NJSD
中国（南京）软件开发者大会
China (Nanjing) Software Developers Conference
2016

微软拥抱Spark的开源之路

曾凯
微软云计算与信息服务实验室(CISL)

Spark 的历史

- 高效通用的大数据处理框架
- 由UC Berkeley, AMPLab的Matei Zaharia创始
- 2010年按BSD许可协议开源
- 2014年成为Apache顶级项目
- 至今有863代码贡献者

Spark核心模型

- 数据抽象为一个*Resilient Distributed Dataset*s (RDDs)
  - 分布式的存储在集群中
  - 可在存储在内存/硬盘
- 用户应用被抽象为做一系列RDDs的转换操作
  - RDDs操作： map, filter, join, reduce, groupBy, …
  - 不存储数据，只存储RDD转换操作历史（lineage）
  - 故障时根据记录的历史自动重新计算生成RDDs

Spark工作原理

- 将日志中的报错信息存入内存里，然后快速的查找一些字符串

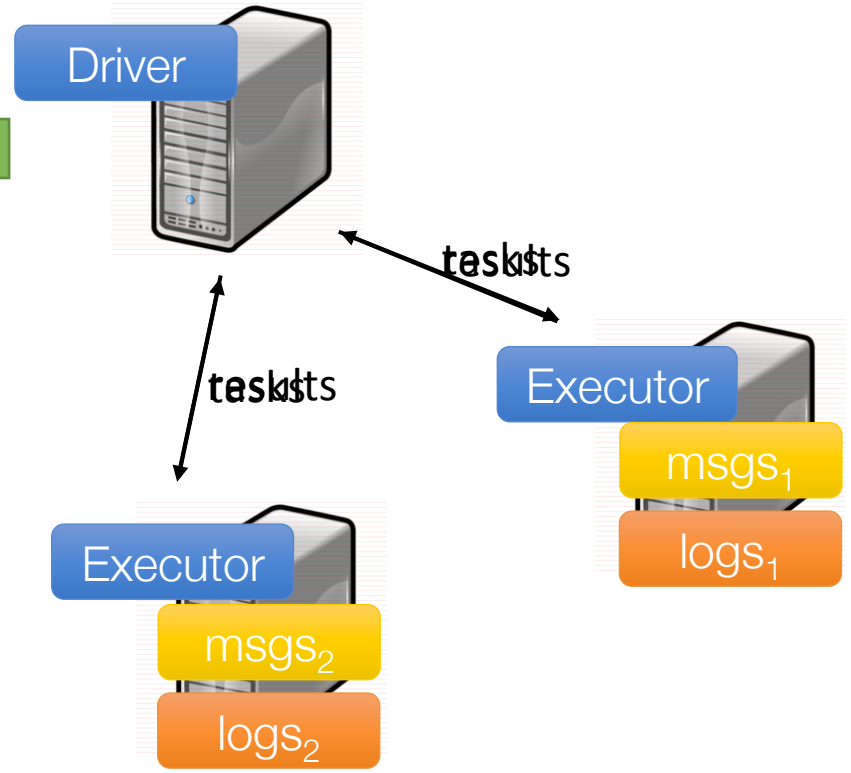
```
// header: LEVEL, MSG
// INFO, msg1
// ERROR, msg2
// …
// …
```

```
val logs = spark.textFile(“hdfs://…”)
val errMsgs = lines.map(_.split(“,”))
                   .filter(_(0) == “ERROR”)
                   .map(_(1))
errMsgs.cache()

errMsgs.filter(_ contains “foo”).count()
```

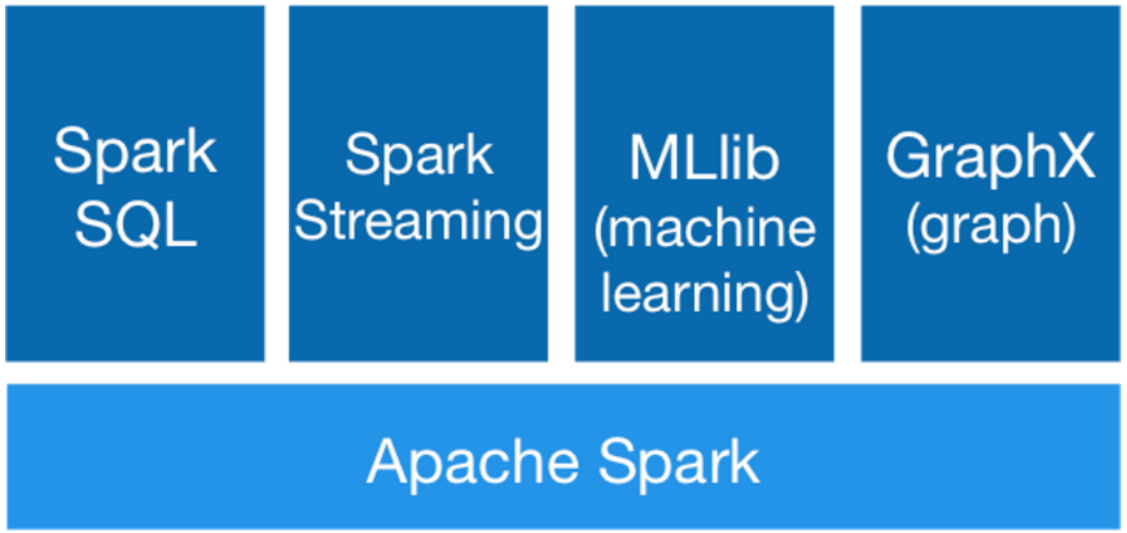
RDD

Transforms



Spark核心组件

- Spark SQL: 结构化数据处理
- Spark Streaming: 流数据处理
- MLlib: 机器学习
- GraphX: 图分析



Spark SQL做日志处理

- 用*DataFrame*s来抽象结构化数据

```
val errMsgs = sqlCtx.read.format(“csv”)
                .load(“hdfs://…”)
                .where(“LEVEL = ‘error’”)
                .select(“MSG”)

errMsgs.cache()


errMsgs.where(“MSG like ‘foo’”)
       .count()
```

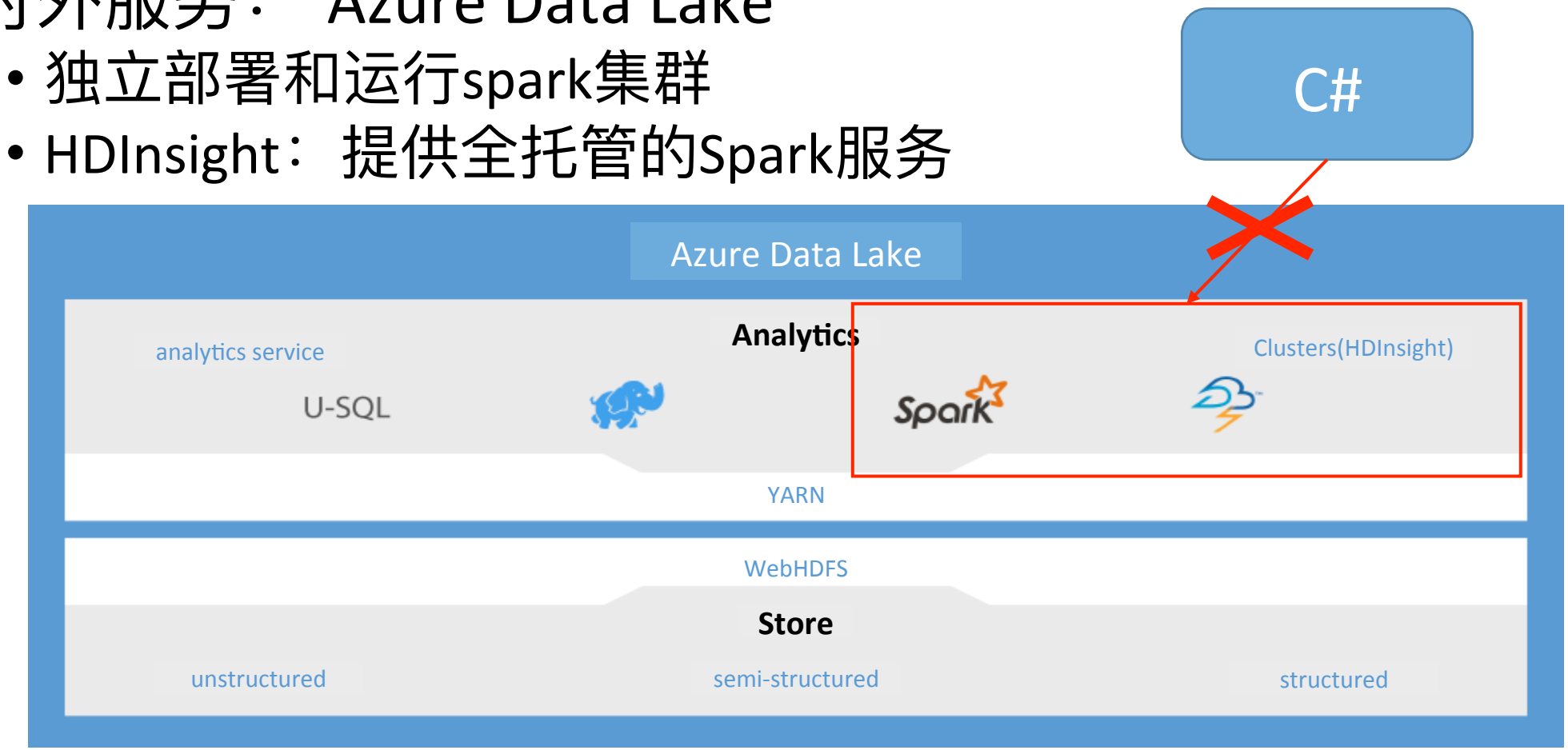
```
// header: LEVEL, MSG
// INFO, msg1
// ERROR, msg2
// …
// …
```

