

aahh 10 11

ACCADEMIA DI BELLE ARTI DI URBINO

SISTEMI INTERATTIVI DUE



# IL SUONO

Caricamento

- Il contenuto del file sonoro deve essere caricato in memoria da disco



Decodifica

- Il suono deve essere decompresso e decodificato



Esecuzione

- Il suono viene eseguito



# PLAY WHILE DOWNLOADING



# PLAY WHILE DOWNLOADING

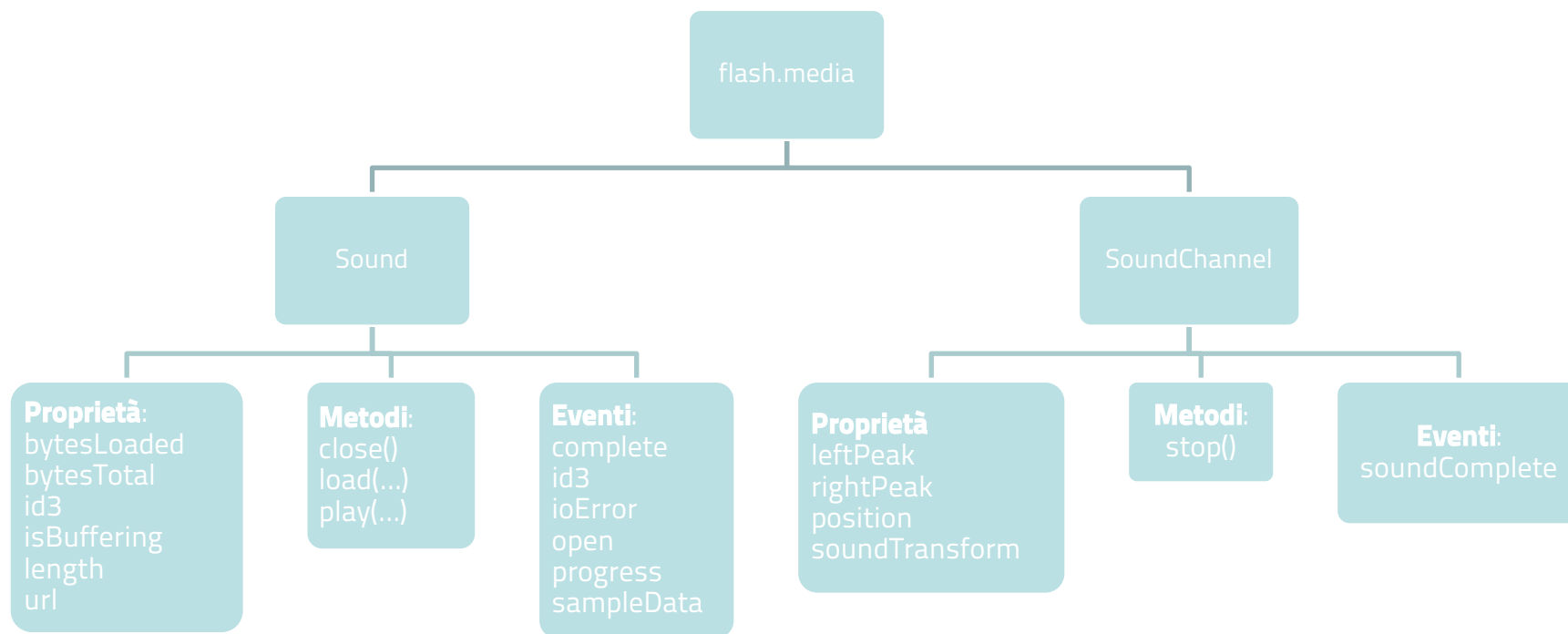
- Efficienza:
  - Rapporto tra tempo necessario per scaricare i dati e tempo necessario per la ricodifica
  - Algoritmi di compressione specifici orientati allo streaming
  - Miglioramento della capacità di calcolo

