The SME Digital Operating System Starter Kit

A practical planning kit for founders and operators who want fewer disconnected tools, safer AI workflows, and a clearer operating model.

Plan before tools
Map the work before choosing another platform.

Govern AI
Add review, cost visibility, source checks, and rollback.

Ship in 90 days
Stabilize, connect, then improve with a realistic roadmap.

Created by Hossam Afifi

For SMEs, consultants, agencies, and operators

Start with the operating model

Most SMEs do not need another dashboard first. They need a clearer system for how work flows.

01

This kit is for SME founders, operators, consultants, and service teams who have outgrown the “website plus scattered tools” stage.

You can use it whether you build internally, work with an agency, assemble a SaaS stack, or assemble a custom open stack using robust headless tools like n8n and Supabase.

> If our business had a digital operating system, what would need to run inside it?

The goal is not to make your business more complex. The goal is to make the repeated work easier to see, improve, and govern.

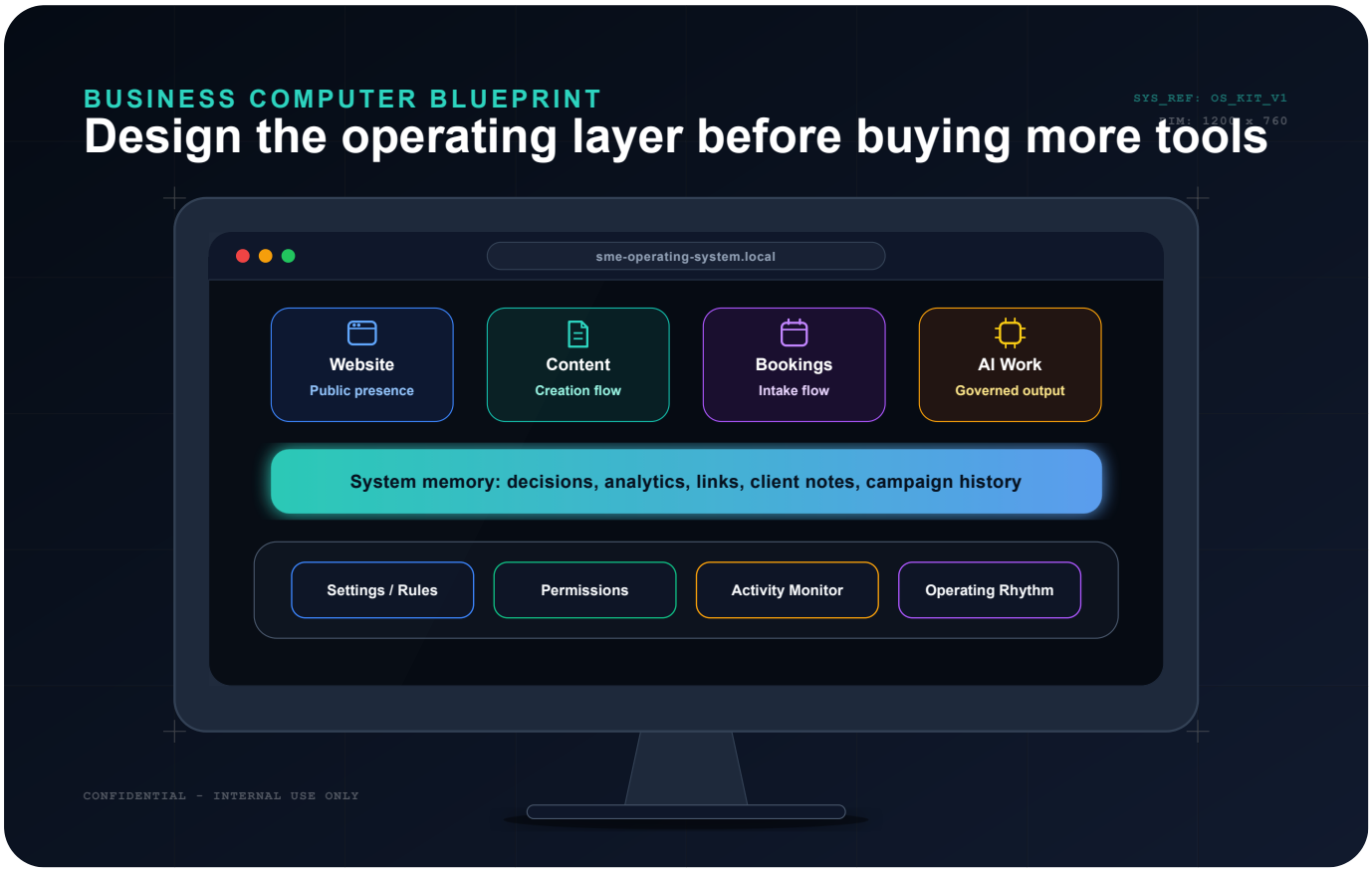
This kit helps you decide

- Which tools should stay, integrate, or disappear.
- Which AI workflows are useful and safe enough to adopt.
- Which business systems need ownership and review.
- What should be improved in the next 90 days.