Spark高兼容性

- 支持Hadoop Yarn， Mesos， standalone等多种部署模式
- 兼容HDFS, Cassandra, Azure, S3等多种存储系统

Spark在微软

- 对外服务： Azure Data Lake
  - 独立部署和运行spark集群
  - HDInsight：提供全托管的Spark服务



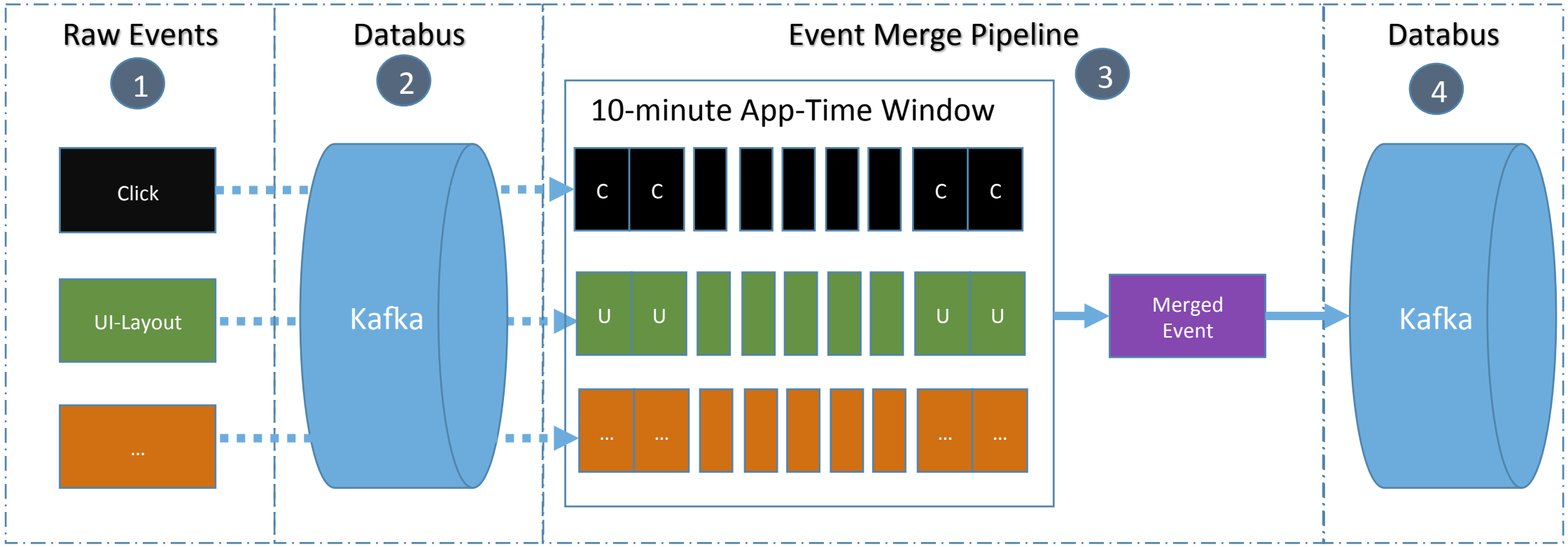
Spark在微软

- 对外应用： Azure Data Lake
  - 部署和运行spark集群
  - HDInsight：提供全托管的Spark服务
- 内部应用：使用spark用于大数据处理流程
  - 如何与已有的.NET逻辑整合?

Bing日志处理

- 近实时的日志处理 (“FastSML”)
  - 数量级：每小时数百TB
  - 用户的点击事件，网页布局信息等
- 下游的应用
  - 使用近实时点击信号改进搜索结果
  - 智能运营 (Operational Intelligence)
  - …

Bing日志处理--FastSML



FastSML+Spark?

Apache Storm (SCP.Net) + Kafka + Microsoft’s内部的基于内存的流数据分析引擎

→ 吞吐量瓶颈

- Spark能改进这个处理流程吗?
- 我们能重用FastSML已有的处理逻辑吗?
  - 如：重用已有的C#库

Spark + .NET

对于工程开发中广泛的使用了.NET的技术的公司

- 使用C#来开发Spark应用
- 整合已有的.NET业务逻辑
- 在Spark库中重用.NET库

Spark + .NET = Mobius!!!

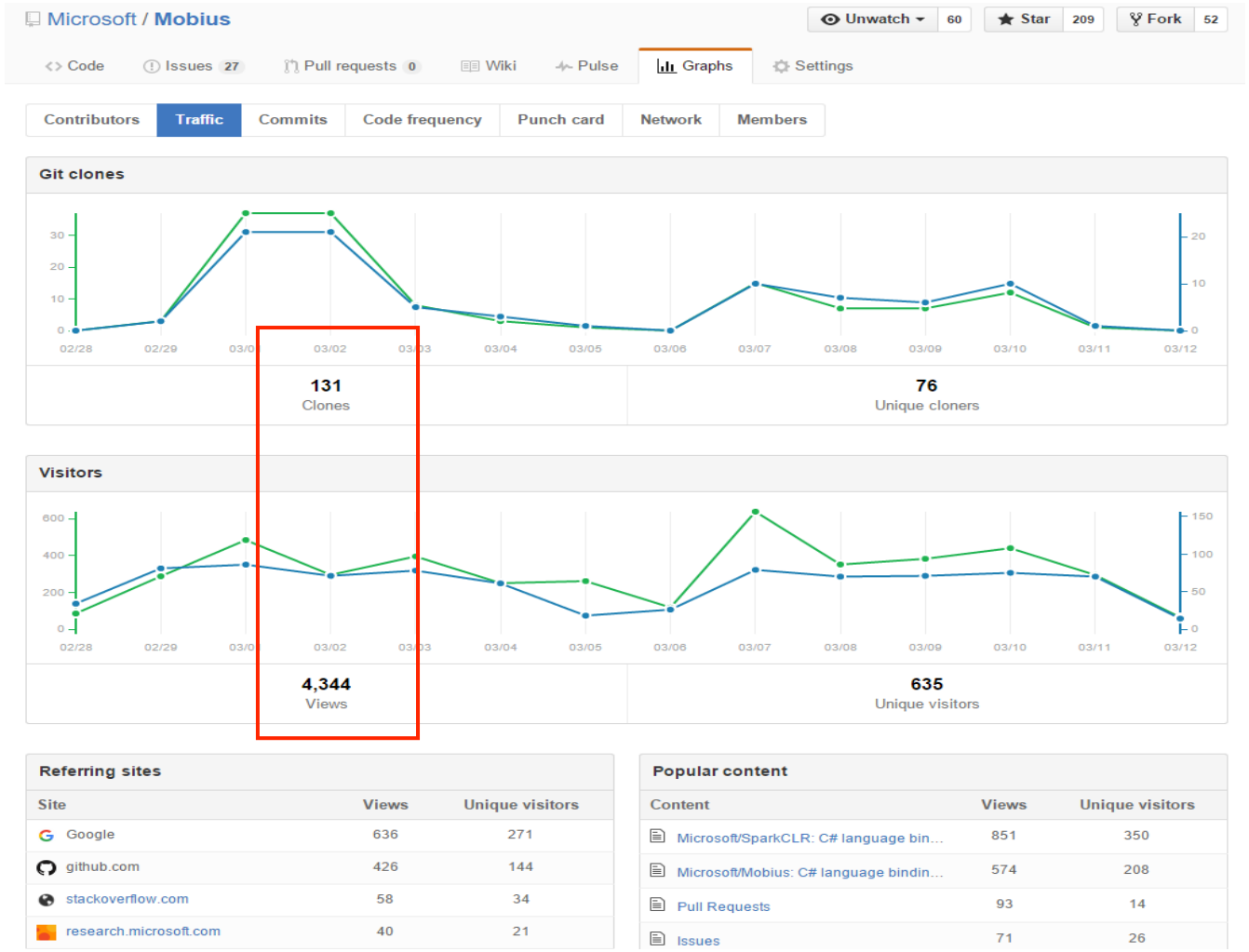
Mobius的历史

- 始于2015年8月
- 由CISL和ASG (Bing)共同开发（最初由5人发起）
- 2015年11月开源
- 使用MIT许可协议

- 已经发布版本 V1.5.2 和 V.1.6.0
- 4月即将发布 V1.6.1

Mobius@github

- Apache Spark Wiki 链接http://github.com/Microsoft/Mobius
- 758次代码提交, 2个发布版本
- 最近两周 – 131下载, 4K访问



