

# What You **Say** is What You Get

## Handsfree Coding in 2023

IGER 2023, Bamberg, Germany

July 22, 2023

Wolle

Video & Slides Available at <https://wolle.science>

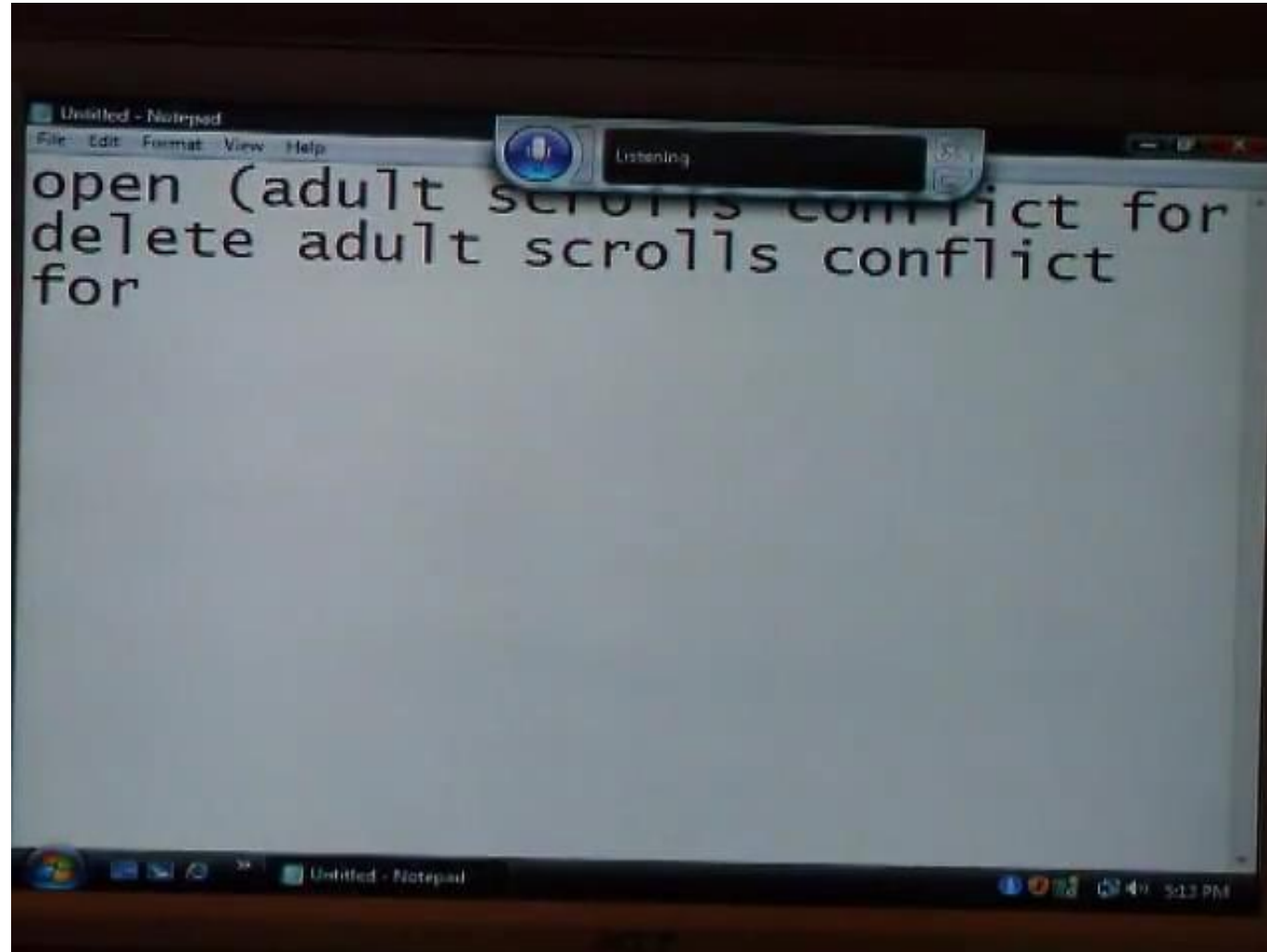
# It's **Simple**, Really!

