What You Say is What You Get

Handsfree Coding in 2023

Techcamp 2023, Hamburg, Germany
June 01, 2023
Wolle

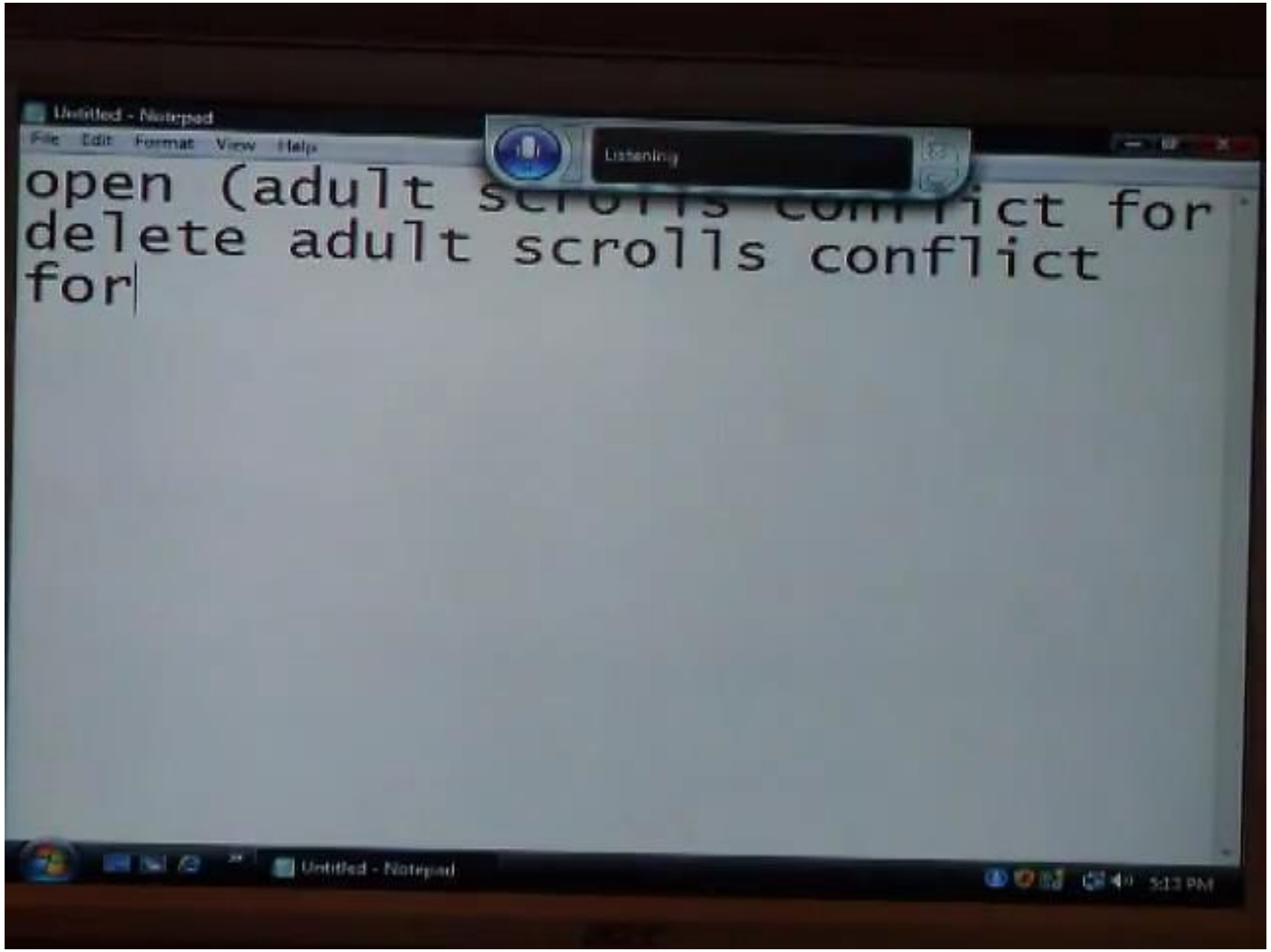
Carl von Ossietzky
Universität
Oldenburg

It‘s Simple, Really!

The requirements:

- ✓ **Microphone**: Every notebook has one!
- ✓ **Speech Recognition Software (SR)**: Included in Windows since 2007!
- ✓ **Voice Command Execution**: Available in every SR software!

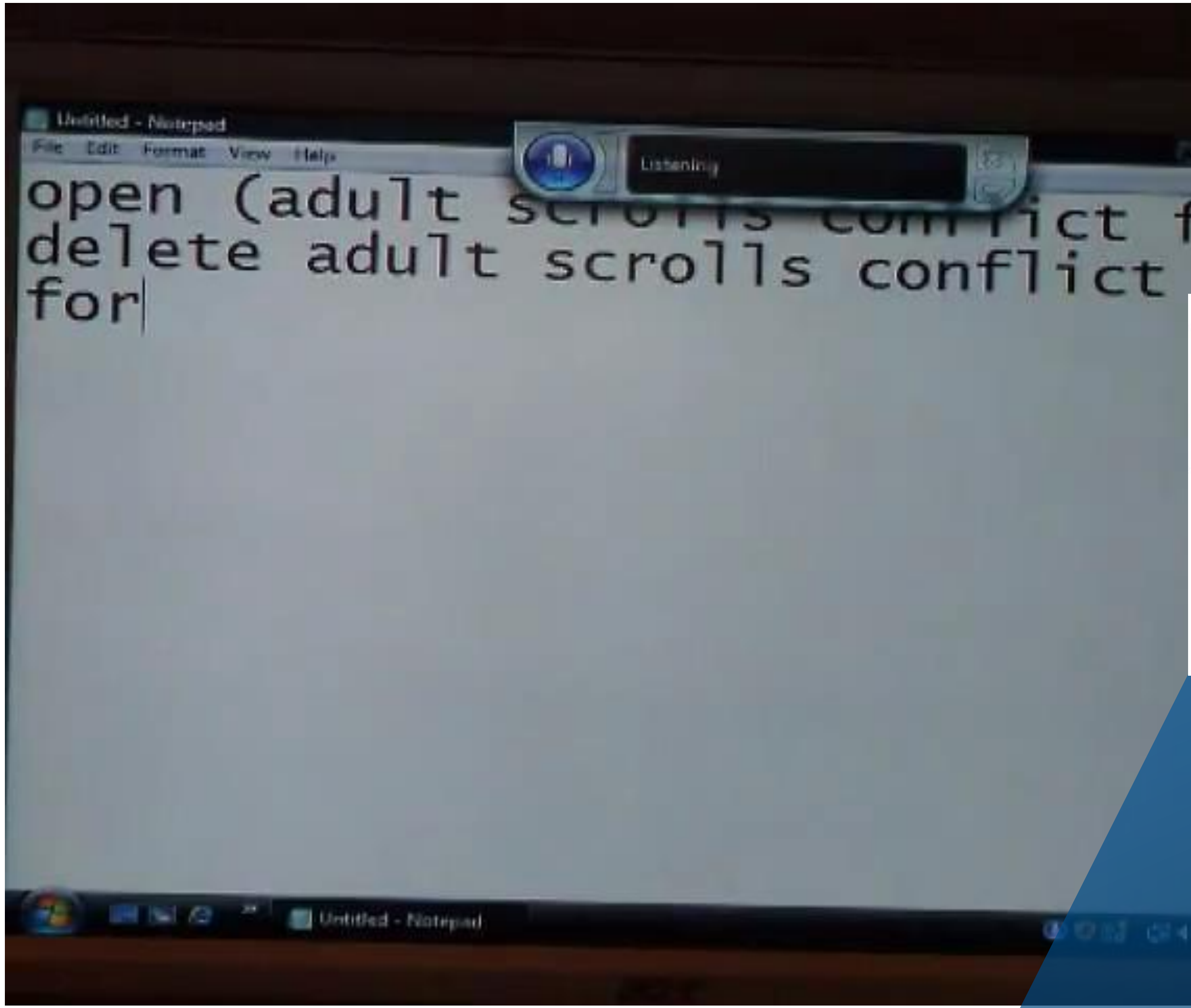
Let Me Just Show You How Easy It is



```
open (adult scrolls conflict for
delete adult scrolls conflict
for
```

scrubadub1. Windows Vista Speech Recognition Tested – Perl Scripting, YouTube, 2007

Let Me Just Show You How Easy It is



```
open (adult scrolls conflict for
delete adult scrolls conflict
for
```

scrubadub1. Windows Vista Speech Recognition Tested - Perl Scripting, YouTube, 2007

