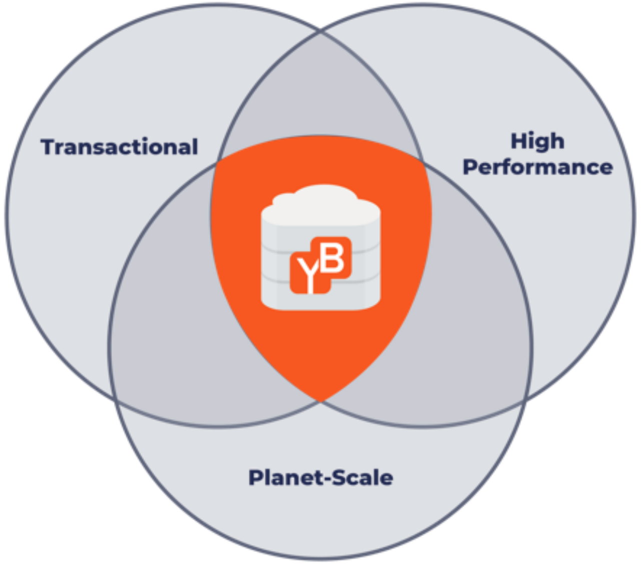


Distributed Database Architecture for GDPR

Karthik Ranganathan

PostgresConf Silicon Valley
Oct 15, 2018

YugaByte

About Us

Founders

Kannan Muthukkaruppan, CEO
Nutanix ♦ Facebook ♦ Oracle
IIT-Madras, University of California-Berkeley

Karthik Ranganathan, CTO
Nutanix ♦ Facebook ♦ Microsoft
IIT-Madras, University of Texas-Austin

Mikhail Bautin, Software Architect
ClearStory Data ♦ Facebook ♦ D.E.Shaw
Nizhny Novgorod State University, Stony Brook

- ✓ Founded Feb 2016
- ✓ Apache HBase committers and early engineers on Apache Cassandra
- ✓ Built Facebook's NoSQL platform powered by Apache HBase
- ✓ Scaled the platform to serve many mission-critical use cases
  - Facebook Messages (Messenger)
  - Operational Data Store (Time series Data)
- ✓ Reassembled the same Facebook team at YugaByte along with engineers from Oracle, Google, Nutanix and LinkedIn

© 2018 All rights reserved.

WHAT IS YUGABYTE DB?

© 2018 All rights reserved.

YugaByte DB

A **transactional, planet-scale** database for building **high-performance** cloud services.

YugaByte DB

NoSQL + SQL

Cloud Native

Design Principles

TRANSACTIONAL

- Single Shard & Distributed ACID Txns
- Document-Based, Strongly Consistent Storage

HIGH PERFORMANCE

- Low Latency, Tunable Reads
- High Throughput

PLANET-SCALE

- Global Data Distribution
- Auto Sharding & Rebalancing

CLOUD NATIVE

- Built For The Container Era
- Self-Healing, Fault-Tolerant

OPEN SOURCE

- Apache 2.0
- **Popular APIs Extended**
  Apache Cassandra, Redis and PostgreSQL (BETA)

WHAT IS GDPR?

© 2018 All rights reserved.

GDPR : **G**eneral **D**ata **P**rotection **R**egulation

Citizens of EU can **control sharing and protection** of their **personal data** by businesses.

Personal Data, also called

PII *(**P**ersonally **I**dentifiable **I**nformation)*

- User name
- Email address
- Date of birth
- Bank details
- Location details
- Computer IP address

Control over personal data

- Consent & data location
- Data privacy and safety
- Right to be forgotten
- Data access on demand

Database concerns

- Notify on data breach
- Data portability
- Ability to fix errors in data
- Restrict processing

Application concerns

© 2018 All rights reserved.

#1 USER CONSENT AND DATA LOCATION

© 2018 All rights reserved.

Data must be stored in EU by default. Businesses need explicit user consent to move it outside.

Why is this hard?

- EU user data lives in that region
- Other countries have compliance regulation – more geo’s
- Public clouds may not have coverage – hybrid deployments
- Architecture depends on data – multiple per service

Think Global Deployments first!

© 2018 All rights reserved.