Mobius的目标

使C#成为Spark原生支持的开发语言

- 批处理
- 流数据处理
- 结构化关系数据处理
- 交互式的数据处理
- 托管服务

Word Count

Scala

```
val textFile = spark.textFile("hdfs://… ")
val counts = textFile.flatMap(line => line.split(" "))
                     .map(word => (word, 1))
                     .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://…")
```

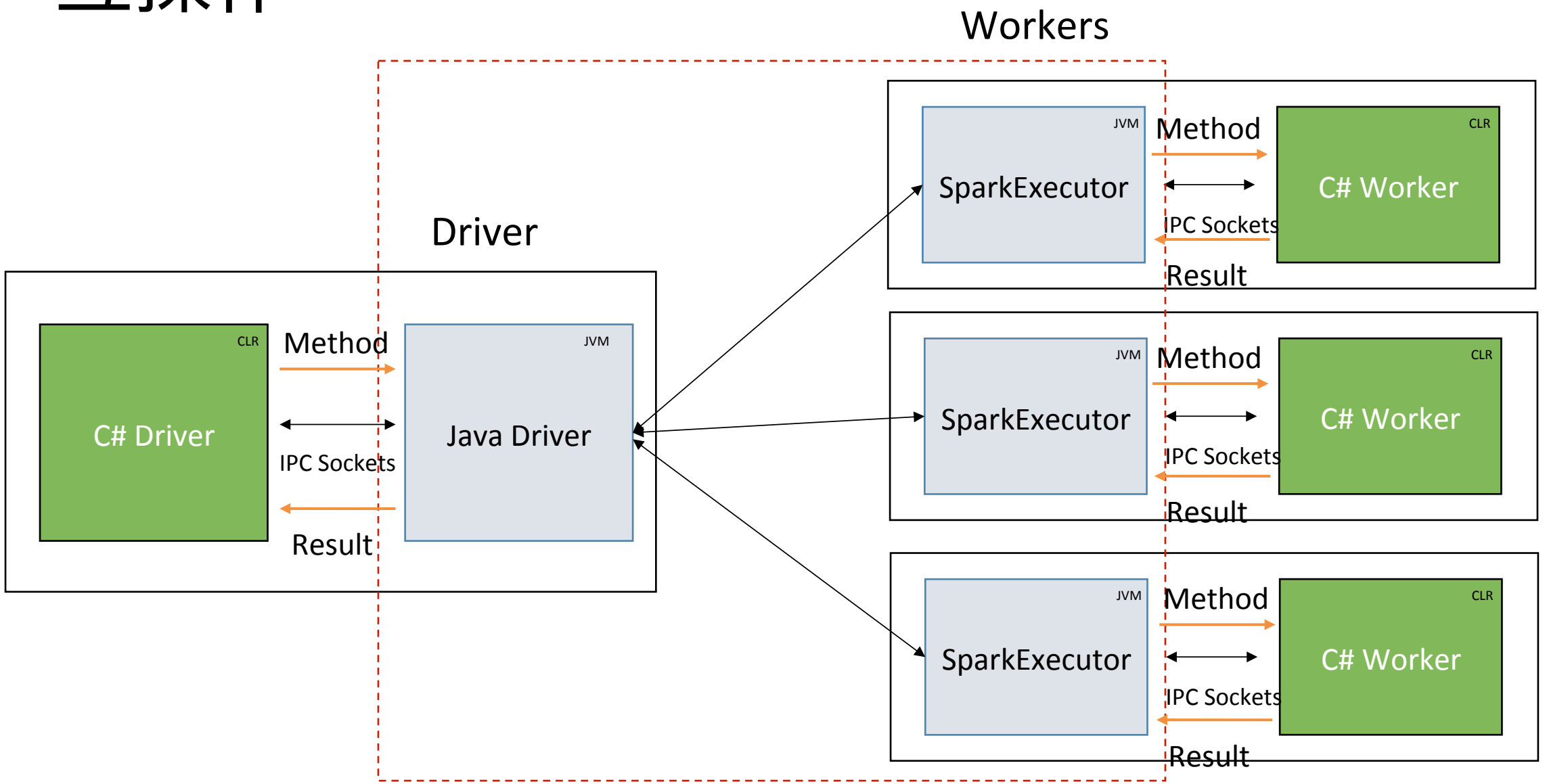
C#

```
var textFile = sparkContext.textFile(@"hdfs://… ")
var counts = textFile.FlatMap(line => line.split(" "))
                     .Map(word => new KeyValuePair<string, int>(word, 1))
                     .ReduceByKey((x,y) => x + y)
                     .Map( wc => string.Format("{0}, {1}", wc.Key, wc.Value));
counts.saveAsTextFile(@"hdfs://… ");
```

