OpenEMR Weekly Meeting Detailed Summary

October 20, 2026

Attendees: Stephen Waite, Stephen Nielson, Brady Miller (by phone), David Eschelbacher, Simon Quigley, Sherwin Gaddis, Eric Stern (OCE), Michael Smith (OCE).

> *Italicized indented passages are background, clarification, and commentary added after the meeting, not statements made on the call. Several items from the meeting chat, added by David Eschelbacher, are folded in where they belong.*

- **Shape of the meeting:** "the core, everyone's here today," with Michael Smith and Eric Stern both attending. Most of the time went to e-prescribing strategy and then to a wide-ranging discussion of how to keep the codebase's architecture coherent while AI-generated PRs pour in. The call closed on the logging engine and ran over, so Simon's PRs were not reached.

Release 8.5.0 — Branching This Weekend

- **Plan:** Brady expects to fork the release branch this weekend and release next Tuesday or Wednesday. It would have happened this week, but his schedule did not allow it.
- **It will be 8.5.0, not 8.4.2.** Minor releases are the norm. The 8.4.1 patch release was needed to fix bugs, and Brady also used it to try the patch mechanism for the first time; working out its bugs means a quick patch path now exists when one is needed.
- **Automation:** the last release went smoothly enough that it could have shipped the next day. This time Brady changed the release tooling — acceptance tests now exercise every package as it is built — and changes like that sometimes break during a real release, so a few glitches are possible.
- **Checking with Jerry:** Brady will let Jerry know the branch is coming. FHIR writes are in, gated behind a prominent warning when an administrator enables a client with write access, and labelled experimental; they are limited to user scope, not patient context. Jerry is working on the audit engine and related task and questionnaire work, with help from another contributor whose PRs Brady has been bringing in.
- **No rush:** with releases now roughly monthly, nothing needs to be squeezed in. It is not like the yearly releases of the past.

> *The release record bears this out: 8.0.0 on February 13, 2026, then 8.2.0 on July 8, 8.3.0 on August 18, 8.4.0 on September 13, and the 8.4.1 patch a week later on September 20. Since the summer the project has run what is often called a release train — a release leaves on schedule with whatever is ready, so a missed feature waits a few weeks rather than a year. The trade-off falls on operators, who face more frequent upgrades; that is part of why the release automation and upgrade testing Brady describes matter.*
>
> *Labelling FHIR writes "experimental" reconciles the positions of the last two meetings. On 10/6 Eric Stern suggested treating the write gate like a feature flag for an area still in development, and on 10/13 Jerry replaced the planned off-by-default global with a warning when write-capable clients are enabled. An experimental label plus the warning gives administrators both signals.*

Inferno Is Running in CI

- **Stephen Waite got Inferno running** in the project's automated testing. He closed an older PR of Brady's on the same task; Brady was glad, having started it — "the first step" — and run out of bandwidth for the other ten.
- **How it came together:** Stephen Waite pinged Stephen Nielson privately, put Claude to work on the Inferno files, bumped them to point at a newer commit, and filled in a couple of missing pieces. Stephen Nielson had an older copy from last December that had grown to 2.5 GB, probably because it bundled the code sets.
- **The limitation:** Inferno only runs when code reaches master, so a PR claiming that something in the FHIR API is broken cannot be checked against Inferno before it is merged. Brady's example: a PR saying the "missing" search modifier does not work — if that were true, he would expect Inferno to fail.

> *Inferno is ONC's open-source FHIR testing framework, and its (g)(10) Standardized API Test Kit is an approved test method for the 170.315(g)(10) certification criterion. Running it in CI turns certification behavior into a regression test: if a change breaks something certification depends on, the project finds out on the next merge rather than at the next certification cycle. A natural next step, if run time allows, would be running a subset on PRs that touch FHIR code, which would close the gap Brady describes.*
>
> *The FHIR ":missing" search modifier finds resources where an element is absent (true) or present (false) — for example, Observations with no value. It is a small feature, but a good illustration of why automated conformance testing matters: an AI-generated PR can assert a bug confidently, and only a test can settle whether it is real.*

