



# Building dev tools at the right level of abstraction

**Ben Davis** CTO

@BenCDavis · ben@gatherdata.co

The data engineering industry is **very fragmented**.

Gather is a **data integration tool for developers**. It makes it really easy to build integration pipelines that push and pull data from various SaaS APIs.

Gather is a **data integration tool for developers**. It makes it really easy to build integration pipelines that push and pull data from various SaaS APIs.



But how did we get here?

Initial motivating problem: **building data pipelines is hard.** We should fix it.

Data pipelining is conceptual. It breaks down into **many use-cases**.

Data pipelining is conceptual. It breaks down into **many use-cases**.

**Batch**

**Streaming**

**ETL**

**...**



People will ask **“Can I use it for this?”** or **“Oh I can you like this right?”**

No **framework** to answer those questions

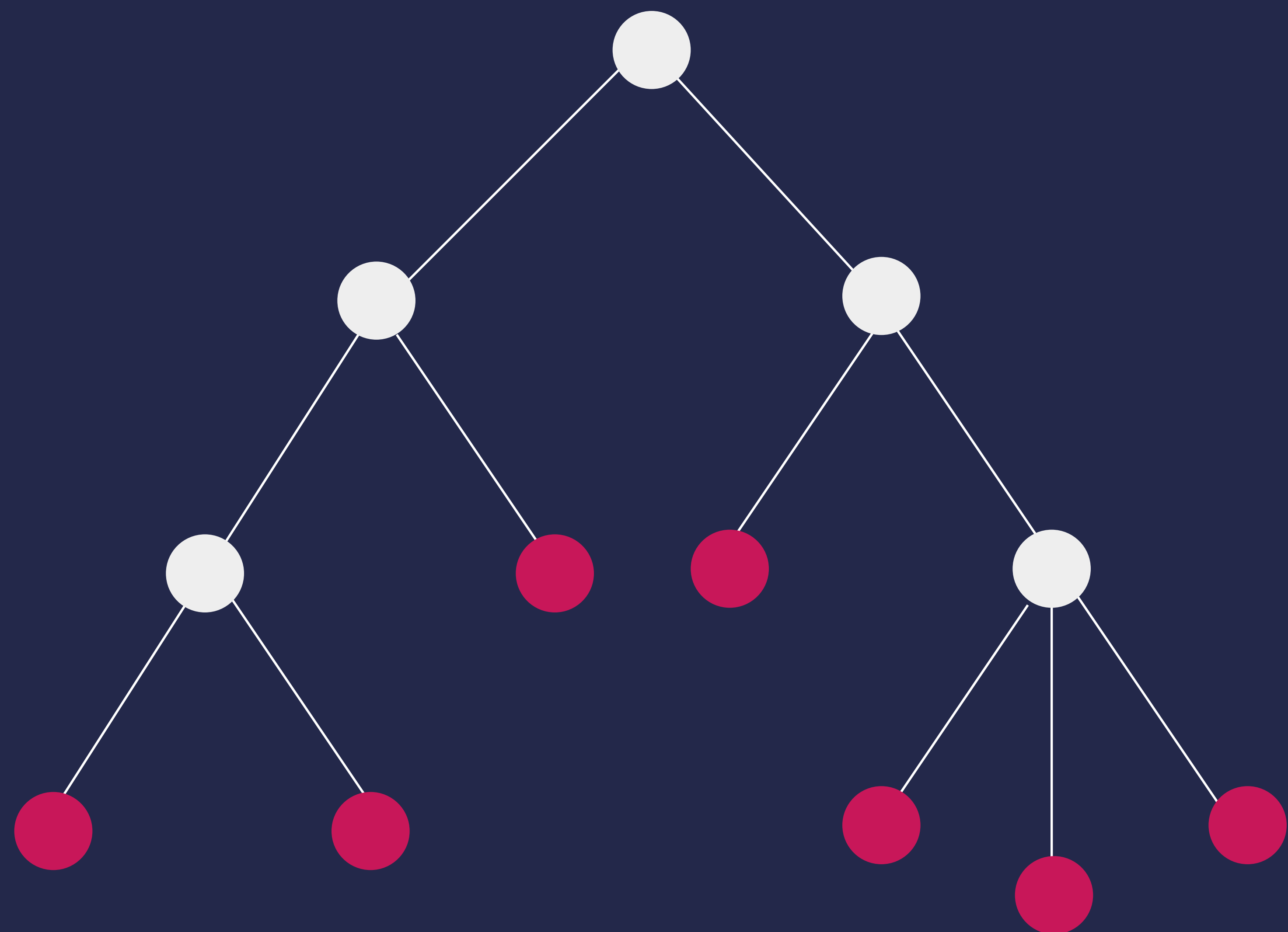
“Abstraction is **amplification of the essential** and  
**elimination of the irrelevant.**”

