DATA SCIENCE @ SCALE

VISUAL DESIGN WITH SERVERLESS SPARK ON KUBERNETES

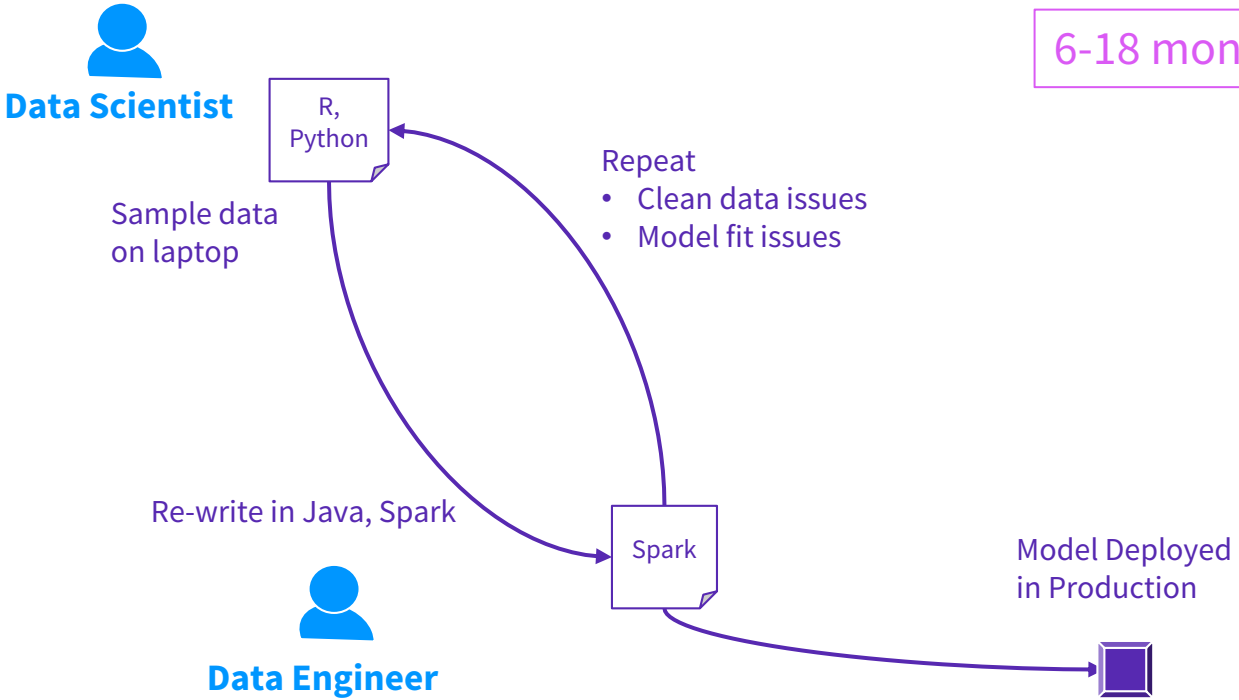
Raj Bains
CEO SimpleDataLabs
raj.bains@simpledatalabs.com

For Modelers, Data Engineers & Analysts

Agenda

1. Data Science at Scale issue
2. POV – Integrated Design & Execution
3. DEMO – Data Science via Integration
4. Resolution
    1. Data Science at scale
    2. ETL in the cloud
5. Building Solution on AWS/Kubernetes
    1. Technology choices
    2. Application Architecture
    3. Dynamic Clusters
6. Conclusion
    1. Skills
    2. Business Value

Problem: ML model to market time



POV: Move beyond Data Lakes

On-Premise

ETL Tool

BI Tool

Advanced Analytics

EDW

COMPUTE

STORAGE

On-Premise Swamp

Scripts

ML Notebooks

HADOOP LAKE 3

HADOOP LAKE 2

HADOOP LAKE 1

COMPUTE

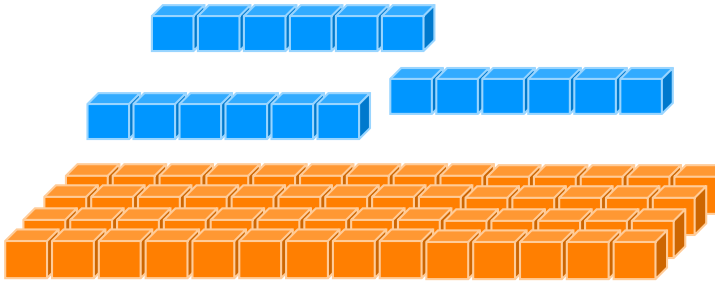
STORAGE

Cloud Serverless

- Single Design Platform
- Single Metadata Catalog
- Sharing and Collaboration

ETL | BI | ML | Notebook

Spark Ephemeral Compute

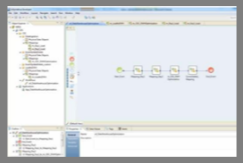


Unlimited Cloud Storage

Ancient History: Tools on EDW

Data Engr.

Informatica,
Ab Initio
Talend



- Visual Data Pipelines
- Data Catalog
- Lineage

Data Analyst

Excel,
Tableau,
Microstrategy

- Interactive Exploration
- Star Schema
- Cube Definition

Data Scientist

Knime,
Alteryx,
RapidMiner

- Data Exploration
- Visual Analytics Pipelines

Operational Data

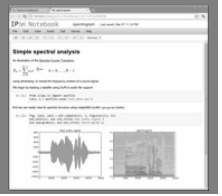
ETL BI ~~ML~~

SQL EDW

Recent History: Scripts on Big Data

Data Engr.

