PayPal AI Compute Platform

Simon Zhang, Senior Engineer Manager, Architecture and Infrastructure

QCON Shanghai, Oct/19 2018

What is PayPal?

2017 Full-Year Results

$13.06B†
REVENUE

$456B
TOTAL PAYMENT VOLUME[1]

7.8B
PAYMENT TRANSACTIONS[2]

$155B
MOBILE PAYMENT VOLUME

2.7B
MOBILE PAYMENT TRANSACTIONS

Prior period metric results for Total Payment Volume and Payment Transactions have been revised to reflect the updated definitions of the metrics. For additional details, please see PayPal's Current Report on Form 8-K filed with the Securities and Exchange Commission on April 10, 2018.

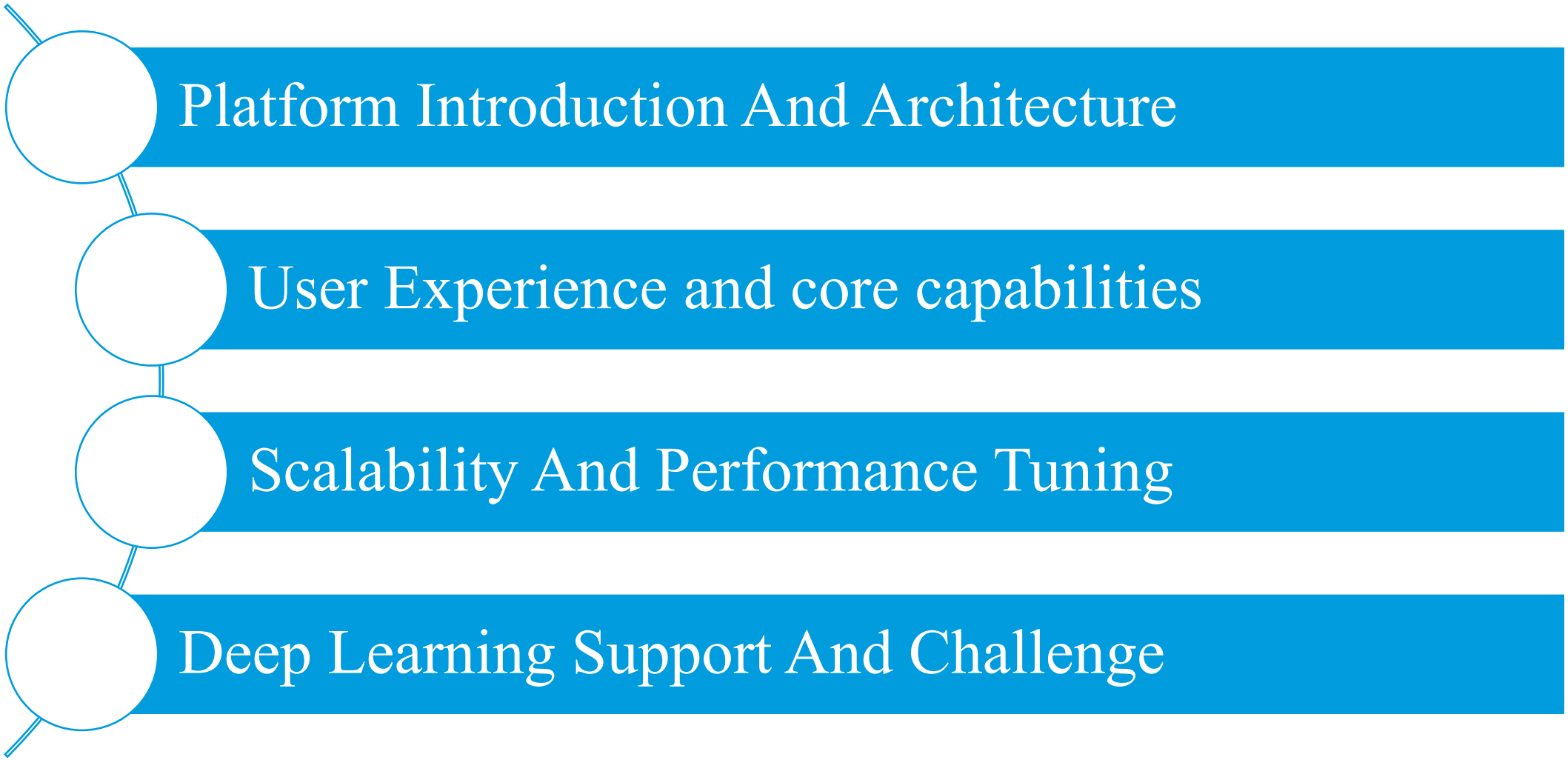
†Non-GAAP.
1 **Total Payment Volume (TPV):** The value of payments, net of reversals, successfully completed through our Payments Platform or enabled by PayPal via a partner payment solution, not including gateway exclusive transactions.
2 Payment Transactions: The total number of payments, net of payment reversals, successfully completed through our Payments Platform, excluding transactions processed through our gateway and Paydiant products.

