

OpenEMR Weekly Meeting Summary

July 21, 2026

Attendees included: Stephen Nielson (chair), Michael Smith (OCE), Eric Stern (OCE), Stephen Waite, Char Miller, Sage (Char's son, first call), David Eschelbacher

NSF I-Corps Program (Char)

- **Char was accepted into NSF I-Corps.** Congratulations extended by the group.
- The cohort was offered for September, which would have given a longer runway, but Char opted for the earlier one to keep momentum with contacts already lined up.
- **Hard window: interviews cannot begin until August 3 (after the first class) and must be finished by August 13 (the last class).**
- Scheduling can happen now, so Char is lining up appointments in advance to avoid losing days at the front of the window.
- Sage will help take some of the phone interviews.

Interview Targets

- San Francisco community-based organizations — homelessness, addiction recovery, and referral care. Large organizations such as Glide were named as likely candidates.
- Char expects to gather three or four people at a single clinic or CBO (Community-based Organization) in one visit (referral care coordinator, social worker, and similar roles), potentially knocking out a batch in one afternoon and evening.
- Caveat noted: some CBOs Char knows are not actively handling prescriptions, so fit needs to be checked.
- Maxine — clinical director for OpenCare IT who handles clinical outreach.
- David agreed to a 20–30 minute clinician interview and shared contact details; text before calling is the working protocol.

New Participant: Sage — Onboarding

- Sage is Char's son, a recent SF State biochemistry graduate, joining to help with support requests and development work coming through the forums.
- Onboarding path discussed: the OpenEMR YouTube channel playlist Brady assembled (roughly seven videos), which walks through environment setup, VS Code, and a first PR end to end.
- The written companion is titled “You can be an OpenEMR developer in five easy steps.” The VirtualBox/Ubuntu portion is out of date and can be skipped; the rest, including the Docker and editor setup, is still useful.
- After a first PR, the “good first issue” labels in GitHub are the next step — Michael added one on June 1.
- Standing offer for Sage to give a weekly update slot on this call.

Release Automation and Security Release Strategy

Where Things Stand

- **Release automation is essentially finished and ready.** The project can now run an effective release-test-release-test cycle.
- **What that surfaces is the security backlog: roughly 100 reported GHSA's**, some potentially critical if they validate. Several reporters are getting impatient.
- Fixes so far have been reviewed manually, merged, and hoped for the best — the confidence gap is the real problem.

The Disclosure Timing Problem

- Once a fix is merged, the vulnerability is effectively public. The more fixes brought in at once, the greater the chance of a bug.
- **A public vulnerability cannot sit unreleased for an extended testing window. Target is one to two days between the fix PR and the release**, then announce the GHSA and make it public.
- Most of these fixes are simple enough that review can confirm they will not create new problems.
- **Brady proposed handling some fixes publicly in master** where severity is genuinely low, skipping the private-fork overhead entirely. High-severity issues still go through the responsible private process.

Reporter Recognition

- Reporters are security researchers following the responsible disclosure program — research firms, students, and paid professionals. They do not receive payment; some have asked, and the answer has been no.
- The incentive is reputational: a published CVE/GHSA credit they can write up and advertise. That is why the most aggressive reporters are pushing hardest for publication.
- Public fixes should still count toward their GHSA as long as the advisory is published and a CVE is obtained — the private fork is not what earns them credit.

Cadence

- The long-standing goal is weekly releases, with critical vulnerabilities addressed within about 24 hours. No official start date has been set; automation was expected to take a couple of weeks and has taken months.
- **Ireland would prefer a two to three week delay before the first security release wave**, given other project deadlines. Whether that is possible depends on how long the researchers will hold.
- Expectation voiced: the more critical items likely need to go out in the next week or two.

Ireland Deployment — Context for the Update Pressure

- **Seven years in production** with the Irish government, covering three of seven regions of the homeless and social inclusion program — Dublin and the two adjacent regions, roughly the most populated third of the country. Serves immigrants, refugees, and the homeless population.
- **Growth: from about 1,800 patients and two staff to 25,000 patients and 58 providers.**
- **Contractual obligation to update within seven days of a publicly disclosed critical or severe vulnerability** — and they are not currently on a release cadence that supports that. They are on 8.0 and need to reach 8.2, with several modules requiring updates first.
- **Running two EMRs side by side with double data entry.** Socrates (owned by Clanwilliam, roughly 95% of the Irish market) is the government-certified system for e-prescribing and lab ordering; OpenEMR holds the designated record set and the clinical content.
- Certification is in progress to reach the government Health Information Exchange network for referrals, prescriptions, and lab ordering — the step that would remove the need for the second system.
- A competing government-sponsored EMR is going through the same certification process but is not expected to be in place until roughly 2031–2032. Effectively a race to certify first.
- The group funded \$50,000 of last year's ONC certification and has contributed financially and in code for seven years.

