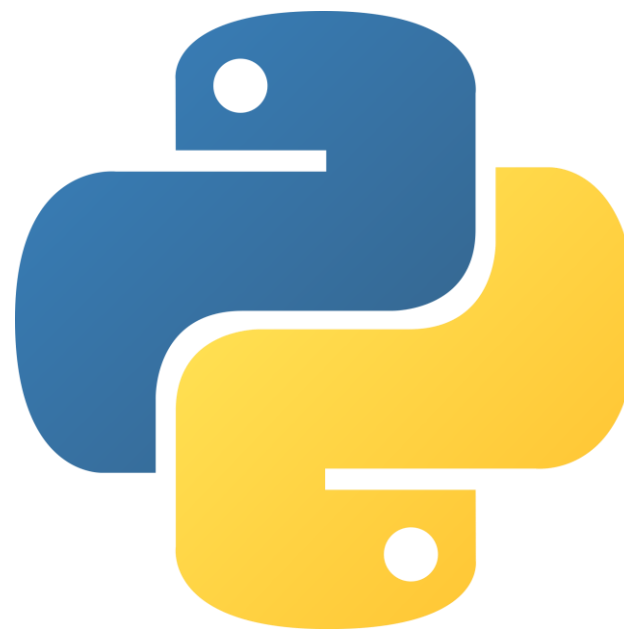


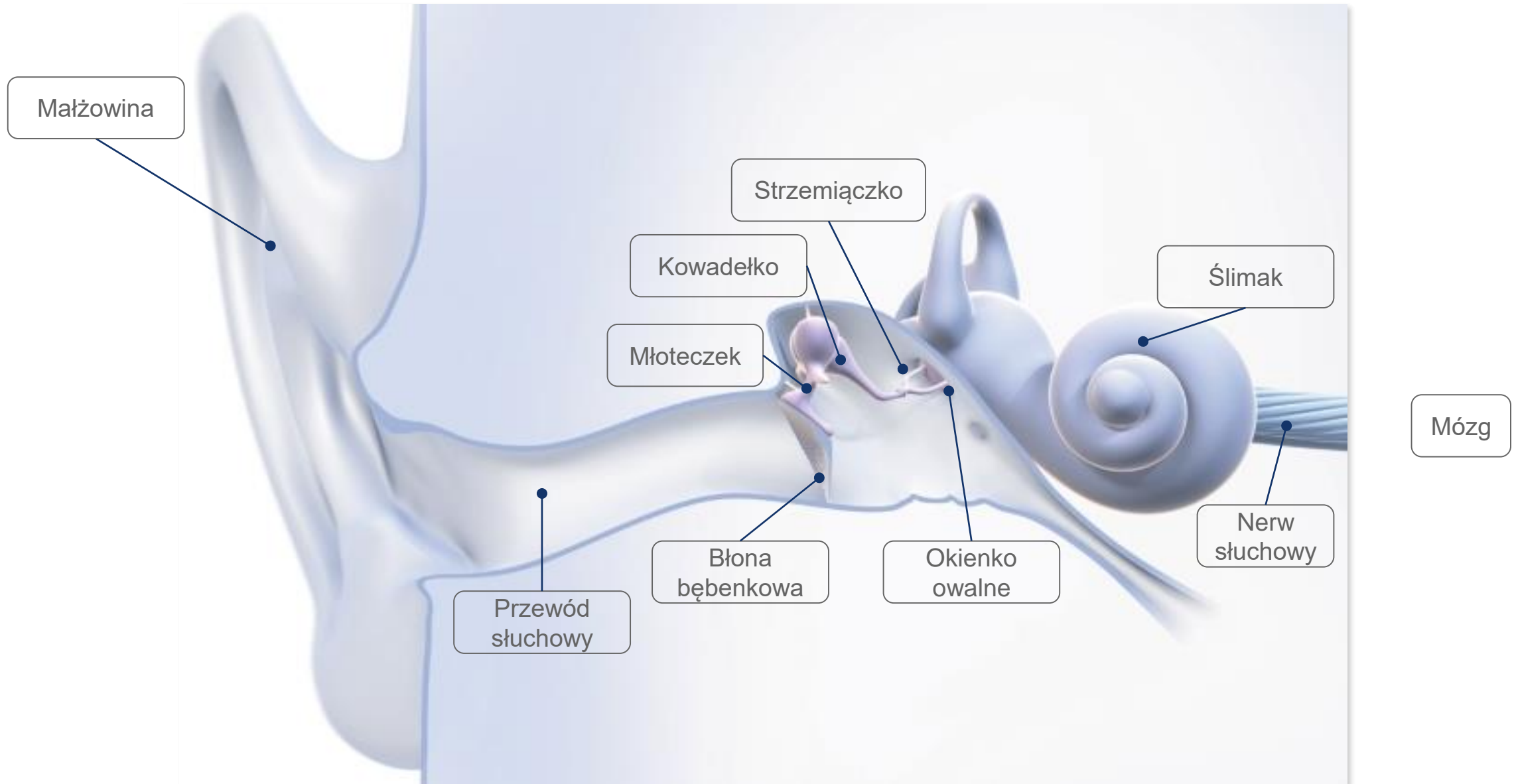
Python w świecie aparatów słuchowych



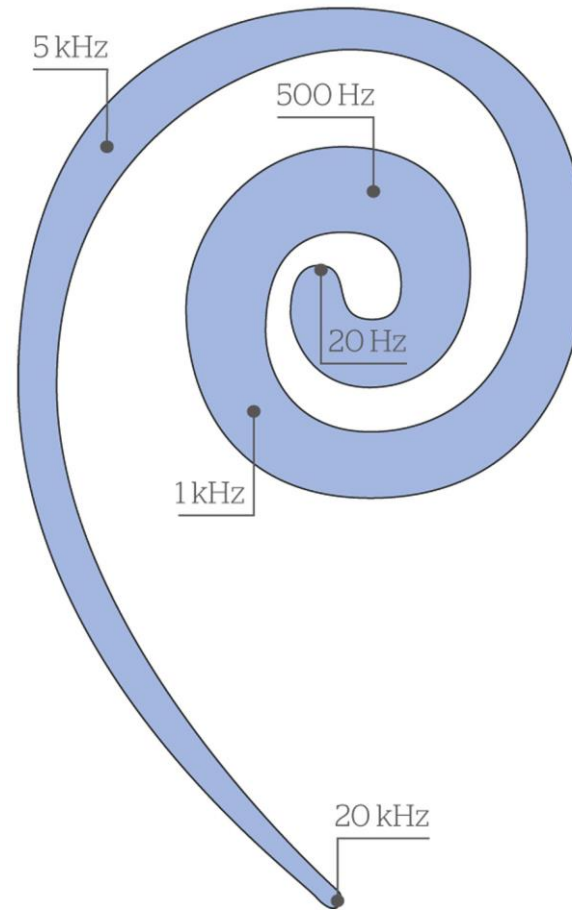
Marta Potocka, Michał Partyka,
Bartłomiej Krztuk
PyWaw – 17.09.2018

- 1 Jak działa słuch?
- 2 Aparaty słuchowe i implanty ślimakowe
- 3 Co znajduje się wewnątrz aparatu
- 4 Co w tym wszystkim robi Python?