API Test Coverage — Portal, Standard API, Then FHIR

- **Volume:** about 300 open PRs, with "every time you turn around, another hundred."
- **Brady is treating the FHIR PRs as part of his security work.** FHIR endpoints for certified deployments are public by regulation, so their authentication and authorization need to be solid. He has used AI-assisted audits to plan full test coverage, including negative testing — confirming that requests which should be refused are refused — especially around OAuth, so that later changes cannot quietly introduce regressions.
- **Progress:** coverage added to the older patient portal API, the standard API nearly done, and FHIR next. Seven or eight open PRs relate to FHIR testing with fixes; he expects to fold them into the audit and bring them in together.
- **Tests find bugs:** adding them has turned up small issues. One example: OpenEMR's FHIR metadata advertises sorting and page limits, but those are implemented on only some endpoints, and the metadata is supposed to describe exactly where they are supported. A contributor's PR also found a number of bugs in standard API filters — less-used features that people tend to work around.
- **SMART launch tests in CI:** Brady brought in tests that run both the EHR launch and the standalone launch on every CI run and then use the API — exercising the fundamentals of OAuth and SMART, with negative tests added.
- **Relationship to Inferno:** Stephen Nielson noted the project has relied on Inferno for much of this validation and might not always be able to, if MITRE's budget were cut. Brady said his tests deliberately do not duplicate Inferno: Inferno covers a limited, mostly happy-path subset and is not designed to catch security regressions. Stephen added that Inferno does check some security aspects, such as confidential clients, but not exhaustively.
- **Why it matters now:** with FHIR writes in, API use is likely to grow.

> *Negative tests are the half of testing that features usually skip. A positive test confirms that an authorized request works; a negative test confirms that an unauthenticated request, an expired token, a token for the wrong patient, or a scope that does not cover the operation is refused. Security regressions almost always show up as a missing refusal, which is why negative tests are the ones that protect an API over time.*
>
> *A FHIR server's CapabilityStatement — the document returned from the /metadata endpoint — declares which resources, interactions, and search parameters it supports. Clients read it to decide what to send. If it claims support for _sort or _count where they are not implemented, a server may simply ignore the parameter, and a client receives unsorted or unpaged results without any error. Accurate capability statements are part of (g)(10) conformance, and they matter more as third-party apps start writing data.*
>
> *Stephen's MITRE remark reflects a real dependency. Much federal health IT test tooling is built under federal contracts, and in April 2025 MITRE — a major federally funded R&D contractor — announced 442 layoffs after a series of contract cancellations. Inferno's code is open source, so it would remain usable either way, but a project-owned test suite of the kind Brady is building is insurance against changes in federal tooling.*

E-Prescribing — Goals, Certification, and Partners

Links: CHPL search · Ensora eRx · Weno · LogiCoy · DAW Systems

The Options on the Table

Background compiled after the meeting from vendor websites and earlier summaries.

Option	Status and background
Ensora eRx (formerly NewCrop)	Long-standing integration in OpenEMR core, provided through authorized OpenEMR vendors such as MI-Squared. Ensora's product offers EPCS, PDMP access, benefit checks, and ePA over Surescripts. The OpenEMR integration is not currently part of a certified configuration.
Weno Exchange	Bundled module. Low cost; uses Weno's own pharmacy network, with coverage gaps reported in some areas (9/22).
LogiCoy eRx	Module in development by LogiCoy, which presented on 10/13. ONC-certified standalone product (CHPL listing 10944) with a direct Surescripts connection.
DAW Systems (ScriptSure)	Proposing; pre-development. An Albany, New York e-prescribing company whose ScriptSure platform claims about 30,000 clinical users and 190 or more integrations, offered stand-alone or embedded in other platforms. Wants to be part of the OpenEMR distribution.
DoseSpot and MDToolbox	Not bundled. Sherwin Gaddis has built integrations with both for clients who needed certified e-prescribing.

DAW Systems

- **Raised from David's chat list:** any more feedback on direction with LogiCoy, and what DAW is. Stephen Nielson did not know DAW; Sherwin did.
- **Sherwin's account:** DAW has been asking Brady for a meeting, and Brady's email queue has been too full to respond — Brady apologized. Sherwin has guided them to build their own module, be its maintainers, and publish it on Packagist so anyone can install it. DAW's focus, though, is being part of the distribution rather than “an afterthought” that people have to find.
- **What DAW wants:** to replace Ensora. Sherwin explained that the direction is the reverse — move Ensora itself into a module and make every option a module that users choose — and that a vendor module can still ship with the distribution, the way the ClaimRev module does, while the vendor maintains the code.
- **History:** DAW attended one or two meetings in the past and then dropped out.
- **Next step:** Sherwin will invite DAW to a Tuesday meeting.
- **A concern raised:** with vendor pitches becoming more common, the call could be inundated every week; a forum-based process might help.

> *DAW Systems makes ScriptSure Cloud ERX. Its 2022 press materials describe Surescripts network connectivity, First Databank clinical data, EPCS accreditation, EHNAC privacy and security accreditation, and a Surescripts White Coat Award for accuracy; its site cites about 30,000 clinical users and 88,000 pharmacy connections. Medplum, an open-source healthcare platform, integrates ScriptSure in two ways — a hosted, authenticated iframe or a custom workflow built on FHIR operations — which suggests the integration patterns DAW could offer OpenEMR.*
>
> *The ClaimRev module is the model Sherwin pointed to: it appears in OpenEMR's Module Manager under Unregistered as the ClaimRev Clearinghouse Connector, ready to install, but it requires a ClaimRev account and is maintained by the vendor. Packagist is the public registry for PHP Composer packages, which is how a vendor-maintained module can be installed — or pulled into the distribution with a single line in composer.json — without its code living in the OpenEMR repository.*

The Goal — a Certified (b)(3) Option

- **Asked what the goal is** — simply as many options as possible? — Stephen Nielson answered: a certified (b)(3) companion product for OpenEMR. Multiple options are welcome, and could even include more than one certified option, but he wants a certified e-prescribing option. LogiCoy is certified.

