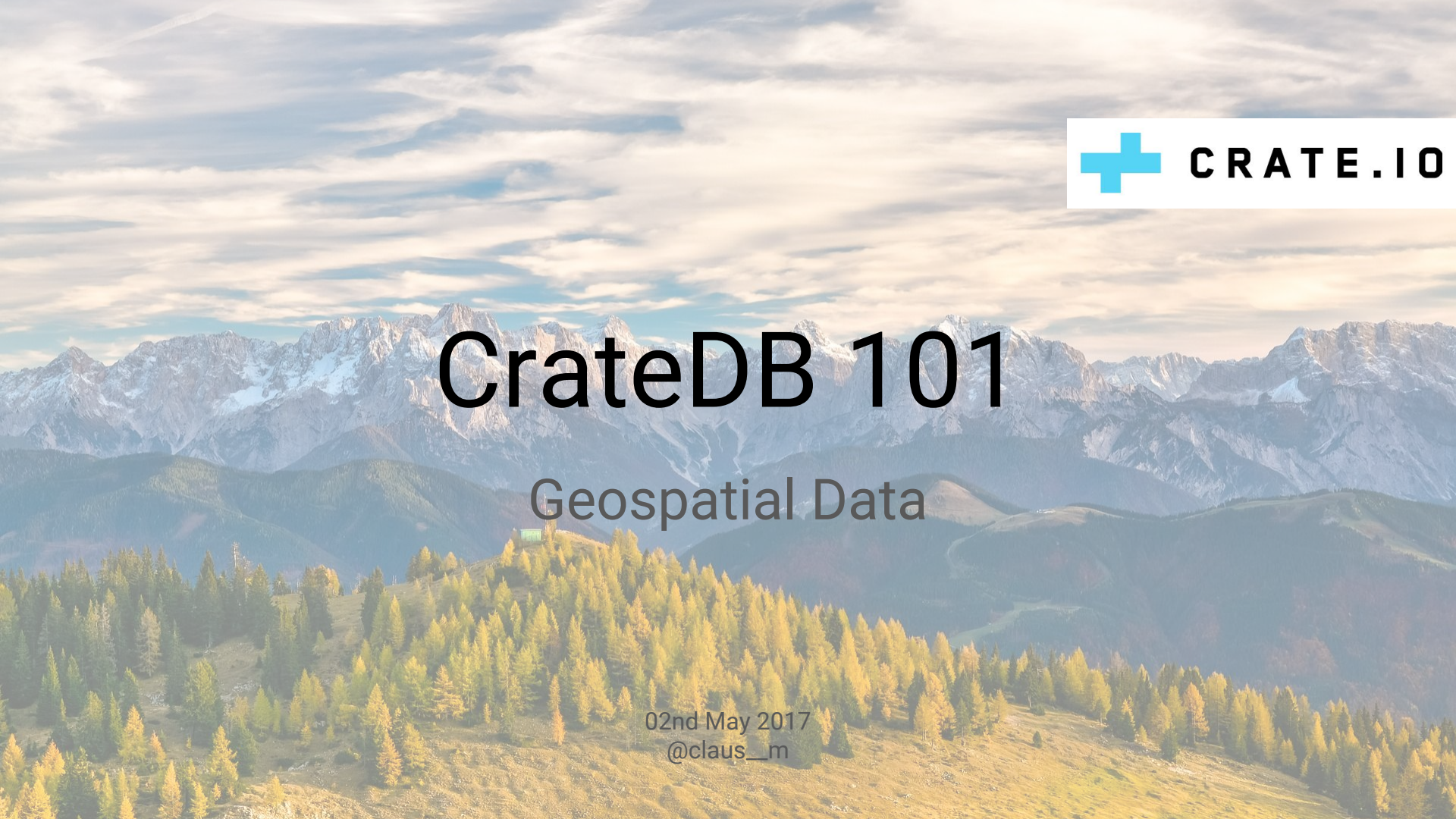


CRATE.IO

CrateDB 101

Geospatial Data

02nd May 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!

Agenda

About machine data
Why is it special?

CrateDB fundamentals
A deep-ish dive

Labs: NYC with CrateDB
A journey through some open data

Wrap up
Next steps!

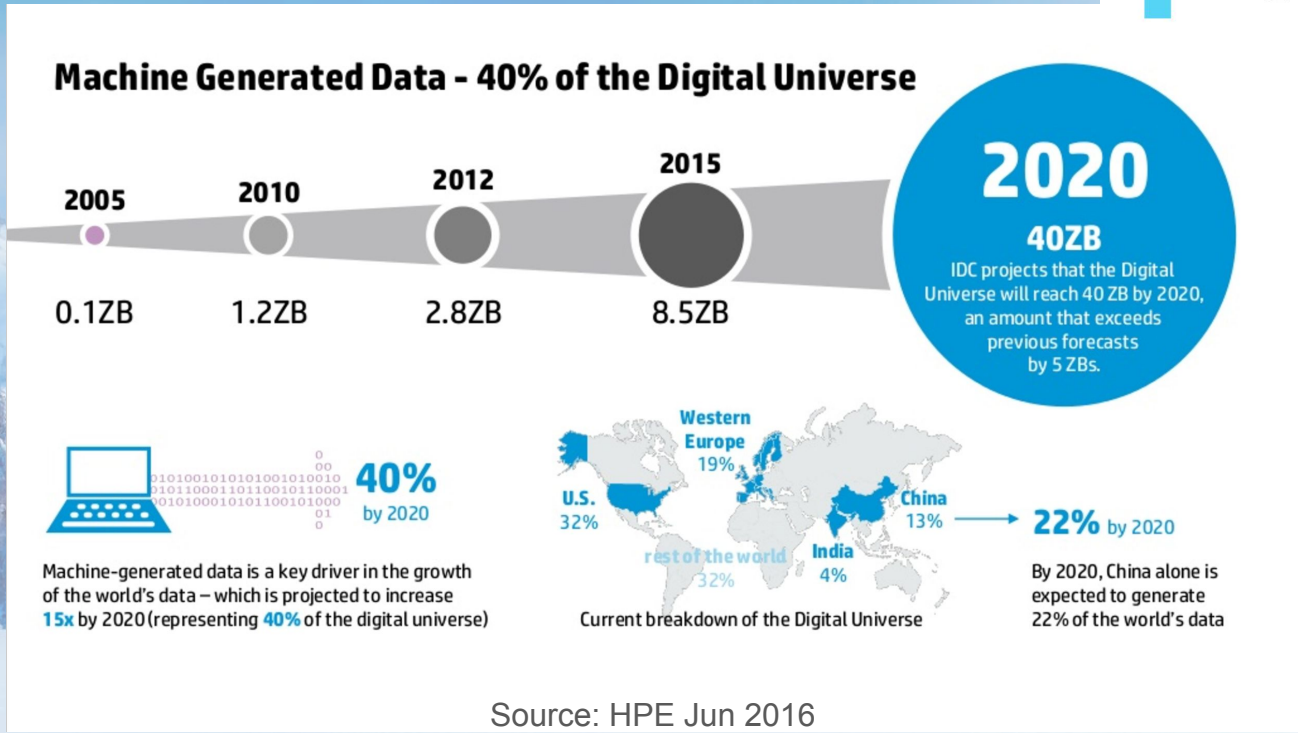
git clone

Find all files here

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


CRATE.IO
Machine Data
@claus__m

CRATE.IO



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

Machine Data Characteristics

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

CRATE.IO

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, ...

