

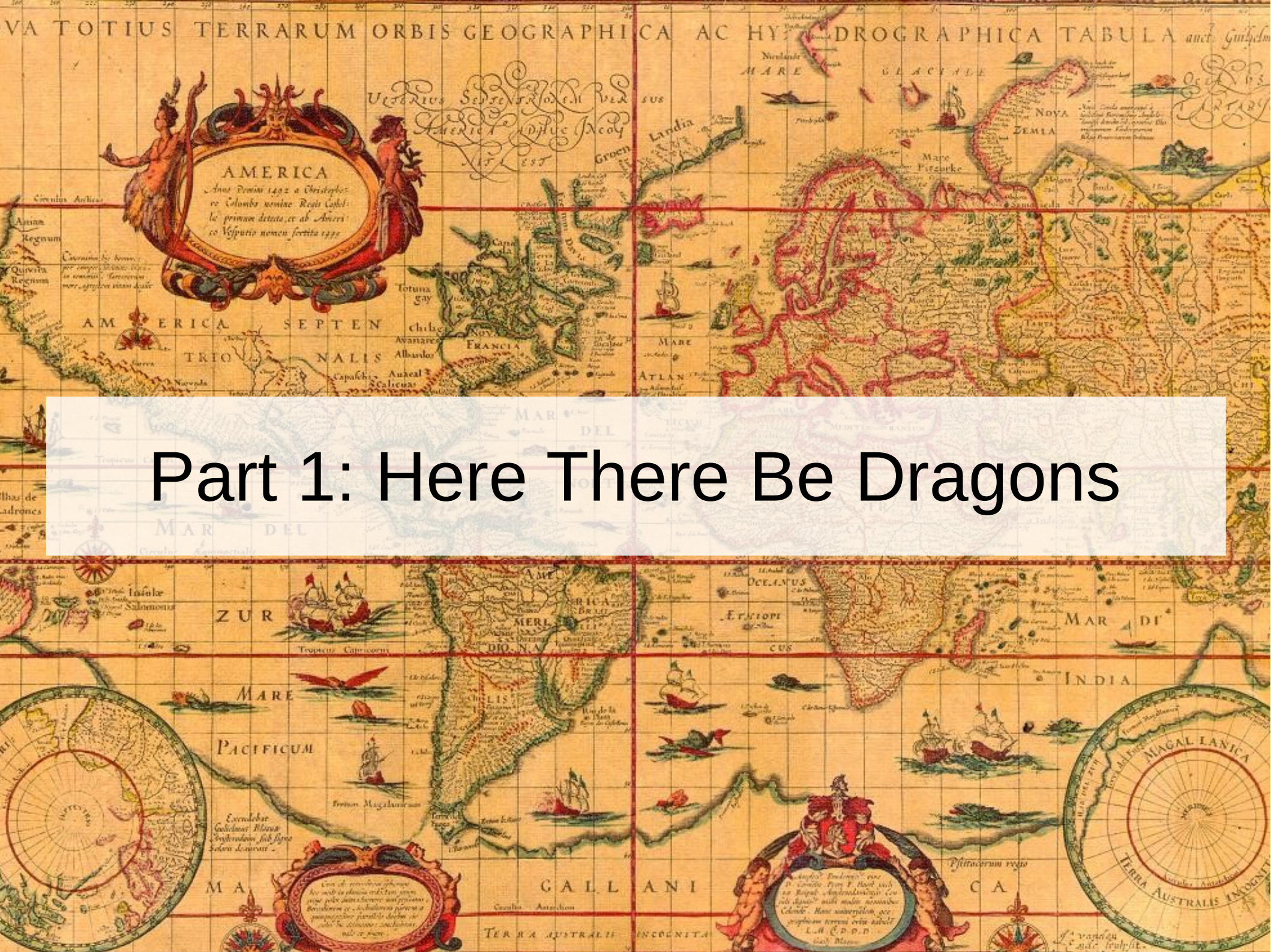
Coding by Voice with Open Source Speech Recognition

David Williams-King
Ph.D. student at Columbia University
dwk@voxhub.io



Too-Much-Typing Disease

- Muscle strength & endurance → 0
 - Could not type, use a pencil, open doors, etc
 - Could not walk, sit for more than ten minutes
 - Very easy (and painful) to accidentally do too much
 - An unknown virus appears to be the culprit
- Repetitive strain injury (RSI)
 - wrists/ulnar nerve (carpal tunnel)
 - medial nerve (tennis elbow)
 - shoulders, neck, fingers, ...



Part 1: Here There Be Dragons

Dragon NaturallySpeaking

- Command-and-control system for Windows
 - Open windows, click buttons, etc
 - Dictate text, select words by voice, corrections
 - Also available as Dragon Dictate for Mac
- Commercial software
 - Normally \$100-\$200
- “MS Word rock star”
 - messes with formatting too much for programming

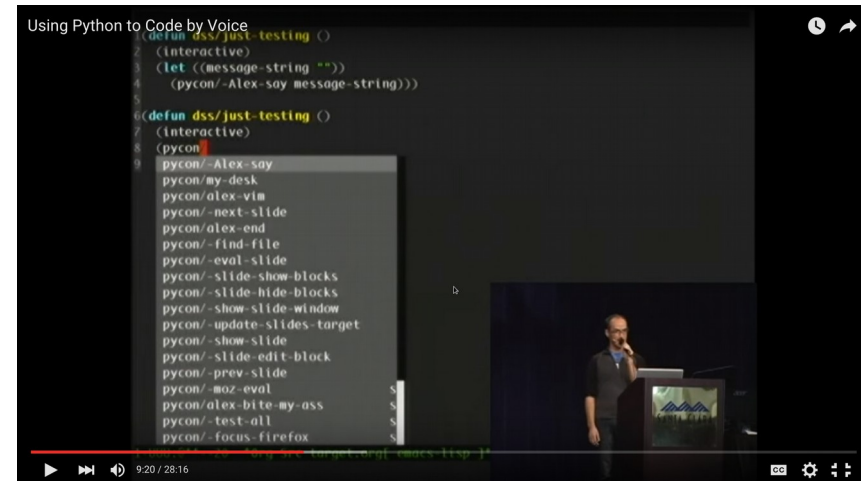
How to Train Hack Your Dragon



“You want me to do *what?*”