# COMPRESSIONE

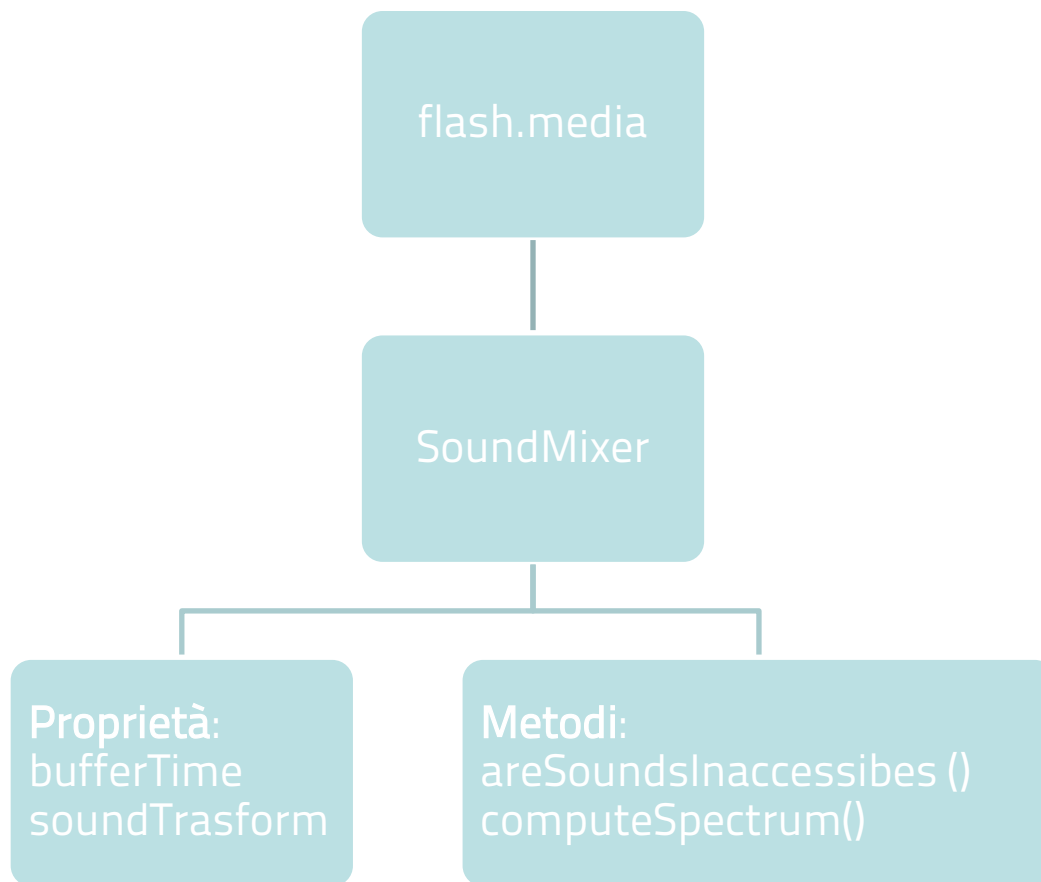
- Lo standard per il suono è MP3
  - CBR: Costant Bit Rate
  - VBR: Variable Bit Rate
- VBR:
  - Alto rapporto tra compressione e qualità
  - Flusso di dati discontinuo
- CBR:
  - Minor efficienza nella compressione
  - Garantisce un rapporto costante da byte e tempo di riproduzione.

# IL SUONO IN ACTIONSCRIPT 3

# IL SUONO IN ACTIONSCRIPT 3



# IL SUONO IN ACTIONSCRIPT 3





# IL PROCESSO

1

- Creo un'istanza della classe sound:  
`var suono:Sound = new Sound();`

2

- Carico il suono:  
`suono.load(new URLRequest(mioFile));`

3

- Assegno l'esecuzione del suono ad un canale:  
`var ch:SoundChannel= mioSuono.play();`

# EVENTI DELLA CLASSE SOUND

complete

- Viene lanciato quando sound ha completato il caricamento del file.

id3

- Lanciato quando sono a disposizione i metadati presenti nei file mp3.

open

- Lanciato quando inizia il caricamento

progress

- Usato per rilevare lo stato del caricamento

sample data

- Può essere usato per creare suoni sintetici.

# USO DI SOUNDCHANNEL

leftPeak

- **Proprietà:** volume del canale sinistro

position

- **Proprietà:** posizione corrente

rightPeak

- **Proprietà:** volume del canale destro

soundTransform

- **Proprietà:** oggetto che definisce volume, panning e altre proprietà del suono.

stop()

- **Metodo:** interrompe l'esecuzione.

