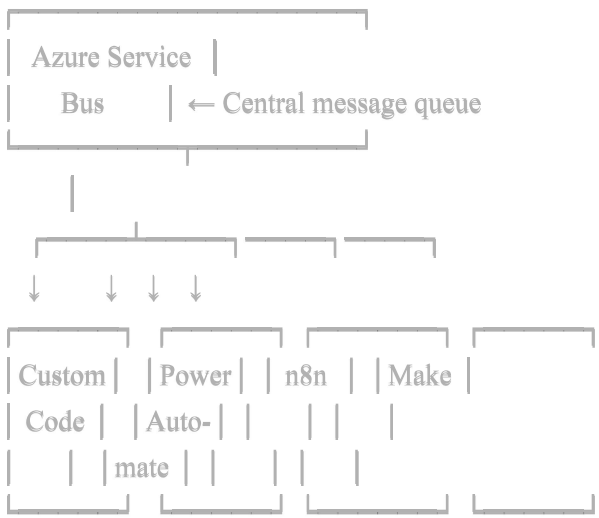
- Reduced Power Automate execution costs

Pattern 3: Multi-Platform Based on Team

Use Case: Different teams use different platforms based on their needs

```
IT Team: Custom Code (Azure Functions)
  ↓
Business Analysts: Power Automate
  ↓
Data Team: n8n
  ↓
Shared: Azure Service Bus (integration point)
```

Architecture:



Benefits:

- Teams use tools they're comfortable with
- Decoupled systems
- Easy to add/remove platforms

Drawbacks:

- Complex governance
- Multiple skill sets needed

- Harder to maintain standards

---

8. Implementation Examples

Example 1: Report Distribution Automation

Requirement: Every Monday, export Power BI report as PDF and email to 100 recipients based on their region.

Power Automate Implementation

yaml

```
Trigger: Recurrence
  Day: Monday
  Time: 8 AM

# Get recipient list
Get items (SharePoint)
  List: Report Recipients

# For each region
Apply to each unique region:

  # Export filtered report
  HTTP: Export Power BI Report
    Method: POST
    URI: https://api.powerbi.com/v1.0/myorg/groups/{workspaceId}/reports/{reportId}/ExportTo
    Body: {
      "format": "PDF",
      "powerBIReportConfiguration": {
        "reportLevelFilters": [{
          "filter": "Region eq '@{currentRegion}'"
        }]
      }
    }

  # Wait for export
  [Poll for completion]

  # Download file
  [Download PDF]

  # Get recipients for this region
  Filter array
    From: @{recipients}
    Where: Region = @{currentRegion}

  # Send emails
  Send email (Outlook)
    To: @{join(map(filtered, 'Email'), ';')}
    Subject: Weekly Report - @{currentRegion}
    Attachments: [PDF]
```

Time to implement: 2-3 hours

Monthly cost: $15 (Power Automate Premium)

