WELCOME TO THE DATA AGE

Innovating with Data

Alex Campos | ac@cloudera.com

@campossalex

What's the expiration date of your company?


200
35mm color print film
200 24exp
Kodak
200
35mm color print film
EVIDENCE


STORE CLOSING
BLOCKBUSTER VIDEO
BLOCKBUSTER VIDEO
4111
ALL PREVIOUSLY VIEWED
DVD's
$9.99 EA.
OR LESS!
STORE CLOSING
EVERYTHING
ENTIRE STORE ON SALE!
NOTHING HELD BACK!
EVIDENCE


EVIDENCE


EVIDENCE

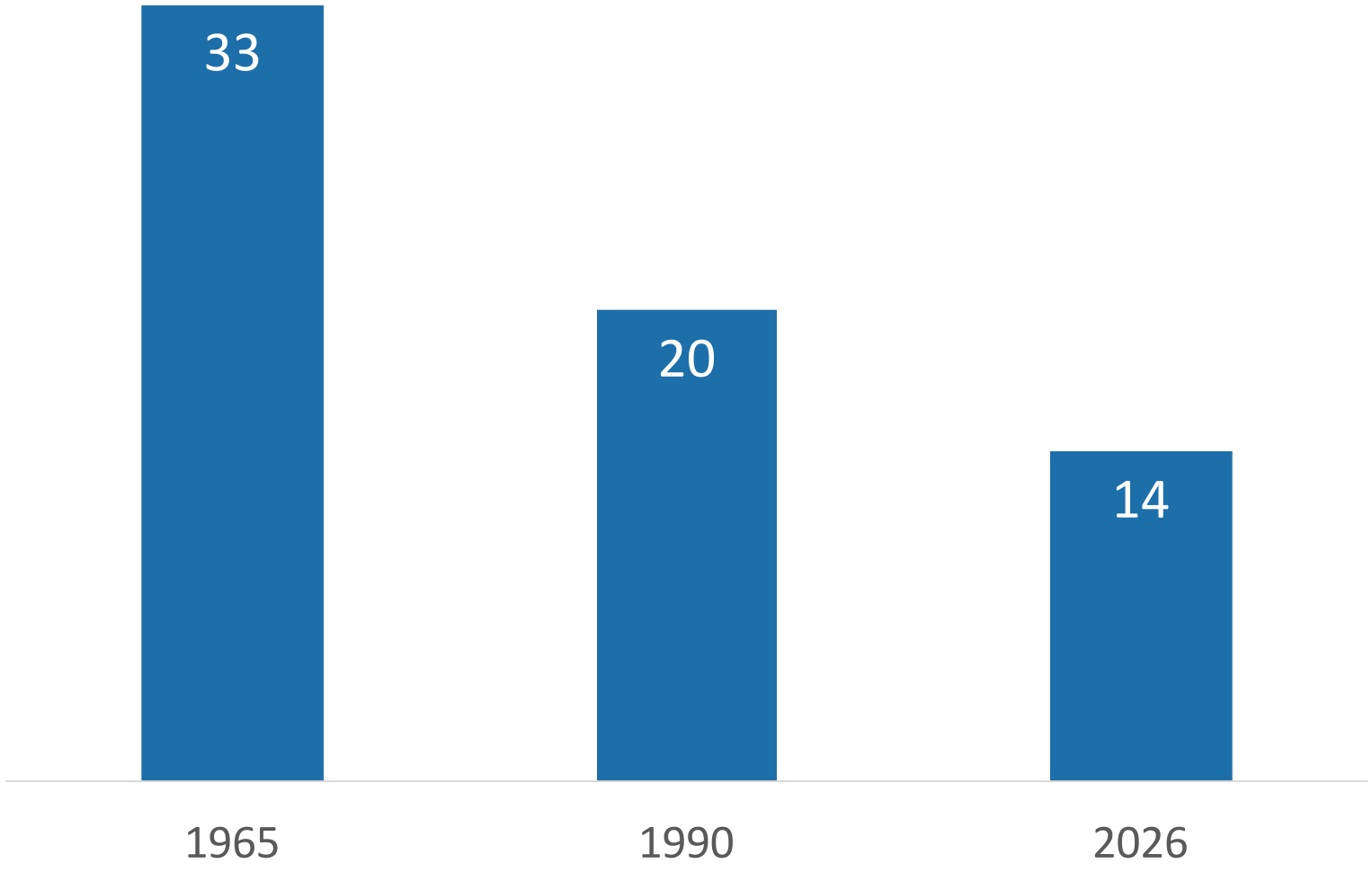

TAXI
EVIDENCE

Now let's talk seriously...

What's the expiration date of your company?