- **Could a practice just add it on CHPL?** Technically, yes — a provider can combine products. But Stephen would rather it be part of OpenEMR's certification, as EMR Direct is. Otherwise anyone checking will see that OpenEMR is not certified for e-prescribing, and will have to work out on their own how to get certified e-prescribing.
- **How the project got here:** Ensora decided not to pursue certification for the OpenEMR integration. MI-Squared obtained quotes, and the work for Ken to update and certify the integration would have been substantial. Ensora's product was certified, but the interface OpenEMR used was not.
- **Cost:** adding (b)(3) to OpenEMR's certification carries an additional certification fee, plus any development work needed.
- **Why a companion product helps:** the fee still applies, but the process is far simpler. With EMR Direct, the certifier watched a demonstration — click through the interface and show the required steps — and nothing more was needed.
- **Agreement:** asked whether everyone was on board with packaging a (b)(3) option, the response was yes.

> *CHPL is ONC's Certified Health IT Product List — "chapel" on the call. Its CMS EHR Certification ID tool lets a participant select one or more certified products and generates an ID once the selection together meets 100% of the Base EHR Definition; ONC's quick reference on it was updated September 1, 2026. So a practice can assemble certified technology from OpenEMR plus a separately certified e-prescribing product. Stephen's point is about visibility and simplicity: if the e-prescribing product is relied-upon software in OpenEMR's own certification, as EMR Direct is for Direct messaging, OpenEMR's listing itself shows the capability.*
>
> *As covered on 10/13, ONC's HTI-4 rule moves (b)(3) to NCPDP SCRIPT 2023011, with electronic prior authorization, and requires products certified to (b)(3) to also be certified to the new real-time prescription benefit criterion, (b)(4), from January 1, 2028. A certified partner that keeps pace with those deadlines carries much of that burden for OpenEMR.*

Who Needs Certified E-Prescribing?

- **Asked to recap what (b)(3) is:** it is the ONC certification criterion for e-prescribing. OpenEMR is certified, but not for e-prescribing, so users who take Medicare or Medicaid payments, or who are subject to state rules requiring a certified EHR, must use a separately certified product.
- **Is the requirement federal or state?** Stephen Nielson: it depends on whether a provider has a CMS billing requirement to use certified EHR technology. Small and rural providers have exceptions that keep being extended. Many states require a certified product for Medicaid billing, and CMS has certified-EHR requirements for larger providers.
- **In practice:** a provider today goes to LogiCoy or another vendor and prescribes through that vendor's website, or a developer integrates OpenEMR with a partner that holds (b)(3). Sherwin has built DoseSpot and MDToolbox integrations for clients who needed exactly that.

> *Three different requirements tend to get blended together here, and it helps to separate them. First, the DEA's EPCS rule (21 CFR Part 1311) governs how controlled substances may be prescribed electronically; any software that meets it will do. Second, the CMS EPCS Program, created by Section 2003 of the 2018 SUPPORT Act, requires prescribers to send at least 70% of their Medicare Part D Schedule II–V prescriptions electronically, measured each calendar year since 2023. Prescribers with 100 or fewer such prescriptions, or in declared disaster areas, are exempt. It too requires only DEA-compliant software, not a certified EHR, and enforcement so far has been notices rather than fines. Third, Medicare's Merit-based Incentive Payment System includes an e-Prescribing measure in its Promoting Interoperability category that does require technology certified to (b)(3), with an exclusion for clinicians writing fewer than 100 permissible prescriptions. Many states also have their own e-prescribing mandates. So certification matters most for MIPS reporting and for programs that require certified EHR technology, while EPCS mandates can be met with any compliant e-prescribing product.*
>
> *MDToolbox's e-prescribing product, MDToolbox-Rx, is Surescripts-connected and certified for EPCS. The vendor offers several integration styles — embedding via API, lighter web-service integration, no-code data sync, an HL7 interface, or an embedded screen in an iframe — which is why it, like DoseSpot, has been a practical choice for one-off client integrations.*

A Universal Interface — or SMART on FHIR?

