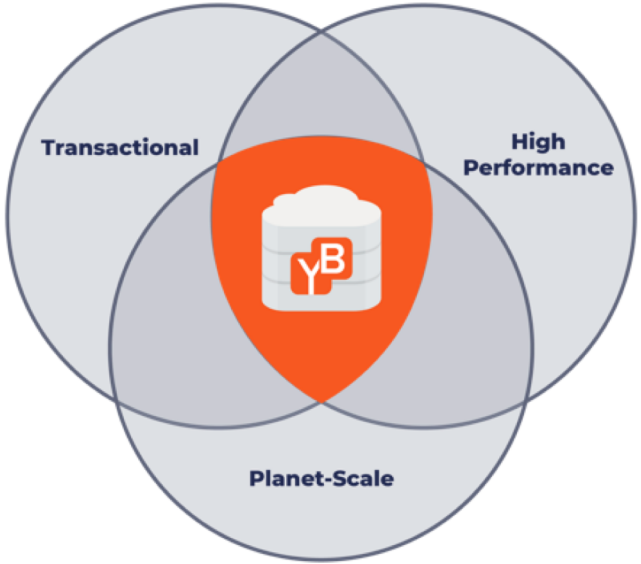


Distributed Database Architecture for GDPR

Karthik Ranganathan
Co-Founder & CTO
Mar, 2019

YugaByte

Introduction

Karthik Ranganathan

Co-Founder & CTO, YugaByte
Nutanix Facebook Microsoft
IIT-Madras, University of Texas-Austin

@karthikr

© 2018 All rights reserved.

What is YugaByte DB?

Distributed SQL (PostgreSQL compatible)

Highly scalable, Fault-Tolerance

High Performance

Low Latency Queries

Cloud Native

Multi-Cloud, multi-region and Kubernetes Ready

© 2018 All rights reserved.

WHAT IS GDPR?

© 2018 All rights reserved.

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

© 2018 All rights reserved.

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

© 2018 All rights reserved.

Personal Data, similar to

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

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

© 2018 All rights reserved.

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.

© 2018 All rights reserved.

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.