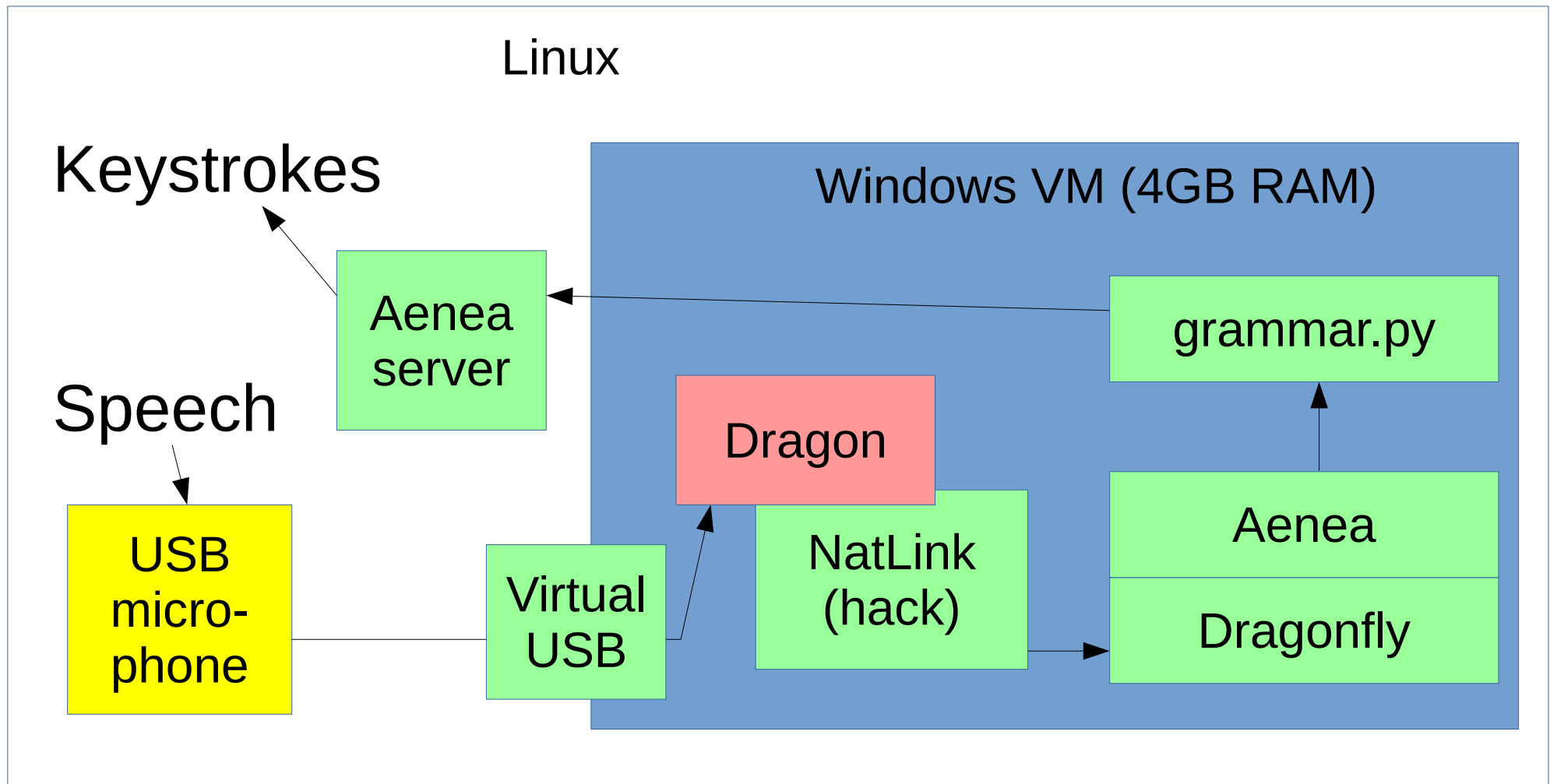
Evolution of Voice Coding

- 1991 **NatLink** is created by Joel Gould of Dragon Systems to allow Python macros
- 2008 **Dragonfly** is written by Christo Butcher, providing a framework for Python grammars
- 2013 Tavis Rudd gives talk at PyCon about custom voice coding on Linux
- 2014 **Aenea** by Alex Roper recreates full Linux voice coding support

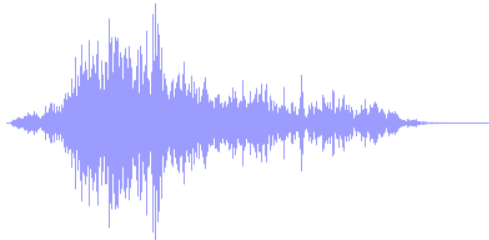


Full Aenea Stack

- Needs Windows VM (VMware/KVM/VirtualBox)



But how can this be used for coding?



?



```
.global loader_main
.global initial_stack
.global entry_point

.global _start

.section .text

# We only need to preserve %rdx and %rsp, according to
# glibc's sysdeps/x86_64/elf/start.S, which cites the SVR4/1386
_start:
    push    %rdx
    mov     %rsp, initial_stack    # save top of stack

    mov     0x8(%rsp), %rdi        # set argc
    lea     0x10(%rsp), %rsi       # set argv

    mov     %rsp, %rbp
    call    loader_main

    mov     initial_stack, %rsp    # restore %rsp
    mov     entry_point, %rbx      # restore %rdx
    pop     %rdx                  # restore %rdx

    jmp     *%rbx                 # jump to entry point

.section .note.GNU-stack, "", @progbits
```

Basic Voice Grammar Design

- NATO-esque alphabet
 - arch, bravo, char, delta, echo, fox, golf, hotel, ...
- Symbols and characters
 - 0-9, space, “slap” for enter, “act” for escape, ...
 - () [] < > { } are “l”/”r” + “en”/”ack”/”angle”/”ace”
- English words
 - sentence hello there → Hello there
 - score merge sort → merge_sort
- Chaining: say sequences without pausing

