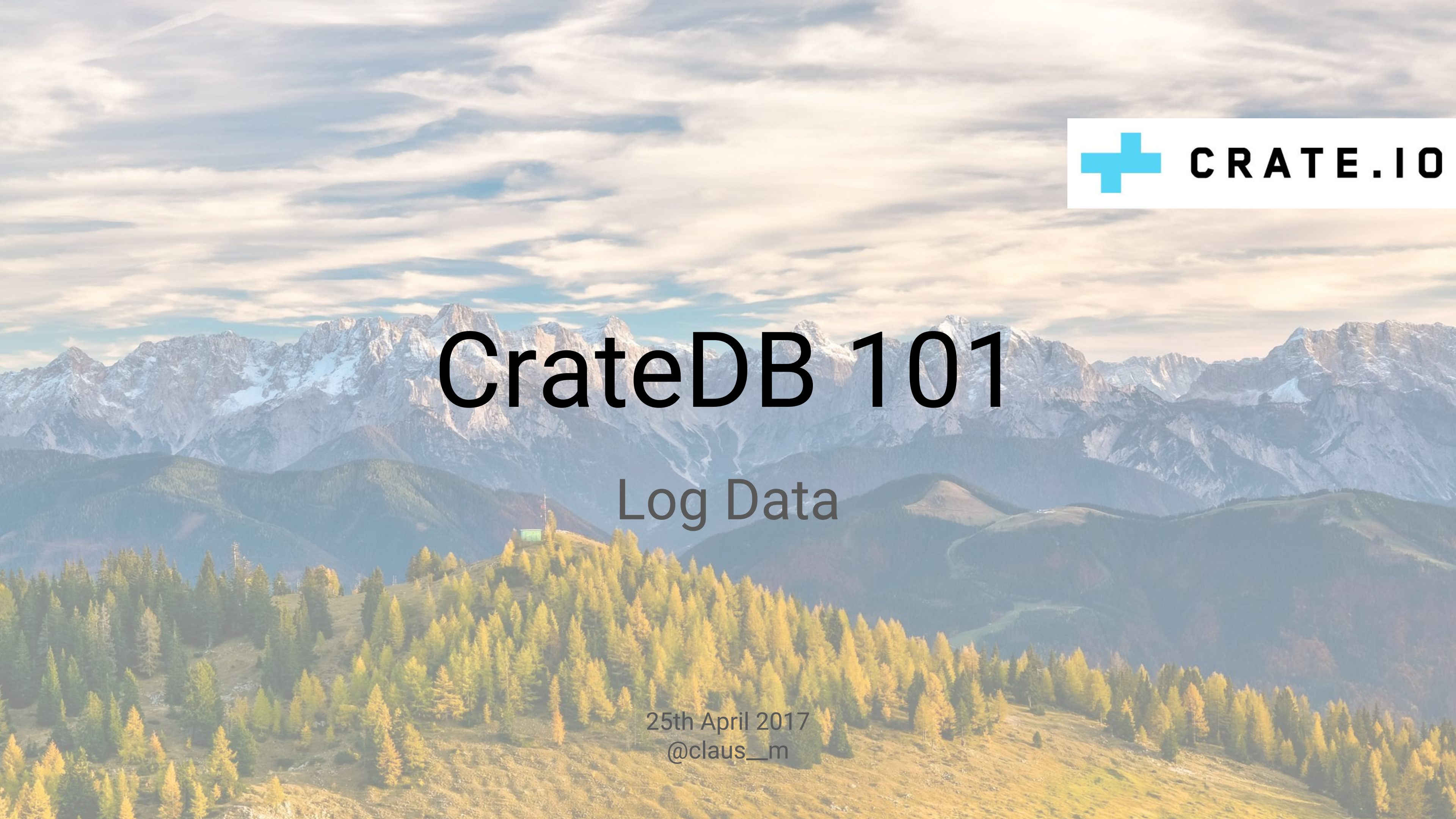


CRATE.IO

CrateDB 101

Log Data

25th April 2017
@claus__m

About

~2yrs at Crate.io
DevRel/Field Engineering/Support/ Integrations/…

Offices
San Francisco, Berlin, Dornbirn (AT)

Talk to me about
Rust, Raspberry Pis, Tech!

@claus__m

Agenda

About machine data
Why is it special?

CrateDB fundamentals
A deep-ish dive

Labs: Log analysis with CrateDB
Fluentd, CrateDB, Grafana

Wrap up
Next steps, more webinars!

