Datomic vs. Crux
and why it matters

@tiagoluchini

https://luchini.nyc

TIAGO LUCHINI

@tiagoluchini
https://luchini.nyc

Greenland
Iceland
Sweden
Norway
Russia
Canada
Poland
Germany
Ukraine
Kazakhstan
Mongolia
France
Italy
Spain
Turkey
United States
Japan
South Korea
North Pacific Ocean
North Atlantic Ocean
Iraq
Iran
Afghanistan
Pakistan
Algeria
Libya
Egypt
Saudi Arabia
India
Mexico
Mali
Niger
Sudan
Chad
Thailand
Nigeria
Ethiopia
Venezuela
Colombia
Kenya
DRC
Indonesia
Papua New Guinea
Tanzania
Brazil
Peru
Angola
Bolivia
Namibia
Madagascar
Botswana
Indian Ocean
Australia
South Pacific Ocean
South Atlantic Ocean
Chile
South Africa
New Zealand
Argentina
Southern Ocean

ANO II - Nº 13
OUTUBRO 1982
Cr$ 350,00

ISSN 0101-3401

Micro Sistemas

A PRIMEIRA REVISTA BRASILEIRA DE MICROCOMPUTADORES

UM ANO
DE MICRO SISTEMAS

VisiCalc

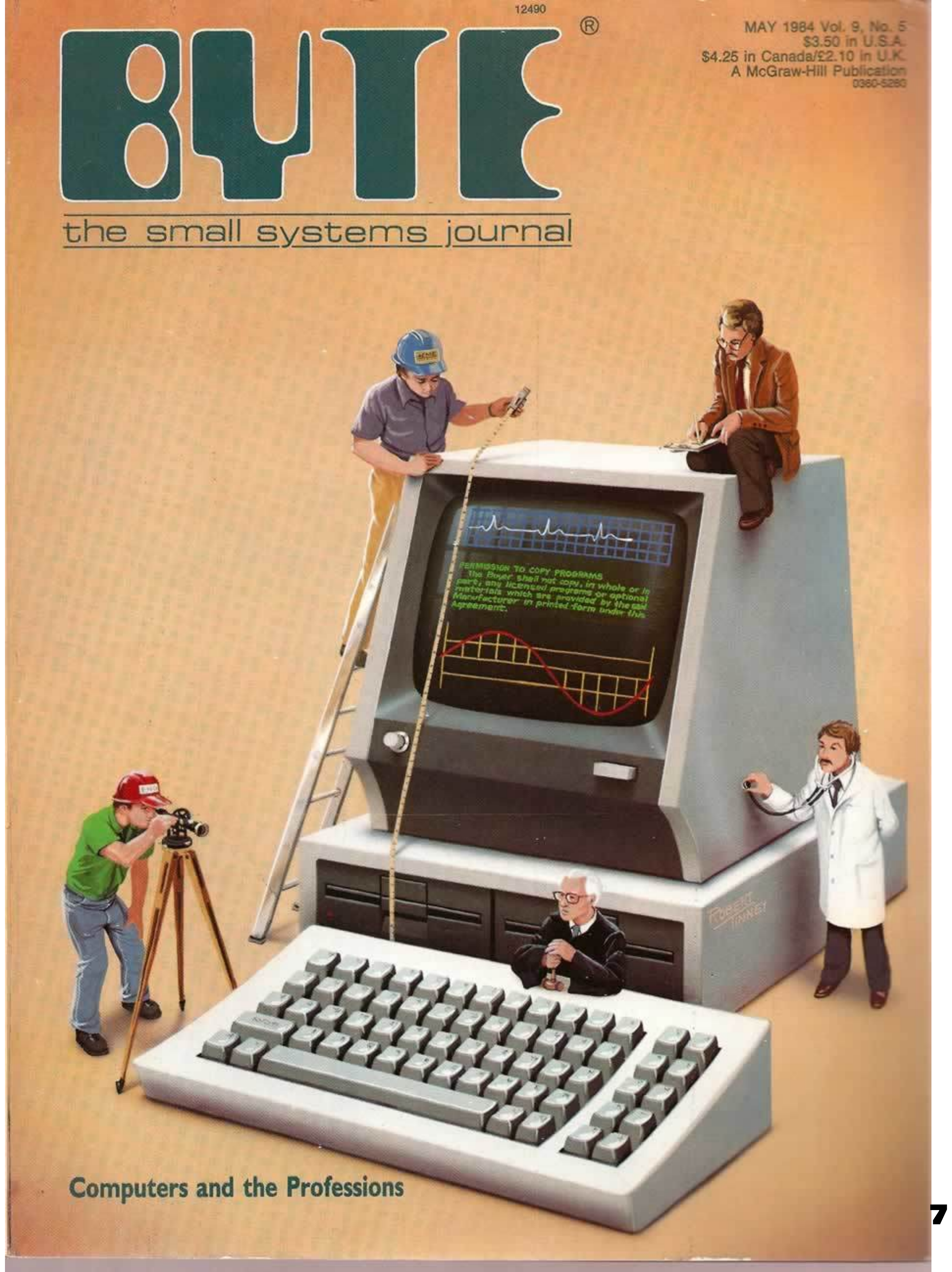
Tecnologia aberta no DEL

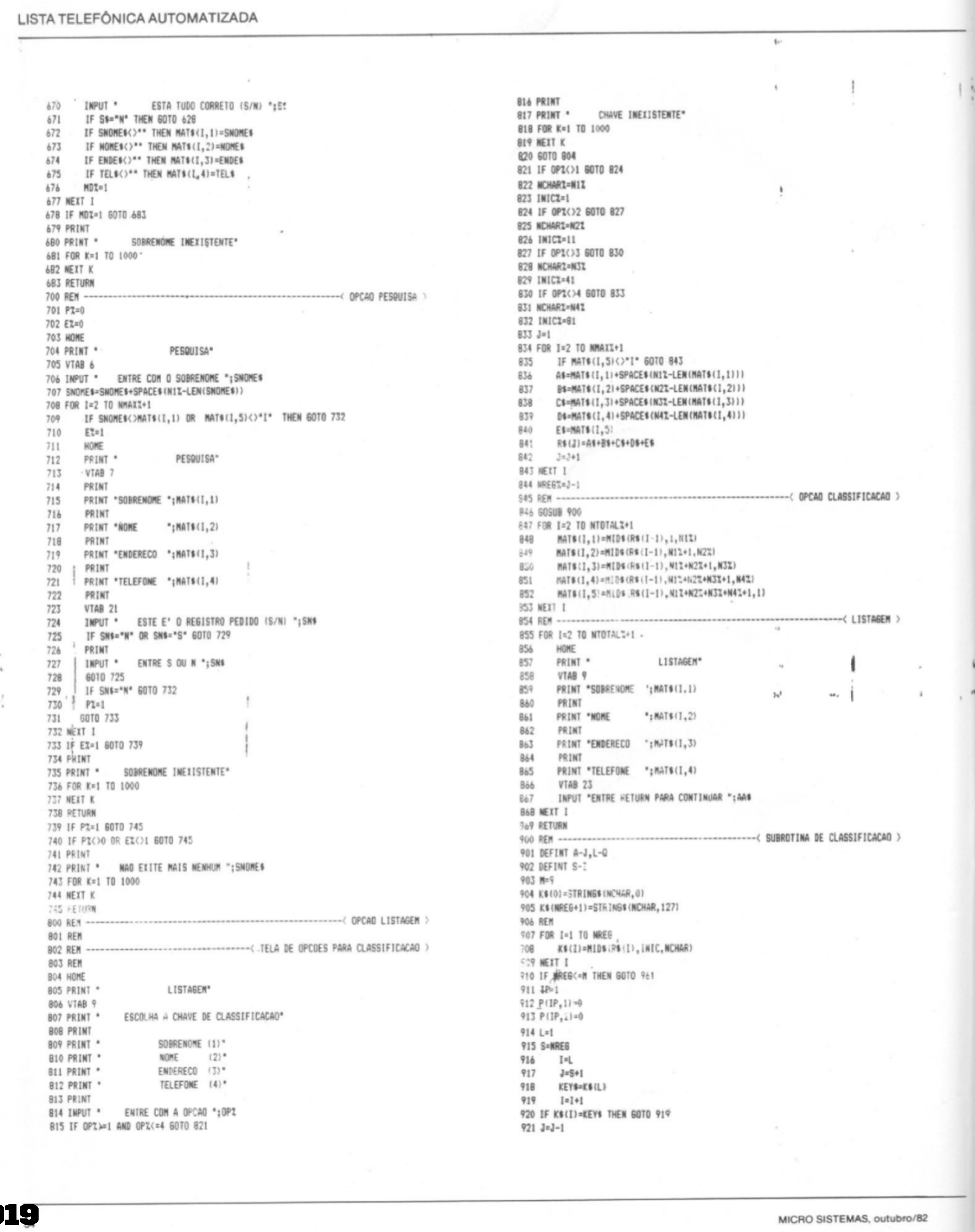
SORTs comparados

Conheça o 6502

Impressoras

ANO II - Nº 13
OUTUBRO 1982
Cr$ 350,00
ISSN 0101-3401
Micro Sistemas
A PRIMEIRA REVISTA BRASILEIRA DE MICROCOMPUTADORES
UM ANO DE MICRO SISTEMAS
VisiCalc
Tecnologia aberta na DEI
SORTs comparados
Conheça o 6502
Impressoras