PayPal

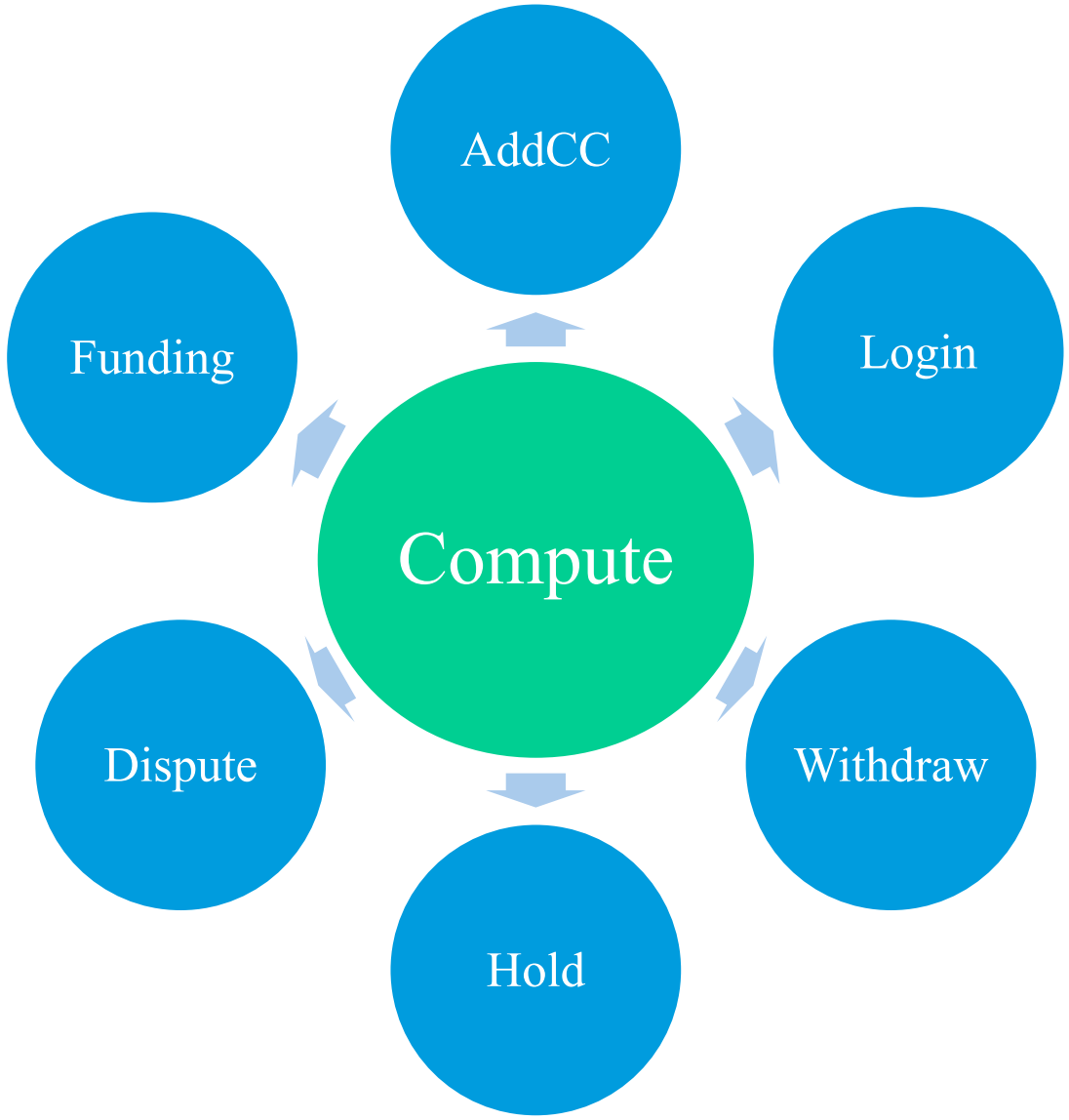
International Payments Risk Challenges

- Very high(2-4%) incoming pressure
- Well organized crime
- Multi-national jurisdiction
- Sophisticated fraudsters
- 100% buyer protection
- ATO, stolen financials, collusion