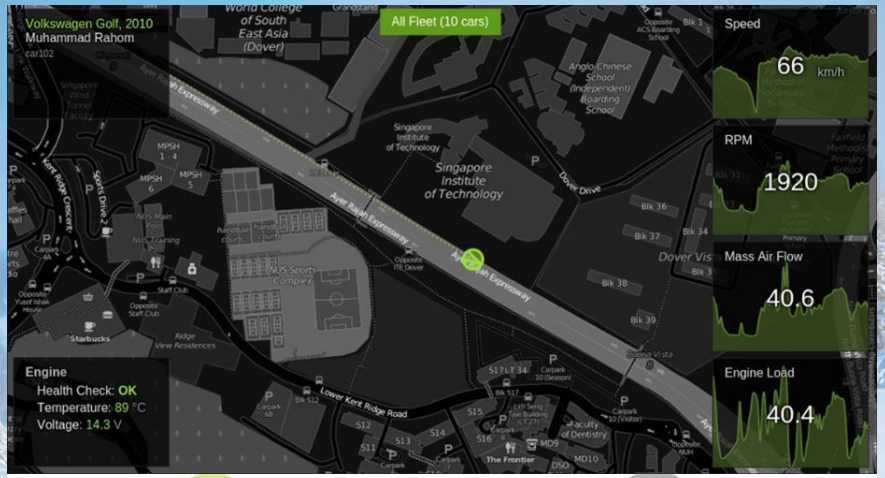
CRATE.IO

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



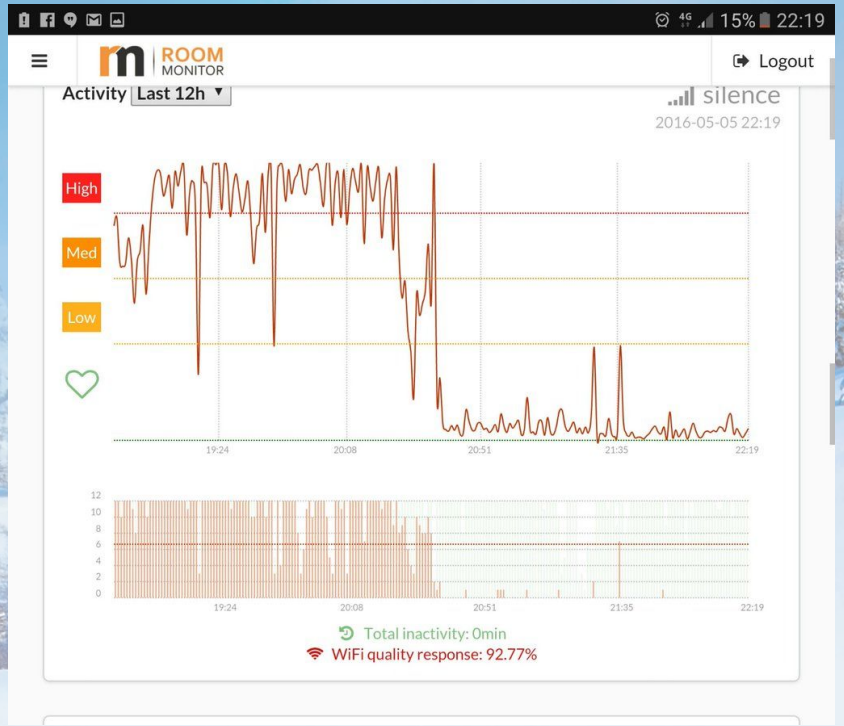
Roomonitor

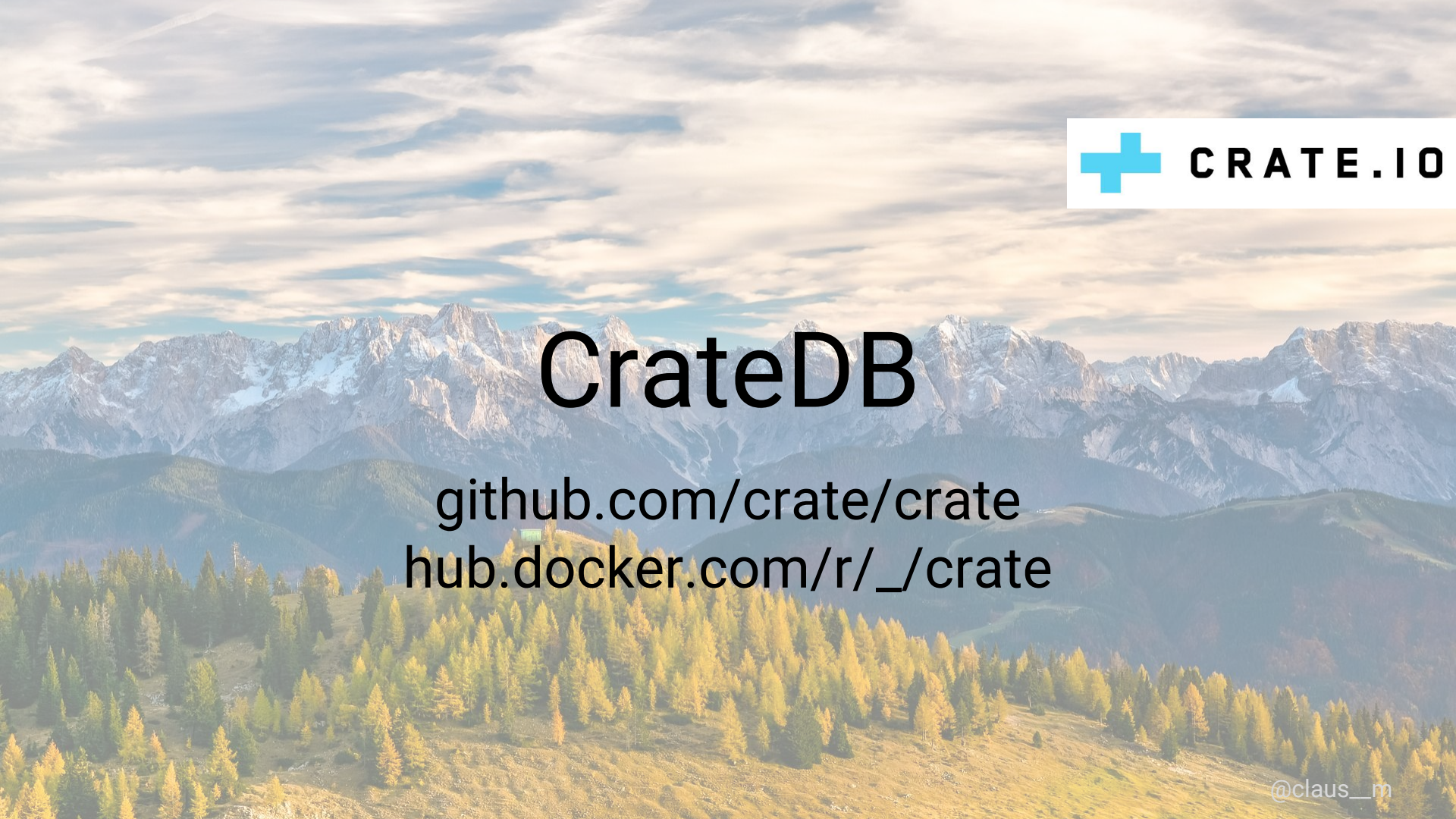
Smart apartments

Monitoring & control climate, occupancy, noise, access

Better efficiency, safer environment

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




CRATE.IO
CrateDB
github.com/crate/crate
hub.docker.com/r/_/crate
@claus__m

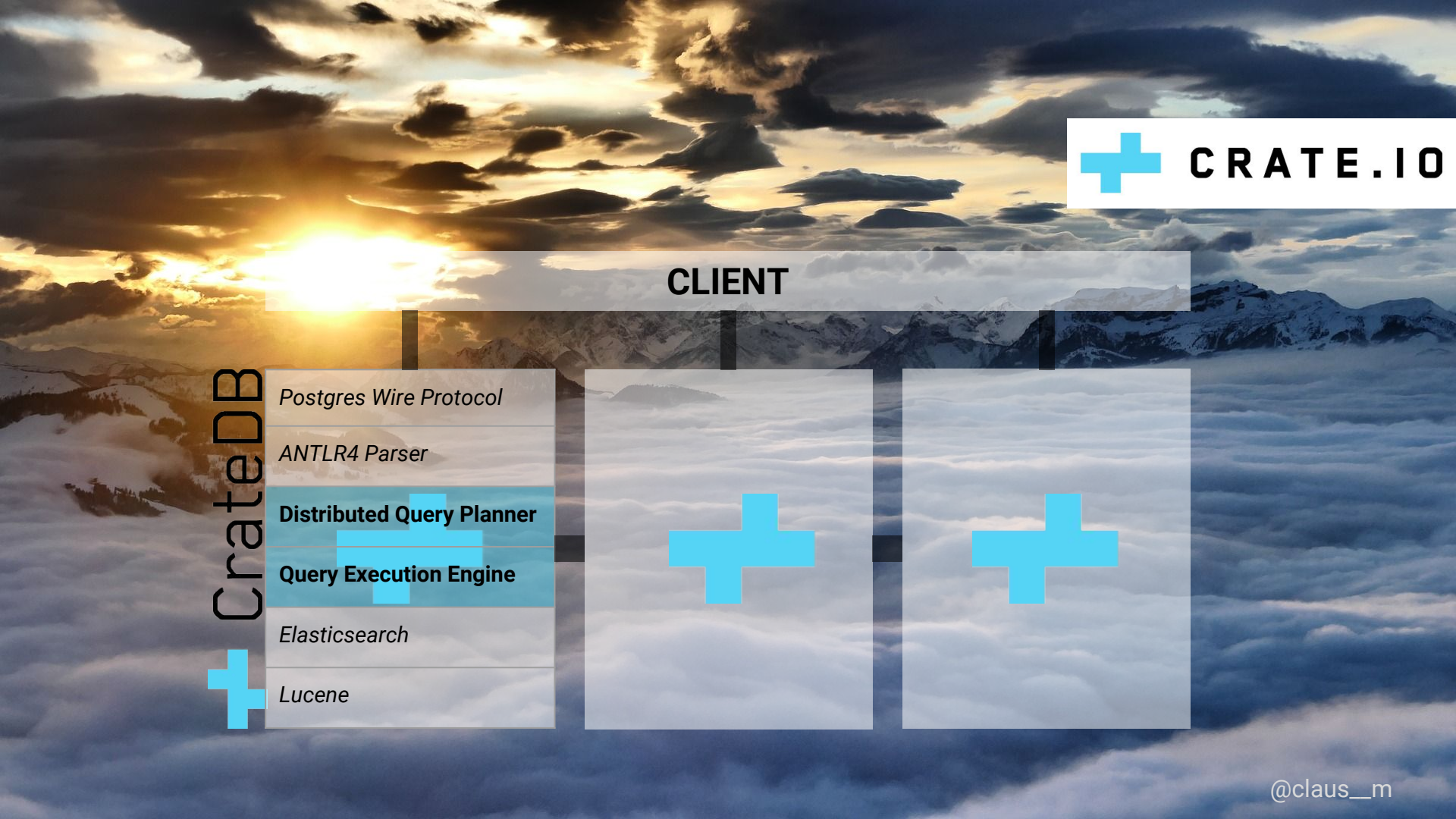
CrateDB

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
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

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, ...

