

OpenEMR Weekly Meeting Detailed Summary

August 18, 2026

Attendees: Stephen Waite, Brady Miller, David Eschelbacher, Eric Stern (OCE), Michael Smith (OCE), Brad Sharp, Sherwin Gaddis, and Bobby Anchanattu (new).

backup.php — Should It Come Back?

Related forum discussion and proposed fix were posted in the meeting chat. Docker upgrade PR:

<https://github.com/openemr/openemr/pull/13624>

- **backup.php has been removed from core on security grounds.** The question raised was whether a reworked implementation could address the underlying concerns well enough to justify bringing it back.

The Argument Against Restoring It

- **The concern is structural rather than implementation-specific.** A backup, by definition, produces a complete copy of everything. Offering that as a core web feature is difficult to reconcile with the project's security posture; it is better suited to an external tool.
- **Counterpoint offered:** nearly every software package offers a full backup, including systems holding PHI — Epic was cited. Response: that is a feature of the database tooling rather than of the application, and it has not been broadly true of web applications for about 15 years.
- **The central issue is the download itself, not the code behind it.** However the feature is written, the likely outcome is that a complete copy of the data ends up sitting in a downloads folder indefinitely, on a machine that is not physically secured and may eventually change hands. Described as the one aspect that cannot be engineered away.
- **Encryption with an expiry window was suggested** as a mitigation. Immediate objection: if the file expires or is unreadable when needed, the backup has no value at the moment it matters most.
- **The wider environment was cited as the reason this matters more than it once did.** As automated tooling becomes more capable across the industry, an unprotected complete copy of a practice's data sitting on a workstation carries materially more risk than it did a few years ago.
- Noted separately that the script also breaks above a practical database size, so its usefulness is limited regardless.

The Argument For Keeping It

- **Telemetry shows approximately 10% of sites use the feature.** Characterized as not huge but still significant — enough that removing it affects a real population.
- **The users are the least technical segment** — the people who most need a one-click option. Counter raised immediately: that argument cuts both ways, because a less technical user base raises rather than lowers the diligence required of the project.
- **The failure mode if it is removed:** that population crashes their OpenEMR and has no restore file. Response: a MySQL dump has the same exposure, but the user is more purposeful about it — they follow a script and place the file deliberately, rather than letting a browser drop it wherever downloads go.
- **Position taken:** if someone is willing to maintain it and the underlying concerns are genuinely resolved, there is no objection in principle. Nobody on the call volunteered to maintain it.
- **The dependency angle:** a ticket was filed months ago for an existing PHP library that performs a database dump. Strategic argument in favor — supporting the use case without the project managing the code shifts maintenance upstream to a specialized tool with more eyes on it. Counter: investigation found it wraps the same command-line calls, so it does not remove the underlying constraint, and dependencies carry their own risks if maintenance lapses or direction changes.

The Maintainer Question

- **Raised as the higher-level issue behind this specific decision:** for any complex or sensitive component brought into OpenEMR, who commits to reviewing it on an ongoing basis? Sensitive areas need more than one reviewer, not just a nominal maintainer.
- Connected to the broader question of bringing on more admins, and whether the group collectively commits to reviewing in the areas they own. Community members willing to take on sustained review responsibility in a specific area would directly change what the project can afford to carry.
- **Resolution reached on this instance:** leaning toward not bringing it back, making the maintainer question moot for this particular script.

The Module Route

- **With the current shape of the module system, someone could write a module that does this.** A CLI module that performs backup, shared and maintained by whoever wants it, was considered acceptable — it is not core, and responsibility sits with the maintainer.
- The distinction drawn: an easier backup mechanism as a module is fine. What is being rejected is a core page that downloads a file containing everything into a user's downloads directory where it is forgotten.

Better Directions

- **Push users toward better platforms.** XAMPP stopped at PHP 8.2, so those installs cannot reach 8.3.0 — a dead end, surfaced that morning via the Telegram channel. Observation that XAMPP users and backup.php users are likely the same population.
- **A Docker migration path for XAMPP was started,** framed as solving both problems at once — it unblocks the upgrade and moves them onto a stack where backups can be handled properly, ideally off-site.
- **Existing tooling:** there is no restore.php, but there are backup and restore bash scripts. They are old and their current state is unverified, though the architecture has not changed enough to expect breakage. Asher's MySQL backup in the DevOps repo was refactored to MariaDB tools; it is not tied to any OpenEMR deployment.
- **Documentation plus AI as the realistic answer for non-technical users.** Proposal: add a short section to CLAUDE.md covering the basics — the traditional backup is a database dump plus the entire OpenEMR directory. That population is already using AI for everything, as visible on the forums. Noted upside: AI can also test the backup by restoring it, which is more than most one-click users ever do.
- **Caveat on restore:** restore requires the command line. For a system in clinical use, someone in the chain of command — staff or contractor — needs to be capable of performing it. Downtime in an EMR is not a cosmetic problem: inability to reach charts is a patient safety issue, and that point was made emphatically.
- **Developer practice noted for contrast:** the Flex capsule approach is a database dump plus the sites directory only, since code base changes are not a concern. Production backups need the full directory, and a restored database has to line up with the code base it is going into — there is a command to run the SQL upgrade to bridge that.
- **Health information exchange raised as a better long-term direction** than a backup page — FHIR and CCDA/CDA export serve as a transfer mechanism, useful for import and export generally, including migrating in from another EMR. Acknowledged as needing more than one meeting to work through properly.