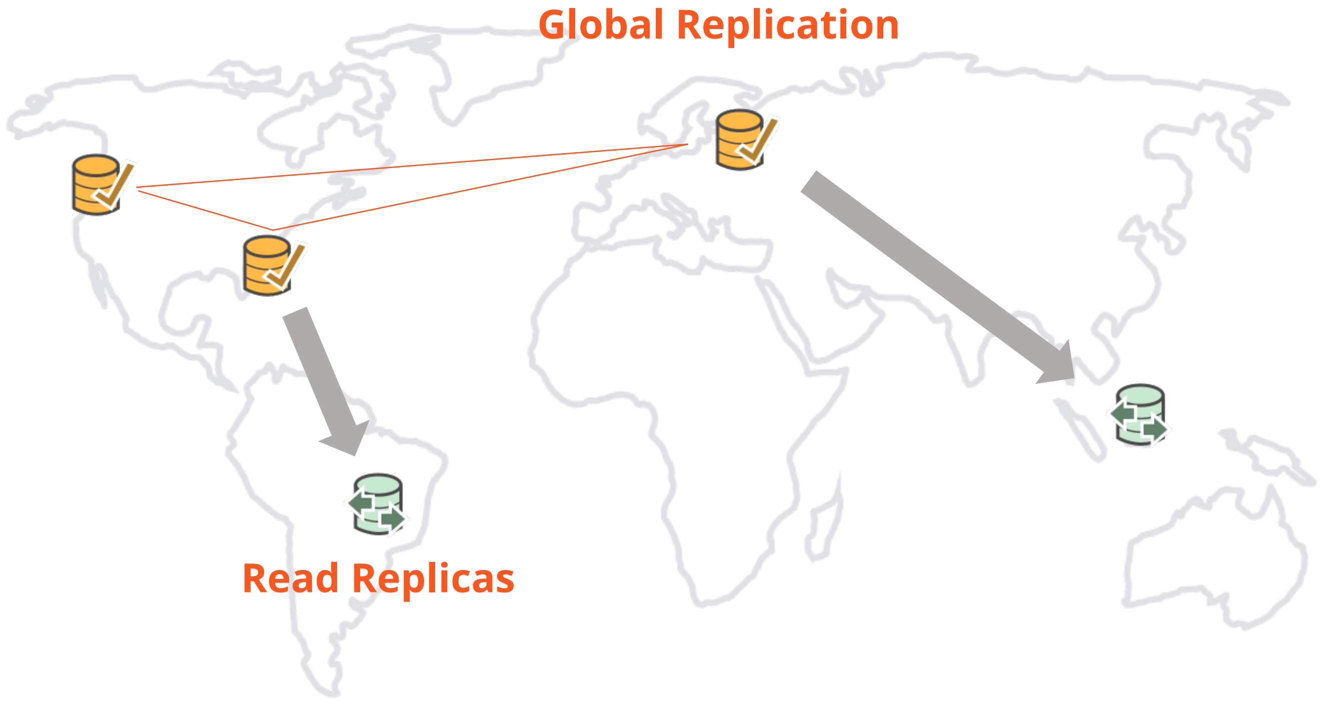
Example – online ecommerce site

- **Products** table needs globally replication – not PII data

© 2018 All rights reserved.

