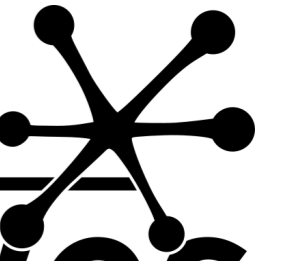


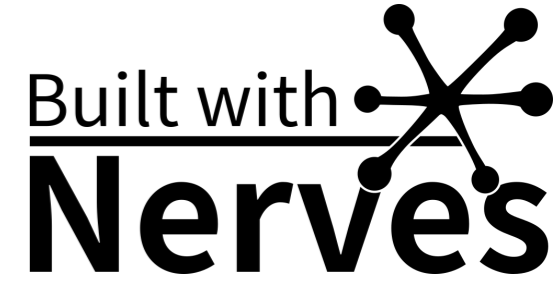
SWEST21/ESS2019 keynote!! の前に,,,



Built with 
Nerves



Elixirとは？



2012年に登場した関数型言語

Erlang VM上で動作

- 高い並列性能を誇る
- 軽量かつ頑強なプロセスモデル
- 耐障害性が極めて高い

Rubyを基にした言語設計

- 習得しやすく生産性が向上する
- WebフレームワークPhoenix
 - 応答性が極めて高い
 - Channel, LiveView





Elixirプログラミング の気持ち良さ



The Elixir **Zen** Style

- **Zen** is the essential beauty.
禅は本質美である
- The essential of programming is data transformation
プログラミングの本質はデータ変換である
- The Elixir **Zen** style is a programming style that pursues data-transformation-based programming.
Elixir **Zen** スタイルはデータ変換プログラミングを追求するスタイルである

```
list = 1..1_000_000 |>  
Enum.to_list  
  
list  
|> Enum.map(&foo(&1))  
|> Enum.map(&bar(&1))
```



Elixir Zen スタイル1: Enum.map+ パイプライン

- 右のコード例

- 1から1,000,000までのリストに

```
1..1_000_000
```

```
|> Enum.map(&M.foo(&1))
```

```
|> Enum.map(&M.bar(&1))
```

- 各要素に foo を適用(x2)
- 各要素に bar を適用(+1)
- した結果を得る

```
defmodule M do
```

```
  def foo( n ), do: n * 2
```

