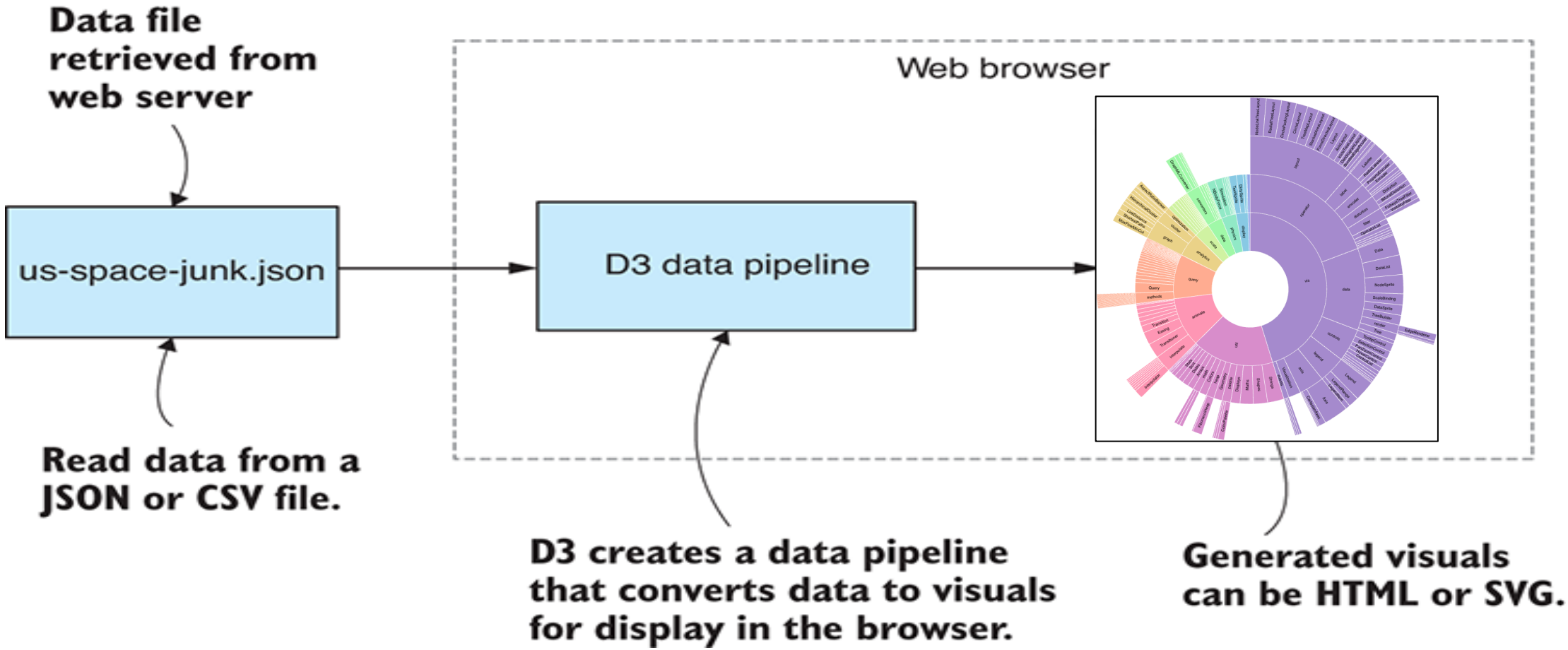
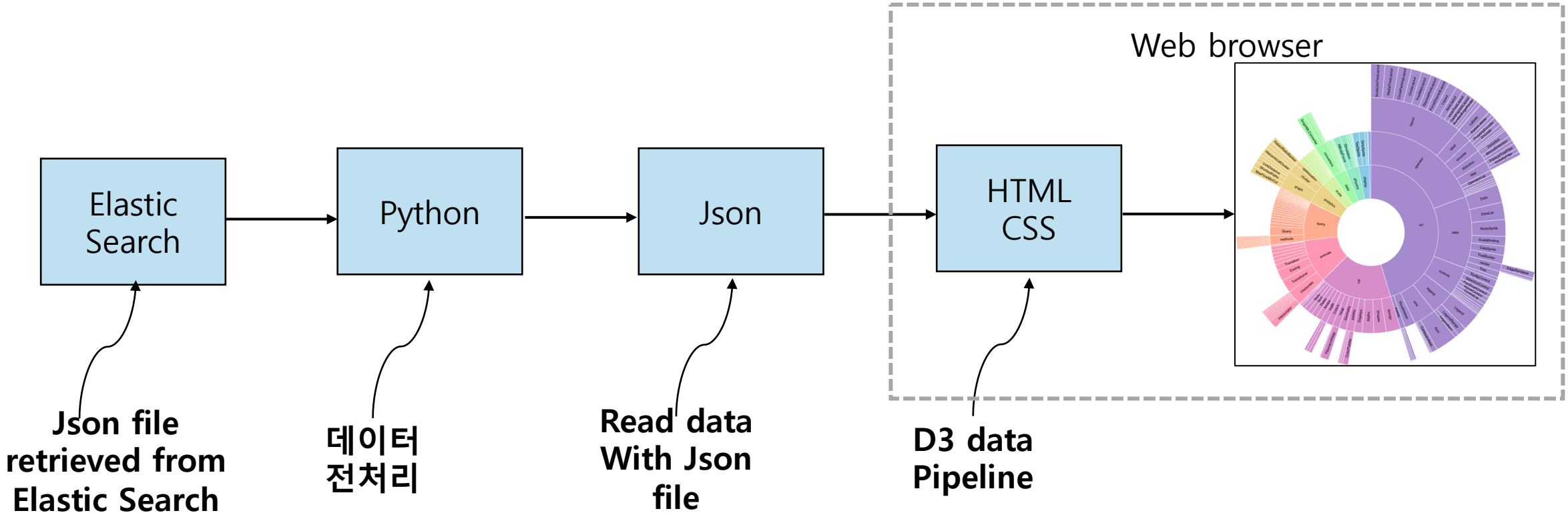
D3.js process



