H_2O WORLD

2017

Sergei Izrailev

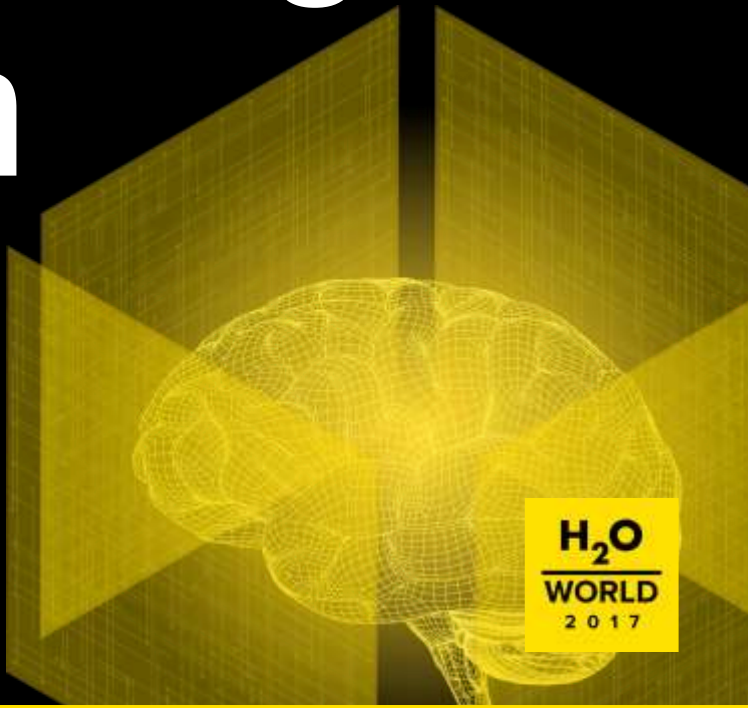
Chief Data Scientist @ Beeswax

@sizrailev / bit.ly/MLatScale / sergei@beeswax.com / www.beeswax.com

BEESWAX

H2O WORLD 2017

Design Patterns for Machine Learning in Production



BEESWAX

H_2O WORLD 2017

Motivation

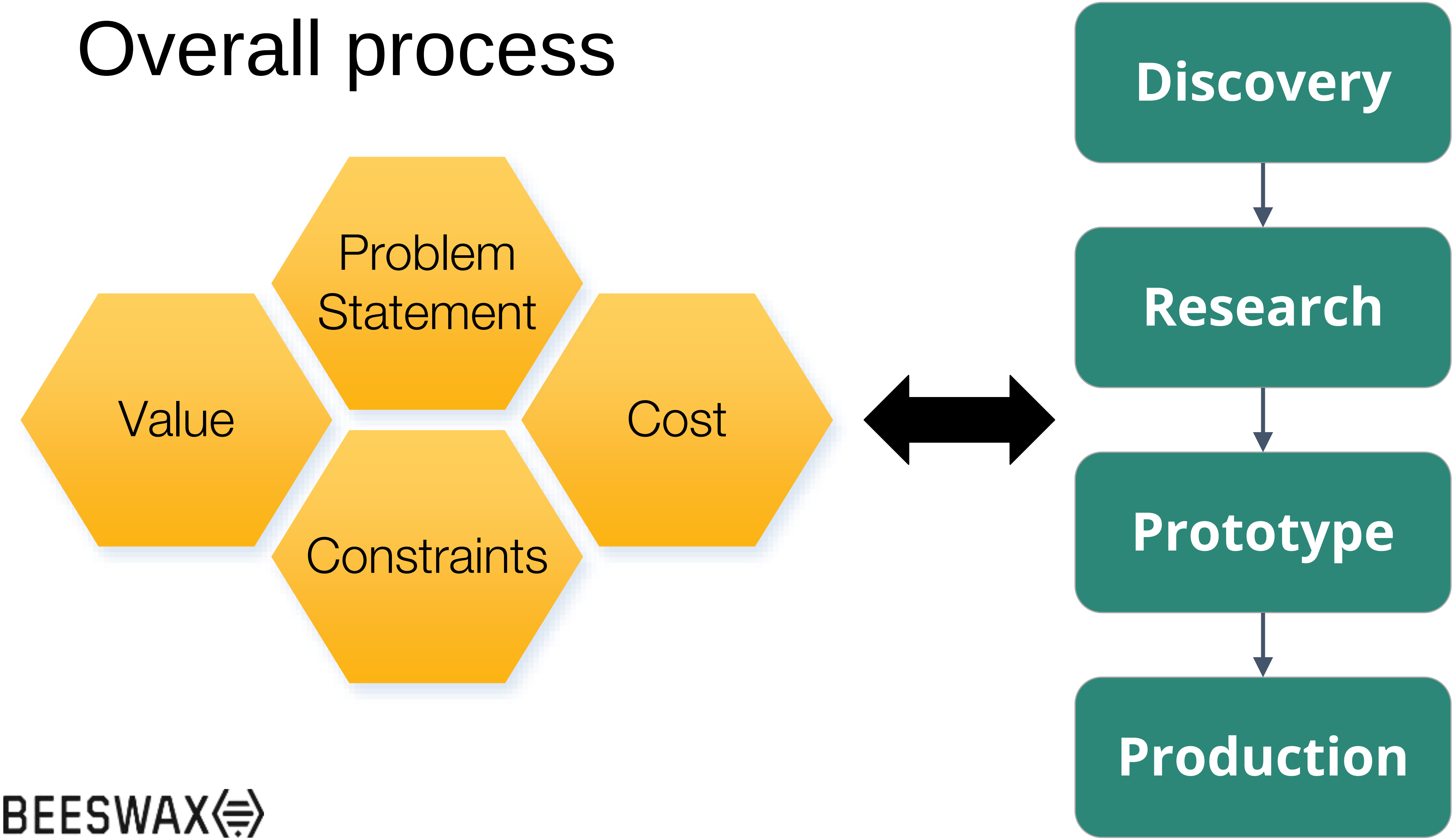
- A widespread need to leverage in-house data
- Data science expertise is available
- And yet, it seems to be too hard to extract value from ML

About

- Beeswax
  - Beeswax is an ad tech startup; 40 employees in NYC and London
  - Founded by three ex-Googlers
  - Real-time bidding (RTB) platform for buying online ads (1M+ QPS)
  - Platform tailored for customers to leverage in-house data science

- Myself
  - Production AI systems in Pharma, Finance and Ad Tech
  - Interested in both technology and organizations
  - bit.ly/MLatScale

Overall process



Start with defining the problem

Problem statement

- Is this the right problem to solve?
- Suppose, we’ve solved the stated problem - what’s the value?
- Is ML the right tool to solve the problem?
- What are the constraints?

Define constraints

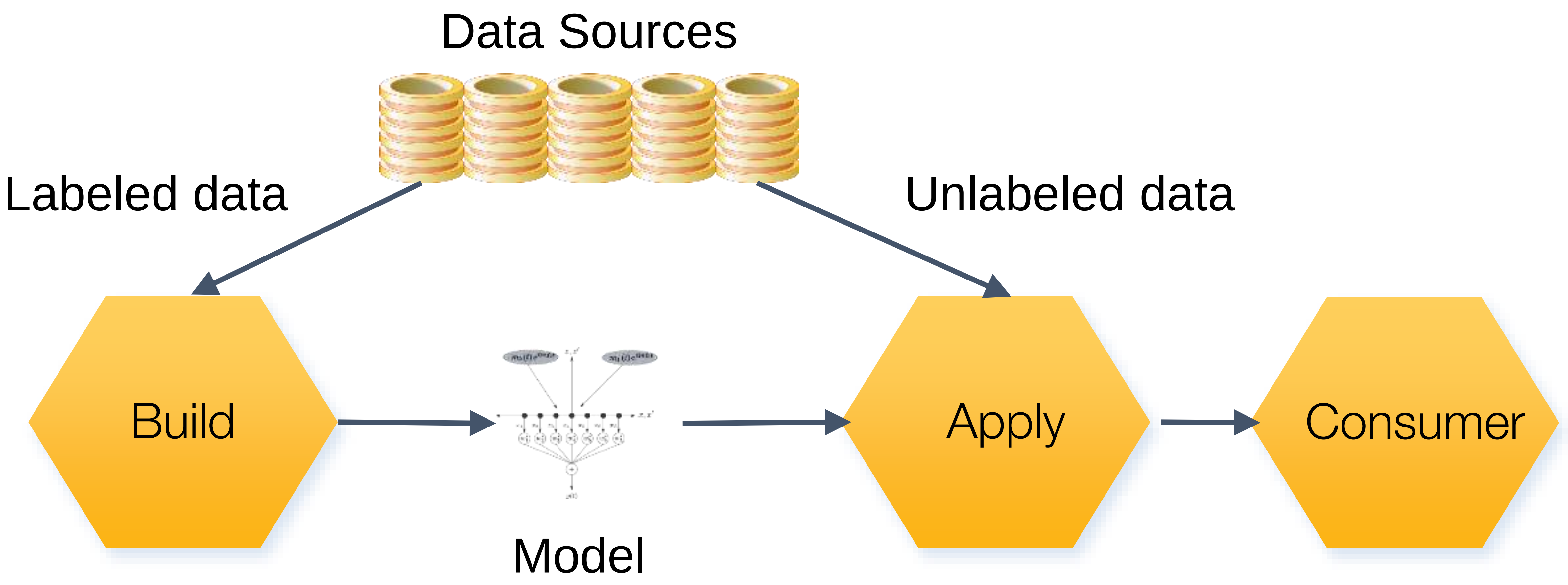
- Existing production environment architecture
- Technology stack
- Available people and their skills
- Requirements for scale

