Real World Crypto
January 11, 2018

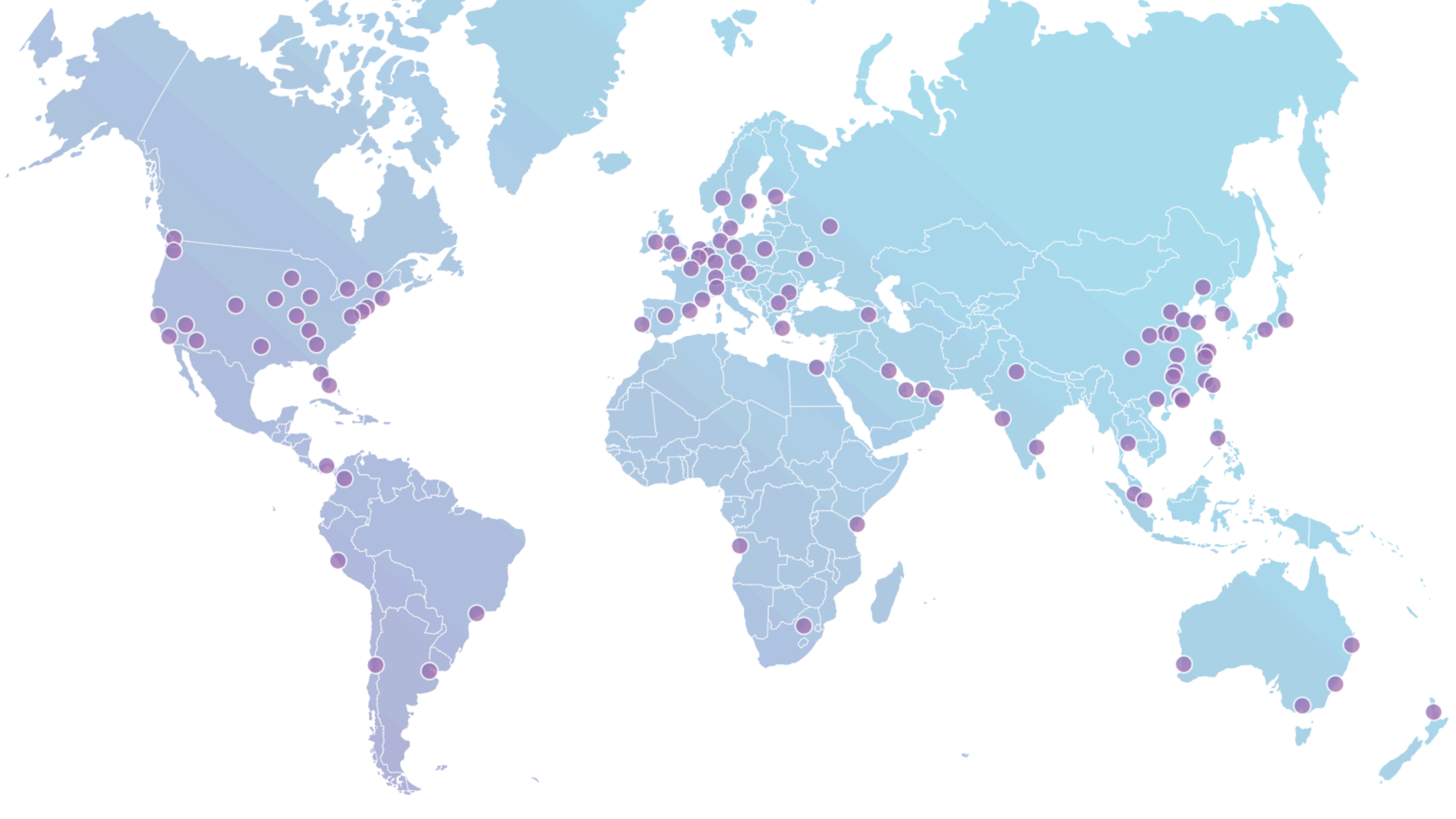
Geo Key Manager

Nick Sullivan (@grittygrease)

Brendan McMillion

Our Problem

Geographically-Distributed Key Management

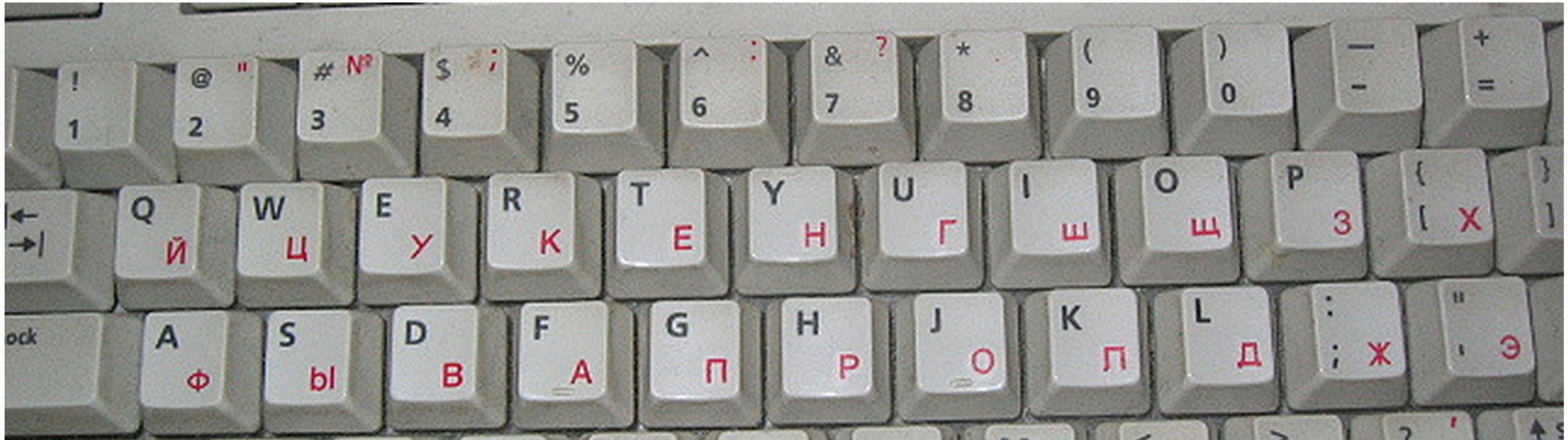


The Daily Dot

Russia begins collecting encryption keys while internet companies, like Facebook, stay silent

So far, WhatsApp, Viber, and Telegram haven't said a public word.

Patrick Howell O'Neill—July 27, 2016 at 4:27PM | Last updated July 27, 2016 at 9:29PM



Some things we have never done

- Cloudflare has never turned over our SSL keys or our customers' SSL keys to anyone.
- Cloudflare has never installed any law enforcement software or equipment anywhere on our network.
- Cloudflare has never terminated a customer or taken down content due to political pressure.
- Cloudflare has never provided any law enforcement organization a feed of our customers' content transiting our network.

If Cloudflare were asked to do any of the above, we would exhaust all legal remedies, in order to protect our customers from what we believe are illegal or unconstitutional requests.

