OpenEMR Weekly Meeting Detailed Summary

October 27, 2026

Attendees: Stephen Waite, Stephen Nielson, Brady Miller, David Eschelbacher, Simon Quigley, Eric Stern (OCE), John Ketcham (DAW Systems), Luis Santos, Prabhu Pathak (LogiCoy), Ansuman Nayak (LogiCoy), Sherwin Gaddis. Joining later: Michael Smith (OCE), Farjad Zaghi, Pedro Camellon.

> *Italicized indented passages are background, clarification, and commentary added after the meeting, not statements made on the call. Items from the meeting chat are folded in where they belong. About five minutes of the meeting were not recorded because of a technical problem; the gap falls between the security discussion and the LogiCoy discussion.*

- **Shape of the meeting:** a full call with two e-prescribing vendors and two newcomers. OpenEMR 8.5.0 was released while the meeting was under way. The group looked back on the response to last week's PHP security emergency, gave DAW Systems and LogiCoy guidance on working with the project, and reviewed one PR in depth — which produced a concrete rule for code that touches the legacy parts of OpenEMR. Using AI to triage PRs was set as next week's topic.

OpenEMR 8.5.0 — Released During the Call

- **No blockers:** asked whether anything else needed to get in, Brady said the project should keep releasing as frequently as it reasonably can — "what gets in gets in." Some contributors had asked for their PRs to make 8.5; they will go into the next one. He was running the dry run at the start of the meeting.
- **It shipped before the hour was up.** Near the end of the call Brady announced that the release was out, with the Docker images still building and the announcements to follow. "All I did was run a dispatch script" — the automation, as one attendee put it, is the real hero.
- **What is still manual:** the AWS packages still go through Asher Densmore-Lynn, and a simple script is planned for those and for the Kubernetes pieces, though Brady thinks the AWS packages are on the way out. The wiki updates are manual too.
- **The real pain is the mailing list:** every announcement brings 30 or 40 removal requests, now on a list of about 22,000. The list lives in DynamoDB, which matches case exactly, so an address stored with capital letters is very hard to find and remove. Brady wondered whether to move to a paid email service that handles unsubscribes automatically.
- **Security advisories:** advisories for the security fixes in the release are published only after the release is out and the Docker images are built; a workflow then regenerates the changelog so they appear at the top.
- **PHP:** the Docker images now use the patched PHP, and Eric added a guard so that an image will fail rather than run on an older PHP.
- **Node is gone from the runtime.** There is no runtime Node in 8.5.0 — "it is out the door." With that blocker removed, the Docker stack can finally be simplified, and the single-binary build should no longer be held back; tests for it still need to be added.

> *The release record continues the roughly monthly rhythm: 8.4.0 on September 13, the 8.4.1 patch on September 20, and 8.5.0 on October 27. The Node milestone closes a thread that ran through these meetings. Schematron moved to PHP first; on 10/6 the CCDA service was close, and the CQM calculator was the last Node service. "Out the door" means all three are done, leaving Node only as a build-time tool for front-end packages. A separate Node.js runtime, with its own patch cycle, no longer ships with every OpenEMR installation.*
>
> *On the mailing list: at about 22,000 subscribers the list is large enough to fall under Gmail's bulk-sender rules, which apply at around 5,000 messages a day to Gmail addresses. Those rules require a one-click unsubscribe header (RFC 8058) and expect requests to be honored within about two days; Yahoo's rules are similar. CAN-SPAM's ten-business-day opt-out rule applies to commercial email, which a project newsletter may not strictly be, but the mailbox providers' rules apply regardless. Managed email services handle all of this automatically. The DynamoDB problem has a standard fix: store a lower-cased copy of each address as the lookup key, plus a one-time pass to normalize existing entries.*
>
> *The AWS packages — OpenEMR Cloud Express, Express Plus, and Standard — date from 2017, when one-click cloud images on AWS Marketplace were the easiest route to a hosted OpenEMR. Container-based deployment has since become the project's preferred shape, which is why maintaining separate AWS images looks less worthwhile.*