- **A question was raised:** rather than each vendor reinventing the wheel, should core have a universal e-prescribing interface, with APIs that every vendor's module connects to? The interface could evolve on its own and would survive a vendor leaving.
- **Stephen Nielson's answer:** that front-loads a lot of work. The better direction is SMART on FHIR: if OpenEMR truly had complete APIs — as Michael and Eric have been pushing — an e-prescribing vendor could ship a SMART app launched with an e-prescribing intent, with a very limited footprint in OpenEMR. OpenEMR has three or four basic launch intents today.
- **Today's reality:** Weno and NewCrop established a pattern that newer integrations follow, still with quite a bit of custom code. Stephen would like to go back to LogiCoy and ask for a SMART on FHIR app, which could then be bundled as an example for others.

- **Who would build it:** it would be a big project with nobody available. Community members might take it on as a contract if LogiCoy paid, though LogiCoy would more likely use its own engineers.
- **Would LogiCoy want it proprietary?** Its module is open source and it wants it included in the codebase, so OpenEMR is in a position to say how it wants it built.

> *The SMART App Launch framework lets an EHR open a third-party app inside its own interface with context — the current patient, user, and encounter — and an optional "intent" telling the app what the user is trying to do. Very few intents are standardized, so an e-prescribing intent would be an OpenEMR convention. The approach also depends on FHIR writes: an e-prescribing app has to write the resulting MedicationRequest back into the chart, which connects this idea directly to the write work discussed on 10/6 and 10/13.*
>
> *The wider industry is moving the same way for prior authorization. HTI-4's new criteria for electronic prior authorization — (g)(31), (g)(32), and (g)(33) — are built on the Da Vinci FHIR implementation guides, which combine CDS Hooks with SMART apps, and HTI-4 also added CDS Hooks and FHIR subscriptions criteria. An e-prescribing architecture built on SMART would sit naturally alongside them.*

Ensora and MI-Squared

- **Asked how important it is to involve Ken and MI-Squared** before deciding, Sherwin said he has a strong opinion. They have been part of the community a long time, but the Ensora situation is holding the project back, and the question is how long to let that continue.
- **Brady: no code will be removed.** The plan is to add the better, certified option and direct people to it. There will be people still using the Ensora integration, and they should not be cut off.
- **No exclusivity:** if a partner wanted exclusivity, the project could not offer it, given Ensora's entrenched position and the number of users tied to it.
- **Direction:** don't embed e-prescribing options in core — except that a (b)(3) partner could be embedded as a core module if that is what certification needs. Long-term, Stephen Nielson would like every e-prescribing option to be a module. Weno already is; Ensora is the odd one out, and its code is small enough that moving it into a legacy module would be straightforward, perhaps with AI's help. From there it could be extracted from core entirely.
- **Next step:** one attendee will reach out to Ken.

> *According to the OpenEMR ePrescribe wiki page, MI-Squared has built and maintained the NewCrop (now Ensora) integration since 2016 and holds a DEA e-prescribing certificate for each OpenEMR iteration that is not valid for other vendors. That arrangement explains the "superposition" mentioned on the call: the integration lives in core, but its certification and support run through a single vendor.*
>
> *The exclusivity point follows from OpenEMR's GPL license: anyone may write and distribute an integration, so the project cannot grant exclusivity over code. What it can offer a partner is position — inclusion in the certified configuration, documentation, and call-outs on the project's pages — which is the kind of partnership discussed on 10/13.*

Partnership and Next Steps

- **Commercial questions** about a partnership — whether and how the project might benefit from being a distribution channel for a partner — were raised and will be discussed by project leadership.
- **How vendors are normally handled:** companies that approach the project through the forum are usually asked to become GitHub sponsors, and may list themselves on the Professional Support page. A vendor seeking a deeper partnership, as LogiCoy is, is a matter for project leadership rather than the weekly call.
- **Deferred:** the next steps with LogiCoy, including who engages with them, will be taken up by project leadership.

Help Offered, and AI Agents in Practice

- **Michael and Eric** both said they had little to raise this week. One said he would love the time and energy to revisit the module work, but other priorities come first. Stephen Nielson: “we just do what we can with the time we have.”
- **Michael offered help:** if anyone wants to ask about a specific PR or area of the code, he and Eric are open to it. Stephen Nielson replied that it is hard to say, because the project is being inundated with PRs.
- **Michael's own approach to the flood:** he writes skills “left and right” and sends AI agents after his inbox and GitHub notifications. Giving an agent access is a real risk, so his agent has its own tightly scoped, read-only credentials rather than his, and cannot open anything sensitive.
- **His personal setup:** Hermes, an open-source agent, running a DeepSeek V4-series model (heard as “V4.1 Flash”) through DeepInfra. He put $50 into a prepaid DeepInfra account in April and has spent about $8; in his view it performs very well for the price.
- **PHI:** asked whether he worried about PHI, Michael said the agent has no PHI access — it is his personal setup with read-only access to some GitHub material, and no access to work systems.