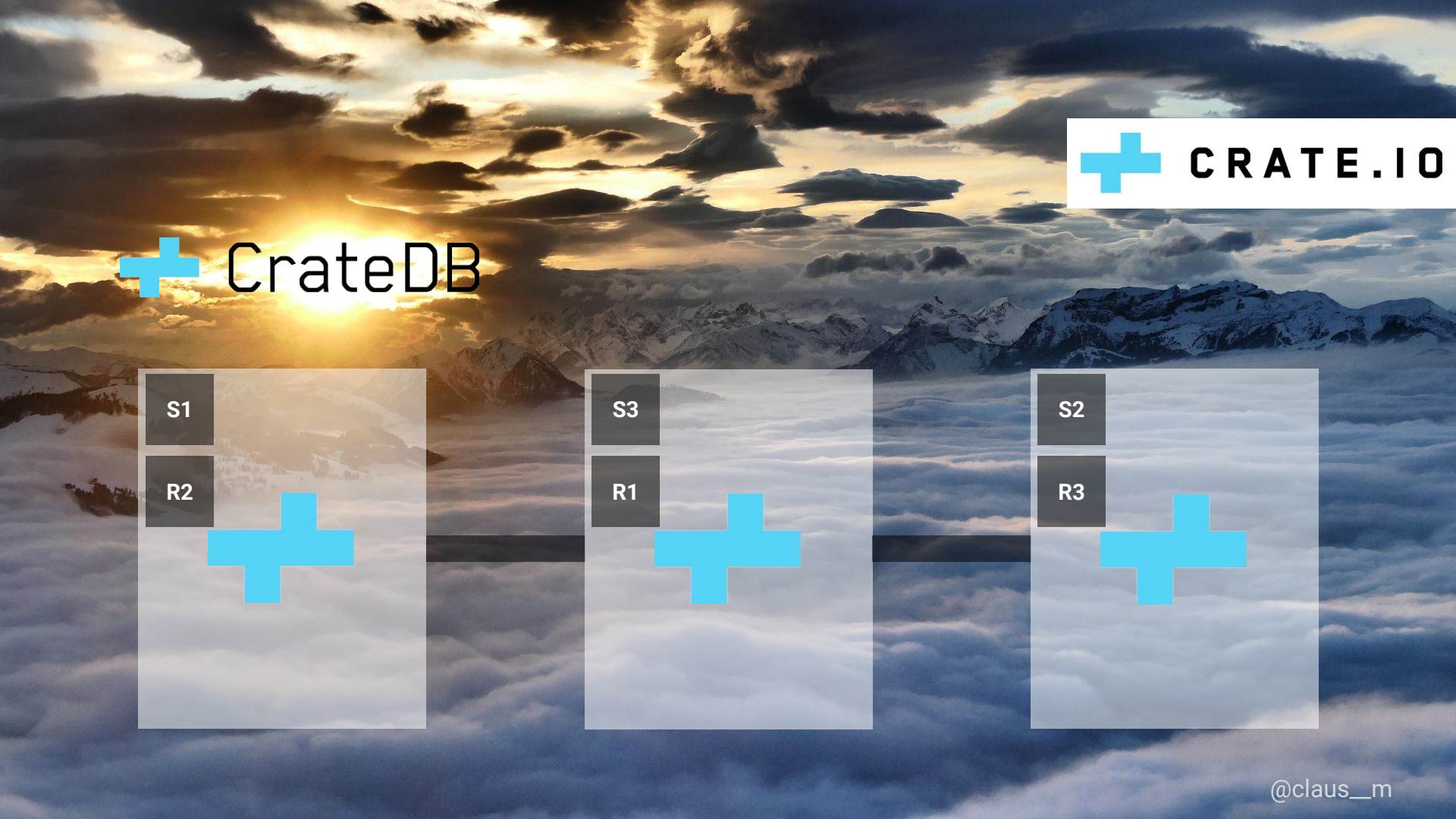
Clustering: Shard Management

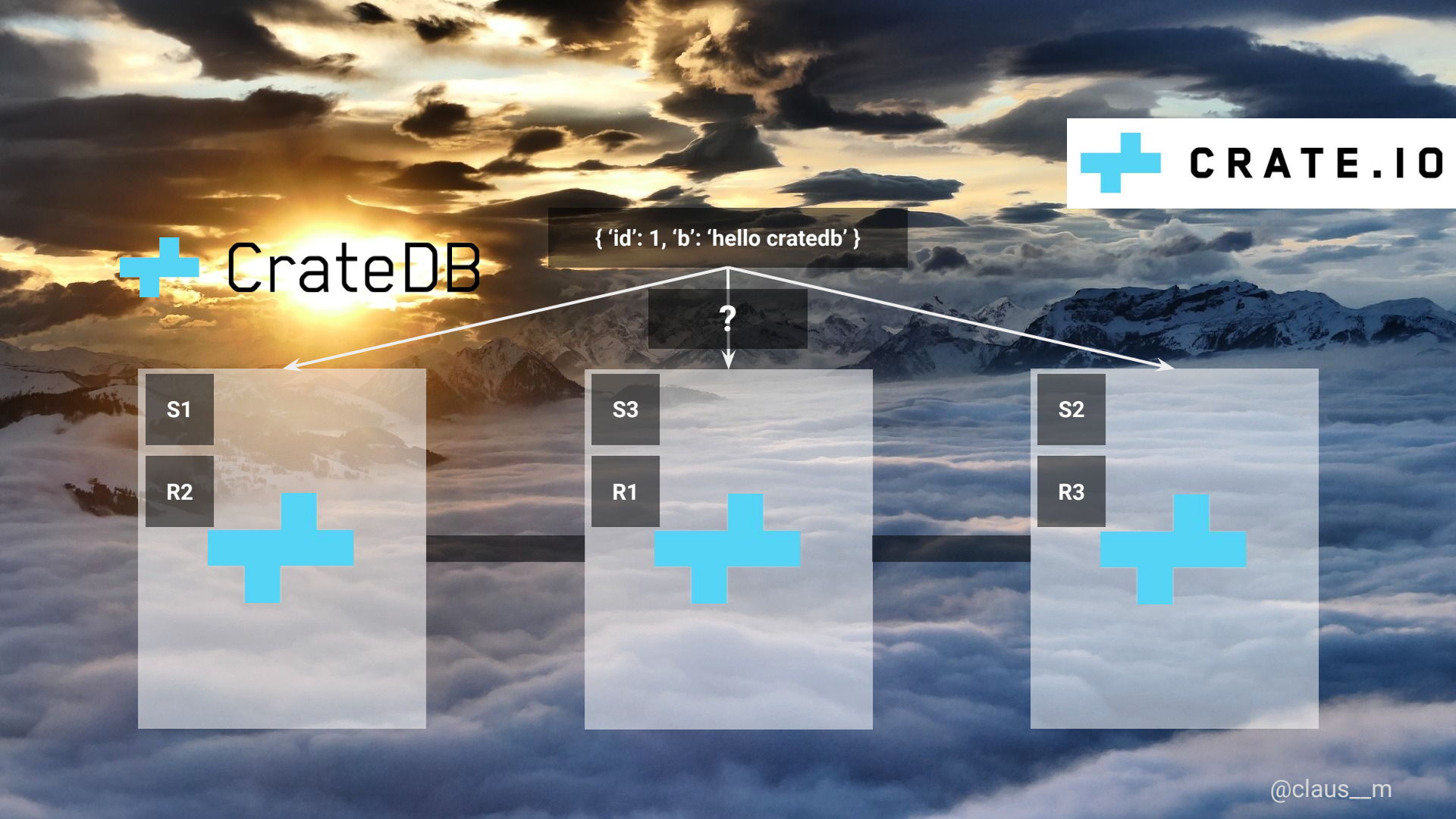
