CASK

Big Data On Tap

Collection of Technical Questions about CDAP

Last Updated : March 5th 2017

Created On : January 16th 2012

CASK

Architecture

High Level Architecture View

Unified Integration Platform - Cask Data Application Platform (CDAP)

Apache 2.0

- Get started faster with Hadoop Initiatives
- Solve skills gap
- Faster adoption of Hadoop through self-service

- Integrate existing solutions
- Build custom solutions faster
- Reduce all boilerplate code

- Future proof your investment and effort
- Portability
- Seamless transitions

Cask Data Application Platform (CDAP)

AGNOSTIC LAYER

Integrated Services
Out-of-box value-add self-service apps for data preparation , data ingestion, data pipelining, data discovery and Metadata

APIs
Programmatic and REST APIs for building and managing applications and datasets

Reusable Components
Collection of reusable components distributable through Cask Public Market or Company Private Market

DISTRO NATIVE LAYER

Data and Application Capabilities
Additional capabilities lacking in the underlying infrastructure to seamlessly manage your valuable data and applications - Orchestration, Versioning, Transaction, Messaging, etc

Infrastructure Integration Services
Integration with varied and complex infrastructure components - HDFS, Kafka, HBase, Zookeeper, etc

Operational Capabilities
Data and Application Operation simplification - Security through Authentication, Authorization.

Support for multi-tenancy for multiple environment on single cluster

Ops Automation, Metrics and Logs aggregation

Replication, Zero-downtime for Application & Data(*)

Hadoop Distribution
Hadoop distribution running on On-Premise, Private Cloud, Public Cloud and Hybrid Environment

- Bootstrap your initiatives with hardened and ready to go solutions
- Break silos and share with other organizations

- Reduce operations cost
- Better capacity planning
- Standard Ops Run book
- Run production applications

Business Benefits

1. 80% Reduction In Code
2. 5x Faster to Market
3. Hybrid Cloud Readiness

CASK DATA APPLICATION PLATFORM

HIGH LEVEL ARCHITECTURE

Apache 2.0

External Systems

Source Systems (Apache NiFi, Streamsets, etc

Streaming

3rd Party Apps

OLTP / ODS

Databases

Flat Files

Cloud Sources

CDAP Edge*

BI & Viz Tools | Ad-hoc Tools | Data Science | Micro-Services

Custom Applications | Custom Plugins | Application Templates

Cask Data Application Platform (CDAP)

Integrated Services

Data Preparation | Data Ingestion | Data Pipeline | Data Discovery | Data Catalog | Data Services

Market

- Packaged Use-cases
- Datapacks
- Pre-built Applications
- Plugins
- Drivers

API

Programmatic API

Application Program Dataset

- Streaming
- Tigon
- Workflow
- Service
- Spark
- MapReduce
- Worker

- Table
- Fileset
- Object Table
- OLAP Cube
- Key-value Table
- Partitioned File

Artifact

Plugin

Application Template

REST APIs

Tools

Testing Framework

SDK - CDAP Standalone

Integration Testing Framework

Maven Archetype

Data and Application Capabilities

Audit Log | Version Management | Partition Management | Workflow Orchestration

Scheduling | Artifact Management | Notification Service | ACID Transactions

Transaction Messaging | Dataset Management | Streams | SQL backed

Operational Capabilities

Multi-Tenancy Namespace | Replication

Service Registry | Upgrade Tools

Log & Metrics | REST API

System Services

App Lifecycle Management | Data Lifecycle Service | Stream Service | Messaging Service | Metadata Service | Log & Metrics Service | Transaction Service

Security and Governance

Kerberos | Secured Store

Authorization | Apache Sentry

Authentication | Apache Ranger

Navigator | Business Tags

Lineage | Data Dictionary

AGNOSTIC LAYER

ADAPTABLE LAYER

Cloudera Manager

HDP Ambari

Cloudera Navigator

Apache Sentry Apache Ranger

KMS

CASK

CDAP 4 Supported Hadoop Distribution & Cloud

HDP	CDH	MapR	Azure	IBM IOP	BigTop	AWS - EMR
2.0	5.0	4.0	3.4	4.1	0.8	4.x
2.1	5.1	5.0	3.5	4.2	1.0	
2.2	5.2	5.1			1.1	
2.3	5.3	5.2				
2.4	5.4					
2.5	5.5					
	5.6					
	5.7					
	5.8					
	5.9					
	5.10					

1. Abstractions & Code

Especially for Spark

1.1 — Code Abstraction Method

Describe your approach and any standards that you follow ?

- CDAP exposes **three major abstractions** – Dataset, Program and Application
- **Application** is container specification for data processing, serving and data storage on big data system
- **Program** defines data processing and data service elements of your data solution
- **Dataset** are programmatic abstraction over one or more underlying data storage systems (analogous to Hibernate for big data systems).
- **Plugin** is a implementation of a application defined interface to reconfigure application behavior. Application can define one or more interfaces for plugins.
- **Artifact** is a tangible by-product produced during SDLC for **Application** and **Plugin** – represented as JAR file
- **Application Template** is a re-configurable blue-print of Application through **Plugins**. (analogous to Java Generics for data applications)

Above abstractions bind to infrastructure specific implementations through CDAP injection framework at runtime

CDAP Standalone — YARN Container become Threads, HDFS become Local Filesystem, HBase becomes LevelDB, etc

1.2 — Code Abstraction Method

How does this impact staying up-to-date with Apache releases ? Distribution based or Apache Release based ? Time lag for each supported distribution or related to Apache standard

- Three pronged approach
  - **Apache Release** up-to-date
  - **Distribution** up-to-date
  - **Technology** up-to-date
- OSS contribution and committers for major apache projects – Apache Flink or Apache Beam (Apache Release), multiple contributions to Apache Spark
- Partnership and release candidate exchange – regular on-going testing through out the year.
- Experimental integrations with Apache Flink and Presto as example ahead of distribution inclusion.

