

ROSCon 2023  
2023/10/19



**ex on N E R V E S**

# *a bare minimum runtime platform for ROS 2 nodes in Elixir*

**Hideki Takase**, Shintaro Hosoai (The Univ. of Tokyo),  
Ryota Kinukawa (Tombo Works), Shunsuke Kikuchi (SAKURA internet, Inc.)



A part of this work is supported by JST CREST JPMJCR21D2 and the commissioned research (04001) by National Institute of Information and Communications Technology (NICT), Japan.



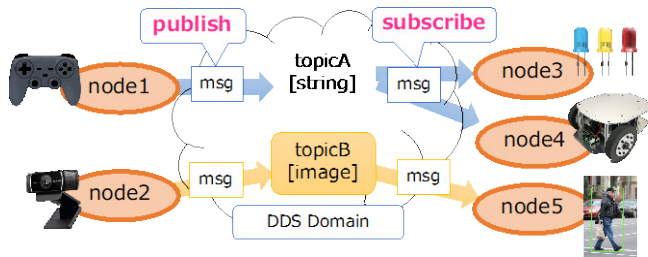


# Topic: Rclex on Nerves

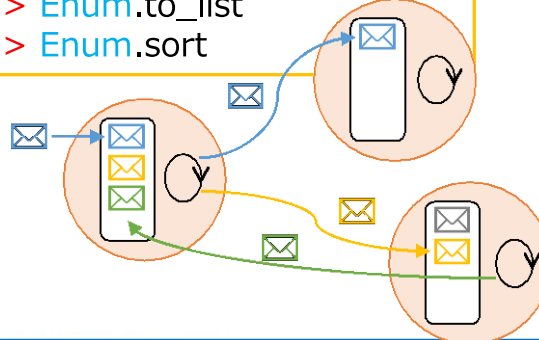
Functional Language



Client Library for Elixir



```
1..1000  
> Flow.from_enumerable()  
> Flow.map(&foo(&1))  
> Flow.map(&bar(&1))  
> Enum.to_list  
> Enum.sort
```



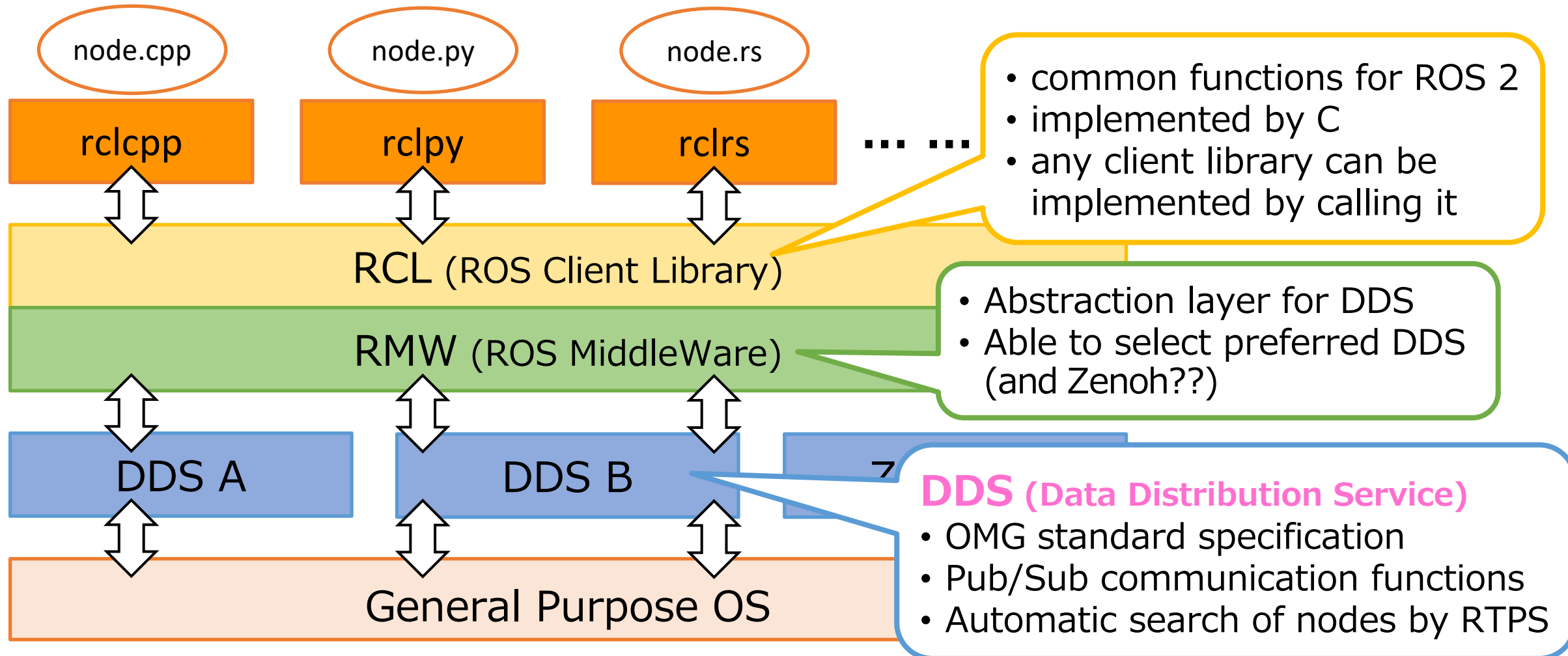
bare minimum and resilient runtime platform



