



PayPal Risk Infra Tech Optimization Practices

Li, Bruce, AERIS Architect

QCON Shanghai, Oct 2018

CNUTCon [上海]
全球运维技术大会

主办方 **Geekbang** **InfoQ**
极客邦科技

50+ 年末充电[⚡]

开发&运维技术干货大盘点

容器

Kubernetes

DevOps

全链路压测

Severless

自动化运维

Service Mesh

Elasticsearch

微服务

使用折扣码 「QCon」 优惠报名 咨询电话: 13269078023

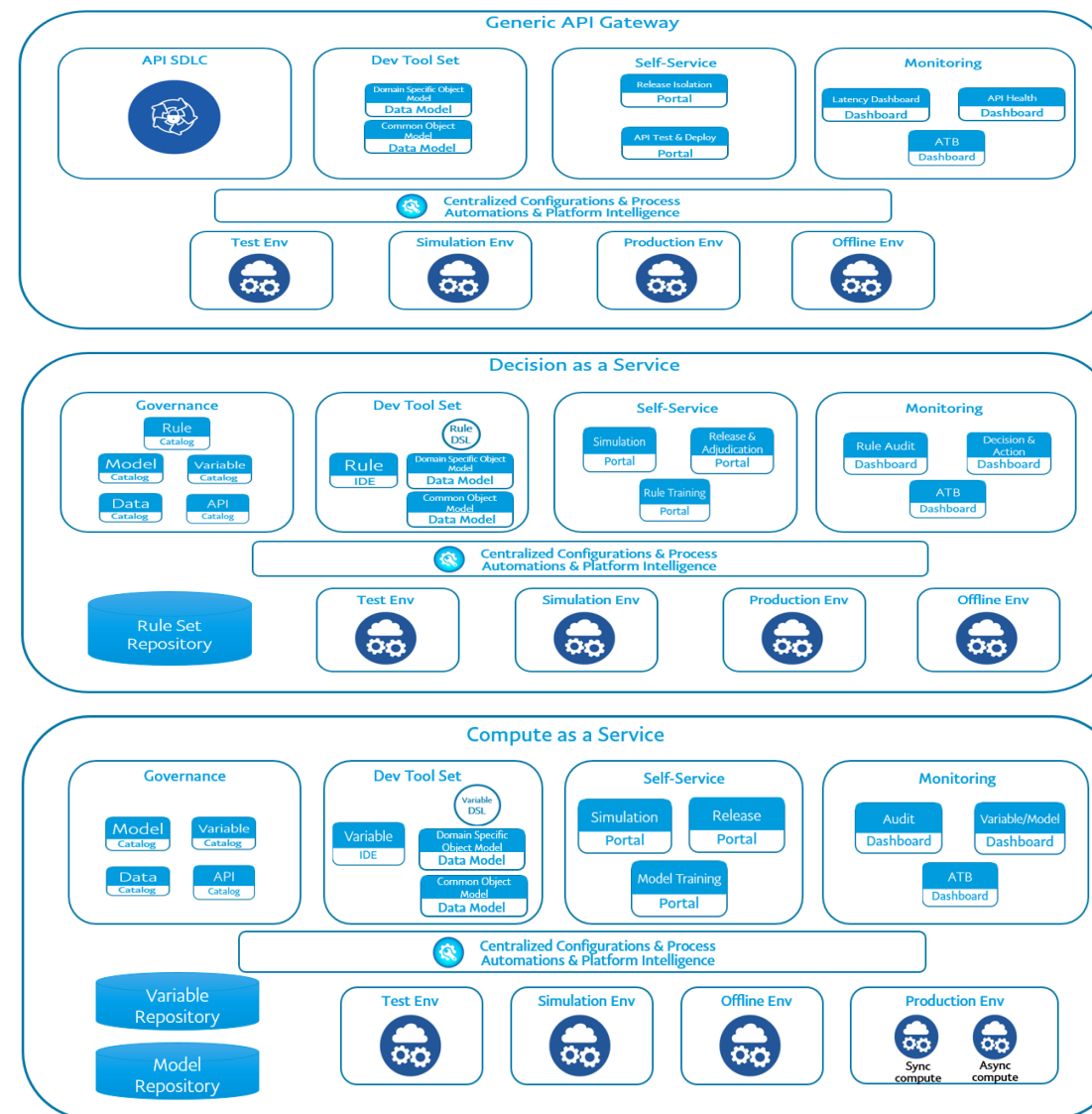
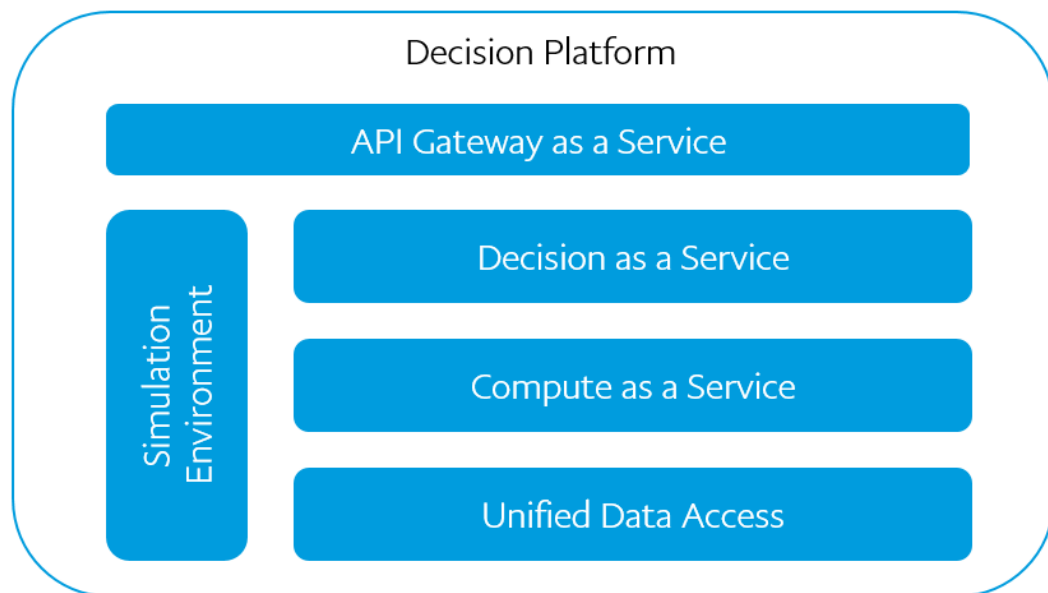


扫码锁定席位

Agenda

1. **Overall Introduction**
2. Asynchronous Workflow
3. Variable DSL
4. Q&A

Online Decision Platform Overview



Unified Dependency Model & Dependency-Driven Execution

Given the variables/models/rules:

- What are the API requirements?
- How to execute the rules/models/variables in a highly efficient way?
- What is the latency expectation because of the data loadings & computations?
- ...