n8n Implementation

```json
```

```
{
 "nodes": [
  {
   "name": "Schedule",
   "type": "n8n-nodes-base.cron",
   "parameters": {
    "triggerTimes": {
     "item": [{"mode": "everyWeek", "weekday": 1, "hour": 8}]
    }
   }
  },
  {
   "name": "Get Recipients",
   "type": "n8n-nodes-base.httpRequest",
   "parameters": {
    "url": "https://your-api.com/recipients"
   }
  },
  {
   "name": "Get Unique Regions",
   "type": "n8n-nodes-base.function",
   "parameters": {
    "functionCode": "const regions = [...new Set(items.map(i => i.json.region))];\nreturn regions.map(r => ({json: {region: r
   }
  },
  {
   "name": "Export Power BI Report",
   "type": "n8n-nodes-base.httpRequest",
   "parameters": {
    "url": "https://api.powerbi.com/v1.0/myorg/groups/{{$env.WORKSPACE_ID}}/reports/{{$env.REPORT_ID}}/ExportT
    "method": "POST",
    "authentication": "oAuth2",
    "sendBody": true,
    "bodyParameters": {
     "format": "PDF",
     "powerBIReportConfiguration": {
      "reportLevelFilters": [{
       "filter": "Region eq '{{$json.region}}'"
      }]
     }
    }
   }
  },
```

```
    {
     "name": "Poll Export Status",
     "type": "n8n-nodes-base.httpRequest",
     "parameters": {
      "url": "={{$node['Export Power BI Report'].json.exportUri}}",
      "method": "GET"
     }
    },
    {
     "name": "Download PDF",
     "type": "n8n-nodes-base.httpRequest",
     "parameters": {
      "url": "={{$node['Poll Export Status'].json.resourceLocation}}",
      "method": "GET",
      "responseFormat": "file"
     }
    },
    {
     "name": "Filter Recipients",
     "type": "n8n-nodes-base.function",
     "parameters": {
      "functionCode": "const region = $node['Get Unique Regions'].json.region;\nconst recipients = $node['Get Recipients'].js
     }
    },
    {
     "name": "Send Email",
     "type": "n8n-nodes-base.emailSend",
     "parameters": {
      "to": "={{$json.email}}",
      "subject": "Weekly Report - {{$node['Get Unique Regions'].json.region}}",
      "attachments": "={{$node['Download PDF'].binary}}"
     }
    }
   ]
}
```

Time to implement: 4-6 hours

Monthly cost: $0 (self-hosted) or $50 (n8n Cloud)

Example 2: Real-Time Alert with Escalation

Requirement: When Power BI alert fires, send Teams message. If not acknowledged in 10 minutes, escalate to manager.

Power Automate Implementation

```yaml
When Power BI alert triggered

# Send adaptive card to team
Post adaptive card (Teams)
  Channel: Operations
  Card:
    Title: Alert - @{alertName}
    Message: @{alertMessage}
    Actions:
      - Acknowledge [button]

# Wait for acknowledgment
Wait for approval
  Timeout: 10 minutes

# If not acknowledged
Condition: Request = Timeout
  THEN
    # Escalate to manager
    Send email
      To: manager@company.com
      Subject: ESCALATION - @{alertName}
      Priority: High
```

Time to implement: 1 hour
Monthly cost: $15

Custom Code Implementation

```typescript
```

```
// Azure Function
import { AzureFunction, Context, HttpRequest } from "@azure/functions";
import { TeamsFx } from "@microsoft/teamsfx";

const httpTrigger: AzureFunction = async (
  context: Context,
  req: HttpRequest
): Promise<void> => {
  const alertData = req.body;

  // Send Teams message
  const teamsfx = new TeamsFx();
  const messageId = await teamsfx.sendAdaptiveCard(
    process.env.TEAMS_CHANNEL_ID,
    {
      type: "AdaptiveCard",
      body: [
        {
          type: "TextBlock",
          text: `Alert: ${alertData.alertName}`,
          weight: "bolder",
        },
        { type: "TextBlock", text: alertData.message },
      ],
      actions: [
        {
          type: "Action.Submit",
          title: "Acknowledge",
          data: { action: "acknowledge", messageId: "PLACEHOLDER" },
        },
      ],
    }
  );

  // Schedule escalation check
  await scheduleEscalation(alertData, messageId);

  context.res = { status: 200, body: "Alert sent" };
};

async function scheduleEscalation(alertData: any, messageId: string) {
  // Use Azure Durable Functions or Queue with delay
  // Check acknowledgment after 10 minutes
```

```
  // If not acknowledged, send escalation email
}
```

Time to implement: 4 hours (includes testing)

Monthly cost: $5

---

9. Vendor Analysis

Microsoft Power Automate

Company Stability: ★★★★★

Microsoft is not going anywhere. Platform is strategic to Microsoft 365.

Product Maturity: ★★★★

Mature product with regular updates. Occasionally breaking changes.

Support Quality: ★★★★★

Enterprise support available. Large community. Microsoft Learn resources.

Lock-in Risk: ★★

High lock-in. Migration is manual and expensive.