> *Hermes Agent is an MIT-licensed autonomous agent released by Nous Research in February 2026, designed to run on a user's own machine or server; DeepSeek's documentation includes a setup guide for using its V4 models with it. DeepInfra is an inference provider that serves open-weight models at low per-token prices, which explains the cost Michael describes. Password managers increasingly offer service accounts designed for exactly this pattern: an agent gets its own token, limited to what it needs and often read-only, and every access is logged.*
>
> *Are there problems with relying on AI too much? — has a few well-understood answers. Agents that read email, issues, or web pages ingest text written by others, and that text can carry instructions; this indirect prompt injection is the top item in OWASP's 2025 list of risks for LLM applications. The defense is what Michael describes: least privilege, so a misled agent cannot do much (OWASP's “excessive agency” item). The second risk is human: automation bias, the tendency to accept a fluent machine answer without checking — Brady's admission later in the meeting about glancing at test files is a mild example. The third is cost and continuity: cheap models make agents affordable, but workflows that depend on one provider's pricing or behavior can change overnight. None of these argue against agents; they argue for keeping humans at decision points and agents on narrow permissions.*
>
> *On PHI specifically: under HIPAA, a cloud AI service that receives protected health information needs a business associate agreement, which is why keeping a personal agent setup entirely away from PHI, as Michael does, is the right line. The same applies to anyone in the community experimenting with agents on clinical data.*

Architecture in the Age of AI-Generated PRs

The Concern

- **Stephen Nielson's "deeply unsettled view":** how can maintainers keep architectural awareness in mind with so many PRs arriving? It is hard to make judicious decisions about where the codebase is and where it needs to go.
- **Michael Smith:** "a question of the ages." His current answer, subject to rapid change, is agents doing regular architectural analysis. It is a forest-and-trees problem: contributors open PRs at the level of trees, so the project also needs agents working at the level of the forest — and those are not free.
- **Eric Stern looked at the open PRs during the call:** the first page was largely first-time contributors, going back several pages, and among those he skimmed he could not identify a single human. Some may be autonomous agents with no person behind them. He suggested considering GitHub's recently added settings that limit first-time contributors, at least while the architecture settles.

Limit New Contributors?

- **A fine line:** it was noted that such a limit would affect genuine newcomers, but someone pointing an AI agent at repositories to inflate their contribution count is not really a contributor, and once a contributor gets one PR merged the limit stops applying.
- **Brady would rather not limit:** he had the same instinct about the flood of security reports — surely most were junk — but almost none have been declined; they are real. He expects most of these PRs are of moderate to high quality. The problem is bandwidth, and the answer is using AI to manage it. He would happily limit junk PRs, but these are not junk.
- **Michael's framing:** with the checks the repository already has, the project is unlikely to get bad code at the level of trees; what it could get is bad architecture. That needs a different amount of context, and OpenEMR has 25 years of architectural divergence that even the best agents struggle with.
- **Michael's proposal:** a third level of review — agents that write and discuss architecture documents, plus a custodian agent that prunes outdated material and keeps the documents compact, so agents get more useful context. Meanwhile, accept PRs once they have been through the wringer, and keep investing in tooling like PHPStan and tests, which scale review. Ideally human review becomes rarely necessary, PRs are accepted as long as they introduce no security problems, and there is another pass between merging and releasing.

> *GitHub has added several controls this year. In February 2026 it introduced repository settings to disable pull requests entirely or restrict them to collaborators. On June 17 it added a cap on the number of open PRs from users without write access, with a bypass list for trusted contributors; draft PRs do not count. A June 29 setting can restrict issue creation to users with write access. The older interaction limits remain: for 24 hours up to six months, a repository can restrict commenting, issues, and PRs to prior contributors — users who have already contributed to the default branch — which matches the "one merged PR and the limit goes away" behavior described on the call. The newer PR cap applies to everyone without write access, not only first-timers.*

Running Documents for AI Context

- **Brady's practice:** because AI is limited by its context, he has AI continually audit and update documentation in the docs directory — running documents describing the frameworks, so an agent can orient itself without auditing the whole codebase each time. CodeRabbit already does some of this. His API work has a large working document that he will add to the docs directory when it is done, so the next AI to pick it up has it. These documents would be produced and kept current by AI, with human oversight.
- **Brady largely agreed with Michael:** with the tooling now in place plus CodeRabbit, code that passes usually comes out fine. Another level of audit for fit with the architecture would make it better. Security-critical areas get extra protection, which is why he is pushing OAuth regression testing.

