
DATAFIBERS
Simplify
Big Data Streaming
Will Du
2017-09-30 @Toronto DataFibers Meetup Autumn

About Will

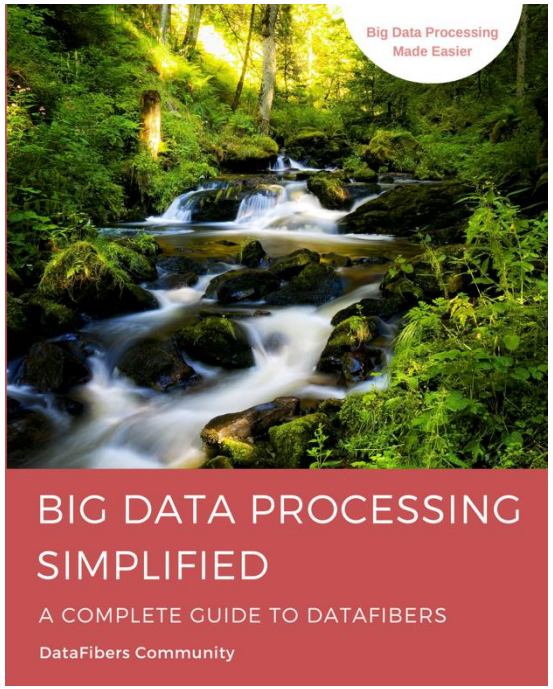
Will Du

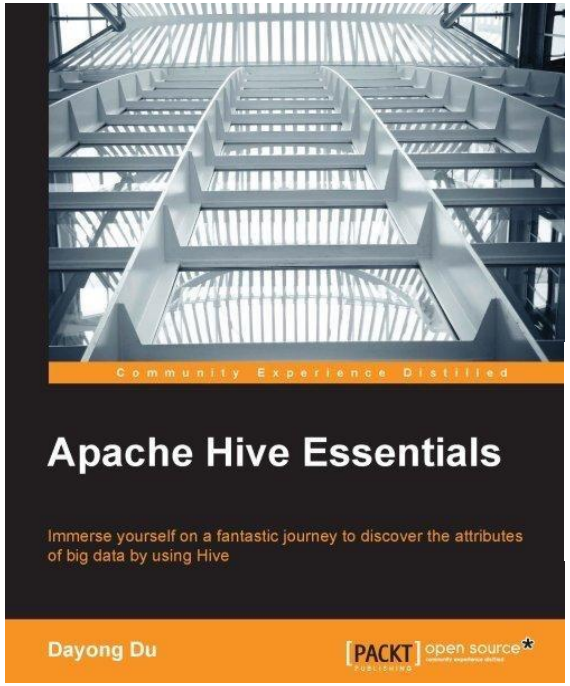
Big Data Practitioner | Author | Coach

TD • Dalhousie University

Toronto, Canada Area • 500+

- ✓ Big data since 2009
- ✓ Author of Apache Hive and Data Stream books
- ✓ Teacher of *Master Big Data Essential* @ IT21
- ✓ Bloggers of Sparkera
- ✓ Co-funder and advisors of few associations and start-ups





TBDPA

Sparkera

ignites big data imaginations

Agenda of Today

- Big Data Stream Processing Introduction
  - Concept
  - Use Case
  - Architecture Evolution
  - Ecosystem
- DataFibers Overview
  - Concept
  - Design
  - Status
- Demo

Again From 3Vs + 1Vs

2017 is the year focusing on data streaming

Velocity

Volume — Hadoop, Spark, HBase, …

Big Data

Text, Json, XML, COBOL, … — Variety

Value — TensorFlow, Caffe, H2O, …

Information Sources

CRM, SCM, ERP | Video | IT Ops | Email | Transactional Data | Mobile | Audio | Texts | Social Media | Search | Images

What is Stream Processing ?

Stream Processing is the handling of units of data on a **record-by-record** basis or over **sliding time windows**. This methodology **limits the amount** of data processed and offers a very contrasting way of looking at the data and the analytics. Complexity is traded in for **speed**. The nature of the analytics with stream processing are usually **filtering**, **correlations**, **sampling** and **aggregations** over the data.

Data Streaming Use Case

- Click streams and web analytics
- Location/GPS based marketing
- Messages in social networks (Twitter, Game, etc.)
- Music/video streaming (Spotify, Netflix, etc.)
- Financial data (stock changes, credit card tx)
- Traffic/Security control (networks, highways)
- Energy generation/consumption/optimization
- Sensor networks
- Weather monitoring
- Monitoring host intrusion







