Scaling and Securing Spark on Kubernetes

Bloomberg

Engineering

Kubecon Europe 2019
May 23, 2019

Ilan Filonenko, ifilonenko@bloomberg.net
Software Engineer, Data Science Infrastructure

TechAtBloomberg.com

Agenda

- Data Science at Bloomberg
- Securing Spark at Bloomberg
- Scaling Spark at Bloomberg with Disaggregated Compute
- Future work

TechAtBloomberg.com

Bloomberg

Engineering

Data Science at Bloomberg

TechAtBloomberg.com

Bloomberg

Engineering

Data Science Platform

Bloomberg has developed a unified, multi-tenant compute environment which allows our engineers to orchestrate, manage, and pipeline their data science workflows.

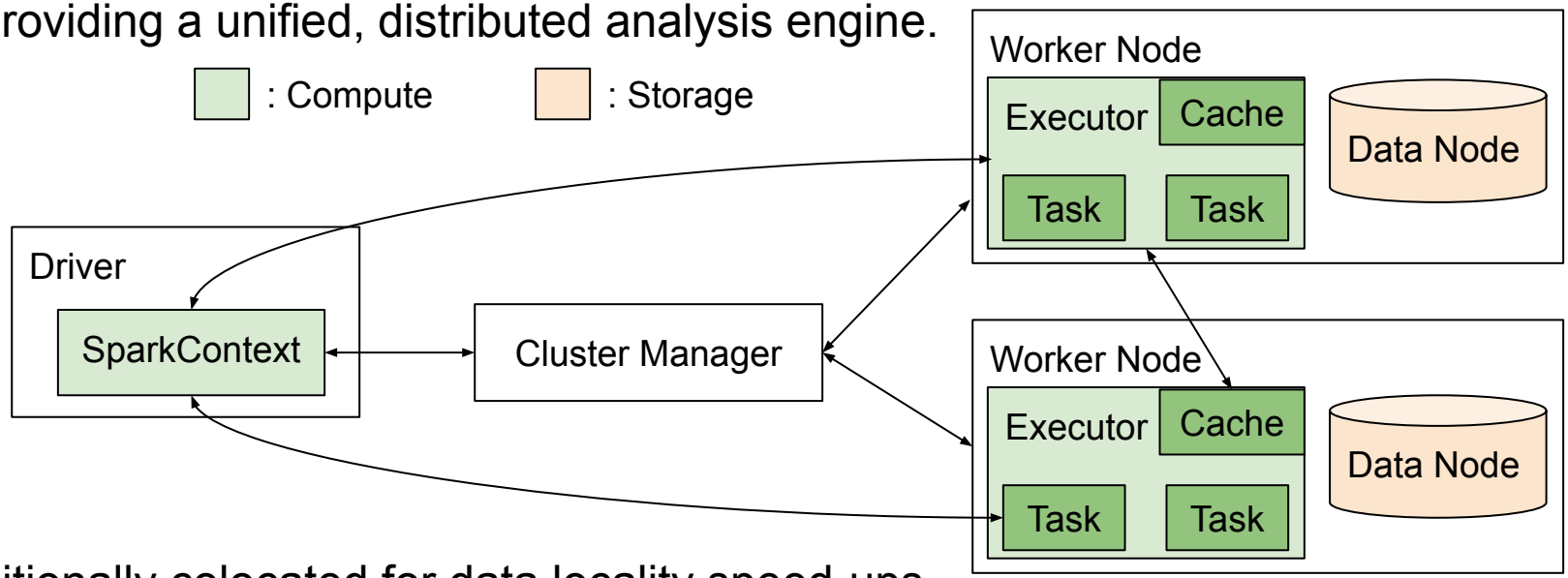
- Variety of ETL and training jobs: Tensorflow, **Spark**, Hypertuning, …
- Identity management: **Kerberized HDFS**, **S3,** Git
- Resource governance: Shared workspaces, resource quotas
- Lambda Inference: Knative service (FAAS) for model inference

Bloomberg

Engineering

Primary ETL component: Apache Spark

Apache Spark plays an integral role, functioning as a robust framework for providing a unified, distributed analysis engine.



Traditionally colocated for data locality speed-ups

TechAtBloomberg.com

Bloomberg

Engineering

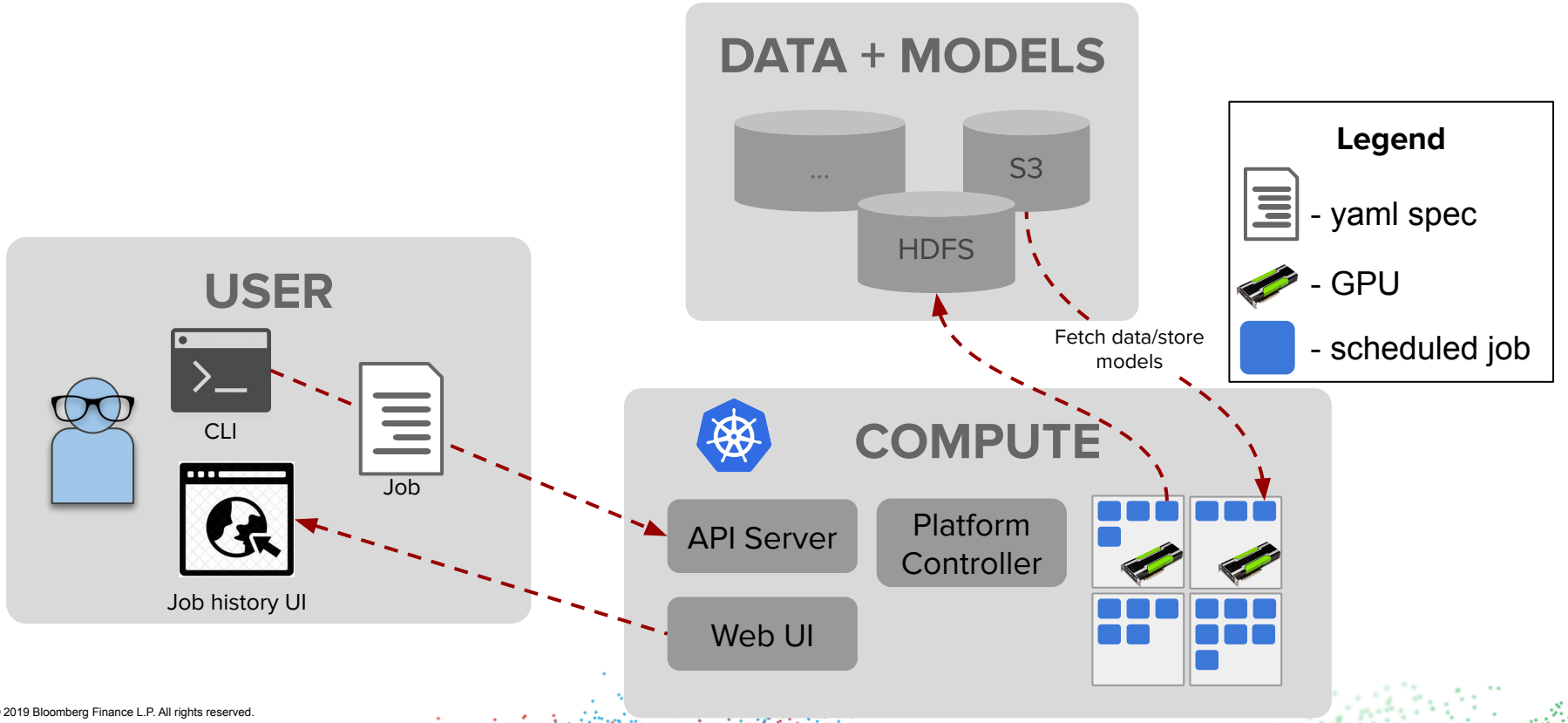
Colocated vs. Disaggregated Compute

Disaggregated: separate clusters for both storage and compute

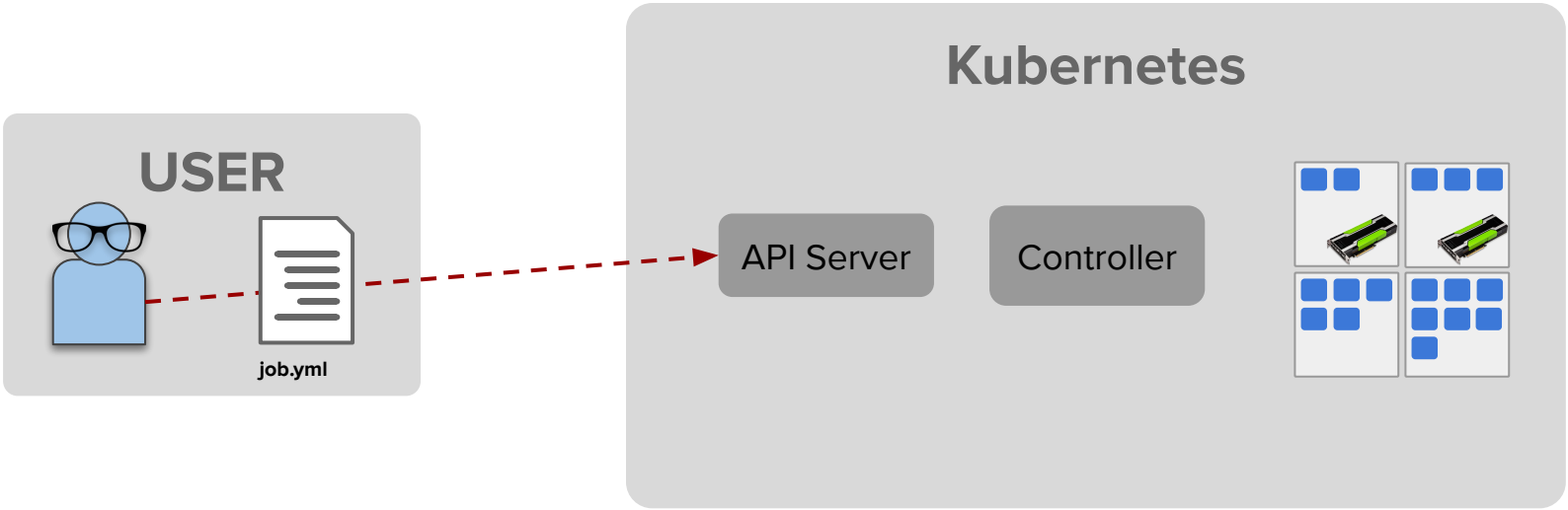
Advantages of Disaggregated Compute:

- Storage:
  - Compute nodes are not necessarily optimized for some large tasks where the writes may run out of disk space
- Hardware:
  - Hardware can be optimized for either compute or storage
  - Cluster management / upgrades can be done separately
  - Allows you to elastically bring up compute notes while storage nodes remain persistent

Data Science Platform architecture



Example Job CRD Lifecycle

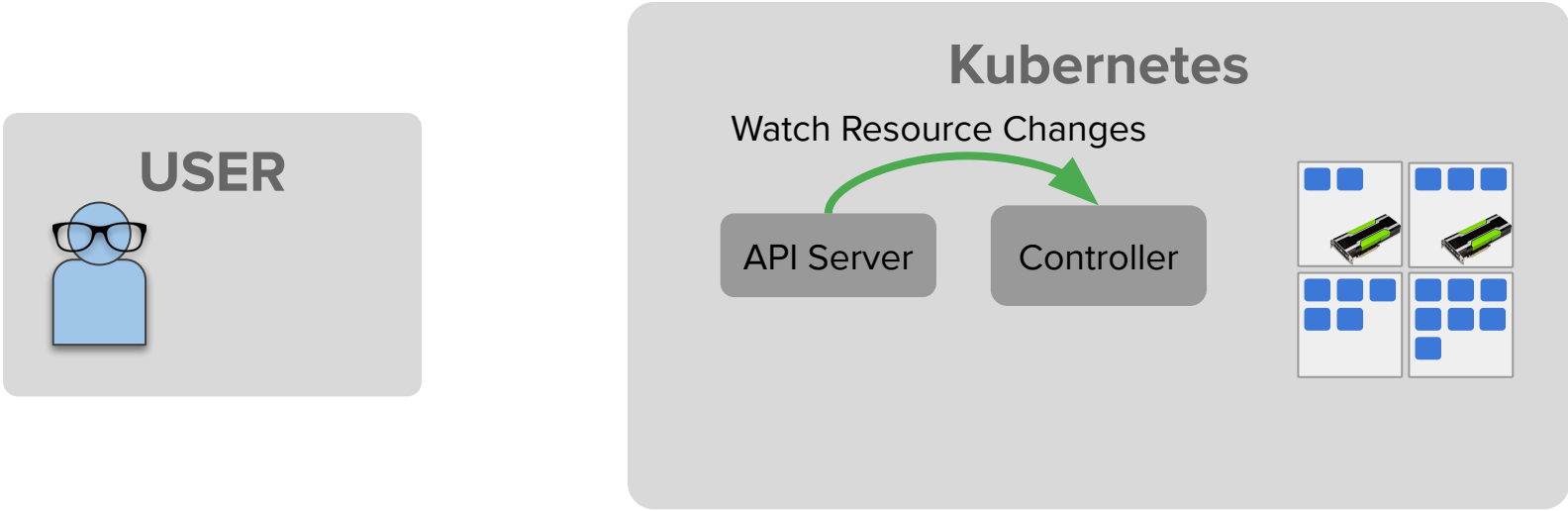


TechAtBloomberg.com

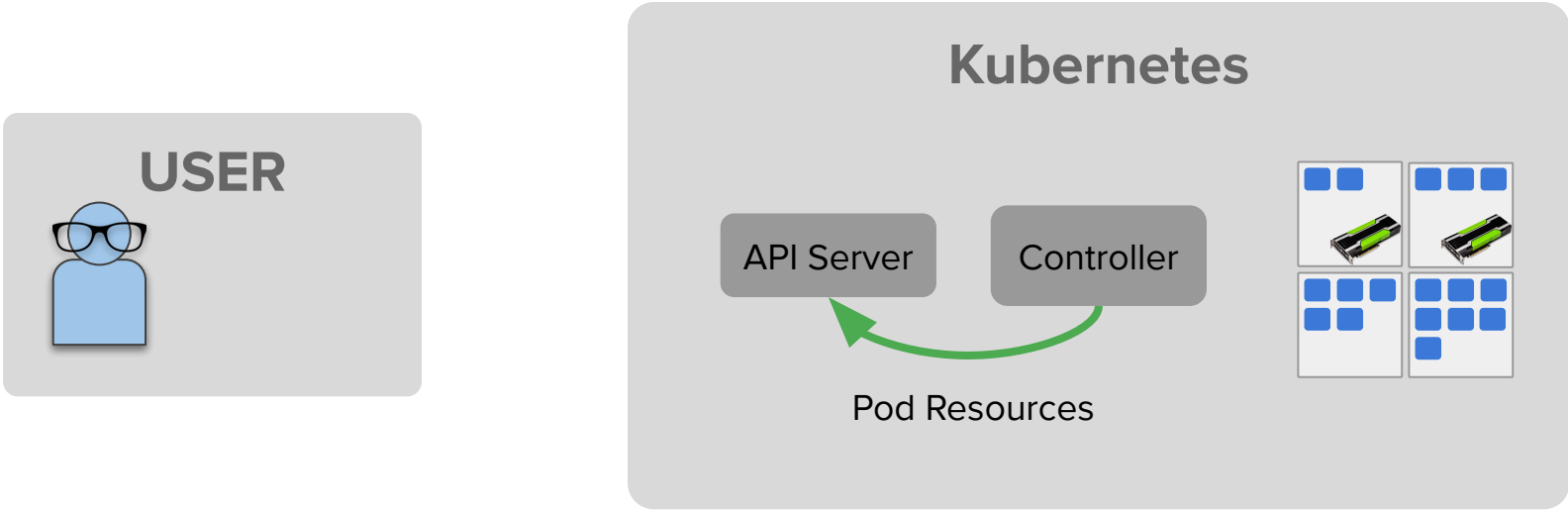
Bloomberg

Engineering

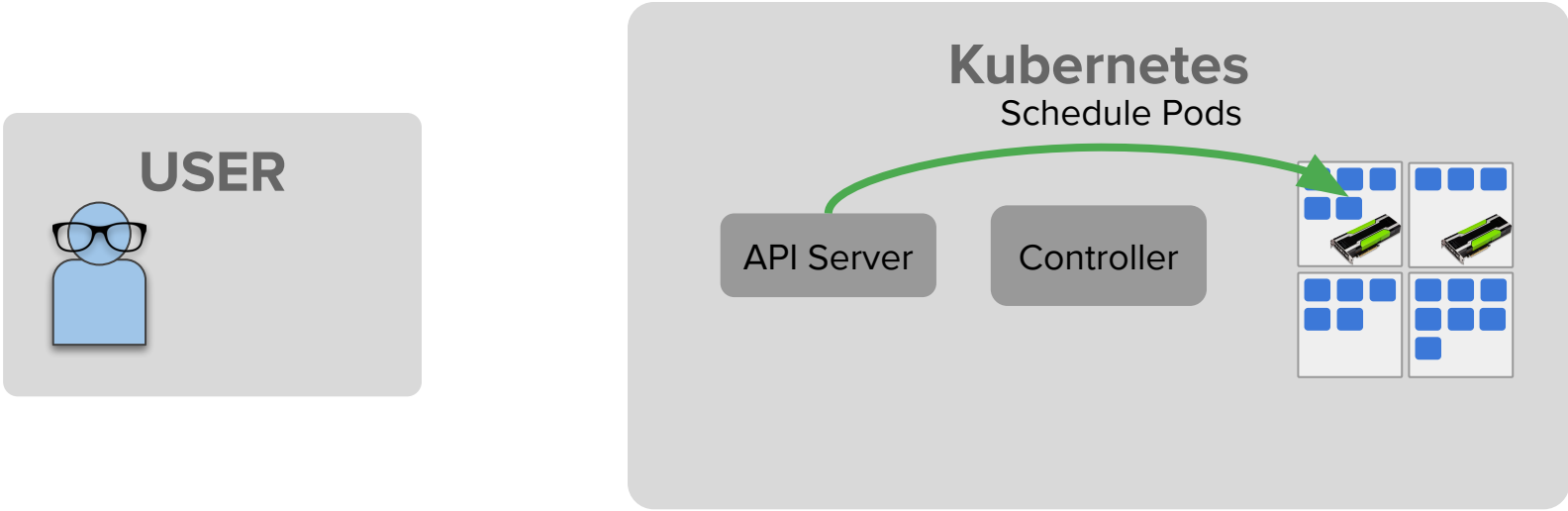
Example Job Lifecycle in Kubernetes



Example Job Lifecycle in Kubernetes



Example Job Lifecycle in Kubernetes



TechAtBloomberg.com

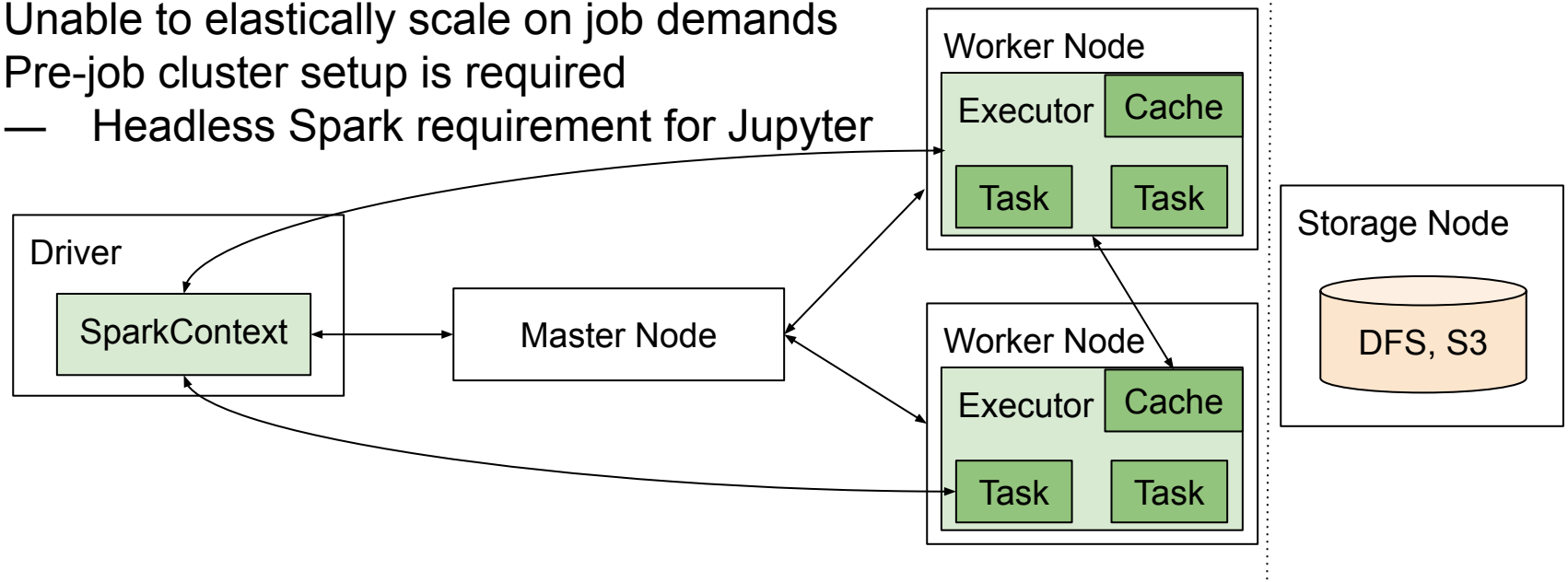
Bloomberg

Engineering

Spark Standalone on Kubernetes

Drawbacks:

- Unable to elastically scale on job demands
- Pre-job cluster setup is required
  - Headless Spark requirement for Jupyter

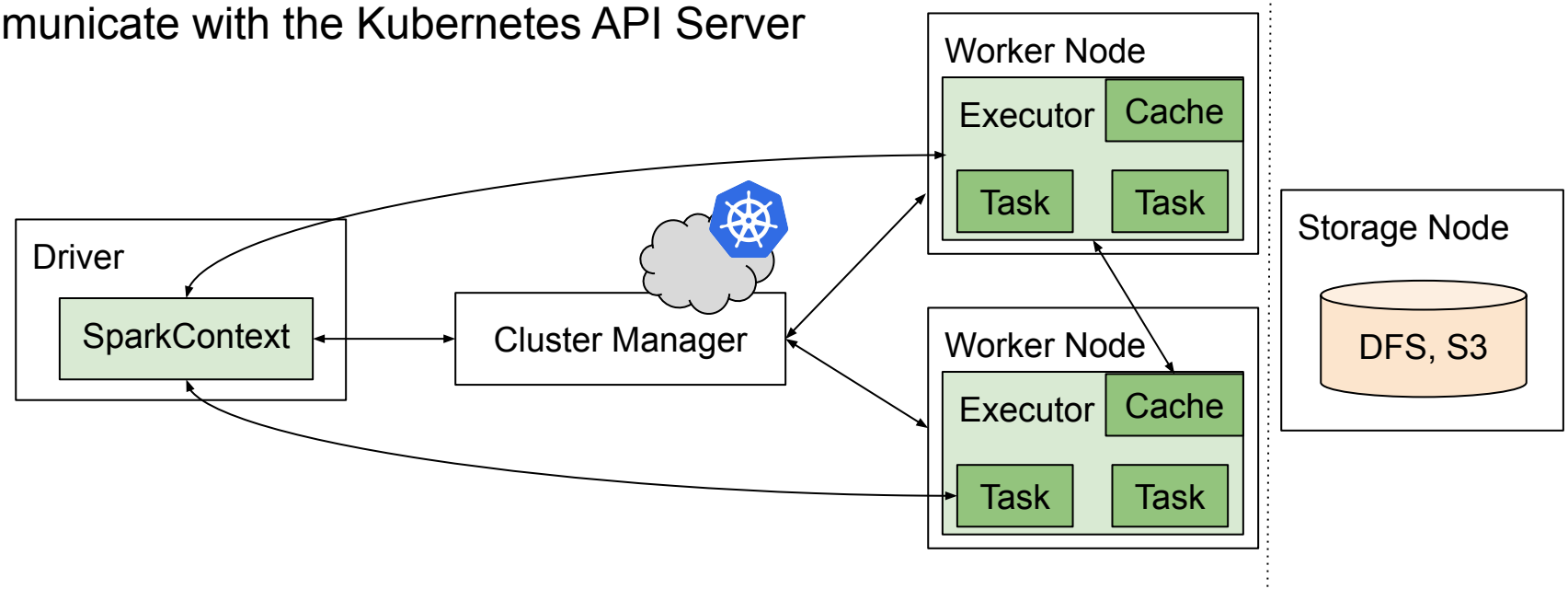


Bloomberg

Engineering

Spark Native on Kubernetes

With Bloomberg's work on native integration, Spark can now directly communicate with the Kubernetes API Server