> *The transcript is garbled here, but the discussion appears to turn to AGENTS.md — rendered as "workday files" — and the relief that one file can now serve every agent. AGENTS.md is a plain Markdown file of project instructions for coding agents, released by OpenAI in August 2025 and contributed in December 2025 to the Agentic AI Foundation, a directed fund of the Linux Foundation co-founded by OpenAI, Anthropic, and Block. By then it had been adopted by more than 60,000 open-source projects and agent frameworks.*
>
> *Michael's custodian point is well founded. Long, accumulating instruction files degrade agent performance: contradictory or stale guidance competes for attention in a limited context window. One technique mentioned near the end of the call — asking an AI to read the project's guidance and list inconsistencies — tends to surface exactly the contradictions that make agents "go off the rails."*

Enforcing Architecture

- **Stephen Nielson wants a way to flag or close PRs on architectural grounds.** Everything needs to move toward MVC-style separation — presentation separate from business logic, separate from data access. He is almost ready to start rejecting PRs that touch the regular legacy PHP files, because he spends much of his time reviewing small fixes to individual files when the codebase needs wholesale architectural change: APIs, service classes, and perhaps Doctrine. An architecture document for AI would let the project close PRs, or have AI explain to contributors what is wrong with their architecture.
- **Brady:** that is a tough one. He leans toward a quicker trigger finger — accept a PR if it is reasonable and passes CodeRabbit — while acknowledging that continues the "big ball of spaghetti." The better answer is a step that iterates on the framework and suggests improvements to a PR, such as converting to Twig, but there is no mechanism for that yet.
- **Eric Stern shares Stephen's concern:** local bug fixes are fine, but if big-picture fixes never land, the project keeps building in the way it is trying to leave behind — and AI is automating that at a scale never seen before. A middle ground: identify what it would take to ensure all new UI goes through Twig and all data access goes through the API layer rather than inline SQL.
- **Tests that cement the wrong design:** Eric has noticed recent PRs that add test coverage but bake in behavior the project does not want going forward, making later improvement harder. Brady admitted he sees test files and thinks "yay, test coverage" without looking further. The worry voiced: AI agents will then bend future designs to keep those tests passing.

> *Eric's point has a name in the testing literature. Tests that record what code currently does, rather than what it should do, are characterization tests — Michael Feathers' term from work on legacy code. They are a valuable safety net before refactoring, but if they are mistaken for specifications they lock in the very behavior a refactoring is meant to change. Labelling them, or keeping them separate from tests of intended behavior, preserves the safety net without the lock-in.*
>
> *Several established techniques could turn an architecture document into something enforceable. "Fitness functions," from the book Building Evolutionary Architectures, are automated checks that guard an architectural property. In PHP, custom PHPStan rules or a layer-dependency tool such as Deptrac can fail a PR that adds inline SQL to a template or calls the database from presentation code. The strangler-fig pattern — routing new behavior through the new architecture while legacy code shrinks around it — is effectively what the front controller and API-first work describe. Rules that apply only to new or changed code avoid demanding that every small fix rewrite its surroundings.*

Next Step — An Architecture Document

- **Stephen Nielson will put together an architecture document,** building on the agent instruction files that already exist, and submit it as a PR for people to review — aiming for the coming week — to see whether it brings more sanity to the process.
- **Cost:** running forest-level agents takes credits; cheaper models make it more feasible.

> *This continues a thread from 9/29, when Eric set out an order of preference for extending OpenEMR — standard APIs, then OpenEMR's own APIs, then a module, then core — and from 10/6, when he argued that enforcing authentication and authorization in one place rather than 300 would make the project far more confident. An architecture document is the human-readable form of those decisions; the enforcement techniques above are how it could scale to hundreds of PRs.*

Issue #2119 — Speeding Up the Logging Engine

Issue: https://github.com/openemr/openemr/issues/2119

Simon's Question

- **Simon plans to tackle issue #2119** and asked whether anything not recorded on the issue comes to mind about making logging more efficient, or whether he should simply pick a direction. In the chat, Eric said he has done some work in that area without knowing the issue existed, and may have input depending on the approach.
- **One quick suggestion:** an index on the log table's date column.
- **Next step:** Simon was asked to post a plan on the issue as a basis for discussion; he offered to open a PR with the direction instead, and either was fine.

> *Issue #2119 is several years old. It describes OpenEMR logging every SQL query at a low level, which slows large installations, especially those running the extensive logging that certification-oriented practices turn on. Brady reported a first step committed in PR #2168, with optimization discussed in #2130, and Rod Roark of Sunset Systems suggested archiving and dumping older entries because the table grows so large.*
>
> *An index on the date column is a trade-off rather than a free win. It speeds up audit reports and date-based purges, but every insert must also update every index, and logging is overwhelmingly a write workload. On a busy system each extra index adds cost to every logged query — exactly the path the issue is trying to make faster.*