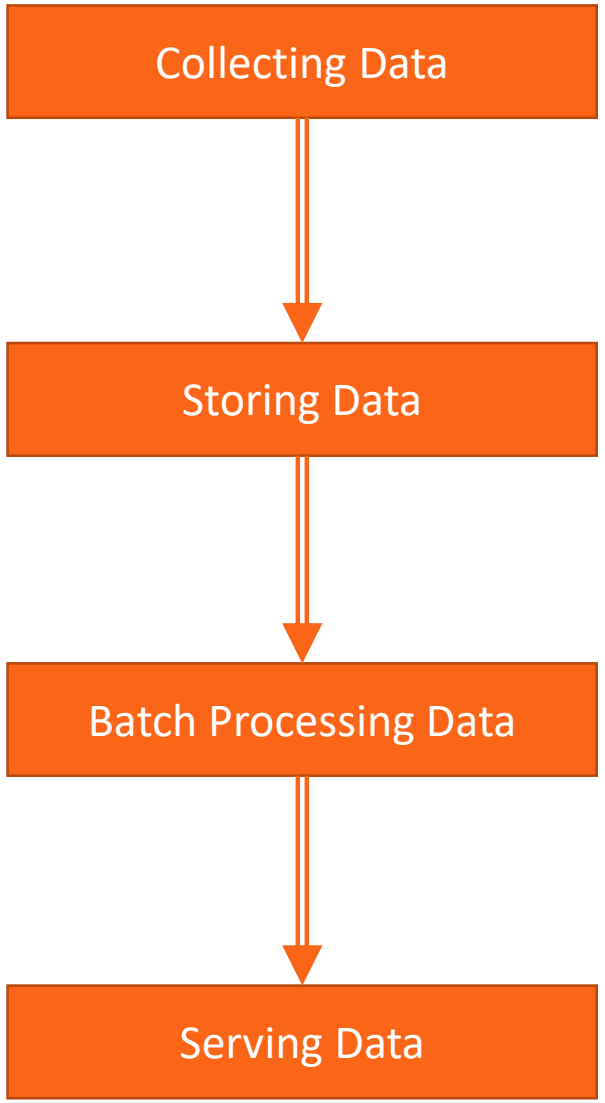
Anything right before your need

Tightly couple with data mining

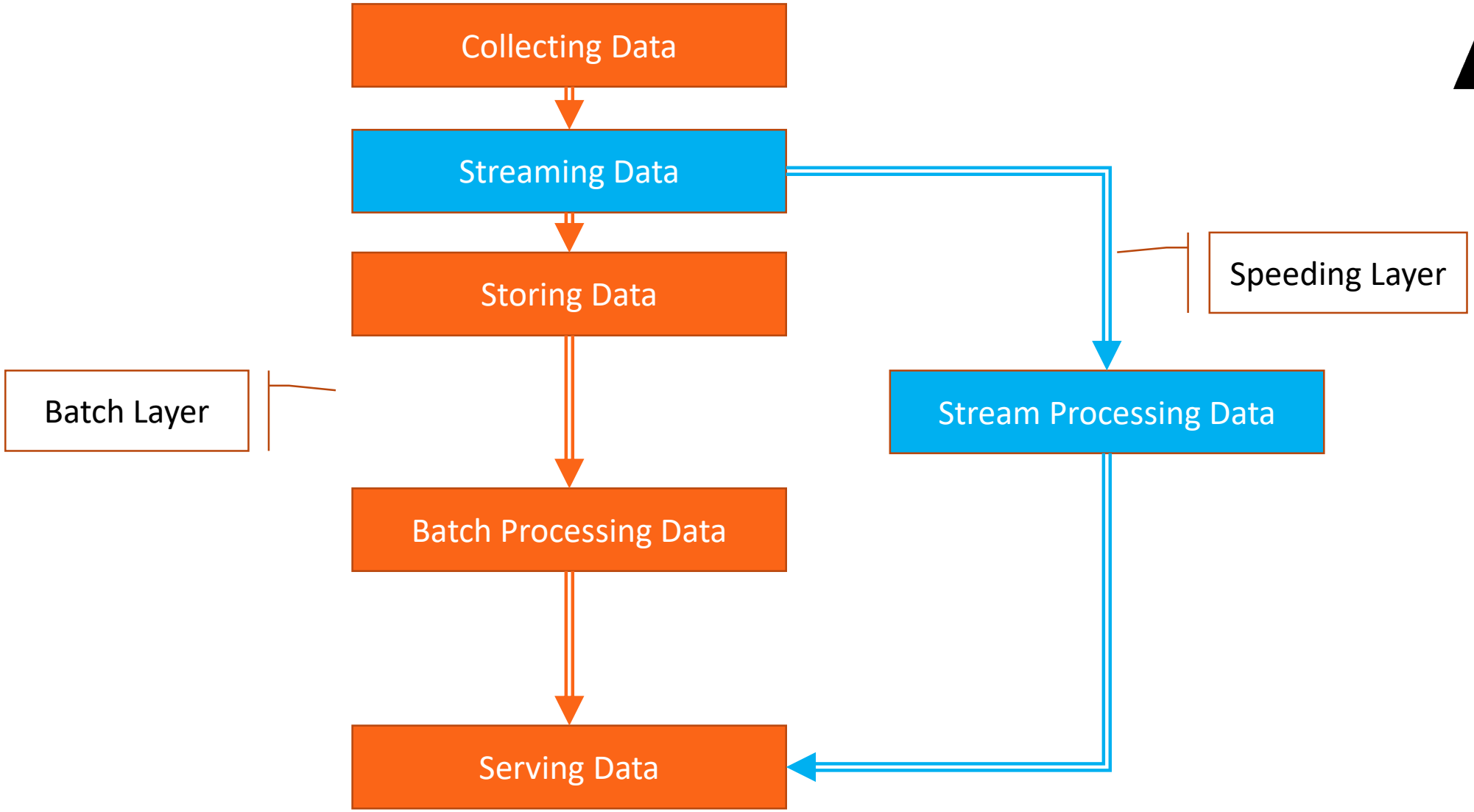
Batch vs. Stream

	Batch processing	Stream processing
Scope	Queries over all or most of the data in the dataset.	Queries or processing over data within a rolling time window, or on just the most recent data record.
Size	Large numbers of records	Individual records or micro batches consisting of a few records.
Performance	Latencies in minutes to hours.	Requires latency in the order of seconds or milliseconds.
Analyses	Complex analytics.	Simple response functions, aggregates, and rolling metrics.