New Attendee: Bobby Anchanattu

Forum thread:

<https://community.open-emr.org/t/feedback-requested-generic-oidc-login-and-ai-integration-for-openemr/26915?u=stephenwaite>

- **Background:** 20+ years in enterprise software, currently at SAP, with several personal projects. Came to OpenEMR while building AI enterprise infrastructure and looking for a proof of concept.
- **What he has built:** an external identity provider integration — OpenID Connect against an external IDP — which he has working. In his setup only the ID token from the external IDP is reused; all authorization remains inside OpenEMR. The changes currently sit in a private repo, and he offered to contribute them if there is interest.
- **Why:** OpenEMR currently handles authentication and authorization entirely through its own identity provider. He wants an external IDP so that OpenEMR and his AI infrastructure work together under one identity.
- He noted a reply on the thread from Andy, approaching from the opposite direction — using OpenEMR itself as the IDP, with work on voice dictation.
- Acknowledged on the call that the thread took a while to get a response.

The AI Use Case

- **Prototype scope:** a clinician logs into OpenEMR and uses an AI tool to ask about a patient — current medications, interactions, chart summary. Also demonstrated: booking an appointment, adding a patient, adding clinician notes. He noted the clinical domain itself is new to him.
- **Strong interest expressed from the clinical side.** Epic already does automatic chart summaries on opening a chart — brief, accurate, and tailored to who is logging in. Described as very useful and slightly unnerving.
- **The back-and-forth conversational model was noted as less common in EMRs**, with the observation that logging requirements may be why. Existing products tend to place AI at specific points — summaries, note generation — rather than open-ended chat.
- **His design uses skills scoped by role** — a different interaction skill for a clinician than for front office or customer relations.
- **Model agnostic:** the infrastructure supports Gemini, Claude, and OpenAI, and can run on the operator's own infrastructure. Connection between OpenEMR and the LLM goes through an MCP server.
- **Placement:** confirmed as a module rather than core. His implementation is a Spring Boot service — chosen because security was easier to implement there — interacting with OpenEMR over standard protocols and the FHIR APIs.

PHI and LLM Vendors

- **His question:** what legal, regulatory, or ethical restrictions apply to sending patient information to an external LLM?
- **Answer given:** in the United States this is HIPAA — PHI has to go through business associate agreements. The starting point is confirming that the companies involved will sign as business associates, with a contract covering how the data is handled.
- **Practical detail shared:** a BAA is obtainable with Claude but only for API access, and it is not given readily. An alternative route is going through a cloud provider — a BAA with GCP applies to all Google models accessed through their agent framework, including Claude. Zero data retention is a condition of these arrangements, which is why they are not available on subsidized plans.
- Bobby noted a contact using a zero data retention agreement with OpenAI at Delta Dental, though as an insurer rather than a practitioner.
- **His product includes data sanitization before anything leaves**, but he flagged the open question of whether an outside party could still correlate sanitized information — an area he intends to work on further.
- **Framing offered:** a public AI is like a friend who cannot keep a secret — do not tell it anything you would not want public. The practical direction is leaving the door open for operators to plug in their own zero-retention models.
- Noted that a disclaimer would be needed making clear this is not for use with a general consumer chatbot, and that requirements vary by country.

Open Evidence

- **Raised as a reference point:** an AI reference tool for physicians, free with account verification, advertisement-supported with light ad load, and offering CME. It holds licensed journal literature through agreements with multiple publishers, which reduces hallucination relative to general models. Described as comparable to UpToDate but different in kind.
- Cited as reportedly used by over 50% of physicians per the New York Times, and unfamiliar to several people on the call.
- Distinction drawn: Open Evidence is a general medical reference and has no access to the medical record, so it cannot produce a chart summary. Bobby's work is scoped to a specific OpenEMR installation.
- Adjacent observation: AI scribe services are already in use, with clinicians carrying smartphones into encounters. Having a companion capability inside the EMR opens up considerable further ground.

