What You **Say** is What You Get
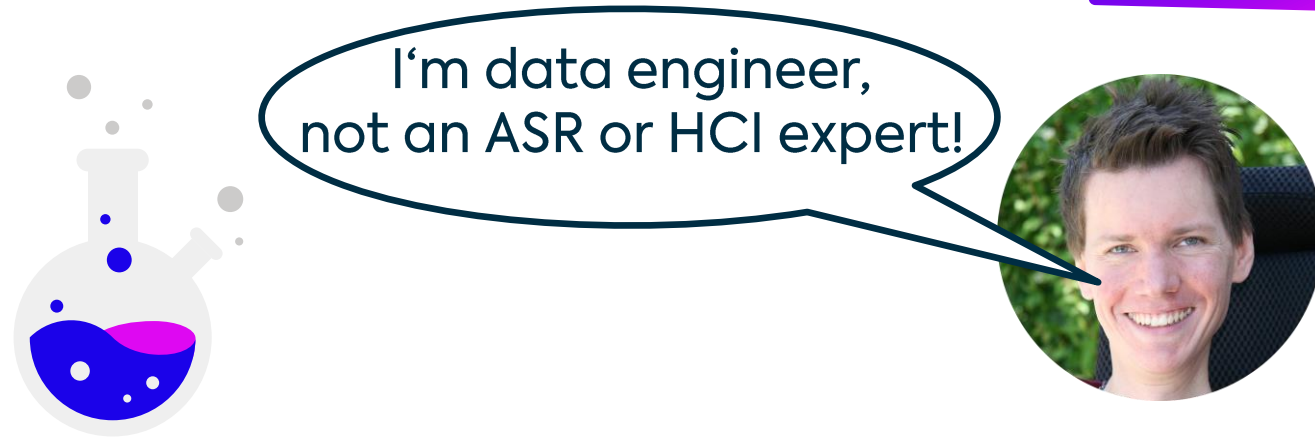
Handsfree Coding in 2021

ARIC Brown Bag Session

Wolfram Wingerath

June 3, 2021

Baqend

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'S GOOOOOO!!!!!!

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

Where's the **Challenge**?

WSR, Dragon, ...

- **Automatic Speech Recognition (ASR)**: optimized for <u>natural</u> languages

    1. Signal processing extracts features from audio recording

    2. Acoustic model recognizes phonemes

    3. Language model finds a matching sequence of words:

    → <u>Default</u>: Every utterance is interpreted as (spoken) text

    (Commands only through special keywords)

Caster, Talon, ...

- **Voice Coding**: optimized for <u>actions & programming</u> languages

    → <u>Default</u>: Everything is interpreted as a command

    (Natural language through special keywords, e.g. `say <utterance>`)

Where's the Challenge ?

Outline: What This Talk is Going to Cover

1 **My Personal Background**

As data engineer at Baqend, 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!

Look,
No Hands!

Basic Voice Control (1/3)

- **Actions & Symbols:**
  - Numbers, brackets, etc.: `one space paren bang slash` ➔ `1_(!/`
  - Modifiers, e.g.: `shift touch` **or** `hold alt` **or** `release control`
- **Spelling** through a phonetic alphabet:
  - NATO alphabet: `alpha bravo charlie delta echo` ➔ `abcde`
  - Optimized alphabet: `air bat cam drum each` ➔ `abcde`
- **Typing**:
  - Dictation: `say hey comma you` ➔ `hey, you`
  - Type recognized text: `phrase hey comma you` ➔ `hey comma you`

Basic Voice Control (2/3)

- **Navigation**, e.g.:

  o Cursor movement: `go left` **or** `go way up` **or** `home`

  o Selection: `mark left` **or** `mark way up` **or** `mark all`

- **Formatters**, e.g.:

  o Camel case: `camel hey you` → `heyYou`

  o Dashed: `kebab hey you` → `hey-you`

  o Dotted: `dotted hey you` → `hey.you`

  o Without whitspace: `smash hey you` → `heyyou`

  o Uppercase: `uppercase hey you` → `HEY YOU`

Basic Voice Control (3/3)