On-disk storage
Multiple files

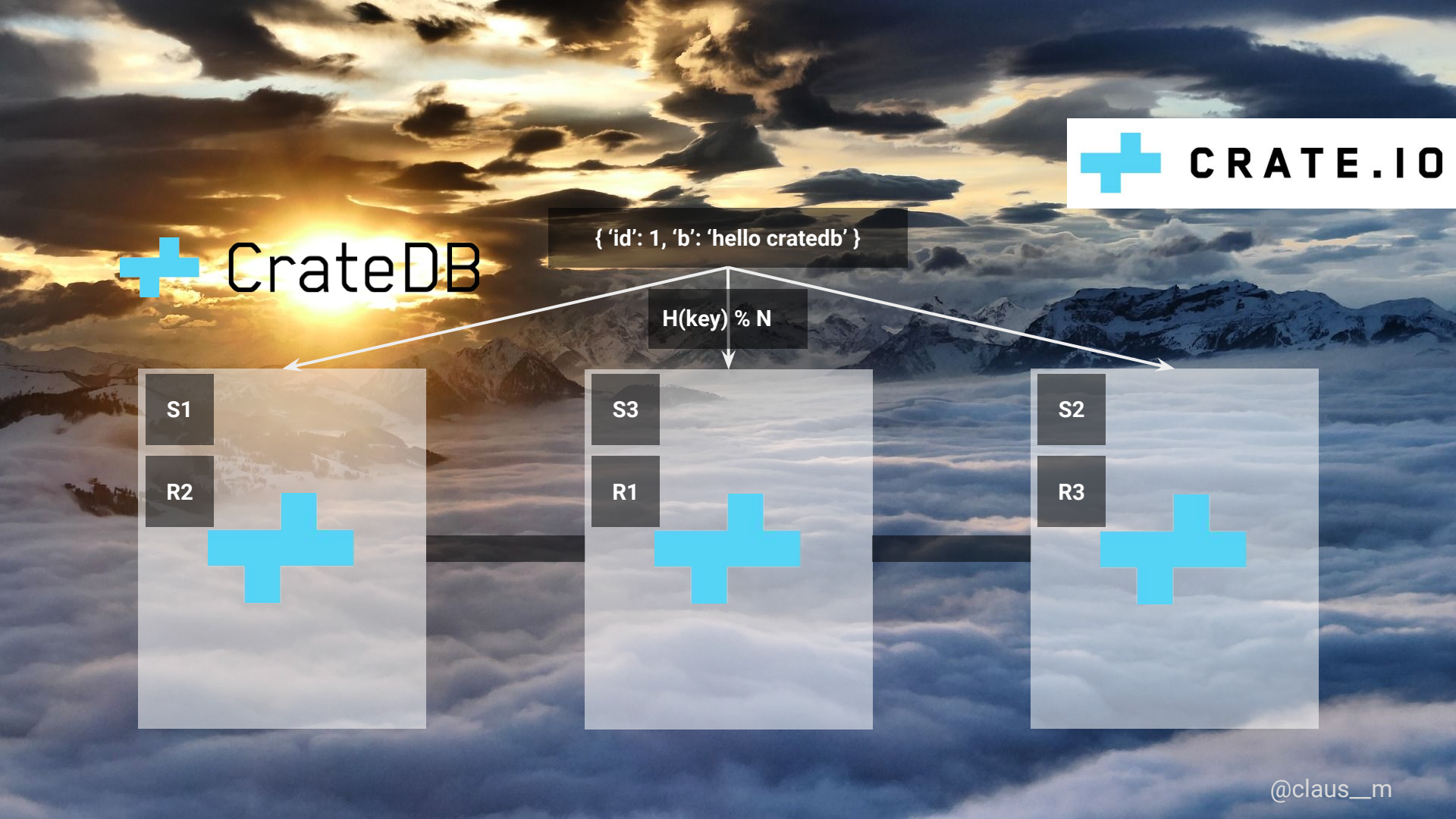
Replication
Copies of initial files (primaries)

