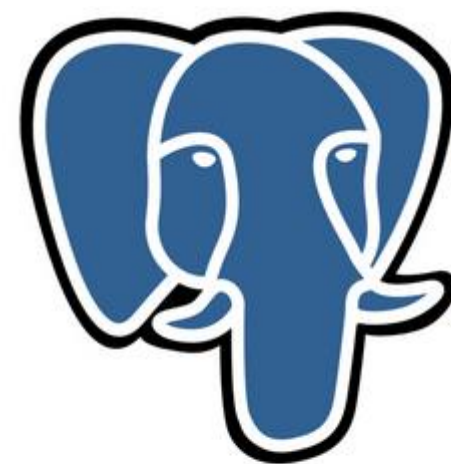


German Election System

Maïsa Ben Salah & Yecine Megdiche
07.02.2021

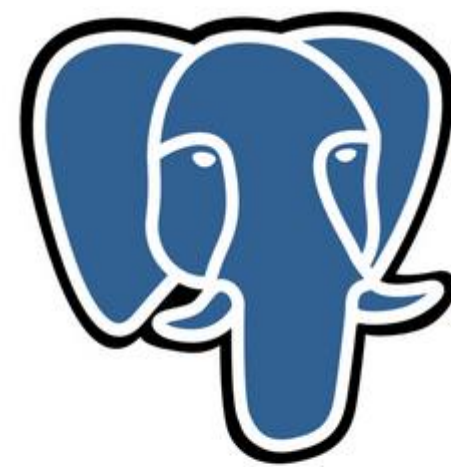
ARCHITECTURE OVERVIEW

ARCHITECTURE OVERVIEW

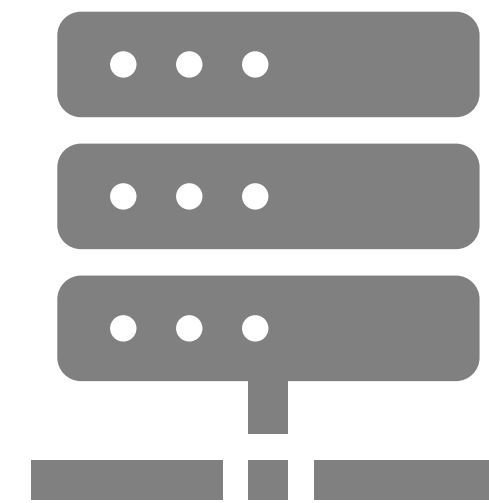


PostgreSQL

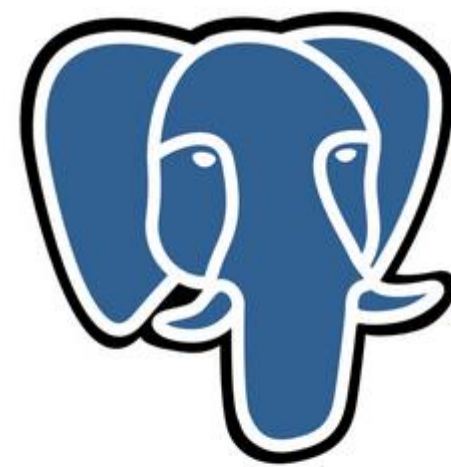
ARCHITECTURE OVERVIEW



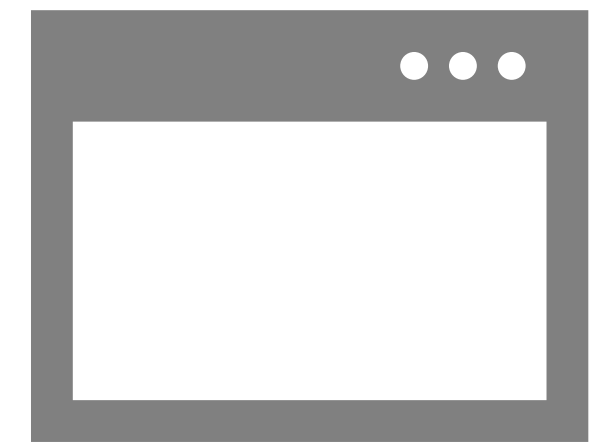
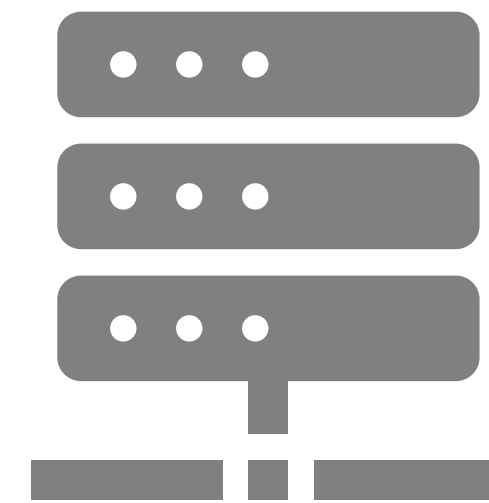
PostgreSQL



