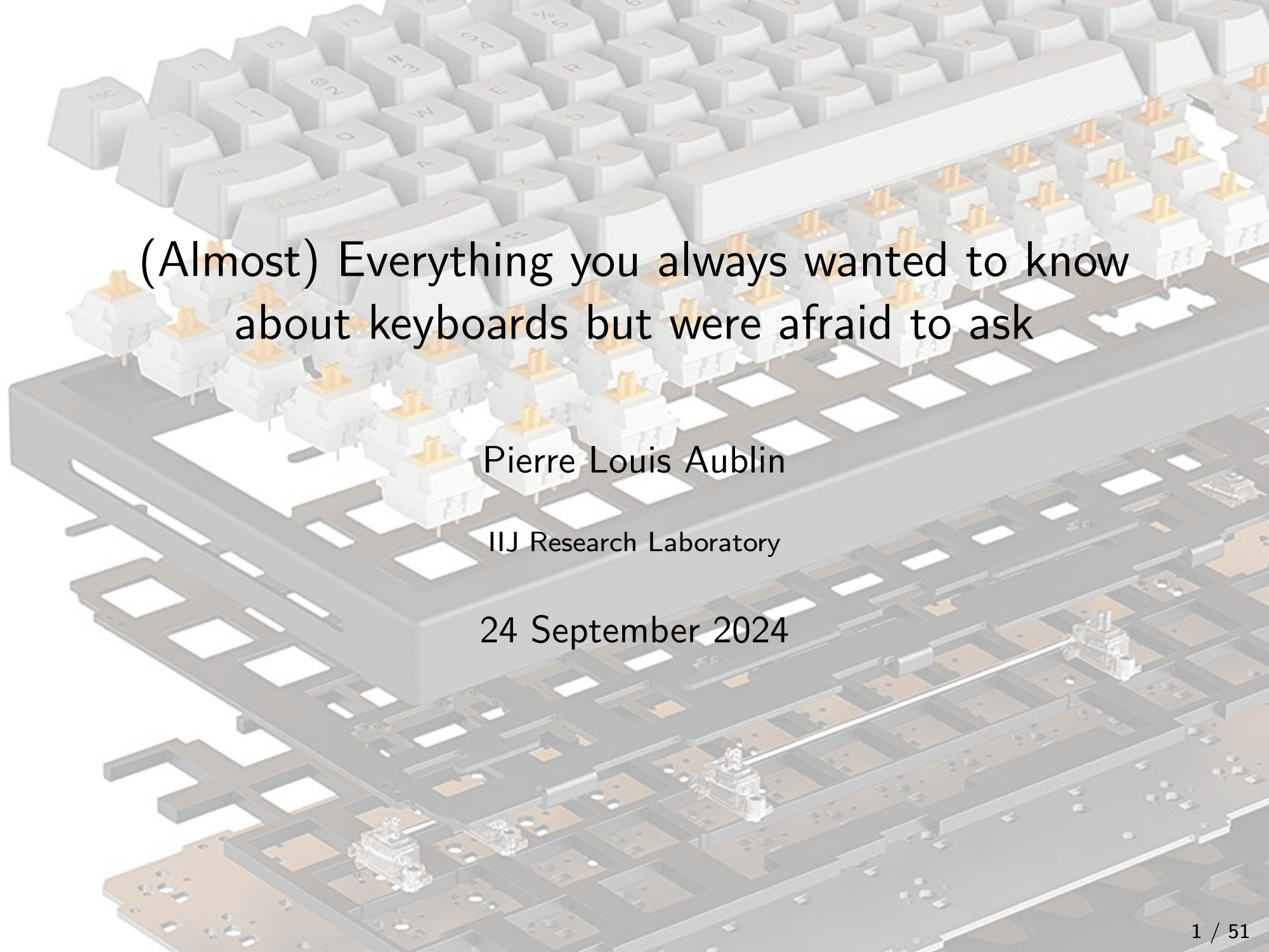


(Almost) Everything you always wanted to know about keyboards but were afraid to ask

Pierre Louis Aublin

IIJ Research Laboratory

24 September 2024

A wise man once said

'To look to the future we must first look back upon the past. That is where the seeds of the future were planted.'

Albert Einstein

Outline

1. Keyboard origins and structure
2. A better keyboard
3. Research project: securing keyboard inputs

This is a keyboard

This is a keyboard

Where does it come from?

A long time ago...

The typewriter was created

Figure: Sholes & Glidden, 1873

Sholes & Glidden QWERTY layout

The rise of QWERTY

Remington and Sons' monopoly

- Remington and Sons: one of largest typewriter manufacturers
- Bought Sholes design

The QWERTY cartel

- QWERTY: not the only layout nor the *best*
- Remington & S. losing money to rivals w/ better designs
- 1893: merge with 4 other manufacturers to control market
- QWERTY became the *de facto* standard
- Early computers used typewriters

This is a keyboard

Why is it structured like this?

Decomposing the keyboard: the numpad

Numpad history

Very useful for data entry (spreadsheet, etc.)

1914: *tenkey* adding machine

1951: UNIVAC-1 console

Decomposing the keyboard: the arrow keys

Numpad history

Originally used to move the cursor (pre-mouse era)