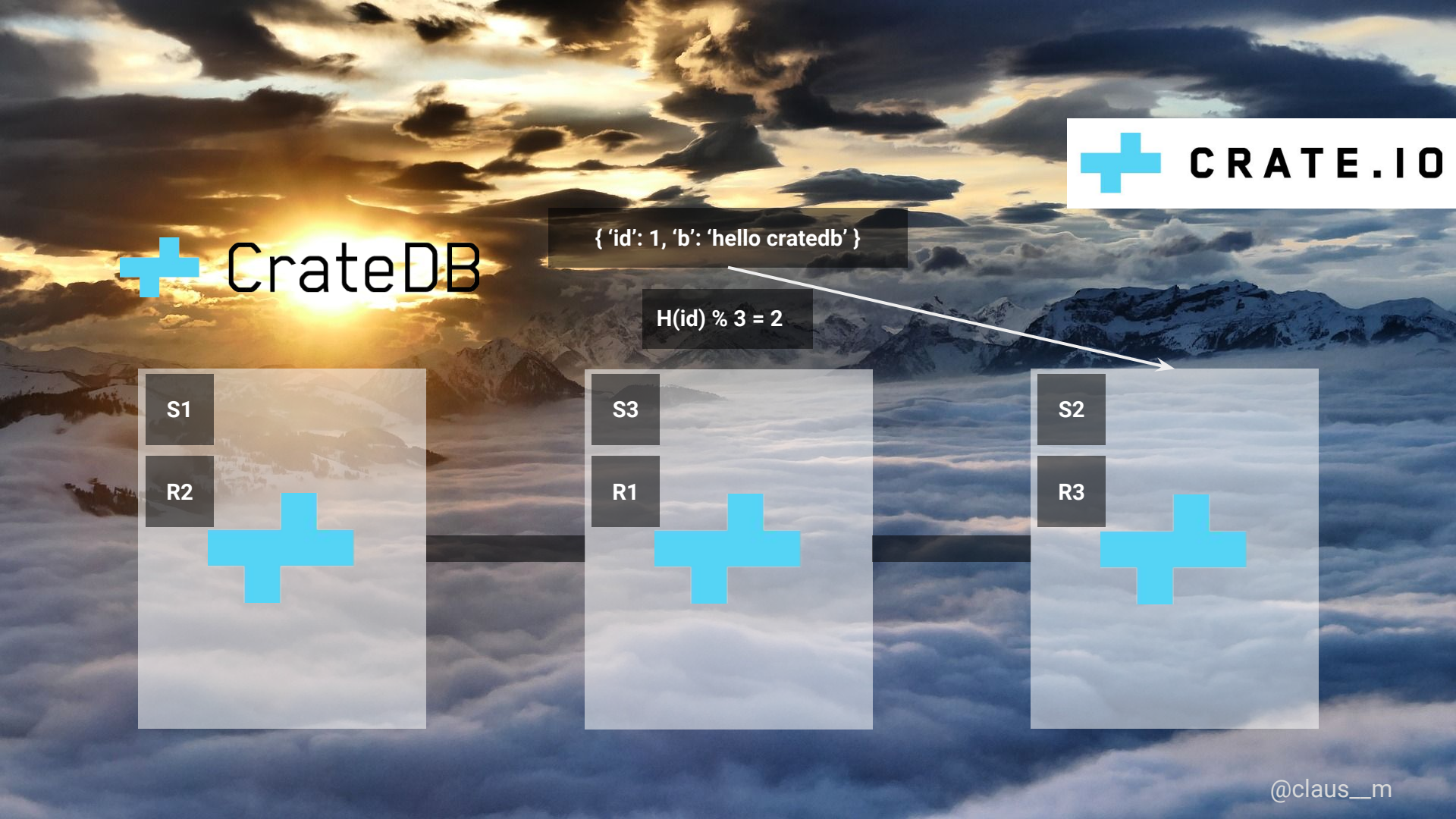
Distribution
Shuffle around shards (primaries & replicas)

