ML Infra at an Early-Stage

Feature Service

Spencer Barton, Data Scientist

April 2019

branch

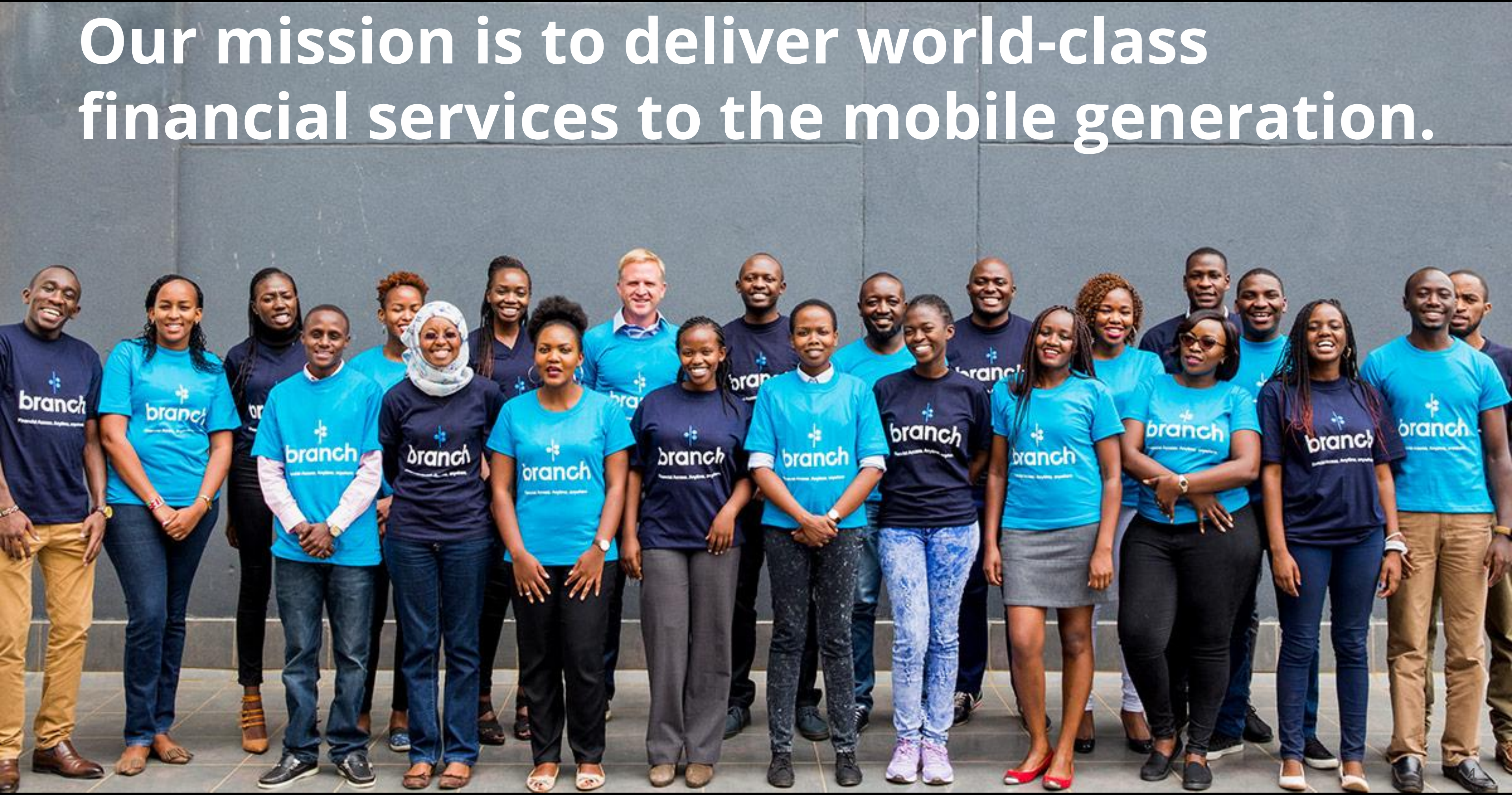
About | How it works | Careers | In the news | Contact | Log Out | Admin

Big challenges require big minds

We're interested in the rising stars with a worldly perspective, a deep interest in financial technology, and an appetite for growth.

See current openings

Our mission is to deliver world-class financial services to the mobile generation.



From Install to Approval in Minutes

1

ANSWER 3 QUESTIONS TO REGISTER

KYC checks with external APIs, mobile data mined and analysed.

2

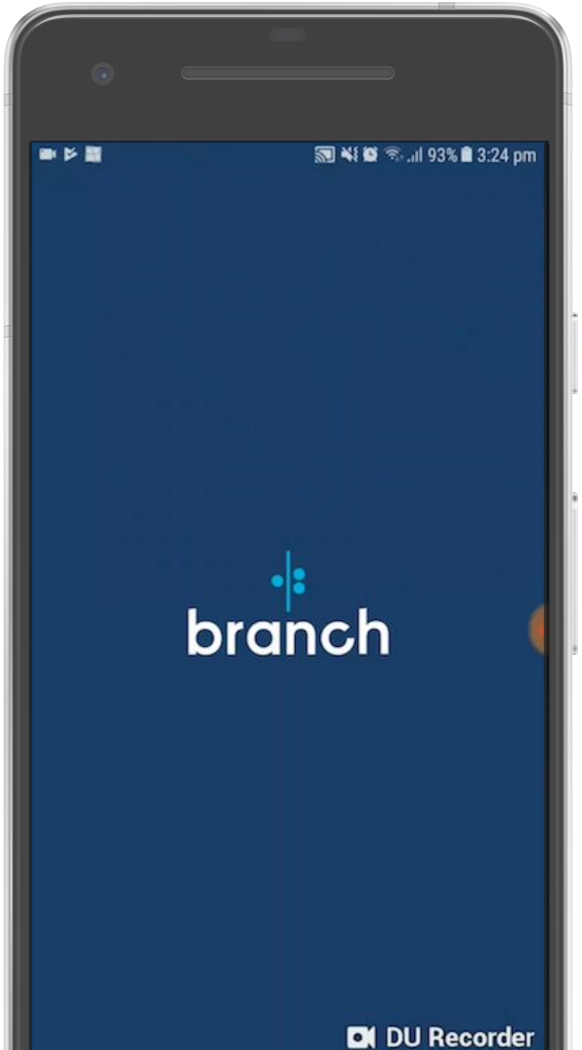
ELIGIBLE LOAN OFFERS ARE DISPLAYED

Credit score calculated in seconds.

3

DEPOSIT TO BANK ACCOUNT OR MOBILE WALLET

Repayment schedule set and monitored.



branch

How Branch works behind the scenes

Collect Phone Data → Generate Features → Credit Model

We collect
- Text messages
- Installed apps
- Contact lists
- In-app events

We extract
- Bank balance
- Number of contacts
- Read the FAQ
- Installed Facebook app

We predict probability of repayment

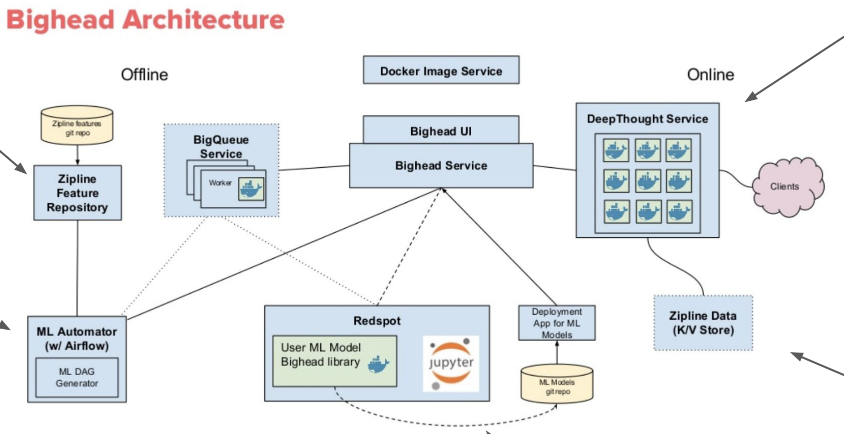
How do I build ML into my product?

Big Firms Can Build Custom ML Infrastructure

5 engineers

airbnb

10 engineers



5 engineers

2 product managers

5 data scientists

10 engineers

Source: [Bighead - Airbnb's End-to-End Machine Learning Platform]

Can the rest of us do machine learning?

We too can build infrastructure but must be strategic.

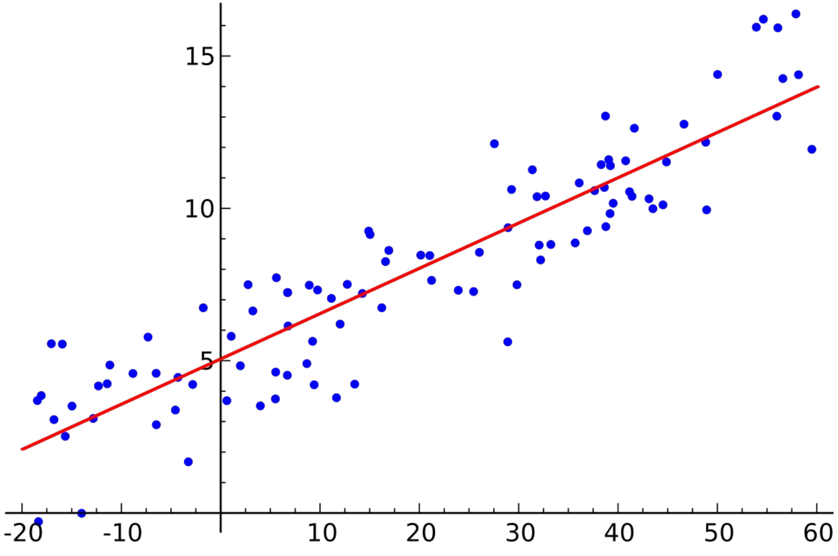
Build a Feature Service!

