

Security pack

IMS Creator, hosted at app.imscreator.com

Version 1.0, November 2026: the service on its own server in Australia

imscreator.com/security-pack · security@imscreator.com

Contents

1. Overview
2. Architecture and data flows
3. Your data
4. Claude and your documents
5. Identity and access
6. Application security
7. Hosting and operations
8. Incident response
9. Service providers
10. Shared responsibility
11. Common questions
12. ASVS level 2 self-assessment
13. Reporting a vulnerability

Overview

For the people who review IMS Creator for their organisation: IT security, privacy, risk and procurement. It describes the service as it runs today, in plain terms, with enough detail to answer a vendor security assessment.

Version 1.0, November 2026. It covers hosted IMS Creator at `app.imscreator.com` in its current configuration: the service runs on a dedicated server in Australia, owned and run by IMS Creator, behind Cloudflare. A version for the cloud configuration (Microsoft Azure) will be published before any customer is moved there, and you will be told.

In brief

- **Your company's information is kept apart.** Each company has its own databases and its own place in the file store. A test on every release signs in as one company and asks for every page with another company's names and ids, and must get nothing back.
- **Our administrators cannot read your content.** Platform administration shows companies, sites, numbers of people and the service's health, never a document or what it says.
- **An authenticator code is asked for before the password**, passwords are checked against known breaches, and every sign-in and change is in your own tamper-evident audit log.
- **Claude runs on your own Claude account**, under your own agreement with Anthropic, AWS or Google. Your key is encrypted with your company's own key and used only for your work. What is sent, and what the provider keeps, is set out in full in Claude and your documents.
- **Drawings are not stored**: they are processed in batches and deleted within 2 hours at most.
- **Encrypted everywhere**: in transit (TLS 1.2 or later), on disk (LUKS2), in backups (encrypted before they leave the server).
- **Backups** continuously archived (at most 15 minutes of changes at risk), kept off-site, restored as a test every quarter.
- **Incidents**: affected customers are told within 72 hours of our becoming aware.

What is in the pack

Section	What it covers
Architecture and data flows	The parts of the service, where each runs, and how data moves between them
Your data	What we keep, where, how it is encrypted, for how long, and how it is returned and deleted
Claude and your documents	What is sent to Claude, when, on whose account, and what Anthropic, AWS or Google keep
Identity and access	Signing in, roles, sessions, the audit log, and our own administrators' access
Application security	How the software is built, tested and kept up to date; plugins; the ASVS assessment
Hosting and operations	The server, the network, monitoring, patching, backups and recovery
Incident response	What we do when something goes wrong, and when you hear from us
Service providers	The companies whose services we use, and what each sees
Shared responsibility	What we look after, and what your company looks after

Section	What it covers
Common questions	Short answers to the questions security questionnaires ask most
ASVS level 2 self-assessment	The OWASP Application Security Verification Standard, requirement by requirement
Reporting a vulnerability	How to tell us about a security problem

The pack is also a single PDF, ims-creator-security-pack.pdf, to attach to your assessment.

What we do not claim

IMS Creator is a young service run by a small team. It has **no independent certification yet** (such as ISO/IEC 27001 or SOC 2). What it has: a design written for security from the start, an OWASP ASVS level 2 self-assessment, automated tests of company isolation on every release, and this pack. Before the service grows beyond its first customers, an independent penetration test will be done, then yearly, and its summary shared with customers. Where something here is planned rather than done, it says so.

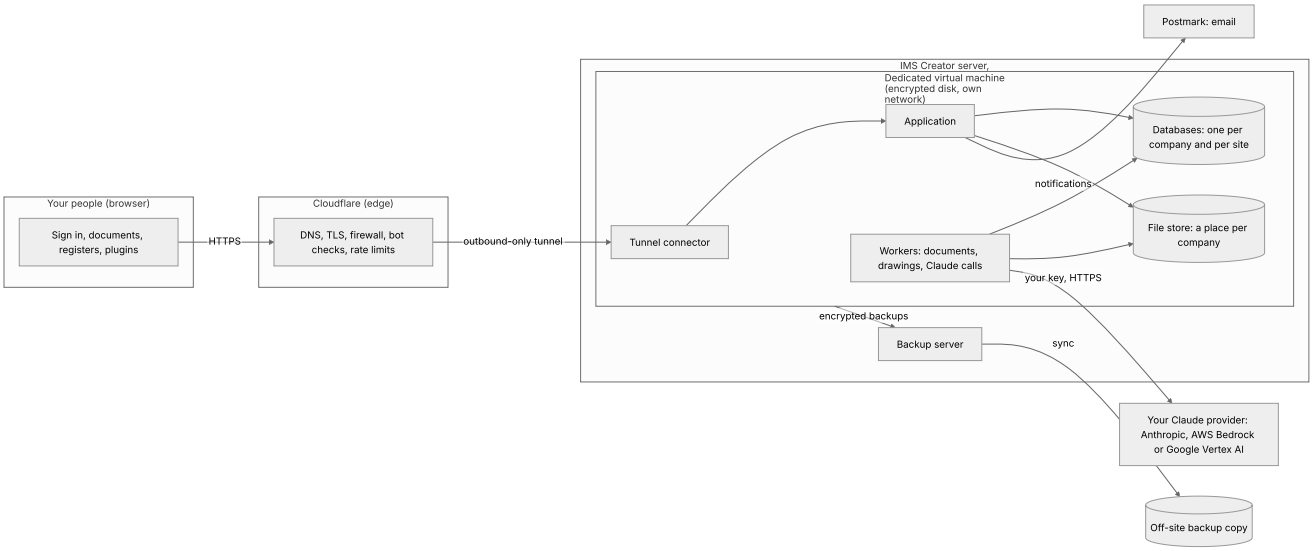
Questions

Write to security@imscreator.com. We answer security questionnaires, and can join a call with your security team.

Architecture and data flows

IMS Creator is a web application: people use it in a browser (current Chrome, Edge, Safari or Firefox), and nothing is installed on your computers.