Lessons From the PHP2Shell Emergency

- **A recap?** Stephen Nielson asked whether anything about the response to last week's PHP emergency should change, or whether that was better handled offline.
- **The response was strong:** the view was that the project handled it as well as any organization could. It was mostly a matter of waiting for the upstream patch, and the past year's work on builds and automation put OpenEMR in a much better position to pull in the new PHP and get builds out quickly. Those involved did “heroic work” that day, putting in long hours.
- **Should the project warn people sooner?** Stephen pointed out that it took 12 to 13 hours before there was anything to ship. A vendor he spoke with that afternoon had not even known what was going on, and had to start fixing its own clients in a hurry. Does the project have a responsibility to tell people it is aware of an issue and working on it?
- **Views offered:** this one was international news, and anyone running servers needed to know regardless. For an urgent OpenEMR vulnerability with a patch, the project should use its email list. For a platform issue, the deployer is ultimately responsible for knowing what software they run. The list is available whenever someone drafts the message — and there will be more of these.
- **Context discussed:** the bug had existed for years. OpenAI and the PHP developers apparently knew about it for around two months before it was accidentally disclosed, and honeypots were seeing exploitation attempts days before the announcement — whether from the leak or from someone's AI finding it independently. “Nothing's safe at this point”; the best response is to fix quickly and move on.
- **The cost:** an attendee reflected on the enormous worldwide cost of one bug — every site spending staff hours at once, which is why so much was crashing — set against a PHP Foundation with very few paid staff. Stephen Nielson noted that it made him think about fundraising, but that is a board discussion.

> *PHP2Shell is two memory-safety flaws in PHP itself, CVE-2026-109618 and CVE-2026-109619, which allow remote code execution on web hosts running PHP 5.4 through 8.5. Ryan Austin of Calif found them in August 2026 while auditing PHP's source with OpenAI's GPT-5.6 Sol, as part of OpenAI's “Patch the Planet” project; Wiz found them independently with its own AI-assisted tooling. Details leaked on October 20 through a Hacker News post, a watchTowr post, and a proof-of-concept video released by mistake. The PHP team moved the release forward a day, publishing PHP 8.2.35, 8.3.36, 8.4.27, and 8.5.12 on October 21. Amazon's threat-intelligence honeypots recorded an exploitation attempt on October 17, four days before disclosure. Government portals across Europe went offline as a precaution, and Wiz compared the event to Heartbleed.*
>
> *Two follow-ups are worth knowing. On October 23 the PHP team disclosed a third, narrower issue, CVE-2026-115560. It affects only applications that switch session serialization handlers at runtime with upload progress enabled, is mitigated by setting session.upload_progress.enabled to 0, and will be fixed in the next routine release. And on staffing: the PHP Foundation's executive director told Reuters it has one full-time staff member and 14 mostly part-time contractors, close to the call's “one dedicated engineer with 20 part-timers.” She also said PHP now receives nearly twice as many proposed code changes as a year ago — the same AI-driven surge OpenEMR is facing.*
>
> *On early warning: in coordinated disclosure, advance patches typically go to Linux distributions and long-term-support vendors, as they did here, while application projects learn at publication. A short “we are aware and working on it” notice, followed by a release announcement, is common practice for projects with large user bases, and the email list exists for exactly that. Pointing users to an authoritative alert — CISA issued one on October 21 — costs little and helps operators who are not following security news.*

LogiCoy — Publishing the Module on Packagist

Links: Packagist · OpenEMR modules on Packagist · logicoy/oe-module-logicoy-erx

The recording resumes partway through this discussion.

- **Stephen Nielson's advice:** until a fully bundled, certified product is in place, publish the module on Packagist. Then anyone can install it with Composer and start LogiCoy's sign-up process without having to talk to LogiCoy first.
- **LogiCoy's status:** the module exists and is installed with a script; it just needed packaging. Prabhu summarized the suggestion for Ansuman, and Stephen posted links in the chat — including a search showing how packages tagged as OpenEMR modules appear together — and pointed to the project's skeleton module as a model.
- **Done during the meeting:** by the end of the call Ansuman had uploaded the module to Packagist as logicoy/oe-module-logicoy-erx.
- **Next:** announce it in LogiCoy's post in the forum's third-party solutions and add-ons area. A request on etiquette: vendors should update their own posts and not advertise their products in other vendors' threads.
- **LogiCoy's question:** having read last week's summary, they asked about the idea of a FHIR-based layer that would let many e-prescribing providers connect in a uniform way. Stephen Nielson: it was an exploratory conversation only. There is no design and no development work, and current integrations work through the database as the existing ones do. The project would not pull the rug out from under anyone, and would consult all the e-prescribing vendors before setting any such direction.

