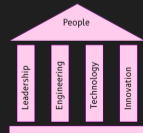


# LONGTERM

The terminal you have been longing for

Daniel Masłowski

IAOTAI



# Agenda



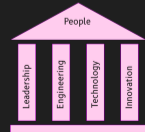
Terminals and Consoles



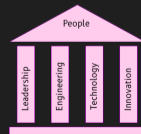
Protocols and Platforms



LONGTERM

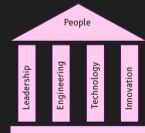


# Terminals and Consoles



# Computers

... started with research.



# Computers

... started with research.

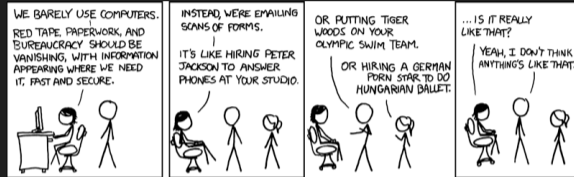
focus on bureaucracy



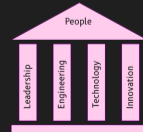
forms, files, folders

business processes

administration



[https://imgs.xkcd.com/ibm/ibm\\_hc\\_3.png](https://imgs.xkcd.com/ibm/ibm_hc_3.png)



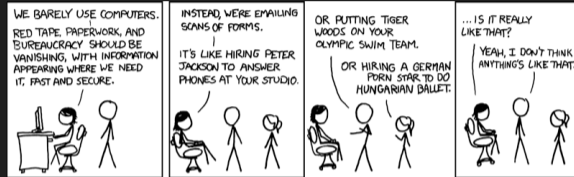
# Computers

... started with research.

focus on bureaucracy



- forms, files, folders
- business processes
- administration



[https://imgs.xkcd.com/ibm/ibm\\_hc\\_3.png](https://imgs.xkcd.com/ibm/ibm_hc_3.png)

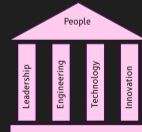
## Applications



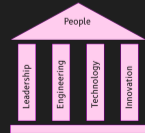
- military
- communication
- entertainment
- gadgets
- ecommerce



- data processing
- ...SaaSaaS
- citizen services
- medicine
- surveillance

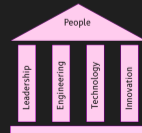


# On Interfaces



# On Interfaces

Simple? Easy to create != easy to consume!

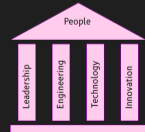




# On Interfaces

Simple? Easy to create != easy to consume!

“API” - often not application, not even programming



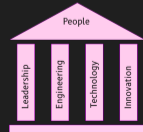
# On Interfaces

Simple? Easy to create != easy to consume!

“API” - often not application, not even programming

```
echo 1 > foo
```

(nothing exciting)



# On Interfaces

Simple? Easy to create != easy to consume!

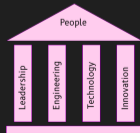
“API” - often not application, not even programming

```
echo 1 > foo
```

(nothing exciting)

```
echo 1 > /dev/led
```

Huh? Do I write a “1” on a physical file to turn on light?



# On Interfaces

Simple? Easy to create != easy to consume!

“API” - often not application, not even programming

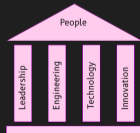
```
echo 1 > foo
```

(nothing exciting)

```
echo 1 > /dev/led
```

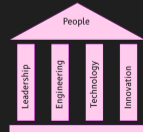
Huh? Do I write a “1” on a physical file to turn on light?

```
vim /dev/null
```



# CLI / POSIX

---



# CLI / POSIX

## wget vs curl



“do one thing and do it well”?



write to stdout vs current dir

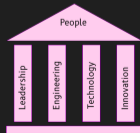


-o vs -O to specify output file

---

<sup>1</sup>https:

[//pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd\\_chap12.html#tag\\_12\\_02](https://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap12.html#tag_12_02)



# CLI / POSIX

wget vs curl



“do one thing and do it well”?



write to stdout vs current dir



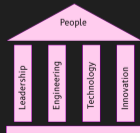
-o vs -O to specify output file

What is the meaning of a single -<sup>1</sup>?