The parts



- **Cloudflare** receives every request: it answers for `app.imscreator.com`, ends TLS, and applies its firewall, bot protection and rate limits. It passes requests to the server through a **tunnel the server opens outwards**: the server has no port open to the internet.
- **The application** serves the pages and the API. Every request is checked for a valid session, the person's company, site and role before anything is read, in one place for the whole application.
- **Workers** do the longer jobs: writing documents with Claude, reading drawings, building reports.
- **Databases** (PostgreSQL): a directory of companies and sign-ins, then **one database for each company and one for each of its sites**. No query spans two companies' databases.
- **File store** (S3-compatible, private): uploaded documents, templates, report images and thumbnails, each company in its own place.
- **Backups** go encrypted to a separate backup server, which keeps an off-site copy.

How information moves

Flow	From	To	Protection
Using the app	Your browser	Cloudflare, then the server	TLS 1.2 or later; HSTS; the tunnel is encrypted
Drafting a document	The server	Your Claude provider	TLS; your own credentials; see Claude and your documents
Reading drawings	A folder on your computer	The server, then back to your folder	TLS; files deleted from the server within 2 hours
Email (invitations, resets, notices)	The server	Postmark, then the recipient	TLS; links are single-use and expire

Flow	From	To	Protection
Backups	The server	The backup server, then off-site	Encrypted on the server before they leave it

Cloudflare ends TLS at its edge, so requests pass through Cloudflare's network decrypted in transit, as with any service behind a content delivery network. Responses from the API are not cached there.

Where it runs

Everything that holds your information runs in **Australia**: the server, its backup server and the off-site copy. Your Claude processing runs wherever your Claude provider and region are (see Claude and your documents).

The cloud configuration (Microsoft Azure, with a region per customer) will be described in its own version of this pack.

Your data

What we keep, and where

Information	Where it is kept	Encrypted	Kept for
Management system documents, registers, plans, approvals and their history	Your company's and your sites' databases	Disk (LUKS2)	While you subscribe
Uploaded documents (standards, templates, reports, register attachments)	Your company's place in the file store	Disk; sensitive files with your company's key	Until removed, or you leave
Drawing register: each drawing's fingerprint, name, title block, tags, links, summary, search text, small page images	Your site's database and file store	Disk	While you subscribe
The drawings themselves	**Only while a batch is processed**, on a temporary encrypted area not backed up	Disk	**Deleted when you collect the results, or after 2 hours at most**
Your Claude connection (API key, or cloud credentials)	Your company's or site's settings	**Your company's or site's own key**, then disk	Until replaced or removed
People: name, email, role, sign-in history	Your company's database (the email also in the directory, to know your company at sign-in)	Disk	While they have an account
Password hashes, authenticator secrets, recovery codes	Your company's database	Argon2id; authenticator secrets with your company's key; recovery codes hashed	While they have an account
Audit log of sign-ins and changes	Your company's database, hash-chained	Disk	While you subscribe; exportable at any time
Support requests and their replies	The platform's directory database	Disk	While you subscribe
The service's own logs (no document content)	The server	Disk	90 days
Backups of all the above	Backup server and an off-site copy	Encrypted before leaving the server	30 daily, 12 monthly

Everything above is held in **Australia**.

Kept apart from other companies

- Each company has its **own databases** (one for the company, one for each site) and its **own place in the file store**. The application chooses the database from the signed-in person's session, never from anything in the request, so a request cannot reach into another company.

- On every release, an automated test signs in as one company and asks for every page and every API call with another company's names and ids: the release is stopped if anything comes back.
- People belong to one company, and sign in with an address at one of the **email domains your company has approved and verified** (a DNS record, or checked by us by hand).

Encryption

- **In transit**: TLS 1.2 or later from the browser to Cloudflare, with HSTS; an encrypted tunnel from Cloudflare to the server; TLS to Claude providers and to Postmark.
- **At rest**: the server's data disk is encrypted (LUKS2). Secrets that are yours, such as your Claude credentials and your people's authenticator secrets, are also sealed with a **key of your company's own** (AES, through the Fernet scheme), which can be rotated.
- **Backups**: encrypted on the server, before they leave it, with a key kept off the server.

What our administrators can see

Platform administration (our console) shows: companies and their sites, how many people, subscription and billing details, support requests you send us, and the service's health (counts of queued and failed jobs, sign-in failures, sessions). It **never shows** a document, a register entry, a file name, a person's documents, or what a job is about, and it cannot open your sites. Our administrators sign in with an authenticator, and from the internet only through Cloudflare Access, as themselves.

Running the server, its administrators could in principle reach the databases directly. That access is limited to the owner, through a private network with keys only, and is used only to operate the service: to restore a backup at your request, or to deal with an incident.

Your data, returned and deleted

- **At any time**: documents download as Word, registers as Excel, and your company admins can export the audit log.
- **On request, and when you leave**: a full export of your company's databases (in PostgreSQL's open format), its files and its settings, delivered encrypted.
- **When you leave**: after the period agreed with you (30 days unless agreed otherwise), your company's databases, its files and its keys are deleted, and its entries in the directory removed. Copies in backups expire with the normal retention (12 months at most); destroying your company's key makes the secrets in those copies unreadable. We confirm the deletion in writing, with the date the last backup copy expires.

Personal information

IMS Creator holds a little personal information about your people (names, work email addresses, roles, sign-in history), and whatever your documents themselves hold. We handle it under the Australian Privacy Principles, as set out in our privacy notice. For the documents you write in IMS Creator, your company decides what goes in them; we process them only to provide the service.

Claude and your documents

IMS Creator writes the documents a management system needs (the context analyses, policies, registers, plans and procedures) with Claude, Anthropic's AI model, and reads drawings with it. This section says exactly what is sent to Claude, when, on whose account, and what happens to it there. It is the part security reviews ask about most.