12490
MAY 1984 Vol. 9, No. 5
$3.50 in U.S.A.
$4.25 in Canada/£2.10 in U.K.
A McGraw-Hill Publication
0360-5280
BYTE
the small systems journal
Computers and the Professions



```
670     INPUT "       ESTA TUDO CORRETO (S/N) ";S$
671     IF S$="N" THEN GOTO 628
672     IF SNOME$<>"" THEN MAT$(I,1)=SNOME$
673     IF NOME$<>"" THEN MAT$(I,2)=NOME$
674     IF ENDE$<>"" THEN MAT$(I,3)=ENDE$
675     IF TEL$<>"" THEN MAT$(I,4)=TEL$
676     MD%=1
677 NEXT I
678 IF MD%=1 GOTO 683
679 PRINT
680 PRINT "       SOBRENOME INEXISTENTE"
681 FOR K=1 TO 1000
682 NEXT K
683 RETURN
700 REM ------------------------------------------------------( OPCAO PESQUISA )
701 P%=0
702 E%=0
703 HOME
704 PRINT "                PESQUISA"
705 VTAB 6
706 INPUT "    ENTRE COM O SOBRENOME ";SNOME$
707 SNOME$=SNOME$+SPACE$(N1%-LEN(SNOME$))
708 FOR I=2 TO NMAX%+1
709     IF SNOME$<>MAT$(I,1) OR  MAT$(I,5)<>"I"  THEN GOTO 732
710     E%=1
711     HOME
712     PRINT "            PESQUISA"
713     VTAB 7
714     PRINT
715     PRINT "SOBRENOME ";MAT$(I,1)
716     PRINT
717     PRINT "NOME      ";MAT$(I,2)
718     PRINT
719     PRINT "ENDERECO  ";MAT$(I,3)
720     PRINT
721     PRINT "TELEFONE  ";MAT$(I,4)
722     PRINT
723     VTAB 21
724     INPUT "    ESTE E' O REGISTRO PEDIDO (S/N) ";SN$
725     IF SN$="N" OR SN$="S" GOTO 729
726     PRINT
727     INPUT "    ENTRE S OU N ";SN$
728     GOTO 725
729     IF SN$="N" GOTO 732
730     P%=1
731    GOTO 733
732 NEXT I
733 IF E%=1 GOTO 739
734 PRINT
735 PRINT "     SOBRENOME INEXISTENTE"
736 FOR K=1 TO 1000
737 NEXT K
738 RETURN
739 IF P%=1 GOTO 745
740 IF P%<>0 OR E%<>1 GOTO 745
741 PRINT
742 PRINT "    NAO EXITE MAIS NENHUM ";SNOME$
743 FOR K=1 TO 1000
744 NEXT K
745 RETURN
800 REM ------------------------------------------------------( OPCAO LISTAGEM )
801 REM
802 REM ---------------------------------( TELA DE OPCOES PARA CLASSIFICACAO )
803 REM
804 HOME
805 PRINT "                LISTAGEM"
806 VTAB 9
807 PRINT "     ESCOLHA A CHAVE DE CLASSIFICACAO"
808 PRINT
809 PRINT "             SOBRENOME (1)"
810 PRINT "             NOME      (2)"
811 PRINT "             ENDERECO  (3)"
812 PRINT "             TELEFONE  (4)"
813 PRINT
814 INPUT "     ENTRE COM A OPCAO ";OP%
815 IF OP%>=1 AND OP%<=4 GOTO 821
816 PRINT
817 PRINT "      CHAVE INEXISTENTE"
818 FOR K=1 TO 1000
819 NEXT K
820 GOTO 804
821 IF OP%<>1 GOTO 824
822 NCHAR%=N1%
823 INIC%=1
824 IF OP%<>2 GOTO 827
825 NCHAR%=N2%
826 INIC%=11
827 IF OP%<>3 GOTO 830
828 NCHAR%=N3%
829 INIC%=41
830 IF OP%<>4 GOTO 833
831 NCHAR%=N4%
832 INIC%=81
833 J=1
834 FOR I=2 TO NMAX%+1
835     IF MAT$(I,5)<>"I" GOTO 843
836     A$=MAT$(I,1)+SPACE$(N1%-LEN(MAT$(I,1)))
837     B$=MAT$(I,2)+SPACE$(N2%-LEN(MAT$(I,2)))
838     C$=MAT$(I,3)+SPACE$(N3%-LEN(MAT$(I,3)))
839     D$=MAT$(I,4)+SPACE$(N4%-LEN(MAT$(I,4)))
840     E$=MAT$(I,5)
841     R$(J)=A$+B$+C$+D$+E$
842     J=J+1
843 NEXT I
844 NREG%=J-1
945 REM ----------------------------------------( OPCAO CLASSIFICACAO )
846 GOSUB 900
847 FOR I=2 TO NTOTAL%+1
848     MAT$(I,1)=MID$(R$(I-1),1,N1%)
849     MAT$(I,2)=MID$(R$(I-1),N1%+1,N2%)
850     MAT$(I,3)=MID$(R$(I-1),N1%+N2%+1,N3%)
851     MAT$(I,4)=MID$(R$(I-1),N1%+N2%+N3%+1,N4%)
852     MAT$(I,5)=MID$(R$(I-1),N1%+N2%+N3%+N4%+1,1)
853 NEXT I
854 REM ---------------------------------------------------------( LISTAGEM )
855 FOR I=2 TO NTOTAL%+1
856     HOME
857     PRINT "                LISTAGEM"
858     VTAB 9
859     PRINT "SOBRENOME  ";MAT$(I,1)
860     PRINT
861     PRINT "NOME       ";MAT$(I,2)
862     PRINT
863     PRINT "ENDERECO   ";MAT$(I,3)
864     PRINT
865     PRINT "TELEFONE   ";MAT$(I,4)
866     VTAB 23
867     INPUT "ENTRE RETURN PARA CONTINUAR ";AA$
868 NEXT I
869 RETURN
900 REM -----------------------------------( SUBROTINA DE CLASSIFICACAO )
901 DEFINT A-J,L-Q
902 DEFINT S-Z
903 M=9
904 K$(0)=STRING$(NCHAR,0)
905 K$(NREG+1)=STRING$(NCHAR,127)
906 REM
907 FOR I=1 TO NREG
908     K$(I)=MID$(R$(I),INIC,NCHAR)
909 NEXT I
910 IF NREG<=M THEN GOTO 961
911 IP=1
912 P(IP,1)=0
913 P(IP,2)=0
914 L=1
915 S=NREG
916     I=L
917     J=S+1
918     KEY$=K$(L)
919     I=I+1
920 IF K$(I)=KEY$ THEN GOTO 919
921 J=J-1
```

DOES HE BITE?
NO, BUT HE CAN HURT YOU IN OTHER WAYS

CRUX & DATOMIC

REASONS WHY YOU SHOULD CARE

Reasons why you should care

1. Immutable Database
2. Query Like a Ninja
3. Unbundle Your Database

Reasons why you should care

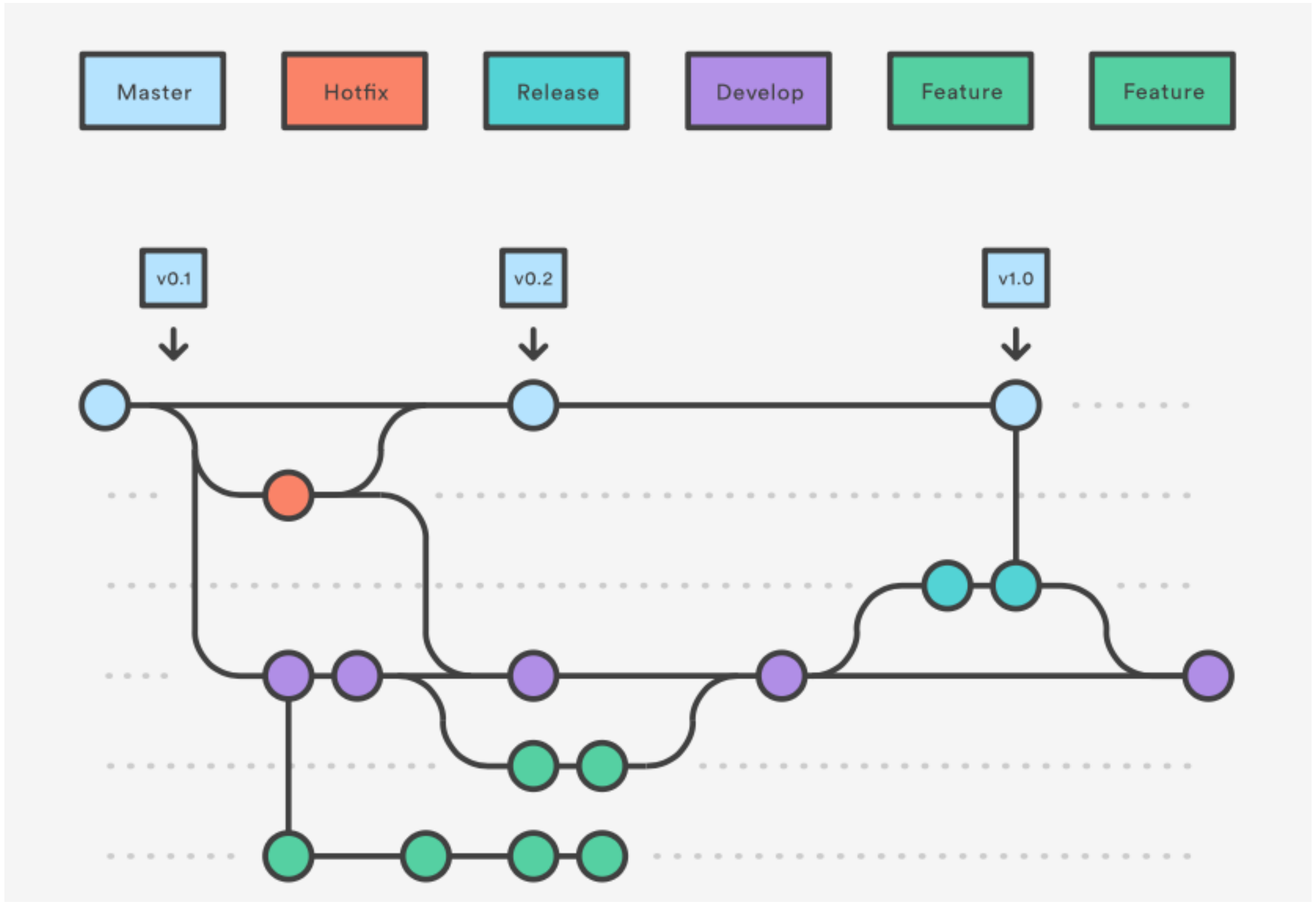
1. Immutable Database
2. Query Like a Ninja
3. Unbundle Your Database

Immutability everywhere

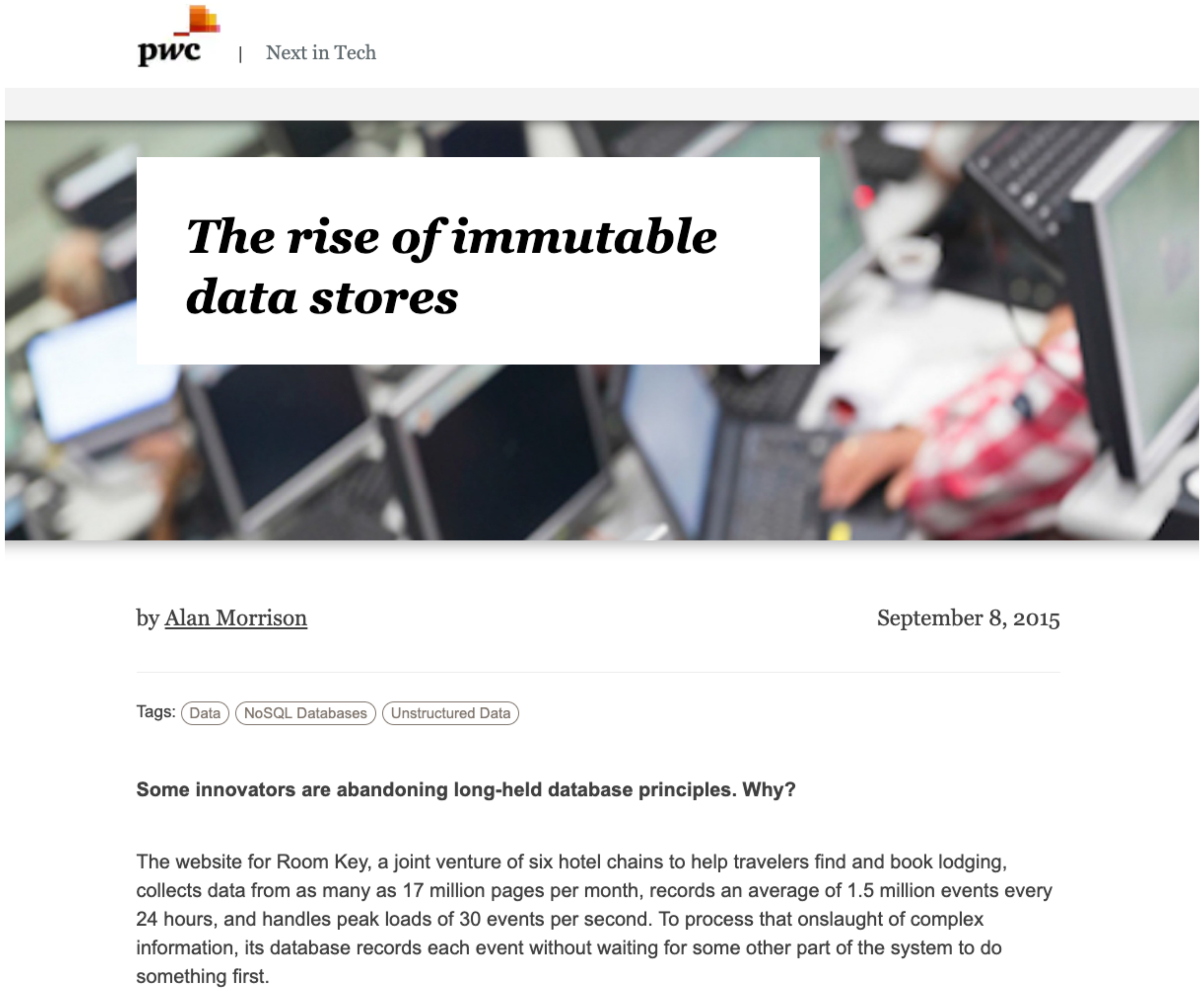
- Haskell, Clojure, Erlang...
- Immutable.js
- CloudFormation, Terraform
- Nix OS

MYTH: DATA IS EPHEMERAL

SEPTEMBER
REPORTS
POTENTIAL CLIENTS
HOLIDAY SCHEDULE
INCOME TAX
UPCOMING EVENTS
APRIL
JANUARY
EMPLOYEES
MORTGAGE
MARCH
MONTHLY EXP
FEBRUARY
VACATION
BUSINESS EXPENSES
BUDGET
DONATIONS
RESUMES
PTO
INSURANCE
TIME CARDS
CHARITIES
CLIENTS
MAILINGS
PROFITS
401 K


Master
Hotfix
Release
Develop
Feature
Feature
v0.1
v0.2
v1.0

ACCRETING FACTS OVER TIME


pwc | Next in Tech

The rise of immutable data stores

by Alan Morrison   September 8, 2015

Tags: Data · NoSQL Databases · Unstructured Data

Some innovators are abandoning long-held database principles. Why?