D3.js process



D3.js process

[20]

```
for index, value in keris_2.items():
    print(index, value)
    currentItem = False
    for key in index:
        print(key, currentItem)
        if not currentItem:
            currentItem = { "name": key, "children": []}
        else:
            currentList = list(filter(lambda item: item['name'] == key, currentItem["children"]))
            print('cl', currentList)
            if len(currentItem) == 0:
                currentItem["children"].append({ "name": key, "children": []})
        print(currentItem)
```

```
('Attack', 'TW', 65236) 2715
Attack False
{'name': 'Attack', 'children': []}
TW {'name': 'Attack', 'children': []}
cl []
{'name': 'Attack', 'children': []}
65236 {'name': 'Attack', 'children': [
cl []
{'name': 'Attack', 'children': []}
('DDos', 'TW', 65239) 1
DDos False
{'name': 'DDos', 'children': []}
TW {'name': 'DDos', 'children': []}
cl []
```

데이터
전처리
ElasticSearch→Python→Json

[16]

```
result=[]
for keys,value in keris_1.items():
    # print('IT', keys, value)
    result = addToGroup(result, keys[0], list(keys)[1:], value)
```

```
data []
currentKey Attack
nextKeys ['숙명여자대학교', 53043000]
value 5
f []
find index 0 {'name': 'Attack', 'children': []}
data []
currentKey 숙명여자대학교
nextKeys [53043000]
value 5
f []
find index 0 {'name': '숙명여자대학교', 'children': []}
data []
currentKey 53043000
```

D3.js process

```
keris_1=json_keris.groupby(['DRULE_ATT_TYPE_CODE1','INST_NM','GEAR_CODE']).size()
keris_1
```

```
DRULE_ATT_TYPE_CODE1  INST_NM  GEAR_CODE
Attack                숙명여자대학교  53043000        5
                                53043001        5
                                53073000     1942
                                53073040        1
                      홍익대학교     53043000        1
                                53073000      748
DDos                  숙명여자대학교  53073000        4
                      홍익대학교     53073000        1
Hack                  숙명여자대학교  53073000        7
Mail                  숙명여자대학교  53073000        1
Malwr                 숙명여자대학교  53073000        4
Web                   숙명여자대학교  53073000        1
dtype: int64
```