Sept 13-14, 2019
thestrangeloop.com

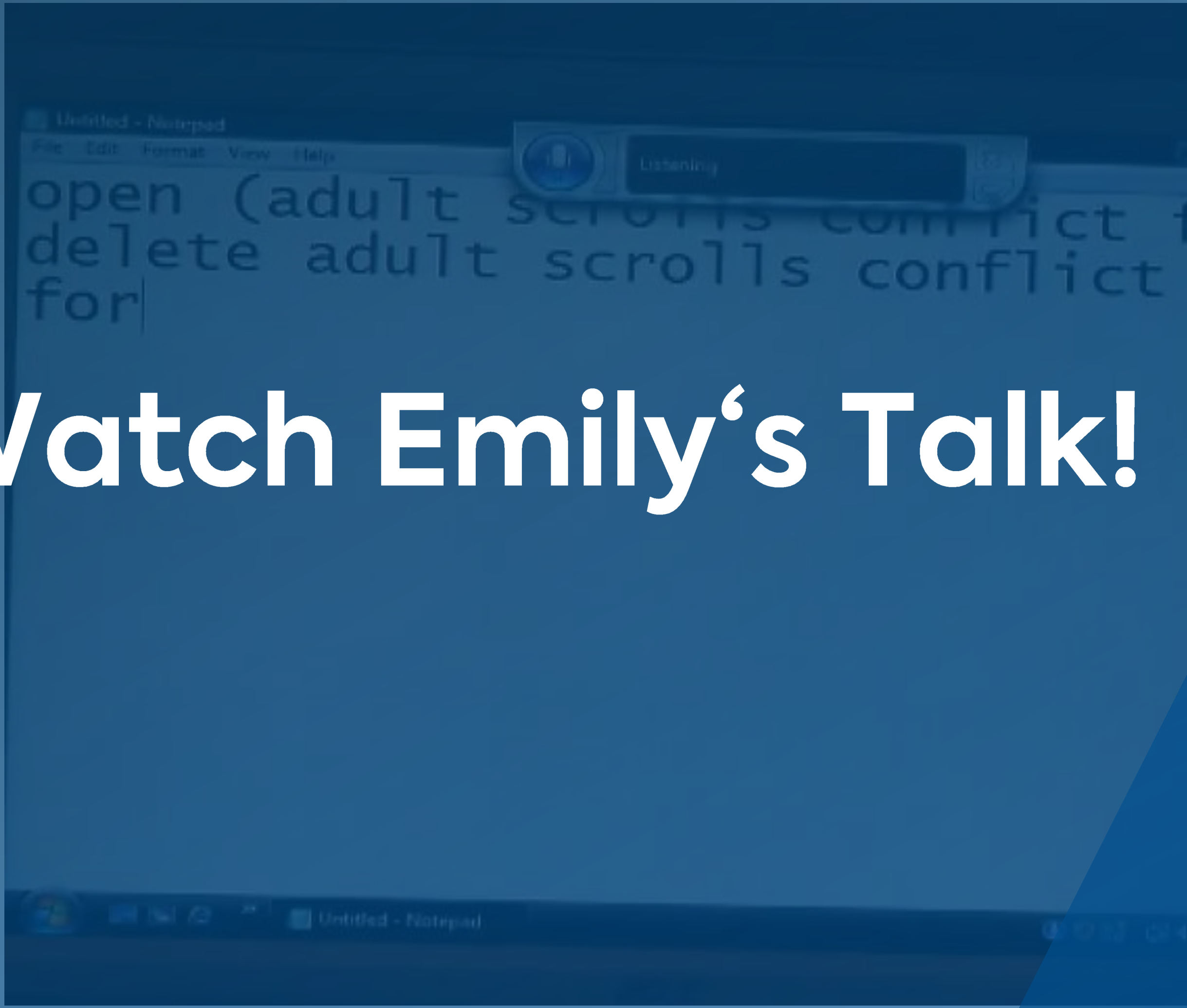
whois emily

- Software Engineer
- GitHub: @2shea
- Twitter: @yomilly
- I write code for Fastly

Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop (2019)

Idea to use this video blatantly stolen from: Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop, 2019

Let Me Just Show You How Easy It is



Untitled - Notepad

File Edit Format View Help

Listening

```
open (adult scrolls conflict for
delete adult scrolls conflict
for
```

Untitled - Notepad

Go Watch Emily‘s Talk!

strangeloop

Sept 13-14, 2019

thestrangeloop.com

whois emily

- Software Engineer
- GitHub: @2shea
- Twitter: @yomilly
- I write code for Fastly

Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop (2019)

scrubadub1. Windows Vista Speech Recognition Tested - Perl Scripting, YouTube, 2007

Idea to use this video blatantly stolen from: Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop, 2019

Where‘s the Challenge?

Outline: What This Talk is Going to Cover

1 **My Personal Background**

As data engineer & scientist, I use handsfree coding every day.

2 **Demo & Usage Examples**

Handsfree coding is awesome and can be useful for everyone!

3 **Setup & Best Practices**

No-cost base setup with optional upgrades (e.g. for eye tracking).

4 **How to Get Started**

Videos, blogs, articles, support & community – engage now!

My Job is Data Science

I Am Wolle

I‘m a data guy, not an ASR or HCI expert!

Wolfram Wingerath
Data Science

Research:
- Stream Processing
- Real-Time Databases
- NoSQL & Cloud Systems
- …

+

Practice:
- Web Caching
- Big Data Analytics
- Anger Management
- …

UHH Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Carl von Ossietzky Universität Oldenburg

Baqend

Look, No Hands!

Basic Voice Control

- **Symbols, Modifiers & Navigation:**

  What you say — What you get

  - Numbers, brackets, etc.: `one space air bat shift one` → `1_ab!`
- **Spelling** through a phonetic alphabet:
  - NATO alphabet: `alpha bravo charlie delta echo` → `abcde`
  - Optimized alphabet: `air bat cam drum each` → `abcde`
- **Command Management** for efficiency, e.g.:
  - Chaining: `paren close paren` `go left` `say hi` → `(hi)`

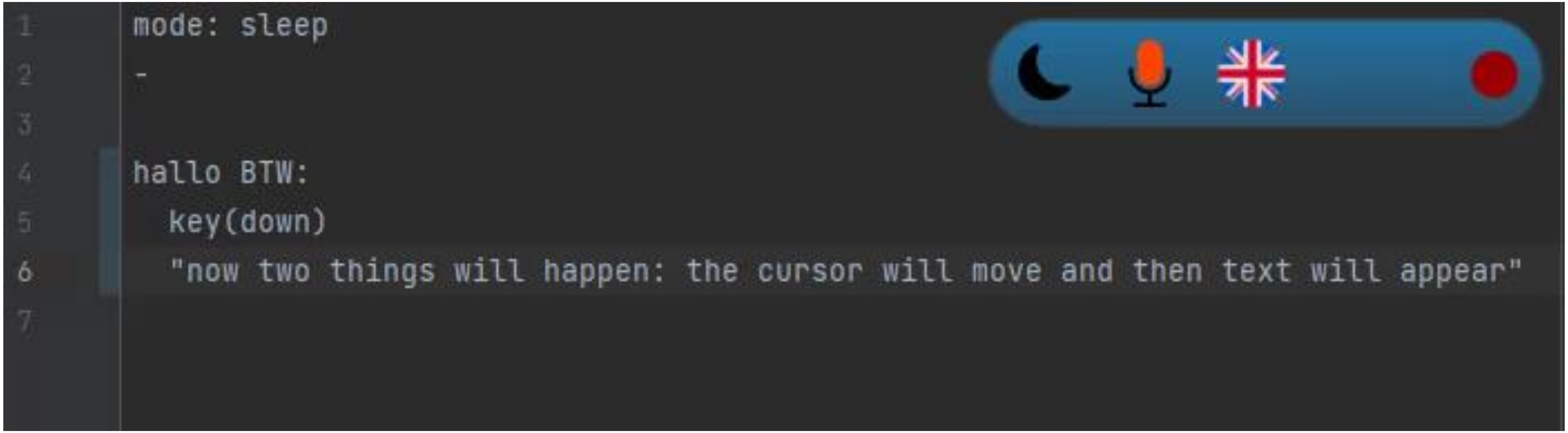
    1.) type: `()`

    2.) move cursor: ⇦

    3.) dictate: `hi`
- **On-the-fly grammar prototyping** through Python live reloading!