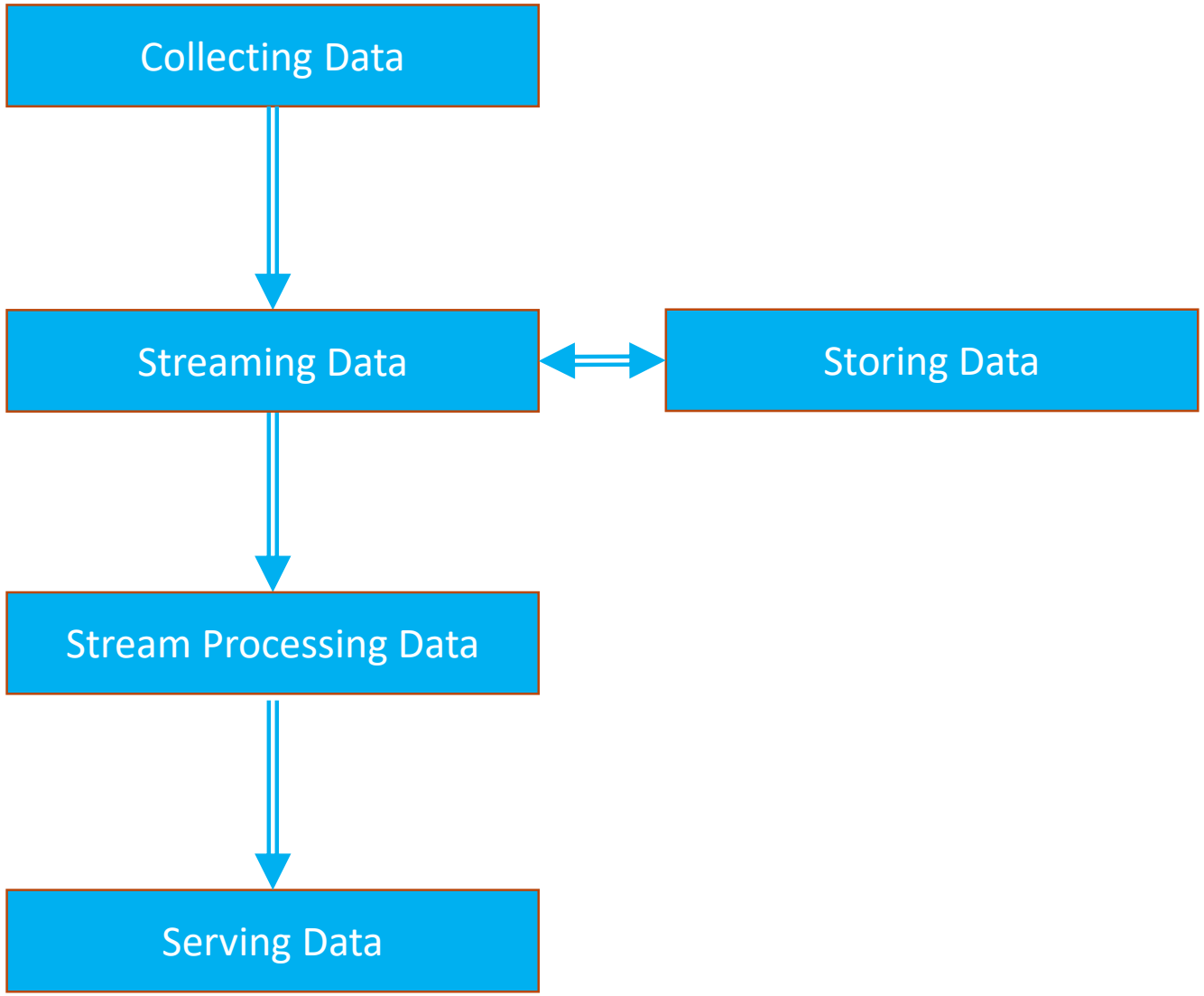
Big Data Processing Evolution – Big Data In Motion I