- Expansion underway into the United Kingdom social inclusion market. Maxine Radcliffe, who ran London's social inclusion, homeless, and refugee programs, was hired over into Ireland and brings the UK contacts; Pete is based in the UK.

CI and Test Infrastructure for Security Fixes

The Gap

- **Private forks do not run the automated test suite.** OCE can run tests internally, but then everyone else has to take their word that the tests passed.
- A private OpenEMR-owned repository would run Actions, but private repos are billed, and the OpenEMR suite is expensive to run.
- **Cost estimate: at least \$300/month, roughly \$3,600/year.** During the large GHSA batch in March and April, OpenEMR work pushed OCE to the ceiling of their enterprise plan — builds can run multiple machine-hours each, with test runs around 45 minutes in parallel.

Options Discussed

- **Sponsorship.** Find one or several community members to cover roughly \$3,600/year. Stephen will ask whether Ireland can budget a donation to the foundation.
- **Trim the test matrix.** Seven databases multiplied by four or five PHP versions, plus Windows, is extensive but marginal in value. Reasonable to run it all while GitHub is paying; not reasonable if OpenEMR is paying.
- **Optimization.** PHPStan scales exponentially with function length, and OpenEMR has a few enormous functions (the EHI export functions, options.inc.php). Splitting those would make PHPStan dramatically faster — and Rector and the rest of the toolchain along with it. An upstream PHPStan fix was contributed but only bought one or two percent; the underlying issue is that PHPStan has no test cases for god classes and god function files this large.
- **Dedicated or self-hosted runners.** Open Source Lab could provide hosting that GitHub self-hosted runners attach to. Lance had assigned someone to it, then GitHub started charging for hosted runners and he redirected resources; it was never picked back up. Even OpenStack resources would be enough to stand up runners.
- **Batching.** Put a set of fixes on a private fork addressing many GHSAs at once and run CI once across the whole set. Treats the symptom, not the cause.
- **Free open source infrastructure programs** are worth surveying — compute-hour grants for qualifying projects.

Self-Hosting Caveats

- GitHub runner images are VM images, not Docker images. A self-hosted runner needs an image as close to GitHub's ubuntu-24.04 as possible or results get weird.
- Running in Docker was tried and did not go well; bare metal or a VM is the realistic path. nektos/act is not good enough to substitute.
- Michael wants to hand the Open Source Lab relationship to a named owner rather than carry it — Asher (the infra person) or Brady are the likely candidates.
- **Stated goal of all of this: better assurance around the project's security posture.**

Getting Users to Actually Update

- Roughly 90% of OpenEMR installs are not updated regularly — Stephen Waite noted he does it about once a year. Large enterprise users update quickly; nearly everyone else does not.
- Position taken: the project's obligation is to make upgrading as easy as possible and to disclose responsibly once a fix exists. Beyond that it is on the operator.
- Blunt framing offered: running PHI on an unpatched system is not the responsible thing to do, and in the United States an IT administrator who knowingly declines to patch a critical vulnerability can face criminal negligence exposure, which carries jail time rather than fines.
- Real barriers acknowledged: installs still on OpenEMR 6 running XAMPP have signed up for a hard upgrade; heavily customized 7.x deployments could face tens of thousands of dollars to either remove customizations or bring them current; some modules broke between 8.0.x and 8.1, blocking production servers outright.
- Pressure is increasing — AI makes vulnerabilities easier to find, and anyone can run models against an open source codebase. Not every finding gets reported; some get exploited or sold.

Ideas Raised

- **A module that puts an out-of-date warning in front of IT admins at login** — not in front of clinical users — stating how many GHSAs have been publicly disclosed and that an update is needed. Nielson wants to build this.
- A management utility running outside Docker that offers one-click updates.
- **Webinars and blog posts** that acknowledge how hard and painful the upgrade is while making the case plainly: here is what you face if you do not, and here is how to migrate from bare metal to the Docker image. Proposed as an agenda item for next month.
- A culture shift: encouraging operators to contribute customizations upstream and to run automated tests against their own customizations, rather than paying for private work that then blocks every future upgrade.

Module Architecture: Async vs. Sync

Where Eric's Proposal Stands

- **Event handling was intentionally left out of scope for V1** — not because removing the event bus is a goal, but because it was not relevant to the first components. A linear model would be more transparent and easier to reason about, but Eric is not opposed to keeping events.
- **All existing module code continues to function exactly as it does today.** The new plugin layer sits in front: it checks whether a request targets one of the new plugins, and if so dispatches and stops. Everything else flows through the existing code paths and responds to events as it always has.
- Recognized that many module developers are actively leveraging the event system for customizations — whether it is the best mechanism is a fair question, but it is in wide use.