What does a feature service do for me?

- Faster development of new features
- Reduce bugs with consistent feature definitions
- Speed-up slow feature calculations
- Easy feature discovery and sharing

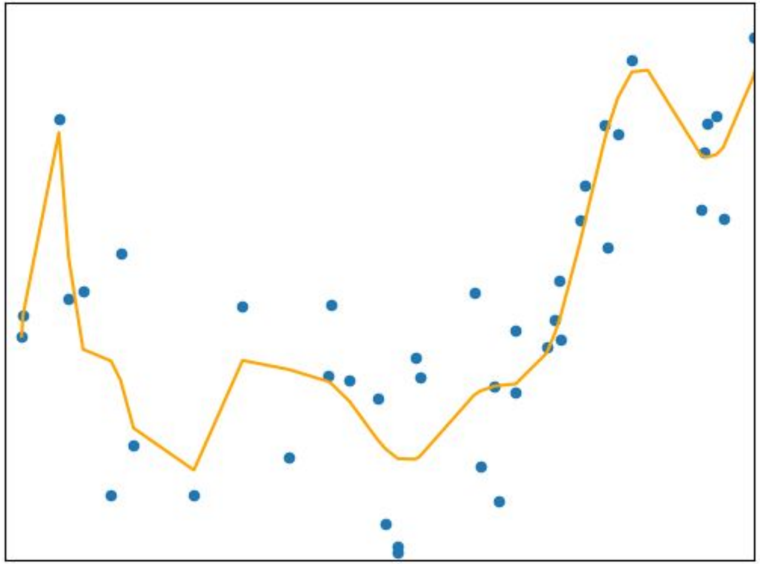
Where do you start?

You want to start basic



https://en.wikipedia.org/wiki/Linear_regression

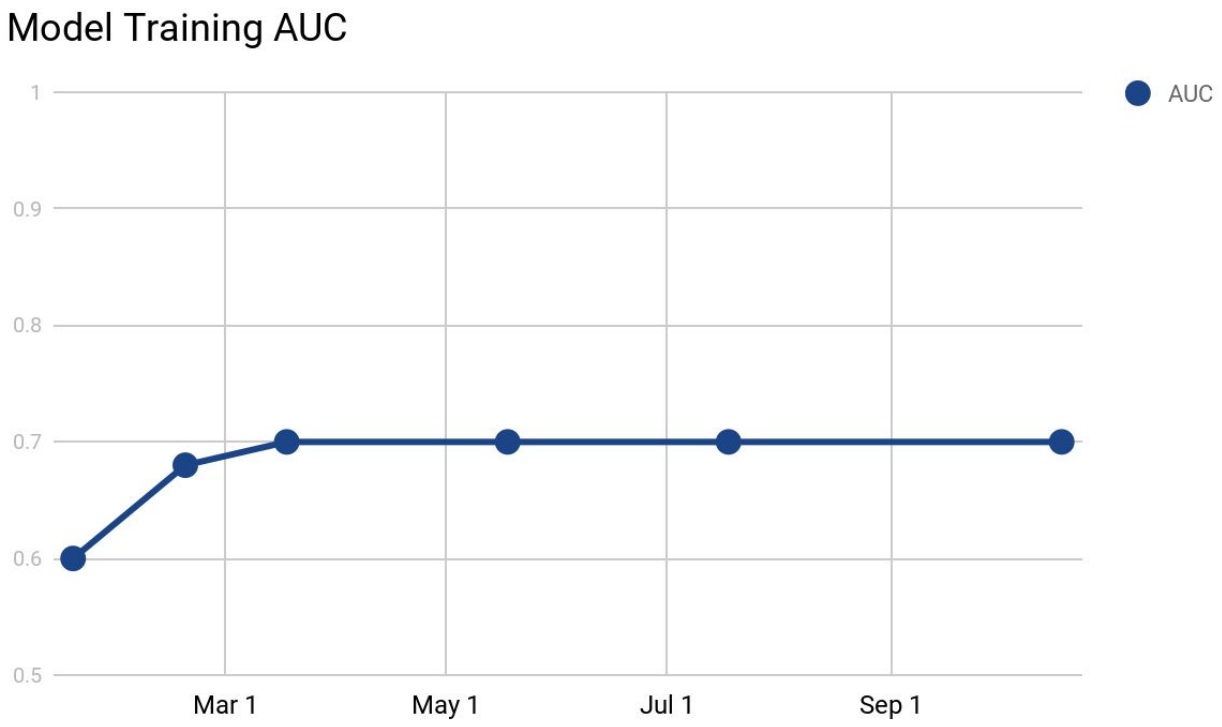
You will gradually mature your ML



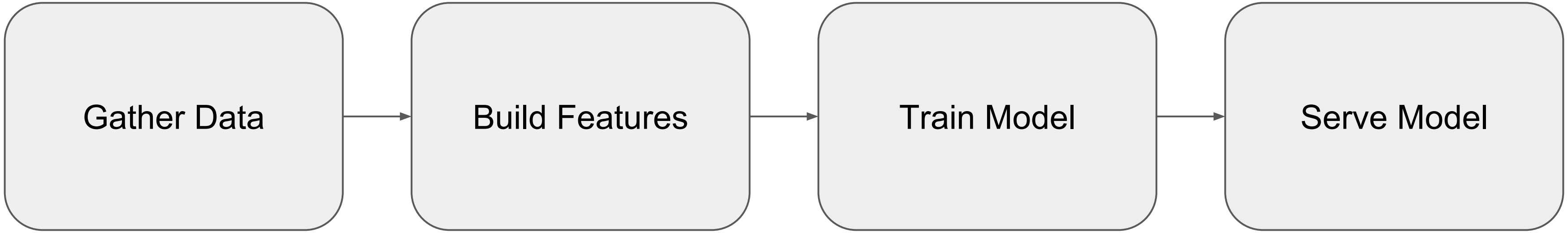
https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

branch

The basics will only get you so far



What do you focus on beyond the basics?



We needed to improve our features

Our data sources were in ok shape but

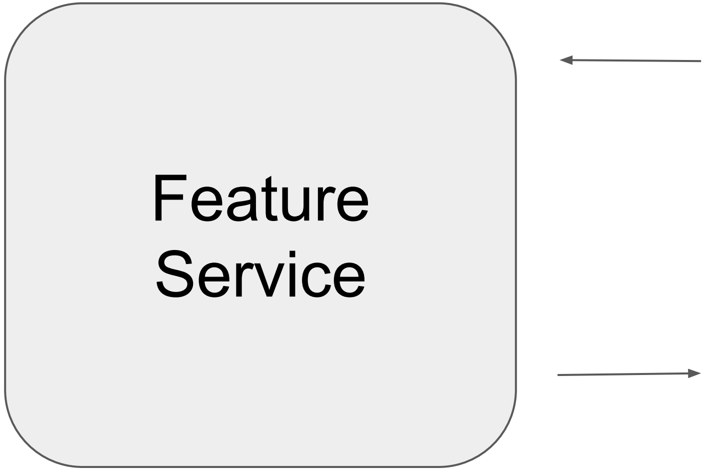
- Differences in features between dev, training and production lead to bugs
- Inconsistent feature definitions lead to bugs
- Feature creation was a training bottleneck

We invested in infrastructure to improve features.

We decided to build a Feature Service

What is a Feature Service?

A Feature Service computes a feature vector for a specific object at a specific time.