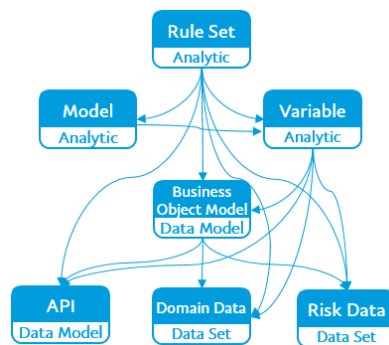
API Gateway Service

Decision Service

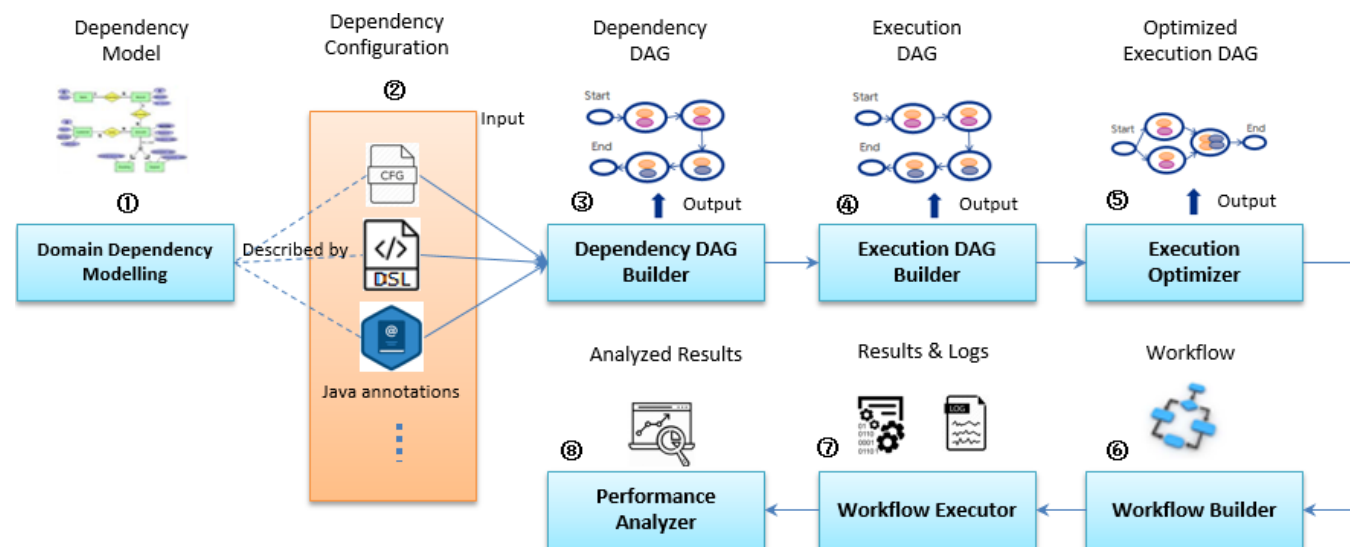
Compute Service

Unified Data Access
(data assets)