The website for Room Key, a joint venture of six hotel chains to help travelers find and book lodging, collects data from as many as 17 million pages per month, records an average of 1.5 million events every 24 hours, and handles peak loads of 30 events per second. To process that onslaught of complex information, its database records each event without waiting for some other part of the system to do something first.

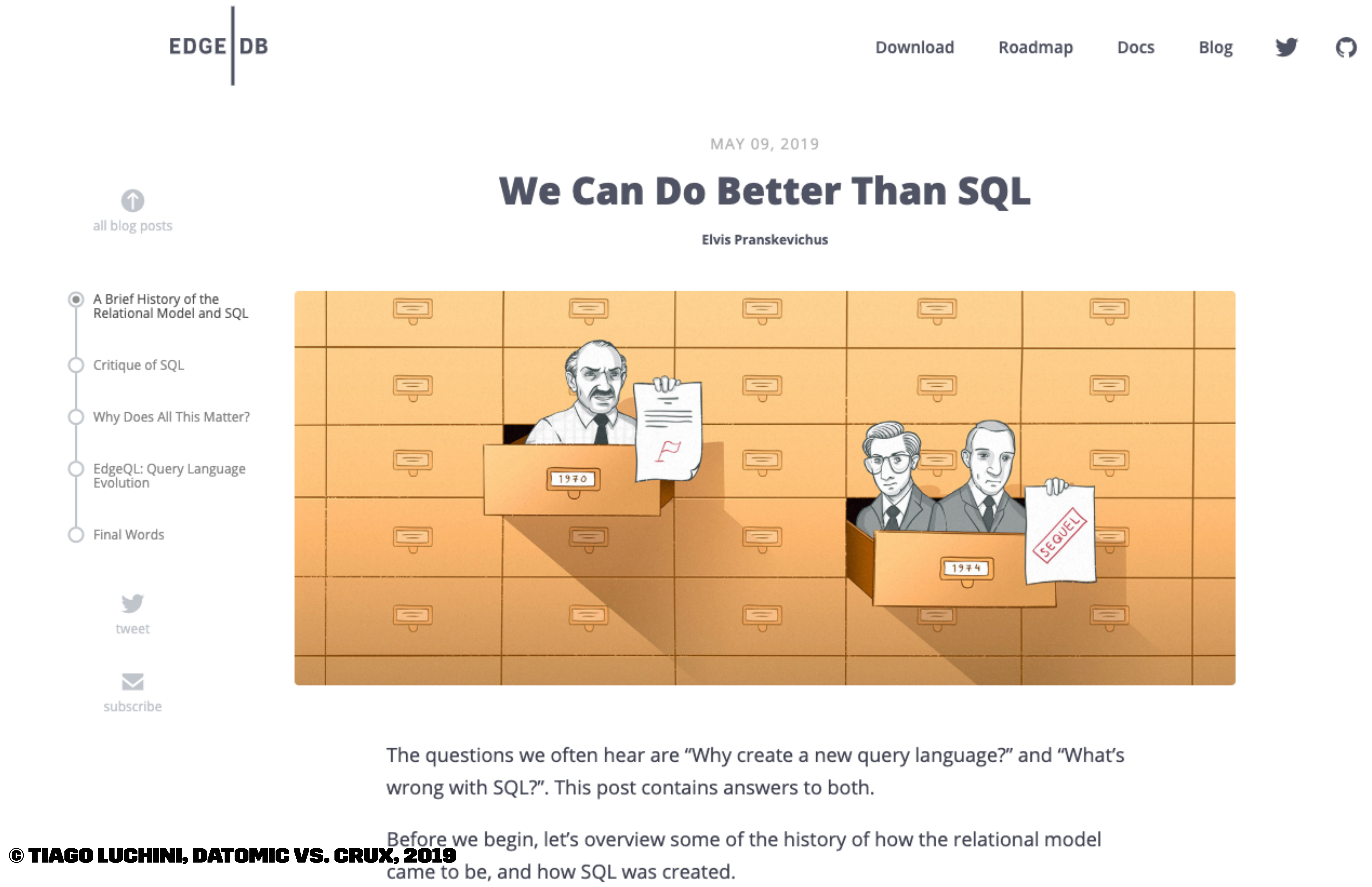
Reasons why you should care

1. Immutable Database
2. Query Like a Ninja
3. Unbundle Your Database

SQL

Critique of SQL

- » lack of proper orthogonality
- » lack of compactness
- » lack of consistency
- » poor system cohesion



EDGE DB

Download Roadmap Docs Blog

MAY 09, 2019

We Can Do Better Than SQL

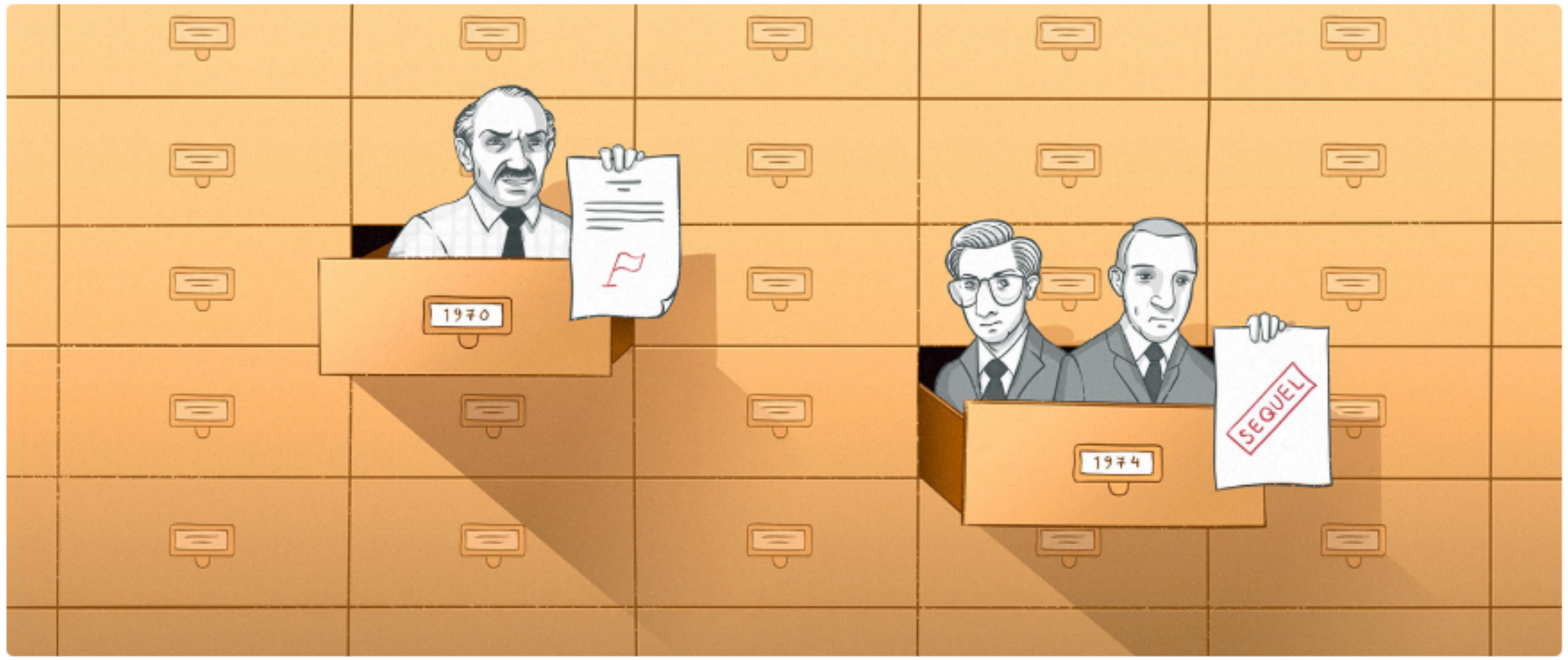
Elvis Pranskevichus

all blog posts

- A Brief History of the Relational Model and SQL
- Critique of SQL
- Why Does All This Matter?
- EdgeQL: Query Language Evolution
- Final Words

tweet

subscribe



The questions we often hear are "Why create a new query language?" and "What's wrong with SQL?". This post contains answers to both.

Before we begin, let's overview some of the history of how the relational model came to be, and how SQL was created.

DATALOG (BOSS SLIDE)

Datalog is a declarative query language with roots in logic programming that combines facts and rules to achieve the same power as relational algebra recursion.

```sql
SELECT greatest_songs.name
FROM artists
INNER JOIN greatest_songsa
ON artists.id = greatest_songs.artist_id
WHERE artists.name = "Luan Santana";
```

JUNTOS E
SHALLOW NOW

```
[:find ?s
 :where
 [?s :greatest-song/artist ?p]
 [?p :artist/name "Luan Santana"]]
```