CRATE.IO

@claus__m

git clone

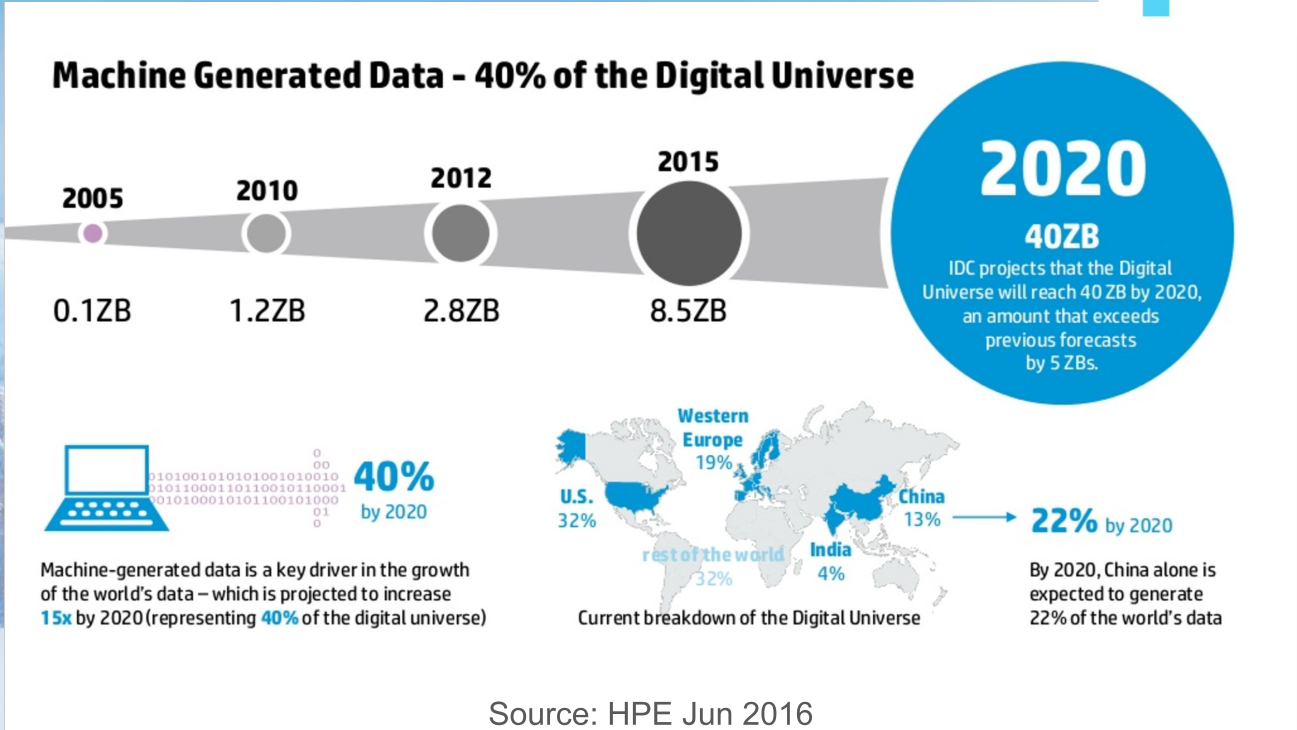
CRATE.IO

Find all files here

https://github.com/crate/webinar.101

CRATE.IO

Machine Data



Machine Generated Data - 40% of the Digital Universe

IDC projects that the Digital Universe will reach 40 ZB by 2020, an amount that exceeds previous forecasts by 5 ZBs.

40% by 2020

Machine-generated data is a key driver in the growth of the world's data – which is projected to increase **15x** by 2020 (representing **40%** of the digital universe)

Current breakdown of the Digital Universe

22% by 2020

By 2020, China alone is expected to generate 22% of the world's data

Source: HPE Jun 2016
http://www.slideshare.net/penumuru/harness-the-power-of-big-data-with-oracle-63438438/9

Machine Data Characteristics

CRATE.IO

Millions of data points/second
Streaming in from sensors, devices, logs, etc.

Data diversity
Structured & unstructured JSON, Blobs

Real-time query performance
Monitoring & alerting

Complex queries of big data volumes
With Terabytes of historic data

Growth
Adding sources often means exponential growth

Machine Data

Internet of Things
Sensors, cameras, ...

Wearables, Gadgets
Location data, interaction data, ...

Logs & Monitoring data
Component health monitoring, access logs, ...

Industry 4.0, Digitization
Production line insights, automation, ...

Vehicles
Location data, health data, ...

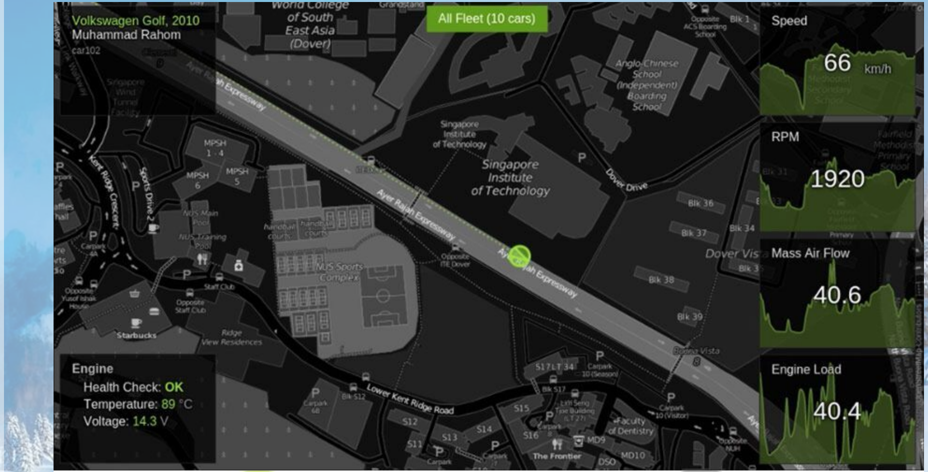
CRATE.IO

Clickdrive.io

Fleet management & vehicle tracking
Vehicle health and tracking data

High ingest rate
2,000 data points per car, per second

In-depth & real-time analysis
Predictive maintenance, accident reconstruction, route/driver efficiency



Volkswagen Golf, 2010
Muhammad Rahom
car102

All Fleet (10 cars)