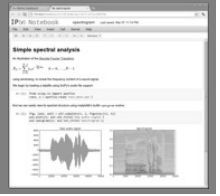
Notebook, Scripts



- Visual Data Pipelines
- Data Catalog
- Lineage

Data Analyst

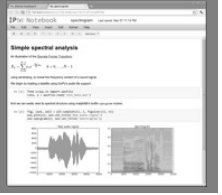
Notebook, Scripts



- Interactive Exploration
- Star Schema
- Cube Definition

Data Scientist

Notebook, Scripts



- Data Exploration
- Visual Analytics Pipelines

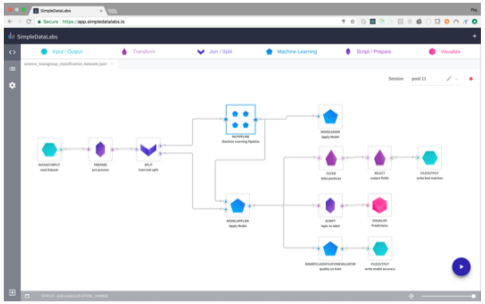
Operational Data

Streaming Data

ETL BI ML

APACHE SPARK

Future: Collaborative Designer



Data Engr.

- Visual Data Pipelines
- Data Catalog
- Lineage

Data Analyst

- Interactive Exploration
- Star Schema
- Cube Definition

Data Scientist

- Data Exploration
- Visual Analytics Pipelines

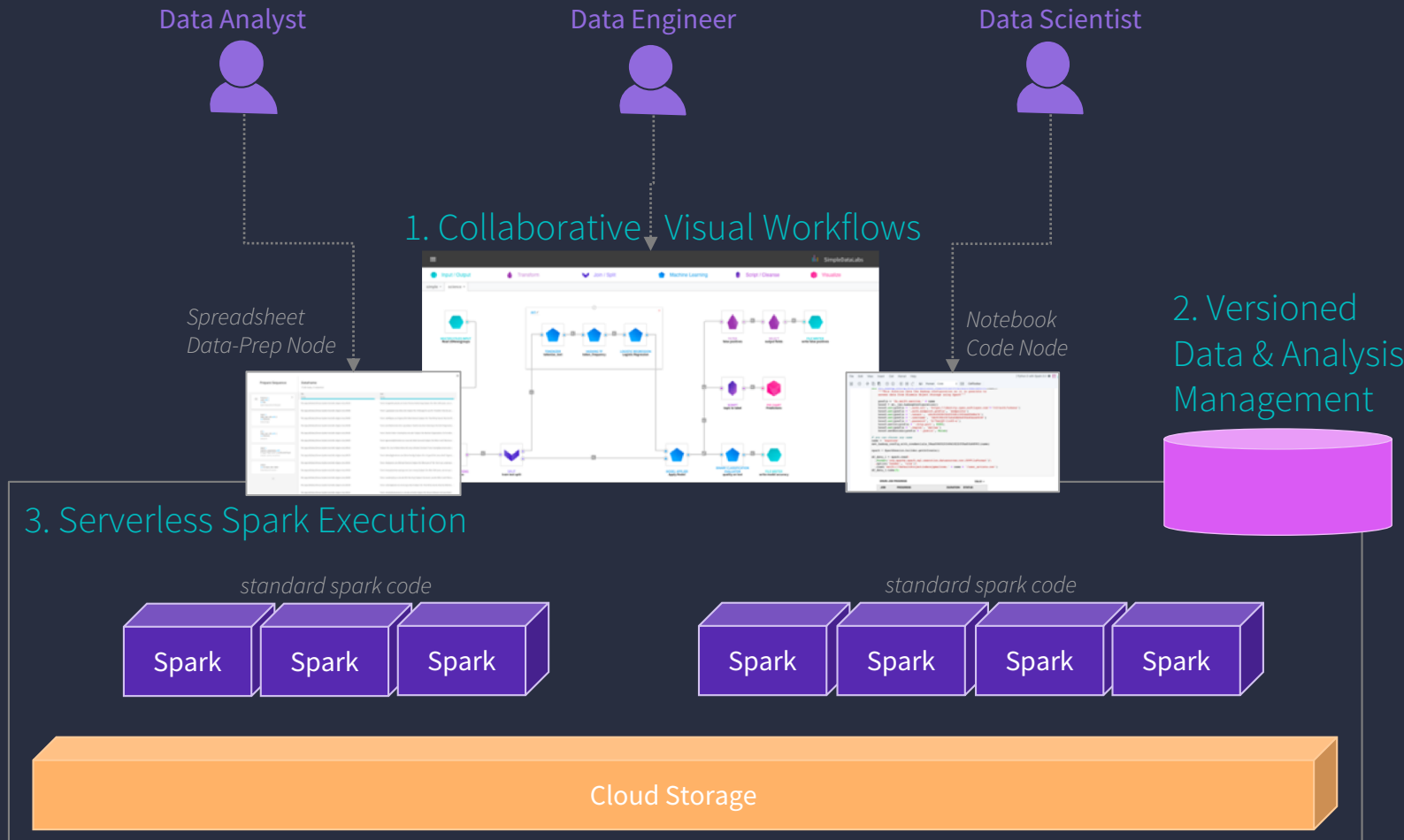
Operational Data

Streaming Data

ETL BI ML

APACHE SPARK

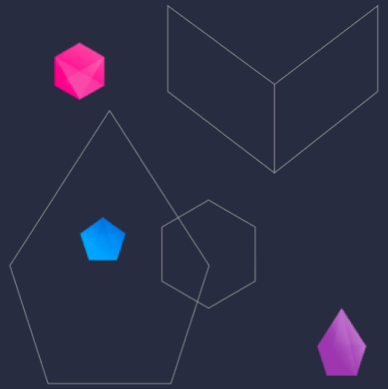
Features


Data Analyst
Data Engineer
Data Scientist
1. Collaborative Visual Workflows
Spreadsheet
Data-Prep Node
Notebook
Code Node
2. Versioned
Data & Analysis
Management
3. Serverless Spark Execution
standard spark code
standard spark code
Spark
Spark
Spark
Spark
Spark
Spark
Spark
Cloud Storage

DEMO TIME

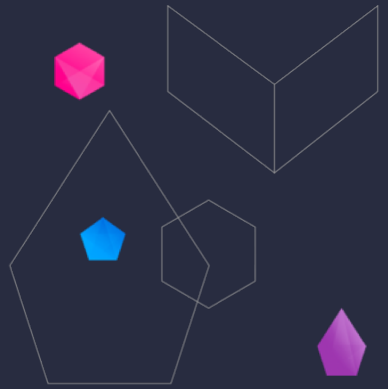
BUILDING A SCIENCE/NOT-SCIENCE CLASSIFIER

20 Newsgroup Dataset



ADVANCED ANALYTICS MARKET

WHERE DOES A SOLUTION LIKE THIS FIT?



