Architecting for Data Science

O'Reilly Software Architecture Conference
Boston, March 19, 2015

Johann Schleier-Smith
CTO, if(we)

@jssmith johann@ifwe.co github.com/ifwe


Data Science

Harvard
Business
Review

DATA

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

value from data

Alternative Definitions

extraction of knowledge from data

making discoveries in the world of big data

statistics + machine learning + scalable computation + visualization + computer science + business acumen + skilled communication

Related and Alternative Language

business intelligence

data mining

predictive modeling

forecasting

statistics

business reporting

knowledge extraction

analytics


value

Types of Value

understanding

product improvements

predictions

projections

revenue

new inspirations

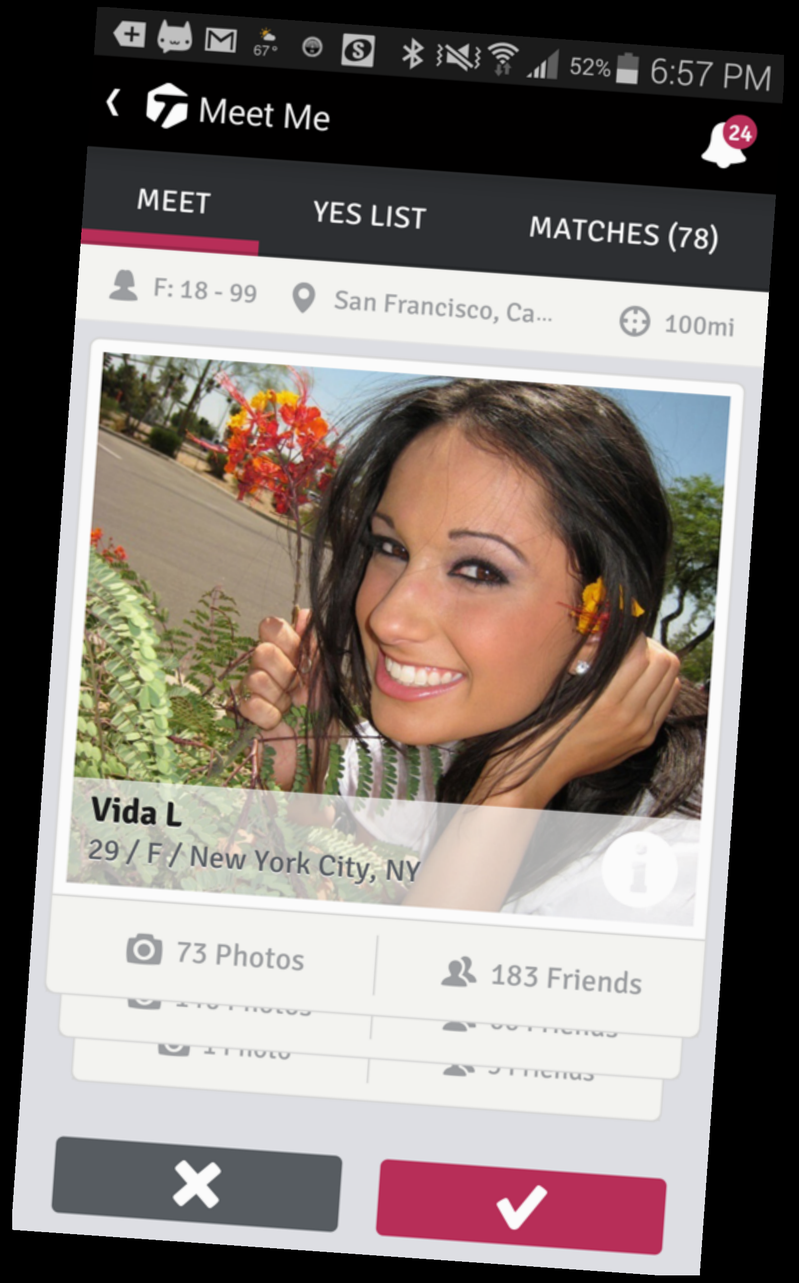
customer satisfaction

Today’s Examples

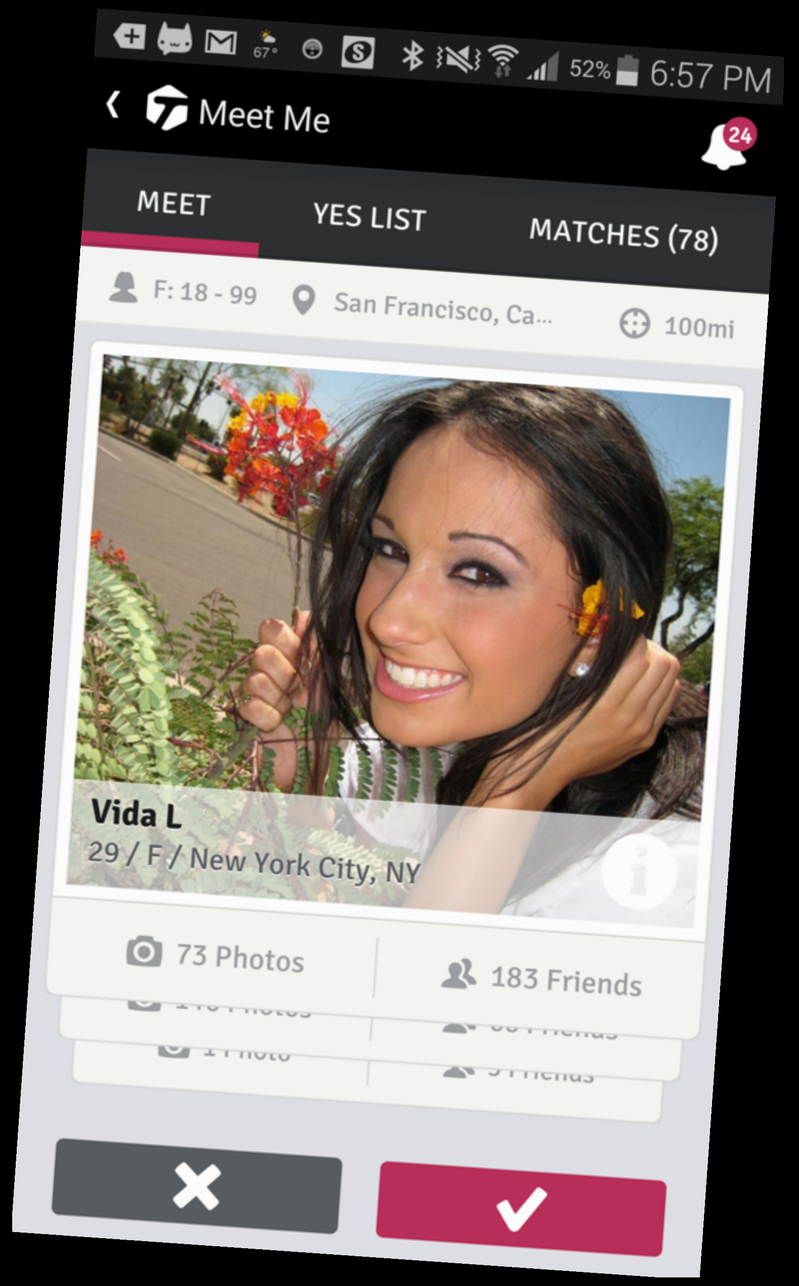
Recommendation engine for dating product

- >10 million candidates to draw from
- >1000 updates/sec

- Must be responsive to current activity
- Users expect instant query results

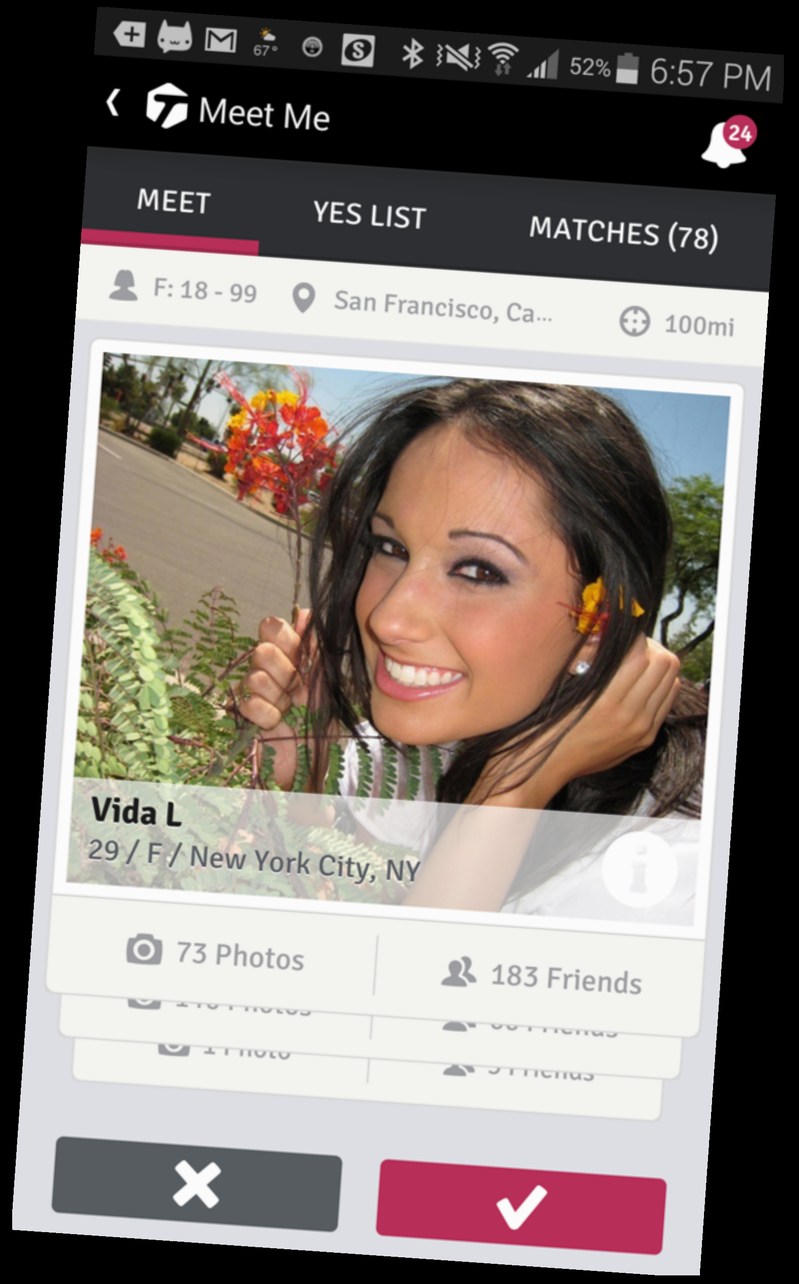


- Real-time is challenging
- Human behavior is complicated, especially in social context
- Previous interactions are perhaps our best hope for predicting future interactions



Value

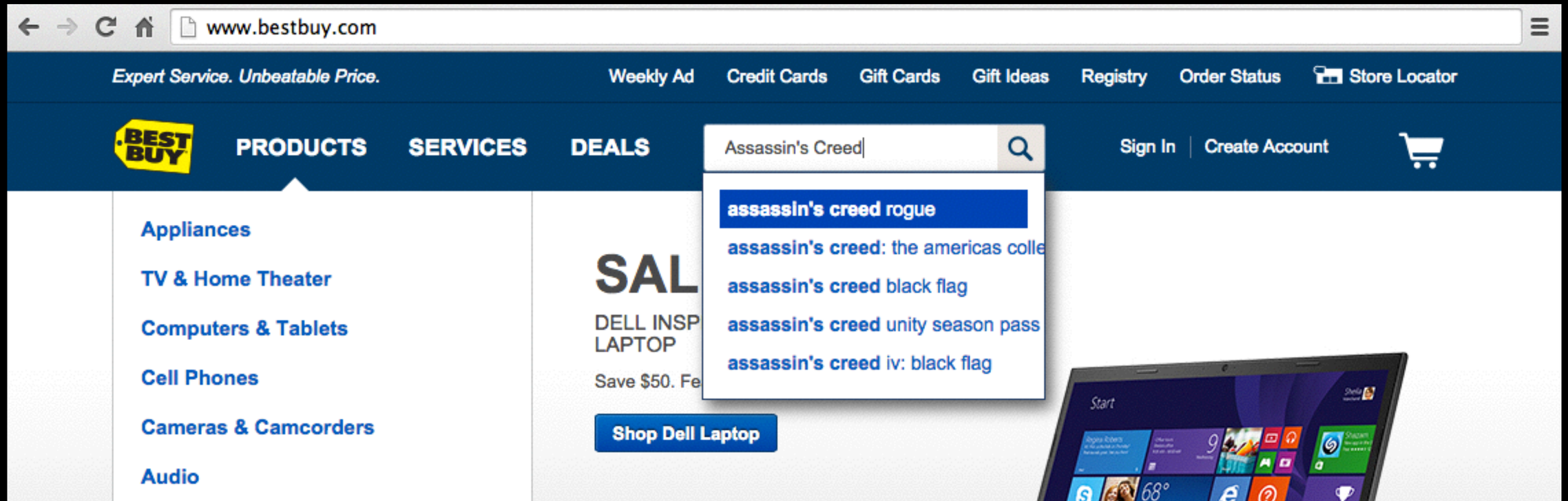
- Human connections ♥
- User engagement ecosystem
- Subscription and other revenues



Kaggle competition with Best Buy data

https://www.kaggle.com/c/acm-sf-chapter-hackathon-small

Kaggle competition with Best Buy data



Expert Service. Unbeatable Price.

Weekly Ad | Credit Cards | Gift Cards | Gift Ideas | Registry | Order Status | Store Locator

BEST BUY

PRODUCTS | SERVICES | DEALS

Search Best Buy

Sign In | Create Account

ASSASSIN'S CREED UNITY

Add to Cart

Assassin's Creed: Unity
Xbox One | SKU:5607008
$59.99

Add to Cart

Assassin's Creed: Unity
PlayStation 4 | SKU:5606027
$59.99

Add to Cart

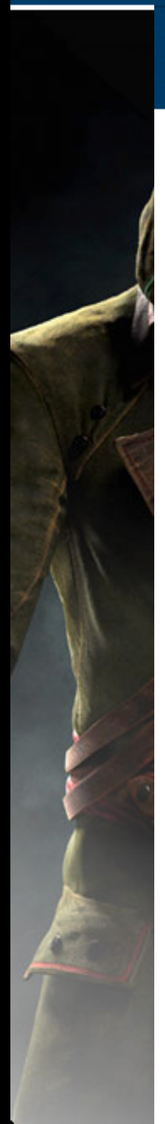
Assassin's Creed: Unity
PC | SKU:5608043
$59.99

Add to Cart

Assassin's Creed: Rogue
Xbox 360 | SKU:7843012
$59.99

Add to Cart

Assassin's Creed: Rogue
PlayStation 3 | SKU:7842013
$59.99



Expert Service. Unbeatable Price.
Weekly Ad | Credit Cards | Gift Cards | Gift Ideas | Registry | Order Status | Store Locator
PRODUCTS | SERVICES | DEALS
Search Best Buy
Sign In | Create Account

FREE SHIPPING on orders $35 and up. Learn more ›

Best Buy › All Categories

"xbox"

SHOP CATEGORY

- Video Games
- Xbox 360
- Xbox 360 Accessories
- Xbox One
- Video Game Accessories
- Xbox One Accessories
- Hard Drives & Storage
- Cables & Connectors

See All (23)

CURRENT OFFERS

- Free Shipping Eligible (2,282)
- Outlet Items (1,128)
- Special Offers (785)
- On Sale (305)
- Package Deals (3)

Featured Products

Microsoft - Xbox One 500GB Console Kinect Assassin's Creed: Unity Bundle
Model: 6RZ-00118
SKU: 8439023
Release Date: 11/02/2014
4.5 out of 5
$499.99

Microsoft - Xbox One Console - Black
Model: 7UV-00077
SKU: 9008182
Release Date: 11/22/2013
4.5 out of 5
$499.99

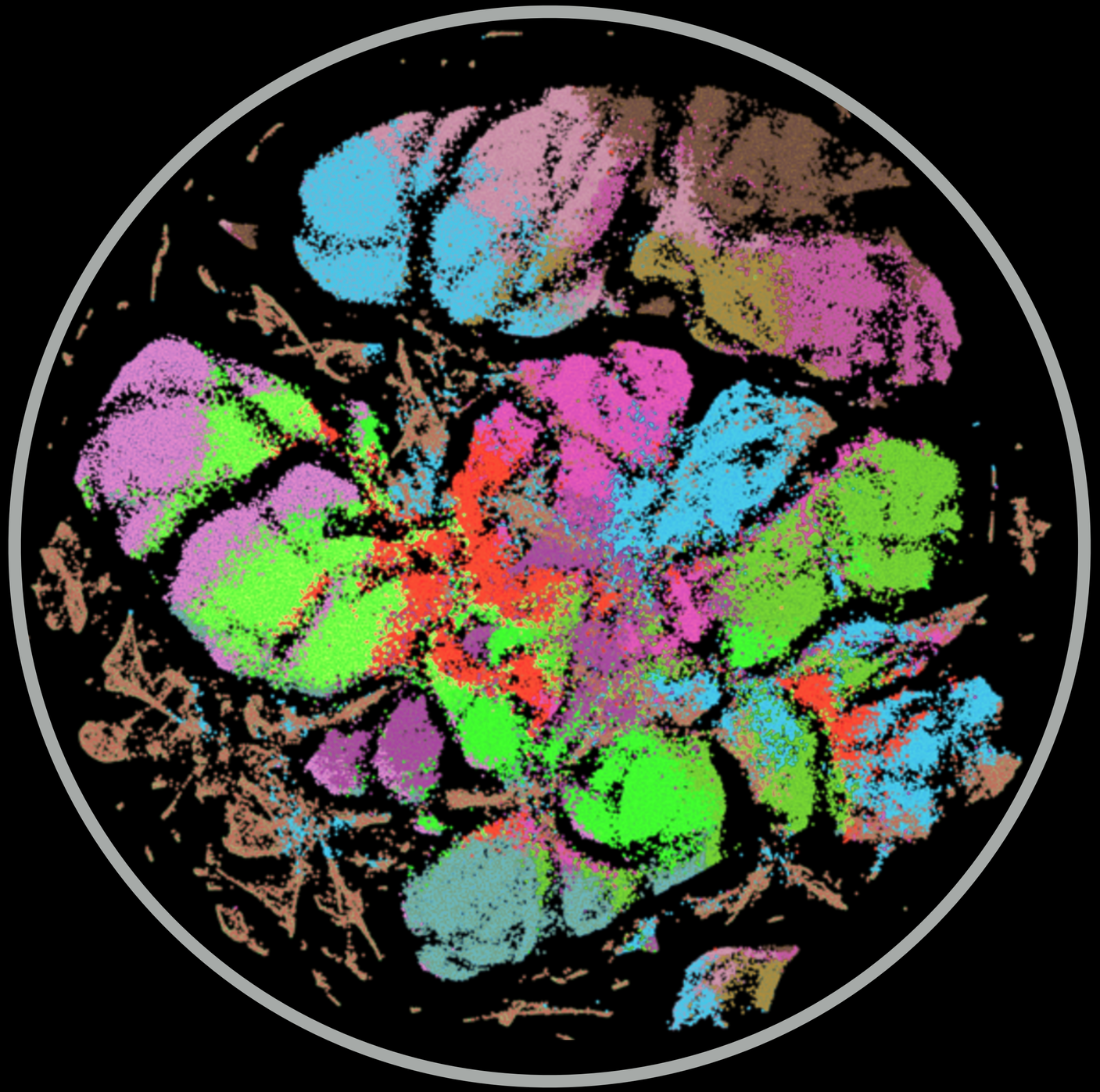
All Items (2,449) | Best Buy Items (2,036) | Marketplace Seller Items (413)

