WASM WG | Meeting Notes

~~**Fortnightly**, Tuesdays at 5:00 pm CET/ 8:00 am PST Tockify Calendar~~
On pause until working groups changes have settled

Charter

Join the Zoom Meeting

Wasm Working Group Slack

Past Meetings (TAG-Runtime channel)

Wasm OCI Subgroup meeting notes: OCI + Wasm Meeting Notes

Month day, year

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 📑

-

Action Items:

June 24th, 2025 (Canceled – no agenda items – temporary pause to future meetings)

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 📑

-

Action Items:

June 10th, 2025 (Canceled – no attendees)

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 🗒️

-

Action Items:

May 27th, 2025 (Canceled – no attendees)

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 🗒️

-

Action Items:

May 13th, 2025 (Canceled – no agenda items)

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 🗒️

-

Action Items:

April 29th, 2025

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- Joe Zhou (Microsoft)
- James Sturtevant (Microsoft)
- Luke Wagner

Agenda: 🗒️

- (Joe) Luke, what do you think about Gravity (https://github.com/arcjet/gravity)
    - (Luke) They are doing something like Jco and generating golang types for interacting. They wanted to run components in Wazero.
- (James) I was wondering if this would be a good forum for how Wasm interplays with MCP. There is a lot of noise about MCP. Component model would make a great interface for describing tools.
    - (Luke) I'd like to hear people talk about it.
    - (Joe) I can say a few words about MCP and Wasm. I have been working on a project to run MCP servers to run in a secure environment. I made a tweet that

stated that the component model could describe a great interface for MCP servers. One of the biggest concerns about MCP security is that they are granted unlimited access to system resources. In the MCP spec / docs there is little said to describe how to isolate / secure. LLM would not be able to answer what the weather's like today. Having an MCP server that provides the ability to fetch data, solves the problem. The proxy can be written in a way that has malicious intent. Tool poisoning may cause a tool to read your ssh keys and exfiltrate the data by hiding its true intent. First step is to sandbox MCP servers. VM, container, or possibly compile to Wasm. It really depends on how much security you want to get. 1) secure the entire thing in a secure environment. 2) Run MCP server as Wasm or within a VM. 3) You have 1 MCP server acting as a platform, then the tools are sandboxed in Wasm / VM / container. Tools are essentially functions. We have functions as a service. We can do roughly the same thing. I'm experimenting with 2 and 3. If they (tools) can be compiled to WASI, we can have fine control. Create an MCP server that embeds Wasmtime or Hyperlight to isolate the execution of each tool. Any component that has exported functions, the host can call into.

- What would the WIT interface look like?
  - (Joe) You just bring me a component. I translate that WIT into a JSON schema. This allows you to bring any component without implementing a specific WIT. For Hyperlight, you can't dynamically invoke.
- (Luke) The JSON describes the invocation schema?
  - (Joe) It is, but not strictly enforced. It's all human readable.
- (Joe) JSON can have JSON recursively. That was tough in WIT.
  - (Luke) This is the recursive types request. <please see vid for details/>. We eventually want to have recursive types, but may be out of scope for 1.0.
- (Luke) When I have a sequence of function calls, a handle may be useful for giving the model something a bit more concrete to work with than when you represent a resource with a number (ptr).
  - (David) So, you are suggesting the model would understand they type and the idea that the handle is an instance of that type?
  - (Luke) Yes. If humans understand OOP, then perhaps, the model will as well.
- (James) Is there any value the WG can provide to help solve these problems? Are there others in the space working on this?
  - (Joe) There is 1 company I know of that is working in this space, it's Dylibso. They have https://mcp.run that provides tools as Wasm modules that run in their server MCPx.

April 15th, 2025

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt (JAF Labs)
- David Justice (Microsoft)
- Joe Zhou (Microsoft)

Agenda: 🗒️

- KubeCon Reflections
    - YouTube Playlist of talks: https://www.youtube.com/playlist?list=PLj6h78yzYM2MP0QhYFK8HOb8UqgbIkLMc
- Wasm I/O Reflections
- Mono repos and composition
    - wac https://github.com/bytecodealliance/wac?tab=readme-ov-file#usage

March 18th, 2025

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt (JAF Labs)
- James Sturtevant (Microsoft)
- Yoshua Wuyts (Microsoft)
- Brandon Mitchell
- David Justice (Microsoft)
- Josh Duffney (Microsoft)

Agenda: 🗒️

- (Yosh Wuyts) Update on the WASI Preview 3 and the current state of the WASI
    - Yosh shares presentation and gives introduction
    - Async in 0.2 is pollable, you call poll and it blocks till one of the operation is done
        - Shows demo of wasi-clocks using pollable function subscription-duration
        - This requires a single runtime that runs at root of component, each component needs one
        - Wasi:io is used by all the other wasi packages
        - This means that if you want to virtualize stuff you need to virtualize pollable, its really hard to do this in 0.2,
    - In 0.3, we remove pollable, it moves into the runtime. If you want to know how this works what one of luke's talks!
        - Now if you want to virtualize something you can independently do just one of the
        - Future plans to make this more composable in the future
        - New example of subscribe-duration that doesn't use pollable, no return type. This will be called wait-for in 0.3
    - What are the stages of getting ot 0.3

        - First, prove it was feasible via draft (done)
        - Second, land specs in component model (done)
        - Third, update wasi interfaces (done). there is 0.3-dev specs in the place
        - Forth, build out binding generation support for rust (done but being refined)
        - Fifth, implement host support in wasitime (in progress in experimental fork)
            - This has been helpful to understand all the things, for example cancelation needs to be in scope, which wasn't initially the plan
        - Sixth, Implement second support in Jco (in progress)
            - Things are solidifying and less unknown.
        - Seventh, Fuzz testing to uncover blind spots in spec and hardening
    - Three grades of stability (from user perspective):
        - Ready for demos (almost ready for this, at kubecon, wasi-io)
        - Ready for integration (gather more real world feedback!)
            - Timeline, once these lands in wasmtime main. Will call for integrators
        - Ready and stable
            - Hard to say when
    - (David) where/how to get involved?
        - (yosh) take a look at the experimental repo, and figure out what will it take to embed it or use it. As far as implementation, we have a full team so mostly feedback and testing the implementation will be most helpful
    - (taylor) How do we make sure all the folks are aware of the updates coming in 0.3.?
        - (yosh) venues like these. There are different tiers of investment, the folks implementing, then the second tier is the ones who attend meetings, follow repos, zullip and places like that have very active conversations. Then the last group that has an interest, and for those are the blog posts and big push in the marketing side of things, particularly when we are ready for integration.
- WASI Cloud discussion items (Taylor and Mossaka)
    - Some discussion items
    - https://github.com/WebAssembly/wasi-config/issues/20
        - Specify key formats for keys is one of the last things
        - Can we restrict the format now, then maybe loosen later
        - (david) - there might be existing keys that need some transforms
            - (taylor) was a concern, but should be easy to
        - (james) -
        - (david) - there is something to be around discoverable, there needs some consistency between env and config when calling `get`
            - There is a difference between env's and config and this is where it gets complicated

        - (David) - overlapping config sources is the main concern. Would love to get feedback from users building apps with this
        - (taylor) - i might ask wasmCloud community meeting
    - https://github.com/WebAssembly/wasi-blobstore/issues/27
        - When using blob stores should there be containers?
            - Suggested this might make it easier
        - No comments, will revisit
    - https://github.com/WebAssembly/wasi-blobstore/issues/34
        - Shared in chat, bit more involved discusion
    - https://github.com/WebAssembly/wasi-keyvalue/issues/57
        - Behaviors in atomic and non-atomic
            - Lists out a few options
        - (david) set is last in writes, swap should be atomic. What is the user expected behavior?
            - Should not be undefined
        - This does look like it is defined and should have this documented, maybe we need a test that we can run against implementations.
- Announcements:
    - 2 weeks will be at kubecon at the tag runtime booth! Come say hi!
    - Wasi-tls merged into wasmtime https://github.com/bytecodealliance/wasmtime/pull/10249
    - Next meeting in a Month

Action Items:

March 4th, 2025

Recording link: CNCF Wasm-wg Meeting 2025-03-04
WASM JS/TS ecosystem slides: https://docs.google.com/presentation/d/1n7QdDxRb7I3AG9rNNfDFfoH3_eGwKX2trKHqMFewlGM/edit#slide=id.g2dd0f9dea64_0_34

Attendance: ✋ (PLEASE ADD YOURSELF)

- Max Brunsfeld (Zed)
- Marshall Bowers (Zed)
- David Justice (Microsoft)
- Till Schneidereit (Fermyon)
- Radu Matei (Fermyon)
- Victor Adossi (Cosmonic)
- Sven Pfennig (Hetzner)
- Joe Zhou (Microsoft)
- Tomasz Andrzejak (Microsoft)

Agenda: 📑

- (Max and Marshall) Zed and their use of the component model to create an extension system for https://zed.dev/
    - Notes:
        - High level experience with WASM
            - Zed went open source a year ago, needed an extension system – lots of demand for language support
            - Max and Marshall worked on this initial support
            - First project was Tree-sitter grammars (already supported compiling to WASM)
            - Wanted other languages to be able to interact/expose LSPs, realized they needed programmability
            - Were embedding JS in the past, didn't want to write JS – took another look at WASM
            - Just after a new wasmtime release – stars aligned and all needs were met (a month before and the pieces may not have been there)
            - **Max:** we wanted to use WASM essentially from the beginning, wrote some modules that passed back and forth JSON objects. Serendipitous timing on the revisit w/ WASM components coming out in `wasmtime`
        - Guest side
            - `crates/extension_api/src/extension_api.rs`
            - Some wrappers for more idiomatic rust experience (Guest side)
            - Using a global instance for the extension and then repeatedly call into it
            - Different version WITs stay in-tree
            - `extension` world (earliest, recent)
            - Found that breaking up into modules made evolution easier
        - Host side
            - On the host side backwards compat is maintained, host holds all different versions, w/ unified API that dispatches
            - Using `wasmtime::component::bindgen` w/ the individual per-version WIT files
            - Do need to do conversions often and add `From` impls for conversions
            - Using resources to expose core zed primitives (ex. The worktree resource), great for avoiding serializing worktree over the wire
            -
        - Question: What was the biggest challenge and biggest benefit?
            - **Marshall:** Did a lot of digging inside the wasmtime source code, using the different bindgen macros/options.
            - Sometimes it was hard to figure out which features to use.

                - Ensuring backwards compatibility was not clear at first, discovered use of `with` directive and was able to properly reuse.
                - Note enough examples of people doing the same thing
                - **Max:** very happy with backwards compatibility, and proper types all the way through
                - Not a clear mental model or sure of optimal use of WIT
                - **Marshall:** Built an extension builder as a different binary, using `cargo component`, for building WASM & distributing, ending up using `ComponentEncoder` itself to build & adapt
                - do some manipulation on the resulting binaries (stripping custom sections for size)
                - Also store the wasm extension version so that the host can know which version to load against
            - Question(Till): Exciting to see this progress, one of the things we've been wanting to see the use of the extension model. What do you think are right now the biggest limitations that you wish didn't exist?
                - **Max:** Wish people could write extensions in Typescript, or Python, Lua, or some language with a smaller learning curve
                - Tried ComponentizeJS a while ago, binary was too big to feel comfortable promoting
                - *We* like Rust but people in the community being able to write TS would be better
            - Question(Taylor): Have you had anyone writing in languages that have decent support (TinyGo?), or has it been mostly in RUst
                - **Marshall:** Rust is the only thing we support, someone could theoretically set up another language but haven't had someone do it yet.
                - Someone was trying to get Zig working but they didn't quite get how complicated it was.
                - Over the wire we pass JSON strings, in the Rust extension crate there are some Serde types to make things easier
                -
            - Question: What's the max size that you think would be reasonable
                - **Max:** would expect a plugin to be around 1Mb
                - Right now they mostly just download language servers and adapt zed, so they don't do much.
                - Till: Something that's coming soon is being able to have a runtime that you can reuse and the extensions can be *just* the JS files.
            - David: We appreciate the work you've done, it's cool!
- (Tomasz Andrzejak et. al.) StarlingMonkey and the JS / TS Wasm ecosystem
    - Notes:
        - **Till:** Tomasz put together the slides but is a bit under the weather so I'll cover

- **Till:** I Started the StarlingMonkey project in the BCA as a big refactoring of the pieces that existed in Fastly's JS compute runtime, largely a C++ codebase embedding SpiderMonkey
- StarlingMonkey is a BCA project, doesn't compile to native code right now – and targets the component model (but that's not required)
- One of the goals is to have everything be highly runtime – core runtime which is small, along with header files that abstracts interactions with outside world, and we have implementations
    - Fastly maintains versions that are implemented in terms of their compute API
- Builtins are shared, but not all – ex. `fetch` is not shared, because there are differences between the host APIs that make the transfer difficult.
- Cmake based build-setup that allows specifying desired builtins, so Fastly can choose specific builtins
- Working at Fermyon and we've been using this in production since last fall, with Fastly doing the same.
- There's nothing specific here, Fastly targets WASI P2 APIs, outputs work in WASI P2 environments
- Fermyon also has an SDK that provides ergonomic bindings (`spin` embeds `wasmtime` and adds additional APIs)
- Overview (see slide) – StarlingMonkey adds things like event loops and other builtins (TextEncoder, streams, etc)
- All of this runs inside the WASM instance linear memory – all inside the runtime
- Since it's interpreted we do more work to make it fast
    - We make startup faster than any native runtime can match by using wizer and loading the JS and running the top level script *before* deploying, and snapshotting the heap
    - This is hard for native code but easy for WebAssembly
    - When at runtime an instance is created (ex. Serverless HTTP), then we do almost no work before running the first line of JS, no cold startups
    - Claims of "no cold start"/"under 5ms", and in that amount of time we've already long shut down – actual request work is trivial compared to starting up the runtime
    - Can be faster than native runtimes, while providing a fresh instance & not incurring startup costs
- Weval & partial evaluation
    - The interpreter can be partially evaluated and turned into machine code – as a preprocessing step. It doesn't work for all use cases (same as Wizer)
    - Serverless use cases which have a deployment step
    - We emit a wasm file that contains optimized code for the interpreter *applied* to the interpreted code