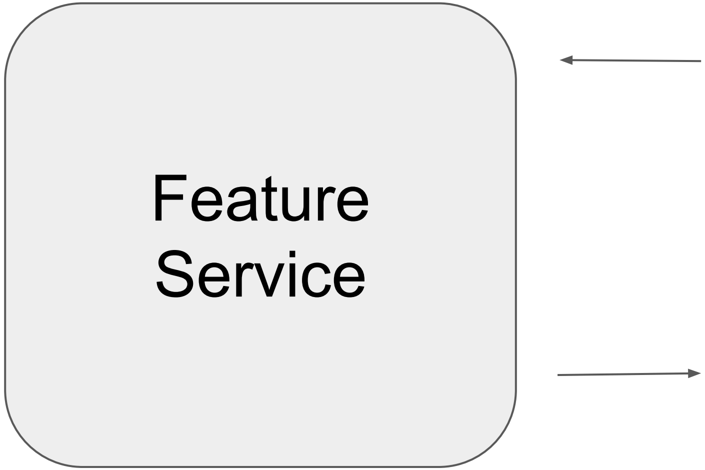
Get features for user 90234

Feature vector for user 90234

```
{
  "average_bank_balance":
324090,
  "number_referrals": 15,
  "read_faq": true
}
```

branch

Features are computed relative to a timestamp



Get features for user 90234 on 2016-10-2

Feature vector for user 90234 on 2016-10-2


```
{
  "average_bank_balance":
504090,
  "number_referrals": 0,
  "read_faq": false
}
```

Features are accessed by a simple API

```
GET feature/bank_balance/v0_1?pid=12314

GET feature/bank_balance/v0_3?pid=1214&date=2017-12-3

GET feature/loan_repayment/v0_1?pid=3531
```

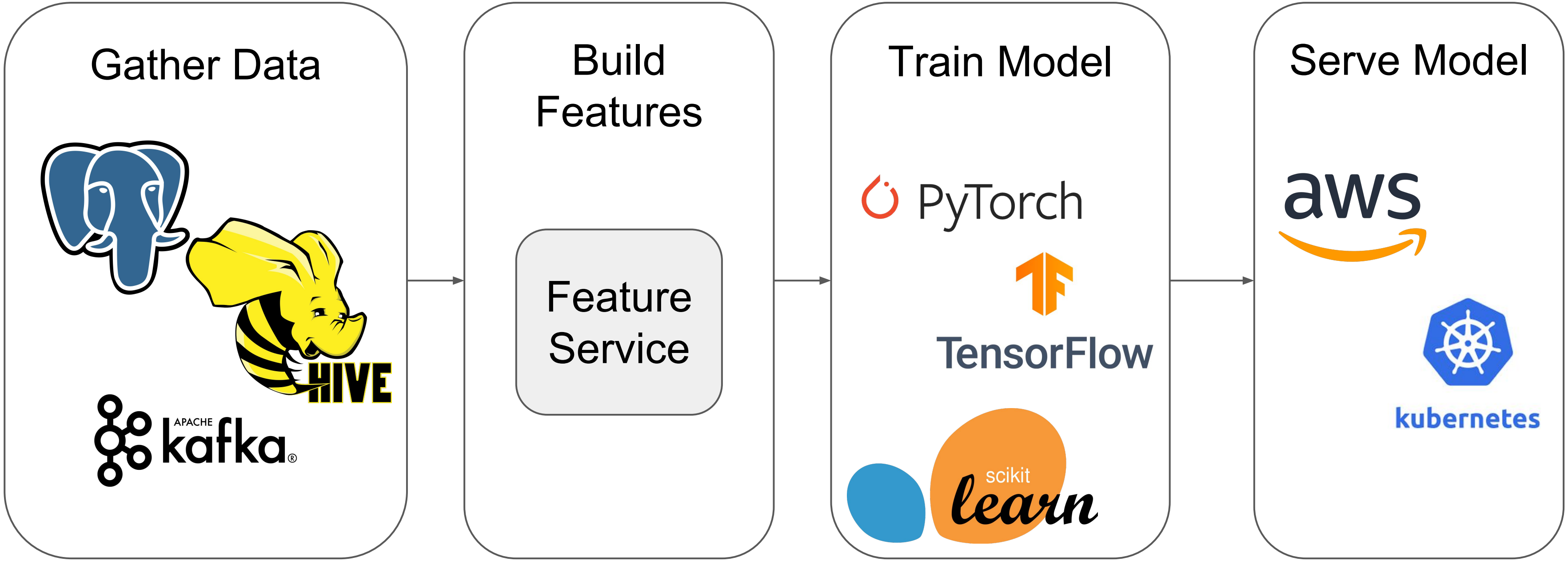
feature name

feature version

date for historical features

pid = primary id, like user id

Why build a custom solution?



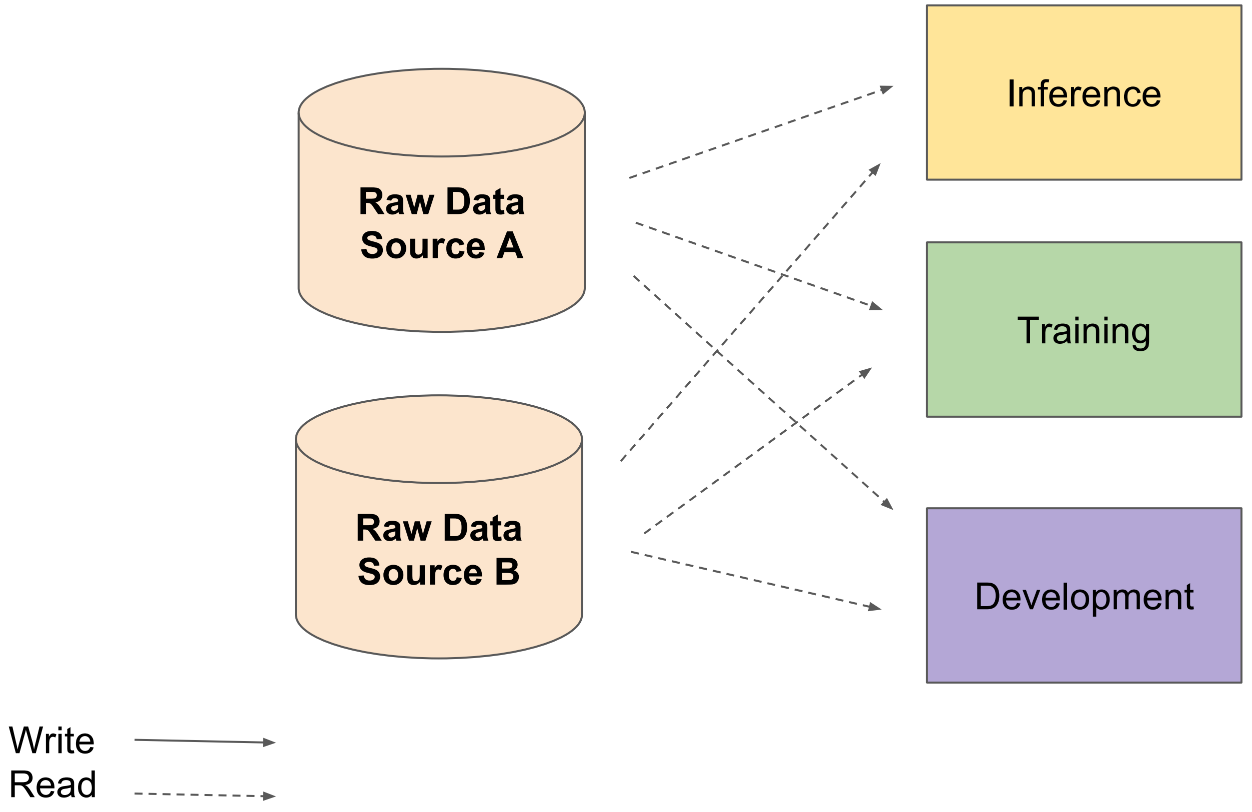
What are we building?

