



# Microservice **Store**

*Connecting Embedded World to enable Innovation*

A new Universe for  **CHERI**

→ Monolithic Approaches and RTOSes mask and hold back the CHERI's modular potential



# Monolithic Software Approaches for CHERI?



→ **Each MicroContainer/Microservice**

- Maps a CHERI Compartment
- Is an Independent Executable;  
Programming Language, Toolchain
- Independently Deployable

★ **Plug&Play**



# 70% embedded Security by CHERI

# 70% embedded Security by CHERI

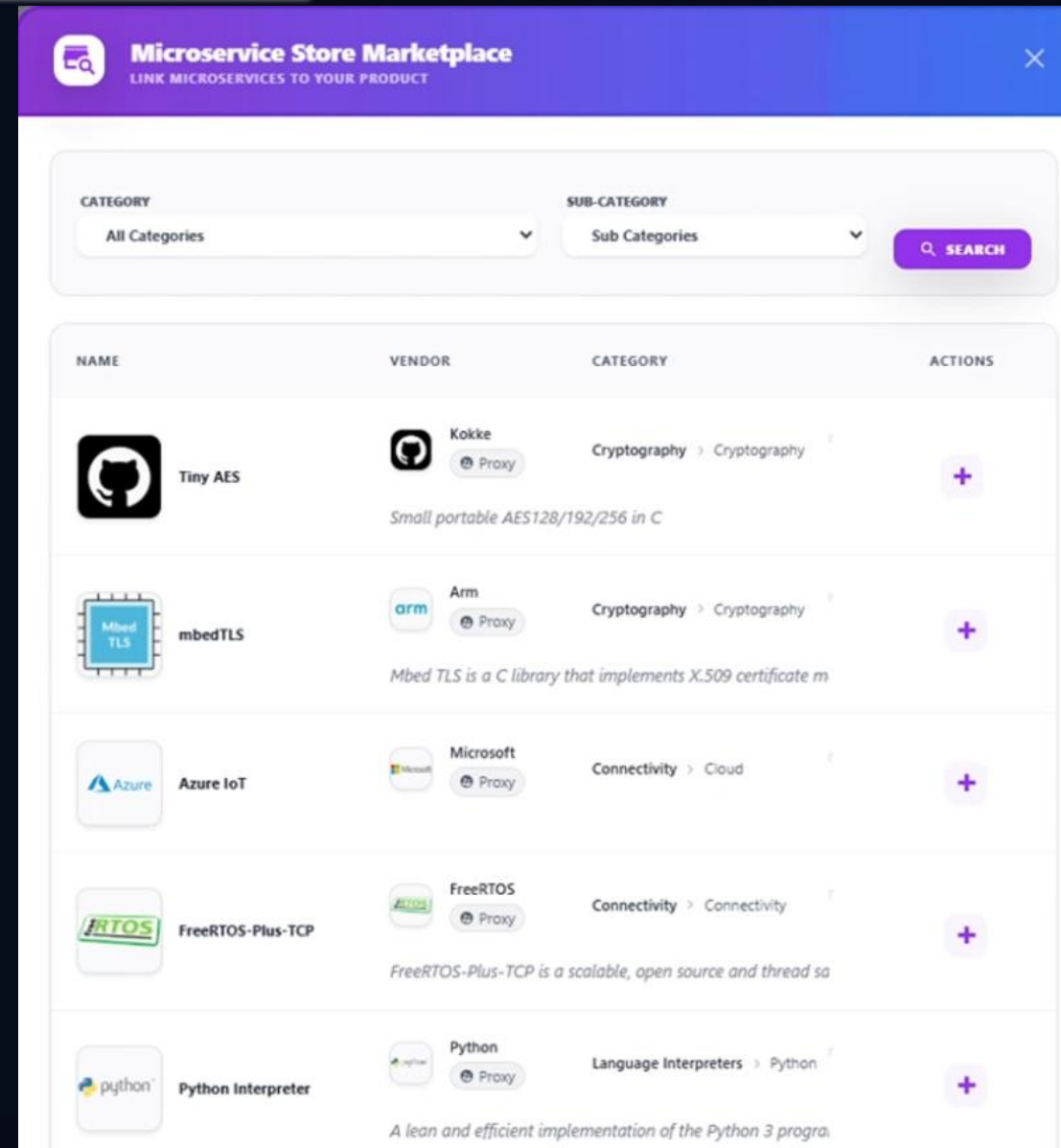
## + 30% embedded Security by Microservice Store