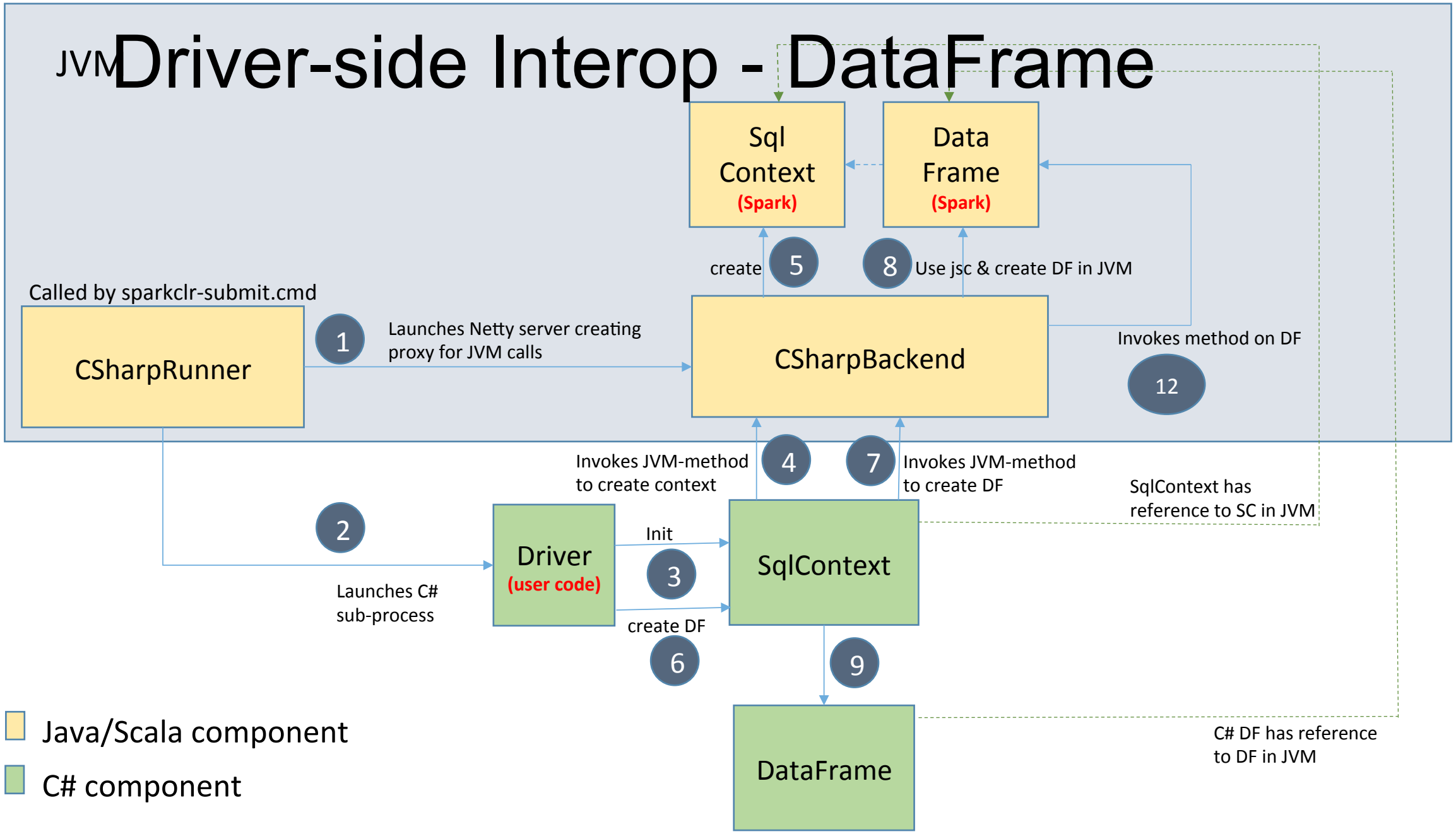
Mobius设计考量

- 避免重复实现spark的核心逻辑
- JVM – CLR (.NET VM) 互操作
  - Spark运行在JVM
  - C#需要CLR来执行
- 重用PySpark和SparkR的设计和代码

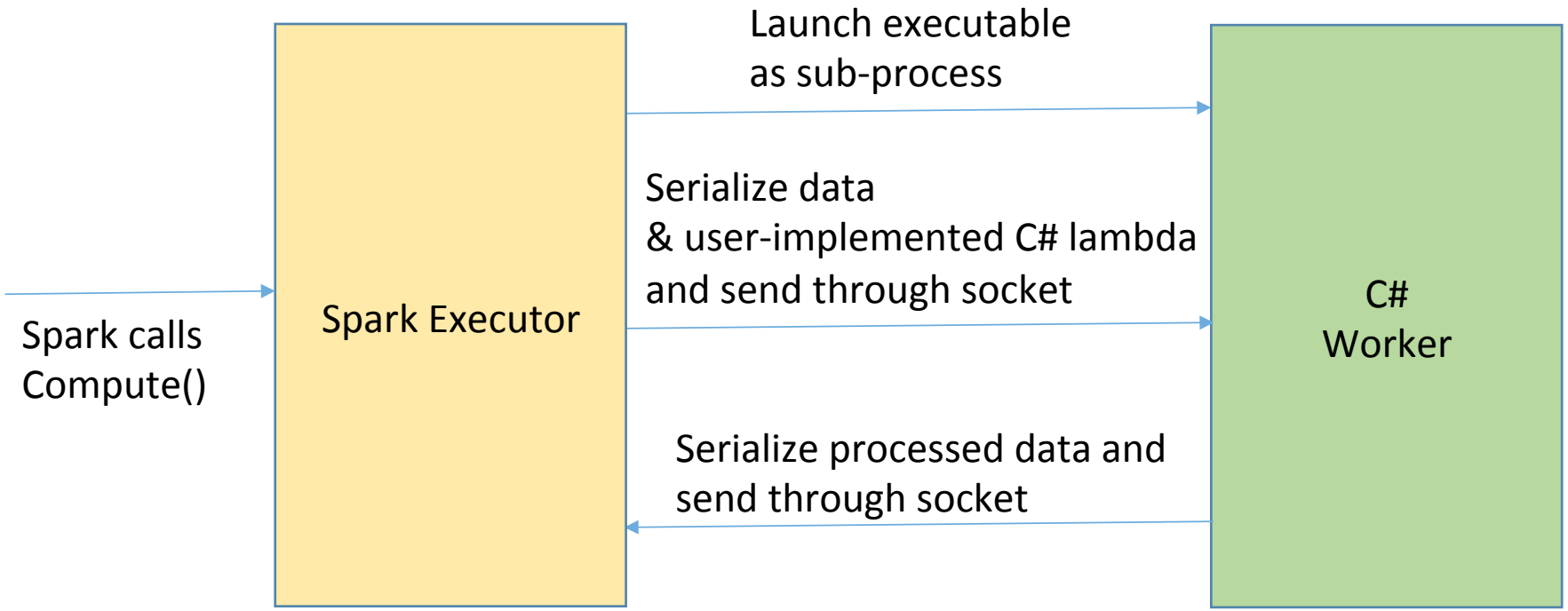
互操作


Workers
Driver
CLR
C# Driver
Method
IPC Sockets
Result
JVM
Java Driver
JVM
SparkExecutor
Method
IPC Sockets
Result
CLR
C# Worker
JVM
SparkExecutor
Method
IPC Sockets
Result
CLR
C# Worker
JVM
SparkExecutor
Method
IPC Sockets
Result
CLR
C# Worker

JVM Driver-side Interop - DataFrame



Executor-side Interop - RDD



Scala component

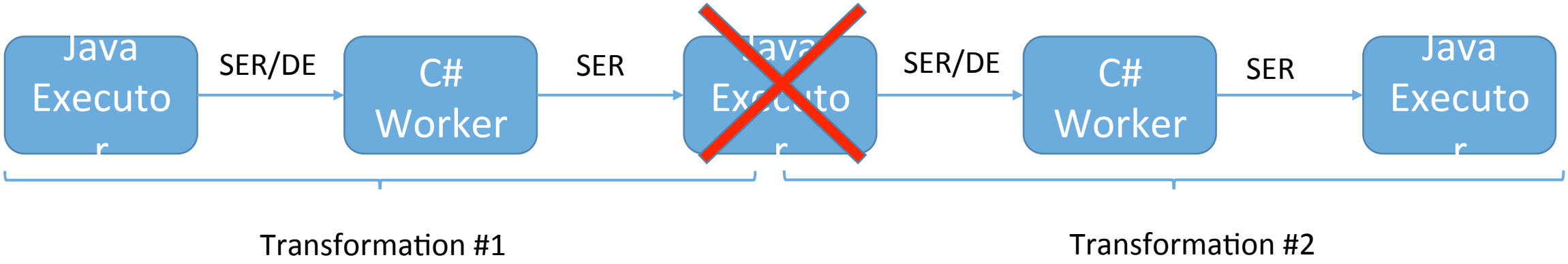
C# component

重用

- Driver端互操作
  - 类似于SparkR的设计
  - 使用Netty server作为JVM的代理
- Worker端互操作
  - 重用PySpark的实现
  - 启动外部进程，序列化/反序列化数据

Mobius性能考量

- Spark Executor重用CSharpWorker
  - 多线程代替多进程
- DataFrame操作无需启动CSharpWorker
  - 重用Spark Core中的所有优化（包括代码生成 codegen）
- 合并C#转换避免多余序列化/反序列化



Mobius进行时

- Mobius已经支持
  - 批处理
  - 流数据处理
  - 结构化关系数据处理
- Mobius即将支持
  - 交互式的数据处理
  - 托管服务

Mobiu命令行

- 交互式执行Spark应用的C#代码片段
- 避免了重复的编译、部署代码
- 应用结果直接展示给用户
- 便于交互式的数据分析，以及快速的开发测试

```
>
> using System.Linq;
> var rdd = sc.Parallelize(Enumerable.Range(0,10),2);
> Console.WriteLine(String.Join(",", rdd.Collect()));
0,1,2,3,4,5,6,7,8,9
> var sum = rdd.Reduce((x, y) => x + y);
WorkerFunc:Microsoft.Spark.CSharp.Core.CSharpWorkerFunc
Runmode:shell
[2016-04-14 13:58:42,787] [21416] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - rddInfo: rddId 25, stageId 13, partitionId 0
[2016-04-14 13:58:42,818] [21416] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - Run mode: shell
[2016-04-14 13:58:42,818] [21416] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - Total received assemblies count: 22
[2016-04-14 13:58:42,865] [21416] [1] [INFO ] [Microsoft.Spark.CSharp.WorkerInputEnumerator] - total elapsed time: 2
[2016-04-14 13:58:42,865] [21416] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - func process time: 9
[2016-04-14 13:58:42,865] [21416] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - command process time: 90
[2016-04-14 13:58:43,006] [11136] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - rddInfo: rddId 25, stageId 13, partitionId 1
[2016-04-14 13:58:43,037] [11136] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - Run mode: shell
[2016-04-14 13:58:43,037] [11136] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - Total received assemblies count: 22
[2016-04-14 13:58:43,084] [11136] [1] [INFO ] [Microsoft.Spark.CSharp.WorkerInputEnumerator] - total elapsed time: 2
[2016-04-14 13:58:43,084] [11136] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - func process time: 9
[2016-04-14 13:58:43,084] [11136] [1] [INFO ] [Microsoft.Spark.CSharp.Worker] - command process time: 79
> sum
45
>
```