Dimensions of ML system scalability

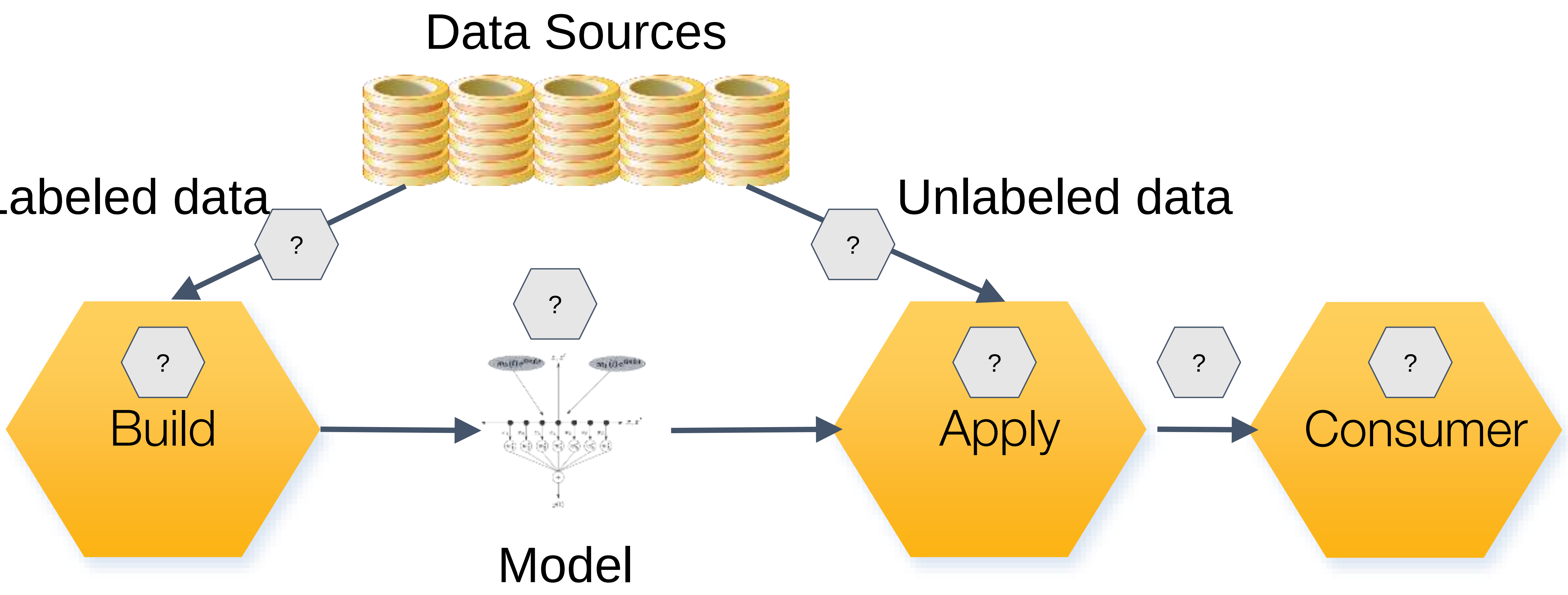
- **Volume**: how much data do we need to process?
- **Velocity**: how quickly does the data change?
- **Variety**: what are the types of data, models, and applications?
- **Veracity**: how accurate are our models?
- **Value**: how does it matter to the ML consumer?
- **Viability**: do the benefits outweigh the costs?

Technical Design of ML Systems

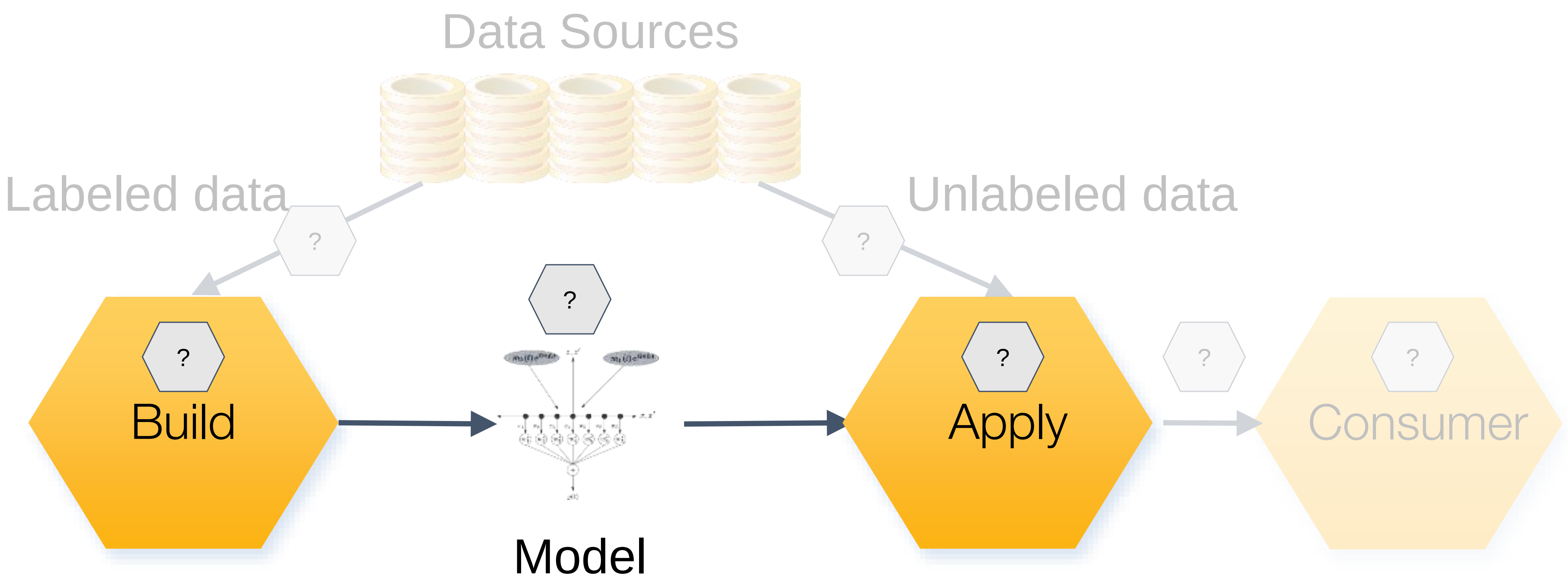
Machine learning systems



ML system design



Model deployment

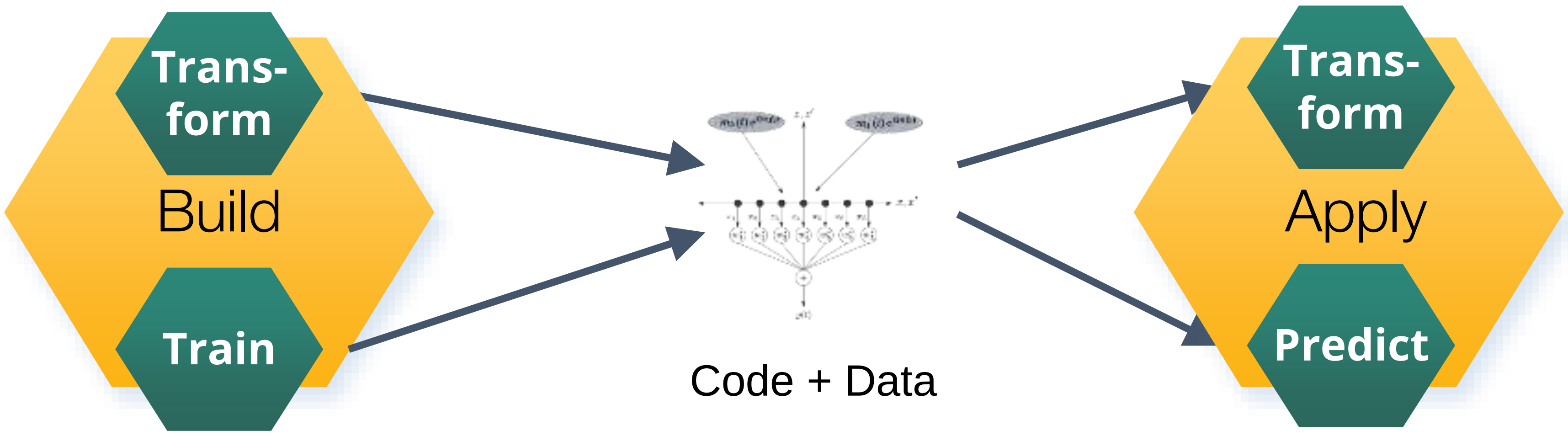


BEESWAX

H2O WORLD 2017

Model deployment

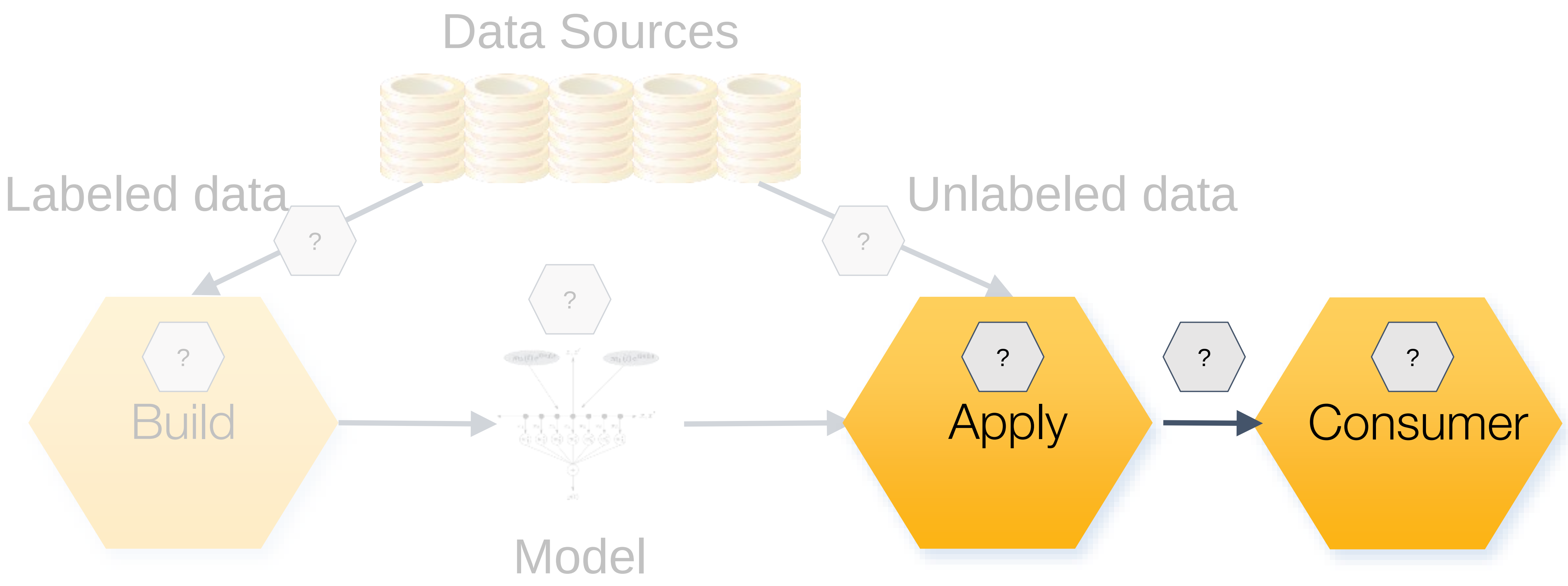
- Data transformations must be the same in training and scoring
- Some transformations are “models” (PCA, top N, TF-IDF)
- Hence, most ML pipelines are DAGs
- These DAGs must be reproduced in production scoring