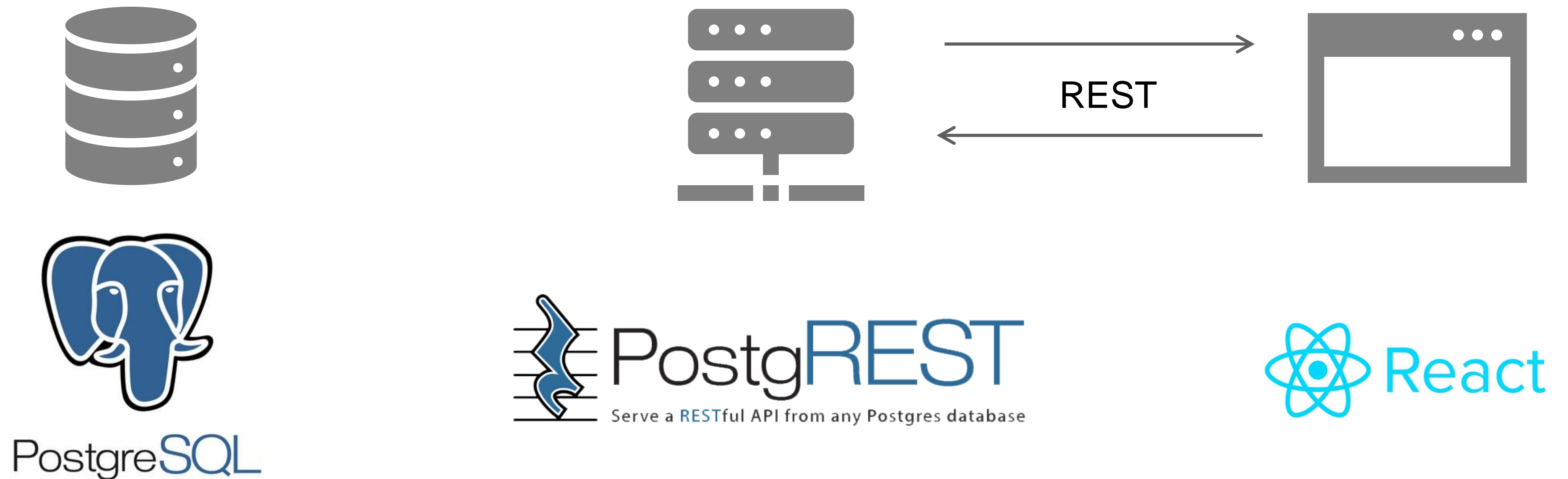
ARCHITECTURE OVERVIEW



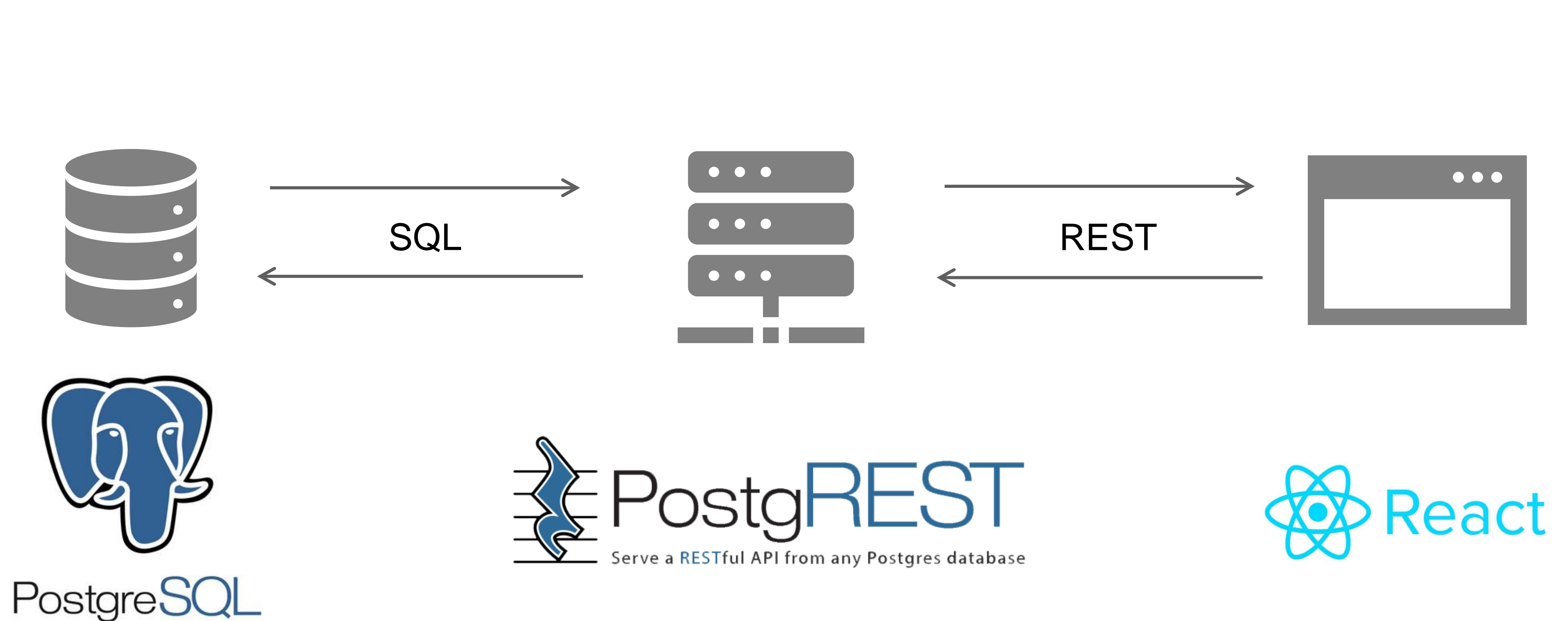
PostgreSQL



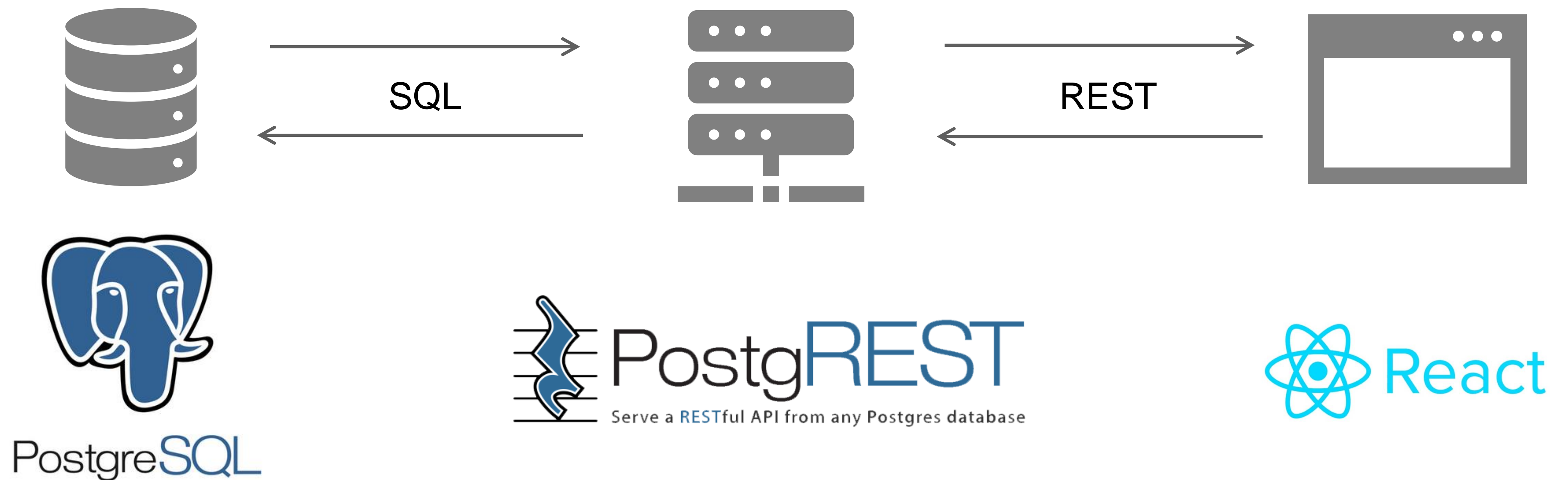
ARCHITECTURE OVERVIEW



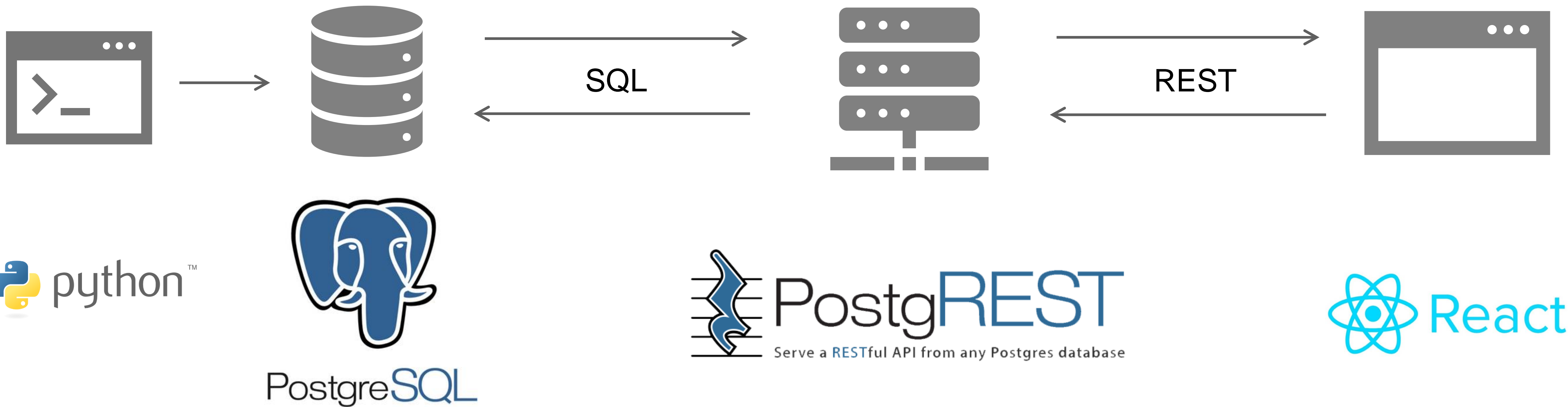
ARCHITECTURE OVERVIEW



ARCHITECTURE OVERVIEW

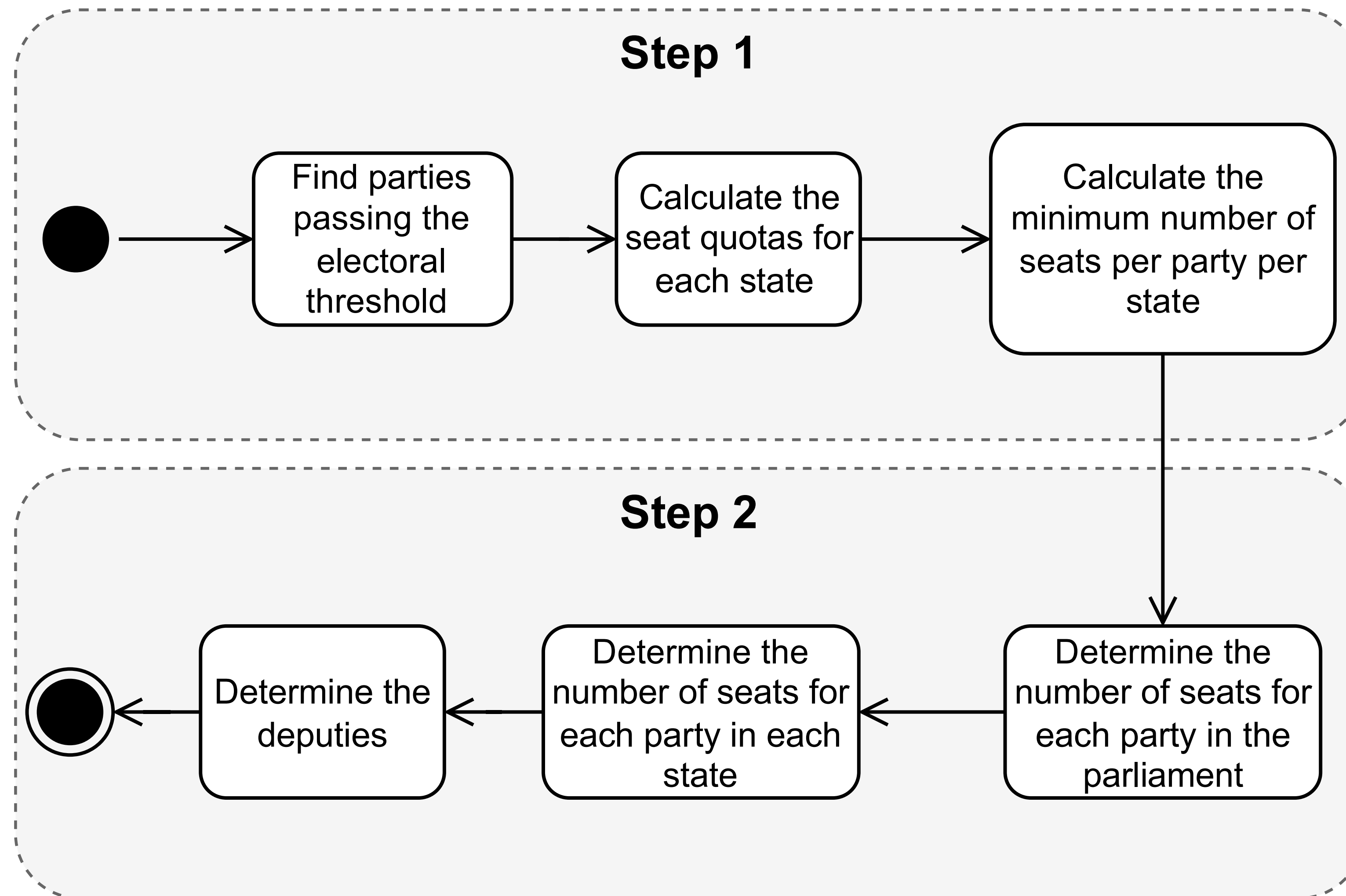


ARCHITECTURE OVERVIEW

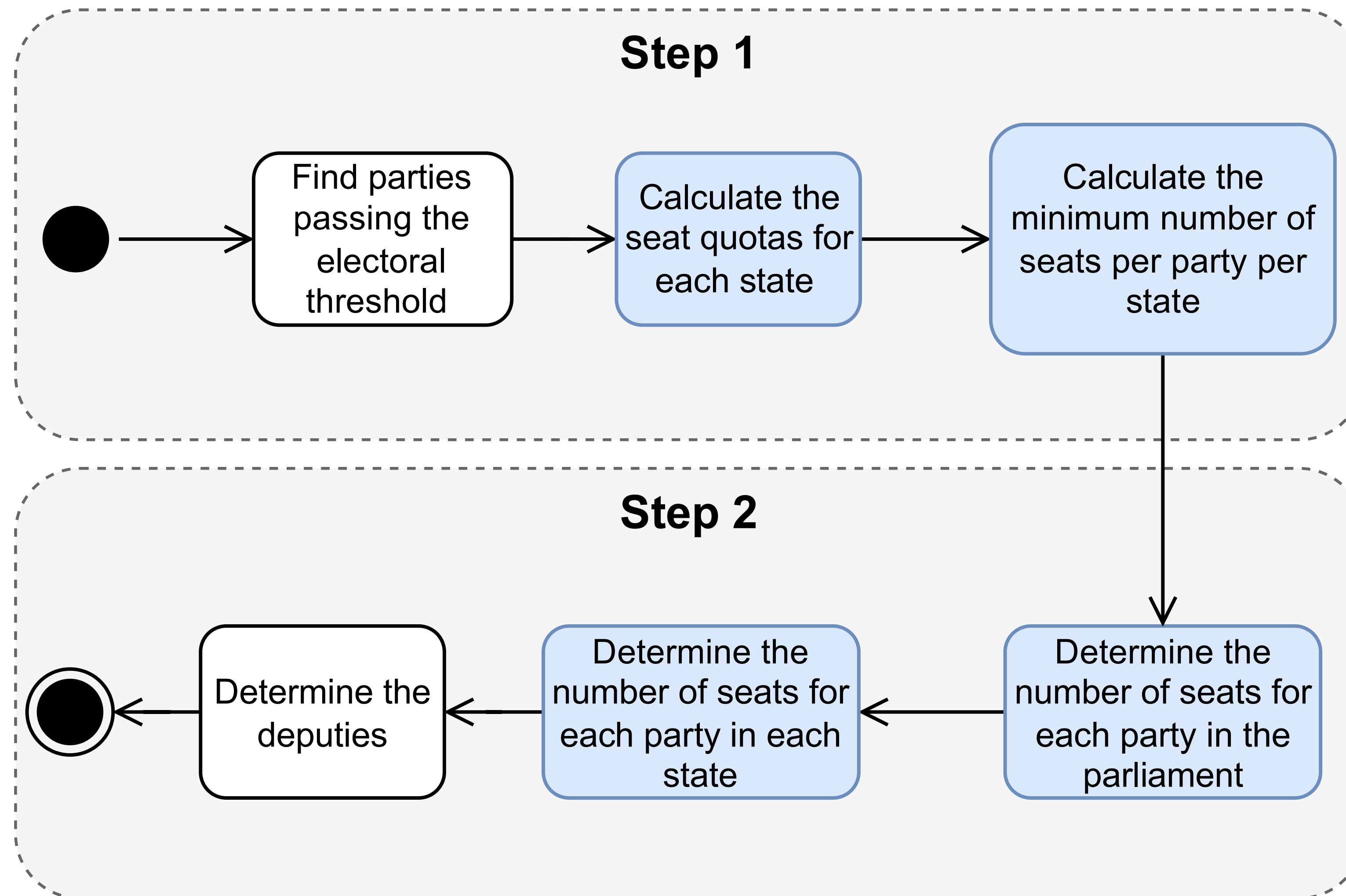


DEMO

Seat Distribution



Seat Distribution



Seat Distribution

Sample Sainte-Laguë Implementation