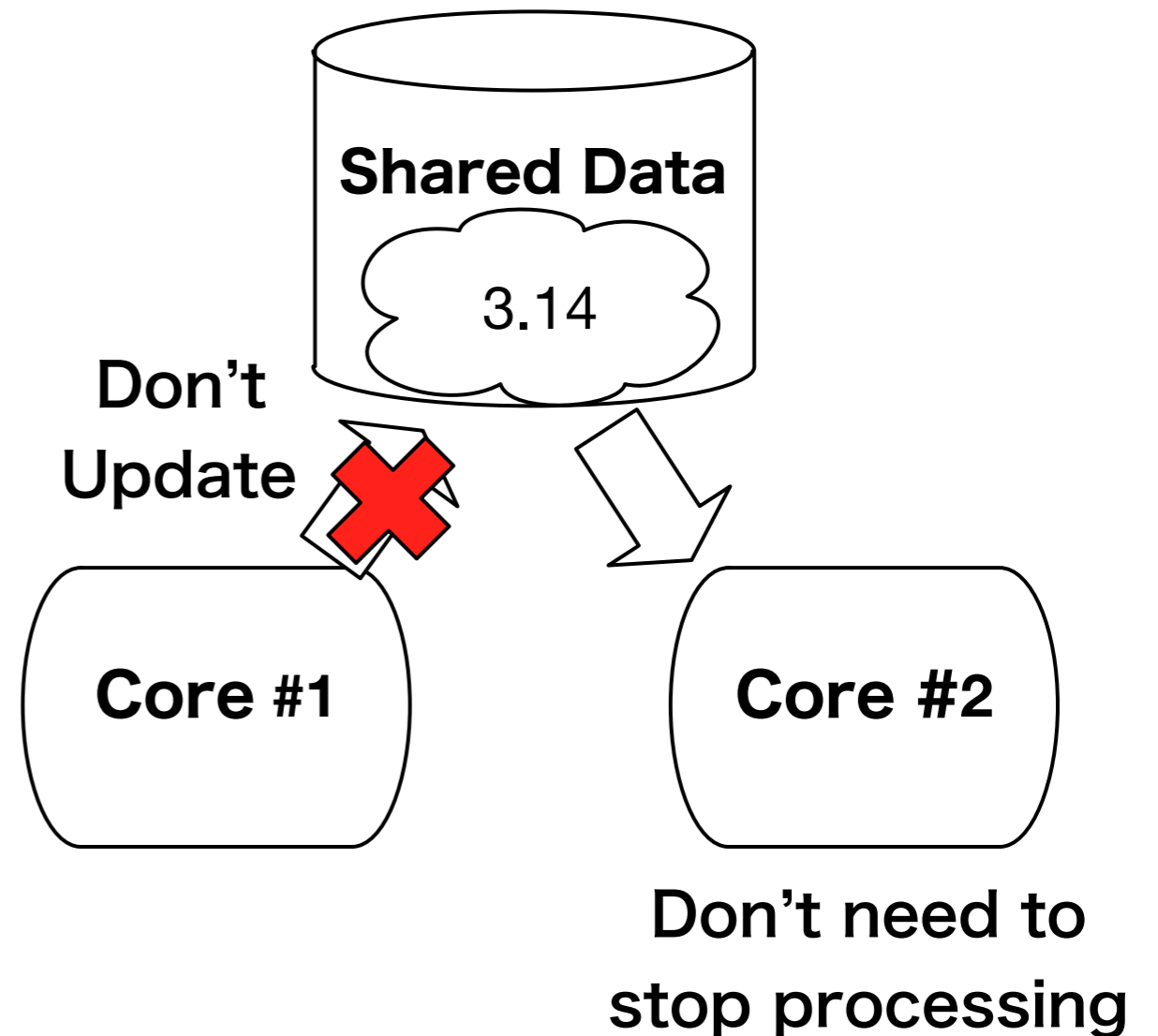
```
  def bar( n ), do: n + 1
```

```
end
```

Elixir は全てがイミュータブルである

```
1..1_000_000  
|> Enum.map(&M.foo(&1))  
|> Enum.map(&M.bar(&1))
```

```
defmodule M do  
  def foo( n ), do: n * 2  
  def bar( n ), do: n + 1  
end
```





Elixir Zen スタイル2: バイナリ+パターンマッチ

- バイナリデコーダを作る場合
 - CPUエミュレータ
 - バイトコードインタプリタ
 - ELFバイナリのダンプ
 - MIDIシーケンサなど
- あるビット列に合致するかどうかで条件分岐するプログラミング
- Elixir ではバイナリ+パターンマッチで美しく書ける！

```
defmodule Midi do
  def decode(<<>>), do: []

  def decode(<<
    8::size(4),
    channel::bitstring-size(4),
    velocity::bitstring-size(8),
    rest::bitstring()
  >>) do
    [{:note_off, channel, velocity}] ++ decode(rest)
  end

  def decode(<<
    9::size(4),
    channel::bitstring-size(4),
    velocity::bitstring-size(8),
    rest::bitstring()
  >>) do
    [{:note_on, channel, velocity}] ++ decode(rest)
  end

  ...
end
```



