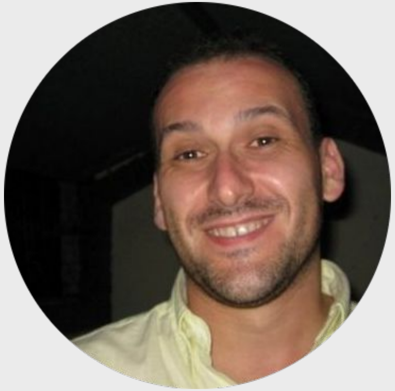
RED HAT SUMMIT

# How to build a European scale instant payments platform

SIA and Red Hat

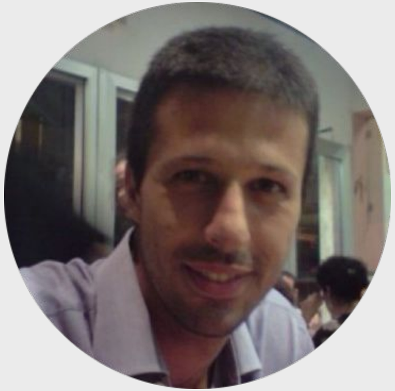
Giovanni Fulco, Giuseppe Bonocore,  Ugo Landini

May 2018



Giovanni Fulco
Software Architect

Ugo Landini
Principal Solution Architect

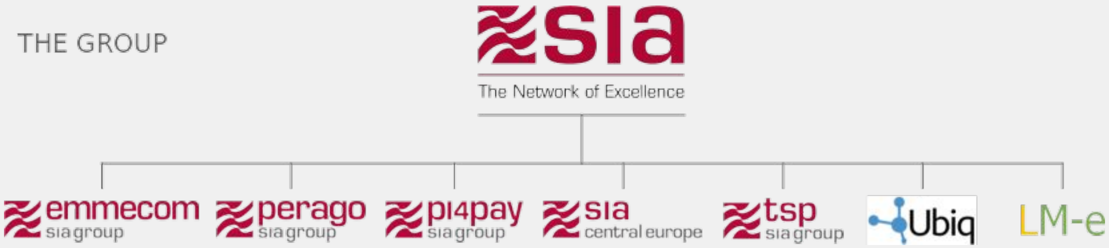


Mattia Ronchi
Senior Technical Analyst

Giuseppe Bonocore
Solution Architect

# SIA in a nutshell

THE GROUP



SIA
The Network of Excellence

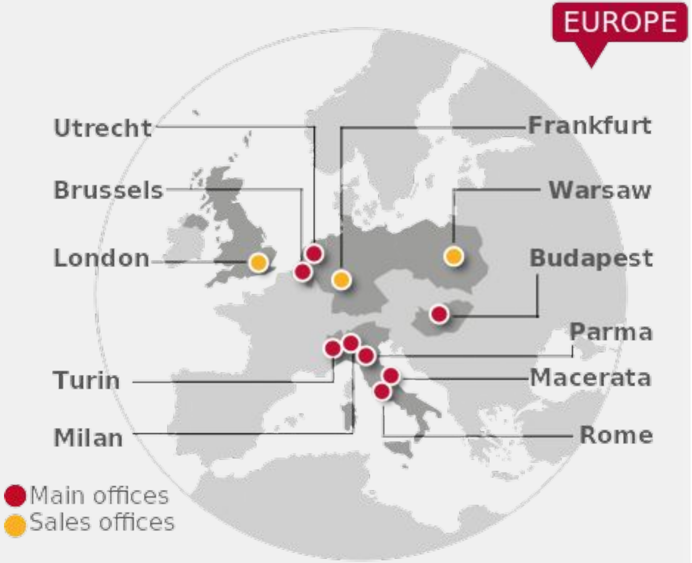
- emmecom sia group
- perago sia group
- pi4pay sia group
- sia central europe
- tsp sia group
- Ubiq
- LM-e

**CARDS** 6.1 billion operations

**PAYMENTS** 3.3 billion transactions

**CAPITAL MARKETS** 56.2 billion financial transactions

**NETWORK** 784 terabytes of data carried



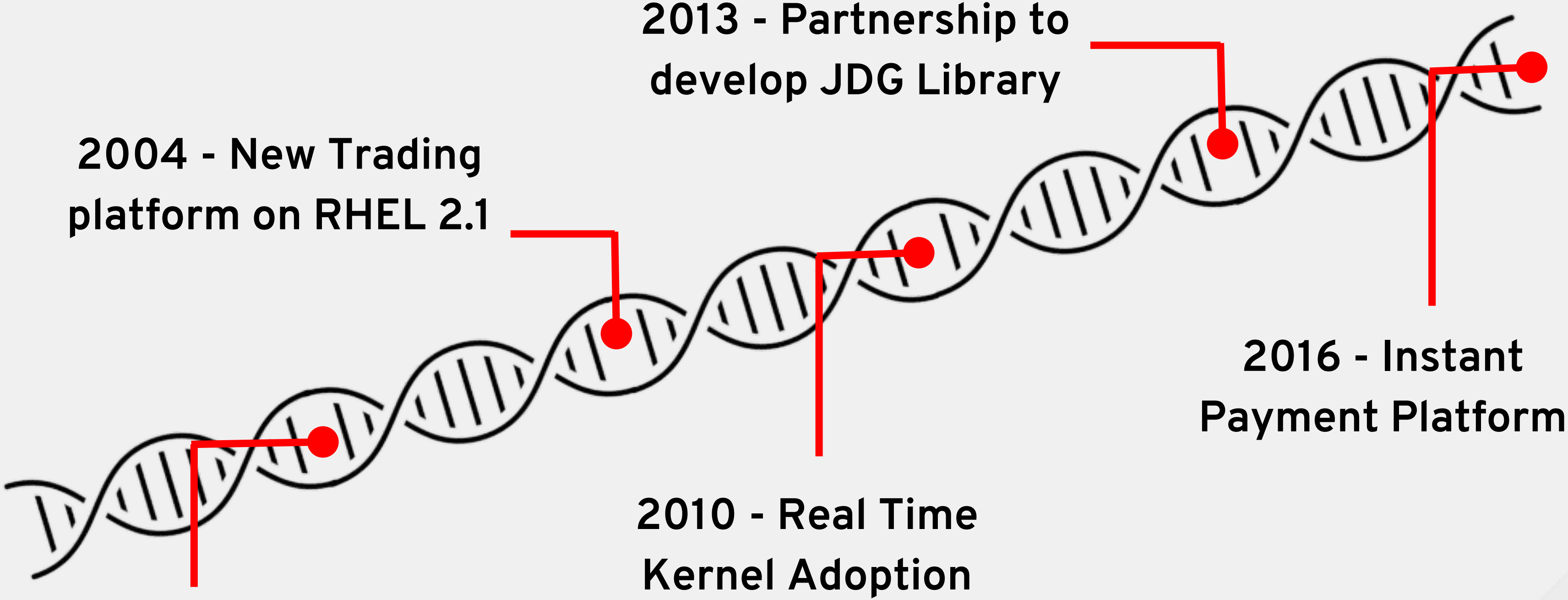
## SIA Vision

*“**Everyone** will be able to use their own money anytime, anywhere, simply and securely.”*

## Red Hat Vision

*“**Open** unlocks the world’s potential.”*

# SIA & Red Hat



2000 - First application delivered on linux OS

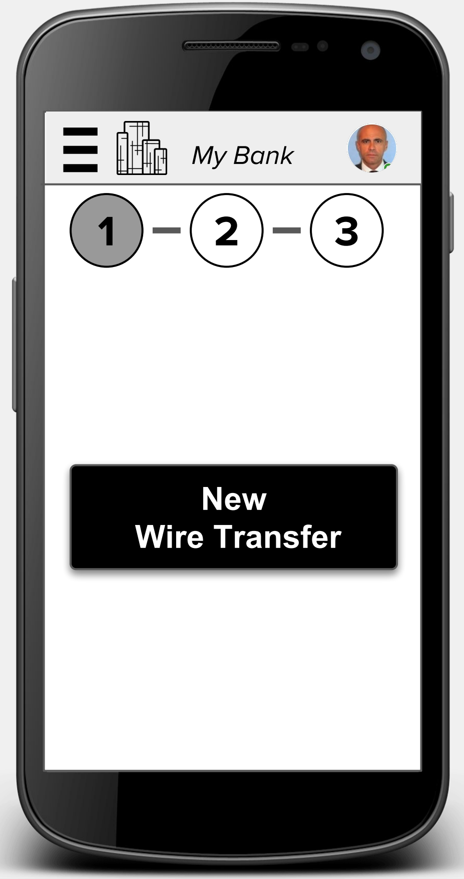
2004 - New Trading platform on RHEL 2.1

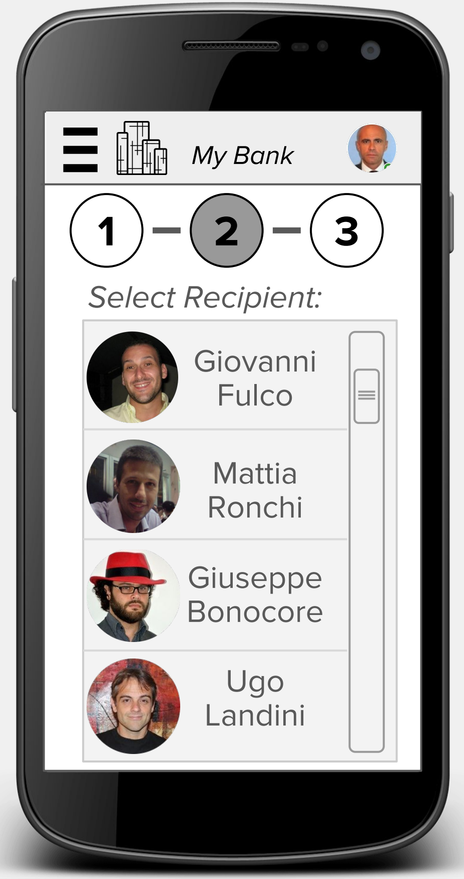
2010 - Real Time Kernel Adoption

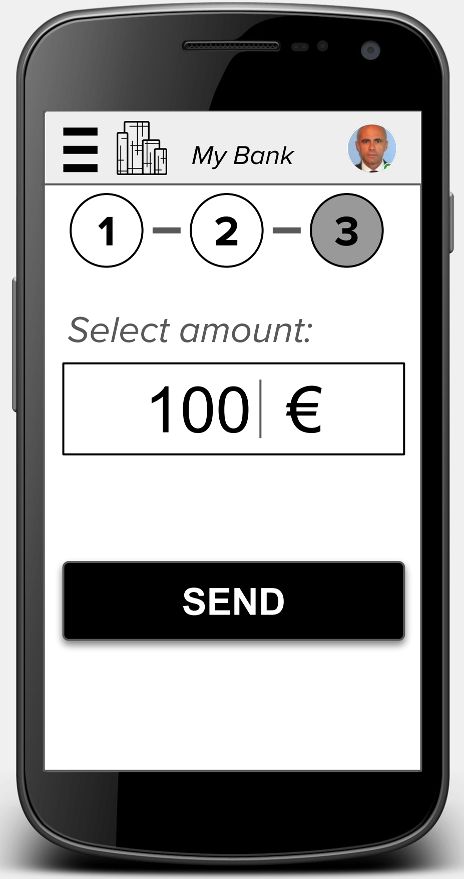
2013 - Partnership to develop JDG Library