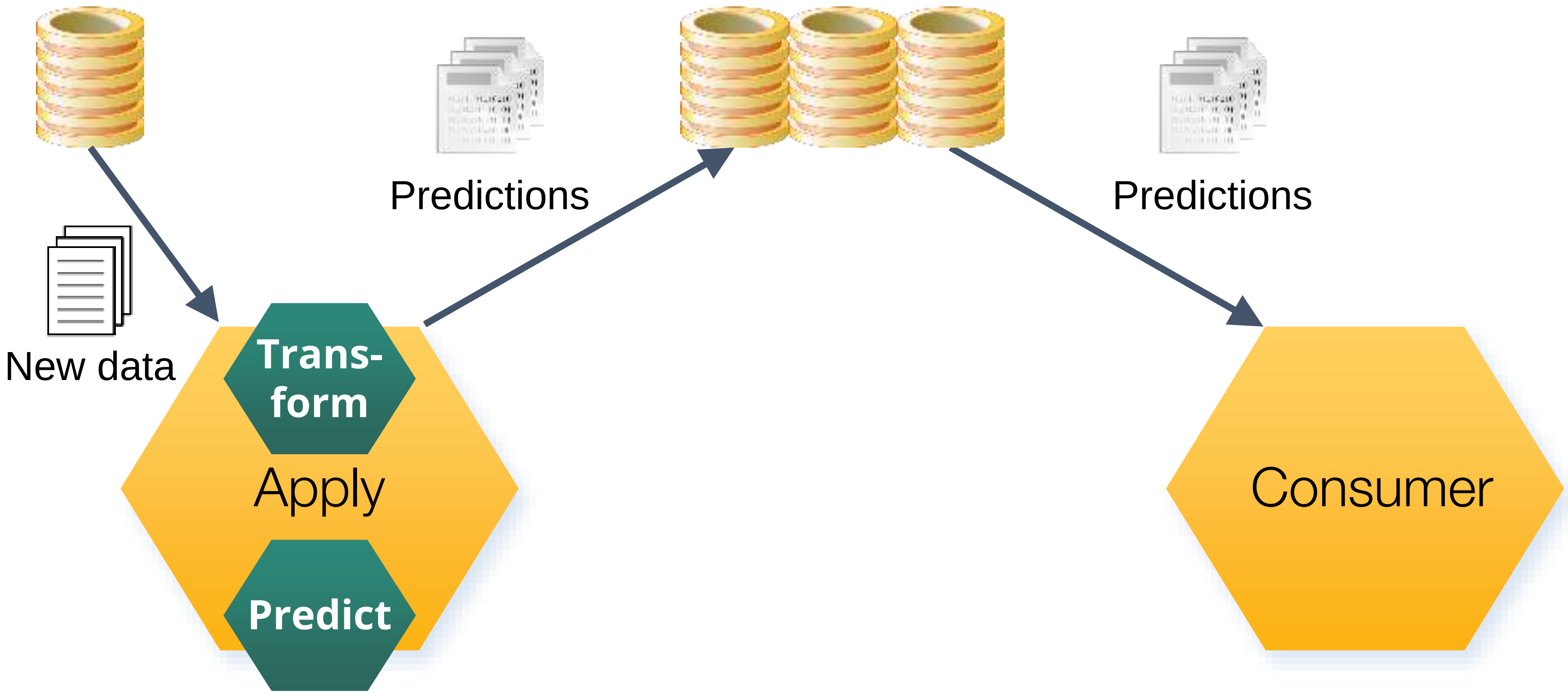
Interface between building and scoring

- In-memory - model is never persisted, train then score
  - single application, also streaming
- Data only - linear coefficients, PMML, etc
  - code is independent
- Serialized objects - Pickle, R, Spark, custom
  - reuse code
- Code + Data - e.g., H2O’s POJO
  - code is generated