1976: HJKL on ADM-3A

1982: inverted T on LK201

Decomposing the keyboard: the nav keys

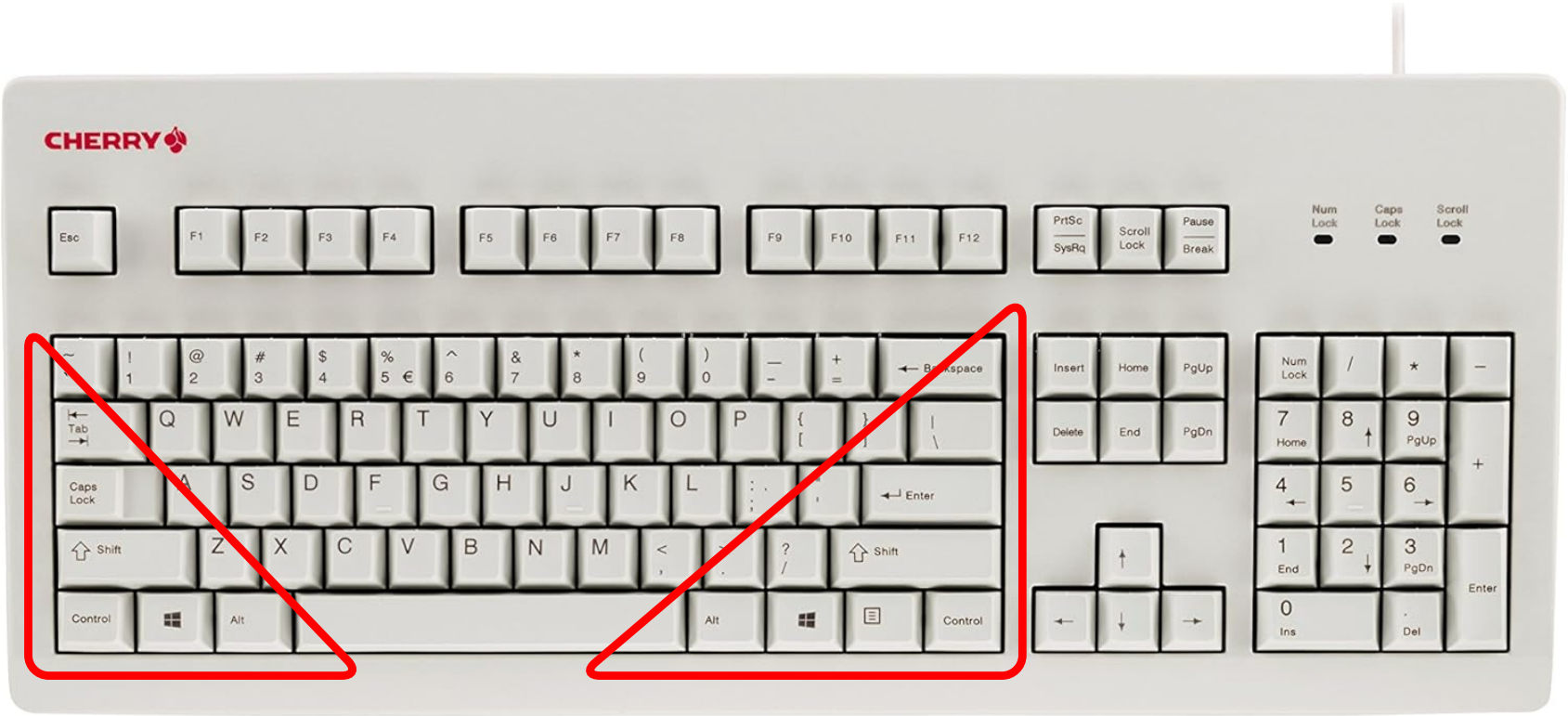
Popularized by IBM Model M keyboard

Decomposing the keyboard: the system keys

Vestige of old IBM PC

Decomposing the keyboard: the modifiers

Originates from typewriters and teleprinters



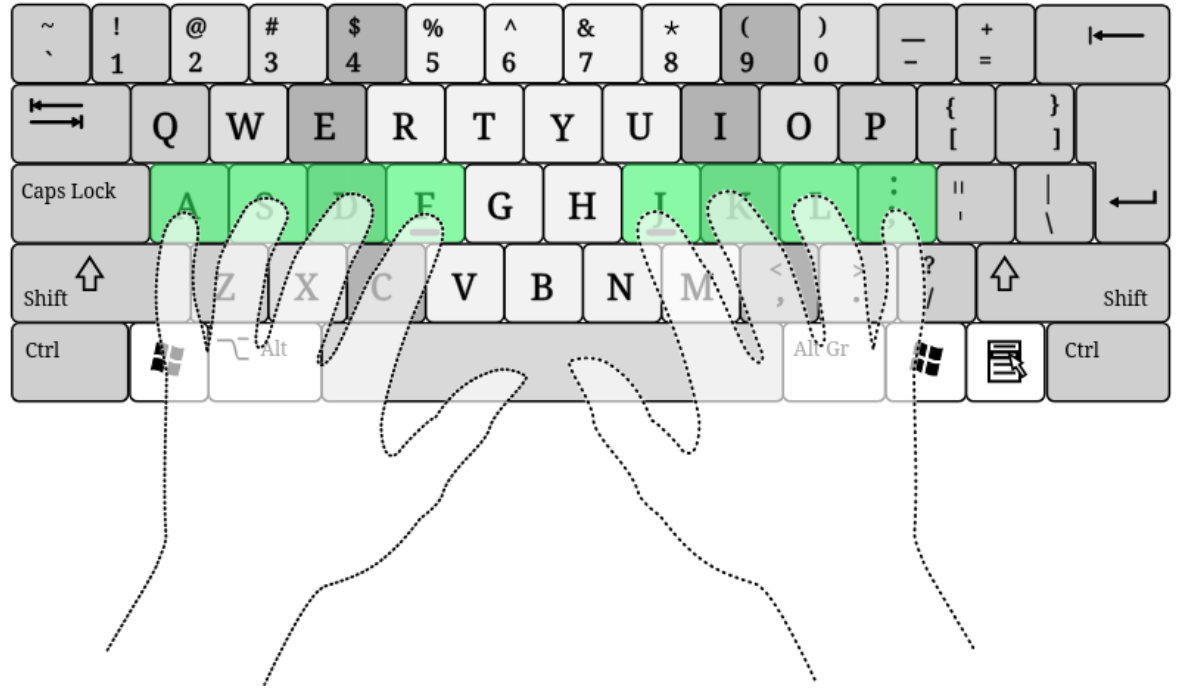
Decomposing the keyboard: the function keys

This is a keyboard

How to use it?

Touch typing: Fingers on the homerow

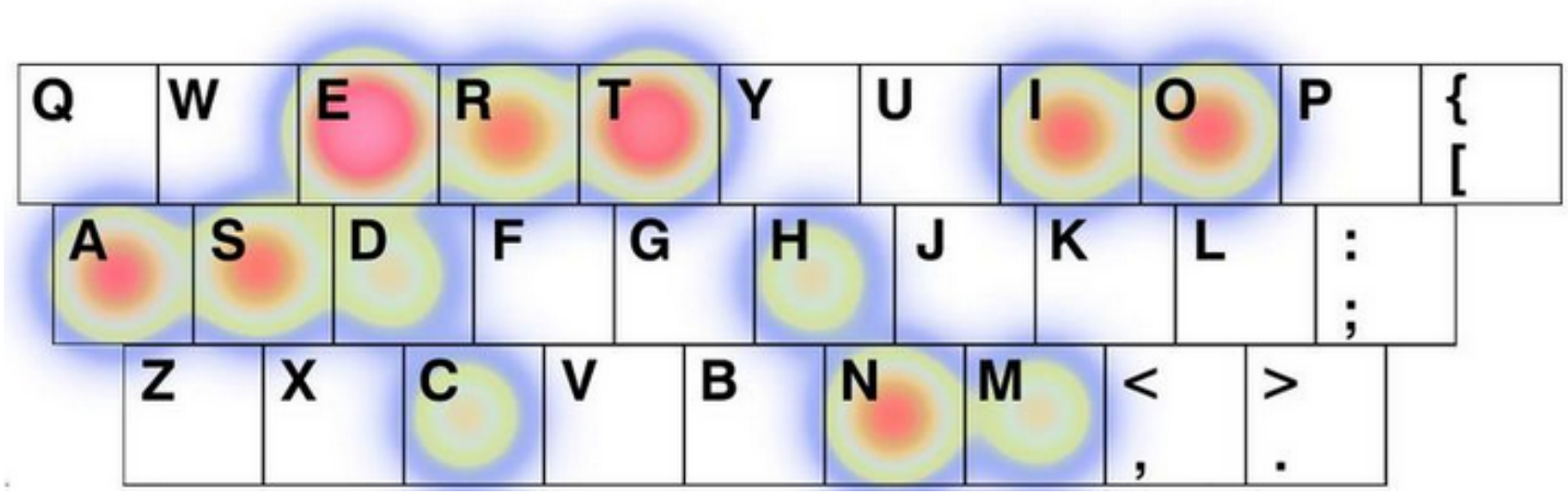
How to use a keyboard



Advantages

- Less finger/hand movements
- Speed
- Keep attention on task
- Less neck pain

QWERTY is not the best layout



- Designed 150 years ago, for typewriters
- Fingers need to move a lot
- Pinky, weakest finger, used for important keys
- Left/right hand not balanced (e.g., *average*)
- Some difficult bigrams (e.g., *cr*, *be*)

Keyboard problems

Problem #1

Most keyboards use an outdated layout: `QWERTY`

Problem #2

Some keys are unnecessary, misplaced, or for outdated functionalities

Problem #3

Using a keyboard can lead to serious injuries (carpal tunnel syndrome, arthritis, tendinitis, ...)

This is a keyboard

Can we do better?

A better keyboard

Ergonomic keyboard

Figure: Typematrix

- Ortholinear
- Enter/backspace in the centre
- Pinky keys easier to press
- Close mouse location

Split keyboard

Figure: Kinesis Advantage 360

- More natural hand position: split and curved
- Thumb cluster

How many keys do you need?

Figure: Cherry keyboard 105 keys

'Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.'