---

<sup>1</sup>https:

[//pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd\\_chap12.html#tag\\_12\\_02](https://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap12.html#tag_12_02)



# CLI / POSIX

## wget vs curl



“do one thing and do it well”?



write to stdout vs current dir



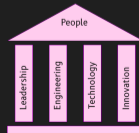
-o vs -O to specify output file

## What is the meaning of a single -<sup>1</sup>?

*Guideline 13: For utilities that use operands to represent files to be opened for either reading or writing, the '-' operand should be used only to mean standard input (or standard output when it is clear from context that an output file is being specified).*

---

<sup>1</sup>[https:](https://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap12.html#tag_12_02)





# CLI / POSIX

## wget vs curl



“do one thing and do it well”?



write to stdout vs current dir



-o vs -O to specify output file

## What is the meaning of a single `-`<sup>1</sup>?

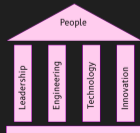
*Guideline 13: For utilities that use operands to represent files to be opened for either reading or writing, the ‘-’ operand should be used only to mean standard input (or standard output when it is clear from context that an output file is being specified).*

Does `wget` take a URL from stdin or write data to stdout?

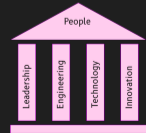
---

<sup>1</sup>[https:](https://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap12.html#tag_12_02)

[//pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd\\_chap12.html#tag\\_12\\_02](https://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap12.html#tag_12_02)

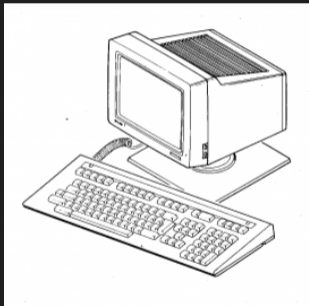


# Getting Physical

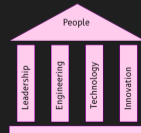


# Getting Physical

Oldschool

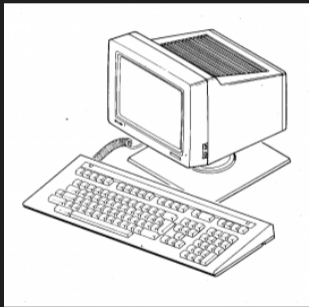


<sup>a</sup>[https://terminals-wiki.org/wiki/index.php/DEC\\_VT330](https://terminals-wiki.org/wiki/index.php/DEC_VT330)



# Getting Physical

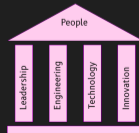
## Oldschool



<sup>a</sup>[https://terminals-wiki.org/wiki/index.php/DEC\\_VT330](https://terminals-wiki.org/wiki/index.php/DEC_VT330)

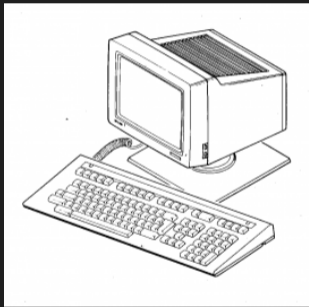
## Terminal

- 🐼 the machine terminating the connection to a “computer” :-)
- 🐼 may connect to multiple computers
- 🐼 back in the days: serial  
today: gigabit ethernet
- 🐼 we SSH to multiple machines nowadays in a *terminal emulator*



# Getting Physical

## Oldschool



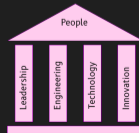
<sup>a</sup>[https://terminals-wiki.org/wiki/index.php/DEC\\_VT330](https://terminals-wiki.org/wiki/index.php/DEC_VT330)

## Terminal

- 🐣 the machine terminating the connection to a “computer” :-)
- 🐣 may connect to multiple computers
- 🐣 back in the days: serial  
today: gigabit ethernet
- 🐣 we SSH to multiple machines nowadays in a *terminal emulator*

## Console

- 🐣 the physical interface, aka HMI (Human Machine Interface)
- 🐣 a physical terminal



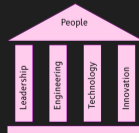
# Unix and Plan 9<sup>4</sup> From Bell Labs

---

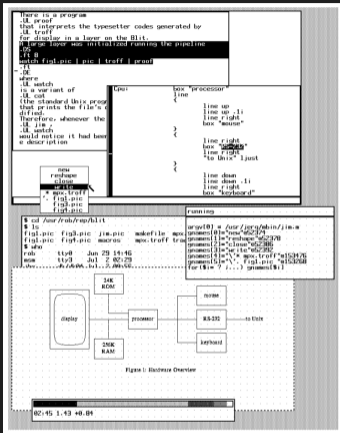
<sup>2</sup>[https://doc.cat-v.org/bell\\_labs/blit/](https://doc.cat-v.org/bell_labs/blit/)

<sup>3</sup><https://web.archive.org/web/20151022021014/http://homeostasis.scs.carleton.ca/~soma/distos/2014w/presotto-plan9.pdf>