        - 2.5-3x speedups are consistently seen
        - **Max(chat):** curious about code size of partially evaluated wasm files
            - **Till:** it doesn't end up being that much bigger
            - We shouldn't have to ship all of StarlingMonkey, the technology is there but just not implemented (not research)
        - **Till:** Still a bit slower than a top tier JS runtime
        - Component Model *does* somewhat suggest to leave highly computational tasks to other languages that are better suited
    - **Tomasz:** There is potential to use Weval to eliminate dead code
        - **Till:** Good point – WASM Build of StarlingMonkey currently has many parts that are not used.
- WinterTC is a technical committee for API interoperability across server-side JS runtimes
    - Doing decently well on support
    - Streams/Crypto will likely take a while longer
- Future
    - Goal is to get to Full WinterTC compatibility
    - Debugging via DAP & VSCode plugin
    - Rust bindings
    - Wasi P3 support
        - Being able to use streams efficiently (interop with something like Rust) as part of a processing pipeline would be great
    - Dedicated CLI & REPL
- **Guy:** StarlingMonkey has been a great simplification, being able to share pieces and compat story
    - ComponentizeJS works with StarlingMonkey to create components that are built to a specific WIT world
    - The default world in StarlingMonkey is the HTTP server proxy world – exporting an incoming handler, and importing all of P2
    - w/ ComponentizeJS you can pass a WIT world in and a JS module that fits the same shape as the WIT world
    - We follow the same bindgen rules when running components
    - Imports become world scoped (ex. `import { … } from "ns:pkg/interface@v.v.v"`)
    - Preview3 support should enable async imports
    - At the end you get a component that has the WIT embedded and matches the WIT world you passed.
    - ComponentizeJS provides features for enabling/disabling P2 imports.
        - Better debugging should be coming which will be usable with TIll's work on debugging (should be just using WASI sockets)

        - Stack traces were released last week thanks to Till's work
    - Typescript support out of the box would be excellent, there is a tracking issue, we suggest using a bundle step for now.
        - **Till(Chat):** Working on removing the bundling requirement
    - Dead code elimination would also be useful for removing code related to typescript (at Wizer-instantiation time)
- **Guy:** Jco project is a multi-purpose toolchain for JS devs working with components
    - Full JS native ecosystem, self-hosts w/ component builds
    - `jco transpile` is the main command which enables running components in JS environments (browsers, NodeJS)
    - Represents the component in JS semantics as if it was a regular JS module, at the end of the day it's bindgen
    - Browser support for P2 is still limited, some work on asyncify work
    - `jco componentize` is also present because ComponentizeJS doesn't have it's own CLI, and StarlingMonkey may get it's own CLI
    - WebIDL support layer webidl2wit – Mendy has done a bunch of work on this, without doing any bindgen we have a sort of smart intrinsic that can interact with the JS global object.
        - By following a few conventions you can automatically WebIDL related bindings
    -
- **Question:** Curious about leveraging Wizer and partial application – what use cases seem better for shipping JS source code vs wizer?
    - **Till:** Absolutely ship the source code
    - Ideally what you'd do is ship the toolchain, including wizer, and have a baseline JS support container that contains this and baseline plugins that include the JS file, as running Weval you get out the JS file and store that on disk.
    - To have things be as small as possible, at load time you can instantiate a blank StarlingMonkey build and avoid termination – in that case you don't have the startup cost, only once
- https://github.com/bytecodealliance/jco
-

Action Items:

Feb 18th, 2025

Recording link: CNCF Wasm-wg Meeting 2025-02-18
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- Jiaxiao Zhou (Mossaka)
- Leon Nunes

Agenda: 🗒️

- ~~Wasi-webgpu/gfx demos and update (Mendy Berger and Sean Isom)~~
- (Mossaka) OCI Artifact with a single layer
    - I want to have a file for a configuration file to configure wasmtime specific behaviors. I'm trying to understand where to configure this in the OCI artifact.
    - (James) I don't think this thing exists in the OCI artifact. I believe this exists as a mount in K8s or some other environment. Also, you might consider WASI-Virt for virtualization of the file system.
    - (Mossaka) In addition to runtime configuration, in a component that exports a HTTP interface and wants to do HTTP requests, would the information be available.
    - (James) This is embedded in the component and in the config metadata of the OCI artifact. I forget the exact name, but I think it's "world". At runtime it should be able to use that to determine if the runtime could run it. The idea is that in a containerd env, you could use the information to determine which runtime to use. There is some desire to do this in containerd, not just for Wasm.
    - (Mossaka) I wonder how that could be done holistically as the information is Wasm specific and embedded into the artifact.
    - (James) There is containerd issue out there describing this.https://github.com/containerd/containerd/issues/10285
- PSA (Mossaka) – Runwasi
    - It is a containerd subproject for running Wasm workloads. Recent optimization updates – we compared the shim workloads and distroless native container w/ wasmtime. We found that the Wasmtime shim is able to execute ~330 tasks / second compared to ~80 tasks / second the native shim. <more perf details, but missed> https://x.com/jiaxiao_zhou/status/1890543622375817265
    - (James) Please clarify what you mean by a distroless image?
    - (Mossaka) The difference is that the distroless native container (Linux container running in runc) vs an OCI artifact running the Wasm via the runwasi shim.

Action Items:

- (reschedule) Wasi-webgpu/gfx demos and update (Mendy Berger and Sean Isom)

Feb 4th, 2025

Recording link: CNCF Wasm-wg Meeting 2024-02-04
Attendance: ✋ (PLEASE ADD YOURSELF)

- Taylor Thomas (Cosmonic)
- David Justice (Microsoft)
- Sven Pfennig (Hetzner)
- James Sturtevant (Microsoft)
- Joe Zhou (Microsoft)

Agenda: 🗒️

- Wasi-tls phase 1 (James)

    - The champions went to the WASI subgroup and got approval to go to phase 1 and they just released the first draft
    - David: Is that thread local storage or something else?
        - This is definitely the transport layer security stuff
    - Why do we need this in WASI?
        - Example: people wanted to use their ORMs and they need TLS.
        - This is a high level interface and you don't actually interact much with the certificates
    - Will wasi-tls be incorporated into wasi-http?
        - James: I don't see it being incorporated as it is very high level. It is akin to the sockets interface
        - How do we handle custom certs/chains of trust in wasi-http?
        - Currently the two interfaces do not interact at all
        - Currently any cert chains are configured at the host level
        - You can also configure it via configuring your application
        - There is also a future where you could compose something like wasi-http with wasi-tls
- Follow up on previous action items

Action Items:

Jan 21st, 2025

Recording link: https://youtu.be/Wg9cp327zJ8
Attendance: ✋ (PLEASE ADD YOURSELF)

- Sven Pfennig (Hetzner)
- Calvin Prewitt (JAF Labs)
- Mossaka (Microsoft)
- Brandon Mitchell

Agenda: 📑

- Golang and components BCA tooling update - Mossaka
    - Our talk at WasmCon https://youtu.be/IM9Leoqc-xY?si=q-Zw3z0z2WDJObAh
    - Bytecodealliance go-modules: https://github.com/bytecodealliance/go-modules
    - Wasi-http-go: https://github.com/ydnar/wasi-http-go

Action Items:

Jan 7th, 2025

Recording link: CNCF Wasm-wg Meeting 2025-01-07
Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt (JAF Labs)
- Kate Goldenring (Fermyon)
- David Justice (MSFT)

- Radu Matei (Fermyon)
- Brandon Mitchell
- Sven Pfennig (Hetzner)
- Mossaka (Microsoft)

Agenda: 🗒️

- Happy New Year!
- Discuss Spin and SpinKube cncf sandbox recommendation document - Kate
    - Notes: Spin and SpinKube Analysis
        - (Kate) Yesterday the Spin / SpinKube analysis was introduced. Ricardo opened the doc yesterday authored by TAG leads and Wasm WG leads. Many of those folks were not involved and it may be slightly misleading.
        - (Taylor) I looked into this and I had made comments, but was misrepresented as all of the WG leads. I know when I reviewed the analysis… <see vid.. Missed context>
        - (Kate) 2 discussions. First I'd like to focus on the discussion the representation as TAG and WG leads.
        - (Taylor) I think that came from someone asking me to fact check something.
        - (Kate) it said community members and leads
        - (Taylor) If I were to take a stab, that is where it came from. I will need to follow up with Ricardo. I don't think that was his intent. I think he discussed with members of the Wasm community. I offered 1 thing on part of the text.
        - (David) The first time i saw the document was yesterday when it was linked. I suppose being a maintainer and contributor to projects in those space that my bias would be inappropriate. I would have wanted to know the document was going around and share my opinion
        - (Taylor): I think we can roll this back up to Ricardo and talk to him from there. I think the CNCF was trying to follow all the processes. I don't think there was malintent here but maybe some slip ups
        - (Danielle from Slack): The first time I saw it was yesterday evening too. It would've been nice to know it was being worked on as a wg chair (for similar reasons as to David, and making sure we got a broad community group) - at least before myself and David had our names put on it by association
        - (Radu): As it stands right now and is linked in the sandbox issue, it is the recommendation from the Runtime TAG that it should not be acceptedf because of the following terms
        - (David): Spin analysis was that it was ready and SpinKube wasnt. The SpinKube review from August 6th.
        - (Joe): Just want to raise a point; the wording in the document is confusing. It seems to be an assessment for both projects, Spin and SpinKube. Is it for 1 or for both projects?
        - (Taylor): The thing I took from this was around the Fermyon tie-ins and having Spin and SpinKube being 2 different projects. I think we need to

come through as the Wasm working group and say what we think need redress

- (James): 1 action item I'm taking away from this is that several people in the working group didn't see the document. That should probably be provided as feedback. I saw the one that came out in Oct.
- (Radu): As someone not familiar with the TAG WG, is it expected that the WG will provide feedback to the TAG?
- (Taylor): Yes, I think that should have happened. I think it could work both ways where afterward the doc could be given to the WG for feedback. … I didn't say this is the Wasm WG recommendation. Do we want to have the conversation now and fill out a new document (next Tuesday is the review)?
- (Kate): Are we making a recommendation document or a response?
- (Taylor): I think we turn this into a new recommendation document.
- (Danielle): I would be a bit concerned with ignoring and replacing the doc as written. I'm uncomfortable having my name associated with something I was not associated with. I would like Ricardo to update his comment on the doc.
- (Taylor): I will follow up with Ricardo right after this meeting to make sure we get this fixed. It does look like he edited a little. Can you check that and make sure we handle this right for you?
- (Kate): I just want to comment that the history of that doc is not complete.
- (Taylor): I believe it was copy and pasted to make editable
- (David): I'm feeling odd about rewriting the doc. We have a conflict of interest here. I don't think it would be proper to have Fermyon folks write the document. Should that be input 100%.
- (James): I don't think we should rush the doc with as much controversy as there has been.
- (Taylor): Sandbox reviews are every 2 months.
- (Radu): It would be very good to have a real conversation on some of the misunderstandings either sync or async.
- (Taylor): I think in terms of speed, let's start with distilling the feedback into a doc. I think this is the quickest, most efficient way to do this. Then we can discuss and make sure each are addressed.
- (Radu): There are comments about what should be sandboxed and functional
- (Kate): I think it's important to keep the doc as it is, so that comments can be address and original content is still available.
- (Taylor): My main concern is that folks are directed to the proper content. Let's keep the doc and make sure that folks go there later after corrected.

  David, you and I work together to get this started.
-

- Formalize interoperability criteria for the CNCF for accepting Wasm sandbox projects - Joe
    - Notes:
        - Joe: This was tightly coupled to the previous item. One of the most contentious points was the vendor specific APIs. It affects interoperability. I also made the comment in the doc. I think of a program that can be written once and run in many places. Anything on top of WASI, should we say that this project has no interoperability with other projects? As a WG group, we should have a criteria for interoperability across projects.
        - David: I think this is a conglomerate of issues. One is historical with the k8s project and all of its dependencies. Multiple cloud vendors and different dependencies. What is being done at the runtime level is slightly different, and we can't make a direct analogy here. We are putting a static binary on a node. We don't have the same worries that were probably driving that comment. I think we need some clarification
        - Taylor: When the reviewers originally looked at SpinKube, they said, "hey there are all these wasm projects in the cncf and spinkube seems to only be supporting Spin". I think we say, what are the things we accept for the standards… W3C, WASI P2 etc. I think the TOC is coming from the previous background of Kubernetes and saying Woah There! And it may not be accurate necessarily. We need to work on a new document async
        - Sven: What is the minimal set for cloud native that everyone should support?
        - Kate: I also think that for people who are not familiar with WASI, they may not know that the interfaces can be used by anyone. PG interface defined by someone could be used by anyone. I think there is also concern that Fermyon was in the namespace for the interfaces. It is normal for rebranding / renaming when accepted for Sandboxing.
        - David: Part of it was the proprietary naming and the other was the naming of SpinKube (including the Kube word). These are all good things and probably need to make sure these things exist.
        - Taylor: Joe are you interested in creating a document and start linking in some issues?
        - Joe: I am happy to do that. Disclaimer, I am a maintainer of SpinKube
        - Taylor: I meant that because David and I will be working on the WG recommendation for Spin, we'd love it if you could work on the doc on the interoperability of Wasm projects.
        - Joe: Clarifying that this may take awhile to define this but i can get it started
- Follow up on project presentations for this year (Taylor)
    - Taylor: We reached out to projects on who we wanted an update from. WebGPU and WasmEdge folks want to give updates.
    - Calvin: I haven't nailed down timing yet, but they're willing to do it (the preview 3 work

    - David: I have been out of town and not working since out last call. I will take it on right after this
    - Joe said he'd present the Go work done by the BCA for Wasm components in his group.