Why Half of the S&P 500 Companies Will Be Replaced in the Next Decade

S&P turnover



S&P

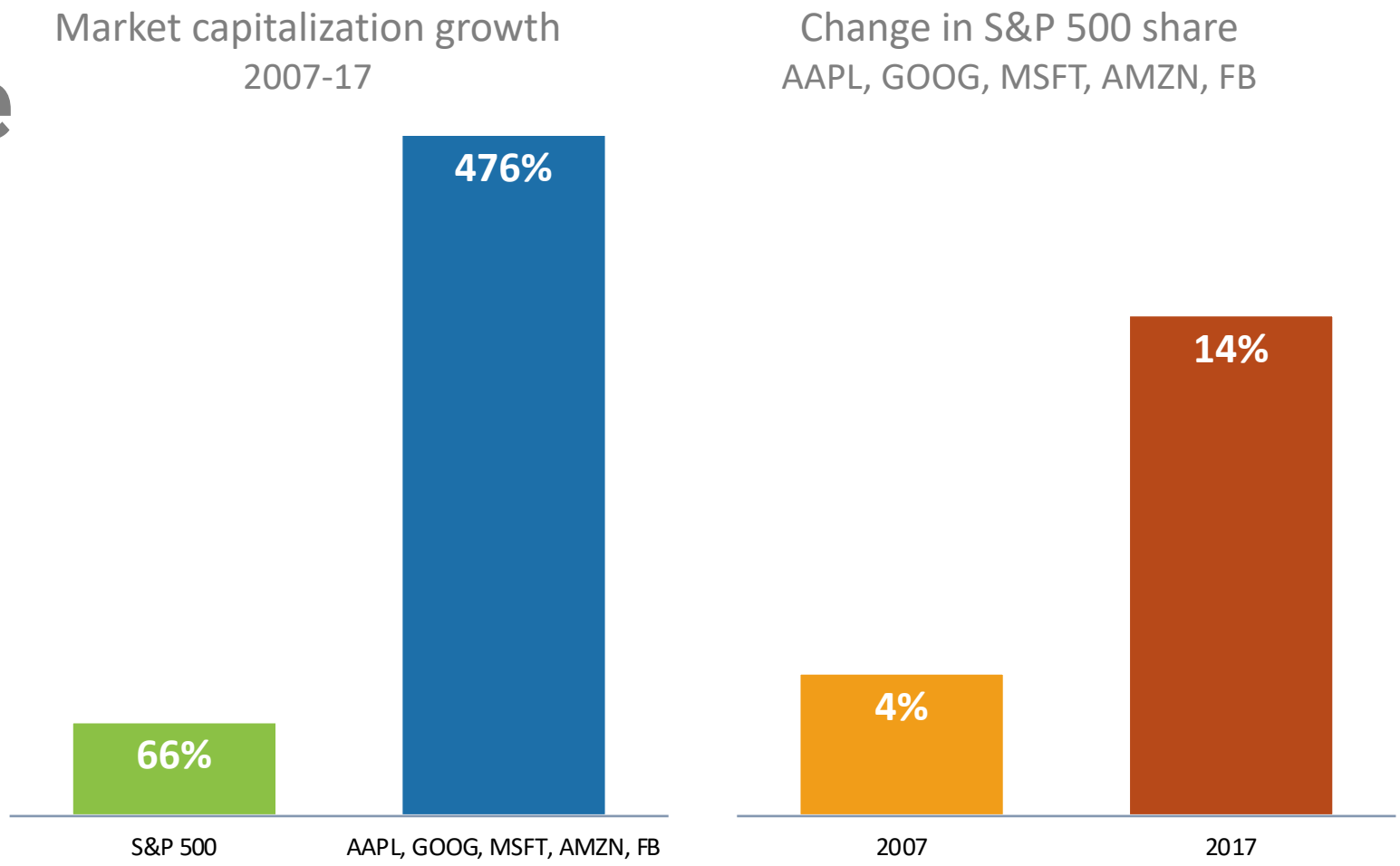
turnover

Focus on Today

Startups Threat

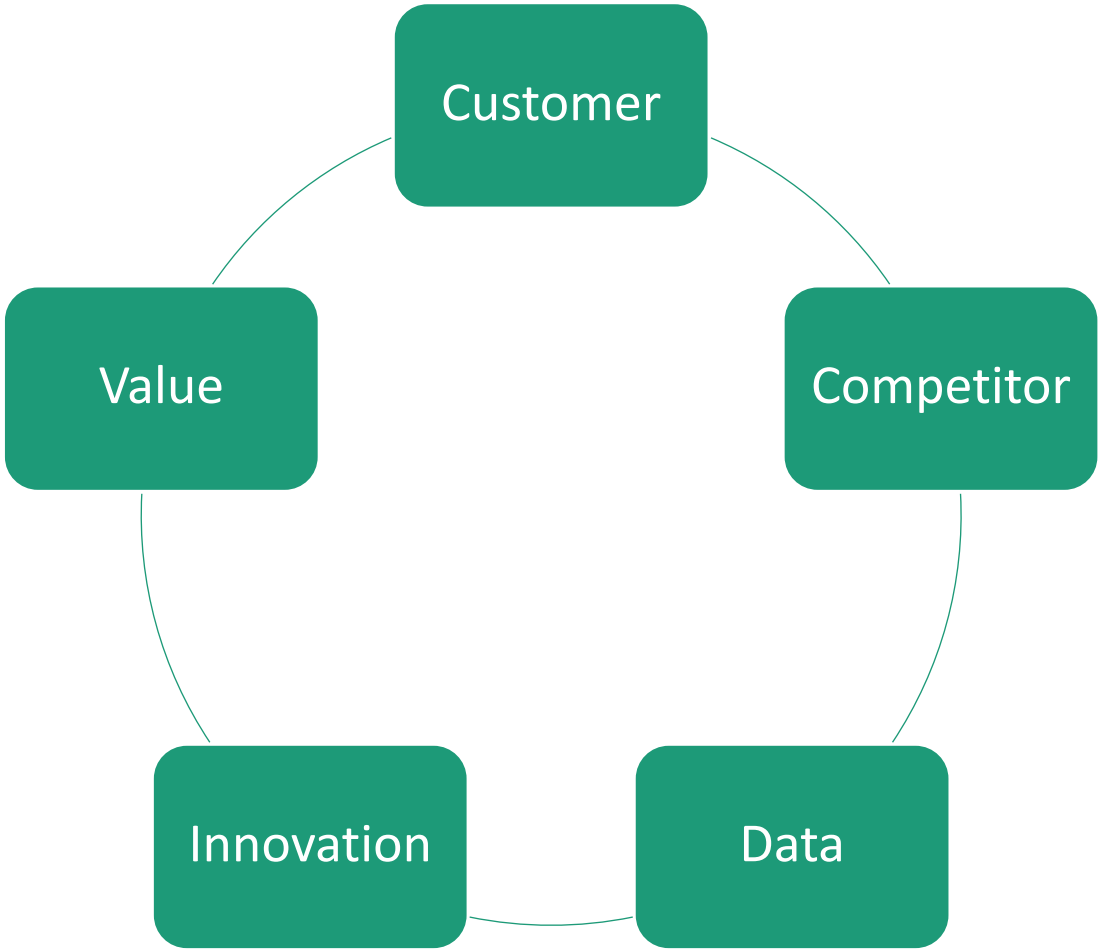
How companies are leveraging data to transform their business