# USO DI SOUNDMIKER

bufferTime

- **Proprietà:** quanto suono viene precaricato durante lo streaming

soundTrasform

- **Proprietà:** l'oggetto soundTrasform che controlla tutti i suoni.

areSounsInaccessible

- **Metodo:** determina se esistono suoni inaccessibili per problemi di dominio

cumputeSpectrum

- **Metodo:** consente di analizzare un *pezzo* di suono.

stopAll()

- **Metodo:** interrompe l'esecuzione di tutti i suoni.

aaahh 10 11

ACCADEMIA DI BELLE ARTI DI URBINO

SISTEMI INTERATTIVI DUE

# IL SUONO IN HYPE

# SOUNDANALYZER

destroy()

- **Metodo:** distrugge l'istanza di SoundAnalyzer e libera risorse.

getFrequencyIndex()

- **Metodo:** ottiene il livello di attività di una determinata frequenza.

getFrequencyRange()

- **Metodo:** ottiene il livello di attività media di una determinata banda di frequenze.

getOctave()

- **Metodo:** ottiene il livello di attività media di una determinata ottava.

stop()

- **Metodo:** Ferma SoundAnalyzer.

# SOUNDANALYZER

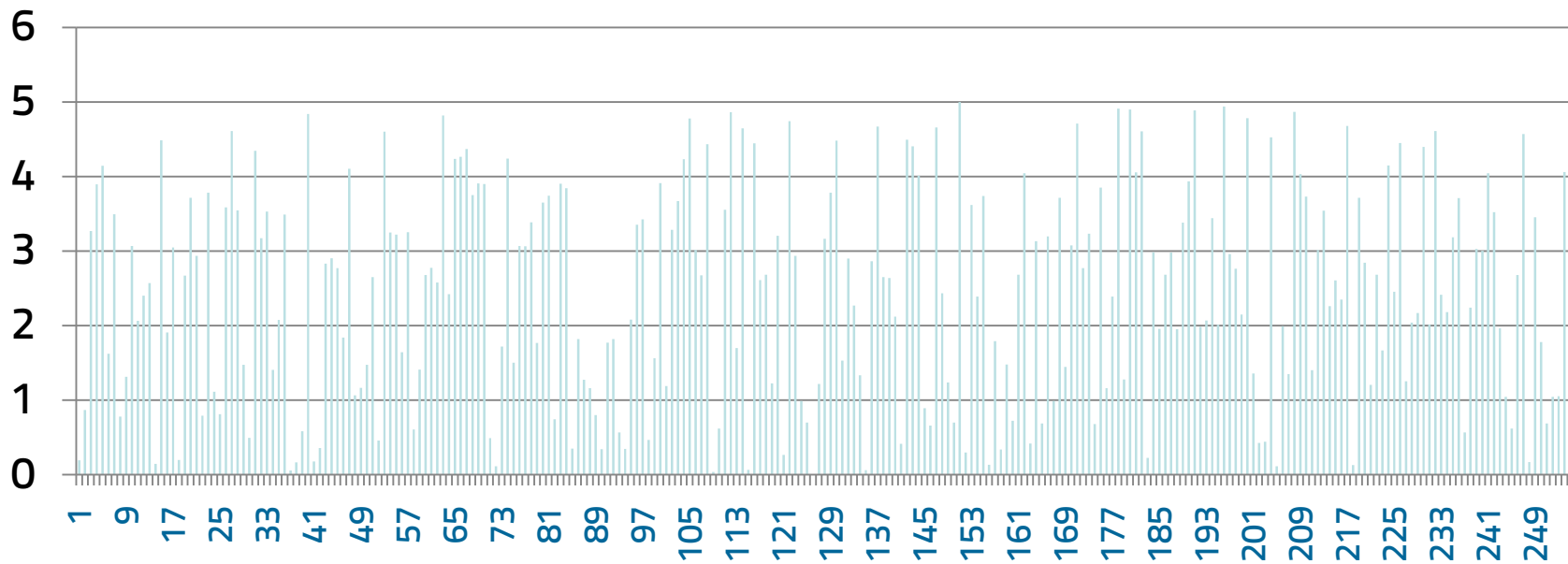
- La frequenza udibile è divisa in 256 bande
- `getFrequencyIndex` controlla una delle bande di frequenza e ne restituisce l'intensità

```
getFrequencyIndex( index:uint ,  
                    min:Number ,  
                    max:Number ) :Number
```

