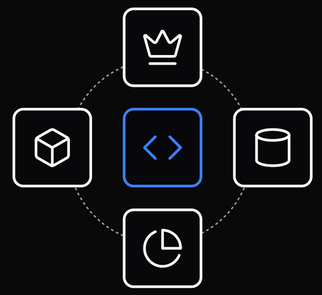
fiveonefour

Hi,
I’m Chris!

Our mission at fiveonefour is to bring
incredible developer experiences
to the data & analytics stack

Fiveonefour Products

Open Source Framework



Moose

Public Beta

Open source data engineering framework for data products and pipelines

- ✓ Production data eng in Py / TS
- ✓ Intuitive abstractions
- ✓ Local-first dev experience

Commercial Hosting

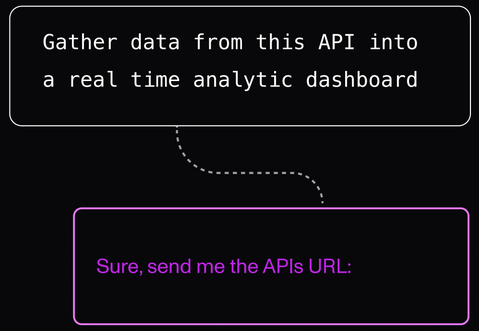
Boreal Cloud

Private Beta

Quickly, securely and scalably get your data-intensive application into production

- ✓ Managed infrastructure
- ✓ Secure & compliant
- ✓ One-click deploy

AI Data Engineer



Aurora AI

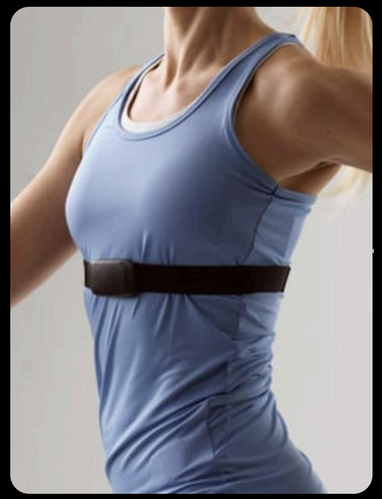
Pre-Alpha

AI agents automate the repetitive and tedious parts of data engineering

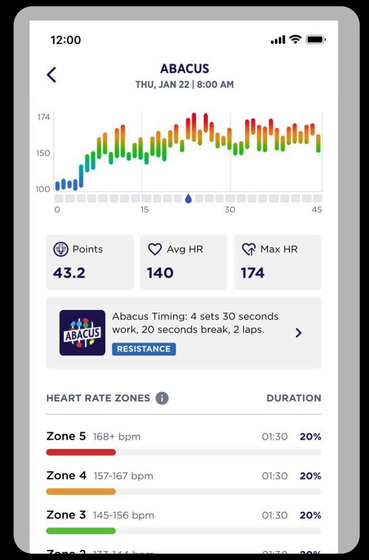
- ✓ Context gathering
- ✓ Code generation
- ✓ Project management