Agenda



Decision And Compute

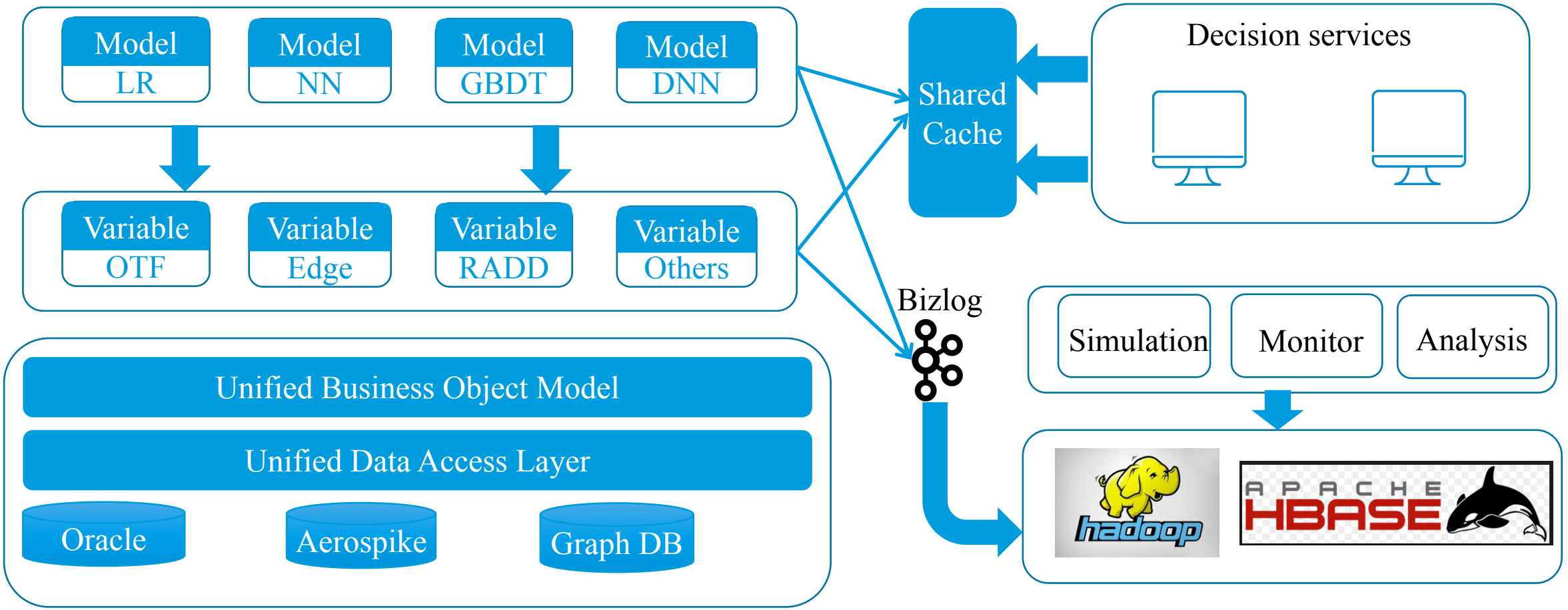


- 250MM risk decisions made per day
- Decisions are made in various CPs
- Decisions heavily rely on models
- Compute is core of decision making

PayPal AI Compute Platform

- Centralized and Unified AI compute platform
- One-stop self-service E2E solution
- Seamless offline-online integration
- Host 100+ Risk, GOPS, Marketing models
- Host 12000+ variable/features running on live
- 100MM calls per day
- Support multiple model algorithms

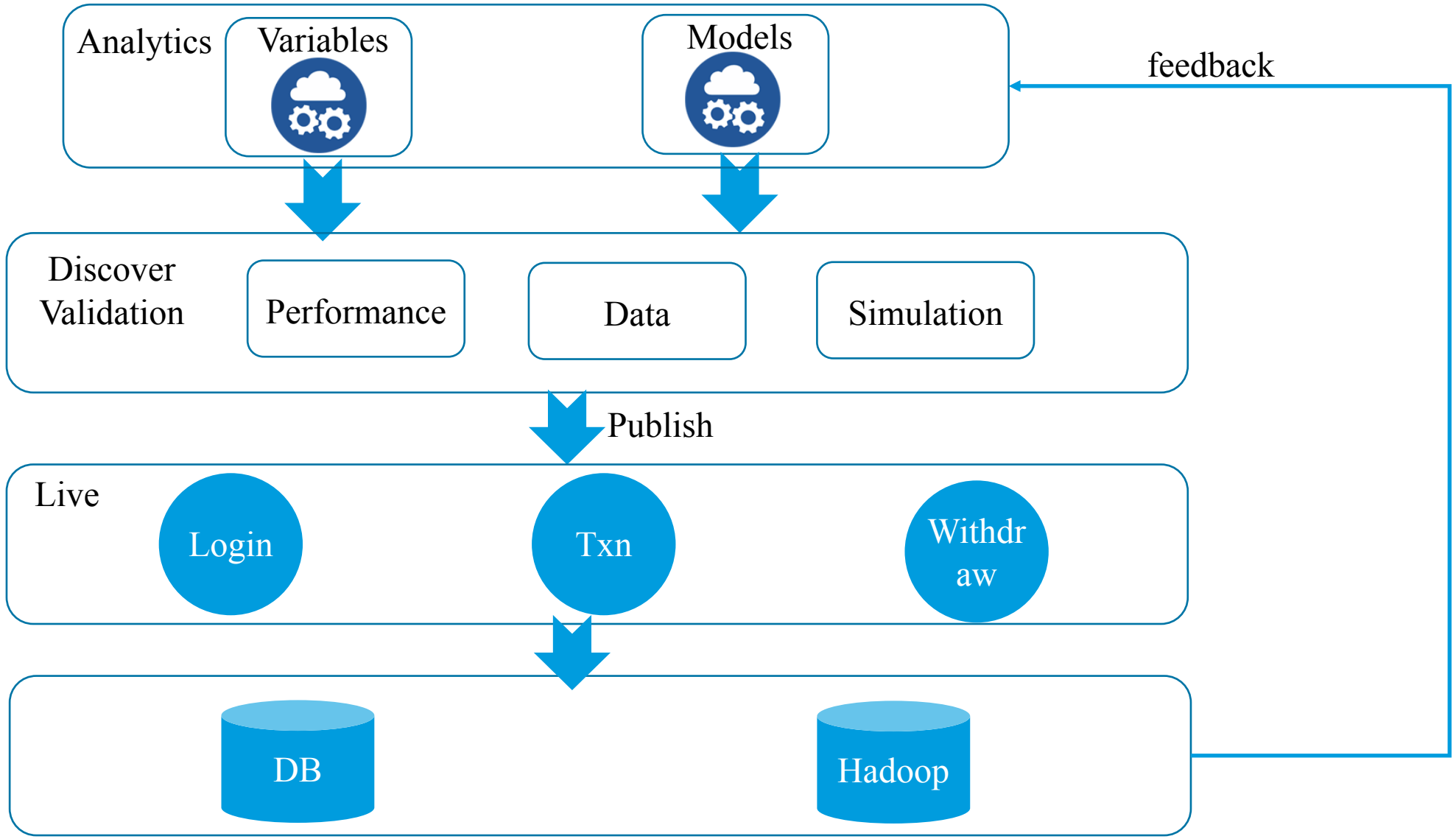
Compute Platform – Reference Architecture