Non-PII Data

Global Replication with YugaByte DB

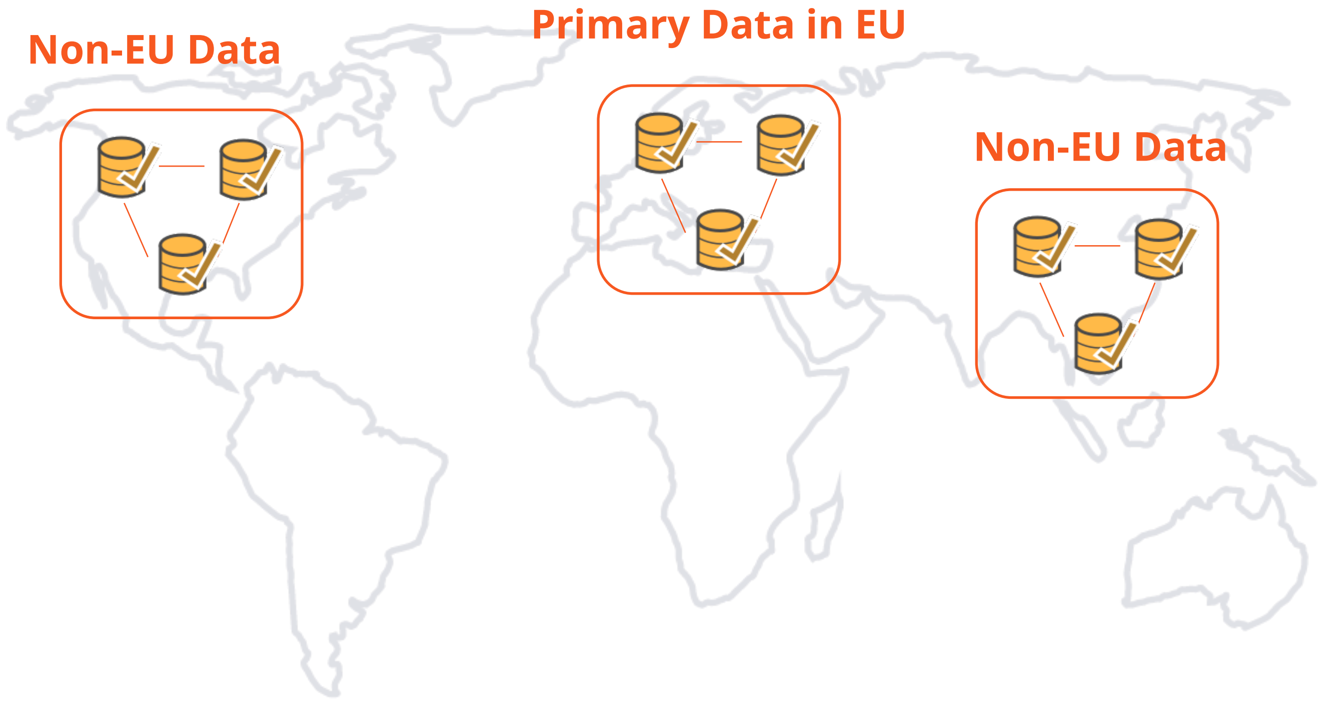


Example – online ecommerce site

- **Users**, **orders** and **shipments** needs locality – PII data
- **Product locations** table needs scale – may be PII