F45
TRAINING
×
fiveonefour

In-gym Heartrate Monitors



Data & Analytics Backend



F45 Mobile App

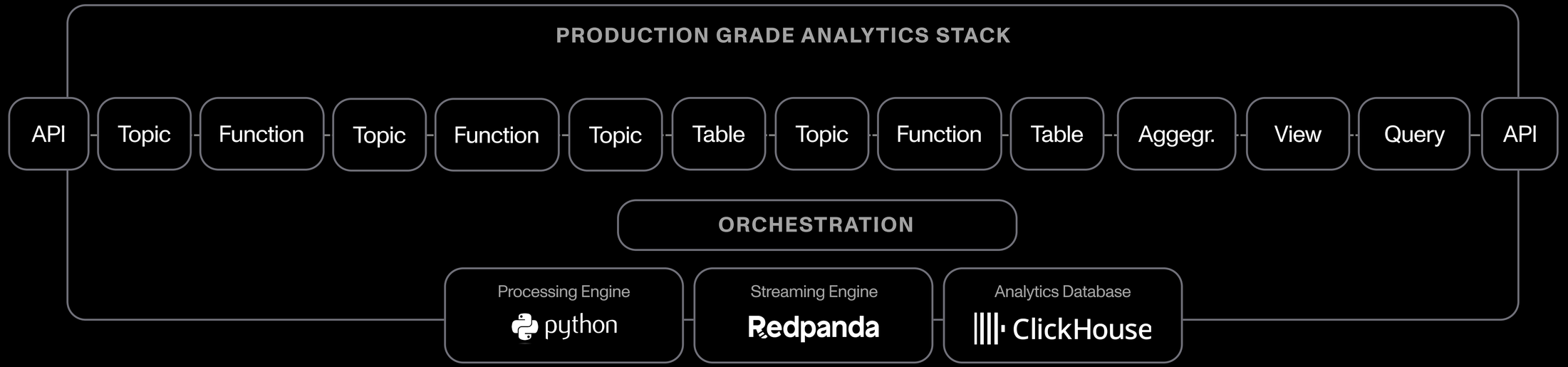
The challenge

120,000+
members

9,458,000
workout sessions

24,300,000,000
data points

How to realize these outcomes



Infrastructure boilerplate

```
# Redpanda broker
resource "docker_container" "redpanda" {
  name  = "redpanda-broker"
  image = "docker.redpanda.com/redpandadata/
redpanda:latest"

  command = [
    "redpanda",
    "start",
    "--smp=1",
    "--reserve-memory=0M",
    "--overprovisioned",
    "--node-id=0",
    "--kafka-
addr=PLAINTEXT://0.0.0.0:29092,OUTSIDE://0.0.0.0:9092",
    "--advertise-kafka-addr=PLAINTEXT://
redpanda:29092,OUTSIDE://localhost:9092"
  ]
```

Application boilerplate

```
pub struct SyncingProcessesRegistry {
    to_table_registry: HashMap<..., JoinHandle<...>>,
    to_topic_registry: HashMap<String, JoinHandle<()>>,
    kafka_config: RedpandaConfig,
    clickhouse_config: ClickHouseConfig,
}

impl SyncingProcessesRegistry {

   pub fn new(
      kafka_config: RedpandaConfig,
      clickhouse_config: ClickHouseConfig) -> Self {
        Self {
            ...
        }
    }
```

Environment boilerplate

```
docker-compose.yml
version: '3.8'

services:
  redpanda:
    image: vectorized/redpanda:latest
    container_name: redpanda
    ports:
      - "9092:9092"  # Kafka API port
      - "9644:9644"  # Admin API port
    command:
      - redpanda
      - start
      - --overprovisioned
      - --smp 1
      - --memory 1G
      - --reserve-memory 0M
```

Some Fun Moose Features

Data Models

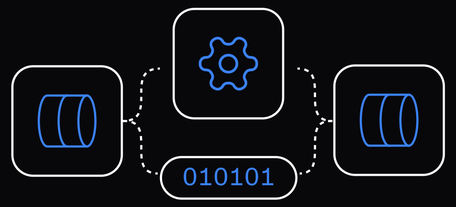
```
/datamodels/events.ts

interface RawEvent {
  id: Key<string>
  sessionId: string
  timestamp: string
 }

interface Event {
  eventId: Key<string>
  sessionId: string
  utc: Datetime
 }
```

Data Model primitive

High performance ingestion pipeline generated from simple data schemas

- ✓ Defined in pure Python/Typescript syntax
- ✓ Type-safe ingest API powered by Rust
- ✓ Managed data sync from Redpanda to ClickHouse

Functions



Function primitive

Auto-orchestrated stream processing for incoming data

- ✓ Strongly typed
- ✓ Just regular Typescript/Python functions
- ✓ Leverage third-party modules or APIs for rich data processing

