

GRADE 9 CBC INTERACTIVE DIGITAL CONTENT

Administrator & Deployment Guide

How the platform is built, published and maintained — and exactly where content may be edited, by whom.



Who this guide is for

- **Scope**

installing, publishing, updating and verifying the Grade 9 platform and its 11 learning-area packages.

- **Audience**

the technical staff at Eirene Technologies Limited responsible for the content build and the web server.

- **Not a learner document**

learners need only the Learner Guide; nothing here is required to study.

- **Authoring console**

authorised staff edit content in the browser at /studio/, behind staff authentication. It is NOT part of the delivered learning areas: every package and every learner-facing page remains a pure static file with no server component.

- **Learners still have no editing capability whatsoever**

the console is a separate, authenticated tool they cannot reach. Part 2 covers exactly who can change what.

IMPORTANT

This document names no server, account or credential. Those are held separately by Eirene Technologies Limited.

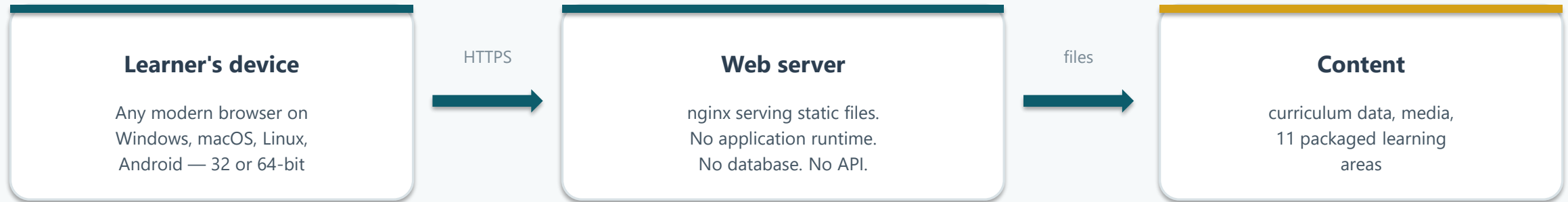
PART 1

How the platform is built

A static HTML5 site with no server-side code and no database.

Platform architecture

Everything a learner runs is plain HTML, CSS and JavaScript. The web server only hands over files.



What this design buys you

No server-side code

Tender spec 1.9 — nothing to patch, exploit or keep running

No external references

Spec 1.10 — a package opens with no internet and calls nothing out

Read-only by construction

There is no write path, so content cannot be altered in the browser

Runs anywhere

Same files serve the website and the offline flash disk

TIP

Because there is no server-side component, "deploying" is only ever copying files into the webroot.

What is published, and where

The webroot holds the portal, the 11 independent learning-area sites, and the downloadable packages.

WEBROOT

Portal pages

index · media library · packages · compliance

the front door; links to everything else

subjects/<slug>/

11 independent learning-area sites

each identical to the package a learner downloads

packages/

11 per-lot .zip + packages.json

one self-contained learning area per file

VERIFIED

subjects/<slug>/ is extracted from the very same .zip that is offered for download, so what a learner studies online and what they take away are byte-identical.

PART 2

Content and edit rights

Where the content lives, who may change it, and how a change reaches a learner.

Edit rights — who can change what

The answer KICD asked for: learners have no edit rights anywhere. Editing exists only behind staff authentication, outside the delivered content.

Learner

Read only

- ✓ Study any lesson
- ✓ Take quizzes and games
- ✓ Download packages
- ✓ Keep their own progress
- X Cannot edit any content
- X No editor is shipped
- X No login, no account

Authorised staff

Full content control

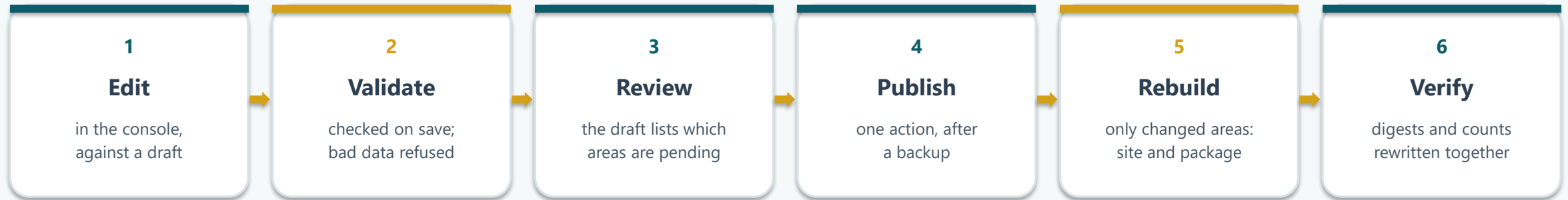
- ✓ Author and revise lessons
- ✓ Add, reorder, remove units
- ✓ Publish to the website
- ✓ Restore from backup
- Signs in to the console
- Edits a draft, not the live site
- Publishes deliberately

VERIFIED

The learning areas themselves contain no code that can write content, so a learner cannot gain edit rights even by mistake. The console is a separate authenticated tool and ships in no package.