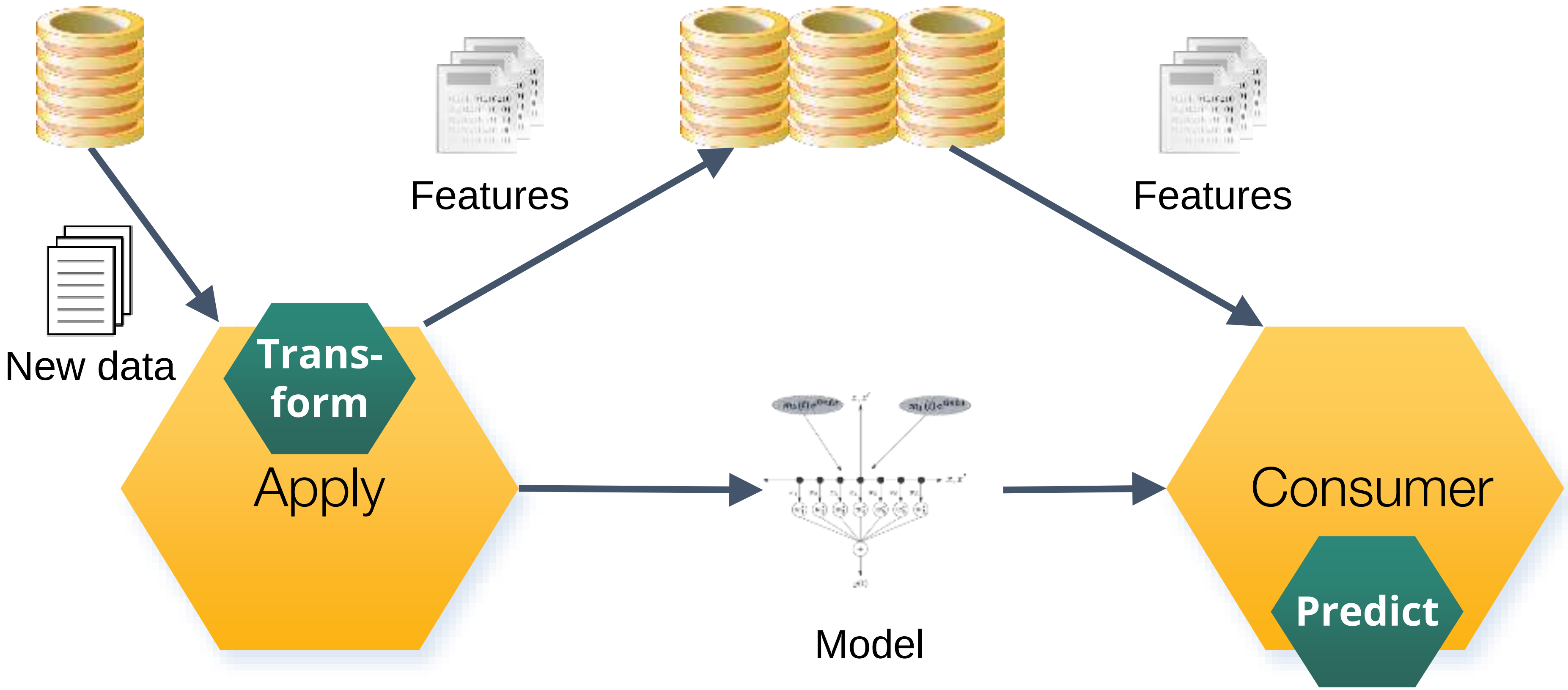
Scoring systems



Batch processing



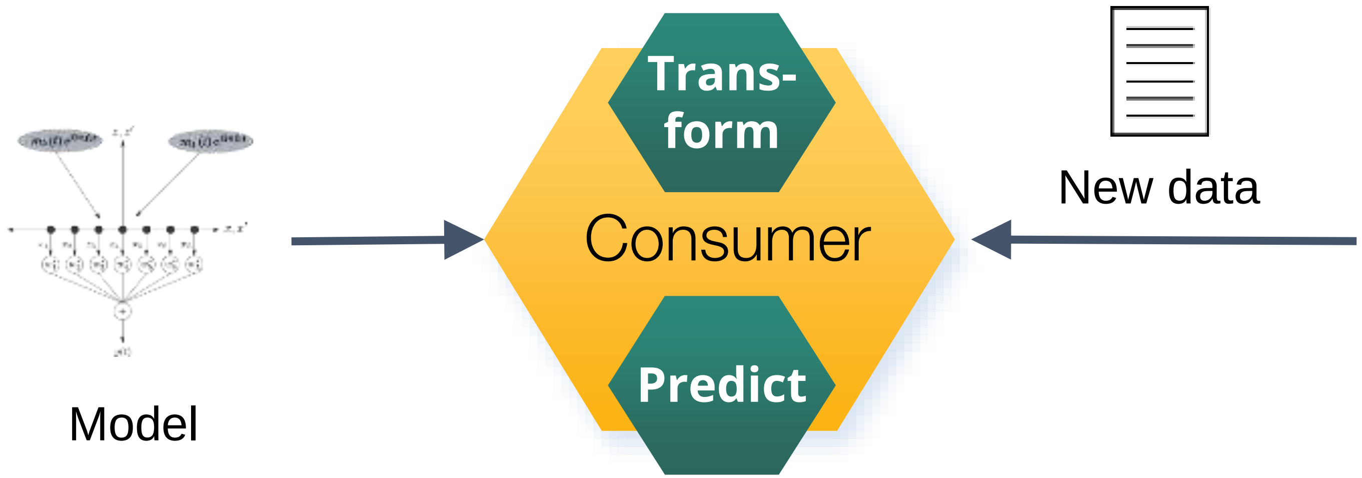
Batch features; consumer predicts



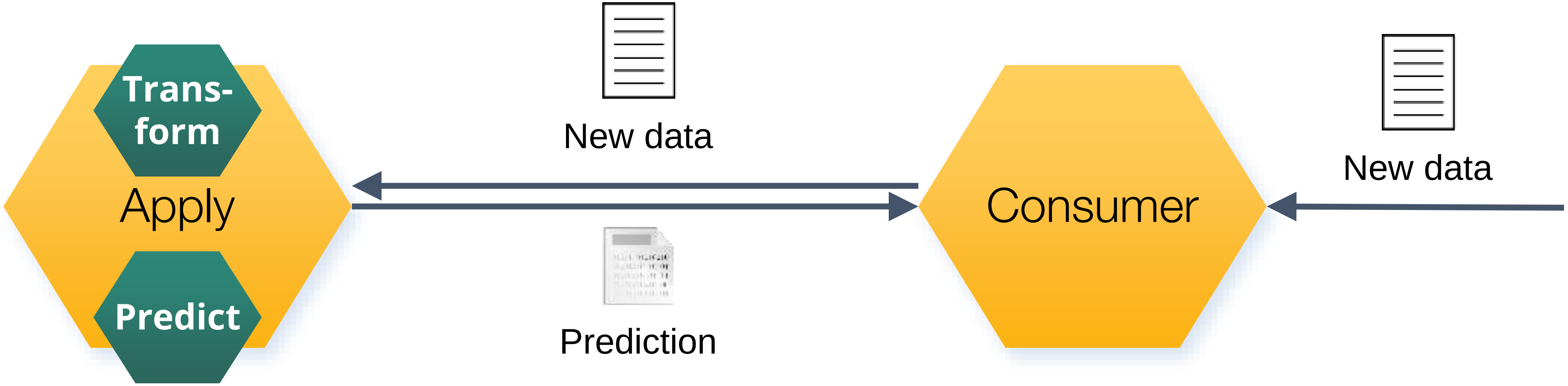
BEESWAX

H2O WORLD 2017

Single-row predictions



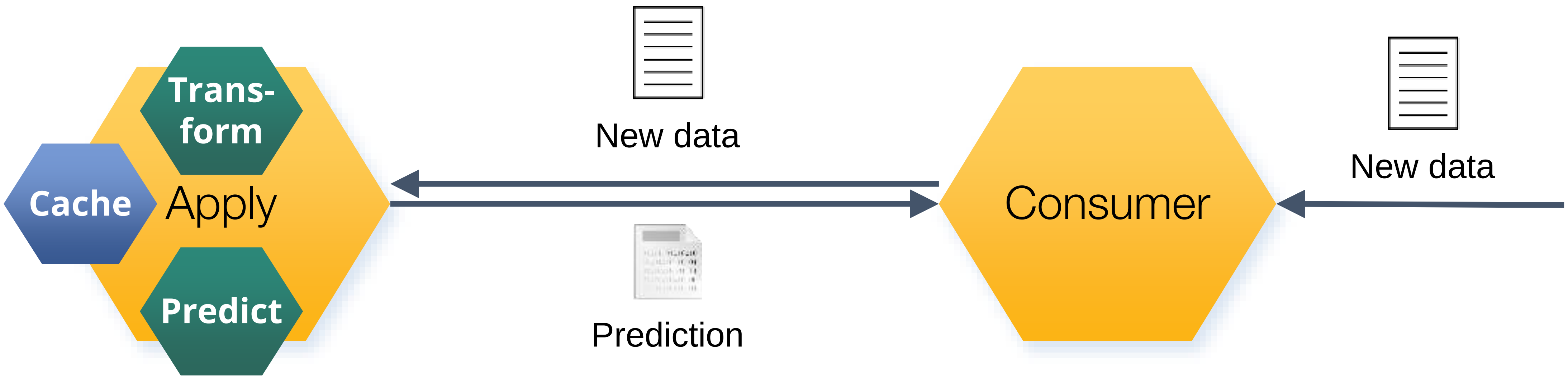
A service for a single row consumer



BEESWAX

H2O WORLD 2017

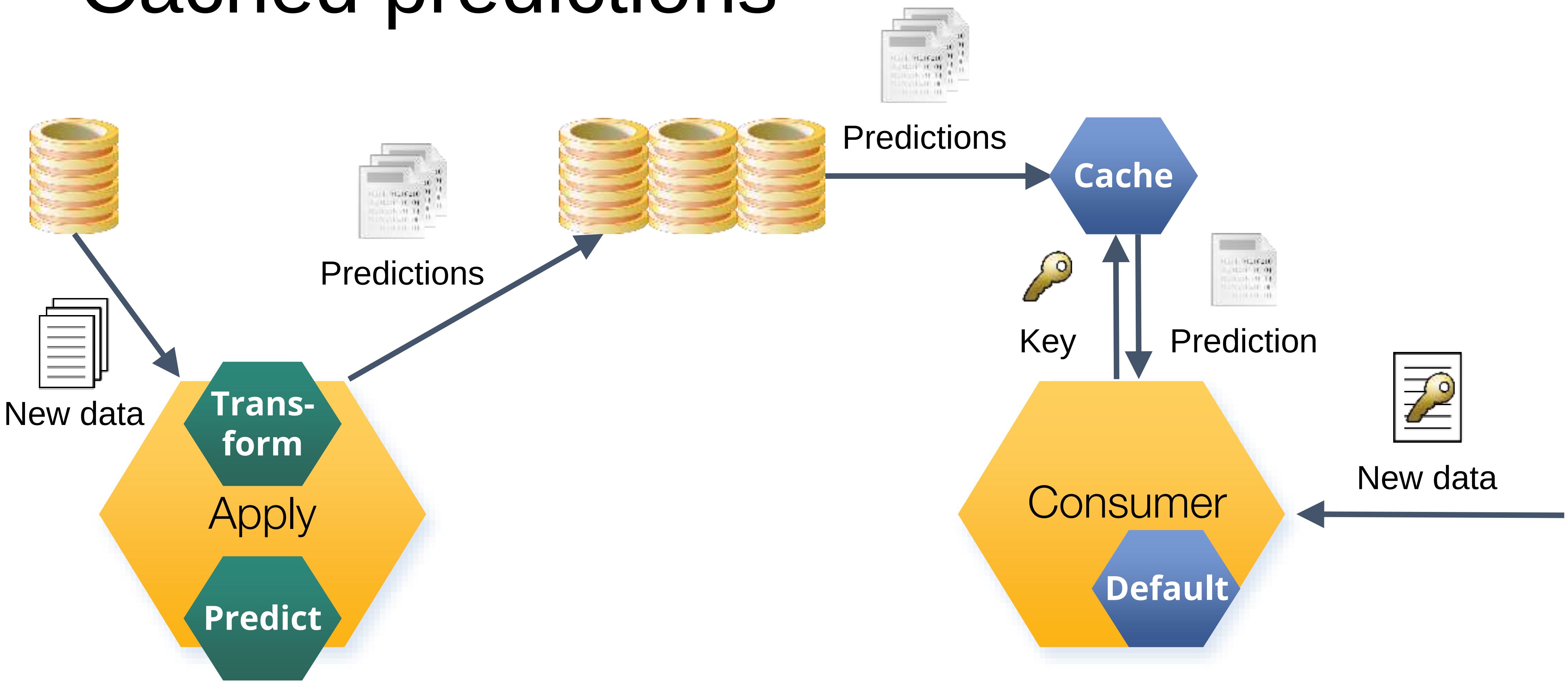
A service for a single row consumer



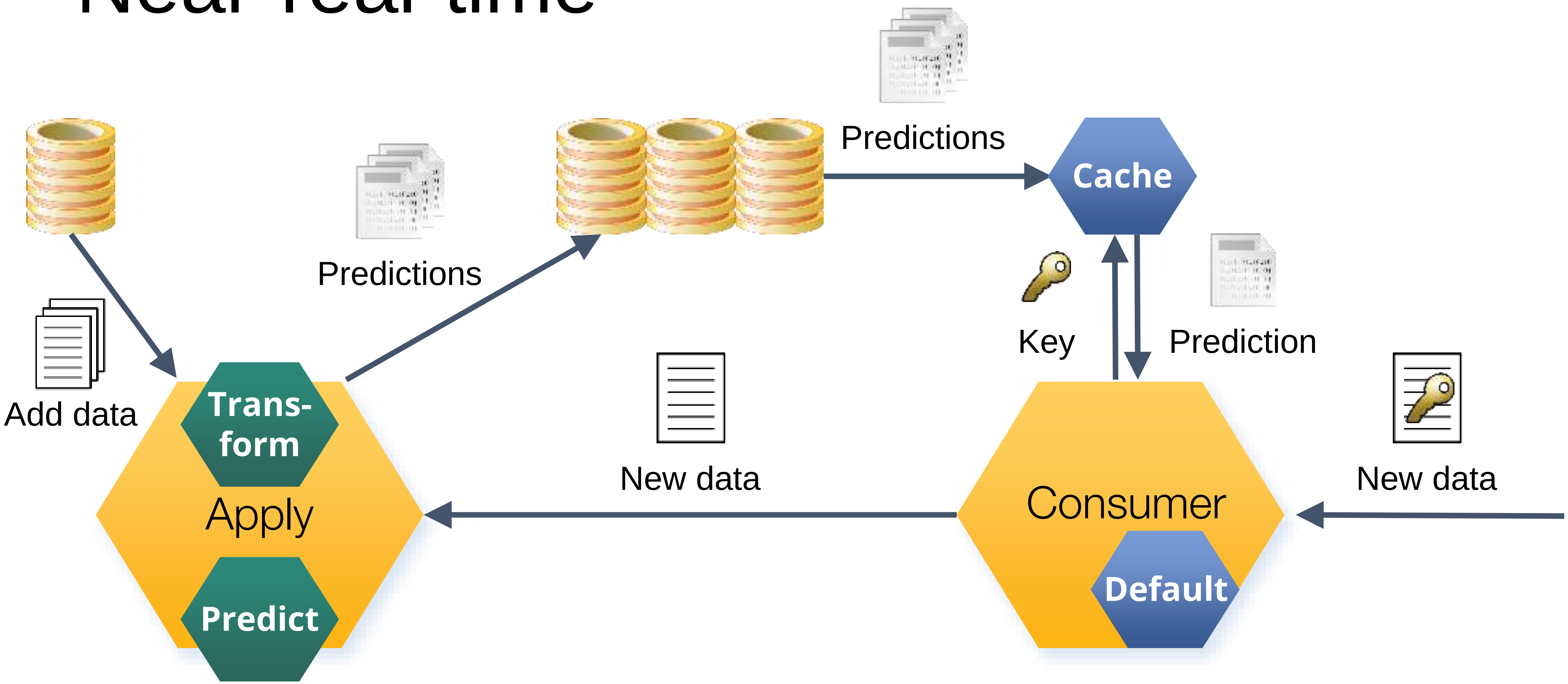
BEESWAX

H2O WORLD 2017

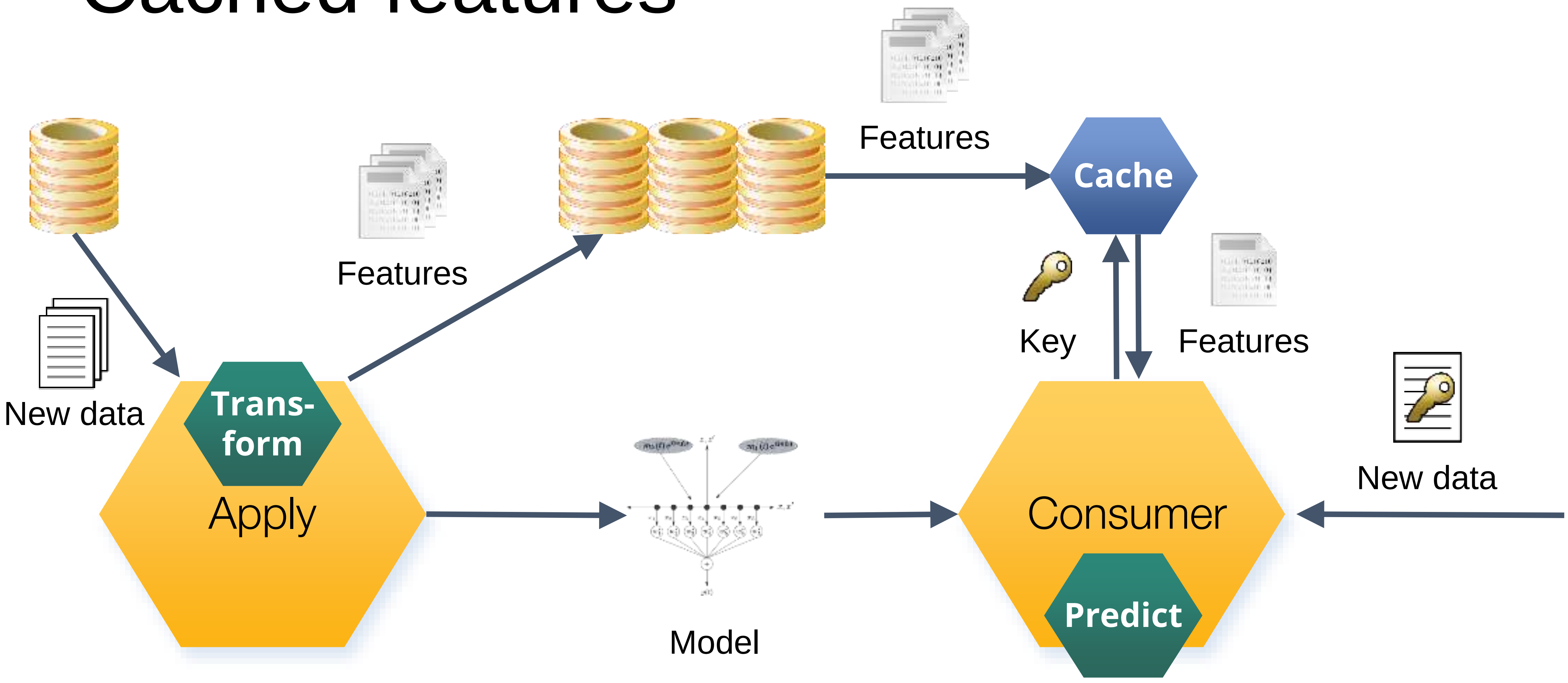
Cached predictions



Near-real-time



Cached features

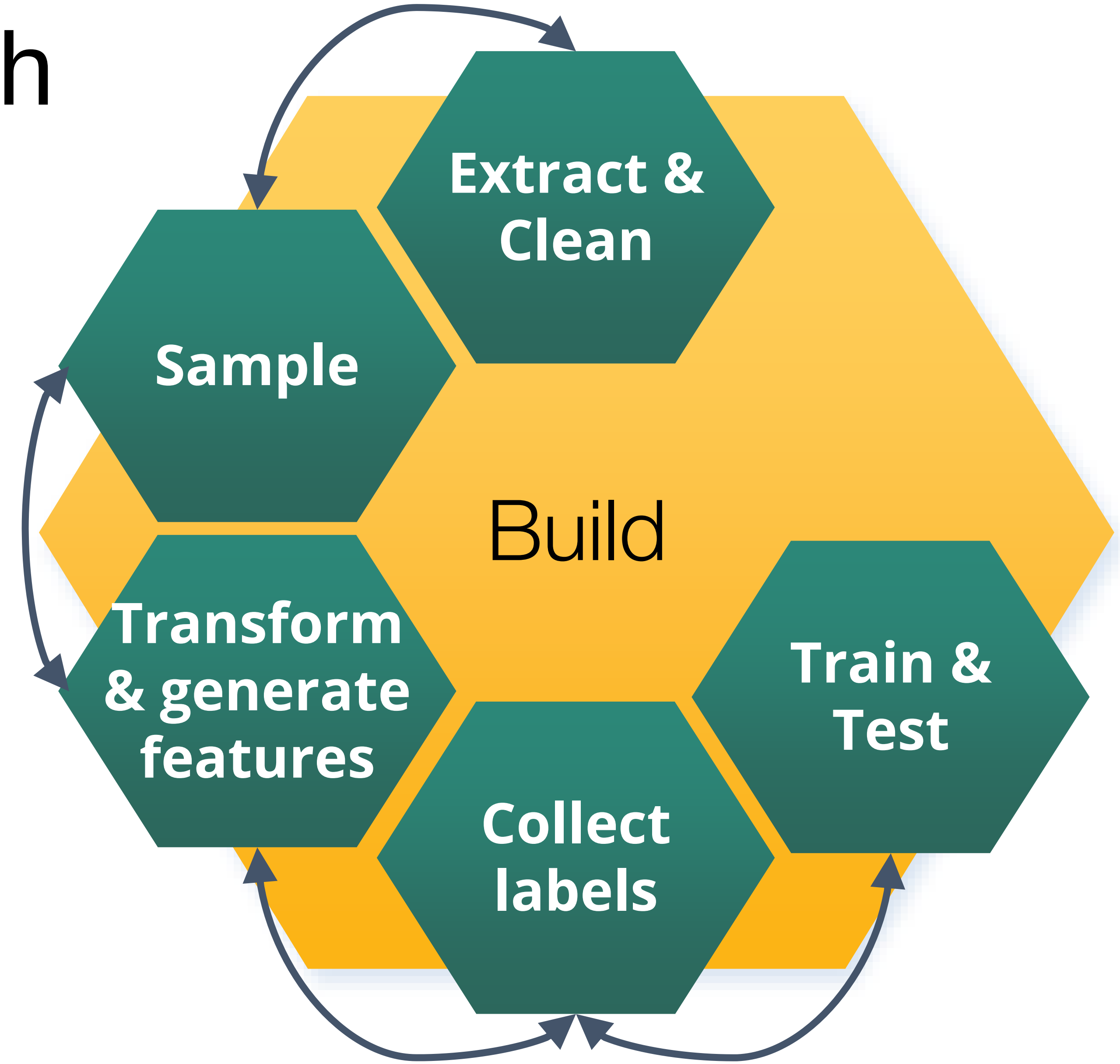


BEESWAX

H2O WORLD 2017

Evolution of ML systems

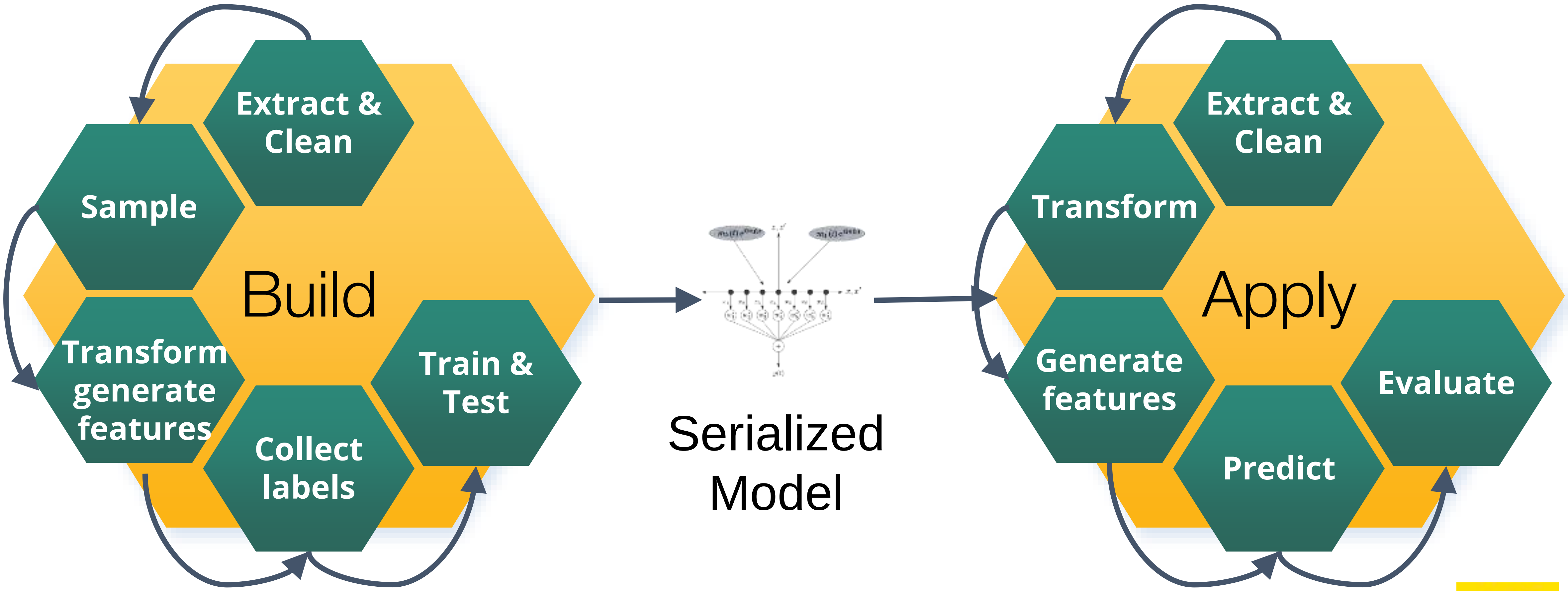