TechAtBloomberg.com

Bloomberg

Engineering

Challenges that still exist in Spark on K8S

Cluster Security

- Administrators might lock down the cluster to not allow pods that are launched in a user's namespace to create customized pods

Secure Data Communication

- Making secure data retrieval a first-class citizen with a managed identity service

Disaggregated Compute

- Extending Spark to work in a Disaggregated environment

Bloomberg

Engineering

Securing Spark at Bloomberg

TechAtBloomberg.com

Bloomberg

Engineering

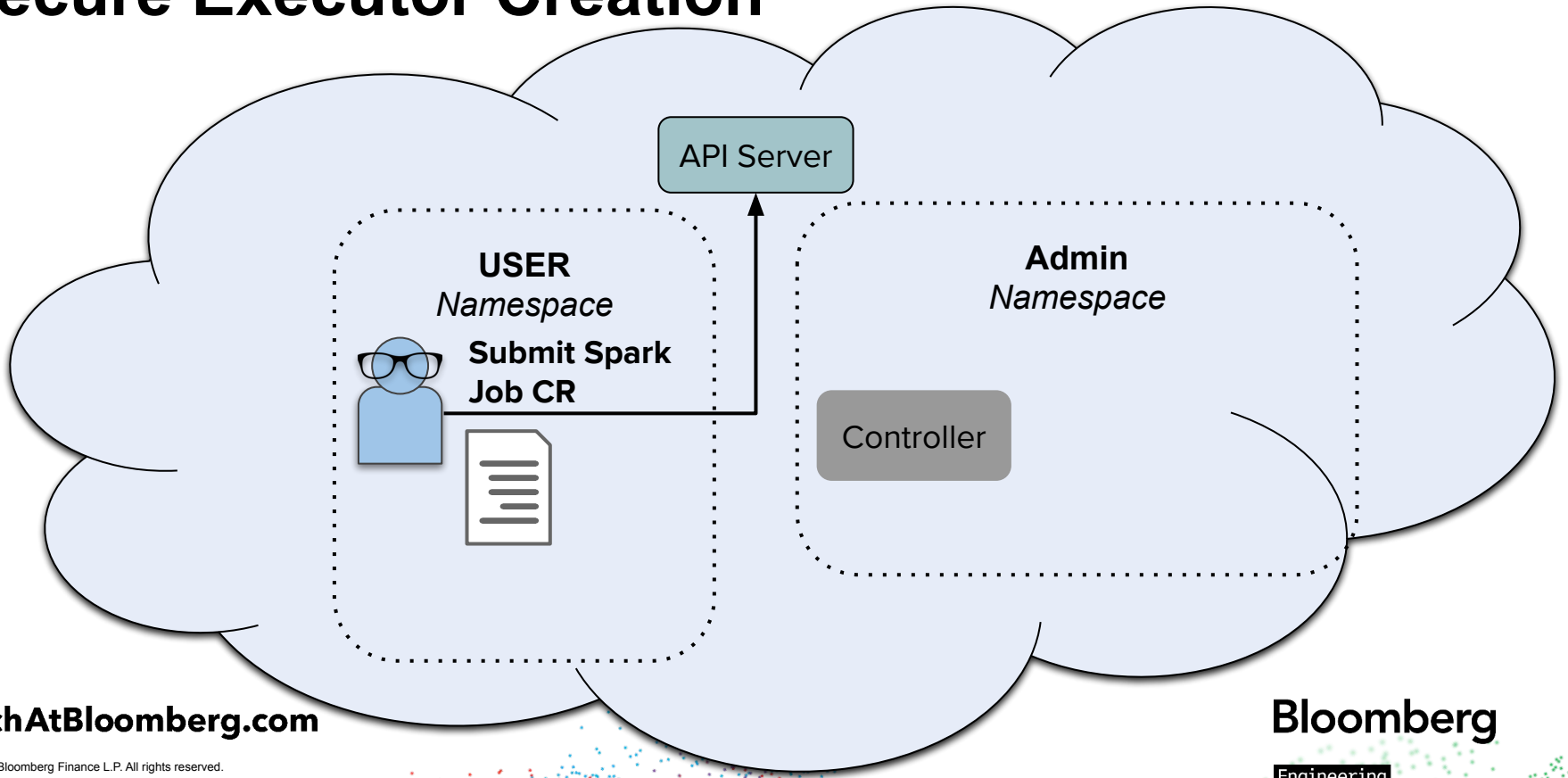
Increased Security

Native provides elastic creation of Executor Pods via API Server

Driver Pod now functions as a Controller

- Problems:
  - Cluster Administrators can have strict security policies that restrict users from launching custom pods in a user's namespace
  - Not relying on a global webhook on all pods
  - No ability to toggle executor creation strategies
- Solution:
  - A pluggable interface that hooks into the **KubernetesClusterSchedulerBackend**
    - defaults with **Pods**, but can be extended to create / update a **CR** called **ExecPodScaler**