Aenea Demo



Aenea Demo

- Aenea mailing list:
<https://groups.google.com/forum/#!forum/dragonflyspeech>

Microphone Hardware

- Good-quality USB microphones: Decent



Samson Meteor



Blue Snowball



Blue Yeti

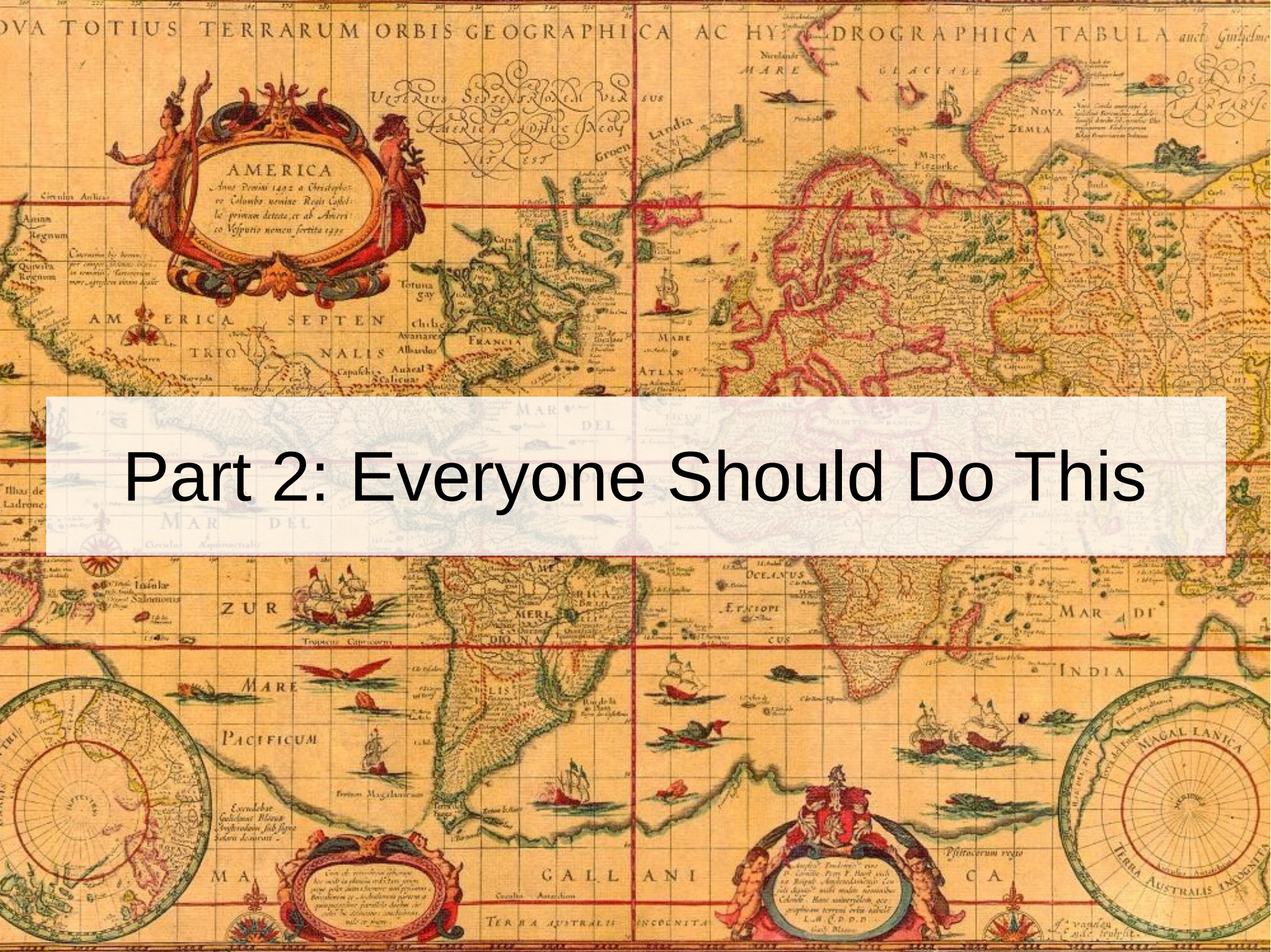
- Professional XLR mics: Amazing



Shure WH20XLR



Audio-Technica 8HEX



Part 2: Everyone Should Do This

Aenea – Available to All?

- Need Windows and Dragon licenses
 - cannot distribute working VM images
 - some people never get Aenea working
- Grammar incompatibility & fragmentation
 - just Python scripts with little enforced form
 - hard to combine grammars from different people
- Significant computing power requirements
- Can we lower the barrier to entry?

Dragons Play Hard to Get

- Buy Dragon instances and run in the cloud
 - Licensing issues (Nuance director of sales)
 - Stability/scripting issues
 - Remote microphone issues
 - USB virt. is high bandwidth, latency sensitive
 - Audio streaming with rtp/voip protocol? Dragon does not open most virtualized microphones
 - Microsoft RDP protocol... any way to use only audio?
- Could provide for about \$5/month by getting Dragon on Ebay and spinning up VMs...
- but there must be a better way.

Other Kinds of Speech Recognition

- Cloud-based speech recognition for smartphones
 - Siri, Google, Nuance... hard to get an API
 - Google Cloud Speech API now has a limited preview
- Dedicated APIs like Hound, Nuance Mobile
 - designed for low volume, quite expensive
- Local smartphone recognition
 - coming soon? papers from Google Research
- Others:
 - Amazon Echo
 - Kickstarter Arduino shields (100 word dictionary)

Time to reinvent the wheel.



How Speech Recognition Works

- Many open source speech recognition toolkits
 - HMM Toolkit (HTK), CMUSphinx, Kaldi
 - Most research happening on Kaldi, so we use it
- Steps:
 - Signal processing: finding features in sound signals
 - Acoustic modeling: recognizing phonemes like /ā/
 - Language modeling: valid sequences of words



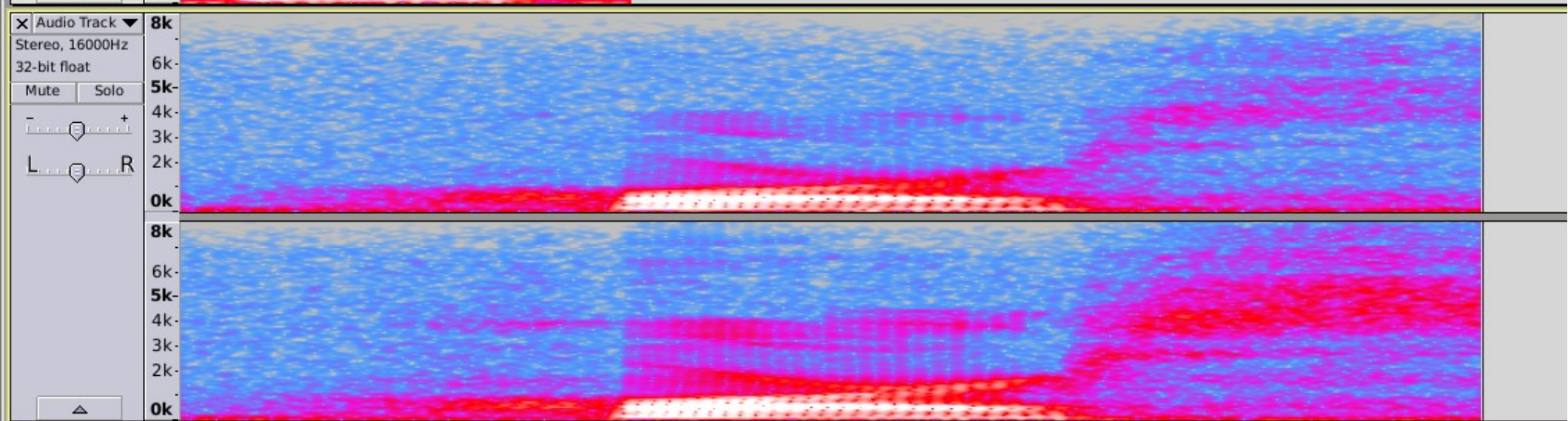
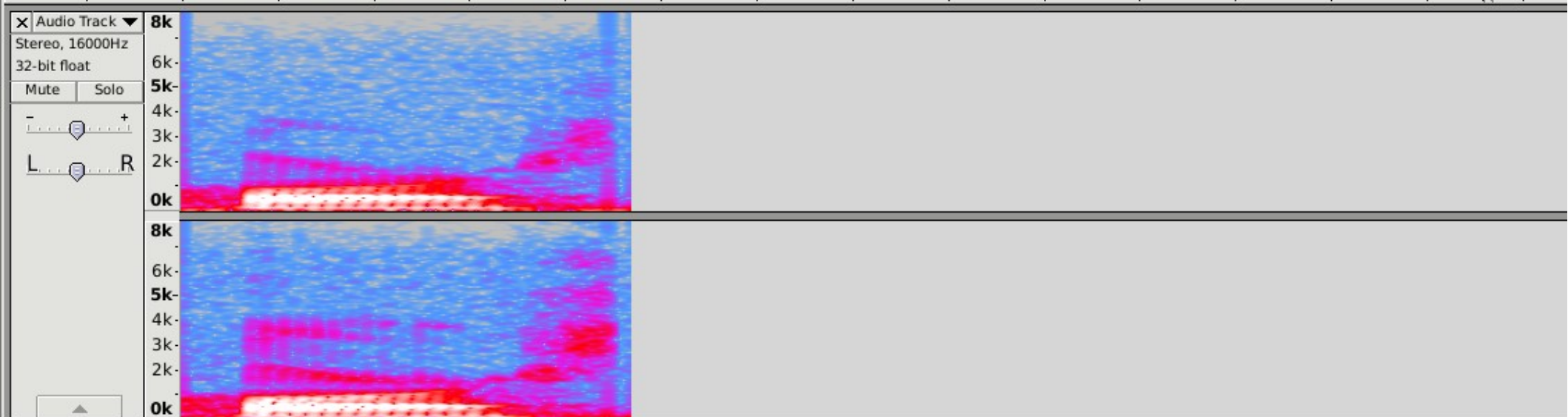
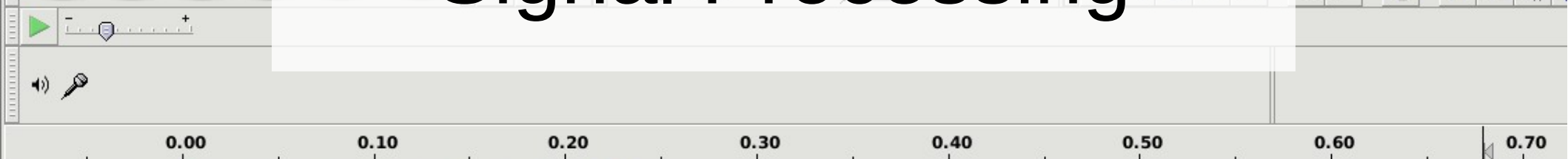
Audacity

