

# OpenEMR Weekly Meeting Summary

July 28, 2026

---

**Attendees included:** Stephen Nielson (chair), Stephen Waite, Michael Smith (OCE), Eric Stern (OCE), Char Miller, David Eschelbacher. Brady Miller referenced but not present.

## Release Status

---

- **Current release: 8.2.0**, out roughly two to three weeks ago (~7/8/2026).
- **Next release: 8.3.0**, targeted for tomorrow (7/29) — a large security release with breaking changes.
- Details of the specific security fixes were held for the release call rather than discussed in the general meeting.
- **Why 8.3.0 and not 8.2.1:** the project is holding at major version 8 because of ONC certification, but otherwise follows semantic versioning as closely as it can. Point releases are for bug fixes; 8.3.0 signals new features and breaking changes. With no major bump available, breaking changes land in the minor.
- **Release automation** is now doing most of the work — the last several releases have been largely automated, with a couple of loose ends surfacing each cycle.
- Noted that downstream deployments with contractual update windows need lead time after a release, so predictable timing matters more than speed. Agreed not to delay the release on that basis.
- **Longer-term goal:** move the community to a weekly update cadence, then eventually daily security patches, to burn down the security backlog. Reducing upgrade friction is an ongoing workstream.

## Modernizing OpenEMR: The Front Controller

---

This was the main discussion. Framing: the two gating issues for modernizing OpenEMR are the module architecture and the front controller — and the module architecture depends on the front controller, so the front controller comes first.

### Current State

- The front controller has technically been in place since 8.1.0 and is functional.
- It handles dispatch and routes requests through to existing legacy pages. No changes to old code are required.
- It is fully backwards compatible. Operationally, adoption is just a web server config change from X to Y — everything keeps working.
- **Nothing currently forces anyone to use it**, which is the core problem.
- Front controller PR: <https://github.com/openemr/openemr/pull/9943>

### Why It Matters

- **Single file entry point.** Today anyone can drop a PHP file into `interface/` and it can do anything — reinvent DB connection strings, skip globals, bypass conventions. This has been seen in code introduced into OpenEMR instances (not core).
- **Eliminates entire classes of bugs, not individual bugs.** Missing CSRF checks on a single page, missing auth checks, and similar GHSA-queue issues disappear when 100% of requests are dispatched through one path. Compared on the call to switching to a memory-safe language.
- **Autoloader and namespaces.** Moving code into `src/` with proper namespaces replaces the large amount of code dedicated to computing web root and doing `require_once` — that is the autoloader's job, currently duplicated everywhere. Doctrine, Symfony filesystem, and Twig templating are simply available.
- The framework tooling largely exists already; most developers do not know it is there or where to find it.

## The Catch — Benefits Are Gated on Universal Adoption

The centralizing wins (CSRF middleware in the front controller, stripping thousands of scattered CSRF calls, simplifying how PID is passed around) cannot be taken until every deployment is on the front controller. The goal of enforcement is to make what would otherwise be breaking changes non-breaking, because everyone is already on the new path.

## This Is a Communication Problem, Not a Technical One

Consensus: there is no technical blocker — the gap is developer understanding and documentation.

- The front controller pattern is familiar to developers who work with modern PHP frameworks daily, but OpenEMR draws contributors from a wide range of backgrounds. The term alone does not convey why the change matters, so the rationale has to be spelled out.
- Stephen Waite asked for a clearly articulated rationale that can be pointed to when pushback arrives later. PR 9943 states the pros have been “discussed at length” but does not restate them.
- Proposed: a written explainer on the forum with links to external resources on the pattern, possibly a 10–15 minute video, and a README/CLAUDE-style snippet so AI coding agents follow the standard by default.

## Proposed Enforcement Path

1. Documentation of the web server config changes.
2. Community announcement and an open feedback window, so anyone with a blocking concern has a clear opportunity to raise it.
3. A PHPStan rule flagging new PHP files outside `src/`, pushing new work toward API endpoints or controllers. Side benefit: makes AI coding agents reticent to generate stray PHP files.
4. Deprecation notice that logs (or optionally throws) and points to a forum thread or issue for objections. Guarding for this already exists in the front controller.
5. A dated announcement: from this point onward, the code standard requires the front controller.
6. Model it on PHP's own deprecation process — warn-only deprecation in 8.4, hard failure by 8.6, with the timeline published in advance.