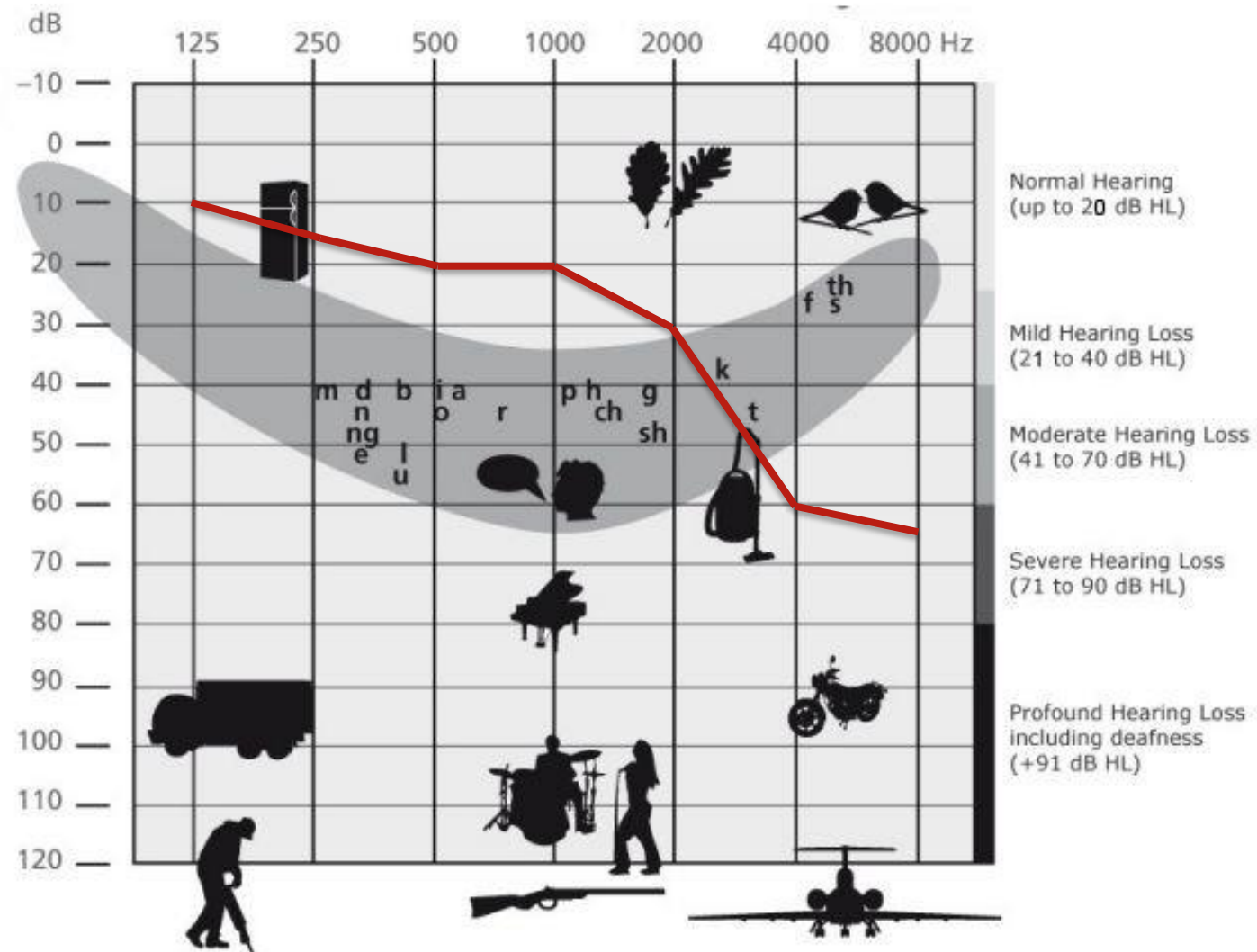
Jak działa słuch?



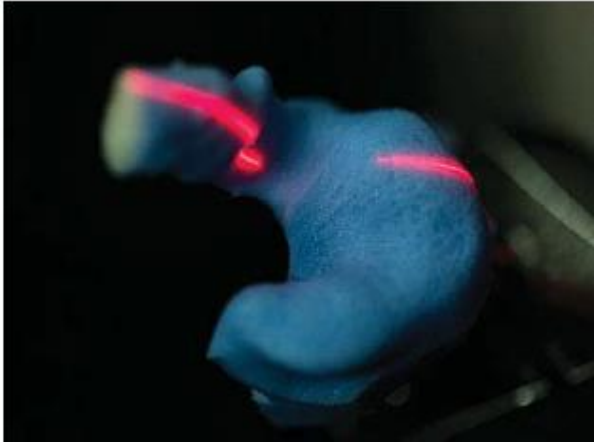
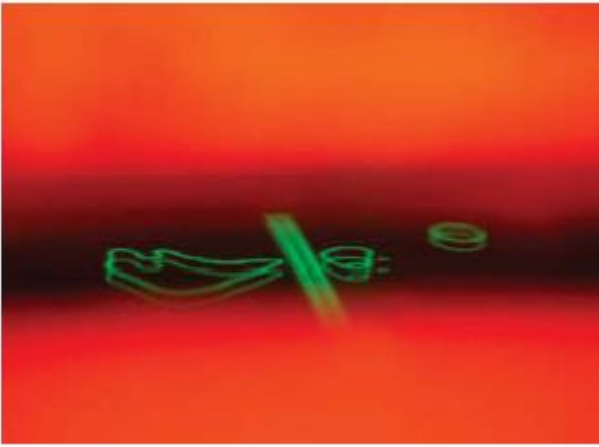
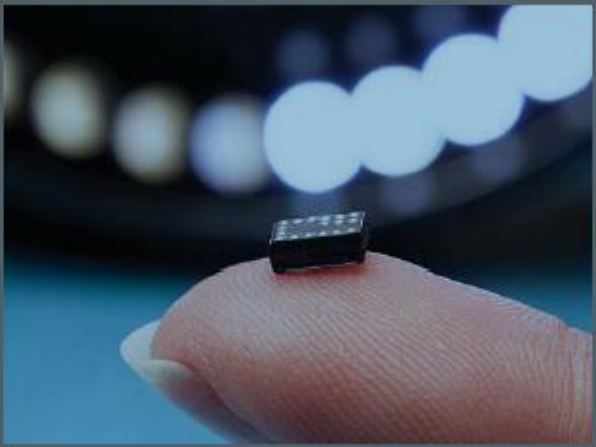
Rozkład częstotliwości w ślimaku