2016 - Instant Payment Platform

# Wire transfer **without** Instant Payments


My Bank
1
2
3
New
Wire Transfer


My Bank
1
2
3
Select Recipient:
Giovanni Fulco
Mattia Ronchi
Giuseppe Bonocore
Ugo Landini



Select amount:

100 €

# … File transfer “magic” happens…

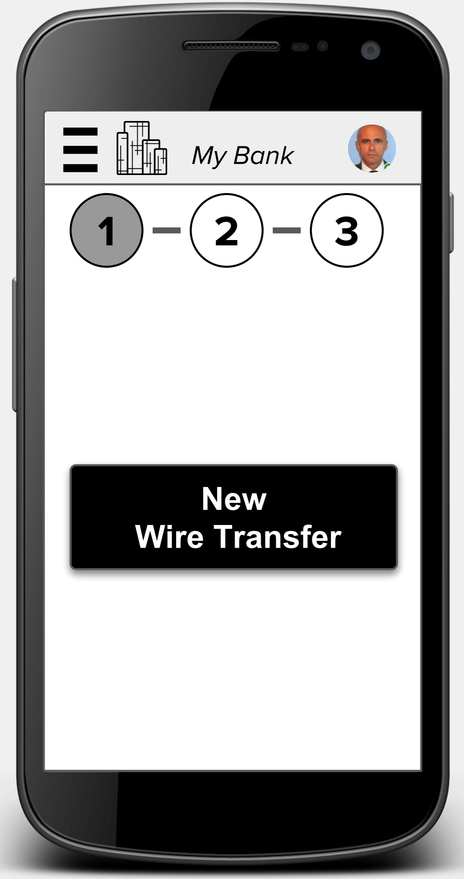
... up to 4 days...

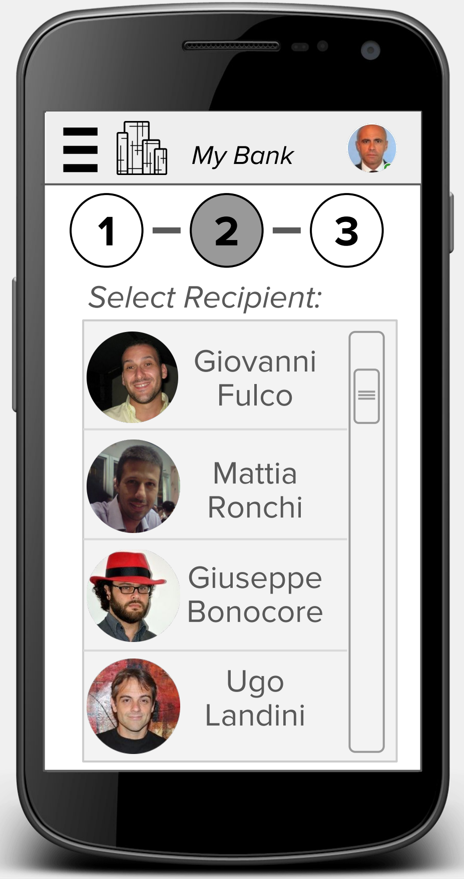
My Homebanking
Search...
OK

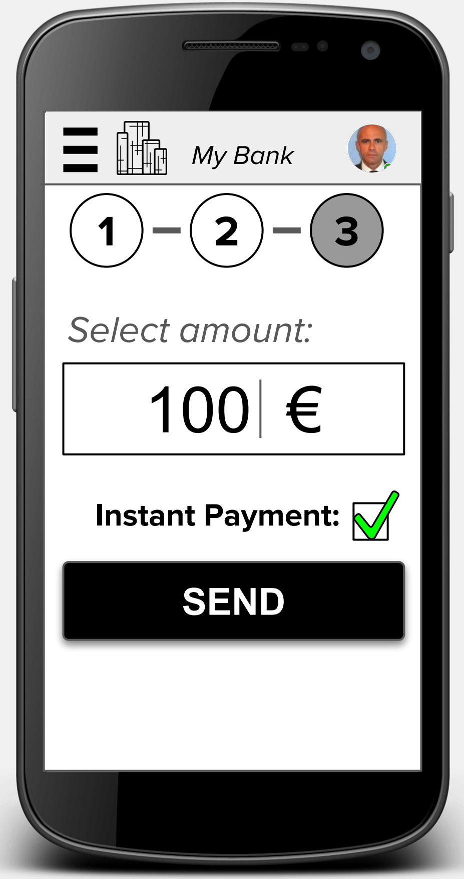
Wire Account Received:

**100 €**

# Wire transfer **with** Instant Payments


My Bank
1
2
3
New
Wire Transfer

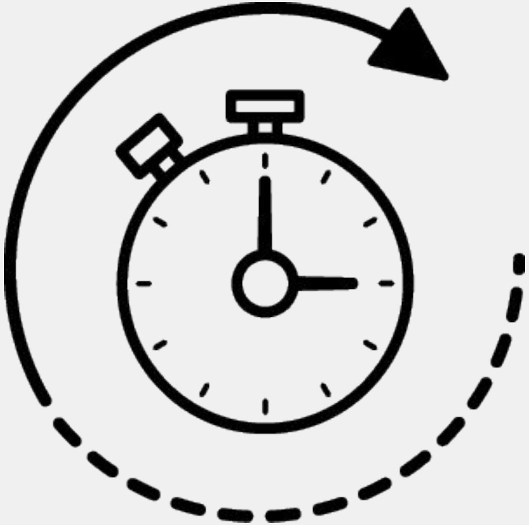

My Bank
1
2
3
Select Recipient:
Giovanni Fulco
Mattia Ronchi
Giuseppe Bonocore
Ugo Landini



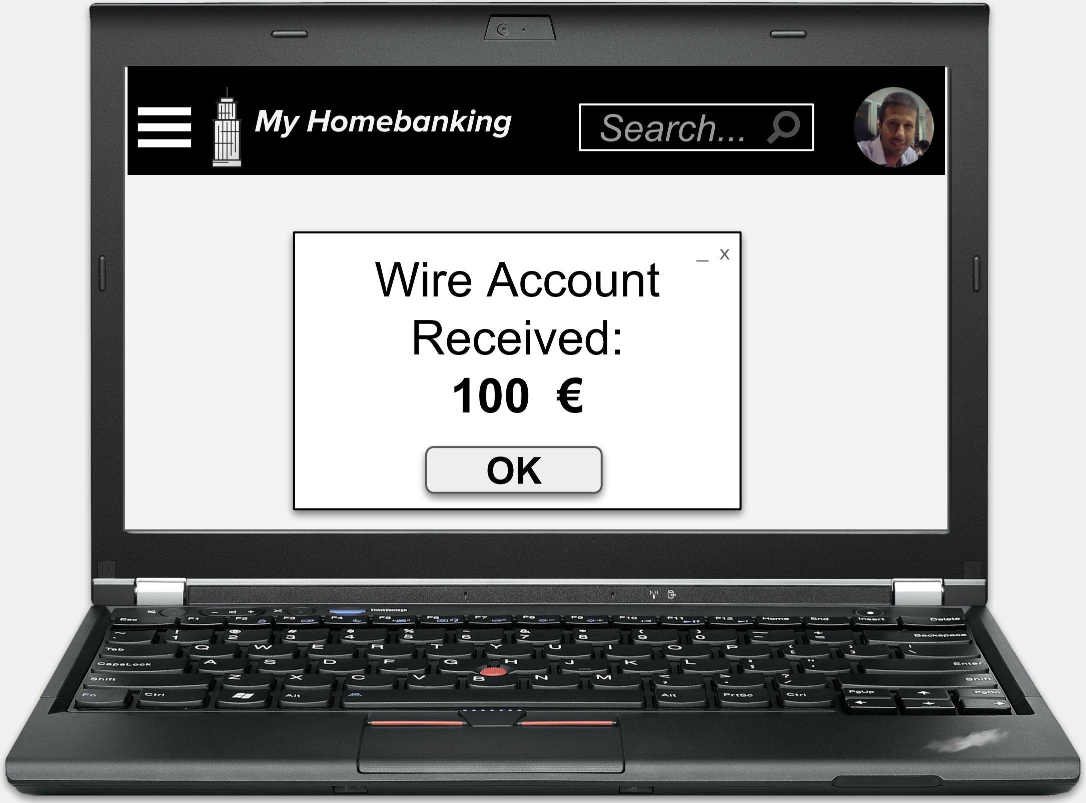
Select amount:

100 €

Instant Payment: ☑



# < 50mS



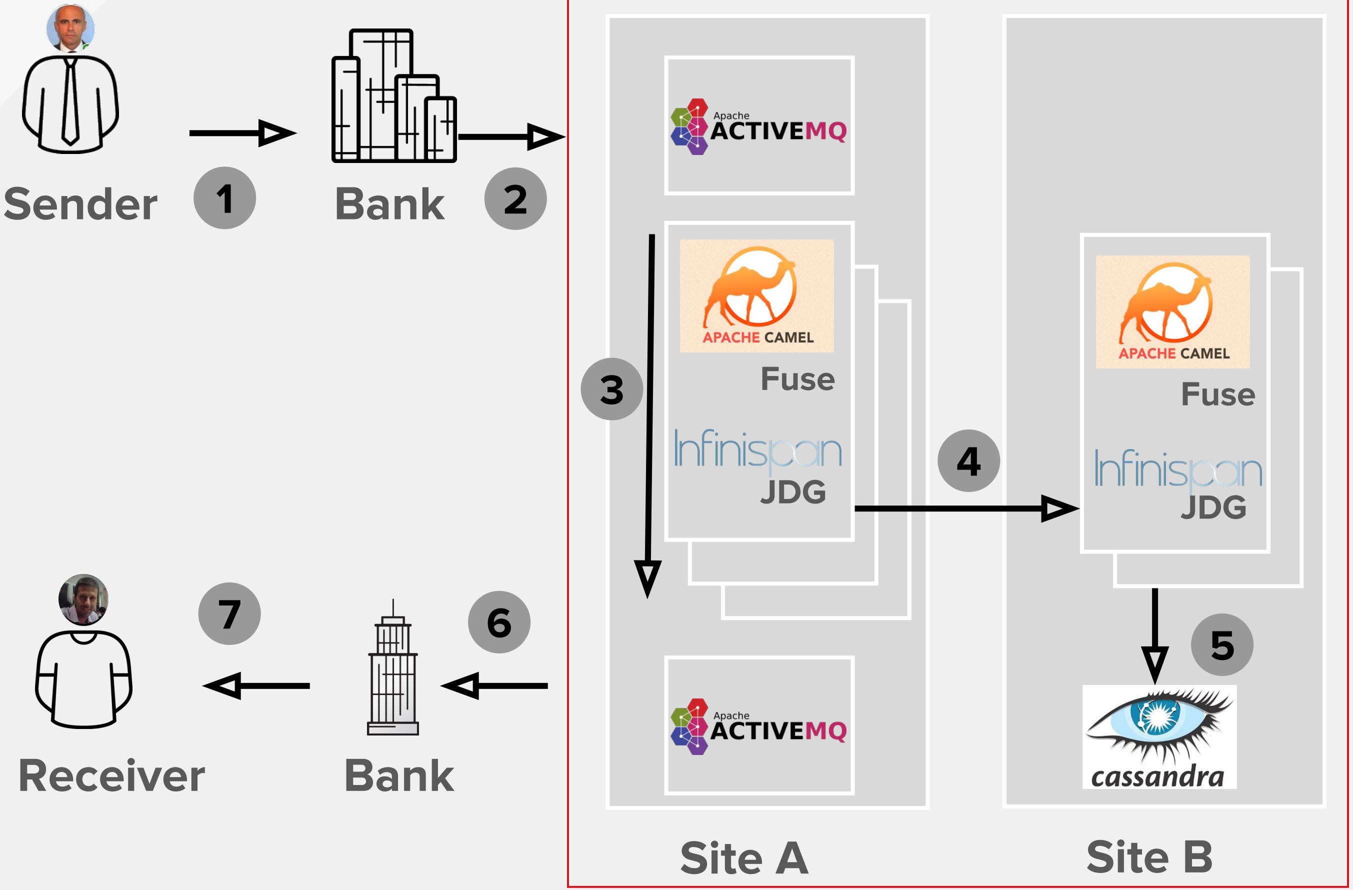
Wire Account Received:

**100 €**

# What’s under the hood

# What’s under the hood

- **JDG**, In Memory Data Grid
  - Scalable and elastic grid for exceptional performances
- **AMQ**
  - High performance messaging
- **FIS**, Camel on OpenShift
  - Superdynamic integrations
- **CASSANDRA,** NoSQL DB
  - Store transaction history for non repudiation / antifraud



1 Request sent to bank

2 Bank calls Instant Payments

3 Request is stored in JDG

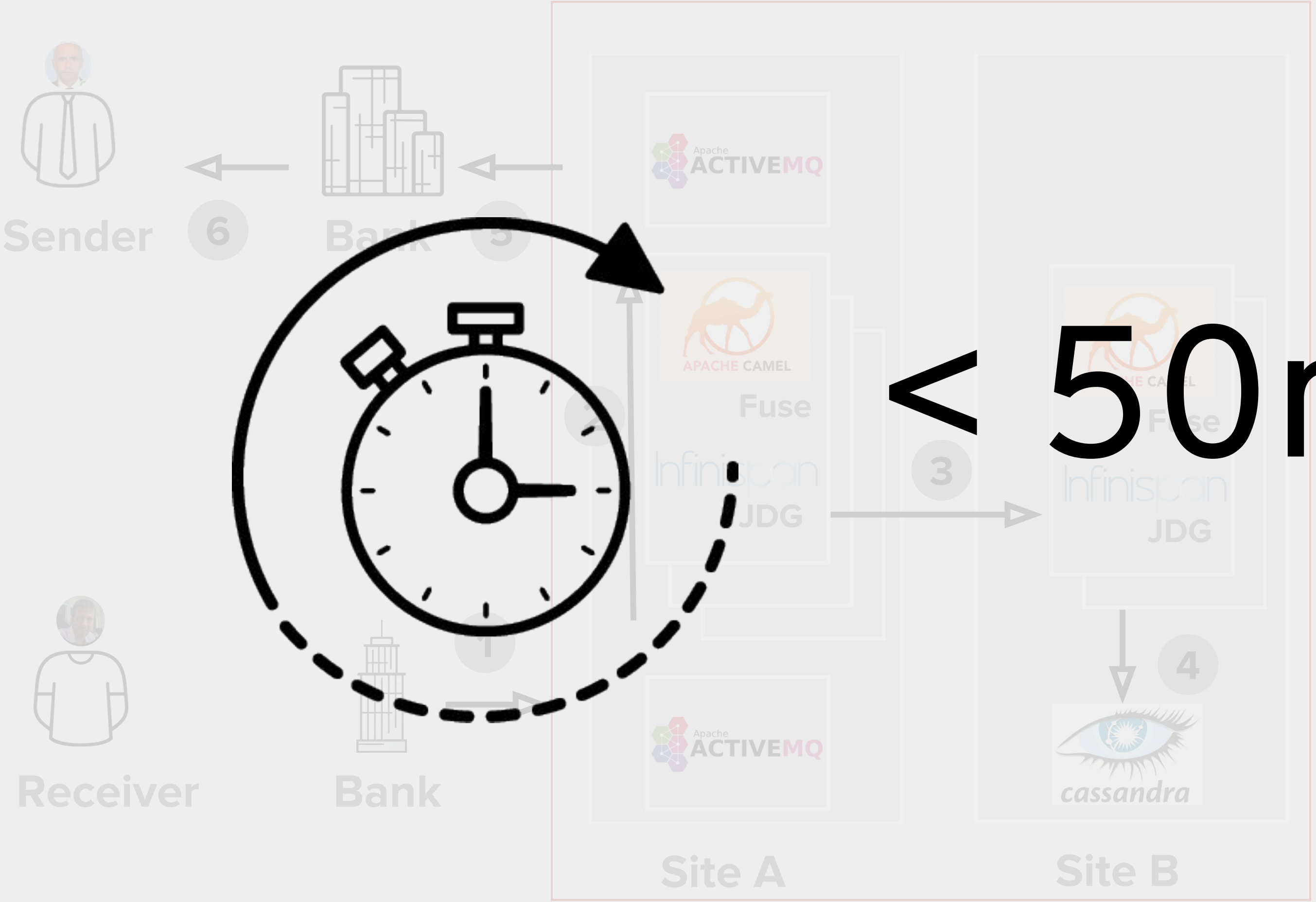
4 JDG replicates cross site

5 Req is stored in Cassandra

6 Tx is propagated to bank

7 Receiver asked for confirm

## Instant Payments



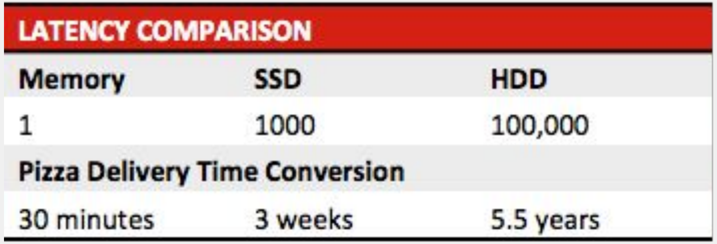
# < 50mS !

1. Bank Confirms Payment
2. Confirmation is stored in JDG
3. JDG replicates cross site
4. Req is stored in Cassandra
5. Tx is propagated to bank
6. Sender is notified

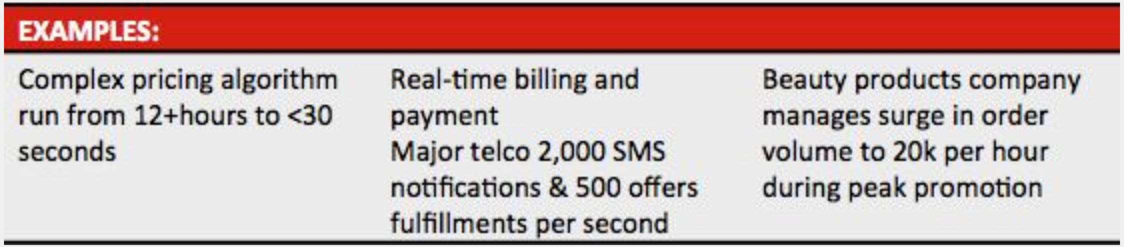
## Instant Payments

# JDG, In Memory Data Grid

# Memory is the new disk

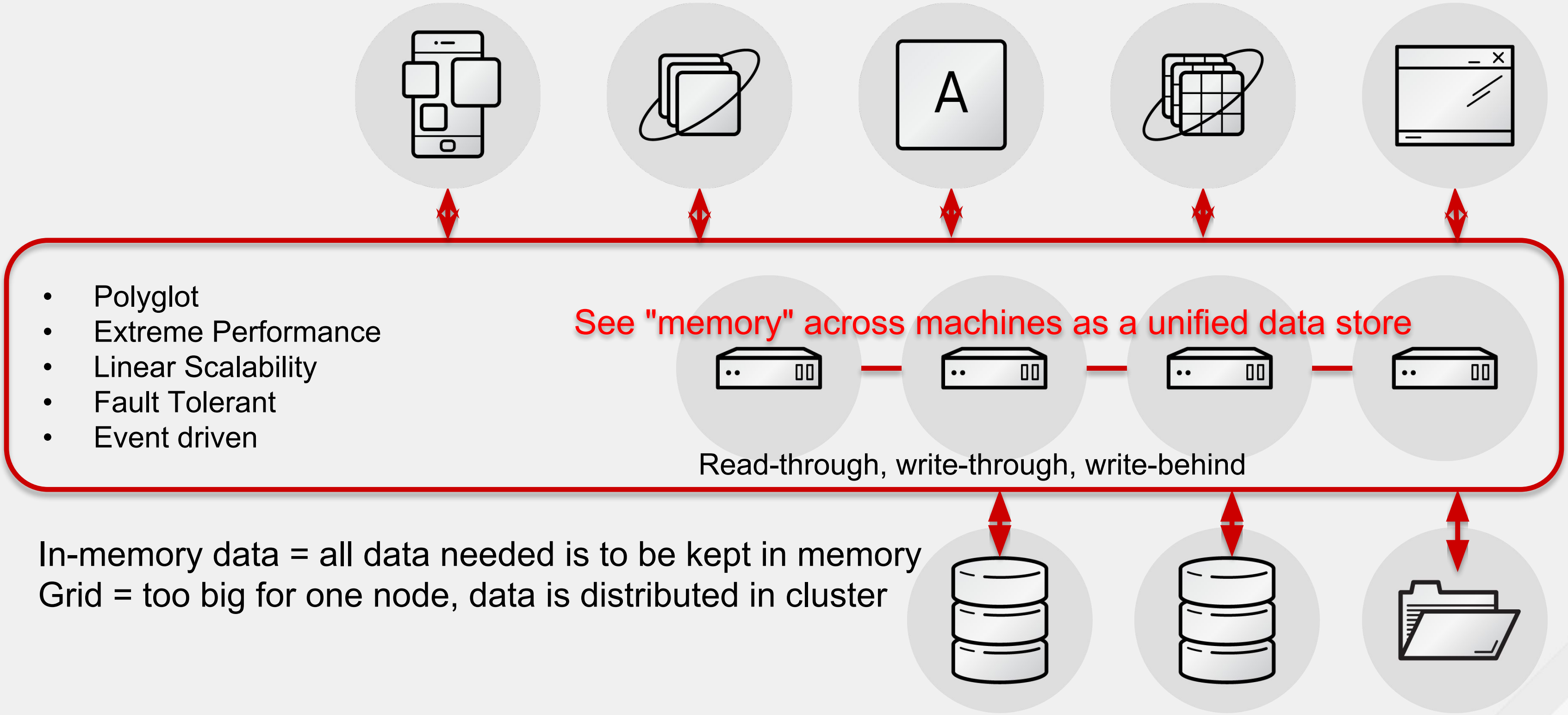


| LATENCY COMPARISON | | |
|---|---|---|
| **Memory** | **SSD** | **HDD** |
| 1 | 1000 | 100,000 |
| **Pizza Delivery Time Conversion** | | |
| 30 minutes | 3 weeks | 5.5 years |



| EXAMPLES: | | |
|---|---|---|
| Complex pricing algorithm run from 12+hours to <30 seconds | Real-time billing and payment<br>Major telco 2,000 SMS notifications & 500 offers fulfillments per second | Beauty products company manages surge in order volume to 20k per hour during peak promotion |