Mobius命令行

- 基于Roslyn开发
  - 开源.NET编译器源码
  - 跨平台支持

Welcome to the .NET Compiler Platform ("Roslyn")

Windows - Unit Tests

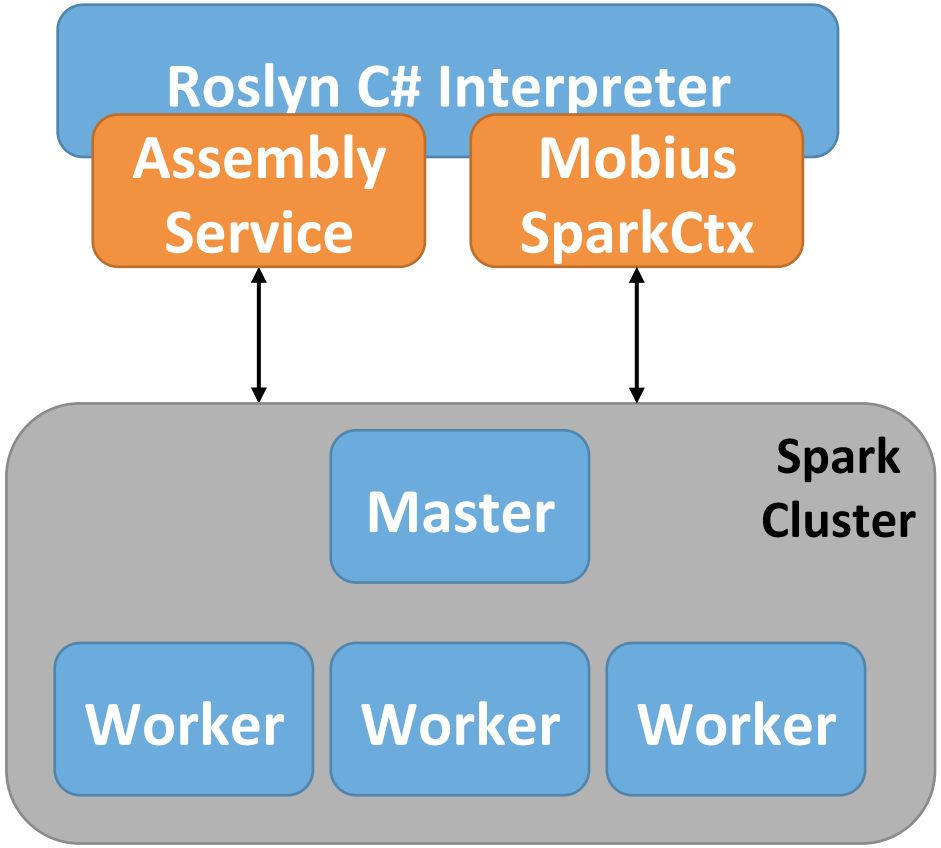
	Debug x86	Debug x64	Release x86	Release x64	Determinism
stabilization (1.2)	build passing	build passing	build passing	build passing	build passing
master (1.3)	build passing	build passing	build passing	build passing	build passing
future-stabilization (2.0 Preview)	build passing	build passing	build passing	build passing	build passing
future (2.0 RC)	build passing	build passing	build passing	build passing	build passing
hotfix	build passing	build passing	build passing	build passing	build passing

Linux/Mac - Unit Tests

	Linux	Mac OSX
stabilization (1.2)	build passing	build passing
master (1.3)	build passing	build passing
future-stabilization (2.0 Preview)	build passing	build passing
future (2.0 RC)	build passing	build passing