Action Items:

Dec. 24th, 2024

Canceled! Enjoy the Holidays!

Dec. 10th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-12-10
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- James Sturtevant (Microsoft)
- Calvin Prewitt (JAF Labs)
- Sven Pfennig (Hetzner)
- Brandon Mitchell

Agenda: 🗒️

- (Justice) Buildpack support for Wasm components and OCI Artifacts?
    - There was a conversation at wasmcon on making it easy to use
    - Intro to build packs and what they do
    - (Taylor) Seems like its a pretty good idea to make things easier
        - Whos going to work on it? It can be difficult for a small group like us to maintain and keep up as we continue to move everything forward.
        - Maybe a net new contributor, to make sure we can keep it going and up to date
        - Please speak up if interested!
    - (Justice) Do we have alternatives?
        - What's wash doing?
        - Do we want to do this in every language?
    - (Taylor) our approach is to create a wrapper up open source tooling. It's meant to make it easier until open source tooling catches up.
        - You do have to add the support for each language
        - It should be in language but it might take some time
    - (Taylor) leverage CNCF machinery, pass it up to tag-runtime and see they can help find someone to contribute
    - (Justice) maybe sig-app? Needs a little spec changes to build pack to support image distribution to support artifacts
    - (Sven) LF Mentorship programs
    - (Taylor) good idea, should consider our priorities and look at when that comes up
    - (Calvin) working with heroku folks in packaging sig in the BCA

        - Interesting conversations at wasmcon
        - Interest in getting more involved and looking hopeful
- Wasmcon videos: https://www.youtube.com/playlist?list=PLbzoR-pLrL6o0UD4PoO0H_RnoToEiWBIS
    - Favorites:
        - Contain yourself (OCI and Wasm) by James and Taylor
        - Go and Wasm by Joe and Randy
        - Is wasm real: Multiple talks with Wasm users on stage.
        - WebGPU
            - Can get Mendy and Sean to come
    - Things that worked well and should replicate: Proximity to kubecon, and the first day where new folks could come and ask questions
    - Is there a wasmcon EU?
        - There is a https://wasm.io/ (CFP closes on Dec 22nd)
        - Might (not confirmed) be a wasm pre-day for kubecon
- People to invite to meetings next year - Taylor
    - CNCF working on tag-reboot
    - Be good to bring in Sean and Mendy to talk about WebGPU – (Taylor)
    - WasmEdge folks (Taylor)
    - Someone to talk about Preview3 (Joel/Yosh/Luke) – (Calvin)
    - Zed Editor – (Justice)
        - https://zed.dev/blog/zed-decoded-extensions
    - StarlingMonkey (Guy, Till, Tomasz) – (Justice)
    - WASI Interfaces
        - Preview 3
        - WASI Cloud
    - State of the languages
        - Go - Joe/Randy
        - Kotlin JetBrains
        - .NET
        - TS/JS
        - Rust
    - VSCode (Dirk Baeumer et. al.) – (Justice)
- Status on projects using Wasm in CNCF
    - Should bring in projects to report
    - What are they working on
    - What do they need help with
    - Challenges
    - Could be a blog post/report
    - This would help with “Advisory Role” to CNCF
        - [David] Whose the target?
        - [Taylor] Tag runtime, but should also be cncf developers
- Announcements and news
    - [Taylor] Wasmcloud made it to incubating in CNCF! 🎉 🎉

    - [David] Hyperlight was opensourced, working on some wasm support

Action Items:

- Follow up on Speakers (assignments above)
- Think about other projects to bring into to talk here

Nov. 26th, 2024 (Canceled)

Nov. 12th, 2024 (KubeCon – Canceled)

Oct. 29th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-10-29

Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice
- Calvin Prewitt
- Taylor Thomas

Agenda: 📑

- (David) Discuss increment and swap from WASI Key Value @0.2.0-draft2

Action Items:

-

Oct. 15th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-10-15

Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice
- Calvin Prewitt
- Taylor Thomas
- Brandon Mitchell
- James Sturtevant

Agenda: 📑

- WASI blob store issue with if-match
    - https://github.com/WebAssembly/wasi-blobstore/issues/22
    - S3 only supports on GET but not PUT
        - (Calvin) You still have an edge case with GET where it doesn't match. You could also still commit with PUT…
        - (Thomas) We did something similar with K/V where we added CAS. Can we do the same thing with S3 or any other implementation? If it can be

papered over, then we have it in the main interface, otherwise we have it in a separate interface.
    - (Calvin) I think we need “a way” for doing transactional updates. It would be nice to have the operations at all. I don’t know if you can fake this in S3.
    - (David) Looked into this as well and found this to be the case. Similar to what we did with other projects, don’t see a way to make this transactional. S3 is something we want to support since its the most popular
    - (Taylor) You just support containers and types. Then for the extended world, it would also support the transactional stuff. That’s like 3 - 6 months out. I’m good if something were to take a borrowed bucket. Luke has also brought up extends where it kind of mixes in extended interface.
    - (Calvin) What do you think we should and when, timeline?
    - (Taylor) Probably best to make another interface that provides the transactional behavior. When push comes to shove, as long as it matches S3, folks will be reasonably happy. Each implementation of a blob store have their own features.
    - (Calvin) I could work on an optional interface to do this.
    - (David) optional interface shouldn’t be to much work but might not be ideal. Maybe you will get down the implementation and see something that will make more sense.
    - (Calvin) getting ergonomics right will be tricky
    - (Taylor) why did wasi come about? Initially thought it was posix system interfaces we wanted but ended up we wanted capabilities. These are suppose to bootstrap the community but really future could be that folks are going to create custom interfaces to benefit platforms you are running on. These will really be the 80% case to get you started. If we can meet the s3 api then that give folks enough to get started and we can iterate and help folks work on better interfaces.

Action Items:

- (Group) identify KubeCon talks of interest

Oct. 1st, 2024

Recording link: CNCF Wasm-wg Meeting 2024-10-01
Attendance: ✋ (PLEASE ADD YOURSELF)

- James Sturtevant
- David Justice
- Calvin Prewitt
- Joe Zhou

- Taylor Thomas
- Brandon Mitchell

Agenda: 🗒️

- Bytecode Alliance Plumbers Summit review (any attendees joining this meeting)
    - Notes:
        - (Calvin) Great to meet everyone in person.
        - (James) I enjoyed the Bird of a Feather time. It was great to have discussions with folks.
            - C# we essentially discussed where we are at and how we can progress. Pavel got sockets running, which was a big one for users trying to connect to SQL Server. C# SIG lang meeting is at this same time on our off weeks.
        - (David) I really enjoyed Dave Bakker's TLS implementation
        - (Taylor) I led a couple Bird of a Feather talks. The packaging talk across languages was super useful. We decided on some new terms. You have a component, a locked component (specific version), bundled component (all ready to execute). Should be able to provide an overview later.
        - (Taylor) WASI Cloud what we need to do to get to phase 3 for config and k/v. WASI blob advancements and what's left to advance. We want to get the ones we are confident on and move them forward. In a month or so, we want to get a vote on advancement.
        - (Joe) We decided to drop WASI SQL for now. There are now 5 interfaces.
        - (Taylor) We still have some conversation in WASI messaging. We really want to get the ones we are confident about graduating.
        - (Multiple) {discussing CAS and batch k/v implementations}
        - (Taylor) WASI-messaging blockers? I would like to focus on the other interfaces first since it is the most contentious.
        - (Taylor) Showing the secrets interface and adding to WASI:config (https://github.com/wasmCloud/wasmCloud/blob/main/wit/secrets/wit/secrets.wit). The key difference is the store interface and reveal interface. It allows you to have a security boundary between who can handle a secret and who can reveal a secret. Perhaps, down the line, we may have specific interfaces like key signing.
            - (James) If you didn't want a component to get secrets at all, is there RBAC or something?
            - (Taylor) That is host functionality.
        - (Calvin) I'm excited about being able to use secrets without revealing them. We could ensure that when we provide a resource to a guest and know that the key is only being passed, but not used in an intermediate component.
- Wasm package tooling demo and update (Taylor)
    - Notes:
        - Terminal demo!!! \o/

    - Been working on Wasm packaging tools that have been integrating the OCI Wasm artifact (see the wg-wasm deliverables page https://tag-runtime.cncf.io/wgs/wasm/deliverables/wasm-oci-artifact/)
    - wkg allows you to work with OCI Wasm artifacts
    - wkg should be able to be used across languages
    - You can use it to push and pull both Wasm apps as well as Wasm component interfaces
    - What I'm going to show today is `wkg wit {update|fetch}`
    - What existed before was a deps file tree. What we changed it to is wit dir and then package into a *.wasm WIT component </shows some WAT>
    - You can see all the imports and exports using `wasm-tools print *.wasm`
    - </shows tree in directory with only src and wit dir> which is much less content than we're used to.
    - Calling `wkg wit fetch` pulls deps into the wit dir and creates a wkg.lock file
    - Then, `cargo build` and it just works.
    - It will be integrated into cargo component soon, and other language toolchains can use this too.
    - </shows wasm-pkg-tools repo https://github.com/bytecodealliance/wasm-pkg-tools#well-known-metadata>
    - The goal is that each language will be able to use the same lock files and config files. That is how we'll be able to use the std toolchain.
    - There might be some movement to merge into wasm-tools. Right now it's separate b/c wasm-tools is not registry aware, but wkg is.
    - (Joe) I'm personally very excited about the wkg OCI build stuff. As a champ for the Go language, every runtime needs this. It will be very useful.

Action Items:

- (David) CNCF take on BA Plumber's Conference paper

Sept. 17th, 2024

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice
- Calvin Prewitt
- Taylor Thomas
- James Sturtevant
- Brandon Mitchell

Agenda: 📑

- (Taylor) Update on OCI

    - Notes
        - Back in May we worked together to standardize a representation of a Wasm artifact. Wanted to show off what that looks like and give some future direction. Shoutout to James for tooling and .NET to deliver OCI support.
        - <Taylor sharing his screen>
        - `wkg` has been released. It is a set of tools for pushing and pulling of OCI Wasm artifacts. If you are pushing runnable artifacts, you can use this. You can see you can pass in auth or insecure. All of these things will be able to pull down any type of Wasm artifact you like.
        - `wkg` also has get and publish. `get` will pull down by package name, and writes out a WIT file. If it's a WIT component, it will write the WIT file. If the component is a Wasm component, it will write the .wasm file.
        - `wkg` publish will publish a component.
        - `wkg` is both OCI and Warg aware.
        - This enables folks to use the same OCI security tools that they use today since it is all OCI compliant.
        - <now showing the config file describing how to configure registries>
        - We are going to move over the WIT tool. There will be no deps.toml anymore. It will be replaced with a tool that will introspect and then pull down the interfaces needed. Lang toolchains will be able to take advantage of this to provide component support.
        - Convenient WASI type crates (Rust):
            - https://crates.io/crates/wasi
            - https://docs.rs/wasmcloud-component/latest/wasmcloud_component/
        - https://crates.io/crates/wstd
            - Might be a contender to donate to Bytecode Alliance. There are a large number of crates that need to be updated. You need something in the guest to be an async executor to run async tasks. Not ready today, but it's getting quite close.

Action Items:

- Bring Plumber Summit opinions and important community topics

Sep. 3rd, 2024

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt
- Taylor Thomas
- David Justice
- James Sturtevant

Agenda: 📑

- State of the Ecosystem questions: https://github.com/cncf/tag-runtime/issues/182
- Tag-runtime and wasm wg will have space at kubecon
- News
    - WasmCon is coming up in November https://events.linuxfoundation.org/wasmcon/program/schedule/
    - Bytecode Alliance Plumbers summit at end of Sept (26th/27th)
    - https://github.com/bytecodealliance/wasm-pkg-tools updates on the way
        - Publish support for wit packages and future support for library pkgs
        - Joe has in Go https://github.com/ydnar/wasm-tools-go/pull/146 to provide a similar experience as C#
    - (Taylor) Open a new PR in the WASI KeyValue interface. It will be a breaking change.