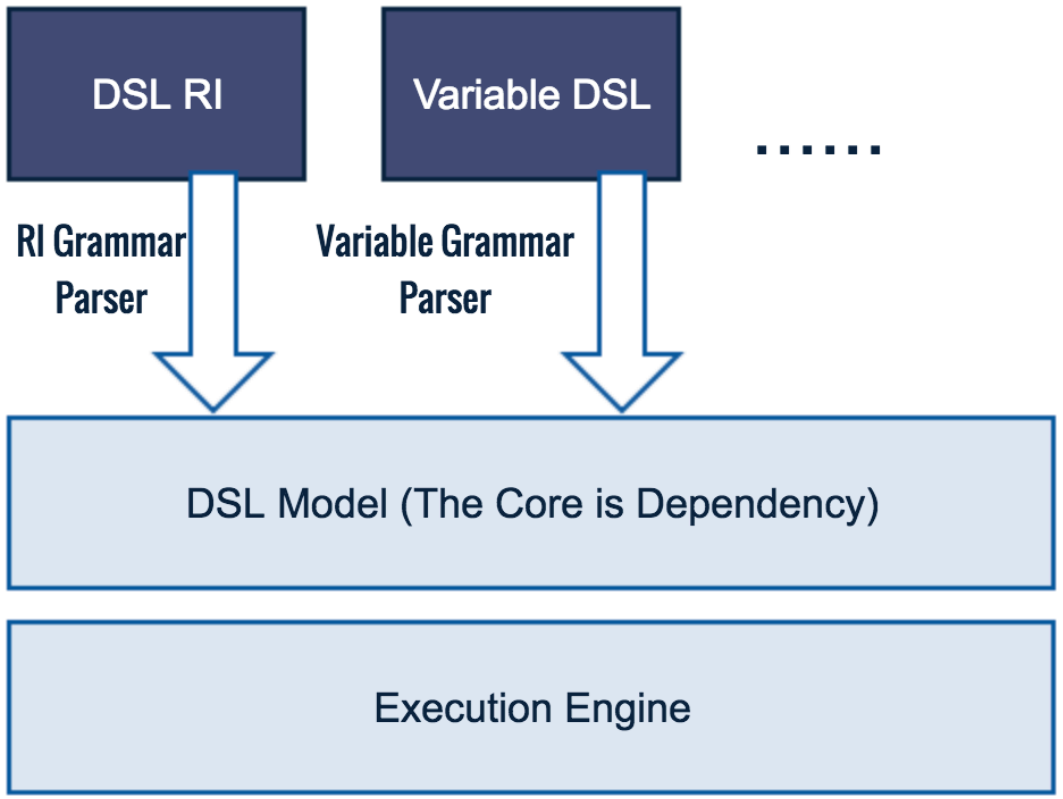
Agenda

- Platform Introduction And Architecture
- **User Experience And Core Capabilities**
- Scalability And Performance Tuning
- Deep Learning Support And Challenge

Over All Experience -- Write Once, Run Everywhere



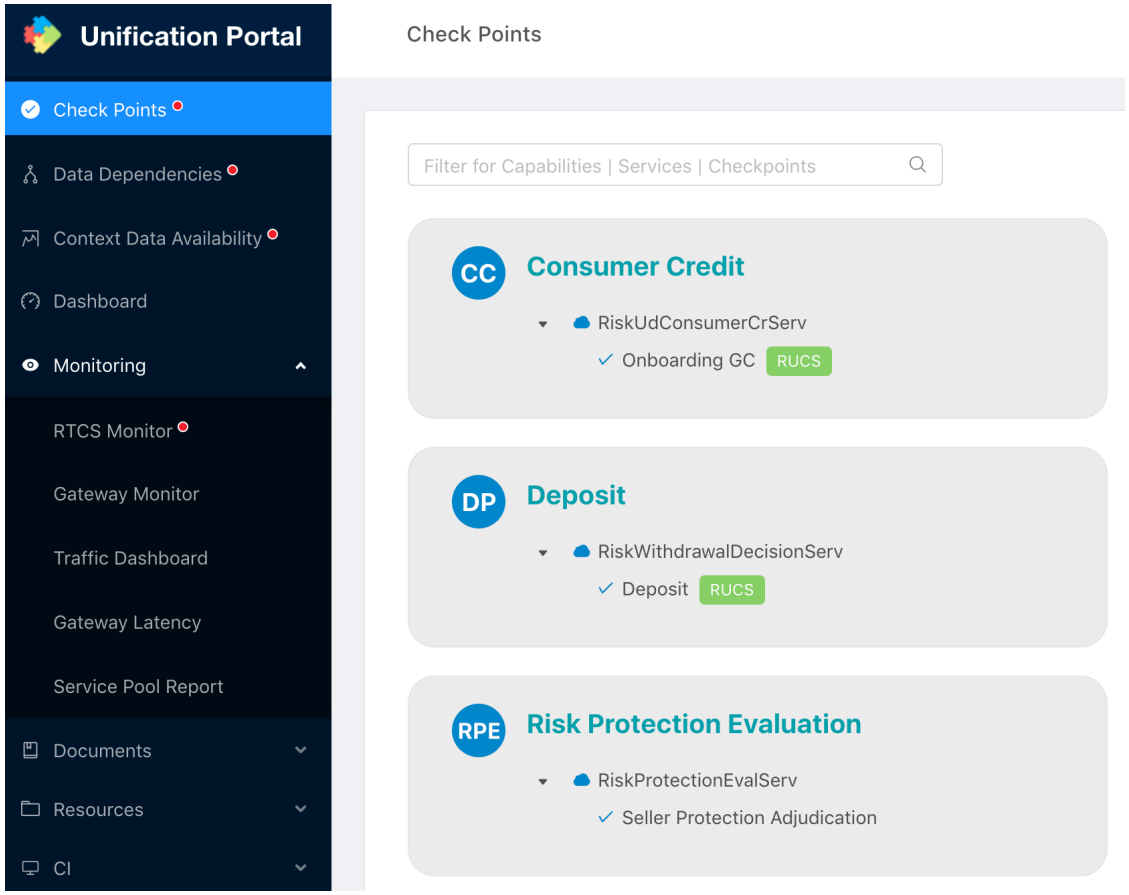
Feature Development -- DSL Based Variable Development