Speed: 66 km/h
RPM: 1920
Mass Air Flow: 40.6
Engine Load: 40.4

Engine
Health Check: OK
Temperature: 89 °C
Voltage: 14.3 V

Ayer Rajah Expressway
Singapore Institute of Technology
NUS Sports Complex
Lower Kent Ridge Road
Dover Drive
Anglo-Chinese School (Independent) Boarding School
World College of South East Asia (Dover)

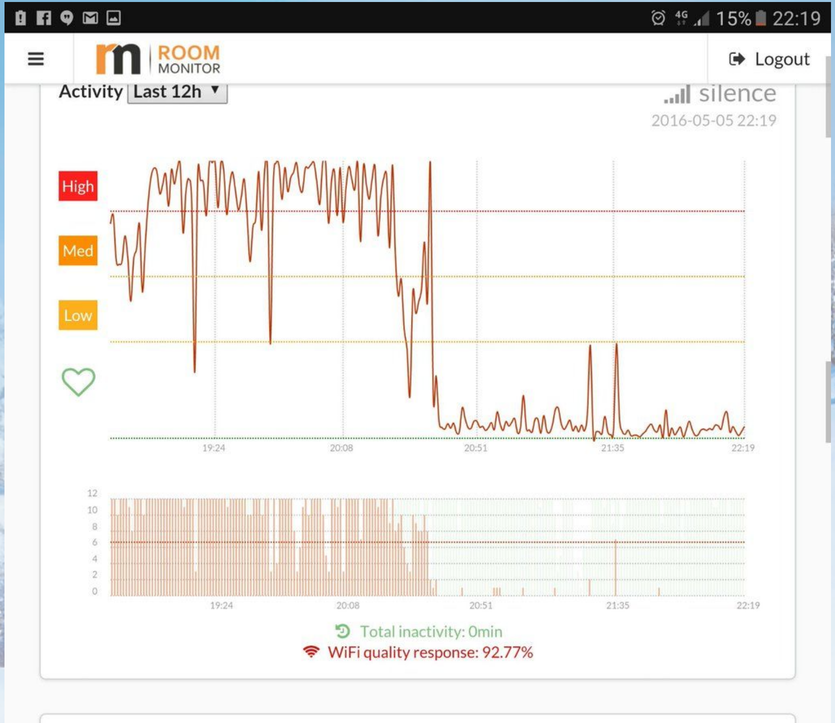
Roomonitor

Smart apartments

Monitoring & control climate, occupancy, noise, access

Better efficiency, safer environment

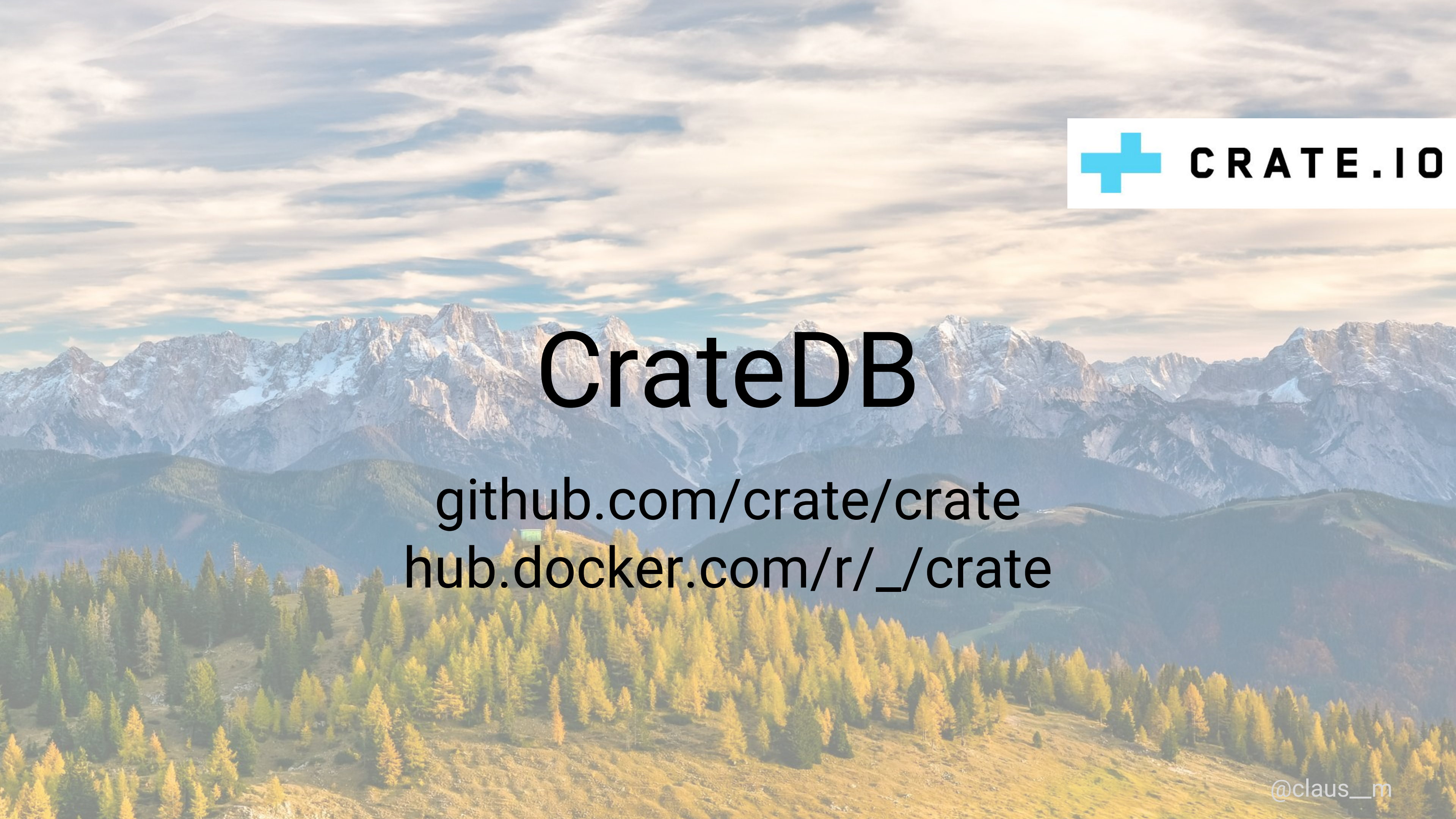
Alerts: AC/heating on with window open, noisy neighbors, ...