Antoine de Saint-Exupéry

Ten Key Less (TKL): 87 keys

Figure: Keychron K8

65% keyboard: 68 keys

Figure: Ziyoulang T8 RGB

60% keyboard: 60 keys

Figure: Happy Hacking Keyboard

40% keyboard: 48 keys

Figure: OLKB Planck

34 keys is all your need

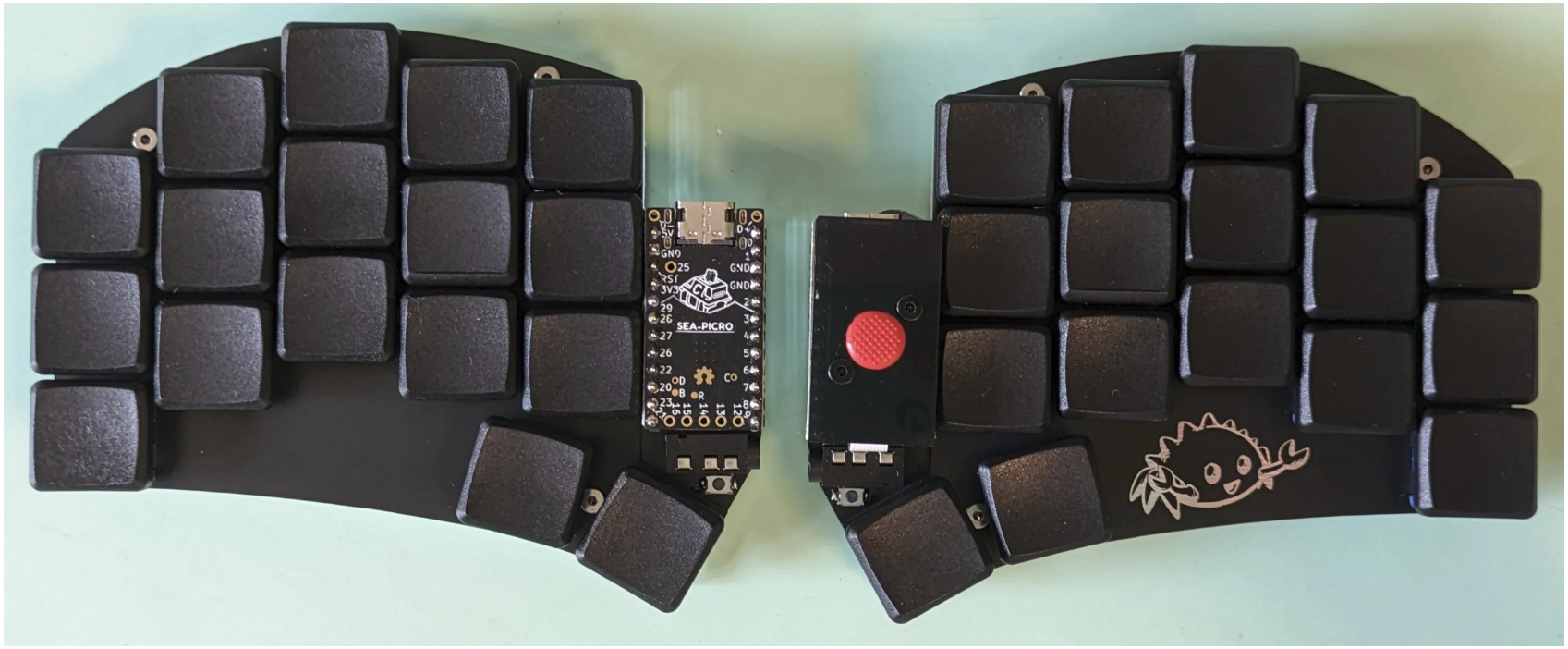
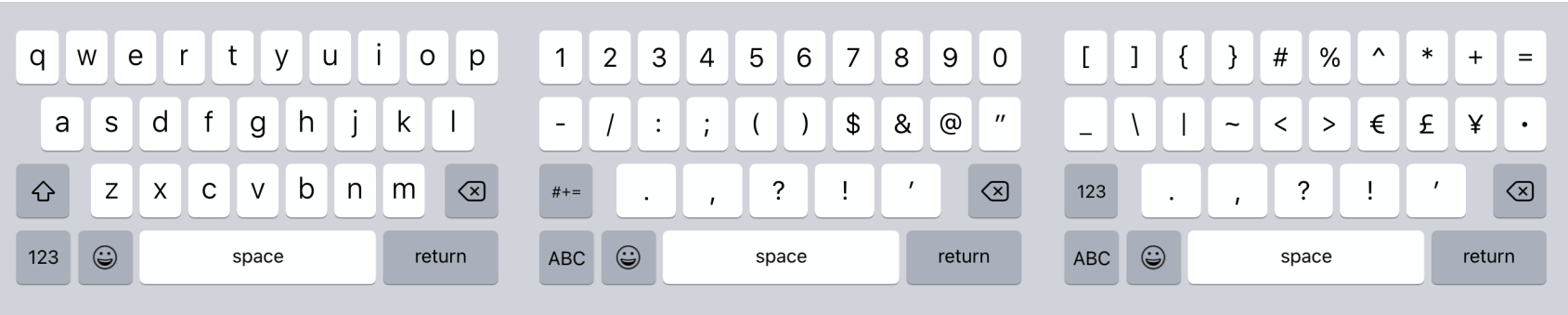


Figure: Ferris sweep

Where are my missing keys?

It's all about layers



Change layer via a special key

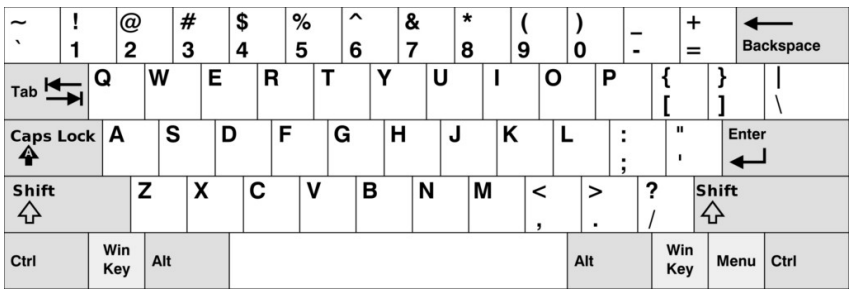
A key press produces a different output depending on the current layer. E.g., similar to `SHIFT` or `CTRL`

Layers can be persistent, one-tap, combo, etc.

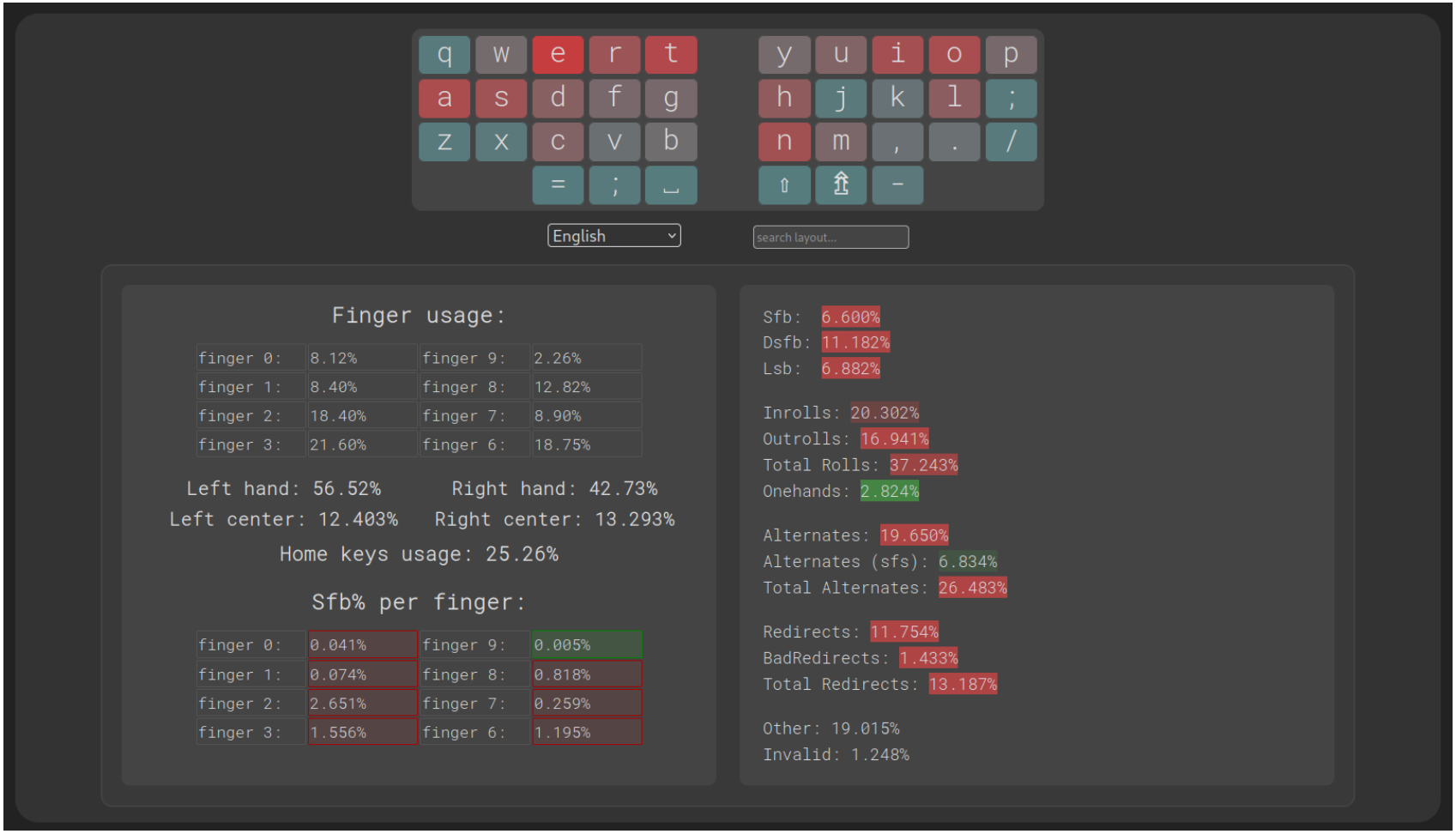
What is the best layout?