Local-first Developer Experience

Moose-cli Dev

Run your end-to-end data infrastructure and application locally

- ✓ Instantly apply latest changes to local infrastructure
- ✓ Work within your IDE
- ✓ Real-time observability for local data workloads

Example F45 Data & Data Models

JSON

Session data

```
"workoutlogo": "https:\/\/f45tv.cdn.f45.com\/lionH
"dropoutminutescounter": "1_minutes 2_minutes ",
"totalusersforsession": "21",
"lionheartstart": "172",
"data": [
    {
        "max": "172",
        "averagebpm": 100,
        "bpmmax": 131,
        "totalpoints": "32.9",
        "name": "Jeff",
        "columns": {
            "calories": [ …
            ],
            "time": [ …
            ],
            "points": [ …
            ],
            "percent": [ …
            ],
            "certainty": [ …
            ],
            "bpm": [ …
            ]
        }
```

Moose Data Model

Raw Session Activity

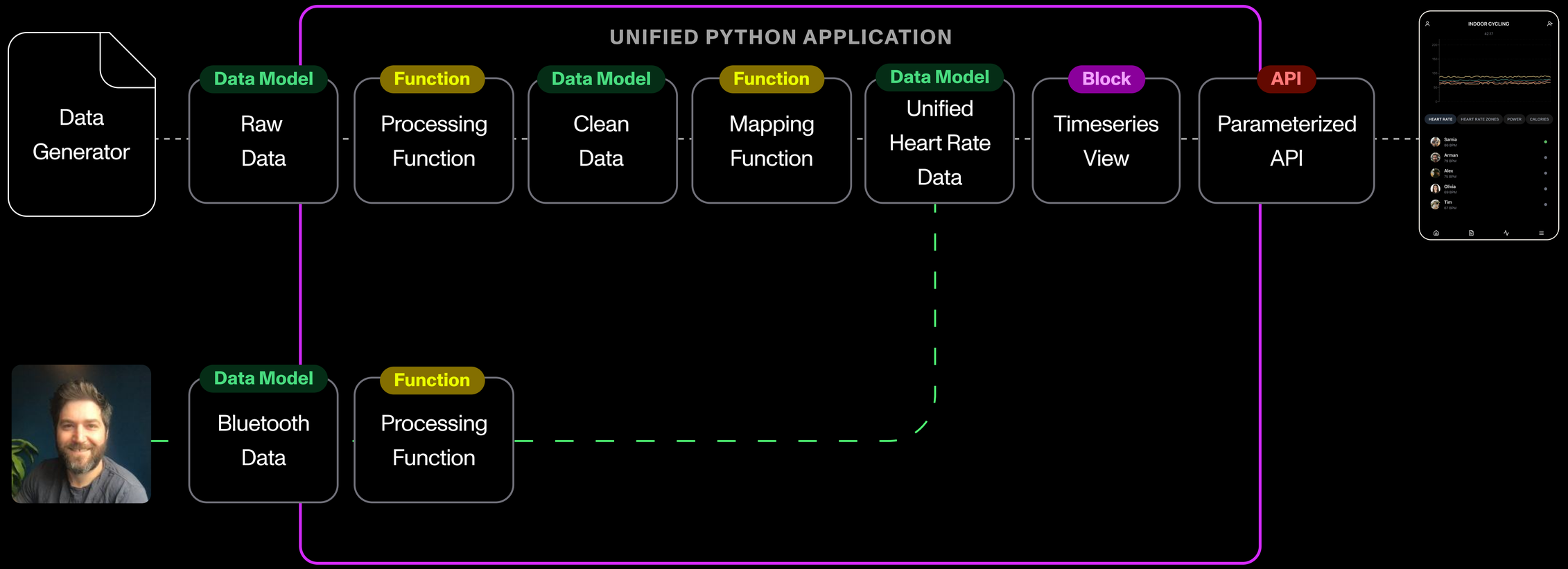
```
import { Key } from "@514labs/moose-lib"

export interface RawSessionActivityPayload_v1 {
    // Inject field for the filename in aws
    file_name: Key<string>;
    workoutlogo: string;
    dropoutminutescounter: string;
    totalusersforsession: string;
    lionheartstart: string;
    data: {
        max: string;
        averagebpm: number;
        bpmmax: number;
        totalpoints: string;
        name: string;
        columns: {
            calories: number[];
            time: number[];
            points: number[];
            percent: number[];
            certainty: number[];
            bpm: number[];
        };
        totalpointsnum: number;
        totalcalories: number;
```