2016-05-05 22:19

Total inactivity: 0min

WiFi quality response: 92.77%



CrateDB

github.com/crate/crate

hub.docker.com/r/_/crate

CrateDB

Shared nothing
All nodes are equal

Partitioning, auto-sharding & replication
Transparent to the user

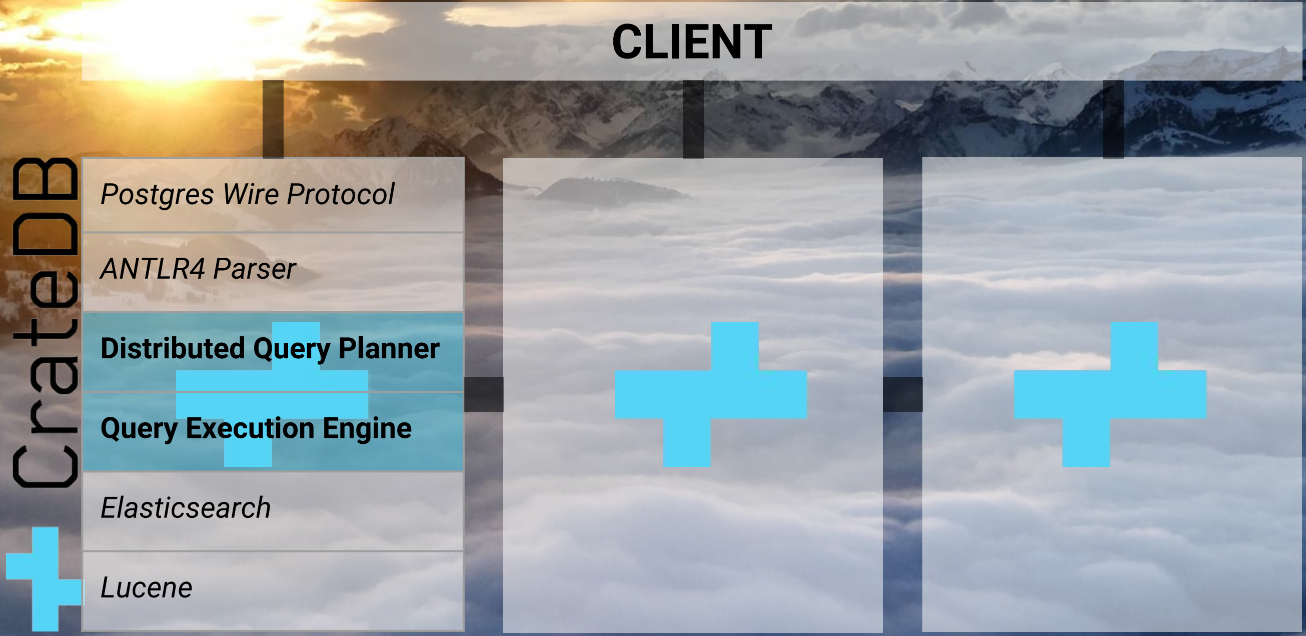
Multi model: Structured & unstructured
Search, queries, aggregates, joins

SQL
SQL!

CRATE.IO

@claus__m

CRATE.IO



CLIENT

CrateDB

Postgres Wire Protocol

ANTLR4 Parser

Distributed Query Planner

Query Execution Engine

Elasticsearch

Lucene

@claus__m

CrateDB Fundamentals

Disk-based index with in-memory caching
Fast and efficient OS caching

Shards: Units of data
Concurrency by distributing shards

Distributed query execution engine
“Push down” queries

CRATE.IO

@claus__m

Lucene: CrateDB Shards

Documents
Rows with expansible columns

Fulltext search: Inverted index
Analyse, tokenise, and search

Compression
LZ4 compression of fields

Field cache
Columnar storage

Data types
Java types: long, int, string, ...