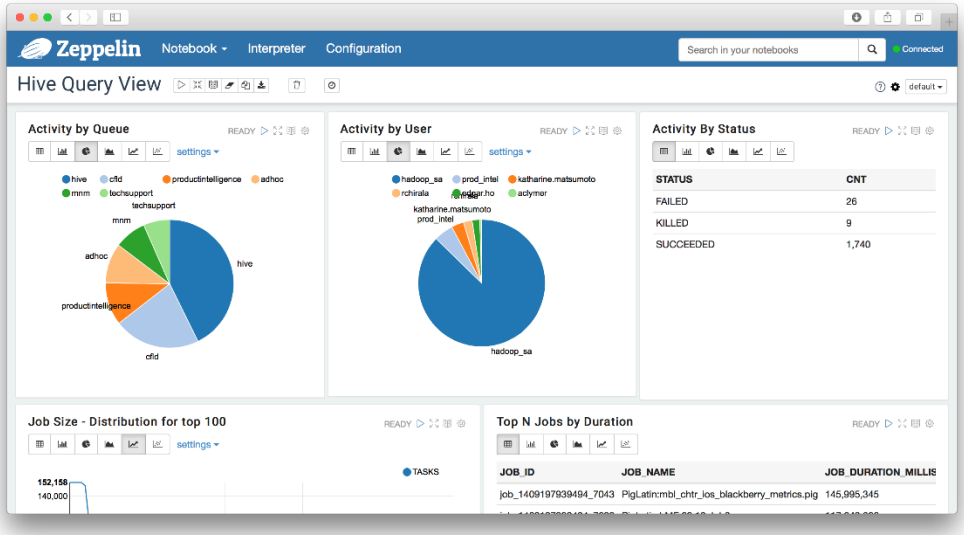
Mobius命令行

- 基于Roslyn开发
- 代码的即时编译和即时部署

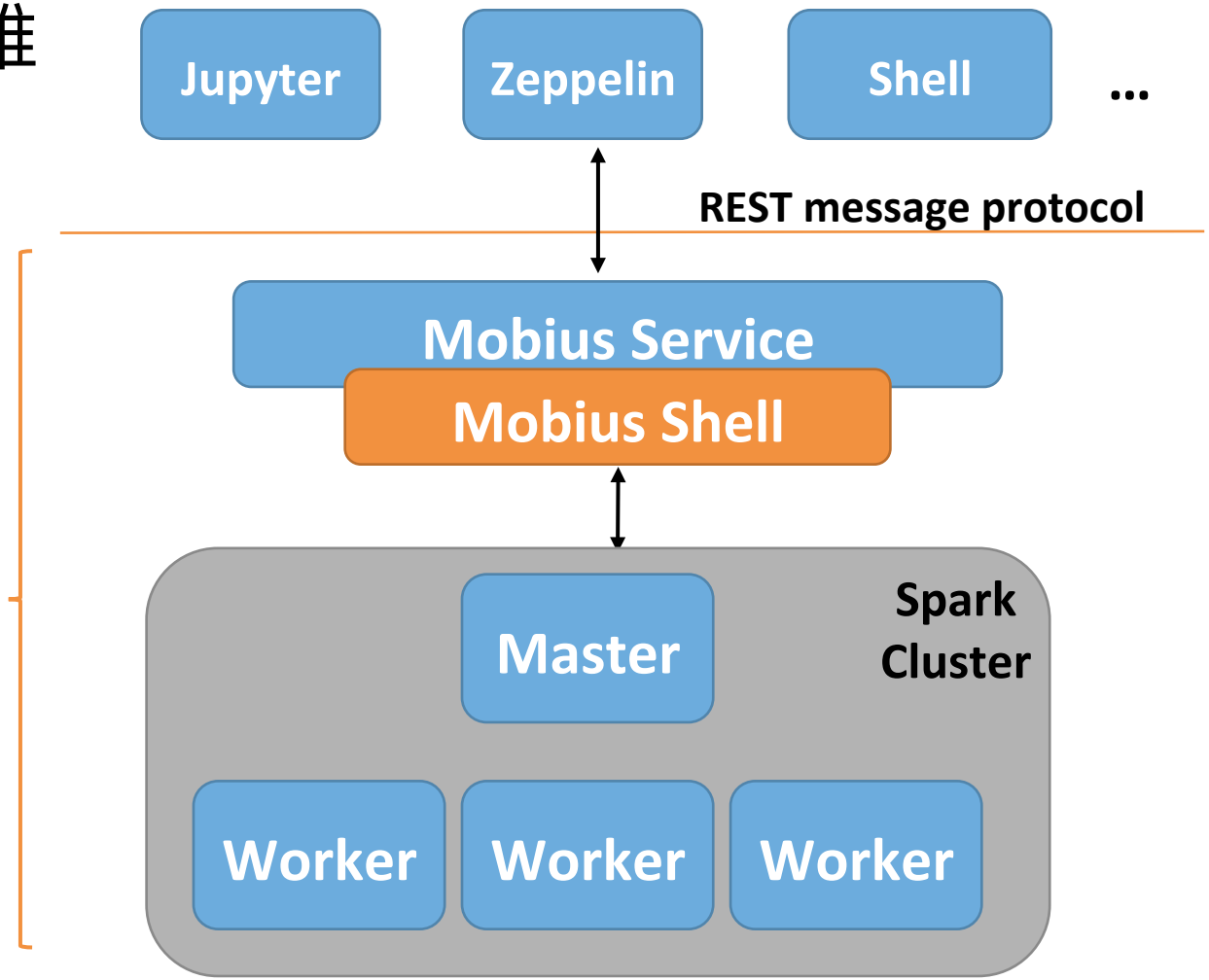


Mobius 即服务（as a service）

- 使用托管的Spark集群，节省运维
- 集成丰富的数据可视化工具
- 灵活的商业模式Pay-by-Job

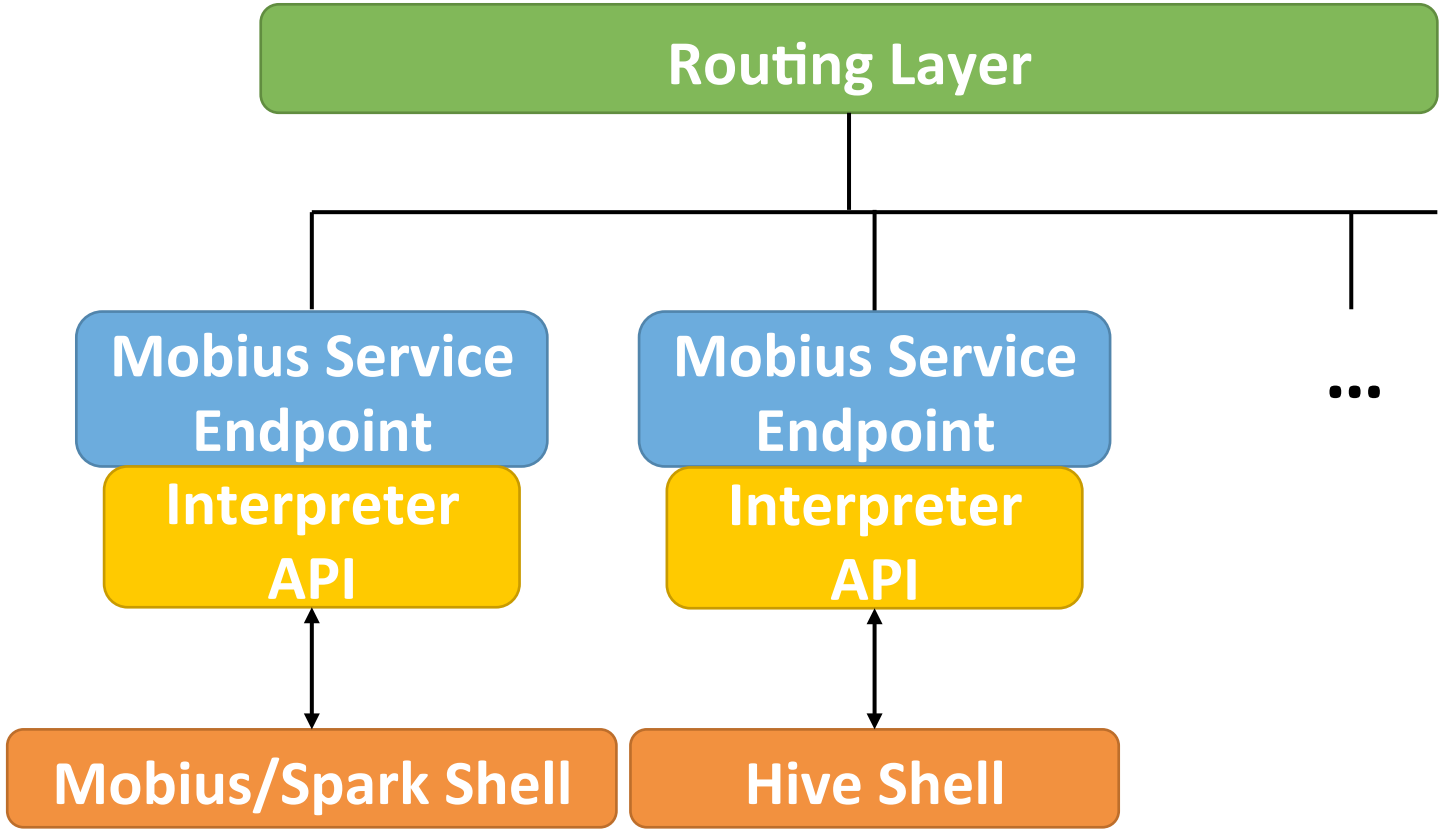


Hosted Service



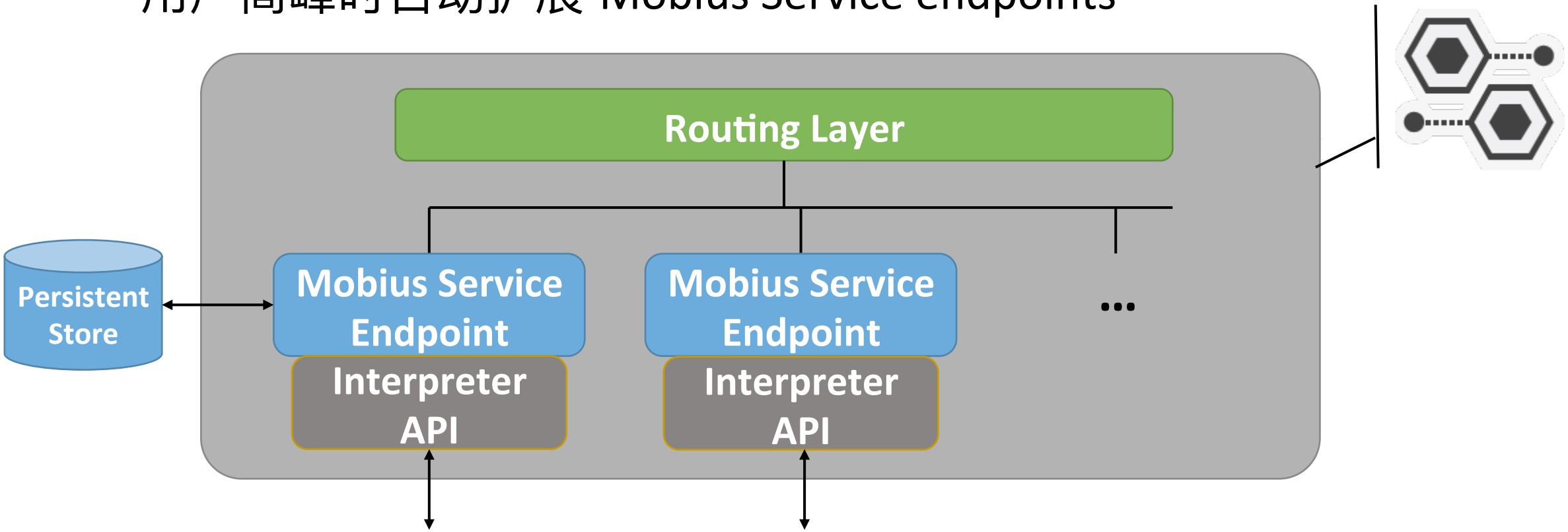
Mobius 服务核心特色

- 更通用的 API，便于集成多种执行引擎



Mobius 服务核心特色

- 伸缩性(Elasticity)和可靠性(Fault Tolerance)
  - 用户高峰时自动扩展 Mobius Service endpoints



加入我们的社区！

- 欢迎加入Mobius的开发社区
  - 使用Mobius开发Spark应用提供反馈
  - 贡献代码@ github.com/Microsoft/Mobius
- Any contribution is welcome!
  - 数据可视化工具的集成: Jupyter/Zeppelin/…
  - 自动部署工具的集成: Puppet/Chef/…
  - …

云计算实验室(CISL)

- Mobius
- Yarn++
- Apache REEF
- Rayon
- Tiered Storage
- …

加入我们的社区！

- 欢迎加入Mobius的开发社区
  - 使用Mobius开发Spark应用提供反馈
  - 贡献代码@ github.com/Microsoft/Mobius
- Any contribution is welcome!
  - 数据可视化工具的集成: Jupyter/Zeppelin/…
  - 自动部署工具的集成: Puppet/Chef/…
  - …
- 关注微软云计算实验室（CISL）
- 曾凯，kaizeng@microsoft.com

谢谢！

Interactive Log Analysis

- Massive Cosmos log analysis: several PBs per day
- Rapid iterative drill-downs for DRIs to diagnose issues

Spark?