On your own Claude account

Claude work runs on **your company's own Claude account**, never on ours, and you choose the provider:

Provider	What you give IMS Creator	Where Claude runs	Your agreement
Anthropic API	An API key from your Claude Console	Anthropic's infrastructure, where Anthropic's terms and your account's settings say	Anthropic's Commercial Terms and Data Processing Addendum
AWS Bedrock	An AWS region and an access key	The AWS region you choose (an Australian region is available)	Your AWS agreement
Google Vertex AI	A Google Cloud project, a region and a service account key	The Google Cloud region you choose (an Australian region is available)	Your Google Cloud agreement

So the provider that processes your content is **your** supplier, under **your** contract, with **your** data settings. You can see every call in your provider's own console, set spend limits there, and revoke the key there at any time. For processing in Australia, choose Bedrock or Vertex AI in an Australian region.

A site can use its own Claude connection instead of the company's, for example to keep one site's costs apart.

Your key

- Checked with one small call when it is entered, then **never shown again**: the app shows only the provider, the region, the key's last four characters, who set it and when.
- **Encrypted with your company's (or site's) own key** and decrypted only for the moment of a call made for your work.
- If the provider refuses it (revoked, out of credit, over its limit), Claude work stops at that site, the reason is shown, and your company admins are emailed once. Unusual use (a daily spike against the past fortnight, or use out of hours) is also emailed to them.
- Every call's token counts are recorded for your site, so your admins can see what Claude costs them.

What is sent to Claude, and when

Claude is called only when someone at your company asks for it (writing or reviewing a document, reading a batch of drawings, proposing asset register entries), or by a scheduled job your admins set up. What each call carries:

Work	What is sent	What comes back
Writing a document (context, risks, objectives, plans, procedures)	Your site's or company's profile (what you entered about it); the documents it builds on; any documents you chose to give it in full (for example an annual report); the instructions for that document	A draft, for a person to review, edit and approve
Reviewing a document	The document, and what has changed since it was written	Proposed changes, for a person to accept or not
Reading drawings	Page images of the drawings in the batch (JPEG and PNG), and close-ups of title blocks and handwriting	Title block details, tags, a summary
Asset information (classes, failure modes, maintenance programs)	The asset classes, their attributes, and a summary of the work order history for those classes	Proposed entries, for a person to accept

Not sent: passwords, authenticator secrets, other companies' information, your people's sign-in history, the audit log. Each call carries only what that piece of work needs.

Web research. When writing context documents, Claude can search the web and read public pages to research your site and its environment (with every fact cited). These searches run **on your provider's side** (Anthropic's web search and web fetch tools), and the search terms Claude writes can include your site's name and location. Web research can be turned off for a site in its settings; Claude then works from your profile alone and lists what should be checked.

What Claude cannot do. Claude is given no tools that act on IMS Creator or your systems: its answers are text, kept as **proposals that a person reviews and approves**. Nothing it writes is run as code. Uploaded documents and drawings are given to it as material to read, not as instructions, and its answers come back as a form with fixed fields, which is what is stored.

What your provider keeps

This is set by your provider's terms, not ours. As published at the time of writing (November 2026); check the current terms through the links below.

Anthropic API

- **No training on your content.** Anthropic's Commercial Terms say: "Anthropic may not train models on Customer Content from Services." Customer Content means your prompts (inputs) and Claude's answers (outputs).
- **Retention.** Anthropic's documentation on API data retention says that prompts and outputs are not retained by default, except for its *covered models* (its most capable, Mythos-class models, such as Claude Fable), which keep prompts and outputs for **30 days** for safety work, deleted automatically after that, with no Anthropic personnel able to read them by default. IMS Creator uses Claude Opus, Sonnet and Haiku, which are not covered models, unless your admins choose a covered model for a site.
- **Kept longer only when flagged or required by law.** Content flagged by Anthropic's automated trust and safety systems may be kept for up to 2 years, and when the law requires it.
- **Zero data retention.** Anthropic offers zero data retention by arrangement with your organisation. IMS Creator works with it: it uses the Messages API, prompt caching and web search (all eligible) and **does not use** the Batch API or the Files API (which keep data for a time), on the hosted

service. If you need web fetch inside a zero data retention arrangement, check its current eligibility with Anthropic.
- **Prompt caching.** When a long document (such as an annual report) is given to Claude for several steps of one piece of work, IMS Creator marks it for caching, so it is read once. Anthropic holds the cached form in memory only, for 5 minutes after last use, and caches are never shared between organisations.

AWS Bedrock: AWS states that Bedrock does not store or log your prompts and completions, does not use them to train models, and does not share them with model providers: Anthropic has no access to them.

Google Vertex AI: governed by your Google Cloud terms for generative AI (including Google's commitment not to train on your data without permission); see Google Cloud's documentation on Vertex AI data governance.

Sources: Anthropic Commercial Terms · Anthropic API and data retention · Data retention for covered models · Prompt caching · Web search tool · Amazon Bedrock data protection · Vertex AI data governance

On our side

- What Claude returns is stored in **your company's own databases**, like everything else you write in IMS Creator.
- While a batch of drawings is being read, Claude's answers are held on the server's encrypted disk until they are collected into the register, then deleted.
- We do not use your content, or Claude's answers about it, for anything but providing the service to you: not to train or tune any model, not to improve prompts with your material, not shared with anyone.

Choices you have

You want	Do this
Claude processing in Australia	Connect AWS Bedrock or Google Vertex AI in an Australian region
No retention at the provider at all	Anthropic API with zero data retention arranged with Anthropic, or Bedrock
No web searches about your sites	Turn off web research in the site's settings
A limit on spend	Set it at your provider; IMS Creator also emails your admins about unusual use
To stop Claude at once	Remove the connection in IMS Creator, or revoke the key at your provider

Identity and access

Your people signing in

- **Usernames are work email addresses** at a domain your company has approved and verified. A person belongs to one company.
- **People are invited** by your company admins (an emailed link, single-use, good for 7 days), or register with an address at your domain and are approved by your admins.
- **Authenticator first.** For an account with an authenticator app (TOTP, RFC 6238), its code is asked for **before the password**, so a password leaked elsewhere cannot even be tried. Your company can require an authenticator for everyone. Ten single-use recovery codes are given once, and only their hashes kept.
- **Passwords** are at least 12 characters, with no composition rules or forced expiry (as NIST SP 800-63B advises), checked against known breached passwords without sending the password anywhere (k-anonymity), and stored with Argon2id.
- **Lockouts**: 5 wrong codes lock the account for 15 minutes, 10 wrong passwords for an hour, and the person is emailed. Sign-in attempts are also limited per address, and Cloudflare's bot check runs on the sign-in steps.
- **Resets** are by single-use emailed links (good for 1 hour), never by temporary passwords, and end every other session. A reset does not get round the authenticator.
- **Single sign-on** (Microsoft Entra ID and others) is planned, not yet available.

