

# NETRONOME

## INTRODUCTION

# Preface

The Internet is a vast network that hosts an incredible amount of activity at any given moment and most, with understandable reasoning, choose to just not think about it. I wanted expose people to the reality of the Internet's prevelance in our society. My intention when starting this project was to create something interactive that combined sound with the staggering number of individuals performing an “ordinary” Internet task at any time.

## INTRODUCTION

# Contents

### EXPLORATION

- 1 Internet Stats
- 2 Sketched Directions

### SOLUTION

- 1 Foundation
  - 1.1 Adding sound
  - 1.2 Adding another sound
  - 1.3 Adding another sound
  - 1.4 Intended results
- 2 Information Architecture

### WIREFRAMES

- 1 Default state
  - 1.1 Default state continued
  - 1.2 Default state continued
  - 1.3 Adding another sound
  - 1.4 Intended results
- 2 Instrument token
  - 2.1 Instrument token continued
- 3 Internet Event token
- 4 Data Exploration

### VISUAL

- 1 Mood board
- 2 Early direction
- 3 Submitted comp
- 4 Animated demo

# **EXPLORATION**

# Internet events per second



**130,027**

YouTube views



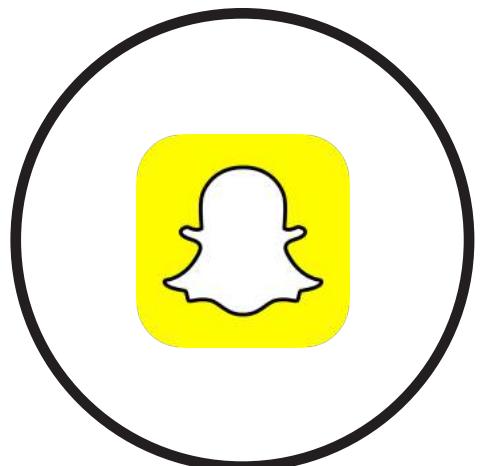
**56,107**

Google searches



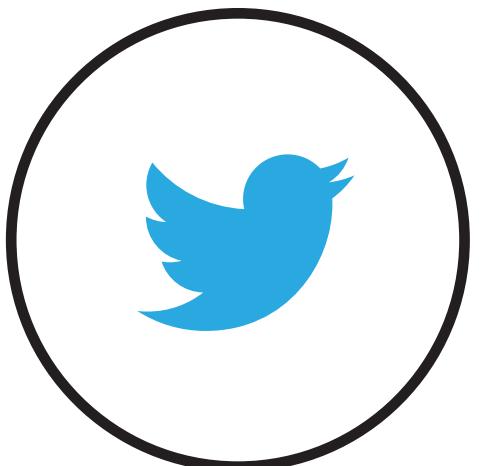
**28,000**

Instagram likes



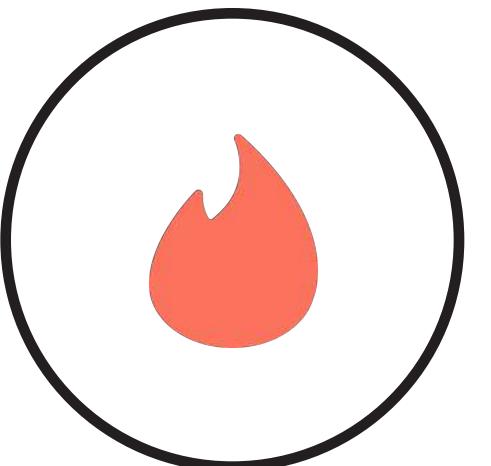
**8,796**

Snapchats sent



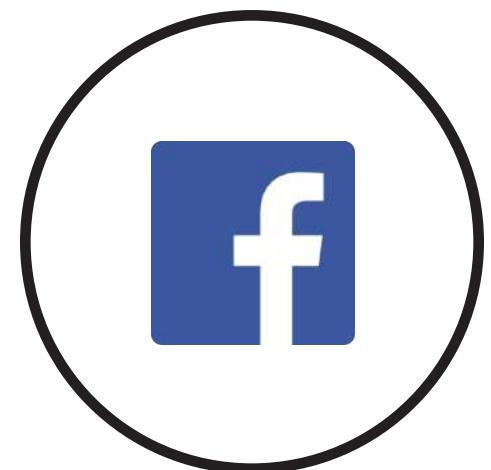
**7,325**

Tweets



**6,944**

Tinder swipes



**4,885**

Facebook statuses



**2,253**

Skype calls



**1,153**

Tumblr posts



**800**

iOS app downloads



**439**

Yelp reviews



**277**

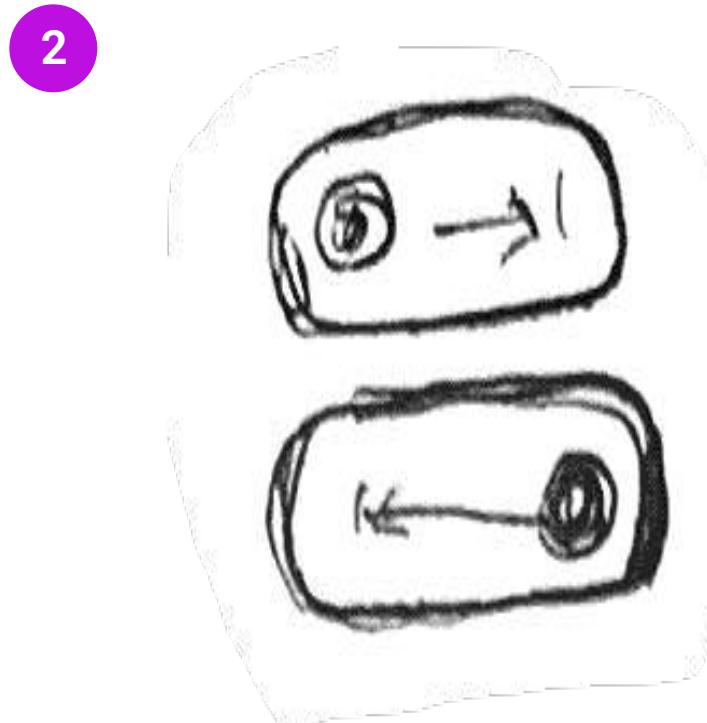
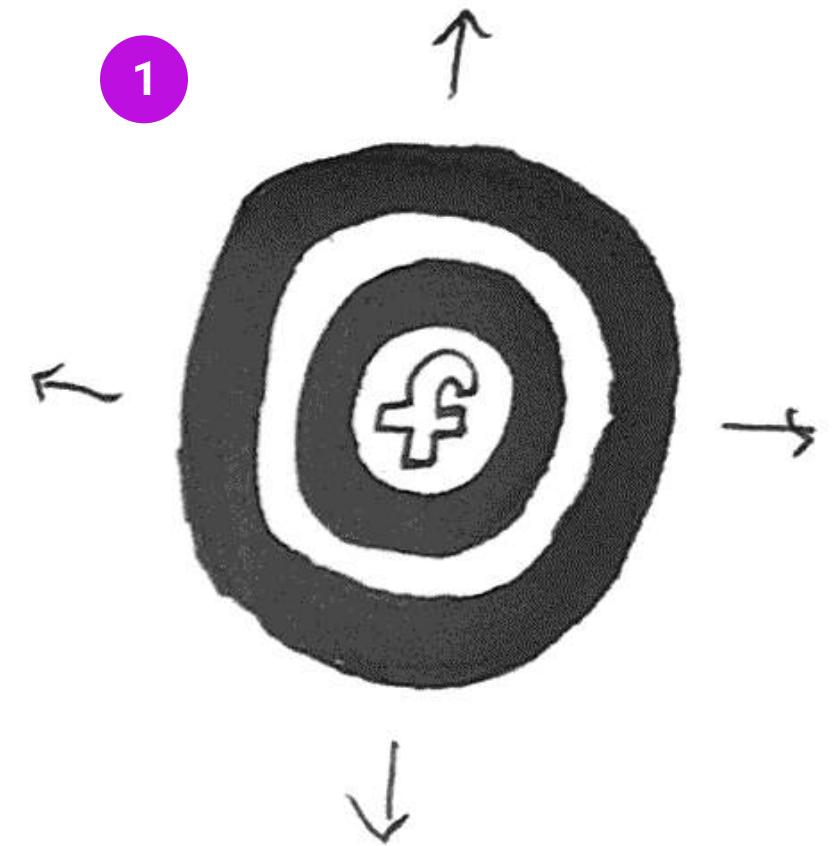
Reddit votes

## EXPLORATION 2

# Sketched directions

### 1 Ringed Orbs

Size and speed of the constantly expanding rings is determined by the frequency of the event. An ominous sound plays.

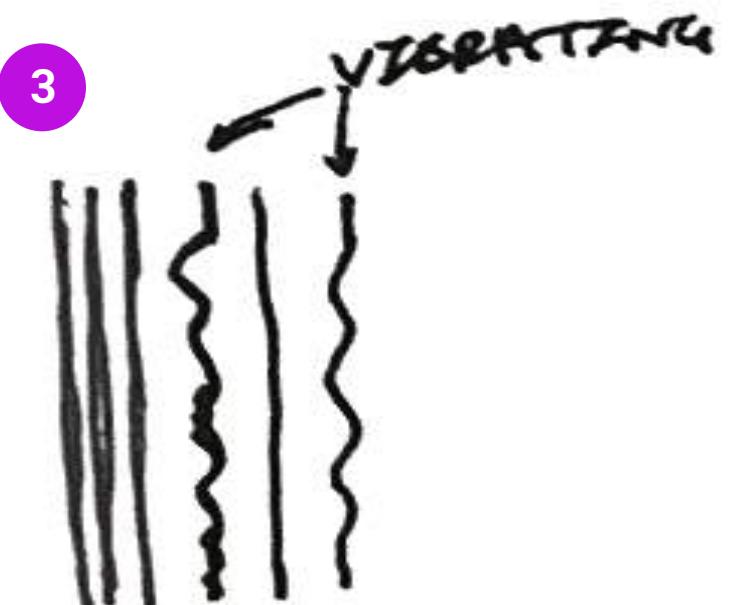


### 2 Contained Animations

A collection of shapes each with a unique, contained rhythmic animation. A short rhythmic sound aligns to the animation.

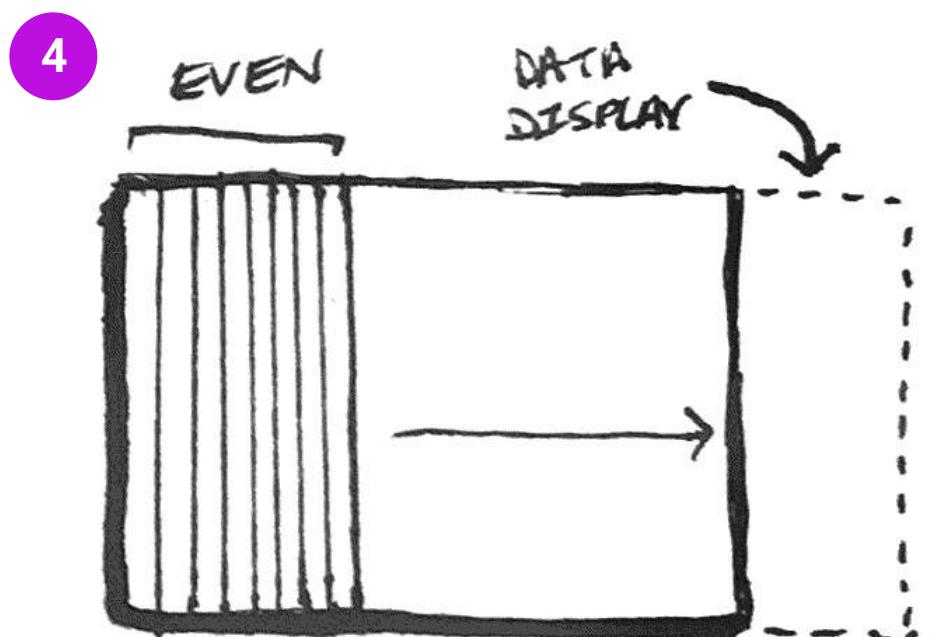
### Vibrating Strings

3 Strings vibrate according to the frequency of an assigned internet event and play sound at a pitch that matches the data.