Research



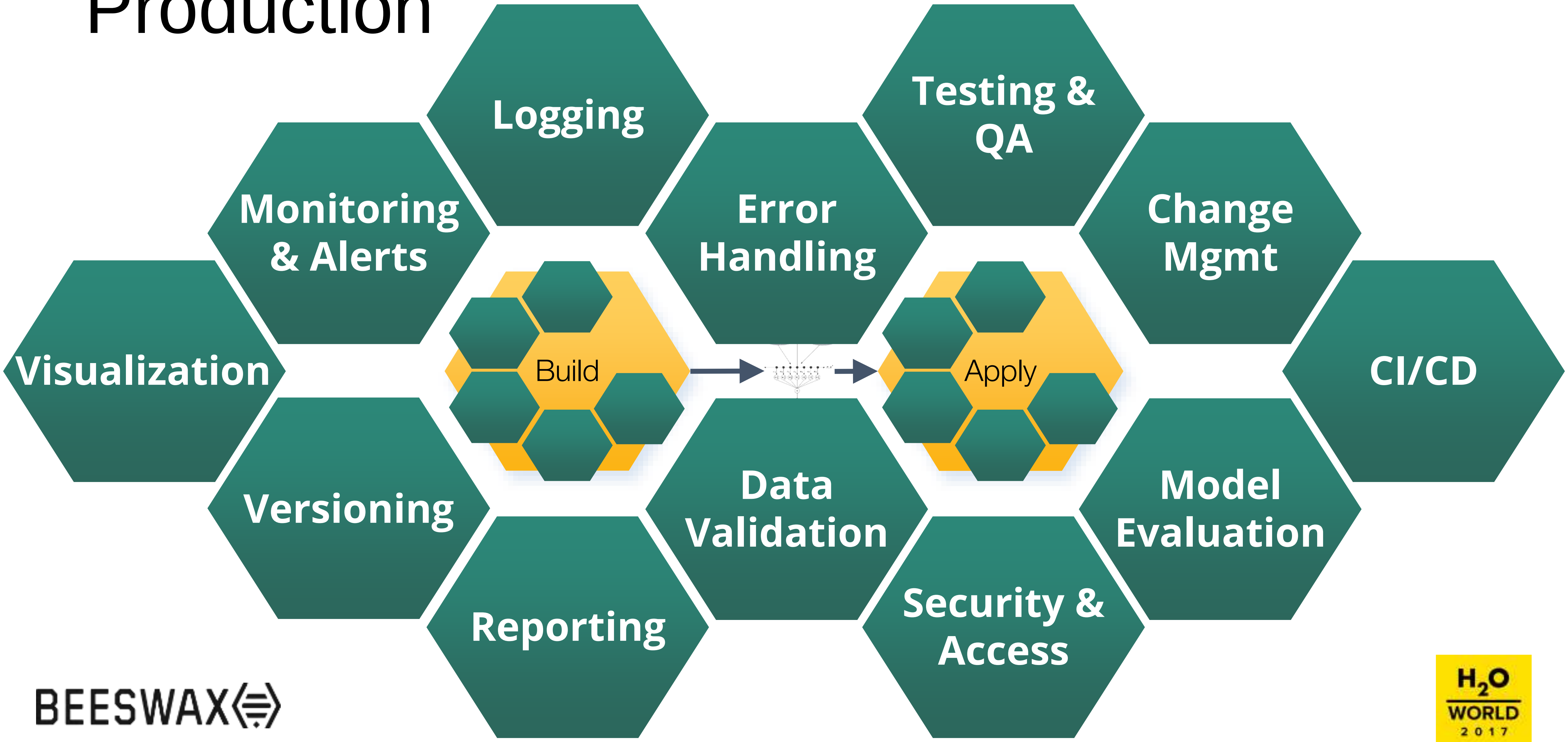
BEESWAX

H2O WORLD 2017

Prototype


Extract & Clean
Sample
Build
Transform generate features
Collect labels
Train & Test
Serialized Model
Extract & Clean
Transform
Apply
Generate features
Predict
Evaluate

Production



Logging
Testing & QA
Monitoring & Alerts
Error Handling
Change Mgmt
Visualization
Build
Apply
CI/CD
Versioning
Data Validation
Model Evaluation
Reporting
Security & Access

Production
Fault-tolerant systems

- We should expect problems
  - models don’t converge; input data changes; bugs
- Produce acceptable results - even when something fails
- Handle predictable error conditions automatically
- Minimal human intervention