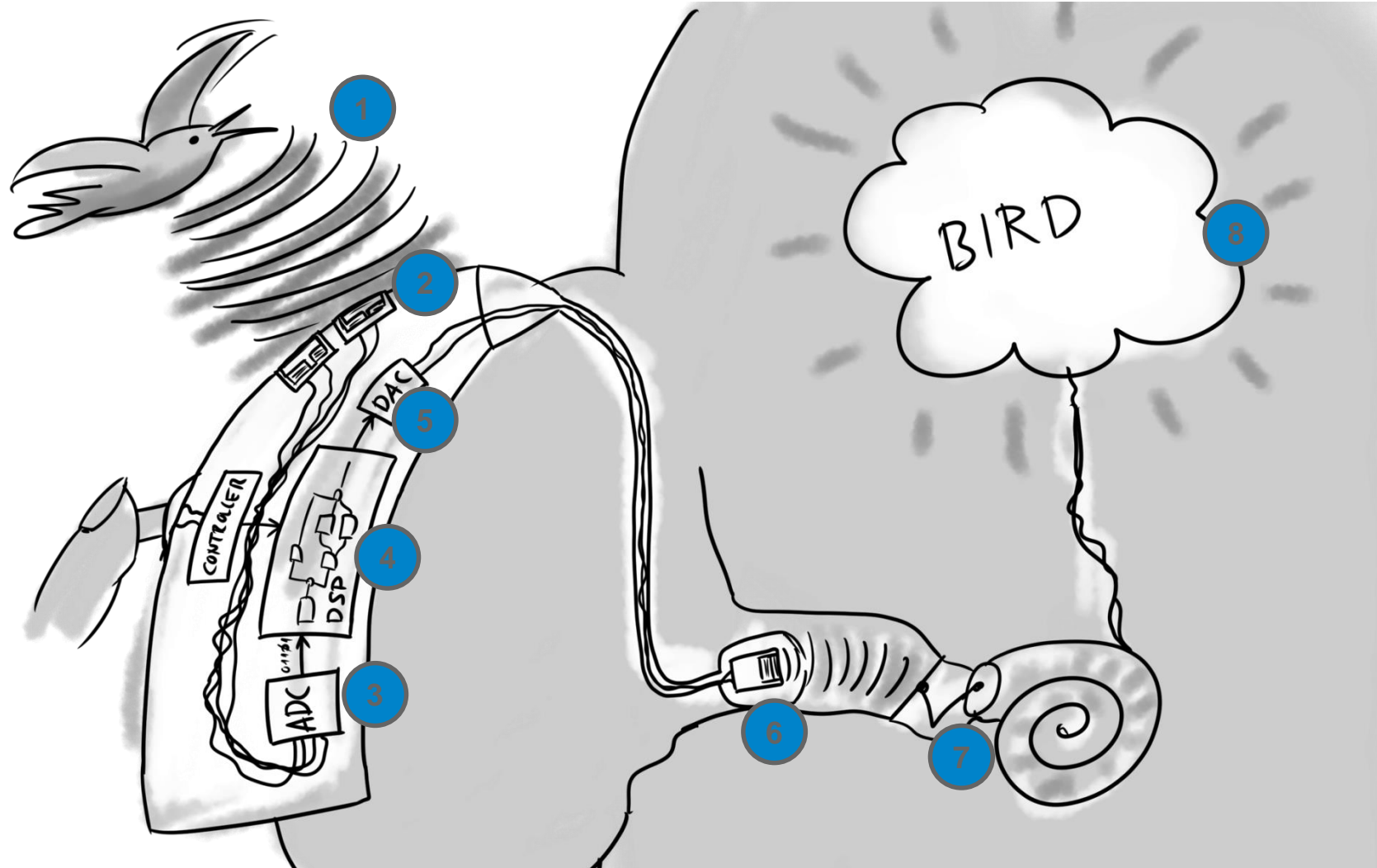
Audiogram



Aparaty słuchowe i implanty ślimakowe

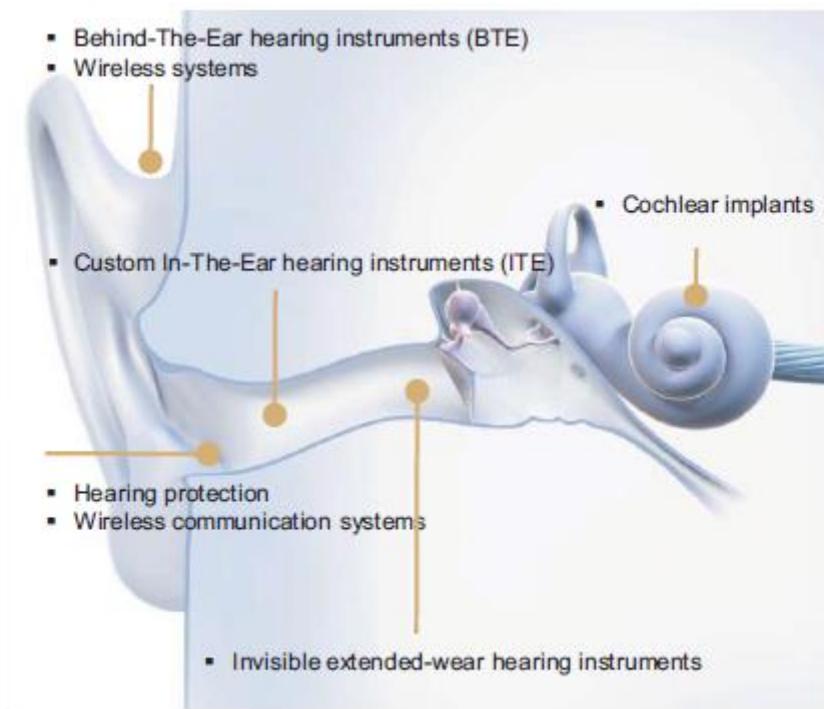


Jak działa aparat słuchowy – przepływ sygnału