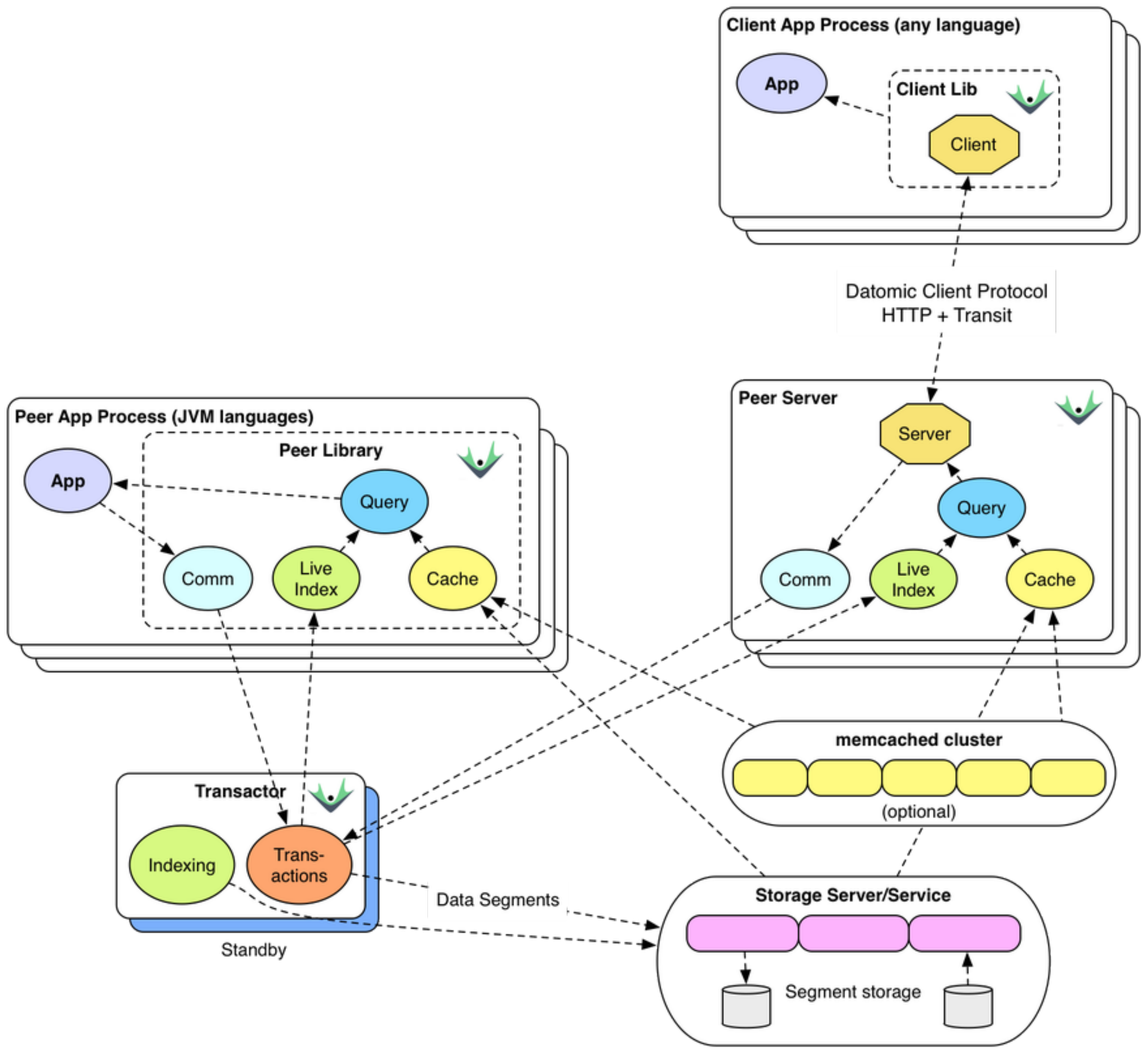
HTTP://WWW.LEARNDATALOGTODAY.ORG

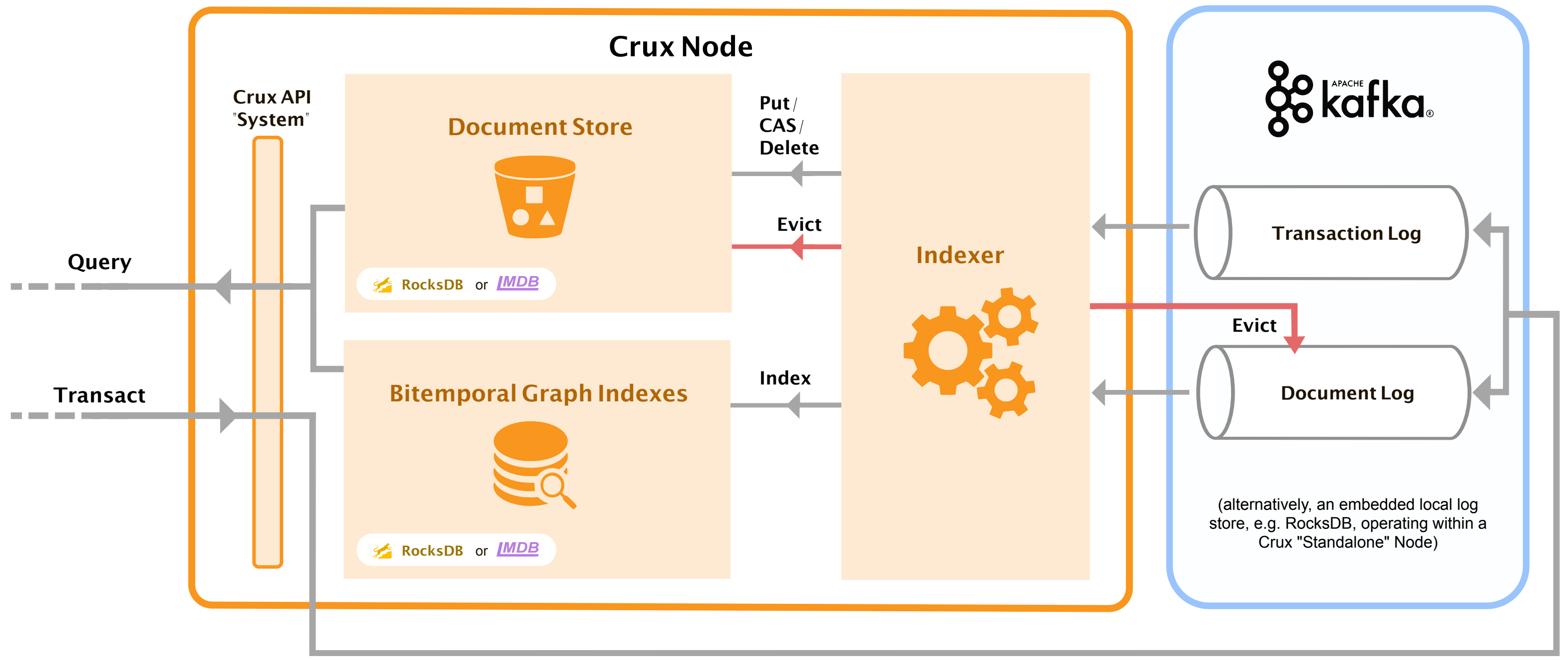
Reasons why you should care

1. "Immutable" Database
2. Query Like a Zen Master
3. <u>The Unbundled Database</u>

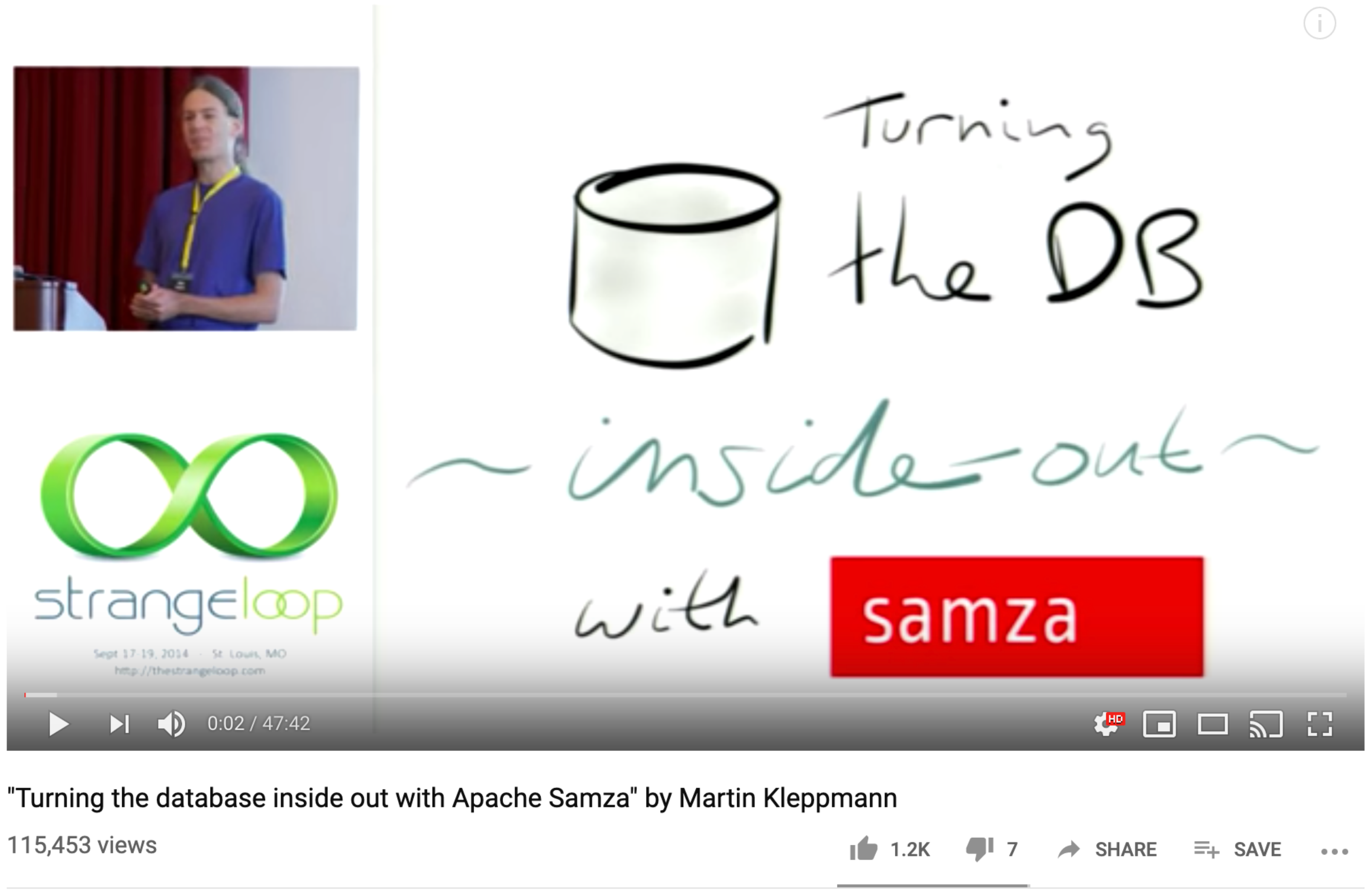
The Unbundled Database (Boss Slide)

“What do we have to gain from turning the database inside out? Simpler code, better scalability, better robustness, lower latency, and more flexibility for doing interesting things with data.”
Martin Kleppmann


Client App Process (any language)
App
Client Lib
Client
Datomic Client Protocol
HTTP + Transit
Peer Server
Server
Query
Comm
Live Index
Cache
Peer App Process (JVM languages)
Peer Library
App
Query
Comm
Live Index
Cache
memcached cluster
(optional)
Transactor
Indexing
Trans-actions
Standby
Data Segments
Storage Server/Service
Segment storage


Crux Node
Crux API `System`
Document Store
RocksDB or LMDB
Bitemporal Graph Indexes
RocksDB or LMDB
Put / CAS / Delete
Evict
Index
Indexer
Query
Transact
kafka
Transaction Log
Evict
Document Log
(alternatively, an embedded local log store, e.g. RocksDB, operating within a Crux "Standalone" Node)

"Turning the Database Inside Out"



"Turning the database inside out with Apache Samza" by Martin Kleppmann

115,453 views

1.2K 7 SHARE SAVE

https://www.youtube.com/watch?v=fU9hR3kiOK0

Reasons Why You Should Care

1. Immutable Database
2. Query Like a Ninka
3. Unbundle Your Database

C R U X

Datomic

P
0 NIN
50000
KO
RYU
DHALSIM
99
FIGHT!
BATTLE 01
arcadegamesfreak

The 10 Critical Things When Talking About Datomic and Crux

1. LICENSING MODEL

1. Licensing Model

- Crux: Open Source (MIT)
- Datomic: Proprietary Only

2. TEMPORAL MODEL

2. Temporal Model

- Crux: Bitemporal
- Datomic: Unitemporal

Datomic's temporal model

- "A datom is an immmutable, point-in-time fact!"
  - Entity
  - Attribute
  - Value
  - Time
- EAVT

Entity	Attribute	Value	Time
`123`	`:person/likes`	`"pizza"`	`1978-07-08T20:19:03.176-00:00`