Sort by: Best Match | Items per page: 15

‹ 1 2 3 … 164 ›

Microsoft - Xbox One Console - Black
Model: 7UV-00077 | **SKU:** 9008182 | **Release Date:** 11/22/2013
Enjoy your entertainment in an all-new way
4.6 (3,175 Reviews)
Check Shipping & Availability

PRICE MATCH GUARANTEE
$499.99
Add to Cart
- Free Shipping on Orders $35



“outgoing and social (heavy messaging --- especially distant recipients and opposite gender, many outgoing comments, many friend requests to distant people), doesn’t play Pets much”

“receives many messages, active user, views many profiles, doesn't use meet me, sends many messages to distant people”

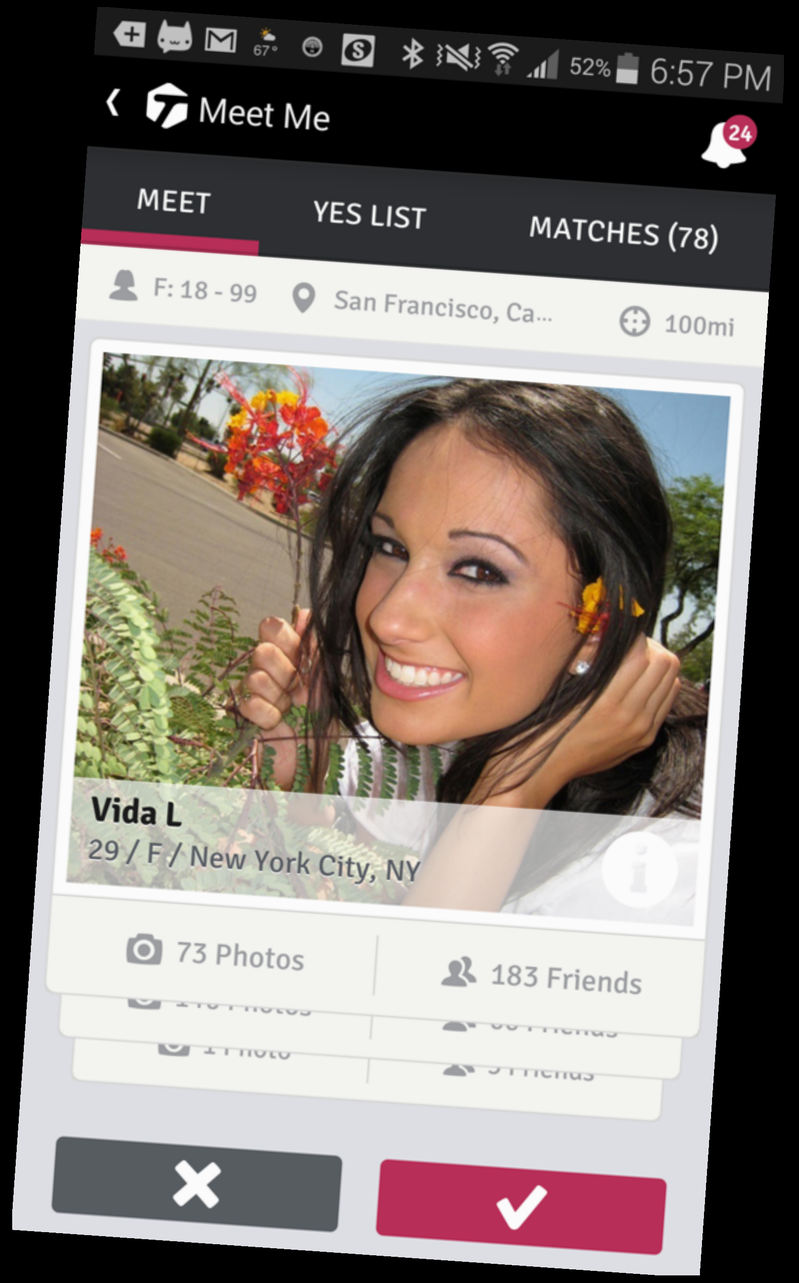
“heavy user overall, (pets, meet me, messaging)”


value


data

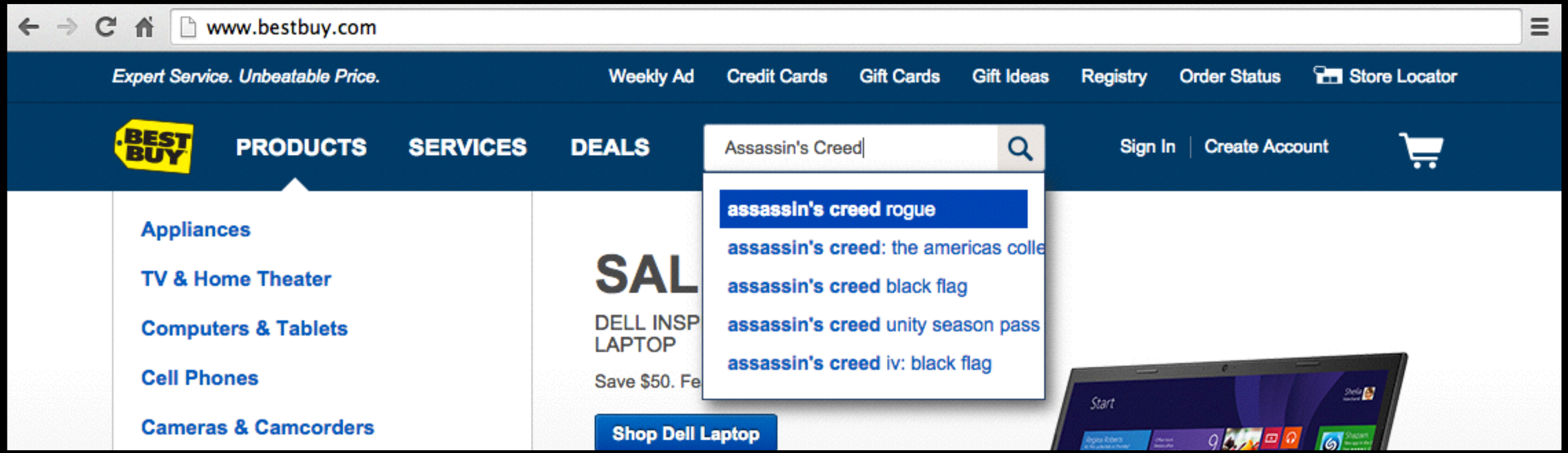
Dating product data

- Vote history
- Social interaction history
- Profile information



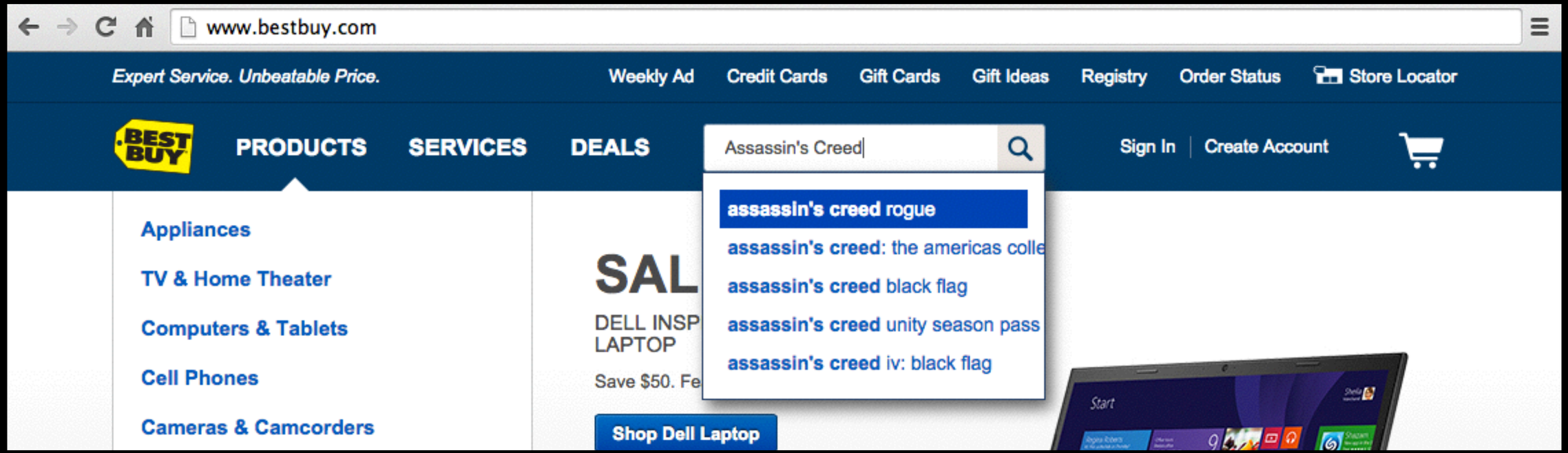
```
{
  "timestamp" : "2011-10-31 09:48:46",
  "query" : "Assassin's Creed",
  "skuSelected" : "2670133"
}
```

product views



product updates

```
{
  "sku" : "1032361",
  "regularPrice" : "19.99",
  "name" : "Need for Speed: Hot Pursuit",
  "description" : "Fasten your seatbelt and
   get ready to drive like your life depends
   on it..."
   ...
}
```



Formats for Data

log files

relational databases

xml files

unstructured documents

web services

spreadsheets

Types of Data

technical data

usage records

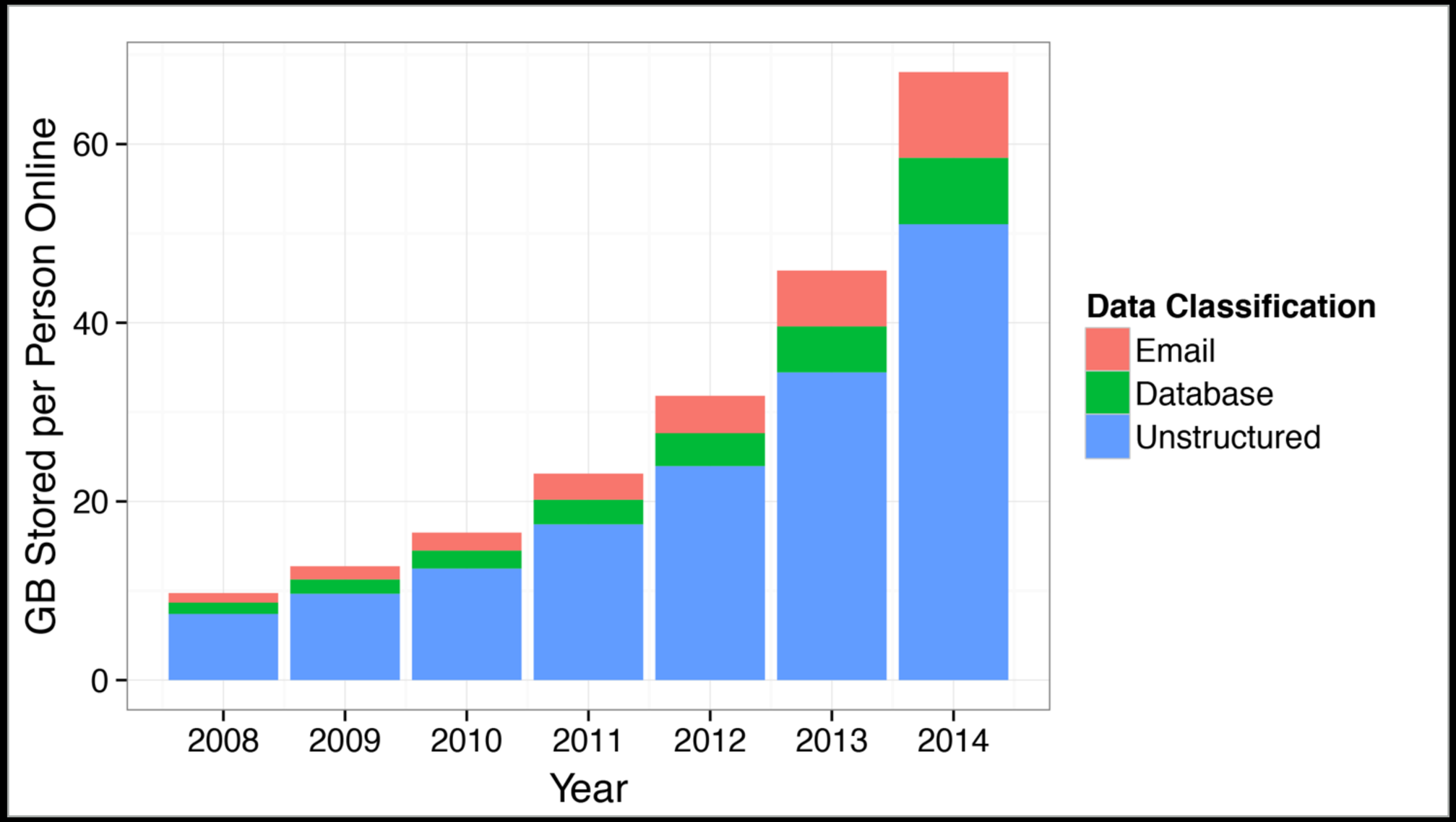
reference data

sensor data

government data

academic data

yet uncollected data



Vasant Dhar. 2013. Data science and prediction. Commun. ACM 56, 12 (December 2013), 64-73.
And International Telecommunication Union (ITU) and United Nations Population Division via www.internetlivestats.com/internet-users/

Trends

data quantity

data variety

data velocity

machine learning maturing

Machine Learning

classification

supervised methods

clustering

unsupervised methods

decision trees

what matters most is mapping of data

Machine Learning Techniques

- Classification - (Logistic Regression, Decision Trees, Random Forests)
- Prediction - (Generalized Linear Models, Support Vector Regression, …)
- Clustering - (K-Means, Hierarchical, Latent Dirichlet Allocation, …)
- Collaborative filtering, …

Features often matter more than choice of algorithm


data

tools of the trade

- Created in 1993
- Implementation of S language but also inherits from Scheme
- Object oriented code is possible but not encouraged
- Vast high-quality package ecosystem
- Data is vectors and *data frames*

Demo

Project: (None)

Untitled1* | mydata

Source on Save | Run | Source

```
1  mydata <- read.csv("~/demo/data-1.csv")
2  plot(mydata)
3  attach(mydata)
4  m <- lm(Y~X)
5  summary(m)
6  abline(m)
7
```

2:1 (Top Level) R Script

Environment | History

Import Dataset | Clear | List

Global Environment

Data

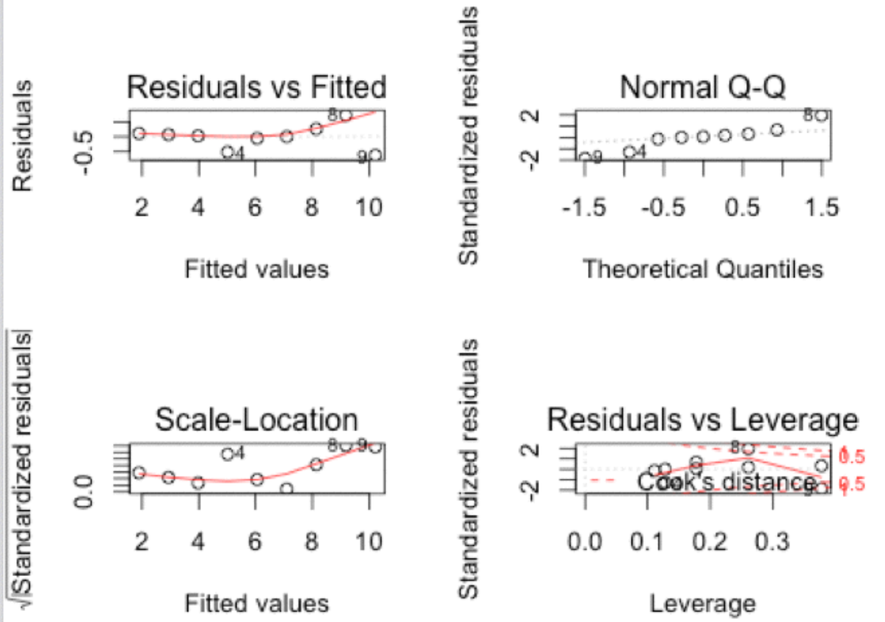
mydata	9 obs. of 2 variables

Values

m	List of 12

Files | Plots | Packages | Help | Viewer

Zoom | Export | Clear All



Console ~/

```
> plot(myData)
> par(mfrow(c(1,1)))
Error in par(mfrow(c(1, 1))) : could not find function "mfrow"
> par(mfrow=(c(1,1)))
> plot(myData)
> attach(myData)
The following objects are masked from myData (pos = 3):

    X, Y

> plot(X~Y)
> m <= lm(X~Y)
Error in m <= lm(X ~ Y) : comparison of these types is not implemented
> m <- lm(X~Y)
> summary(m)

Call:
lm(formula = X ~ Y)

Residuals:
```

- Statistics
- Visualization
- Machine learning
- REPL, scripts, interactive IDE
- In-memory data sets