Jak działa implant

HI Hearing Instruments



CI Cochlear Implants

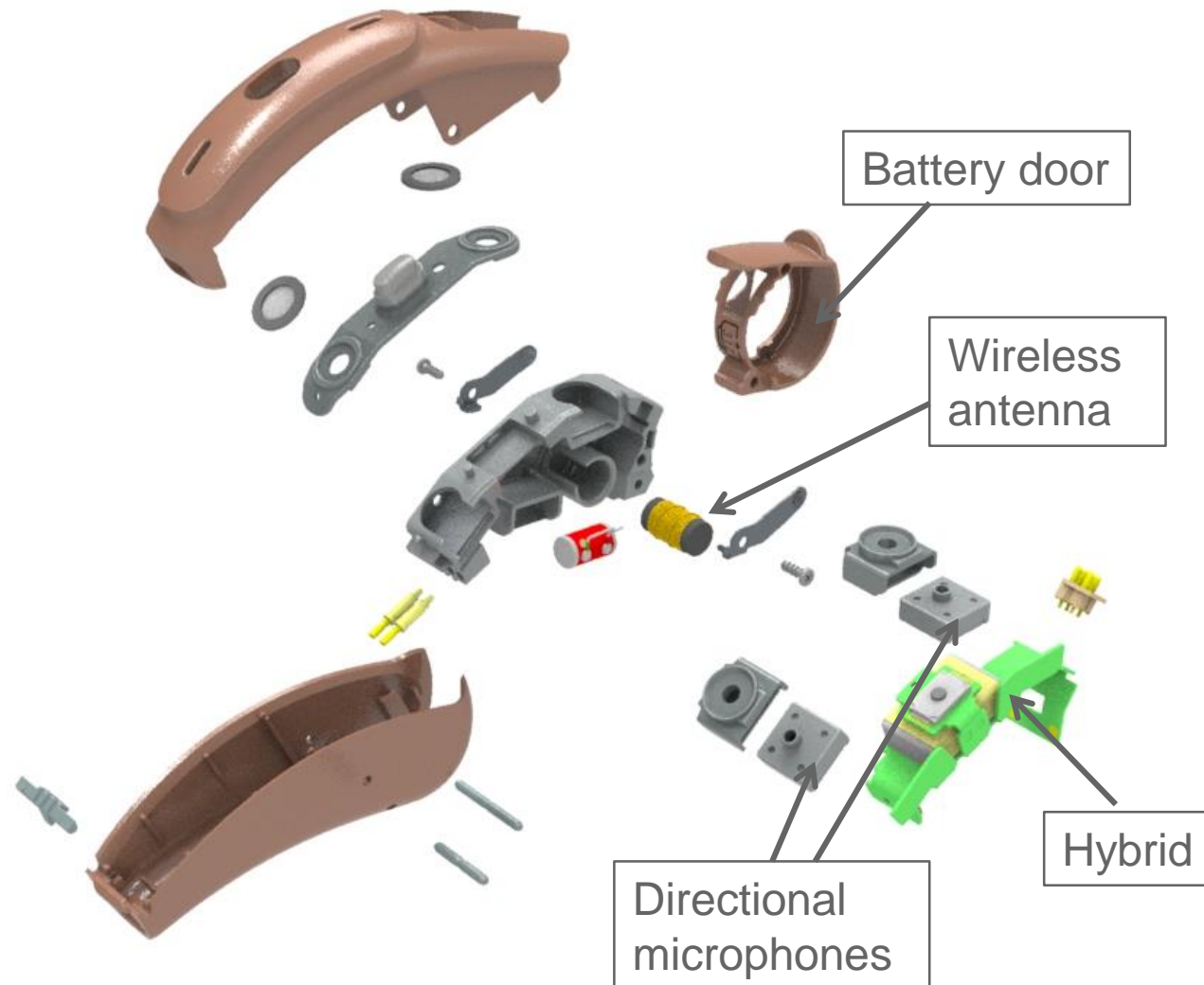


Cochlear Implants

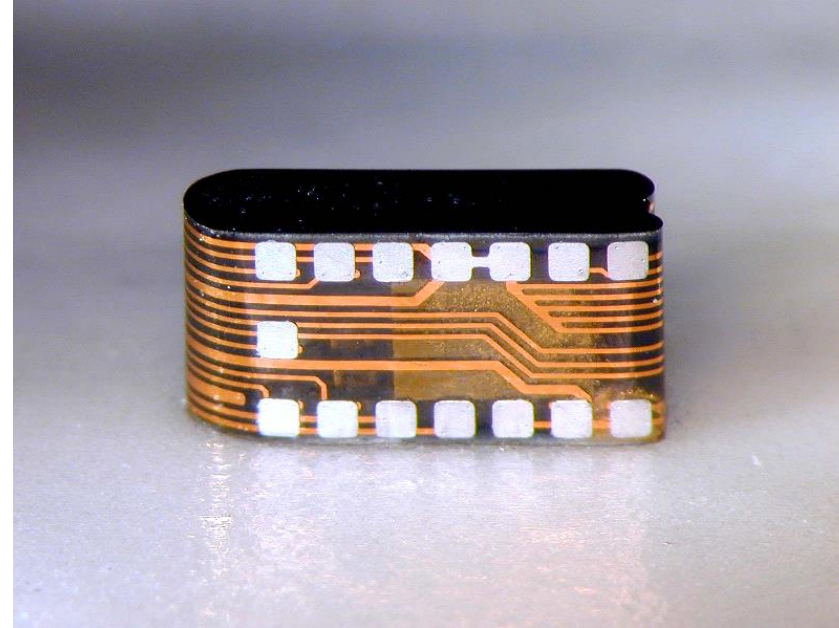
This video contains a simulation of cochlear implants with a various number of channels on speech and music.

Audio from <http://www.sens.com/helps/>

Co znajduje się wewnątrz aparatu?

