

Ghislain Fourny

Big Data for Engineers Spring 2020

3. Object Storage



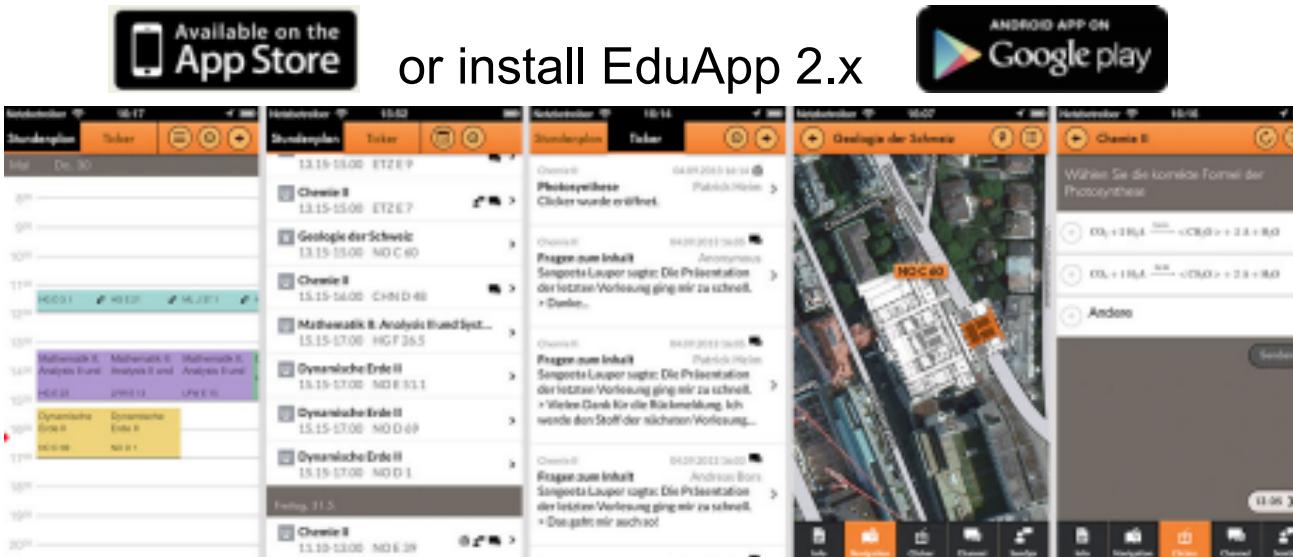
Where are we?

Last lecture:
Reminder on
relational databases

Poll

Go *now* to:

<https://eduapp-app1.ethz.ch/>



Where are we?

Relational databases
fit on a
single machine

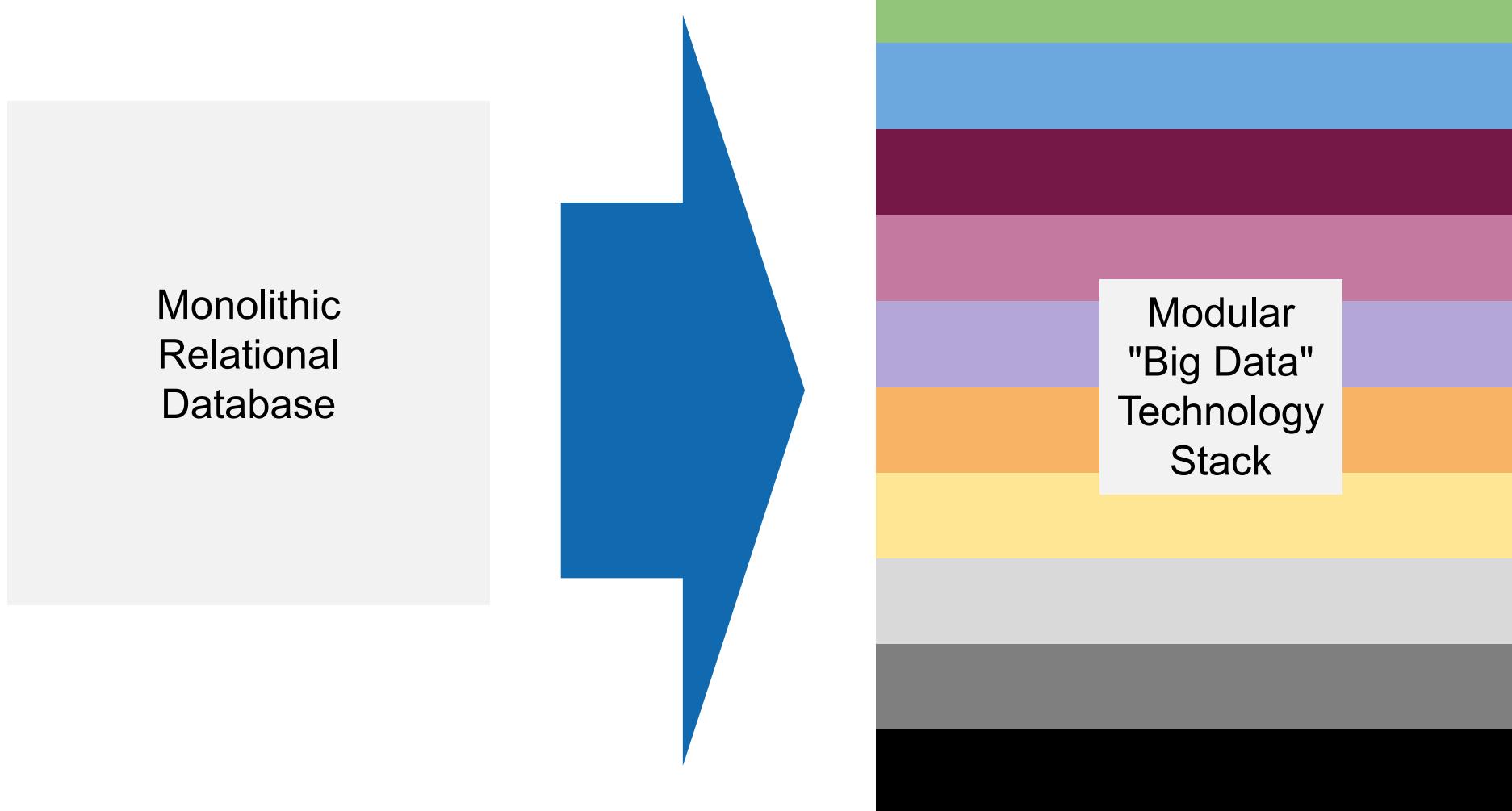


Where are we?

Petabytes
do not fit
on a
single machine



The lecture journey



Not reinventing the wheel

99%

of what we learned
with 48 years of
SQL and relational
**can be
reused**

Important take-away points

Relational algebra:

- ✓ Selection
- ✓ Projection
- ✓ Grouping
- ✓ Sorting
- ✓ Joining

Important take-away points

Language

- ✓ SQL
- ✓ Declarative languages
- ✓ Functional languages
- ✓ Optimizations
- ✓ Query plans
- ✓ Indices

Important take-away points

What a table is made of

- ✓ Table
- ✓ Rows
- ✓ Columns
- ✓ Primary key

Important take-away points

SQL

Consistency constraints

- |  Tabular integrity
- |  Domain integrity
- |  Atomic integrity (1st normal form)
- |  Boyce-Codd normal form

Important take-away points

Consistency constraints	
SQL	<ul style="list-style-type: none">✗ Tabular integrity✗ Domain integrity✗ Atomic integrity (1st normal form)✗ Boyce-Codd normal form
NoSQL	<ul style="list-style-type: none">NEW Heterogeneous dataNEW Nested dataNEW Denormalized data

Important take-away points

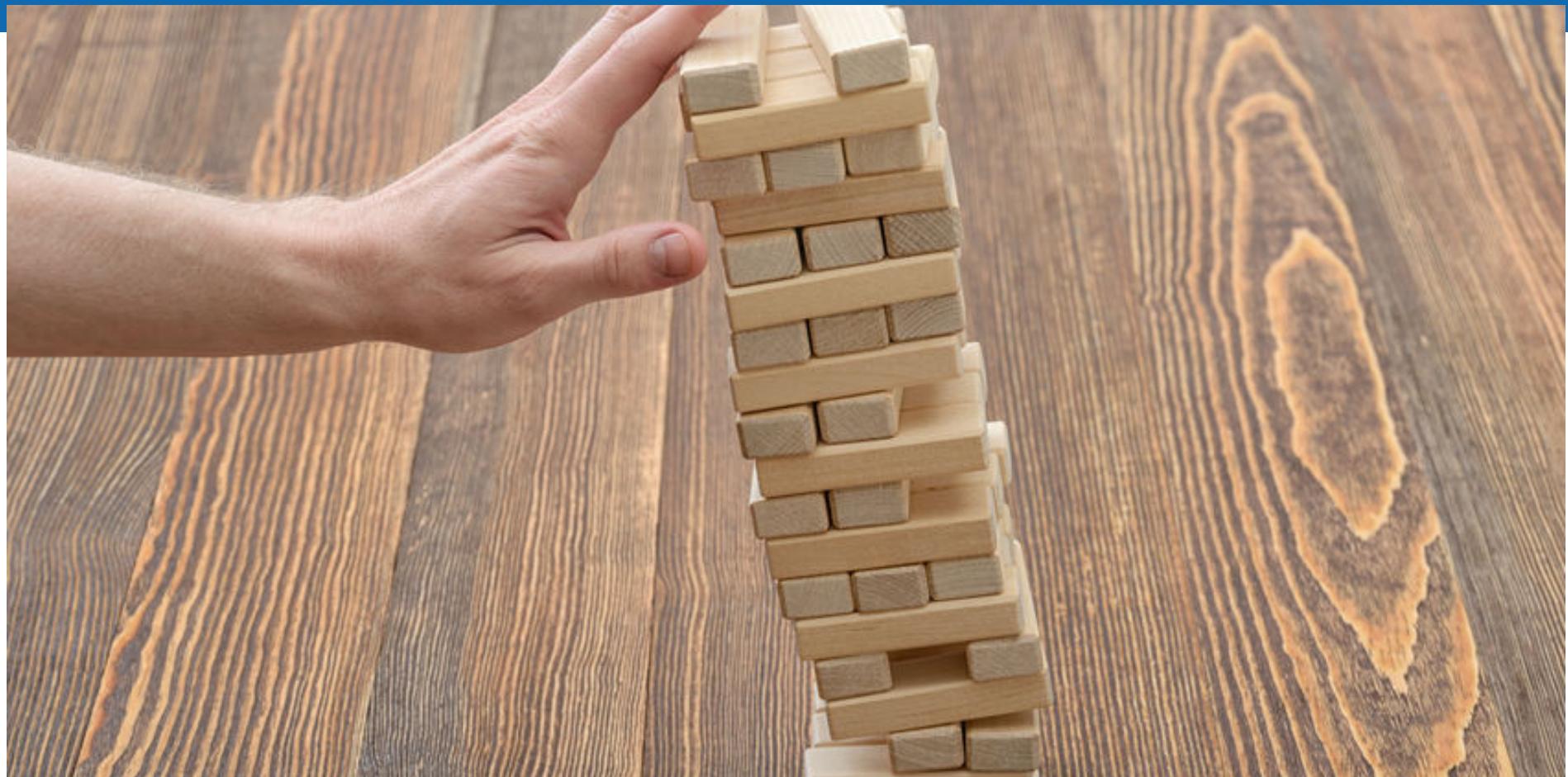
ACID

Transactions

- ✖ Atomicity
- ✖ Consistency
- ✖ Isolation
- ✖ Durability

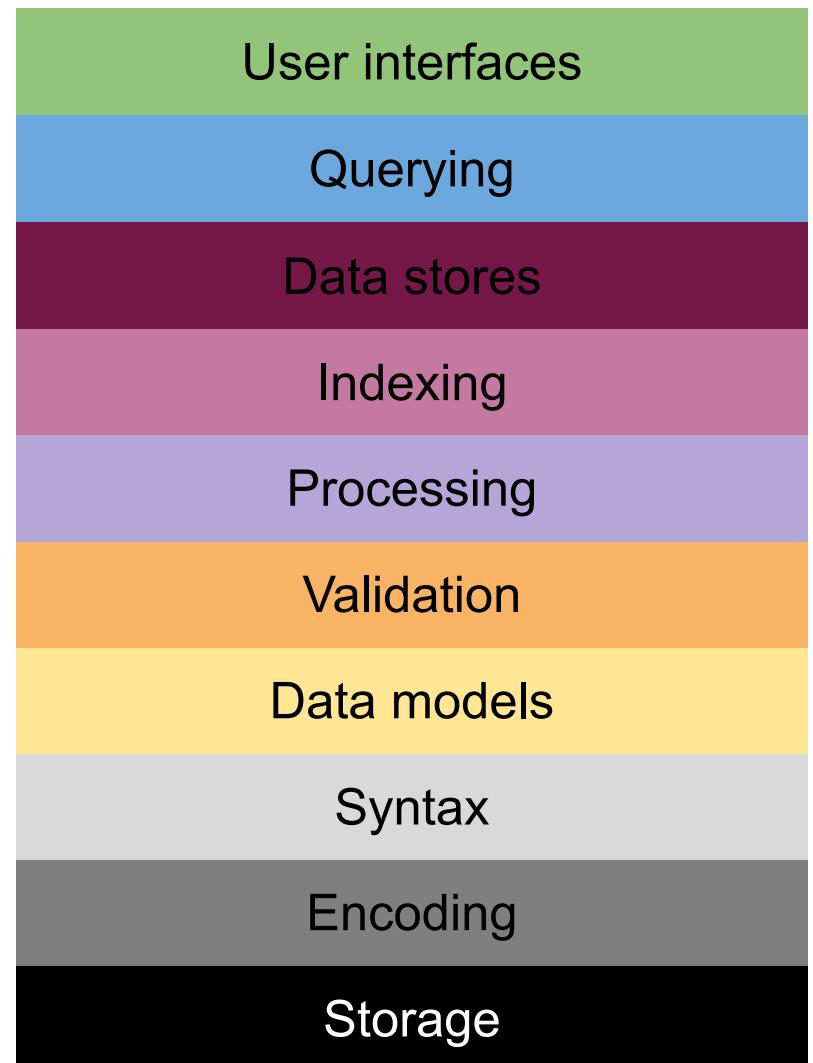
Important take-away points

Transactions	
ACID	<ul style="list-style-type: none">✗ Atomicity✗ Consistency✗ Isolation✗ Durability
CAP	<ul style="list-style-type: none">NEW Atomic ConsistencyNEW AvailabilityNEW Partition toleranceNEW Eventual Consistency