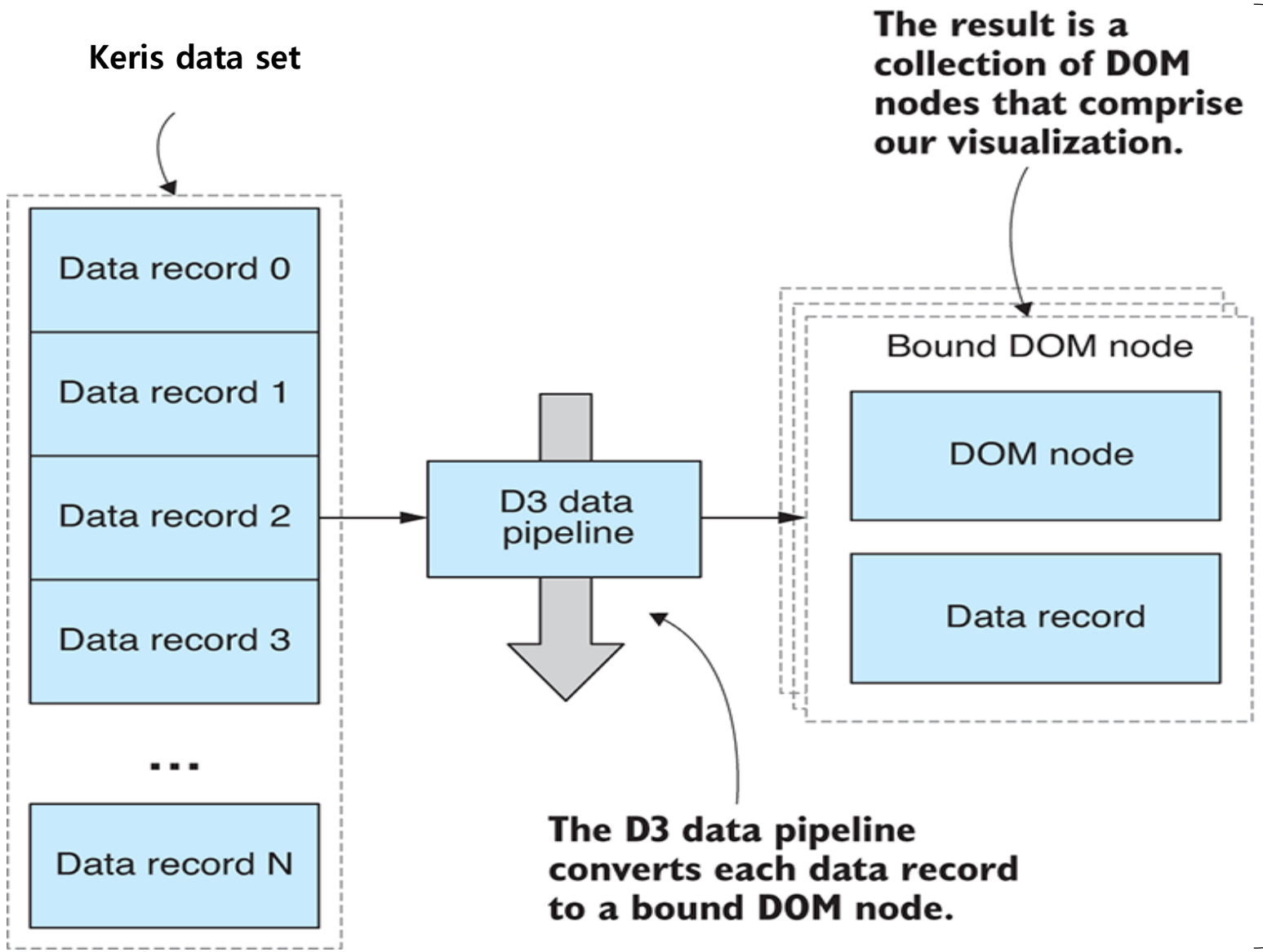
[15]

```
def addToGroup(data, currentKey, nextKeys, value):
    print('data', data)
    print('currentKey', currentKey)
    print('nextKeys', nextKeys)
    print('value', value)

    if len(nextKeys) == 0:
        # we are in the value children, no need to calculate more
        data.append({ "name": currentKey, "value": value })
        print('append', data)
    else:
        # it has to find existing element with name, or add new one
        filterList = list(filter(lambda item: item['name'] == currentKey, data))
        print('f', filterList)
        if len(filterList) == 0:
            data.append({ "name": currentKey, "children": [] })
            currentKeyIndex = len(data) - 1
        else:
            currentKeyIndex = data.index(filterList[0])

        # with index of item with name, add the rest of the item to children
        currentKeyValue = data[currentKeyIndex]
        print('find index', currentKeyIndex, currentKeyValue)
        currentKeyValue["children"] = addToGroup(currentKeyValue["children"], nextKeys[0], nextKeys[1:], value)
        data[currentKeyIndex] = currentKeyValue

    return data
```


```
[
    {
        "name": "Attack",
        "children": [
            {
                "name": "숙명여자대학교",
                "children": [
                    {
                        "name": 53073000,
                        "value": 1931
                    },
                    {
                        "name": 53073001,
                        "value": 6
                    },
                    {
                        "name": 53073002,
                        "value": 1
                    }
                ]
            },
            {
                "name": "홍익대학교",
                "children": []
            }
        ]
```