Task: Distribute n seats over parties.

Algorithm: Highest averages method
(Höchstzahlverfahren):

Seat Distribution

Sample Sainte-Laguë Implementation

Task: Distribute n seats over parties.

Algorithm: Highest averages method
(Höchstzahlverfahren):

1. Divide the votes for each party by
 $0,5; 1,5; \dots; n - 0,5$

```
WITH h AS (  
    SELECT party,  
           votes / (s.a - 0.5) AS ratio  
    FROM total,  
         generate_series(1, n) AS s(a)  
)
```

Seat Distribution

Sample Sainte-Laguë Implementation

Task: Distribute n seats over parties.

Algorithm: Highest averages method (Höchstzahlverfahren):

1. Divide the votes for each party by
 $0,5; 1,5; \dots; n - 0,5$
2. Sort the previous results and give
seats to the first n ratios.

```
WITH h AS (  
    SELECT party,  
           votes / (s.a - 0.5) AS ratio  
    FROM total,  
         generate_series(1, n) AS s(a)  
    ORDER BY ratio DESC  
    LIMIT n  
)
```

Seat Distribution

Sample Sainte-Laguë Implementation

Task: Distribute n seats over parties.

Algorithm: Highest averages method (Höchstzahlverfahren):

1. Divide the votes for each party by $0,5; 1,5; \dots; n - 0,5$
2. Sort the previous results and give seats to the first n ratios.
3. Count the number of seats per party

```
WITH h AS (  
    SELECT party,  
           votes / (s.a - 0.5) AS ratio  
    FROM total,  
         generate_series(1, n) AS s(a)  
    ORDER BY ratio DESC  
    LIMIT n  
)  
SELECT party, COUNT(*) as total_seats  
FROM h  
GROUP BY party
```


Benchmark



Test Number	Number of users	Wait-time (seconds)	Number of Requests	Requests per second	Average response time (ms)
0 (Baseline)	1	1	452	1.51	18
1	100	5	8950	29.86	27
2	100	1	43815	146.13	17
3	500	5	44619	148.82	41
4	500	1	51515	171.84	2221

All tests ran a total of five minutes (CPU: Intel i7-8750H (12) @ 4.100GHz, RAM: 12GB), OS: Manjaro Linux x86_64, 4.19.217-1-MANJARO).

**Thank you for your
attention**