Feature Service

- Server infrastructure
- Cache infrastructure
- A Python framework

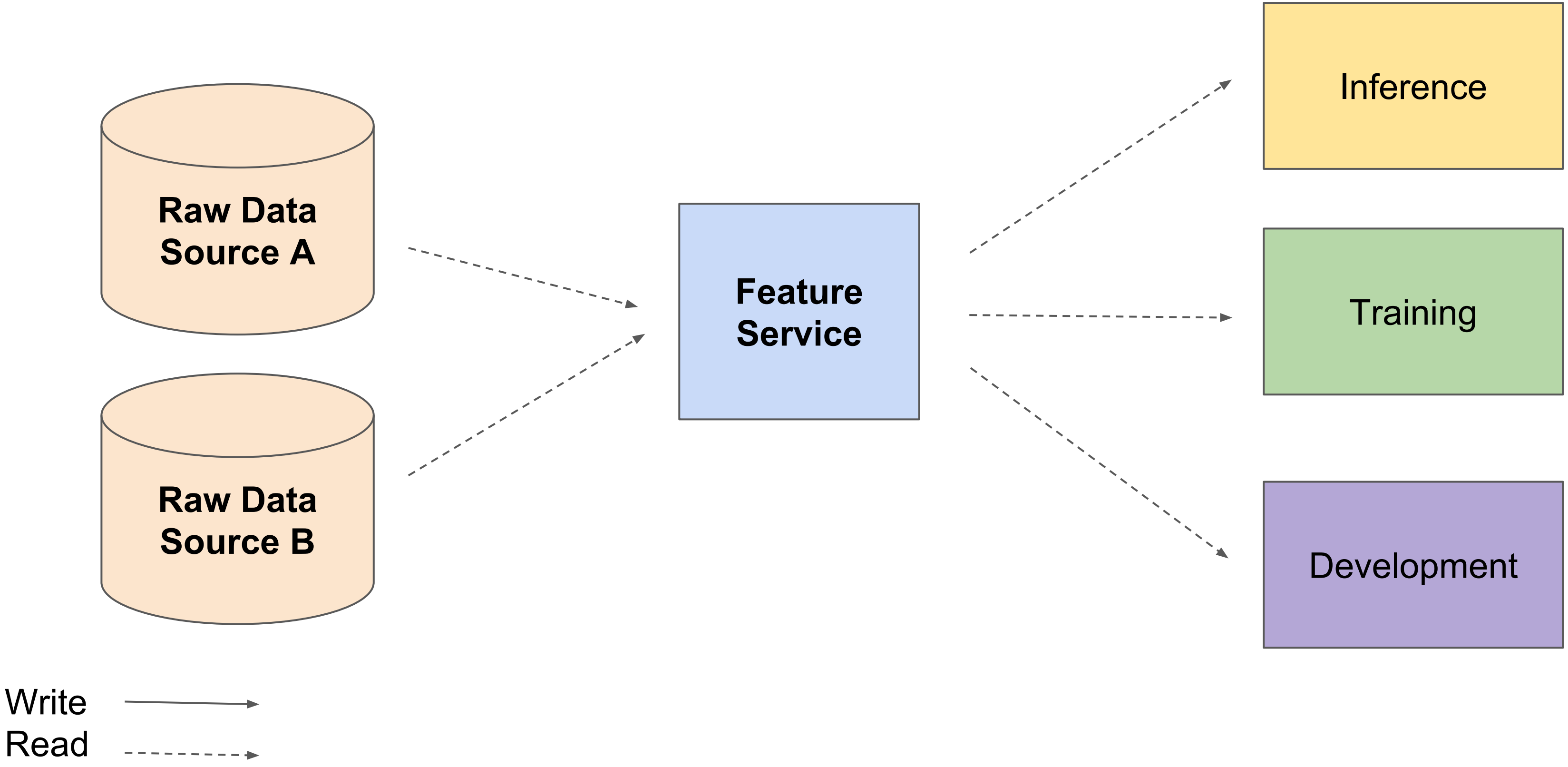
branch

Data source dependencies were messy

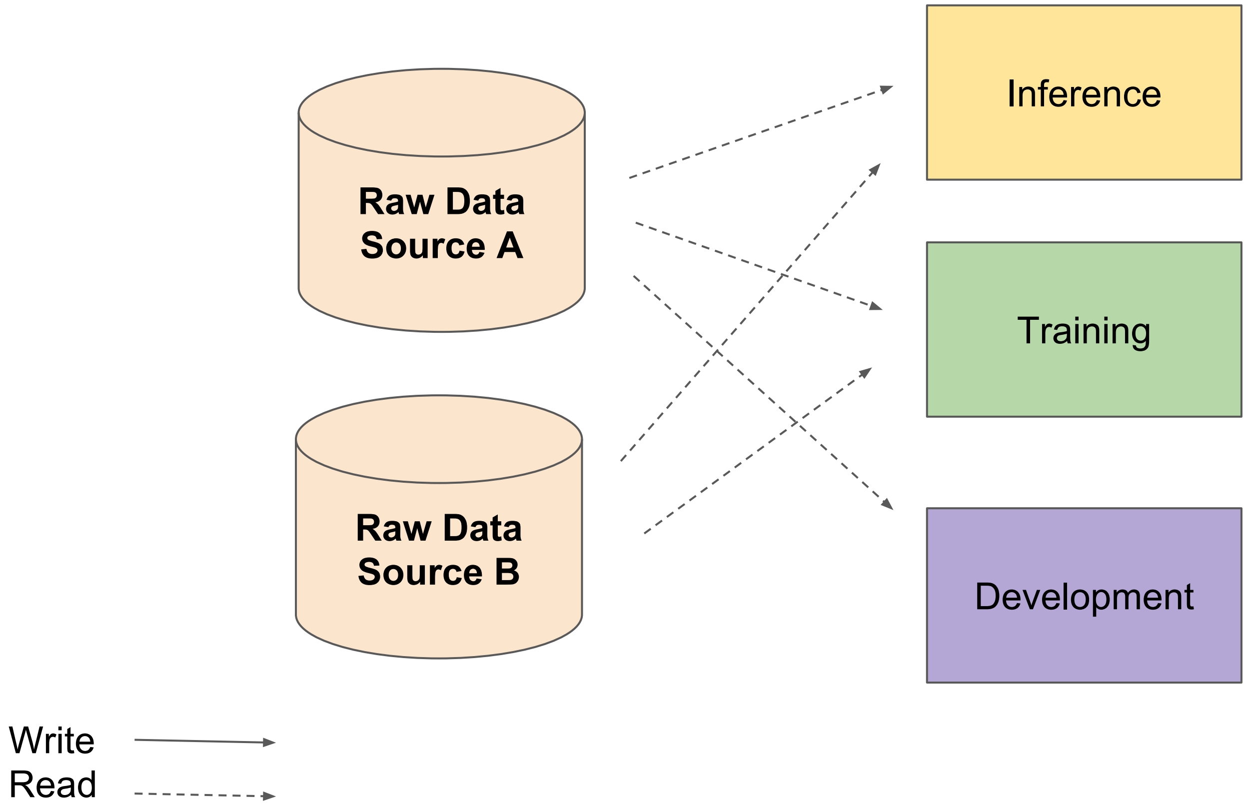


branch

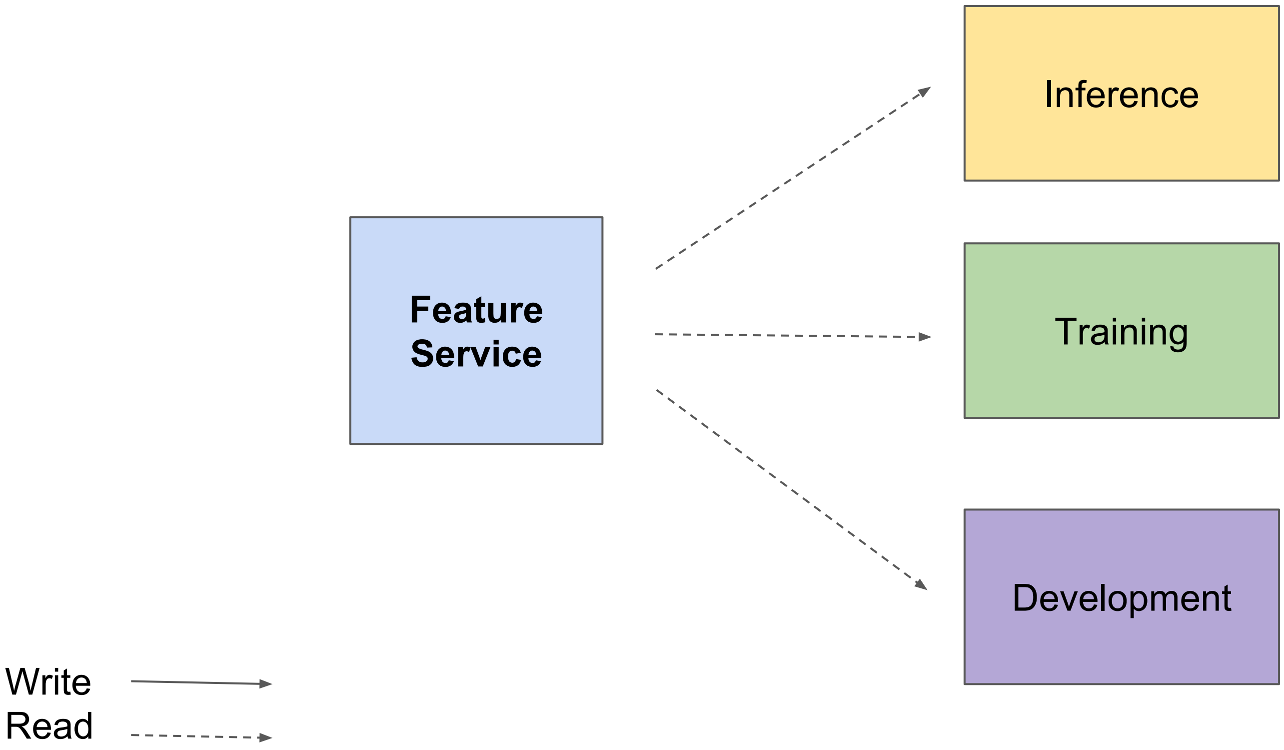
We abstracted complicated data sources



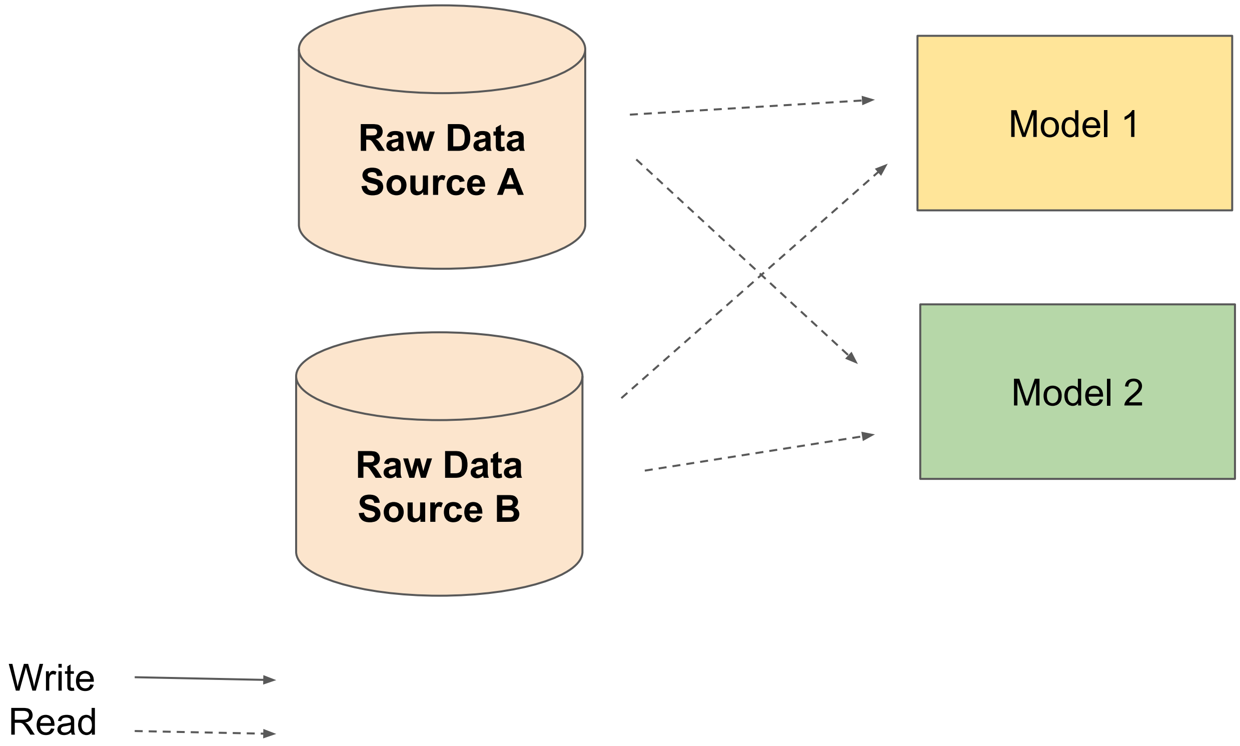
Features were being created all over the place