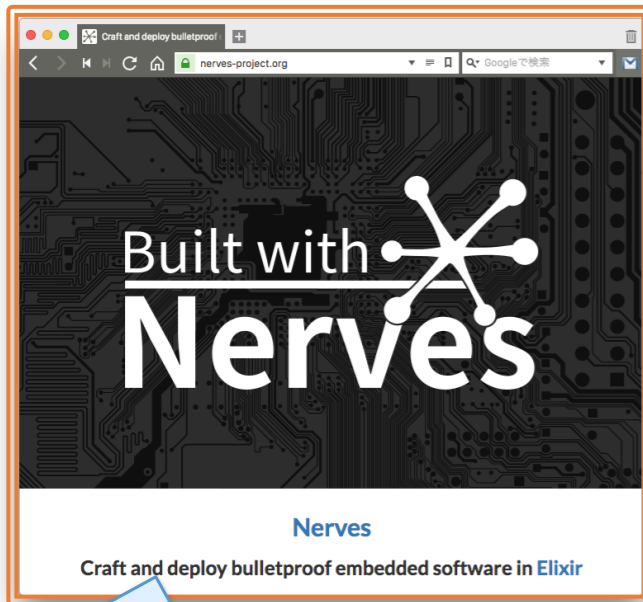

Elixir Zen スタイル3: デバイスドライバプロセス

- 従来だとI/O操作には同期・排他制御が必要だが煩雑
- ElixirではあるデバイスのI/Oを操作するプロセスを1つに集約 (言わばデバイスドライバ)
- 他のプロセスからI/Oを操作したい時にはプロセス間通信でメッセージを送る
- メッセージはデバイスドライバに入る前に直列化されるので、デバイスドライバの中で同期・排他制御をしなくていい！
- この仕組みを抽象化したモデルがアクターモデル (Actor model)
- デバイスドライバはPortやNIFを使って、C言語などの他言語を組み込める
- 詳しくは Justin の基調講演で図示・解説があるので、お楽しみに！



ElixirでIoT!!

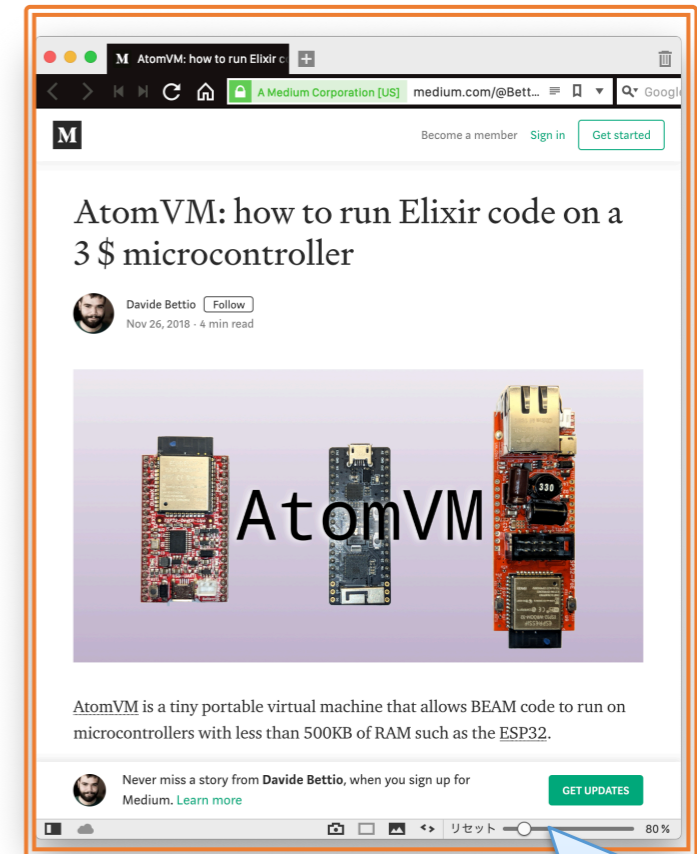
Built with 
Nerves



- ラズパイ等が主な対象
- メモリサイズ数10MB!
- Elixir ecosystem連携!



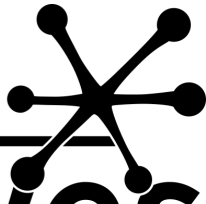
- HWもSWもErlangセットで提供!!
 - ✓ 300MHz Cortex-M7 & 64MB Flash
 - ✓ On-board WLAN & Pmodコネクタ
- V2の[Kickstarter Project](#)達成!!



- ESP32/STM32でElixir/Erlangが動く!!
- ファームは約700KB!!
- 機能実装はまだ限定的



Nerves!!

Built with 
Nerves

<https://nerves-project.org/>

Craft and deploy bulletproof embedded software in **Elixir**

Platform

Pack your whole application into as little as 12MB and have it start in seconds by booting a lean cross-compiled Linux directly to the battle-hardened Erlang VM.

Framework

Let Nerves take care of the network, discovery, I/O, firmware updates and more. Focus on what matters and have fun writing robust and maintainable software.

Tooling

Go from "mix new" to running code on your device in minutes. From cross-compilation to remote device access, our tools got you covered.

