.NET Conf 2022

Perché WebAssembly

Sponsors

beanTech
IT moves your business

fluentis
the ERP ready to go live!

[stesi]
Powered by Innovation

With the support of:

<packt>

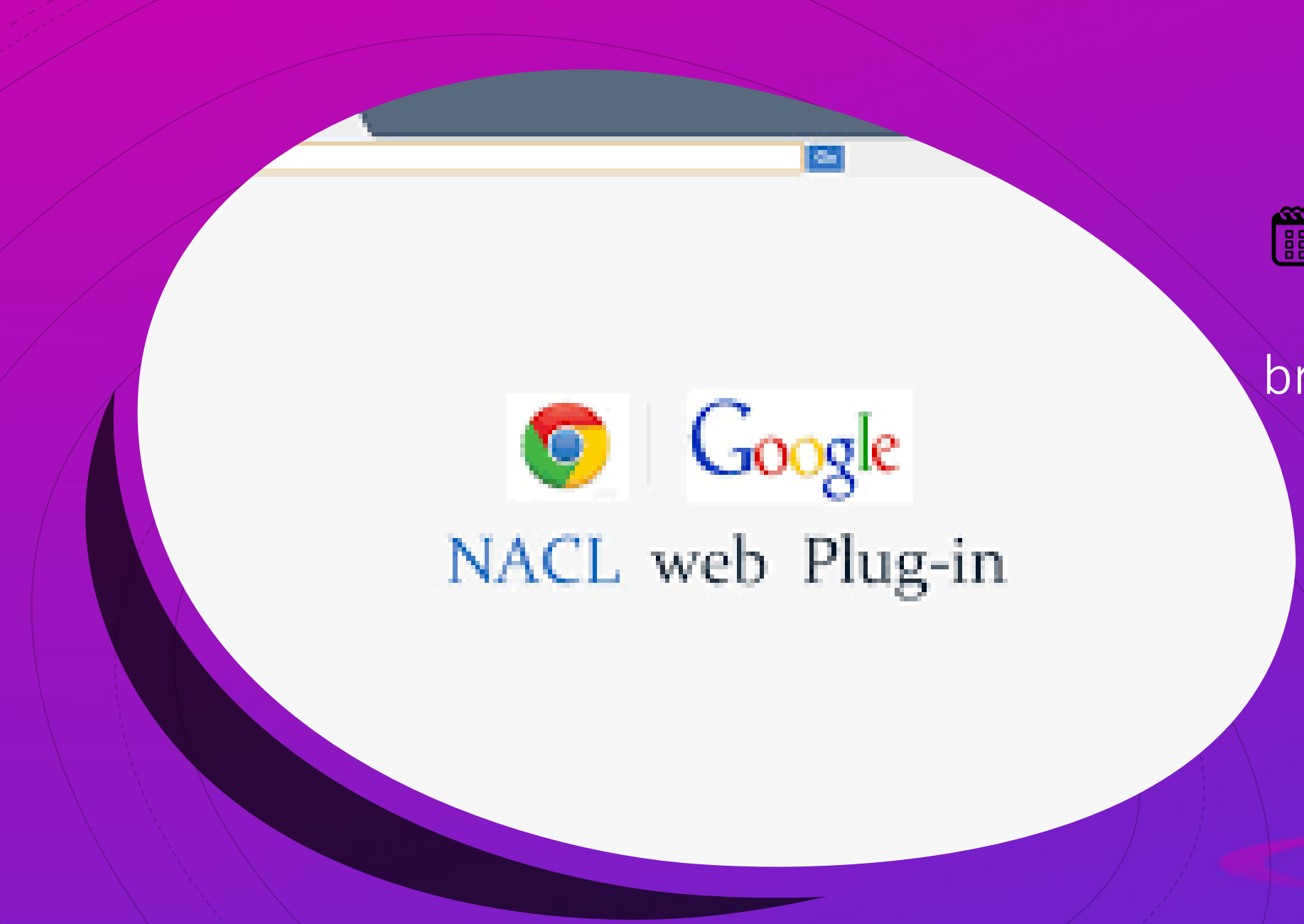
innOva
community

1989

Tim Berners-Lee

1995 – *Bernard Eich*

JavaScript!
e nulla fu più come prima!