Clustering: Shard Management

On-disk storage
Multiple files

Replication
Copies of initial files (primaries)

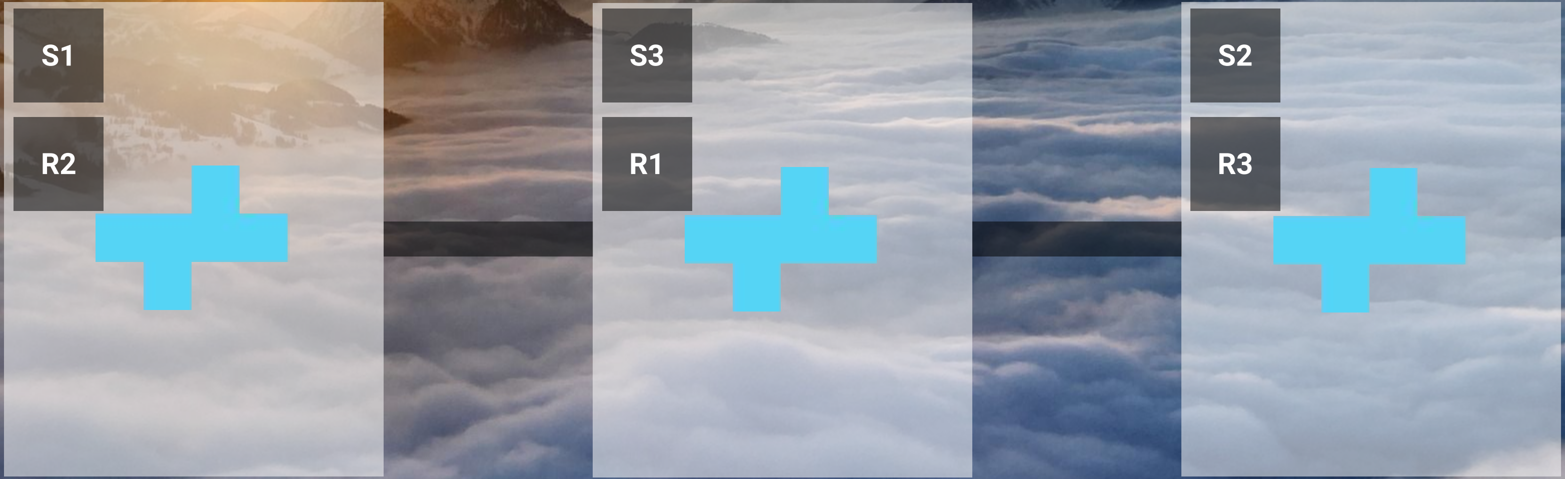
Distribution
Shuffle around shards (primaries & replicas)

Cluster state
Stores shard locations based on _id

CRATE.IO

CRATE.IO

CrateDB



S1
R2

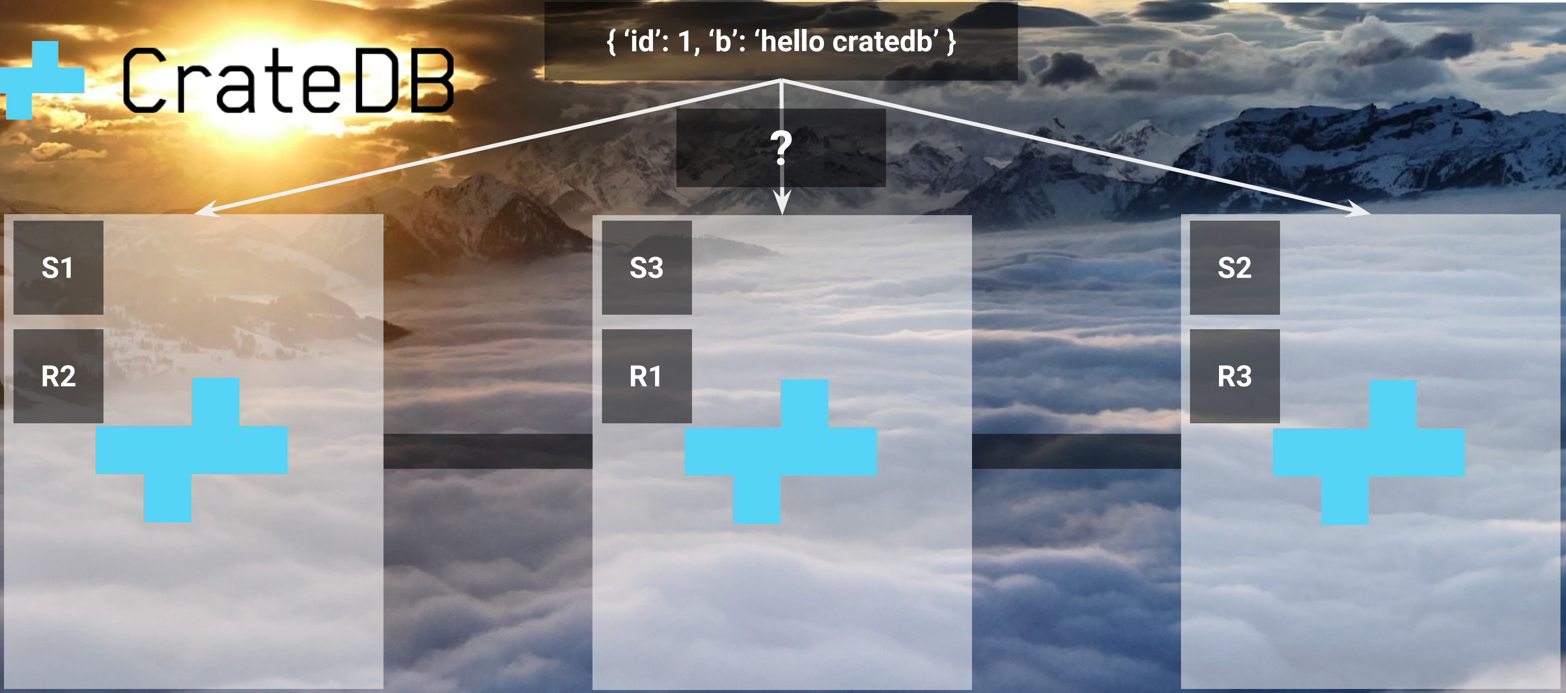
S3
R1

S2
R3

@claus__m

CRATE.IO

CrateDB



{ ‘id’: 1, ‘b’: ‘hello cratedb’ }

?