> *The package as published: type openemr-module, licensed GPL-3.0-or-later, version v1.0.9, published on Packagist on October 27 at 18:35 UTC — during the meeting — with its source at github.com/Logicoy/oe-module-logicoy-erx. The openemr-module package type is what the OpenEMR module installer plugin looks for: Composer places the module where OpenEMR's Module Manager can find it, ready to register, install, and enable. With the source public under the GPL, anyone can now review the code.*
>
> *LogiCoy's question shows that vendors read these summaries closely. Stephen's answer matches what was said on 10/20: a SMART on FHIR approach is the long-term preference, but it is an idea, not a plan, and any change would be made with vendors rather than imposed on them.*

DAW Systems — How to Work With the Foundation

- **Introductions:** John Ketcham, chief revenue officer of DAW Systems, an e-prescribing vendor, has been talking with Sherwin about integrating DAW's product into OpenEMR. New to the group, he apologized for not requesting agenda time formally.
- **His questions:** how DAW would work with the Foundation, what contracting and commercial terms look like — he assumed some form of contribution to the project — and whether there are BAAs, vetting, or other steps needed before DAW allocates resources to a build.
- **How the project is organized, as Stephen Nielson explained it:** the OpenEMR Foundation governs compliance, certification, and legal attestation for the project. Its board meets monthly, with Stephen Waite as president and Brady Miller as executive director; Stephen Nielson is director of engineering. The technical project is run by about five volunteer administrators. A smaller governance group meets weekly on contracts, negotiations, and governance, and a tentative contract ultimately goes to a board vote.
- **Two paths:** a vendor does not need the board to offer a solution to the community — it can build its own module, or contract someone in the community to build one, and offer it to users who want it. Bundling with OpenEMR's certified product is different: the project is not interested in bundling anything that is not ONC certified, and that path runs through the certification process. Commercial terms are for the board, not the weekly technical call.
- **John's summary:** “almost like fan fiction” — anyone can build on the stack and put it out there, but the certified version takes a formal process, and that is what DAW is interested in.
- **Next steps:** DAW confirmed it is ONC certified and listed on CHPL. Stephen asked DAW to contact Stephen Waite, who posted his email in the chat; send the CHPL links so the project can verify the certification; and plan a product demo, deciding whether DAW would build the integration or hire it out. John said the product team would decide, and DAW would probably hire it out. Sherwin, who is on the board, has already been engaging with DAW, but the process should now become more formal, including Brady as executive director and the governance group.
- **NDA first:** John said DAW would want an NDA in place before a demo and deep dive into the APIs. Any NDA would run through the board, so the conversation starts with Stephen Waite and Brady.

> *The OpenEMR Foundation is a 501(c)(3) nonprofit that manages the project's infrastructure, accepts tax-deductible donations, and holds its legal assets and certifications, including the ONC certification that a bundled e-prescribing partner would join. Its board's monthly meeting agendas are posted on the community forum. The two-path answer matches Sherwin's advice to DAW reported on 10/20, and the ClaimRev model of a vendor-maintained module shipped with the distribution.*
>
> *On BAAs: an e-prescribing service handles protected health information on behalf of the practices that use it, so under HIPAA it signs business associate agreements with those practices — the covered entities — not with the open source project. The Foundation's role is certification and governance, which is why Stephen steered the BAA question toward the certification conversation rather than treating it as a prerequisite for building a module.*

PR 14053 — Printable Active Medication List

Pull request: https://github.com/openemr/openemr/pull/14053

Where Simon's PRs Stand

- **Three ready:** Simon has 15 PRs open, and three are ready for a solid review — 14053, 14254, and 14393. The other 12 could use a first pass by human eyes but are still failing a check or need another iteration. He thanked Stephen Waite for recent review comments, which have helped.
- **LBF statements:** the LBF statements PR, decided on 10/13, now has a demo linked from the pull request.

What the PR Does

- **Purpose:** medication reconciliation. Simon's client wants a single report showing active medications.
- **In the interface:** the medication card on the patient dashboard now shows how many medications are inactive, with a print button beside it that opens the new report; the report is also available in the custom patient report.
- **From the PR description:** the printable list separates active medications from inactive or historical ones, showing name, dose, start and end dates, and comments. The medication card and patient report now draw on the same data; duplicates across sources are removed; and tests cover duplicates, missing or invalid dates, and the active-versus-historical distinction.