*–K.K Aggarwal*

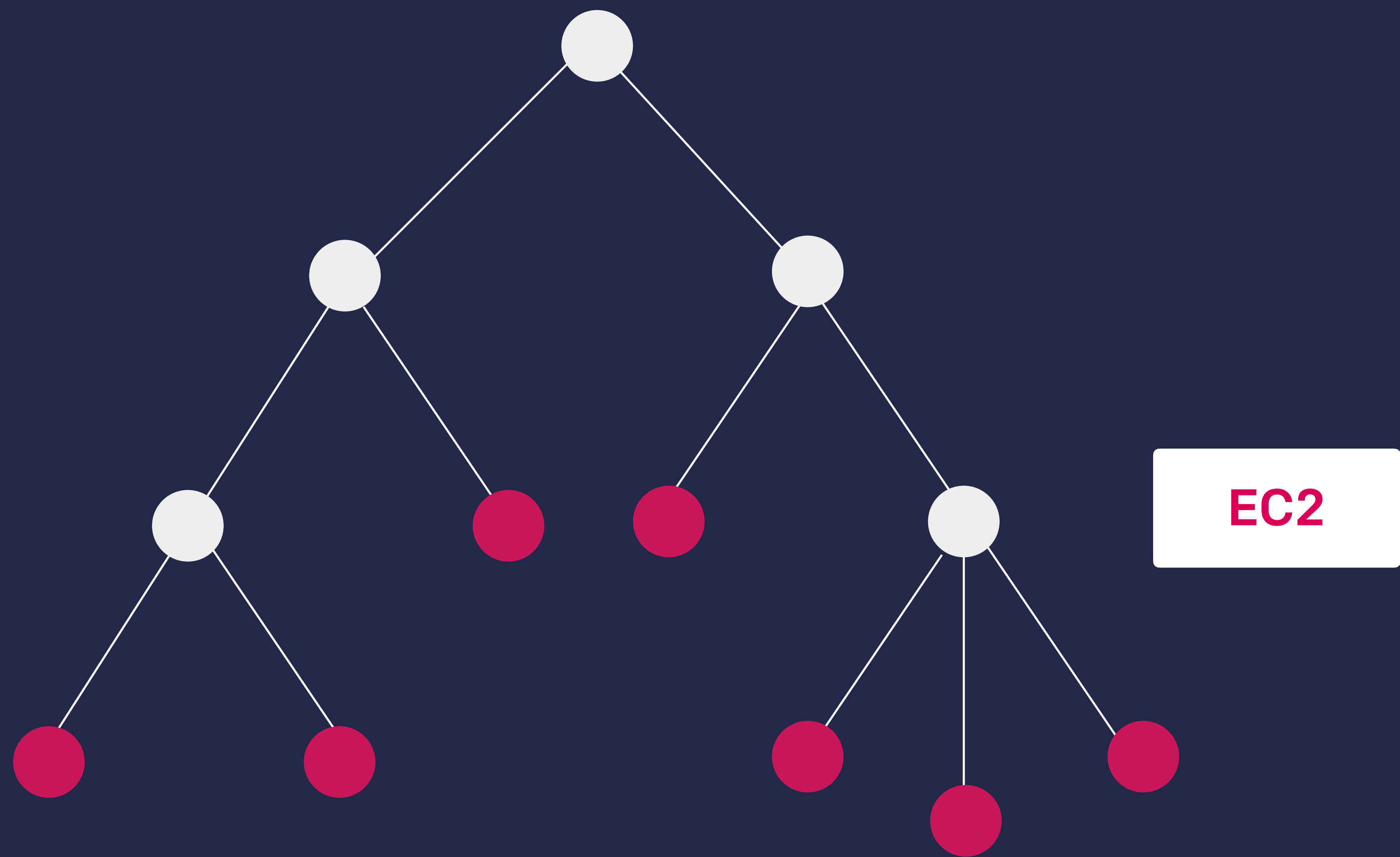
Building companies and products requires **choosing a set of abstractions**

The question is **what use-cases are you abstracting away** in your product? How many are there?