- Easy diagnostics and recovery in case of fatal errors

Conway's law:
"Any piece of software reflects the organizational structure that produced it."

People Questions

- Who is developing training?
- Who is developing scoring?
- Who is responsible for training in production?
- Who is responsible for scoring in production?
- Who deploys new models and model updates?
- Who is responsible for quality control?

People's Functions

Product management

Data science

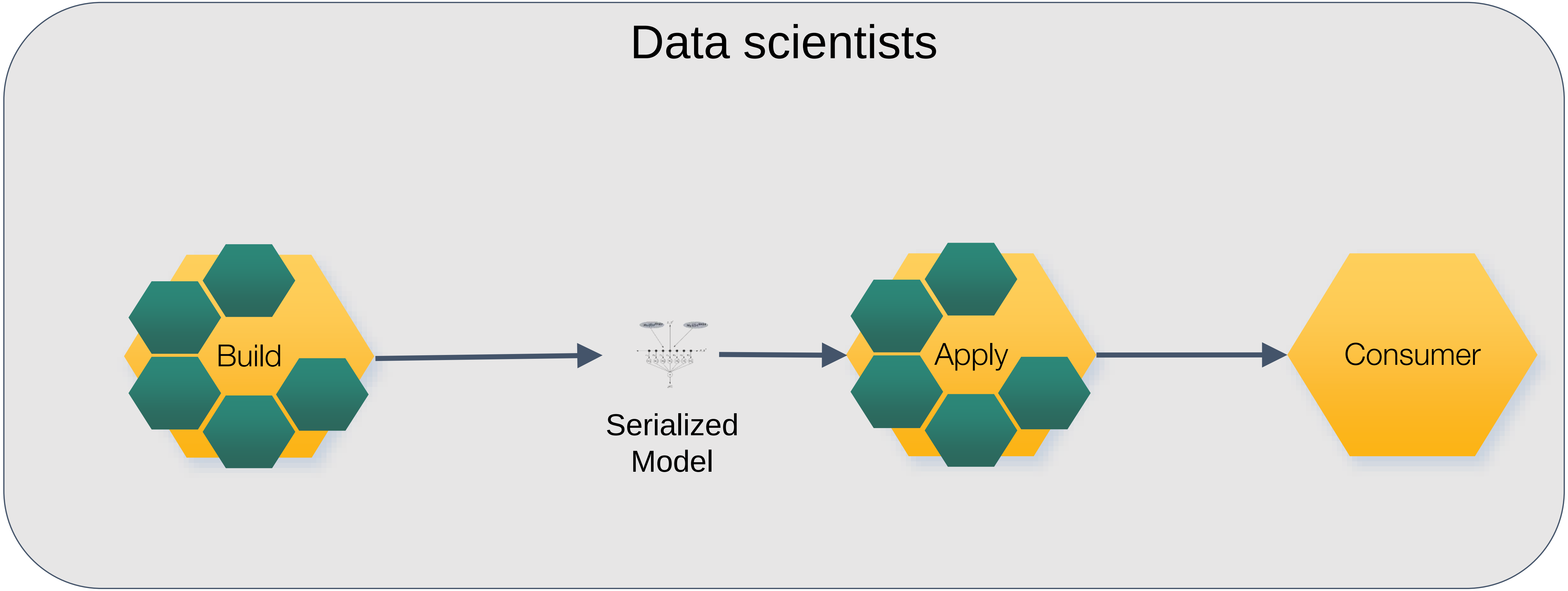
Data engineering

Application engineering (RT server-side applications, client-side applications)

UX

- Front-end development
- Data collection (e.g., logging)
- Code deployment
- Testing and QA
- Infrastructure provisioning
- Support