Unified Dependency Model

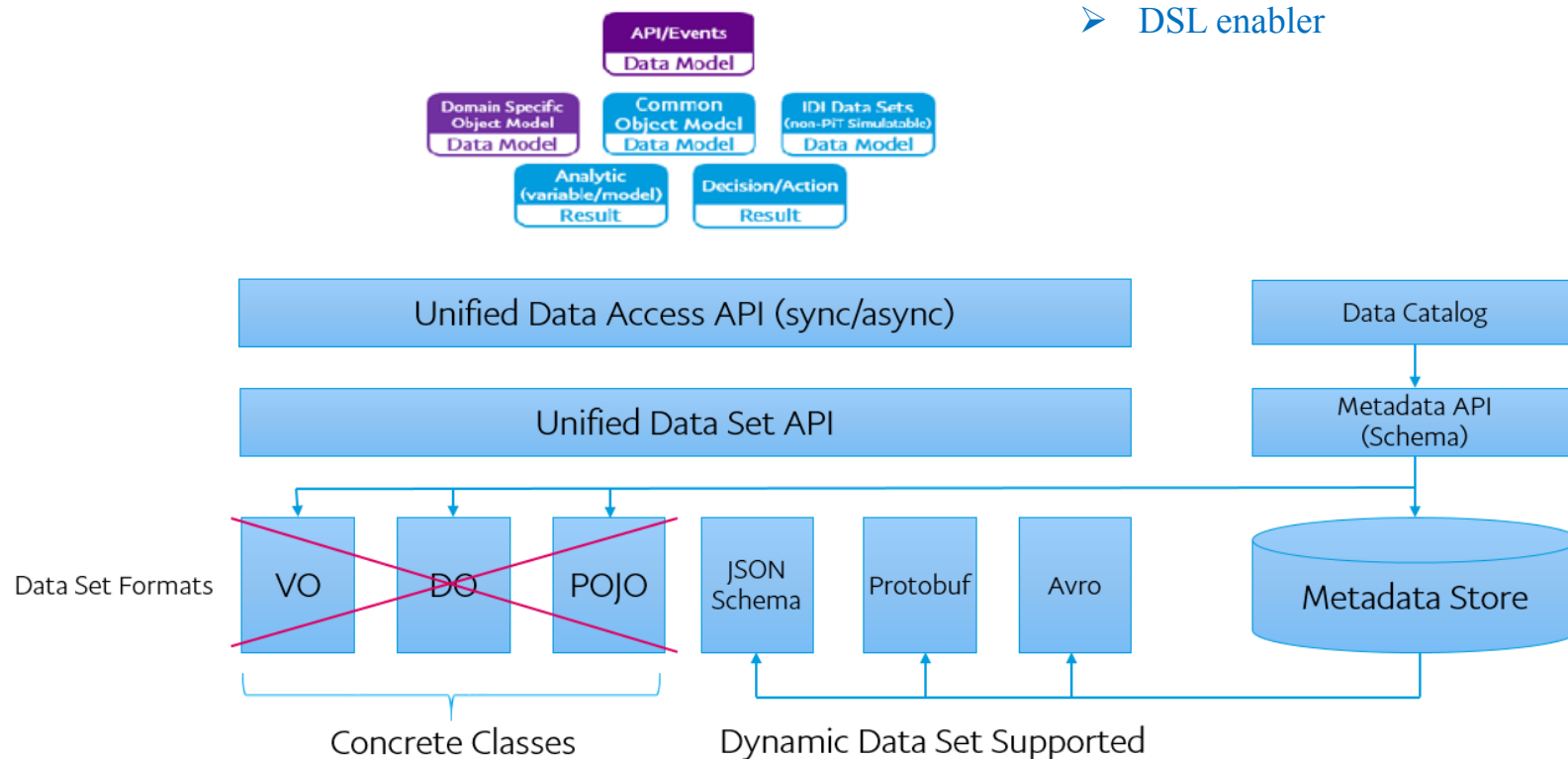


Execution

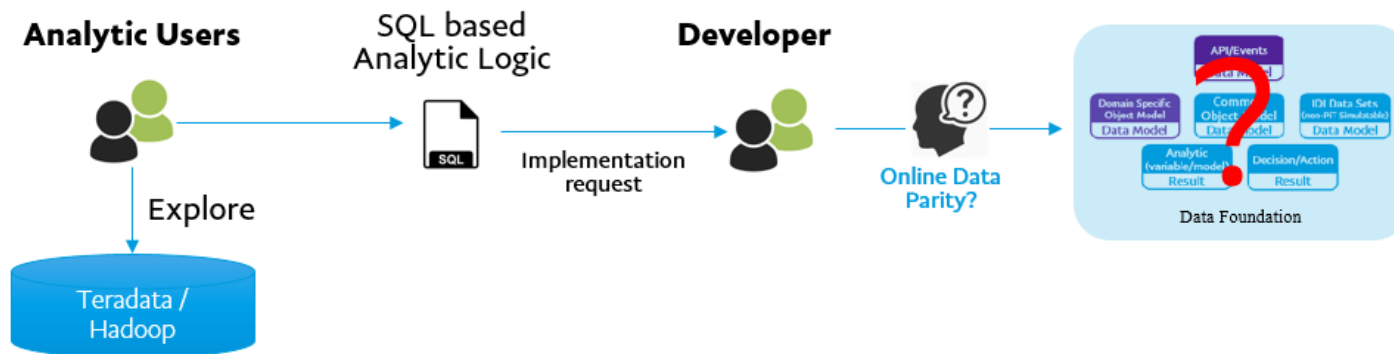


Data Set Abstraction & Unified Data Access

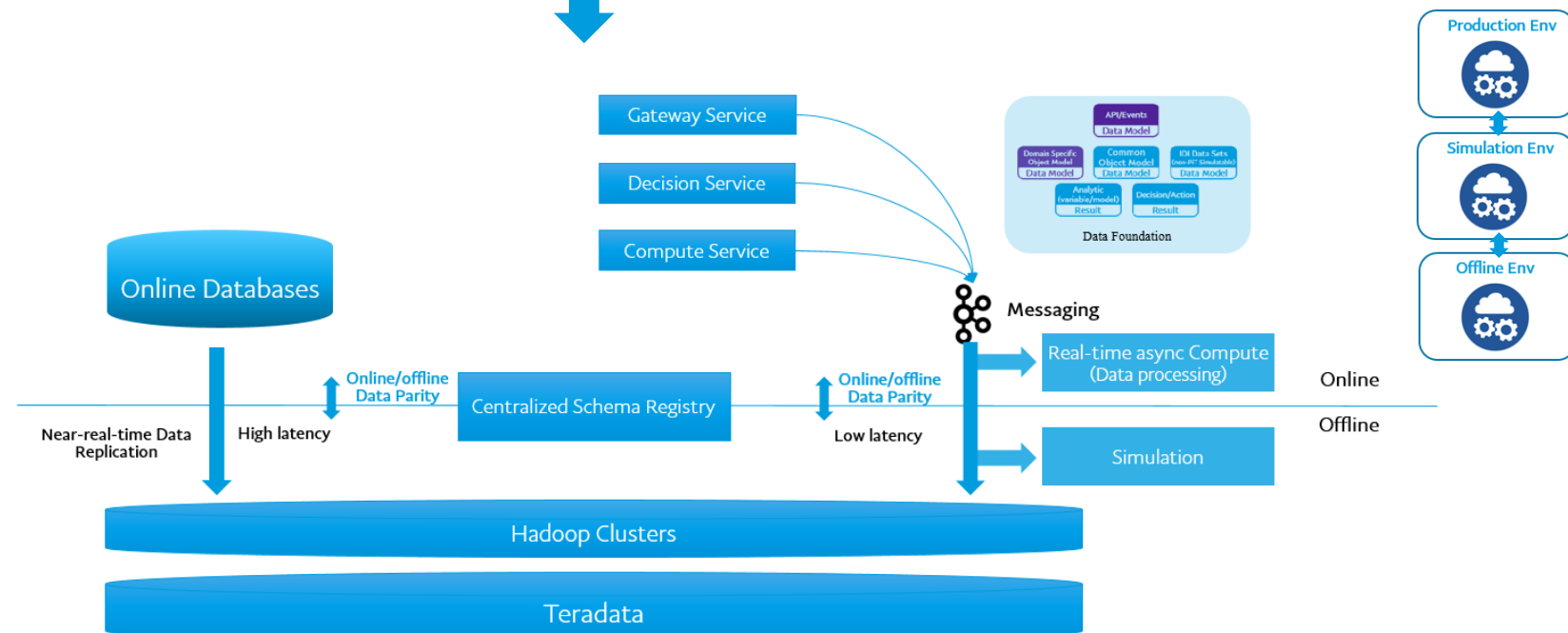
- Centralized configuration/metadata-driven data set management
- Enable centralized data governance
- Enable data dynamic release (no impact to services)
- Foundation of portable compute/decision engine
- DSL enabler