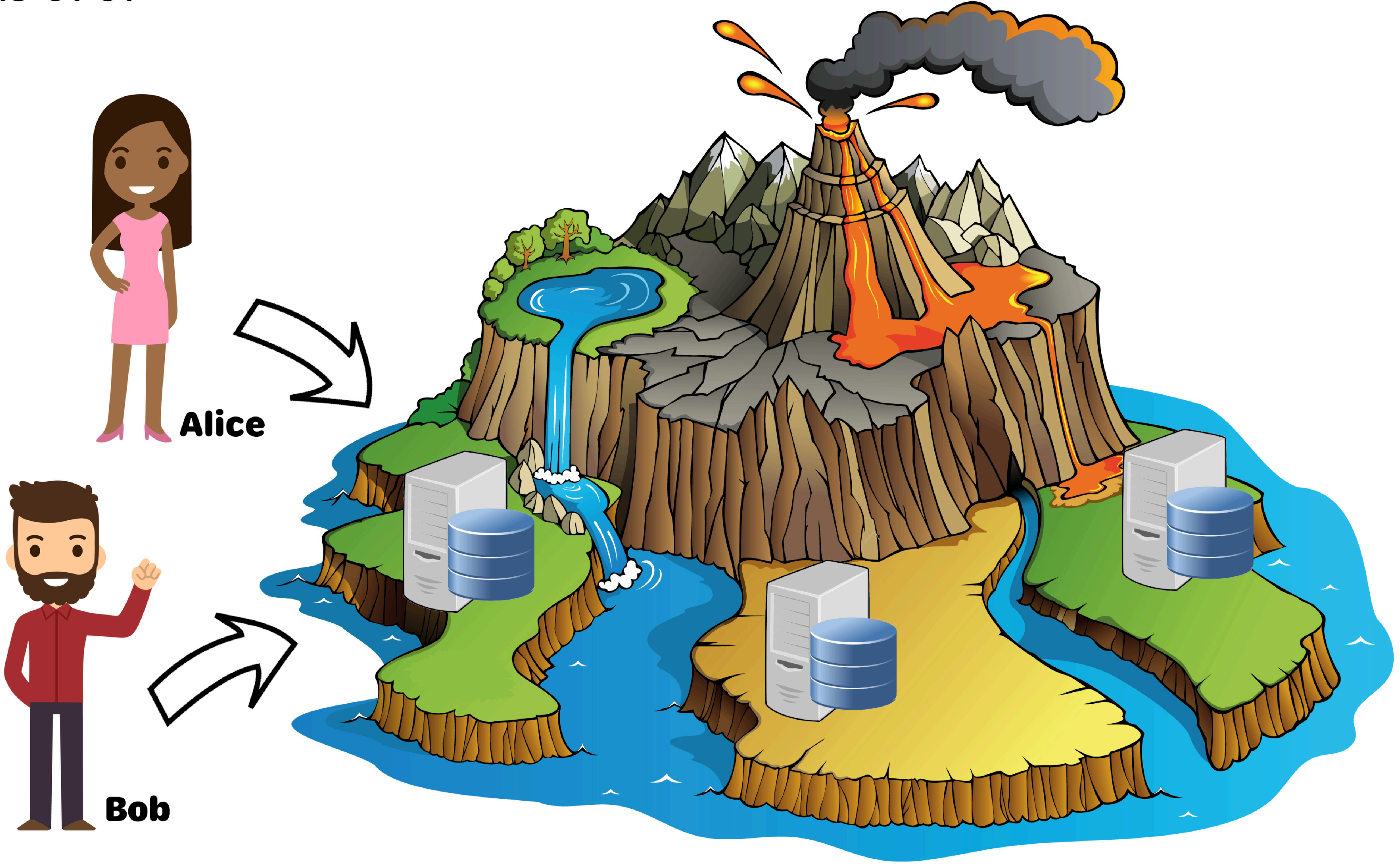
Entity	Attribute	Value	Time
`123`	`:person/likes`	`"pizza"`	`1978-07-08T20:19:03.176-00:00`
`123`	`:person/name`	`"John"`	`1978-07-08T20:19:03.176-00:00`
`123`	`:contact/phone`	`"+1 (305) 555-5524"`	`2009-03-01T14:24:32.074-00:00`
`123`	`:contact/phone`	`"+1 (646) 341-3367"`	`2017-10-25T08:22:06.165-00:00`