Data-driven companies are accelerating in value

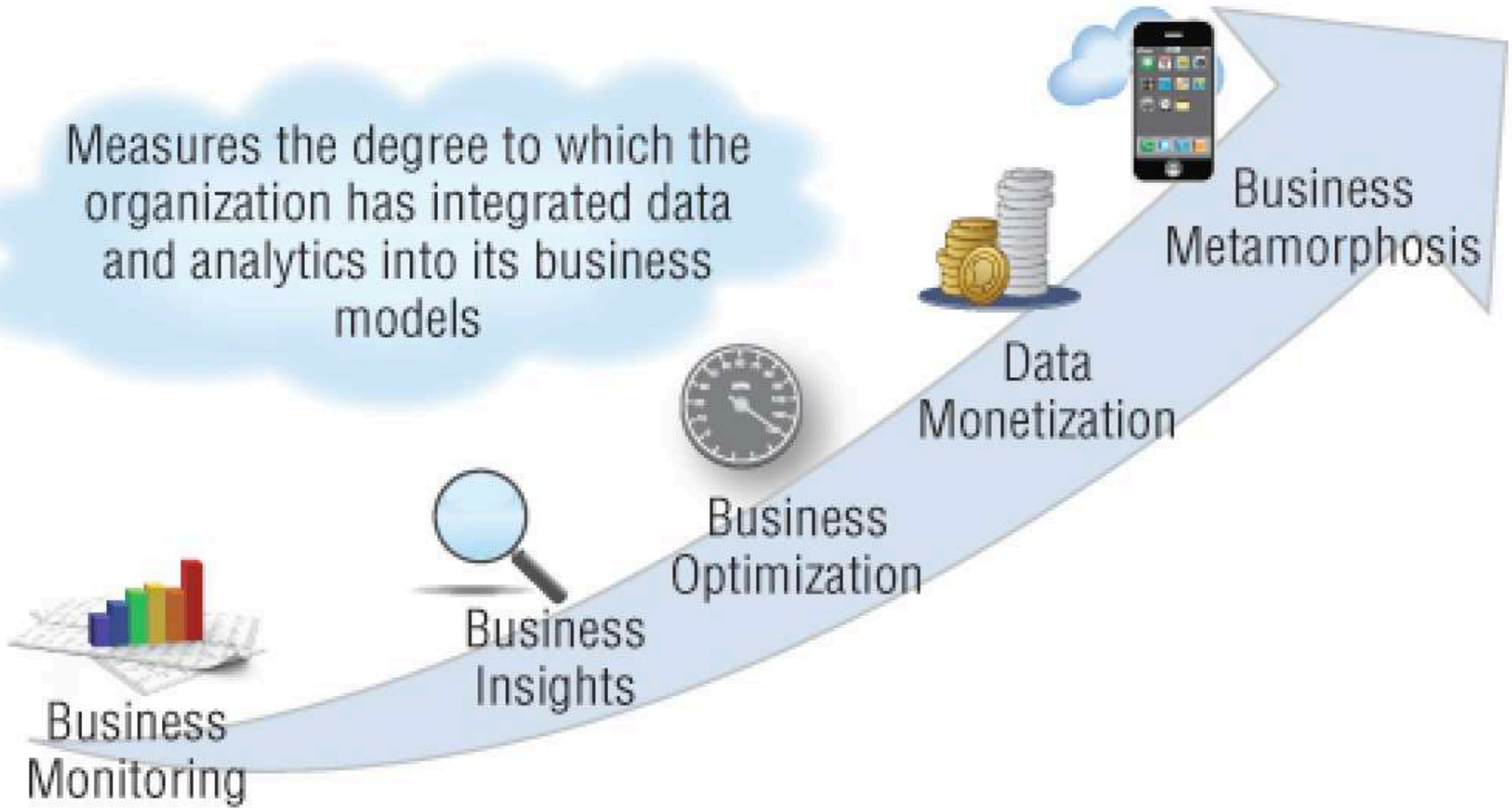


Source: Yahoo Finance and http://siblisresearch.com/data/total-market-cap-sp-500/

Digital Transformation is pushing more analytics



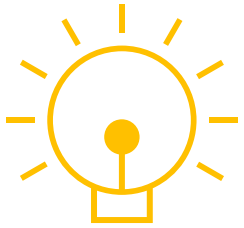
5 Domains of Digital Transformation, Digital Transformation Playbook, David Rogers



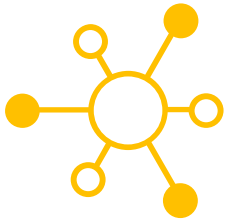
Big Data Business Maturity Index, Bill Schmarzo

Big Data in Action

GROW BUSINESS



CONNECT PRODUCTS & SERVICES (IoT)



PROTECT BUSINESS

Big Data in Action

HEALTH

SOCIAL

RESEARCH

EDUCATION

ASTRONOMY

SCIENCE

The rise of Big Data Platforms

Hadoop, the crazy elephant

An Open Source distributed platform to solve Big Data problems

The beginnings

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google*

ABSTRACT

We have designed and implemented the Google File System, a scalable distributed file system for large distributed data-intensive applications. It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients.

While sharing many of the same goals as previous distributed file systems, our design has been driven by observations of our application workloads and technological environment, both current and anticipated, that reflect a marked departure from some earlier file system assumptions. This has led us to reexamine traditional choices and explore radically different design points.

The file system has successfully met our storage needs. It is widely deployed within Google as the storage platform for the generation and processing of data used by our service as well as research and development efforts that require large data sets. The largest cluster to date provides hundreds of terabytes of storage across thousands of disks on over a thousand machines, and it is concurrently accessed by hundreds of clients.

In this paper, we present file system interface extensions designed to support distributed applications, discuss many aspects of our design, and report measurements from both micro-benchmarks and real world use.

Categories and Subject Descriptors

D [4]: 3—*Distributed file systems*

General Terms

Design, reliability, performance, measurement

Keywords

Fault tolerance, scalability, data storage, clustered storage

*The authors can be reached at the following addresses: {*sanjay,hgobioff,shuntak*}*@google.com*.*

SOSP'03, October 19–22, 2003, Bolton Landing, New York, USA.

1. INTRODUCTION

We have designed and implemented the Google File System (GFS) to meet the rapidly growing demands of Google's data processing needs. GFS shares many of the same goals as previous distributed file systems such as performance, scalability, reliability, and availability. However, its design has been driven by key observations of our application workloads and technological environment, both current and anticipated, that reflect a marked departure from some earlier file system design assumptions. We have reexamined traditional choices and explored radically different points in the design space.

First, component failures are the norm rather than the exception. The file system consists of hundreds or even thousands of storage machines built from inexpensive commodity parts and is accessed by a comparable number of client machines. The quantity and quality of the components virtually guarantee that some are not functional at any given time and some will not recover from their current failures. We have seen problems caused by application bugs, operating system bugs, human errors, and the failures of disks, memory, connectors, networking, and power supplies. Therefore, constant monitoring, error detection, fault tolerance, and automatic recovery must be integral to the system.

Second, files are huge by traditional standards. Multi-GB files are common. Each file typically contains many application objects such as web documents. When we are regularly working with fast growing data sets of many TBs comprising billions of objects, it is unwieldy to manage billions of approximately KB-sized files even when the file system could support it. As a result, design assumptions and parameters such as I/O operation and block sizes have to be revisited.