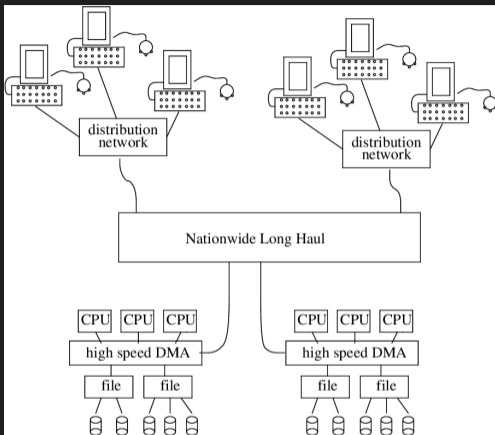
<sup>4</sup>[https://doc.cat-v.org/plan\\_9/1st\\_edition/designing\\_plan\\_9](https://doc.cat-v.org/plan_9/1st_edition/designing_plan_9)



# Unix and Plan 9<sup>4</sup> From Bell Labs



Unix and Blit

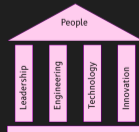


Plan 9 and Gnot

<sup>2</sup> [https://doc.cat-v.org/bell\\_labs/blit/](https://doc.cat-v.org/bell_labs/blit/)

<sup>3</sup> <https://web.archive.org/web/20151022021014/http://homeostasis.scs.carleton.ca/~soma/distos/2014w/presotto-plan9.pdf>

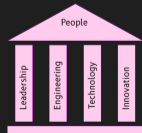
<sup>4</sup> [https://doc.cat-v.org/plan\\_9/1st\\_edition/designing\\_plan\\_9](https://doc.cat-v.org/plan_9/1st_edition/designing_plan_9)



# Plan 9 from Bell Labs<sup>7</sup>?

---

<sup>5</sup><https://css.csail.mit.edu/6.824/2014/papers/plan9.pdf>





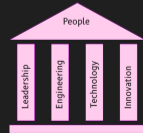
# Plan 9 from Bell Labs<sup>7</sup>?

*The early catch phrase was to build a UNIX out of a lot of little systems, not a system out of a lot of little UNIXes. The problems with UNIX were too deep to fix, but some of its ideas could be brought along.<sup>5</sup>*

---

<sup>5</sup><https://css.csail.mit.edu/6.824/2014/papers/plan9.pdf>

<sup>6</sup><https://www.tuhs.org/Archive/Documentation/AUUGN/AUUGN-V12.1.pdf>



# Plan 9 from Bell Labs<sup>7</sup>?

*The early catch phrase was to build a UNIX out of a lot of little systems, not a system out of a lot of little UNIXes. The problems with UNIX were too deep to fix, but some of its ideas could be brought along.*<sup>5</sup>

## Legacy

Early papers can be found in archives<sup>6</sup>.

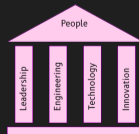
Sources have been released and relicensed: <https://9p.io/plan9/>

---

<sup>5</sup><https://css.csail.mit.edu/6.824/2014/papers/plan9.pdf>

<sup>6</sup><https://www.tuhs.org/Archive/Documentation/AUUGN/AUUGN-V12.1.pdf>

<sup>7</sup>[https://en.wikipedia.org/wiki/Plan\\_9\\_from\\_Bell\\_Labs](https://en.wikipedia.org/wiki/Plan_9_from_Bell_Labs)



# Plan 9 from Bell Labs<sup>7</sup>?

*The early catch phrase was to build a UNIX out of a lot of little systems, not a system out of a lot of little UNIXes. The problems with UNIX were too deep to fix, but some of its ideas could be brought along.*<sup>5</sup>

## Legacy

Early papers can be found in archives<sup>6</sup>.

Sources have been released and relicensed: <https://9p.io/plan9/>

## Today

Plan 9 Foundation now owns the rights: <https://p9f.org/>

The *International Workshop on Plan 9* continues: <https://iwp9.org>

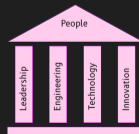
IWP9 2025 will be held in Paris.

---

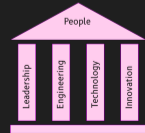
<sup>5</sup><https://css.csail.mit.edu/6.824/2014/papers/plan9.pdf>

<sup>6</sup><https://www.tuhs.org/Archive/Documentation/AUUGN/AUUGN-V12.1.pdf>

<sup>7</sup>[https://en.wikipedia.org/wiki/Plan\\_9\\_from\\_Bell\\_Labs](https://en.wikipedia.org/wiki/Plan_9_from_Bell_Labs)

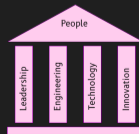


# Physical Today



# Physical Today

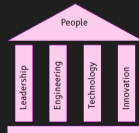
Are you debugging SBCs with extra screens because ... HDMI ... ?