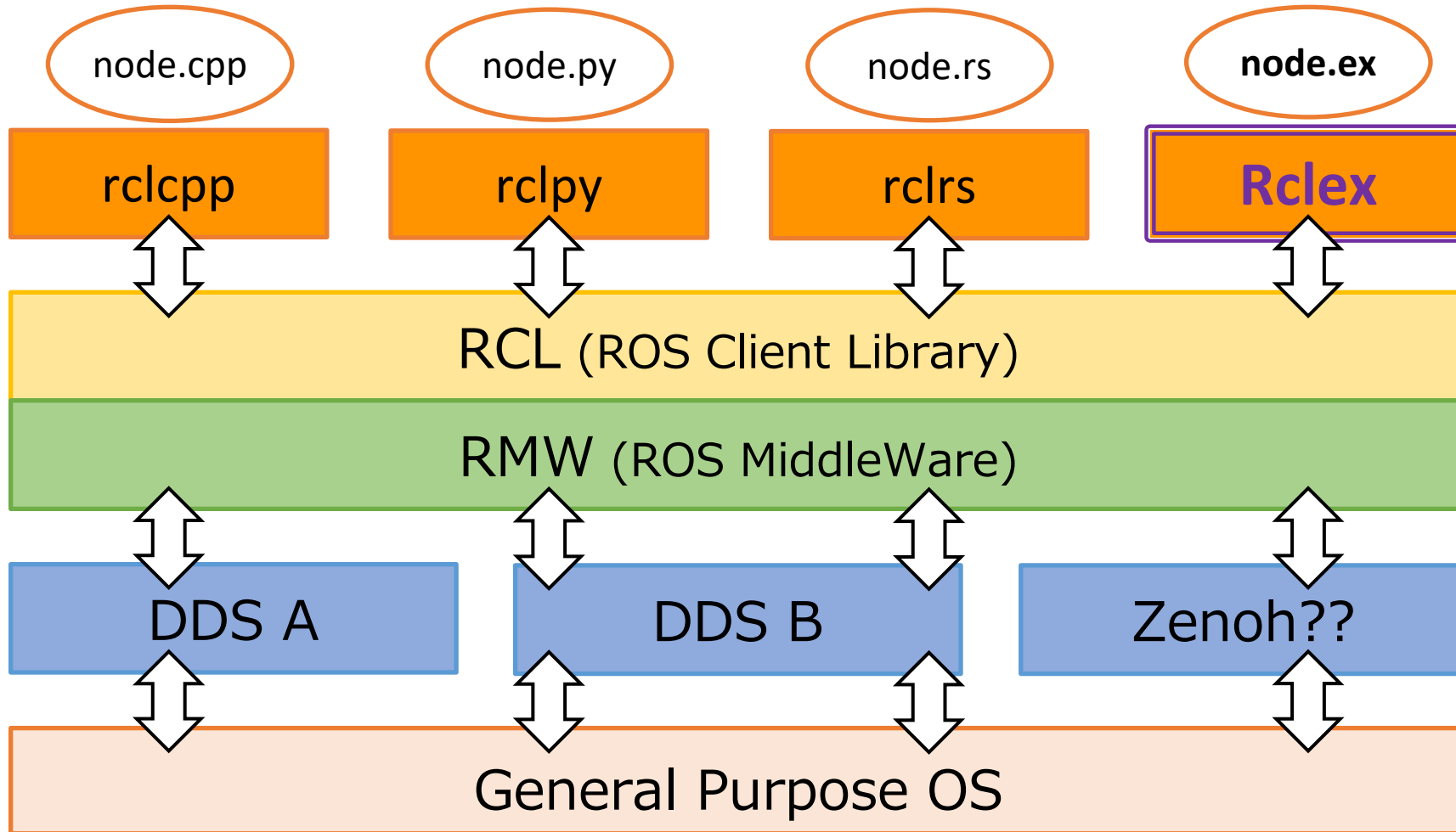
**N E R V E S**



# ROS 2 is awesome!!



# ROS 2 Client Library for,,,



 [github.com/rclex](https://github.com/rclex)