Secure Executor Creation

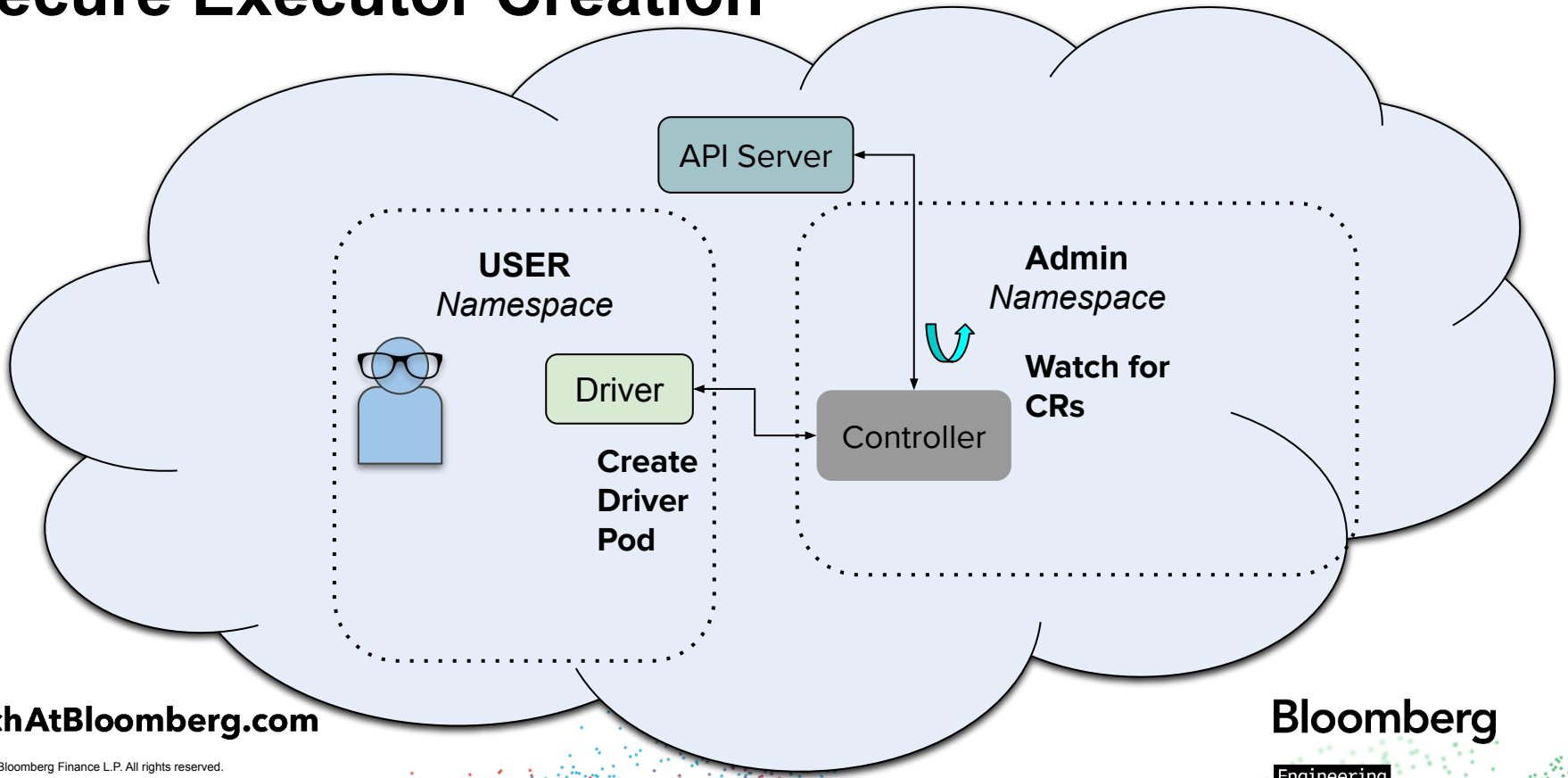


TechAtBloomberg.com

Bloomberg

Engineering

Secure Executor Creation

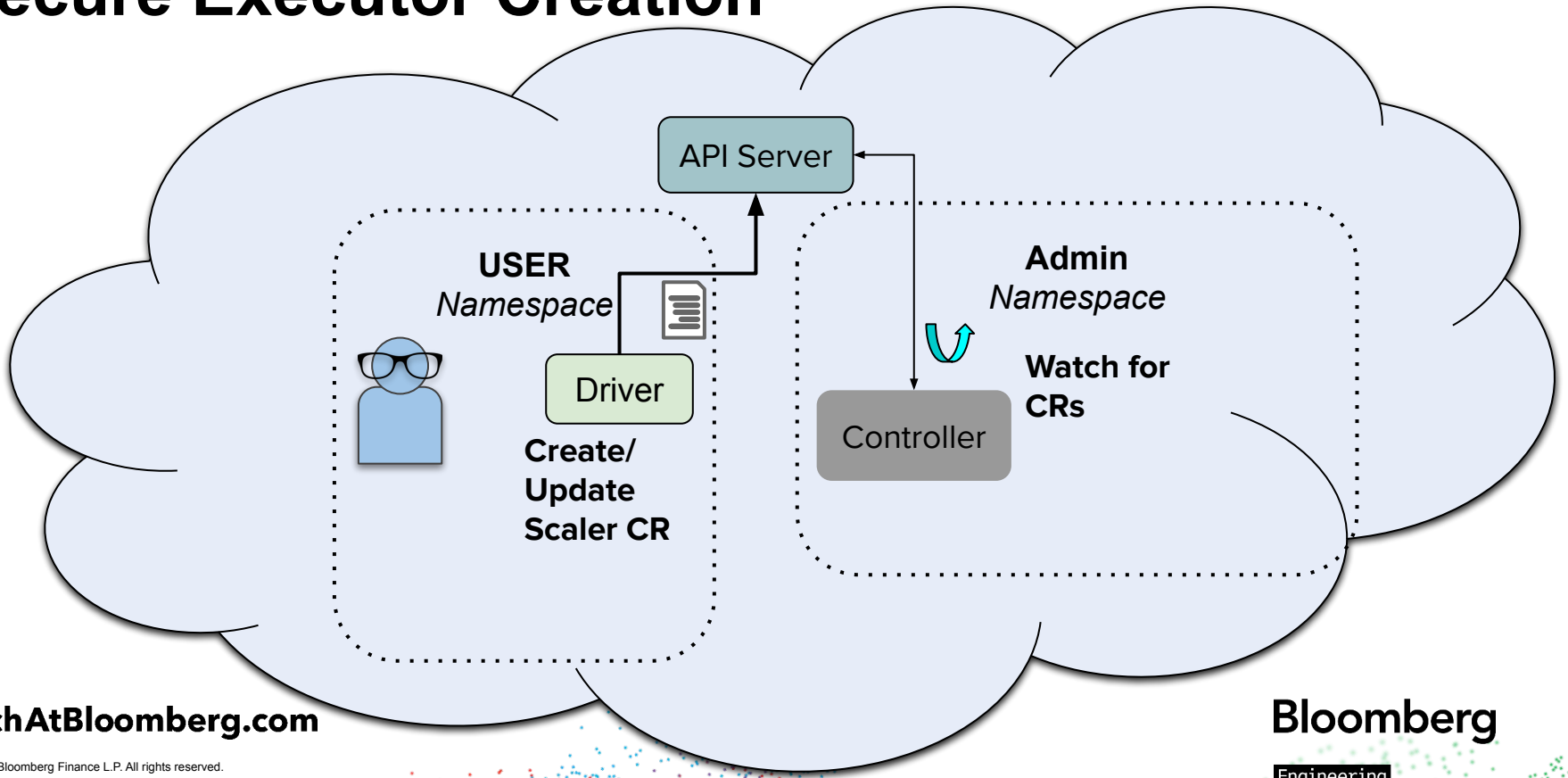


TechAtBloomberg.com

Bloomberg

Engineering

Secure Executor Creation

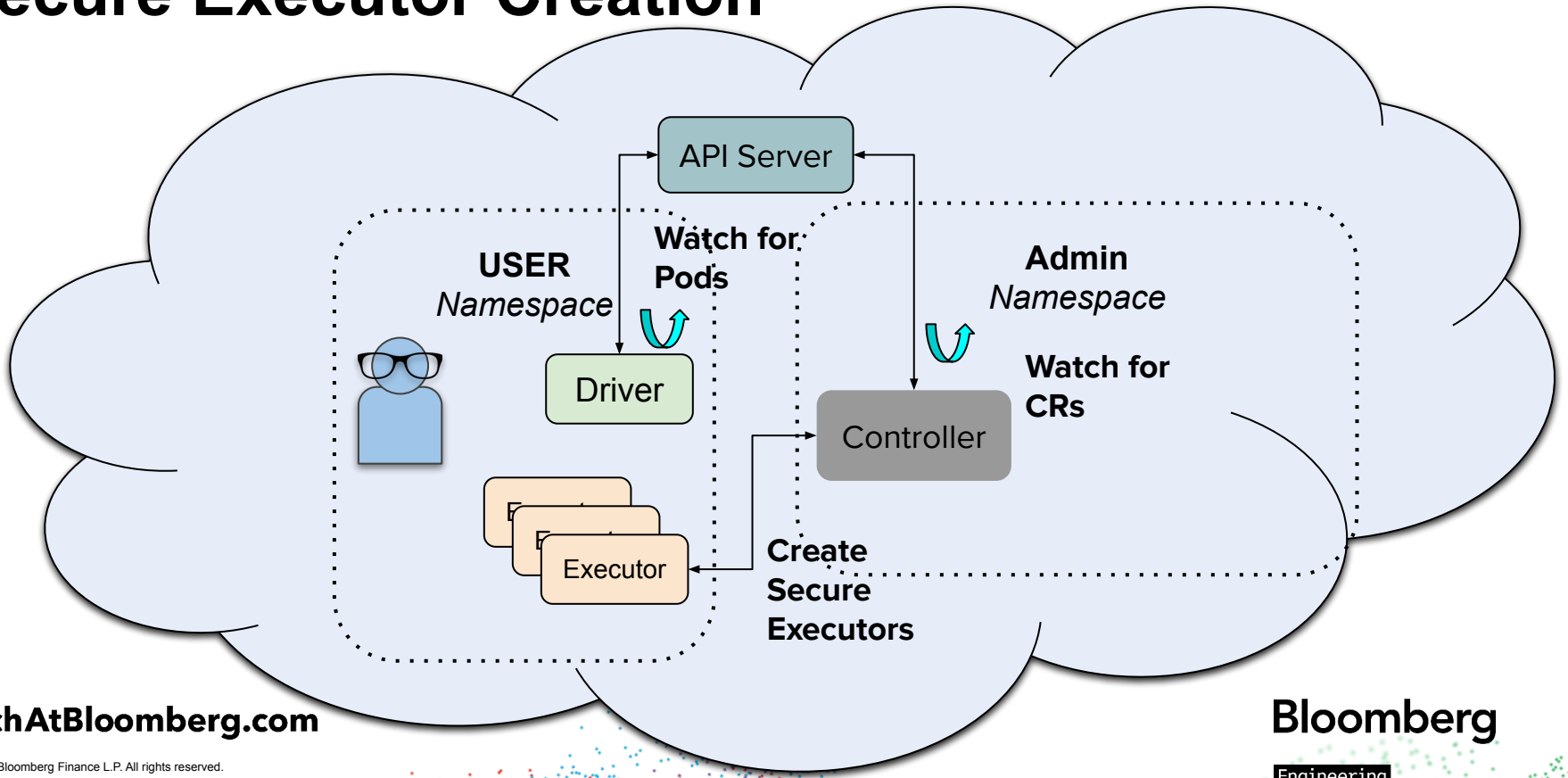


TechAtBloomberg.com

Bloomberg

Engineering

Secure Executor Creation



TechAtBloomberg.com

Bloomberg

Engineering

Secure Data Communication