2011 - *Google*

Sul mio
browser funziona!

2013 - *Mozilla*

Asm.js

Forse ci siamo

2015

Brendan Eich..

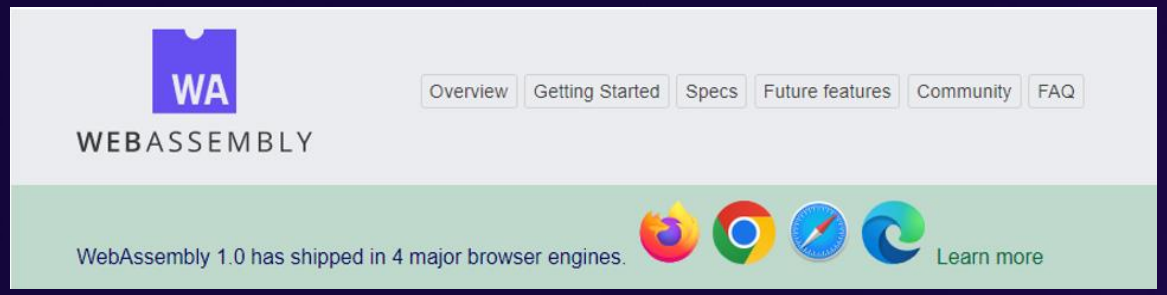
ancora!

Chi ha bisogno di
WASM?

Chi ha bisogno di Wasm?

jn0va community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

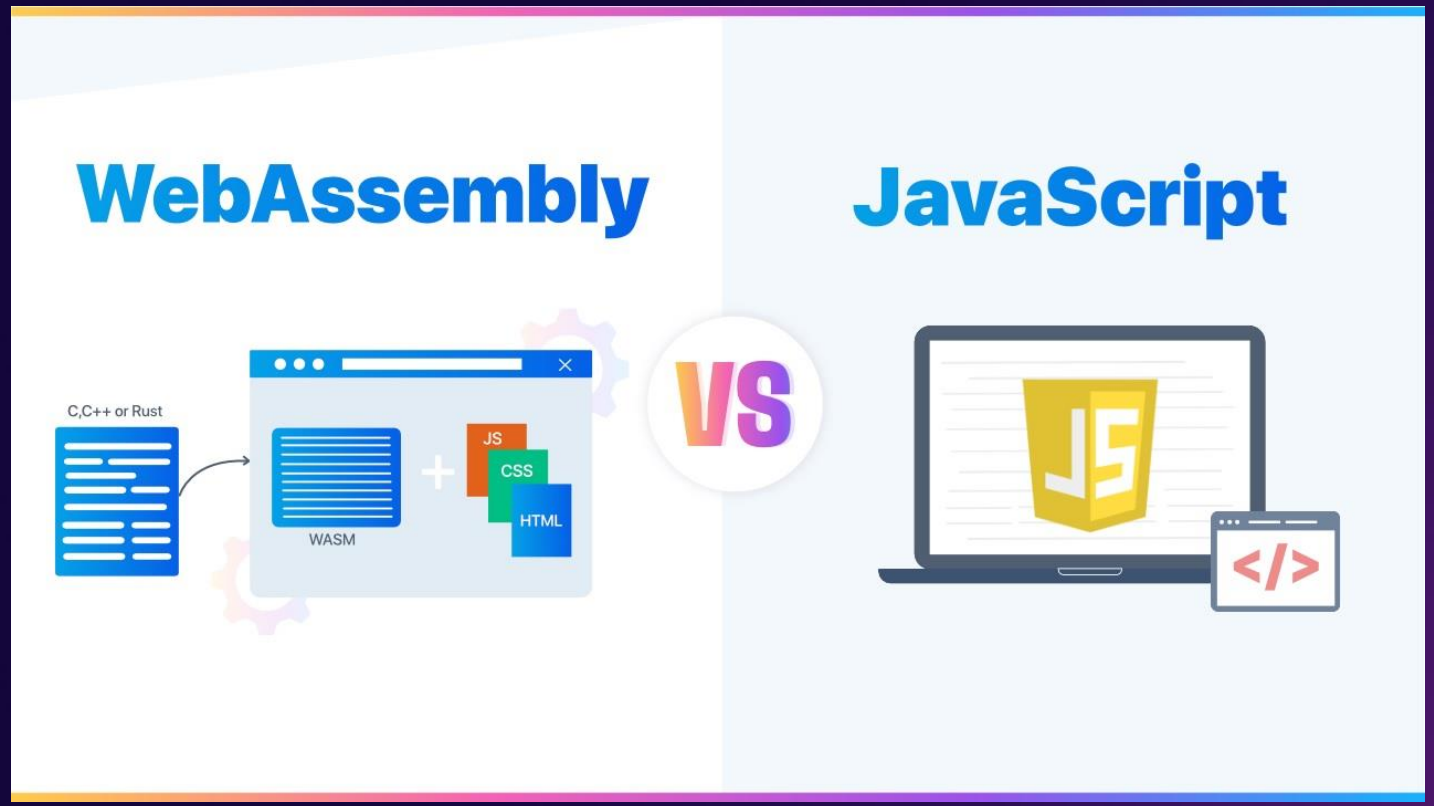
Cos'è WebAssembly?



WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

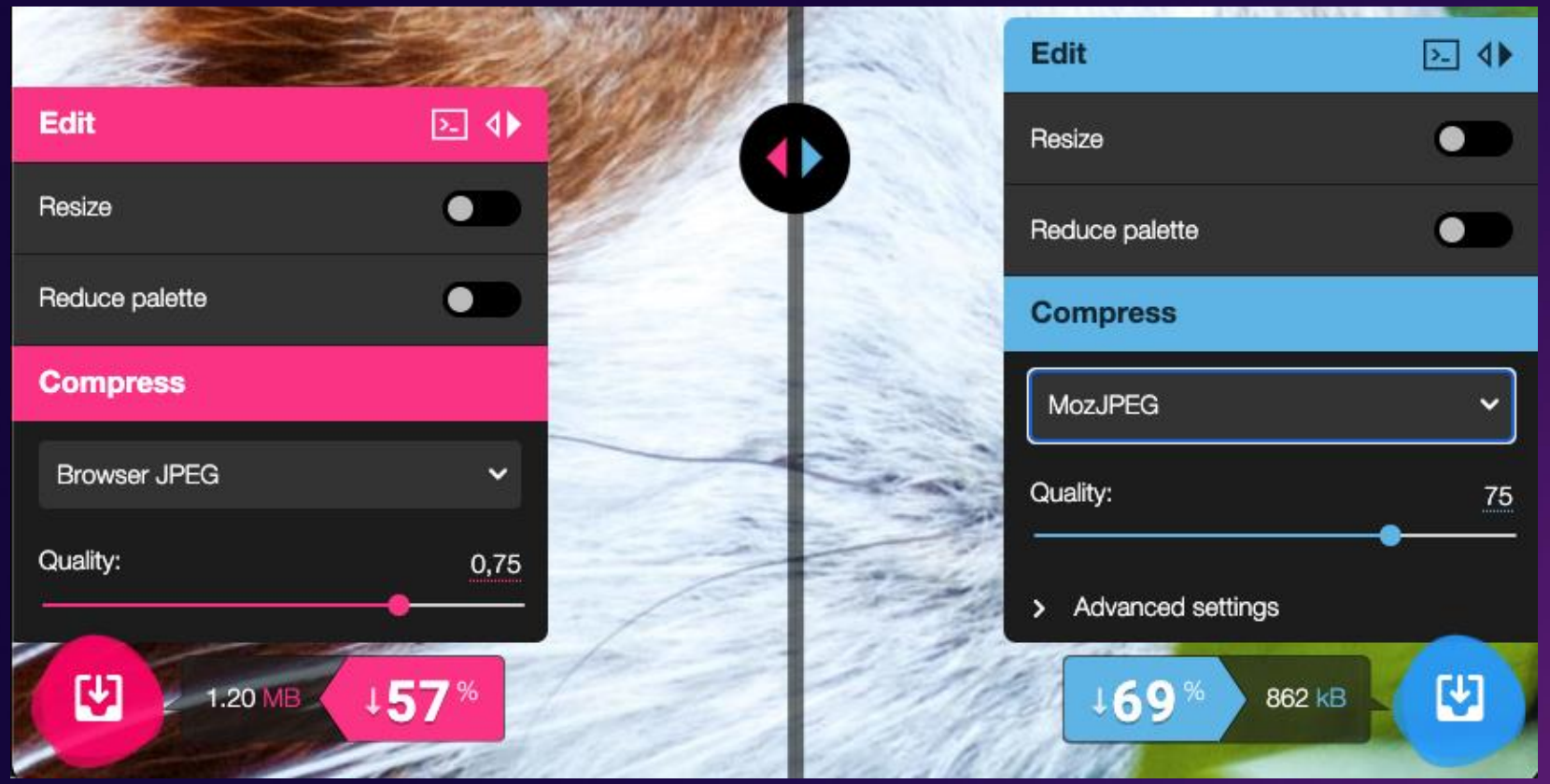
jn0va community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