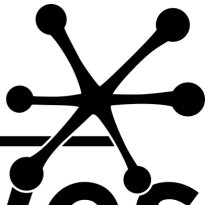
Nerves is young, but already powers rock-solid shipping industrial products! Check us out if you are a hearty experimenter and interested in a new way of creating embedded systems.

Nerves is also fully open source with the majority of code licensed under [Apache 2.0](#). Nerves does make use of non-Apache 2.0 code such as the Linux kernel and Buildroot. Licensing details of non-Elixir components is provided via Buildroot tooling. Licensing details for Elixir components is available through [hex.pm](#).





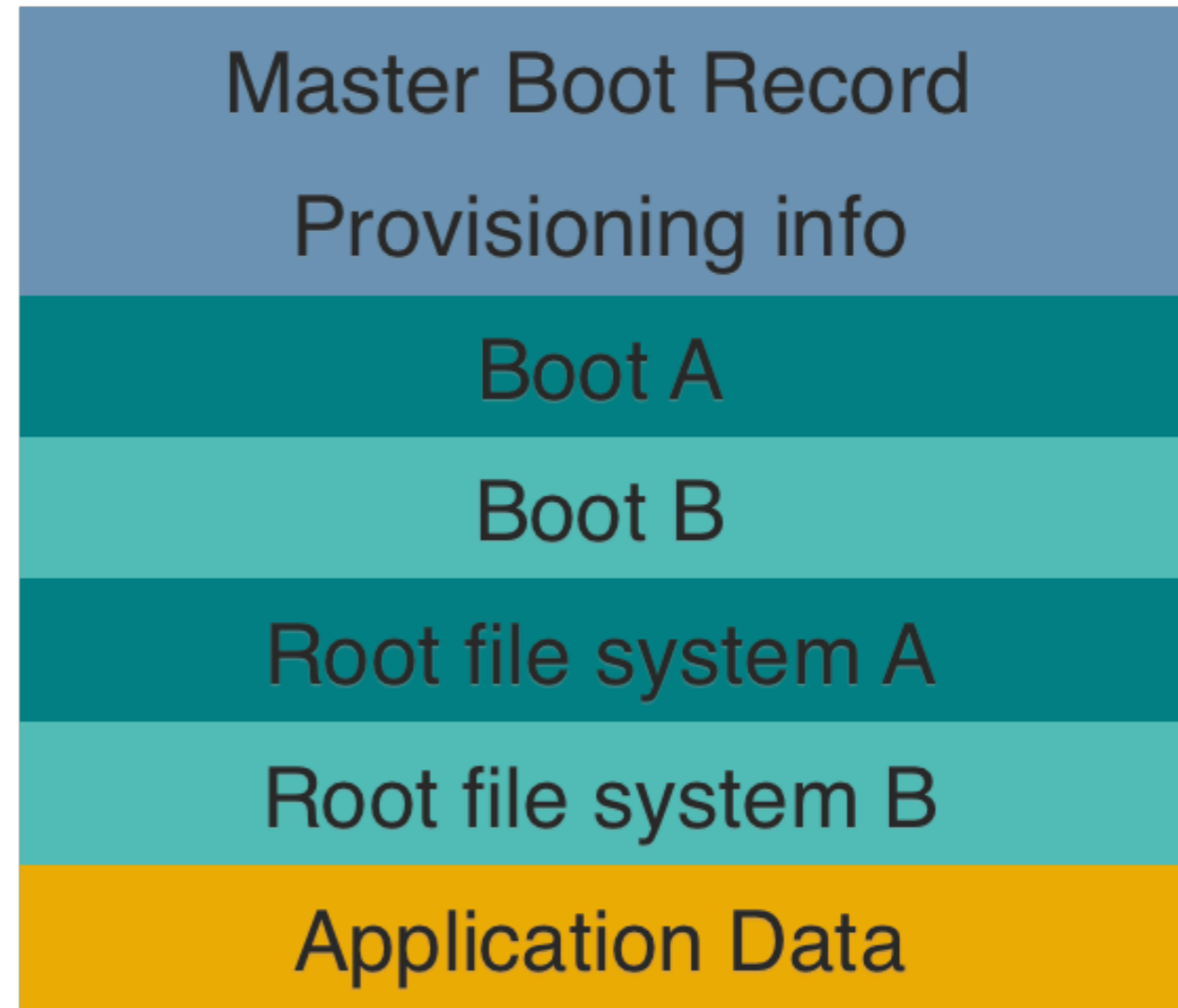
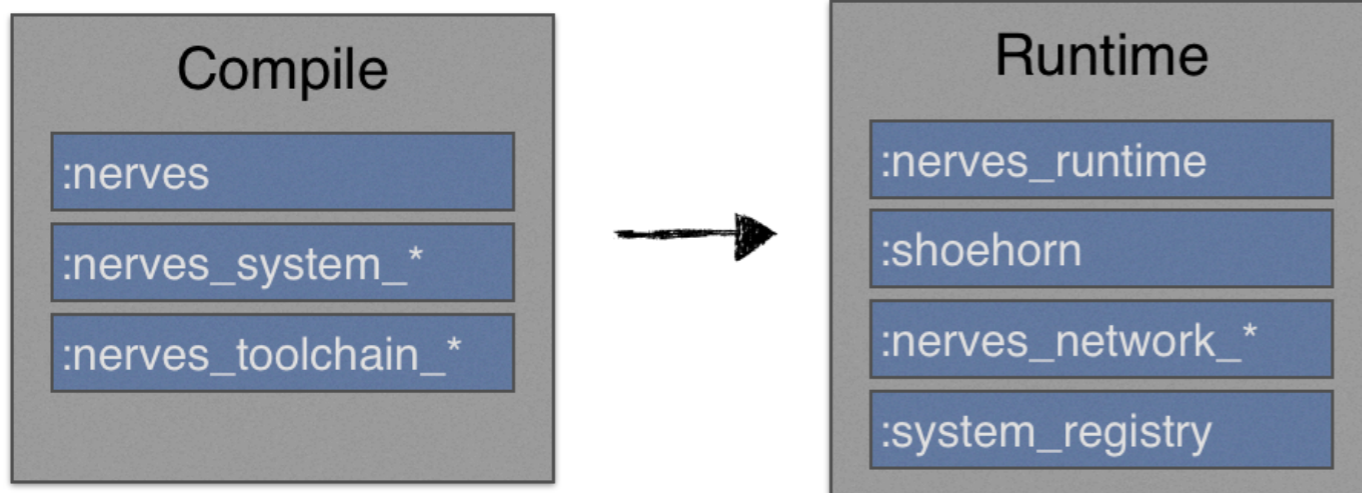
Platform