Moose Data Model

Clean Session Activity

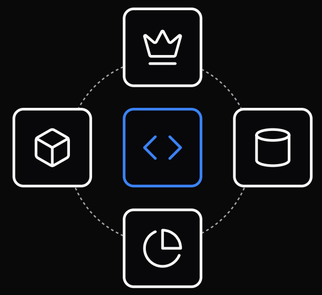
```
export const SessionActivityConfig: DataModelConf
    storage: {
      enabled: true,
      deduplicate: true, // uses sessionID_hash t
      order_by_fields: ['user_id', 'workout_date'
    },
};

export interface SessionActivity {
    sessionID_hash: string
    file_name: string;
    workout_logo: string;
    total_users_for_session: number;
    //  seen in V1 not in V2: dropoutminutescount
    lionheart_start: number;
    data_total_points: number;
    data_name: string;
    data_columns_time: number[];
    data_columns_points: number[];
    data_columns_percent: number[];
    data_columns_certainty: number[];
    data_columns_bpm: number[];
    data_columns_calories: number[];
    data_total_calories: number;
```

Let’s add some live data!!



Fiveonefour Products

Open Source Framework



Moose

Public Beta

Open source data engineering framework for data products and pipelines

- ✓ Production data eng in Py / TS
- ✓ Intuitive abstractions
- ✓ Local-first dev experience

Commercial Hosting

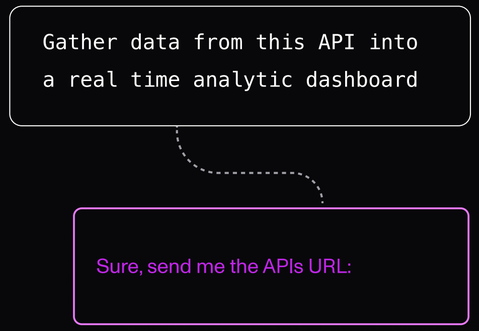
Boreal Cloud

Private Beta

Quickly, securely and scalably get your data-intensive application into production