# We love Elixir!!



elixir=

## Functional language (appeared in 2012)

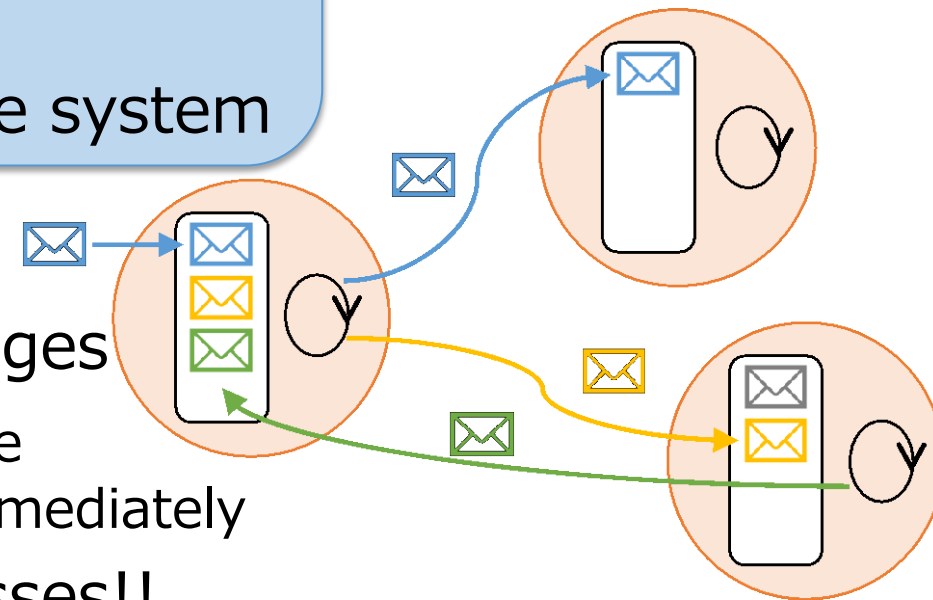
### Operated on Erlang VM (BEAM)

- lightweight processes with robustness
- highly concurrency/parallelism
- soft real-time feature
- easy to realize distributed and fault tolerance system



### • Similar to **Actor Model**

- Actors (processes) send and receive messages
- "Let it Crash": The problematic process should be promptly crashed and restored immediately
- We can spawn a massive number of processes!!



# We love Elixir!!



elixir

## Functional language (appeared in 2012)

### Language Design based on Ruby

- easy to learn! familiar grammar and notation
- very highly productive despite functional
- high communication performance suitable for IoT



### example Elixir code

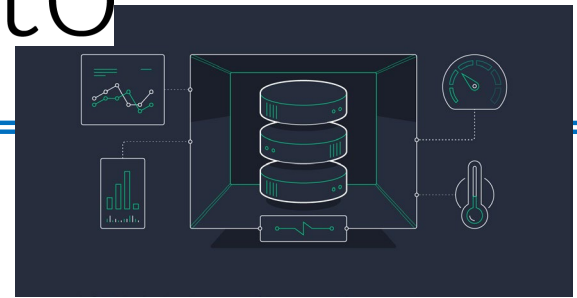
```
1..1000
|> Flow.from_enumerable()
|> Flow.map(&foo(&1))
|> Flow.map(&bar(&1))
|> Enum.to_list
|> Enum.sort
```

*Programming should be about transforming data*

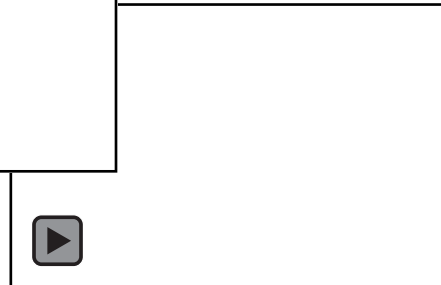
- Data flow and parallel processing can be described intuitively with **Enum Flow** |>