Crux's Bitemporal Model

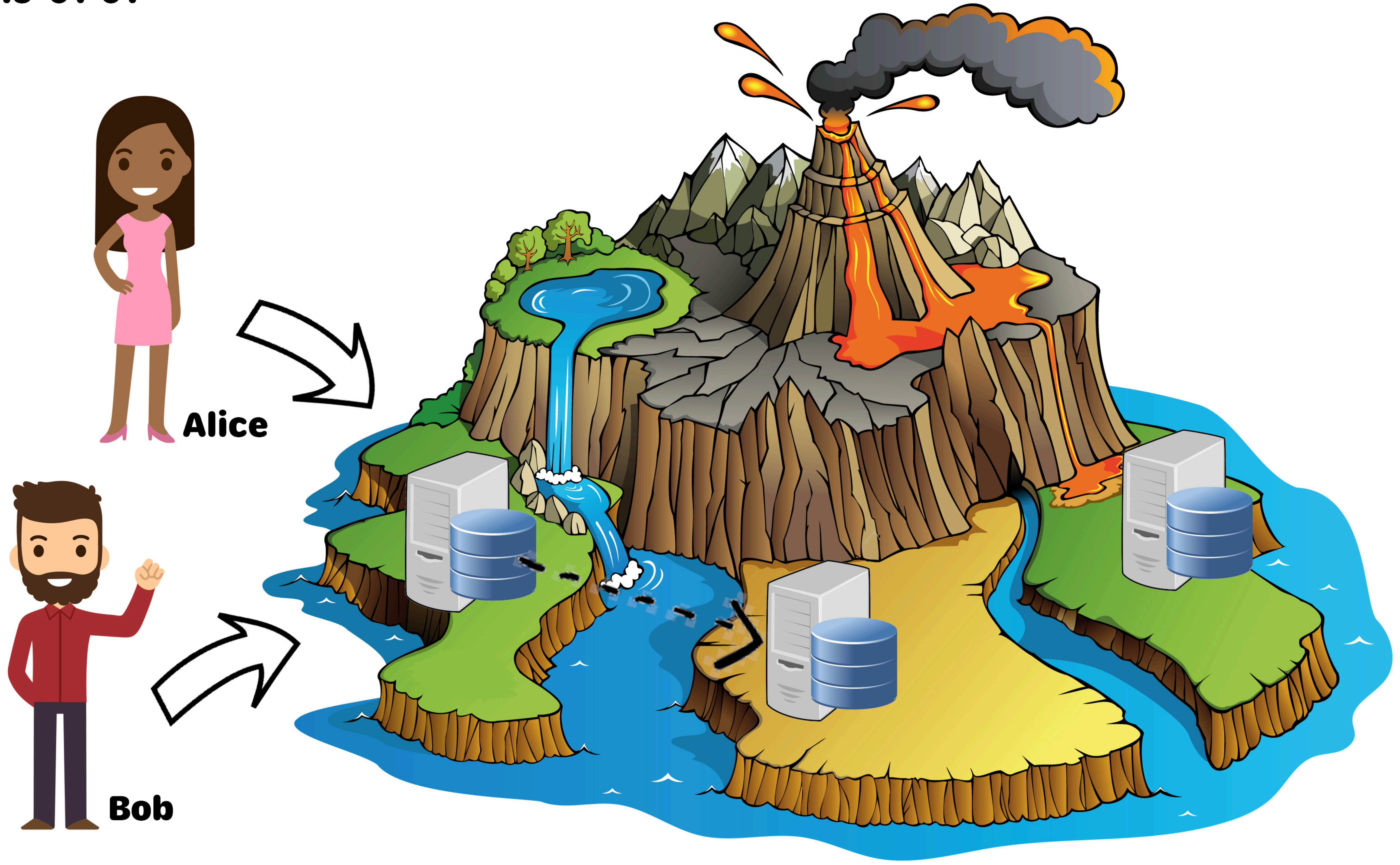
- Transaction Time
- Valid Time

2019-01-01

2019-01-01

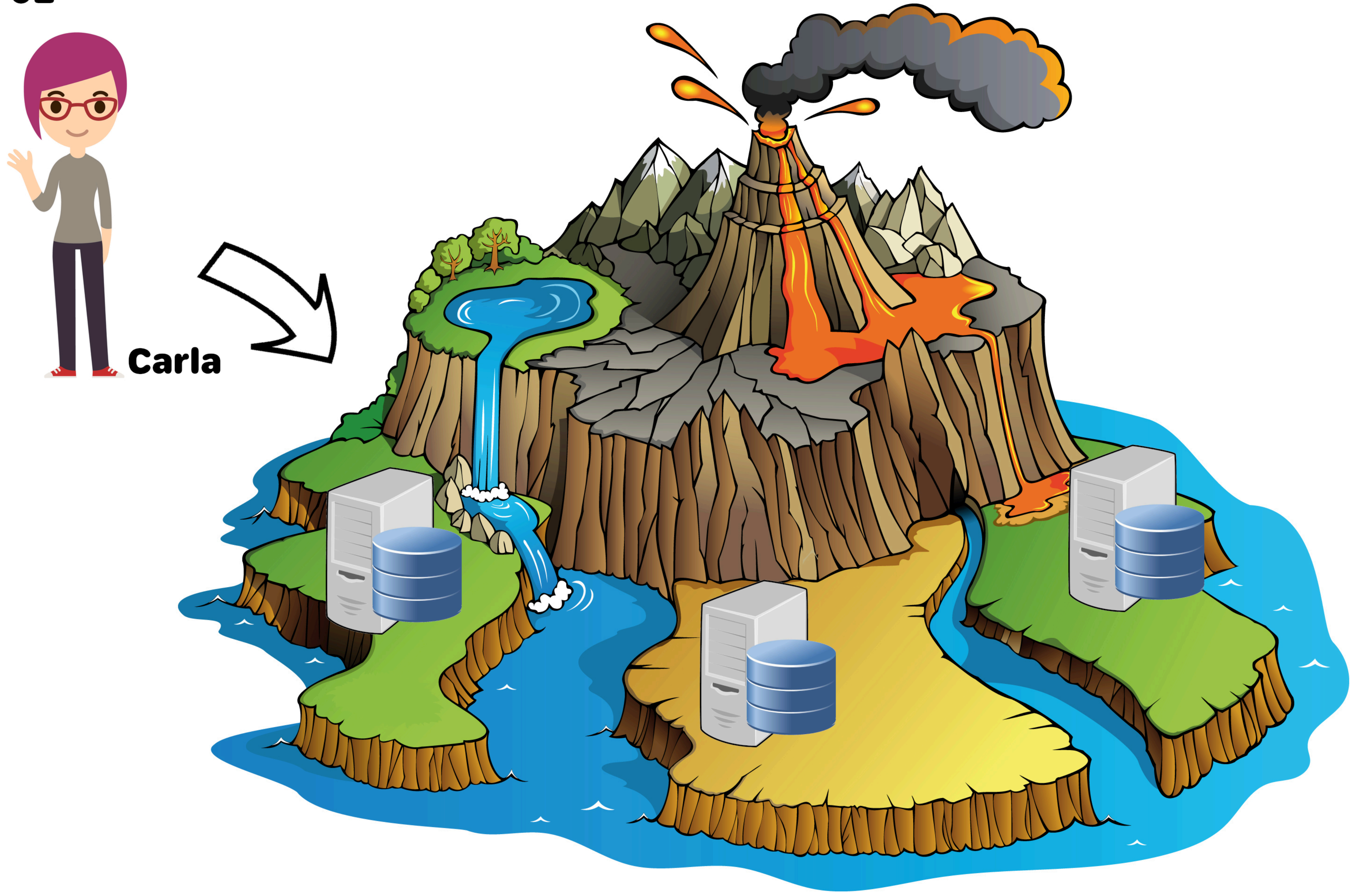


2019-01-01

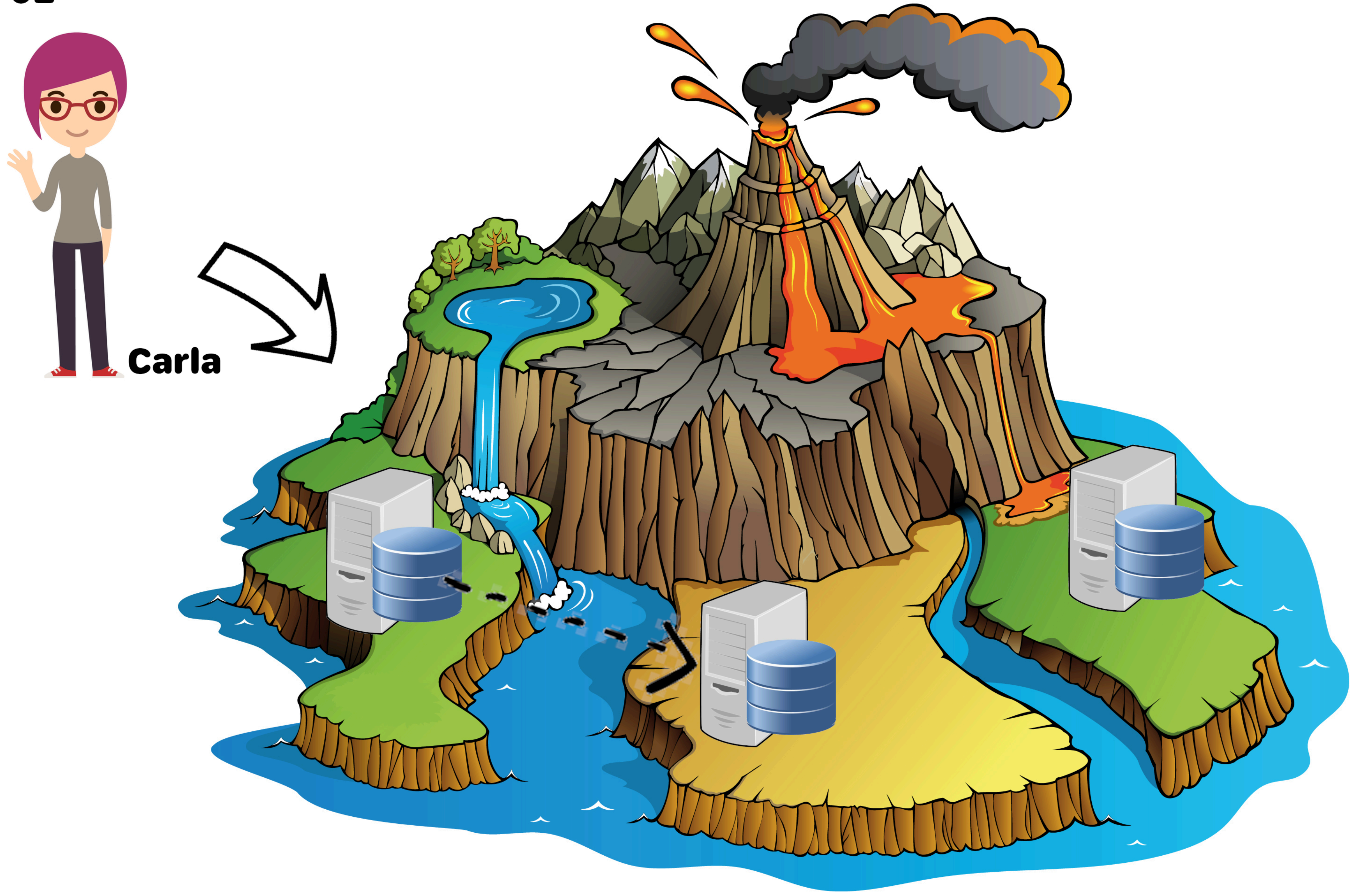


2019-01-02

2019-01-02

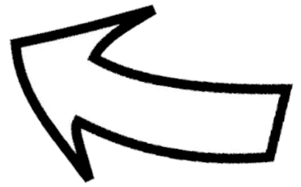


2019-01-02



2019-01-03

2019-01-03