The requirements:

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

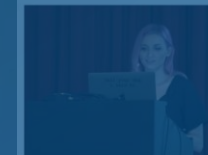


# Let Me Just Show You How **Easy** It is



# Let Me Just Show You How **Easy** It is

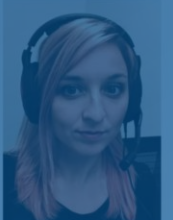
## Go Watch Emily's Talk!



Sept 13-14, 2019  
thestrangeloop.com

### whois emily

- Software Engineer
- GitHub: @zshea
- 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**?



# Where's the **Challenge**?

WSR, Dragon, ...

- **Automatic Speech Recognition (ASR):** optimized for natural languages
  1. Signal processing extracts features from audio recording
  2. Acoustic model recognizes phonemes
  3. Language model finds a matching sequence of words:
    - Default: Every utterance is interpreted as (spoken) text  
(Commands only through special keywords)
- **Voice Coding:** optimized for actions & programming languages
  - Default: Everything is interpreted as a command  
(Natural language through special keywords, e.g. `say <utterance>`)

Talon, Dragonfly ...

# Handsfree Coding: How It **Actually** Looks



Demo 4 (Nav)

Code Navigation / grep / search

1-UU-:0\*\* -ZO target.org All (1,0) \*E\* (Org KZ Hl Ab. F(11))

Using Dragonfly!

# **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

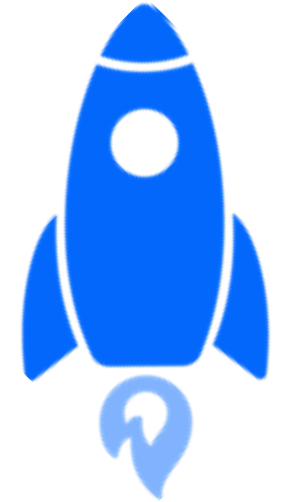
# I Am **Wolle**



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
- ...

Look,  
**No Hands!**



# Basic Voice Control

- **Symbols, Modifiers & Navigation:**

○ Numbers, brackets, etc.: one space air bat shift one → 1\_ab!

What you say: one space air bat shift one  
What you get: 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

# Live **Demo**: Dynamic Scripting

```
1 mode: sleep
2 -
3
4 hello Bamberg:
5   key(down)
6   "now we have a button press and text"
7
8
```

# Handsfree Coding

- **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: Talon

```
1  import React from 'react';
2  import styled from 'styled-components';
3
4  import Icon from '@components/Icon';
5
6  function IconButton({ icon, children }) {
7    return (
8      <Wrapper>
9        <Icon icon={icon} />
10       {children}
11     </Wrapper>
12   );
13 }
14
15 const Wrapper = styled.button`
16   background-color: var(--color-primary);
17   font-size: 2rem;
18   cursor: pointer;
```





# Handsfree Coding: Talon + **Cursorless**

- Available on GitHub: [github.com/cursorless-dev/](https://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: Talon + 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;
  }
  return num;
}

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

export {};
```

(moving code around)

# Handsfree Coding: Talon + Cursorless

```
const foo = 0;
makeEven(true, foo + 1);

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

  return num;
}

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

const bar = "hello";