From a lesson edit to a learner's screen

Every content change follows this one path. Edits land in a draft; the live site changes only when Publish is pressed.



The two gates that protect quality

Gate 1 — lesson validator

Refuses a save with no title, a quiz answer that indexes no option, or a link in the reading; house-style issues are flagged as warnings rather than blocking.

Gate 2 — package conformance

Rejects a build that is not self-contained: external references, wrong asset types, missing index.html, or more than one learning area in a package.

DO NOT

Never hand-edit files in the webroot. Publishing regenerates them and the change is lost — edit in the console instead.

Where each kind of content lives

Content	Source of truth	Reaches the learner via
Lesson text, outcomes, key terms	the lesson's JSON source file	assembled into the curriculum, then built into the package
Quiz questions	the same lesson file	the same build
Lesson video	generated narrated MP4	bundled inside each package's media folder
Audio narration	produced on the learner's device	nothing to ship — spoken from the transcript
Games and animations	built-in engines, driven by the lesson	no asset files; drawn by the browser
Subject names, colours, order	the curriculum data file	portal cards and every package

TIP

Audio, games and animations ship no media files at all — which is how packages stay small enough for a flash disk.

PART 3

Publishing an update

Build, verify, back up, publish, confirm — in that order, every time.

The publishing run

1

Edit

Sign in to the console and make the change. It is saved to a draft; the live site is untouched.

2

Review

The console shows which learning areas are pending. Check that list is what you expect.

3

Publish

Press Publish. Every file about to change is backed up first.

4

Rebuild

Only the changed areas are regenerated — online site, encrypted copy, manifest digests and the downloadable package together.

5

Confirm

Work through the verification checklist on the next slide, in a real browser.

VERIFIED

The deploy script refuses to run unless all 11 packages and the manifest are present and consistent.

Post-publication verification checklist

Check	Expected result
Home page	Loads, and the totals match the content you just built
Every learning area opens	All 11 Study links open their own site
Every package downloads	All 11 .zip links return a file, sizes as built
Flash-disk bundle	Downloads, and contains 11 separate learning areas
A package is self-contained	Unzip it, disconnect, open index.html — lessons and video still work
Counts agree	Portal totals equal the sum of the packages — no drift
Check in a real browser	Not only with a command-line tool — see the caching slide
Other sites on the server	Unaffected and still responding

IMPORTANT

If any check fails, restore the backup the deploy script made before investigating.

PART 4

Running it day to day

Caching, security, backups and the faults worth knowing in advance.

Caching — the rule that matters most

The site keeps working offline by caching itself. Get the policy wrong and visitors keep seeing an old build.

Code, data and pages

HTML · JavaScript · CSS · curriculum

NETWORK FIRST — always fetched fresh,
the cache is only a fallback when offline

Media

Video · audio · images · fonts

CACHE FIRST — never changes, so it is
kept locally and never re-downloaded

Why it matters

A cache-first policy on code served visitors an old version of this site long after a correct update — and a command-line check could never see it, because the caching happens inside the browser. Always confirm an update in a real browser as well.

VERIFIED

The independent learning-area sites are excluded from the portal's cache, so they are always served fresh.

Security, integrity and licensing

Area	How it is handled
Transport	HTTPS with an automatically renewed certificate; plain HTTP is redirected
Content encryption	AES-256 with PBKDF2 key derivation, demonstrated by a secure loader in every package
Package integrity	Each package carries a manifest with SHA-256 digests of its contents
No external calls	Packages reference nothing outside themselves — verified on every build
Learner privacy	Progress is stored only in the learner's own browser; nothing is collected or transmitted
Licence	Perpetual licence to KICD; copyright remains with Eirene Technologies Limited

TIP

The encryption demonstration key ships inside the package by design — it proves the mechanism; it is not a secret.

Faults worth knowing in advance

Symptom	Cause and remedy
Visitors see an old version	A stale browser cache. Confirm the caching policy above, then reload. Never diagnose with a command-line tool alone.
A learning area 404s	The package was not extracted during publish. Re-run the deploy; it rebuilds every learning-area site.
Portal and package totals differ	Curriculum and packages were rebuilt separately. Always rebuild and publish them together.
Package fails conformance	Usually an external reference or a disallowed file type. Fix the source, rebuild — never patch the .zip.
Certificate warning	Automatic renewal did not run. Check the renewal timer is enabled.
Staging lost after a restart	Temporary directories are cleared on reboot. Stage in a persistent location.

VERIFIED

Every publish leaves a timestamped backup of the previous site — restoring is always an option.

Summary for administrators

- Static HTML5 only — no server-side code, no database, nothing to keep running.
- Learners cannot edit content because no editing interface is shipped.
- Content changes follow one path: author → validate → assemble → build → verify → publish.
- Portal and packages must be rebuilt and published together, or their totals drift.
- Confirm every update in a real browser, not only from the command line.