Live Demo

Dynamic Scripting (Live Demo)



```
mode: sleep
-

hallo BTW:
  key(down)
  "now two things will happen: the cursor will move and then text will appear"
```

HandsfreeCoding

- **Context-dependent behavior**, for example:
  - C#: `funky test funk` → `private void testFunk()`
  - JavaScript: `funky test funk` → `function testFunk()`
- **Intuitive IDE shortcuts** such as
  - `"run code"` instead of `<shift-f10>`
  - `"find usage"` instead of `<ctrl-alt-f7>`
- **Powerful templates**, `e.g.:`

```
action(user.code_state_if):
  insert("if () {}")
  key(left enter up end left left left)
```

Handsfree Coding: Cursorless

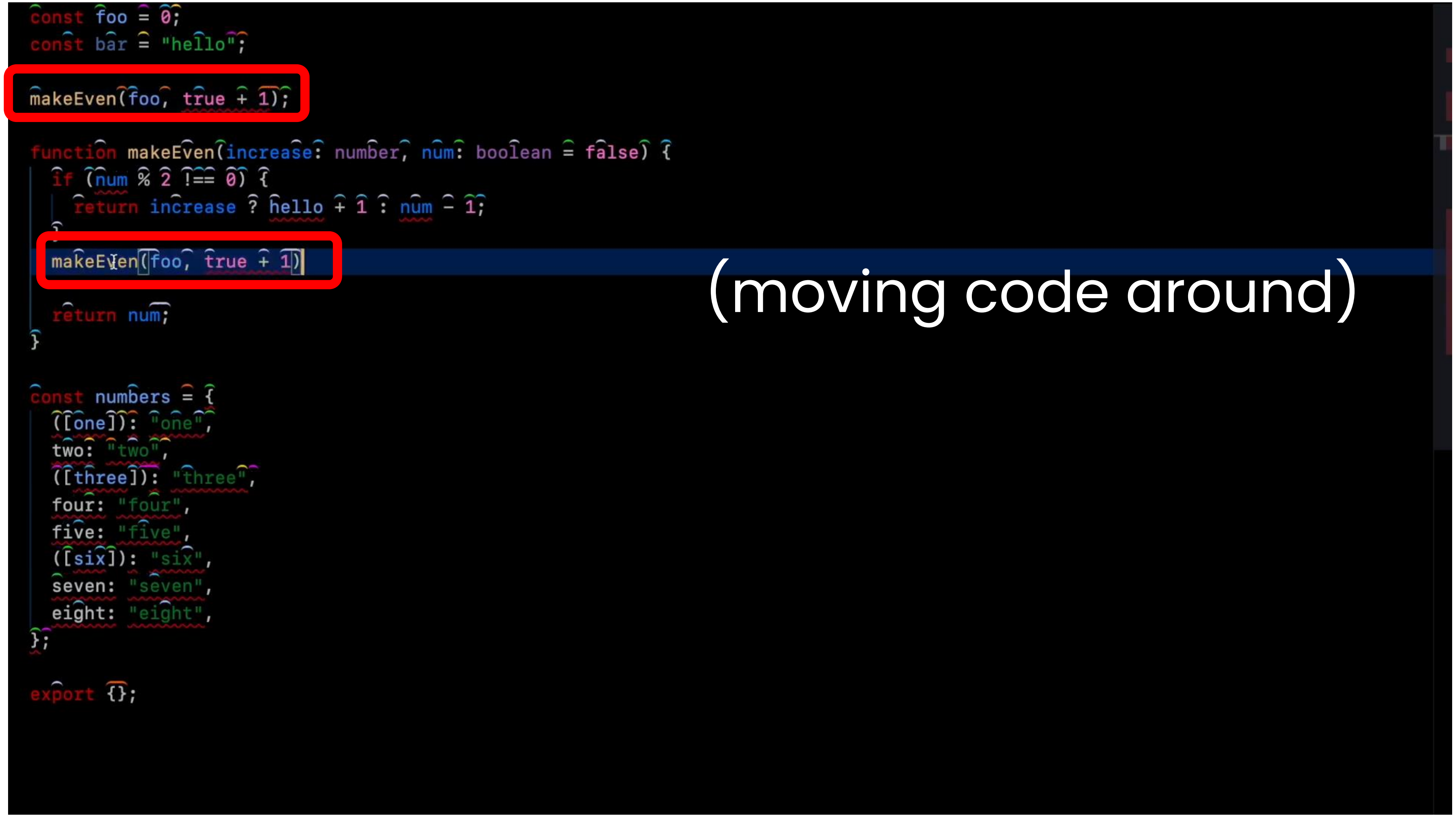
- Available on GitHub: github.com/cursorless-dev/
- VSCode extension
- Spoken language for **structural code editing**
  - Decorates every token on screen with a **mark**
  - tokens can be selected via combination of mark and **scope**
  - **actions** operate on the specified tokens

- Example:

`bring`	`call`	`vest`
action	scope	mark

(copy the function call with the marked „v“ to where my cursor is)

Handsfree Coding: Cursorless

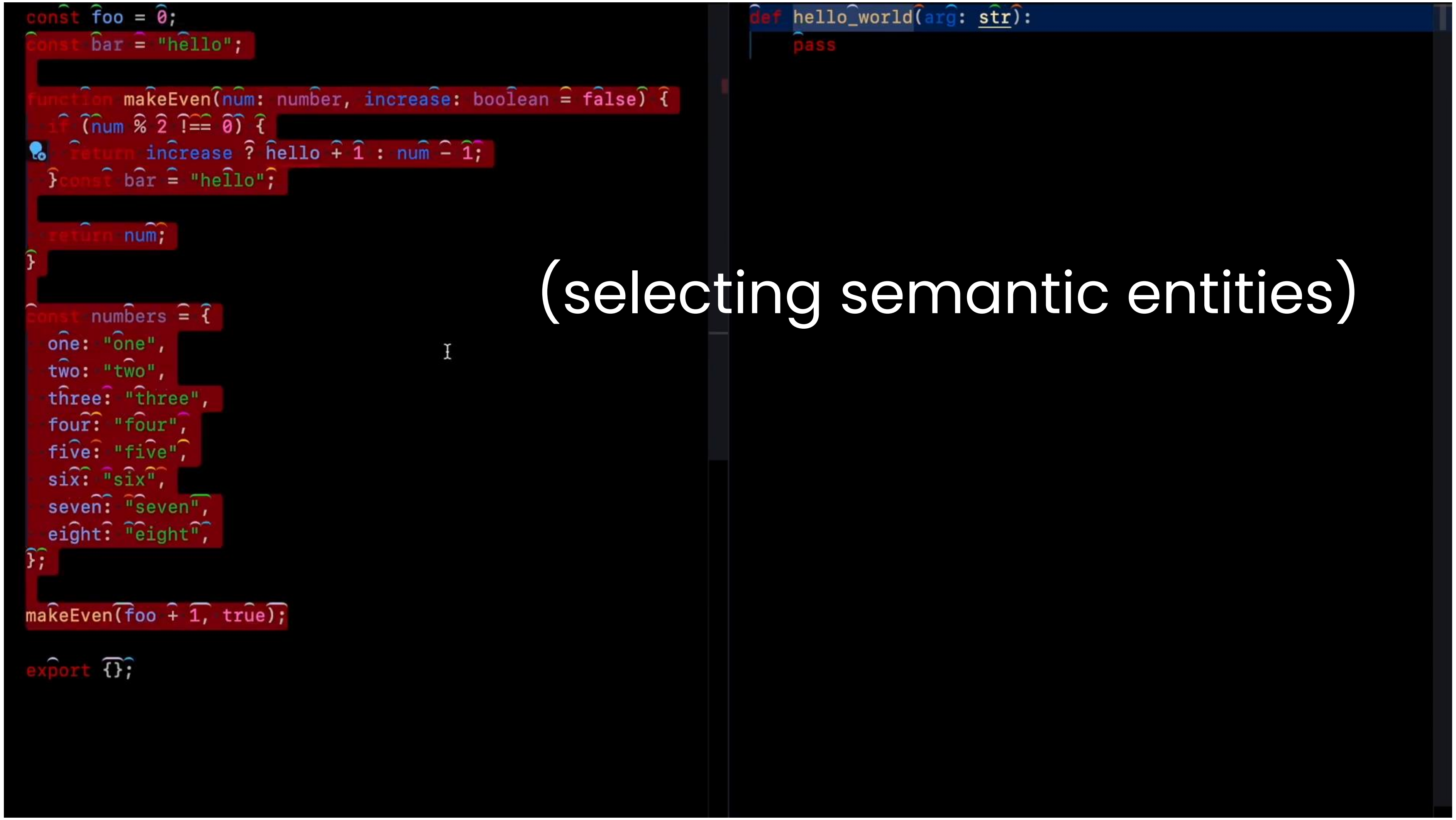


```
const foo = 0;
const bar = "hello";

makeEven(foo, true + 1);

function makeEven(increase: number, num: boolean = false) {
  if (num % 2 !== 0) {
    return increase ? hello + 1 : num - 1;
  }

  makeEven(foo, true + 1)

  return num;
}

const numbers = {
  ([one]): "one",
  two: "two",
  ([three]): "three",
  four: "four",
  five: "five",
  ([six]): "six",
  seven: "seven",
  eight: "eight",
};

export {};
```