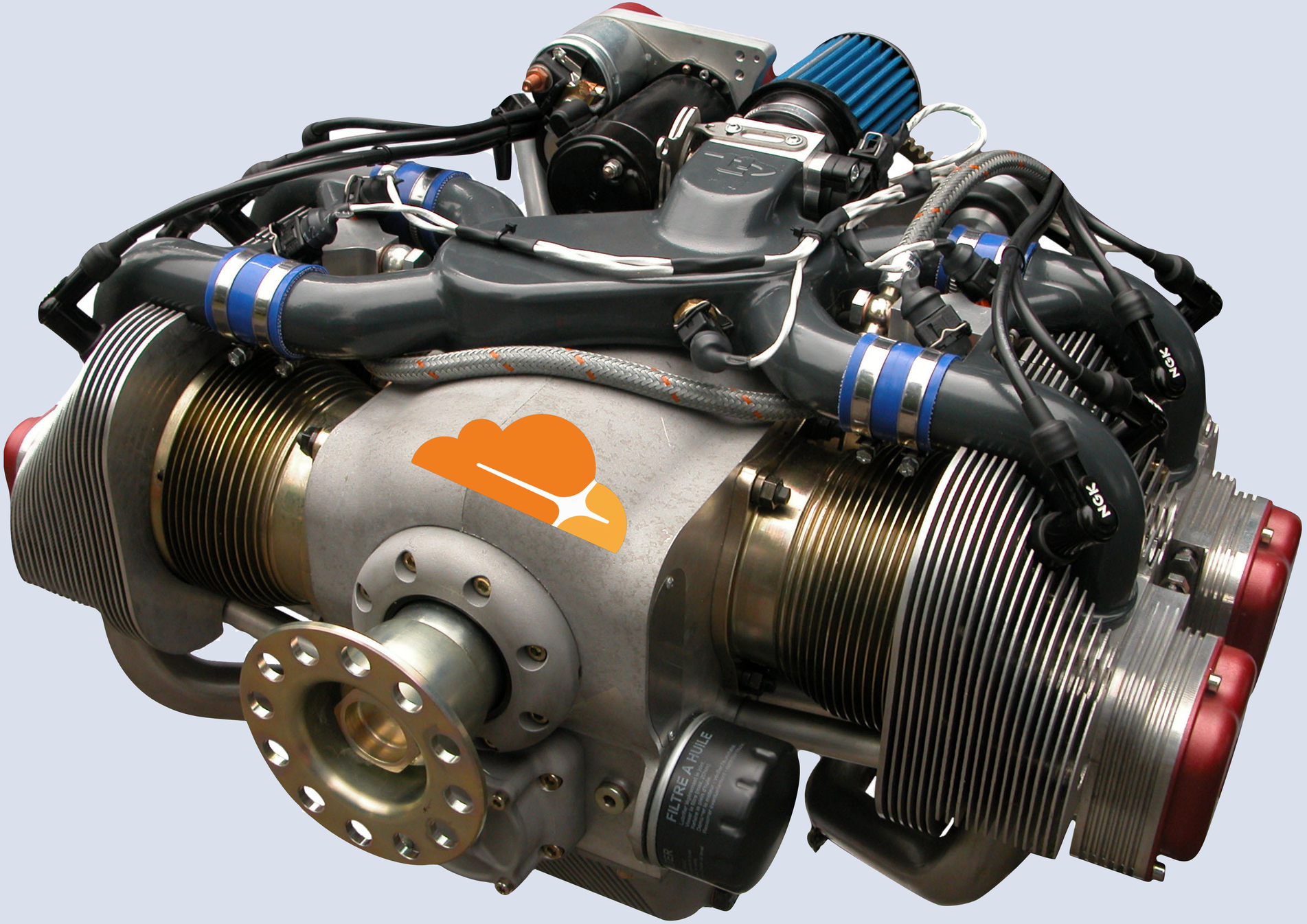
Customer's choice

Choose where in the world their keys are kept

Deployability

Work within existing constraints

Support network expansion


NGK
NGK
FILTRE A HUILE

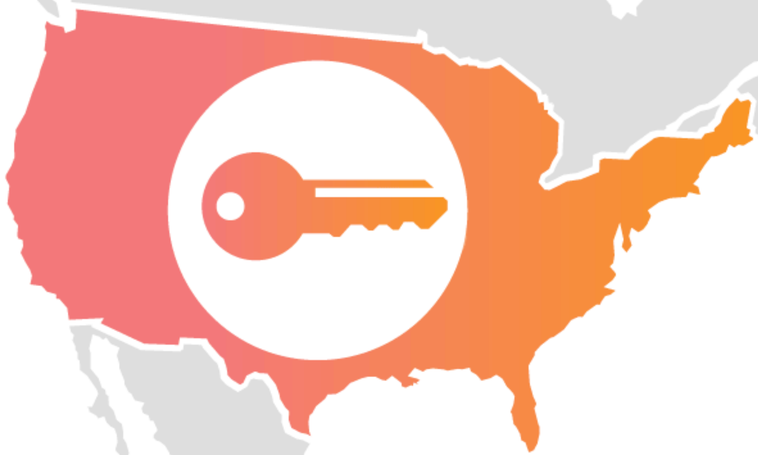
Constraint

Legacy client software

Component

Keyless SSL

Keyless SSL



Initial Request

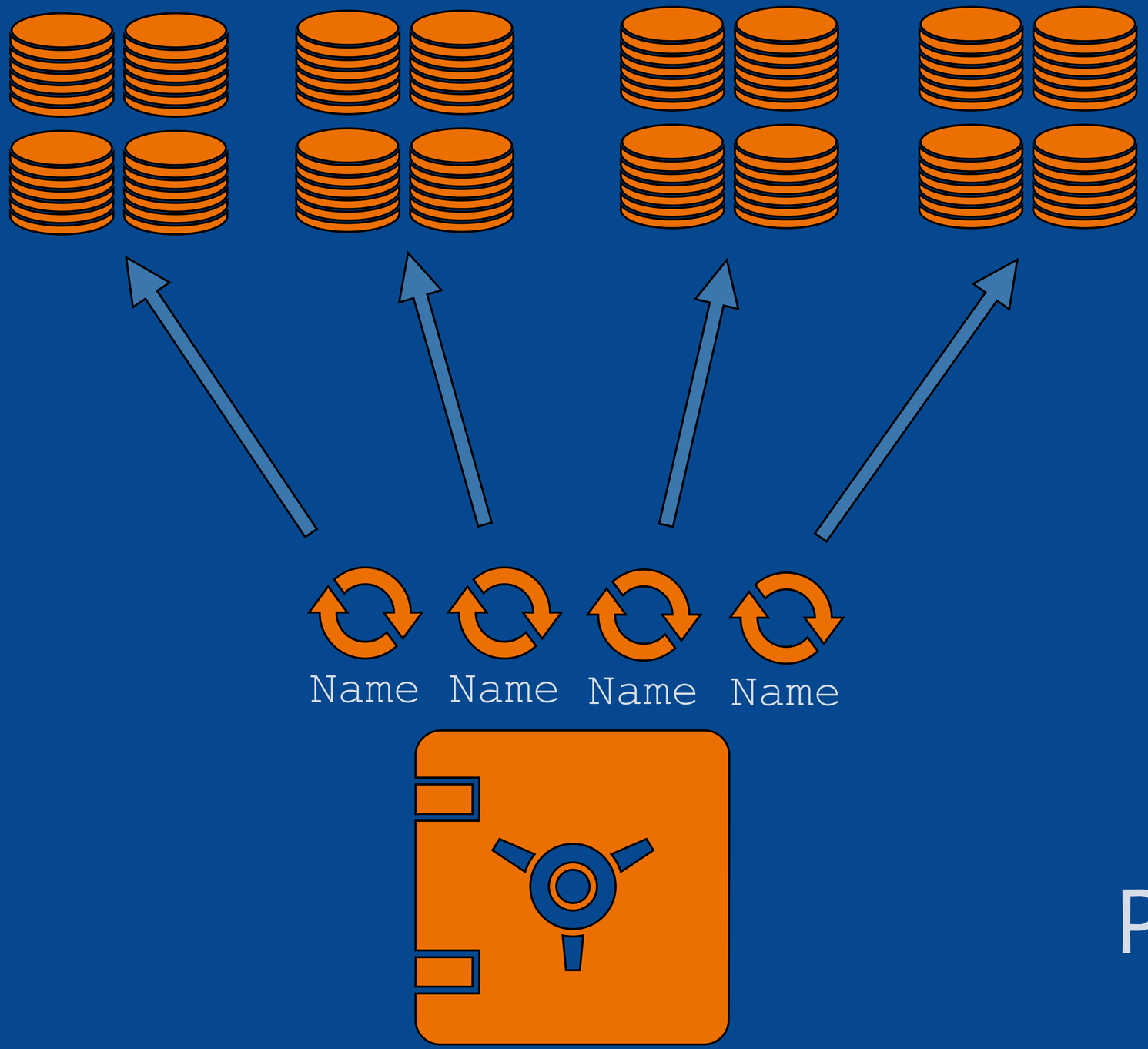
Latency Cost

Amsterdam to Dusseldorf	3ms
London to Moscow	50ms
Los Angeles to Belgrade	170ms
Brisbane to Muscat	500ms

