Gemini / LLM Usage Dashboard

The **Gemini Usage** dashboard provides operational visibility into the Large Language Model usage generated by Open Clinical History.

The page is:

```
/gemini_usage.php
```

It displays:

- current LLM configuration state
- today's request budget
- today's token budget
- historical request and token usage
- usage by provider and model
- current configured provider limits
- recent patient-document LLM usage

The page is **read-only**.

Configuration changes are made under:

Admin → Configuration

Purpose

Open Clinical History uses an LLM during several processing operations, including:

- patient-history segmentation
- clinical event extraction
- document reconciliation
- grounded SNOMED resolution
- SNOMED repair
- anatomical body-layer selection

- clinical audit
- image/SNOMED catalogue construction

Some patient histories can require dozens of LLM calls.

The Usage dashboard provides a single location for monitoring that activity.

Conceptually:

```
Open Clinical History
        |
        +--> Source segmentation
        |
        +--> Event extraction
        |
        +--> Reconciliation
        |
        +--> SNOMED resolution
        |
        +--> Anatomy selection
        |
        +--> Clinical audit
        |
        v
Configured LLM provider
        |
        v
Usage recorded
        |
        v
llm_usage
        |
        v
Gemini Usage Dashboard
```

Usage Tracking

LLM usage is recorded in:

```
llm_usage
```

Usage is grouped by:

```
date
+
provider
+
model
```

For example:

```
2026-08-25
gemini
gemini-3.5-flash-lite
```

has its own request and token counters.

This allows Open Clinical History to retain usage history even if the application or worker processes restart.

Dashboard Refresh

The Usage dashboard automatically refreshes every:

```
30 seconds
```

This makes it useful while:

- importing patient histories
- running large extraction batches
- rebuilding image/SNOMED mappings
- processing terminology workloads

There is no need to manually reload the page continuously.

Database Day

At the top of the dashboard, the page displays:

```
database day YYYY-MM-DD
```

The current date is obtained from the database using:

```
CURDATE()
```

where possible.

This is significant because daily usage limits are also recorded according to the database day.

The database server's timezone should therefore be configured appropriately for the installation.

Time Period

Historical usage can be viewed over:

```
7 days
30 days
90 days
1 year
```

The default is:

```
30 days
```

The period can also be selected through the URL:

```
gemini_usage.php?days=7

gemini_usage.php?days=30

gemini_usage.php?days=90
```

```
gemini_usage.php?days=365
```

Any unsupported value falls back to 30 days.

LLM Status

The status indicators near the top of the page show the current runtime configuration.

For example:

```
LLM enabled

Gemini key configured

provider gemini

active model gemini-3.5-flash-lite
```

LLM Enabled

This reflects:

```
llm_enabled
```

from **Admin → Configuration**.

If disabled, Open Clinical History will not create an LLM provider for processing requests.

The usage dashboard remains available so historical usage can still be inspected.

Gemini Key Configured

This indicates whether:

```
llm_gemini_api_key
```

contains a value.

The API key itself is **never displayed** on the Usage dashboard.

The indicator shows only:

```
configured
```

or:

```
missing
```

Provider

The dashboard displays the currently configured:

```
llm_provider
```

The current implementation uses:

```
gemini
```

The underlying LLM architecture is provider-aware, so usage records retain the provider name rather than assuming all historical requests necessarily belong to Gemini.

Active Model

The currently configured model comes from:

```
llm_gemini_model
```

For example:

```
gemini-3.5-flash-lite
```

Changing the configured model does not remove usage recorded against an earlier model.

Historical usage remains visible separately.

Today's Request Budget

The first budget panel displays:

```
Today's request budget
```

For example:

```
2,145 / 50,000
```

The configured maximum comes from:

```
llm_daily_request_cap
```

The counter represents requests recorded today for the **currently active provider and model**.

The dashboard also calculates:

- percentage used
- requests remaining

For example:

```
4.29% used · 47,855 remaining
```

Request Cap Enforcement

Before each LLM request, Open Clinical History checks the current usage for:

```
today
+
active provider
+
active model
```

If the number of requests has already reached the configured cap, the request is blocked before Gemini is contacted.