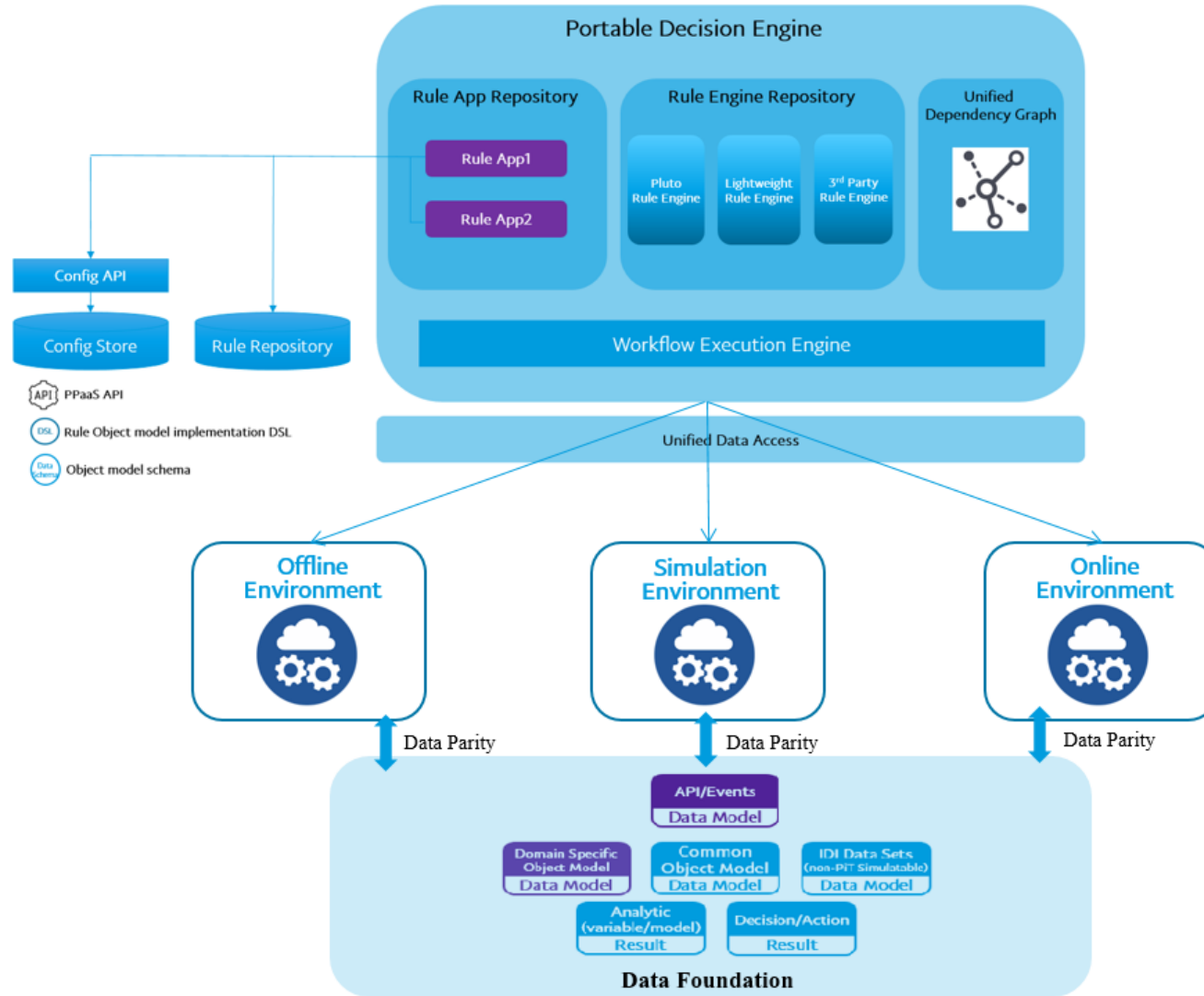
Online-offline Data Parity



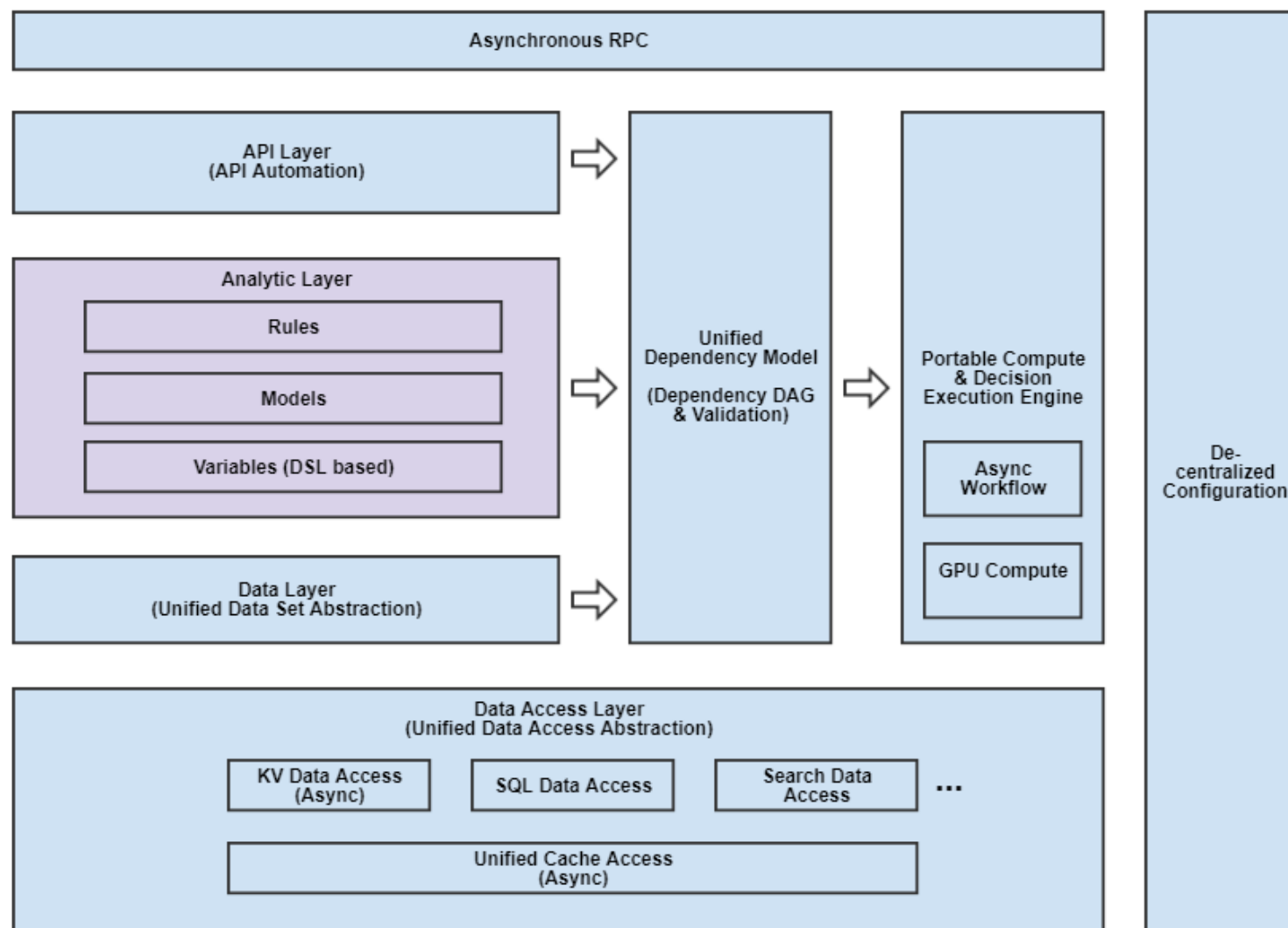
Change User Behavior



Portable Execution Engine



Unified Decision/Computation Execution Stack



Agenda

1. Overall Introduction
- 2. Asynchronous Workflow**
3. Variable DSL
4. Q&A

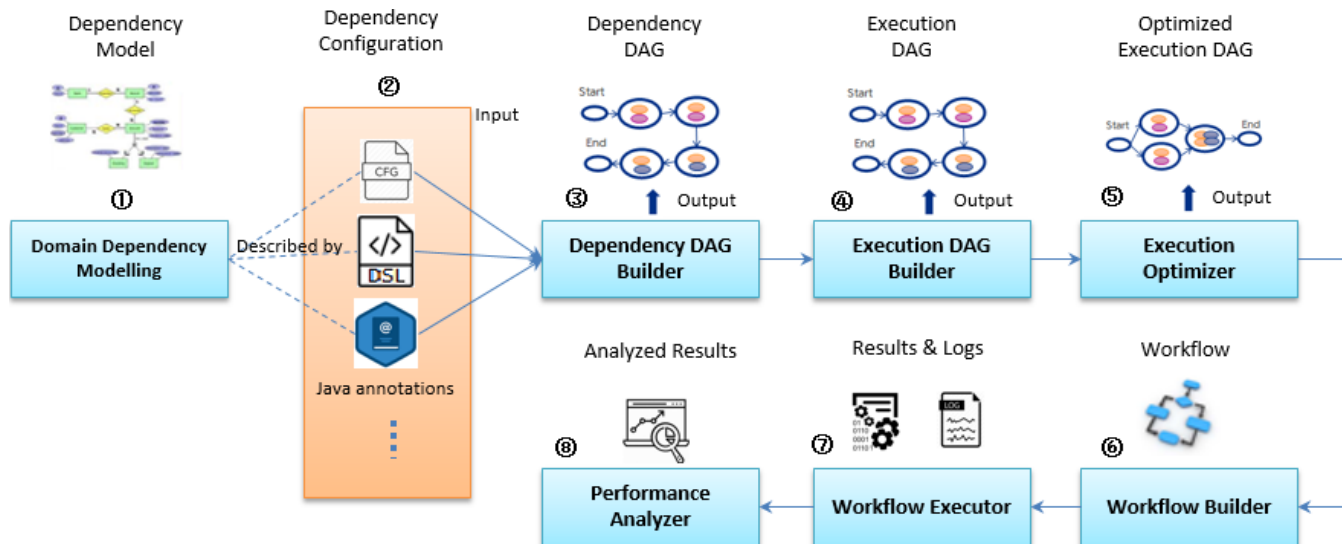
Asynchronous Workflow

Why?