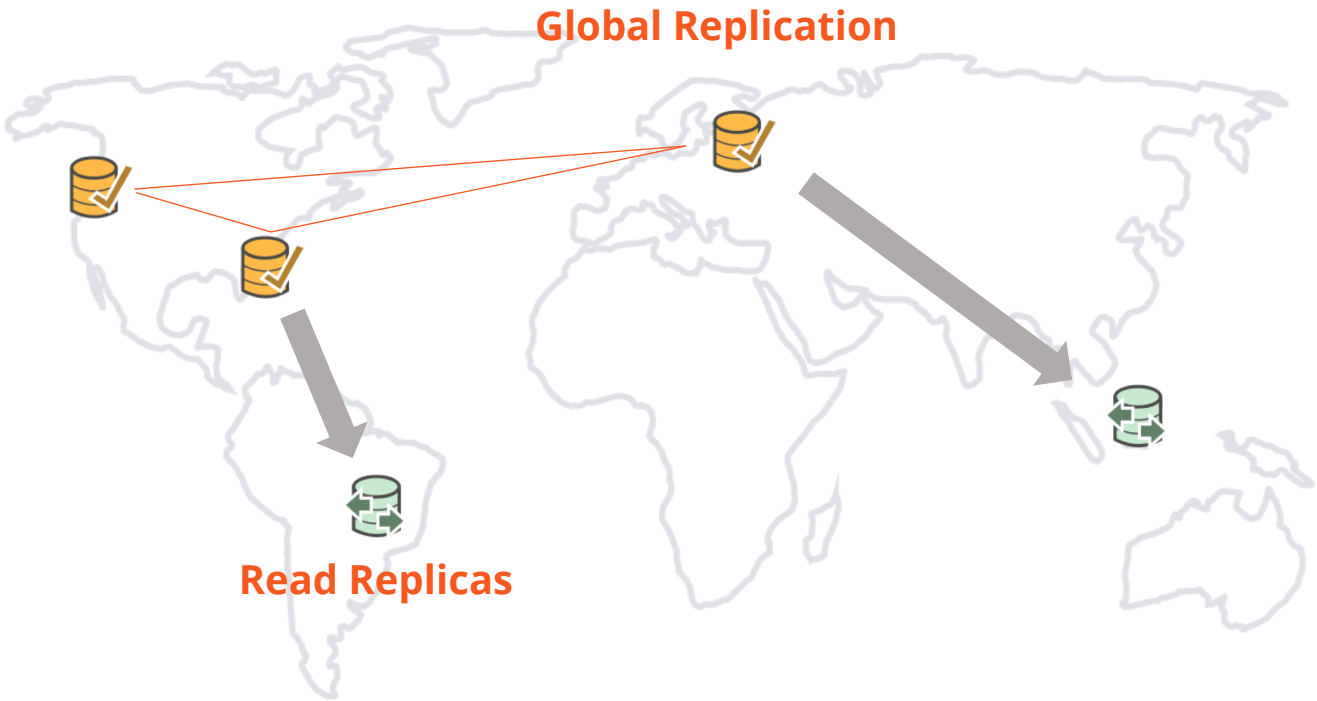
Example – online ecommerce site

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

© 2018 All rights reserved.

Non-PII Data

Global Replication with YugaByte DB



© 2018 All rights reserved.

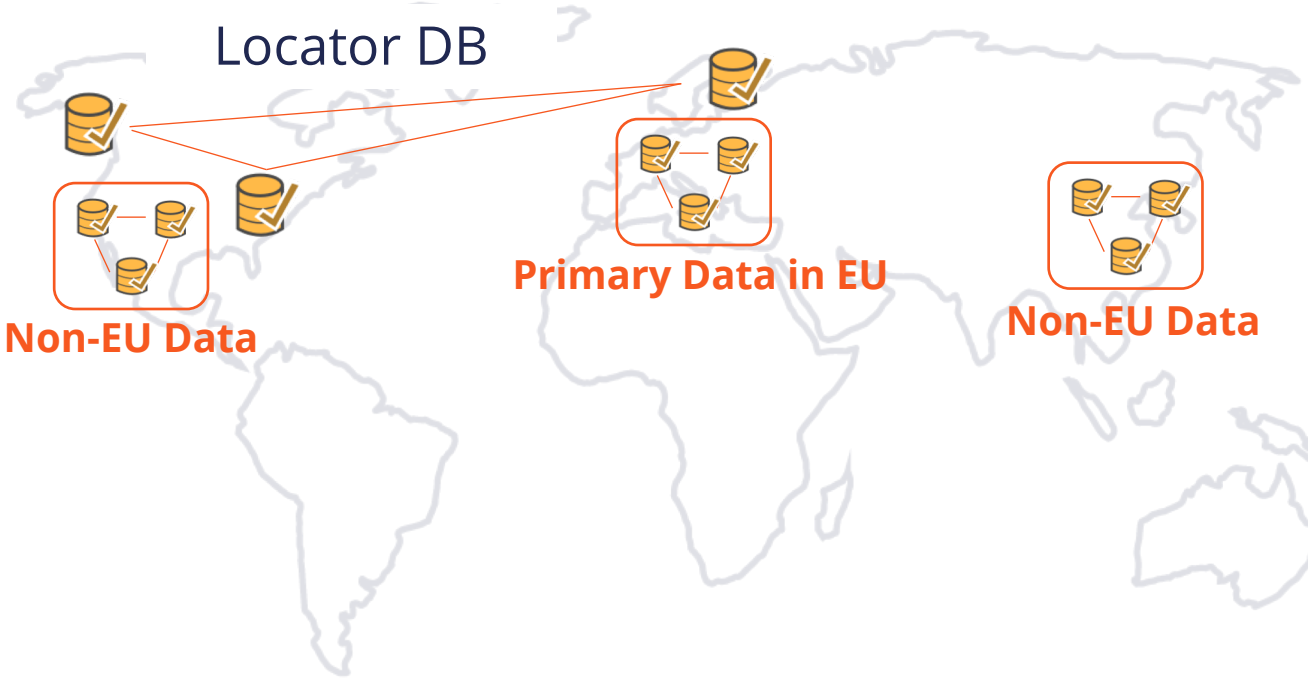
Example – online ecommerce site

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

© 2018 All rights reserved.

PII Data

Geo-Partitioning with YugaByte DB



© 2018 All rights reserved.