Big Data Processing Evolution – Big Data In Motion II



Big Data Processing Evolution – Big Data In Motion III

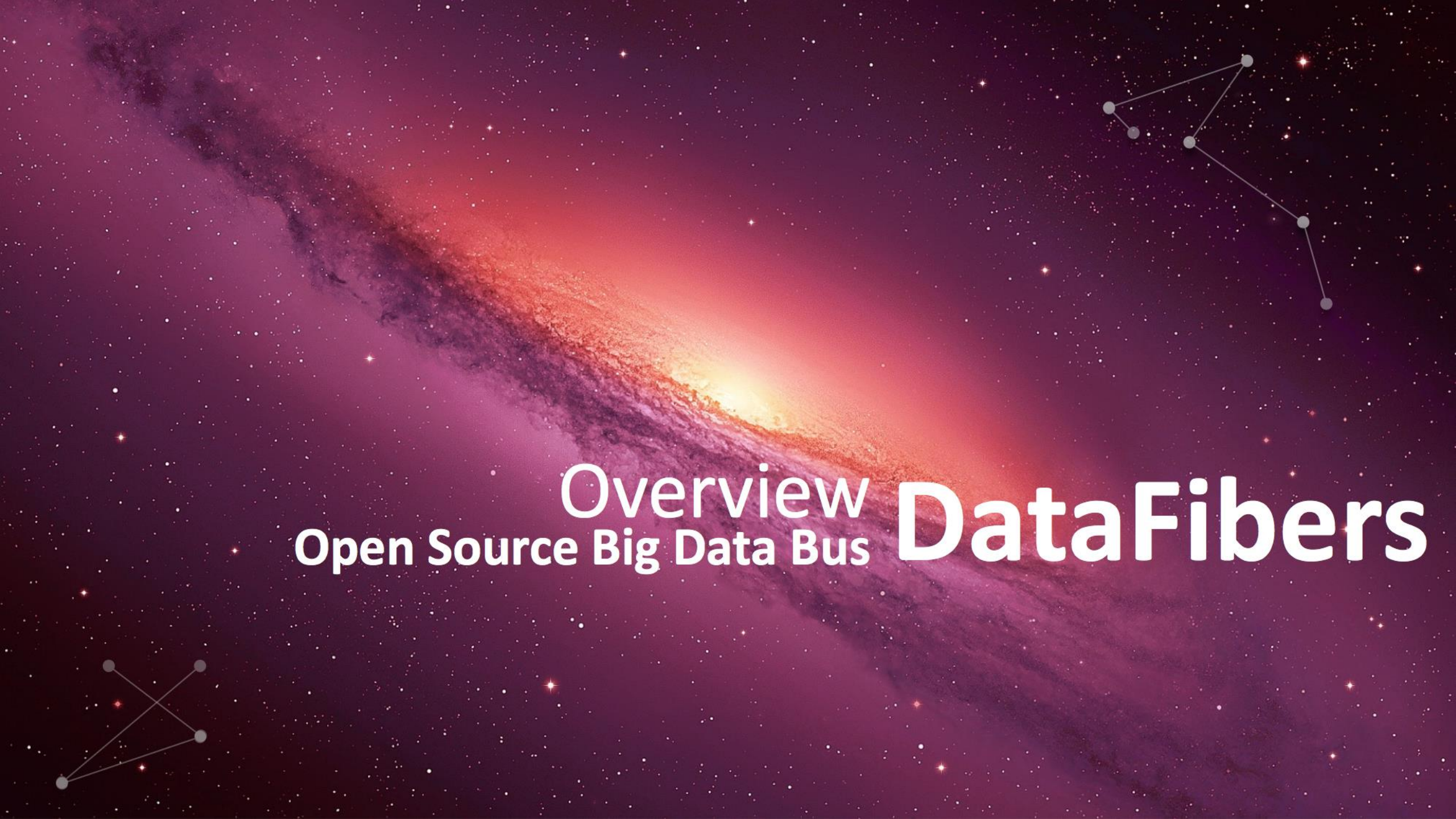


Stream Processing Ecosystem Comparison

	Flume	NiFi	Gearpump	Apex	Kafka Streams	Spark Streaming	Storm	Storm + Trident	Samza	Flink	Ignite Streaming	Beam (*GC DataFlow)
Current version	1.7.0	1.1.1	incubating	3.5.0	0.10.1.1	2.1.0	1.0.2	1.0.2	0.11.0	1.2.0	1.8.0	0.4.0
Category	DC/SEP	DC/SEP	SEP	DC/ESP	ESP	ESP	ESP/CEP	ESP/CEP	ESP	ESP/CEP	ESP/CEP	SDK
Event size	single	single	single	single	single	micro-batch	single	mini-batch	single	single	single	single
Available since (incubator since)	June 2012 (June 2011)	July 2015 (Nov 2014)	(Mar 2016)	Apr 2016 (Aug 2015)	May 2016 (July 2011)	Feb 2014 (2013)	Sep 2014 (Sep 2013)	Sep 2014 (Sep 2013)	Jan 2014 (July 2013)	Dec 2014 (Mar 2014)	Sep 2015 (Oct 2014)	Jan 2017 (Feb 2016)
Main backers	Apple Cloudera	Hortonworks	Intel Lightbend	Data Torrent	Confluent	AMPLab Databricks	Backtype Twitter	Backtype Twitter	LinkedIn	dataArtisans	GridGain	Google
Delivery guarantees	at least once	at least once	exactly once at least once (with non-fault-tolerant sources)	exactly once	at least once	exactly once at least once (with non-fault-tolerant sources)	at least once	exactly once	at least once	exactly once	at least once	exactly once*
State management	transactional updates	local and distributed snapshots	checkpoints	checkpoints	local and distributed snapshots	checkpoints	record acknowledgements	record acknowledgements	local snapshots distributed snapshots (fault-tolerant)	distributed snapshots	checkpoints	transactional updates*
Fault tolerance	yes (with file channel only)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes*
Out-of-order processing	no	no	yes	no	yes	yes	yes	yes	yes (but not within a single partition)	yes	yes	yes*
Event prioritization	no	yes	programmable	programmable	programmable	programmable	programmable	programmable	yes	programmable	programmable	programmable
Windowing	no	no	time-based	time-based	time-based	time-based	time-based count-based	time-based count-based	time-based	time-based count-based	time-based count-based	time-based
Back-pressure	no	yes	yes	yes	N/A	yes	yes	yes	yes	yes	yes	yes*
Primary abstraction	Event	FlowFile	Message	Tuple	KafkaStream	DStream	Tuple	TridentTuple	Message	DataStream	IgniteDataStreamer	PCollection
Data flow	agent	flow (process group)	streaming application	streaming application	process topology	application	topology	topology	job	streaming dataflow	job	pipeline
Resource management	native	native	YARN	YARN	Any process manager (e.g. YARN, Mesos, Chef, Puppet, Salt, Kubernetes, ...)	YARN Mesos	YARN Mesos	YARN Mesos	YARN	YARN Mesos	YARN Mesos	integrated*
Auto-scaling	no	no	no	yes	yes	yes	no	no	no	no	no	yes*
In-flight modifications	no	yes	yes	yes	yes	no	yes (for resources)	yes (for resources)	no	no	no	no
API	declarative	compositional	declarative	declarative	declarative	declarative	compositional	compositional	compositional	declarative	declarative	declarative
Primarily written in	Java	Java	Scala	Java	Java	Scala	Clojure	Java	Scala	Java	Java	Java
API languages	text files Java	REST (GUI)	Scala Java	Java	Java	Scala Java Python	Scala Java Clojure Python Ruby	Java Python Scala	Java	Java Scala Python	Java .NET C++	Java Python
Notable users	Meebo Sharethrough SimpleGeo	Macquarie Telecom	Intel Levi's Honeywell	Capital One GE Predix PubMatic	N/A	Kelkoo Localytics AsiaInfo Opentable Faimdata Guavus	Yahoo! Spotify Groupon Flipboard The Weather Channel Alibaba Baidu Yelp WebMD	Klout GumGum CrowdFlower	LinkedIn Netflix Intuit Uber	Alibaba Bouygues Ericsson King Otto Group Zalando	GridGain	N/A

What's Happening in Ecosystem For Streaming

- SEP, 2017 – Pinterest runs over 1000 Kafka brokers.
- AUG, 2017 – Apache Beam released 2.1.0 with new APIs for AmqIO, CassandraIO, & HCatalogIO
- AUG, 2017 – Apache Flink release 1.3.2 for bug fix
- JUL, 2017 – Databricks Spark 2.2 structure streaming is production ready
- JUN, 2017 – HDP has released the new Streaming Analytics Manage to make stream development easier
- JUN, 2017 – Kafka 0.11.0.0. is published with new features include exactly-once semantics

