Spin Developers Community Meeting Agenda and Minutes

The Spin Developers Community Meeting takes place every 4th Monday of the month at 11 a.m. Eastern Time (5 p.m. Berlin Time) //

Zoom link:
https://us05web.zoom.us/j/86869644377?pwd=d3l1S0ZQWDRBcWlHLzNRejVHRW44UT09

Recordings of previous meetings:
https://www.youtube.com/playlist?list=PLhDWvGTL1Z3kIghOCoAeaHzeUGXa6gH8h

2024-05-20 15:00 UTC

Agenda

- *Assignments for this meeting*
  - Moderator: Michelle
  - Notes: Karen
- *Demo*
  - [Matt] Example apps using JS Frameworks enabled by the new JS SDK
- *Announcements*
  - [Karen] Upcoming events
    - Kubernetes Community Days (KCD) New York '24 (May 22 // New York, NY, US)
      - Leveling Up Wasm Support in Kubernetes (Matt Butcher)
    - AarhusJS Meet-Up (June 4 // Aarhus, Denmark)
      - Server-side WebAssembly with Spin (Mikkel Mørk Hegnhøj)
    - NDC Oslo '24 (June 12 -13 // Oslo, Norway)
      - Lightning Talk: An introduction to the WebAssembly component model (Mikkel Mørk Hegnhøj)
      - Building cloud-native applications using WebAssembly and Containers (Mikkel Mørk Hegnhøj)
    - Cloud Native and Kubernetes Oslo Meet-Up (June 10 // Oslo, Norway)
      - Details TBD
    - DevOps Days Amsterdam '24 (June 19 // Amsterdam, the Netherlans)
      - Workshop: Build a Magic 8-Ball App with Spin and WebAssembly (Sohan Maheshwar)
  - A Security Advisory has been published along with a 2.4.3 patch release
- *Call to Actions*
- *Open floor for community questions/discussion topic*

- *Community Intros*

2024-04-22 15:00 UTC

Agenda

- *Assignments for this meeting*
  - Moderator: Melissa
  - Notes: Karen
- *Demo*
  - [Ryan] Spin-test
    - New alpha level quality piece of software
    - Way to run tests against Spin (everything is virtualized/emulated – nothing goes out to the outside word)
    - Almost entirety of WASI hasn't yet been implemented
- *Announcements*
  - [Karen] Upcoming events
    - Conf42 Golang 2024 (Apr 25 // online)
      - Go-ing Serverless with WebAssembly (Thorsten Hans)
    - KubeHuddle (May 7 // Toronto, Canada)
      - Phippy's Field Guide to Wasm (Matt Butcher & Karen Chu)
    - DevOps Days Seattle (May 15 // Seattle, WA, US)
      - A Greener, Cost-Effective Cloud with Serverless WebAssembly (Kate Goldenring)
    - Kubernetes Community Days (KCD) New York (May 22 // New York, NY, US)
      - Leveling Up Wasm Support in Kubernetes (Matt Butcher)
- *Call to Actions*
  - SIPs - Feedback welcome
    - Spin Testing
    - Guest application access to Metadata
    - Configuring and Emitting Observability
- *Open floor for community questions/discussion topic*
  - Next community meeting will tentatively be pulled up a week to May 20 – May 27th is Memorial Day in the US
  - Spin-info (Radu Matei) – needs help with making output much nicer (now just a wall of text)
  - Vamshi Reddy – university student; contributor to Spin
- *Community Intros*

2024-03-25 15:00 UTC *Note time due to daylight savings

Agenda

- *Assignments for this meeting*
  - Moderator: Melissa
  - Notes: Melissa
- *Demo*
  - [Michelle] SpinKube
- *Announcements*
  - [Melissa] Upcoming events
    - Mar 26: O'Reilly Live Webinar: Building Cloud-Native Applications with Serverless WebAssembly (Sohan Maheshwar)
    - Apr 17: Open Source Summit NA '24
      - A Greener, Cost-Effective Cloud with Serverless WebAssembly - Kate Goldenring, Fermyon Technologies, Inc (Kate Goldenring)
  - [Melissa] The MQTT Trigger is now available!
- *Call to Actions*
  - MQTT Trigger - try it out!
  - Find out more about SpinKube in the docs and Slack channel
- *Open floor for community questions/discussion topic*
- *Community Intros*