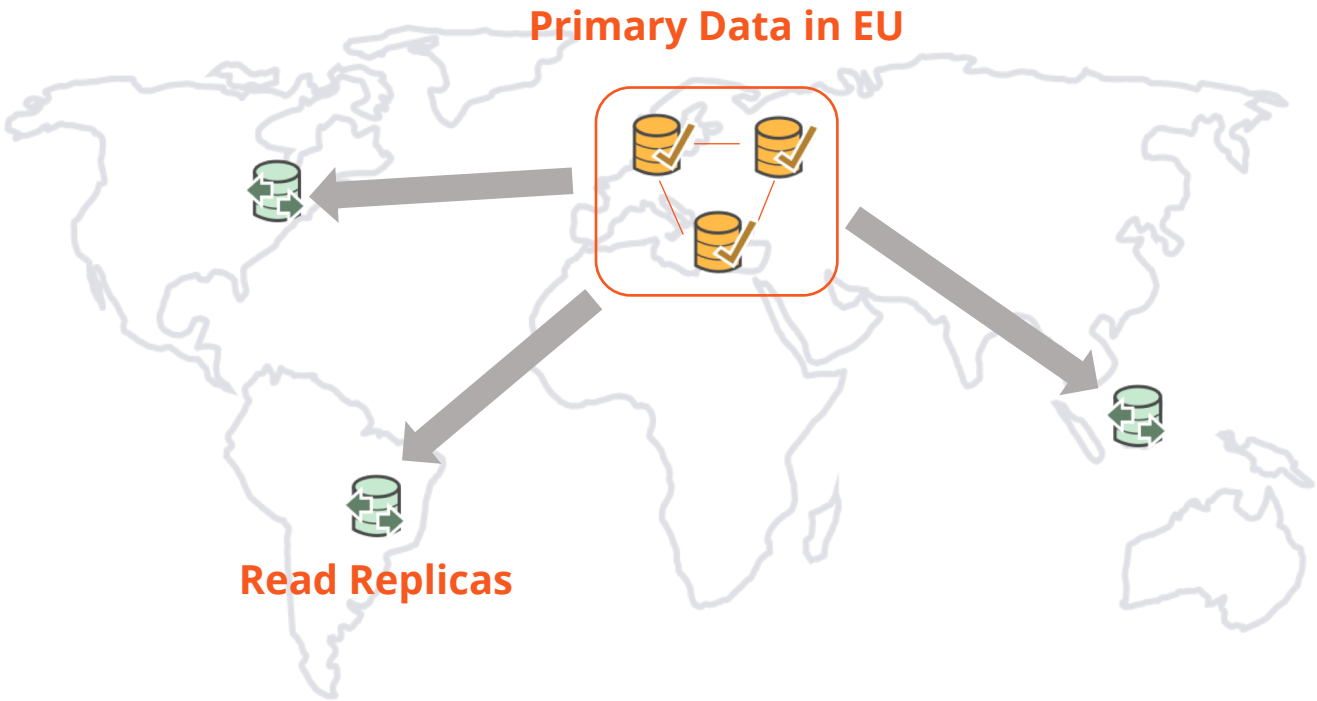
Replicate data on demand to other geo's

- 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



© 2018 All rights reserved.

#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.

© 2018 All rights reserved.