**index** -> banda di frequenza da rilevare

**min**, **max** -> valori minimo e massimo  
restituiti

## getFrequencyIndex()

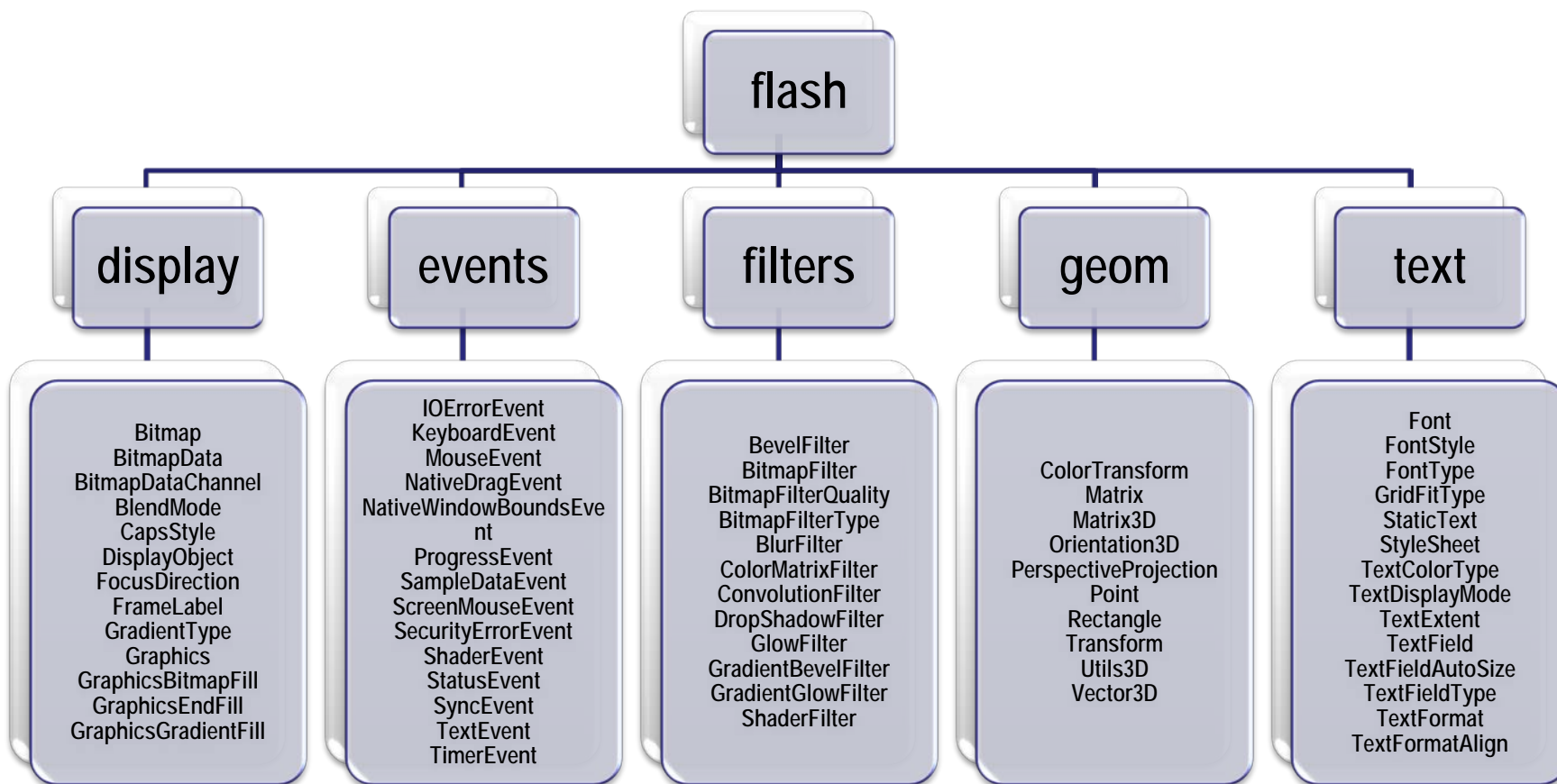


```
for (var i = 0; i < 265; i++) {  
    getFrequencyIndex(i, 0, 5);  
}
```