- **Cloudera Distribution (CDH)**
  - Pre-Release CDH Integration – CDH 5.10 has back ported some of HIVE 2.0 features, CDAP latest already works on CDAP 5.10
  - CDH 5.10 released on Jan 31st 2017.
  - Cloudera Test environment runs CDAP Integration Tests on daily basis
  - CDAP nightly ITN across all versions of CDH
  - Worst-case lag time of 3 weeks (based on last years release schedules)

CDAP 4 Supported Hadoop Distribution & Cloud

HDP	CDH	MapR	Azure	IBM IOP	BigTop	AWS - EMR
2.0	5.0	4.0	3.4	4.1	0.8	4.x
2.1	5.1	5.0	3.5	4.2	1.0	
2.2	5.2	5.1			1.1	
2.3	5.3	5.2				
2.4	5.4					
2.5	5.5					
	5.6					
	5.7					
	5.8					
	5.9					
	5.10					

1.3 — Code Abstraction Method

Do you directly call components (especially spark components) or do you leverage an Adaptive Execution Layer ? If you use an adaptive execution layer, please explain how you interact with the analytic engine (like Spark - API) ?

- CDAP Applications currently call Spark APIs directly
  - Minimal impact in terms of performance
  - Pipeline Application Template includes **Planner** (not adaptive execution layer) to layout the execution plan of Plugins and Workflow at deployment time and **NOT at runtime**
- Plan to integrate with Spark Adaptive Execution Layer when available

1.4 — Code Abstraction Method

How much functionality is covered by the abstractions — especially to develop Spark Analytics ?

- **All Spark capabilities are available** to developer.
- **Spark ML and other such libraries** can be used for analytics.
- **CDAP doesn't hide** any Spark functionalities
- Only If required by business – CDAP supports **higher order compositions of Spark functionalities** that can expose only the functionalities required for analytic functions
  - Plugins of Application Template could **choose NOT to expose** Spark to developers
  - E.g. Aggregate Plugin Interface (simple aggregation API) **will be simple interface** for performing aggregations

1.5 — Code Abstraction Method

How much functionality is covered with the API calls ?

- With CDAP Abstraction over Spark, **you don't lose any functionality**
- New business critical capabilities and functionalities are introduced
  - Exactly once-semantics,
  - ACID properties - for data consistency,
  - Integration with Reliable Messaging and CDAP Service discovery from within Spark Program
  - Audit Logs &
  - Metadata

1.6 — Exporting Code

Code exported via API? And in what language (XML)? Specify range of capabilities.

Custom Applications and Plugins

- **Custom Applications and Plugins** of CDAP are packaged as JAR file along with JSON configuration - similar to how web application are packaged.
- The package can contain other language scripts, but JAR is the unit of delivery to CDAP system
- CDAP provides REST APIs / CLI & User Interface for deploying applications and custom versioning application being deploy.
- Once **CDAP Edge is publicly available**, the artifact JAR file can be pushed the same way.

Pipelines

- **CDAP Pipelines** are currently exported as **JSON** specification
- CDAP Provides REST APIs / CLI to deploy JSON to create pipelines in different environment
- Same holds true for deploying pipeline to the **CDAP Edge environment**, it's just source / sinks are different for that particular environment.

1.7 — Exporting Code

At the end of the pilot, I want to take the code to another platform, what would we export from your platform? In what cases would you have to rewrite the code?

Custom Applications and Plugins

- **Custom Applications and Plugins** are developed using your favorite IDE and the code for those are checked into a source control system.
- **Exporting to non-CDAP environment –** then one would have to remove the wrapping CDAP APIs, build it and deploy it to non-CDAP environment using the environment specific set of tools.
- **To another CDAP based environment – then the build system generated artifact** in the form of JAR file that can then moved to another environment seamlessly without having to change a single line of code – as long as CDAP environments are compatible.

Pipelines

- **CDAP Pipelines** are currently exported as **JSON** specification
- **Exporting to non-CDAP environment–** the JSON specification has to be translated and Plugins need to be mapped to the target specification – it's some code, but not that much depending on target system requirements.
- **To another CDAP based environment** – then there is absolutely no change required.
- If the format for **JSON specification has been changed** and if you were migrating to a non-compatible environment, an upgrade tool is provided by CDAP to achieve seamless transition.

1.9 — What language is the code abstracted into (if any)--and is there a code-free option on-top?

Spark SQL, Spark Streaming and Other Examples

- Spark Streaming, MapReduce and Spark are **abstracted as JSON specification within CDAP Pipelines**. This is a **code-free option**.
- The **pipelines translate into configuring the CDAP pipeline Application Template** through the configuration specified (No Code Generation).
- **Programmatic abstraction** is available for Spark Streaming.
- **Spark SQL is currently not abstracted** and we have no higher level constructs. SQL itself is high level.

2.0 — Code Extensibility

Describe the supported languages, capabilities and methods for developing / utilizing custom developed code functions / procedures / modules / extensions.

- Currently, any JVM based languages are supported for building Application, Application Templates, Programs, Datasets and Plugins
- To run modules from different language pipelines provides a Run plugin allowing one to run modules from any language
- Python APIs will be available soon for building specification of pipelines to be deployed.