Mechanics Versus Requirements

- **Context from OCE's attendees and Stephen Nielson:** the mechanics of the logging engine and what actually needs to be logged are not the same thing today. In effect, every query generates a log insert. That is not what the requirements dictate.
- **Log meaningful events instead:** someone accessed PHI, someone signed something — the actions the ONC specifications require. That would cut log volume dramatically. The inserts themselves are not slow, but on a busy installation the log table can grow to many times the size of all other data combined, most of it not needing to be logged.
- **Why it was built this way:** the original design avoided making every call site decide whether to log. Logging everything is the ultimate safe default — nothing is ever forgotten.
- **A missed opportunity:** logs are naturally append-only, but MySQL's standard storage is not. Truly append-only storage is very efficient and removes a lot of architectural difficulty, and it makes log rotation straightforward. Making the logger append-only, at least optionally, would make logging far more efficient.
- **The ARCHIVE engine:** MySQL has an append-only ARCHIVE storage engine. A comment in the chat: it looks promising, but may not be enabled by default in all installations — based on a two-minute read of the documentation, not first-hand experience.
- **The longer-term answer:** as the project moves toward an API architecture with a centralized permission model, the event audit logger becomes a natural hook into the permission check — "I'm about to do a thing" goes to an engine that decides whether it needs logging — which would cut volume massively. It is a big architecture change. "So, Simon, we just want you to rewrite the system." Simon: "Give me a couple weeks." He was jokingly offered a $50 DeepSeek gift card.

Select Logging and Performance

- **Simon's reading of the requirements:** he does not think ONC certification requires logging every SELECT, though he would check the exact text. A client of his reported a much slower system in the past few days; he had turned on select logging, and switching to a patient from the calendar went from about 5 seconds to 20–25 seconds.
- **Michael Smith:** loading a patient can involve around a thousand database queries. OCE has tried to contribute all its fixes upstream, though some got lost in the shuffle. OpenEMR has many N+1 query problems — a busy calendar page can take 30,000 queries instead of four. Fixing those pages makes logging faster for free, and turning on select logging is one way to find them.
- **Stephen Nielson:** select logging carries a big overhead because every read creates a write, which thrashes the indexes; administrators who enable it need to retune InnoDB's buffers. The logger already has event categories that decide what gets logged, and more logic could go there. He has not used ARCHIVE in 12 to 15 years and could not advise on it.
- **Eric** will keep working with Simon, since the issue touches areas he has worked in.

> *How OpenEMR decides today, from the EventAuditLogger code: logging is governed by globals — enable_auditlog as the master switch, audit_events_query for SELECT statements, and per-category switches for patient records, scheduling, orders, lab orders, lab results, security administration, and other. SELECTs against uncategorized tables are never logged, and break-glass users can be forced to log everything. An optional ATNA connection (enable_atna_audit) forwards audit events to an external repository using the IHE ATNA profile. So the category machinery Stephen mentions already exists; the question is what level it works at.*
>
> *What certification actually asks for supports Simon's instinct. Criterion 170.315(d)(2) requires recording actions related to electronic health information using content from ASTM E2147-18 — date and time, patient, user, type of action (additions, deletions, changes, queries, print, copy), and the data accessed — plus tamper resistance and the ability to detect alteration. HIPAA's audit-controls standard (45 CFR 164.312(b)) is similarly framed around activity in systems containing PHI. Both describe actions on patient data at a meaningful level, such as "user X viewed patient Y's medication list," not a row for every SQL statement. Event-level logging could plausibly meet the requirements with a small fraction of today's volume, though anyone pursuing it should confirm against the test procedure.*
>
> *The ARCHIVE engine fits half of the problem. It accepts INSERT and SELECT but not UPDATE or DELETE, compresses rows with zlib, and allows an index only on an AUTO_INCREMENT column; it has no transactions, and every SELECT scans the whole table. That makes writes cheap and the storage naturally tamper-resistant, but makes audit reports by patient or date slow. The chat comment was right about availability: in MariaDB, which OpenEMR's Docker images use, the ARCHIVE plugin ships but is not installed by default. A common design pairs a small, indexed "hot" log in InnoDB for recent activity and reporting with periodic rotation of older entries into ARCHIVE tables or compressed files, so retention and reporting needs are met separately.*
>
> *An N+1 problem is a page that runs one query to fetch a list and then one more query per item — a schedule of 200 appointments becomes 201 queries, or many more when each appointment fetches its patient, provider, and status separately. The fix is to fetch related data in batches or with joins. Select logging multiplies the cost, because each of those reads also becomes a logged write, which is why fixing N+1 pages and slimming the logger reinforce each other.*
>
> *The permission-check hook connects to the 10/6 discussion: if authorization is enforced in one central place, that place knows exactly which user is doing what to which patient's data — precisely the content ASTM E2147 asks for. FHIR has a matching resource, AuditEvent, so the same events could later be exposed in a standard format.*