Solving ML model to market time

6-18 months

Sample data on laptop

R, Python

Repeat
- Clean data issues
- Model fit issues

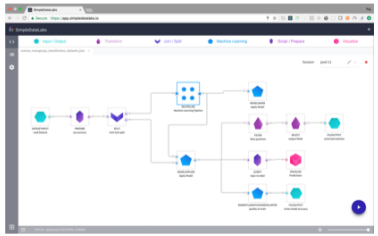
Re-write in Java, Spark

Spark

Data Engineer

Model Deployed in Production

Data Scientist



Few Weeks!

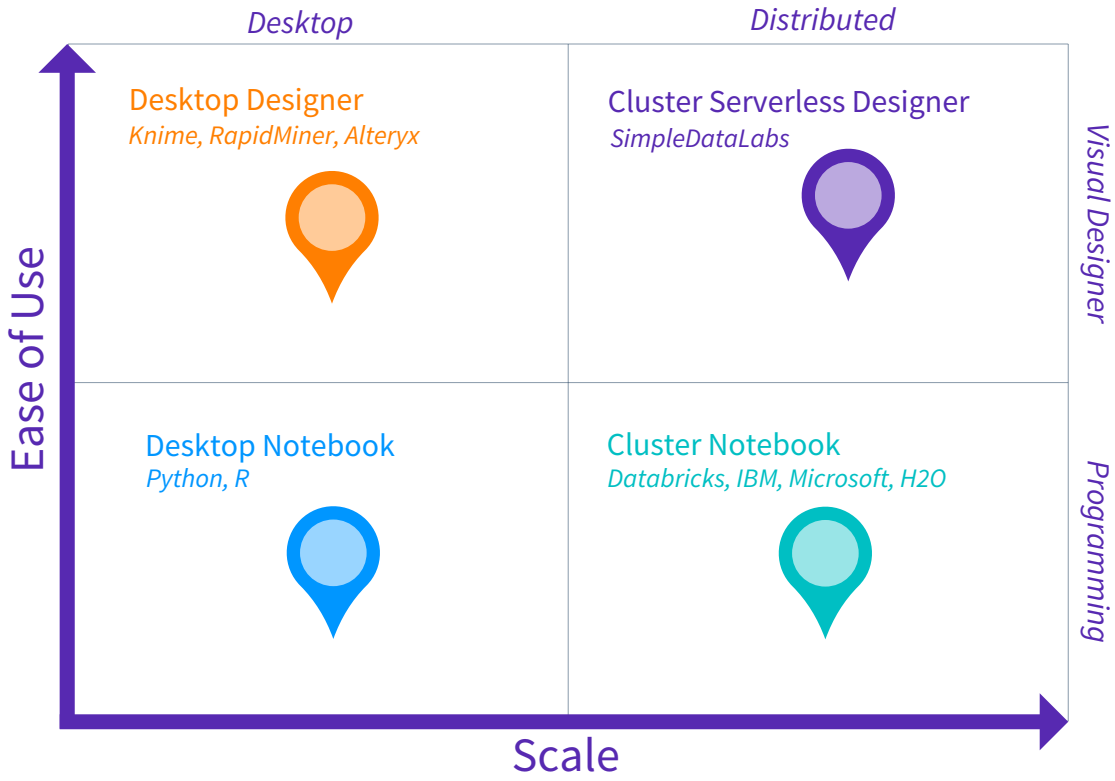
Start with large dataset
Build models visually or code
Model is correct the first time

One-click-deploy model

Integrate model with application

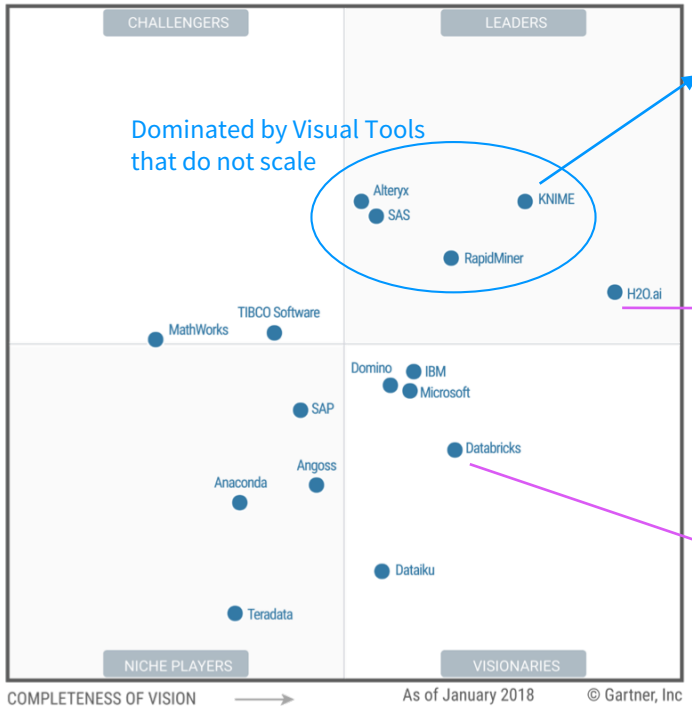
Data Engineer

Market Fit: Simplicity & Scale



Market Fit: Simplicity & Scale

Gartner Magic Quadrant 2018



Visual Tools Struggle with Scale

KNIME

Performance and scalability: Reference customers reported issues with large-scale deployments and performance on large datasets. A KNIME Server deployment is currently limited to a single host.