JavaScript is dead. Long live to JavaScript!



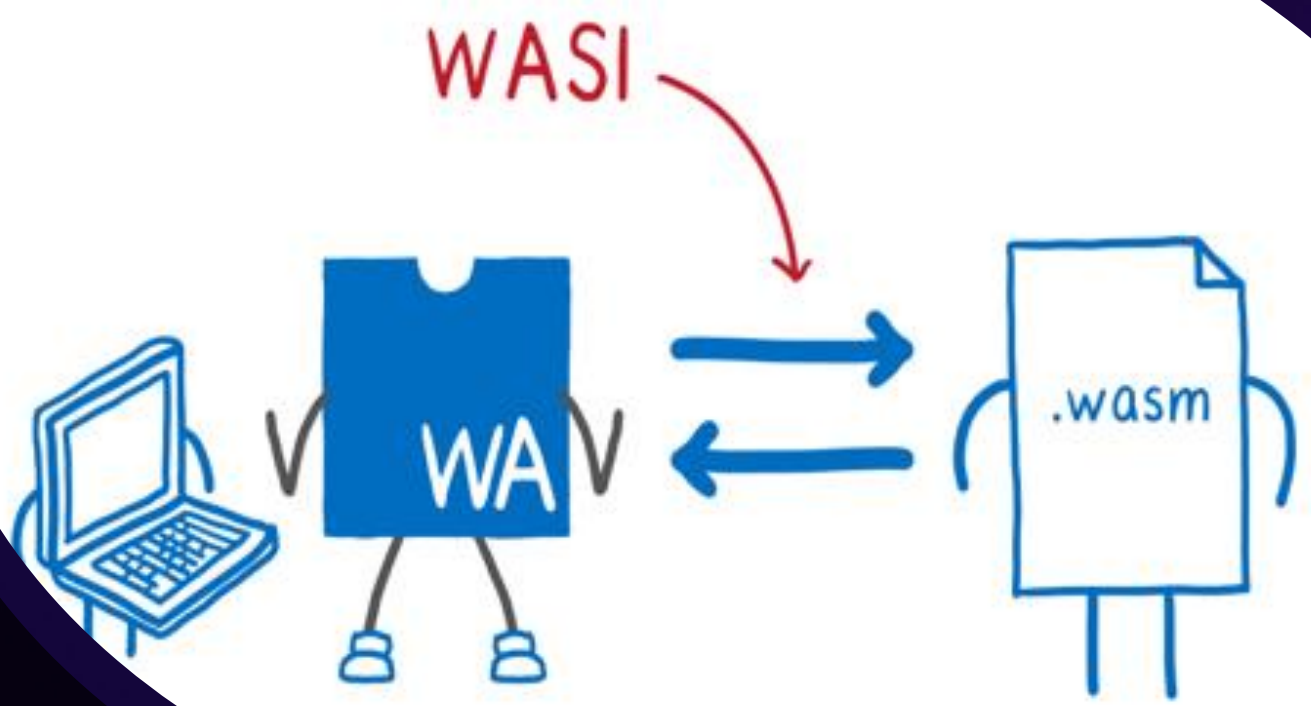
jn0va community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
‹packt›