# Physical Today

Are you debugging SBCs with extra screens because ... HDMI ... ?

Do you still see TTYs... *virtual* TTYs, and if so, why in 2024?

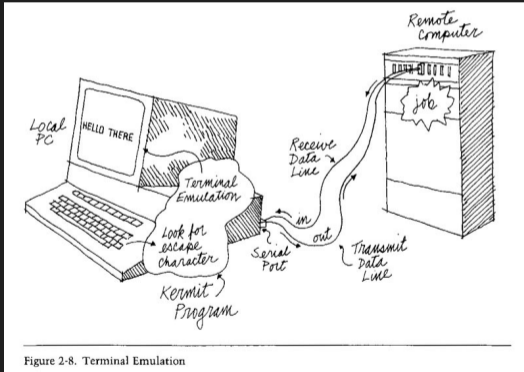


# Physical Today

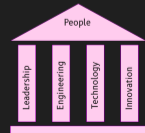
Are you debugging SBCs with extra screens because ... HDMI ... ?

Do you still see TTYs... *virtual* TTYs, and if so, why in 2024?

Have you heard of Kermit? It is still being developed...



<https://www.kermitproject.org/diagram.html>



# Nice Hardware

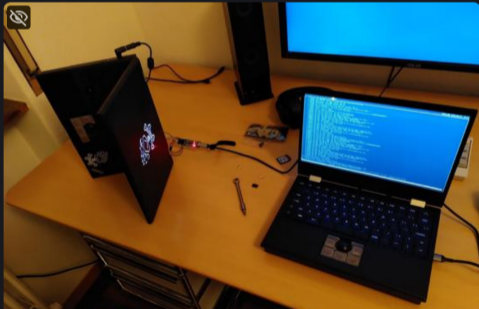


Niklaus 'vimja' Hofer  
@vimja@tooting.ch

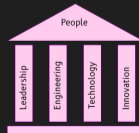
Once again I have one of my @mntmn reforms hooked up via serial.

This time I'm trying to get Gentoo running on the i.mx8mplus.

After an initial failed attempt, I can now boot reliably. Unfortunayely though, the keyboard is not accepting any input. Probably an initrd issue? Will have to do some more tinkering some time...



<sup>a</sup> <https://tooting.ch/@vimja/112668406067498207>





# Nice Hardware

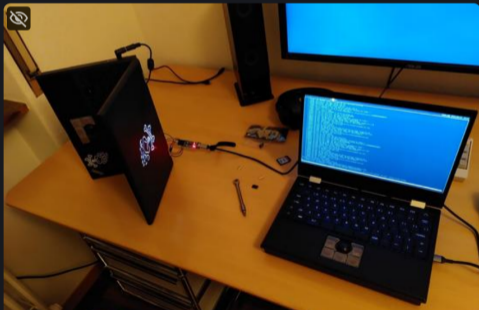


Niklaus 'vimja' Hofer  
@vimja@tooting.ch

Once again I have one of my @mntmn reforms hooked up via serial.

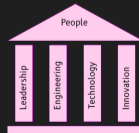
This time I'm trying to get Gentoo running on the i.mx8mplus.

After an initial failed attempt, I can now boot reliably. Unfortunayely though, the keyboard is not accepting any input. Probably an initrd issue? Will have to do some more tinkering some time...



<sup>a</sup><https://tooting.ch/@vimja/112668406067498207>

How many computers  
can you still connect via  
serial these days?



# Nice Hardware

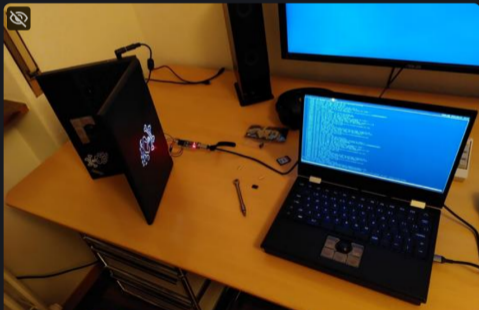


Niklaus 'vimja' Hofer  
@vimja@tooting.ch

Once again I have one of my @mntmn reforms hooked up via serial.

This time I'm trying to get Gentoo running on the i.mx8mplus.