Json data that describes each object

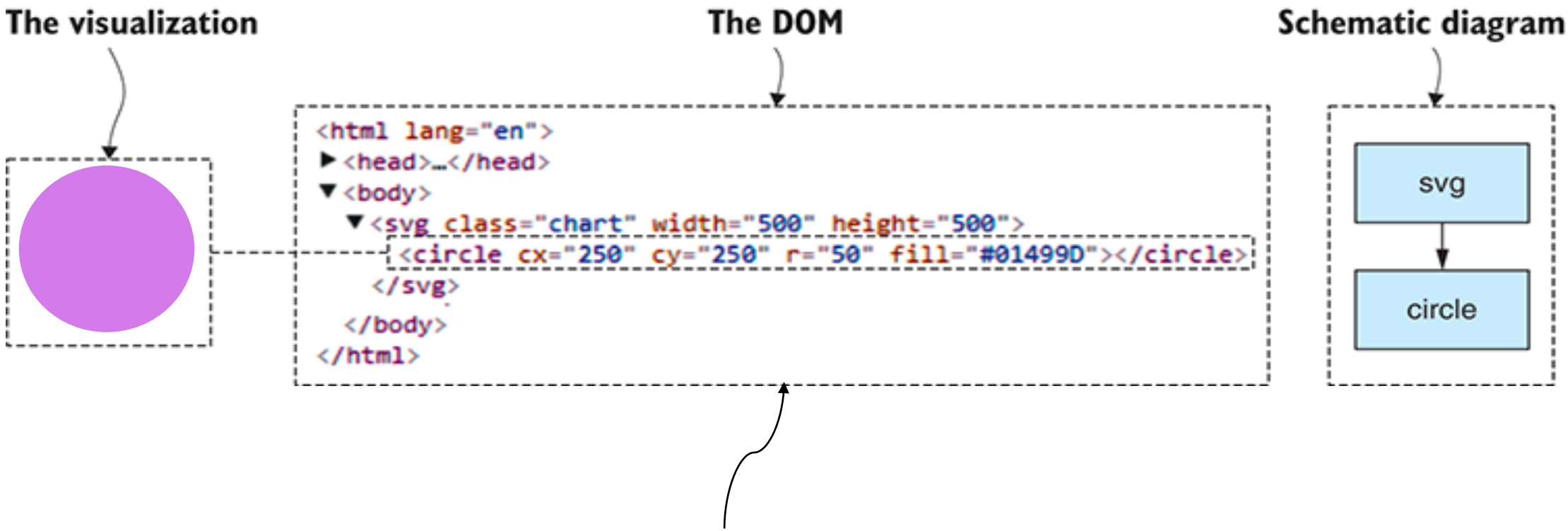
D3.js process



D3 produces “bound DOM nodes” for each data record.