# We love Elixir!!

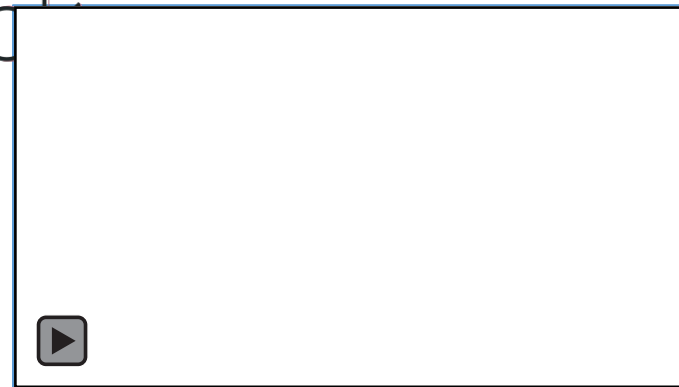
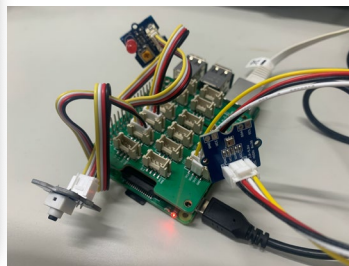
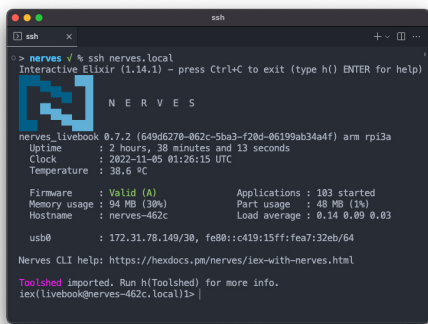


- Awesome ecosystem!



- Nx**
- Multi-dimensional typed arrays (aka tensors)
  - Numerical definitions (defn)
  - A subset of Elixir for numerical computation
  - Automatic differentiation

```
1 | model =  
2 |   Axon.input({nil, 784})  
3 |   |> Axon.dense(128, activation: :relu)  
4 |   |> Axon.dropout(rate: 0.5)  
5 |   |> Axon.dense(10, activation: :softmax)
```





- 
- 
- Actors on **BEAM** send and receive messages
    - (I believe) Actor Model is suitable for robot systems





# Simple Publisher by Rcllex

```
1  defmodule RcllexNode do
2    def pub(num_node) do
3      Rcllex.rclexinit()
4      |> Rcllex.create_nodes('rclex_node', num_node)
5      |> Rcllex.create_publishers('chatter', :single)
6      |> Rcllex.Timer.timer_start(1000, &func/1)
7    end
8
9    def func(publisher_list) do
10     n = length(publisher_list)
11     msg_list = Rcllex.initialize_msgs(n, :string)
12     Enum.map(0..(n - 1), fn index ->
13       data = "Hello World from rclex_node_" <> to_string(index)
14       IO.puts("publish message: #{data}")
15       Rcllex.setdata(Enum.at(msg_list, index), data, :string)
16     end)
17
18     Rcllex.Publisher.publish(publisher_list, msg_list)
19   end
20 end
```

setting node info.  
with pipe operator

create node by  
name and index

number  
of nodes

create topic  
:single-> one topic  
:multi-> multiple

set callback func  
and timer interval

processing a data  
message with Enum

create list of  
messages for  
each node

publish message



# Example of Rcllex on Native Ubuntu

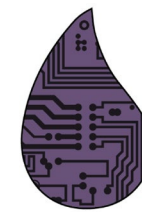
default ROS 2 env



Rcllex on Ubuntu



# NERVES



Elixir Circuits




Phoenix Framework

## Developed by [Nerves Project!](#)

- Dedicated IoT Platform for Elixir
  - fully compatible with Elixir
  - useful libraries for IoT systems