S1

R2

S3

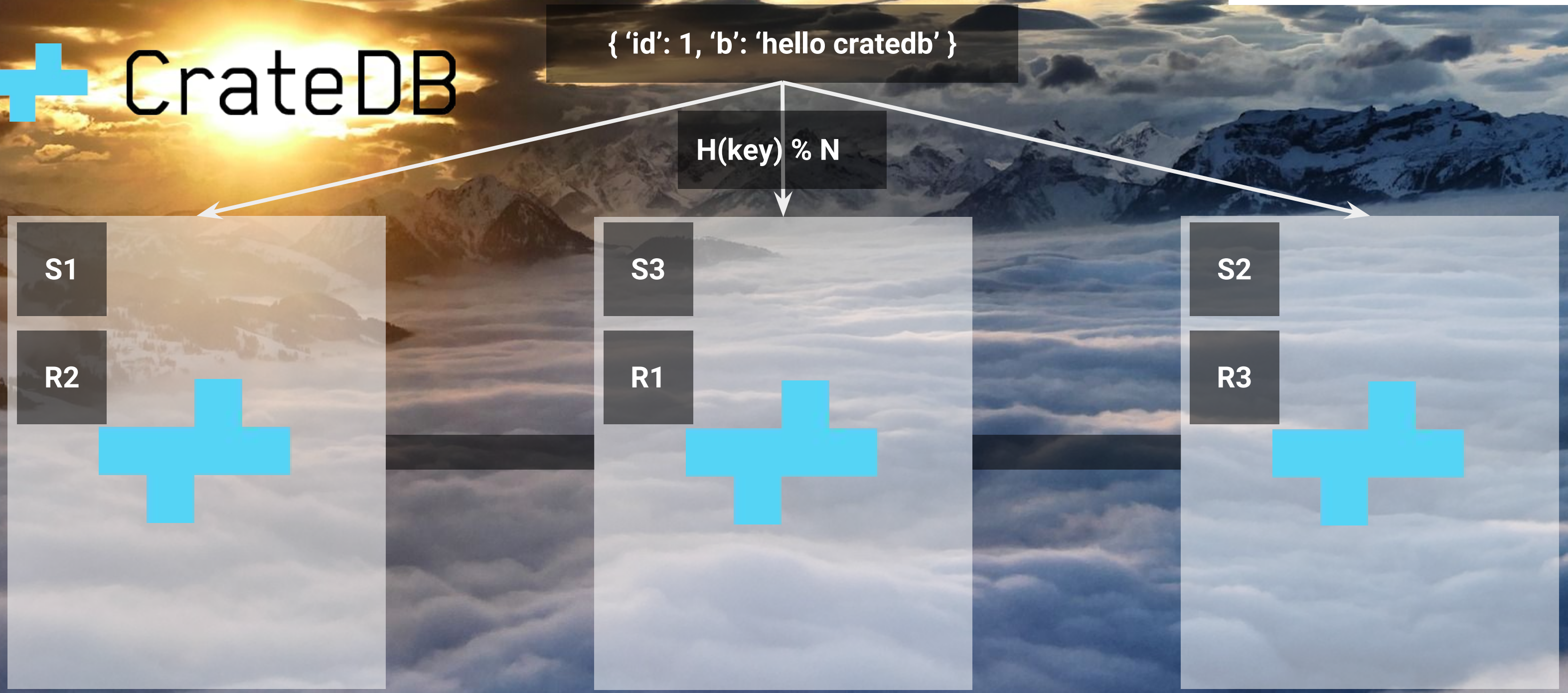
R1

S2

R3

CRATE.IO

CrateDB



{ ‘id’: 1, ‘b’: ‘hello cratedb’ }

H(key) % N

S1

R2

S3

R1

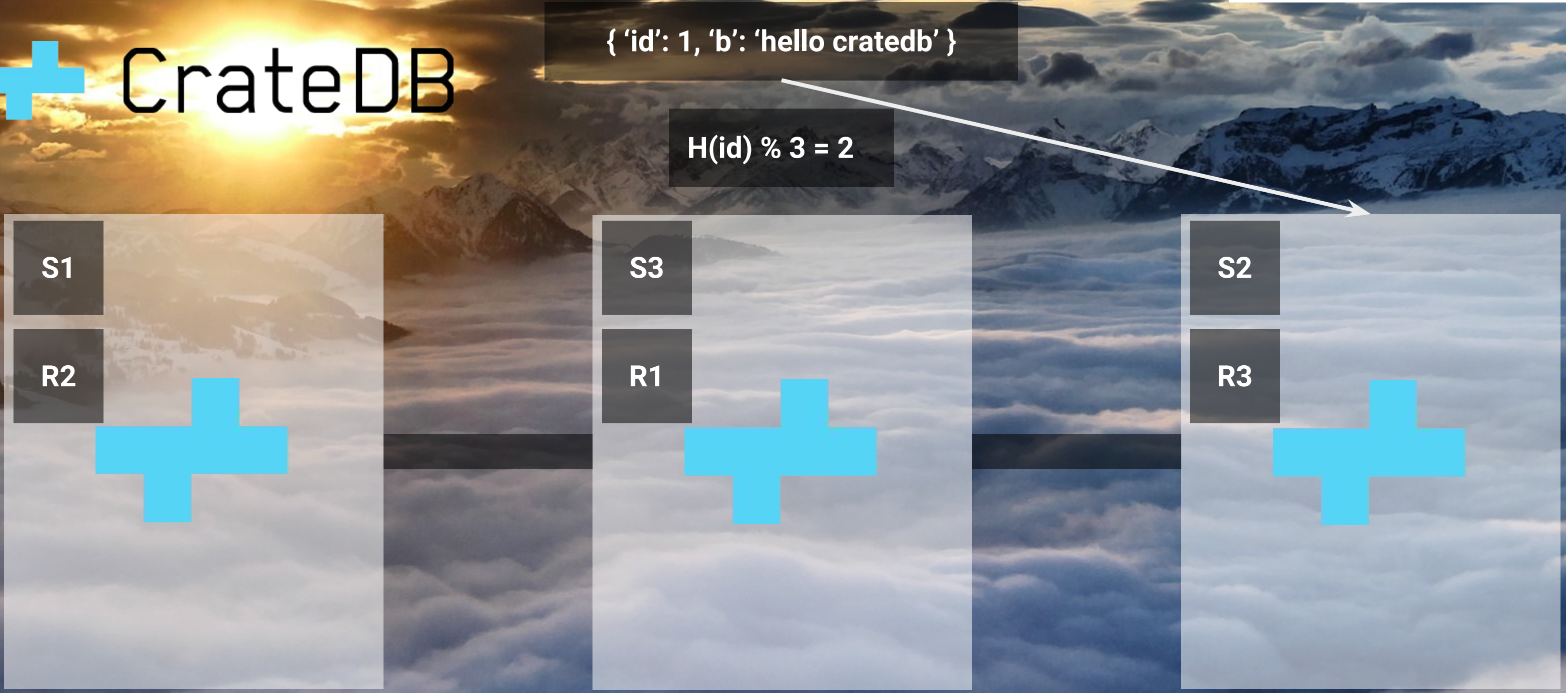
S2

R3

@claus__m

CRATE.IO

CrateDB



{ ‘id’: 1, ‘b’: ‘hello cratedb’ }

H(id) % 3 = 2

@claus__m

Log Data Use Cases

CRATE.IO

Marketing
Popularity, impact of sites

Audit trails
Who looked at what?

Intrusion detection
Hacked?

Service monitoring
DDoS attacks? 404s?