export {};
```

(swapping arguments)

# Handsfree Coding: Talon + **Cursorless**

```
const foo = 0;
const bar = (loading...) function makeEven(num: number, increase?:
boolean): any
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 {};
```

(selecting semantic entities)

# Handsfree **Browsing**

+ **Vimium** browser extension: [vimium.github.io/](https://vimium.github.io/)

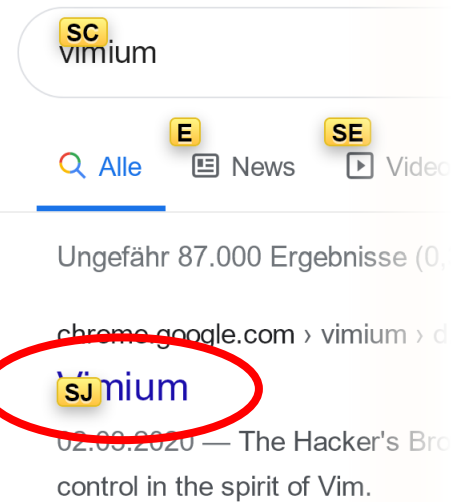
Example navigation without keyboard:

1. Show clickable links by pressing the `f` key
2. Press `sj` keys to click on Vimium link

+ Natively compatible with tools like **Talon**

- Simplify shortcuts with easy-to-remember utterances
- Optional: eye tracking + noise control to select links with your gaze

+ **Rango** browser extension for Talon: [github.com/david-tejada/rango](https://github.com/david-tejada/rango)



# Handsfree Browsing: Vimium

The screenshot shows a Google search for "Talon Voice". The search bar contains "Talon Voice" and has shortcuts for dictation (D), voice search (AD), and search (SD). Below the search bar are navigation shortcuts: E for "Alle", AE for "Videos", SE for "Shopping", F for "News", AF for "Bilder", SF for "Einstellungen", and G for "Suchfilter".

The search results show approximately 16,500,000 results in 0.46 seconds. The first result is "Talon Voice" from talonvoice.com, with shortcuts for dictation (D), voice search (AD), and search (SD). The description mentions it's for Windows and offers features like Voice Control and Noise Control. Below the title are shortcuts for "J" (John), "AJ" (AJ search), and "SJ" (SJ Angelog).

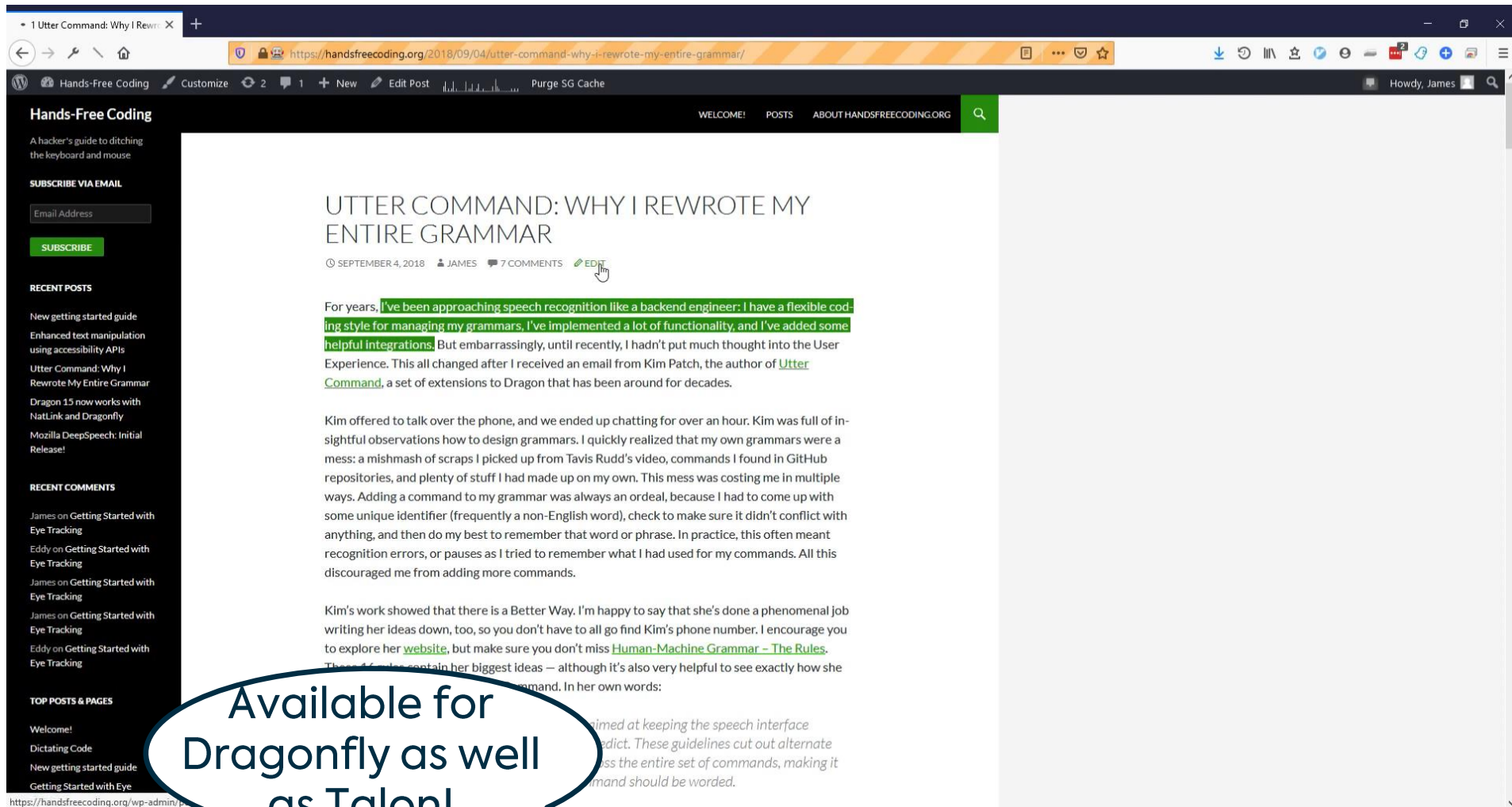