(moving code around)

Pokey Rule. Old Cursorless Tutorial, YouTube (2021)

Handsfree Coding: Cursorless



```
const foo = 0;
const bar = "hello";

function makeEven(num: number, increase: boolean = false) {
  if (num % 2 !== 0) {
    return increase ? hello + 1 : num - 1;
  }const bar = "hello";

  return num;
}

const numbers = {
  one: "one",
  two: "two",
  three: "three",
  four: "four",
  five: "five",
  six: "six",
  seven: "seven",
  eight: "eight",
};

makeEven(foo + 1, true);

export {};
```

```
def hello_world(arg: str):
    pass
```

(selecting semantic entities)

Pokey Rule. Old Cursorless Tutorial, YouTube (2021)

Snappy Noise Control With Parrot

- Available on GitHub: github.com/chaosparrot/parrot.py
- Noise-controlled actions with latency <50ms
- Workflow
  (1) Record sounds
  (2) Train model for recognition
  (3) Map sounds to actions
- Compatible (and recommended in combination with) with other tooling:
  - Often used with Project IRIS (eye tracking)
  - Can be used to produce **Talon-compatible** models

Custom noises for
your Talon grammar!

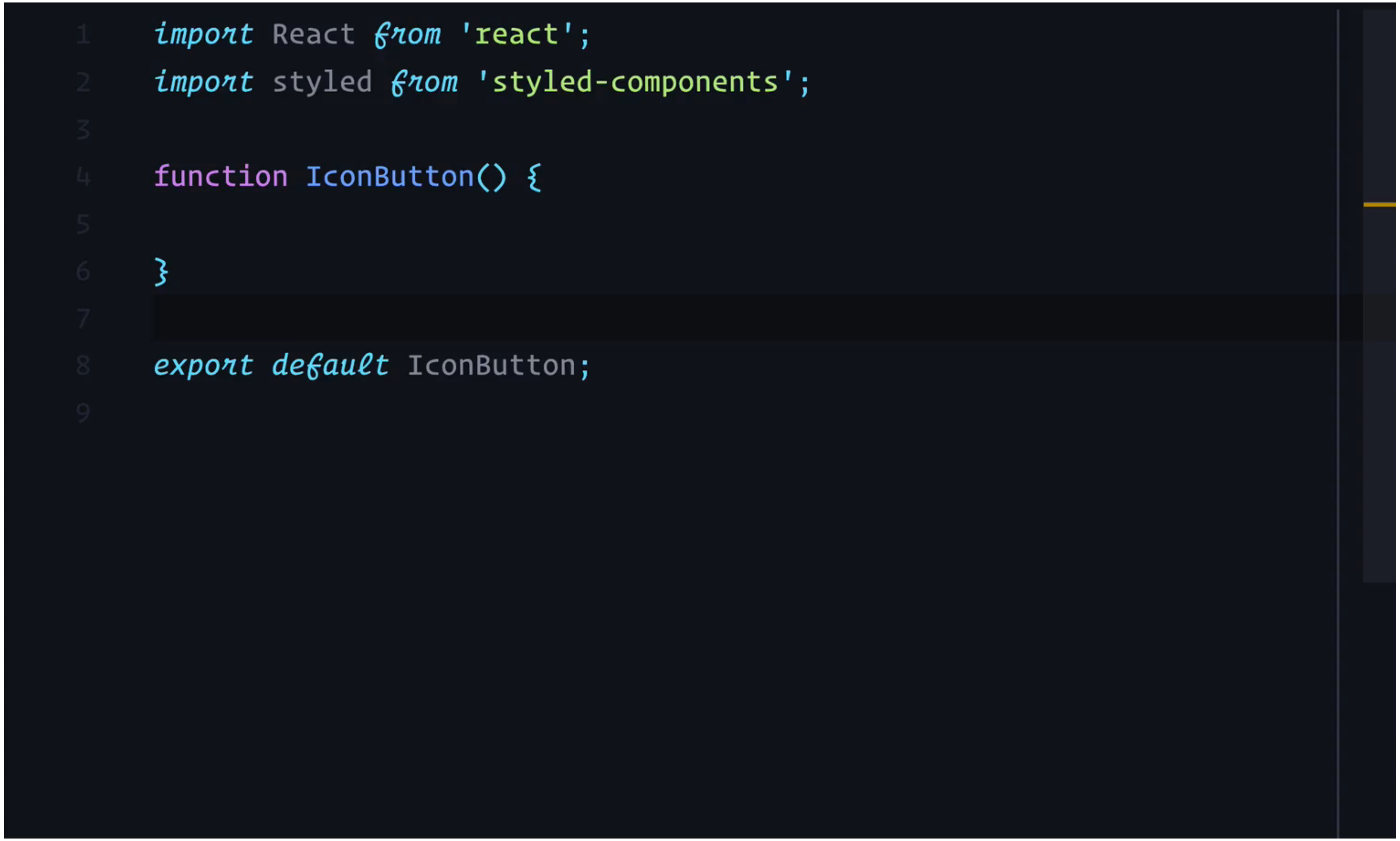
Eye Tracking & Noise Recognition

- **Calibration** for adjusting your eye tracker to your current position
- **Noises** for actions (e.g. clicking & right-clicking):
  - Extremely low latency (<50ms)
  - Talon currently supports `*pop*` & `*hiss*`
  - Custom noise models available via Parrot

github.com/chaosparrot/parrot.py

- **Different Modes** for convenience:
  - Zoom: (1) `*pop*` for zooming, (2) `*pop*` for clicking
  - Head tracking: eye gaze (jumps) + head movement (adjustment)
- **Debug** mode & camera overlay

Handsfree Coding: Talon



```
import React from 'react';
import styled from 'styled-components';

function IconButton() {

}

export default IconButton;
```

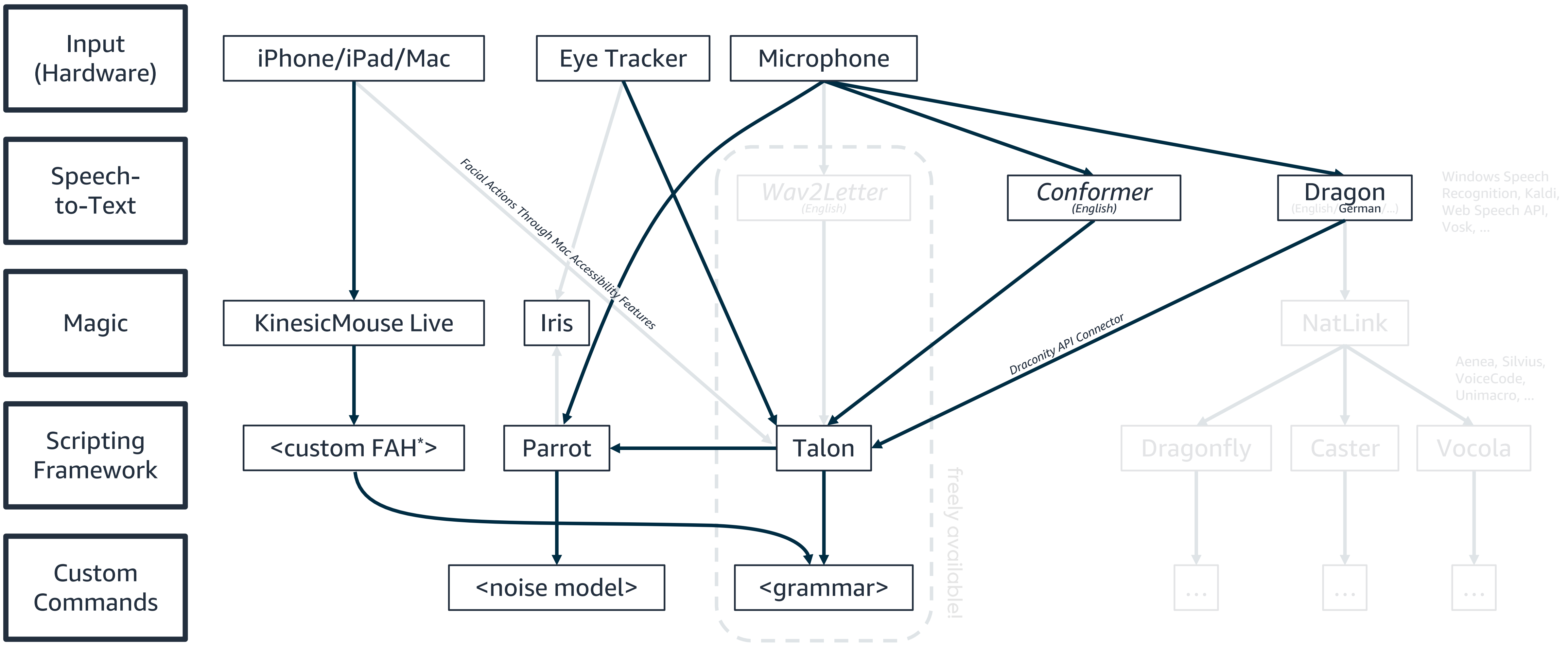
Josh W. Comeau. Hands-free Coding, Josh W. Comeau's Blog (2020)