# JDG: JBoss Data Grid



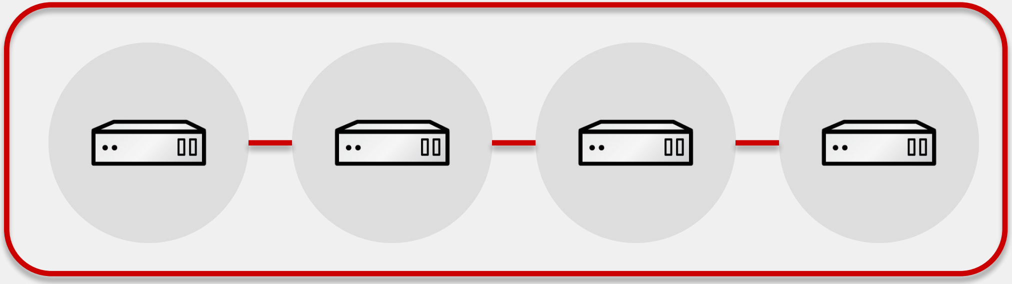
- Polyglot
- Extreme Performance
- Linear Scalability
- Fault Tolerant
- Event driven

See "memory" across machines as a unified data store

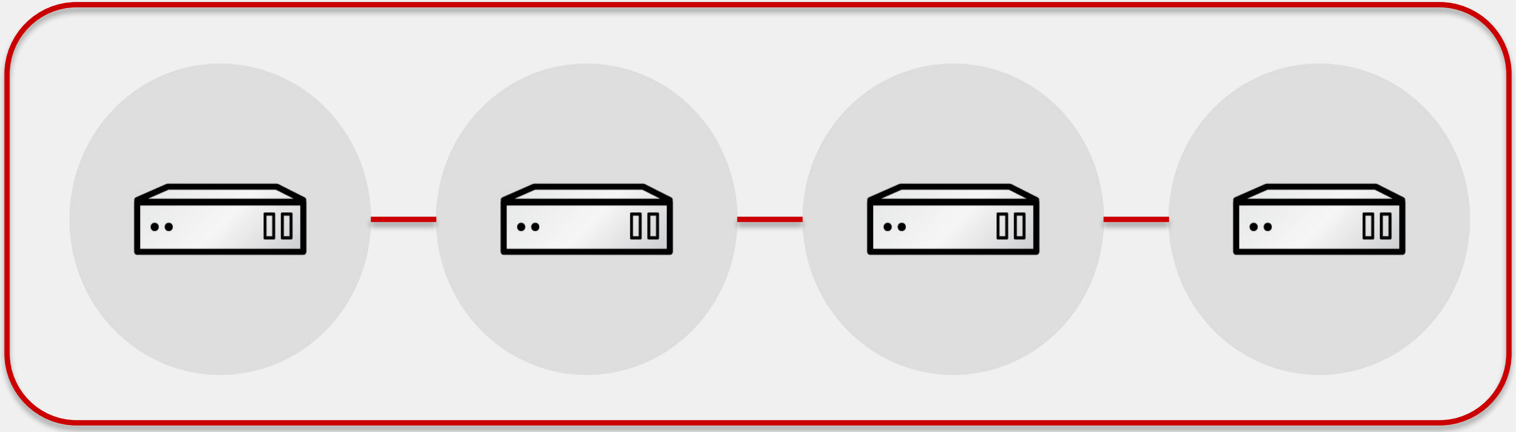
Read-through, write-through, write-behind

In-memory data = all data needed is to be kept in memory
Grid = too big for one node, data is distributed in cluster

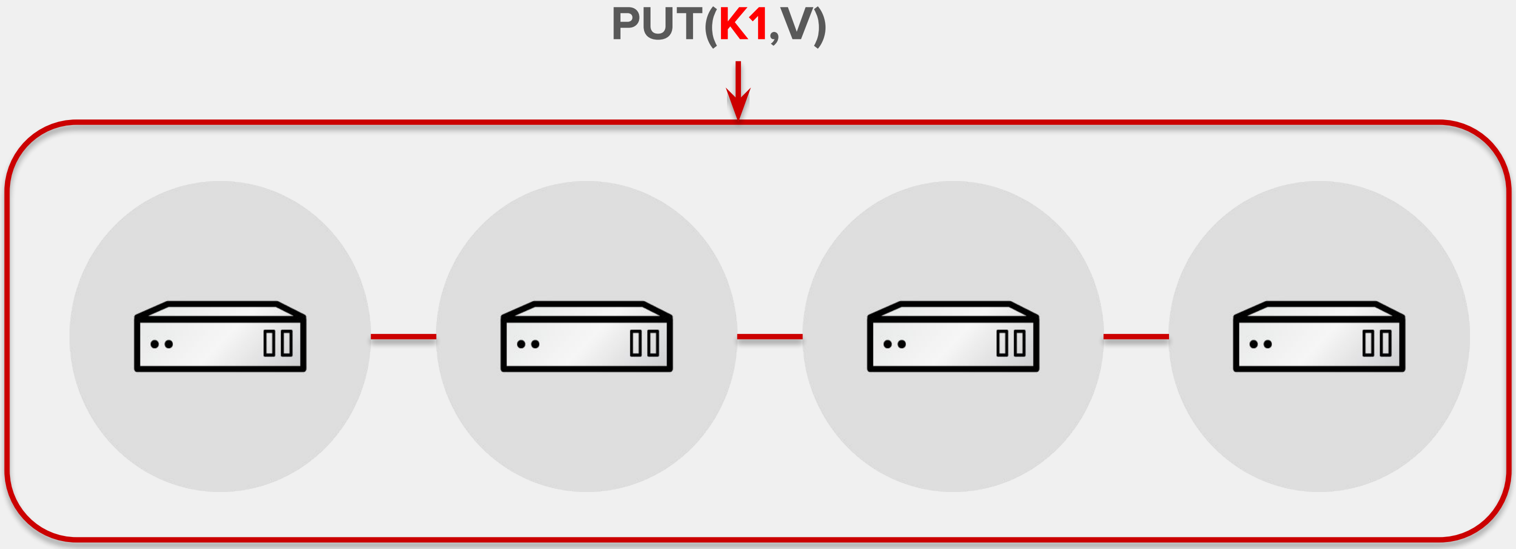
# Memory cluster



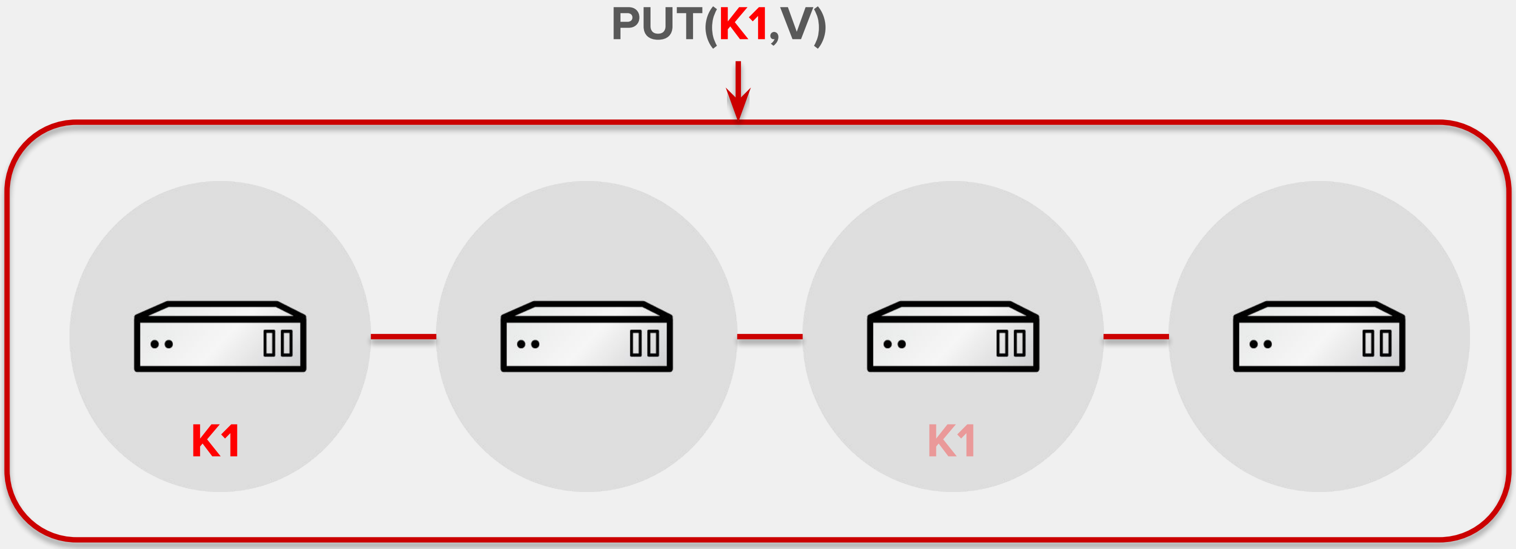
# Distributed mode (typically 1 replica)