### Embedded Security Manager











- Handles all device security requirements; PSA 10 Goals
- Treats each CHERI Compartment as a Virtual Machine  
(Independent Secure Update, Boot, Lifecycle, Attestation )

## Plug&Play CHERI Microservices

- Bridging CHERI into existing SW Ecosystems
  - No waiting for native adoption; biggest barrier
- Go to market in weeks, reduce development costs.



The screenshot displays the 'Microservice Store Marketplace' interface. At the top, there is a search bar with 'All Categories' and 'Sub Categories' dropdowns, and a 'SEARCH' button. Below this is a table listing various microservices. Each row includes a name, a vendor logo, a category path, and an 'Actions' column with a plus sign icon. The services listed are Tiny AES, mbedTLS, Azure IoT, FreeRTOS-Plus-TCP, and Python Interpreter.

NAME	VENDOR	CATEGORY	ACTIONS
 Tiny AES	 Kokke Proxy	Cryptography > Cryptography	+
<i>Small portable AES128/192/256 in C</i>			
 mbedTLS	 Arm Proxy	Cryptography > Cryptography	+
<i>Mbed TLS is a C library that implements X.509 certificate m</i>			
 Azure IoT	 Microsoft Proxy	Connectivity > Cloud	+
 FreeRTOS-Plus-TCP	 FreeRTOS Proxy	Connectivity > Connectivity	+
<i>FreeRTOS-Plus-TCP is a scalable, open source and thread sa</i>			
 Python Interpreter	 Python Proxy	Language Interpreters > Python	+
<i>A lean and efficient implementation of the Python 3 progra</i>			

## Single (Cloud) Dashboard to design CHERI Powered Devices and manage Field Devices

**CloudTestProduct**  
Vendor: Smart Products Ltd

[I Need Support](#) [Certify My Product](#)

### Installed Images

Images running on the products.

[Upload Your Executable](#) [Generate by AI](#) [Add from Microservice Store](#)