2.1 — Code Extensibility

Describe the breadth, depth and velocity of any community of developers that may exist to create custom "modules" / "extensions" capabilities.

- Moderately sized community for building Plugins right now.
- We see around 8-10 plugins a month being developed.
- We see developers from enterprise building their own custom plugins.
- Message board sees around 10 new posts a day.

2. Spark Orchestration

2.1 — Orchestrating Spark engines

Spark SQL orchestration

- Developers have the ability to call Spark SQL from their Application using the APIs provided by CDAP
- Developers can also chain different Spark SQLs within Workflow
- Developers can also expose REST APIs for querying RDDs using Spark SQL.

Spark Streaming orchestration

- Developers can create custom Spark Streaming applications using Spark and CDAP APIs
- They can also use CDAP Real-Time pipelines for building Spark Streaming applications.
- The Spark Streaming pipelines are managed by the CDAP Platform

2.2 — Orchestration methods

What mechanisms are used to start and control jobs?

- CDAP natively supports Spark Program lifecycle management through REST APIs (Platform provided)
- CDAP Workflow Program can submit Spark Job

How do you manage how Spark code is deployed and tuned on the cluster?

- Spark code is packaged into an Artifact (Bundled JAR) that is deployed using platform provided REST APIs on to the cluster
- Metrics collected and exposed can be used to understand the bottlenecks – Spark metrics as well as CDAP generated metrics for detecting bottlenecks.
- Memory and VCores can be configured on per-job basis through RESTful interface or Programmatically

2.3 — Orchestration methods

Describe your approach to error handling especially with regard to use of job scheduling systems (yours or another job scheduling tool).

- Irrespective of how the job is schedule either through CDAP or an external system (using REST API) the handling of job failures is consistent across both (External scheduler can be used – trigger jobs through REST APIs)
- **Notifications** can be generated to notify system or users.
- **Handling Data Processing Failure** : Assume a Spark Job fails while processing data between time-range $\{t_1, t_2\}$ and it's scheduled to run every 30 mins.
  - Depending on configuration either the workflow will be failed state to ensure serial processing of data (e.g. sessionization)
  - Or next run is started with data being processed from $\{t_2+30, t_2+60\}$
  - Or next run could start from $\{t_1, t_2+30\}$

What range of Spark orchestration capabilities are covered through your interface?

- Spark jobs that are integrated with CDAP Workflow can be triggered based on schedule specified as crontab for now and in 4.2 release then can be event based (Data availability, Another workflow finishing, etc)
- Workflow allows passing of tokens from one Spark Job to other allowing jobs to be chained and operate on information passed from previous job run.
- Workflow allows forking and joining for parallel execution of Spark jobs, they also allow conditional flows
- Workflow can be triggered through REST APIs
- Workflow concurrency can be controlled

2.4 — What language is the code abstracted into?

Spark SQL

- Spark SQL can be issued through Java, Scala or Clojure
- Spark SQL can also be executed through a CDAP Service (REST API) - with this support you can issue Spark SQL queries through the languages that supports HTTP connectivity.

Spark Streaming

- Java, Scala and Clojure
- CDAP Real-time pipelines – JSON specification
  - Python and Java APIs are being added to Platform that generate specification.

2.5 — Other Stream Processing Options [Flink, Storm]

Which other micro-batch or streaming engines are you integrated with?

Tigon & Experimental Integration with Apache Flink, Esper

3. Index/Search

3.1 — Provide index and search capabilities that can scale beyond 1 billion records per day.

Specify proven use cases--how they were delivered and at what volume & velocity…including which major carriers this has been implemented in -- for index/search…

We can integrate with Indexing solutions, but we don't provide the solution for index/search capabilities

Our Premise

You are looking for a indexing technology that is capable of scaling to 1 billion records per day. The above response reflects lack of specific search technology that is built by Cask. But, we can ingest data into a Search engine using Streaming pipelines.

3.2 — Implement standard log event/message format

If so--what standard do you follow and how do you implement it?

Not Applicable, as section-3.1 is something that we don’t solve.

Our Premise

Please refer to previous slide for the assumption made for this response.

3.3 — Provide all capabilities necessary to build a lambda architecture for log-event time-series data

If so--specify how you would accomplish this. Essentially an architecture that allows data to stay in cluster and complex query responses in less than 2 minutes?

- CDAP platform supports processing and serving capabilities through Programs and SQL integrations (HIVE)
- CDAP Datasets (OLAP / Time-series, etc) support transactional updates for both Batch and Real-time
- Partial writes are not visible to any viewer / reader when a transaction is in progress.
- SQL access to CDAP datasets ensures that only successfully committed data is visible for the query.
- Selecting of visible rows for successfully committed transactions is a low-cost filter.
- For non ad-hoc queries, the data can be exposed through CDAP Services through RESTful APIs
- For ad-hoc queries we integrate with distribution provided SQL engines through HCatalog.

4. Agent

4.1 — Interaction with agents

Do you configure / manage other agents outside of those you developed? What types?

CDAP does not manage the lifecycle of any external agents like installing, configuring and monitoring.

Our Premise

You are looking for a technology that is capable of deploying the agents in the field and manage the lifecycle of those agents remotely.

4.2 — Light Weight Publisher Agent

What types of systems/devices can your agents be put on?

CDAP at present has no agents that can be put on devices.

CDAP Edge (In development, due Q2’ 2017) will be an agent that will be installed in highly constrained environments - 512 MB / 2 cores

Our Premise

If you are looking for a immediate solution for deploying agents on devices in the field.

4.3 — Subscriber Agent

What agents can you subscribe to?

