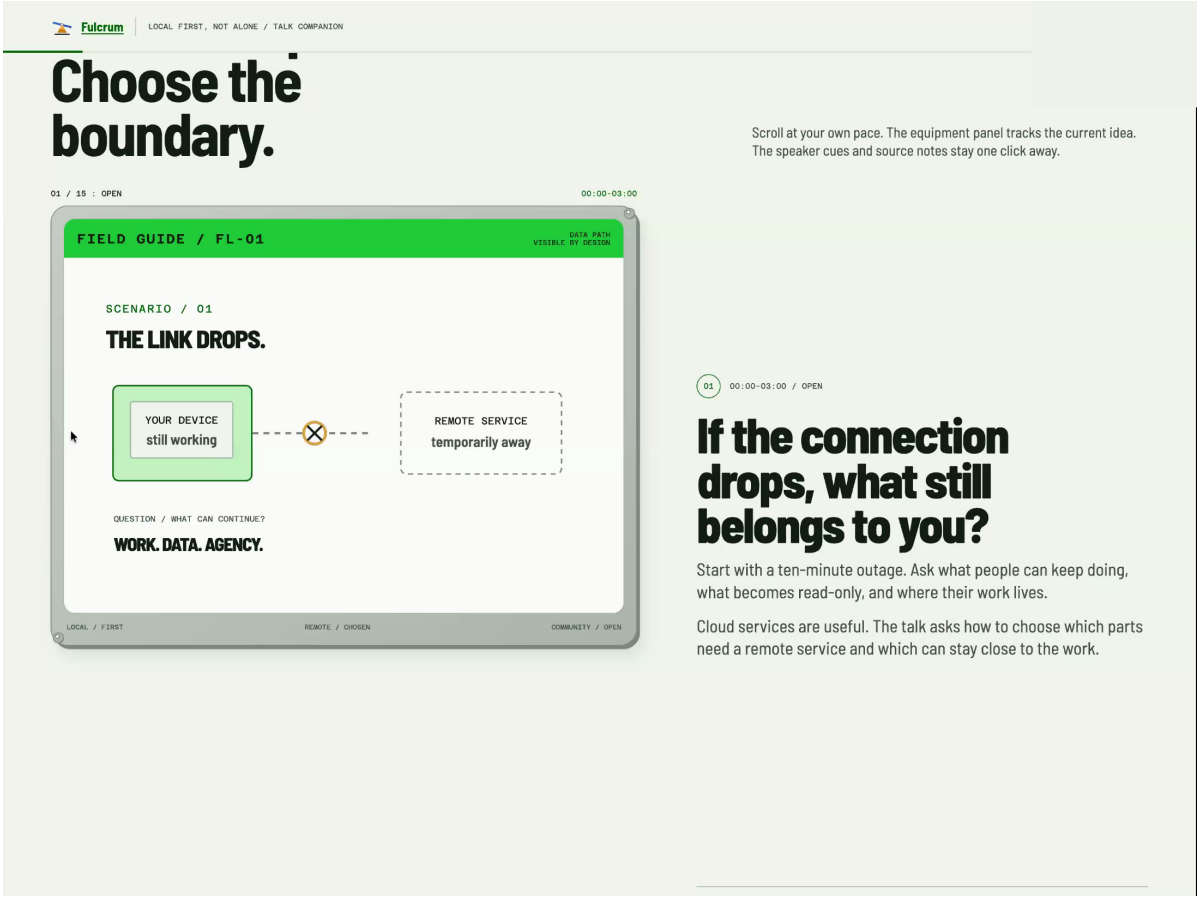
Fulcrum：本地优先的智能体控制平面

Mo Abualruz · 2026-09-26 · 中文译文

演讲与问答全文

演讲与问答可读版逐字稿 · 中文译文。2026 年 9 月 29 日二次校订：对照 Soniox 转录、Zoom 字幕和演示材料，整理重复起句、断句及术语，保留发言顺序、例子、限定条件与讲者观点。中文逐段对照英文校订。原录像从讲述中途开始，缺失开场未补写；时间对应交付视频。部分疑点音频另经 Soniox 复核。仍无法确认的表述已标注；未逐句人工回听。

断网之后，还能做什么



现场演示画面 · 00:01:00

00:00:00 · Mo Abualruz

互联网突然断了，这时候你怎么办？停止工作，什么都不推进了吗？这就是本地优先的系统设计会发挥作用的地方：即使遇到服务中断，我也必须能访问继续工作所需的数据，至少能读取它们。我并不认为答案只有一个，比如“把一切都放在云端”或“把一切都放在本地”。你需要考虑各种边界情况，让任何一处故障都不至于造成灾难，而且能够恢复。即使某个环节受阻、某项服务停机，你仍有办法继续工作。这是思考自己的工作环境时很重要的一点。

00:01:07 · Mo Abualruz

这种思路也适用于软件设计：你可以把软件分隔成不同部分，进行分层。安排自己的工作时，也应该准备相应的方法和环境，尽可能避免被卡住，让工作继续推进。如果大家对本地优先、自托管，或者这方面的做法有问题，也可以提出来。

00:01:47 · Mo Abualruz

我的思路是：自托管一些服务，让它们与云端互补，也在发生中断时提供备用方案。

00:02:00 · Mo Abualruz

具体的设计思路是——把互联网上或云服务商那里对你至关重要的东西，保留一份副本。你也可以把家庭实验室当作一个云服务商。比如，如果你已经部署并备份了一个好用的任务管理系统，就不一定非要用 Jira。有一个很不错的开源项目，叫 Plane。

00:02:33 · Mo Abualruz

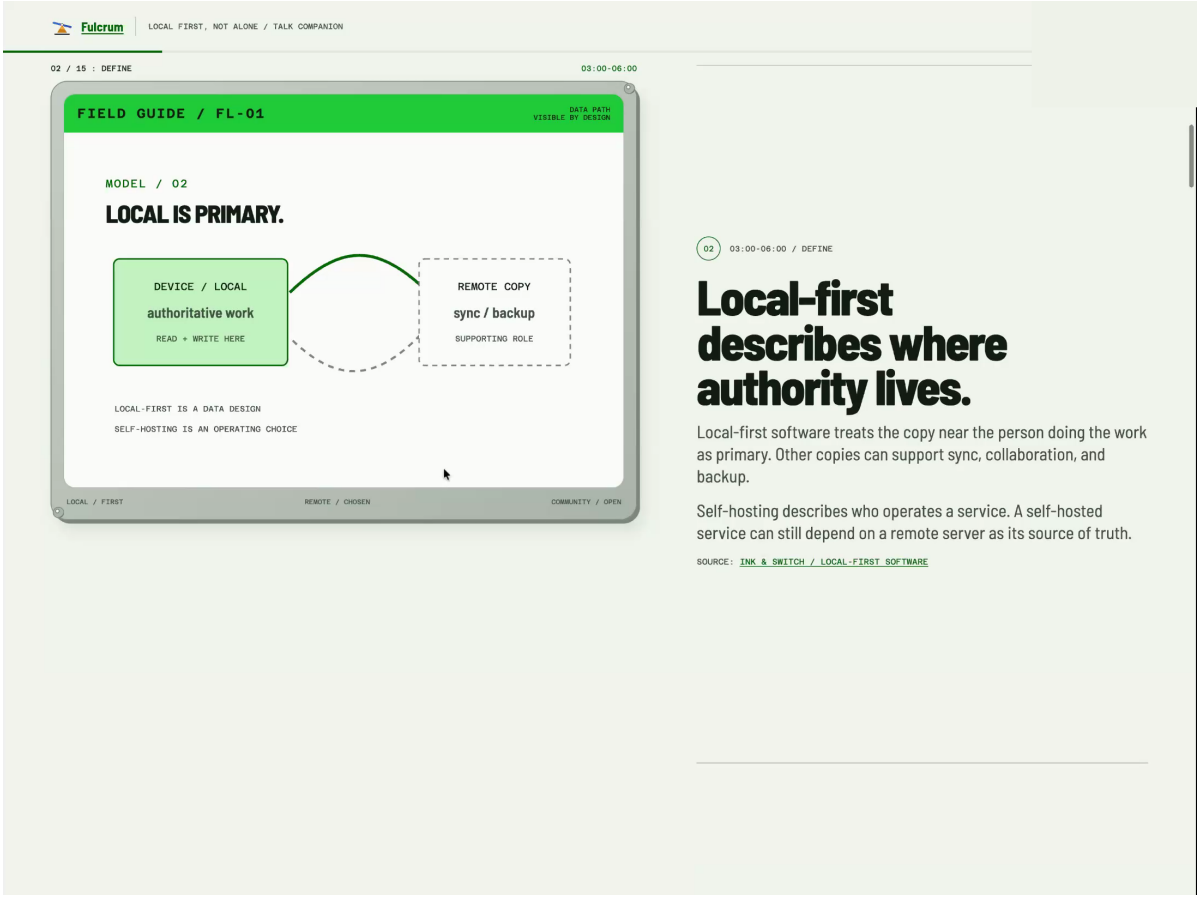
它几乎具备 Jira 的所有功能，但它是开源的。你可以自行托管、与其他系统集成，也可以邀请朋友或同事，一起做项目和规划。不过，做到这些还不够。

00:02:51 · Mo Abualruz

如果运行服务的那台机器出了问题，你还需要一些 Markdown 文件、JSON 文件之类的东西来记录工作，让你仍然能够查看当前进度："服务停机之前，我们做到这里了。"这样就不必停下。这就是本地优先的思路：保留能让工作继续推进的数据副本。我自己托管了大多数服务，但也安排了不同层级。例如，本地大模型负责一些工作、管理记忆；如果本地服务不可用，就考虑切换到云端模型。［原句关于能否连接云端的表述有矛盾，此处按上下文整理。］

00:03:35 · Mo Abualruz

每个人都需要先弄清楚，自己通常在哪些地方被卡住，然后设计办法，让工作尽可能继续，让自己保持投入，并清楚当前正在做什么。特别是在 CLI 会话或聊天会话里，系统可能给你一大段文字，说接下来要做第一、第二、第三步。我会增加一些图来帮助理解，因为到了一定程度，单靠文字很难让人跟上整个过程。我的基本想法是：让你能够继续工作的那些关键信息，大多只需要读取，所以它们必须在你当前使用的设备上、本地可用。家庭实验室里的机器也是一个服务提供方。如果你离开家还需要工作，就必须在随身设备上保留本地数据，甚至不能把能否继续工作完全寄托在家里的服务器上。



现场演示画面 · 00:04:30

00:05:00 · Mo Abualruz

比如，出门旅行时，你也应该能继续做事，而不是被卡住。有一篇介绍本地优先设计的好论文，我可以把链接发到聊天里。现在显示的是全屏吗？这篇论文很值得分享。会后我也可以把演示文稿分享出来，里面有我使用的全部参考资料。

00:05:38 · Mo Abualruz

这篇论文讨论什么是本地优先软件，以及应该怎样思考它。基本想法就是：保留属于你自己的副本。现在我们出门会带信用卡，但随身带一点现金仍然有用。不是每一家商店、每一座城市，也不是所有地方都一直联网。

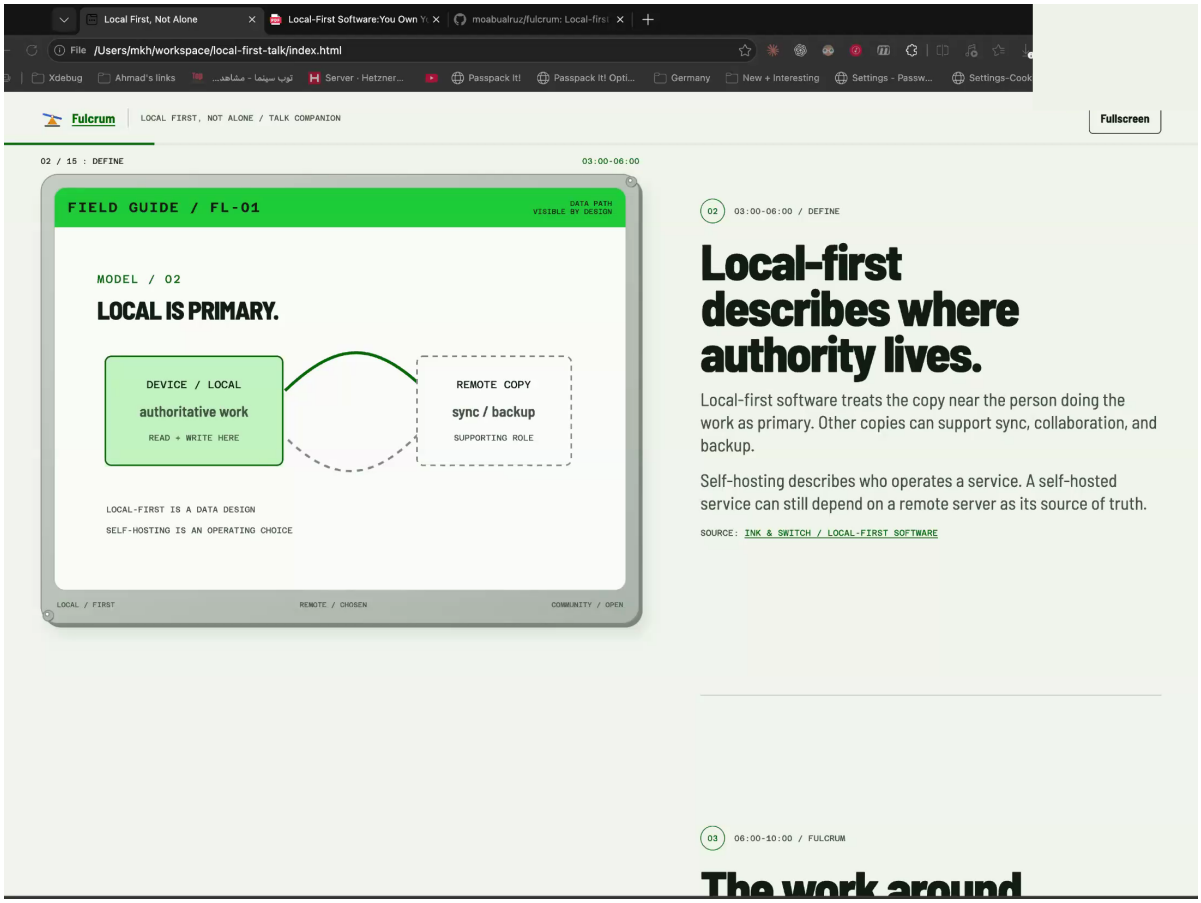
00:06:05 · Mo Abualruz

甚至不是每个国家都有相同的支付条件。可以用这个比喻来理解：不管服务是家庭实验室里自托管的，还是云端的，你都可以使用“信用卡”；但最后还要有一点“现金”，以防万一。出了问题，你仍然有自己需要的东西；没有出问题时，同一套系统持续给你提供同一份数据，你也不必为了得到相同的信息，在多个系统之间来回操作。也就是说，通过统一的交互入口管理工作。