Nielson's Concerns

- **Plugins load too late in the lifecycle.** Many existing modules hook in during global initialization and respond to events such as patients or services being saved. If hooks cannot be introduced early, those modules cannot migrate.
- **No appetite for three or four module systems.** The goal is to migrate both the Laminas modules and the custom modules onto the plugin architecture.
- **Each new approach pushes the migration pathway further out.** The more areas that change independently, the harder migration becomes down the road.
- Proposal: move the plugin lifecycle higher, and provide an additional adapter on top of plugins exposing an event dispatcher contract, so legacy modules have a migration route while still taking advantage of Doctrine. Eric's response: at face value that sounds reasonable, but it needs specifics.
- The FHIR Subscription API was raised previously as a possible mechanism. Eric has not had a chance to follow up and would not call it validated.

What Existing Modules Actually Use

- Patient save is the big one; there are smaller ones around appointments and services.
- A large share is rendering hooks — which Nielson would rather handle by overriding a template file than through an event.
- Script and asset output, where modules inject additional scripts.
- Menu changes and globals initialization — possibly better served by a manifest file than by events.

Doctrine Events

- **Doctrine runs its own event manager, deliberately decoupled from the Symfony event dispatcher.** They look similar but are two separate systems with no clean way to bridge them.
- The current prototype uses Doctrine events only for automatic field values — UUID autofill, updated-at timestamps.
- Direction agreed: the hook should be a general entity-save event, not patient-specific, so encounters, claims, and anything else are covered.
- **Modules will need to be able to respond to Doctrine events, so those have to be exposed.**
- Performance case raised: batch importing 10,000 CCDAs should not run hooks in serial. Doctrine supports a transaction-type mode that defers or suppresses events, though suppressing carries risk — deferred execution is the safer pattern. Doctrine's unit-of-work fires post-commit with the full change set, which likely covers this already.
- Longer-term design from earlier discussions: backend plugins separate from a front-end unit handling menus and UI forms — define the APIs in one place and the UI that talks to them separately, for a cleaner separation of concerns.

Front Controller as the Prerequisite

- **Nielson is comfortable bringing the plugin architecture in as it stands**, but will not be able to migrate current modules onto it for some time until the outstanding issues are addressed.
- **Consensus: certain OpenEMR features cannot be implemented effectively until everyone is on the front controller.** The only reliable way to introduce hooks early in the request is in the front controller itself.
- `globals.php` has been serving as the de facto front controller for modules — acknowledged as a poor implementation.
- The front controller has been committed for months with unit tests and substantial testing; the remaining work is getting the default web server configs updated to route through it. With Brady's Docker work now in OpenEMR, making those configs automatic for Docker users is not a big leap.
- A hook already exists that can either warn or forbid based on a runtime setting. The dev farm could be flipped to hard error to test what breaks when the non-front-controller path is unsupported.
- **Agreed: next week's topic is a strategy for sunseting the non-front-controller run model.**

Action Items

Owner	Action
Stephen Nielson	Carve out a day to work through the OpenEMR issue and PR queue and get a lay of the land on the last month and a half of work.
Stephen Nielson	Ask the Ireland group whether they can budget a donation toward CI infrastructure (roughly \$3,600/year).
Stephen Nielson	Build the admin-facing module that flags out-of-date installs and published GHSA's at login.
Michael Smith	Restart the conversation with Lance at Open Source Lab about hosting for self-hosted runners, and hand the relationship to a named owner (Asher or Brady).
Michael / Brady	Settle the low- vs. high-severity handling split and begin working down the GHSA backlog.
Eric Stern	Skim the modules committed in core and categorize which events they actually rely on.
Eric Stern	Assess moving the plugin lifecycle earlier and adding an event-dispatcher adapter for migrating legacy modules.
David	Schedule a webinar and/or blog post series on why and how to move to Docker and stay current.
Char Miller	Schedule NSF I-Corps interviews now; conduct between 8/3 and 8/13. Target San Francisco CBOs, Maxine Radcliffe, and David.
Community	Split the oversized functions (EHI export, options.inc.php) to speed up PHPStan, Rector, and the rest of the toolchain.
Next week	Strategy for sunseting the non-front-controller run model of OpenEMR.

Notes and Carry-Overs

- **Multisite deprecation to be discussed next meeting.**
- No official date has been set for starting weekly releases or the first security release wave — flagged as needing a decision at the next meeting.
- A named owner is still needed for the Open Source Lab / infrastructure relationship.