@claus__m

Log Data Characteristics

CRATE.IO

Text
Varying formats, vendor-specifc, …

High volume
Access logs, error logs - attacks?

Different sources
Apache, nginx, …

CrateDB for Log Data

CRATE.IO

Horizontal scalability
Scale as you grow

Reduced tech stack
Fewer moving parts

Fulltext via SQL
Powerful text analysis

Flexibility
Schema evolution built in

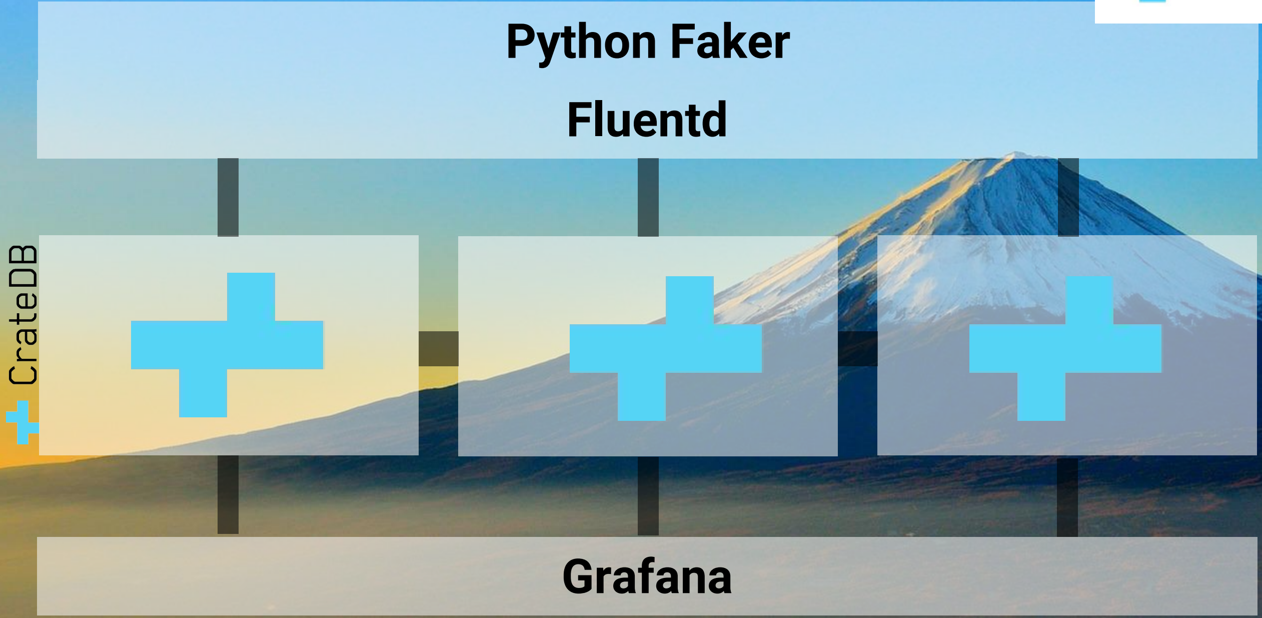
Built-in tools for logistics
Generated columns, partitioning, ...