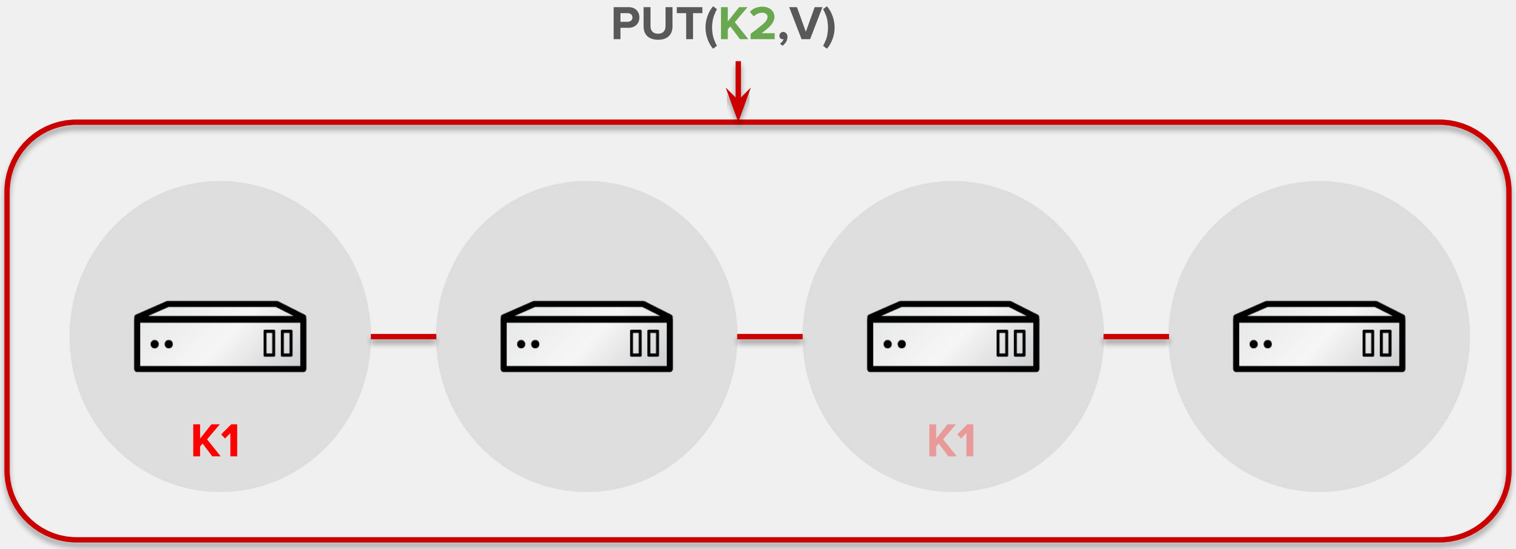
# Adding a value



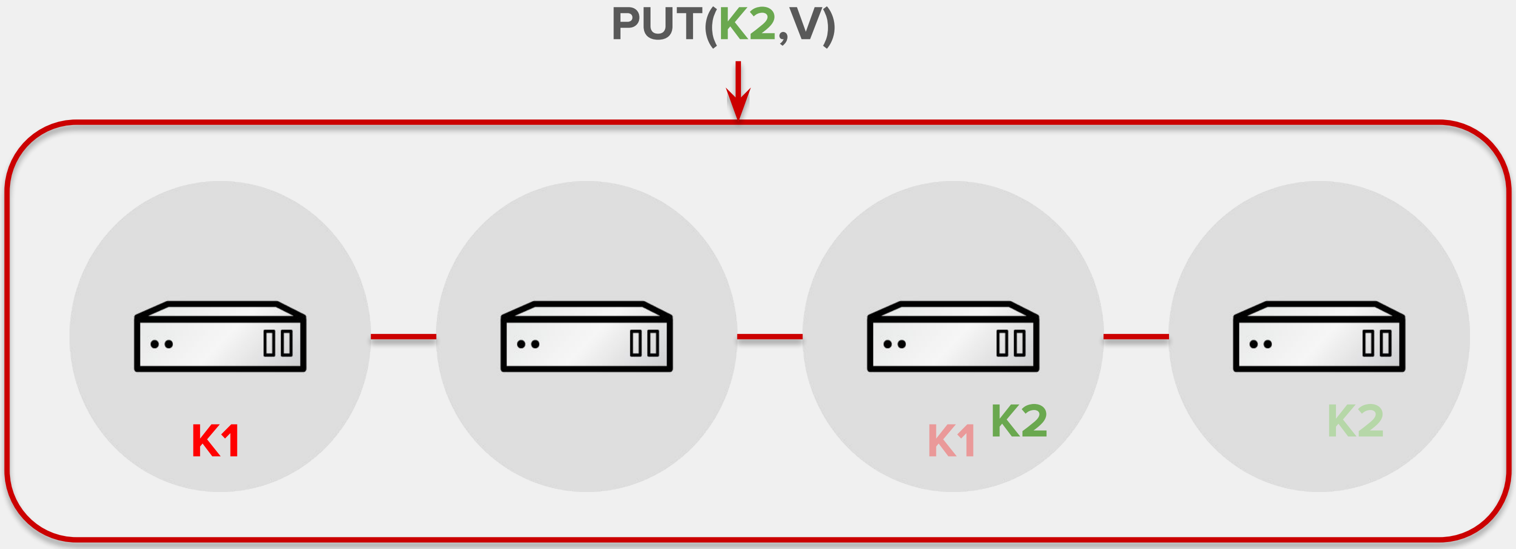
# Adding a value



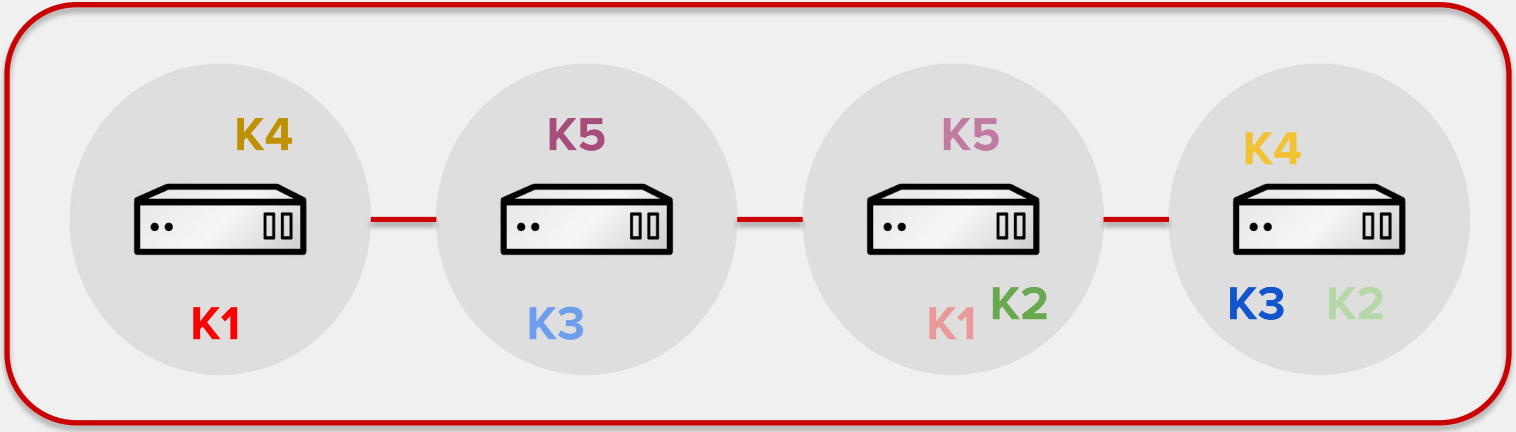
# Values are distributed



# Values are distributed



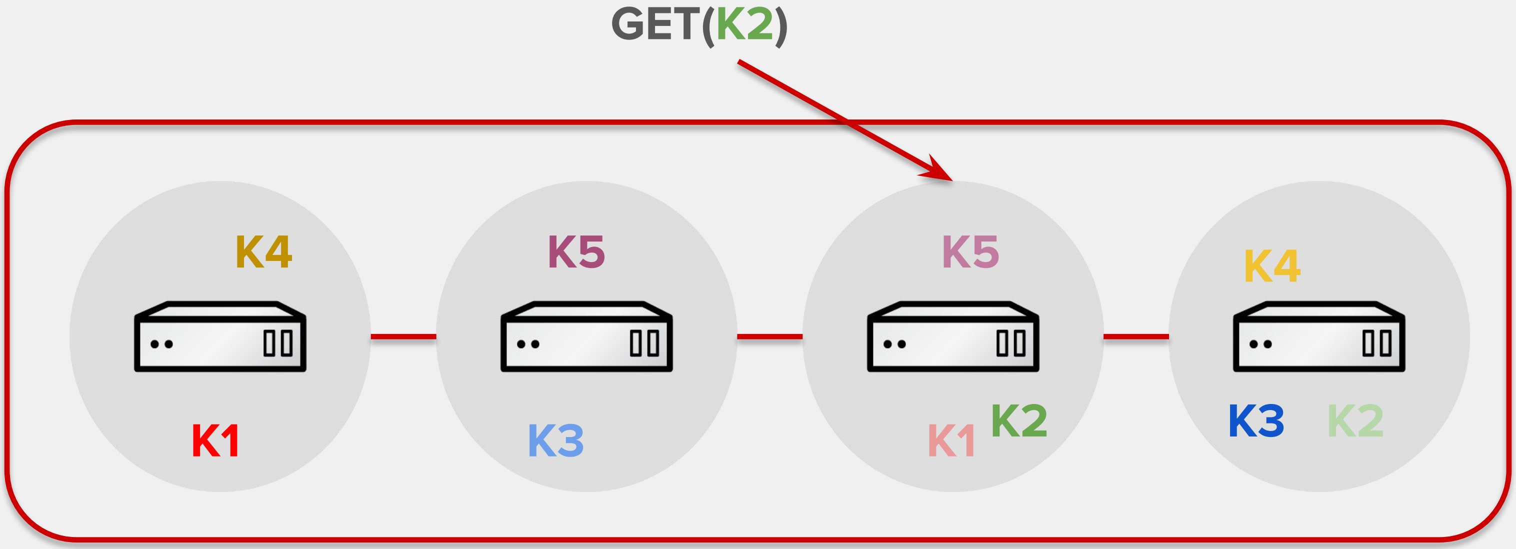
# Keys are distributed consistently through the cluster



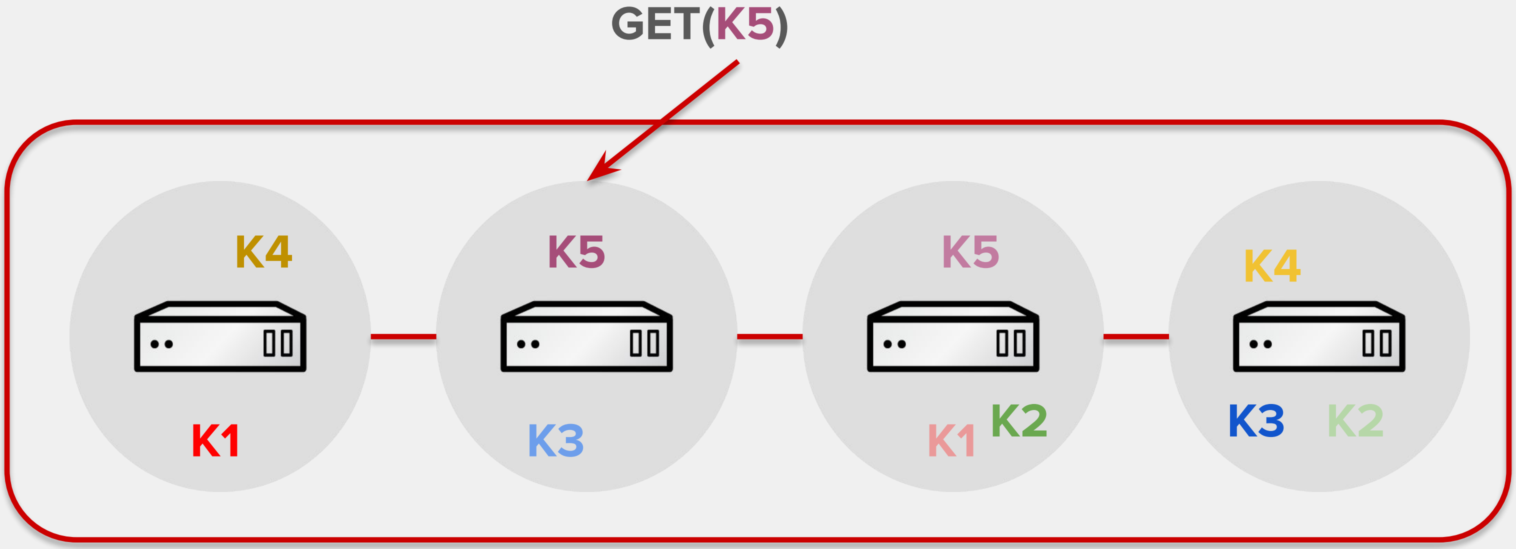
# Consistent Hashing

Given the same topology (i.e. number of nodes), **same** keys are **always** hashed on the **same** nodes. **Consistent hashing** can be calculated on clients too, so they can reach the **right** node for any given **key**