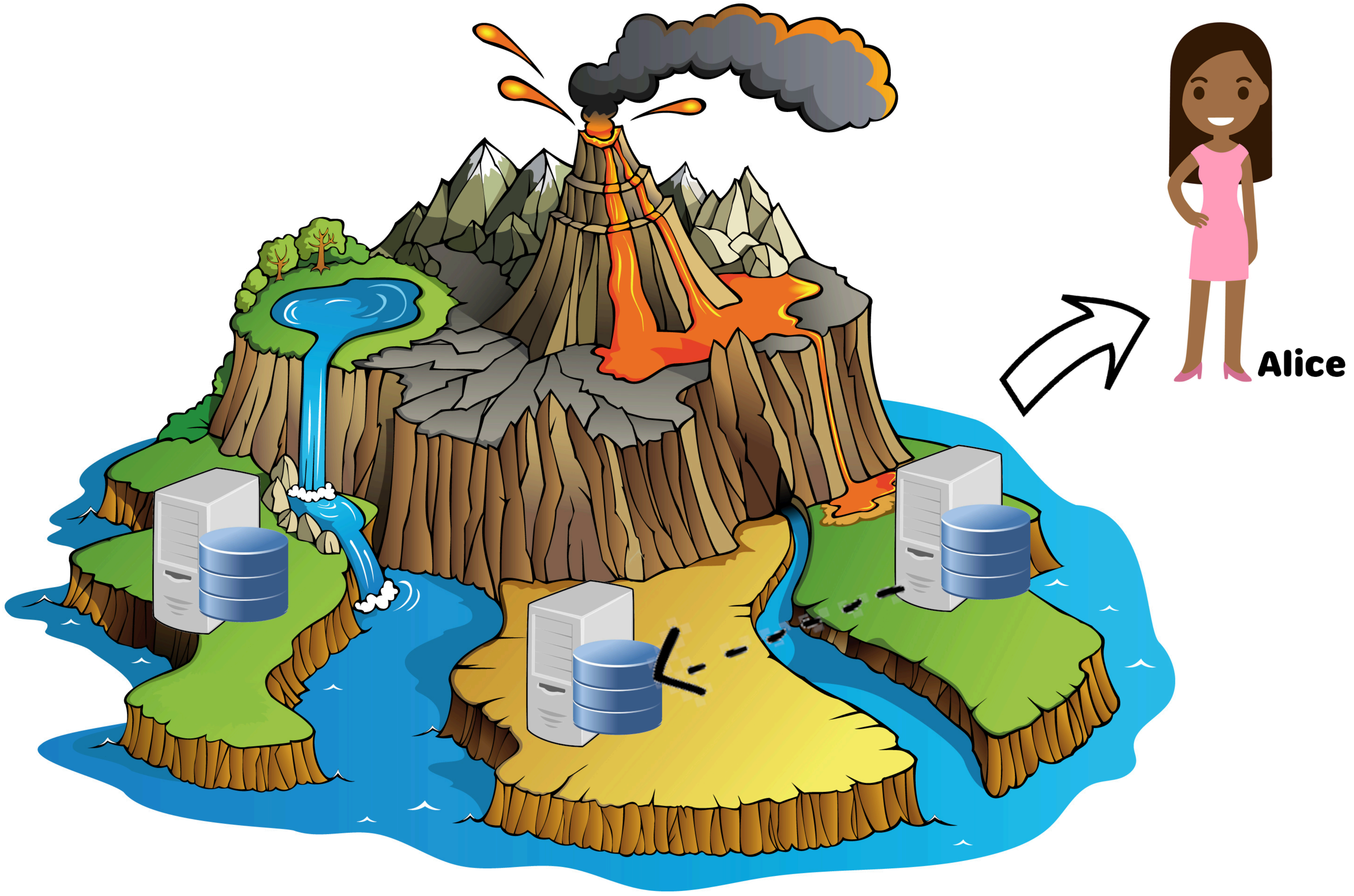


2019-01-03

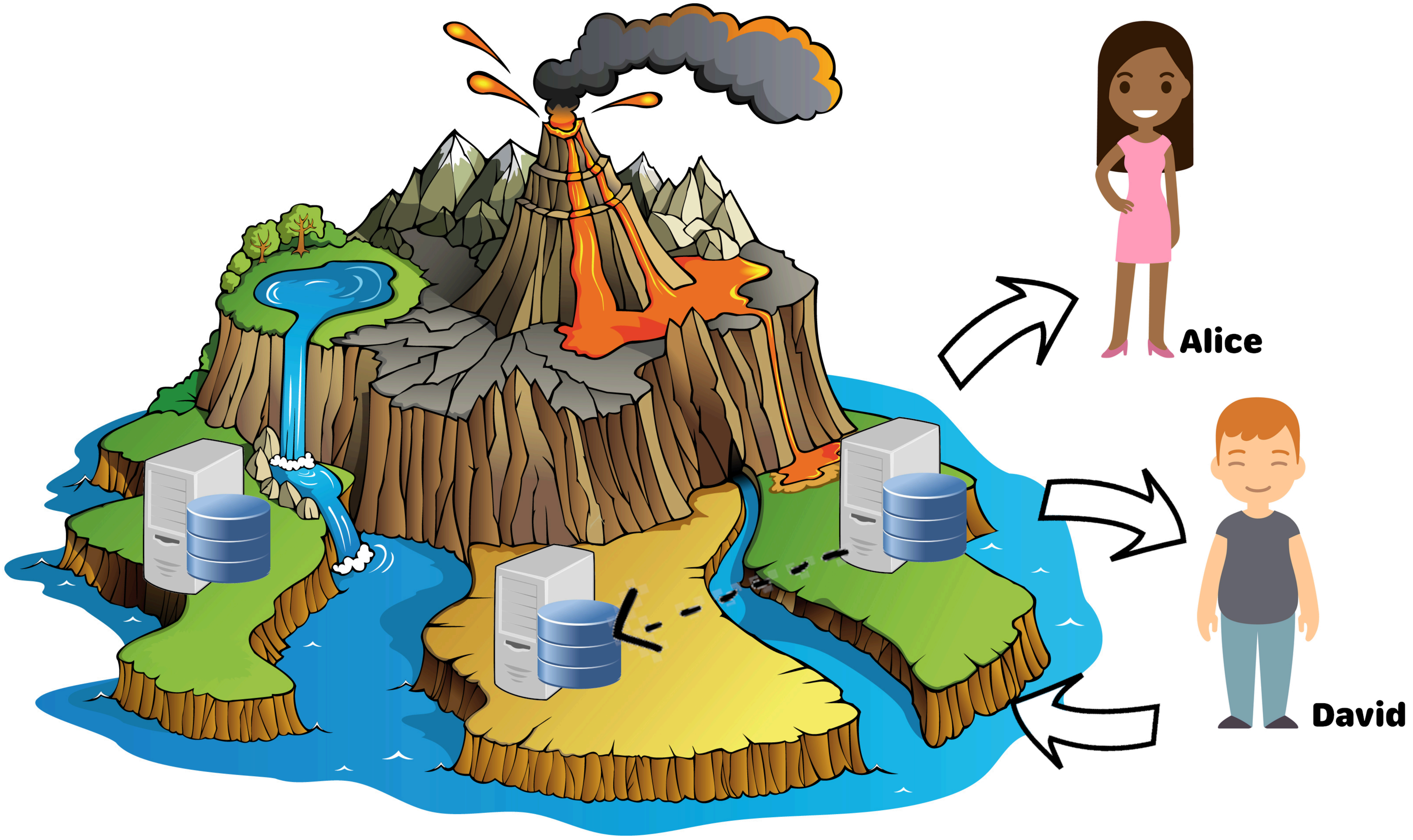
2019-01-04

2019-01-04

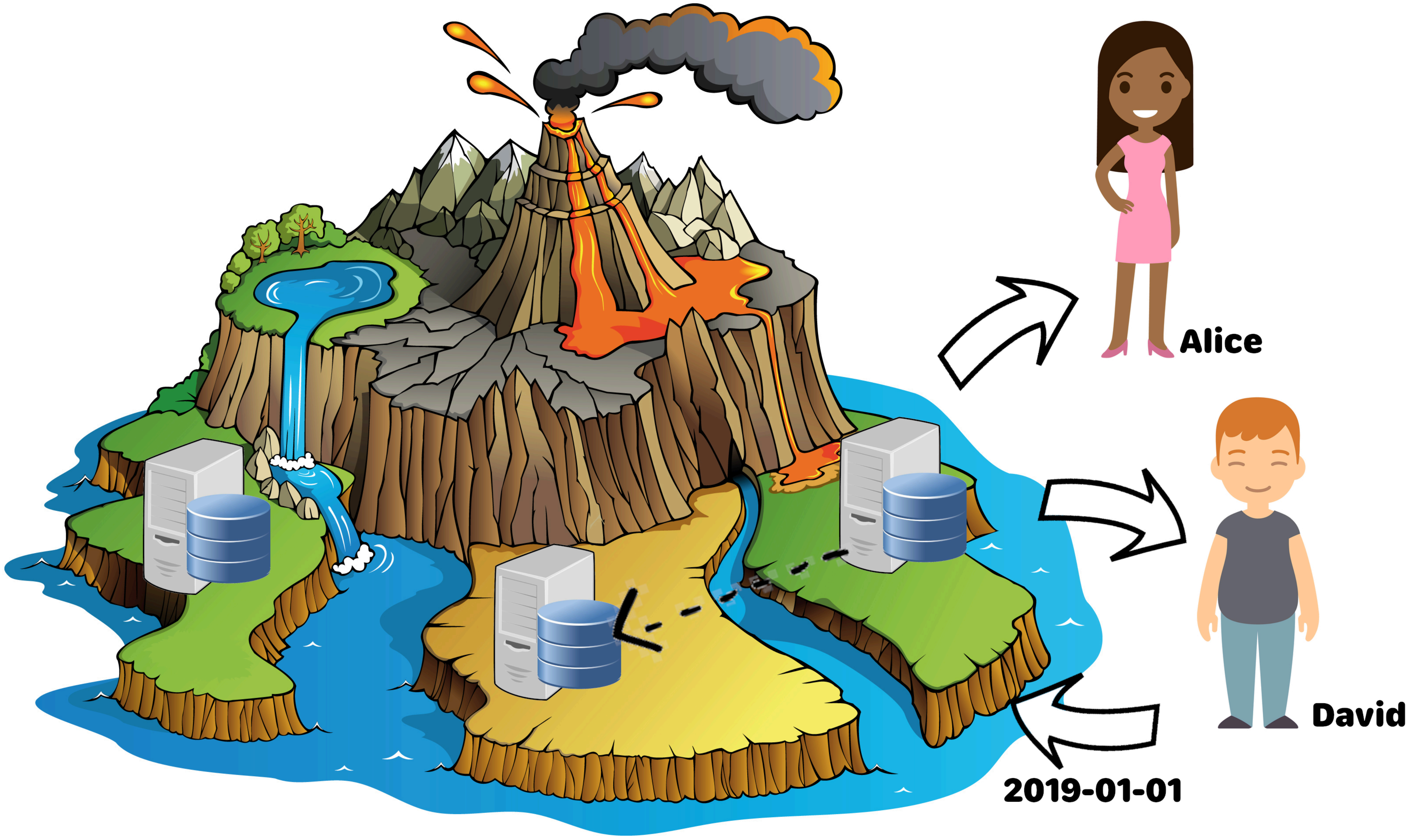
2019-01-04



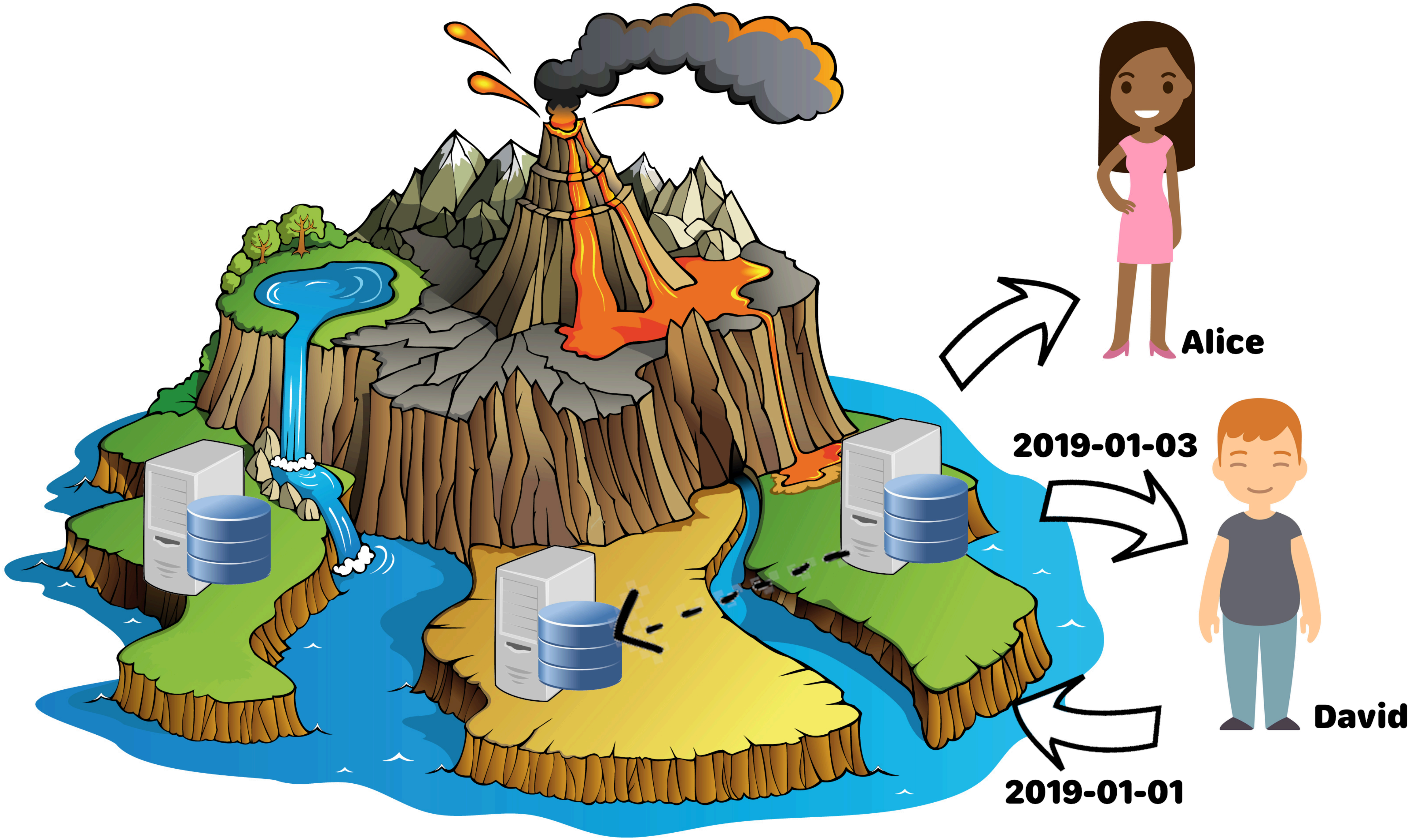
2019-01-04



2019-01-04



2019-01-04



WHO'S ON THE ISLAND ON THE 2ND?

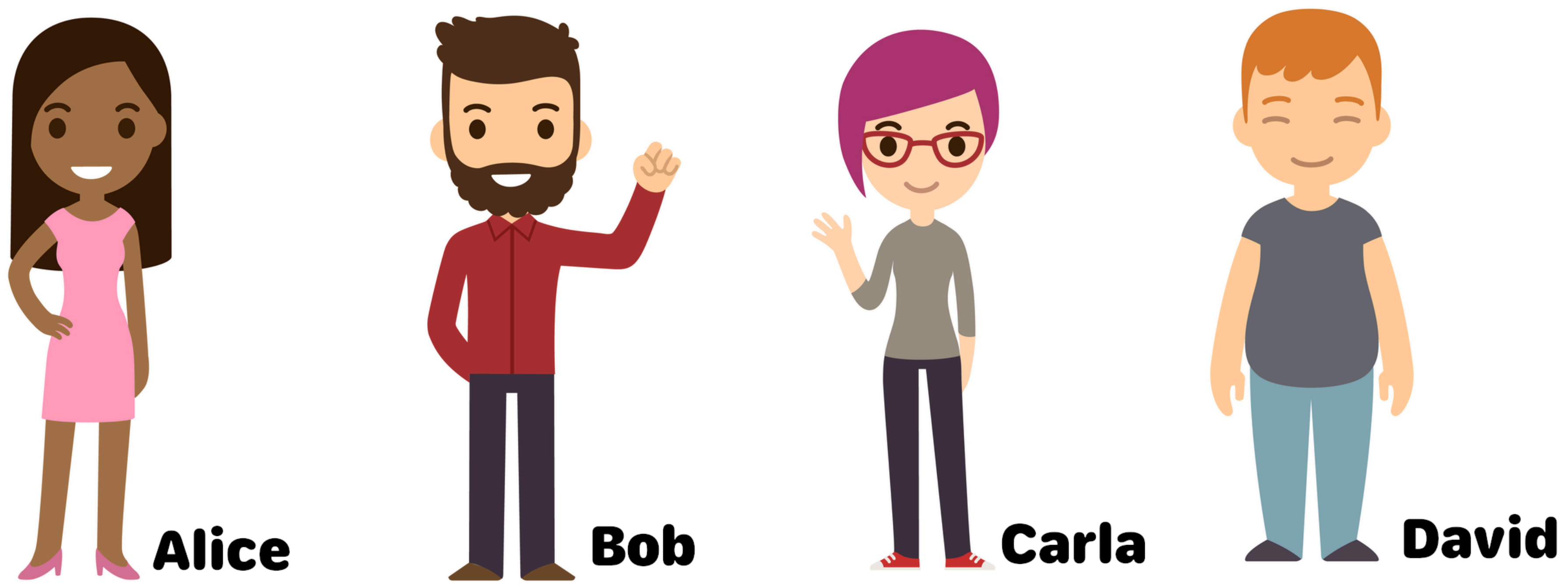
Who's on the island?