Action Items:

-

Aug. 20th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-08-20
Attendance: ✋ (PLEASE ADD YOURSELF)

- Bailey Hayes
- Luke Wagner
- James Sturtevant
- David Justice
- Kevin Conner
- Brandon Mitchell

Agenda: 📑

- [jsturtevant] Demo using the OCI artifacts to build a c# wasi-http component
    - csproj in C# showing building for wasi-wasm and the BytecodeAlliance.Componentize.DotNet.Wasm.SDK to make it easy to build a component from .NET.
    - Component bindings in C# and James demo'd bringing in WIT from OCI registry with the above componentize SDK.
    - Shows the wit.wasm and then all of the language bindings.
    - Spec for Wasm OCI layer https://tag-runtime.cncf.io/wgs/wasm/deliverables/wasm-oci-artifact/
    - James uses wasmtime serve against the built Wasm component
    - James mentions that Taylor has this working with `cargo-component` for Rust. It behaves essentially the same as this C# demo.
    - James used a tool built by Taylor and says that being able to point wit-bindgen directly at a component, and it worked right away.
    - Then can publish this component to OCI with wkg.
    - https://github.com/bytecodealliance/wasm-pkg-tools/tree/main/crates/wkg
    - https://github.com/bytecodealliance/componentize-dotnet

- [knrc] CEL Playground Presentation
    - https://playcel.undistro.io/
    - Written in Go and compiles to Wasm to run in the web
- https://wasmcloud.com/innovation-day
    - CNCF wasmCloud community-driven half-day presentations and Wasm hackathon

Action Items:

Aug. 6th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-08-06
Attendance: ✋ (PLEASE ADD YOURSELF)

- Taylor Thomas
- Calvin Prewitt
- Leon Nunes
- Brandon Mitchell
- David Justice
- Sven Pfennig
- James Sturtevant

Agenda: 🗒️

- Next deliverables to work on as a group (group discussion)
    - Todo:
        - More whitepaper/demo's
        - Finding new tools in the ecosystems to discuss
        - Ecosystem updates(across all platforms)
        - WASM architectures in the wild/production or Case studies where WASM has been used
- Wasi-cloud update (Taylor)
    - Todo find people who can do a case study(runwasi, other CNCF projects)
    - Links to the wasmtime wasicloud
        - https://github.com/bytecodealliance/wasmtime/tree/main/crates/wasi-runtime-config
        - https://github.com/bytecodealliance/wasmtime/tree/main/crates
        - Research around Key/Value store https://github.com/WebAssembly/wasi-keyvalue/pull/46
    - Conditional Statemtents https://github.com/WebAssembly/wasi-blobstore/issues/22
    -
- Updating Deliverables OCI page with implementers (Taylor)
- Kubernetes Workload Identity in SpinKube (Justice)
    - https://github.com/spinkube/skips/pull/9

Action Items:

- Talk to TAG Runtime about getting some time at the TAG runtime booth in KubeCon (Taylor)

July 23rd, 2024

Recording link: CNCF Wasm-wg Meeting 2024-07-23
Attendance: ✋ (PLEASE ADD YOURSELF)

- Brandon Mitchell
- James Sturtevant
- Leon N
- Taylor Thomas
- David Justice
- Luke Wagner
- Sven Pfennig

Agenda: 🗒️

- Update on OCI integration work (Taylor)
    - Work continues on wasm packaging and cargo-component integration
    - James - Containerd (https://github.com/containerd/containerd/issues/10179)
        - Needs some help on this if someone wants to work on it right now otherwise will get to it in the next few weeks.
    - https://github.com/bytecodealliance/componentize-dotnet/issues/28
- Wasi-messaging discussion (Taylor)
    - Portability criteria
    - Request/reply interface + abandon/complete message
- Next deliverables to work on as a group (group discussion)
- Announcements
    - LFX Internship: https://github.com/cncf/mentoring/blob/main/programs/lfx-mentorship/2024/03-Sep-Nov

Action Items:

July 9th, 2024

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 🗒️

- Cancelled

Action Items:

June 25th, 2024

Recording link: (To be added…)

Attendance: ✋ (PLEASE ADD YOURSELF)

- Taylor Thomas (Cosmonic)
- David Justice (Microsoft)
- Calvin Prewitt (JAF Labs)
- James Sturtevant (Microsoft)
- Brandon Mitchell
- Leon Nunes

Agenda: 🗒️

- WASI CG Update on Phase transitions and voting (Taylor)
    - Notes
        - WIT additions for an interface can have @unstable added without needing the full vote. Once it will become stable it will require a vote.
        - https://github.com/WebAssembly/component-model/blob/main/design/mvp/WIT.md#feature-gates
        - Unstable = beta feature and can not be relied upon. It is an unstable portion of the interface.
        - Since = it is a new addition according to semver. I'm going to add a function foo, and if someone is using an interface greater than or equal to the version they are using, they can use foo. We've learned from the mistakes of the past and this allows us to version forward.
        - Since and Unstable are live an available to use now.
- TAG Runtime Wasm Questions (Taylor)
    - Notes
        - We were asked to do this about a month ago. I finally got around to adding some questions. This is how TAG runtime measures the fit for Sandbox and Incubation projects.
        - {Taylor sharing the doc}
        - (James) where it fits in the wasm space; perhaps, we should consider where it fits in the cloud native space.
        - (Taylor) {adding clarifying question to the doc}. This is good and similar to why we moved OCI stuff into Bytecode Alliance.
        - (Calvin) What are these questions meant to do?
        - (Taylor) If someone is coming in for Sandbox, will it meet the goals of the CNCF and TAG Runtime. This is to provide information for the TAG to help guide questions for the specific domain.

PSAs:

- (Taylor) Update on OCI work
    - WKG (wackage https://github.com/bytecodealliance/wasm-pkg-tools): is available for use to push and pull OCI artifacts for components.
    - (James) I have an issue open in containerd. We also have some internal tooling in runwasi. I'm using the library that Taylor has made, but I need that package to move forward. I'm going to bring it to the community calls to get some feedback.
    - (Taylor) We're moving it into language build chains. Working on cargo component now.

    - (James) How are you integrating it into cargo component?
    - (Taylor) {pulls up wasm pkg tools repo and shares}
    - (Calvin) cargo component has warg integration. We are building an abstraction interface to abstract the underlying protocols used (OCI, Warg, other).
    - (James) I'm one of the maintainers for .NET component build tools. I'm wondering if there is a way I can integrate this work?
    - (Taylor) It's the things that are standardized that will be used broadly. We are targeting development and deployment use cases. I need to document it, but you should be able to specify a toml file at a default location that everyone can use. It would be nice to have a config format at a given path that would enable each language to use it.
    - (Calvin) It is worth noting that there is an OCI subcommand. There are other ways to map namespace:name that would be used to resolve a mapping a registry name (bridging the gap between identifiers between Warg and OCI URIs).
    - (Taylor) work continues apace.
    - (Calvin) I assume you need everything in .NET?
    - (James) It pulls in a bunch of tools. I could potentially pull in the package to extract the WIT. I need to see how this works.
    - (Calvin) It would be good to get you involved now so we can get feedback to ensure it works for your use case.
    - (James) Maybe when you are ready you could do a demo.
    - (Taylor) That's the plan. Each language will do it a bit differently. Golang will likely use `go generate`. I'm at work on cargo component today. Hoping to have something next meeting.
    - (Calvin) `wac` also is registry aware. There is also a tool for publishing WIT to a registry and will likely be a subcommand in `wkg`.
    - (Taylor) We are trying to get rid of the need to copy and paste interfaces, and just pull down the packages you need.
    - (David) When do you think folks can start publishing their component interfaces.
    - (Taylor) You can do this today with the CLI tools. Creating a new WIT interface and then push it to a registry, we want to do that soon. We want to bundle publish to OCI and then be able to pull it down. Next step is making it easy for a developer to pull and then use the interface. (Calvin) probably some time in July for OCI, for today, Warg.
- WasmCon
    - CFP Closes: Monday, July 8 at 11:59 pm PDT (UTC -8)
    - (Taylor) would like to do a dinner with folks who are around.

Action Items:

June 11th, 2024

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- Taylor Thomas (Cosmonic)
- Jiaxiao (Joe) Zhou (Microsoft)
- twas

Agenda: 🗒️

- TAG Runtime Wasm Questions
    - Follow up from the previous meeting. Please come prepared to discuss any group feedback.
- Charter Update (Taylor)
- OCI Work Update (Taylor)
- Announcement: https://github.com/cncf/sandbox/issues/103 (Joe)

Action Items:

May 28th, 2024

Recording link: (To be added…)

Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- Luke Wagner (Fastly)
- Calvin Prewitt (JAF Labs)
- Taylor Thomas (Cosmonic)
- James Sturtevant
- Brandon Mitchell

Agenda: 🗒️

- Observability in Wasm (Joe)
    - Specifically in the OCI runtimes landscape
    - (Joe) shares document (linked above)
    - Looking for feedback
    - Where we are
        - Some Projects are working on OTEL support
        - Containerd 1.6 has support for OTEL with PR open to pass tracing to context to shim
        - Cri-o has support for 1.23
        - Runwasi - add support this month
        - Youki - How do we pass tracing context (annotations or spec?)
        - WAsmruntimes
            - Spin - experimental support
            - Wasmcloud has OTEL since 1.0
    - Demo - of runwasi OTEL support
    - (Taylor) - What's the plan?
    - Where we want to be
        - Broaden OTEL adoption
        - See all the way from request in Containerd to execution in wasmtime (or other runtime)

- David - these traces seem to be on the Infrastructure side vs application workload. How do you see traces in infra vs application?
    - (joe) - there is a difference between the two. End users only want to see application. But this document put emphasis on operational side for operators to understand their cluster
- (Luke) - plan to get parameters passed to wamtime, export name and parameters
- (taylor) - this is what we do in wasmcloud
- (james) - we started using this and found a performance issue very early in adding
- (joe) - thinks this might be more on the application side vs the operational side.
- (luke) - is there a single stream or more separate endpoints
- (joe) - yes there are filters, needs more
- (david) - thinking about talking between components. This is the application domain space. In the kubernetes context, sidecars are used to collect and transport, should shims share their OTEL context? They could be going to same place but different folks looking at it. There could be value but is it a step over a boundary? Needs more discussion
- (Joe) - whats next continued
    - Improve logs, tracs, metrics across all ociruntimes to wasm runtimes
    - Perf measurements - net impact is minimal
    - Document - how to configure all these things
- (Taylor) - are you collaborating with wasi-observe? So this can be passed seamlessly into application boundary
- (joe) - haven't but will take a look
- (Danielle) - Caleb took this over from Danielle and Joe has been working with them
- (Taylor) - Wasmtime might not be doing it due to perf, wasmcloud has noticed this, in particular some scenarios cause this significantly. What have you tried around this performance? Who have you started talking with wasmtime folks?
- (joe) - early stages, haven't talked with them yet. Agree that there is overhead, did introduce feature flags in all the projects so far, but will continue monitor this and add more abilities as we go
- (joe) - who should we talk with at wasmtime?
- (luke) - next wasm component model meeting would be a good spot to bring it up. There is also a wasmtime meeting, lots of overlap
- (Luke) - is there a way to turn off when noone is really using it?
- (joe) - macro expands and adds a little but overhead but can use different levels
- (luke) - is there a dynamic flag that can be turned on?
- (Danielle) - either specify a collector or not.
- (luke) - it is on process level. Turn it on it on/off by restarting process
- (taylor) - there is a lot of things you can do, but it gets pretty complicated with different samplers. But wasi-observe lets you swap this out.
- (luke) - in firefox could dial up or down the logging

    - (danielle) - when it was on it was below a AWS overhead. Depends on how fine grain.
- OCI crate update (Taylor)
    - Follow up from last week, talked to a bunch of different folks
    - Puting it into BCA - https://github.com/bytecodealliance/governance/issues/88
    - It is separate package, so we could have maintainers from many communities
    - In foundation, if you have any questions
    - (calvin) Wasm-package tools
        - Creating shared tooling that can talk to warg and oci backends
        - Wkg cli that lets you do this in simple step
    - Rust crate be used in wasm-pacakge
    - Support for crate did make it into wasmcloud and seems to be working.
- TAG Runtime Wasm Questions (All)
    - Take a look at this doc and provide feedback for questions on the runtime sandbox projects
    - Do this async and come back in two weeks to discuss. June 11th is next meeting

Action Items:

- TAG Runtime Wasm Questions - add questions

May 14th, 2024

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- Taylor Thomas (Cosmonic)
- David Justice (MSFT)
-

Agenda: 🗒️

- OCI Updates (Taylor and James)
    - OCI crate update and where should we put it (Taylor)
        - OCI crate is in progress, but not ready for today
        - Where should we put it when we're ready?
            - Trying to put it in containerd, Taylor is going to host it until we figure that out
    - OCI doc live: https://tag-runtime.cncf.io/wgs/wasm/deliverables/wasm-oci-artifact/
        - It is merged and available on the website
- Update on wasi-messaging (David/Taylor/Joe)
    - PR will be open soon with some changes
    - If you'd like to get involved, let us know!
- Wasm news
    - (Taylor) WasmCon has been moved to alongside KubeCon in NA. The CFP will be reopening. If you have other things to submit, please do so. Please also go

out to talk to all you Wasm friends, browser and server-side, and get them to come hang out!

Action Items:

- Joe going to talk to containerd org people and see if we can put the wasm-oci crate there

April 30th, 2024

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- Lann Martin (Fermyon)
- Brandon Mitchell (independent)
- Dan Bryant (independent)
- David Justice (MSFT)
- Taylor Thomas (Cosmonic)
- Jiaxiao (Joe) Zhou (MSFT)

Agenda: 📑

- OCI Manifest Discussion and Approval
    - WASI OCI Design
    - One open question from Lann.
    - Summary of last meeting: we are trying to find a way to represent Wasm components / bins in an OCI image spec, so we can have tooling that works across many projects. It allows us to build tooling across projects. We would like to eventually support a component that can be exploded into individual layers to save on storage and make layers more reusable.
    - The big points are
        - The config media type - v0 media type. May want to rev to additional versions as we learn more.
            - There are some fields you will recognize from in the config media type.
            - Layer digest to provide uniqueness to the config. We decided not to use the root fs, but rather use the digests.
            - If it is the Wasi P2 type then you should include these imports and exports. It allows the runtime to ensure it has the exports that are expected. This allows rejection if exports are not available at runtime.
        - The layers will all be of type application Wasm. The top layer is the entry type. There is some new work being done in Wasm that allows us to do content hash encoding that will allow us to glue it back together. It will be defined in the top component.
            - Each has a Wasm media type
        - Why use index 0 vs another media type for the top layer?