- ✓ Managed infrastructure
- ✓ Secure & compliant
- ✓ One-click deploy

AI Data Engineer

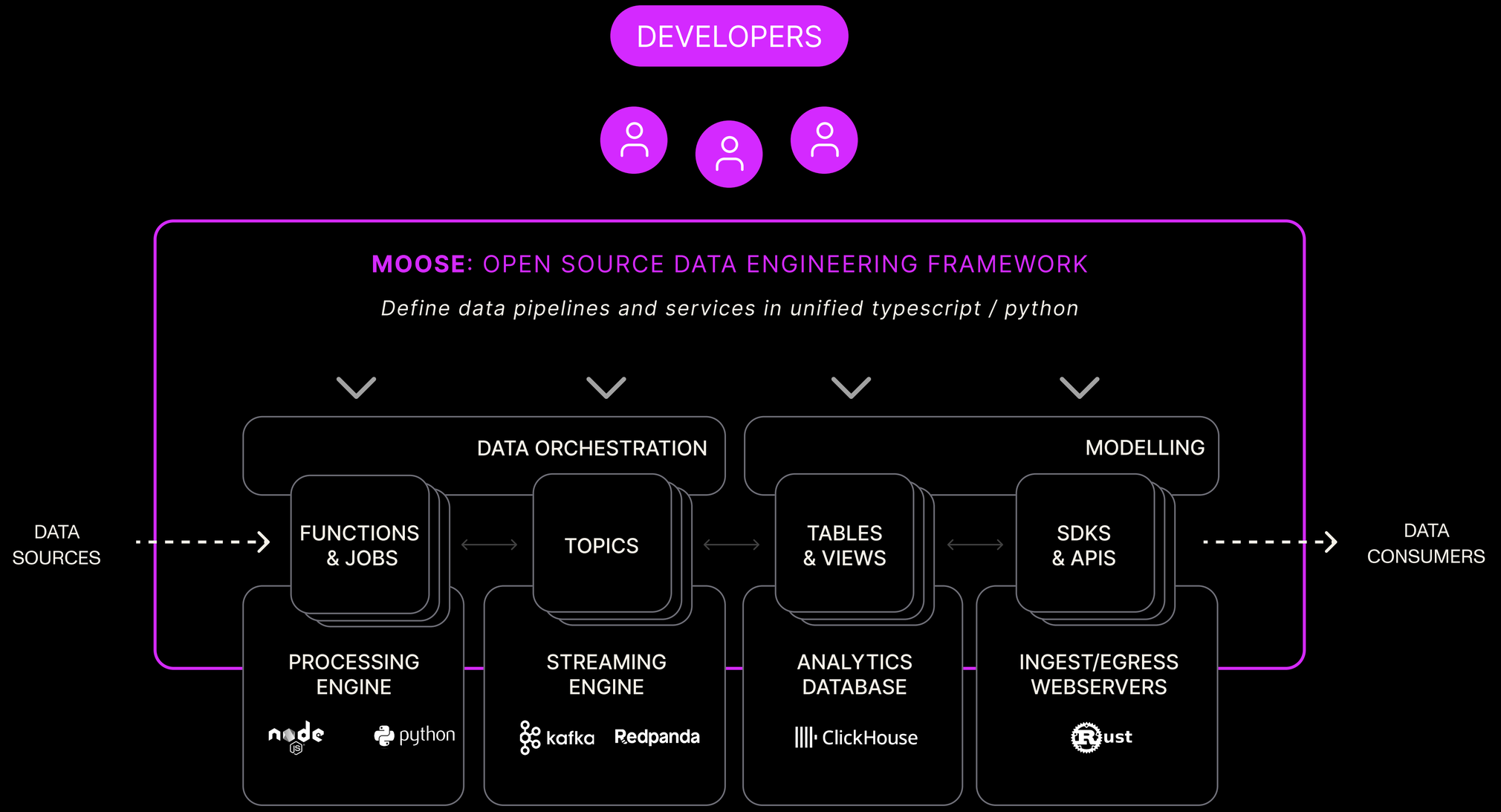


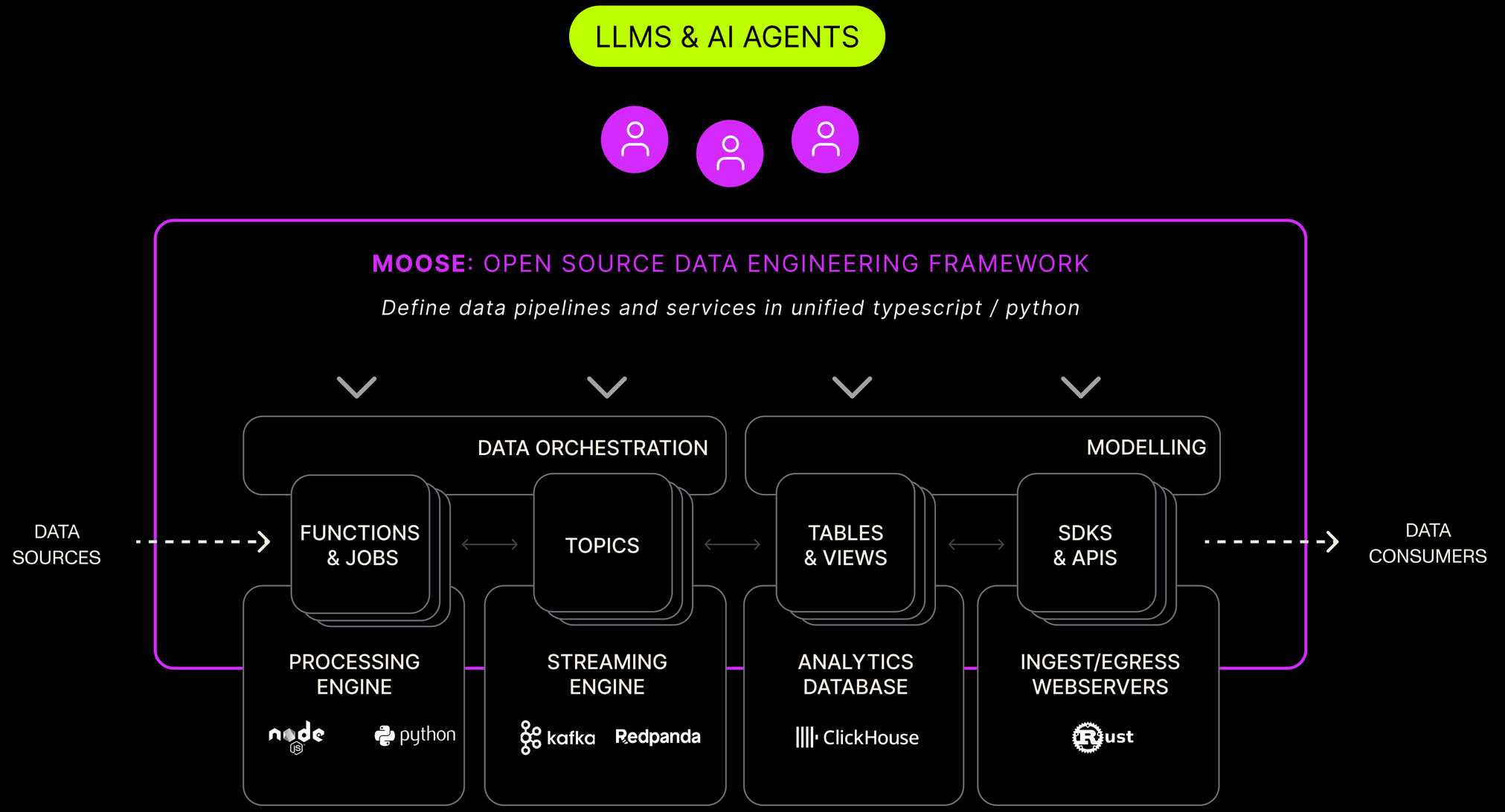
Aurora AI

Pre-Alpha

AI agents automate the repetitive and tedious parts of data engineering

- ✓ Context gathering
- ✓ Code generation
- ✓ Project management


DEVELOPERS
MOOSE: OPEN SOURCE DATA ENGINEERING FRAMEWORK
Define data pipelines and services in unified typescript / python
DATA ORCHESTRATION
MODELLING
DATA SOURCES
FUNCTIONS & JOBS
TOPICS
TABLES & VIEWS
SDKS & APIS
DATA CONSUMERS
PROCESSING ENGINE
node
python
STREAMING ENGINE
kafka
Redpanda
ANALYTICS DATABASE
ClickHouse
INGEST/EGRESS WEBSERVERS
Rust


LLMS & AI AGENTS
MOOSE: OPEN SOURCE DATA ENGINEERING FRAMEWORK
Define data pipelines and services in unified typescript / python
DATA ORCHESTRATION
MODELLING
DATA SOURCES
FUNCTIONS & JOBS
TOPICS
TABLES & VIEWS
SDKS & APIS
DATA CONSUMERS
PROCESSING ENGINE
node
python
STREAMING ENGINE
kafka
Redpanda
ANALYTICS DATABASE
ClickHouse
INGEST/EGRESS WEBSERVERS
Rust

Thank you!

getmoose.dev

github.com/514-labs/moose