00:06:48 · Mo Abualruz

统一交互入口还有一个关键原因：你最重要的、作为依据的数据，包括上下文、记忆和代码，都应该掌握在自己手里。比如，当前正在做的工作才是最新的，而不是 GitHub 上的那一份。本地副本里有最新任务、正在开发的功能以及新的想法。

智能体周围的工作



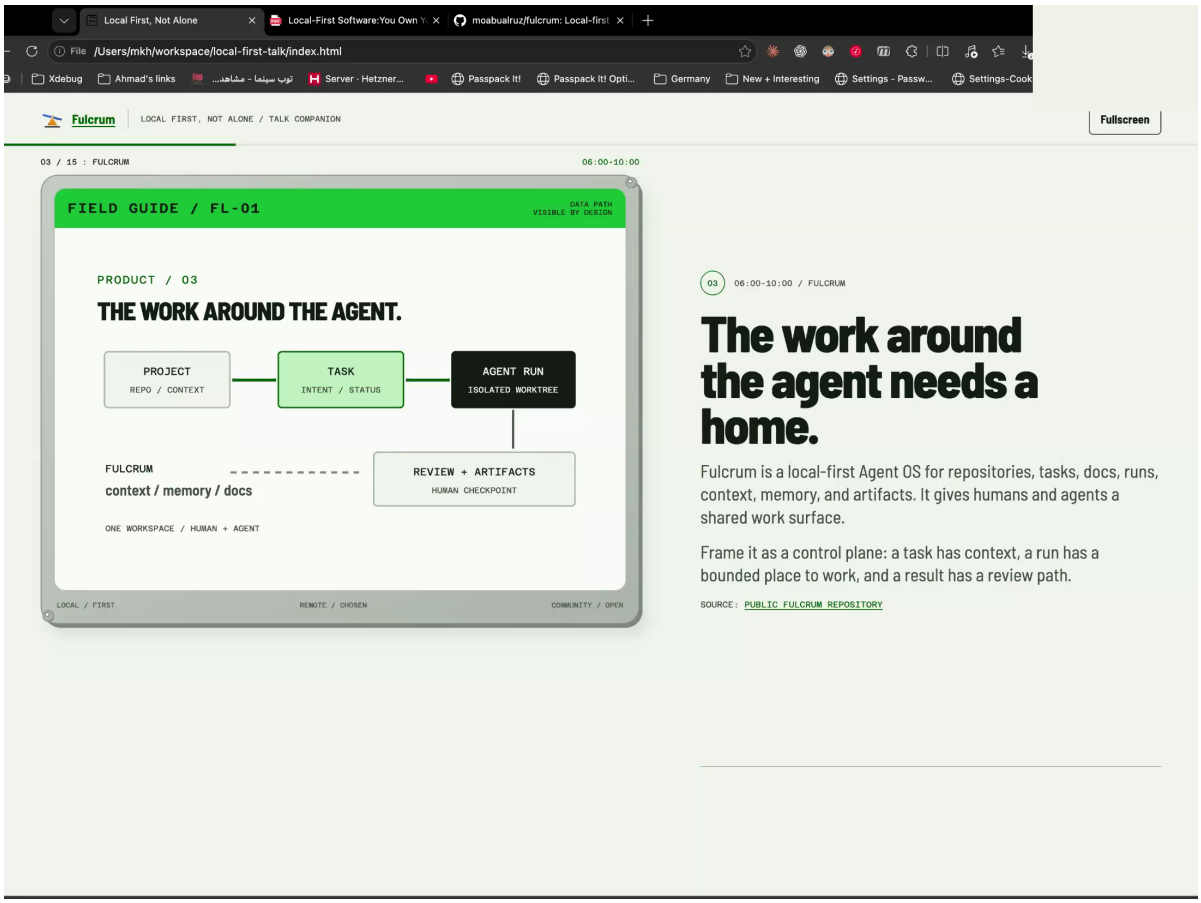
现场演示画面 · 00:07:30

00:07:23 · Mo Abualruz

这些都记录在本地。即使 GitHub 停机，你的机器上仍有代码，代码库和文件都在，就可以继续工作。但如果你依赖的是 Honcho——这是一个开源项目，也提供可以订阅的记忆管理服务——断网后就无法向它查询记忆。所以你可能还需要一些本地记忆文件，或者用小模型运行一个 Honcho 实例，把云端 Honcho 中的数据复制下来。

00:07:58 · Mo Abualruz

本地的小模型也许没有云服务商提供的模型强，但至少你保留了副本，可以继续工作。这就是构建 Fulcrum 时的想法，也是我理解的本地优先。对于智能体工作流，我主要按项目和任务拆分工作，再把每一次智能体运行看作一次具体执行。



现场演示画面 · 00:10:30

00:08:31 · Mo Abualruz

任务可以是规划，也可以是评审。回到 Fulcrum——抱歉，我讲得有点跳跃——工作推进到某个阶段时，就需要评审。我有技术背景，所以会亲自检查。但如果你没有那么熟悉技术，可以先请 Qwen 完成任务，再请云端能力更强的 Kimi 模型［具体版本听辨不清］评审。这样，大部分任务和代码由 Qwen 27B 来完成，成本会低一些：先用小模型做，再用更贵的模型检查。

00:09:25 · Mo Abualruz

我可能没有足够的技术经验，无法发现全部错误；也可能正在学习一门新语言，需要借助 Kimi 的反馈。例如："原来在 Go 里应该这样做，这种写法是一个不好的模式。"小模型犯了错，你可以通过前沿模型给出的评审意见学到东西。［原口述此处对模型名称有重复，按前后评审关系整理。］

00:09:51 · Mo Abualruz

Fulcrum 必须把相关内容都连接起来：记忆、文档、任务上下文，以及评审所需的信息。不能只是丢过去一个 PR，说"帮我检查一下"。无论请人还是请智能体评审，都不能只给一个 PR。根据我做工程经理的经验，我们总会先问："你正在做什么？""这个 PR 的价值是什么？你想解决什么问题？"这些信息不仅对人重要，对执行任务或评审工作的智能体也重要。它需要理解你究竟想实现什么。随便改一些代码、随机生成一些 token，并不等于完成了工作。你必须确认，这些工作确实为你创造了价值。

00:10:44 · Mo Abualruz

把这些事情讲清楚，后面的工作就容易得多。我会引导智能体遵循一种带有测试驱动特点的工作方式，但更准确地说，它是价值驱动的。相比让智能体大量编写单元测试，我更希望它验证实际价值，也就是进行端到端测试。

00:11:10 · Mo Abualruz

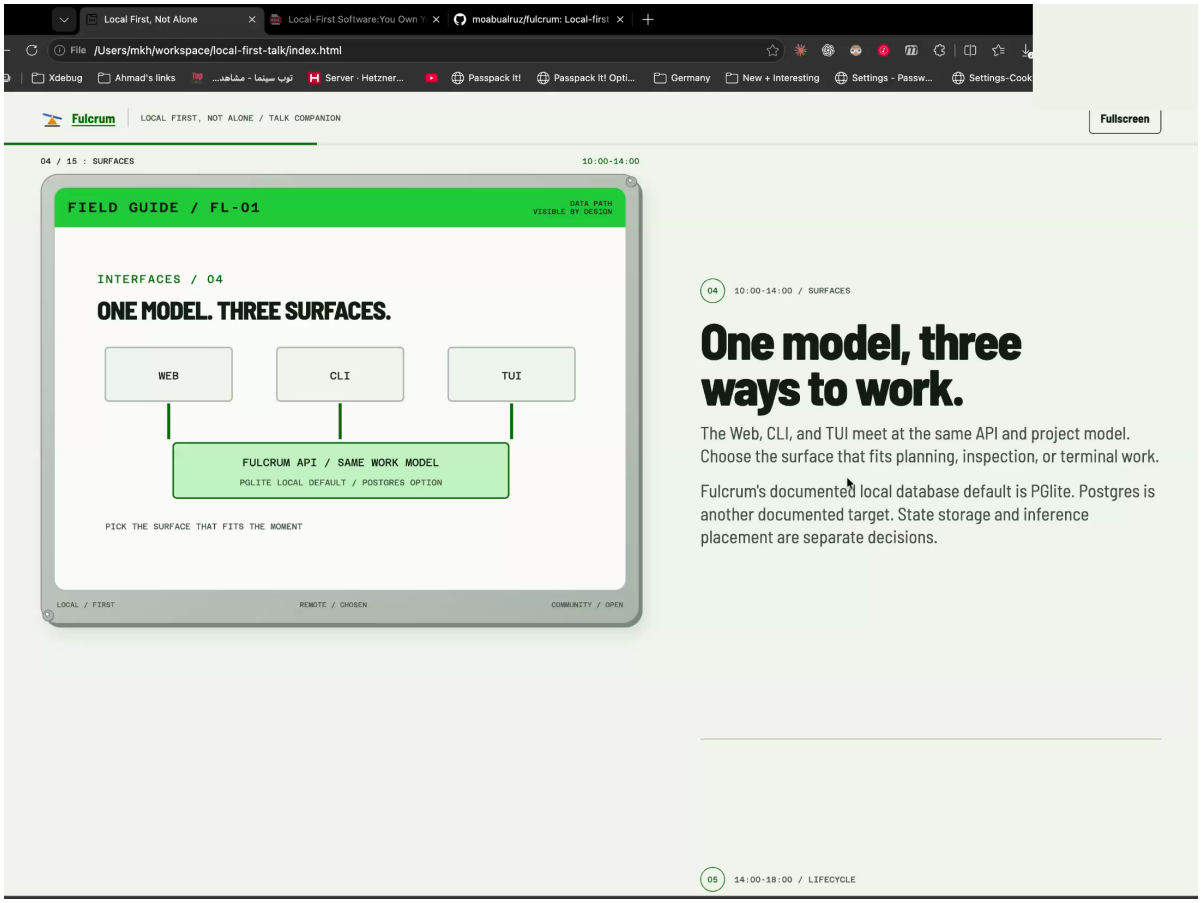
我要它确认用户工作流、用户故事是否正确，以及我描述的所有价值，是否都能通过一个可测试的流程来验证。我擅长代码，所以希望智能体和 AI 补充我的能力：代码评审、技术判断我自己能做得不错，而 AI 对我最有价值的帮助，是检查用户侧的各种边界情况，验证改动是否有效。抱歉，刚才有一个电话。所以我会继续在 Fulcrum 中增加工具，覆盖更多人实际需要的智能体工作流。

00:12:11 · Mo Abualruz

这是一个开源项目。我加进去的功能，是为了补充自己的技能和经验。但主持人开始使用 Fulcrum 时，他有不同的技能，也许需要额外的功能来支持他的工作。现有功能可能已经有用，但他仍可以增加一两个功能来适配自己的流程。因此，我认为这类为工作提供上下文和可观测性的项目应该开源，不应该只停留在个人探索或私有代码库中。

00:13:02 · Mo Abualruz

所以，我也为 Fulcrum 提供了三种交互界面。TUI 面向喜欢终端、希望直接在终端中查看工作状态的人；CLI 则供人或智能体调用，执行具体操作；Web 界面方便我们完整地查看系统并与之交互。



现场演示画面 · 00:13:50

00:13:31 · Mo Abualruz

它们使用同一个系统、同一套 API、同样的调用，只是呈现方式不同。这样，人、智能体和共同工作的同事，都可以选择适合自己的入口。例如，两个人分别负责不同的事情，也可以通过这些入口与智能体协作。这就是这里展示的三种交互方式。