- CDAP provides capabilities as client for connecting to agents through TCP, UDP and HTTP protocols.
- Syslog Agent, MQTT & Netflow

5. End to End Data Pipeline

5.1 — Pipeline mechanisms

Describe how you build and manage pipelines

- CDAP provides a **studio environment for visually** building, configuring, deploying and managing pipelines.
- Feature rich, **drag and drop UI** to build pipelines, ability to deploy Plugins and version management.
- Ability to improve productivity with **pipeline preview** for validating correctness
- Visually built pipelines can be exported as **JSON specification**
- **Automation supported through REST APIs and CLI** for deploying, running, monitoring, reporting pipelines.

5.2 — Multi-tier pipelines

Describe your approach to creating tiers of reusable frameworks--from job steps to complete pipelines or combinations of pipelines

Pipeline can send message to other pipelines to trigger processing using REST / TMS

Pipeline Weaving with Reliable Messaging

Pipeline Templates with Macro support

With Macro support, enables re-configuration of the pipeline for different scenarios

Source | **Sink** | **Transform** | **Analytics** | **Actions** | **Post Action**

Build different implementation of Plugins

Plugin Interface

Add more interfaces to pipelines to customize and add new plugin types

Application Template

Add a new application template to create new type of pipelines

5.3 — Back-pressure

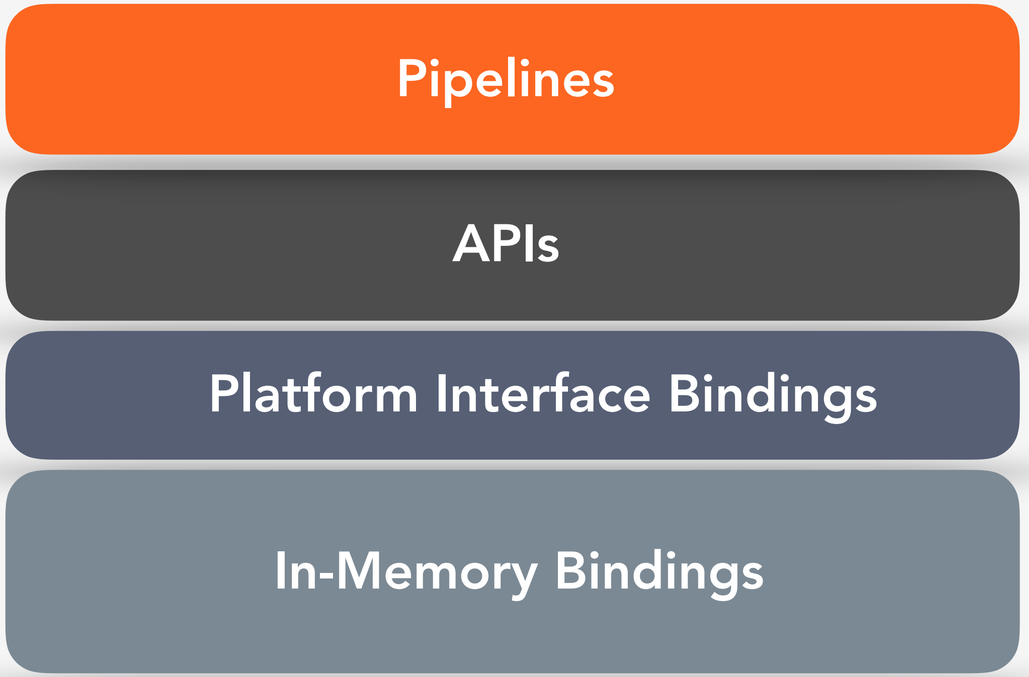
Specify how you handle back-pressure--and explain why this gives you an advantage

- All the Plugins within the real-time pipeline work in conjunction with each other.
- If the Plugin with-in a real-time pipeline is unable to catch with the rate of ingestion, enabling the back-pressure option would cause the system to automatically slow-down the input events rate to the plugin. This can cascade all the way till the source.
- Enabling back-pressure ensures that the entire pipeline doesn't crashes or show unexpected behavior or degrade overall performance due to a slow Plugin within the Pipeline.
- If the option is not turned on, then the user needs to ensure that Plugin handles the max input rate of the events into the Plugin – if not the pipeline will have unexpected behavior.

5.4 — Edge pipelines and ingestion

Describe your capabilities to manage edge pipelines and edge ingestion when the initial steps occur outside of Hadoop. Please specify other tools you may integrate with to help fill this space

- Capabilities to manage edge pipelines and edge ingestion outside of Hadoop - **Not Available Yet**
- **CDAP Standalone runs outside of Hadoop** on laptops or docker or VM environment, but it's not production ready.
- **CDAP Edge** is a project in the works that would run **lean version** of current CDAP Standalone in a constrained environment on the Edge.
- Customer Requirement to run in **environment as small as 512 MB / 2 cores** for doing edge level aggregation and push-up.