Scalable Platforms Struggle with Usability

H20

•Ease of use: **H2O.ai's toolchain is primarily code-centric. Although this typically increases flexibility and scalability, it impedes ease of use and reuse.**

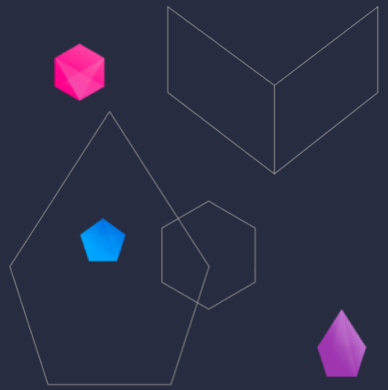
•Data preparation and interactive visualization: These capabilities are problematic for all code-centric platforms, of which H2O.ai's is one**. Nonetheless, H2O.ai's platform will prove challenging for clients expecting more interactivity and better, easier-to-use data ingestion, preparation and visualization capabilities**. Capabilities for the entire early part of the data pipeline are far less developed than the quantitative parts of H2O.ai's offerings.

DATABRICKS

Debugging capabilities: Most customers use Databricks for "do it yourself" machine learning. In addition to the debugging capabilities that Databricks already offers, reference customers wish the vendor could provide debugging features better suited to the needs of data scientists**. Databricks would also benefit from an integrated development environment (IDE)** with comprehensive facilities for enterprise-grade debugging, development and version control, in addition to the currently offered IDE on GitHub that leaves many reference customers dissatisfied.

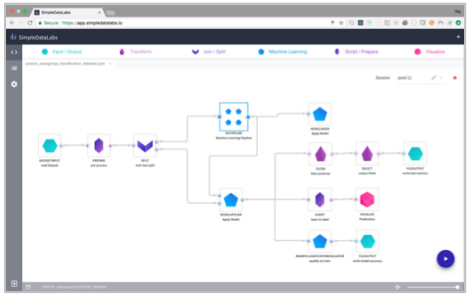
ETL MARKET

WHERE DOES A SOLUTION LIKE THIS FIT?

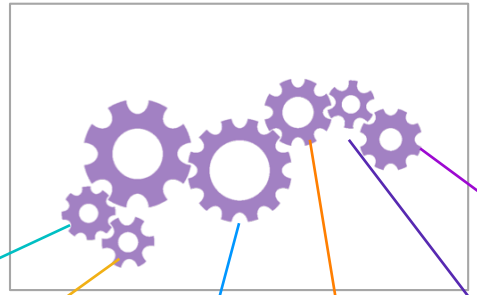


Solving ETL move to the cloud

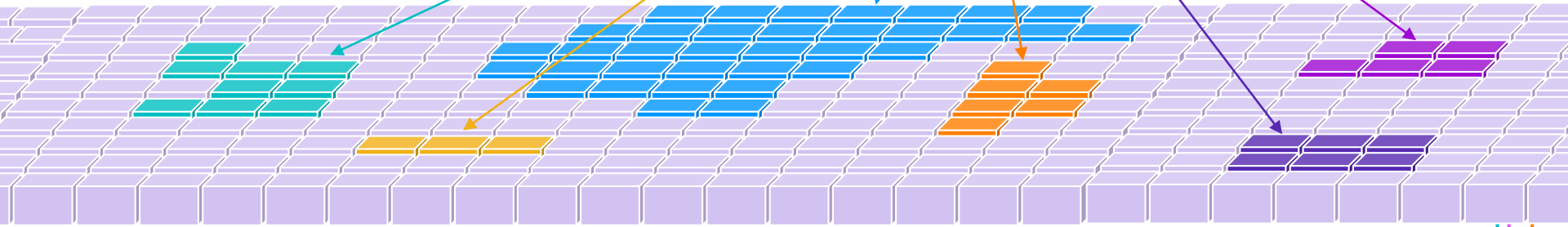
Design



Execution

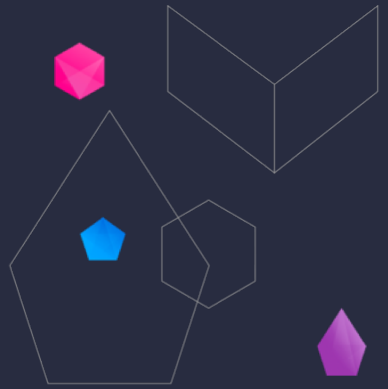


Dynamically Scaling
Many Spark Clusters
on Kubernetes

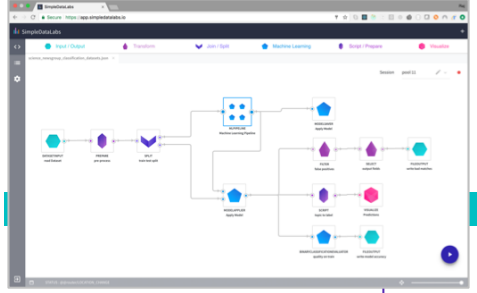


PRODUCT DESIGN

ARCHITECTING THE APPLICATION



SDL App Structure



React, Redux, GraphQL

Scala, Akka, Play Framework

SDL Application Layer

- App
- Codegen
- Execution
- Metadata
- Management

Spark Cluster Layer

- Notebook Kernel
- Spark Master
- Spark History
- Spark Worker
- Spark Worker
- Spark Worker

