Ready-to-use products

Use containers like **apps on a smartphone**. Partners, open source communities or INSYS created fully functional containers for a specific pupose Users just download, configure and use them.

Example: icom Data Suite (iDS)

📰 ? 🗄	C Activate profile	IN <mark>SYS</mark> icom
atus ita points	Event	
iteway U	Description signal quality undershot ->publish message	
ata logger essages ents Iministration elp	Event Analogue data point has changed Data point mdbDp20 - Funkfrequenz user-defined value 238 value of data point Check for event upon start-up	
	Action Send message Send message mqttMsg1 - weather station publish mes >	
	Save settings	

Figure 1:Configure instead of program

Example: Mirasoft AnyViz

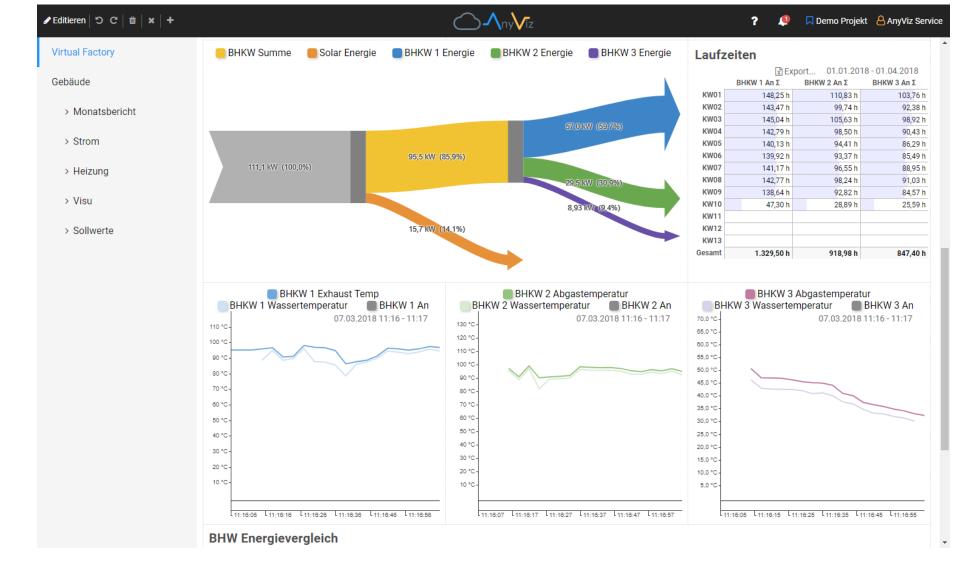


Figure 2: Similar to iDS, but completelly cloud based

Pros and cons

- Configuring instead of programming
- Support from vendor
- May cost license fees
- Maybe hard to extend (add features)

Recommendation

Ideal for **non-programmers**.

Containers - where to start?!

Different approaches to run software on the edge of the cloud, respective use case, development resources and scalability INSYS icom, Michael Kress

Well known Linux distributions

Use Linux distributions like on a **Raspberry Pi**. Minimalistic versions of standard Linux distributions are often well known to user. There are common packet managers included (like apt-get, apk), so additional software can be installed like on a normal Linux PC.

Example: Debian Linux

Figure 3: Debian is a base for a lot of other distributions.

Example: Alpine Linux

Figure 4: Alpine Linux tries to be as slim as possible.

Pros and cons

- Well known tools (like packet managers)
- Incredible wide range of available software
- Very good community support
- Dependent on distribution decisions
- Container size can quickly grow very large
- Updates after long time can be problematic

Recommendation

Ideal for **fast PoC** (Proof of Concept).

Basic Containers

Use script languages to program without need for a (cross) compiler.

There are containers with installed programming languages like Python or NodeJS. Users can log into such a container and immediately write their programs. Alternatively a program is written on a PC and simply copied into the container.