5.5 — Pipelines outside of Hadoop

List and describe the range of connectors (file types, DB Connectors, other mechanisms, etc...) you support for data ingestion

- **File Types Support** – Text File, CSV, Parquet, AVRO, ORC, Mainframe - Fixed Length, Excel, XML, JSON, Fixed Length Text Files & HL7.
- **Database & Warehouse** (Driver based) – Oracle, MySQL, PostgreSQL, Netezza, Vertica, SQL Server, Redshift, Snowflake
- **Columnar and Document Store** – Cassandra, MongoDB, **Kudu***
- **Real-time Source** – Kafka, JMS, Rabbit MQ, Active MQ, Twitter, Stream (CDAP), TCP, UDP, HTTP
- **Watcher Sources** – File
- **Other Sources** – FTP, UNC Window, HTTP Poller
- **Cloud Sources** – S3, Kinesis, DynamoDB, Azure Queue, Azure Event Hub
- **Bulk Import** – Oracle, Netezza, MySQL, PostgreSQL, Redshift, Snowflake.
- **CDC Support* –** SQL Server, MongoDB and Oracle (Log Miner)
- **HTTP** - **Salesforce*** & Marketo
- **Search Indexes** – Elasticsearch

*** In Active Development**

5.6 — Pipelines outside of Hadoop

List and describe the range of connectors (file types, DB Connections, other mechanisms, etc…) you support for data persistence / output / presentation

- **File Types Support** – CSV, Parquet, AVRO, ORC, XML, JSON, Snapshot Files
- **Database & Warehouse** – Oracle, MySQL, PostgreSQL, Netezza, Vertica, SQL Server, Redshift, Snowflake, Hive
- **Columnar and Document Store** – Cassandra, MongoDB, **Kudu***
- **Real-time Sink** – Kafka, JMS, Rabbit MQ, Active MQ, Twitter, Stream (CDAP), TCP, UDP, HTTP
- **Other Sink** – FTP, TCP, UNC Window
- **Cloud Sources** – S3, Kinesis, DynamoDB, Azure Queue, Azure Event Hub
- **Bulk Import and Export Support** – Oracle, Netezza, MySQL, PostgreSQL
- **Custom CDAP Dataset** – OLAP Cube, Timeseries, Lookup Table
- **Search Indexes** – Elastic Search, Solr

*** In Active Development**

5.7 — Can Pipeline run on Spark or MapReduce or Spark Streaming

List and describe the range of connectors (file types, DB Connections, other mechanisms, etc…) you support for data persistence / output / presentation

- **File Types Support** – CSV, Parquet, AVRO, ORC, XML, JSON, Snapshot Files
- **Database & Warehouse** – Oracle, MySQL, PostgreSQL, Netezza, Vertica, SQL Server, Redshift, Snowflake, Hive
- **Columnar and Document Store** – Cassandra, MongoDB, **Kudu***
- **Real-time Sink** – Kafka, JMS, Rabbit MQ, Active MQ, Twitter, Stream (CDAP), TCP, UDP, HTTP
- **Other Sink** – FTP, TCP, UNC Window
- **Cloud Sources** – S3, Kinesis, DynamoDB, Azure Queue, Azure Event Hub
- **Bulk Import and Export Support** – Oracle, Netezza, MySQL, PostgreSQL
- **Custom CDAP Dataset** – OLAP Cube, Timeseries, Lookup Table
- **Search Indexes** – Elastic Search, Solr

*** In Active Development**

6. Data Access

1.6 — Visualization and reporting

Describe any visualization and/or dashboard capabilities provided within your tool

- CDAP **aggregates metrics** from underlying infrastructure, system, pipeline and custom applications
- **Current version of CDAP** provides **minimal capabilities** for building operational dashboards for monitoring pipelines and custom applications.
- New **Application Level Monitoring dashboards** are currently planned for CDAP 4.2 release (April' 17)
- Generally enterprises use best of dashboard tools for operations, hence, from **CDAP perspective we make it easy to ingest data into those external systems**. Data is pushed through Kafka or can be pulled through REST APIs.

Describe any capabilities provided by your tool that would support reporting or ad-hoc querying against the data that separates your from your competitors.

- All CDAP Datasets and Streams are **automatically backed by HIVE / HCatalog** - HBase, HDFS or composite (Virtual Tables). Supports JDBC / ODBC for Datasets.
- Real-Time ingestion into CDAP Streams can be **queried in realtime**.
- **Pre-Built CDAP OLAP Dataset** allow for real-time or batch ingestion into Cube. Also, supported by a **CDAP Service for reporting data through REST APIs** to external visualization tools.
- Support for creating **Ad-Hoc query-able views over Dataset and Streams** without copying or moving data – True schema on Read

CASK

7. Additional considerations

7.1 — Built ground-up for Hadoop ?

What advantages do you have due to this ?

- Entire CDAP system is native to Hadoop
- CDAP architecture inherently allows Pipelines and Custom Application to **scales horizontally**
- Scales as your Hadoop installation scales - 3 nodes, 200 nodes, 650 nodes, 1000 nodes seamlessly
- Does not require out of Hadoop deployment for CDAP system — uses Apache YARN for CDAP run-time
- No specialized hardware required for CDAP

What specific Hadoop ecosystem components, including processing engines do you support / not support ?

- **Data Storage and Access Systems** — Apache HDFS, Apache HBase, Apache HIVE, Apache Impala, Apache YARN, HCatalog, Apache Kudu
- **Data Serialization** — Apache AVRO, ORC, Parquet
- **Processing System** — Apache MapReduce, Apache Spark (Spark Streaming, Spark SQL), Apache Twill, Apache Tez
- **Distributed Coordination** — Apache Zookeeper
- **Distributed Transactions** — Apache Tephra
- **Data Transport** — Apache Flume, Apache Kafka
- **Security** — Apache Sentry, Apache Ranger
- **Intelligence** - Spark ML, Mahout
- **Experimental** — Presto, Apache Flink, **Planned** — Apache Beam
- **Installation and Management** — Apache Ambari, Navigator, Cloudera Manager
- **Not Supported** — Apache Pig, Apache Storm

7.2 — Open Source

What components are not open source ?

- **None of CDAP is proprietary**
- All of **CDAP is 100% open source** – System, User Interfaces, Integrations, Tools and Build configurations
- Licensed under **Apache 2.0**

7.3 — Performance and Scalability

Describe how and under what conditions the tool performance begins to degrade as well as limitations or considerations on scalability.