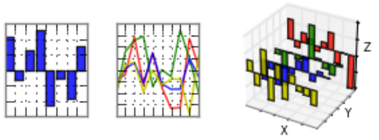
python™

IP[y]: IPython
Interactive Computing

scikit learn

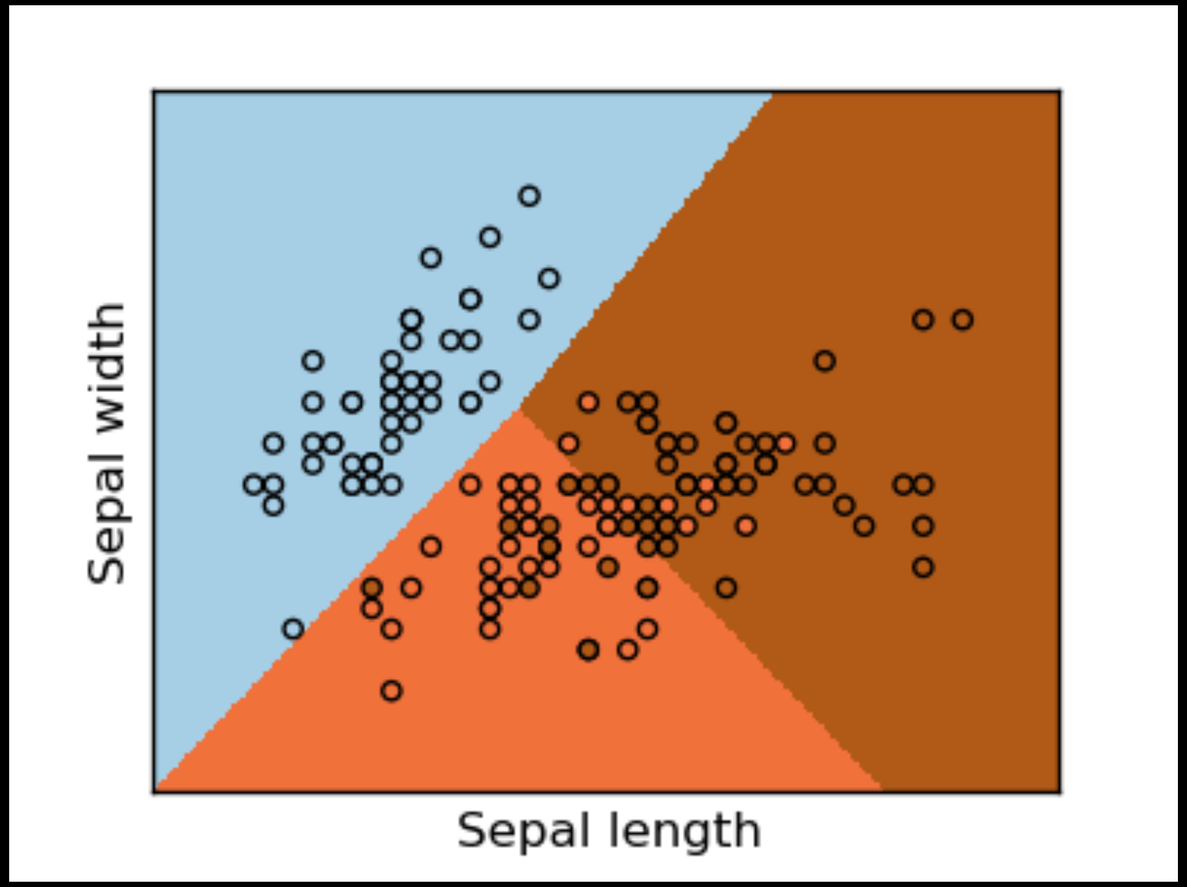
pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



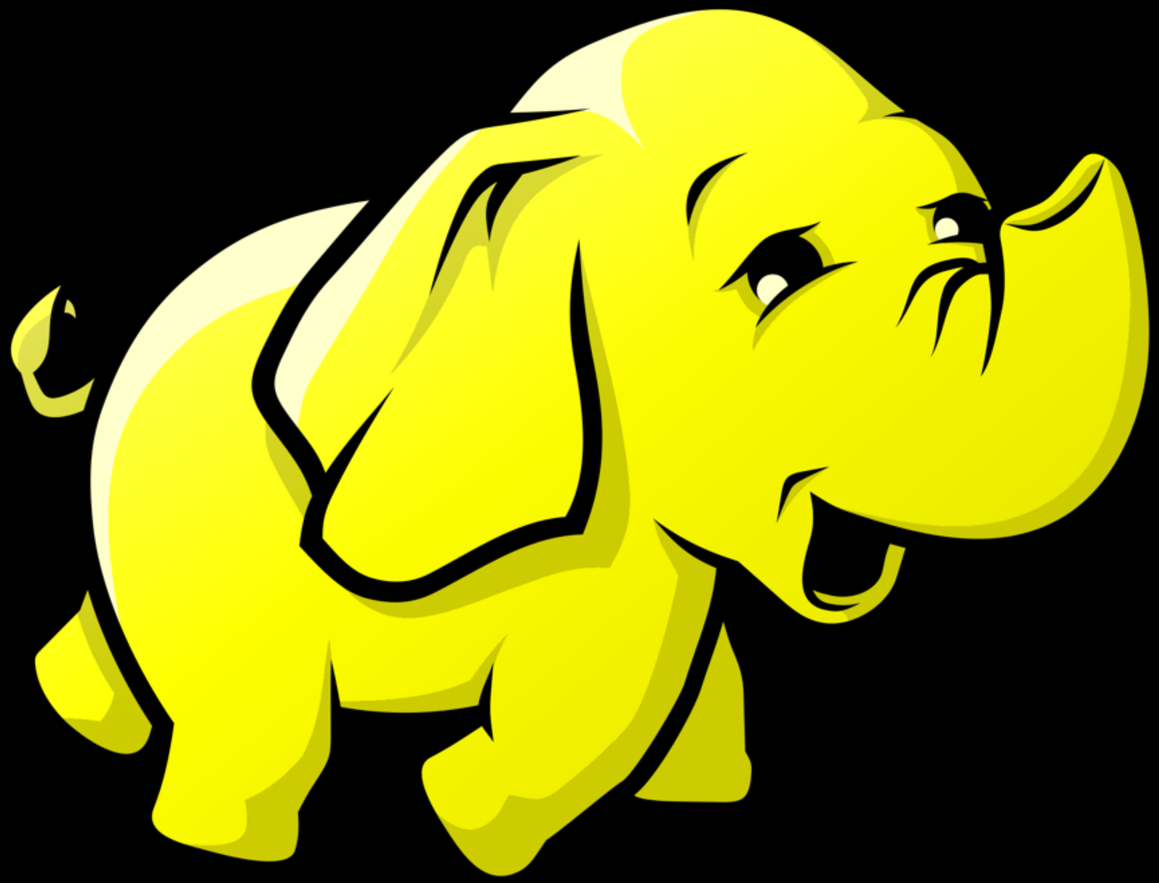
NumPy

SM StatsModels
Statistics in Python

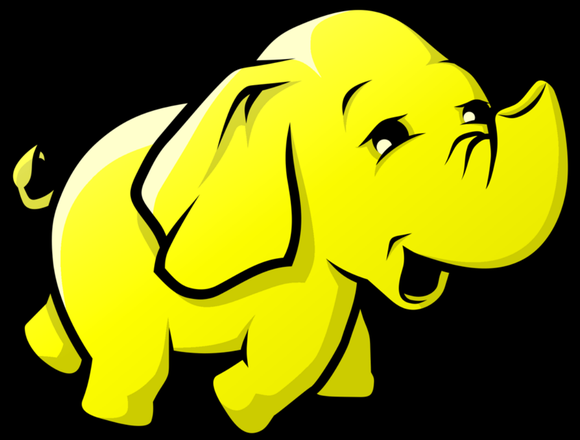


http://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html

- More of a general purpose language than R
- Arrays and matrices as basic data structures
- Supports data frames through Pandas
- Sophisticated machine learning libraries
- Generally limited to in-memory data sets



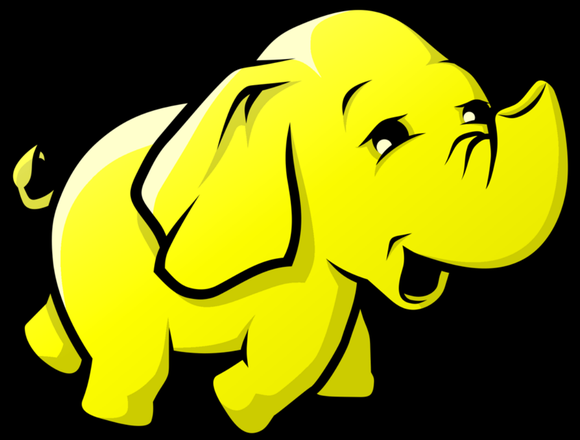
- Leverages commodity hardware to store large data sets at low cost
- Vibrant and diverse ecosystem
- Popular but not always best solution
- Probably best viewed as marketing terminology, as opposed to technology



130 freely licensed open source projects listed in the Hadoop Ecosystem Table

https://hadoopecosystemtable.github.io/

Category	Number of projects
Distributed Filesystem	7
Distributed Programming	18
NoSQL Database	4
Document Data Model	3
Stream Data Model	1
Key-Value Data Model	4
Graph Data Model	3
NewSQL	9
SQL-On-Hadoop	11
Data Ingestion	11
Service Programming	7
Scheduling	3
Machine Learning	6
Benchmarking	5
Security	3
System Deployment	12
Applications	5
Development Frameworks	2
Categorize Pending	16



Hadoop for Data Scientists

- Pulling data from repository (SQL, Hive)
- MapReduce programming (Java, Scala, Pig, Python)
- Spark in-memory framework is gaining adoption rapidly

tools rarely used in data science

version control

automated testing

shared code

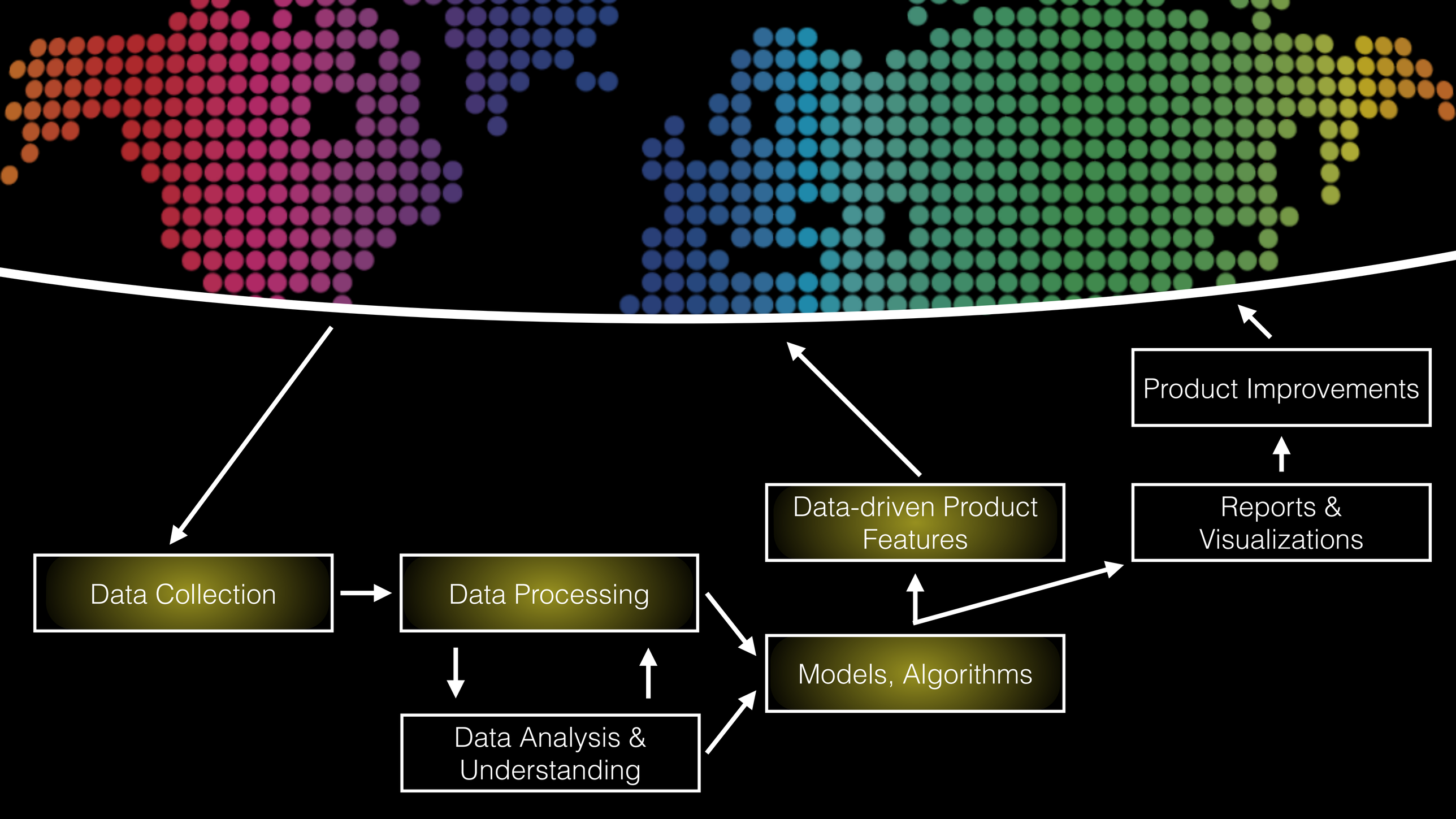
automated deployment

agile methodology

code review

software architecture


the cycle
of data science


Data Collection
Data Processing
Data Analysis & Understanding
Models, Algorithms
Data-driven Product Features
Reports & Visualizations
Product Improvements


data science
at
if(we)

if(we)

- Profitable startup actively pursuing big opportunities in social apps
- Millions of users on existing products
- Thousands of social contacts per second

what it should look like

1. Gain understanding of the product usage
2. See opportunity to make the product better
3. Create training data
4. Train predictive models
5. Put models in production
6. See improvements

what it often looks like

1. Gain understanding of the product usage
2. See opportunity to make the product better
3. Pull records from relational database to create interesting features (usually aggregates)
4. Train predictive models
5. Go implement models for production
6. See improvements

1. Gain understanding of the product usage
2. See opportunity to make the product better
3. Pull records from relational database to create interesting features (usually aggregates)
4. Train predictive models
5. Go implement models for production
6. See improvements

1. Gain understanding of the product usage
2. See opportunity to make the product better
3. Pull records from relational database to create interesting features (usually aggregates)
4. Train predictive models
5. Go implement models for production
6. See improvements

Cool!
Was it worth it?

implementation
pain points

- Data scientist hands model description to software engineer
- May need to translate features from SQL to Java
- Aggregate features require batch processing
- May need to adjust features and model to achieve real-time updates
- Fast scoring requires high-performance in-memory data structures