Sessions

- Sessions are kept on the server: a 256-bit random id, of which only a hash is stored, in a `__Host-` cookie that is HttpOnly, Secure and SameSite=Strict.
- They end after 12 hours without use, and after 7 days in any case; when the password, the person's roles or their authenticator change; and when they or an admin signs them out.
- People can see where they are signed in and sign out there. Company admins can sign anyone at the company out.

Roles

Role	Can
Company admin	Everything at every site; people, roles, approved domains, the Claude connection, the subscription
Site admin	Everything at their site, including its people and settings
Engineer	Write and change the documents, registers and plugin records at their site
Viewer	Read their site's documents and registers

Every request is checked for the person's company, site and role before anything is read, in one place for the whole application, denying by default. A site's id in a request must belong to the person's company and memberships.

The audit log

Your company admins see an audit log of sign-ins and failed attempts, authenticator changes, password resets, people and role changes, invitations, settings and Claude connection changes, plugin installs, exports and approvals. Each entry is **chained to the one before by a hash**, so an entry changed afterwards shows. It can be exported at any time.

Break-glass

So your company can always get back in, it has a **break-glass account**: its password is given to you once, sealed, for your safe; it needs an authenticator; and every use emails all your company admins and is flagged for review.

Our own administrators

- Platform administrators are separate accounts, not people of any company, and **reach no company's content** (see Your data).
- They sign in with an authenticator. From the internet they can sign in only **through Cloudflare Access, as themselves** (a request that came through Cloudflare otherwise cannot sign in as an administrator, or use an administrator's session).
- The server is administered over a private network (Tailscale), with SSH keys only: no passwords, no direct root sign-in. Today that is the owner alone.
- If we sign your people out during an incident, your company admins see it in your audit log.

Application security

How it is built

- **A small, known stack**: Python (FastAPI) on the server, PostgreSQL, an S3-compatible store, and a React front end. Standard, well-reviewed libraries for everything security-sensitive: Argon2id for passwords, AES (Fernet) for secrets, HMAC-SHA256 for signatures. No cryptography of our own.
- **Security in the design.** The architecture and its threat model are written down, and the application is assessed against the **OWASP Application Security Verification Standard (ASVS) 4.0.3, level 2**: see the self-assessment, requirement by requirement, with what is met, what is partial and what is planned.
- **Company isolation in one place.** The check of session, company, site and role happens in one layer for the whole application, not route by route, so a new page cannot forget it.

How each release is tested

Every change goes through the same pipeline before it can reach the server:

1. The **whole test suite** (several hundred tests, many repeated for every page and role), including the **isolation tests**: signed in as one company, every page and API call is asked for with another company's names and ids, and must return nothing.
2. **Dependency scanning**: Python packages against known vulnerabilities (pip-audit); the container image against known vulnerabilities (Trivy).
3. The image is built, tested again, scanned, and published, **pinned by its commit**.

The server updates itself only to images its own pipeline built, tested and scanned, and can roll back to the one before.

In the browser

- **Content Security Policy**: scripts, styles, images and connections from the app itself only; no inline scripts.
- Frames refused (`frame-ancestors 'none'`), no content-type sniffing, a strict referrer policy, a restrictive permissions policy, cross-origin opener isolation, and HSTS.
- Every request that changes something must carry a header only the app sends, on top of SameSite=Strict cookies, so another site cannot make one for you (cross-site request forgery).
- Everything shown is escaped by React; there is no HTML from users.

Input, files and the API

- Every request body is validated against a typed schema; database queries are parameterised (SQLAlchemy); there is no deserialisation of untrusted objects.
- Uploads are checked by content type, size and (for archives) contents; kept outside the web root; drawings deleted within 2 hours. Downloads are by id, never by a path from the request. Requests above 110 MB are refused.
- No server-side fetching of addresses users give, other than by the research tools Claude runs at your provider.
- The API's own documentation pages are not published on the hosted service.

Plugins

Plugins add registers and tools to a site, from a reviewed marketplace.

- A plugin's functions run as **WebAssembly in a sandbox**: no files, no network, and limits on memory, processing and time.
- Each runs **inside one site**, with only the permissions your admin consented to when installing it, and only if **signed with the platform's key after review**.
- Installs, updates and removals are in your audit log, and we can suspend a plugin everywhere at once.

Secrets and keys

- No secrets in the code (the repository is scanned). Platform secrets are held on the server in files readable only by the service (mode 600), on the encrypted disk.
- Your company's key, and each site's, seal your secrets; they can be rotated, and old secrets are re-sealed.
- Planned: keys held in a cloud key vault, with the move to the cloud configuration.

Known limits

From the ASVS self-assessment, stated plainly:

- Containers still run as root inside the virtual machine (they drop capabilities and are not privileged).
- Logs are kept on the server rather than shipped to a central, separate log store.
- Container images are not yet signed; production is protected by deploying only its own pipeline's images.
- No independent penetration test yet: one is planned before the service grows beyond its first customers, then yearly.

Hosting and operations

The server

- A **dedicated virtual machine** (Ubuntu Server LTS) on a server owned by IMS Creator, in Australia, used for nothing else that holds customer information.
- Its **data disk is encrypted** (LUKS2). Customer data, the temporary drawing area, secrets and logs are all on it.
- **No port is open to the internet**: the machine opens one outbound, encrypted tunnel to Cloudflare, and Cloudflare passes requests to it. The router has no port forwarded.
- It sits on **its own network segment**. Its firewall (nftables) lets SSH in only from the private administration network (Tailscale), with keys only, no passwords and no root sign-in. The containers **cannot reach the owner's home or management networks**: rules rebuilt each time Docker starts keep them out.
- The database and the file store listen only inside the machine; neither is published.
- **Security updates** are applied automatically (unattended upgrades) for the operating system. The application is updated from its own tested images, with rollback (see Application security).