For example:

```
Recorded today: 50,000
Configured cap: 50,000
        |
        v
New LLM request
        |
        v
BLOCKED
```

The caller receives an LLM budget error explaining that the daily request cap has been reached.

Unlimited Request Budget

A configured request cap of:

```
0
```

means:

```
Unlimited
```

The dashboard displays the infinity symbol for the budget rather than calculating a percentage.

Today's Token Budget

The second budget panel displays the token allowance.

For example:

```
2,400,000 / 8,000,000
```

The configured maximum comes from:

```
llm_daily_token_cap
```

The token total is:

```
prompt tokens
+
output/thinking tokens
```

Prompt Tokens

Prompt tokens are reported by Gemini as:

```
promptTokenCount
```

These represent the input supplied to the model, including the effective system and user prompt.

Output / Thinking Tokens

For Gemini 3.x models, Open Clinical History includes both:

```
candidate/output tokens
+
thinking tokens
```

in the recorded output figure.

Conceptually:

```
output_tokens =
candidatesTokenCount
+
thoughtsTokenCount
```

This is deliberate.

Thinking tokens are still part of model consumption, even though the model's internal thinking content is not included in the clinical JSON returned to the application.

Therefore:

> **Output / thinking** is a usage figure, not simply the number of visible response tokens.

Token Cap Enforcement

Before calling the LLM, Open Clinical History calculates:

```
today's prompt tokens
+
today's output/thinking tokens
```

If that value has already reached the configured daily token cap, the request is blocked.

For example:

```
Current recorded use: 8,000,000
Configured limit:     8,000,000
```

```
Next request → blocked
```

Token Caps Are Pre-request Guards

The exact number of tokens a request will consume cannot be known until the provider has processed it.

The token budget is therefore checked **before** a request using already recorded consumption.

For example:

```
Current usage: 7,990,000
Daily cap:     8,000,000
        |
        v
Request allowed
        |
        v
Request consumes 20,000 tokens
        |
        v
New recorded total: 8,010,000
```

The cap can therefore be exceeded slightly by the request that crosses the boundary.

With multiple workers making requests concurrently, several requests can also pass their pre-request checks before another worker's usage has been recorded.

The daily caps should therefore be treated as **protective application limits**, not exact billing ceilings.

Daily Budgets Are Per Provider and Model

This is an important implementation detail.

The budget gate reads usage using:

```
usage_day
+
provider
+
model
```

Therefore:

```
gemini / model A
```

and:

```
gemini / model B
```

have separate daily usage buckets.

For example:

```
Model A today
48,000 / 50,000 requests

Administrator changes to Model B

Model B today
0 / 50,000 requests
```

The dashboard will now show Model B's budget because Model B is the active model.

The previous Model A usage remains recorded and visible in historical/provider statistics.

This means the configured daily cap is currently effectively a:

> **daily cap per provider/model combination**

rather than one global LLM allowance across the entire installation.

Today: Active Model

The first summary card displays today's usage for the active provider/model.

It includes:

```
requests
prompt tokens
output/thinking tokens
```

For example:

```
TODAY · ACTIVE MODEL


315 requests
482,741 prompt
211,309 output/thinking
```

If the configured model was changed today, this card shows only usage for the **new active model**.

It does not combine today's usage from earlier models.

Period Token Usage

The next card shows total token usage over the selected reporting period.

For example:

```
30 DAY TOKENS
```

```
17,128,086


11,660,943 prompt
5,467,143 output/thinking
```

Unlike the daily budget panels, the historical period totals include:

> **all recorded providers and models**

within the selected period.

The total is:

```
prompt tokens
+
output/thinking tokens
```

Period Request Usage

The request card displays total requests over the selected reporting period.

For example:

```
30 DAY REQUESTS


8,139
```

It also calculates:

```
average tokens/request
average prompt tokens/request
average output tokens/request
```

The average total is:

```
period prompt tokens + period output tokens
-------------------------------------------
```

```
            period requests
```

This can be useful for spotting changes in workload characteristics.