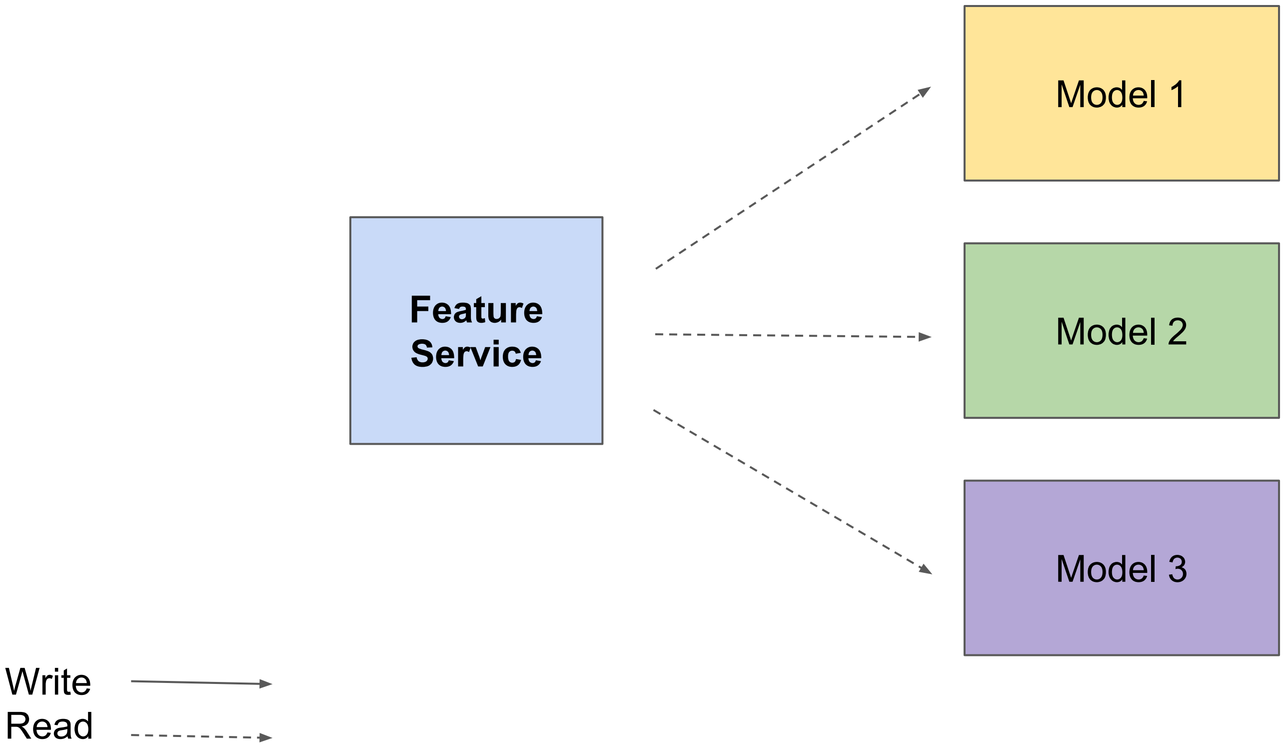
Every step of ML shares consistent features



New models were recreating features



ML models now share the same features

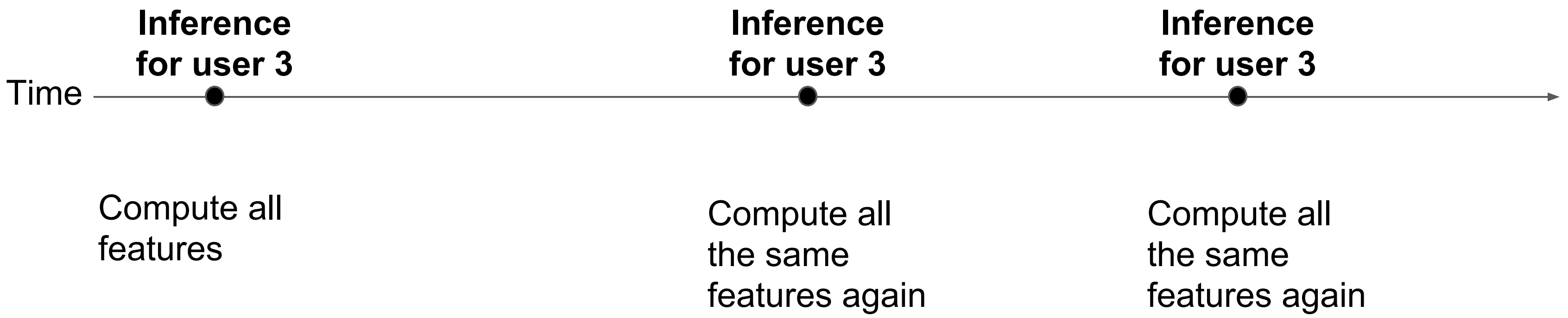


The Feature Service server helps a lot

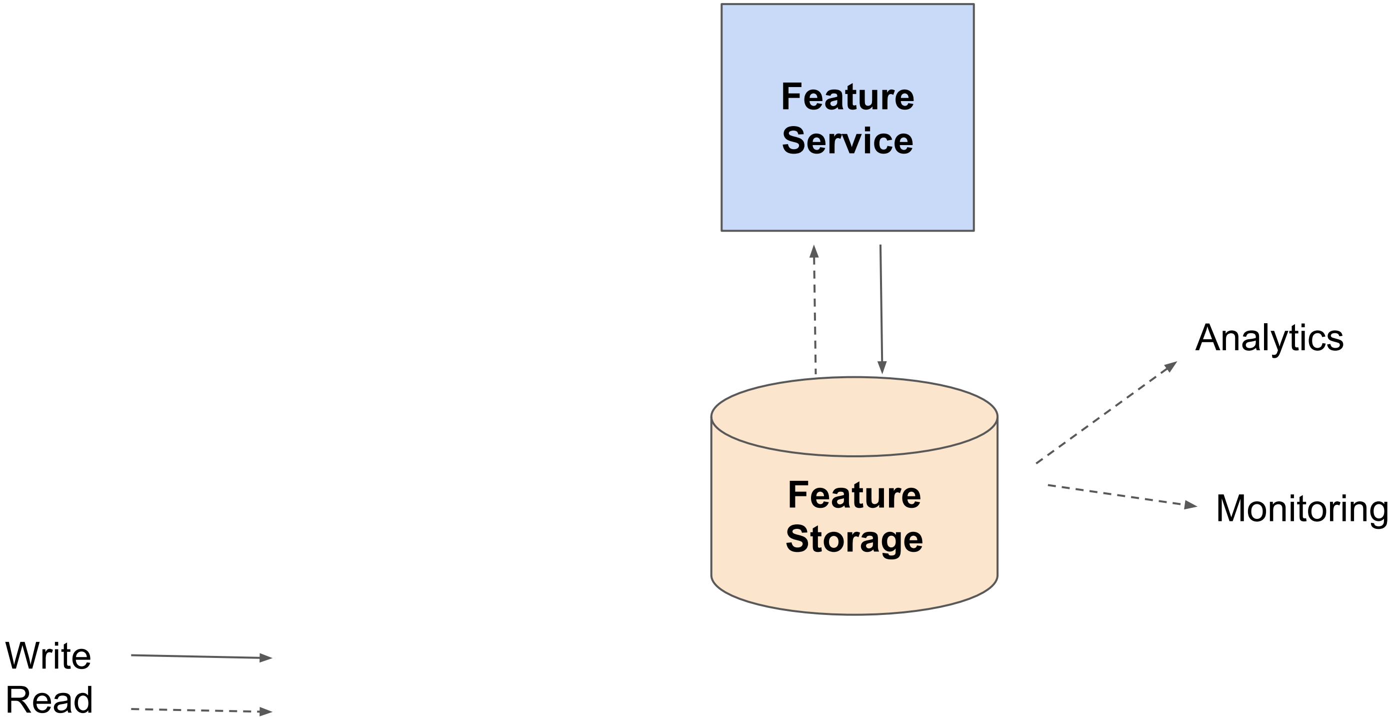
- Abstracted data sources
- Shared features
- Consistent features

Now onto storage….