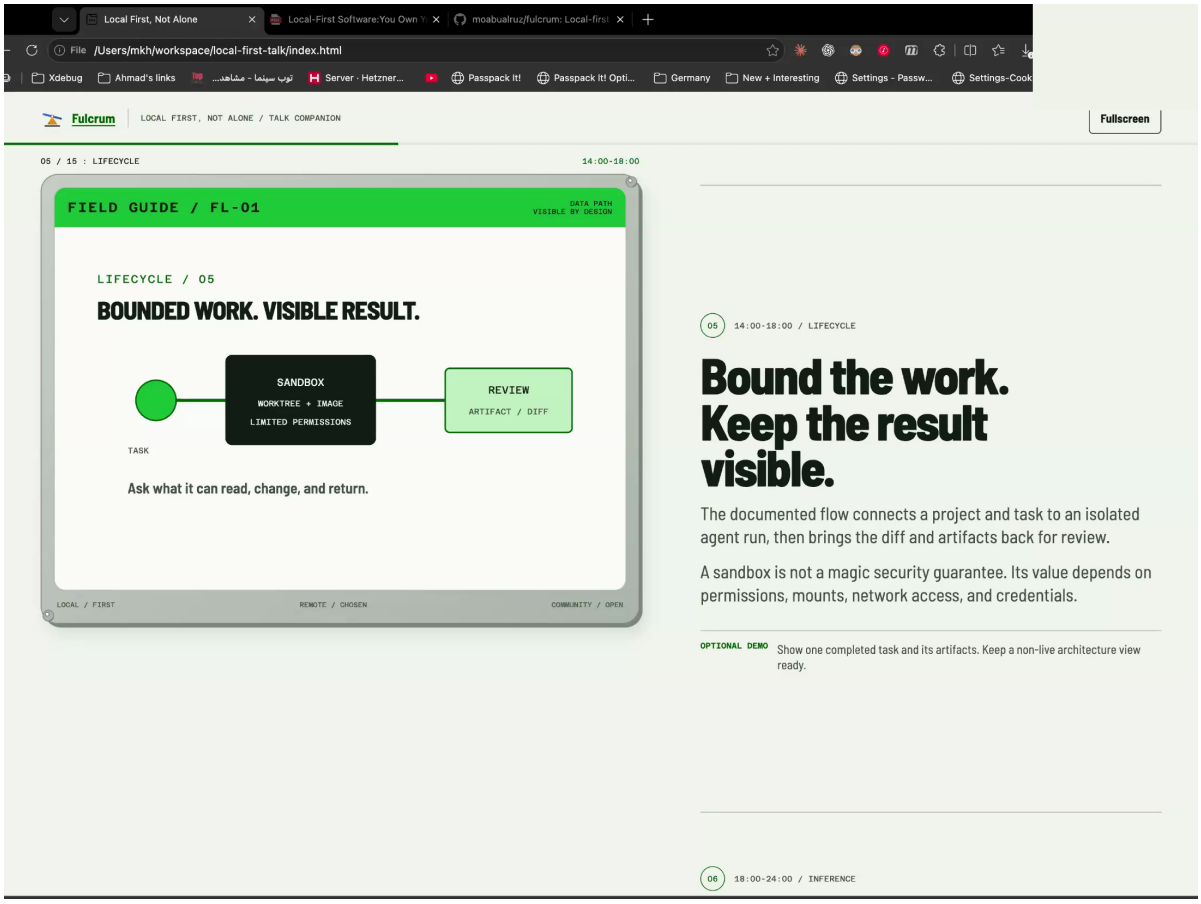
00:14:03 · Mo Abualruz

整体上都通过 JSON 通信。我不想深入所有技术细节：这里使用 gRPC，让 CLI 和 TUI 调用 API 更快一些。它给人一种事件驱动的体验：数据可以推送到前端，前端也能向后端传递数据，是双向通信。我觉得它比常规方式更快。不过，在握手方面，我认为它不如完整的 HTTP 交互可靠，所以不会用它与远程系统集成。

00:14:48 · Mo Abualruz

但当系统在本地运行，界面和进程都由你自己控制时，速度、清晰度以及进程封装方面的收益很适合这个场景。这也是我能从同一套底层能力提供三种不同界面的原因。

任务边界、结果与工作流



现场演示画面 · 00:16:00

00:15:08 · Mo Abualruz

接下来讲 Fulcrum 怎样组织实际工作。这对我很重要。我不太喜欢限制很死的沙箱，但又希望并行推进更多工作，给自己和智能体一定的修改自由，同时避免破坏已有内容。因此，我围绕 Git worktree，也就是 Git 工作树，构建了这层隔离。

00:15:36 · Mo Abualruz

无论是智能体还是我，都可以创建一个新的 Git 工作树，在里面自由修改、实验和制作原型，而不会破坏主工作区。各项工作可以独立进行。等完成评审并得到我的认可后，再把这些工作合并回本地的主分支。

00:16:02 · Mo Abualruz

这些都发生在本地。这里说的评审不是远程 PR 的代码评审，而是创建 PR 之前，对智能体或自己产出的结果进行检查，再将其合并回原来的分支或主工作区。这样的轻量隔离给我们修改和实验的自由，也尽量避免损坏当前工作，或让命令跑到不该操作的地方。我不希望工作目录里的操作被误当成针对整台机器的操作。

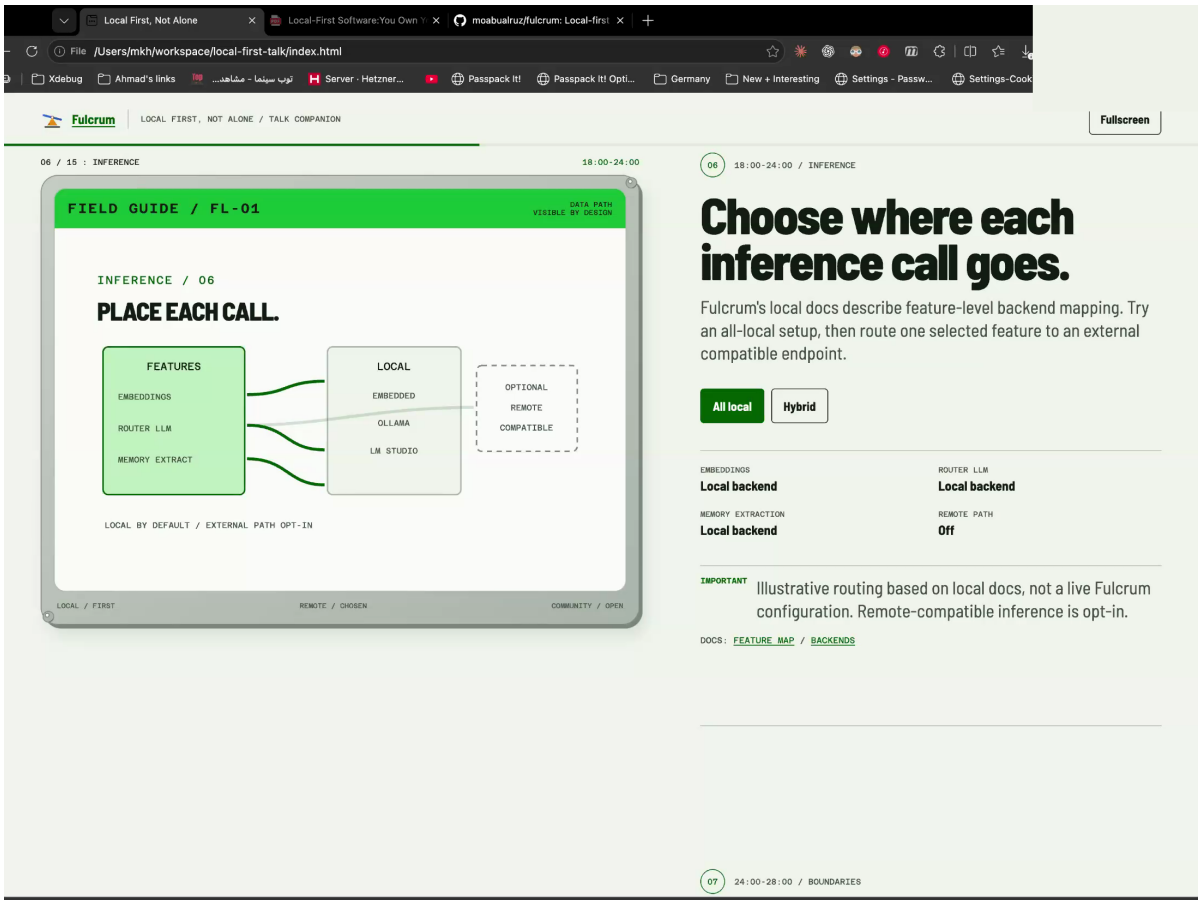
00:16:36 · Mo Abualruz

这里有明确的工作目录和 Git 工作树。我们先检查所有输出，再把它们合并到相应的工作分支。

00:16:50 · Mo Abualruz

再讲一下集成。刚才那张图放在哪里了？找不到也没关系，我可以解释。配置自己的 Fulcrum 时，可以把所有大模型交互都设为使用本地模型，也可以接入不同的 harness，也就是智能体运行框架，来执行工作。harness 自身有它自己的配置。

推理路由与可追溯性



现场演示画面 · 00:19:30

00:17:31 • Mo Abualruz

你可以让 harness 只使用云端推理，也可以让它使用本地模型；这些是在 harness 层完成的配置。Fulcrum 不会干预你如何配置它，因为这样能给使用者更多灵活性。例如，你希望编码用 Kimi，但规划任务用自己的 Qwen 模型就够了。如果我强制要求 harness 与 Fulcrum 必须使用同一个模型提供方，就会过于受限。至于 Fulcrum 自身需要的能力，例如把文档与任务联系起来，其中会涉及嵌入。

00:18:39 • Mo Abualruz

我不太会把它叫作 RAG，它更像一张图：我们在上下文节点之间建立联系。比如，这个文件与这个任务、这份计划有关。你可以从整体上看到各种数据如何连接，以及它们与实际工作的关系。把任务交给 harness 时，尤其是面对能力较弱、需要更多引导的模型，这一点很有用。

00:19:12 • Mo Abualruz

我们可以取出相关的图链接，一起交给 harness，让它开始工作。所以这里会用到嵌入。你可以让嵌入在 OpenAI 上运行，也可以使用 Claude 模型。不过，嵌入在本地运行的成本很低，大部分模型只有三四 GB。

00:19:39 • Mo Abualruz

因为这里的目标不是一套 RAG 检索流程，甚至不需要重排序。我们主要是要保证数据经过嵌入、在图里互相关联，并且能够快速取回。LLM 路由器则负责选择用哪个模型。

00:19:54 • Mo Abualruz

我会为路由设置一些规则。例如，做规划、希望在本地工作且有本地模型可用时，尽量用 Qwen 27B；如果只是快速做一轮调研，我不介意用 Gemma 4B。它要做的只是调用我的 web MCP 和［搜索工具名听辨不清］，取回搜索结果，再给我一份调研摘要。

00:20:23 • Mo Abualruz

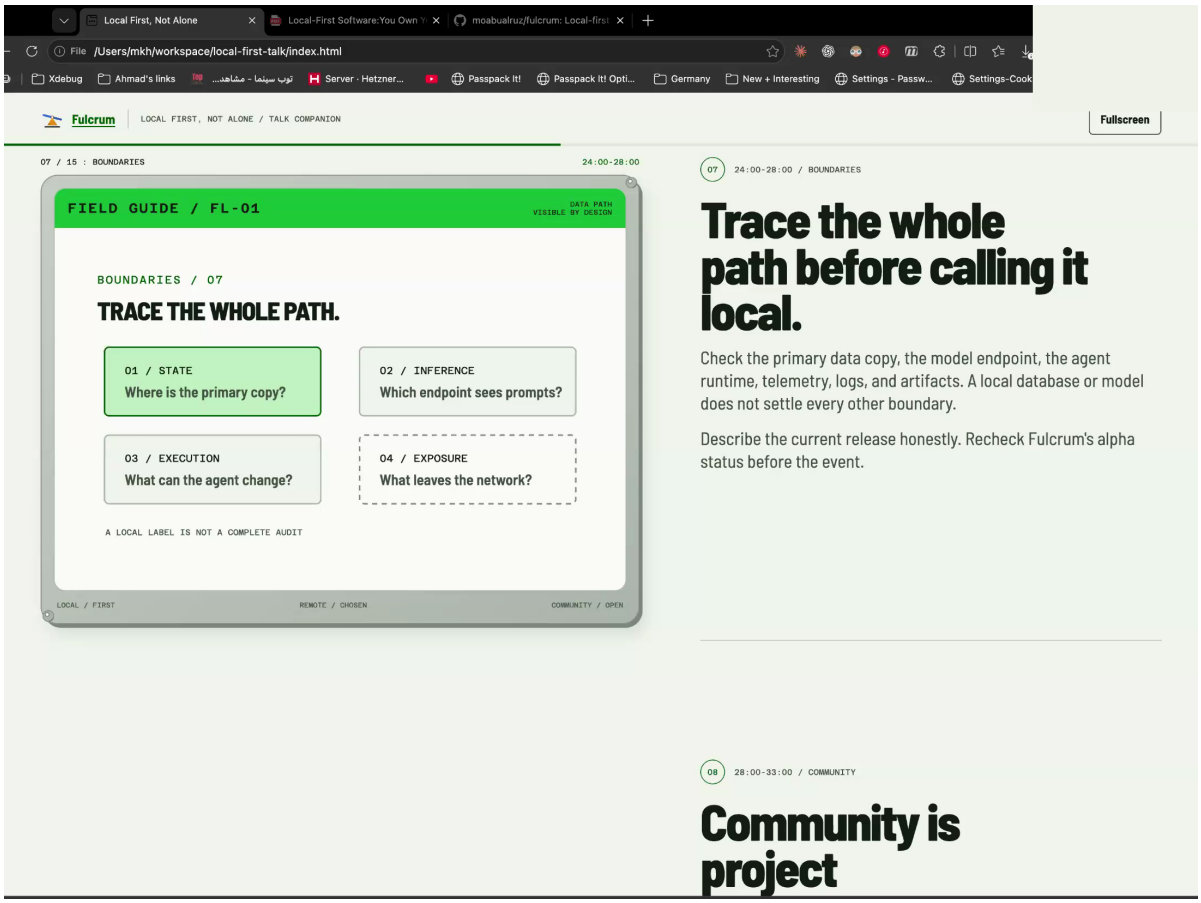
这不需要特别强的智能，只要能找到我在网上查询的东西，并整理成摘要就够了。我会把规则写进配置文件，每个用户都可以这样配置：第一层、第二层、第三层分别用哪些模型，如何运行。它们可以全部在本地，也可以混合使用本地模型和云端推理服务。