Tool

Provisioning System

CLOUDFLARE



Edge Machines

Template

Provisioning Server

Component	Constraints
Provisioning System	Non-interactive, Identity-based

CLOUDFLARE

Component

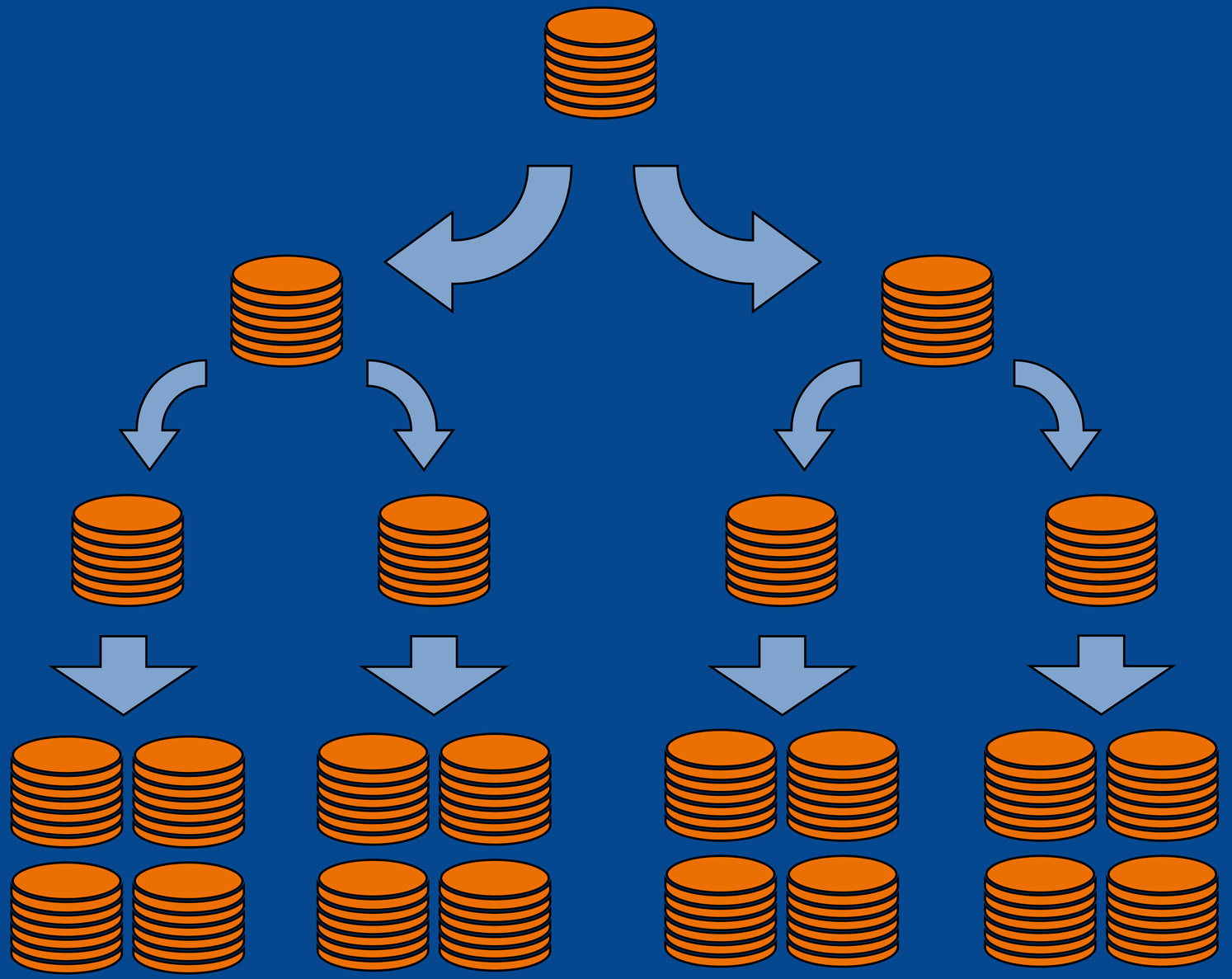
Globally Synchronized Database

Master Database

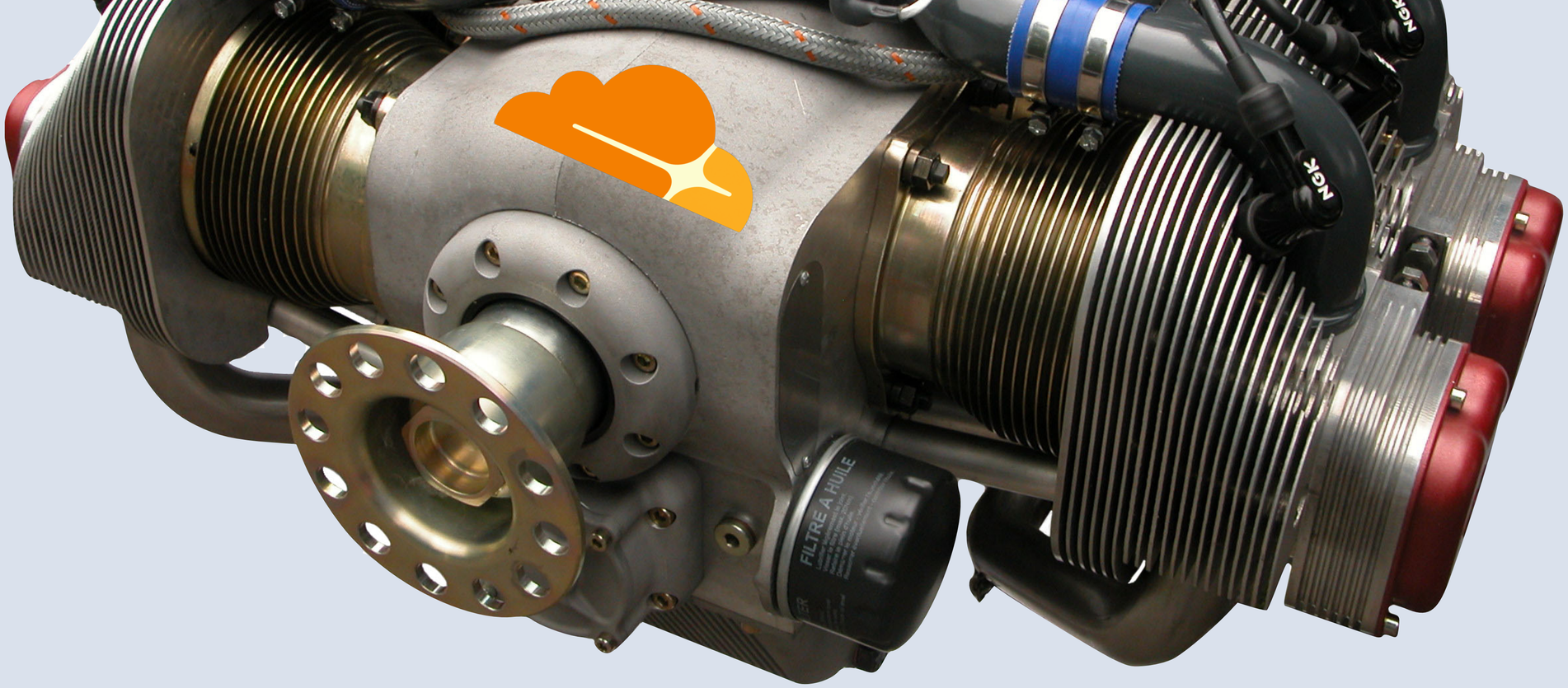
Regional Master

Location Master

Local Copy



Component	Constraints
Globally Synchronized Database	Bandwidth-limited, Broadcast