Google Kubernetes Orchestration

AWS

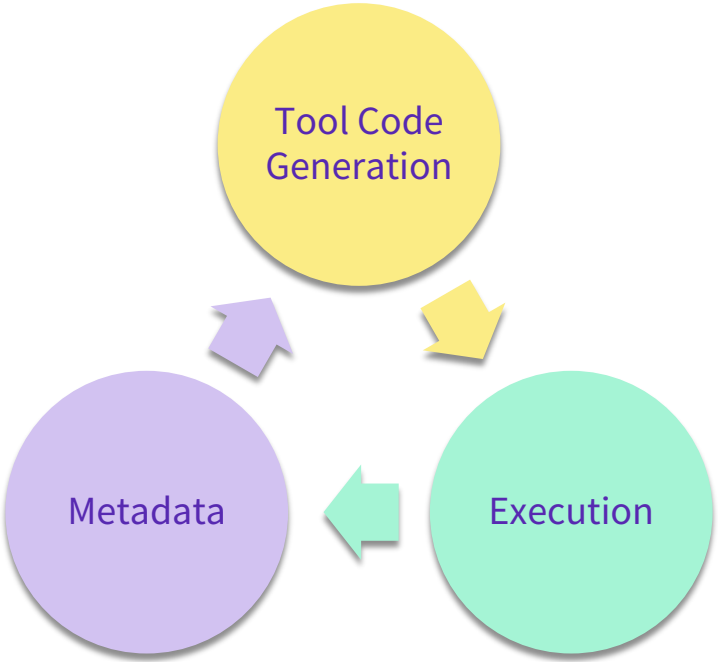
- RDS Postgres
- EFS
- S3

ON-PREM

GCE

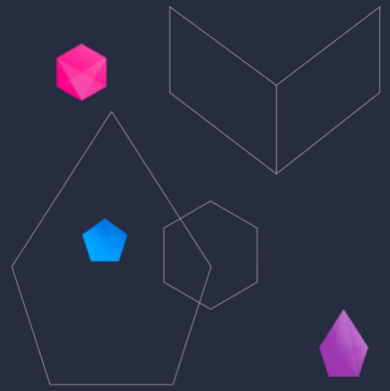
AZURE

Whole Stack - Tightly Coupled Core

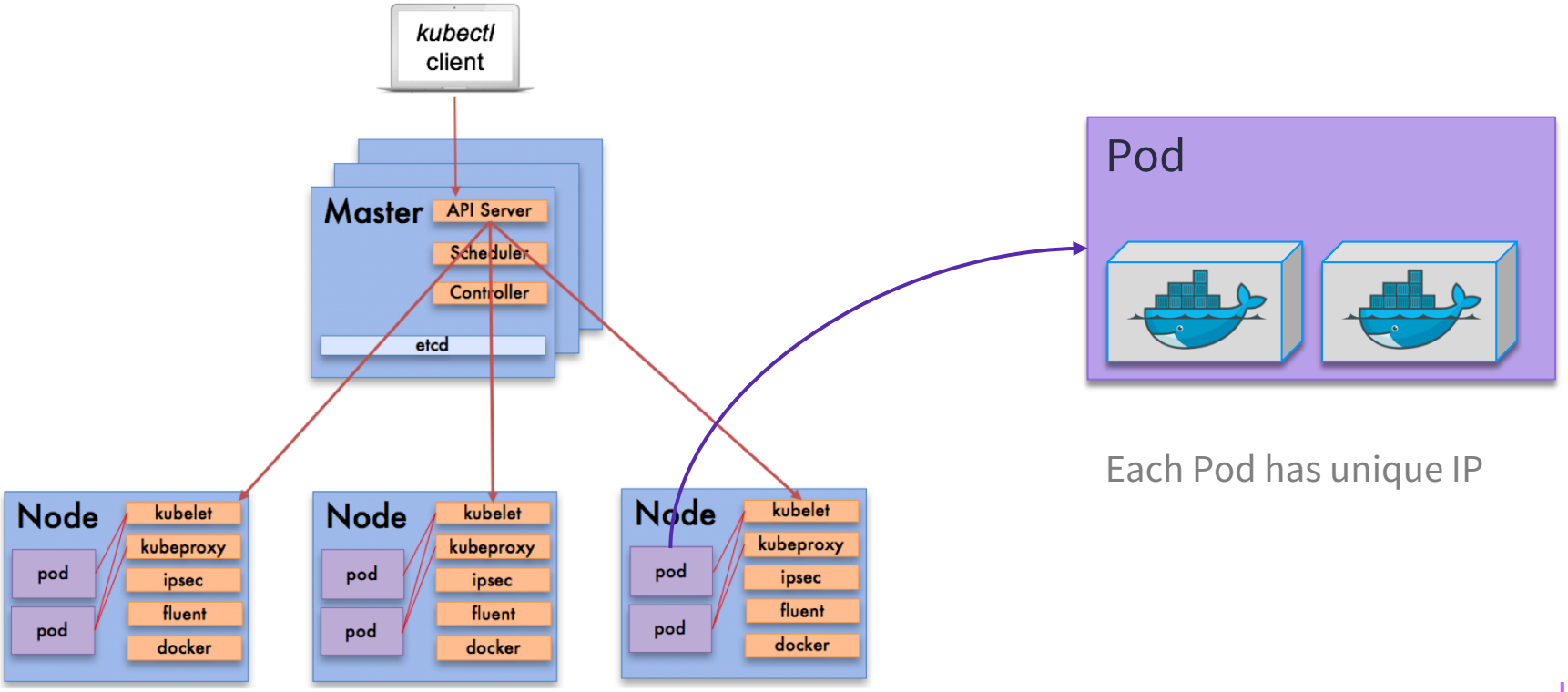


BUILDING ON KUBERNETES

Developing Serverless Spark



Kubernetes Architecture



Kubernetes Minimal Vocabulary

- Master, Node – Node types in Cluster
- Pods – Collections of Containers
- Deployment - Pod Spec
- Persistent Volumes
- Secrets
- Jobs
  - CronJob
- DaemonSet / Sidecar
- Scheduling
  - Labels
  - Taints