Metrics to compare layouts

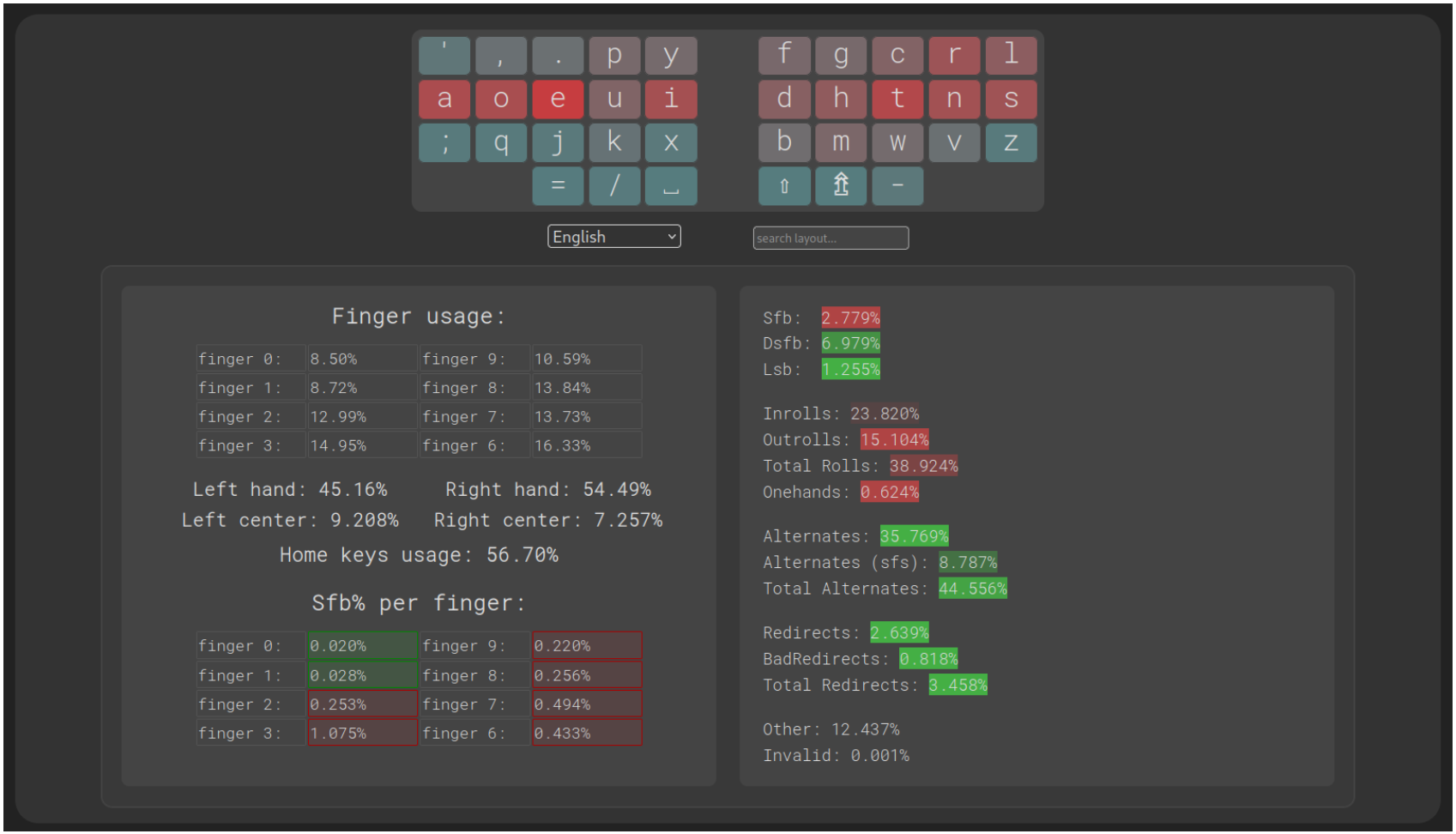
Metric	Description	Example	Target
Home keys		`lag`	100
Left/Right		`soap`	1.0
Sfb	Same finger bigram: pressing two keys in succession w/ same finger	`fr`	0
Rolls	pressing two keys with one hand, and third with other hand	`our`	100
Alternate	alternate keys between hands	`and`	100
Redirects	one-handed trigram where direction changes	`sad`	0



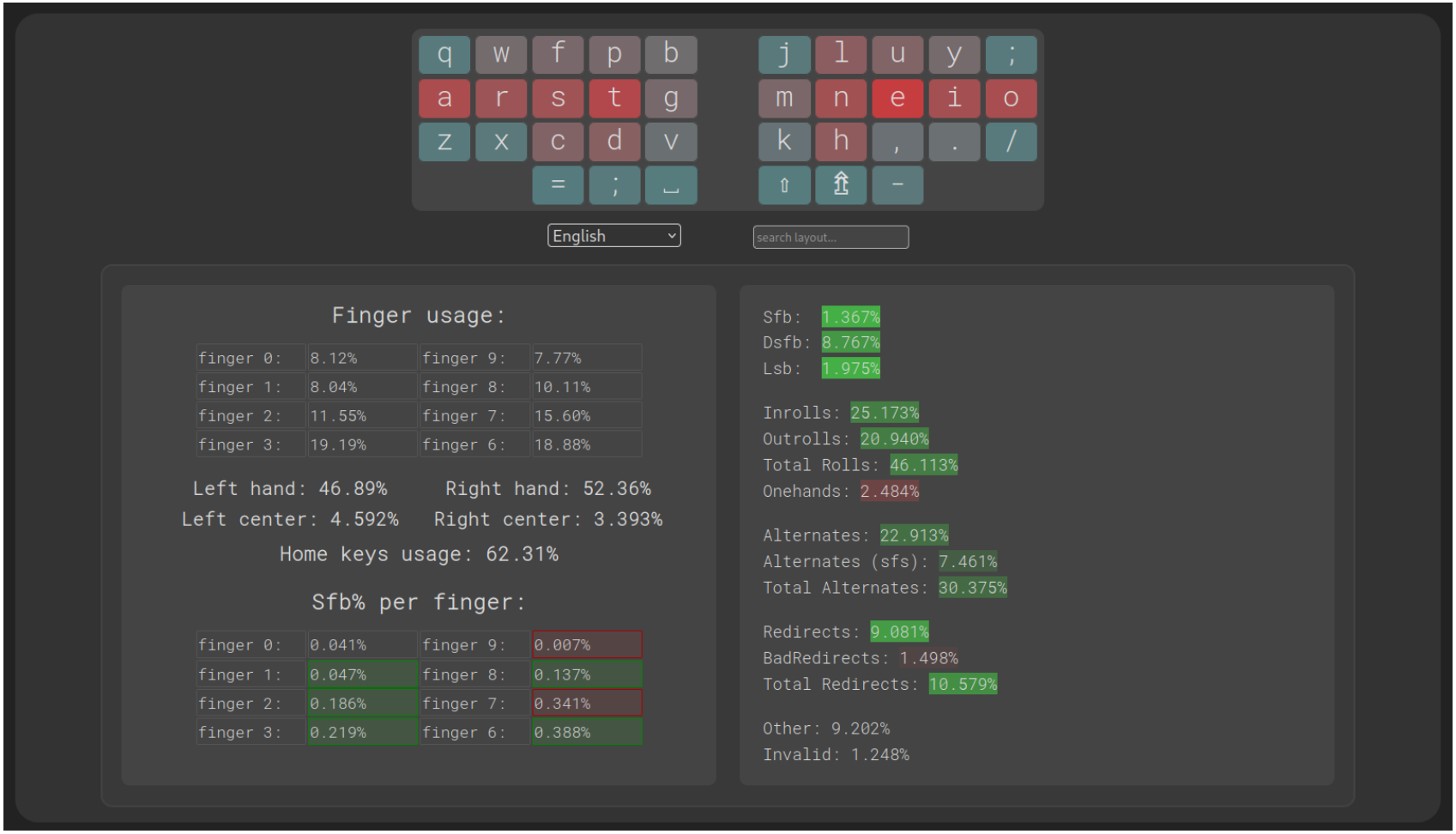
Qwerty


q w e r t
a s d f g
z x c v b
= ; ␣
y u i o p
h j k l ;
n m , . /
⇧ ⇧ -
English
search layout...
Finger usage:
finger 0: 8.12%
finger 1: 8.40%
finger 2: 18.40%
finger 3: 21.60%
finger 9: 2.26%
finger 8: 12.82%
finger 7: 8.90%
finger 6: 18.75%
Left hand: 56.52%
Right hand: 42.73%
Left center: 12.403%
Right center: 13.293%
Home keys usage: 25.26%
Sfb% per finger:
finger 0: 0.041%
finger 1: 0.074%
finger 2: 2.651%
finger 3: 1.556%
finger 9: 0.005%
finger 8: 0.818%
finger 7: 0.259%
finger 6: 1.195%
Sfb: 6.600%
Dsfb: 11.182%
Lsb: 6.882%
Inrolls: 20.302%
Outrolls: 16.941%
Total Rolls: 37.243%
Onehands: 2.824%
Alternates: 19.650%
Alternates (sfs): 6.834%
Total Alternates: 26.483%
Redirects: 11.754%
BadRedirects: 1.433%
Total Redirects: 13.187%
Other: 19.015%
Invalid: 1.248%