new thinking

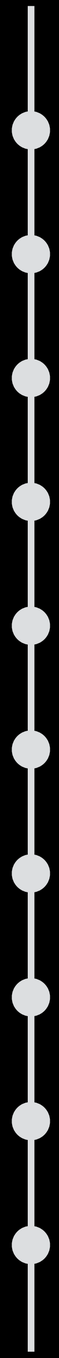

new architecture

one right way to data

one right way to data

event history

everything is an event



Bob registers

Alice registers

Alice updates profile

Bob opens app

Bob sees Alice in recommendations

Bob swipes yes on Alice

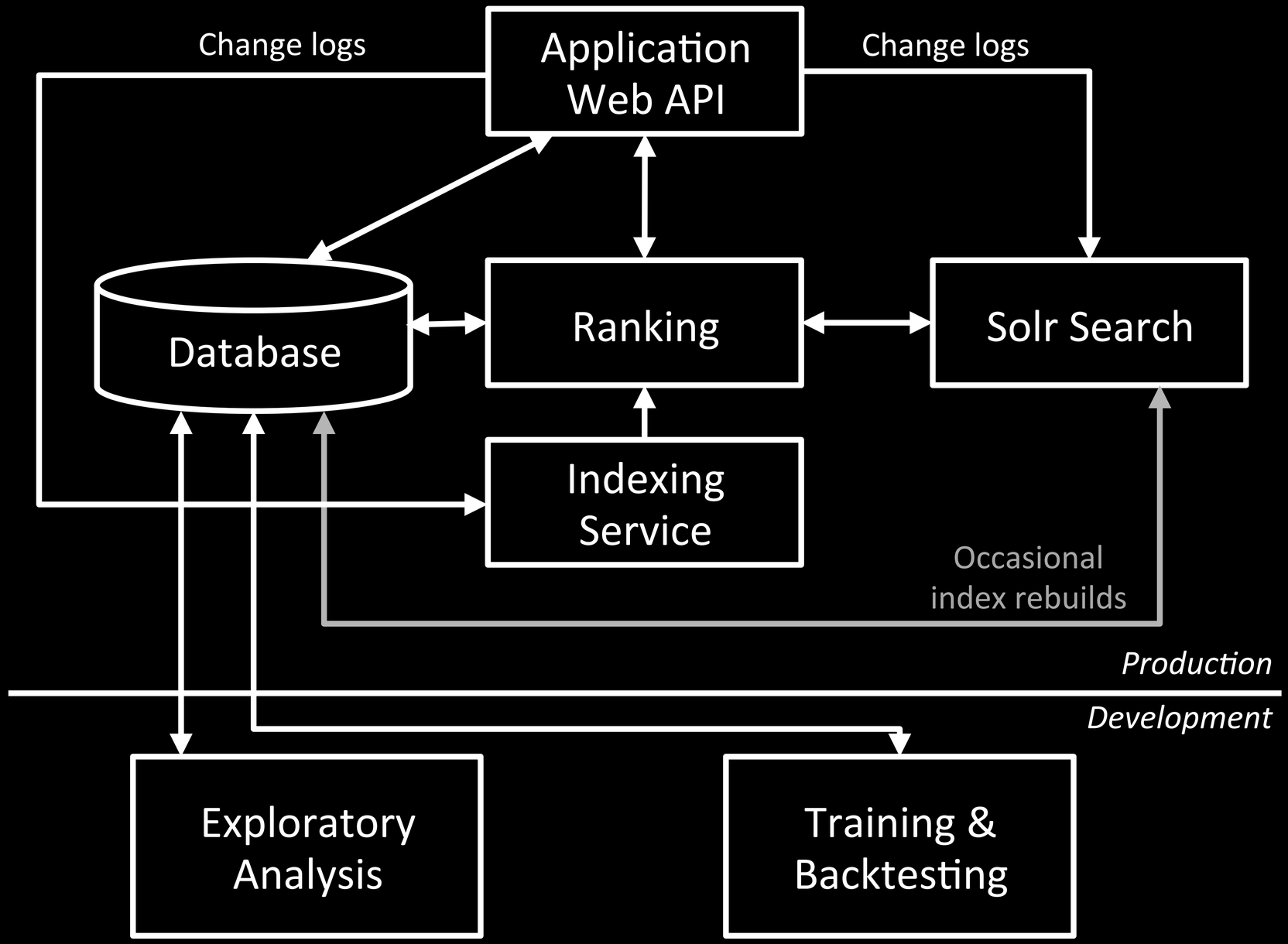
Alice receives push notification

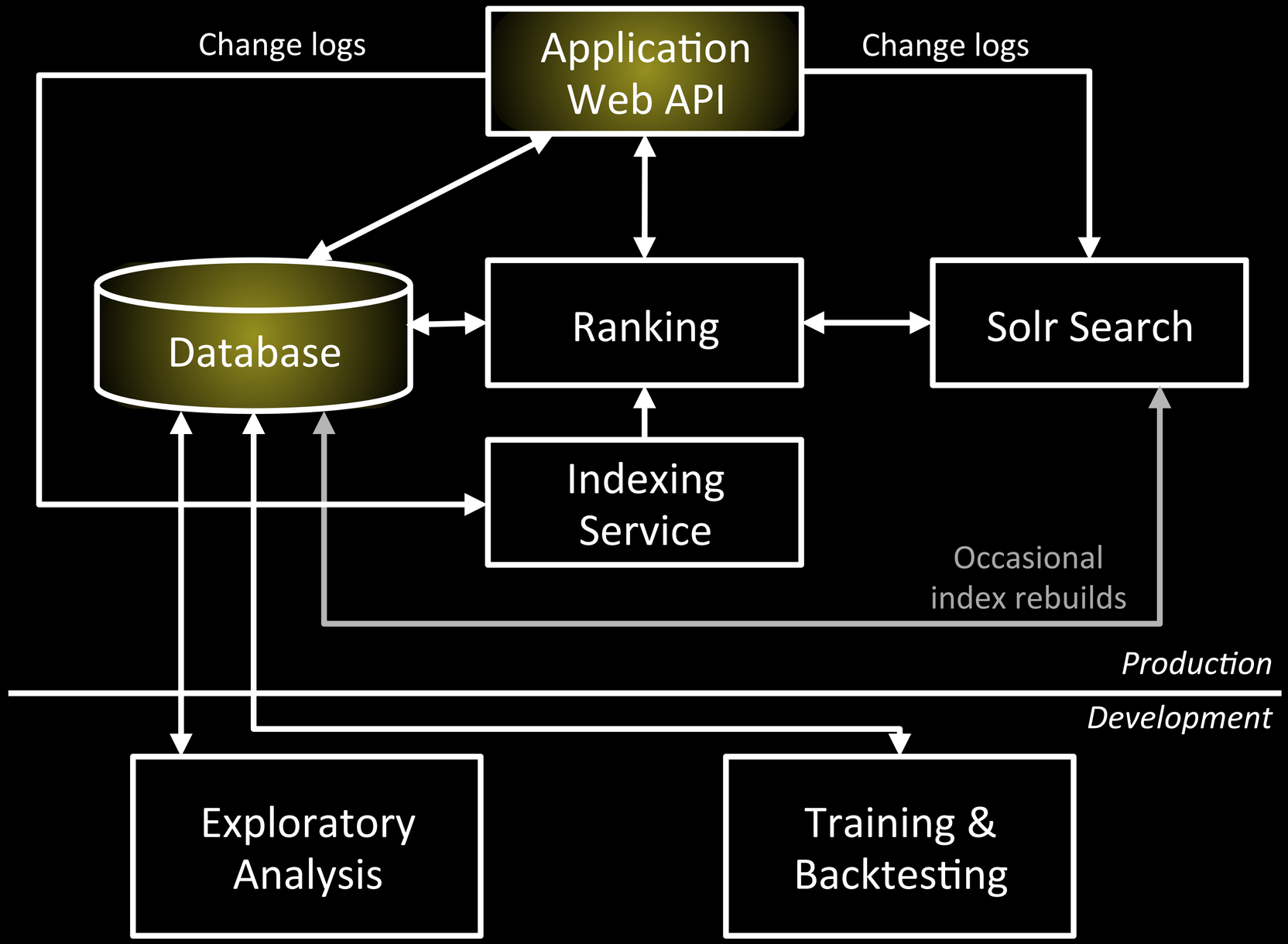
Alice sees Bob swiped yes

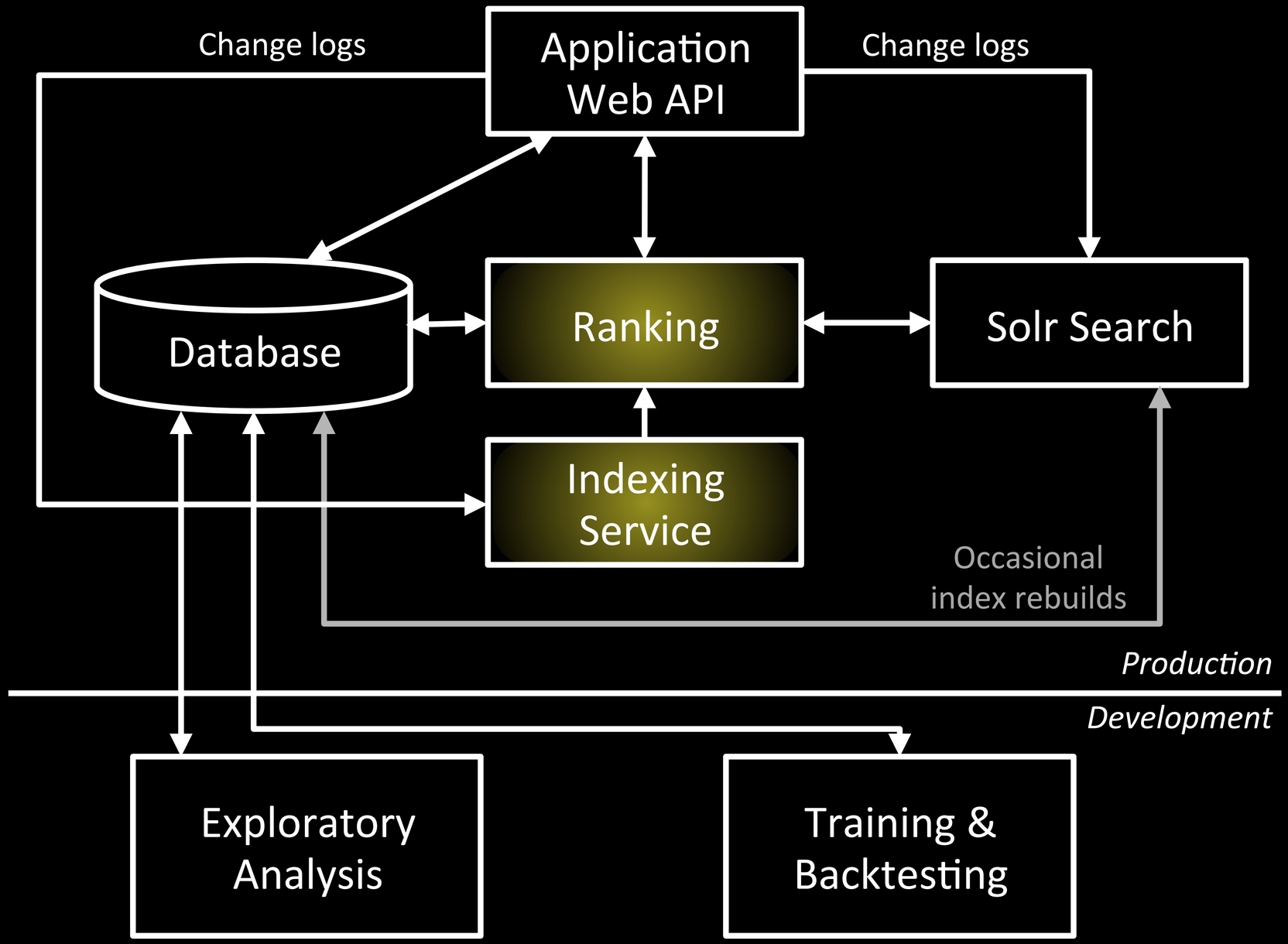
Alice swipes yes

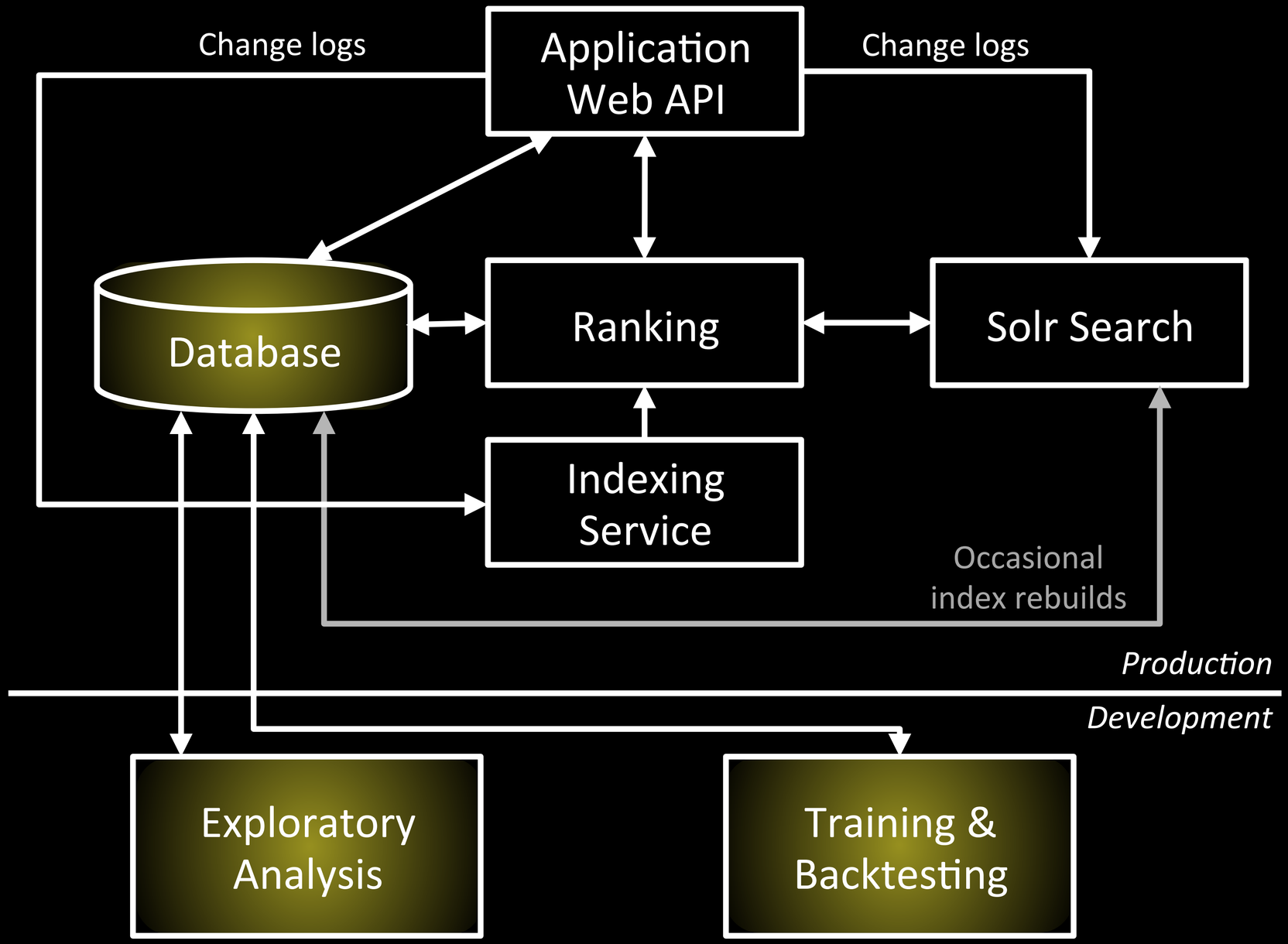
Alice sends message to Bob

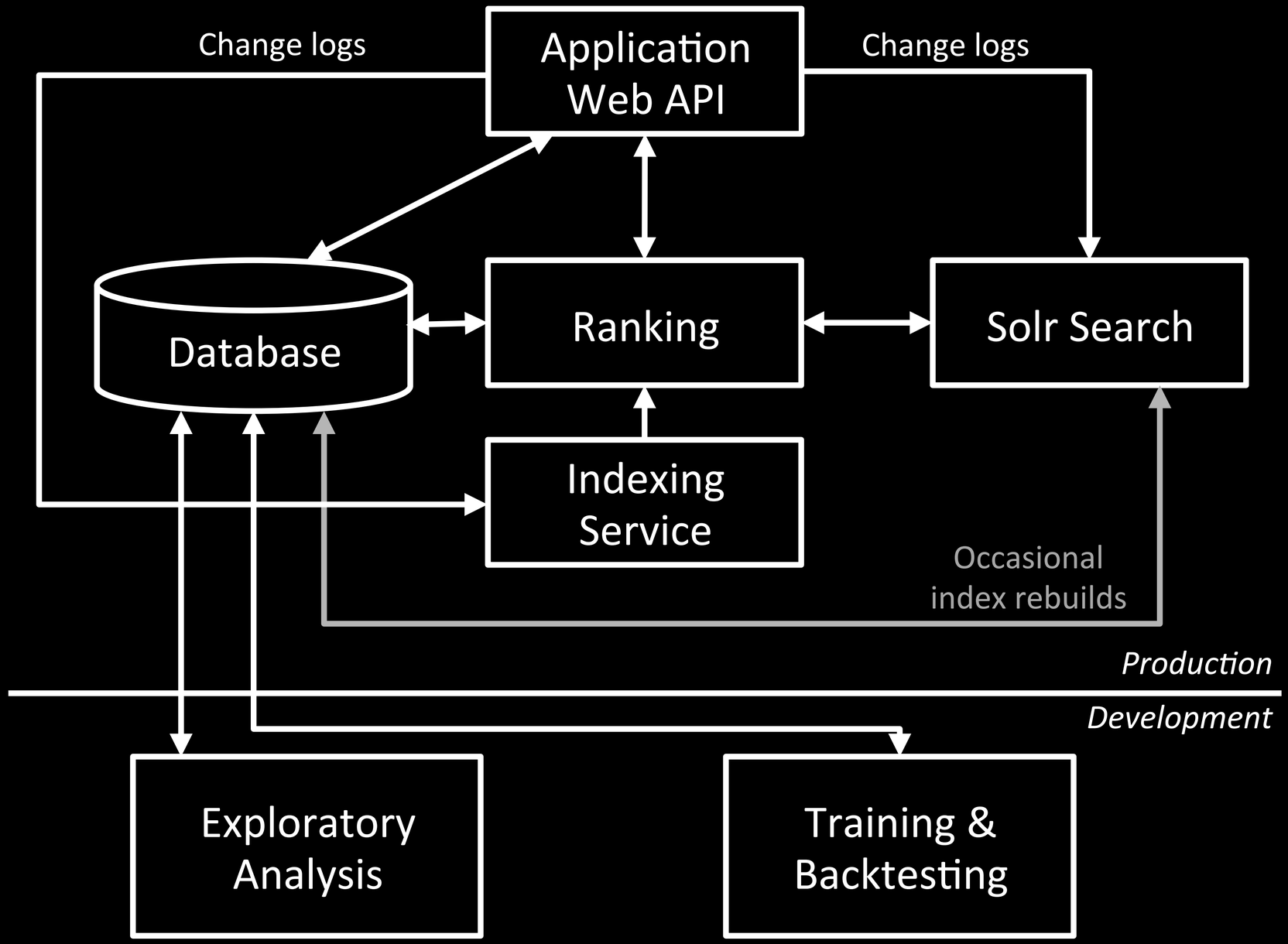
architecture comparison

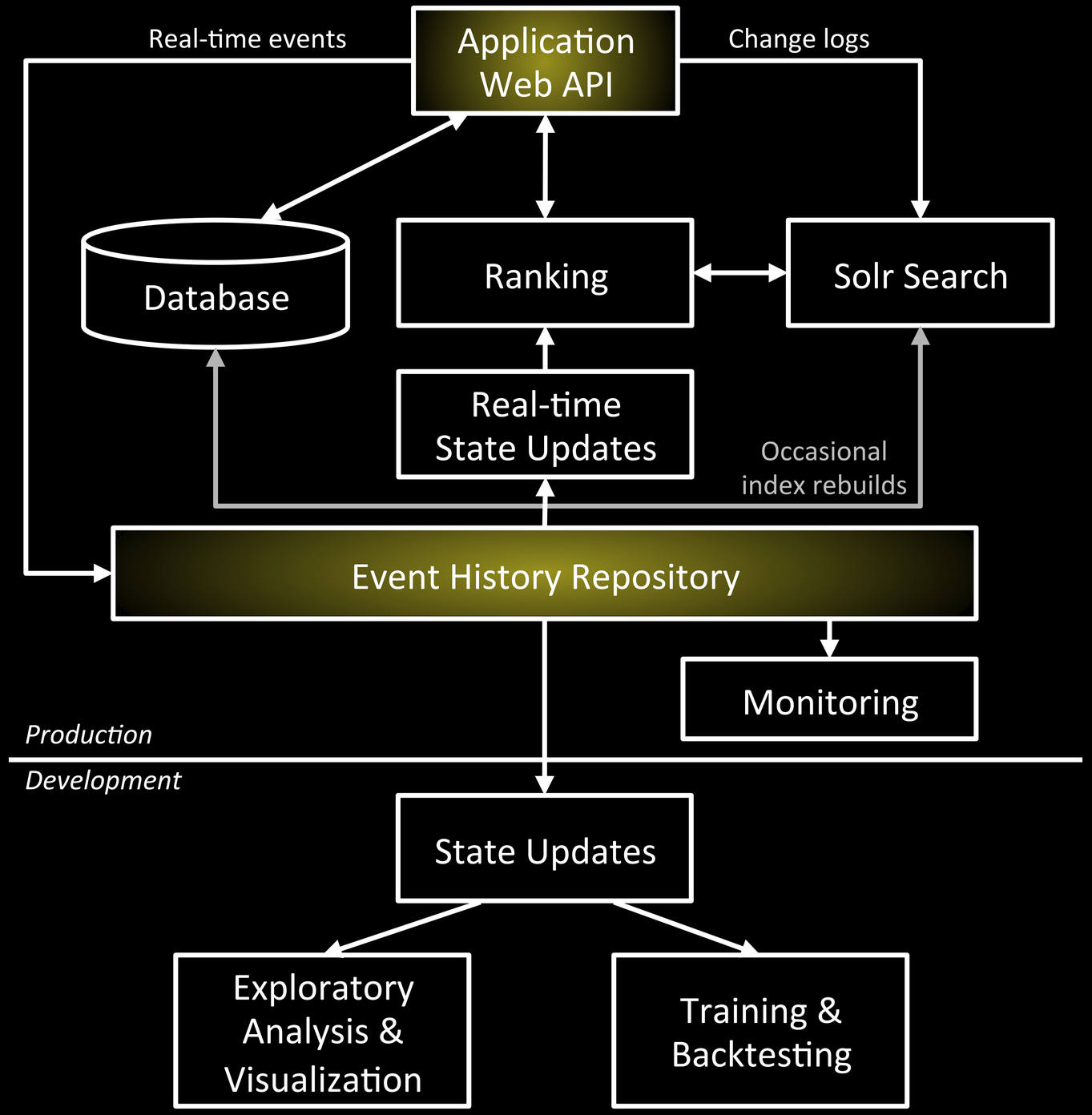

Change logs
Application
Web API
Change logs
Database
Ranking
Solr Search
Indexing
Service
Occasional
index rebuilds
Production
Development
Exploratory
Analysis
Training &
Backtesting

