Fraunhofer IESE

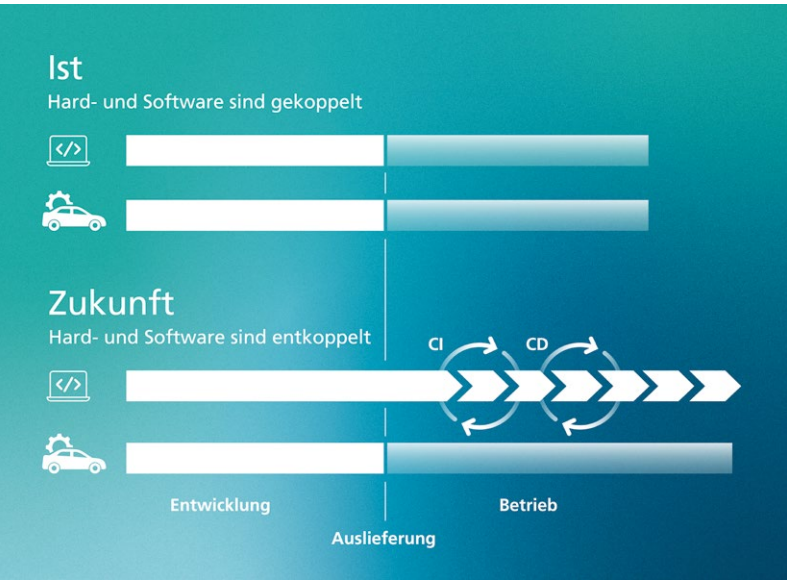
WebAssembly als Hardware-Abstraction-Layer

WebAssembly in Software Defined Defense

www.iese.fraunhofer.de

Software Defined Defence: Entkopplung von Software und Hardware

Das Hauptversprechen von softwaredefinierten Ansätzen ist die Trennung von Hardware und Software. WebAssembly und WASI fungieren als Hardware-Abstraktionsschicht und ermöglichen ein effizientes Deployment durch Plattformunabhängigkeit. Zudem fördern sie bereits in der Entwicklungsphase innovative Architekturen.

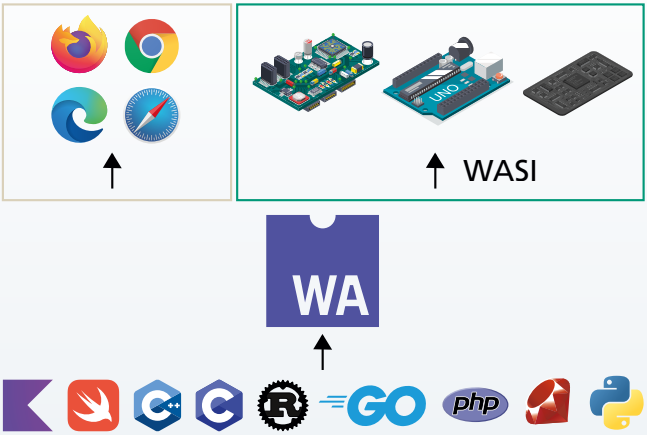


Software definierte Vorhaben sind ein zukünftiges Entwicklungsparadigma.

Jetzt ist der richtige Zeitpunkt, sich genauer mit der Technologie zu beschäftigen und disruptive Auswirkungen auf das eigene Geschäftsmodell zu evaluieren.

Sprechen Sie uns gerne an für Impulse, Ideen oder Kooperationen!

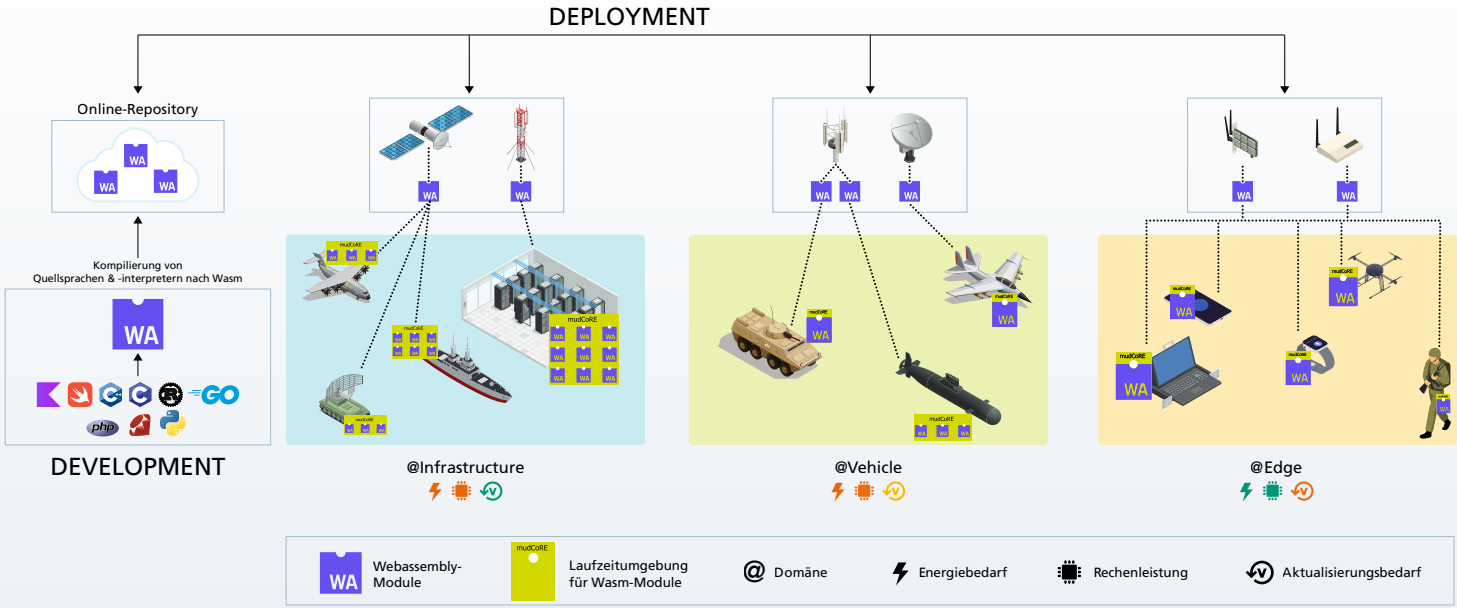
Weitere Informationen zu WebAssembly:



Ein Hardware Abstraction Layer (HAL) ist eine Software-Schicht, die als einheitliche Schnittstelle zwischen Hardware und Software fungiert. Dadurch funktioniert die Software unabhängig von einer spezifischen Hardware-Implementierung: Entwickler schreiben Software, die auf verschiedenen Hardware-Plattformen lauffähig ist – ohne sich um die spezifischen Details der Hardware zu kümmern.

WebAssembly, kurz Wasm, startete ursprünglich als Webtechnologie. Erstmalige Beachtung erhielt Wasm 2019, als die vier größten Browserhersteller (Mozilla, Apple, Google und Microsoft) die Technologie standardmäßig auslieferten.

WebAssembly ist als offener **W3C Standard** ein **Kompilierungsziel** und keine Programmiersprache. Diese Besonderheit verhilft zu disruptiven Systementwicklungsansätzen. WebAssembly liefert in einer **Laufzeitumgebung** (dediziert oder Browser) eine nahezu **native Performance** und überzeugt mit **Security**-Eigenschaften wie starkem Sandboxing.
Die beliebtesten Sprachen, wie z. B. **C, C++, Rust, Go, Python, Kotlin, Swift,** … unterstützen WebAssembly.



Grafik links: WebAssembly als Kompilierungsziel im Einsatz sowohl im Browser, als auch systemseitig.

Grafik oben: Neuartige Software-Architekturen ermöglicht durch WebAssembly + Laufzeitumgebung.

WebAssembly bedient das gesamte **Cloud-Edge-Kontinuum**: von leistungsstarken Servern bis hin zu ressourcenbeschränkten eingebetteten Systemen. Diese Eigenschaft sorgt für eine Art **Revolution im Software-Engineering** und die ersten Unternehmen, sowie Forschungsprojekte nutzen selbige Vorteile!

Potenziale durch den Einsatz von WebAssembly

- Rapid Development & Deployment durch **Plattformunabhängigkeit** (x86, ARM, RISC-V, …)
- Disruptive Software-Architekturen: **Cloud bis Edge inkl. Webunterstützung** für KI-, Robotik-, Simulation-, digitaler Zwilling oder Fahrzeuganwendungen
- **Legacy**-Systemen **einbinden**
- **Sicheres** Ausführen von **3rd-Party-Code** auf eigener Hardware
- Rust + Wasm = **Memory Safety + Isolation**

Wir bieten Ihnen

- **Potenzialanalyse**
  Grundlagen, Anwendungsfälle und Auswirkungen von Wasm für Ihr Vorhaben
- **Studien**
  Gemeinsamer Auftaktworkshop zum Definieren Ihrer Ziele, industrieübergreifende Experteninterviews, Publikationen und Konferenzen
- **Proof-of-Concept**
  Demonstrator Ihres Vorhabens inkl. techn. Abschlussbericht, sowie Source-Code & Dokumentation
- **Implementierung**
  Wasm in Ihre Entwicklungsprozesse und Architektur einbetten, Innovation-Transfer-Lab, Reallabor
- **Standardisierung von Wasm & WASI**
  je nach Bedarf

Kontakt

Nils Brand
Business Development
Division «Digital Innovation & Smart City»
Tel. +49 631 6800-2220
Mobil +49 170 9360863
nils.brand@iese.fraunhofer.de

Fraunhofer-Institut für
Experimentelles Software Engineering IESE
Fraunhofer-Platz 1
67663 Kaiserslautern
www.iese.fraunhofer.de