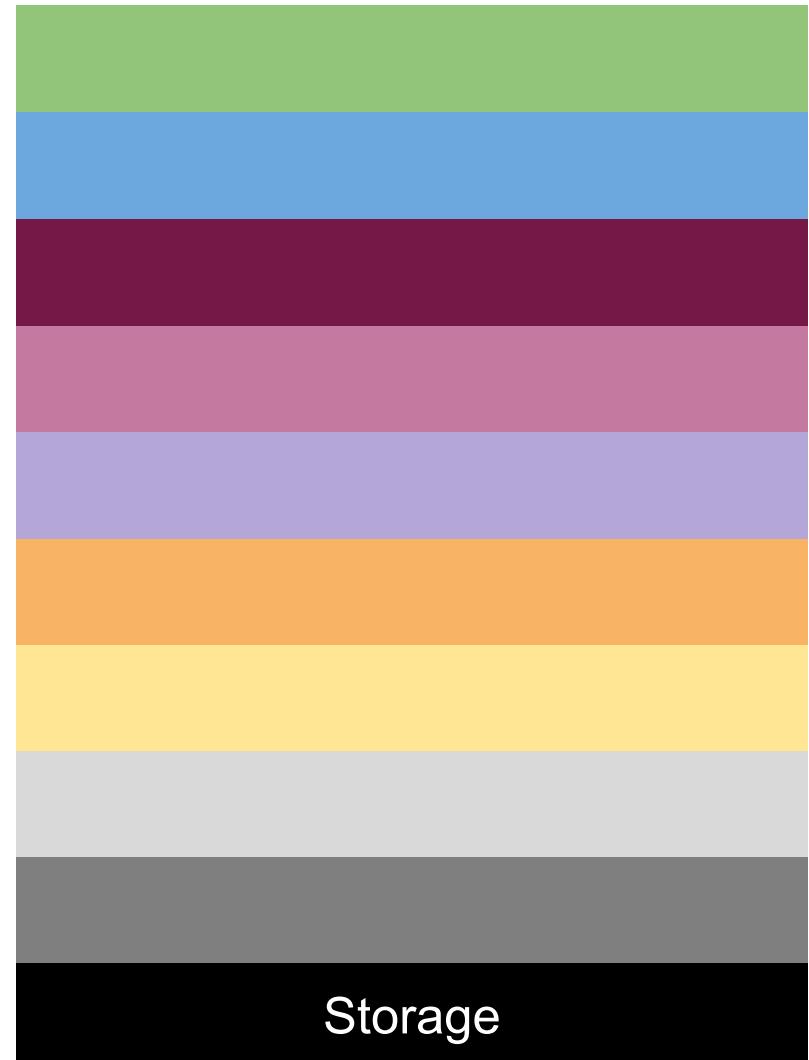
The stack

The stack



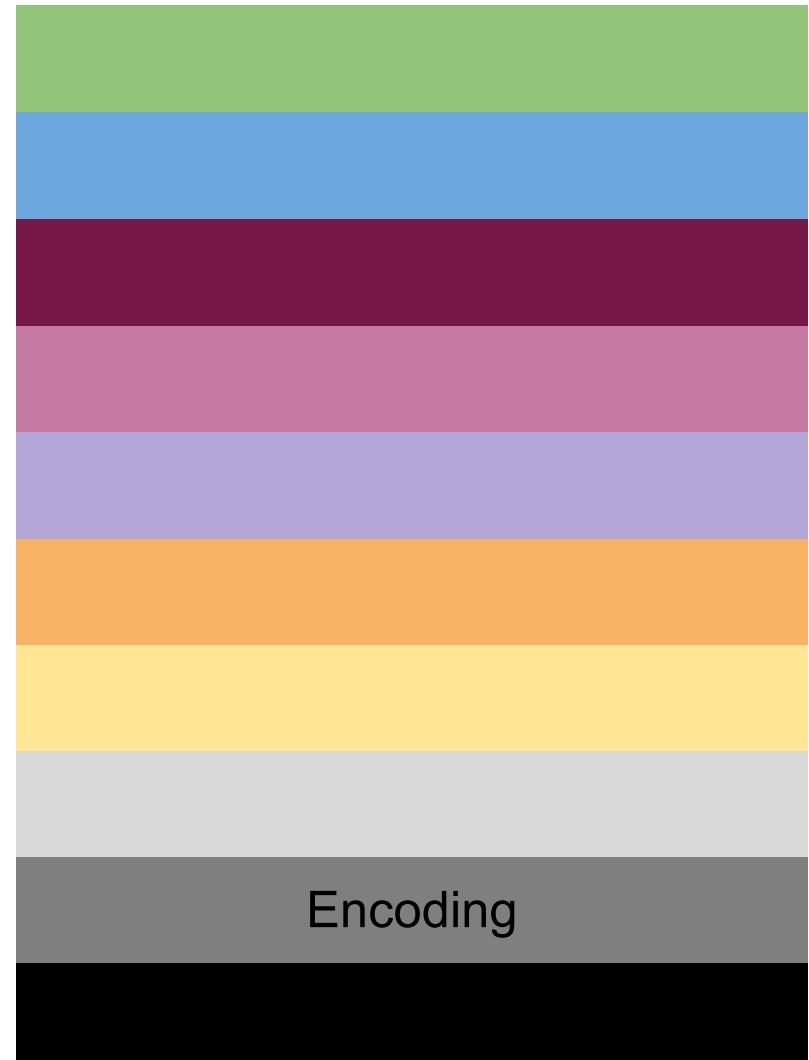
The stack: Storage

Local filesystem
NFS
GFS
HDFS
S3
Azure Blob Storage



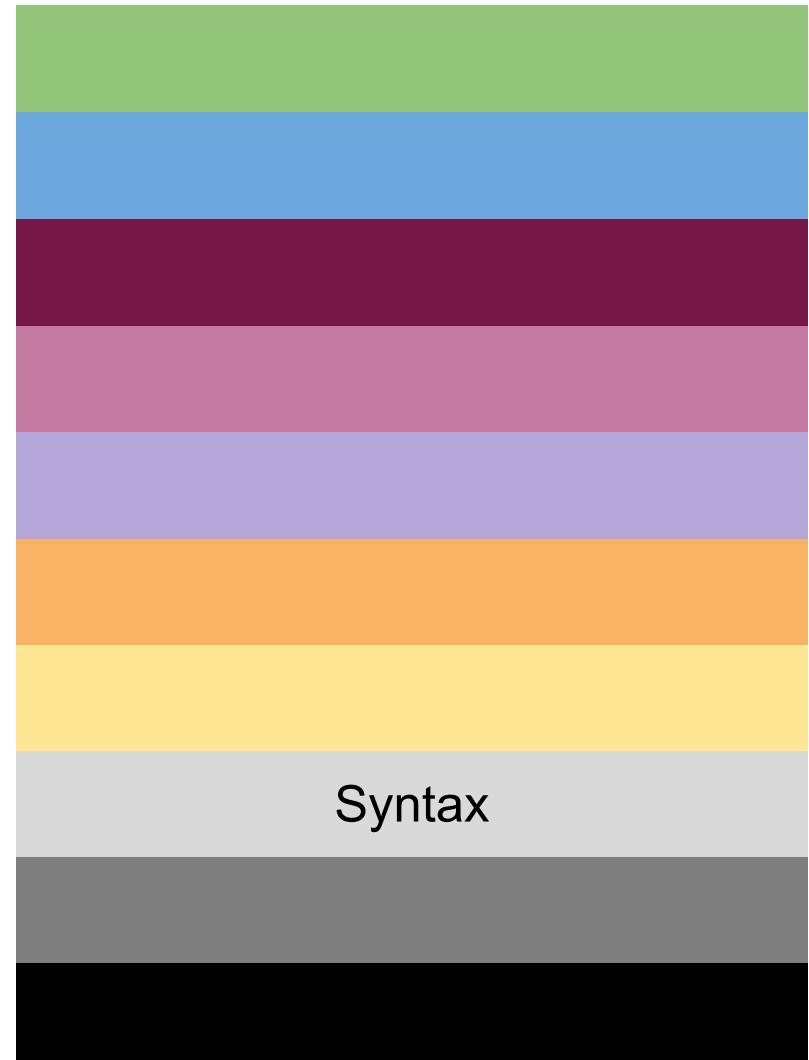
The stack: Encoding

ASCII
ISO-8859-1
UTF-8
BSON



The stack: Syntax

Text
CSV
XML
JSON
RDF/XML
Turtle
XBRL



The stack: Data models

Tables: Relational model
Trees: XML Infoset, XDM
Graphs: RDF
Cubes: OLAP



The stack: Validation

XML Schema
JSON Schema
Relational schemas
XBRL taxonomies

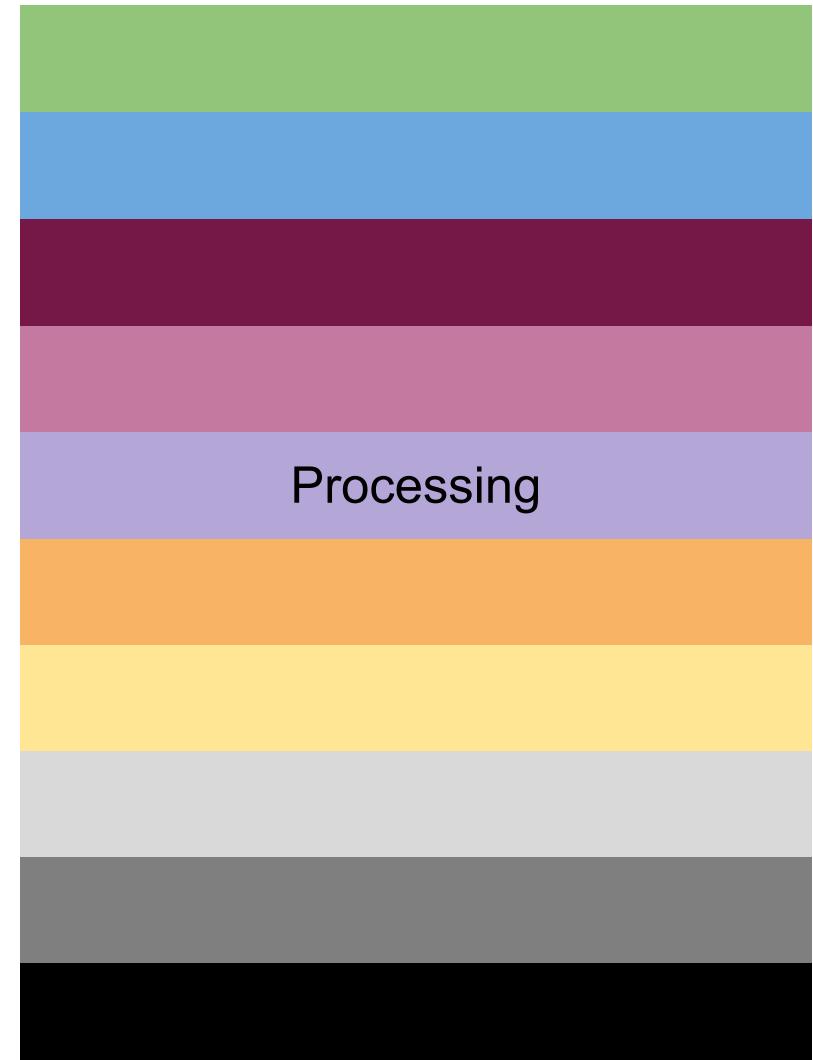


The stack: Processing

Two-phase processing:
MapReduce

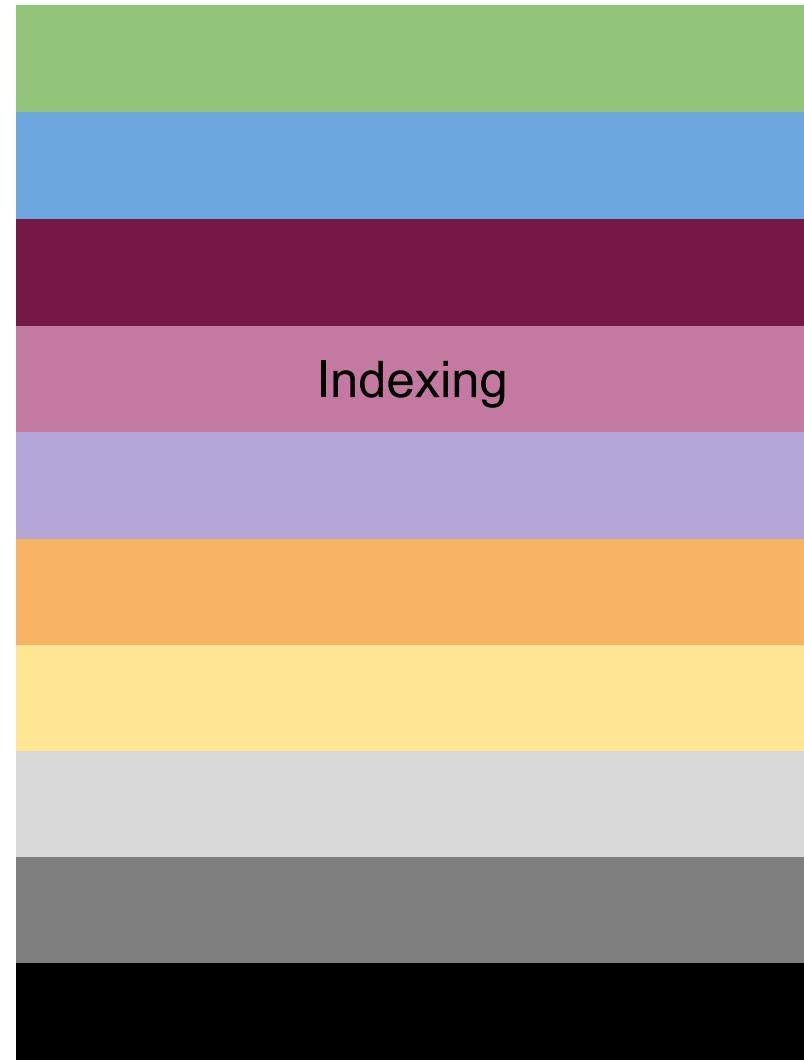
DAG-driven processing:
Tez, Spark, Flink, Ray

Elastic computing:
EC2



The stack: Indexing

Key-value stores
Hash indices
B-Trees
Geographical indices
Spatial indices



The stack: Data stores

RDBMS
(Oracle/IBM/Microsoft)

MongoDB

CouchBase

ElasticSearch

Hive

HBase

MarkLogic

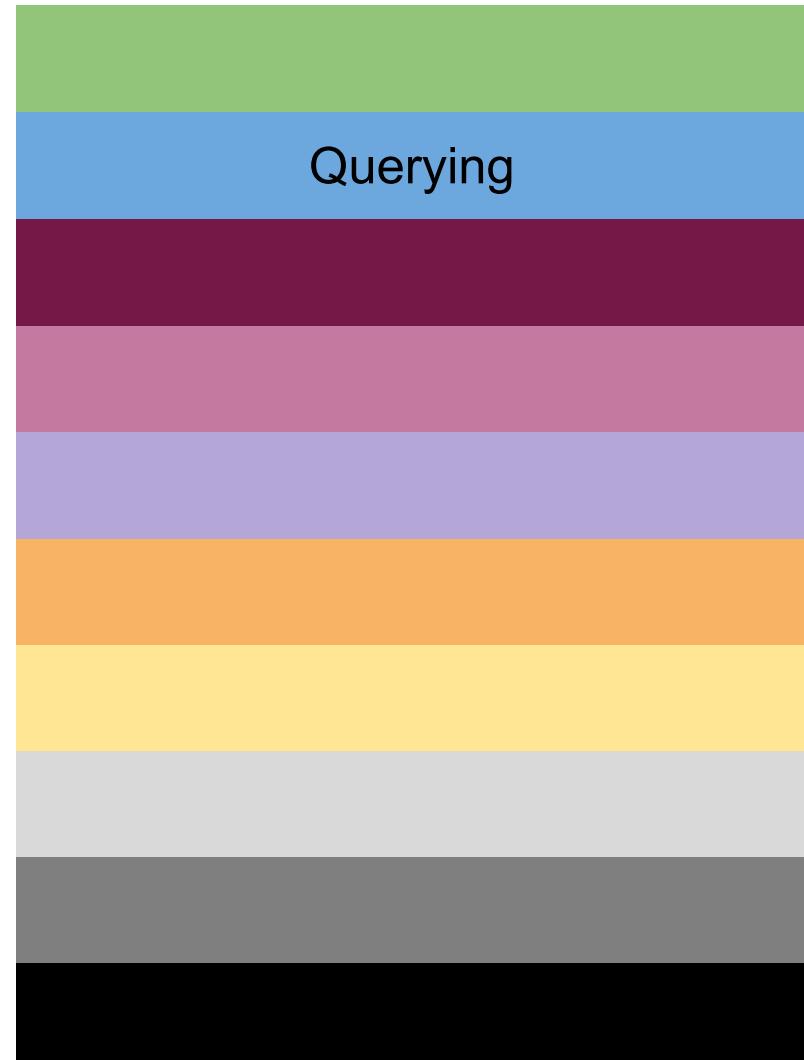
Cassandra

...



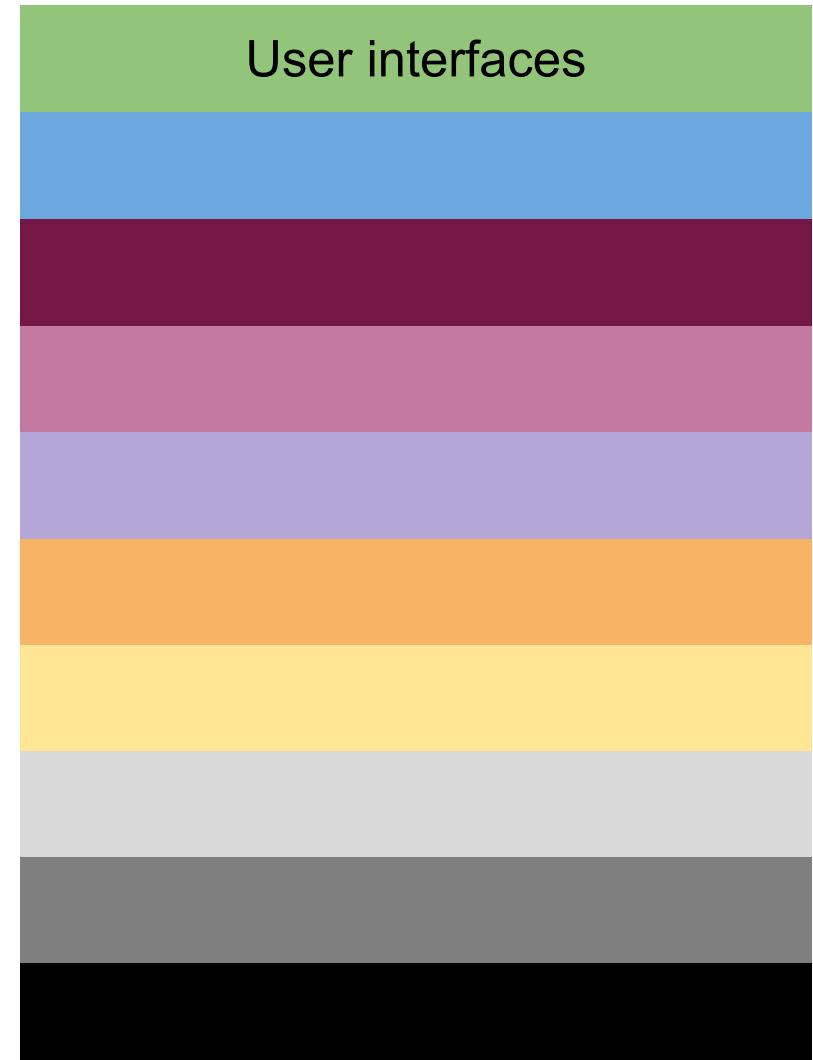
The stack: Querying

SQL
XQuery
JSONiq
N1QL
MDX
SPARQL
REST APIs



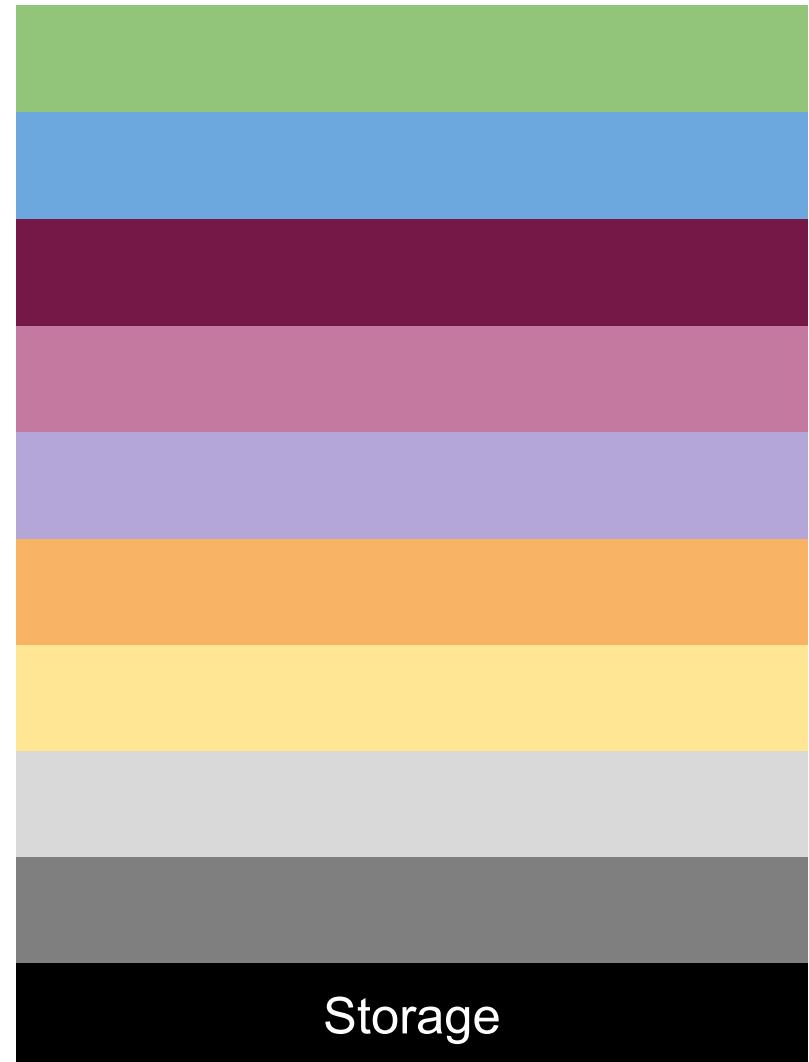
The stack: User interfaces (UI)

Excel
Access
Tableau
Qlikview
BI tools



The stack: Storage

Today!