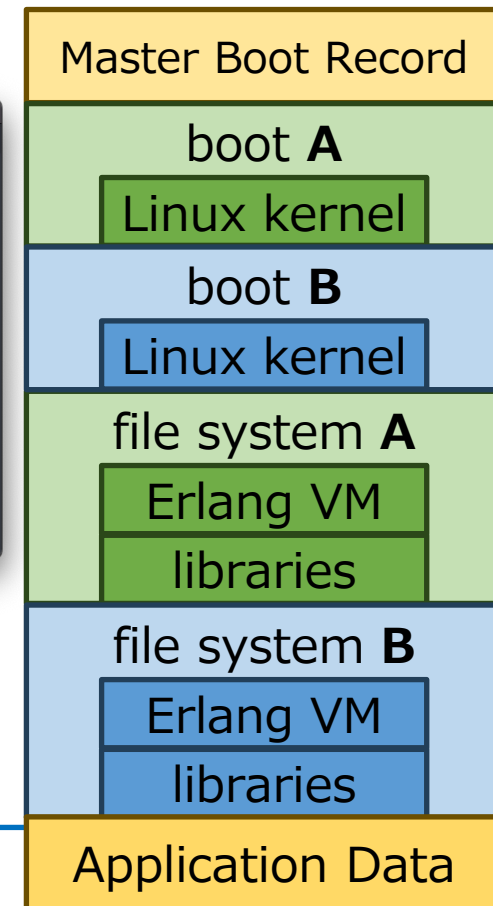
- *bare-minimum!!*

- too small (one linked fw,  $\sim > 30$  MB)
- portable and easy to configure 

- *resilient!!*

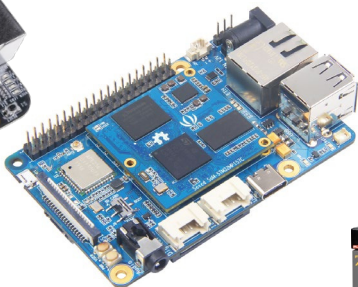
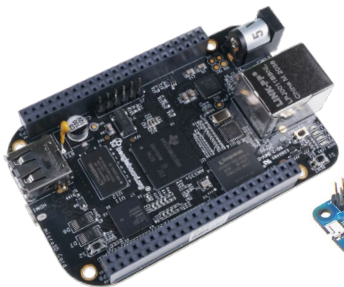
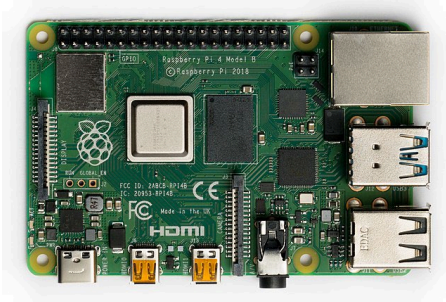
- of course, robust thanks to Erlang VM!!
- file system is duplicated and read-only for reliability

```
ssh
ssh x
o> nerves ✓ % ssh nerves.local
Interactive Elixir (1.14.1) - press Ctrl+C to exit (type h() ENTER for help)
NERVES
nerves_livebook 0.7.2 (649d6270-062c-5ba3-f20d-06199ab34a4f) arm rpi3a
Uptime : 2 hours, 38 minutes and 13 seconds
Clock : 2022-11-05 01:26:15 UTC
Temperature : 38.6 °C
Firmware : Valid (A) Applications : 103 started
Memory usage : 94 MB (30%) Part usage : 48 MB (1%)
Hostname : nerves-462c Load average : 0.14 0.09 0.03
usb0 : 172.31.78.149/30, fe80::c419:15ff:fea7:32eb/64
Nerves CLI help: https://hexdocs.pm/nerves/iex-with-nerves.html
Toolshed imported. Run h(Toolshed) for more info.
iex(Livebook@nerves-462c.local)1>
```



[github.com/rclex](https://github.com/rclex)

# Supported Targets



ROS_DISTRO	arm64v8	arm32v7
Foxy		

# ex on How to try it out?

- Only 7 steps, just in 5 min (I hope,,, :D

1. create & prepare project for target

```
$ mix nerves.new rclex_on_nerves --target rpi4
```

2. Install Rclex into Nerves project

```
add {:rclex, "~> 0.9.2"}, into mix.exs and $ mix deps.get
```

3. prepare ros2 resources to Nerves filesystem

```
$ mix rclex.prep.ros2 --arch arm64v8
```