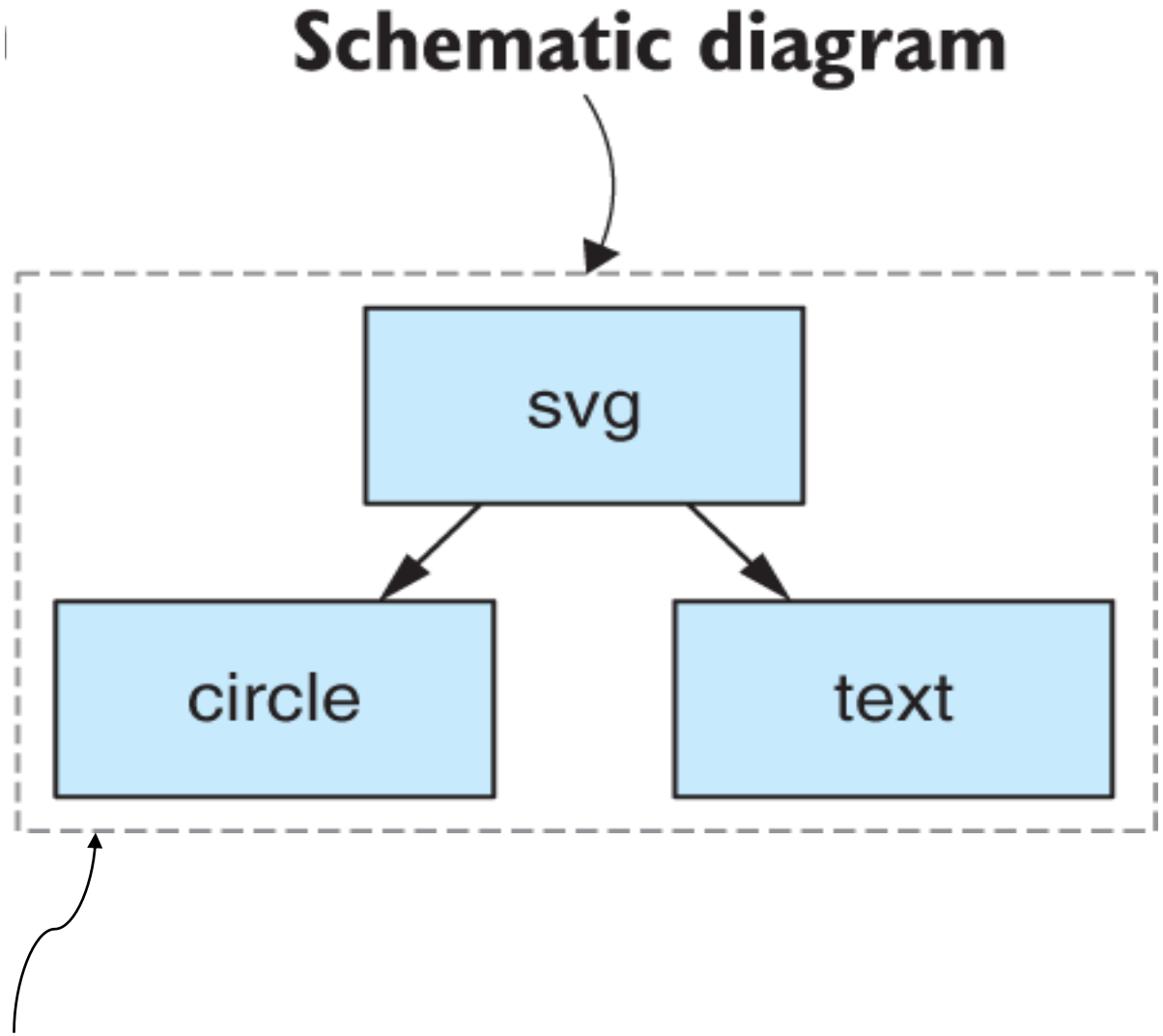
Dom is Document Object Model which is a cross-platform and language-independent interface