Key Capabilities

- User friendly DSL grammar analysts develop variables with SQL like grammar
- Define variables offline define and test variables on offline big data platform
- Advanced simulation support verify and predict variable results easily on offline
- Offline-online parity 99% match rate between offline simulation and online execution
- Hot publish to online easy and safe live deployment

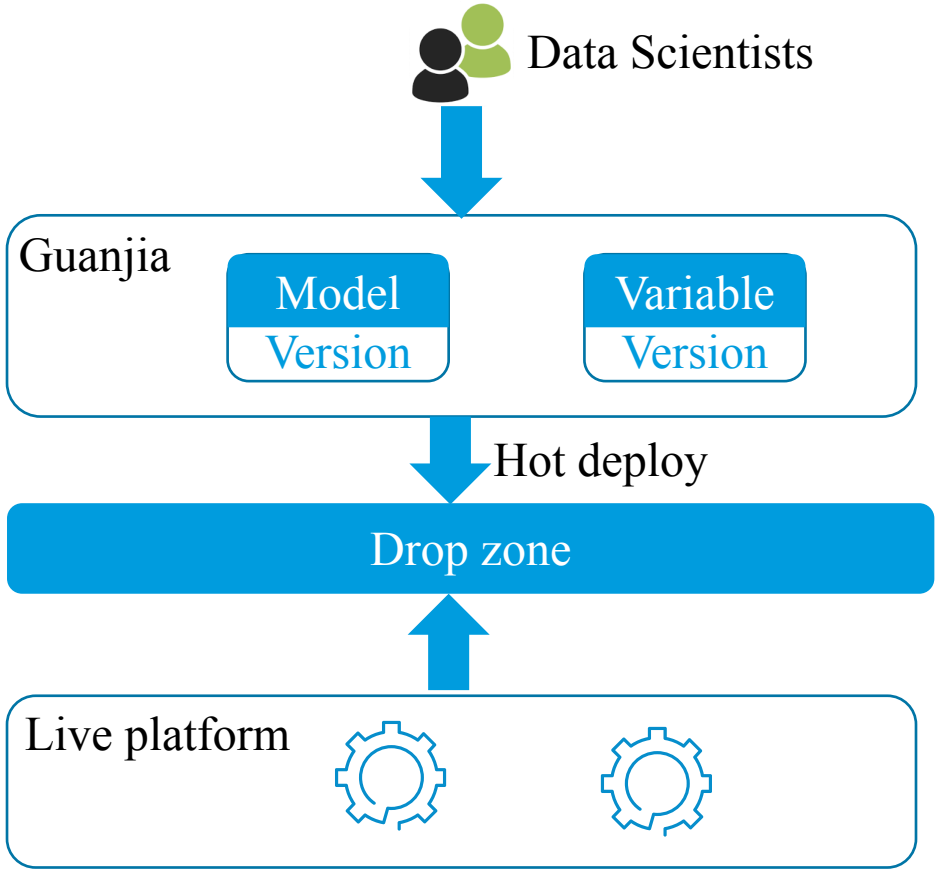
Validation And Integration -- Management Portal



Key Capabilities

- Fast data discovery discover data availability with given model and check point
- Unified data dependency identify data dependencies for any models and variables
- Performance prediction predict performance and capacity with given models and check points
- Information dashboard check points and integration status sharing
- Inter-CP data sharing sharing model and variable results between check points