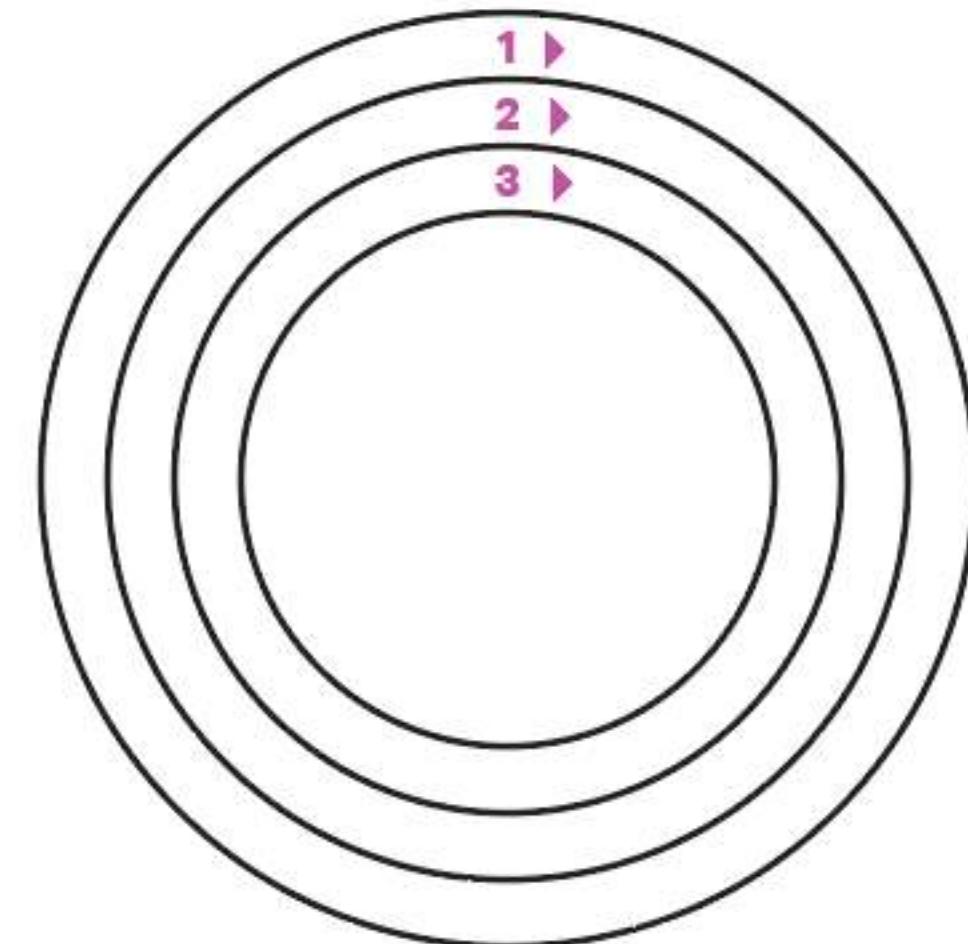
### Barcode Boxes

4 Boxes with lines that animate across, creating a fill, at a rate based on the frequency of the assigned event. The sound is brash and industrial.

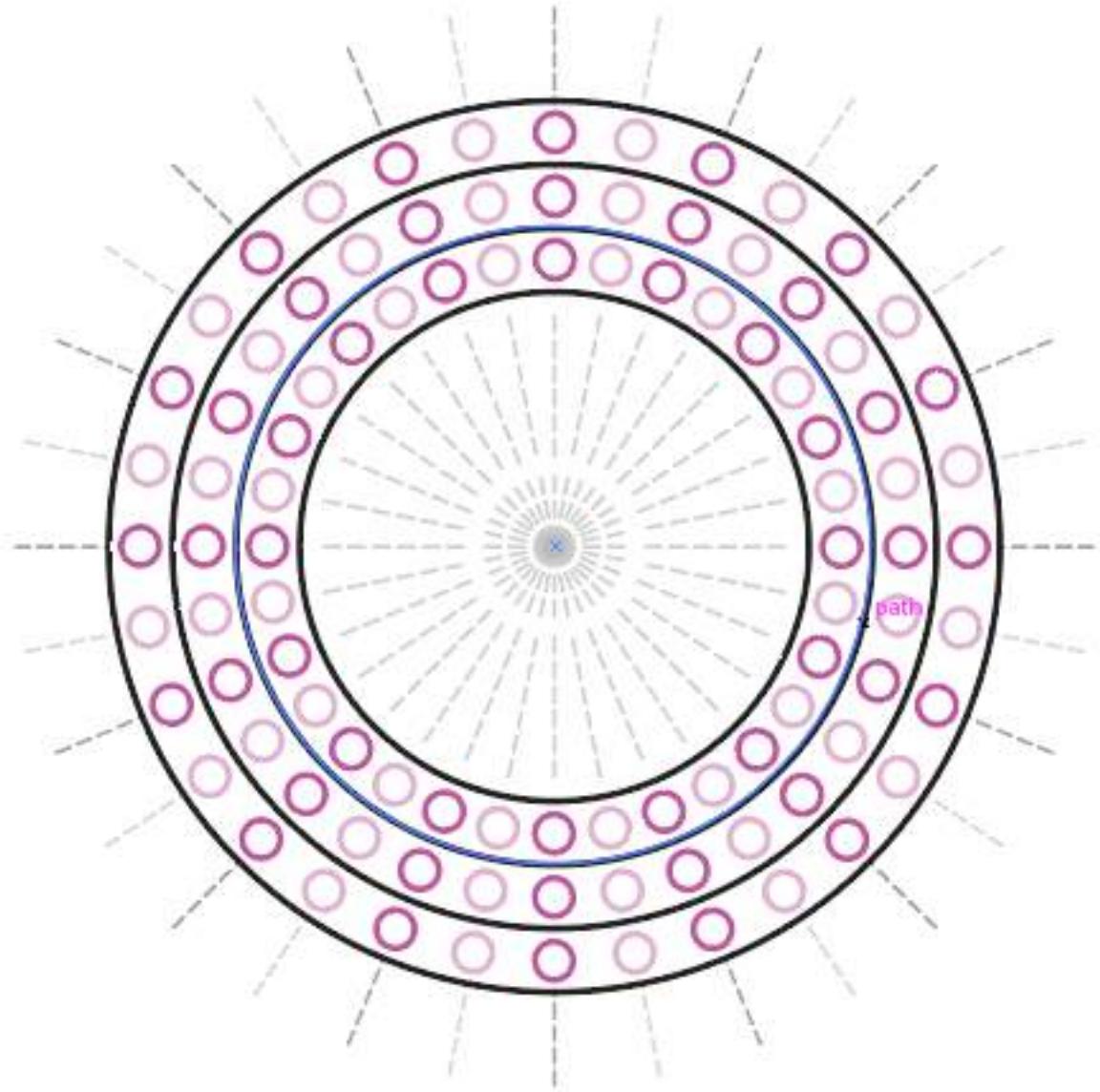


# **SOLUTION**

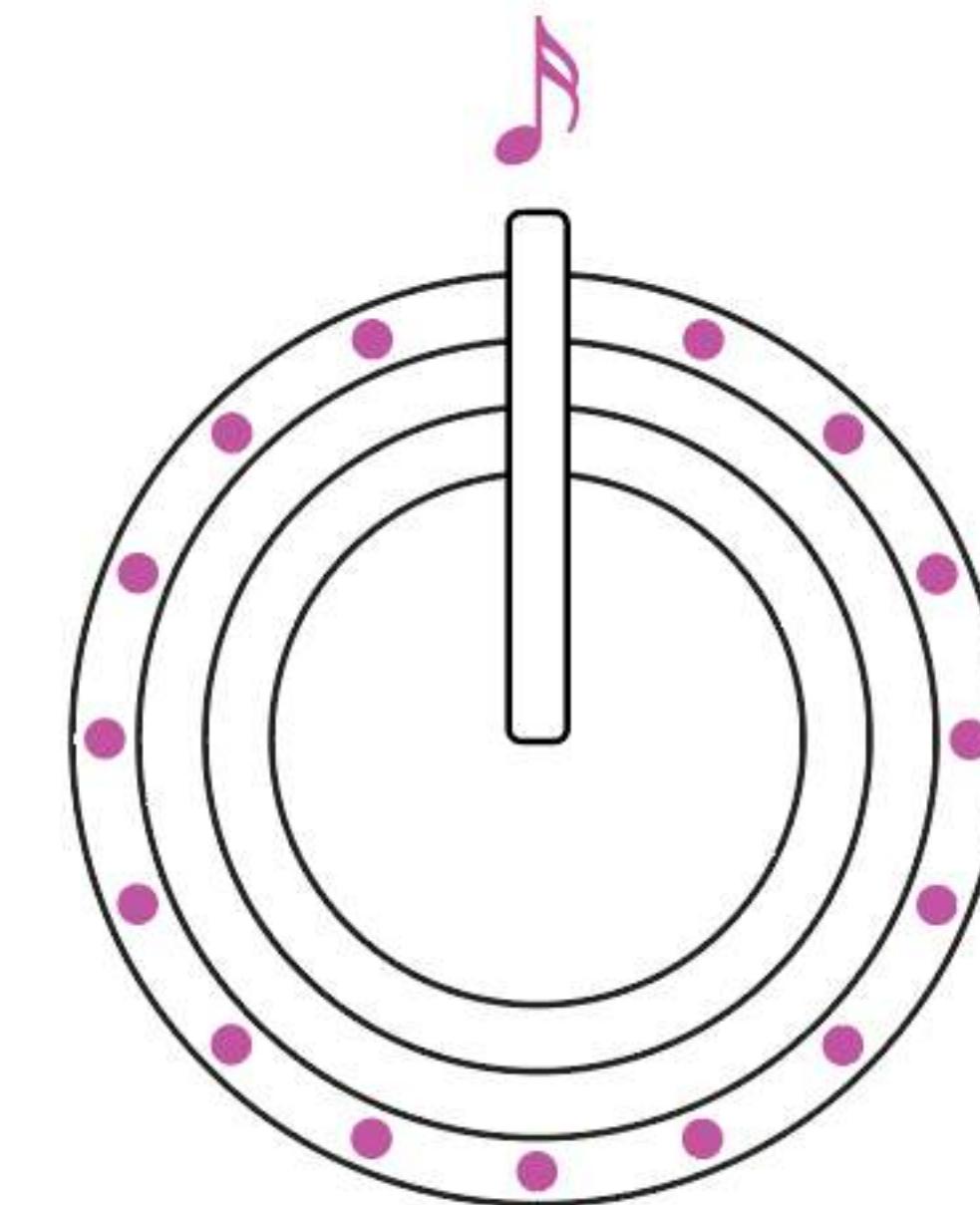
# Foundation

**Rotating Tracks**

Each instrument is made up of 3 rings each representing a different sound from that instrument. The rings rotate together at a constant rate.

**Segmentation**

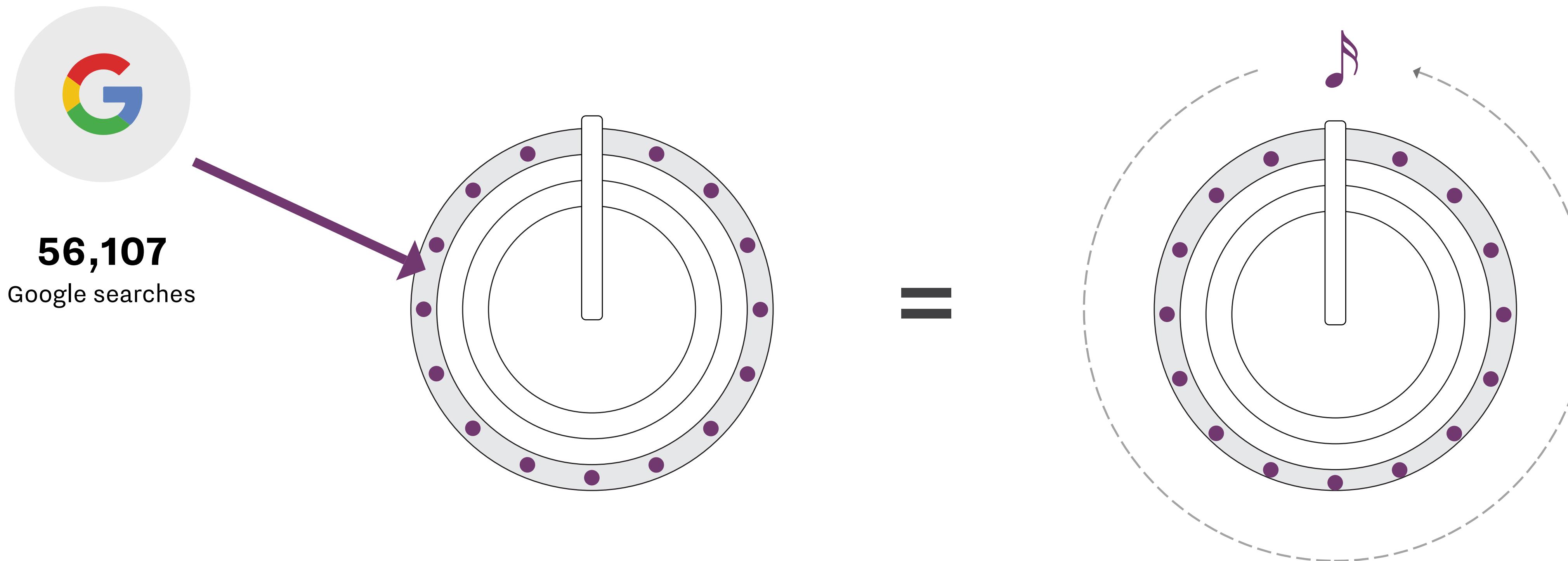
Each ring is divided into thirty-two segments with a spot for a sound available on each of the dividing lines. More frequent events fill up more sound spots.