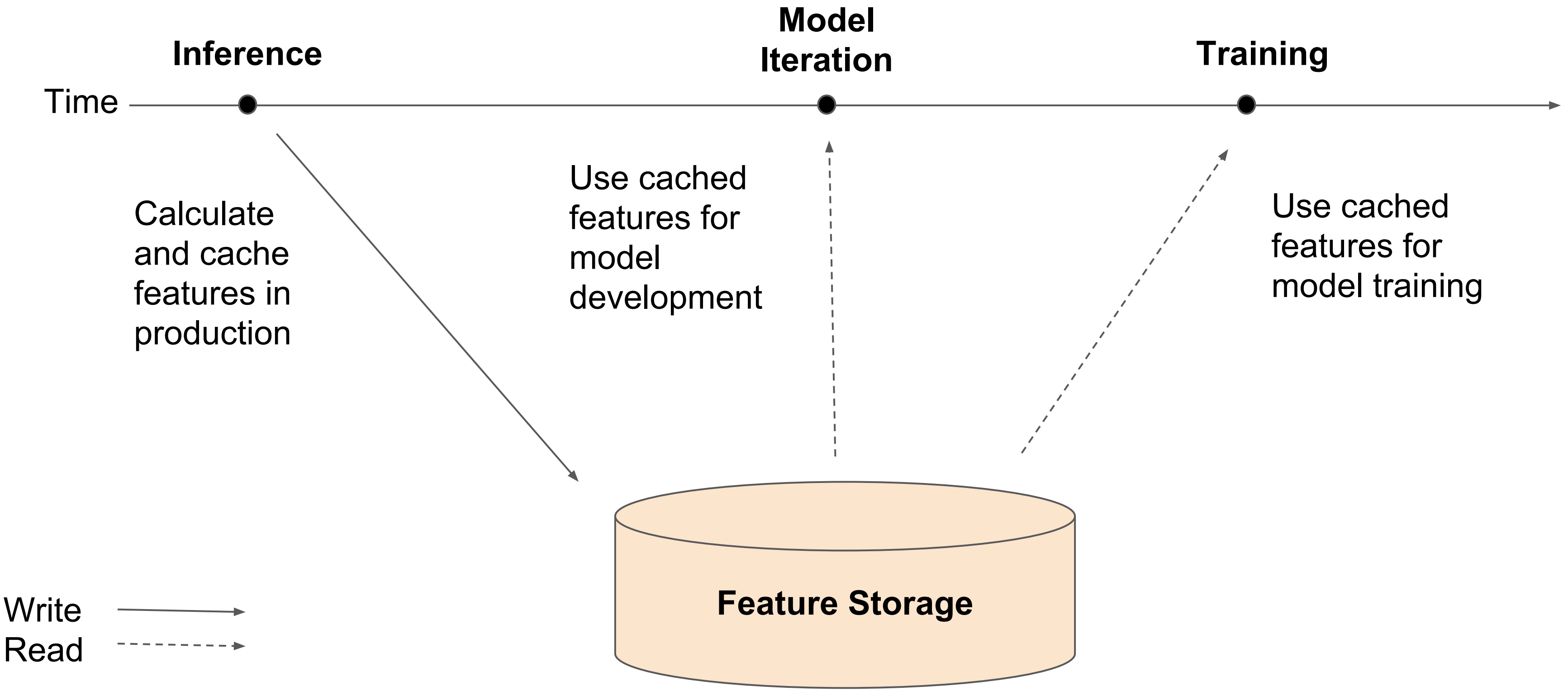
Features were computed once and forgotten



We built feature storage and caching



We sped up training with a cache

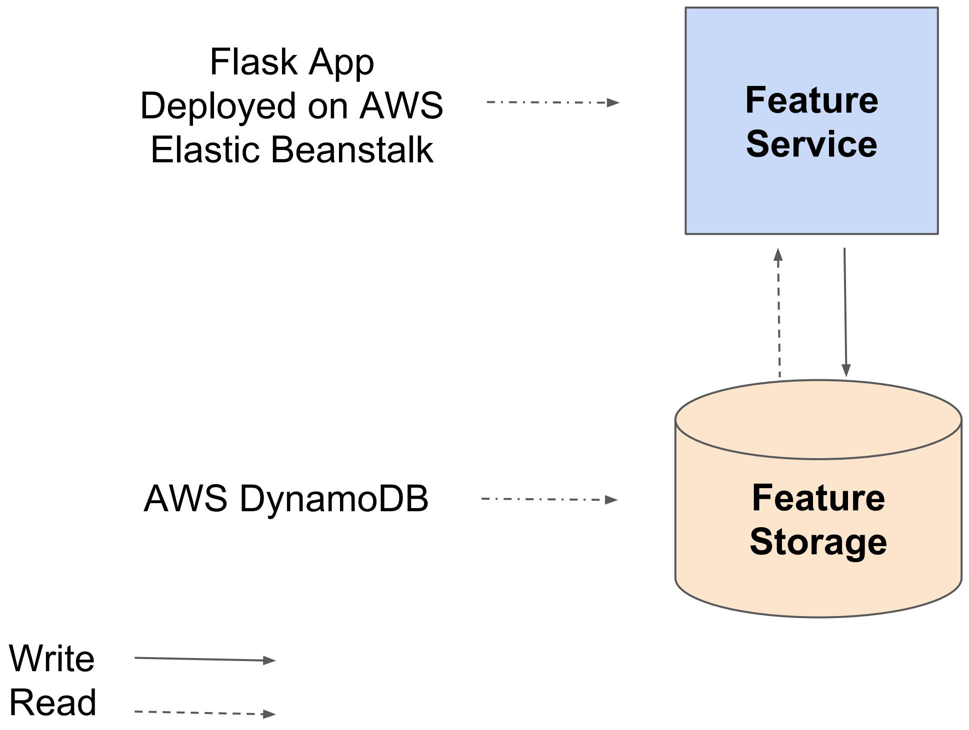


branch

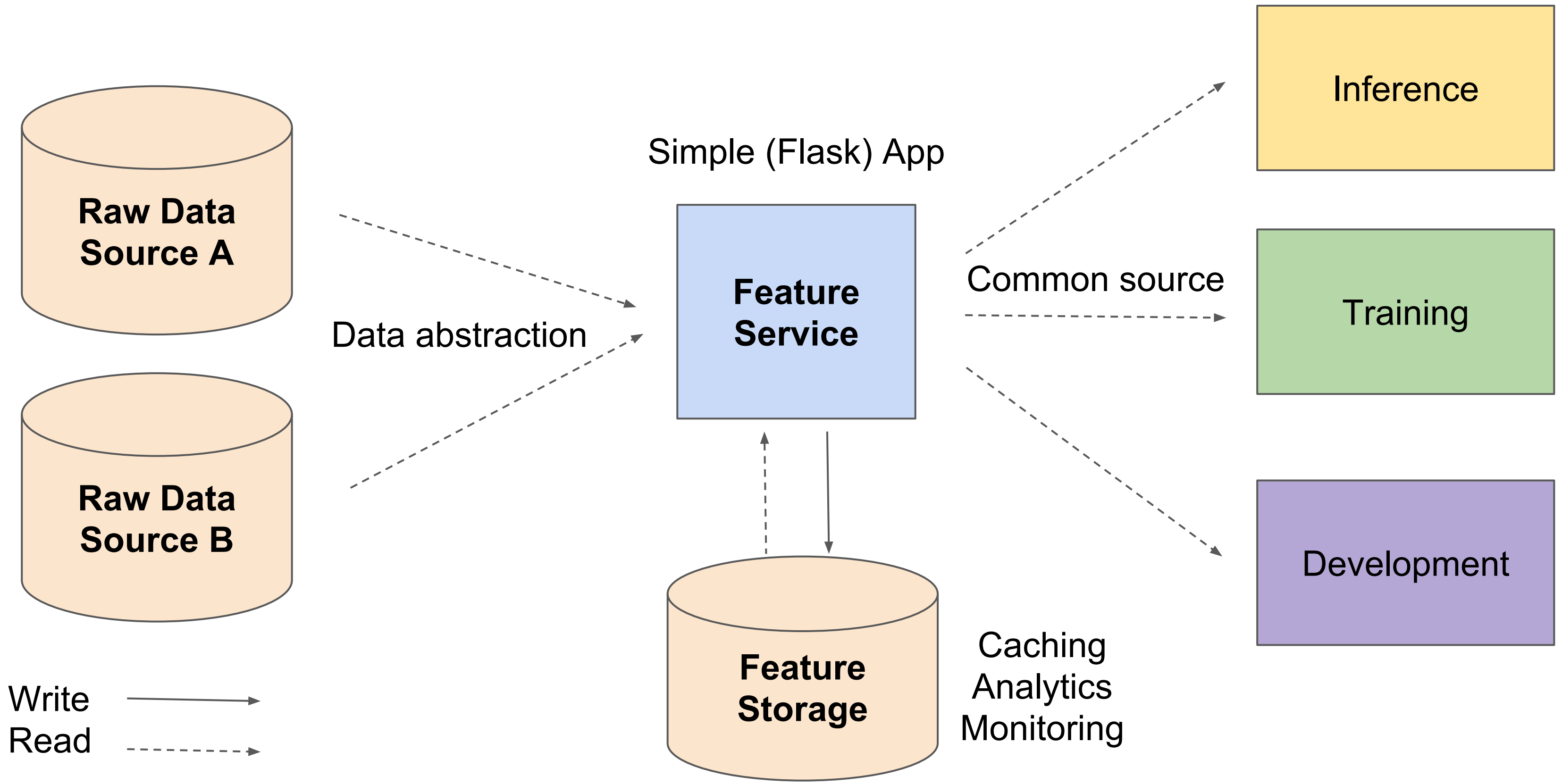
Feature storage helps too

- Remove recomputation of features
- Enable analytics and monitoring
- Increase training speed