Built with 
Nerves

- ブートローダ + rootfs + Erlang OTP + Elixirアプリ
—括でビルドして
SDカード等へ書き込み

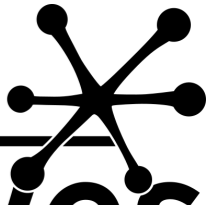


nerves_system_*
~30 MB

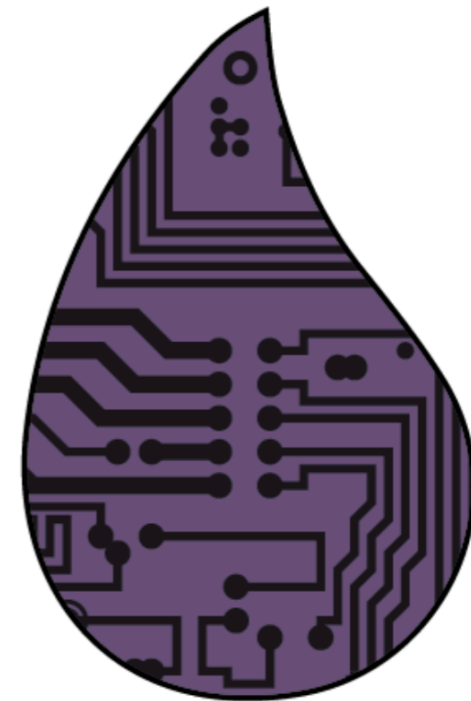




Framework

Built with 
Nerves

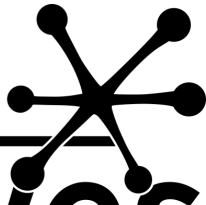
- IoTデバイス開発に欠かせない強力なライブラリ
 - Elixir Circuits: GPIO, I2C, SPI, UART
 - IoTモジュール向けライブラリも
- 通常のElixirライブラリも利用可
 - 描画系ライブラリ Scenic や Webフレームワーク Phoenixとも容易に連携可能！