After an initial failed attempt, I can now boot reliably. Unfortunayely though, the keyboard is not accepting any input. Probably an initrd issue? Will have to do some more tinkering some time...



<sup>a</sup> <https://tooting.ch/@vimja/112668406067498207>

How many computers  
can you still connect via  
serial these days?

## Observation

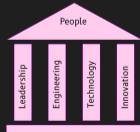
We have fancy TUI app  
frameworks nowadays



ratatui (Rust)



Bubble Tea (Go)



# Nice Hardware



Niklaus 'vimja' Hofer  
@vimja@tooting.ch

Once again I have one of my @mntmn reforms hooked up via serial.

This time I'm trying to get Gentoo running on the i.mx8mplus.

After an initial failed attempt, I can now boot reliably. Unfortunayely though, the keyboard is not accepting any input. Probably an initrd issue? Will have to do some more tinkering some time...



<sup>a</sup> <https://tooting.ch/@vimja/112668406067498207>

How many computers can you still connect via serial these days?

## Observation

We have fancy TUI app frameworks nowadays



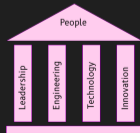
ratatui (Rust)



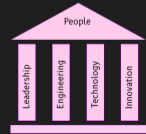
Bubble Tea (Go)

## Question

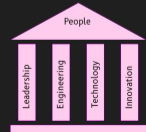
Can we serve rich apps via serial and keep SBCs headless?



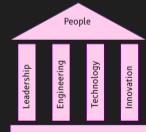
# Internet^WNetwork of Things



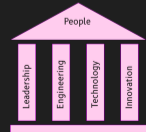
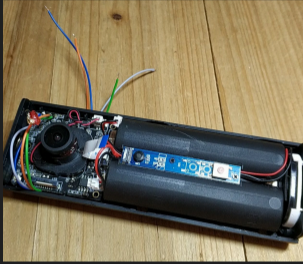
# Internet^WNetwork of Things



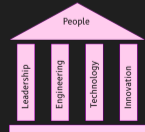
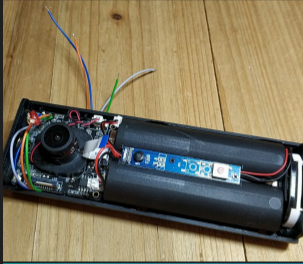
# Internet^WNetwork of Things



# Internet^WNetwork of Things

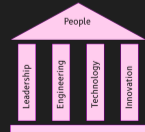
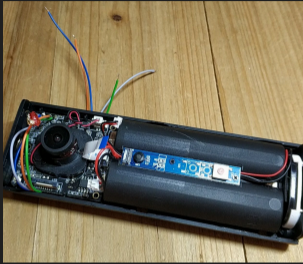


# Internet^WNetwork of Things

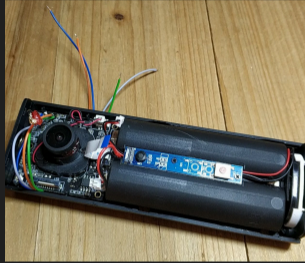




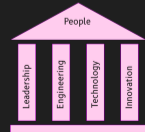
# Internet^WNetwork of Things



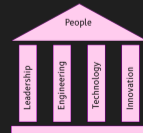
# Internet^WNetwork of Things



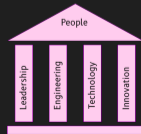
NoT  
Network  
of Things



# Protocols and Platforms



centre and cpu



# centre and cpu

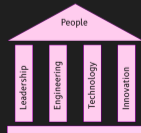
centre

from Harvey OS

serves DHCP+TFTP+HTTP

<https://github.com/harvey-os/go>

cmd/centre



# centre and cpu

centre

from Harvey OS

serves DHCP+TFTP+HTTP

<https://github.com/harvey-os/go>

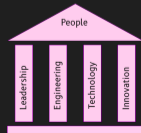
cmd/centre

cpu

the other one

works on Linux

<https://github.com/u-root/cpu>



# centre and cpu

centre

from Harvey OS

serves DHCP+TFTP+HTTP

<https://github.com/harvey-os/go>

cmd/centre

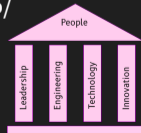
cpu

the other one

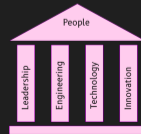
works on Linux

<https://github.com/u-root/cpu>

<https://chemnitzer.linux-tage.de/2022/en/programm/beitrag/226/>  
*Drivers From Outer Space - Fast, Simple Driver Development*

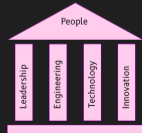


DEMO



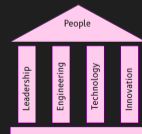


# The Web



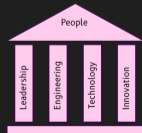
# The Web

The thing that started as a platform for documents...