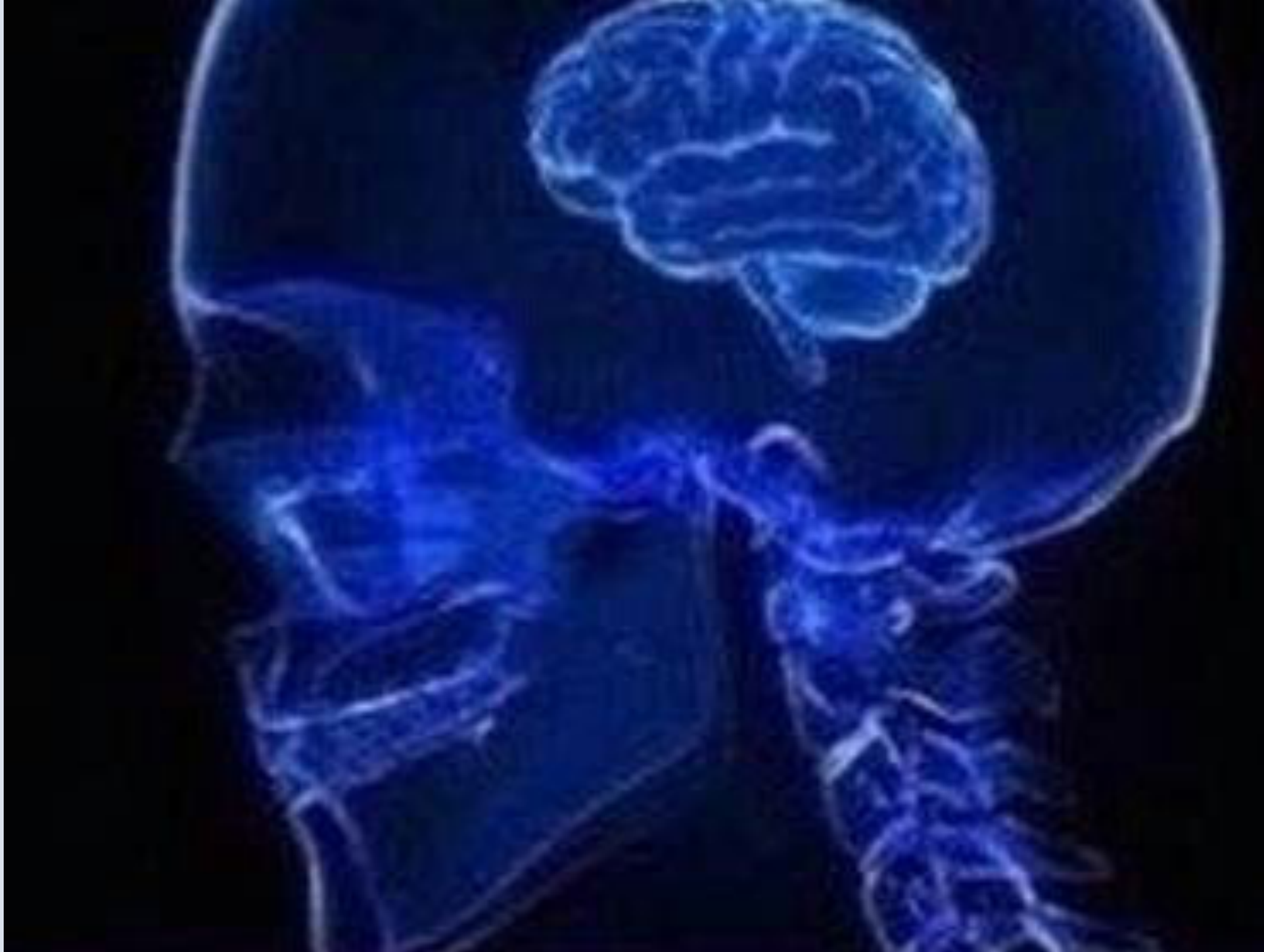


Identity-based provisioning system

Broadcast database of keys

High-latency fallback

Symmetric Cryptography

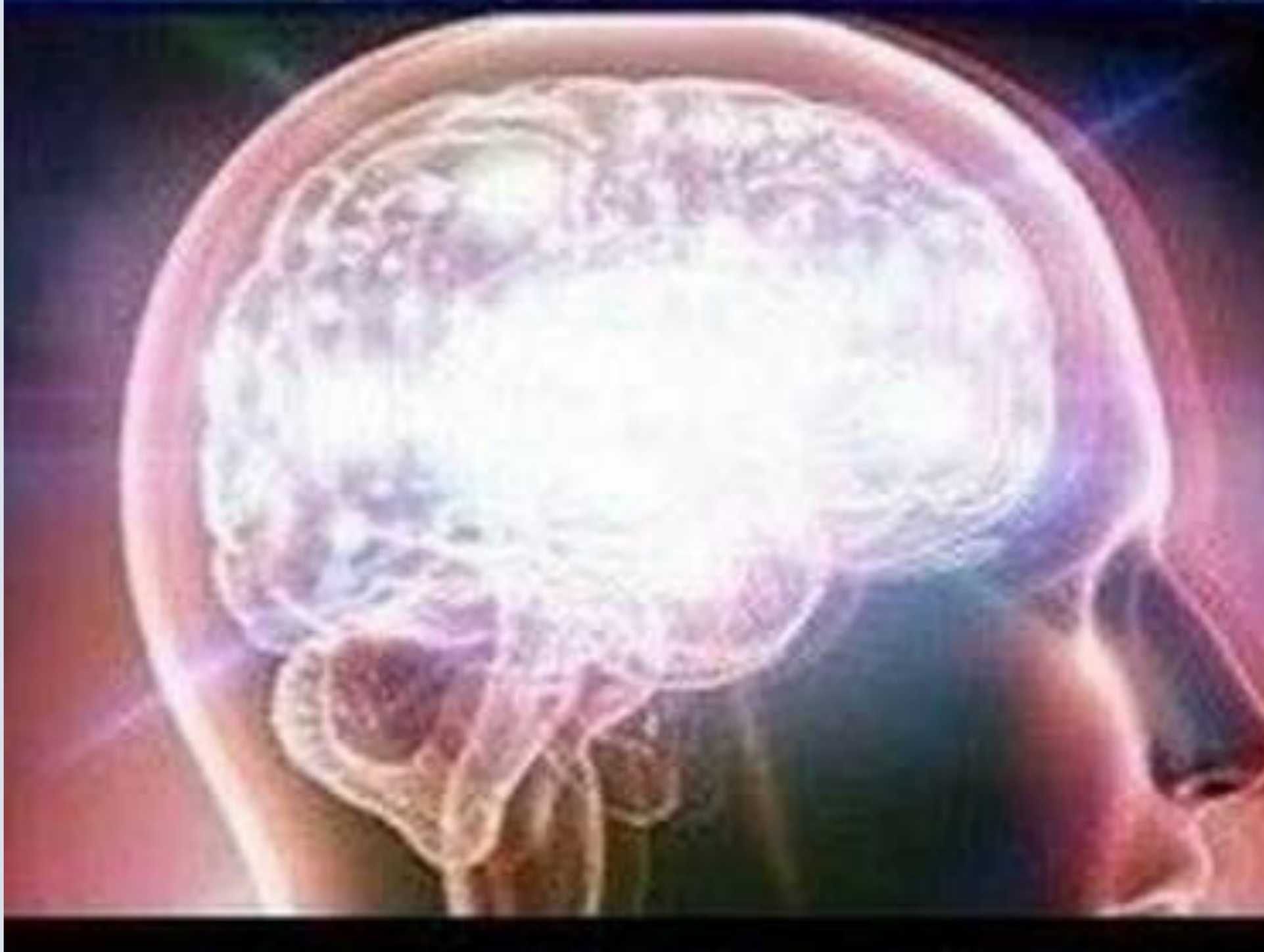


CLOUDFLARE

Asymmetric Cryptography

CLOUDFLARE

Pairing-based Cryptography



CLOUDFLARE

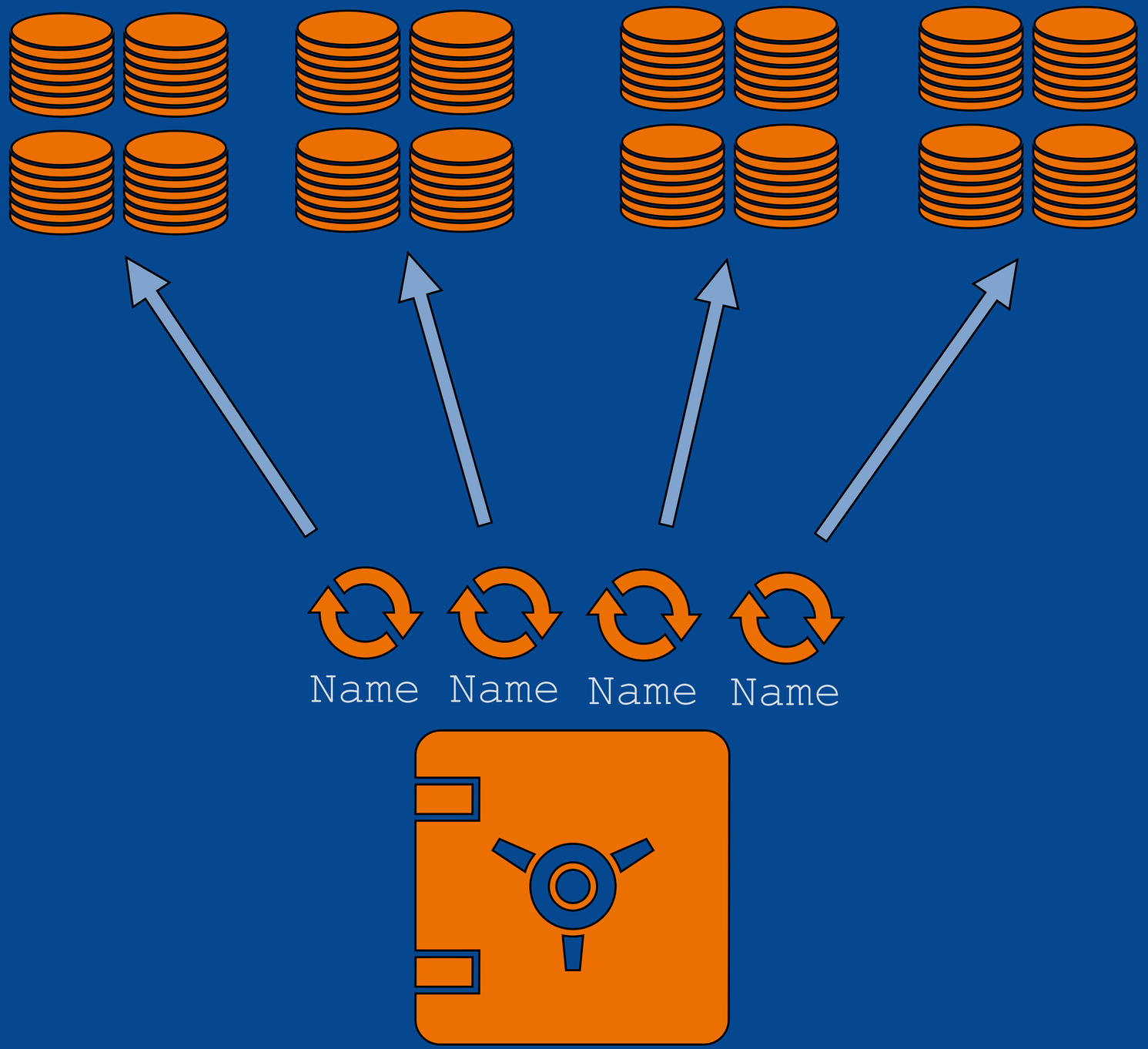
Fully Homomorphic Encryption

CLOUDFLARE

Identity-based encryption

- **Public Key**: used to encrypt data to any identity (like “machine2”)
- **Master Key**: provisions private keys to identities
- **Private Key**: decrypts ciphertext

- Allows encryption to identities even if they don’t have a key yet



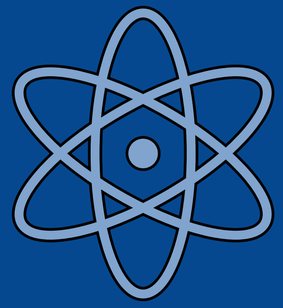
Participants

Private Keys

Extract

Master Key

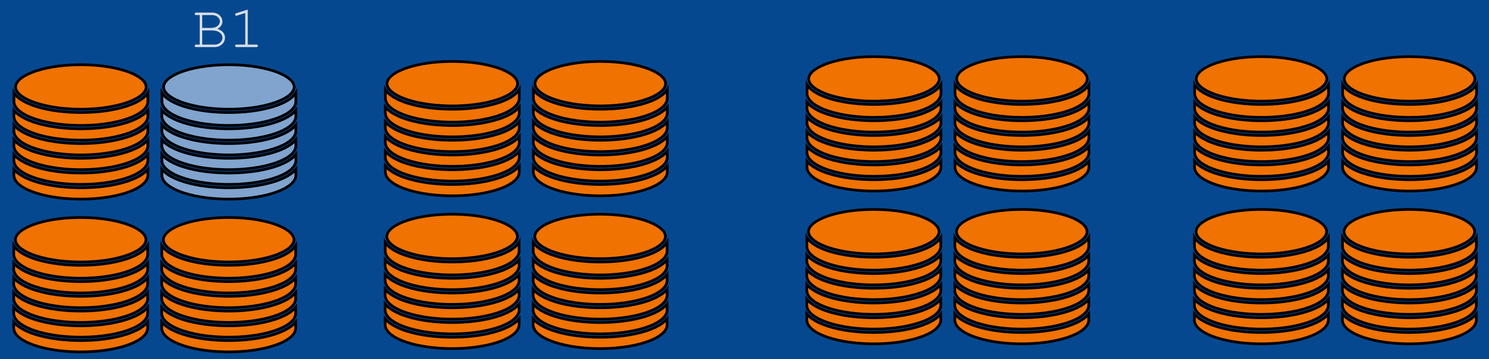
Public Key



Encrypt

Ciphertext

Decrypt



Bilinear Pairings

$$e: G_1 \times G_2 \longrightarrow G_T$$

$$e(P + Q, R) = e(P, R) \cdot e(Q, R)$$

$$e(P, Q + R) = e(P, Q) \cdot e(P, R)$$

First functional IBE by Boneh & Franklin (2001)

Identity-based *broadcast* encryption

- **Public Key**: used to encrypt data to *any number of identities up to k*
- **Master Key**: provisions private keys
- **Private Key**: decrypts ciphertext

Identity-based *revocation*

- **Public Key**: used to encrypt data to *all identities except for k*
- **Master Key**: provisions private keys
- **Private Key**: decrypts ciphertext

IBBE and IBR with short ciphertexts

Delerableé (2007)