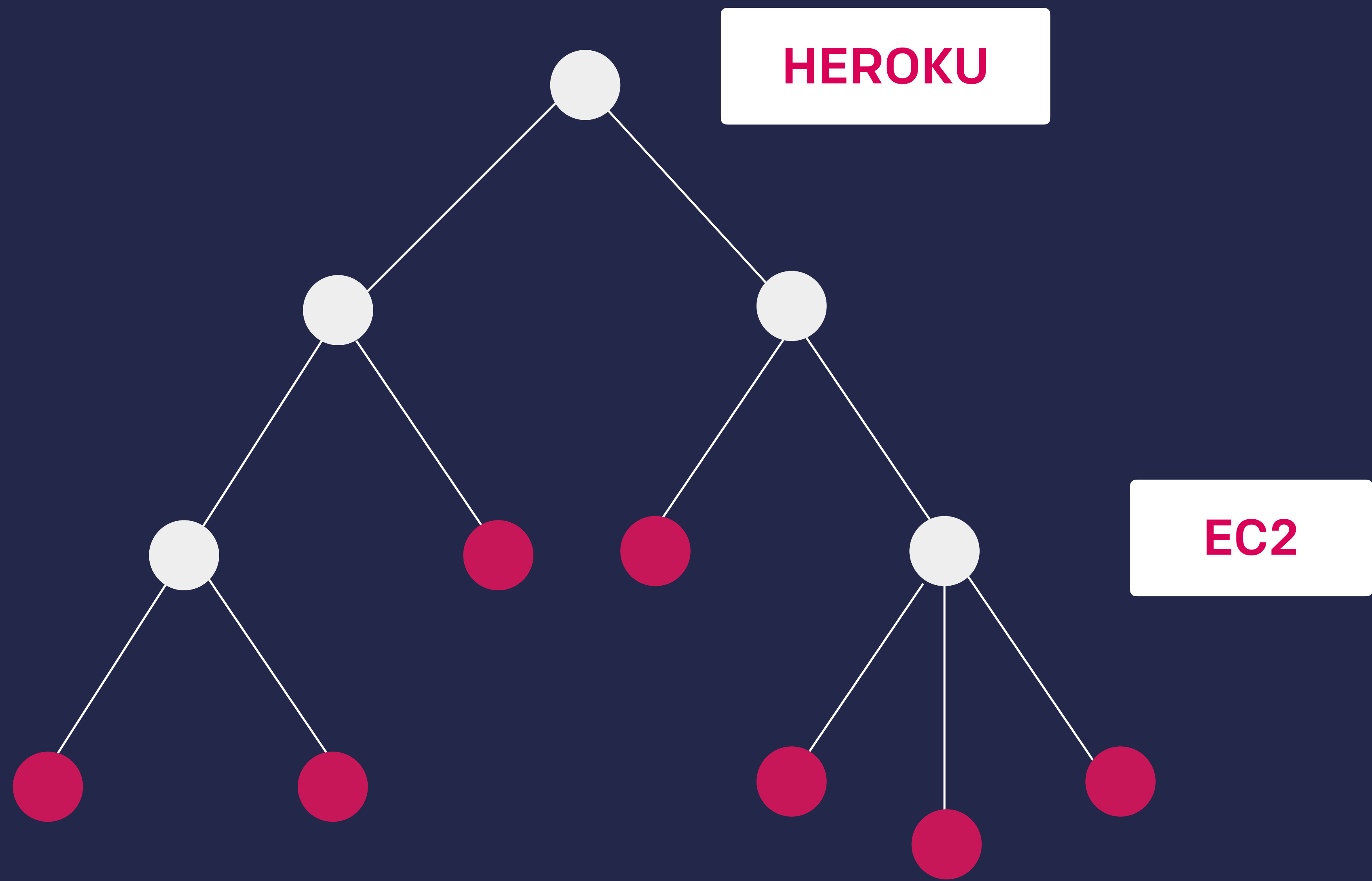
Inspired by Cheng Lou (Facebook)



Inspired by Cheng Lou (Facebook)