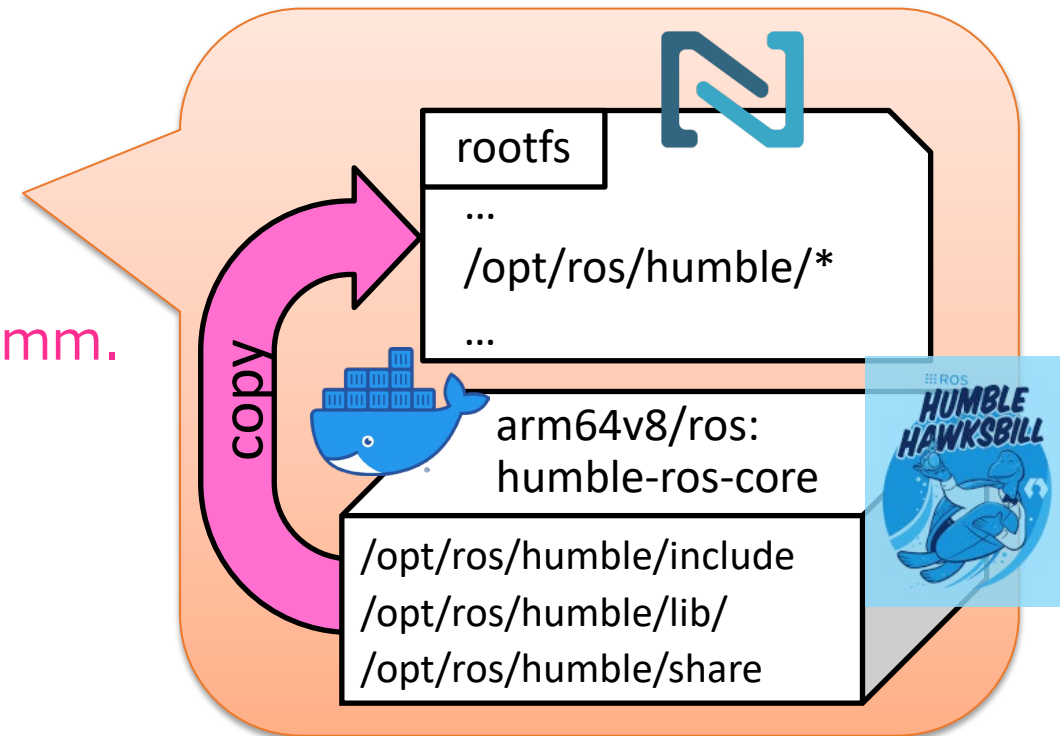
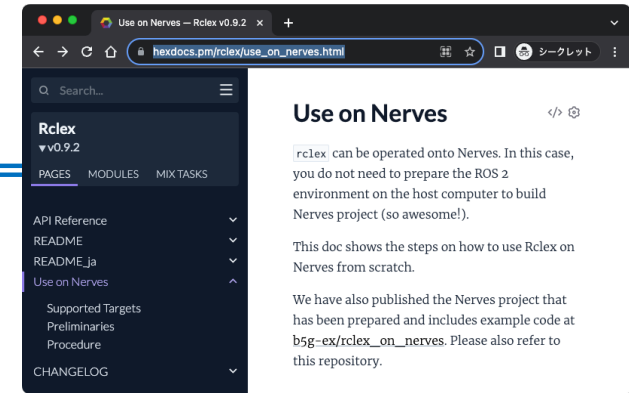
4. generate resources for msg types of topic comm.

```
$ mix rclex.gen.msgs
```

5. set LIBRARY\_PATH into Nerves firmware

6. enjoy writing Rclex code!!

7. finally, build firmware, and burn to microSD



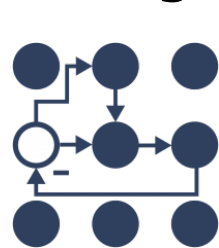


- For **ROS** developers
  - will acquire performance, especially in communication
  - will want to employ Elixir/Erlang ecosystem
  - will get a **bare minimum and resilient environment!!**

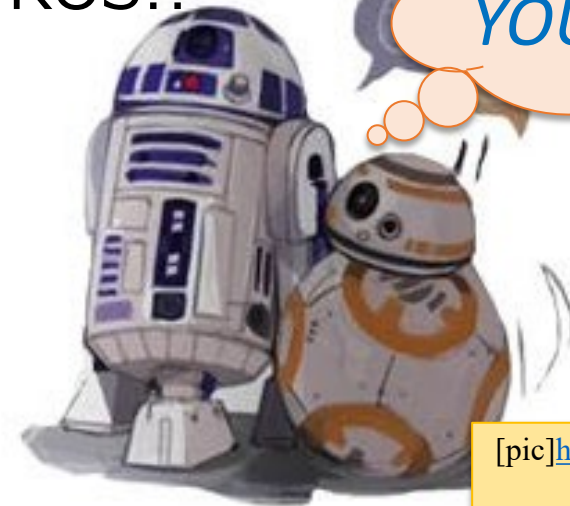
Phoenix Framework



- For **Elixir** developers
  - will get autonomous communication
  - will utilize OSS packages published for ROS!!



GAZEBO



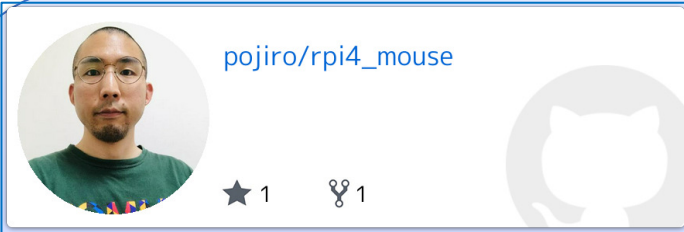
MAY THE **BEAM**  
BE WITH  
**YOUR ROBOT!!**



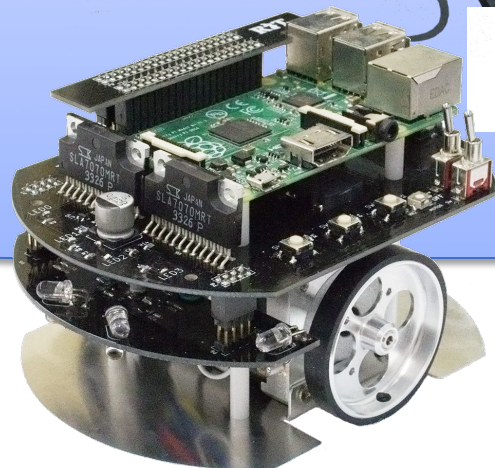
# 論よりRUN!!