Cluster state
Stores shard locations based on _id

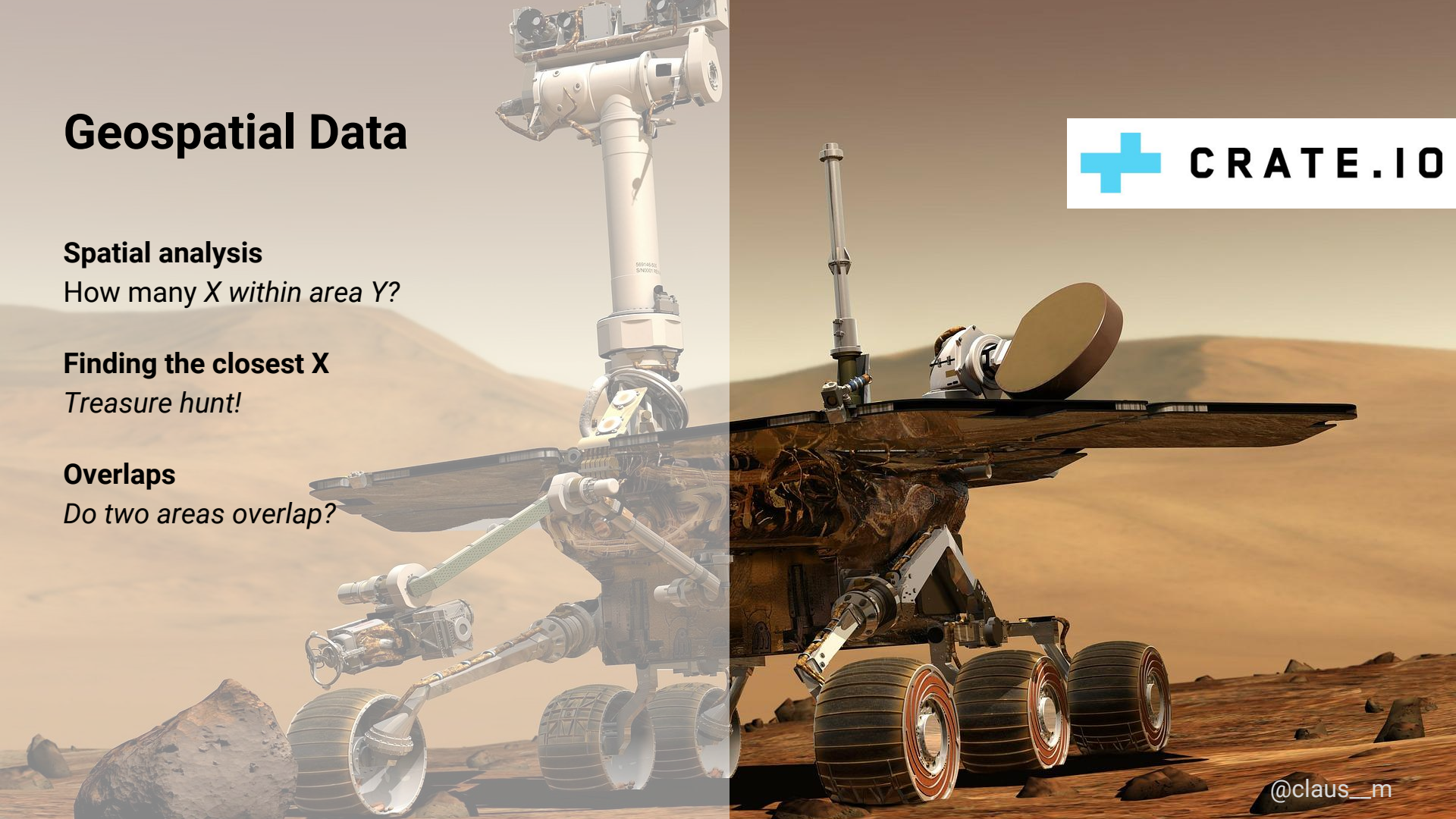

CRATE.IO
CrateDB
S1
R2
S3
R1
S2
R3
@claus__m


CRATE.IO
CrateDB
{ ‘id’: 1, ‘b’: ‘hello cratedb’ }
?
S1
R2
S3
R1
S2
R3
@claus__m


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
S1
R2
S3
R1
S2
R3
@claus__m

Geospatial Data



Spatial analysis
How many *X within area Y*?

Finding the closest X
Treasure hunt!

Overlaps
Do two areas overlap?

CRATE.IO

@claus_m

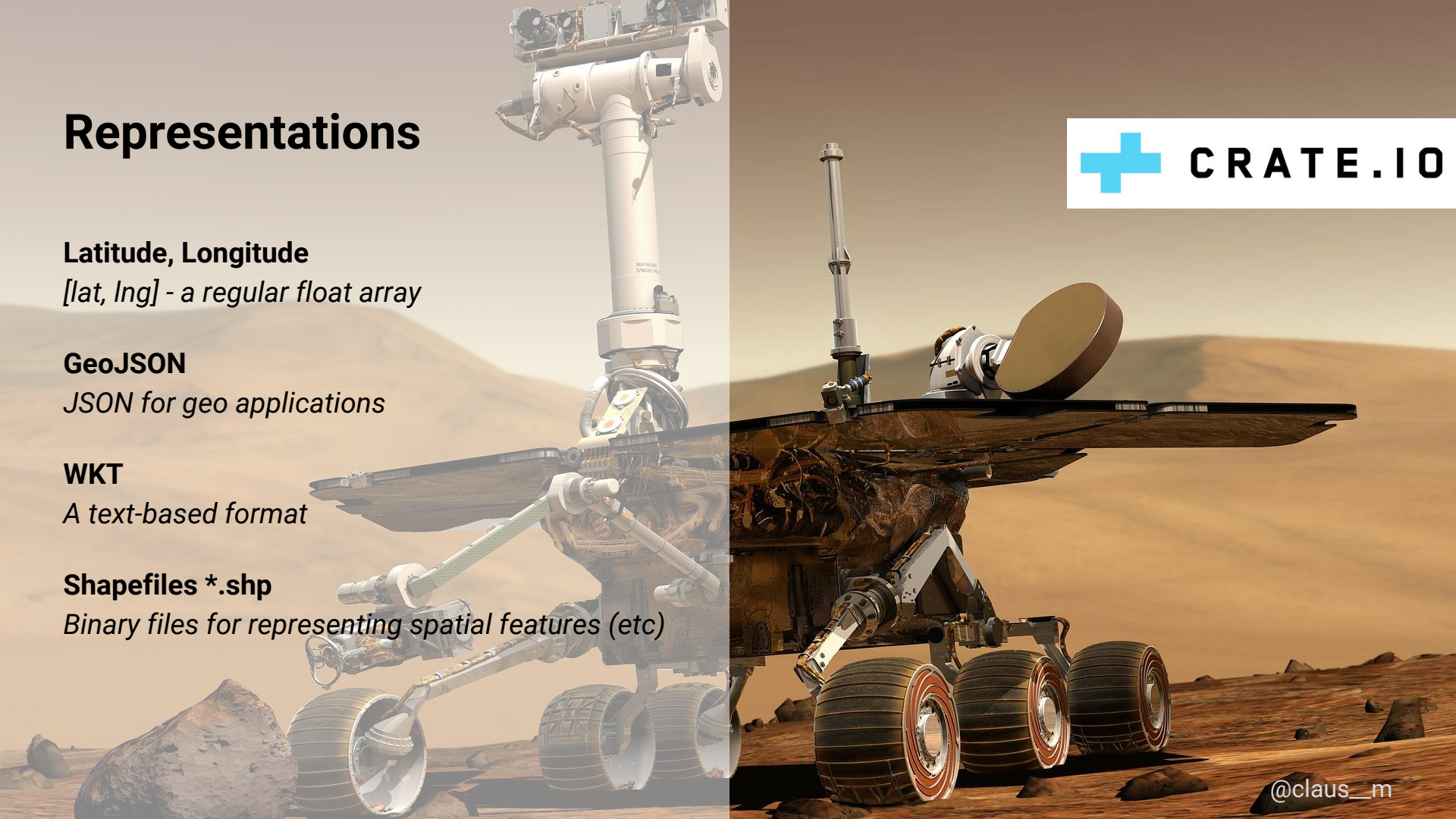
Representations

Latitude, Longitude
[lat, lng] - a regular float array

GeoJSON
JSON for geo applications

WKT
A text-based format

Shapefiles *.shp
Binary files for representing spatial features (etc)