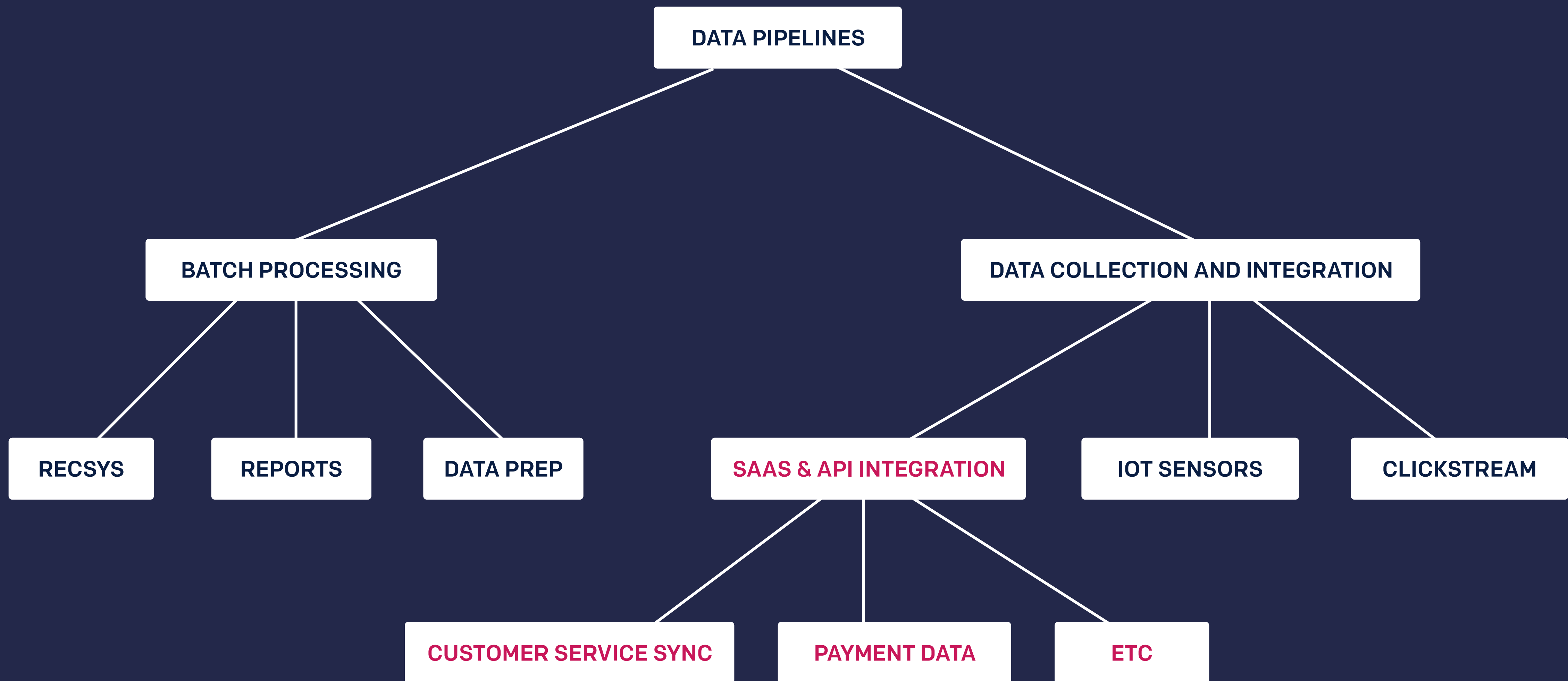


Inspired by Cheng Lou (Facebook)









# Value prop

- No deployment from user
- Not writing api adapters and glue code
- Off the self connectors
- Pre-built authentication
- Not writing tests and worrying about fragile code

The product should **abstract away the complexities** of those specific use-cases while maintaining **flexibility and expressibility**