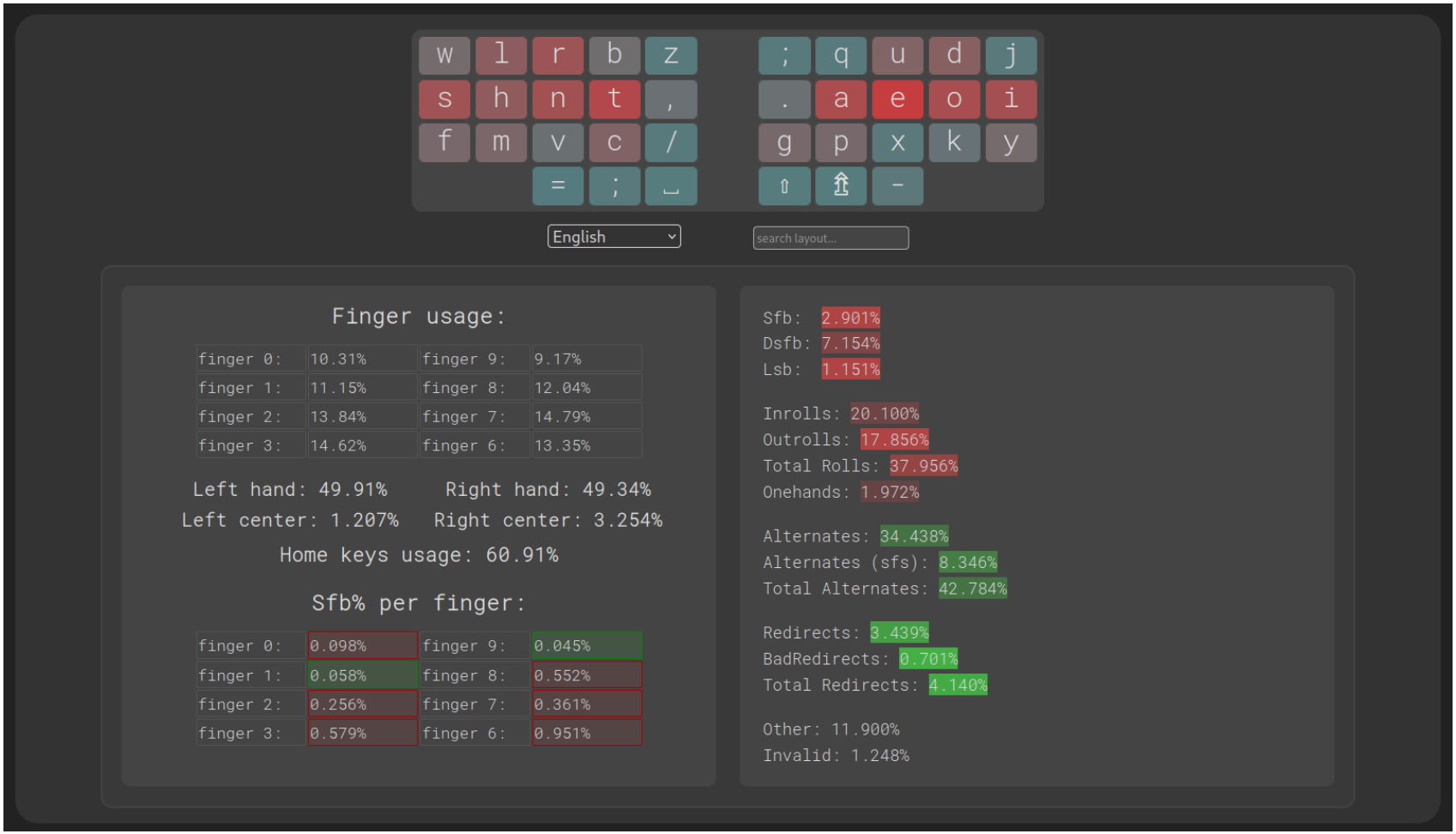
Dvorak


' , . p y
a o e u i
; q j k x
= / ␣
f g c r l
d h t n s
b m w v z
⇧ ⇧ -
English
search layout...
Finger usage:
finger 0: 8.50%
finger 1: 8.72%
finger 2: 12.99%
finger 3: 14.95%
finger 9: 10.59%
finger 8: 13.84%
finger 7: 13.73%
finger 6: 16.33%
Left hand: 45.16%
Right hand: 54.49%
Left center: 9.208%
Right center: 7.257%
Home keys usage: 56.70%
Sfb% per finger:
finger 0: 0.020%
finger 1: 0.028%
finger 2: 0.253%
finger 3: 1.075%
finger 9: 0.220%
finger 8: 0.256%
finger 7: 0.494%
finger 6: 0.433%
Sfb: 2.779%
Dsfb: 6.979%
Lsb: 1.255%
Inrolls: 23.820%
Outrolls: 15.104%
Total Rolls: 38.924%
Onehands: 0.624%
Alternates: 35.769%
Alternates (sfs): 8.787%
Total Alternates: 44.556%
Redirects: 2.639%
BadRedirects: 0.818%
Total Redirects: 3.458%
Other: 12.437%
Invalid: 0.001%

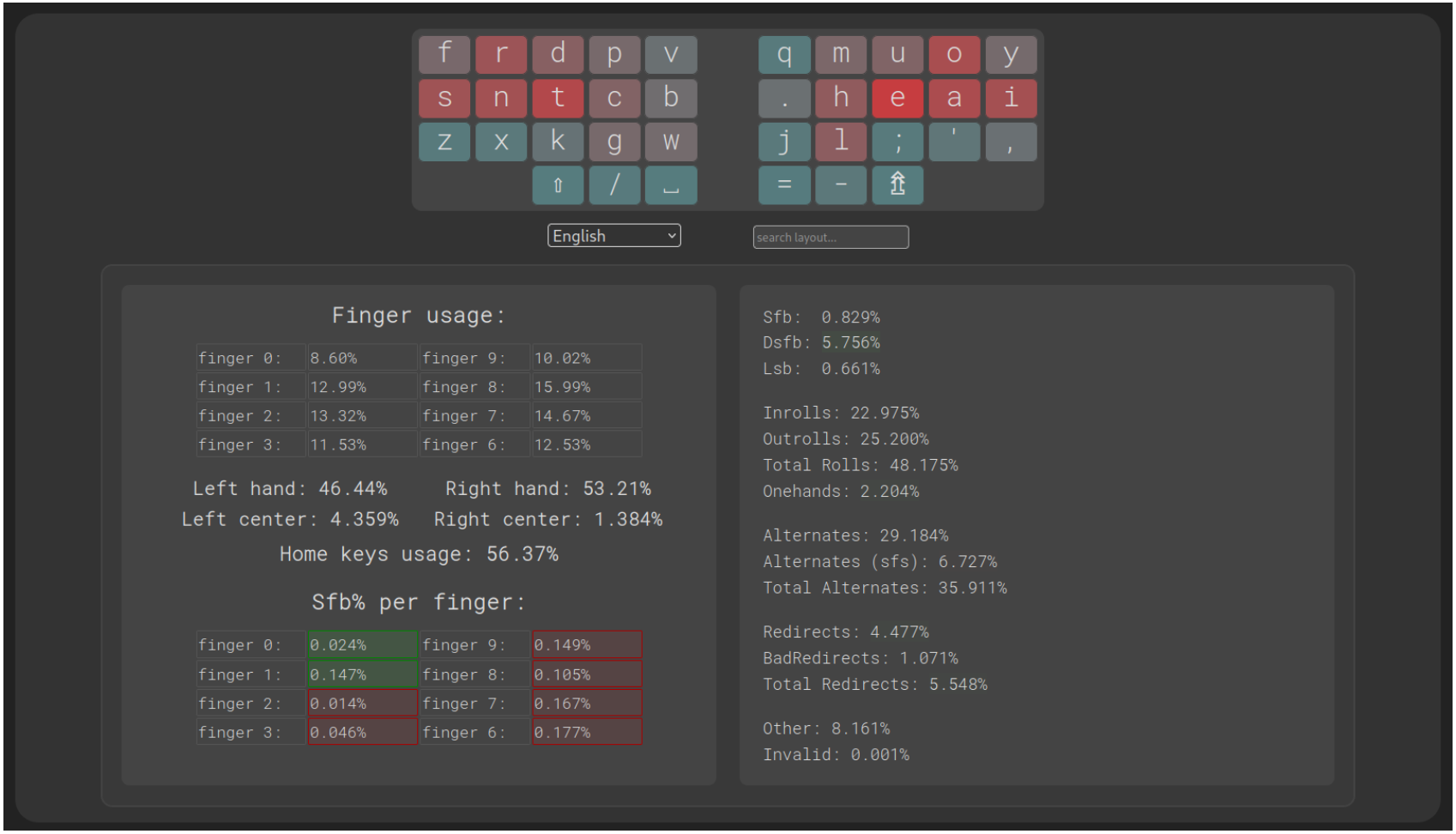
Colemak-DH


q w f p b
a r s t g
z x c d v
= ; ␣
j l u y ;
m n e i o
k h , . /
⇧ ⇫ -
English
search layout...
Finger usage:
finger 0: 8.12%
finger 1: 8.04%
finger 2: 11.55%
finger 3: 19.19%
finger 9: 7.77%
finger 8: 10.11%
finger 7: 15.60%
finger 6: 18.88%
Left hand: 46.89%
Right hand: 52.36%
Left center: 4.592%
Right center: 3.393%
Home keys usage: 62.31%
Sfb% per finger:
finger 0: 0.041%
finger 1: 0.047%
finger 2: 0.186%
finger 3: 0.219%
finger 9: 0.007%
finger 8: 0.137%
finger 7: 0.341%
finger 6: 0.388%
Sfb: 1.367%
Dsfb: 8.767%
Lsb: 1.975%
Inrolls: 25.173%
Outrolls: 20.940%
Total Rolls: 46.113%
Onehands: 2.484%
Alternates: 22.913%
Alternates (sfs): 7.461%
Total Alternates: 30.375%
Redirects: 9.081%
BadRedirects: 1.498%
Total Redirects: 10.579%
Other: 9.202%
Invalid: 1.248%