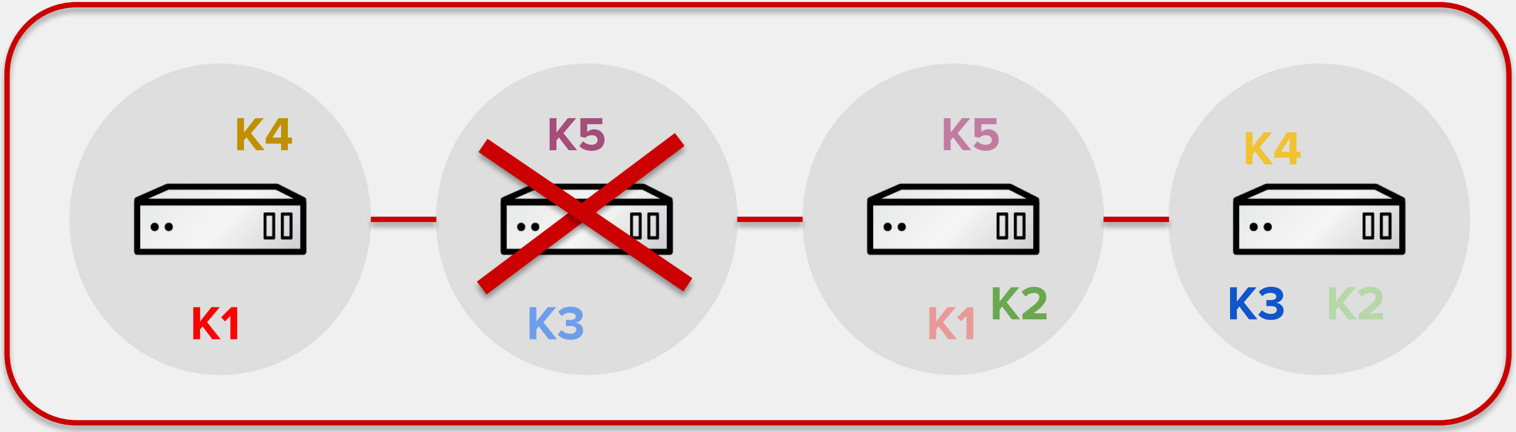
# So reads go directly on the “right” node...



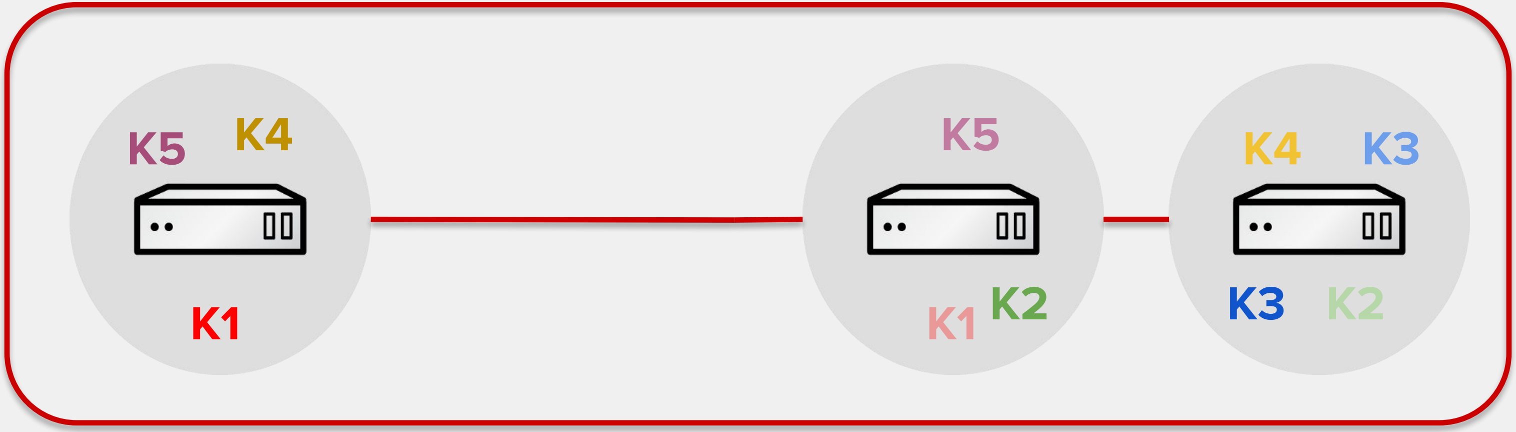
# … thx to consistent hashing



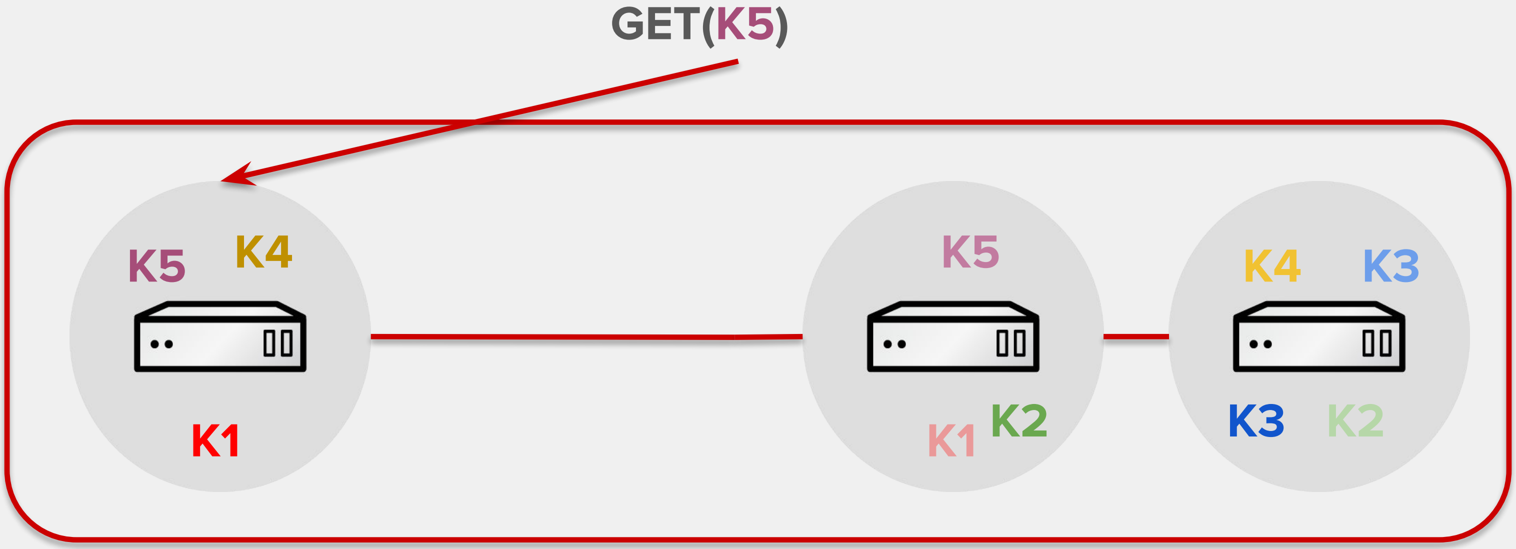
# If you lose or add a node (topology change)...



# ...keys are automatically redistributed



# The client gets the new topology too...



# Data Affinity

Data affinity means **co-locating data** together to improve performance and scalability

Data affinity means **co-locating computing code** with data **too**

# Data Affinity

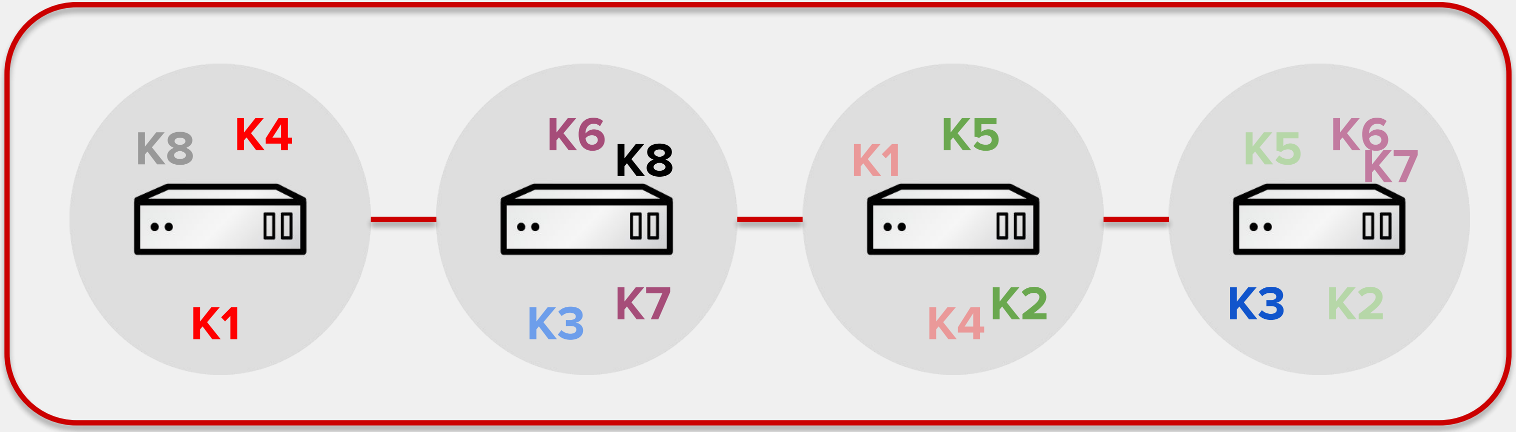
“Grouping” together all the **affine** data, for example:

- All **Customer** data
- All **Credit card** data
- Whatever **partitioning** criteria is better

Gives:

- **Highest** possible performance
- **Lowest** possible round trips

# Same color == same group



# Project Requirements

# Euro Banking Association - RT1

***RT1** is an **instant payment system** that will provide the European payments industry with a pan-European infrastructure platform for **real-time payments** in euro from day one of the SEPA Instant Credit Transfer Scheme.*

# Constraints

**5,000** tx/sec

**Active/Active** geo sites

**27 Mln** Payments/day

**0** Message loss

**< 900 ms** Roundtrip

**24/7/365** Availability

# Challenges and Solutions

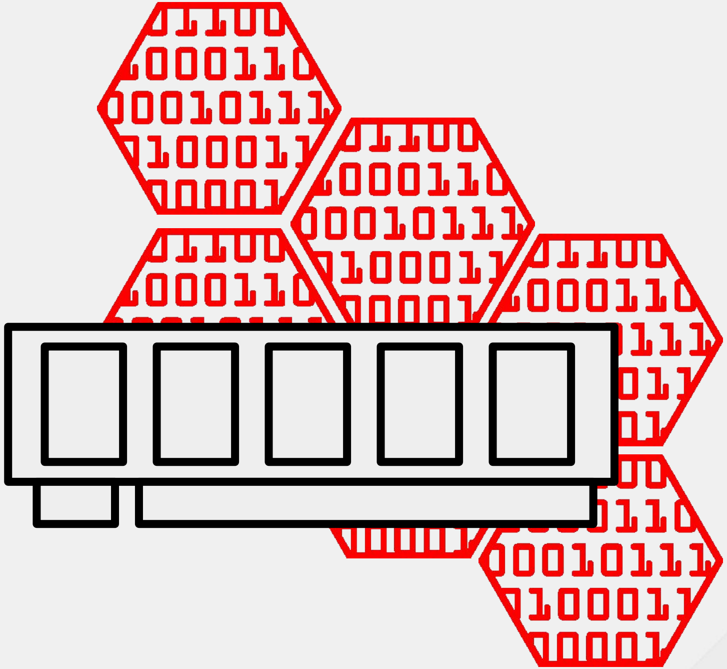
# Challenge

Due to low latency and replica requirements, we could not adopt a traditional db-centric application

# Solution

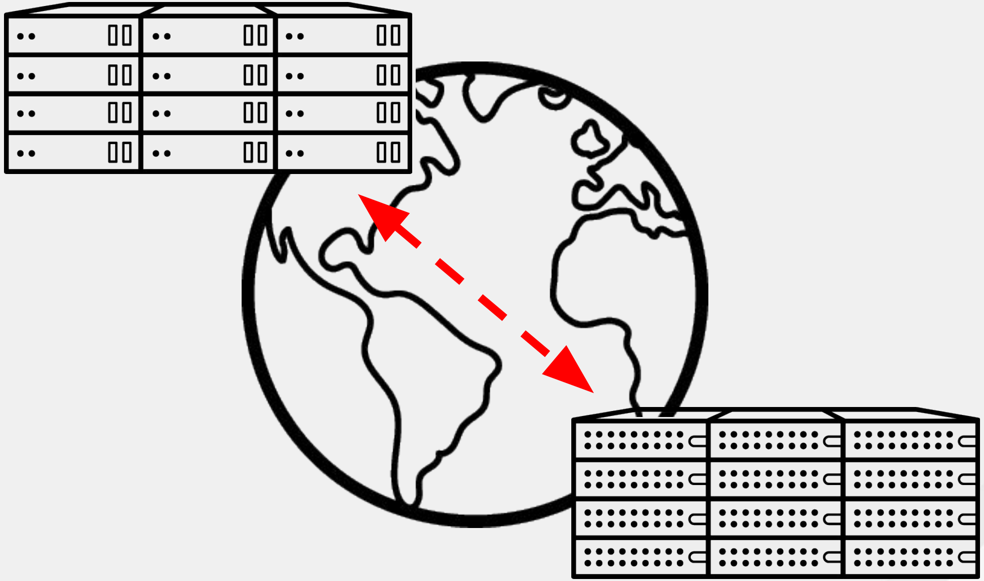
Non DB Centric application

We adopted a full in-memory solution relying on **Jboss DataGrid**



# Challenge

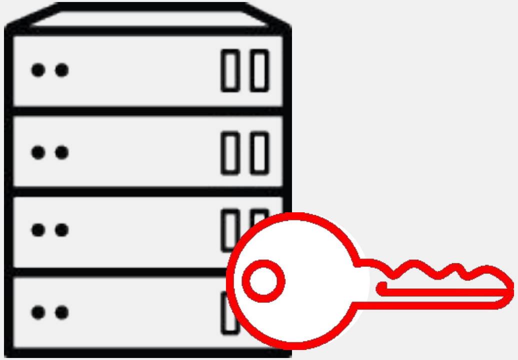
Cross-site replica needs to be synchronous, and the number of remote operations must be low



# Solution

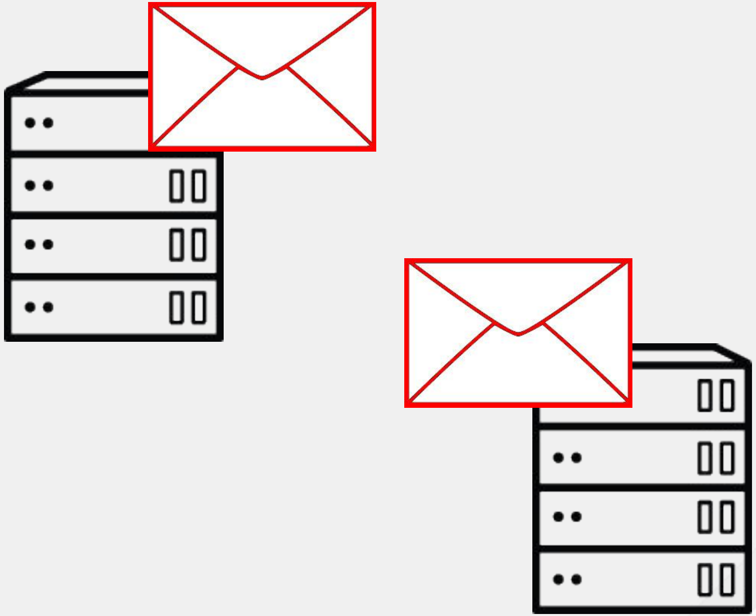
Minimize remote communication

DataGrid key affinity:
Each transaction is handled by the node which owns the relevant data.



# Challenge

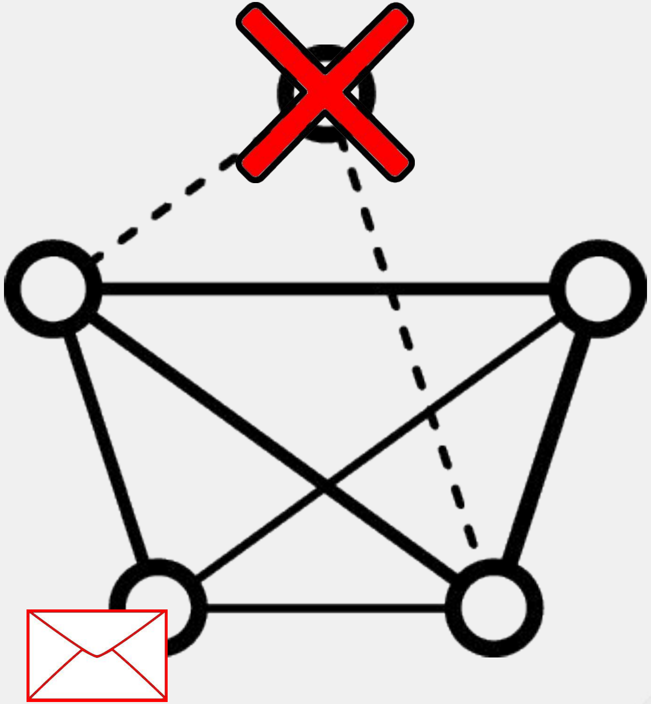
Incoming messages need to be managed in an highly available, cross site infrastructure



# Solution

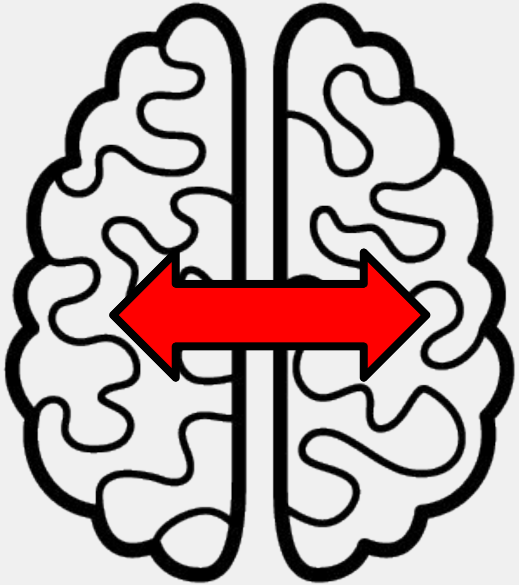
Highly available messaging

AMQ Network of brokers allow highly available, zero message loss topologies (site disaster resilient)



# Challenge

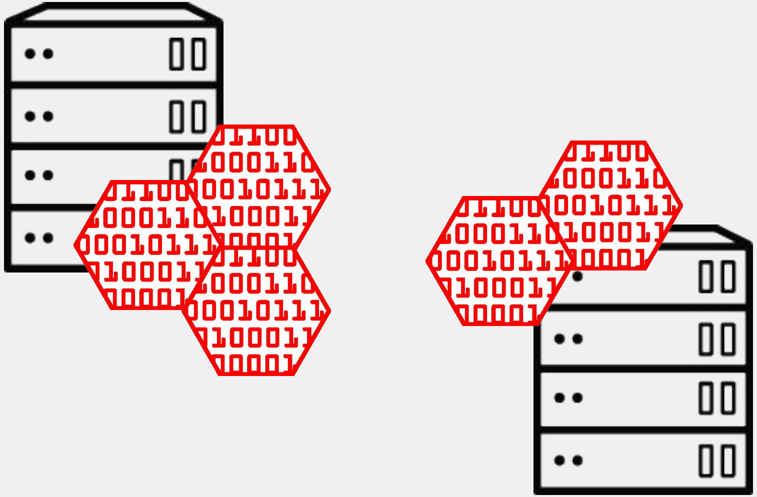
Multi site active-active increases the risks of split brains

