

OpenEMR Weekly Meeting Detailed Summary

September 1, 2026

Attendees: Brady Miller, Stephen Nielson, Stephen Waite, Michael Smith (OCE), Eric Stern (OCE), Sherwin Gaddis, Luis M. Santos (MedicalMasses), Ian McDonald (MedicalMasses), Kelsey Stephenson (MedicalMasses, new), Ravi Kafle, David Eschelbacher.

Meeting Platform and Recording Logistics

The Problem

- **Zoom chat only displays to people present when a message is posted.** Anyone joining late does not see the recording announcement, so it would have to be re-posted repeatedly. No pin function was found.
- **The call runs on a personal Zoom account,** which makes one person a bottleneck for recordings and file access, and stores recordings in a personal cloud account. Agreed that an organizational account would be better.
- A basic organizational plan was estimated at roughly \$10–20 per month, though the comparison favored using an account the foundation already holds.

Google Workspace as the Alternative

- **The foundation already holds a Google Workspace nonprofit account,** so moving the meeting to Google Meet was raised as the obvious option — shared access, no single bottleneck, and files available to the group.
- **Confirmed on the call:** the nonprofit plan does not impose the 60-minute limit that applies to free consumer accounts, and includes roughly 100 TB of pooled storage.
- **Key limitation:** the no-cost nonprofit tier does not include native meeting recording or transcription — those require an upgrade to a discounted paid tier. So the current workflow of capturing the meeting and running summarization separately would continue.
- **From the supporting research circulated:** the free nonprofit tier gives 150 participants, up to 24-hour meetings, and up to 2,000 users at \$0. Business Standard is roughly \$3.50–\$4.20 per user per month and adds meeting recording.
- **The pricing trap identified:** free and paid users cannot be mixed on one domain, so enabling native AI note-taking for one or two people would require upgrading every user on the domain. For a small number of note-takers, a third-party meeting assistant sitting on top of the call is far cheaper than upgrading the whole organization.
- **Practical concerns raised:** whether host-key handoff works the same way, and protection against uninvited participants — both assumed present but to be verified with a test meeting.
- Expected transition friction: people will join the Zoom out of habit for a while, so two meetings may run in parallel briefly. Considered acceptable.
- Brief tangent on whether a generic room could bridge Zoom and Meet, and on open standards such as Jitsi — previously looked at for telehealth infrastructure. Not pursued.
- One caveat voiced: comfort with moving to Google provided nothing the foundation does triggers an automated account closure.

Business Development — The Mental Health RFI

<https://vendor.myfloridamarketplace.com/search/bids/detail/16676>

Assessment of Fit

- **Inpatient is a significant weakness for OpenEMR** and the RFI is centered on it. Direct assessment given: it is unclear whether a bid would be competitive in that space.
- **Scale is the core problem.** The RFI describes bed management for roughly 3,000 inpatients. The largest OpenEMR inpatient deployment anyone on the call had heard of was 100–200 beds. Scaling to that level was called a tall order and logistically demanding.
- **What exists versus what is missing:** inventory management, prescriptions, and some pharmacy capability are present. Bed management, triage, the associated inpatient reporting, and alerting do not exist and would all have to be built.
- **The facility is a residential mental health and substance abuse hospital.** Comparable residential facilities have been done with OpenEMR through customization, but none of that work exists in the mainline product.
- **The bid would effectively be the cost of turning OpenEMR into what they need,** which was judged likely to exceed what they intend to spend. The upside noted: if won, it would bring inpatient capability into OpenEMR.
- **An 11-day response window** puts it out of reach for those on the call who might otherwise have engaged.

The Foundation's Role

- **The foundation would not be the prime bidder.** It does not deliver healthcare systems or build solutions directly — it refers vendors out. That position was stated clearly and confirmed.
- **Historical precedent:** a consortium of vendors was once assembled to submit a bid, described as a hail-mary attempt and not something anyone wants to repeat.
- **What the foundation can do:** provide letters of support and general backing. A vendor or vendor consortium should hold the bid itself.