Handsfree Gaming: Eyes + Face + Voice/Noise

wolle.science/twitch

The Base Setup

Popular Handsfree Coding Stacks: Overview



Input (Hardware)

Speech-to-Text

Magic

Scripting Framework

Custom Commands

iPhone/iPad/Mac

Eye Tracker

Microphone

Wav2Letter *(English)*

Conformer *(English)*

Dragon (English/German/...)

Windows Speech Recognition, Kaldi, Web Speech API, Vosk, ...

Facial Actions Through Mac Accessibility Features

KinesicMouse Live

Iris

NatLink

Draconity API Connector

Aenea, Silvius, VoiceCode, Unimacro, ...

<custom FAH*>

Parrot

Talon

Dragonfly

Caster

Vocola

<noise model>

<grammar>

...

...

...

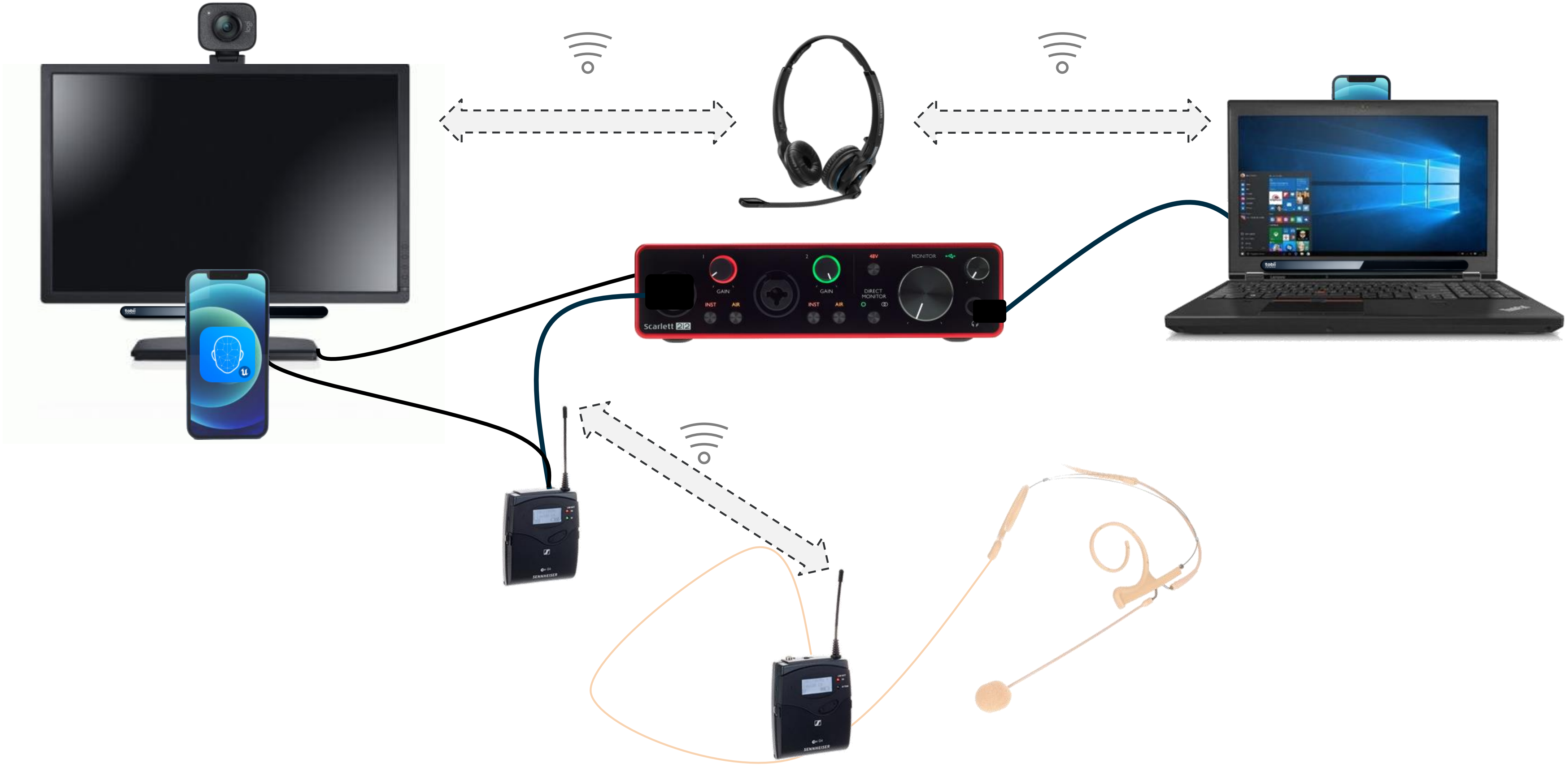
freely available!

*Facial Action Handling
Please note that this overview is NOT complete: On every level, there are MANY other options!

This overview was inspired by:
https://dictation-toolbox.github.io/dictation-toolbox.org/ (accessed: January 4, 2021)

Upgrades & Add-Ons

Multi-Computer Setup



Pitfalls & Challenges

Recognition Accuracy Issues

- **Microphone** determines accuracy!
  - *Build quality*: built-in < gaming headset < stage mic
  - *Positioning*: consistent, close to your mouth, away from all noise
  - *Mixed bag: Noise canceling* via hardware or software (e.g. RTX Voice)
- **Environment**: Minimize noise for you and annoyance for others!
  - Suspend ASR / mute mic accordingly (e.g. via push-to-talk pedal)
- **Homophones** should be avoided, e.g. through:
  - Grammar optimization to avoid ambiguity
  - Clear pronunciation

Potential Privacy Issues

- **Watch Your Tongue**: Passwords & confidential info may be leaked …
  - … through plain acoustics (beware *eavesdroppers*!)
  - … as they are stored your *command history*!
  - … to involved third parties (e.g. with *Web Speech*)
- **Watch Your Transmitter**: Wireless solutions are often not encrypted!
- **Watch Your Eyes**: Your eye movement may give away a lot

  → perhaps avoid continuous eye tracking ;-)

<insert eye tracking challenge joke here>

Workflow & Anger Management Issues

- **Beware the Trolls**: Having an audience generally does not help!

  → Prepare to hear „Format C“ from your colleagues a lot

- **Keep your calm**: Shouting at the computer will not help, either!

  → Stay in your neutral voice, even when raging inside ...

- **Avoid Voice Strain**: Find a comfortable way to speak A LOT!

  → e.g. use your natural voice & drink a lot of tea

- **Command chaining**: Anticipate what is going to happen!

  → Practice, practice, practice!

General Issues

- **Multilanguage support** is still in ist infancy

  → Non-English language models all have their problems

  → Designing command libraries for different languages means effort

- **Complex setup** with many moving parts:

  → Random stuff sometimes just happens, get used to it!