The Business Computer Blueprint

A simple model for seeing your website, content, bookings, AI, and operations as one system.

02



Layer	Business equivalent	Planning question
Desktop	The place your team works from	Where does daily digital work begin?
Apps	Website, content, booking, CRM, newsletter, analytics	Which jobs need dedicated workflows?
Memory	Decisions, links, campaign history, client notes	What should the system remember?
Settings	GDPR, brand rules, roles, AI limits	What should be controlled centrally?
Activity monitor	Analytics, conversions, tasks, audit logs	What signals trigger action?

Worksheet: your current business computer

Complete this before discussing vendors or subscriptions.

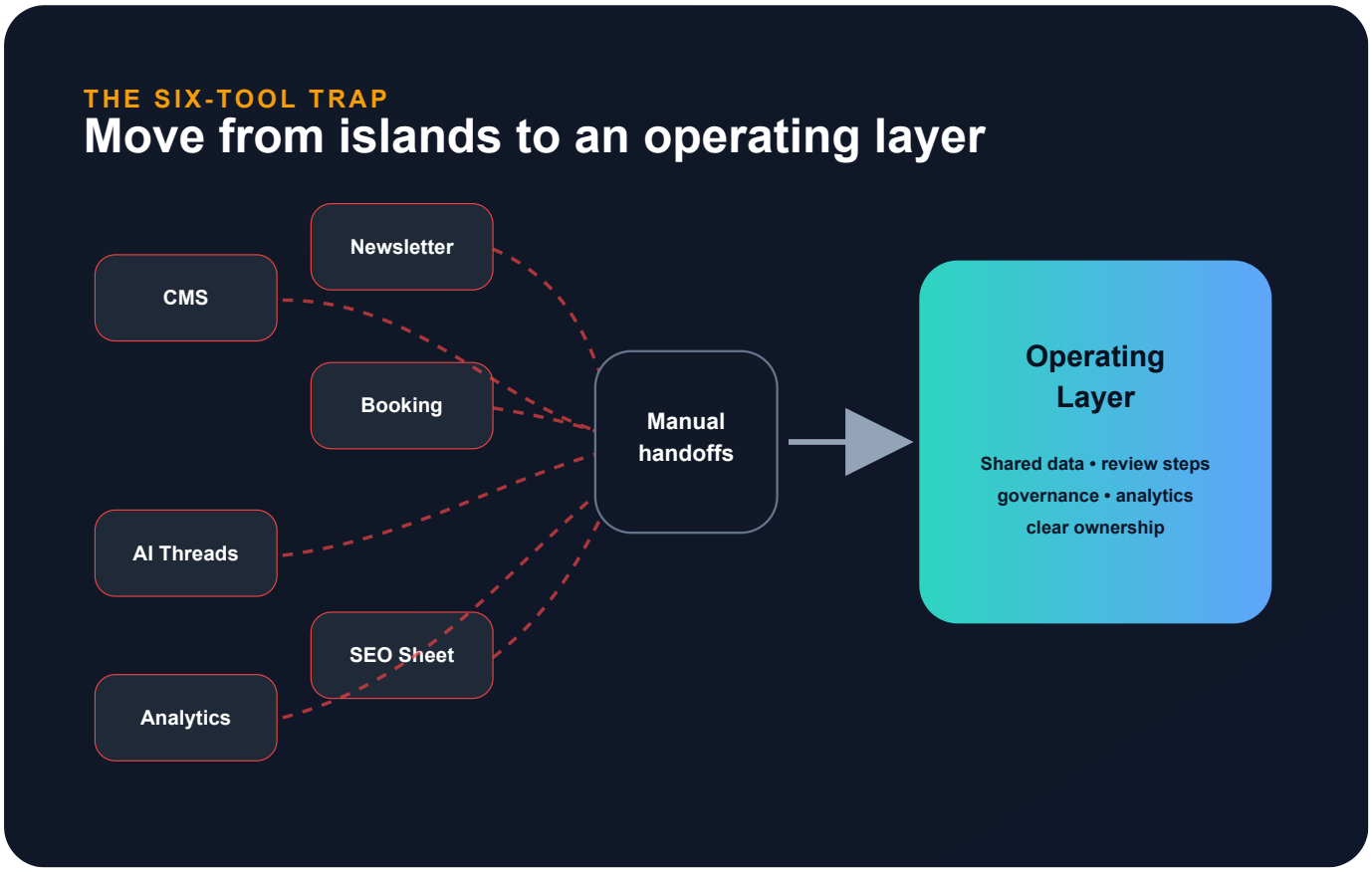
03

Layer	Current tool/process	Pain 1–5	What needs to improve?
Website / public presence			
Content creation			
SEO / visibility			
Newsletter / follow-up			
Booking / intake			
Client operations			
AI usage			
GDPR / governance			
Analytics / decisions			

The Six-Tool Trap Escape Map

The problem is not the number of tools. The problem is unmanaged seams between tools.

04



Symptoms

- Copy-paste operations.
- Invisible decisions.
- No feedback loop.
- Unclear ownership.

Decision rule

Replace or integrate a tool when it duplicates data, creates weekly manual work, blocks reporting, adds compliance risk, or harms the customer experience.

Worksheet: tool-friction inventory

Map handoffs before deciding what to keep, replace, or integrate.