Implement end-to-end encryption on day #1

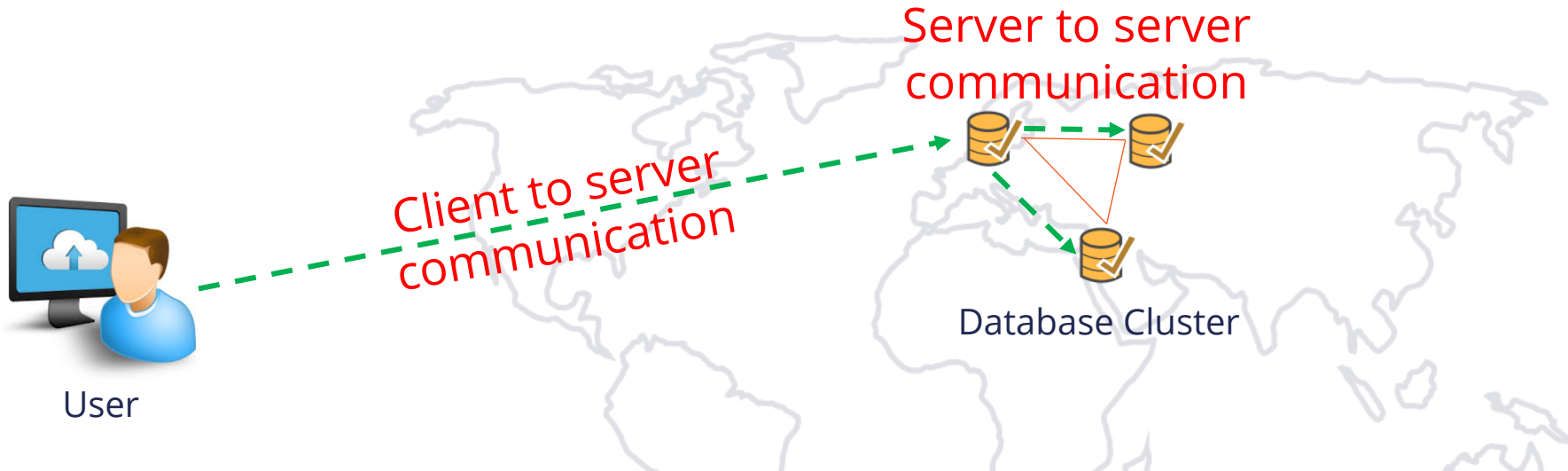
© 2018 All rights reserved.

Encrypt All Network Communication

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

© 2018 All rights reserved.

TLS Encryption



© 2018 All rights reserved.

Encryption All Storage

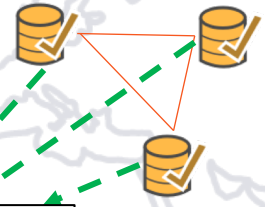
- Encryption at rest
- Integrate with external Key Management Systems
- Ability to rotate keys on demand

Use app level encryption if needed. 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

User



abase Cluster

Encryption on disk

Key Management Service

© 2018 All rights reserved.

#3 RIGHT TO BE FORGOTTEN

© 2018 All rights reserved.

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

© 2018 All rights reserved.

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

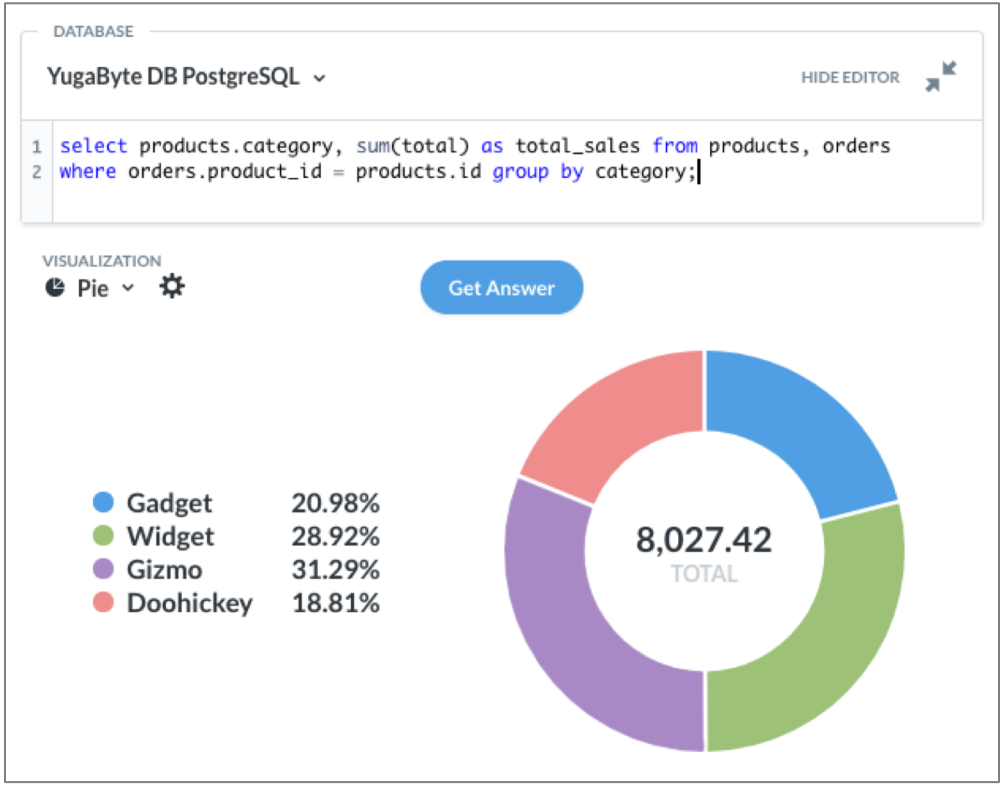
Analytics

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


© 2018 All rights reserved.