Third, most files are mutated by appending new data rather than overwriting existing data. Random writes within a file are practically non-existent. Once written, the files are only read, and often only sequentially. A variety of data share these characteristics. Some may constitute large repositories that data analysis programs scan through. Some may be data streams continuously generated by running applications. Some may be archival data. Some may be intermediate results produced on one machine and processed on another, whether simultaneously or later in time. Given this access pattern on huge files, appending becomes the focus of performance optimization and atomicity guarantees, while caching data blocks in the client loses its appeal.

Fourth, co-designing the applications and the file system API benefits the overall system by increasing our flexibility.

Adoption and community growth

Cloud-first, the new buzzword

And then, The Power of We

Architecture and Platform Evolution

CLOUD EXPERIENCE

Easy to use, self-service, on-demand, elastic, consumption

COMPUTE & STORAGE

Separation in public and private clouds for increased performance

KUBERNETES & CONTAINERS

Adoption as standard operating environment for flexibility and agility

STREAMING & ML/AI

Multi-function analytics for the data-driven enterprise

The "Hadoop" Philosophy

DISAGGREGATED SOFTWARE STACK

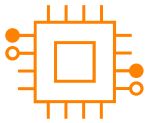
Storage +
Compute +
Security +
Governance +
SQL, DE, DF, ML

EXTREMELY LARGE-SCALE

Distributed systems
Commodity infrastructure, hardware & cloud

OPEN SOURCE!

Open data standards
Community scale



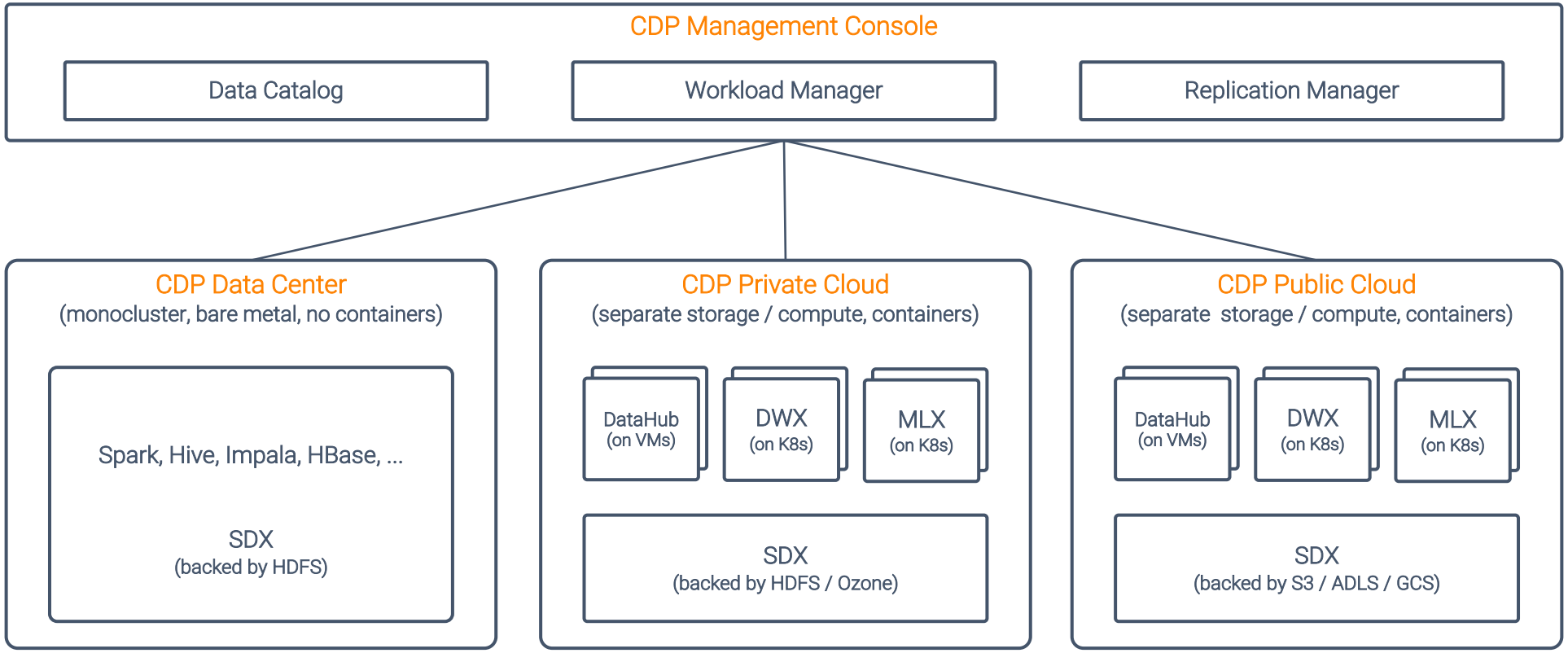
EVOLVING ECOSYSTEM

Diverse technologies
Innovation at every layer, independently

A movement towards a modern architecture for managing and analyzing data

Cloudera Data Platform

Multi-Cloud Hybrid Architecture



Enterprise Data Cloud

Multi-Cloud Hybrid Architecture

Hybrid, Multi-Cloud

Multi-Function

Secure & Governed

Cloud Experience

Data-driven journey,
but from expectation to reality...