- Master Key: constant
- Public Key: linear in k
- Private Key: constant
- Ciphertext: constant

Attrapadung, Libert, de Panafieu (2010)

- Master Key: constant
- Public Key: linear in k
- Private Key: linear in k
- Ciphertext: constant

Barreto-Naehrig Curves

$$e: E(F_p) \times E'(F_{p2}) \longrightarrow F_{p12}$$

BN256

128-bit security level*

implementation in Go by Adam Langley

10x speedup by Brendan McMillion on x86_64

faster than network round-trip from Zürich to Geneva

Cloudflare IBBE and IBR with BN256

Identity (IBBE)

- Master Key: **226B**
- Public Key: **k64B + 578B**
- Private Key: **k64B + 64B**
- Ciphertext: **192B** (batching)

Broadcast (IBR)

- Master Key: **64B**
- Public Key: **k64B + 384B**
- Private Key: **k64 + 192B**
- Ciphertext: **192B**

Simplified Geo Key Manager

1. Each location is provisioned a private key with its name
2. Customer: “I want my TLS key in Zürich and New York”
3. Encrypt TLS key to the name of those locations
4. Distribute encrypted key + “available in Zürich or New York”
5. When a connection comes in
    a. Decrypt key with location’s private key, or
    b. Connect to Zürich or New York with Keyless SSL

Desired Semantics

- **Whitelist**
- Put keys in multiple chosen locations
- Option to put keys in “new” locations based on region
- **Blacklist**
- Put keys in region, but exempt specific location

Key Encapsulation

Encrypt TLS key with a Key Encryption Key (KEK)