Example – Website Analytics

- User no longer identifiable
- Hashed data still useful!



© 2018 All rights reserved.

#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.

© 2018 All rights reserved.

Tag Tables and Columns with PII

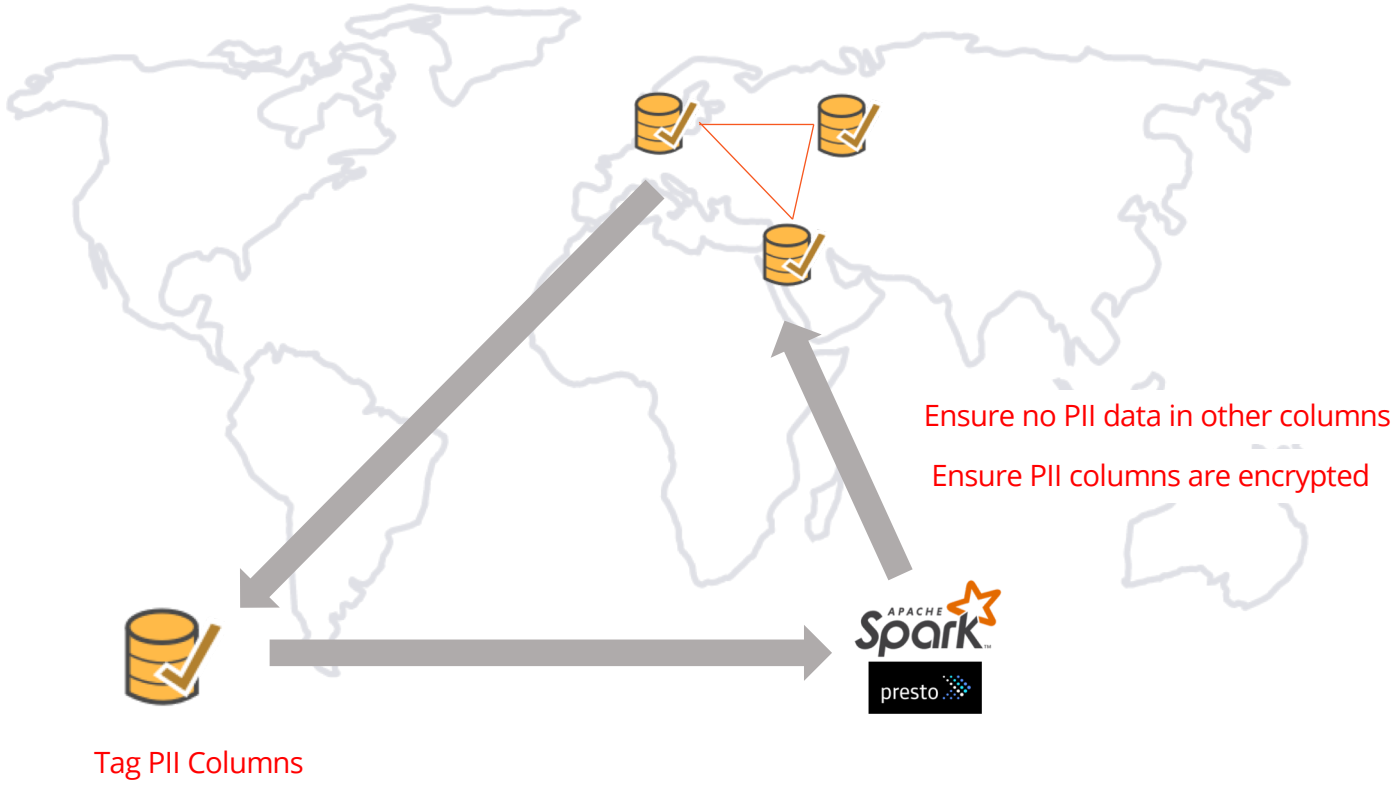
- Store in a separate information architecture table
- Make tagging a part of the process
- Easy to find what PII data is stored on demand

© 2018 All rights reserved.