Kubernetes Deployment Spec

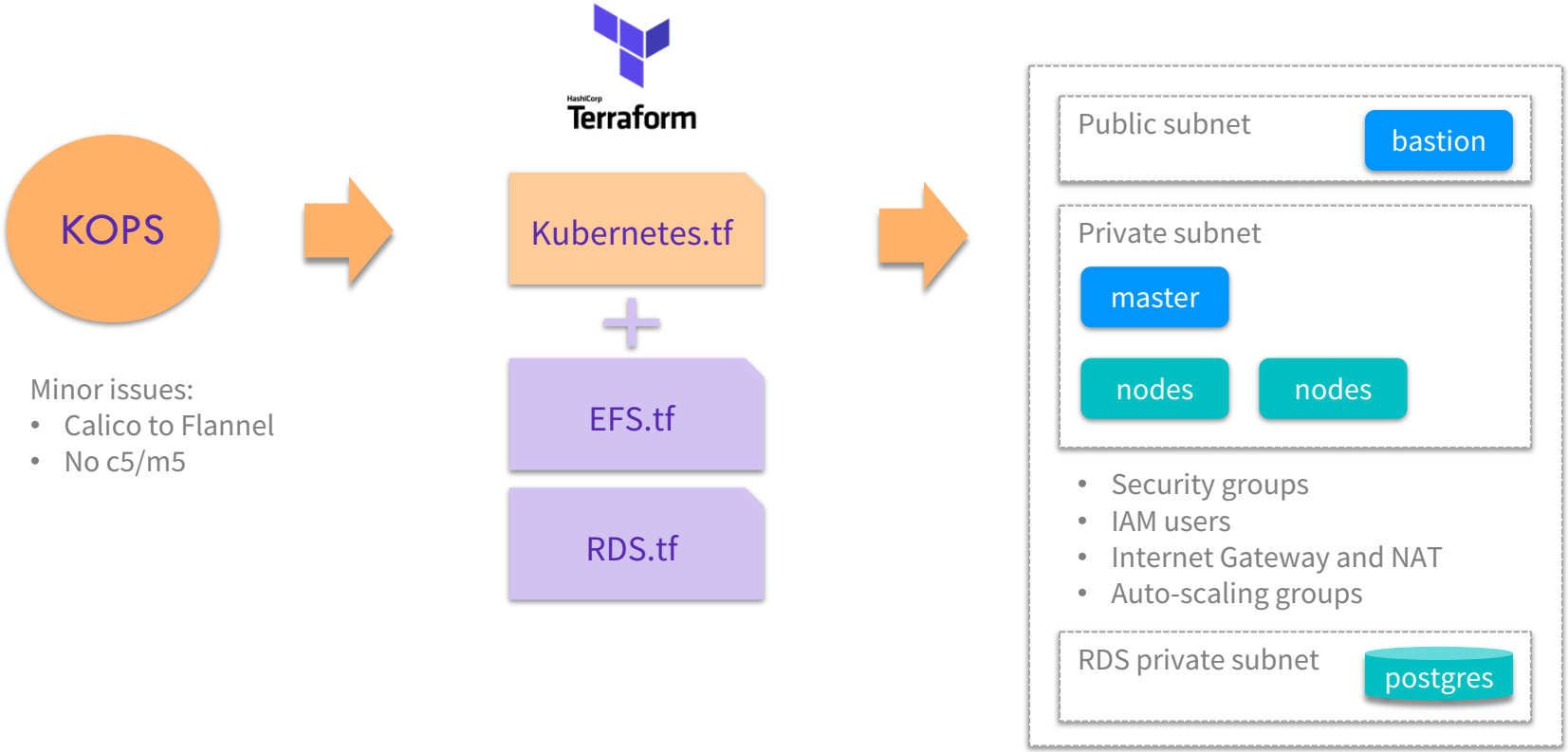
```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: spark-deployment
spec:
  replicas: 1
  template:
    metadata:
      name: spark-pod
      labels:
        component: spark-pod
    spec:
      containers:
        - name: sparkmaster
          image: {{ spark_image_repository_uri }}
          ports:
            - containerPort: 8081
            - containerPort: 7077
            - containerPort: 6066
          command: ["/start-master.sh"]
          args: ["7077", "8081"]
```

```
- name: sparkworker1
  image: {{ spark_image_repository_uri }}
  ports:
    - containerPort: 8082
    - containerPort: 7078
  command: ["/start-worker.sh"]
  args: ["spark://spark-master:7077", "8082"]
  volumeMounts:
    - mountPath: /app/session
      name: session-volume-session-1
  resources:
    requests:
      cpu: 700m
      memory: 2048Mi
```

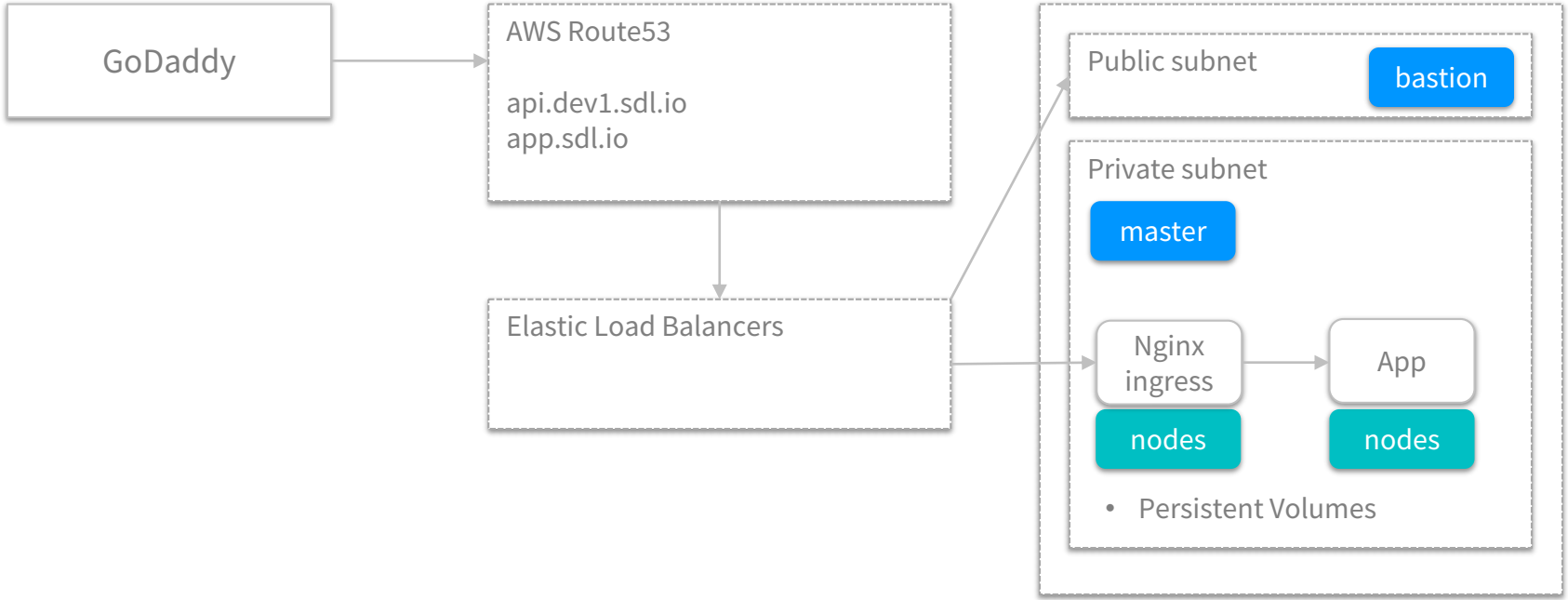
Which technology to use?