CRATE.IO

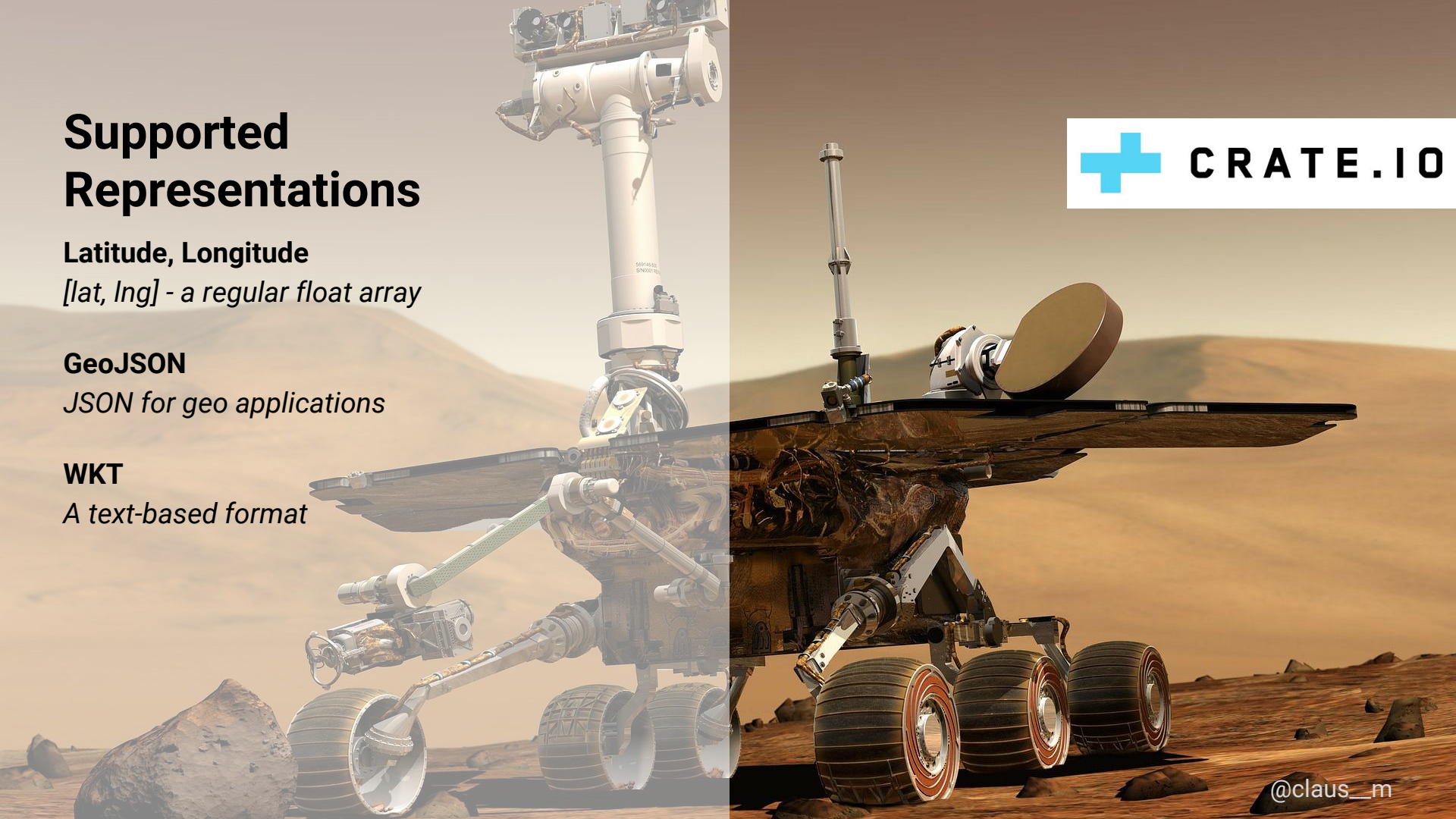
@claus_m

Supported Representations

Latitude, Longitude
[lat, lng] - a regular float array

GeoJSON
JSON for geo applications

WKT
A text-based format



CRATE.IO

CrateDB for Geospatial Data

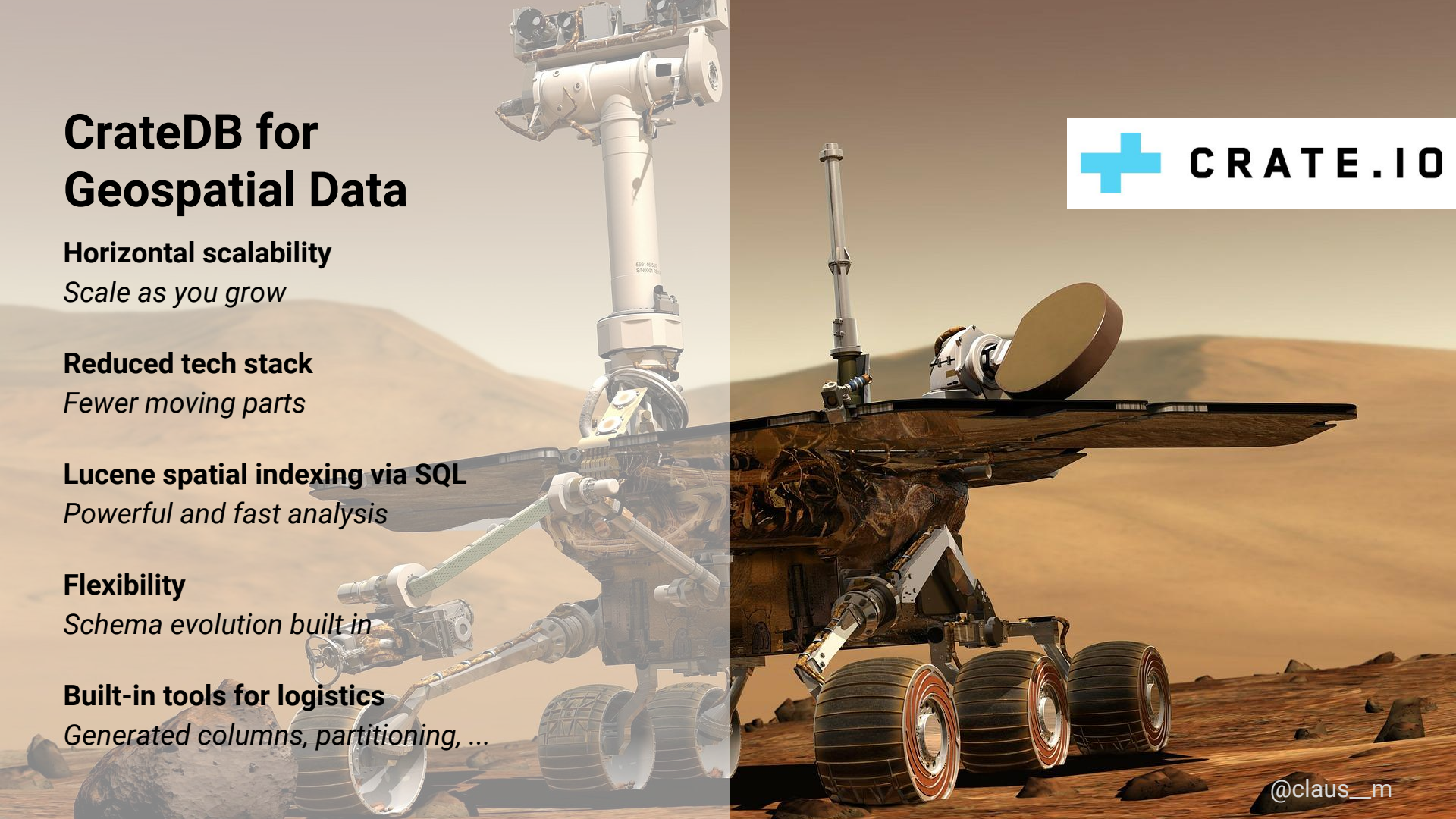
Horizontal scalability
Scale as you grow