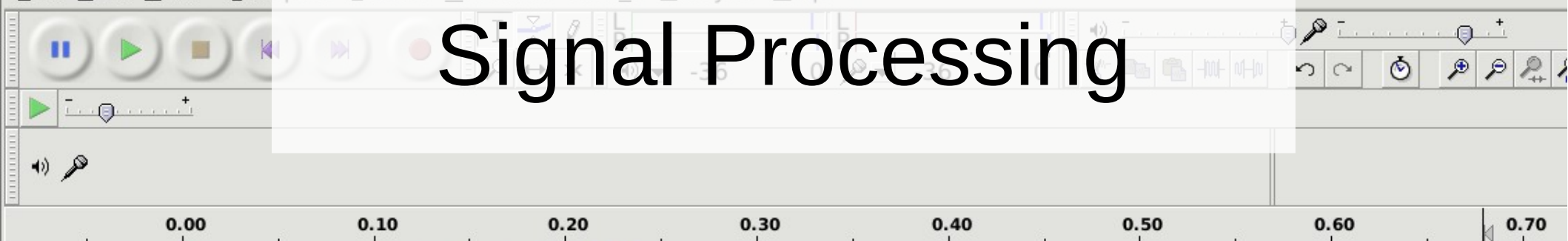


File Edit View Transport Tracks Generate Effect Analyze Help



Signal Processing





Signal Processing

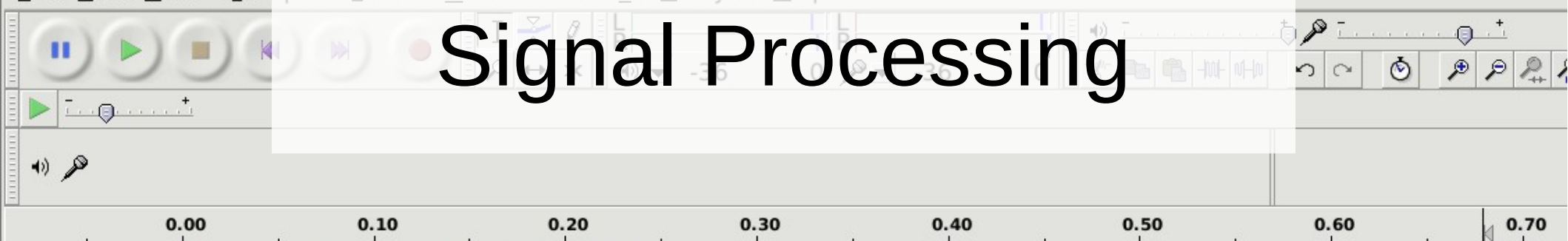
“horse”

- Speech: 16k, phone: 8k
- vowels have formants
- 's' is a fricative sound, above 4k