At the edge (Cloudflare)

- TLS 1.2 or later, HSTS, and Cloudflare's managed firewall rules and bot protection in front of the application.
- Rate limits on the sign-in steps, on top of the application's own.
- A bot check (Cloudflare Turnstile) on the sign-in, password reset, registration and sign-up pages.
- Our administrators' own access goes through Cloudflare Access, as themselves.

Watching over it

- The application checks its own health (the database, the file store, free disk space) and the console shows it, with alerts for anything wrong: a process down, a disk nearly full, a backup that failed or is late, sign-in failures out of the ordinary, Claude usage spikes, break-glass use.
- An external check watches the sign-in page and the health endpoint, and alerts the owner.
- The service's logs hold no document content, and are kept for 90 days.

Backups and recovery

What	How	How often	Kept
Databases (every company's, and the directory)	Continuous archiving of every change, with full and differential backups	Continuous: at most **15 minutes** of changes at risk	30 daily, 12 monthly
File store (uploaded documents, images)	File backup	Nightly	30 daily, 12 monthly
Each company's databases on their own	Nightly copies, so one company can be restored without touching any other	Nightly	30 days
Configuration	In version control, and in the backups	On change	History

- Backups are **encrypted on the server before they leave it** (Proxmox Backup Server's client-side encryption), with the key kept off the server.
- They go to a **separate backup server**, which keeps an **off-site copy**: three copies, two kinds of storage, one off-site.
- **A restore is tested every quarter**, into a separate machine, and the time it took recorded. Targets: at most 15 minutes of changes lost (RPO) and **4 hours** to restore the service (RTO).

Changes

Changes to the service go through version control and the release pipeline (see Application security). Changes to the server's configuration are made through automation playbooks, kept in version control, rather than by hand.

The cloud configuration

The move to Microsoft Azure (regional deployments, managed databases, keys in a key vault, central logging) will be described in its own version of this pack, before any customer is moved.

Incident response

An incident is anything that has, or may have, exposed, changed or lost customer information, or stopped the service: a breach, a compromised account or key, a vulnerability being exploited, a failed backup we depend on, a long outage.

What we do

1. **Contain, at once.** We can suspend a company, sign everyone at a company (or everyone) out, rotate your company's keys and the platform's secrets, revoke a plugin everywhere, or take the service offline by closing the tunnel.
2. **Preserve the evidence.** A snapshot of the server, its logs, the affected companies' audit logs and Cloudflare's logs, before any of them roll over. A written log is kept from the first minute: what was seen, when, what was done.
3. **Assess.** Which companies, which information, how, and since when, from the audit logs, the platform's events and the web logs.
4. **Tell you.** Each affected customer is told **within 72 hours of our becoming aware**: what happened, what information, what we have done, and what you should do. We tell you sooner where we can, and keep you updated.
5. **Recover.** Fix the cause, restore from backup if needed, bring companies back, and watch closely.
6. **Learn.** Within two weeks: what happened, why, and what changes; the procedures are updated. Affected customers get the summary.

Your obligations

We help you with your own: the Australian **Notifiable Data Breaches** scheme (we give you what you need to assess and notify), privacy and breach laws elsewhere, and any obligations your company has as critical infrastructure, with prompt and detailed information.

Who to tell, and how

- Report a suspected incident or vulnerability to **security@imscreator.com**. See Reporting a vulnerability.
- Tell us at once if you think one of your people's accounts, or your Claude key, has been compromised: your company admins can also sign the person out everywhere, reset their authenticator, and replace the Claude connection themselves, straight away.

Outages

If the service is unavailable, we restore it from the most recent backup if need be, within the 4-hour target, and tell affected customers what happened.

Service providers

The companies whose services IMS Creator uses to run the hosted service, what each one sees, and where.

Provider	What for	What it sees	Where
Cloudflare	DNS, TLS at the edge, the tunnel to the server, firewall, bot checks, Access for our administrators, email forwarding for imscreator.com addresses	Requests and responses pass through its network in transit (it ends TLS); it keeps request logs (addresses, paths, times), not your content	Global network
Postmark (ActiveCampaign)	Sending the service's email: invitations, password resets, notices, support replies	The recipient's name and address, and the email's text (which can include a support reply)	United States
GitHub (Microsoft)	The source code, the release pipeline, the container images	Code only: no customer information	United States
Tailscale	The private network our administrators reach the server through	Connection details of our administrators' devices; no customer information	Global
Bluehost	This website (imscreator.com) and its contact form	What you send through the contact form	United States

The server, its backup server and the off-site backup copy are owned and run by IMS Creator in Australia; no third party hosts or holds your information at rest.

Your Claude provider is yours

Anthropic, AWS (Bedrock) or Google (Vertex AI) process your content when Claude writes for you, but **under your own account and your own agreement** with them, using credentials you give us: they are your suppliers, not ours, and you choose which, and where. See Claude and your documents.

Changes

We tell customers before adding a provider that would see your information, so you can object.

Shared responsibility

Security is shared: some of it is ours, some is your company's.