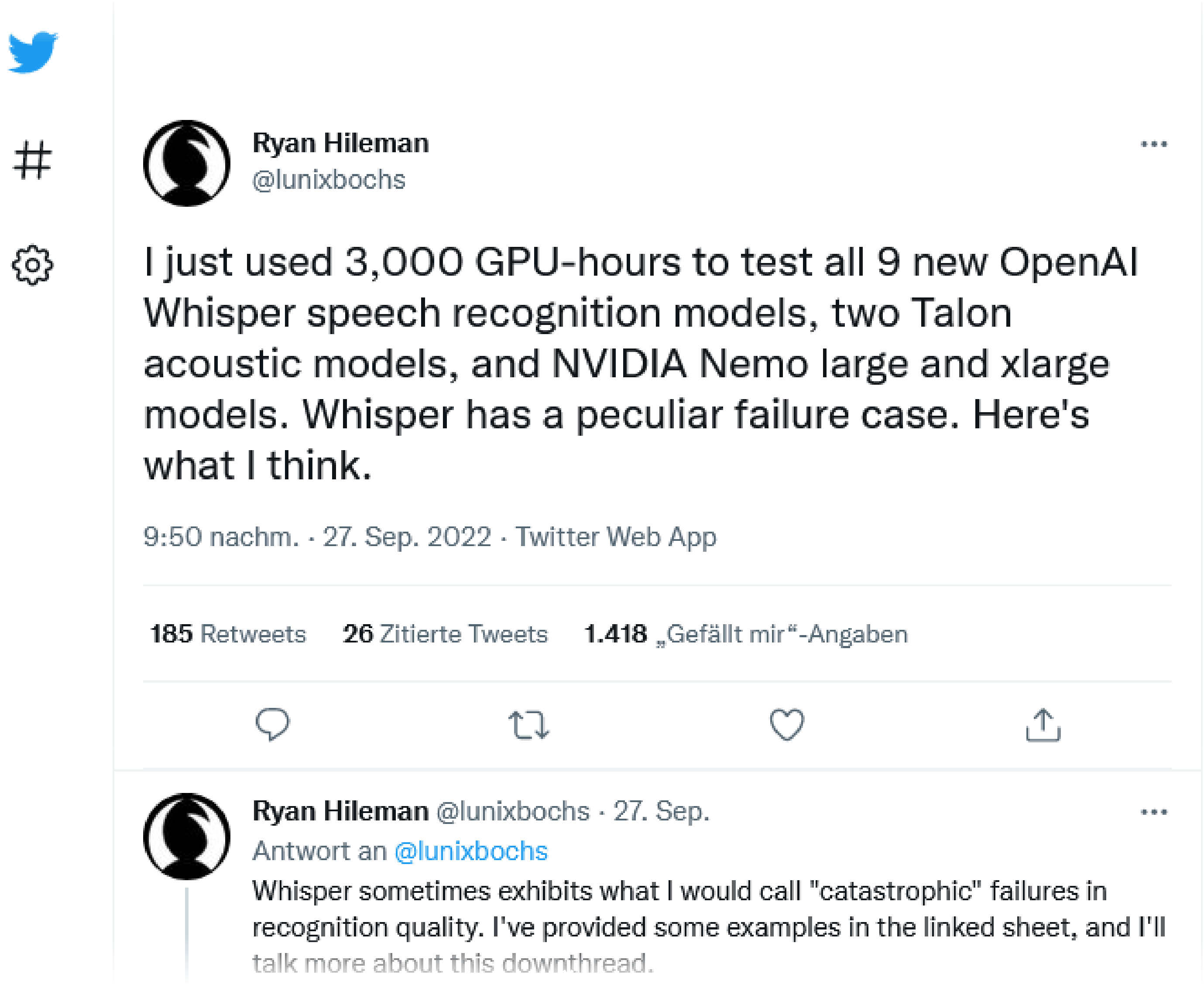
  → Fallback to manual input sometimes necessary …

- **MACHINE LEARNING!!!**

  → Models often reflect typical issues (data bias, data quality issues, …)

  → Sometimes you have to just hope for the best …

Model Issues



Ryan Hileman
@lunixbochs

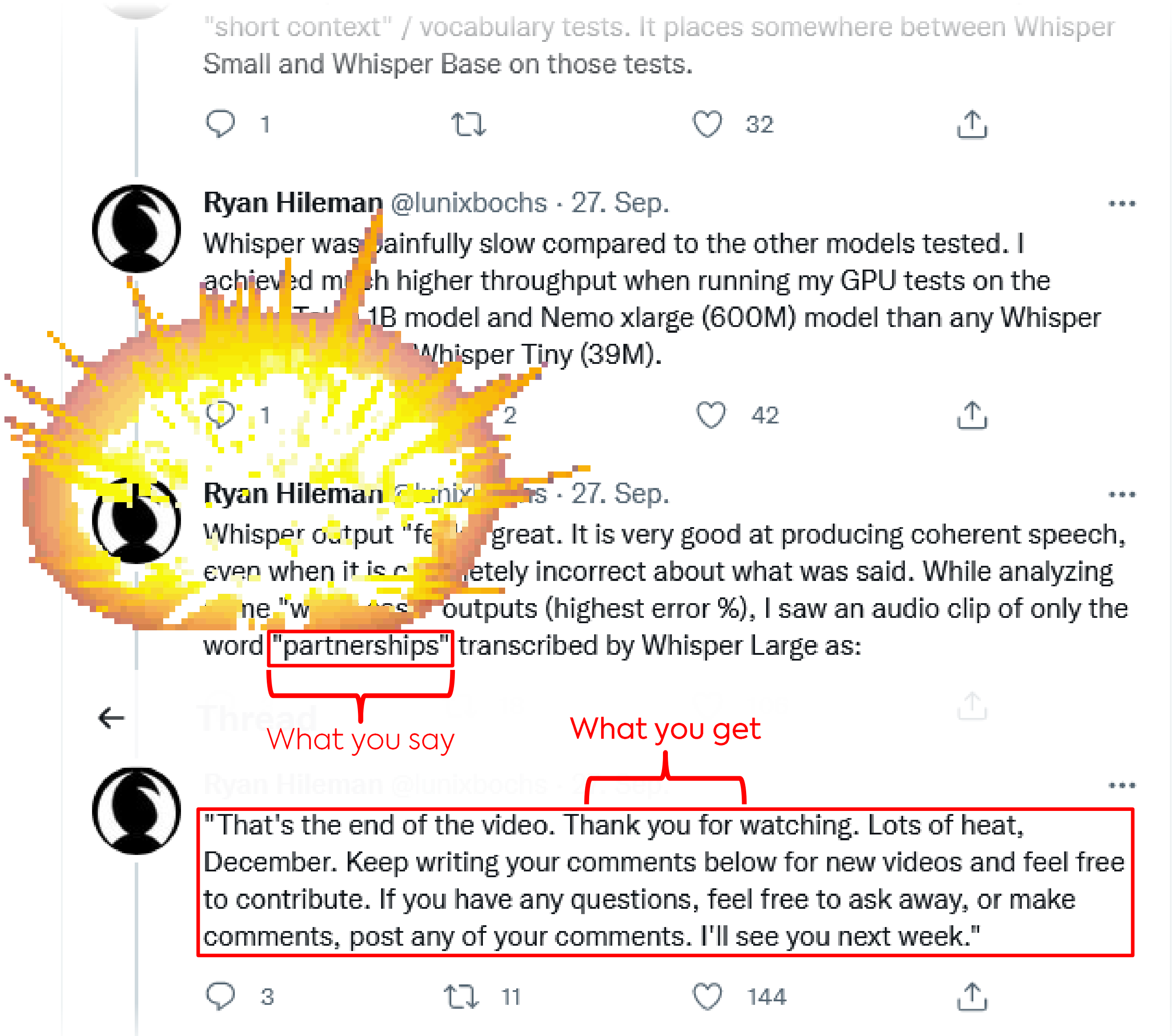
I just used 3,000 GPU-hours to test all 9 new OpenAI Whisper speech recognition models, two Talon acoustic models, and NVIDIA Nemo large and xlarge models. Whisper has a peculiar failure case. Here's what I think.

9:50 nachm. · 27. Sep. 2022 · Twitter Web App

185 Retweets 26 Zitierte Tweets 1.418 „Gefällt mir"-Angaben

Ryan Hileman @lunixbochs · 27. Sep.
Antwort an @lunixbochs
Whisper sometimes exhibits what I would call "catastrophic" failures in recognition quality. I've provided some examples in the linked sheet, and I'll talk more about this downthread.

Model Issues



"short context" / vocabulary tests. It places somewhere between Whisper Small and Whisper Base on those tests.

Ryan Hileman @lunixbochs · 27. Sep.

Whisper was painfully slow compared to the other models tested. I achieved much higher throughput when running my GPU tests on the [illegible] 1B model and Nemo xlarge (600M) model than any Whisper [illegible] Whisper Tiny (39M).

Ryan Hileman @lunix[illegible]s · 27. Sep.

Whisper output "fe[illegible]" great. It is very good at producing coherent speech, even when it is c[illegible]etely incorrect about what was said. While analyzing [illegible]ne "w[illegible]" outputs (highest error %), I saw an audio clip of only the word "partnerships" transcribed by Whisper Large as:

What you say

What you get

Ryan Hileman @lunixbochs · 27. Sep.

"That's the end of the video. Thank you for watching. Lots of heat, December. Keep writing your comments below for new videos and feel free to contribute. If you have any questions, feel free to ask away, or make comments, post any of your comments. I'll see you next week."