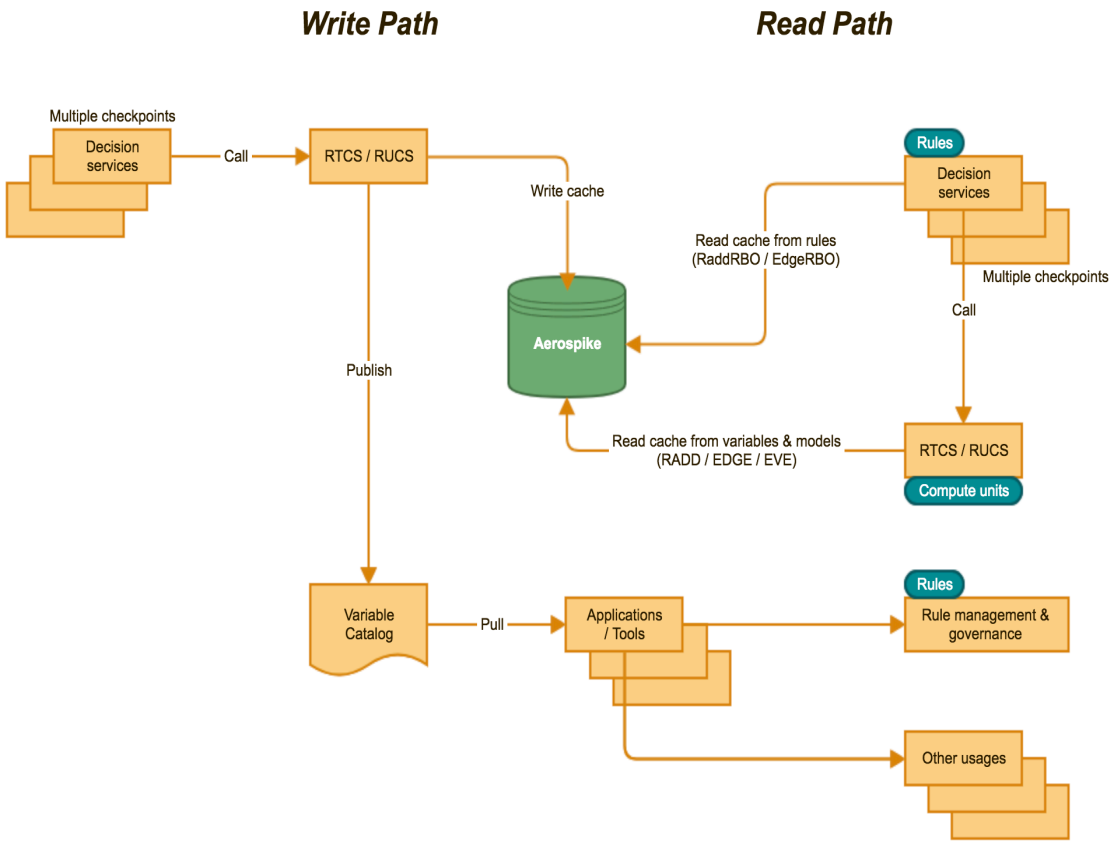
Publish -- Hot Model And Variable Deployment



Key Capabilities:

- One stop offline to online publish channel publish model from offline and pickup from online
- On-demand model DAG refresh refresh model DAG within 10 minutes
- Fast rollout rollout entire pool within 30 mins
- Easy test and verification on-demand rollout to specific boxes for test
- Version control and history management track all model history and fast rollback

Enhancement -- Inter Checkpoints Sharing



Key Capabilities:

- Configuration based data sharing
- Real-time results sharing between CPs
- Enrich capability in less data CPs
- Profile based risk control

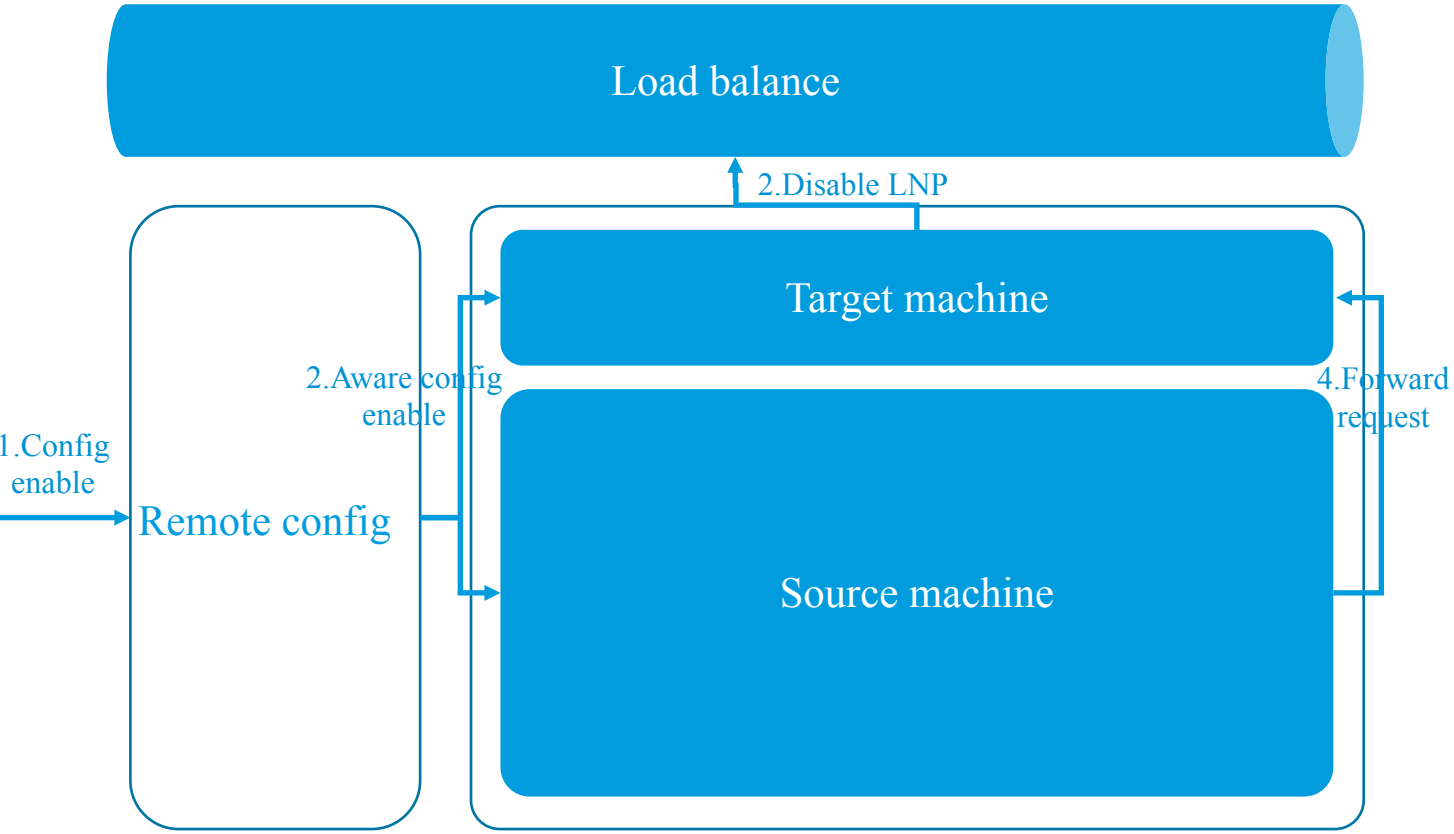
Agenda

- Platform Introduction And Architecture
- User Experience And Core Capabilities
- **Scalability And Performance Tuning**
- Deep Learning Support And Challenge