> *Medication reconciliation — comparing what a patient is actually taking with what the record says, at every transition of care — is a Joint Commission National Patient Safety Goal, and incorporating reconciled medication lists is part of ONC's 170.315(b)(2) certification criterion. A clean, printable list of active medications is the kind of artifact a clinician hands a patient to confirm at a visit. That is why separating active from historical entries, and removing duplicates, matters clinically as well as visually.*

Legacy Code and Twig — The Debate

- **The litmus test:** Michael Smith raised last week's discussion about protecting the legacy parts of the code such as interface/. Without being excessively strict, could the report and summary code in this PR move to a more MVC part of the codebase? Stephen Nielson agreed that is the preference. Michael's PHPStan rule, which will enforce it, has not yet been merged.
- **This PR's legacy change is tiny:** the change to patient_report.php is a one-line bug fix for an untranslated string. So the question is whether touching a legacy file should mean converting the whole script to Twig, or only part of it.
- **Stephen Nielson:** AI is good at this kind of conversion, so if someone touches a file it should move, even though that slows PRs. There needs to be a cutoff date for fairness: older PRs should not be held to rules that did not exist when they were written, and they can be merged with an override unless the change is simple.
- **Michael's alternative framing:** given how hard it is to review the bulk of agentic contributions with human eyes, the project simply should not automatically accept or approve changes to legacy or deprecated parts of the code. A human can still approve them; they just won't get the priority a machine can give.
- **Eric Stern:** fixing a bug in an existing legacy template is not a big deal. A substantial addition — one or two files here were essentially new templates — should be done in Twig.
- **How the PHPStan rule will work in practice:** PRs that follow unwanted patterns will fail the check with a red X. A human reviewer can override it — for a bug fix, for example. Stephen said that, given the volume, he only reviews PRs with green check marks, apart from ones where the only failure is Codecov. An attendee added: “I've never liked echoing HTML.”
- **Eric on how to convert:** Twig is extensible and compartmentalized, so even a block within a page can be its own template. Wholesale conversion of an interface file tends to duplicate structure across templates; better to extract shared page structure so it is reused. He did something similar months ago, pulling a section of an interface file out into Twig — legacy files can render Twig themselves.
- **What is wrong with the legacy pattern, in Stephen's words:** echoing HTML with inline access checks, and conditionals on variables with no clear scope. The MVC structure OpenEMR already supports puts access control in the controller and leaves rendering to the Twig template, rather than mixing very different concerns.
- **Whole file or pieces?** One view preferred converting a whole file or nothing, rather than many small slices, while acknowledging that whole-file conversion carries more risk of regressions. Eric did not feel strongly. Because OpenEMR's interface is built from iframes, pages share little, so there is limited reuse to gain. Twig base templates exist — for error pages and the overall header and layout — but are rarely used, and should be used consistently. He would rather see steady movement toward Twig than block progress by optimizing the template structure.

The Decision — A Cutoff for Legacy Changes

- **Agreed:** once Michael's PHPStan rule is merged, new PRs must comply — Twig templates and MVC structure, rather than new code in legacy files. A serious bug fix can still be approved with a human override. The cutoff will be set at today's or the next admin meeting, given a week to bake, and adjusted if it reveals a real gap in the workflow.

> *The legacy areas were defined in the chat. Eric: broadly, interface/ and library/ are legacy. Michael: at a lower level, a really long, mostly procedural file with HTML and PHP mixed together is deprecated. Pairing a written rule with an automated check is the approach recommended on 10/20 under "fitness functions": a PHPStan rule turns an architecture document into something that holds across hundreds of PRs without a human remembering it each time.*
>
> *Michael's framing — not banning legacy changes, but declining to auto-approve them — matters for fairness to newcomers. A first-time contributor fixing a one-line bug should not have to rewrite a 2,000-line legacy page, and the human override keeps that possible. Eric's point maps onto Twig's own tools: template inheritance ({% extends %} with blocks) shares page structure, and {% include %} or {% embed %} pulls in reusable fragments. That is how a single section of a legacy page can move to Twig without converting the rest — the strangler-fig approach in miniature.*

Review Notes on the PR

- **Too much in one class:** Stephen Nielson's main concern was the new active medication list service. A service class is generally for business logic or data access, but this one also produces HTML. He suggested a separate controller — perhaps a card controller for the medication card — with the HTML moved into a Twig template, a Twig extension, or a render helper. He has seen several PRs turn their service class into a "god class" that does everything, which makes code hard to test.
- **A stray change:** an unrelated comment change in a utility file looked like an artifact of a rebase.
- **Screenshots, please:** for PRs that change the interface, screenshots help reviewers enormously — AI-written PRs often leave them out. Simon will add screenshots and split up the class.