PII Data

Geo-Partitioning with YugaByte DB



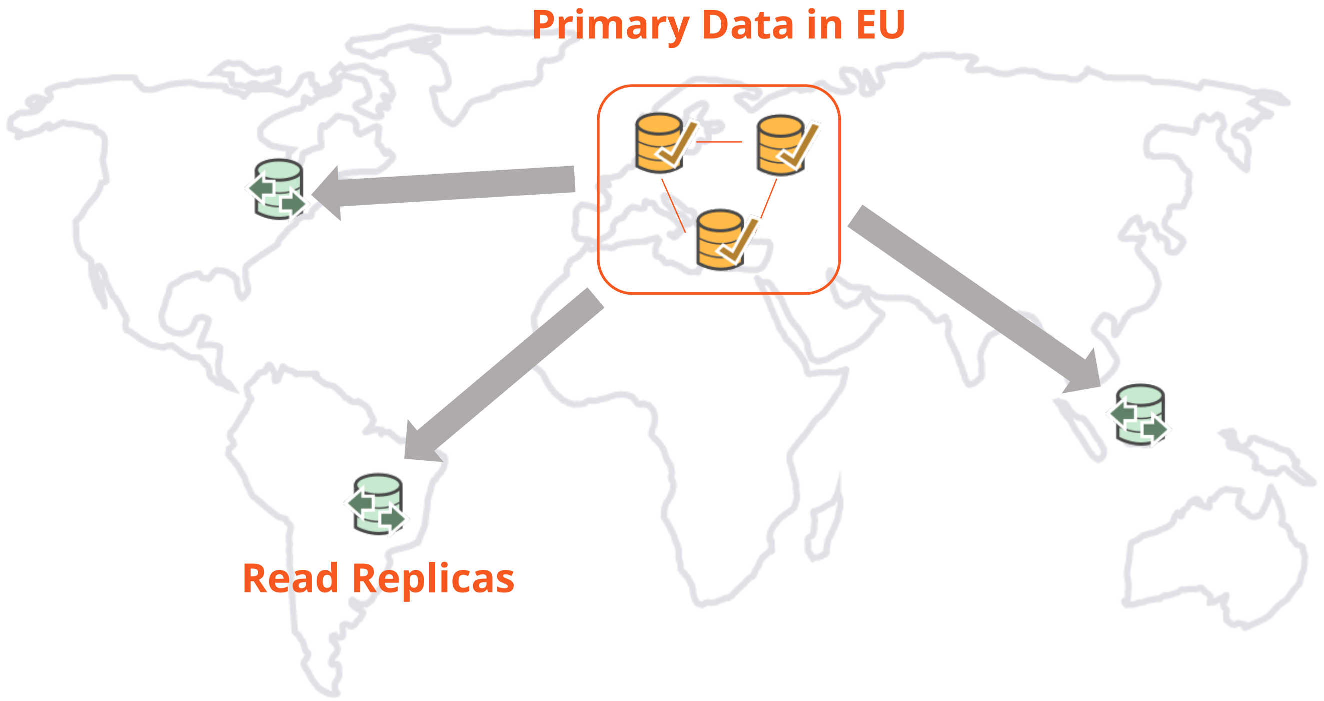
Replicate data on demand to other geo’s

- User may be ok with replicating data
- Read replicas on demand (for remote, low-latency reads)
- Change data capture (for analytics)

© 2018 All rights reserved.

PII Data with YugaByte DB

Read Replicas with YugaByte DB



#2 DATA PRIVACY AND SAFETY

© 2018 All rights reserved.

Data must be secured by using best practices by default. Users need to be notified on breach.

Implement end-to-end encryption on day #1

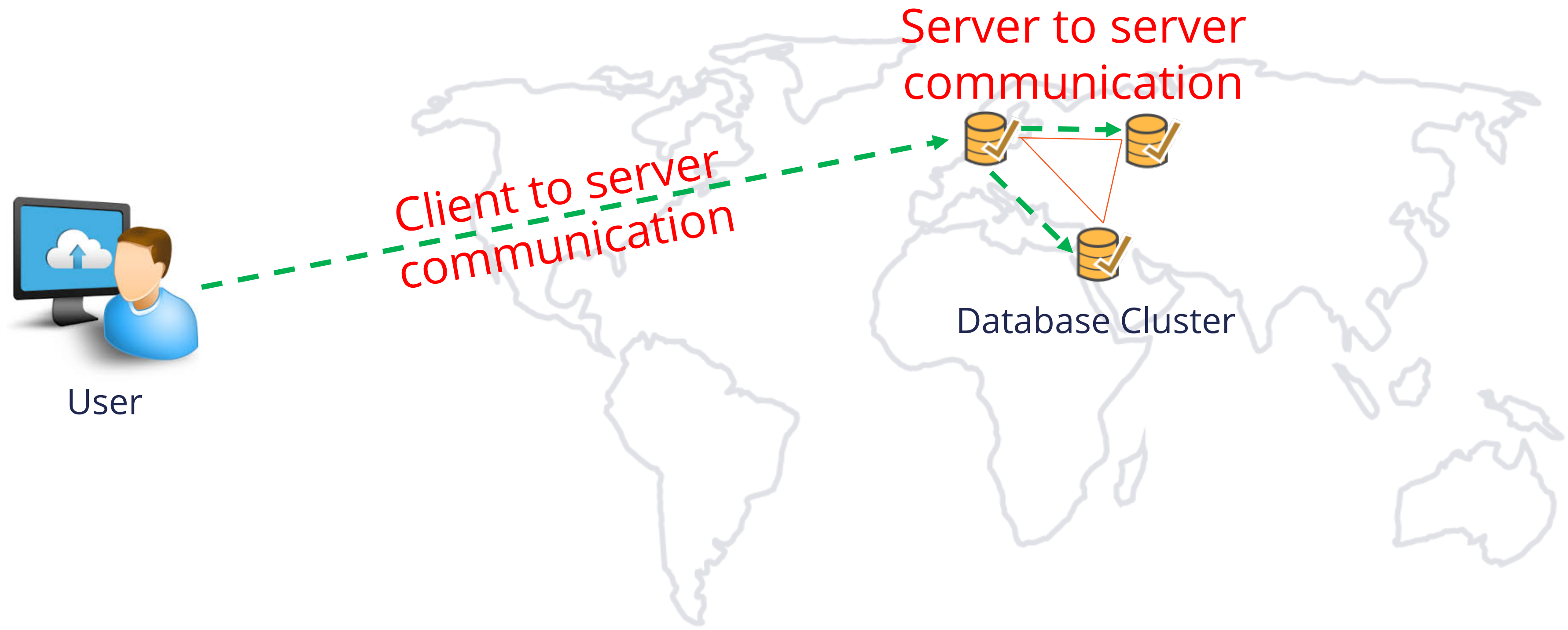
© 2018 All rights reserved.

Encrypt All Network Communication

- Use TLS Encryption
- Between client and server for app interaction
- Between database servers for replication

YugaByte © 2018 All rights reserved.