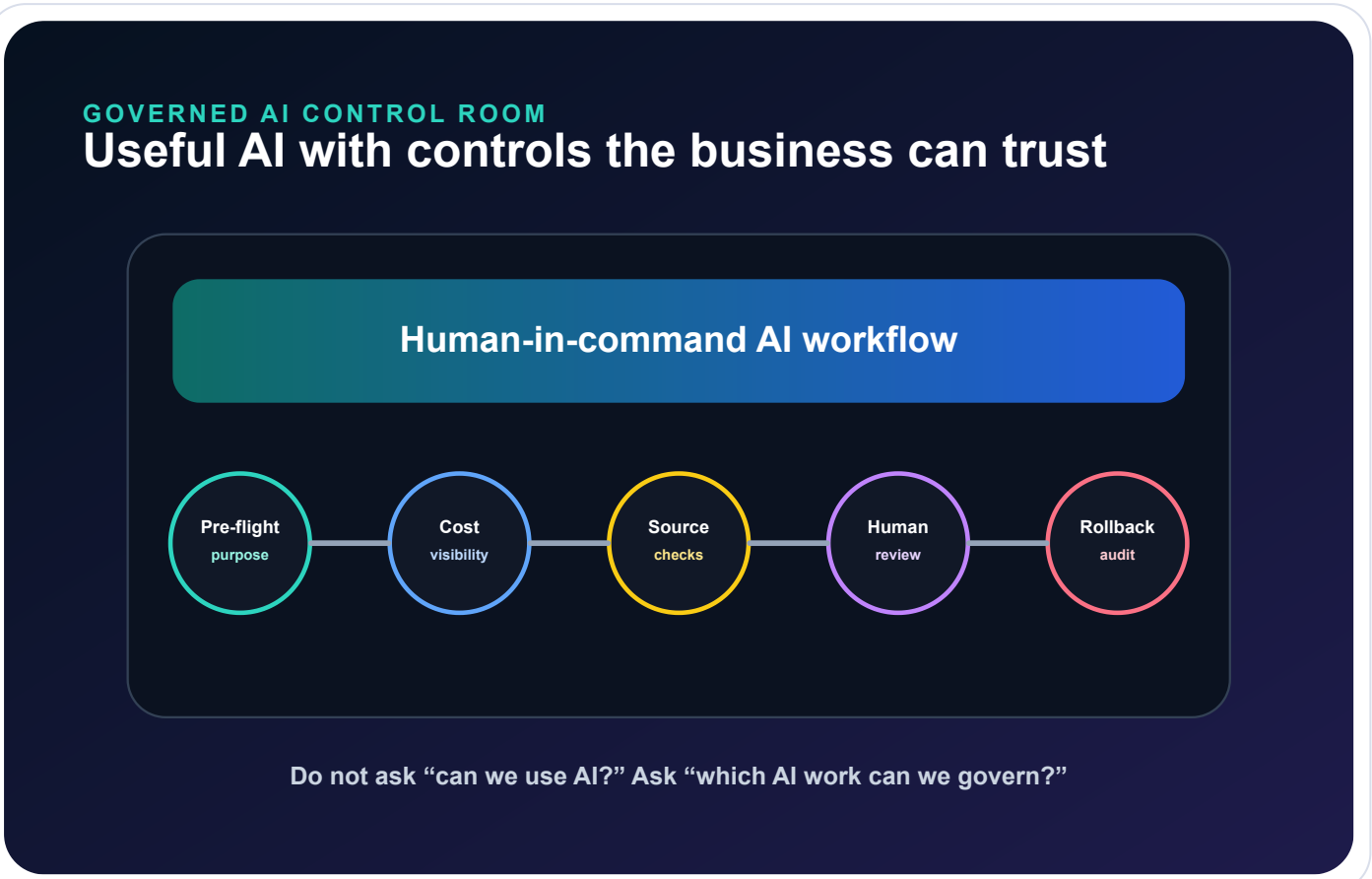
05

Tool	Main job	Data enters	Data leaves	Manual handoffs	Decision
CMS / website					
Newsletter					
Booking					
CRM / client list					
SEO tool / sheet					
AI tool					
Analytics					

The Governed AI Control Room

AI maturity means useful AI work with enough control that the business can trust the output.

06



Control	Why it matters	Practical test
Pre-flight purpose	Stops random AI usage	Can the user explain the job before generating?
Cost visibility	Prevents surprise spend	Can an owner see what AI activity costs?
Source awareness	Reduces hallucination risk	Are factual claims checked against credible sources?
Human review	Protects quality and brand	Can a human approve before publication?
Rollback / audit	Creates accountability	Can the business see what changed and reverse it?

Worksheet: AI workflow audit

07

Start with workflows where errors are visible before they reach the customer.

AI use case	Business value	Risk level	Reviewer	Source check?	Rollback?	Priority
Blog drafting						
SEO suggestions						
Customer support draft						
Newsletter draft						
Legal/compliance text						
Internal process docs						

The Open-Stack Architecture

Combine Supabase as your relational memory and n8n as your logic coordinator.

08

SMEs often fall into the trap of purchasing heavy, monolithic ERPs or letting SaaS tools operate as disconnected islands. The modern alternative is a modular **Open Stack**.

By pairing **Supabase** (PostgreSQL database) with **n8n** (workflow engine), you create a highly secure, private digital core that mirrors your business rules.

> Your database is your single source of truth; your workflows are the pipes connecting it to the world.

This separation of data and logic means you can replace frontend interfaces or downstream APIs without breaking your historical business memory.

Core Stack Components

1. Supabase (The Memory)
Manages user roles, row-level security (RLS), document uploads, and structured customer data.

2. n8n (The Logic Processor)
Executes API calls, orchestrates AI prompt pipelines, sends transactional emails, and schedules cron tasks.