- "as of" - Transaction time
- "as at" - Valid time

WHO'S ON THE ISLAND ON THE 2ND? (AS OF THE 3RD)

Alice
Bob
Carla

WHO'S ON THE ISLAND ON THE 2ND? (AS OF THE 4TH)


Alice
Bob
Carla
David

Bitemporality (boss slide)

- Ingest out-of-order temporal data from upstream timestamping systems
- Reconcile temporal data across eventually consistent systems
- Dealing with backlog corrections

AUDIT ALL DATA CHANGES AND PERFORM DATA FORENSICS

3. STORAGE MODEL

3. Storage Model

- Crux: Document-based
- Datomic: Entity-Attribute-Value-based

Crux being document-based means

Crux being document-based means

1. Transactions: apply to the whole doc

Crux being document-based means

1. Transactions: apply to the whole doc
2. Whole document gets changed on transactions

Crux being document-based means

1. Transactions: apply to the whole doc
2. Whole document gets changed on transactions
3. Only root-level nodes are indexed

Crux being document-based means

1. Transactions: apply to the whole doc
2. Whole document gets changed on transactions
3. Only root-level nodes are indexed
4. Document relationships can be tricky

4. SCHEMA MODEL

4. Schema Model

- Crux: No Schema Definition/Enforcement
- Datomic: Required Schema Definition/Enforcement

Crux having no schema means

- No validation (anything goes)
  - Decoupled concerns
- Silver lining
  - No explicit schema evolution process needed
  - Good practice: implement your own validations

5. DEPLOYMENT MODEL

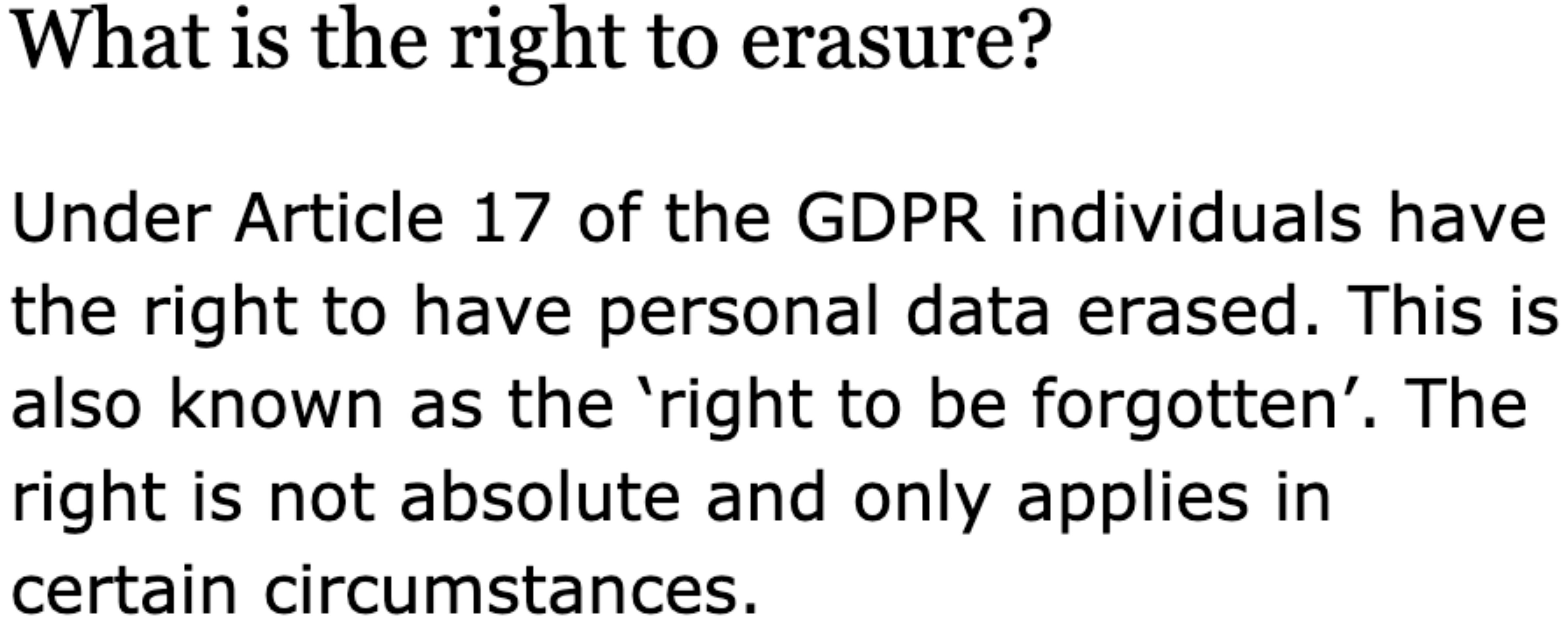
5. Deployment Model

- Crux: Self Hosted
  - Confluent Cloud (KaaS)
  - Managed hosting available (via Juxt)
- Datomic:
  - Datomic On-prem: Self Hosted
  - Datomic Cloud: Self Hosted (AWS Cloud-based)

Some Implications

- Both can be easily developed locally
  - run locally, self-host somewhere, or containerize it
- Crux locally can be a standalone (crux-core)
- Datomic Cloud requires an AWS connection for development

6. PRIVACY MODEL



What is the right to erasure?

Under Article 17 of the GDPR individuals have the right to have personal data erased. This is also known as the ‘right to be forgotten’. The right is not absolute and only applies in certain circumstances.

6. Privacy Model

- Crux: "Eviction" operation
- Datomic:
  - Datomic On-prem: "Excision" operation
  - Datomic Cloud: nothing available

7. QUERY DIFFERENCES

7. Query Differences

7. Query Differences

1. Queries are not portable across

7. Query Differences

1. Queries are not portable across
2. Crux does not support Pull Query

7. Query Differences

1. Queries are not portable across
2. Crux does not support Pull Query
3. Datomic does not have Lazy Queries

7. Query Differences

1. Queries are not portable across
2. Crux does not support Pull Query
3. Datomic does not have Lazy Queries
4. Crux does not have sophisticated look up refs

7. Query Differences

1. Queries are not portable across
2. Crux does not support Pull Query
3. Datomic does not have Lazy Queries
4. Crux does not have sophisticated look up refs
5. Datomic requires clause order

7. Query Differences

1. Queries are not portable across
2. Crux does not support Pull Query
3. Datomic does not have Lazy Queries
4. Crux does not have sophisticated look up refs
5. Datomic requires clause order
6. Several other minor differences

8. TRANSACTION DIFFERENCES

8. Transaction Differences

8. Transaction differences

1. Transactions are not portable across

8. Transaction Differences

1. Transactions are not portable across
2. Crux does not support Components

8. Transaction Differences

1. Transactions are not portable across
2. Crux does not support Components
3. Crux does not support Transaction meta model

8. Transaction Differences

1. Transactions are not portable across
2. Crux does not support Components
3. Crux does not support Transaction meta model
4. Crux does not support Transaction functions

8. Transaction differences

1. Transactions are not portable across
2. Crux does not support Components
3. Crux does not support Transaction meta model
4. Crux does not support Transaction functions
5. Datomic's injection speed is bound to transactor

8. Transaction Differences

1. Transactions are not portable across
2. Crux does not support Components
3. Crux does not support Transaction meta model
4. Crux does not support Transaction functions
5. Datomic's injection speed is bound to transactor
6. And remember: doc-based vs. attribute-based

9. OTHER ARCHITECTURAL DIFFERENCES

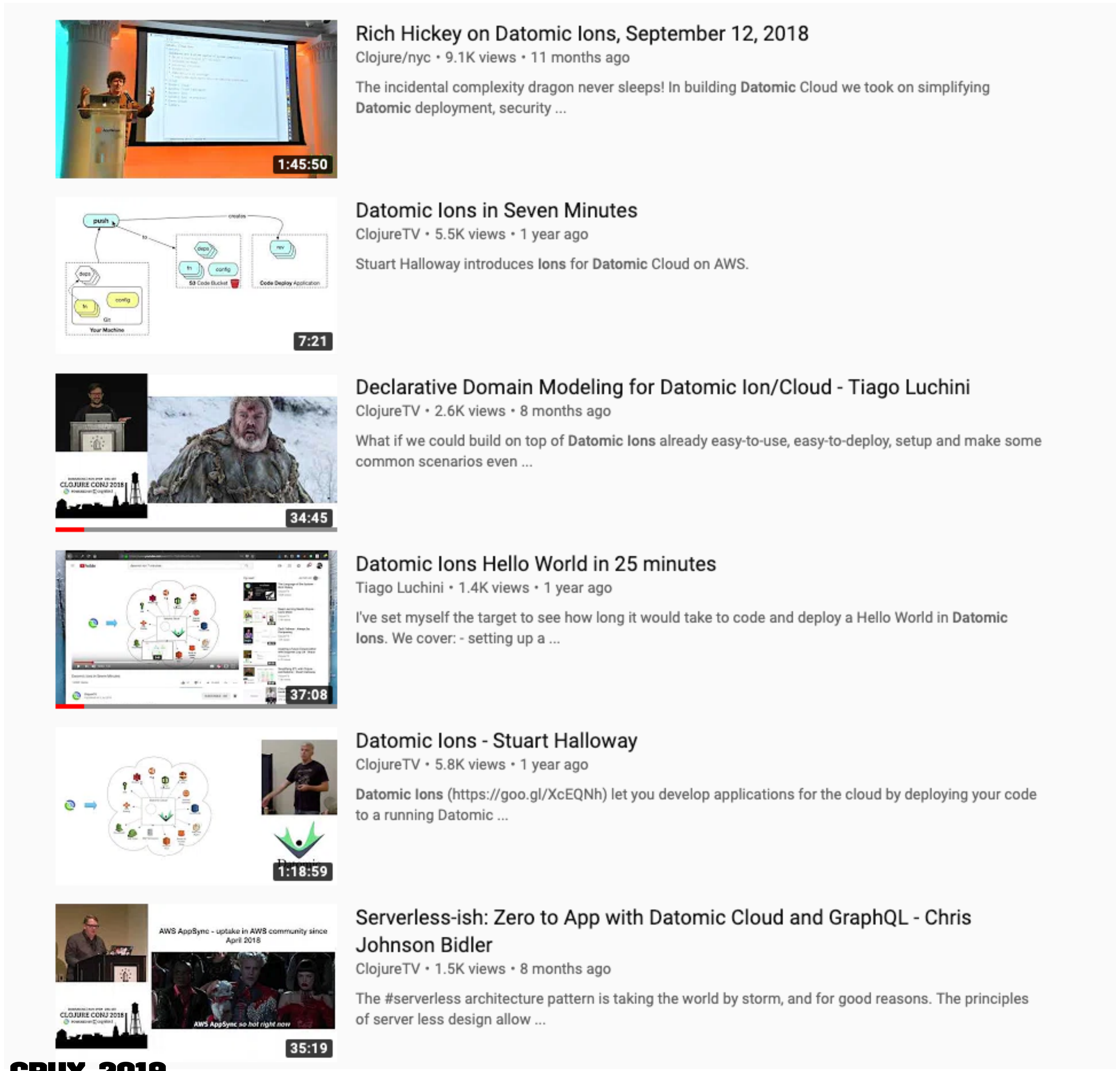
9. OTHER ARCHITECTURAL DIFFERENCES

	CRUX	DATOMIC
Topology	Each Node Storage/ Index	Global Shared Storage/Index
Index Sharding	No Auto Sharding	Peers auto-shard based on working sets
Transactor model	Single Unpartitioned Kafka Topic	Explicit Transactor

10. OTHER "FREEBIES"

10. Other "Freebies"

- Datomic
    - Datomic Cloud
    - Datomic Ions
    - Datomic Cast



Rich Hickey on Datomic Ions, September 12, 2018
Clojure/nyc • 9.1K views • 11 months ago
The incidental complexity dragon never sleeps! In building **Datomic** Cloud we took on simplifying **Datomic** deployment, security ...

Datomic Ions in Seven Minutes
ClojureTV • 5.5K views • 1 year ago
Stuart Halloway introduces **Ions** for **Datomic** Cloud on AWS.

Declarative Domain Modeling for Datomic Ion/Cloud - Tiago Luchini
ClojureTV • 2.6K views • 8 months ago
What if we could build on top of **Datomic Ions** already easy-to-use, easy-to-deploy, setup and make some common scenarios even ...

Datomic Ions Hello World in 25 minutes
Tiago Luchini • 1.4K views • 1 year ago
I've set myself the target to see how long it would take to code and deploy a Hello World in **Datomic Ions**. We cover: - setting up a ...

Datomic Ions - Stuart Halloway
ClojureTV • 5.8K views • 1 year ago
Datomic Ions (https://goo.gl/XcEQNh) let you develop applications for the cloud by deploying your code to a running Datomic ...

Serverless-ish: Zero to App with Datomic Cloud and GraphQL - Chris Johnson Bidler
ClojureTV • 1.5K views • 8 months ago
The #serverless architecture pattern is taking the world by storm, and for good reasons. The principles of server less design allow ...

10. Other "Freebies"

- Crux
  - Open Source
  - I.e. Active Object Backend
  - Coming up: Crux console

CLOJURE IS GREATLY SERVED ON THE DATABASE SIDE

With either Crux or Datomic:

1. Immutable Database
2. Query Like a Ninja
3. Unbundle Your Database

THANK YOU!

@tiagoluchini
https://luchini.nyc