Storage: from a single machine to a cluster

Storage

Data
needs to be
stored
somewhere

Database

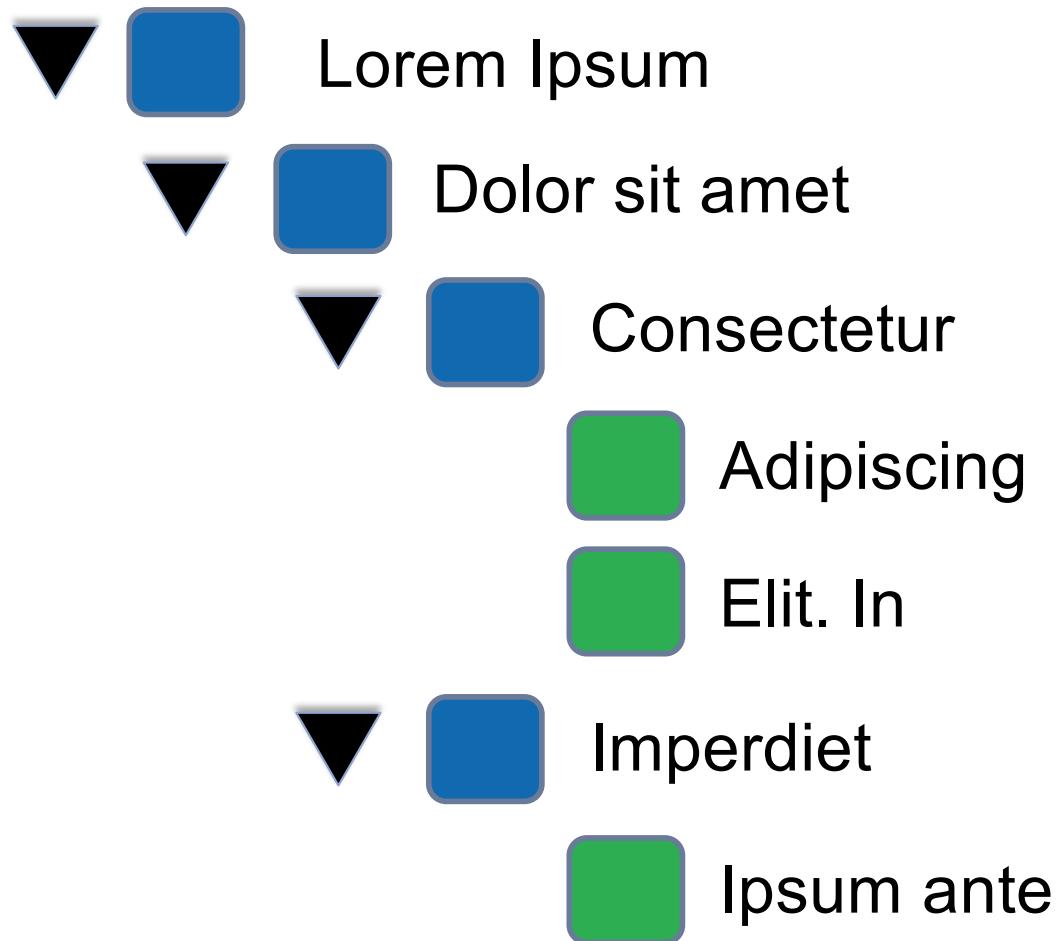
Storage

Let's start from the 70s...



Vitaly Korovin / 123RF Stock Photo

File storage



Files
organized
in a
hierarchy

What is a file made of?

Content
+ **Metadata**

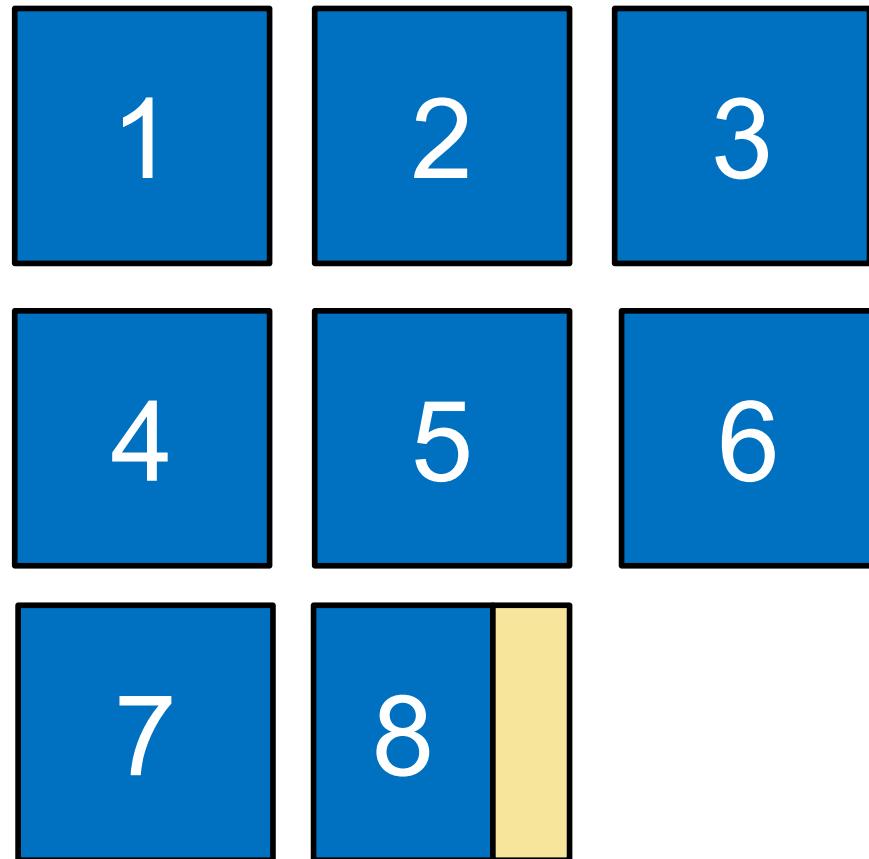
File

File Metadata

```
$ ls -l
total 48
drwxr-xr-x  5 gfourny staff  170 Jul 29 08:11 2009
drwxr-xr-x 16 gfourny staff  544 Aug 19 14:02 Exercises
drwxr-xr-x 11 gfourny staff  374 Aug 19 14:02 Learning Objectives
drwxr-xr-x 18 gfourny staff  612 Aug 19 14:52 Lectures
-rw-r--r--  1 gfourny staff 1788 Aug 19 14:04 README.md
```

Fixed "schema"

File Content: Block storage



Files
content
stored in
blocks

Local storage



Local Machine

Local storage



Local Machine
LAN (NAS)



LAN = local-area network
NAS = network-attached storage

Local storage



Local Machine



LAN (NAS)



WAN



LAN = local-area network

NAS = network-attached storage

WAN = wide-area network

Scaling Issues



Aleksandr Elesin / 123RF Stock Photo

Scaling Issues



Aleksandr Elesin / 123RF Stock Photo

Scaling Issues

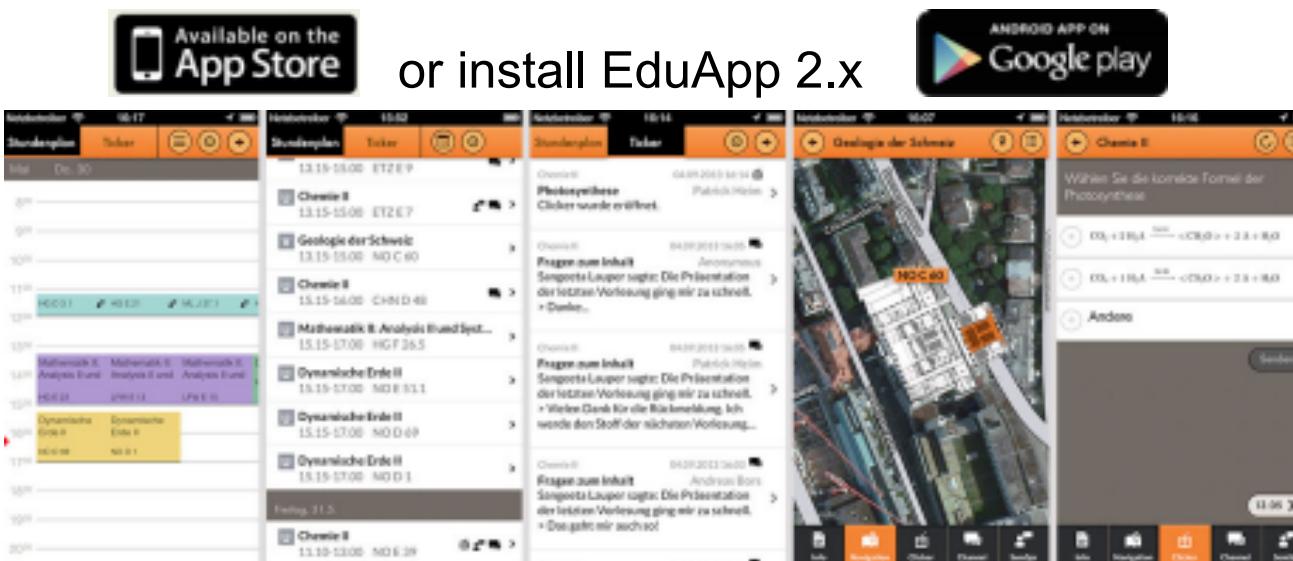


Aleksandr Elesin / 123RF Stock Photo

Poll

Go *now* to:

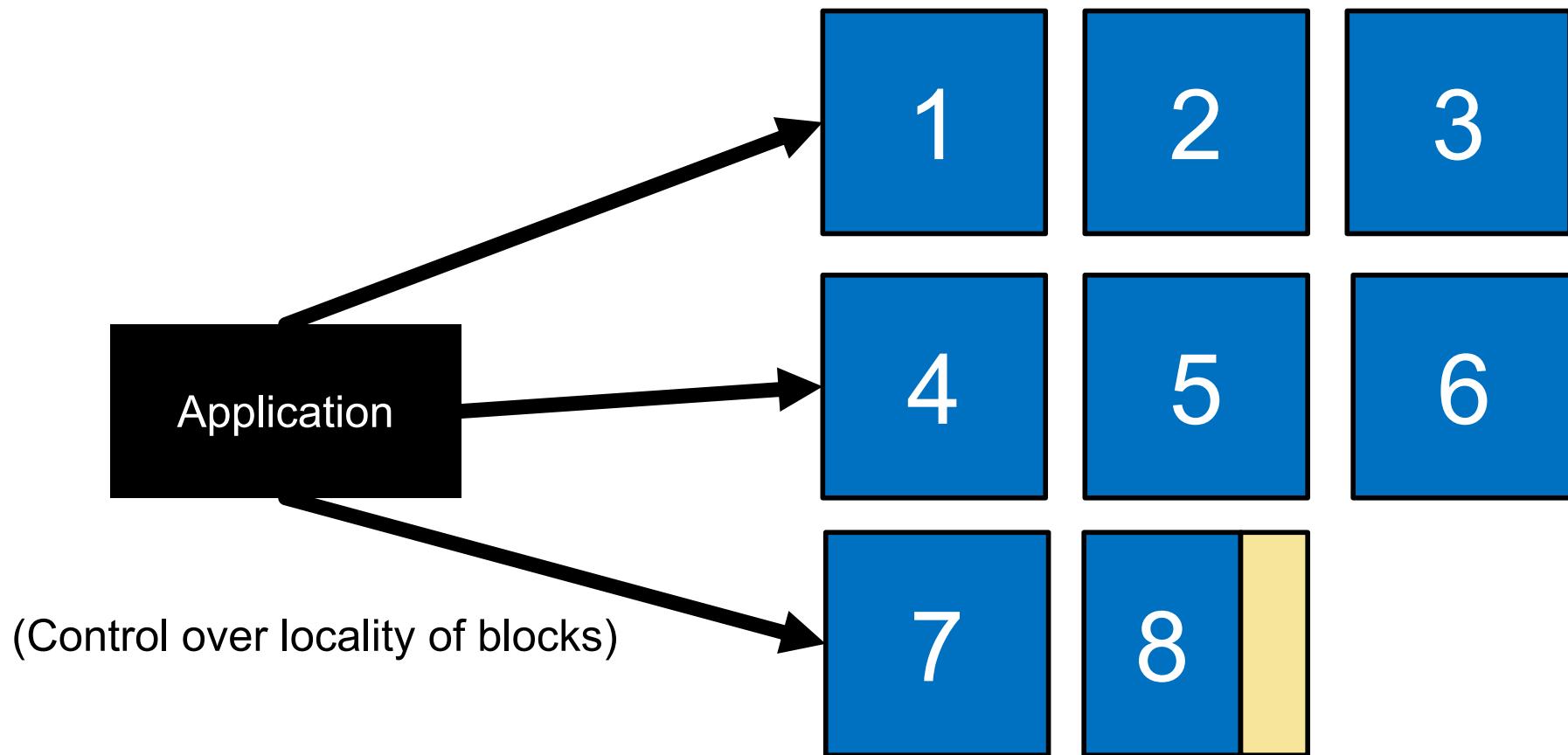
<https://eduapp-app1.ethz.ch/>



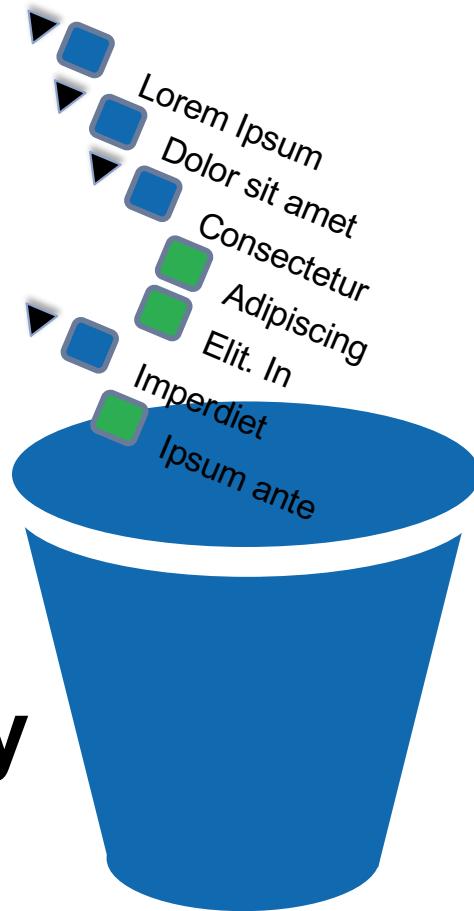
Analogy: Tesla batteries (up to 2017)



Better performance: Explicit Block Storage

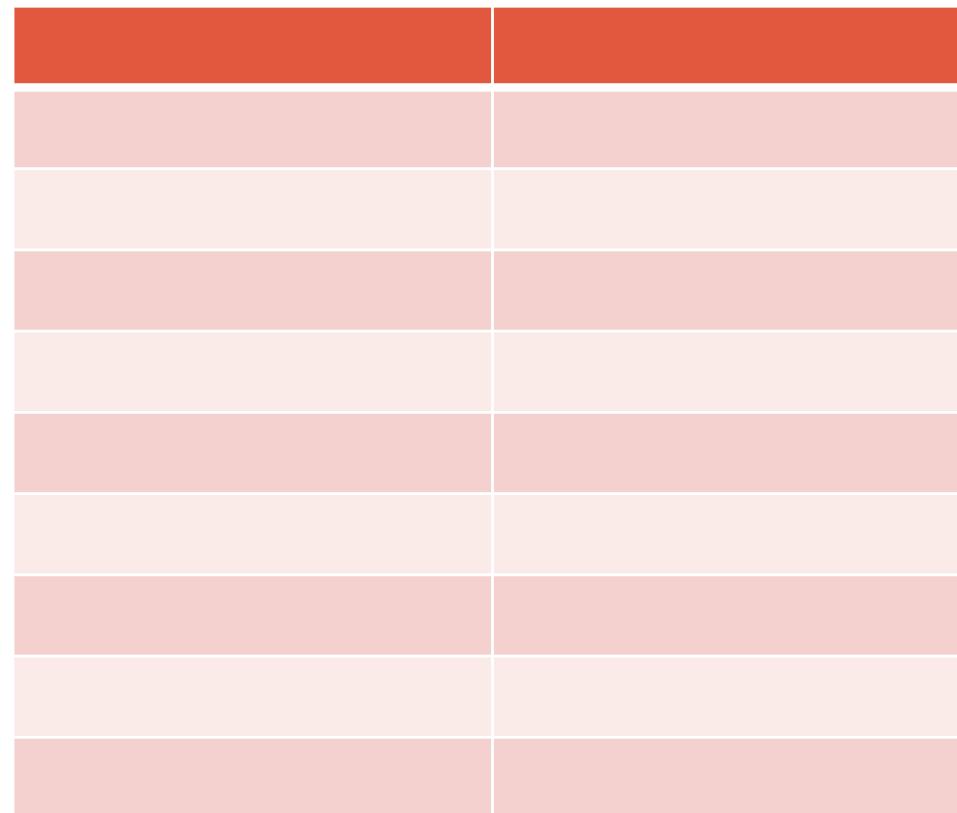


So how do we make this scale?



**1. We throw away
the hierarchy!**

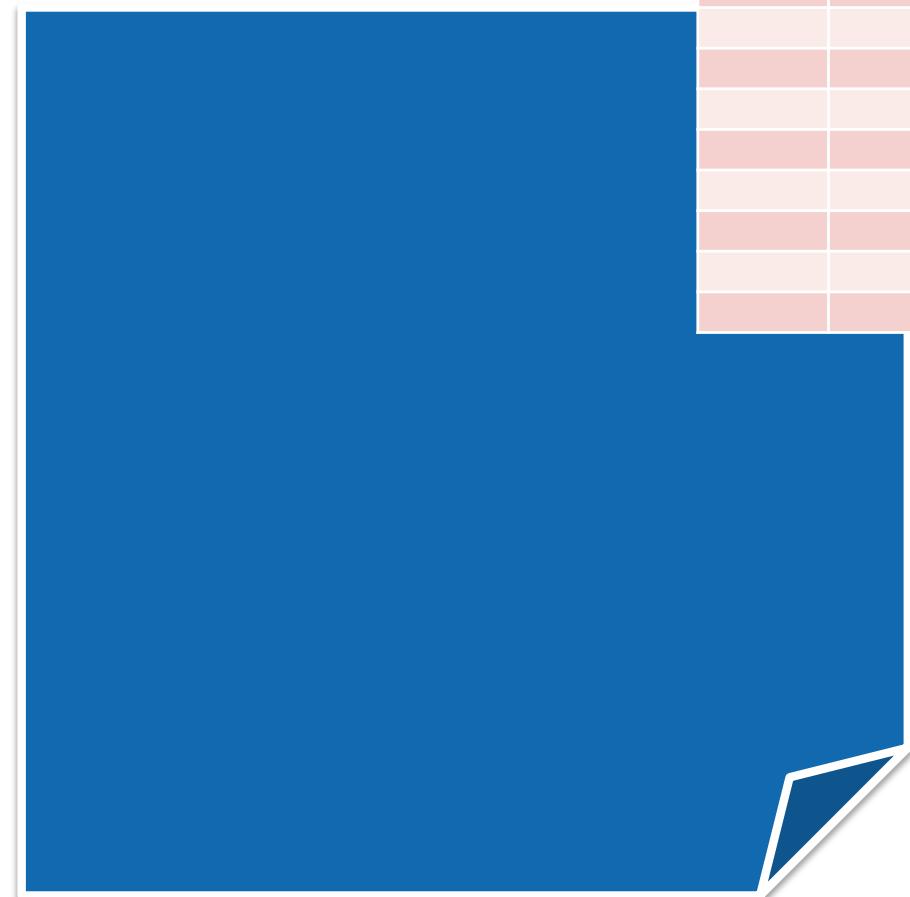
So how do we make this scale?



**2. We make
metadata flexible**

So how do we make this scale?

ID →



3. We make the data model trivial

So how do we make this scale?