> *A "god class" is the textbook name for a class that knows and does too much. The usual remedy follows the separation Stephen described: services hold business rules and data access, controllers handle requests and permissions, and templates render. Each piece can then be unit-tested on its own — a service can be tested without a browser, and a template without a database. A pull-request template that prompts for screenshots whenever the UI changes is a cheap way to make that request automatic.*

A Demo-Site Detour

- **Demo credentials changed:** while showing the PR on the public demo, Stephen found that a visitor had changed the admin password, on the alternate demos too. It is a recurring nuisance.
- **Past and possible fixes:** Brady once ran a script on the demo farm that reset the passwords every five or ten seconds, after a period when someone was changing them within seconds, but it stopped working for some of the demo instances. Another suggestion was a fail2ban-style rule that watches the logs and blocks a repeat offender for 24 hours. Anyone willing to write the rule would be welcome.

> *fail2ban is a long-standing open-source tool that watches log files for patterns — repeated failed logins, for example — and temporarily blocks the offending IP address with a firewall rule. Using it against a web application means writing a filter for the application's own logs. For a public demo, where credentials are deliberately published, an automatic reset on a short timer is often simpler and more robust than trying to tell playful visitors from malicious ones.*

Guidance for AI Agents — AGENTS.md, Skills, and Architecture Docs

- **In the chat, Pedro Camellon asked** whether the legacy rules would be noted in the agent instructions. Michael: yes, along with a stricter CI check. Pedro observed that the repository has CLAUDE.md and AGENTS.md files but no skills folder, and suggested breaking the guidance into multiple skill files — how to write tests, how to build a template — that agents load on demand. He also asked how often agents ignore the rules in AGENTS.md; Michael could not say.
- **Stephen Nielson's architecture guidance:** he has posted proposed architecture guidance for AI agents, drafted with AI, in the admins' chat, and asked the admins to review it. It incorporates what is already in AGENTS.md and adds architectural direction not covered there. It is meant as a set of living documents, updated as decisions are made, and broken up so that an AI reads only the context it needs — which makes it a little more awkward for human reviewers. (“I'll have my AI look at it.”)
- **Why the reasoning matters:** it was noted that keeping the context behind a decision with the decision makes a big difference. When a human overrides a rule because it makes sense in a particular case, that context is what justifies the judgment call.
- **Describe how things are, not only how they should be:** Stephen's example: while drafting with Claude, the AI stated a textbook templating rule that the project has deliberately chosen not to follow, at least for now — guidance that would have misled every agent reading it. Codifying how things actually work versus how the project wants them to work is an important distinction.

> *Agent Skills are the format Pedro was describing. Anthropic introduced them in 2025 and published them as an open standard on December 18, 2025 (agentskills.io). Microsoft, OpenAI, GitHub, Cursor, Atlassian, and Figma were among the early adopters. A skill is a folder with a SKILL.md file of instructions plus any scripts or examples it needs. Agents load it only when a task calls for it — what Anthropic calls progressive disclosure — so detailed guidance does not crowd the context on unrelated work. The natural division is AGENTS.md for short, always-on rules and skills for task-specific procedures, which is close to the “broken up” structure Stephen describes.*
>
> *On Pedro's question about how often agents ignore instructions: there is no published rate for any particular project, but adherence is known to fall as instruction files grow longer or contradict themselves. That is the argument for both of the measures Michael mentioned, compact guidance and a CI check. Eric's point about recording the reasoning has an established form too: architecture decision records, short documents that capture a decision, its context, and its consequences, so later readers — human or AI — know when an exception is justified.*

New Faces and the Review Queue

Introductions

- **Pedro Camellon** is an AI engineer and full-stack developer specializing in AI, machine learning, and cloud architecture, employed by Mitek. He is new to the codebase and has been reviewing PRs.
- **Farjad Zaghi, DO,** is a cardiologist practicing in New York (Nefesh Medical PLLC). He recently started using OpenEMR and customizes it for his practice with Claude. When a change has to touch core, he submits it as a PR. He is not a professional coder but tries to help where he can, and called the meeting "fantastic."