Scalability And Performance --Challenges

- Single point of failure
- Intensive variable and model computation
- Limited SLA with heavy runtime data loading
- High memory and CPU pressure
- Discrepancy from storages
- Case by case troubleshooting and analysis
- Monitoring and alert

Measurement -- Live LnP Test Framework

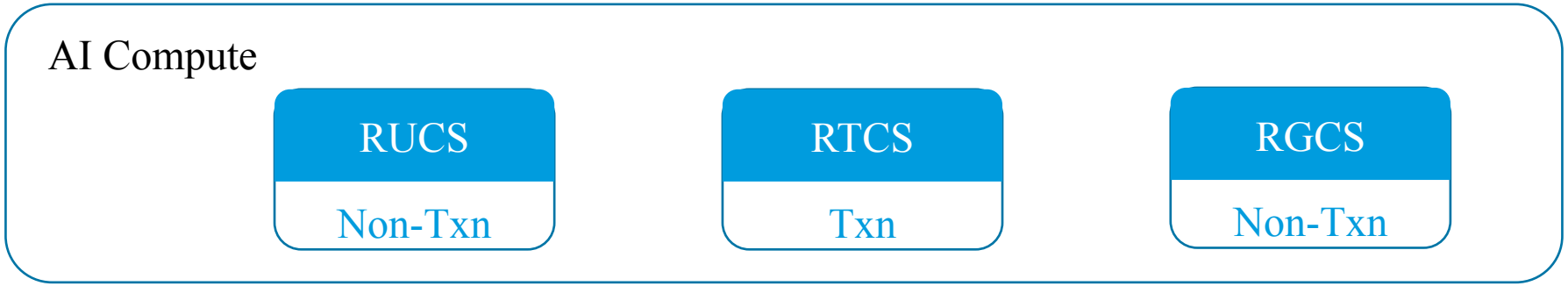


Key capabilities

- Fast testing with live traffic manipulate live traffic to test performance
- Configuration based traffic control deploy testing in 30 minutes
- Live traffic distribution simulation simulate different live traffic distribution
- Safety control avoid polluting live data

Safety Control -- Self Protection And Troubleshooting

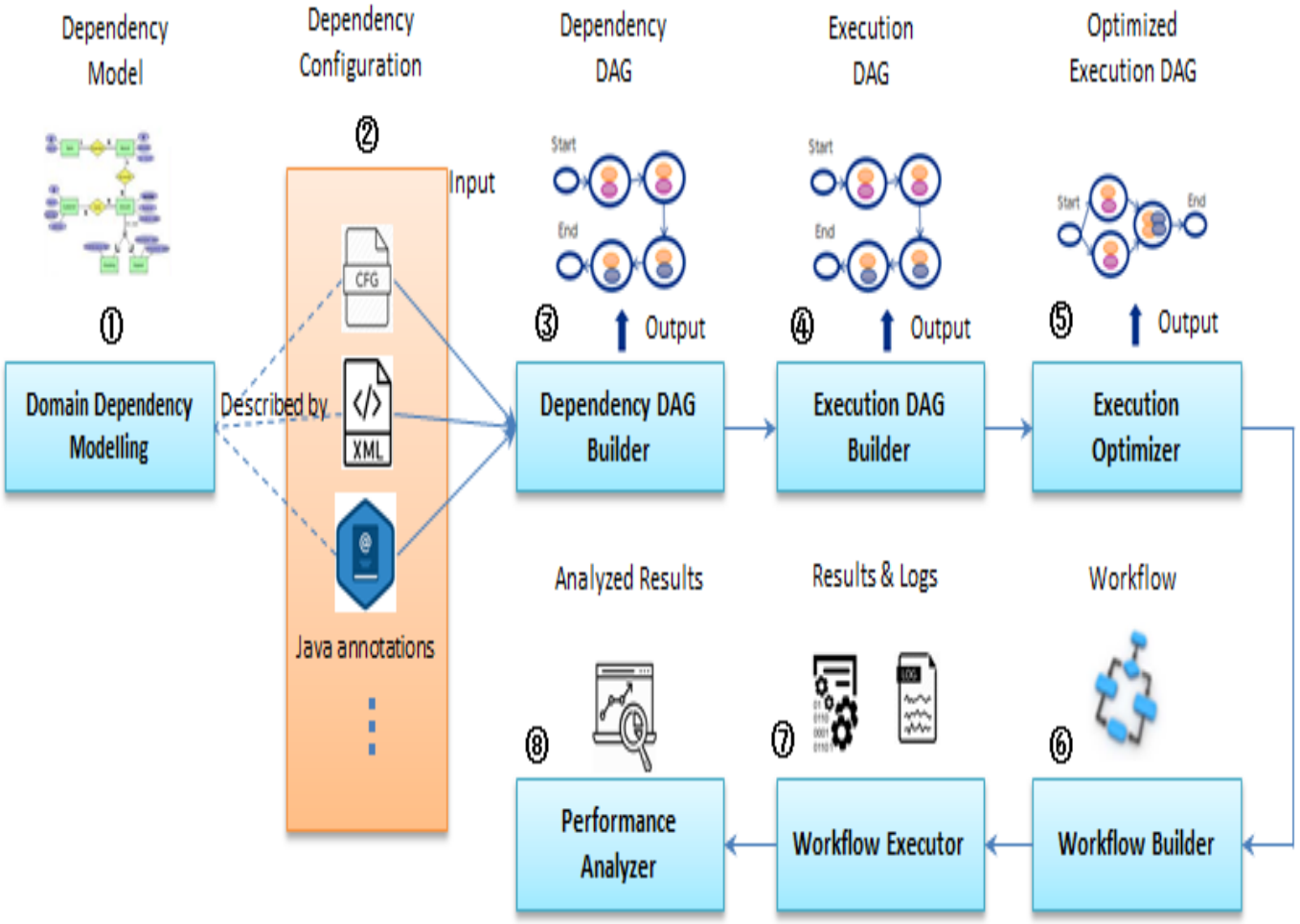
- Pooling strategy
- Reject request when service is unhealthy
- Disable slow nodes via remote configuration
- Node level timeout
- Critical path log for troubleshooting
- Last model score return during disaster