"ron yori run"

The RUN is mightier than the word



[https://github.com/pojiro/rpi4\\_mouse](https://github.com/pojiro/rpi4_mouse)



Raspberry Pi Mouse V3  
by RT Corporation

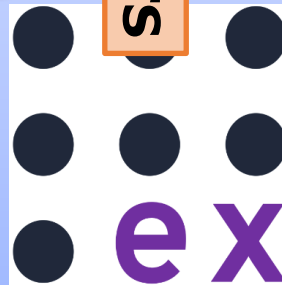


Momo (WebRTC Native Client) by Shiguredo

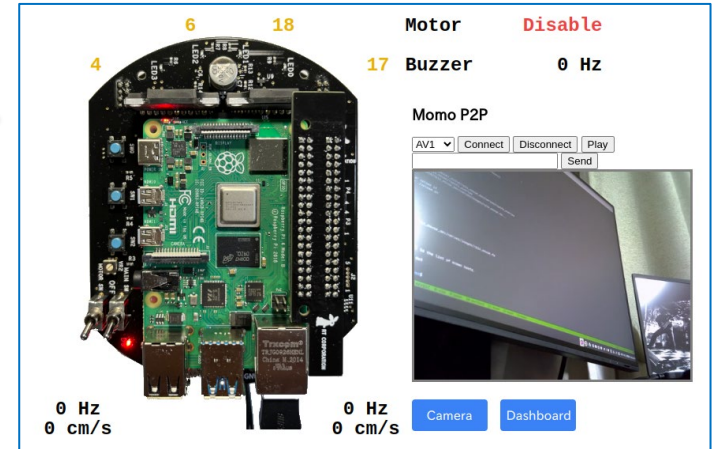
Phoenix Framework



sensors



control motors



/cmd\_vel

/sensors



論よりRUN!! *"ron yori run"*

The RUN is mightier than the word

ge  
se  
no

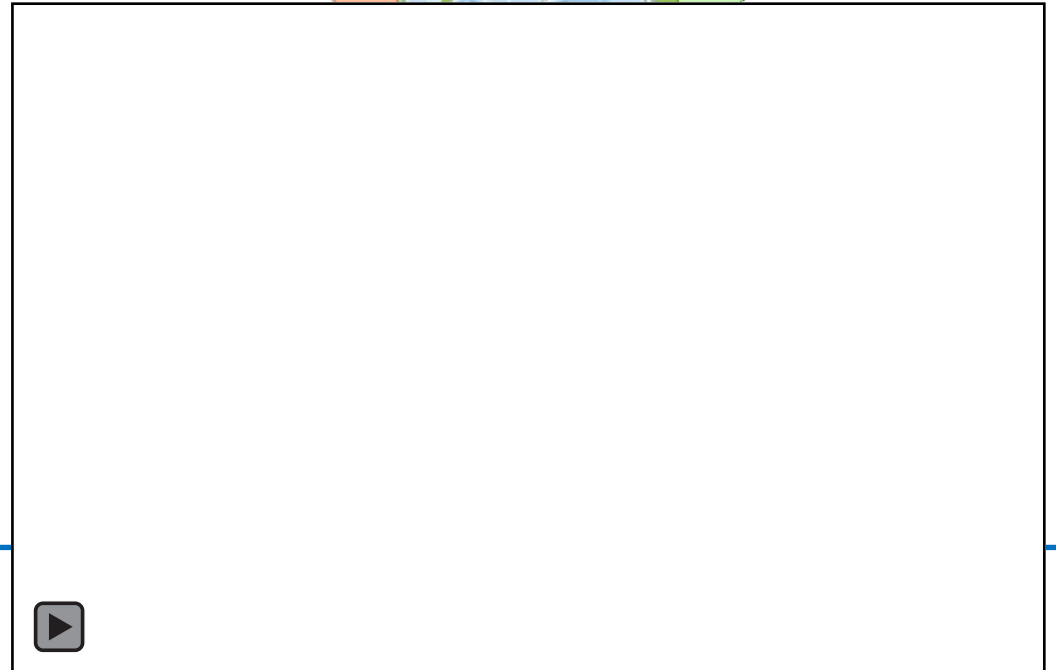
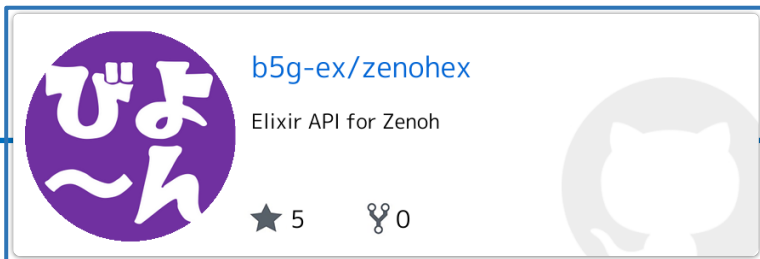
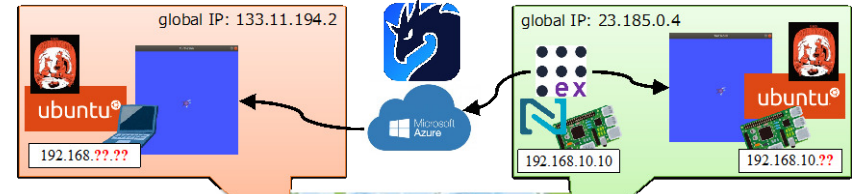
# Rclex on Nerves with Zenoh