The Backlog

- **Michael Smith:** outside contributions used to run at maybe 10 to 15 PRs a month; the project now gets about 100. The answer is to automate as much as possible and look for good candidates to become additional maintainers, while making sure the codebase does not turn into a big ball of spaghetti.
- **Stephen Nielson:** as a healthcare application, OpenEMR cannot simply accept everything; it has to be prudent, using AI to automate as much as it can. Farjad expects it to get worse as more people use AI, and suggested AI on the project's side to filter.
- **Getting a PR reviewed on the call:** asked how people can queue up, the suggestion was to post in the weekly meeting thread on the community forum. With only one or two PRs reviewed per call, there was talk of a lottery — and, jokingly, of paying for reviews as a fundraising solution. Michael, in the chat: "except AI is pay-to-play."
- **Forum follow-up:** Stephen Waite has replied on the forum to a newer contributor, Anun, who could not attend and posted questions after last week's summary.

From the Chat

- **PR 14007 (Farjad):** a one-line fix in Configure Orders and Results. When OpenEMR creates a new procedure type, the SQL placeholders and bind values are out of order, so every field can silently shift into the wrong column. Stephen Nielson has approved it, and Anun independently reproduced the bug and verified the fix on current master. It still merges cleanly. Farjad asked whether it could be merged, and offered to open a backport to the 8.5 release branch.
- **Priorities:** Farjad would love a way to speed up reviews, and Pedro asked how contributors can tell what the project's current priorities are.
- **HTTP/3:** Simon switched a client deployment to HTTP/3 with nginx late last week. It took a few tweaks and bug fixes, but he recommends it.

> *The bug in PR 14007 is a classic hazard of parameterized SQL. Placeholders are filled strictly in order from the bind array, so if the array lists values in a different order from the columns, each value lands in the wrong column — and when the column types are compatible, nothing errors. In a lab-orders configuration table that means silently corrupted order definitions, which is why a one-line fix with an independent reproduction is the ideal small PR. Backporting it to the release branch would put it in the next 8.5 patch rather than waiting for 8.6.*
>
> *nginx added HTTP/3, which runs over QUIC on UDP, in version 1.25.0. A typical setup keeps the existing listener for HTTP/1.1 and HTTP/2 over TCP, adds a "listen 443 quic reuseport" listener, and advertises HTTP/3 with an Alt-Svc response header so browsers know to try it. The common pitfalls are a firewall that blocks UDP port 443 — browsers then silently stay on HTTP/2 — and replay risk if 0-RTT early data is enabled for requests that change data. Applications that make many small requests over imperfect networks tend to benefit most.*
>
> *Pedro's question about priorities has no single answer page today. The closest things are these weekly summaries, the forum meeting thread, and labels on GitHub. A short, pinned list of current priorities — the legacy cutoff, FHIR testing, security fixes — would help newcomers aim their effort, and would give an AI triage tool something to rank against.*

AI for Triage — Next Week's Topic

- **Michael's proposal:** if it can be done economically, use AI for more than reviewing PRs and writing code — use it to triage and prioritize pull requests, and for other higher-level work that may make more difference than reading the code itself.
- **Stephen Nielson's example:** he recently spent five or ten minutes working out that a PR should never have been accepted; AI could have flagged and closed it automatically. Triage could also surface small bug fixes quickly and route critical or billing-related PRs to the right people.
- **Agreed** as a discussion topic for next week.
- **Old PRs versus new guidance:** asked how hard it will be to handle older PRs that stray from the new architecture guidance, the answer was to "rip off the Band-Aid," as with PHPStan. Old PRs touching legacy code can still come in, while new ones will be steered by AI that has the rules in place. With good AI context, a contributor can simply be asked to "follow OpenEMR's best practices and clean up this PR," and the project will need instructions to help people do that. GitHub Copilot could in principle rebuild a PR to match, but it has become expensive, and the free tier's limits run out quickly.
- **Brady's view:** CodeRabbit is doing a good job. Many queued PRs are bug fixes that pass CodeRabbit and need only one human to confirm they are reasonable; the project just has to keep chipping away at them.
- **Closing:** Stephen Nielson told the newcomers that an AI summary of the meeting will be posted on the community forum, and encouraged questions and review requests there.