Optimization -- Memory Tuning

- Enable distributed cache for heavy data loading
- Optimize code to cache frequent accessed business object
- Reduce variable memory footprint
- Adjust GC policy
- Zing JVM practice

Optimization -- Engine Optimization

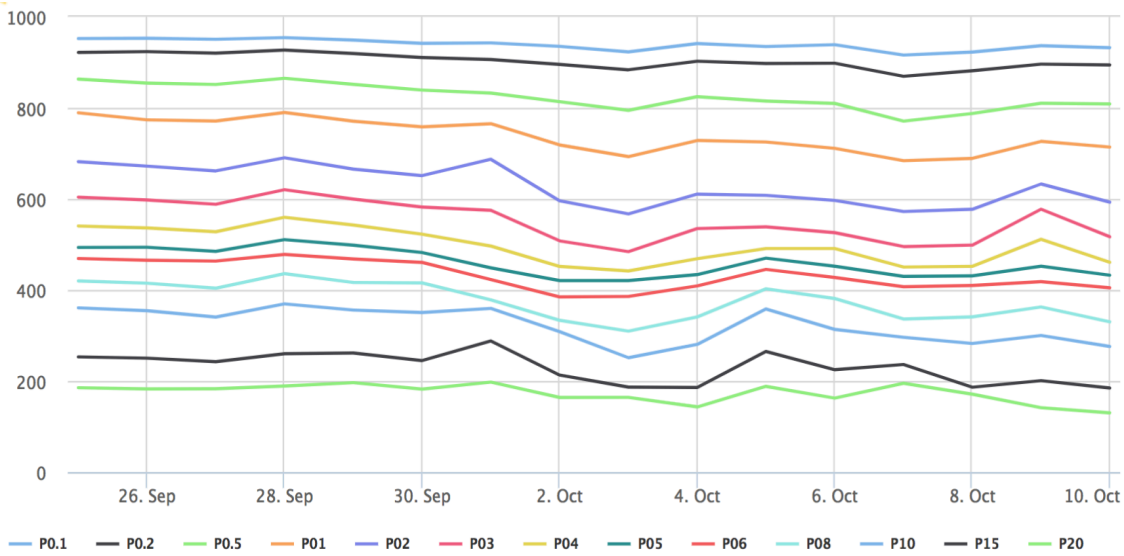


Key capabilities

- Static DAG optimization merge light nodes during DAG build
- Dynamic DAG optimization node group based on runtime metrics
- Remote configuration based strategy 30 minutes rollout without restart
- Metrics analysis automation tools to automate live metrics analysis

Monitoring And Alerts

- One-page online monitoring for all CPs and all key metrics
- Offline model and variable distribution monitoring
- Flow based decision monitoring
- Key metrics alerting to avoid noises



Agenda

- Platform Introduction And Architecture
- User Experience And Core Capabilities
- Scalability And Performance Tuning
- **Deep Learning Support And Challenge**

Deep Learning Inference Support In Compute Service

Java Inference Client

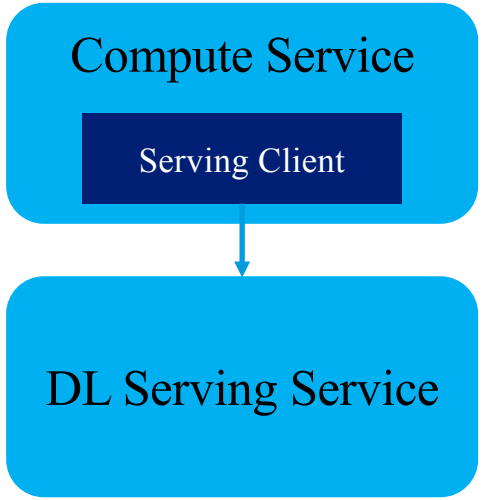
Pros:

DNN/CNN/RNN are All Supported Natively

Cons:

CPU Bound, Not Isolated from Compute Service

TensorFlow Serving



Pros:

TF Serving is Supported by Google

Cons:

Need Extra Resources

gRPC is http 2.0 based

Only TF model spec is supported

Thank You!