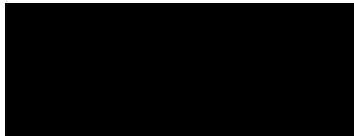
4. We use commodity hardware

... and we get Object Storage

... and we get Object Storage

"Black-box" objects

... and we get Object Storage

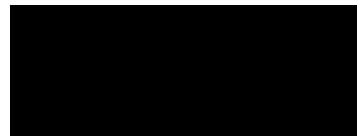


"Black-box" objects



Flat and global key-value model

... and we get Object Storage



"Black-box" objects

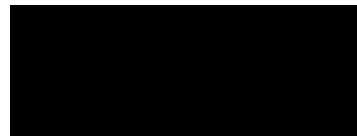


Flat and global key-value model



Flexible metadata

... and we get Object Storage



"Black-box" objects



Flat and global key-value model



Flexible metadata



Commodity hardware



Scale

**One machine's not good enough.
How do we scale?**



Approach 1: scaling **up**



Approach 1: scaling **up**



Approach 2: scaling **out**



Approach 2: scaling **out**



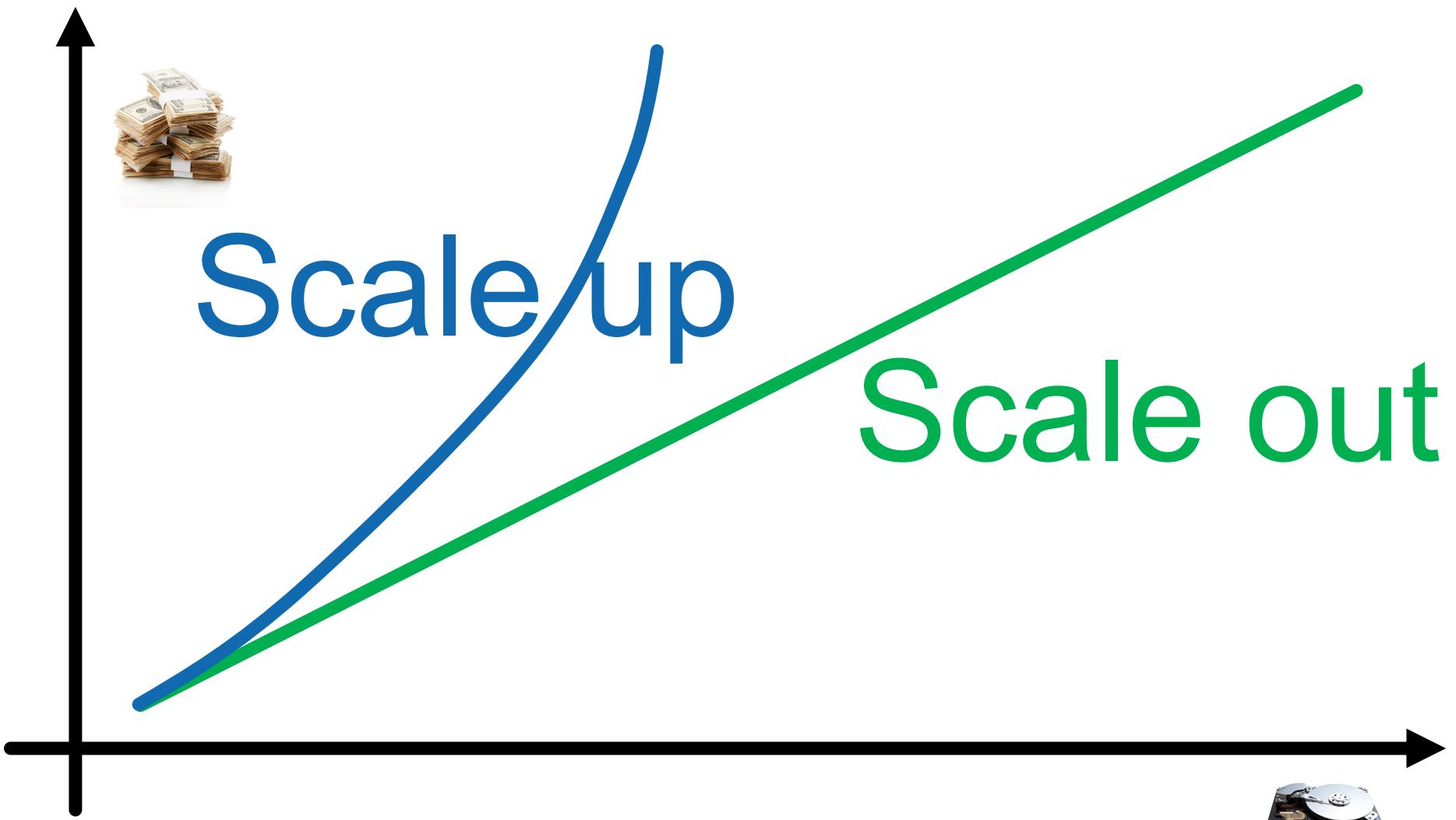
Approach 2: scaling **out**



Approach 2: scaling **out**



Hardware price comparison



Approach 3: be smart



Approach 3: be smart

“You can have a second computer **once you’ve shown you know how to use the first one.**”

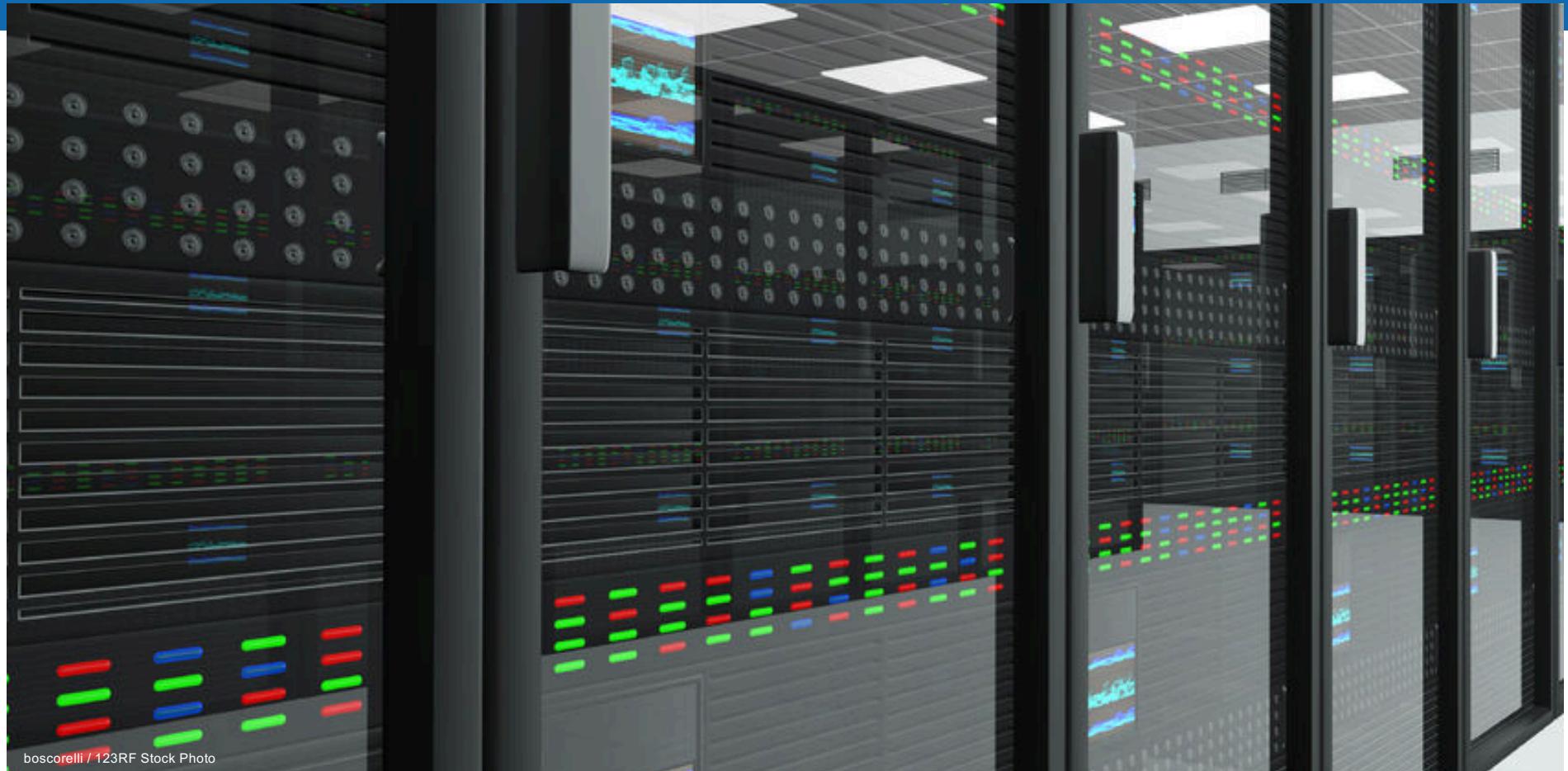
Paul Barham

In this lecture

Approach 2



Scale out

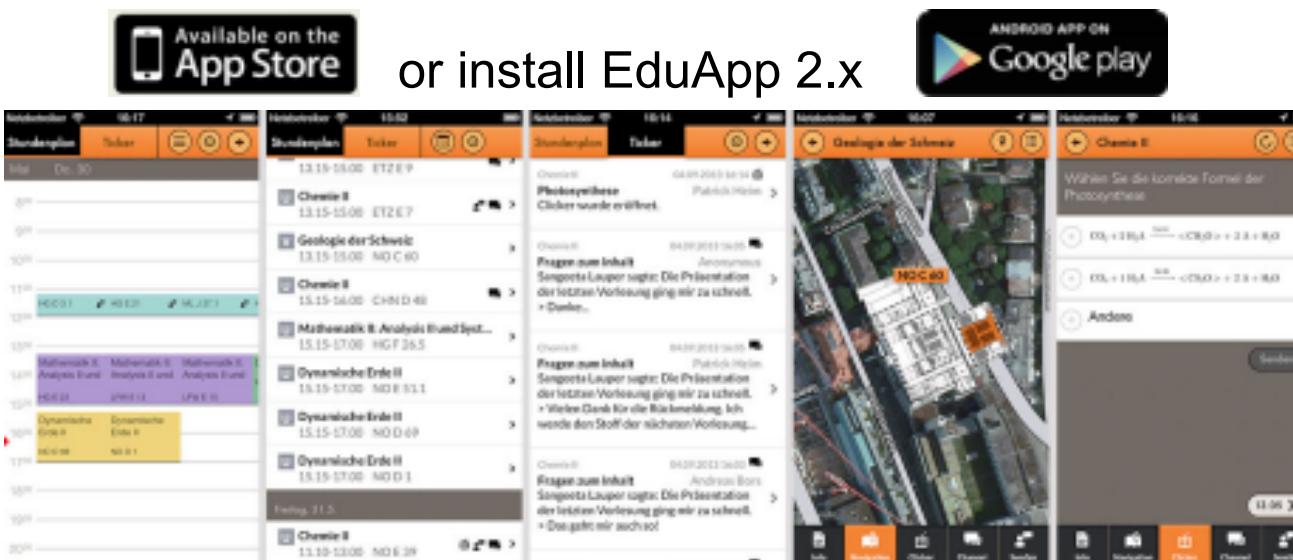


Data centers

Poll

Go *now* to:

<https://eduapp-app1.ethz.ch/>



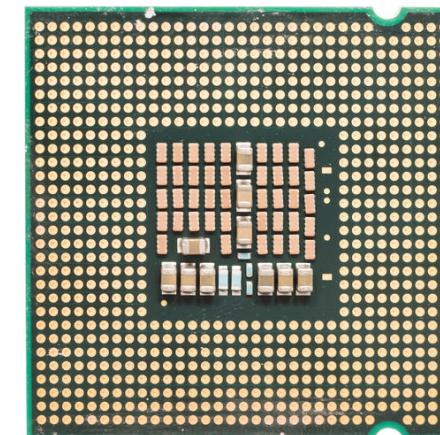
or install EduApp 2.x

Numbers - computing



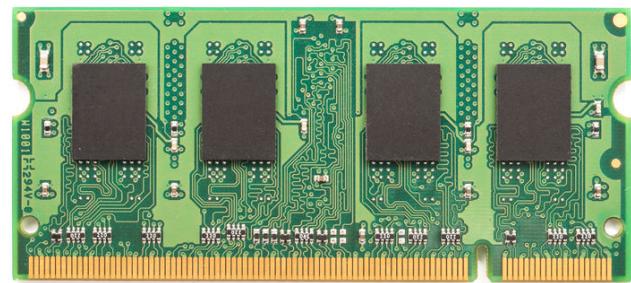
1,000-100,000
machines in a data center

1-100 cores
per server



Numbers - storage

1-20 TB
local storage
per server



16GB-6TB
of RAM per server

Numbers - network

1-100 GB/s
network bandwidth
for a server



Racks

Height in "rack units"
(e.g., 42 RU)



Racks

Modular:

- servers
- storage
- routers
- ...