Deployment Tools	Ansible and Terraform
Deployment Strategy	Dynamic or Static or Mix
Orchestration	Mesos or K8s or Managed K8s
Base Technology	VM or Containers

Setting up K8s on AWS



Setting up DNS and Pods - Ansible

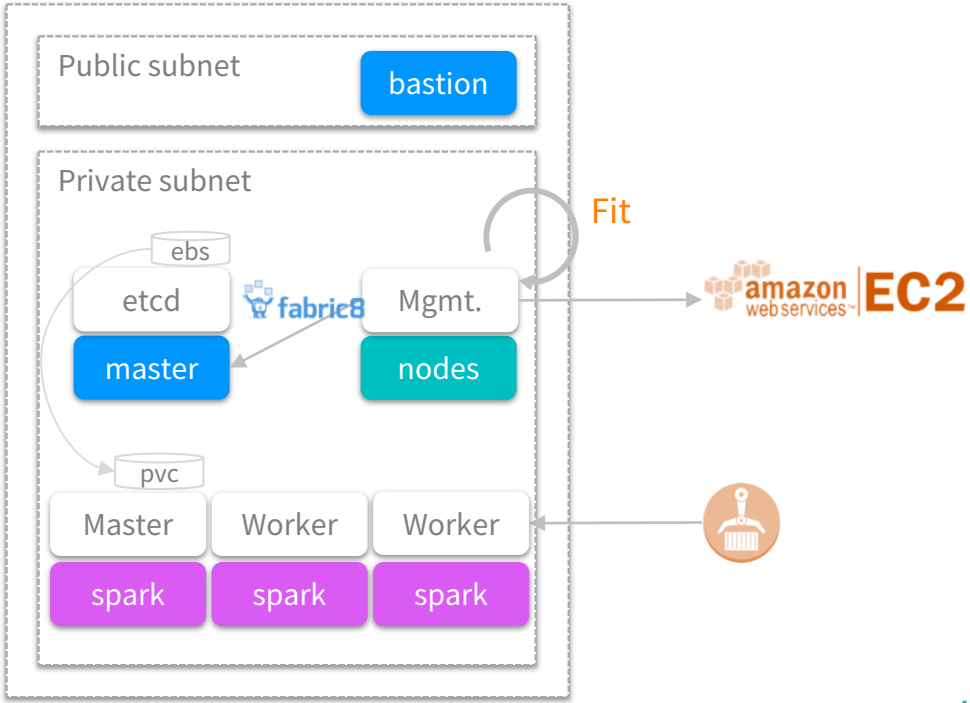


Dynamic cluster spin-up

Spinning up New Clusters

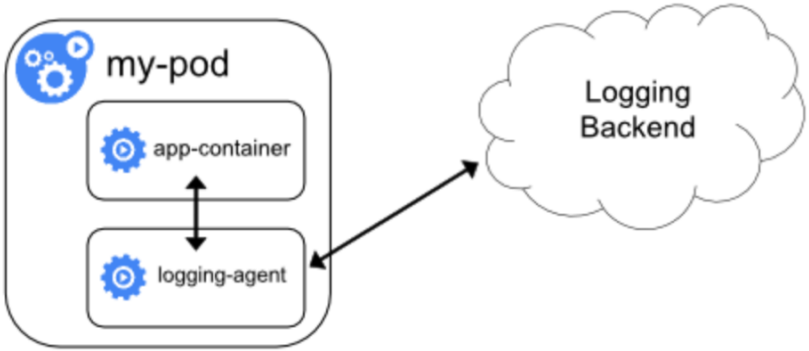
Latency of Spark Cluster spin-up