Product

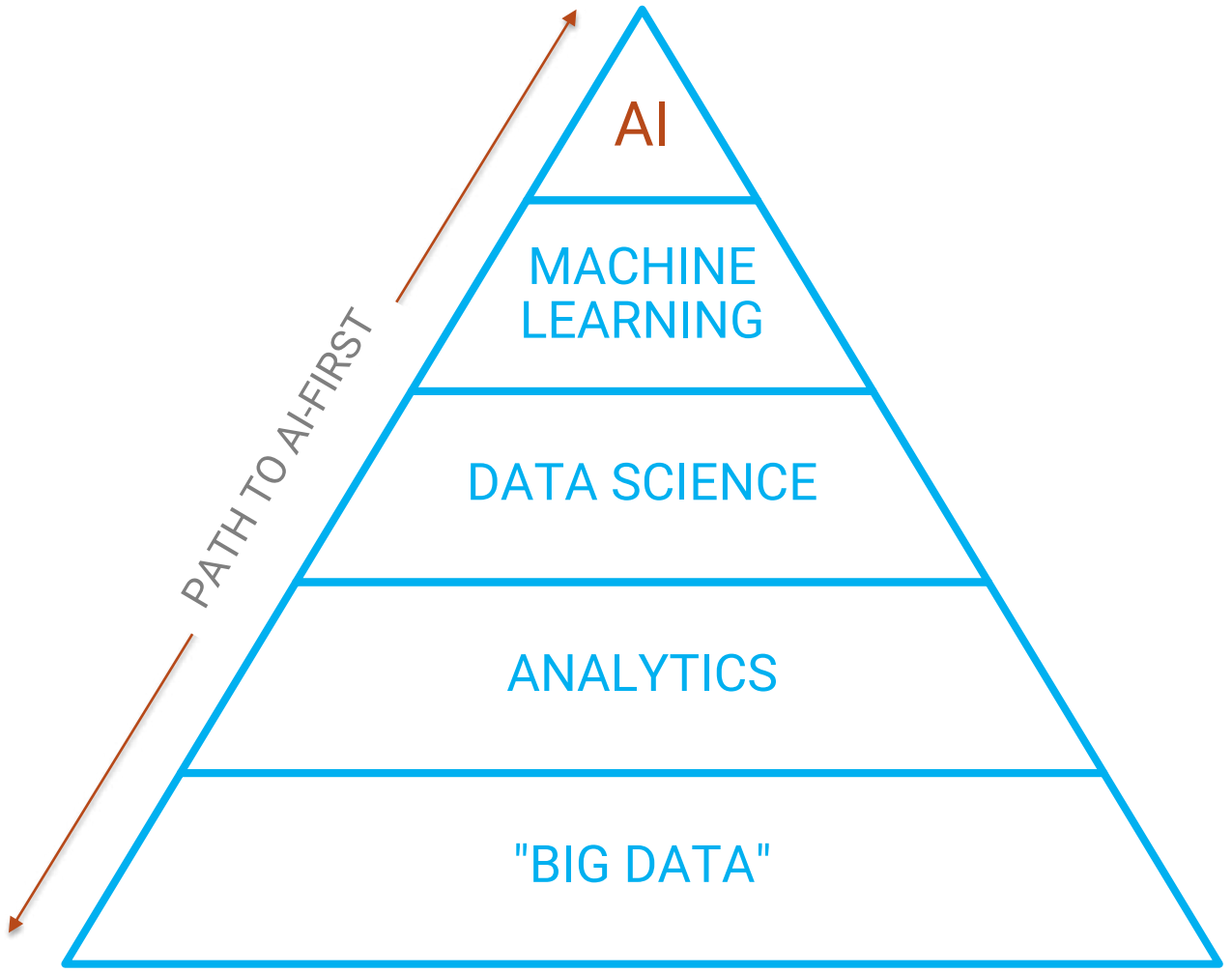
Crayola
burnt orange
naranja quemado
mandarina atómica
tangerine atomique
Crayola
inchworm
verde oruga
chenille
Crayola
periwinkle
pervenche
Crayola.com
Crayola
jazzberry jam
palo de rosa
rose chalouille


Strategy
Execution
Success
RESEARCH
WEB
DATA
TECH
COMMERCE
1000000 USERS

So,
what’s the secret sauce?

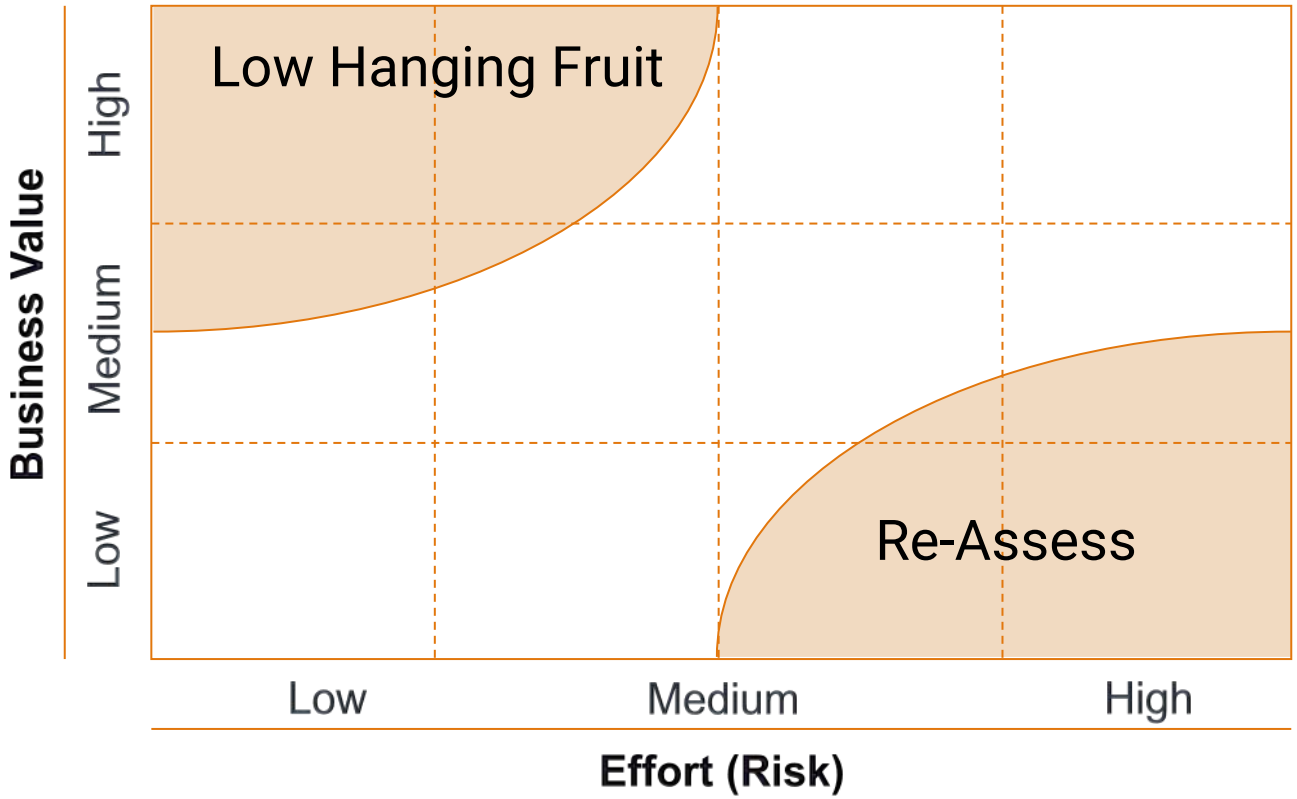
There is no secret sauce...