Halmak


w l r b z ; q u d j
s h n t , . a e o i
f m v c / g p x k y
= ; ␣ ⇧ ⇪ -
English
search layout...
Finger usage:
finger 0: 10.31% finger 9: 9.17%
finger 1: 11.15% finger 8: 12.04%
finger 2: 13.84% finger 7: 14.79%
finger 3: 14.62% finger 6: 13.35%
Left hand: 49.91% Right hand: 49.34%
Left center: 1.207% Right center: 3.254%
Home keys usage: 60.91%
Sfb% per finger:
finger 0: 0.098% finger 9: 0.045%
finger 1: 0.058% finger 8: 0.552%
finger 2: 0.256% finger 7: 0.361%
finger 3: 0.579% finger 6: 0.951%
Sfb: 2.901%
Dsfb: 7.154%
Lsb: 1.151%
Inrolls: 20.100%
Outrolls: 17.856%
Total Rolls: 37.956%
Onehands: 1.972%
Alternates: 34.438%
Alternates (sfs): 8.346%
Total Alternates: 42.784%
Redirects: 3.439%
BadRedirects: 0.701%
Total Redirects: 4.140%
Other: 11.900%
Invalid: 1.248%

Recurva


f r d p v
s n t c b
z x k g w
⇧ / ␣
q m u o y
. h e a i
j l ; ' ,
= - ⇧
English
search layout...
Finger usage:
finger 0: 8.60%
finger 1: 12.99%
finger 2: 13.32%
finger 3: 11.53%
finger 9: 10.02%
finger 8: 15.99%
finger 7: 14.67%
finger 6: 12.53%
Left hand: 46.44%
Right hand: 53.21%
Left center: 4.359%
Right center: 1.384%
Home keys usage: 56.37%
Sfb% per finger:
finger 0: 0.024%
finger 1: 0.147%
finger 2: 0.014%
finger 3: 0.046%
finger 9: 0.149%
finger 8: 0.105%
finger 7: 0.167%
finger 6: 0.177%
Sfb: 0.829%
Dsfb: 5.756%
Lsb: 0.661%
Inrolls: 22.975%
Outrolls: 25.200%
Total Rolls: 48.175%
Onehands: 2.204%
Alternates: 29.184%
Alternates (sfs): 6.727%
Total Alternates: 35.911%
Redirects: 4.477%
BadRedirects: 1.071%
Total Redirects: 5.548%
Other: 8.161%
Invalid: 0.001%

Layouts summary

Metrics (%)	QWERTY	Dvorak	Colemak-DH	Halmak	Recurva
Home keys	25	57	62	61	56
Left/Right	1.3	0.8	0.9	1.0	0.9
Sfb	6.6	2.8	1.4	2.9	0.8
Rolls	37	39	46	38	48
Alternate	26	45	30	43	36
Redirects	13	3	11	4	6

Stats for English language

Your own layout

Most layouts optimized for English writing

Give your inputs to a program

`https://github.com/xsznix/keygen`

Your own layout

Most layouts optimized for English writing

Give your inputs to a program

`https://github.com/xsznix/keygen`

Figure: Layout optimized for the Rust programming language

Your own layout

Most layouts optimized for English writing

Give your inputs to a program

`https://github.com/xsznix/keygen`

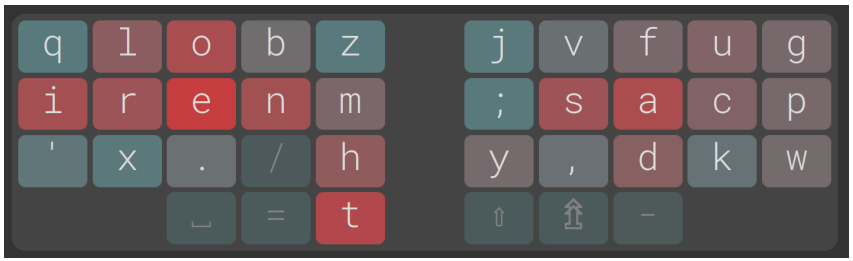


Figure: Layout optimized for Rust

Metrics (%)	QWERTY	Dvorak	Iren
Home keys	25	57	51
Left/Right	1.3	0.8	1.5
Sfb	6.6	2.8	2.6
Rolls	37	39	42
Alternate	26	45	25
Redirects	13	3	13

Stats for English language

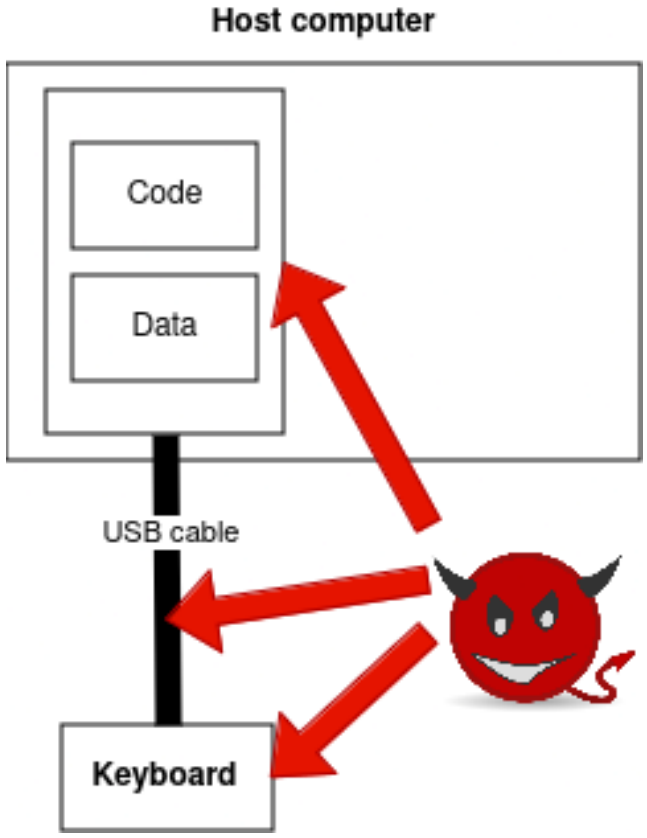
Palantir: Secure Keyboard Inputs

Palantir: Securing keyboard inputs

Industrial Espionnage

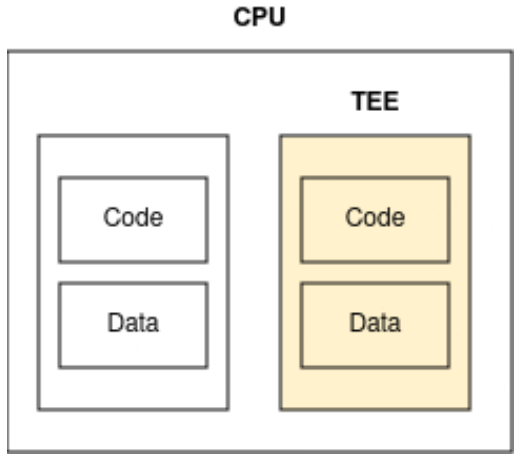
Aimbots in Online Video Games

Root cause: inputs originate from untrusted world



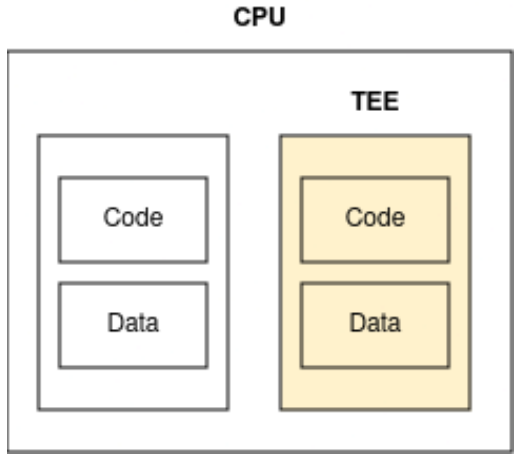
Trusted Execution Environment (TEE)

- Trusted, tamper-proof component on the computers



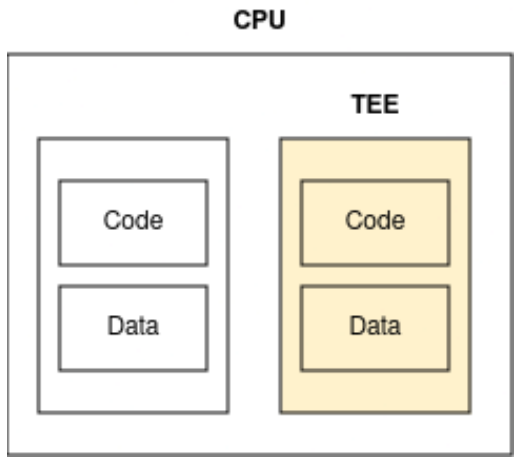
Trusted Execution Environment (TEE)

- Trusted, tamper-proof component on the computers
- provides security guarantees of code and data inside
  - confidentiality
  - integrity