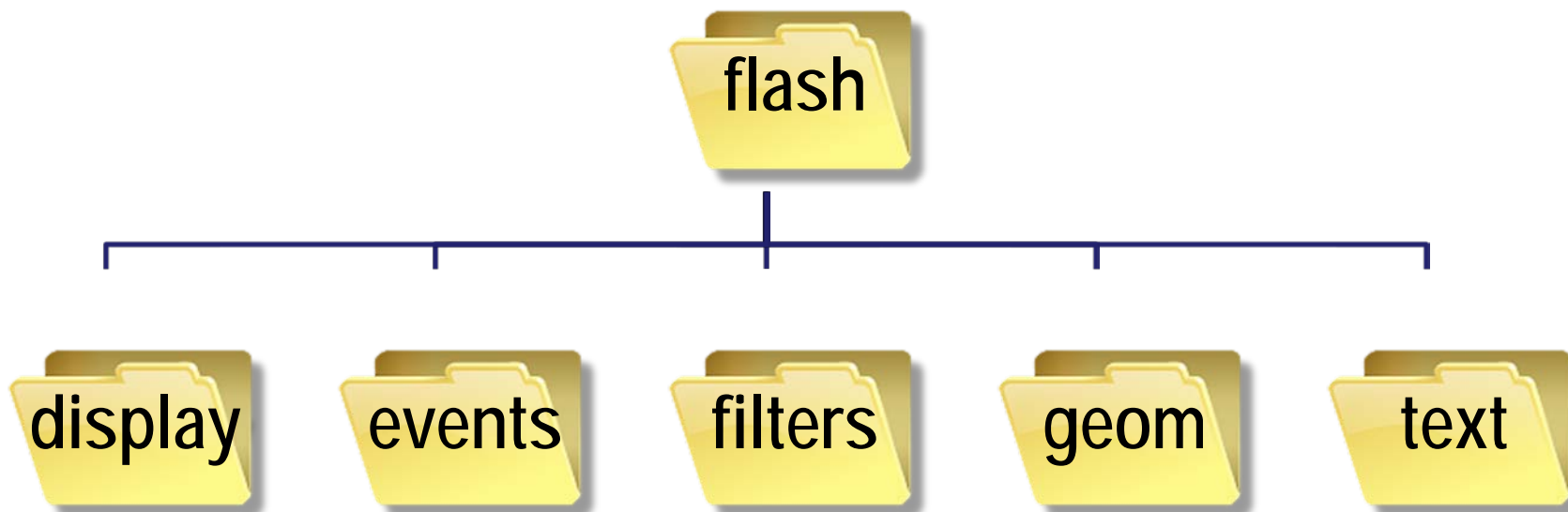


# USARE UNA LIBRERIA DI CLASSI ESTERNA