**Sound Bar**

Sound is created when an occupied sound spot passes through the sound bar. Notes in the same spot in each ring will pass at the same time.

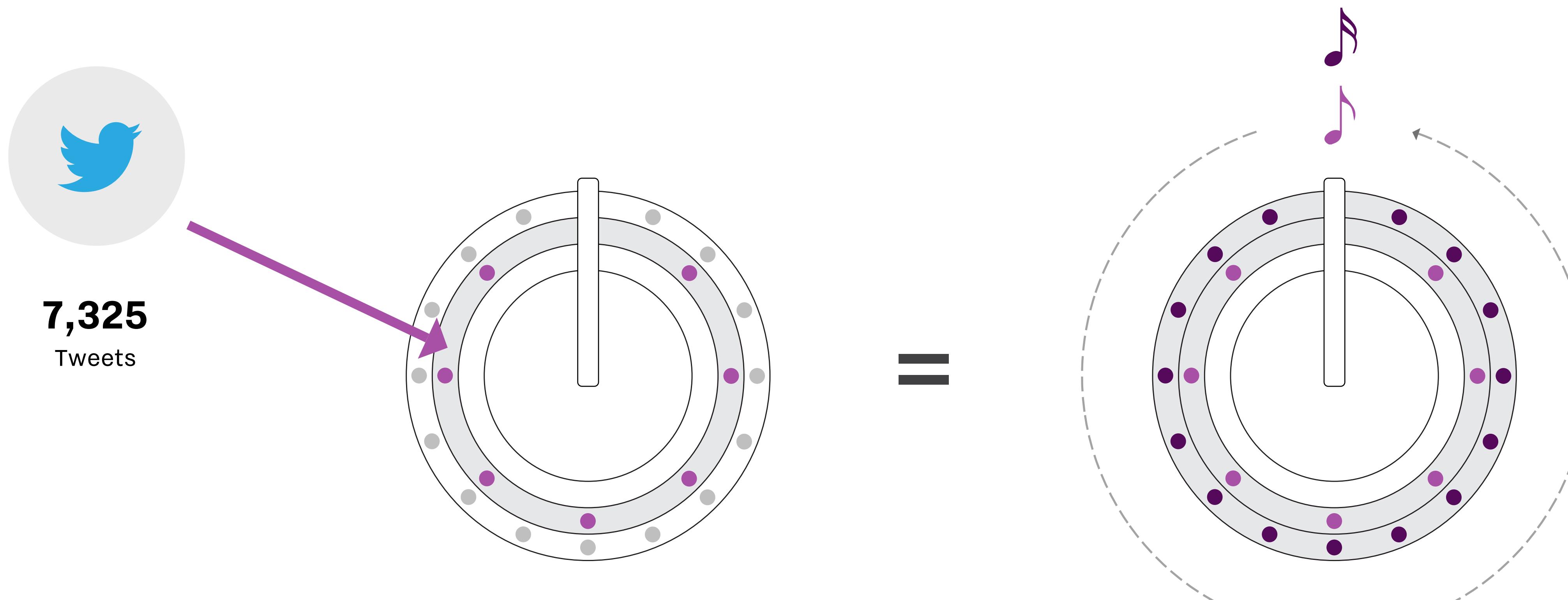
SOLUTION 1.1

# Adding sound to an instrument



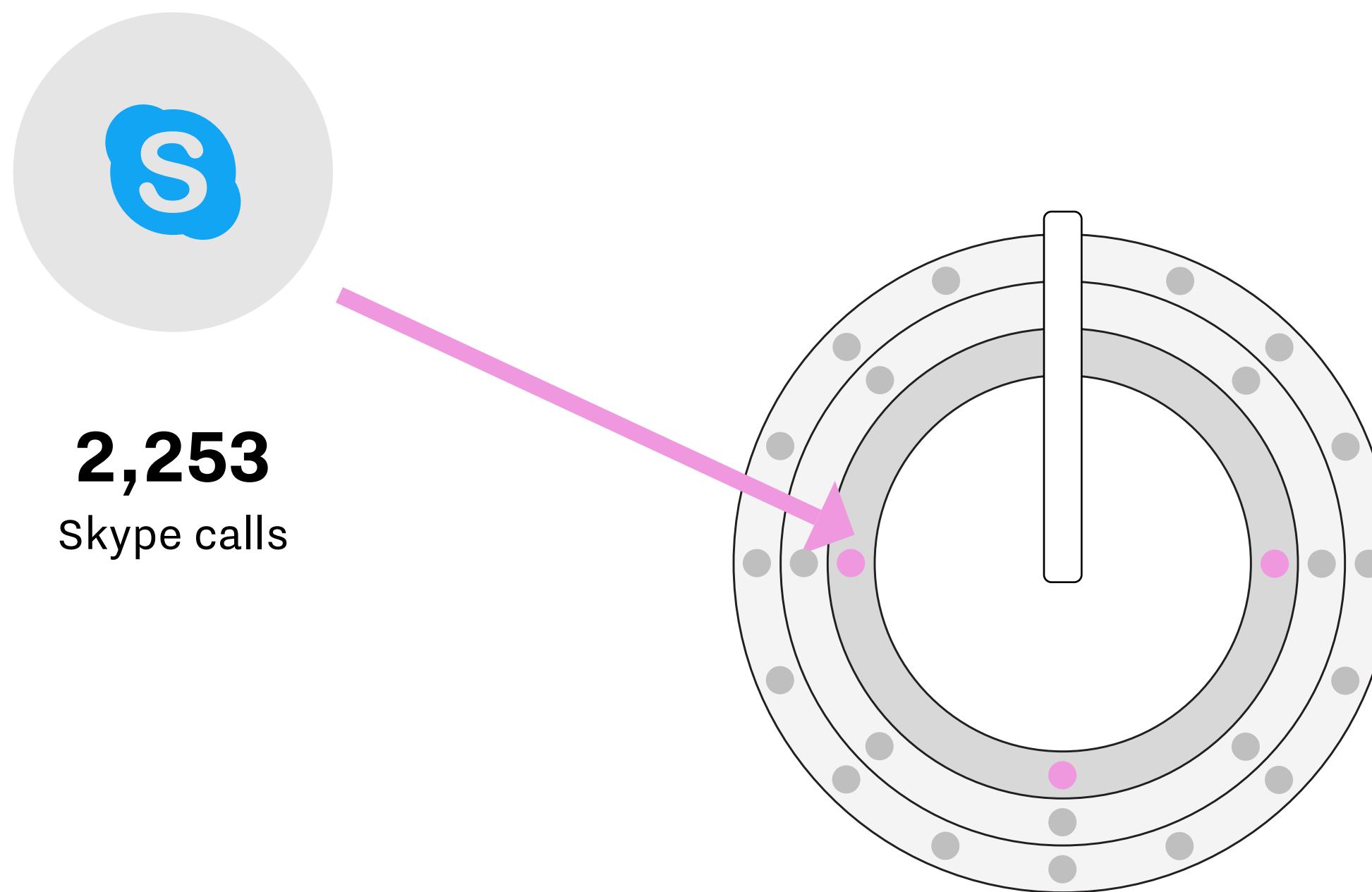
SOLUTION 1.2

# Adding another sound to an instrument

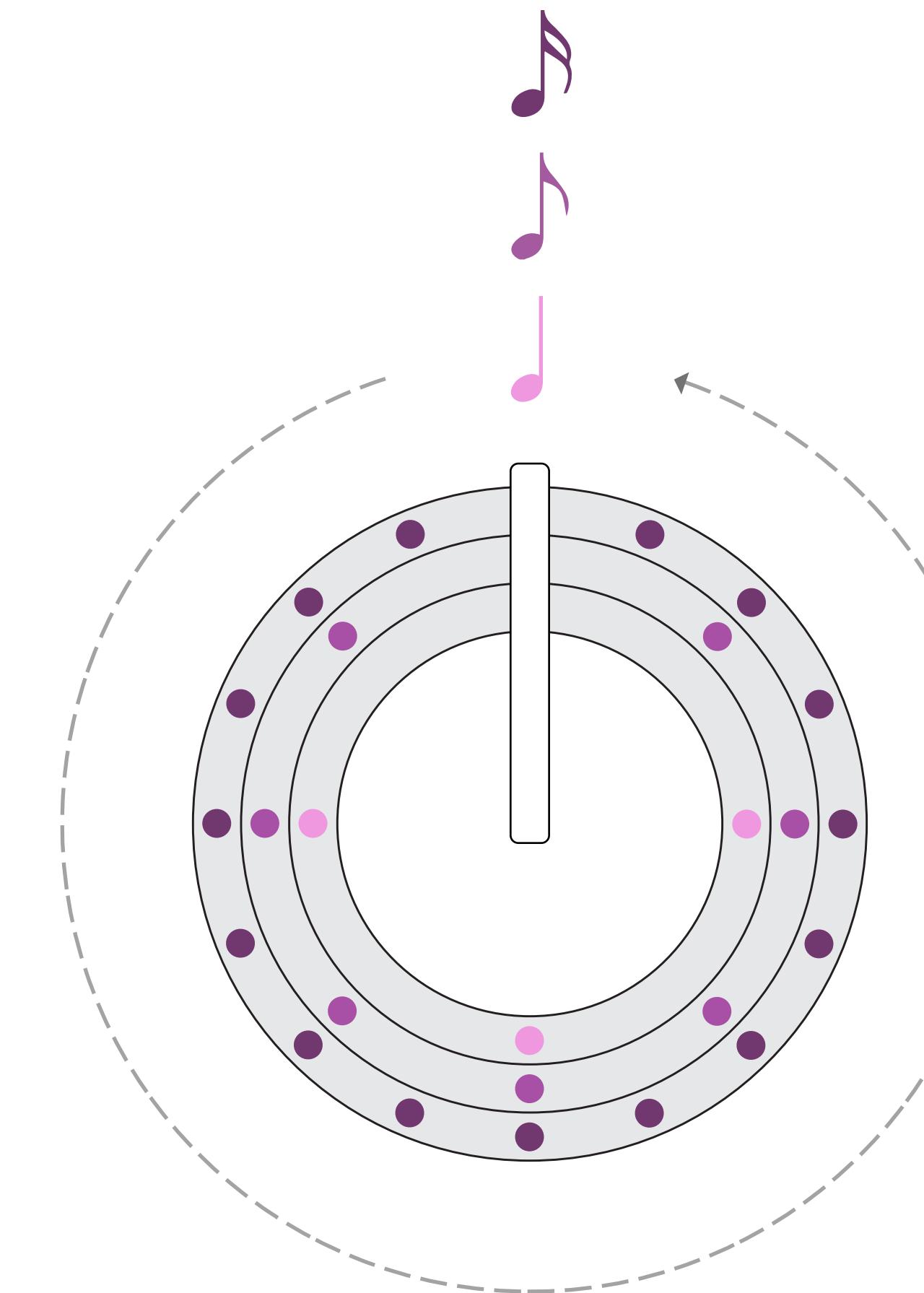


SOLUTION 1.3

# Adding another sound to an instrument

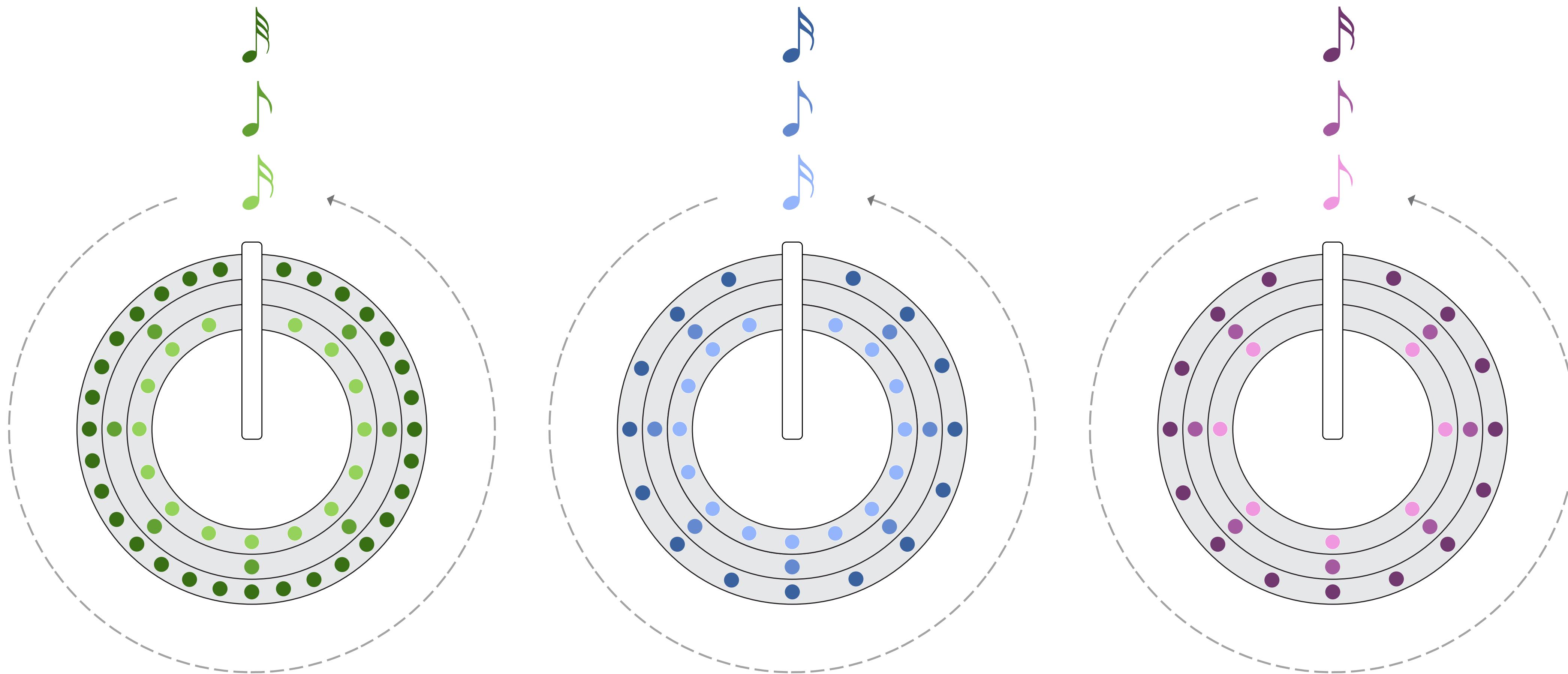


=



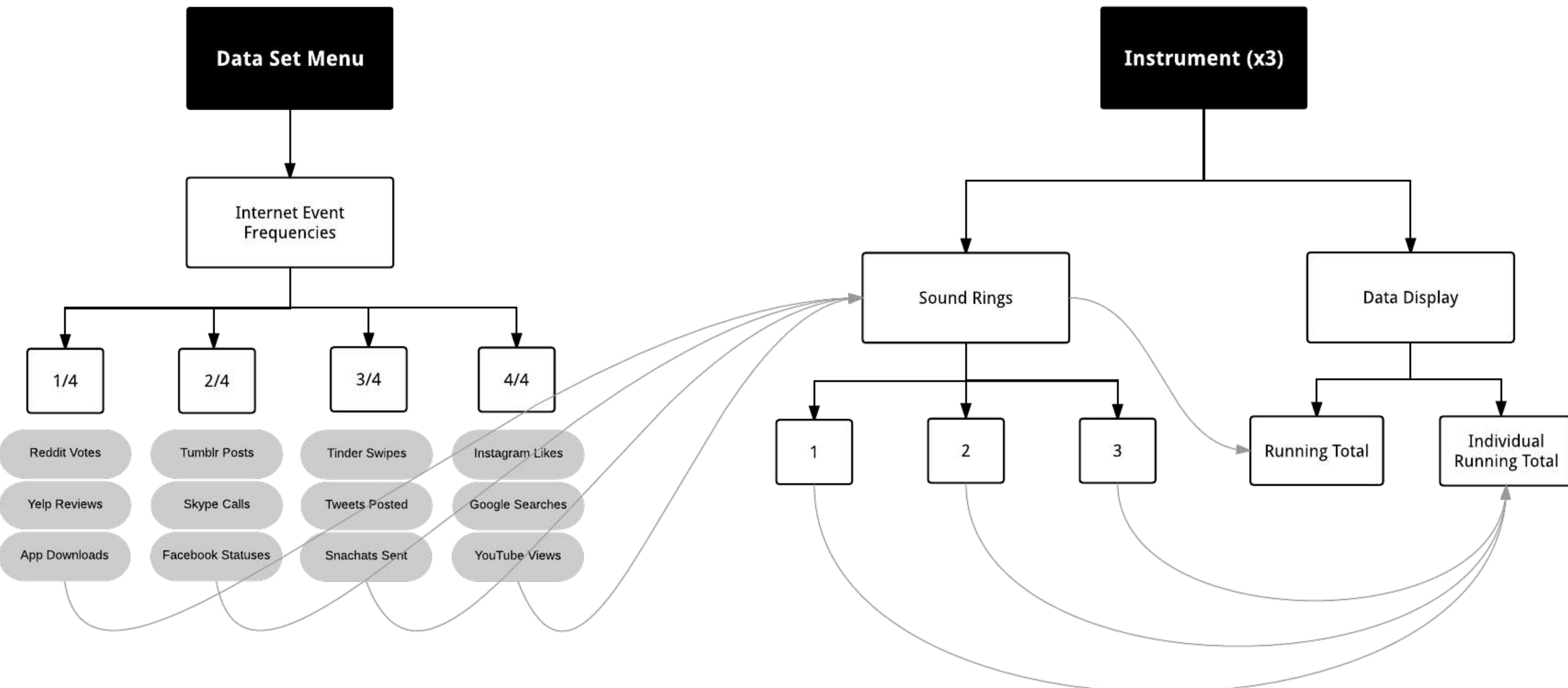
SOLUTION 1.4

# Intended result



SOLUTION 2