2024-02-26 16:00 UTC

Agenda

- *Assignments for this meeting*
  - Moderator: Melissa
  - Notes: Karen
- *Demo*
  - [Joel] Using isyswasfa to experiment with composable concurrency for the Wasm Component Model
    - Experimental polyfill for this async dev experience
- *Announcements*
  - [Karen] Upcoming events
    - Conf42 Python 2024 - Online // Feb 29 @ 5 PM GMT
      - Build a Serverless WebAssembly App in Python using Spin (Sohan Maheshwar)
    - DEVWorld '24 // Fri Mar 1 @ 16:00 CET

            - A Greener, Cost-Effective Cloud with Serverless WebAssembly (Sohan Maheshwar)
        - Open Source Day '24 // Thur Mar 7 @ 16:00 CET
            - A Greener, Cost-Effective Cloud with Serverless WebAssembly (Sohan Maheshwar)
        - SCaLE 21X // Thur Mar 14 @ 10:00 PT
            - Create a First AI Program with TypeScript and Serverless WebAssembly (Caleb Schoepp & Justin Pflueger)
        - Wasm I/O // Thur Mar 14 - Fri Mar 15
            - WebAssembly Component Model: What? How? And, why you should not ignore it! (Thorsten Hans) // Thur Mar 14 @ 11:25
            - Sustainability with Wasm? Faster, Greener Computing (Panel with Danielle Lancashire) // Thur Mar 14 @ 14:05
            - Deconstructing WebAssembly Components (Ryan Levick) // Fri Mar 15 @ 12:40
            - Accelerating ML Inferencing with WebAssembly & Spin 2.0 (Radu Matei) // Fri Mar 15 @ 16:30
        - KubeCon + CloudNativeCon EU '24 // Tues Mar 19 - Fri Mar 22
            - Building Serverless AI Apps with Spin and WebAssembly (Matt Butcher) // Tues Mar 19 @ 16:25
            - Leveling up Wasm Support in Kubernetes (Matt Butcher) // Thur Mar 21 @ 14:30
- *Call to Actions*
    - SIPs (Spin Improvement Proposals)
- *Open floor for community questions/discussion topic*
- *Community Intros*

2024-01-22 16:00 UTC

Agenda

- *Assignments for this meeting*
    - Moderator: Melissa
    - Notes: Karen
- *Demo*
    - [Joel] Python SDK with componentize-py
        - New Spin SDK - Python wrapper around Rust SDK
        - https://github.com/fermyon/spin-python-sdk-v2
        - Single command can regenerate Python API from source WIT files
        - Any experience Python dev is able to contribute to this project

- *Announcements*
    - [Karen] Upcoming events
        - NYC Kubernetes Meet-up
            - Jan 23; New York City NY, USA
        - Basta!
            - Feb 12 - 16; Frankfurt, Germany
        - Civo Navigate NA '24
            - Feb 20 - 21; Austin TX, USA
        - WebAssembly Austin Meet-up
            - Feb 20; Austin TX, USA
    - [Melissa] Spin 2.2 release at the end of the month will include Wasmtime 17
        - Wasi-preview-2: up for vote on Jan 25
        - Great news for stability and to have to play around for the component model
- *Call to Actions*
    - Look for the Spin 2.2 release next week
- *Open floor for community questions/discussion topic*
- *Community Intros*

2023-11-27 16:00 UTC

Agenda

- *Assignments for this meeting*
    - Moderator: Melissa
    - Notes: Karen
- *Demo*
    - [Kate] Authentication Component
        - Can have ubiquitous components that can be interchanged
        - Repo and blog post online explaining component model
        - Spin 2.0 allows 2 components to be 1 component
- *Announcements*
    - [Melissa] No Community meeting in December, see you in 2024!
    - [Karen] Upcoming events
        - Stockholm Cloud Native Meet-Up // Nov 27
            - Session: Server-side WebAssembly with Spin (with Mikkel Mørk Hegnhøj)
        - KCD Utrecht // Dec 1
            - Session: A Greener, Cost-Effective Cloud with Serverless WebAssembly (with Sohan Maheshwar)
        - KubeDay India // Dec 8
            - Session: A Greener, Cost-Effective Cloud with Serverless WebAssembly (with Sohan Maheshwar)

            - Might add a meet-up in Bengaluru on Dec 9
        - AI.dev // Dec 12 - 13
            - Session: What Is It Like to Be an LLM? (with Matt Butcher)
            - Sponsoring – Fermyon table
- *Call to Actions*
    - [Melissa] Build Components!
- *Open floor for community questions/discussion topic*
- *Community Intros*