00:20:51 • Mo Abualruz

通过这些配置，就可以决定每个请求交给哪里处理。这就是混合系统：LLM 路由器可以连接 LM Studio、Ollama、Claude、Kimi，或其他兼容 OpenAI 接口的推理服务。我另外做了一个工具来适配 Claude，因为 Claude API 与 OpenAI API 不兼容。它封装了 Claude SDK，连接到你的账户，再把 Claude 的端点转换成兼容 OpenAI 的端点。

00:21:39 • Mo Abualruz

这个工具主要是为了我自己方便。我们也可以在 Fulcrum 中原生支持 Anthropic 的消息结构和 Claude 的 API 调用方式。这是可以增加的功能。我记得中国的 MiniMax 好像也支持类似的方式；它与 OpenAI 的推理接口约定不同。



现场演示画面 · 00:23:00

00:22:13 · Mo Abualruz

接下来是刚才提到的可追溯性。无论做什么项目，这都很重要，不只是做 AI 才需要。你需要能够检查和观察工作：现在发生了什么，进度到了哪里？

00:22:39 · Mo Abualruz

对于某些角色，还需要知道机器的运行情况：瓶颈在 CPU、内存，还是 I/O？这些状态都应该能看见。在 Fulcrum 中，我们还要观察推理：调用了多少次？是不是太多了？能否通过优化配置，让它做得更好？

00:23:05 · Mo Abualruz

也许执行方式还可以改进。现在我会让智能体同时参与多个项目。我后来发现，执行过程产生了太多请求、太多 PR，也有很多零散文件。因此，我会要求它把整棵依赖树中的相关工作一起取出来，放在一个 PR 里完成。

00:23:35 · Mo Abualruz

有时候收到一个评审请求，里面只有一两个文件，无法体现完整上下文，也很难判断它的价值。智能体拿到了我们提供的关系图，或许可以看到全貌，但我作为人却不容易理解。所以我会引导它：能不能把有关联的内容一起拿出来？因为图里也记录了 issue 之间的依赖关系。

00:24:02 · Mo Abualruz

能不能把相互依赖的 issue 放在一次评审中一起处理，让这次评审更有意义？如果没有这些可视化路径和视图，看不到、也无法与这些关系交互，就很难发现工作流中值得改进的薄弱点。因此，可追溯的路径和视图对我很重要：它们让我看见整个过程，并优化自己的流程。

00:24:36 · Mo Abualruz

这些行为都可以通过配置和指令调整，没有写死。我用配置文件和 Markdown 文件来引导智能体。如果把步骤硬编码成一层僵硬的流程，也许适合我，却会限制别人根据自身需求使用它。接下来我准备转向社区部分：为什么 Fulcrum 需要社区，为什么像现在这样的知识分享、交流活动很重要，以及为什么用户和贡献者组成的社区，对这类项目和其他开源项目都很重要。在这之前，有 Fulcrum 相关的问题可以先提。

00:25:43 · Mo Abualruz

大家如果有关于 Fulcrum 本身的问题，现在可以问。

00:25:57 · 主持人

谢谢 Mo。大家如果有问题，现在可以提出来。

社区、能动性与共享知识

00:26:10 · 主持人

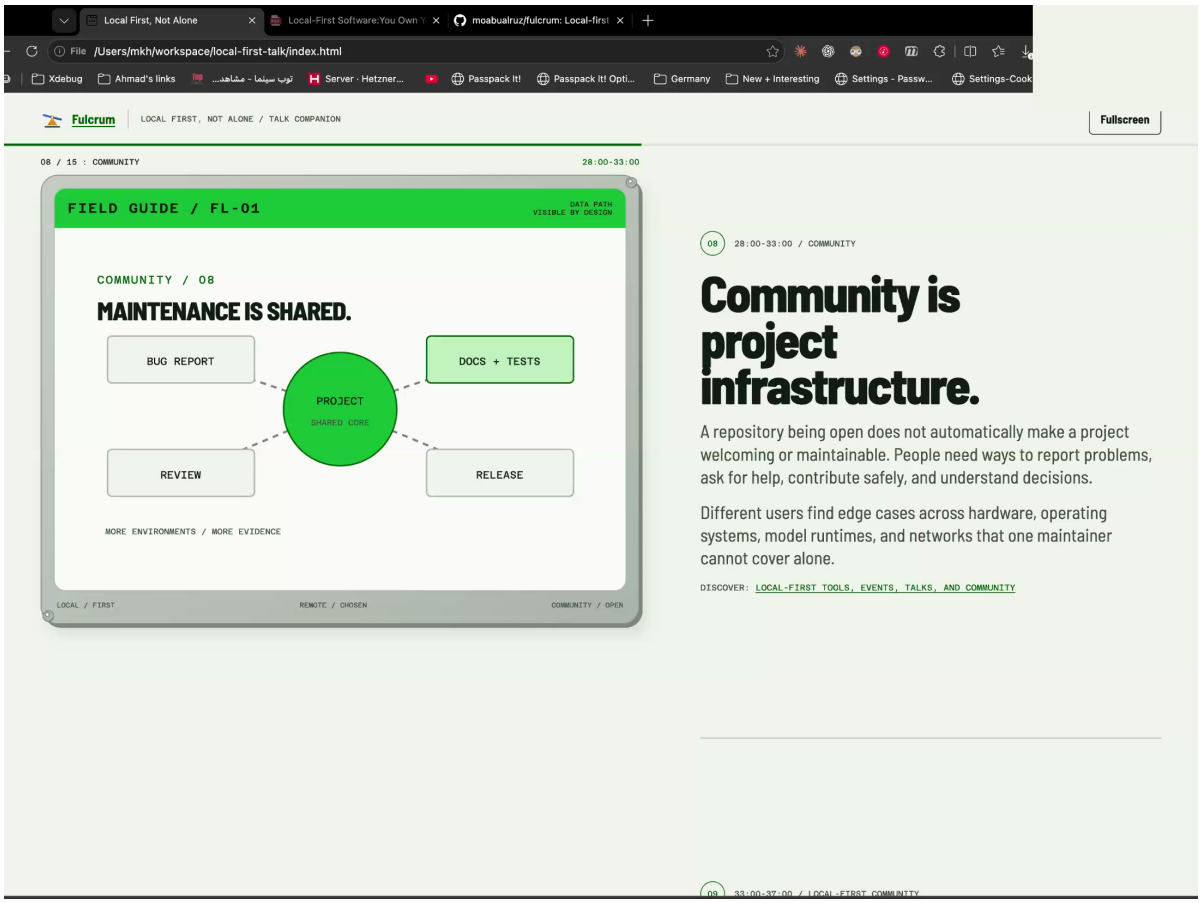
大家如果有问题，可以先提。他刚才讲了 Fulcrum 的基本运作方式，接下来会讲为什么社区对 Fulcrum 很重要，以及项目怎样与社区结合。大家可以先围绕项目本身提问。

00:26:45 · 主持人

好像有人在聊天框里贴了 GitHub 链接。

00:26:53 · Mo Abualruz

没关系，是我把链接分享出来，方便大家看看，然后提问。最后也会留一段问答时间，大家可以先浏览。这里有一点很重要：Fulcrum 是开源的，不代表它就已经百分之百地对社区友好。人们需要参与项目的途径。我觉得自己已经尽力让它更容易参与，但可能仍有需要改进的地方，比如如何提交 issue、如何联系我，以及将来有更多维护者时，如何联系其他维护者。任何开源项目都需要考虑这些问题。



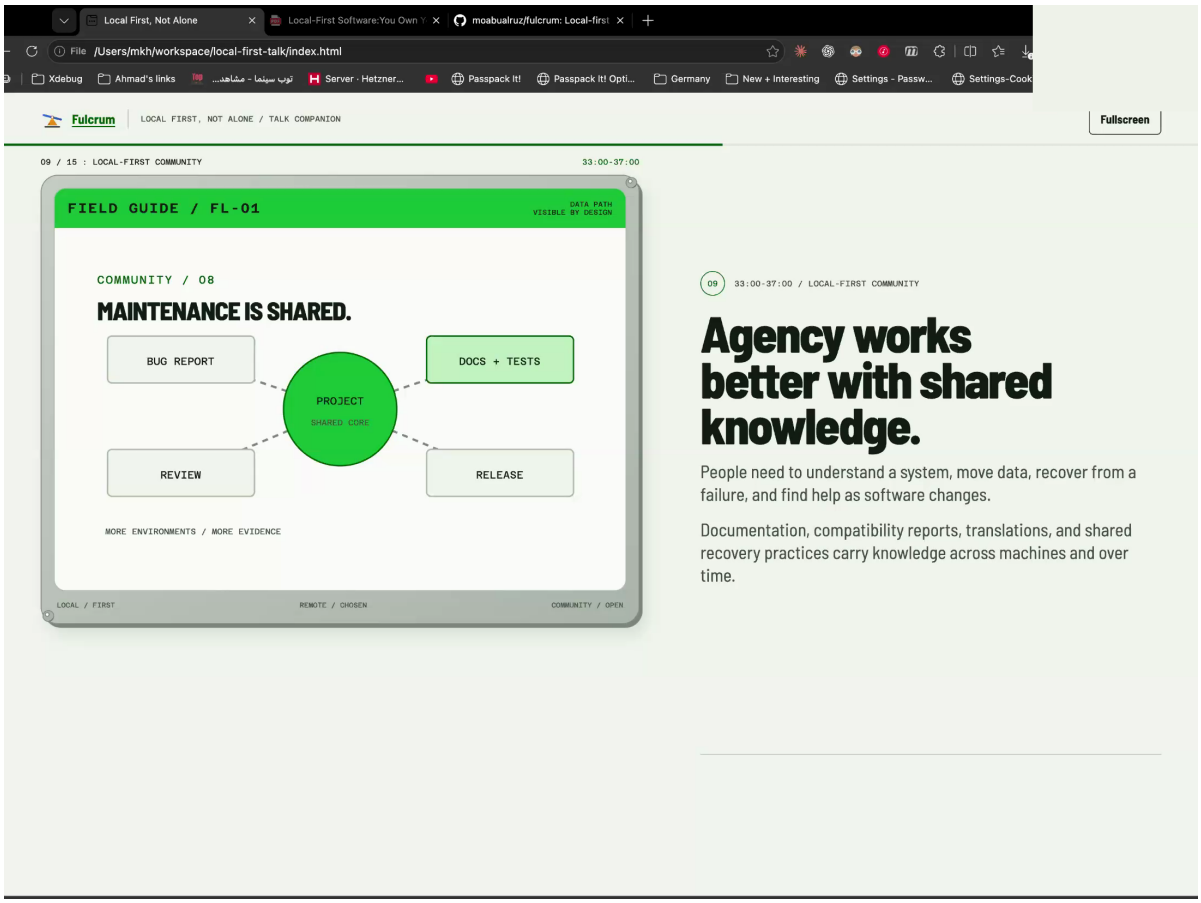
现场演示画面 · 00:28:20

00:27:48 · Mo Abualruz

做自己的项目时，也应该考虑这些方面。如果决定开源，就不只是开发产品，还需要建设社区。社区是另一条需要持续维护的成长路径。它越大，你能获得的反馈就越多，产品和想法也就越有机会被打磨得更好。

00:28:19 · Mo Abualruz

贡献不只是写代码或报告问题，也包括交流想法。像今天这样的活动，我把自己在想的事情讲出来，再听大家提问，听其他人讲自己的项目，就能让所有人受益，分享经验和知识。如果希望 Fulcrum 不只是我个人使用的本地智能体操作系统，社区就非常重要。我必须努力围绕它建立一个社区。即使没能做到，它仍然可以是一个对我有用的工具，但不会获得广泛采用，改进速度也会慢下来，长期积累的能力也会不同。人的能动性在共享知识的基础上能更好地发挥。教会别人一些东西，会让他们拥有更多选择。我们说“知情决策”比单纯做一个决定更好，因为基于信息、知识和学习作出的判断，通常更准确。