## Impact on Operators

- Non-Docker operators must update their Apache/Nginx configuration.
- The change is roughly 10 to 20 lines and simpler than the current config, not harder.
- Docker-based deployments remain the recommended path and will get the change automatically. For others, clear public documentation of the required config should make the translation straightforward.
- **Open question raised by Eric:** is the messaging targeted at developers, operators, or both? These are different audiences with different asks.

## Multisite Deprecation

---

- The front controller does not support multisite, and the admins previously decided to deprecate multisite over security concerns — but it is unclear how widely that decision was communicated. A prior forum post drew no responses either way.
- Simplest framing proposed: state plainly that the front controller does not support multisite, then give a warning period of several releases where both legacy routing and multisite remain supported.
- Related note: the API requires the site ID in the URL (`api/default/api/...`), whereas `interface/` carries it through session — a friction point for front-end integration.
- RFC to deprecate multisite: <https://github.com/openemr/openemr/issues/11387>
- Require running in document root (likely needs demo farm updates): <https://github.com/openemr/openemr/issues/12690>

## Demo Farm and Docker

---

- The demo farm is not currently configured to use the front controller, nor is the standard Docker image.
- The demo farm does run Docker and uses web subdirectories, not multisite. Updating the Docker image to use the front controller should propagate automatically.
- Existing web server config PR: <https://github.com/openemr/openemr/pull/11550> — still in draft, and now badly out of date given Brady's recent Docker overhaul (expect heavy merge conflicts).
- **Agreed:** stand up an “up for grabs” demo running front controller routing so problems surface early. Eric will patch the relevant Nginx config out of 11550 into a fresh PR and likely abandon the old one; OCE is busy with an enterprise customer, so no hard deadline.
- Integration tests already run against the front controller, though module code coverage is uncertain.

## Next Week's Agenda: Docker Refactor

---

Michael proposed consolidating the current sprawl of Dockerfiles and Compose files into a single root-level setup using Docker Compose profiles:

- Default “docker compose up” gives you a production configuration (no phpMyAdmin in prod).
- COMPOSE\_PROFILES=development adds Mailpit and the dev services.
- One hardened base image doing the minimum, rather than trying to support many development workflows.
- Puts the Dockerfile and Compose file at the repo root, which is the native convention — historically avoided because people started running the dev-convenience setup in production, though the project now does want Docker used in production.

Received enthusiastically; confirmed as next week's agenda item.

## NSF I-Corps Update (Char)

---

- The first NSF I-Corps seminar begins shortly.
- Scheduling at least 10 customer interviews for the coming week, with 20 required by the 13th. These are introductory interviews.
- The national cohort is expected around October–November, running seven weeks, and will require more in-depth interviews.
- Plans to target community-based organizations and tech shows for participants.
- Outreach continues to contacts in the UK and Ireland, where work on system design and clinical workflows is further along.
- An interview with Stephen Nielson is to be scheduled for the coming week.

## Action Items

| Owner           | Action                                                                                                                                                                                                 |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Stephen Nielson | Write the community forum post/announcement on the front controller: benefits, rationale, external resources, and the direction the project is taking. Possibly record a 10–15 minute explainer video. |
| Eric Stern      | Surface existing foundation work already completed in GitHub so the wider community can see it.                                                                                                        |
| Eric Stern      | Extract the Nginx config from PR 11550 into a fresh PR and stand up an “up for grabs” demo running front controller routing.                                                                           |
| Team            | Write documentation for the Apache/Nginx front controller configuration.                                                                                                                               |
| Team            | Add a PHPStan rule / deprecation notice flagging new PHP files outside src/.                                                                                                                           |
| Team            | Decide and publish the enforcement deadline (proposed: warn-only in 8.4, hard failure in 8.6).                                                                                                         |
| Team            | Communicate the multisite deprecation decision publicly.                                                                                                                                               |
| Char Miller     | Schedule 10+ NSF I-Corps interviews for the coming week; reach 20 by the 13th.                                                                                                                         |
| Next week       | Docker refactor discussion — consolidating to a single root-level Compose file using profiles.                                                                                                         |

## Open Questions

- Is the front controller messaging aimed at developers, operators, or both?
- What is the actual population of operators running current-but-non-Docker installs who would be affected?
- Where does the demo farm transition sit in the timeline relative to the 8.4 soft deprecation?