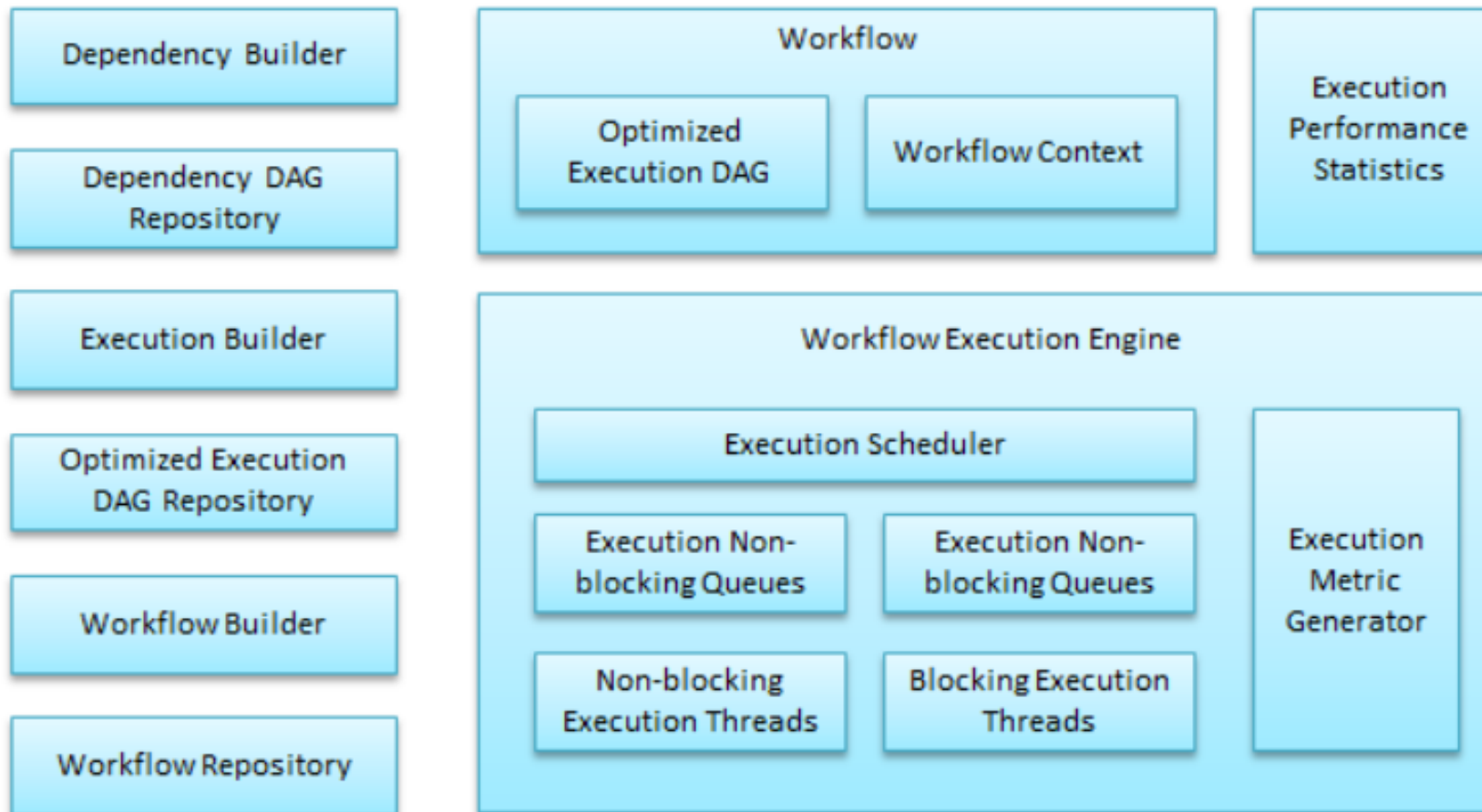
- Too many trivial nodes and the scheduling costs are high
- Highly concurrent blocking IO operations require large number of parallel threads, make scheduling costs even higher

Solution:

1. Make trivial node scheduling/executions highly efficient
2. Generate more trivial nodes by making blocking operations non-blocking, which will benefit from #1