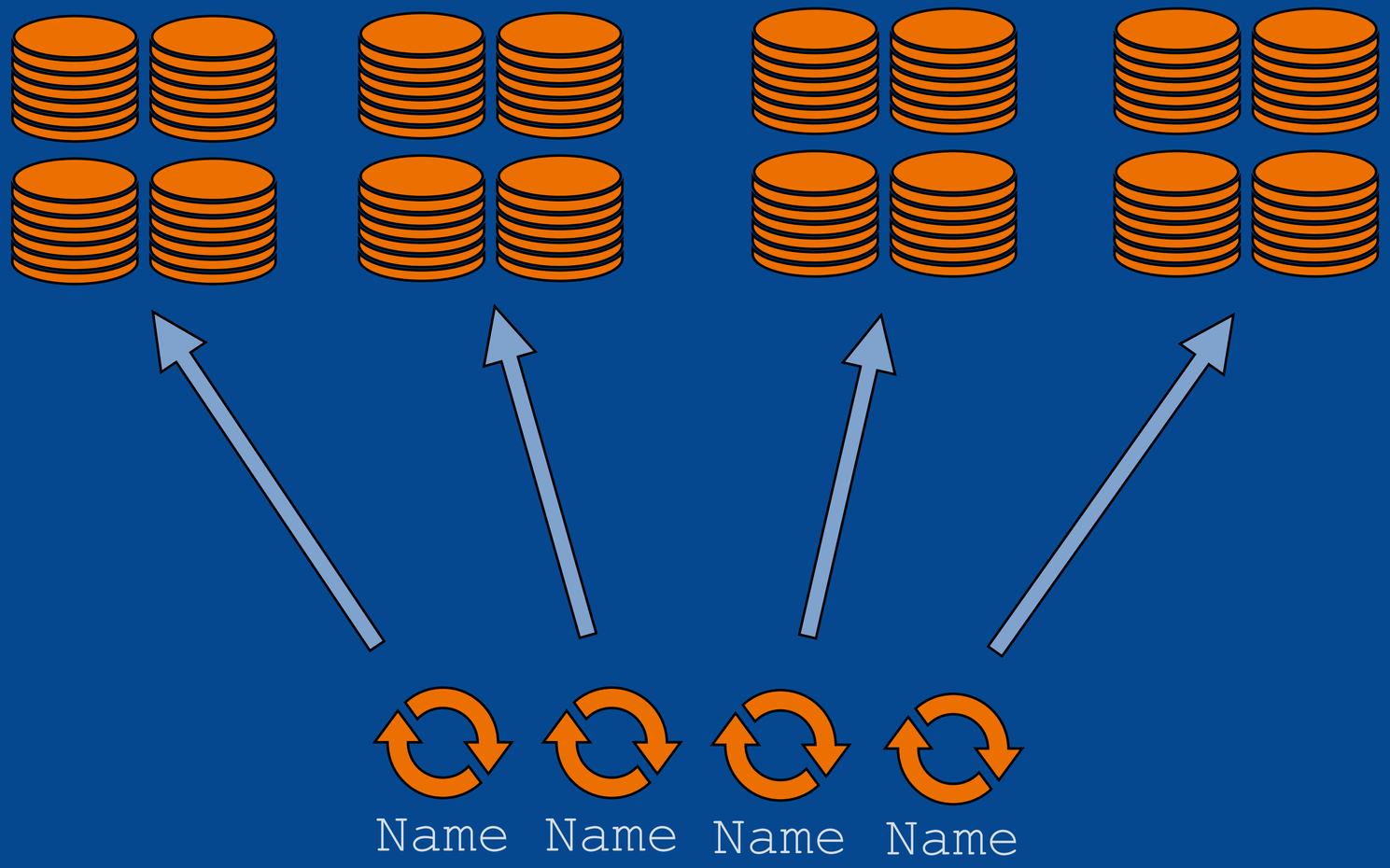
Split KEK in two

(e.g. KEK = KEK1 ⊕ KEK2)

KEM(kek1) for regions

KEM(kek2) for blacklisted locations

KEM(kek) for whitelisted locations



Edge Machines

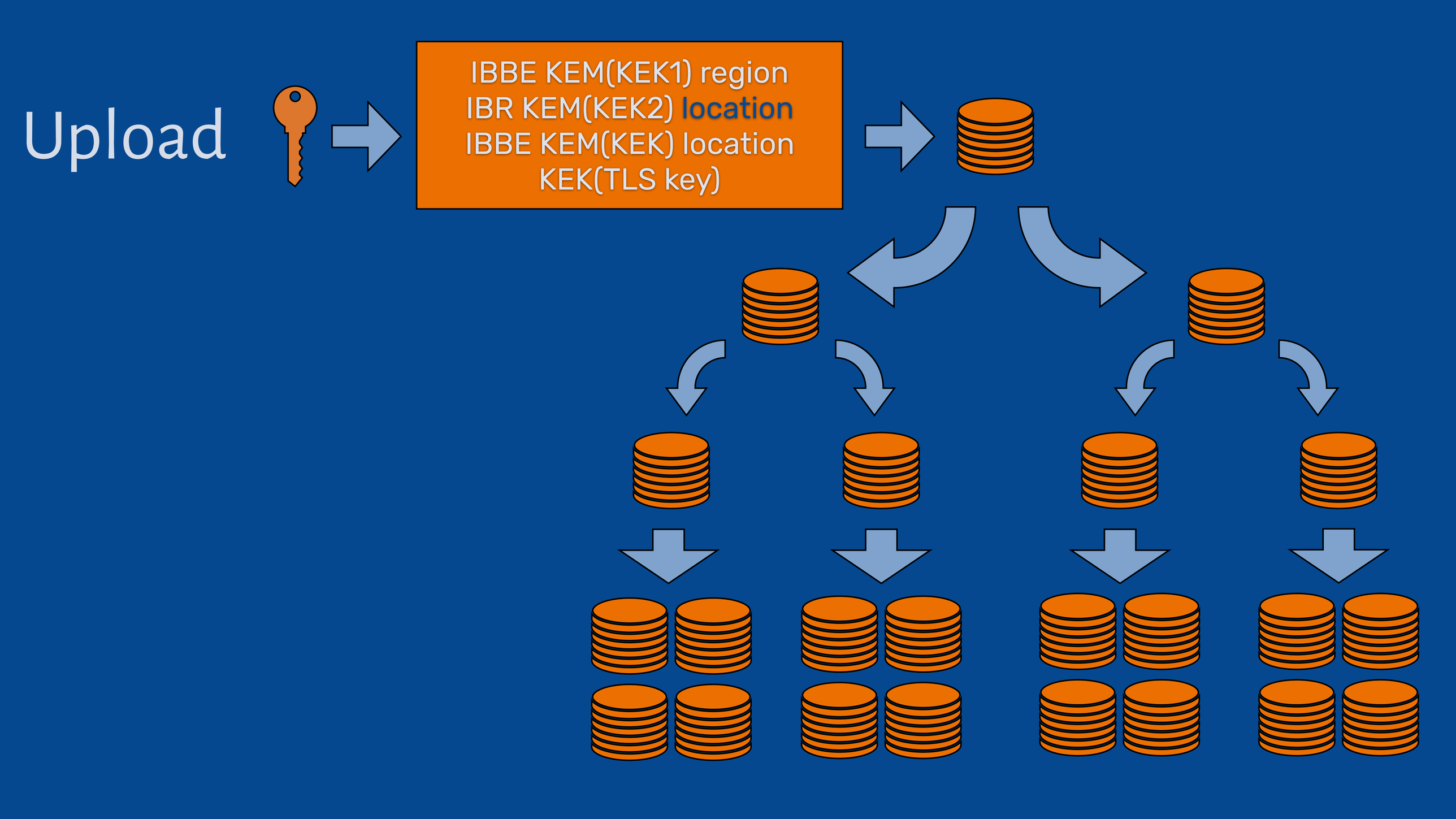
Private Keys

Extract

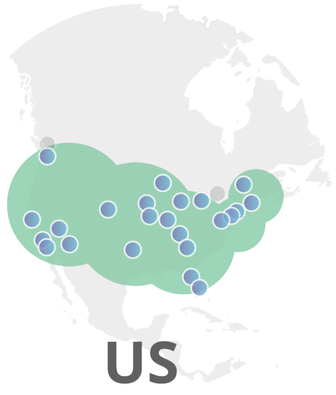
Master Key

Provisioning Server

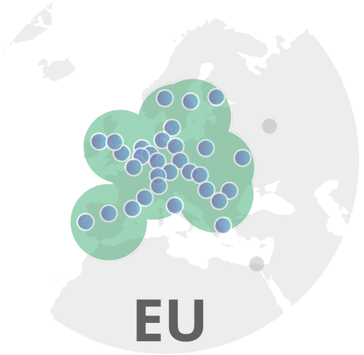
Upload


IBBE KEM(KEK1) region
IBR KEM(KEK2) location
IBBE KEM(KEK) location
KEK(TLS key)