# Information Architecture



# **WIREFRAMES**

# Default State wireframe

**1 Timer**

Represents the elapsed time since the instruments started playing. Displaying the seconds that have passed reinforces that the data set is the frequency of an internet event per second.

**2 Play/Pause controls**

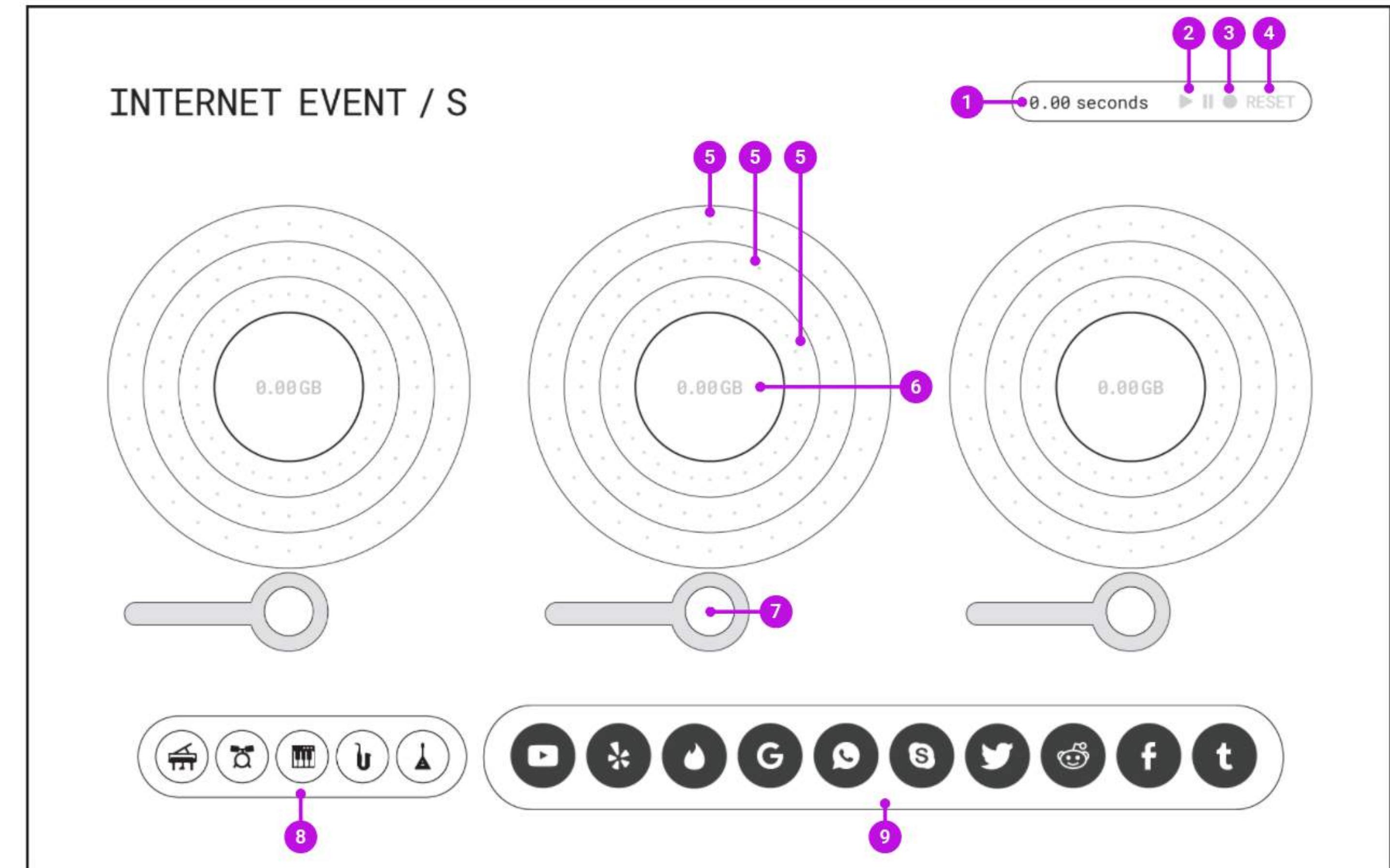
Allows the user to start and stop the instruments from playing. Located at the top and near the overall timer to reinforce that it controls the entire piece.

**3 Record button**

Will record the current loop created by the user and allow them to share it as an MP4 or MP3 file across various platforms.

**4 Reset button**

Resets the all intruments to completely blank and sets the timer back to 0.