> *AI triage is a different job from AI review. Review asks whether a change is correct; triage asks what kind of change it is, how risky, which area it touches, and who should look first. That can be done cheaply from the PR metadata and diff, without deep analysis. Typical outputs are labels (bug fix, feature, billing, security-sensitive, legacy area), a size and risk tier, and a suggested reviewer, using routing rules much like a CODEOWNERS file. Because a triage agent reads text written by contributors, it inherits the prompt-injection caution discussed on 10/20: it should label and route, not merge or close without a human.*

Items Discussed and Proposed

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

Area	Discussed / Proposed
Release 8.5.0	Released during the call; Docker images, advisories, and announcements to follow. Runtime Node is gone.
Release chores	AWS and Kubernetes steps to be scripted. Mailing-list unsubscribes are the painful manual step; a managed email service was suggested.
PHP2Shell response	Handled well. For urgent OpenEMR vulnerabilities, use the email list; platform issues are ultimately the deployer's responsibility, though a heads-up email can go out if someone drafts it.
Docker	With Node gone, revisit simplifying the Docker stack; add tests for the single-binary build.
LogiCoy	Module published on Packagist; announce in the forum add-ons area. No FHIR-based e-prescribing standard is in development.
DAW Systems	Two paths explained: community module, or certified bundling through the Foundation. Next: contact Stephen Waite and Brady; DAW wants an NDA before a demo.
PR 14053	Reviewed. Split the service class (controller plus Twig); add screenshots.
Legacy-code cutoff	Once the PHPStan rule is merged, new PRs must use Twig and MVC rather than legacy files; human override for bug fixes; one-week bake.
AI guidance	Stephen Nielson's architecture documents for admin review; skills-based structure suggested.
Demo site	Restore automatic password resets or add a fail2ban-style rule.
PR review requests	Post in the weekly meeting thread on the forum.
PR 14007	Merge and a backport to the 8.5 branch requested in the chat.

Future Discussion and Next Meeting Agenda

Proposed for upcoming meetings, drawn from this week's discussion and chat; not a fixed agenda.

Item	Details
AI triage of PRs	Agreed as next week's discussion topic: classifying, prioritizing, and routing PRs with AI.
PRs queued for review	Simon's 14254 and 14393, plus 14053 after changes; Farjad's 14007, with a possible 8.5 backport; the LBF statements demo.
Legacy-code cutoff	Merge date for the PHPStan rule, the announcement, and a review after the first week.
Architecture guidance	Admin review of Stephen Nielson's documents; whether to adopt skills alongside AGENTS.md.
DAW	Progress through Stephen Waite and Brady; a possible demo under NDA.
LogiCoy	Forum announcement of the Packagist module; partnership through the governance process.
Mailing list	Evaluate a managed email service with automatic unsubscribe handling.
Docker stack	Simplification now that runtime Node is gone; tests for the single-binary build.
Security notices	Whether to adopt a standing practice for heads-up emails during platform emergencies.
Fundraising	Raised by the PHP emergency; a board topic.
Carried over	Multisite, subdirectory, and non-Docker deprecation with the migration guide, and procedure writes over FHIR (pending since 10/6); logging engine #2119; Ensora and MI-Squared outreach; contributor limits.

Notes

- **The connecting thread:** PHP and OpenEMR are living the same AI-era story from opposite ends. AI audits found a decade-old bug in PHP that took down sites worldwide, and AI-assisted contributors have roughly doubled PHP's incoming changes while increasing OpenEMR's outside PRs nearly tenfold. Both projects are answering the same way: automate, triage, enforce standards in tooling, and look for sustainable funding.
- **From debate to rule:** the 10/20 discussion about architecture and AI-generated PRs became a concrete policy in a single review. A legacy-code cutoff is tied to an automated check, with a human override, and PR 14053 was its first test case.
- **A clear vendor path:** the two routes for e-prescribing vendors — a community module anyone can install, or certified bundling through the Foundation — were laid out publicly and apply equally to LogiCoy and DAW. LogiCoy's module is now publicly available under the GPL.
- **Gap in the record:** about five minutes of the call were not recorded, between the security discussion and the LogiCoy discussion. Anyone who recalls decisions from that stretch is welcome to add them.
- Ways to help: vendors and integrators can publish OpenEMR modules on Packagist with the openemr-module type; contributors can follow the Twig and MVC direction for new code, add screenshots to UI changes, and request reviews in the forum's weekly meeting thread. Anyone experienced with list management or HTTP/3 deployments is welcome to share what has worked. Comments on the forum or at the weekly call are welcome.