- **Command History & Help:**

  - Manage history: `command history` **or** `command history clear`

  - Show & hide help: `help active` **or** `help close`

- **Application management**, e.g.:

  - Launch & focus: `launch Firefox` **or** `focus Firefox`

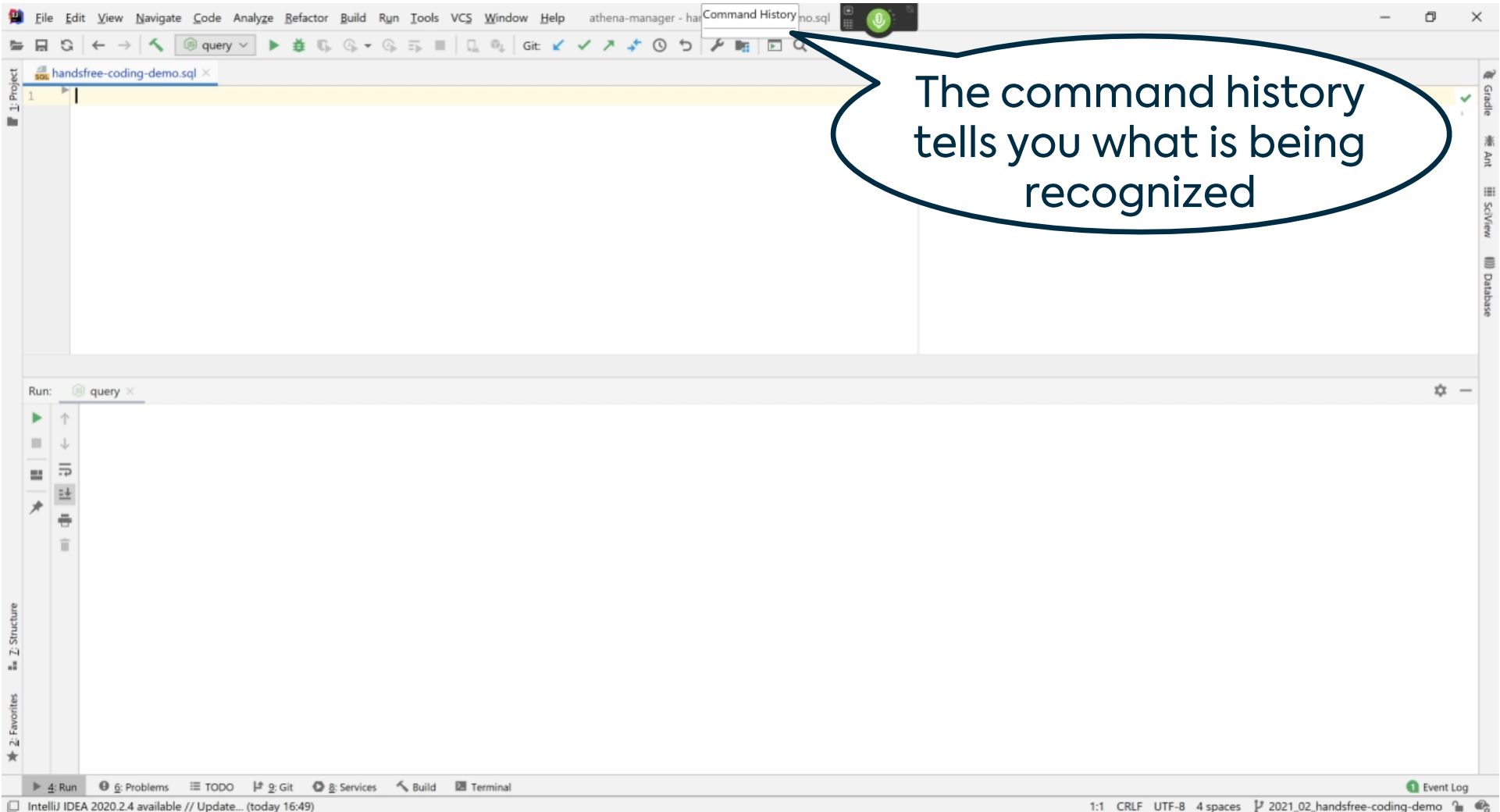- **Command Management** for efficiency, e.g.:

  - Repetition: `one third` ➜ `111`

  - Chaining: `paren close paren` `go left` `say hi` ➜ `(hi)`

    1.) type: 2.) move cursor: 3.) dictate:

    `()`   `⇐`   `hi`

Live Demo

Handsfree **Analytics** in Action

The command history tells you what is being recognized

Handsfree Software Development

- **Different modes** for different languages, 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)
```

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
  - Currently `*pop*` & `*hiss*`, more coming soon
  - Personal experience: `*hiss*` gets triggered easily by words with „s"

- **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

> You can help by providing training data: noise.talonvoice.com/

Noise Recognition: Extremely Snappy

Handsfree Software **Development**
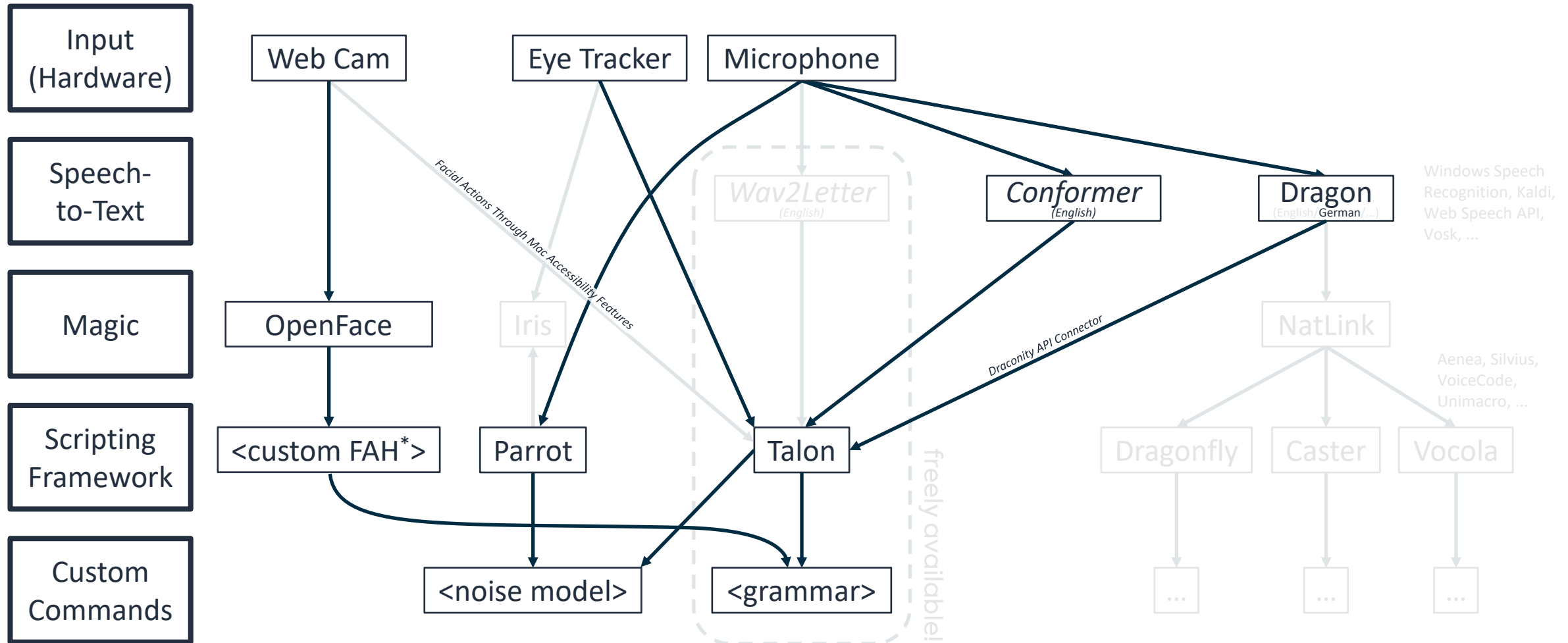
```
1    import React from 'react';
2    import styled from 'styled-components';
3
4    function IconButton() {
5
6    }
7
8    export default IconButton;
9
```

The Base Setup

Popular Handsfree Coding **Stacks** : Overview

Input (Hardware)	Web Cam	Eye Tracker	Microphone
Speech-to-Text		*Wav2Letter* *(English)*	*Conformer* *(English)* Dragon *(English, German/...)*
Magic	OpenFace	Iris	NatLink