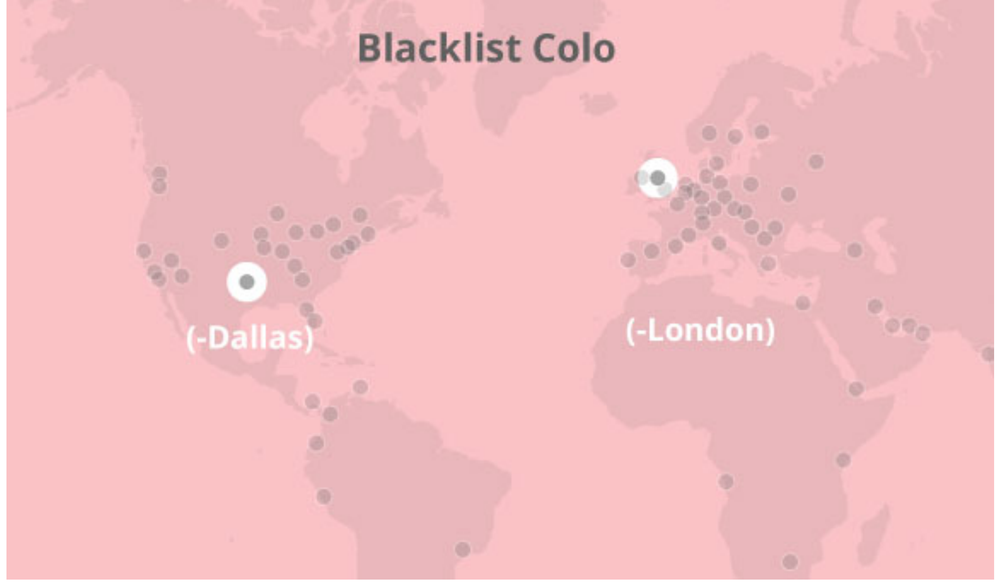
Whitelist Regions



US

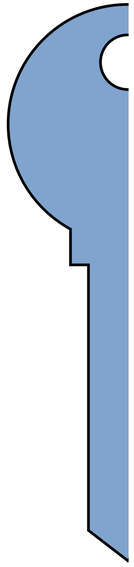


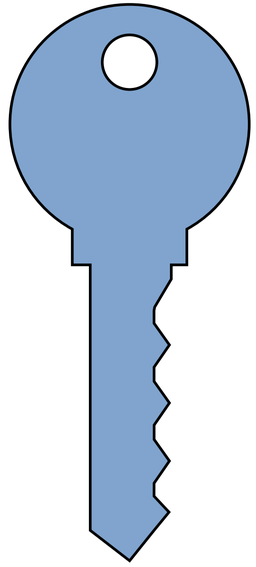
EU

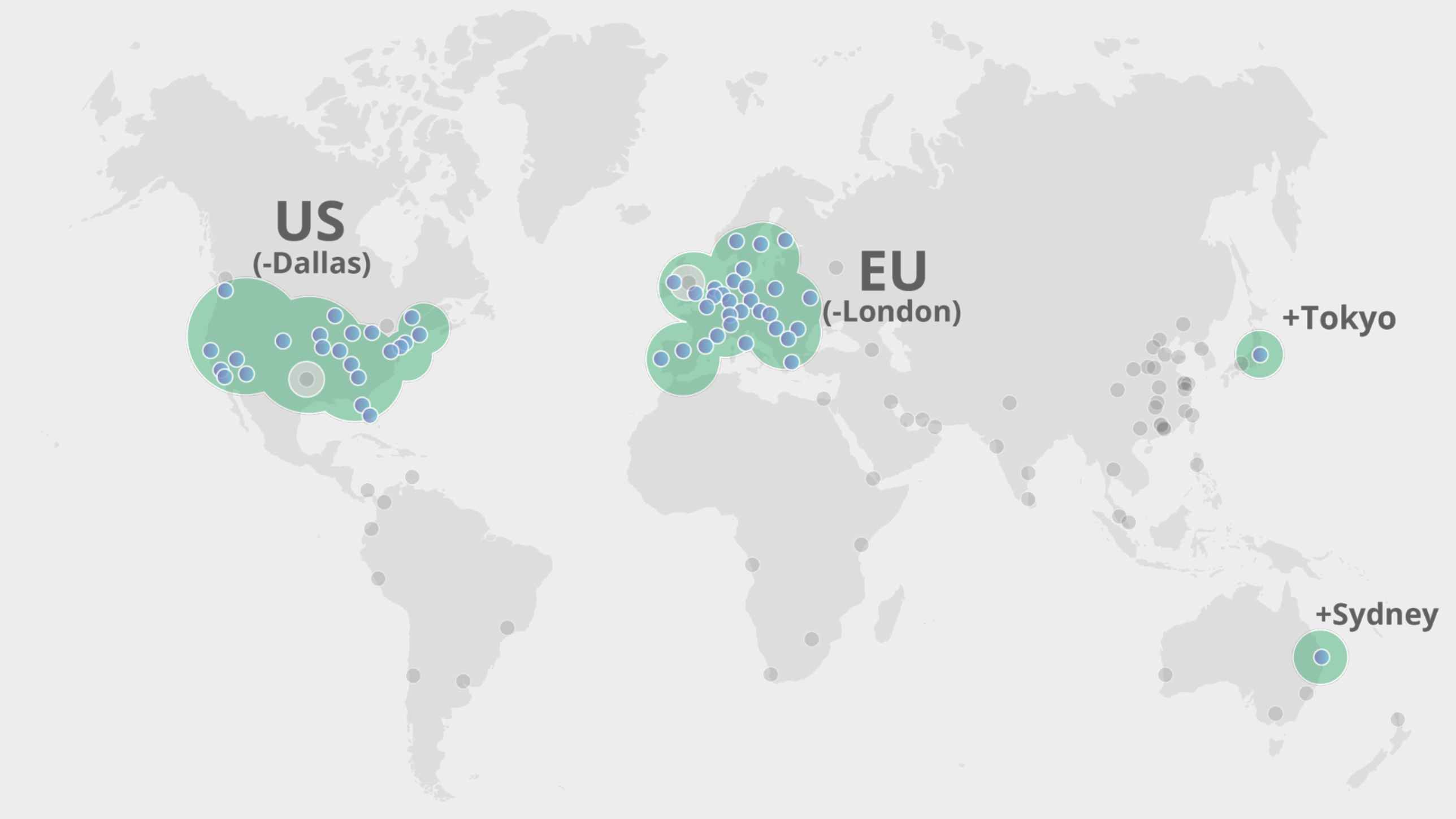


Whitelist Colo

+Sydney +Tokyo






US
(-Dallas)
EU
(-London)
+Tokyo
+Sydney

Geographically Distributed Key Management

With cryptographically-enforced access control

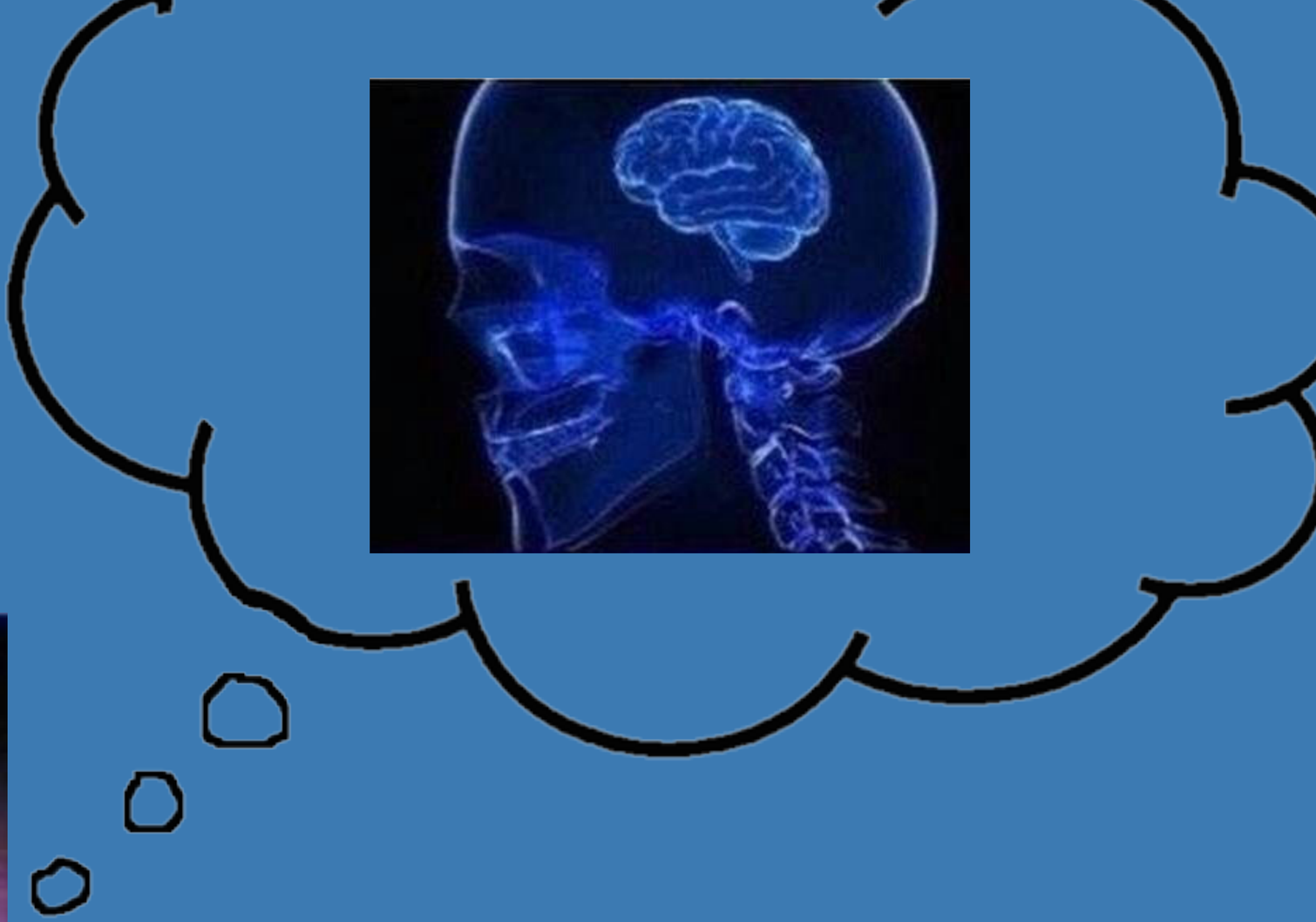
Real World Crypto
January 11, 2018

Geo Key Manager

Nick Sullivan (@grittygrease)

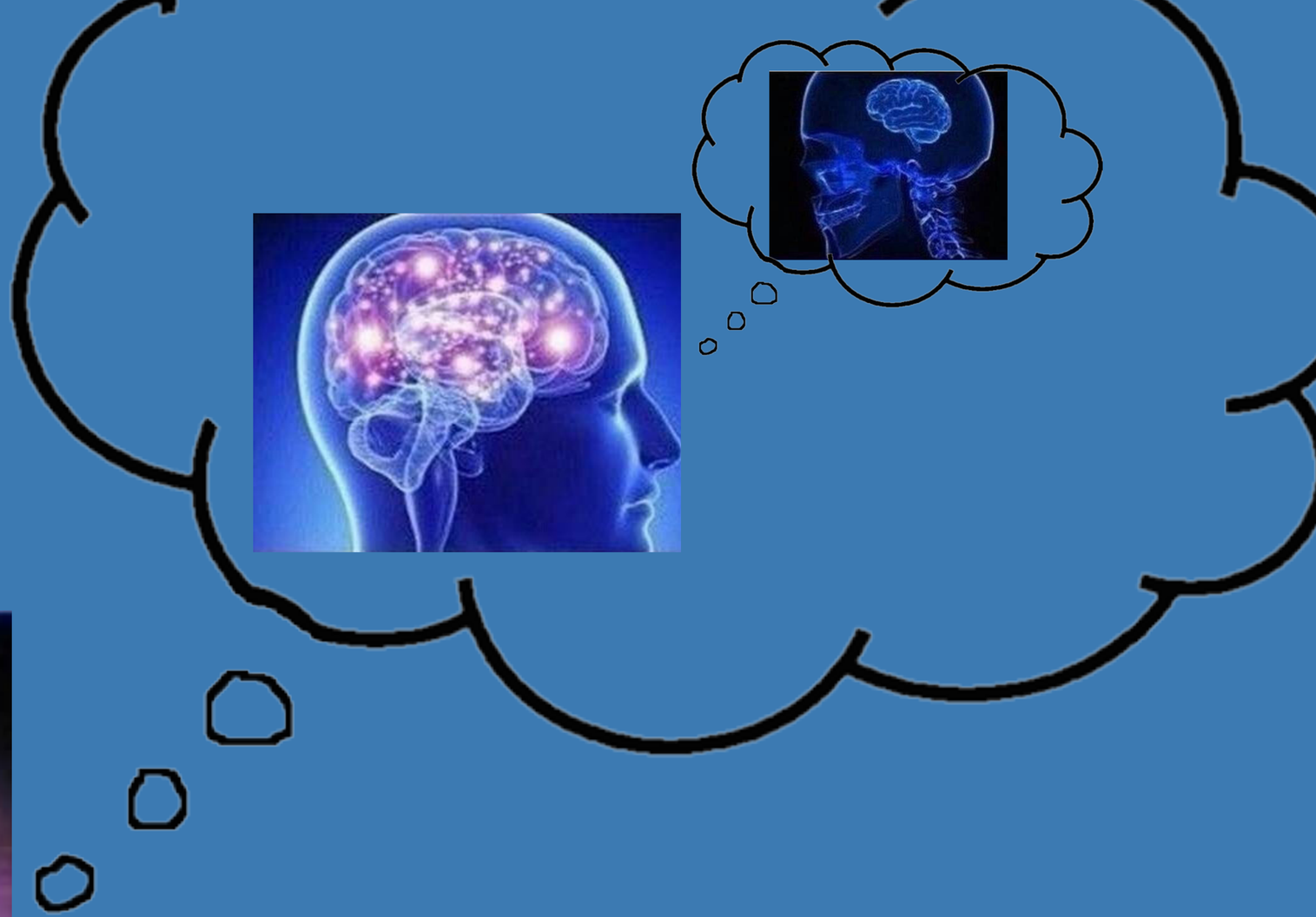
Brendan McMillion

One pairing per symmetric key



One pairing per Diffie-Hellman public key

One key exchange per symmetric key



For each config, generate scalars a, b, c

IBBE KEM(a) region, aP
IBR KEM(b) location, bP
IBBE KEM(c) location, cP

For each TLS key, generate scalar d
compute KEK = d(aP+bP)
KEK escrow = d(cP)

KEK(private key)
KEK escrow(KEK)
dP

decrypt c, compute KEK escrow c(dP), decrypt KEK or

decrypt a and b and compute KEK = (a+b)dP

Share KEMs between keys

Real World Crypto
January 11, 2018

Geo Key Manager

Nick Sullivan (@grittygrease)

Brendan McMillion

References