Rack servers

1-4 RU

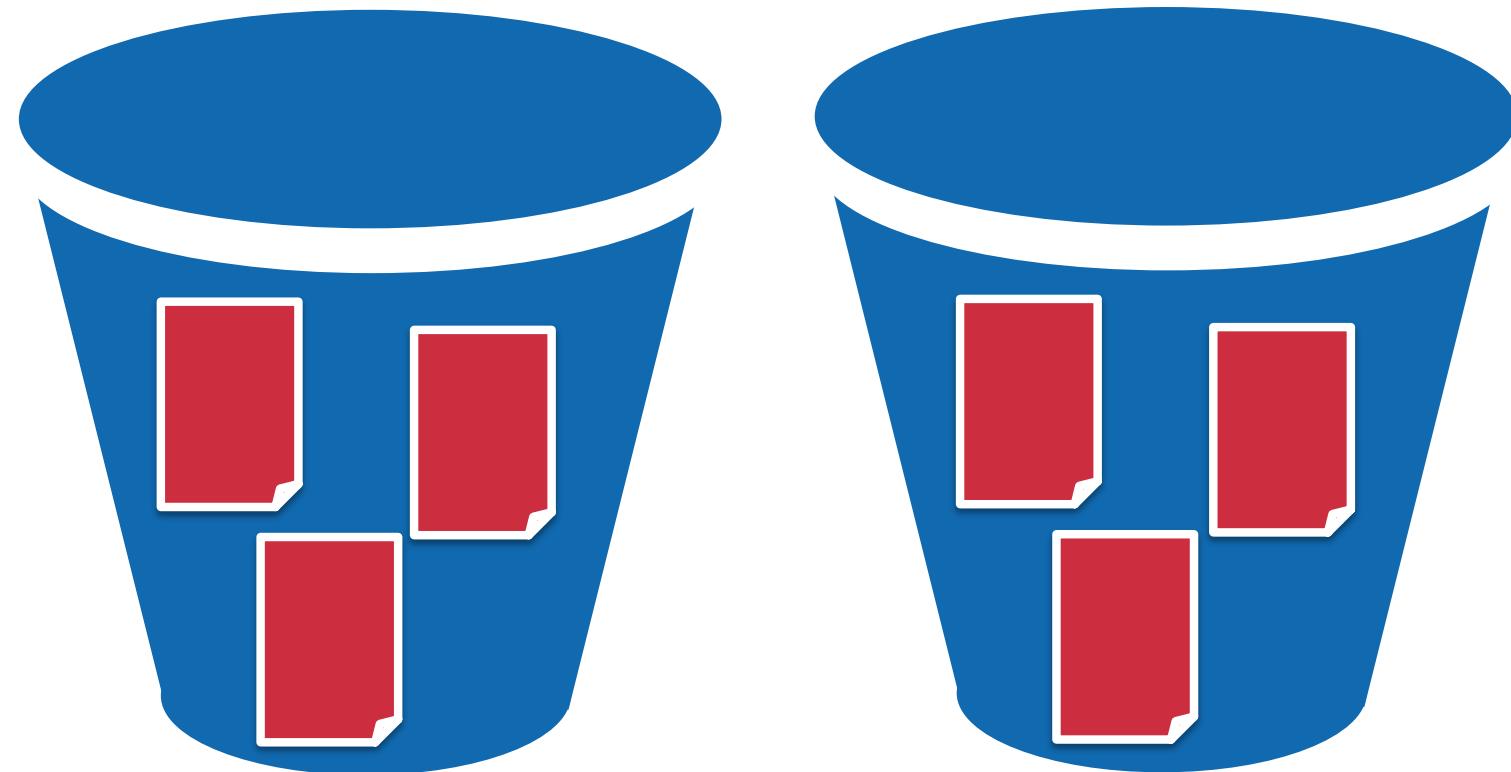


Lenovo ThinkServer RD430 Rack Server



Amazon S3

S3 Model



S3 Model

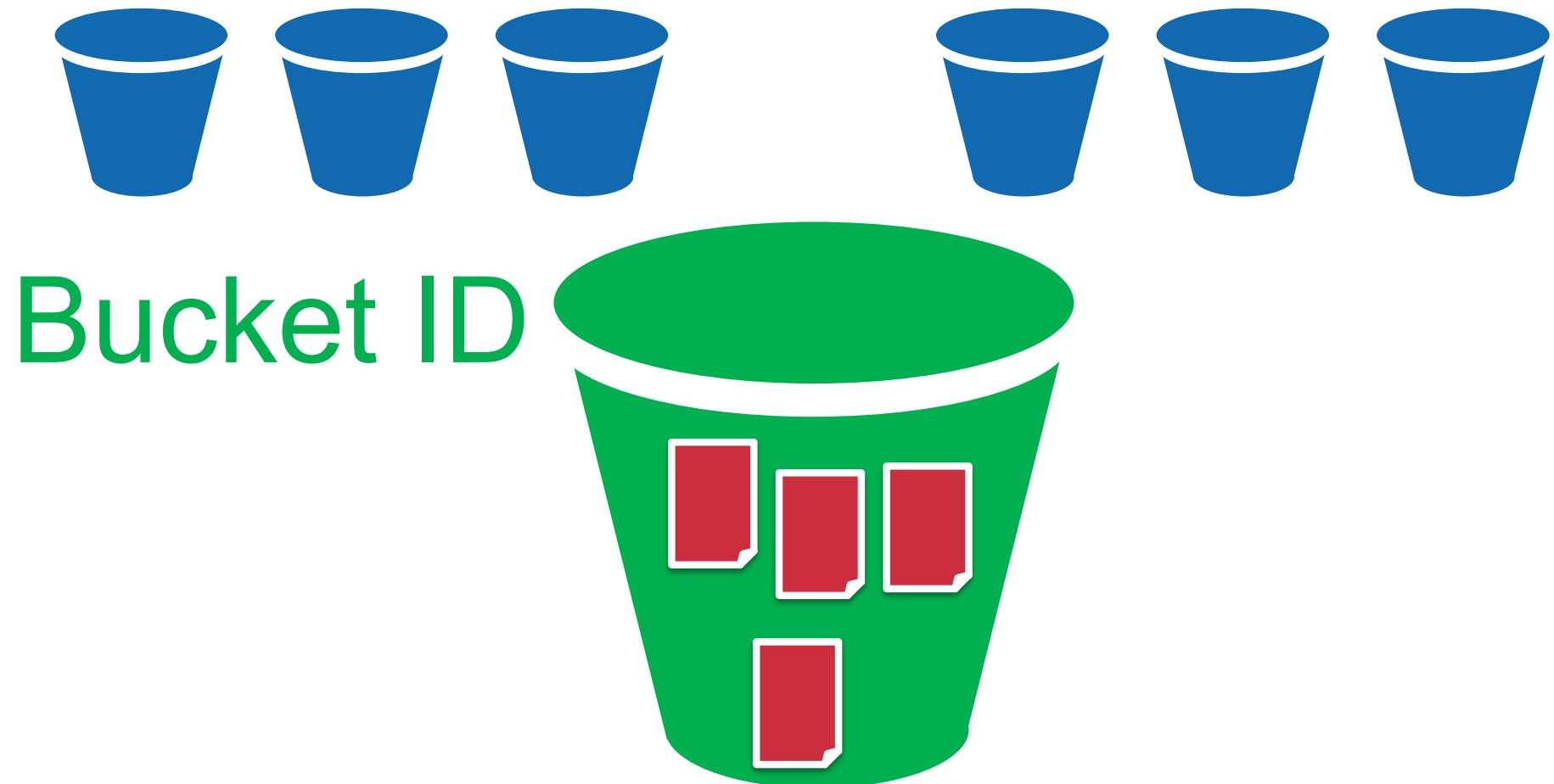


S3 Model

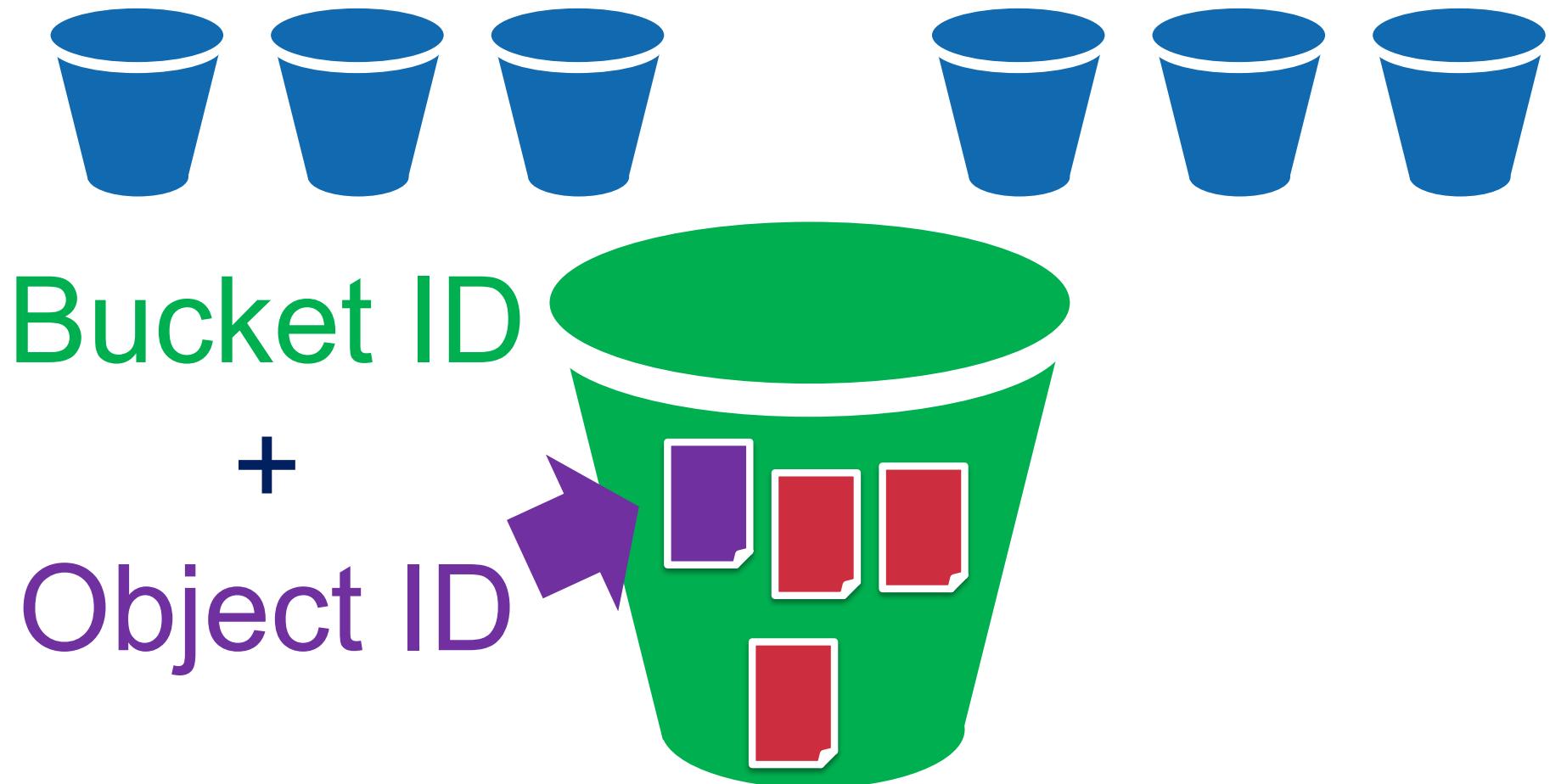


Bucket ID

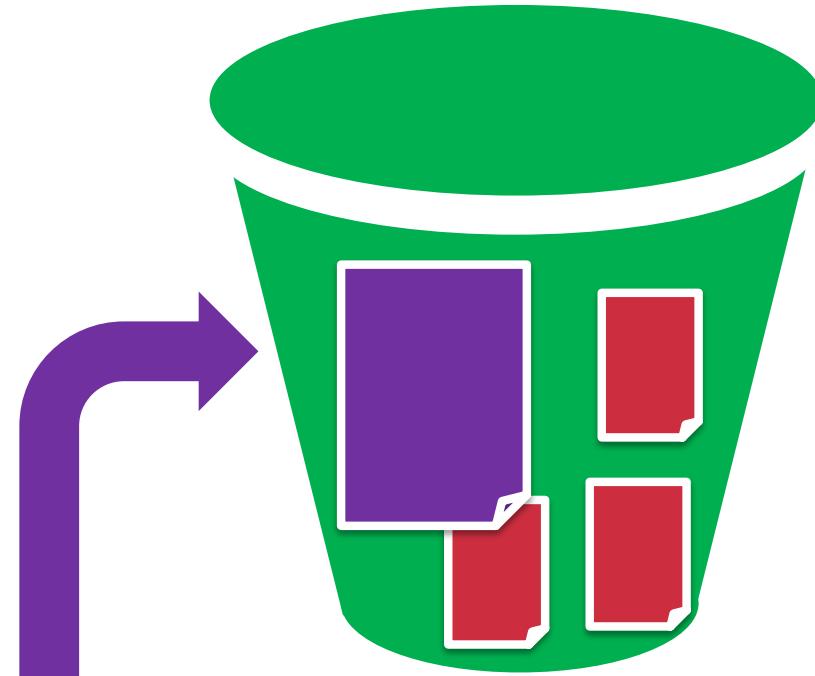
S3 Model



S3 Model

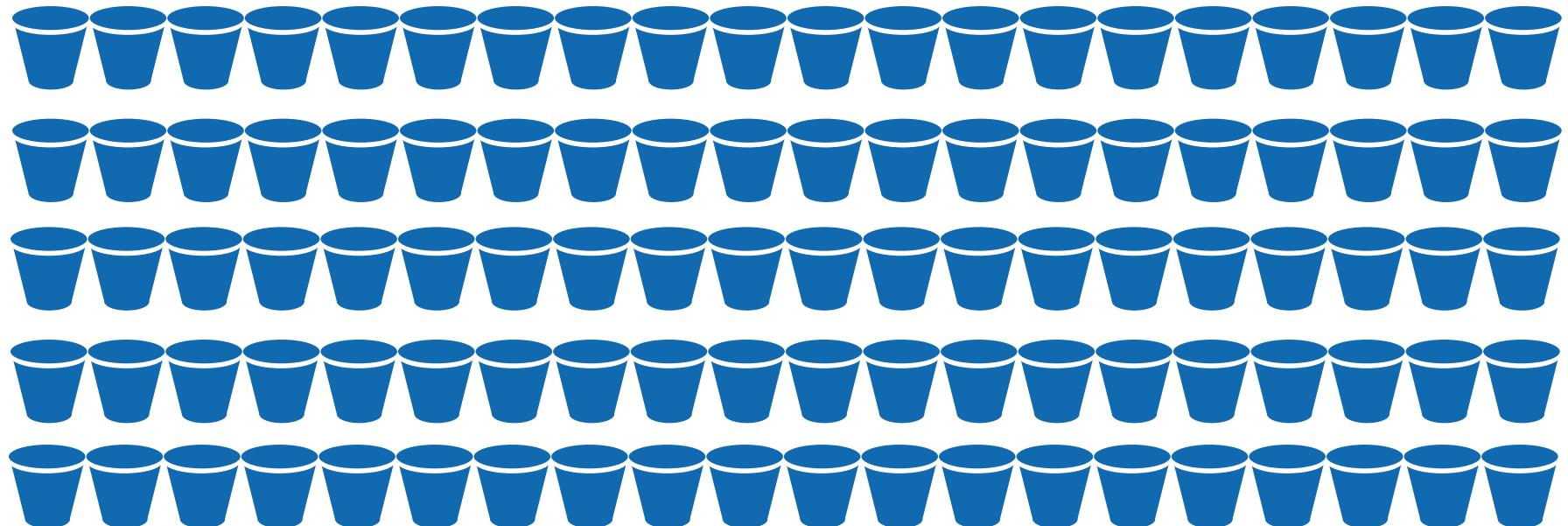


Scalability



Max. 5 TB

Scalability



100/account

(more upon request)

Durability

99.999999999%

Loss of 1 in 10^{11} objects in a year

Availability

99.99%

Down 1h / year

More about SLAs

9 9 9 9 9 9 9

More about SLA

SLA	Outage
99%	4 days/year
99.9%	9 hours/year
99.99%	53 minutes/year
99.999%	6 minutes/year
99.9999%	32 seconds/year
99.99999%	4 seconds/year

More about SLA

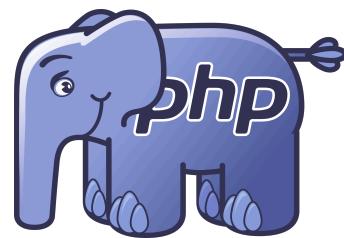
Amazon's approach:

Response time < 10 ms

in *99.9% of the cases*

(rather than **average** or **median**)