D3.js process

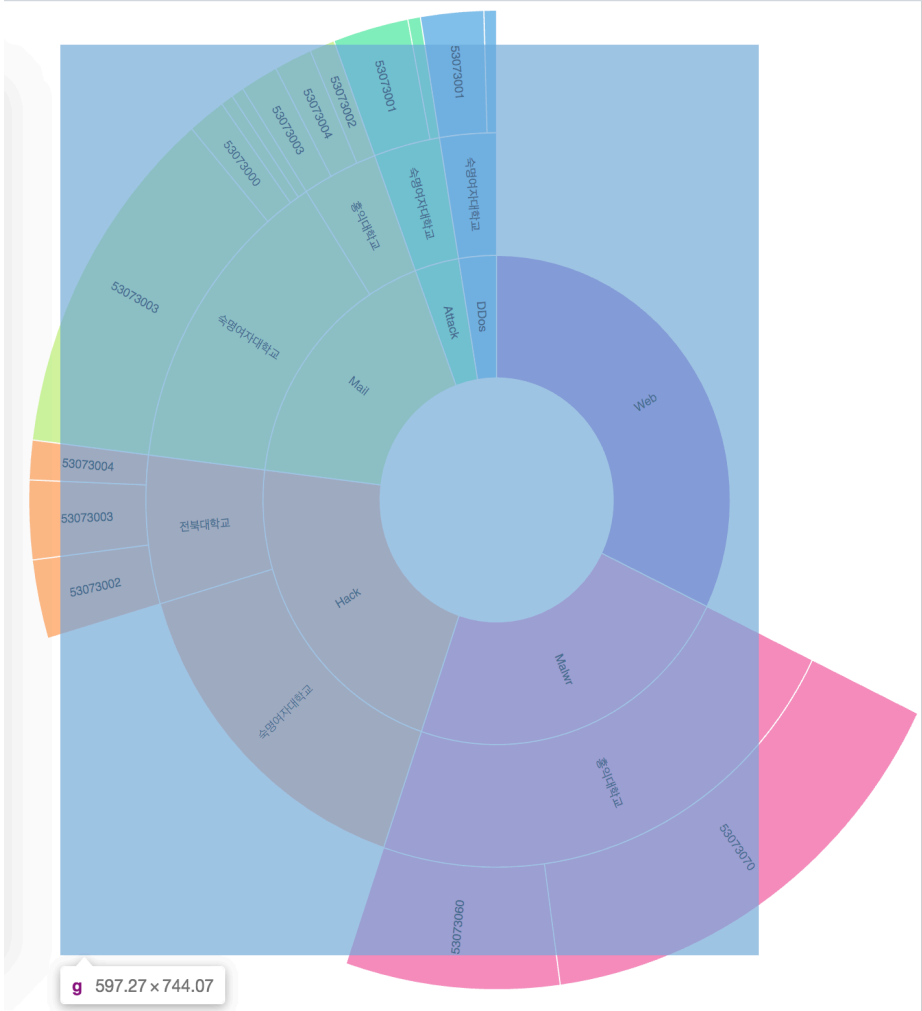


An SVG circle element manually added to SVG (with DOM viewed in Chrome DevTools)

D3.js process



Adding a circle element to the SVG and styling



Elements Console Sources Network Performance Memory Application Security Lighthouse

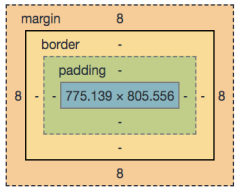
```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
…▼<body> == $0
  ▶<script type="text/javascript">…</script>
    <!-- Code injected by live-server -->
  ▶<script type="text/javascript">…</script>
  ▼<svg width="800" height="800" viewBox="-399,-398.9996948242188,797.9996948242188">
    ▶<g fill-opacity="0.6">…</g>
    ▼<g pointer-events="none" text-anchor="middle" font-size="10" font-family="sans-serif">
        <text transform="rotate(-32.08695652173913) translate(150,0) rotate(0)" dy="0.35em">Web</text>
        <text transform="rotate(67.30434782608694) translate(150,0) rotate(0)" dy="0.35em">Malwr</text>
        <text transform="rotate(147.91304347826087) translate(150,0) rotate(180)" dy="0.35em">Hack</text>
        <text transform="rotate(218.34782608695656) translate(150,0) rotate(180)" dy="0.35em">Mail</text>
        <text transform="rotate(255.1304347826087) translate(150,0) rotate(180)" dy="0.35em">Attack</text>
        <text transform="rotate(265.30434782608694) translate(150,0) rotate(180)" dy="0.35em">DDos</text>
        <text transform="rotate(67.30434782608694) translate(250,0) rotate(0)" dy="0.35em">홍익대학교</text>
        <text transform="rotate(136.17391304347825) translate(250,0) rotate(180)" dy="0.35em">숙명여자대학교</text>
        <text transform="rotate(175.30434782608694) translate(250,0) rotate(180)" dy="0.35em">전북대학교</text>
        <text transform="rotate(212.08695652173918) translate(250,0) rotate(180)" dy="0.35em">숙명여자대학교</text>
        <text transform="rotate(243.39130434782612) translate(250,0) rotate(180)" dy="0.35em">홍익대학교</text>
        <text transform="rotate(255.1304347826087) translate(250,0) rotate(180)" dy="0.35em">숙명여자대학교</text>
        <text transform="rotate(265.30434782608694) translate(250,0) rotate(180)" dy="0.35em">숙명여자대학교</text>
        <text transform="rotate(54) translate(350,0) rotate(0)" dy="0.35em">53073070</text>
        <text transform="rotate(95.47826086956522) translate(350,0) rotate(180)" dy="0.35em">53073060</text>
        <text transform="rotate(168.26086956521743) translate(350,0) rotate(180)" dy="0.35em">53073002</text>
        <text transform="rotate(177.65217391304355) translate(350,0) rotate(180)" dy="0.35em">53073003</text>
        <text transform="rotate(184.69565217391312) translate(350,0) rotate(180)" dy="0.35em">53073004</text>
        <text transform="rotate(208.1739130434783) translate(350,0) rotate(180)" dy="0.35em">53073003</text>
        <text transform="rotate(231.65217391304355) translate(350,0) rotate(180)" dy="0.35em">53073000</text>
        <text transform="rotate(239.47826086956525) translate(350,0) rotate(180)" dy="0.35em">53073003</text>
        <text transform="rotate(244.1739130434783) translate(350,0) rotate(180)" dy="0.35em">53073004</text>
        <text transform="rotate(248.08695652173924) translate(350,0) rotate(180)
```

html body

Styles Computed Layout Event Listeners »