2023-10-23 15:00 UTC

Agenda

- *Assignments for this meeting*
    - Moderator: Melissa
    - Notes: Karen
- *Demo*
    - [Ryan] Rust SDK changes
        - Using Spin 1.5 Rust SDK
        - More powerful than before/handlers are more flexible now
        - More support to come
- *Announcements*
    - [Karen] Upcoming events
        - KubeCon + CloudNativeCon NA '23 // Tues Nov 7
            - "Tutorial: Building Cloud-Native Applications Using WebAssembly and Containers" with Mikkel Mørk Hegnhøj (Fermyon), Melissa Klein, (Fermyon), & Ralph Squillace (Microsoft)
        - KCD Denmark '23 // Nov 14
            - Workshop: Hands on with WebAssembly and Kubernetes (with Mikkel Mørk Hegnhøj)
        - Stockholm Cloud Native Meet-Up // Nov 27
            - Talk: Server-side WebAssembly with Spin (with Mikkel Mørk Hegnhøj)
- *Call to Actions*
    - [Melissa] Spin 2.0 Board
        - Feel free to test
- *Open floor for community questions/discussion topic*
- *Community Intros*

Minutes

- *Attendance*

2023-09-25 15:00 UTC

Agenda

- *Assignments for this meeting*
  - Moderator: Melissa
  - Notes: Michelle
- *Demo*
  - [Ryan] AI Inferencing with Spin
- *Announcements*
  - [Melissa] Upcoming events
    - Cloud Native Aalborg Meet-Up on Wed Sept 27
      - "Service-side WebAssembly with Spin" with Mikkel Mørk Hegnhøj
    - DockerCon on Thursday Oct 5
      - "Wasm on Docker Desktop with Spin with" with Matt Butcher
    - KubeCon + CloudNativeCon NA '23 on Tues Nov 7
      - "Tutorial: Building Cloud-Native Applications Using WebAssembly and Containers" with Mikkel Mørk Hegnhøj (Fermyon), Melissa Klein, (Fermyon), & Ralph Squillace (Microsoft)
- *Call to Actions*
  - [Melissa] Spin 2.0 Board
- *Open floor for community questions/discussion topic*
- *Community Intros*

Minutes

- *Attendance*
  a. *Ryan L*
  b. *Michelle Dhannai*
  c. *Mikkel H*
  d. *Melissa*
  e. *Joel Dice*
  f. *Jonathan Pelham*
  g. *Steve Sloka*
  h. *Sohan Maheshwar*
- *Demo - For AI Inferencing with Spin*
  a. *github.com/fermyon/ai-examples*
  b. *Sentiment analysis example*

    c. *Give it a sentence and it will tell you if it is positive, neutral, or negative in nature*
    d. *Using the llama2-chat model → is 3.6G so pre-downloaded*
    e. *Starting by creating a new app using the spin http-rust template with spin new*
    f. *Taking text from request body and feed it into the llm*
    g. *Converting bytes from body into string*
    h. *Import llm functionality from the spin sdk (with import statement)*
    i. *Spin build*
    j. *Feed body into llm using llm::infer(model, prompt). Will get back a response.*
    k. *Included in the response is text and usage data (how quickly you got a response, etc).*
    l. *Pass text from llm response to http response*
    m. *Spin build -u, curl and pass text*
    n. *Got an error, component not allowed to use llama2 AI model. Need to add the ai_model to allow in the spin.toml file.*
    o. *Now getting an error that ai_model does not exist. Need to move ai model that we downloaded from the internet to the right place under the .spin/ai-models directory*
    p. *Curl again, takes a long time. By default we run on CPU, can run on GPU. Can also run on fermyon cloud.*
    q. *Steve: deployed and it works!*
    r. *Jonathon:*
        - *Is this quite performance limited currently? My understanding is wasm is singlethreaded currently? ?*
            - *Ryan: yes wasm is currently singlethreaded. prompts can only be done one at a time and not parallel. Prompts are expensive.*
        - *If it was a service I presume you would have multiple wasm instances handling them? Is that for version 2 of your demo perhaps?*
            - *Ryan: yes can have multiple wasm instances running*
            - *Each inference has to be handled serially but can do multiple if enough compute power*
            - *Joel: could potentially offer an async interface and you wouldn't necessarily have multithreaded support you would just need to be able to handle requests concurrently*
- *Announcements*
    a. *Workshops and talks on using spin in the upcoming conferences*
- *Calls to Action*
    a. *There is a consider-for-2.0 tag and can put an issue on the New column of that board*
- *Open floor*
    a. *Steve:*
        - *Ryan: complex question so will try to answer it*
            - *In some ways, being able to compose components together as we talked about before like not having to serialize / go through http*