AutoPilot Watchdog Alerts

Customer (or other DRIs)

Alerting

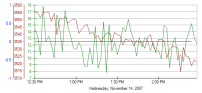
DRI
(Designated Responsible Individual)

Vendors/Secondary DRIs
Eavesdrop and document incident

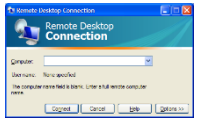
Triage

Cosmos

Scope Studio



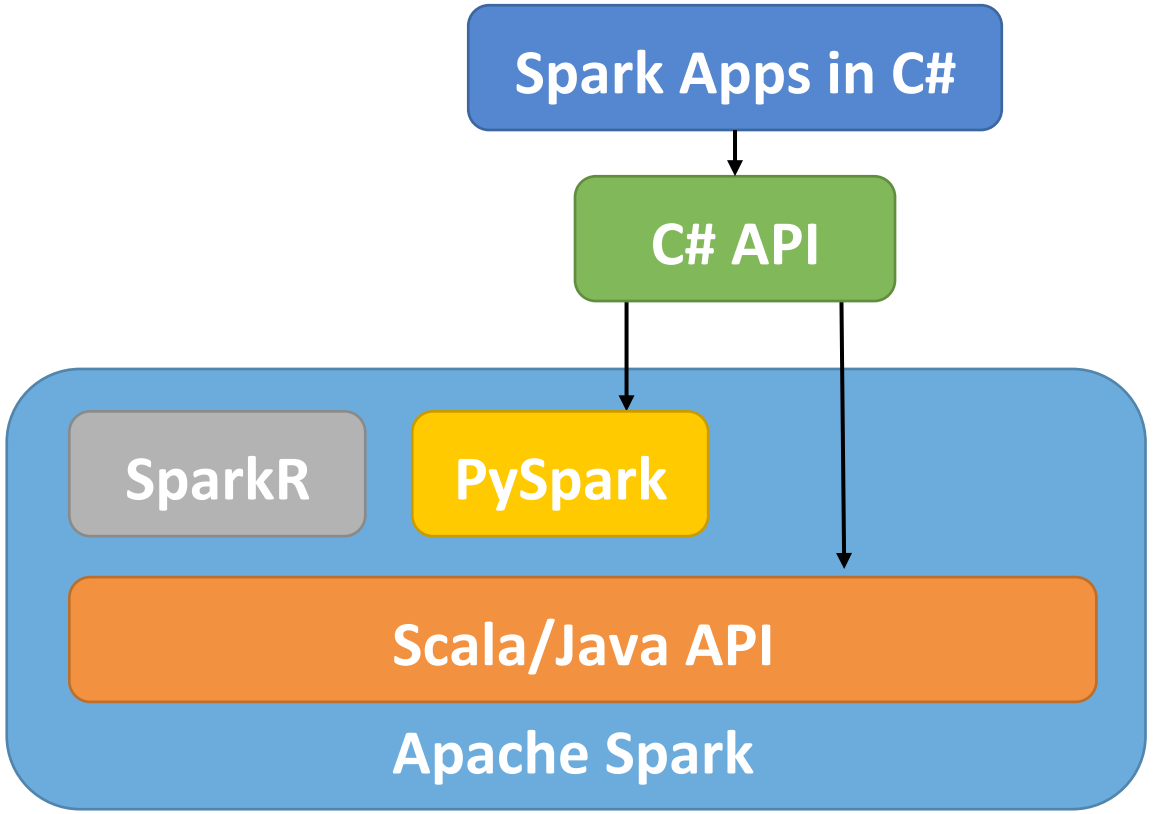
Perf Counters
Examine 1 hour, 2 hours… 14 days