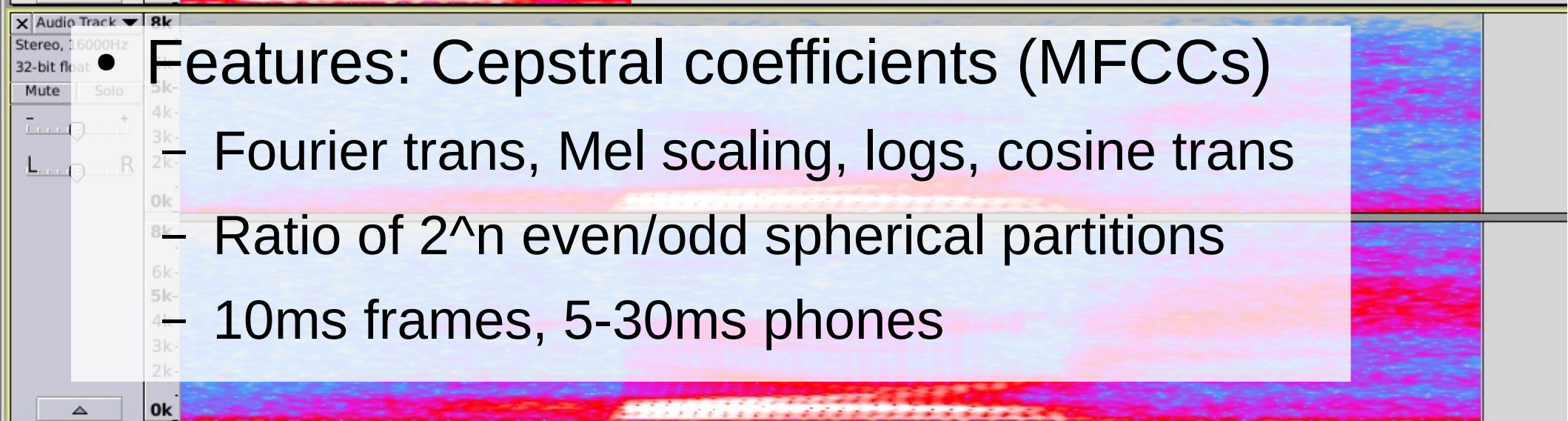
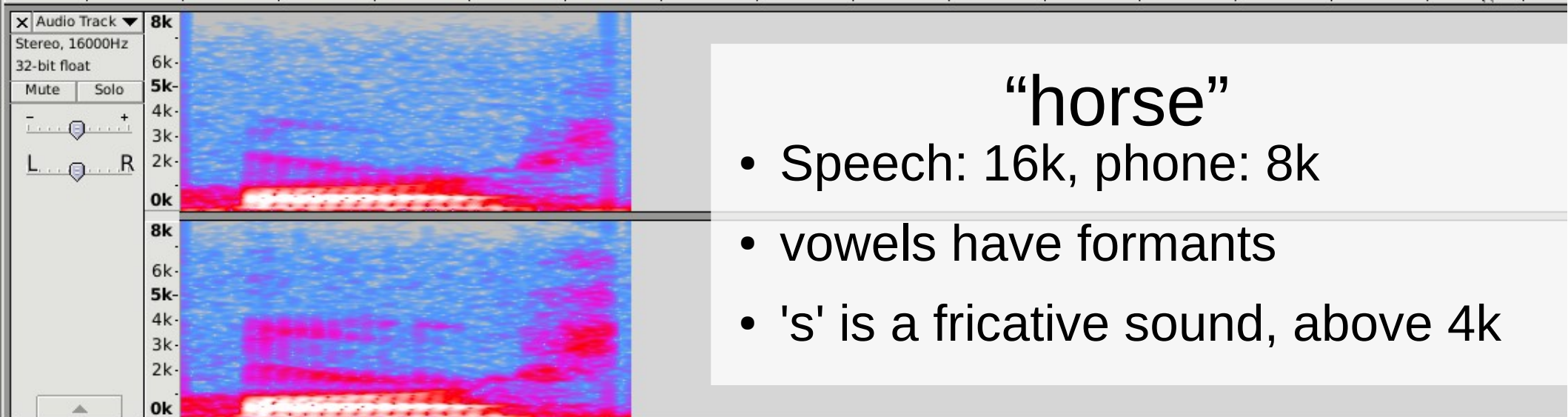


Signal Processing



“horse”

- Speech: 16k, phone: 8k
- vowels have formants
- 's' is a fricative sound, above 4k



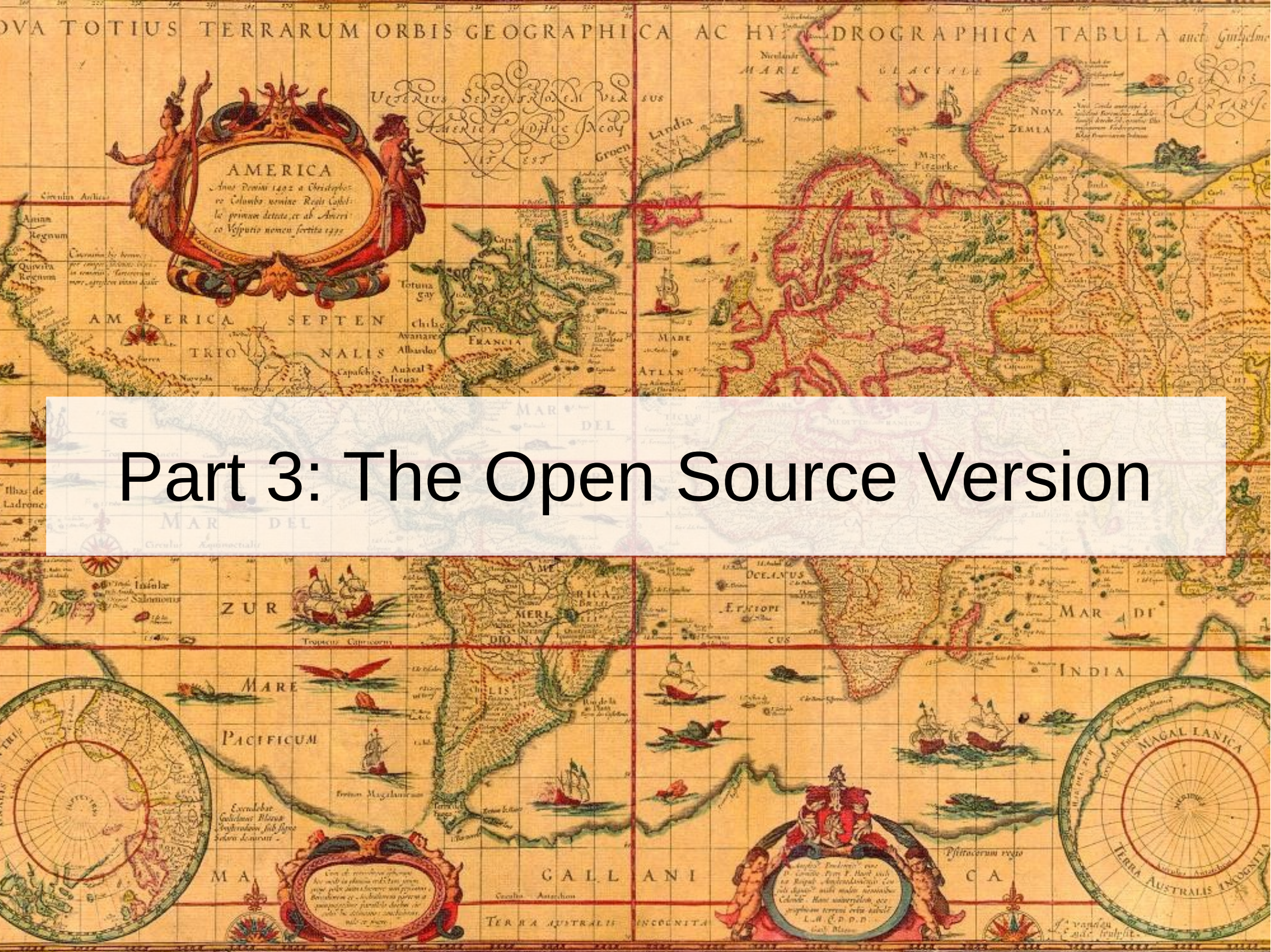
- Features: Cepstral coefficients (MFCCs)
 - Fourier trans, Mel scaling, logs, cosine trans
 - Ratio of 2^n even/odd spherical partitions
 - 10ms frames, 5-30ms phones

Acoustic Modeling

- Train with hundreds of hours of speech
- Learn individual phonemes
 - Model with Gaussian Mixture Models (GMMs) or deep neural networks (DNNs)
- Model speech with Hidden Markov Models
- Extremely computationally intensive
 - Even a 24-core server with 48GB RAM takes days
 - Pretrained models available (tedlium, librispeech)

Language Modeling

- N-gram language model (e.g. 3-gram)
 - Google 5-gram, Dragon BestMatch IV, BestMatch V
 - Hidden Markov Model searched in greedy fashion with the Viterbi algorithm
- To change the commands that may be spoken, we must model a new language