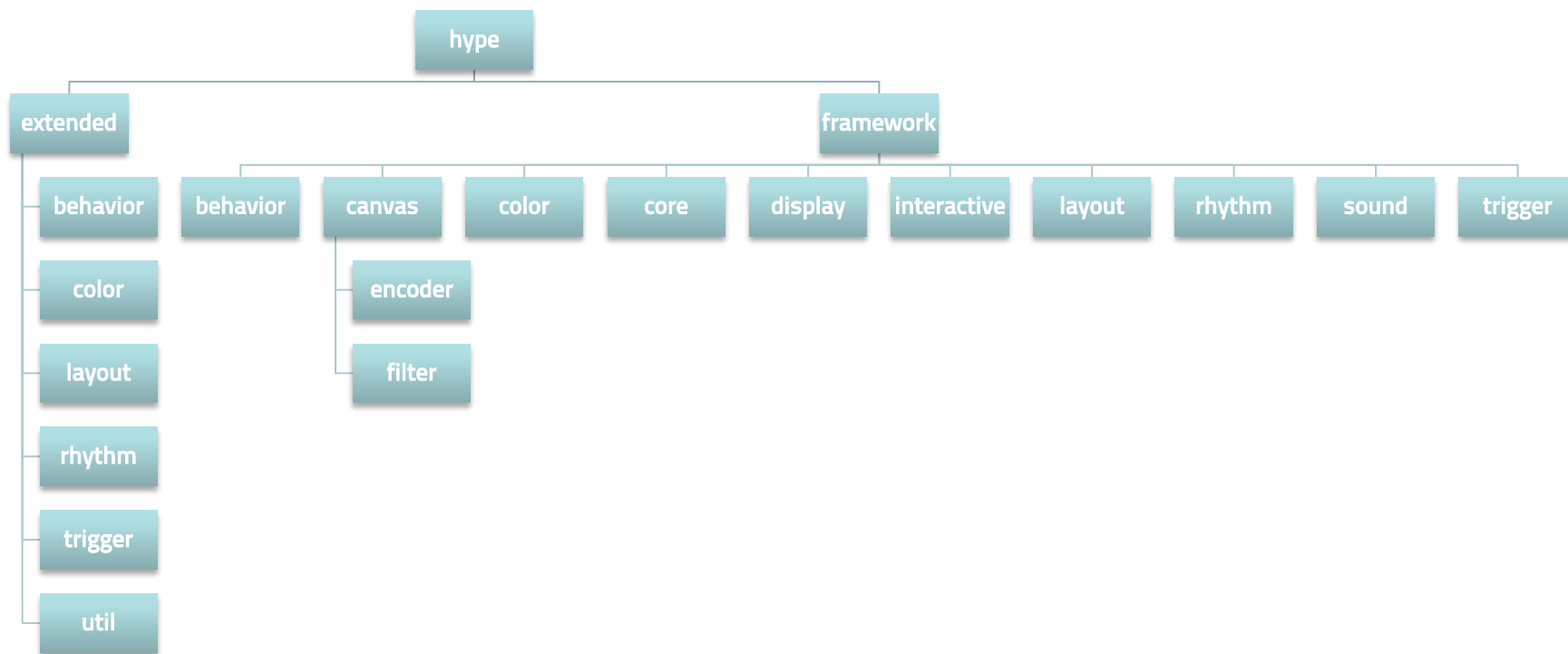
# ANCORA SUI PACKAGES



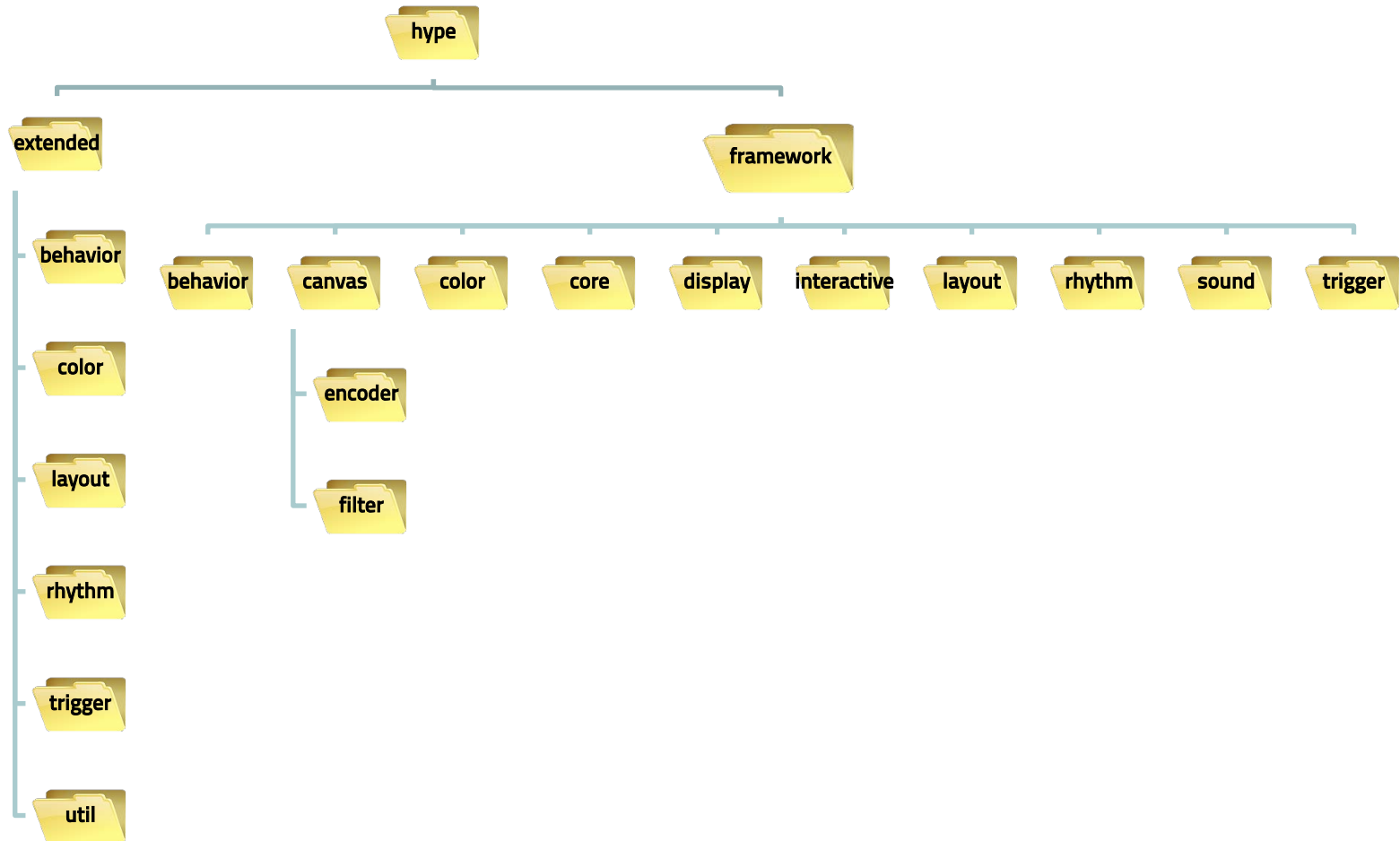