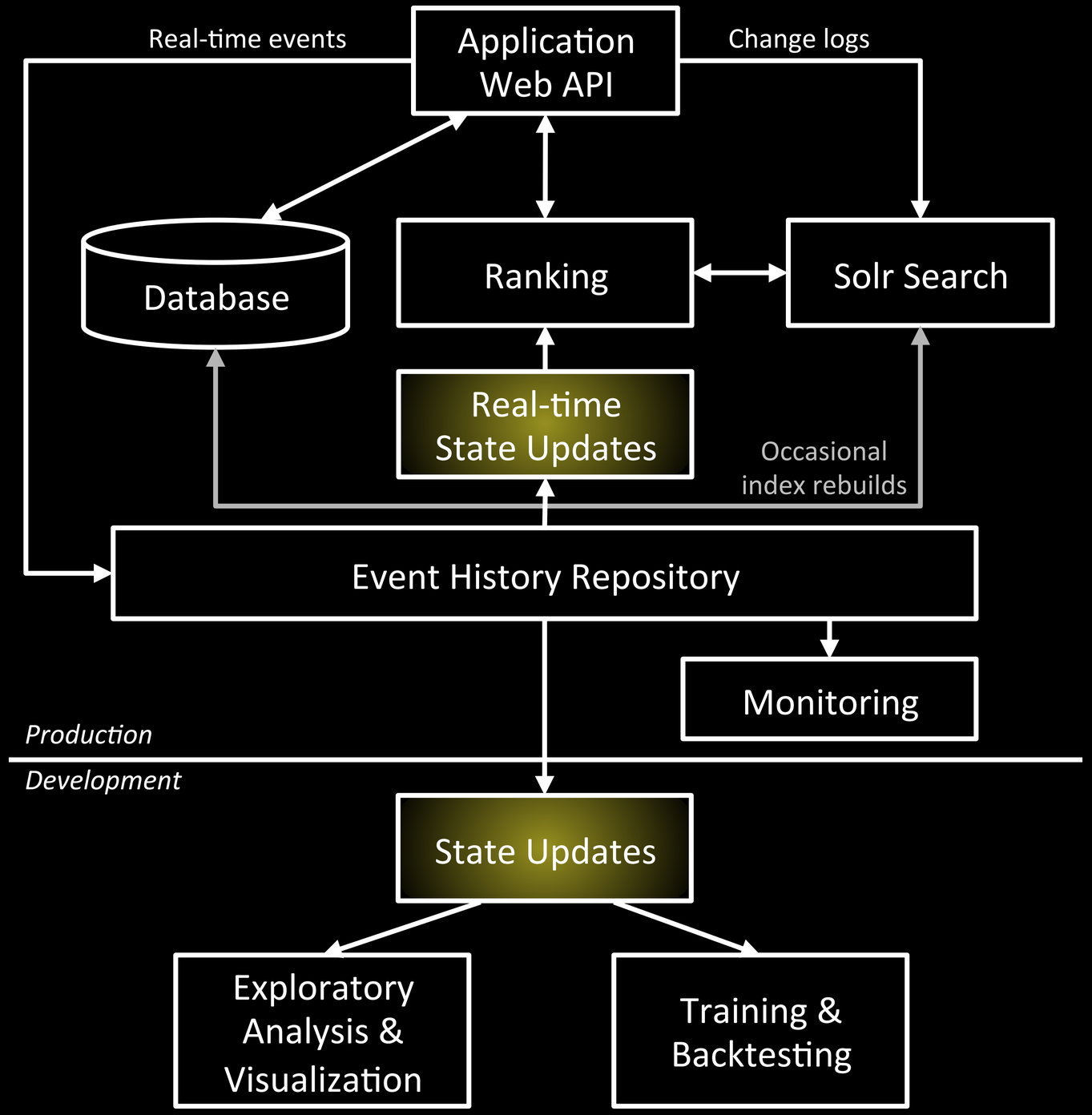

Change logs
Application
Web API
Change logs
Database
Ranking
Solr Search
Indexing
Service
Occasional
index rebuilds
Production
Development
Exploratory
Analysis
Training &
Backtesting

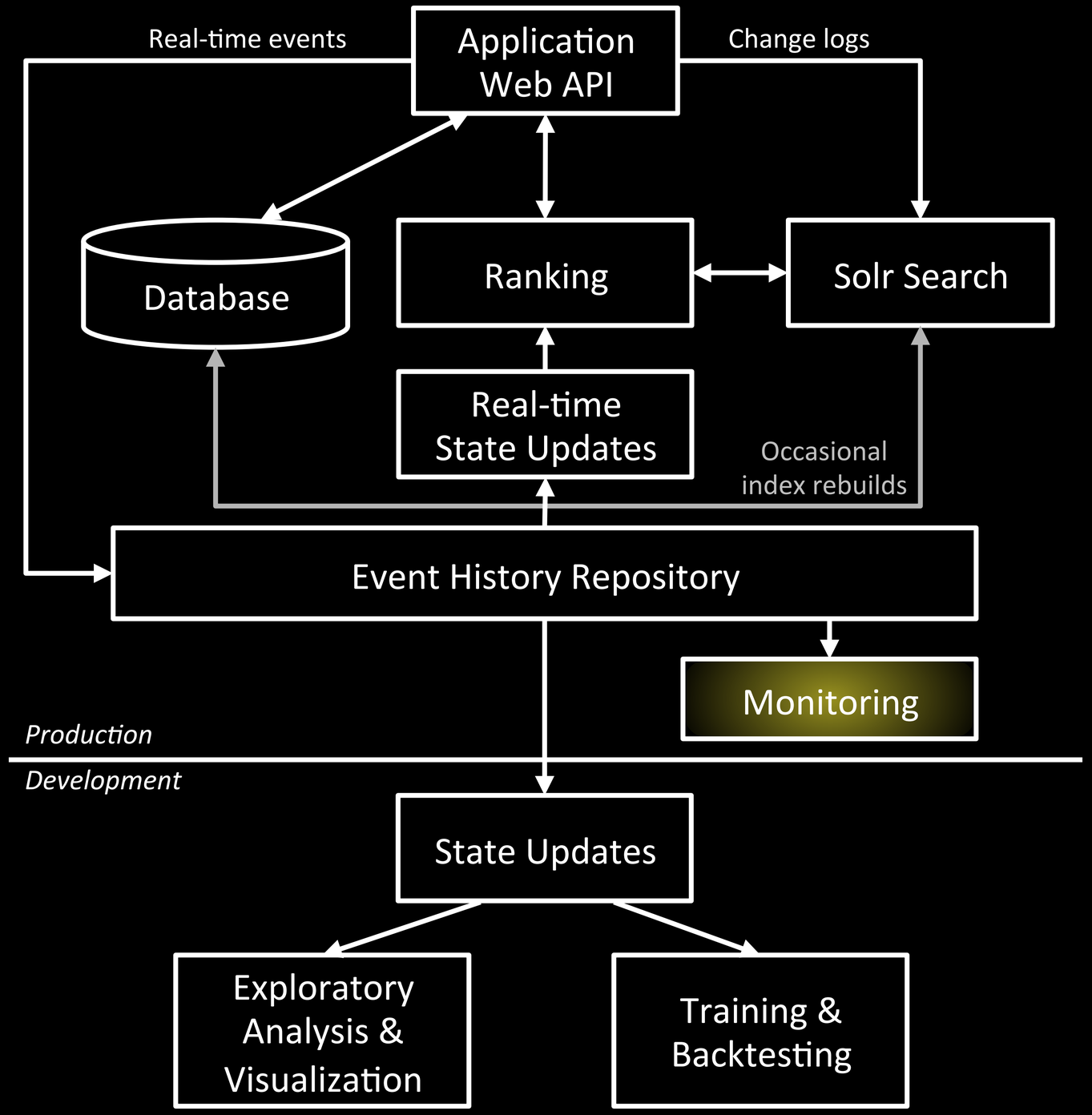

Change logs
Application
Web API
Change logs
Database
Ranking
Solr Search
Indexing
Service
Occasional
index rebuilds
Production
Development
Exploratory
Analysis
Training &
Backtesting

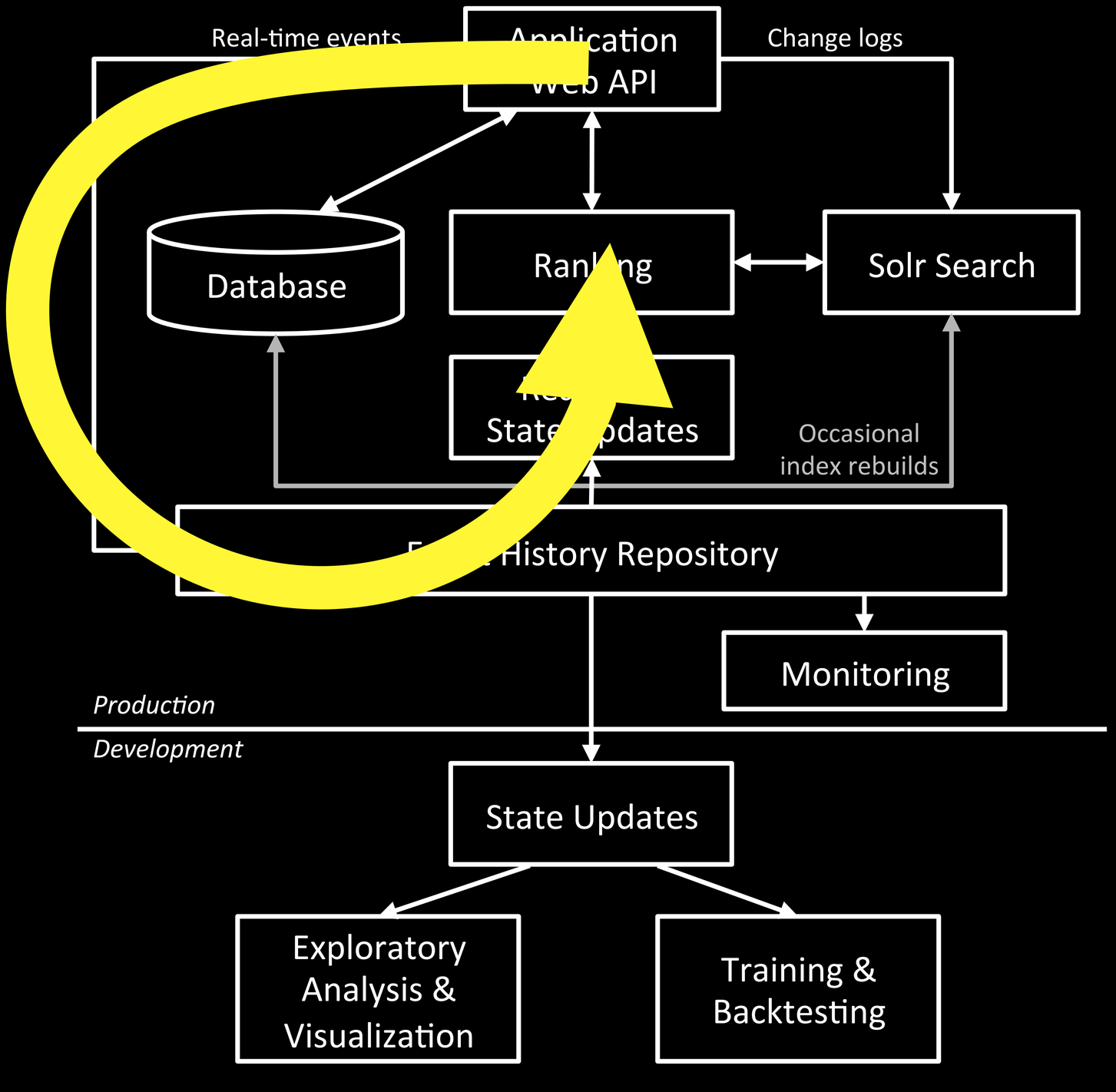

Change logs
Application
Web API
Change logs
Database
Ranking
Solr Search
Indexing
Service
Occasional
index rebuilds
Production
Development
Exploratory
Analysis
Training &
Backtesting

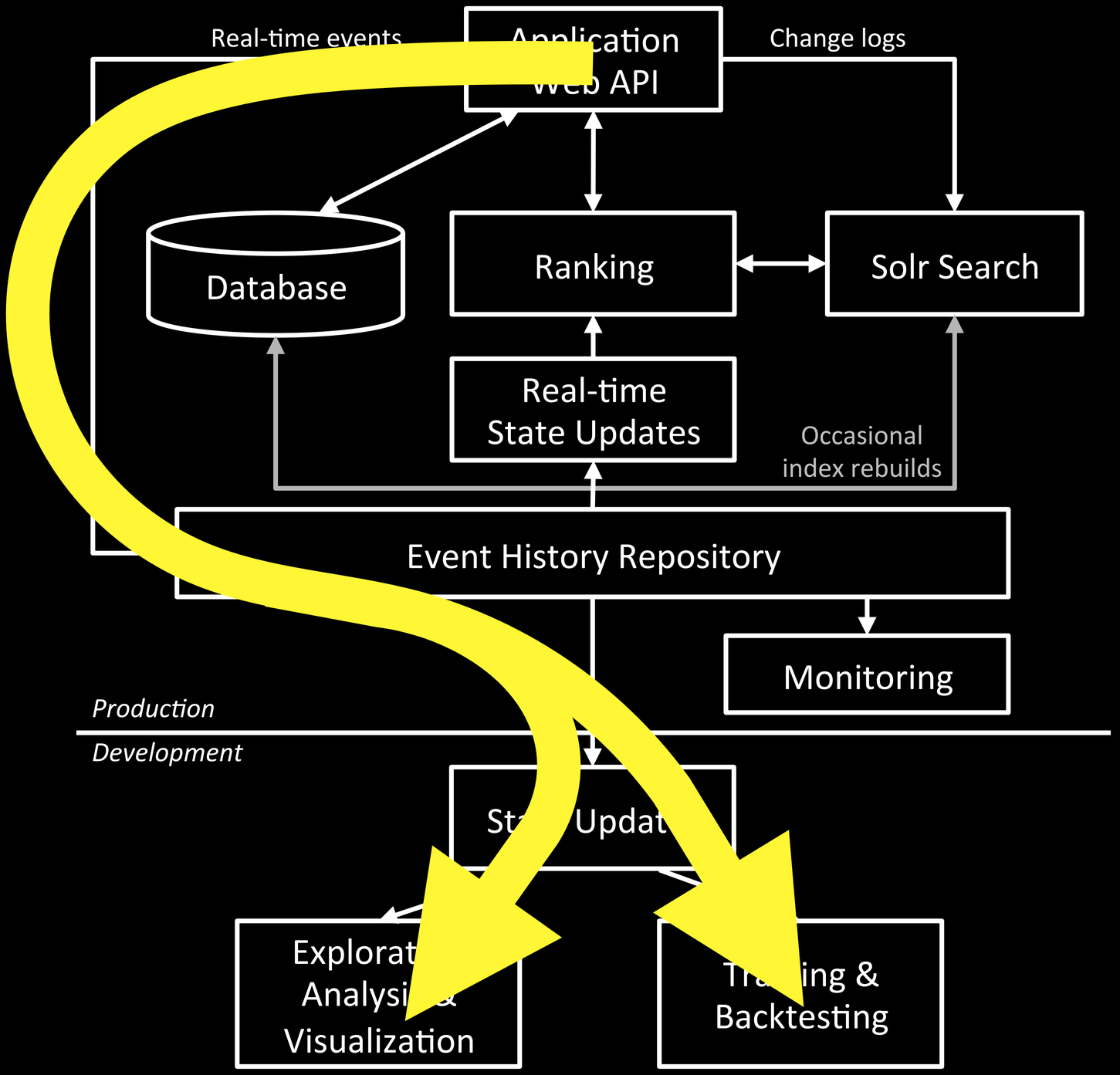

Change logs
Application
Web API
Change logs
Database
Ranking
Solr Search
Indexing
Service
Occasional
index rebuilds
Production
Development
Exploratory
Analysis
Training &
Backtesting


Real-time events
Application
Web API
Change logs
Database
Ranking
Solr Search
Real-time
State Updates
Occasional
index rebuilds
Event History Repository
Monitoring
Production
Development
State Updates
Exploratory
Analysis &
Visualization
Training &
Backtesting


Real-time events
Application
Web API
Change logs
Database
Ranking
Solr Search
Real-time
State Updates
Occasional
index rebuilds
Event History Repository
Monitoring
Production
Development
State Updates
Exploratory
Analysis &
Visualization
Training &
Backtesting


Real-time events
Application
Web API
Change logs
Database
Ranking
Solr Search
Real-time
State Updates
Occasional
index rebuilds
Event History Repository
Monitoring
Production
Development
State Updates
Exploratory
Analysis &
Visualization
Training &
Backtesting