Why This is Still Worth All the Hassle

Productivity

- Speed up input-heavy tasks
- Faster navigation through easy-to-remember shortcuts

Convenience

- Intuitive interfaces
- Relieve your hands

Accessibility

Compensate handicaps:

- Injuries (e.g. broken hand)
- Repetitive stress injury (RSI)
- Cubital Tunnel Syndrome
- …

General Awesomeness

- Talk to your computer!!!

Helpful Resources & Outlook

Tooling Recommendations (Incomplete!)

- **Talon** (Free of Charge): talonvoice.com / talon.wiki
  - Voice coding for Win / Linux / Mac!
  - Starter Grammar (English): github.com/knausj85/knausj_talon
- parrot.py (noise control): github.com/chaosparrot/parrot.py
- Cursorless (code editing for VSCode): github.com/cursorless-dev
- Rango (handsfree browsing): github.com/david-tejada/rango
- Paid Upgrades:
  - Talon Premium Support: patreon.com/join/lunixbochs
  - Dragon Speech Recognition: nuance.com/dragon/

Alternatives: Speech Recognition

- Speech Recognition
  - WSR (Windows Speech Recognition): Built into Windows
  - Kaldi: github.com/kaldi-asr/kaldi
  - Vosk (ASR on mobile devices!): github.com/alphacep/vosk-api
  - Web Speech API (compatible with Talon through Chrome or Firefox)
- Scripting:
  - NatLink: sourceforge.net/p/natlink/
  - Dragonfly: github.com/dictation-toolbox/dragonfly
  - Caster: github.com/dictation-toolbox/Caster
  - Vocola (Voice Command Language): vocola.net

Recommended Talks

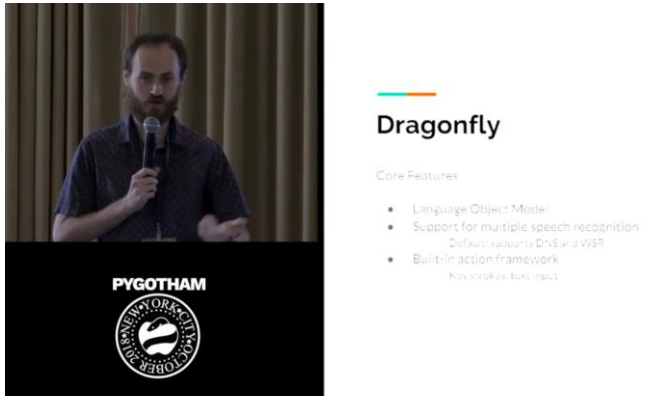
Sept 13-14, 2019
thestrangeloop.com

whois emily

- Software Engineer
- GitHub: @2shea
- Twitter: @yomilly
- I write code for Fastly

2/45

▷ Emily Shea. Voice Driven Development: Who needs a keyboard anyway?, Strange Loop (2019)

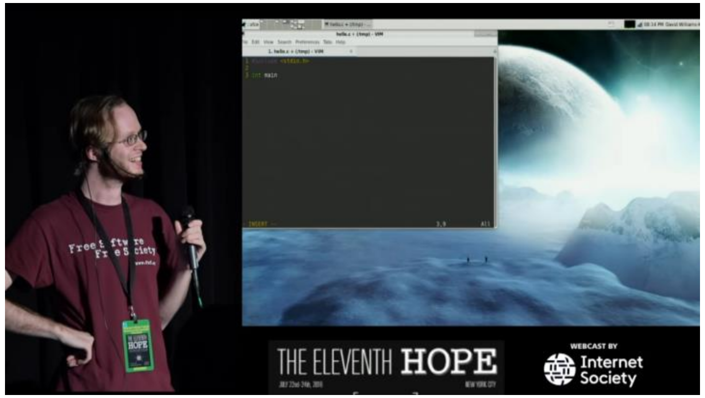


Dragonfly

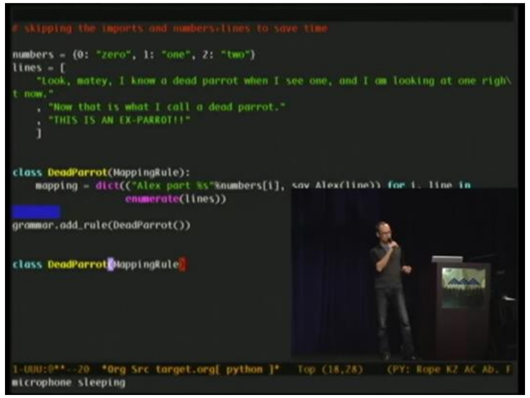
Core Features

- Language Object Model
- Support for multiple speech recognition
  - Default supports DNS and WSR
- Built-in action framework
  - Keystrokes, text input

▷ Boudewijn Aasman. Coding by Voice with Dragonfly, PyGotham (2018)



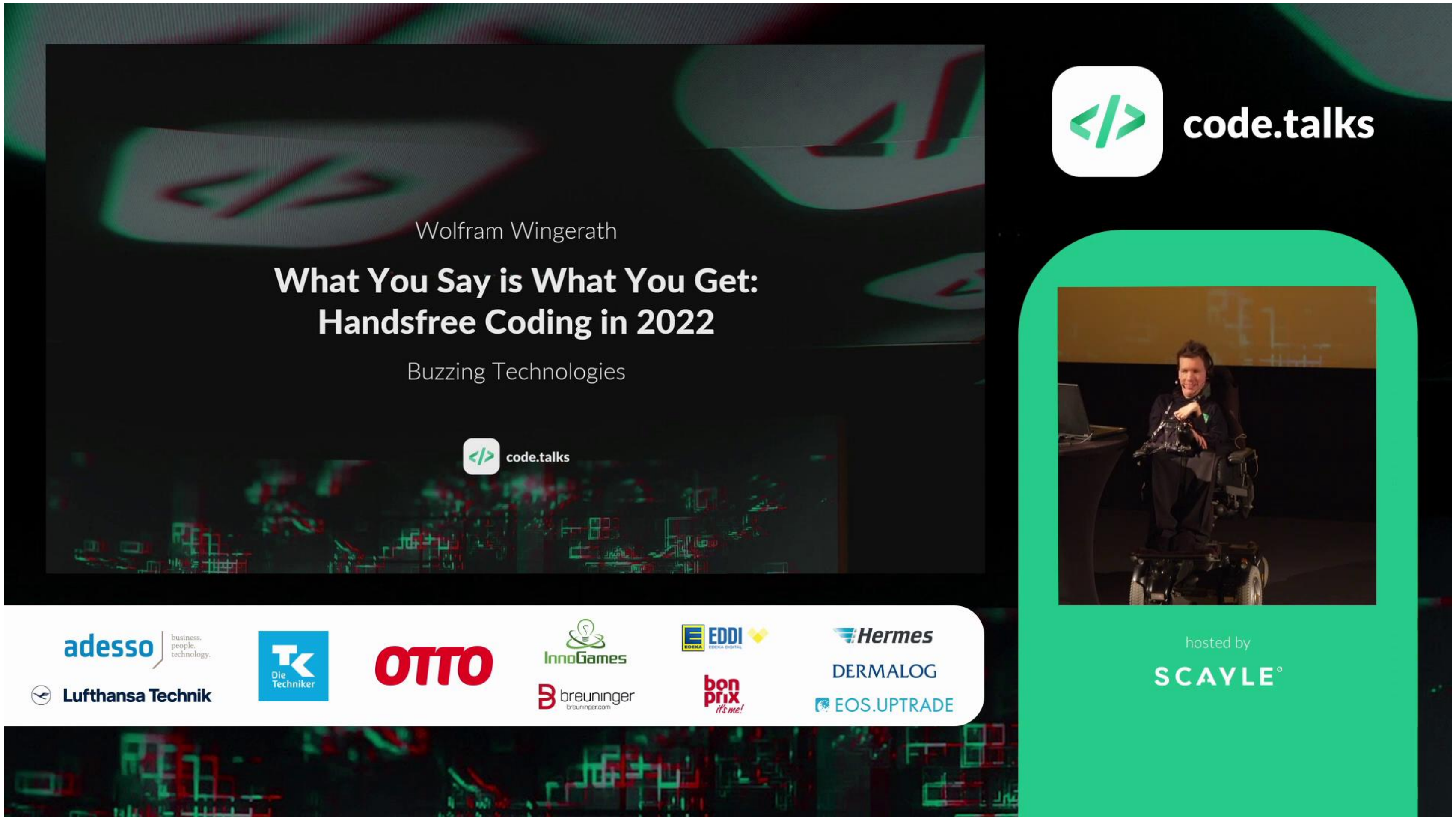
▷ David Williams-King. Coding by Voice with Open Source Speech Recognition, The Eleventh Hope (2016)



```
# skipping the imports and numbers+lines to save time

numbers = {0: "zero", 1: "one", 2: "two"}
lines = [
    "Look, matey, I know a dead parrot when I see one, and I am looking at one righ\
t now."
    , "Now that is what I call a dead parrot."
    , "THIS IS AN EX-PARROT!!"
    ]


class DeadParrot(MappingRule):
    mapping = dict(("Alex part %s"%numbers[i], say Alex(line)) for i, line in
                   enumerate(lines))

grammar.add_rule(DeadParrot())


class DeadParrot(MappingRule)



1-UUU:@**--20  *Org Src target.org[ python ]*   Top (18,28)    (PY: Rope K2 AC Ab. F
microphone sleeping
```