# Default State wireframe

**5 Sound rings**

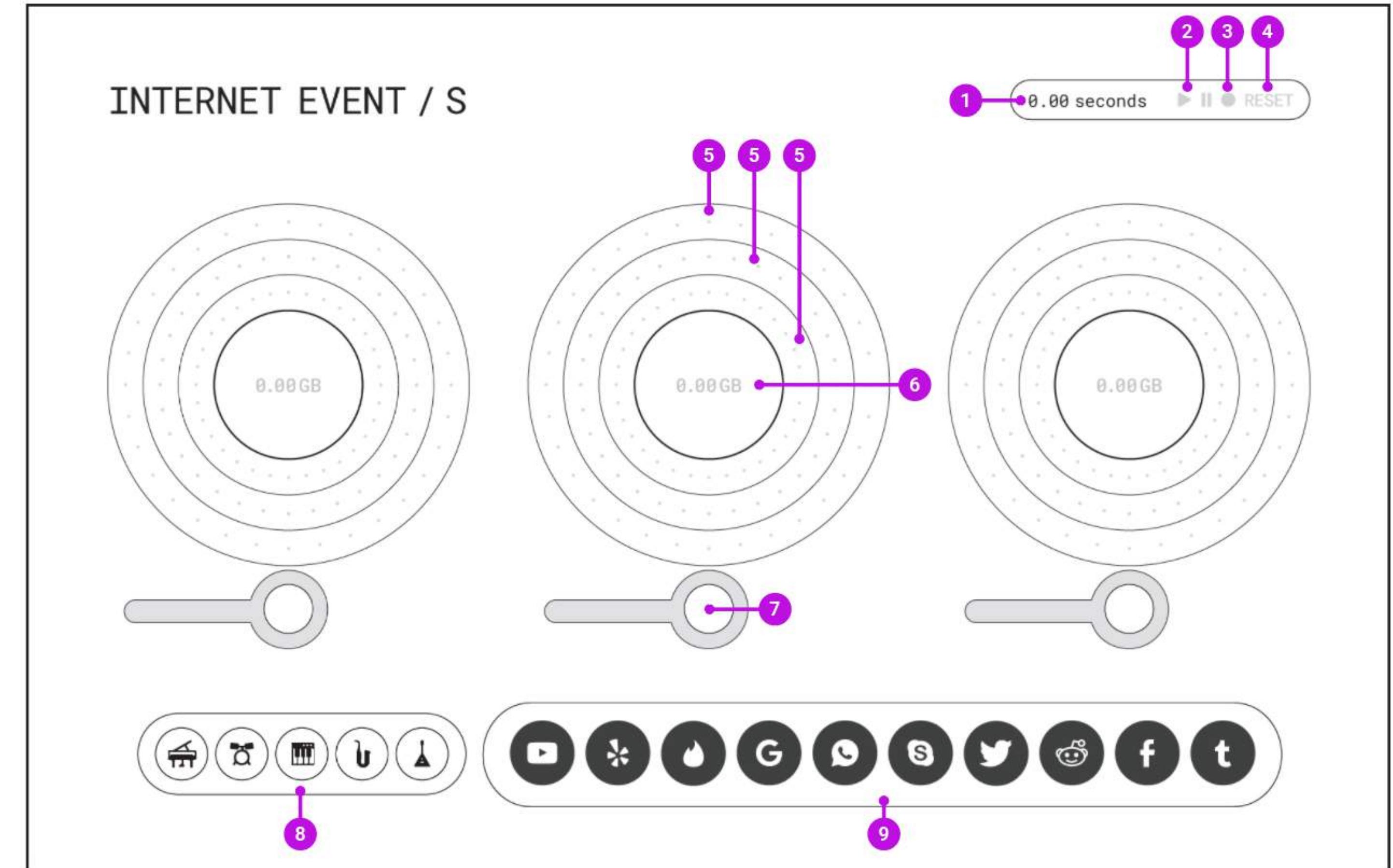
Accept dragged and dropped internet event tokens in order to be filled. Rings cannot be engaged until the instrument is turned on. Each ring represents a different tone/sound from the assigned instrument.

**6 Instrument data**

Displays the gigabytes per second currently being processed by the instrument given the internet even tokens assigned to its sound rings.

**Sound pin (empty)**

Accept dragged and dropped instrument tokens in order to be filled. Currently in the off state, assigning an instrument rotates the pin and turns the entire instrument on.



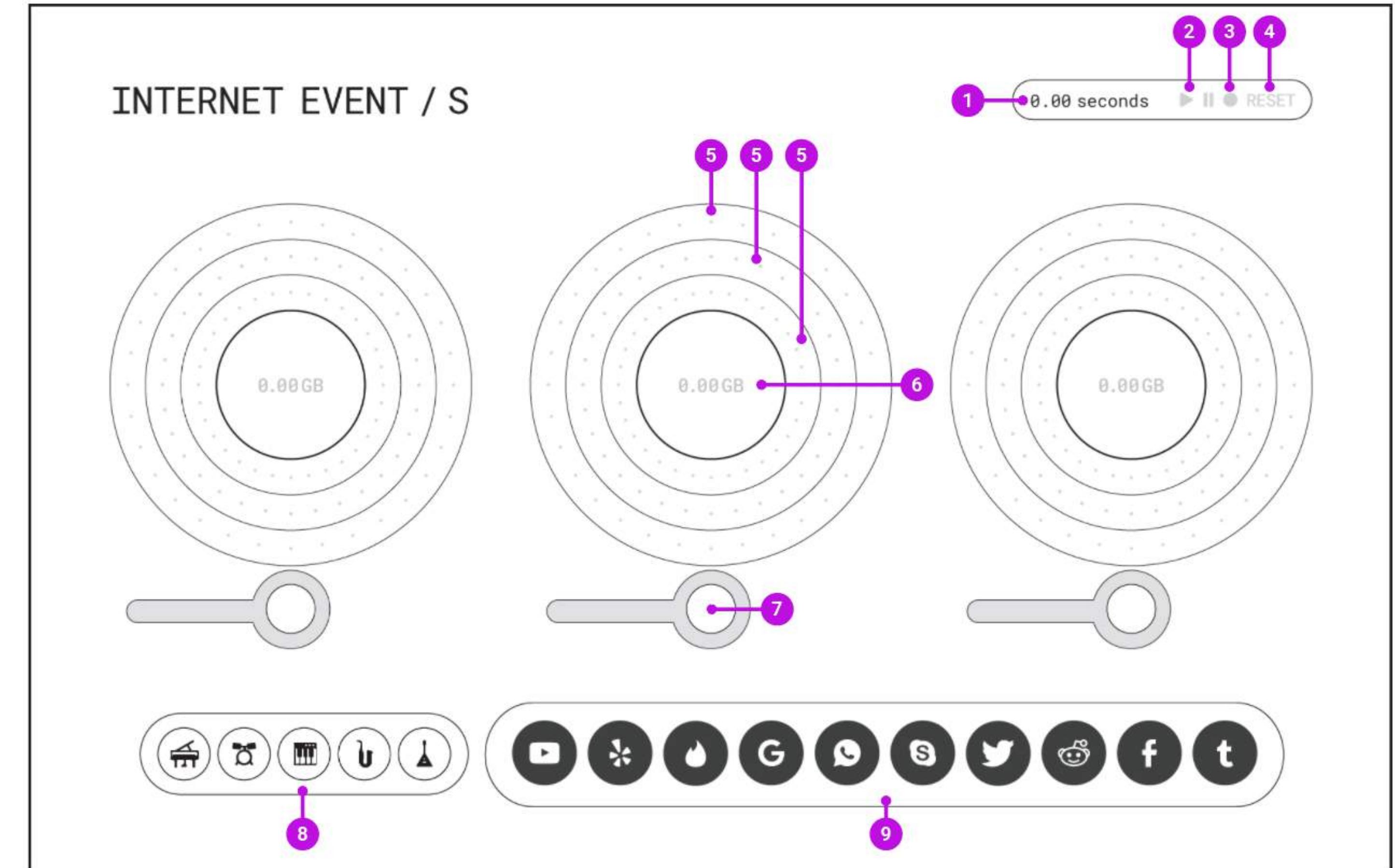
# Default State wireframe

## 8 Instrument tokens

Tokens representing instruments available to be assigned to empty sound pins. Located at the bottom to remain in vicinity of the pins and make it a short trip between them.

## 9 Internet event tokens

Tokens representing the internet events available to be assigned to sound rings. Located at the bottom so offer resemblance to the dock in MacOS and therefore seem engagable.



# Instrument Token Added wireframe

**1 Timer**

Begins automatically once any instrument has been turned on.

**2 Play/Pause controls**

Play button is automatically engaged once an instrument has been turned on. This hints at the buttons ability to control the instruments.

**3 Sound rings**

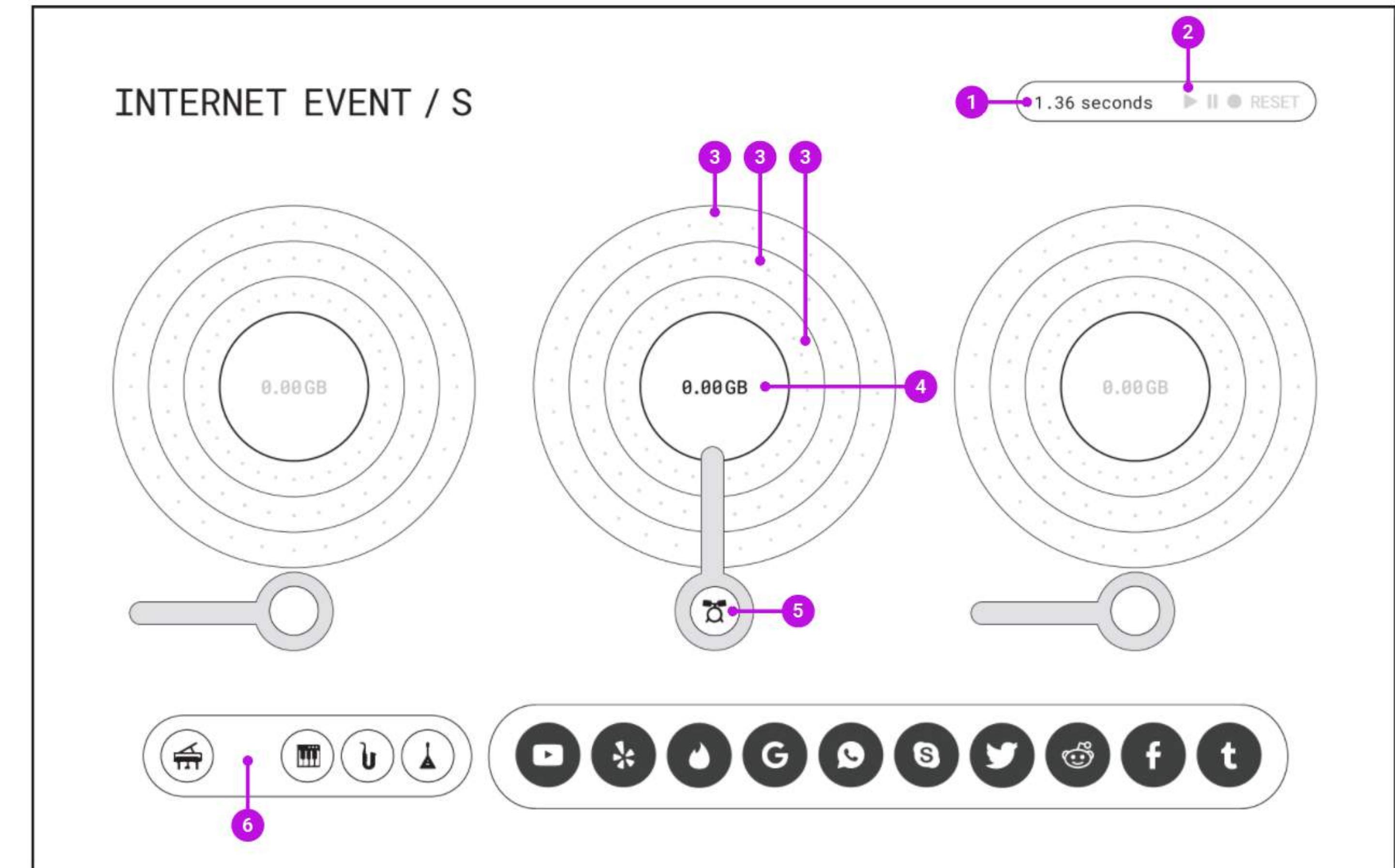
Automatically begin spinning in unison once the pin has been engaged even if they haven't had data added yet.

**4 Instrument Data**

Becomes prominent to reinforce that the entire instrument is on even if no data is being currently being processed.

**5 Sound pin (engaged)**

Becomes upright and engaged after an instrument token is dropped in. Token can be removed by dragging it off the pin and releasing into any empty space on the screen.