TLS Encryption


Server to server communication
Client to server communication
User
Database Cluster

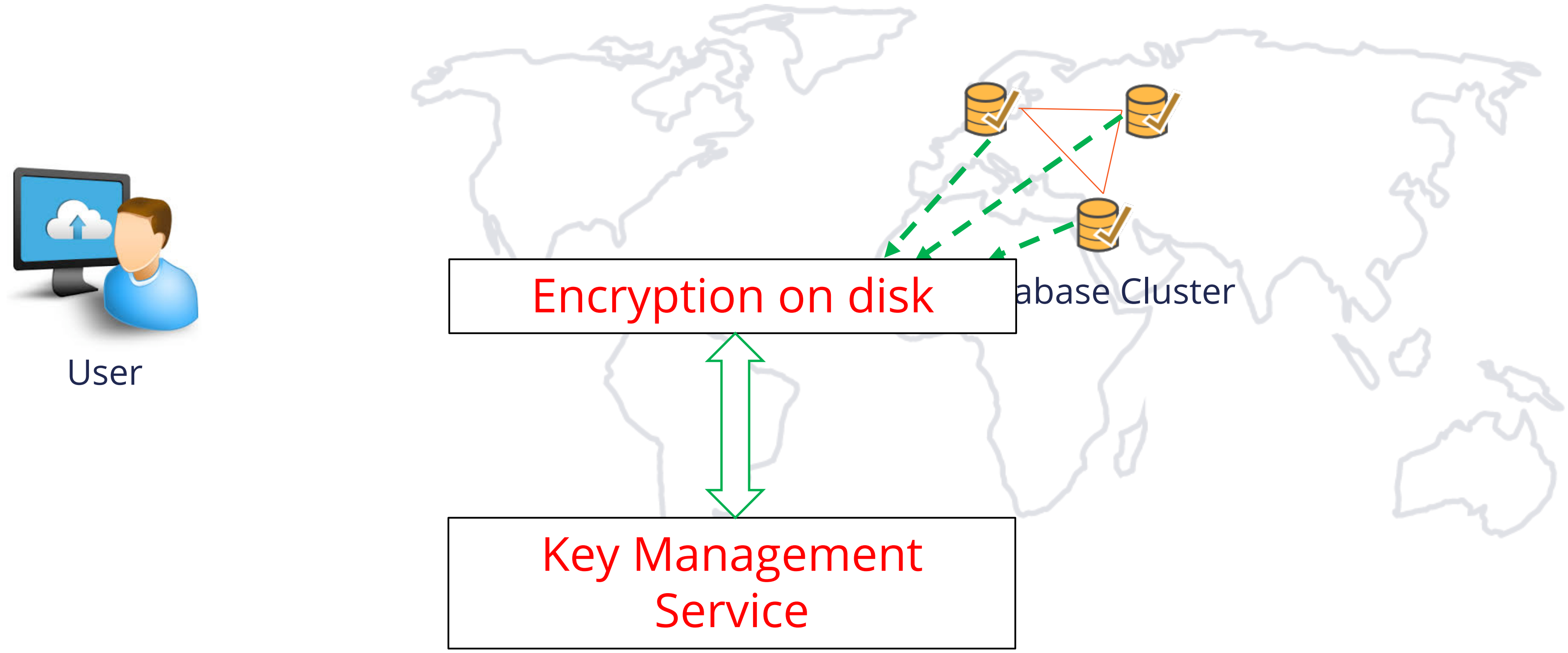
Encryption All Storage

- Encryption at rest
- Integrate with external Key Management Systems
- Ability to rotate keys on demand

Have a key-value table with id to cipher key. Encrypt PII data with the cipher key for fine-grained control. More in the next section.

© 2018 All rights reserved.

Encryption at Rest



#3 RIGHT TO BE FORGOTTEN

© 2018 All rights reserved.

Data must be erased if on explicit request or when data is no longer relevant to original intent.

Use Encryption of Data Attributes

- Have a key-value table with id to cipher key
- Encrypt PII data with the cipher key on write
- Decrypt PII data on access
- Delete cipher key to forget PII data

© 2018 All rights reserved.