现场演示画面 · 00:30:20

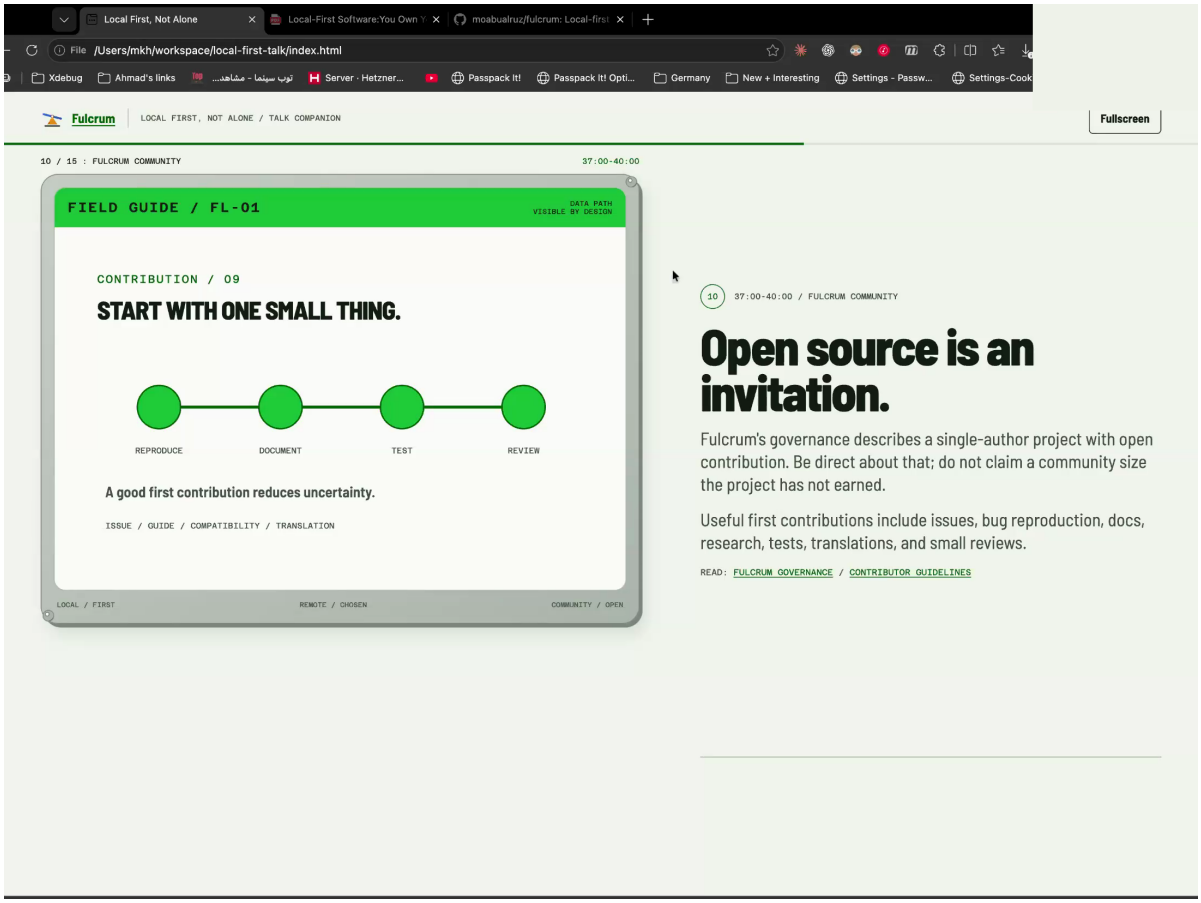
00:29:49 · Mo Abualruz

举个例子：小时候，至少我遇到过，有人拿硬币逗我，说“你想要这两枚，还是这一枚？”小孩子会觉得两枚比一枚好，但我们需要学会把硬币翻过来，看看每一枚的面值。如果没有人教我，我可能就不知道该怎么比较。原来这两枚各值一，而那一枚值五，我应该选那一枚。所以，这些判断是需要学习的。这就是怎样形成知情决策。如果人们没有学习和交流的自由，没有能动性，当然仍然可以作决定，但那与充分了解情况后作出的决定，不在同一个层次。

00:30:45 · Mo Abualruz

系统应该帮助你充分了解情况，作出正确决定，并推动事情继续向前。接下来讲讲贡献。

文档、贡献与本地服务



现场演示画面 · 00:33:00

00:31:10 · Mo Abualruz

前两类贡献主要是技术性的：提供问题复现，以及完善文档。在文档这件事上，我觉得很多工程师做得还不够。

00:31:30 · Mo Abualruz

文档很关键。我知道它很难写、很琐碎，也不是开发过程中最有趣、最吸引人的部分，但它很有用。尤其在 AI 时代，更多的文档不仅能帮助建设社区，让人们充分了解产品、判断怎样改进它，也能帮助智能体快速获取上下文并据此工作。有人说代码本身就是文档，这有道理，但不是每个人都能轻松读懂代码。

00:32:15 · Mo Abualruz

不是所有人都有足够的技术背景。对智能体来说，细节写在代码注释里，还是放在文档网站上，可能差别不大；但如果是刚毕业、技术经验还不够丰富、正在学习的新人，第一天就让他一头扎进代码，会很困难。

00:32:43 · Mo Abualruz

文档能够引导人，也是一项建设社区的基础。我希望 Fulcrum 不仅有技术文档，还包括用户手册等内容。另一个重要方面是测试：你、贡献者以及其他人，怎样验证项目所声称的能力？怎样确认它真的能工作？新增一项功能后，怎样确认它与已有功能仍然配合正常？这些都需要测试。

00:33:25 · Mo Abualruz

社区越大，就越需要你和其他成员一起参与评审。评审也是积累知识的一种方式。刚才说过，你可以请智能体通过评审教你一门语言的用法，但也可以向人学习。和社区中的人一起交流，往往更有参与感。

00:33:52 · Mo Abualruz

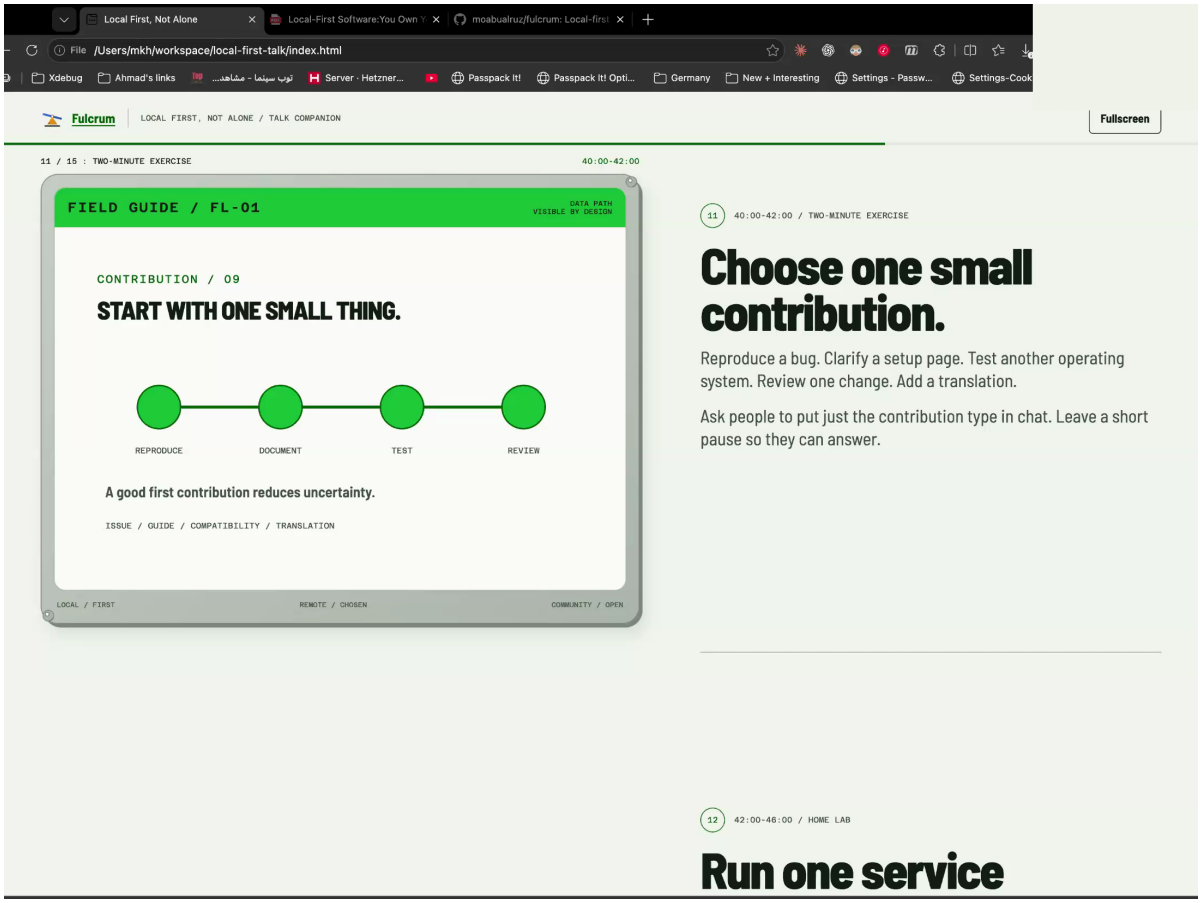
两种体验不太一样。智能体的代码评审可能只有几行，告诉你“这是一种不好的写法，所以要这样改”。如果与一个人交谈，请他解释为什么不好，他可能会带你更深入地理解。当然，这也取决于对方，有些人的沟通能力未必那么强。

00:34:20 · Mo Abualruz

但至少在我认识的大多数人中，只要你问一句“为什么”，让对方知道自己真的想学，通常就能打开话题，获得更多细节。

00:34:36 · Mo Abualruz

我还想鼓励大家：不要因为问题很小，或者觉得不值得一提，就不去贡献。可以开一个 issue。这不只是针对 Fulcrum，而是针对你使用的任何开源项目。如果有某个不好的地方，浪费了你的时间或精力，就值得反馈。



现场演示画面 · 00:35:50

00:35:13 · Mo Abualruz

我们经常遇到功能不太好用，但自己找个小办法绕过去就算了，也不告诉别人哪里有问题。这样，产品就可能沿着错误的方向继续发展。大家觉得不好，却不说、不提交 issue；有时已经在本地修好了，也不把代码贡献回去，或把这个办法作为一种可能的解决方案分享出来。

00:35:42 · Mo Abualruz

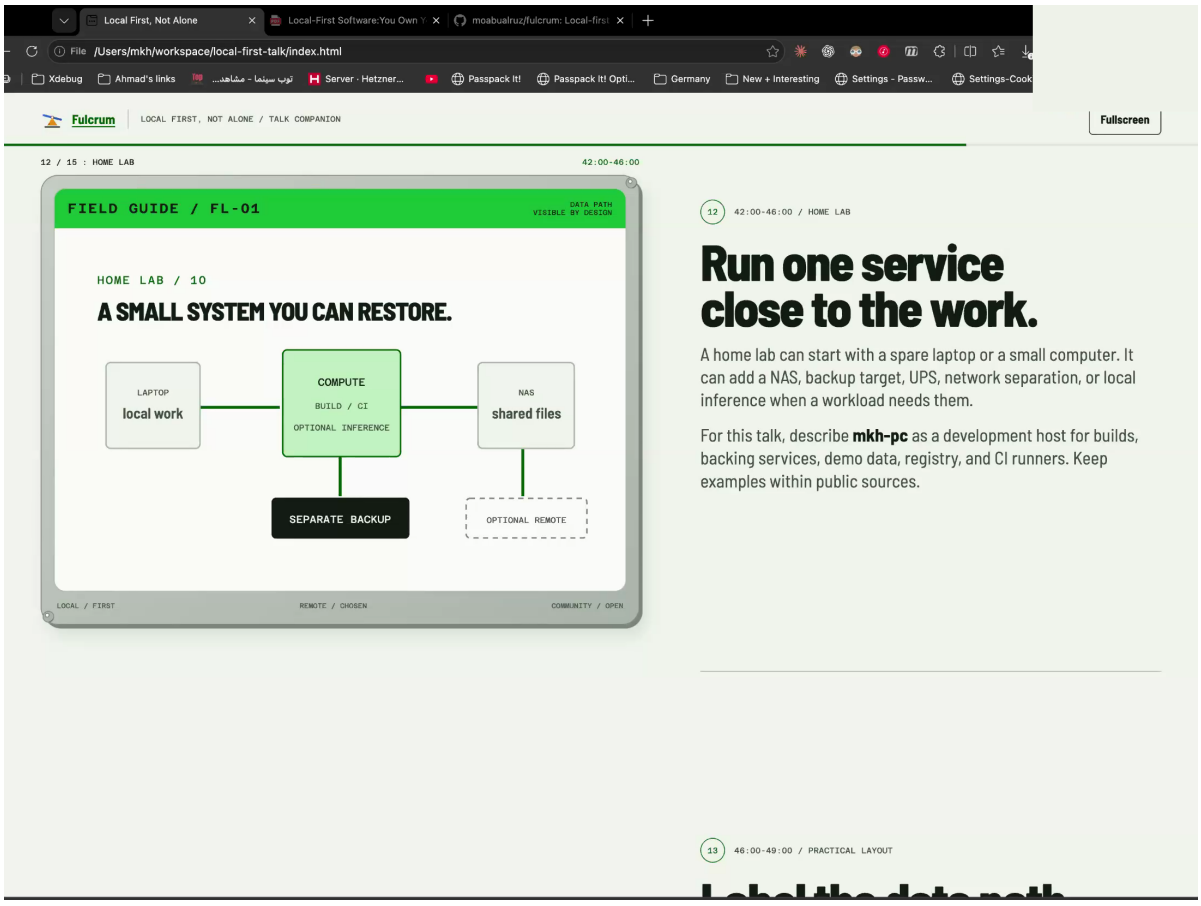
所以，我想鼓励大家：只要你知道哪里不对，能够复现、能够说明，就去贡献、去报告。说清楚“发生了什么”“我需要什么”“我希望怎样改进”。社区会因此成长，你的贡献也会受到重视。

00:36:07 · Mo Abualruz

至少，你能知道某个让你不舒服的地方是不是有意设计的。比如，一项操作很烦，却是为了处理安全问题。因为你问了，才明白：多按一个按钮、多做这一步，是为了保护数据。

00:36:34 · Mo Abualruz

理解之后，你也许还是觉得麻烦，但能够接受。如果从来不问，这个疑惑就会一直留在脑子里。关于开源和社区建设，我先讲到这里。接下来介绍一下自己的部署：除了 Fulcrum，我还运行了哪些支持开发工作的服务。在这之前，大家如果有关于贡献或社区建设的问题，可以先问。