Real-time events
Application
Web API
Change logs
Database
Ranking
Solr Search
State Updates
Occasional
index rebuilds
History Repository
Monitoring
Production
Development
State Updates
Exploratory
Analysis &
Visualization
Training &
Backtesting


Real-time events
Application
Web API
Change logs
Database
Ranking
Solr Search
Real-time
State Updates
Occasional
index rebuilds
Event History Repository
Monitoring
Production
Development
Exploratory
Analysis &
Visualization
Training &
Backtesting

Event History API

```
trait EventHistory {
  def publishEvent(e: Event)

  def getEvents(
    startTime: Date,
    endTime: Date,
    eventFilter: EventFilter,
    eventHandler: EventHandler
  )
}
```

Event History API

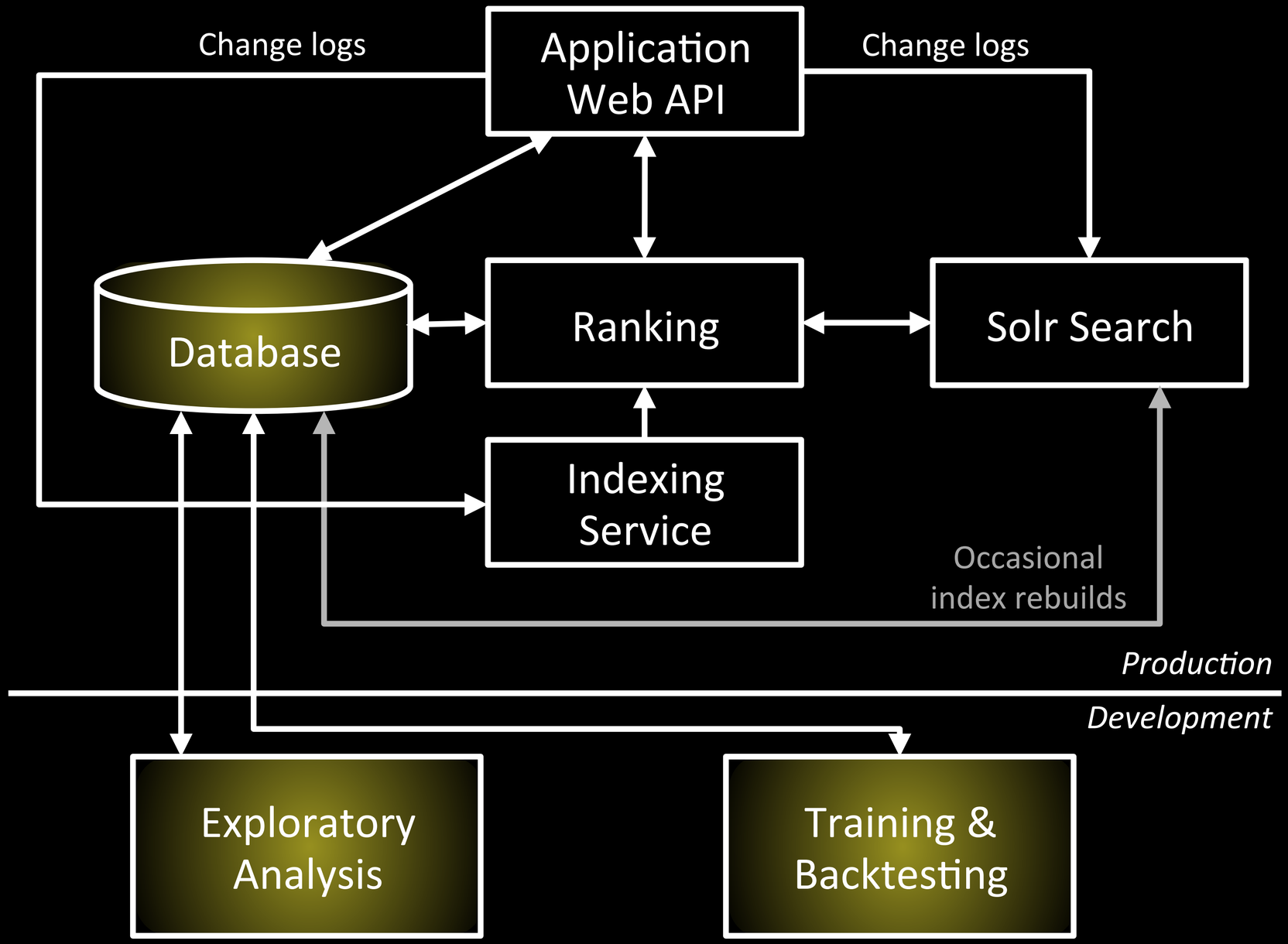
```
trait EventHistory {
  def publishEvent(e: Event)

  def getEvents(
    startTime: Date,
    endTime: Date,
    eventFilter: EventFilter,
    eventHandler: EventHandler
  )
}
```

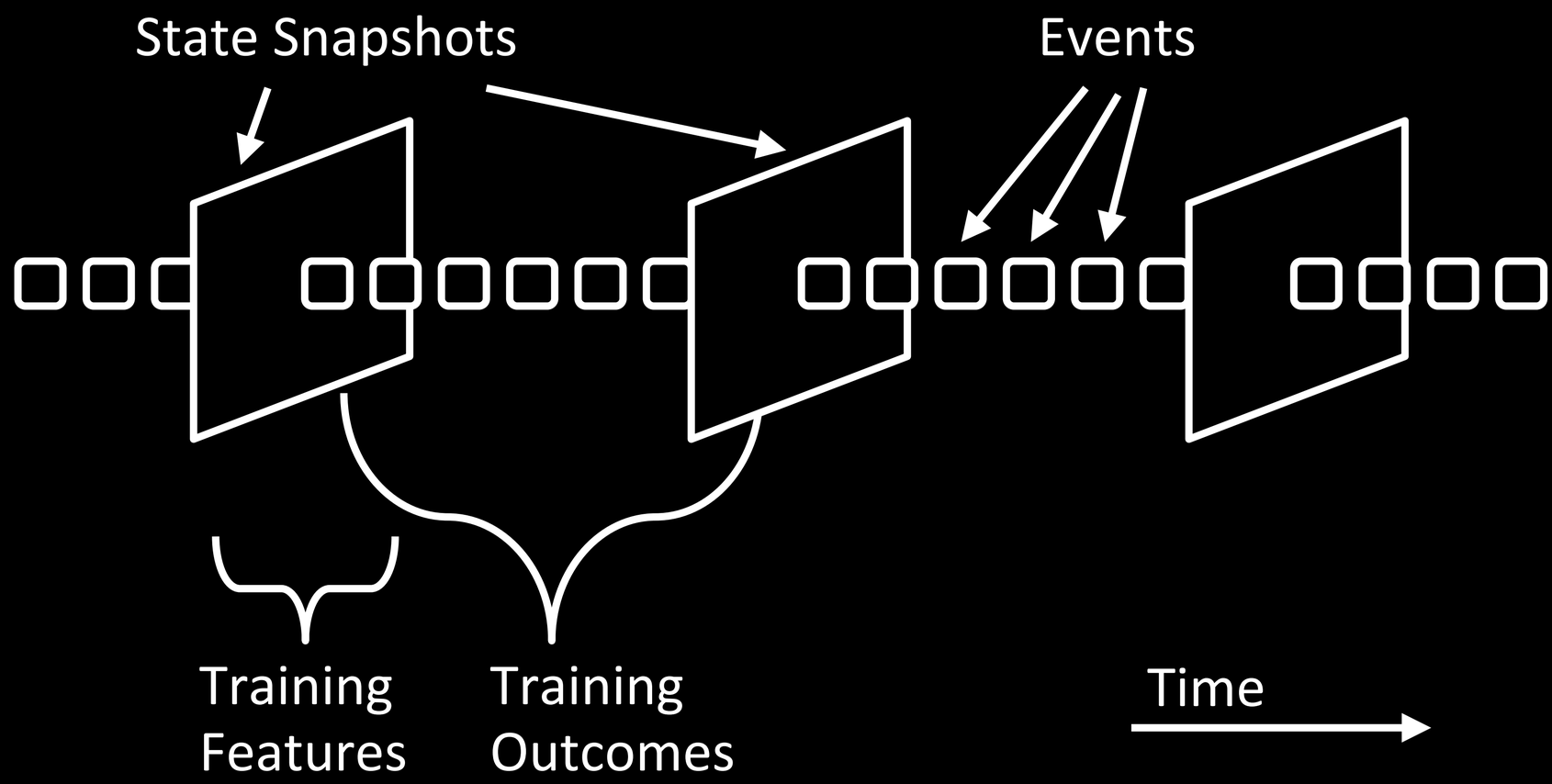
Event History API

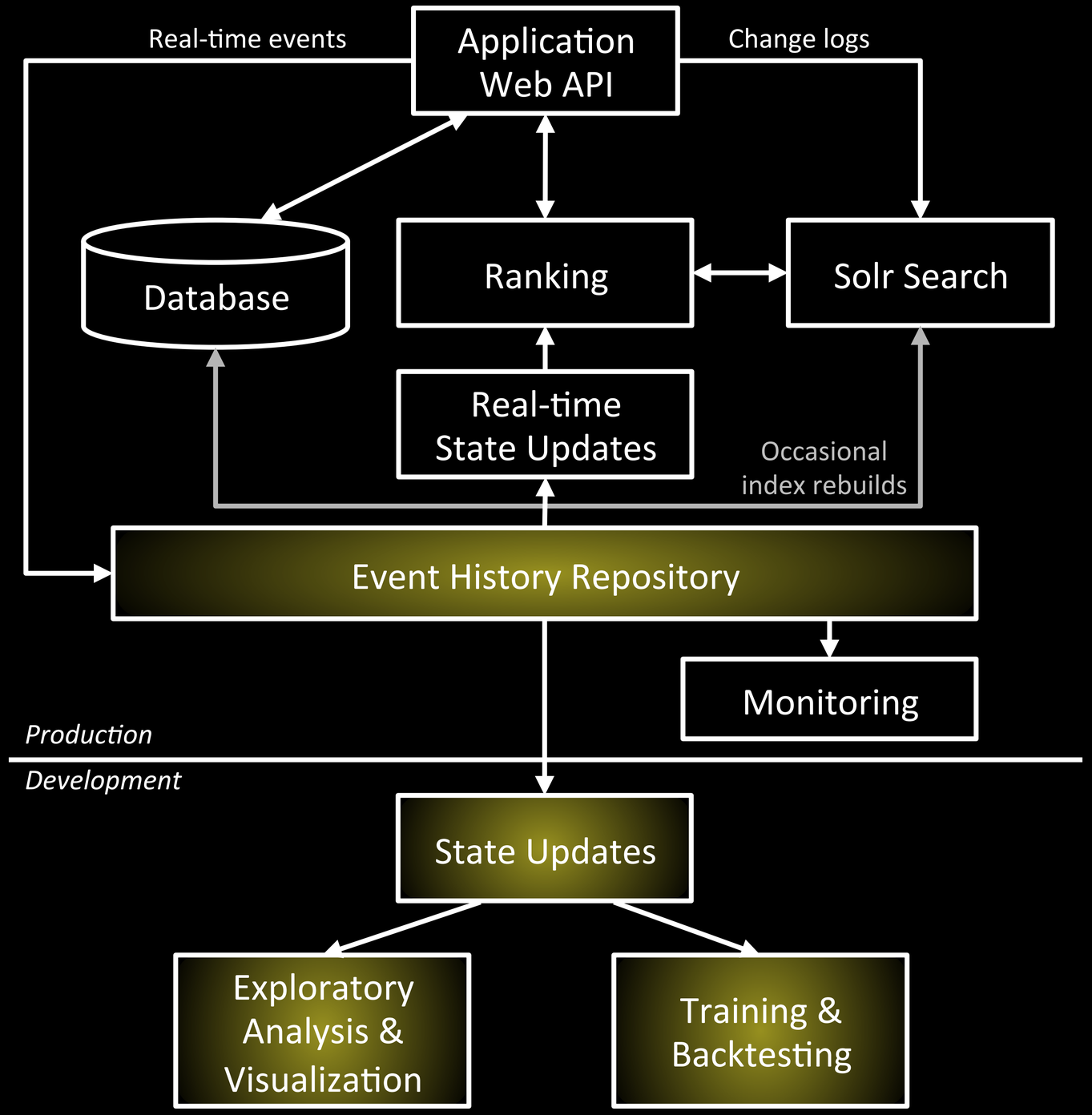
```
trait EventHistory {
  def publishEvent(e: Event)

  def getEvents(
    startTime: Date,
    endTime: Date,
    eventFilter: EventFilter,
    eventHandler: EventHandler
  )
}
```

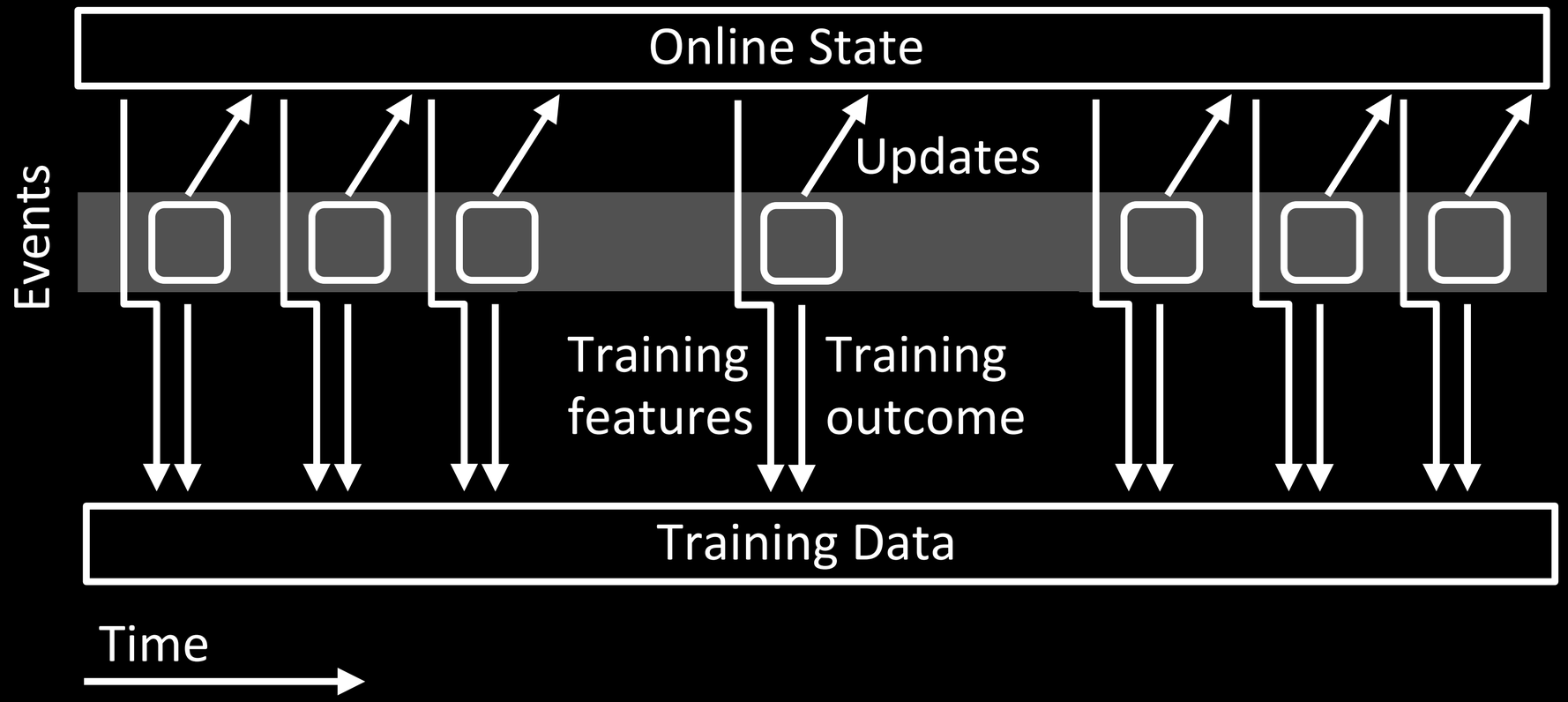
+∞ for real-time streaming

training data comparison


Change logs
Application
Web API
Change logs
Database
Ranking
Solr Search
Indexing
Service
Occasional
index rebuilds
Production
Development
Exploratory
Analysis
Training &
Backtesting

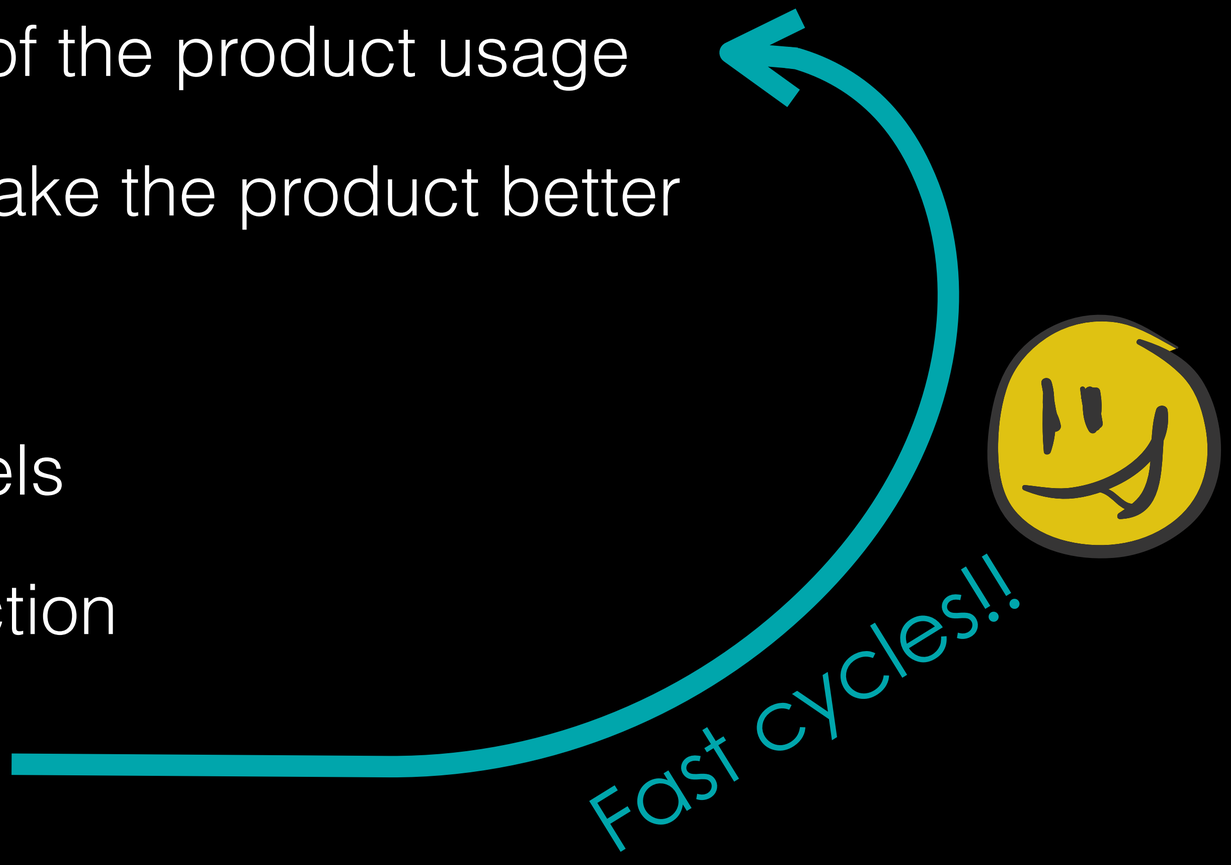

State Snapshots
Events
Training Features
Training Outcomes
Time


Real-time events
Application
Web API
Change logs
Database
Ranking
Solr Search
Real-time
State Updates
Occasional
index rebuilds
Event History Repository
Monitoring
Production
Development
State Updates
Exploratory
Analysis &
Visualization
Training &
Backtesting


Online State
Events
Updates
Training features
Training outcome
Training Data
Time


SYRIAN ORTHODOX CLUB Est. 1852
רחוב הנזיר הסיריאני
طريق دير السريان
SYRIAC CONVENT ST.
ARARAT ST.