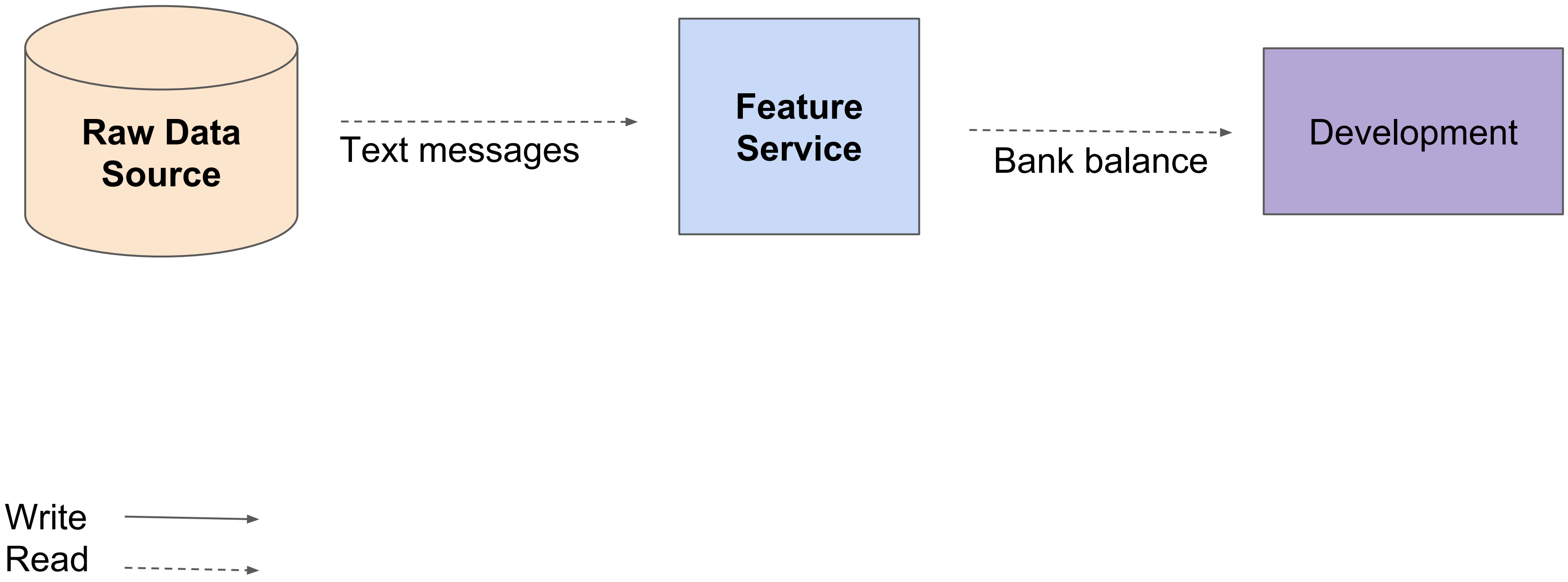
We built with simple components



Simple infrastructure solved many problems



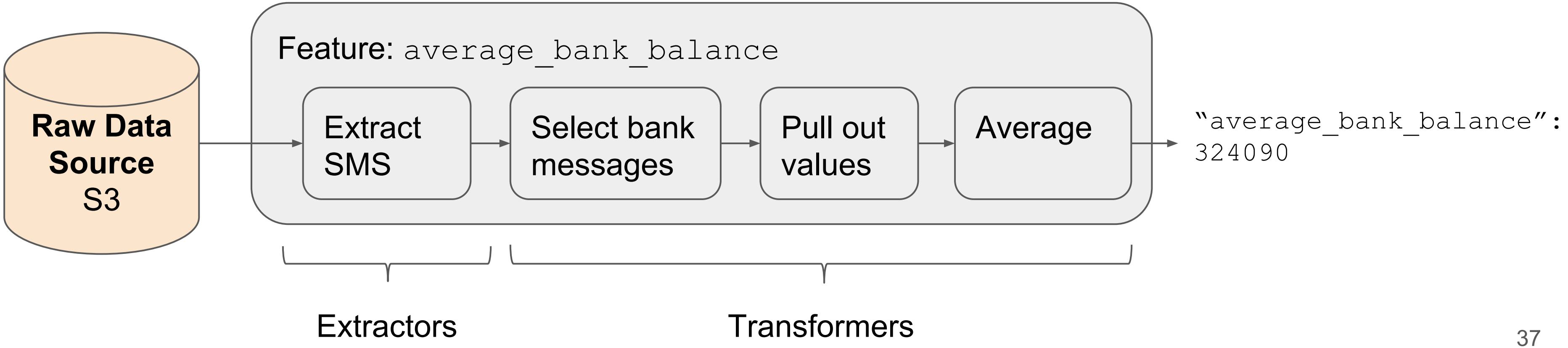
How do we actually generate features?



We built a framework

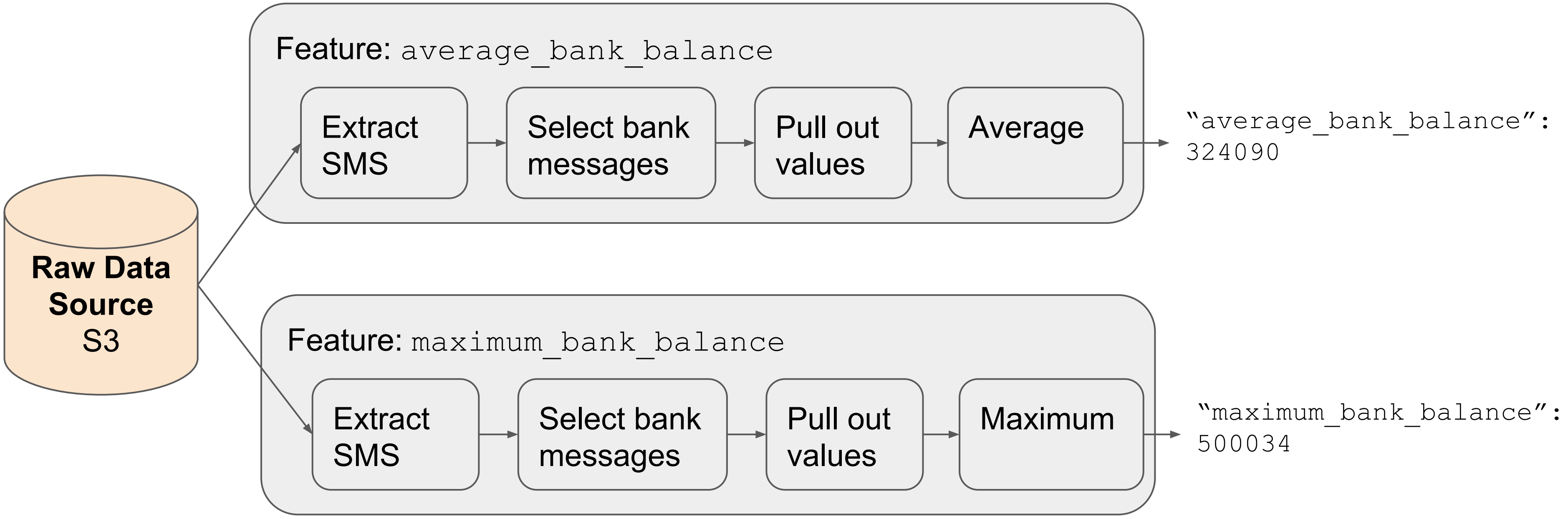
Features are composed of

- One or more *Extractors* which pull data from a Raw Data Source
- Many *Transformers* which convert the data into a numeric or categorical features