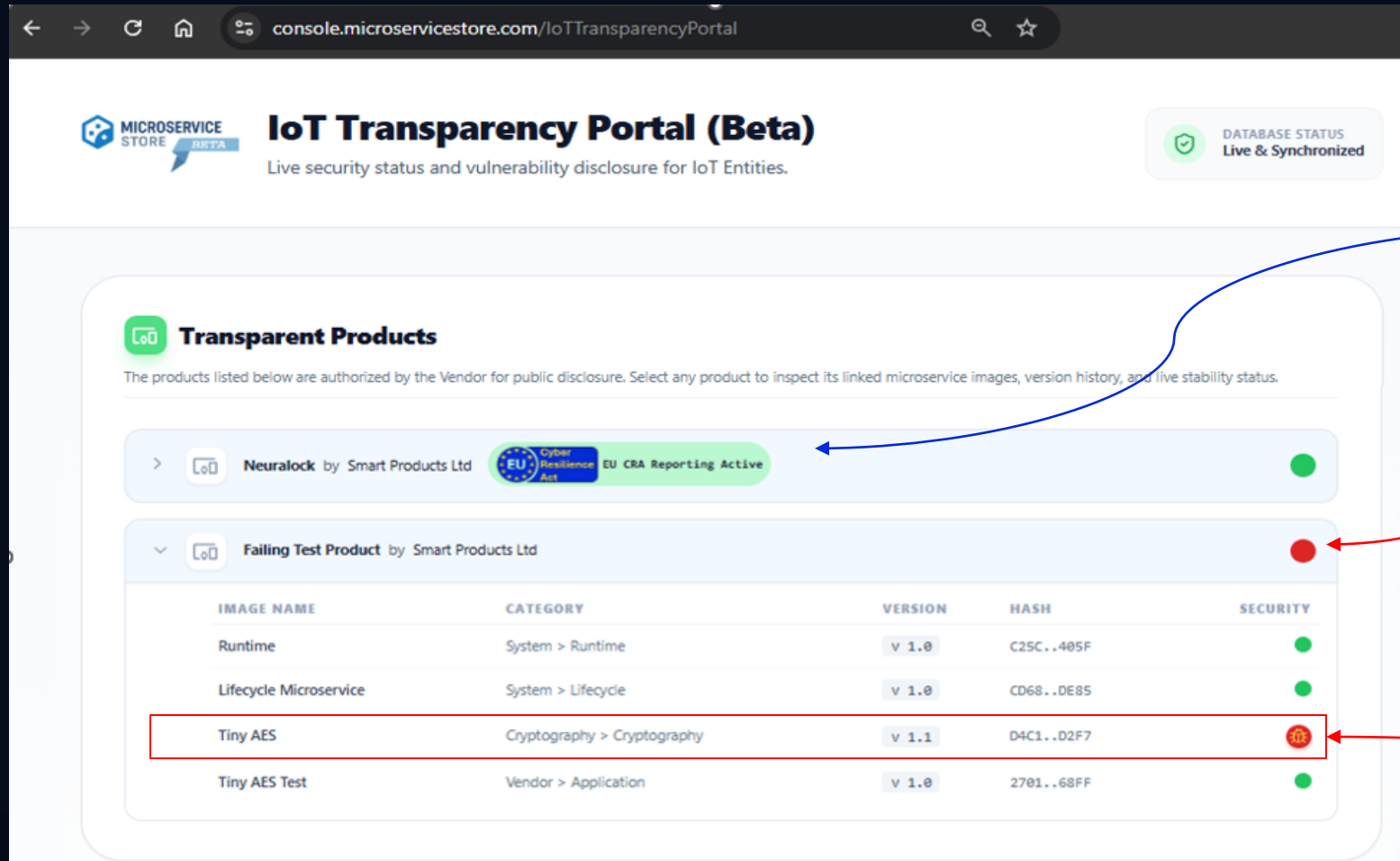
NAME	VERSION	TYPE	ATTRIBUTES ?	ACTIONS
<b>Microservice Runtime</b>	1.0	Runtime		
<b>Lifecycle Microservice</b>	1.0	Microservice (System)	XIP NotEncrypted	
<b>AWS IoT Device</b> <span>Vulnerable</span>	1.0	Microservice	XIP NotEncrypted Requested Resources: TLS Assigned Root Params: AWSIOT_DEVCERT AWSIOT_PKEY	API
<b>AWS IoT Device Test</b>	1.0	User Application	XIP NotEncrypted Assigned Resources: TIMER4 HW_AES	

- IoT Product Design
- Vulnerability Monitoring
- In-Field Lifecycle Management
- Supply Chain Management

# Compliance Automation for CHERI

→ EU CRA, UK PSTI Compliance is Automated

👉 **Live and public now** : <https://console.microservicestore.com/IoTTransparencyPortal>



The screenshot shows the IoT Transparency Portal (Beta) interface. At the top, there is a navigation bar with the Microservice Store logo and a 'DATABASE STATUS Live & Synchronized' indicator. The main heading is 'IoT Transparency Portal (Beta)' with the subtitle 'Live security status and vulnerability disclosure for IoT Entities'. Below this, there is a section for 'Transparent Products' with a sub-heading 'The products listed below are authorized by the Vendor for public disclosure. Select any product to inspect its linked microservice images, version history, and live stability status.' Two product cards are visible: 'Neuralock by Smart Products Ltd' with a green 'EU CRA Reporting Active' badge, and 'Failing Test Product by Smart Products Ltd' with a red status indicator. Below the product cards is a table of microservices with columns for IMAGE NAME, CATEGORY, VERSION, HASH, and SECURITY.