# Options for the product

# Options for the product

- UI for specific use-cases

# Options for the product

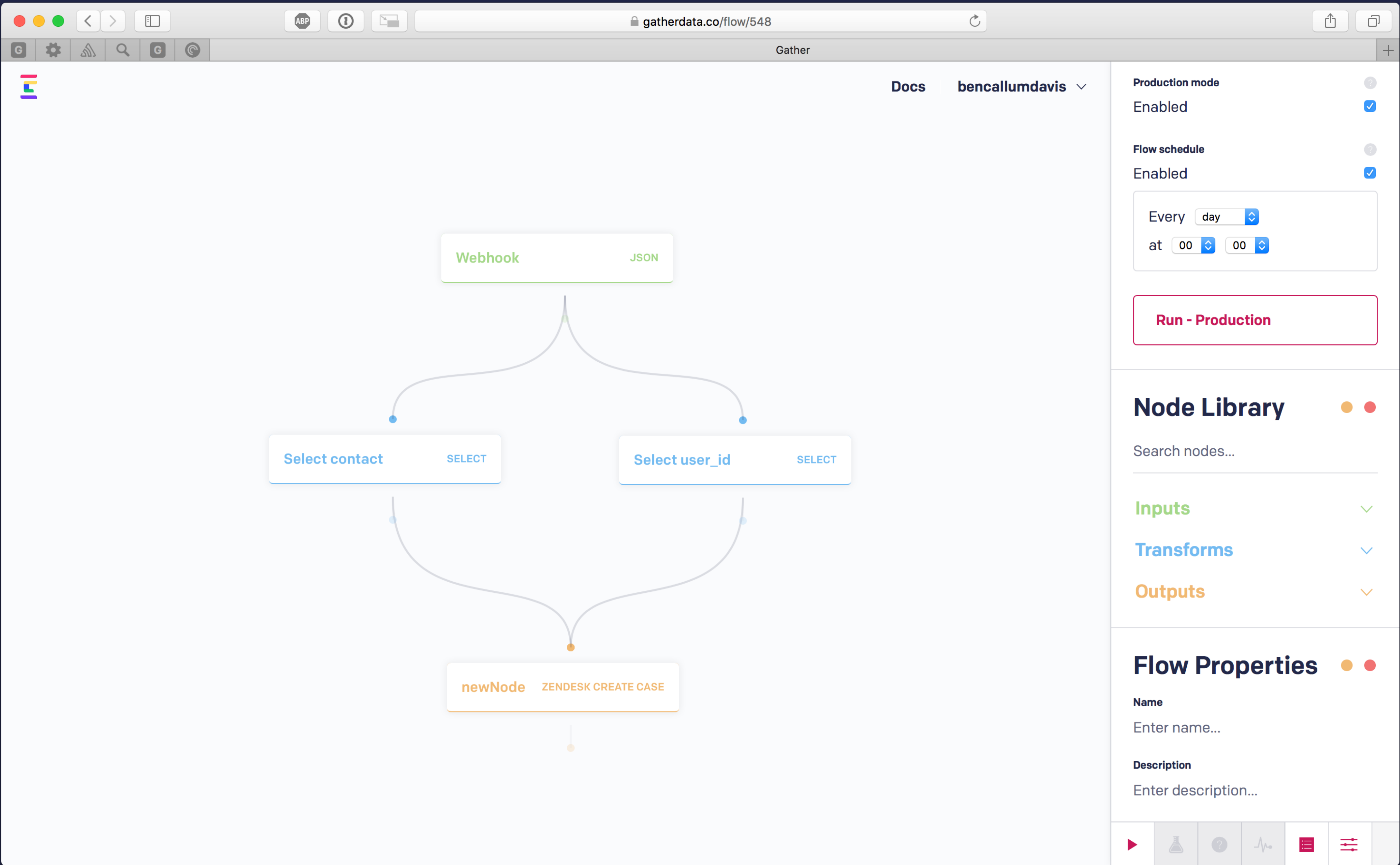
- UI for specific use-cases
- Python SDK

# Options for the product

- UI for specific use-cases
- Python SDK
- Kubernetes-like declarative data flow?



Kubernetes is the right inspiration because it operates  
at the **same level of abstraction**



```
1  args: {}
2  nodes:
3    '1':
4      args:
5        remote_url:
6          documentation: An RSS feed
7          name: remote_url
8          required: true
9          type: str
10         value: http://rss.sciam.com/ScientificAmerican-Global
11      meta:
12        position:
13          x: 480
14          y: 311
15        name: Scientific American
16        type: fetcher/rss
17    '2':
18      args:
19        join:
20          documentation: Either `inner` or `outer` - default `outer`. Inner will keep
21            only rows that intersect.
22          name: join
23          required: false
24          type: str
25          value: outer
26      inputs:
27        - '1'
28        - '3'
29      meta:
30        position:
31          x: 620
32          y: 569
33        name: Concatenate Rows
34        type: operator/bind_rows
```

# Conclusion

- Starting at too higher level of abstraction
- Building the tree is hard
- Building a product that is misaligned with where you've position yourself on that tree

**THANKS FOR  
LISTENING**

**TALK TO ME. PLEASE**

**[ben@gatherdata.co](mailto:ben@gatherdata.co)**