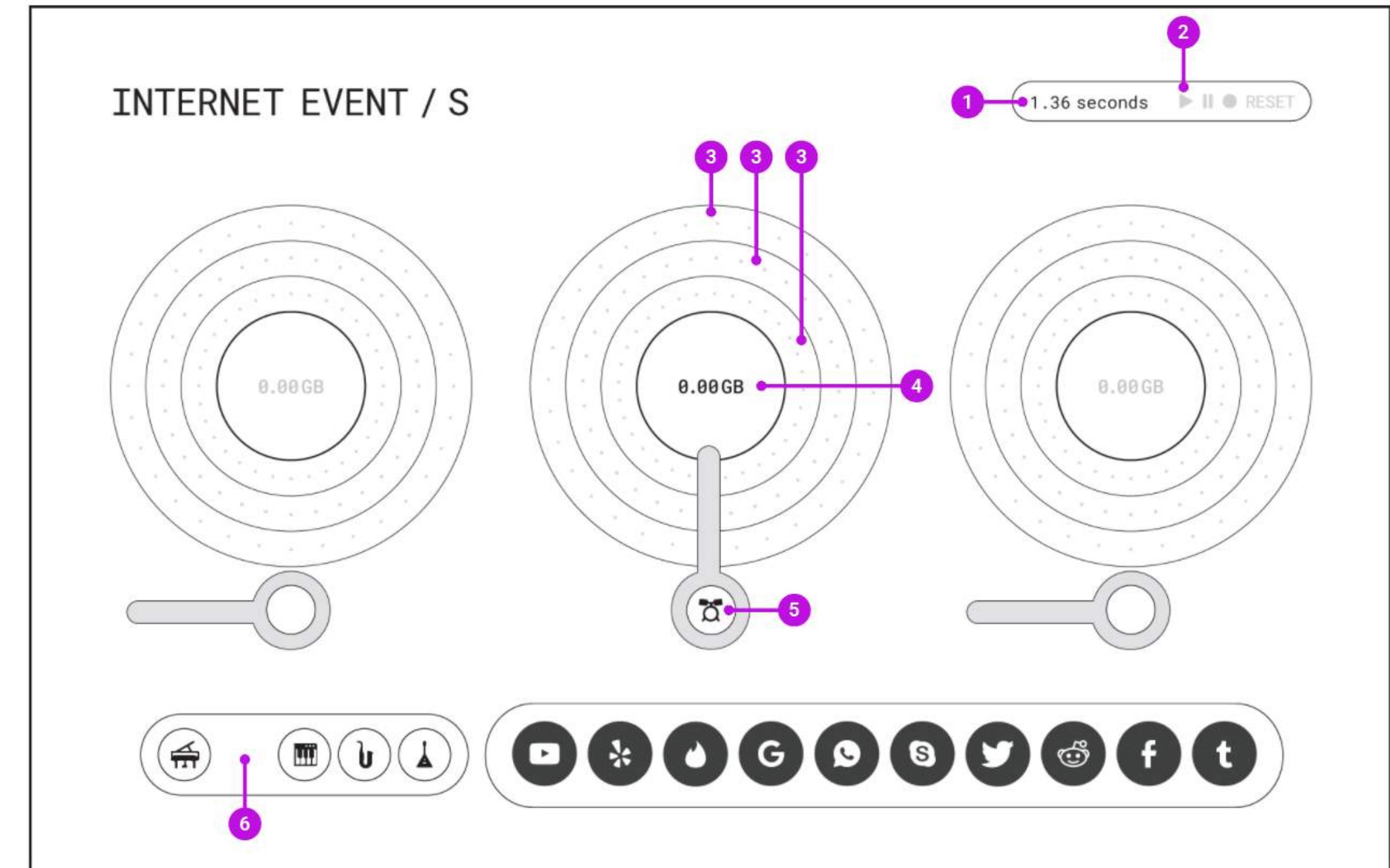
# Instrument Token Added wireframe

## 6 Sound pin (engaged)

Becomes upright and engaged after an instrument token is dropped in. Token can be removed by dragging it off the pin and releasing into any empty space on the screen.

## 7 Instrument tokens

Can only be used in once instrument at a time, therefore the spot remains empty when a token in being used.



# Event Token Added wireframe

## 1 Sound Ring (engaged)

After an internet events token has been placed into the ring, dots appear that correlate to how frequently that event happens per second. To remove an assigned internet even, drag it off the ring and drop in into any empty space.

## 2 Instrument data

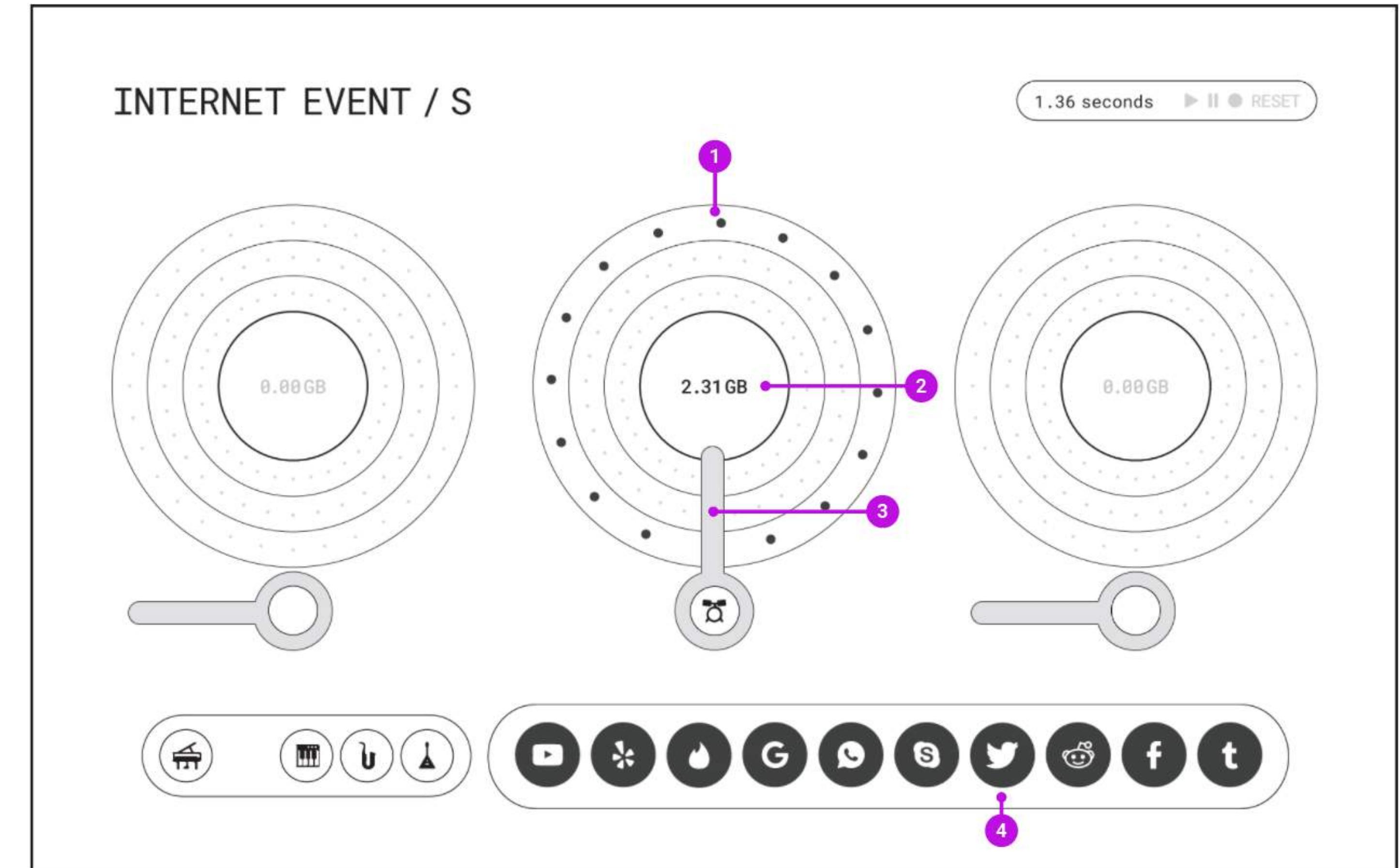
Updated to reflect the data being processed due to the internet event assigned to the sound ring.

## 3 Sound pin

As a dot passes under the pin, the sound of that ring is played.

## 4 Internet events tokens

Can be assigned multiple times, therefore a token reappears in the tray after it has been assigned to a sound ring.



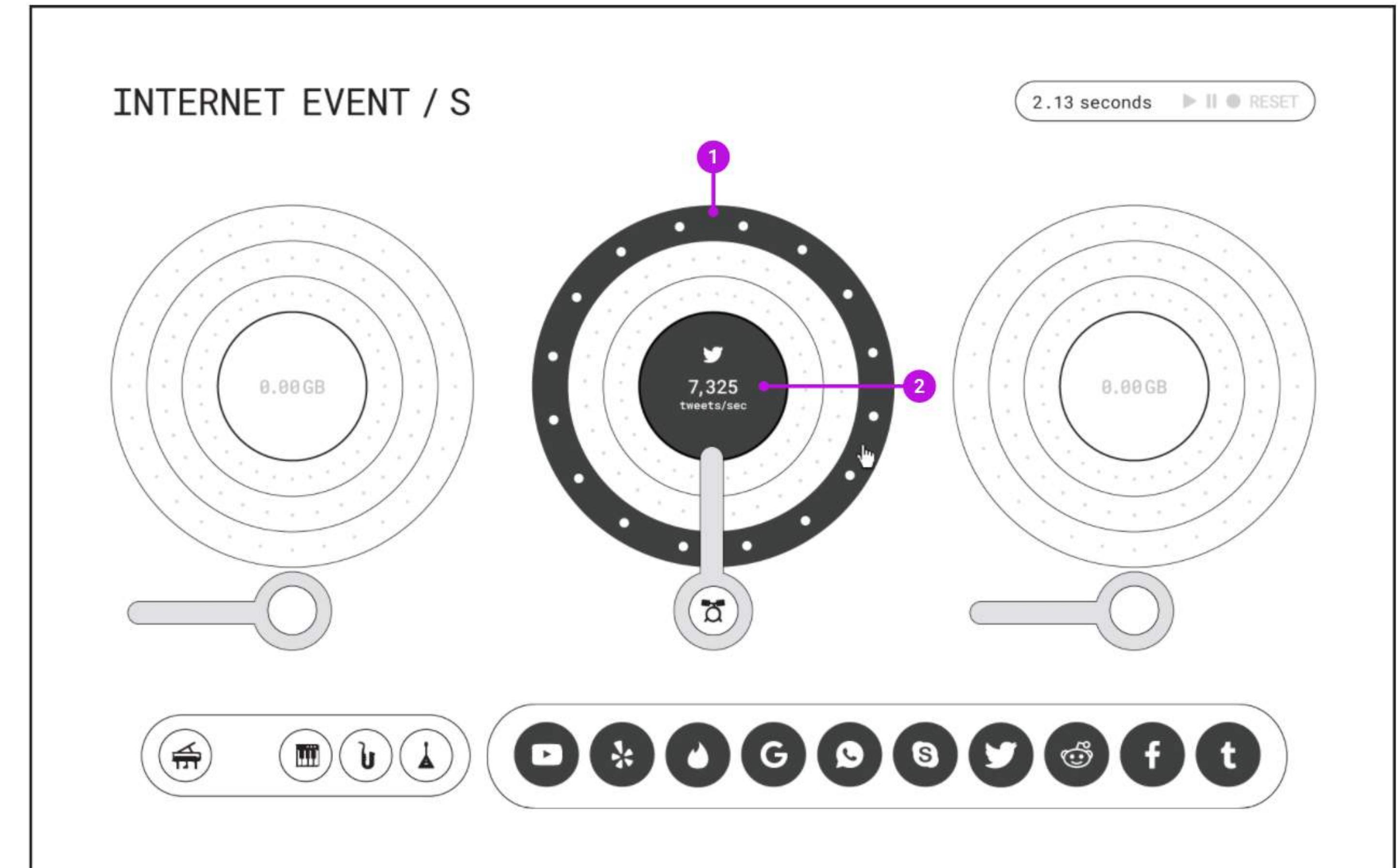
# Data Exploration wireframe

## 1 Sound Ring (engaged and hovered)

Hovering an engaged ring allows the user to see what information is being represented. The ring continues to rotate at the same pace and play sounds while it is hovered.