IMAGE NAME	CATEGORY	VERSION	HASH	SECURITY
Runtime	System > Runtime	v 1.0	C25C..405F	●
Lifecycle Microservice	System > Lifecycle	v 1.0	CD68..DE85	●
Tiny AES	Cryptography > Cryptography	v 1.1	D4C1..D2F7	🚫
Tiny AES Test	Vendor > Application	v 1.0	2701..68FF	●

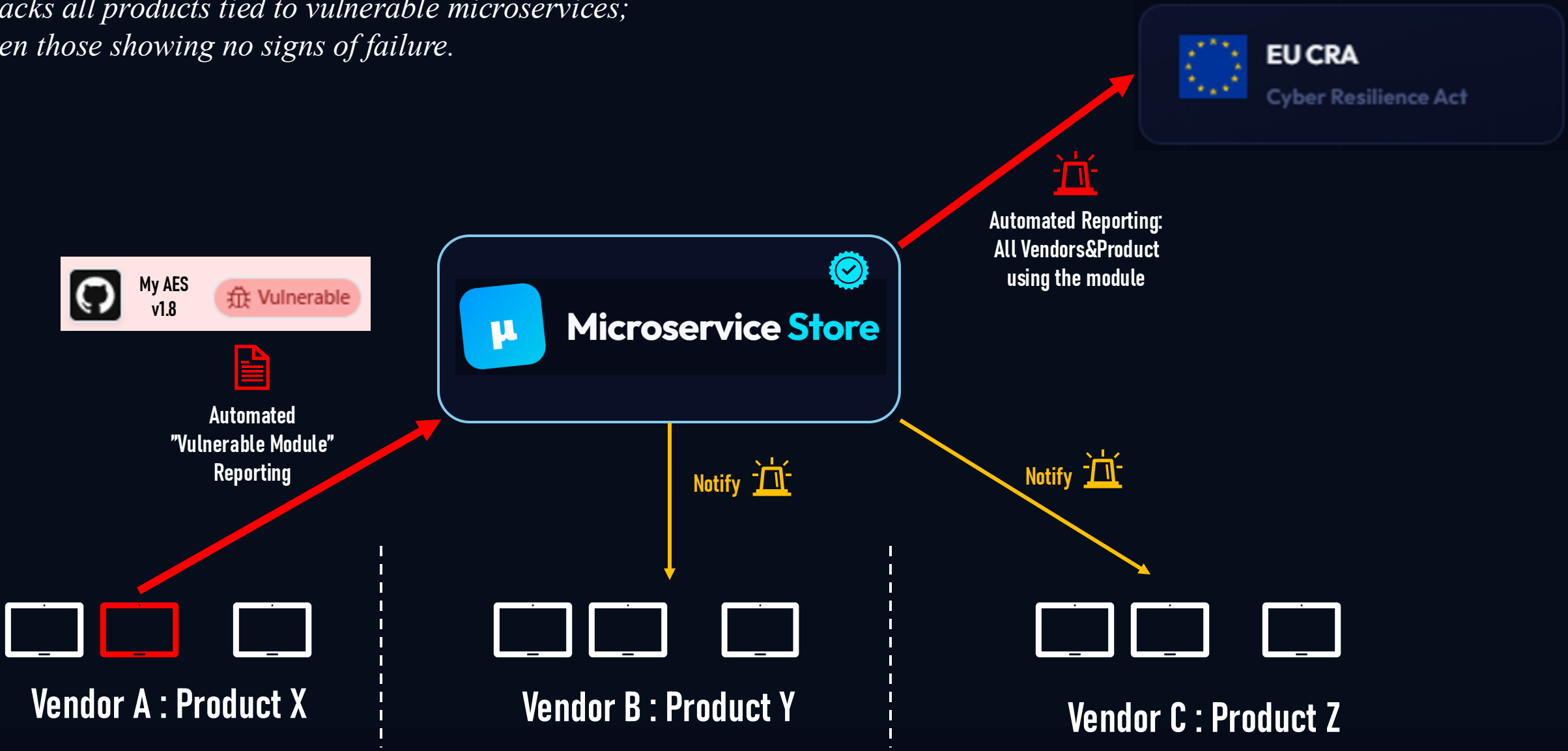
Automated Vulnerability Report to EU CRA.

Vulnerable Products  
Combined Module-Level SBOM and Vulnerability Reporting.