        - *We don't expose that in a way that Spin users can take advantage of that yet and we are working on that*
        - *Right now, you continue to break*
        - *WASI preview 2 is the next version of WASI (webassembly standard interface). Standard way for components to talk to each other. Targeted for towards November/end of the year ish with wasmtime 14 in a format when that is a little more stable. Once that is there, we can offer some stable way for Spin users to take advantage of that.*
        - *Joel: plan for 2.0 is to support wasi preview 2 in whatever state it's at. Current goal is to ship wasi preview 2 at end of november. Spin 2.0 comes out before then but there should be adapters in place and have stable component model support. Until now, things have been breaking since there is heavy dev in tooling. Composing multiple Spin components*
            -
- *Community intros*

2023-08-28 15:00 UTC

Agenda

- *Assignments for this meeting*
    - Moderator: Melissa
    - Notes: Karen
- *Demo*
    - [Joel] WasmCon preview
        - https://pypi.org/project/componentize-py
        - https://github.com/dicej/componentize-py/tree/main/examples
            - Hoping to see component model work to wrap up in the next month or two
            - Addresses native extension aspect
        - https://github.com/dicej/spin/tree/wasi-cloud-core
        - Joe (MSFT) + Joel = exploring WASI-cloud-core for Spin
            - Have gotten feedback that Spin isn't vendor agnostic
- *Announcements*
    - [Melissa] Spin Up Hub
        - Can find templates, plug-ins, sample applications

        - To help you get started on building apps for your own use
        - Released in July
        - Always expanding and adding new items
        - Can easily add to it with a PR in the developer repo – just need a markdown file + link to the repo
    - [Karen] Civo Navigate EU' 23
        - Real World, Server-Side, WebAssembly with Spin
        - Build a Magic 8-Ball App with Spin and WebAssembly
    - [Karen] WasmCon '23
        - Workshop
        - Taking WASI-Cloud-Core for a Spin
        - Introducing Componentize-Py: A Tool for Packaging Python Apps as Components
    - [Karen] Bytecode Alliance "Componentize The World" Hackathon
- *Call to Actions*
    - [Melissa] Submit your project to Spin Up Hub
    - [Melissa] Demos for future meetings
        - Doesn't alway have to come from Spin maintainers – would love to have community members demo – please reach out on Discord
- *Open floor for community questions/discussion topic*
- *Community Intros*

Minutes

2023-07-24 21:00 UTC

Agenda

- *Assignments for this meeting*
    - Moderator: Michelle Dhanani
    - Notes: Karen Chu
- *Demo*
    - [Michelle] Spin doctor
    - [Michelle] triage rotation app
- *Announcements*
    - [Melissa] SQLite Host Component
    - [Brian] wit bindgen updates
- *Call to Actions*
    - [Melissa] Survey for meeting time
        - https://forms.gle/HpGdBdaEGXKW4GkU9
    - [Melissa] Spin 2.0 - what do you want to see?
        - Please submit issues to issue queue
- *Open floor for community questions/discussion topic*

- *Community Intros*

Minutes

- *Melissa Klein*
- *Karen Chu*
- *Brian Hardock*
- *Michelle Dhanani*
- *Joel Dice*
- *Karthik Ganeshram*
- *Joe Zhou*

- *Demo*
    - [Michelle] Spin doctor
        - Command you can run on your Spin apps - will diagnose issues and make suggestions on how to fix them[Michelle] Walkthrough of GitHub Oauth example in Spin
        - Very much a work in progress
    - [Michelle] triage rotation app
        - Been working on this with Brian and Karthik
        - Work in progress
        - Pick-2: Spin app that generates who the 2 ppl (from list of maintainers) are to help with triage
        - Schedule-month: create schedule for a month using config variable
        - GitHub Oauth app
            - Login button
            - GitHub app
            - Backend handler for redirect from GitHub
- *Announcements*
    - [Melissa] SQLite Host Component
        - Introduced in Spin 1.4
        - When you reference the DB in your manifest file, Spin will take care of it
        - Need to use API (in language SDKs)
        - Spin will handle all the communications for you
        - Play around with it & let us know what you think
    - [Brian] wit bindgen updates
        - Since Spin 1.4.1 release – updated Spin to use 0.8 version of wit bindgen; uses most up-to-date wit syntax
        - To support the component model – one step closer now
        - Rust SDK no longer takes a dependency on wit bindgen
        - [Joel Dice] Python reuses Rust SDK (not sure if it's using the latest SDK) but there should be no changes for the user – things should just work
- *Call to Actions*
    - [Melissa] Survey for meeting time