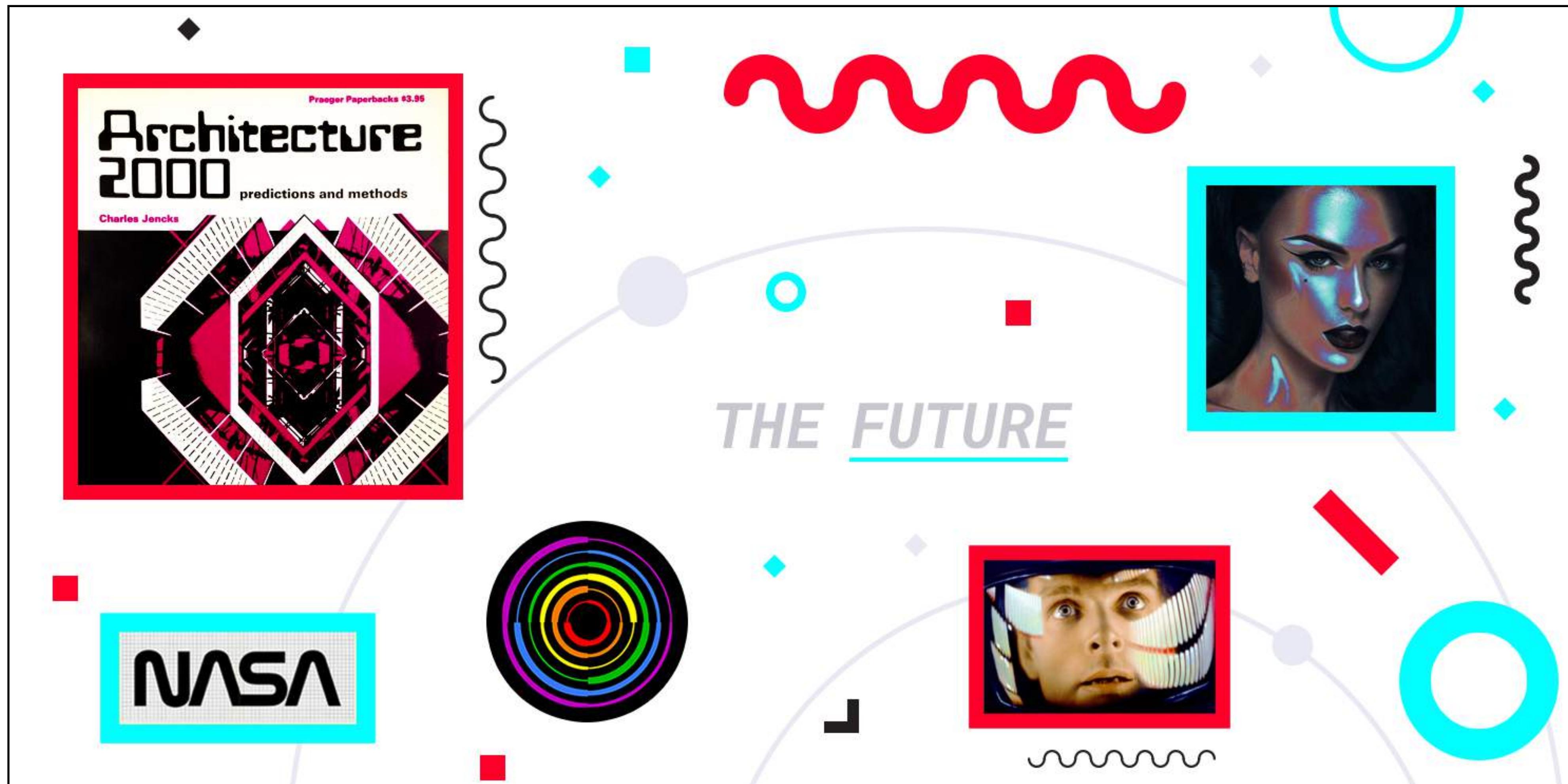
## 2 Instrument data

When a sound ring is hovered, the event and frequency being represented is displayed ..