Spark can support a variety of data sources through the DataFrame interface. At Bloomberg, our Data Sources are always secure and can only be accessible with an authorized and validated identity.

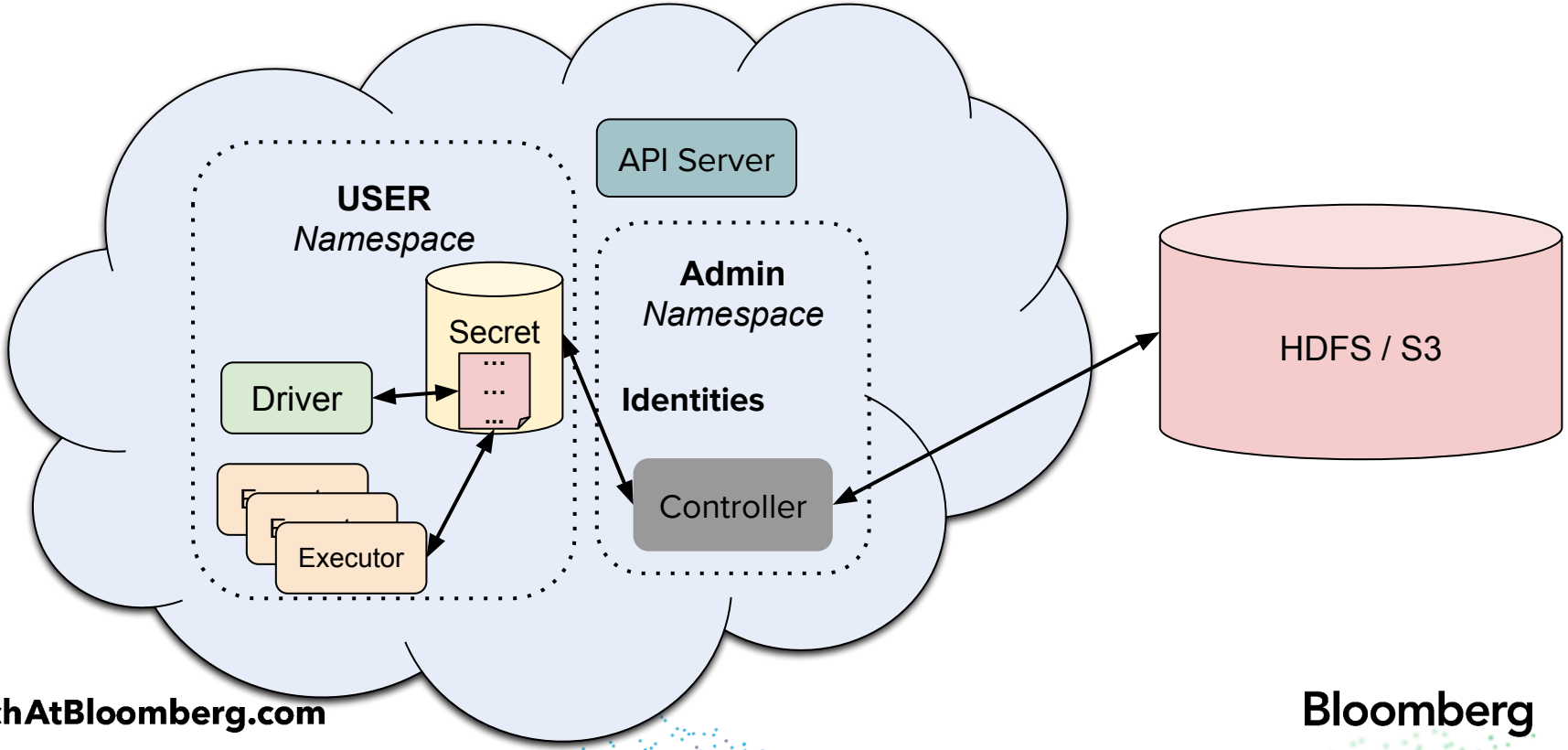
To simplify the data science experience, we provided a managed token service for Kerberized HDFS and S3:

- Simple inclusion of identity name within the Job specification
- Automated Hadoop Token renewal by the Job Controller
- Standardized token logic across all jobs