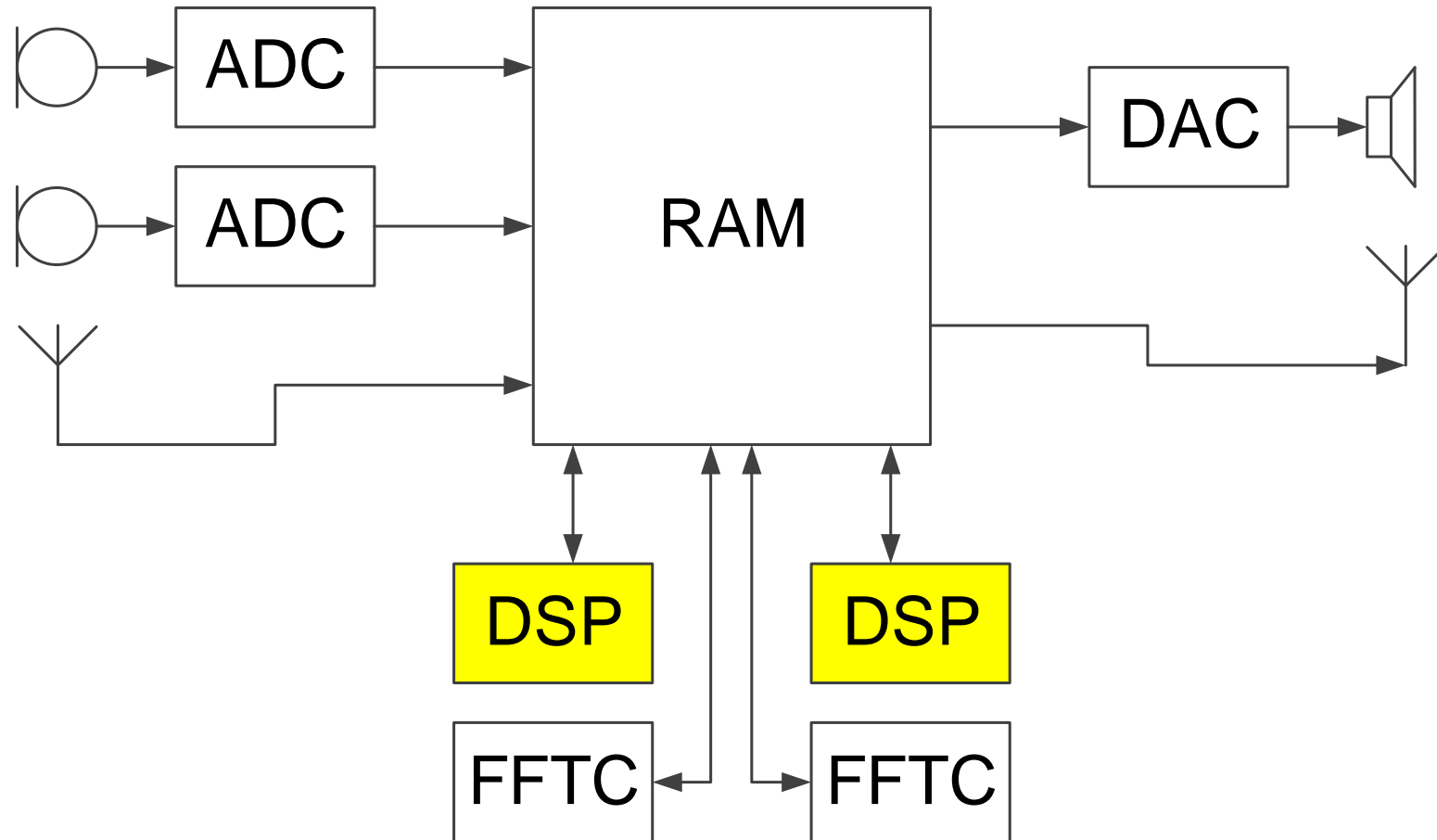


Hybrid – serce aparatu słuchowego



- DSP, Controller, Wireless Chip, Bluetooth, Power Management, Memory
- ultra niski pobór energii