3. External APIs (The Touchpoints)
Stripe for payments, Resend for email, Tavily for web research, and Webflow or Astro for the frontend.

DATAFLOW ARCHITECTURE

USER / FRONTEND
Fills a booking form, writes a blog draft, or uploads a document receipt.

→

SUPABASE (DATABASE)
Stores row and triggers immediate webhook via the `pg_net` PostgreSQL extension.

→

n8n ENGINE (LOGIC)
Executes AI check agent, calls external APIs, and saves approved edits back to DB.

Real-Time Event Triggers

Avoid interval polling. Configure Supabase to notify n8n instantly on database changes.

09

The Anti-Pattern: Interval Polling

Most basic automations check the database on a timer (e.g., “every 15 minutes”). This is slow, wasteful, and exhausts workflow plan execution limits with zero-value checks.

The Expert Technique: Event-Driven Push

By enabling the `pg_net` extension in Supabase, you can set up a database trigger. The moment a row is inserted or modified, PostgreSQL makes an asynchronous HTTP POST request to your n8n webhook.

- **Sub-second latency:** Actions trigger instantly when records change.
- **Zero wasted polling:** n8n only runs when actual changes occur.
- **Resilient Architecture:** Calls run asynchronously; DB operations succeed even if n8n is offline.

Supabase SQL: Trigger Setup

```
create extension if not exists pg_net;

create or replace function send_to_n8n()
returns trigger as $$
begin
  perform net.http_post(
    url := 'https://n8n.domain.com/webhook',
    headers := '{"Content-Type": "application/json"}',
    body := json_build_object(
      'event', TG_OP, 'record', row_to_json(NEW)
    )::text
  );
  return NEW;
end;
$$ language plpgsql security definer;

create trigger on_booking_insert
  after insert on bookings
  for each row execute function send_to_n8n();
```

💡 **Pro-Tip: Multi-Environment Routing**

In your trigger function, you can inspect database variables or schemas to route webhooks to your n8n `/active` production webhook endpoint or your `/test` staging endpoint automatically, keeping development isolated from production database records.