Part 3: The Open Source Version

New Speech System: Silvius

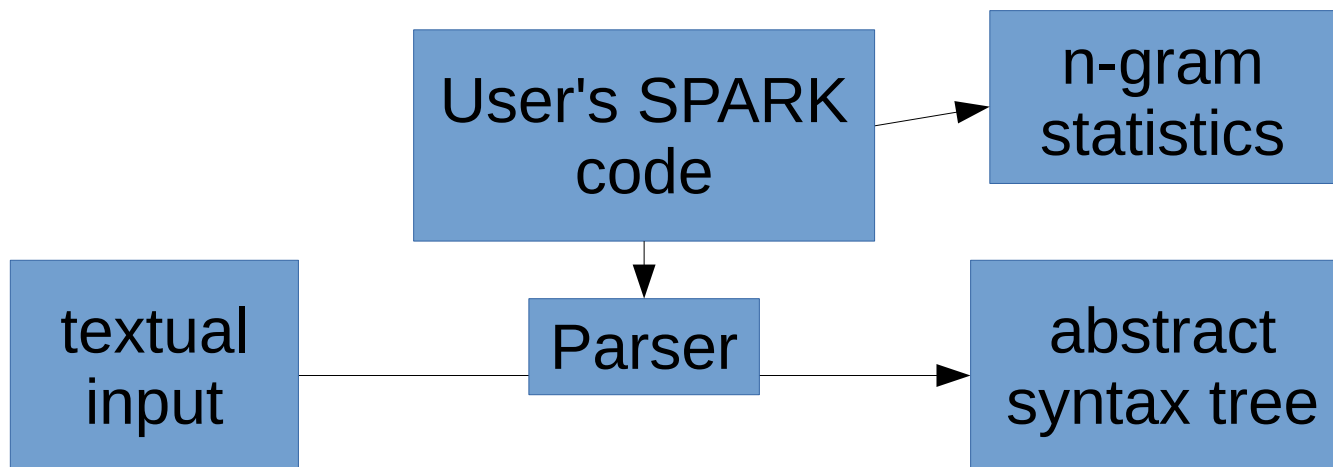
- Requirements:
 - Open source code, freely available speech models
 - Can run locally or in the cloud
 - User-provided custom speech grammar
- Goal: speech recognition with minimum hassle
 - low computing resources required
 - simple installation requirements
 - maybe even no software installation at all?
 - a true voice keyboard

How To Use a Custom Grammar

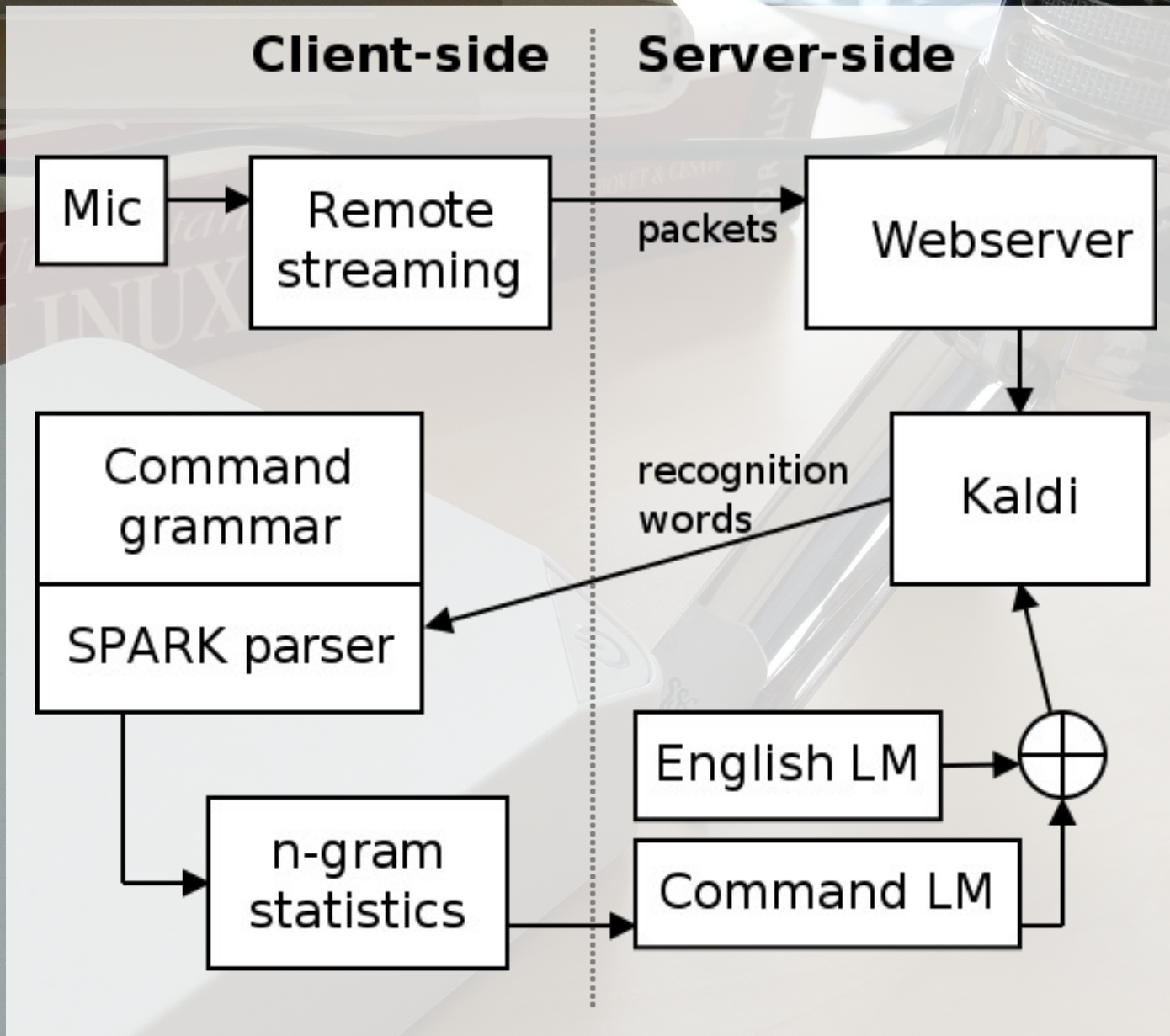
- Rule-based language models (Thrax, julius)
 - not good at handling mistakes
- Merge two language models together?
 - Mandarin & English at Baidu (10k hours of speech)
 - Retrain with command words interspersed?
 - Linear combination: use $\alpha * L1 + (1-\alpha) * L2$
 - I use 80% English, 20% command LM
- The grammar must support iterating over it to extract the valid sequences for a LM

Silvius Grammars

- Written in Python with SPARK parsing toolkit
 - Create parser tree with meta-Python objects
 - Can walk the parser tree to generate n-gram LM
 - Parser converts text to an abstract syntax tree
 - Walk the AST and execute commands
- Like a compiler-compiler with introspection