Best For:

- Microsoft-first organizations
- Enterprise compliance needs
- Teams with limited coding skills

---

n8n

Company Stability: ★★★

Smaller company but growing. Fair-code model provides some protection.

Product Maturity: ★★★★

Rapidly maturing. Active development. Some rough edges.

Support Quality: ★★★

Community support is good. Enterprise support available with license.

Lock-in Risk: ★★★★

Low lock-in. Can export workflows. Self-hosting option.

Best For:

- Cost-conscious organizations
- High-volume automation
- Teams with DevOps capabilities

Make

Company Stability: ★★★★☆

Established company with strong backing. Consistent track record.

Product Maturity: ★★★★★

Very polished product. Extensive connector library.

Support Quality: ★★★★☆

Good support. Extensive documentation. Active community.

Lock-in Risk: ★★☆☆☆

High lock-in. Proprietary format. Manual migration required.

Best For:

- Quick implementation needs
- Multi-SaaS integrations
- Teams wanting simplicity

---

10. Recommendations by Organization Type

Small Business (<50 employees)

Recommended: Make or n8n Cloud

Why:

- Low cost
- Quick to implement
- No infrastructure burden
- Easy to use

Start with Make if:

- Team is non-technical
- Need lots of SaaS integrations
- Want polish over customization

Start with n8n Cloud if:

- Budget is tight
- Team has some technical skills
- Want flexibility for growth

Mid-Market (50-500 employees)

Recommended: Power Automate OR n8n Self-Hosted

Choose Power Automate if:

- Already using Microsoft 365
- Limited IT staff
- Compliance certifications needed
- Budget allows ($20-30K/year)

Choose n8n Self-Hosted if:

- Have DevOps team
- High automation volume
- Want cost control
- Need customization

Enterprise (500+ employees)

Recommended: Hybrid Approach

Architecture:

- **Power Automate:** User-facing workflows, Microsoft 365 integration

- **Custom Code:** High-performance, critical systems
- **n8n Self-Hosted:** Middle-tier automation, cost optimization

Why Hybrid:

- Leverage strengths of each platform
- Reduce costs where possible
- Maintain flexibility
- Meet diverse team needs

Healthcare Organizations

Recommended: Power Automate OR n8n Self-Hosted

Power Automate if:

- Need Microsoft BAA
- Want vendor compliance support
- Have Epic/Cerner + Microsoft ecosystem

n8n Self-Hosted if:

- Data sovereignty critical
- Strong internal compliance team
- Budget constraints
- Full control required

NOT Recommended: Cloud platforms (Make, Zapier) due to PHI flowing through third-party servers

---

Startups (Pre-Product Market Fit)

Recommended: Make

Why:

- Move fast
- Low upfront cost

- No infrastructure
- Can change later

Avoid:

- Custom code (too slow)
- Self-hosted anything (overhead)
- Long-term planning (you'll change direction)

Conclusion

There is no universal "best" platform. The right choice depends on:

1. **Your technical capabilities**
2. **Your budget and scale**
3. **Your compliance requirements**
4. **Your Microsoft ecosystem investment**
5. **Your timeline**

Decision Summary:

If you are...	Choose...
Microsoft-first organization	Power Automate
Budget-constrained	n8n Cloud or Make
High-volume automation	n8n Self-Hosted
Non-technical team	Power Automate or Make
Need full control	Custom Code or n8n Self-Hosted
Healthcare/Compliance	Power Automate or n8n Self-Hosted
Startup	Make
Enterprise	Hybrid approach

Getting Started:

1. **Assess your requirements** using the framework in Section 5
2. **Calculate your TCO** using scenarios in Section 3

3. **Prototype with 2-3 platforms** (pick most likely candidates)
4. **Evaluate over 30 days**
5. **Make informed decision**

Need Help?

MBIC helps organizations choose and implement the right automation stack. We're platform-agnostic and will recommend what's best for YOUR situation.

Services:

- Platform selection consulting
- TCO analysis
- Proof of concept implementations
- Migration planning
- Hybrid architecture design

Contact: Email: hello@mbic.us
Website: mbic.us