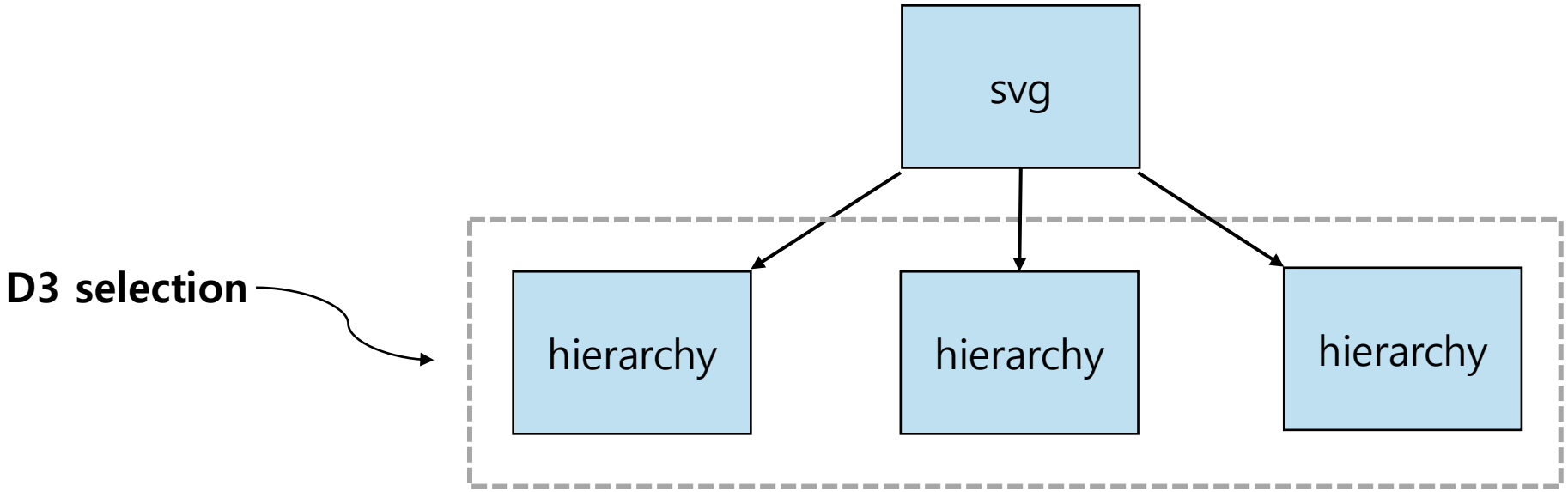
Filter :hov .cls +

```
element.style {
}
body {                                  user agent stylesheet
  display: block;
  margin: ▶ 8px;
}
```