The Silvius Architecture



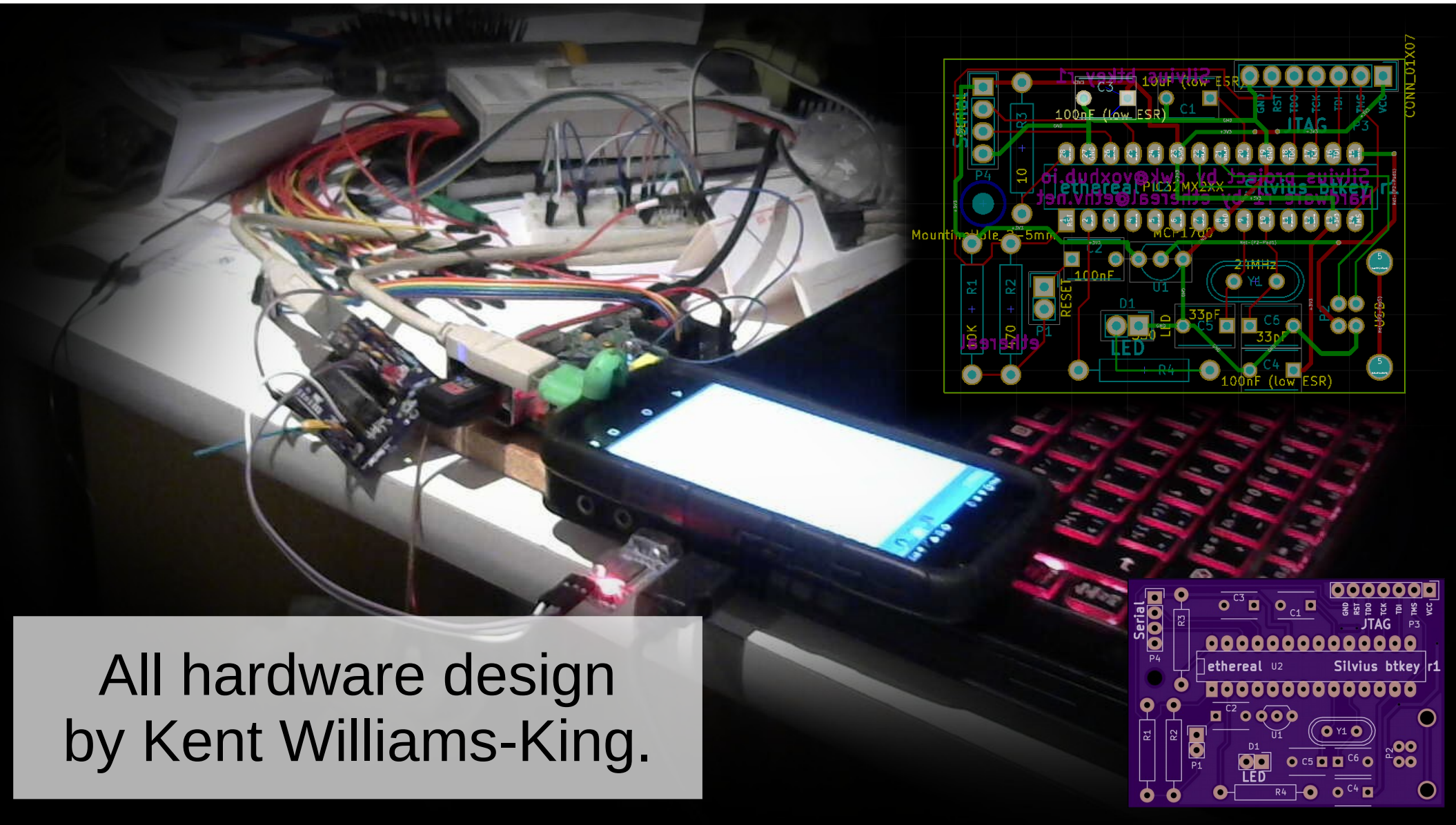
Huge thanks to Tanel Alumäe for the gstreamer server!

Use Cases

- Run full recognition locally (2.4GB RAM)
- Use cloud servers for recognition
 - can provide service for about \$4/month
- Run recognition on embedded systems
 - can run on a “voice box” or smartphone
 - smartphone microphones are getting quite good
- Use recognition results on any computer, without installing any software
 - bluetooth → fake USB keyboard