        - https://forms.gle/HpGdBdaEGXKW4GkU9
        - Want to reach as many people as possible
        - 3 question survey to figure out what days/times work for people
        - May move or alternative to a new time
    - [Melissa] Spin 2.0 - what do you want to see?
        - Please submit issues to issue queue
- *Open floor for community questions/discussion topic*
- *Community Intros*

2023-06-26 21:00 UTC

Agenda

- *Assignments for this meeting*
    - Moderator: Joel Dice
    - Notes: Kenzie Olson
- *Demo*
    - [Matheus] Spinlet: Spin plugin to run wasm32-wasi (preview2) command components as plugins
- *Announcements*
    - [Joel] WASI Preview 2 Updates
    - [Joel] Python ecosystem developments
- *Call to Actions*
    - [Joel] Python ecosystem
- *Open floor for community questions/discussion topic*
- *Community Intros*

Minutes

- *Attendance*
    - Joel Dice
    - Kenzie Olson
    - Karen Chu
    - Karthik Ganeshram
    - Mikkel Hegnhoj
    - Ivan Towlson
    - Steve Sloka
    - Brian Hardock
    - Till Schneidereit
    - Michelle Noorali
    - Mossaka (Joe Zhou)
    - Adam Reese
    - Matheus Cardoso

    - Justin Pfleuger
- *Demo*
    a. [Matheus] Spinlet - an experiment to run plugins within a controlled environment. Acts like an operating system within a workspace
    b. [Matheus] A big motivation is to update a plugin with just a single binary, rather than compiling for every architecture
    c. [Matheus] Right now, its a plugin that can run any other type of plugin. WASI is pretty powerful, only missing HTTP which can be emulated
    d. [Matheus] We can run scripts, embed a quick js runtime, rust, python. We can run all of these scripts in this environment
    e. [Matheus] There is some access control you need to give Spinlet
    f. [Matheus] If you check the shell, you can see granular level of access for the plugin
    g. [...] This plugin wraps the spin CLI itself, so you can run familiar commands like `spin build` and it's also specifiable in the config.toml
    h. [...] You can have finer grained control on upgrades. The goal is to ultimately integrate this into Spin via PR
    i. [Joel] - it's been a bit awkward to publish plugins for each platform, so it's great to see this being used in such a relevant way
    j. [Ivan] - do you have access to the output of the command so you can set up diagnostics for failures
        - [Matheus] It does have access to the output if you give it the capability. I've been thinking about forking each of the diagnoses and treatments from spin doctor
        - [Matheus] Still playing with adding more Wasm commands
        - [Matheus] Shouldn't be too hard to add Spin doctor in the future
- *Announcements*
    a. WASI Preview 2 update
        - [Joel] it will be the first stable snapshot of the Wasm component model. You can isolate various privileges to different parts of the app, cross-language composition. Looking forward to seeing this graduate to stable. We should see full UDP/TCP/HTTP support as well (off the shelf libraries should be supported as well). For HTTP, streaming + concurrent outbound requests will have these features out of the box!
        - Trevor + Pat @ Fastly have been working on host implementation in Wasmtime. We'll be able to merge our work upstream, and then adopt in Spin as well as Fermyon Cloud
        - Resource support will give us the ability to implement object oriented API to represent things like DB connections / HTTP headers as objects. This will make for more ergonomic APIs
        - Security of components, e.g. ensuring handles are non-foragable, will be hardened as well