Elixir Circuits



Tooling

Built with 
Nerves

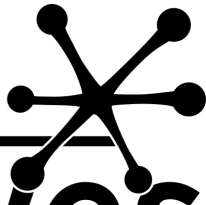
- mixによるプロジェクト管理
 - いつものElixirアプリ開発と一緒に!
 - クロスコンパイルは裏でよしなに
- IExでのインタプリタ実行も可能
 - VirtualEther越しのssh接続
 - /dev/tty* 越しのserial接続
- ファーム・アプリの書き込み
 - SDカードに書き込み
 - VirtualEther越しの更新
 - NervesHub** : [Device to the Cloud!!](#)

```
takase@MacBookPro ~  
$ ssh nerves.local  
Warning: Permanently added 'nerves.local,172.25.58.85' (RSA) to the list of  
known hosts.  
Interactive Elixir (1.8.1) - press Ctrl+C to exit (type h() ENTER for help)  
Toolshed imported. Run h(Toolshed) for more info  
RingLogger is collecting log messages from Elixir and Linux. To see the  
messages, either attach the current IEx session to the logger:  
  
RingLogger.attach  
  
or print the next messages in the log:  
  
RingLogger.next  
  
iex(starter@nerves.local)1> ** (EXIT) interrupted  
iex(starter@nerves.local)1> ** (EXIT) interrupted  
iex(starter@nerves.local)1> ** (EXIT) interrupted  
iex(starter@nerves.local)1> exit  
Connection to nerves.local closed.  
  
takase@MacBookPro ~  
$
```

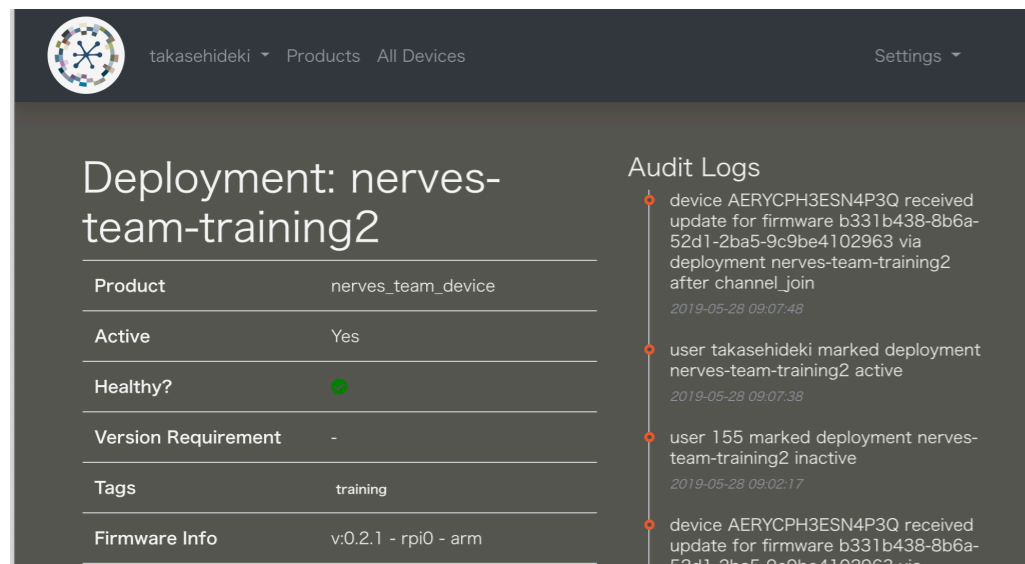




NervesHub

Built with 
Nerves

- サーバ経由のOTA (Over The Air) で Nervesアプリを**リモートデプロイ**！
 - X.509署名証明書とNervesKey回路でセキュアな接続経路を実現
 - 更新先とファームを任意指定可