Human-in-the-Loop Gates

How to safely govern AI workflows with a secure approval routing system.

10

The Governance Challenge

Allowing AI agents to contact customers or publish content directly is highly risky. Practical maturity requires a lightweight verification layer where operators inspect generated draft content first.

The One-Click Inbox Workflow

Combining Supabase states and n8n secure webhook responses enables a simple review system that reviewers can operate in one tap on mobile:

1. **Stage:** AI builds draft and inserts it into a Supabase `review_queue` table with status `'pending'` and a random `approval_token` (UUID).
2. **Notify:** n8n sends email alert containing draft and links: `/approve?token=UUID` and `/reject?token=UUID`
3. **Release:** Clicking either link contacts the n8n webhook, which verifies the token, changes status to `'approved'`, and releases the task downstream.

Approval Queue Database Schema

```
create table review_queue (
  id uuid default gen_random_uuid() primary key,
  content_type text not null,
  raw_draft text not null,
  status text default 'pending' check (status in
    ('pending', 'approved', 'rejected')),
  approval_token uuid default gen_random_uuid(),
  reviewer_email text,
  created_at timestamptz default now()
);

alter table review_queue enable row level security;

create policy "manager_select" on review_queue
  for select to manager using (true);
```

Security Tip: Token Expiration and Clean Up

Configure your n8n validation node to reject tokens that are older than 24 hours or have already been processed. Once approved, overwrite the `approval_token` in Supabase to null to ensure the link can never be clicked or verified again.

Centralized AI Cost Ledger

How to track every token, prompt, and cent to prevent runaway API billing.

The Risk: API Billing Spikes

Unmonitored developer loops, recursive parsing agents, or bloated processing prompts can rack up hundreds of dollars in API fees in hours without triggering external billing alerts.

Continuous Usage Logging

By capturing metadata from n8n LLM nodes and writing to a structured ledger in Supabase, you build an instant cost audit dashboard.

This allows you to compute exact costs per execution run, compare model efficiency (e.g. GPT-4o vs. lightweight models), and write safety triggers that shut down workflows if costs spike.

SQL Ledger Table Schema

```
create table ai_cost_ledger (
  id uuid default gen_random_uuid() primary key,
  workflow_id text not null,
  model_used text not null,
  prompt_tokens int not null,
  completion_tokens int not null,
  usd_cost numeric(8,6) not null,
  run_timestamp timestamptz default now()
);

create or replace function check_runaway_cost()
returns trigger as $$
begin
  if NEW.usd_cost > 0.50 then
    perform net.http_post(
      'https://hooks.slack.com/services/...',
      '{"Content-Type": "application/json"}',
      json_build_object('text', '🚨 High AI
Cost!')::text
    );
  end if;
  return NEW;
end;
$$ language plpgsql;
```

💡 **Formula: Compute Cost Dynamically in n8n**

Calculate the exact run cost in a Code node using token numbers returned by LLMs before logging to Supabase:

`const cost = (promptTokens * 0.000005) + (completionTokens * 0.000015);`

This lets you log costs in real-time, independent of supplier invoicing delays.

The Digital Systems Planning Canvas

Use this before a build sprint, agency brief, SaaS migration, or internal automation project.

12

Primary 90-day outcome

In the next 90 days, this system should improve:

Current friction

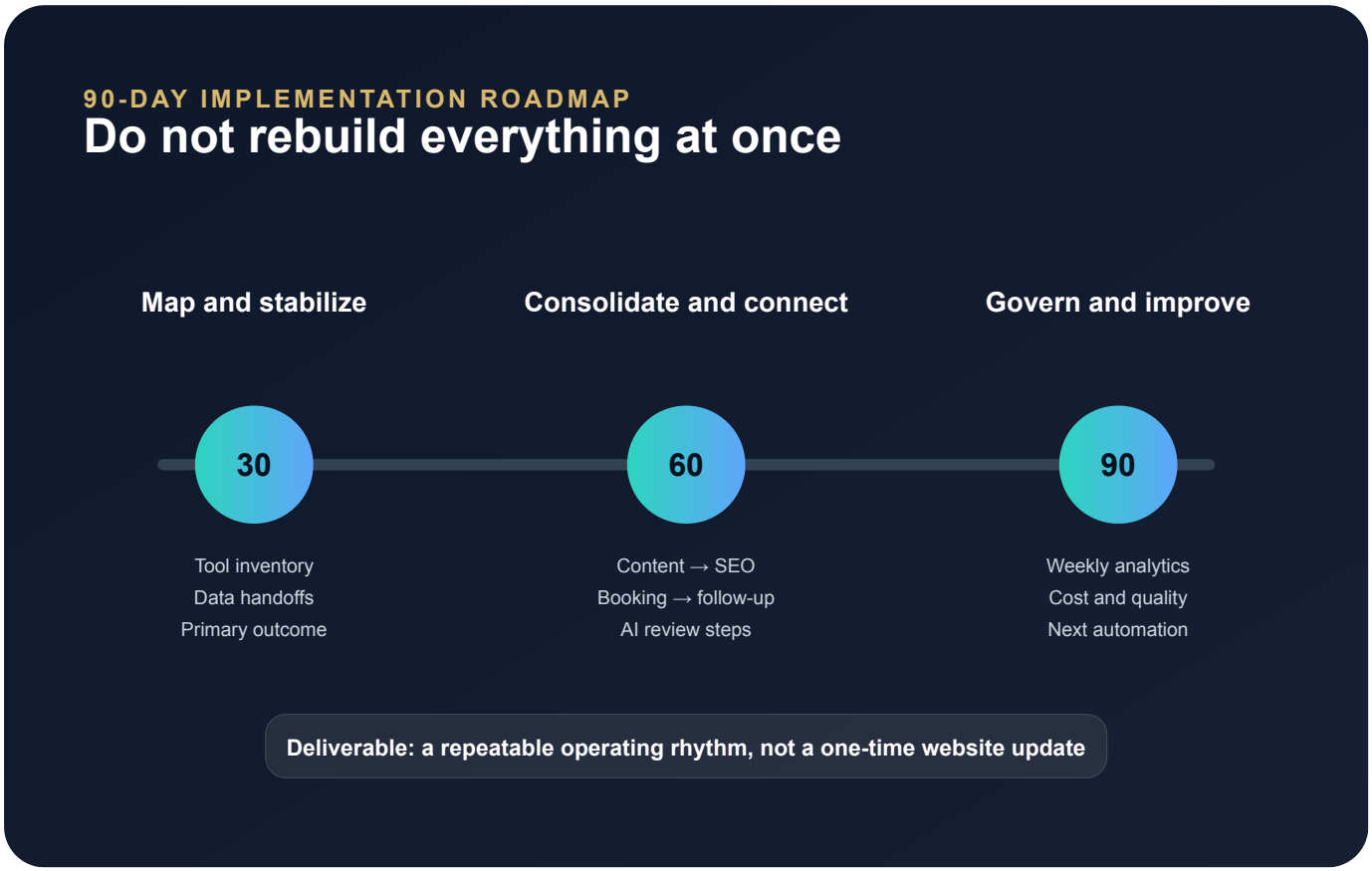
Where does work slow down today?

Success signals

- [ ] Leads per month
- [ ] Manual hours saved
- [ ] Organic impressions
- [ ] Booking conversion rate

The 90-Day Implementation Roadmap

Map and stabilize, then connect, then govern and improve.



What to automate first

- Frequent.
- Low-risk.
- Rule-based.
- Easy to review.
- Connected to a measurable outcome.

Delay these

- Autonomous customer replies.
- Legal advice generation.
- Financial decisions.
- Public posting without approval.

Final checklist

If you cannot answer these yet, the next step is not another tool. The next step is system planning.

- [ ] What business outcome should improve in 90 days?
- [ ] Which weekly manual work should disappear?
- [ ] Which customer or client experience should become smoother?
- [ ] Which data should live in one place?
- [ ] Which AI workflows are allowed?
- [ ] Who reviews AI output?
- [ ] What needs an audit trail?
- [ ] What should happen when content is published?
- [ ] What should happen after someone books a call?
- [ ] What will we measure every week?

Hosting your stack for under $10/month

For most SMEs, you don't need expensive enterprise plans. You can spin up Supabase (the free tier handles up to 500,000 DB records and 50,000 monthly active users) and run your n8n workflow engine on hosting platforms like Elestio, Railway, or Fly.io for $5–$10/month. This gives you complete ownership of your database, security, and logic, with zero vendor lock-in.