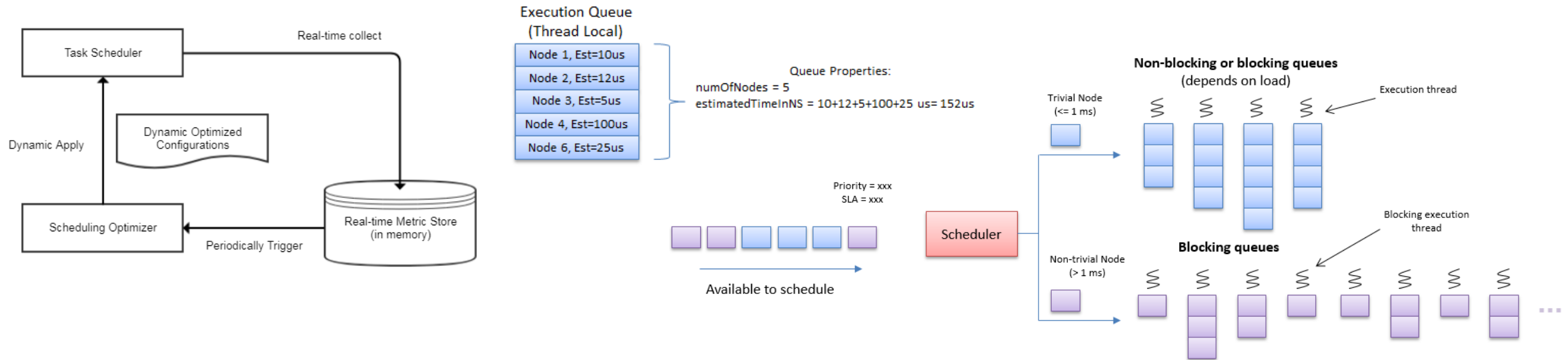


Asynchronous Workflow High Level Design



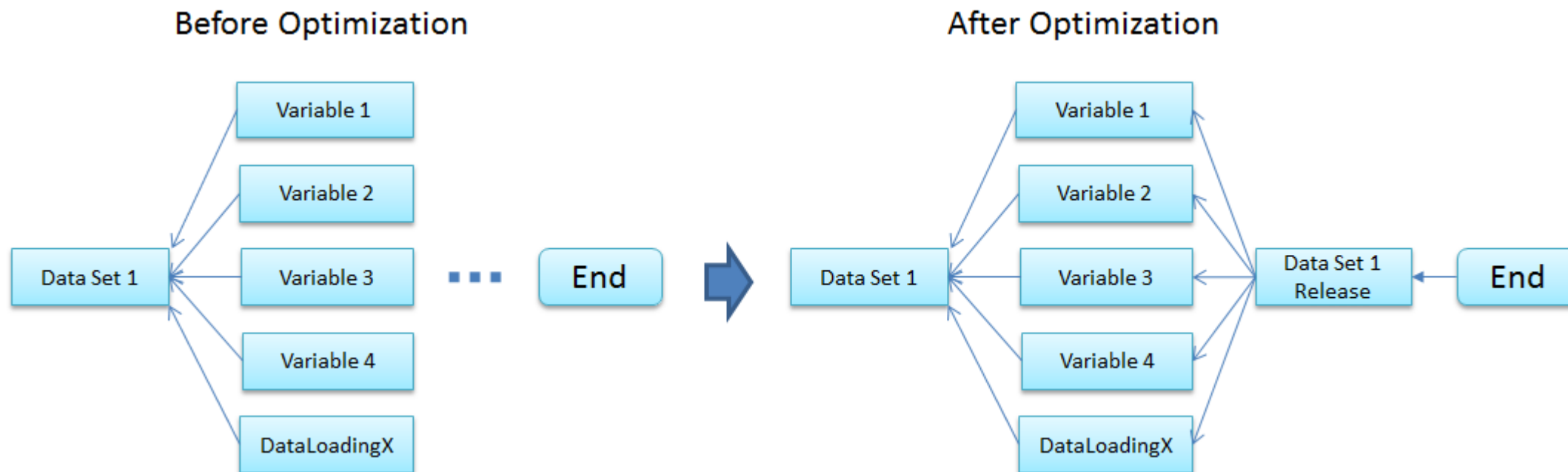
Tiered Node Scheduling

- Keep collecting node execution metrics
- Bind workflow thread to its queue (can be shared)
- Separate scheduling for trivial nodes and non-trivial nodes
- Smartly select scheduling algorithm for each tier:
 - Fair scheduler
 - Deadline scheduler
 - Prioritized scheduler



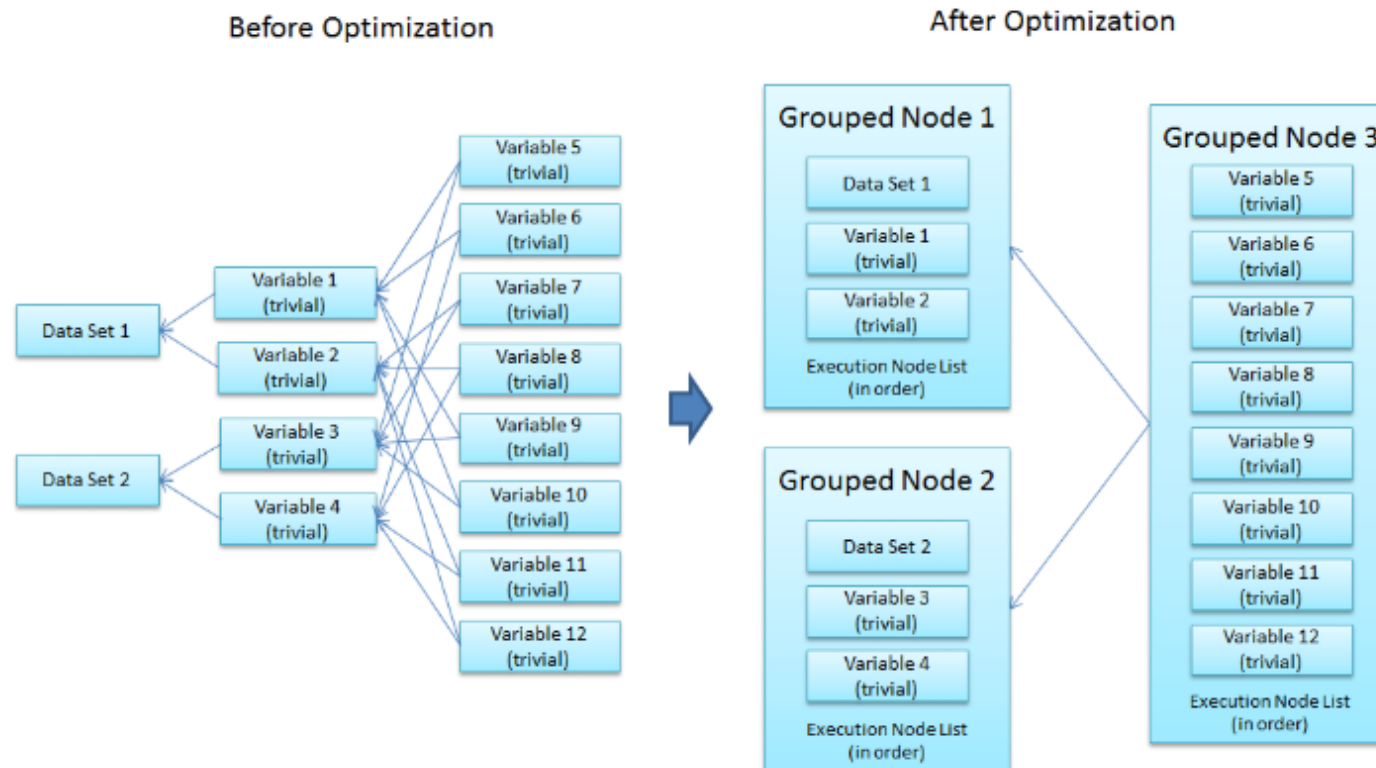
Execution Optimizer - Auto Release Optimizer