1. Gain understanding of the product usage
2. See opportunity to make the product better
3. Create training data
4. Train predictive models
5. Put models in production
6. See improvements



Live Demo

https://github.com/ifwe/antelope

https://www.kaggle.com/c/acm-sf-chapter-hackathon-small

- Open source implementation derived from if(we)’s proprietary platform
- Not ready scale or production, but useful for demonstration purposes
- Seeking collaborators

product update events

```
{
  “timestamp” : “2012-05-03 6:43:15”,
  “eventType” : “ProductUpdate”,
  “eventProperties” : {
    “sku” : “1032361”,
    “regularPrice” : “19.99”,
    “name” : “Need for Speed: Hot Pursuit”,
    “description” : “Fasten your seatbelt and
     get ready to drive like your life depends
     on it...”
    ...
  }
}
```

product view events

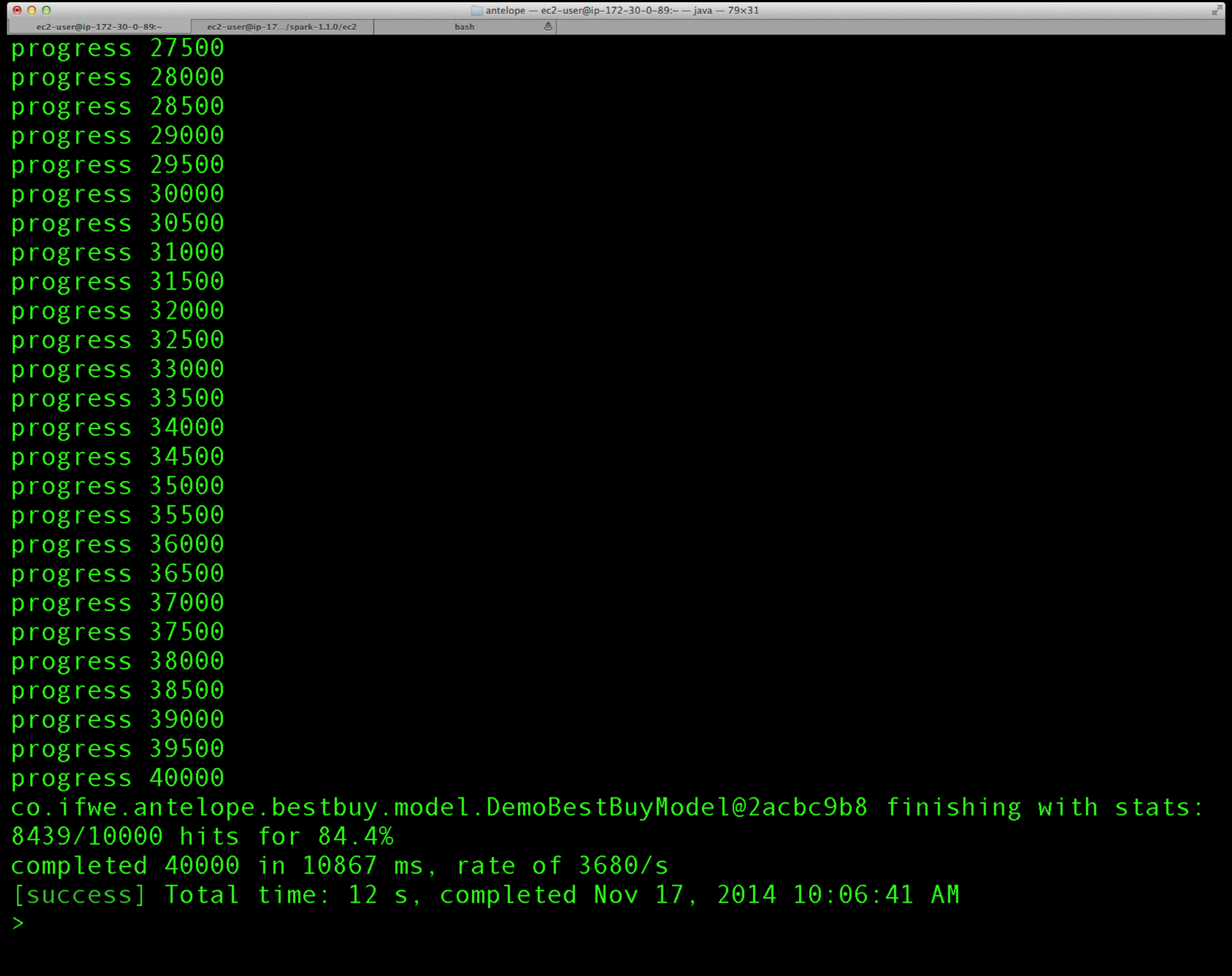
```
{
  “timestamp” : “2011-10-31 09:48:46”,
  “eventType” : “ProductView”,
  “eventProperties” : {
    “query” : “Modern warfare”,
    “skuSelected” : “2670133”
  }
}
```

demo

Try it yourself, code and instructions at:
https://github.com/ifweco/antelope/blob/master/doc/demo.md

```scala
class TermPopularity extends Feature[SC] {
  val ct = s.counter(queryTerms, skuViewed)

  override def score(implicit ctx: SC): (Long) => Double = {
    val terms = ctx.query.normalize.split(" ")
    id => terms.map { term =>
      ct(term, id) div ct(term)
    }.product
  }
}

class TfIdf extends  Feature[SC] {
  val termFreq = s.counter(productNameUpdatedTerms, skuUpdated)
  val docFreq = s.set(productNameUpdatedTerms, skuUpdated)
  val docs = s.set(skuUpdated)

  override def score(implicit ctx: SC): (Long) => Double = {
    val terms = ctx.query.normalize.split(" ")
    val n = docs.size()
    id => terms.map { term =>
      val tf = termFreq(term, id)
      val df = docFreq.size(term)
      sqrt(tf) * sq(1D + log(n/(1D+df)))
    }.sum
  }
}
```

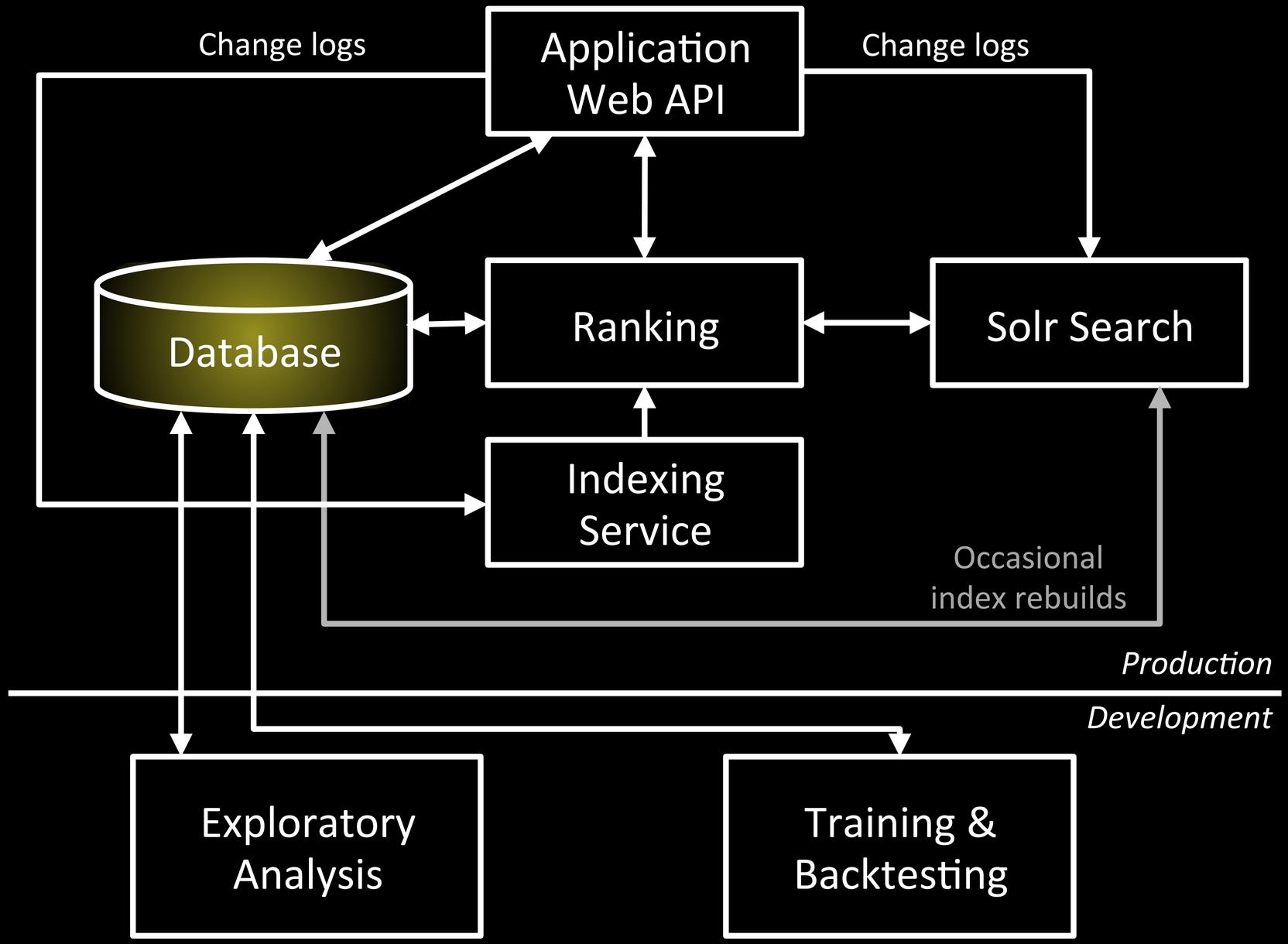
```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.3565   0.0000   0.0030   2.3765

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  7.15236    0.30019   23.83   <2e-16 ***
feature_1    2.01115    0.04070   49.41   <2e-16 ***
feature_2   11.95880    0.61440   19.46   <2e-16 ***
feature_3    3.59281    0.07072   50.80   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 55452  on 39999  degrees of freedom
Residual deviance: 13978  on 39996  degrees of freedom
AIC: 13986

Number of Fisher Scoring iterations: 12

>
> # save output to text file
> write(as.vector(coef(mylogit)[-c(1)]) / as.vector(colsd),
+   file='r_logit_coef.txt', sep=",", ncolumns=1000)
>
Johanns-MacBook-Pro-2:training_small johann$
```

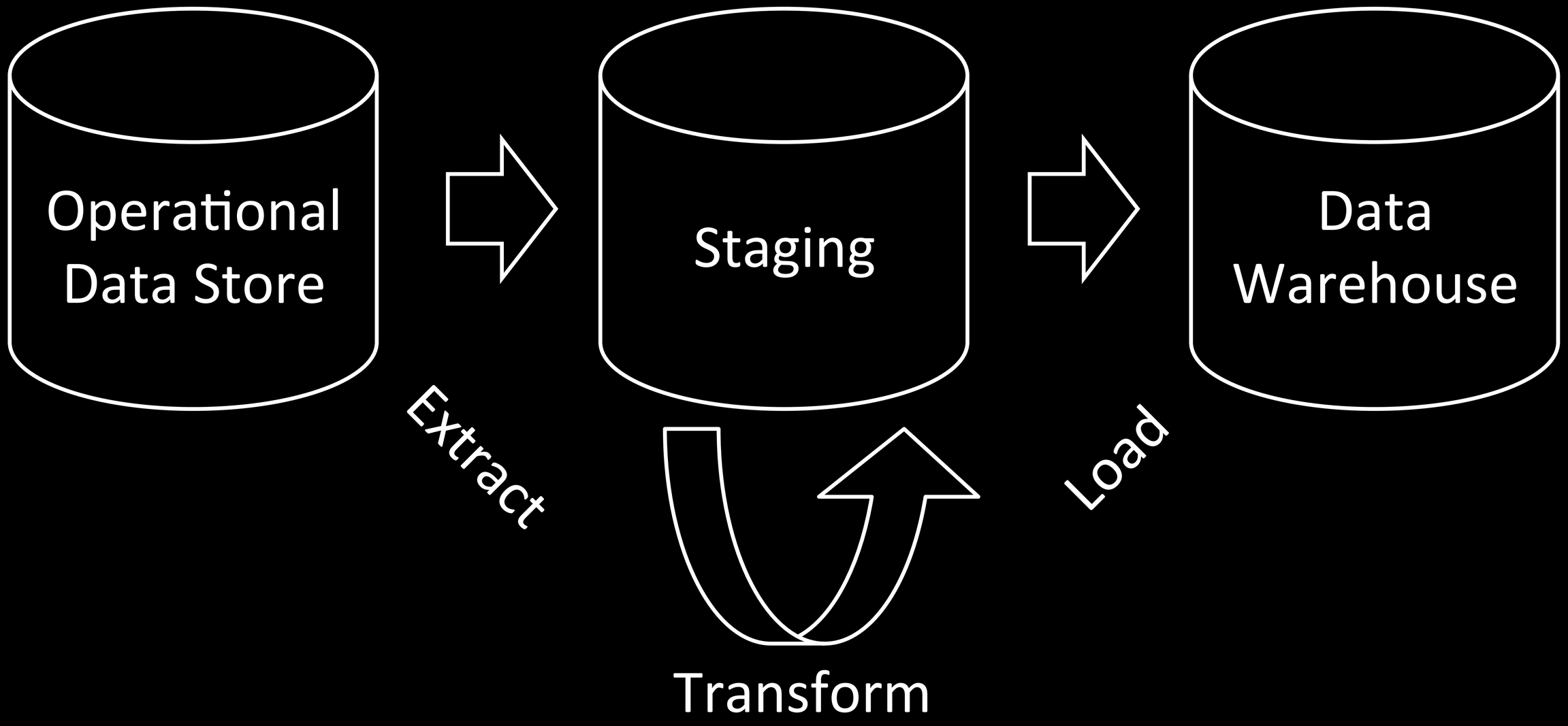

antelope — ec2-user@ip-172-30-0-89:~ — java — 79×31
ec2-user@ip-172-30-0-89:~
ec2-user@ip-17.../spark-1.1.0/ec2
bash
progress 27500
progress 28000
progress 28500
progress 29000
progress 29500
progress 30000
progress 30500
progress 31000
progress 31500
progress 32000
progress 32500
progress 33000
progress 33500
progress 34000
progress 34500
progress 35000
progress 35500
progress 36000
progress 36500
progress 37000
progress 37500
progress 38000
progress 38500
progress 39000
progress 39500
progress 40000
co.ifwe.antelope.bestbuy.model.DemoBestBuyModel@2acbc9b8 finishing with stats:
8439/10000 hits for 84.4%
completed 40000 in 10867 ms, rate of 3680/s
[success] Total time: 12 s, completed Nov 17, 2014 10:06:41 AM
>


new
tools :)

Architecture for Data Science

data warehousing


Change logs
Application
Web API
Change logs
Database
Ranking
Solr Search
Indexing
Service
Occasional
index rebuilds
Production
Development
Exploratory
Analysis
Training &
Backtesting