Bloomberg

Engineering

Secure Data Communication

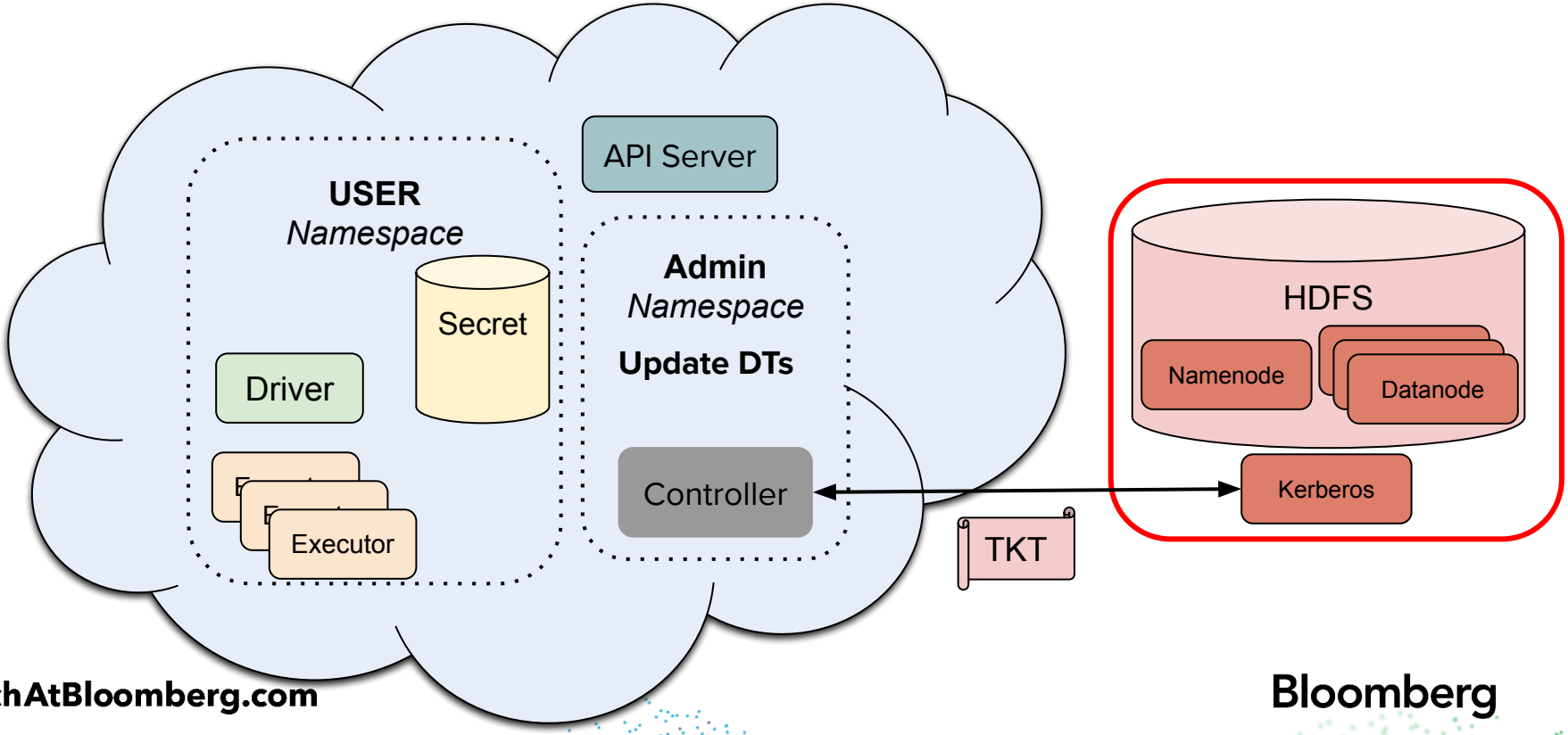


TechAtBloomberg.com

Bloomberg

Engineering

Secure Data Communication

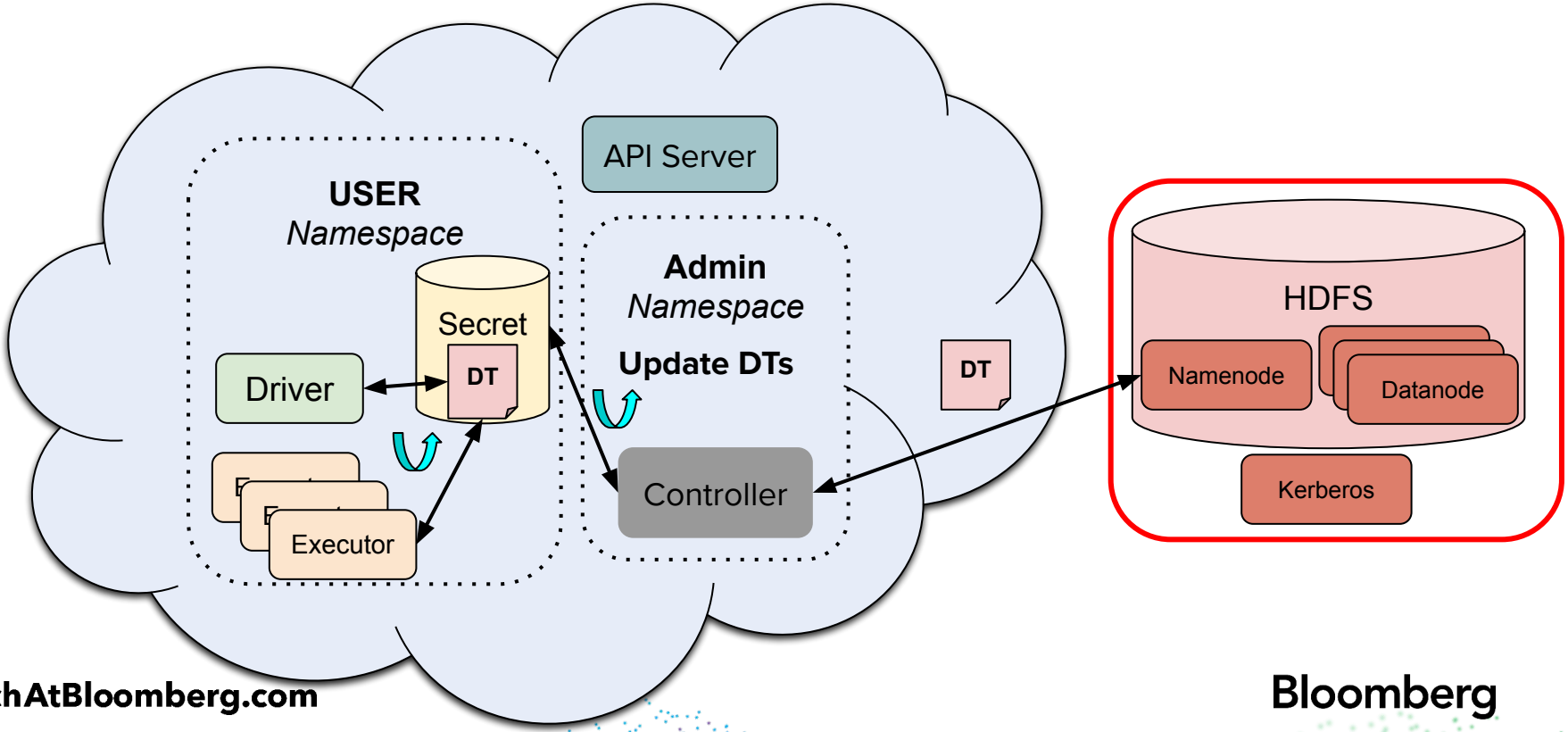


TechAtBloomberg.com

Bloomberg

Engineering

Secure Data Communication



TechAtBloomberg.com

Bloomberg

Engineering

Scaling Spark at Bloomberg with Disaggregated Compute

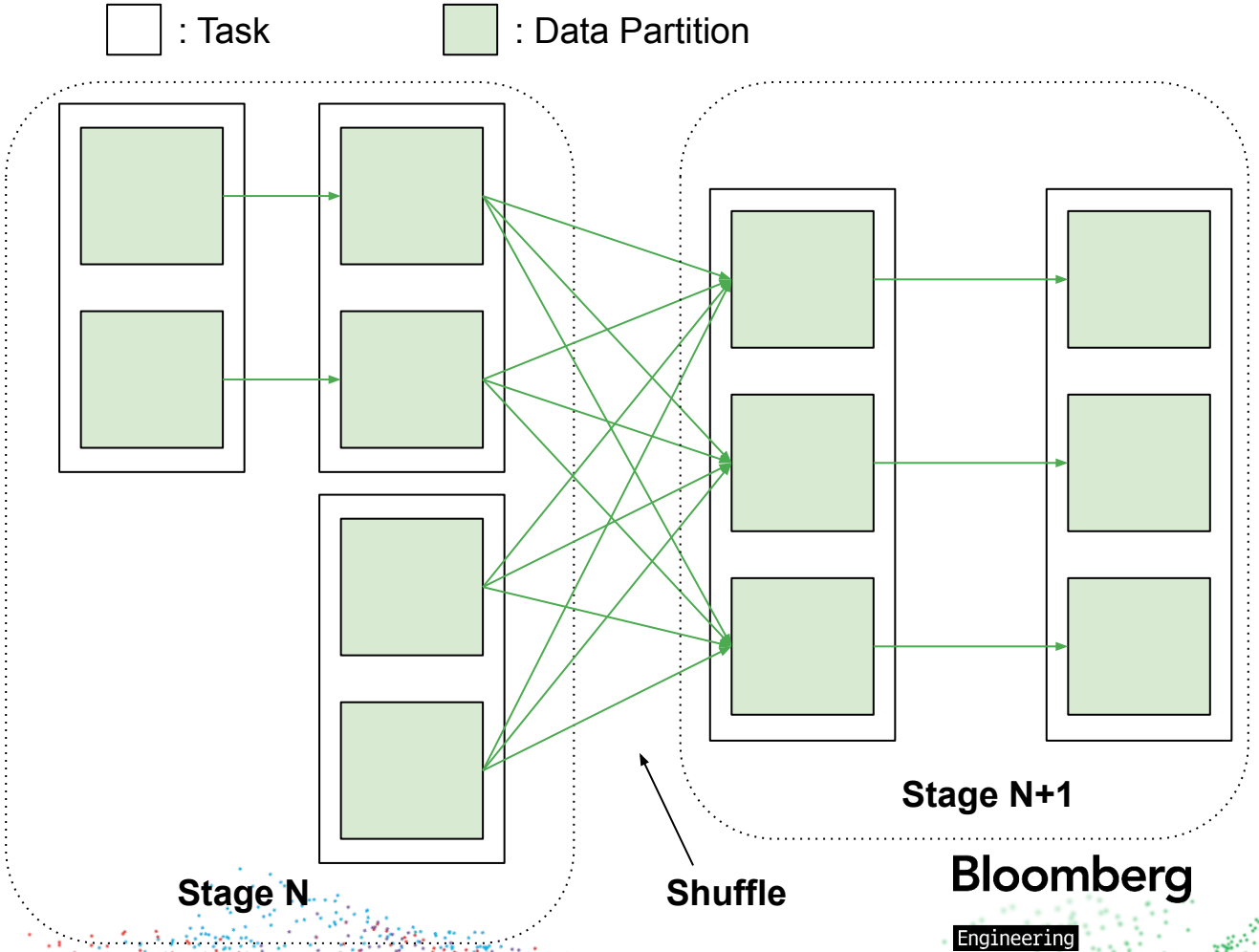
TechAtBloomberg.com

Bloomberg

Engineering

Spark Shuffle

- **Operator** Graph
- Spark submits Graph to **DAG** scheduler upon **Action**
- Operators have task stages
- **Stage** contains tasks based on data **partition**
- Stages passed to **Task Scheduler**
- **Shuffle** (all-to-all)
- **Dynamic Allocation**



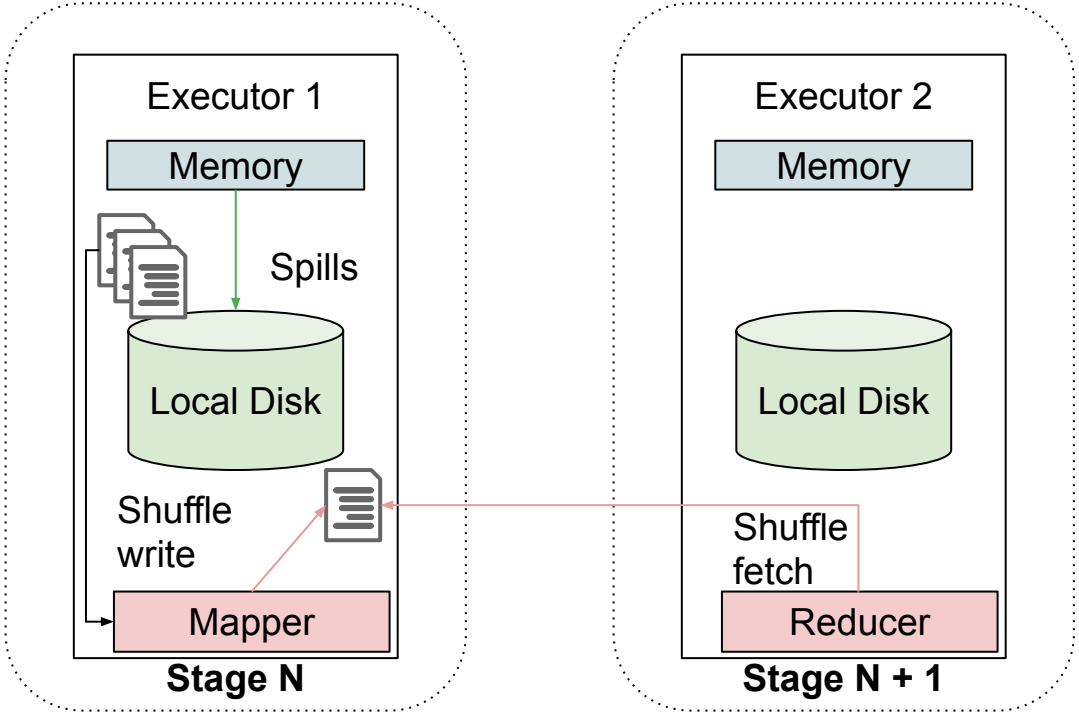
TechAtBloomberg.com

Bloomberg

Engineering

Spark Temporary Files

- **Spill Files**
  - Memory spills to disk as a File
- **Shuffle**
  - Output file used by later stages
  - If Spill files exist, they will be merged