Use Case: Ottimizzazione Immagini nel Browser



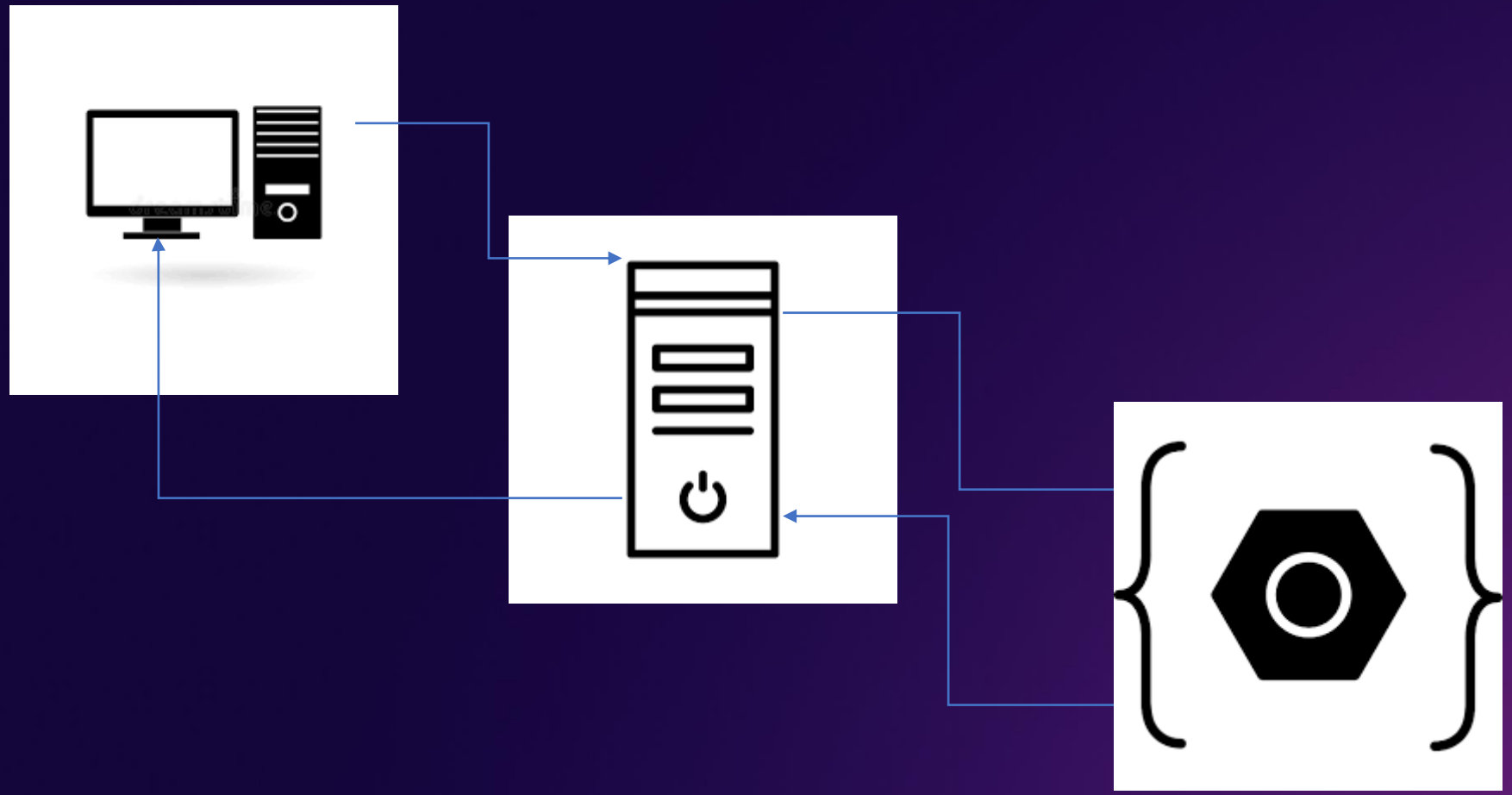
jn0va community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