CRATE.IO

Labs.

@claus__m

CRATE.IO

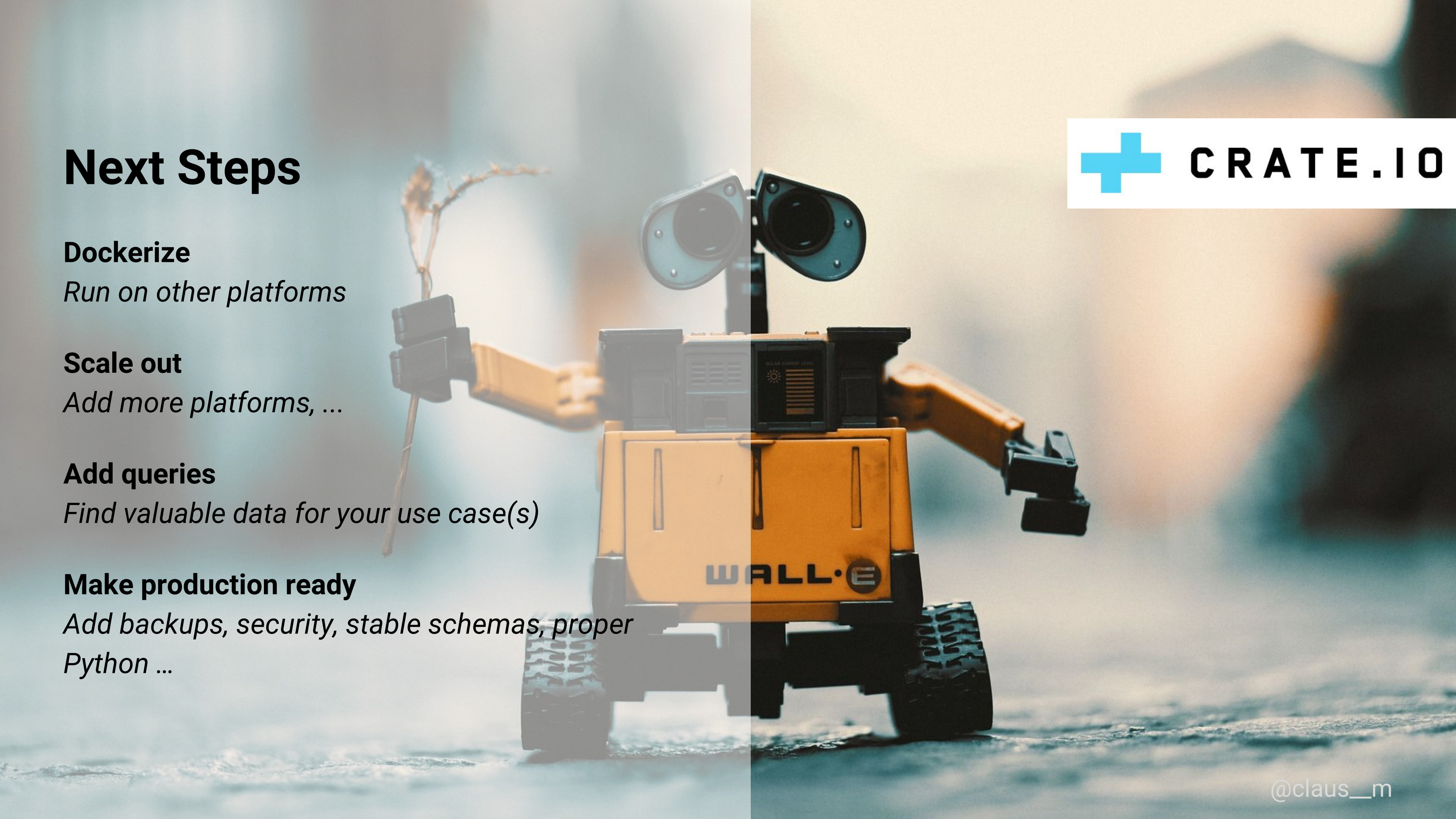

Python Faker

Fluentd

CrateDB

Grafana

@claus__m



CRATE.IO

Next Steps

Dockerize
Run on other platforms

Scale out
Add more platforms, ...

Add queries
Find valuable data for your use case(s)

Make production ready
Add backups, security, stable schemas, proper Python …

CRATE.IO

CrateDB for Log Data

SQL for all data
No (re)indexing required

Distributed by nature
Scale out as you go

High performance
Increase concurrency as you go

Powerful tools for text mining
Fulltext analysis via SQL

Links

CRATE.IO

https://github.com/crate

https://github.com/crate/webinar.101

https://crate.io

Thanks!

Follow us on twitter
@crateio *@claus__m*

Next webinar: Geospatial data, 2nd May