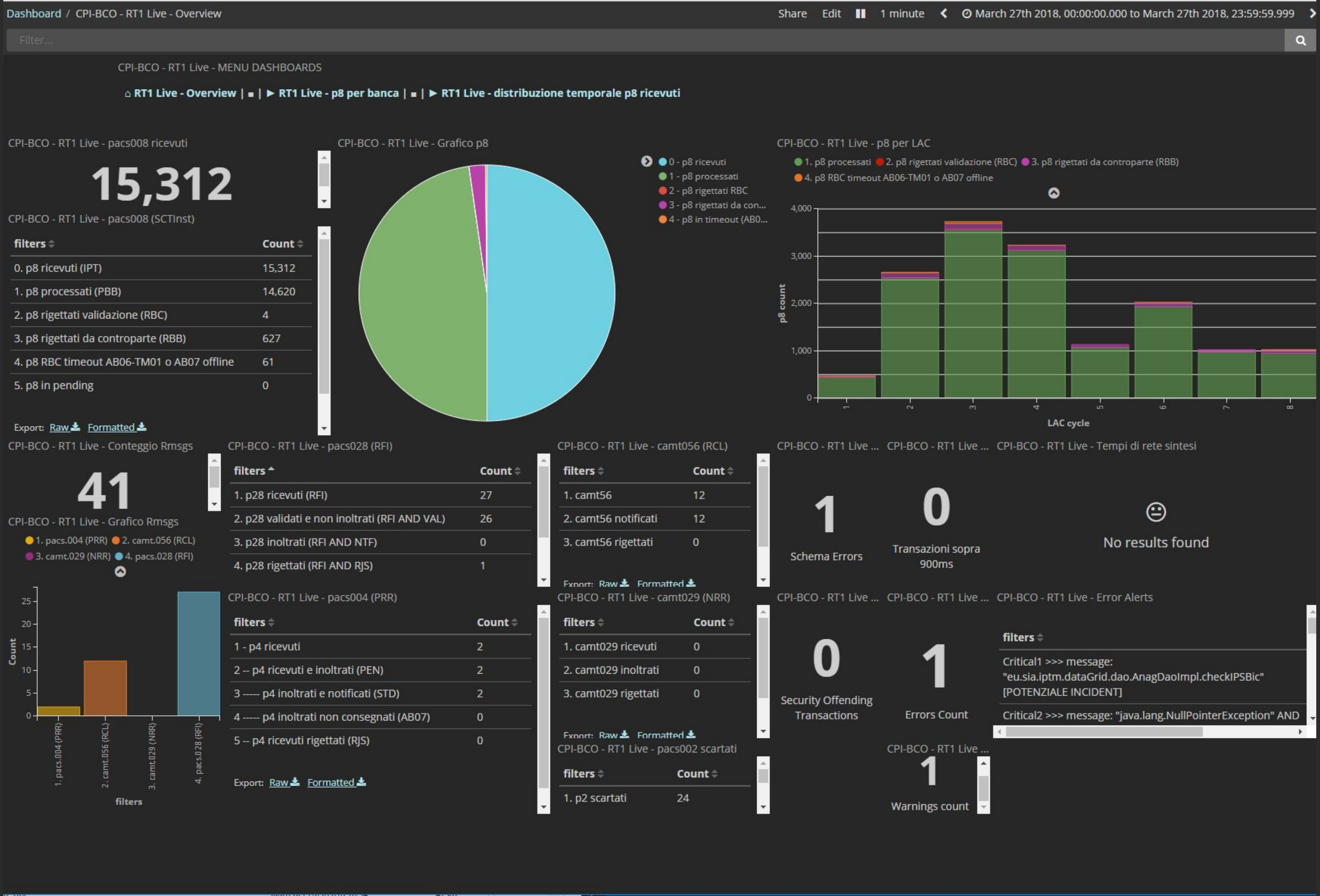


# Solution

Avoid split brain

Datagrid owner distribution between sites allow to operate in case of network split




Dashboard / CPI-BCO - RT1 Live - Overview
Share
Edit
1 minute
March 27th 2018, 00:00:00.000 to March 27th 2018, 23:59:59.999
Filter...

CPI-BCO - RT1 Live - MENU DASHBOARDS

⌂ RT1 Live - Overview | ■ | ► RT1 Live - p8 per banca | ■ | ► RT1 Live - distribuzione temporale p8 ricevuti

CPI-BCO - RT1 Live - pacs008 ricevuti

15,312

CPI-BCO - RT1 Live - pacs008 (SCTInst)

| filters | Count |
|---|---|
| 0. p8 ricevuti (IPT) | 15,312 |
| 1. p8 processati (PBB) | 14,620 |
| 2. p8 rigettati validazione (RBC) | 4 |
| 3. p8 rigettati da controparte (RBB) | 627 |
| 4. p8 RBC timeout AB06-TM01 o AB07 offline | 61 |
| 5. p8 in pending | 0 |

Export: Raw Formatted

CPI-BCO - RT1 Live - Grafico p8

CPI-BCO - RT1 Live - p8 per LAC

● 1. p8 processati
● 2. p8 rigettati validazione (RBC)
● 3. p8 rigettati da controparte (RBB)
● 4. p8 RBC timeout AB06-TM01 o AB07 offline
4,000
3,000
2,000
1,000
0
p8 count
1
2
3
4
5
6
7
8
LAC cycle

CPI-BCO - RT1 Live - Conteggio Rmsgs

41

CPI-BCO - RT1 Live - Grafico Rmsgs

CPI-BCO - RT1 Live - pacs028 (RFI)

| filters | Count |
|---|---|
| 1. p28 ricevuti (RFI) | 27 |
| 2. p28 validati e non inoltrati (RFI AND VAL) | 26 |
| 3. p28 inoltrati (RFI AND NTF) | 0 |
| 4. p28 rigettati (RFI AND RJS) | 1 |

CPI-BCO - RT1 Live - pacs004 (PRR)

| filters | Count |
|---|---|
| 1 - p4 ricevuti | 2 |
| 2 -- p4 ricevuti e inoltrati (PEN) | 2 |
| 3 ----- p4 inoltrati e notificati (STD) | 2 |
| 4 ----- p4 inoltrati non consegnati (AB07) | 0 |
| 5 -- p4 ricevuti rigettati (RJS) | 0 |

Export: Raw Formatted

CPI-BCO - RT1 Live - camt056 (RCL)

| filters | Count |
|---|---|
| 1. camt56 | 12 |
| 2. camt56 notificati | 12 |
| 3. camt56 rigettati | 0 |

Export: Raw Formatted

CPI-BCO - RT1 Live - camt029 (NRR)

| filters | Count |
|---|---|
| 1. camt029 ricevuti | 0 |
| 2. camt029 inoltrati | 0 |
| 3. camt029 rigettati | 0 |

Export: Raw Formatted

CPI-BCO - RT1 Live - pacs002 scartati

| filters | Count |
|---|---|
| 1. p2 scartati | 24 |

CPI-BCO - RT1 Live ...

1

Schema Errors

CPI-BCO - RT1 Live ...

0

Transazioni sopra 900ms

CPI-BCO - RT1 Live - Tempi di rete sintesi

No results found

CPI-BCO - RT1 Live ...

0

Security Offending Transactions

CPI-BCO - RT1 Live ...

1

Errors Count

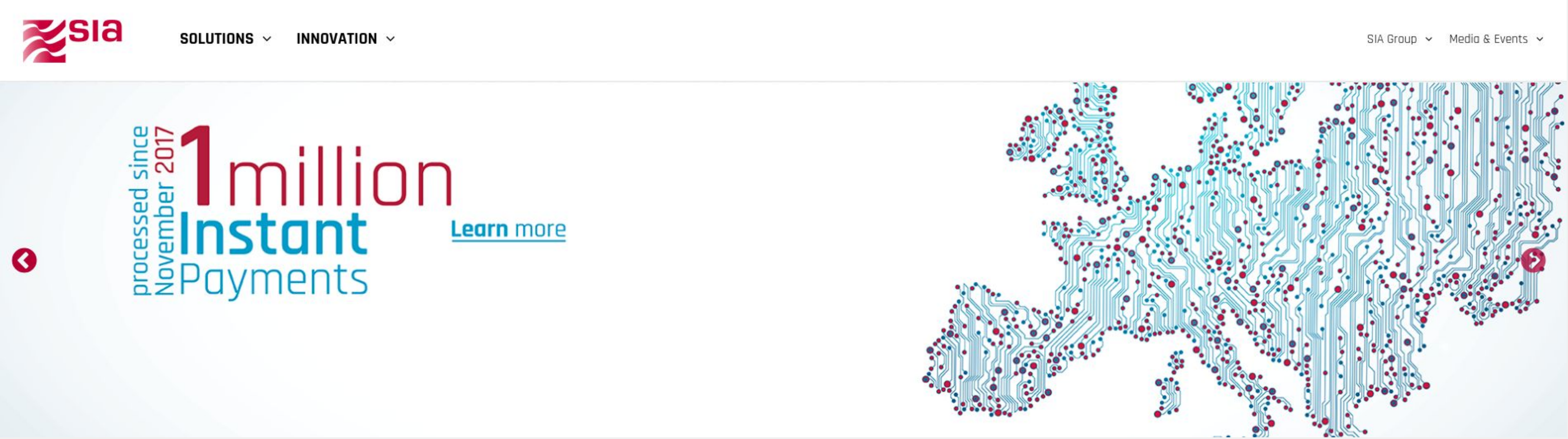
CPI-BCO - RT1 Live ...

1

Warnings count

CPI-BCO - RT1 Live - Error Alerts

| filters |
|---|
| Critical1 >>> message: "eu.sia.iptm.dataGrid.dao.AnagDaoImpl.checkIPSBic" [POTENZIALE INCIDENT] |
| Critical2 >>> message: "java.lang.NullPointerException" AND |



processed since November 2017

1 million Instant Payments

Learn more

# Next Steps

# Next Steps

EnMasse

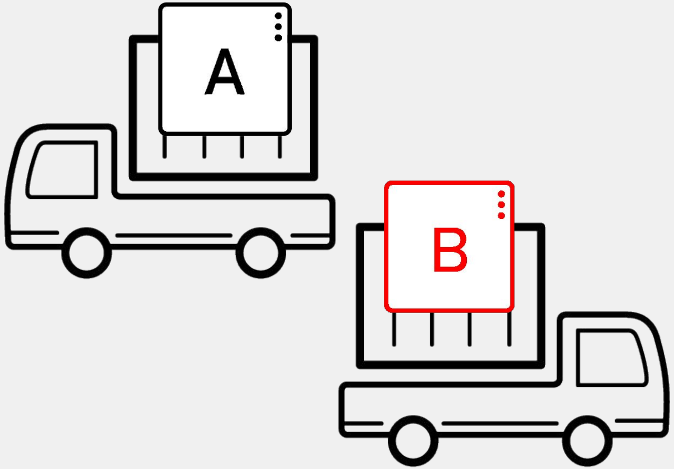
Adoption of **EnMasse** (Messaging As A Service in OpenShift), to streamline the management of queues

enmasse

# Next Steps

Release and scale, without service disruption

Containerization with OpenShift to allow safer releases of newer version and instances



# Next Steps

Broader Circuit

Federate other payment circuits, in order to reach more customers

RED HAT SUMMIT

# THANK YOU

g+ plus.google.com/+RedHat

f facebook.com/redhatinc

in linkedin.com/company/red-hat

twitter.com/RedHat

YouTube youtube.com/user/RedHatVideos