# ANCORA SUI PACKAGES



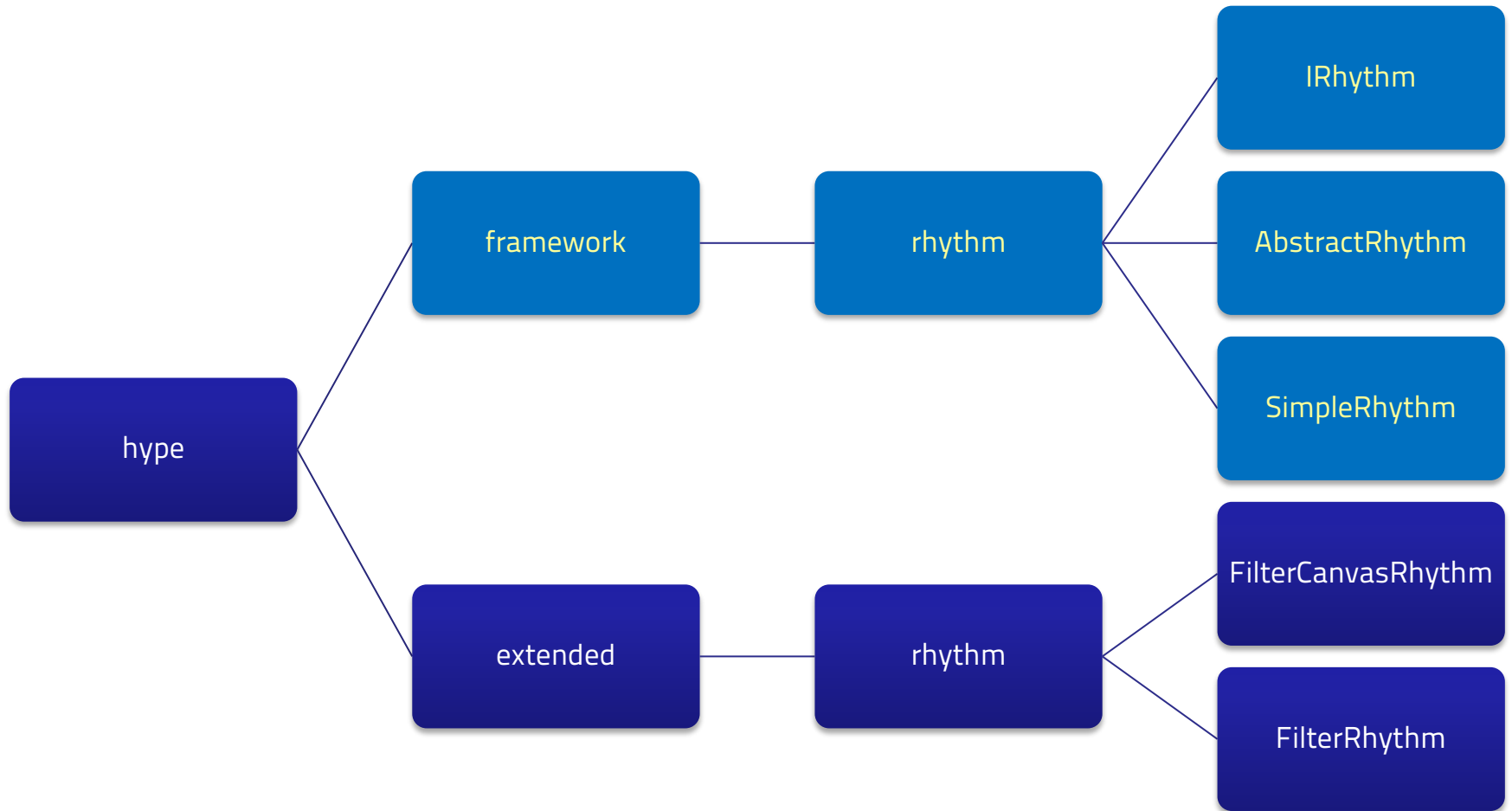
# I PACKAGES DI HYPE