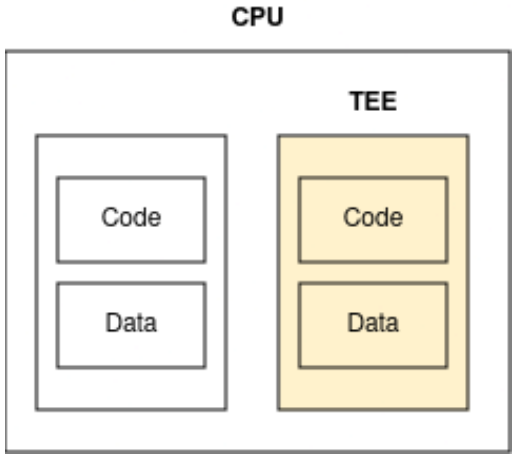
Trusted Execution Environment (TEE)

- Trusted, tamper-proof component on the computers
- provides security guarantees of code and data inside
  - confidentiality
  - integrity
- secure against strong attacker w/ access to both SW and HW



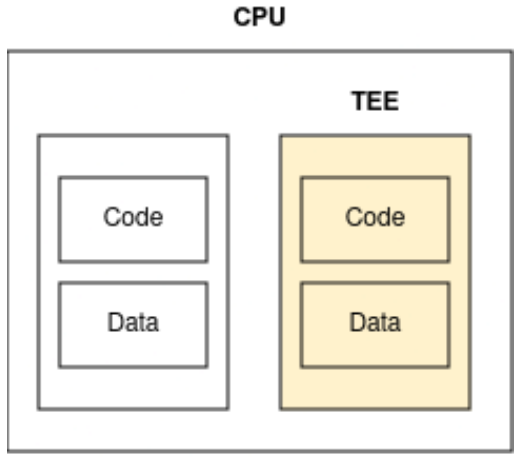
Trusted Execution Environment (TEE)

- Trusted, tamper-proof component on the computers
- provides security guarantees of code and data inside
  - confidentiality
  - integrity
- secure against strong attacker w/ access to both SW and HW
- examples: Intel SGX, ARM TrustZone, RISC-V Keystone



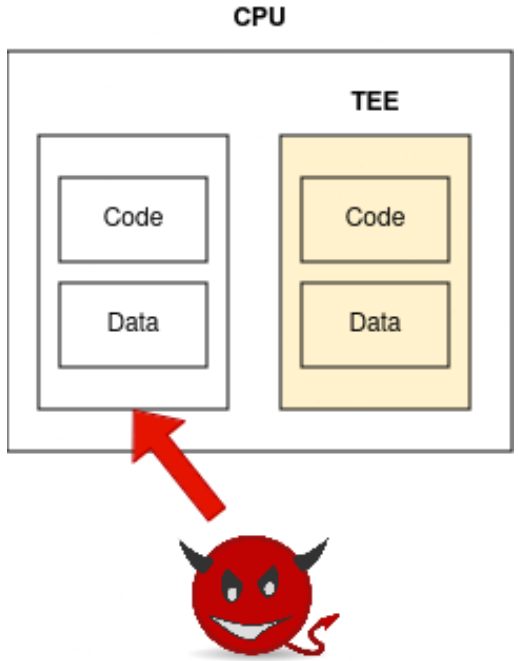
Trusted Execution Environment (TEE)

- Trusted, tamper-proof component on the computers
- provides security guarantees of code and data inside
  - confidentiality
  - integrity
- secure against strong attacker w/ access to both SW and HW
- examples: Intel SGX, ARM TrustZone, RISC-V Keystone



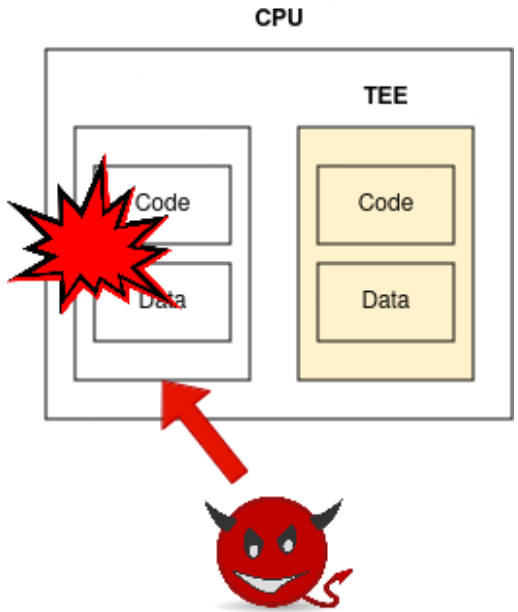
Trusted Execution Environment (TEE)

- Trusted, tamper-proof component on the computers
- provides security guarantees of code and data inside
  - confidentiality
  - integrity
- secure against strong attacker w/ access to both SW and HW
- examples: Intel SGX, ARM TrustZone, RISC-V Keystone



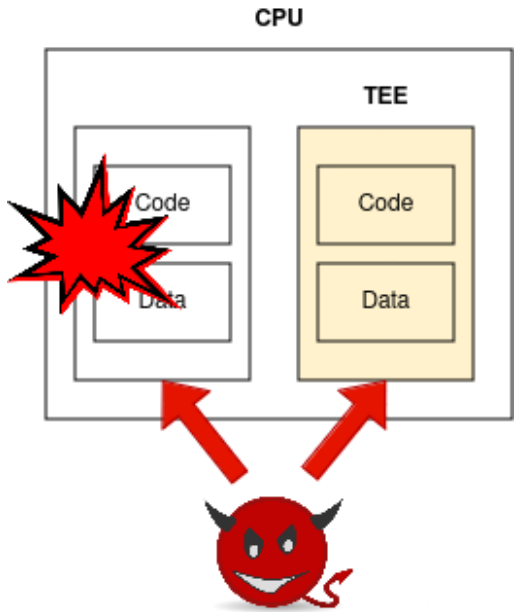
Trusted Execution Environment (TEE)

- Trusted, tamper-proof component on the computers
- provides security guarantees of code and data inside
  - confidentiality
  - integrity
- secure against strong attacker w/ access to both SW and HW
- examples: Intel SGX, ARM TrustZone, RISC-V Keystone



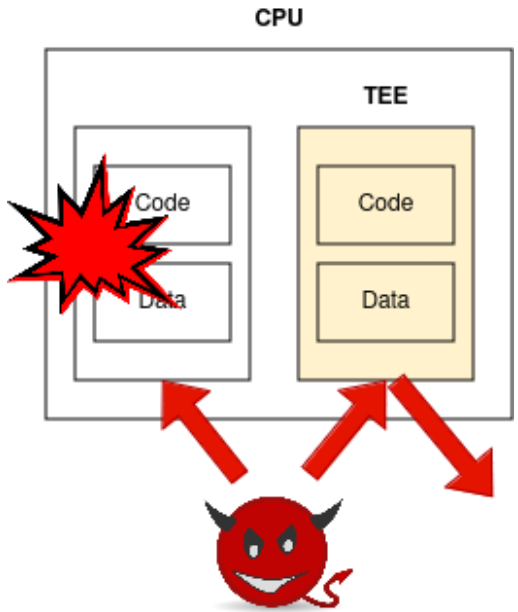
Trusted Execution Environment (TEE)

- Trusted, tamper-proof component on the computers
- provides security guarantees of code and data inside
  - confidentiality
  - integrity
- secure against strong attacker w/ access to both SW and HW
- examples: Intel SGX, ARM TrustZone, RISC-V Keystone

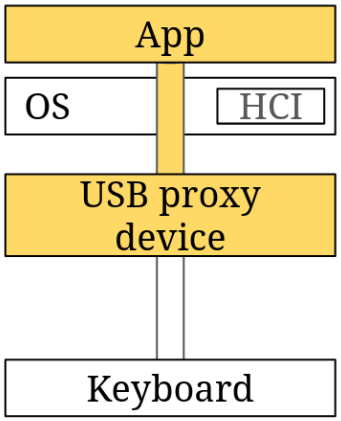


Trusted Execution Environment (TEE)

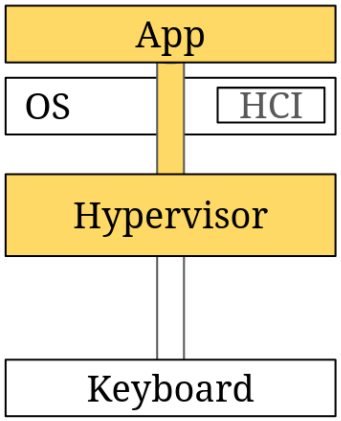
- Trusted, tamper-proof component on the computers
- provides security guarantees of code and data inside
  - confidentiality
  - integrity
- secure against strong attacker w/ access to both SW and HW
- examples: Intel SGX, ARM TrustZone, RISC-V Keystone