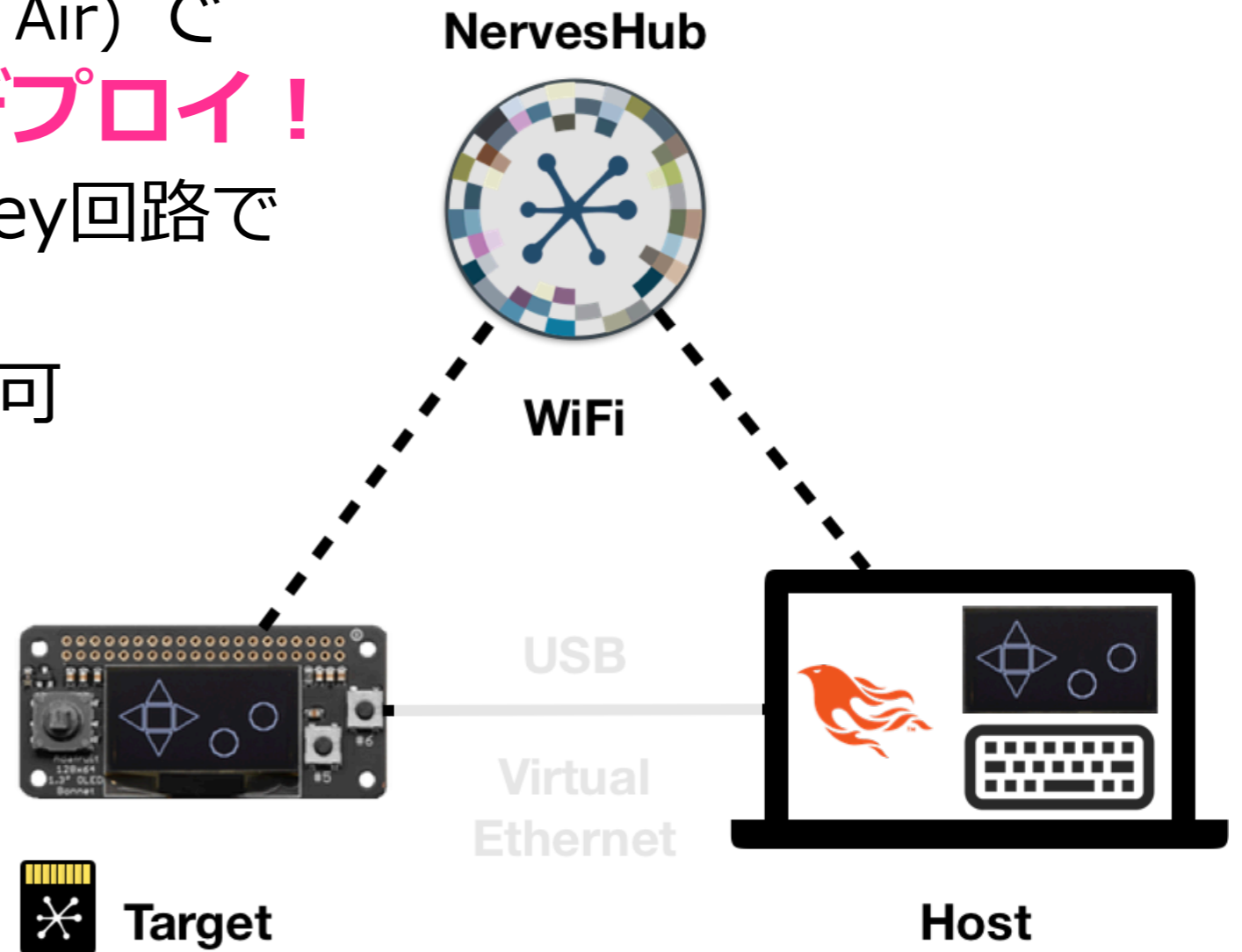


Deployment: nerves-team-training2

Product	nerves_team_device
Active	Yes
Healthy?	●
Version Requirement	-
Tags	training
Firmware Info	v:0.2.1 - rpi0 - arm

Audit Logs

- device AERYCPH3ESN4P3Q received update for firmware b331b438-8b6a-52d1-2ba5-9c9be4102963 via deployment nerves-team-training2 after channel_join
2019-05-28 09:07:48
- user takashideki marked deployment nerves-team-training2 active
2019-05-28 09:07:38
- user 155 marked deployment nerves-team-training2 inactive
2019-05-28 09:02:17
- device AERYCPH3ESN4P3Q received update for firmware b331b438-8b6a-52d1-2ba5-9c9be4102963 via



SWEST21/ESS2019 keynote!! の前に,,,



Built with 
Nerves