The second result is "Coding with voice dictation using Talon Voice - Josh Comeau" from joshwcomeau.com. The description is dated 09.12.2020 and mentions using Talon Voice for software development. Shortcuts for "K" (Joshwcomeau.com), "h" (blog), and "SK" (SK Seite übersetzen) are shown.

The third result is "Getting Started | Talon Wiki" from talonwiki.org. The description explains that Talon uses a speech recognition engine. Shortcuts for "L" (Talon.wiki), "g" (getting\_started), and "SL" (SL Seite übersetzen) are shown. Below the title are shortcuts for "M" (M public vs. Beta vs. Legacy), "SM" (SM figuration), "P" (P stfing out the basics), and "SP" (SP ic Usage).

The fourth result is "Talon · GitHub" from github.com. The description mentions it's the next-generation voice control. Shortcuts for "AS" (AS ub.com), "t" (talonvoice), and "SW" (SW Seite übersetzen) are shown.

At the bottom, there is a "Videos" section with a search bar containing "SW" and a video thumbnail for "Talon Voice Control: basics". A tooltip "Open link in current tab" is visible over the video thumbnail.

# Handsfree Browsing: Gaze OCR



James Stout, [Gaze OCR: Talon support and 10 new features!](#), Hands-Free Coding Blog (2022)



James Stout, [Say what you see: Efficient UI interaction with OCR and gaze tracking](#), Hands-Free Coding Blog (2020)

# Snappy Noise Control With **Parrot**

- Available on GitHub: [github.com/chaosparrot/parrot.py](https://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!



# Handsfree Gaming: **Noise** Recognition



# 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
- **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