Example: Python scripts

This little script subscribes to an MQTT broker and displays received data on a simple HTTP server.

import paho.mqtt.client as mqtt import http.server import socketserver def on_connect(client, userdata, flags, rc): client.subscribe("machine/temp") def on_message(client, userdata, msg): with open("index.html", "w", encoding='utf-8') as file: file.write(f"<html><body>Temperature: {msg.payload}</body></html>") 12 mqtt_client = mqtt.Client() 13 mqtt_client.on_connect = on_connect 14 mqtt_client.on_message = on_message 15 mqtt_client.connect_async("mqtt.broker.de", 8889) 16 mqtt_client.loop_start() 18 Handler = http.server.SimpleHTTPRequestHandler 19 with socketserver.TCPServer(("", 8080), Handler) as httpd: httpd.serve_forever()

Pros and cons

- Minimal environment for low container size
- Programming experience like on PC
- Independent from architecture
- Sustainable because of little mainantance
- No separate SDK or PC software neccessary,
- programming within container
- Dependent on creator of base container
- Extending functionality can be problematic

Recommendation

Ideal for **scripters** to solve smaller tasks.

Use build scripts and SDK (Software Development Kit) to create own containers. Intended for experts: Use the programming language of your choice, use available open or closed source as you please. The container might even consist of only a single binary! Collection of useful links: https://m3-container.net/#scripts

Basic container as template

Figure 5: Scripts on github in combination with the SDK create containers



Ideal for **professional developers**, optimal for mass roll out on huge scale.

Develop for yourself

There are build scripts on github that serve as templates for own containers. Use for all programming languages like C/C++, go or C#.

	ontainer: Create container for devices built upon		EINSTS MRA, MRU		- •	
\rightarrow C d $\overline{\mathbf{A}}$	A https://github.com/insys-icom/M3_	Container		☆	රි 🗄	
insys-icom / M3_Conta	ainer	Q Type [] to searc	h	>_ + ▼ ⊙ [1	n 🗛 👯	
Code 🕙 Issues 🕅 Pull r	equests 🕑 Actions 🖽 Projects	🛱 Wiki 🕕 Security	🗠 Insights	鐐 Settings		
M3_Container (Public)		🛠 Edit Pins 👻	O Unwatch 9	ד 😵 Fork 19 ד 🖧 Sta	r 12 👻	
양 master → 양 2 branches	𝔊 0 tags	Go to file Add file •	<> Code -	About	\$¥	
😷 m-kress fix link again	Create container for devices built upon INSYS´ M3 architecture, like INSYS MRX, MRO, SCR or ECR.					
closed_packages	add example how to trace network traffic in a container 8 months ago					
doc	update a lot of the open source packag	update a lot of the open source packages 2 months ago		া Readme আ⊉ MIT license		
images	added files to rootfs_skeleton 7 years ago		-∿ Activity			
oss_packages	update a lot of the open source packages 2 months ago		2 months ago	☆ 12 stars		
rootfs_skeleton	update a lot of the open source packag	update a lot of the open source packages 2 months ago		 9 watching 9 forks 		
rootfs_staging	huge update of everything; from now o	huge update of everything; from now on please use M3SD last year		Report repository		
scripts	fix link again last week					
working	huge update of everything; from now on please use M3SD last year		Releases			
.gitignore	allow container_pcap in closed_source	allow container_pcap in closed_sources 8 months ago		No releases published		
LICENSE	change license of scripts to MIT; update first doc site last year		Create a new release			
README.md	modify readme again		last year	Packages		
≘ README.md			Ø	No packages published Publish your first package		
Project web	page: https://m3-co	ontainer.ne	et	Contributors 9		

Pros and cons

• Control over everything

• Minimal container size

• Integrate into your existing CI/CD pipeline

- 100% traceable builds
- Minimum surface for hacking attempts
- Least external dependencies
- Requires deeper knowledge
- Initially requires more development time

Recommendation