Vendor Referrals

Professional support vendor list: https://www.open-emr.org/wiki/index.php/Professional_Support

- **Sherwin Gaddis** has done some inpatient work but not at this scale, and does not have availability.
- **Other vendors with large bed management experience were discussed.** The project's general practice is to refer first to vendors who contribute back to the community, though there is capable talent in the wider ecosystem to collaborate with.
- **An existing community PR covering inpatient functionality** was noted as worth reviewing by anyone competing on the proposal, with the caveat that it is AI-generated and has not been clinically validated.
- **CapMinds** noted as a long-standing tier-one vendor. Historical note: the list once required a periodic commit to remain listed.
- **Suggestion to Luis:** list MedicalMasses on the support vendor list at the highest tier and advertise the specific areas they target, so opportunities come to them directly and the project can pass leads on.

Where OpenEMR Fits Instead

- **Outpatient is the strength** — particularly organizations that want to own their own data and workflows. Small clinics fit well.
- **The sweet spot described:** practices too large for simple products but not large enough to spend Epic money, especially in behavioral health, that need a level of customization other solutions do not offer and are willing to invest development time.
- **International data ownership is a growing driver.** Contracts are being lost by Epic and Cerner around the world as organizations move away from dependence on a US monopoly.
- **Guam was revisited** — a 100–120 bed hospital, more tractable than the RFI. MedicalMasses is trying to reconnect and may bundle their in-house radiology platform. Stephen Waite flagged as a useful contact for radiology billing domain expertise if that goes forward.

- **Ian described a comparable NIH engagement:** automating the large phantom MRI ACR accreditation process by extending an existing open source project and adding their reporting infrastructure, making measurements automatic and standardized across five scanners with a design that should scale further.
- Luis noted his focus is shifting toward consulting, with HL7 as the primary goal and NIH-side technology transfer as a source of tools he could bring to the community.

PR Review: Code Set Import Automation

PR under discussion: <https://github.com/openemr/openemr/pull/13706>

What It Does

- **Automates code set imports** — ICD-10, SNOMED, RxNorm and value sets — so a Docker container can be configured on initial setup without going through the user interface.
- **Origin:** Luis needed to bulk import dental codes in value set format for a prospective dental clinic deployment and found no easy path. Examining the existing DevOps import tool showed it was largely a wrapper around OpenEMR's internal import facilities, so he wrote something more direct.
- The same effort led him toward auto-generated FHIR resources for demo data.
- **CI checks were failing** and he asked for help interpreting them. One flagged issue was lack of code coverage.

The Guidance Given

- **Make it a Symfony console command rather than a standalone script.** That way it runs inside or outside Docker, is callable from anywhere, and sits alongside the existing commands — some for OpenEMR itself, some for release tooling.

Command examples referenced:

<https://github.com/openemr/openemr/tree/884d95f8c1daa606284d027ac1ac5daa0f3ace18/src/Common/Command>

- **Extract the logic into a testable service class,** with the CLI calling out to it. A recent refactor was offered as the pattern to follow — the command is a thin wrapper and the service holds the logic.

Reference implementation:

<https://github.com/openemr/openemr/blob/master/src/Console/Command/Codes/UpdateMappingsCommand.php>

and the service it calls: <https://github.com/openemr/openemr/blob/master/src/Services/CodeTypes/CodeTypeMappingUpdater.php>

- **Keep UI support as a separate PR,** since front-end changes need their own testing.
- **The architectural rationale stated:** OpenEMR is moving away from one-off flat-file scripts toward single entry points into a testable code base. Much existing logic is embedded in the interface itself; the pattern being applied is to extract data logic into a service, update the interface to use it, and let the command line use the same service — two paths into one piece of tested code.
- **Design philosophy exchange:** Luis noted his Unix-style habit of writing separate programs and going straight to a script. Response: a fair philosophy, but the project's architectural direction is consolidation into a class hierarchy that can be tested and automated. Accepted graciously on both sides.
- **On testing:** isolated unit tests preferred where possible — the suite runs roughly 4,000 units quickly via composer — but integration tests are welcome where that is the straightforward path, since database abstractions are not finished. Any coverage is a win.
- **Practical advice:** point AI at the code base to find where things belong; CLAUDE.md documents structure and testing conventions. Ask explicitly for coverage, because it needs reminding.
- **Noted benefit:** supporting both UI and CLI paths will likely require improving core code, so the contribution benefits the original code base as a side effect.