[github.com/chaosparrot/parrot.py](https://github.com/chaosparrot/parrot.py)

# Handsfree **Gaming**: Head + Eye Tracking



# Handsfree Gaming: Facial Actions



# Handsfree **Gaming**: Eyes + Face + Voice/Noise

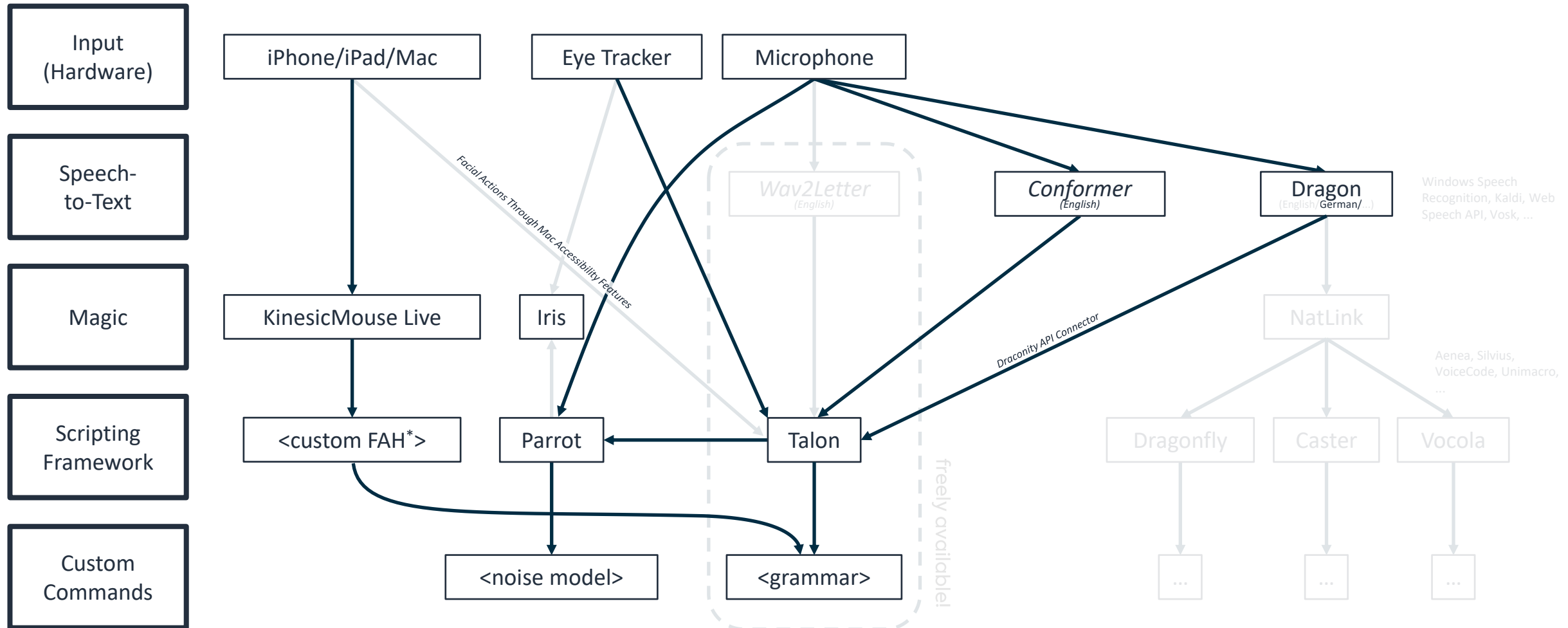


[wolle.science/twitch](https://www.twitch.tv/wolle.science)

A professional microphone is centered in the foreground, slightly out of focus. Behind it is a blurred laptop screen displaying a webpage with a grid of images. The entire image has a blue color overlay.

# The Base **Setup**

# Popular Handsfree Coding **Stacks**: Overview



\*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)

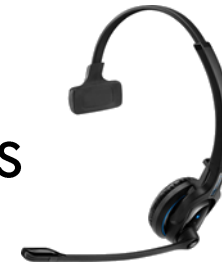


**Upgrades &  
Add-Ons**



# Supplementary Equipment

- **High-quality (!) microphone:** Get one now!
  - A wired (or good Bluetooth!) headset
  - Steno-mask: For noisy places & special looks
  - XLR mic for maximum accuracy



Sennheiser MB Pro 1/2



DPA 4188

→ + sender/receiver for max. convenience

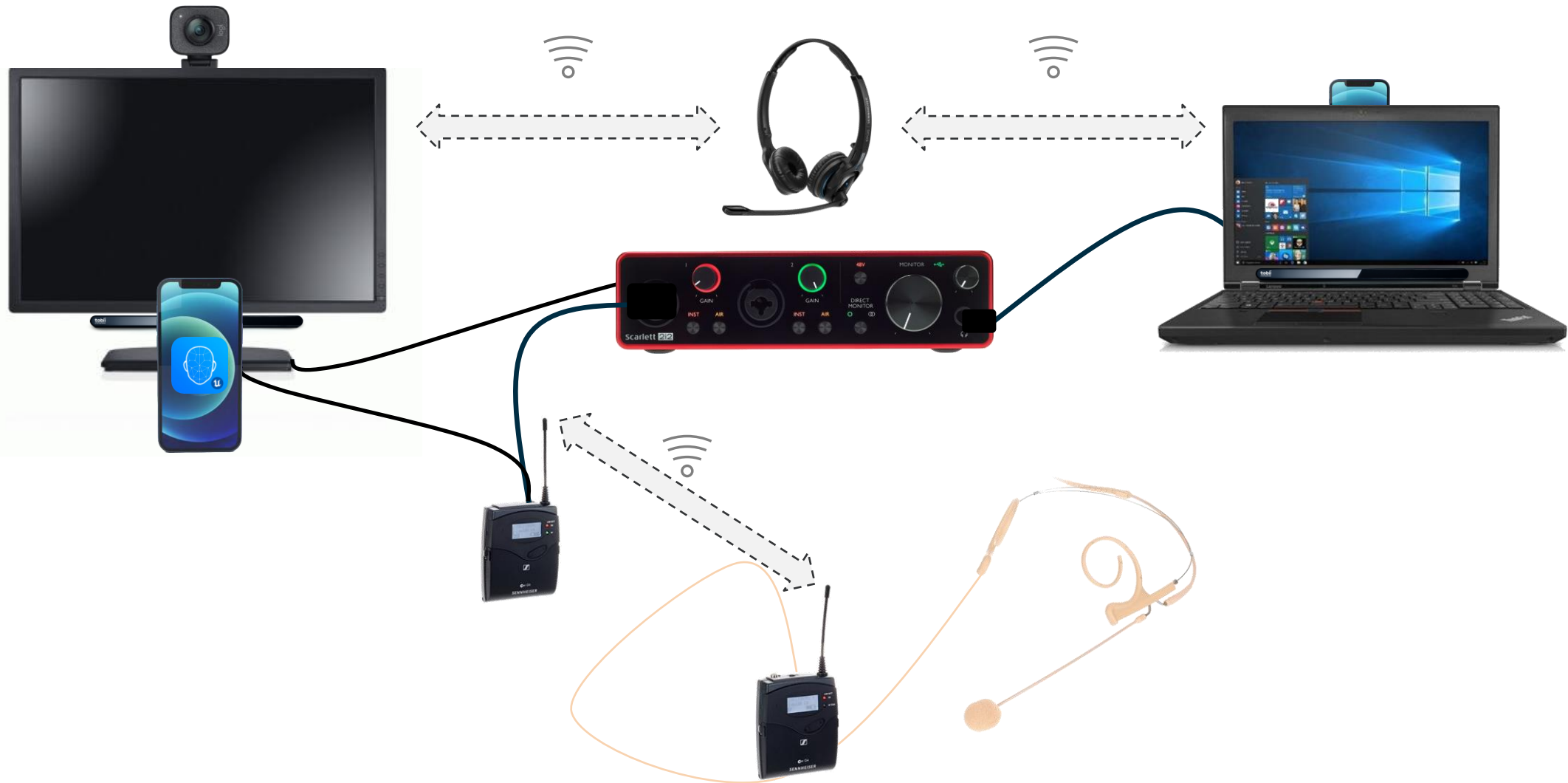
- **Foot pedal** for push-to-talk
- **Eye tracker:** Tobii 4C & 5 supported by Talon
  - Multi-monitor support coming (?)



Sennheiser EW 112P G4 E-Band



# Multi-Computer Setup



A person is running on a sidewalk, captured in a side profile. They are wearing a dark top, red shorts, and blue sneakers. A banana peel is on the ground just ahead of their right foot. The scene is overlaid with a dark blue tint. The text 'Pitfalls & Challenges' is written in white on the right side of the image.

# Pitfalls & Challenges

# Recognition **Accuracy** Issues

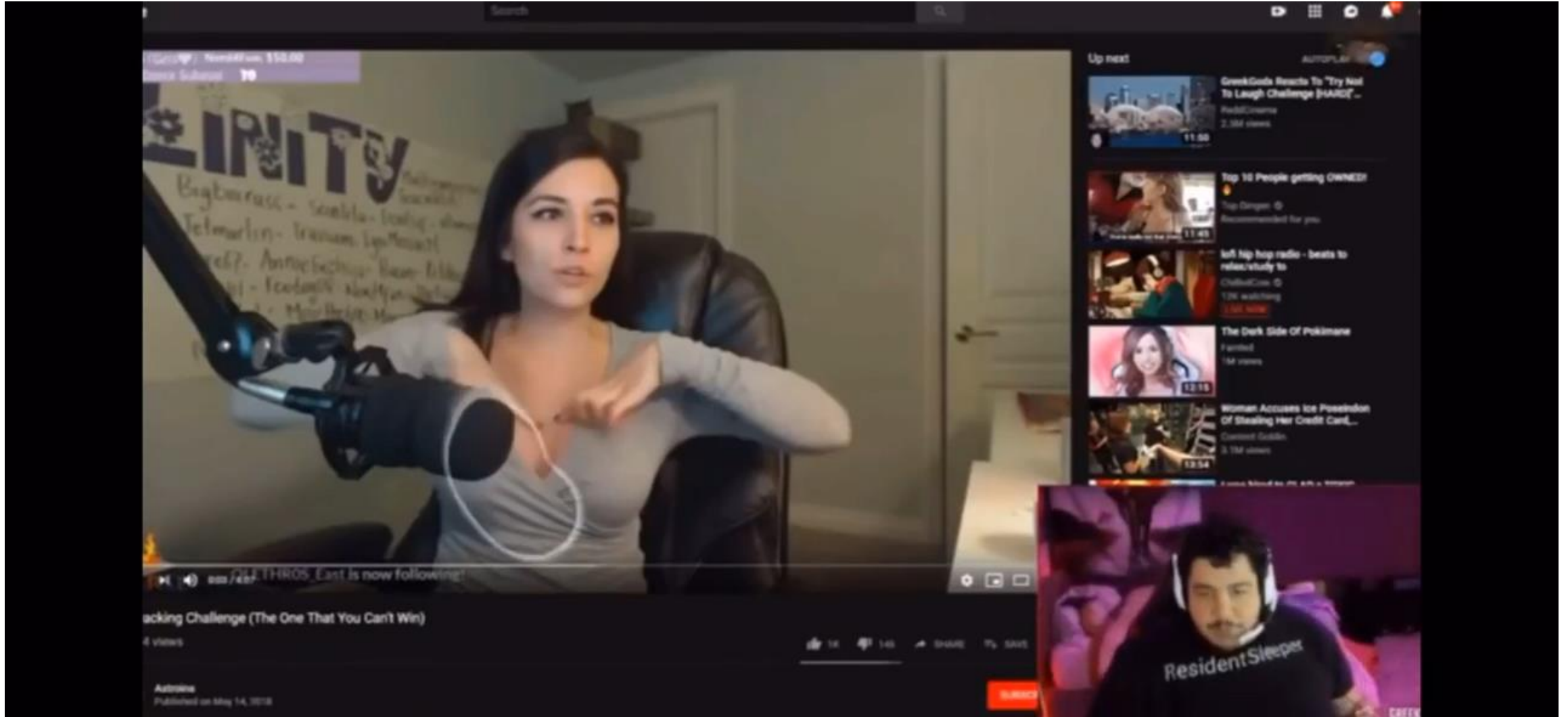
- **Microphone** determines accuracy!