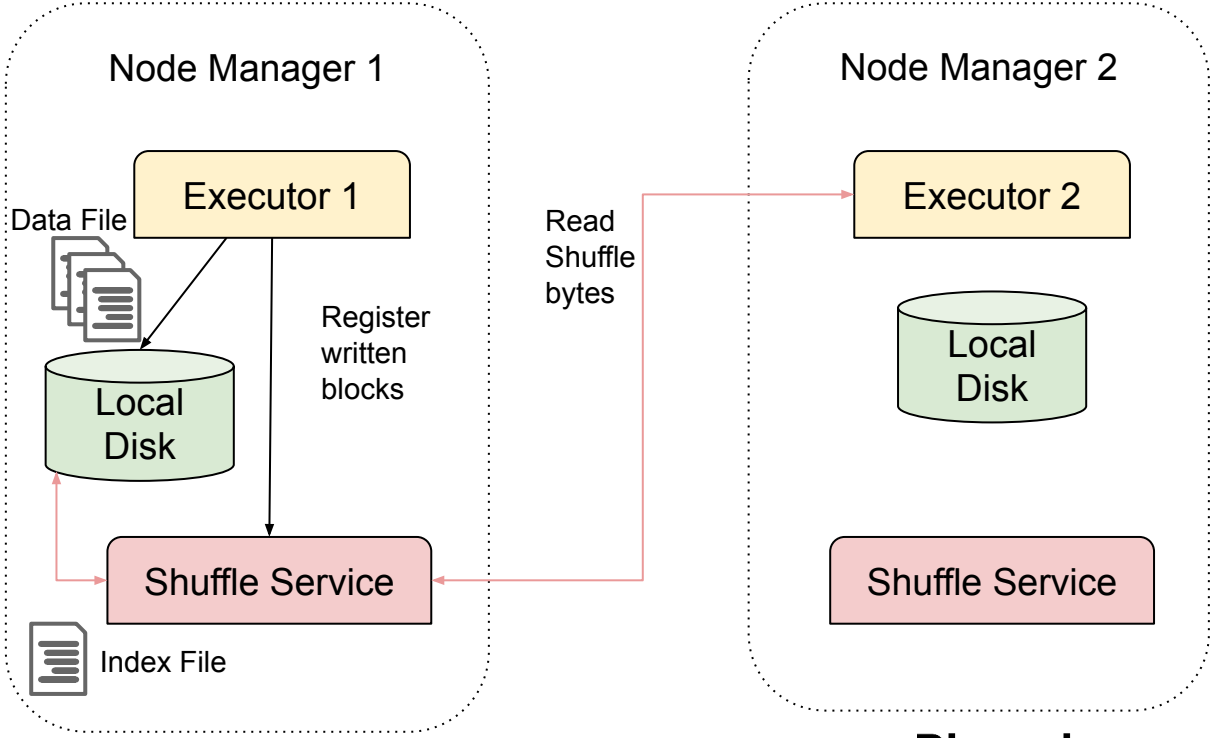


Current state of the External Shuffle Service

Shuffle Service
- Index files (seek $$)
- File Server

ESS
- Dynamic Allocation

Problems:
- Lack of isolation
  - Same host
- Lack of replication
  - Lineage $$
- Continuous Uptime
  - Failure means it's unschedulable
- What if in K8s?



TechAtBloomberg.com

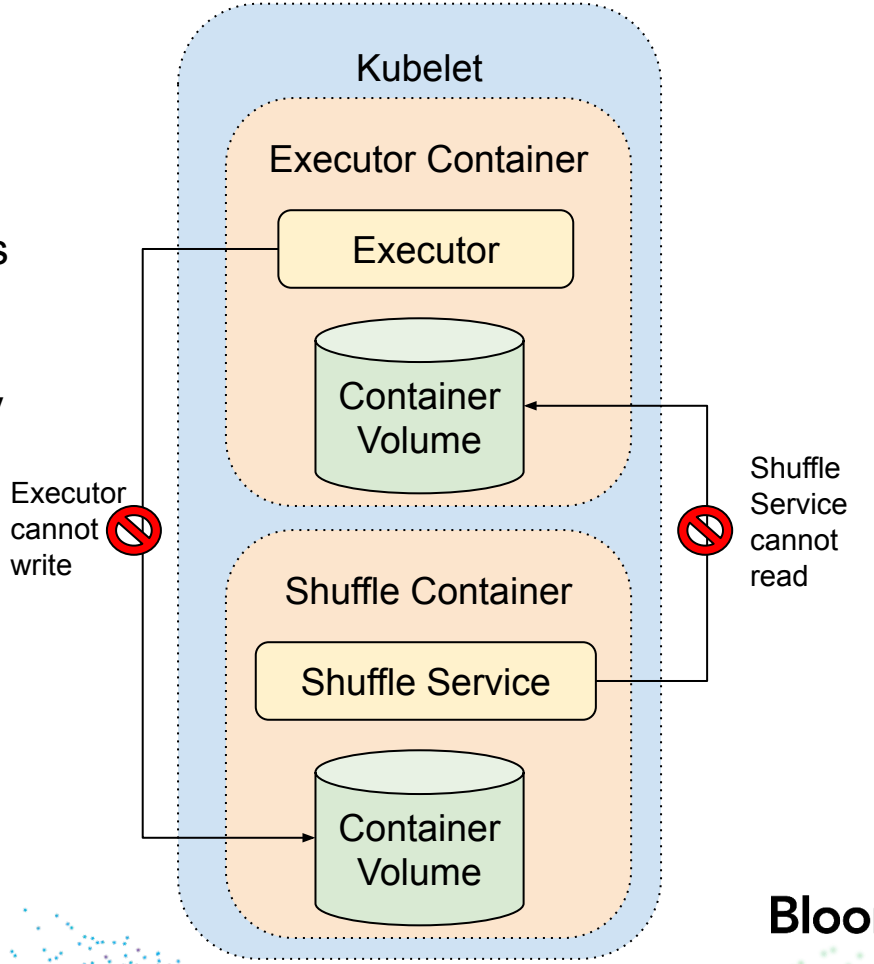
Bloomberg

Engineering

In Kubernetes

Current design breaks in Kubernetes

- Co-located storage is not always possible in containerized environments
- Isolation via cluster admin policy might make it impossible



Bloomberg

Engineering

API for Pluggable Remote Storage

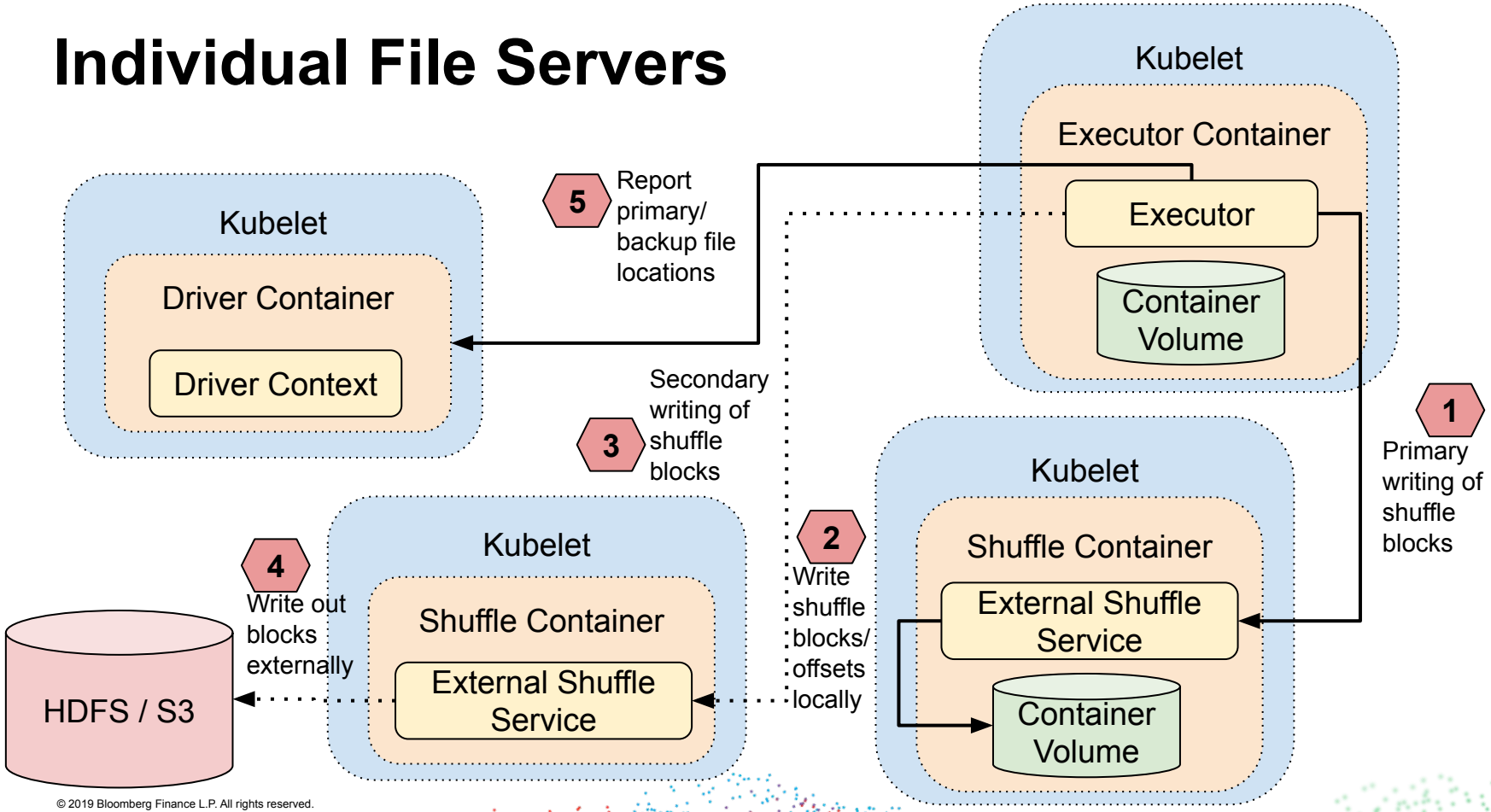
The solution Bloomberg envisioned is a several-month effort with developers from Palantir, Uber, Cloudera, LinkedIn, etc.:

An API within the current shuffle implementation for pluggable writing and reading of shuffle bytes

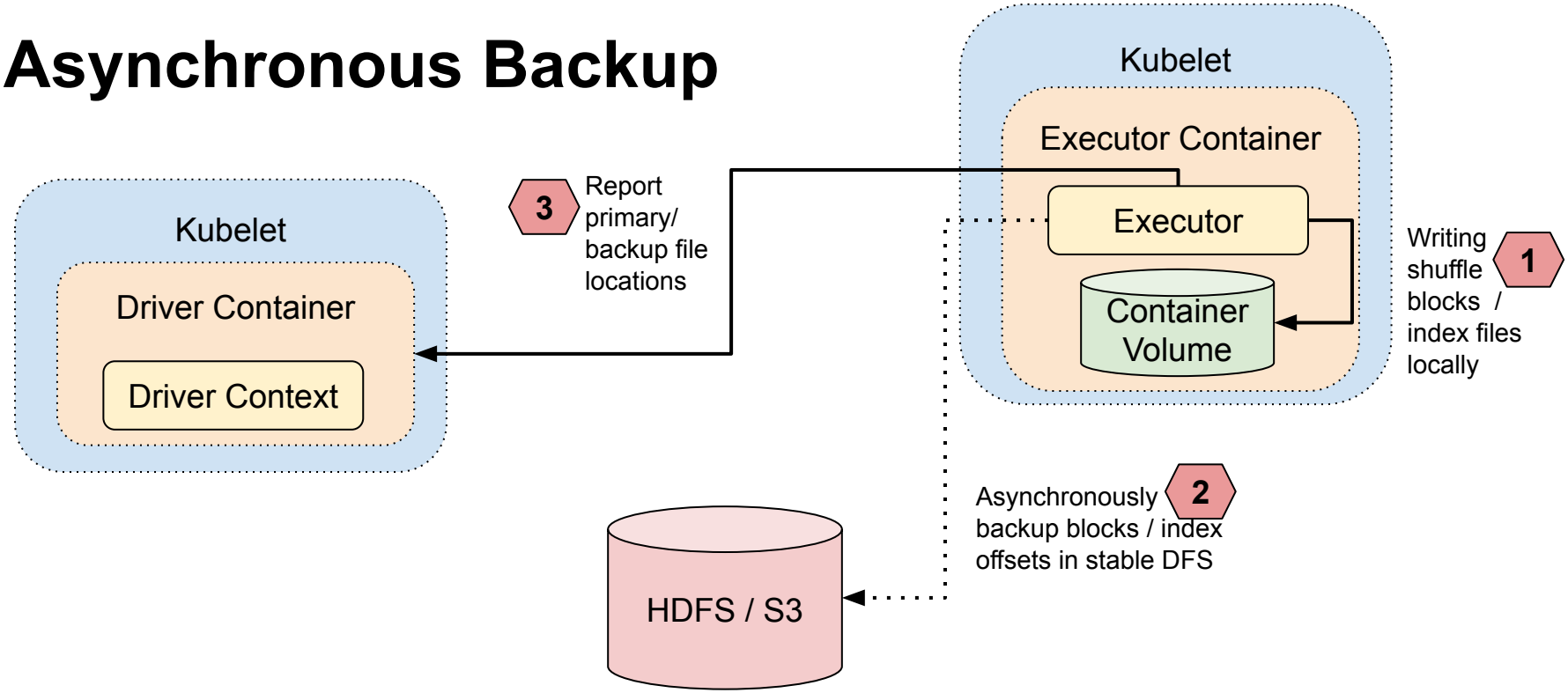
Plugin Tree:

- Driver Components
  - Lifecycle
- Executor Components
  - Lifecycle, Shuffle Writer, Shuffle Reader
- Shuffle Locations

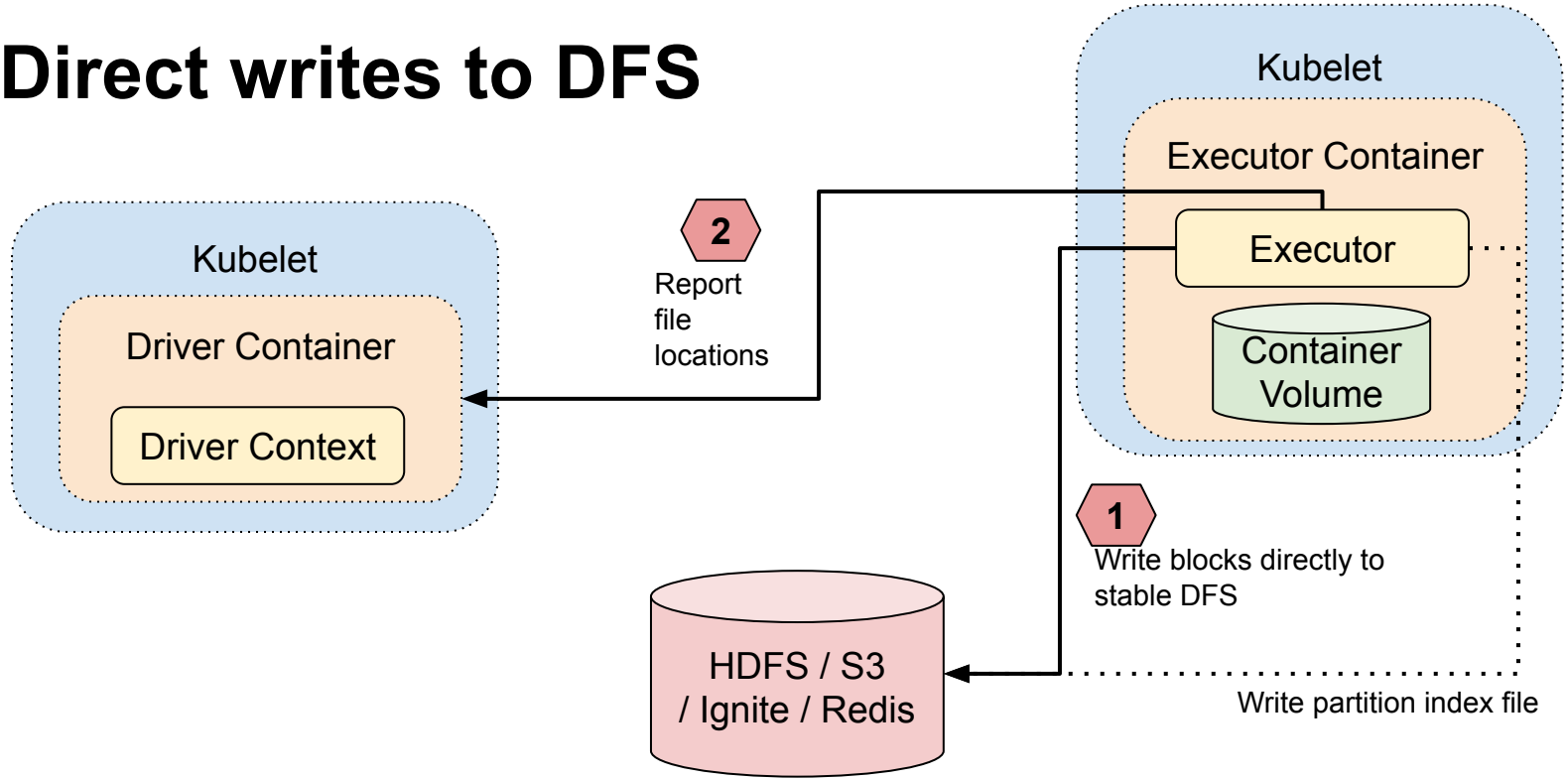
Individual File Servers



Asynchronous Backup



Direct writes to DFS



Future Work

TechAtBloomberg.com

Bloomberg

Engineering

Edge Cases and Further Extensions

There is an effort underway to merge this upstream as part of [SPARK-25299]

Future work includes:

- Exception Handling in the DAG Scheduler
- Extending separate APIs to handle Spills and Cache (Spark Temporary File)
- Performance benchmarking across different implementations

Thank you

Questions?

Bloomberg

Engineering

Kubecon Europe 2019
May 23, 2019

Ilan Filonenko, ifilonenko@bloomberg.net
Software Engineer, Data Science Infrastructure

TechAtBloomberg.com