Code Set Licensing and Staging

- **Do not bundle the code databases themselves** — SNOMED in particular is very large and carries copyright restrictions. The established pattern is to provide a file and run an import. Confirmed by Luis: no code databases are included; the module works against the directories where files are expected to be dropped.
- **Use the staging directory.** It carries checks built up over years — whether the file set is new or old, whether all required files are present. ICD-10 changes file names and file counts most years, so entries are tracked in the database with hashes. Language variants exist, including a US ONC variant and a Spanish variant, and the staging mechanism tracks those too.
- **Consensus:** leverage the existing checks rather than duplicating them. Optionally accept a directory path argument, defaulting to staging.
- **Adjacent idea:** the demo farm could use these commands to auto-populate code sets. More broadly, moving operations onto command line tools streamlines a lot.

ICD-10 Mid-Year Updates

- **An ICD-10 mid-year issue was opened.** Opening the zip revealed an addenda file; there were no new ICD-10-CM codes this cycle, though PCS — the procedure code version, which may not be used in OpenEMR — did have codes.
- **Implication:** there now appears to be a six-month escape hatch for mid-year releases, so imports may not be a purely annual concern. Expect inconsistent file names and packaging, including double or triple-zipped archives.

Security Process and Workload

Discussed in general terms only; no specific vulnerabilities were described on the call. Details of the project's handling process are omitted from this published summary.

The Pressure

- **Report volume is rising sharply across the industry,** driven by AI-assisted analysis of open source code bases. This is not specific to OpenEMR, though projects handling patient data likely attract more attention than most.
- **Report quality is high.** The proportion of false positives is much lower than expected — automated tooling is finding genuine issues at a rate that would not have been practical for human researchers a few years ago.
- **Framed as a survivability question for open source projects generally:** whether a volunteer-and-vendor maintained project can keep pace with automated discovery.

Process Constraints

- **Embargoed work loses the project's tooling.** Ordinary pull requests receive continuous integration, automated review, and several iterations of correction before merge. Work carried out privately receives none of that, which makes it slower and more error-prone.
- **It does not scale at current volume.** The private path reliably introduces merge and testing problems that hold up releases, and nobody has hours available to untangle them. The goal is a release that runs from a single dispatch command.
- **Directional agreement was reached on how different classes of issue are handled,** with the most sensitive categories continuing to follow the private process and the remainder moving through the normal development workflow. Release cadence was deferred to the release call.
- **A concern raised and shared:** discomfort with any approach that publishes information without simultaneously giving operators a mechanism to update. This is a significant argument for faster and more predictable releases.

Structural Fixes Matter More Than Process

- **Estimate offered:** a large majority of reported issues would resolve as a byproduct of work already underway — improving the database connection layer and migrating ad hoc PHP templates to Twig, which handles output escaping properly.
- **Improving the upgrade path was argued to be the highest-leverage work available.** If upgrading were cheap and reliable, the entire disclosure question would matter far less, because operators could simply stay current.
- **The real upgrade obstacle is customized code, not the database.** Docker already auto-upgrades; what breaks is customization getting overwritten. A mechanism to preserve customizations and custom modules through an upgrade is what is actually needed — the same conclusion reached in earlier meetings.