branch

Extractors and Transformers are shared



Framework example

Everything is built on base classes with automated testing

Features are built on versioned extracts and transforms

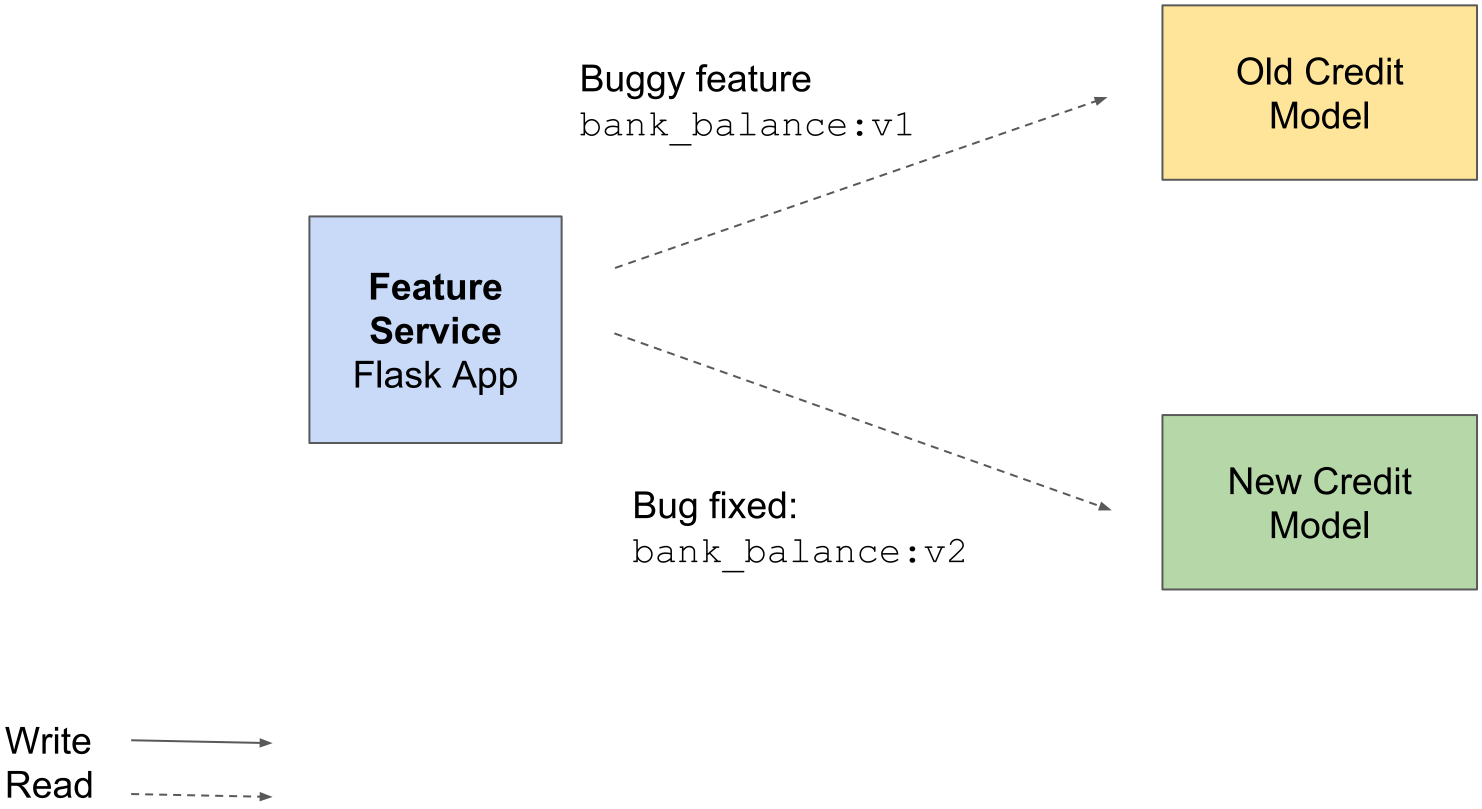
As flexible as Python

Chain of transformations

Custom one-off transforms

```python
from framework.feature import Feature
from framework.transform import Transform

from extract.sms.v0_3 import SmsExtract

from transform.filters.filter_row_values.v0_1 import FilterRowValues
from transform.mappers.pluck_regex_value.v0_2 import PluckRegexValue
from transform.column_utilities.select_column.v0_1 import SelectColumn
from transform.column_utilities.rename_column.v0_1 import RenameColumn


class AverageBankBalance(Feature):

    BANK_ADDRESSES = [
        "MPESA",
        "NIC",
        "CHASE",
    ]

    def __init__(self, **kwargs):
        pipeline = [
            SmsExtract(),
            FilterRowValues(column="address", values=self.BANK_ADDRESSES),
            PluckRegexValue(column="message", pattern=r"KSH(\d+)"),
            SelectColumn(column="message"),
            RenameColumn(from="message", to="average_bank_balance"),
            WeightedAverage(column="average_bank_balance"),
        ]

        super().__init__(pipeline=pipeline, **kwargs)
```

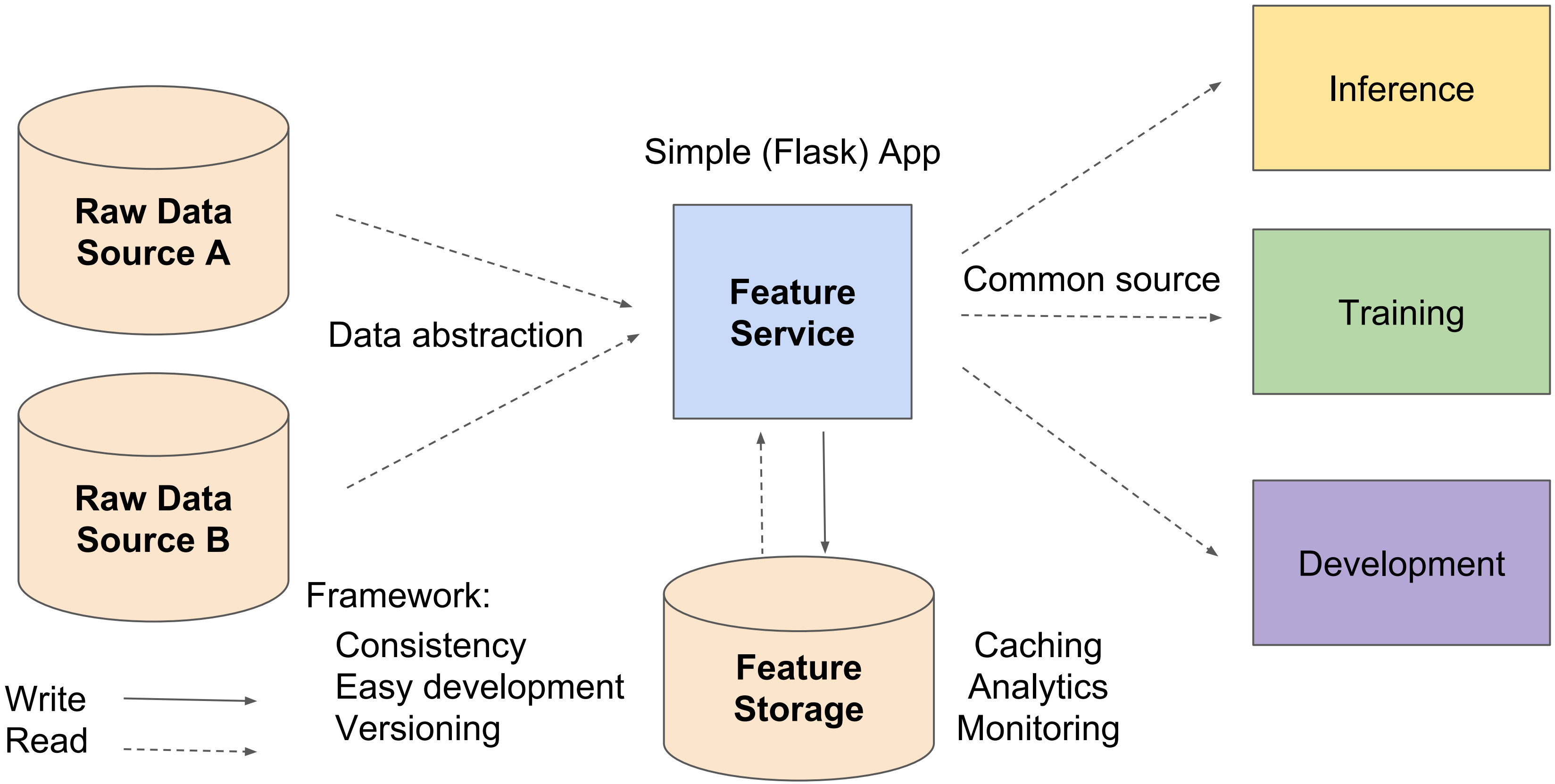
Feature versions support new models



The framework makes development easy

- Feature definitions are consistent
- New features are easy to build from shared components
- Versioning allows backwards compatibility and bug fixes

The Feature Service solves many problems



Should I build a Feature Service?

- Is feature quality a problem for you?
- Are your data sources complex and varied?
- Do you want to support multiple models?
- Are your features difficult to compute?

branch

We're benefitting from our Feature Service

- Feature generation time reduced!
- Fixed a lot of bugs by using the framework!
- New models without remaking features!
- New data scientists can contribute within a week of joining!
- And our model performance has improved!

What should I take away?

- You don’t have to be a big company to use ML infrastructure
- But your resources are limited so be strategic
- And invest in a Feature Service!
- Stay informed because the landscape changes fast
  - Airbnb Big Head may be open sourced soon

The Team

Dennis Van Der Staay

Dave Bernthal

Ting Ting Liu

Nick Handel

Spencer Barton

Thank You!

Spencer Barton
spencer@branch.co

Appendix

Who else is talking about Feature Services?

- Nick Handel delivering an earlier version of this presentation
- Varant Zanoyan, Zipline at Airbnb
- Uber's Michelangelo