Example - Storing User Profile Data

```
SET email=foo@bar.com FOR USER ID=XXX
```

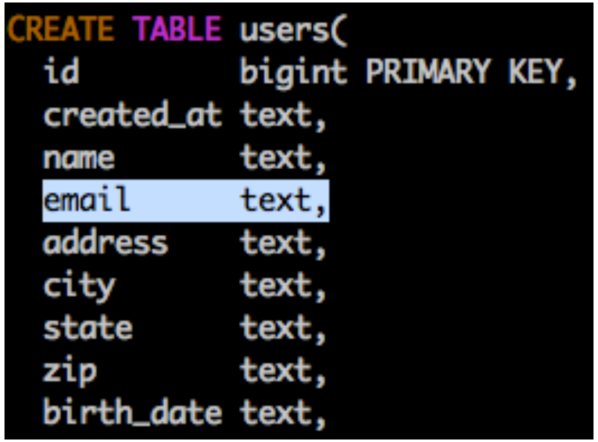
Get encryption key for user

- Reads require decryption
- Data not accessible without key

Encryption PII Data

```
SET email=ENCRYPTED FOR USER ID=XXX
```

Store encrypted data



```
CREATE TABLE users(
  id         bigint PRIMARY KEY,
  created_at text,
  name       text,
  email      text,
  address    text,
  city       text,
  state      text,
  zip        text,
  birth_date text,
```

© 2018 All rights reserved.

Use Anonymization of Data Attributes

- Many cases where value not needed
- Anonymize PII data with one way hash functions
- Use hashed ids for in data warehouse
- There is no PII data if hashed ids are used!

© 2018 All rights reserved.

Example – Website Analytics

```
USER=foo@bar.com CHECKED OUT PRODUCT=X, CATEGORY=Gadget
```

One-way hash
user id

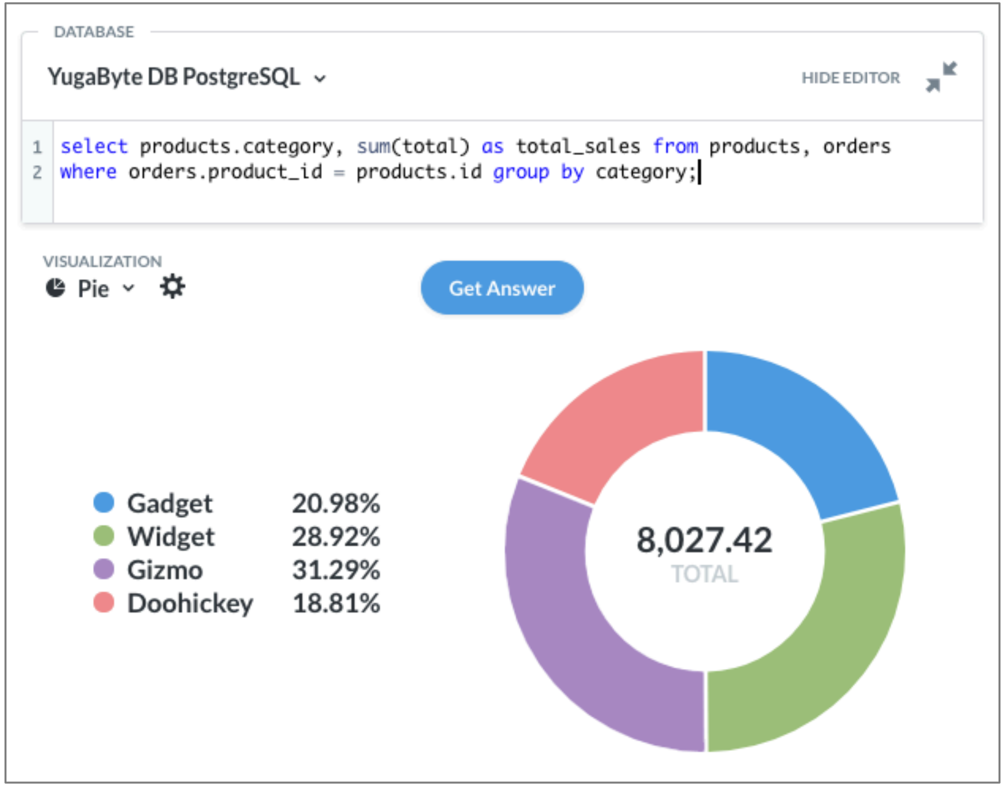
Analytics

```
USER=HASHED_VAL CHECKED OUT PRODUCT=X, CATEGORY=Gadget
```

YugaByte © 2018 All rights reserved.

Example – Website Analytics

- User no longer identifiable
- Hashed data still useful!



```
select products.category, sum(total) as total_sales from products, orders
where orders.product_id = products.id group by category;
```

● Gadget	20.98%
● Widget	28.92%
● Gizmo	31.29%
● Doohickey	18.81%

#4 DATA ACCESS ON DEMAND

© 2018 All rights reserved.