Area	IMS Creator	Your company
The service: servers, network, encryption, patching, backups, monitoring	✔	
The application's security, testing and updates; keeping companies apart	✔	
Incident response, and telling you within 72 hours	✔	Telling us of anything suspicious; your own notifications
Who has an account, and their roles		✔ Your company admins invite, approve and remove people, and review roles
Approved email domains	Checking a domain belongs to you when we approve it by hand	✔ Approving and verifying your domains; removing those you no longer use
Authenticators	Offering them, and asking for the code first	✔ Requiring them for everyone (recommended), and keeping recovery codes safe
The break-glass account	Sealing and issuing its password	✔ Keeping it in your safe, and reviewing every use
The audit log	Keeping it complete and tamper-evident	✔ Reviewing it, or exporting it to your own monitoring
Claude	Sealing your key, using it only for your work, alerting on unusual use	✔ Your agreement with the provider, its region and data settings, spend limits, revoking keys
What goes into documents	Processing it only to provide the service	✔ Deciding what your documents contain, and reviewing what Claude proposes before approving it
Plugins	Reviewing, signing and sandboxing them	✔ Choosing which to install, and the permissions you consent to
Your people's devices and browsers		✔ Kept up to date and protected
Leaving	Exporting your information, then deleting it, and confirming	✔ Asking for the export, and keeping it

Common questions

Short answers to what security questionnaires ask most. Each links to the section with the detail.

The company and the service

Question	Answer
What is the service?	A web application for writing and running an integrated management system (ISO 9001, 14001, 45001, 55001 and others): its documents, registers and plans, at one or more sites.
Who runs it?	IMS Creator, in Australia.
Where is our data held?	In Australia: the server, its backup server and the off-site copy. Claude processing is where your own Claude provider and region are. Data
Is it multi-tenant?	Yes, with a database per company and per site, and a separate place in the file store per company. Data
Do you have ISO 27001 or SOC 2?	Not yet. An OWASP ASVS level 2 self-assessment is published; an independent penetration test comes before the service grows beyond its first customers, then yearly. Overview
Can we see a penetration test report?	When the first one is done, its summary is shared with customers.

Data

Question	Answer
Is data encrypted at rest?	Yes: the data disk (LUKS2), your secrets again with your company's own key, and backups encrypted before they leave the server. Data
In transit?	Yes: TLS 1.2 or later everywhere, with HSTS. Architecture
Can your staff read our documents?	Our console cannot show them. Server access is limited to the owner, over a private network, for operating the service, such as a restore you ask for. Data
How long do you keep our data?	While you subscribe. Drawings themselves: 2 hours at most. Logs: 90 days. Backups: 30 daily and 12 monthly. Data
Can we get our data out?	Documents as Word and registers as Excel at any time; a full export on request and when you leave. Data
What happens when we leave?	A full export, then deletion of your databases, files and keys; backups expire within 12 months; confirmed in writing. Data
Is our data used to train AI?	No. Not by us, and under Anthropic's Commercial Terms not by Anthropic; AWS Bedrock and Google Vertex AI make the same commitment. Claude

Claude and AI

Question	Answer
Whose AI account is used?	Yours: Anthropic API, AWS Bedrock or Google Vertex AI, under your own agreement. Claude

Question	Answer
What is sent to the AI?	Only what each piece of work needs: your site's profile and related documents to write a document; page images to read drawings; asset classes and a work history summary for asset information. Claude
What does the provider keep?	Anthropic: not retained by default except for its covered models (30 days); longer only if flagged or required by law; zero data retention available. Bedrock: not stored. Claude
Can we keep AI processing in Australia?	Yes: AWS Bedrock or Google Vertex AI in an Australian region. Claude
Does the AI act on its own?	No. Its answers are proposals that people review and approve; it has no tools that act on the service or your systems. Claude
Does the AI search the web?	When writing context documents, if web research is on for the site (it can be turned off). The searches run at your provider. Claude
How is our API key protected?	Encrypted with your company's own key, never shown again after it is entered, used only for your work. Claude

Access

Question	Answer
Do you support MFA?	Yes: an authenticator app, asked for before the password; your company can require it for everyone. Access
Single sign-on?	Planned (Microsoft Entra ID and others), not yet available. Access
Password policy?	At least 12 characters, checked against breached passwords, stored with Argon2id, no forced expiry (NIST SP 800-63B). Access
Session timeout?	12 hours without use, 7 days in any case. Access
Is there an audit log?	Yes: sign-ins, changes, exports and approvals, hash-chained so tampering shows, exportable. Access
Role-based access?	Yes: company admin, site admin, engineer, viewer, checked on every request. Access

Operations

Question	Answer
Backups?	Continuous (at most 15 minutes at risk), encrypted, off-site, restored as a test every quarter. Operations
Recovery targets?	RPO 15 minutes, RTO 4 hours. Operations
Patching?	Operating system security updates automatically; the application from its own tested and scanned images. Operations
Vulnerability scanning?	Dependencies (pip-audit) and container images (Trivy) on every release. Application
Breach notification?	Within 72 hours of our becoming aware. Incidents
Subprocessors?	Cloudflare, Postmark, GitHub (code only), Tailscale (administration only), Bluehost (this website only). Providers

Question	Answer
Business continuity?	Restore from encrypted off-site backups to a new machine; targets as above. Operations

ASVS level 2 self-assessment

Hosted IMS Creator (`app.imscreator.com`, Stage A) against the Application Security Verification Standard, level 2, as at go-live. **Met**: in place, with evidence (a test, code or a setting). **Partial**: in place with a stated limit. **Gap**: not yet, with the plan. Chapters and requirement numbers are ASVS 4.0.3's. Reviewed by: the owner, before go-live; again before Stage B and after any major change. The independent penetration test (before Stage B or a third customer) checks this assessment.