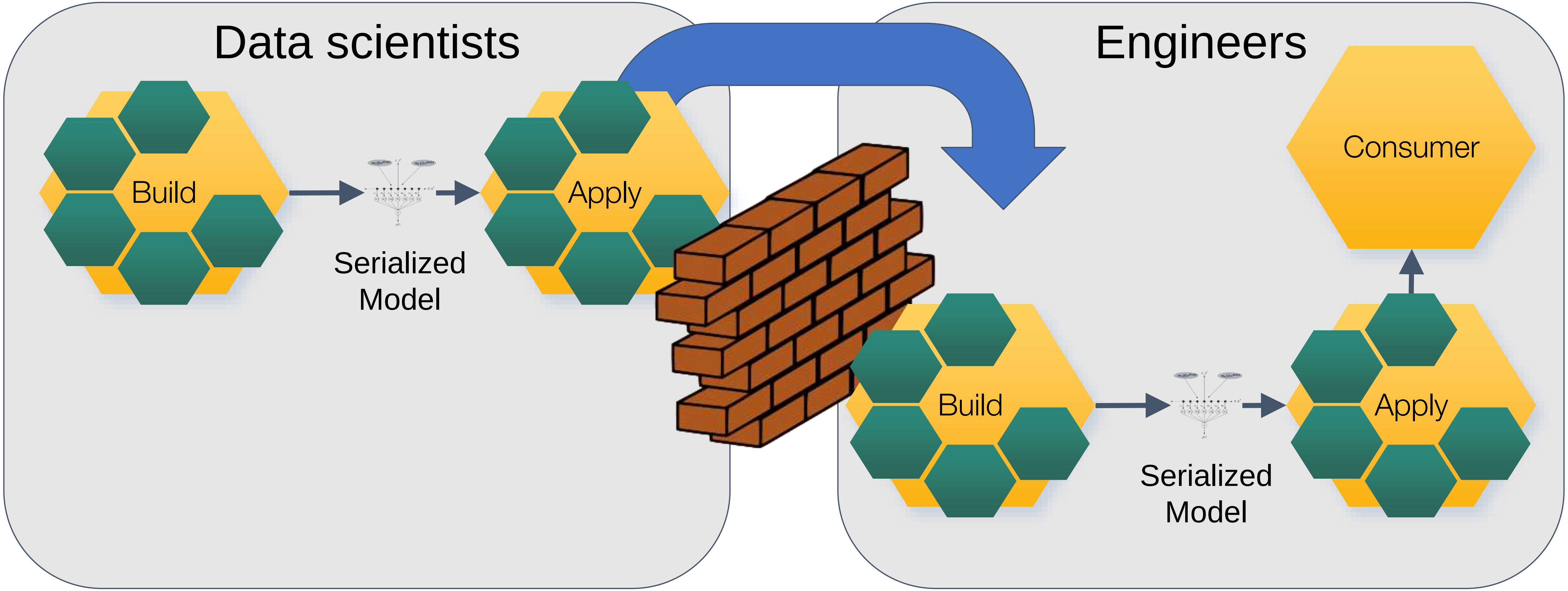
Data scientists as consumers



BEESWAX

H_2O WORLD 2017

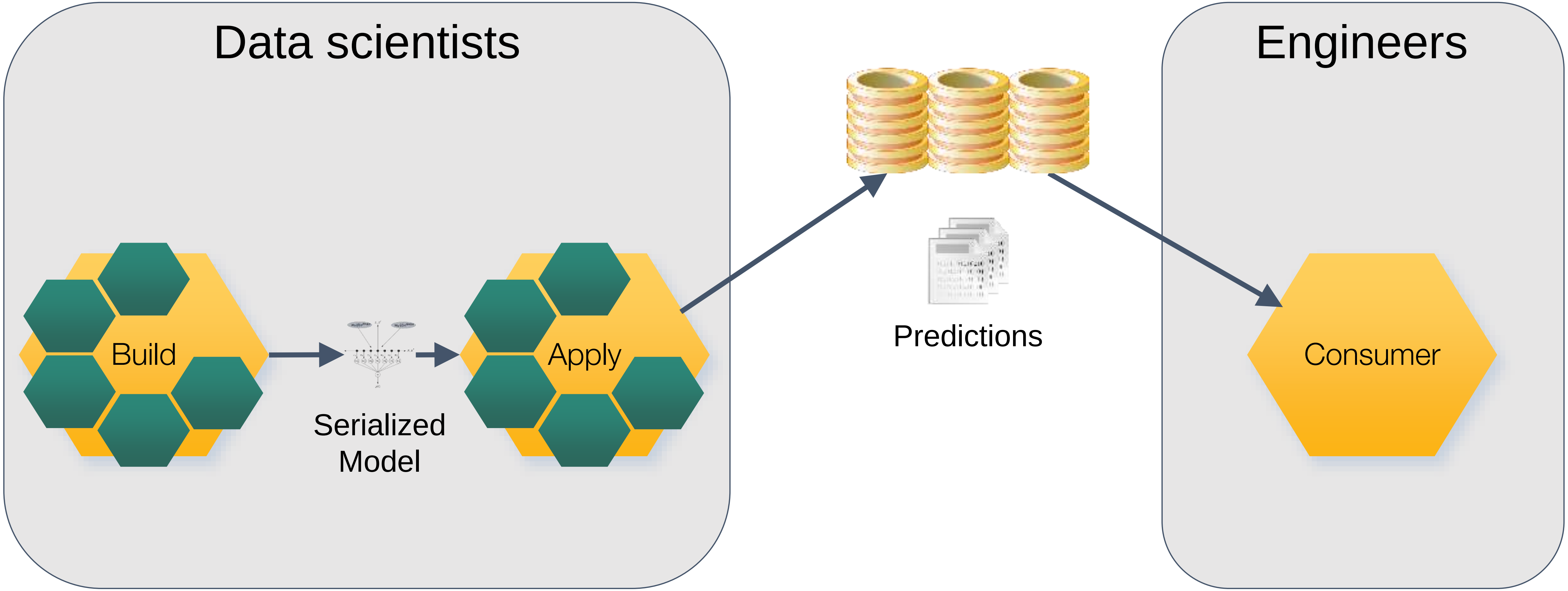
Over the wall



BEESWAX

H2O WORLD 2017

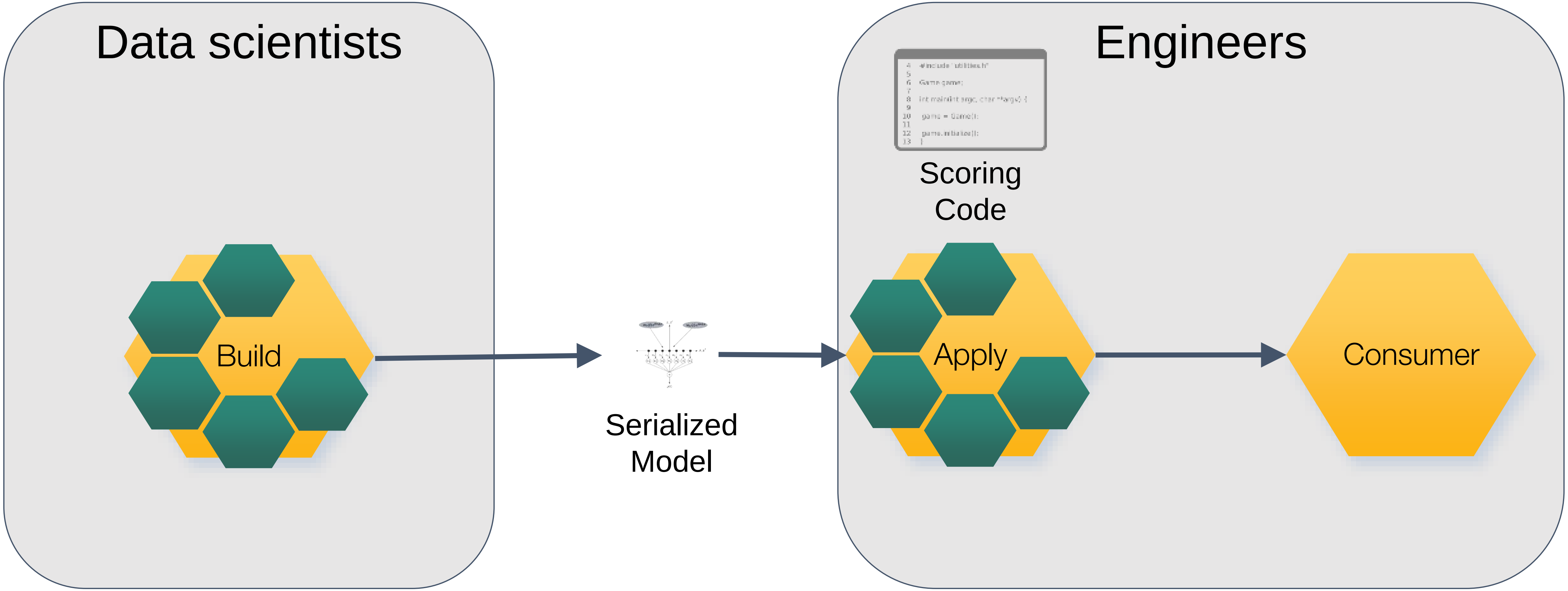
Deliver predictions only (aka “black box”)



BEESWAX

H_2O WORLD 2017

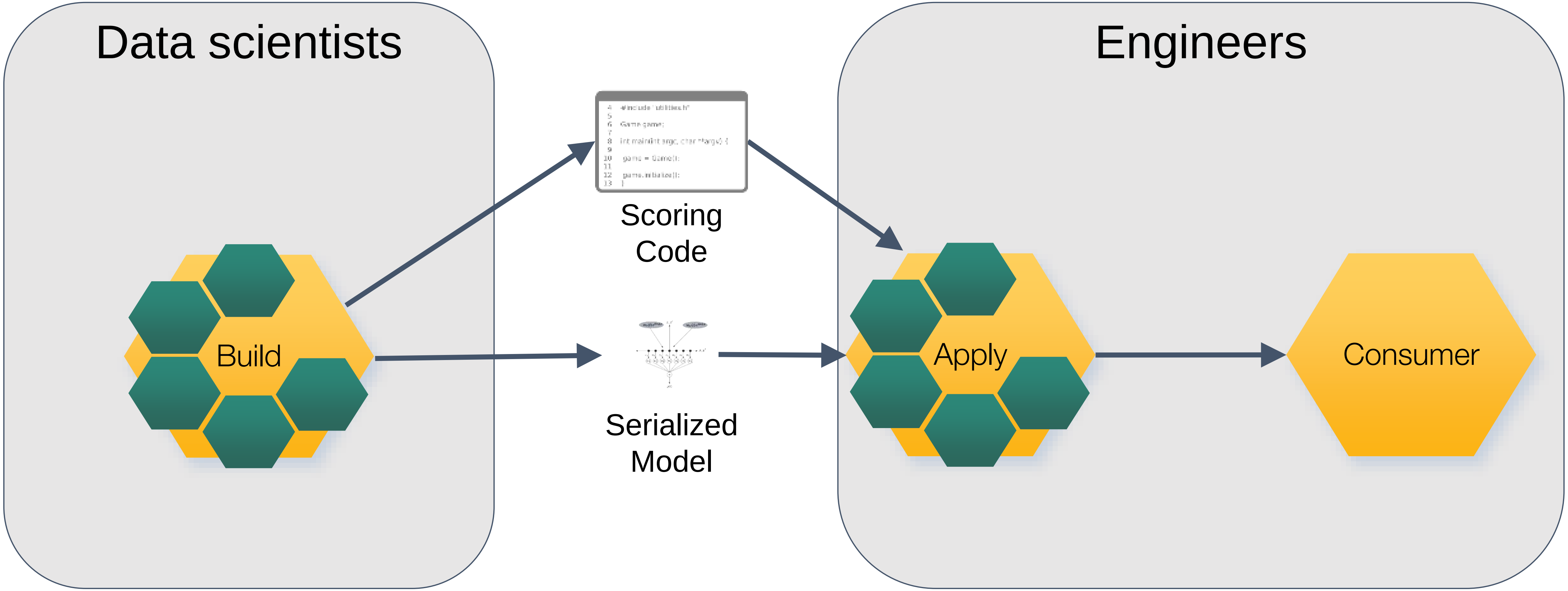
Deliver data (serialized model)