# The Web

The thing that started as a platform for documents...  
... has turned into a platform for applications.

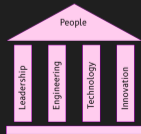


# The Web

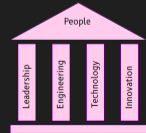
The thing that started as a platform for documents...

... has turned into a platform for applications.

Let's embrace it and push it further down!

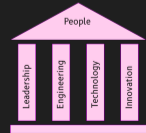


# WebGAP (Web General Application Protocol)



# WebGAP (Web General Application Protocol)

**NOTE:** There is no implementation yet. This is a concept.

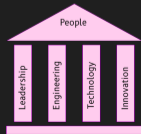


# WebGAP (Web General Application Protocol)

**NOTE:** There is no implementation yet. This is a concept.

## Idea

Deploy Wasm code as apps to any gadget, such as a wristband, a camera, a doorbell, a fridge...



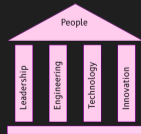
# WebGAP (Web General Application Protocol)

**NOTE:** There is no implementation yet. This is a concept.

## Idea

Deploy Wasm code as apps to any gadget, such as a wristband, a camera, a doorbell, a fridge...

Let that serve back apps to consume the gadget again.





# WebGAP (Web General Application Protocol)

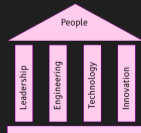
**NOTE:** There is no implementation yet. This is a concept.

## Idea

Deploy Wasm code as apps to any gadget, such as a wristband, a camera, a doorbell, a fridge...

Let that serve back apps to consume the gadget again.

Make use of existing runtime environments, such as a web browser or desktop, draw simple graphics, similar to DEC ReGIS, take it further.



# WebGAP (Web General Application Protocol)

**NOTE:** There is no implementation yet. This is a concept.

## Idea

Deploy Wasm code as apps to any gadget, such as a wristband, a camera, a doorbell, a fridge...

Let that serve back apps to consume the gadget again.

Make use of existing runtime environments, such as a web browser or desktop, draw simple graphics, similar to DEC ReGIS, take it further.

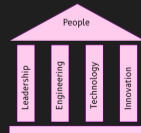


WebAssembly is widely available and used in all sorts of scenarios

WebSerial lets you use a serial port from the web browser

define (app specific) protocols

can also be used in a simpler environment



# WebGAP (Web General Application Protocol)

**NOTE:** There is no implementation yet. This is a concept.

## Idea

Deploy Wasm code as apps to any gadget, such as a wristband, a camera, a doorbell, a fridge...

Let that serve back apps to consume the gadget again.

Make use of existing runtime environments, such as a web browser or desktop, draw simple graphics, similar to DEC ReGIS, take it further.



WebAssembly is widely available and used in all sorts of scenarios

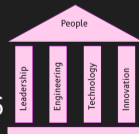
WebSerial lets you use a serial port from the web browser

define (app specific) protocols

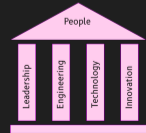
can also be used in a simpler environment

Have you heard of “Wasmlet”s?

[https://twitter.com/wasm3\\_engine/status/1465294919422119936](https://twitter.com/wasm3_engine/status/1465294919422119936)

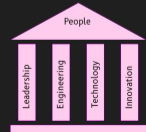


# WebALE (Web Application Launch Environment)



# WebALE (Web Application Launch Environment)

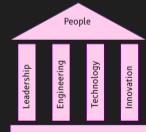
You know the app launcher from your phone.



# WebALE (Web Application Launch Environment)

You know the app launcher from your phone.

Desktop icons have always been similar.

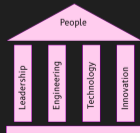


# WebALE (Web Application Launch Environment)

You know the app launcher from your phone.

Desktop icons have always been similar.

How about drawing the launcher, just like `drawterm`?



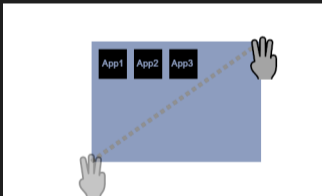
# WebALE (Web Application Launch Environment)

You know the app launcher from your phone.

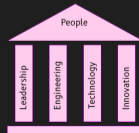
Desktop icons have always been similar.