For example, a significant increase in average tokens per request may indicate:

- larger extraction batches
- larger prompts
- more complex patient records
- increased SNOMED candidate context
- a change in model behaviour
- a configuration change

Lifetime Recorded

The **Lifetime Recorded** card displays all usage currently present in:

```
llm_usage
```

It includes:

- total tokens
- total requests
- first recorded usage day
- last recorded usage day

For example:

```
17,128,086 tokens


8,139 requests


2026-07-30 → 2026-08-24
```

The term **lifetime** means:

> The lifetime of the usage records currently retained in the Open Clinical History database.

It does not necessarily mean all Gemini usage since the installation was originally created.

If the `llm_usage` table has been cleared, recreated or introduced after the system began operating, earlier provider usage will not appear.

Usage Trend

The **Usage trend** section contains two charts.

Tokens per day

Displays:

```
prompt
+
output/thinking
```

tokens recorded on each day.

The chart automatically scales relative to the highest-use day in the selected period.

Hovering over a bar displays the date and token count.

Requests per day

Displays the total number of LLM requests recorded each day.

This is useful for identifying:

- large import runs
- SNOMED/image rebuild activity
- bursts of patient processing
- changes in processing volume

Zero-use Days

The chart deliberately includes days where no LLM usage was recorded.

For example:

```
1 Aug   0
2 Aug   0
3 Aug   4,182
4 Aug   0
```

This ensures that the horizontal timeline represents the complete selected period rather than showing only days on which activity occurred.

Provider and Model Usage

The **Provider and model usage** table breaks the selected period down by:

```
provider
+
model
```

Columns include:

Column	Meaning
Provider	LLM provider
Model	Model used
Requests	Number of recorded requests
Prompt	Prompt/input tokens
Output / thinking	Output plus Gemini thinking tokens
Total tokens	Prompt + output/thinking
Avg / request	Average total tokens per request

The currently configured provider/model is marked:

```
active
```

Historical Model Changes

If the model has changed, the table may contain multiple rows.

For example:

```
gemini   gemini-2.x-model          1,240 requests
gemini   gemini-3.5-flash-lite     6,899 requests   active
```

This makes it possible to see how much processing was performed by each model.

The statistics are not rewritten when the active model changes.

Current Provider Limits

The **Current provider limits** section displays the settings currently defined under:

Admin → Configuration

These include:

```
Daily requests
Daily tokens
Max output / request
Max prompt chars
Timeout
```

These values are read directly from `app_config`.

The dashboard does not provide controls for modifying them.

Daily Requests

Configuration key:

```
llm_daily_request_cap
```

Controls the daily request gate.

A value of:

```
0
```

means unlimited.

Daily Tokens

Configuration key:

```
llm_daily_token_cap
```

Controls the daily recorded token gate.

The total includes:

```
prompt
+
output
+
Gemini thinking tokens
```

A value of:

```
0
```

means unlimited.

Maximum Output per Request

Configuration key:

```
llm_max_output_tokens
```

Controls the `maxOutputTokens` value sent with Gemini requests.

This is **not a daily limit**.

It limits an individual model response.

If Gemini reaches this maximum before completing a response, Open Clinical History detects:

```
MAX_TOKENS
```

and treats the response as failed rather than attempting to consume truncated clinical JSON.

The resulting error advises reducing the input size or increasing the output-token allowance.

Maximum Prompt Characters

Configuration key:

```
llm_max_chars_per_request
```

This is measured in:

```
characters
```

not tokens.

Before an LLM request is made, Open Clinical History calculates:

```
strlen(system prompt)
+
strlen(user prompt)
```

If the result exceeds the configured value, the request is blocked locally.

This provides protection against accidentally sending an unexpectedly large prompt.

Timeout

Configuration key:

```
llm_gemini_timeout_seconds
```

This controls the HTTP timeout for an individual Gemini request.

For example:

```
120 sec
```

means an individual request may wait for up to approximately two minutes for the provider response before the request is treated as failed.

Transient pipeline retry behaviour is controlled separately by the retry settings under **Admin → Configuration**.

Recent Document Usage

The **Recent document usage** section displays the latest:

```
20
```

patient-history documents with recorded LLM activity.

A document is included when any of the following is greater than zero:

```
requests
prompt_tokens
output_tokens
```

The records are ordered by the document's most recent update time.

Patient / Document

Where patient information is available, the dashboard attempts to display the patient name from patient attributes.