Degradation

- Underlying infrastructure upgrades and outages degrades the performance of running applications - **tradeoff is resiliency**
- Network latency between zookeeper nodes affects – CDAP Services – REST APIs, UI & Applications running.
- If you are running on private cloud VM, the host and other VMs can affect performance – CPU, Memory and Network

Limitations

- Number of concurrent Batch Pipeline that can be started simultaneously 200 / minute (Goal is to get it to 2000 / second - customer requirement in your domain)
- Transacted operations are currently limited to 5,000,000 operations / second.
- Transaction server currently is not load balanced

7.4 — Debugging

When something goes wrong, what skill set is typically needed to resolve the issue? Specify a few common problems in development and deployment--and how they would be resolved. Specify any coding skills that may be necessary.

Infrastructure under CDAP has Issues

- E.g. not enough containers, CDAP logs and management metrics will highlight the issue
- Operations team member in combination with logs and management dashboard can diagnose the issue.

Pipeline is failing or App Failing or Plugin failing

- Combination of pipeline logs and metrics can help pin-point the issue
- If developer is required - can attach debugger to pipeline or application or can use the unit testing framework to reproduce the issue.
- Most of the time logs and metrics are good enough

Common Development Problems

- Not Handling Null Data or Number Conversion Exceptions
- Filtering incorrect data
- Incorrect transformation
- Not handling incorrect data
  - Pipeline Preview capability allows you to dry run and inspect data to resolve issues

Common Deployment Problems

- Not enough capacity on cluster to run multiple jobs.
- Invalid Hadoop configuration (e.g. setting replication factor of 4 on 3 node cluster)
- CDAP makes every possible attempt to report issues during startup and runtime through logs and metrics

8. Security Considerations

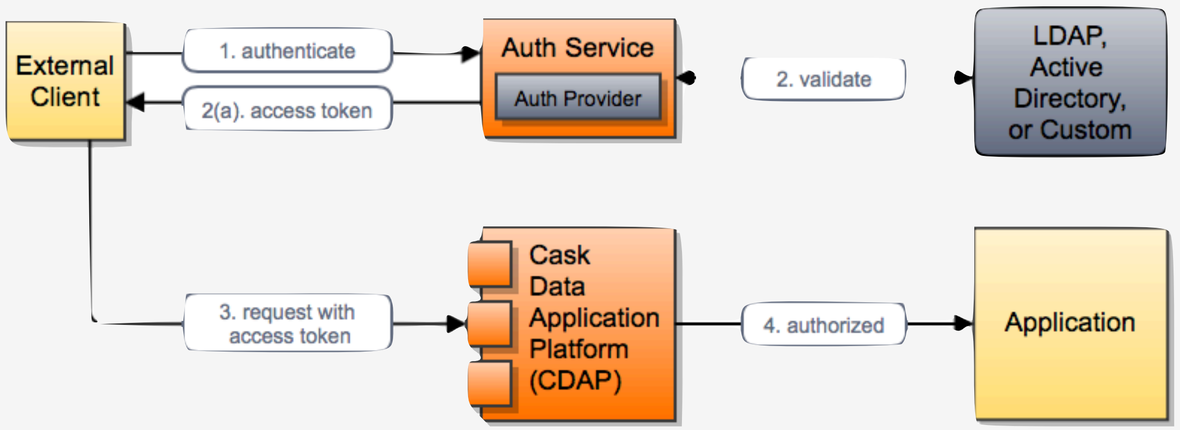
8.1 — Confirm integration or support--and provide a short description of what you mean (how &/or to what extent)

Kerberos Integration

- Support both Kerberos authentication and delegation tokens for all common services (HDFS, YARN, Hive, Spark).
- Supports user impersonation per application via Kerberos authentication.
- Automatic delegation updates for all types of long running applications (Spark streaming, REST services, long running daemons, etc)

Authentication

- Supported via Jetty authentication module.
- Already has LDAP and MTLS supported out of the box
- AWS IAM Integration – In Progress
- Extensible Custom Authentication



8.2 — Confirm integration or support--and provide a short description of what you mean (how &/or to what extent)

Authorization

- Authorization allows users to enforce fine-grained access control on CDAP entities: namespaces, artifacts, applications, programs, datasets, streams, and secure keys.
- All operations on these entities—listing, viewing, creating, updating, managing, deleting—are governed by authorization policies.
- Built-in support for authorization and is pluggable to any external authorization system
- Supported integration with Apache Sentry, Apache Ranger & Active Directory
- AWS IAM Integration In-Progress

Secured Communication at Edge

- All REST services are configurable to use TLS/SSL

8.3 — Confirm integration or support--and provide a short description of what you mean (how &/or to what extent)

Sentry Integration

- CDAP Authorization can be backed by Sentry via the CDAP Apache Sentry authorization extension
- Apache Sentry Model, Policy Engine and Sentry Binding provided out-of-box
- Easy configuration for integrating Apache Sentry and CDAP
- Integrated with HUE User Interface

Cloudera Navigator Integration

- Cloudera Navigator for business metadata search
- Discovery to CDAP entities from Cloudera Navigator
- More enrichment metadata integration in 2017.