▷ Tavis Rudd. Using Python to Code by Voice, PyCon US (2013)

CodeTalks Video



Wolfram Wingerath

What You Say is What You Get: Handsfree Coding in 2022

Buzzing Technologies

code.talks

hosted by

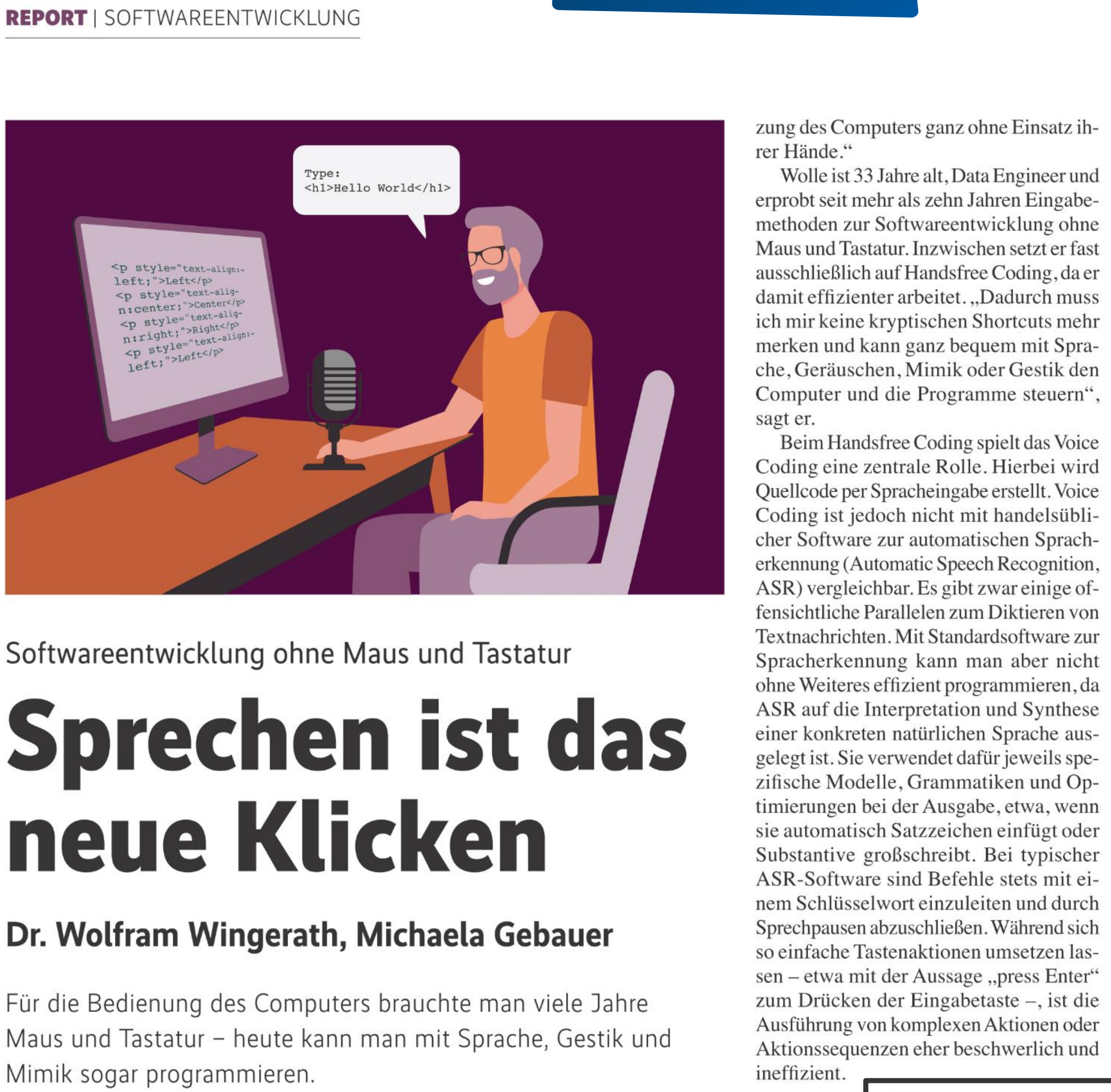
SCAYLE

Wolfram Wingerath. What You Say is What You Get: Hands-Free Coding in 2022, CodeTalks (2022)

Articles & Blogs

- Emily Shea: whalequench.club/
  - Talon user
  - Very good starter instructions
- James Stout: handsfreecoding.org/
  - Dragonfly user
  - Huge collection of relevant blog posts
- Josh W. Comeau (2020): joshwcomeau.com/blog/hands-free-coding/
- Dusty Phillips (2020): dusty.phillips.codes/2020/02/15/on-voice-coding/
- Max Gravenstein (2018): medium.com/hubabl/handsfree-fe70980f36b/

Heise Article



REPORT | SOFTWAREENTWICKLUNG

Softwareentwicklung ohne Maus und Tastatur

Sprechen ist das neue Klicken

Dr. Wolfram Wingerath, Michaela Gebauer

Für die Bedienung des Computers brauchte man viele Jahre Maus und Tastatur – heute kann man mit Sprache, Gestik und Mimik sogar programmieren.

zung des Computers ganz ohne Einsatz ihrer Hände.“

Wolle ist 33 Jahre alt, Data Engineer und erprobt seit mehr als zehn Jahren Eingabemethoden zur Softwareentwicklung ohne Maus und Tastatur. Inzwischen setzt er fast ausschließlich auf Handsfree Coding, da er damit effizienter arbeitet. „Dadurch muss ich mir keine kryptischen Shortcuts mehr merken und kann ganz bequem mit Sprache, Geräuschen, Mimik oder Gestik den Computer und die Programme steuern“, sagt er.

Beim Handsfree Coding spielt das Voice Coding eine zentrale Rolle. Hierbei wird Quellcode per Spracheingabe erstellt. Voice Coding ist jedoch nicht mit handelsüblicher Software zur automatischen Spracherkennung (Automatic Speech Recognition, ASR) vergleichbar. Es gibt zwar einige offensichtliche Parallelen zum Diktieren von Textnachrichten. Mit Standardsoftware zur Spracherkennung kann man aber nicht ohne Weiteres effizient programmieren, da ASR auf die Interpretation und Synthese einer konkreten natürlichen Sprache ausgelegt ist. Sie verwendet dafür jeweils spezifische Modelle, Grammatiken und Optimierungen bei der Ausgabe, etwa, wenn sie automatisch Satzzeichen einfügt oder Substantive großschreibt. Bei typischer ASR-Software sind Befehle stets mit einem Schlüsselwort einzuleiten und durch Sprechpausen abzuschließen. Während sich so einfache Tastenaktionen umsetzen lassen – etwa mit der Aussage „press Enter“ zum Drücken der Eingabetaste –, ist die Ausführung von komplexen Aktionen oder Aktionssequenzen eher beschwerlich und ineffizient.

Wolfram Wingerath, Michaela Gebauer: Sprechen ist das neue Klicken, iX 9/2021 (https://wingerath.cloud/2021/ix)

Closing Recommendations

- **Keep it simple**: Prioritize ease-of-use over efficiency at the start (in particular: get used to an existing grammar before optimizing it)

- **Keep it reasonable**: Try to find use cases that make sense for you (e.g.: I‘m not giving this talk handsfree, since I can use my index finger)

- **Keep it in mind**: Handsfree coding might save you one day (revisit this talk when you struggle with RSI, broken hand, etc.)

Thanks! So What Now?

Slack
talonvoice.slack.com ✓ Join the community!

Subscribe to the mailing list!

GI Initiative
handsfree-coding.gi.de ✓ Try out handsfree coding!

Patreon
patreon.com/lunixbochs ✓ Support Talon Development!

Slides Available at https://wolle.science

Wolfram Wingerath wolle@uol.de

Carl von Ossietzky
Universität
Oldenburg