Simon's PRs — Ready for Review

- Simon has 13 open PRs, five of them fully green and ready; he hopes to have all 13 ready by next week. They were not reached on the call. The five ready PRs, from the notes:

PR	What it does
13852	Fits printed patient statements to double-window envelopes (#9, #10, or custom sizes) via a new Billing global. Stephen Waite requested changes — move the globals callback into a class under src/ that renders a Twig template, and move the JavaScript into a separate asset — and Simon reports this is resolved.
14253	Improves the 837P claim generator's warnings when billing or service-facility ZIP codes are not nine digits, explaining the rejection risk; foreign postal codes get their own warning. Holding such claims was split into a separate PR.
14254	Sends NPIs as digits only in 837 claims and UB-04 fields, stripping hyphens and spaces that the NM109 element does not allow.
14393	Stops an infinite loop when a repeating appointment's next date does not advance, which could leave the calendar and flow board walking the series forever; adds tests.
14737	Wraps facility saves in database transactions and reports “not saved” if the facility row is missing, instead of a silent false success.

> *Four of the five are billing and scheduling fixes from a working practice — the kind of small, well-tested PR that the size-first triage discussed on 10/13 favors. They also illustrate the architecture debate above in miniature: 13852's requested changes, moving logic into a src/ class with a Twig template, are exactly the “convert to Twig” guidance that an architecture document could give contributors automatically.*

Items Discussed and Proposed

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

Area	Discussed / Proposed
Release 8.5.0	Release branch to be forked this weekend; release targeted for next Tuesday or Wednesday. Jerry to be told.
Inferno	Running in CI on merges to master. Cannot validate PRs before merge.
API testing	Coverage added to the portal API, standard API nearly done, FHIR next, including negative tests. FHIR metadata to describe sorting and paging accurately.
E-prescribing goal	A certified (b)(3) companion product, part of OpenEMR's certification as EMR Direct is. Agreed.
DAW	Advised to build and maintain its own module. To be invited to a weekly call.
Integration approach	Long-term preference for SMART on FHIR apps over a bespoke e-prescribing service layer.
Ensora	Not removed; long-term move into a module. MI-Squared to be contacted about a certified path.
LogiCoy partnership	Commercial questions and engagement referred to project leadership.
AI agents	Michael's practice: agents on narrowly scoped, read-only credentials; no PHI.
PR volume and architecture	Limiting first-time contributors raised; no decision. Stephen Nielson to draft an architecture document as a PR.
Logging engine	Direction favored: log meaningful events rather than every query; fix N+1 pages; explore append-only storage.
Simon's PRs	Five ready for review; not reached.

Future Discussion and Next Meeting Agenda

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

Item	Details
Simon's PRs	Five ready (13852, 14253, 14254, 14393, 14737), with up to 13 by next week. Suggested as the first PR item.
Logging engine (#2119)	Simon's plan or PR; Eric's earlier work in the area.
DAW	Presentation, if Sherwin's invitation is accepted.
LogiCoy	Report back on direction after project leadership discussion.
Ensora / MI-Squared	Outcome of the outreach about a certified path.
Architecture document	Stephen Nielson's PR for review.
Contributor limits	Whether to use any of GitHub's contribution controls; discussed without a decision.
Release 8.5.0	Status after branching, and any glitches from the new release tooling.
FHIR testing	Brady's consolidation of the FHIR test and fix PRs.
Carried over	Multisite, subdirectory, and non-Docker deprecation with Eric's migration guide, and procedure writes over FHIR — pending since 10/6.

Notes

- **The connecting thread:** three apparently separate discussions — e-prescribing, the PR flood, and the logging engine — ended at the same answer. A clean API layer would let an e-prescribing vendor ship a SMART app instead of custom code; a centralized permission check would let the logger record meaningful events instead of every query; and an enforced architecture would let AI-generated PRs strengthen the codebase instead of entrenching legacy patterns. That is the foundation Michael and Eric have argued for since the summer, now arriving from three directions.
- **Two views of the PR flood that are not really opposed:** Brady prioritizes throughput and trusts tooling to keep quality up; Stephen Nielson and Eric worry about direction. Both agree that the gate belongs in automation rather than in human review of every PR — they differ on what the gate should check. An architecture document serves both: it gives the tooling something to enforce.
- **E-prescribing has a clear goal now:** a certified (b)(3) companion product, with every other option welcome as a module and nothing forcibly removed.
- **Carried forward:** the deprecation plan and procedure writes have now waited three weeks.
- Contributors — especially those working with AI assistance — can help most by routing new UI through Twig and new data access through the service and API layers, and by keeping tests focused on intended behavior. Anyone with experience running audit logging at scale on MySQL or MariaDB could usefully weigh in on issue #2119, and practitioners who e-prescribe in OpenEMR remain the best judges of what a certified option needs to do well. Comments on the forum or at the weekly call are welcome.