现场演示画面 · 00:39:30

00:37:23 · 主持人

他说，如果大家对参与贡献或社区建设有什么问题，现在可以提。如果没有，他接下来会介绍自己目前的环境和部署安排。

00:37:48 · 主持人

大家还有问题吗？也可以先看看他的 GitHub。

00:38:01 · Mo Abualruz

好。

00:38:02 · 主持人

好。

00:38:03 · Mo Abualruz

如果现在没想到问题，最后还可以再问。这一部分大约还有五分钟，之后我们会有充足的问答时间。重要的是，不只是 Fulcrum，也不只是某一个服务：没有什么东西能单独满足全部需求。你需要把开发环境作为一个整体来考虑。我家里有一台 PC，上面运行了很多服务。一些硬盘用作普通的 NAS，我妻子和孩子用来存照片等文件。我也在上面运行一些 GitHub Actions runner。因为我不想每个月为了运行 Actions 向 GitHub 支付一千美元。它本质上就是一组需要执行的步骤；既然我自己有机器，就可以把任务交给它。顺便说，这件事不难：创建一个 GitHub App，配置相应的 IP 或 API，就可以把 runner 的请求放入队列。

00:39:17 · Mo Abualruz

然后你可以在自己选择的机器上运行它。做一个这样的项目，可能一天就够，也已经有开源项目帮助管理这些事情。如果没有公网 IP，我想可以使用 NetBird，或类似的个人 VPN mesh 产品，让 GitHub 能访问本地机器并将 Actions 加入队列。

00:39:43 · Mo Abualruz

如果这台机器上有非常重要的数据，还需要考虑 UPS，也就是不间断电源。停电时，要让它继续运行一个小时，需要多少电池？我不太了解中国的情况，想来供电比较稳定；但至少在我所在的中东地区，停电是常见的事。

00:40:11 · Mo Abualruz

如果这台机器对工作至关重要，就要把这些事情考虑进去。按照刚才说的本地优先思路，我会确保笔记本电脑上有全部数据，而且笔记本能靠电池运行很久。但有些服务它承担不了，例如我的 GitHub Actions 需要那台 PC 来跑。工作进行到某个阶段，仍然离不开它。这时就需要 UPS，至少让关键部分保持运行。

00:40:43 · Mo Abualruz

网络怎么设计，需要多少上传和下载带宽，存储容量多大，都要衡量价值与成本。我为什么愿意购买比较贵的硬盘，保存妻子和孩子的照片？因为他们喜欢拍照，我们给苹果和谷歌支付的额外存储费用，一度达到每月大约两百美元。

00:41:19 · Mo Abualruz

另一种选择是开始清理旧照片，逐张看哪些要删、哪些不要。可是，对妻子和孩子来说，那些照片和记忆的价值…… [此处转入观众提问，原句未说完。]

问答：模型、上下文与协作

00:41:37 · 观众（聊天提问）

谢谢分享！不好意思，我可能来晚了。这与让一个智能体运行本地大模型有什么不同？例如，用 DeepSeek harness 运行 Qwen？我需要说服我们的董事会转用它。谢谢！

00:41:37 · Mo Abualruz

抱歉，看到一个问题。

00:41:52 · Mo Abualruz

你可以在 Fulcrum 中使用不同的 harness。我想这个问题是在问 Fulcrum。它也可以直接调用 Ollama 等服务进行推理，但 Fulcrum 本身不是一个 harness。它是一个本地优先的集成层，把你的数据、联系和上下文组织在一起。

00:42:19 · Mo Abualruz

你可以把记忆保存在 Fulcrum 的文件里，也可以把 Honcho 或 Mem0 接成记忆后端。重点是，不管使用哪一种 harness、采用哪一种工作流，相关内容之间都有上下文关联，你能够推进工作，也能够观察整个过程。

00:42:44 · Mo Abualruz

你可以看见自己正在执行哪些步骤。更新计划或任务时，不必在 Fulcrum 和另外两个产品之间来回切换。它提供一个能完成大部分工作的地方，让你查看、规划和执行工作。

00:43:11 · Mo Abualruz

我有没有直接集成 DeepSeek harness？它比较新，我还没有把它接入 Fulcrum，也没有测试过两者一起运行，只单独试用过它。我已经为 Pi 做过集成，而 DeepSeek harness 与 Pi 有一些相似之处，所以我想，增加一个小插件，应该就比较容易接进来。

00:43:37 · Mo Abualruz

［讲者开始读下一条关于追踪与审计的问题。］

00:43:55 · 观众（聊天提问）

随着大模型版本更新、项目持续推进，Fulcrum 是否为记忆、上下文和技能提供追踪与审计功能？

00:43:55 · Mo Abualruz

追踪或者审计功能。如果你有执行审计的 skill，当然可以使用。不过，如果你问的是增加一个专门的审计步骤——我现在是把审计放在评审里面。抱歉。你刚才说了什么吗？

00:44:19 · Mo Abualruz

好，在我的工作流中，审计属于评审的一部分。但像刚才说的，如果没有社区反馈，我可能就不知道，是否有人需要在 Fulcrum 的工作流里设置一个独立的审计步骤。你使用之后，可以提交一个任务或 issue，让我来做；如果技术经验足够，也可以直接提交一个增加这项功能的 PR，说明自己已经试过。比如："它运行正常，我觉得把审计放在这个位置很合适。"所以，目前还没有独立的审计步骤。不过，我有 doctor 脚本，可以检查技能和安装状态。Fulcrum 中这些诊断脚本，主要检查 Fulcrum 自身的安装，以及与 harness 的集成情况，并不是专门审计记忆和上下文内容。

00:45:14 · Mo Abualruz

但这可能是一个好建议。你可以提交 issue，我会看；如果你有思路，也愿意自己做，就可以直接动手。

00:45:28 · Mo Abualruz

这两种方式我都欢迎。

00:45:32 · 观众（聊天提问）

同时运行很多智能体时，共享上下文，以及维护这些上下文，听起来有些棘手。

00:45:32 · Mo Abualruz

对，它可以运行多个智能体。你可以在设置里指定同时运行多少个。默认情况下，一个智能体处理一个任务；但也可以通过指令，引导它把多个任务放在同一个上下文中，作为一个整体处理。

00:45:56 · Mo Abualruz

关于共享上下文，你不需要花太多精力手动维护上下文窗口。你不能在这里直接改动关系图，但如果你认为构图、嵌入，以及计划、决策和 issue 之间的关联方式存在浪费，数据太多、可以减少，那么这些可以通过代码和配置来调整。

00:46:23 · Mo Abualruz

如果现有配置还不够，那就是一个应该解决的问题。尤其使用小模型时，我会尽量缩小上下文，提供文件链接，而不是把整个文件塞进去。某些文件非常大，只有 32K 上下文窗口的小模型根本装不下。所以这里会进行一些管理。以后也可以把压缩程度、何时将正文替换成文件链接等做成配置。但目前的设计，基本上假定大多数人的上下文容量并不充裕。

00:47:18 · Mo Abualruz

这就是 Fulcrum 现在的设计方式。

00:47:32 · Mo Abualruz

好。

00:47:34 · 观众（聊天提问）

你为这个本地优先项目投入了很多工作。动力从哪里来？是什么让你坚持下去？什么个人经历促使你开始做 Fulcrum？

00:47:34 · Mo Abualruz

因为它满足了我自己的需要。最好的解决方案，往往来自你真正遇到的问题：先把自己的问题解决，再想怎样把方案做得更通用。如果别人也遇到类似问题，我们能怎样帮助他们？阿拉伯语里有一句话，al-hajjah umm al-ikhtiraa，意思是“需要是发明之母”。发现自己的需要之后，不要止步于“这两行代码解决了我的问题”，而是再多想一层：怎样让它更稳健，既适合我，也适合别人？等它逐渐成长，再考虑能不能从当前项目中拆出来，成为一个独立的工具，帮助更多人。

00:48:51 · Mo Abualruz

一旦看到自己做的东西真的在运行、确实有帮助，动力就会形成一种反馈循环。工作变得更容易、更快、更愉快，token 用量也减少了。例如，原来每个月给 OpenAI 和 Claude 的推理费用一共是一千美元，现在这里一百、那里一百，就省下了八百美元。看到这些结果，我就会觉得“这很不错”，至少对我如此。

00:49:24 · Mo Abualruz

它也许不是对所有人都最好的工具，但自己的工作方式得到改善，能从实际使用中获得反馈，对我很重要。我认为做任何项目都应该这样：先看自己需要什么，再看做出的东西产生了什么价值，这些价值怎样回到自己身上。最后，再想怎样把它做得更通用，让更多人的生活受益，也许能给世界留下一点影响。别人可能不知道你的名字，但知道这个项目帮到了自己，改善了生活。

00:50:11 · Mo Abualruz

这一直是我的出发点。

太阳能、本地计算与完整成本

00:50:21 · 观众（聊天提问）

两个问题：未来，你是否设想过，不同用户的智能体之间也能进行规划、编排或协作，而不只是一个用户的智能体？另外，如果想搭建太阳能系统来运行本地 AI，你觉得怎样做比较好？

00:50:21 · Mo Abualruz

第一个问题是，不同用户的智能体之间能否进行规划和编排。

00:50:33 · Mo Abualruz

这是一个重要的上下文问题，我认为更接近上下文工程。我们现在仍然倾向于把智能体当作人来理解和交流，但智能体不是人。我把它们看作生产力工具，或者速度很快的打字机。即使现在已经很聪明，它们仍然是非常强的自动补全工具：以前补全一行，现在补全整个文件、页面甚至项目。

00:51:12 · Mo Abualruz

应该更多地把智能体和 AI 当作工具，而不是人。你可以让很多人协作，但如果是我和主持人合作，我们仍然需要花时间彼此沟通。假设我们使用同一个 Fulcrum 实例，各自做了规划；之后开一次会，对齐思路，再请一个智能体检查两份计划中是否有冲突，合并成一份总体计划，并分别获得我们的同意。它可能五分钟就能完成这部分工作。

00:51:59 · Mo Abualruz

但如果我们从来没有彼此沟通，就不会有这样的工作流。人与人之间的交流仍然需要存在，我们仍然需要人、需要交谈。变化发生在沟通之后：我们不必再开五个小时的会，一项项写下阻碍和冲突，可以先让 AI 检查所有材料，生成一份报告。

00:52:28 · Mo Abualruz

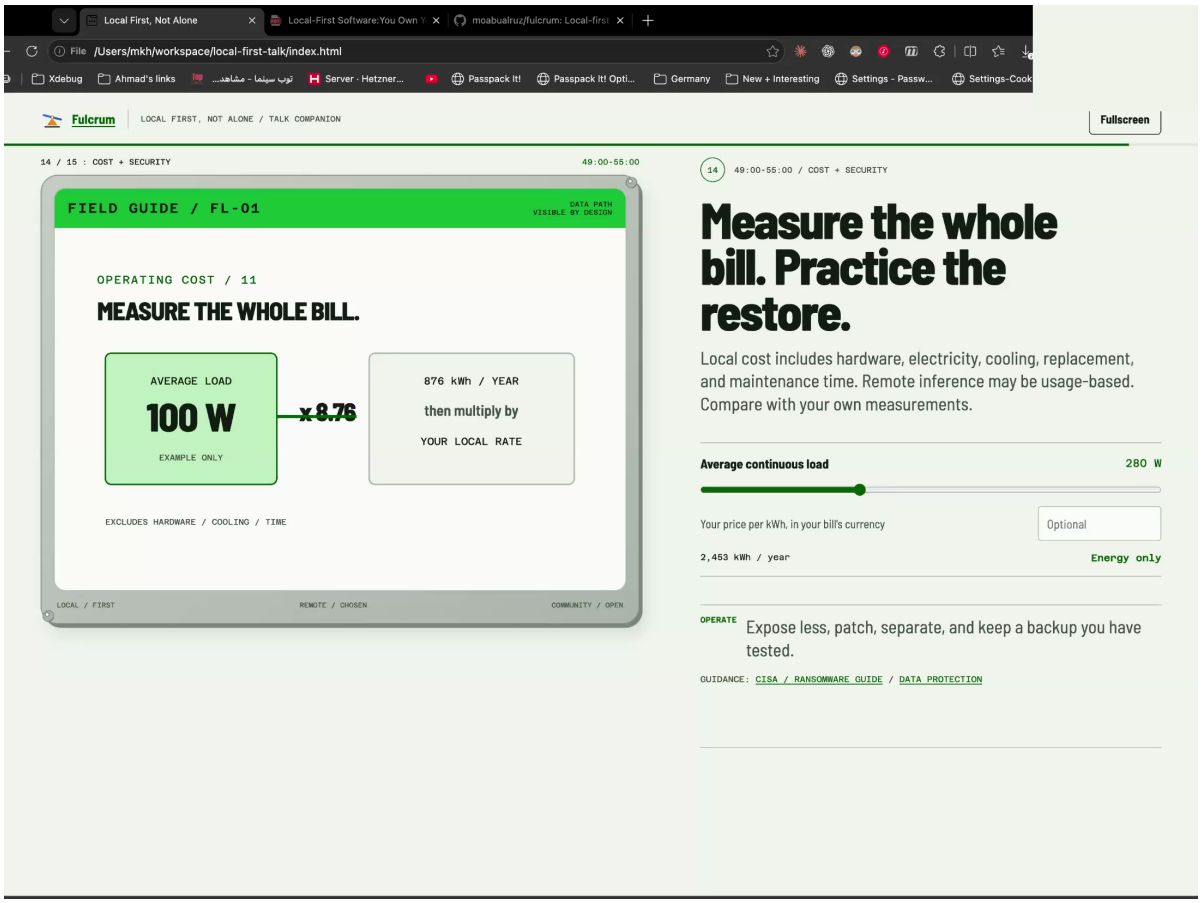
我们花五分钟看一看，确定这几项、那几项，再把结果交给 AI，让它合并为总体计划，并按配置运行相应数量的智能体。这种思路不局限于 Fulcrum，也不局限于某个 harness，任何运行框架都可以采用。把人当作人，把 AI 当作工具。我认为 AI 是今年最好的工具，也是日历发明以来最好的生产力工具：日历让人们安排时间、开展协作，而 AI 是在那之后出现的下一个最好的东西。