Sinch Fax Module

Module: <https://packagist.org/packages/opencoreemr/oce-module-sinch-fax> — provider: <https://sinch.com/voice/fax-api/>

- **It was not designed as a generic fax abstraction** — it was built specifically for Sinch. Stated clearly in response to the question of whether it could serve as the basis for a general solution.
- **How it would be designed as a generic solution:** roughly three layers, one being the interface. The genuinely useful piece would be a consistent interface within OpenEMR for document import and export regardless of source — fax, emailed PDF, or anything else — while still carrying source-specific attributes such as the originating phone number and the ability to move documents between patients. Those design questions do not depend on the fax provider.
- **Licensing:** GPL, so anyone wanting to use it as the basis for a generic fax capability is free to do so.
- **Interface:** built from the ground up, described as having a more modern look than the existing fax/SMS module, with the caveat that aesthetics are subjective. Screenshots are in the README on the GitHub repo, showing settings and a recent-faxes queue with direction.
- **PHP version documentation issue found during the call:** the package metadata states PHP 8.5 or later, but it was rolled back to 8.4 and the text was not updated. It has to work on 8.4 in practice, because Composer will not install into an OpenEMR environment without 8.4 support without significant workarounds — a consequence of the Zend dependencies.
- **Module style:** written before much of the thinking on the new module style, and it was noted that there is not really a new style yet — which led directly into the next topic.
-

Module System, Front Controller, and the Demo Farm

- **The status of the new module system was raised and deferred to next week** as a topic in its own right.
- **The front controller is the blocker on any practical rollout of the new module system.**
- **Three interrelated deprecations:** multisite, the front controller, and running OpenEMR out of a subdirectory. A number of decisions were made previously and are simply pending implementation.
- **Draft PRs exist, including one on the demo farm update,** intended to build awareness of the agreed deprecation plans and get them implemented. Follow-up expected in a week or two given current time constraints.
- **The demo farm specifically:** an up-for-grabs front controller demo was attempted but requires a PR to work with the demo farm, because of where instances are served from — root versus a subdirectory. Some form of shim would be needed to run it as-is.
- **The likely solution:** move instances to subdomains rather than paths — e.g. 1a1b.openemr.io instead of a subdirectory. Considered solvable, but it means a batch of new domains, each carrying AWS domain configuration. There are currently around 11 instances, and this would roughly double the domain configuration work. Possible candidate for automation.
- **htaccess handling:** the front controller is designed to handle all of the htaccess styles present in core, but modules installed via Composer would not be handled well, so follow-up work is expected there.
- Offer made to review specific notes on limitations if they are gathered up, with the caveat that most of it is still in working memory rather than written down.

Release and Docker

- **The recent release went well.** A number of bugs had to be sorted out during the process, but the next cycle should be considerably faster as a result.
- **Positive comment on the new acceptance testing:** every artifact now goes through functional testing.
- **Docker work deferred to next week.** Originating from an AWS upgrade that went awry — the accumulated changes since 7.0.2 caused problems, and the fixes are being pushed upstream so others do not have to hotfix the same issues.
- **PR 13624** is largely protective — guarding the upgrade path rather than adding capability. Part of the difficulty is that some deployments may not have Docker configured correctly to begin with, since it relies on shared volumes.
- Release is going back on the agenda for next week alongside the Docker items.

Items Discussed and Proposed

Recorded as topics raised and directions favored, not as assigned work or commitments.

Area	Discussed / Proposed
backup.php	Leaning toward not restoring it to core. If someone wants the capability, a CLI module maintained outside core was raised as the acceptable route.
Documentation	Add a short section to CLAUDE.md covering how a backup is actually made, so AI-assisted users can be walked through it. Two steps: database dump plus the OpenEMR directory.
Documentation	Verify whether the existing backup and restore bash scripts still work, and make them current if so.
XAMPP migration	A Docker migration path for XAMPP users, prompted by XAMPP stopping at PHP 8.2, leaving those installs unable to reach 8.3.0.
Docker	PR 13624 — protections around the upgrade path, pushed upstream from an AWS upgrade that went awry.
HIE / FHIR	Raised as a better long-term direction than a backup page — FHIR and CCDA export as both a transfer and an import/export mechanism. Noted as needing more than one meeting to work through.
External IDP	Bobby Anchanattu offered to contribute his external IDP / OIDC changes if the project is interested. Currently in a private repo.
AI module	Direction indicated: a module rather than core. Bobby's infrastructure is Spring Boot, interacting with OpenEMR over MCP and the FHIR APIs.
Demo farm	Front controller demo blocked on serving path — likely needs per-instance subdomains rather than directories, plus AWS domain configuration and htaccess handling.
Next week	New module system status, front controller, multisite, and running from a directory. Release back on the agenda. Docker items.

Notes

- Recurring theme: the tension between accessibility and security ran through the entire backup discussion. The project's historical position has been to make installation and operation as easy as possible; the emerging view is that for anything touching PHI, the easy path may itself be the risk, and the better answer is moving users onto platforms where the right thing happens by default.