Scripting Framework	<custom FAH*>	Parrot	Talon Dragonfly Caster Vocola
Custom Commands		<noise model>	<grammar>

Facial Actions Through Mac Accessibility Features

Draconity API Connector

freely available!

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

Aenea, Silvius, VoiceCode, Unimacro, ...

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

Upgrades & Add-Ons
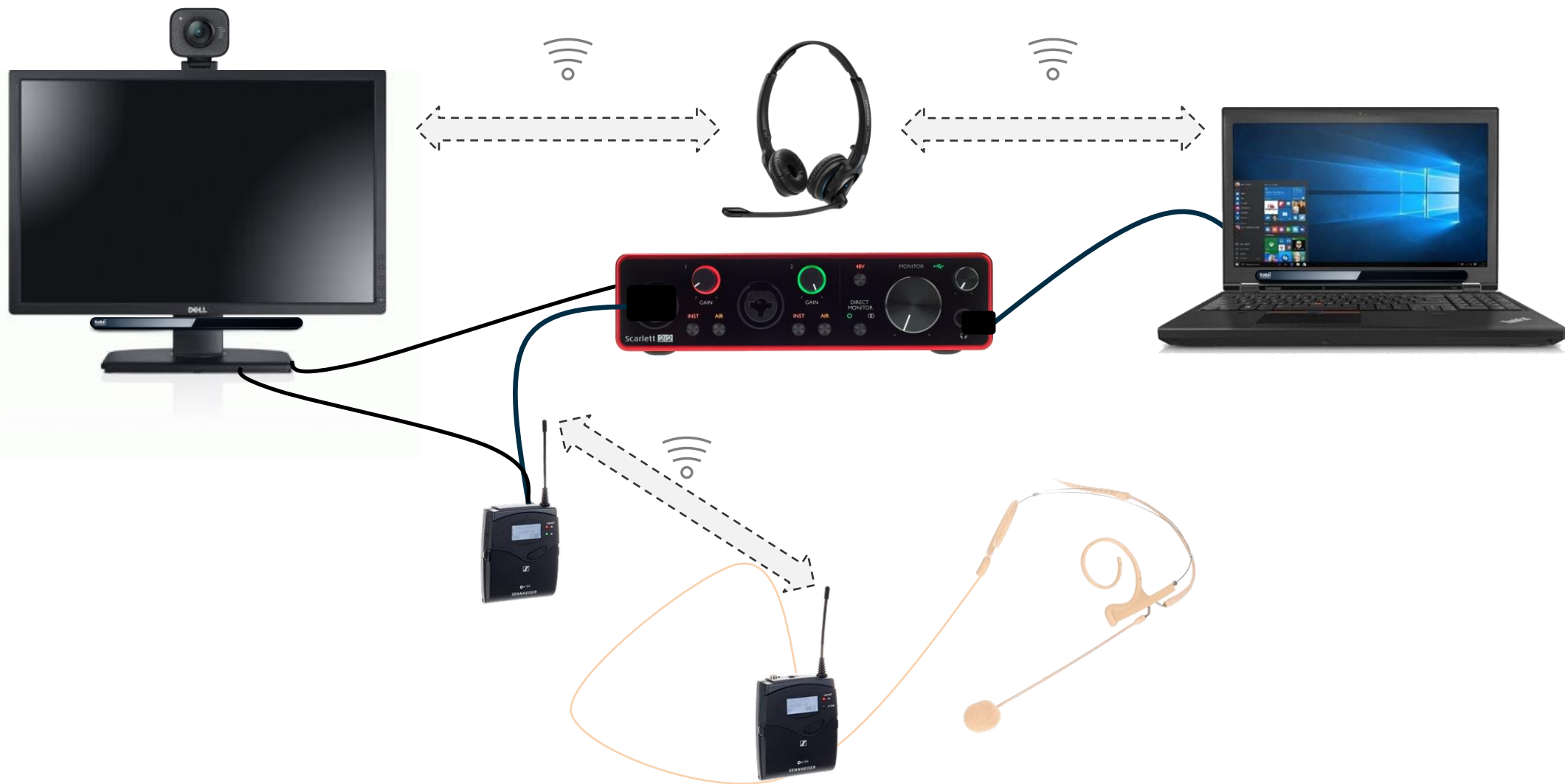
Multi-Computer Setup

Pitfalls & Challenges

Recognition Accuracy

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

Workflow & Anger Management

- **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!

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!!!

The **Hoff** approves!

Helpful Resources & Outlook

Recommended Tooling & Documentation

- **Talon** (Free of Charge): talonvoice.com

  - Voice coding for Win / Linux / Mac!

  - Wiki: talon.wiki

  - Speech Recognition: github.com/facebookresearch/wav2letter

  - Starter Grammar (English): github.com/knausj85/knausj_talon

- Paid Upgrades:

  - Dragon Speech Recognition: nuance.com/dragon/

  - Web Speech API (compatible with Talon through Chrome or Firefox)

  - Premium Support (Patreon): patreon.com/join/lunixbochs

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
- Scripting:
  - NatLink: sourceforge.net/p/natlink/
  - Dragonfly: github.com/dictation-toolbox/dragonfly
  - Caster: github.com/dictation-toolbox/Caster
  - Vocola (Voice Command Language): vocola.net

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/
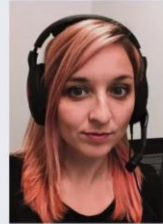- Max Gravenstein (2018): medium.com/hubabl/handsfree-fe70980f36b/

Recommended Talks

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

Boudewijn Aasman. Coding by Voice with Dragonfly, PyGotham (2018)
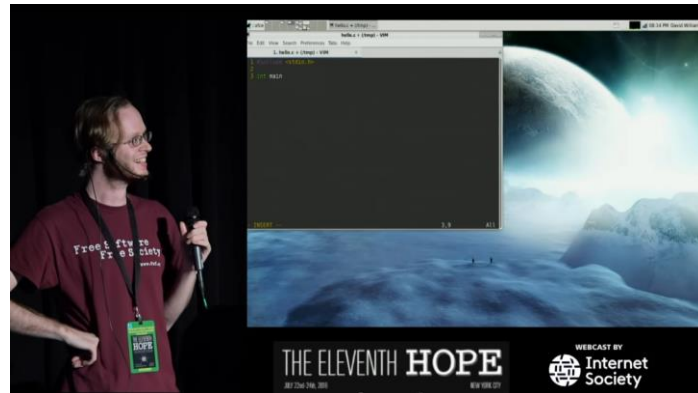
David Williams-King. Coding by Voice with Open Source Speech Recognition, The Eleventh Hope (2016)
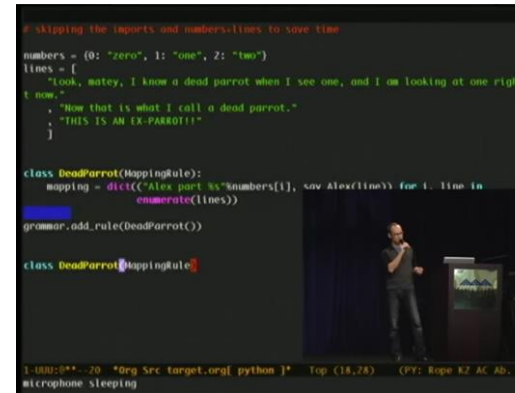
Tavis Rudd. Using Python to Code by Voice, PyCon US (2013)

And There is Much **More** to It ...

- Advanced best practices, e.g.

  o Multi-language support, dynamic scripting

  o Hardware recommendations, interface / HUD, ...

  o Using applications, e.g. handsfree browsing

- More input channels, e.g.:

  o Custom noise models

  o OCR & images recognition

  o Control via facial expression!

- Privacy considerations

More live demos!!!

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 <u>for you</u>

  (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.)