        - Goal is to get into Wastime 12 in August, but you can start experimenting sooner (just know these are unstable so there may be breaking changes in the future)
            - [Steve] Are there notes on how to run this ourselves?
            - [Joel] There are a couple of demos in a PR - https://github.com/fermyon/spin/pull/1553
    b. Componetize-py update
        - [Joel] There are several snapshot branches, highly experimental
        - [...] Generic tool for converting python apps to Wasm components. Will add native support for libraries like NumPy and other popular libraries (which often use other languages behind the scenes)
        - Until publishers of NumPy start publishing official WASI builds, pip install won't be supported. We're hoping to get there.
        - We'll create a library of build scripts to build these popular languages to WASI. As we start to flush out proof of concepts, we can hopefully start integrating with CIs
        - There's a cargo component for Rust. It still takes quite a bit to pull everything together in their experimental phase, but in the next few months this will be something folks can use off the shelf.
        - This should be the basis for the next generation for Spin python SDK
- *Questions*
    a. [Mossaka] on the wit-bindgen work, are you going to do for all the crates? I will help to get it land on go-bindgen at least. I probably can help on the c-bindgen crate as well, since I am familiar with the code base.
        - [Joel] Will be done for the C and Rust. Hopefully that will serve as a model for Go
    b. [Mateus] Will SDKs be needed less and less?
        - [Joel] We'll have a standard interface for doing HTTP not tied to a particular company. We'll fit it WIT files and application to run the app on a platform of your choice. Whether that makes SDKs obsolete is a more subtle question. For example, Spin has ad-hoc WIT files for edge functionality. Also wit bindgen doesn't support every language edge case out of the box (for example, Adam overhauled Go SDK to make it more idiomatic and friendly for Go developers). There will always be scope for adding additional functionality.

2023-05-22 21:00 UTC

Agenda

- *Assignments for this meeting*
    - Moderator: Melissa Klein
    - Notes: Karen Chu

- *Demo*
    - Azure CosmosDB implementation for KV
- *Announcements*
    - [Joel] Component model support
    - [Joel] Sqlite prototype in main branch and open PR for interface
    - [Joel] 1.3 Milestone
- *Call to Actions*
    - [Ivan] Sqlite host component SIP
- *Open floor for community questions/discussion topic*
- *Community Intros*

Minutes

- Attendance:
    - Melissa Klein
    - Brian Hardock
    - Ivan Towlson
    - Joel Dice
    - Karthik Ganeshram
    - Kate Goldenring
    - Matheus Cardoso
    - Mikkel Mørk Hegnhøj
    - Radu Matei
    - Steve Sloka
    - Till Schneidereit
    - Karen Chu
    - Justin Pflueger
    - Duma Mtungwa
- Demo:
    - [Radu] Azure DB key has changed
- Announcements:
    - [Joel] 1.3 – planned for early June; color coded outputs (hopefully easier to see/distinguish) + cloud plug-in support
        - [Kate] have had a plug-in system for a while (has been specific to what you need for Spin), moving commands to a plug-in (called “cloud”) will slim down the functionality; cloud will be an installed plug-in; next steps will be to distribute Spin via package managers
        - [Radu] Cloud plug-in WIP: https://github.com/fermyon/cloud-plugin
    - [Joel] Sqlite prototype in main branch and open PR for interface
        - Ryan Levick has been working on this
        - [Ivan] https://github.com/fermyon/spin/blob/main/docs/content/sips/013-sqlite.md
        - [Ivan] https://github.com/fermyon/spin/pull/1534

        - If this doesn't get done for v1.3 release, will hold for v1.4 – looking for feedback
    - [Joel] Component model support
        - Work in the WASI world from Bytecode Alliance
        - What Spin calls components aren't actually "components"
        - In Spin v1.1 – introduced true component support (official SDKs won't give you real components though – you have to know what you're doing)
        - Aiming for v1.4
        - Want true native support but needs to wait for WASI preview 2 – maybe August/September timeframe
- CTAs:
    - Sqlite feature
        - https://github.com/fermyon/spin/blob/main/docs/content/sips/013-sqlite.md & https://github.com/fermyon/spin/pull/1534
            - Looking for feedback
- Community Questions/Discussions
    - [Matheus] still trying to understand difference between modules and components – what are the benefits long term vs short term?
        - [Joel] Shared nothing architecture that underpins components; level of security/sandboxing; ability to adapt a component that was intended to executed in one content in another context
        - [Till] advantages of these building blocks; component model contains a design for which components get linked together (module linking); polyglot architecture; memory (layout and architecture) does not have to be exposed
        - [Joel] high-level type checking (whereas core module type checking only considers primitive integer and floating point types)
- Community intros
    - [Steve Sloka] works for Outerbase; learning Spin and looking to use it for databases
    - [Matheus Cardoso] contributing to the project and learning it; not having to deal with Docker or Kubernetes in the future fascinates Matheus