        - The key thing about representing components is that they can be both a library and an executable. You can always re-compose a component to be an executable. We don't want to indicate it's a different media type. We used index 0 as the entry point to indicate the entry point.
        - (Lann) One reason not to not use the media type. Browser can not load a different media type than the current media type.
        - (Brandon) If you pull the image from registry, it is not going to say the Wasm media type. Rather, it will say it's a blob stream.
        - This may be registry implementation specific.
        - (Taylor) It is just an indication of where to launch from. We can clarify why we chose index 0 in documentation.
        - (Lann) Does v0 imply instability?
            - (Brandon) Not specifically to me. From our side, it's just a unique string.
            - (Taylor) We'll document why we started there.
            - (James) Perhaps, we'll change it to v1. It's worth considering.
- Open question from Lann: (registries SHOULD accept extra / unknown layers, and consumers should be the only ones that reject extra layers)
    - (Brandon) Yes, registries should not care. The goal is consumers not registries.
    - (James) I don't think that consumers should reject it. If we introduce an expanded view of components, we shouldn't need to bump versions.
    - (Brandon) What should consumers do in the future if they encounter multiple layers? Once you have tooling that works with multiple layers, ok. You may break existing clients.
    - (Lann) How are things like external SBoMs / sigs are attached? Are there other kinds of data that are attached to an image that we'd want to put into a layer.
    - (Brandon) I hope that there would be an additional manifest that points to the target image manifest.
    - (James) Possibly as a referrer.
    - (Brandon) It should not mutate the target image.
- Any other questions / concerns / comments?
- Where will this design land?
    - (Taylor) suggestion, are we ok moving forward, then we can decide where to put it.
    - (David) Did we have any other concerns from Radu other than the comment in the doc?
- (Liam) Has anyone done a road show with major registry players?

            - (Taylor) This has been compatible with the OCI image spec (thank you, Brandon). I think we take this to registries and see if they want to support it.
            - (Liam) The meetings before the meetings will help it not land correctly.
            - (Taylor) This is not binding, it's more of a community guidance.
            - (Liam) Do we have the right people / stakeholder to talk to.
            - (Taylor) I think it is a good follow up as we are starting to make implementations. Imagine going to GitHub and you see a nice UI where it shows the component. However, no one needs to do anything specific to support it.
        - Any hard no's or questions?
            - (Sajay) Are there any runtime maintainers like containerd that need to review this?
                - (James) once we have agreement here, I was planning on starting a discussion with containerd folks.
        - (Taylor) Doesn't look like there's major concerns. At this point, we should decide where we can put this (out of a Google doc). I know there's folks that are interested in working on this. I would like to avoid single threading the conversation. We can branch out to the interested parties.
        - (James) Was asked where media types came from and related questions in the containerd group. They advised to get some feedback. Now that we have that feedback, we can post this and go back.
        - (Sajay) When we did the OCI spec, for each implementor we opened a PR so that we could add issues to track, and not have a single uber issue.
        - (James) This would help to solicit feedback from interested parties like Liam was suggesting.
        - (Taylor) Where should we put it?? Luke suggested we put it near the component model. Other option could be its own repo. Probably be good to put it in an existing place.
        - (Lann) Suggested to put it in https://github.com/cncf/tag-runtime/tree/main/recommendations/sandbox
        - (Taylor) This is something that we will need to bring up to TAG Runtime, so this makes sense. I second the recommendation. Perhaps, under wasm-wg and open a PR.
- PSA's Community News
    - WasmCloud hit v1.0!! It is ready for folks to use. Please send any questions over to Taylor
    - WasmCon will have an important update blog post tomorrow.
    - https://wa.dev/ should have some upcoming updates
    - We are going to have a vote to move WASI key / value to phase 2 and then progress to phase 3 with implementations. We are also working on WASI messaging & blob store to do the same thing

    - We have been working in WIT Bindgen to add C# resource support. Big shout out to Joel Dice implementing resource support!
        - https://github.com/dicej/dotnet9-wasi-http-example
        - https://github.com/bytecodealliance/wit-bindgen/pull/939
    - Reminder that the Bytecode Alliance community stream at 9pm US ET: https://www.youtube.com/live/qqLAUz3-W3k?si=LvkK107xuMnPSDKY

Action Items:

- (James / Taylor) Open the PR for the proposal in cncf/tag-runtime.

April 16th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-04-16
Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt (JAF Labs)
- David Justice (Microsoft)
- Taylor Thomas (Cosmonic)
- Luke Wagner (Fastly)
- Brandon Mitchell (independent)
- Jiaxiao Zhou (Microsoft)
- Danielle Lancashire (Fermyon)

Agenda: 📑

- OCI Wasm Proposal Report and Discussion (James and Taylor)
    - WASI OCI Design
    - Notes
        - Over the past couple months we have been working together as a sub-group. We wanted to create an agreed upon standard for OCI Wasm Artifacts.
        - This is not a W3C standard. This is just an agreed prototype.
        - The manifest and config compose the spec. The manifest is meant to be used for any component / module. There are 2 key things. The media type will stay the same as an image manifest. The config media type will be (see the doc). We will be able to add a v1, v2 as the config type evolves.
        - If you see something that's missing or wrong, please let us know. We may add existing or add to the next version of the media type
        - We wanted to leave the layer definition open to enable exploding components / imploding components such that we can not duplicate layers and efficiently store layers.
        - Exports (functions exported) and Imports (functions needed to be imported to run) in Wasm components
        - Config
            - Arch: wasm
            - Os: wasip2 aligning to GOOS values

                - Layer_digest: are included as layer digest to ensure the config is unique
                - If wasip2, there must be a component section, which will contain a subset of what you would see in a WIT world definition. The imports / exports are defined within a world.
                - Target: is the world that the component intends to target. It is optional, but does enable indexing. It also provides information for runtimes to hint at the component interface.
            - Any implementor should be able to inspect the WIT interface for a component and be able to construct a manifest and config.
            - The idea is that as a standard, everyone will be able to share a component back and forth.
            - Where would static files like Spin.toml live if not in a layer?
                - WASI-Virt or some other component as a virtualized filesystem.
                    - This would make everything a component inside the manifest. This would allow more interoperability in comparison to having multiple layer media types that some consumers will not understand how to handle.
                - (Luke) We could make each file a data segment and perhaps, make each data segment a layer which would enable radical deduplication of content.
            - Where should the specification live? CNCF? Bytecode Alliance?
    - PSAs
        - WasmCon is coming up!! Come join us!
    - New in Wasm
        - WASI-KeyValue is ready for implementation (Thank you, Kate, Taylor, Joe, and Luke). Probably working on blob and messaging next.

Action Items:

- (Taylor) Add agenda item to give a go/no-go for Wasm OCI Artifacts proposal.

April 2nd, 2024

Recording link: CNCF Wasm-wg Meeting 2024-04-02
Attendance: ✋ (PLEASE ADD YOURSELF)

- James Sturtevant
- Calvin Prewitt (JAF Labs)
- Daniel Macovei (JAF Labs)

Agenda: 📑

- OCI update
    - Getting really close to being done
    - Next week will likely be the last meeting and then bring it back the week after to the main Wasm WG
- Wasi-cloud update
    - Key-value updates: https://github.com/WebAssembly/wasi-keyvalue/pull/41

    - Runtime config: https://github.com/WebAssembly/wasi-runtime-config
- News
    - Now available in public beta, a Wasm Component registry, https://wa.dev; Uses the Warg protocol and soon OCI as well. Also, see the WasmIO talk: https://youtu.be/2_-10mRN30s?si=-u_B1WWsrcu8849M
    - Cloud native wasm day: https://www.youtube.com/playlist?list=PLj6h78yzYM2MQteKoXxICTWiUdZYEw6RI

Action Items:

March 5th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-03-05
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- Brandon Mitchell (independent)
- Jiaxiao Zhou (Microsoft)
- Taylor Thomas (Cosmonic)
- Danielle Lancashire (Fermyon)
- James Sturtevant (Microsoft)
- Calvin Prewitt (JAF Labs)
- Bailey Hayes (Cosmonic)
- Eric Gregory (Cosmonic)

Agenda: 🗒️

- Wasi-virt overview and demo (Guy Bedford)
    - A tool that allows you to virtualize away parts of the subsystem.
    - Currently this only supports import virtualization. It doesn't support server virtualization
    - Right now this virtualizes the whole world due to limitations in how streams are imported and exported. But that might be just fine
    - Recommendation: Highly recommend watching this meeting as notes were hard to take during discussion

Action Items: (none)
Notes:

- (Bailey) soon to have an event calendar for Bytecode Alliance: https://bytecodealliance.org/articles/february-stream

February 20th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-02-20
Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt (JAF Labs)
- James Sturtevant (Microsoft)

- Angel M Miguel (VMware / Broadcom)
- Brandon Mitchell (independent)

Agenda: 🗒️

- OCI Update - Started designing config.media type json schema: OCI + Wasm Meeting Notes
    - (James) Started to work on the config media-type. You can check about it on the attached doc
    - (James) Next week we have a new meeting to continue working on it (*you have a link to the meeting at the top of this document*)
- KubeCon prep (Taylor)
    - In person meetup?
        - (Taylor) Will work on this.
        - (Taylor) We can meet right after the co-located conference day (Tuesday).
            - (Danielle) There's a K8s-contrib fest at that time
            - (Taylor) I will send a message on the Wasm-WG channel to see if there's interest.
    - Talks to attend
        - (Taylor) Wasm + OCI lightning talk.
        - (Angel) Running LLMs efficiently in K8s (including a Wasm experiment)
- Wasm news (everyone)
    - (Taylor) Working on WASI Cloud core.
        - Plan to open a PR on WASI key-value with the Fermyon team.
    - (Danielle) Working on the WASI-observe standard.
        - I will let the people know on the channel when the PR is ready.
- About Wasm-WG meeting
    - (Taylor) We will skip the Wasm-WG meeting on 19th March due to KubeCon.

Action Items:

February 6th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-02-06

Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt (JAF Labs)
- Saiyam Pathak(Civo)
- Taylor Thomas (Cosmonic) (moderator)
- James Sturtevant (Microsoft)
- David Justice (Microsoft)
- Christoph Voigt (Liquid Reply)
- Angel M Miguel (Broadcom)
- Brandon Mitchell (independent)

Agenda: 🗒️

- Wasi-cloud interface overview (Taylor)
    - Notes:

- Worlds describe the interface provided by the component both host and guest.
- Aimed at hitting ~80% of features in a space (spaces: e.g. key/value, messaging, wasi-http, …)
- SQL is up in the air b/c it's REALLY hard to build a good abstraction over all the different types of SQL, like Oracle, SQLite, MSSQL, PG.
- WASI-HTTP:
    - Currently in phase 3
    - Luke Wagner gave a great description of the complexity in async / streaming during the Bytecode Alliance Plumbers Summit (Bytecode Alliance Plumber's Summit 2024 Day 2)
    - There are a couple imports from other WASI interfaces.
    - Two handlers are described incoming-handler, outgoing-handler
- WASI Runtime Config
    - The idea behind a runtime config is that it is different from your env vars. Often systems don't offer env vars. It is intended to be the lowest level interface to get configuration data. Only includes get for a key or get-all.
    - Folks have probably already used WASI env vars. We'll probably do something that will allow users to virtualize WASI env vars with runtime-config
- WASI Key / Value
    - This is the other one we are getting close on.
    - I'm going to go through the interface and then describe some current / on-going work.
    - The naming has gotten kind of complicated. We would like to simplify to provide eventual consistent by default, but could be better.
    - We are struggling with naming (open-bucket) doesn't really open a connection to a bucket…
    - Batch operations allow you to do many operations
    - Atomic includes compare and swap (CAS) as well as increment
    - The refactor changes top level functions into resources so you can have self dangling functions.
- WASI Cloud Core
    - We are not moving the world in entirety. We are moving each sub package along as they are ready.
- (Christoph) I have some questions about WASI-SQL.
- (Taylor) We'll cover that a little later.
- WASI Messaging
    - Messaging needs a little love.
    - Messages don't have a reply-to which is quite common in messaging brokers.