Ability to inform a user about what data is being used, for what purpose and where it is stored.

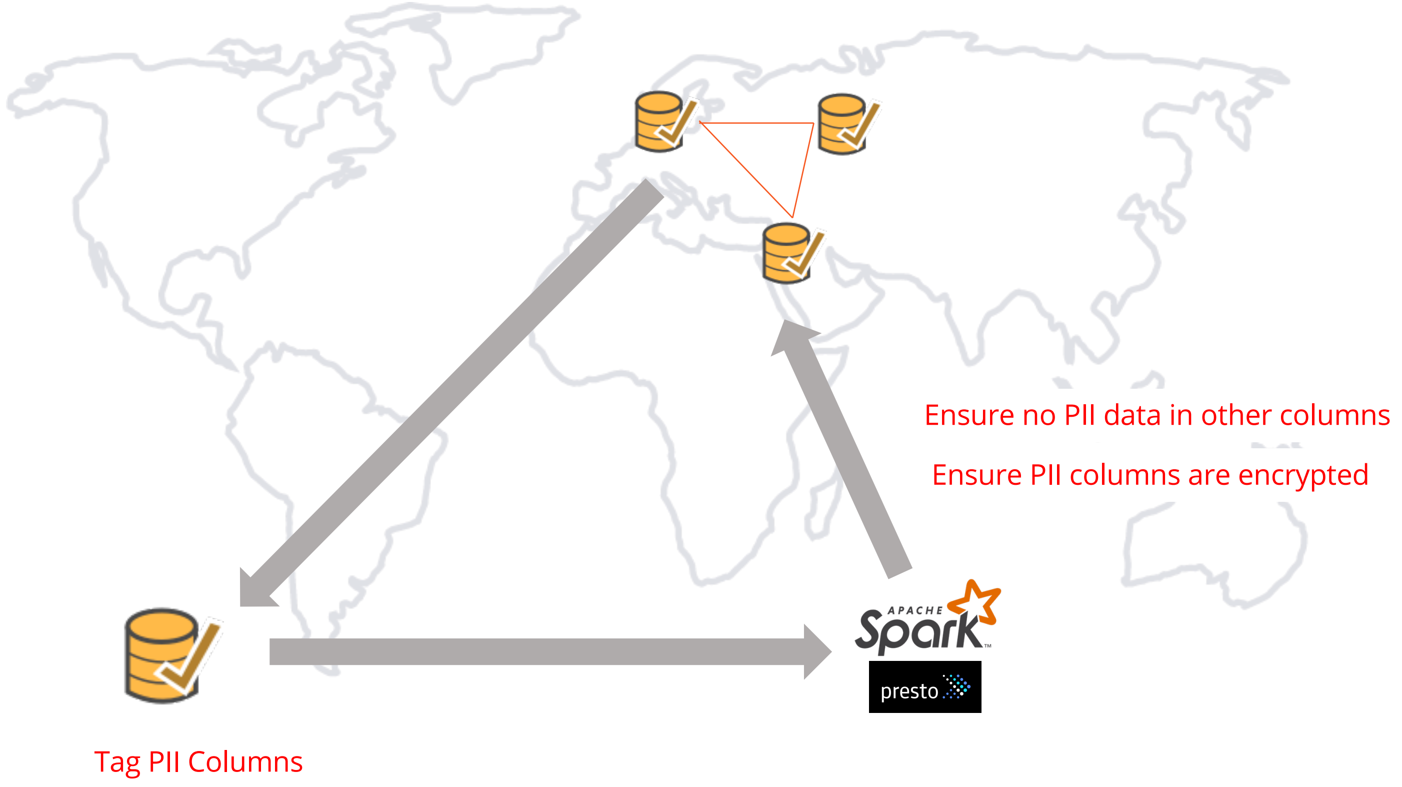
Tag Tables and Columns with PII

- Store in a separate information architecture table
- Make tagging a part of the process
- Easy to find what PII data is stored on demand

Run Continuous Compliance Checks

- Ensure PII are encrypted
- Ensure non-PII columns do not have sensitive data
- Use Spark/Presto to perform scan periodically
- Run scan on a read replica to not impact production