- Release objects as early as possible based on dependency understanding
- Improve GC performance



Execution Optimizer – Trivial Node Auto Group

- Reduce the number of trivial nodes
- Improve node scheduling cost

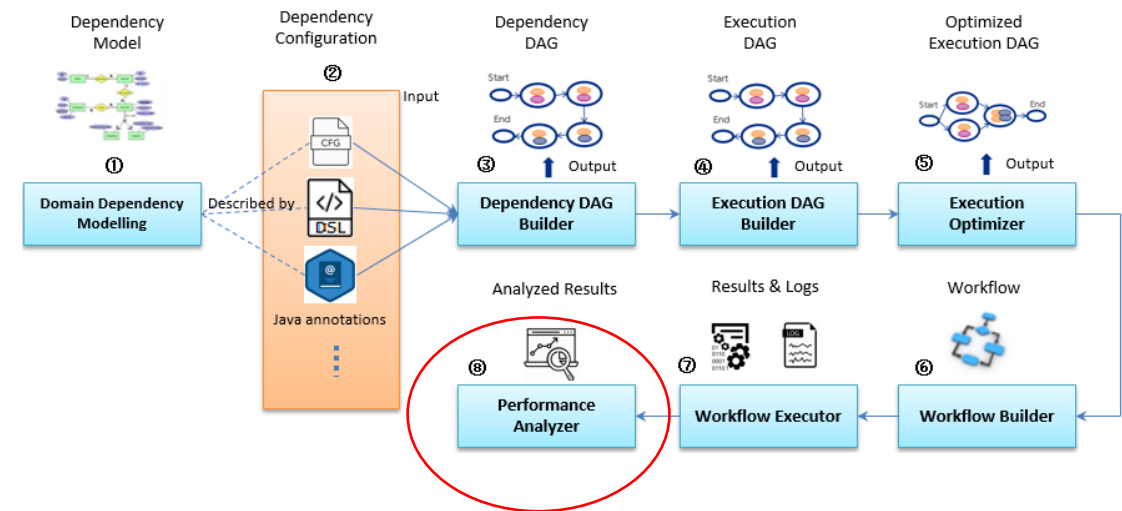
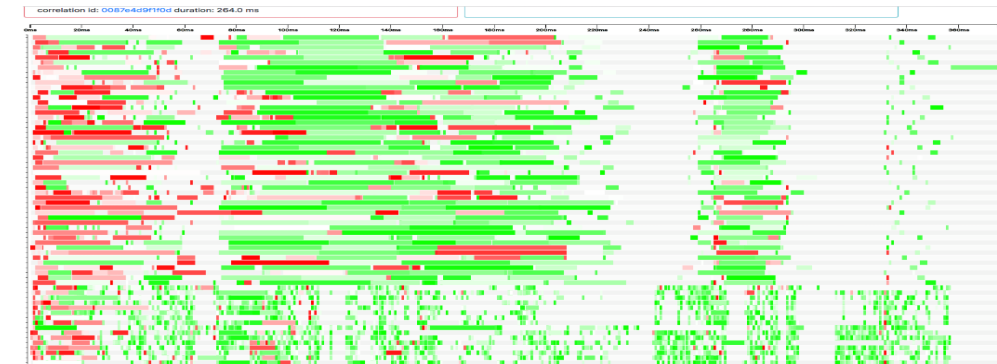


Workflow Performance Analyzer

Before optimization



After optimization



Critical Path Analytics

Async Workflow Critical Path		Filter:
Metric	Value	
CRITICAL_PATH_1(1)	c2935073b78b4 38117.356:0.356 start(0,0) startNode(0,0) preComputeHookNode.Perf_Simulator(0,0) preComputeStartNode(0,0) FINAL(3,54) end(0,0)	
CRITICAL_PATH_2(1)	409983d1c8ede 712.669:27.1.26 start(0,0) startNode(0,0) preComputeHookNode.Perf_Simulator(0,0) preComputeStartNode(0,0)) end(0,0)	
CRITICAL_PATH_3(1)	88f45dc8d7526 299.199:82.0.82 start(0,0) startNode(0,0) preComputeHookNode.Perf_Simulator(0,0) preComputeStartNode(0,0)) end(0,0)	
CRITICAL_PATH_4(1)	1c71531e24478 470.370:72.4.68 start(0,4) startNode(0,0) preComputeHookNode.white_list(0,0) preComputeStartNode(0,0)) end(0,0)	
CRITICAL_PATH_5(1)	c4b8c2eb528b8 264.177:77.0.77 start(0,0) startNode(0,0) preComputeHookNode.Perf_Simulator(0,0) preComputeStartNode(0,0)) end(0,0)	

Agenda

1. Overall Introduction
2. Asynchronous Workflow
3. **Variable DSL**
4. Q&A

Variable DSL

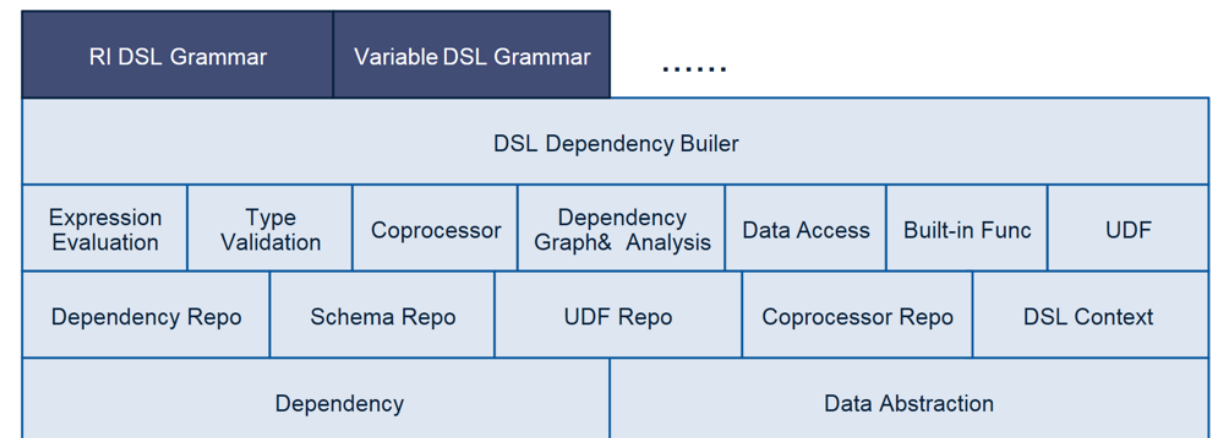
Why?