Summary

Chapter	Status	The main points
V1 Architecture	Met	Documented design and threat model (CLOUD-ARCHITECTURE.md); tenancy enforced in one place (the host); isolation proven by tests on every build
V2 Authentication	Met	NIST 800-63B passwords, breached-password check, Argon2id, TOTP before the password, recovery codes, lockouts, emailed single-use resets
V3 Sessions	Met	Server-side sessions, 12 h idle / 7 days, revocable, ended on password, role and MFA changes; `__Host-` HttpOnly Secure SameSite=Strict cookie
V4 Access control	Met	Deny by default; company from the session only; roles checked per request; operators reach no company content (tests)
V5 Validation and encoding	Met	Pydantic models and typed routes; parameterised SQL (SQLAlchemy); React escaping; paths and uploads checked
V6 Cryptography	Partial	Standard libraries only (Argon2id, Fernet/AES, HMAC-SHA256); per-site and per-company keys, rotatable; master-key wrapping comes with Stage B
V7 Errors and logging	Partial	Audit log per company, hash-chained, exportable; no secrets or content in logs; central log shipping is a Stage B item
V8 Data protection	Met	No drawings kept; Claude keys sealed, never shown; caching off for the app shell; backups encrypted client-side
V9 Communications	Met	TLS 1.2+ at Cloudflare, HSTS, tunnel encrypted; internal services on the Docker network only
V10 Malicious code	Partial	Dependencies pinned and scanned (Trivy, pip-audit); production updates itself only to images its own CI built, tested and scanned, pinned by commit, with rollback; ClamAV on uploads when built in; code signing of images, and branch protection as the gate to production, are Stage B items
V11 Business logic	Met	Rate limits on sign-in steps; last-admin and break-glass protections; Claude spend alerts; batches limited in size and time
V12 Files and resources	Met	Uploads by content type, size and zip checks, outside the web root, deleted within 2 hours; downloads by id, path-checked
V13 API	Met	JSON API with CSRF header requirement, typed schemas, 413 above 110 MB, no API docs exposed on the host
V14 Configuration	Partial	Security headers and CSP; secrets in `.env` (mode 600); containers drop capabilities; containers still run as root

V1 Architecture, design and threat modelling

Req.	Status	Evidence
1.1.1-1.1.7 Secure development, threat model, components documented	Met	docs/CLOUD-ARCHITECTURE.md (sections 4, 5, 11, 12: components, tenancy, controls, threat model); this assessment
1.2 Authentication architecture	Met	One sign-in path (readiness/web/host.py); low-privilege database use per company database; operators separate from people
1.4 Access control architecture	Met	The host checks session, company, site and role before any site API; enforced in one place, not per route; `tests/test_isolation.py`
1.5 Input and output architecture	Met	Serialisation through Pydantic/JSON only; no deserialisation of untrusted objects
1.6 Cryptographic architecture	Partial	Keys per site and per company (`rotate-keys`); a master key (KMS) wrapping them is Stage B
1.8 Data protection	Met	Data classification (CLOUD-ARCHITECTURE.md 9.1); drawings not stored
1.9 Communications	Met	All external traffic through the Cloudflare tunnel; Postgres and MinIO not published
1.12-1.14 Uploads, configuration	Met	Separate temporary batch volume, excluded from backups; pinned images; deploy runbook

V2 Authentication

Req.	Status	Evidence
2.1.1-2.1.12 Password security	Met	12-256 characters, any characters, no composition rules or expiry, breached-password check at setting (k-anonymity), paste allowed, change requires the current password (`identity.password_problem` , tests in `test_signin.py`)
2.2.1 Anti-automation	Met	5 wrong codes: 15-minute lock; 10 wrong passwords: 1-hour lock; per-address limit (60 per 5 minutes); Cloudflare rate limits
2.2.2-2.2.3 Notify of authentication events	Met	Locks, authenticator resets and break-glass use are emailed
2.3 Authenticator lifecycle	Met	Invitations and resets by single-use, expiring links (7 days, 1 hour); no temporary passwords on the hosted service
2.4.1-2.4.5 Credential storage	Met	Argon2id (argon2-cffi defaults); older scrypt hashes rehashed at sign-in
2.5 Credential recovery	Met	Emailed reset link only; ends every session; does not bypass the authenticator
2.7-2.8 Out-of-band and one-time codes	Met	TOTP (RFC 6238), each code once (`mfa_last_step`), 10 hashed single-use recovery codes, secrets sealed with the company's key
2.10 Service authentication	Met	No default or shared service accounts in the app; database and store credentials only in `.env`

V3 Session management

Req.	Status	Evidence
3.2.1-3.2.3 Session binding	Met	256-bit random session ids, only their hash stored; new id at each sign-in; cookie value signed
3.3.1-3.3.4 Logout and timeout	Met	Sign-out revokes server-side; 12 h idle, 7 days absolute; list and end other sessions; password change ends the others
3.4.1-3.4.5 Cookies	Met	`__Host-` prefix, Secure, HttpOnly, SameSite=Strict, Path=/ (`test_hardening.py`)
3.5 Token-based sessions	n/a	Not used
3.7.1 Re-authentication before sensitive changes	Partial	A password change needs the current password, removing an authenticator needs a code; other admin actions rely on the session (acceptable at 12 h idle; to revisit)

V4 Access control

Req.	Status	Evidence
4.1.1-4.1.5 General	Met	Deny by default (401/403 before any site route); the company is the session's, never a request's
4.2.1 IDOR	Met	Every GET route swept with the other company's names and ids (`test_isolation.py`, 70 routes, 200+ requests)
4.2.2 CSRF	Met	SameSite=Strict plus the `X-ORT-Request` header on every state-changing request (`test_signin.py`)
4.3.1 Admin interfaces	Met	Platform administration requires an administrator with an authenticator; from the internet only through Cloudflare Access as that administrator (its signed token checked on every request); reaches no company's site
4.3.2 Directory browsing	Met	No listings served; static assets by name only

V5 Validation, sanitisation and encoding

Req.	Status	Evidence
5.1 Input validation	Met	Pydantic models for settings; typed parameters; allow-lists for roles, regions, providers, kinds
5.2 Sanitisation	Met	File and path names cleaned (`insider._safe`, `localfolder.rel`, `storage._check`)
5.3 Output encoding	Met	React escapes by default; the two raw HTML insertions are the app's own help pages (from its Markdown) and the QR SVG it generates
5.3.4 SQL injection	Met	SQLAlchemy throughout; the few textual statements use bound parameters or fixed identifiers
5.3.8 OS command injection	Met	Subprocesses take argument lists, never a shell string
5.5 Deserialisation	Met	JSON and YAML `safe_load` only