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.

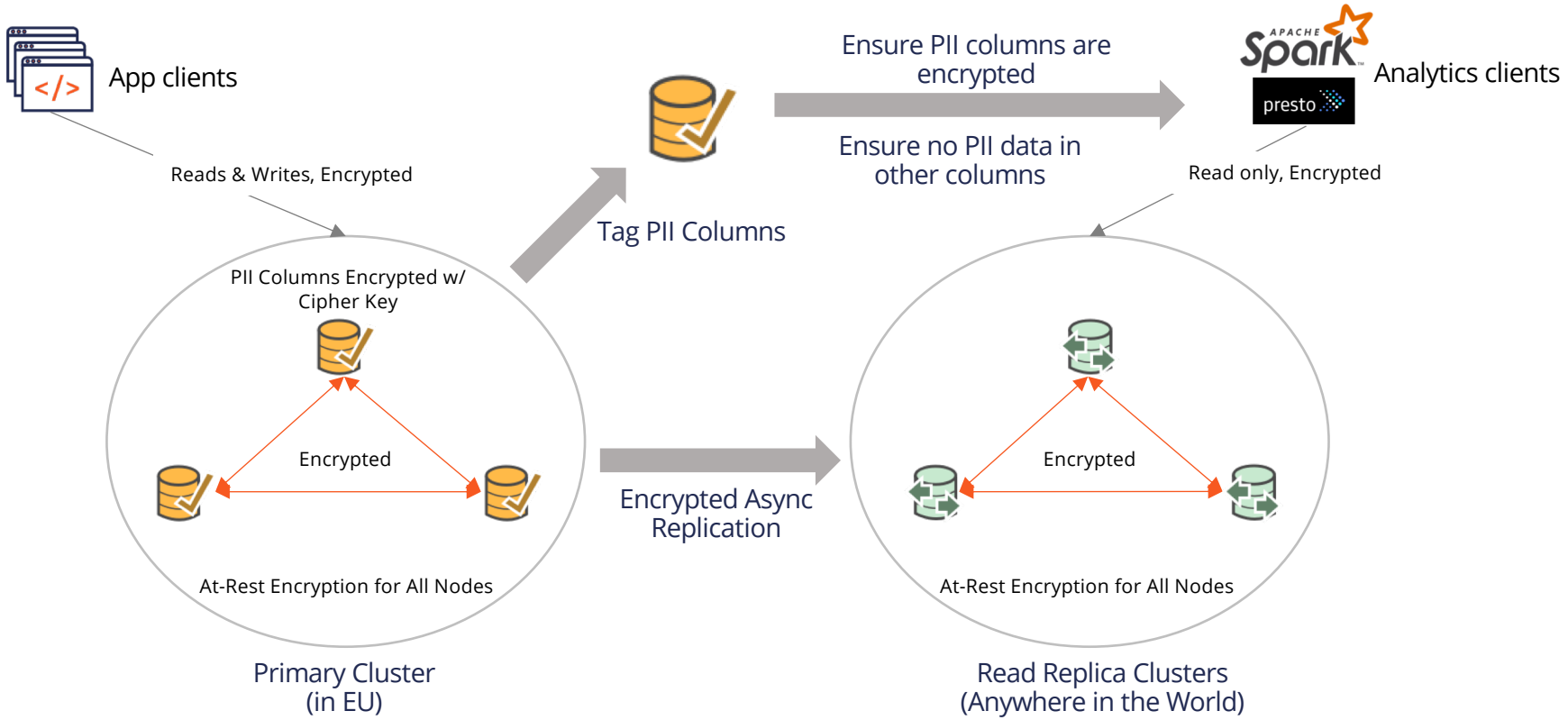


© 2018 All rights reserved.

PUTTING IT ALL TOGETHER

© 2018 All rights reserved.

GDPR Reference Architecture



© 2018 All rights reserved.

Thank You!

Try it at

docs.yugabyte.com/latest/quick-start