Passing Credentials to Workflows



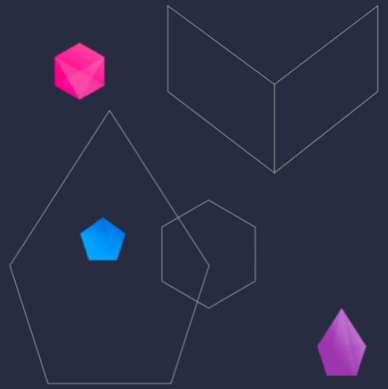
Logging, Monitoring

- Logging
  - Sidecar
- Monitoring
  - Prometheus
  - Separate for Spark

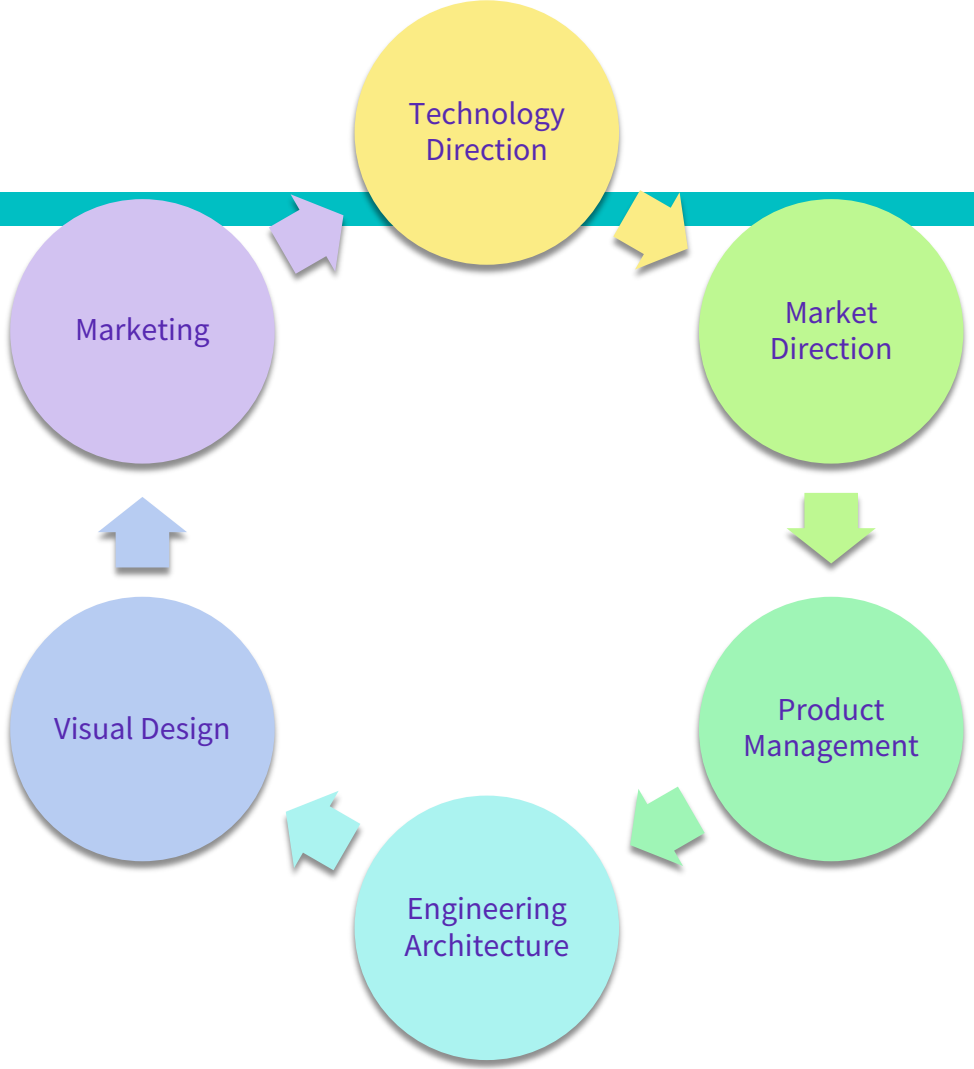


CONCLUSION

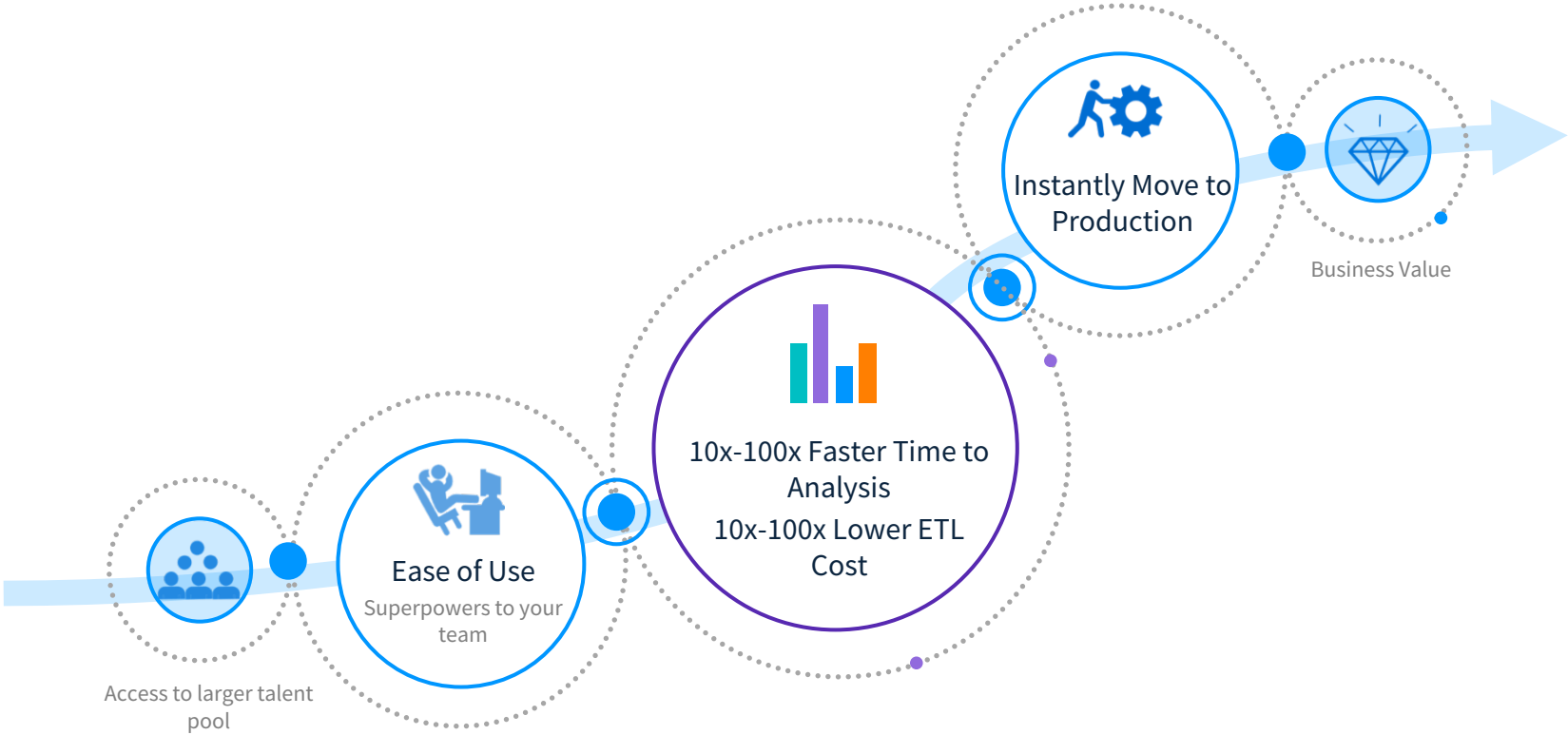
- SKILLS TO BUILD
- BUSINESS VALUE



Skills to build



Business Value and Impact



QUESTIONS