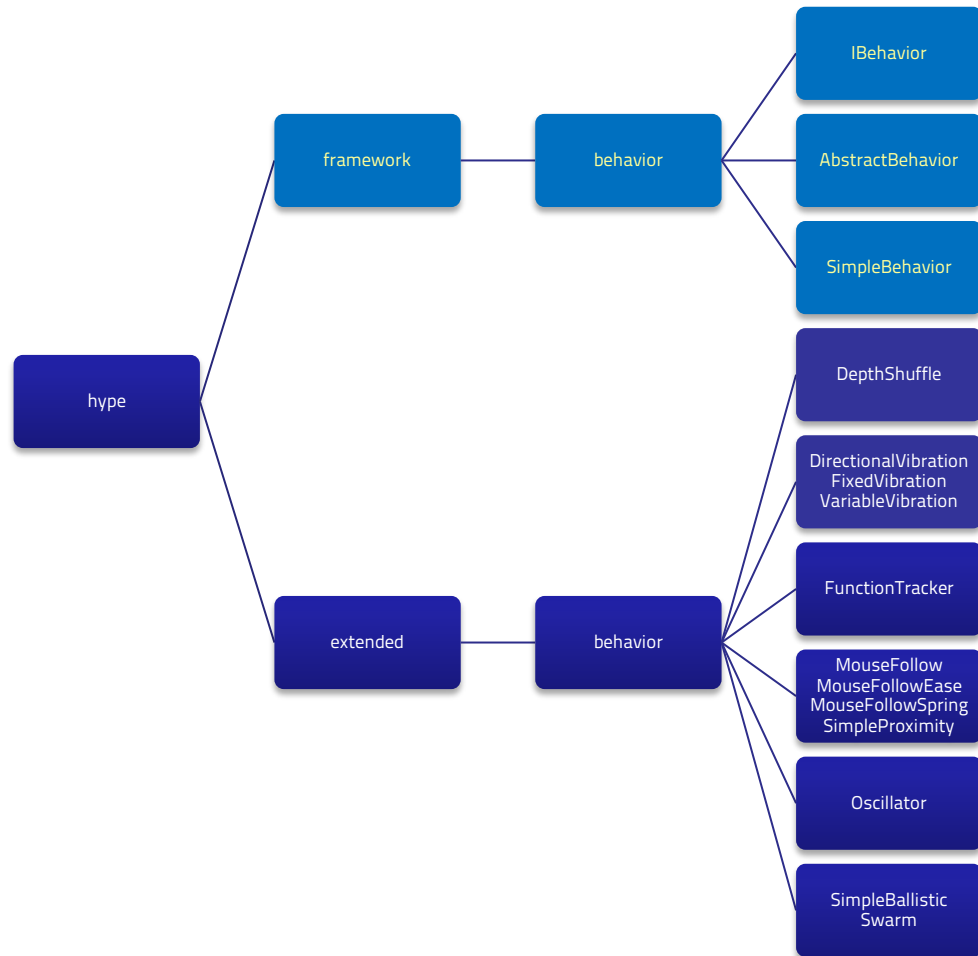
# I PACKAGES DI HYPE



# RHYTHM



# BEHAVIOR



# TRIGGER