Tooling and Distributing the Work

- **GitHub’s advisory API is severely limited.** Advisories can be read and edited, but comments cannot be read programmatically, items cannot be marked closed, and there is no way to query which have activity. Described as built and never finished — a major reason the volume is hard to manage.
- **Idea raised:** an external tool using the advisory feed as a backend, so work can be tracked and assigned out. Caveat immediately noted — such a tool would itself become a sensitive target and would need to be built carefully.
- **On bringing in help:** this work has so far been limited to admins and maintainers, because recognizing issues requires deep familiarity with the code base. Openness was expressed to distributing lower-severity items more widely as trust relationships develop, though current tooling does not support that split well.
- **Stated plainly as a valuable contribution opportunity:** building a tool to help manage and distribute the advisory workload. Nobody currently has time for it, and it would make a real difference.
- Luis Santos expressed interest in contributing to security work.

Release Status

- **Target: next week.** The last release was roughly two weeks prior. Position taken: cut what is ready and ship rather than waiting to bundle more.
- **A calendar bug must go out.** In 8.3, non-admin users doing facility management get a blank grey box where calendar events should be — a facility check against the event facility ID wipes the display. Admins do not see it. Client test systems panicked believing data had been erased; it had not. Fix is already in master.
- **Root cause:** something lost in the Twig migration. On inspection the problematic condition turned out to exist in the old code across the day, month, and week views — not an AI hallucination as first suspected, but a faithful port of a bad pattern.
- **On branch cut timing:** roughly three days before release. Long enough for acceptance testing to confirm packages, Docker images, and artifacts build and pass; short enough to avoid needing to backport much, while still leaving a window for late items. The previous cut ran a week only because of mechanism bugs.
- **Merge conflicts in this cycle** trace to work done on a branch built from an older code base — unavoidable regardless of target branch.
- **On 8.3 stability:** bug reports are coming in, but the majority of changes since release have been bug fixes rather than new features. Ireland completed full regression testing successfully. Considered acceptable — notably better than 8.0.0.
- **A beta release was raised** but faster regular releases were preferred instead.
- **More Twig migration is wanted** — it improves the application, though it will surface more issues of this kind. Sherwin has Twig conversion PRs outstanding, proposed for review on the next call.

Items Discussed and Proposed

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

Area	Discussed / Proposed
Meeting platform	Move the weekly call to the foundation's Google Workspace, so access to recordings and files is not dependent on one person. Requires updating the wiki page.
Recording notice	Zoom chat only shows to those present when posted, so a recording announcement has to be repeated for late joiners. A pinned or persistent notice would be better; the wiki call details page now carries it.
Luis Santos	Refactor PR 13706: extract the importer logic into a testable service class and expose it as a Symfony console command rather than a standalone script. UI support to be a separate PR.
Luis Santos	Consider listing MedicalMasses on the OpenEMR professional support vendor list and advertising target areas.
Security	Directional agreement reached on process handling. Release cadence to be settled at the release call.
Tooling	An external tool to manage the advisory workload and allow lower-severity items to be distributed was raised as a valuable contribution opportunity — nobody currently has time to build it.
Release	Target next week. Branch cut roughly three days before release to allow acceptance testing and a window for late items.
Bug	Calendar broken in 8.3 for non-admin users doing facility management — fix is in master.
Twig migration	Sherwin has Twig conversion PRs outstanding. Proposed as a topic for the next call.
ICD-10	Mid-year addenda now appear to exist; no new CM codes this cycle but PCS had some. Import tooling may need to handle a six-month update path.

Notes

- An aside on AI-assisted development ran through several topics: general agreement that it is an enhancement requiring a heavy hand rather than a replacement for system thinking, and an expectation that a wave of maintenance and security problems will follow from AI-generated code being deployed without that care.
- Recurring theme: nearly every thread came back to throughput. The security discussion, the release cadence, the PR guidance toward testable services, and the platform move all turned on removing bottlenecks — whether a single person holding an account, a process that breaks automation, or logic trapped in the interface where nothing can test it. Contributors willing to take on tooling or testing work would relieve real pressure.