It checks, in order:

```
display_name
full_name
name
```

If no name is available, the patient record number is used.

If neither is available:

```
Unlinked patient
```

is displayed.

Below the patient identifier the dashboard displays:

```
document ID
source filename
```

For example:

```
Example Patient


doc 142 · specialist-letter.txt
```

Document Status

The status column shows the current `history_document` state.

Depending on the workflow this may include states such as:

```
uploaded
processing
proposed
committed
failed
```

This provides useful context when reviewing LLM consumption.

A document with unusually high usage that is still processing or failed may warrant investigation.

Document Requests

The request counter represents LLM activity accumulated against that:

```
history_document
```

during clinical processing.

It can include model calls made by multiple pipeline stages rather than only the initial clinical extraction.

For example:

```
Document
   |
   +--> segmentation requests
   |
   +--> extraction requests
   |
   +--> SNOMED resolution requests
   |
```

```
    +--> body-layer requests
    |
    +--> reconciliation/audit requests
    |
    v
document request total
```

This is why a single patient document can result in many LLM requests.

Document Prompt and Output Usage

For each recent document, the dashboard displays:

```
Prompt
Output
Total
```

where:

```
Total = Prompt + Output
```

These counters are cumulative for that document's recorded processing activity.

They are useful for comparing how computationally expensive different patient records have been.

Recent Documents Are Not Filtered by the Period

Selector

The:

```
7 days
30 days
90 days
1 year
```

selector controls the historical usage statistics and charts.

It does **not** change the **Recent document usage** section.

That table always displays the latest 20 documents with recorded LLM usage, regardless of which reporting period is selected.

Why One Document May Use Many Requests

Open Clinical History deliberately separates the clinical processing problem into multiple bounded model operations.

For example:

```
Large patient history
        |
        v
Source segmentation
        |     several requests
        v
Event extraction
        |     several requests
        v
Document reconciliation
```

```
        |     one or more requests
        v
SNOMED grounding
        |     one or more requests
        v
Body mapping
        |     one or more requests
        v
Clinical audit
              one or more requests
```

A document showing:

```
60 requests
```

does not imply the document was sent to the LLM 60 times in its entirety.

It reflects the series of bounded processing operations required to construct and audit the final clinical history.

Usage and Application Budgets

The usage dashboard is not merely informational.

The values recorded in:

```
llm_usage
```

are also used by the LLM budget gate.

Conceptually:

```
New LLM request
      |
      v
Read today's llm_usage
```

```
      |
      +--> Request cap reached? ----> BLOCK
      |
      +--> Token cap reached? ------> BLOCK
      |
      +--> Prompt too large? -------> BLOCK
      |
      v
Call Gemini
      |
      v
Receive usage metadata
      |
      v
Update llm_usage
```

The budget state therefore survives:

- browser reloads
- PHP request completion
- application-worker restarts
- separate worker processes

Usage Is Recorded After a Successful Model Response

The shared `llm_usage` counter is updated after the configured LLM provider successfully returns a usable response.

This has an important operational implication.

Requests that fail before Open Clinical History receives a usable result, for example:

- HTTP errors
- transport failures
- timeouts
- safety refusal

- truncated `MAX_TOKENS` responses
- empty responses

are not necessarily represented in `llm_usage`.

Therefore:

> The Open Clinical History usage dashboard should be treated as an operational application-usage record, not as an authoritative Gemini billing statement.

For definitive provider billing information, use the billing/usage facilities supplied by the LLM provider.

No Cost Estimate

The dashboard deliberately does **not** attempt to calculate monetary cost.

Gemini pricing can depend on:

- model
- API tier
- input tokens
- output tokens
- thinking tokens
- provider pricing changes
- account arrangements

Open Clinical History does not store a pricing table.

Displaying a calculated dollar figure would therefore risk becoming misleading as provider pricing changes.

The dashboard reports the underlying usage metrics instead.

Missing Usage Table

If:

```
llm_usage
```

does not exist, the dashboard displays:

```
LLM usage table missing.
```

Historical usage cannot be displayed until the required database schema has been installed.

This does not itself establish whether Gemini is configured correctly.