  - *Build quality*: built-in < gaming headset < stage mic
  - *Positioning*: consistent, close to your mouth, away from all noise
  - *Mixed bag*: *Noise canceling* via hardware or software (e.g. RTX Voice)
- **Environment**: Minimize noise for you and annoyance 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>

# Potential Privacy Issues



# 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 its 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**  
@linuxbochs



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** @linuxbochs · 27. Sep.  
Antwort an @linuxbochs



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.

1



32



**Ryan Hileman** @linuxbochs · 27. Sep. ...

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

1



42



**Ryan Hileman** @linuxbochs · 27. Sep. ...

Whisper output "feels" great. It is very good at producing coherent speech, even when it is completely incorrect about what was said. While analyzing some "worst case" outputs (highest error %), I saw an audio clip of only the word "partnerships" transcribed by Whisper Large as:

← Thread  
What you say

What you get



**Ryan Hileman** @linuxbochs · 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."

3



144



# Model Issues

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

1



32



**Ryan Hileman** @linuxbochs · 27. Sep. ...

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

1



42



**Ryan Hileman** @linuxbochs · 27. Sep. ...

Whisper output "feels" great. It is very good at producing coherent speech, even when it is completely wrong about what was said. While analyzing some "worse case" examples (10%), I saw an audio clip of only the word "part" as transcribed by Whisper. The audio clip was as:

When you say "What do you get?"



**Ryan Hileman** @linuxbochs · 27. Sep. ...

"That's the end of the video. Thank you for watching. Lots of heat, December. Keep watching 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."

3



144



# 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!!!

It's **Awesome!**



A person with long hair is seen from behind, looking out at a harbor at night. In the background, several large cranes and ships are visible, illuminated by lights. The water in the foreground is dark with some reflections. The overall scene is dimly lit, with a blue and dark color palette.

# Helpful Resources & **Outlook**

# Tooling **Recommendations** (Incomplete!)



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

# Alternatives: **Speech** Recognition

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

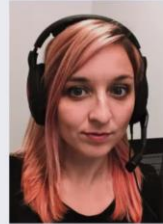
# Recommended Talks



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)



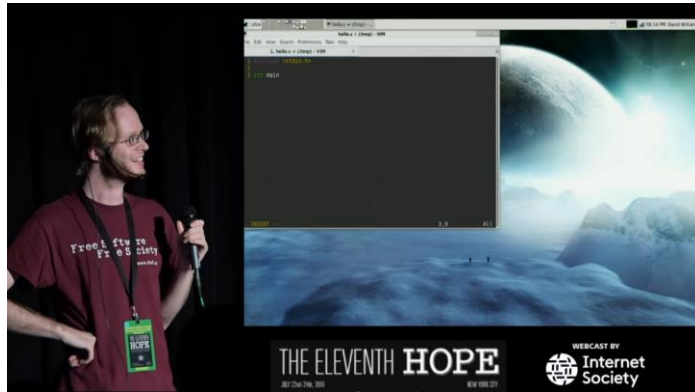
## Dragonfly

Core Features

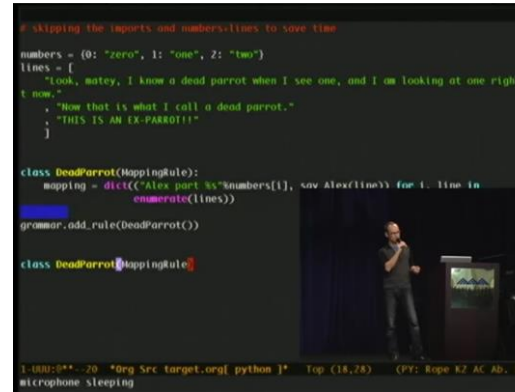
- Language Object Model
- Support for multiple speech recognition
  - Default supports DVS and WSR
- Built-in action framework
  - Raw strokes text input



Boudewijn Aasman. [Coding by Voice with Dragonfly, PyGotham](#) (2018)



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



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

code.talks

adesso | business. people. technology. | Die Techniker | OTTO | InnoGames | EDDI | Hermes | DERMALOG | Lufthansa Technik | breuninger | bonprix | EOS.UPTRADE

hosted by  