Smart Roadmap



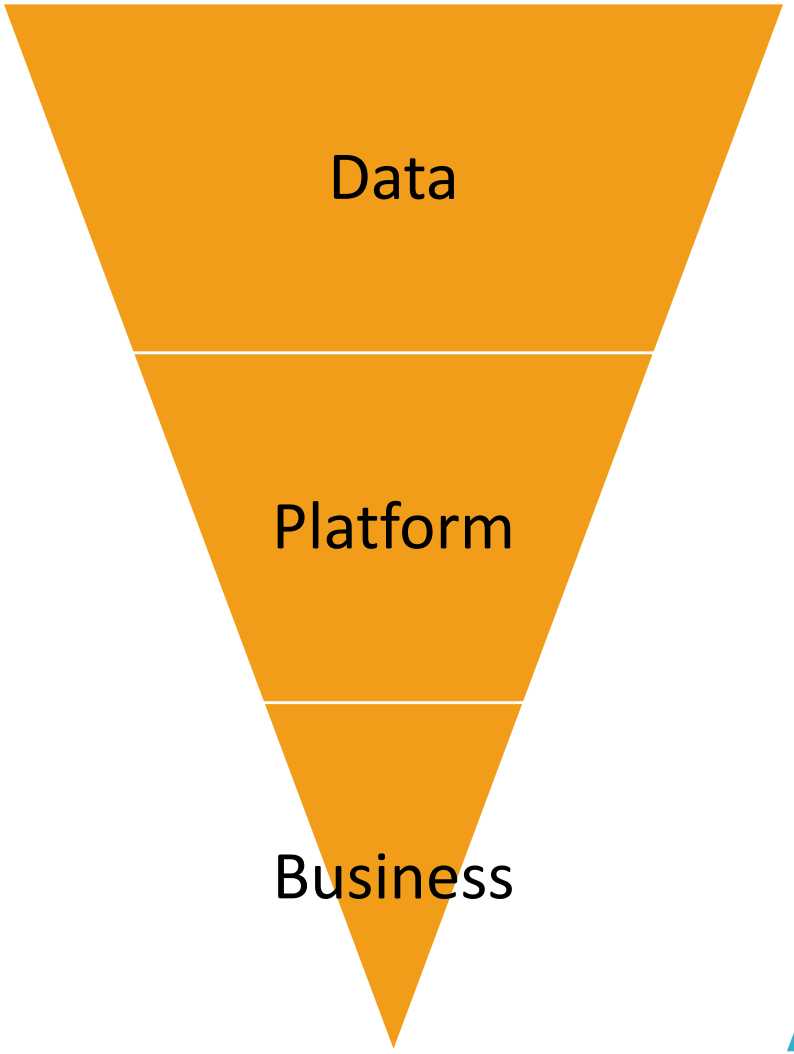
Data science and machine learning hierarchy of needs