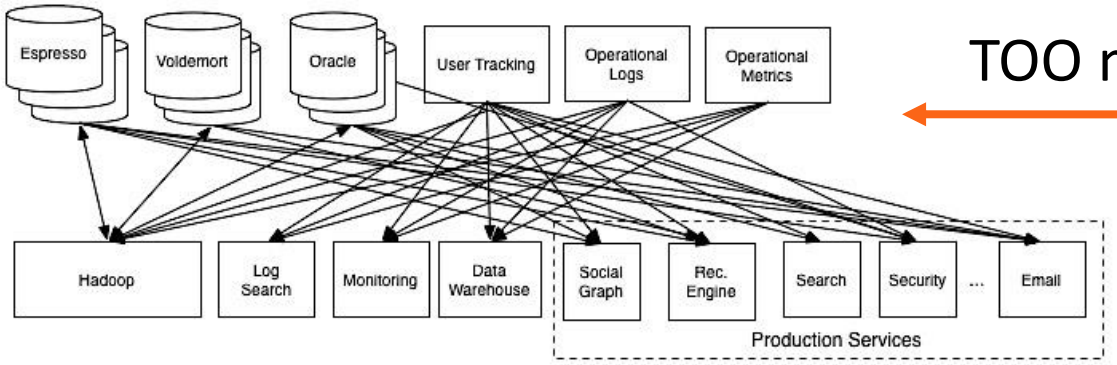

Overview
Open Source Big Data Bus
DataFibers

What is DataFibers?

- **DataFibers** – short as Fibers, a open source big data processing framework.

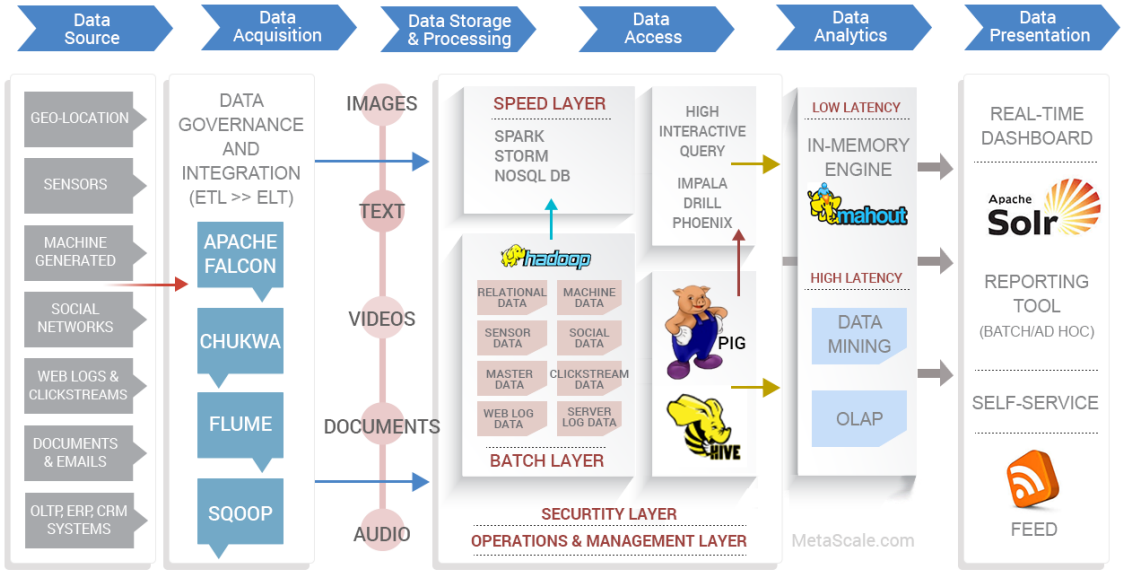
- DataFibers simplifies the roadmap for enterprise to process big data no matter in batch or stream.