V6 Stored cryptography

Req.	Status	Evidence
6.1 Data classification	Met	CLOUD-ARCHITECTURE.md 9.1
6.2.1-6.2.6 Algorithms	Met	Fernet (AES-128-CBC + HMAC-SHA256), Argon2id, SHA-256, HMAC-SHA256; no home-made crypto (TOTP per RFC 6238 with HMAC-SHA1, as the RFC)
6.3 Random values	Met	`secrets` / `os.urandom` for ids, tokens, keys
6.4.1-6.4.2 Key management	Partial	Keys per site and company, rotatable (`rotate-keys` , runbook); keys are files on the encrypted disk, not in a vault: KMS in Stage B

V7 Error handling and logging

Req.	Status	Evidence
7.1.1-7.1.4 Log content	Met	No passwords, codes, tokens or Claude keys logged; drawing content not logged
7.2 Security events	Met	Company audit log: sign-ins and failures with address, locks, roles, invitations, MFA, domains, Claude connections, key rotation, exports, break-glass
7.3.1-7.3.4 Log protection	Partial	Audit entries hash-chained (tampering shown on Company, `test_hardening.py`), exportable as CSV; application logs stay on the VM (capped); central shipping in Stage B
7.4 Error handling	Met	Generic messages to users; no stack traces in responses

V8 Data protection

Req.	Status	Evidence
8.1 General	Met	Drawings never stored (batches deleted within 2 hours, excluded from backups); per-company databases and bucket prefixes
8.2 Client-side	Met	No sensitive data in local storage (only the theme, and folder handles in IndexedDB); `Cache-Control: no-cache` on the app shell
8.3 Sensitive private data	Met	Claude keys never shown after saving; operators see counts, not content; export and deletion runbooks

V9 Communications

Req.	Status	Evidence
9.1 Client communications	Met	TLS 1.2+ (Cloudflare), HSTS (12 months), always HTTPS
9.2 Server communications	Met	Tunnel encrypted; Claude, Postmark, DNS-over-HTTPS and breach-check calls over TLS with verification; internal traffic on the private Docker network

V10 Malicious code

Req.	Status	Evidence
10.2 Malicious code search	Partial	Reviewed by the owner; no remote code loading; automatic updates take only the repository's own CI-built images (whoever can push to the branch reaches production, so pushes are the owner's)
10.3 Application integrity	Partial	Images built by CI from the repository, scanned (Trivy), pinned by commit; image signing in Stage B

V11 Business logic

Req.	Status	Evidence
11.1.1-11.1.8	Met	Steps in order (code before password, within 2 minutes); limits on sign-in, upload size and batch count; the last company admin and the break-glass account protected; Claude use alerts

V12 Files and resources

Req.	Status	Evidence
12.1 Upload	Met	Size limits (500 MB a file, 110 MB a request); zip bombs refused (ratio, total, count); `test_localfolder.py`
12.2 Integrity	Met	Type by content, matching the name; SHA-256 checked on arrival
12.3 Execution	Met	Uploads never executed or served as-is; paths cannot leave the batch (`rel` , zip-slip check)
12.4 Storage	Met	Outside the web root, on a separate volume
12.5-12.6 Download and SSRF	Met	Downloads by id within the batch or store; no user-supplied URLs fetched by the server

V13 API

Req.	Status	Evidence
13.1 General	Met	Same authentication and authorisation for every API route
13.2 RESTful	Met	CSRF header on changes; JSON content type; methods per action; request size limit (413)

V14 Configuration

Req.	Status	Evidence
14.1 Build and deploy	Met	CI builds, tests (incl. isolation on Postgres) and scans; deploy runbook with staging
14.2 Dependencies	Met	Pinned, scanned in CI (Trivy, pip-audit)
14.3 Unintended disclosure	Met	No server paths or versions of components in responses to companies; API docs not on the host
14.4 HTTP security headers	Met	CSP (scripts by hash), X-Content-Type-Options, X-Frame-Options, Referrer-Policy, Permissions-Policy, COOP, HSTS (`test_hardening.py`)

Req.	Status	Evidence
14.5 Request validation	Met	Host header fixed by the tunnel; CORS not enabled
Container hardening (beyond ASVS)	Partial	`no-new-privileges`, all capabilities dropped, memory limits; still root inside, writable file systems: to change with a non-root image user

Fixed for go-live

Found by this assessment and fixed before go-live (commit with this file):

- Security headers and a content security policy (V14.4) on the host and on a site of its own.
- The session cookie bound to the host on HTTPS (`__Host-` prefix) (V3.4).
- Requests over 110 MB refused before they are read (V13.2).
- The company audit log hash-chained, its integrity shown, exportable (V7.3).
- Dependency auditing in CI (pip-audit) beside the image scan (V14.2).
- The restore drill automated on Postgres (`tests/test_restore.py`).
- Earlier in the build: the drawing register's prompts no longer describe one company's drawings as another's; the Claude key never falls back to the host's own; links not spent by a password that is refused.

Open items

Item	Plan
Master key (KMS) wrapping company keys (V6.4)	Stage B (Key Vault / KMS)
Central, tamper-resistant log shipping (V7.3)	Stage B
Re-authentication before sensitive admin actions (V3.7)	Next release: re-enter the password or a code for role changes to company admin, break-glass and Claude connection changes
Containers as a non-root user, read-only file systems	Next release: an `amtoolkit` user in the image, folder ownership set by the init service
Image signing (V10.3)	Stage B (cosign)
Per-company Postgres roles	Before the third customer
Independent penetration test	Before Stage B or a third customer

Reporting a vulnerability

If you believe you have found a security problem in IMS Creator, please tell us.

- Write to **security@imscreator.com** with what you found, how to reproduce it, and what it could let someone do. Screenshots and requests help.
- We acknowledge within **2 business days**, tell you what we will do within **5**, and let you know when it is fixed.
- Please give us reasonable time to fix it before telling anyone else, and we will credit you if you wish.

While testing, please:

- Use only accounts of your own, and never read, change or delete information that is not yours.
- Do not run denial-of-service tests, automated scanning at volume, or social engineering of our people or customers.
- Stop and tell us as soon as you reach information that is not yours.

We will not pursue anyone who reports in good faith and keeps to these rules.

Customers: report a suspected incident the same way, or through Support in the app for anything that is not urgent.