RDP — Cosmos Machines
Open individual connections for troubleshooting

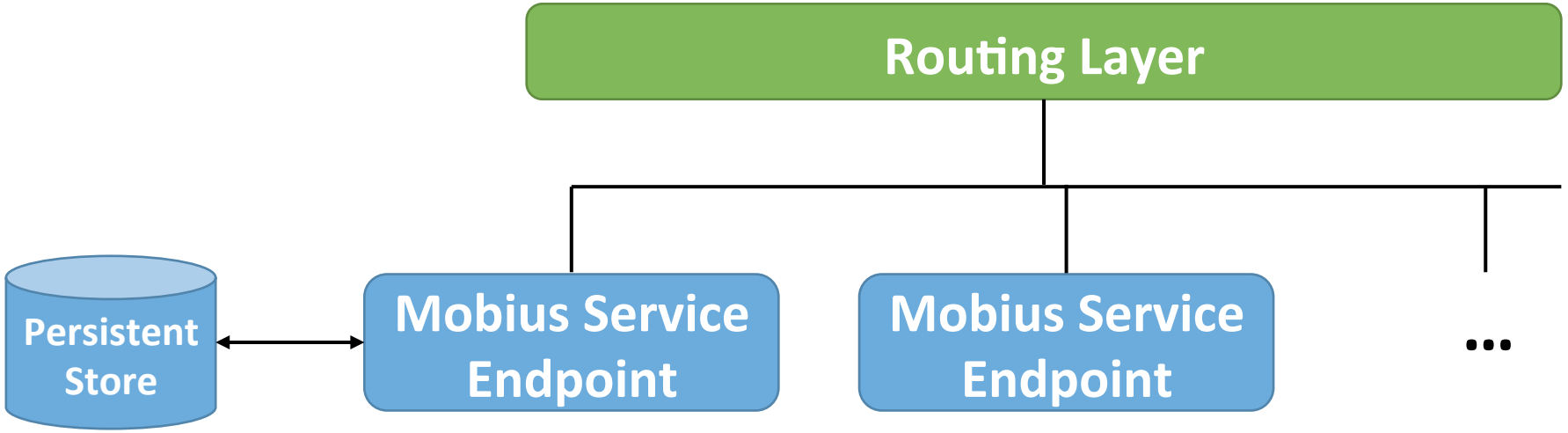
Architects/Developers/Other DRIs

C# API for Spark



Mobius Service Differentiators

- Better fault tolerance--session persistence and replay



Mobius Service Differentiators

- Elasticity and Fault Tolerance
  - Auto-scale # of Mobius Service endpoints

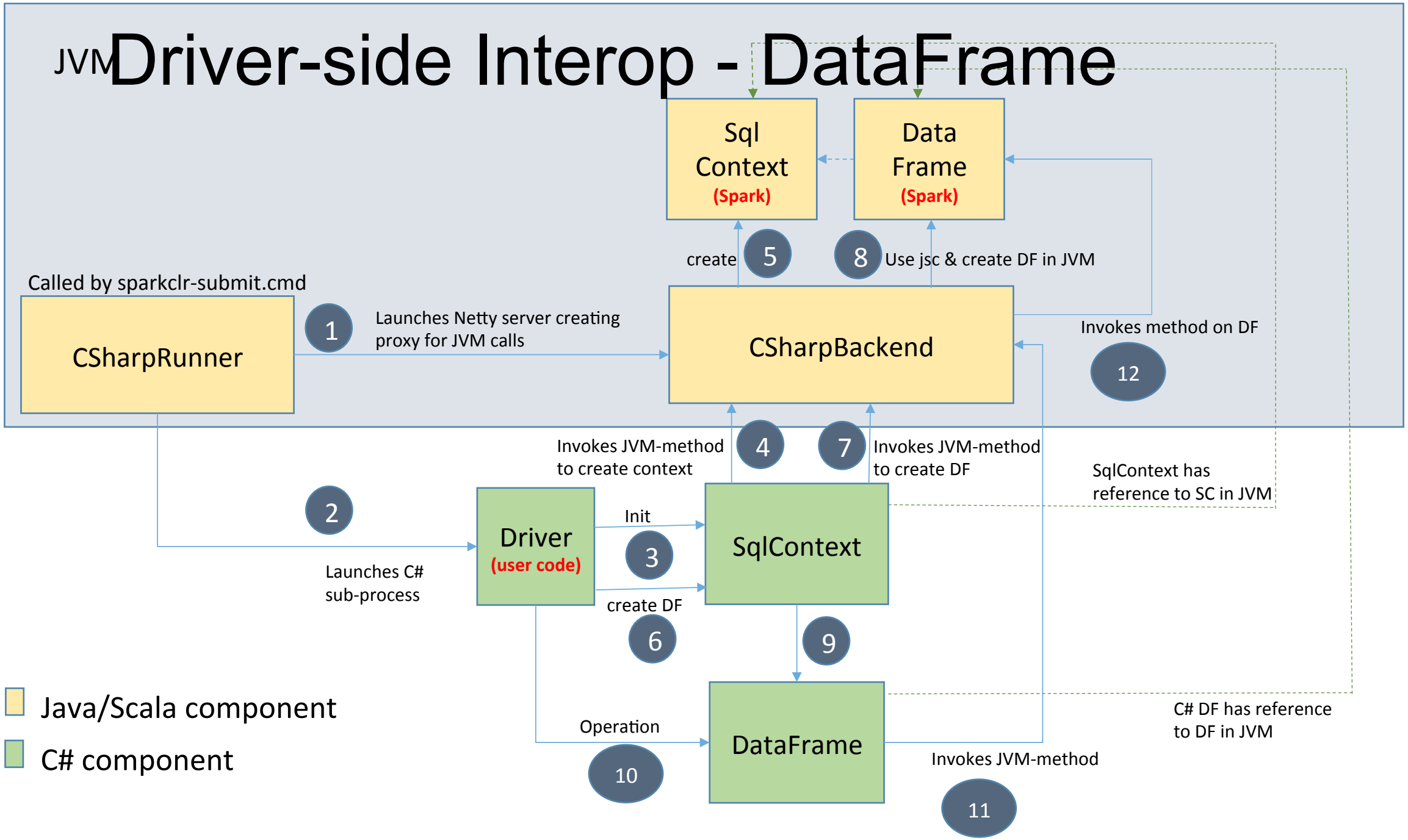
Orleans - Distributed Actor Model

build passing | nuget v1.1.3 | pull requests closed in about 15 hours | issues closed in 4 days

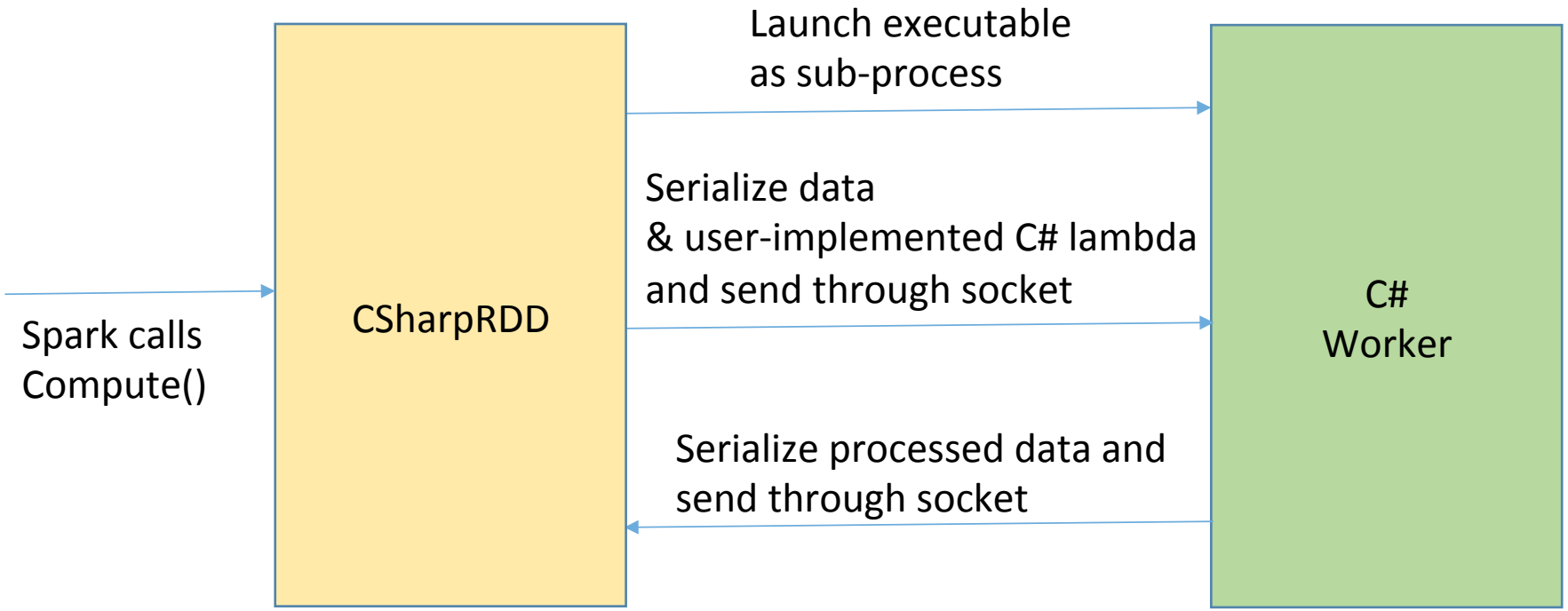
gitter join chat

Help Wanted Issues 40

JVM Driver-side Interop - DataFrame



Executor-side Interop - RDD



CSharpRDD is used only when customized C# code is used in transformation

Scala component

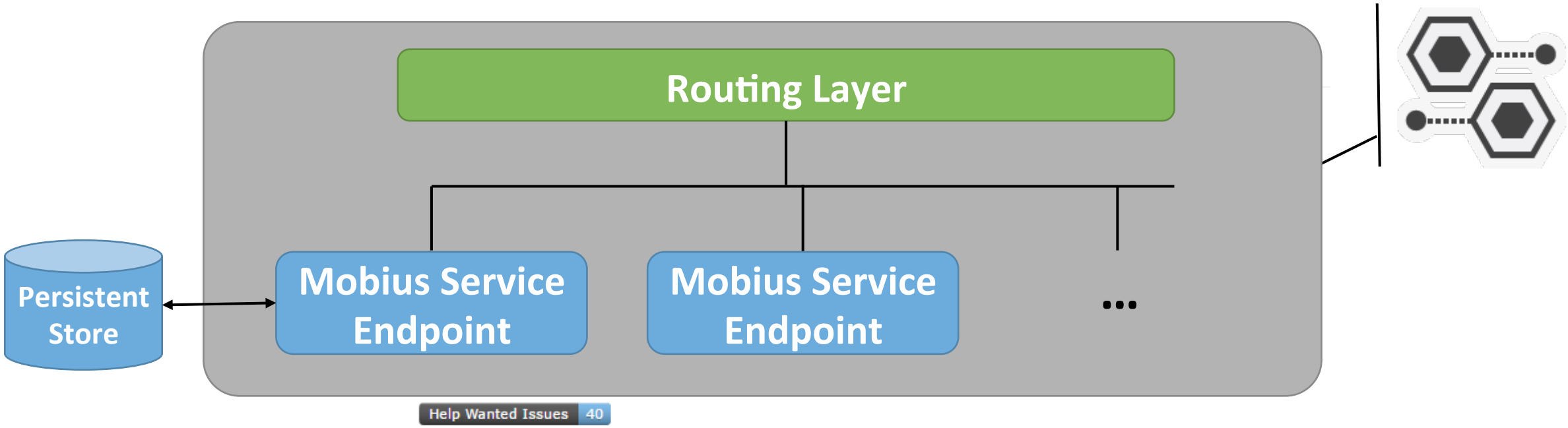
C# component

Mobius Performance Considerations

1. One CSharpWorker process for each JVM executor process
   - PySpark forks a Python process for each task thread in JVM executor process
   - New C# option of one thread for each task thread in JVM executor process
2. C# operations are pipelined when possible
   - Map & Filter RDD operations in C# need data to be passed from JVM to C#, incurring the cost of serialization and deserialization
   - C# operations are pipelined when possible to minimize data passing
3. DataFrame operations without C# UDFs do not require CSharpWorker
   - Same execution plan optimization and code generation in Spark Core
   - Perform the same as Scala applications

Mobius Service Differentiators

- Elasticity and Fault Tolerance
  - Use ***virtual actor*** model
  - Auto-scale # of Mobius Service endpoints



Linux Support

- Mono and CoreCLR (ongoing) for Mobius on Linux
- GitHub project uses Travis for CI in Ubuntu 14.04.3 LTS
  - Unit tests and samples (functional tests) are run
- More info @ linux-instructions.md

CSharpRDD

- C# 需要CLR来执行
  - 没有C#转换 => 执行全部依靠JVM
- RDD<byte[]>
  - 序列化C# worker的输出
- Transformations are pipelined when possible
  - 避免不必要的（反）序列化开销