8.4 — Confirm integration or support--and provide a short description of what you mean (how &/or to what extent)

Auditing Support [if so--how]

- CDAP has in-built support for capturing audit logs for Applications and Datasets running or being accessed
- Activities include the creation, modification, and deletion of an entity.
- Also includes modification of the entity's metadata
- Computing lineage for all the data access by different entities.
- Out-of-box without any additional configuration or development
- Tracks Non-Hadoop sources and sinks as external Datasets
- Published on messaging system to integrate with third party systems

Centrify Integration

- No integration currently available
- Requires implementation of plugin for integration with Centrify

8.5 — Confirm integration or support--and provide a short description of what you mean (how &/or to what extent)

Encryption Support

- Rely on the underlying infrastructure (e.g. HDFS encryption)

KMS Support

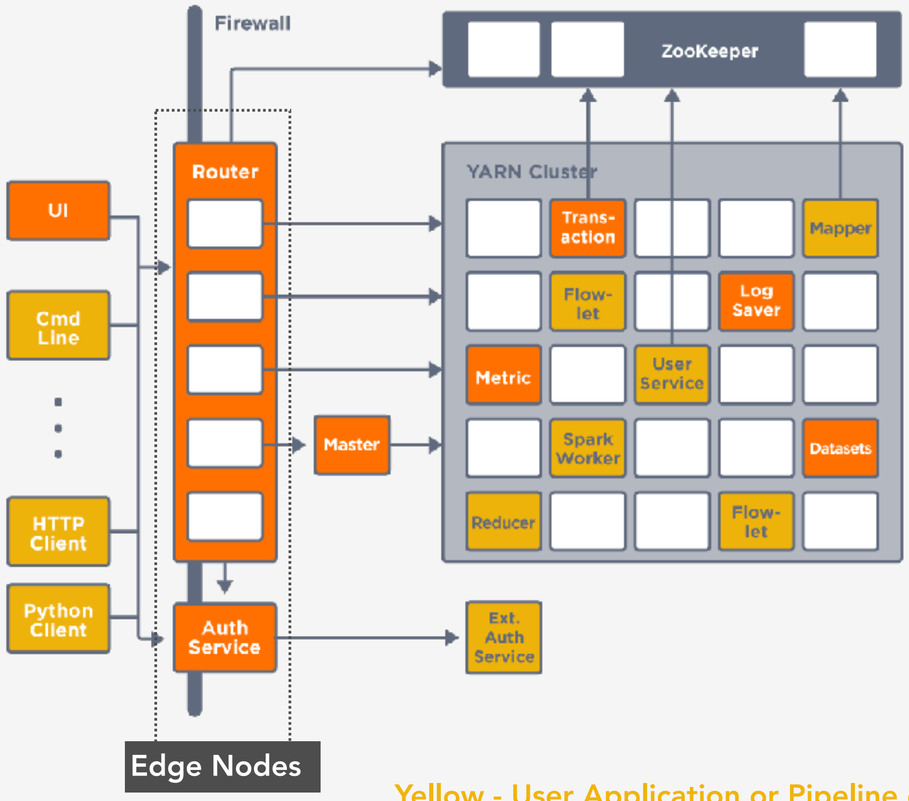
- Provides programmatic and REST API access for secrets management backed Hadoop KMS.
- CDAP programs and data pipelines have secure access to the keys via the programatic APIs and Macro's support

9. Infrastructure

9.1 — Infrastructure requirements for deployment

Does the tool install and reside fully within the Hadoop cluster?

- Majority of CDAP resides on the Hadoop Cluster itself within YARN Containers
- Few components are installed on Edge Nodes (a.k.a Jump Nodes)
  - CDAP Router, CDAP UI
  - CDAP Auth-Service and CDAP Master
- No of instances as per redundancy requirements by business
- CDAP Router & CDAP UI are required to be on edge node because they are services that can be behind a VIP when multiple instances are running for load balancing.
- CDAP Auth-Service & CDAP Master needs to impersonate in secured mode, hence need to be started with *kinit*



Yellow - User Application or Pipeline or Tools

Orange - CDAP Components

9.2 — Infrastructure requirements for deployment

What additional required and/or optional hardware / software components might be needed that could necessitate separate licensing / configuration?

Hardware Requirement

- Generally all Hadoop clusters have Edge nodes (a.k.a Jump nodes)
- Few of CDAP components installed on Edge nodes will require dedicated machines depending on business case
- All services can be installed on single node in Development or Experimental environment