00:53:14 · Mo Abualruz

它的价值很大。我不太认同那种“AI 末日论”，觉得缺乏逻辑。人们对矩阵乘法如此恐惧，在我看来有些过头；我觉得不了解数学的人，才会害怕数学。基本上，AI 是一种数学工具，很有用，能帮助你提高生产力。

00:53:40 · Mo Abualruz

它有时非常聪明，有时又非常笨，所以要检查它，确保它按照你希望的方式完成你希望的工作。这样你就能提高效率，也能获得那种“我做了一件对自己和别人有价值的事”的正向反馈。



现场演示画面 · 00:55:00

00:54:05 · Mo Abualruz

谈到本地托管，还必须考虑电力、散热、设备更换和维护。我快速讲一下，时间差不多了，想多留一些时间给大家提问。无论本地优先、云服务还是自托管，都要检查自己的支出，以及从中得到的价值。

00:54:35 · Mo Abualruz

如果目的是学习，做完实验后就可以把环境拆掉，不必因为从中学过东西，就永远让它运行。实验有成本，也有收益。比如，你想运行一个大小为 50 GB 的模型，不一定就要买一张 RTX 6000。

00:55:01 · Mo Abualruz

如果朋友有，可以问问能不能借用两小时，做个实验、学习一下。如果每天都要用，再考虑怎样把它纳入自己的环境。花一万五千美元买显卡，能不能带来相当价值的知识、经验和使用收益？不要仅仅因为某样东西新、看起来好，就决定去做。

00:55:29 · Mo Abualruz

要从现有条件中获得尽可能好的价值。如果云端能让你以最合适的成本实验、学习、改进和做项目，那就在云端做。只是要确保文件和知识都有本地备份，这样云服务出了问题，你也不会被卡住。这里举个快速的成本例子：假设我本地运行的机器消耗 500 瓦电力，电价按 5 来算，那么一年光电费就要付 21,000。这是一个美国价格的例子。［口述中的电价单位和计算结果不清楚，此处保留原述数字。］

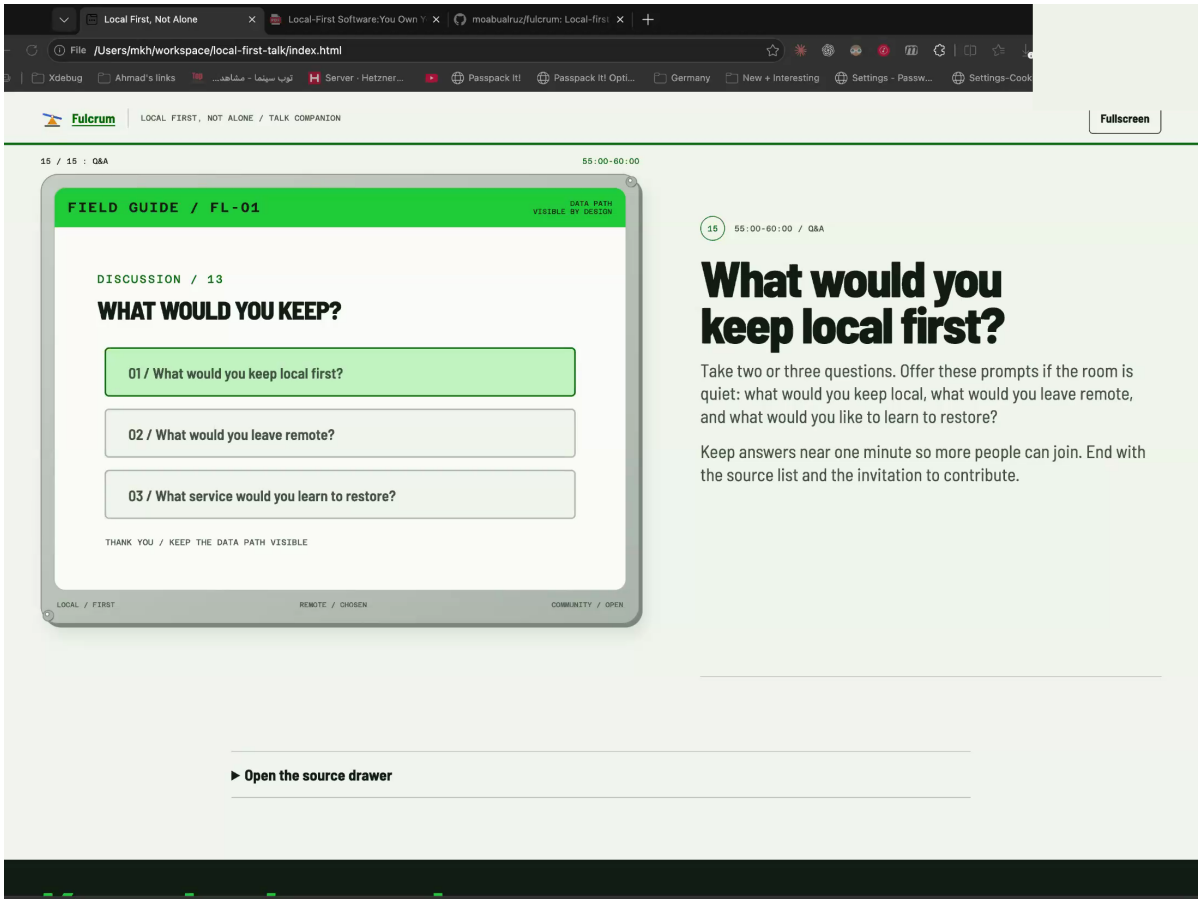
00:56:17 · Mo Abualruz

这个成本就不划算了。用 21,000，我可以使用很多云服务，完成大量工作。然后配一台更小、也许只有 20 瓦的机器，把文件和数据整理好，保留在本地。这样，任何一个云服务商出了问题，我仍然有保障。所以，要平衡本地优先、自托管和云服务分别带来的价值。

00:56:48 · Mo Abualruz

钱不是从树上长出来的。我们需要确认，花出去的钱确实换回了相应的价值。

现场交流与 Fulcrum 演示



现场演示画面 · 00:58:30

00:57:00 · Mo Abualruz

最后，可以经常问自己三个问题：哪些东西留在本地？哪些东西放在云端？备份和恢复怎么做？这就是我建议的思考方式。

00:57:26 · Mo Abualruz

抱歉，稍微超时了几分钟。如果还有问题，我可以再留十分钟交流。

00:57:39 · 主持人

大家如果有问题，还可以有十分钟时间提问和讨论。

00:58:03 · 主持人

我有一个问题：在整个 Fulcrum 项目中，你觉得目前最困难的部分是什么？你正在努力解决的主要挑战是什么？

00:58:29 · Mo Abualruz

我觉得 Fulcrum 的社区建设进展有些慢。不过，这也是个比较新的项目，我想大约才八个月，确实不能抱怨太多。但这件事很棘手。即使项目不开源，让它进入市场、做推广，也不是一件容易的事。

00:58:57 · Mo Abualruz

技术人员也需要具备这些技能，不一定要成为最好的营销人员，但至少有两件事应该会做：第一，介绍自己，讲清楚自己的经历、成绩和技能；第二，介绍正在做的产品，说明它的价值。这并不是一套只有专业人士才能理解的知识。

00:59:29 · Mo Abualruz

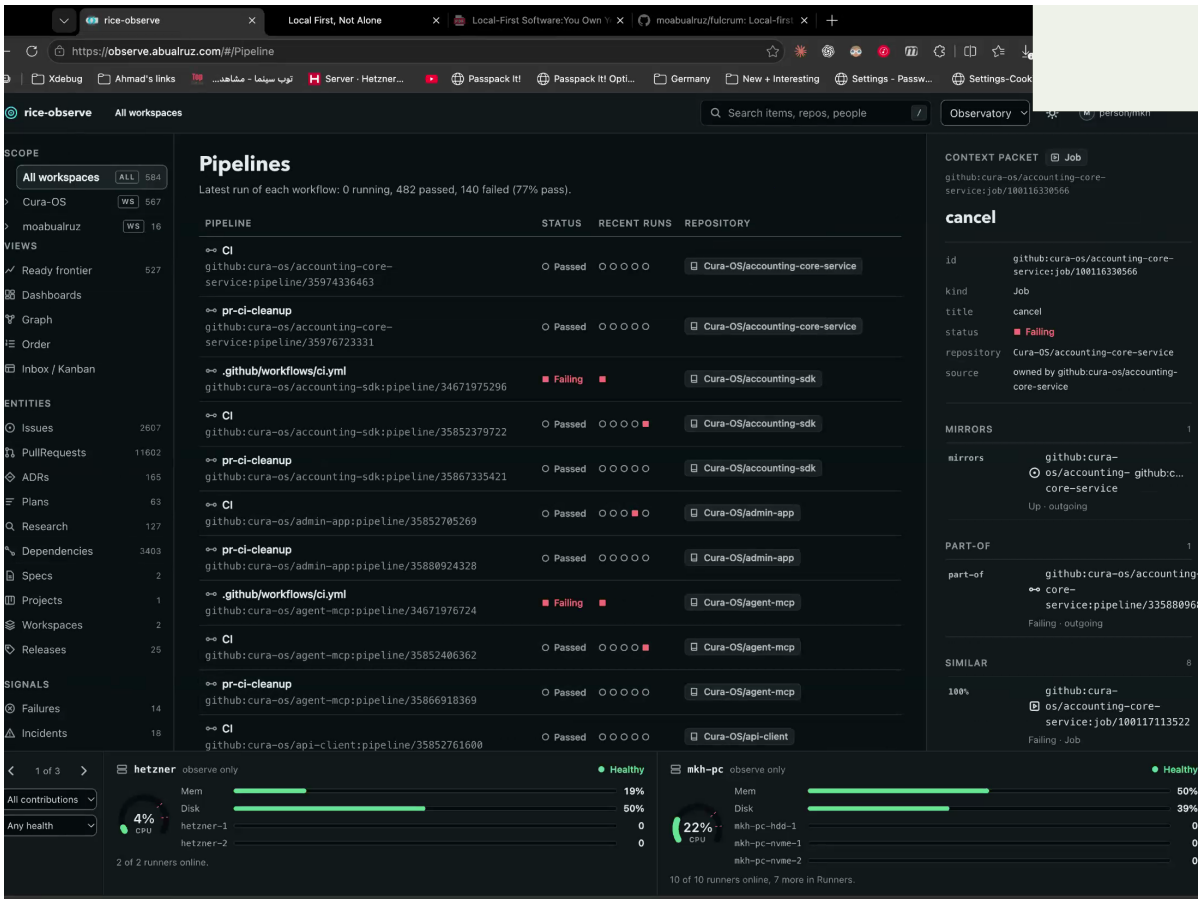
可惜我并不擅长营销，但我知道自己应该在这方面下功夫。我能够很好地介绍自己，有时却不会介绍自己的项目。我发布 Fulcrum 时，特意使用了 alpha 这个词：虽然项目已经做出来了——但没有社区，就不能说它已经经过充分的实际检验，也不能说它最适合所有人。所以，在建立社区之前，我不会宣称它已经是 beta 或最终产品。另外，我最近也在搬家：六个月前离开德国，回到了约旦。发布 Fulcrum 后，我没有花太多力气推广，只是在 X 账户和个人网站上发了消息，就继续做其他事情了。我一直在用它，却没有专注于营销和社区建设。因此，这次分享我不只想介绍自己的项目，也想谈谈大家的项目，以及我自己在管理 Fulcrum 时做得不好的地方。

01:00:49 · Mo Abualruz

从技术上说，它可以用于——日常场景，对，不只是开发。但如果没有社区，我很难独自想全其他用途和案例。有人带来一个具体场景，说“我这样使用 Fulcrum，但它在这个环节失败了”，我们才有机会把这一部分改好，适配更多情况。说到可观测性，我正在开发另一个工具，希望让它成为 Fulcrum 的一部分，或者一个配套项目。

01:01:34 · Mo Abualruz

我看看放在哪里了。对，就是这个。我需要先登录。最近我在给本地设备做安全升级，现在连自己托管的服务也启用了 OTP，要输入一次性验证码。我觉得对本地托管服务来说，这可能有点过头，但谁知道呢，毕竟无法确定谁会接触到你的数据。好，就是这个，我把它打开。