**SCAYLE**

The video player interface features a dark background with a glowing green and red grid pattern. The title and speaker information are centered in white text. A small code.talks logo is positioned below the title. The bottom of the player contains a row of sponsor logos. On the right side, there is a vertical green bar with a rounded top containing a photo of Wolfram Wingerath in a wheelchair, with the text 'hosted by SCAYLE' below it.

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

# Articles & Blogs

- Emily Shea: [whalequench.club/](https://whalequench.club/)
  - Talon user
  - Very good starter instructions
- James Stout: [handsfreecoding.org/](https://handsfreecoding.org/)
  - Dragonfly user
  - Huge collection of relevant blog posts
- Josh W. Comeau (2020): [joshwcomeau.com/blog/hands-free-coding/](https://joshwcomeau.com/blog/hands-free-coding/)
- Dusty Phillips (2020): [dusty.phillips.codes/2020/02/15/on-voice-coding/](https://dusty.phillips.codes/2020/02/15/on-voice-coding/)
- Max Gravenstein (2018): [medium.com/hubabl/handsfree-fe70980f36b/](https://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](https://wingerath.cloud/2021/ix), 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](https://talonvoice.slack.com)



**Ask questions!**  
**Enjoy the community!**

Subscribe to the mailing list!

**GI Initiative**  
[handsfree-coding.gi.de](https://handsfree-coding.gi.de)



**Try out handsfree coding!**

**Patreon**  
[patreon.com/lunixbochs](https://patreon.com/lunixbochs)



**Support Talon Development!**

Video & Slides Available at <https://wolle.science>

Wolfram Wingerath    [wolle@uol.de](mailto:wolle@uol.de)