- pamięć: ~7.5 Mb ~1MB
- procesory: 5.64 MHz
- 1.46mm x 5.51mm x 3.46mm

Przetwarzanie dźwięku



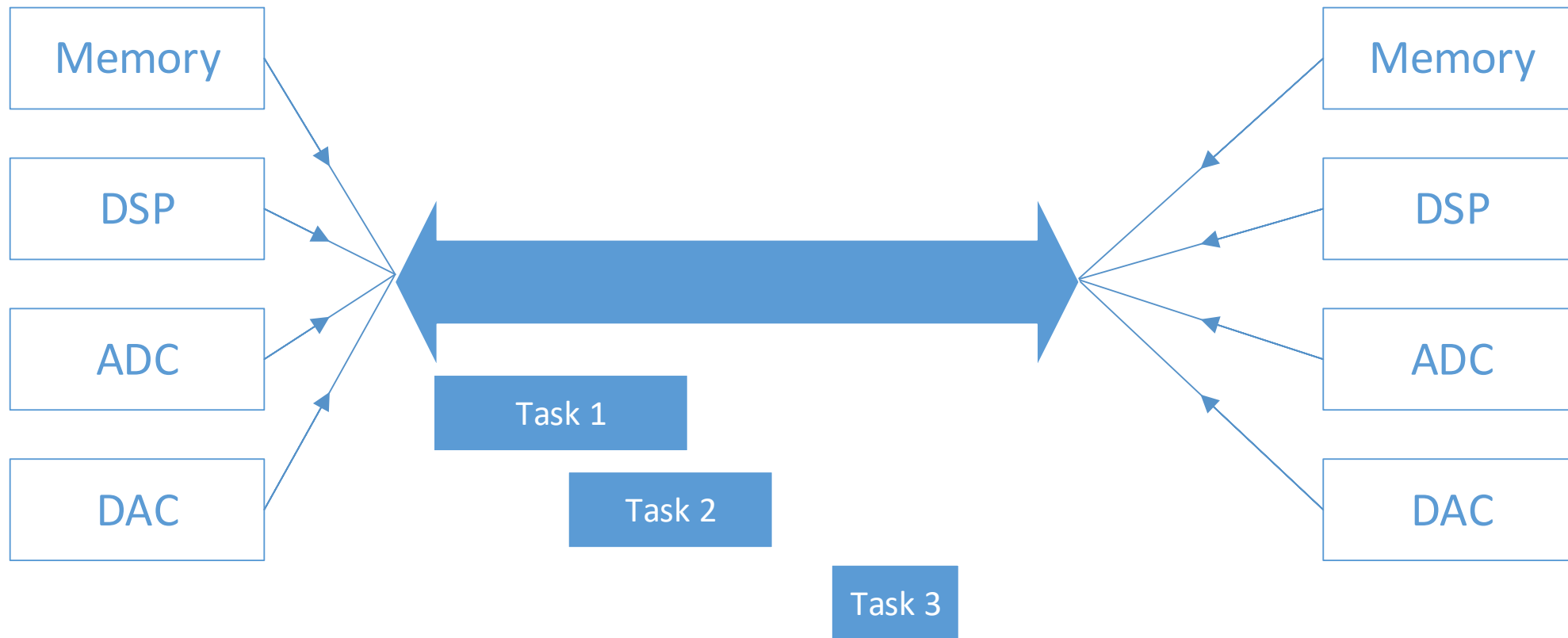
Co znajduje się wewnątrz aparatu?