Operational
Data Store
Staging
Data
Warehouse
Extract
Load
Transform


RIP
ETL

~~extract~~ ⇨ transform ⇨ ~~load~~

log ⇨ transform ⇨ use

Data Architecture

dimensional modeling

relational modeling

Warehouse Design

normalization

denormalization

star schema

slowly changing
dimensions

slowly changing dimensions

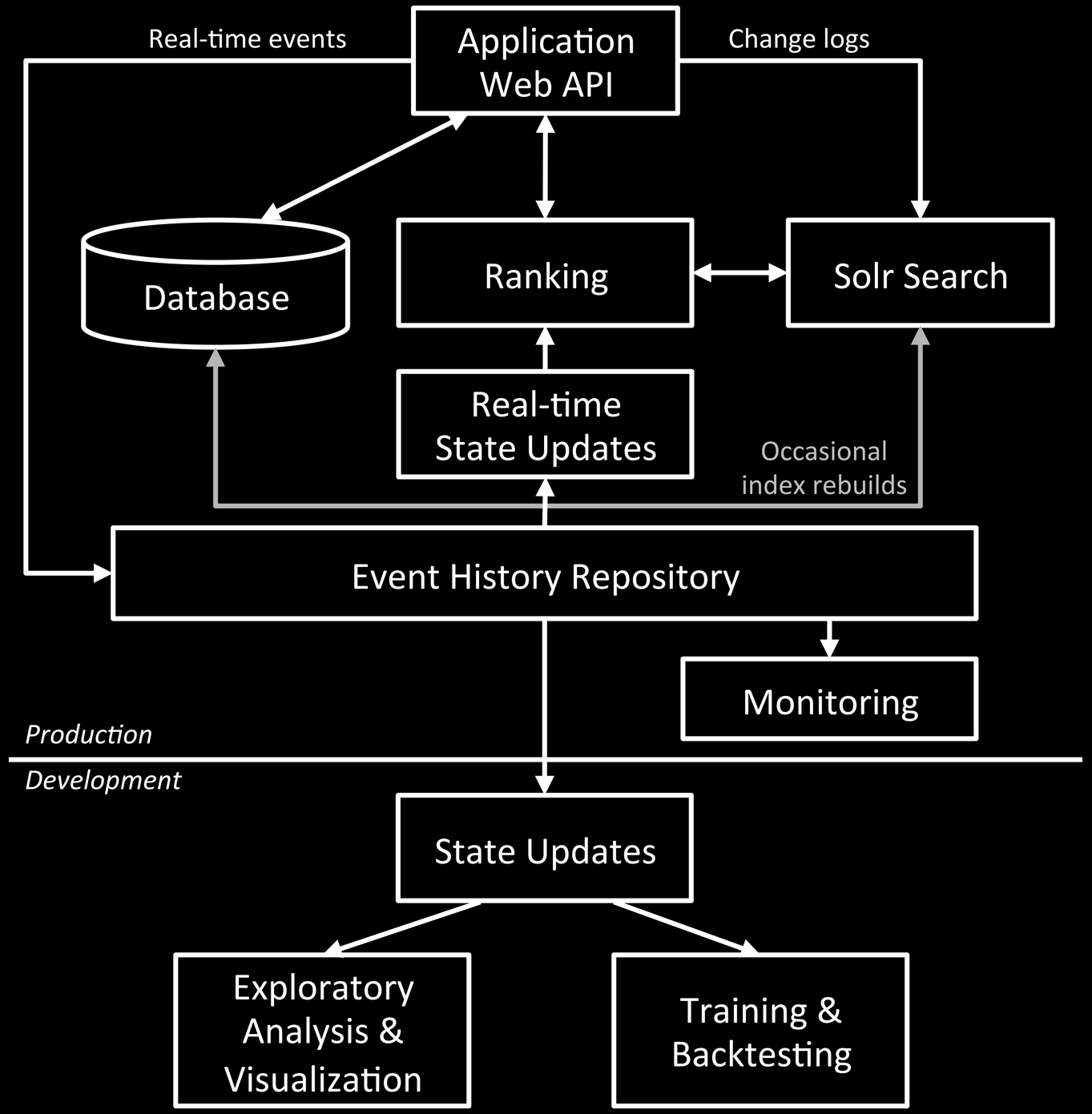
type 1: overwrite the old data

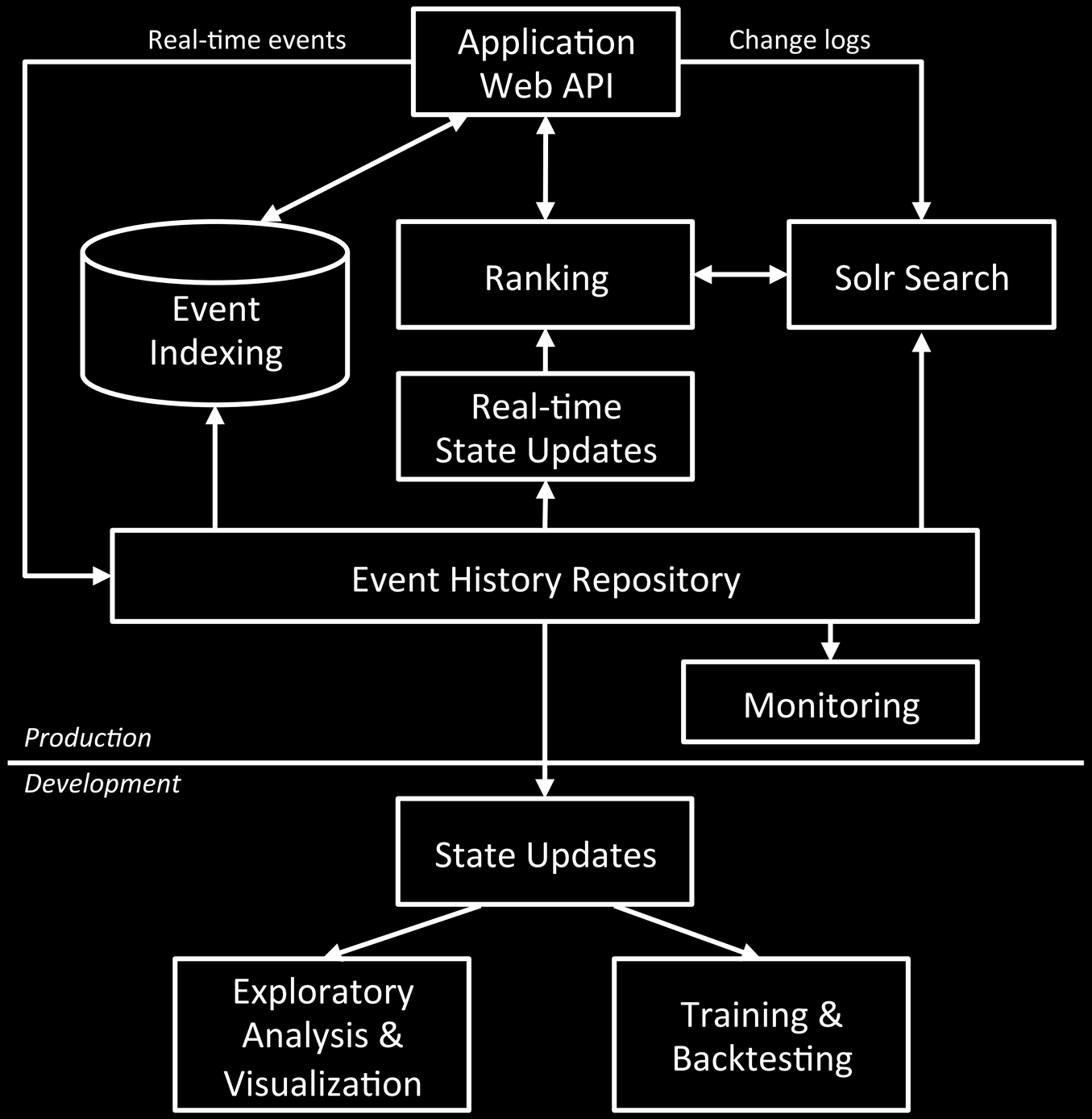
type 2: multiple rows with versioning

type 3: extra columns for older versions

~~slowly changing dimensions~~

event history


Real-time events
Application
Web API
Change logs
Database
Ranking
Solr Search
Real-time
State Updates
Occasional
index rebuilds
Event History Repository
Monitoring
Production
Development
State Updates
Exploratory
Analysis &
Visualization
Training &
Backtesting


Real-time events
Application
Web API
Change logs
Event
Indexing
Ranking
Solr Search
Real-time
State Updates
Event History Repository
Monitoring
Production
Development
State Updates
Exploratory
Analysis &
Visualization
Training &
Backtesting

```
trait EventHistory {
  def publishEvent(e: Event)

  def getEvents(
    startTime: Date,
    endTime: Date,
    eventFilter: EventFilter,
    eventHandler: EventHandler
  )
}
```

event history design

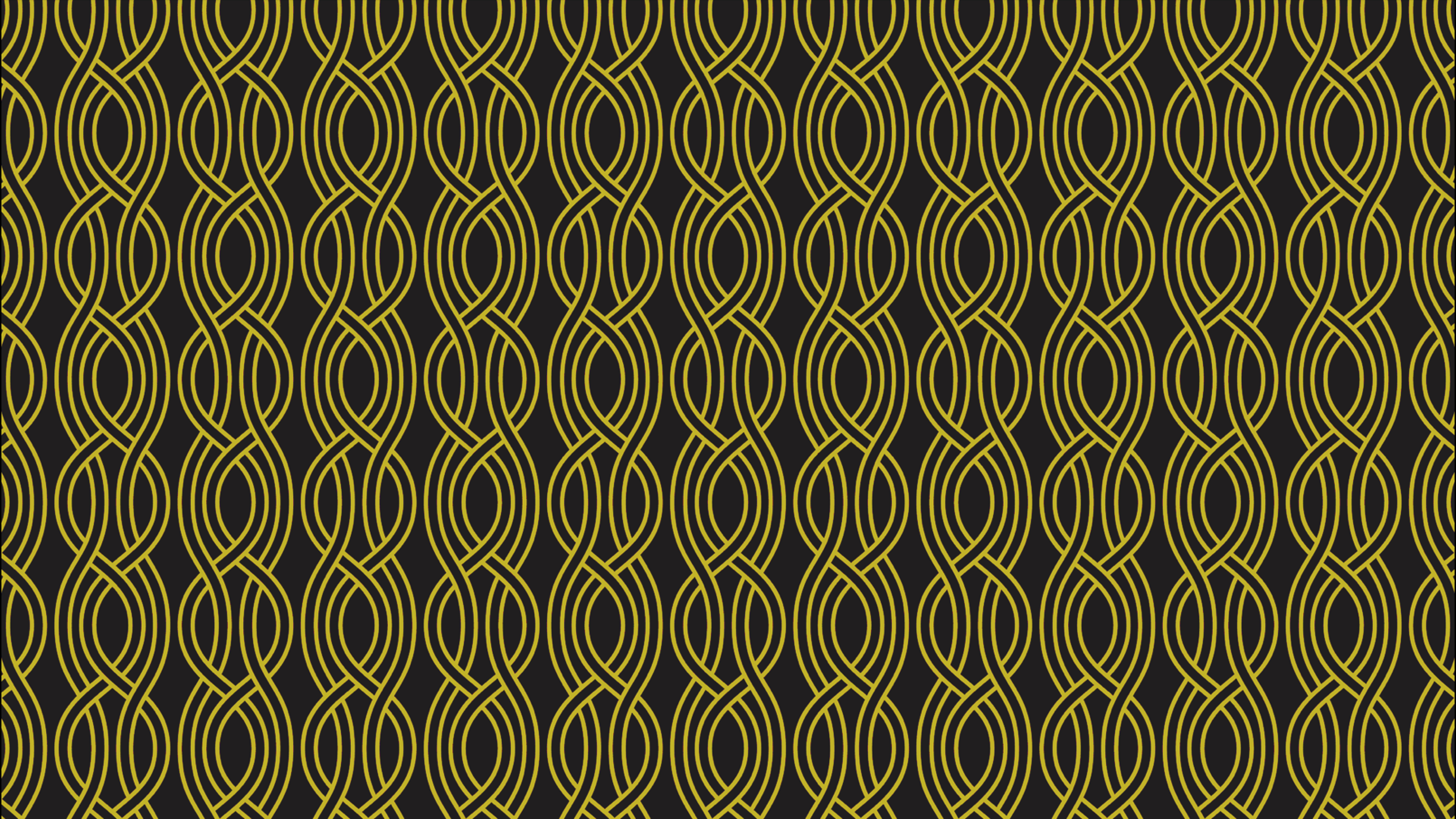
Data Architecture

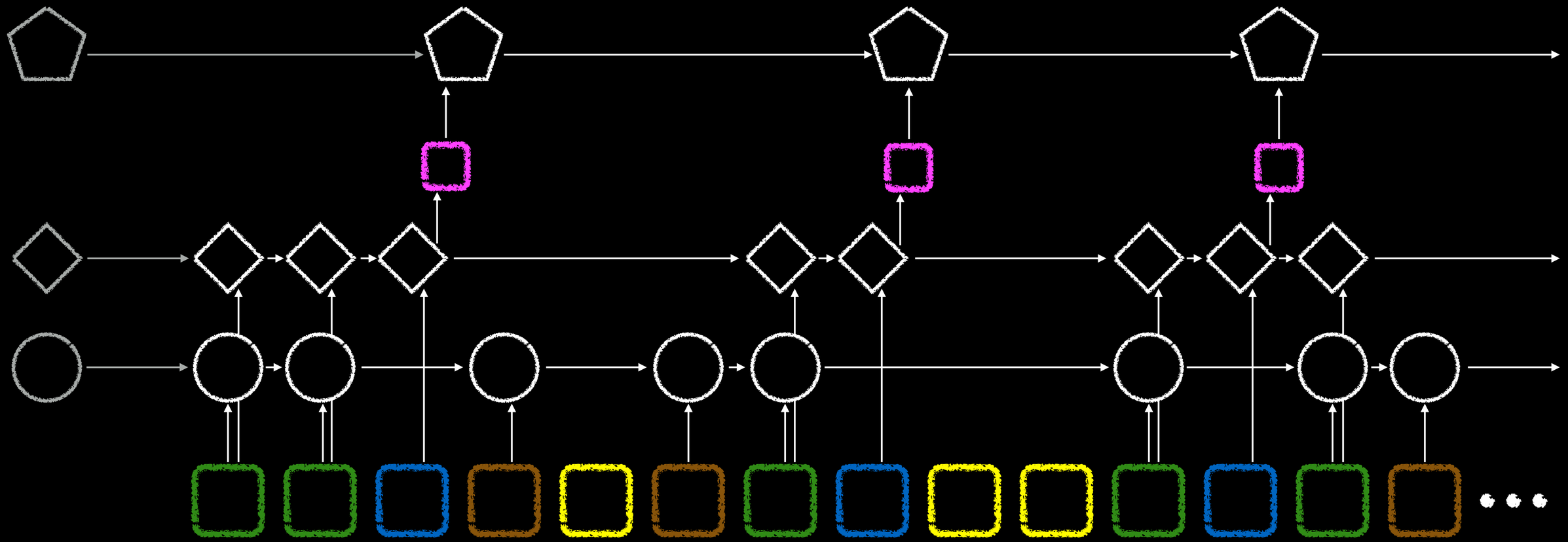
log a lot

ok to denormalize

think about the types

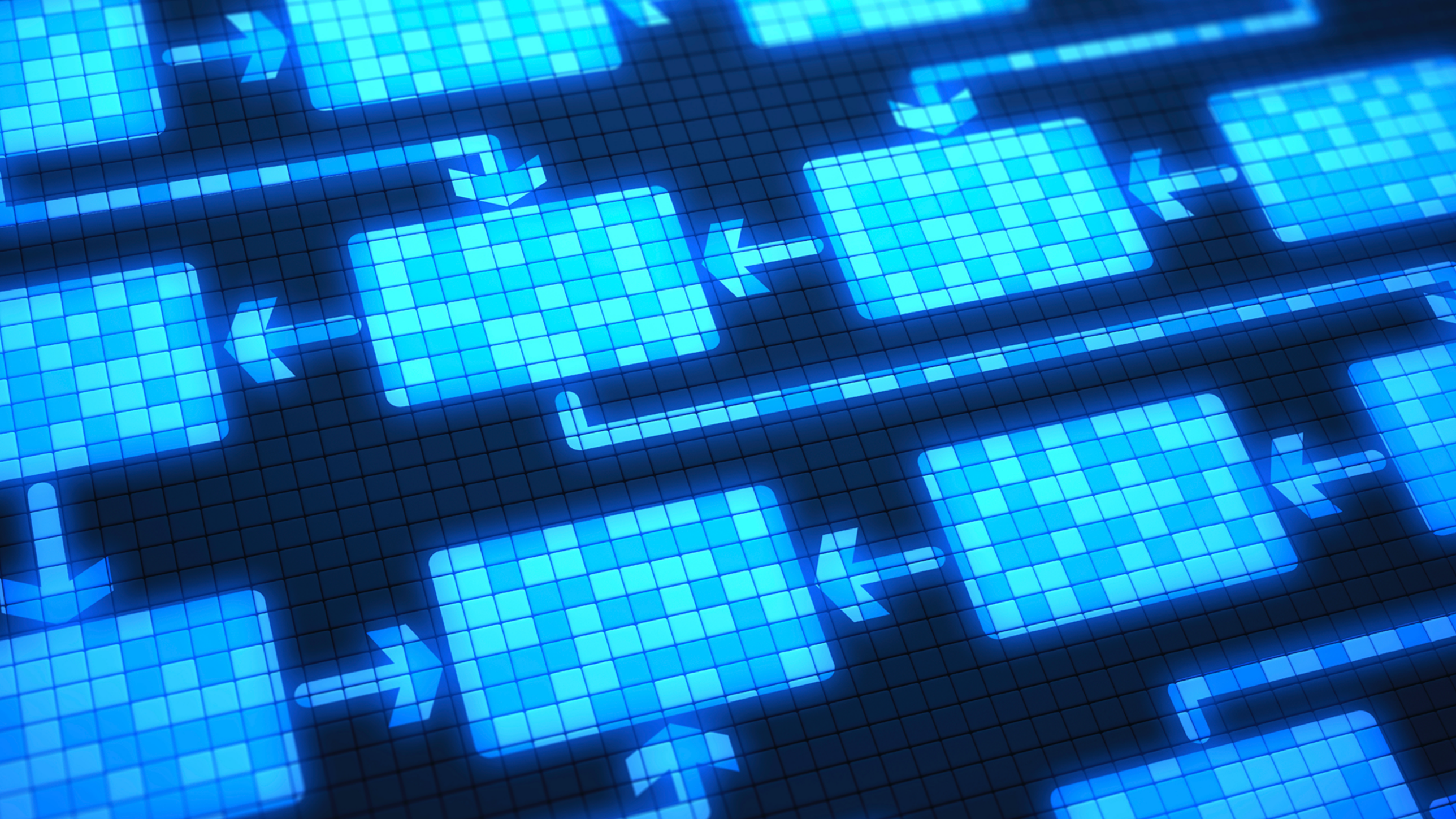
- Make sure that events are simple facts
- Files are ok for event history, don’t really need a database
- Use an object hierarchy to model events in code
- Use online features that are efficient to update incrementally
- Write efficient implementations before than scaling out
- Functional style makes it easier
- Encourage reactive processing





Data Quality

- Matters more than transformations, more than algorithms
- Data that doesn’t make sense often indicates an application bug
- Do assertions, e.g., make sure things aren’t happening out of order



- All data in form of events – no exceptions!
- Same feature code in production and development
- Emphasis on creative feature engineering
- Quick cycles between ideas and production

Try the Antelope Demo:
https://github.com/ifwe/antelope/blob/master/doc/demo.md

github.com/ifwe/antelope

@jssmith

if(we)