        - There is send and receive. The consumer is a bit different. You have to form subscriptions. I lean toward host defining subscriptions.
        - This one is very close as well. We'll need to work together with implementers to make sure the features / interfaces work.
    - WASI Blob Store
        - This one is going to be very similar to an S3 API. These are moving pretty quickly.
        - We have types like object ids, object metadata, etc.
        - I'm going to need help with stream APIs, but we need to review that and make sure we are supporting it properly.
        - This is the one where we NEED streams b/c we expect to have very large blobs.
    - WASI SQL
        - (Christoph) What would be the tradeoff of using WASI SQL if it was not part of WASI Cloud.
            - (Taylor) We want to say across environments that we support "WASI Cloud". We are not doing prepared statements
            - The SQL library in Golang is probably something we'd try to shoot for, but there are concerns in how to do this well.
            - If someone is motivated to translate the Golang SQL package into WIT, please do and open a PR.
            - I'm not sure there is a perfectly generic way of doing this.
        - (Christoph) Is there a train of thought about what gets into WASI or WASI Cloud Core?
            - (David) This was a topic we discussed at Bytecode Alliance Plumber's Summit 2024 Day 2. There was not a set of hard and fast rules, but rather a set of heuristics (specifications, many vendor implementations of some similar set of behaviors, does the domain fit into WASI, etc.)
    - WASI Observe
        - Not into WASI as of yet, but there has been some discussion
        - Can I do OTel stuff from the WASI side?
- OCI update (James, or Taylor if James isn't there)
- Update on whitepaper (Sven and Taylor)
    - This is going to be delayed until after Kubecon due to time needed to prep talks and update things to preview 2
- Conference talk schedules are live, what should we attend?
    - Wasm I/O in Barcelona
    - KubeCon EU Wasm Day in Paris
    - KubeCon EU

        - **CALL TO ACTION**: everyone in the group, please curate talks and we'll publish those out to folks.
        - (Christoph) Will the cloud-native rejects have Wasm content?
            - (Taylor) I'd imagine there's at least 1 there.
- Wasm news (everyone)
    - Preview 2 is live!!!! Woohoo!! wit@0.2.0
    - (Christoph) Kwasm, your favorite Wasm runtime installer. We kicked off version 2, which aims to be a generic containerd shim runtime installer. We have started a dashboard to organize work. You can currently install shims. We are currently tackling the lifecycle part (delete, update, installed, etc.). If you are interested, please come help us out!
        - https://github.com/orgs/KWasm/projects/3/views/1
        - https://github.com/KWasm/kwasm-operator/tree/feat-lifecycle-manager
        - CNCF Slack #kwasm
    - (Taylor) I think WasmEdge is getting there with their component model implementation.
    - (Taylor) If you are using WasmCloud, please give some feedback about preview 2 and wasmCloud 1.0

Action Items:

January 23rd, 2024

Recording link: (CNCF Wasm-wg Meeting 2024-01-23)
Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt (JAF Labs)
- Taylor Thomas (Cosmonic)
- David Justice (Microsoft)
- Danielle Lancashire (Fermyon)
- Christoph Voigt (Liquid Reply)
- Brandon Mitchell (Independent)
- Sohan Kunkerkar (Red Hat)

Agenda: 📑

- Update on white paper (Taylor)
    - Doc is in a new spot Wasm for Platform Engineers
- Update on OCI effort (Taylor)
    - Things are in progress and we have a weekly meeting on the off week from this meeting
- Wasm news (open floor)
    - Wasm imports for Go are starting to make progress
    - Please get your feedback in for wasi cloud!
    - The JCO reference implementation is done, so wasi preview 2 will be voted on this week
    - Cloud Native Wasm Day EU and Wasi I/O are coming up soon

Action Items:

January 9th, 2024

Recording link: CNCF Wasm-wg Meeting 2024-01-09
Attendance: ✋ (PLEASE ADD YOURSELF)

- Sven Pfennig (Liquid Reply)
- Calvin Prewitt (JAF Labs)
- Rajas Kakodkar (VMware)
- David Justice (Microsoft)
- Taylor Thomas (Cosmonic)
- Liam Randall (Cosmonic)
- James Sturtevant (Microsoft)
- Danielle Lancashire (Fermyon)
- Brandon Mitchell (Independent)
- Steve Sloka (Outerbase)
- Sajay Antony (Microsoft)
- Angel M Miguel (VMware / Broadcom)

Agenda:

- (Joe Zhou) Update on Wasi Cloud Core
    - Notes:
        - <begin Joe's presentation>
        - Short intro from @mossaka (Joe Zhou)
        - Shared doc: WASI Cloud Core Standardization
        - WASI and WASI preview 2. Many of the Wasm runtimes have WASI preview 1 support.
        - WASI preview 2 was stabilized at the end of last year, which included a move from WITX to WIT and component model
        - WASI preview 2 consist of 2 main sets of APIs, command line and http apps
        - Would like to standardize a WASI Cloud Core to add to the 2 existing
        - Core set of distributed application APIs
        - 2 major guiding principles
            - Portability: be able to ship app from 1 platform to another platform. Be able to move business logic from one place to another.
            - 80% target for interfaces: the interfaces are abstracted. They will not offer every feature for a given service. This includes key value, runtime config, locking, etc.
        - Many cloud native applications do not require sockets and other POSIX APIs. Instead you will see APIs like key value stores and HTTP
        - Each of the interfaces have been proposed to the W3C WASI sub group. Each sub interface of WASI Cloud Core is individually progressing in phased graduation.

        - WASI Cloud Core is an aggregate of each of the sub interfaces (KV, Blob, SQL, etc.)
        - Fermyon has a WASI HTTP implementation in Spin
        - </end joe's presentation>
        - How / when can folks add feedback
            - Would like to move them to phase 2 in the future
            - Can engage in the standardization process through the GH repo
            - See the "Elaboration of User Stories" in the WASI Cloud Core repo to add user stories / needs
- (James/Taylor) OCI + components effort
    - WASI OCI Design
    - Notes:
        - We are thinking about having folks join on the off weeks to discuss this topic further.
        - **Problem:** at this point, we have several different ways to put Wasm artifacts into OCI registries. We would like to standardize the way Wasm is stored to enable cross tool compatibility. We'd also like to put multiple components into an artifact so we can compose them together at runtime.
        - Created the doc that structures the proposal as well as a straw man.
        - Please dive into the doc and add comments / feedback.
        - Warg Development?
            - Lann M. is working on a prototype to focus on persisting artifacts in OCI registries. Calvin is looking at OCI implementations and coming up to speed with it. Looking at using OCI for publishing WASI preview 2 for interfaces. There are questions about the Warg protocol and how it will proceed. There are a lot of discussions happening.
            - There is a Dev Plumbers summit upcoming that should bring some answers.
- (Sven/Taylor) Proposed whitepaper structure
    - Notes:
        - Doc link:
        - What are the benefits and how do they align to the 7 attributes of the Plat Eng CNCF paper.
        - Enumerated list of cases / how you can run in production
        - It is a fast moving field and will likely need to be updated every 6-12 months.
        - I don't want to keep the feedback open too long (1-2 weeks), so we can get working on the paper. We'll bring it back after the draft for feedback, then publish.
        - Wanted to present for awareness.
- PSA(s)
    - (Liam) Some of the Wasm Plumbers Summit will be broadcasted. Baby steps due to budget, but we are trying to make it more accessible. Details soon.

Action Items:

Dec 26, 2023 (Canceled)

Dec 12, 2023

Recording link: CNCF Wasm-wg Meeting 2023-12-12
Attendance: ✋ (PLEASE ADD YOURSELF)

- Taylor Thomas (Cosmonic)
- Sven Pfennig (Liquid Reply)
- Angel M Miguel (Broadcom)
- Luke Wagner (Fastly)
- James Sturtevant (Microsoft)
- Liam Randall (Cosmonic)
- Ivan Font (Red Hat)
- David Justice (Microsoft)
- Christoph Voigt (Liquid Reply)
- Linda Zhou

Agenda: 📑

- Goals, ideas, and structure for Wasm + Platform white paper (Sven and Taylor)
    - Notes:
        - https://docs.google.com/document/d/1-h5hgr99WdqrMWGHiHKBIOepRzLtnaAmd0wBng-OyYA/edit?pli=1#heading=h.tkyjt02axff5
        - Who, What, When… Who might be obvious, but… Who do we think would be the target audience? Are there other audiences to consider?
        - (Angel) I think we should consider developers too. How they consume and get the benefits from Wasm
        - (Liam) It's gitops and where it meets the developer platform. Where those worlds come together.
        - (Sven) How about enterprise decision makers that think about establishing IDPs.
        - IDP: Internal Development Platform
        - (Taylor) CTOs, directors, et al. Any other target audiences? I think 3 is probably good.
        - (Christoph) What about security architects or related roles?
        - (Sven) I see them between the exec decision makers and the developers, but are not really determining an IDP.
        - (Taylor) I don't see security archs are the core audience. However, there is probably another paper there (in that topic area).
        - (Taylor) When do we want to have this done?
        - (Liam) Think about starting with "Why" (the book was mentioned). We should explain and be honest about the shortcomings of containers; what's the friction. When you design a platform, you are negotiating what

the responsibilities of the platform are and where the abstraction layer is established.
        - (Angel) Having a resource that could be shared would be useful.
        - (Taylor) What do people think about the structure? Is the goal to convince people to use Wasm?
        - (Liam) I think that is an ambitious goal. We should think about raising awareness. If we could get people to try this out, that would be cool.
        - (Angel) Having the mindshare is a good goal for the document.
        - (Taylor) What are the problems that we think Wasm solves?
            - Non-functional requirements and how the component model impacts designing software
            - Efficient resource usage
            - Developers spend their time on ops and maintenance; devs spend 80% of time there
            - Devs and SREs must know each others' jobs
            - The cost of "free" is really high. Do it in the platform or 5000 times in each app.
            - When I build a platform internally, will I expose the developers to the low-level abstraction / virtualization or will I provide a set of options.
        - (Sven) Points out the CNCF whitepaper. We should try to apply what we can do with Wasm on the platform (reduction of cognitive load, etc).
        - (Liam) I love pulling in the 7 platforms. Victor to come to your question about when container, VMs, Wasm. Perhaps, that should be another paper. Sven to your question around IDPs, I've had the pleasure / pain of building many IDPs over the years. We always found it helpful to target them as a single product. Lambda doesn't let you target directly firecracker. This may be more of an opinion than a rule.
        - (Taylor) Move to work async and delegate on doc. One more item I've been thinking about is what do people use to deploy to production. What tools / projects.
    - Notes (next whitepaper)
        - (David) What do we think about OCI / registry / runtime whitepaper?
        - (Taylor) If we were to say one way to integrate with Wasm, then we may have too much bias or we may end up with an ever updating list of how you integrate with containers.
        - (Liam) New paper idea: "When Wasm? When Containers?"
        - (Christoph) I like the observability model. It shifts the way you use K8s and there is a whole area that needs to be investigated and how we'll work with it in the future.
        - (Taylor) I was looking back at deliverables and scope. I'm wondering if some of these things would be best done in vertical architectures. Here are 3 different ways you can run Wasm in K8s. Here's some examples. What are the challenges you will run into in prod.

    - Notes (wasi-cloud-core)
        - Example: https://github.com/WebAssembly/wasi-keyvalue ; https://github.com/WebAssembly/wasi-cloud-core
        - (Taylor) Now that we are past Preview 2, we should all take a look at the next set of 80% interfaces. One that I'm working on is: https://github.com/WebAssembly/wasi-runtime-config/pull/4. This is a way to fetch config vars / secrets and would want to validate them the folks here. Mostly PSA for folks to start taking a look at this stuff. If you have concerns of discussion, please drop them in the Slack channel. We should definitely have an opinion on these things.
        - (Angel) Wrt to Wasm Worker Server, we expect to be more active with components and will be active in the future.
        - (Taylor) Now is the time to comment to get the proper use cases covered. When we get to KubeCon EU, I want to have components that will run on all of our things. It's going to be the game changer. It's not going to be the killer app. It's going to be the killer use case. I just saw thing thing run in K8s, outside, around, and it's the same code 🤪.
    - PSA:
        - We are close to announcing the Bytecode Alliance plumbers meeting.
        - WasmCon 2024 should be up soon.
        - VOTE for Bytecode Alliance: voting is still open.

Action Items:

Nov 28, 2023

Recording link: CNCF Wasm-wg Meeting 2023-11-28
Attendance: ✋ (PLEASE ADD YOURSELF)

- Calvin Prewitt (JAF)
- Danny Macovei (JAF)
- Liam Randall (Cosmonic)
- David Justice (Microsoft)
- Ivan Font (Red Hat)
- Angel M Miguel (VMware / Broadcom)
- James Sturtevant (Microsoft)
- Joonas Bergius (Cosmonic)
- Saiyam Pathak (Civo)
- Shivay Lamba (Meilisearch)
- Brooks Townsend (Cosmonic)
- Sven Pfenng (Liquid Reply)
- Christoph Voigt (Liquid Reply)
- Linda Zhou

Agenda: 📑

- Registries
    - What are the happenings with Warg? (Calvin and Danny?)

        - (Calvin) Warg is an open source protocol to build Wasm registries. The main design decisions are:
            - Libraries and interfaces can be published.
            - Federated.
            - Build on top of existing resources like OCI / Blob stores.
        - (C) New releases are published to append-only logs, protecting the registry against common Software Supply Chain attacks and providing verification capabilities.
        - (C) Current status of OSS development:
            - Minimal reference server implementation.
            - Warg CLI and rust libraries to interact with the protocol.
            - Protocol specification.
        - (C) Introducing the current API endpoints:
            - Checkpoints allow third-party clients to verify the status of the entire registry in a specific point of time.
            - Several API endpoints focus on verification, ensuring that clients validate all the package information when retrieving them.
        - (Sven) How can you delete / forget a package and how it affects storage?
            - (Calvin) Logs will be there, so you cannot remove the package information. However, when a user tries to download a removed package, the content-bytes could be not available in the storage.
        - (Danny) We are in Zulip for any question or comment. Feel free to reach out there.
        - Links:
            - Warg Registry repo: https://github.com/bytecodealliance/registry/
            - Warg OpenAPI spec: https://redocly.github.io/redoc/?url=https://raw.githubusercontent.com/bytecodealliance/registry/main/crates/server/openapi.yaml
            - Warg Zulip stream: https://bytecodealliance.zulipchat.com/#narrow/stream/352111-warg
    - What is up in OCI + wasm? (James, Jorge with additional commentary)
        - https://docs.google.com/document/d/1l5Or2Cp4QTnXTIIOCAOZ8TxQb6FH-iwUQbqOA8PvXf0
        - (James) Initially, we bundled Wasm components inside regular container images. Then, we started moving to specific OCI packages so other projects can consume them in different platforms.
        - (J) The OCI spec includes multiple layers. Currently, we have a config.mediaType section which it's required for Kubernetes. On the layers side, we have multiple ones including the different Wasm components.
            - Please, refer to the Wasm and OCI next steps document to check those.