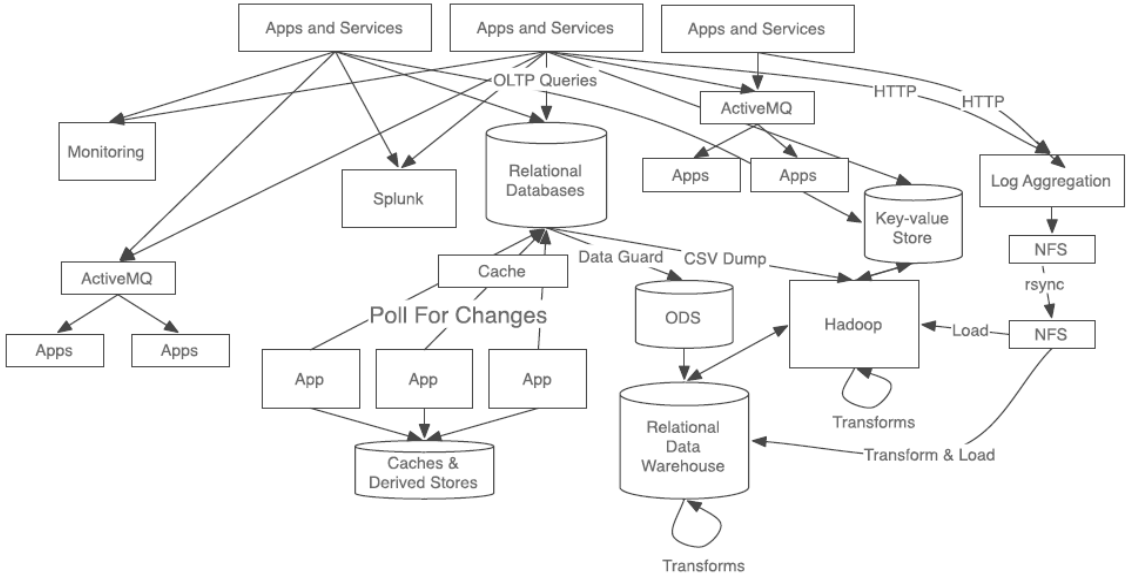
The Pain Before Data Bus



TOO much pipes & work

TOO complex

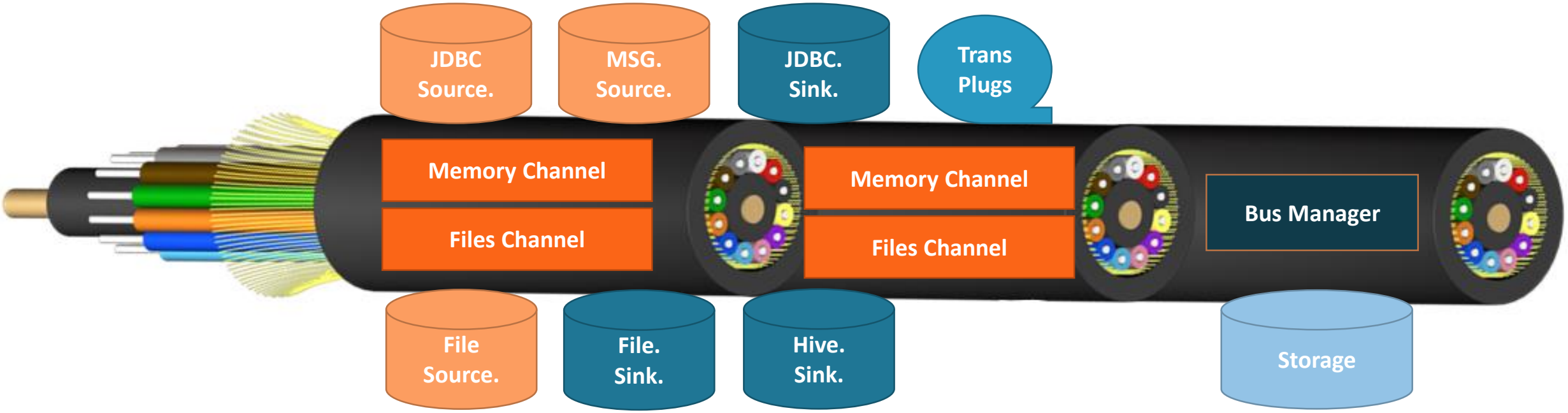




TOO many steps & layers, and too slow

DataFibers Logic Architecture

- Publisher, Subscriber, Transformer,
- Memory Channel, File Channel
- Meta and Data Storage



DataFibers Technical Stacks

- Apache Hadoop, Alluxio (Tachyon), Kudu
- Spark, Flink, Beam
- Hive/HBase Warehouse
- Kafka, Vertx
- MongoDB, HBase, PostgreSQL, Elastic
- Java, Shell, JavaScript, React/Angular, REST
- Vagrant, Dockers
- Git

DataFibers Features

- Dynamic and transparent routing data processing
- Transformation between jobs
- Simple and powerful (especially on transformation) API
- Data subscribe, pull, and push
- Topic exploration, management, and subscription
- Support batch, stream, and hybrid data processing
- Data cache, replay, reprocess
- Messaging metadata for data discovery and optimized access
- Inter-bus connector, bus-hub and data market place

DATAFIBERS

One Day At Work

One day, Tom is at his boss's office

Tom, we are going to start our big data project. I recommended you to the CEO that you will be a good leader for this. We'll start a POC of this from this week.

Here is the vision and plan.

What ?? – A POC of Big Data

- Import our existing stream stock information to the POC
- Stream the data to a MongoDB so that existing web service can use it
- All data should be sent to HDFS/Hive in terms of archive and batch/stream process
- The data stream should also be available in our dashboard template for real time monitoring and reporting (with backend such as MySQL)
- A GUI for stream task management and monitor
- A flexible design to stream data to more other places or system
- Different LOB will need full/partial data isolation, subscription, publication, etc.