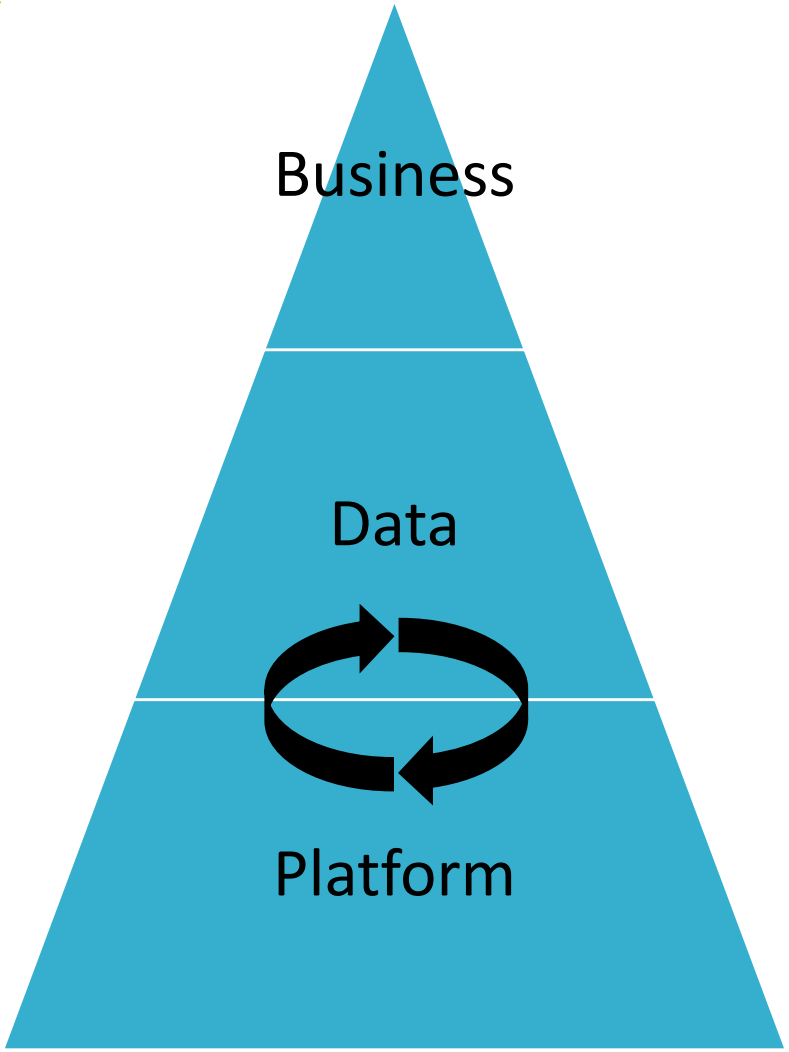
Smart Roadmap



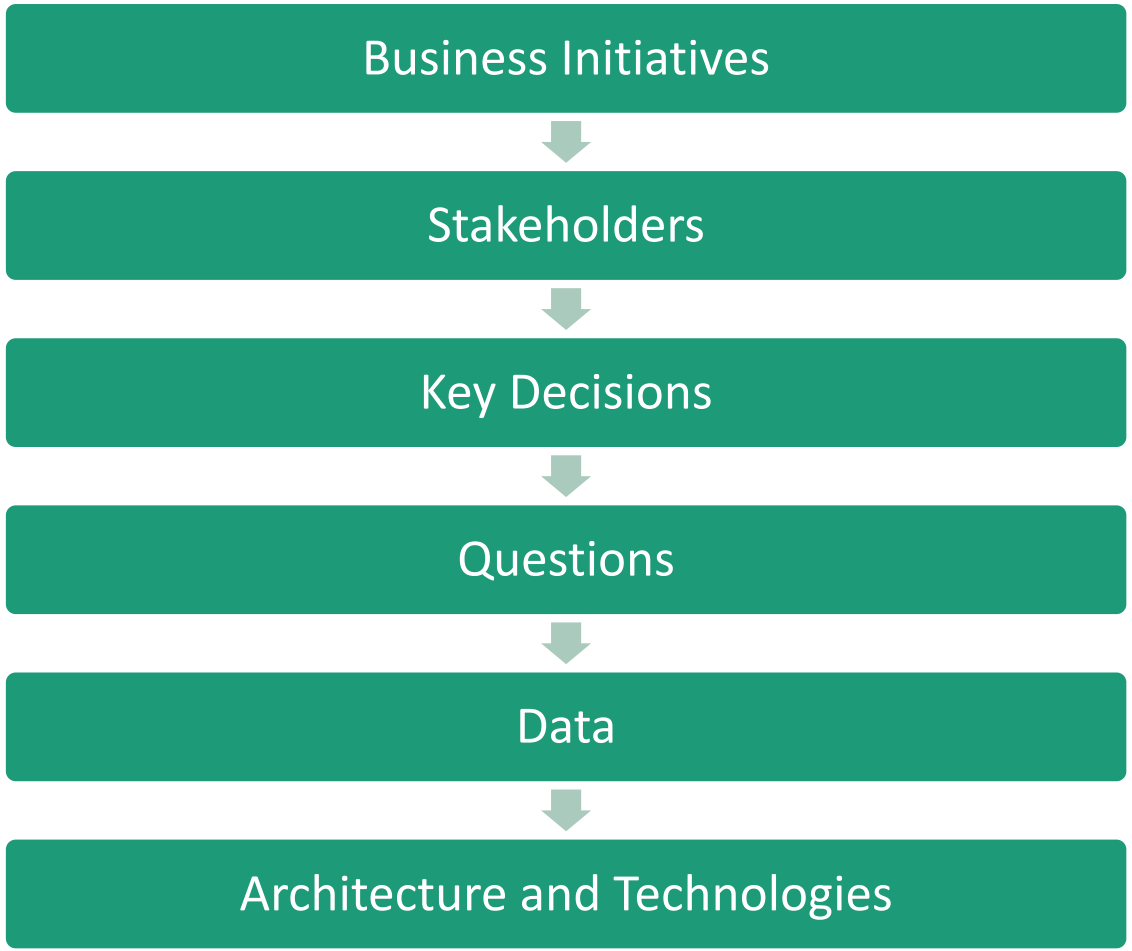
Prioritization Matrix

Align with Business Needs





Align with Business Needs



Start with the end in mind

Use Case Factory

Data

Analytics Capabilities

Design

Generation Capabilities

Use Case Factory

When Design Thinking meets Analytics

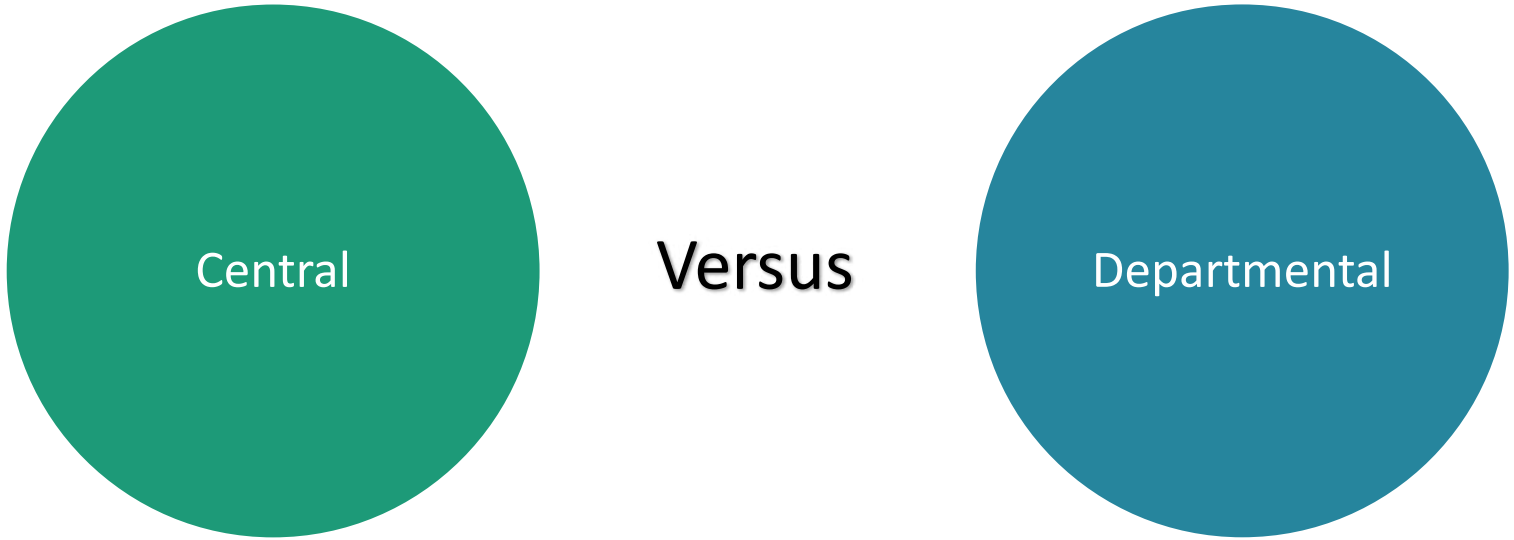
- Agility to create analytics use cases
- Fail fast, learn in the path
- Fast pace from Lab to Production

Data Literacy

Do you speak data?