        - (J) For Spin (Fermyon), it includes specific layers that contain the application configuration. In addition to that, the OCI types are different in these cases.
        - (Jorge Prendes) To clarify it, the file layers are oriented to Linux systems (OS specifics). So, adding arbitrary files to the OCI specification may break the compatibility with specific runtimes.
            - (JP) There's a WASI-runtime-config specification that looks for consolidating the different configuration files (like Spin.toml). It's still under discussion, but it's a spec that will simplify all these issues.
        - (J) Some future developments / improvements:
            - (J) Work on the tooling integrating this spec in existing tools (Docker) or new ones (CLIs, Warg, etc).
        - (J) Looking for feedback! Feel free to comment on the document.
        - (David Justice) Could you comment about Wasm precompilation in the OCI spec?
            - (J) Some runtimes come with precompilation features like Wasmtime and WasmEdge. It prepares the module to run directly in the given architecture and platform.
            - (JP) Wizer works in a different way by running a Wasm module until it reaches a non-deterministic step and creates a snapshot as another Wasm module. This new module is not arch / platform specific and can run in any environment.
        - (Ivan) How do these precompiled vs non-precompiled modules work across nodes in a cluster environment?
            - (J) The OCI artifacts are totally portable between nodes. They can download the different resources and prepare based on the node specifics.
- WasmEdge update on component model work
    - (Michael) WasmEdge is actively working on supporting the component-model proposal.
        - (M) The first step was to support the new elements in the Wasm specification to parse component modules.
        - (M) Then, we are working on supporting multiple WASI preview2 like WASI-NN, WASI-sockets, etc. There are some WasmEdge specific details as many of these proposals are implemented as plugins. Users can download WasmEdge CLI with the specific plugins they need.
        - (M) There are different challenges with the binary format and WasmEdge. To overcome this, we started with the subset of the component-model binary specifics to support the WASI preview2.
            - (M) Our goal is to present component-model support early next year (for KubeCon EU or Wasm IO).
            - (M) One of the first components we are targeting is the hello-wasi-http example.

        - (David Justice) Is WasmEdge the second runtime working on the Component-Model?
            - (Michael) Yes. Before it was difficult to start integrating it as the binary format was changing and there isn't much documentation. That changed and know there's better communication so we started working on it.
            - (Luke) This is the main reason why preview2 is marked as stable, as preview3 should work on the same context. It's a good time to implement the binary format.
- (David Justice) Luke, could you comment on ABI stability?
    - (Luke) In 2 weeks we plan to have a final meeting to decide what to include in the component-model repository, so the versions are stable. Those will continue growing while keeping compatibility with previous versions by leveraging adapters.

Action Items:

Nov 14, 2023

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- Ivan Font (Red Hat)
- Sven Pfennig (Liquid Reply)
- Taylor Thomas (Cosmonic)
- Calvin Prewitt (JAF)
- Linda Zhou

Agenda: 📑

- KubeCon NA takeaways?
    - One idea for our first white paper come the new year is something along the lines with “Wasm from the Point of View of a Platform Engineer” (obviously we need a better title)
    - (T) It feels like we have moved past someone asking what a “Wasm” is, and now folks know what Wasm is and want to build things. (not at Wasm day)
        - Many folks didn’t know about the component model. It was an “Oh my gosh” moment.
        - There was a definite switch.
    - (Sven) Who is interested in Wasm for platform engineers?
        - Taylor, Jorge, David raised their hands
    - (T) Perhaps, we could have a talk about OCI + Wasm Registries update.
        - Would we also like to discuss start in a whitepaper
- "Wasm for Platform Engineering" whitepaper proposal:
    - Document to drop ideas for the whitepaper. We will review them in December.
- ~~Wamr?~~
- ~~WasmEdge update on component model work~~

Action Items:

Oct 31, 2023

Recording link: CNCF Wasm-wg Meeting 2023-10-31
Attendance: ✋ (PLEASE ADD YOURSELF)

- Taylor Thomas
- Calvin Prewitt
- David Justice (Microsoft)
- Luke Wagner
- Jorge Prendes
- James Sturtevant
- Linda Zhou

Agenda: 📑

- Project introduction:
    - (Prendes, Goff, Justice) Runwasi (https://github.com/containerd/runwasi)
        - Notes:
            - Runwasi is a containerd project that allows you to run wasm workloads
            - Want to create a way to run your image without packing in the runtime (hence why there are shims)
            - Compared to native containers, they are much smaller.
            - The shim API is really small/thin because it is all about starting, stopping, and managing tasks
            - Why do we do this? Let's you run wasm side by side with normal containers in places like Kubernetes
            - Example: A k8s deployment using a wasm runtime class
            - A related project are the aforementioned containerd wasm shims: https://github.com/deislabs/containerd-wasm-shims
            - They only want to include the wasm binary and not the runtime so that it preserves the ability to run anywhere (including Windows!)
            - This approach also means you have to break out of two sandboxes
            - Questions:
                - Luke: What is the part of the RuntimeClass that actually "links" it to wasm?
                    - Yes, it is the name that gets referenced by the deployment
                - Luke: I saw how AKS did this before, how does it work with these runtime classes?
                    - This is where kwasm comes in, which is a controller that knows how to install and deploy the shim and runtime class
                    - https://kwasm.sh/
                - Taylor: How do you run this stuff if you aren't a k8s admin?

        - There aren't any discussions or answers on this yet, will talk in the slack channel
    - Taylor: When should you use this vs not using this?
        - Docker uses this to avoid platform dependence
        - It is also good for stuff they already have deployed
    - Luke: Could the RuntimeClass just be the interface name (like http-proxy) and then the operator decides how it is provided?
        - Right now the wasm binary is put in the rootfs. We want to move away more to a platform based model that allows you to ask for or enumerate the worlds you need from your runtime. Containerd could select the runtime from the image
- Component model overview (whole group, started by Taylor)
    - Notes:
        - Intention is to get a shared context for KC NA about the state of the component model.
        - https://bytecodealliance.org/articles/webassembly-the-updated-roadmap-for-developers ← Bailey has been driving
            - 3 tiered roadmap
                - WASI core of the standard model
                - Component Model (resources, handles)
                - WASI as the collection of standard interfaces (Worlds)
            - Rough date for preview 2 is December'ish
            - Need 2 implementations in both wasmtime and JCO (https://github.com/bytecodealliance/jco)
            - 1 key point that a lot didn't grok; we called preview2 an island of stability. Really, it's a window. Everything that builds off of the stable ABI can be added to preview2. https://github.com/WebAssembly/WASI/blob/main/preview2/README.md#proposal-requirements-for-inclusion-in-wasi-preview-2
            - Just b/c WASI cloud is not part of preview2 doesn't mean that it can't be added after preview2 landing due to the stability of the shared interfaces.
            -
        - Currently voting on renaming WASI to WebAssembly Standard Interfaces
        - Preview2 will not land for Kubecon. However, it will land late Nov. or Dec.
        - Excited for preview2 to stabilize
        - Wasmtime is going to contain http-server support?
            - Wasmtime 14 has resources
        - BCA community stream is tonight 7pm ET. We will be talking about Wasmtime 14 and 15. We are expecting wasmtime 15 to point to all the upstream WASI org interfaces (WIT IDL files).
        - The WASI HTTP work landed for 15.

        - If you are interested in the work needed to land preview2 of WASI take a look at the project board (link to github)
        - Highly recommend you try this stuff out in case you run into things (we have). Please give the feedback upstream so they can get fixed. It gets harder to make breaking changes as we strive to provide stability for the core interfaces.
        - Wasm Social Sunday before KC NA: https://www.eventbrite.com/e/wasm-social-tickets-736290965097
- (covered in Oct 17th meeting) ~~Your guide to wasm talks during KubeCon (David)~~

Action Items:

- Bailey going to reach out to wamr to see if they'd be willing to talk at next meeting
- (Jorge) Reaching out to Michael Yuan (WasmEdge)

Oct 17, 2023

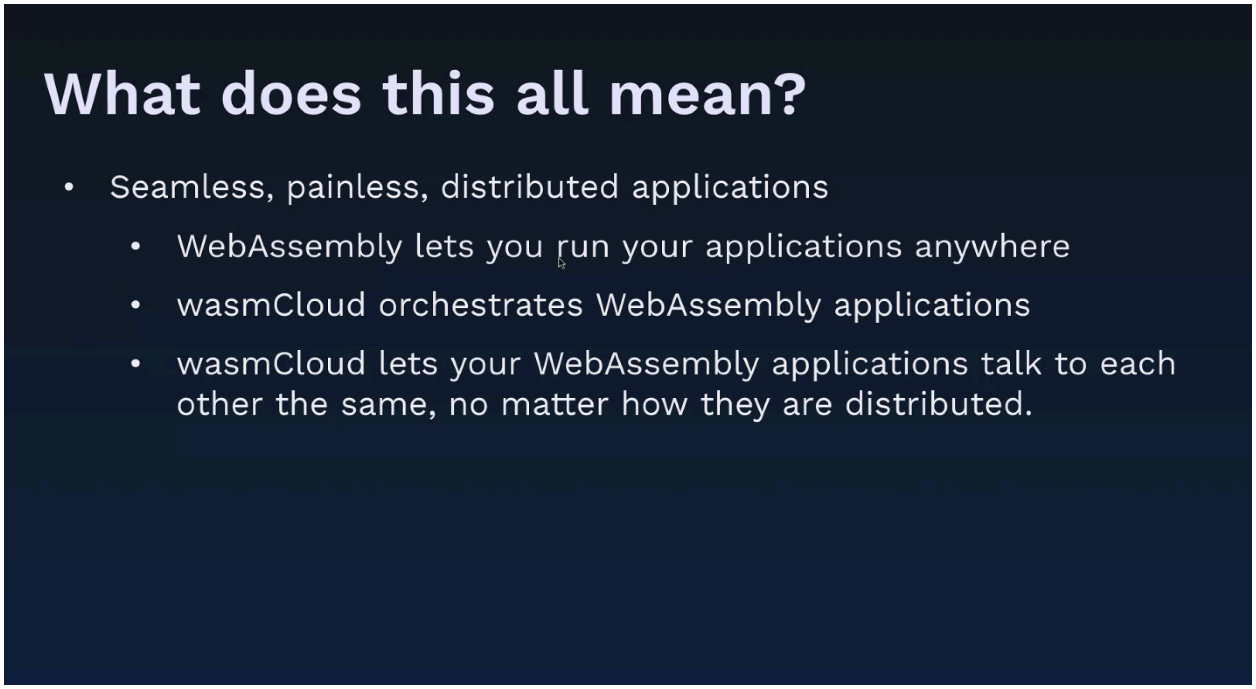
Recording link: CNCF Wasm-wg Meeting 2023-10-17
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- Ivan Font (Red Hat)
- Rafael Fernández López (VMware)
- Angel M Miguel (VMware)
- Calvin Prewitt (JAF)
- Danny Macovei (JAF)
- Sven Pfennig (Liquid Reply)
- Ralph Squillace (Microsoft)
- Taylor Thomas (Cosmonic)
- Danielle Lancashire (Fermyon)
- Jorge Prendes (Docker)
- Hung-Ying Tai (Second State)
- Linda Zhou

Agenda: 🗒️

- Project introduction:
    - WasmCloud (Taylor)
        - Notes:
            - WasmCloud is a CNCF project. It falls in the Application Framework category (Wasm Landscape).
            - It's a wasm orchestrator, simplifying the deployment and connection between the applications you deployed. It's platform agnostic, so you can deploy applications in multiple environments.
            - It relies on multiple CNCF projects like OCI, NATS, OpenTelemetry, etc.
            - Demo time!
                - Actors are Wasm components.

        - Currently, some parts rely on WasmCloud specific libraries. However, the end goal is to remove all the specific libraries and make applications compatible between projects and platforms.
        - In the first example, the application uses WASI types instead of WasmCloud specifics.
        - The application configuration defines the different services to connect and the required Wasm APIs (via Component-Model).
        - In the demo, Taylor showed how to compose an application and connect different providers in the Cosmonic UI (based on WasmCloud internally).
            - The application uses a KV store but it doesn't know which one. You can swap the provider to a different datastore and keep using the same code.
    - Summary slide!



- Wasm Workers Server (Rafael)
    - Notes:
        - Wasm Workers Server focuses on developers, allowing them to build applications in different languages. It's compatible with different platforms and frameworks.
        - You can find all the documentation and many different examples in the project repository.
            - https://workers.wasmlabs.dev/

        - https://github.com/vmware-labs/wasm-workers-server/tree/main/examples
    - Supported languages today: Ruby, Rust, JavaScript, Python, Go and Zig.
        - The interpreters like Ruby and Python come for the WebAssembly Language Runtimes OSS project.
    - The Wasm Workers Server provides multiple features for the applications like Static assets, routing, etc.
    - At worker / function level, it provides multiple features:
        - Environment variables
        - Folders
        - HTTP outgoing requests
        - Etc.
    - For the future, there are multiple ideas in the roadmap:
        - Improve host / guest communication and improve component model support.
        - Support more WASI contracts / APIs to increase the compatibility.
    - Demo!
        - You can run an application directly from an external git repository or from the local filesystem.
        - The JavaScript workers are compatible with other platforms like Cloudflare Workers.
        - You can also provide more complex applications composed by multiple workers in the given folder.
- KubeCon Talks on Wasm
    - Phippy's Field Guide to Wasm - Karen Chu & Matt Butcher, Fermyon
        - https://sched.co/1R2ri
    - eBPF + Wasm: Lightweight Observability on Steroids - Michael Yuan, WasmEdge & Yusheng Zheng, eunomia-bpf
        - https://sched.co/1R2uf
    - Bringing Cloud Native WASM to the mainstream with WASM Working Group - Shivay Lamba, Meilisearch & Kevin Hoffman, Cosmonic
        - https://sched.co/1RQZf
    - Tutorial: Building Cloud-Native Applications Using WebAssembly and Containers - Mikkel Mørk Hegnhøj & Melissa Klein, Fermyon; Ralph Squillace, Microsoft
        - https://sched.co/1R2mE
    - Runwasi: WebAssembly Serverless for Containerd - Angel M De Miguel Meana, VMware & Francisco Cabrera, Microsoft
        - https://sched.co/1R2nF
    - Delivering Backends Like Frontends with WebAssembly - Brooks Townsend, Cosmonic
        - https://sched.co/1R2r8
- New and Notable in Wasm!