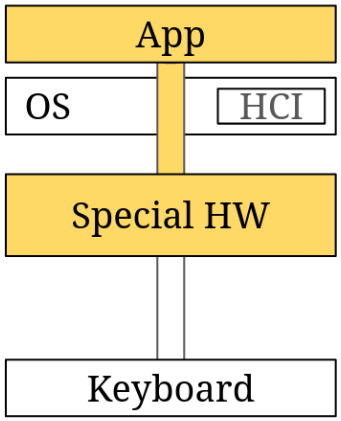
Related work



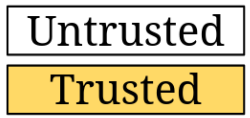
TIO [Stancu, '19]
SGX-USB [Jang, '24]



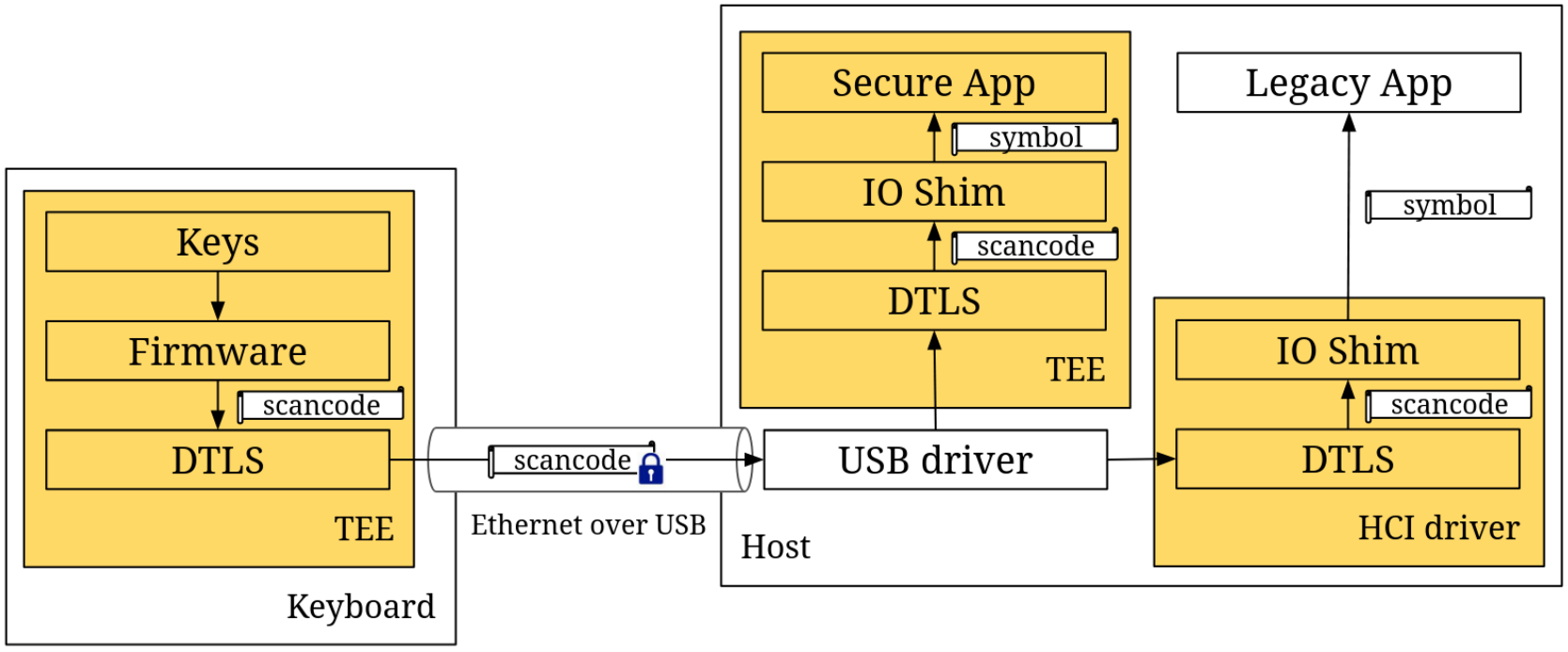
SGXIO [Weiser, '17]



Aurora [Liang, '20]



Palantir architecture



Challenges

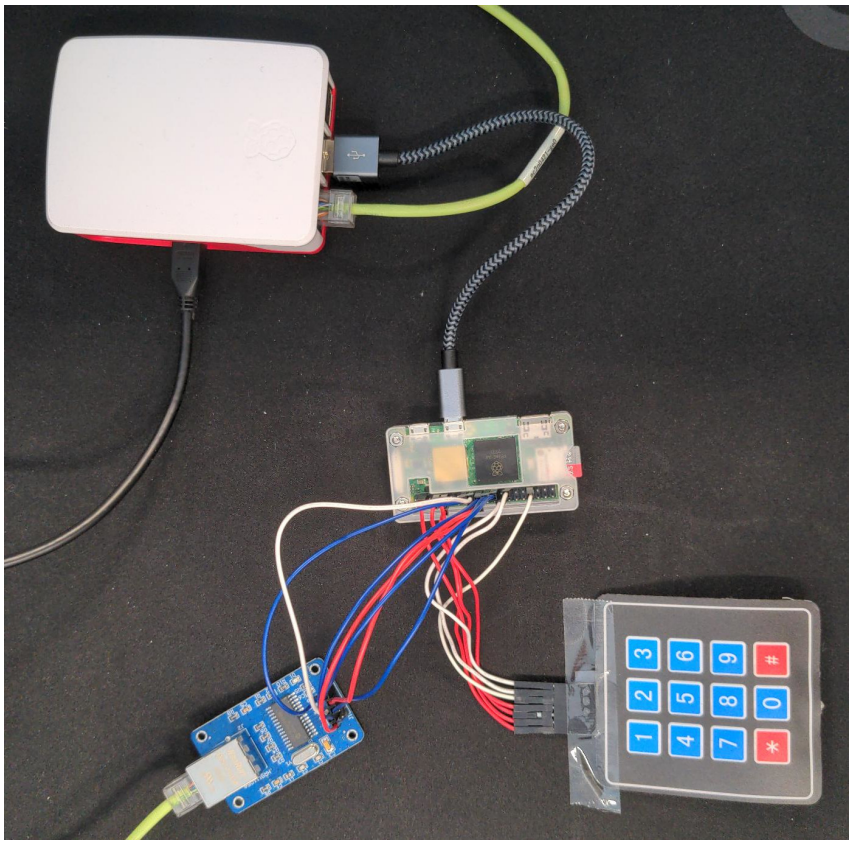
Keys distribution and TEEs attestation

Assumption

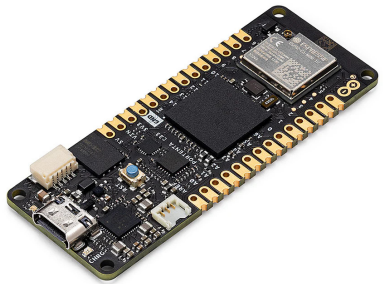
Keyboard tamper-resistant

Hardware Prototype

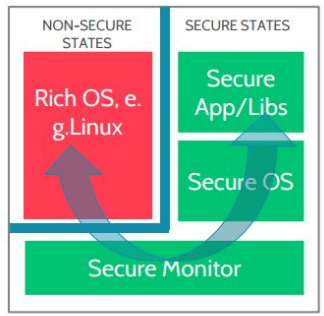
Raspberry Pi Nano2 W (Cortex-A)



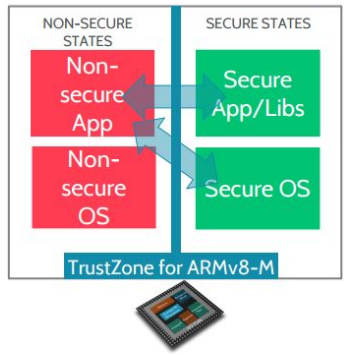
Arduino Portenta C33 (Cortex-M)



TrustZone for ARMv8-A



TrustZone for ARMv8-M



Preliminary results

	UDP	DTLS	ratio
Throughput (scancode/s)	8000	2200	0.28
Latency (μs)	123	445	3.6x

Preliminary results

	UDP	DTLS	ratio
Throughput (scancode/s)	8000	2200	0.28
Latency (μs)	123	445	3.6x

Comparison

- Pro gamer: 10 actions per second
- Gaming mouse: 1000 updates per second
- Gamepad latency: 1.4ms

Conclusion

Create your own keyboard 101

You need:

- PCB
- Microcontroller: RP2040-based
- Switches and keycaps
- Firmware: QMK, ZMK

You can get components at:

- Yushakobo, Akihabara: `https://yushakobo.jp/`
- `https://shop.beekeeb.com/product`

And then practice at:

- `https://www.keybr.com/`
- `https://monkeytype.com/`

Acknowledgements

- https://docs.google.com/document/d/1_a5Nzbkwyk1o0bvTctZrtgsee9jSP-6I0q3A0_9Mzm0/
- https://bit.ly/layout-doc-v2
- https://www.youtube.com/watch?v=unMXQTSQEak
- https://www.youtube.com/watch?v=x7LQevYn7d0
- https://oxey.dev/playground/index.html

Conclusion

- Keyboards are ubiquitous
- But many of us use an antique design
- There are better alternatives to suit your needs
- main entrypoint for your ideas; require strong security
- We propose a novel way to secure keyboard inputs