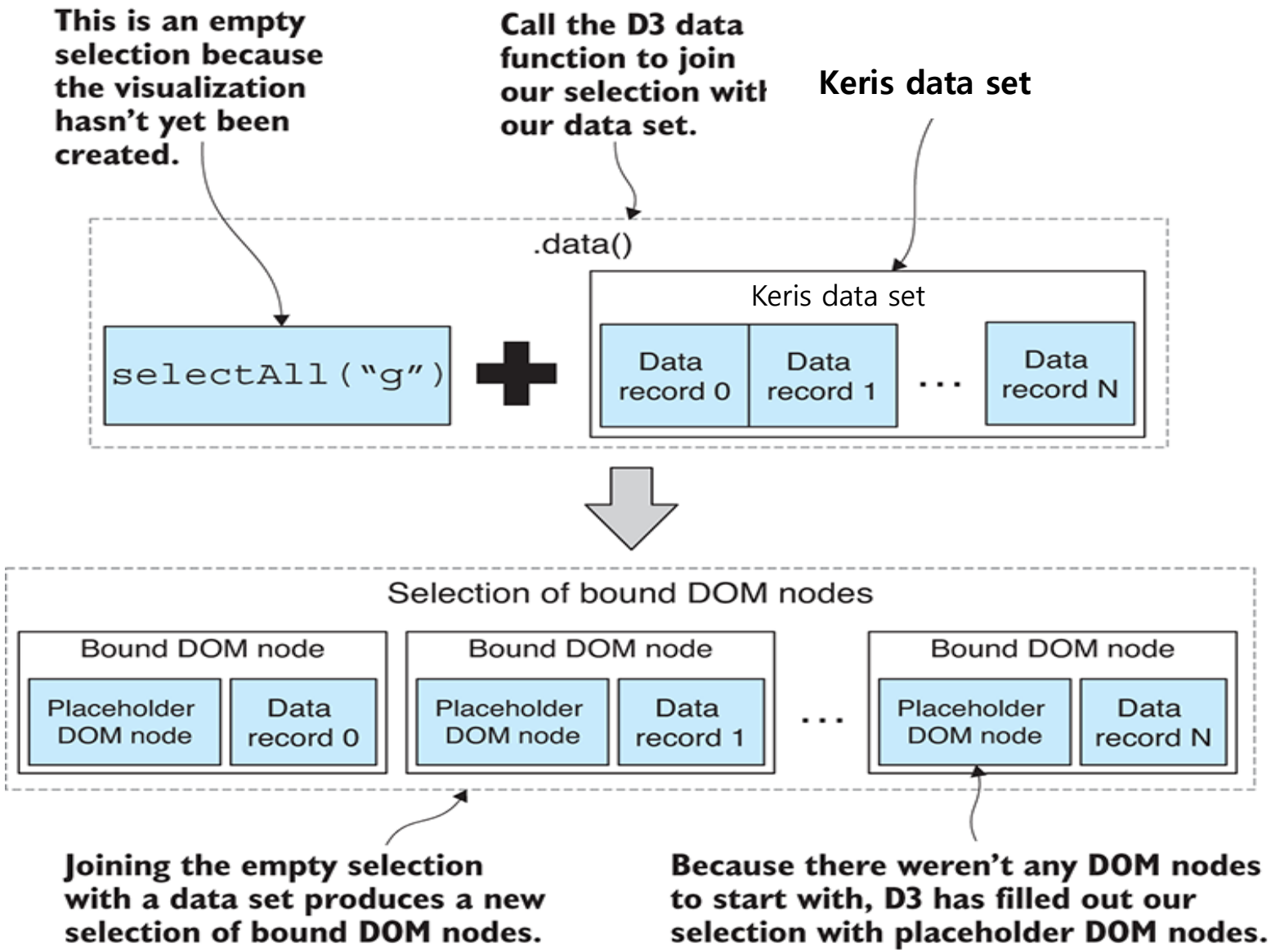


The DOM

D3.js process



D3.js process



```
const partition = data => d3.partition()
  .size([2 * Math.PI, radius])
  (d3.hierarchy(data)
    .sum(d => d.value)
    .sort((a, b) => b.value - a.value))

const arc = d3.arc()
  .startAngle(d => d.x0)
  .endAngle(d => d.x1)
  .padAngle(d => Math.min((d.x1 - d.x0) / 2, 0.005))
  .padRadius(radius / 2)
  .innerRadius(d => d.y0)
  .outerRadius(d => d.y1 - 1)

const root = partition(data);

const svg = d3.select("body")
  .append("svg")
  .attr("width", width)
  .attr("height", width);

svg.append("g")
  .attr("fill-opacity", 0.6)
  .selectAll("path")
  .data(root.descendants().filter(d => d.depth))
  .join("path")
  .attr("fill", d => { while (d.depth > 1) d = d.parent; return color(d.data.name); })
  .attr("d", arc)
  .append("title")
  .text(d => `${d.ancestors().map(d => d.data.name).reverse().join("/")}\n${format(d.value)}`);

svg.append("g")
  .attr("pointer-events", "none")
  .attr("text-anchor", "middle")
  .attr("font-size", 10)
  .attr("font-family", "sans-serif")
  .selectAll("text")
  .data(root.descendants().filter(d => d.depth && (d.y0 + d.y1) / 2 * (d.x1 - d.x0) > 10))
  .join("text")
  .attr("transform", function (d) {
    const x = (d.x0 + d.x1) / 2 * 180 / Math.PI;
    const y = (d.y0 + d.y1) / 2;
    return `rotate(${x - 90}) translate(${y},0) rotate(${x < 180 ? 0 : 180})`;
```

D3.js-Chord Diagram

- It shows relationships among a group of entities
- For example, consider a hypothetical population of people wi th different hair colors: black, blonde, brown and red. Each pe rson in this population has a preferred hair color for a dating partner
- A chord diagram visualizes these relationships
- https://github.com/zziuni/d3/wiki/Chord-Layout
- https://www.visualcinnamon.com/2016/06/orientation-gradien t-d3-chord-diagram

D3.js-Sunburst

- Radial space-filling visualization
- It shows the cumulative values of subtrees
- It is commonly used to visualize file systems which the size of files within nested folders
- https://observablehq.com/@d3/sunburst

D3.js-Zoomable Circle Packing

- It is the arrangement of circles inside some demarcation so th at none of the circles overlap
- It intuitively observe clusters of similar concepts based on ho w tightly the circles are packed
- It also displays hierarchy where you can get smaller clusters o f circles packed within a bigger circle which itself is arranged next to or within other circles
- https://observablehq.com/@d3/zoomable-circle-packing