- Enable configuration-based analytic release
- Unified variable definition for both online and offline
- Dependency inline, no additional configuration needed
- Achieve more efficient executions than code based variables (exactly-once execution for any level of DSL expression with perfect hash optimizations)

About Expression:

- An expression must return a value
- Support Math Calc, Bool Calc and Compare
- Support property access, list access.
- Built-in func: map, reduce, if-else, load data
- Support UDF

...



Dependency is the Key

Core APIs of dependencies

```
Set<Dependency> getThisDirectlyDependsOn();  
Set<Dependency> getDependsOnThisDirectly();  
String getDependencyId();  
DependencyType getDependencyType();  
Object calcValue(DSLContext context);
```

Two layers of dependencies

Global Dependency

Input contexts

Variables

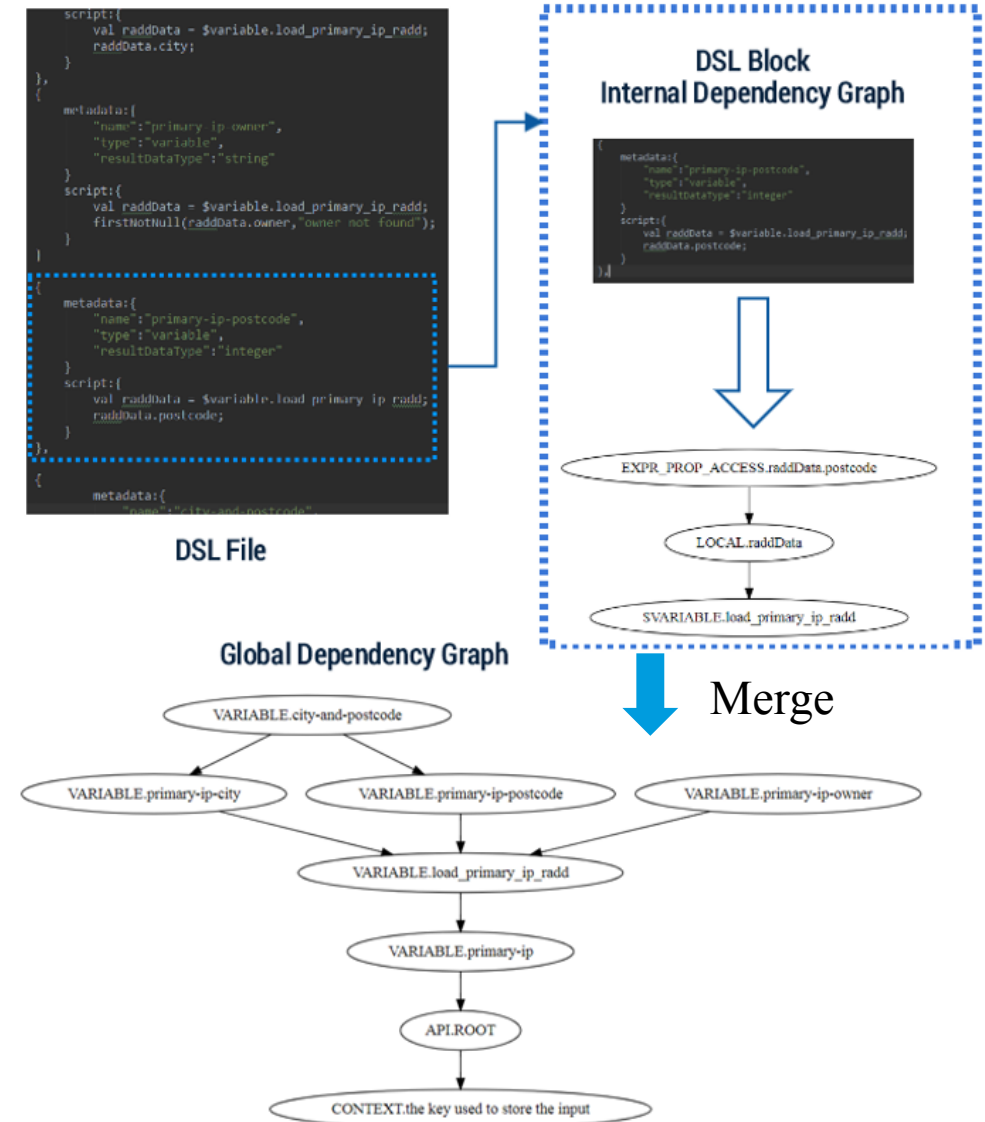
Variable Internal Dependency

Literal Value, local value, global dependency

Math Calc, Bool calc, compare

If-else, map, reduce, create, UDF...

Data set loading



DSL Coprocessor

- DSL evaluation callback
- Non-intrusive inspection
- Make DSL easy to debug

Coprocessor API:

```
DependencyEvalCoprocessor
  getName(): String
  getPriority(): int → PriorityEntity
  onException(DSLContext, T, Throwable, String): Object
  onHitCache(DSLContext, T, Object): Object
  postEvaluation(DSLContext, T, Object): Object
  preEvaluation(DSLContext, T): Object
```

```
{
  metadata:{
    "name":"primary-ip-city",
    "type":"variable",
    "resultDataType":"string",
    "coprocessors":[
      {"name":"logExpr"},
      {"name":"logInputParams"}
    ]
  }
  script:{
    val raddData = $variable.load_primary_ip_radd;
    raddData.city;
  }
},
```

```
DSLLoggerEvalCoprocessor.java
37
38 @Override
39 public Object postEvaluation(DSLContext context, DSLB
40 System.out.println("Value:" + value); value: "Shan
41 return null;
42 }
43 If needed, add breakpoint for local debugging
44 }
```

Variables

- this = (DSLLoggerEvalCoprocessor@1106)
- context = (DSLContext@1102)
- expression = (VariableDependency@1103)
- expressions = (ArrayList@1108) size = 2
- blockName = "primary-ip-city"

```
DSL Block Dependency ID:VARIABLE.primary-ip-city
=====start=====
This depends on:
VARIABLE.load_primary_ip_radd:com.paypal.risk.dataset.BaseDataSet@4f933fd1
This depends on no data loading
=====end=====
Value:Shanghai
```

logExpr will print block Name onPreEval
and block return value onPostEval

logInputParams will print all dependency value
and data loading result if any

极客时间VIP年卡

每天6元, 365天畅看全部技术实战课程

- 20余类硬技能, 培养多岗多能的混合型人才
- 全方位拆解业务实战案例, 快速提升开发效率
- 碎片化时间学习, 不占用大量工作、培训时间



技术创新的浪潮接踵而来， 继续搬砖还是奋起直追？

云数据

AI

区块链

架构优化

高效运维

CTO技术选型

微服务

新开源框架

会议：2018年12月07-08日 培训：2018年12月09-10日

地址：北京·国际会议中心



Q & A
Thank You!