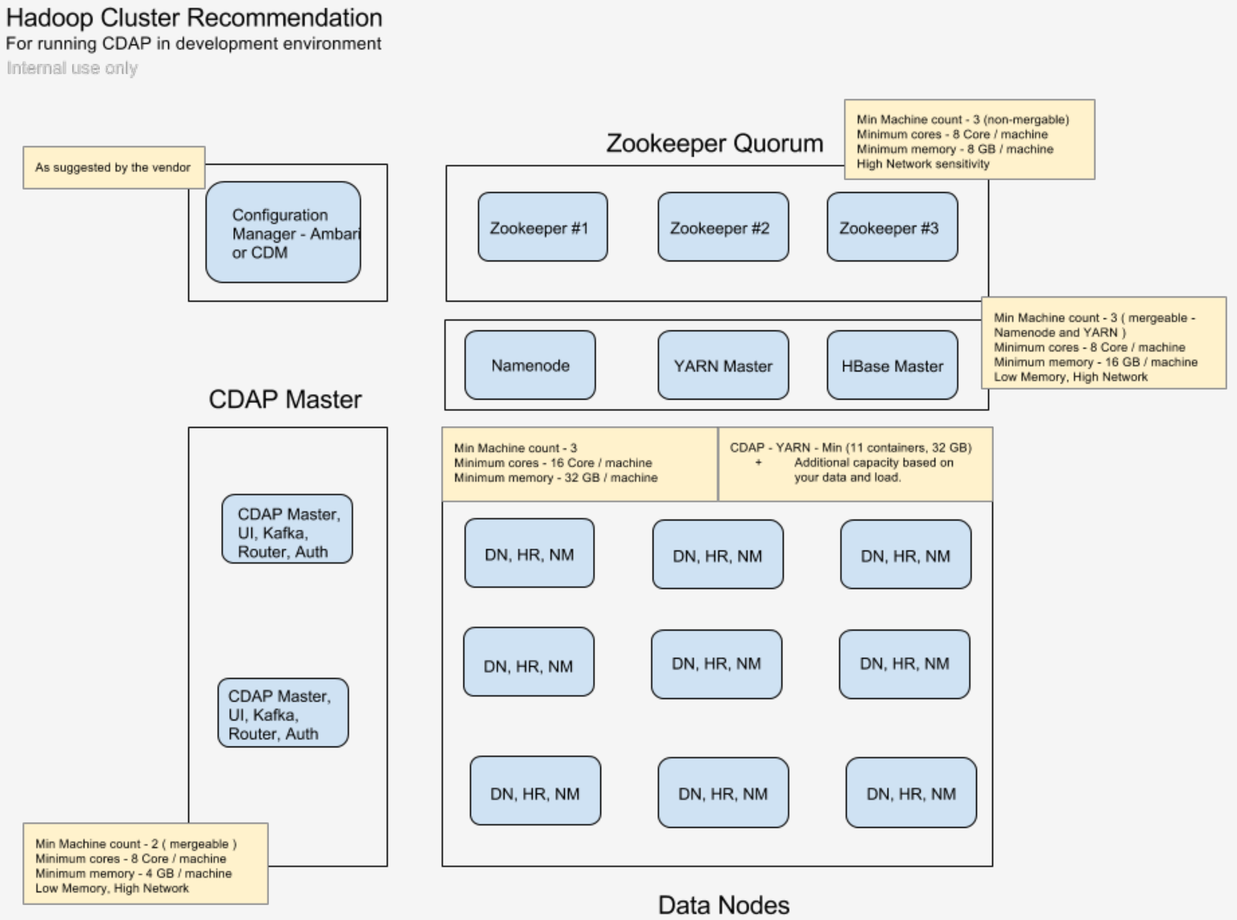
Software Requirement

- No additional software is required for CDAP to run other than Hadoop.

9.3 — Infrastructure requirements for deployment

Provide a sample hardware / software configuration and related sizing assumptions for the tool

- Shown on the right is the configuration and setup for a typical stable development environment
- This is what Cask recommends the customer to have, but depending on budget and hardware available a viable configuration for development would be achieved.
- For production, we would work with customer to provide them the sizing guidelines.



10. Ingestion capabilities

Ingest from Streaming Sources, Databases, Files & Cloud Sources

10.1 What real-time stream ingestion data capabilities exists in CDAP ?

- **CDAP**
- ECA system includes a Rule Engine (RE) for specifying expressions for trigger alerts depending on telemetry being aggregated in real-time from devices
- Rule Engine (RE) works in conjunction with Generic Event Parser (GEP) for parsing multi-format events emitted by devices.
- It generates different aggregates depending on the needs to the expression being specified.
- E.g. Let's say you are deploying a new type of monitoring device in the field, the device will send telemetry data about various measures captured in the field along with device, location, version, etc information. Operator can specify different aggregations and business rules for trigger alerts. This kind of system has to operate at massive scale in real-time

CASK

11. Product Roadmap Clarification

Recent Questions on Roadmap

11.1 Response to product related questions

- **Event, Condition and Action (ECA) system for handling IoT workload**
  - ECA system includes a Rule Engine (RE) for specifying expressions for trigger alerts depending on telemetry being aggregated in real-time from devices
  - Rule Engine (RE) works in conjunction with Generic Event Parser (GEP) for parsing multi-format events emitted by devices.
  - It generates different aggregates depending on the needs to the expression being specified.
  - E.g. Let's say you are deploying a new type of monitoring device in the field, the device will send telemetry data about various measures captured in the field along with device, location, version, etc information. Operator can specify different aggregations and business rules for trigger alerts. This kind of system has to operate at massive scale in real-time

- **Non-Functional Attributes**
  - **Scalability** – Evaluating Business Rules in Real-time to generate reliable alerts
  - **Adaptability** – Support for handling multi-format and multi-version of data
  - **Simplicity** – Easy ways to add/ modify conditions for generating alerts
  - **Modularity** – Support for adding more type of data parsing and rule types

11.1 Response to product related questions

Data Management capabilities - partition expiration, SQL Integration, High order data patterns

- Unlike other data integration tools, CDAP naturally provides data management capabilities for managing datasets on the cluster. Every other tool treats Hadoop as DB, so they will only ingest.
- CDAP manages addition, expiration of data partitions maintained. E.g. In HIVE, partitions are created and managed as part of data lifecycle.
- Data sinks on HDFS are automatically integrated with HCatalog – enabling immediate query capability without any human intervention or setup.
- Dataset API allows one to programmatically build high order data patterns – like for e.g. Geo-Fenced Dataset, CDR
- The Datasets can then be directly integrated with data ingestion pipelines .

12. Migrating environments

How do you go from Dev, QA, Staging to Production

12.1 — Migrating Environments

How to move pipelines that you have build in CDAP from one environment to other

- All CDAP Pipelines are configuration represented in JSON
- They define individual plugins used, their version and configurations
- They also specify the connectivity graph
- Along with schedule definition, environment in which the pipeline should be run, and environment settings.

CASK

Thank You