



Decide what you actually need — and start at the lowest rung.

You rarely need to “build software.” Most work — especially with data — happens in a chat or a single Artifact. Read the need, pick the lowest rung that does the job, and only build when a real need forces it.



Do it in chat or an Artifact NO SETUP

Most analysis, drafts, summaries and one-off interactive tools. Upload a file or paste, and ask.
Start here.



Connect a tool A FEW CLICKS

Reach your real data and systems with a connector, or plug in a ready-made service. **No code.**



Build or commission software A PROJECT

Only when it must serve many users, keep data between sessions, integrate deeply, or **run unattended.**

WHEN YOU NEED...

THE STEP

REACH FOR — SIMPLEST FIRST

◆ DATA — GET IT & MAKE IT USABLE

My data is in a file (Excel, CSV, PDF, doc)	→ Bring it in	→ Upload it straight to Claude — no tool needed
It lives in a system (email, Drive, SharePoint, Slack)	→ Connect the source	→ Connectors (M365 · Google · Slack) · or export to CSV
It sits in a company database or app	→ Get an extract	→ A CSV export · a read-only connector · an IT-provided view
It's messy, inconsistent, half-typed	→ Clean & tidy	→ Just ask Claude to clean it · Excel / Sheets
I need to combine or reshape several sources	→ Combine & transform	→ Claude (it can run the calculations) · Excel Power Query

◆ DATA — MAKE SENSE OF IT · VISUALS & INSIGHTS

Find trends, outliers, segments, totals	→ Analyse	→ Ask Claude to analyse it · pivot tables
See it — charts, comparisons, breakdowns	→ Visualise	→ Claude Artifact (live charts) · Excel charts
A dashboard others refresh & share live	→ BI / dashboard	→ Power BI · Looker Studio · Tableau
What does it mean? what should we do?	→ Insight & recommendation	→ Ask Claude to interpret & recommend
Turn it into something I can send	→ Report / deck	→ Claude → Word · PDF · PowerPoint
Refresh it on a schedule, hands-off	→ Automate the refresh	→ A scheduled agent / job — a build (rung 3)



When you build for others, name the component first.

Software that other people rely on is assembled from a handful of standard parts. Decide which parts the job genuinely needs — then pick a proven tool for each. Most jobs need only a few.

WHEN YOU NEED...	THE COMPONENT	GOOD TOOLS
◆ BUILD SOFTWARE OTHERS RELY ON — USUALLY A PROJECT (RUNG 3)		
More than one user, each with a login	→ Authentication	→ Supabase Auth · Clerk · SSO
Staff / company single sign-on + roles	→ SSO + access roles	→ Microsoft Entra ID · Okta · Google
Data that must persist between sessions	→ Database	→ Postgres — Supabase · Neon
Files / images people upload & keep	→ File storage	→ Supabase Storage · AWS S3 · Cloudflare R2
Put it online for others to use	→ Hosting	→ Netlify · Vercel
Send email automatically (reminders, digests)	→ Email	→ Postmark · Resend
Take payments / run subscriptions	→ Payments	→ Stripe · Paddle
AI features built into the product	→ LLM API	→ Anthropic API (Claude)
Know when it breaks & who uses it	→ Monitoring & analytics	→ Sentry · PostHog
Track changes & roll back safely	→ Source control + CI/CD	→ GitHub (+ Actions)
◆ GOVERN — CUTS ACROSS EVERYTHING		
Personal or regulated data	→ Region & data agreements + access control	→ Pick provider region · least-privilege · audit logs
A risky or irreversible action	→ Human approval gate	→ A confirm step · never auto-run · treat fetched data as untrusted

THE ARCHITECT'S REFLEX

When a request lands, don't ask *"which tool?"* — ask *"what do I actually need to produce, and what's the lowest rung that gets me there?"* Most data work — getting it, cleaning it, analysing it, charting it, writing it up — is just **ask Claude**. Drop to rung 3 (build software) only when a real need forces it: many users · must persist · must run on its own. And put a human gate on anything irreversible.