Dashboard Query Failure

If the table exists but usage cannot be queried, the page reports:

```
Usage dashboard error.
```

with the underlying database error.

The remainder of the page attempts to remain available where possible.

Missing Document Tables

The usage dashboard can operate without the patient-document detail section.

If:

```
history_document
```

is unavailable, aggregate LLM statistics can still be displayed.

If:

```
patient
```

is unavailable, the document query can still operate without patient demographics.

This makes the high-level LLM usage monitoring relatively independent of the patient-history reporting tables.

Privacy Consideration

The **Recent document usage** section can display:

- patient name
- patient record number
- source filename
- document status

The page should therefore be considered an administrative view containing potentially identifiable clinical information.

Access should be restricted to authorised Open Clinical History administrators and operators.

The dashboard itself does not display the contents of the patient document or the Gemini API key.

Interpreting the Dashboard

High token usage with relatively few requests

This can indicate:

- large prompts
- large extraction batches
- extensive SNOMED candidate context
- large model responses

Check:

```
Max prompt chars
Max output / request
```

and the document-level usage.

High request count with relatively low token usage

This may indicate:

- many small source segments
- many small clinical records
- conservative batch sizes
- terminology resolution occurring in numerous small batches

Review the LLM pipeline batching settings under **Admin → Configuration**.

One document uses significantly more tokens than others

Check:

- source-document size
- number of extracted events
- extraction/reconciliation complexity
- SNOMED repair activity
- anatomical mapping activity
- audit behaviour

The document's processing status can also indicate whether repeated or incomplete processing should be investigated.

Today's budget says zero but historical usage is high

This is normal when:

- no requests have been made today; or
- the active model has changed.

The daily budget panels display only the currently active:

```
provider + model
```

while the historical period totals aggregate all recorded providers and models.

Recommended Operational Use

The Usage dashboard is useful during:

Patient import testing

Watch request and token consumption while tuning the extraction pipeline.

Large import batches

Confirm that processing is not approaching configured daily limits.

Model changes

Compare historical usage between old and new models.

Configuration tuning

Observe whether changes to:

- chunk sizes
- extraction batch sizes
- grounded candidate limits
- repair rounds
- body mapping batch sizes

materially alter LLM consumption.

Troubleshooting

Identify documents associated with unusually high model usage.

Related Configuration

The primary settings shown or enforced by this dashboard are:

```
llm_enabled
llm_provider
llm_gemini_api_key
llm_gemini_model
llm_gemini_timeout_seconds
llm_daily_request_cap
llm_daily_token_cap
llm_max_output_tokens
llm_max_chars_per_request
```

Additional pipeline settings can also materially influence overall usage, including:

```
llm_max_chars_per_source_segment
llm_max_chars_per_extraction_batch
llm_max_segments_per_extraction_batch
llm_transient_retries
llm_grounded_candidate_limit
llm_grounded_search_terms_per_event
llm_grounded_repair_rounds
llm_body_selection_batch_size
llm_body_candidate_limit
```

These are managed under:

Admin → Configuration

Related Components

Component	Purpose
`gemini_usage.php`	Read-only usage dashboard
`lib/llm.php`	Provider interface, Gemini client, budget enforcement and usage recording
`lib/app_config.php`	Runtime LLM configuration and limits
`llm_usage`	Daily provider/model usage counters
`history_document`	Per-document accumulated LLM usage
`admin_config.php`	LLM provider and budget configuration

Summary

The Gemini Usage dashboard answers four operational questions:

```
Is the LLM configured?
        |
        v
```

```
How much have we used today?
        |
        v
How much have we used historically?
        |
        v
Which patient-processing workloads are consuming it?
```

The global usage record is stored by:

```
day
+
provider
+
model
```

while patient histories also retain their own document-level usage totals.

Together these provide both:

```
SYSTEM VIEW
How much LLM capacity is Open Clinical History using?

and

DOCUMENT VIEW
Which patient-processing jobs are using it?
```

The dashboard provides visibility only.

Admin → Configuration remains the source of truth for provider settings and budget limits.

Revision #1
Created 2026-08-25 10:20:44 UTC by Admin
Updated 2026-08-25 10:21:09 UTC by Admin