BEESWAX

H2O WORLD 2017

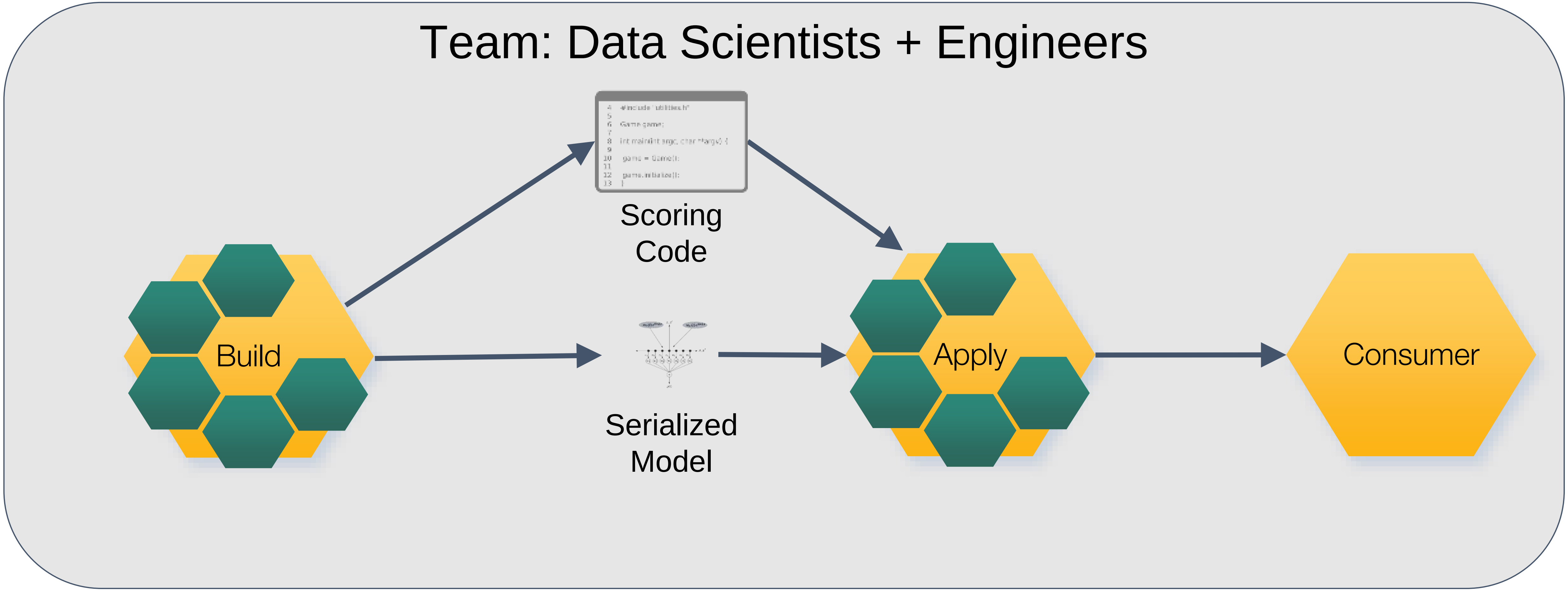
Deliver code and data



BEESWAX

H2O WORLD 2017

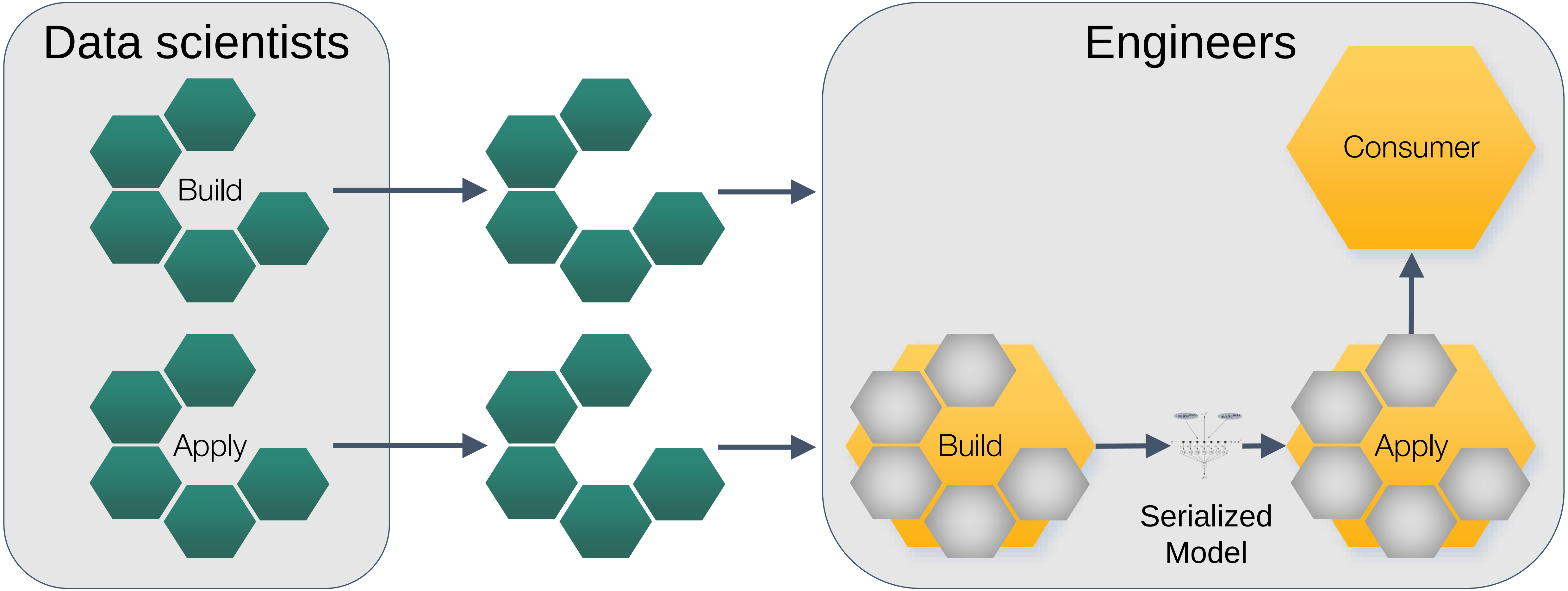
Cross-functional team



BEESWAX

H2O WORLD 2017

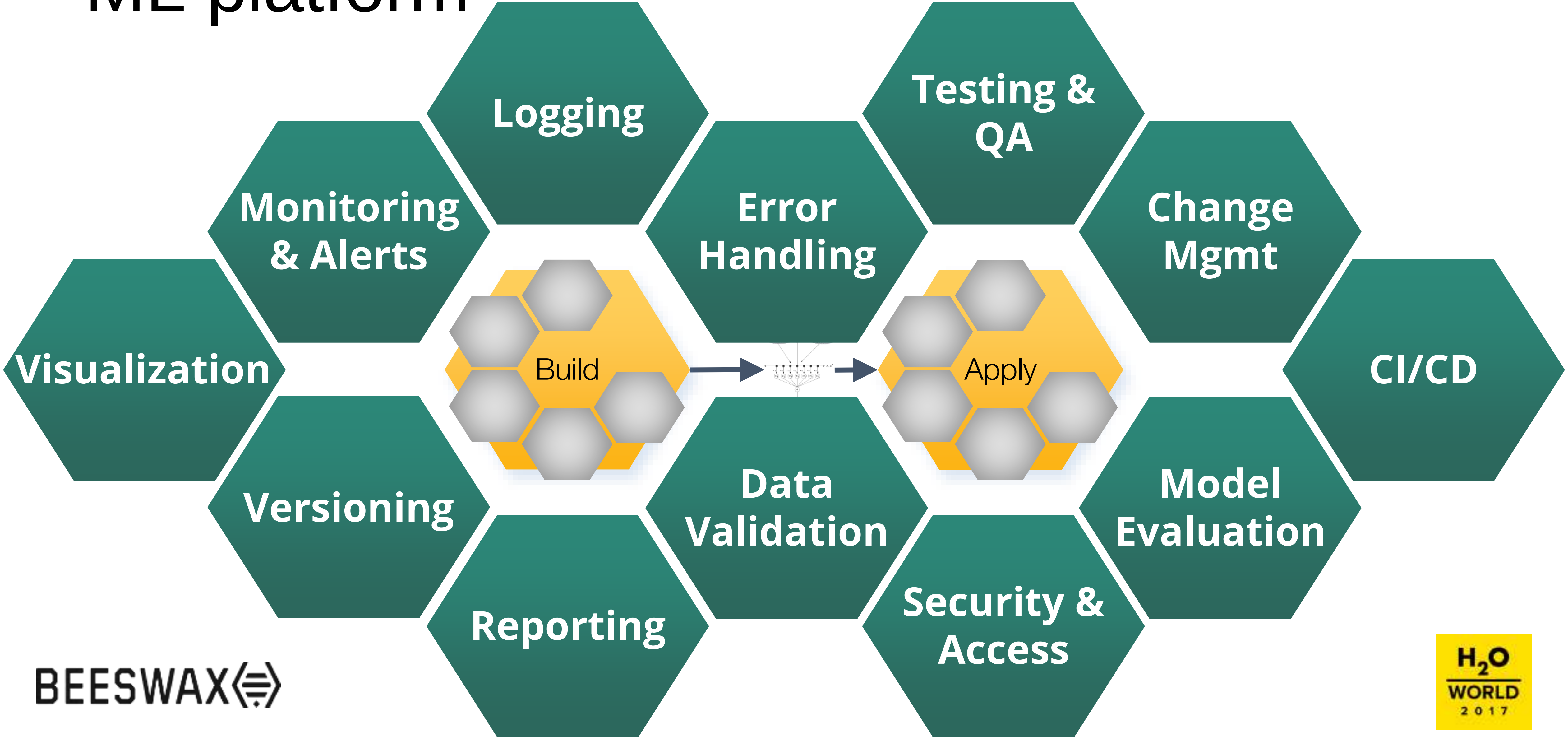
ML platform



BEESWAX

H2O WORLD 2017

ML platform



BEESWAX

H_2O WORLD 2017

Conclusion

- Find the right problem
- Define constraints
- Design components and interfaces
- Take into account organizational constraints
- Production can’t be an afterthought
- The process is a lot of work, but it’s not rocket science

Questions?

Yes, we are hiring…

sergei@beeswax.com

BEESWAX

H$_2$O WORLD 2017