Vulnerable & Quarantined Module  
with Name, Version, Hash, Access Policy ...

# Centralised CHERI Security Hub

*Tracks all products tied to vulnerable microservices; even those showing no signs of failure.*





New Embedded Era  
**Microservice Store**

**Thank you**

**Contact Us**

 [info@microservicestore.com](mailto:info@microservicestore.com)

 [www.microservicestore.com](http://www.microservicestore.com)

See next slides for more details 

# One Ecosystem. *Every Player Connected.*

Click any node to explore how it works and who delivers it.



## → **Supply-Chain Security:** *”Protecting the entire fleet and operations.”*

- Automated In-Field Security Lifecycle
- Automated SBOM and Vulnerability Reporting
- EU CRA, UK PSTI Compliance Automation



EU CRA

Cyber Resilience Act



UK PSTI

Product Security Act

## → **Device Protection:** *“Protecting the embedded device pro-actively”*

- Fault Containment : Compartmentalisation (Any CPU Architecture)
- Security Requirement Handling
- Module-Level Vulnerability Reporting



psacertified™  
by GlobalPlatform

Global  
Platform™  
**SESIP**

## → **Microservice Protection:** *“Protecting each sub-embedded module individually”* 💡

- Expands device protection requirements onto each Microservice, as individual virtual environments.
- Isolation
- Microservice-Level Vulnerability Reporting.

## → High-Technology with Simplified Development

*“Eliminating challenges of traditional software development: Fast, Efficient, Secure”*

- **Platform-Agnostic:** *Implement Once Run Everywhere: Any CPU/Toolchain/IDE/Programming-Language*
- **No Migration, No Learning Curve**
- **No Performance/Runtime Overhead:** *Microservice Binaries are native-machine instructions (No-runtime Interpretation)*
- **Unbrickable:** *As stated in the previous slide*

```
main.c
1  #include "SysCall.h"
2
3  int main(void)
4  {
5      /* Platform Agnostic; No Clock/HW/Driver Initialisation */
6
7      LOG_INFO("Hello World from My Microservice");
8
9      /* Each Container/Microservice is also multithread */
10     pthread_create(&thread_id, &attr, &thread_start, &tinfo);
11
12     /* POSIX API is Available */
13     sockfd = socket(AF_INET, SOCK_STREAM, 0);
14     connect(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
15
16     while (true) { }
17
18     return 0;
19 }
```

# Build a Product In Seconds; Store and AI



Installed Images  
Images running on the products.

Upload Your Executable

Generate by AI

Add from Microservice Store

## Generate Yourself

- For private usage/product
- For Public (In Store)

### Generate your Microservice by AI

SELECT AI GENERATION ENGINE

Claude Opus 4.6 (5 votes)

MICROSERVICE NAME: AES

GENERATION PROMPT (REQUIREMENTS):  
Generate an AES Encryption Microservice that supports only CBC mode and a 256-bit key length. The interface shall be PSA Crypto API.

GENERATION ATTRIBUTES: Unit Test (100% Test Coverage, Static Analysis), MISRA-C

VERIFICATION: Remote HW Test (Reserves a physical test slot on our cloud-connected boards to verify logic post-generation.)

Generate

### Microservice Store Marketplace

LINK MICROSERVICES TO YOUR PRODUCT

SEARCH: All Categories, Sub Categories

NAME	VENDOR	CATEGORY	ACTIONS
Tiny AES	Kokke	Cryptography > Cryptography	+
mbedTLS	Arm	Cryptography > Cryptography	+
Azure IoT	Microsoft	Connectivity > Cloud	+
FreeRTOS-Plus-TCP	FreeRTOS	Connectivity > Connectivity	+
Python Interpreter	Python	Language Interpreters > Python	+

## Plug&Play Microservices from Microservice Store

### “App Store Moment for IoT”

- Get any functionality in seconds (No code development, no integration, no compile)
- Runs as Isolated and Access Controlled; Third-Party Microservices cannot break the device.