API



**Driver
(JDBC...)**

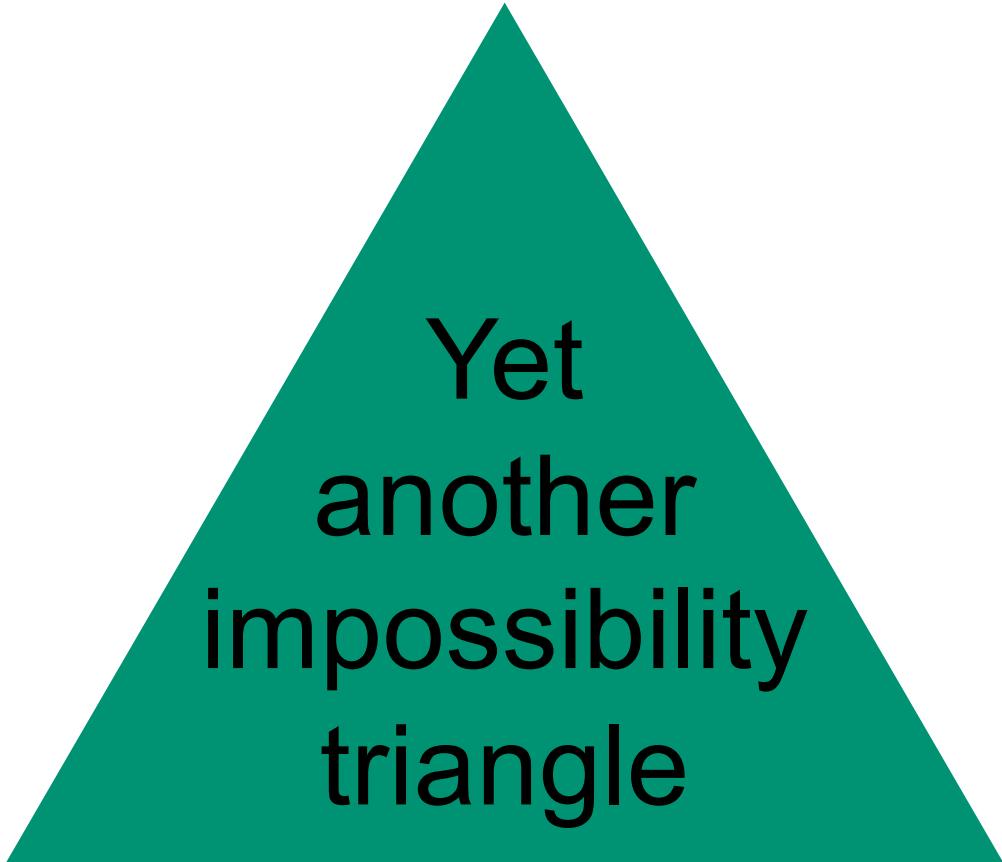


SOAP



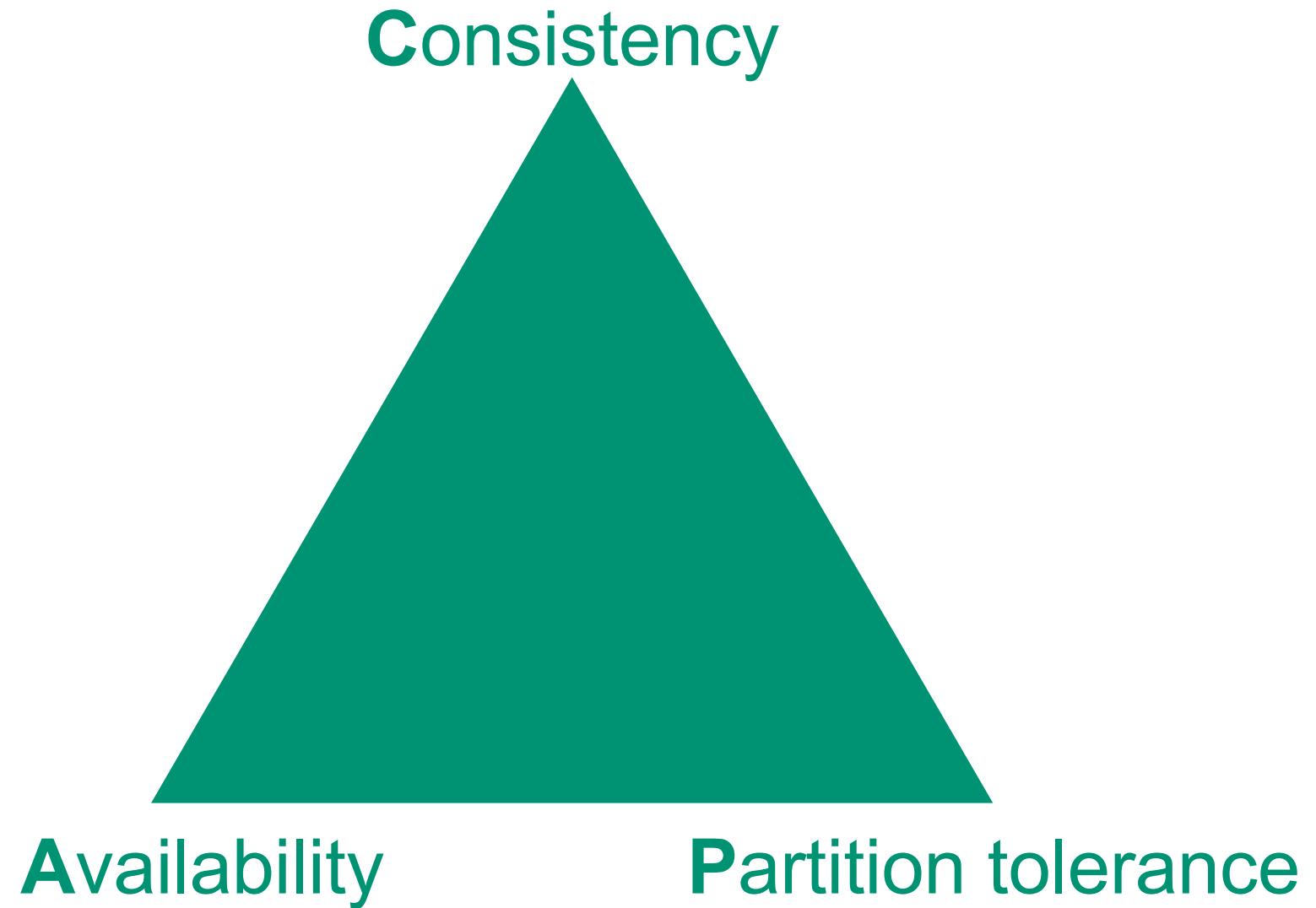
REST

The new era: the CAP theorem

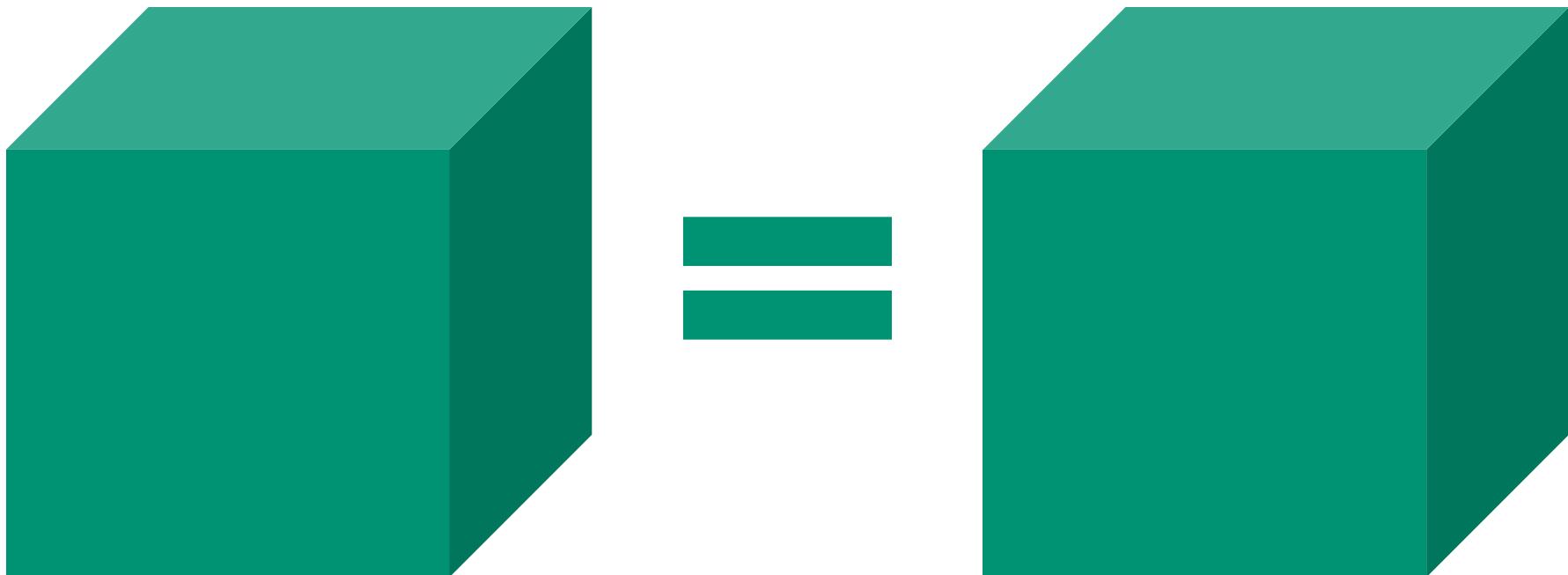


Yet
another
impossibility
triangle

The new era: the CAP theorem

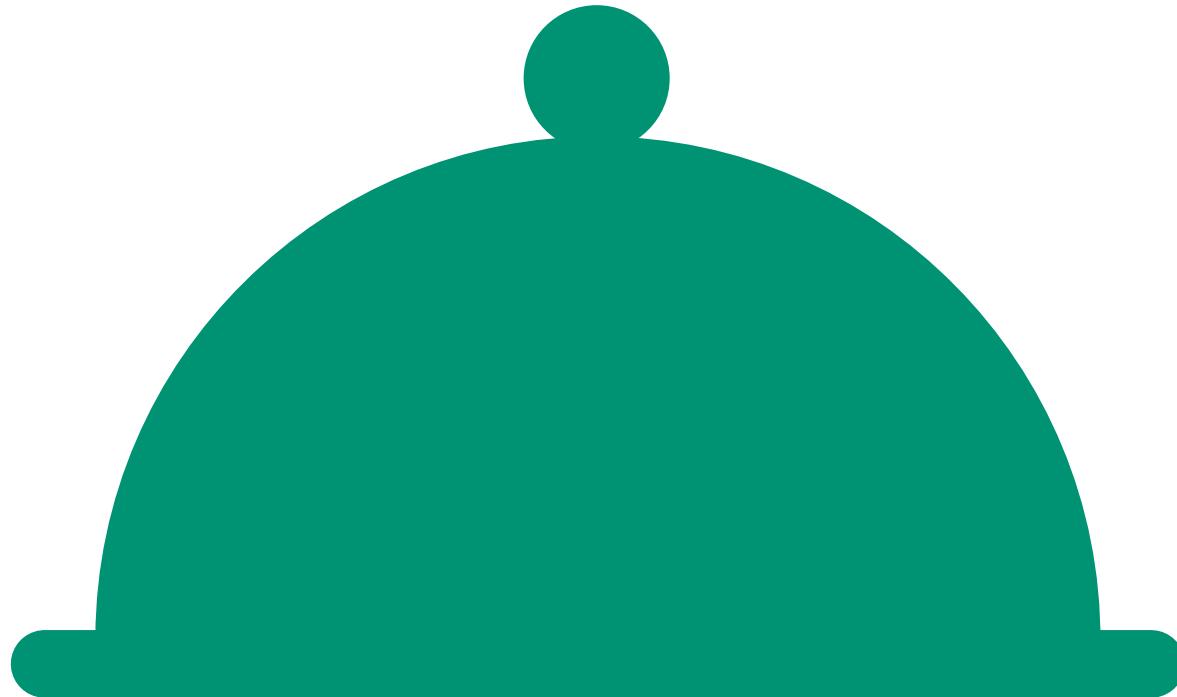


(Atomic) Consistency



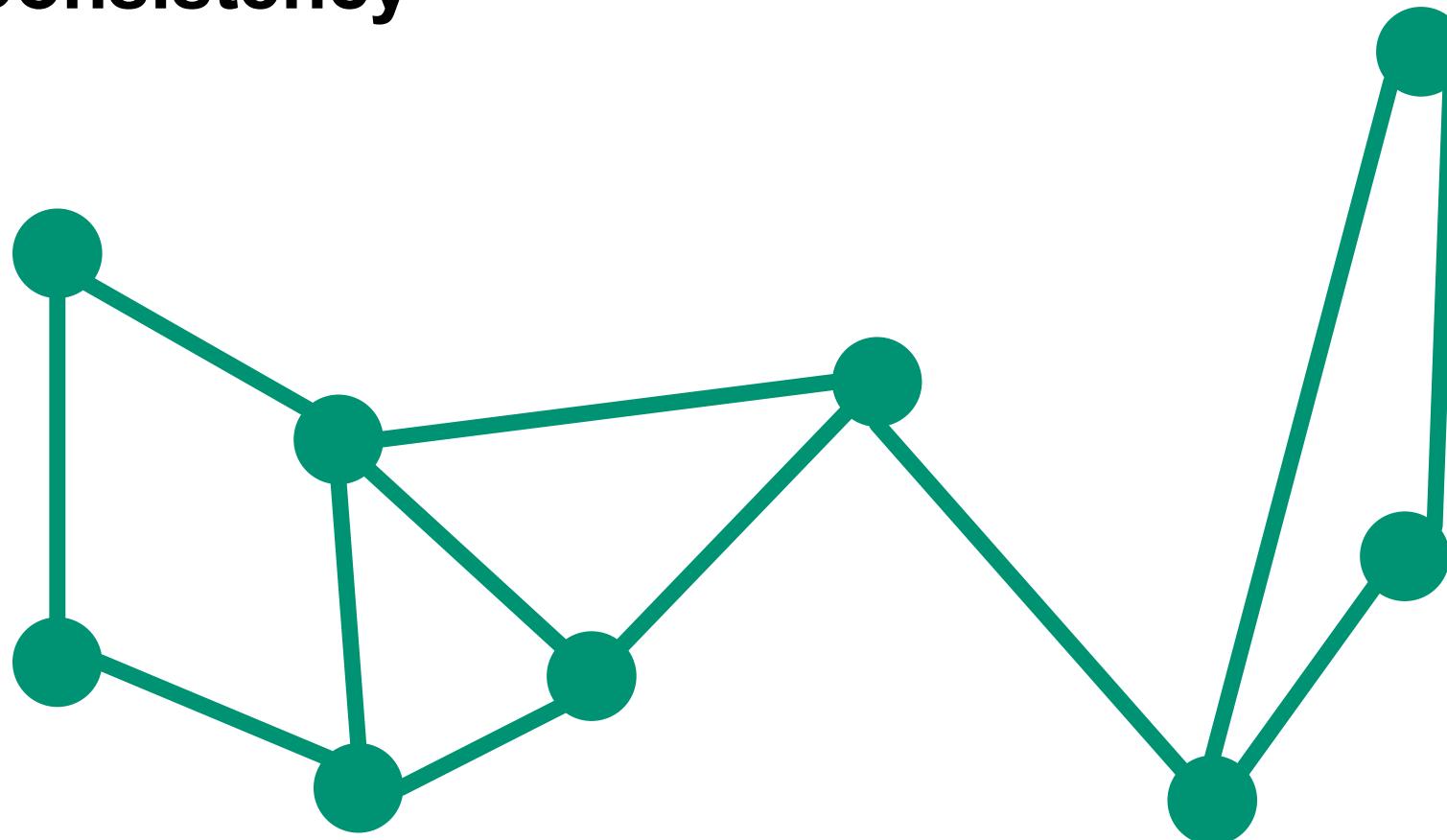
All nodes see the same data.

Availability

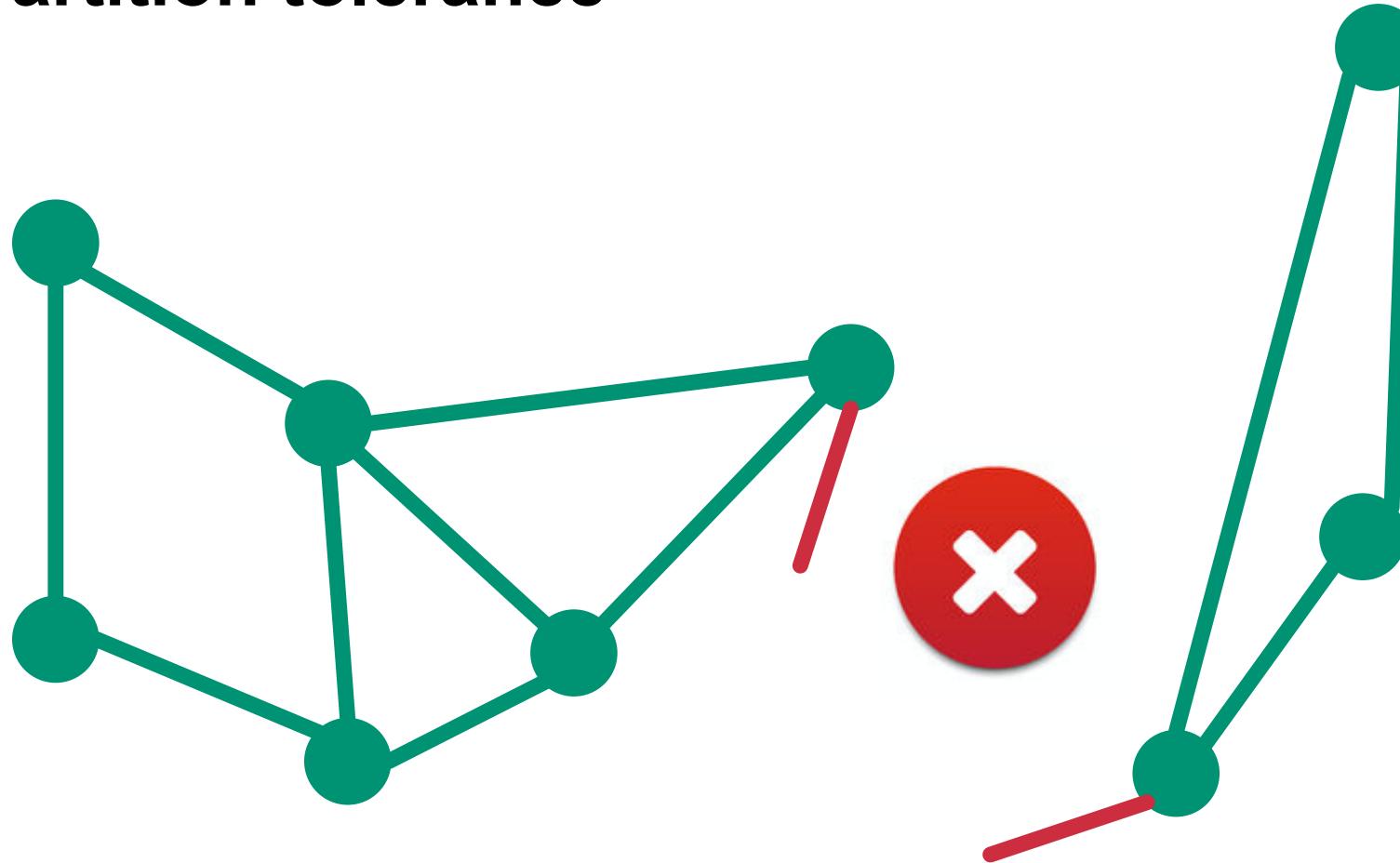


It is possible to query the database at all times.

Consistency



Partition tolerance



The database continues to function even if the network gets partitioned.



REST APIs

HTTP protocol



Sir Tim Berners-Lee



Version	RFC
1.0	RFC 2616
1.1	RFC 7230
2.0	RFC 7540

Resources



Resource (URI)

<http://www.ethz.ch/>

<http://www.mywebsite.ch/api/collection/foo/object/bar>

urn:isbn:0123456789

<mailto:sheldon.lee.cooper@ethz.ch>

Resources



Resource (URL)

<http://www.ethz.ch/>

<http://www.mywebsite.ch/api/collection/foo/object/bar>

Resources



Resource (URN)

urn:isbn:0123456789

<mailto:sheldon.lee.cooper@ethz.ch>

Resources



Resource (URI)

`http://www.mywebsite.ch/api/collection/foo/object/bar?id=foobar#head`

Resources



Resource (URI)

<http://www.mywebsite.ch/api/collection/foo/object/bar?id=foobar#head>
scheme

Resources



Resource (URI)

`http://www.mywebsite.ch/api/collection/foo/object/bar?id=foobar#head
authority`

Resources



Resource (URI)

`http://www.mywebsite.ch/api/collection/foo/object/bar?id=foobar#head
path`

Resources



Resource (URI)

`http://www.mywebsite.ch/api/collection/foo/object/bar?id=foobar#head`
query

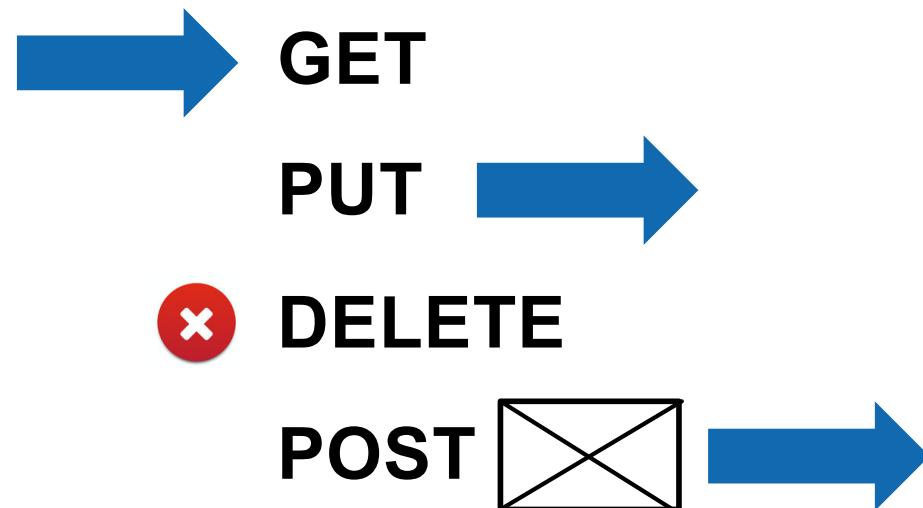
Resources



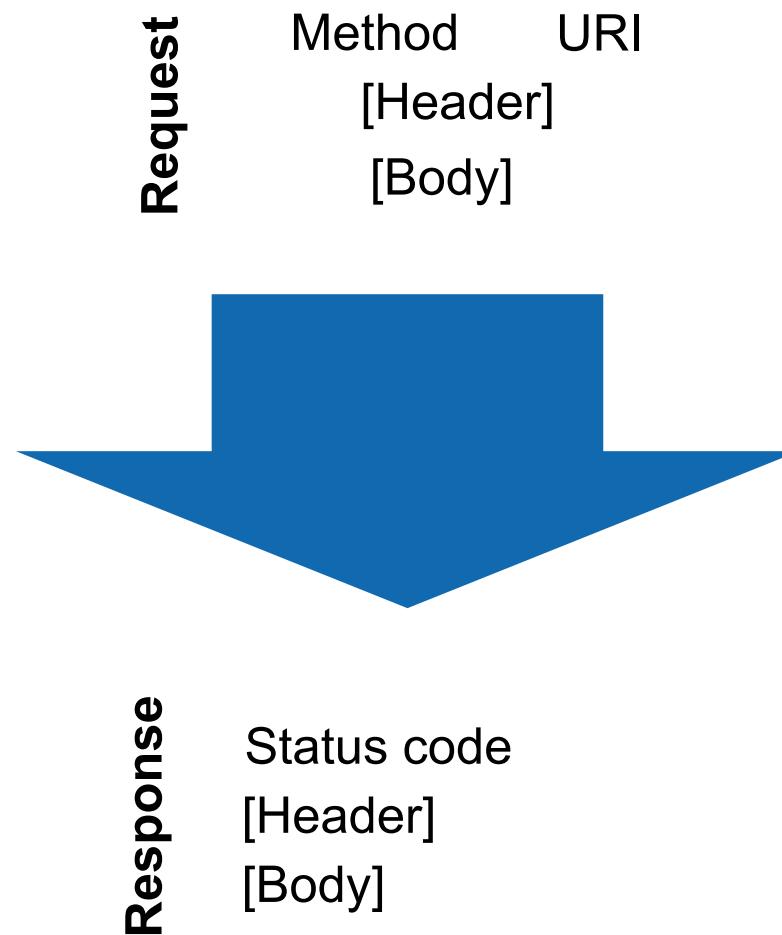
Resource (URI)

`http://www.mywebsite.ch/api/collection/foo/object/bar?id=foobar#head
fragment`

HTTP Methods



HTTP Protocol



Example

GET /index.html HTTP/1.1

Host: www.example.com



HTTP/1.1 200 OK

Date: Tue, 25 Sep 2018 09:48:34 GMT

Content-Type: text/html; charset=UTF-8

Content-Length: 138

Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT

Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)

ETag: "3f80f-1b6-3e1cb03b"

Accept-Ranges: bytes

Connection: close

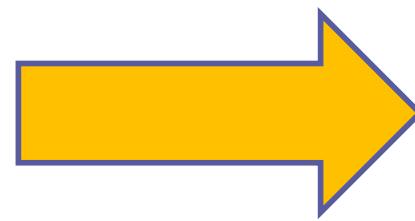
```
<html> <head> <title>An Example Page</title>
</head> <body> Hello World, this is a very simple
HTML document. </body> </html>
```

GET



(Side-effect free)

PUT

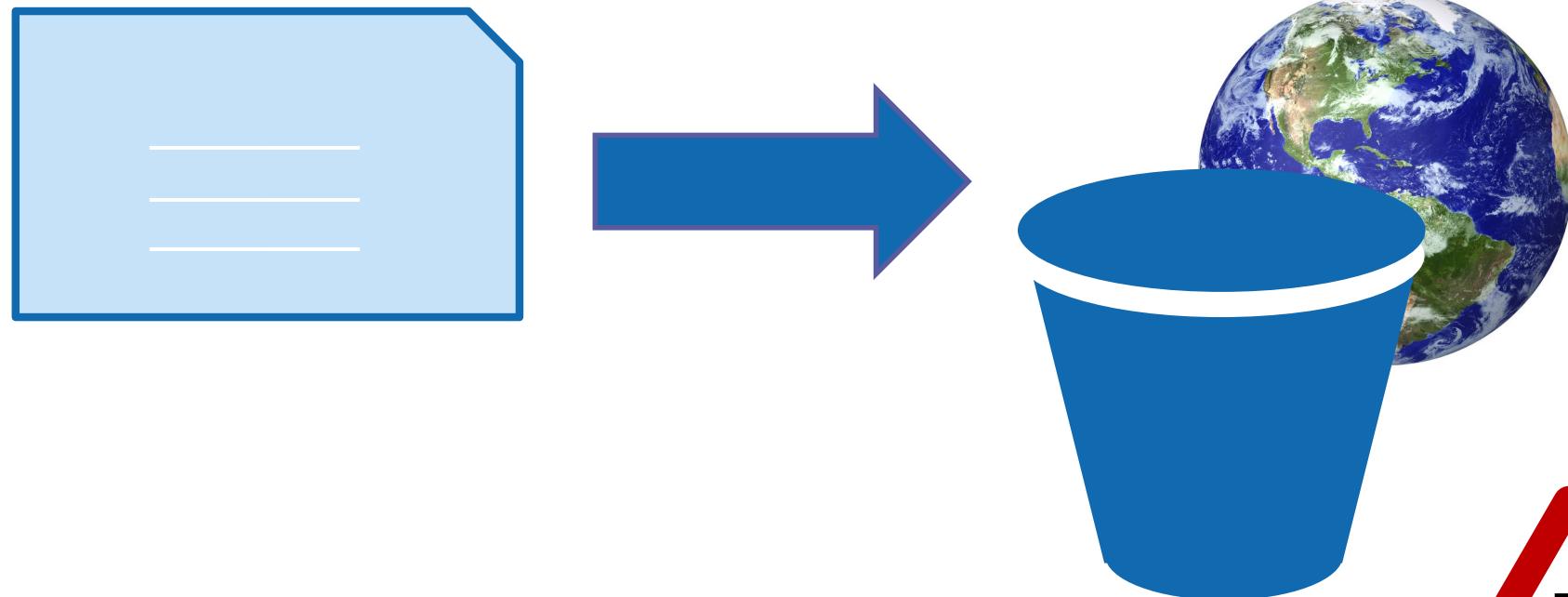


(Idempotent)

DELETE



POST



Most generic: side effects



REST with S3: Buckets



[http://*bucket*.s3.amazonaws.com](http://bucket.s3.amazonaws.com)

REST with S3: Objects



`http://bucket.s3.amazonaws.com/object-name`

S3 REST API



PUT Bucket
DELETE Bucket
GET Bucket

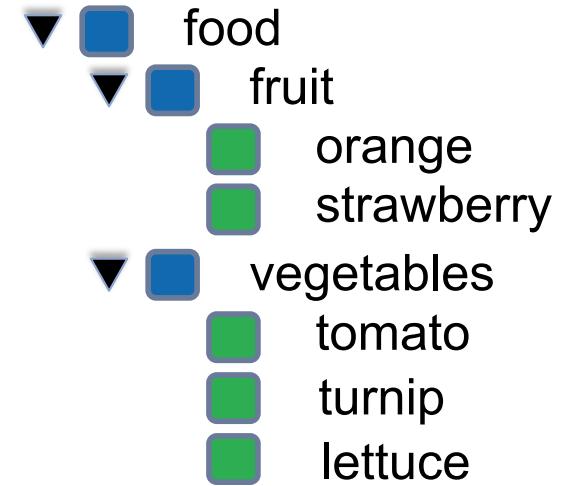


PUT Object
DELETE Object
GET Object

Folders: is S3 a file system?