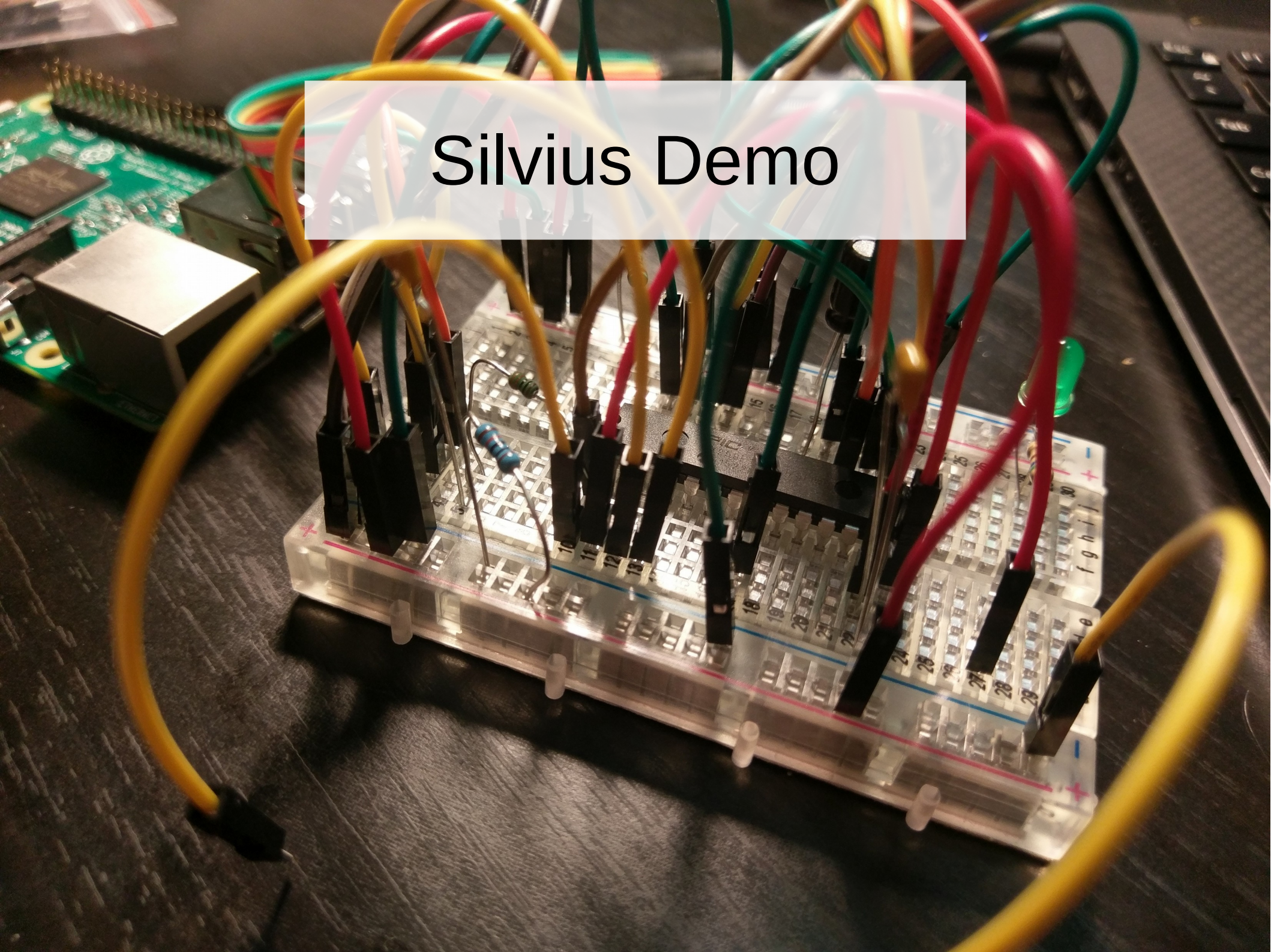
Bluetooth → USB fake keyboard

- Allows a phone to generate laptop keystrokes



All hardware design
by Kent Williams-King.

Silvius Demo





Silvius: Online Programming by Voice

Silvius is an open source system for writing code by voice. Speak characters and words and they are typed for you automatically. You can try it on our server, or set it up yourself.

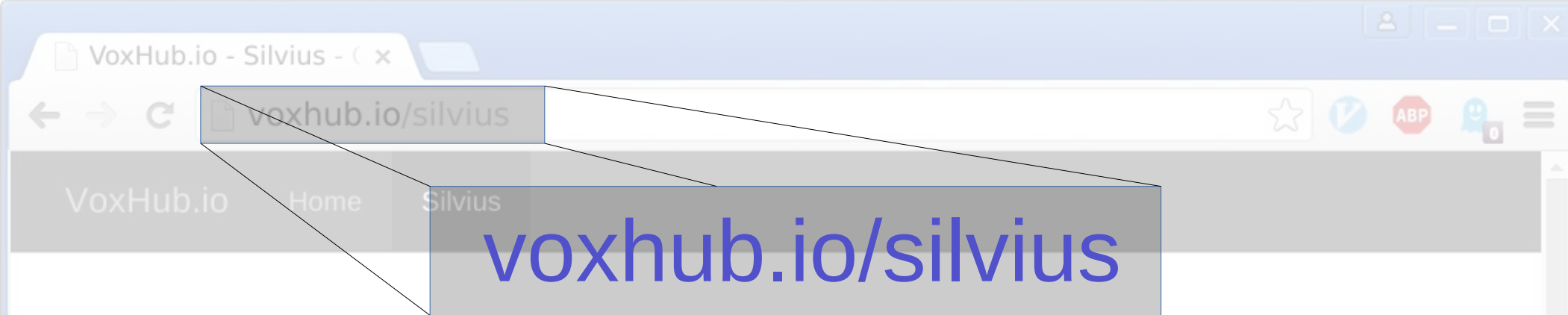
Setup Instructions

1. Install dependencies (only works on Linux currently):

```
$ sudo apt-get install git python python-ws4py python-pyaudio libjansson4  
xdotool
```

2. Download client: ([GitHub page](#) )

```
$ git clone https://github.com/duke/silvius.git
```



Silvius: Online Programming by Voice

Silvius is an open source system for writing code by voice. Speak characters and words and they are typed for you automatically. You can try it on our server, or set it up yourself.

- (2x) Online Silvius servers for public use
- Eventually: grammar database
- Eventually: hardware configuration database

Summary

When you can't type,



harness speech recognition



and code by voice.



Summary

When you can't type,



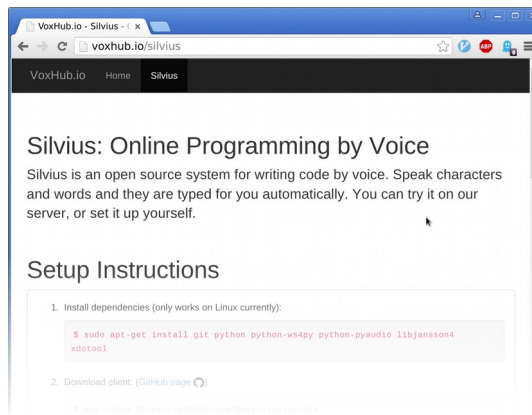
harness speech recognition



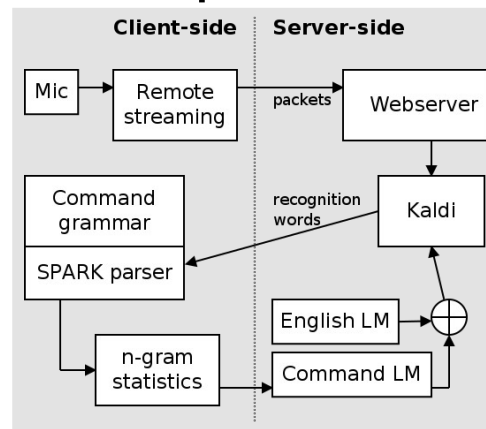
and code by voice.



If you find this interesting,
Silvius makes it easy



to experiment



and build new ways of
interacting with computers.



Acknowledgements

- Silviu would not have been possible without:
 - Tanel Alumäe's kaldi-gstreamer-server!
 - Professor Hoomayoon Beigi's guidance
 - The Kaldi speech recognition toolkit. Thanks Dan :)
- Other notable mentions:
 - John Aycok's SPARK parser toolkit
 - Tavis Rudd and Alex Roper and Susan Cragin...
 - And all the many people who have maintained NatLink, Dragonfly, and Aenea over the years

Questions.



For more information

- These slides: <http://voxhub.io/static/hope.pdf>
- Silvius: <http://voxhub.io/silvius>
 - Open sourced in 3 repositories on Github
- Tavis Rudd's talk:
<https://www.youtube.com/watch?v=8SkdfdXWYal>
- Aenea mailing list:
<https://groups.google.com/forum/#!forum/dragonflyspeech>
- Kaldi speech toolkit: <http://kaldi-asr.org/>

What if I have RSI?

- See a neurologist, and physiotherapists
- Increase breaks, reduce use, ergonomics
 - stop playing computer games :(
 - workrave forces you to stop typing on a schedule
 - make sure desk height & chair setup are optimal
 - get a good backpack to carry stuff, try wrist braces
- Get better hardware
 - Goldtouch (or Kinesis) keyboards are amazing
 - Use a trackball, or Wacom drawing tablet for extensive mousing
- It gets better. Eventually.

Computing Hardware

- Aenea
 - Windows VM, i7-3517U/i5-6200U, 4GB virtual RAM
- Silvius
 - Low-end x86 CPU needed at the moment
 - i7-4700HQ locked at 1.2GHz
 - i3-5005U at 2.0GHz
 - RAM: 2.4GB