

WASI: WebAssembly System Interface

WASI je modulární systémové rozhraní pro [WebAssembly](#). Jeho cílem je vytvořit standard, který umožní spouštět binární moduly Wasm na jakémkoliv operačním systému nebo cloudu s nativním výkonem a vysokou bezpečností.

Solomon Hykes (spoluzakladatel Dockeru) o něm prohlásil: „*Kdyby v roce 2008 existovalo WASI, nemuseli bychom vymýšlet Docker.*“

1. Proč WASI vzniklo?

WebAssembly je skvělý engine pro výpočty, ale sám o sobě nemá přístup k počítači (nemůže otevřít soubor, vypsát text do konzole nebo poslat data po síti).

- V prohlížeči tyto funkce zajišťuje **JavaScript**.
- Mimo prohlížeč (na serveru) však JavaScript chybí, a proto bylo nutné vytvořit rozhraní, které by tyto systémové volání definovalo bezpečně a univerzálně.

—

2. Klíčové vlastnosti WASI

A. Portabilita (Write Once, Run Anywhere)

Kód zkompileovaný do Wasm s podporou WASI poběží stejně na Windows, Linuxu i macOS bez nutnosti úpravy zdrojového kódu. Stačí mít nainstalovaný tzv. **Wasm Runtime** (např. Wasmtime nebo Wasmer).

B. Bezpečnost založená na oprávněních (Capability-based Security)

WASI používá velmi striktní bezpečnostní model. Na rozdíl od běžných aplikací, které mají přístup ke všemu, co jim dovolí uživatel, modul WASI nemá přístup k **ničemu**, co mu není výslovně předáno.

- **Příklad:** Pokud chcete, aby aplikace WASI přečetla soubor `data.txt`, musíte jí při spuštění předat konkrétní popisovač (file descriptor) tohoto souboru. Aplikace „nevidí“ zbytek vašeho disku.

[Image showing WASI security model: User grants specific file capabilities to Wasm Sandbox]

—

3. Jak WASI funguje?

WASI definuje sadu systémových funkcí (např. `fd_read`, `path_open`), které jsou podobné standardu **POSIX** (používaném v Linuxu/Unixu).

1. Vývojář napíše kód (např. v `[[it:sw:rust_programming|Rustu]]`).
2. Kompilátor (např. LLVM) přeloží kód do Wasm a mapuje systémové volání na standard **WASI**.
3. **Wasm Runtime** na cílovém stroji přeloží tato volání na skutečné instrukce operačního systému.

4. WASI vs. Docker

WASI je často srovnáváno s kontejnery, ale mají odlišné vlastnosti:

Vlastnost	Docker (Kontejnery)	WASI (Wasm)
Rychlost startu	Sekundy / stovky milisekund	Mikrosekundy (téměř okamžitě)
Velikost	Stovky MB (celý OS image)	Kilobajty / jednotky MB
Izolace	Úroveň procesů OS	Úroveň instrukcí v sandboxu
Výkon	Nativní	Téměř nativní (JIT kompilace)

5. Praktické využití

- **Edge Computing:** Rychlé spouštění malých funkcí v CDN (např. Cloudflare Workers).
- **Plugins:** Aplikace mohou umožnit uživatelům psát pluginy v jakémkoliv jazyce (Rust, Go, C++), které pak bezpečně spustí přes WASI.
- **Cloud Native:** Budoucnost mikroslužeb, které jsou lehčí a bezpečnější než tradiční kontejnery.
- **Blockchain:** Mnoho blockchainů používá WASI pro své runtime prostředí, aby zajistily deterministický běh kódu na všech uzlech.

Související články:

- [WebAssembly: Základy technologie](#)
- [Rust: Ideální jazyk pro WASI](#)
- [Docker: Porovnání s kontejnery](#)

Tagy: *it wasm wasi cloud-native serverless rust system-programming*

From:

<https://serviceit.cz/> - **IT ENCYKLOPEDIE**

Permanent link:

<https://serviceit.cz/doku.php?id=wasi>

Last update: **2026/01/02 20:23**