现场演示画面 · 01:02:50

01:02:29 · Mo Abualruz

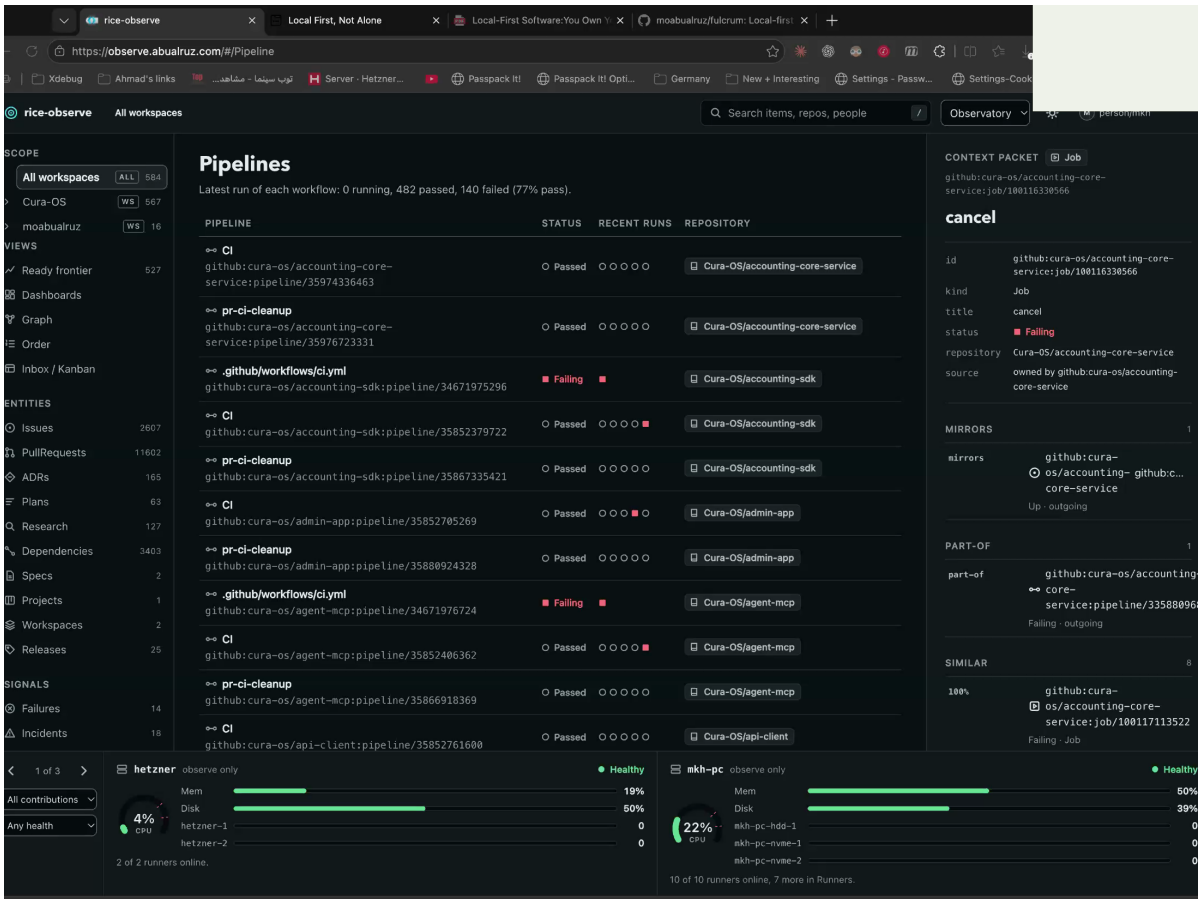
这是 rice-observe。我想把它做成另一个平台，用来观察任务和流水线。还想在 Observe 与 Fulcrum 之间做一个集成插件，让你可以在同一个地方看到各项工作正在发生什么、运行得怎么样。

01:02:57 · 主持人

［提问听辨不完整；随后讲者回应项目是否已公开发布。］

01:02:59 · Mo Abualruz

还没有，它仍在开发，尚未准备好，所以没有发布，代码库还是私有的。我的做法是先持续开发，等它不再频繁出错，再把私有仓库切换成公开仓库。目前仍有很多问题，我还在修正配置。



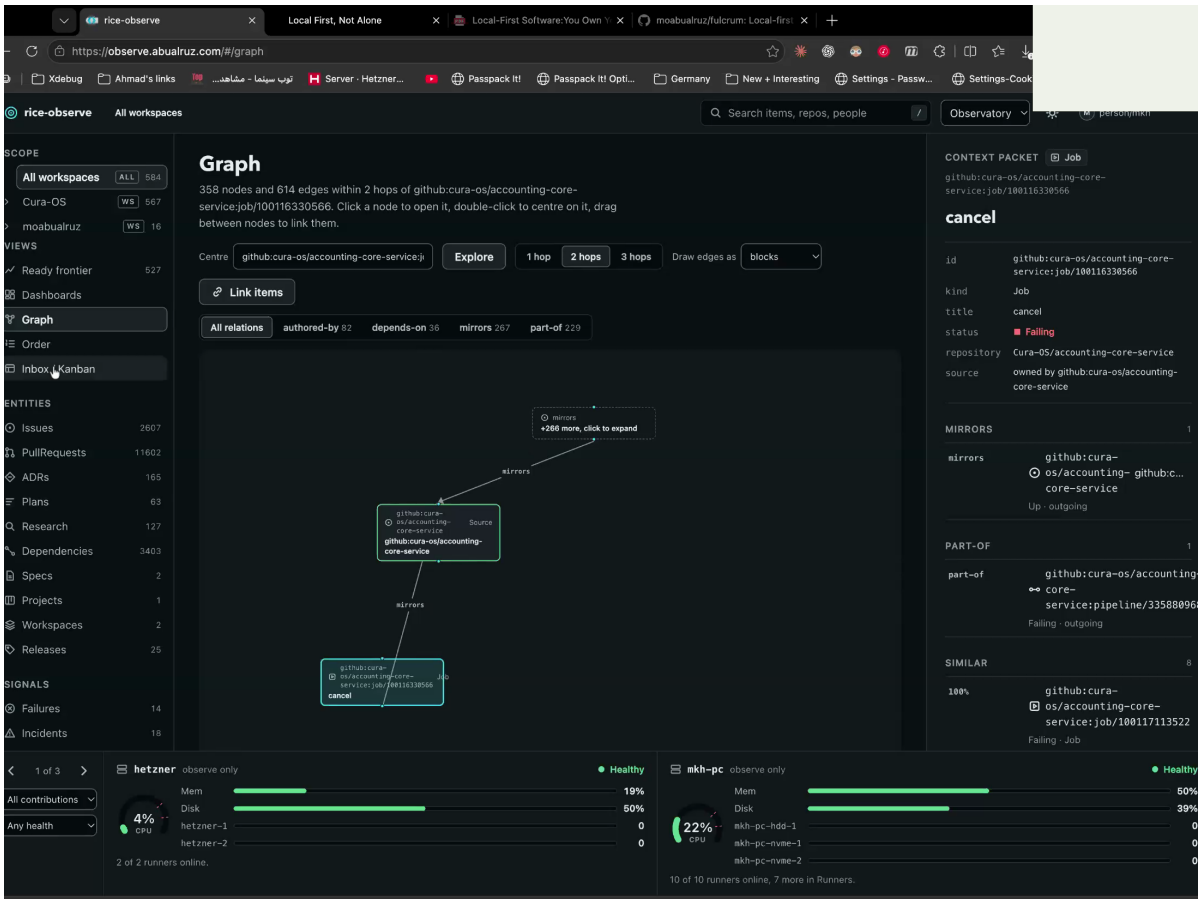
现场演示画面 · 01:04:00

01:03:32 · Mo Abualruz

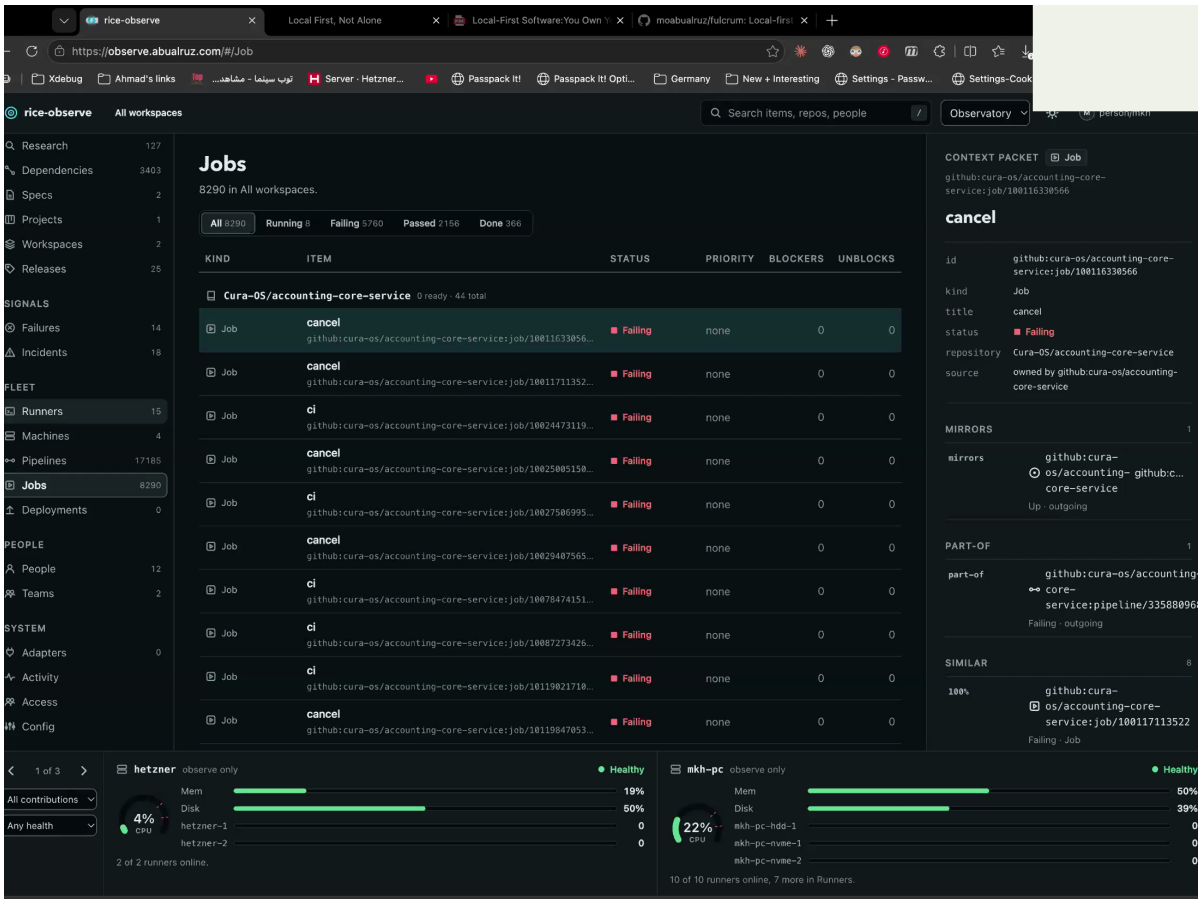
例如，runner 会卡住，队列会堆满，任务结束后不清理文件。我正在解决这些问题。等情况比较稳定，就会把它作为独立项目发布，但也希望它能集成进 Fulcrum。我既想针对 ADR、计划、runner 等内容提供专门视图，也希望有一个统一的地方把这些信息连起来。

01:04:02 · Mo Abualruz

这样就不必在多个地方来回跳转。我可以点击一条链接，展开查看："这是哪个任务？具体细节是什么？为什么失败？"我希望在总览和细节之间灵活切换：平时能快速看到高层状态，需要时又能深入到很细的层次，而且最好不超过两次点击。这符合我理解和观察事情的方式。rice-observe 准备好后，我可能会在 X 上宣布，也会把链接发给主持人。它会纳管我的 Mac、笔记本、PC，以及一台托管其他服务的 Hetzner 节点，并展示每台机器的健康状态。这里还能看见 Fulcrum 建立的同一份关系图，例如我正在查看的这个 issue，以及项目中运行的 PR。现在它还有些慢，需要加速，也有很多问题没有解决，所以仓库仍然是私有的。这里能查看任务细节：失败、失败、失败——正如大家看到的，还需要修复很多地方。这也是我还没公开它的原因。至于为什么把它做成独立项目——



现场演示画面・01:05:30



现场演示画面 · 01:06:30

01:06:11 · Mo Abualruz

因为有些人不需要 Fulcrum 的全部能力，也没有大量智能体编程工作，但有机器需要管理、有 PR 和 DevOps 工作要处理，Observe 对他们可能更有帮助。如果两者都需要，也可以一起用。等两个项目各自做好之后，我还需要开发插件，让它们协同工作。另一个好处是，它能减少触及 GitHub 查询限额的情况。和智能体一起工作时，频繁查询各种信息，可能会收到提示：“API 限额用完了，请等两个小时。”我不想因此停下，所以把数据同步到本地，不必不停向 GitHub 发请求。即使达到限额，我仍然知道哪些 PR 已经打开、自己正在处理哪些 issue。

01:07:23 · 主持人

对，确实如此。

01:07:26 · Mo Abualruz

以后如果还有这样的活动，我很愿意再参加。今天很愉快，希望大家也有收获。

01:07:37 · 主持人

当然，非常感谢。

01:07:42 · Mo Abualruz

谢谢大家。我得先走了。

01:07:45 · 主持人

好的。

01:07:46 · Mo Abualruz

可惜其他几天的活动在工作日，时间对我来说有些早，正好在工作时间里。如果能抽出时间，我会作为听众参加，因为我很喜欢学习，也想了解别人的经验。谢谢大家今天参加。

01:08:08 · 主持人

谢谢，我们保持联系。

01:08:12 · Mo Abualruz

谢谢，再见。

01:08:13 · 主持人

再见，谢谢大家。