Future consideration

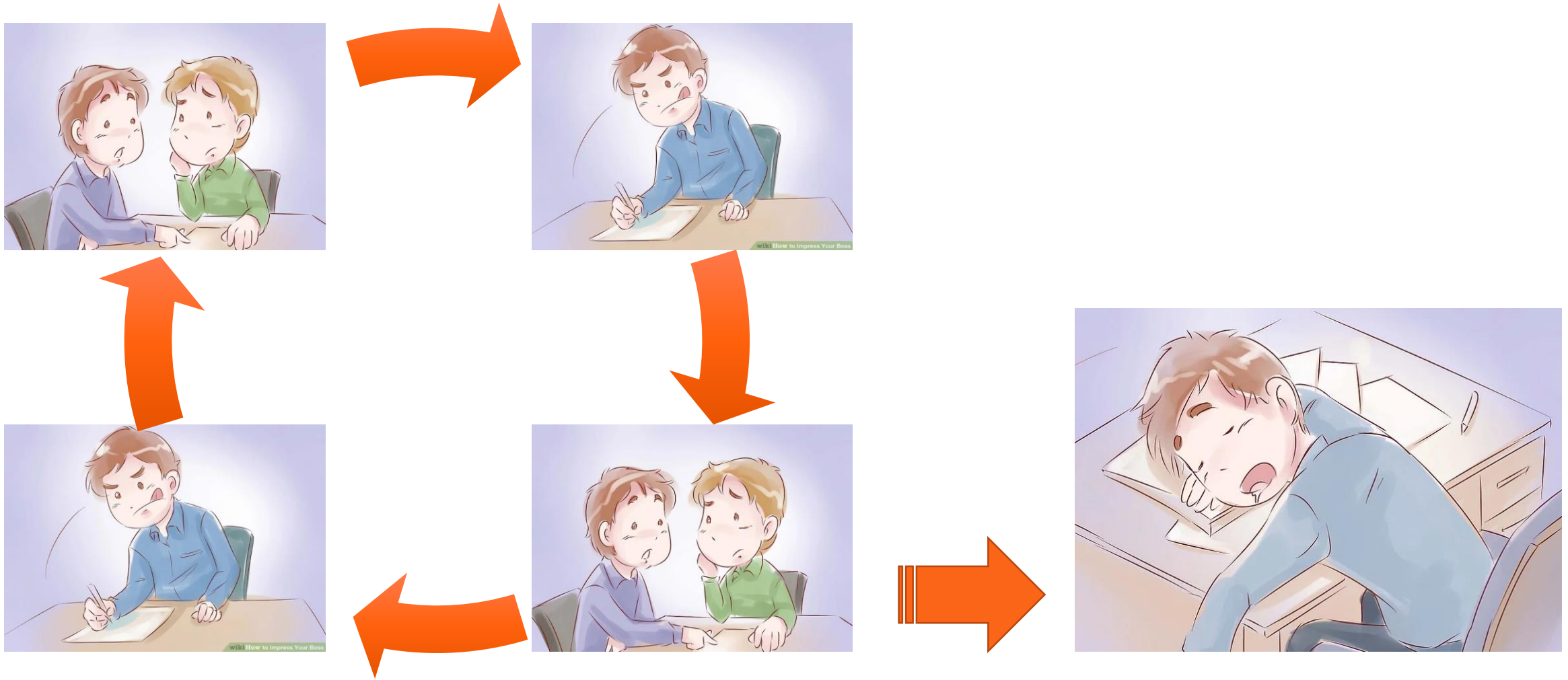
- a light weight ETL on the stream
- ML online/offline with streaming data
- Stream & Batch processing