Nick Sullivan, Douglas Stebila “An Analysis of TLS Handshake Proxying”
http://files.douglas.stebila.ca/files/research/papers/TrustCom-SteSul15.pdf

Dan Boneh, Matt Franklin “Identity-Based Encryption from the Weil Pairing”
https://crypto.stanford.edu/~dabo/papers/bfibe.pdf

Paulo S. L. M. Barreto and Michael Naehrig “Pairing-Friendly Elliptic Curves of Prime Order”
https://eprint.iacr.org/2005/133.pdf

Augusto Jun Devegili, Michael Scott, and Ricardo Dahab “Implementing Cryptographic Pairings over Barreto-Naehrig Curves"
https://eprint.iacr.org/2007/390.pdf

Taechan Kim and Razvan Barbulescu , “Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case”
https://eprint.iacr.org/2015/1027

Cécile Delerablée "Identity-based broadcast encryption with constant size ciphertexts and private keys.”
https://link.springer.com/content/pdf/10.1007/978-3-540-76900-2_12.pdf

Dan Boneh, Craig Gentry, Brent Waters, “Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys”
https://eprint.iacr.org/2005/018.pdf

Nuttapong Attrapadung, Benoıt Libert, and Elie de Panafieu “Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts”
https://pdfs.semanticscholar.org/5da9/eaa24ba749f1ae193800b6961a37b88da1de.pdf