Logical
(Browsing)

Physical
(Object keys)



/food/fruits/orange
/food/fruits/strawberry
/food/vegetables/tomato
/food/vegetables/turnip
/food/vegetables/lettuce

Static website hosting

`http://<bucket-name>.s3-website-us-east-1.amazonaws.com/`

The JSON Query Language

Decades of Lessons Learnt
JSONiq is a query and processing language specifically designed for the popular JSON data model. The main ideas behind JSONiq are based on lessons learnt in more than 40 years of relational query systems and more than 20 years of experience with designing and implementing query languages for semi-structured data.

Complex Processing
A JSONiq program is an expression; the result of the program is the result of the evaluation of the expression. Expressions have fundamental role in the language: every language construct is an expression, and expressions are fully composable. Project, Filter, Join, Group... Like SQL, JSONiq can do all that.

The SQL of NoSQL
JSONiq is an expressive and highly optimizable language to query and update NoSQL stores. It enables developers to leverage the same productive high-level language across a variety of NoSQL products.

Status

As of 2019, the JSONiq specifications are **stable and maintained**. They have almost not changed since 2013, as version 1.0 was published. Support is provided on StackOverflow.

Version 0.42 of the specification of the JSONiq extension to XQuery is kept online (although deprecated) as it is used in IBM Websphere.

JSONiq has two brothers! Look at our brand new TYSON specification -- JSON with type annotations -- as well as the JSound 2.0 schema language.

The Syntax

Dataset hosting

Overview

Type a prefix and press Enter to search. Press ESC to clear.

[Upload](#) [+ Create folder](#) [Download](#) [Actions ▾](#) EU (Frankfurt) [⚙️](#)

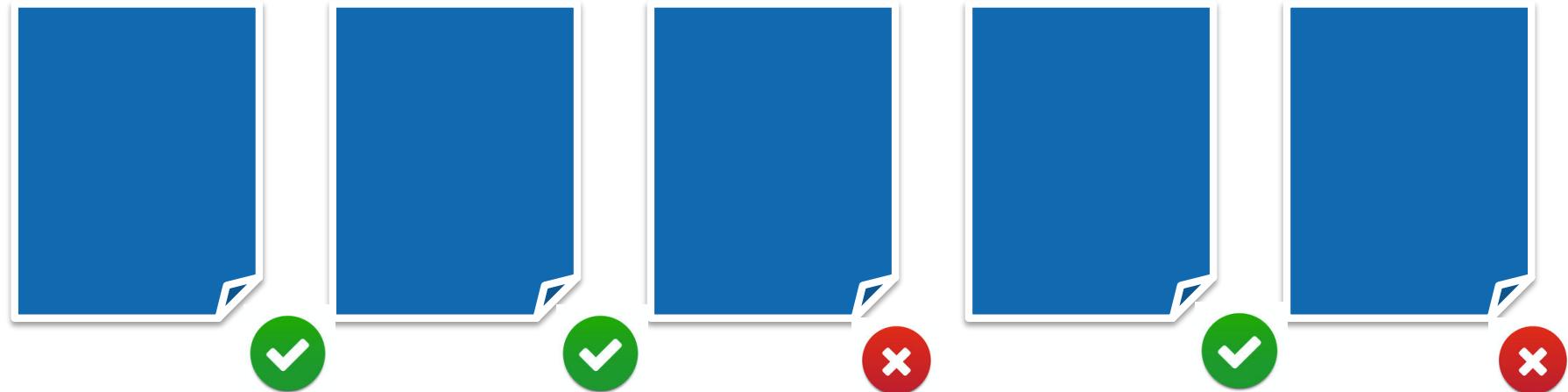
<input type="checkbox"/> Name ▾	Last modified ▾	Size ▾	Storage class ▾
<input type="checkbox"/> xaa	Oct 16, 2019 10:54:22 AM GMT+0200	548.4 MB	Standard
<input type="checkbox"/> xab	Oct 16, 2019 10:54:22 AM GMT+0200	545.7 MB	Standard
<input type="checkbox"/> xac	Oct 16, 2019 10:54:22 AM GMT+0200	565.0 MB	Standard
<input type="checkbox"/> xad	Oct 16, 2019 10:54:23 AM GMT+0200	557.1 MB	Standard
<input type="checkbox"/> xae	Oct 16, 2019 10:54:22 AM GMT+0200	550.9 MB	Standard
<input type="checkbox"/> xaf	Oct 16, 2019 10:54:33 AM GMT+0200	556.4 MB	Standard
<input type="checkbox"/> xag	Oct 16, 2019 10:54:33 AM GMT+0200	554.3 MB	Standard
<input type="checkbox"/> xah	Oct 16, 2019 10:54:34 AM GMT+0200	571.0 MB	Standard
<input type="checkbox"/> xai	Oct 16, 2019 10:54:34 AM GMT+0200	558.9 MB	Standard
<input type="checkbox"/> xaj	Oct 16, 2019 10:54:34 AM GMT+0200	567.9 MB	Standard
<input type="checkbox"/> xak	Oct 16, 2019 10:54:44 AM GMT+0200	563.1 MB	Standard
<input type="checkbox"/> xal	Oct 16, 2019 10:54:45 AM GMT+0200	566.7 MB	Standard
<input type="checkbox"/> xam	Oct 16, 2019 10:54:45 AM GMT+0200	570.4 MB	Standard
<input type="checkbox"/> xan	Oct 16, 2019 10:54:46 AM GMT+0200	562.0 MB	Standard
<input type="checkbox"/> xao	Oct 16, 2019 10:54:46 AM GMT+0200	567.7 MB	Standard
<input type="checkbox"/> xap	Oct 16, 2019 10:54:56 AM GMT+0200	561.5 MB	Standard

Viewing 1 to 54



More on Storage

Replication



Fault tolerance

Faults



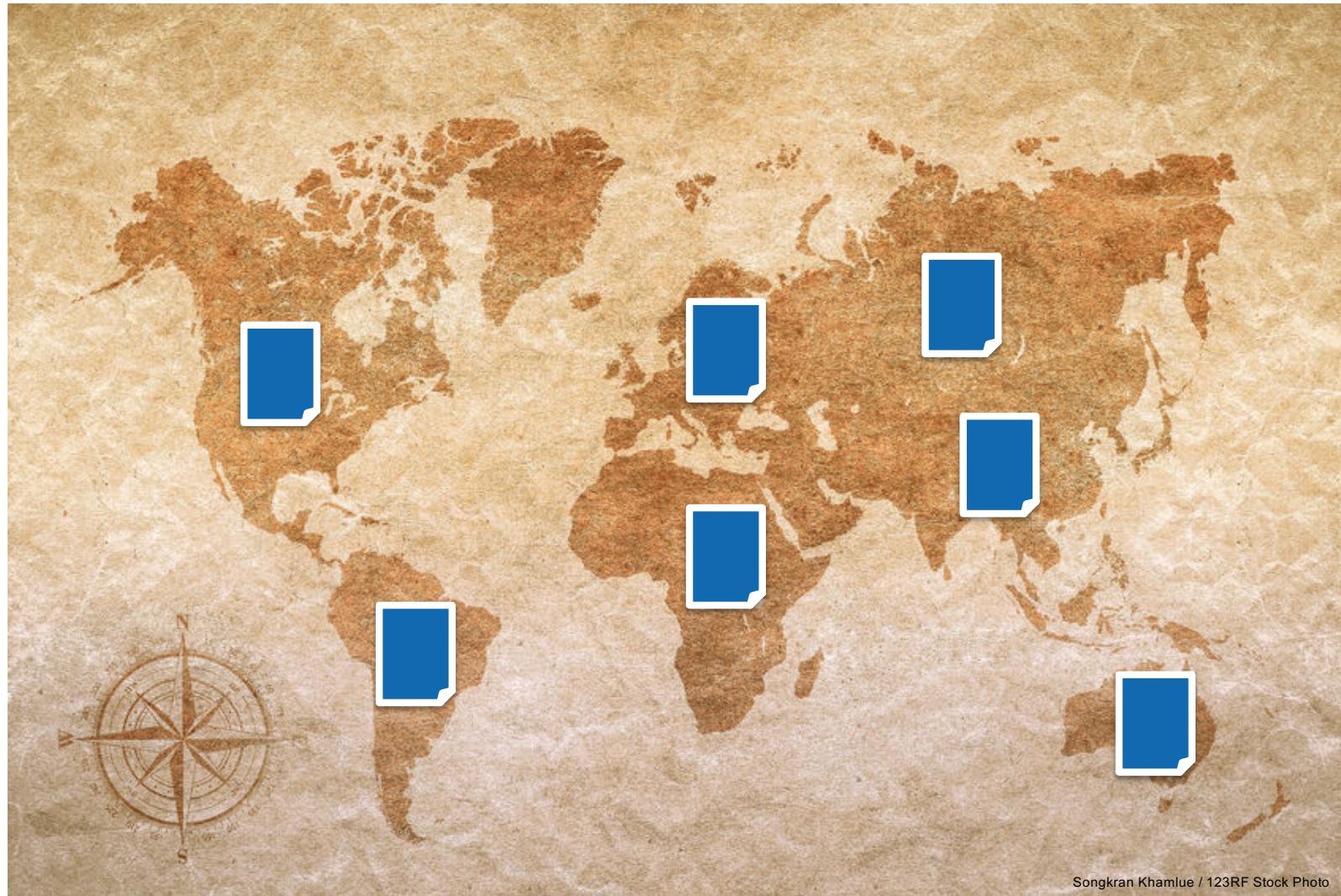
Local (node failure)

versus



Regional (natural catastrophe)

Regions



Songkran Khamlue / 123RF Stock Photo

Regions

1. Optimize latency

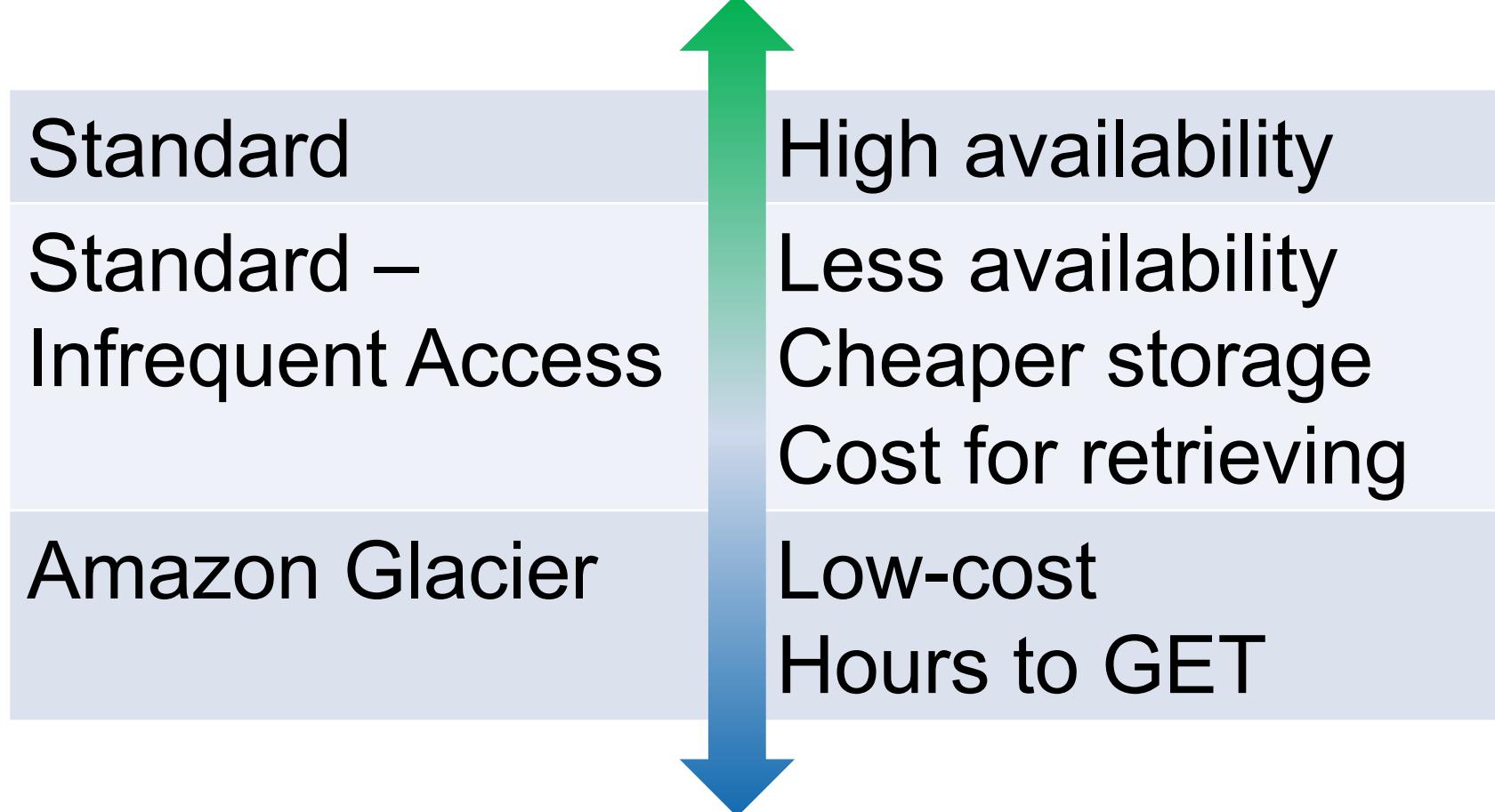


Songkran Khamlue / 123RF Stock Photo

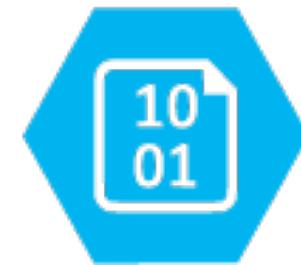
Regions



Storage Class



Microsoft Azure Blob Storage

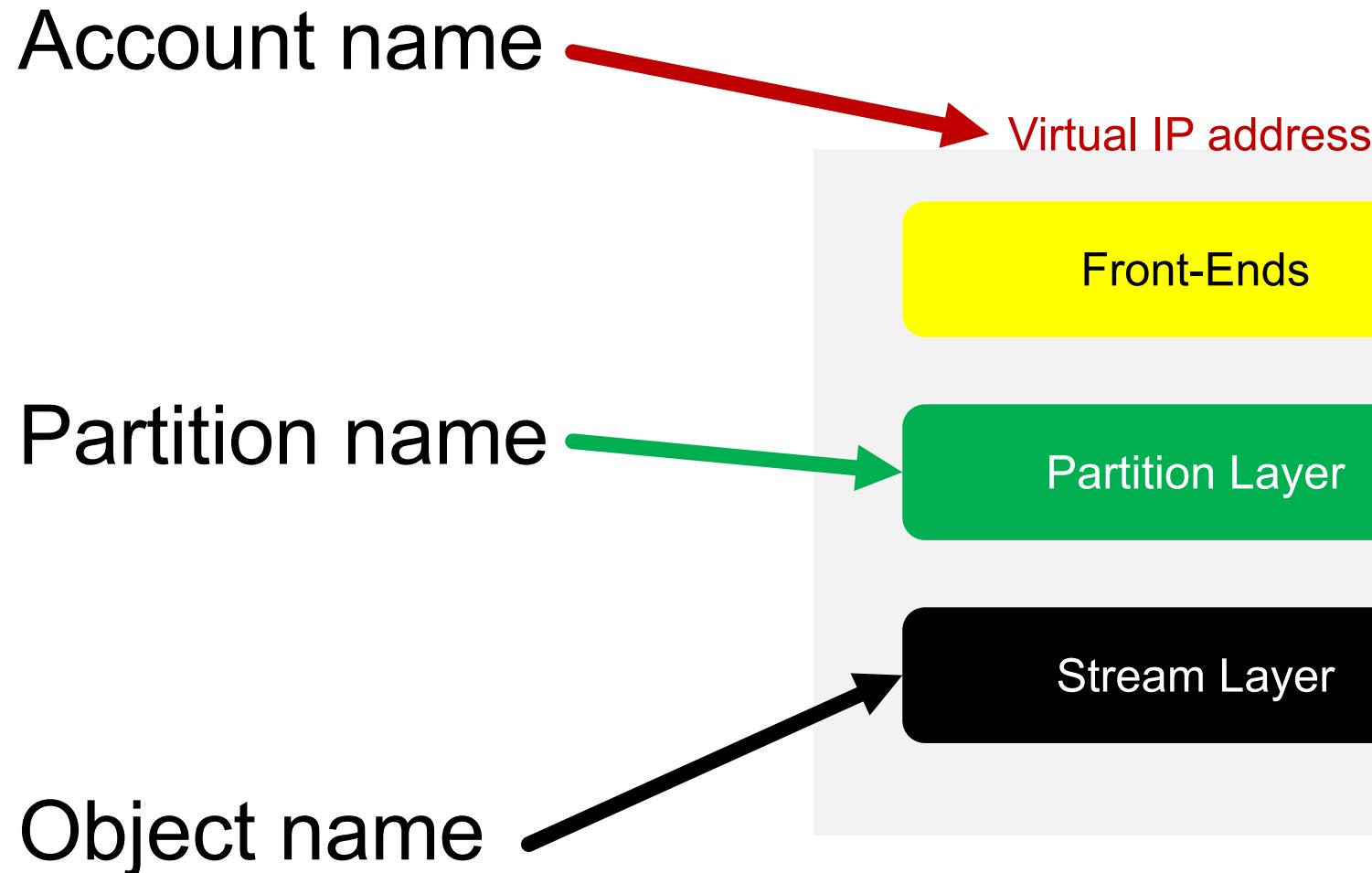


Azure Blob Storage

Overall comparison Azure vs. S3

	S3	Azure
Object ID	Bucket + Object	Account + Container + Blob
Object API	Blackbox	Block/Append/Page
Limit	5 TB	4.78 TB (block) 195 GB (append) 8 TB (page)