Reduced tech stack
Fewer moving parts

Lucene spatial indexing via SQL
Powerful and fast analysis

Flexibility
Schema evolution built in

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



CRATE.IO

@claus_m


CRATE.IO
Labs.
@claus__m

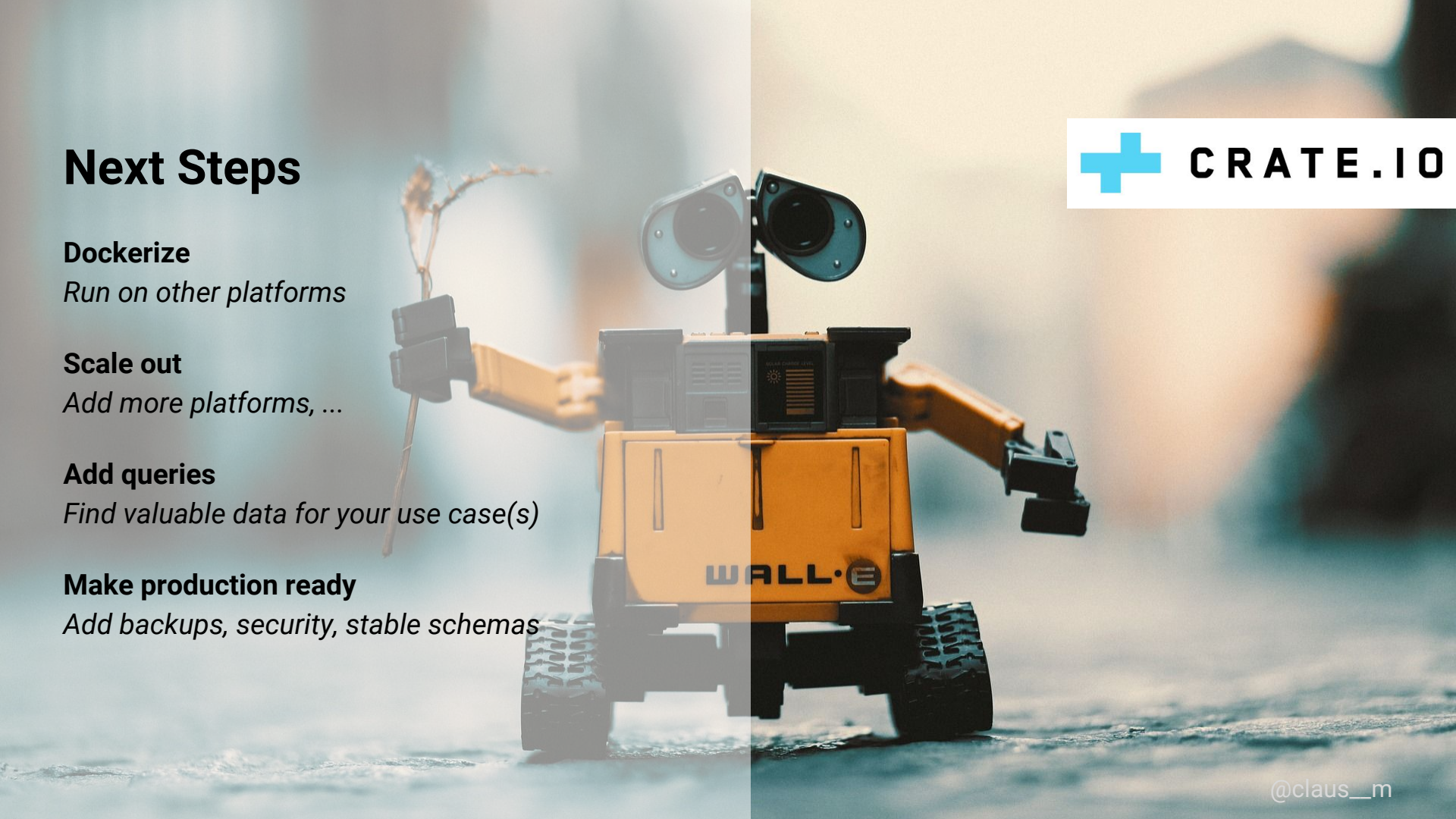
Next Steps

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



CRATE.IO

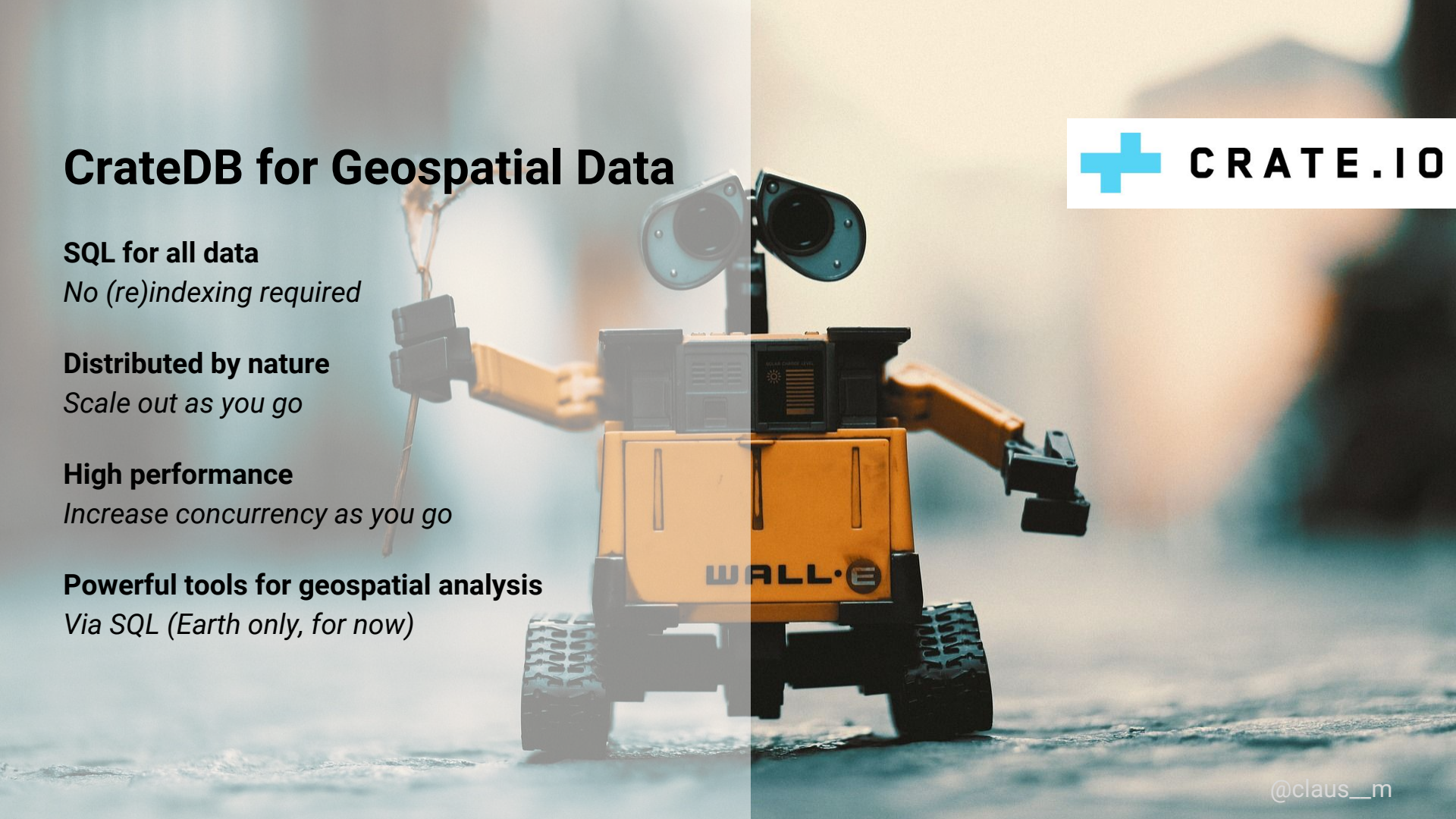
CrateDB for Geospatial Data

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 geospatial analysis
Via SQL (Earth only, for now)



CRATE.IO

Links

https://github.com/crate

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

https://crate.io

Follow us on twitter
@crateio *@claus__m*

Thanks!