How about drawing the launcher, just like `drawterm`?

Touch friendly variant: draw a diagonal with 3 fingers.



Add an intermediate step for establishing a connection to and dropping your initial app onto a remote gadget.

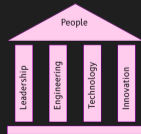




# Platform System Interface

The *Platform System Interface* project (PSI) is a collection of design ideas, specifications, tools and other resources all around hardware platforms, firmware, bootloaders, OS interfacing and user experience.

<https://github.com/platform-system-interface>



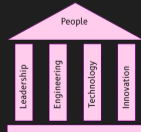
# Platform System Interface

The *Platform System Interface* project (PSI) is a collection of design ideas, specifications, tools and other resources all around hardware platforms, firmware, bootloaders, OS interfacing and user experience.

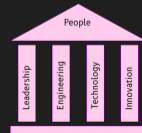
<https://github.com/platform-system-interface>

Talk: Platform System Interface - Design and Evaluation of Computing as a Whole

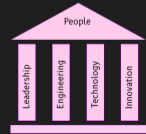
in-depth discussion of design paradigms and complexity in computing  
<https://metaspora.org/platform-system-interface-computing-as-a-whole.pdf>



LONGTERM

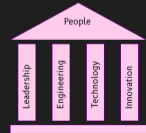


The terminal you have been longing for



# The terminal you have been longing for

What would a terminal in 2024 look like? What features would it have?

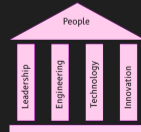


# The terminal you have been longing for

What would a terminal in 2024 look like? What features would it have?

## Hardware / IO

- 👤 touch screen, keyboard, touchpad
- 👤 GPIO ports
- 👤 audio ports/switch
- 👤 ethernet (switch)
- 👤 multiple serial ports
- 👤 HDMI (switch) + input/capture
- 👤 USB hub
- 👤 switch buttons
- 👤 glorified KVM switch? with a serial switch!



# The terminal you have been longing for

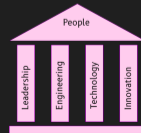
What would a terminal in 2024 look like? What features would it have?

## Hardware / IO

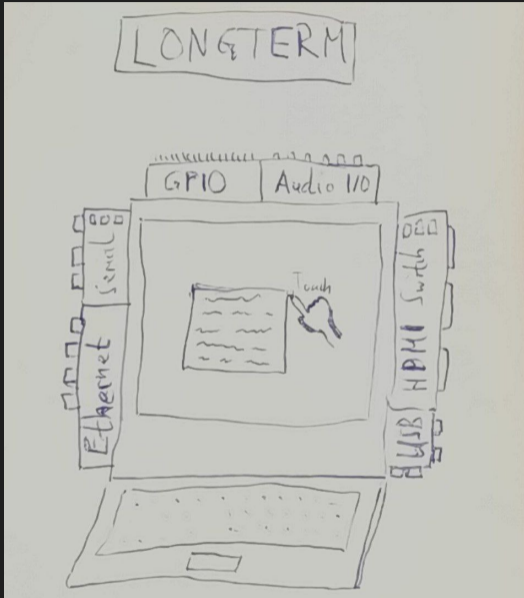
- 👤 touch screen, keyboard, touchpad
- 👤 GPIO ports
- 👤 audio ports/switch
- 👤 ethernet (switch)
- 👤 multiple serial ports
- 👤 HDMI (switch) + input/capture
- 👤 USB hub
- 👤 switch buttons
- 👤 glorified KVM switch?
- 👤 with a serial switch!

## Platform / Design

- 👤 graphical desktop + web browser
- 👤 web native desktop?
- 👤 built-in tools
  - ▶ centre
  - ▶ cpu
  - ▶ drawterm
- 👤 show serial console of attached device by default?
- 👤 protocols
  - ▶ XMODEM
  - ▶ Kermit
  - ▶ ReGIS
  - ▶ WebGAP

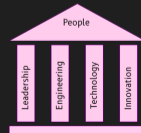


# Sketch and Logo



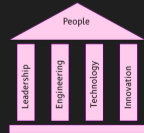
# LONG TERM

accent color: #E68E41



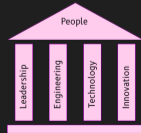


What if...



What if...

... we compile Rust...

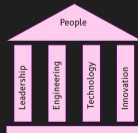


What if...

... we compile Rust...



... to Wasm...



What if...

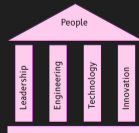
... we compile Rust...