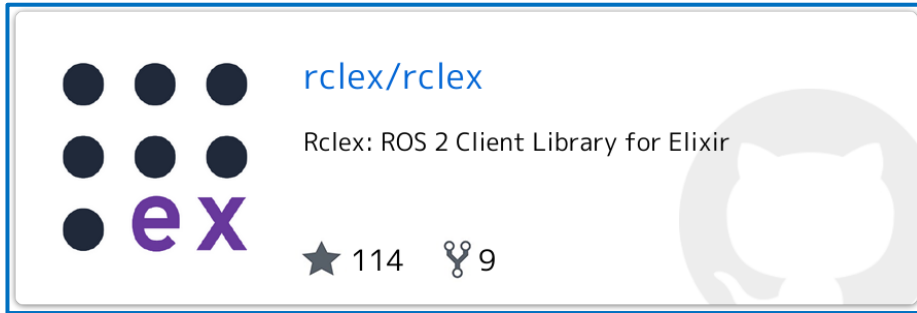
- Eloquent, Scalable, and Fast network protocol
- The most promising candidate for a new Tier-1 RMW??

## • Our Trial

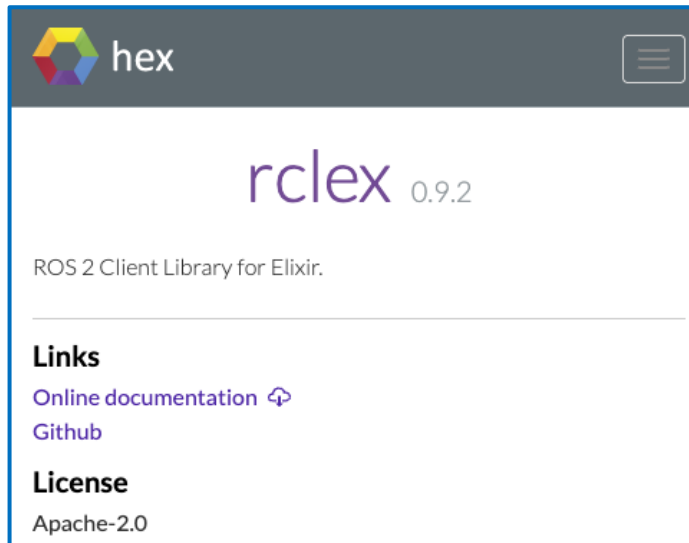
- Rclex msgs on Nerves go beyond the Pacific Ocean!!
  - ✓ [Code BEAM America 2023](#)  
[\[Slide\]](#) [\[YouTube\]](#) [\[demo\(on X\)\]](#)
- (WiP) Zenohex [\[GitHub\]](#)



# Wrapping Up



<https://github.com/rclex/rclex>



<https://hex.pm/packages/rclex>

- **Rclex**: Robotics meets Elixir!
  - **ROS** will acquire **Elixir**'s performance
  - **Elixir** will utilize **ROS** packages
- Ongoing/upcoming works
  - Service, Action and Parameter
  - actively integration with Zenoh
  - documentation, academic paper,,,



**STAR our repo! try it out!!  
and, Give your contributions!!!**