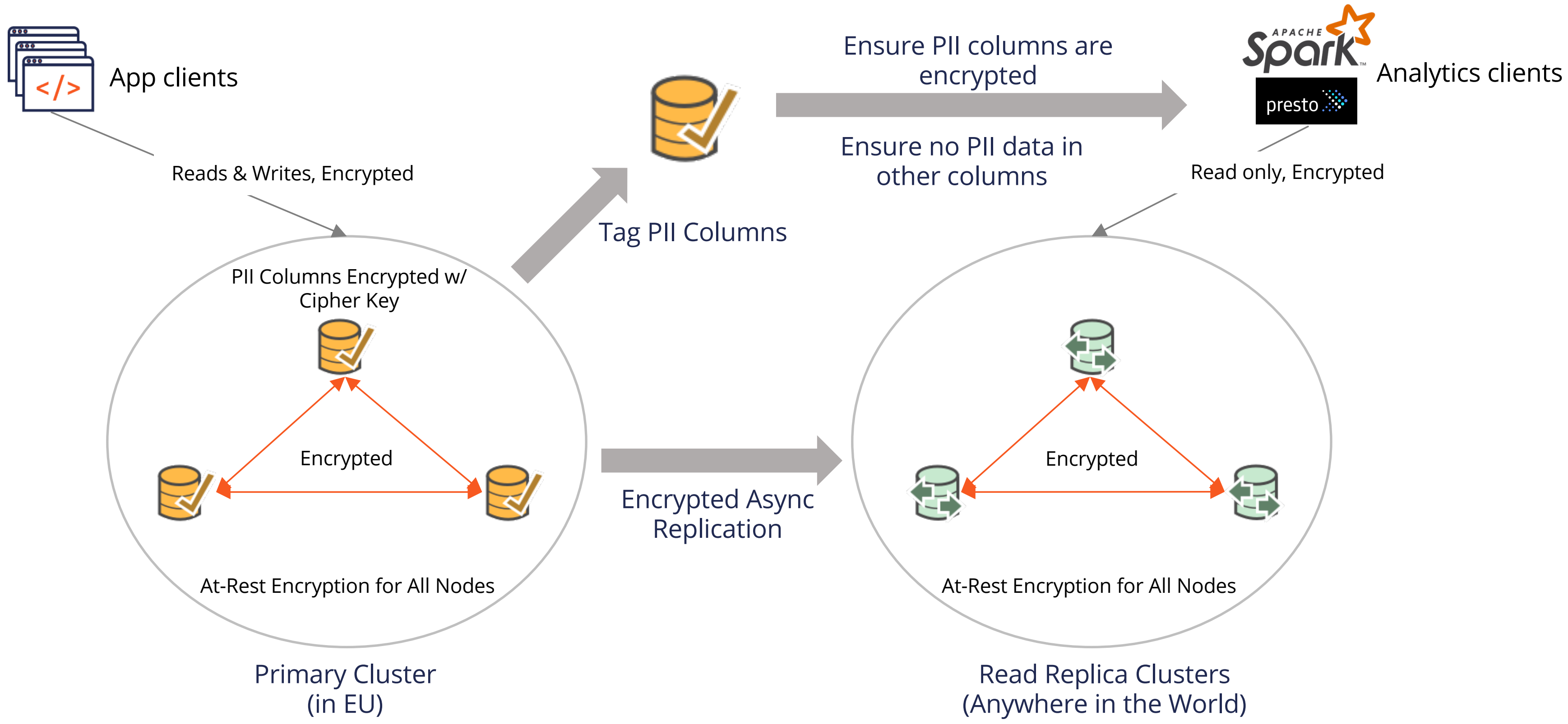
© 2018 All rights reserved.


Ensure no PII data in other columns
Ensure PII columns are encrypted
APACHE Spark
presto
Tag PII Columns

PUTTING IT ALL TOGETHER

© 2018 All rights reserved.

GDPR Reference Architecture


App clients
Reads & Writes, Encrypted
Tag PII Columns
Ensure PII columns are encrypted
Ensure no PII data in other columns
Analytics clients
presto
Read only, Encrypted
PII Columns Encrypted w/ Cipher Key
Encrypted
At-Rest Encryption for All Nodes
Encrypted Async Replication
Encrypted
At-Rest Encryption for All Nodes
Primary Cluster (in EU)
Read Replica Clusters (Anywhere in the World)

Karthik Ranganathan

YugaByte

2018 October 16 09:00

How YugaByte DB implements distributed PostgreSQL

A hands-on introduction to YugaByte DB

Questions?

Try it at docs.yugabyte.com/quick-start