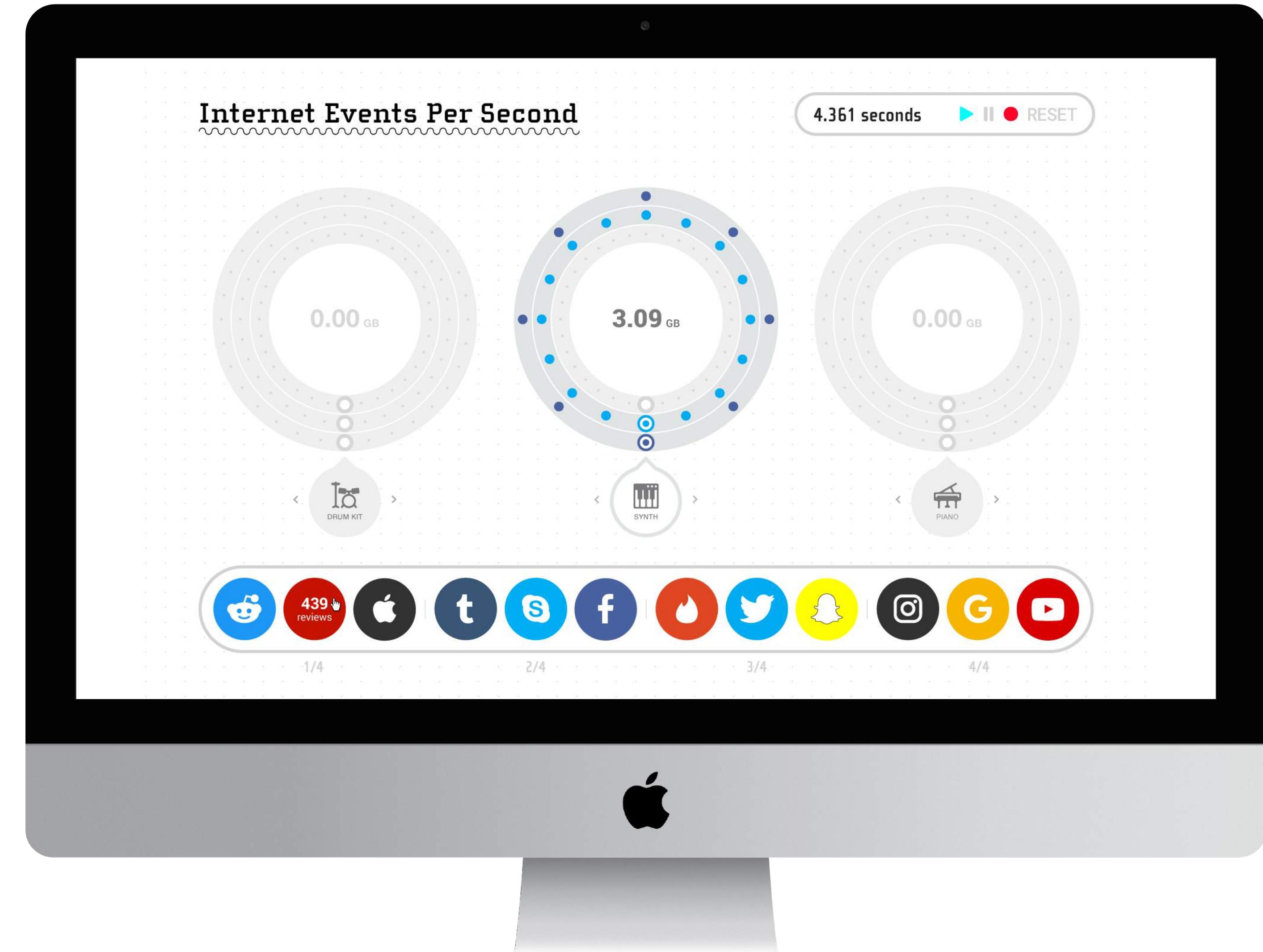
**VISUAL**

# Mood board



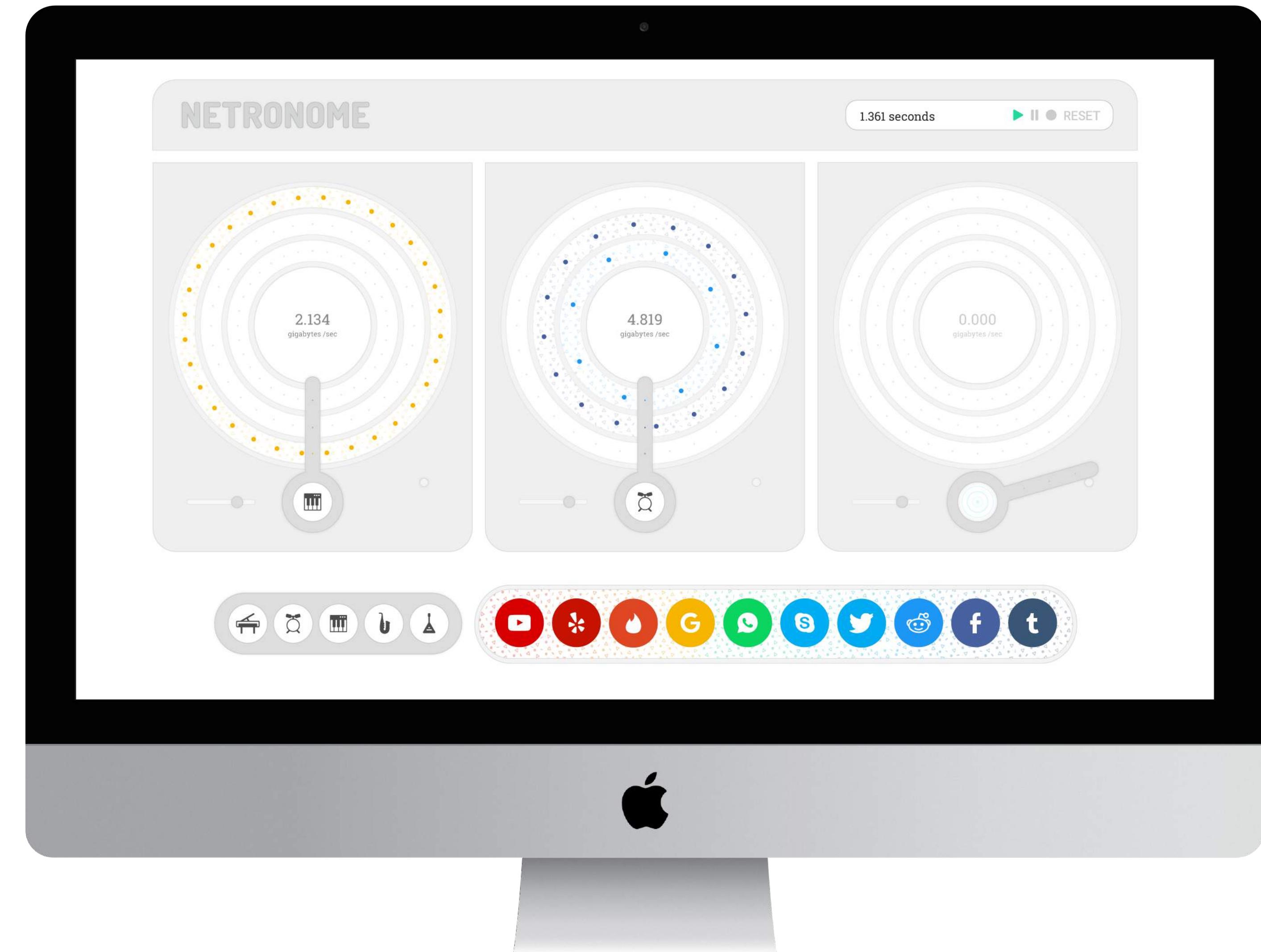
VISUAL 2

# Early direction



VISUAL 3

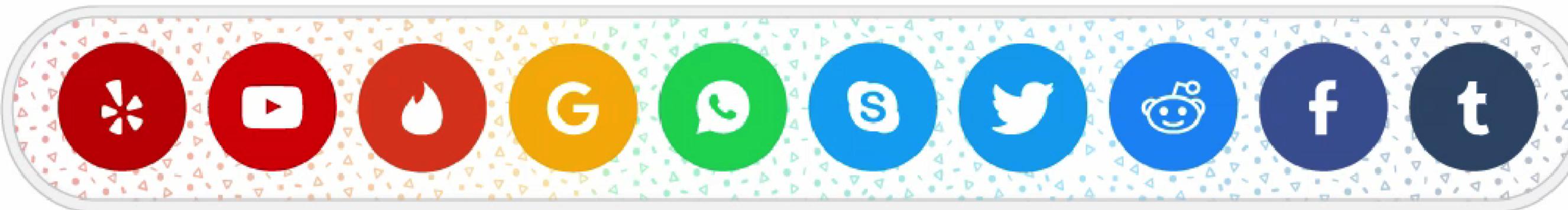
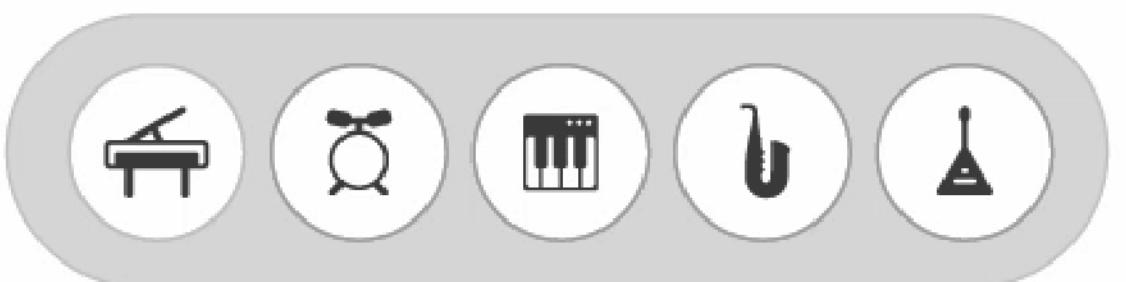
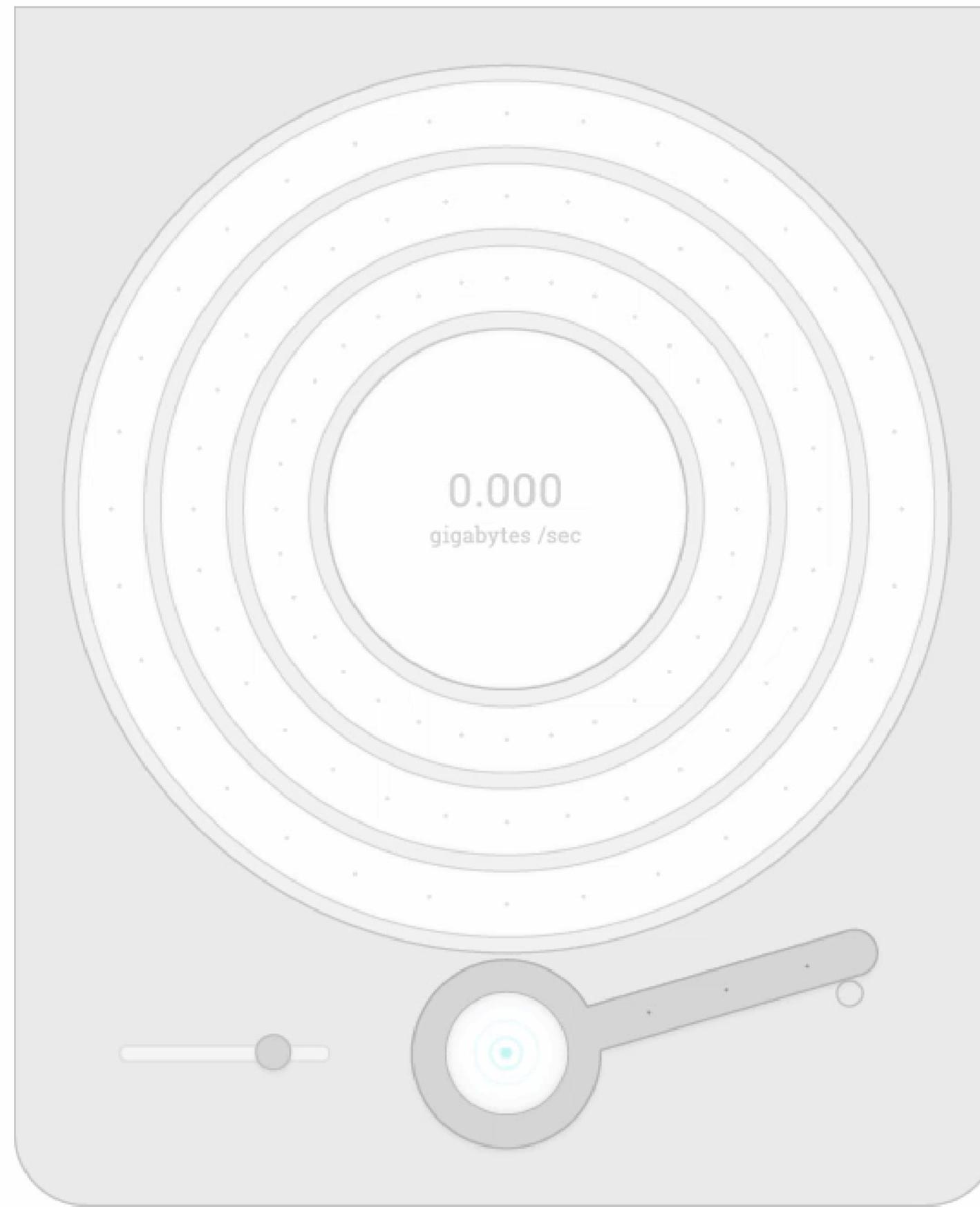
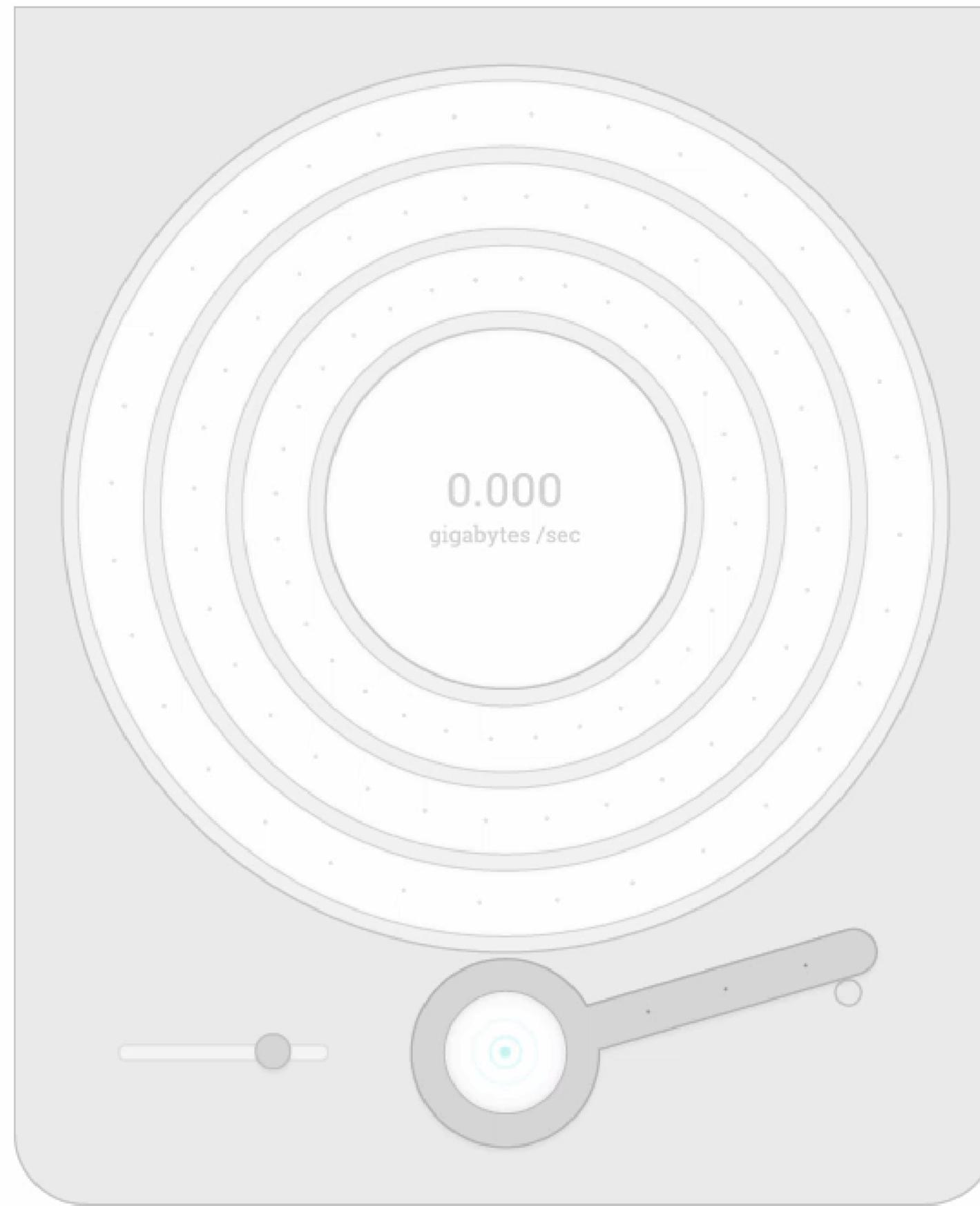
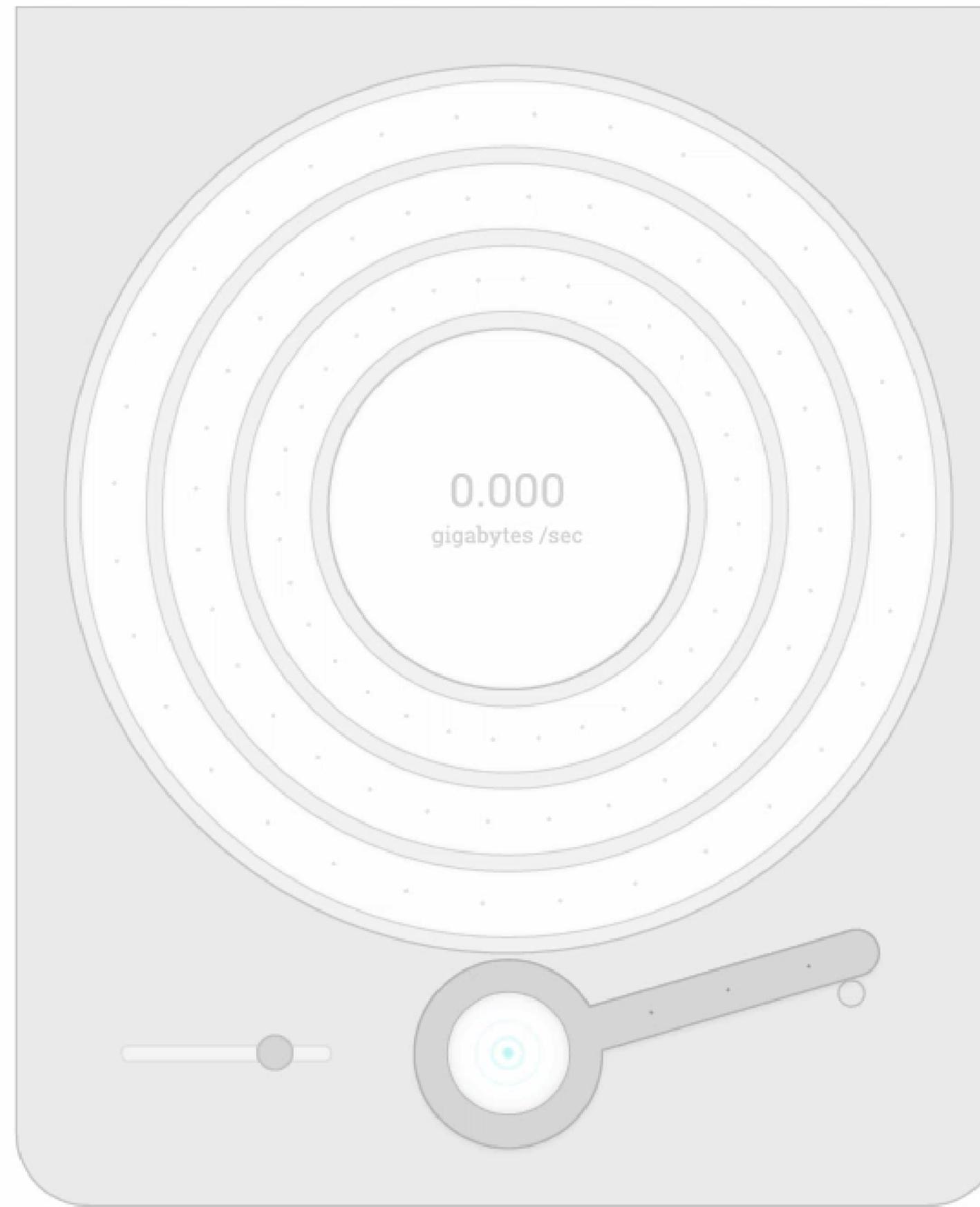
# Submitted comp



# NETRONOME

0.00 seconds

▶ II ● RESET



**END+THANKS**