    - Warg (Calvin)
    - Resources (Taylor)
    - W3C Wasm Community Group (Angel)

Action Items:

Oct 3rd, 2023

Recording link: CNCF Wasm-wg Meeting 2023-09-07
Attendance: ✋ (PLEASE ADD YOURSELF)

- Angel M Miguel (VMware)
- Taylor Thomas (Cosmonic)
- Jorge Prendes (Docker)
- David Justice (Microsoft)
- Linda Zhou

Agenda: 📑

- How can we help people learn about the different projects?
    - Let's bring in the different runtimes and application runtime projects to talk about what they can do
- Should we have a components call? Yes
- What is new in the world of wasm?
    - There are some Machine Learning runtimes that can actually compile to Wasm
        - Spin and WasmEdge for example have already added some support for llama.cpp
    - Component model has landed resources and there are some wit syntax changes
    - People are starting to notice wasm in academia: https://www.cs.cmu.edu/news/2023/webassembly-research-center
    - https://dev.to/thangchung/how-to-run-webassemblywasi-application-spin-with-dapr-on-kubernetes-2b8n running dapr running side by side with wasm
    - Cloud Native Wasm Day is coming: https://events.linuxfoundation.org/kubecon-cloudnativecon-north-america/co-located-events/cloud-native-wasm-day/ sign up!
- White paper
    - The emerging technologies in the community
    - Warg, OCI, and end to end flow with K8s / containerd
        - We need to search for a given interface implementation in the development registry. This should probably not be in an OCI registry.

Action Items:

- Discussed topics scheduled for the next few meetings
- Follow up with wamr to see if they could present (David)
- We want to try starting each meeting with 5-10 minutes of "news" where we can talk about new things that have released and other relevant news

Sept 19th, 2023 (Canceled - no agenda items to cover)

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 🗒️
Action Items:

Sept 7th (8am @ WasmCon), 2023

Recording link: CNCF Wasm-wg Meeting 2023-09-07
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (Microsoft)
- Christoph Voigt (Liquid Reply)
- Angel M Miguel (VMWare)
- Jorge Prendes (Docker)
- Linda Zhou

Agenda: 🗒️
Action Items:

August 22nd, 2023

Recording link: CNCF Wasm-wg Meeting 2023-08-22
Attendance: ✋ (PLEASE ADD YOURSELF)

- Angel M Miguel (VMware)
- Christoph Voigt (Liquid Reply)
- Shivay Lamba
- David Justice (Microsoft)
- Sven Pfennig (Liquid Reply)
- Linda Zhou

Agenda: 🗒️

- PSAs:
    - WasmCon Sept. 6th - 7th in Bellevue, WA
    - Notes:
        - (David) Just want to highlight the conference and be sure that everyone is aware of it. You will find a lot of interesting talks and people about Wasm, components, AI, etc.
        - (Bailey) The "Componetize the World" hackathon will be right after WasmCon!
- (David Justice) containerd shim lifecycle manager proposal: https://hackmd.io/@devigned/SJJPWsAj2
    - Notes:

        - (David) Installing containerd shims is a complex and error-prone process. You need to install the binaries, update the configuration, etc. This new proposal discusses a new containerd-shim lifecycle to simplify all these operations.
        - (David) Baking new images to install shims is a costly operation. The Wasm ecosystem is growing rapidly, so we have new versions almost every week.
        - (David) The proposed CRD allows you to define the location of the shims, how to pull them, the runtime class, how to deploy them and where (node selectors).
        - (David) The installation process must ensure the node always ends up with Containerd in a running state. If it fails, it should rollback to the latest version.
        - (Christoph) The installation life cycle requires restarting Containerd. How does the controller ensure Containerd is running? Containerd cannot run if the configuration is malformed or incorrect.
            - (David) There are different approaches we can take here. The controller can backup the current configuration before updating it. If something fails, it can rollback the configuration.
            - (Christoph) If this happens, the rollout will stop, won't it?
            - (David) Yes!
        - (Luke) Based on the current status of KWasm (an operator that installs containerd-shims in the nodes), is this proposal an extension or a high level abstraction?
            - (David) This is a more opinionated extension of KWasm. It allows you to install any shim or group of shims in your cluster. With KWasm, the group of specific shims is tight to the version you're installing.
        - (Luke) Can I install this in a vanilla cluster or does it require any extra configuration?
            - (David) It should work! We need to add more details to the proposal like the node architecture.
            - (Sven) There's a demo about KWasm in which we show how to install it in a vanilla cluster.
        - (David) Note that this is a generic CRD, so anyone can add new shims to Containerd. This may open the possibility of upstreaming it to Containerd.
            - (Angel) I'm pretty excited about this. Shims have been around for a long time, although Wasm was one of the first ecosystems to take full advantage of this feature. I believe that a generic controller may open new use cases in different ecosystems.
- (Chris Aniszczyk): Review upcoming Wasm Landscape l.cncf.io/wasm
    - Notes:

- (Angel) Some projects may be present in two places, like Spin, WasmCloud and Wasm Workers Server. These projects may be placed in both Bare metal and application frameworks. Shall we duplicate?
    - (Christoph / David) We can leave this for maintainers. They will decide the feature they want to highlight.
- (Luke / David) Edge is a fuzzy word as it means different things for different people. We may need to clarify it.
- (David) In the language section, the languages have different support level. Some of them are fully compatible, while others only have partial support. Shall we add any clarification here?
    - (Victor) For a newcomer, showing them is useful to understand the different languages that you may need.
    - (Luke) We may define the compatibility in terms of API support / components.
- (Angel) There are several hosted platforms that offer Wasm capabilities because they're using runtimes like V8 under the hood, like Cloudflare. Shall we add them?
    - (Christoph) At this point it's easy to add them as there are only a few platforms. However, we may need to revisit if the list grows.
    - (Angel) We can add the ones that promoted the Wasm use case.If other
- (Jorge) Shall we add a new card for containerd-shims that targets Wasm? Shall we include a new card for products using Wasm?
    - (Angel) +1 on this. I believe it's good to see products that use Wasm as a foundational technology. +1 for the shims too.

Action Items:

August 8th, 2023

Recording link: CNCF Wasm-wg Meeting 2023-08-08
Attendance: ✋ (PLEASE ADD YOURSELF)

- David Justice (msft)
- Christoph Voigt (Liquid Reply)
- Saiyam Pathak(Civo)
- Jorge Prendes (Docker)
- Sven Pfennig (Liquid Reply)
- Rajas Kakodkar (VMware)
- Bailey Hayes (Cosmonic)
- Hung-Ying Tai(WasmEdge)
- Angel M Miguel (VMware)
- Nikhita Raghunath (VMware)
- Alan Poon
- Chris Aniszczyk (CNCF/LF)

- Linda Zhou

Agenda: 📑

- (Alan Poon) Wasm Mock Server: sandbox proposal https://github.com/cncf/sandbox/issues/50
    - Notes:
        - First project to offer man in the middle (MITM) testing / automation
        - Why use Wasm, it can be compiled via many languages, import standard libs
        - Mock, reply, and automation covered today
        - Show the mock server web interface, starts the mock, loads a browser in browser, and demonstrates mocking the inner browser request
        - Demonstrates mock server web interface replayer for mocking a web site, records a report of the traffic with expectations and response.
        - Demonstrates automation
- (Saiyam Pathak & Sven Pfennig) Intro to KWasm https://KWasm.sh

Action Items:

July 25th, 2023

Recording link: https://youtu.be/1VsoJ2hBLBk

Attendance: ✋ (PLEASE ADD YOURSELF)

- Jorge Prendes (Docker)
- David Justice (msft)
- Ashutosh (Elastic)
- Rajas Kakokdar (VMware)
- Karen Chu (Fermyon)
- Luke Wagner
- Daniel Krook (CNCF)
- Michael Yuan (WasmEdge)
- Nabarun Pal (VMware)
- Vivian Hu (WasmEdge)
- Till Schneidereit (Fermyon)
- Danielle Lancashire (Fermyon)
- Saiyam Pathak (Civo)
- Sven Pfennig (Liquid Reply)
- Linda Zhou

Agenda: 📑

- Rajas and Ashutosh presenting a k8s scheduler extension
- Introducing WasmEdge
    - The WasmEdge incubation proposal: https://github.com/cncf/toc/pull/1102
- Till introducing the BCA
- [Tentative] KWASM

Action Items:

- Kick-off discussion about K8s related worlds to describe different plugin interfaces like K8s scheduler, Admission controller, …
- Think of an objective way to compare Wasm runtimes

July 11th, 2023 (Canceled - Scheduling constraints with planned presenters)

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 🗒️

-

Action Items:

July 4th, 2023 (Canceled - Public Holiday in the USA)

Recording link: (To be added…)
Attendance: ✋ (PLEASE ADD YOURSELF)

-

Agenda: 🗒️
Action Items:

-

June 27th, 2023

Recording link: CNCF Wasm-wg Meeting 06-27-2023
Attendance: ✋ (PLEASE ADD YOURSELF)

- Sven Pfennig (Host)
- David Justice (Notes)
- Rajas Kakodkar
- Angel M Miguel
- Christoph Voigt
- Jorge Prendes
- James Sturtevant
- Jiaxiao (Joe) Zhou
- Jesús González
- Linda Zhou

Agenda: 🗒️

- Update CNCF Wasm Landscape: https://landscape.cncf.io/wasm
    - Notes… (varying levels of accuracy)

    - (Sven) The landscape is a bit outdated. It is a living document and everyone can add the projects they are working on. Please consider updating it in the upcoming weeks. For example, runwasi would be a good thing to add. If you have any ideas, please reach out to me or create a PR.
    - (Angel) Please share how to add items to the landscape
    - (Sven) Will add it in Slack
    - Initial Github isse of WASM Landscape: https://github.com/cncf/landscape/issues/2387
- [jstur] Wasm and OCI
    - Media types and look at artifact used in https://github.com/containerd/containerd/pull/8699
    - Notes…
    - (jstur) OCI artifacts is a topic that spans multiple projects in the CNCF and BCA so would be a good topic for this group. (Shows a document with details about image layers and manifests w/ prototypes). Today, I'd like to show you what one of these artifacts looks like. In this PR I've pushed up some examples of what this media type layer might look like. This may become more apparent after I show a module.

      I've already build the modules and loaded them into a registry. I'm going to regctl to show what the image manifest looks like. This is based on the OCI artifact guidelines. We are currently using the image config rather than a custom manifest config. This has served well, but we may need to change the configuration media type if we need to. This one being shown is a single module. There is 1 layer with "application/vnd.w3c.wasm.module.v1+wasm" media type. This layer is pulled and used to boot the wasm module in a containerd shim.

      (showing ctr run for the artifact and runs "hello" artifact wasm)...

      (showing regctl to show multiple components in a single artifact)
      This example shows multiple components and a module configuration that tells the shim how to compose the components together.

      The main thing here is we need to decide what media types to use. There has been discussion about it, but it's not settled.
    - (justice) Can you talk about components and the config tying them together?
    - (jstur) (shows Wasm compose tool and the example repository illustrating how to compose multiple components together)

      You can inject components into a service. Runwasi is taking two components, stitching them together, and then running them in the shim.
    - (Sven) This is awesome. It is getting closer to the idea of reusable components. Is this currently in the main branch of runwasi.
      (jstur) No. It is not in runwasi yet. This is a prototype just yet. I didn't want to

spend too much time on the component side of things, b/c it is in so much flux. The focus is more on how we can use OCI artifacts.
    - (Angel) I agree it's looking really nice. Great to see projects working together. I have a comment on the configuration file. I understood that Wasm compose is temporary until support for component binding is more mature. My recommendation, it might be good to standardize on something more generic than the specific compose tool.
    - (jstur) That's great feedback. One thing I'll mention is that David is working on the Warg registry, and we are trying to align these projects so that someone can push to Warg and use via containerd.

Action Items:

- Add runwasi to the CNCF landscape.

June 20th, 2023

Attendance: ✋ (PLEASE ADD YOURSELF)

- Danielle Lancashire
- Angel De Miguel
- Sven Pfennig
- Jesus Gonzalez
- Asen Alexandrov
- Victor Lu
- David Justice
- Linda Zhou

Agenda: 📑

- Review wg-wasm comments one by one
    - "end users" (accepted)
        - (Sven): accept; we are not just focused on developers
    - "How do we empower Wasm…"
        - (Angel + Danielle): closed
    - "Wasm space…"
        - (Angel): closed
    - "Documentation and Examples for CNCF projects that wish to integrate with WebAssembly runtimes"
        - (Danielle): By covering folks who are using CNCF projects, we are not helping project maintainers to integrate Wasm into their projects. I think yes, but my opinion is not strong.
        - (Angel): I think we can cover this as part of the vertical use cases.
        - (Danielle): sgtm
        - (Angel): accept and close
    - Krustlet
        - (Angel): should we keep it?

        - (Sven): It was a good early project. Perhaps, it should move to archive. It is something interesting to look at. I would discuss with maintainers before. I don't think we should decide something like that. For us, we can keep it as something we are interested in.
        - (Danielle): Good example of prior art.
        - (Angel): We'll keep it. Resolved.
    - TAG-Runtime mailing list
        - (Angel): I think it's good for visibility and to not have folks needing to join multiple lists.
    - (Angel): I think we are good to go. I will send a final suggestion on Thurs or Friday(Sven): perhaps, Krustlet
    - (group): Kubewarden, OPA, Dapr, BCA, API machinery, SIG-Arch, etc
    - (Angel): it would be useful to me to hear from Wasm focused projects initially to help me to map the two worlds. It would be useful to have folks from other perspectives.
    - (Victor): It would be good to get folks who have experience with what was successful / not successful in OpenStack and others.
    -