Oprogramowanie

- Kontrola urządzenia
 - Zarządzanie energią, zarządzanie ładowaniem
 - Obsługa komend użytkownika (zmiana programu, regulacja głośności)
 - Obsługa parowania z innymi urządzeniami (telefony, telewizory, komputery)
- Klasyfikator otoczenia dźwiękowego
- **Przetwarzanie dźwięku, wzmacnianie konkretnych częstotliwości**
- Komunikacja między aparatami, beam-forming (kierunkowy odbiór sygnału)
- Współpraca z oprogramowaniem do dopasowania charakterystyki aparatu do wady słuchu

Co w tym wszystkim robi Python?

Analiza przepływu danych w DSP



Analiza przepływu danych w DSP



- Kilkaset komponentów logicznych w kodzie
- Kompilację każdego można sparametryzować na kilkanaście sposobów
- Różnice w ilości cykli procesora potrzebnych do wykonania kodu
- Optymalizacja → zmniejszenie liczby cykli → zmniejszenie częstotliwości pracy → zwiększenie czasu działania na baterii
- Osiągnięty zysk – około 4% * wszystkie najnowsze modele na rynku

- Migracja Matlab → Python
- Generowanie i analiza raportów wydajności
- I wiele innych...
- Przygotowanie do migracji Python2 → Python3

Migracja SVN do Git



git

Python migration cmd use

```
cmd = ['git clone', src, dst]  
run_cmd(cmd, title='git clone')
```

```
scm.checkout()
```

```
scm.get_revision()
```

```
scm.get_modified()
```



Jenkins

Jenkins stage z użyciem python script

```
stage('Running Migration') {
  steps {
    sh """#!/bin/bash
    source activate python2_dsp-g30-migration
    rm *.log
    rm *.log.*
    chmod +x /media/Share/workspace/dsp-g30-migration/authors_g30.txt
    chmod +x /media/Share/workspace/dsp-g30-migration/authors_g30.sh
    svn2git-migration \
    --destination="\$WORKSPACE\" \
    --mode=${params.mode} \
    --repository=${params.repository} \
    --config="\$WORKSPACE/configuration.json\" \
    --authors-prog="\$WORKSPACE/authors_g30.sh\" \
    --authors-txt="\$WORKSPACE/authors_g30.txt\" \
    --lfs --gc --push --log="\$WORKSPACE\"
    """
  }
}
```

CONDA

Jenkins pipeline conda environment

```
stage('Creating Environment') {  
    steps {  
        sh """#!/bin/bash  
conda env update --quiet --prune --file environment.yml --name python2_dsp-g30-migration  
source activate python2_dsp-g30-migration  
pip install -U -r requirements.txt  
"""  
    }  
}
```

- Tworzenie windowsowych plików wykonywalnych
- Łatwa dystrybucja narzędzi
- Brak potrzeby instalowania pythona i bibliotek u użytkownika końcowego

Dziękujemy!

Pytania?