## Generate A Microservice by AI

### “Revolutionising embedded AI”

- High Success AI: Generate a single functionality for a platform-agnostic environment.
- Runs as Isolated and Access Controlled : Trusted-AI

“Microservice Store let Product Vendors to download plug-and-play Microservice from third-party securely.”

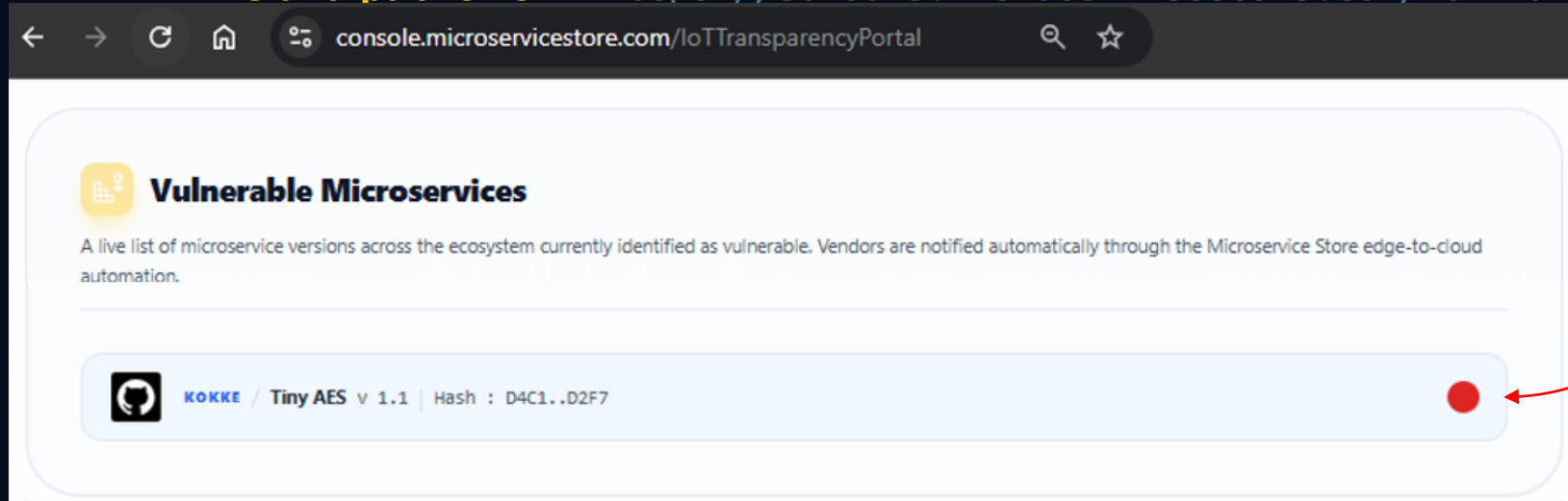
→ **Vendor Reporting and EU CRA Compliance is Automated.**

✓ Vulnerabilities quarantined on the field device.

✓ **Centralized Security Hub; Tracks all products tied to vulnerable microservices; even those showing no signs of failure.**



✓ **Live and public now** : <https://console.microservicestore.com/IoTTransparencyPortal>



Live Status of Vulnerable Microservices from the Store.

Developer must provide a fix in finite-days to prevent ban from the store.



# Built for *Every Player* in the Ecosystem

Your personalised dashboard is one click away.



## Product Vendor

Ship secure products fast

- Install verified Microservices instantly. One-Click.
- Automated Fleet Management: provisioning, OTA, vulnerability reporting.
- EU CRA / UK PSTI compliance
- Third-party Software — Safely
- Trusted AI Code Generation

[Product Dashboard →](#)



## Developer

Build, publish, and earn!

- Open your store; start innovation
- Publish Microservices globally
- Recurring revenue on every deployment
- No IP Exposure
- Track, debug, evolve in-field
- Skip marketing and sales, concentrate on what you do best; engineering!

[Developer Dashboard →](#)



## Partner

Join us! Let us build together!

- Design houses: Embedded Development Support
- Security experts: Certification, Compliance
- Domain Expertise
- Tool Vendors: IDEs, Test Tools
- AI providers: Code Generation
- Cloud: Monitoring & Analytics

[Join Partner Program →](#)