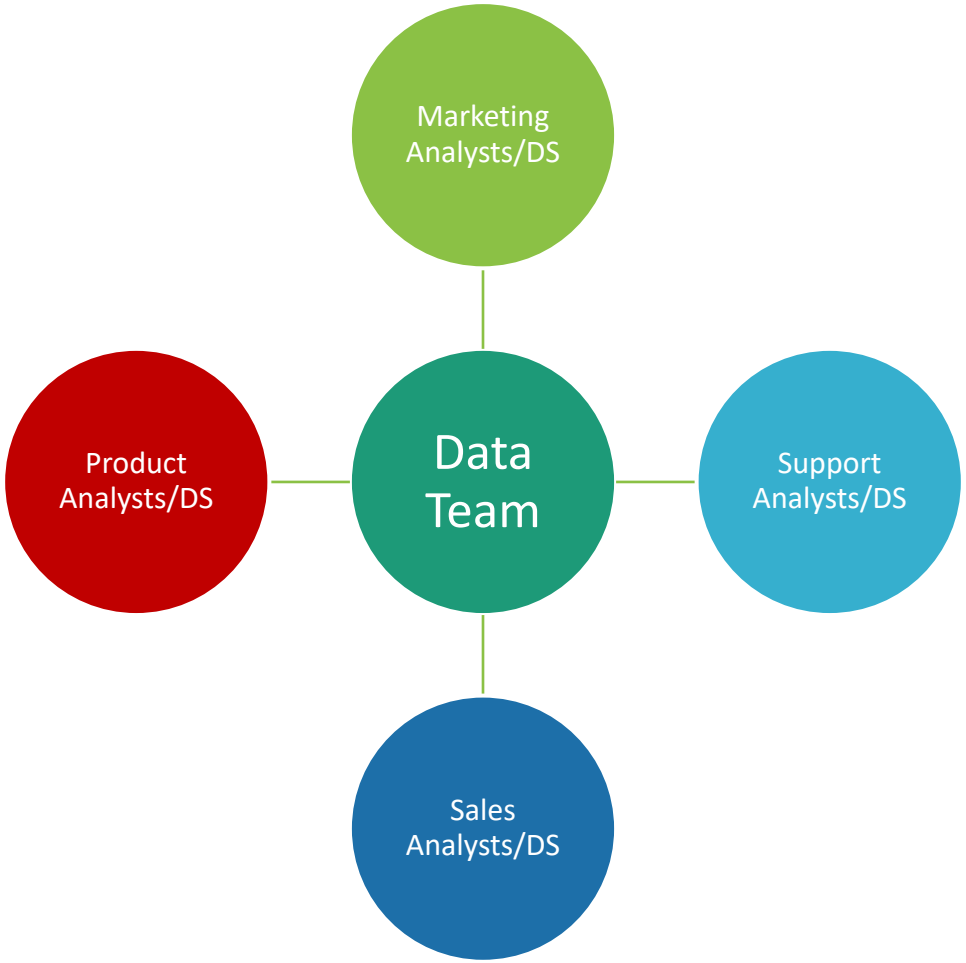
Democratize Analytics
… and data

Organizational Structure Dilemma



Democratize Analytics

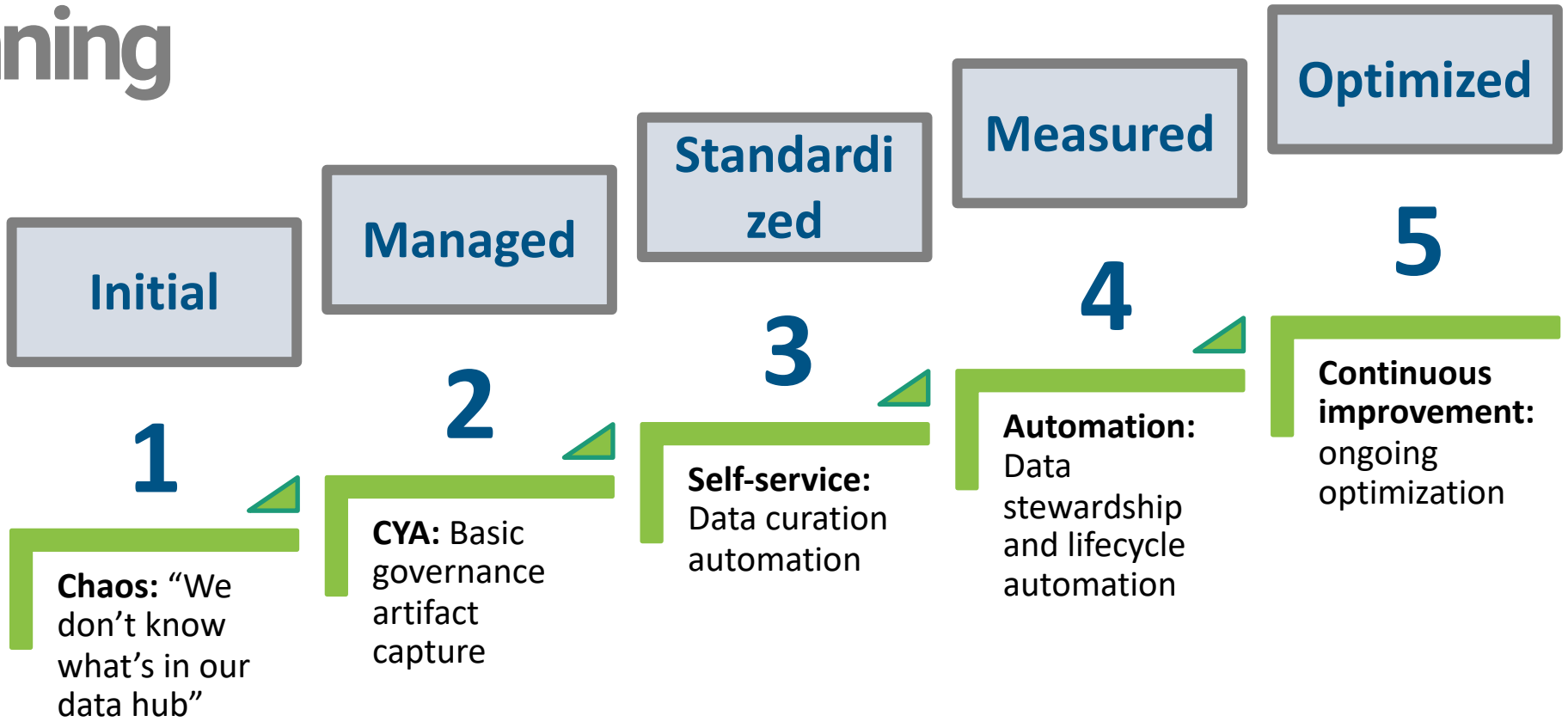
… and data



Hud-and-spoke Structure

Govern, from the beginning

Govern, from the beginning



5 keys to success

1. Build a data-driven culture
2. Develop the right team and skills
3. Be agile/lean in development
4. Leverage DevOps for production
5. Right-size data governance

Download Presentation

https://github.com/campossalex

Thanks!!!

Alex Campos | ac@cloudera.com

https://www.linkedin.com/in/campossalex/
@campossalex