What's More – since it is ONLY a POC

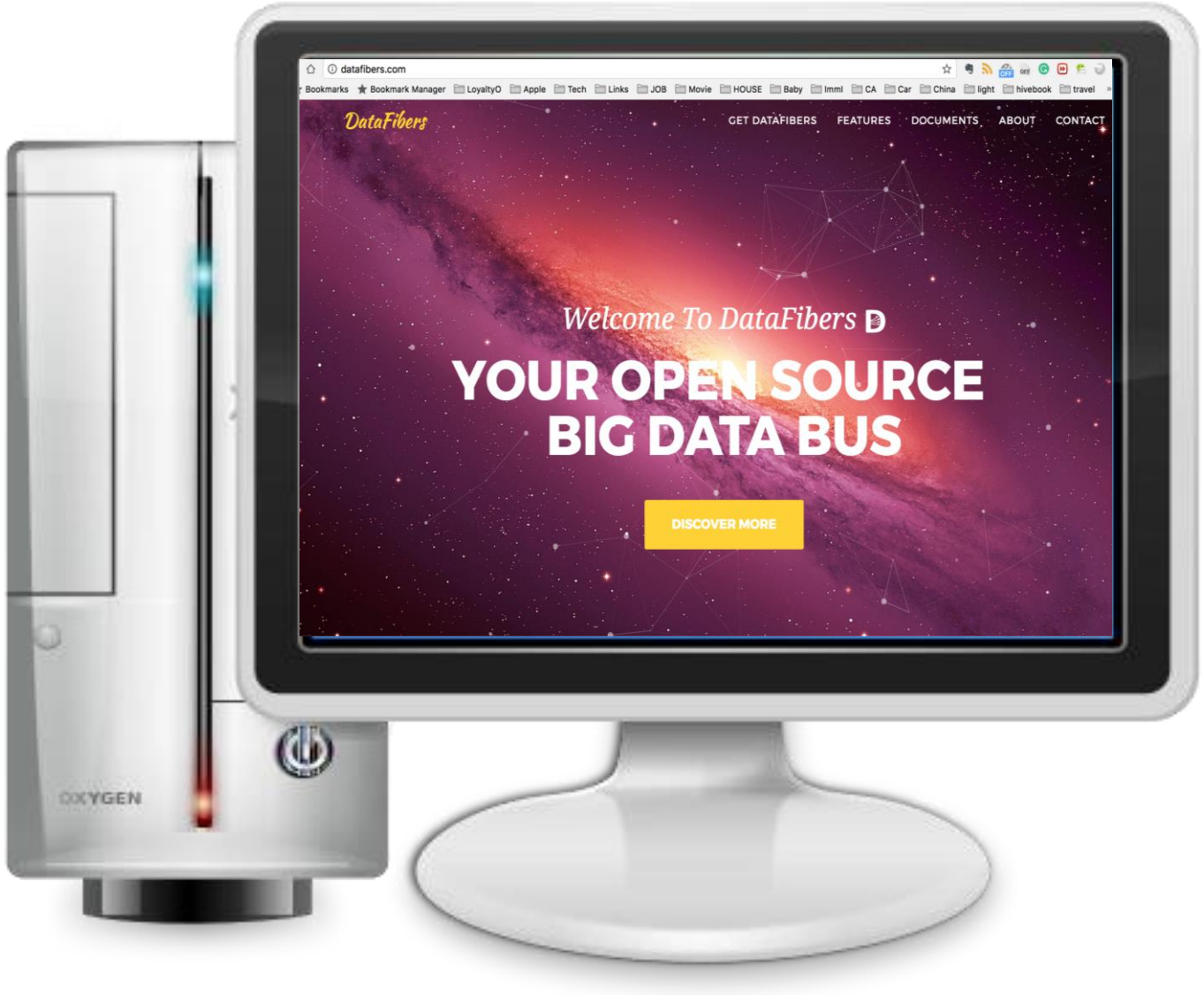
We only have
***TWO** weeks*

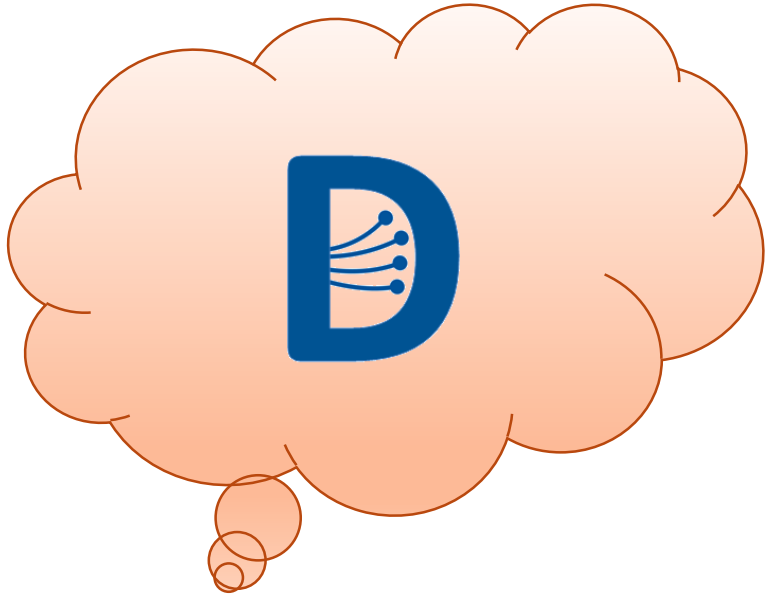
No Budget
For More Help

Then What's Next – 10 days



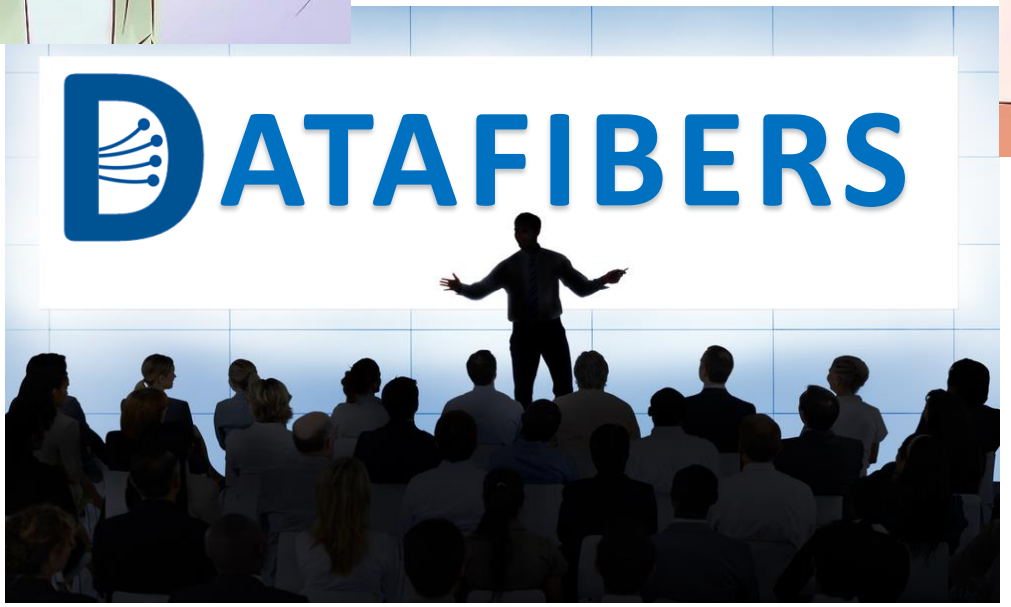
Suddenly – Remember What's Heard Recently





DATAFIBERS

As Result



Typical DataFibers User Case...

Dot end lines are either batch or real-time stream

Internal Discovery Portal

BI Dashboard Portal

Data Auditing

Other DataFibers Instance

Key-Value Store

Memory Cache

3rd Party Feeds

transform

Iterative Training

Lookup

Enterprise Big Data Bus - DataFibers

NA Data Hub

NA MDM

NA CRM

NA EDW

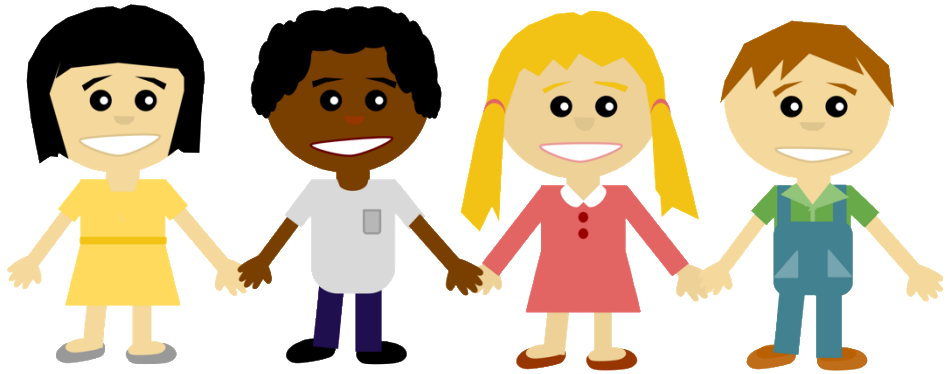
Asia Data Hub

EU Data Hub

What We Offers …

- An opportunity to make things big and different
- Experience the cutting edge of big data technology
- An chance to work with top-notch big data professionals
- Open source communities
- Start-up opportunities
- Advices, help, reference, friendship, etc.

Welcome to Join Us



How to Participant

- Get to know about the project at datafibers.org | datafibers.com
- Join our discussion and ask questions at datafibers@googlegroups.com
- Watch and Star us in GitHub@datafibers-community
- Fork and Pull Request when you have ideas to contribute
- Contact Us for participant at datafibers@gmail.com
- Hear our news and events @wechat or github

DATAFIBERS

Thank You

Come and Join Us