... to Wasm...



... and use it in an app?



What if...

... we compile Rust...



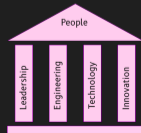
... to Wasm...



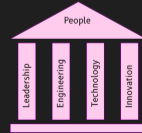
... and use it in an app?



Magic happens - we can use native code on web platforms!



# Howto

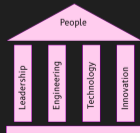


# Howto

## Getting started

<https://lannonbr.com/blog/2020-01-07-rust-wasmpack/>

<https://rustwasm.github.io/docs/wasm-pack/>



# Howto

## Getting started

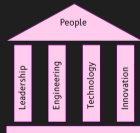
<https://lannonbr.com/blog/2020-01-07-rust-wasmpack/>

<https://rustwasm.github.io/docs/wasm-pack/>

## TL;DR

```
cargo install wasm-pack
```

```
wasm-pack new my-rust-wasm-foo
```





# Howto

## Getting started

<https://lannonbr.com/blog/2020-01-07-rust-wasmpack/>

<https://rustwasm.github.io/docs/wasm-pack/>

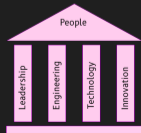
## TL;DR

```
cargo install wasm-pack  
wasm-pack new my-rust-wasm-foo
```

## The glue

<https://github.com/wasm-tool/wasm-pack-plugin>

<https://rustwasm.github.io/docs/wasm-pack/tutorials/hybrid-applications-with-webpack/using-your-library.html>



# Howto

## Getting started

<https://lannonbr.com/blog/2020-01-07-rust-wasmpack/>

<https://rustwasm.github.io/docs/wasm-pack/>

## TL;DR

```
cargo install wasm-pack  
wasm-pack new my-rust-wasm-foo
```

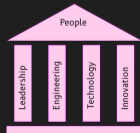
## The glue

<https://github.com/wasm-tool/wasm-pack-plugin>

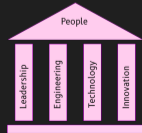
<https://rustwasm.github.io/docs/wasm-pack/tutorials/hybrid-applications-with-webpack/using-your-library.html>

## More glue

```
cargo add gloo-utils
```



# The Rust side



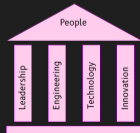
# The Rust side

```
extern crate wasm_bindgen;
use gloo_utils::format::JsValueSerdeExt;
use serde::{Deserialize, Serialize};
use wasm_bindgen::prelude::*;

/// ...

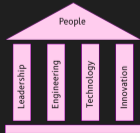
#[derive(Serialize, Deserialize)]
struct Foo {
    bar: u32,
    baz: String,
}

#[wasm_bindgen]
pub fn some_fun(data: JsValue) -> JsValue {
    /// ...
    let foo = Foo::new { bar: 42, baz: "Rust Wasm" };
    JsValue::from_serde(&foo).unwrap()
}
```



# The JavaScript side

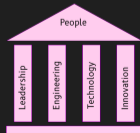
```
import { some_fun } from "./rs/pkg";  
  
/* ... */  
  const res = some_fun({ woopWoop: 1337 });  
  console.info(res);  
/* ... */
```



# The JavaScript side

```
import { some_fun } from "./rs/pkg";  
  
/* ... */  
const res = some_fun({ woopWoop: 1337 });  
console.info(res);  
/* ... */
```

But that is synchronous and blocking!



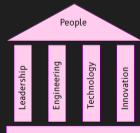
# The JavaScript side

```
import { some_fun } from "./rs/pkg";  
  
/* ... */  
const res = some_fun({ woopWoop: 1337 });  
console.info(res);  
/* ... */
```

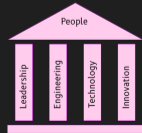
But that is synchronous and blocking!

<https://rustwasm.github.io/wasm-bindgen/reference/js-promises-and-rust-futures.html>

[https://rustwasm.github.io/wasm-bindgen/api/wasm\\_bindgen\\_futures/](https://rustwasm.github.io/wasm-bindgen/api/wasm_bindgen_futures/)



Thanks! :)





# Follow Me



Daniel  
Maslowski

<https://github.com/orangecms>  
<https://twitter.com/orangecms>  
<https://mastodon.social/@cyrevolt>  
<https://twitch.tv/cyrevolt>  
<https://youtube.com/@cyrevolt>

<https://github.com/platform-system-interface>

<https://metaspora.org/longterm.pdf>

License: CC BY 4.0 <https://creativecommons.org/licenses/by/4.0/>