Azure Architecture: Storage Stamp



Azure Architecture: One storage stamp



10-20 racks * 18 storage nodes/rack (30PB)



Azure Architecture: Keep some buffer

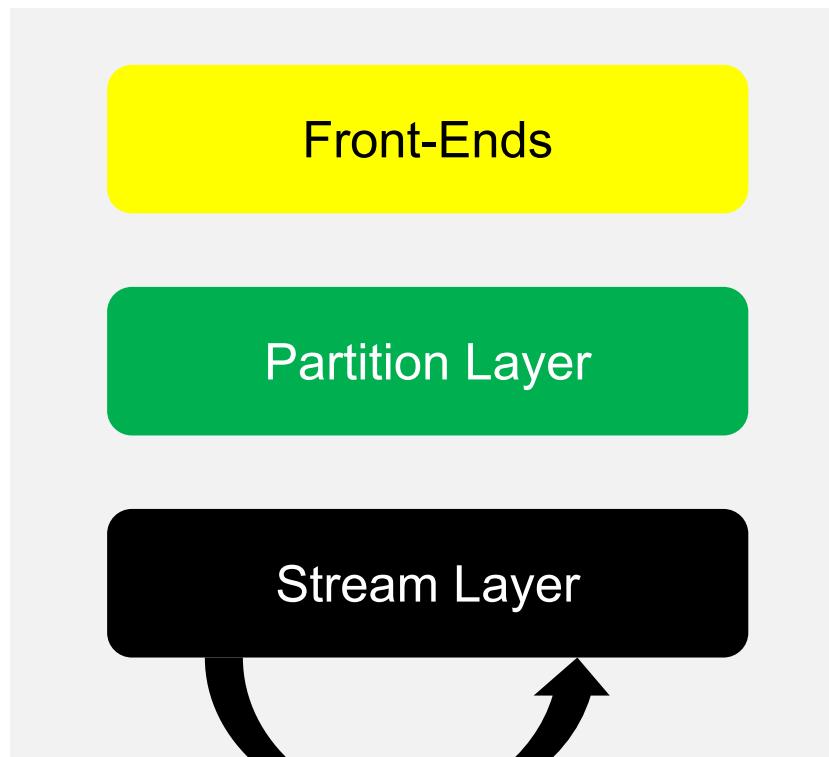


kept below 70/80% storage capacity



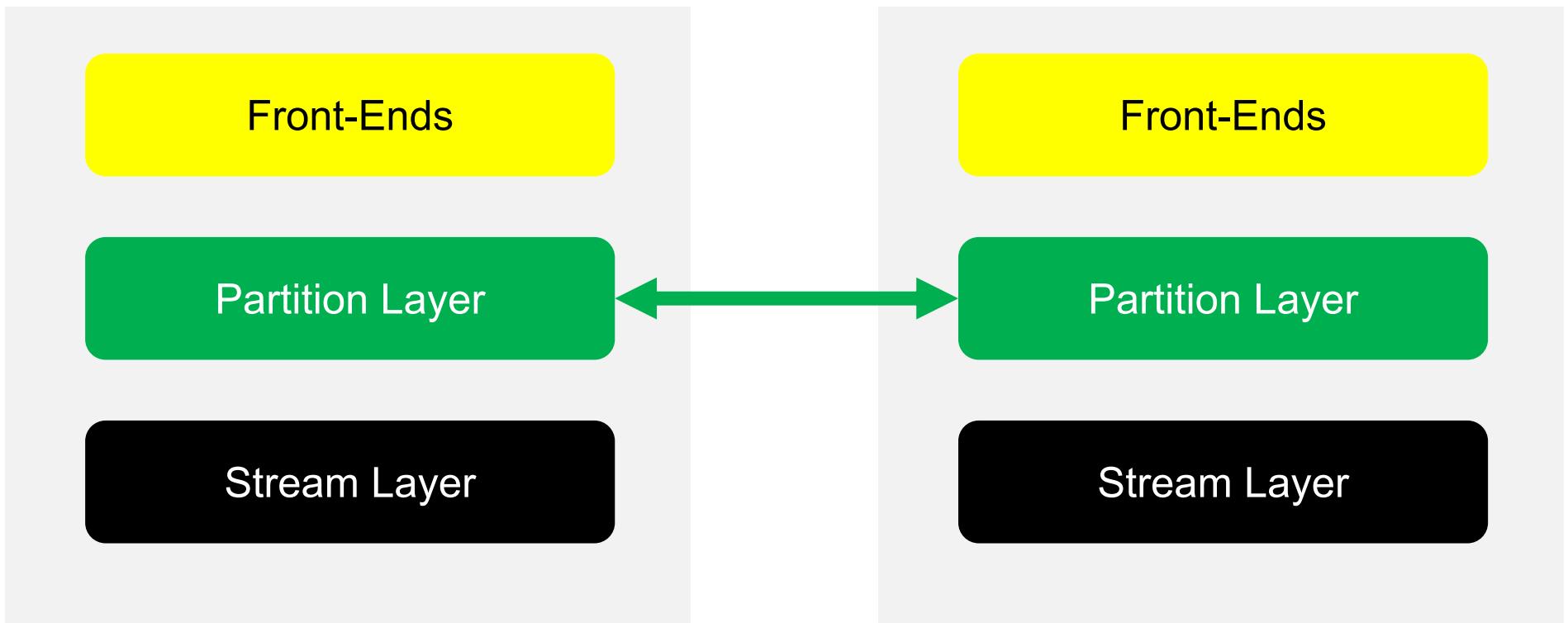
Storage Replication

Intra-stamp replication (**synchronous**)

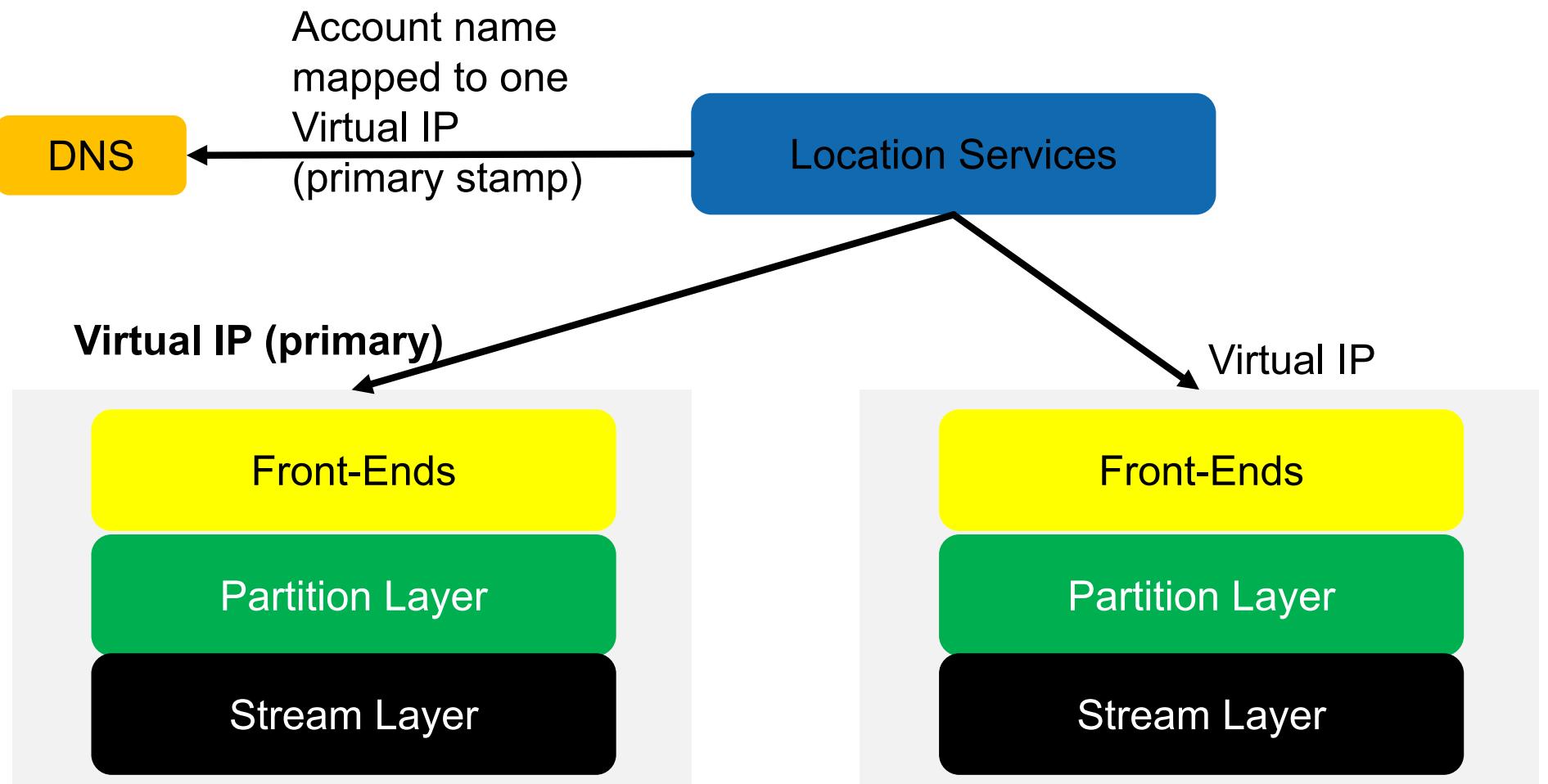


Storage Replication

Inter-stamp replication (**asynchronous**)



Location Services

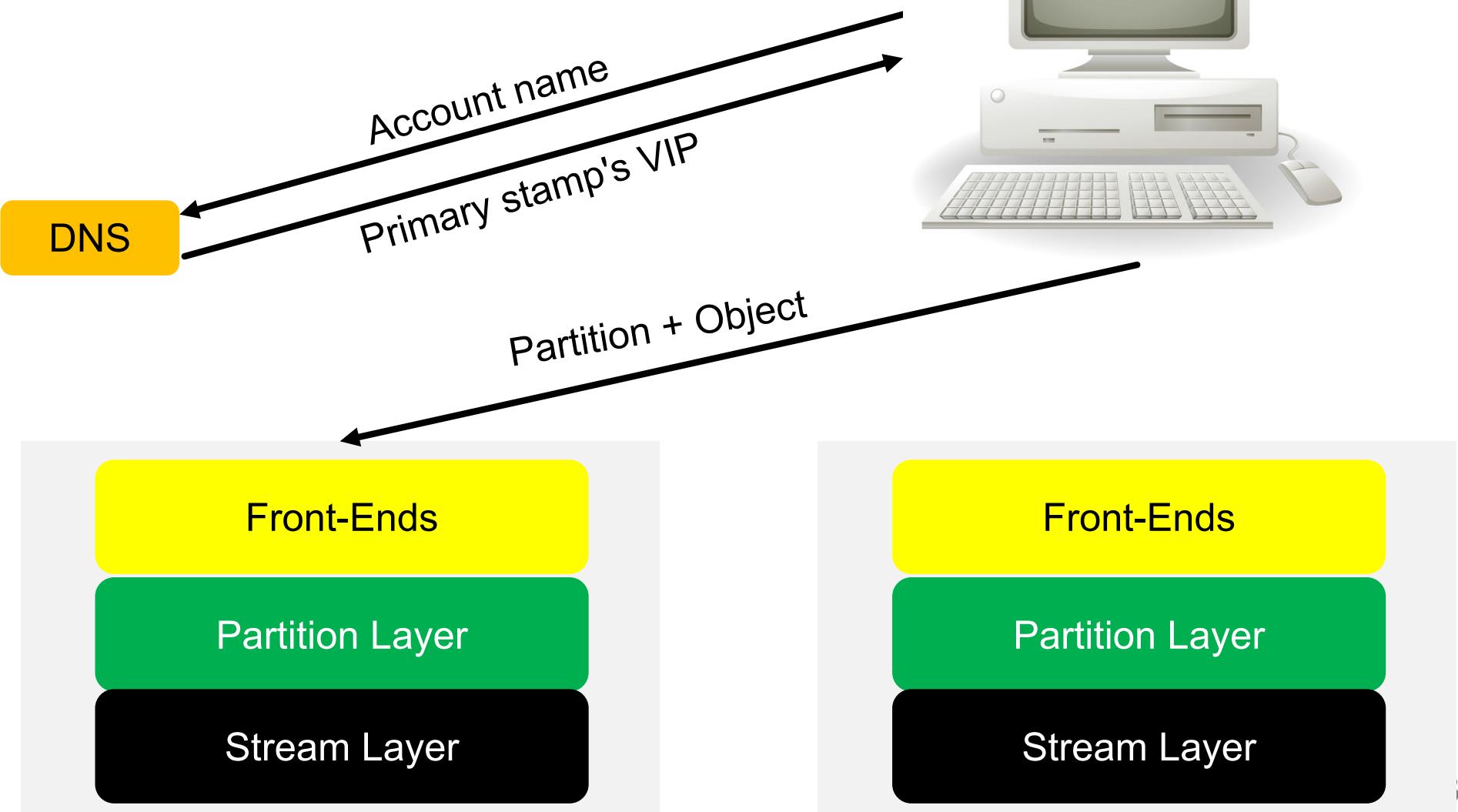


Location Services



North America
Europe
Asia

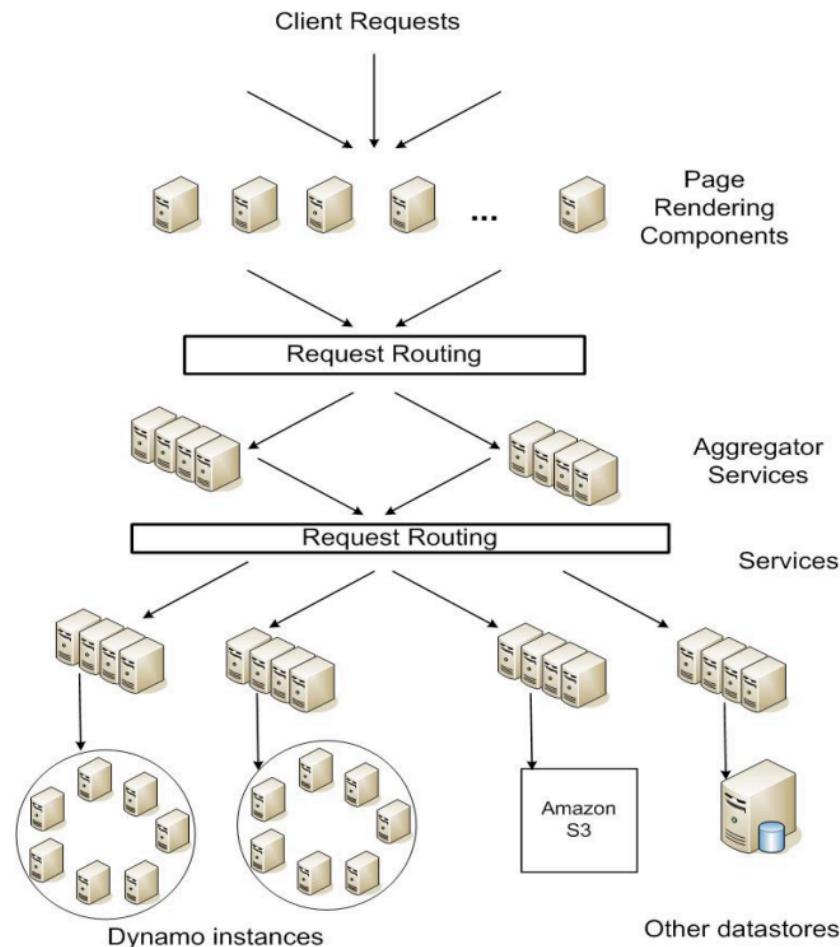
Location Services



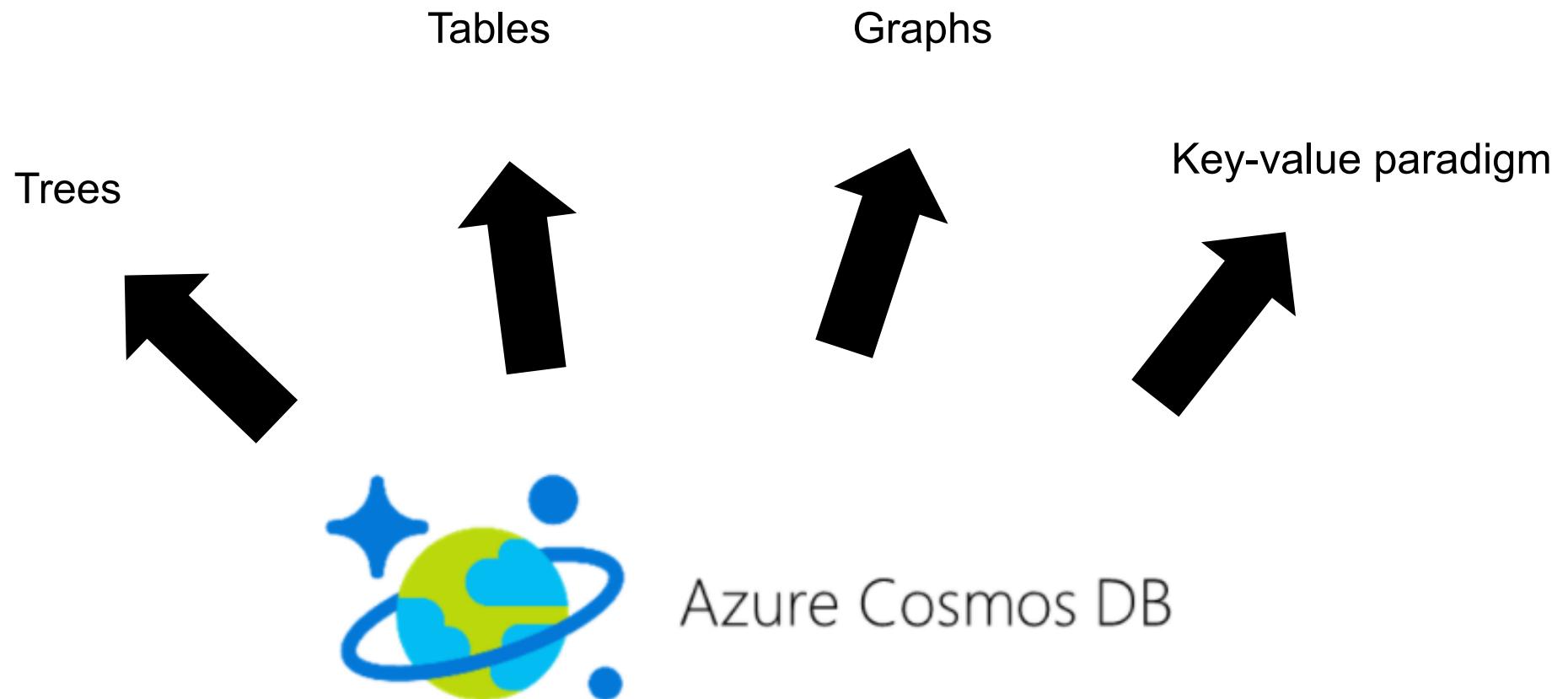


Last words

Amazon mindset



Azure mindset



Take away messages: how to scale out?

- Simplify the model!
- Buy cheap hardware!
- Remove schemas!