WASI



jnova community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

WAGI



innova community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

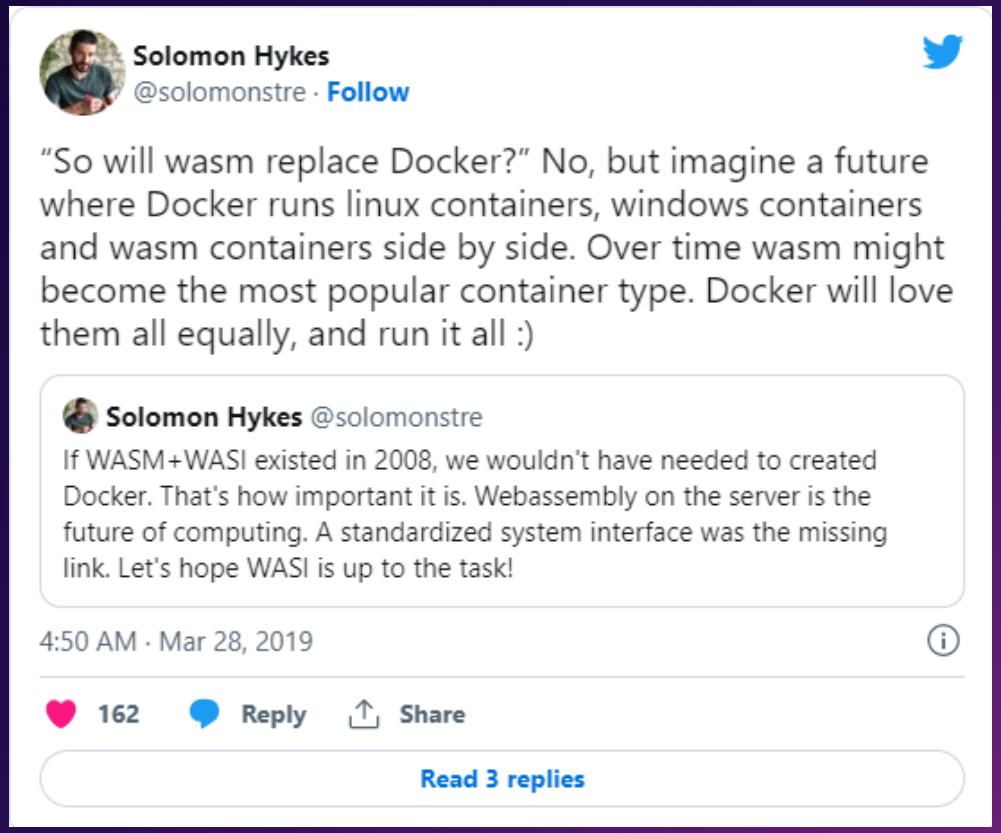
WebAssembly and .NET

- Enhanced debugging
- Hot reload improvements
- Expanded web crypto
- Improved JavaScript interop
- Multithreading (preview)

beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

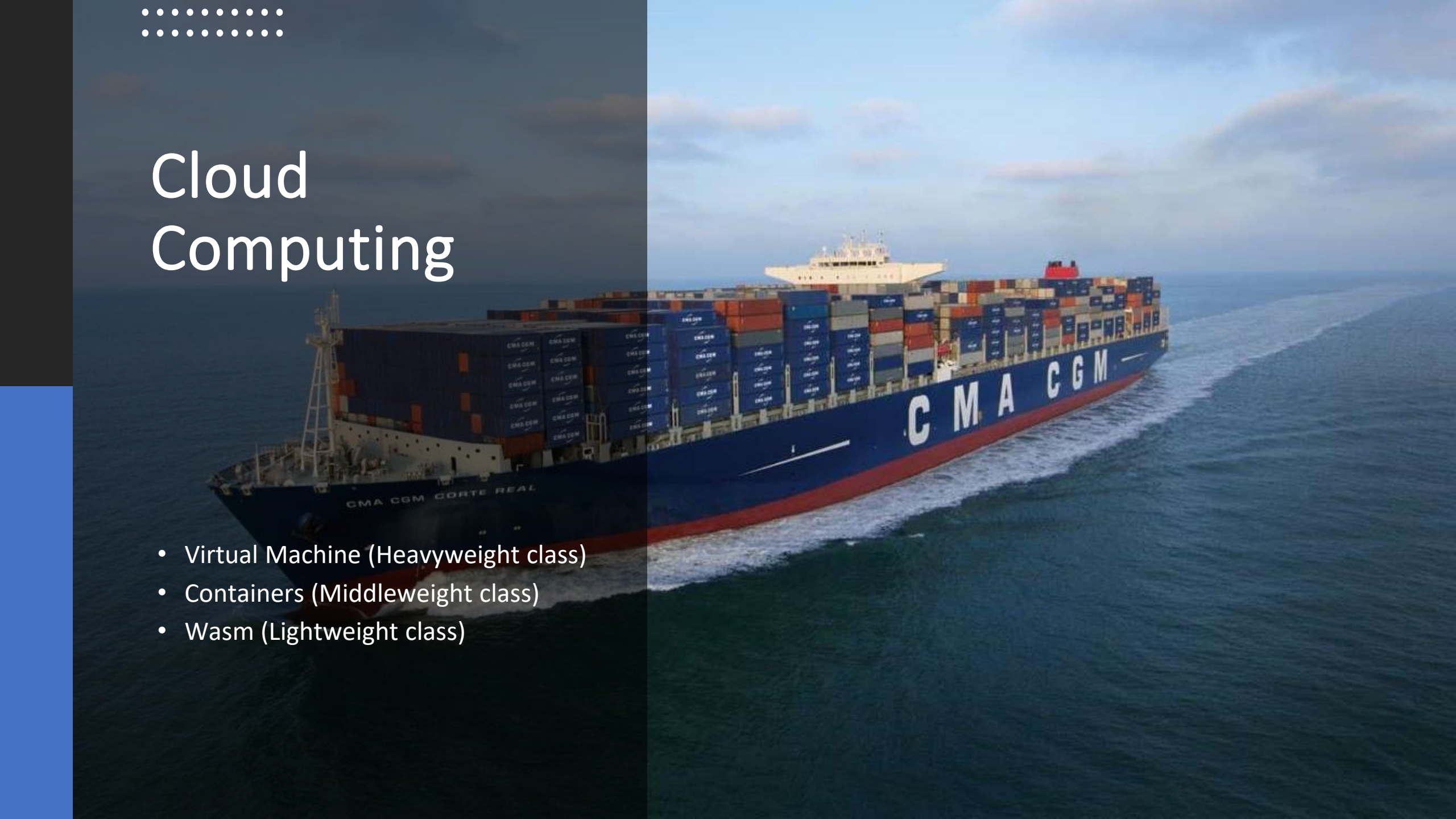
Demo

WebAssembly & Docker

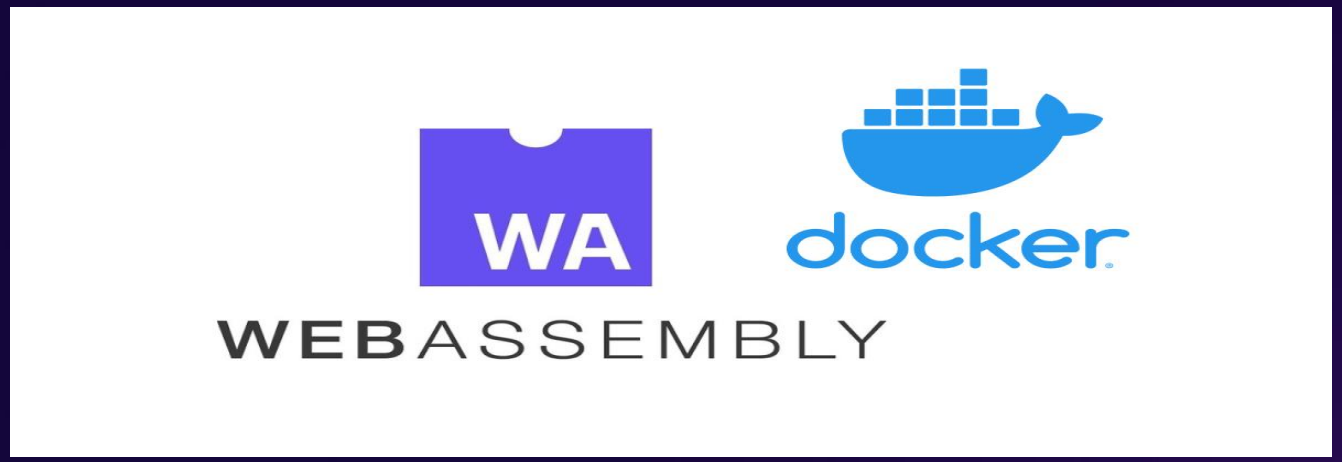


inOva community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

Cloud Computing



- Virtual Machine (Heavyweight class)
- Containers (Middleweight class)
- Wasm (Lightweight class)



Usiamo WebAssembly per

- Avviare e arrestare microservizi che svolgano, ognuno, un semplice task (AWS Lamba, Azure Functions?)
- Mantengono il consumo di risorse al minimo
- Possono scalare rapidamente

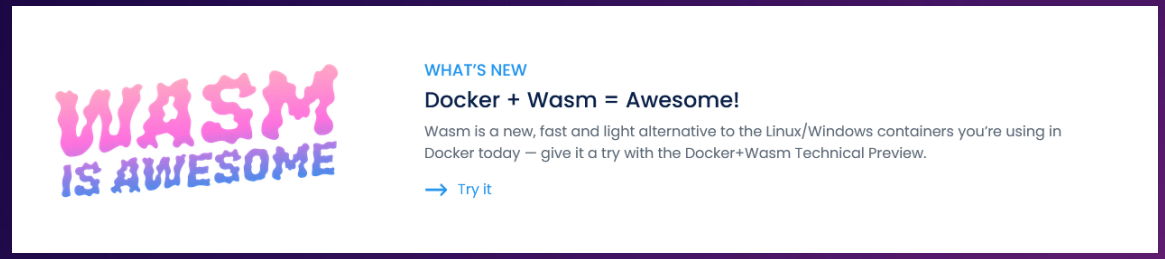
Usiamo Docker per

- Mantenere lo stato e organizzare le comunicazioni tra diversi moduli WebAssembly
- Long Running Process (Saga)
- Processi con carichi prevedibili

jn0va community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

Docker loves WebAssembly

> The Docker+wasm announcement makes perfect sense. We no longer live in a single-runtime world: there are Linux containers, windows containers, and wasm containers. OCI can package them all, I should be able to build and run them all with @docker. — Solomon Hykes, co-founder of Docker.



jn0va community
beanTech IT moves your business
fluentis the ERP ready to go live!
[stesi] Powered by Innovation
<packt>

Demo

About Us

alberto.acerbis@intre.it

stefano.maffeis@intre.it

Stefano Maffeis

Alberto Acerbis