

Projects & Anonymise Guide

Set up AI once, so every tender analysis is fast, safe, and comprehensive

A BMS Cyber-Sales Training guide · Practical Cyber × DACTA × Apexagen

The problem with a plain chat is you re-paste your reference material every time, and a big tender blows past what the model can hold. The fix is to give the AI a **standing base** — your Vault — and only hand it the tender on top.

Two paths. The trust boundary decides which.

	(Co-Work / your own Claude Code / Codex)	(ChatGPT / Claude / Gemini)
Data stays private?	Yes — your own instance	No — treat as public
Feed the raw client tender?	Yes	Never — anonymise first
How it holds the big context	Reads only the docs it needs, per clause	Load the Vault into a Project once
Best for	Lots of files, deepest output	Most people, most of the time

Path A · Consumer web UI — set up a Project

A **Project** (Claude *Projects*, ChatGPT *Projects*, Gemini *Gems*) is a chat workspace with **its own reference documents and standing instructions**. Set it up once; every tender chat inside it inherits the base.

1. **Create the Project** — call it “*BMS Tender Response*”.
2. **Upload your Vault as the Project’s reference docs** — buckets 1–3 from the *Tender-Prep Kit*: the product-capability matrix, your posture/anchor library, the CCoP clause map, the 7 Pillars. (*No live client data here — this is your own reusable material.*)
3. **Paste the standing Project instructions** (below) into the Project’s custom-instructions box.
4. **Per tender: anonymise first** (Path B), then start a new chat *inside the Project*, paste the de-identified tender, and ask it to triage + set a posture per clause — comparing against your uploaded Vault.

Why a Project beats re-pasting: the Vault is loaded once and persists; each chat starts already

knowing your products and postures, so you only spend the context window on *this* tender.

The standing Project instructions — copy this in

You are helping a Mitsubishi Electric BMS salesperson respond to building tenders. Use the reference documents in this Project as ground truth about our products, our standard postures (anchors), and the CCoP / 7-Pillars / IEC 62443 mappings.

For any tender I paste:

1. Triage every cybersecurity clause: Must-have / Nice-to-have / Risky red-herring.
2. Set a posture per clause: Comply / Not-comply (+ a better alternative) / Clarify / each with a one-line rationale that CITES the relevant reference doc.
3. Flag any clause that is indefensible to "Comply" to, contradicts itself, or is a drafting slip.

Rules:

- Ground every "Comply" in a product capability or an anchor from the reference docs. If it isn't supported there, say "Not supported in refs – [UNVERIFIED]", don't invent.
- Never use a forbidden phrasing (100% secure, CCoP-certified product, "mandatory" to non-CII client).
- You draft and compare; I set the final posture and sign it.

Always show your plan first and wait for my "go" before drafting the full response.

Path B · Anonymise — the safe bridge to any consumer tool

The trap to avoid: you cannot paste a raw, sensitive client tender into ChatGPT *to have it anonymised* — that already leaked it. Anonymisation has to happen **before** the consumer tool ever sees it. Two safe ways:

- **Have a private agent?** Paste the raw tender into your **Co-Work / own Claude Code** instance (your data, your control) and have it output the de-identified extract with the prompt below. Then take that clean extract to your consumer Project.
- **No private agent?** Anonymise **by hand** — find-and-replace the client name, authority, contract/reference number, project name and address with placeholders — before pasting anywhere.

The anonymise prompt — for use in a PRIVATE instance only

Here is a real client tender. Produce a DE-IDENTIFIED, exportable extract I can safely analyse in a consumer AI tool.

Do this:

- Replace the client/authority name with "[CLIENT]", the project name with "[PROJECT]" the contract/tender reference with "[REF]", and any address/site name with "[SITE]".
- Remove names of individuals, phone numbers, and emails.
- KEEP the clause numbers and the full clause text (they carry no identity).

- KEEP the sector/building type in generic form (e.g., "a healthcare + data-hall build because it drives whether CCoP applies.
Output ONLY the cleaned extract, clause by clause, ready to copy. Then list, separately, every substitution you made so I can verify nothing identifying slipped through.

Then **verify the substitution list by eye** before the extract leaves your control.

The per-tender flow (pin this)

1. **Anonymise** → clean, exportable extract (Path B).
2. **New chat inside your Project** → paste the extract.
3. **“Triage + set a posture per clause, using my reference docs.”**
4. **Verify every posture by hand** — especially every “Comply” and anything it sounds *too* sure about (the Eloquence Trap).
5. **Transcribe the verified postures into your Posture Ledger** on the client’s own file.

Speed is real; **verification is non-negotiable**. The AI does the intellectual labour; you keep the accountability labour — and you sign.

Pairs with the **Tender-Prep Kit** (what to load) and the practice tender at bms.ethanseow.com/practice-tender.html. Practical Cyber × DACTA × Apexagen.